

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)

на тему:

**«Додаток "SportLife" для спортивних занять різних
груп м'язів»**

Виконав студент групи ПЗк-23-1
ЛЕВКОВИЧ Олександр Олексійович

Керівник роботи:
ГОДЛЕВСЬКИЙ Юрій Олександрович

Рецензент:
САВІЦЬКИЙ Роман Святославович

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

Зав. кафедри інженерії
програмного забезпечення
Тетяна Вакалюк
«26» лютого 2025 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **ЛЕВКОВИЧ Олександр Олексійович**

Керівник роботи: **ГОДЛЕВСЬКИЙ Юрій Олександрович**

Тема роботи: **«Додаток "SportLife" для спортивних занять різних груп м'язів»**,

затверджена Наказом закладу вищої освіти від **«14» лютого 2025 р., №61/с**

Вихідні дані для роботи: об'єктом дослідження є забезпечення покращених та ефективних умов тренування для людей які не мають змогу відвідувати тренувальні заклади.

Консультанти з бакалаврської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Годлевський Юрій	27.02.2025р.	27.02.2025р.
2	Годлевський Юрій	27.02.2025р.	27.02.2025р.
3	Годлевський Юрій	27.02.2025р.	27.02.2025р.

РЕФЕРАТ

Випускна кваліфікаційна робота бакалавр складається з вебдодатку тренувальної платформи та пояснювальної записки. Пояснювальна записка до випускної кваліфікаційної роботи містить 81 сторінок, 77 ілюстрацій, 29 таблиць та 30 літературних джерел.

Метою роботи є створення соціально розширеного вебдодатку, в якому будуть зручні інструменти дистанційних тренувань та соціального спілкування.

У роботі було запропоновано створення соціального вебдодатку для тренувань на базі технологічного стеку MERN (MongoDB, Express.js, React, Node.js) визначено основні цілі створення, розглянуто різні способи вирішення задачі, розглянуто аналоги вибраної теми, такі як MyFitnessPal, Nike та Peleton, проаналізовано методи обробки і аналізу даних. Також розроблена база даних для системи та описано архітектуру і взаємодію деяких основних компонентів.

КЛЮЧОВІ СЛОВА: ВЕБДОДАТОК, MERN, ТРЕНУВАЛЬНА ПЛАТФОРМА, СОЦІАЛЬНИЙ.

					ІПЗ.КР.Б – 121 – 25 – ПЗ						
Змн.	Арк.	№ докум.	Підпис	Дата	Додаток "SportLife" для спортивних занять різних груп м'язів			Літ.	Арк.	Акрушів	
Розроб.		Левкович О.О.									
Керівник		Годлевскій Ю.О.								3	95
								ФІКТ Гр. ІПЗк-23-1			
Н. Контр.		Савіцкій Р.С.									
Затверд.		Вакалюк Т.А.									

ABSTRACT

The bachelor's final qualification work consists of a web application of a training platform and an explanatory note. The explanatory note to the final qualification work contains 81 pages, 77 illustrations, 29 tables and 30 literary sources.

The aim of the work is to create a socially enhanced web application, which will have convenient tools for remote training and social communication.

The work proposed the creation of a social web application for training based on the MERN technology stack (MongoDB, Express.js, React, Node.js), the main goals of creation were determined, various ways of solving the problem were considered, analogues of the selected topic, such as MyFitnessPal, Nike and Peleton, methods of data processing and analysis were analyzed. A database for the system was also developed the architecture and interaction of some of the main components were described.

KEYWORDS: WEB APPLICATION, MERN, TRAINING PLATFORM, SOCIAL.

		Левкович О.О.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

ЗМІСТ

РЕФЕРАТ	3
ABSTRACT	4
ЗМІСТ	5
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	6
ВСТУП.....	7
РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ЗАСТОСУВАННЯ ТРЕНУВАЛЬНОГО ДОДАТКУ	9
1.1 Постанова задачі.....	9
1.2 Аналіз аналогів програмного продукту	10
1.3 Вибір архітектури та обґрунтування технологій вебдодатку для дистанційних тренувань	15
1.4 Обґрунтування вибору інструментальних засобів.....	17
Висновок для першого розділу	18
РОЗДІЛ 2. ПРОЕКТУВАННЯ ВЕБСАЙТУ ДИСТАНЦІЙНИХ ТРЕНУВАНЬ	20
2.1 Визначення варіантів використання.....	20
2.2 Розробка бази даних системи.	28
2.3 Проектування та реалізація алгоритмів роботи системи	34
2.4 Реалізація функціоналу системи тренувального вебдодатку.	38
Висновок другого розділу	41
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ВЕБДОДАТКОМ.....	42
3.1 Порядок встановлення та налаштування параметрів системи.	42
3.2 Структура інтерфейсу тренувального вебдодатку.....	43
3.3 Тестування роботи програмного комплексу	73
Висновок до третього розділу.....	77
ВИСНОВКИ	78
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	79
ДОДАТКИ.....	82

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API – application programming interface

MERN – стек з набору таких технологій як – MongoDB, Express.js, React, Node.js

REST – Representational State Transfer

CRUD – create, read, update, delete

БД – база даних

		Левкович О.О.			ІІЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Актуальність теми. Сучасний світ технологій все частіше стають невід'ємною частиною повсякденного життя, зокрема в галузі охорони здоров'я та фізичної активності. Зростаюча популярність здорового способу життя потребує у створенні цифрових рішень, які допоможуть користувачам у вирішенні сучасних проблем зі здоров'ям у дистанційному плані, найчастіше у більшій кількості варіантів тренування створено в спеціальних закладах, що не є комфортно для деяких людей у відвідування через роботу або інші подібні справи.

Щоб додаток для планування тренувань і догляду за здоров'ям справді працював ефективно й був корисним для користувачів, важливо одразу визначити ключові вимоги до його функціональності. Треба прагнути до забезпечення найефективніших та комфортних умов для користувачів додатку тобто, розробити додаток без збоїв, щоб інтерфейс був зрозумілим і приємним, а користувачі могли досягати своїх цілей – будь то покращення фізичної форми, підтримання або покращення здоров'я чи просто бажання займатися спортом. Також додаток має надавати можливість користувачам комунікувати між собою – ділитися власними досягненнями з тренером або обмінюватися враженнями чи думками з іншими.

Для забезпечення таких умов з'явилися подібні платформи для дистанційних тренувань, які забезпечують сконцентруватися за своїм здоров'ям, полегшують життя у плануванні справ та тренувань. Це включає подібні платформи як Nike, Peleton та інші. Такі платформи створюють максимально комфортні та ефективні умови для людей, прислуховуючись до побажань користувачів. Також подібні умови створені і для тренерів які бажають працювати та забезпечити для людей більш обширний спектр тренувань у якомусь з напрямлень.

Метою є розробка вебдодатку для вирішення проблем з бажанням але не завжди мають змогу відвідувати тренажерні зали через брак часу, домашні

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		7

справи чи інші обставини. Кінцевий результат цього додатку повинен спростити для людей змогу тренувань зробити тренування зручними у домашніх умовах.

Для досягнення запропонованої мети необхідно виконати наступні завдання, провести аналіз аналогів, сформулювати вимоги до функціоналу майбутнього вебдодатку з врахуванням потреб цільової аудиторії, розробити дизайн інтерфейсу, орієнтуючись на простоту далі реалізувати фронтенд і бекенд та забезпечивши стабільну роботу, налаштувати базу даних для зберігання інформації, забезпечити безпеку персональних даних користувачів згідно з сучасними стандартами також провести тестування системи та виправити виявлені помилки.

Об’єктом дослідження є забезпечення покращених та ефективних умов тренування для людей які не мають змогу відвідувати тренувальні заклади.

Предметом дослідження є функціональність тренувальних додатків, їх структура, ідея, інтерфейс, способи забезпечення комфортних умов тренувань або інших наданих додатком услуг.

Додаток має базовий функціонал дистанційних тренувань з записом результатів, створенням статистики проходження які покращують знання в розумінні тренувань.

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		8

РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ЗАСТОСУВАННЯ ТРЕНУВАЛЬНОГО ДОДАТКУ

1.1 Постанова задачі

У сучасному світі чимало людей зустрічаються з труднощами через сидячий спосіб життя, стрес, перевантаження та брак часу для регулярних занять у спортивному залі.

Разом з тим, зростає розуміння важливості фізичної активності для підтримки здоров'я, зменшення ризику захворювань та загального поліпшення якості життя. У відповідь на ці виклики великої популярності набувають дистанційні тренування – форма фізичної активності, яку можна виконувати вдома або в будь якому зручному місці з використанням цифрових технологій.

Однак на практиці існуючі рішення не завжди повністю відповідають потребам користувачів. До основних проблем, що виникають під час організації дистанційного тренувального процесу, можуть належати такі ситуації як неможливість зворотного зв'язку з тренером, відсутність єдиного інструменту для прогресу та комунікації, недостатньо або відсутність знань для правильного та безпечного виконання вправ також складність для фітнес-тренерів та спортивних клубів у віддаленому супроводі великої кількості клієнтів.

Отже, виникає потреба у створенні інформаційного додатку нового типу, що поєднував би в собі зручність для користувача, гнучкість у налаштуванні програм тренувань, можливість комунікації з тренером, зберігання статистики, а також функціональність для адміністрування спортивної діяльності.

Метою даної кваліфікаційної бакалаврської роботи є розробка інформаційного додатку SportLife для організації дистанційних спортивних тренувань, який задовольняє потреби як звичайних користувачів, фітнес-тренерів та адміністраторів цього додатку.

У ході роботи над проєктом передбачається здійснення кількох ключових етапів, що охоплюють як аналітичну, так і практичну частину створення інформаційного додатку для дистанційних спортивних тренувань. На першому

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		9

етапі буде проведено детальний аналіз сучасних інформаційних технологій, які використовуються у сфері фітнесу та онлайн-тренінгу.

Особливу увагу буде приділено вивченню потреб кінцевих користувачів як спортсменів, так і тренерів. Розуміння їх очікувань щодо функціональності, зручності використання та взаємодії з додатком дозволить розробити оптимальну структуру і логіку роботи системи. На основі зібраних вимог буде сформовано загальну архітектуру майбутнього вебдодатку, яка передбачає ефективну взаємодію між компонентами та можливість масштабування.

У межах розробки функціоналу буде реалізовано інструменти для формування тренувальних планів з урахуванням детальної інформації та цілей користувача. Крім цього, у системі буде створено комунікаційні способи чату або створення постів, також створення кімнати для проведення стрімів, друзям або тренеру з урахуванням збереження статистики, що дозволить зручно відстежувати результати.

Важливим аспектом стане реалізація ефективної взаємодії між користувачем і тренером через чат, коментар, пости або інші інтерактивні засоби. Усі ці елементи будуть об'єднані в єдиний зручний інтерфейс, який забезпечить доступність додатку у вигляді вебверсії.

У результаті реалізації проєкту очікується створення актуального продукту, який дасть можливість не лише покращити фізичну форму користувачів у дистанційній формі, а й оптимізує діяльність тренерів, відкриваючи нові можливості для розвитку дистанційних спортивних послуг.

1.2 Аналіз аналогів програмного продукту

На сучасному ринку додатків для здоров'я та фітнесу існує багато рішень, що полегшують користувачам слідкування за раціоном, тренуваннями та загальним фізичним станом. У цьому розділі ми проаналізуємо три найпопулярніші аналоги: MyFitnessPal[3] (рис. 1.1), Nike Training Club[5] (рис. 1.2) та Peleton[6] (рис.1.3).

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

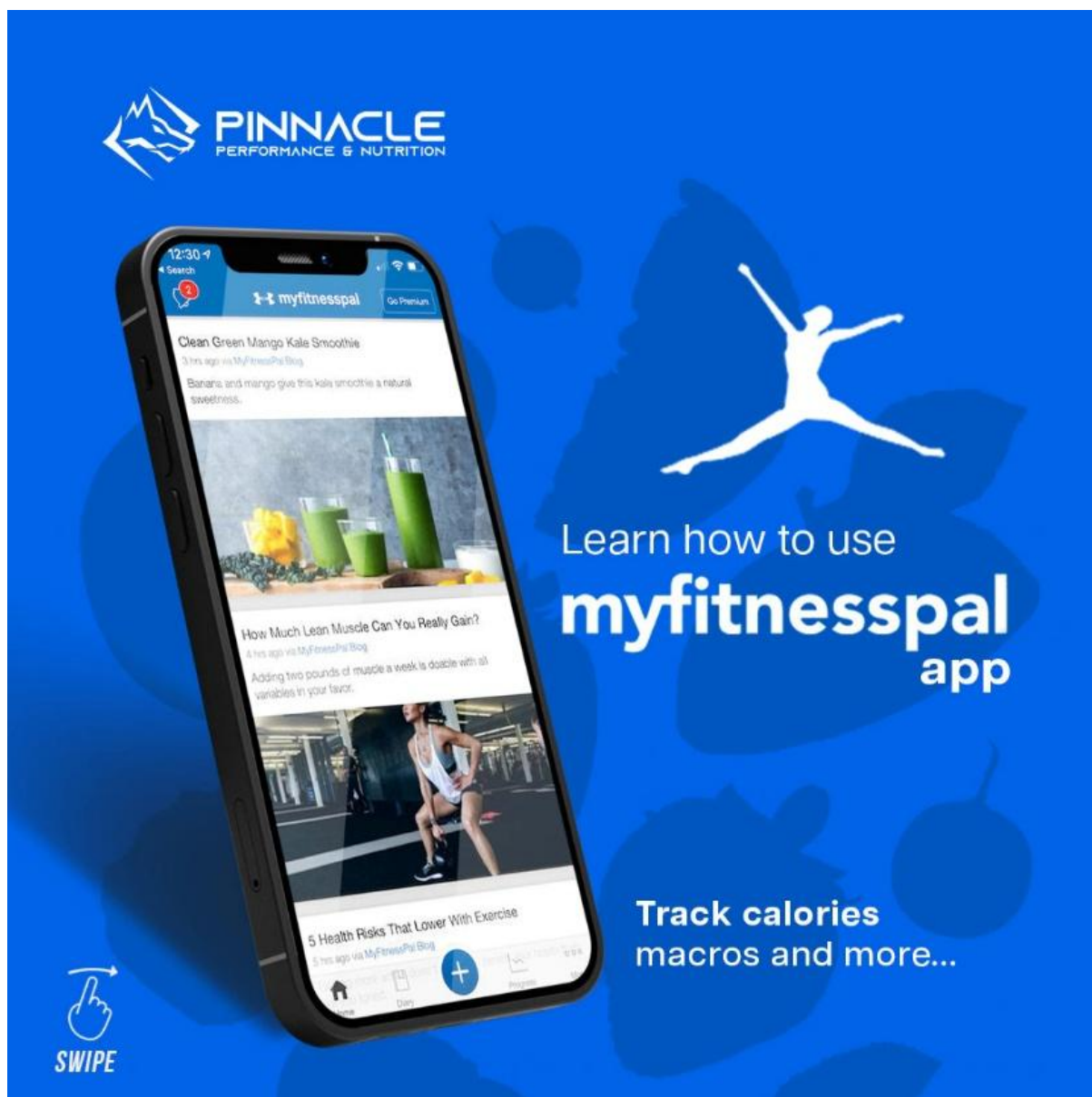


Рис. 1.1 – Реклама програмного продукту MyFitnessPalApp

MyFitnessPal – один із найвідоміших додатків, орієнтованих на облік їжі та контроль за споживанням калорій. Ключові особливості цього продукту:

Функції обліку харчування: Користувач може вести щоденник їжі, додавати продукти з великої бази даних і відстежувати загальну калорійність.

Інтеграція з фітнес-пристроями: Додаток підтримує синхронізацію з багатьма носимими гаджетами, що дає змогу автоматично збирати дані про фізичну активність.

Статистика та звіти: Програма надає графіки, діаграми та звіти, що допомагають користувачам аналізувати свій прогрес.

Мотивація та спільнота: Завдяки спільноті користувачів, можна

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк. 11
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

обмінюватися порадами та отримувати додаткову мотивацію.

Перевагами можна зазначити велику базу даних продуктів, що робить введення даних швидким та зручним, зручний інтерфейс, який не вимагає спеціальної технічної підготовки також безкоштовна базова версія.

Недоліками можна перелічити те що, додаток більше орієнтований на облік їжі, ніж на самі тренування, для користувачів які шукають віддалені тренування, функціонал може бути обмеженим також інтерфейс не завжди виглядає сучасно, що може впливати на розуміння користування додатком.

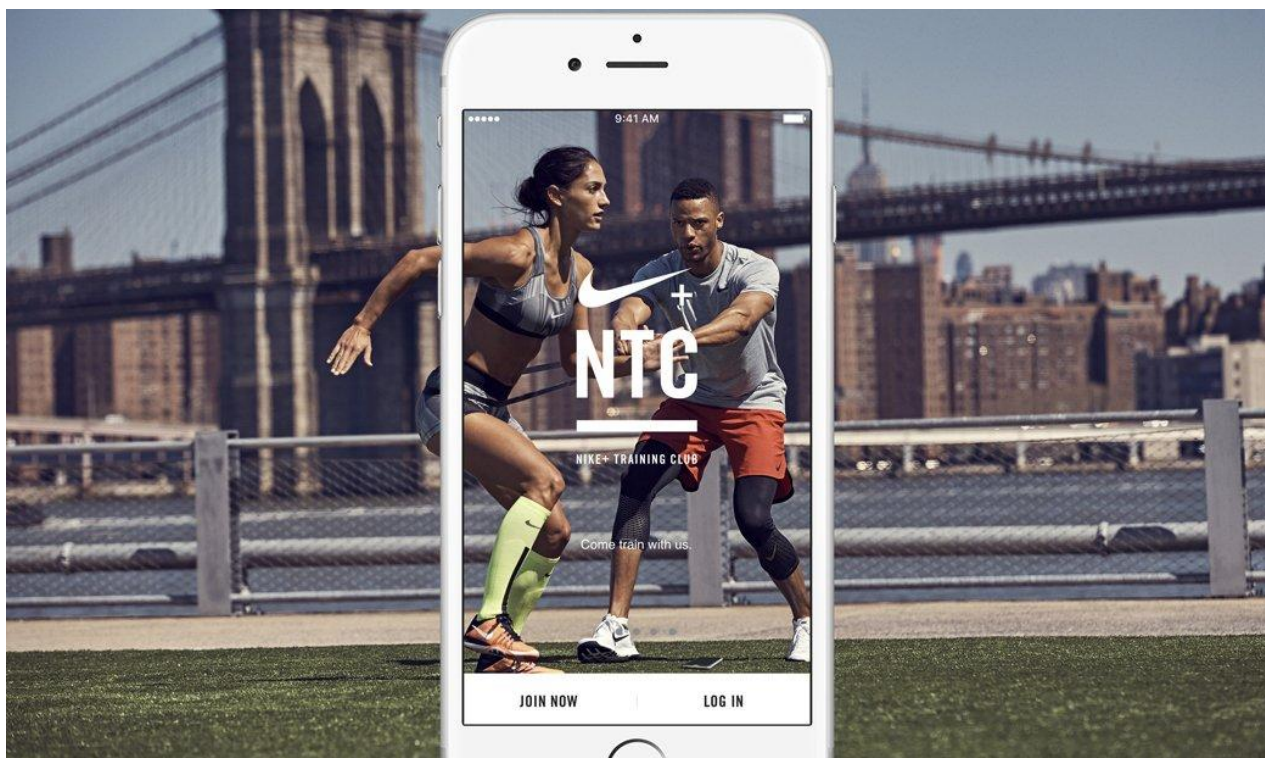


Рис. 1.2 – Логотип програмного продукту Nike Training Club

Аналізуючи додаток Nike, тобто мається на увазі Nike Training Club (NTC), який пропонує безкоштовні тренувальні програми та відеоінструкції. Основними характеристиками NTC личать такі як різноманітність тренувальних програм, додаток містить безліч тренувальних режимів для будь-якого рівня підготовки, високоякісні відео та зображення, що демонструють вправи, роблять процес тренування більш наочним. Також додаток дає великий обсяг порад від професійних тренерів, тобто користувачі отримують рекомендації від досвідчених тренерів, що допомагає правильно виконувати вправи.

Перевагами можна назвати такі сторони як висока якість контенту,

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		12

особливо відеоуроків, широкий вибір тренувальних програм для різних цілей та простий у використанні інтерфейс.

Недоліками можна зазначити відсутність персоналізації тренувального процесу, тобто всі користувачі отримують подібний набір вправ, незалежно від індивідуальних особливостей та можливості інтерактивного зворотного зв'язку.



Рис. 1.3 – Логотип програмного продукту Peleton

Peleton – більш комплексне рішення, орієнтоване на інтерактивні тренування із використанням живих занять та відеоконтенту високої якості.

Основні функції Peleton це те, що користувачі можуть брати участь у живих заняттях, отримувати рекомендації від тренера в реальному часі і змагатися з іншими учасниками.

Відеоуроки з професійними тренерами, зняті з високою якістю, що створює відчуття перебування в реальному тренажерному залі. Також додаток має функції, що дозволяють користувачам спілкуватися між собою, ділитися досягненнями та підтримувати один одного.

В перевагу можу зазначити такі ідеї додатку як високий рівень залученості завдяки інтерактивним і живим тренуванням, сучасний та привабливий дизайн і можливість отримувати зворотний зв'язок від тренера під час тренування.

А саме недоліки зазначу саме такі ідеї як висока вартість підписки, що може

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		13

стати бар'єром для багатьох початківців, необхідність стабільного інтернет-з'єднання, також обмежена доступність для тих, хто віддає перевагу тренуванням у власному темпі, без прямої участі тренера.

