

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту  
(повна назва)

Кафедра Інформатики  
(повна назва)

**КВАЛІФІКАЦІЙНА РОБОТА**  
**Пояснювальна записка**

рівень вищої освіти перший (бакалаврський)

**РОЗРОБКА ВЕБЗАСТОСУНКУ ДЛЯ СПОРТИВНОГО ЗАЛУ**  
(тема)

Виконав:  
студент 4 курсу, групи ІТІНФ-20-3

Спасьонов І.О.  
(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки  
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика  
(повна назва освітньої програми)

Керівник проф. Машталір С.В.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри \_\_\_\_\_  
(підпис)

Кобилін О.А.  
(прізвище, ініціали)

2024 р.

## Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту  
(повна назва)Кафедра Інформатики  
(повна назва)Рівень вищої освіти перший (бакалаврський)Спеціальність 122 Комп'ютерні науки  
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика  
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

«\_\_\_\_» \_\_\_\_\_ 2024 р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУстудентові Спасьонову Іллі Олександровичу  
(прізвище, ім'я, по батькові)1. Тема роботи Розробка вебзастосунку для спортивного залу

затверджена наказом університету від 20 травня 2024 року № 464 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 25 травня 2024 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі, мова гіпертекстової розмітки HTML5, каскадні таблиці стилів CSS, мова програмування JavaScript, платформа розробки Node.js, бібліотека для розробки інтерфейсів з відкритим кодом React, документо-орієнтована система керування базами даних з відкритим кодом MongoDB, середовище розробки Microsoft Visual Studio Code.

4. Перелік питань, що потрібно опрацювати в роботі

1. Огляд існуючих вебзастосунків для спортивного залу по всьому світу.

2. Проектування архітектури застосунку.

3. Моделювання бази даних для вебсайту.

4. Розробка вебзастосунку для спортивного залу.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Актуальність проблеми спортивних залів, постановка задачі, діаграма прецедентів для вебзастосунку спортивного залу, діаграма послідовностей для вебзастосунку спортивного залу, зображення результатів тестування програми.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	08.04.2024	
2	Аналіз завдання, підбір літератури	08.04.24-12.04.24	
3	Аналіз літератури з досліджуваної проблеми	13.04.24-15.04.24	
4	Аналіз технічних засобів	16.04.24-17.04.24	
5	Розробка алгоритмів	18.04.24-23.05.24	
6	Програмна реалізація	24.04.24-17.05.24	
7	Оформлення пояснювальної записки	18.05.24-23.05.24	
8	Перевірка на плагіат	27.05.2024	
9	Рецензування	28.05.2024	
10	Підготовка презентації та доповіді	29.05.24-04.06.24	
11	Занесення роботи в електронний архів	05.06.2024	
12	Попередній захист кваліфікаційної роботи	05.06.2024	

Дата видачі завдання 8 квітня 2024 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_ проф. Машталір С.В.  
(підпис) (посада, прізвище, ініціали)

## РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 57 с., 35 рис., 30 джерел.

СПОРТИВНИЙ ЗАЛ, ВЕБЗАСТОСУНОК, МОВА ПРОГРАМУВАННЯ JAVASCRIPT, ФРЕЙМБОРК REACT, МОВА ГІПЕРТЕКСТОВОЇ РОЗМІТКИ HTML5, КАСКАДНІ ТАБЛИЦІ СТИЛІВ CSS, MONGODB.

Об'єктом роботи є створення вебзастосунку для спортивного залу.

Метою роботи є розробка зручного та багатофункціонального застосунку для автономного спортивного залу.

Додаткові функції були розроблені для забезпечення унікальності продукту. Вебсайт призначений для користування клієнтам та адміністраторам. Клієнт має змогу оформлювати підписку на спортзал та мати доступ до довідника з вправами для тренувань. Для адміністратора створено функції для роботи з обліковими записами клієнтів, а також перегляд статистики вебсайту.

GYM, WEB APPLICATION, JAVASCRIPT PROGRAMMING LANGUAGE, REACT FRAMEWORK, HTML5 HYPERTEXT MARKUP LANGUAGE, CSS CASCADE STYLE SHEETS, MONGODB.

The object of the work is to create a web application for a gym.

The purpose of the work is to develop a convenient and multifunctional application for an autonomous gym.

Additional features have been developed to ensure the uniqueness of the product. The website is intended for use by customers and administrators. The client can sign up for a gym and have access to a guide with exercises for training. Functions for working with customer accounts and viewing website statistics have been created for the administrator.

## ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів .....	6
Вступ.....	7
1 Аналіз предметної області та постановка задачі .....	8
1.1 Сучасний стан розвитку вебзастунків для спортивних залів.....	8
1.2 Поточний аналіз стану вебсайтів для спортивних залів у інтернет просторі.....	13
1.3 Тенденція спортзалів без персоналу, керованих за допомогою вебпорталу .....	14
1.4 Постановка задачі.....	15
2 Моделювання системи вебзастосунку .....	16
2.1 Специфікація вимог до застосунку.....	16
2.2 Моделювання структури та наповнення вебзастосунку для спортивного залу .....	17
2.3 Методики Frontend частини .....	20
2.4 Методики Backend частини .....	25
2.5 Проєктування бази даних застосунку.....	28
3 Розробка вебзастосунку для спортивного залу.....	32
3.1 Вибір інструментальних засобів для реалізації поставленої задачі .....	32
3.2 Етапи розробки вебзастосунку для спортивного залу.....	35
3.3 Тестування розробленого вебзастосунку та аналіз результатів.....	37
3.4 Перспективи подальшої роботи.....	52
Висновки.....	54
Перелік джерел посилання .....	55

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

MVC – Model View Controller (модель вид контролер)

API – Application Programming Interface (інтерфейс програмного застосунку)

SQL – Structured Query Language (структурована мова запитів)

NoSQL – Not only SQL (не тільки структурована мова запитів)

JSON – JavaScript Object Notation (запис об'єктів JavaScript)

REST – Representational State Transfer (передача представницького стану)

HTML – HyperText Markup Language (мова розмітки гіпертексту)

CSS – Cascading Style Sheets (каскадні таблиці стилів)

UDF – Unidirectional Data Flow (односпрямований потік даних)

HTTP – Hypertext Transfer Protocol (протокол передачі гіпертексту)

QR – Quick Response (швидка відповідь)

ІМТ – індекс маси тіла

## ВСТУП

Робота над створенням вебзастосунку спортивного залу вкрай актуальна в сучасному світі, оскільки спостерігається зростаючий попит на фізичну активність та здоровий спосіб життя. З урахуванням загального збільшення обізнаності про важливість здорового способу життя та розуміння впливу регулярних тренувань на фізичне й психічне здоров'я, люди все більше віддають перевагу відвідуванню спортивних закладів.

Створення вебсайту для спортивного залу дає можливість клієнтам зручно та ефективно користуватися послугами залу. Він надає можливість зареєструватися, придбати абонементи та здійснювати оплату онлайн, що значно спрощує процес для користувачів.

З урахуванням поширення тенденції до онлайн-покупок і зручності використання вебплатформ, створення вебсайту для тренажерного залу стає важливим елементом конкурентоспроможності та забезпечує закладу доступ до широкого кола клієнтів. Більше того, розвиток технологій і впровадження нових функцій на вебсайті, таких як особисті кабінети з персоналізованими тренувальними програмами і порадами, роблять його ще більш привабливим для клієнтів і сприяють їх задоволенню від використання послуг тренажерного залу.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

## 1.1 Сучасний стан розвитку вебзастосунків для спортивних залів

Створення вебзастосунку в сьогоднішні дні це дуже відповідальний і трудомісткий процес. у нинішніх реаліях інтернет технологія вийшла на зовсім новий рівень з множинними перевагами та технологіями, що спрощують цю діяльність. Для створення вебсайту необхідно володіти відразу кількома навичками, такими як створення дизайну, який залучатиме користувачів, грамотне написання frontend і backend частини, завдяки якому сайт буде зрозумілий і простий у використанні, а також тестування, щоб знайти недоліки та виправити всі помилки перед релізом.

Неможливо уявити сучасне життя без вебзастосунків, адже якісно створений вебсайт – це перший шлях до успішного маркетингу. Відомість компанії в Інтернеті через їх вебсайт стала ключовим аспектом в сучасному бізнесі. Професійно розроблений вебсайт не лише привертає увагу потенційних клієнтів, але й створює позитивне враження про компанію та її послуги..

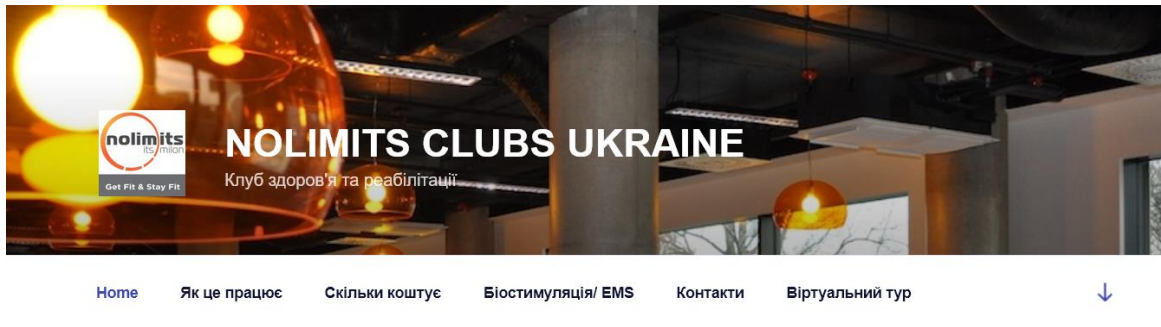
Але незважаючи на це, все ще є компанії, які не приділяють достатньої уваги цьому виду діяльності. Недбало розроблений вебсайт може призвести до втрати клієнтів і негативно вплинути на репутацію компанії.

Тому важливо вдосконалювати процес розробки вебзастосунків та приділяти належну увагу якості кожного етапу, щоб забезпечити успішну роботу вебсайту і задоволення потреб користувачів.

Було проведено аналіз тренажерного залу «Nolimits Clubs», що знаходиться у місті Київ, вебсайт якого показано на рисунку 1.1 [1].

Переваги вебзастосунку тренажерного залу «Nolimits Clubs»:

- є інформація щодо вартості абонементу;
- форма для зворотнього зв'язку;
- є геолокація тренажерного залу.



#### NOLIMITS CLUBS UKRAINE – РОЗУМНИЙ ФІТНЕС ІЗ НІМЕЧЧИНИ

*nolimits clubs – це просто, безпечно, ефективно для  
будь-якого віку та рівня підготовки!*

*nolimits clubs Ukraine пропонує:*

- Необмежену кількість індивідуальних тренувань за фіксованою ціною за місяць у будь-який зручний для Вас час.
- Заняття на комп'ютеризованих тренажерах від провідного світового виробника техніки для спорту та реабілітації *TM milon*:
  - автоматичне налаштування під кожного члена клубу,
  - строго регульована амплітуда для збереження суглобів, захисту від

Рисунок 1.1 – Вебсайт тренажерного залу «Nolimits Clubs»

Недоліки вебзастосунку тренажерного залу «Nolimits Clubs»:

- дизайн дуже різкий та не якісно зроблена верстка;
- немає змоги купити абонемент онлайн;
- немає фото та опису тренажерів;
- не зручний для користування вебсайт.

Проаналізувавши тренажерний зал «Gym4Fit», що знаходиться у ТРЦ «Sunmall» у місті Харків, вебсайт якого показано на рисунку 1.2. Оскільки це один з філіалів мережі «Gym4Fit», складно робити аналіз вебсайту саме цього залу [2].

Переваги вебзастосунку тренажерного залу «Gym4Fit»:

- є опис спортзалу;
- наведені телефонні контакти;
- надано графік роботи спортзалу;
- відгуки на сайті.

## GYM4FIT

Это большой фитнес-центр, который расположился на 5-м и 6-м этажах ТРЦ «Sun Mall»

На обширной территории в 2000 м<sup>2</sup> разместились:

- тренажерный зал
- кардио-зона с множеством беговых дорожек, велотренажеров и орбитреков
- зал для бокса, боксерский ринг, татами для борьбы
- залы групповых занятий (для детей и взрослых)
- массажный кабинет
- салон красоты
- солярий
- фитнес-бар.

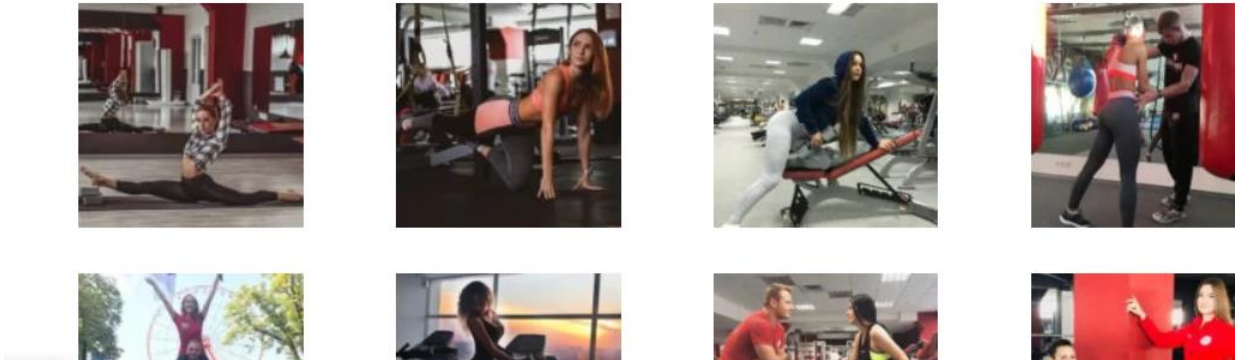


Рисунок 1.2 – Вебсайт тренажерного зала «Gym4Fit»

Недоліки вебзастосунку тренажерного залу «Gym4Fit»:

- дуже не зручний вебсайт з маленьким обсягом необхідної інформації для користувача;
- немає вартості абонементів та не вказано їх типи;
- не зручно читати текст;
- немає фото тренажерів та внутрішнього вигляду спортзалу.

Вебсайт тренажерного залу «Gym4U», зображений на рисунку 1.3, поки що є найкращим варіантом, з проаналізованих вебсайтів. Тренажерний зал знаходиться у місті Київ [3].

Переваги вебзастосунку тренажерного залу «Gym4U»:

- гарно виконаний дизайн та функціонал вебсайту;
- надана вся необхідна інформація для користувача;
- фотографії вигляду тренажерного залу;
- показані усі види абонементів з цінами;
- є точна геолокація.



Рисунок 1.3 – Вебсайт тренажерного залу «Gym4U»

Недоліки вебзастосунку тренажерного залу «Gym4U»:

- неможливо купити абонемент онлайн;
- немає змоги передивитися корисні вправи для тренування.

Як говорилося раніше, майже кожна компанія має свій власний вебсайт для тієї чи іншої мети. але створення такого сайту включає не один чи два, а багато різних і важких етапів.

Етапи створення вебсайту:

- визначення ніші та розписання цілей для постановки задачі;
- аналіз інших вебсайтів у вибранній ніші;
- розробка технічного завдання;
- створення власного дизайну;
- програмування клієнтської та серверної сторони;
- тестування вебсайту;
- закінчення роботи та здача проєкту;
- проведення оновлень за необхідністю.

