Форма № 24

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій
(повне найменування факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка
до дипломного проєкту (роботи)
бакалавр
(ступінь вищої освіти)

на тему ПРОГРАМНА РЕАЛІЗАЦІЯ FRONT-END ЧАСТИНИ ІТ-ФОРУМУ
(назва теми)
SOFTWARE IMPLEMENTATION OF A FRONTEND PART OF IT-FORUM

Виконав(ла): студент(ка) 4 курсу, групи КНТ-229

Спеціальності 122 Комп'ютерні науки
(код і найменування спеціальності)

Освітня програма (спеціалізація)
Комп'ютерні науки

ЗАЙЦЕВ З. В.
(ПРІЗВИЩЕ та ініціали)

Керівник ФЕДОРОНЧАК Т. В.
(ПРІЗВИЩЕ та ініціали)

Рецензент ШИТІКОВА О. В.
(ПРІЗВИЩЕ та ініціали)

2023

Форма № 25

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет ФКНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 122 Комп'ютерні науки
(код і найменування)
Освітня програма (спеціалізація) Комп'ютерні науки
(назва освітньої програми (спеціалізації)

ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ, д.т.н, проф.
_____Сергій СУББОТІН
«_____» _____20 23 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ЗАЙЦЕВА Захара Володимировича
(ПРІЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Програмна реалізація front-end частини IT-форуму. Software Implementation of a Frontend Part of IT-forum

керівник проєкту (роботи) к.т.н., доцент, ФЕДОРОНЧАК Тетяна Василівна,
(науковий ступінь, вчене звання, ПРІЗВИЩЕ, ім'я, по батькові)
затверджені наказом закладу вищої освіти від « 04 » квітня 20 23 року № 87
2. Строк подання студентом проєкту (роботи) 29 травня 2023 року
3. Вихідні дані до проєкту (роботи) технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області, 2. Вибір і обґрунтування структури системи, 3. Розробка програмного забезпечення, 4. Керівництво програміста. 5. Керівництво користувача.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)
Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРІЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	ФЕДОРОНЧАК Т. В., доцент		
Нормоконтроль	ДЕЙНЕГА Л. Ю., ст. викл.		

7. Дата видачі завдання « 04 » квітня 20 23 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір технологій розробки.	2 тиждень	Розділ 2
4	Розробка архітектури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділ 3, 4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4, 5
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка) _____ Захар ЗАЙЦЕВ
(підпис) (Ім'я ПРІЗВИЩЕ)

Керівник проєкту (роботи) _____ Тетяна ФЕДОРОНЧАК
(підпис) (Ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 202 с., 72 рис., 7 табл., 3 дод., 20 джерел.

BOOTSTRAP, BLADE, ДИЗАЙН, FRONT-END, ІНТЕРФЕЙС, АДАПТИВНІСТЬ, МАКЕТ, ВЕРСТКА, КОНТЕНТ.

Об’єкт дослідження – онлайн-платформа для обговорення тем на IT-тематику.

Предмет дослідження – клієнтська частина форуму для IT-фахівців.

Мета роботи – розробка дизайну для форуму IT-фахівців та імплементація клієнтської частини форуму.

Матеріали, методи та технічні засоби: менеджер пакетів прикладного рівня Composer, Blade шаблонізатор, мова стилів CSS, об'єктноорієнтована прототипна мова програмування JavaScript, мова розмітки HTML, набір інструментів Bootstrap, візуальний редактор TinyMCE, графічний редактор Photoshop 2021.

Результати. З використанням зазначених технологій було розроблено вебінтерфейс для зручної навігації, створювання та обговорення тем, з урахуванням методів адаптивності та pixel-perfect верстки.

Висновки. Реалізовано ефективне та привабливе дизайнерське рішення для онлайн-платформи з обговорення тем на IT-тематику, що осучаснює погляд на форуми, роблячи пошук інформації максимально спрощеним та вільним від зайвих деталей.

Галузь використання – поліпшення користувацького досвіду на IT-форумах, спільнотах або блогах, з можливістю використання в освітніх та наукових цілях.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 202 pages, 72 figures, 7 tables, 3 appendixes, 20 sources.

BOOTSTRAP, BLADE, DESIGN, FRONT-END, INTERFACE, ADAPTIVENESS, LAYOUT, LAYOUT, CONTENT.

The object of the study is an online platform for discussing IT topics.

The subject of the research is the client part of the forum for IT specialists.

The purpose of the work is to develop a design for the forum of IT specialists and implement the client part of the forum.

Materials, methods and technical means: application-level package manager Composer, Blade templater, CSS style language, JavaScript object-oriented prototyping language, HTML markup language, Bootstrap toolset, TinyMCE visual editor, Photoshop 2021 graphic editor.

The results. With the use of these technologies, a web interface was developed for convenient navigation, creation and discussion of topics, taking into account the methods of adaptability and pixel-perfect layout.

Conclusions. An effective and attractive design solution has been implemented for an online platform for discussing IT topics, which modernizes the look at forums, making the search for information as simple as possible and free from unnecessary details.

Field of use – improving the user experience on IT forums, communities or blogs, with the possibility of use for educational and scientific purposes.

ЗМІСТ

Реферат 4
Зміст 6
Перелік скорочень та умовних познак 9
Вступ 10
1 Аналіз предметної області 12
1.1 Опис предметної області 12
1.2 Імплементація front-end частини форуму 15
1.2.1 Розгляд функціоналу форумів 15
1.2.2 Розробка інтерфейсу користувача 18
1.2.3 Оптимізація швидкості роботи 22
1.3 Аналіз існуючих IT-форумів 24
1.3.1 Огляд Reddit Programming 24
1.3.2 Огляд GitHub Community Forum 26
1.3.3 Огляд Stack Overflow 27
1.4 Висновки за розділом 29
2 Вибір і обґрунтування структури системи 31
2.1 Вибір мови розробки 31
2.1.1 Мова розмітки 31
2.1.2 Мова стилів 32
2.1.3 Мова додаткової логіки 34
2.2 Вибір фреймворку для розробки клієнтської частини сайту 36
2.3 Вибір додаткових інструментів 38
2.3.1 Набір інструментів Bootstrap 38
2.3.2 Візуальний редактор тексту TinyMCE 6 40
2.3.3 Менеджер пакетів прикладного рівня 41
2.3.4 Контроль версій 43
2.4 Схематична структура IT-форуму 46
2.5 Висновки за розділом 50
3 Розробка програмного забезпечення 52
3.1 Загальні відомості про розробку 52
3.2 Розроблена логіка сайту 57

3.3.1 Скрипт logicAuth.js 57
3.3.2 Скрипт logicCategories.js 57
3.3.3 Скрипт logicDescription.js 58
3.3.4 Скрипт logicPost.js 58
3.3.5 Скрипт logicPostPage.js 64
3.3 Механізм ролей у front-end 65
3.4 Реалізація механізму запитів на сервер 67
3.2.1 Довідка про роботу проєкту 67
3.2.2 Форми видалення посту 68
3.2.3 Форми видалення закладок 68
3.2.4 Форма пошуку 69
3.2.5 Форма створення категорії 69
3.2.6 Форма редагування категорії 70
3.2.7 Форма реєстрації 70
3.2.8 Форма входу до облікового запису 71
3.5 Налаштування редактору Tiny MCE 73
3.5.1 Основні відомості про налаштування 73
3.5.2 Панель редагування публікації 74
3.5.3 Панель редагування коментарів 75
3.6 Візуальна складова клієнтської частини 75
3.6.1 Головна сторінка 75
3.6.2 Сторінка посту 77
3.6.3 Сторінка списку користувачів 79
3.6.4 Сторінка категорій 80
3.6.5 Сторінка створення теми 84
3.6.6 Сторінка опису сайту 84
3.6.7 Сторінка профілю користувача 86
3.6.8 Сторінка закладок 89
3.6.9 Сторінка входу до облікового запису 90
3.6.10 Сторінка реєстрації нового облікового запису 91
3.6.11 Форми помилок 93
3.6.12 Підвал 94
3.7 Висновки за розділом 94
4 Керівництво програміста 95

4.1 Призначення та умови використання 95
4.2 Встановлення та початок роботи 95
4.3 Розташування файлів у структурі проєкту 96
4.4 Підключення ресурсів 97
4.5 Висновки за розділом 98
5 Керівництво користувача 99
5.1 Обліковий запис користувача 99
5.2 Створення теми 103
5.3 Взаємодія з постами 105
5.4 Список категорій 107
5.5 Висновки за розділом 108
Висновки 109
Перелік джерел посилання 110
Додаток А Технічне завдання 112
Додаток Б Текст програми 116
Додаток В Слайди презентації 195

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

IT – Information Technology
JS – JavaScript
CSS – Cascading Style Sheets
HTML – HyperText Markup Language
UI – User Interface
UX – User Experience
JPG – Joint Photographic Experts Group
PNG – Portable Network Graphics
SVG – Scalable Vector Graphics
HTTP – HyperText Transfer Protocol
GIF – Graphics Interchange Format
phpBB – PHP Bulleting Board
PHP – Hypertext Preprocessor
CDN – Content Delivery Network
XSS – Cross Site Scripting
SVN – Subversion
SASS – Syntactically Awesome Style Sheets
LESS – Leaner Style Sheets
PEAR – PHP Extension and Application Repository
ORM – Object-Relational Mapping
SPA – Single Page Applications
CSRF – Cross-Site Request Forgery

ВСТУП

Комунікація між людьми завжди була важливим елементом життя. Завдяки розвитку технологій, сьогодні ми можемо легко і швидко зв'язатися з ким завгодно у будь-якій точці світу. Інтернет-форуми, зокрема й ті, що спрямовані на сферу інформаційних технологій, є одним з найпоширеніших засобів комунікації між людьми зі схожими інтересами. Вони дозволяють знайти відповіді на будь-які запитання, обговорити актуальні теми та проблеми з іншими учасниками спільноти, отримати поради від професіоналів, а також поділитися своїм досвідом та знаннями з іншими. Це потужний інструмент для розвитку та підвищення кваліфікації в галузі ІТ. Проте, у всіх наявних форумах є ряд проблем, які можуть зіпсувати користувацький досвід.

Складність використання форуму може впливати на взаємодію нових користувачів з існуючими. Не всі форуми мають інтуїтивно зрозумілі та прості інтерфейси, які дозволяють легко створювати нові теми або відповіді на питання. Це може призвести до того, що новачки не зможуть почувати себе комфортно на форумі.

Через погану оптимізацію сайту виникає проблема повільної швидкості завантаження сторінок, що може також впливати на взаємодію користувачів з форумом. Якщо сторінки завантажуються повільно, користувачі можуть втратити інтерес і звернутися до інших джерел інформації.

Недостатній захист користувальницьких даних та відсутність заходів проти зловмисних атак можуть призвести до витоку конфіденційної інформації, що може відбитися на іміджі форуму та його популярності серед користувачів.

Також існує проблема застарілості дизайну на форумах, що проявляється не тільки в тому, що дизайн може бути непривабливим та нецікавим, але й неадаптованим до сучасних стандартів та вимог

користувачів. Форуми є ключовими платформами для обміну знаннями та досвідом між програмістами та розробниками, а також для пошуку розв’язань технічних проблем. Якщо форуми не вдосконалюватимуть свій інтерфейс та функціональність, це може привести до зменшення ефективності інформаційного обміну та зниження якості програмного забезпечення, що розробляється.

Тож, виходячи з низки проблем, що були зазначені вище, було прийнято рішення розробити користувальницький інтерфейс для форуму, беручи до уваги сучасні практики для побудови дизайну та досвіду користувача [1].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис предметної області

Вебдизайн – розділ дизайну, що має під собою процес створення вебсайтів та розробку зовнішнього вигляду вебсторінок, включаючи їх структуру, макет, кольорову гаму, шрифти, графіку, фотографії та інші елементи.

Вебсторінка - це документ у форматі HTML, який можна переглянути в Інтернеті за допомогою веббраузера. Вона може містити текст, зображення, відео та інші мультимедійні елементи, які відображаються на екрані користувача. Вебсторінки можуть бути динамічними, що означає, що вони можуть мінятися в залежності від дій користувача або стану бази даних.

UI/UX дизайн – це проєктування інтерфейсу та взаємодії користувача з вебсайтом або додатком з метою забезпечення зручного та легкого користування.

Верстка - це процес створення HTML-коду, який відображається веббраузером і візуально відображає структуру та зміст вебсторінки. Вона містить в собі розміщення тексту, зображень, відео та іншого контенту на вебсторінці та розташування їх у відповідному порядку та форматування їх з використанням CSS.

Інтернет-форум – це сайт або застосунок, що дозволяє користувачам обговорювати різні теми, обмінюватися думками та інформацією, ставити питання і відповідати на них. Форуми можуть бути створені для різних цілей, включаючи обговорення конкретної теми, розміщення оголошень, підтримки спільнот тощо.

У вебфорумів, як і у більшості вебзастосунків, застосовується клієнт-серверна архітектура (див. рис. 1.1). Вона є найпоширенішою моделлю в розробці програмного забезпечення. За цією моделлю, взаємодія між програмою та її користувачем відбувається через дві частини: клієнтську та серверну [2].

Сторона серверу відповідає за обробку запитів від клієнтської частини та надання відповідей. У вебфорумах серверна сторона виконує завдання розміщення та зберігання повідомлень, контролює доступ користувачів до форуму, забезпечує безпеку взаємодії.

Клієнтська частина форуму - це програмне забезпечення, що запускається на браузері користувача та дозволяє відправляти запити на сервер та отримувати відповіді. Вона складається з інтерфейсу користувача, що надає можливість користувачам переглядати та взаємодіяти з форумом, а також з клієнтської логіки, що виконує різні дії з даними, отриманими від серверної частини.

Обмін даними між клієнтом та сервером відбувається за допомогою протоколу HTTP, що дозволяє передавати дані від клієнта до сервера та назад [2].

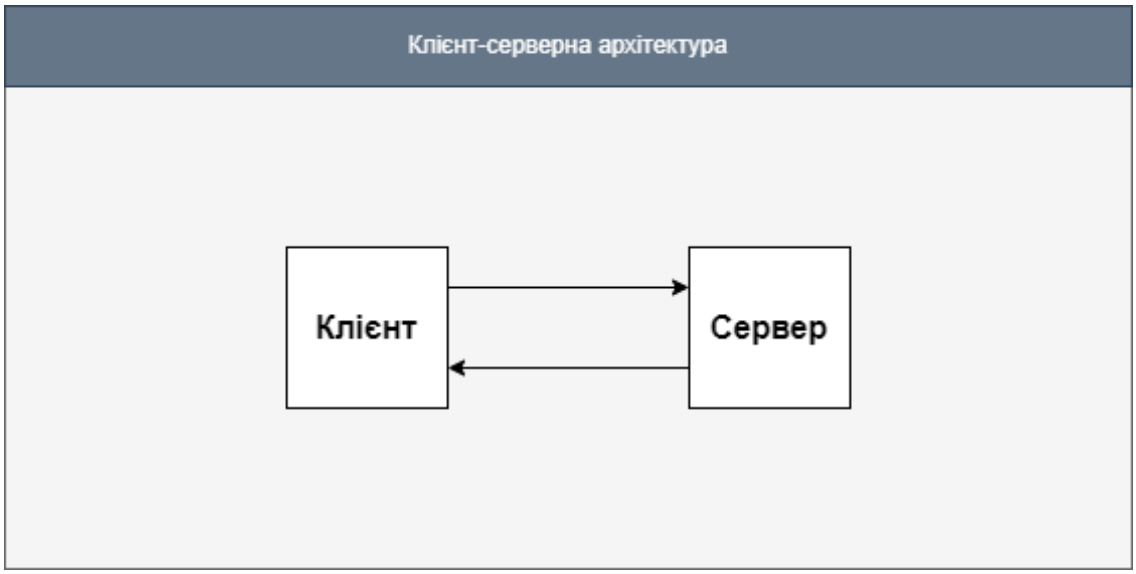

Рисунок 1.1 – Схема клієнт-серверної архітектури

Варто також згадати, які саме функції користувачі мають. Наприклад, вони можуть створювати свої облікові записи, щоб мати доступ до різних функцій, таких як створення нових постів та публікацій, коментування наявних, редагування своїх власних повідомлень та багато іншого.

Створення нових постів та публікацій може бути корисним для того, щоб поділитися своїми думками, ідеями, запитаннями або ж просто для того, щоб почати дискусію на якусь тему. Це може бути дуже корисно, особливо для людей, які хочуть отримати допомогу або поради від інших користувачів.

Крім того, користувачі можуть коментувати існуючі пости та публікації, щоб висловлювати свої думки або надавати додаткову інформацію на тему дискусії. Редагування повідомлень також є важливою функцією, оскільки воно дозволяє виправити будь-які помилки або додати додаткову інформацію після публікації.

Усі ці функції дозволяють користувачам взаємодіяти між собою та обговорювати теми, що їх цікавлять, з іншими людьми, які мають подібні інтереси.

З поширенням соціальних мереж здається, що форуми втратили свою популярність, але це далеко не так. Форуми зберігають віру в традиційну комунікацію на тематичні та професійні теми. Для тих, хто шукає детальну інформацію про конкретну тему, форуми залишаються надійним джерелом знань та взаємодії з експертами в різних галузях.

Порівняльна характеристика форумів та соціальних мереж наведена у табл. 1.1.

Таблиця 1.1 – порівняльна характеристика форумів та соціальних мереж

Особливості	Форуми	Соціальні мережі
Спрямованість	Тематичні	Багатопрофільні
Функціонал	Обмін думками, досвідом, інформацією	Спілкування, пошук друзів, відео, фото, новини
Кількість користувачів	Зазвичай менше, ніж у соціальних мережах	Переважна більшість

Продовження таблиці 1.1

Особливості	Форуми	Соціальні мережі
Спосіб взаємодії	В основному текстовий	Різноманітні формати взаємодії
Рівень взаємодії	Менше, ніж у соціальних мережах	Більший, багатофункціональний

Враховуючи всі переваги та недоліки, можна зробити висновок, що форуми мають свої особливості та можуть бути більш ефективними для певних цільових аудиторій. Наприклад, для тих, хто шукає точну та детальну інформацію, форум може стати кращим варіантом, оскільки на ньому зазвичай зібрано велику кількість експертів, які готові допомогти та відповісти на запитання. Також, форуми дають можливість більш детально обговорювати певну тему та взаємодіяти з користувачами, які мають однакові інтереси.

1.2 Імплементація front-end частини форуму

1.2.1 Розгляд функціоналу форумів

Форуми є важливими елементами вебсайтів і застосунків, що дозволяють користувачам обговорювати різноманітні теми, ділитися інформацією та знаннями, розв’язувати проблеми та ставити питання. Функціонал форумів повинен бути зрозумілим і зручним для користувачів, а також дозволяти ефективно обмінюватися інформацією та взаємодіяти з іншими користувачами.

Фундаментальною функцією будь-якого форуму є можливість створювати теми та повідомлення. Це дозволяє користувачам обговорювати різні теми та висловлювати свої думки [3]. Щоб створення тем та повідомлень було максимально зручним для користувачів, fron-end розробник повинен забезпечити наявність простої та зрозумілої форми для

цих дій. Наприклад, на формі для створення теми повинні бути поля для введення назви теми, короткого опису, тексту повідомлення та можливість вибору категорії, до якої належить тема. Для повідомлень форма має містити поле для введення тексту повідомлення та можливість цитування попереднього повідомлення.

Важливо забезпечити можливість відповіді на повідомлення та їх цитування. Це дозволяє користувачам звернутися до конкретного повідомлення та зручно продовжувати обговорення. Наприклад, можна додати кнопку "Цитувати", щоб користувач міг вибрати текст, який потрібно процитувати та вставити його в своє повідомлення.

Форматування тексту. Іноді користувачам необхідно підкреслити важливі деталі в темі або коментарі. Таку проблему вирішують формами форматування, які бувають як примітивними (дозволяють підкреслювати, робити жирний або курсивний текст), так і більш просунуті, що можуть виділяти кольором, додавати поля коду, створювати списки, ховати текст під спойлери та вшивати в текст посилання. Форматування тексту також допомагає підвищити зрозумілість повідомлень та сприяє легшому сприйняттю інформації. Наприклад, використання заголовків і підзаголовків може допомогти організувати текст та підвищити його читабельність.

Як і створювання тем, основоположним функціоналом є пошук інформації. Користувач має мати можливість шукати теми та повідомлення за різними критеріями, такими як дата, автор, ключові слова та інші параметри. Для цього можна створити форму пошуку, де користувач може вводити параметри пошуку та вибирати фільтри [3].

У розробці механізму пошуку необхідно забезпечити швидкість та точність пошуку. Швидкість пошуку може бути забезпечена за допомогою використання ефективних алгоритмів пошуку та індексації даних на форумі. Точність пошуку можна покращити за допомогою використання різних технік, до прикладу «Розумний пошук», де пошукова система знає не тільки ключові слова, а й контекст, у якому вони використовуються, або «Корекція

помилок» - враховування можливих орфографічних помилок у запиті користувача. Можна згадати ще «Ранжування результатів». Ця техніка дозволяє системі відсортувати результати пошуку за певним критерієм. Наприклад, система може ранжувати результати за кількістю переглядів, кількістю відповідей, датою оновлення та іншими параметрами, щоб найбільш важливі результати були на першій сторінці.

Пошуку інформації сприяє гарна структуризація тем, що забезпечується ще й таким важливим аспектом, як категорії. Вони можуть бути створені залежно від тематики форуму, наприклад, на форумі про ігри можуть бути категорії для різних видів ігор, на форумі про книги - категорії для різних жанрів книг тощо. Крім того, на форумах можуть бути встановлені підкатегорії, які дозволяють більш детально структурувати теми та повідомлення в рамках кожної категорії. Наприклад, на форумі про ігри підкатегорії можуть бути створені для різних видів ігор, таких як стратегії, рольові ігри, спортивні ігри тощо.

Можна пригадати ще особливість форумів, завдяки якій тему можна наповнити шляхом прикріплення файлів до повідомлень. Це дозволяє користувачам ділитися додатковою інформацією, такою як зображення, відео чи документи, які зроблять обговорення більш інформативним та зрозумілим для інших користувачів.

Проте, якщо будь-яка людина, що зайде на сайт, зможе робити на ньому буквально усе що завгодно, то це все переросте у, так би мовити, хаос. Тож, форум треба обмежувати. Обмеження реалізовується введенням облікових записів і рольових систем. У залежності від налаштувань форуму, різні типи користувачів можуть мати різні рівні доступу та права [3]. Наприклад, гостям може бути дозволено лише переглядати форум без можливості створення повідомлень, тоді як зареєстрованим користувачам можна дозволити створювати нові теми та відповідати на повідомлення. Адміністратори ж мають повний доступ до всіх функцій форуму, включаючи управління користувачами, модерацію повідомлень, редагування тем тощо.

Таким чином, рольова система на форумах дозволяє краще контролювати діяльність користувачів, захищати форум від спаму та нецензурних висловлювань, а також забезпечує зручний інтерфейс для обговорення тем.

Але як зрозуміти, чи тема дійсно варта уваги? Звісно, можна повністю переглянути її опис та коментарі, проте є швидший спосіб – система оцінювання та рейтингу.

Система оцінювання та рейтингу - це інструмент, який дозволяє користувачам оцінювати повідомлення та коментарі на форумі. Це може бути корисним для визначення найбільш корисних інформаційних джерел та відсіювання непотрібних повідомлень.

Зазвичай система оцінювання має форму кнопок «плюс» та «мінус», що дозволяють користувачам голосувати за або проти повідомлення. Це дає змогу збирати статистику та зрозуміти, які повідомлення є найбільш корисними та цікавими для користувачів.

Деякі форуми також дозволяють користувачам встановлювати різні рівні вагомості голосів. Наприклад, досвідчені користувачі можуть мати більш вагомий голос, що дозволяє їм впливати на рейтинг повідомлень більше, ніж новачки.

Оцінки повідомлень можуть використовуватися для встановлення рейтингів та підвищення звідності користувачів, які надали корисну інформацію. Рейтингова система може надати іншим користувачам змогу зрозуміти, які користувачі є найбільш корисними та які повідомлення є найбільш авторитетними на форумі [4].

1.2.2 Розробка інтерфейсу користувача

Завдяки розвитку технологій та інтернету, сучасний світ став дуже насиченим інформацією. Кожен день ми стикаємось з великою кількістю даних, що потребують нашої уваги. У таких умовах важливо мати зручний

інтерфейс, який допоможе нам легко та швидко зорієнтуватися на сайті та знайти необхідну інформацію.

Зручність інтерфейсу означає, що користувачі можуть легко знаходити та використовувати всі функції та елементи форуму. Це може бути досягнуто шляхом забезпечення простоти та зрозумілості інтерфейсу, який дозволить користувачам швидко зорієнтуватись на сторінці та знайти потрібну інформацію.

Логічний інтерфейс означає, що всі елементи та функції повинні бути розміщені таким чином, щоб користувачі могли легко зрозуміти, як вони працюють та які вони роблять. Наприклад, якщо форум має багато категорій, то краще розмістити їх в окремому меню, щоб користувачі могли швидко знайти потрібну категорію, а не були змушені шукати її серед багатьох інших елементів [5].

Розташування елементів на сторінках повинно відповідати принципам дизайну та досвіду користувача (UX). Це каже нам про те, що елементи повинні розташовуватись таким чином, щоб користувачі могли легко знаходити потрібну інформацію та взаємодіяти з форумом без зайвих зусиль. Основними принципами розташування елементів на форумі є:

– ієрархія. Елементи на сторінці повинні бути розміщені відповідно до їх важливості та значущості. Найважливіші елементи, такі як меню чи кнопки для створення нових тем, повинні бути поміщені на більш видимих місцях. Менш важливі елементи, наприклад, інформація про користувача, можна розмістити на менш видимих місцях;

– рівномірність. Елементи повинні бути розміщені рівномірно на сторінці для забезпечення зручності користування. Наприклад, якщо важливі кнопки розташовані з одного боку, то інші елементи також повинні бути розміщені з іншого боку, щоб створити рівномірний баланс;

– простота. Форум повинен мати простий та лаконічний дизайн, який не заважатиме користувачам. Наприклад, відсутність зайвих елементів, таких

як анімації чи великих зображень, зробить сторінки простішими та зрозумілими;

– контрастність. Кольори та шрифти повинні бути контрастними, щоб забезпечити читабельність та зручність користування. Наприклад, якщо фон сторінки світлий, то тексти повинні бути написані темним кольором, щоб були легко читабельні;

– адаптивність. Форум повинен мати адаптивний дизайн для різних пристроїв та розмірів екранів. Наприклад, якщо користувач переглядає форум на мобільному телефоні, то елементи повинні бути зменшені [5].

Сучасні дизайнери та front-end розробники все більше користуються паттернізованними схемами розробки сайтів, приділяючи увагу розміщенню елементів інтерфейсу в областях підвищеної концентрації уваги. Пов’язано це з дослідженнями Якоба Нільсена, засновника компанії Nielsen Norman Group.

Він провів у 2006-му році дослідження з вивчення того, як користувачі переглядають сторінки сайтів. У своїх дослідженнях він виявив, що користувачі не читають сторінки по вертикалі, а замість цього швидко сканують їх, шукаючи ключові слова та фрази, що відповідають їхнім потребам. Новими дослідженнями 2016-го року було виведено три основні типи поведінки при скануванні сторінок сайтів, це F-патерн, Z-патерн та Діаграма Гутенберга [6]. Діаграми зображено на рис. 1.2.

F-патерн – це типова поведінка користувача, коли він сканує сторінку зверху вниз, зліва направо, утворюючи букву "F". Користувачі зазвичай спочатку переглядають верхній рядок заголовків, потім спускаються на другий рядок і шукають ключові слова або фрази, що їх цікавлять. Далі вони пересуваються вздовж лівого краю сторінки, шукаючи важливі інформаційні блоки, і закінчують перегляд сторінки, проскакуючи кілька рядків тексту вниз [7].

Z-патерн - це поведінка, яка схожа на F-патерн, але замість перегляду верхнього рядка заголовків, користувачі швидко сканують сторінку вздовж

діагоналі, утворюючи букву "Z". При цьому вони переглядають верхній лівий кут, потім переходять на праву сторону екрана, скануючи по діагоналі вниз, і закінчують на нижньому рядку.

Діаграма Гутенберга - це складніша поведінка, коли користувачі проскановують сторінку, пересуваючись з пункту на пункт, звертаючи увагу на ключові слова та фрази, які виділяються на фоні решти тексту [8].

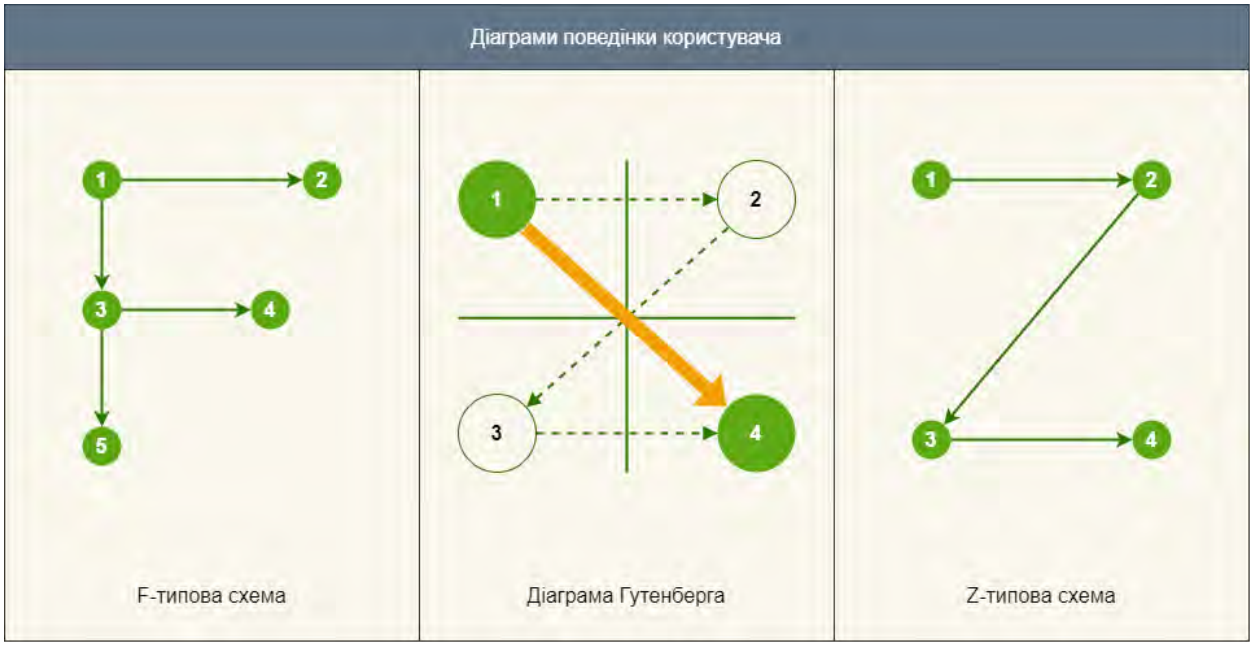

Рисунок 1.2 – Зображення трьох типів зчитування інформації користувачем

Дослідження компанії внесло значний вплив на процес розробки вебдизайну. Воно дозволило дизайнерам-верстальникам краще зрозуміти, як люди сприймають та взаємодіють зі сторінками сайтів. За результатами дослідження, дизайнери почали звертати більше уваги на розташування найважливішої інформації та елементів дизайну на верхній частині сторінки, забезпечуючи більш зручний та швидкий доступ до неї для користувачів [8]. Крім того, вони також стали використовувати більше візуальних елементів, що привертають увагу користувачів та допомагають їм швидше зорієнтуватись на сторінці. Дослідження Якоба Нільсена було кроком у напрямку забезпечення кращої взаємодії між користувачами та вебсайтами.

1.2.3 Оптимізація швидкості роботи

Оптимізація швидкості роботи є надзвичайно важливою для будь-якого вебсайту, в тому числі й для форумів. Не тільки швидкість завантаження сторінок, але й загальний час роботи сайту можуть значно вплинути на користувачів та їх задоволеність від взаємодії з сайтом. Тому оптимізація швидкості роботи має велике значення для успішної роботи форуму.

Зменшення часу завантаження сторінок - одна з основних задач, з якою стикаються front-end розробники. Наприклад, оптимізація зображень для швидкого завантаження - це один зі способів зменшення часу завантаження сторінок. Якщо розмір зображення занадто великий, сторінка буде завантажуватись довше, що може призвести до незадоволення користувачів. Серед популярних методів оптимізації роботи зображень можна виділити такі методи:

– обмеження формату. Front-end розробник може обмежити список форматів, які користувачі зможуть завантажуватися на сервер. До прикладу, залишити JPG, GIF та WebP;

– використання lazy-loading для зображень. Це означає, що зображення завантажується тільки тоді, коли воно з'являється на екрані користувача, що зменшує час завантаження сторінки [9];

– векторні зображення для іконок та логотипів. Вони мають значно менший розмір файлу та можуть бути масштабовані без втрати якості.

Мінімізація кількості запитів до сервера - ще одна важлива задача. Чим менше запитів відбувається до сервера, тим швидше сторінка завантажується. Один зі способів зменшення кількості запитів - це злиття (concatenation) CSS та JavaScript файлів, що дозволить зменшити кількість запитів до сервера. Крім того, варто розглянути можливість використання CDN (Content Delivery Network) – це дозволить користувачам отримувати

статичні файли (наприклад, зображення, JavaScript та CSS файли) з сервера, який є найближчим до них, що дозволить зменшити час завантаження.

Також існує такий спосіб зменшення запитів до сервера, як використання кешування. Кешування це один з найважливіших інструментів для оптимізації швидкості завантаження сторінок вебсайту. Цей процес полягає в збереженні копії вмісту сторінки на стороні користувача, щоб під час наступних запитів до сторінки не потрібно було завантажувати весь вміст знову з сервера. Воно застосовується для різних елементів вебсторінки, таких як зображення, стилі, скрипти та сторінки в цілому. Використання кешування дозволяє зменшити кількість запитів до сервера та збільшити швидкість завантаження сторінок [10].

Кешування може бути реалізоване як на стороні клієнта (браузера), так і на стороні сервера. На стороні клієнта можна використовувати локальне кешування браузера, яке дозволяє зберігати копії файлів на комп'ютері користувача. На стороні сервера можна використовувати його в оперативній пам'яті сервера або використовувати спеціальне програмне забезпечення для збереження кешованого вмісту.

Щоб забезпечити ефективне кешування, важливо дотримуватися кількох правил. Наприклад, кешувати треба тільки статичний вміст, який не змінюється з часом. Кешування не повинно використовуватися для динамічного вмісту, який змінюється відвідувачами сайту. Також важливо встановити правильний час життя кешу, щоб забезпечити його актуальність, але не витрачати зайвих ресурсів на збереження непотрібної інформації [10].

Залежно від того, які дані ми кешуємо, може знадобитися валідація даних перед використанням їх з кешу. Наприклад, якщо ми зберігаємо результати запитів до API в кеші, то ми повинні перевіряти, чи є ці дані дійсними та не застарілими, перед тим, як використовувати їх знову. Валідація вхідних даних на front-end зазвичай виконується за допомогою JavaScript. При відправці форми на сервер, JavaScript перевіряє правильність введених даних користувачем. Наприклад, для поля електронної пошти

перевіряється, чи введена адреса має правильний формат (наприклад, «username@example.com»). Якщо дані введено неправильно, користувач отримає повідомлення про помилку.

На back-end валідація даних проводиться на стороні сервера, що забезпечує вищий рівень безпеки. На сервері проводиться перевірка на коректність даних та перевірка на допустимість введених даних. Наприклад, при реєстрації користувача на сервері перевіряється, чи введений пароль відповідає вимогам безпеки (наприклад, мінімальна довжина, наявність різних символів тощо).

Крім перевірки коректності даних, валідація може бути корисною для захисту від атак, таких як SQL-ін'єкції та XSS (Cross-Site Scripting) атаки. Наприклад, з використанням валідації можна перевіряти вхідні дані для впливу на запит до бази даних. Це дозволить запобігти виконанню шкідливого коду та захистити базу даних від атак.

Крім того, важливо пам'ятати, що валідація - це не повний захист від атак, тому слід використовувати й інші заходи безпеки, наприклад, ескейпінг (escaping) даних при їхньому відображенні на сторінці та використання параметризованих запитів до бази даних [10].

1.3 Аналіз існуючих IT-форумів

1.3.1 Огляд Reddit Programming

Reddit - це вебсайт, який можна порівняти з форумом, де користувачі з усього світу можуть обговорювати різні теми. Reddit складається з безлічі підрозділів, які називаються «subreddits», і кожен підрозділ присвячений певній темі. Інтерфейс зображено на рис. 1.3.

Reddit Programming - це підфорум на платформі Reddit, присвячений програмуванню та розробці програмного забезпечення. У цьому підфорумі можна знайти обговорення технічних питань, новини зі світу програмування, поради щодо кар'єри в IT-індустрії, а також спільноту програмістів, які

діляться своїми досвідами та знаннями. Теми, які обговорюються на Reddit Programming, дуже різноманітні. Вони можуть стосуватися різних мов програмування, фреймворків, технологій та інструментів розробки. Тут можна знайти поради щодо розв’язання конкретних проблем, обговорення підходів до розробки програмного забезпечення, рецензії на книги та онлайн-курси з програмування, а також новини про зміни в технологіях та проєктах програмного забезпечення [11].

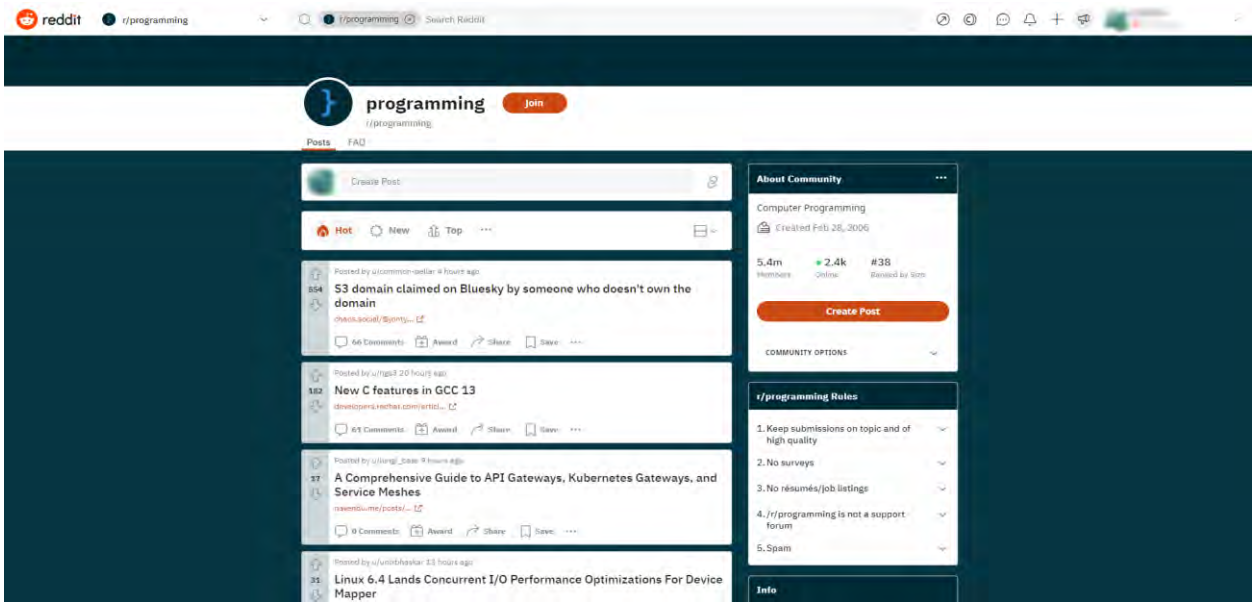

Рисунок 1.3 – Зображення стрічки останніх тем на Reddit Programming

Декілька плюсів Reddit Programming:

– дуже широка аудиторія. Reddit має понад 50 мільйонів унікальних користувачів щомісяця, тож це велика база потенційних читачів та підписників;

– значний обсяг інформації. На Reddit Programming можна знайти інформацію про будь-яку технологію та мову програмування, а також знайти поради та розв’язання проблем, що пов'язані з розробкою програмного забезпечення.

Деякі мінуси Reddit Programming:

– недостатньо модерації. Через велику кількість користувачів та контенту, деякі повідомлення можуть бути неякісними або не відповідати тематиці;

– негативна атмосфера. У деяких темах може бути досить агресивна або неприємна атмосфера, що здійснює негативний вплив на спільноту та спричинює неприємності користувачам;

– часті зміни алгоритму. Reddit регулярно змінює свій алгоритм та розміщення контенту, що може створювати проблеми для користувачів, які привчилися до певної системи.

1.3.2 Огляд GitHub Community Forum

GitHub Community Forum - це форум для спілкування, дискусій та обміну досвідом між користувачами GitHub. Форум присвячений різним темам, пов'язаним з GitHub, включаючи використання GitHub для розробки програмного забезпечення, відкритого коду, роботи з Git і багатьох інших тем [12]. Інтерфейс зображено на рис. 1.4.

Основні переваги GitHub Community Forum:

– велика спільнота користувачів, що забезпечує можливість швидкого отримання відповіді на питання, які можуть виникнути під час роботи з GitHub;

– наявність різних тематичних категорій та підкатегорій, що спрощує пошук необхідної інформації;

– можливість створення нових тем для обговорення, що дозволяє користувачам поширювати свій досвід та знання з іншими;

– можливість відстежувати нові повідомлення та теми, що вас цікавлять.

Основні недоліки GitHub Community Forum:

– наявність багатьох різних тематичних категорій та підкатегорій може зробити пошук необхідної інформації складним для новачків;

– залежність від активності користувачів форуму, що може призвести до зниження швидкості отримання відповіді на питання;

– можливість зустріти невірну інформацію в деяких повідомленнях, оскільки форум побудований на базі взаємодії користувачів між собою без підтримки від офіційних представників GitHub.

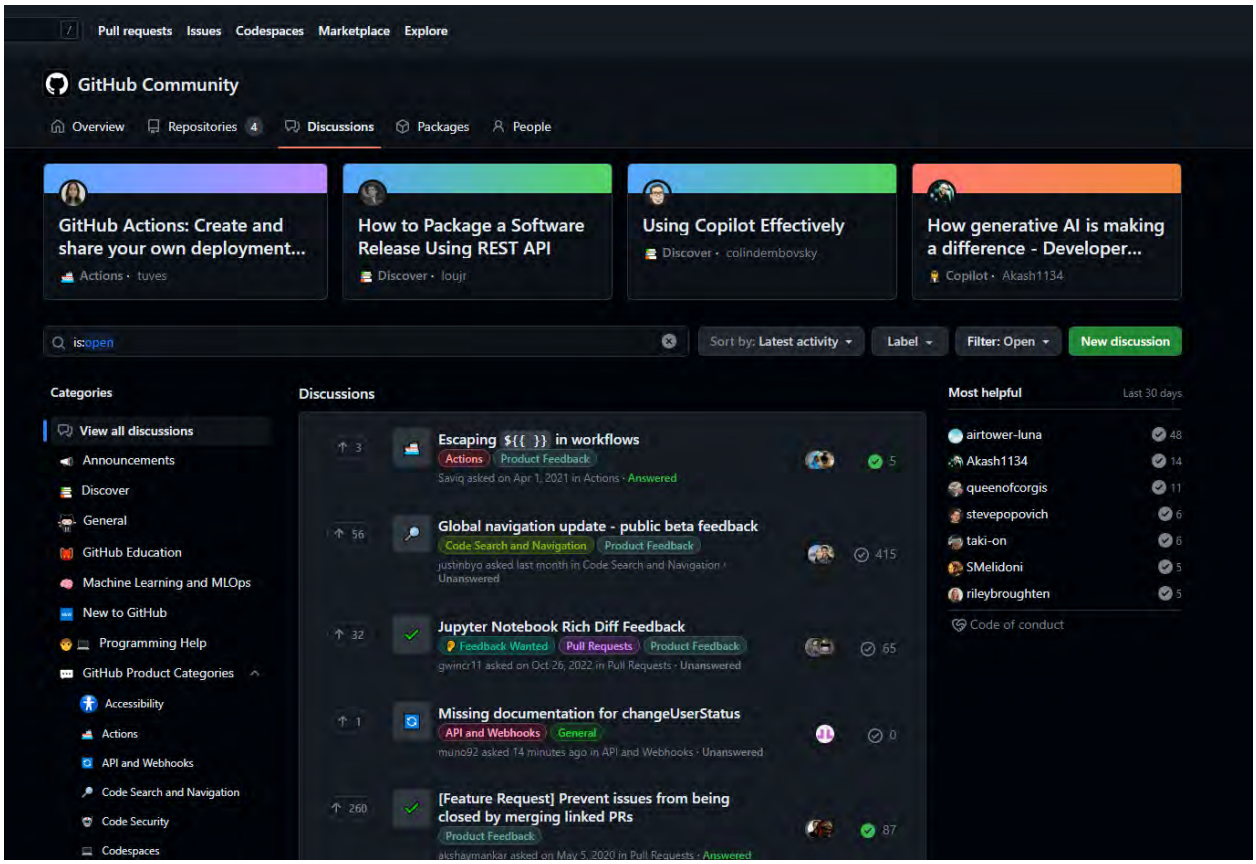

Рисунок 1.4 – Стрічка обговорень на GitHub Community

1.3.3 Огляд Stack Overflow

Stack Overflow - це один з найбільш відомих та популярних інтернет-форумів для програмістів. Сайт містить багато корисної інформації, що допомагає розв’язувати проблеми, які виникають при програмуванні [13]. Інтерфейс зображено на рис. 1.5.

Як і в багатьох інших форумів, основною функцією є можливість ставити запитання та отримувати відповіді на них від інших користувачів. Запитання можуть стосуватися будь-яких тем, пов'язаних з програмуванням, від базових до складних, таких як алгоритми та структури даних, розробка програмного забезпечення та технології. Користувачі також можуть голосувати за корисні відповіді, що дозволяє залучати до кращих відповідей, а також отримувати нагороди за внесок у розвиток форуму.

Одним з переваг Stack Overflow є високий рівень технічної експертизи його користувачів, що забезпечує якість відповідей та допомоги. Також сайт має зручний інтерфейс та потужну пошукову систему, що дозволяє знайти відповіді на запитання швидко та ефективно.

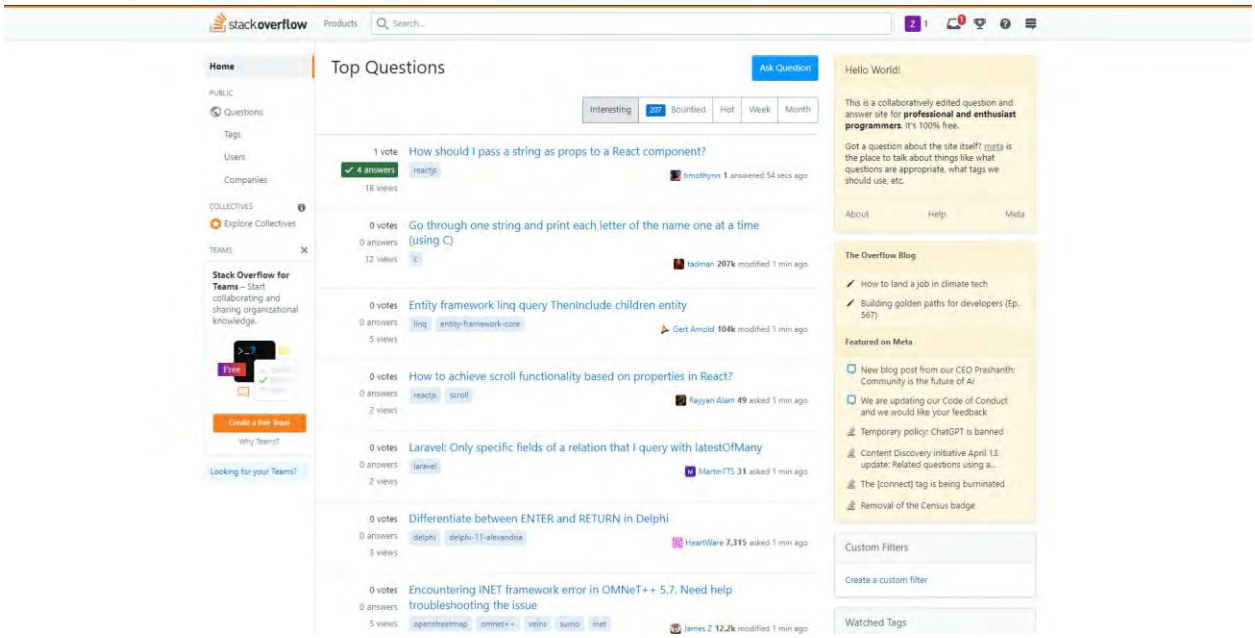

Рисунок 1.5 – Стрічка останніх постів на Stack Overflow

Плюси:

– велика й активна спільнота програмістів з усього світу, яка готова допомогти з різноманітними проблемами;

– вагато корисної інформації та відповідей на різні запитання;

– зручний пошук, що дозволяє швидко знайти відповіді на багато запитань;

– система репутації, що дозволяє програмістам заробляти бали за корисні відповіді та інші внески в спільноту;

– різноманітні категорії, що дозволяють шукати відповіді на запитання з різних технологій та мов програмування.

