

Форма № 24

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний університет «Запорізька політехніка»

**Інститут інформатики та радіоелектроніки**  
**Факультет комп'ютерних наук та технологій**  
(повне найменування інституту, факультету)

**Кафедра комп'ютерних систем та мереж**  
(повне найменування кафедри )

## Пояснювальна записка

до дипломного проекту (роботи)

**бакалавра**

(ступінь вищої освіти (освітній ступінь))

на тему **РОЗРОБКА КОМП'ЮТЕРНОЇ СИСТЕМИ ФІТНЕС-КЛУБУ**

---

Виконав: студент 4 курсу, групи КНТ-527  
спеціальності \_\_\_\_\_

123 «Комп'ютерна інженерія»

(код і найменування спеціальності)

Освітня програма (спеціалізація) \_\_\_\_\_

«Комп'ютерна інженерія»

Гурєєв Костянтин Олегович

(прізвище та ініціали)

Керівник Кудерметов Р.К.

(прізвище та ініціали)

Рецензент Куляба-Харитоновна Т.І.

(прізвище та ініціали)

м. Запоріжжя  
2021 рік

Форма № 25

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
**Національний університет «Запорізька політехніка»**  
(повне найменування вищого навчального закладу)

Інститут, факультет інформатики та радіоелектроніки, комп'ютерних наук і технологій  
 Кафедра «Комп'ютерні системи та мережі»  
 Ступінь вищої освіти (освітній ступінь) бакалаврський  
 Спеціальність 123 Комп'ютерна інженерія  
(код і найменування)  
 Освітня програма (спеціалізація) Комп'ютерна інженерія  
(назва освітньої програми (спеціалізації))

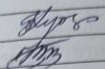

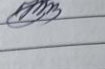

**ЗАТВЕРДЖУЮ**  
 Завідувач кафедри Кудерметов Р.К.  
 «   »     2021 року

**ЗАВДАННЯ**  
 НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТА

Гурєєва Костянтина Олеговича  
(прізвище, ім'я, по батькові)

- Тема проекту (роботи) Розробка комп'ютерної системи фітнес-клубу
- керівник проекту (роботи) Кудерметов Равіль Камілович, к. т. н., доцент  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)  
 затверджені наказом вищого навчального закладу від «17» березня 2021 року № 81
- Строк подання студентом проекту (роботи) 15 травня 2021 року
- Вихідні дані до проекту (роботи) діяльність фітнес-клубу, статистичні відомості щодо продажу абонементів різних видів, попит на пропоновані види діяльності
- Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1) Аналіз технічного завдання;  
2) Проектування системи;  
3) Розробка системи.
- Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)  
Плакат 1 - Схема бази даних;  
Плакат 2 - Макети сторінок системи;  
Креслення 1 – Алгоритм внесення персональних даних клієнта;  
Плакат 3 – Основні сторінки системи.

## 6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконав завдання
1-4	Кудерметов Р.К., к.т.н., доцент		
нормоконтроль	Щербак Н.В., ст. викл.		

7. Дата видачі завдання 01.02.2021 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Прізвище
1	Аналіз технічного завдання	05.02.2021 р.	
2	Опис предметної області	10.02.2021 р.	
3	Аналіз аналогів	20.02.2021 р.	
4	Огляд на вибір програмного забезпечення	01.03.2021 р.	
5	Проектування комп'ютерної системи	15.03.2021 р.	
6	Розробка структури системи	25.03.2021 р.	
7	Розробка алгоритмів роботи модуля з боку користувача та адміністратора	31.03.2021 р.	
8	Розробка системи	20.04.2021 р.	
9	Тестування системи	30.04.2021 р.	
10	Оформлення отриманих результатів у ПЗ	5.05.2021 р.	
11	Оформлення графічного матеріалу	10.05.2021 р.	
12	Оформлення допоміжного матеріалу	15.05.2021 р.	

Студент



К.О. Гурєв

(ініціали та прізвище)

Керівник проекту (роботи)



Р.К. Кудерметов

(ініціали та прізвище)

## РЕФЕРАТ

ПЗ: 62 с., 23 рис., 5 табл., 1 додаток, 13 джерел.

CSS, HTML, JAVASCRIPT, JS, NODE, POSTGRESQL, WWW, ВЕБ-САЙТ, ІНТЕРФЕЙС, МАКЕТ, МОВА ПРОГРАМУВАННЯ, СКБД

Об'єкт розробки – комп'ютерна системи фітнес-клубу.

Метою дипломного проєкту є розробка комп'ютерної системи фітнес-клубу, яка дозволить представити всю необхідну інформацію клієнтам та обробляти персональні дані клієнтів, що підвищить рівень зацікавленості існуючих та потенційних клієнтів, розширить ринок діяльності та покращить імідж фітнес-клубу.

Пояснювальна записка складається з трьох розділів.

У першому розділі проведено опис предметної області функціонування фітнес-клубу, аналіз аналогічних систем, визначені вимоги до системи, проведено вибір програмних засобів для розробки комп'ютерної системи.

У другому розділі розглянуті питання щодо проєктування комп'ютерної системи, подана функціональна схема системи, макет головних сторінок веб-додатку системи, проведено концептуальне та логічне проєктування бази даних, отримано ER-модель бази даних.

Третій розділ присвячений розробці комп'ютерної системи, використовуючи платформу Node.js та систему керування базою даних PostgreSQL, проведено тестування розробленої системи.

В результаті виконання дипломного проєкту розроблено комп'ютерну систему фітнес-клубу, яка задовольняє всім поставленим вимогам.

## ЗМІСТ

Скорочення та умовні позначки .....	6
Вступ.....	7
1 Аналіз технічного завдання.....	8
1.1 Опис предметної області .....	8
1.2 Розгляд аналогів .....	10
1.3 Вимоги до системи.....	11
1.4 Вибір програмних засобів .....	13
2 Проектування комп'ютерної системи .....	21
2.1 Архітектура системи .....	21
2.2 Проектування бази даних .....	26
3 Розробка комп'ютерної системи.....	32
3.1 Алгоритм роботи модуля з боку користувача.....	32
3.2 Алгоритм роботи модуля з боку адміністратора .....	34
3.3 Розробка бази даних.....	36
3.4 Розробка веб-інтерфейсу користувача .....	37
3.5 Розробка веб-інтерфейсу модуля моніторингу інформації клієнтів.....	38
3.6 Тестування роботи системи .....	47
Висновки .....	49
Перелік джерел посилання .....	50
Додаток А.....	52

Перелік графічного матеріалу:

Плакат 1 - Схема бази даних;

Плакат 2 - Макети сторінок системи;

Креслення 1 – Алгоритм внесення персональних даних клієнта;

Плакат 3 – Основні сторінки системи.

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БД	–	база даних
ІК	–	інтерфейс користувача
ЗМІ	–	засоби масової інформації
КС	–	комп'ютерна система
ПЗ	–	програмне забезпечення
СКБД	–	система керування базами даних
CSS	–	cascading style sheets
JS	–	java script

## ВСТУП

У часи економічної нестабільності більшість підприємств почали шукати нові шляхи розвитку. Враховуючи стрімке поширення інформаційних та телекомунікаційних технологій в Україні та зростаючу кількість користувачів мережі Інтернет, одним з найперспективніших шляхів розвитку є автоматизації бізнес-процесів [1].

Керівництво багатьох компаній: від невеликих фірм до великих корпорацій вже давно оцінило всі переваги від впровадження КС та уявити компанію, що працює на сучасному ринку, без КС практично неможливо. Якісна система, вбираючи в себе всю корисну інформацію та працюючи цілодобово, є найкращою візитною карткою будь якої організації, будь то комерційна фірма або державна установа. Саме тому, на сьогоднішній день розробка КС організації є актуальним питанням.

Основною метою створення КС є збільшення обсягів діяльності організації, розширення ринків діяльності, покращення іміджу на ринку. Основними перевагами впровадження КС є:

- охоплення широкого кола споживачів;
- залучення нових клієнтів;
- можливість цілодобового надання інформації;
- підтримка клієнтів в режимі online;
- зручність і простота у наданні інформації клієнтові;
- захищеність інформації від несанкціонованого доступу.

Враховуючи вище сказане, можливо зробити висновок, що тема дипломного проекту «Розробка комп'ютерної системи фітнес-клубу» є актуальною та доцільною.

## 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

### 1.1 Опис предметної області

У даній роботі розглядається організація роботи фітнес-клубу. Фітнес-клуб - підприємство, головним видом діяльності якого є надання клієнтам послуг для покращення фізичних показників людини. Таким чином, збільшення прибутку такої організації, як фітнес-клуб, можливо за допомогою збільшення кількості відвідувачів.

Основну організацію роботи фітнес-клубу ведуть адміністратори. Адміністратори звітують про ситуацію в залі на вимогу генерального директора. Наданням послуг в фітнес-клубі займаються тренери. Клієнт фітнес-клубу це споживач послуг. У таблиці 1.1 наведено основні класи та характеристики користувачів системи фітнес-клубу.

Таблиця 1.1 – Класи та характеристики користувачів

<b>Клас користувачів</b>	<b>Характеристики користувачів</b>
Адміністратор	Це співробітник компанії, що займається наповненням сайту контентом.
Тренер	Це співробітник компанії, який спілкується з клієнтами особисто, працівник, що зв'язує фітнес-клуб з клієнтом, надає кваліфіковані консультації клієнтам.
Клієнт	Користувачі, які зацікавлені в придбанні абонементу.

Тренер – працівник, що зв'язує фітнес-клуб з клієнтом. Незважаючи на надання повної самостійності у виборі фізичних вправ, багато клієнтів потребують кваліфікованої консультації працівників фітнес-клубу.

Обов'язки тренера:

- зустріч клієнта в залі і створення загальної атмосфери гостинності;



- допомога у виборі необхідних вправ, надання вичерпної консультації про кожен тренажер;

- безперервне слідкування за клієнтами;

- виконання доручень керівника;

Адміністратор - співробітник, посадові обов'язки якого передбачають забезпечення штатної роботи комп'ютерної техніки, мережі та програмного забезпечення.

Обов'язком адміністратора:

- додавання потрібної інформації або додаткової конфігурації у КС фітнес-клубу, а також обслуговування сайту;

- обслуговування робочих станцій (встановлення та переустановлення ПЗ, настройка і обслуговування оргтехніки, резервне копіювання), вивантаження / завантаження / оновлення інформації WWW.

Клієнт – це персону, яка бажає придбати абонемент у фітнес-клубі.

Задачі та цілі клієнта:

- вивчення інформації про фітнес-клуб який йому подобається;

- пошук місць, де можливо займатись спортом;

- консультація вибраного фітнес-клубу з професіоналом;

- приймання рішення (вибір абонементу);

- покупка абонементу.

Для поліпшення і збільшення кількості клієнтів було запропоновано створити комп'ютерну систему фітнес-клубу з можливістю в ньому створити систему для моніторингу показників клієнтів. Окрім стандартного шаблону комп'ютерної системи фітнес-клубу, було вирішено додати можливість вести базу даних клієнтів, яка буде містити такі дані як (ПІБ, адресу проживання, мобільний телефон, дату народження та місце роботи), а також персональні дані клієнта, які будуть вимірюватися кожен місяць (вага, зріст, об'єм грудей, об'єм під груддю, об'єм талії, об'єм живота, об'єм стегон, обхват руки, обхват гомілки). Кожен тренер має своїх клієнтів, має доступ тільки до бази своїх клієнтів, має знімати персональні дані клієнтів та заносити їх в базу. Таким

чином, кожен тренер може пред'явити клієнтові інформацію про зовнішні показники тіла за минулий місяць, щоб він на власні очі зміг подивитися на зміни свого тіла в той чи інший бік. Після вибору потрібного абонементу, клієнту потрібно сплатити вартість абонементу, для того щоб тренер почав знімати зовнішні показання клієнта.

Отже, після детального вивчення проблем у сфері розробки інформаційних систем та у сфері фітнес-залів в цілому, були виділені наступні задачі до дипломної роботи:

- розробити комп'ютерну систему для фітнес-клубу, яка буде надавати повну інформацію о послугах;
- розробити окремий модуль для моніторингу показників клієнтів, який повинен надавати можливість тренерам додавати, видаляти та редагувати показники клієнтів.

## **1.2 Розгляд аналогів**

Для того, щоб спроектувати КС фітнес-клубу були розглянуті аналоги таких сайтів фітнес-клубів як «JP-STUDIO» [2] та «Fitness River» [3].

На сайті Fitness River було виявлені такі недоліки:

- занадто велика кількість пустих та пасивних областей;
- головне меню зливається з основним змістом сайту;
- немає списку існуючих абонементів та їх цін;
- відсутня можливість заморозки абонементу;
- не можна здійснити пошук необхідної послуги за словесним описом.

Проте сайт Fitness River привабив такими можливостями:

- кожна послуга дуже вдало та коротко описана на сайті;
- наявність запису на пробне заняття;

- зручна та зрозуміла структура сайту та вдала панель навігації;
- відображення на головній сторінці актуальних акцій.

На сайті «JP-STUDIO» було виявлені такі недоліки:

- дуже не вдала та незрозуміла панель навігації;
- відсутня можливість написати фітнес-клубу онлайн;
- сторінка завантажується повільно.

Сайт «JP-STUDIO» привабив такими функціями:

- наявність прас листу;
- наявність вибору абонементів з коротким описанням про них;
- наявність відео фрагменту на головній сторінці веб-сайту.

Для КС, що проектується були запозичені такі переваги:

- зручна та зрозуміла структура сайту та вдала панель навігації;
- відображення на головній сторінці актуальних акцій;
- наявність прас листу;
- наявність вибору абонементів з коротким описанням про них;
- наявність відео фрагменту на головній сторінці веб-сайту.

### **1.3 Вимоги до системи**

Комп'ютерна система фітнес-клубу повинна дозволяти клієнтам отримувати інформаційну підтримку про існуючі послугу, акції і новини, можливість дізнатись ціни на абонементи, подивитись місце знаходження фітнес-клубу, дізнатись розклад тренерів та інформацію про них.

Вимоги до програмного середовища:

- система повинна працювати з усіма Інтернет-браузерами;
- система встановлена на сервері, що працює під управлінням Windows;

- система повинна допускати доступ користувачів як з внутрішньої мережі компанії, так і через Інтернет-підключення з будинків користувачів.

Обмеження дизайну та реалізації:

- система повинна використовувати БД postgresQL;
- весь код HTML повинен відповідати стандарту 5.0;
- усі сценарії повинні бути написані на Node, JS;
- верстка сайту повинна бути адаптивною та кросбраузерною.

Вимоги до інтерфейсу системи:

- інтернет-сторінки повинні надавати повну можливість навігації по ресурсу;
- екрани виводу користувачів повинні бути стандартних розмірів різноманітних пристроїв (монітори персональних комп'ютерів, смартфони, планшети).

Web-сайт повинен спілкуватися з адміністратором сайту через програмний інтерфейс, виконуючи наступні операції:

- надавати можливість адміністратору авторизуватися в системі для доступу в адміністративної панель;
- надавати можливість адміністратору додавати, видаляти, редагувати інформацію на сайті через програмні інтерфейси адміністративної панелі.

Вимоги до продуктивності:

- система повинна підтримувати тисячу підключень одночасно;
- всі інтернет-сторінки повинні завантажуватися не більше 10 секунд по повільному модемному з'єднанню 500кб/сек;
- загрузка відповідей на запити на екрані повинна займати не більше 5 секунд після того, як користувач виконав запит.

## 1.4 Вибір програмних засобів

На ранніх етапах розвитку Інтернету, розробка комп'ютерної системи зводилася до створення файлової структури з HTML-сторінок та ручного розміщення в них тексту, зображень, елементів навігації та посилань. Тоді, це не було настільки трудомістким заняттям, щоб дбати про його автоматизацію, оскільки сайти здебільшого представляли невеликі проєкти, які робилися для автора чи його колег.

З часом обсяги інформації почали експоненціальне зростати, збільшилося число відвідувачів сайтів, збільшилися трудовитрати на підтримку інформаційної системи в актуальному стані. Власник системі був змушений більшу частину часу витратити не на безпосереднє розміщення статті або публікації, а на внесення супутньої інформації, на зразок посилань на цю статтю, зміни меню навігації або виправлення відомостей.

Виходом з даної ситуації стало створення спеціального класу програм – систем керування контентом, які спроможні виконувати рутинні операції, не пов'язані з безпосереднім створенням статей.

Для створення власної інформаційної системи необхідно вирішити такі завдання:

- ознайомитися з сучасними Інтернет технологіями і, по можливості, використовувати їх у своїй розробці;
- вивчити програмний інструментарій, застосовуваний для розробки і створення системі;
- виявити і врахувати методи і способи подання на web-сторінках різних видів інформації, що не перешкоджають їх доступності;
- ознайомитися з основними правилами і рекомендаціями по розробці і створенню web-сайтів і неухильно дотримуватися їх у своїй практиці;
- визначитися зі структурою web-сторінок;
- вибрати стратегію розробки та створення комп'ютерної системи.

В середовищі програмістів давно стало традицією те, що для кожної нової області, в якій впроваджуються програми, створюється нова мова програмування, яка буде найбільш зручною для цієї області і врахує всі її особливості. Для Інтернету на даний момент такими мовами, є HTML, CSS, Ruby, PHP, Python, Java, JavaScript, Node. Ці мови підтримують практично всі сучасні хостингові компанії, вони інтегрується з багатьма базами даних, є безкоштовними, має відкриті початкові коди і велику бібліотеку готових програм (скриптів).