На початку створення вебсайту, як і всього іншого, необхідно визначитися з темою або нішою для нашого вебсайту, при цьому

проаналізувавши інші вебсайти з такою самою нішою, щоб уникнути проблем з авторськими правами.

Після успішного вибору та аналізу ніші, можна приступати до роботи, а точніше до написання технічного завдання, документа в якому враховані всі тонкощі для подальшої розробки вебсайту, починаючи від дизайну, функціоналу і закінчуючи термінами тестування та здачі вебсайту.

Далі фахівець приступає до розробки дизайну, який має бути приємним для очей та не відштовхувати інших користувачів. Для цього спочатку робиться приблизний макет, далі якщо потрібно, то замовник вносить редагування, і після остаточного узгодження, макет передається до рук фахівця з написання коду, тобто до програміста.

Програміст робить так, щоб отриманий макет став справжнім вебсайтом, використовуючи всі свої навички і необхідні технології та мови програмування. Для клієнтської частини найбільш популярними є технології HTML, CSS та мова програмування Javascript. Для серверної ж потрібен досвід у створенні баз даних, а також однієї з наступних мов програмування, таких як PHP, C#, Python тощо.

Однак, це ще не все, перед здачею сайту замовнику, необхідно провести ретельне тестування, щоб уникнути багів, а також інших системних неполадок. тому вебсайт потрапляє до рук тестувальнику, який і займається цим. Тестувальник перевіряє функціональність, безпеку та продуктивність сайту, щоб гарантувати його відповідність всім вимогам і очікуванням замовника.

Перед здаванням вебсайту, необхідно ознайомити замовника з усім функціоналом і також написати йому короткий документ, в якому будуть вказані всі тонкощі роботи з даним вебзастосунком.

## 1.2 Поточний аналіз стану вебсайтів для спортивних залів у інтернет просторі

Далеко не секрет, що якісний вебзастосунок – це один із ключів до успішного ведення бізнесу, і з індустрією спортивних залів те саме, кожному спортзалу потрібен хороший сайт. Для кращого розуміння теми вебсайти для спортзалів було вивчено літературні та інтернет джерела, пов’язані зі спортзалами та розробкою сайтів.

У інтернет джерелі [4] розповідається про те, для чого спортзалам потрібен неймовірний вебсайт, а також надані поради та приклади вже успішних спорт центрів, які познали висоти та мають якісні сайти.

У джерелі [5] надано 20 прикладів найкращих вебсайтів тренувальних залів з коротким описом, чому і завдяки яким функціям саме вони стали успішними.

У книзі [6] описаний популярний шлях навчання фреймворку React. Це одна з найбільш визнаних книг у світі в області веброботки на базі JavaScript. Її основна мета – надати читачам структурований підхід до вивчення React і розробки вебзастосунків. Книга допомагає підвищити ефективність та продуктивність в процесі навчання та розробки, надаючи зрозумілі пояснення та практичні завдання для закріплення матеріалу. Вона допомагає розробникам уникнути необхідності перебудовувати код з нуля, завдяки якісному та систематизованому підходу до вивчення та використання фреймворку React.

Книга [7] авторства Джеффри Зельдмана є справжнім класиком у світі вебдизайну та розробки. Ця книга визнана як одна з найбільш авторитетних у галузі, яка пропагує використання вебстандартів для створення якісних та доступних вебсайтів. Зельдман пропонує зрозумілі пояснення та ілюстрації, що допомагають читачам краще зрозуміти концепції вебстандартів та їх вплив на процес розробки. Книга стимулює розробників до використання стандартів і допомагає їм уникнути необхідності постійно переробляти код з

нуля, завдяки якісному та систематизованому підходу до вебдизайну та розробки.

У статті [8] йдеться про нову систему спортзалом без персоналу всередині, що допомагає власникам економити великі суми, а також дає клієнтам почуття свободи.

### 1.3 Тенденція спортзалів без персоналу, керованих за допомогою вебпорталу

Сучасне життя вимагає швидкості, зручності та ефективності у всіх сферах. Це стосується і фітнесу. Люди шукають способи тренуватися, які б відповідали їхнім розпорядкам, бюджетам і потребам. Одним із інноваційних рішень стають спортзали без персоналу, керовані за допомогою вебпорталу.

Популярність спортзалів без персоналу, керованих через вебпортали, пояснюється кількома факторами. По-перше, це гнучкість графіка: завдяки вебпорталу клієнти можуть займатися спортом у будь-який час доби, не обмежуючись годинами роботи фітнес-центру або розкладом тренера. Далі, економія часу: відсутність необхідності очікувати чергу на тренажерах чи займати час на реєстрацію та підписання документів робить тренування більш ефективним.

Також, це підвищена конфіденційність: для деяких людей тренуватися без сторонніх очей є більш привабливою опцією. Самостійно в зоні приватності можна більш вільно експериментувати з різними вправами та вагами. Не останнім аспектом є підтримка на відстані: деякі вебпортали надають можливість планувати тренування, вибирати програми та отримувати поради від фітнес-експертів, не виходячи з дому.

Спортзали без персоналу використовують різноманітні технології для підвищення якості тренувань. Оплата також відбувається через вебпортал, що забезпечує зручність та швидкість управління фінансами клієнтів.

Хоча концепція спортзалу без персоналу може мати свої виклики, такі як забезпечення безпеки клієнтів та належне обслуговування обладнання, переваги цього підходу, такі як гнучкість графіка та зручність користування, зазвичай переважають. У світі, де технології швидко розвиваються, спортзали без персоналу, керовані за допомогою вебпорталу, можуть стати новим стандартом у фітнес-індустрії, надаючи клієнтам більше свободи та контролю над своїм фітнес-досвідом [9].

#### 1.4 Постановка задачі

Основним завданням є розробити зручний та багатофункціональний сайт для спортивного залу, який давав би можливість клієнтам дізнатися всю необхідну інформацію про спортзал, а також можливість придбати абонемент та подальші операції з ним онлайн.

Об'єктом роботи є створення вебзастосунку для спортивного залу.

Метою роботи є розробка зручного та багатофункціонального застосунку для автономного спортивного залу.

Головною задачею є створення зручного та багатофункціонального вебзастосунку для спортивного залу, де клієнти зможуть оформлювати абонементи та роботи з ним подальше операції через вебсайт та мати доступ до довідника з корисними вправами для тренування, та адміністраторам буде дана змога моніторингу облікових записів клієнтів, а також перегляд статистики. Буде використовуватися JavaScript, Node.js, React, Rapid API, CSS та MongoDB для розробки системи.

Для досягнення мети необхідно вирішити такі завдання:

- провести аналіз існуючих вебсайтів для спортивних залів;
- розробити технічне завдання;
- створити власний дизайн;
- розробити вебсайт з використанням сучасних технологій.

## 2 МОДЕЛЮВАННЯ СИСТЕМИ ВЕБЗАСТОСУНКУ

### 2.1 Специфікація вимог до застосунку

Застосунок для спортивного залу – це вебсайт, що дозволяє клієнтам створювати особисті облікові записи, виконувати перегляд корисних вправ для тренувань, а також зручно оформляти онлайн-підписки на послуги залу. Оскільки зал є автономним, тобто в ньому немає персоналу, для зручності після кожного оформленого абонементу користувачу буде надаватися секретний пароль, який він може конвертувати у QR-код, котрий в результаті буде працювати як персональний ключ клієнта, та зможе відчиняти двері до спортивного залу.

Система підтримує 3 типи користувачів:

– «Клієнт» – він же зареєстрований користувач з даною роллю вже пройшов реєстрацію спортивного клубу та є його членом. Повинен мати змогу перегляду головну сторінку вебсайту як звичайний користувач, мати доступ до персонального кабінету, сторінки з усіма вправами для тренувань у спортзалі, мати змогу оформлювати онлайн підписку на послуги залу та отримати персональний QR-код, що є ключом-доступом до залу;

– «Адміністратор» – користувач з даною роллю відповідає за безпеку клубу, повинен мати змогу переглядати облікові записи всіх клієнтів, їхню контактну інформацію а також стан підписок, мати змогу видаляти облікові записи клієнтів, що порушують правила клубу, а також перегляди статистику за популярністю онлайн підписок;

– «Незареєстрований користувач» – користувач з цією роллю повинен мати змогу переглядати головну сторінку вебсайту та отримати інформацію щодо спортзалу, також мати доступ до створення облікового запису на платформі для подальших функцій.

## 2.2 Моделювання структури та наповнення вебзастосунку для спортивного залу

Моделювання – це важливий крок у розробці будь-якого вебзастосунку, адже воно дозволяє детально продумати всі можливі сценарії використання системи, а також описати дії та ролі користувачів.

Для вебзастосунку спортивного залу, як і для будь-якого програмного продукту, важливо чітко визначити, хто і що буде робити в системі. Для було використано діаграму прецедентів, яка зображена на рисунку 2.1, яка допоможе візуалізувати взаємодію користувачів з вебзастосунком.

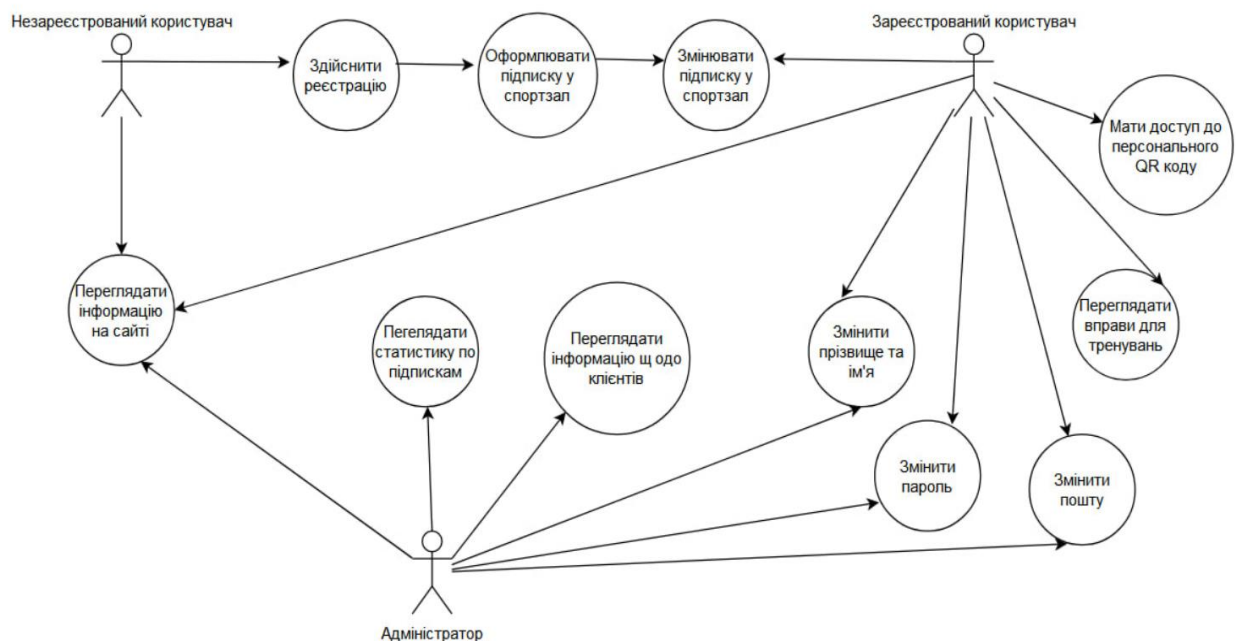


Рисунок 2.1 – Діаграма прецедентів вебзастосунку для спортивного залу

Після створення діаграми прецедентів, наступним кроком у моделюванні вебзастосунку спортивного залу є створення діаграм послідовності. Ці діаграми деталізують кроки, які виконує система у відповідь на дії користувача.

Для кращого розуміння, створюються дві діаграми послідовностей: діаграма послідовності для користувача (рис. 2.2) та діаграма послідовності для адміністратора (рис. 2.3).

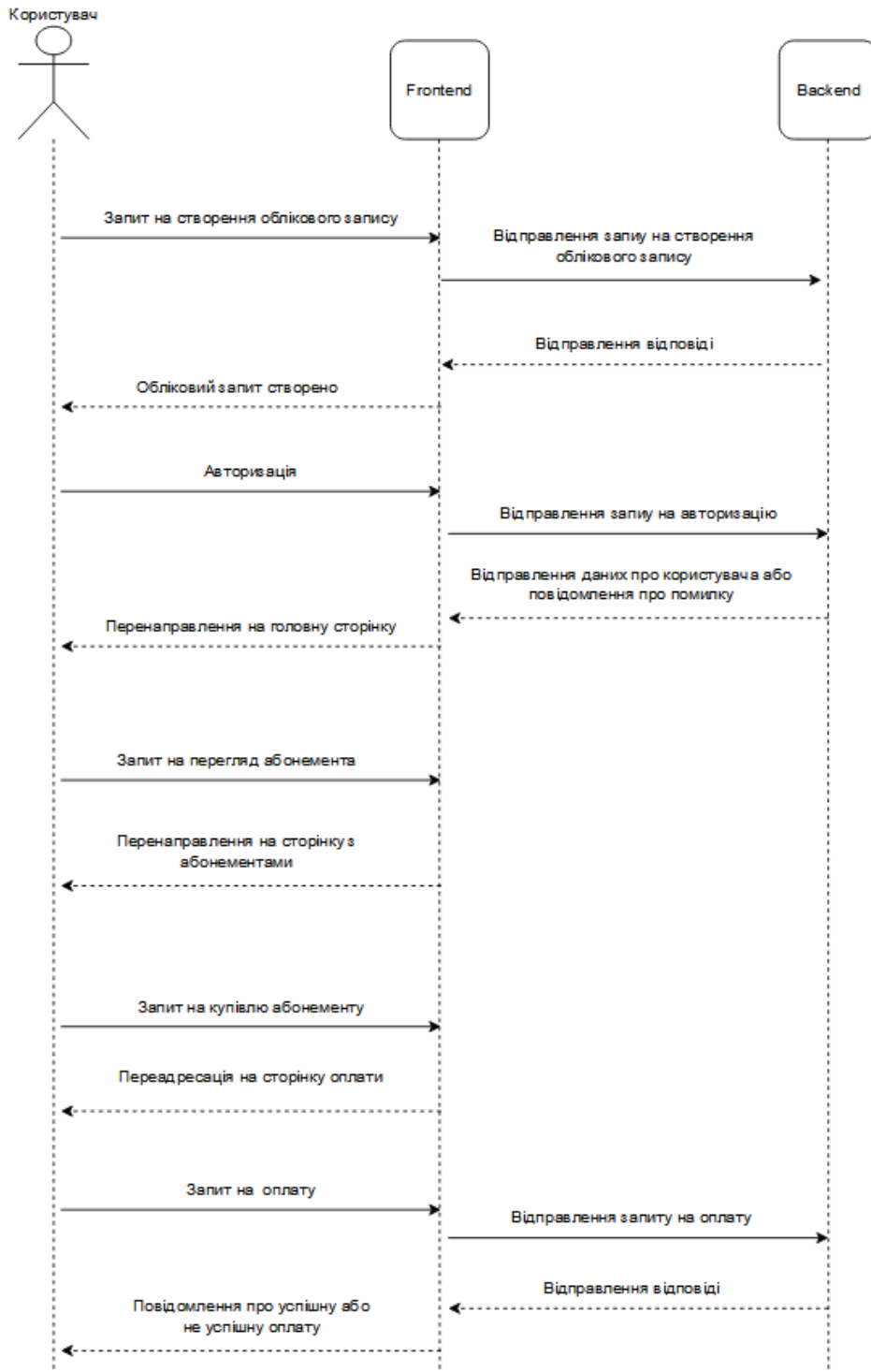


Рисунок 2.2 – Діаграма послідовностей для вебзастосунку для спортивного залу з боку користувача