Мінуси:

– велика кількість запитань, відповідей та дописів може створювати складність у знаходженні потрібної інформації;

– можливість отримання неправильних відповідей, оскільки будь-який користувач може додавати відповіді без перевірки їх правильності;

– можливість отримання негативної відповіді від членів спільноти, що може бути заслужено або незаслужено;

– іноді користувачі можуть бути неввічливі та не дотримуватися правил спілкування;

– спільнота часто використовує англійську мову, що може створювати бар'єри для користувачів, які не розмовляють англійською.

1.4 Висновки за розділом

Під час аналізу предметної області було детально розглянуто технічну складову форумів. Було зазначено, що форуми побудовані на клієнт-серверній архітектурі, що дає змогу забезпечувати ефективний обмін даними між користувачем та сервером.

Окрім того, було проведено порівняння форумів з соціальними мережами, де було зазначено, що форуми забезпечують більш структурований та організований обмін інформацією. У порівнянні з соціальними мережами, форуми забезпечують більш ретельну модерацію контенту та підтримку правил поведінки на форумі, що знижує ризик небажаних повідомлень або спаму.

До переваг форумів можна віднести великий потенціал для взаємодії користувачів між собою, що дає можливість швидко та ефективно

розв’язувати проблеми та ділитися досвідом. Також форуми дають можливість створювати тематичні спільноти з різних галузей, що забезпечує високу якість та цінність обміну інформацією.

Однак, при розробці форуму необхідно звернути увагу на оптимізацію швидкості роботи, яка може суттєво впливати на задоволення користувачів від використання форуму. Також важливо розробляти зручний та інтуїтивно зрозумілий інтерфейс користувача, що сприяє покращенню взаємодії між користувачами та забезпечує більш зручне використання форуму.

Також було розглянуто імплементацію front-end частини форуму, де було зазначено, що розробка інтерфейсу користувача відіграє важливу роль в покращенні користувацького досвіду. Було висвітлено важливість оптимізації швидкості роботи форуму, що дозволяє зменшити час очікування користувача на завантаження сторінок форуму.

Загалом, під час аналізу предметної області було з'ясовано, що форуми забезпечують більш структурований та організований обмін інформацією порівняно з соціальними мережами. Важливу роль в покращенні користувацького досвіду відіграє розробка інтерфейсу користувача та оптимізація швидкості роботи форуму.

2 ВИБІР І ОБҐРУНТУВАННЯ СТРУКТУРИ СИСТЕМИ

2.1 Вибір мови розробки

2.1.1 Мова розмітки

Розробка клієнтської частини вебзастосунків є надзвичайно важливим етапом в процесі створення повноцінного вебсайту чи застосунку, яка передбачає вибір правильної мови програмування для реалізації проєкту, а також розробку інтерфейсу користувача, що забезпечує зручне та просте взаємодію з додатком. Вона має під собою інтерфейс користувача, який складається з HTML-коду, стилів, описаних за допомогою CSS, та логіки, написаної з використанням JavaScript.

HTML є стандартною мовою розмітки для вебсторінок, що використовується для створення інтерфейсу користувача форуму та містить в собі різноманітні елементи, такі як кнопки, поля введення тексту, списки, форми та інші. HTML-код складається з різних тегів, які визначають структуру та вміст інтернет-сторінки. Також підтримує багатомовність, що означає, що можна створювати сайти на різних мовах.

Є альтернативні мови розмітки, наприклад XML. Вона заснована на тому ж принципі, що і HTML, тобто використовує теги для визначення структури документа. Однак, XML більш гнучка і може використовуватися для створення будь-якого типу документа, а не тільки вебсторінок. Щодо використання його для розробки вебфорумів, то його використання є недоцільним. Однією з причин є те, що XML є менш простим і менш зрозумілим для більшості людей, ніж HTML. Також, XML дозволяє використовувати будь-які теги, що призводить до складнішої структури документа, що ускладнює розуміння та обробку даних.

Крім того, HTML має дуже широку підтримку в браузерах, тоді як підтримка XML є обмеженою. Це означає, що використання XML для

розробки вебфоруму може призвести до проблем з сумісністю та доступністю [14].

2.1.2 Мова стилів

Використання мови стилів є необхідним елементом в розробці сучасного форуму. Це дозволяє забезпечити консистентний та привабливий вигляд форуму для користувачів, а також забезпечує його гнучкість та підтримку різних пристроїв та екранів.

CSS використовується для опису вигляду вебсторінок, включаючи кольори, шрифти, фонові зображення, розміри елементів і т.д. CSS-стилі дозволяють розділити відображення вебсторінки від її структури та логіки, що забезпечує більшу гнучкість та підтримує різні пристрої та екрани [15].

Існує декілька альтернатив CSS, які можуть використовуватися для створення стилів вебсторінок, порівняльна характеристика яких наведена в табл. 2.1.

Таблиця 2.1 – Порівняльна характеристика мов стилів

Технологія	Опис	Переваги	Недоліки
CSS	Каскадні таблиці стилів - це мова опису, яка використовується для відображення веб-сторінок.	Простота використання, швидкість розробки.	Обмежена функціональність, важко підтримувати стилі на великих проектах, відсутність можливості використання змінних та функцій.
SASS	Розширена версія CSS, що дозволяє використовувати змінні, функції та інші корисні функції.	Підтримка змінних та функцій, можливість використання умов, інтерполяції та	Потребує встановлення компілятора, який перетворює SASS в CSS, що збільшує час розробки.

		вкладеності.	

Продовження таблиці 2.1

Технологія	Опис	Переваги	Недоліки
LESS	Розширений варіант CSS, який дозволяє використовувати змінні, функції та інші корисні функції.	Підтримка змінних та функцій, можливість використання умов, інтерполяції та вкладеності.	Обмежена функціональність порівняно з SASS, потребує встановлення компілятора.
Stylus	Це препроцесор CSS, що надає велику кількість корисних функцій, таких як змінні, функції та умови, без необхідності використовувати фігурні дужки та крапки з комами.	Простота синтаксису, підтримка змінних та функцій, можливість використання умов, інтерполяції та вкладеності.	Потребує встановлення компілятора, менша популярність порівняно з SASS та LESS.

Хоча ці альтернативи можуть мати свої переваги, використання CSS є стандартом для стилізації вебресурсу та вважається оптимальним варіантом для розробки вебдизайну.

CSS дозволяє відокремити відображення інтернет-сторінки від її структури та логіки, що полегшує роботу розробників та підвищує ефективність процесу розробки. Завдяки йому можна легко змінювати стиль та відображення елементів без впливу на функціональність сайту.

З використанням цієї мови можна забезпечити адаптивний дизайн IT-форуму, що дозволяє коректно відображати її на різних пристроях та екранах. Наприклад, можна використовувати медіазапити для налаштування відображення вебсторінки на мобільних пристроях, планшетах та десктопах.

Крім того, CSS дозволяє знизити завантаження вебсторінки та підвищити її продуктивність. Наприклад, можна використовувати кешування CSS-файлів для швидкого завантаження стилів на сторінці, а також оптимізувати розмір файлів CSS, що зменшить час завантаження вебсторінки та забезпечить її швидку роботу [15].

2.1.3 Мова додаткової логіки

JavaScript використовується для створення додаткової логіки та взаємодії з користувачем на вебфорумі. Він дозволяє створювати вебсторінки з більшою інтерактивністю та розширює можливості HTML та CSS. JavaScript може використовуватись для валідації введеного користувачем даних, зміни вмісту вебсторінки без перезавантаження сторінки та інших функцій.

JavaScript є однією з найпопулярніших мов програмування для розробки клієнтської частини вебзастосунків, проте, існують і альтернативи, які також можуть бути використані для розробки front-end складової вебсторінок. Порівняльна характеристика наведена в табл. 2.2.

Таблиця 2.2 – Порівняльна характеристика мов логіки вебсторінок

Функціональність/ технологія	JavaScript	CoffeeScript	Dart	Elm
Парадигма програмування	Мультипарадигмова (об'єктно-орієнтована, процедурна, функціональна)	Функціональна	Об'єктно-орієнтована	Функціональна
Компіляція до JavaScript	Ні	Так	Так	Так
Типізація	Слабка	Слабка	Сильна	Сильна
Статична типізація	Ні	Ні	Так	Так
Асинхронність	Callbacks, Promises,	Callbacks, Promises,	Futures, Streams,	Signals, Mailboxes,

	Async/await	Async/await	Async/await	Tasks
Функціональна програмація	Так	Так	Частково	Так

Продовження таблиці 2.2

Функціональність/ технологія	JavaScript	CoffeeScript	Dart	Elm
Оператори	Стандартні оператори	Можливість створювати свої оператори	Стандартні оператори	Стандартні оператори
Додаткові функції	Багато, враховуючи функції вищих порядків	Багато, враховуючи функції вищих порядків	Багато	Вбудована підтримка імутабельності даних

Попри деякі переваги, JS є однією з найбільш популярних та поширених мов програмування у світі. Вона є стандартом для розробки клієнтських та серверних застосунків вебтехнологій, що дає змогу знайти велику кількість ресурсів та бібліотек для її використання [15].

Обравши JavaScript, ми отримаємо доступ до великої кількості ресурсів, бібліотек та інструментів, а також до широкої спільноти розробників, що робить його більш практичним та зручним вибором.

Разом HTML, CSS та JS допомагають створити зручний та привабливий інтерфейс користувача, з легкою навігацією. Вони також забезпечують більшу гнучкість та підтримку для різних пристроїв та екранів. Однак, слід звернути увагу на оптимізацію швидкості завантаження вебсторінок, особливо при розробці форуму, який потребує багато інтерактивності та взаємодії з користувачами. Швидкість завантаження вебсторінок є ключовим фактором для забезпечення задоволення користувача та збільшення ефективності форуму. Для оптимізації швидкості завантаження вебсторінок важливо використовувати правильні техніки та оптимізовані зображення, уникати надмірного використання скриптів та зменшувати кількість запитів до сервера [15].

2.2 Вибір фреймворку для розробки клієнтської частини сайту

Фреймворки - це одна з найважливіших технологій для розробки програмного забезпечення, особливо у наш час, коли розмір та складність програмних проєктів зростає з кожним днем. Фреймворки надають структуровану архітектуру та платформу для розробки програмного забезпечення, що робить процес розробки більш ефективним та продуктивним.

Один з головних переваг їх використання полягає в тому, що вони надають готовий набір інструментів, бібліотек та методів, що зменшує кількість написаного коду та робить процес розробки простішим та швидким. Крім того, фреймворки забезпечують високий рівень безпеки та стабільності, що є особливо важливим для розробки програмного забезпечення, яке використовується у бізнес-процесах.

Зараз на ринку існує безліч різних фреймворків, які забезпечують розробникам потужні інструменти та технології для розробки програмного забезпечення. Порівняльна характеристика найвідоміших з них наведена у табл. 2.3.

Таблиця 2.3 – Порівняльна характеристика фреймворків

Фреймворк	Мова програмування	Тип	Ключові особливості	Ступінь складності	Популярність
Vue	JavaScript	Frontend	Декларативний синтаксис, компонентна архітектура	Середня	Висока
React	JavaScript	Frontend	Компонентний підхід, висока швидкість	Висока	Надзвичайно висока

			роботи, можливість побудови SPA		

Продовження таблиці 2.3

Фреймворк	Мова програмування	Тип	Ключові особливості	Ступінь складності	Популярність
Angular	TypeScript	Frontend	Компонентний підхід, вбудована обробка форм, двостороннє зв'язування даних	Висока	Висока
Laravel	PHP	Backend Frontend	Простота використання, вбудована авторизація та маршрутизація, підтримка відносин між таблицями	Низька	Висока

У зв'язку з тим, що серверна частина проєкту розроблялась з використанням фреймворка Laravel, було розглянуто також цей варіант. Зважаючи на його переваги порівняно з іншими фреймворками, він був обраний як найкращий для реалізації даного проєкту.

Серед найбільш корисних функцій фреймворку Laravel є Blade-шаблонізатор [16].

Blade є одним з ключових інструментів у Laravel, який допомагає розробникам створювати більш динамічні та ефективні вебсайти.

Наслідування та компоненти дозволяють розширювати та перевикористовувати код, що є корисною функцією для підтримки масштабування проєкту. Директиви дозволяють використовувати різноманітні функції, такі як умовні оператори та цикли, що забезпечує більш гнучкість у створенні шаблонів.

Blade має кілька ключових особливостей, які роблять його зручним та потужним шаблонізатором:

– синтаксис: Blade використовує простий та зрозумілий синтаксис для створення шаблонів. Наприклад, для вставки змінних використовуються подвійні фігурні дужки «{{ }}»;

– наслідування: Blade дозволяє використовувати наслідування шаблонів, що дозволяє розширювати та перевикористовувати код;

– компоненти: Blade дозволяє використовувати компоненти, які дозволяють створювати повторювані елементи, які можна використовувати в різних місцях на сайті;

– директиви: Blade дозволяє використовувати директиви, які забезпечують додаткові функції та можливості в шаблонах. Наприклад, директива @foreach дозволяє виконувати цикл над масивом;

– кешування: Blade підтримує кешування шаблонів, що дозволяє покращити продуктивність сайту та зменшити навантаження на сервер.

Кешування шаблонів є важливою функцією для покращення продуктивності вебсайту та зменшення навантаження на сервер, що робить Blade дуже корисним інструментом для веброзробників. В цілому, Laravel та Blade-шаблонізатор забезпечують простий та ефективний інструмент для створення вебсайтів [16].

Загалом, Blade-шаблонізатор є потужним інструментом, який дозволяє зручно та швидко організовувати сторінки сайту, забезпечуючи при цьому зрозумілий та лаконічний код. Це дозволяє розробникам зосередитися на логіці програми та бізнес-логіці, а не на деталях відображення.

2.3 Вибір додаткових інструментів

2.3.1 Набір інструментів Bootstrap

При розробці front-end частини вебзастосунків, розробники мають великий вибір додаткових інструментів, які допоможуть збільшити

продуктивність, поліпшити якість коду, та зробити застосунок більш користувацький та привабливим.

З таких інструментів варто виокремити Bootstrap. Це один з найбільш популярних інструментів для розробки клієнтської частини вебзастосунків, і знаходить широке застосування в індустрії веброзробки. Фреймворк розроблений на базі CSS, JavaScript та HTML, тому його використання не потребує глибоких знань програмування. Основною метою є спрощення процесу розробки вебсайтів та застосунків, забезпечення швидкості, ефективності та зручності. Він надає широкий вибір готових компонентів та стилів, які можна використовувати в розробці вебзастосунків, що дозволяє значно збільшити продуктивність розробника та зменшити час розробки.

Всі компоненти фреймворку створені відповідно до найкращих практик розробки вебсайтів, тому можна бути впевненим в їхній якості та сумісності з різними браузерами [17].

Основні переваги Bootstrap полягають у тому, що він:

– має багато готових компонентів та стилів;

– підтримує браузери, які використовуються найбільшою кількістю користувачів;

– має активну спільноту розробників, яка надає підтримку та допомогу.

Окрім того, цей набір інструментів має вбудовану систему сітки, яка дозволяє легко створювати респонсивні дизайни для будь-яких розмірів екранів. Сітка дозволяє розбити сторінку на рядки та колонки, що значно спрощує розміщення контенту на сторінці. За допомогою класів, які встановлюють ширину та відступи для кожної колонки, можна легко створювати складні дизайни, які добре працюють на різних пристроях та екранах.

Bootstrap став дуже популярним у світі веброзробки, тому що він дозволяє розробникам зосередитися на функціональності своїх застосунків, а не на візуальному оформленні [17].

2.3.2 Візуальний редактор тексту TinyMCE 6

Використання візуального редактора тексту на форумі дозволяє користувачам більш ефективно взаємодіяти один з одним та зосередитися на змісті своїх повідомлень, а не на їх вигляді. Завдяки використанню редактора можна легко форматувати текст, додавати зображення, таблиці, посилання та інші елементи. Це особливо корисно для форумів з технічною спрямованістю, де можна детально обговорювати питання та ділитися інформацією з іншими користувачами.

Один з найпопулярніших візуальних редакторів для форумів - це TinyMCE 6, який надає користувачам зручний інтерфейс для редагування та стилізації тексту без потреби знань мови розмітки HTML. Його основні функції містять вставку тексту з форматуванням, додавання мультимедійного контенту, створення таблиць та списків, редагування коду HTML, підтримку кросбраузерності та респонсивного дизайну [18].

Крім того, TinyMCE 6 має багато інших корисних функцій, таких як можливість роботи зі списками, вставки таблиць, редагування зображень та багато іншого. Він також має гнучку настройку, що дозволяє налаштувати редактор під свої потреби, включаючи визначення доступних опцій форматування, налаштування шрифтів та кольорів, включення чи виключення певних кнопок та інші настройки.

Завдяки використанню цього текстового редактора на форумах, користувачі можуть легко та швидко додавати зображення, відео та інші елементи до своїх повідомлень, що робить їх більш зрозумілими та цікавими для інших користувачів. Крім того, це дозволяє використовувати більш різноманітні формати тексту, такі як вирівнювання, зміна розміру шрифту та інші ефекти, що допомагають зробити повідомлення більш привабливими та зрозумілими.

Одна з головних переваг TinyMCE 6 полягає в тому, що він є повністю підвладним кастомізації, тобто можна налаштувати редактор з врахуванням потреб користувачів та вимог проєкту. Крім того, цей інструмент має велику спільноту розробників, що створює додаткові плагіни та розширення для його функціонала [18].

2.3.3 Менеджер пакетів прикладного рівня

В сучасному світі розробка програмного забезпечення стала дуже складною задачею, яка вимагає від розробників великих зусиль та уваги до деталей. Однією з найбільш важливих задач, яку необхідно вирішувати в розробці програмного забезпечення, є управління залежностями. Це дуже важлива задача, оскільки програмне забезпечення може містити десятки тисяч файлів та бібліотек, які потрібно враховувати та змінювати з часом.

Один з найкращих способів управління залежностями - використання менеджера пакетів прикладного рівня. Цей інструмент дозволяє розробникам легко управляти залежностями та версіями пакетів, не витрачаючи велику кількість часу та зусиль.

У світі PHP є кілька менеджерів пакетів прикладного рівня, які дозволяють легко встановлювати та оновлювати залежності для проектів PHP, а саме Composer, PEAR, Phing, Symfony Flex та CakePHP.

Composer - це найпопулярніший менеджер пакетів для PHP [19]. Він дозволяє легко встановлювати та оновлювати залежності з пакетів, розміщених на Packagist, який є репозиторієм для пакетів PHP. Автоматично встановлює залежності та дозволяє легко оновлювати їх до новіших версій. Він також дозволяє налаштувати autoload, щоб автоматично завантажувати класи пакетів при їх використанні [19].

PEAR - це старіший менеджер пакетів для PHP. PEAR має свій власний репозиторій пакетів та пропонує понад 5000 пакетів. Хоча він все ще використовується, він вважається менш популярним порівняно з Composer.

Phing - це менеджер залежностей, що використовується для збирання, тестування та розгортання PHP-проєктів.

Symfony Flex - це менеджер пакетів, розроблений для Symfony-фреймворку. Він дозволяє легко встановлювати та оновлювати залежності для Symfony-проєктів.

CakePHP - це менеджер залежностей для CakePHP-фреймворку. Він дозволяє легко встановлювати та оновлювати залежності для проєктів на CakePHP. Він також має свій власний репозиторій пакетів.

Їх порівняльна характеристика наведена в табл. 2.4.

Таблиця 2.4 – Порівняльна характеристика менеджерів пакетів

Функції / Менеджери пакетів	Composer	PEAR	Phing	Symfony Flex	CakePHP
Встановлення та видалення пакетів	+	+	+	+	+
Оновлення пакетів	+	+	-	+	+
Робота з залежностями	+	+	+	+	+
Автоматичне завантаження залежностей	+	-	-	+	+
Робота з локальним репозиторієм	+	-	-	+	+
Сумісність з PSR-стандартами	+	-	-	+	+
Наявність плагінів	-	+	+	+	+
Документація	+	+	+	+	+
Самостійність	+	-	+	-	-

З таблиці можна зробити висновки, що Composer є дуже гнучким та потужним менеджером залежностей прикладного рівня для PHP. На відміну

від інших менеджерів, він має значно більший репозиторій пакетів та широку спільноту, що дозволяє швидко знаходити та встановлювати необхідні залежності.

Також, Composer простий у використанні та дуже зручний для автоматичної інсталяції та оновлення залежностей. Його можна інтегрувати з будь-яким іншим інструментом для збірки та розгортання проєкту, що робить його універсальним інструментом для веброзробки.

Крім того, Composer має дуже широкі можливості для настройки, що дозволяє налаштувати його під конкретні потреби, такі як додавання власних репозиторіїв, налаштування версій пакетів та інші.

Таким чином, Composer є найкращим вибором для менеджменту залежностей прикладного рівня для PHP, завдяки своїм потужним можливостям, широкій спільноті та простоті використання [19].

2.3.4 Контроль версій

Контроль версій - це незмінний елемент будь-якої сучасної розробки програмного забезпечення. Використання інструментів контролю версій є необхідним для ефективної роботи в команді та зберігання історії змін проекту. У цьому есе я обговорю будь-які переваги, які мають інструменти контролю версій, та їх використання у розробці IT-форуму.

Однією з головних переваг використання інструментів контролю версій є можливість зберігати історію змін проєкту. Завдяки цьому можна повернутися до будь-якої попередньої версії, якщо потрібно. Це забезпечує збереження всіх попередніх змін та врахування досвіду попередніх версій. Крім того, використання інструментів контролю версій забезпечує зручний доступ до робочої копії проєкту для всіх членів команди, що забезпечує ефективну роботу та збільшує продуктивність.

Ще одною перевагою використання інструментів контролю версій є зручність роботи з вітками. Відколи проєкт стає складнішим, розробники

часто працюють над окремими функціями та підсистемами в окремих вітках. Завдяки віткам, розробники можуть працювати одночасно над різними функціями, не взаємодіючи між собою, та пізніше об'єднати всі зміни в основну вітку. Це забезпечує ефективну роботу та зменшує ризик відчуження членів команди. Існує багато різних інструментів для контролю версій, але найбільш популярними з них є Git, SVN та Mercurial.

Git - це найбільш популярна система контролю версій, що дозволяє працювати з розподіленими репозиторіями, має потужні гілки та злиття, а також забезпечує можливість працювати з віддаленими репозиторіями, зокрема з GitHub та Bitbucket [20].

SVN - це система контролю версій, яка працює з централізованими репозиторіями, тобто всі зміни зберігаються на сервері, а клієнти отримують та зберігають дані з сервера. SVN має меншу кількість гілок та злиття порівняно з Git, але дозволяє легко керувати доступом до файлів та змінювати їх історію.

Mercurial - це інша система контролю версій, що працює з розподіленими репозиторіями. Вона пропонує схожий набір функцій з Git, але має простіший та зрозуміліший інтерфейс для користувачів. Крім того, Mercurial є більш гнучкою системою в порівнянні з SVN, але менш популярною за Git. Порівняльна характеристика наведена в табл. 2.5.

Таблиця 2.5 – Порівняльна характеристика інструментів контролю версій

Функціональність	Git	SVN	Mercurial
Тип системи контролю версій	Розподілена	Централізована	Розподілена
Вітки	Потужна підтримка віток та злиття гілок	Підтримка віток, але злиття може бути складним	Потужна підтримка віток та злиття гілок
Швидкість	Швидка операція злиття та вилучення файлів, повільна	Повільне злиття, але швидка ініціалізація та клонування	Швидка операція злиття та вилучення файлів, повільна

	ініціалізація		ініціалізація

Продовження таблиці 2.5

Функціональність	Git	SVN	Mercurial
Співпраця	Підтримка розподіленої співпраці	Підтримка централізованої співпраці	Підтримка розподіленої співпраці
Слідування за файлами	Відстеження змін в кожному файлі	Тільки цілі файли можуть бути відстежені	Відстеження змін в кожному файлі
Керування розгалуженням	Потужна підтримка розгалуження та злиття гілок	Обмежена підтримка розгалуження та злиття гілок	Потужна підтримка розгалуження та злиття гілок
Конфлікти злиття	Вирішення конфліктів може бути складним	Легко вирішувати конфлікти злиття	Легко вирішувати конфлікти злиття

За порівнянням з SVN та Mercurial, Git має кілька переваг. Він є найбільш поширеним інструментом для контролю версій, має широку спільноту користувачів та забезпечує швидкий та ефективний процес роботи зі сховищем. Git має гнучкий та масштабований підхід до віток і злиття, що дозволяє розробникам працювати з проєктами різної складності. Він підтримує розподілену розробку, що забезпечує гнучкість та надійність у віддаленому співробітництві [20].

Git дозволяє зберігати кожен крок у процесі розробки, що дає можливість відновлювати будь-яку попередню версію проєкту за потребою. Крім того, він дозволяє кільком розробникам працювати над однією роботою одночасно, об'єднуючи їх зміни в один потік змін. Git також має безліч корисних функцій, включаючи керування гілками, відгалуження, коміти та об'єднання, що спрощують процес розробки та забезпечують оптимальну організацію проєкту [20].

Загалом, Git є потужним інструментом для контролю версій, який забезпечує швидкий та ефективний процес роботи з проектами різної складності, надійність у віддаленому співробітництві та гнучкість у

керуванні версіями. Його корисні функції дозволяють розробникам працювати з розробками більш організовано та продуктивно, що робить Git відмінним вибором для будь-якого розробника програмного забезпечення.

2.4 Схематична структура IT-форуму

Схематичне зображення – це один з найкращих способів проєктування сайтів, тому що воно дає можливість зрозуміти взаємні зв'язки між сторінками та формами сайту, а також дає образ всіх сторінок сайту в цілому. Це допоможе проєктувальникам зрозуміти, як розміщуються всі сторінки на сайті, та дозволить дослідити їх пов'язаність.

IT-форум складається з форм, материнських та дочірніх сторінок. Дочірні сторінки – це підсторінки, на які можливо потрапити лише через материнські сторінки. Деякі з них не доступні звичайному користувачу, задля збереження безпеки сайту, проте доступні адміністрації форуму.

На рис. 2.1 представлена схематична структура сайту, яка допомагає зрозуміти взаємодію між сторінками та формами, що були визначені для створення повноцінного сайту. Додатково, схематичне зображення дає змогу проаналізувати посилання між сторінками, щоб знайти найкращий спосіб навігації по сайту.

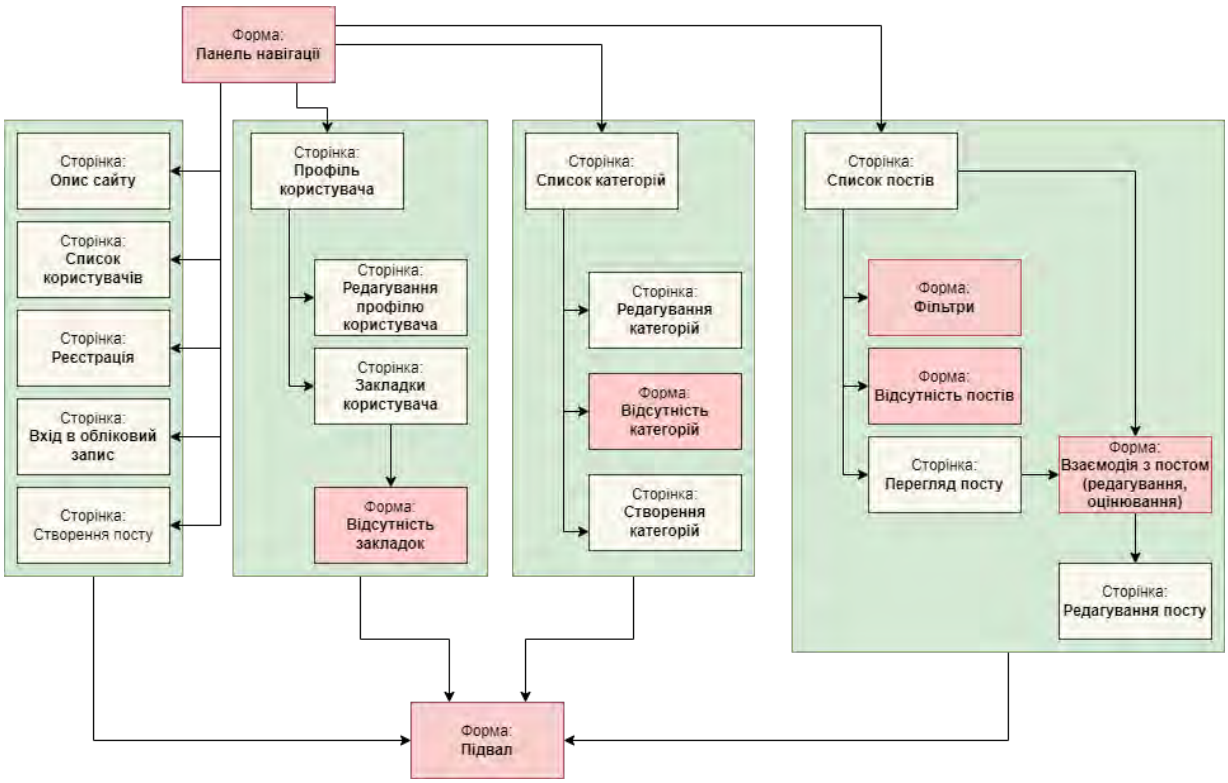

Рисунок 2.1 – Схематична структура сайту

Загальна навігація по IT-форуму здійснюється за допомогою панелі навігації, яка розташована на зафіксованій позиції, відображаючись по всьому сайту. Вона являє собою набір посилань, які дозволяють користувачам переміщатися між різними розділами сайту та доступати до різної інформації. Також, для більш зручного пошуку, на такій панелі розташований рядок пошуку. Він містить поле вводу, яке дозволяє користувачеві швидко та зручно знаходити потрібні дані у великій базі даних.

Сторінка списку постів. Вона є головною сторінкою будь-якого сайту. Ця сторінка містить перелік постів які були опубліковані на сайті. Зазвичай пости сортуються за датою публікації, що дозволяє користувачам знайти потрібний матеріал швидше. Для деяких сайтів сторінка списку постів містить мініатюри зображень, заголовки та короткі анотації тексту.

Сторінка опису сайту. Сторінка опису сайту є важливим елементом вебсайту, який потрібний для інформування відвідувачів про його мету, призначення та послуги.

Сторінка списку користувачів. Сторінка списку користувачів становить собою сторінку зі списком користувачів, які зареєстровані на сайті. Ця сторінка призначена для забезпечення прозорості сайту та можливості будь-кому переглядати активну діяльність інших користувачів. Зазвичай, ця сторінка містить імена користувачів, їх аватари, можливість переглянути їх профілі та іншу інформацію.

Сторінки входу та реєстрації. Сторінки входу та реєстрації є найпершим кроком для використання будь-якого вебсайту або програми. Вони містять посилання або поля, за допомогою яких користувач може авторизуватися на сайті чи програмі через введення свого імені користувача та пароля.

Сторінка створення посту (теми). Це інтерфейс, де користувачі можуть створювати нові теми для обговорення. На цій сторінці можна ввести назву нової теми, а також опис теми. Також можна обрати категорію для вашої теми та вказати теги, які будуть пов'язані з темою. Це допоможе іншим користувачам легше знайти пост.

Сторінка списку категорій. Ця сторінка являє собою візуальний список тем, які можуть бути обговорювані на форумі. Кожна категорія містить ряд підкатегорій, які додають додаткову структуру та деталізацію для кожної теми. Завдяки цьому, форум залишається порядним та зручним для участі в ньому. Завдяки цьому, учасники можуть знайти на форумі теми, які їх цікавлять, та мати можливість почати обговорення по них.

Сторінка профілю користувача. Вона надає користувачу можливість переглядати та змінювати свої персональні дані. Тут можна змінювати назву користувача, пароль та інші дані. Також ця сторінка часто використовується для відображення статистики користувача та його активності.

Як було зазначено на початку, сайт має материнські та дочірні сторінки. Такі сторінки, як «Редагування профілю» та «Редагування посту», вважаються підсторінками.

Редагування профілю користувача. Ця підсторінка дозволяє змінювати свої особисті дані, такі як ім'я, прізвище, адресу електронної пошти, пароль та інше. Доступ до неї є в кожного зареєстрованого користувача і вхід до неї відбувається через «Сторінку користувача».

Закладки користувача. В разі, якщо користувач захоче зберегти цікаві для нього теми, то переглянути їх зможе саме на цій сторінці. Доступ до закладок є лише в користувача і вхід відбувається як і в випадку з редагуванням профілю.

Створення категорій. Ця сторінка має під собою інтерфейс створення нових категорій чи підкатегорій. Перехід відбувається з материнської сторінки «Списку категорій».

Редагування категорій. Якщо категорія була створена з помилками, то розробники мають передбачити можливість для її редагування. Перехід відбувається з материнської сторінки «Списку категорій».

Перегляд посту. Це місце, де користувачі можуть переглянути пост і прокоментувати його. Тут відображатимуться всі повідомлення, які були залишені до даного посту, і буде доступний інтерфейс для додавання нових коментарів. Крім того, користувачі можуть переглядати інформацію про автора даного посту, а також переглядати статистику посту. Перехід відбувається зі сторінки «Списку постів».

На сайті також мають бути певні форми, які реалізували б певний функціонал та вивід інформації.

Форма фільтрів. Форма фільтрів для списку постів дозволяє користувачам показувати тільки ті пости, які відповідають заданим критеріям. За допомогою такої форми можна відфільтрувати пости за датою або за рейтингом.

Форма про відсутність закладок. Особлива форма для сторінки із закладками, що свідчить про їх відсутність через певні причини.

Форма відсутності категорій. Використовується для зазначення пустої сторінки, де мав би бути виведений список.

Форма відсутності постів. Якщо форум тільки було створено і користувачі ще не встигли створити жодної теми, існує ця форма, яка б про це повідомила.

Форма взаємодії з постом. Ця форма використовується як адміністрацією, так і звичайними користувачами, наприклад, для оцінювання теми, її редагування, видалення та додавання до закладок.

Підвал. Це остання частина сайту, яка розташовується знизу сторінки. Зазвичай він містить інформацію про авторські права, політику конфіденційності, посилання на соціальні мережі та інші контактні дані. Доступ до нього є з усіх сторінок форуму.

З аналізу схематичної структури IT-форуму можна зробити висновок, що IT-форум складається з певних сторінок і форм, які дозволяють користувачам здійснювати багато різних дій, що допоможуть їм знайти потрібну інформацію і залишати корисні коментарі. Завдяки продуманій структурі форуму користувачі можуть зручно знаходити ту інформацію, яку вони шукають, легко орієнтуватися на сайті та з комфортом обмінюватися думками з іншими учасниками.

2.5 Висновки за розділом

У цьому розділі було розглянуто та визначено основні технології розробки клієнтської частини IT-форуму.

Основними мовами верстки обрано HTML, CSS та JavaScript, які виконують різноманітні функції у розробці інтерфейсу користувача.

В якості фреймворку визначено Laravel, який надає гнучкість та швидкість розробки, а також забезпечує високий рівень безпеки. Також він надає шаблонізатор Blade для роботи з представленням даних в застосунку, який дозволяє легко та швидко створювати шаблони з використанням PHP-коду та вбудовуваних команд.

Для вебдизайну та розробки інтерфейсу користувача буде використовуватися Bootstrap 4 - потужний та зручний набір інструментів для створення адаптивних та красивих вебсторінок з використанням CSS та JavaScript.

В якості візуального редактора обрано TinyMCE, що дозволить користувачам легко редагувати та форматувати текст.

Для контролю версій та управління кодом використовуватиметься Git, який забезпечує збереження копій коду на різних етапах розробки та дозволяє працювати в команді над проєктом. Менеджер пакетів Composer буде використовуватися для управління залежностями розробки та автоматичного встановлення потрібних бібліотек.

В завершенні, було визначено і розглянуто складові схеми структури IT-форуму. В завершенні, було визначено і розглянуто складові схеми структури IT-форуму.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Загальні відомості про розробку

Front-end частина вебзастосунку була розроблена за допомогою таких технологій, як HTML, CSS та JavaScript, з використанням CSS-фреймворка Bootstrap, Blade-шаблонізатора та Composer (для керування залежностями) і контролю версій Git. Файлова система проєкту зображена у вигляді табл. 3.1.

Таблиця 3.1 – Файлова система проєкту

№	Тип	Розділ	Назва
1	Шаблони	auth	login.blade.php
			outBackground.blade.php
			register.blade.php
		categories	controlPanel.blade.php
			create.blade.php
			edit.blade.php
1	Шаблони		index.blade.php
			notFoundCategories.blade.php
		description	descriptionPage.blade.php
		layouts	app.blade.php
			error.blade.php
			footer.blade.php
			headerNavigation.blade.php
			layout.blade.php
		posts	create.blade.php
			edit.blade.php
			notFoundPosts.blade.php
			show.blade.php
			showBookmarkButtons.blade.php
			showCommentatorsList.blade.php
			showLikeButtons.blade.php
		users	edit.blade.php
			index.blade.php
			notFoundBookmarks.blade.php
			show.blade.php
			userBookmarks.blade.php
			userComments.blade.php
		users	userPosts.blade.php

		views	index.blade.php

Продовження таблиці 3.1

№	Тип	Розділ	Назва
2	Стилі	CSS	categoryStyles.css
			descriptionStyles.css
			filter.css
			footerStyles.css
			logRegStyles.css
			mainStyles.css
			navBar.css
			postEdit.css
			postsList.css
			postStyles.css
			tinymceStyles.css
			userList.css
			userLogo.css
			userProfileStyles.css
3	Логіка	JS	logicAuth.js
			logicCategories.js
			logicDescription.js
			logicPost.js
			logicPostsPage.js
			logicUserEditPage.js
			tinymceCore.js

Розглянемо призначення кожного з розроблених шаблонів.

Розділ «auth» містить в собі шаблони, що потрібні для реалізації авторизації та реєстрації користувача:

– «login.blade.php» – шаблон, який містить рядки для входу в акаунт;

– «register.blade.php» – шаблон, що має в собі інтерфейс для реєстрації нового користувацького акаунту;

– «outBackground.blade.php» – шаблон, який використовується в «login.blade.php» і «register.blade.php», та містить в собі зображення і логотип сайту.

Розділ «categories» містить в собі шаблони, що потрібні для відображення категорій та взаємодії з ними:

– «controlPanel.blade.php» – виведення окремої категорії чи під категорії в списці, для подальшої взаємодії з ними (редагування чи видалення);

– «create.blade.php» – сторінка створення нової категорії;

– «edit.blade.php» – сторінка редагування існуючої категорії;

– «index.blade.php» – структурована сторінка, на якій відображаються усі категорії та під категорії;

– «notFoundCategories.blade.php» – шаблон, завдяки якому, при відсутності категорій, сторінка буде містити повідомлення про це.

Розділ «description» містить в собі шаблон «descriptionPage.blade.php», що реалізовує сторінку опису сайту.

Розділ «layouts» містить головні шаблони сайту:

– «app.blade.php» – шаблон навігаційної панелі, включаючи посилання на вхід, реєстрацію та вихід із системи;

– «error.blade.php» – шаблон, що дозволяє відображати повідомлення про помилки;

– «footer.blade.php» – шаблон, що реалізовує підвал сайту;

– «headerNavigation.blade.php» – панель навігації по сайту, що містить кнопки для переходу на сторінку опису сайту, головну сторінку, сторінку зі списком користувачів, сторінку категорій, сторінку створення теми, а також рядок пошуку та випадаючий список профілю користувача;

– «layout.blade.php» – головний шаблон сайту, в якому підключаються усі глобальні стилі, інструменти та виводиться контент сайту.

Розділ «posts» містить в собі шаблони зображення та взаємодії з темами:

– «create.blade.php» – шаблон створення нової публікації з можливістю встановлення заголовку, категорії та написанням вмісту;

– «edit.blade.php» – шаблон редагування існуючої публікації;

– «notFoundPosts.blade.php» – шаблон, що вказує на відсутність публікацій;

– «show.blade.php» – шаблон, що зображує тему, містить елементи взаємодії з нею та коментарі;

– «showBookmarkButtons.blade.php» – шаблон взаємодії з постом, завдяки якому можна додавати публікацію в закладки, та прибрати його з них;

– «showLikeButtons.blade.php» – шаблон взаємодії з постом, що дозволяє оцінювати його.

В розділі «users» створено шаблони, що візуалізують певні функції та сторінки, пов’язані з користувачами:

– «edit.blade.php» – сторінка редагування інформації про користувача;

– «index.blade.php» – сторінка списку існуючих користувачів, де в кожного користувача зображено його фотографію, ім’я, псевдонім та кількість створених постів і коментарів;

– «notFoundBookmarks.blade.php» – шаблон, що вказує на відсутність закладок в користувача;

– «show.blade.php» – сторінка користувача, на якій розміщена інформація про нього, його фотокартка та активність на сайті;

– «userBookmarks.blade.php» – шаблон, завдяки якому користувач може переглянути те, які пости він додав у закладки;

– «userComments.blade.php» – шаблон, що зображує активність користувача, а саме список коментарів написаних на форумі;

– «userPosts.blade.php» – шаблон, що відображає усі створені теми.

Розділ «views» містить в собі шаблон «index.blade.php», який є головною сторінкою на сайті і містить список існуючих постів, що можуть бути відсортованими за допомогою фільтрів.

Розглянемо призначення файлів зі стилями:

– «categoryStyles.css» – стилі, що використовуються сторінками, пов’язаними з категоріями;

– «descriptionStyles.css» – стилі, що використовуються сторінкою опису сайту;

– «filter.css» – стилі, що реалізовують фільтри, на сторінці зі списком постів, у вигляді випадаючого списку;

– «footerStyles.css» – стилі, що використовуються для реалізації підвалу сайту;

– «logRegStyles.css» – стилі, що використовуються на сторінках авторизації та реєстрації;

– «mainStyles.css» – головний файл стилів, класи якого використовуються на форумі для реалізації повторюваних елементів;

– «navBar.css» – стилі, що використовуються для реалізації панелі навігації у шапці сайту;

– «postEdit.css» – стилі, що використовуються сторінкою редагування тем;

– «postsList.css» – стилі, що зображують пости на головній сторінці сайту;

– «postStyles.css» – стилі, що використовуються сторінкою постів, та застосовуються для блоків тем, категорій та коментарів;

– «tinymceStyles.css» – файл стилів для кастомізації інтерфейсу редактору тексту TinyMCE;

– «userList.css» – файл стилів, що використовується сторінкою списку зареєстрованих користувачів;

– «userLogo.css» – файл стилів, що використовується для зображення фотокартки користувачів у блоках постів та в списках користувачів;

– «userProfileStyles.css» – файл стилів, що використовується для елементів сторінки профілю користувача.

3.2 Розроблена логіка сайту

3.3.1 Скрипт logicAuth.js

Скрипт встановлює слухача подій на зміну розміру вікна браузера. При зміні розміру вікна, функція updateAuthStyles перевіряє, чи відповідає розмір правого блоку визначеному значенню. Якщо розмір не відповідає, то фонове зображення приховується, меню та правий блок отримують нові CSS-класи, а правий блок змінює свої властивості padding.

```
window.addEventListener('resize', updateAuthStyles);

const imgWith = 1040, mintWidthRightMenu = 200;

// Функція автоприховання фонового зображення при зміні розмірів вікна
function updateAuthStyles() {
   let rightBlock = document.getElementById("rightBlock");
   if(window.innerWidth - imgWith < mintWidthRightMenu) {
      document.getElementById("authImg").style.display = "none";
      document.getElementById("rightMenu").className = "";
      rightBlock.className = "row justify-content-center";
      rightBlock.style.padding = "10%";
      rightBlock.style.paddingTop = "30%";
   } else {
      document.getElementById("authImg").style.display = "block";
      document.getElementById("rightMenu").className = "input-block";
      rightBlock.className = "row";
      rightBlock.style.padding = "0";
   }
}
```

3.3.2 Скрипт logicCategories.js

Скрипт автоматично встановлює розмір тексту для основних категорій в залежності від довжини тексту. Він шукає всі посилання з класом "main-category-text" (назва головної категорії) і встановлює їм розмір шрифту, який залежить від довжини тексту. Якщо довжина тексту менше заданого мінімуму, то використовується максимальний розмір шрифту, інакше розмір шрифту зменшується залежно від коефіцієнту.

```
window.onload = function() {
   setFontsizeForMainCategory();
```

```
};

// Функція автоматично встановлює розмір тексту основних категорії в залежності від довжини
тексту
function setFontsizeForMainCategory(){
   const links = document.querySelectorAll('a.main-category-text');
   let fSize, maxSize = 240, koefSmallText = 0.6, minSymbols = 9;
   links.forEach(link => {
      if(link.textContent.length < minSymbols){
         fSize = maxSize / link.textContent.length;
      } else {
         fSize = maxSize / (koefSmallText * link.textContent.length);
      }
      link.style.fontSize = '' + String(fSize) + 'px'; });}
```

3.3.3 Скрипт logicDescription.js

Цей скрипт автоматично змінює клас елементів сторінки, що мають ідентифікатор "mainContainer" та "footerRow", що вказує на заміну класу "container" на "container-fluid" та "row" на "centered-content", відповідно. Таким чином, використання розширеного аналогу "container-fluid" дозволяє займати більшу площу екрана та створювати більш респонсивний дизайн.

```
// Функція автоматично замінює Bootstrap-контейнер на його розширений аналог
window.onload = function () {
   var obj = document.getElementById('mainContainer');
   obj.classList.remove("container");
   obj.classList.add("container-fluid");
   var objFixFooter = document.getElementById('footerRow');
   objFixFooter.classList.remove("row");
   objFixFooter.classList.add("centered-content");
}
```

3.3.4 Скрипт logicPost.js

Даний файловий скрипт містить в собі ряд функцій для роботи зі сторінками тем. На таких сторінках знаходяться різні блоки, які відповідають за відображення постів, коментарів та кнопок взаємодій.

Перша функція прибирає весь html код та вбудовані CSS-стилі з тексту, якщо вони там є.

```
// Функція встановлення html-елементу тексту коментаря без html коду
function cutHTML() {
   document.getElementById('TextWithoutHTML').value =
```

```
tinyMCE.activeEditor.getBody().textContent;
};
```

На форумі є можливість створювати відповіді на коментарі при натисненні на відповідну кнопку, що зображена на рис. 3.1.