Базуючись на отриманих даних, складемо порівняльну табл. 1.1 характеристик згаданих платформ та власного вебдодатку.

Таблиця 1.1

Порівняння трьох аналогів з проектом

Функція	MyFitnessPal	Nike Training Club	Peloton	SportLife
Персоналізація тренувань	Немає	Типові програми тренувань.	Вибір типів та інтенсивності, адаптивність через метрики	Різний вибір тренувальних планів під різні групи. Змога трансляції тренувань.
Відеоконтент	Немає	Понад 100 відеотренувань із демонстраціями вправ та поясненнями тренерів	live-трансляції і записані класи різної тривалості	Live-трансляцій усім користувачам хто знаходиться в кімнаті з карточками вправ гіф або іншим форматом.
Живі заняття & зворотний зв'язок	Немає	Немає	live-класи з відгуками тренера та чатом для учасників	Заняття відбуваються улюбий зручний час та режимі реального часу
Спільнота	Вбудована спільнота та форуми	Виклики та обмін досвідом у соцмережах NTC	live-лідерборд, "high-fives", групові челенджі	Так чати, спільні цілі

Продовження таблиці 1.1

CRM для тренерів/адмін	Немає	Немає	Немає	Так, адмін-панель для управління клієнтами та іншими даними вебдодатку
Вартість	Freemium: базова безкоштовна, Premium \$19.99/місяць або \$79.99/рік	Повністю безкоштовно	App Membership \$12.99/місяць	Безкоштовний Open Source проєкт
Платформи	iOS, Android, Web	iOS, Android	iOS, Android, Web	Web-додаток

Отже, розробка додатку SportLife прагне поєднати вирішення проблем всіх цих продуктів, пропонуючи персоналізований підхід, інтерактивну взаємодію з тренерами, можливість обліку прогресу та збереження історії тренувань.

1.3 Вибір архітектури та обґрунтування технологій вебдодатку для дистанційних тренувань

Для відтворення клієнт-серверної архітектури нам потрібно розробити основну механіку спілкування, тобто клієнт спілкуватися з сервером за допомогою запитів, в той час коли сервер оброблює його та повертає результат або у разі помилки повертає помилку.

REST API являється центральною механікою спілкування серверу, який відповідає саме за обробку запиту від клієнтської сторони.

REST – аббревіатура від REpresentational State Transfer. Це не протокол, а стиль архітектури. Точніше, один з існуючих типів синхронного зв'язку для вебсервісів. Розробники API можуть використовувати REST по різному, обираючи свій власний підхід на етапі розробки[2].

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		15

Також для стилізації вебдодатку було використано звичайний CSS[8] – мову стилів, яка використовується для оформлення HTML документів[9].

JSON (JavaScript Object Notation) – це найпоширеніший формат представлення даних, що слугує для передачі інформації між різноманітними програмами. Він базується на текстовому форматі, що забезпечує комфортний обмін даними між різними мовами програмування[23].

Для втілення першої частини в процесі розробки архітектури вебдодатку було застосовано стек MERN (MongoDB, Express.js, React, Node.js)[1] та такі бібліотеки як SocketIO, SocketIO-client, PeerJs, яка нині є визнаним рішенням для створення сучасних вебдодатків, завдяки її ефективності, можливості масштабування та легкості в роботі тобто це сучасний технологічний стек, який поєднує в собі потужність чотирьох ключових технологій для створення повноцінних вебдодатків.

Ця архітектура базується на декількох ключових складових, кожна з яких відіграє свою певну роль у загальній організації архітектури додатку.

Кожен компонент стеку MERN виконує чітко визначену функцію у процесі розробки вебдодатку:

MongoDB – документно-орієнтована БД NoSQL, яка дозволяє гнучко зберігати структури даних у форматі JSON-подібних документів. Це особливо зручно при зберіганні тренувальних програм, результатів користувача, статистики фізичної активності та іншої інформації з вебдодатком. MongoDB забезпечує високу продуктивність, масштабованість та простоту взаємодії з іншими частинами нашого стеку MERN[25].

Express.js – серверний фреймворк для Node.js, який забезпечує обробку HTTP запитів, маршрутизацію, авторизацію та реалізацію логіки API. У розрізі SportLife він відповідає за створення RESTful ендпоінтів[10].

React.js – бібліотека для побудови інтерфейсу користувача. Вона дозволяє створювати динамічні, реактивні SPA (single page application) з високою швидкістю завантаження та відгуку[13].

Node.js – середовище виконання JavaScript на сервері, яке дозволяє

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		16

створювати масштабовані серверні рішення. Node.js, що важливо для обробки великої кількості запитів у реальному часі [14].

Socket.IO – клієнт інтегрується в клієнтську частину, скажімо, в React-додаток, даючи змогу клієнтам підключатися до кімнат, обмінюватися повідомленнями й отримувати оновлення, не потребуючи перезавантаження сторінки (наприклад, live-сесій тренувань або обміну повідомленнями в чаті). Це забезпечує синхронний процес тренувань для кількох осіб, де всі зміни видно миттєво[16].

PeerJS – обрано для забезпечення прямої відеокommунікації між користувачами за принципом peer-to-peer. Це відкриває можливість організувати трансляцію відео з камери тренера для всіх, хто знаходиться в кімнаті, без потреби використовувати посередницький сервер для передавання відеопотоку [22].

Cloudinary – це хмарне сховище для медіафайлів, яке значно спрощує роботу з зображеннями та відео. Воно автоматично оптимізує файли, роблячи їх легшими для завантаження, наприклад створює пряме посилання, що допомагає прискорити роботу сайту. Також хмарне сховище забезпечує надійність і безпеку, збережені медіафайли не загубляться і будуть доступні глобально навіть при збоях на сервері[7].

В цілому, вибір цих технологій був продиктований завданням, яке стояло перед проєктом – створення інтерактивного спортивного середовища з реальним спілкуванням між його учасниками. Завдяки використанню єдиної мови програмування – JavaScript на кожному з рівнів (від клієнта до серверу та бази даних), функціональний стек MERN та інші технології дають можливість суттєво спростити процес розробки.

Не менш важливо, що всі складові стеку MERN є легко доступні, що дозволяє ефективно працювати без потреби у ліцензуванні, активно використовувати бібліотеки та ресурси спільноти.

1.4 Обґрунтування вибору інструментальних засобів

Для відтворення тренувального додатку важливо відтворити комфортні умови для його розробки, вибрати відповідні інструмента та технології. Такими

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		17

інструментами можна перелічити.

Visual Studio Code – це безкоштовне програмне забезпечення, розроблене Microsoft, що призначене для підтримки розробників у створенні різноманітних програмних продуктів.

Visual Studio Code багатий на корисні інструменти, що роблять процес програмування більш прискореним та комфортним. Перевагами можна назвати такі пункти як те, що вона пропонує підтримку великої кількості мов, включаючи Python, мову розмітки HTML та інші. Далі, зручний редактор коду, з функціями підказок, автозаповнення та підсвічування помилок. Visual Studio Code дозволяє створювати проекти різних типів, починаючи від простих додатків і до складних вебсервісів та мобільних застосунків. До того ж, вона володіє простим та зрозумілим інтерфейсом, великим вибором розширень для розширення функціональності.

Postman – це більше ніж утиліта, це універсальний майданчик, що полегшує роботу розробників та тестувальників на кожному етапі взаємодії з API: від етапів розробки та тестування до документування та інтеграції[19].

MongoDB Compass – це офіційний графічний інтерфейс (GUI) для взаємодії з MongoDB. Розроблений спеціально для зручності розробників, він дозволяє переглядати, змінювати та аналізувати дані, не вдаючись до написання команд у командному рядку[20].

Висновок для першого розділу

У першому розділі кваліфікаційної бакалаврської роботи проведено невеликий огляд актуальних аналогів, використання інформаційних технологій в області дистанційних спортивних тренувань. Розглянуто значення цифрових інструментів у сприянні фізичній активності, формуванні індивідуальних тренувальних планів, а також у забезпеченні інтерактивної взаємодії між тренером та користувачем.

Проведено дослідження та порівняльний аналіз аналогів функціональних можливостей популярних фітнес-платформ: MyFitnessPal, Nike Training Club та Peloton. За результатами проведеного аналізу складено порівняльну таблицю, що

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		18

дозволила визначити як спільні риси, так і відмінності між існуючими рішеннями.

Під час проектування архітектури вебдодатку було обґрунтовано вибір фундаментального стеку проєкту MERN (MongoDB, Express.js, React, Node.js) на мою думку як найбільш відповідної для розробки сучасного, масштабованого та легкого у підтримці та розвитку рішення. Також обґрунтовано вибір інструментальних засобів для створення проєкту.

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		19

РОЗДІЛ 2. ПРОЕКТУВАННЯ ВЕБСАЙТУ ДИСТАНЦІЙНИХ ТРЕНУВАНЬ

2.1 Визначення варіантів використання

Основною метою полягають у створенні зручного та функціонального інструменту для планування тренувань користувачів. Програмний продукт повинен спрощувати процес організації тренувань для користувачів та забезпечувати індивідуальний підхід за допомогою аналізу їхніх характеристик. Опис кожного прецеденту наведено у відповідних табл. 2.1 - 2.18.

Таблиця 2.1

Виявлення варіанту використання реєстрації та входу

Назва варіанту використання	Реєстрація та вхід
Основні актори	Користувач, тренер
Короткий опис	Користувач створює аккаунт або входить у систему через форму авторизації.

Таблиця 2.2

Виявлення варіанту використання перегляду тренувань

Назва варіанту використання	Перегляд тренувань
Основні актори	Користувач
Короткий опис	Отримує список доступних тренувань, фільтрує за категоріями, або може ознайомитись з планами в яких описано більш детальна інформація.

Таблиця 2.3

Виявлення варіанту використання перегляду інформації щодо тренувань

Назва варіанту використання	Перегляд інформації про тренування
Основні актори	Користувач

Продовження таблиці 2.3

Короткий опис	Має можливість переглядати опис таких полів як, цілі, тривалість, рівень складності та інструкції до самих тренувань.
---------------	---

Таблиця 2.4

Виявлення варіанту використання початку тренування

Назва варіанту використання	Початок тренування
Основні актори	Користувач
Короткий опис	Для цього користувач спочатку створює кімнату на сторінці планів тренувань, по власному бажанню користувач надсилає запрошення іншим та проходить його.

Таблиця 2.5

Виявлення варіанту використання збереження прогресу

Назва варіанту використання	Збереження прогресу
Основні актори	Користувач
Короткий опис	Система створює результат та дату проходження після завершення плану тренування, далі дані які зберігає у профілі власника створеної кімнати, ці.

Таблиця 2.6

Виявлення варіанту використання створення постів

Назва варіанту використання	Створення постів
Основні актори	Користувач

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк. 21
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

Продовження таблиці 2.6

Короткий опис	На сторінці новини та інформації постів знаходить кнопку зробити пост, заповнити поля та натиснути кнопку створити пост.
---------------	--

Таблиця 2.7

Виявлення варіанту використання перегляду історії активності

Назва варіанту використання	Перегляд історії активності
Основні актори	Користувач, тренер
Короткий опис	Має змогу бачити свої попередні тренування.

Таблиця 2.8

Виявлення варіанту використання редагування профілю

Назва варіанту використання	Редагування профілю
Основні актори	Користувач, тренер, адмін
Короткий опис	Дає змогу змінювати свої особисті дані облікового запису, тобто його цілі, аватар, пароль та інші налаштування які доступні у сторінці.

Таблиця 2.9

Виявлення варіанту використання вибору особистого тренера

Назва варіанту використання	Вибір особистого тренера
Основні актори	Користувач
Короткий опис	Має змогу подати заявку на особистого тренера для допомоги у подальших тренувань.

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		22

Таблиця 2.10

Виявлення варіанту використання прийняття заявки на особистого тренера

Назва варіанту використання	Прийняття заявок на особистого тренера
Основні актори	Тренер
Короткий опис	Коли користувач надсилає заявку тренеру, тренер отримує у власних повідомленнях заявку яка має інформацією з вибором підтвердити або відхилити.

Таблиця 2.11

Виявлення варіанту використання корегування постів

Назва варіанту використання	Видалення та редагування постів.
Основні актори	Користувач, тренер, адміністратор
Короткий опис	Користувач або тренер можуть видаляти та редагувати свої існуючі пости, але адміністратор може робити теж саме для всіх постів.

Таблиця 2.12

Виявлення варіанту використання аналізу активності користувачу

Назва варіанту використання	Аналіз активності користувачу
Основні актори	Тренер
Короткий опис	Переглядає статистику користувачу скільки тренувань пройдено, які обрано та загальне коло статистики на які м'язи було здійснено навантаження.

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		23

Таблиця 2.13

Виявлення варіанту використання чату

Назва варіанту використання	Чат
Основні актори	Користувач, тренер, адміністратор
Короткий опис	Може отримувати повідомлення від таких же тренерів, користувачів.

Таблиця 2.14

Виявлення варіанту використання зміни ролі

Назва варіанту використання	Зміна ролі.
Основні актори	Користувач, тренер, адміністратор
Короткий опис	Користувач та тренер можуть написати заявку для зміни ролі у налаштувань профілю для зміни ролі. Адміністратор може змінювати роль всім користувачам у їх профілі.

Таблиця 2.15

Виявлення варіанту використання розсилки повідомлень

Назва варіанту використання	Розсилка повідомлень
Основні актори	Адміністратор.
Короткий опис	Має змогу у меню профілю на вкладці розсилки надсилати повідомлення всім як на пошту так і у особисті повідомлення вебдодатку.

Таблиця 2.16

Виявлення варіанту використання прийняття заявок

Назва варіанту використання	Прийняття заявок на пошту
-----------------------------	---------------------------

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		24

Продовження таблиці 2.16

Основні актори	Адміністратор
Короткий опис	Має доступ до корпоративної пошти SportLife в якій може приймати заявки, відповідати на питання, писати листи іншим користувачам та інше.

Таблиця 2.17

Виявлення варіанту використання корегування коментарів

Назва варіанту використання	Написання та видалення коментарів
Основні актори	Користувач
Короткий опис	Користувач додатку має змогу написати коментар на посту, та можливість видалити його.

Таблиця 2.18

Виявлення варіанту використання збереження постів та тренувальних планів

Назва варіанту використання	Збереження постів або тренувальних планів.
Основні актори	Користувач
Короткий опис	Має змогу зберегти тренувальний план або пост в закладки.

Основним учасником взаємодії із системою SportLife виступає пересічний користувач, для якого розроблено інтуїтивний інтерфейс. Він гарантує повноцінну взаємодію з платформою, що призначена для дистанційних спортивних занять. Робота з додатком стартує з етапу реєстрації, в ході якого користувач заповнює ключові особисті дані, зокрема електронну пошту, пароль та загальні відомості про себе. Після успішної реєстрації та авторизації,

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		25

користувачу відкривається доступ до головної сторінки додатку, де зібрано основний функціонал.

Бізнес-вимоги:

Метою проєкту SportLife є створення платформи для дистанційних спортивних тренувань, котра надає користувачам здатність проходити тренування які розроблені командою.

Система повинна забезпечувати доступ до тренувань, реєстрацію користувачів, персоналізацію тренувальних програм, моніторинг прогресу та зворотний зв'язок від тренерів.

Цільова аудиторія проєкту – це люди різних вікових категорій, які бажають займатися фізичною активністю вдома або в зручному для них місці. Вони можуть бути новачками у спорті або досвідченими спортсменами, котрим потрібні персоналізовані тренувальні програми. Також система орієнтована на тренерів, які будуть створювати й адаптувати програми для користувачів.

Вимоги користувачів:

Основними функціональними вимогами є: користувач повинен мати змогу реєструватися в системі, входити в свій акаунт і переглядати тренування, фільтруючи їх за категоріями. Він також може обирати персонального тренера, тренування, створювати пости, мати зв'язок з персональним тренером, ознайомлюватися структурою тренувань, отримувати опис вправ, а також розпочати виконання. Після цього користувач може зафіксувати свій прогрес і завершити тренування. Користувач повинен мати можливість переглядати історію своїх тренувань та аналізувати досягнення. Тренер повинен мати можливість пропонувати персоналізовані програми для користувачів, переглядати активність користувачів і отримувати зворотний зв'язок від них.

Нефункціональні вимоги:

Система має бути побудована на стеку MERN (MongoDB, Express, React, Node.js), задля забезпечення високої продуктивності. Також важливо забезпечити безпеку при зберіганні даних і підтримку інтеграції з платіжними системами для можливості оплати підписок або окремих тренувальних програм.

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		26

Платформа повинна відповідати вимогам щодо захисту персональних даних. Всі дані мають бути шифровані, аби забезпечити їхню конфіденційність та безпеку.

Також інтерфейс додатку повинен бути простим і зручним для користувача, з адаптивним дизайном для зручності користування на різних пристроях. Крім того, система повинна підтримувати систему сповіщень, що нагадує користувачам про заплановані тренування. Діаграма використання зображена на рис. 2.1.

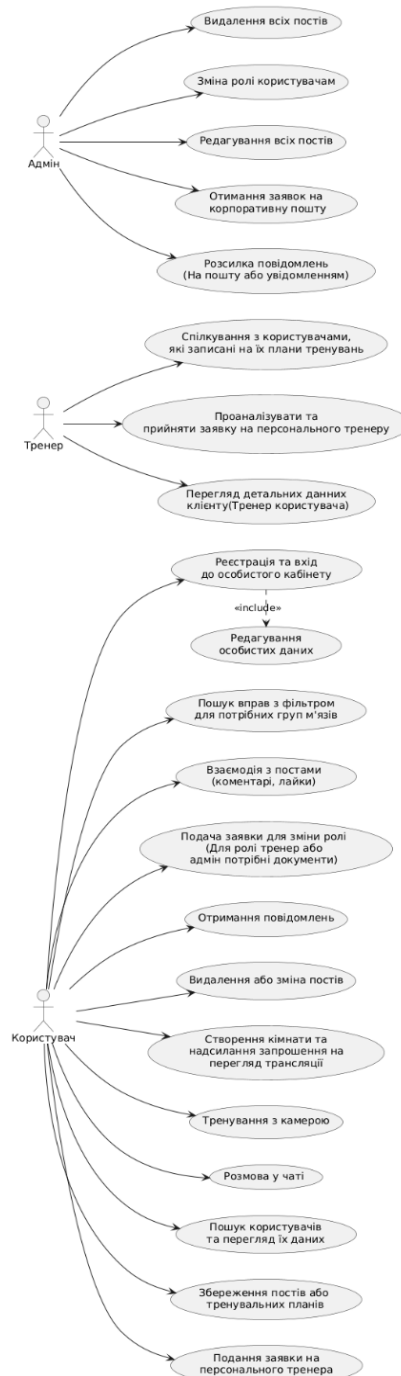


Рис. 2.1 – Діаграма варіантів використання системи

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				27
Змн.	Арк.	№ докум.	Підпис	Дата		

2.2 Розробка бази даних системи.

Для налагодження ефективного зберігання, опрацювання та обміну інформацією в системі SportLife, було обрано MongoDB – документо-орієнтовану базу даних, яка відноситься до NoSQL-рішень. Такий вибір пояснюється гнучкістю при зберіганні як структурованої, так і напівструктурованої інформації, швидкістю доступу до даних та бездоганною інтеграцією з стеком функціональних технологій MERN.

MongoDB передбачає збереження даних у вигляді документів формату BSON, структура яких значною мірою нагадує JSON. Цей підхід спрощує взаємодію між клієнтською стороною (React) і серверною частиною (Node.js/Express), оскільки дані не потребують трудомісткої конвертації при передачі між фронтендом та бекендом.

База даних системи включає десять основних таблиць, які дають змогу забезпечувати зберігання та обробку всіх необхідних даних для коректної роботи застосунку. Наведено структури таблиць бази даних (табл. 2.19 – 2.28)

Основу становить колекція Accounts, що містить інформацію про користувачів системи.

Для реалізації функціоналу обміну повідомленнями між користувачами використовується колекція Chats, яка зберігає дані про активні чати. Усі вправи, що використовуються в тренувальних планах, організовано у структуровану колекцію Exercises.

Система також підтримує сповіщення для користувачів, дані яких зберігаються в Notifications.

Соціальна складова додатку реалізована через колекцію Posts, де зберігаються пости користувачів.

Для обробки взаємодій типу заявок чи запитів призначена колекція Requests.

Проведення дистанційних тренувань передбачає наявність Rooms, тобто віртуальних кімнат, де відбуваються сесії.

Пошта користувачів, які бажають отримувати розсилки, додаються до колекції Subscriptions.

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		28

Дані про тренерів представлено в окремій колекції Trainers, а вся інформація про тренувальні плани організована у Workouts.

Деякі з полів в перелічених таблицях є ідентифікаторами на інші таблиці, виступаючи посиланнями, тобто зв'язки між сутностями таблиць наведено на рис.

2.2.