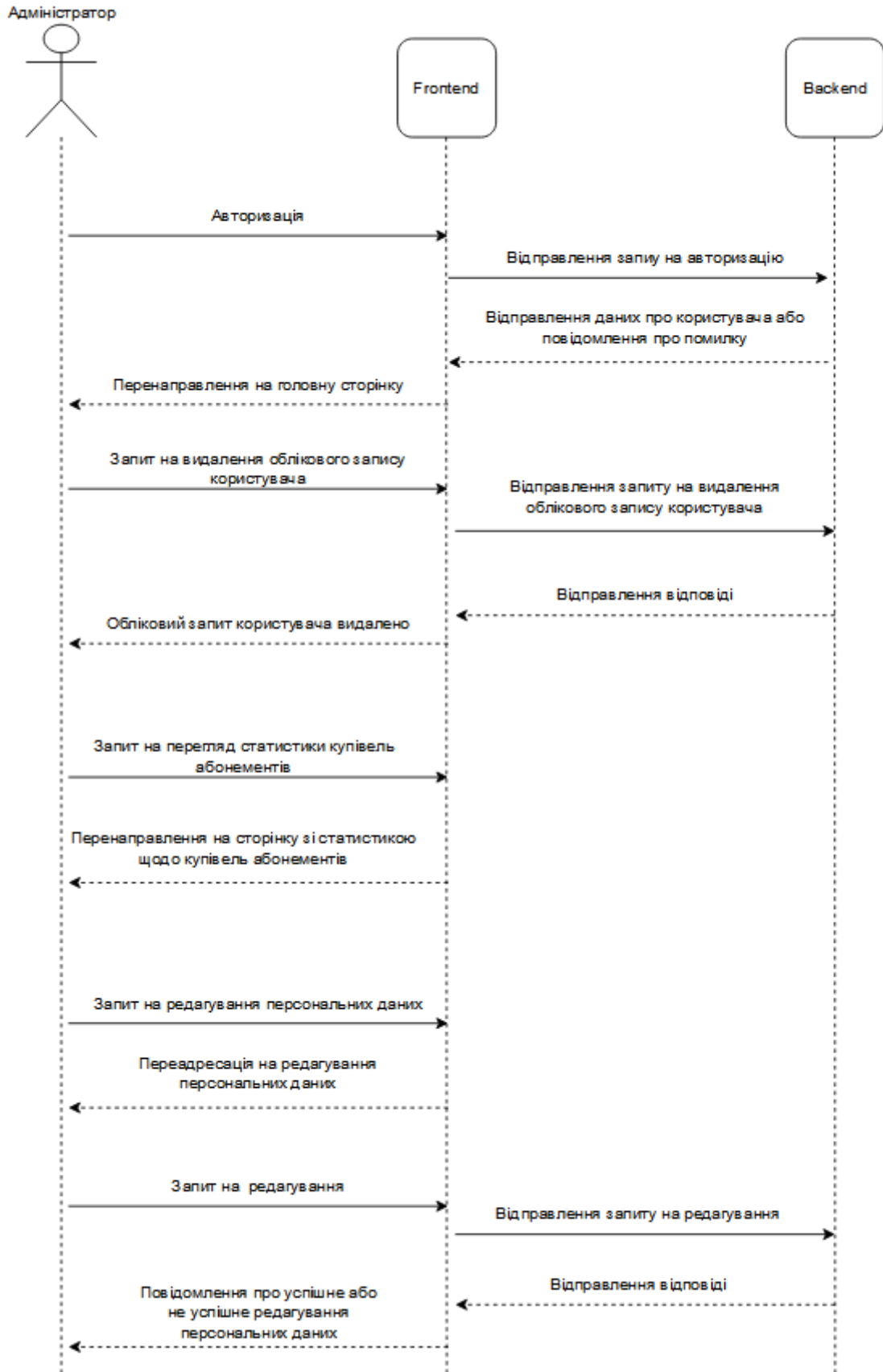


Рисунок 2.3 – Діаграма послідовностей для вебзастосунку для спортивного залу з боку адміністратора

Кожна з діаграм має користувача та дві компоненти сайту – Frontend і Backend. Frontend відображається для клієнта і представляє собою інтерфейс, через який він взаємодіє. Backend прихований від користувача і розміщений на сервері.

На рисунку 2.2 представлено звичайного користувача та його взаємодію з вебзастосунком спортивного залу. Він може оглядати сайт, реєструватися та авторизуватися. Для початку тренувань користувачу необхідно зробити купівлю абонементу до спортзалу, для цього він має перейти на відповідну сторінку, вибрати відповідний до своїх вимог тариф та провести оплату.

На рисунку 2.3 показано адміністратора та його взаємодію з вебзастосунком. Для отримання всіх необхідних функцій потрібно авторизуватися. Після цього адміністратор має доступ до переглядання та видалення облікових записів клієнтів. Адміністратор також має можливість переглядати статистику за придбаними абонементами щоб прийняти правильні рішення щодо покращення та попиту спортивного залу.

### 2.3 Методики Frontend частини

Frontend частина на основі бібліотеки React відповідає за взаємодію з усіма Frontend технологіями, такими як HTML, CSS і JavaScript. Головна роль React полягає у рендерингу компонентів. Під час створення компонентів на React, розробник використовує JavaScript для опису їх зовнішнього вигляду та функціональності. За допомогою React можна створювати різноманітні компоненти, такі як кнопки, форми, списки тощо. Кожен компонент можна розглядати як окремий блок, який можна повторно використовувати в різних частинах сторінки.

Крім того, React взаємодіє з HTML і CSS, які використовуються для створення розмітки та оформлення сторінок. HTML відповідає за структуру

сторінки, а CSS – за її вигляд. Розробник може використовувати React для відображення компонентів на сторінці відповідно до вже створеної розмітки та стилів.

Також React може взаємодіяти з JavaScript, який може бути використаний для додаткової функціональності, наприклад, для зміни стану компонентів або взаємодії з іншими елементами на сторінці.

Узагальнюючи, React взаємодіє з HTML, CSS і JavaScript, щоб створювати динамічні та взаємодіючі інтерфейси користувача. Він забезпечує зручний спосіб створення компонентів та оптимізує рендеринг для швидкої продуктивності сторінки.

У React немає жорсткої вимоги до певної методології або бібліотеки, але в спільноті React часто використовуються різні методології та бібліотеки, такі як MobX, Redux, Flux, Query, React Hooks та UDF [10-16].

MobX – це бібліотека для управління станом даних у JavaScript-застосунках, яка пропонує простий та ефективний спосіб управління складними структурами даних. Основна концепція MobX полягає в тому, щоб створити спостережувані об'єкти, які автоматично оновлюються при зміні їхнього стану.

У MobX стан програми представлений у вигляді дерева об'єктів, які можуть бути спостережуваними. Кожен раз, коли змінюється стан спостережуваного об'єкта, всі компоненти, які підписані на цей стан, автоматично оновлюються. Це забезпечує декларативну модель програмування, де ви просто вказуєте, які частини вашого інтерфейсу повинні відображати яку частину стану, а MobX бере на себе управління оновленням цих компонентів при зміні стану.

MobX також дозволяє використовувати реактивні обчислення, що дозволяє автоматично обчислювати похідні значення на основі вхідних даних. Це дозволяє створювати складні логічні операції, які автоматично оновлюються при зміні вихідних даних.

Однією з головних переваг MobX є його простота використання. Це робить його особливо популярним для швидкої розробки прототипів та невеликих проєктів. Крім того, MobX має активну спільноту користувачів і багато ресурсів для навчання, що робить його досить доступним для вивчення та використання для різних типів проєктів.

Redux – це бібліотека для управління станом даних у JavaScript-застосунках, особливо популярна в засобах розробки вебзастосунків, зокрема у вебфреймворку React. Він створений на базі архітектурного паттерну Flux, проте має деякі особливості, які роблять його більш простим у використанні та розумінні.

У Redux стан програми представлений у вигляді одного об'єкта, який називається Store. Один централізований об'єкт для управління станом дозволяє зберігати всі дані застосунка в одному місці, що робить його структуру більш передбачуваною та легкою для супроводження.

Дії (actions) – це прості об'єкти, які описують зміну стану програми, і вони відправляються до Store для обробки.

Одним з ключових принципів Redux є чистота функцій. Усі зміни стану відбуваються за допомогою чистих функцій, які називаються редукторами. Редуктор – це чиста функція, яка приймає попередній стан та дію і повертає новий стан. Цей підхід робить Redux дуже прозорим та передбачуваним, оскільки він розділяє логіку зміни стану від логіки компонентів.

Redux також має поняття середовища дій (middleware), яке дозволяє розширити функціональність Redux, додавши додаткову логіку для обробки дій перед їхнім потраплянням до Store. Це може бути корисно для реалізації асинхронних дій або для логування.

Одним з головних переваг Redux є його масштабованість. Він дозволяє легко керувати складними структурами даних та великими застосунками, забезпечуючи чистоту та передбачуваність коду. Крім того, велика спільнота користувачів та багато додаткових плагінів та інструментів для розширення

функціональності роблять Redux дуже популярним вибором для управління станом даних в React-застосунках.

Flux – це методологія архітектури для створення користувацьких інтерфейсів у вебзастосунках, розроблена в компанії Facebook з метою поліпшення масштабованості та спрощення налагодження коду у складних вебзастосунках.

Основна ідея Flux полягає в тому, щоб спростити управління станом програми, забезпечуючи суворий односпрямований потік даних. Це означає, що дані рухаються лише в одному напрямку: від вхідних даних, через дії користувача, до змін у стані та відображення на екрані. Такий підхід дозволяє створювати прості та передбачувані застосунки, оскільки вся логіка управління даними зосереджена в одному місці, а кожен елемент застосунку знає тільки про свій стан і не впливає на інші елементи.

У Flux, стан програми представлений у вигляді одного об'єкта, що називається Store. Кожен компонент програми отримує доступ тільки до свого стану, який передається через властивості (props). Коли користувач взаємодіє з елементами застосунку, це викликає дію (Action), яка передає дані в Dispatcher. Dispatcher надсилає дію всім зареєстрованим Store, які оновлюють свій стан і сповіщають усі компоненти, що використовують їхній стан, про необхідність оновлення.

Отже, Flux забезпечує чітку структуру застосунка та спрощує його супровід і розширення. Однак, реалізація Flux може вимагати більше коду, ніж простіші методології, і може бути складною для розуміння початківцями-розробниками. Наразі існує безліч бібліотек і фреймворків, які надають реалізацію Flux для JavaScript, такі як Redux, Alt.js і Reflux.

Query – це бібліотека, яка дозволяє легко та ефективно керувати станом даних та взаємодіяти з сервером у вебзастосунках, особливо в застосунках, побудованих на фреймворку React. Однією з ключових особливостей Query є його фокус на декларативному підході до опису потреб у даних та операцій з ними.

За допомогою Query можна легко описати, які дані потрібно завантажити з сервера та яким чином ці дані повинні бути використані у вашому інтерфейсі. Бібліотека автоматично керує завантаженням та кешуванням даних, оптимізуючи продуктивність вашого застосунка.

Однією з важливих концепцій у Query є «запити» (queries) та «мутації» (mutations). Запити використовуються для отримання даних з сервера, тоді як мутації використовуються для внесення змін у ці дані. Це дає можливість створювати потужні та гнучкі застосунки, які можуть ефективно взаємодіяти з сервером та реагувати на зміни в даних в реальному часі.

Крім цього, Query надає ряд інших корисних функцій, таких як підтримка кешування даних, автоматичне оновлення інтерфейсу при зміні даних та можливість використання спеціальних хуків для зручного підключення даних до компонентів React.

React Hooks – це інноваційний підхід у React, який дозволяє використовувати функціональні компоненти з усіма можливостями стану та іншими особливостями React без необхідності створювати класові компоненти. Цей підхід пропонує більш простий та елегантний спосіб керування станом компонентів.

Основна ідея React Hooks полягає в тому, щоб винести логіку стану з компонентів та створити її окремо, використовуючи спеціальні функції-хуки. Ці хуки можна викликати всередині функціональних компонентів для отримання доступу до стану, ефектів, контексту тощо.

За допомогою React Hooks можна працювати зі станом у функціональних компонентах, а також використовувати життєвий цикл компонентів, створювати власні хуки для повторного використання логіки і багато іншого. У React існують кілька вбудованих хуків, таких як `useState`, `useEffect`, `useContext`, `useReducer` і т. д., які надають доступ до різних функціональностей React і можуть бути використані для вирішення різних завдань, таких як обробка подій, взаємодія з сервером та керування станом компонента.

React Hooks дозволяє значно скоротити кількість коду, спростити його читання та зробити його більш зрозумілим. Крім того, він підвищує продуктивність програми, оскільки уникати створення зайвих екземплярів класових компонентів і спрощує процес оновлення компонентів.

Методологія UDF (Unidirectional Data Flow) – це підхід до розробки програмного забезпечення, в основі якого лежить організація потоку даних в одному напрямку. У цьому підході дані передаються від верхнього компонента до нижнього, а не в обидва напрямки, як у традиційному підході MVC (Model-View-Controller). Це означає, що вся взаємодія між компонентами здійснюється через один централізований об'єкт, який зазвичай називається Store.

У методології UDF використовується певна структура для управління станом даних, яка називається Store. Цей Store є центральним елементом, через який проходять всі дані та дії в застосунку. Компоненти відправляють дії (actions) до Store, який обробляє їх та оновлює свій стан. Після цього він повідомляє всі підписані компоненти (views) про зміни, щоб вони могли відображати актуальний стан даних на екрані.

Одним з ключових принципів UDF є принцип інверсії залежностей, що означає, що компоненти не повинні напряму залежати один від одного. Замість цього, вони повинні взаємодіяти через інтерфейси та події, що дозволяє зробити компоненти більш масштабованими та полегшує тестування коду.

Отже кожна з цих методологій має свої переваги та недоліки і може бути застосована залежно від специфічних вимог та потреб проєкту [17].

## 2.4 Методики Backend частини

На основі платформи Node.js, яка дозволяє виконувати JavaScript на сервері, можна побудувати повноцінну back-end частину вебзастосунку, не

вдаючись до інших мов програмування. Node.js використовує той же рушій V8 JavaScript, що і браузер Google Chrome, що забезпечує його швидкість та ефективність.

За допомогою Node.js можна розробляти вебсервери, API, мікросервіси, інструменти командного рядка та багато іншого. Крім того, Node.js дозволяє взаємодіяти з різними базами даних, такими як MongoDB, MySQL, PostgreSQL та іншими. Він також надає розширюваність і гнучкість, дозволяючи розробникам створювати власні модулі для конкретних завдань.