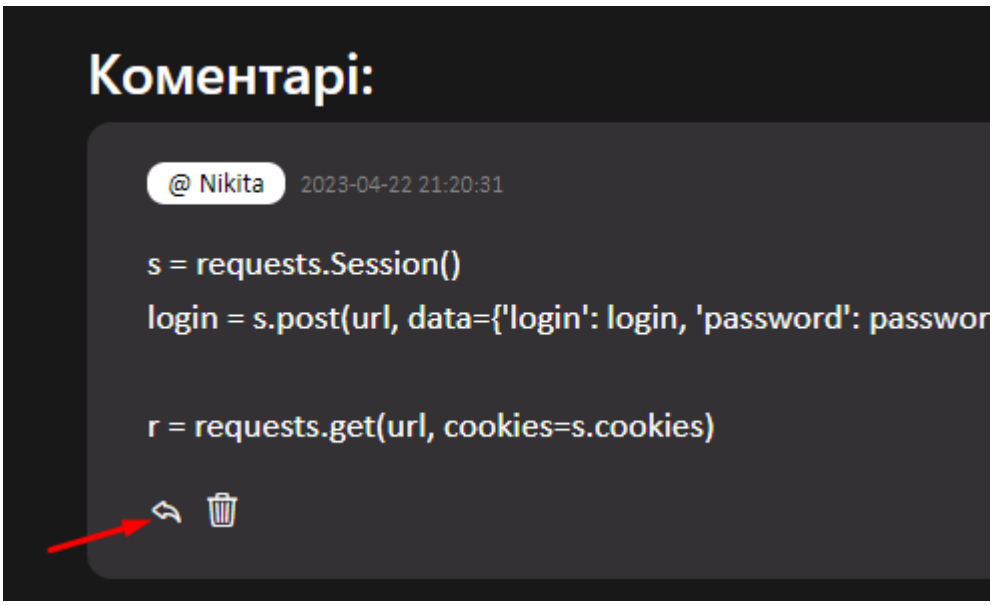

Рисунок 3.1 – Зображення кнопки відповіді

При натисненні на неї створюється спеціальний блок, який допомагає зорієнтуватися людині, на яке саме повідомлення вона відповідає. В разі, якщо кнопка була натиснута випадково і відповідь потрібно прибрати, користувач може скористуватися кнопкою відміни (рисунок 3.2), яка викликає функцію «deleteReplyOnComment()».

```
// Функція видалення html-блоку відповіді на коментар
function deleteReplyOnComment(){
   document.getElementById('replyMessageDiv').remove();
}
```

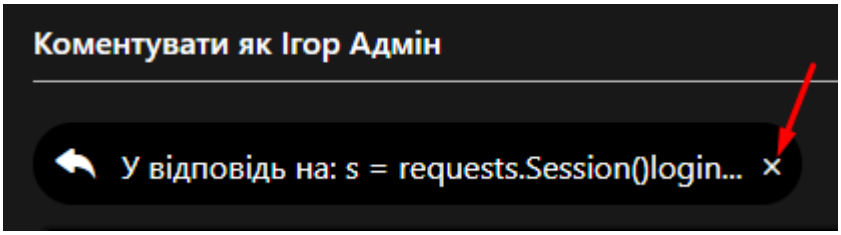

Рисунок 3.2 – Зображення кнопки відміни

Створюється форма «У відповідь на: …» функцією «setReplyOnComment()».

```
// Функція створення html-блоку відповіді на коментар
function setReplyOnComment(intro, id){
   if(document.getElementById('replyMessageDiv') !== null) {
      deleteReplyOnComment();
   }

   var obj = document.createElement('div');
   obj.id = "replyMessageDiv";
   obj.classList.add('reply-line');

   obj.innerHTML = '<i class="bi bi-reply-fill reply-icon"></i>' +
               '<div class="reply-text">У відповідь на: ' + intro + '</div>' +
               '<button type="button" class="cancel-button" aria-label="Close"
onClick="deleteReplyOnComment()">' +
               '<i class="bi bi-x"></i></button><input type="hidden" name="reply_message_id" value=' +
id + '>' +
               '<input type="hidden" name="reply_message_text" value="' + intro + '">';
   document.getElementById('replyMessagePoint').after(obj);

   smoothScroll('replyMessagePoint');
};
```

Функція «editComment()» встановлює текст коментаря в графічному редакторі тексту, виклик функції модифікації кнопки, змінює тип запиту на сервер з post на patch, та скролить до редактора.

```
// Функція редагування коментаря
function editComment(intro, id, htmlText, reply_message_id, reply_message_text, post_id) {
   tinymce.activeEditor.setContent(htmlText);
   if (reply_message_text != '') {
      setReplyOnComment(reply_message_text, reply_message_id);
   }
   SendToSaveButtons(id, post_id);

   let input = document.createElement("input");
   input.type="hidden";
   input.name="_method";
   input.value="patch";
   document.getElementById('comment-form').append(input);

   smoothScroll('replyMessagePoint');
}
```

У «SendToSaveButtons()» відбувається модифікація кнопки «Відправки». При редагуванні коментаря вона становиться кнопкою «Збереження».

```
// Функція заміни кнопки "Відправити" кнопкою "Зберегти"
function SendToSaveButtons(comment_id, post_id) {
   var SendButton = document.getElementById('send-button');
   SendButton.innerHTML = "Зберегти";
   SendButton.onclick = function () {
      saveEditedComment(comment_id,post_id);
   }
}
```

Зберігається коментар при натисненні кнопки «Зберегти», яка викликає функцію «saveEditedComment()».

```
// Функція збереження відредагованного коментаря
function saveEditedComment(comment_id, post_id) {
   cutHTML();
   document.getElementById('comment-form').action = '/posts/' + post_id.toString()+ '/comment/' +
comment_id.toString();
}
```

«currentYPosition()», допоміжний метод обчислення поточного положення фокусу на сторінці.

```
// Визначає поточне положення скролу
function currentYPosition() {
   // Firefox, Chrome, Opera, Safari
   if (self.pageYOffset) return self.pageYOffset;
   // Internet Explorer 6 - standards mode
   if (document.documentElement && document.documentElement.scrollTop)
      return document.documentElement.scrollTop;
   // Internet Explorer 6, 7 and 8
   if (document.body.scrollTop) return document.body.scrollTop;
   return 0;
}
```

«elmYPosition()» – функція, яка математично розраховує, де лежить певний елемент, роблячи це через його ідентифікатор.

```
// Визначає положення елемента
function elmYPosition(eID) {
   var elm = document.getElementById(eID);
   var y = elm.offsetTop;
   var node = elm;
```

```
    while (node.offsetParent && node.offsetParent != document.body) {
        node = node.offsetParent;
        y += node.offsetTop;
    } return y;
}
```

Метод, який реалізовує фізику плавного скролу, використовуючи попередні дві функції «elmYPosition()» та «currentYPosition()» – «smoothScroll()».

```
// Функція скролу до якоря
function smoothScroll(eID) {
    var startY = currentYPosition();
    var stopY = elmYPosition(eID);
    var distance = stopY > startY ? stopY - startY : startY - stopY;
    if (distance < 100) {
        scrollTo(0, stopY); return;
    }
    //var speed = Math.round(distance / 100);
    //if (speed >= 20) speed = 20;
    speed = 10;
    var step = Math.round(distance / 25);
    var leapY = stopY > startY ? startY + step : startY - step;
    var timer = 0;
    if (stopY > startY) {
        for ( var i=startY; i<stopY; i+=step ) {
            setTimeout("window.scrollTo(0, "+leapY+")", timer * speed);
            leapY += step; if (leapY > stopY) leapY = stopY; timer++;
        } return;
    }
    for ( var i=startY; i>stopY; i-=step ) {
        setTimeout("window.scrollTo(0, "+leapY+")", timer * speed);
        leapY -= step; if (leapY < stopY) leapY = stopY; timer++;
    }
}
```

«setCategories()» – функція яка імпортує дані про категорії в JS-скрипт.

```
// Функція встановлення категорій
function setCategories(name, id) {
    categoriesName.push(name);
    categoriesId.push(id);
}
```

Додавання категорій до теми відбувається при натисненні кнопки «+ Категорія», що реалізовано за допомогою «createCategoriesInput()».

```
// Функція створення полей вводу категорій
function createCategoriesInput(){
    let containerId = "categ_" + String(categoriesInputCounter);
    let selectElement = document.createElement('select');
    selectElement.name = "category_" + String(categoriesInputCounter);
    selectElement.id = "category_" + String(categoriesInputCounter);
    selectElement.classList = "category-dropdown";

    for (let i = 0; i < categoriesName.length; i++) {
        const optionElement = document.createElement("option");
        optionElement.value = categoriesId[i];
        optionElement.text = categoriesName[i];
        selectElement.appendChild(optionElement);
    }

    const buttonElement = document.createElement("button");
    buttonElement.innerHTML = '<i class="bi bi-x"></i>';
    buttonElement.classList = "category-delete-button";
    buttonElement.addEventListener("click", (event) => {
        event.preventDefault();
        document.getElementById(containerId).remove();
    });

    let containerElement = document.createElement("div");
    containerElement.id = containerId;
    containerElement.appendChild(selectElement);
    containerElement.appendChild(buttonElement);
    containerElement.classList = "d-inline category-box";

    document.getElementById('insertInputPlace').after(containerElement);


    categoriesInputCounter = categoriesInputCounter + 1;
}
```

«setOldCategoriesInput()». При редагуванні існуючої теми відображаються вже встановлені категорії (рис. 3.3)

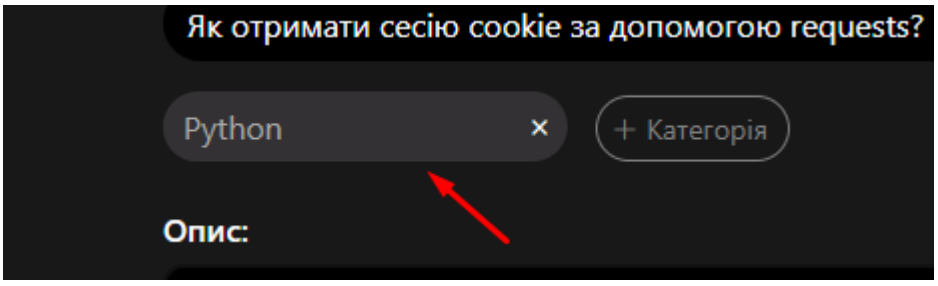

Рисунок 3.3 – Зображення виведених категорій

```
// Функція автоматично додає категорії, якщо такі є
function setOldCategoriesInput(id){
    createCategoriesInput();

    let selectId = "category_" + String(categoriesInputCounter-1);

    let selectElement = document.getElementById(selectId);
```

```
    let options = selectElement.options;
    for (let i = 0; i < options.length; i++) {
        if (options[i].value === id) {
            options[i].selected = true;
        }
    }
}
```

3.3.5 Скрипт logicPostPage.js

Цей скриптовий файл містить набір функцій, пов’язаних з фільтрацією постів.

Перша функція модифікує адресний рядок, вбудовуючи в нього параметри фільтрів.

```
// Функція вбудови до посилання критерію сортування
function setSortingFilter(e, param, value) {
    e.href = setUrlParam(param, value);
};
```

Друга функція генерує саме посилання.

```
// Функція вбудови параметрів сортування до посилання
function setUrlParam(param, value) {
    let newURL = window.location.protocol + '//' + window.location.host + window.location.pathname +
'?';

    let url = decodeURIComponent(window.location.search.substring(1)),
        urlVar = url.split('&'),
        flag = false,
        params;

    for (let i = 0; i < urlVar.length; i++) {
        params = urlVar[i].split('=');

        if (params[0] === param) {
            params[1] = value;
            flag = true;
        }

        if (params[0]) {
            newURL = newURL + params[0] + '=' + params[1] + '&'
        }
    }

    if (flag == false) {
        newURL = newURL + param + '=' + value
    }

    return newURL
};
```

Третя функція встановлює фокус на останньому обраному типі в фільтрах.

```
// Функція автоматично встановлює останньо-вибраний користувачем фільтер пошуку (за типом)
function setSortTypeText(sorting) {
    let id;
    switch (sorting) {
        case "rating":
            id = 'type1';
            break;
        case "amount_views":
            id = 'type2';
            break;
        case "created_at":
            id = 'type3';
            break;
    }
    document.getElementById(id).classList.toggle("filter-category-active");
}
```

Четверта функція зберігає останній вибраний вид впорядкування в фільтрах.

```
// Функція автоматично встановлює останньо-вибраний користувачем фільтер пошуку (за
зростанням)
function setSortGrowthText(sorting) {
    let id;
    switch (sorting) {
        case "desc":
            id = 'growth1';
            break;
        case "asc":
            id = 'growth2';
            break;
    }
    document.getElementById(id).classList.toggle("filter-category-active");
}
```

3.3 Механізм ролей у front-end

На сайті існують сторінки та елементи інтерфейсу, відображення яких залежить від рольової системи, в якій є 3 ролі: гість, зареєстрований користувач, адміністратор.

Для прикладу, в гостя немає можливості отримати доступ до форми керування постами на головній сторінці, зареєстровані користувачі можуть керувати лише своїми постами, а адміністратор взаємодіє з будь якою темою.

```
@if (Auth::check())
   @if(Auth::id() == $post->user['id'] || Auth::user()->hasRole('admin'))
      <div class="dropdown threeDots">
         <button class="threeDots-button" type="button" id="dropdownMenuButton" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false">
            <div><i class="bi bi-three-dots-vertical"></i></div>
         </button>
         <div class="dropdown-menu dropdown-additions" aria-labelledby="dropdownMenuButton">
            <a class="dropdown-item dropdown-item-additions" href="/posts/{{$post->id}}/edit"><i
class="bi bi-pencil indent"></i>Редагувати</a>

            <form action="/posts/{{$post->id}}" method="post">
               {{csrf_field()}}
               {!! method_field('delete') !!}
               <!-- Кнопка Видалення -->
               <button type="submit" class="dropdown-item dropdown-item-additions"><i class="bi bi-
trash3 indent"></i>Видалити</button>
            </form>

         </div>
      </div>
   @endif
@endif
```

Логіка оцінювання постів на сайті схожа на ту, що застосовується для тем. Усі користувачі, крім гостей, мають можливість виставляти оцінки постам. Однак, редагувати та видаляти пости можуть тільки автор поста та адміністратор.

```
@if (Auth::check())
      <div class="dropdown threeDots">
         <button class="threeDots-button" type="button" id="dropdownMenuButton" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false">
            <div><i class="bi bi-three-dots-vertical"></i></div>
         </button>
         <div class="dropdown-menu dropdown-additions" aria-labelledby="dropdownMenuButton">
            @if(Auth::id() == $post->user['id'] || Auth::user()->hasRole('admin'))
               <!-- Кнопка Редагування -->
            <a class="dropdown-item dropdown-item-additions" href="/posts/{{$post->id}}/edit"><i
class="bi bi-pencil indent"></i>Редагувати</a>

            <form action="/posts/{{$post->id}}" method="post">
               {{csrf_field()}}
               {!! method_field('delete') !!}
               <!-- Кнопка Видалення -->
               <button type="submit" class="dropdown-item dropdown-item-additions"><i class="bi bi-
trash3 indent"></i>Видалити</button>
            </form>
```

```
            @endif
            @include('posts.showLikeButtons')
            @include('posts.showBookmarkButtons')
          </div>
        </div>
@endif
```

3.4 Реалізація механізму запитів на сервер

3.2.1 Довідка про роботу проєкту

У даній роботі, клієнтська частина не містить жодних даних. Замість цього, для обробки певних маніпуляцій з даними, вона звертається до сервера. Цей процес здійснюється за допомогою механізму запитів, який представлений у вигляді форм (від назви тегу «form»). Нижче наведений приклад, щоб краще уявити, як вони виглядають.

```
<form action="/posts/{{$post->id}}" method="post">
   {{csrf_field()}}
   {!! method_field('delete') !!}
   <button type="submit"></button>
</form>
```

Форми обов’язково працюють по якійсь дії (action), яка залежить від системи маршрутизації, що надається серверним програмним забезпеченням. Взаємодіяти з сервером front-end розробник може лише методами post та get, хоча система маршрутизації має в собі загалом 4 типи запитів: get, post, patch та delete. Тож, чи можемо ми використовувати останні два запити? Так, і допомагає нам в цьому саме Blade-шаблонізатор із функцією «method_field».

Також, форма містить csrf_field функцію, що генерує приховане поле в HTML-формі з токеном CSRF, яке дозволяє захистити від атак на основі підробки запитів між сайтами. Цей токен зберігається в сесії на сервері. При відправці форми, значення CSRF з форми зіставляється зі значенням збереженого токена в сесії. Якщо вони співпадають, запит вважається довіреним і виконується, якщо ж ні - запит відхиляється.

3.2.2 Форми видалення посту

На форумі існують певні публікації, які ми можемо видалити за потреби. Для досягнення цієї мети створено відповідний код, який дозволяє видаляти ці публікації в зручний спосіб.

```
<form action="/posts/{{$post->id}}" method="post">
   {{csrf_field()}}
   {!! method_field('delete') !!}
   <!-- Кнопка Видалення -->
   <button type="submit" class="dropdown-item dropdown-item-additions"><i class="bi bi-trash3
indent"></i>Видалити</button>
</form>
```

Якщо автор теми хоче видалити свою тему, він має можливість це зробити через профіль, за допомогою готової функції:

```
<form action="/posts/deleteInProfile/{{$post->id}}" method="post">
   {{csrf_field()}}
   {!! method_field('delete') !!}
   <!-- Кнопка Видалення -->
   <button type="submit" class="dropdown-item dropdown-item-additions"><i class="bi bi-trash3
indent"></i>Видалити</button>
</form>
```

3.2.3 Форми видалення закладок

Профіль користувача містить сторінку з усіма збереженими закладками, де може окремо видалити зайві збереження.

```
<form action="/users/{{$user->id}}/bookmarks/{{$bookmark->pivot['id']}}" method="post">
   {{csrf_field()}}
   {!! method_field('delete') !!}
   <button type="submit" class="dropdown-item dropdown-item-additions"><i class="bi bi-award-fill
indent"></i>Видалити</button>
</form>
```

Якщо ж виявиться необхідним видалити усе зі списку збережених тем, існує функція видалення усіх закладок.

```
<form action="/users/{{$user->id}}/allDeleteBookmarks" method="post">
   {{csrf_field()}}
   {!! method_field('delete') !!}
   <button type="submit" class="button-apply">Видалити</button>
</form>
```

3.2.4 Форма пошуку

Ця форма дозволяє системі зібрати необхідну інформацію для складання запиту на пошук публікацій та передає його на сервер для подальшої обробки.

```
<form action="/search" class="searchBar-cust" method="get">
   <div class="search">
      @isset($searchText)
         <input class="searchBar-form" type="search" placeholder="Пошук" aria-label="Search" id="s"
name="s" autocomplete="off" value="{{$searchText}}">
      @else
         <input class="searchBar-form" type="search" placeholder="Пошук" aria-label="Search" id="s"
name="s" autocomplete="off">
      @endisset

      <button type="submit" class="dandruff-button"><i class="bi bi-search"></i></button>
   </div>
</form>
```

3.2.5 Форма створення категорії

На форумі є спеціальна сторінка з формою для створення категорій та підкатегорій. В ній ми маємо можливість задавати назву та, якщо це під категорія, задавати над категорію.

```
<form action="/categories/create" method="post">

   {{csrf_field()}}
   <div class="form-group">
      <label for="name"><h6>Назва:</h6></label>
      <input class="form-control cat-name-input" type="text" name="name" id="name"
value="{{old('name')}}">
   </div>

   <br>

   <div class="form-group">
      <label for="name"><h6>Надкатегорія (за необхідності):</h6></label>
      <div><select class="custom-select heirs-dropdown" name="heir_id">
         <option value="" selected>Немає</option>
         @foreach($basicCategories as $category)
```

```
            <option value={{$category->id}}>{{$category->name}}</option>
          @endforeach
        </select>
        </div>
    </div>

    <br>

    <div class="form-group">
        <button class="button-apply" type="submit">Створити</button>
    </div>

    @include('layouts.error')

</form>
```

3.2.6 Форма редагування категорії

У випадку, коли необхідно відредагувати вже існуючу категорію, існує функція редагування.

```
<form action="/categories/{{$category->id}}/edit" method="post">

    {{csrf_field()}}
    {!! method_field('patch') !!}


    <div class="form-group">
        <h6 for="name">Назва:</h6>
        <input class="input-group-text heirs-dropdown text-sm-left" type="text" name="name" id="name"
value="{{$category->name}}">
    </div>


    <div class="form-group">

        <div class="form-group">
            <button class="button-apply" type="submit">Оновити</button>
        </div>

    </div>

    @include('layouts.error')

</form>
```

3.2.7 Форма реєстрації

Для того щоб повноцінно користуватися сайтом, користувач має створити свій обліковий запис. Наступна форма має рядки для введення особистої інформації.

```
<form method="POST" action="{{ route('login') }}">
   @csrf
   <div>
      <div class="login-margin"><h5>Увійти в акаунт</h5></div>
      <div>
         <input id="email" type="email" class="form-control input-line @error('email') is-invalid
@enderror" name="email" placeholder="пошта" value="{{ old('email') }}"  required
autocomplete="email" autofocus>

         @error('email')
         <span class="invalid-feedback" role="alert">
                  <strong>{{ $message }}</strong>
               </span>
         @enderror
      </div>

      <div>
         <input id="password" type="password" class="form-control input-line @error('password') is-
invalid @enderror" placeholder="пароль" name="password" required autocomplete="current-
password">

         @error('password')
         <span class="invalid-feedback" role="alert">
                  <strong>{{ $message }}</strong>
               </span>
         @enderror
      </div>

      <br>
      <div class="interaction-margin">
         <button type="submit" class="btn regist-button">
            {{ __('Увійти') }}
         </button>

         @if (Route::has('password.request'))
            <a class="btn btn-link text-light" href="{{ route('password.request') }}">
               {{ __('Забув пароль?') }}
            </a>
         @endif

      </div>

      <div class="interaction-margin">
         <div class="pt-3">
         <input class="form-check-input" type="checkbox" name="remember" id="remember" {{
old('remember') ? 'checked' : '' }}>

         <label class="form-check-label" for="remember">
            {{ __("Запам'ятати мене") }}
         </label>
         </div>
      </div>
   </div>
</form>
```

3.2.8 Форма входу до облікового запису

Якщо обліковий запис вже існує, користувач має змогу увійти до нього за допомогою наступної форми:

```
<form method="POST" action="{{ route('login') }}">
   @csrf
   <div>
      <div class="login-margin"><h5>Увійти в акаунт</h5></div>
      <div>
         <input id="email" type="email" class="form-control input-line @error('email') is-invalid
@enderror" name="email" placeholder="пошта" value="{{ old('email') }}"  required
autocomplete="email" autofocus>

         @error('email')
         <span class="invalid-feedback" role="alert">
                  <strong>{{ $message }}</strong>
               </span>
         @enderror
      </div>

      <div>
         <input id="password" type="password" class="form-control input-line @error('password') is-
invalid @enderror" placeholder="пароль" name="password" required autocomplete="current-
password">

         @error('password')
         <span class="invalid-feedback" role="alert">
                  <strong>{{ $message }}</strong>
               </span>
         @enderror
      </div>

      <br>
      <div class="interaction-margin">
         <button type="submit" class="btn regist-button">
            {{ __('Увійти') }}
         </button>

         @if (Route::has('password.request'))
            <a class="btn btn-link text-light" href="{{ route('password.request') }}">
               {{ __('Забув пароль?') }}
            </a>
         @endif

      </div>

      <div class="interaction-margin">
         <div class="pt-3">
         <input class="form-check-input" type="checkbox" name="remember" id="remember" {{
old('remember') ? 'checked' : '' }}>

         <label class="form-check-label" for="remember">
            {{ __("Запам'ятати мене") }}
         </label>
         </div>
      </div>
   </div>
</form>
```

3.5 Налаштування редактору Tiny MCE

3.5.1 Основні відомості про налаштування

В IT-форумі існує система ролей, де для кожної ролі існує свій список можливостей. Серед таких можливостей – елементи редагування тексту.

На рис. 3.4 зображено інтерфейс редактору для ролі «Зареєстрований користувач».

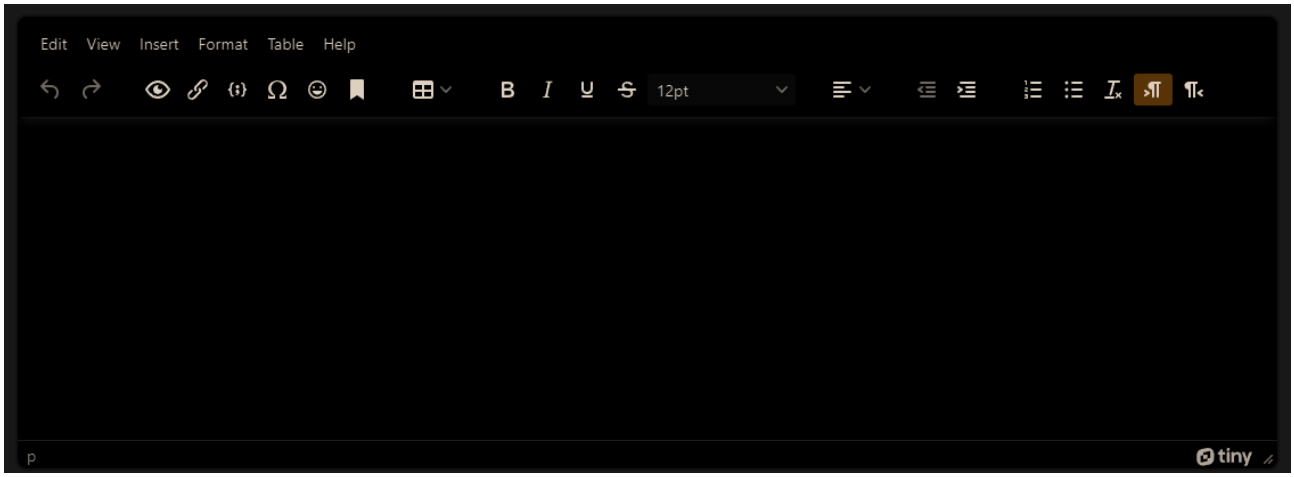

Рисунок 3.4 – Інтерфейс редактору для зареєстрованого користувача

На рис. 3.5 зображено інтерфейс редактору для ролі «Адміністратора».

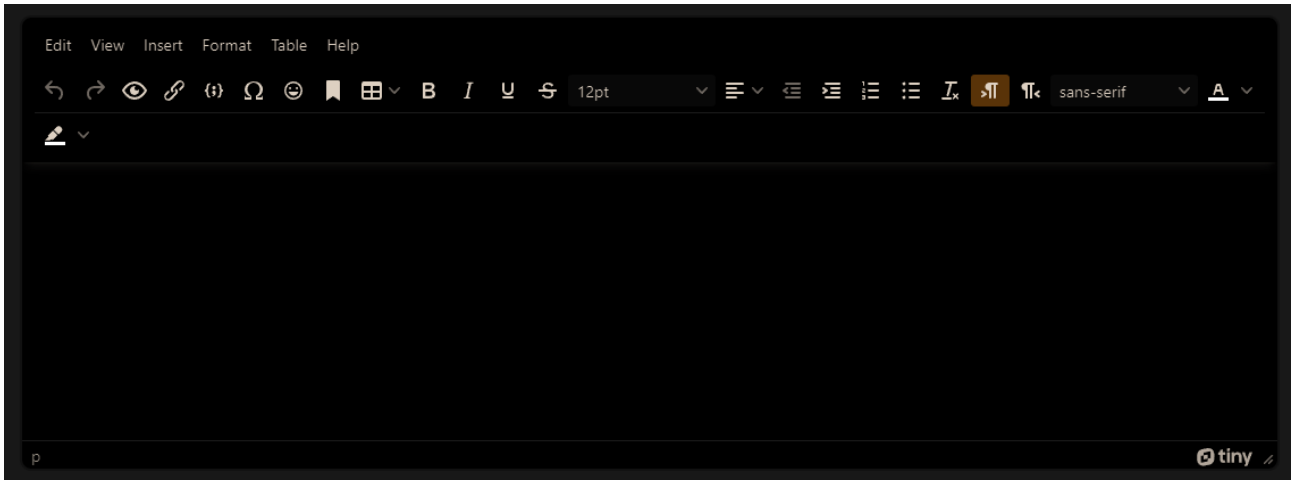

Рисунок 3.5 – Інтерфейс редактору для адміністратора

Незареєстровані користувачі не мають змоги створювати пости та писати коментарі до тем, тож інтерфейсу для них не передбачено.

3.5.2 Панель редагування публікації

Для «зареєстрованого користувача» передбачено обмежений список модифікацій тексту:

– відміна та повторення останньої дії;

– попередній перегляд;

– вбудовування посилання;

– вбудовування коду;

– вставка спеціальних символів;

– вставка реакцій;

– вставка якоря;

– таблиця;

– зміна стилю тексту;

– опція вирівнювання тексту;

– вставка відступу;

– списки;

– очистка стилів;

– горизонтальна лінія;

– вставка часу.

У випадку з «адміністраторами», список модифікацій, який був доступний у звичайного користувача, доповнюється.

– зміна кольору тексту;

– зміна кольору фону тексту;

– зміна шрифту.

3.5.3 Панель редагування коментарів

Редактор коментаря має менший функціонал в порівнянні з постом, що обумовлене концепцією дружнього інтерфейсу. Коментарі не мають перевантажувати сторінку зайвими деталями. Тож, панель редагування коментарів містить в собі наступне:

– попередній перегляд;
– вбудовування посилання;
– вбудовування коду;
– вставка спеціальних символів;
– вставка реакцій;
– зміна стилю тексту;
– опція вирівнювання тексту;
– вставка відступу;
– списки;
– очистка стилів.

3.6 Візуальна складова клієнтської частини

3.6.1 Головна сторінка

Головна сторінка, це перше, що бачить користувач, коли заходить на форум. На ній зображується список останніх постів, їх автори та статистика. Також, в залежності від того, чи є користувач адміністратором, чи ні, зовнішній вигляд може трохи відрізнятися. Так, звичайний користувач буде бачити інтерфейс, як зображено на рис. 3.6.

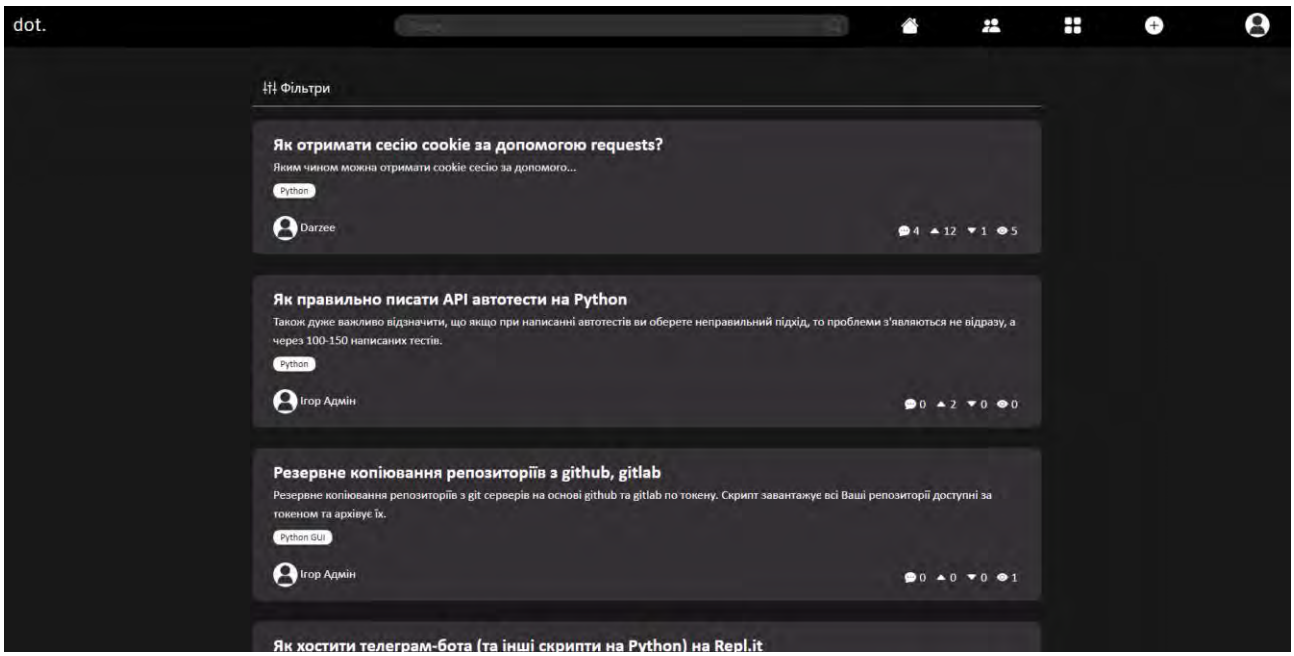

Рисунок 3.6 – Інтерфейс головної сторінки очима звичайного користувача

Натомість адміністратору стане доступним меню взаємодії з постом, за допомогою якого він зможе редагувати чи видаляти його. Відмінність інтерфейсу можна побачити на рис. 3.7.

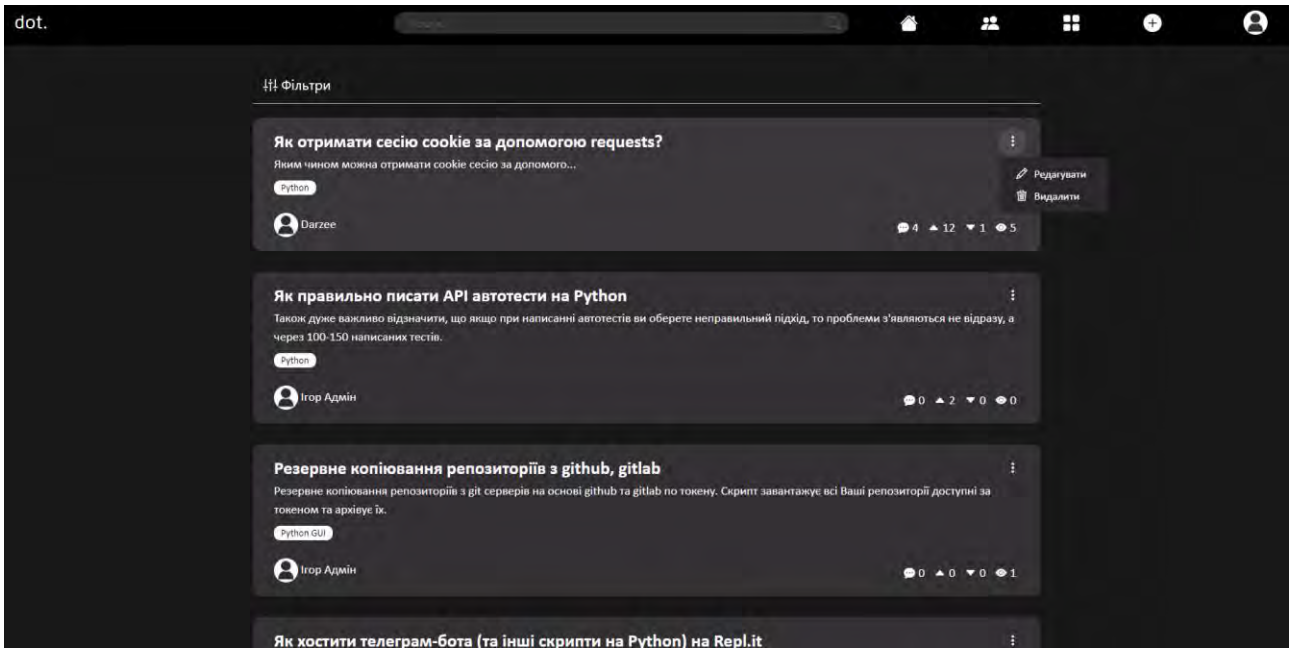

Рисунок 3.7 – Інтерфейс головної сторінки очима адміністратора

3.6.2 Сторінка посту

На сторінці посту користувач може знайти інформацію про тему. У темі можна не тільки ознайомитися з її заголовком, а й отримати інформацію про автора та дату її публікації. Крім того, тема може містити інформацію про категорію, до якої вона належить, що дозволяє легко зорієнтуватися в контексті.

На сторінці посту є поле для коментування, яке дозволяє користувачам вільно висловлювати свої думки та спілкуватися з іншими учасниками форуму. Коментатор може використовувати графічний редактор тексту, щоб зробити свою відповідь більш читабельною та зрозумілою.

До того ж, можна знайти список коментарів, які були опубліковані під темою. Це дає можливість користувачам обговорювати тему, ділитися своїми думками та висловлювати свої погляди на неї. Крім того, користувач може відповідати на коментарі інших користувачів, що забезпечує взаємодію та обмін інформацією між учасниками форуму.

На рис. 3.8 зображено приклад теми.

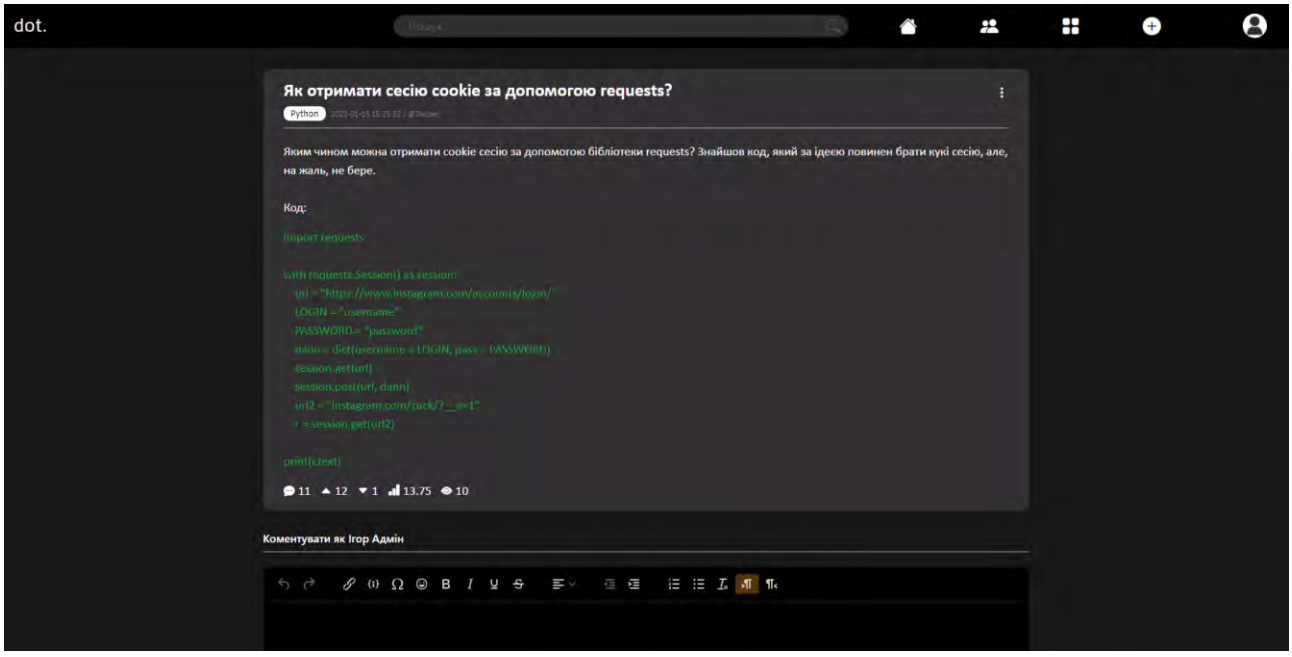

Рисунок 3.8 – Зображення теми

На рис. 3.9 зображено приклад списку коментарів.

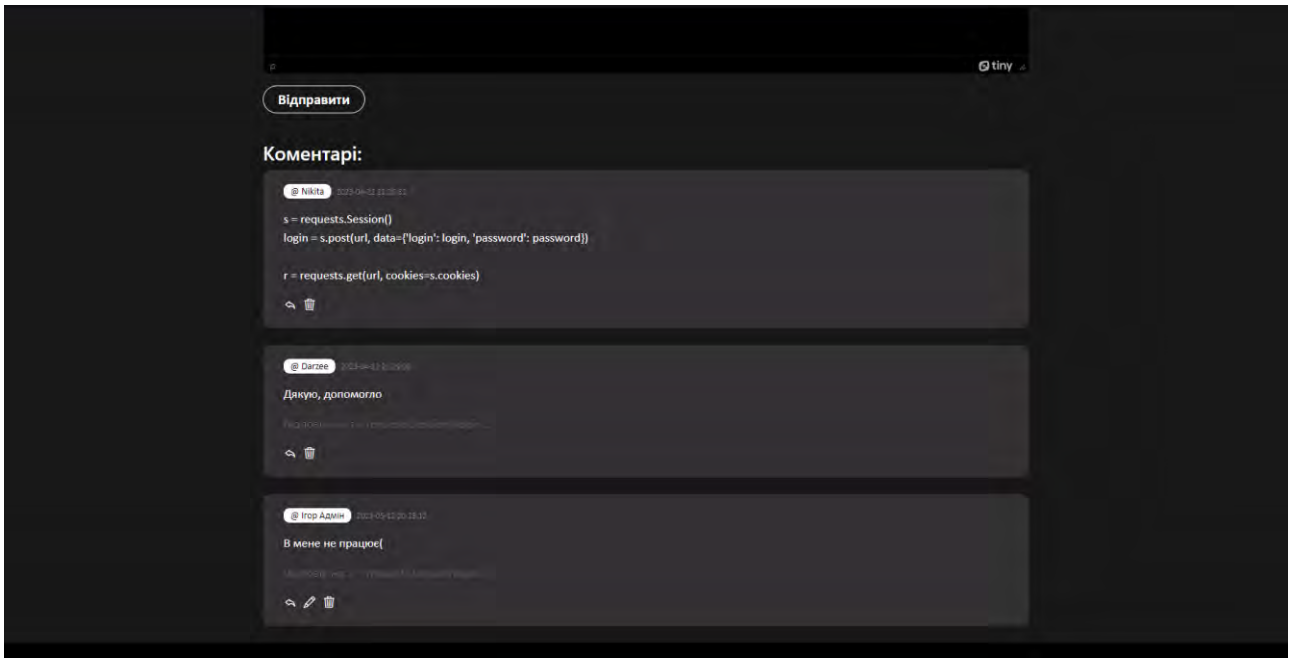

Рисунок 3.9 – Зображення списку коментарів

На рис. 3.10 зображено створення відповіді на коментар.

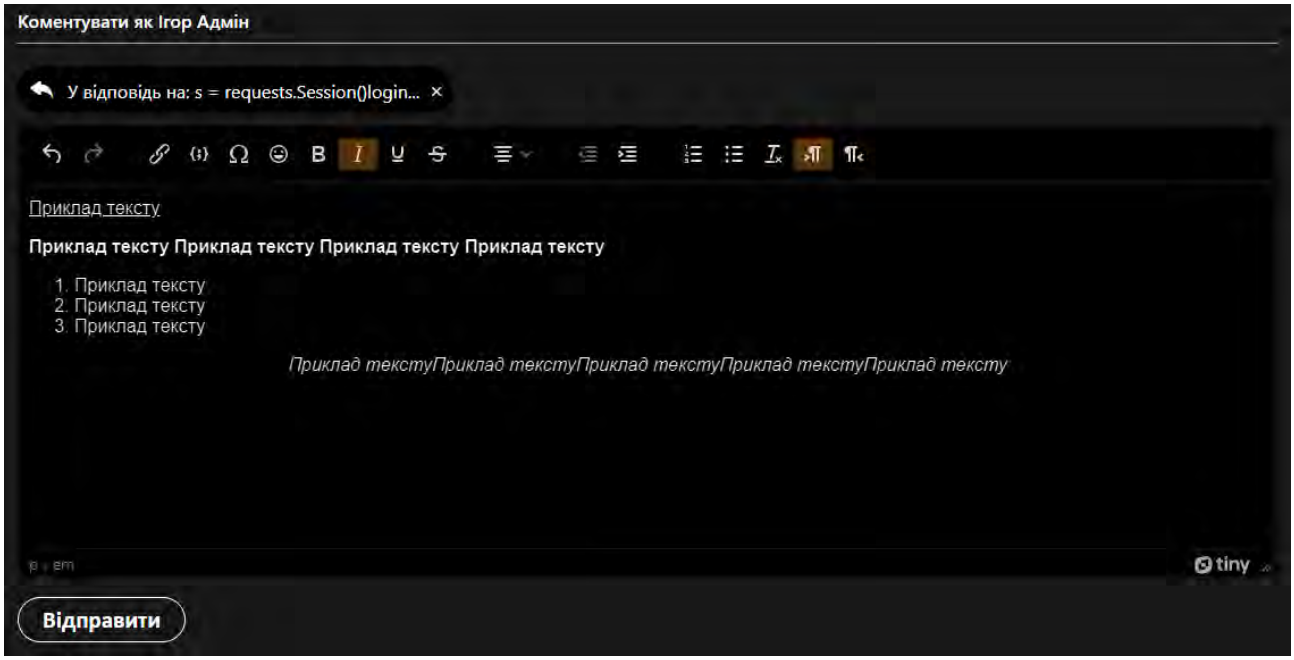

Рисунок 3.10 – Зображення створення відповіді

На рис. 3.11 зображено створену відповідь.

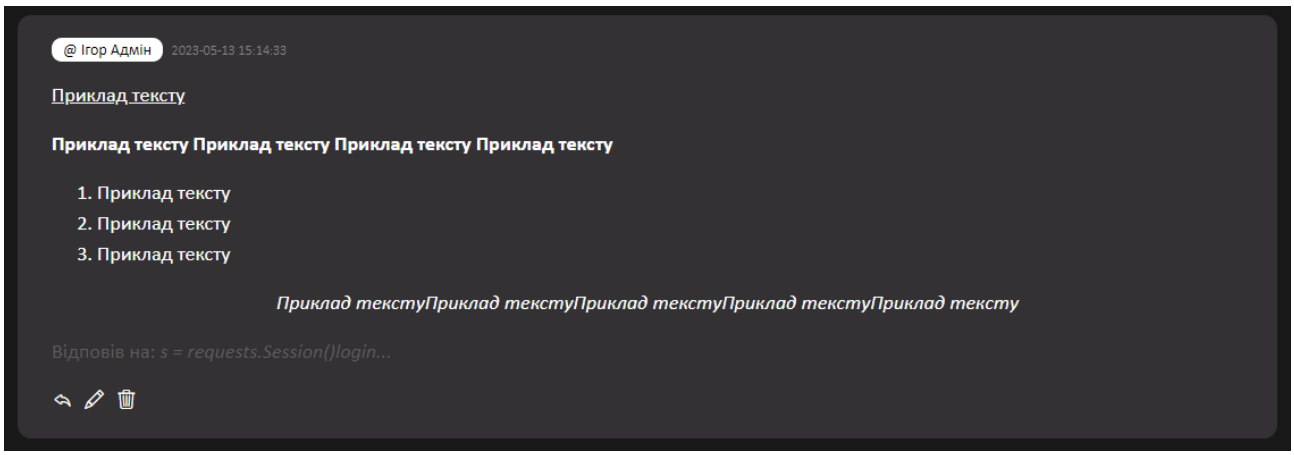

Рисунок 3.11 – Створена відповідь

3.6.3 Сторінка списку користувачів

В IT-форумі є можливість переглядати список користувачів, для чого відведена окрема сторінка, до якої можна перейти через панель навігації. Він розбитий на плитки, в кожній з яких зображується фотографія, ім’я, псевдонім, кількість створених тем та написаних коментарів.

Сайт враховує, що кількість зареєстрованих користувачів може сягати великої кількості, тому на сторінці передбачено пагінацію. Список можна побачити на рис. 3.12.

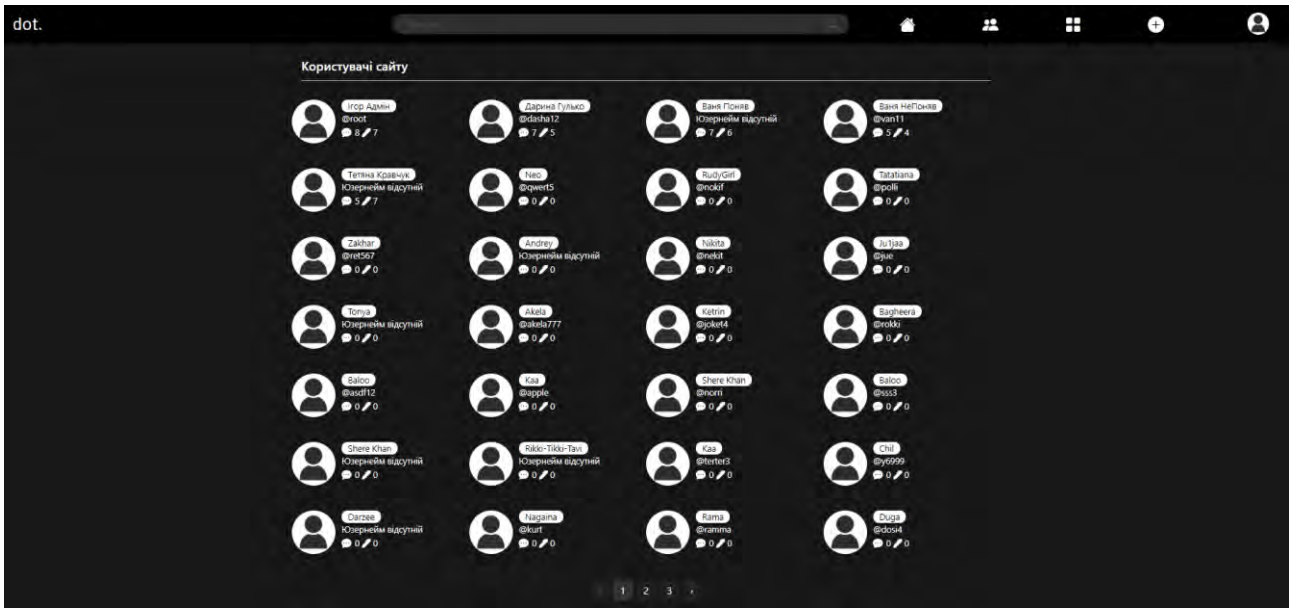

Рисунок 3.12 – Сторінка списку зареєстрованих користувачів

3.6.4 Сторінка категорій

Одна зі сторінок, доступних через панель навігації, присвячена категоріям. Тут вони відображені у вигляді плиток, включаючи не лише основні категорії, але й підкатегорії.

Подібно до списку постів, ця сторінка також відрізняється залежно від прав доступу користувача, який звернувся до неї.

На рис. 3.13 зображена сторінка так, як би її бачив звичайний користувач.