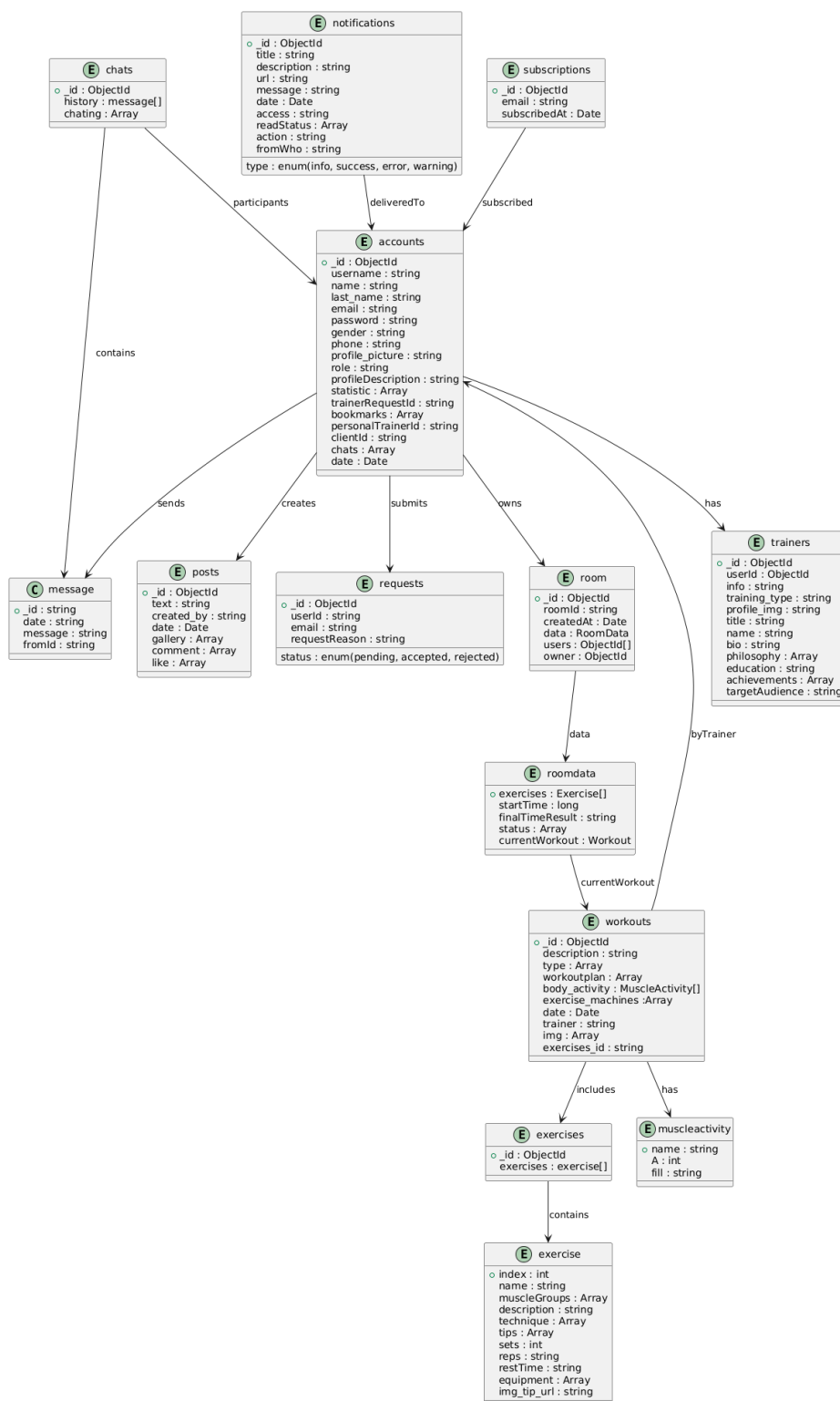


Рис. 2.2 – ER-діаграма

Таблиця 2.19

Табличка користувачів

Accounts		
Поле	Тип	Призначення
_id	ObjectId	Ідентифікатор користувача
username	string	Логін користувача
email	string	Пошта користувача
password	string	Пароль користувачів
date	date	Дата створення користувача
name	string	Ім'я користувача
gender	string	Стать користувача
phone	string	Телефоний номер користувача
profile_picture	string	Картинка профілю користувача
role	string	Роль користувача у вебдодатку
last_name	string	Прізвище користувача
profileDescription	string	Опис профілю
statistic	Array	Весь результат проходжених планів тернувань.
trainerRequestId	String	Ідентифікатор запиту зміни ролі.
bookmarks	Array	Масив збереженого матеріалу.
personalTrainerId	ObjectId	Ідентифікатор запиту на персонального тренера
clientId	string	Ідентифікатор клієнту

Таблиця 2.20

Табличка чатів

Chats		
_id	ObjectId	Ідентифікатор чату
history	Array	Масив повідомлень.

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		30

Продовження таблиці 2.20

chating	Array	Масив двох людей чату.
---------	-------	------------------------

Таблиця 2.21

Табличка вправ

Exercises		
_id	ObjectId	Ідентифікатор вправ
exercises	Array	Масив вправ плану

Таблиця 2.22

Табличка повідомлень

Notifications		
_id	ObjectId	Ідентифікатор повідомлення
title	String	Титульна назва повідомлення
message	String	Текст повідомлення
date	Date	Дата
url	String	Посилання на яке повідомлення дає змогу перейти
access	String	Доступ всім або якомусь конкретному користувачу
readStatus	Array	Масив переглянутих це повідомлення
type	String	Тип повідомлення

Таблиця 2.23

Табличка постів

Posts		
_id	ObjectId	Ідентифікатор постів
text	string	Контент посту
date	Date	Дата створення посту

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		31

Продовження таблиці 2.23

gallery	Array	Масив галереї посту
comment	Array	Масив коментарів
like	Array	Масив лайків
created_by	String	Ідентифікатор хто створив пост

Таблиця 2.24

Табличка заявок

Requests		
_id	ObjectId	Ідентифікатор запиту
status	String	Статус запиту
email	String	Пошта від якої запит
requestReason	String	Причина запиту

Таблиця 2.25

Табличка кімнат

Rooms		
_id	ObjectId	Ідентифікатор кімнати
roomId	string	Спеціальний код кімнати
createdAt	Date	Дата створення
data	Object	Дата даних цієї кімнати
users	Array	Масив всіх користувачів в створеній кімнаті
owner	ObjectId	Власник кімнати

Таблиця 2.26

Табличка пошт розсилки

Subscriptions		
_id	ObjectId	Ідентифікатор підписки

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		32

Продовження таблиці 2.26

email	String	Пошта підписника
subscribedAt	Date	Дата підписника

Таблиця 2.27

Табличка тренерів

Trainers		
_id	ObjectId	Ідентифікатор тренера
bio	string	Короткий опис тренера
training_type	string	Тип його тренувань
profile_img	string	Посилання на картинку профілю
title	string	Заголовок тренера
name	string	Ім'я
userId	ObjectId	Ідентифікатор в колекції аккаунту
philosophy	Array	Масив додаткового тексту
education	string	Освіта
achievements	Array	Масив досягнень тренера
targetAudience	string	Поставлена ціль аудиторії

Таблиця 2.28

Табличка тренувань

Workouts		
_id	ObjectId	Ідентифікатор тренера
title	string	Заголовок плану
description	string	Опис плану
type	Array	Масив типів тренувань, теги
workoutplan	Object	Короткий опис тренування
body_activity	Array	Масив активності, на які м'язи впливає це тренування.

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		33

exercise_machines	Array	Масив можливих інструментів
data	Data	Дата створення плану
img	Array	Масив галереї плану
exercise_id	ObjectId	Ідентифікатор вправ для тренувального плану
trainer	string	Ім'я тренера

MongoDB пасує для розробки сервісів, що працюють в режимі реального часу, систем аналітики чи вебдодатки, де визначальним є швидкий доступ до даних без зайвих складнощів. Її адаптивність та простота дає змогу додавати нові функції до проєкту, не вдаючись до масштабної перебудови структури бази даних.

2.3 Проектування та реалізація алгоритмів роботи системи

Під час розробки інформаційної системи SportLife особлива увага приділялася побудові логіки взаємодії між користувачами, тренерами та основними модулями системи. На цьому етапі було створено набір алгоритмів, які визначають, як саме відбуваються ключові процеси – реєстрація користувачів, редагування даних профілю, створення заявки на зміну ролі, вибір тренера тощо.

Безпосередньо ж вебдодаток можливо умовно розділити на певні функціональні модулі, кожен з яких відповідає за окрему частину логіки системи. Для наочності була сконструйована діаграма ключових складових частин системи – діаграма компонентів. Вона відображає взаємозв'язки між основними модулями, такими як модулі аутентифікації, тренувань, взаємодії з тренером, чату, управління сповіщеннями та інші. Подібне компонентне подання дозволяє глибше зрозуміти внутрішню організацію додатку та логіку взаємодії між його частинами наведено на рис. 2.3.

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		34

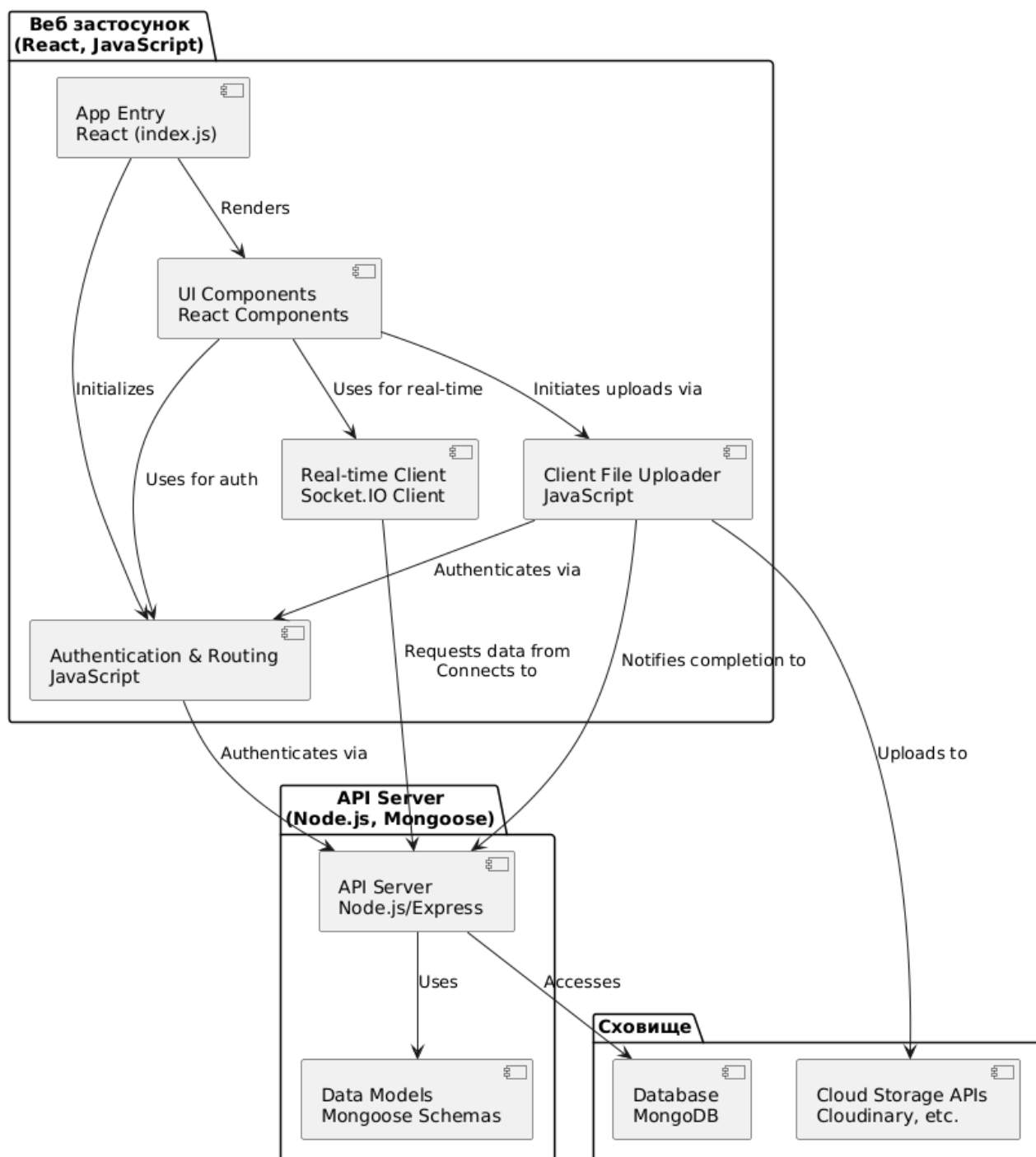


Рис. 2.3 – Діаграма компонентів додатку

Усі алгоритми в системі реалізовані за допомогою Node.js. Як серверний фреймворк використовується Express.js, що дозволяє легко обробляти запити й керувати маршрутами.

Для зберігання даних використовується база даних MongoDB. Робота з нею спрощується завдяки бібліотеці Mongoose, яка допомагає створювати зручні структури даних і легко працювати з колекціями на сервері.

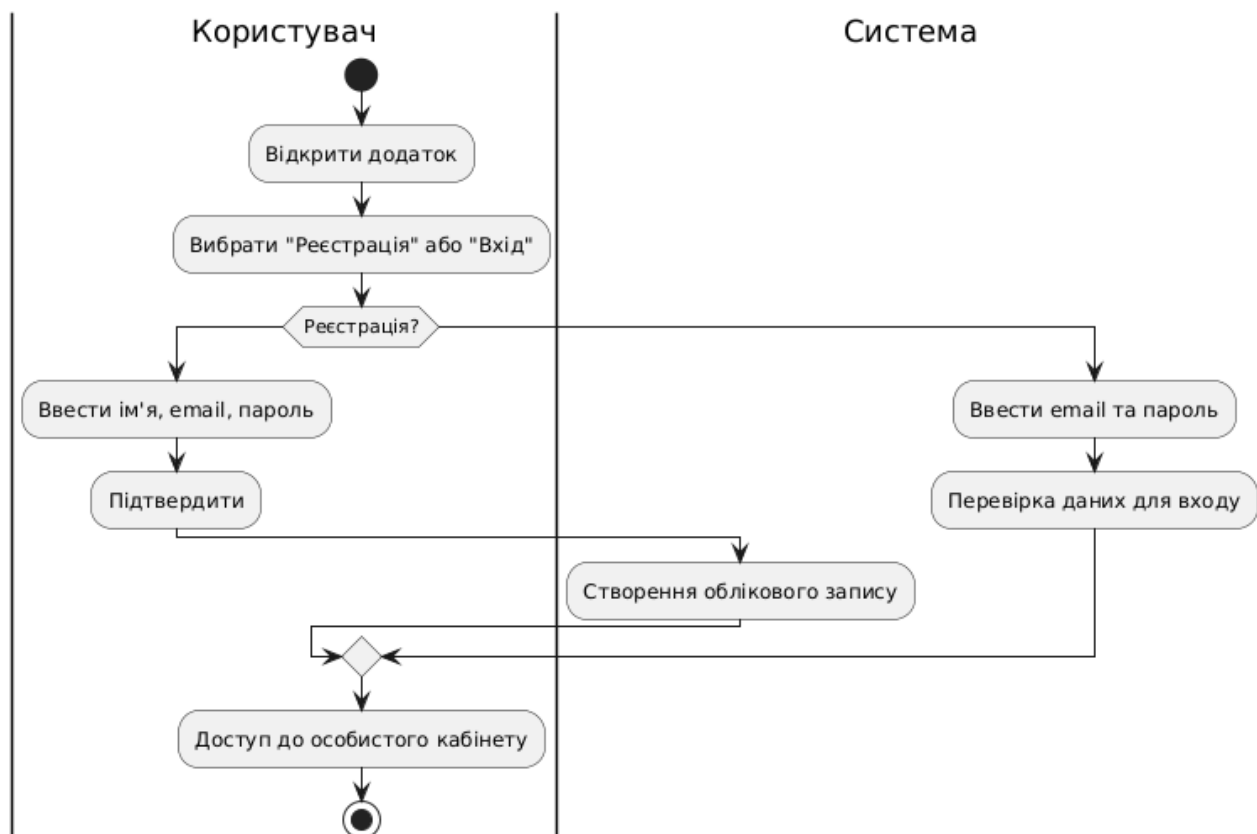


Рис. 2.4 – Діаграма активності для авторизації

Як показано на діаграмі рис. 2.4, для початку користувач відкриває додаток і переходить до форми авторизації.

Він може обрати один із варіантів «Вхід в систему» чи «Реєстрація».

Під час спроби входу в систему користувач побачить сторінку з формою для введення облікових даних.

У разі невірно введених даних, користувач залишиться на тій самій сторінці для повторного введення логіну та паролю. Якщо введені дані є коректними, система надасть доступ до функціоналу відповідно до ролі користувача.

У разі вибору опції «Реєстрація», відобразиться форма, в якій користувач зобов'язаний заповнити всі обов'язкові поля для успішного проходження валідації. Після валідації на стороні клієнта, дані надсилаються на сервер, де здійснюється їх ретельна перевірка, зокрема на предмет дублювання або інших помилок.

Наступна діаграма активності на рис. 2.5 ілюструє процес взаємодії користувача із системою під час оновлення особистих даних, зміни ролі та створення заявки на особистого тренера які можна зробити відразу після авторизації в систему.

Користувач входить до особистого кабінету, з лівого боку екрану вибирає функцію редагування даних і вводить нову інформацію, наприклад, ім'я, email, телефон чи інші данні користувача. Система обробляє ці дані, оновлює їх і зберігає зміни, після чого надсилає користувачу повідомлення про успішне оновлення.

Далі користувач має змогу змінити свій профіль, скажімо, на "Тренер". У випадку прийняття такого рішення, спершу необхідно пройти процедуру аутентифікації. Користувачу потрібно підготувати обґрунтування, вибрати потрібну роль, долучити офіційні документи, що ідентифікують його особу, і надіслати запит на зміну. По завершенні, потрібно дочекатися підтвердження від корпоративної електронної пошти, або перевіряти статус заявки. У разі позитивного результату, система оновлює роль, зберігає зміни та відправляє сповіщення про успішне оновлення. Коли ж користувач бажає зберегти попередню роль, система просто зупиняє процес без жодних подальших дій.

Якщо користувач має намір записатися до особистого тренера, йому слід у лівому меню відшукати розділ "Особистий тренер", де висвітлено всю актуальну інформацію стосовно персонального тренера. Далі необхідно клікнути на кнопку "Надіслати запрошення", заповнити поле "Примітка" та обрати бажаного тренера зі списку доступних.

Після цього потрібно буде зачекати, поки тренер схвалить заявку. У випадку підтвердження – відобразиться інформація про тренера, а статус заявки зміниться на "підтверджено".

Діаграма демонструє чітку послідовність дій та рішення, які приймаються на кожному етапі взаємодії.

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		37

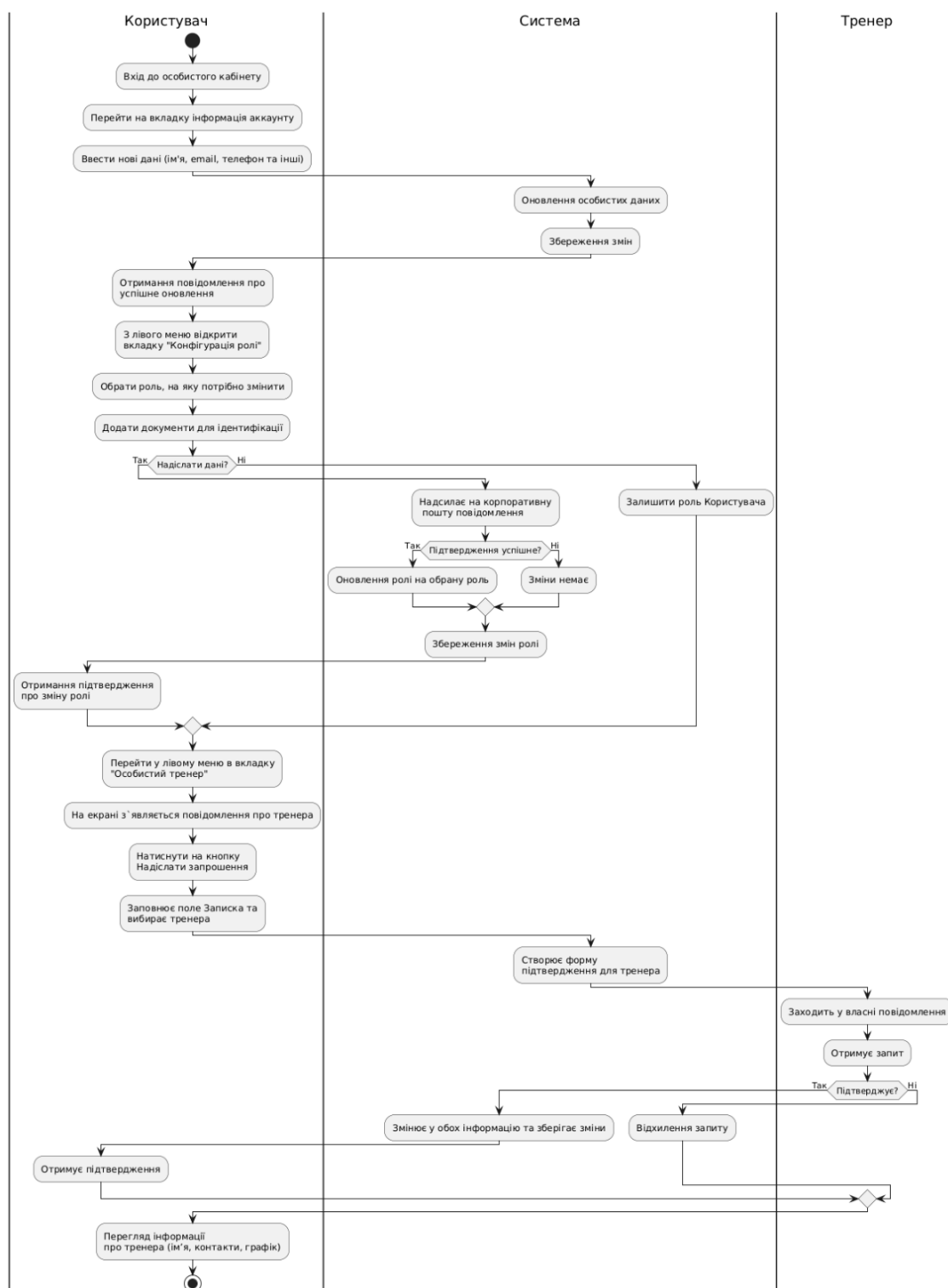


Рис. 2.5 – Діаграма активності для редагування власних даних

2.4 Реалізація функціоналу системи тренувального вебдодатку.