Завдяки високій продуктивності, масштабованості і гнучкості, Node.js став популярним вибором для створення серверних частин вебзастосунків. Велика спільнота розробників підтримує безліч пакетів і бібліотек, що спрощують розробку на Node.js.

Також Node.js має велику популярність, оскільки з ним можна використовувати велику кількість різних архітектурних стилів, патернів, та принципів програмування. Найпопулярнішими з них є REST, MVC та SOLID.

REST – це архітектурний стиль для розробки API (Інтерфейс прикладного програмування). Цей підхід сприяє спрощенню інтеграції між різними системами та забезпечує високу гнучкість при розробці веб-додатків. Він визначає набір принципів та обмежень для створення вебсервісів, які є легкими для використання та масштабованими. REST API використовують HTTP методи (GET, POST, PUT, DELETE) для взаємодії з ресурсами.

REST API працює з ресурсами, які можуть бути будь-якими даними, такими як користувачі, продукти або замовлення. Кожен запит до REST API повинен містити всю необхідну інформацію для його виконання. Сервер не повинен зберігати стан між запитами. Клієнт надсилає запити до сервера, а сервер повертає відповіді. Клієнт не повинен знати внутрішню реалізацію сервера та повинен використовувати HTTP методи для створення, читання, оновлення та видалення ресурсів. REST API повинні використовувати загальний інтерфейс на основі HTTP для взаємодії з різними ресурсами.

Щодо переваг можна зазначити, що REST API прості у використанні та розумінні, легко масштабуються для обробки великої кількості запитів та можуть використовуватися клієнтами з різних платформ.

MVC – це архітектурний патерн для розробки програмного забезпечення, який часто використовується у вебзастосунках. Він розділяє застосунок на три основні компоненти:

- модель (Model): модель представляє дані та бізнес-логіку програми. Вона відповідає за зберігання, отримання та оновлення даних;

- вигляд (View): вигляд відповідає за візуальне представлення даних користувачеві. Він містить HTML, CSS та JavaScript для відображення даних та реагування на дії користувача;

- контролер (Controller): контролер обробляє дії користувача та взаємодіє з моделлю та видом. Він отримує запити від користувача, оновлює модель за необхідності та повідомляє view про зміни даних.

Переваги MVC включають розділення інтересів, що покращує організацію коду, збільшення повторного використання коду, оскільки компоненти можуть бути використані в різних частинах програми, спрощене тестування, оскільки кожен компонент можна тестувати окремо, а також наявність широкої підтримки та ресурсів для використання даного патерну.

SOLID – це аббревіатура, яка представляє собою набір принципів об'єктно-орієнтованого програмування, які спрямовані на створення гнучких, розширюваних і легко супроводжуваних систем. Кожна літера в аббревіатурі SOLID представляє один із цих принципів:

- принцип єдиної відповідальності (Single Responsibility Principle): кожен клас або модуль повинен мати лише одну причину для зміни. Це означає, що кожен компонент програми повинен виконувати лише одну конкретну функцію або завдання;

- принцип відкритості/закритості (Open/Closed Principle): система повинна бути відкритою для розширення, але закритою для модифікацій. Це

означає, що класи або модулі повинні бути легко розширюваними без зміни їхнього внутрішнього коду;

– принцип підстановки Барбари Лісков (Liskov Substitution Principle): об'єкти базового класу повинні бути замінними своїми підтипами, тобто, код, який використовує базовий клас, повинен працювати коректно з будь-яким підтипом цього класу;

– принцип розділення інтерфейсу (Interface Segregation Principle): клієнти не повинні залежати від інтерфейсів, які вони не використовують. Це означає, що класи не повинні примушуватися реалізувати методи, які вони не використовують;

– принцип інверсії залежностей (Dependency Inversion Principle): високорівневі модулі не повинні залежати від деталей низькорівневих модулів. Обидва типи модулів повинні залежати від абстракцій. Крім того, абстракції не повинні залежати від деталей, а деталі повинні залежати від абстракцій. Цей принцип спрощує зміну та розширення системи, оскільки він дозволяє замінювати деталі без зміни високорівневого коду.

Кожен з цих принципів є теоретичним принципом, який сприяє створенню коду високої якості, що легко супроводжується та модифікується [18-24].

## 2.5 Проєктування бази даних застосунку

База даних – це організована колекція даних, яка зберігається та управляється системою у такий спосіб, що її можна легко доступати, оновлювати та управляти. Бази даних використовуються для зберігання інформації в структурованому форматі, яка може бути подана у вигляді таблиць, рядків та стовпців. Вони використовуються в різних галузях, включаючи бізнес, освіту, медицину, науку тощо.

Бази даних можна класифікувати за різними критеріями, включаючи тип даних, які вони зберігають (реляційні, NoSQL, графові тощо), структуру даних (плоскі, ієрархічні, мережеві), методи доступу до даних (SQL, NoSQL), та способи організації (централізовані, розподілені).

Реляційні бази даних є одним з найпоширеніших типів і використовуються для зберігання даних у вигляді таблиць, де кожен рядок представляє запис, а кожний стовпчик – атрибут. Таблиці можуть бути пов'язані між собою за допомогою ключів. NoSQL бази даних, натомість, забезпечують гнучкіший підхід до зберігання даних, дозволяючи зберігати структури, що можуть варіюватися, та обробляти великі об'єми даних.

Структура бази даних є ключовим елементом, який визначає, наскільки ефективно та надійно можна зберігати та обробляти дані. Правильно спроектована структура дозволяє легко взаємодіяти з даними, забезпечує їх цілісність та безпеку. Крім того, вона дозволяє ефективно керувати зв'язками між даними та виконувати запити до бази даних.

Оскільки бази даних відіграють ключову роль у багатьох системах, важливо розуміти потреби користувачів та забезпечувати відповідну структуру для забезпечення їх надійності та ефективності. Також необхідно регулярно робити резервні копії баз даних для захисту від втрати даних та забезпечення захисту від несанкціонованого доступу та атак.

Вибір бази даних є важливим етапом при розробці будь-якого вебсайту. Підбираючи базу даних, розробники зазвичай враховують такі критерії, як потужність, швидкість, масштабованість, гнучкість та вартість.

У випадку створення вебсайту, обрана база даних MongoDB може мати кілька обґрунтованих причин. MongoDB є документно-орієнтованою базою даних, яка забезпечує гнучкість та простоту в роботі з даними. Вона дозволяє зберігати дані у вигляді документів у форматі JSON, що зручно для розробки вебзастосунків, оскільки відповідає структурі даних, що використовується у JavaScript.

Крім того, MongoDB добре підходить для сучасних вебсайтів, які потребують гнучкості та масштабованості. Вона легко масштабується горизонтально, дозволяючи додавати нові сервери для обробки збільшеного навантаження без значного перетворення архітектури застосунку. Це особливо важливо для вебсайтів, які очікують зростання обсягу даних та користувачів з часом.

Також MongoDB має активну спільноту та широкий екосистему інструментів, що полегшує розробку та підтримку вебсайтів. І, звичайно ж, вибір бази даних залежить від конкретних вимог та особливостей проєкту, але MongoDB може бути привабливим варіантом для багатьох вебсайтів завдяки своїм перевагам у гнучкості, масштабованості та зручності в роботі з даними.

База даних міститиме кілька моделей, таких як користувач (User), підписка (Subscription), оформлення підписки (Purchase) і QR-код (QR\_Code), а також відповідні зв'язки між ними. На рисунку 2.4 показана приблизна структура цієї бази даних.

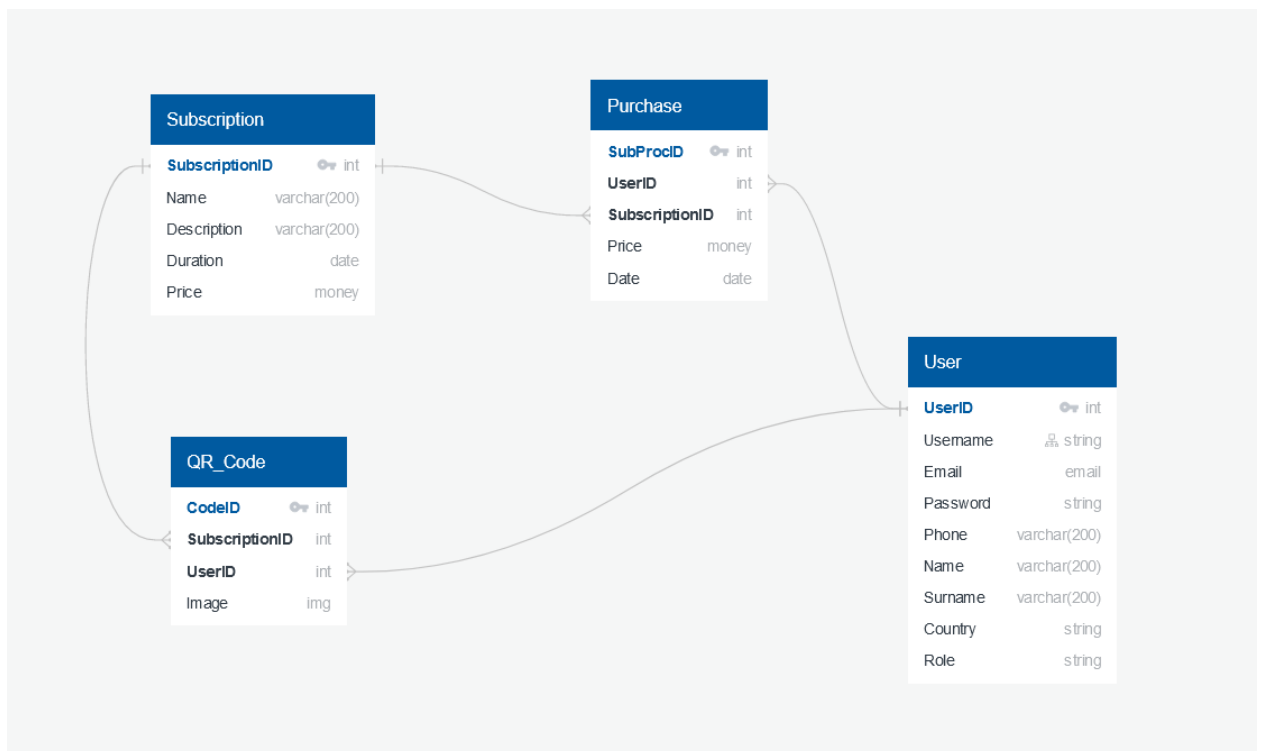


Рисунок 2.4 – Структура бази даних

Модель User включає поля, необхідні для користувача для автентифікації, такі як юзернейм, ім'я, прізвище, адреса електронної пошти, пароль, номер телефону, країна та роль.

Модель Subscription містить інформацію про підписку, тобто абонемент, та має наступні поля: назва, опис, ціна та тривалість. Модель також містить зв'язки з покупкою та QR-кодом.

Модель QR-code містить інформацію про персональний ключ користувача, який і надає змогу користуватися спортивним залом. Має такі поля, як код, зображення, а також зв'язки з підписками та клієнтами через моделі User\_QR і Subs\_QR.

Модель Purchase відстежує інформацію про купівлю, тобто вже сам процес оформлення підписки та мають поля: ціна, дата, а також має зв'язки з користувачами та підписками.

MongoDB буде використовуватися як система управління базами даних, оскільки вона надає безліч можливостей для зберігання та маніпулювання даними в гнучкому форматі JSON. Така структура дозволяє легко розширювати та модифікувати базу даних у майбутньому, наприклад, додавати нові поля до документів чи колекцій, не порушуючи існуючої структури. Кожен документ містить набір ключ-значення, які відображають характеристики об'єктів, що зберігаються в цій колекції.

Завдяки такому підходу, база даних стає потужним інструментом для роботи з даними, що дозволяє зберігати, оновлювати та отримувати необхідну інформацію відповідно до потреб користувачів системи. Крім того, база даних може бути легко масштабована, що дозволяє адаптувати її до зростаючих вимог бізнесу та збільшення обсягів даних.

Загалом, структура цієї бази даних представляє собою добре організовану модель, яка забезпечує ефективне зберігання і доступ до даних, необхідних для роботи інформаційної системи [25].

### 3 РОЗРОБКА ВЕБЗАСТОСУНКУ ДЛЯ СПОРТИВНОГО ЗАЛУ

#### 3.1 Вибір інструментальних засобів для реалізації поставленої задачі

У процесі дослідження засобів розробки в обраній предметній області для створення вебзастосунку для спортивного залу було виявлено велике різноманіття інструментів. Розвиток програмного забезпечення для спортивних закладів стимулює розвиток як середовищ розробки, так і додаткових інструментів, створених як спеціалістами-ентузіастами, так і великими компаніями [26].

Зокрема, для створення даного вебзастосунку були обрані такі інструменти:

- Draw.io: для створення діаграм та схем архітектури вебзастосунку, що допомагає візуалізувати структуру проєкту та спрощує роботу з командою. З його допомогою можна швидко і зручно створювати різноманітні діаграми, такі як блок-схеми, UML-діаграми, потокові схеми, архітектурні плани тощо, що дозволяє чітко представляти структуру проєкту, взаємозв'язки між компонентами та логіку його функціонування. Використання Draw.io спрощує комунікацію в команді, оскільки всі учасники можуть однаково розуміти інформацію, представлену у вигляді діаграм, що сприяє кращому розумінню та спільному прийняттю рішень. Крім того, Draw.io надає можливість зберігати та ділитися діаграмами в різних форматах, що сприяє зручній співпраці в команді та забезпечує збереження даних для майбутнього використання;

- Microsoft Visual Studio Code: як основне середовище розробки для зручного написання коду і керування проєктом. Це потужний, безкоштовний текстовий редактор, розроблений Microsoft. Він широко використовується серед програмістів та розробників для написання коду різноманітних мов програмування, включаючи JavaScript, HTML, CSS, Python, Java, C++, і багато інших. VS Code має інтуїтивний і зручний інтерфейс, що дозволяє

швидко орієнтуватися навіть новачкам. За допомогою розширень (extensions), розробники можуть налаштувати редактор під свої потреби, додавши функціональність для підтримки конкретних мов програмування, фреймворків, інструментів, та інших функцій. Вбудована підтримка Git дозволяє взаємодіяти з репозиторіями, виконувати коміти, відкати зміни, переглядати історію, та інше. Включає в себе розумне автодоповнення коду, підсвічування синтаксису, рефакторинг, швидкі горячі клавіші та багато іншого, що полегшує написання та редагування коду. В цілому, Microsoft Visual Studio Code надає розробникам потужний і зручний інструмент для написання, редагування та керування кодом у різних мовах програмування, що робить його ідеальним вибором для основного середовища розробки для багатьох проєктів;

– MongoDB Compass для MongoDB: для управління базою даних MongoDB з інтуїтивно зрозумілим інтерфейсом. Це офіційний візуальний інструмент для адміністрування баз даних MongoDB, який призначений для спрощення управління базою даних MongoDB, надаючи інтуїтивно зрозумілий інтерфейс, який дозволяє адміністраторам та розробникам легко взаємодіяти з даними. MongoDB Compass дозволяє візуалізувати структуру даних, виконувати запити, аналізувати дані, створювати та керувати індексами, а також проводити адміністративні дії, такі як резервне копіювання та відновлення бази даних. Він має інтуїтивний і зручний інтерфейс, що робить роботу з базою даних MongoDB більш простою та ефективною;