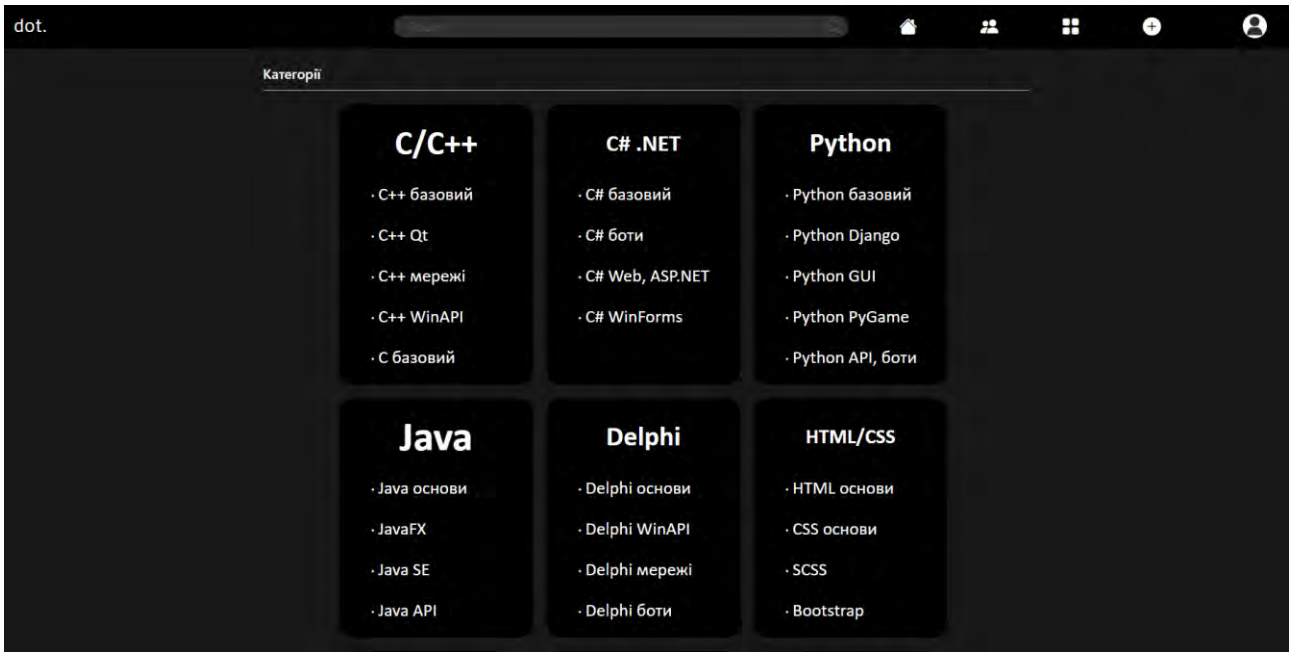

Рисунок 3.13 – Сторінка категорій очима звичайного користувача

При вході на сторінку в якості адміністратора, на екрані з'являються дві кнопки (див. рис. 3.14), які дозволяють взаємодіяти з категоріями: редагування та створення нових.

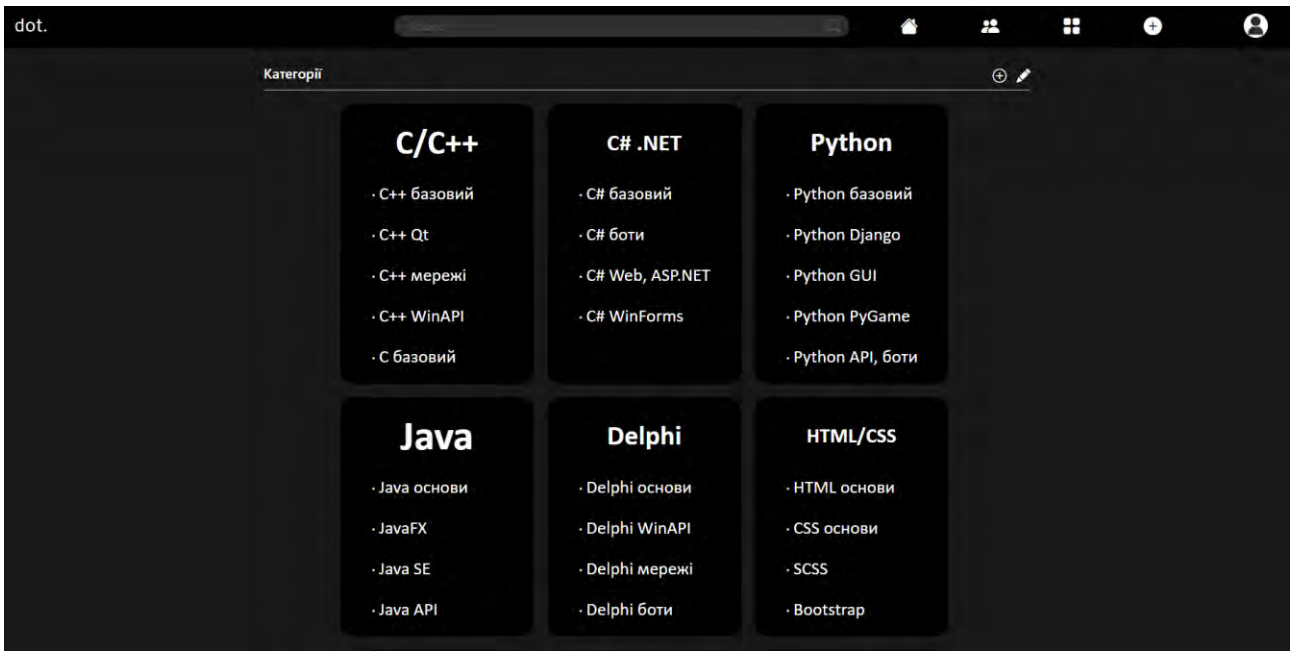

Рисунок 3.14 – Сторінка категорій очима адміністратора

При обранні будь-якої категорії, користувач отримає доступ до сторінки з усіма постами, які належать до цієї категорії, представленими в зручному форматі. Побачити це можна на рис. 3.15.

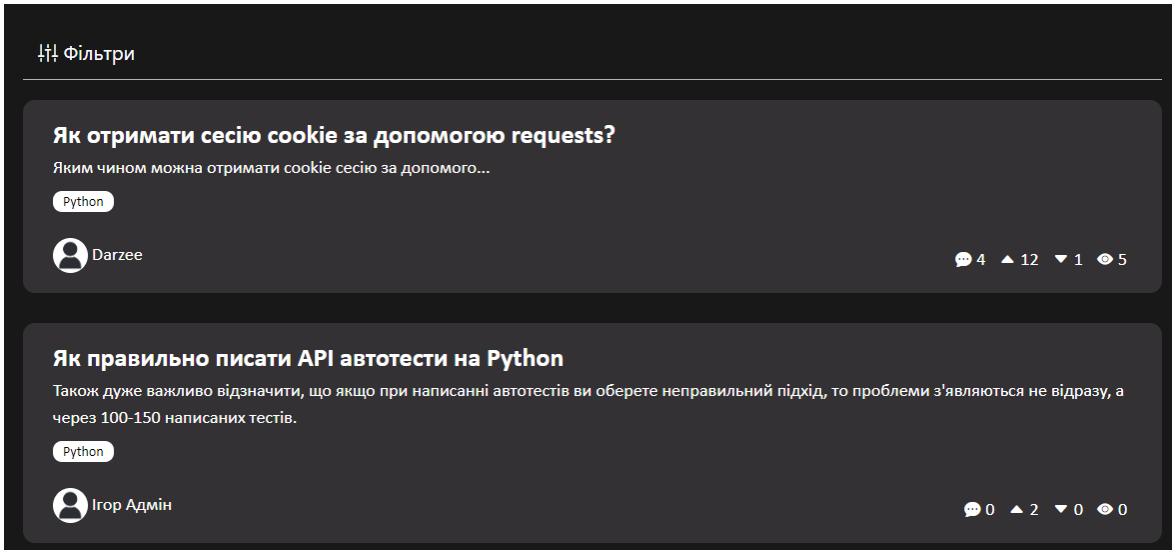

Рисунок 3.15 – Список постів певної категорії

Коли адміністратору потрібно додати нові категорії або підкатегорії, він переходить на сторінку зі спеціальною формою, де вказує назву та, якщо потрібно, надкатегорію. Сторінка зображена на рис. 3.16.

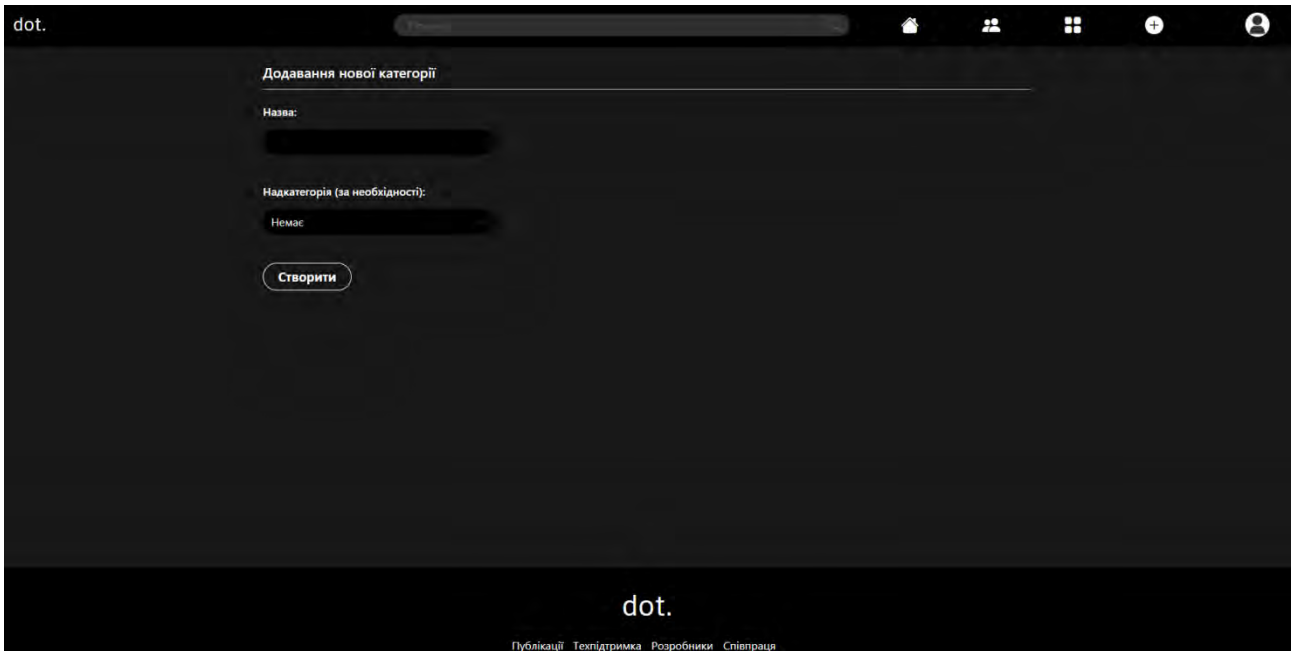

Рисунок 3.16 – Сторінка додавання нової категорії

У разі необхідності зміни категорій, адміністратор може натиснути відповідну кнопку з зображенням "олівця", що перенаправить його на сторінку вибору необхідної категорії (див. рис. 3.17). На цій сторінці можна внести необхідні зміни, зокрема видалити чи змінити назву категорії, використовуючи інтерфейс для редагування (див. рис. 3.18).

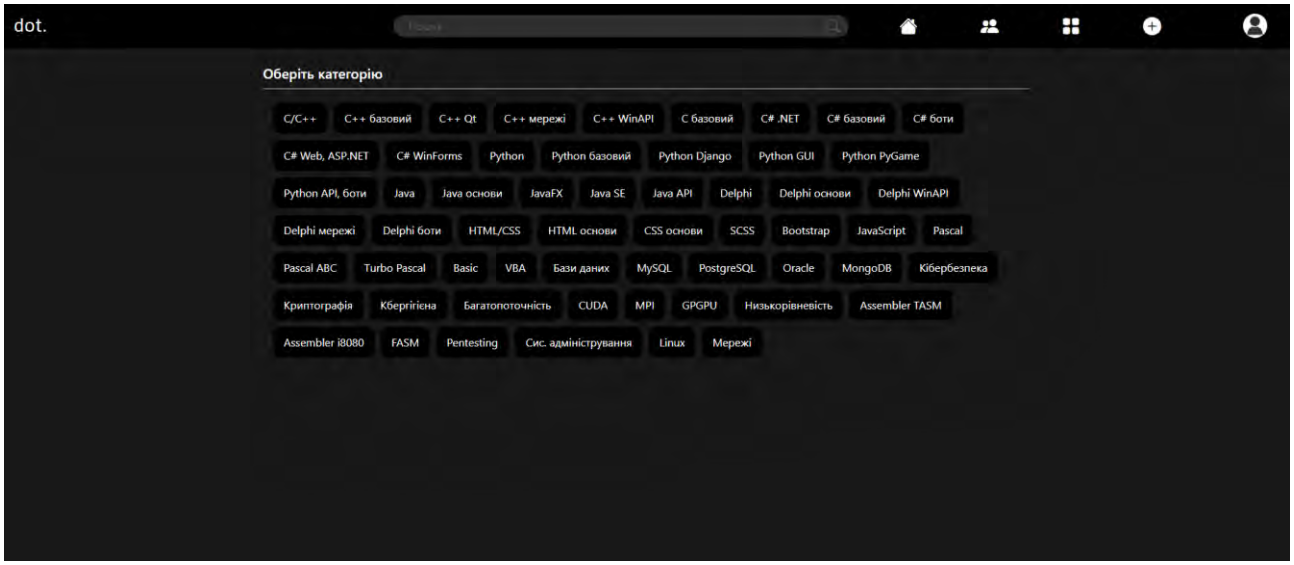

Рисунок 3.17 – Сторінка обирання категорії

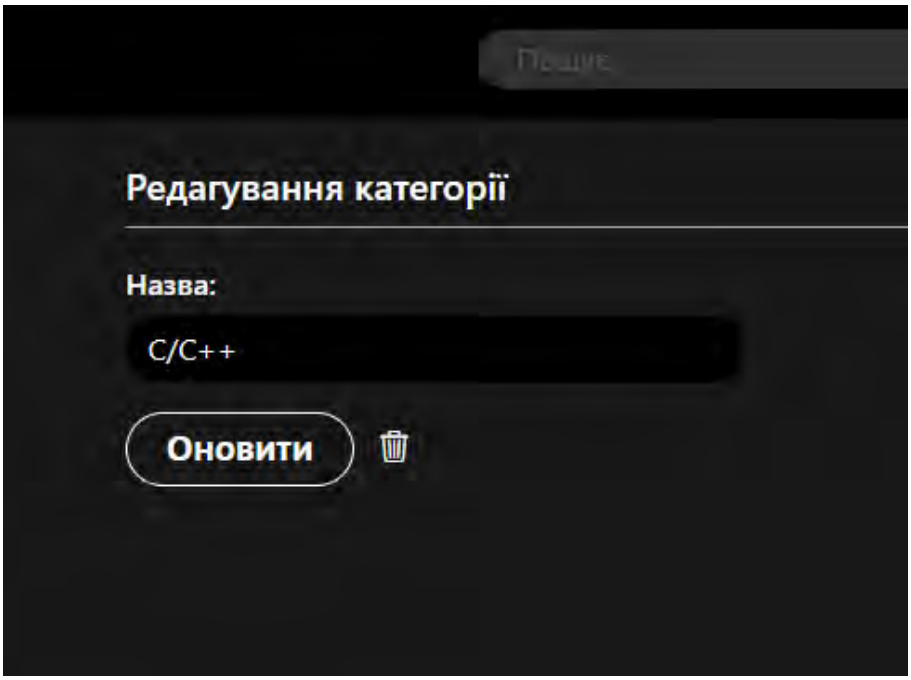

Рисунок 3.18 – Інтерфейс редагування категорії

Щоб запобігти випадкове видалення, передбачено вигулькове вікно, зображене на рис. 3.19.

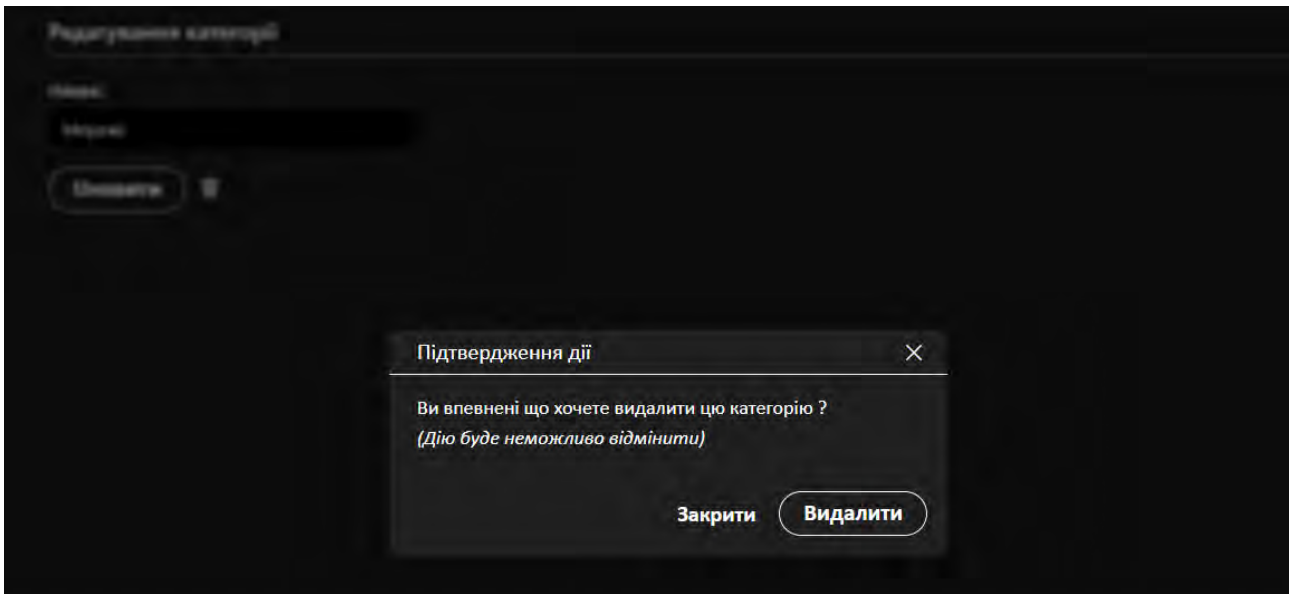

Рисунок 3.19 – Вигулькове вікно попередження про видалення

3.6.5 Сторінка створення теми

Теми мають шаблон, що містить рядок заголовку, категорію та редактор опису. Сторінка зображена на рис. 3.20.

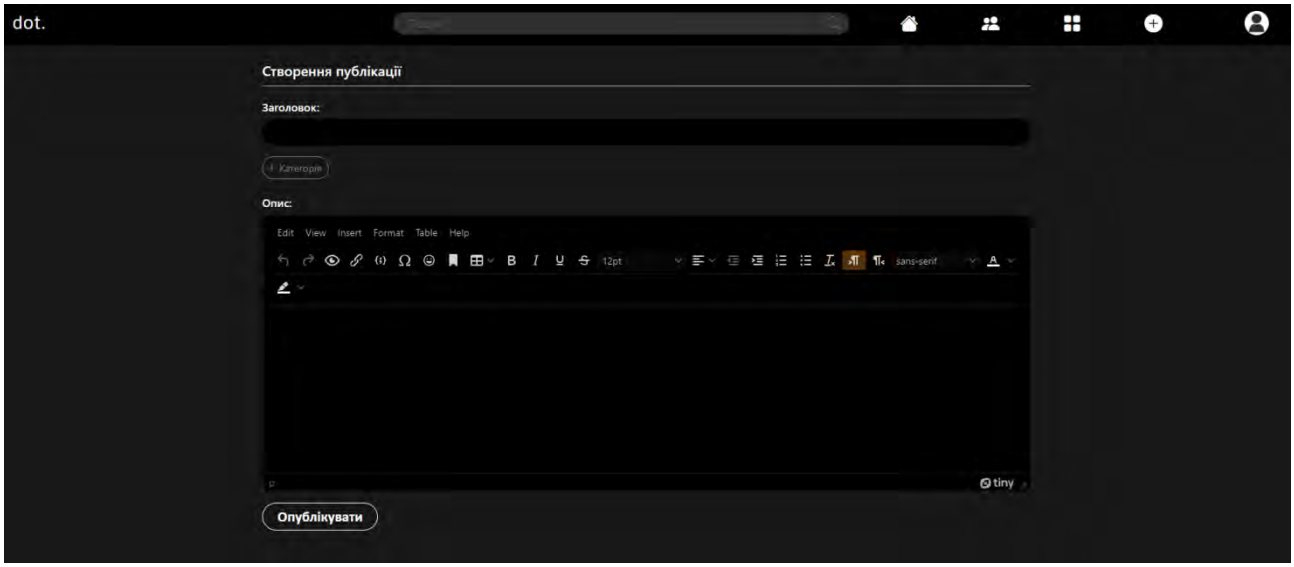

Рисунок 3.20 – Сторінка створення теми

3.6.6 Сторінка опису сайту

Сторінка опису сайту є невід’ємною частиною будь-якого вебпроєкту, що дозволяє користувачам знайомитися з основними відомостями про сайт. Перейти до неї можна через логотип сайту, що знаходиться в лівому верхньому кутку. Вигляд сторінки зображено на рис. 3.19 – 3.21.

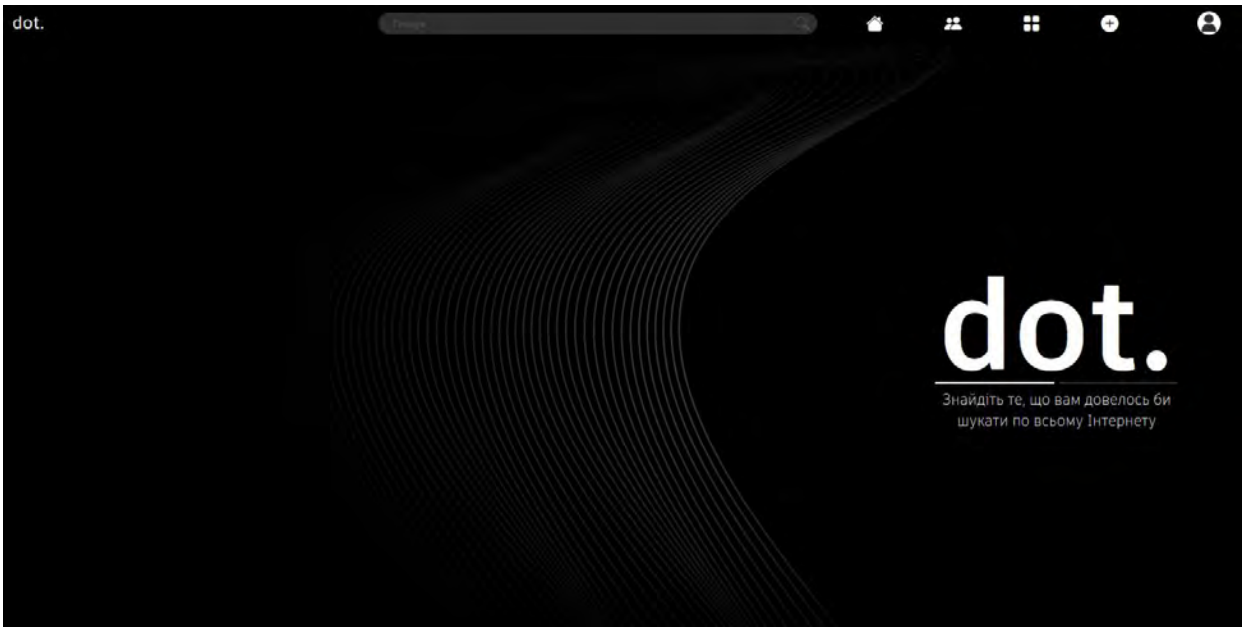

Рисунок 3.19 – Сторінка опису сайту, частина перша

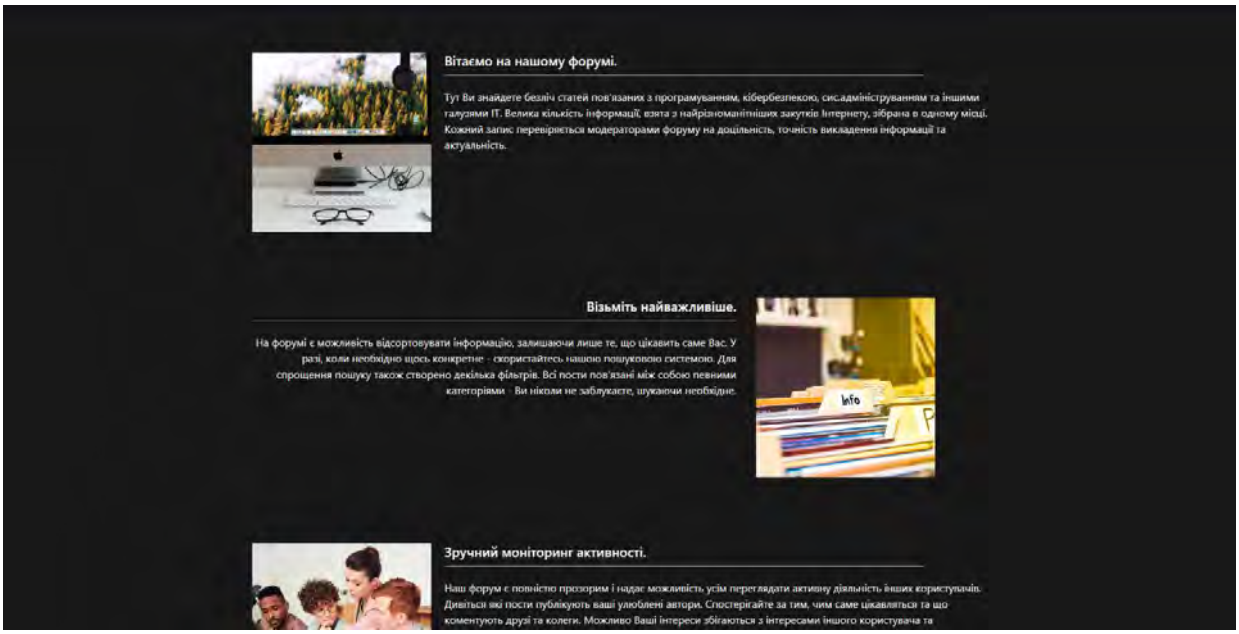

Рисунок 3.20 – Сторінка опису сайту, частина друга

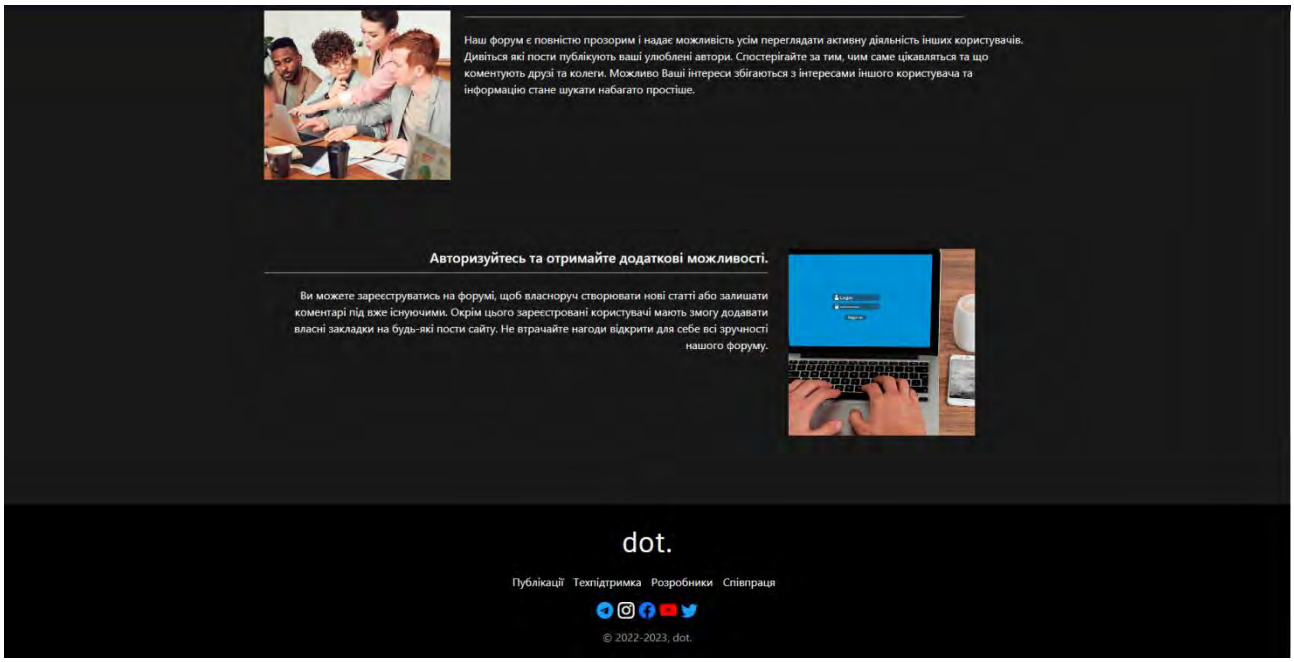

Рисунок 3.21 – Сторінка опису сайту, частина третя

3.6.7 Сторінка профілю користувача

Останнім елементом в панелі навігації є фотографія користувача, при натисненні на яку відкриється панель взаємодії з акаунтом, що зображено на рис. 3.22.

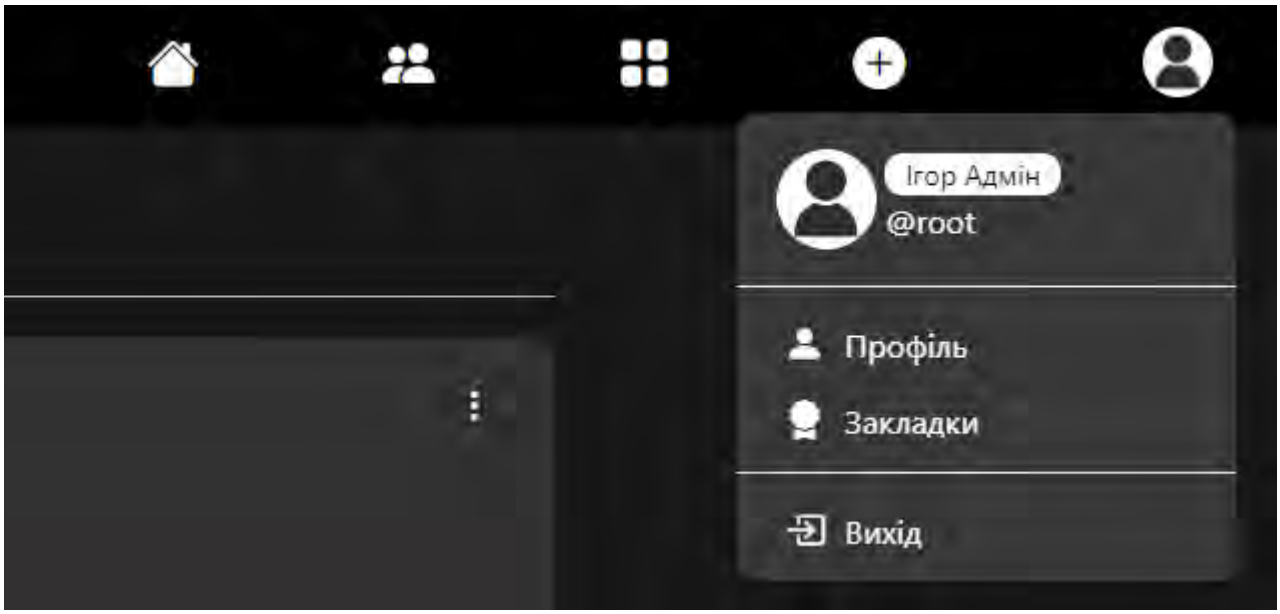

Рисунок 3.22 – Панель керування акаунтом

Як видно з рис. 3.22, користувач на цій панелі зможе побачити своє ім’я, псевдонім, перейти до профілю та сторінки із закладками, або взагалі вийти з облікового запису.

На рис. 3.23 зображено інтерфейс профілю, в якому знаходиться багато елементів. В першу чергу, ми бачимо інформацію про користувача, яку завжди можна змінити через спеціально відведену для цього сторінку, яку бачимо на рис. 3.24.

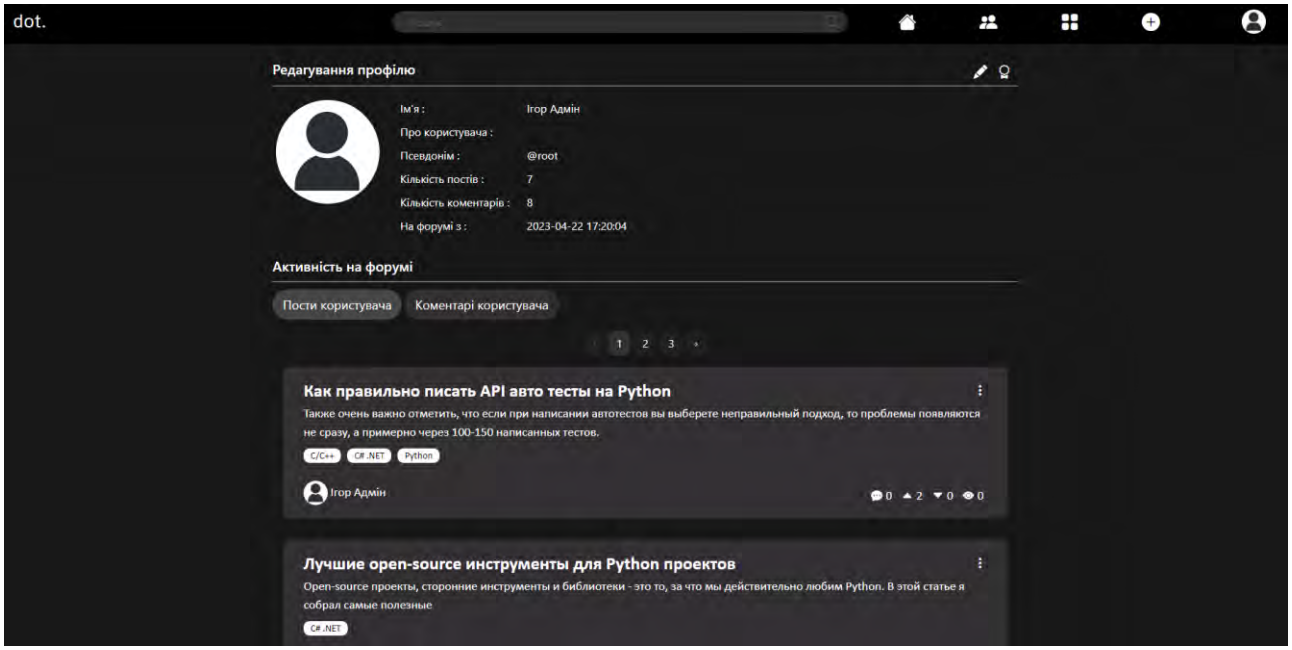

Рисунок 3.23 – Профіль користувача

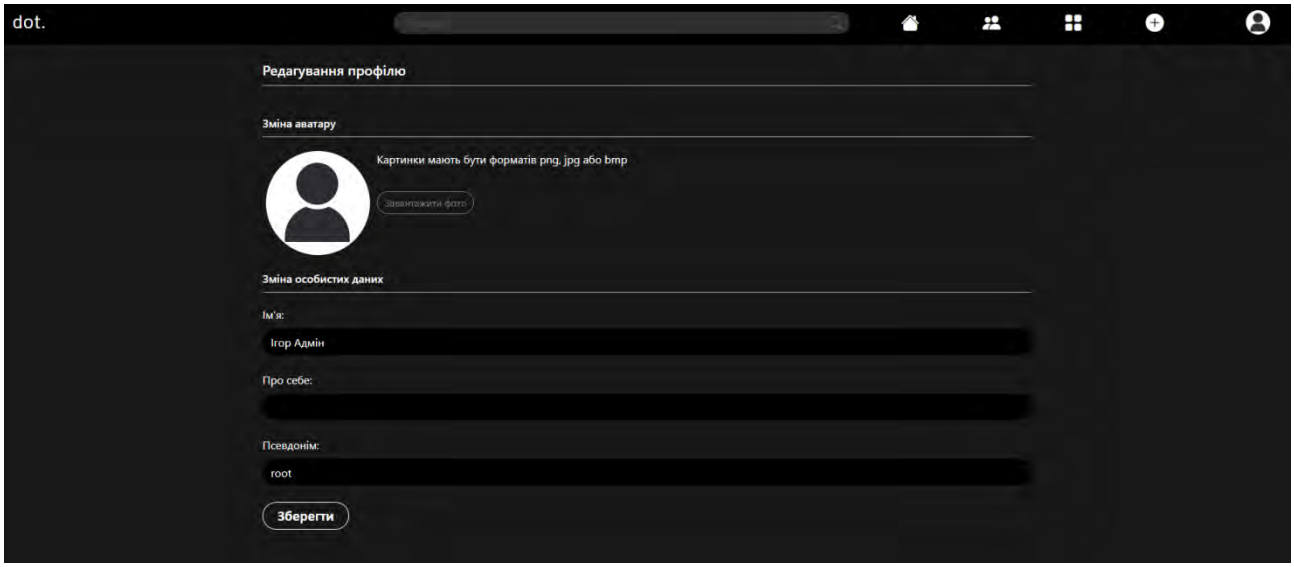

Рисунок 3.24 – Сторінка редагування даних про користувача

Змінюючи дані про себе, користувач має змогу змінити свою фотографію. Приклад зміни фотографії можна побачити на рис. 3.25 – 3.26.

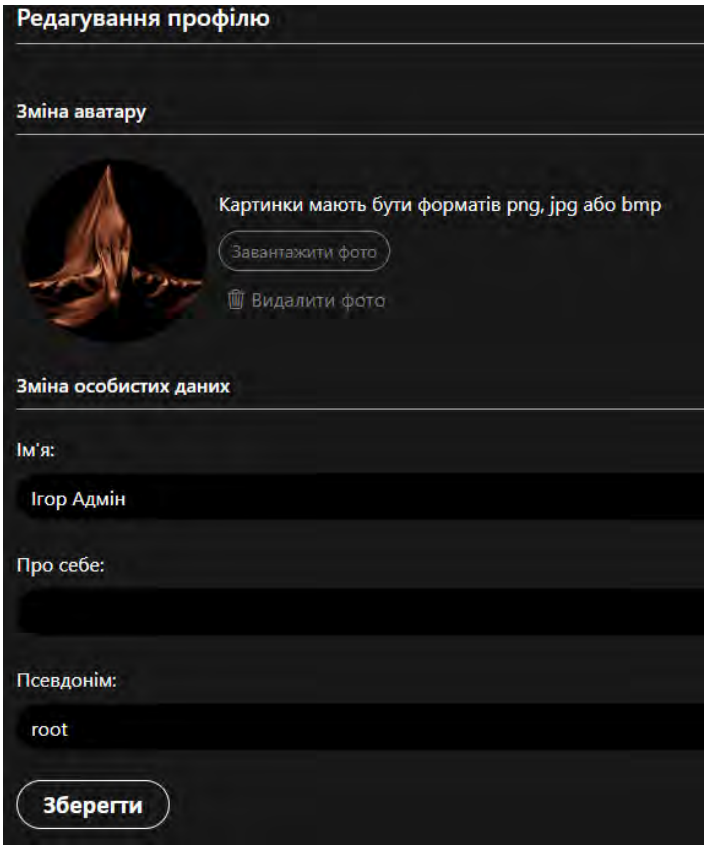

Рисунок 3.25 – Завантажене фото в редакторі

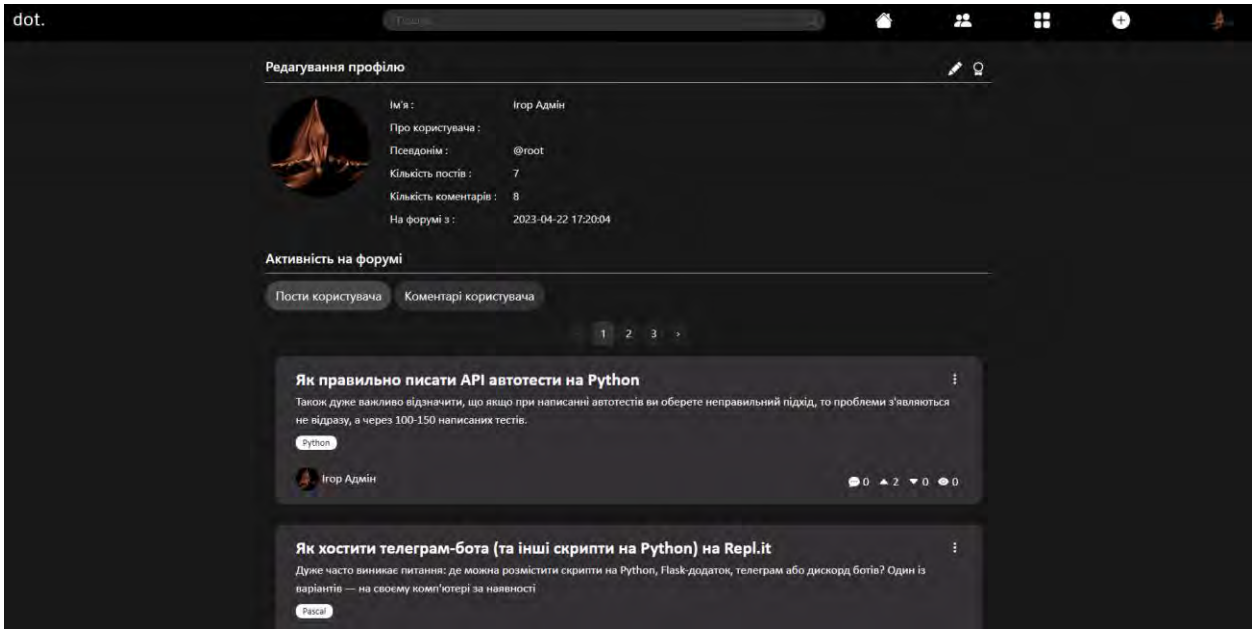

Рисунок 3.26 – Завантажене фото в профілі

3.6.8 Сторінка закладок

Ця сторінка містить усі закладки користувача (див. рис. 3.27). Тут можна переходити на збережені теми, або видалити їх зі списку. В разі, якщо користувачу більше не потрібні усі ці закладки, він може видалити їх, натиснувши «Видалити всі закладки», проте сайт застережить про те, що це незворотній процес. Вигулькове вікно, яке скаже про це, зазначено на рис. 3.28.

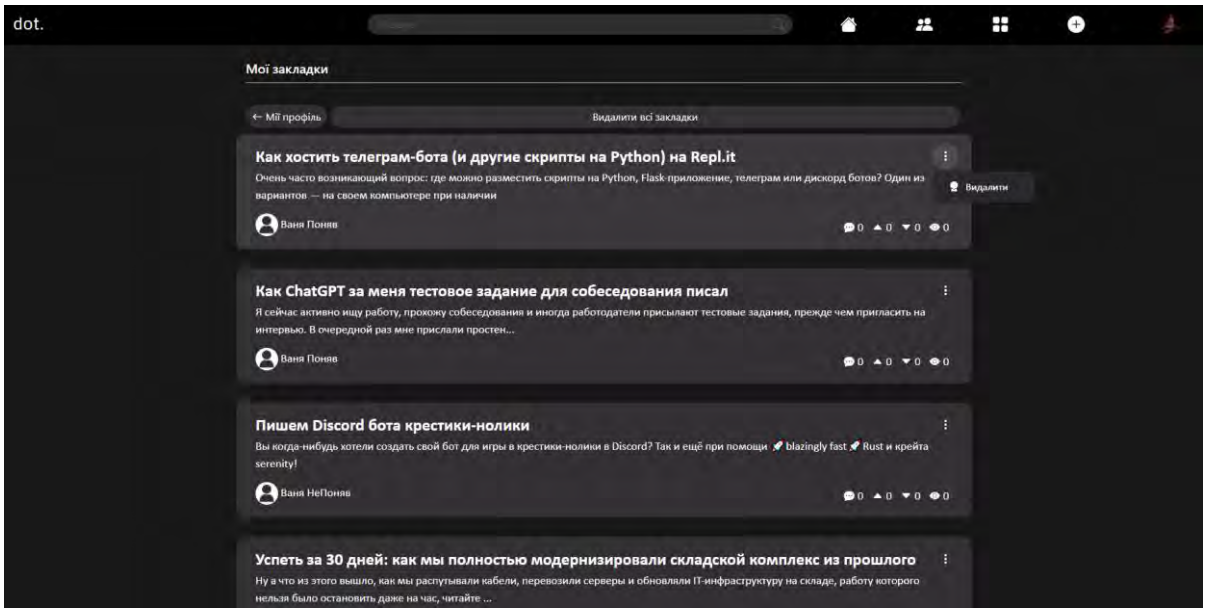

Рисунок 3.27 – Зображення списку закладок

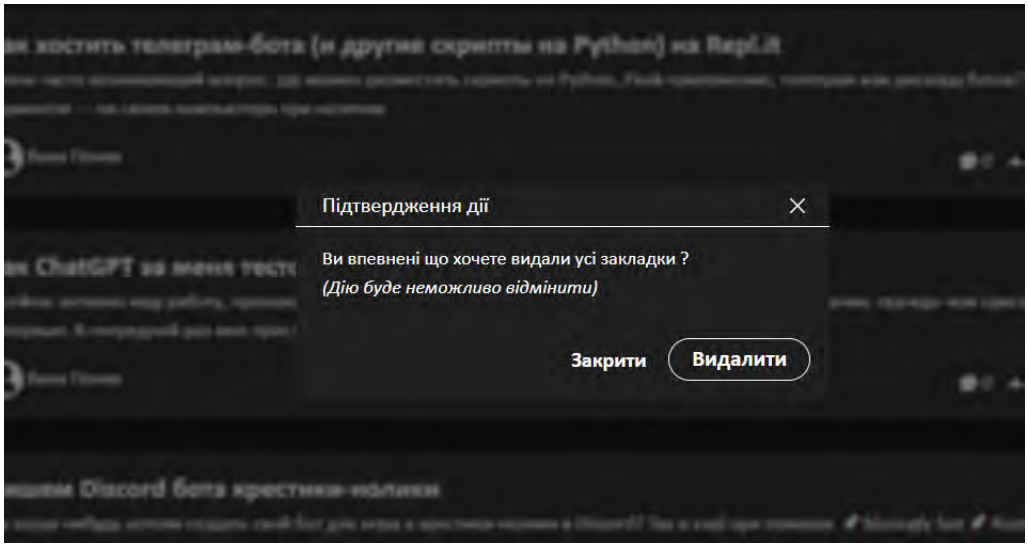

Рисунок 3.28 – Вигулькове вікно про незворотність дії

3.6.9 Сторінка входу до облікового запису

Для того, щоб отримати доступ до входу до свого профілю, необхідно спочатку до нього увійти. Для цього в панелі навігації потрібно обрати «Увійти», після чого нас перемістить до сторінки входу, яка зображена на рис. 3.29.

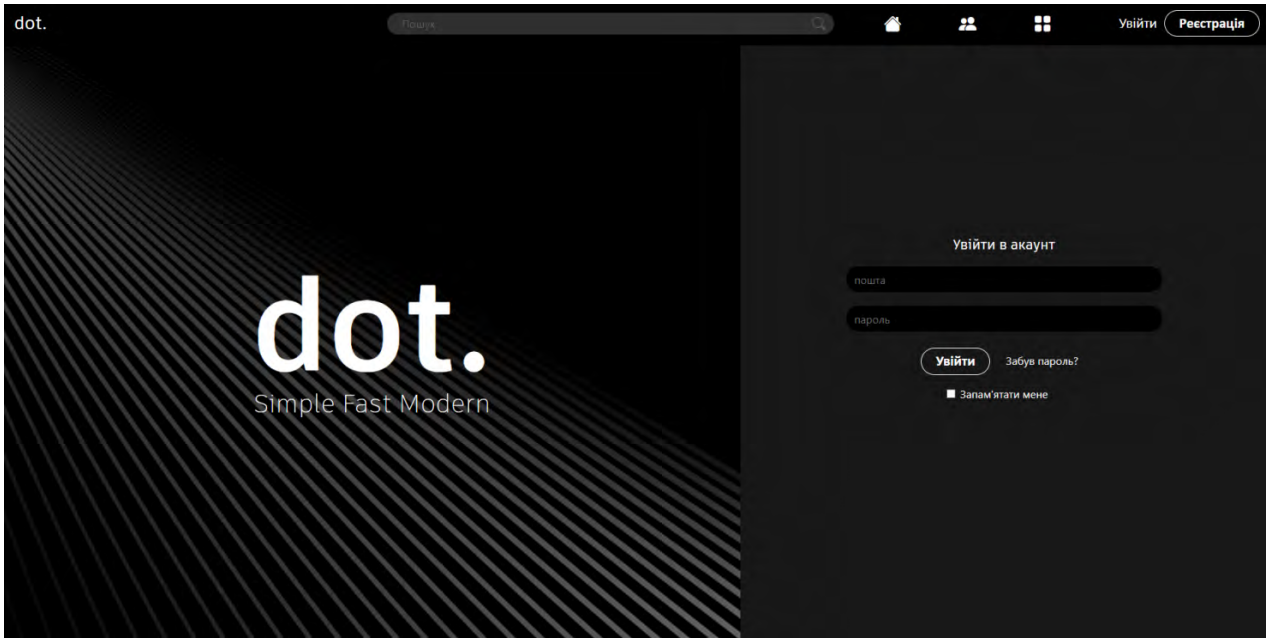

Рисунок 3.29 – Сторінка входу

Якщо користувач помилився з введеними даними, виведуться поля застереження, в яких буде сказано про невірно заповнені рядки. Помилки можна побачити на рис. 3.30.