Слід зазначити, що HTML не є мовою програмування [4, 5]. І справа тут навіть не в тому, що у HTML немає компілятора, а тільки вбудований в браузер інтерпретатор: Basic і JavaScript теж обходяться одними інтерпретаторами, але це не заважає їм називатися мовами програмування. Справа в тому, що в HTML відсутній головний атрибут, властивий будь-якій мові програмування, - команди. На HTML не можна задати послідовність дій, а можна тільки описати, як браузер повинен вводити на екран той чи інший документ.

HTML - це типовий мову розмітки, тобто з його допомогою можна оформляти документи, створювати посилання, але ніяк не писати програми. Навіть спеціальні ефекти, які можна побачити на web-сторінках, створюються не за допомогою HTML, а з використанням додаткових засобів - наприклад, вбудованих в документ програм на мові JavaScript або PHP. Будь-браузер дозволяє переглянути вихідний текст HTML-документа

Стилем або CSS (Cascading Style Sheets, каскадні таблиці стилів) називається набір параметрів форматування, який застосовується до елементів документа, щоб змінити їх зовнішній вигляд [5]. Можливість роботи зі стилями здавна включають в розвинені видавничі системи і текстові редактори, тим самим дозволяючи одним натисканням кнопки надати тексту заданий, заздалегідь встановлений вид. Тепер це доступно і творцям сайту, коли колір, розміри тексту і інші параметри зберігаються в певному місці і легко «прикручуються» до будь-якого тегу. Ще однією перевагою стилів є те, що вони

пропонують набагато більше можливостей для форматування, ніж звичайний HTML.

CSS являє собою потужну систему, яка розширює можливості дизайну і верстки веб-сторінок.

PHP - найпоширеніша мова веб-програмування з легким синтаксисом, яку можна швидко освоїти [5, 6]. Є досить універсальна і застосовувати її можна в створенні абсолютно різних інтернет-ресурсів.

Існує безліч довідників і посібників по даному мови, а також готових рішень. Більш того, PHP вміє ефективно працювати з двадцятьма різними базами даних, з яких найбільшою популярністю користується MySQL.

Незважаючи на те, що дана мова деякі програмісти списують з рахунків і вважають застарілим, переважна більшість інтернет-сайтів розробляються саме на ньому і він не збирається здавати позиції. Більшість вакансій і проектів на фрілансі спрямовані саме на тих розробників, які здатні працювати з PHP. Реалізувати на ньому можна будь бекенд, в тому числі інтернет-магазини і соціальні мережі.

Python - потужний інструмент, здатний справлятися з найскладнішими обчисленнями. Взято на озброєння світовими гігантами на кшталт Google і Dropbox.

На відміну від PHP, освоїти Python новачкові буде складно. Синтаксис на перший погляд набагато складніше і вельми специфічний, тому що тут реалізується зовсім інший підхід до оголошення змінних. Але потім приходить усвідомлення того, що писати цією мовою дуже легко і приємно. Незрівняним перевагою є більш короткий вихідний код.

Ентузіасти не раз проводили порівняння. Писати сайт на Python набагато більш ергономічно, так як код краще структурований і займає менше місця. В такому кодї легко орієнтуватися і вносити правки.

Присутня велика кількість навчальних матеріалів по даному мови, включаючи друковану продукцію та онлайн-курси.

Також фахівець, який володіє мовою Python, зіткнеться з меншою конкуренцією на ринку праці. Правда і вакансій теж менше.

JavaScript - мова інтерфейсу, яка використовується для створення та розробки веб-сайтів, настільних додатків та ігор. JavaScript працює у всіх браузерах і може працювати з програмами, які не розміщені в Інтернеті. Він підтримує як функціональні, так і об'єктно-орієнтовані стилі програмування, і в основному, це ваш підхід до створення приголомшливих призначених для користувача інтерфейсів і веб-сайтів / додатків / ігор, які виглядають супер круто. Розуміння JavaScript важливо, навіть якщо ваше серце налаштоване на розвиток серверної частини. Компоненти, структури даних і алгоритми JavaScript застосовуються практично до будь-якого іншого мови.

Node - відносно молода мова, створена в 2009 році. Для його вивчення досить знати JavaScript. Мова дозволяє створювати спритні веб-додатки, виконувані на стороні сервера. На сьогоднішній день існує вже 200 000 різних модулів для Node.js і їх кількість постійно зростає [7, 8].

Технології Node.js вибирають для використання настільки великі компанії, як Microsoft, Yahoo і інші.

Найкраще застосовувати цю мову в тому випадку, якщо ви маєте справу з веб-інфраструктурою, яка тісно пов'язана з мобільними додатками. Node.js дозволяє вносити зміни в режимі реального часу. До того ж технології Node.js дозволяють ефективно використовувати оперативну пам'ять і потужність процесора, а також швидше розробляти додатки.

Отже, після порівняння наведених вище мов програмування для створення ІС, було вирішено обрати такі мови: для фронт-енду HTML, CSS та JavaScript, для бек-енду Node.js, тому що ці системи цілком відповідають вимогам створюваного мною ІС. Ці мови мають достатній функціонал та достатню кількість модулів та бібліотек, котрі я можу використовувати або за відсутністю потрібного мені, створити його.

MySQL підтримує SQL (структурована мова запитів) і може застосовуватися в якості SQL-сервера [9]. Це означає, що спілкуватися з



сервером можна на мові SQL: клієнт посилає серверу запит, той його обробляє і віддає клієнту тільки ті дані, які були отримані в результаті цього запиту. Тим самим клієнтові не потрібно викачувати дані і робити обчислення, як, наприклад, в Microsoft Access.

Крім того, MySQL - це ПЗ з відкритим кодом, тобто його можна вільно вивчати та змінювати. Пакет розповсюджується на умовах GPL (General Public License), його можна безкоштовно завантажити з Інтернету для некомерційного використання.

З появою Інтернет-технологій, що дозволяють створювати динамічні Web-сторінки, надзвичайно зріс попит і на СКБД, які найбільш повно підходили б для цього по швидкодії, надійності і стабільності. І тут добре проявив себе пакет MySQL, який вийшов швидким, простим і надійним, але, правда, за рахунок погіршення функціональності (відразу обмовимося, що розробники MySQL обіцяють додати відсутні функції вже в найближчих версіях програми).

За великим рахунком, відсутність деяких функцій, які були принесені в жертву швидкодії і надійності, не створює великих клопотів користувачам (хоча іноді якийсь дискомфорт і має місце). Для роботи з повноцінною корпоративною базою даних MySQL недотягує, але з повсякденними завданнями MySQL справляється досить добре.

PostgreSQL — об'єктно-реляційна система керування базами даних. Є альтернативою як комерційним СКБД (Oracle Database, Microsoft SQL Server, IBM DB2 та інші), так і СКБД з відкритим кодом (MySQL, Firebird, SQLite).

Порівняно з іншими проектами з відкритим кодом, такими як Apache, FreeBSD або MySQL, PostgreSQL не контролюється якоюсь однією компанією, її розробка можлива завдяки співпраці багатьох людей та компаній, які хочуть використовувати цю СУБД та впроваджувати у неї найновіші досягнення.

Сервер PostgreSQL написаний на мові C. Зазвичай розповсюджується у вигляді набору текстових файлів із сирцевим кодом. Для інсталяції необхідно відкомпілювати файли на своєму комп'ютері і скопіювати в деякий каталог.

Вона підтримує більшу частину стандарту SQL і пропонує безліч сучасних функцій:

- складні запити;
- зовнішні ключі;
- тригери;
- змінювані уявлення;
- транзакційна цілісність;
- багатOVERсійність.

Крім того, користувачі можуть всіляко розширювати можливості PostgreSQL, наприклад створюючи свої:

- типи даних;
- функції;
- оператори;
- агрегатні функції;
- методи індексування;
- процедурні мови.

А завдяки вільній ліцензії, PostgreSQL дозволяється безкоштовно використовувати, змінювати і поширювати всім і для будь-яких цілей - особистих, комерційних чи навчальних. Також існує графічна оболонка для цієї СКБД.

PgAdmin - це вільне кроссплатформне (працює і в \* nix, і в Windows) програмне забезпечення, яке надає графічний інтерфейс для роботи з базою даних.

Отже, після порівняння наведених вище СКБД для створення КС, було вирішено обрати СКБД PostgreSQL, тому що ця система цілком відповідає вимогам створюваного мною ІС. Ця СКБД має достатній функціонал, котрий я можу використовувати.

Зараз існує дуже багато різноманітних редакторів та інструментів для створення дизайну. Але я обирала між двома всім відомим програмами для створення дизайну – це Adobe Photoshop та Adobe Illustrator.

Adobe Illustrator являє собою програмний продукт для створення векторної графіки, у якої, як відомо, є деякі важливі особливості [10]. Наприклад, векторне зображення повинно мати можливість нескінченного варіювання масштабом, адже саме ця властивість є першою вимогою для друку. Adobe Illustrator розпорядженні дивовижними програмними налаштуваннями, що робить його унікальним графічним «інструментом», хоча і спостерігається велика схожість з Corel Draw.

Adobe Illustrator, як правило, використовується для:

- розробки логотипів, фірмових знаків та інших символів компанії;
- підготовки до друку;
- веб-графіки;
- створення інших векторних зображень.

Незважаючи на те, що Adobe Illustrator може бути повною мірою використаний для створення сайтів, Photoshop і раніше залишається кращою програмою у формуванні конструкцій користувача інтерфейсу і інших графічних елементів для вашої веб-сторінки.

Illustrator не може реалізовувати анімацію. Якщо ви шукаєте програмний засіб для розробки анімації, то Illustrator точно не для вас. Кращим рішенням буде створити зображення в Illustrator і перевести їх у будь-яку відповідну програму для анімації, наприклад, Adobe Flash.

В Adobe Illustrator фільтри не «функціонують» точно так само, як в Photoshop. Якщо Ви вважали, що фільтри цих двох програм створюють аналогічні ефекти, то Ви помилялися

Як вже говорилося вище, Illustrator відмінно підходить для дизайну логотипу, товарних знаків, розробки різного брендингу, а також створення вітальних листівок, запрошень на весілля та іншого.

Після виходу Illustrator CS4 стало можливим створювати складні брошури, річні звіти і навіть книги. Єдине, чим Illustrator поступається Indesign, так це відсутністю деяких утиліт, таких як шаблони книги або нумерація сторінок. Незважаючи на це, Illustrator версії CS4 і вище має підтримку декількох

монтажних областей для створення «книг», але лише для того, щоб створювати невеликі брошури або переводити кілька сторінок дизайну в PDF формат для демонстрації результатів роботи вашим клієнтам.

Adobe Photoshop виявляється найбільш популярним з програм Adobe [11]. Найголовнішим, що обов'язково потрібно знати про Photoshop, це його приналежність до програм для роботи з пікселями, тобто з растровими зображеннями, які залишатимуться у фіксованому розмірі.

На початку свого розвитку Photoshop був створений для мінімальної обробки фотографій, а не для якихось масштабних цілей. Надалі Adobe помітили, що багато користувачів пристосувалися використовувати Photoshop для створення складних конструкцій для користувача інтерфейсу, графіки на веб-сторінках, банерів, текстових ефектів і багато чого іншого, і почали розробляти функціонал, який став незамінним помічником будь-якого дизайнера і художника.

Adobe Photoshop, як правило, використовується для:

- редагування фото і кольорокорекції;
- розробки дизайну користувацького інтерфейсу мобільних додатків;
- веб-графіки;
- реалізації фотоефектів і ефектів з текстом.

Photoshop є потужним і практично незамінним інструментом для всіляких експериментів з фотографіями, кольорокорекцією, а також реалізації дивовижних ефектів, які в сучасному світі прирівнюються до творів мистецтва. Крім цього, Photoshop вельми ефективний для розробки дизайну веб-сторінок, деяких веб-елементів, банерів, а так само «прикрас» для мобільних і комп'ютерних додатків.

Інструменти Photoshop, прямо таки, «заточені» для роботи з веб-графікою: перетворення графіки у веб-формати GIF, JPEG, PNG. Безпечна палітра кольорів для однакового відображення в будь-яких браузерях. Створення анімації -

анімованих банерів, ролловеров - кнопок, що змінюються при наведенні курсору та інші події.

Спеціальний модуль програми - оптимізація графіки для розміщення в Мережі (Файл - Зберегти для Web та пристроїв), що володіє значною кількістю інструментів. У вкладках модуля видно результати декількох варіантів оптимізації одночасно, що дозволяє їх порівняти за співвідношенням якості-розмір і вибрати потрібний.

Розподіл зображення на фрагменти інструментом Раскройка або з використанням панелі Шари. Фрагменти збираються на веб-сторінці за допомогою HTML-коду або шарів CSS, причому код генерується програмою Photoshop. Цей механізм зручний для створення макетів веб-сторінок або, наприклад, меню.

Тоді кожному фрагменту необхідно призначити гіперпосилання, виділивши його і викликавши діалогове вікно «Параметри фрагмента». У цьому вікні можна задати атрибути Alt, Title і інші параметри гіперпосилання.

Отже, після порівняння, було прийняте рішення використовувати у якості інструмента для створення дизайну саме Adobe Photoshop.

## **2 ПРОЄКТУВАННЯ КОМП'ЮТЕРНОЇ СИСТЕМИ**

### **2.1 Архітектура системи**

Структура КС - деревоподібне ієрархічне представлення всіх елементів, включаючи інформаційні розділи та їх підрозділи. Вона повинна бути на вищому рівні. Кожен відвідувач повинен відразу розуміти структуру сайту, види послуг, як знайти необхідну інформацію і при цьому знаходити місця, які вже переглянув з будь-якого місця сайту без напружень [12].

Перед тим як створити свою систему потрібно скласти чітку структуру, представивши його у вигляді книги у якої є зміст, який і є навігацією по КС.

Назви основних розділів, розташовані на головній сторінці повинні нести загальну інформацію про вміст, а в підрозділі - деталізувати її. Таким чином, користувач, рухаючись по сайту, буде цілеспрямовано переміщатися від загального до конкретного.

Хороша і прозора структура системи, легка навігація - є ознакою хорошого тону. Потрібно придумати назви розділах і підрозділах, коротко охарактеризувати їх (вказати яка інформація буде в них розміщена).

Ідеальна структура КС має один вхід і один вихід. Відвідувач, заходячи на головну сторінку сайту, знаходить потрібну йому інформацію, під якою знаходиться посилання на сторінку, де її можна знайти. Таким чином, відвідувач просувається від головної сторінки до внутрішніх, що дозволяє йому не пропустити ту інформацію, яка буде йому цікава і корисна. Єдиний вихід з сайту у відвідувача лежить через відвідування як мінімум двох сторінок, що збільшує значення лічильника відвідувань. Дана структура сайту будується за принципом «воронки». Особливо ефективна така структура при створенні інтернет - магазину, корпоративного і бізнес - сайту, де виходом є сторінка замовлення товару.

Зовнішня структура - це те, яким чином буде розташовуватися інформація на сторінці при виборі кожного пункту меню [13]. Зовнішня структура КС зображена на рисунку 2.1.

У кожній ІС є шапка, де зазвичай пишеться заголовок (підзаголовок) сайту, навігаційне меню, де пишуться назви рубрик, міток та інша корисна інформація, основна частина ІС (на рисунку - область контенту), де публікується основна інформація (просто кажучи - там, де сама стаття), підвал, де показана деяка службова інформація.

Звичайно, така структура - не догма, існують сайти без шапки, без футера (підвалу), з однією або декількома бічними колонками (вона називаються сайдбар), сайдбари можуть розташовуватися праворуч, ліворуч або з обох сторін, але в цілому така структура оптимальна, дозволяє подати всю потрібну інформацію.

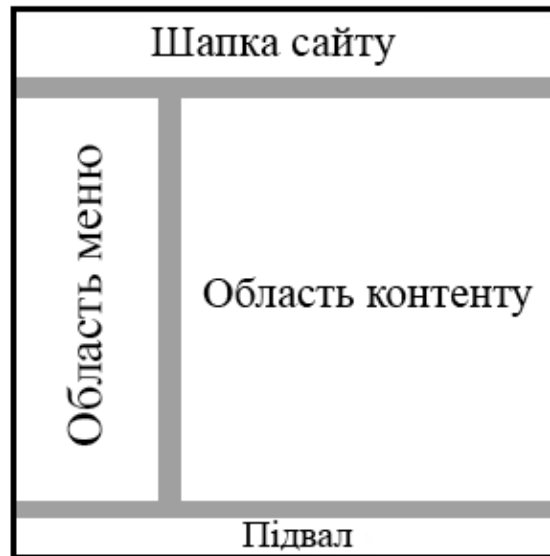


Рисунок 2.1 – Зовнішня структура системи

Коли відвідувач потрапляє до КС, то він або шукає потрібну інформацію, або реагує на подачу системи. І в тому і в іншому випадку потрібно його зацікавити. При цьому обличчям КС є головна сторінка, а значить, саме вона повинна бути самою інформативною частиною. При цьому важливе значення для відвідувача має простота структури.

Внутрішня структура моєї системи складається з інформаційної системи та бази даних. З внутрішньою структурою інформаційної системи визначитися набагато складніше. В першу чергу, внутрішня структура залежить від напрямку інформаційної системи, та від того, якою тематики буде інформаційна система. Функціональну схему зображено на рисунку 2.2.

Від структури КС безпосередньо залежить, чи зможе відвідувач знайти цікавий для нього матеріал, наскільки йому буде комфортно на сайті. І тут може виникнути питання: яка з двох структур важливіше? Обидві важливі і жодної з них нехтувати не можна.