– Node.js: в якості середовища виконання JavaScript для серверної частини застосунку, що забезпечує швидку та масштабовану обробку запитів. Одним із головних переваг Node.js є його здатність забезпечити швидку та ефективну обробку запитів, що робить його ідеальним вибором для створення вебзастосунків, які потребують великої продуктивності та масштабованості. Node.js використовує асинхронний та подієвий підхід до програмування, що дозволяє ефективно обробляти багатопоточні операції і

реагувати на події в реальному часі без блокування потоку виконання. Це робить його ідеальним вибором для розробки вебзастосунків, які вимагають високої відповідності та масштабованості. Завдяки великому екосистемі модулів та бібліотек, Node.js надає розробникам широкі можливості для створення різноманітних вебзастосунків, включаючи вебсервери, API, чат-застосунки, стрімінгові сервіси та багато іншого. Він також підтримується активною спільнотою розробників та постійно розвивається, що робить його одним з найпопулярніших та потужних інструментів для розробки вебзастосунків у сучасному програмуванні;

– React: для розробки користувацького інтерфейсу з високою швидкодією та можливістю повторного використання компонентів. Однією з головних переваг React є його висока швидкодія та ефективність, що робить його ідеальним вибором для створення динамічних та інтерактивних інтерфейсів. React базується на концепції компонентного підходу, де кожен елемент інтерфейсу представлений як окремий компонент. Це дозволяє розробникам легко організувати та керувати великими застосунками, розбивши їх на невеликі незалежні компоненти, які можна повторно використовувати у різних частинах застосунку або навіть в інших проєктах. Такий підхід спрощує розробку, тестування та підтримку коду, а також підвищує його читабельність та масштабованість;

– HTML та CSS: для створення структури та оформлення вебсторінок. це основні мови веброзробки, які використовуються для створення структури та оформлення вебсторінок. HTML відповідає за структуру та контент вебсторінки, визначаючи розміщення різних елементів, таких як текст, зображення, таблиці, форми та інші. CSS відповідає за стиль та вигляд вебсторінки, надаючи можливість встановлювати колір, шрифт, розмір, межі, відступи та інші властивості для кожного елемента на сторінці. Використання HTML та CSS дозволяє розробникам створювати естетично привабливі та добре структуровані вебсторінки, що забезпечує кращий користувацький досвід та покращує зручність використання вебзастосунків.

Завдяки своїй простоті та доступності, HTML та CSS є основою для будь-якого вебзастосунку чи сайту, незалежно від його складності та функціональності;

– MobX: для управління станом застосунку і забезпечення реактивності вебінтерфейсу. Основним принципом MobX є спрощення управління станом, де стан програми представлений як один об'єкт, який автоматично оновлюється при зміні відповідних даних. Це дозволяє розробникам просто визначати, які частини інтерфейсу мають бути оновлені при зміні стану, без необхідності вручним чином втручатися в цей процес;

– RapidAPI: для інтеграції з зовнішніми сервісами та отримання доступу до спортивних даних або інших корисних ресурсів через API. це платформа для інтеграції з різноманітними зовнішніми сервісами та отримання доступу до їх функціональності через API (інтерфейс програмування застосунків). За допомогою RapidAPI, розробники можуть легко з'єднувати свої застосунки з широким спектром сервісів, таких як соціальні мережі, фінансові інструменти, ігрові платформи, геолокаційні сервіси, спортивні дані та багато іншого. RapidAPI надає простий та зручний інтерфейс для пошуку, вибору та інтеграції з різноманітними сервісами, а також надає інструменти для керування та моніторингу використання API, що дозволяє розробникам ефективно використовувати зовнішні сервіси у своїх застосунках.

### 3.2 Етапи розробки вебзастосунку для спортивного залу

Розробка вебзастосунку для спортзалу – це складний та тривалий процес, який потребує поділу на етапи. Першим кроком є аналіз літературних джерел з веброзробки та спортивного бізнесу, щоб використовувати сучасні технології та інструменти, які продовжують активно розвиватися [27-29].

Після аналізу починається етап проєктування системи, де вирішуються питання взаємодії користувача з застосунком. Далі, розробляється дизайн вебсайту, з фокусом на сучасному та зручному інтерфейсі, що відповідає потребам користувачів спортзалу.

Після проєктування переходимо до написання коду, де використовується React, щоб забезпечити ефективну та швидку роботу вебзастосунку. Крім того, використовується база даних для зберігання інформації про користувачів та інші дані, що є важливими для функціонування спортивного закладу.

Для більшої безпеки користувачів, всі паролі у базі даних є зашифрованными (рис. 3.1). Це стандартний підхід у сфері інформаційної безпеки, який дозволяє уникнути можливості несанкціонованого доступу до конфіденційної інформації навіть у випадку компрометації бази даних. Такий захист забезпечує високий рівень конфіденційності та захищає дані користувачів від потенційних загроз.

```
email : "illia.spasonov@nure.ua"  
password : "$2b$04$I3byNZzFjRmvZYxpnEM0uvCOZUakOnYABt2e19Gtj/srqdi.Qkdq"  
name : "Illia Spasonov"
```

Рисунок 3.1 – Зашифрований пароль користувача у базі даних

У базі даних знаходиться чотири таблиці: user, subscription, qrcode, purchase (рис. 3.2). Для забезпечення безпеки вирішено мати лише одного адміністратора, що є одним з головних кроків у забезпеченні безпеки та ефективного управління системою. Зміна ролі адміністратора безпосередньо в базі даних є простим та надійним способом забезпечення адміністративного контролю над системою, що дозволяє забезпечити мінімальну кількість привілеїв адміністраторам, зменшуючи ризик недозволених дій та забезпечуючи безпеку даних користувачів.

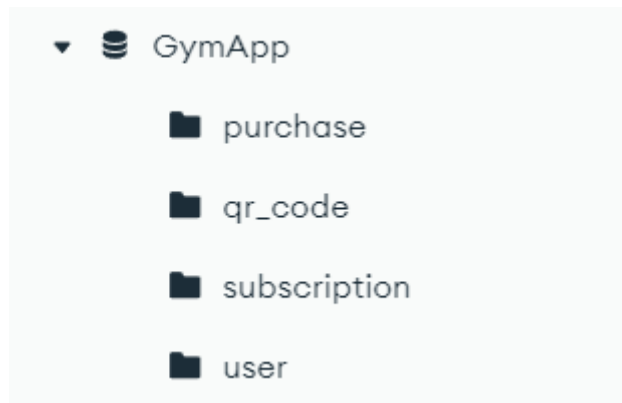


Рисунок 3.2 – База даних вебзастосунку для спортивного залу

### 3.3 Тестування розробленого вебзастосунку та аналіз результатів

Під час фінального етапу розробки перед передачею готового проєкта замовнику необхідно провести повне тестування застосунку, бо це є критично важливим етапом. Для того, щоб нічого не пропустити, найкраще провести тестування в три етапи:

- вебсайт перевірявся в процесі створення всіх функціональних можливостей. Важливо було протестувати кожну функцію на етапі розробки;
- кожна сторінка була ретельно перевірена на відповідність дизайну. Важливість дизайну не можна недооцінювати, адже він впливає на емоційне сприйняття користувачем і його бажання обрати саме цей готель;
- після завершення розробки всіх сторінок була проведена остаточна перевірка функціональності вебсайту. Цей етап дозволив упевнитися, що всі можливості працюють згідно з технічним завданням.

Ці три етапи тестування грають критичну роль у забезпеченні якості та надійності вебзастосунку перед його передачею замовнику, оскільки тестування як процес передбачає детальну перевірку якості програмного забезпечення згідно з його документацією. Перевірка функціональності на ранніх етапах розробки дозволяє вчасно виявити та виправити будь-які недоліки, що забезпечує ефективну роботу всіх функцій під час фінального

тестування. Ретельна перевірка відповідності дизайну додає важливий елемент естетичності та комфорту взаємодії з користувачем. Нарешті, остаточна перевірка функціональності перед виходом в продакшн є ключовою для впевненості в тому, що вебзастосунок готовий до роботи в реальних умовах і здатний забезпечити найвищий рівень задоволення користувачів. Такий систематичний підхід до тестування дозволяє забезпечити високу якість та успішний запуск проєкту.

Основна сторінка вебзастосунку, зображена на рисунку 3.3, виглядає однаково для всіх користувачів, незалежно від того, чи зареєстровані вони або авторизовані. На цій сторінці представлений основний функціонал та інформація, доступні для перегляду всім відвідувачам. Для отримання доступу до додаткових функцій та персоналізованого контенту, користувачі повинні пройти процедуру реєстрації або авторизації. Таким чином, ця сторінка служить універсальним вступом до сервісу, забезпечуючи зручний доступ до основних можливостей вебзастосунку без необхідності входу в систему.

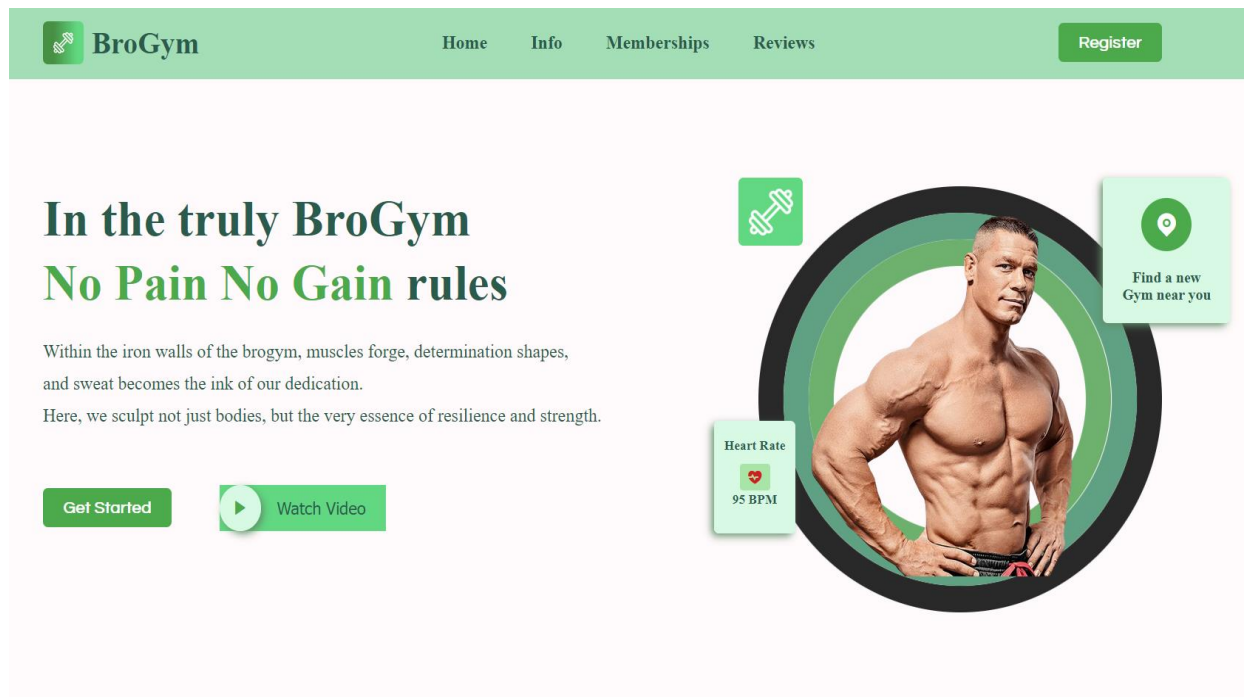


Рисунок 3.3 – Основна сторінка застосунку для всіх видів користувачів

На цій сторінці користувачі можуть переглянути всю важливу інформацію про спортзал, а саме:

- знайти інформацію про те, чому спорт та вправи мають переваги для здоров'я людини (рис. 3.4);
- мотиваційний напис та зображення, які дозволять підняти бойовий настрій (рис. 3.5);
- за допомогою онлайн калькулятора підрахувати власний індекс маси тіла, який повідомить про справжній стан ваги користувача (рис. 3.6);
- переглянути всі доступні абонементи у спортивний зал (рис. 3.7);
- дізнатися контактний адрес та номер телефону (рис. 3.8);
- переглянути відгуки клієнтів спортивного залу (рис. 3.9).



Рисунок 3.4 – Переваги вправ

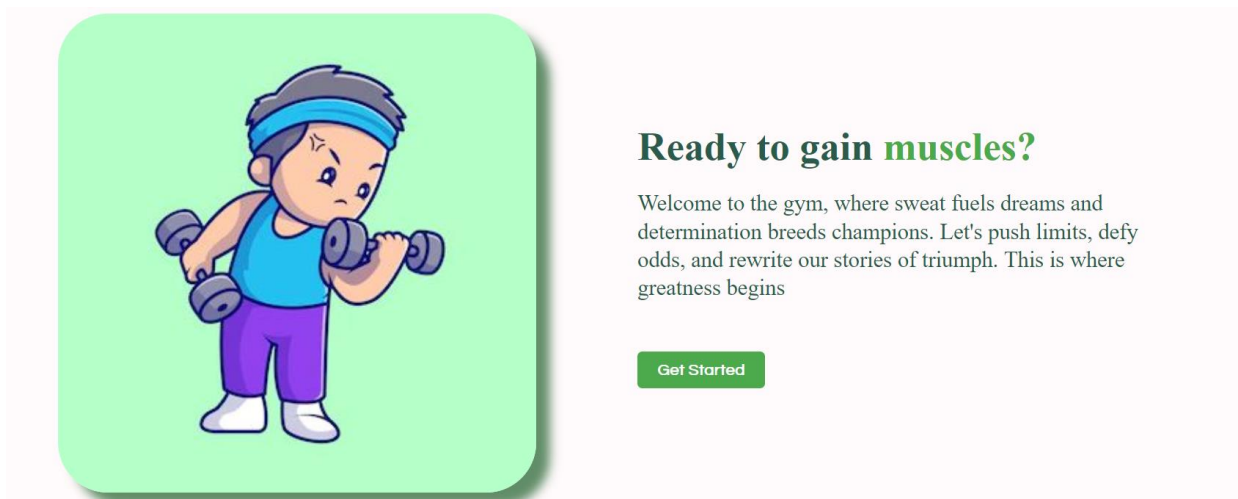


Рисунок 3.5 – Мотиваційний напис

## Calculate Your BMI

The body mass index (BMI) calculator calculates body mass index from your weight and height.

Рисунок 3.6 – Калькулятор ІМТ

## Gym Pricing Plan

Pleasant and very profitable offers at the lowest prices

### 1 Month Membership

\$50 /month

- Unlimited access to the gym
- Customer support
- 1 month access
- Standart option
- 3 classes per week

### 1 Year Membership

\$25 /month

- Unlimited access to the gym
- Customer support
- 12 month access
- Best option
- Unlimited classes per week

### 3 Months Membership

\$40 /month

- Unlimited access to the gym
- Customer support
- 3 month access
- Premium option
- 5 classes per week

Рисунок 3.7 – Список абонементів у спортивний зал

## BroGym

In the BroGym, sweat writes the story of resilience, where iron forges muscles and determination shapes destinies

### Company

- Our program
- Our plan
- Become a member

### Quick links

- Info
- Memberships
- Reviews

### Contact us

Kharkiv, Ukraine  
Nauki Avenue, 14  
+380971234567

Copyright - 2024 developed by Illia Spasonov. All rights reserved.