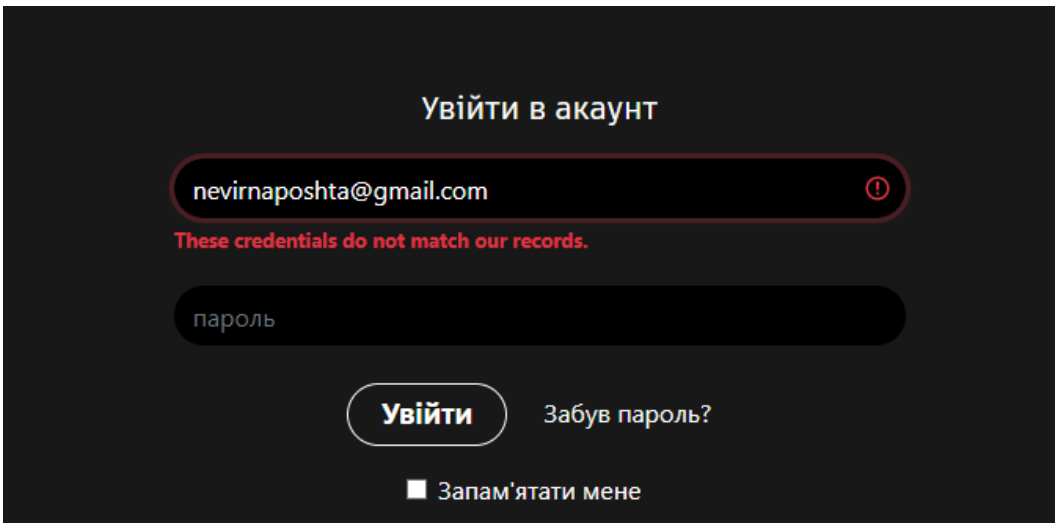

Рисунок 3.30 – Помилка, якщо введено дані від неіснуючого аканту

3.6.10 Сторінка реєстрації нового облікового запису

У випадку, якщо користувач ще не має свого облікового запису, він може його створити на сторінці реєстрації, зображеної на рис. 3.31.

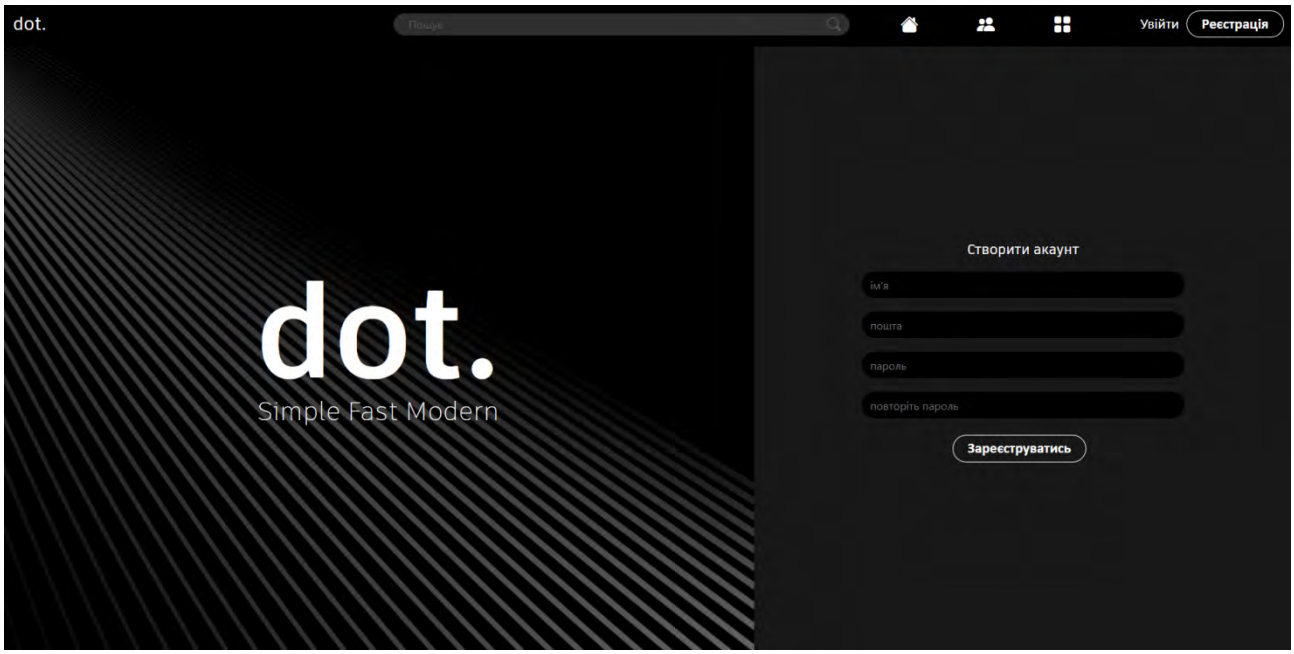

Рисунок 3.31 – Сторінка реєстрації

Для успішної реєстрації, гостю потрібно ввести правильно дані. Наприклад, якщо в рядку пошти не використати закінчення у форматі «@mail.com», то зазначиться помилка, зображена на рис. 3.32.

Також існує правило для паролей – вони мають бути довжиною не менше восьми символів, про що зазначається на рис. 3.33.

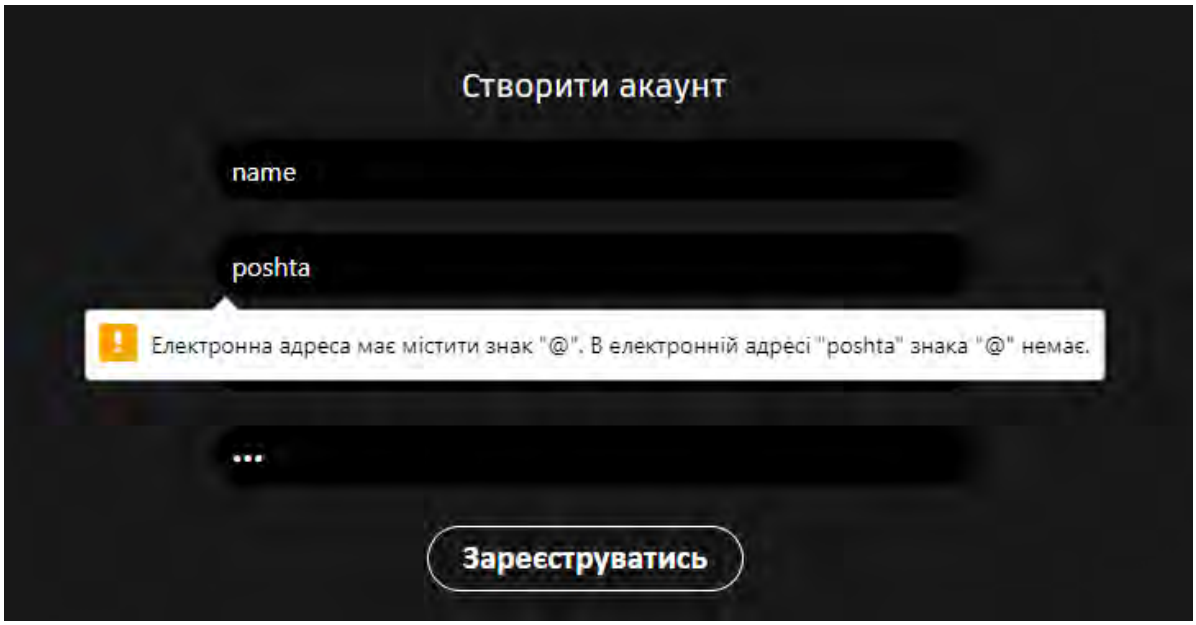

Рисунок 3.32 – Помилка реєстрації, неправильно написана пошта

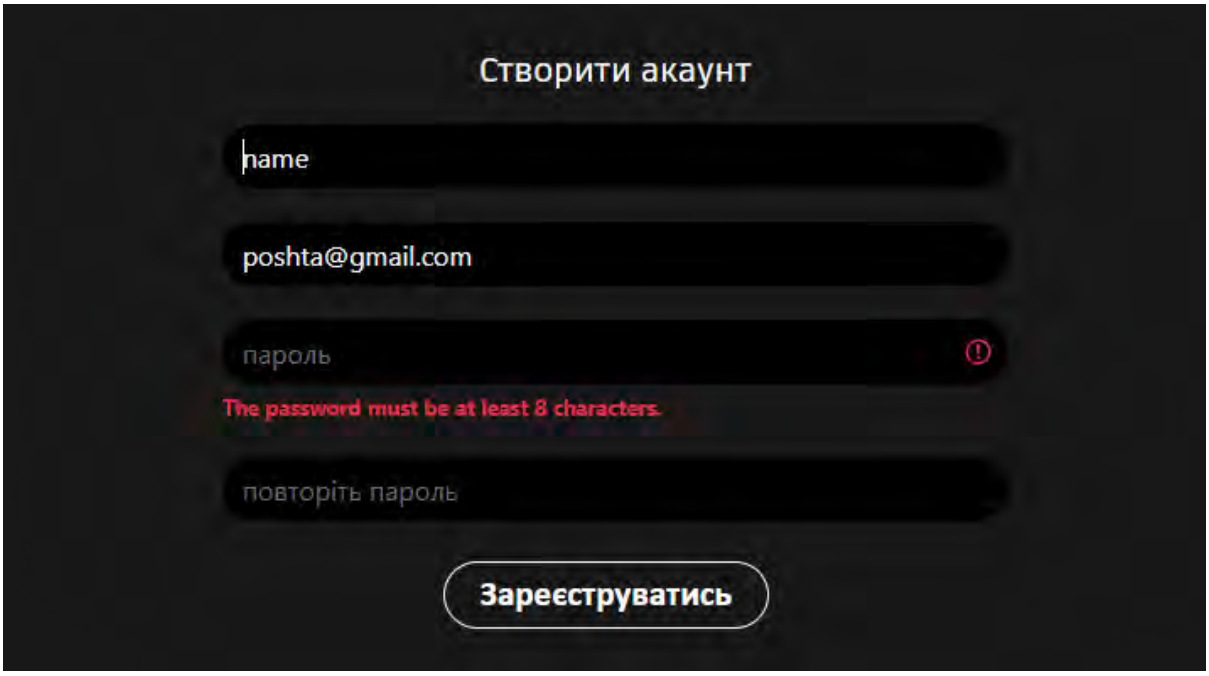

Рисунок 3.33 – Помилка реєстрації, неправильно написаний пароль

3.6.11 Форми помилок

На форумі іноді трапляються випадки, коли на сторінках немає ніяких записів або елементів, тож для цього потрібно передбачити форми, які будуть казати користувачам про ці проблеми. Приклад таких форм наведено в рис. 3.34 – 3.35.

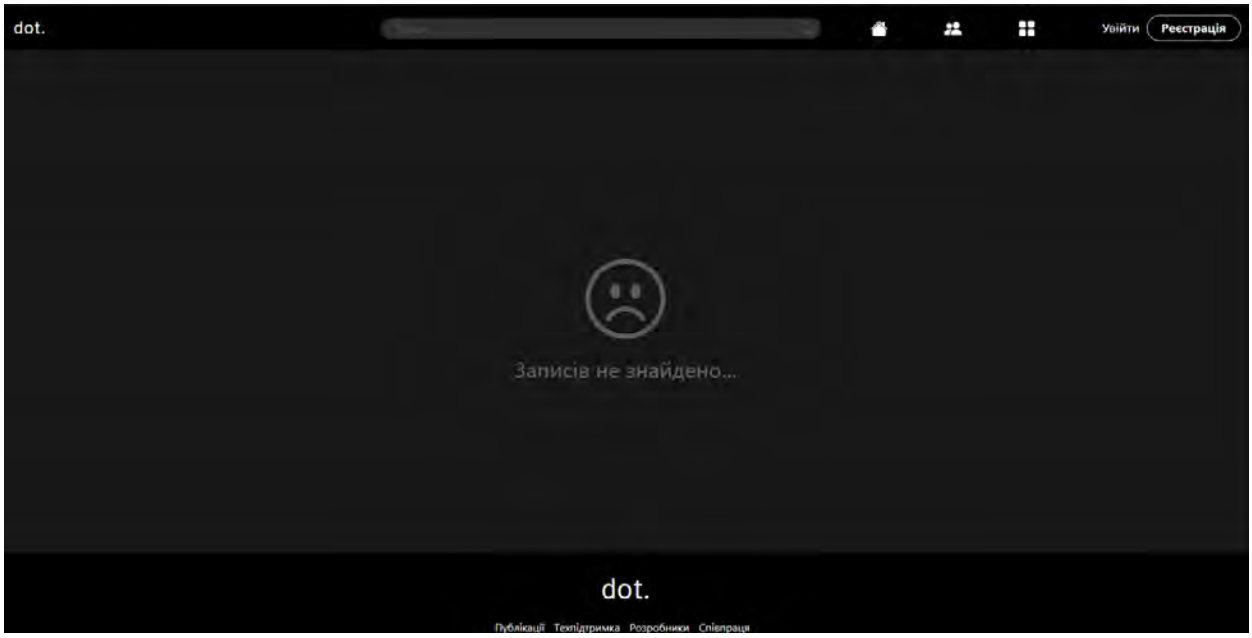

Рисунок 3.34 – Приклад відсутності записів

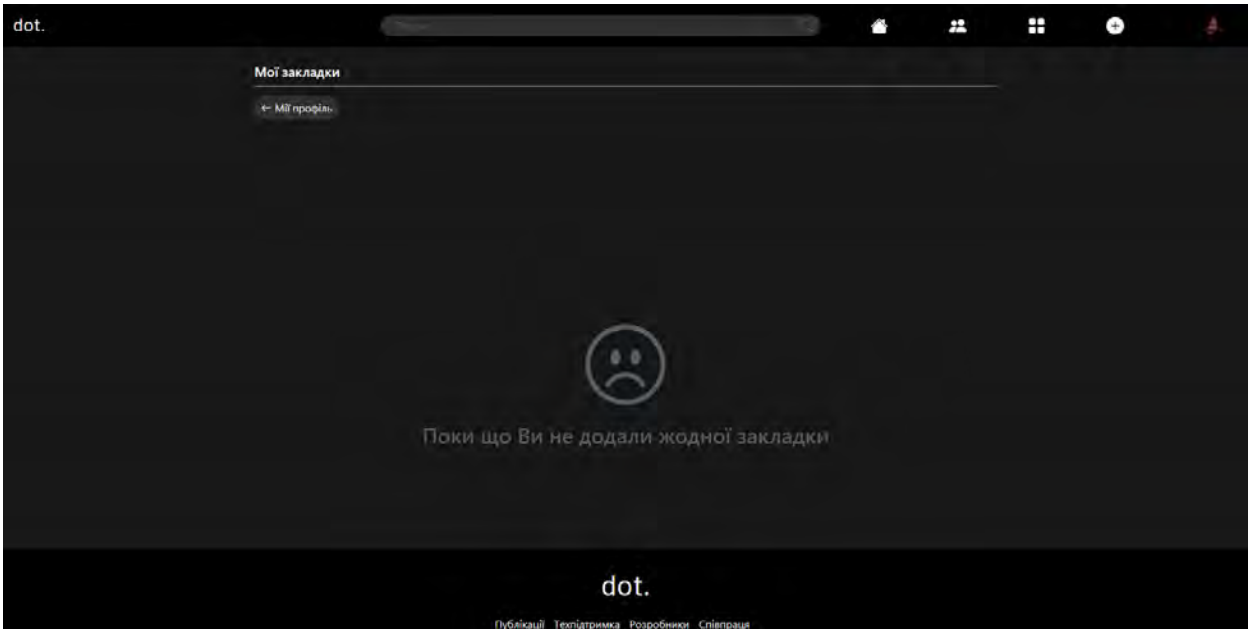

Рисунок 3.35 – Приклад відсутності закладок

3.6.12 Підвал

Підвал – нижня частина вебсторінки, що зазвичай містить інформацію про авторські права, зв'язок з власником сайту, посилання на додаткові ресурси, посилання на соціальні мережі та інші корисні ресурси. Його зображення можна побачити на рис. 3.36.

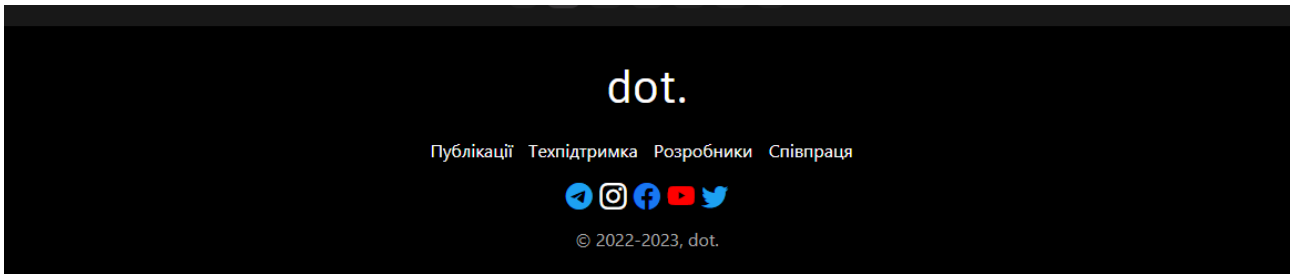

Рисунок 3.36 – Підвал сайту

3.7 Висновки за розділом

У цьому розділі було розглянуто низку важливих аспектів, що стосуються процесу розробки програмного забезпечення для вебсайту. Він містив загальні відомості про розробку, розглянуто файлову структуру проєкту, а також додаткову логіку сайту, яка допомагає покращити його функціональність та зручність використання.

Детально була розглянута реалізація механізму запиту на сервер, що дозволяє користувачам взаємодіяти з вебсайтом та виконувати різні дії.

Нарешті, була показана Візуальна складова клієнтської частини, яка дозволяє користувачам легко взаємодіяти з вебсайтом та зручно користуватися його функціоналом.

4 КЕРІВНИЦТВО ПРОГРАМІСТА

4.1 Призначення та умови використання

Фронт-енд частина IT-форуму є необхідною та незамінною складовою цього проєкту, що виконує цілу низку значущих функцій. Вона спрямована на посилення комфорту користувачів, поліпшення їх користувацького досвіду та забезпечення максимально зручної взаємодії з форумом.

Для успішної реалізації клієнтської частини вимагаються спеціальні технічні характеристики, що гарантують його ефективну роботу:

– 1 Гб Оперативної пам’яті;

– більше 2Гб вільного простору на жорсткому диску;

– процесор на 2ГГц;

– відеокарта з підтримкою WebGL і OpenGL;

– мережевий адаптер зі швидкістю передачі даних не менше 10 Мбіт/с.

IT-форум відрізняється відмінною структурованістю, що дозволяє забезпечити користувачам багато переваг і зручностей під час використання сайту.

4.2 Встановлення та початок роботи

Для розробки front-end частини варто використовувати середовище розробки JetBrains PHPStorm. Це інтегроване середовище надає програмістам широкий набір інструментів та функціональності, спеціально створених для розробки вебзастосунків. Воно підтримує автоматичне завершення коду, перевірку синтаксису, рефакторинг, налагодження та інші корисні функції, що сприяють покращенню продуктивності розробника.

Для встановлення проєкту необхідно розпакувати архів «it-forum», який містить усі необхідні файли, та інтегрувати його в файлову систему.

Повноцінна робота застосунку вимагає виконання першочергових налаштувань. По-перше, встановлюється Composer командою «composer install», оскільки цей інструмент забезпечує ефективне керування залежностями в PHP-проєктах.

По-друге, важливо застосувати «npm i» для автоматичного оновлення модулів проєкту до потрібних версій. Ця команда забезпечить, що всі необхідні залежності будуть належним чином встановлені та актуалізовані.

Для запуску сайту на вбудованому сервері використовується команда «php artisan serve». При виконанні цієї команди, сервер стартує і починає слухати вказаний порт на локальній системі. Після запуску вебзастосунок, розроблений з використанням Laravel, стає доступним для перегляду та взаємодії через веббраузер.

У разі потреби установки додаткових графічних модулів, таких як модулі авторизації та реєстрації, необхідно виконати певну послідовність команд. Спочатку, після встановлення цих модулів, потрібно виконати команду «npm install» у терміналі, а також команду «npm run dev» для компіляції JS-коду. Ці команди можна поєднати в одну – «(npm install) -and (npm run dev)». Додатково, виконання команди «npm run build» у другому терміналі сприятиме компіляції та збірці результуючого JS-коду.

4.3 Розташування файлів у структурі проєкту

Для подальшої розробки проєкту необхідно дотримуватися чітких правил щодо розташування файлів, з метою забезпечення належного функціонування клієнтської частини.

Ці правила визначають чіткий шлях для розміщення різноманітних складових клієнтської частини, таких як HTML-шаблони, стилі CSS, скрипти JavaScript та мультимедійні ресурси.

Структура проєкту передбачає, що всі сторінки та форми повинні бути розміщені у директорії «resources». Ця директорія виконує роль основного

контейнера для цих елементів і допомагає упорядковано організувати їх. Кожна сторінка чи форма може бути розміщена у відповідній папці в межах цієї директорії в залежності від їхньої функціональності або категорії.

Одночасно, стилі, логіка та зображення повинні бути розміщені у відповідних папках директорії «it-forum/public/src». Ця директорія виступає як основне місце для збереження ресурсів, які використовуються в HTML-шаблонах. Розміщення цих ресурсів у відповідних папках забезпечує правильну роботу механізму, оскільки використовуються вони з саме цієї директорії. Важливо врахувати, що в разі розташування цих ресурсів у будь-якому іншому місці, механізм взаємодії може не працювати належним чином.

4.4 Підключення ресурсів

У проєкті передбачено два типи підключення додаткових ресурсів: глобального та локального характеру. Глобальні ресурси, такі як набір інструментів Bootstrap, призначені для використання у всьому проєкті, тому їх підключення має бути здійснене таким чином, щоб кожна сторінка мала доступ до цих ресурсів. Ця процедура здійснюється у файлі «layout.blade.php», який зображений на рис. 4.1.

```
<!-- Bootstrap -->
<script src="https://code.jquery.com/jquery-3.5.1.min.js" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha38
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js" integrity="sha384-wfSDF2E50Y2D1uUd

<link href="https://cdn.jsdelivr.net/npm/summernote@0.8.18/dist/summernote-bs4.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/summernote@0.8.18/dist/summernote-bs4.min.js"></script>
<!-- Bootstrap Icons -->
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.10.3/font/bootstrap-icons.css">
<!--------CSS Styles-------->
<!-- Main styles for site pages-->
<link rel="stylesheet" href="/src/css/mainStyles.css">
<!-- Header CSS -->
<link rel="stylesheet" href="/src/css/navBar.css">
```

Рисунок 4.1 – Глобальне підключення ресурсів

Локальне підключення ресурсів передбачене для використання окремими сторінками, кожна з яких має свій власний файл стилів. Це зроблено з метою запобігання повторного завантаження цих файлів кожного разу при відвідуванні сайту, тим самим оптимізуючи швидкість завантаження сайту. Цей підхід дозволяє зменшити обсяг передачі даних та поліпшити продуктивність системи завантаження.

Приклад локального підключення зображено на рис. 4.2.

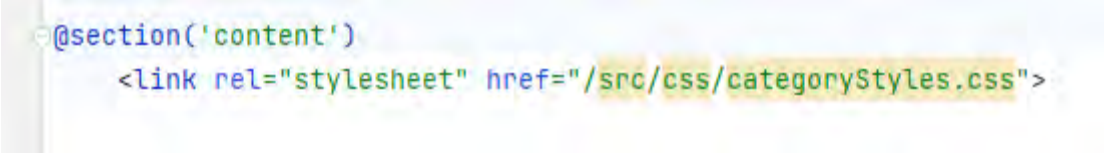

Рисунок 4.2 – Локальне підключення ресурсів

4.5 Висновки за розділом

У розділі було ретельно розглянуто ключові принципи, що стосуються подальшої розробки Front-end частини IT-форуму. В ньому були обговорені важливі технічні вимоги та рекомендоване програмне забезпечення, які використовувалися та рекомендувалися розробникам для ефективної праці над проєктом.

Також був наданий опис процедури запуску проєкту на локальному сервері, що дозволить розробникам відтворити та тестувати Front-end частину у контрольованому середовищі. Це сприятиме ефективній взаємодії з іншими складовими системи та вирішенню потенційних проблем.

У розділі були розглянуті різні типи підключень ресурсів і наведені відповідні способи їх підключення. Визначено, що існують два типи підключень: глобальне та локальне.

Додатково, були викладені чіткі правила та рекомендації щодо розміщення файлів, пов'язаних з клієнтською частиною, щоб забезпечити належну роботу вебзастосунку.

5 КЕРІВНИЦТВО КОРИСТУВАЧА

5.1 Обліковий запис користувача

Почати роботу з сайтом варто зі створення облікового запису. Це надасть деякі привілеї, яких немає у звичайних гостей. Для того потрібно з будь-якої сторінки через панель навігації натиснути на кнопку «Реєстрація», що зображено на рис. 5.1.

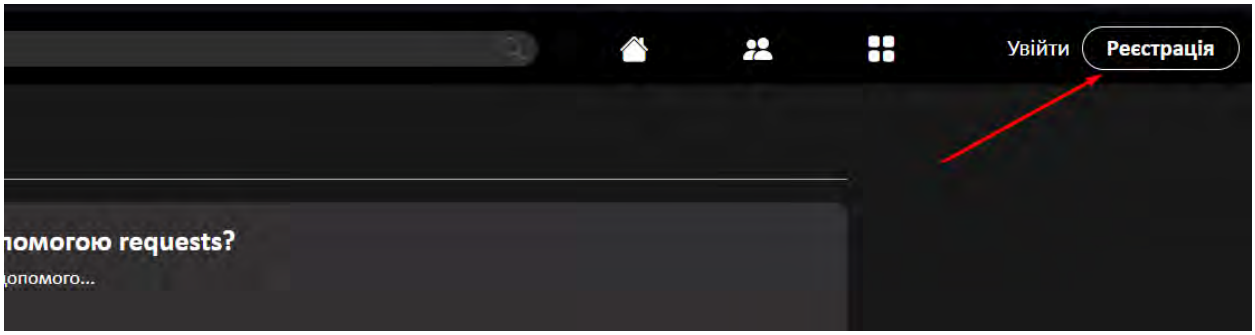

Рисунок 5.1 – Кнопка реєстрації на панелі навігації

Нас направить на сторінку реєстрації, де потрібно буде надати про себе деякі дані. Приклад даних наведено на рис. 5.2.

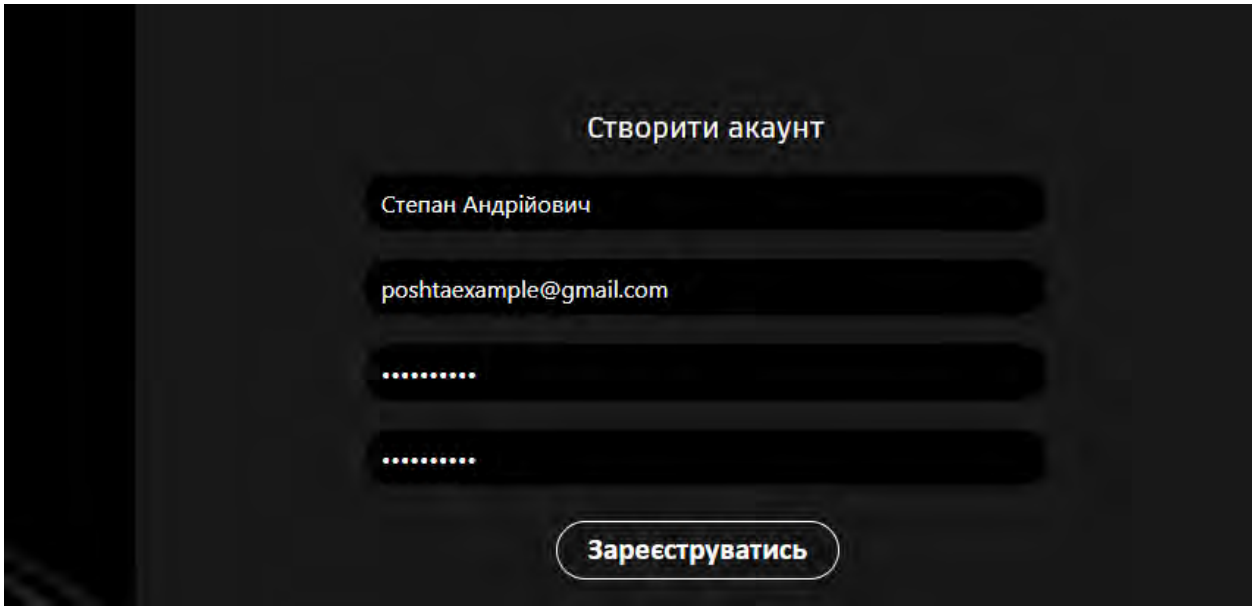

Рисунок 5.2 – Приклад даних для реєстрації

Якщо дані було введено вірно, нас перенесе на головну сторінку.

Зареєстровані користувачі бачуть замість кнопок «Входу» та «Реєстрації» свою фотокартку, якщо звісно він її додавав. В іншому випадку, буде розміщено стандартне зображення (див. рис. 5.3).

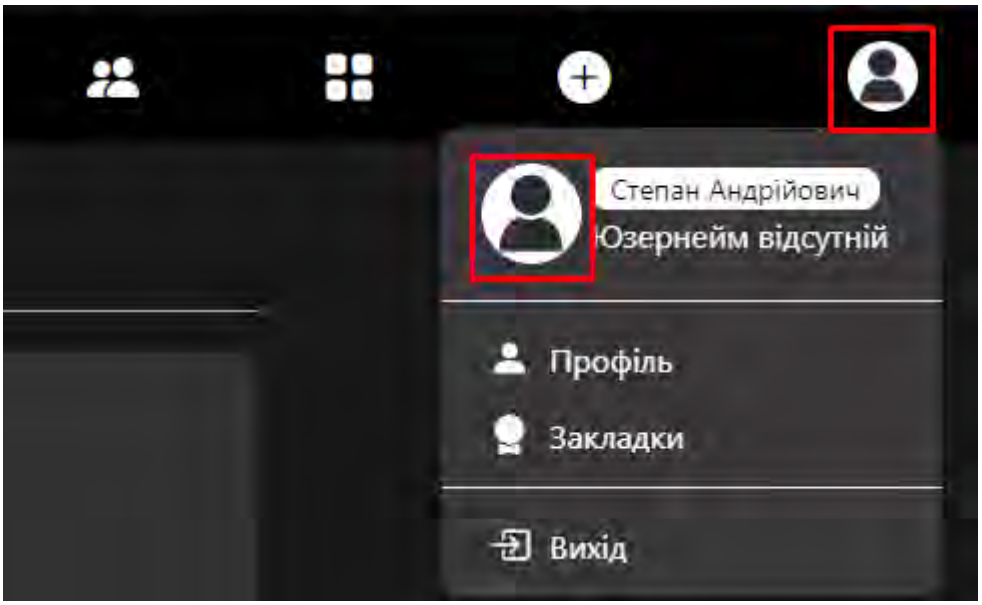

Рисунок 5.3 – Стандартне зображення користувача

Після успішної реєстрації, основним бажанням користувача є персоналізація його профілю шляхом додавання додаткової інформації про себе, встановлення псевдоніма та зміни фотографії профілю. Ці дії можна здійснити шляхом використання зручного і попередньо налаштованого інтерфейсу, доступного через сторінку профілю.

На рис. 5.4 зображено вигляд профілю користувача, де чітко виділена кнопка, що дозволяє перейти до сторінки редагування профілю. Ця кнопка надає зручний шлях для здійснення необхідних змін та внесення оновлень до особистої інформації користувача.

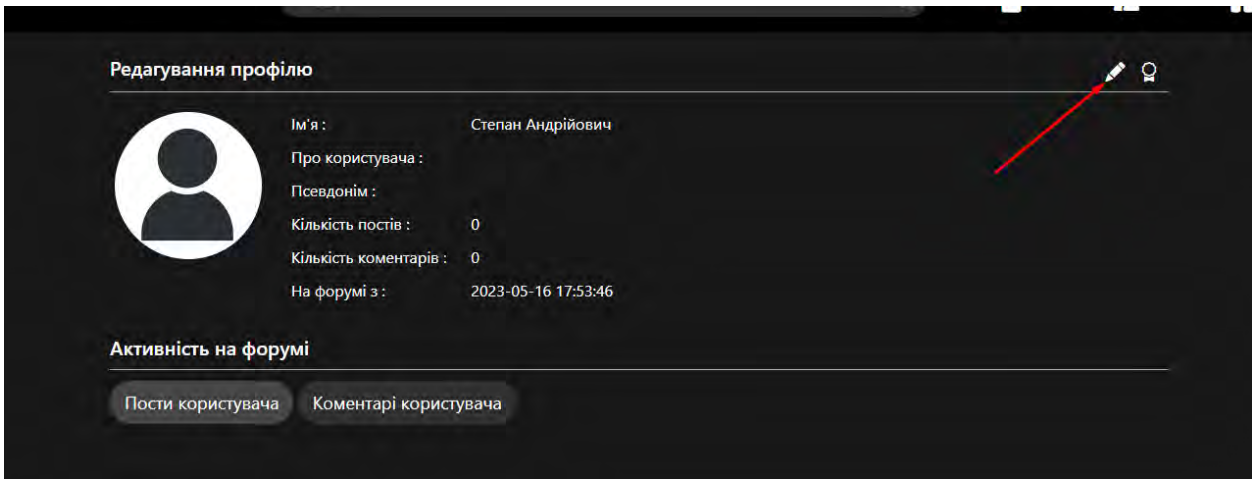

Рисунок 5.4 – Місцезнаходження кнопки переходу до редагування

На рис. 5.5 представлений інтерфейс, який включає 3 текстові поля, призначені для зміни особистих даних користувача. Додатково, інтерфейс містить поле для зміни зображення профілю. Це поле дозволяє користувачеві завантажити нове зображення, яке буде використовуватися як фотографія профілю.

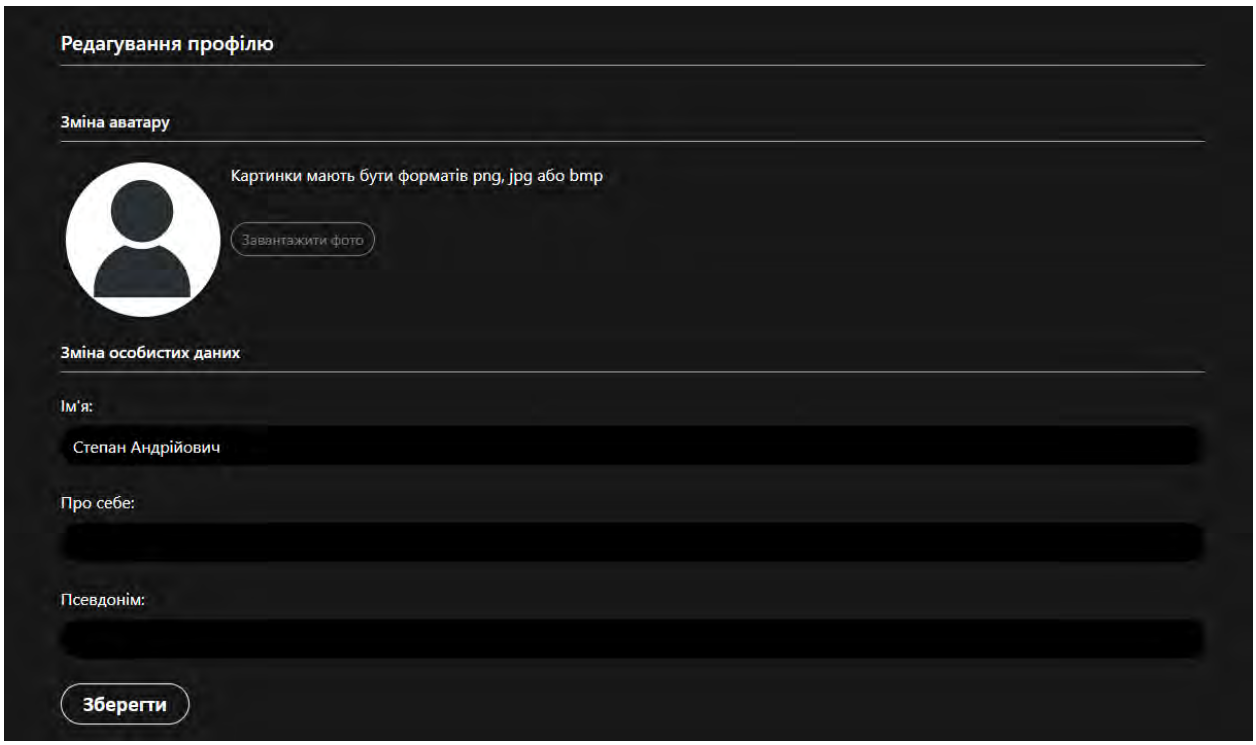

Рисунок 5.5 – Інтерфейс редагування даних

При завантаженні фотографії профілю на сайт, важливо враховувати деякі обмеження, щоб забезпечити правильну роботу та ефективність системи. Сайт встановлює певні вимоги щодо формату та розміру завантажуваних зображень.

По-перше, зображення повинно мати дозволений формат. Це може бути формат JPEG, PNG або інший, який підтримується сайтом. Якщо фото має непідтримуваний формат, сайт не дозволить його завантажити та зберегти, і користувачу буде надано відповідне повідомлення про помилку.

По-друге, розмір зображення також має бути в межах, прийнятних для завантаження. Якщо фото є занадто великим, це може призвести до перевищення обмежень сервера або сповільнення процесу завантаження. Тому, сайт може встановити обмеження на максимальний розмір файлу для завантаження фотографії профілю.

На сторінці профілю доступна функція перегляду його активності, що надає цінну інформацію про те, як користувач взаємодіє з IT-форумом. Ця функція включає два важливих аспекти: перегляд створених тем і перегляд написаних коментарів. Їх зображено на рис. 5.6

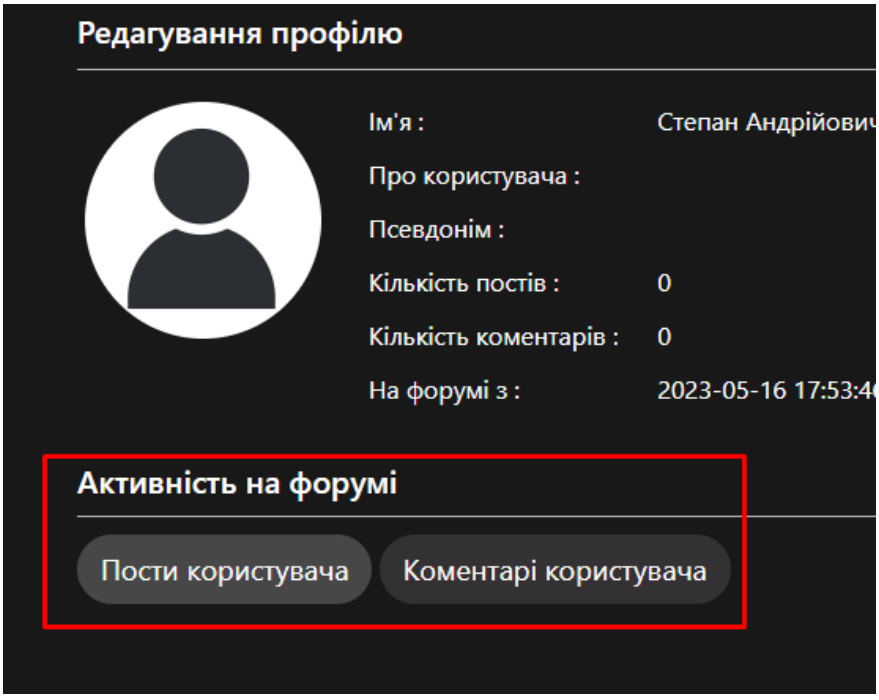

Рисунок 5.6 – Створені пости користувача

У сторінці профілю користувача є можливість переглядати збережені закладки. Ця функція дозволяє користувачеві відзначати цікаві його матеріали або теми і зручно зберігати їх для майбутнього доступу.

Однак, спочатку, коли користувач ще не має жодної збереженої закладки, відповідний розділ буде порожнім. Це видно на рис. 5.7, де вказано, що відсутні будь-які збережені закладки.

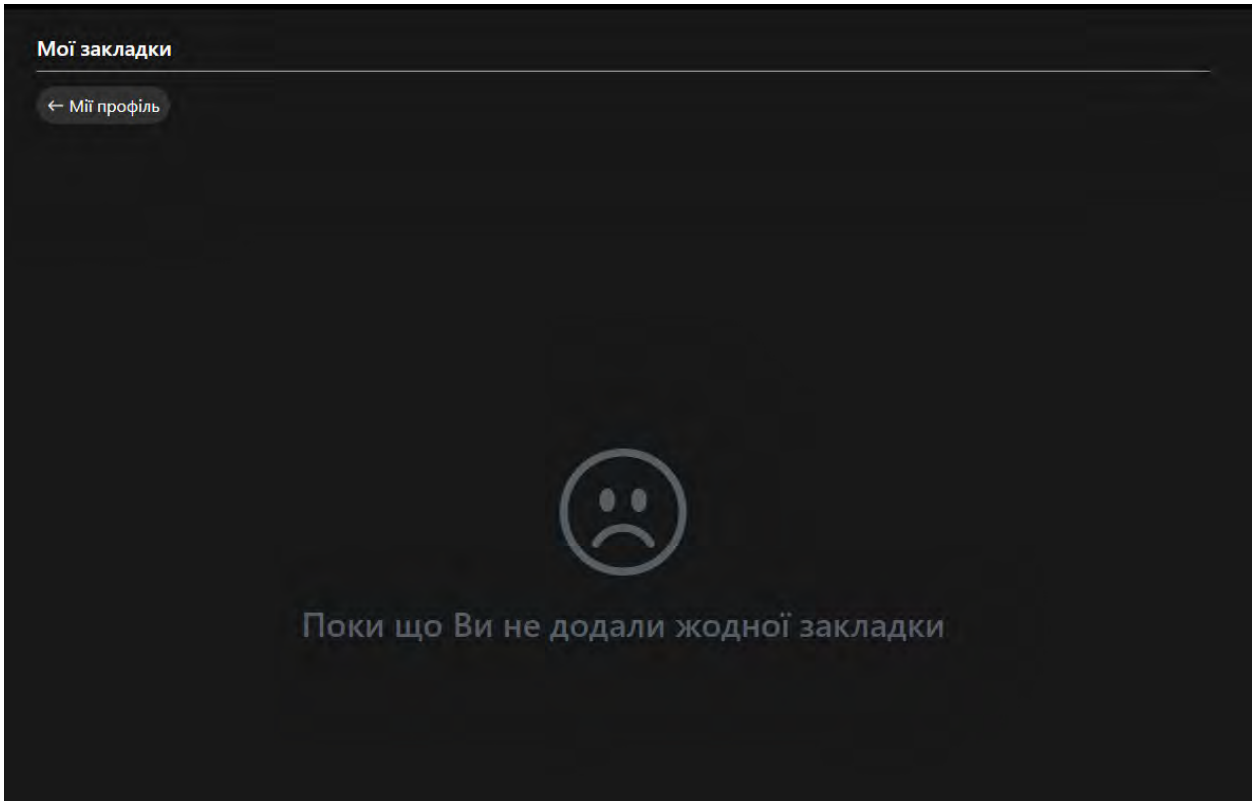

Рисунок 5.7 – Відсутність закладок

5.2 Створення теми

Якщо користувачеві потрібно вирішити якусь проблему або задати питання, він швидше за все захоче створити нову тему на форумі. Для цього існує вкладка «Створення публікації», розташована на панелі навігації, як показано на рис. 5.8.

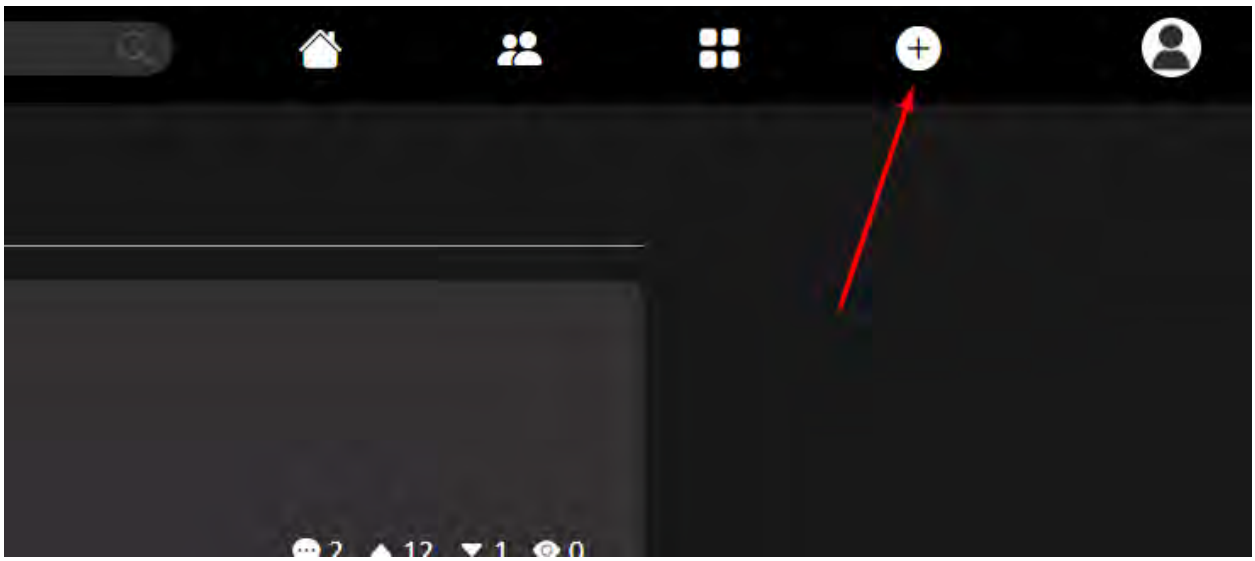

Рисунок 5.8 – Вкладинка «Створення публікації»

Процес створення нової теми розпочинається з введення заголовку, який служить коротким описом проблеми чи запитання. Ця частина має велике значення, оскільки інші користувачі частіше звертають увагу на заголовки і вони допомагають привернути увагу до публікації. Після введення заголовку користувач переходить до наступного кроку.

Наступним етапом є вибір категорій, до яких може бути призначена нова тема. Користувач може обрати одну або декілька категорій, причому рекомендується вибирати ті, які найкраще відповідають темі публікації. Це допомагає іншим користувачам знаходити теми, що їх цікавлять, і забезпечує більш зорієнтований обмін інформацією.

Останнім кроком є написання опису теми. Користувач може вільно виражати свої думки та модифікувати вміст опису, використовуючи доступний графічний редактор. Опис дозволяє більш детально пояснити суть проблеми або запитання та надає можливість користувачам краще розуміти контекст теми.

Увесь цей функціонал зображений на рис. 5.9.