Із зовнішньої структурою КС визначитися легше. Звичайно кожен з структур можна вдосконалити і досить серйозно, але спочатку можна дотримуватися її. І змінювати її в майбутньому легше, ніж внутрішню структуру.



Рисунок 2.2 – Функціональна схема комп'ютерної системи

В своєму дипломному проекті я розробляю КС для фітнес-клубу. Отже, було обрано лінійну внутрішню структуру сайту, бо вона найкраще відповідає потребам користувачів та потенційних клієнтів КС. Внутрішню структуру моєї системи можна побачити на рисунку 2.3. На рисунку 2.4 зображено макет головної сторінки КС.

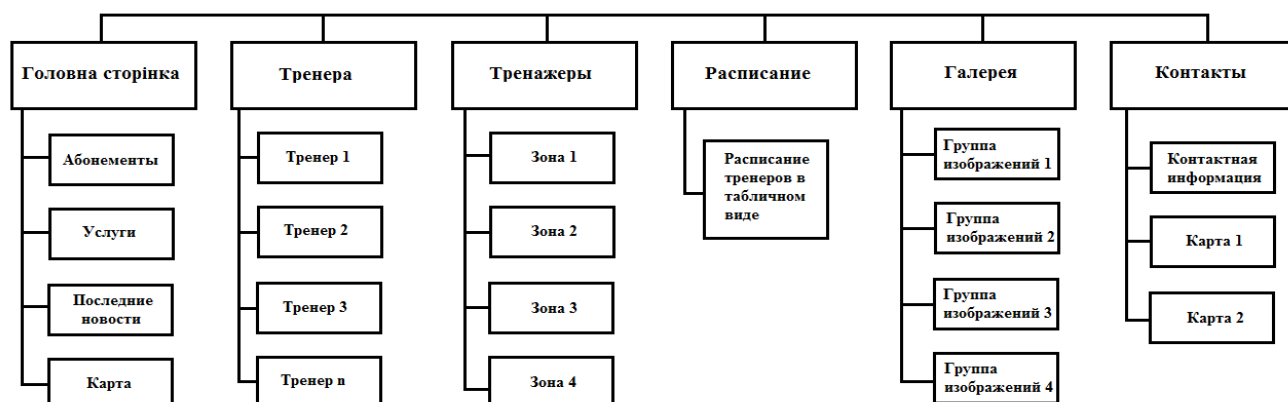


Рисунок 2.3 – Структурна схема інформаційної системи





Рисунок 2.4 – Макет головної сторінки

В наші дні фітнес клуб - популярна модель бізнесу, яка може приносити непоганий дохід. Але конкуренція в цій ніші дуже велика, тому власникам таких фірм доводиться думати про те, як зробити свій бізнес краще. Хороше рішення - дати можливість клієнтам слідкувати за змінами показників їх зовнішнього вигляду. Оскільки готових модулів та скриптів я не знайшов, було вирішено створити власний. Така функціональність реалізується за допомогою серверної мови Node.js та реляційної база даних PostgreSQL, котрий я розробив у текстовому редакторі Atom.

Фітнес-клуб з можливістю моніторингу змін показників клієнтів значно виграє в порівнянні зі звичайними фітнес-клубами, тому що:

– бізнес стає унікальним. Дуже багато залів пропонують низькі ціни або найновіше обладнання для тренування і далеко не кожен з них має систему для моніторингу показників клієнтів, зручний для використання, де кожен тренер може пред'явити клієнтові інформацію про зовнішні показники тіла за минулий місяць, щоб на власні очі зміг подивитися на зміни свого тіла в той чи інший бік. Таких сайтів вкрай мало, а значить ця функція дає шанс вийти на лідируючі позиції у своїй ніші і зробити ім'я свого фітнес-клубу популярним;

– це гарний спосіб задовольнити навіть самих вибагливих клієнтів. Кожна людина мріє виглядати ідеально, знаючи показники свого тіла, клієнт буде прагнути покращити ці показники до ідеалу, тому, маючи цей додаток клієнтові буде набагато легше слідкувати за своїм тілом. Отже фітнес-клуб з функцією моніторингу зовнішніх показників тіла клієнта - реальний шанс догодити навіть самим вибагливим відвідувачам;

– фітнес-клуб с функцією моніторингу може стати відмінним джерелом для креативних подарунків. Тому що вибір подарунка завдає чимало клопоту, а можливість подарувати абонемент у фітнес клуб в якому клієнт зможе слідкувати за своїм тілом, завжди може стати у нагоді. Такі подарунки завжди піднімають настрій і ніколи не лежать без діла.

Отже, переваги КС фітнес-клубу з можливістю слідкування зовнішніх показників тіла клієнта на обличчя, тому було прийняте остаточне рішення створити цей ІС с допоміжним модулем для моніторингу.

## **2.2 Проєктування бази даних**

Процес, в ході якого вирішується, який вигляд буде у знову створюваної бази даних, називається проєктуванням бази даних. На етапі проєктування необхідно передбачити всі можливі дії, які можуть виникнути на різних етапах життєвого циклу бази даних. Необхідно проаналізувати запити користувачів,

вибрати інформаційні об'єкти та їх характеристики і на основі аналізу структурувати предметну область.

Аналіз предметної галузі доцільно розбити на три фази:

- аналіз концептуальних вимог та інформаційних потреб;
- виявлення інформаційних об'єктів і зв'язків між ними;
- побудова концептуальної моделі предметної області та проектування

концептуальної схеми бази даних.

Основна мета проектування бази даних - це скорочення надмірності збережених даних, а отже, економія об'єму використовуваної пам'яті, зменшення витрат на багаторазові операції відновлення надлишкових копій і усунення можливості виникнення протиріч через збереження в різних місцях відомостей про один й той самий об'єкт. Результатом проектування бази даних є перетворення опису предметної галузі у внутрішню схему бази даних.

При проектуванні бази даних вирішуються три основних проблеми:

- проблема адекватного відображення предметної галузі та інформаційних потреб користувачів у концептуальній моделі. Цю проблему називають проблемою інфологічного проектування баз даних, на якому створюють змістовні концептуальні моделі;

- проблема знаходження способу адекватного відображення об'єктів предметної галузі в абстрактні об'єкти моделі даних так, щоб це відображення не суперечило семантиці предметної галузі, і було, по можливості, найкращим. Цю проблему називають проблемою логічного проектування баз даних. Ціль логічного етапу проектування - організація даних, виділених на попередньому етапі, у форму, прийняту в обраній СКБД чи СКБД обраного класу.

- проблема забезпечення ефективності виконання запитів до бази даних, тобто як, з урахуванням особливостей конкретної СКБД, розташувати дані у зовнішній пам'яті. Це проблема фізичного проектування баз даних. Мета цього етапу - вибір раціональної структури збереження даних і методів доступу до них.

У ході проектування БД доводиться вирішувати наступні задачі:

- виявлення і представлення даних і зв'язків між ними, необхідних для всіх основних областей застосування даного додатка і будь-яких існуючих груп його користувачі;
- створення моделі даних, здатних підтримувати виконання будь-яких необхідних трансакцій операцій обробки даних;
- розробка попереднього варіанта проекту, структура якого дозволяє задовольнити всі основні вимоги з боку продуктивності системи.

При розробці концептуальної моделі необхідно, в першу чергу, визначити сутність. Виходячи з аналізу предметної області, можна виділити наступні сутності:

- «Клієнти» (Код\_клієнта, ПІБ, Адреса\_проживання, Email, Мобільний\_телефон, Дата\_народження, Місце\_роботи, Коментарі);
- «Заміри» (Код\_клієнта, Дата\_замірів, Вага, Зріст, Об'єм\_грудей, Об'єм\_під\_груддю, Об'єм\_талії, Об'єм\_живота, Об'єм\_стегон, Обхват\_руки, Обхват\_гомілки);
- «Відвідування» (Код\_клієнта, Дата\_відвідування);
- «Абонементи» (Код\_клієнта, Вид\_абонементу, Початок\_абонементу, Кінець\_абонементу).

Важливим етапом складання концептуальної моделі бази даних є визначення зв'язків між сутностями. Виходячи з вимог технічного завдання, можна визначити наступні взаємозв'язки між сутностями:

- «Заміри» → «Клієнти» (М:1) – кожен клієнт може мати багато замірів на кожен місяць, але всі заміри відносяться тільки до одного клієнта (рис. 2.5);

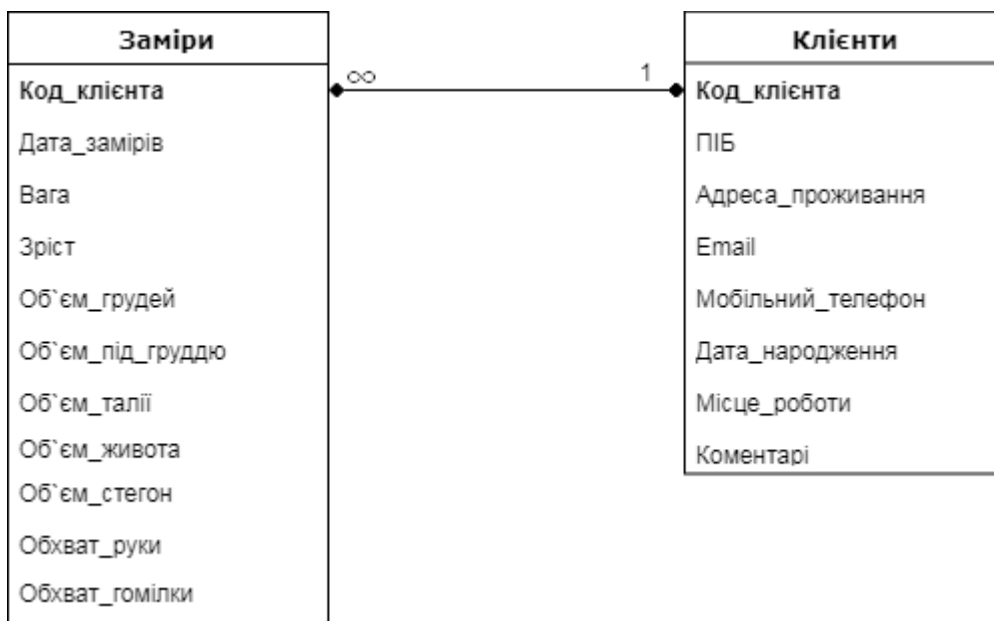


Рисунок 2.5 – Відношення «Заміри» → «Клієнти» (M:1)

– «Клієнти» → «Абонементи» (1:M) – кожен клієнт після закінчення дійсності абонементу, купляє новий, тобто може мати багато абонементів, але всі ці абонементи відносяться тільки до одного клієнта (рис. 2.6);

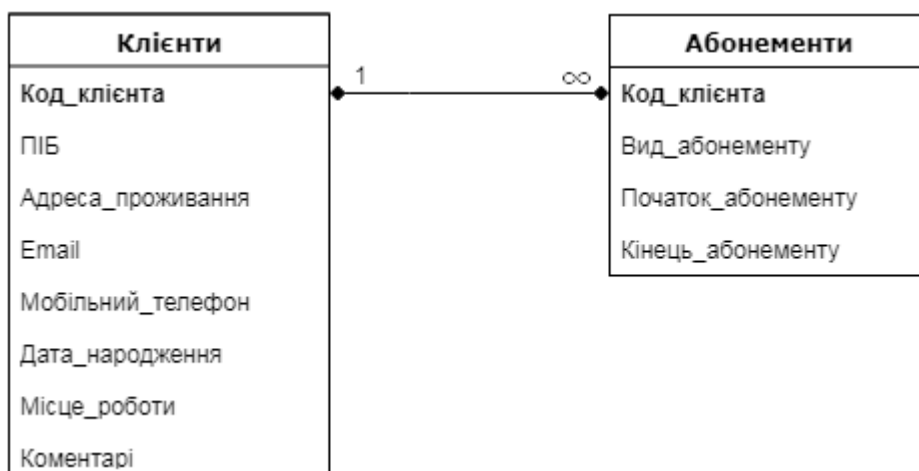


Рисунок 2.6 – Відношення «Клієнти» → «Абонементи» (1:M)

– «Клієнти» → «Відвідування» (1:M) – кожен клієнт відвідує фітнес-зал n-ну кількість разів, в залежності від типу абонементу, тобто один клієнт може мати багато відвідувань, але всі ці відвідування відносяться тільки до одного клієнта (рис. 2.7).

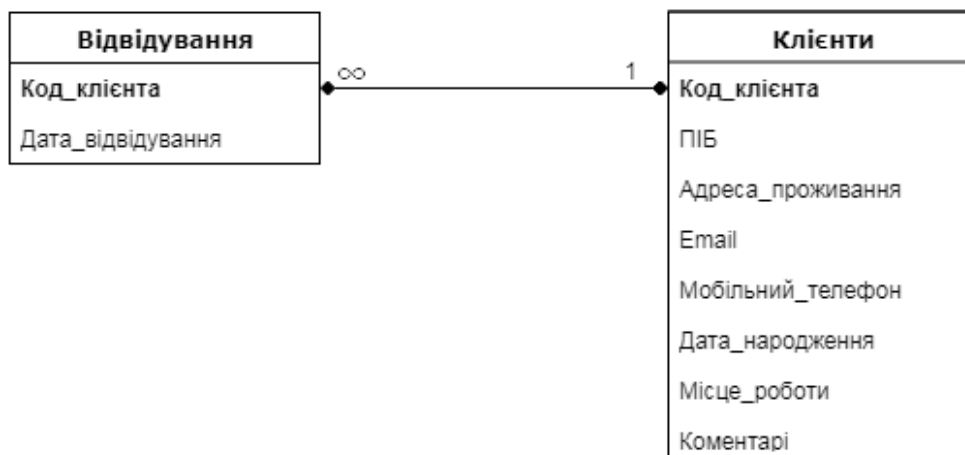


Рисунок 2.7 – Відношення «Клієнти» → «Відвідування» (1:М)

Метою етапу логічного проектування бази даних є перетворення концептуальної моделі в логічну модель.

В процесі проектування логічної моделі бази даних визначається кількість і склад таблиць для зберігання вихідної інформації. Кожна таблиця представляється назвою, переліком атрибутів і первинним ключем. Визначаються зв'язку між таблицями, а також обмеження цілісності даних. Отримана логічна модель перевіряється на відповідність правилам нормалізації.

База даних складається з 4 таблиць:

- «Клієнти» (таблиця 2.1);
- «Заміри» (таблиця 2.2);
- «Відвідування» (таблиця 2.3);
- «Абонементи» (таблиця 2.4).

Таблиця 2.1 – Склад відношення «Клієнти»

Атрибут	Тип даних	Допустимі значення	Обов'язковість	Примітки
Код_клієнта	Лічильник	Натуральні числа	Так	Первинний ключ
ПІБ	Текстовий	Кирилиця	Так	
Email	Текстовий	Кирилиця	Ні	
Дата народження	Текстовий	Кирилиця	Ні	
Місце роботи	Текстовий	Кирилиця	Ні	
Коментарі	Текстовий	Кирилиця	Ні	

Таблиця 2.2 – Склад відношення «Заміри»

Атрибут	Тип даних	Допустимі значення	Обов'язковість	Примітки
Код_клієнта	Числовий	Натуральні числа	Так	Зовнішній ключ
Дата_замірів	Лічильник	Натуральні числа	Так	Первинний ключ
Вага	Числовий	Натуральні числа	Так	
Зріст	Числовий	Натуральні числа	Так	
Об'єм_грудей	Числовий	Натуральні числа	Так	
Об'єм_під_груддю	Числовий	Натуральні числа	Так	
Об'єм_талії	Числовий	Натуральні числа	Так	
Об'єм_живота	Числовий	Натуральні числа	Так	
Об'єм_стегон	Числовий	Натуральні числа	Так	
Обхват_руки	Числовий	Натуральні числа	Так	
Обхват_гомілки	Числовий	Натуральні числа	Так	

Таблиця 2.3 – Склад відношення «Відвідування»

Атрибут	Тип даних	Допустимі значення	Обов'язковість	Примітки
Код_клієнта	Числовий	Натуральні числа	Так	Зовнішній ключ
Дата_відвідування	Текстовий	Кирилиця	Так	Первинний ключ

Таблиця 2.4 – Склад відношення «Абонементи»

Атрибут	Тип даних	Допустимі значення	Обов'язковість	Примітки
Код_клієнта	Числовий	Натуральні числа	Так	Зовнішній ключ
Вид_абонементу	Текстовий	Кирилиця	Так	Первинний ключ

Продовження таблиці 2.4

Атрибут	Тип даних	Допустимі значення	Обов'язковість	Примітки
Початок_абонементу	Текстовий	Кирилиця	Так	
Кінець_абонементу	Текстовий	Кирилиця	Так	

На основі проведеного аналізу будується ER-модель бази даних, яка представлена на рис. 2.8.