Функціонал вебдодатку для тренувань сконструйовано, щоб забезпечити результативну взаємодію тренера та користувача на відстані. Головне завдання системи – створити комфортне місце для планування, реалізації та оцінювання процесу тренувань з можливістю тренуватися разом.

Тож давайте розглянемо логіку одного з модулів системи, а саме модуля початку тренувань з тренувального плану. Робота з модулем починається з створення кімнати тренувань.

Для початку реалізації було використано бібліотеку Socket.IO на сервері та Socket.IO-client на боці клієнта для створення функціоналу тренувальних кімнат, до котрих можуть приєднуватися декілька користувачів. Кожна кімната формується динамічно, відповідно до сесії тренування. Користувачі, котрі приєдналися, можуть бачити оновлення та саму трансляцію камери, які надходять без перезавантаження сторінки. Завдяки цьому створюється ефект онлайн-присутності, що особливо важливе для мотивації під час дистанційного тренування. Передбачено також обмеження доступу до кімнат.

Приєднатися можуть тільки авторизовані користувачі, що гарантує безпеку і контроль над сесіями. У разі спроби доступу стороннього користувача сервер блокує підключення, і клієнт перенаправляється на сторінку з відповідним повідомленням. Socket.IO також дозволяє підтримувати постійне з'єднання між клієнтом і сервером, що робить обмін даними стабільним і швидким навіть за значного навантаження.

Для створення кімнати застосовується подія joinRoom, яка спрацьовує на стороні клієнта, коли користувач вирішує увійти до певної кімнати.

Кожна кімната ідентифікується унікальним номером і асоціюється з користувачами, котрі до неї приєднались. Приклад коду наведено в додатку В.

На сервері, після отримання події joinRoom, сокет приєднується до кімнати. Наступним етапом є перевірка на наявність відповідного запису у базі даних. Якщо кімната ще не існує, створюється новий об'єкт з інформацією про власника, список учасників і початкові дані. У разі, якщо кімната вже існує, але користувач ще не був її частиною, він додається до переліку учасників.

В обох випадках, усім клієнтам, що перебувають у кімнаті, надсилається оновлений список учасників (chatHistory), інформація про власника (roomOwner) та актуальний стан збережених даних (receiveData). До того ж, в момент створення кімнати викликається функція для визначення загальної кількості

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		39

користувачів в кімнаті, що розсилається усім підключеним.

Далі для опису візьмемо приклад коду наведений у додатку Г для оновлення даних які надсилає хост користувачам.

Коли надходить подія `updateData`, система відшукує відповідну кімнату за її ідентифікатором. Якщо кімната не знайдена або бракує потрібних даних, обробка припиняється.

Крім того, перевіряється, чи є користувач тим, хто запустив подію (власником кімнати). Це забезпечує, що тільки власник кімнати має право вносити зміни до даних поточного тренування.

Наступним візьмемо приклад коду з додатку Г, коли користувач залишає кімнату або добровільно від'єднується. Ця логіка втілена через подію `disconnectData`, котра передається на сервер при втраті з'єднання або навмисному виході з кімнати.

Сервер обробляє подію, отримуючи ідентифікатори кімнати та користувача. Якщо кімната не існує, подальші дії не виконуються. У разі, якщо користувач є власником кімнати (тобто тренером або ініціатором сесії), система автоматично видаляє всю кімнату з бази даних, сповіщає інших користувачів про її закриття через подію `roomClosed` та завершує обробку.

Також розглянемо останній приклад коду з додатку Д, який завантажує файли які не знаходяться на сервері у хмарне сховище Cloudinary. Це потрібно, зокрема, для аватарів в профілях, фотографій з тренувань або медіа, які додаються до постів.

Для здійснення цього розроблено окрему універсальну асинхронну функцію. Вона приймає на вхідлюбий файл, потім формує спеціальний об'єкт `FormData`. До нього додаються необхідні метадані, після цього виконується HTTP-запит методом POST на стандартний API Cloudinary для завантаження зображень.

У разі успішного завершення операції, функція повертає пряме посилання (`secure_url`) на щойно завантажене зображення яке можна використовувати будь де.

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		40

Висновок другого розділу

У другому розділі було розглянуто більш детально основний технологічний стек MERN.

На базі отриманих даних було проаналізовано роль кожного з компонентів у побудові сучасного вебдодатку для дистанційних тренувань, а також описано принципи їх взаємодії.

Зокрема, було підкреслено переваги використання MongoDB як гнучкої документно-орієнтованої бази даних, Express.js і Node.js для створення надійного серверного API, а також React для побудови інтерфейсу користувача з можливістю реактивної роботи в реальному часі.

Також додатково було спроектовано базу даних, проаналізовано кожну створену колекцію, яку було проаналізовано та створено зв'язки між сутностями та представлені як діаграма класів також було проаналізовано діаграму компонентів додатку.

Була продемонстрована також невелика частина коду з основного проєкту, де було розібрано переваги використання асинхронних функцій та обробки подій, які відбуваються в режимі реального часу, із застосуванням Socket.IO.

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		41

РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ВЕБДОДАТКОМ

3.1 Порядок встановлення та налаштування параметрів системи.

Для розробки інтерфейсу використано клієнтську бібліотеку React, що базується на технологічному стеку MERN. Це дозволило застосувати динамічний, компонентний підхід у створенні інтерфейсу. Внаслідок цього було досягнуто зручної навігації, можливості швидко оновлювати дані, не перезавантажуючи сторінку, та підвищеної інтерактивності в процесі взаємодії користувача з програмою.

React забезпечила на себе підтримку маршрутизації, керування станом додатку, а також інтеграцію в реальному часі з використанням socket-підключень.

Для старту роботи з системою, знадобиться здійснити декілька легких дій, інсталювати безпосередньо сам проєкт, встановити всі компоненти, які від нього залежать, а також запустити його з терміналу чи будь-якого іншого редактора коду. Після цього система буде повністю підготовлена до роботи.

Якщо виникне потреба, можна додатково налаштувати середовище (тобто змінні середовища, з'єднання з базою даних або ключі API), проте для базового запуску ці дії здебільшого не є обов'язкові.

Далі представимо діаграму розгортання, яка функціонує як візуальне представлення конфігурації обчислювальних вузлів та також складових системи під час розгортання.

Наступний рис. 3.1 зображує приклад діаграми розгортання, яка демонструє взаємне розташування вузлів та компонентів системи. На ній показано, як саме елементи програми розміщуються у середовищі, де вона працює. Зокрема, видно розподіл між клієнтською частиною, яка виконується у браузері користувача, та серверною частиною, що працює на окремому сервері. Також відображено взаємозв'язки із зовнішніми сервісами, такими як база даних чи служби сповіщень.

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		42

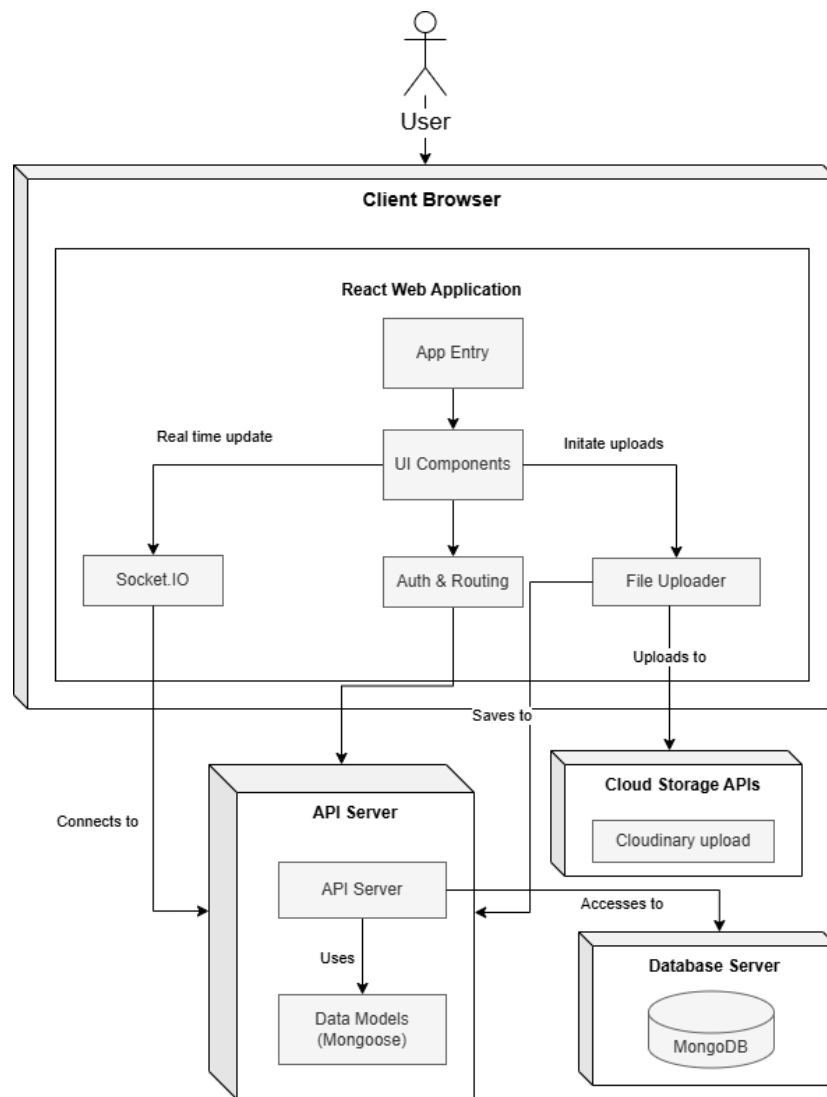


Рис. 3.1 – Діаграма розгортання

3.2 Структура інтерфейсу тренувального вебдодатку.

Веб застосунок розпочинається з головної сторінки, з якої можна переходити до інших розділів. На головній сторінці користувач стикається з стислим вступом до вебдодатку, який відкриває перед ним ключові функції та доступні можливості.

Можна відокремити на головній сторінці меню навігації для швидкого переміщення по блокам в яких розміщено декілька свіжих публікацій, також, на цій сторінці знаходяться картки з інформацією про тренерів, які містять фотографії та короткі біографії, що сприяє швидкому ознайомленню з командою професіоналів. В нижній частині сторінки передбачене поле для підписки на розсилку, де користувач може поділитися своєю електронною поштою, аби

отримувати свіжі новини та ексклюзивні пропозиції напряму на свою електронну пошту (рис. 3.2 – 3.5).

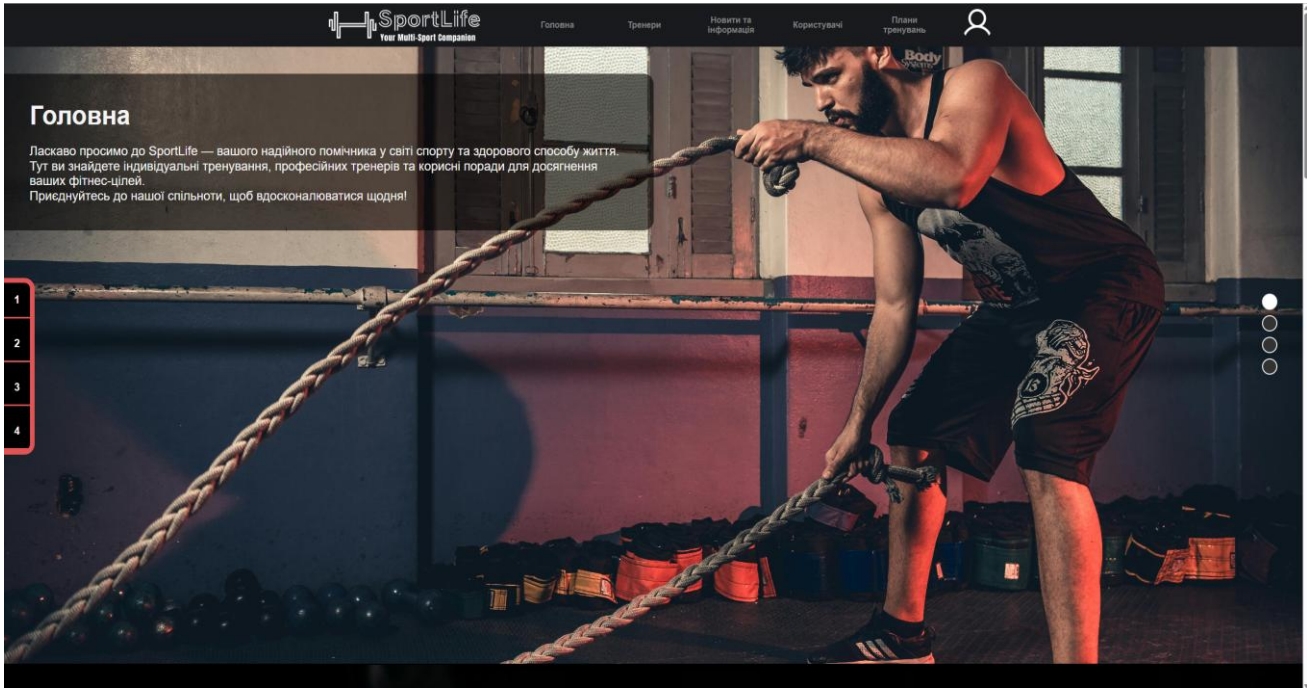


Рис. 3.2 – Головна сторінка

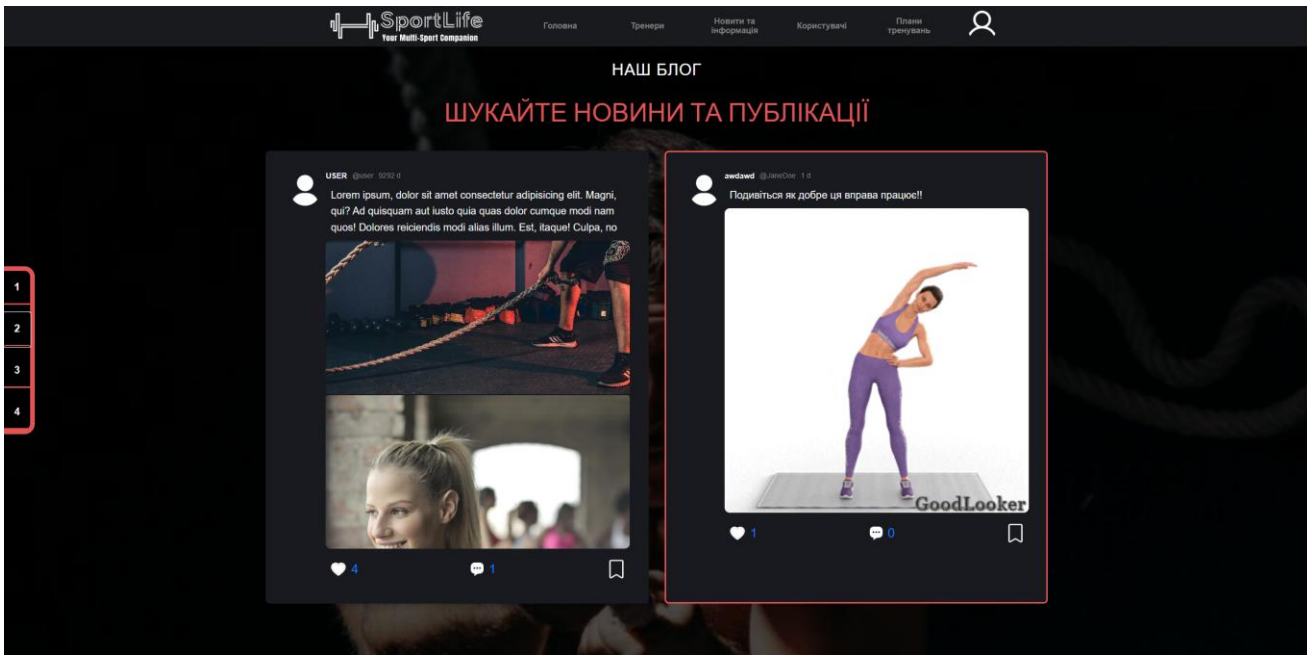


Рис. 3.3 – Головна сторінка - свіжі публікації

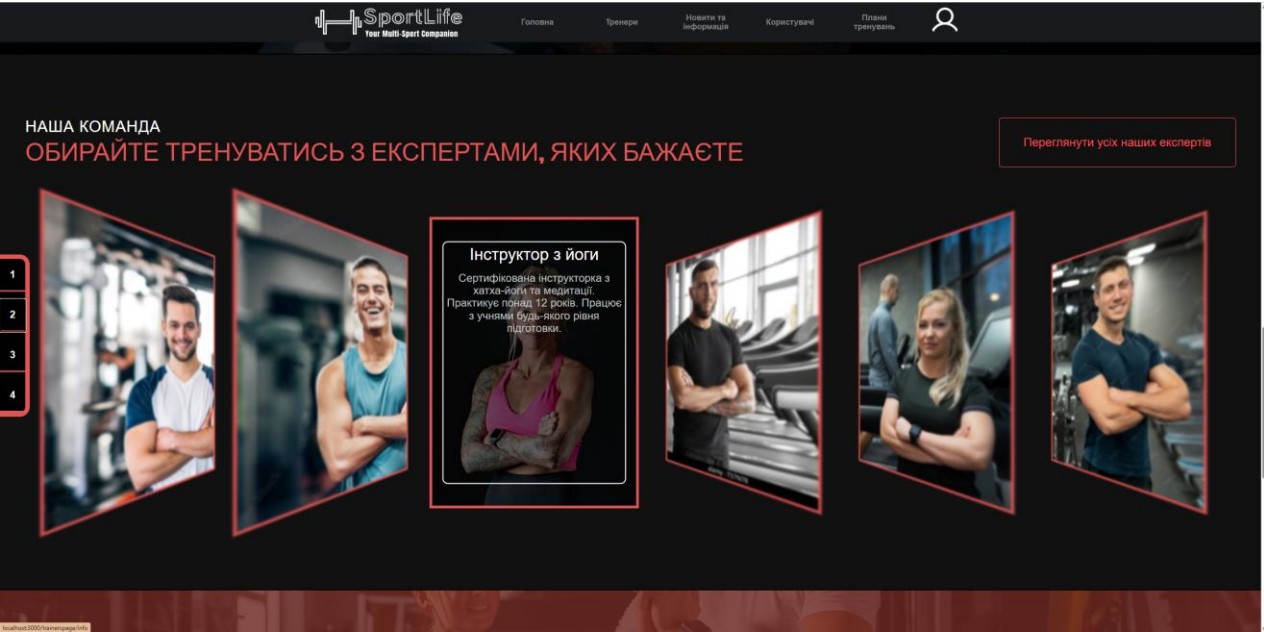


Рис. 3.4 – Головна сторінка – команда тренерів

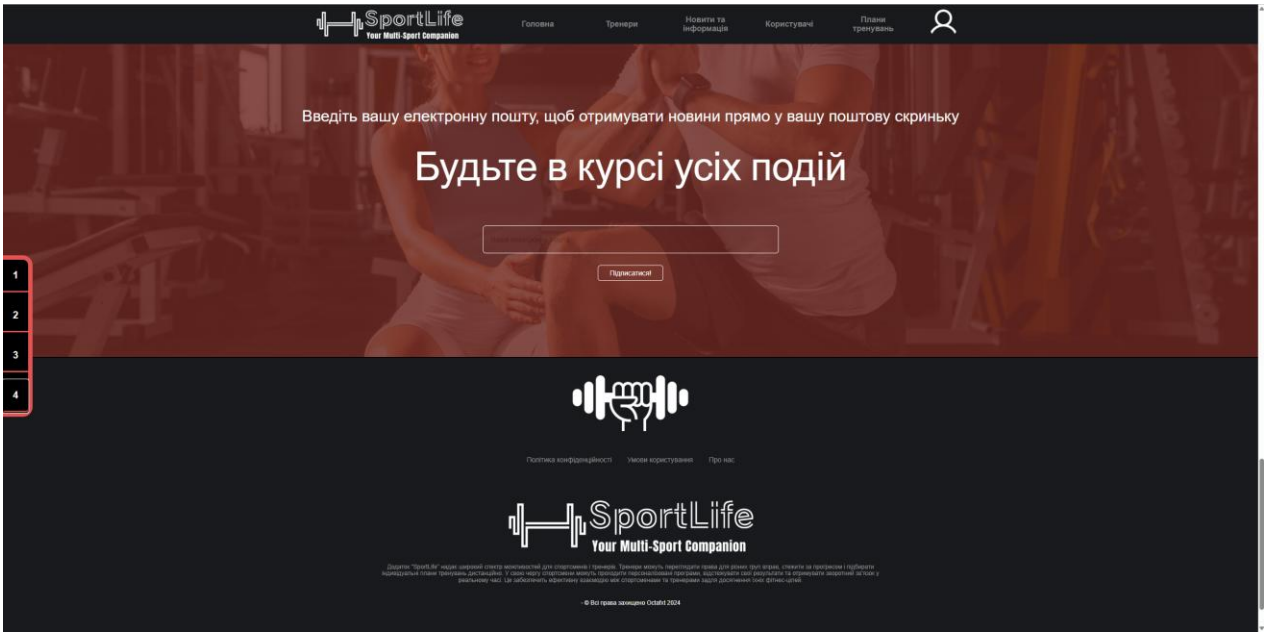


Рис. 3.5 – Головна сторінка – форма пошти для розсилки

Окрім головної сторінки користувачу дається змогу на перехід по сторінкам тренери – основна інформація команди тренерів, новини та інформація – сторінка в якій знаходяться загальнодоступна інформація людей які діляться щодня чимось новим, на сторінці користувачі можна знайти користувачів зареєстрованих на сайті та подивитися невелику їхню інформацію на сторінці планів тренувань можна передивитись лише інформацію планів, створити кімнату не зареєстровані

користувачі не мають змогу, як і інші дії. Для того, щоб користувач мав змогу повноцінно користуватися платформою, потрібна авторизація (рис. 3.6).