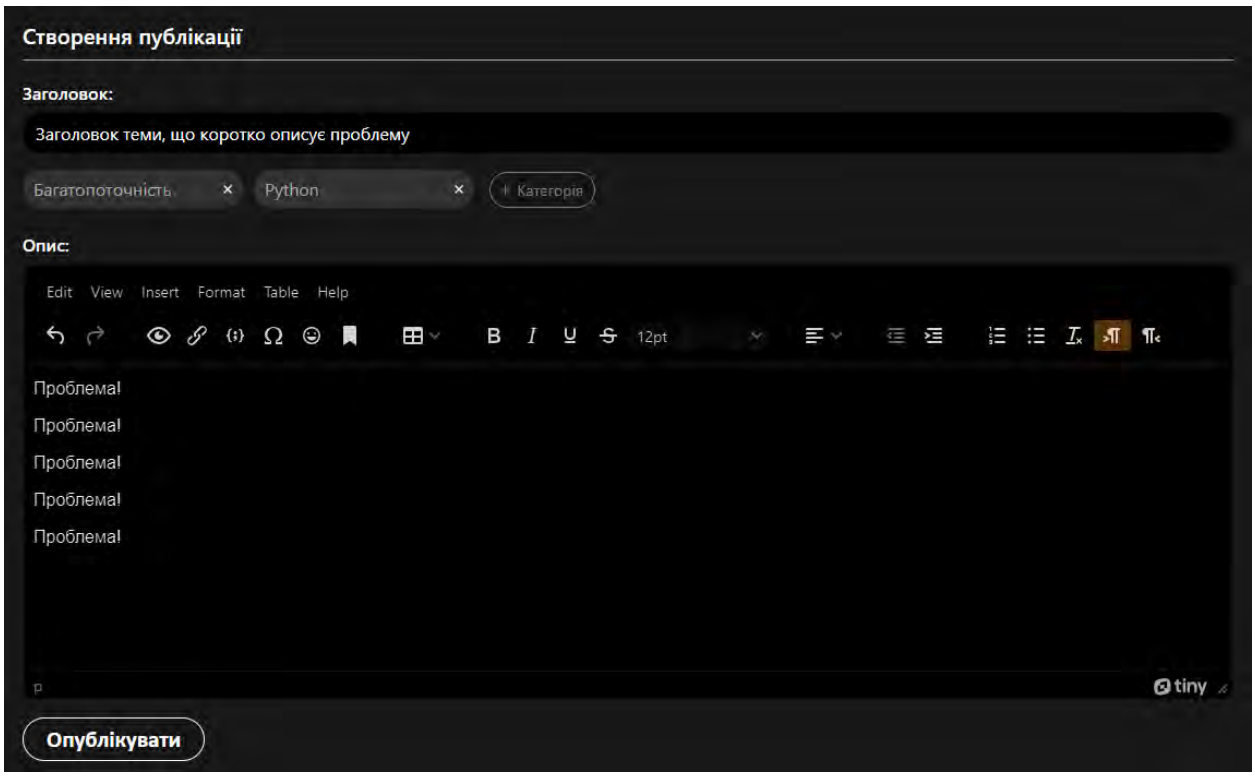

Рисунок 5.9 – Приклад оформлення теми в редакторі

5.3 Взаємодія з постами

На форумі будь-який користувач має можливість взаємодіяти з постами, але доступні функції взаємодії варіюються в залежності від ролі користувача. Головна сторінка форуму, відображає список доступних публікацій.

Автори постів мають додаткові можливості для управління своїми постами. Вони можуть редагувати вміст своїх постів та видаляти їх. Це здійснюється за допомогою випадаючого списку дій, як показано на рис. 5.10. Цей інтерфейс дозволяє авторам зручно керувати своїми постами, забезпечуючи можливість внесення змін або видалення відповідно до їх потреб.

Ці функції взаємодії надають користувачам більшу контроль над їх постами та дозволяють змінювати їх відповідно до ситуації або своїх уявлень. Така гнучкість дозволяє краще управляти власним вмістом та забезпечує покращену взаємодію на форумі.

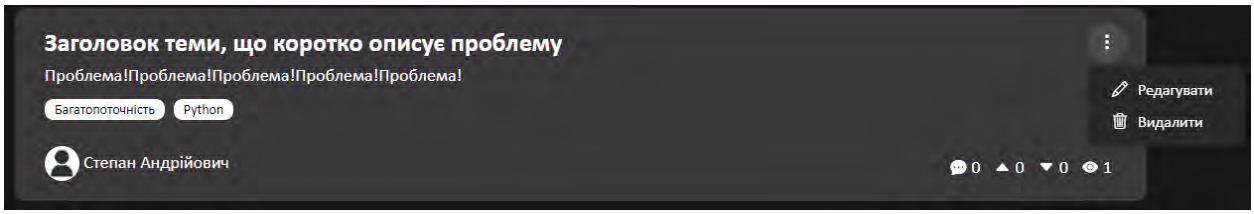

Рисунок 5.10 – Випадаючий список

Перейшовши по темі, нам відкривається її сторінка, приклад якої зображено на рис. 5.11, де також є список дій.

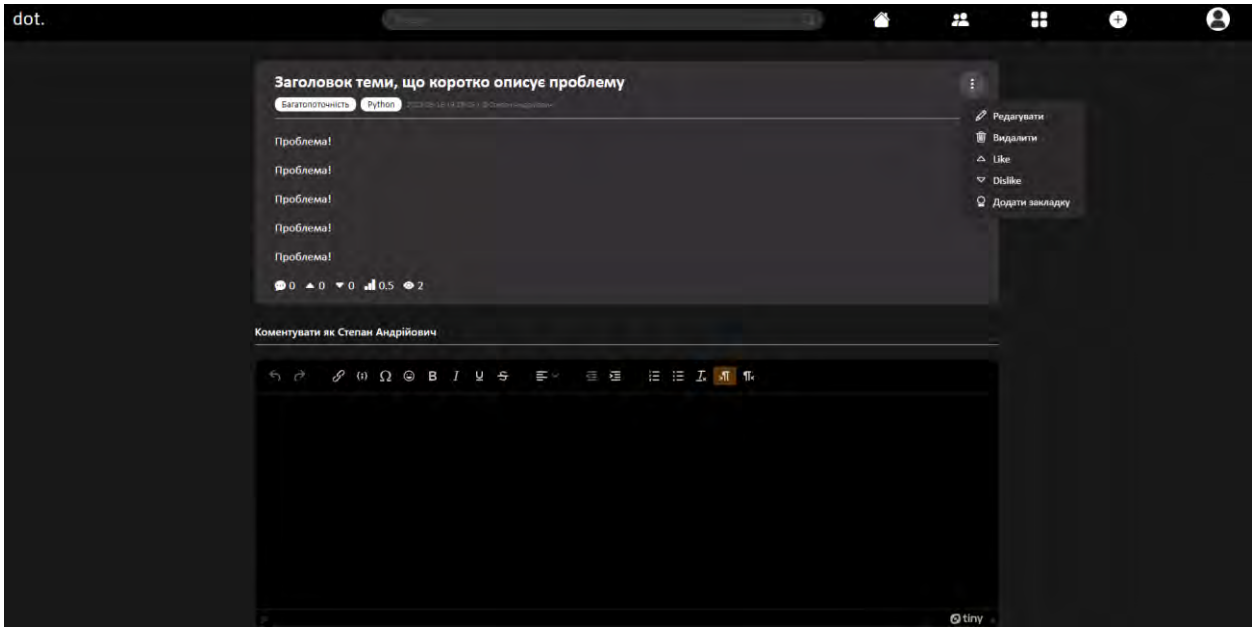

Рисунок 5.11 – Сторінка теми

Як видно з рисунку вище, в темі ми можемо побачити її рейтинг, кількість відповідей та кількість переглядів. Крім того, є можливість оцінювання теми, показана кількість осіб, яким сподобалося або не сподобалося це.

Оцінювання та додавання до закладок здійснюються за допомогою випадаючого списку дій. Цей інтерфейс дозволяє користувачам вибрати відповідну опцію, яка відповідає їхнім потребам. Наприклад, вони можуть оцінити тему позитивно або негативно, а також додати її до своїх закладок для подальшого зручного доступу.

5.4 Список категорій

Якщо користувачу потрібно знайти теми певної категорії, або подивитися, які категорії обговорюються на форумі, він може зробити це на відповідній сторінці, до якої можна потрапити через панель навігації, що можна побачити на рис. 5.12.

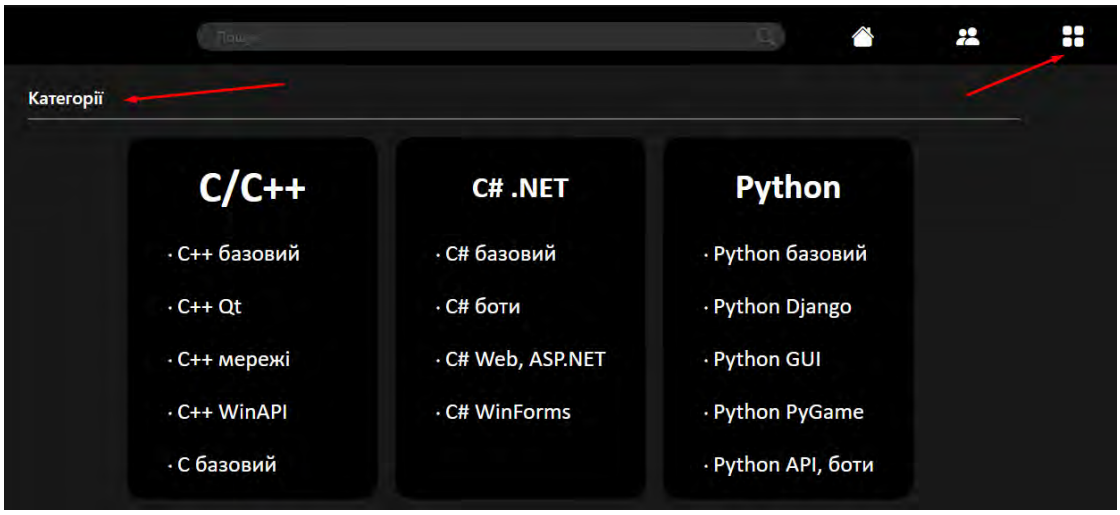

Рисунок 5.12 – Список категорій

При натисканні на будь-яку з категорій, користувачеві буде відображено список наявних тем, який можна додатково фільтрувати за певними категоріями. Це надає зручну можливість користувачам зорієнтуватись і знайти теми, що цікавлять їх, на основі їхніх вподобань та потреб. На рис. 5.13 демонструється інтерфейс зі списком тем та функцією фільтрації.

Цей функціонал сприяє зручному навігаційному досвіду користувачів, дозволяючи їм швидко знайти теми, що відповідають їхнім інтересам та специфічним потребам.

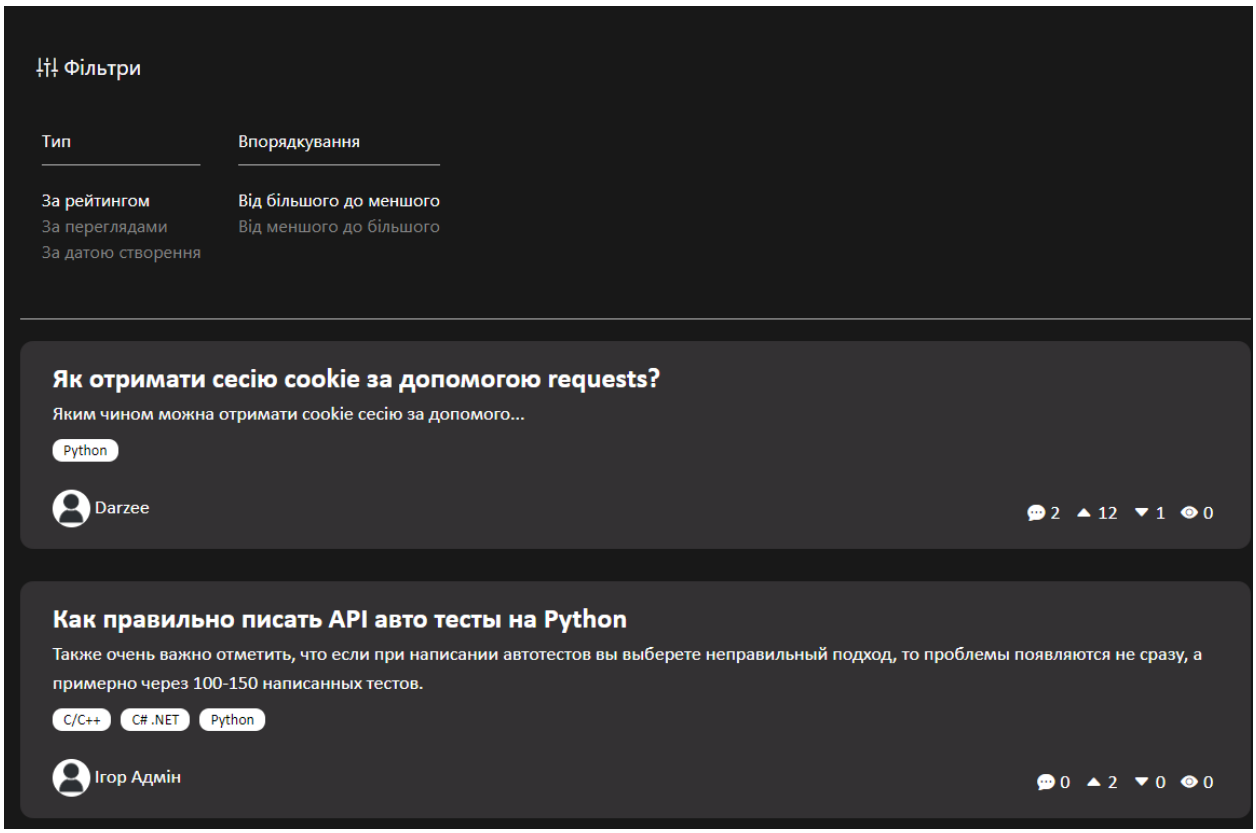

Рисунок 5.13 – Список відсортованих постів за категорією

5.5 Висновки за розділом

В цьому розділі було детально розглянуто основні можливості, які надаються користувачам у взаємодії з front-end частиною сайту. Було наведено приклади оформлення постів, які демонструють різноманітність стилів та форматувань, що доступні для виразу думок та спілкування на форумі.

Крім того, було наголошено на важливості дотримання правил користування форумом. Ці правила створені з метою забезпечення зручності та безпеки для всіх учасників спільноти. Вони включають обмеження щодо контенту, який можна публікувати, вимоги до поведінки та комунікації між користувачами.

ВИСНОВКИ

Під час розробки дипломного проєкту було розроблено front-end частину IT-форуму, використовуючи сучасні методи розробки. В рамках аналізу предметної області була детально розглянута технічна складова форумів, а також обґрунтовано важливість розробки інтерфейсу користувача.

Для реалізації проєкту були використані мови програмування HTML, CSS та JavaScript. Основною платформою розробки став фреймворк Laravel, який забезпечив потужні функціональні можливості та швидкість розробки. Додатково були використані інструменти, такі як Bootstrap для забезпечення швидкого та зручного оформлення елементів інтерфейсу, а також TinyMCE для інтеграції текстового редактора у форми вводу.

Ця комбінація технологій та інструментів дозволила створити функціональну та естетичну front-end частину IT-форуму, забезпечуючи зручний та привабливий досвід користувачів.

IT-форум – це онлайн-платформа, створена з метою об'єднання спільноти IT-професіоналів, студентів, ентузіастів та всіх, хто цікавиться галуззю інформаційних технологій. Цей проєкт дозволяє користувачам отримувати та обмінюватися цінними знаннями, досвідом та ідеями з усього світу.

Користувачі можуть розміщувати запитання, ділитися своїми проектами, пропонувати рішення для проблем, а також обговорювати актуальні теми та тенденції у світі IT. Це стимулює активний обмін інформацією та сприяє вирішенню проблем, з якими стикаються учасники.

Розроблений проект є незамінною платформою для всіх, хто бажає поглибити свої знання та розширити професійну мережу контактів. Він допомагає вирішувати проблеми, спілкуватися з однодумцями, навчатися від експертів та співпрацювати над проектами.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Парненко В.С. Веб-дизайн як фундамент сучасного Віртуального середовища / В.С. Парненко // Праці Одеського політехнічного університету. – 2013. – №2. – С. 247–249.

2. Методичні вказівки до проведення самостійної роботи та практичних занять з дисципліни “Архітектури та технології WEB-служб” для студентів напрямму 6.050103 “Програмна інженерія” денної форми навчання / С.О. Степаненко. – Запоріжжя: ЗНТУ, 2016. – 23 с.

3. A. H. Biriyai Online Discussion Forum: A Tool for Effective StudentTeacher Interaction [Text] / A. H. Biriyai // International Journal of Applied Science-Research and Review. – 2014. – Vol. 1, №3. – P. 2–5.

4. Morzy, M. Evolution of Online Forum Communities / M. Morzy // The Influence of Technology on Social Network Analysis and Mining. Lecture Notes in Social Networks. – 2013. – Vol 6. – P. 615–630. – https://doi.org/10.1007/978-3-7091-1346-2_27

5. Романюк О.Н. Веб-дизайн і комп’ютерна графіка / Романюк О.Н., Катєльніков Д.І., Косовець О. П. – Вінниця: ВНТУ, 2007. – 144 с.

6. How People Read Online: New and Old Findings [Electronic resource]. – Access mode: https://www.nngroup.com/articles/how-people-read-online/

7. F-Shaped Pattern For Reading Web Content [Electronic resource]. – Access mode: https://www.nngroup.com/articles/f-shaped-pattern-reading-web-content-discovered/

8. Веб-технології та веб-дизайн / О. Г. Трофименко, О. Б. Козін, О. В. Задерейко, О. Є. Плачінда. – Одеса : Фенікс, 2019. – 284 с.

9. Lazy Load [Electronic resource]. – Access mode: https://martinfowler.com/eaaCatalog/lazyLoad.html

10. Steve Souders High Performance Web Sites: Essential Knowledge for Front-End Engineers / Steve Souders // O'Reilly Media. – 2007. – 170 p.

11. Reddit programming [Electronic resource]. – Access mode: https://www.reddit.com/r/programming/

12. GitHub Community Forum [Electronic resource]. – Access mode: https://github.com/orgs/community/discussions

13. Stack Overflow [Electronic resource]. – Access mode: https://stackoverflow.com

14. Глоба Л.С. Розробка інформаційних ресурсів та систем / Л.С. Глоба. – К: КПІ, 2013. – 378 с.

15. Молчанов В. П. Основи проектування WEB-видань / В. П. Молчанов. – Х: ХНЕУ ім. С. Кузнеця, 2017. – 159 с.

16. Laravel [Electronic resource]. – Access mode: https://laravel.com/docs/10.x/routing

17. Bootstrap 4 [Electronic resource]. – Access mode: https://getbootstrap.com/docs/4.0/getting-started/introduction/

18. TinyMCE 6 [Electronic resource]. – Access mode: https://www.tiny.cloud/docs/tinymce/6/

19. Composer Documentation [Electronic resource]. – Access mode: https://getcomposer.org/doc/

20. Git Documentation [Electronic resource]. – Access mode: https://git-scm.com/doc

ДОДАТОК А

Технічне завдання

А.1 Підстава для розробки

Підставою для розробки стало завдання на дипломну роботу з теми: «Програмна реалізація front-end частини ІТ-форуму», затверджене наказом №87 від 4 квітня 2023 року по Національному університету «Запорізька політехніка».

А.2 Призначення розробки

Призначення розробки – створення програмного продукту, який забезпечує користувачам доступ до front-end частини ІТ-форуму, включаючи відображення тем, повідомлень, взаємодію з іншими користувачами та інші функції, необхідні для повноцінного використання форуму. Також, створення індивідуального дизайну та функціоналу відповідно до вимог та потреб користувачів форуму.

А.3 Вимоги до розроблювального програмного забезпечення

А.3.1 Вимоги до функціональних характеристик

Програмна реалізація front-end частини має забезпечувати:

– ефективний, зручний та привабливий дизайн інтерфейсу;

– оптимізованість програмного забезпечення;

– реєстрування нового користувача;

– вхід до існуючого облікового запису;

– перехід по головним сторінкам сайту через навігаційне поле;

– створення та перегляд публікацій;

– оцінювання публікацій;

– редагування та видалення існуючих тем;

– створення нових і перегляд існуючих категорій;

– перегляд списку існуючих користувачів;

– перегляд профілів користувачів;

– редагування профілю користувача;

– пошук тем по ключовим словам у рядку пошуку;

– фільтрація новин;

– створення та перегляд закладок;

– пагінація сторінок.

А.3.2 Вимоги до надійності

Програмна реалізація front-end частини ІТ-форуму має забезпечувати коректну роботу завдяки:

– стійкості до навантаження;

– мінімізації помилок;

– безпеці даних;

– зручності користування;

– оптимізації.

А.3.3 Вимоги до експлуатації

Для забезпечення оптимальної роботи програмного забезпечення ІТ-форуму, важливо використовувати сучасні веб-браузери, які підтримують останні стандарти веб-розробки. Це гарантує сумісність з новітніми технологіями, функціями та покращеннями, що забезпечуються оновленнями фреймворків та бібліотек.

Оновлені версії веб-браузерів надають доступ до передових можливостей HTML, CSS та JavaScript, що дозволяють розробникам створювати більш ефективний та інтерактивний інтерфейс користувача. Вони підтримують сучасні технології, такі як анімація, графічні ефекти, веб-

компоненти та багато іншого, що сприяє покращенню візуального враження та користувацького досвіду.

А.3.4 Вимоги до складу та параметрів технічний засобів

Комфортне використання ІТ-форуму забезпечується за набором таких характеристик:

– 1 Гб Оперативної пам’яті;

– більше 2Гб вільного простору на жорсткому диску;

– процесор на 2ГГц;

– відеокарта з підтримкою WebGL і OpenGL;

– мережевий адаптер зі швидкістю передачі даних не менше 10 Мбіт/с.

А.3.5 Вимоги до маркування й пакування

Програмне забезпечення може пакуватися на будь-якому з’ємному накопичувачі. На пакуванні має бути зазначена назва проекту – «Програмна реалізація front-end частини ІТ-форуму».

ДОДАТОК Б

Текст програми

Б.1 Текст файлу views/home.blade.php

```
@extends('layouts.app')

@section('content')
<div class="container">
   <div class="row justify-content-center">
      <div class="col-md-8">
         <div class="card">
            <div class="card-header">{{ __('Dashboard') }}</div>

            <div class="card-body">
               @if (session('status'))
                  <div class="alert alert-success" role="alert">
                     {{ session('status') }}
                  </div>
               @endif

               {{ __('You are logged in!') }}
            </div>
         </div>
      </div>
   </div>
</div>
@endsection
```

Б.2 Текст файлу views/index.blade.php

```
@extends('layouts.layout')

@section('content')
   @if(count($posts))


      <div class="row">
         <script>
            window.onload = function () {
               setSortTypeText('{{$filters['sortingType']}}');
               setSortGrowthText('{{$filters['sortingGrowth']}}');
            }
         </script>
         <script type="text/javascript" src="/src/js/logicPostsPage.js"></script>



         <!--     -->
         <div id="accordion">
            <div class="filter-box">
```

```
                    <div class="card-header filter-header" id="headingTwo">
                        <button class="filter-button" data-toggle="collapse" data-target="#collapseTwo" aria-expanded="false" aria-controls="collapseTwo">
                            <i class="bi bi-sliders2-vertical"></i> Фільтри
                        </button>
                    </div>
                    <div id="collapseTwo" class="collapse" aria-labelledby="headingTwo" data-parent="#accordion">
                        <div class="card-body">
                            <div class="filter-block">
                                <div>Тип</div>
                                <div class="line"></div>
                                <div><a class="filter-category" href="" id="type1" onclick="setSortingFilter(this, 'type', 'rating')">За рейтингом</a></div>
                                <div><a class="filter-category" href="" id="type2" onclick="setSortingFilter(this, 'type', 'amount_views')">За переглядами</a></div>
                                <div><a class="filter-category" href="" id="type3" onclick="setSortingFilter(this, 'type', 'created_at')">За датою створення</a></div>
                            </div>

                            <div class="filter-block">
                                <div>Впорядкування</div>
                                <div class="line"></div>
                                <div><a class="filter-category" href="" id="growth1" onclick="setSortingFilter(this, 'growth', 'desc')">Від більшого до меншого</a></div>
                                <div><a class="filter-category" href="" id="growth2" onclick="setSortingFilter(this, 'growth', 'asc')">Від меншого до більшого</a></div>
                            </div>

                        </div>
                    </div>
                </div>
            </div>
            <!--    -->
            <div class="line"></div>


        @foreach($posts as $post)
            <!-- Ті самі пости, що виводяться на сторінці-->
            <div class="posts-container">
                <div class="col">
                    @if (Auth::check())
                        @if(Auth::id() == $post->user['id'] || Auth::user()->hasRole('admin'))
                            <div class="dropdown threeDots">
                                <button class="threeDots-button" type="button" id="dropdownMenuButton" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                                    <div><i class="bi bi-three-dots-vertical"></i></div>
                                </button>
                                <div class="dropdown-menu dropdown-additions" aria-labelledby="dropdownMenuButton">
```

```
                                <a class="dropdown-item dropdown-item-additions" href="/posts/{{$post->id}}/edit"><i class="bi bi-pencil indent"></i>Редагувати</a>

                                <form action="/posts/{{$post->id}}" method="post">
                                    {{csrf_field()}}
                                    {!! method_field('delete') !!}
                                    <!-- Кнопка Видалення -->
                                    <button type="submit" class="dropdown-item dropdown-item-additions"><i class="bi bi-trash3 indent"></i>Видалити</button>
                                </form>

                            </div>
                        </div>
                    @endif
                @endif

                <a href="/posts/{{$post->id}}" class="title-font">{{ $post->title }}</a>
                <p>{{ $post->intro }}</p>

                    <div class="post-category-line">
                    @if(count($post->categories))
                        @foreach($post->categories as $category)
                            <div class="category"><a class="category-text" href="/categories/{{$category->id}}/posts">{{$category->name}}</a></div>
                        @endforeach
                    @endif
                    </div>

                <br>

                <div class="userName-box">

                    @if($post->user['photo'])
                        <a href="/users/{{$post->user['id']}}"><img src="{{$post->user->getPhotoPath($post->user['id']).$post->user['photo']}}" class="post-user-logo"></a>
                    @else
                        <a href="/users/{{$post->user['id']}}"><img src="../../src/img/default_user_photo.png" class="post-user-logo"></a>
                    @endif

                    <a class="post-list-author" href="/users/{{$post->user['id']}}">{{ $post->user['name'] }}</a>
                </div>
                <div class="info-box">
                    <!-- Comments -->
                    <span class="info-elem"><i class="bi bi-chat-dots-fill post-info"></i>{{ $post->amount_comments }}</span>
                    <!-- Likes -->
                    <span class="info-elem"><i class="bi bi-caret-up-fill post-info"></i>{{ $post->amount_likes }}</span>
                    <!-- Dislikes -->
```

```
                        <span class="info-elem"><i class="bi bi-caret-down-fill post-info"></i>{{ $post->amount_dislikes }}</span>
                        <!-- Views -->
                        <span class="info-elem"><i class="bi bi-eye-fill post-info"></i>{{ $post->amount_views }}</span>
                      </div>
                      <br>
                    </div>
                  </div>
                  <div class="w-100"></div>
                @endforeach
              </div>

              <div class="pagination-area">
                <div class="pagination-box">
                  {{ $posts->appends(['s' => request()->s,
                              'type' => request()->type,
                              'growth' => request()->growth
                              ])->links() }}
                </div>
              </div>

            @else
              @include('posts.notFoundPosts')
            @endif

          @endsection
```

Б.3 Текст файлу login.blade.php

```
          @extends('layouts.layout')

          @section('content')
          <script type="text/javascript" src="/src/js/logicDescription.js"></script>
          <script type="text/javascript" src="/src/js/logicAuth.js"></script>
          <link rel="stylesheet" href="/src/css/logRegStyles.css">

          <div class="row" id="rightBlock">

            @include('auth.outBackground')
            <div class="col-md-5 right-block">
              <div class="input-block" id="rightMenu">
                <form method="POST" action="{{ route('login') }}">
                  @csrf
                  <div>
                    <div class="login-margin"><h5>Увійти в акаунт</h5></div>
                    <div>
                      <input id="email" type="email" class="form-control input-line @error('email') is-invalid @enderror" name="email" placeholder="пошта" value="{{ old('email') }}"  required autocomplete="email" autofocus>
```

```
                    @error('email')
                    <span class="invalid-feedback" role="alert">
                            <strong>{{ $message }}</strong>
                        </span>
                    @enderror
                </div>

                <div>
                    <input id="password" type="password" class="form-control input-line @error('password') is-invalid @enderror" placeholder="пароль" name="password" required autocomplete="current-password">

                    @error('password')
                    <span class="invalid-feedback" role="alert">
                            <strong>{{ $message }}</strong>
                        </span>
                    @enderror
                </div>

                <br>
                <div class="interaction-margin">
                    <button type="submit" class="btn regist-button">
                        {{ __('Увійти') }}
                    </button>

                    @if (Route::has('password.request'))
                        <a class="btn btn-link text-light" href="{{ route('password.request') }}">
                            {{ __('Забув пароль?') }}
                        </a>
                    @endif

                </div>

                <div class="interaction-margin">
                    <div class="pt-3">
                    <input class="form-check-input" type="checkbox" name="remember" id="remember" {{ old('remember') ? 'checked' : '' }}>

                    <label class="form-check-label" for="remember">
                        {{ __("Запам'ятати мене") }}
                    </label>
                    </div>
                </div>
            </div>
        </form>
    </div>
</div>
</div>

@endsection
```

Б.4 Текст файлу register.blade.php

```
@extends('layouts.layout')

@section('content')
<script type="text/javascript" src="/src/js/logicDescription.js"></script>
<script type="text/javascript" src="/src/js/logicAuth.js"></script>
<link rel="stylesheet" href="/src/css/logRegStyles.css">

<div class="row" id="rightBlock">

   @include('auth.outBackground')
   <div class="col-md-5 right-block">

      <div class="input-block" id="rightMenu">

         <form method="POST" action="{{ route('register') }}">
            @csrf
            <div>
               <div class="login-margin"><h5>Створити акаунт</h5></div>
               <div>
                  <input id="name" type="text" class="form-control input-line @error('name') is-invalid @enderror" name="name" placeholder="ім'я" value="{{ old('name') }}" required autocomplete="name" autofocus>

                  @error('name')
                  <span class="invalid-feedback" role="alert">
                        <strong>{{ $message }}</strong>
                     </span>
                  @enderror
               </div>

               <div>
                  <input id="email" type="email" class="form-control input-line @error('email') is-invalid @enderror" name="email" placeholder="пошта" value="{{ old('email') }}" required autocomplete="email" autofocus>

                  @error('email')
                  <span class="invalid-feedback" role="alert">
                        <strong>{{ $message }}</strong>
                     </span>
                  @enderror
               </div>

               <div>
```

```
                    <input id="password" type="password" class="form-control input-line @error('password') is-invalid @enderror" placeholder="пароль" name="password" required autocomplete="current-password">

                    @error('password')
                    <span class="invalid-feedback" role="alert">
                            <strong>{{ $message }}</strong>
                        </span>
                    @enderror
                </div>

                <div>
                    <input id="password-confirm" type="password" class="form-control input-line" placeholder="повторіть пароль" name="password_confirmation" required autocomplete="new-password">
                </div>

                <br>
                <div class="interaction-margin">
                    <button type="submit" class="btn regist-button">
                        {{ __('Зареєструватись') }}
                    </button>
                </div>

            </div>

        </form>

    </div>

  </div>
</div>
@endsection
```

Б.5 Текст файлу outBackground.blade.php

```
<div class="col-md-7 left-block" id="authImg">
  <div class="dot-logo">dot.</div>
  <div class="dot-description">Simple Fast Modern</div>
</div>
```

Б.6 Текст файлу controlPanel.blade.php

```
@extends('layouts.layout')

@section('content')
  <link rel="stylesheet" href="/src/css/categoryStyles.css">
```

```
        <br>

        <div>
            <h5>Оберіть категорію</h5>
        </div>
        <div class="line"></div>

        <div class="cat-list">
            @if(count($categories))
            <div class="row">
                @foreach($categories as $category)
                    <a class="cat-brick" href="/categories/{{$category['id']}}/edit">{{$category['name']}}</a>

                    @foreach($category['heirs'] as $heir)
                        <a class="cat-brick" href="/categories/{{$heir->id}}/edit">{{$heir->name}}</a>
                    @endforeach

                @endforeach
            </div>
            @else
                @include('categories.notFoundCategories')
            @endif
        </div>

    @endsection
```

Б.7 Текст файлу categories/create.blade.php

```
    @extends('layouts.layout')

    @section('content')
        <link rel="stylesheet" href="/src/css/categoryStyles.css">

        <br>
        <h5 class="d-inline">Додавання нової категорії</h5>
        <div class="line"></div>

        <form action="/categories/create" method="post">

            {{csrf_field()}}
            <div class="form-group">
                <label for="name"><h6>Назва:</h6></label>
                <input class="form-control cat-name-input" type="text" name="name" id="name" value="{{old('name')}}">
            </div>

            <br>
```

```
            <div class="form-group">
                <label for="name"><h6>Надкатегорія (за необхідності):</h6></label>
                <div><select class="custom-select heirs-dropdown" name="heir_id">
                    <option value="" selected>Немає</option>
                    @foreach($basicCategories as $category)
                        <option value={{$category->id}}>{{$category->name}}</option>
                    @endforeach
                </select>
                </div>
            </div>

            <br>

            <div class="form-group">
                <button class="button-apply" type="submit">Створити</button>
            </div>

            @include('layouts.error')

        </form>

    @endsection
```

Б.8 Текст файлу categories/edit.blade.php

```
    @extends('layouts.layout')

    @section('content')
        <link rel="stylesheet" href="/src/css/categoryStyles.css">
        <br>

        <div>
            <h5>Редагування категорії</h5>
        </div>
        <div class="line"></div>

        <div class="form-box">
            <form action="/categories/{{$category->id}}/edit" method="post">

                {{csrf_field()}}
                {!! method_field('patch') !!}


                <div class="form-group">
                    <h6 for="name">Назва:</h6>
                    <input class="input-group-text heirs-dropdown text-sm-left" type="text"
name="name" id="name" value="{{$category->name}}">
                </div>
```

```
                <div class="form-group">

                    <div class="form-group">
                        <button class="button-apply" type="submit">Оновити</button>
                    </div>

                </div>

                @include('layouts.error')

            </form>
        </div>

        <div class="cat-trash-button">
            <button class="cat-trash" type="submit" data-toggle="modal" data-target="#confirmWindow"><i class="bi bi-trash3"></i></button>
        </div>

        <!-- Modal confirm window-->
        <div class="modal blur-bg" id="confirmWindow" tabindex="-1" role="dialog" aria-labelledby="confirmWindowTitle" aria-hidden="true">
            <div class="modal-dialog modal-dialog-centered" role="document">
                <div class="mod-win-body">
                    <div class="mod-win-head">
                        <div class="mod-win-head-text"><h5 class="modal-title" id="confirmWindowTitle">Підтвердження дії</h5></div>
                        <button type="button" class="mod-win-button-close" data-dismiss="modal" aria-label="Close">
                            <span aria-hidden="true"><i class="bi bi-x-lg"></i></span>
                        </button>
                    </div>
                    <div class="modal-body mod-win-center">
                        Ви впевнені що хочете видалити цю категорію ?
                        <br>
                        <i>(Дію буде неможливо відмінити)</i>
                        @if(is_null($category->heir_id))
                            <div class="mod-win-addit-text">Дана категорія має підкатегорії. Її видалення автоматично видалить усі залежні категорії!</div>
                        @endif
                    </div>
                    <div class="mod-win-foot">
                        <button type="button" class="close-button" data-dismiss="modal">Закрити</button>
                        <form action="/categories/{{$category->id}}" method="post">
                            {{csrf_field()}}
                            {!! method_field('delete') !!}
                            <button type="submit" class="button-apply">Видалити</button>
                        </form>
                    </div>
                </div>
            </div>
        </div>
```

```
@endsection
```

Б.9 Текст файлу categories/index.blade.php

```
@extends('layouts.layout')

@section('content')
    <link rel="stylesheet" href="/src/css/categoryStyles.css">
    <script type="text/javascript" src="/src/js/logicCategories.js"></script>

    <br>
    <h5 class="d-inline">Категорії</h5>

    @if (Auth::check())
        @if(Auth::user()->hasRole('admin'))
            <div class="category-control-panel d-inline">
                <a class="category-control-button" href="/categories/create" role="button" title="Додати категорію"><i class="bi bi-plus-circle"></i></a>
                <a class="category-control-button" href="/categories/controlPanel" role="button" title="Панель керування категоріями"><i class="bi bi-pencil-fill"></i></a>
            </div>
        @endif
    @endif

    <div class="line"></div>

    <!-- Картки з категоріями -->
    @if(count($categories))
        <div class="row category-block">
            @foreach($categories as $category)
                <div class="category-card">
                    <div class="main-category"><a class="main-category-text" href="/categories/{{$category['id']}}/posts">{{$category['name']}}</a></div>
                    <div class="heirs-block">
                        @if(count($category['heirs']))
                            @foreach($category['heirs'] as $heir)

                                <div class="heirs"><i class="bi bi-dot"></i><a class="heirs-text" href="/categories/{{$heir->id}}/posts">{{$heir->name}}</a></div>
                            @endforeach
                        @else
                            <div class="heirs no-categories">Підкатегорій немає</div>
                        @endif
                    </div>

                </div>
                <br>
```

```
            @endforeach
        </div>
    @else
        @include('categories.notFoundCategories')
    @endif

@endsection
```

Б.10 Текст файлу notFoundCategories.blade.php

```
<div class="error-message">
    <i class="bi bi-emoji-frown"></i>
    <div>
        <h2>Упс, поки-що тут порожньо</h2>
    </div>
</div>
```

Б.11 Текст файлу descriptionPage.blade.php

```
@extends('layouts.layout')

@section('content')
    <!-- Description Page Styles -->
    <link rel="stylesheet" href="/src/css/descriptionStyles.css">
    <script type="text/javascript" src="/src/js/logicDescription.js"></script>

    <div class="description-box">
        <div class="description-bg">
            <div class="description-slider-box">
                <div class="description-title-text">dot.</div>
                <div>
                    <div class="slider-line slider-line-active"></div>
                    <div class="slider-line"></div>
                </div>
                <div class="description-text">Знайдіть те, що вам довеось би шукати по
всьому Інтернету</div>
            </div>
        </div>
        <br>
    </div>

    <div class="container">
        <div class="row about-info-block">
            <div class="col">
                <img src="../../src/img/Hello.png" class="about-img about-left">
                <h5>Вітаємо на нашому форумі.</h5>
                <div class="line about-line-left"></div>
                <p>Тут Ви знайдете безліч статей пов'язаних з програмуванням,
```

```
                кібербезпекою, сис.адмініструванням та іншими галузями IT.
                Велика кількість інформації, взята з найрізноманітніших закутків
                Інтернету, зібрана в одному місці. Кожний запис перевіряється
                модераторами форуму на доцільність, точність викладення інформації
та актуальність.
              </p>
            </div>
          </div>

          <div class="row about-info-block">
            <div class="col-8 about-right">
              <h5>Візьміть найважливіше.</h5>
              <div class="line"></div>
              <p>На форумі є можливість відсортовувати інформацію,
                залишаючи лише те, що цікавить саме Вас. У разі,
                коли необхідно щось конкретне - скористайтесь нашою
                пошуковою системою. Для спрощення пошуку також створено
декілька
                фільтрів. Всі пости пов'язані між собою певними категоріями - Ви
                ніколи не заблукаєте, шукаючи необхідне.
              </p>
            </div>
            <div class="col-3">
              <img src="../../src/img/infoSorting.png" class="about-img">
            </div>
          </div>


          <div class="row about-info-block">
            <div class="col">
              <img src="../../src/img/activityMonitoring.png" class="about-img about-
left">
              <h5>Зручний моніторинг активності.</h5>
              <div class="line about-line-left"></div>
              <p>Наш форум є повністю прозорим і надає можливість усім
переглядати
                активну діяльність інших користувачів. Дивіться які пости публікують
                ваші улюблені автори. Спостерігайте за тим, чим саме цікавляться та
що
                коментують друзі та колеги. Можливо Ваші інтереси збігаються з
інтересами
                іншого користувача та інформацію стане шукати набагато простіше.
              </p>
            </div>
          </div>

          <div class="row about-info-block">
            <div class="col-8 about-right">
              <h5>Авторизуйтесь та отримайте додаткові можливості.</h5>
              <div class="line"></div>
              <p>Ви можете зареєструватись на форумі, щоб власноруч створювати
нові статті
```

```
                або залишати коментарі під вже існуючими. Окрім цього зареєстровані користувачі
                мають змогу додавати власні закладки на будь-які пости сайту. Не втрачайте нагоди
                відкрити для себе всі зручності нашого форуму.
              </p>
            </div>

            <div class="col-3">
              <img src="../../src/img/signUp.png" class="about-img">
            </div>
          </div>
        </div>
      @endsection
```

Б.11 Текст файлу footer.blade.php

```
      <div class="container-fluid footer-block">
        <div class="row justify-content-center" id="footerRow">
          <div class="col-3 content">
            <div class="footer-logo">dot.</div>
            <div class="footer-interactions">
              <span><a href="/" class="footer-links">Публікації</a></span>
              <span><a href="" class="footer-links">Техпідтримка</a></span>
              <span><a href="" class="footer-links">Розробники</a></span>
              <span><a href="" class="footer-links">Співпраця</a></span>
            </div>
            <div class="social-media">
              <span><a href="" class="telegram"><i class="bi bi-telegram"></i></a></span>
              <span><a href="" class="instagram"><i class="bi bi-instagram"></i></a></span>
              <span><a href="" class="facebook"><i class="bi bi-facebook"></i></a></span>
              <span><a href="" class="youtube"><i class="bi bi-youtube"></i></a></span>
              <span><a href="" class="twitter"><i class="bi bi-twitter"></i></a></span>
            </div>
            <div class="copyrighting">© 2022-2023, dot.</div>
          </div>
        </div>
      </div>
```

Б.12 Текст файлу headerNavigation.blade.php

```
      <nav class="navbar navbar-expand-lg navbar-dark navbar-cust">

        <div class="logo-box">
```

```
            <a class="navbar-brand logo" href="/description">dot.</a>
            <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarColor01" aria-controls="navbarColor01" aria-expanded="false" aria-label="Toggle navigation">
              <span class="navbar-toggler-icon"></span>
            </button>
          </div>

          <div class="searchBar-box">
            <form action="/search" class="searchBar-cust" method="get">
              <div class="search">
                @isset($searchText)
                  <input class="searchBar-form" type="search" placeholder="Пошук" aria-label="Search" id="s" name="s" autocomplete="off" value="{{$searchText}}">
                @else
                  <input class="searchBar-form" type="search" placeholder="Пошук" aria-label="Search" id="s" name="s" autocomplete="off">
                @endisset

                <button type="submit" class="dandruff-button"><i class="bi bi-search"></i></button>
              </div>
            </form>
          </div>

          <div class="collapse navbar-collapse r-side-nav-bar row" id="navbarColor01">
            <div class="button-panel col">
              <a class="panel-elem" href="/"><i class="bi bi-house-fill navbar-button"></i></a>
              <a class="panel-elem" href="/users"><i class="bi bi-people-fill navbar-button"></i></a>
              <a class="panel-elem" href="/categories"><i class="bi bi-grid-fill navbar-button"></i></a>

              @if (Auth::check())
                <a href="/posts/create" class="panel-elem"><i class="bi bi-plus-circle-fill navbar-button"></i></a>
              @endif
            </div>

            @guest
              @if (Route::has('login'))
                <a href="{{ route('login') }}" class="login-button">Увійти</a>
              @endif
              @if (Route::has('register'))
                <a href="{{ route('register') }}" class="regist-button">Реєстрація</a>
              @endif


            @else

              <!-- User logo dropdown -->
```

```
                <div class="dropdown">
                    <button class="btn user-logo-dropdown" type="button" id="dropdownMenuButton" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                        @if(Auth::user()->photo)
                            <img src="{{Auth::user()->getPhotoPath(Auth::id()).Auth::user()->photo}}" class="post-user-logo">
                        @else
                            <img src="../../src/img/default_user_photo.png" class="post-user-logo">
                        @endif
                    </button>
                    <div class="dropdown-menu dropdown-menu-right user-logo-menu">
                        <div class="menu-user-info">
                            <div class="info-inline">
                                @if(Auth::user()->photo)
                                    <img src="{{Auth::user()->getPhotoPath(Auth::id()).Auth::user()->photo}}" class="menu-user-avatar">
                                @else
                                    <img src="../../src/img/default_user_photo.png" class="menu-user-avatar">
                                @endif
                            </div>
                            <div class="info-inline">
                                <div class="category"><div class="category-text">{{ Auth::user()->name }}</div></div>
                                <div>
                                    @if(Auth::user()->username)
                                        {{Auth::user()->username}}
                                    @else
                                        Юзернейм відсутній
                                    @endif
                                </div>
                            </div>
                        </div>
                        <div class="dropdown-divider"></div>
                        <a class="menu-user-item" href="/users/{{Auth::id()}}"><i class="bi bi-person-fill menu-user-icons"></i>Профіль</a>
                        <a class="menu-user-item" href="/users/{{Auth::id()}}/bookmarks"><i class="bi bi-award-fill indent menu-user-icons"></i>Закладки</a>
                        <div class="dropdown-divider"></div>
                        <a class="menu-user-item" href="{{ route('logout') }}" onclick="event.preventDefault(); document.getElementById('logout-form').submit();"><i class="bi bi-box-arrow-in-right menu-user-icons"></i>Вихід</a>
                        <form id="logout-form" action="{{ route('logout') }}" method="POST">
                            @csrf
                        </form>
                    </div>
                </div>

            @endguest
```

```
    </div>
</nav>
```

Б.13 Текст файлу layout.blade.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- The above 3 meta tags *must* come first in the head; any other head content must come *after* these tags -->
    <title>IT Forum</title>

    <!-- Bootstrap -->
    <script src="https://code.jquery.com/jquery-3.5.1.min.js" crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>

    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js" integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6" crossorigin="anonymous"></script>

    <link href="https://cdn.jsdelivr.net/npm/summernote@0.8.18/dist/summernote-bs4.min.css" rel="stylesheet">
    <script src="https://cdn.jsdelivr.net/npm/summernote@0.8.18/dist/summernote-bs4.min.js"></script>
    <!-- Bootstrap Icons -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.10.3/font/bootstrap-icons.css">
    <!--------CSS Styles-------->
    <!-- Main styles for site pages-->
    <link rel="stylesheet" href="/src/css/mainStyles.css">
    <!-- Header CSS -->
    <link rel="stylesheet" href="/src/css/navBar.css">

    <!-- Footer CSS -->
    <link rel="stylesheet" href="/src/css/footerStyles.css">
```

```
            <!-- Posts List CSS -->
            <link rel="stylesheet" href="/src/css/postsList.css">

            <!-- User Logo CSS-->
            <link rel="stylesheet" href="/src/css/userLogo.css">

            <!-- User List CSS-->
            <link rel="stylesheet" href="/src/css/userList.css">

            <!-- Filter CSS-->
            <link rel="stylesheet" href="/src/css/filter.css">

            <!-- Post CSS -->
            <link rel="stylesheet" href="/src/css/postStyles.css">

            <!-- TinyMCE Custom Styles -->
            <link rel="stylesheet" href="/src/css/tinymceStyles.css">

            <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media
queries -->
            <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
            <!--[if lt IE 9]>
            <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
            <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
            <![endif]-->


            <script
src="https://cdn.tiny.cloud/1/8cand97lughjs8hcgaz479948txr7cv39scp9squys33pdlg/tinymce/6/ti
nymce.min.js" referrerpolicy="origin"></script>
        </head>
        <body class="page-colors">

            @include('layouts.headerNavigation')

            <div class="container" id="mainContainer">
                @yield('content')
            </div>

            @include('layouts.footer')

            <script type="text/javascript" src="/src/js/tinymceCore.js"></script>

        </body>

        </html>
```

Б.14 Текст файлу posts/create.blade.php

```
@extends('layouts.layout')

@section('content')
    <link rel="stylesheet" href="/src/css/postEdit.css">
    <script>
        window.onload = function () {
            @foreach($categories as $category)
                setCategories('{{$category->name}}', '{{$category->id}}');
            @endforeach
        }
    </script>
    <script type="text/javascript" src="/src/js/logicPost.js"></script>
    <br>
    <h5>Створення публікації</h5>
    <div class="line"></div>

    <form action="/post" method="post">
        {{csrf_field()}}

        <div class="form-group">
            <h6>Заголовок:</h6>
            <input class="form-control title-line" type="text" name="title" id="title"
value="{{old('title')}}">
        </div>

        <div>
            <div id="insertInputPlace"></div>
            <div class="d-inline">
                <button type="button" class="add-cat-button"
onclick="createCategoriesInput()"><i class="bi bi-plus-lg"></i> Категорія</button>
            </div>
        </div>
        <br>

        <div class="form-group">
            <h6>Опис:</h6>
            <div class="dark-skin-editor">
                @if(Auth::user()->hasRole('admin'))
                    <textarea class="form-control" type="text" name="htmlText"
id="tinymceForAdmin">{{old('mainText')}}</textarea>
                @else
                    <textarea class="form-control" type="text" name="htmlText"
id="tinymceForPostCreator" >{{old('mainText')}}</textarea>
                @endif
            </div>
        </div>

        <input name="mainText" type="hidden" id="TextWithoutHTML">

        <div class="form-group">
            <button class="btn regist-button" type="submit"
onclick="cutHTML()">Опублікувати</button>
```

```
            </div>
            @include('layouts.error')

        </form>

    @endsection
```

Б.15 Текст файлу posts/edit.blade.php

```
    @extends('layouts.layout')

    @section('content')
        <link rel="stylesheet" href="/src/css/postEdit.css">
        <script>
            window.onload = function () {
                @foreach($categories as $category)
                setCategories('{{$category->name}}', '{{$category->id}}');
                @endforeach