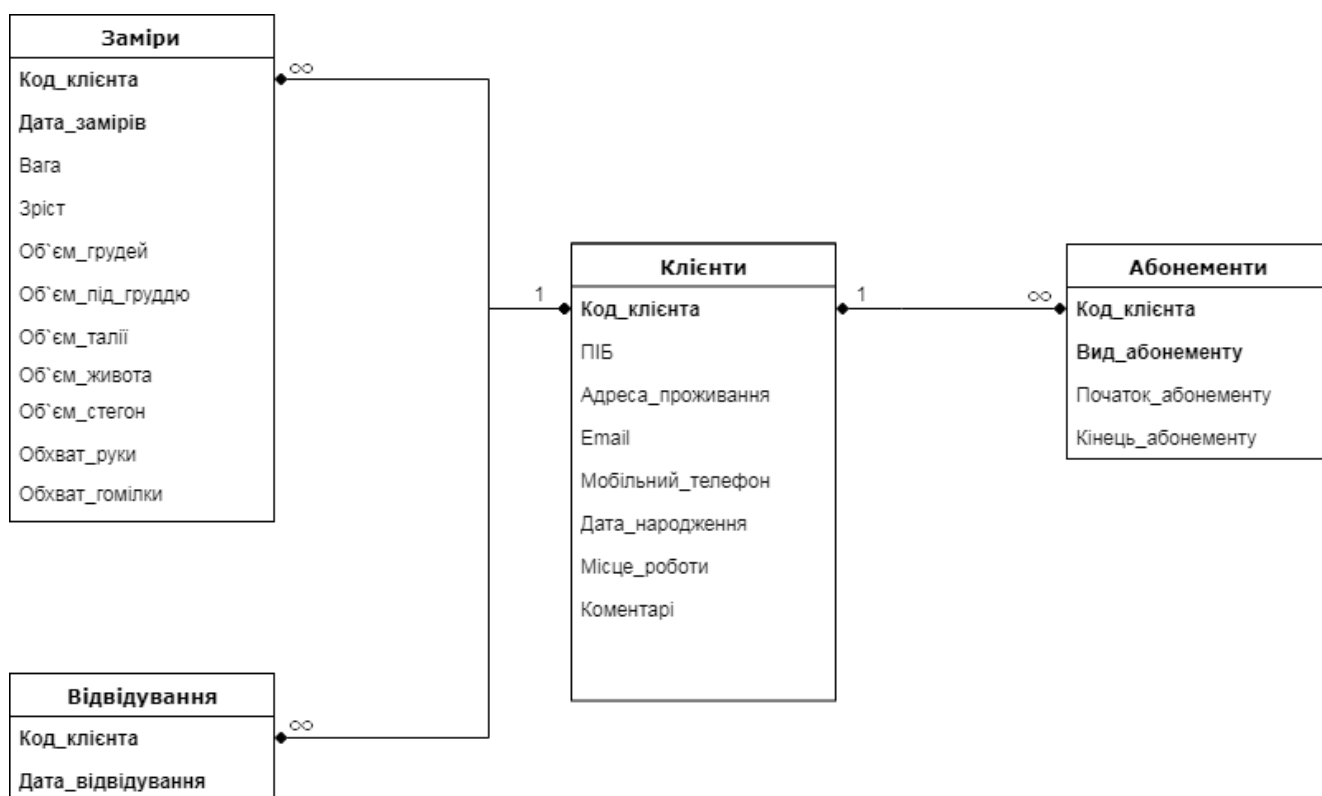


Рисунок 2.8 – ER-модель бази даних

### 3 РОЗРОБКА КОМП'ЮТЕРНОЇ СИСТЕМИ

#### 3.1 Алгоритм роботи модуля з боку користувача

Коли остаточно було вирішено розроблювати власний скрипт із можливістю моніторингу зовнішніх показників тіла клієнта, потрібно було розробити детальний алгоритм роботи цієї системи.



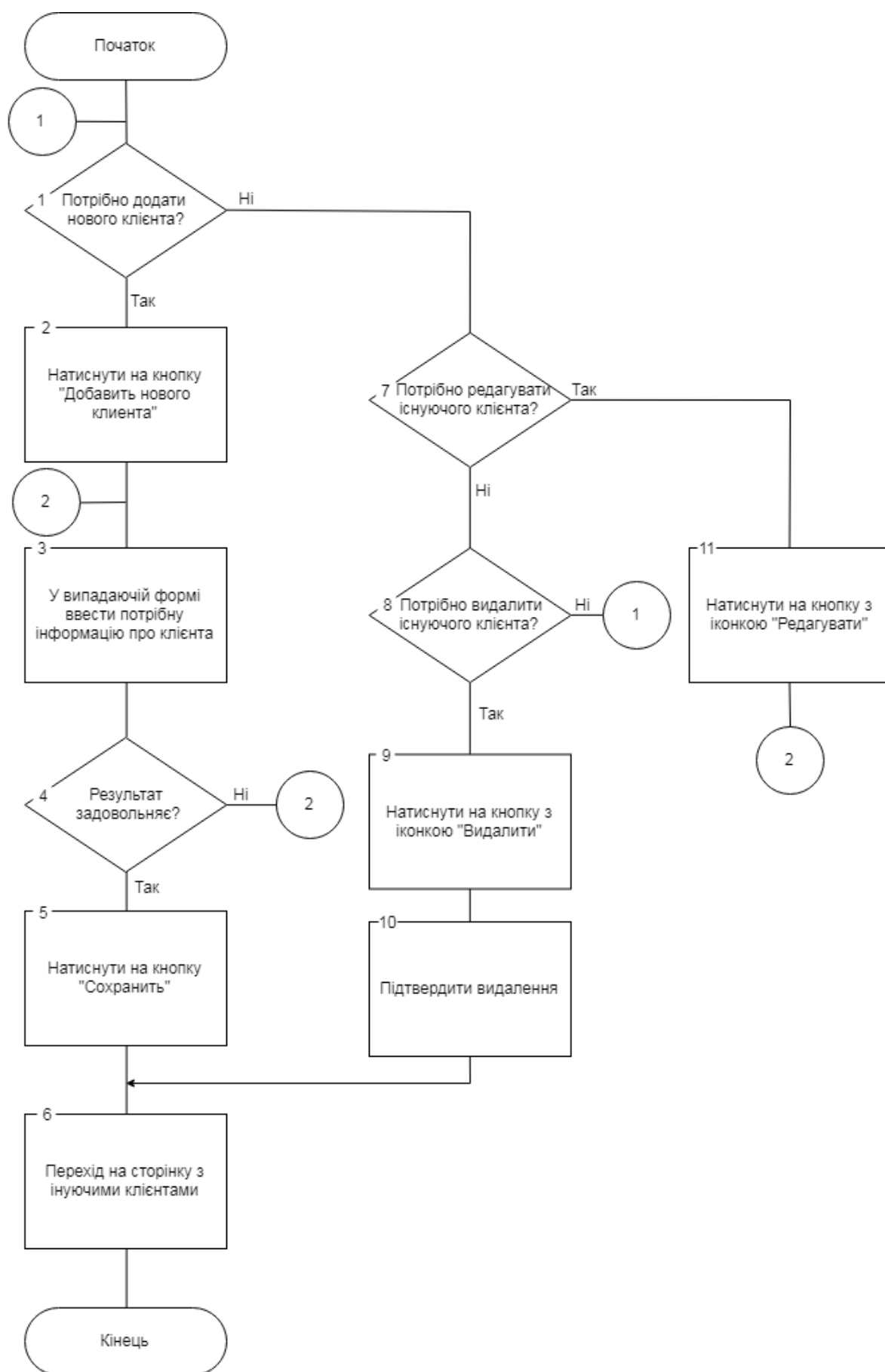


Рисунок 3.1 – Алгоритм внесення персональних даних клієнта

Системи повинна бути доступною та зрозумілою з першого погляду для як тренерів, так і для клієнтів котрі будуть переглядати свої показники. Вона повинна працювати з мінімальним втручанням адміністратора та працювати без збоїв. Тому що кожен збій у системі – це один втрачений клієнт. Алгоритм внесення персональних даних клієнта показано на рисунку 3.1.

### **3.2 Алгоритм роботи модуля з боку адміністратора**

Алгоритм роботи з боку користувача дуже легкий та доступний. Та з боку адміністратора теж ніяких складностей виникнути не повинно.

Коли адміністратор входить до адмін панелі за допомогою програми PgAdmin3, одразу на головній сторінці він бачить базу даних з створеними таблицями та їх полями. Для того щоб побачити дані які ввів користувач(тренер), адміністратору потрібно клацнути правою кнопкою по необхідній таблиці та обрати «Просмотр данных»,у випадаючому списку «Просмотреть первые 100 строк», у діалоговому вікні що відкрився адміністратор побачить сто останніх даних що були заповнені тренером. У цьому вікні адміністратор також може додавати дані які після збереження синхронізуються з веб-сайтом. Адміністратор зможе додати таку інформацію як: ПІБ, адресу проживання, email, мобільний телефон, дату народження, місце роботи, коментарі, вага, ріст, об`єм грудей, об`єм під груддю, об`єм талії, об`єм живота, об`єм стегон, обхват руки, обхват гомілки.

Отже все, що необхідно від адміністратора сайту при роботі з даним модулем підтримувати стабільну роботу всього модуля та бази даних, створити окрему сторінку для кожного тренера з окремим логіном та паролем, щоб кожен тренер зміг бачити інформацію тільки про своїх клієнтів. Панель адміністратора PgAdmin3 показана на рисунку 3.2

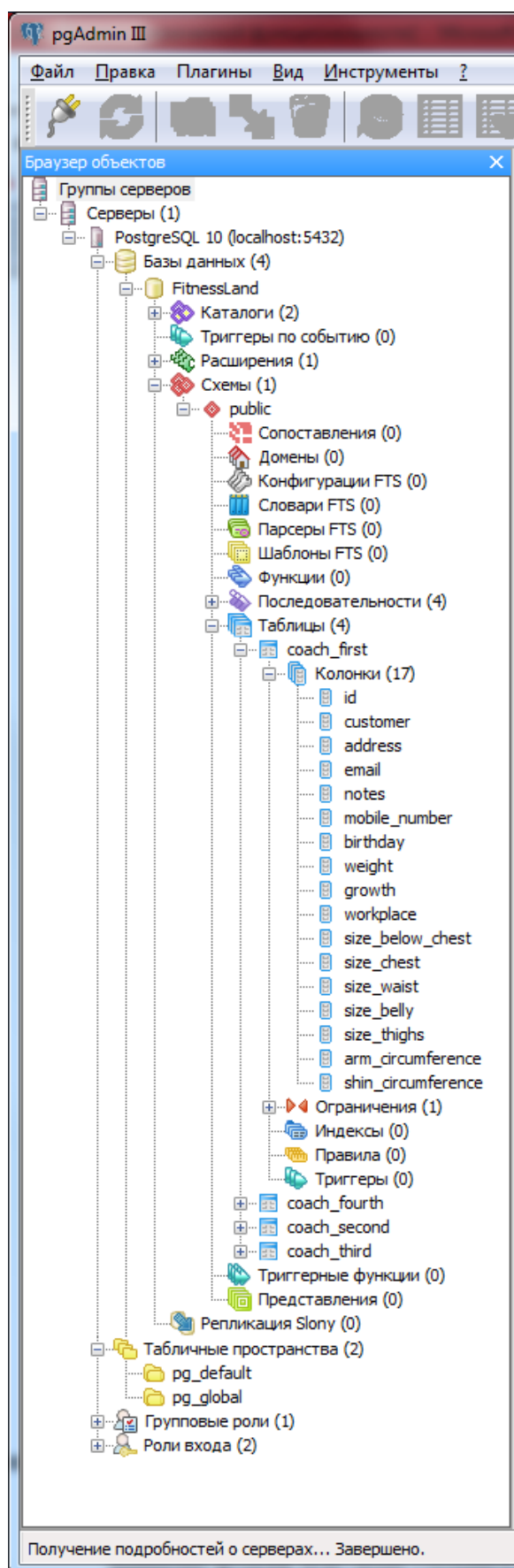
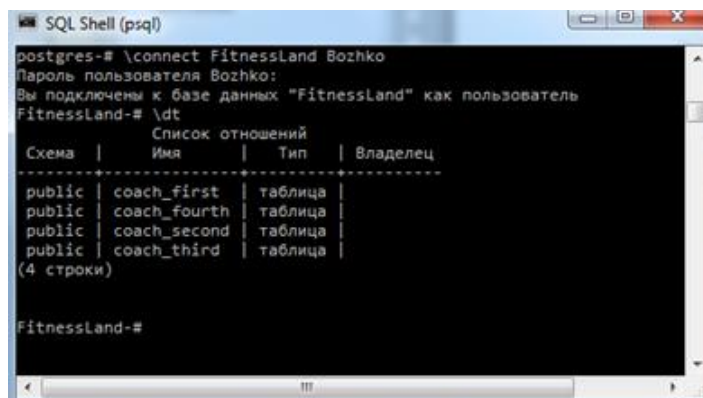


Рисунок 3.2 – Адміністративна панель фітнес-клубу

### 3.3 Розробка бази даних

Схема бази даних показана на рисунку 3.3.

Таблиці `coach_first`, `coach_second`, `coach_third`, `coach_fourth`, мають однакові поля, які містять інформацію про зовнішні дані клієнтів. Таблиці мають такі поля: ПІБ, адресу проживання, мобільний телефон, дату народження та місце роботи, вага, ріст, об'єм грудей, об'єм під груддю, об'єм талії, об'єм живота, об'єм стегон, обхват руки, обхват гомілки. На рисунку 3.4 зображено структуру таблиці `coach_first`.

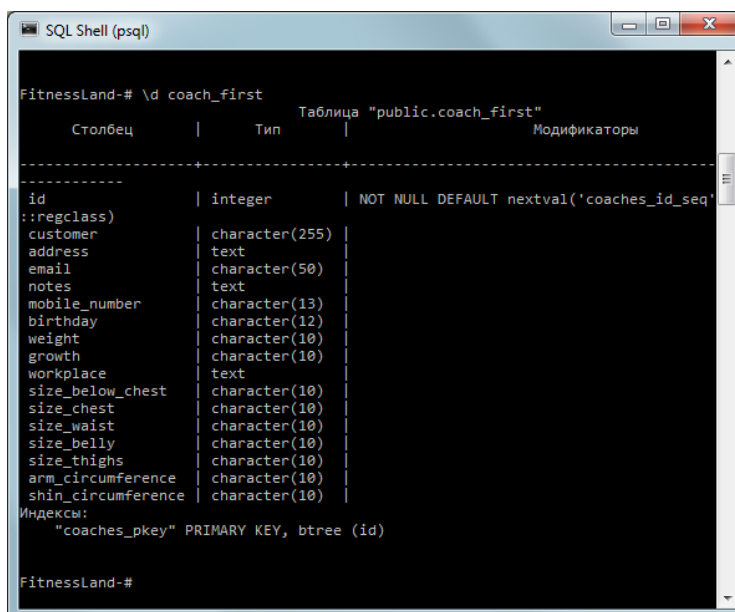


```

SQL Shell (psql)
postgres-# \connect FitnessLand Vozhko
Пароль пользователя Vozhko:
Вы подключены к базе данных "FitnessLand" как пользователь
FitnessLand-# \dt
          Список отношений
Схема |      Имя      | Тип | Владелец
-----+-----+-----+-----
public | coach_first  | таблица | public
public | coach_fourth | таблица | public
public | coach_second | таблица | public
public | coach_third  | таблица | public
(4 строки)

FitnessLand-#
  
```

Рисунок 3.3 – Схема бази даних



```

SQL Shell (psql)
FitnessLand-# \d coach_first
          Таблица "public.coach_first"
Столбец | Тип | Модификаторы
-----+-----+-----
id       | integer | NOT NULL DEFAULT nextval('coaches_id_seq'
::regclass)
customer | character(255) |
address  | text |
email    | character(50) |
notes    | text |
mobile_number | character(13) |
birthday | character(12) |
weight   | character(10) |
growth   | character(10) |
workplace | text |
size_below_chest | character(10) |
size_chest | character(10) |
size_waist | character(10) |
size_belly | character(10) |
size_thighs | character(10) |
arm_circumference | character(10) |
shin_circumference | character(10) |
Индексы:
 "coaches_pkey" PRIMARY KEY, btree (id)

FitnessLand-#
  
```

Рисунок 3.4 – Структура таблиці `coach_first`

### 3.4 Розробка веб-інтерфейсу користувача

Інтерфейс користувача - це сукупність засобів, за допомогою яких користувач взаємодіє з різними пристроями (з комп'ютером або побутовою технікою) або іншим складним інструментарієм (системою). Інтерфейс користувача - це такий різновид інтерфейсів, в якому з одного боку – людина, з іншого – машина (пристрій, програмне забезпечення).

Інтерфейс користувача часто розуміють лише як зовнішній вигляд програмного забезпечення, але таке розуміння є надто вузьким, оскільки саме за допомогою інтерфейсу користувач сприймає програму в цілому та використовує її функціональність. Інтерфейс користувача забезпечує підтримку прийняття рішень у визначеній предметній галузі та визначає порядок використання ПЗ і документації до нього. В дійсності, інтерфейс користувача об'єднує усі елементи і компоненти ПЗ, які здатні впливати на взаємодію користувача з програмним забезпеченням.

До таких елементів належать:

- набір задач, які користувач розв'язує за допомогою ПЗ;
- використовувана програмним забезпеченням метафора;
- елементи управління ПЗ;
- навігація між блоками ПЗ;
- візуальний (і не тільки)
- дизайн вікон та екранних форм програми та інші складові (рис. 3.5).



Рисунок 3.5 - Складові інтерфейсу користувача

### 3.5 Розробка веб-інтерфейсу модуля моніторингу інформації клієнтів

В даному підрозділі я опишу веб-інтерфейс модуля для моніторингу зовнішніх характеристик клієнтів фітнес-клубу. Як вже розповідалося, інтерфейс має бути зручним та доступним кожному тренеру, котрий заходить на сайт. Після роздумів, було вирішено створити мінімальний дизайн та мінімум кроків, що ведуть до додавання нового клієнта, змінення інформації про вже існуючого клієнта або видалення існуючих клієнтів. Макет сторінки модуля показана на рисунку 3.6.