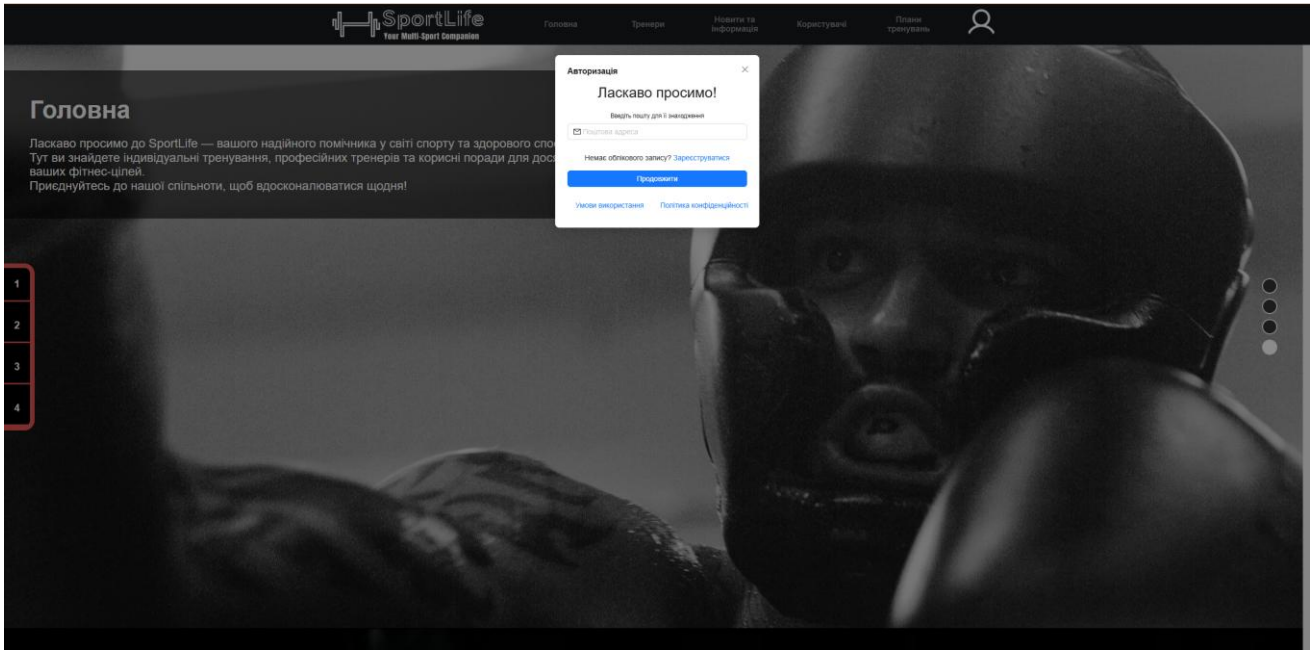


Рис. 3.6 – Модальне вікно авторизації перевірки пошти

Тут нам дають два варіанти або відразу перейти на форму авторизації, або спочатку вписати пошту яку модальне вікно перевірить на наявність у базі. Якщо пошта буде знайдено тоді виведеться відповідне повідомлення також при невдалому знаходженні (рис. 3.7).

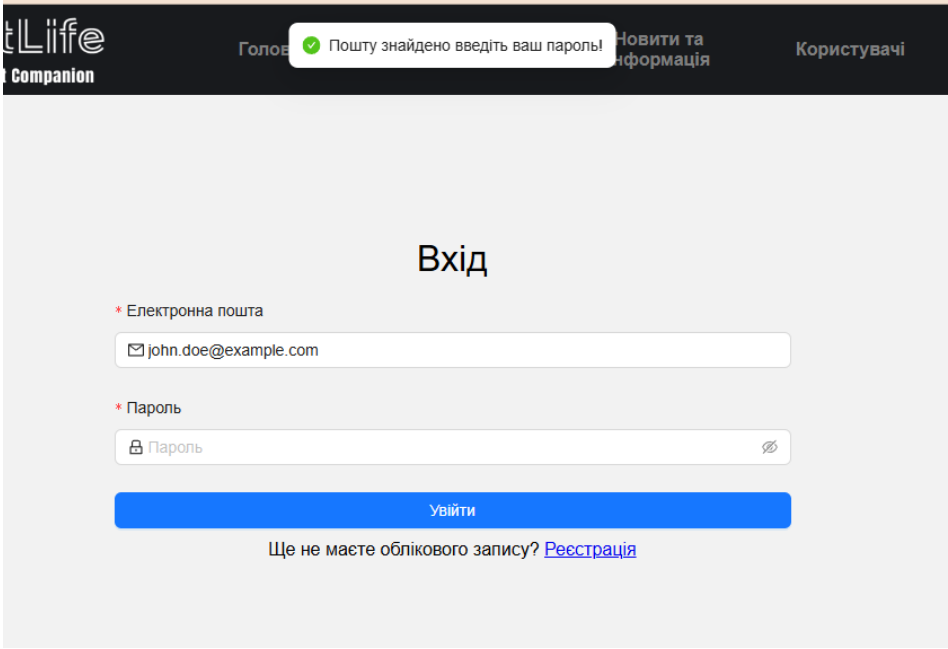
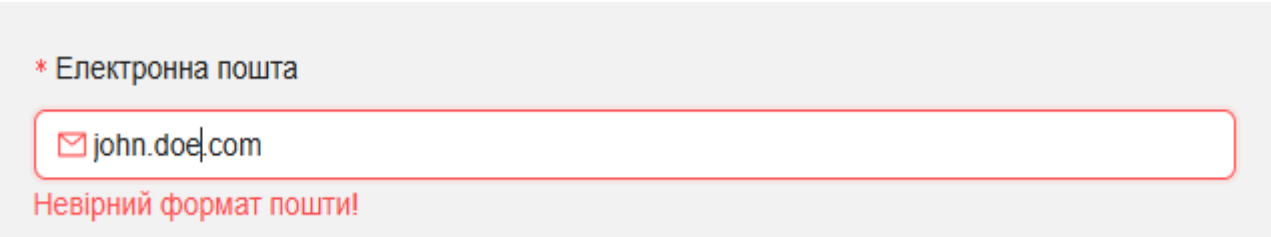


Рис. 3.7 – Форма авторизації знайденої пошти

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		46

На сторінці було виявлено пошту та додано у форму авторизації автоматично з відповідним повідомлення. Наступним кроком потрібно перевірити валідацію даних, перевіряти будемо на правильний формат пошти або чи знайдена пошта в базі чи ні (рис. 3.8 – 3.9).

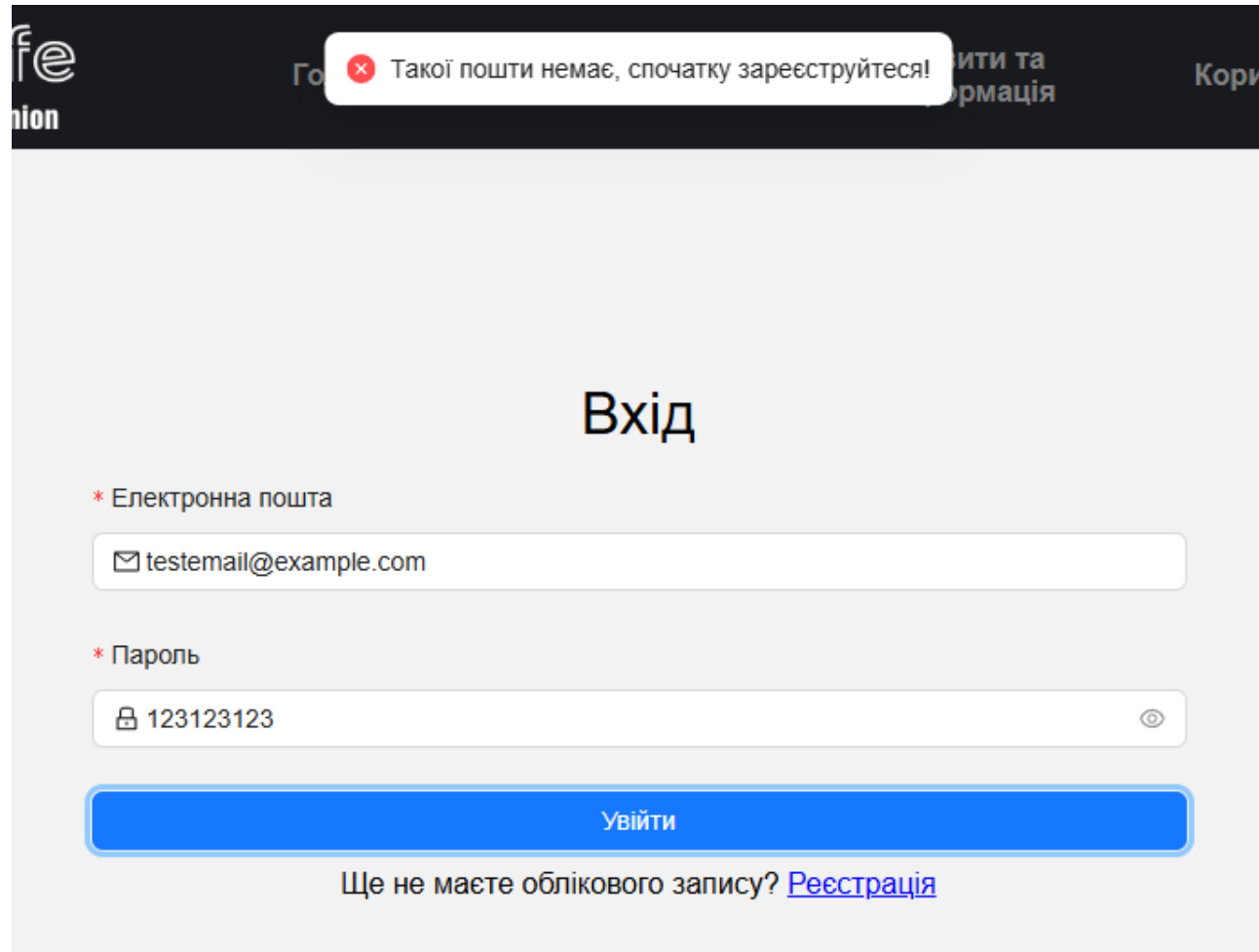


* Електронна пошта

✉ john.doe.com

Невірний формат пошти!

Рис. 3.8 – Помилка про неправильний формат пошти



Go ✖ Такої пошти немає, спочатку зареєструйтеся! [Вийти та інформація](#) [Користувач](#)

Вхід

* Електронна пошта

✉ testemail@example.com

* Пароль

🔒 123123123 [👁](#)

[Увійти](#)

Ще не маєте облікового запису? [Реєстрація](#)

Рис. 3.9 – Користувача з вказаною електронною адресою не знайдено.

Якщо натиснути на кнопку “Реєстрація” тоді форма авторизації зміниться на форму реєстрації в якій додаються додаткові поля (рис. 3.10).

Рис. 3.10 – Форма авторизації - реєстрація

На цій сторінці користувач має можливість створити новий обліковий запис, якщо його ще немає. Щоб це зробити, необхідно вказати всі обов'язкові відомості: ім'я, прізвище, електронну адресу, пароль та інші. Після заповнення всіх полів користувач натискає кнопку «Зареєструватися», щоб завершити процес створення акаунту. Але у нашому випадку будемо використовувати форму авторизації (рис. 3.11).

Рис. 3.11 – Форма авторизації – вхід

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		48

Також як і форма реєстрації форма авторизації входу забезпечує валідацію даних форми на пусті дані та інші помилки які були проаналізовані раніше повертаючи відповідну помилку (рис. 3.12).

Вхід

* Електронна пошта

✉ Електронна пошта

Введіть електронну пошту!

* Пароль

🔒 Пароль

Введіть пароль!

Увійти

Ще не маєте облікового запису? [Реєстрація](#)

Рис. 3.12 – Форма авторизація – помилки входу

Щоб продовжити авторизацію вводимо тестові дані для тестового облікового запису (рис. 3.13).

Вхід

* Електронна пошта

✉ john.doe@example.com

* Пароль

🔒 password123

Увійти

Ще не маєте облікового запису? [Реєстрація](#)

Рис. 3.13 – Форма авторизація – заповнення даних входу

Після успішної авторизації, можна помітити на головній сторінці вебдодатку деякі зміни.

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		49

У правому нижньому куті буде зображено іконку чатів (рис. 3.14). Нагорі, в навігаційній панелі, можна помітити іконку облікового запису. Натиснувши на них, відкриється випадаючі меню, де ви зможете або перейти до свого профілю, або вийти з облікового запису (рис. 3.15). Поряд з кнопкою облікового запису, буде також кнопка повідомлень, яка відкриє випадаючі меню зліва яке міститиме всі попередні сповіщення (рис. 3.16).

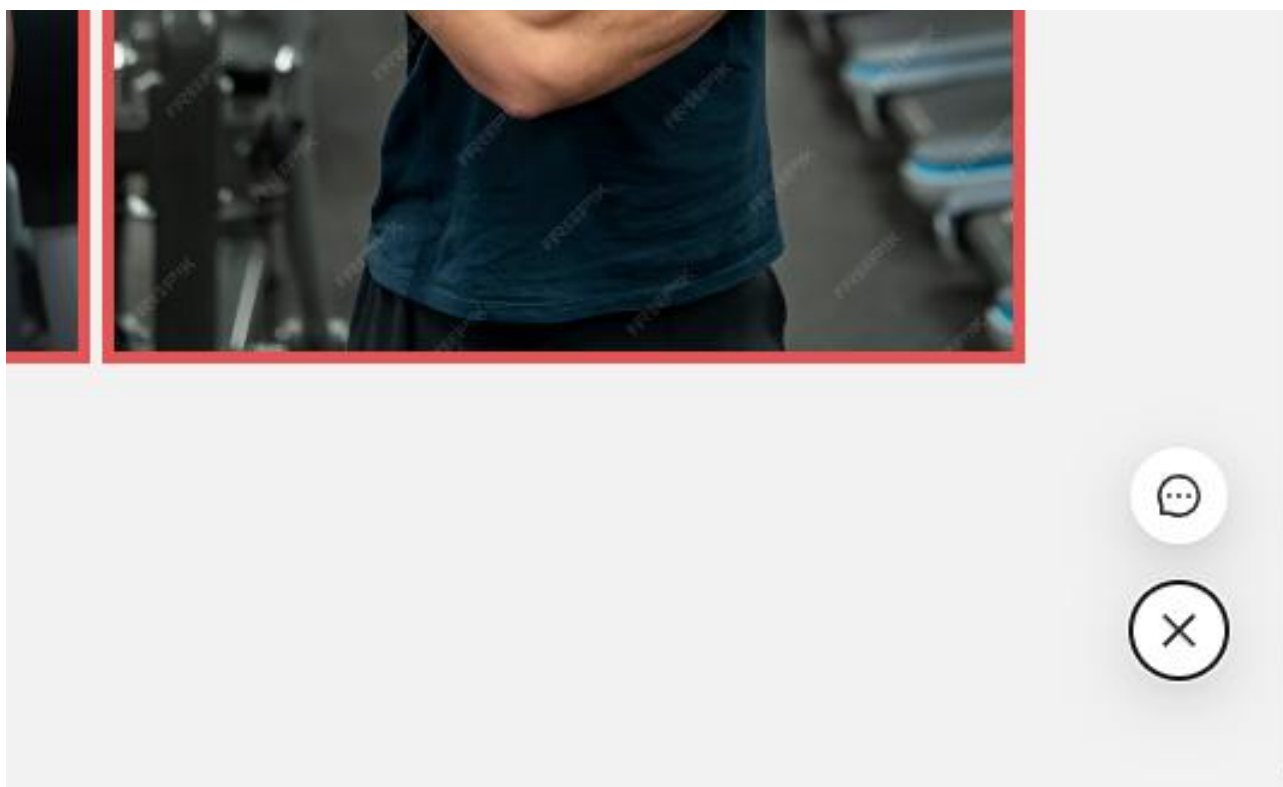


Рис. 3.14 – Кнопка з елементом чату

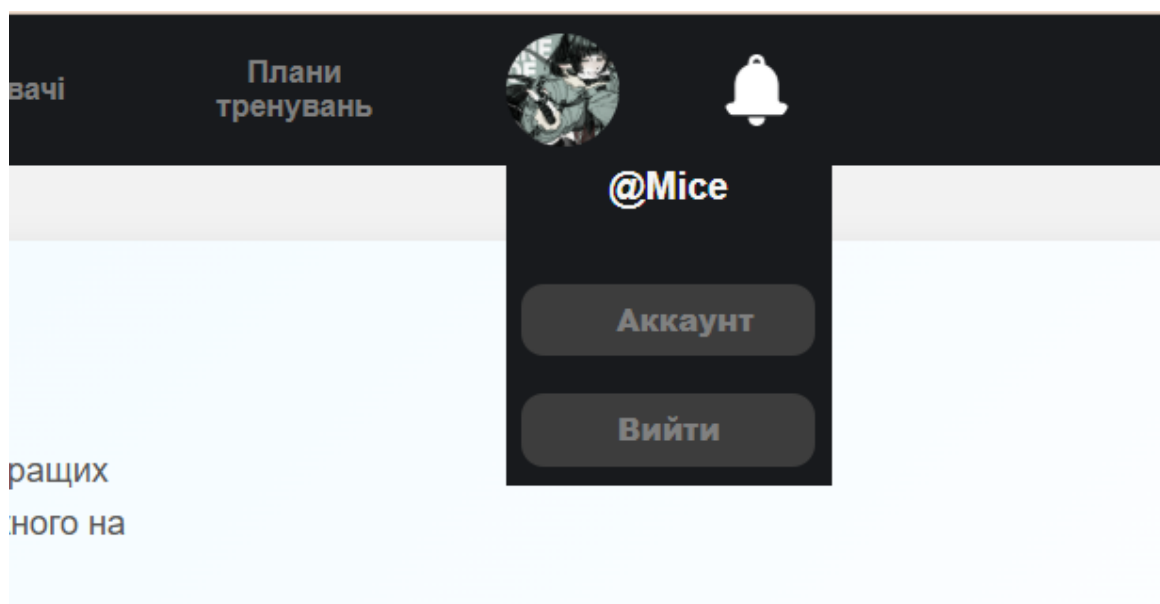


Рис. 3.15 – Випадаючі меню профілю

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		50

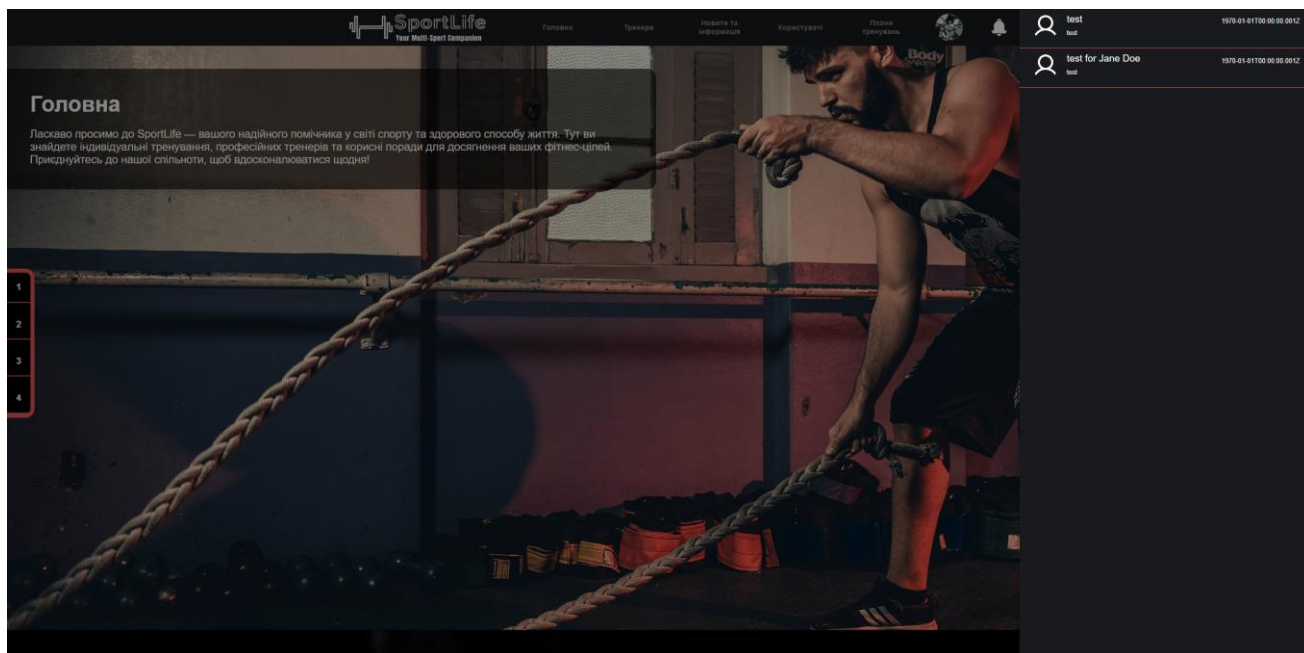


Рис. 3.16 – Меню повідомлень користувачів

Повернімося до (рис. 3.15), де ми помітимо два розділи: “Аккаунт” та “Вихід”. При натисканні на “Аккаунт”, користувача буде переадресовано на сторінку для оновлення інформації про обліковий запис. Там можна змінювати особисті дані: ім'я, адресу електронної пошти, пароль. Крім цього, передбачені такі опції як запит на особистого тренера, перегляду історії виконаних вправ, аналізу статистики фізичної активності, і, за потреби, налаштування ролі в системі. Також доступна можливість перегляду збережених постів чи обраних тренувальних планів. Спочатку детально розглянемо інтерфейс сторінки редагування даних та її можливості.