                @foreach($post->categories as $category)
                setOldCategoriesInput('{{$category->id}}');
                @endforeach
            }
        </script>
        <script type="text/javascript" src="/src/js/logicPost.js"></script>

        <br>
        <h5>Редагування публікації автора <a href="/users/{{$post->user['id']}}"
class="text-decoration-none text-light">{{$post->user['name']}}</a></h5>
        <div class="line"></div>

        <form action="/posts/{{$post->id}}" method="post">

            {{csrf_field()}}
            {!! method_field('patch') !!}

            <div class="form-group">
                <h6>Заголовок:</h6>
                <input class="form-control title-line" type="text" name="title" id="title"
value="{{$post->title}}">
            </div>

            <div>
                <div id="insertInputPlace"></div>
                <div class="d-inline">
                    <button type="button" class="add-cat-button"
onclick="createCategoriesInput()"><i class="bi bi-plus-lg"></i> Категорія</button>
                </div>
            </div>
            <br>

            <div class="form-group">
```

```
                <h6>Опис:</h6>
                <div class="dark-skin-editor">
                    @if(Auth::user()->hasRole('admin'))
                        <textarea class="form-control" type="text" name="htmlText"
id="tinymceForAdmin">{{$post->htmlText}}</textarea>
                    @else
                        <textarea class="form-control" type="text" name="htmlText"
id="tinymceForPostCreator" >{{$post->htmlText}}</textarea>
                    @endif
                </div>
            </div>

            <input name="mainText" type="hidden" id="TextWithoutHTML">

            <div class="form-group">
                <button class="btn regist-button" type="submit"
onclick="cutHTML()">Зберегти</button>
            </div>

            @include('layouts.error')
        </form>


    @endsection
```

Б.16 Текст файлу notFoundPosts.blade.php

```
<div class="error-message">
    <i class="bi bi-emoji-frown"></i>
    <div>
        <h2>Записів не знайдено...</h2>
    </div>
</div>
```

Б.17 Текст файлу posts/show.blade.php

```
@extends('layouts.layout')

@section('content')
    <link rel="stylesheet" href="/src/css/postEdit.css">
    <script type="text/javascript" src="/src/js/logicPost.js"></script>
        <div class="">
            <!--  -->
            <div class="post-block">
                <div class="col">
                    @if (Auth::check())
                        <div class="dropdown threeDots">
```

```
                                <button class="threeDots-button" type="button" id="dropdownMenuButton" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                                    <div><i class="bi bi-three-dots-vertical"></i></div>
                                </button>
                                <div class="dropdown-menu dropdown-additions" aria-labelledby="dropdownMenuButton">
                                    @if(Auth::id() == $post->user['id'] || Auth::user()->hasRole('admin'))
                                        <!-- Кнопка Редагування -->
                                    <a class="dropdown-item dropdown-item-additions" href="/posts/{{$post->id}}/edit"><i class="bi bi-pencil indent"></i>Редагувати</a>

                                    <form action="/posts/{{$post->id}}" method="post">
                                        {{csrf_field()}}
                                        {!! method_field('delete') !!}
                                        <!-- Кнопка Видалення -->
                                        <button type="submit" class="dropdown-item dropdown-item-additions"><i class="bi bi-trash3 indent"></i>Видалити</button>
                                    </form>
                                    @endif
                                    @include('posts.showLikeButtons')
                                    @include('posts.showBookmarkButtons')
                                </div>
                            </div>
                        @endif

                        <h2 class="blog-post-title">{{$post->title}}</h2>
                        <div class="blog-post-data">

                            <span>
                            @if(count($post->categories))
                                @foreach($post->categories as $category)
                                    <div class="category"><a class="category-text" href="/categories/{{$category->id}}/posts">{{$category->name}}</a></div>
                                @endforeach
                            @endif
                            </span>

                            <span>{{$post->created_at}} / </span>
                            <span><a href="/users/{{$post->user['id']}}" class="blog-post-author"><span>@</span>{{$post->user['name']}}</a></span>
                        </div>
                        <div class="line"></div>

                        <p>{!! $post->htmlText !!}</p>


                        <div class="blog-post-info">
                            <!-- Comments -->
                            <span class="blog-post-info-elem" title="Comments amount"><i class="bi bi-chat-dots-fill post-info"></i>{{ $post->amount_comments }}</span>
```

```
                        <!-- Likes -->
                        <span class="blog-post-info-elem" title="Likes amount"><i class="bi
bi-caret-up-fill post-info"></i>{{ $post->amount_likes }}</span>
                        <!-- Dislikes -->
                        <span class="blog-post-info-elem" title="Dislikes amount"><i
class="bi bi-caret-down-fill post-info"></i>{{ $post->amount_dislikes }}</span>
                        <!-- Rating -->
                        <span class="blog-post-info-elem" title="Rating"><i class="bi bi-bar-
chart-fill"></i> {{ $post->rating }}</span>
                        <!-- Views -->
                        <span class="blog-post-info-elem" title="Views amount"><i class="bi
bi-eye-fill post-info"></i>{{ $post->amount_views }}</span>
                    </div>
                </div>
            </div>
            <!--  -->


            @if (Auth::check())
                <form action="/posts/{{$post->id}}/comment" method="post"
id="comment-form">
                    {{csrf_field()}}
                    <div class="form-group">
                        <h6>Коментувати як <a class="comment-as" href="/users/{{
Auth::user()->id }}">{{ Auth::user()->name }}</a></h6>
                        <div class="line"></div>
                        <div id="replyMessagePoint"></div>

                        <div class="dark-skin-editor"><textarea class="form-control"
type="text"                                                    name="htmlText"
id="tinymceForCommentator">{{old('htmlText')}}</textarea></div>
                        <input name="mainText" type="hidden" id="TextWithoutHTML">
                    </div>
                    <div class="form-group">
                        <button class="btn regist-button" type="submit"
onclick="cutHTML()" id="send-button">Відправити</button>
                    </div>
                    @include('layouts.error')
                </form>
            @endif
            <br>

            <div><h3>Коментарі:</h3></div>
            @foreach($post->comments as $comment)
                <div id="{{$comment->pivot->id}}">
                    <div class="comment-box">
                        <div class="comment-info">
                            <span><a class="comment-author" href="/users/{{$comment-
>pivot->user_id}}">@ {{$comment->name}}</a></span>
                            <span>{{$comment->pivot->created_at}}</span>
                        </div>
```

```
                        <p class="card-text">{!! $comment->pivot->htmlText !!}</p>
                        @if(!is_null($comment->pivot->reply_message_id))
                          <p class="reply">Відповів на: <i class="answer-to"
onclick="smoothScroll('{{$comment->pivot->reply_message_id}}')">{{$comment->pivot-
>reply_message_text}}</i></p>
                        @endif

                          @if (Auth::check())
                          <button type="submit" title="Відповісти" class="comment-
interaction" onclick="setReplyOnComment('{{$comment->pivot->intro}}','{{$comment-
>pivot->id}}')"><i class="bi bi-reply"></i></button>
                            @if(Auth::id() == $comment->pivot->user_id || Auth::user()-
>hasRole('admin'))
                              @if(Auth::id() == $comment->pivot->user_id)
                              <button type="submit" title="Редагувати" class="comment-
interaction" onclick="editComment('{{$comment->pivot->intro}}',
                                                                    '{{$comment->pivot-
>id}}',
                                                                    '{{$comment->pivot-
>htmlText}}',
                                                                    '{{$comment->pivot-
>reply_message_id}}',
                                                                    '{{$comment->pivot-
>reply_message_text}}',
                                                                    '{{$comment->pivot-
>post_id }}')"><i class="bi bi-pencil"></i></button>
                              @endif
                                <form class="d-inline" action="/posts/{{$post-
>id}}/comment/{{$comment->pivot->id}}" method="post">
                                  {{csrf_field()}}
                                  {!! method_field('delete') !!}
                                  <button type="submit" title="Видалити"
class="comment-interaction" ><i class="bi bi-trash3"></i></button>
                                </form>
                            @endif
                          @endif
                        </div>
                      </div>
                      <br>
                    @endforeach
                  </div>
        @endsection
```

Б.18 Текст файлу showBookmarkButtons.blade.php

```
        @if(!is_null($bookmark))
          <form action="/posts/{{$post->id}}/bookmarks/{{$bookmark->id}}"
method="post">
            {{csrf_field()}}
            {!! method_field('delete') !!}
```

```
            <button type="submit" class="dropdown-item dropdown-item-additions"><i
class="bi bi-award-fill indent"></i>Прибрати закладку</button>
          </form>
        @else
          <form action="/posts/{{$post->id}}/bookmarks" method="post">
            {{csrf_field()}}
            <button type="submit" class="dropdown-item dropdown-item-additions"><i
class="bi bi-award indent"></i>Додати закладку</button>
          </form>
        @endif
```

Б.19 Текст файлу showCommentatorsList.blade.php

```
<div class="col-4">
  <div class="card border-danger">
    <div class="card-header bg-danger text-white">
      Коментатори
    </div>

    <div class="card-body">

      @foreach($commentators as $commentator)
        <h5>
          <a href="/users/{{$commentator->pivot->user_id}}">{{$commentator-
>name}}</a>
        </h5>
      @endforeach
    </div>
  </div>
</div>
```

Б.20 Текст файлу showLikeButtons.blade.php

```
@if (Auth::check())

  @if($userLike != NULL)
    @if($userLike->grade == True)
      <form action="/posts/{{$post->id}}/like/{{$userLike->id}}" method="post">
        {{csrf_field()}}
        {!! method_field('delete') !!}
        <button type="submit" class="dropdown-item dropdown-item-additions"
id="likeButton"><i class="bi bi-caret-up-fill indent"></i>Like</button>
      </form>

      <form action="/posts/{{$post->id}}/like/{{$userLike->id}}" method="post">
        {{csrf_field()}}
        {!! method_field('patch') !!}
        <input name="grade" value=0 type="hidden">
```

```
                    <button type="submit" class="dropdown-item dropdown-item-additions"
id="dislikeButton"><i class="bi bi-caret-down indent"></i></i>Dislike</button>
                </form>
              @else
                <form action="/posts/{{$post->id}}/like/{{$userLike->id}}" method="post">
                  {{csrf_field()}}
                  {!! method_field('patch') !!}
                  <input name="grade" value=1 type="hidden">
                  <button type="submit" class="dropdown-item dropdown-item-additions"
id="likeButton"><i class="bi bi-caret-up indent"></i>Like</button>
                </form>

                <form action="/posts/{{$post->id}}/like/{{$userLike->id}}" method="post">
                  {{csrf_field()}}
                  {!! method_field('delete') !!}
                  <button type="submit" class="dropdown-item dropdown-item-additions"
id="dislikeButton"><i class="bi bi-caret-down-fill indent"></i>Dislike</button>
                </form>
              @endif
            @else
              <form action="/posts/{{$post->id}}/like/" method="post">
                {{csrf_field()}}
                <input name="grade" value=1 type="hidden">
                <button type="submit" class="dropdown-item dropdown-item-additions"
id="likeButton"><i class="bi bi-caret-up indent"></i>Like</button>
              </form>

              <form action="/posts/{{$post->id}}/like/" method="post">
                {{csrf_field()}}
                <input name="grade" value=0 type="hidden">
                <button type="submit" class="dropdown-item dropdown-item-additions"
id="dislikeButton"><i class="bi bi-caret-down indent"></i></i>Dislike</button>
              </form>
            @endif
          @endif
```

Б.21 Текст файлу users/edit.blade.php

```
          @extends('layouts.layout')

          @section('content')
            <link rel="stylesheet" href="/src/css/userProfileStyles.css">

            <script
src="https://cdnjs.cloudflare.com/ajax/libs/filereader/2.1.0/filereader.min.js"></script>
            <script type="text/javascript" src="/src/js/logicUserEditPage.js"></script>

            <br>
            <h5>Редагування профілю</h5>
            <div class="line"></div>
```

```
            <br>
            <h6>Зміна аватару</h6>
            <div class="line"></div>

            <form action="/users/{{$user->id}}" method="post" enctype="multipart/form-
data">

            <div class="avatar-block">
               @if($user->photo)
                  <img src="{{$user->getPhotoPath($user->id).$user->photo}}" class="user-
avatar" id="newPhoto">
                  <div>
                     <br>
                     <span>Картинки мають бути форматів png, jpg або bmp</span>
                     <br>
                     <input type="file" name="photo" id="photoInput" accept=".jpg, .png, .bmp">
                     <button class="input-photo" type="button"
id="customPhotoInput">Завантажити фото</button>
                     <br>
                     <button id="deletePhotoButton" type="reset" class="delete-photo"
onclick="deletePhoto()"><i class="bi bi-trash3"></i> Видалити фото</button>
                  </div>
               @else
                  <img src="../../src/img/default_user_photo.png" class="user-avatar"
id="newPhoto">
                  <div>
                     <span>Картинки мають бути форматів png, jpg або bmp</span>
                     <br>
                     <br>
                     <input type="file" name="photo" id="photoInput" accept=".jpg, .png, .bmp">
                     <button class="input-photo" type="button"
id="customPhotoInput">Завантажити фото</button>
                     <br>
                     <button id="deletePhotoButton" type="reset" class="hide-delete-photo-button
delete-photo" onclick="deletePhoto()"><i class="bi bi-trash3"></i> Видалити фото</button>
                  </div>
               @endif
            </div>


               {{csrf_field()}}
               {!! method_field('patch') !!}

               <br>
               <h6>Зміна особистих даних</h6>
               <div class="line"></div>
               <div>
                  <label for="name">Ім'я:</label>
                  <input class="form-control usr-profile-input" type="text" name="name"
id="name"
```

```
                    @if(old('name') == '')
                        value="{{$user->name}}"
                    @else
                        value="{{old('name')}}"
                    @endif
                >
            </div>

            <br>
            <div>
                <label for="description">Про себе:</label>
                <input class="form-control usr-profile-input" type="text" name="description"
id="description"
                    @if(old('description') == '')
                        value="{{$user->description}}"
                    @else
                        value="{{old('description')}}"
                    @endif
                >
            </div>

            <br>
            <div>
                <label for="username">Псевдонім:</label>
                <input class="form-control usr-profile-input" type="text" name="username"
id="username"
                    @if(old('username') == '')
                        value="{{substr($user->username, 1)}}"
                    @else
                        value="{{old('username')}}"
                    @endif
                >
            </div>

            <input type="hidden" id="flagPhoto" name="flagPhoto">


            <br>
            <div>
                <button class="btn regist-button" type="submit">Зберегти</button>
            </div>
            <br>

            @include('layouts.error')

        </form>

    @endsection
```

Б.22 Текст файлу users/index.blade.php

```
@extends('layouts.layout')

@section('content')
   @if(count($users))
      <br>
      <h5>Користувачі сайту</h5>
      <div class="line"></div>
      <div class="row user-list-pos">
         @foreach($users as $user)
            <div class="media block">
               <div class="my-auto">
                  @if($user->photo)
                     <a href="/users/{{$user->id}}"><img src="{{$user->getPhotoPath($user->id).$user->photo}}" class="user-list-logo"></a>
                  @else
                     <a href="/users/{{$user->id}}"><img src="../src/img/default_user_photo.png" class="user-list-logo"></a>
                  @endif
               </div>
               <div class="media-body">
                  <div class="user-info">
                     <div class="category"><a class="category-text" href="/users/{{$user->id}}">{{$user->name}}</a></div>
                     <div>
                        @if($user->username)
                           {{$user->username}}
                        @else
                           Юзернейм відсутній
                        @endif
                     </div>
                     <span><i class="bi bi-chat-dots-fill post-info" title="Comments amount"></i>{{$user->amount_comments}}</span>
                     <span><i class="bi bi-pen-fill" title="Posts amount"></i> {{$user->posts_amount}}</span>
                  </div>
               </div>
            </div>
         @endforeach
      </div>

      <div class="pagination-area">
         <div class="pagination-box">
               {{ $users->links() }}
         </div>
      </div>
   @endif

@endsection
```

Б.23 Текст файлу users/show.blade.php

```
@extends('layouts.layout')

@section('content')
  <link rel="stylesheet" href="/src/css/userProfileStyles.css">

  <div class="col">
      <br>
      <h5 class="d-inline">Редагування профілю</h5>
      @if (Auth::check())
        @if(Auth::id() == $user->id)
          <div class="user-control-panel d-inline">
            <a  href="/users/{{$user->id}}/edit"  class="user-control-button"><i class="bi bi-pencil-fill"></i></a>
            <a  href="/users/{{$user->id}}/bookmarks"  class="user-control-button"><i class="bi bi-award indent"></i></a>
          </div>
        @endif
      @endif
    <div class="line"></div>


    <div class="about-block">
      @if($user->photo)
        <img  src="{{$user->getPhotoPath($user->id).$user->photo}}"  class="user-avatar">
      @else
        <img src="../../src/img/default_user_photo.png" class="user-avatar">
      @endif

      <div>
        <div class="title-margin">Ім'я :</div>

        <div class="title-margin">Про користувача :</div>

        <div class="title-margin">Псевдонім :</div>

        <div class="title-margin">Кількість постів :</div>

        <div class="title-margin">Кількість коментарів :</div>

        <div class="title-margin">На форумі з :</div>
      </div>

      <div>
        <div class="data-margin">{{$user->name}}</div>

        <div class="data-margin">{{$user->description}}</div>

        <div class="data-margin">{{$user->username}}</div>

        <div class="data-margin">{{$user->posts_amount}}</div>
```

```
                <div class="data-margin">{{$user->amount_comments}}</div>

                <div class="data-margin">{{$user->created_at}}</div>
              </div>
            </div>

            <br>
            <h5>Активність на форумі</h5>
            <div class="line"></div>
            <a href="/users/{{$user->id}}" class="is-button-default
                                        @isset($posts) is-button-active @endisset
                                        @isset($comments) is-button-disabled @endisset
                                    ">Пости користувача</a>
            <a href="/users/{{$user->id}}/comments" class="is-button-default
                                        @isset($posts) is-button-disabled @endisset
                                        @isset($comments) is-button-active @endisset
                                    ">Коментарі користувача</a>
            <div  id="userProfileContent">
              @yield('userProfileContent')
            </div>
          </div>

        @endsection
```

Б.24 Текст файлу notFoundBookmarks.blade.php

```
<div class="error-message">
  <i class="bi bi-emoji-frown"></i>
  <div>
    <h2>Поки що Ви не додали жодної закладки</h2>
  </div>
</div>
```

Б.25 Текст файлу userBookmarks.blade.php

```
@extends('layouts.layout')

@section('content')
  <link rel="stylesheet" href="/src/css/userProfileStyles.css">
  <br>
  <h5>Мої закладки</h5>
  <div class="line"></div>
  <div  class="d-inline"><a  href="/users/{{$user->id}}"  class="no-link-decor"><span
class="back-to-profile-button"><i class="bi bi-arrow-left"></i> Мій профіль</span></a></div>


  @if(count($bookmarks))
    <div class="d-inline">
```

```
                <button type="button" class="bookmarks-delete-button" data-toggle="modal" data-target="#confirmWindow">
                    Видалити всі закладки
                </button>
            </div>

            <!-- Modal confirm window-->
            <div class="modal blur-bg" id="confirmWindow" tabindex="-1" role="dialog" aria-labelledby="confirmWindowTitle" aria-hidden="true">
                <div class="modal-dialog modal-dialog-centered" role="document">
                    <div class="mod-win-body">
                        <div class="mod-win-head">
                            <div class="mod-win-head-text"><h5 class="modal-title" id="confirmWindowTitle">Підтвердження дії</h5></div>
                            <button type="button" class="mod-win-button-close" data-dismiss="modal" aria-label="Close">
                                <span aria-hidden="true"><i class="bi bi-x-lg"></i></span>
                            </button>
                        </div>
                        <div class="modal-body mod-win-center">
                            Ви впевнені що хочете видали усі закладки ?
                            <br>
                            <i>(Дію буде неможливо відмінити)</i>
                        </div>
                        <div class="mod-win-foot">
                            <button type="button" class="close-button" data-dismiss="modal">Закрити</button>
                            <form action="/users/{{$user->id}}/allDeleteBookmarks" method="post">
                                {{csrf_field()}}
                                {!! method_field('delete') !!}
                                <button type="submit" class="button-apply">Видалити</button>
                            </form>
                        </div>
                    </div>
                </div>
            </div>


            <div class="row">
                @foreach($bookmarks as $bookmark)
                    <div class="posts-container">
                        <div class="col">
                            <div class="dropdown threeDots">
                                <button class="threeDots-button" type="button" id="dropdownMenuButton" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                                    <div><i class="bi bi-three-dots-vertical"></i></div>
                                </button>
                                <div class="dropdown-menu dropdown-additions" aria-labelledby="dropdownMenuButton">
```

```
                                <form action="/users/{{$user->id}}/bookmarks/{{$bookmark->pivot['id']}}" method="post">
                                    {{csrf_field()}}
                                    {!! method_field('delete') !!}
                                    <button type="submit" class="dropdown-item dropdown-item-additions"><i class="bi bi-award-fill indent"></i>Видалити</button>
                                </form>

                            </div>
                        </div>


                        <a href="/posts/{{$bookmark->pivot['post_id']}}" class="title-font">{{ $bookmark->title }}</a>
                        <p>{{ $bookmark->intro }}</p>
                        <br>
                        <div class="userName-box">

                            @if($bookmark->user->photo)
                                <a href="/users/{{$bookmark->user->id}}"><img src="{{$bookmark->user->getPhotoPath($bookmark->user->id).$bookmark->user->photo}}" class="post-user-logo"></a>
                            @else
                                <a href="/users/{{$bookmark->user->id}}"><img src="../../src/img/default_user_photo.png" class="post-user-logo"></a>
                            @endif

                            <a class="post-list-author" href="/users/{{$bookmark->user->id}}">{{ $bookmark->user->name }}</a>
                        </div>
                        <div class="info-box">
                            <span class="info-elem"><i class="bi bi-chat-dots-fill post-info"></i>{{ $bookmark->amount_comments }}</span>
                            <span class="info-elem"><i class="bi bi-caret-up-fill post-info"></i>{{ $bookmark->amount_likes }}</span>
                            <span class="info-elem"><i class="bi bi-caret-down-fill post-info"></i>{{ $bookmark->amount_dislikes }}</span>
                            <span class="info-elem"><i class="bi bi-eye-fill post-info"></i>{{ $bookmark->amount_views }}</span>
                        </div>
                        <br>
                    </div>
                </div>
                <div class="w-100"></div>
            @endforeach
        </div>

        <div class="pagination-area">
            <div class="pagination-box">
                {{ $bookmarks->links() }}
            </div>
        </div>
```

```
    @else
      @include('users.notFoundBookmarks')
    @endif
  @endsection
```

Б.26 Текст файлу userComments.blade.php

```
  @extends('users.show')

  @section('userProfileContent')
    <br>
    <div class="pagination-area">
      <div class="pagination-box">
        {{ $comments->links() }}
      </div>
    </div>


    <div class="col">
      @foreach($comments as $comment)

        <div class="row">
          <div class="posts-container">

            <div class="">
              <a  href="/posts/{{$comment->pivot->post_id}}/#{{$comment->pivot->id}}" class="title-font"><h5>Коментар: {{ $comment->pivot->intro }}</h5></a>
              <p  class="created-at">Створено:  {{$comment->pivot->created_at}}</p>
              <div class="line"></div>

            </div>
            <i>У пості:</i>
            <div class="on-post">{{$comment->title}}</div>
            <div class="line"></div>
          </div>


        </div>
      @endforeach
    </div>


    <br>
    <div class="pagination-area">
      <div class="pagination-box">
        {{ $comments->links() }}
      </div>
    </div>
  @endsection
```

Б.27 Текст файлу userPosts.blade.php

```
@extends('users.show')

@section('userProfileContent')
   <br>
   <div class="pagination-area">
      <div class="pagination-box">
         {{ $posts->links() }}
      </div>
   </div>

   <div class="col">
         @foreach($posts as $post)
            <!-- Ті самі пости, що виводяться на сторінці-->
            <div class="posts-container">
               <div class="col">
                  @if (Auth::check())
                     @if(Auth::id() == $post->user['id'] || Auth::user()->hasRole('admin'))
                        <div class="dropdown threeDots">
                           <button class="threeDots-button" type="button" id="dropdownMenuButton" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                              <div><i class="bi bi-three-dots-vertical"></i></div>
                           </button>
                           <div class="dropdown-menu dropdown-additions" aria-labelledby="dropdownMenuButton">
                              <a class="dropdown-item dropdown-item-additions" href="/posts/{{$post->id}}/edit"><i class="bi bi-pencil indent"></i>Редагувати</a>

                              <form action="/posts/deleteInProfile/{{$post->id}}" method="post">
                                 {{csrf_field()}}
                                 {!! method_field('delete') !!}
                                 <!-- Кнопка Видалення -->
                                 <button type="submit" class="dropdown-item dropdown-item-additions"><i class="bi bi-trash3 indent"></i>Видалити</button>
                              </form>

                           </div>
                        </div>
                     @endif
                  @endif

                  <a href="/posts/{{$post->id}}" class="title-font">{{ $post->title }}</a>
                  <p>{{ $post->intro }}</p>
```

```
                        <div class="post-category-line">
                            @if(count($post->categories))
                                @foreach($post->categories as $category)
                                    <div class="category"><a class="category-text"
href="/categories/{{$category->id}}/posts">{{$category->name}}</a></div>
                                @endforeach
                            @endif
                        </div>

                        <br>

                        <div class="userName-box">

                            @if($post->user['photo'])
                                <a href="/users/{{$post->user['id']}}"><img src="{{$post->user-
>getPhotoPath($post->user['id']).$post->user['photo']}}" class="post-user-logo"></a>
                            @else
                                <a href="/users/{{$post->user['id']}}"><img
src="../../src/img/default_user_photo.png" class="post-user-logo"></a>
                            @endif

                            <a class="post-list-author" href="/users/{{$post->user['id']}}">{{
$post->user['name'] }}</a>
                        </div>
                        <div class="info-box">
                            <!-- Comments -->
                            <span class="info-elem"><i class="bi bi-chat-dots-fill post-
info"></i>{{ $post->amount_comments }}</span>
                            <!-- Likes -->
                            <span class="info-elem"><i class="bi bi-caret-up-fill post-
info"></i>{{ $post->amount_likes }}</span>
                            <!-- Dislikes -->
                            <span class="info-elem"><i class="bi bi-caret-down-fill post-
info"></i>{{ $post->amount_dislikes }}</span>
                            <!-- Views -->
                            <span class="info-elem"><i class="bi bi-eye-fill post-info"></i>{{
$post->amount_views }}</span>
                        </div>
                        <br>
                    </div>
                </div>
                <div class="w-100"></div>
            @endforeach


        </div>
        <br>
        <div class="pagination-area">
            <div class="pagination-box">
                {{ $posts->links() }}
            </div>
        </div>
```

```
@endsection
```

Б.28 Текст файлу navBar.css

```
/* Additions for Navigation Bar*/
.navbar-cust{
    background-color: #000;
    position: sticky;
    z-index: 1400;
    top: 0;
}

.navbar-button{
    font-size: 26px;
    color: white;
}

.panel-elem{
    width: 33%;
    margin-right: 0.5em;
}

.r-side-nav-bar{
    position: relative;
}

.button-panel{
    width: 25%;
    display: flex;
}

/* Logo styles */
.logo{
    font-family: "Artifakt Element Medium";
    color: white;
    font-size: 26px;
}

/* Div that contains logotype*/
.logo-box{
    width: 30%;
}

/* The whole search bar styles*/
.searchBar-box{
    width: 40%;
}

.searchBar-cust{
    width: 90%;
```

```
}

.searchBar-form{
    width: 94%;
    height: 100%;
    color: white;
    background-color: #2f2f2f;
    border: none;
    outline: none;
}

.searchBar-form::placeholder{
    color: #5e5e5e;
    transition: 0.1s linear;
}

.searchBar-form:hover::placeholder{
    color: white;
}

.searchBar-form:focus::placeholder{
    color: white;
}

.search{
    border-radius: 20px;
    width: 100%;
    height: 2em;
    background-color: #2f2f2f;
    padding-left: 20px;
    border: none;
    display: flex;
}

.dandruff-button{
    background: none;
    outline: none;
    border: none;
    color: #5e5e5e;
    font-size: 20px;
    transition: 0.1s linear;
}

.dandruff-button:focus{
    outline: none;
}

.dandruff-button:hover{
    color: white;
}

/* Deleting default cross(clearing line) from search bar */
```

```
.searchBar-form[type=text]::-ms-clear {  display: none; width : 0; height: 0; }
.searchBar-form[type=text]::-ms-reveal {  display: none; width : 0; height: 0; }
.searchBar-form[type="search"]::-webkit-search-decoration,
.searchBar-form[type="search"]::-webkit-search-cancel-button,
.searchBar-form[type="search"]::-webkit-search-results-button,
.searchBar-form[type="search"]::-webkit-search-results-decoration { display: none; }
/*--------------------------------*/

/* Account Buttons Style */
/* Login Button */
.login-button{
   font-family: Calibri;
   font-size: 21px;
   color: white;
   text-decoration: none;
   margin-right: 0.5em;
}

.login-button:hover{
   color: white;
   text-decoration: none;
}

/* Register Button */
.regist-button{
   font-family: Calibri;
   font-weight: bold;
   font-size: 21px;
   background: none;
   color: white;
   border: 1px solid white;
   border-radius: 20px;
   padding-left: 1em;
   padding-right: 1em;
   padding-top: 0.15em;
   padding-bottom: 0.15em;
   margin-right: 0.5em;;
   transition: 0.1s linear;
   text-decoration: none;
}

.regist-button:hover{
   background-color: white;
   color: black;
   text-decoration: none;
}

/* User Logo Dropdown */

.user-logo-dropdown{
   background: none;
   margin-right: 20px;
```

```
    border: none;
}

.user-logo-dropdown:hover{
    background: none;
    border: none;
}

.user-logo-dropdown:focus{
    background: none;
    border: none;
    box-shadow: none;
}

.user-logo-menu{
    width: 250px;
    background-color: #3a383a;
    margin-right: 20px;
    border: none;
    border-radius: 10px;
    box-shadow: 0px 5px 5px rgba(0, 0, 0, 0.2);
}

.menu-user-info{
    padding: 10px 20px 10px;
    color: white;
}

.menu-user-avatar{
    height: 50px;
    width: 50px;
    border-radius: 25px;
}

.info-inline{
    display: inline-block;
    vertical-align: top;
}

.menu-user-item{
    display: block;
    width: 100%;
    padding: 0.25rem 1.5rem;
    clear: both;
    font-weight: 400;
    color: white;
    text-align: inherit;
    text-decoration: none;
    white-space: nowrap;
    background-color: transparent;
    border: 0;
}
```

```
.menu-user-item:hover{
    color: white;
    text-decoration: none;
}

.menu-user-icons{
    font-size: 20px;
    margin-right: 10px;
}
```

Б.29 Текст файлу postEdit.css

```
/* Post Creation */
.title-line{
    color: white;
    width: 100%;
    border-radius: 20px;
    background-color: black;
    border: none;
}

.title-line:hover{
    background-color: black;
    border: none;
    box-shadow: none;
}

.title-line:focus{
    color: white;
    background-color: black;
    border: none;
    box-shadow: none;
}

/* Categories */
.add-cat-button{
    font-size: 14px;
    color: grey;
    background: none;
    border: 1px solid grey;
    border-radius: 20px;
    padding-top: 5px;
    padding-bottom: 5px;
    padding-right: 10px;
}

.add-cat-button:hover{
    color: white;
    background: grey;
```

```
    transition: 0.1s linear;
}

.add-cat-button:focus{
    outline: none;
}

.category-dropdown{
    color: darkgrey;
    background-color: #333133;
    border: none;
    border-radius: 25px;
    box-shadow: none;
    padding-right: 10px;
    padding-left: 10px;
    padding-top: 5px;
    padding-bottom: 5px;
    appearance: none;
}

.category-dropdown:focus{
    border: none;
    outline: none;
    box-shadow: none;
}

.category-delete-button{
    color: white;
    background: none;
    outline: none;
    border: none;
}

.category-delete-button:focus{
    color: white;
    background: none;
    outline: none;
    border: none;
}

.category-box{
    border-radius: 25px;
    padding-top: 7px;
    padding-bottom: 7px;
    margin-right: 10px;
    background-color: #333133;
}

.reply-line{
    display: inline-block;
    margin-bottom: 10px;
    color: white;
```

```
    padding-left: 10px;
    background-color: black;
    border: none;
    border-radius: 20px;
    padding-bottom: 5px;
}

.reply-line:focus{
    color: white;
    background-color: black;
    border: none;
    box-shadow: none;
}

.reply-icon{
    font-size: 24px;
}

.reply-text{
    display: inline-block;
    margin-left: 10px;
}

.cancel-button{
    font-size: 18px;
    border: none;
    background: none;
    color: white;
}

.cancel-button:hover{
    background: none;
    color: white;
}

.cancel-button:focus{
    outline: none;
}
```

Б.30 Текст файлу postList.css

```
.posts-container{
    color: white;
    font-family: Calibri;
    font-size: 18px;
    background-color: #333133;
    width: 100%;
    border-radius: 12px;
    padding-top: 15px;
    padding-left: 15px;
```

```
    padding-right: 15px;
    margin-bottom: 30px;
    transition: 0.1s linear;
}

.posts-container:hover{
    background-color: #3a383a;
}

.title-font{
    color: white;
    font-size: 26px;
    font-family: Calibri;
    font-weight: bold;
    text-decoration: none;
}

.title-font:hover{
    color: white;
    text-decoration: none;
}

.info-box{
    position: absolute;
    right: 20px;
    bottom: 20px;
}

.post-info{
    color: white;
    font-family: Calibri;
    font-size: 16px;
    padding-right: 5px;
}

.post-list-author{
    color: white;
    text-decoration: none;
    font-weight: normal;
}

.post-list-author:hover{
    color: white;
    text-decoration: none;
    font-weight: normal;
}

.post-category-line{
    margin-bottom: 25px;
    margin-top: -10px;
}
```

```
.info-elem{
    padding-left: 10px;
}

.userName-box{
    position: absolute;
    left: 15px;
    bottom: 20px;
}

.threeDots{
    width: 40px;
    height:40px;
    position: absolute;
    right: 5px;
}

.threeDots-button{
    width: 100%;
    height: 100%;
    background: none;
    border: none;
    border-radius: 20px;
    color: white;
    transition: 0.1s linear;
}

.threeDots-button:hover{
    background-color: #4f4b4f;
    border: none;
    color: white;
}

.threeDots-button:focus{
    background-color: #4f4b4f;
    border: none;
    outline: none;
    color: white;
}

.dropdown-additions{
    background-color: #2b292b;
    border-radius: 10px;
}

.dropdown-item-additions{
    color: white;
}

.dropdown-item-additions:hover{
    color: white;
    background-color: #4f4b4f;
```

```
}

.dropdown-item-additions:focus{
    outline: none;
}

.indent{
    margin-right: 10px;
}
```

Б.31 Текст файлу postStyles.css

```
.post-block{
    color: white;
    font-family: Calibri;
    font-size: 18px;
    background-color: #333133;
    width: 100%;
    border-radius: 12px;
    padding-top: 15px;
    padding-bottom: 15px;
    padding-left: 15px;
    padding-right: 15px;
    margin-top: 30px;
    margin-bottom: 30px;
}

.blog-post-title{
    color: white;
    font-size: 26px;
    font-family: Calibri;
    font-weight: bold;
}

.blog-post-data{
    width: 100%;
    color: #878787;
    font-size: 12px;
    font-family: Calibri;
    font-weight: lighter;
    text-decoration: none;
}

.blog-post-info-elem{
    padding-right: 10px;
}

.blog-post-author{
    color: #878787;
    font-size: 12px;
```

```
    font-family: Calibri;
    font-weight: lighter;
    text-decoration: none;
}

.blog-post-author:hover{
    color: white;
    font-size: 12px;
    font-family: Calibri;
    font-weight: lighter;
    text-decoration: none;
}

.category{
    display: inline-block;
    text-align: center;
    font-weight: normal;
    font-size: 14px;
    color: black;
    background-color: white;
    border-radius: 9px;
    margin-right: 5px;
    padding-left: 10px;
    padding-right: 10px;
}

.category-text{
    color: black;
    text-decoration: none;
    font-weight: normal;
}

.category-text:hover{
    color: black;
    text-decoration: none;
    font-weight: normal;
}

.line{
    background-color: #b4b4b4;
    height: 1px;
    width: 100%;
    margin-top: 10px;
    margin-bottom: 20px;
}

.comment-box{
    color: white;
    font-family: Calibri;
    font-size: 18px;
    background-color: #333133;
    border-radius: 12px;
```

```
    padding-top: 20px;
    padding-bottom: 20px;
    padding-left: 30px;
    padding-right: 30px;
}

.comment-info{
    width: 100%;
    color: #878787;
    font-size: 12px;
    font-family: Calibri;
    font-weight: lighter;
    text-decoration: none;
}

.comment-author{
    display: inline-block;
    text-align: center;
    font-weight: normal;
    font-size: 14px;
    color: black;
    background-color: white;
    border-radius: 9px;
    margin-right: 5px;
    padding-left: 10px;
    padding-right: 10px;
    text-decoration: none;
}

.comment-author:hover{
    color: black;
    text-decoration: none;
}

.comment-as{
    color: white;
    text-decoration: none;
}

.comment-as:hover{
    color: #5e5e5e;
    text-decoration: none;
    transition: 0.1s linear;
}


.comment-interaction{
    font-size: 17px;
    font-family: Calibri;
    font-weight: lighter;
    color: white;
    text-decoration: none;
```

```
    background: none;
    border: none;
    margin-left: -5px;
}

.comment-interaction:hover{
    color: white;
    text-decoration: none;
    background-color: #2a2a2a;
    border-radius: 9px;
    text-align: center;
    border: none;
    transition: 0.2s linear;
}

.comment-interaction:focus{
    outline: 0;
}

.reply{
    color: #5e5e5e;
}

.answer-to{
    color: #5e5e5e;
}

.answer-to:hover{
    color: white;
    cursor: pointer;
}
```

Б.32 Текст файлу tinymceStyles.css

```
.dark-skin-editor{
    filter: invert(1);
}
```

Б.33 Текст файлу userList.css

```
.block{
    width: 25%;
    padding-bottom: 40px;
}

.user-info{
    margin-left: 10px;
    font-size: 14px;
```

```
}

.user-list-pos{
   margin-top: 30px;
}
```

Б.34 Текст файлу userLogo.css

```
.post-user-logo{
   height: 35px;
   width: 35px;
   border-radius: 20px;
}

.user-list-logo{
   height: 70px;
   width: 70px;
   border-radius: 35px;
}
```

Б.35 Текст файлу userProfileStyles.css

```
/* User profile */

.about-block{
   font-size: 20px;
   display: flex;
}

.title-margin{
   font-size: 16px;
   margin-left: 20px;
   margin-bottom: 10px;
   min-height: 24px;
}

.data-margin{
   font-size: 16px;
   padding-left: 25px;
   margin-bottom: 10px;
   min-height: 24px;
}

.user-control-button{
   color: white;
   font-size: 20px;
   margin-left: 10px;
}
```

```
.user-control-button:hover{
    color: white;
}

.user-control-panel{
    float: right;
}

/* Comments */
.created-at{
    color: #878787;
    font-size: 14px;
    font-family: Calibri;
    font-weight: lighter;
    text-decoration: none;
    margin: 0;
}

.on-post{
    font-style: italic;
    margin-bottom: 15px;
}

/** Button Active **/
.is-button-active{
    background-color: #474747;
}

/** Button Disabled **/
.is-button-disabled{
    background-color: #333133;
}

/** Button Default **/
.is-button-default{
    font-size: 18px;
    color: white;
    border-radius: 30px;
    padding-top: 10px;
    padding-bottom: 10px;
    padding-left: 15px;
    padding-right: 15px;
    text-decoration: none;
}

.is-button-default:hover{
    color: white;
    background-color: #474747;
    text-decoration: none;
    transition: 0.1s linear;
}
```

```
/* Bookmarks page */
.back-to-profile-button{
    width: 10%;
    color: white;
    background-color: #333133;
    text-align: center;
    padding-top: 7px;
    padding-left: 10px;
    padding-right: 10px;
    padding-bottom: 7px;
    border-radius: 20px;
    text-decoration: none;
    margin-bottom: 10px;
}

.back-to-profile-button:hover{
    background-color: #3a383a;
    transition: 0.1s linear;
}

.bookmarks-delete-button{
    width: 88%;
    color: white;
    background-color: #333133;
    text-align: center;
    padding-top: 5px;
    padding-bottom: 5px;
    border-radius: 20px;
    text-decoration: none;
    border: none;
    margin-bottom: 10px;
    margin-top: 15px;
}

.bookmarks-delete-button:hover{
    background-color: #3a383a;
    transition: 0.1s linear;
}

.bookmarks-delete-button:focus{
    outline: none;
}


/* Initially hides the delete button in edit if there is no picture */
.hide-delete-photo-button{
    display: none;
}


/* User profile edit page */
```

```
.user-avatar{
    width: 150px;
    height: 150px;
    object-fit: cover;
    object-position: 50% 50%;
    border-radius: 100px;
    margin-right: 10px;
    margin-left: 5px;
}

.avatar-block{
    display: flex;
}

#photoInput{
    position: absolute;
    left: -9999px;
}

.input-photo{
    font-size: 14px;
    color: grey;
    background: none;
    border: 1px solid grey;
    border-radius: 20px;
    padding-top: 5px;
    padding-bottom: 5px;
    padding-right: 10px;
    padding-left: 10px;
    margin-top: 10px;
    margin-bottom: 10px;
}

.input-photo:hover{
    color: white;
    background-color: grey;
    transition: 0.1s linear;
}

.input-photo:focus{
    outline: none;
}

.delete-photo{
    color: grey;
    border: none;
    background: none;
    text-align: center;
}

.delete-photo:focus{
    color: white;
```

```
    outline: none;
}

.usr-profile-input{
    color: white;
    background-color: black;
    border: none;
    border-radius: 20px;
}

.usr-profile-input:focus{
    color: white;
    background-color: black;
    border: none;
    box-shadow: none;
}
```

Б.36 Текст файлу categoryStyles.css

```
/* index */
.category-control-panel{
    float: right;
}

.category-control-button{
    color: white;
    font-size: 20px;
    margin-left: 10px;
}

.category-control-button:hover{
    color: white;
}

.category-block{
    margin-left: 10%;
}

.category-card{
    background-color: black;
    border-radius: 20px;
    margin-right: 20px;
    margin-bottom: 20px;
}

.main-category{
    width: 200px;
    height: 70px;
    text-align: center;
    margin-top: 20px;
```

```
    margin-left: 40px;
    margin-right: 40px;
    margin-bottom: 10px;
}

.main-category-text{
    color: white;
    font-family: Calibri;
    font-weight: bold;
    line-height: 70px;
    font-size: 3.5vw;
    word-wrap: break-word;
}

.main-category-text:hover{
    color: white;
    text-decoration: none;
}

.heirs{
    margin-left: 40px;
    margin-right: 40px;
    margin-bottom: 20px;
}

.heirs-block{
    margin-top: 20px;
}

.heirs-text{
    color: white;
    font-family: Calibri;
    font-weight: normal;
    font-size: 26px;
    word-wrap: break-word;
}

.heirs-text:hover{
    color: white;
    text-decoration: none;
    font-family: Calibri;
    font-weight: normal;
    font-size: 26px;
    word-wrap: break-word;
}

.no-categories{
    color: #5e5e5e;
    font-size: 20px;
    text-align: center;
}
```

```
/* create */

.cat-name-input{
    color: white;
    background-color: black;
    border: none;
    border-radius: 20px;
    width: 30%;
}

.cat-name-input:focus{
    color: white;
    background-color: black;
    border: none;
    box-shadow: none;
}

.heirs-dropdown{
    width: 30%;
    color: white;
    background-color: black;
    border: none;
    box-shadow: none;
    border-radius: 20px;
}

.heirs-dropdown:focus{
    border: none;
    outline: none;
    box-shadow: none;
}

.heirs-dropdown-value:after{
    border-radius: 20px;
    border: none;
    box-shadow: none;
}

/* control panel */
.cat-brick{
    color: white;
    font-size: 16px;
    text-align: center;
    background-color: black;
    border-radius: 10px;
    padding-top: 10px;
    padding-bottom: 10px;
    padding-left: 15px;
    padding-right: 15px;
    margin-right: 10px;
    margin-bottom: 10px;
}
```

```
.cat-brick:hover{
    color: white;
    text-decoration: none;
}


.cat-list{
    margin-left: 30px;
}


.cat-delete{
    font-size: 20px;
    font-family: Calibri;
    font-weight: lighter;
    color: white;
    text-decoration: none;
    background: none;
    border: none;
    margin-left: 5px;
}

.cat-delete:hover{
    color: white;
    text-decoration: none;
    color: white;
    background-color: #2a2a2a;
    border-radius: 9px;
    text-align: center;
    border: none;
    transition: 0.2s linear;
}

.cat-delete:focus{
    outline: 0;
}

.form-box{
    height: 85px;
}

.cat-trash{
    font-size: 17px;
    font-family: Calibri;
    color: white;
    text-decoration: none;
    background: none;
    border: none;
}

.cat-trash:hover{
```

```
    color: white;
    text-decoration: none;
    background-color: #2a2a2a;
    border-radius: 9px;
    transition: 0.2s linear;
}

.cat-trash:focus{
    outline: 0;
}

.cat-trash-button{
    margin-left: 130px;
}
```

Б.37 Текст файлу descriptionStyles.css

```
.description-box{
    max-width: 100%;
}

.description-bg{
    background: url("../img/description-img.jpg");
    background-size: cover;
    height: 48vw;
}

.container-fluid{
    padding: 0;
}

/* DOT container with two greeting lines*/
.description-slider-box{
    text-align: center;
    max-width: 20%;
    display: block;
    padding-top: 15%;
    margin-left: auto;
    margin-right: 5%;
}

.description-title-text{
    font-family: "Artifakt Element";
    font-weight: bold;
    font-size: 10vw;
    max-height: 12vw;
}

.description-text{
    font-family: "Artifakt Element";
```

```
    font-weight: lighter;
    font-size: 1.15vw;
}

.slider-line{
    border-bottom: 3px solid #2f2f2f;
    width: 48%;
    float: left;
    margin-right: 1%;
    margin-left: 1%;
    margin-bottom: 2%;
}

.slider-line-active{
    border-bottom: 3px solid white;
}

.about-info-block{
    margin-top: 75px;
    margin-bottom: 100px;
}

.about-img{
    width: 270px;
    height: 270px;
    object-fit: cover;
    object-position: 50% 50%;
}

.about-right{
    text-align: right;
}


.about-line-left{
    width: 65%;
    margin-left: 26.25%;
}

.about-left{
    margin-right: 20px;
    float: left;
}
```

Б.38 Текст файлу filter.css

```
.filter-button{
    font-size: 20px;
    color: white;
    border: none;
```

```
    background: none;
    border-radius: 10px;
    padding-top: 10px;
    padding-bottom: 10px;
    padding-left: 15px;
    padding-right: 15px;
    margin-left: -20px;
}

.filter-button:hover{
    font-size: 20px;
    color: white;
    background-color: #333133;
    border-radius: 10px;
    padding-top: 10px;
    padding-bottom: 10px;
    padding-left: 15px;
    padding-right: 15px;
    margin-left: -20px;
    transition: .1s linear;
}

.filter-button:focus{
    outline: none;
}

.filter-box{
    background: none;
    margin-top: 20px;
    margin-bottom: -20px;
    width: 100%;
}

.filter-header{
    background: none;
    border-bottom: none;
}

.filter-block{
    display: inline-grid;
    margin-bottom: 40px;
    margin-right: 30px;
}

.filter-category{
    color: #878787;
    font-weight: normal;
    background: none;
    border: none;
    outline: none;
}
```

```
.filter-category:hover{
    color: white;
    background: none;
    border: none;
    outline: none;
    text-decoration: none;
    transition: .1s linear;
}

.filter-category:focus{
    color: white;
    background: none;
    border: none;
    outline: none;
}

.filter-category-active{
    color: white;
    background: none;
    border: none;
    outline: none;
}
```

Б.39 Текст файлу footerStyles.css

```
.footer-block{
    background-color: black;
    position: relative;
}

.content{
    padding-top: 25px;
    color: white;
    text-align: center;
}

.centered-content{
    display: flex;
}

.footer-links{
    color: white;
    text-decoration: none;
    padding-right: 10px;
}

.footer-links:hover{
    color: white;
    text-decoration: none;
}
```

```
.footer-logo{
    font-family: "Artifakt Element Medium";
    color: white;
    font-size: 40px;
    text-align: center;
}

.social-media{
    font-size: 24px;
    margin-bottom: 10px;
}

/* interactions */
.footer-interactions{
    margin-top: 15px;
    margin-bottom: 10px;
}