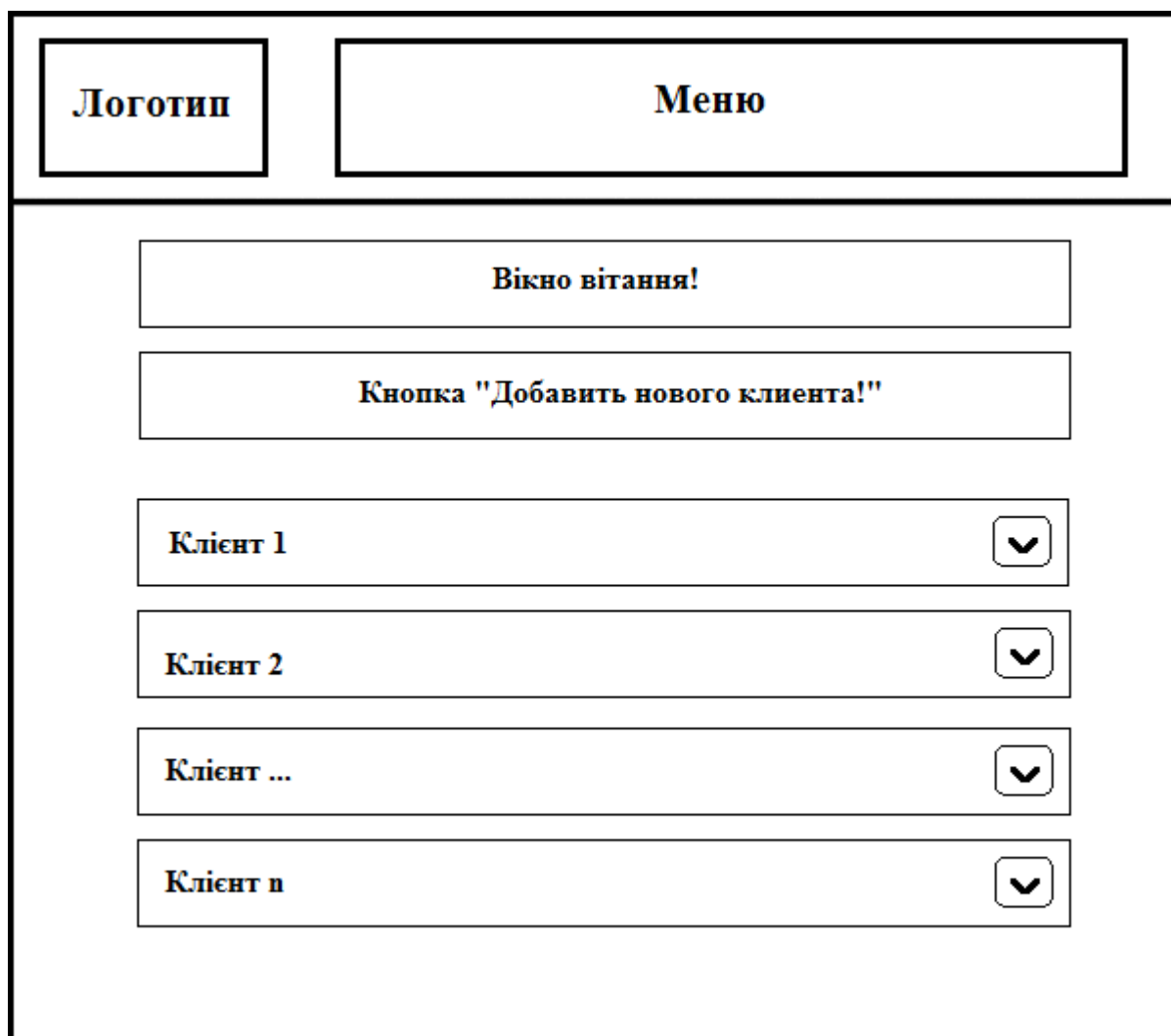


Рисунок 3.6 – Макет сторінки модуля моніторингу інформації про клієнтів

Коли зі структурою було вирішено, я перейшов до створення веб-інтерфейсу. Простий дизайн мав підкреслити простоту самого процесу додавання, видалення, редагування та перегляду інформації. Отже, веб-інтерфейс сторінки для моніторингу клієнтів зображено на рисунку 3.7.

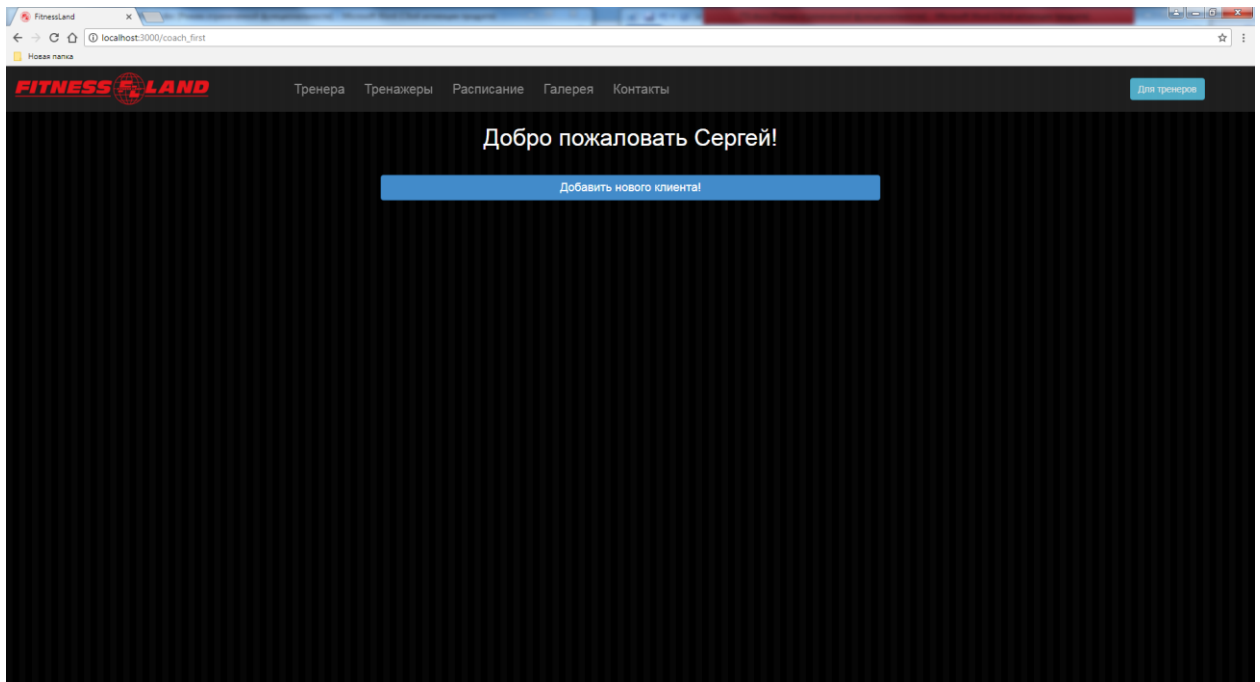


Рисунок 3.7 – Веб-интерфейс сторінки модуля моніторингу інформації про клієнтів

Але це лише, пустий макет, котрий можна побачити на даній сторінці. Приклад додавання нового клієнта зображено на рисунку 3.8.

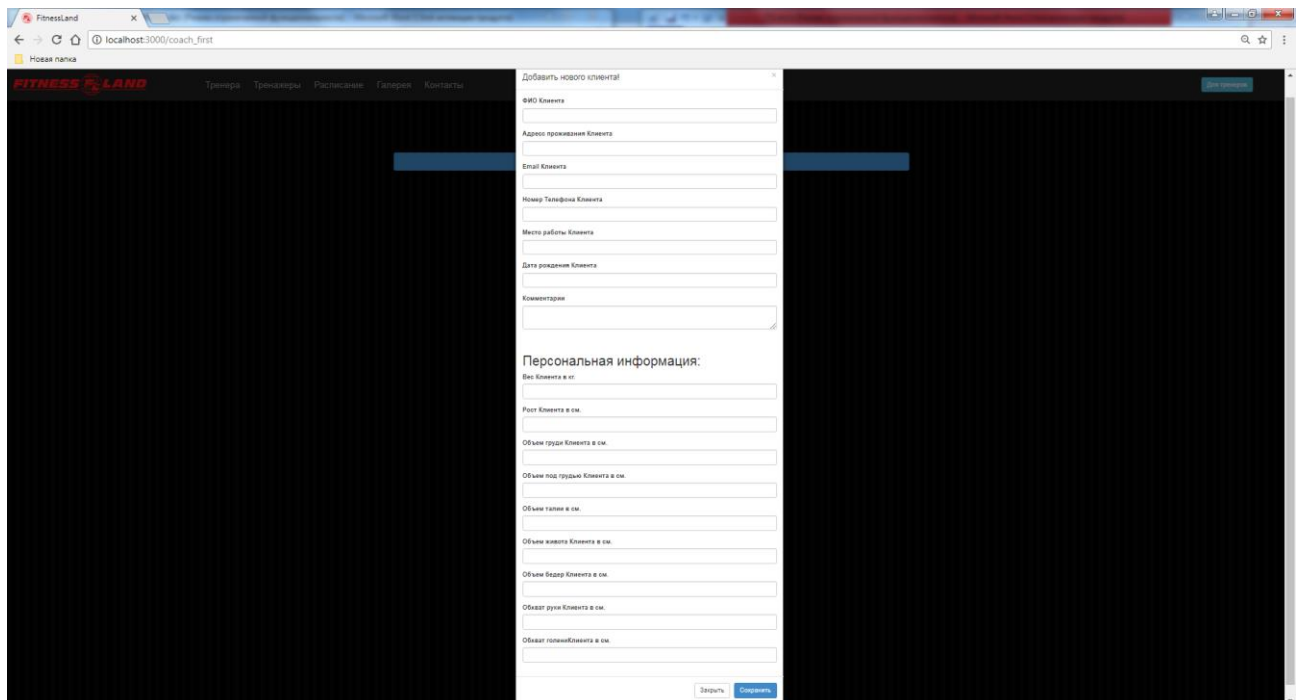


Рисунок 3.8 – Приклад додавання нового клієнта



Після натискання кнопки «Сохранить», на сторінці з'являється вікно з новим клієнтом, та якщо розгорнути це вікно ми побачимо всю інформацію про клієнта. Також дві кнопки для редагування та видалення існуючих клієнтів.

Отже, дизайн вийшов простий, але дуже зрозумілий для будь-якого тренера або клієнта.

У попередньому розділі я розробив дизайн свого модуля. Він вийшов простий, мінімалістичний та зрозумілий для кожного користувача. Дизайн самого сайту має занадто не відрізнятися. Тому у програмі Adobe Photoshop був розроблений простий дизайн. Далі згідно цього дизайну була розроблена верстка. Використовувалася мова CSS.

Головна сторінка – це інформація, яка постає перед користувачем при переході за адресою на сайт. Іншими словами, головна сторінка – це перше, з чим зустрічається відвідувач, потрапивши на сайт. Цьому правилу підпорядковуються всі сайти в Інтернеті: контент-провайдери, модні інтернет-магазини, потужні портали і багатолюдні форуми.

Призначення головної сторінки будь-якого сайту – це забезпечення такого «прийому» відвідувача, щоб, в ідеалі, він став користувачем. Або, принаймні, щоб затримався на сайті протягом тривалого часу.

На рисунку 3.9 представлений веб-інтерфейс головної сторінки розроблюваної web-системи, який містить область перегляду новин, спеціальних пропозицій, перелік послуг, панель швидкого доступу, кнопка паузи відео(для користувачів з повільною швидкістю Інтернету або слабким комп'ютером) та кнопка входу у систему та логотип,при натисканні на нього відбуватиметься повернення до головної сторінки.

**FITNESS LAND** Тренера Тренажеры Расписание Галерея Контакты Регистрация Для тренеров

Новогодний эксперимент! Как и многие известные клубы мы решили попробовать выпустить годовые клубные карты по низкой цене 2018 грн. Количество карт ограничено (всего 100 шт)

### АБОНЕМЕНТЫ

- Абонемент на 1 МЕСЯЦ
- Абонемент на 3 МЕСЯЦА
- Абонемент на 6 МЕСЯЦЕВ
- Абонемент на 12 МЕСЯЦЕВ

### Услуги

**STRETCHING**  
Эффективная система тренировок, направленная на укрепление мышц брюшного пресса. Во второй части урока выполняются упражнения на развитие гибкости.

**АВТ**  
Самый урок для тренировки мышц ног и брюшного пресса. Класс предназначен для всех уровней подготовки.

### Последние новости

**ПРИТЯЖАЕМ НА ДЕНЬ РОЖДЕНИЯ!**  
Близится 5-й День рождения клуба "Планета Фитнес - Гроцка". Администрация клуба по традиции приготовила уникальную спортивно-развлекательную программу, специально разработанную именно в стенах клуба. Фуршет 20 ноября в 19:00 в тренажерном зале состоится турнир по народному танцу, участниками которого могут стать как мужчины, так и женщины, интересующиеся жизнью клуба.

**ЧЕМПИОНАТ ПО ПЛАВАНИЮ КЛУБОВ "FitnessLand"**  
10 ноября в аквазале бассейна клуба "FitnessLand" состоится очередной чемпионат по плаванию среди членских клубов "FitnessLand", участие в котором могут принять как взрослые, так и совсем юные любители плавания. К участию в заплывах среди детей в возрасте до 7 лет допускаются малыши и девочки данной возрастной категории, показавшие лучший результат на предварительных стартах внутриклубных соревнований.

**Турнир по Жиму штанги лежа, среди клубов компании FitnessLand.**  
20 октября в Делта Fitness пройдет очередной турнир по Жиму штанги лежа среди клубов компании Делта Нейдинг Спортсменами проводится по правилам ФТР в дивизионе зачет. Жим лежа без экипировки.  
Высшая категория:  
• Мужчины: 74, 83, 93, 105, 105+  
• Женщины: 52, 57, 63, 63+

Фитнес-студия "FITNESSLAND" - проспект Мотороудиненка, 58/2, Запорожье, Запорожская область, 69051  
3.7 ★★★★★ Отзовитесь 3  
Увеличить карту

Запорожье  
ШЕВЧЕНКОВСКИЙ РАЙОН

+38 093 976 0588  
+38 098 976 0588  
+38 098 976 0588

**FITNESS LAND**  
© All rights reserved  
FitnessLandZP@gmail.com

Рисунок 3.9 – Веб-інтерфейс головної сторінки розробленої web-системи

На рисунку 3.10 представлений веб-інтерфейс сторінки «Тренера» розроблюваної системи, який містить інформацію про всіх тренерів закладу та їх інформацію(режим роботи, посилання на соціальні мережі та мобільний телефон).

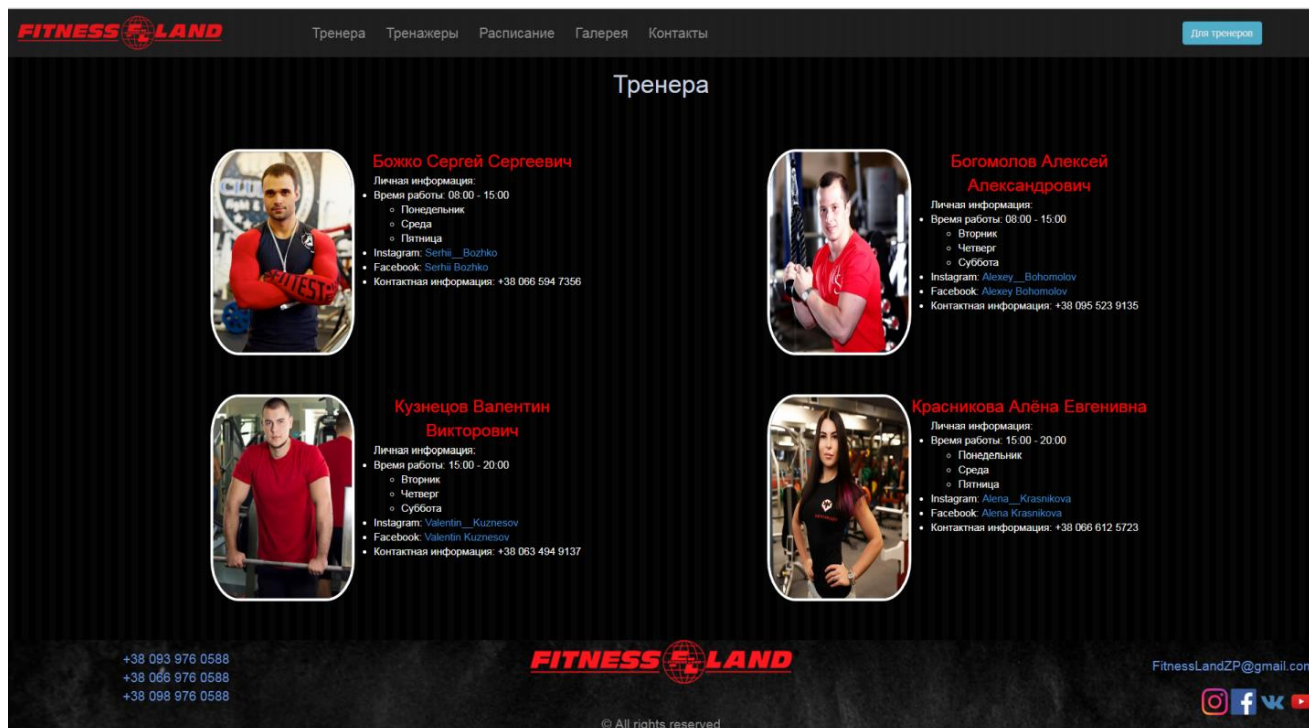


Рисунок 3.10 – Веб-інтерфейс сторінки «Тренера» розроблюваної web-системи

На рисунку 3.11 представлений веб-інтерфейс сторінки «Тренажеры» розроблюваної web-системи, який містить інформацію про фітнес-клуб, на які зони він поділений та коротку інформацію про ці зони.