Рисунок 3.8 – Контактний адрес та номер телефону спортивного залу

## Reviews



**Illia Spasonov**

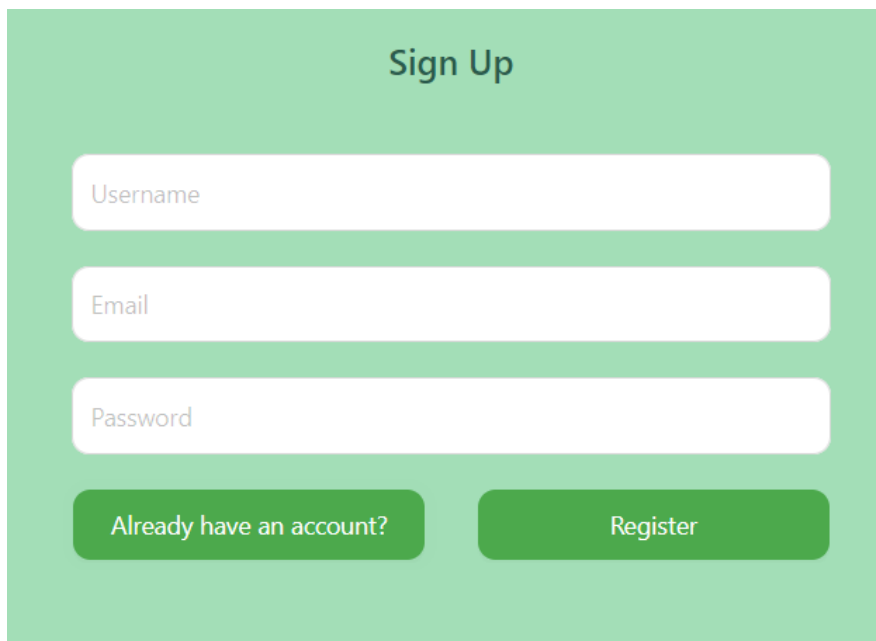
**Rating:** ★★★★★

Brogym is a hub of motivation and achievement! From the moment you walk in, the air is charged with determination. The trainers are highly knowledgeable, the equipment is first-rate, and the community is extremely supportive.

Рисунок 3.9 – Відгуки клієнтів

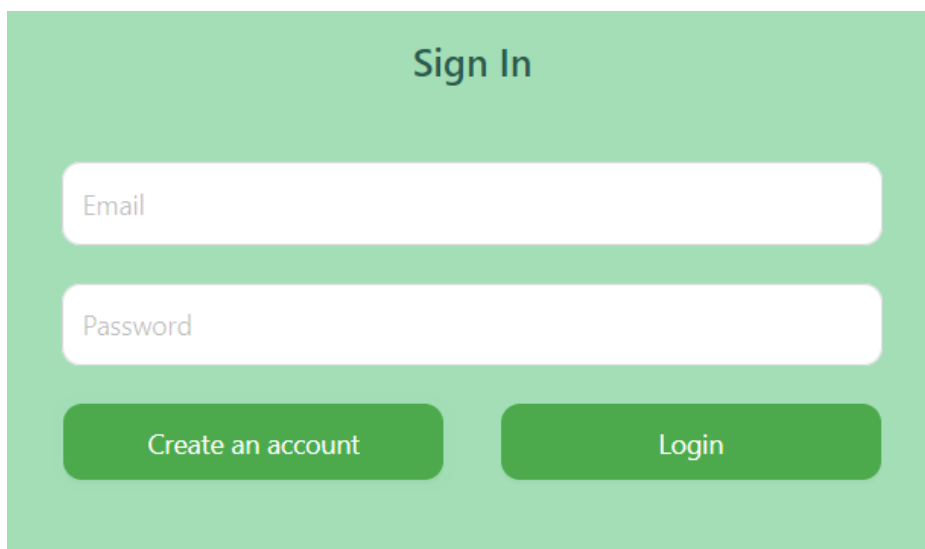
Після успішного зовнішнього огляду вебсайту користувач може створити новий обліковий запис (рис. 3.10), або зайти вже в наявний для доступу до подальших функцій застосунку (рис 3.11). Реєстрація дозволяє користувачу створити профіль, зберігати налаштування та отримувати доступ до ексклюзивного контенту. Авторизація в уже наявний обліковий

запис відкриває можливість використовувати збережені дані для подальших функцій при роботі з вебсайтом. Таким чином, процес реєстрації та авторизації є ключовим для розширення функціональних можливостей та покращення користувацького досвіду на вебзастосунку.



The image shows a 'Sign Up' form with a light green background. At the top, the text 'Sign Up' is centered. Below it are three white input fields with rounded corners, labeled 'Username', 'Email', and 'Password' from top to bottom. At the bottom of the form, there are two green buttons with white text: 'Already have an account?' on the left and 'Register' on the right.

Рисунок 3.10 – Форма реєстрації



The image shows a 'Sign In' form with a light green background. At the top, the text 'Sign In' is centered. Below it are two white input fields with rounded corners, labeled 'Email' and 'Password' from top to bottom. At the bottom of the form, there are two green buttons with white text: 'Create an account' on the left and 'Login' on the right.

Рисунок 3.11 – Форма логіну

Під час реєстрації та логіну користувачу потрібно ввести всі необхідні дані вірно, такі як пароль, ім'я користувача, електронну адресу, також не

можна залишати поля пустими, або заповненими з помилками, інакше з'явиться повідомлення про помилку, приклад однієї з яких продемонстровано на рисунку 3.12, та користувач не зможе зробити наступний крок, оскільки кнопка буде несприйнятлива для натискання, як це продемонстровано на рисунку 3.13. Це забезпечує захист від неправильного або неповного введення даних, гарантує, що всі необхідні дані будуть отримані, та підвищує безпеку і якість користувацького досвіду. Впровадження таких перевірок допомагає уникнути можливих проблем, пов'язаних з некоректним введенням даних, та забезпечує більш надійний і безпечний процес реєстрації та авторизації.

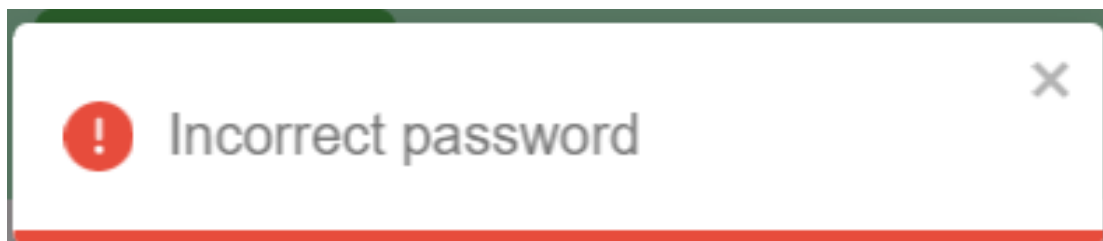


Рисунок 3.12 – Приклад помилки при введенні не вірного паролю

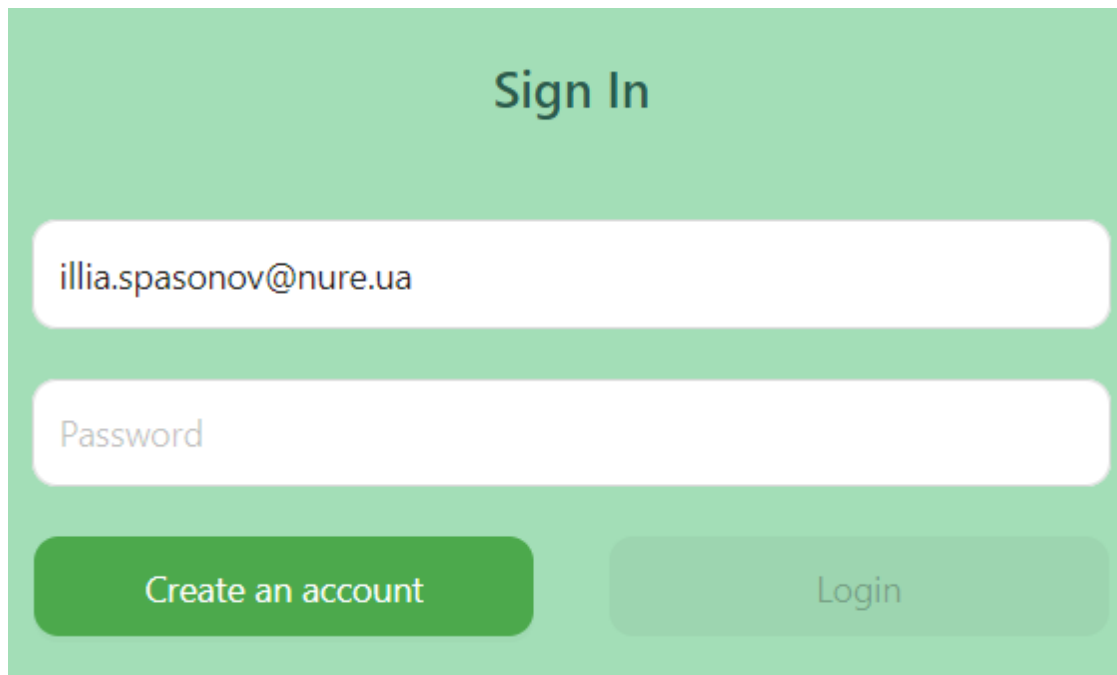


Рисунок 3.13 – Несприйнятлива для натискання кнопка логіну

При завершенні реєстрації або входу до облікового запису хедер головної сторінки змінюється, вітаючи користувача зі входом до персонального кабінету та додаючи дві кнопки для зміни даних або виходу з аккаунта, як це продемонстровано на рисунку 3.14.

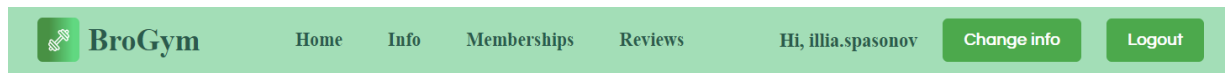


Рисунок 3.14 – Оновлений хедер головної сторінки

Також користувачу відкриваються вже абсолютно нові функції, перша з них – можливість пошуку або просто перегляд дуже великої кількості будь-якої вправи для тренувань, що зображено на рисунку 3.15.

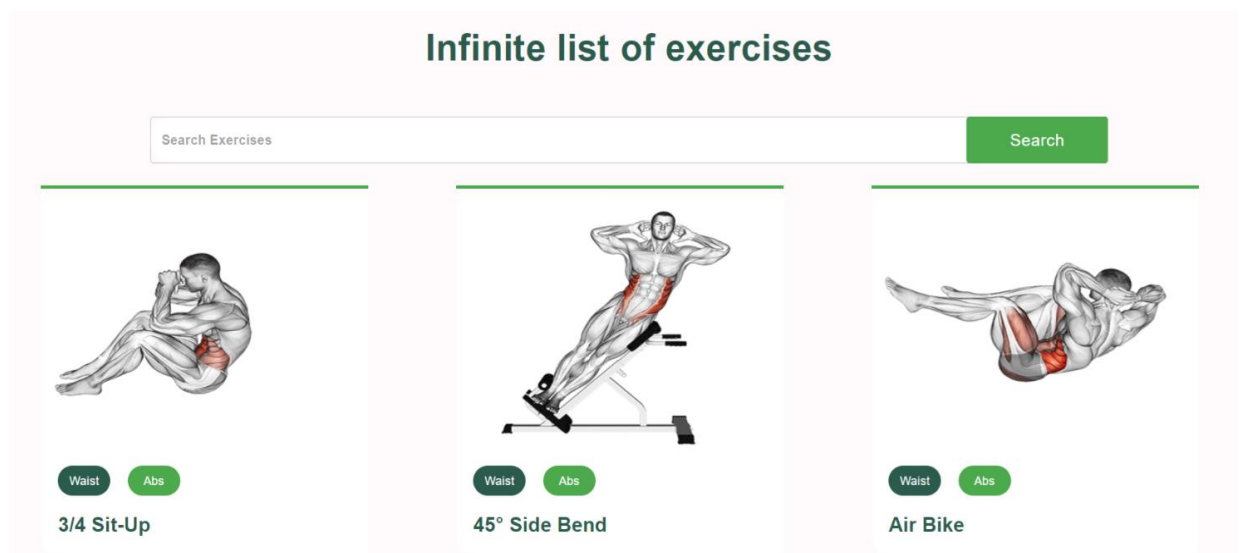


Рисунок 3.15 – Пошук та безмежний список вправ для тренувань

З цього списку користувач може вибрати будь-яку вправу, натиснути на неї, та для нього буде відкрита нова сторінка з детальним описом вибраної вправи, інформація про те, які групи м'язів вона активує (рис. 3.16), посилання на відео інших користувачів, котрі показують правильну техніку при виконанні (рис. 3.17), а також список схожих вправ за групою м'язів (рис. 3.18) та обладнанням (рис. 3.19). Ця функція дозволяє користувачам отримати детальну інформацію про вправи, які вони обирають, щоб

правильно виконувати їх та досягати максимальних результатів у тренуваннях. Посилання на відео дозволяє переглянути демонстрацію вправи в реальному часі, що сприяє кращому розумінню правильної техніки та уникненню травм. Крім того, список схожих вправ допомагає розширити або змінити тренувальну програму, щоб досягти різноманітності та ефективності у тренуваннях. Така детальна інформація сприяє покращенню користувацького досвіду та досягненню бажаних фізичних результатів.