/* Social Media */

.twitter{
    color: #1da1f2;
}

.twitter:hover{
    color: #1da1f2;
}

.facebook{
    color: #1877f2;
}

.facebook:hover{
    color: #1877f2;
}

.telegram{
    color: #1da1f2;
}

.telegram:hover{
    color: #1da1f2;
}

.instagram{
    color: #fff;
}

.instagram:hover{
    color: #fff;
```

```
}

.youtube{
   color: #ff0000;
}

.youtube:hover{
   color: #ff0000;
}

/* Copyright */
.copyrighting{
   color: darkgrey;
   margin-bottom: 20px;
}
```

Б.40 Текст файлу logRegStyles.css

```
/* Left block */
.left-block{
   background-image: url("../img/logReg.png");
   background-size: cover;
   background-position: center;
   padding: 0;
   height: 93vh;
}

.dot-logo{
   font-family: "Artifakt Element";
   font-size: 10vw;
   font-weight: bold;
   margin-top: 25%;
   text-align: center;
}

.dot-description{
   font-family: "Artifakt Element";
   font-size: 2vw;
   font-weight: lighter;
   text-align: center;
   margin-top: -5%;
}

/* Right block */

.right-block{
   padding: 0;
}

.input-block{
```

```
    left: 25%;
    right: 25%;
    top: 25%;
    background: none;
    border: none;
    padding-top: 35%;
    padding-left: 20%;
    padding-right: 20%;
}

.input-line{
    color: white;
    width: 100%;
    margin-top: 20px;
    border-radius: 20px;
    background-color: black;
    border: none;
}

.input-line:hover{
    background-color: black;
    border: none;
    box-shadow: none;
}

.input-line:focus{
    color: white;
    background-color: black;
    border: none;
    box-shadow: none;
}

.login-margin{
    font-family: "Artifakt Element";
    font-weight: normal;
    margin-top: 2%;
    text-align: center;
}

.interaction-margin{
    display: flex;
    justify-content: center;
}
```

Б.41 Текст файлу mainStyles.css

```
input:-webkit-autofill,
input:-webkit-autofill:hover,
input:-webkit-autofill:focus{
    -webkit-text-fill-color: #fff;
```

```
    -webkit-box-shadow: 0 0 0px 1000px #000000 inset;
    transition: background-color 5000s ease-in-out 0s;
}

html, body {
    height: 100%;
}

#mainContainer {
    min-height: 80vh;
}

.page-colors{
    background-color: #181818;
    color: white;
}

.no-link-decor:hover{
    text-decoration: none;
}

/* Pagination */
.pagination>li>a, .pagination>li>span {
    color: white;
    margin-left: 5px;
    background-color: #202020;
    border: none;
    border-radius: 10px;
    transition: 0.1s linear;
}

.pagination>li>a:hover, .pagination>li>span:hover, .pagination>li>a:focus,
.pagination>li>span:focus {
    color: #fff;
    background-color: #333133;
    border: none;
}

.pagination>.disabled>.page-link{
    background-color: #202020;
}

.pagination>.active>.page-link{
    background-color: #333133;
}

.page-link:focus{
    box-shadow: none;
}

.page-item:last-child .page-link {
    border-top-right-radius: 0;
```

```
    border-bottom-right-radius: 0;
    border-radius: 10px;
}

.page-item:first-child .page-link {
    border-top-left-radius: 0;
    border-bottom-left-radius: 0;
    border-radius: 10px;
}

.pagination-area{
    width: 100%;
    text-align: center;
}

.pagination-box{
    display: inline-block;
}

/* error */
.error-message{
    color: #565e64;
    font-size: 120px;
    width: 100%;
    height: 100%;
    padding-top: 25%;
    text-align: center;
}

/* Modal confirm window */
@keyframes fade-in {
    from { opacity: 0; }
    to { opacity: 1; }
}

.blur-bg{
    opacity: 0;
    animation-name: fade-in;
    animation-duration: 0.3s;
    animation-fill-mode: forwards;
    backdrop-filter: blur(2px);
    transition: 0.05s linear;
}

.blur-bg:focus{
    opacity: 0;
    animation-name: fade-in;
    animation-duration: 0.3s;
    animation-fill-mode: forwards;
    backdrop-filter: blur(2px);
    transition: 0.05s linear;
}
```

```
.mod-win-head{
    display: flex;
    flex-shrink: 0;
    align-items: center;
    justify-content: space-between;
    padding-left: 25px;
    padding-top: 5px;
    border-top-left-radius: 10px;
    border-top-right-radius: 10px;
    background-color: #202020;
}

.mod-win-center{
    background-color: #202020;
    padding-left: 25px;
    border-top: 1px solid white;
}

.mod-win-foot{
    display: flex;
    flex-shrink: 0;
    flex-wrap: wrap;
    align-items: center;
    justify-content: flex-end;
    padding: 1em;
    border-bottom-left-radius: 10px;
    border-bottom-right-radius: 10px;
    background-color: #202020;
}

.mod-win-body{
    font-family: Calibri;
    font-size: 18px;
    background: none;
    position: relative;
    display: flex;
    flex-direction: column;
    width: 100%;
    color: white;
    pointer-events: auto;
    background-clip: padding-box;
    outline: 0;
}

.mod-win-button-close{
    color: white;
    margin-right: 15px;
    background: none;
    outline: none;
    border: none;
}
```

```
.mod-win-button-close:focus{
    background: none;
    outline: none;
    border: none;
}

.mod-win-head-text{
    padding-bottom: 5px;
    font-family: Calibri;
}

.mod-win-addit-text{
    margin-top: 30px;
    font-size: 13px;
    color: grey;
    font-family: Calibri;
}

/* Button Apply */
.button-apply{
    font-family: Calibri;
    font-weight: bold;
    font-size: 21px;
    background: none;
    color: white;
    border: 1px solid white;
    border-radius: 20px;
    padding-left: 1em;
    padding-right: 1em;
    padding-top: 0.15em;
    padding-bottom: 0.15em;
    transition: 0.1s linear;
    text-decoration: none;
}

.button-apply:hover{
    font-family: Calibri;
    font-weight: bold;
    font-size: 21px;
    background: white;
    color: black;
    border: 1px solid white;
    border-radius: 20px;
    padding-left: 1em;
    padding-right: 1em;
    padding-top: 0.15em;
    padding-bottom: 0.15em;
    transition: 0.1s linear;
    text-decoration: none;
}
```

```
.button-apply:focus{
    outline: none;
}

/* Close Button */
.close-button{
    background: none;
    outline: none;
    border: none;
    color: white;
    font-family: Calibri;
    font-weight: bold;
    font-size: 20px;
    margin-right: 15px;
}

.close-button:focus{
    background: none;
    outline: none;
    border: none;
}

/* error block */
.error-block-message{
    color: white;
    background-color: black;
    border: 1.5px solid #dc3545;
}
```

Б.42 Текст файлу logicAuth.js

```
window.addEventListener('resize', updateAuthStyles);

const imgWith = 1040, mintWidthRightMenu = 200;

// Функція автоприховання фонового зображення при зміні розмірів вікна
function updateAuthStyles() {
    let rightBlock = document.getElementById("rightBlock");
    if(window.innerWidth - imgWith < mintWidthRightMenu) {
        document.getElementById("authImg").style.display = "none";
        document.getElementById("rightMenu").className = "";
        rightBlock.className = "row justify-content-center";
        rightBlock.style.padding = "10%";
        rightBlock.style.paddingTop = "30%";
    } else {
        document.getElementById("authImg").style.display = "block";
        document.getElementById("rightMenu").className = "input-block";
        rightBlock.className = "row";
        rightBlock.style.padding = "0";
    }
```

```
}
```

Б.43 Текст файлу logicCategories.js

```
window.onload = function() {
   setFontsizeForMainCategory();
};

// Функція автоматично встановлює розмір тексту основних категорії в залежності від довжини тексту
function setFontsizeForMainCategory(){
   const links = document.querySelectorAll('a.main-category-text');
   let fSize, maxSize = 240, koefSmallText = 0.6, minSymbols = 9;
   links.forEach(link => {
      if(link.textContent.length < minSymbols){
         fSize = maxSize / link.textContent.length;
      } else {
         fSize = maxSize / (koefSmallText * link.textContent.length);
      }
      link.style.fontSize = '' + String(fSize) + 'px';
   });
}
```

Б.44 Текст файлу logicDescription.js

```
// Функція автоматично замінює Bootstrap-контейнер на його розширений аналог
window.onload = function () {
   var obj = document.getElementById('mainContainer');
   obj.classList.remove("container");
   obj.classList.add("container-fluid");
   var objFixFooter = document.getElementById('footerRow');
   objFixFooter.classList.remove("row");
   objFixFooter.classList.add("centered-content");
}
```

Б.45 Текст файлу logicPost.js

```
// Функція встановлення html-елементу тексту коментаря без html коду
function cutHTML() {
   document.getElementById('TextWithoutHTML').value = tinyMCE.activeEditor.getBody().textContent;
};

// Функція видалення html-блоку відповіді на коментар
function deleteReplyOnComment(){
   document.getElementById('replyMessageDiv').remove();
```

```
        }

        // Функція створення html-блоку відповіді на коментар
        function setReplyOnComment(intro, id){
            if(document.getElementById('replyMessageDiv') !== null) {
                deleteReplyOnComment();
            }

            var obj = document.createElement('div');
            obj.id = "replyMessageDiv";
            obj.classList.add('reply-line');

            obj.innerHTML = '<i class="bi bi-reply-fill reply-icon"></i>' +
                    '<div class="reply-text">У відповідь на: ' + intro + '</div>' +
                    '<button type="button" class="cancel-button" aria-label="Close"
onClick="deleteReplyOnComment()">' +
                    '<i class="bi bi-x"></i></button><input type="hidden"
name="reply_message_id" value=' + id + '>' +
                    '<input type="hidden" name="reply_message_text" value="' + intro + '">';
            document.getElementById('replyMessagePoint').after(obj);

            smoothScroll('replyMessagePoint');
        };

        // Функція редагування коментаря
        function editComment(intro, id, htmlText, reply_message_id, reply_message_text,
post_id) {
            tinymce.activeEditor.setContent(htmlText);
            if (reply_message_text != '') {
                setReplyOnComment(reply_message_text, reply_message_id);
            }
            SendToSaveButtons(id, post_id);

            let input = document.createElement("input");
            input.type="hidden";
            input.name="_method";
            input.value="patch";
            document.getElementById('comment-form').append(input);

            smoothScroll('replyMessagePoint');
        }

        // Функція заміни кнопки "Відправити" кнопкою "Зберегти"
        function SendToSaveButtons(comment_id, post_id) {
            var SendButton = document.getElementById('send-button');
            SendButton.innerHTML = "Зберегти";
            SendButton.onclick = function () {
                saveEditedComment(comment_id,post_id);
            }
        }

        // Функція збереження відредагованного коментаря
```

```
        function saveEditedComment(comment_id, post_id) {
            cutHTML();
            document.getElementById('comment-form').action = '/posts/' + post_id.toString()+
'/comment/' + comment_id.toString();
        }


        // Визначає поточне положення скролу
        function currentYPosition() {
            // Firefox, Chrome, Opera, Safari
            if (self.pageYOffset) return self.pageYOffset;
            // Internet Explorer 6 - standards mode
            if (document.documentElement && document.documentElement.scrollTop)
                return document.documentElement.scrollTop;
            // Internet Explorer 6, 7 and 8
            if (document.body.scrollTop) return document.body.scrollTop;
            return 0;
        }

        // Визначає положення елемента
        function elmYPosition(eID) {
            var elm = document.getElementById(eID);
            var y = elm.offsetTop;
            var node = elm;
            while (node.offsetParent && node.offsetParent != document.body) {
                node = node.offsetParent;
                y += node.offsetTop;
            } return y;
        }

        // Функція скролу до якоря
        function smoothScroll(eID) {
            var startY = currentYPosition();
            var stopY = elmYPosition(eID);
            var distance = stopY > startY ? stopY - startY : startY - stopY;
            if (distance < 100) {
                scrollTo(0, stopY); return;
            }
            //var speed = Math.round(distance / 100);
            //if (speed >= 20) speed = 20;
            speed = 10;
            var step = Math.round(distance / 25);
            var leapY = stopY > startY ? startY + step : startY - step;
            var timer = 0;
            if (stopY > startY) {
                for ( var i=startY; i<stopY; i+=step ) {
                    setTimeout("window.scrollTo(0, "+leapY+")", timer * speed);
                    leapY += step; if (leapY > stopY) leapY = stopY; timer++;
                } return;
            }
            for ( var i=startY; i>stopY; i-=step ) {
                setTimeout("window.scrollTo(0, "+leapY+")", timer * speed);
```

```
        leapY -= step; if (leapY < stopY) leapY = stopY; timer++;
    }
}

let categoriesName = [];
let categoriesId = [];
let categoriesInputCounter = 0;

// Функція встановлення категорій
function setCategories(name, id) {
    categoriesName.push(name);
    categoriesId.push(id);
}

// Функція створення полей вводу категорій
function createCategoriesInput(){
    let containerId = "categ_" + String(categoriesInputCounter);
    let selectElement = document.createElement('select');
    selectElement.name = "category_" + String(categoriesInputCounter);
    selectElement.id = "category_" + String(categoriesInputCounter);
    selectElement.classList = "category-dropdown";

    for (let i = 0; i < categoriesName.length; i++) {
        const optionElement = document.createElement("option");
        optionElement.value = categoriesId[i];
        optionElement.text = categoriesName[i];
        selectElement.appendChild(optionElement);
    }

    const buttonElement = document.createElement("button");
    buttonElement.innerHTML = '<i class="bi bi-x"></i>';
    buttonElement.classList = "category-delete-button";
    buttonElement.addEventListener("click", (event) => {
        event.preventDefault();
        document.getElementById(containerId).remove();
    });

    let containerElement = document.createElement("div");
    containerElement.id = containerId;
    containerElement.appendChild(selectElement);
    containerElement.appendChild(buttonElement);
    containerElement.classList = "d-inline category-box";

    document.getElementById('insertInputPlace').after(containerElement);


    categoriesInputCounter = categoriesInputCounter + 1;
}

// Функція автоматично додає категорії, йкщо такі є
function setOldCategoriesInput(id){
    createCategoriesInput();
```

```
    let selectId = "category_" + String(categoriesInputCounter-1);

    let selectElement = document.getElementById(selectId);
    let options = selectElement.options;
    for (let i = 0; i < options.length; i++) {
        if (options[i].value === id) {
            options[i].selected = true;
        }
    }
}
```

Б.46 Текст файлу logicPostsPage.js

```
// Функція вбудови до посилання критерію сортування
function setSortingFilter(e, param, value) {
    e.href = setUrlParam(param, value);
};

// Функція вбудови параметрів сортування до посилання
function setUrlParam(param, value) {
    let newURL = window.location.protocol + '//' + window.location.host + window.location.pathname + '?';

    let url = decodeURIComponent(window.location.search.substring(1)),
        urlVar = url.split('&'),
        flag = false,
        params;

    for (let i = 0; i < urlVar.length; i++) {
        params = urlVar[i].split('=');

        if (params[0] === param) {
            params[1] = value;
            flag = true;
        }

        if (params[0]) {
            newURL = newURL + params[0] + '=' + params[1] + '&'
        }
    }

    if (flag == false) {
        newURL = newURL + param + '=' + value
    }

    return newURL
};
```

```
// Функція автоматично встановлює останньо-вибраний користувачем фільтр пошуку (за типом)
function setSortTypeText(sorting) {
  let id;
  switch (sorting) {
    case "rating":
      id = 'type1';
      break;
    case "amount_views":
      id = 'type2';
      break;
    case "created_at":
      id = 'type3';
      break;
  }
  document.getElementById(id).classList.toggle("filter-category-active");
}

// Функція автоматично встановлює останньо-вибраний користувачем фільтр пошуку (за зростанням)
function setSortGrowthText(sorting) {
  let id;
  switch (sorting) {
    case "desc":
      id = 'growth1';
      break;
    case "asc":
      id = 'growth2';
      break;
  }
  document.getElementById(id).classList.toggle("filter-category-active");
}
```

Б.47 Текст файлу logicUserEditPage.js

```
// Функція автоматично спрацьовує при завантаженні сторінки
window.onload = function () {
  document.getElementById('photoInput').addEventListener('change', handleFileSelect, false);
  setEventForPhotoUpload();
}

// Функція видалення фотографії профілю
function deletePhoto() {
  document.getElementById('flagPhoto').value = "deleted";
  document.getElementById("newPhoto").src = "../../src/img/default_user_photo.png";
  document.getElementById('deletePhotoButton').style.display  = 'none';
}

// Функція виведення превью нового фото профілю
```

```
function handleFileSelect(event) {
    var files = event.target.files;
    var filePhoto = files[0];
    if (filePhoto.type.match('image.*')) {
        document.getElementById('deletePhotoButton').style.display  = "block";
        var reader = new FileReader();
        reader.onload = (function(theFile) {
            return function(e) {
                document.getElementById('newPhoto').src = e.target.result;
            };
        })(filePhoto);
        reader.readAsDataURL(filePhoto);
    } else {
        alert("Неправильний тип файлу. Допустимі формати: jpg, png, bmp.");
    }
}

// Функція зв'язує кастомну кнопку з полем вводу файлу через подію кліку
function setEventForPhotoUpload() {
    document.getElementById('customPhotoInput').addEventListener('click', () => {
        document.getElementById('photoInput').click();
    });
}
```

Б.48 Текст файлу tinymceCore.js

```
tinymce.init({
    selector: '#tinymceForPostCreator',
    plugins: 'anchor help visualblocks visualchars preview link codesample table charmap emoticons hr insertdatetime lists formatpainter directionality',
    contextmenu: 'link',
    toolbar: 'undo redo | preview link codesample charmap emoticons anchor | table | bold italic underline strikethrough fontsize | align | outdent indent | numlist bullist formatpainter removeformat ltr rtl',
    menubar: 'edit view insert format table help forDev',
    content_css: "/src/css/tinymceStyles.css",
    menu: {
        edit: { title: 'Edit', items: 'undo redo | cut copy paste pastetext | selectall | searchreplace' },
        view: { title: 'View', items: 'visualaid visualchars visualblocks | preview' },
        insert: { title: 'Insert', items: 'link codesample | charmap emoticons hr anchor insertdatetime' },
        format: { title: 'Format', items: 'bold italic underline strikethrough superscript subscript codeformat | styles fontsize align lineheight | language | removeformat' },
        table: { title: 'Table', items: 'inserttable | cell row column | advtablesort | tableprops deletetable' },
        help: { title: 'Help', items: 'help' }
    },
    style_formats: [
        { title: 'Inline', items: [
```

```
                    { title: 'Bold', format: 'bold' },
                    { title: 'Italic', format: 'italic' },
                    { title: 'Underline', format: 'underline' },
                    { title: 'Strikethrough', format: 'strikethrough' },
                    { title: 'Superscript', format: 'superscript' },
                    { title: 'Subscript', format: 'subscript' }
                ]},
                { title: 'Blocks', items: [
                    { title: 'Paragraph', format: 'p' },
                    { title: 'Blockquote', format: 'blockquote' },
                    { title: 'Div', format: 'div' },
                    { title: 'Pre', format: 'pre' }
                ]},
                { title: 'Align', items: [
                    { title: 'Left', format: 'alignleft' },
                    { title: 'Center', format: 'aligncenter' },
                    { title: 'Right', format: 'alignright' },
                    { title: 'Justify', format: 'alignjustify' }
                ]}
            ]
        });

        tinymce.init({
            selector: '#tinymceForAdmin',
            plugins: 'anchor help visualblocks visualchars preview link codesample table charmap emoticons hr insertdatetime lists formatpainter directionality',
            contextmenu: 'link',
            toolbar: 'undo redo preview link codesample charmap emoticons anchor table bold italic underline strikethrough fontsize align outdent indent numlist bullist formatpainter removeformat ltr rtl fontfamily forecolor backcolor',
            menubar: 'edit view insert format table help forDev',
            content_css: "/src/css/tinymceStyles.css",
            menu: {
                edit: { title: 'Edit', items: 'undo redo | cut copy paste pastetext | selectall | searchreplace' },
                view: { title: 'View', items: 'visualaid visualchars visualblocks | preview' },
                insert: { title: 'Insert', items: 'link codesample | charmap emoticons hr anchor insertdatetime' },
                format: { title: 'Format', items: 'bold italic underline strikethrough superscript subscript codeformat | styles fontsize align lineheight | language | forecolor backcolor | removeformat' },
                table: { title: 'Table', items: 'inserttable | cell row column | advtablesort | tableprops deletetable' },
                help: { title: 'Help', items: 'help' }
            },
            style_formats: [
                { title: 'Inline', items: [
                    { title: 'Bold', format: 'bold' },
                    { title: 'Italic', format: 'italic' },
                    { title: 'Underline', format: 'underline' },
                    { title: 'Strikethrough', format: 'strikethrough' },
                    { title: 'Superscript', format: 'superscript' },
```

```
                    { title: 'Subscript', format: 'subscript' }
                ]},
            { title: 'Blocks', items: [
                    { title: 'Paragraph', format: 'p' },
                    { title: 'Blockquote', format: 'blockquote' },
                    { title: 'Div', format: 'div' },
                    { title: 'Pre', format: 'pre' }
                ]},
            { title: 'Align', items: [
                    { title: 'Left', format: 'alignleft' },
                    { title: 'Center', format: 'aligncenter' },
                    { title: 'Right', format: 'alignright' },
                    { title: 'Justify', format: 'alignjustify' }
                ]}
        ]
    });

    tinymce.init({
        selector: '#tinymceForCommentator',
        plugins: 'visualblocks visualchars link codesample charmap emoticons hr insertdatetime lists formatpainter directionality',
        contextmenu: 'link',
        toolbar: 'undo redo | link codesample charmap emoticons bold italic underline strikethrough | align | outdent indent | numlist bullist formatpainter removeformat ltr rtl',
        menubar: false,
        content_css: "/src/css/tinymceStyles.css"
    });
```

ДОДАТОК В

Слайди презентації

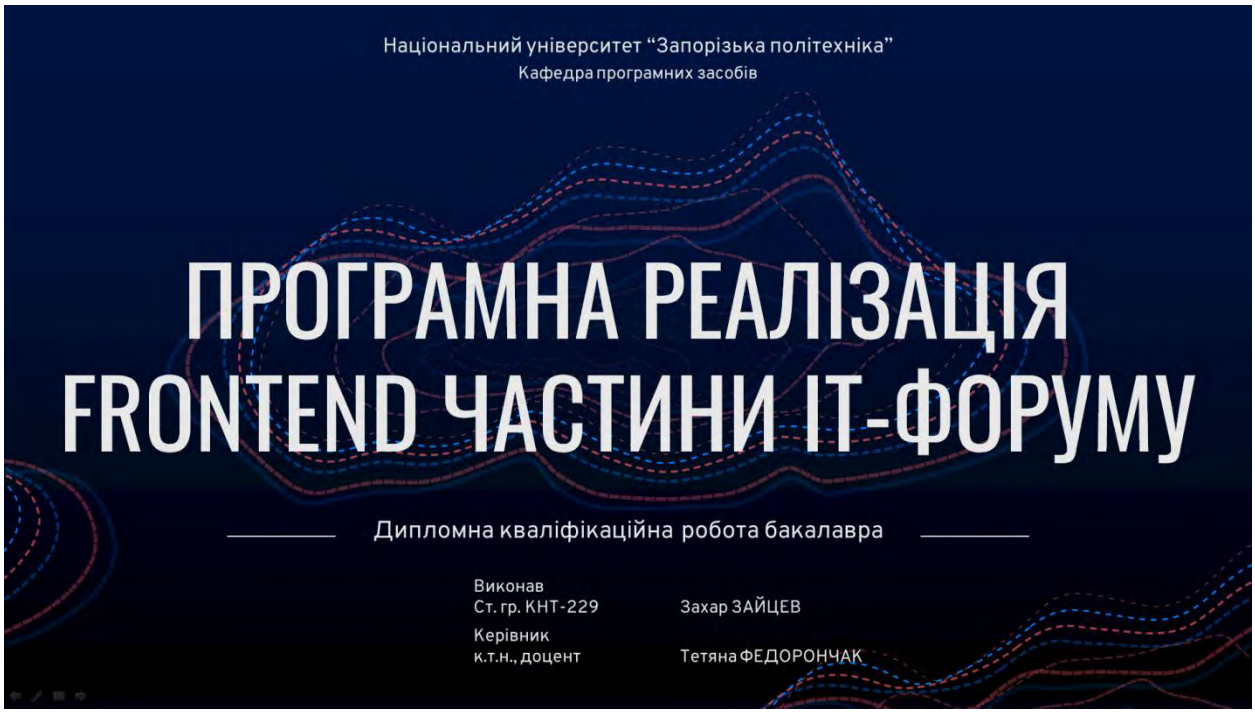

Рисунок В.1 – Титульний слайд

Рисунок В.2 – Слайд об’єкту, предмету та мети роботи

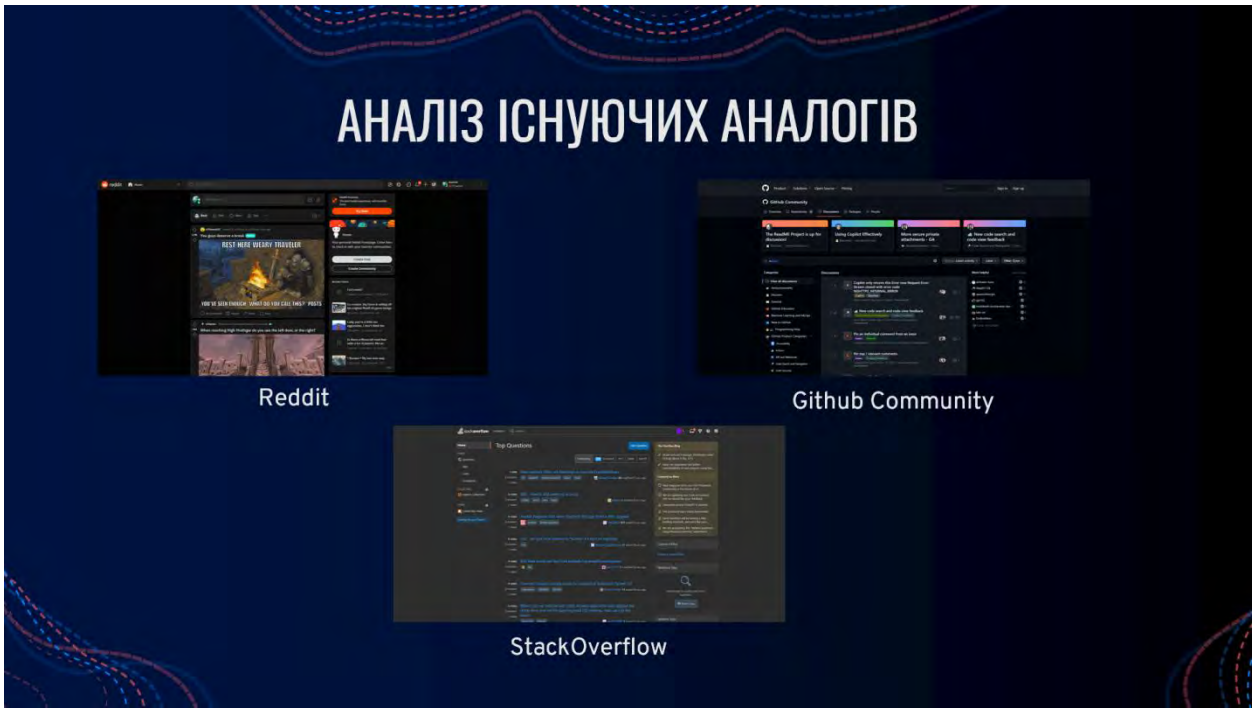

Рисунок В.3 – Слайд аналізу аналогів програмного забезпечення

	HTML	CSS	JAVASCRIPT
ВИКОРИСТАННЯ	Основна мова для створення структури та контенту веб-сторінок	Основна мова для стилізації та вигляду веб-сторінок	Основна мова для програмування веб-сторінок та взаємодії з користувачем
ФУНКЦІОНАЛЬНІ МОЖЛИВОСТІ	Визначення структури, розмітки та елементів веб-сторінок	Задання стилів, дизайну та вигляду веб-сторінок	Керування поведінкою, взаємодією з користувачем та обробкою подій
ПІДТРИМКА БРАУЗЕРАМИ	Повна підтримка всіма сучасними веб-браузерами	Повна підтримка всіма сучасними веб-браузерами	Повна підтримка всіма сучасними веб-браузерами
ВІДНОСНА СКЛАДНІСТЬ	Найбільш доступна та проста для вивчення	Середня складність, вимагає знань основ веб-розмітки	Середня складність, вимагає знань основ програмування

Рисунок В.4 – Слайд визначених мов програмування

Рисунок В.5 – Слайд аналізу існуючих фреймворків

Рисунок В.6 – Слайд схематичної структури сайту

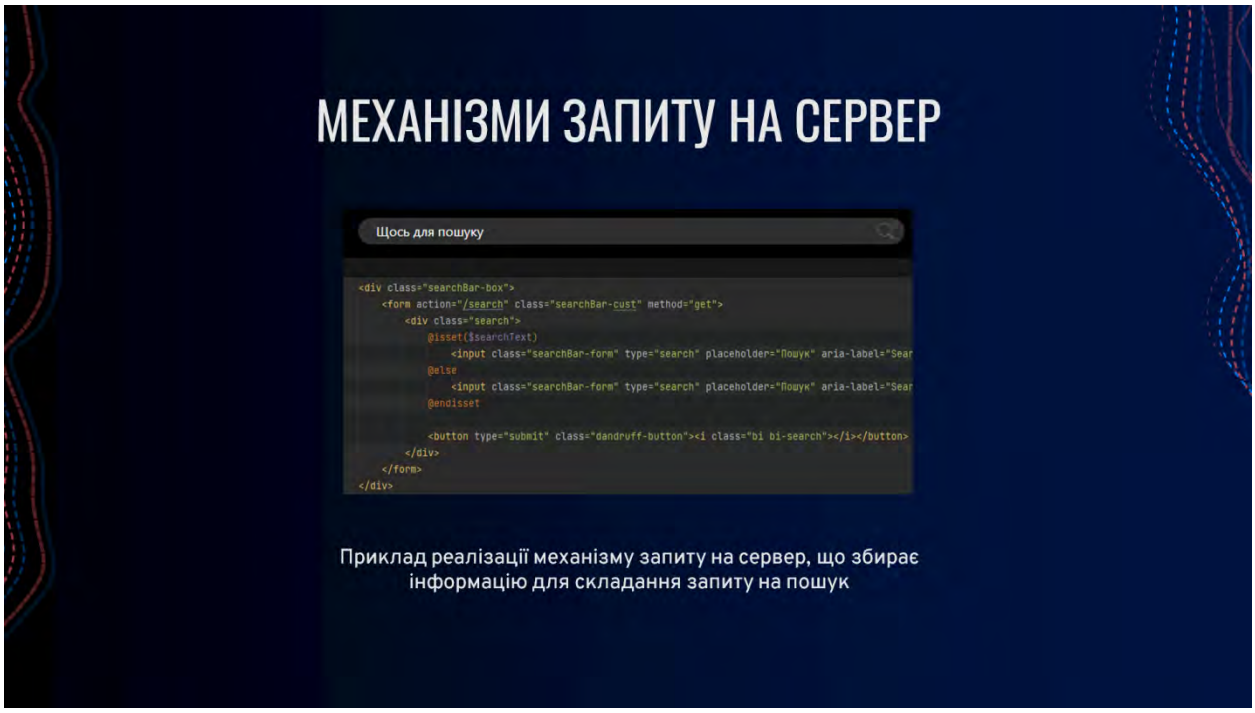

Рисунок В.7 – Слайд прикладу механізму запитів

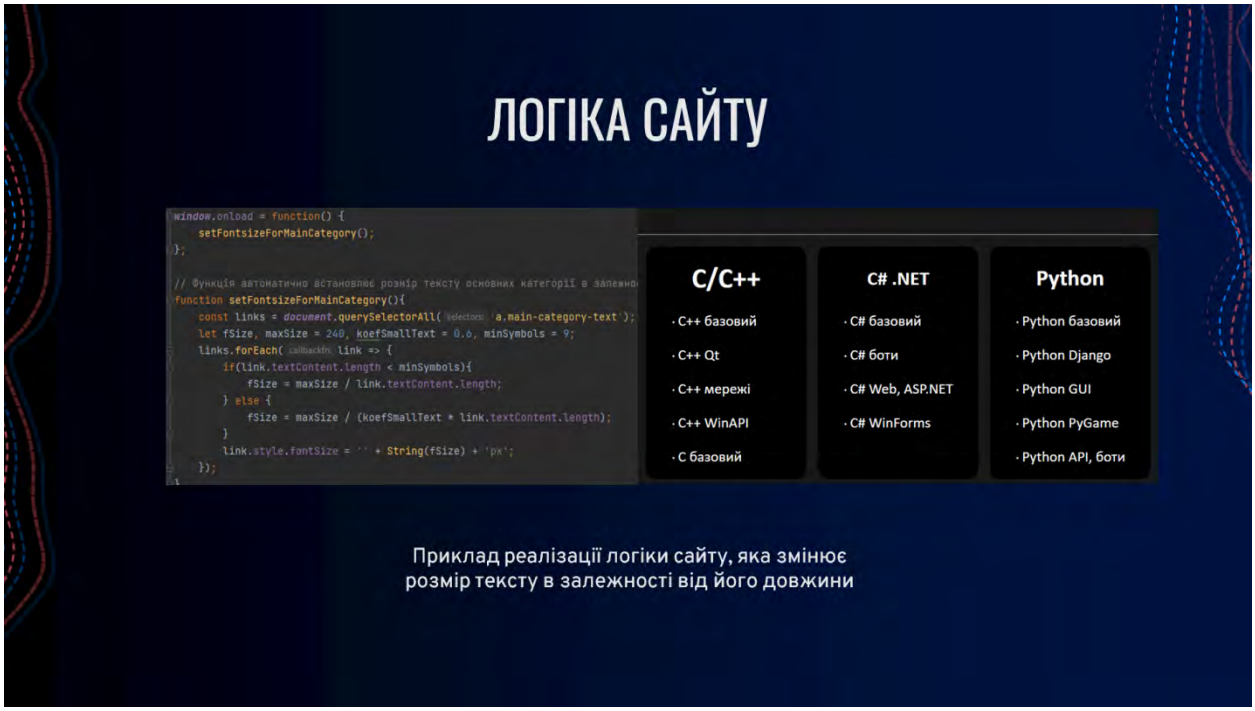

Рисунок В.8 – Слайд прикладу логіки сайту

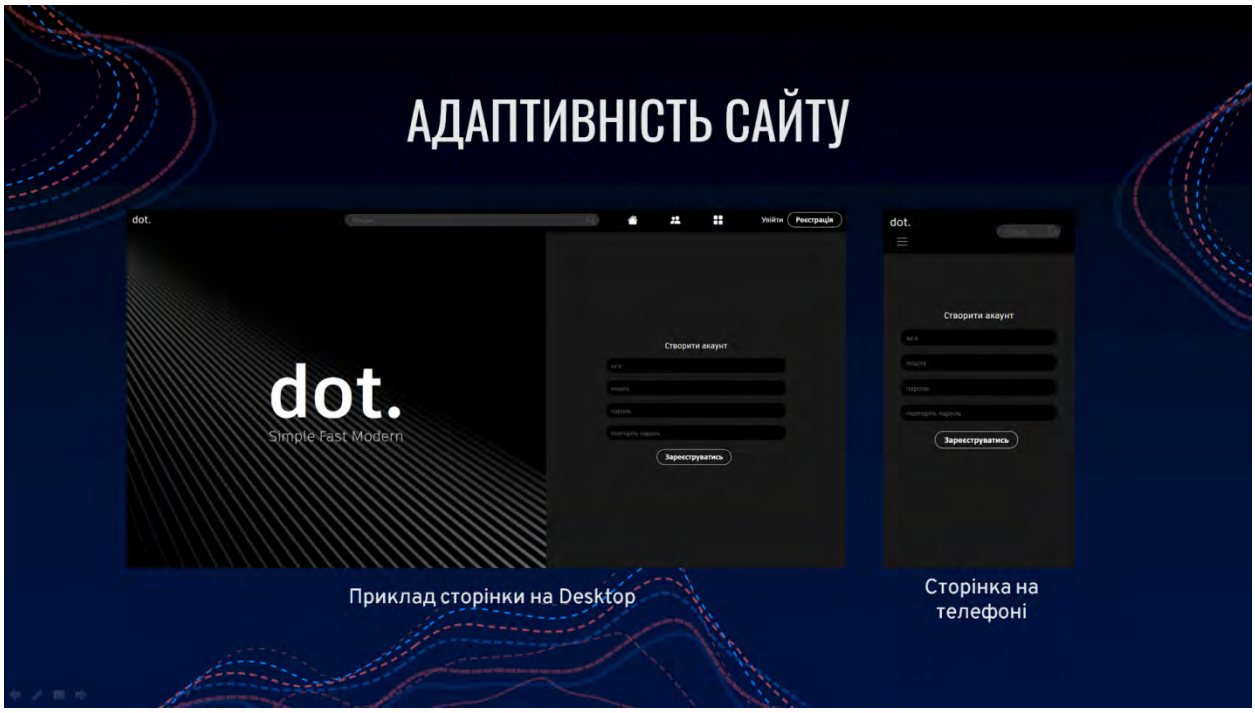

Рисунок В.9 – Слайд демонстрації адаптивності сайту

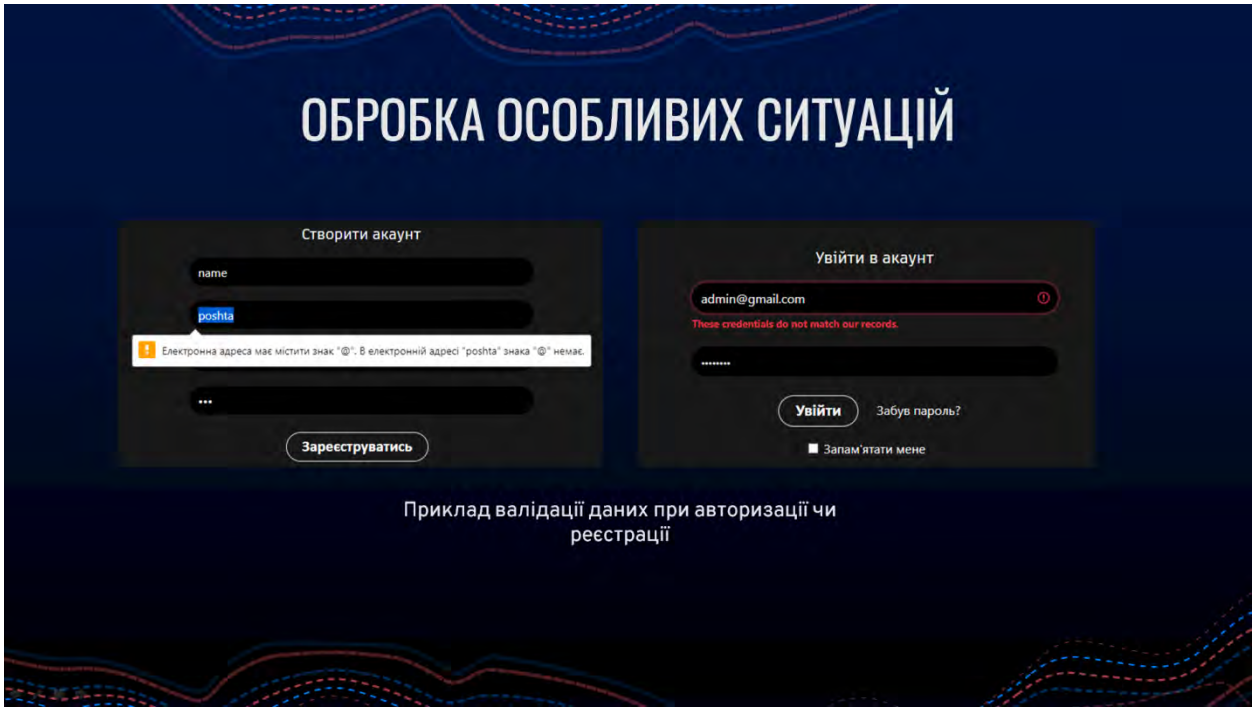

Рисунок В.10 – Слайд першого прикладу обробки особливих ситуацій

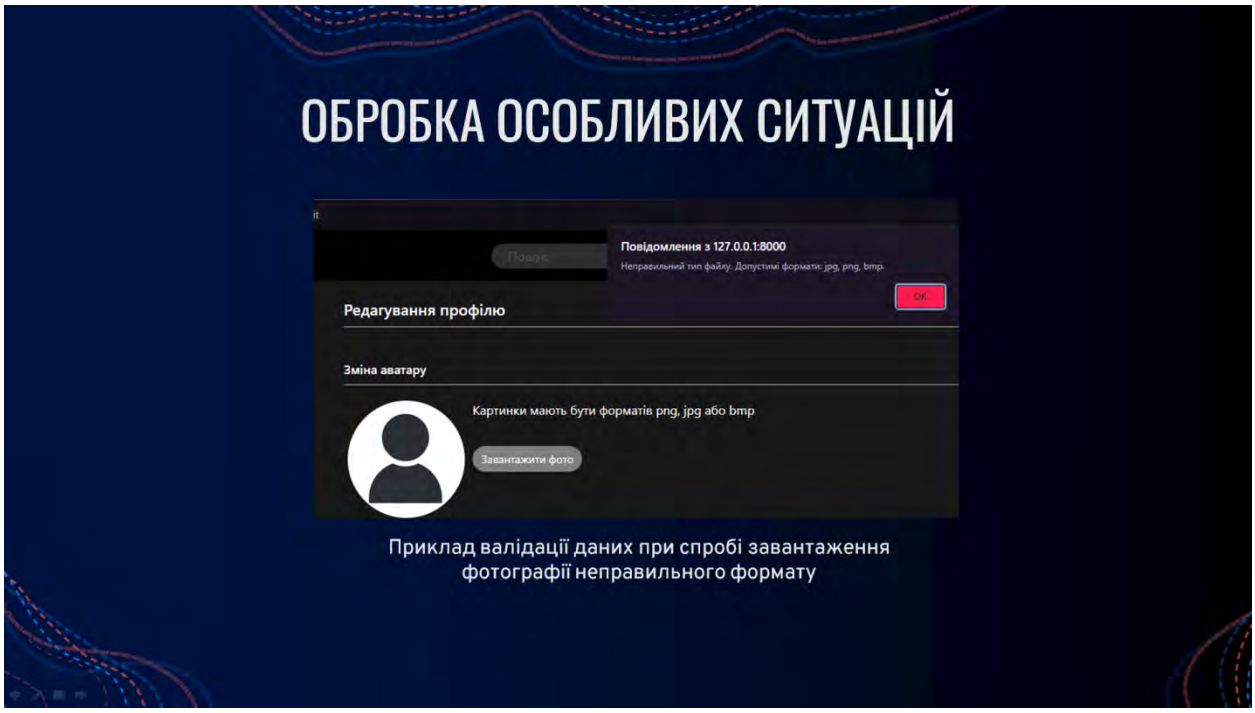

Рисунок В.11 – Слайд другого прикладу обробки особливих ситуацій

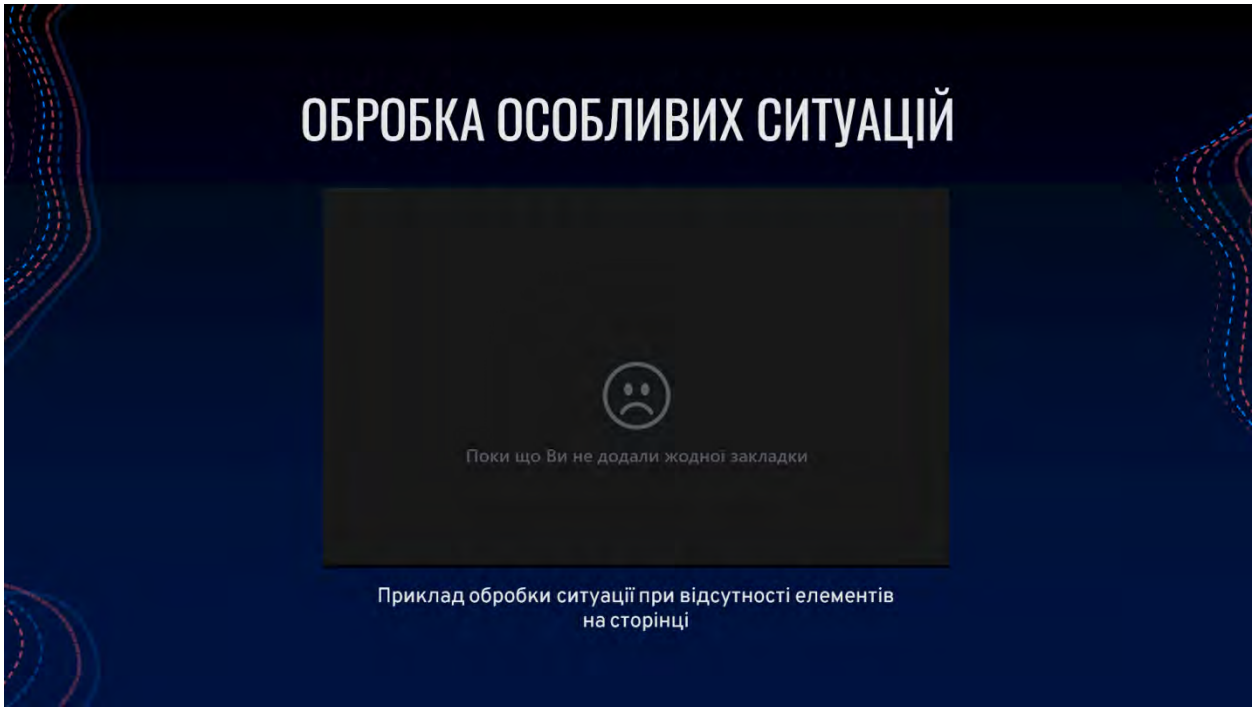

Рисунок В.12 – Слайд третього прикладу обробки особливих ситуацій

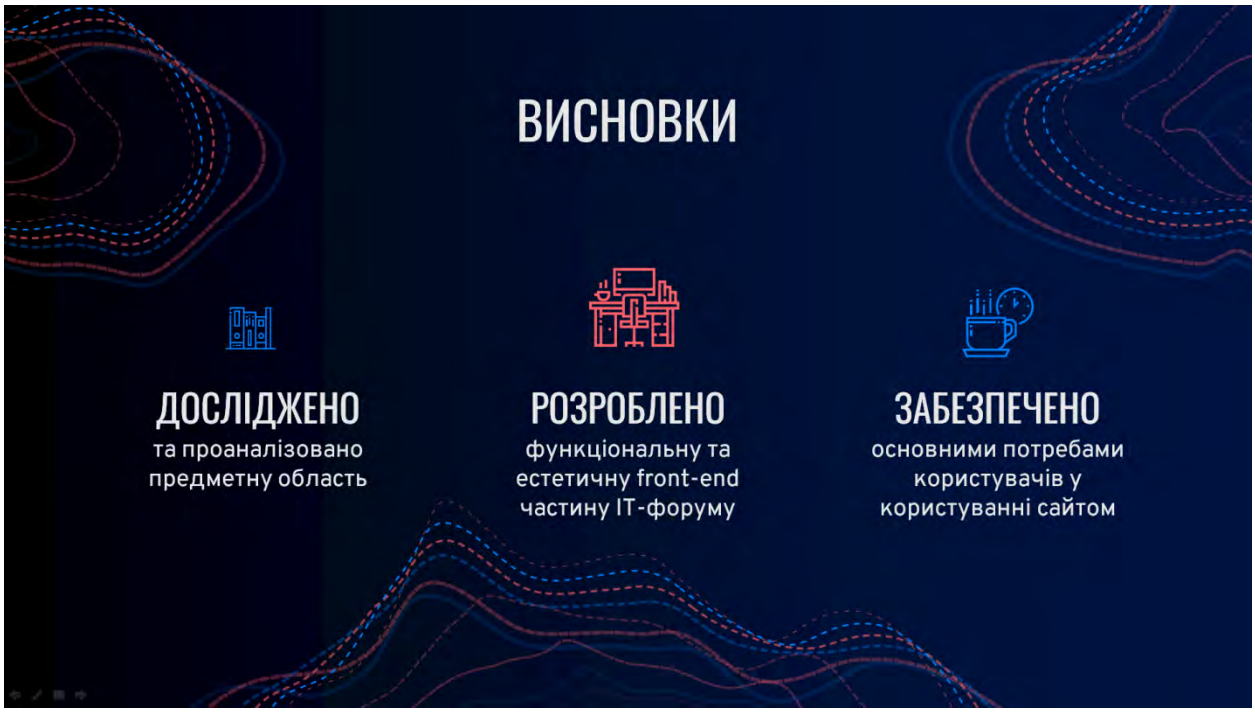

Рисунок В.13 – Слайд висновків за роботою