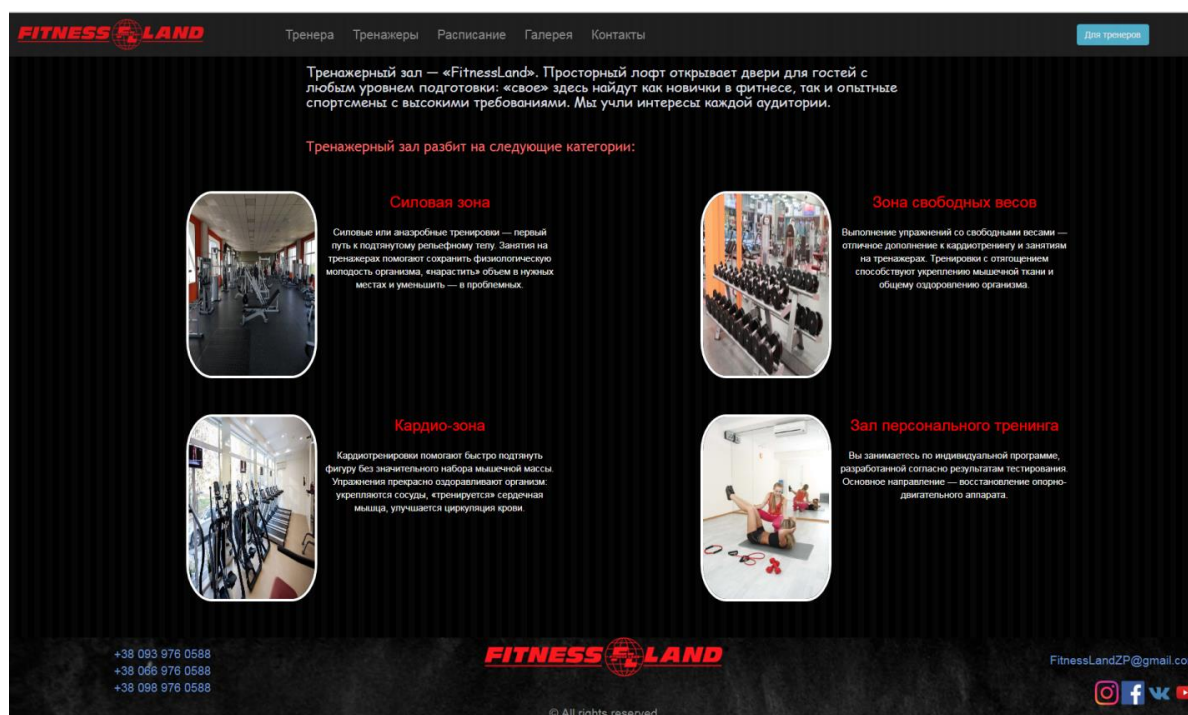


Рисунок 3.11 – Веб-интерфейс сторінки «Тренажеры» розроблюваної web-системи

На рисунку 3.12 представлений веб-інтрефейс сторінки «Расписание» розроблюваної web-системи, який містить розклад роботи всіх тренерів у зручному для користувача вигляді.

**FITNESS LAND** Тренера Тренажеры Расписание Галерея Контакты [Для тренеров](#)

**РАСПИСАНИЕ ЗАНЯТИЙ ФИТНЕС КЛУБА "FitnessLand"**

■ Божко Сергей ■ Красникова Алёна ■ Богомолов Алексей ■ Валентин Кузнецов

	Понедельник	Вторник	Среда	Четверг	Пятница	Суббота	Воскресенье
08:00	Божко Сергей	Богомолов Алексей	Божко Сергей	Богомолов Алексей	Божко Сергей	Богомолов Алексей	Выходной
09:00	Божко Сергей	Богомолов Алексей	Божко Сергей	Богомолов Алексей	Божко Сергей	Богомолов Алексей	Выходной
10:00	Божко Сергей	Богомолов Алексей	Божко Сергей	Богомолов Алексей	Божко Сергей	Богомолов Алексей	Выходной
11:00	Божко Сергей	Богомолов Алексей	Божко Сергей	Богомолов Алексей	Божко Сергей	Богомолов Алексей	Выходной
12:00	Божко Сергей	Богомолов Алексей	Божко Сергей	Богомолов Алексей	Божко Сергей	Богомолов Алексей	Выходной
13:00	Божко Сергей	Богомолов Алексей	Божко Сергей	Богомолов Алексей	Божко Сергей	Богомолов Алексей	Выходной
14:00	Божко Сергей	Богомолов Алексей	Божко Сергей	Богомолов Алексей	Божко Сергей	Богомолов Алексей	Выходной
15:00	Божко Сергей	Богомолов Алексей	Божко Сергей	Богомолов Алексей	Божко Сергей	Богомолов Алексей	Выходной
16:00	Красникова Алёна	Валентин Кузнецов	Красникова Алёна	Валентин Кузнецов	Красникова Алёна	Валентин Кузнецов	Выходной
17:00	Красникова Алёна	Валентин Кузнецов	Красникова Алёна	Валентин Кузнецов	Красникова Алёна	Валентин Кузнецов	Выходной
18:00	Красникова Алёна	Валентин Кузнецов	Красникова Алёна	Валентин Кузнецов	Красникова Алёна	Валентин Кузнецов	Выходной
19:00	Красникова Алёна	Валентин Кузнецов	Красникова Алёна	Валентин Кузнецов	Красникова Алёна	Валентин Кузнецов	Выходной
20:00	Красникова Алёна	Валентин Кузнецов	Красникова Алёна	Валентин Кузнецов	Красникова Алёна	Валентин Кузнецов	Выходной

+38 093 976 0588  
+38 066 976 0588  
+38 098 976 0588

**FITNESS LAND** FitnessLandZP@gmail.com

© All rights reserved.

Рисунок 3.12 – Веб-интерфейс сторінки «Расписание» розроблюваної web-системи

На рисунку 3.13 представлений веб-інтерфейс сторінки «Галерея» розроблюваної web-системи, який містить фотографії фітнес-клубу, які поділені на зони. На рисунку 3.14 представлено одне з відкритих зображень.



Рисунок 3.13 – Веб-інтерфейс сторінки «Галерея» розроблюваної web-системи

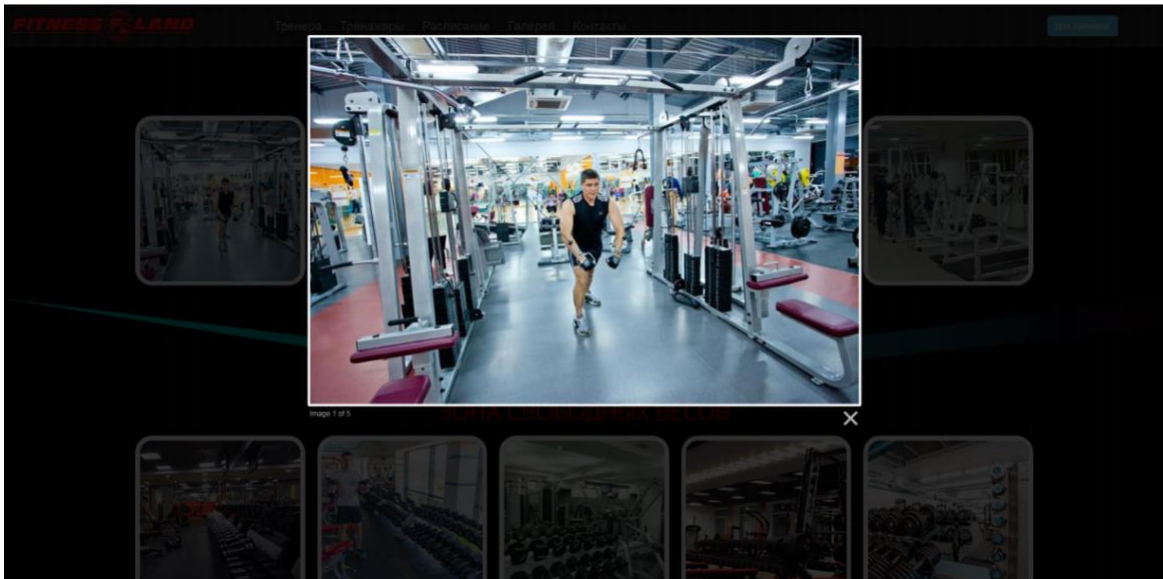


Рисунок 3.14 – Веб-інтерфейс сторінки «Галерея» з відкритим зображенням

На рисунку 3.15 представлений веб-інтерфейс сторінки «Контакты» розроблюваної web-системи, який містить контактну інформацію про фітнес-клуб та дві google карти (вид зі супутника та звичайна карта).

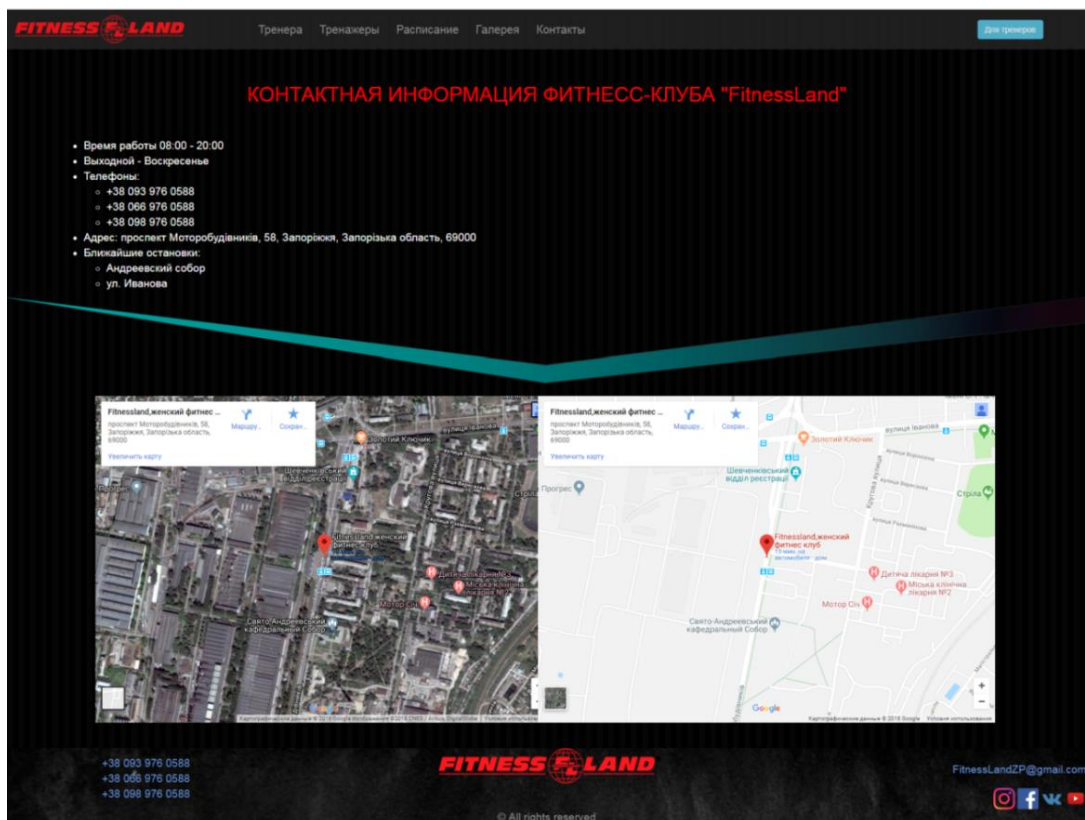


Рисунок 3.15 – Веб-інтерфейс сторінки «Контакты» розроблюваної web-системи

### 3.6 Тестування роботи системи

В попередньому розділі я повністю розробив та налаштував свій модуль (скрипт) для додавати нових клієнтів, редагувати інформації про клієнтів та видаляти клієнтів які перестали відвідувати фітнес-клубу. Тепер залишилося тільки перевірити та протестувати працездатність цього додатка. Я вирішив, що оскільки модуль повністю написаний з нуля, буде доцільно зробити перевірку тестування як з боку користувача, так і з боку адміністратора, котрий повинен відповідати за внесену інформацію у базу даних.

По-перше, треба протестувати кросбраузерність. Тобто перевірити, чи у всіх популярних браузерах добре відображається цей модуль, та чи коректно він у них працює.

Для цього я заходив на сторінку модуля з різних операційних систем та різних браузерів.

Отриманий після проведення тестування результат був задовільний. У найпопулярніших браузерах, таких як Firefox, Chrome, Opera, Yandex сторінка сайту не тільки відображалася як треба, а й коректно функціонувала. Навіть не тільки в останніх версіях. Також модуль не підвів й у IE 8.

Окрім ОС Windows, було проведене тестування й на ОС X Yosemite у браузері Safari. Ніяких проблем у роботі модуля я не зустрів.

Наступним кроком – була перевірка працездатності модуля з мобільного пристрою. Як на платформі Android, так і на iOS, весь сайт, а також сторінка створеного модулю відображалась коректно.

Далі мною було проведене тестування щодо додавання нових клієнтів, редагування інформації про клієнтів та видаляння клієнтів які перестали відвідувати фітнес-клубу, робота модулю також була коректною.

Отже, в ході цього тестування було показано, що модуль для моніторингу інформації про клієнтів працює як слід на різних операційних системах, у різних браузерах, та з мобільних пристроїв.

Тепер слід перевірити коректну роботу зі сторони адміністратора.

Спочатку додамо нового клієнта, в якому введемо всю інформацію(щоб всі поля були заповнені). Після введення усіх необхідних даних, новий клієнт був успішно створений та відображений на сторінці.

Тепер перейдемо до адміністративної частини, відкриємо програму pgAdmin3. У переліку останніх введених даних, побачимо щойно створеного нового клієнта.

Потім редагуємо інформацію про вже існуючого клієнта, та натиснемо кнопку «Сохранить». Після цих дій знову зайдемо у pgAdmin3, та побачимо зміни які я зробив в існуючим клієнтом.

І останнє це видалення існуючого клієнта. Після натиснення кнопки з іконкою видалення та підтвердження про видалення, з бази даних та з сторінки тренеру, зникають дані про клієнта.

Отже, після проведення повного тестування як клієнтської, так і адміністративної сторони, ми побачили, що модуль для моніторингу інформації про клієнтів працює на відмінно, дуже якісно та стабільно.



## ВИСНОВКИ

В ході виконання дипломного проєкту було розроблено комп'ютерну систему фітнес-клубу, яка дозволяє представити всю необхідну інформацію клієнтам та обробляти персональні дані клієнтів.

В результаті проведення аналізу предметної області були виділені та виконані наступні задачі:

- розроблено веб-додаток з повною інформацією фітнес-клубу для залучення нових клієнтів;
- розроблено БД, яка містить всі необхідні дані о клієнтах;
- розроблено веб-додаток для роботи з БД для внесення та редагування інформації.

Проведено опис предметної області функціонування фітнес-клубу, аналіз аналогічних систем, визначені вимоги до системи, проведено вибір програмних засобів для розробки комп'ютерної системи.

Розглянуті питання щодо проєктування комп'ютерної системи, подана функціональна схема системи, розроблені макети головних сторінок веб-додатку системи, проведено концептуальне та логічне проєктування бази даних, отримано ER-модель бази даних.

Розроблено комп'ютерну систему, використовуючи платформу Node.js та систему керування базою даних PostgreSQL. Проведено тестування розробленої системи.