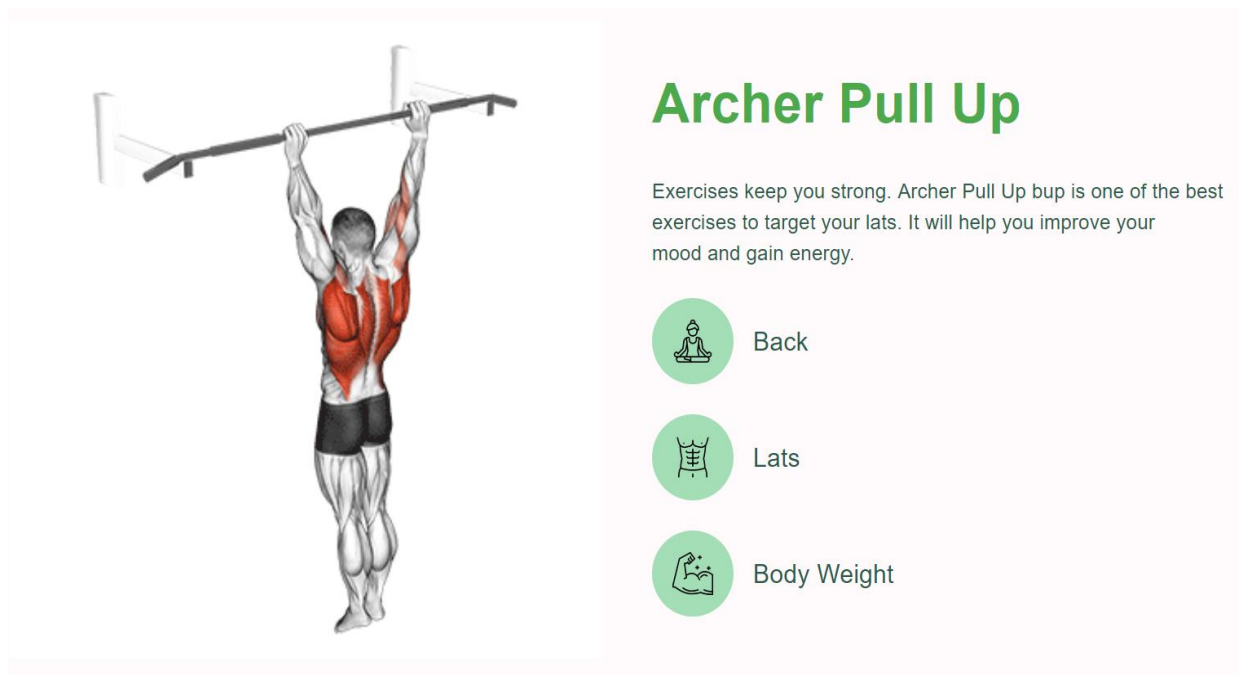


Рисунок 3.16 – Сторінка з детальною інформацією про вибрану вправу

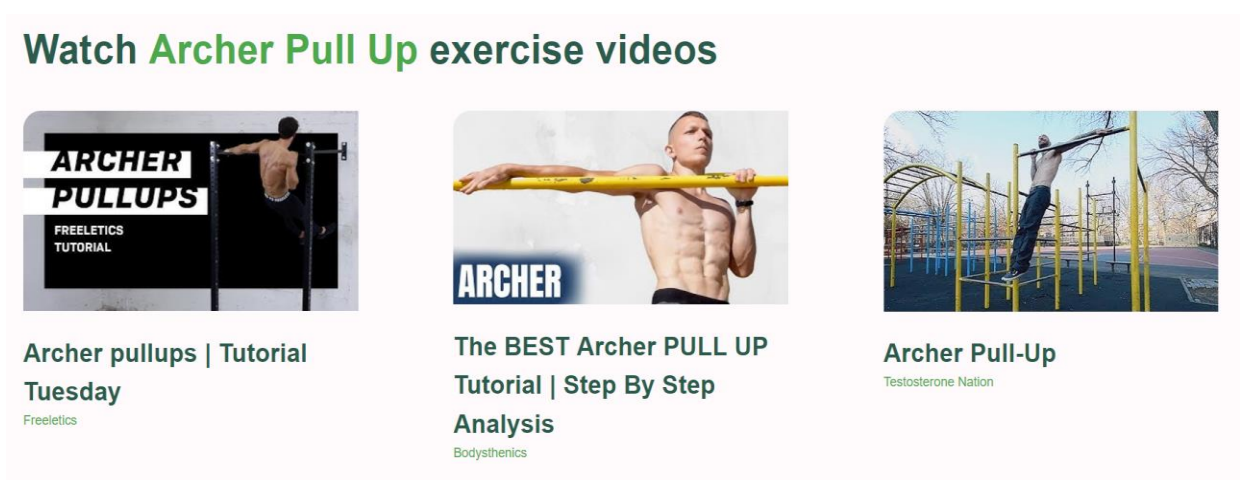


Рисунок 3.17 – Меню з відео інших користувачів по вибраній вправі

### Similar Target Muscle exercises

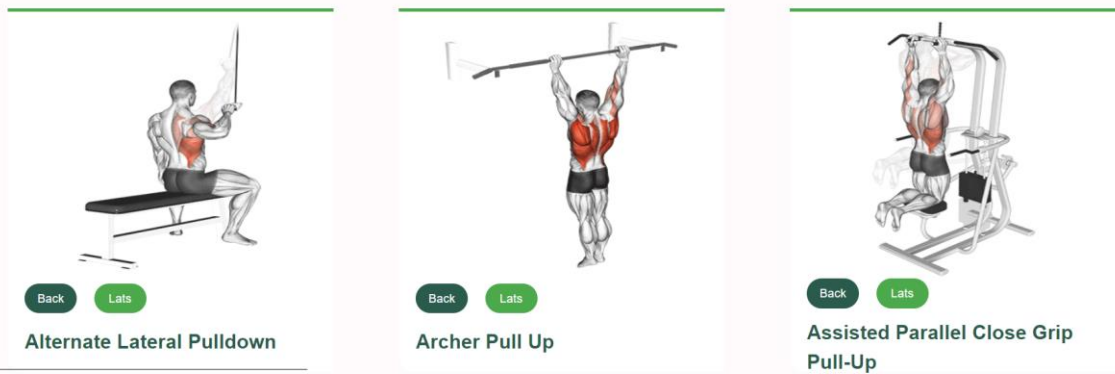


Рисунок 3.18 – Список схожих вправ за групою м'язів

### Similar Equipment exercises

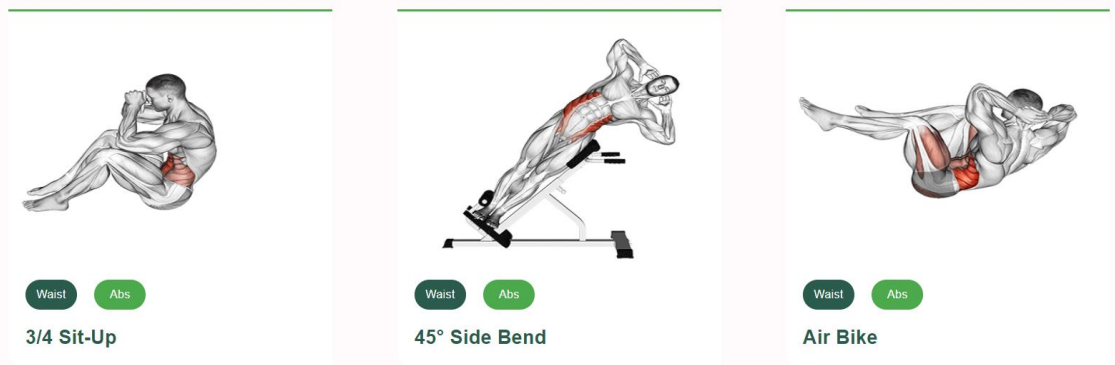


Рисунок 3.19 – Список схожих вправ за обладнанням

Наступною функцією є змогу відкрити сторінку з абонементом та вже придбати собі один з трьох варіантів, в залежності від потреб. Варіанти абонементів представлені на один місяць, три місяці та один рік, з відповідними щомісячними цінами, в залежності чим довше буде діяти підписка, тим менше буде зніматися щомісячна оплата. Після вибору конкретного абонементу, користувач здійснює оплату за допомогою своєї банківської карти, що зображено на рисунку 3.20. Головне також увести всі дані карти вірно, бо інакше оплату неможливо буде провести. В разі успішної оплати, абонемент додається до облікового запису користувача та йому буде відображено скільки часу залишилися витікання терміну купленого

абонементу, що продемонстровано на рисунку 3.21. Це дозволяє користувачам зручно та ефективно вибирати та придбавати абонементи безпосередньо через вебзастосунок. Оплата за допомогою банківської карти забезпечує швидкий та безпечний процес транзакції. Важливо враховувати правильність введення даних картки, щоб уникнути будь-яких проблем з оплатою.

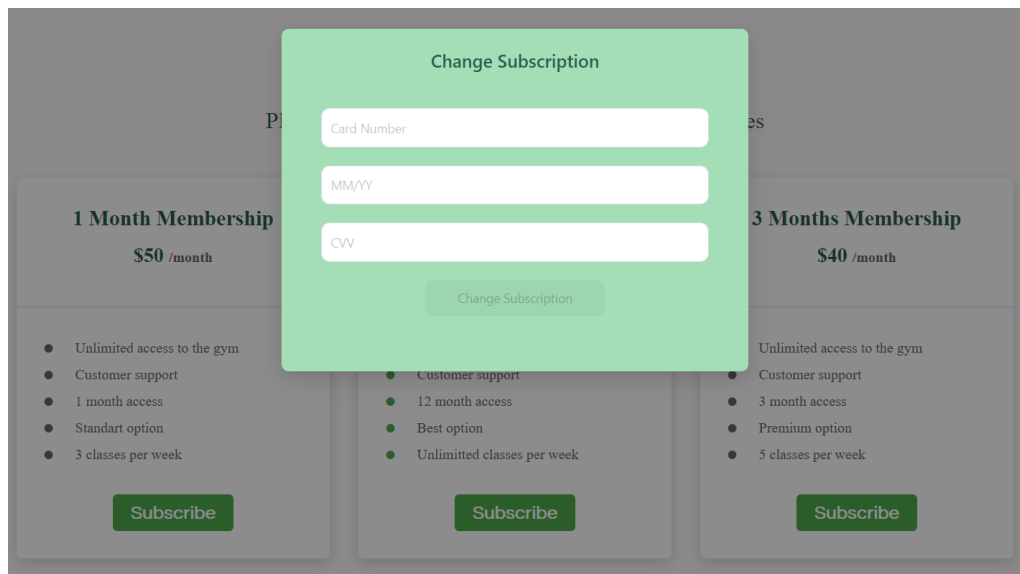


Рисунок 3.20 – Процес оплати при купівлі абонементу

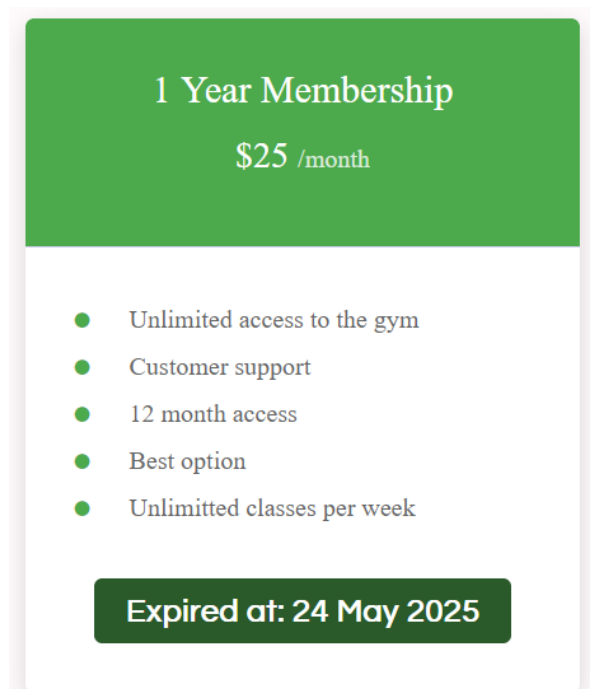


Рисунок 3.21 – Придбаний абонемент з терміном його закінчення

Якщо користувач вже є клієнтом спортивного залу, тобто придбав абонемент, для нього відкривається змога отримати власний QR-код, який буде діяти як персональний ключ при входу у спортзал. Тобто, після успішної оплати, користувачу надається не тільки абонемент, а ще й спеціальний код із різних шістнадцяти символів, який показано на рисунку 3.22, котрий може бути перероблено на приватний QR-код, що зображено на рисунку 3.23, при натисканні на котрий його буде завантажено у файли персонального комп'ютера. Отримання персонального QR-коду для доступу до спортивного залу дозволяє зручно та ефективно пройти процедуру входу без необхідності носити з собою фізичний абонемент. Цей QR-код може бути збережений у мобільному пристрої користувача та використаний при кожному візиті до залу, що додає зручності та безпеки, адже не потрібно турбуватися про втрату абонементу або його погашення.

**Your entrance code: UZI3vXrRH0RVKbRe**

Рисунок 3.22 – Персональний код для входу у спортивний зал

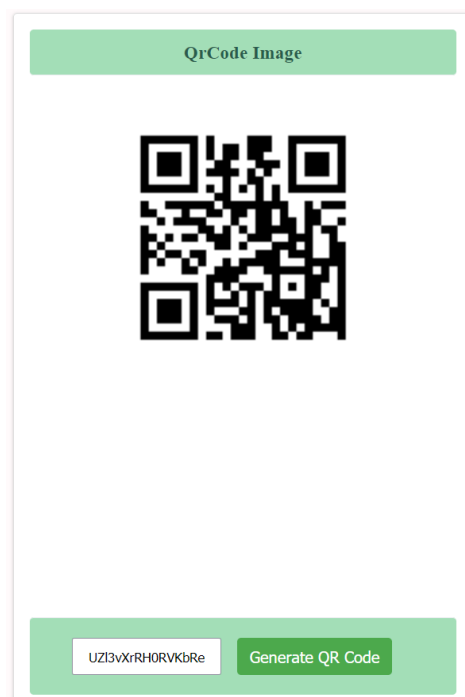


Рисунок 3.23 – Конвертований персональний код у QR-код

Для надійності працездатності QR-коду додано можливість перевірки його аутентифікації за допомогою двох способів:

– завантажити код з комп'ютера на вебсайт, де за допомогою алгоритму буде вираховано точний набір символів та відображено користувачу для перевірки у вигляді строки (рис. 3.24);

– активувати вебкамеру на комп'ютері або ноутбуку, піднести вже телефон з вже завантаженим до нього QR-кодом, котрий буде зчитано та також відображено набір символів для перевірки у вигляді строки (рис. 3.25).