Рис. 3.17 – Сторінка редагування профілю

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				51
Змн.	Арк.	№ докум.	Підпис	Дата		

На сторінці “Аккаунт” (рис. 3.17) користувач бачить вказану при реєстрації дані, яку він може змінювати. Коли він вносить зміни і тисне кнопку “Зберегти”, дані у профілі оновлюються – система зберігає введені користувачем дані (рис. 3.18 – 3.19).

Опис профілю

Оновлена інформація профілю!

Зберегти

Рис. 3.18 – Приклад оновлення з полем “Опис профілю”

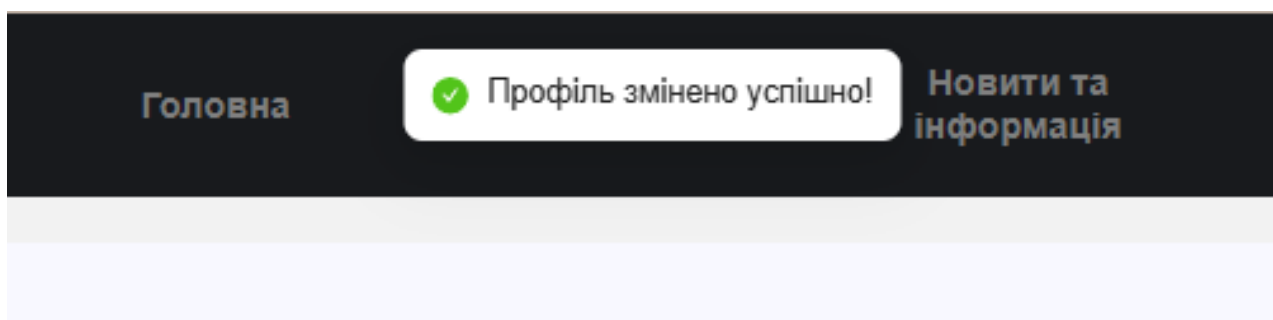


Рис. 3.19 – Повідомлення про успішне оновлення

Також на цій сторінці є місце для зміни паролю (рис. 3.20). Там передбачена спеціальна форма, де треба вказати старий пароль, а потім два рази – новий пароль. Якщо після натискання кнопки “Змінити пароль” все заповнено правильно і система проходить перевірку, пароль буде замінено (рис. 3.21).

Зміна пароля

* Старий пароль

* Новий пароль

* Підтвердження нового пароля

Змінити пароль

Рис. 3.20 – Форма для зміни паролю

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		52

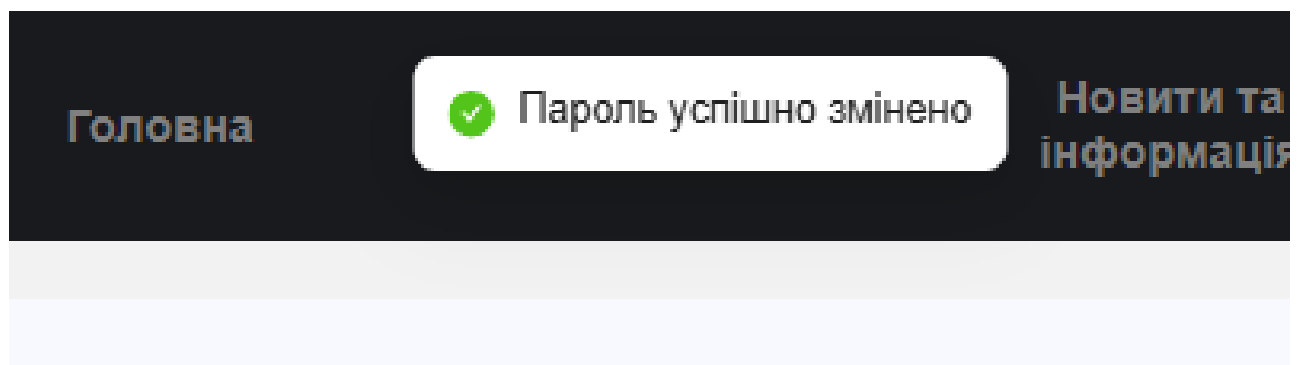


Рис. 3.21 – Повідомлення про успішну зміну паролю

Ще як можна замітити на цій формі передбачено редагування фотографії профілю (рис. 3.22). Якщо натиснути на кнопку “Змінити фото профілю” тоді користувачу відкриється меню для вибору нової фотографії профілю. Після вибору користувачу назва нової картинки з’являється знизу від самої кнопки (рис. 3.23), щоб зберегти дію потрібно натиснути кнопку “Зберегти”, і тоді дані оновляться де вийде відповідне повідомлення (рис. 3.19). Якщо користувачу не потрібно фотографія тоді йому всього лише потрібно натиснути кнопку “Видалити” з’явиться звичайна фотографія (рис. 3.24) і також після цього натиснути “Зберегти”, щоб дія збереглися у базі даних.

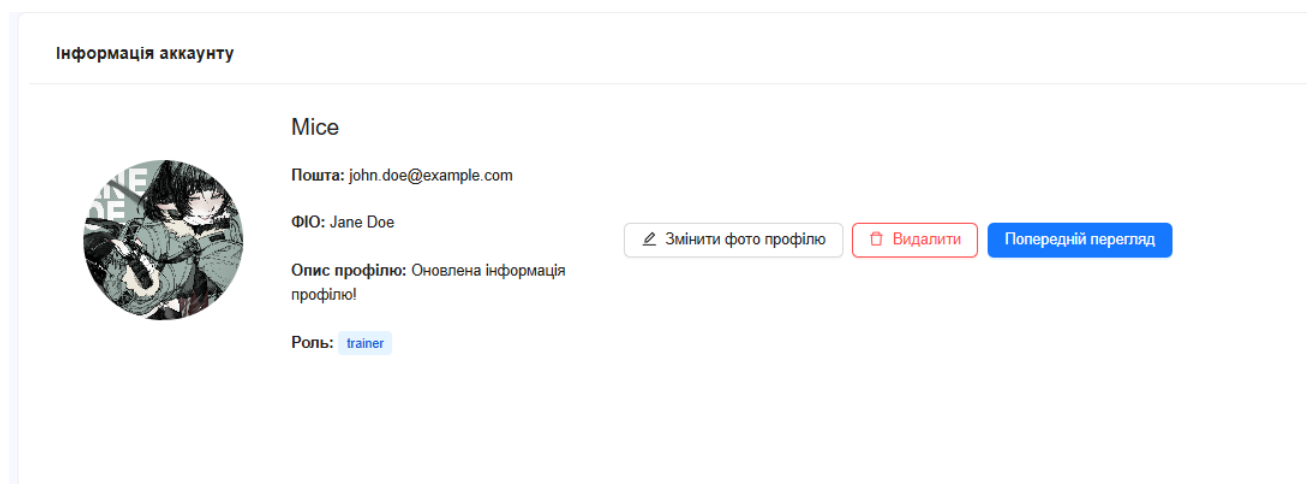


Рис. 3.22 – Форма редагування фотографії профілю

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		53

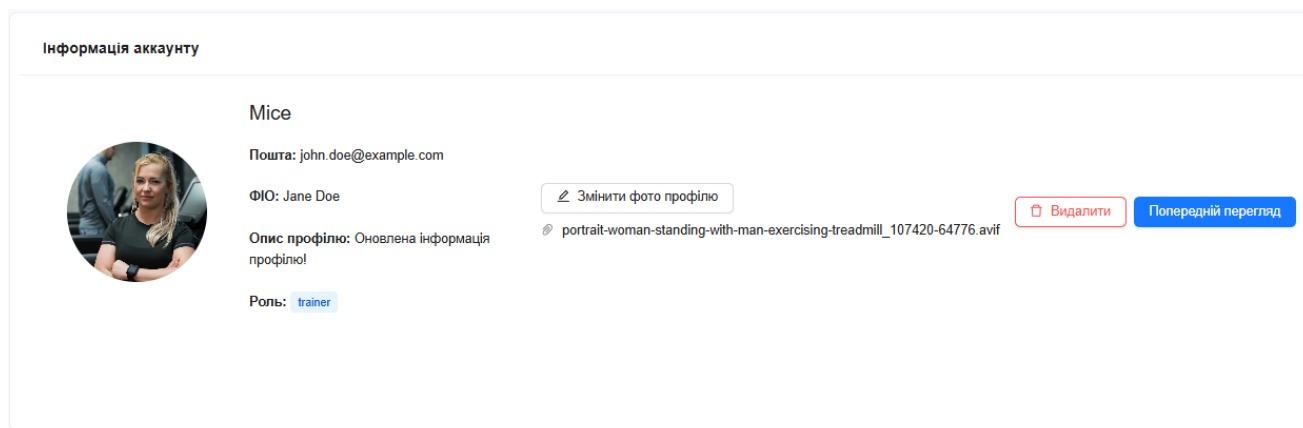


Рис. 3.23 – Приклад зміни фотографії

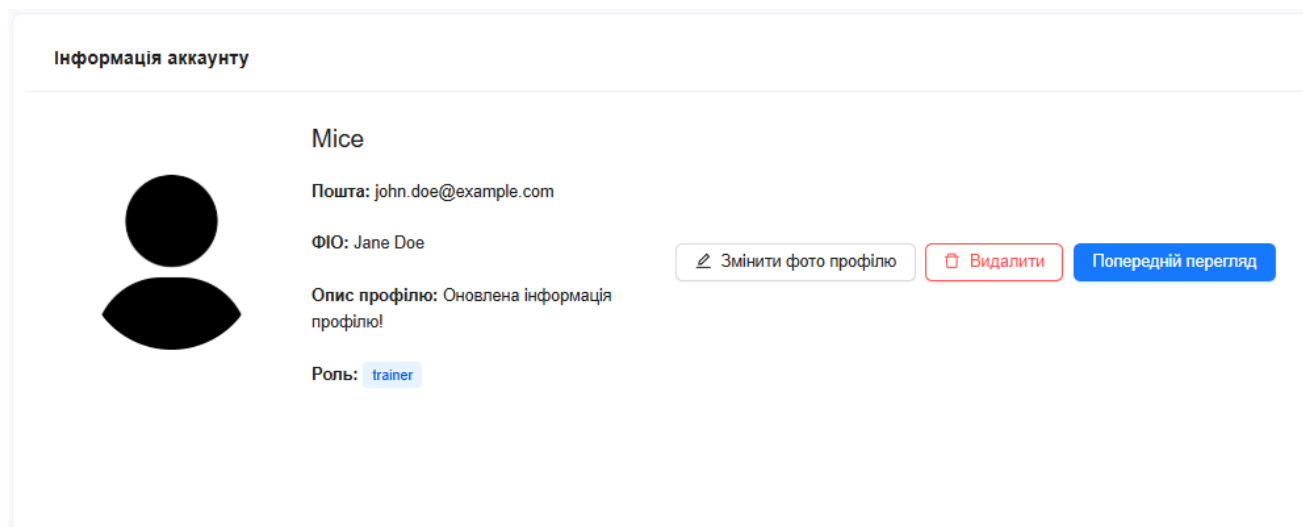


Рис. 3.24 – Приклад видаленої фотографії

Також на цій ж формі є змога перевірити завантажену фотографію у профілі, для цього потрібно натиснути кнопку “Попередній перегляд”, тоді відкриється модальне вікно (рис. 3.25) в якому буде видно повністю всю фотографію.



Рис. 3.25 – Фотографія профілю у повний розмір

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк. 54
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

На вкладці “Персональний тренер” випадваючого меню з ліва, в нас з’являється сторінка персонального тренера (рис. 3.26), де ми можемо подати заявку настигнувши на кнопку “Надіслати запрошення” до тренера, якого ми хоті б щоб був нашим тренером. При натисканні з’являється меню з формою заявки зправа (рис. 3.27), яке містить одне поле “Записка” та кнопка вибору тренера “Запросити”.

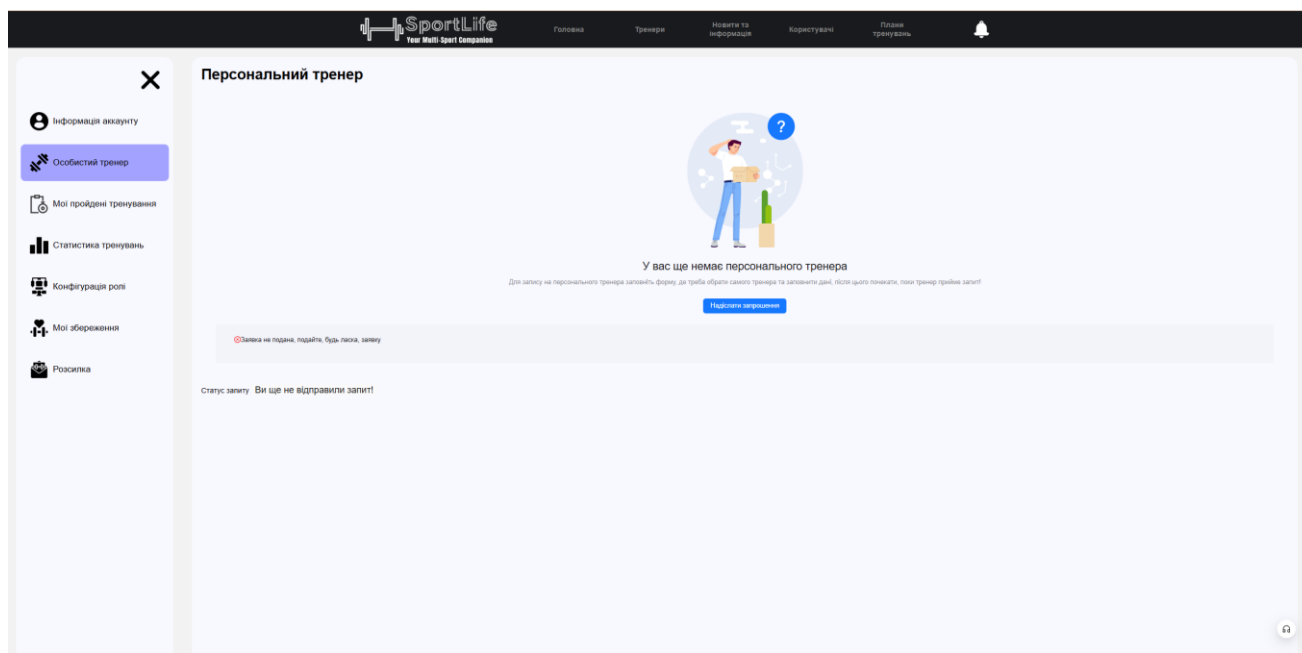


Рис. 3.26 – Сторінка персонального тренера

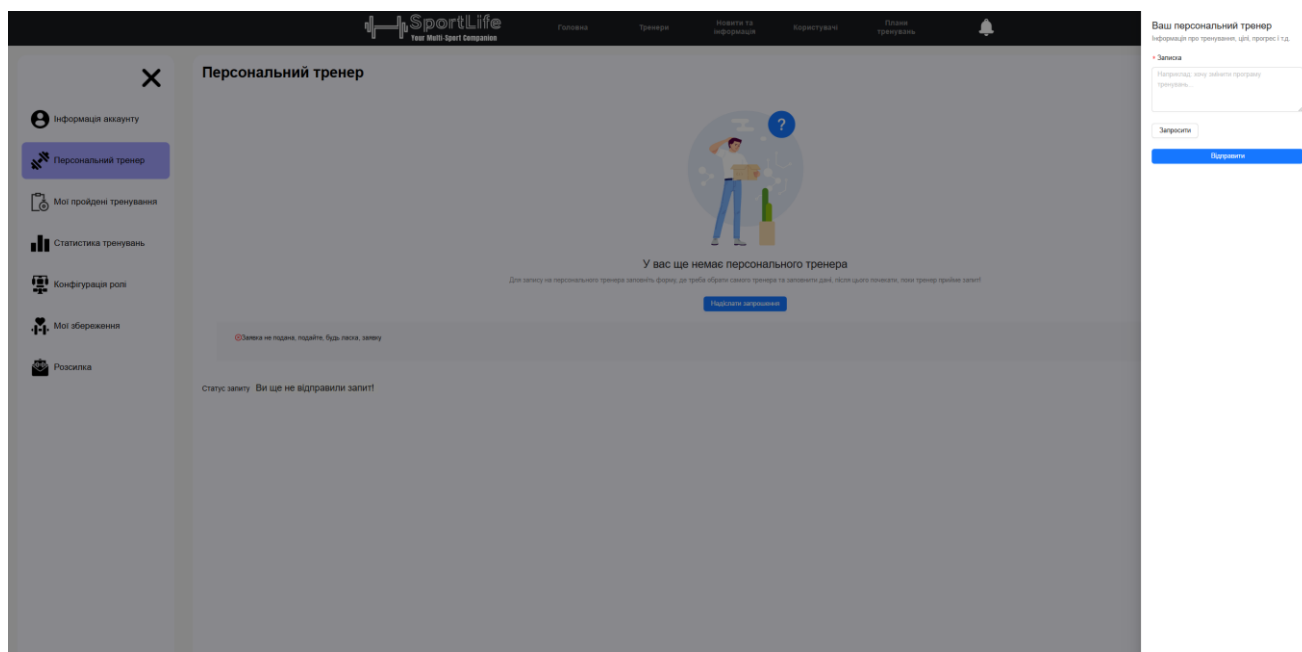


Рис. 3.27 – Меню з формою запрошення на персонального тренера

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		55

Після заповнення даних на формі (рис. 3.28), ми надсилаємо заявку тренеру та чекаємо відповіді частково передивлюючись поле “Статус запиту” в якому знаходиться статус заявки.

Ваш персональний тренер
Інформація про тренування, цілі, прогрес і т.д.

* Записка

Добро дня, ви мені потрібні як персональний тренер тому, що....

Запросити

Інформація про користувача

Ім'я: Jane Doe

Email: john.doe@example.com

Телефон: +380981654877

Роль: trainer

Опис: Оновлена інформація профілю!

Відправити

Рис. 3.28 – Заповнена форма заявки на персонального тренера

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		56

На наступній вкладці “Мої пройдені тренування” знаходяться всі деталі тренування які проходив користувач профілю (рис. 3.29), тобто історія всіх тренувань, які мають інформацію пройдених вправ самого плану.

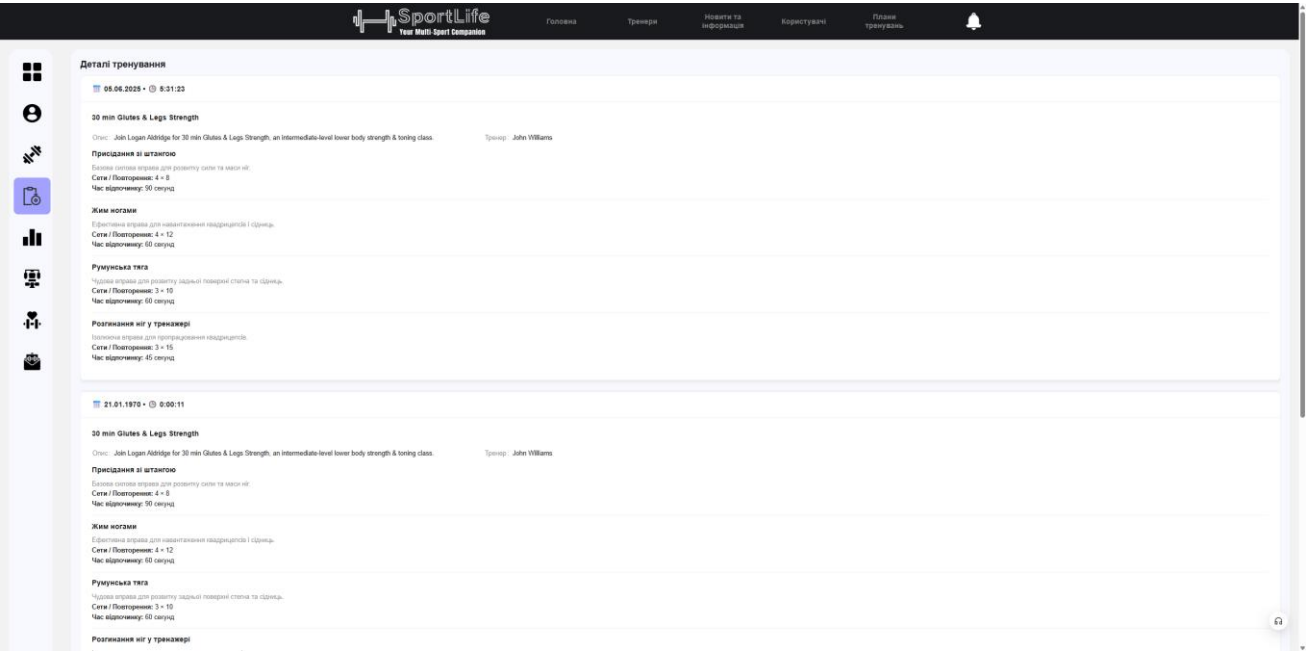


Рис. 3.29 – Сторінка історії тренувань користувача

Наступним йде вкладка “Статистика тренувань”, ми можемо побачити загальну нагрузку м’язів як статистику та календар тренувань (рис. 3.30).