В результаті виконання дипломного проєкту розроблено комп'ютерну систему фітнес-клубу, яка задовольняє всім поставленим вимогам.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Самойленко Л. Б., Застосування технологій електронної комерції для підвищення конкурентоспроможності вітчизняних підприємств. Ефективна економіка № 4, 2015. [Електронний ресурс] – Режим доступу: <http://www.economy.nayka.com.ua/?op=1&z=3997>.
2. JP Studio - студія персонального тренінга. [Електронний ресурс] – Режим доступу: <http://www.jp-studio.zp.ua>.
3. Фитнес в Запоріжжє: фитнес, пилатес, йога, зумба. [Електронний ресурс] – Режим доступу: <https://www.fitnessriver.com.ua>.
4. Лоусон, Б. Изучаем HTML5. Библиотека специалиста [Текст] / Хуан Гоше. – М. : Вильямс, 2014. – 936 с.
5. Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 / Р. Никсон – Питер, 2015. – 688 с.
6. Що таке PHP. [Електронний ресурс]. – М.: Документація PHP. [та ін.], 2021. – Режим доступу: <http://www.phpcoders.org.ua>.
7. Mario, C. Node.js Design Patterns [Текст] / Хуан Гоше. – М. : Casciaro, 2016. – 496 с.
8. Що таке Node.js? [Електронний ресурс]. – М.: Документація Node.js. [та ін.], 2021. – Режим доступу: <https://nodejs.org>.
9. Що таке MySQL? [Електронний ресурс]. – М.: Документація MySQL. [та ін.], 2021. – Режим доступу: <http://allweb.com.ua>.
10. Приходько, А. Adobe Illustrator для всех [Текст] / Анна Приходько. – М. : Эксмо, 2014. – 264 с.
11. Комолова, Н. Adobe Photoshop CC для всех [Текст] / Нина Комолова, Елена Яковлева. – М. : Лори, 2014. – 624 с.
12. Правила написання Software requirements specification; Архітектура додатків малою кров'ю; Створення архітектури програми [Електронний ресурс]. – Режим доступу: <https://habrahabr.ru>

13. Архітектура серверу та сайту [Електронний ресурс]: Огляд об'єктної моделі. – Режим доступу: [https://msdn.microsoft.com/ru-ru/library/ms473633\(v=office.12\).aspx](https://msdn.microsoft.com/ru-ru/library/ms473633(v=office.12).aspx)

## ДОДАТОК А

### Програмний код системи модуля моніторингу зовнішніх показників клієнтів

#### Лістинг А.1 - Файл index.js

```

var fs = require('fs');
var http = require('http');
var express = require('express');
var path = require('path');
var bodyParser = require('body-parser');
var cons = require('consolidate');
var dust = require('dustjs-helpers');
var pg = require("pg");
var app = express();
var connect =
"postgres://Bozhko:admin@localhost:5432/FitnessLand";

app.use(bodyParser.json());
app.use(bodyParser.urlencoded({extended: false }));
app.set('view engine', 'ejs');
app.use('/public', express.static('public'));
/*var server = http.createServer(function(req, res){
  console.log("Url of our page: " + req.url);
  res.writeHead(200, {'Content-Type': 'text/html; charset=utf-
8'});
  var readMainPage = fs.createReadStream(__dirname +
'/index.html', 'utf8');
  readMainPage.pipe(res);
});
server.listen(3000, '127.0.0.1');*/
app.get('/', function(req, res){
res.render('main')
});
app.get('/index_coach', function(req, res){
res.render('index_coach')
});
app.get('/staff', function(req, res){
res.render('staff')
});
app.get('/gallery', function(req, res){
res.render('gallery')
});
app.get('/timetable', function(req, res){
res.render('timetable')
});
app.get('/contacts', function(req, res){
res.render('contacts')
});
app.get('/coach_first', function(req, res){

pg.connect(connect, function(err, client, done) {
  if(err) {

```

```

        console.log("not able to get connection "+ err);
        res.status(400).send(err);
    }
    client.query('SELECT * FROM
coach_first',function(err,result) {
        if(err){
            console.log(err);
            res.status(400).send(err);
        }
        res.render('coach_first', {coach_first:
result.rows});
        done(); // closing the connection;
    });
});
app.post('/add_first', function(req, res){

    pg.connect(connect,function(err,client,done) {
        if(err){
            console.log("not able to get connection "+ err);
            res.status(400).send(err);
        }
        client.query("INSERT INTO coach_first(customer,
address, email, mobile_number, workplace, birthday, notes, weight,
growth, size_chest, size_below_chest, size_waist, size_belly,
size_thighs, arm_circumference, shin_circumference) VALUES($1, $2,
$3, $4, $5, $6, $7, $8, $9, $10, $11, $12, $13, $14, $15, $16)",
            [req.body.customer, req.body.address,
req.body.email, req.body.mobile_number, req.body.workplace,
req.body.birthday, req.body.notes,
            req.body.weight, req.body.growth,
req.body.size_chest, req.body.size_below_chest,
req.body.size_waist, req.body.size_belly, req.body.size_thighs,
req.body.arm_circumference, req.body.shin_circumference ]);
        done();
        res.redirect('/coach_first');
    });
});
app.delete('/delete_first/:id', function(req, res){
    pg.connect(connect,function(err,client,done) {
        if(err){
            console.log("not able to get connection "+ err);
            res.status(400).send(err);
        }
        client.query("DELETE FROM coach_first WHERE id =
$1",
            [req.params.id]);
        done();
        res.sendStatus(200);
    });
});
app.post('/edit_first', function(req, res){
    pg.connect(connect,function(err,client,done) {

```

```

        if(err){
            console.log("not able to get connection "+ err);
            res.status(400).send(err);
        }
        client.query("UPDATE coach_first SET
customer=$1, address=$2, email=$3, mobile_number=$4, workplace=$5,
birthday=$6, notes=$7, weight=$8, growth=$9, size_chest=$10,
size_below_chest=$11, size_waist=$12, size_belly=$13,
size_thighs=$14, arm_circumference=$15, shin_circumference=$16 WHERE
id = $17",
            [req.body.customer, req.body.address,
req.body.email, req.body.mobile_number, req.body.workplace,
req.body.birthday, req.body.notes,
            req.body.weight, req.body.growth,
req.body.size_chest, req.body.size_below_chest,
req.body.size_waist, req.body.size_belly, req.body.size_thighs,
req.body.arm_circumference, req.body.shin_circumference,
req.body.id]);
        done();
        res.redirect('/coach_first');
    });});
app.listen(3000);

```

## Лістинг А.2 - Файл coach\_first.ejs

```

<!DOCTYPE html>
<html lang="en">
<head>
    <title>FitnessLand</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.0/css/bootstrap.
min.css">
    <link
href="https://fonts.googleapis.com/css?family=Montserrat"
rel="stylesheet" type="text/css">
    <link href="https://fonts.googleapis.com/css?family=Lato"
rel="stylesheet" type="text/css">
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.0/jquery.min.j
s"></script>
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.mi
n.js"></script>
    <script src="/public/js/jquery-1.7.2.min.js"></script>
    <script src="/public/js/bootstrap.min.js"></script>
    <link rel="stylesheet" href="/public/css/stylesheet1.css" >
    <link rel="stylesheet" href="/public/css/footer.css" >
    <link rel="stylesheet" href="/public/css/admin.css" >
    <script src="/public/js/script.js"></script>

```

```

    <script src="/public/js/validation.js"></script>
    <link rel="shortcut icon" href="/public/images/icon1.ico"
type="image/x-icon">
    <script src="/public/js/coach_first.js"></script>
</head>
<body>
    <style>
    body{
        -webkit-perspective: 100%;
            perspective: 100%;
        -webkit-perspective-origin: center center;
            perspective-origin: center center;
        background: #000;
        background-image: -webkit-linear-gradient(left, rgba(57,
50, 255, 0.025) 50%, transparent 30%);
        background-image: linear-gradient(to right, rgba(255, 255,
255, 0.025) 50%, transparent 30%);
        background-size: 2rem 100%;
    }
    </style>
    <% include blocks/header_etc.ejs %>
    <h1 id="welcome">Добро пожаловать Сергей!</h1>
    <button type="button" class="btn btn-primary btn-block center-
block" id="add_btn" data-toggle="modal" data-target="#formModal">
        Добавить нового клиента!
    </button>
    <br />
    <br />
    <div class="">
        <% coach_first.forEach(function (coach) { %>
    <div class="well" id="admin_css">

        <h4><%= coach.customer %>
            <button class="btn btn-default pull-right" data-
toggle="collapse" href="#<%= coach.id %>" aria-expanded="false"
            aria-controls="<%= coach.id %>">
                <span class="glyphicon glyphicon-chevron-down" aria-
hidden="true"></span>
            </button></h4>
            <div class="collapse" id="<%= coach.id %>">
                <br />
                <p> <strong>Адресс:</strong> <%= coach.address
%></p>
                <p> <strong>Email:</strong> <%= coach.email %></p>
                <p> <strong>Номер телефона:</strong> <%=
coach.mobile_number %></p>
                <p> <strong>Место работы:</strong> <%=
coach.workplace %></p>
                <p> <strong>Дата рождения:</strong> <%=
coach.birthday %></p>
                <p> <strong>Комментарии:</strong> <%= coach.notes
%></p>
                <br>

```

```

        <h2>Персональная информация:</h2>
        <p> <strong>Вес:</strong> <%= coach.weight %></p>
        <p> <strong>Рост:</strong> <%= coach.growth %></p>
        <p> <strong>Объем груди:</strong> <%=
coach.size_chest %></p>
        <p> <strong>Объем под грудью:</strong> <%=
coach.size_below_chest %></p>
        <p> <strong>Объем талии:</strong> <%=
coach.size_waist %></p>
        <p> <strong>Объем живота:</strong> <%=
coach.size_belly %></p>
        <p> <strong>Объем бедер:</strong> <%=
coach.size_thighs %></p>
        <p> <strong>Обхват руки:</strong> <%=
coach.arm_circumference %></p>
        <p> <strong>Обхват голени:</strong> <%=
coach.shin_circumference %></p>
        <br>
        <button class ="btn btn-info edit-customer"
data-toggle="modal"
data-target="#editFormModal"
data-id="<%= coach.id %>"
data-customer="<%= coach.customer %>"
data-address="<%= coach.address %>"
data-email="<%= coach.email %>"
data-mobile_number="<%= coach.mobile_number %>"
data-workplace="<%= coach.workplace %>"
data-birthday="<%= coach.birthday %>"
data-notes="<%= coach.notes %>"
data-weight="<%= coach.weight %>"
data-growth="<%= coach.growth %>"
data-size_chest="<%= coach.size_chest %>"
data-size_below_chest="<%=
coach.size_below_chest %>"
data-size_waist="<%= coach.size_waist %>"
data-size_belly="<%= coach.size_belly %>"
data-size_thighs="<%= coach.size_thighs %>"
data-arm_circumference="<%=
coach.arm_circumference %>"
data-shin_circumference="<%=
coach.shin_circumference %>" >
        <span class="glyphicon glyphicon-edit" aria-
hidden="true"></span>
        </button>
        <button class ="btn btn-danger delete-customer"
data-id="<%= coach.id %>">
        <span class="glyphicon glyphicon-remove" aria-
hidden="true"></span>
        </button>
    </div>
</div>
<% }) %>
</div>

```



```

<!-- Add new Modal -->

<div class="modal fade" id="formModal" tabindex="-1"
role="dialog" aria-labelledby="formModalLabel">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <form method="post" action="/add_first" >

        <div class="modal-header">
          <button type="button" class="close" data-
dismiss="modal" aria-label="Close"><span aria-
hidden="true">&times;</span></button>
          <h4 class="modal-title" id="myModalLabel">Добавить
нового клиента!</h4>
        </div>
        <div class="modal-body">
          <div class="form-group">
            <label > ФИО Клиента</label>
            <input type="text" class="form-control"
name="customer" value="">
          </div>
          <div class="form-group">
            <label > Адрес проживания Клиента</label>
            <input type="text" class="form-control"
name="address" value="">
          </div>
          <div class="form-group">
            <label > Email Клиента</label>
            <input type="text" class="form-control" name="email"
value="">
          </div>
          <div class="form-group">
            <label > Номер Телефона Клиента</label>
            <input type="text" class="form-control"
name="mobile_number" value="">
          </div>
          <div class="form-group">
            <label > Место работы Клиента</label>
            <input type="text" class="form-control"
name="workplace" value="">
          </div>
          <div class="form-group">
            <label > Дата рождения Клиента</label>
            <input type="text" class="form-control"
name="birthday" value="">
          </div>
          <div class="form-group">
            <label > Комментарии</label>
            <textarea class="form-control"
name="notes"></textarea>
          </div>
          <br>

```

```

        <h2>Персональная информация:</h2>
        <div class="form-group">
            <label > Вес Клиента в кг.</label>
            <input type="text" class="form-control"
name="weight" value="">
        </div>
        <div class="form-group">
            <label > Рост Клиента в см.</label>
            <input type="text" class="form-control"
name="growth" value="">
        </div>
        <div class="form-group">
            <label > Объем груди Клиента в см.</label>
            <input type="text" class="form-control"
name="size_chest" value="">
        </div>
        <div class="form-group">
            <label > Объем под грудью Клиента в см.</label>
            <input type="text" class="form-control"
name="size_below_chest" value="">
        </div>
        <div class="form-group">
            <label > Объем талии в см.</label>
            <input type="text" class="form-control"
name="size_waist" value="">
        </div>
        <div class="form-group">
            <label > Объем живота Клиента в см.</label>
            <input type="text" class="form-control"
name="size_belly" value="">
        </div>
        <div class="form-group">
            <label > Объем бедер Клиента в см.</label>
            <input type="text" class="form-control"
name="size_thighs" value="">
        </div>
        <div class="form-group">
            <label > Обхват руки Клиента в см.</label>
            <input type="text" class="form-control"
name="arm_circumference" value="">
        </div>
        <div class="form-group">
            <label > Обхват голениКлиента в см.</label>
            <input type="text" class="form-control"
name="shin_circumference" value="">
        </div>
    </div>
    <div class="modal-footer">
        <button type="button" class="btn btn-default" data-
dismiss="modal">Закреть</button>
        <input type="submit" class="btn btn-primary"
value="Сохранить" />
    </div>

```

```

        </form>
      </div>
    </div>
  </div>

  <!-- Edit Modal -->

  <div class="modal fade" id="editFormModal" tabindex="-1"
role="dialog" aria-labelledby="editFormModalLabel">
    <div class="modal-dialog" role="document">
      <div class="modal-content">
        <form method="post" action="/edit_first" >
          <input type="hidden" id="edit-form-id" name="id">
            <div class="modal-header">
              <button type="button" class="close" data-
dismiss="modal" aria-label="Close"><span aria-
hidden="true">&times;</span></button>
              <h4 class="modal-title"
id="myModalLabel">Изменить информацию о клиенте!</h4>
            </div>
            <div class="modal-body">
              <div class="form-group">
                <label > ФИО Клиента</label>
                <input type="text" class="form-control"
name="customer" id="edit-form-customer">
              </div>
              <div class="form-group">
                <label > Адрес проживания Клиента</label>
                <input type="text" class="form-control"
name="address" id="edit-form-address">
              </div>
              <div class="form-group">
                <label > Email Клиента</label>
                <input type="text" class="form-control" name="email"
id="edit-form-email">
              </div>
              <div class="form-group">
                <label > Номер Телефона Клиента</label>
                <input type="text" class="form-control"
name="mobile_number" id="edit-form-mobile_number">
              </div>
              <div class="form-group">
                <label > Место работы Клиента</label>
                <input type="text" class="form-control"
name="workplace" id="edit-form-workplace">
              </div>
              <div class="form-group">
                <label > Дата рождения Клиента</label>
                <input type="text" class="form-control"
name="birthday" id="edit-form-birthday">
              </div>
              <div class="form-group">
                <label > Комментарии</label>

```

```

        <textarea class="form-control" name="notes"
id="edit-form-notes"></textarea>
    </div>
    <br>
    <h2>Персональная информация:</h2>
    <div class="form-group">
        <label > Вес Клиента в кг.</label>
        <input type="text" class="form-control"
name="weight" id="edit-form-weight">
    </div>
    <div class="form-group">
        <label > Рост Клиента в см.</label>
        <input type="text" class="form-control"
name="growth" id="edit-form-growth">
    </div>
    <div class="form-group">
        <label > Объем груди Клиента в см.</label>
        <input type="text" class="form-control"
name="size_chest" id="edit-form-size_chest">
    </div>
    <div class="form-group">
        <label > Объем под грудью Клиента в см.</label>
        <input type="text" class="form-control"
name="size_below_chest" id="edit-form-size_below_chest">
    </div>
    <div class="form-group">
        <label > Объем талии в см.</label>
        <input type="text" class="form-control"
name="size_waist" id="edit-form-size_waist">
    </div>
    <div class="form-group">
        <label > Объем живота Клиента в см.</label>
        <input type="text" class="form-control"
name="size_belly" id="edit-form-size_belly">
    </div>
    <div class="form-group">
        <label > Объем бедер Клиента в см.</label>
        <input type="text" class="form-control"
name="size_thighs" id="edit-form-size_thighs">
    </div>
    <div class="form-group">
        <label > Обхват руки Клиента в см.</label>
        <input type="text" class="form-control"
name="arm_circumference" id="edit-form-arm_circumference">
    </div>
    <div class="form-group">
        <label > Обхват голениКлиента в см.</label>
        <input type="text" class="form-control"
name="shin_circumference" id="edit-form-shin_circumference">
    </div>
</div>
<div class="modal-footer">

```

```

        <button type="button" class="btn btn-default" data-
dismiss="modal">Закрыць</button>
        <input type="submit" class="btn btn-primary"
value="Сохранить" />
    </div>
</form>
</div>
</div>
</div>
</div>
</body>

```

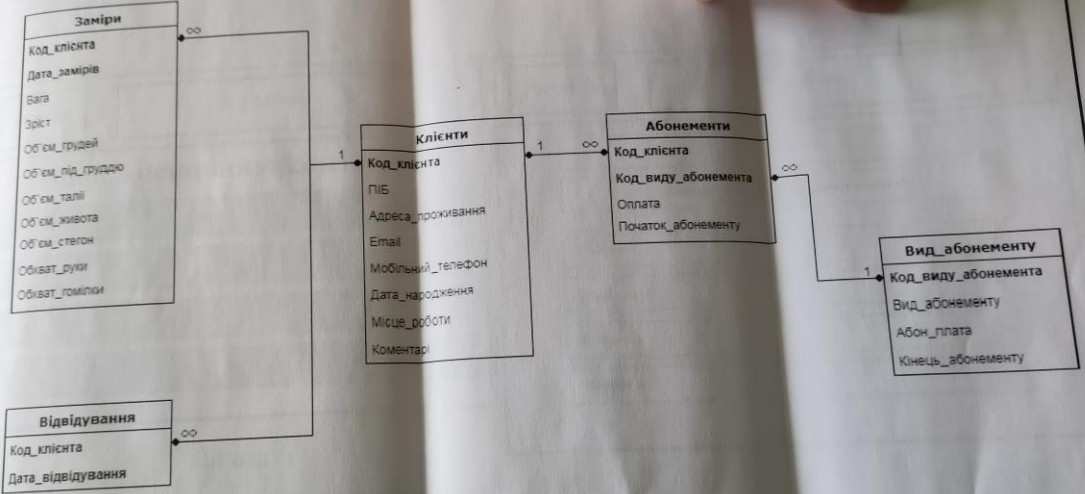
### Лісцінг А.3 - Файл coach\_first.js

```

$(document).ready(function(){
    $('#delete-customer').on('click', function(){
        console.log('Удаление клиента...');
        var id = $(this).data('id');
        var url = '/delete_first/'+id;
        if(confirm('Удалить Клиента?')){
            $.ajax({
                url: url,
                type: 'DELETE',
                success: function(result){
                    console.log('Удаление клиента...');
                    window.location.href='/coach_first';
                },
                error: function(err){
                    console.log(err);
                }
            });
        }
    });
});
$('#edit-customer').on('click', function(){
    $('#edit-form-customer').val($(this).data('customer'));
    $('#edit-form-address').val($(this).data('address'));
    $('#edit-form-email').val($(this).data('email'));
    $('#edit-form-
mobile_number').val($(this).data('mobile_number'));
    $('#edit-form-workplace').val($(this).data('workplace'));
    $('#edit-form-birthday').val($(this).data('birthday'));
    $('#edit-form-notes').val($(this).data('notes'));
    $('#edit-form-weight').val($(this).data('weight'));
    $('#edit-form-growth').val($(this).data('growth'));
    $('#edit-form-size_chest').val($(this).data('size_chest'));
    $('#edit-form-
size_below_chest').val($(this).data('size_below_chest'));
    $('#edit-form-size_waist').val($(this).data('size_waist'));
    $('#edit-form-size_belly').val($(this).data('size_belly'));
    $('#edit-form-size_thighs').val($(this).data('size_thighs'));
    $('#edit-form-
arm_circumference').val($(this).data('arm_circumference'));