Рисунок 3.24 – Верифікація QR-коду за допомогою завантаження його на вебсайт

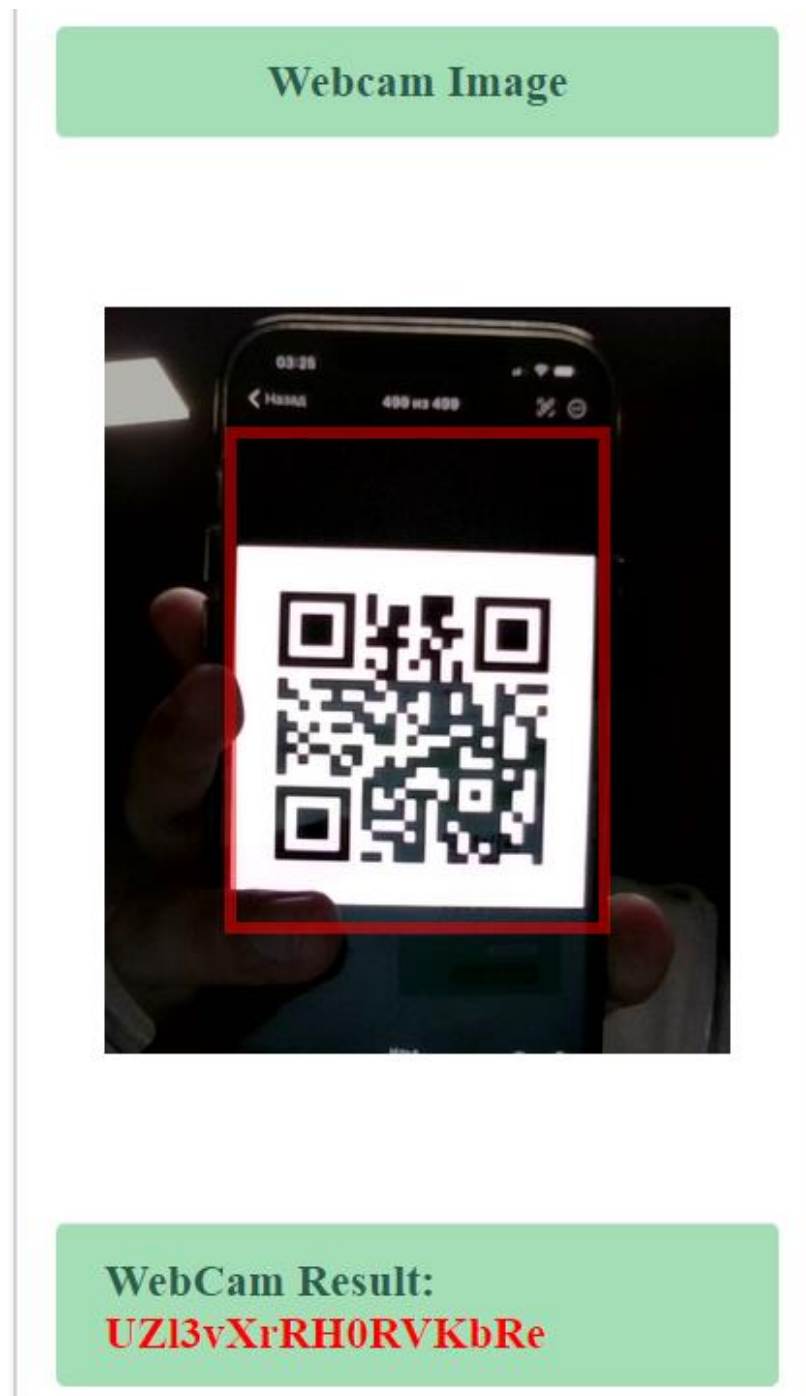


Рисунок 3.25 – Верифікація QR-коду за допомогою вебкамери

Не варто забувати, що при реєстрації користувач міг ввести якісь дані неправильно, для того щоб це виправити також додана можливість редагування облікового запису при введенні усіх нових даних у вірному форматі, зображеного на рисунку 3.26 та успішно зредаговане ім'я користувача на рисунку 3.27. Це дозволить користувачам оновлювати свої особисті дані у разі зміни інформації або виявлення помилок. Важливо мати

можливість корегувати дані, щоб забезпечити точність та актуальність інформації в обліковому записі, оскільки редагування облікового запису сприяє покращенню користувацького досвіду та забезпечує користувачам зручність у керуванні своїм профілем.

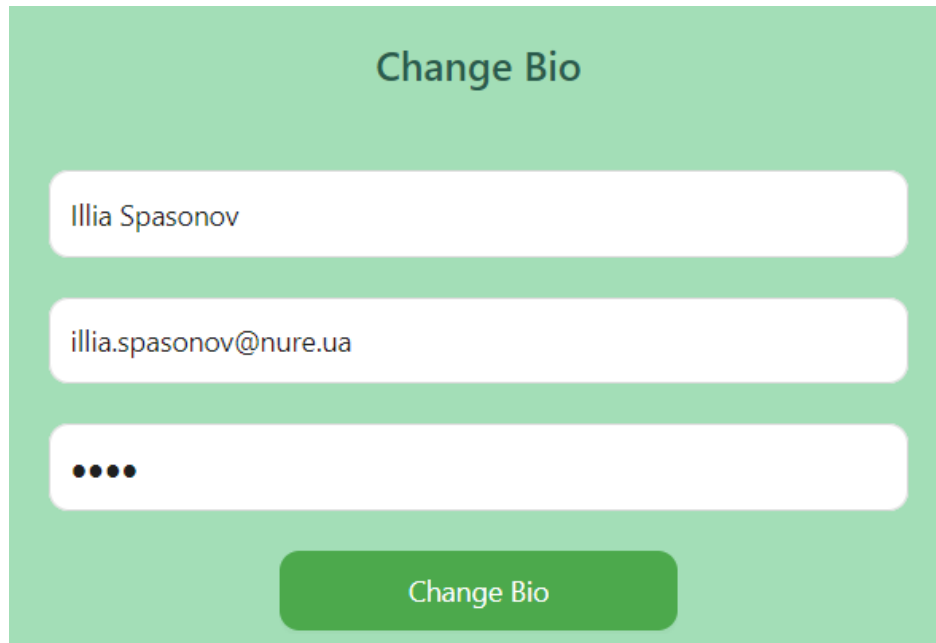


Рисунок 3.26 – Редагування облікового запису користувача

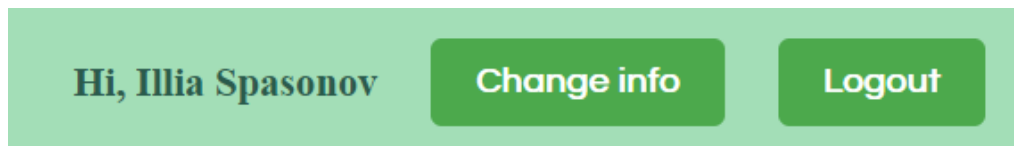


Рисунок 3.27 – Успішне редагування ім'я користувача

Роль адміністратора важлива для забезпечення безпеки та правильного функціонування вебзастосунку. Він відповідає за моніторинг та аналіз даних, виявлення можливих загроз безпеці, впровадження заходів захисту, а також за розвиток та підтримку системи у цілому. Адміністратор має повний доступ до адміністративних інструментів, необхідних для ефективного управління вебзастосунком та забезпечення його надійності.

В нашому випадку адміністратор має доступ до статистики по проданим абонементам, що зображено на рисунку 3.28.

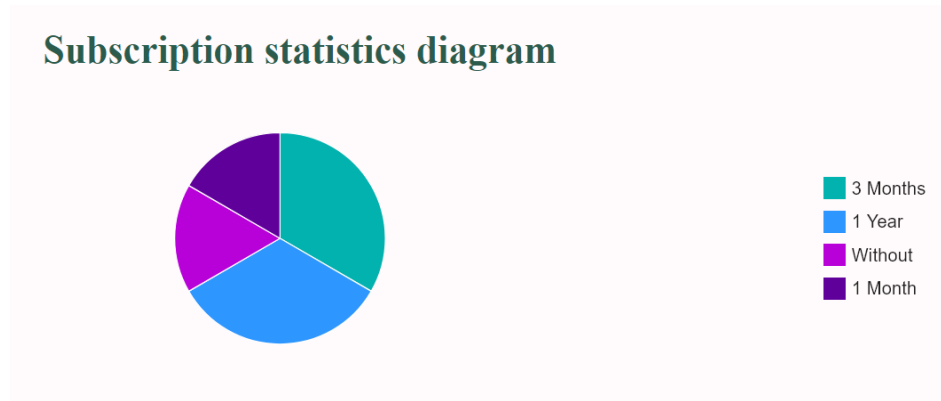


Рисунок 3.28 – Статистика по проданим абонементам у панелі адміністратора

Завершено перевірку всіх можливих сценаріїв системи. Весь функціонал застосунку працює коректно, і жодних дефектів виявлено не було. Відсутність виявлених дефектів свідчить про високу якість та надійність розробленого застосунку. Тепер, коли всі функції працюють коректно, можна зосередитися на впровадженні та підтримці продукту, забезпечуючи задоволення користувачів та досягаючи їхніх цілей. Важливо продовжувати моніторити роботу застосунку та реагувати на будь-які можливі проблеми для забезпечення тривалої та успішної експлуатації.

### 3.4 Перспективи подальшої роботи

Розроблений вебзастосунок надає можливість користувачам авторизуватися та реєструватися у програмі, змінювати персональні дані, вибирати та купувати різні абонементи у спортивний зал, переглядати корисні вправи для тренувань, а також мати доступ до свого QR-коду, який діє як персональний ключ клієнта.

У майбутніх випусках застосунку планується включення наступних опцій:

- можливість створювати власні програми тренувань, комбінуючи вже надані списки вправ;

- додати більше можливих ролей до застосунку, таких як персональний онлайн-тренер, з можливістю проводити з ним тренування онлайн;
- live-chat з адміністраторами або тренерами спортивного залу для відповідей на питання та вирішення проблем;
- сторінку з раціоном харчування під різні види задач, чи то є схуднення, чи набір м'язової маси;
- додати можливість зміни мови застосунку для користувачів з інших країн;
- розробити окрему сторінку, яка буде виглядати як соціальна мережа, де користувачі зможуть публікувати фотокартки себе та свого результату;
- перенести даний застосунок у мобільну версію для смартфонів на базі IOS та Android.

## ВИСНОВКИ

У рамках кваліфікаційної роботи був розроблений і реалізований вебзастосунок для спортивного залу.

Під час виконання цієї роботи було проаналізовано багато різних вебсайтів для спортивних залів для більш детального розуміння їх можливостей. виходячи з цієї інформації було розроблено сучасний вебпристрій для зручності та залучення клієнтів до спортивної зали, та опрацьовано наступні завдання:

- на головній сторінці вказано всю необхідну інформацію, що потрібна клієнту при виборі спортзалів;
- можливість реєстрації нового користувача для створення облікового запису;
- можливість оформити абонемент онлайн з подальшим отриманням особистого QR-коду який діє як персональний ключ для входу в спортивний зал;
- перегляд багатьох корисних вправ для тренувань усередині застосунку;
- проведено тестування розробленого застосунку, за результатами якого були визначені переваги і недоліки даного програмного застосунку.

У ході роботи завдяки використанню React і Node.js, було створено зручний і легко масштабований інтерфейс користувача, а також ефективно оброблювання даних на серверній стороні. Також використовувалися сучасні технології і фреймворки, що дозволили забезпечити високу швидкість роботи та підвищити безпеку даних користувачів у застосунку.

Результати роботи апробовано у вигляді тез доповідей під час VIII Міжнародної науково-практичної конференції «INNOVATIVE DEVELOPMENT OF SCIENCE, TECHNOLOGY AND EDUCATION» [30].

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Nolimits clubs. URL: <https://nolimitsclubs.com.ua/> (дата звернення 22.04.2024).
2. Gym4Fit. URL: <https://sunmall.com.ua/katalog/sport/gym4fit> (дата звернення 22.04.2024).
3. Gym4U. URL: <https://www.gym4u.com.ua/> (дата звернення 22.04.2024).
4. Glofox: how to create a great gym website right now. URL: <https://www.glofox.com/blog/how-to-create-a-great-gym-website-right-now/> (дата звернення 22.04.2024).
5. My Personal Trainer Website: the 20 best gym website designs. URL: <https://www.mypersonaltrainerwebsite.com/blog/the-20-best-gym-website-designs> (дата звернення 22.04.2024).
6. Robin Wieruch (2022) The Road to Learn React: Your Journey to Master Plain Yet Pragmatic React. Js, 2, pp. 193-210.
7. Jeffrey Zeldman (2003) Designing with web standards, 5, pp. 234-310.
8. Health Club Management: Everyones talking about Staffless gyms. URL: <https://www.healthclubmanagement.co.uk/health-club-management-features/Everyones-talking-about-Staffless-gyms/36048> (дата звернення 22.04.2024).
9. Tvoroshenko I., and Maksimenko H. (2021) Research of regression and modular testing of web applications, Abstracts of IV international science conference «Science, theory and practice» (October 12-15, 2021). Tokyo, Japan, pp. 406-411.
10. Rawat, P., & Mahajan, A. N. (2020). ReactJS: A modern web development framework. International Journal of Innovative Science and Research Technology, 5(11), 698-702.
11. Ranjan, A., Sinha, A., & Battewad, R. (2020). JavaScript for Modern Web Development: Building a Web Application Using HTML, CSS, and JavaScript. BPB Publications.

12. Saks, E. (2019). JavaScript Frameworks: Angular vs React vs Vue.
13. Tvoroshenko, I. (2020). Information technologies for decision-making on the conditions of spatially distributed objects. In I International Scientific and Practical Conference. Problems and perspectives of modern science and practice, Austria (pp. 40-44).
14. Tvoroshenko, I.S., & Maksimenko, H. (2021). To the question of analysis of existing mechanisms of web application testing.
15. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2022). Tools for Fast Metric Data Search in Structural Methods for Image Classification. IEEE Access, 10, pp. 12738-12746.
16. Mashtalir, V., Ruban, I., & Levashenko, V. (Eds.). (2019). Advances in Spatio-Temporal Segmentation of Visual Data (Vol. 876). Springer Nature.
17. Kuzminska, O., Mazorchuk, M., Morze, N., & Kobylin, O. (2019, June). Digital learning environment of ukrainian universities: The main components to influence the competence of students and teachers. In International Conference on Information and Communication Technologies in Education, Research, and Industrial Applications (pp. 192-210). Springer, Cham.
18. Tvoroshenko I., and Maksimenko H. (2021) Research of regression and modular testing of web applications, Abstracts of IV international science conference «Science, theory and practice» (October 12-15, 2021). Tokyo, Japan, pp. 319-382.
19. Sitnikov D., Titova O., Minukhin S., Kovalenko A., Titov S. (2018). Informativity of Association Rules from the Viewpoint of Information Theory. Conference: 2018 IEEE International Scientific-Practical Conference Problems of Infocommunications. Science and Technology.
20. Tvoroshenko, I. (2020). Information technologies for decision-making on the conditions of spatially distributed objects. In I International Scientific and Practical Conference. Problems and perspectives of modern science and practice, Austria (pp. 41-45).

21. Гороховатський В., Передрій О., Творошенко І., Марков Т. (2023) Матриця відстаней для множини компонентів структурного опису як інструмент для створення класифікатора зображень, Сучасні інформаційні системи, 7(1), С. 6-11.

22. Iryna, T., & Maksym, K. (2021). Research results of functional, white box and smoke testing methods for mobile applications. Trends in science and practice of today, 5, 415.

23. Tvoroshenko, I.S., & Bielinskyi, Y. (2021). On the features of methods of processing and recognition of handwritten text.

24. Творошенко, І.С. (2021). Технології прийняття рішень в інформаційних системах: навч. посібник. Харків: ХНУРЕ.

25. Titova O., Vysotskyi D. (2021) The development of a subsystem for palliative patients information support. Trends in science and practice of today. Abstracts of V International Scientific and Practical Conference. Ankara, Turkey. Pp. 401–404.

26. Mashtalir, S. V., Stolbovoi, M. I., & Yakovlev, S. V. (2019). Hybrid Approach to Clustering Various Lengths Video. Journal of Automation and Information Sciences, 50(3).

27. Gupta, S., & Sharma, R. (2021). React Reimagined: Revolutionizing Web Application Development. Journal of Web Engineering, 8(2), 111-122.

28. Patel, M., & Desai, N. (2019). The Power of Django: Crafting Robust Web Applications. Web Development Quarterly, 12(4), 242-251.

29. Singh, K., & Patel, D. (2022). Vue.js: The Rising Star in Web Application Development. Web Development Journal, 15(3), 207-213.

30. Спасьонов І., Машталір С. (2024). ГЕНЕРАЦІЯ ТА ВЕРИФІКАЦІЯ QR-КОДУ ЯК ПЕРСОНАЛЬНОГО КЛЮЧА КЛІЄНТА СПОРТИВНОГО ЗАЛУ. VIII Міжнародна науково-практична конференція «INNOVATIVE DEVELOPMENT OF SCIENCE, TECHNOLOGY AND EDUCATION» (May 9 – 11, 2024). Vancouver, Canada, с. 258-261.