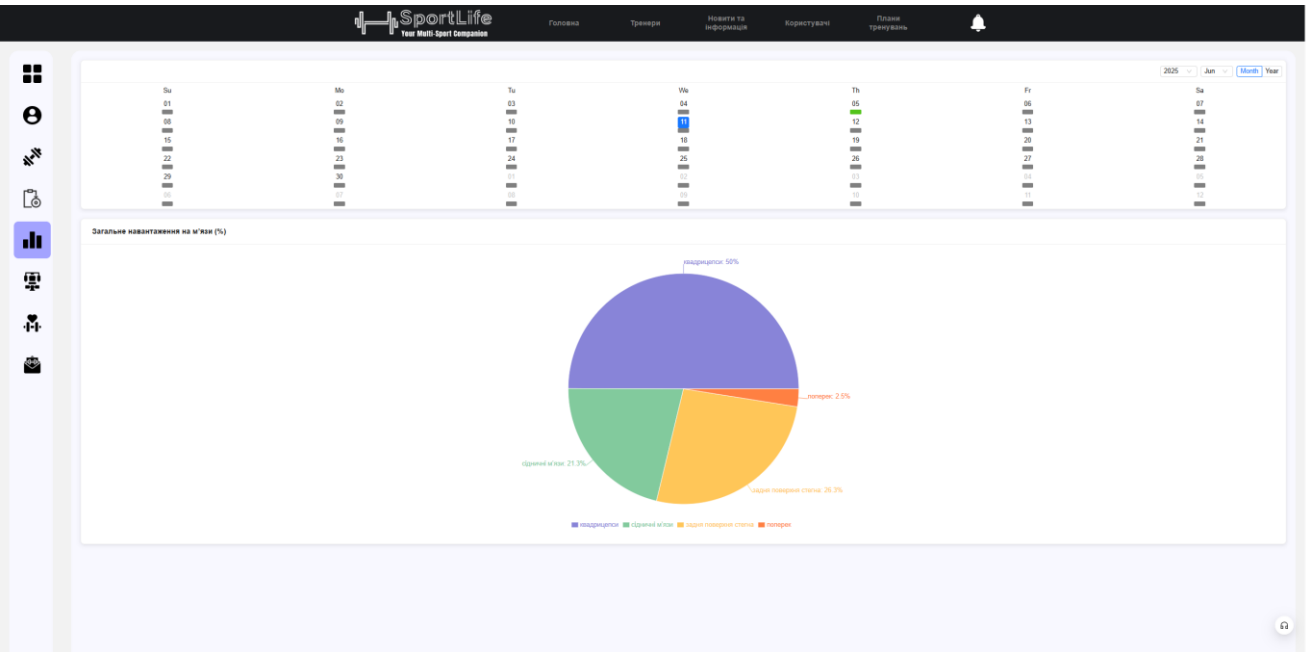


Рис. 3.30 – Сторінка статистики користувача

Далі йде вкладка “Конфігурація ролі”, в якій у користувача є змога подати заявку на зміну ролі (рис. 3.31), на сторінці з'являється поле з заповненням заявки та вибору відносно потрібної ролі.

Також на сторінці можемо побачити невелику інформацію щодо ролей (рис. 3.32).

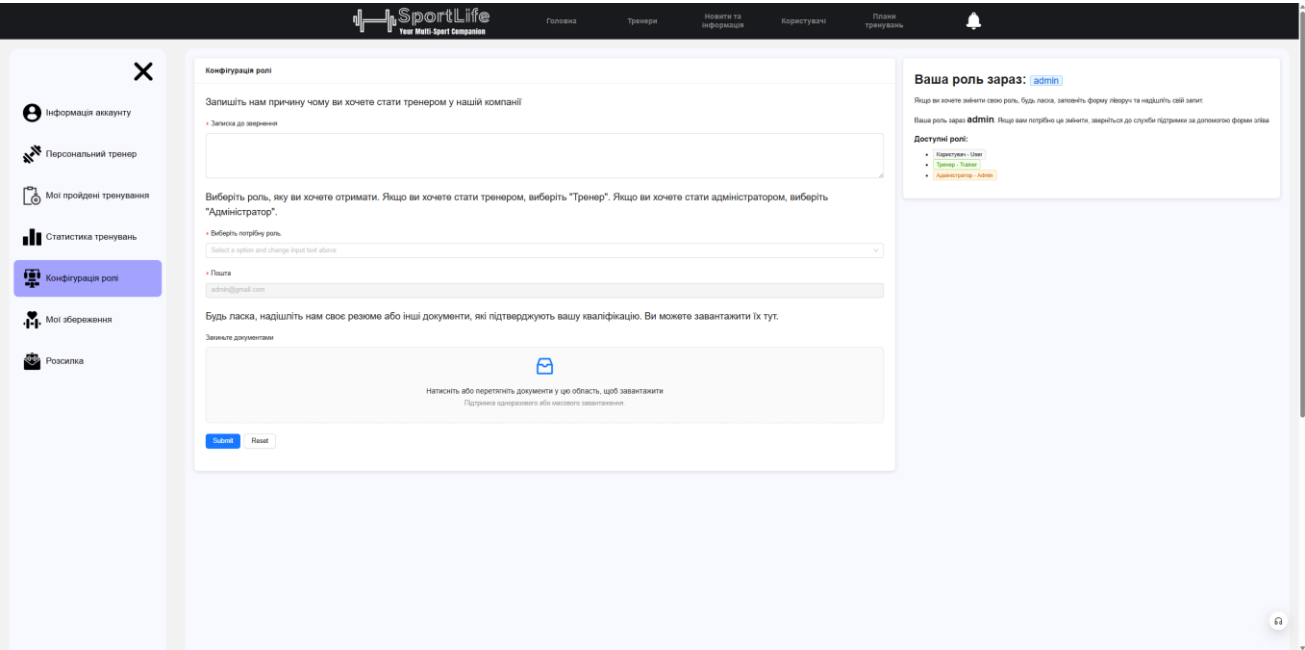


Рис. 3.31 – Сторінка з формою заявки на зміну ролі

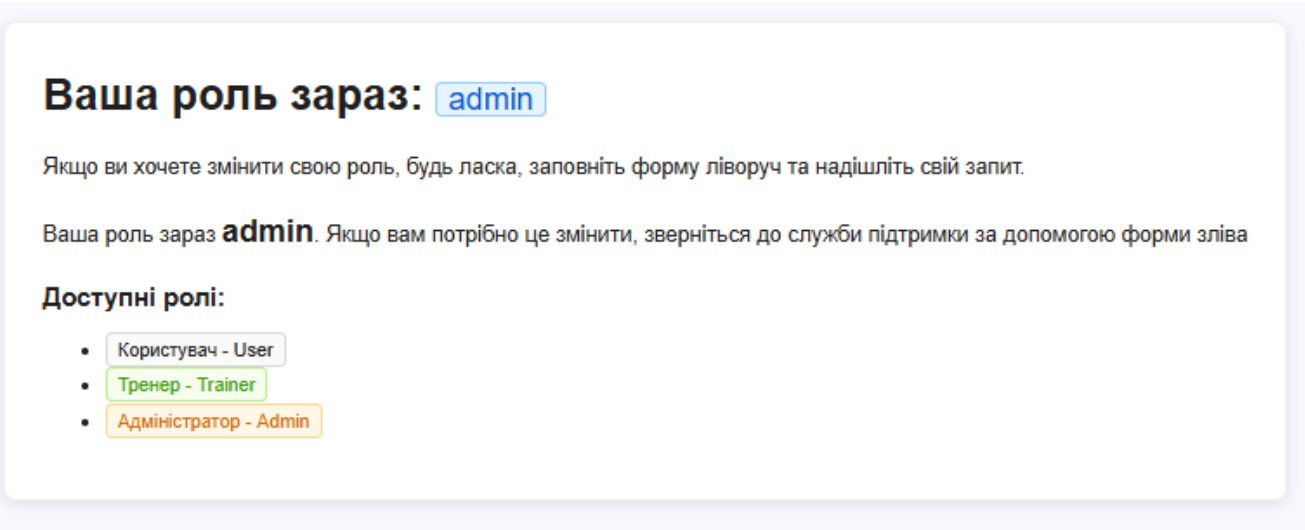


Рис. 3.32 – Елемент на сторінці з інформацією щодо ролі

Далі по лівому меню в сторінці профілю йде вкладка “Мої збереження”, на цій вкладці розташовано всі збережені пости та плани тренування (рис. 3.33) які зберіг користувач натиснувши на кнопку збереження.

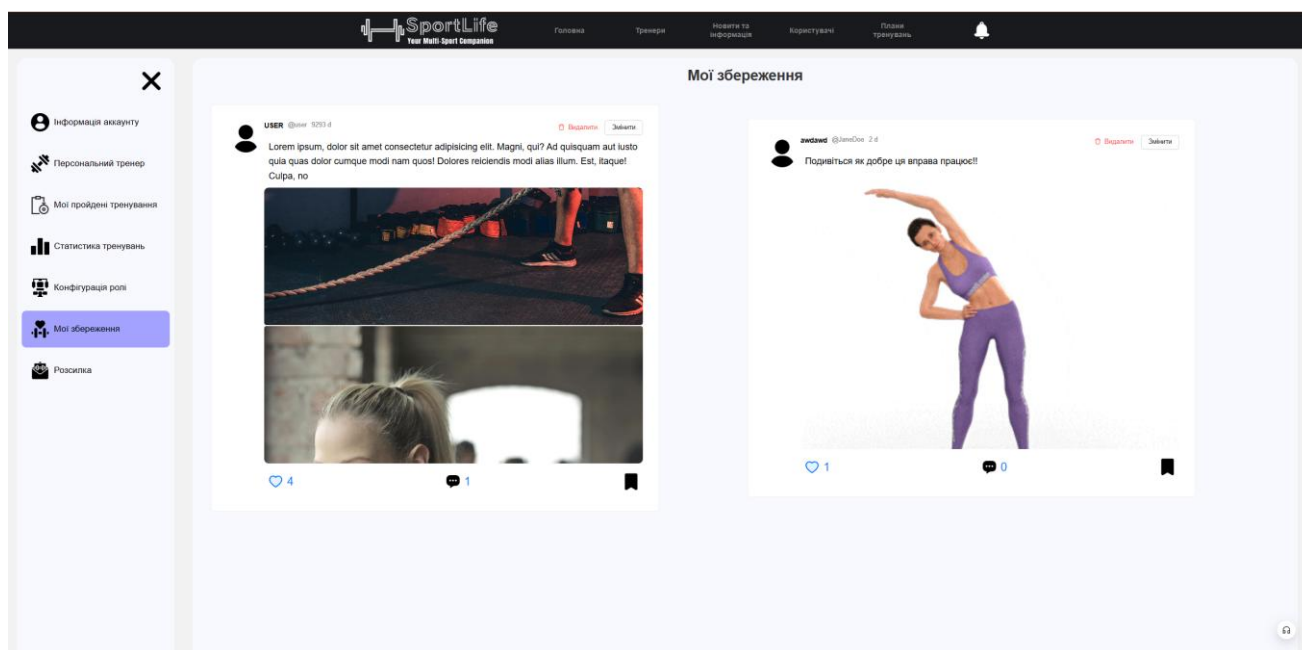


Рис. 3.33 – Сторінка збережених елементів користувача

Також у лівому меню буде розташовано ще одна вкладка яка несе назву “Розсилка” виключно для користувачів з роллю адміністратор, в якій знаходиться дві форми для розсилки (рис. 3.34). Перша форма розсилає повідомлення вебдодатку всім користувачам. Друга форма робить розсилку пошти всім користувачам які записалися на розсилку.

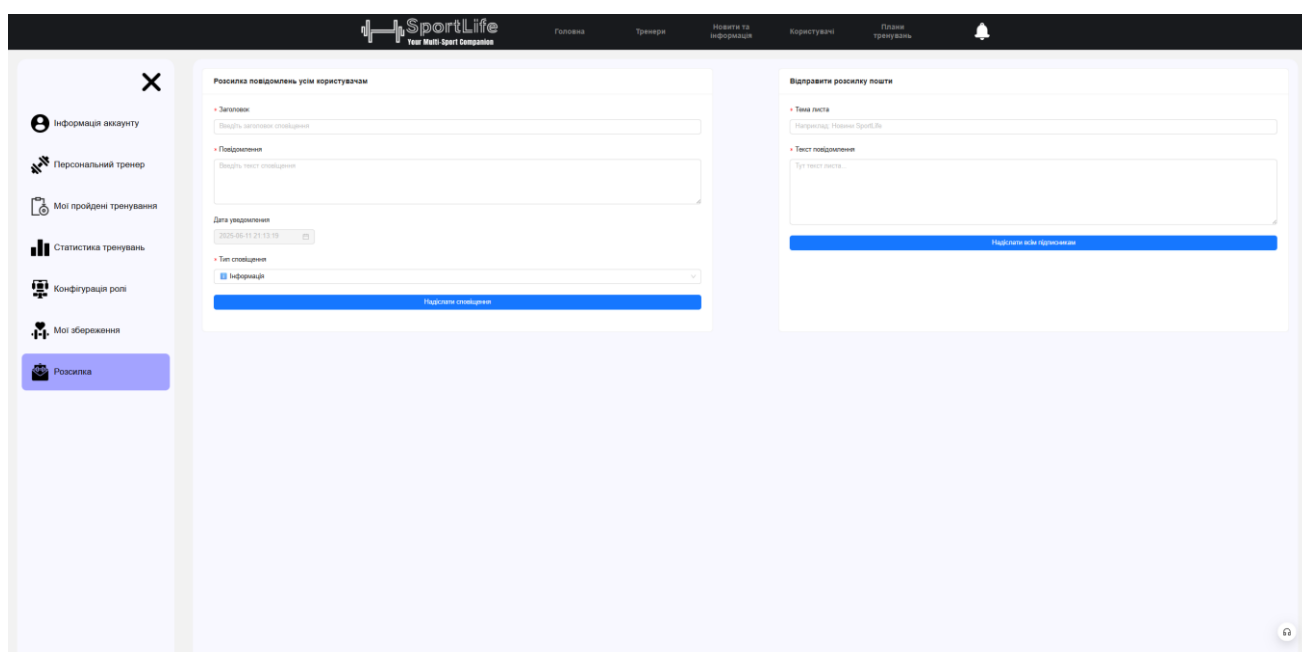


Рис. 3.34 – Сторінка невеликої адмін панелі розсилки

Після перегляду сторінки “Аккаунт” користувач має змогу повернутися на головну сторінку, а з неї переміститися до наступного розділу – “Тренери”. На

сторінці “Тренери” розміщена карточки з інформацією про всіх доступних фахівців (рис. 3.35). Якщо навести на карточку або перейти на основну сторінку тренерів (рис.3.36 - 3.37), користувач може ознайомитися з їхнім досвідом роботи, спеціалізацією та іншими важливими відомостями, які допоможуть підібрати оптимального спеціаліста для тренувань.

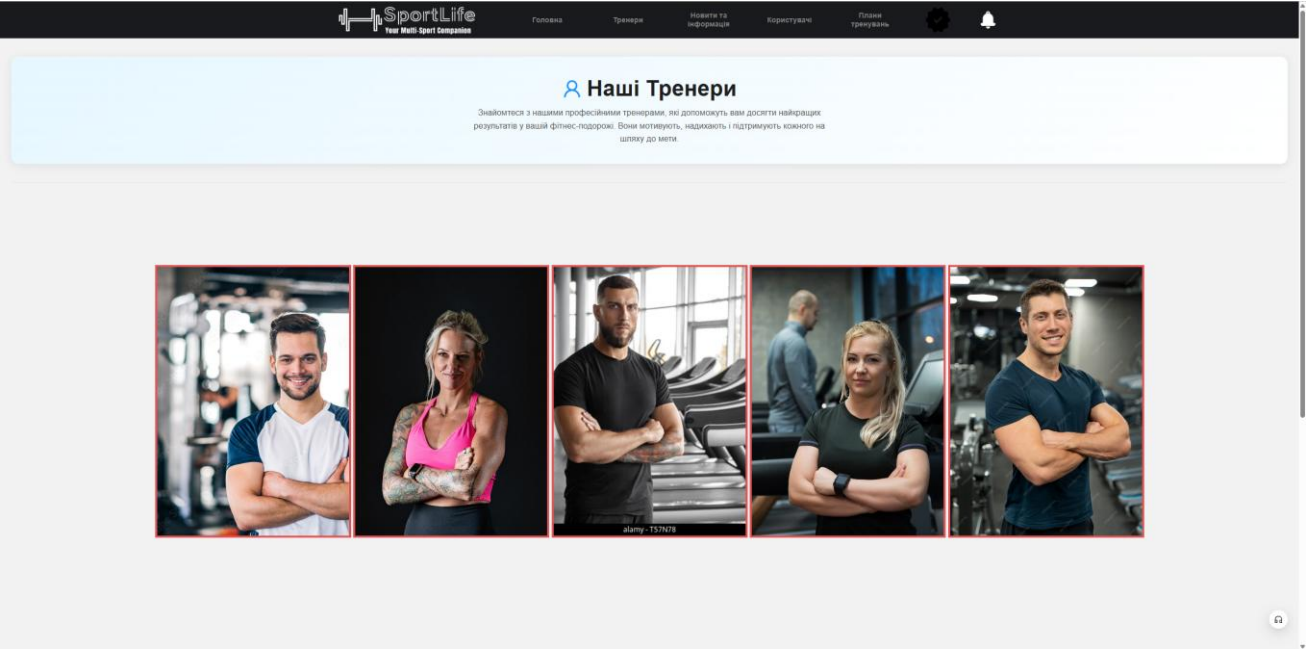


Рис. 3.35 – Сторінка інформації про тренерів

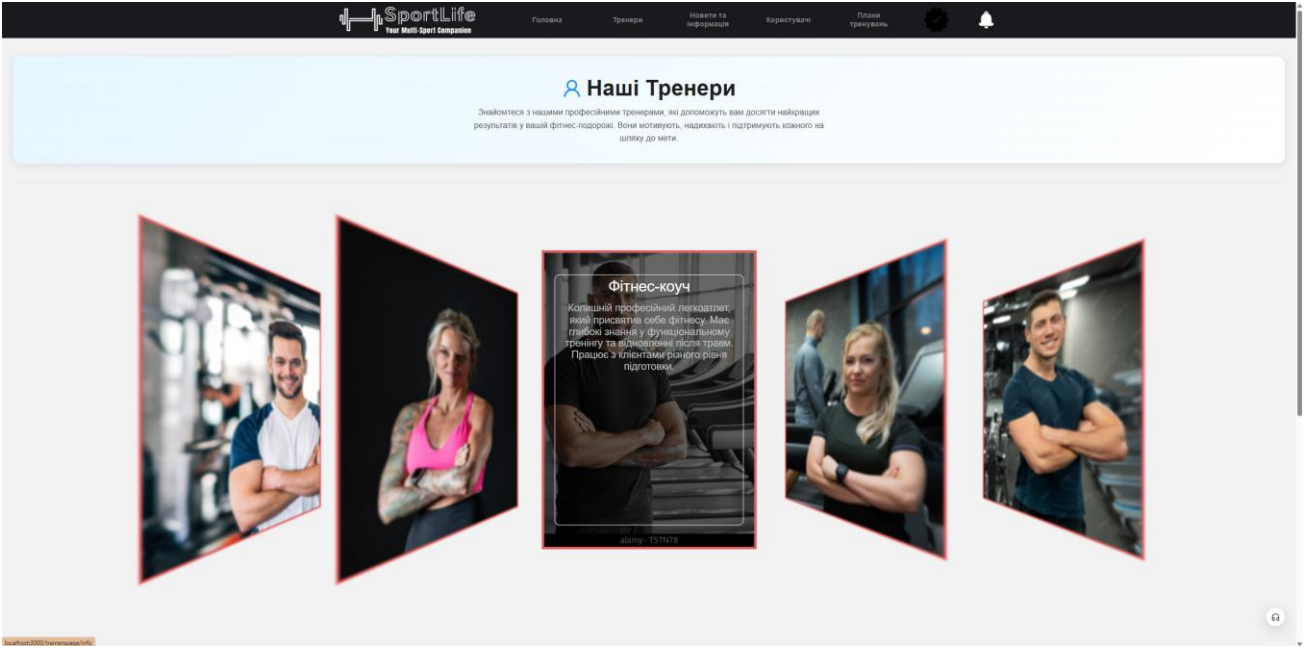


Рис. 3.36 – Коротка інформація на карточці тренера

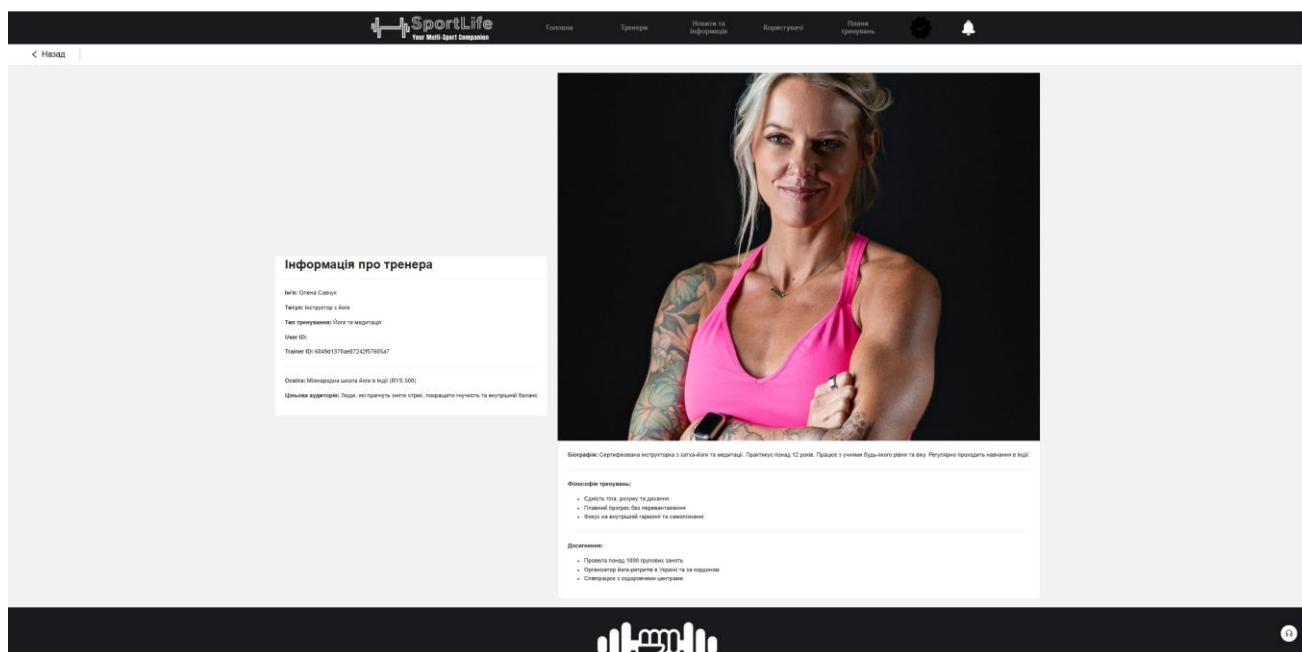


Рис. 3.37 – Сторінка більш детальної інформації при натисканні на карточку

Сторінка “Новини та інформація” в вебдодатку дозволяє переглядати пости, які пишуть користувачі. Вона слугує стрічкою, де публікуються оновлення, думки або важлива інформація, що стосується тренувань чи спільноти (рис. 3.38). Кожен допис має певну структуру. У верхній частині розміщується інформація про користувача: його ім’я нікнейм, дата створення а також фото профілю. Далі йде текстова частина, в якій користувач ділиться своїм думкамя, досягами та іншим. У межах одного посту також можуть бути додані зображення – не більше двох. Відтак сторінка дозволяє отримувати як текстову, так і візуальну інформацію від інших учасників платформи.