```

```
    $('#edit-form-  
shin_circumference').val($(this).data('shin_circumference'));  
    $('#edit-form-id').val($(this).data('id'));  
  });  
});
```



Зм.	Лист	№ документа	Підпис	Дата
Розробив		Гурєва К.О.	<i>[Signature]</i>	02.01
Перевірив		Кудерметов Р.К.	<i>[Signature]</i>	
Н. контр.		Щербак Н.В.	<i>[Signature]</i>	
Затвердив		Кудерметов Р.К.	<i>[Signature]</i>	

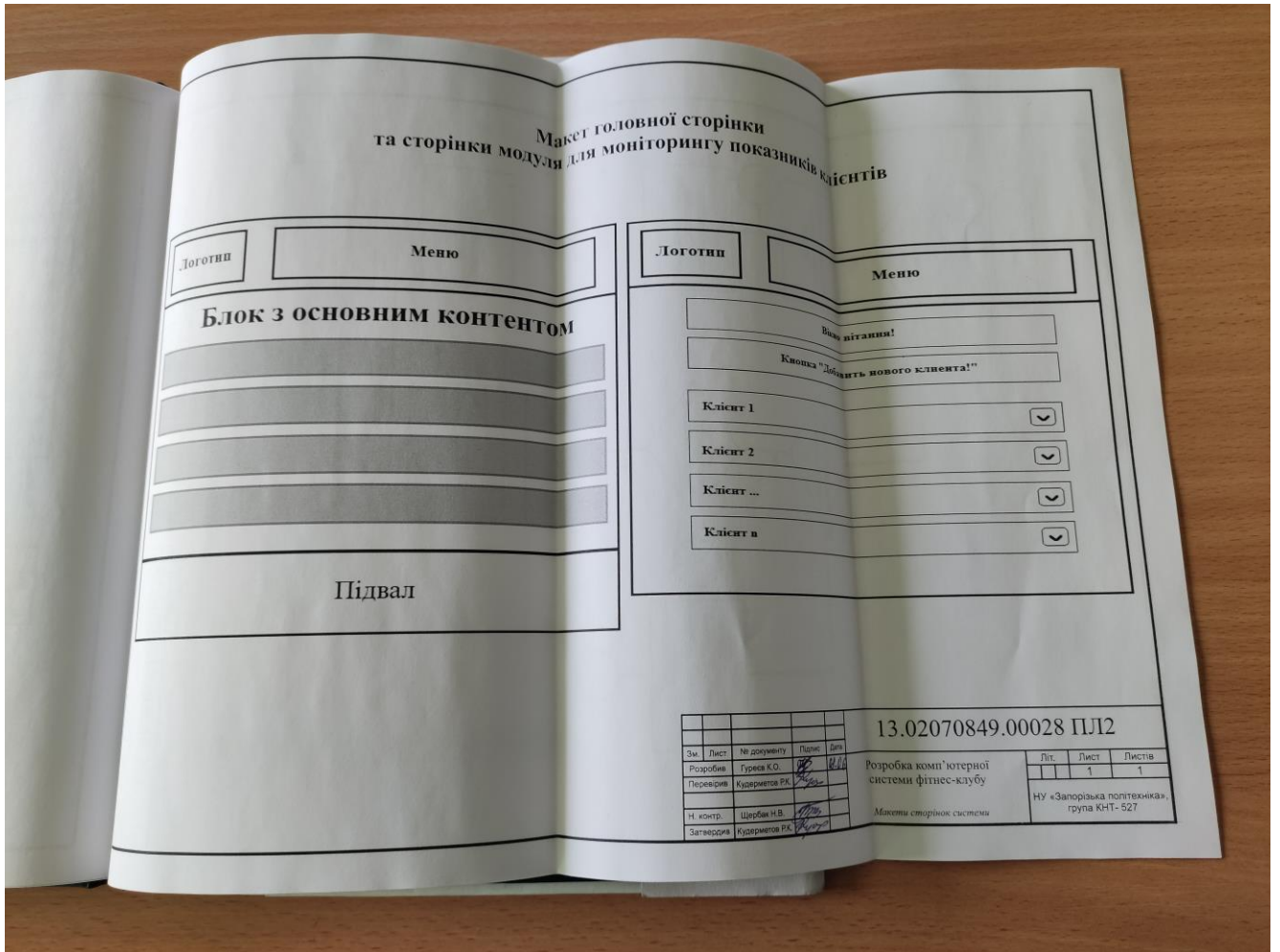
13.02070849.00028 ПЛ1

Розробка комп'ютерної системи фітнес-клубу

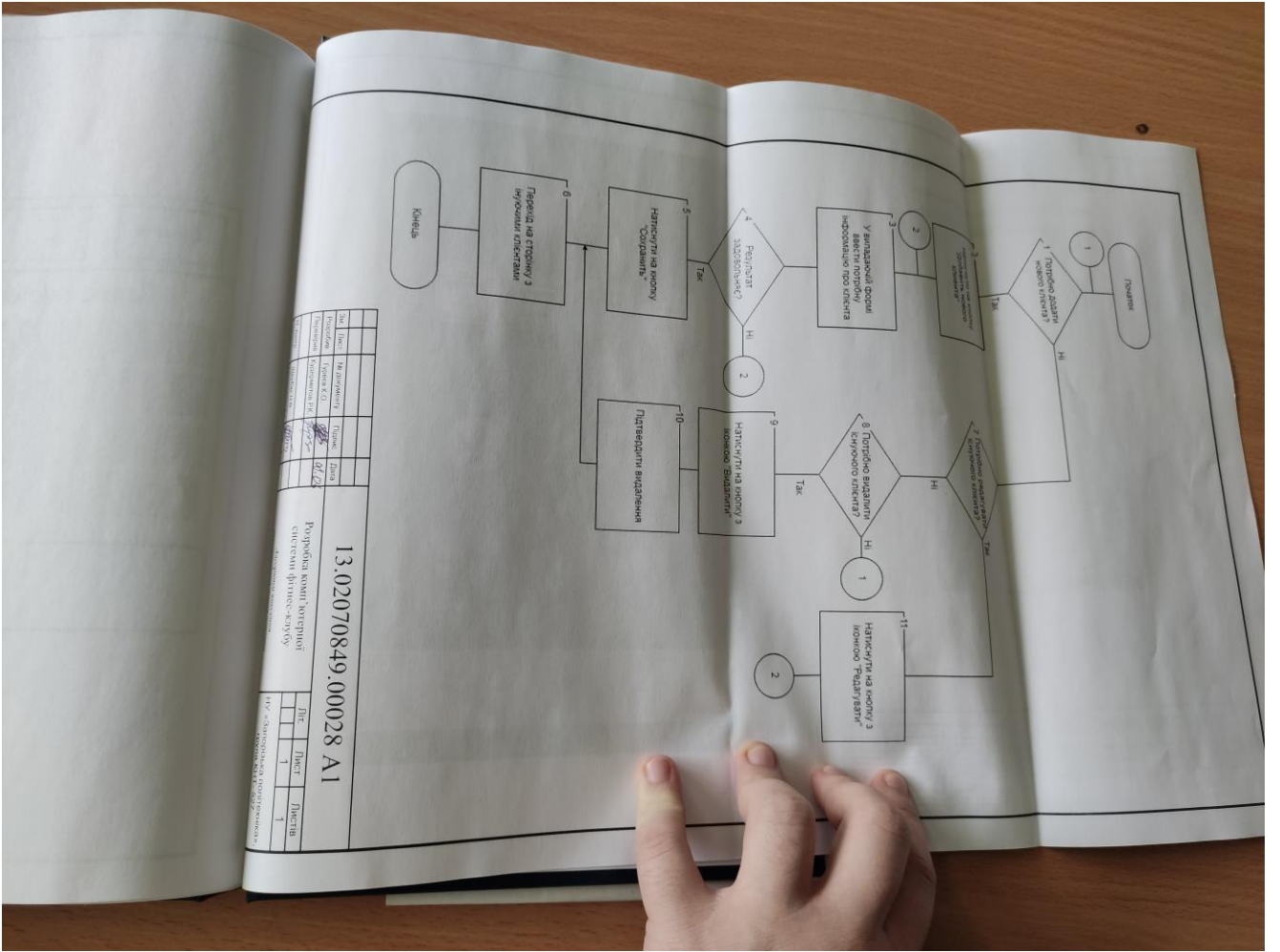
ER-модель

Літ.	Лист	Листів
1	1	1

НУ «Запорізька політехніка» група КНТ-527







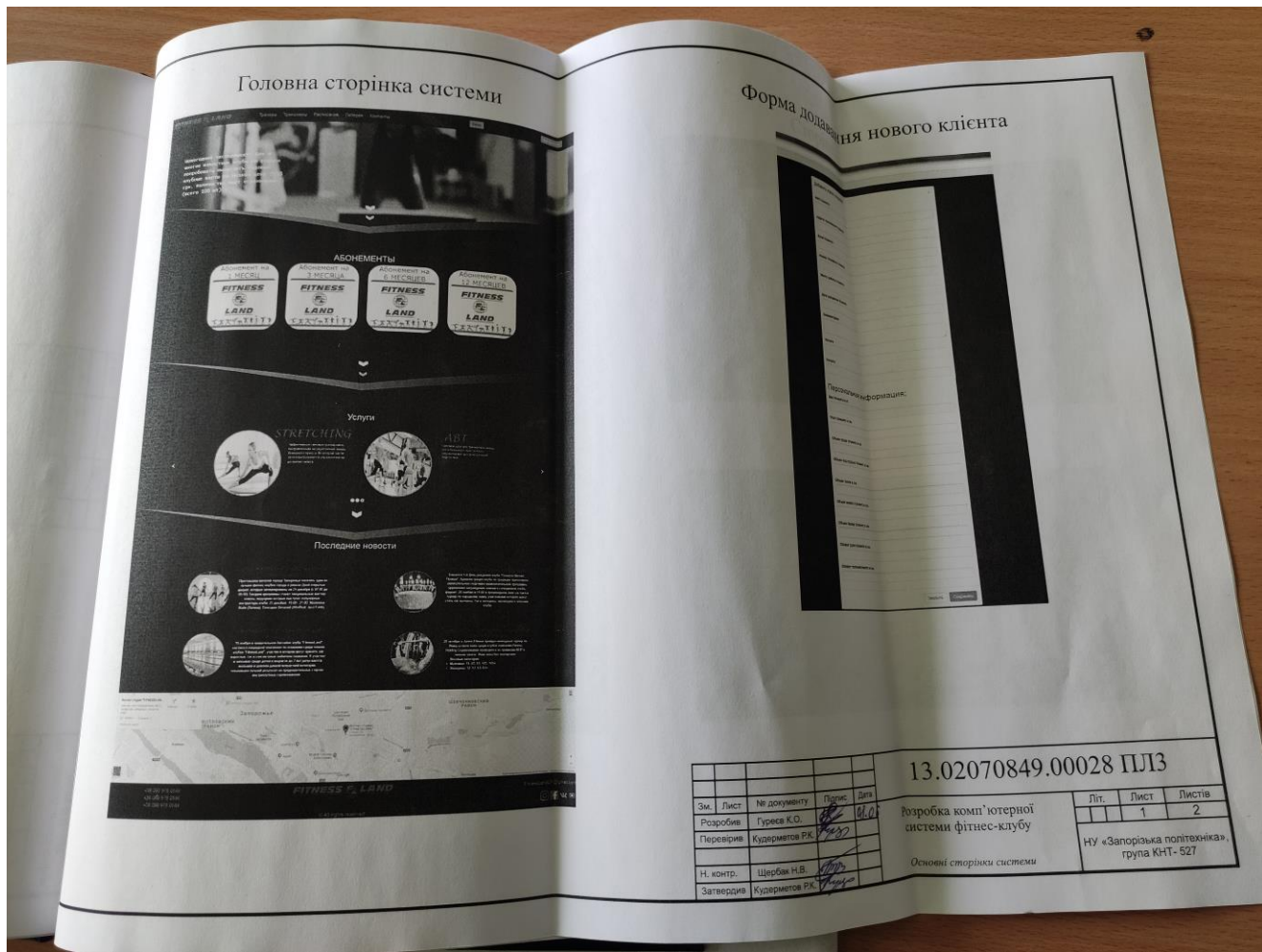
Знак	Лист	На документах	Таблиця	Листа
Продукція	Товари	КД	202	202
Продукція	Учбові посібники	ВР	202	202
Продукція	Учбові посібники	ВР	202	202

13.02070849.00028 A1

Розробка книг пошуку списком фінансування

Лист	Лист	Лист	Листів
1	1	1	1

14V - Сторінка 1 з 1



### Головна сторінка системи

### Форма додавання нового клієнта

Місія: Створити ідеальне місце для тренування, де кожен зможе досягти своїх цілей і стати кращим. Ми пропонуємо професійні тренування, які допоможуть вам стати кращим спортсменом і досягти своїх цілей.

АБОНЕМЕНТЫ

- Абонемент на 1 місяць
- Абонемент на 3 місяці
- Абонемент на 6 місяців
- Абонемент на 12 місяців

Услуги

- STRETCHING
- ART

Последние новости

13.02.2018 FITNESS LAND

Ім'я: \_\_\_\_\_

Прізвище: \_\_\_\_\_

Повне ім'я: \_\_\_\_\_

Адреса: \_\_\_\_\_

Телефон: \_\_\_\_\_

Е-mail: \_\_\_\_\_

Підписання форми: \_\_\_\_\_

13.02070849.00028 ПЛЗ

Зм.	Лист	№ документа	Писав	Дата
Розробив		Гурєв К.О.	<i>[Signature]</i>	06.02
Перевірив		Кудримов Р.К.	<i>[Signature]</i>	
Н. контр.		Шербак Н.В.	<i>[Signature]</i>	
Затвердив		Кудримов Р.К.	<i>[Signature]</i>	

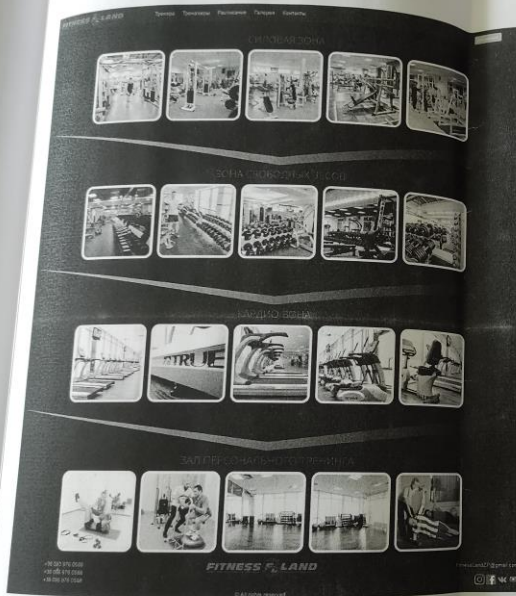
Розробка комп'ютерної системи фітнес-клубу

Основні сторінки системи

Літ.	Лист	Листів
	1	2

НУ «Запорізька політехніка», група КНТ-527

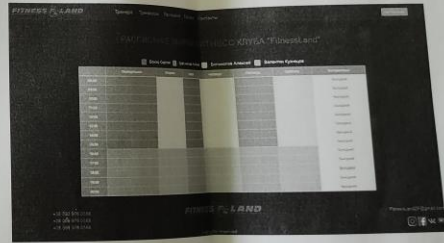
### Сторінка «Галерея»



### Сторінка «Тренера»



### Сторінка «Розклад»



Зм.	Лист	№ документи	Полос.	Дата
Розробив		Гурєва К.О.	<i>[Signature]</i>	2020
Перевірив		Кудерметов Р.А.	<i>[Signature]</i>	
Н. контр.		Щербак Н.В.	<i>[Signature]</i>	
Затвердив		Кудерметов Р.А.	<i>[Signature]</i>	

13.02070849.00028 ПЛІЗ

Розробка комп'ютерної системи фітнес-клубу

Освітня сторінка системи

Літ.	Лист	Листів
	2	2

НУ «Запорізька політехніка», група КНТ-527