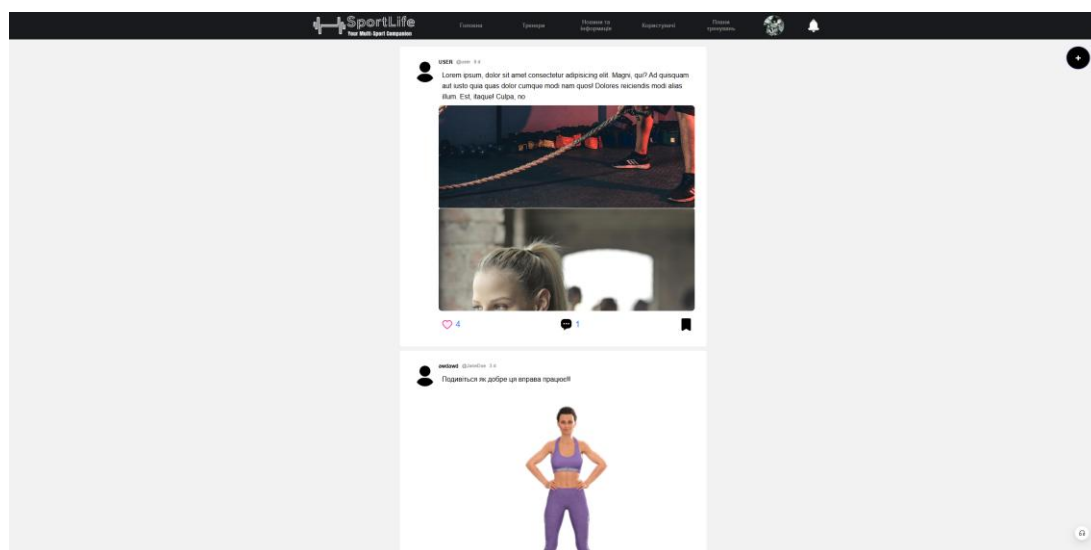


Рис. 3.38 – Сторіна новин та інформації

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		61

Також можна підмітити у верхній частині екрану елемент з плюсом, якщо навести на цей елемент він розкриється з текстом “Створити пост” (рис. 3.39), якщо натиснути на нього користувача перекидає на сторінку створення посту, створювати пост можуть тільки зареєстровані користувачі (рис. 3.40).

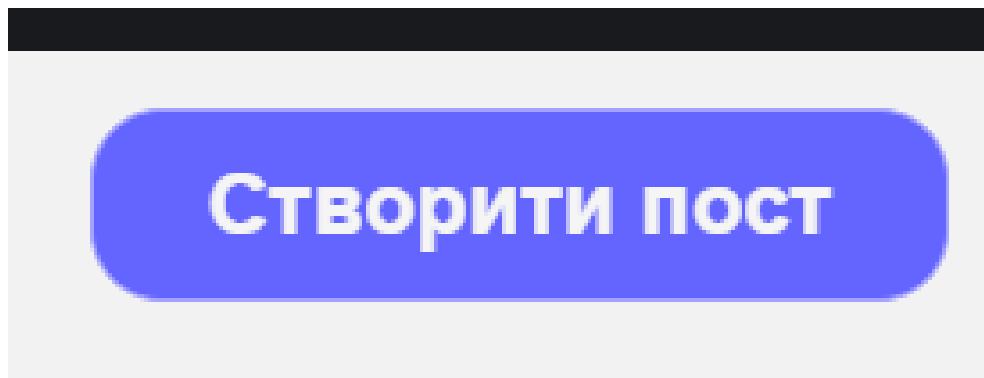


Рис. 3.39 – Кнопка для створення поста

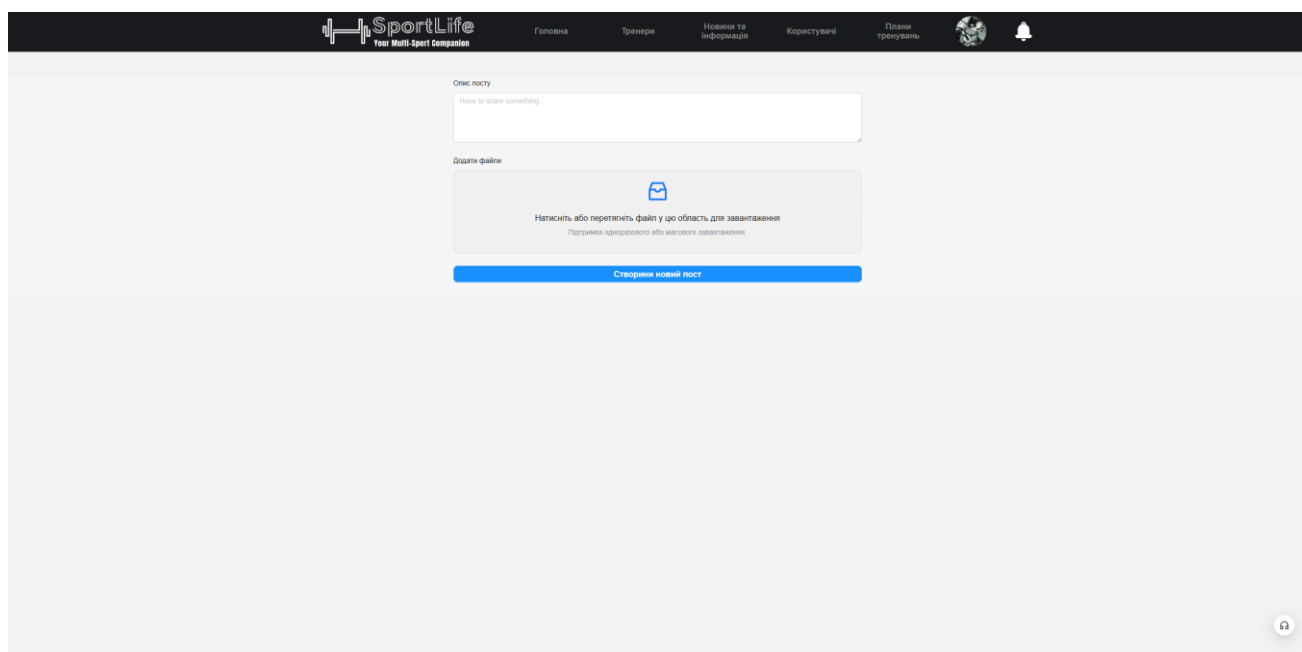


Рис. 3.40 – Сторінка створення посту

Також якщо створити пост, в правому кутку з’явиться дві кнопки для коригування, перша кнопка це “Видалити” та “Змінити” (рис. 3.41). Якщо натиснути на кнопку “Видалити” тоді з’явиться попередження про видалення (рис. 3.42), і якщо прийняти тоді пост видаляється. Наступне це кнопка “Змінити”, якщо натиснути на кнопку тоді користувача перекине на сторінку редагування свого створеного посту (рис. 3.43). Якщо користувач має роль адміністратора тоді ці кнопки корегування доступні для усіх постів.

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		62

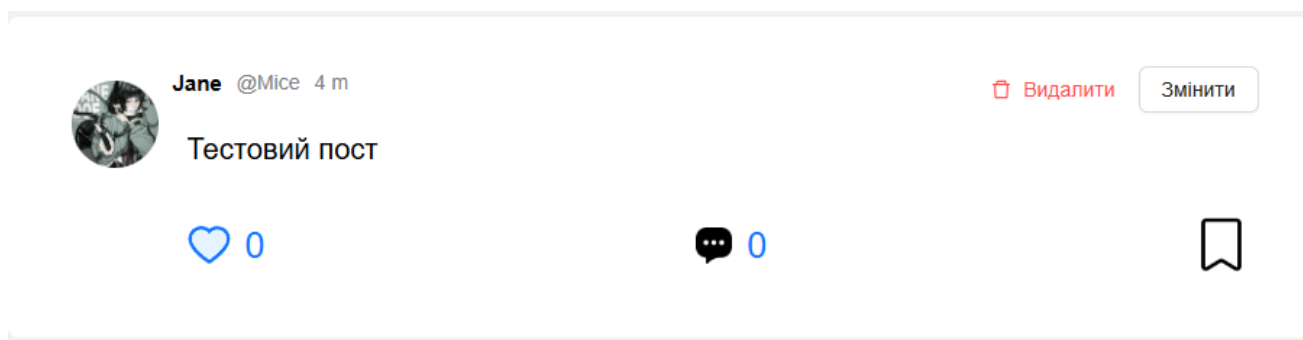


Рис. 3.41 – Користувач створив пост

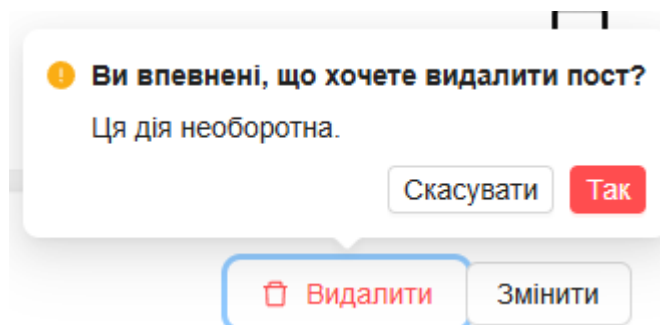


Рис. 3.42 – Користувач натиснув на кнопку видалення

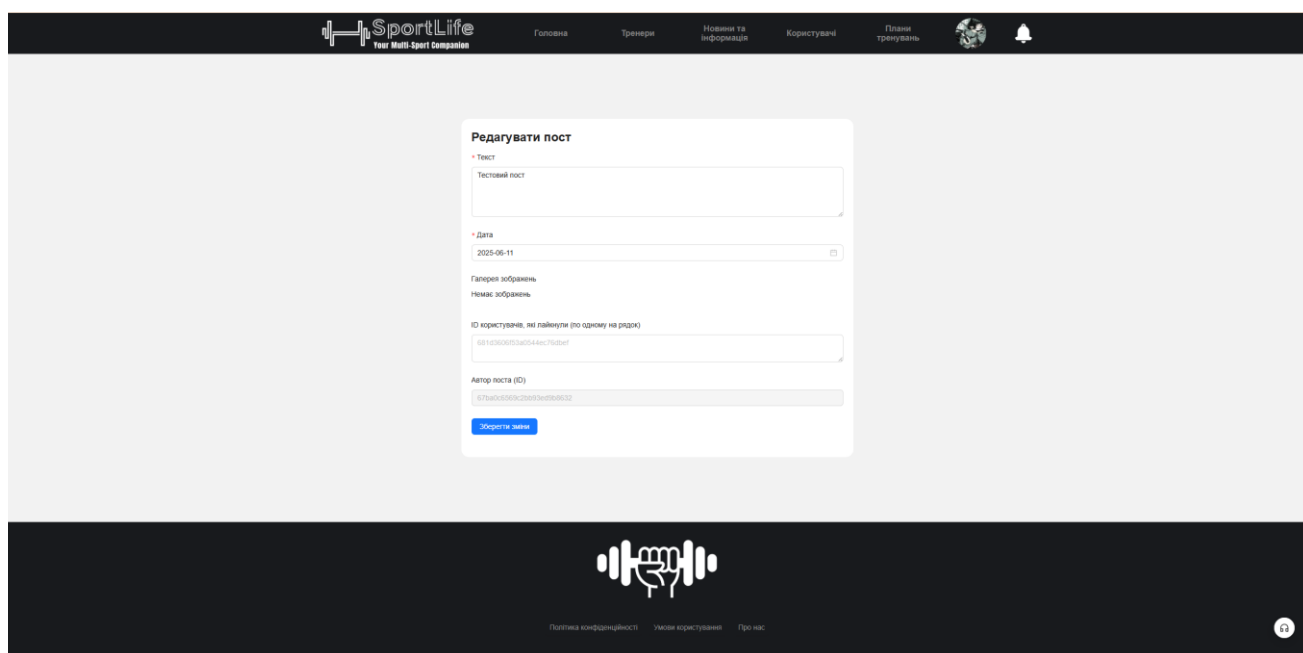


Рис. 3.43 – Сторінка редагування посту

Також якщо користувач натисне на елемент вподобайку, тоді елемент змінить коляр на більш рожевий та добавиться до кількості вподобань сам користувач (рис. 3.44). Якщо користувач натисне на елемент коментаря тоді його перекине на сторінку коментарів, які зможе написати або видалити вже написанні (рис. 3.45). Останній елемент у пості виконує функцію збереження (рис. 3.46). При

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		63

натисканні, відповідний пост додається до вкладки “Мої збереження”, яка розташована на сторінці “Аккаунт”.

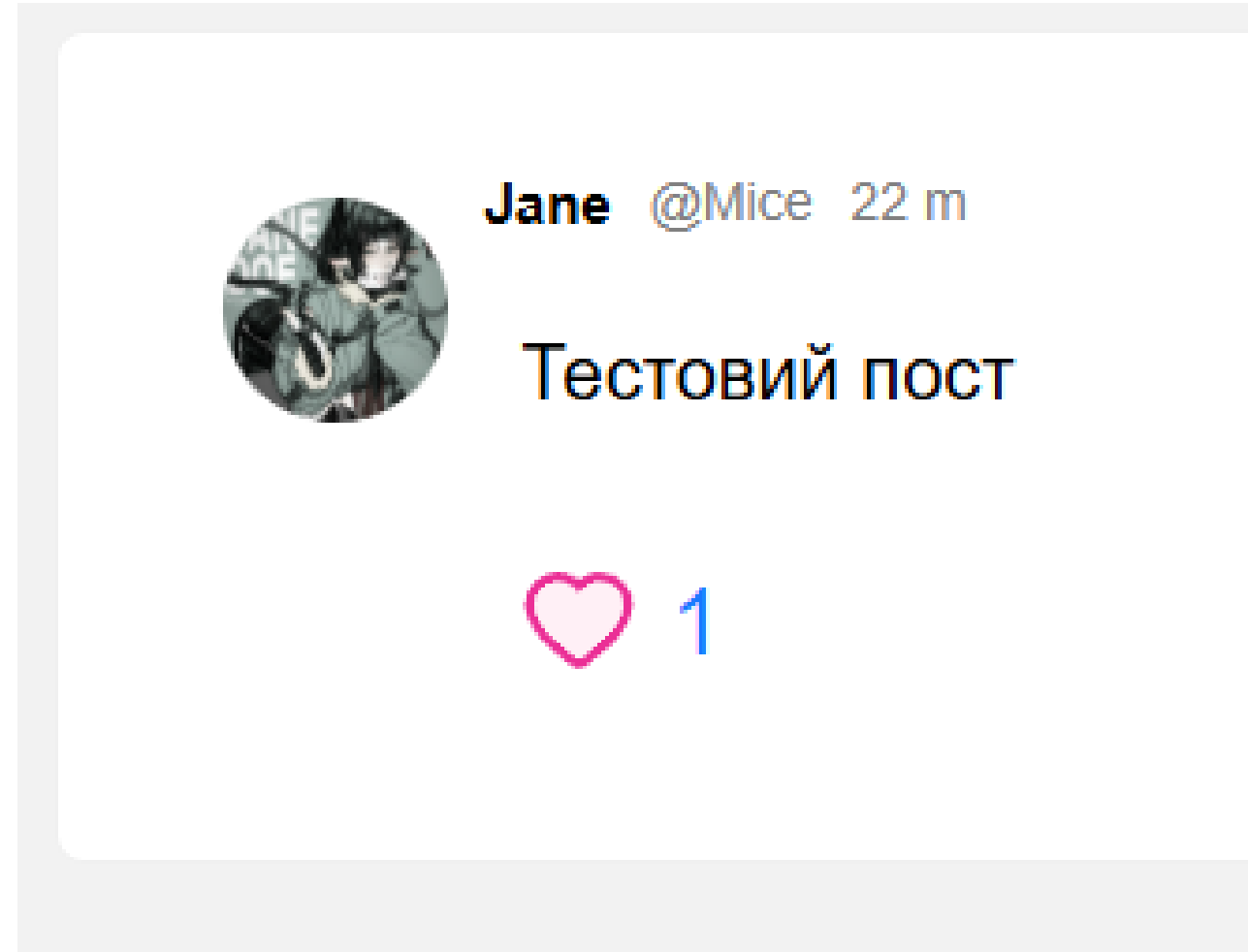


Рис. 3.44 – Натиснута вподобайка

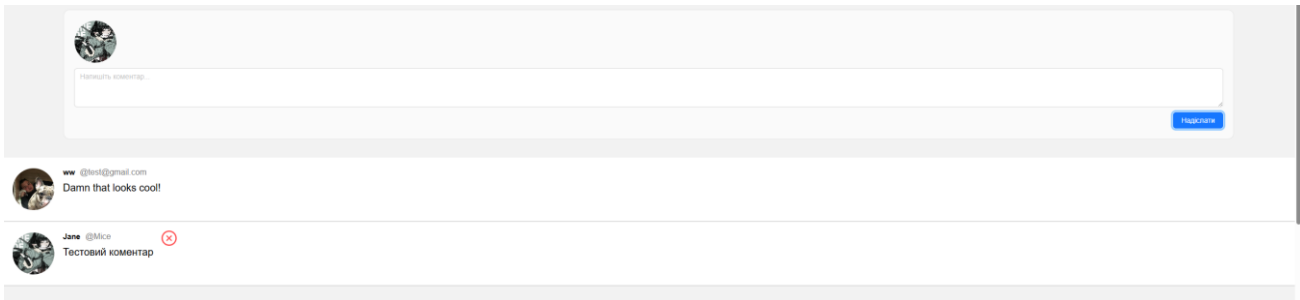


Рис. 3.45 – Написаний коментар користувачем

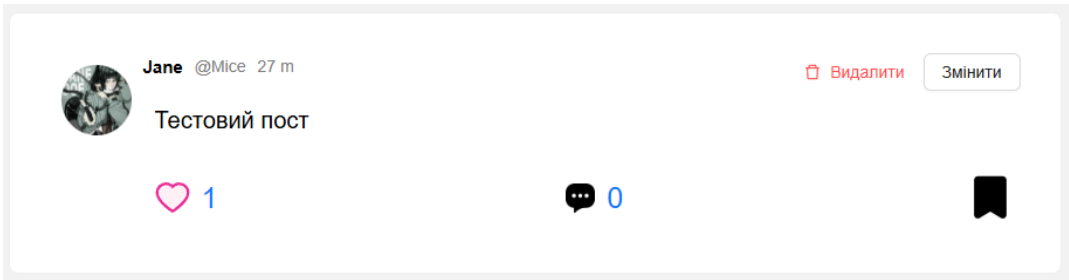


Рис. 3.46 – Натиснута кнопка збереження

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		64

Сторінка “Користувачі” містить список усіх зареєстрованих користувачів вебдодатку (рис. 3.47). На сторінці можна замітити пошук користувачів по пошті, ім'я або нікнейму (рис. 3.48). При натисканні на будь-який з блоків, відбувається перенаправлення на особисту сторінку профілю обраного користувача Інформація користувача представлена у вигляді окремого блоку, де міститься такі блоки як - ім'я, фото профілю, опис, дата створення облікового запису та інше. Також на сторінці профілю можемо побачити усі пости створені користувачем (рис. 3.49).



Рис. 3.47 – Сторінка користувачів

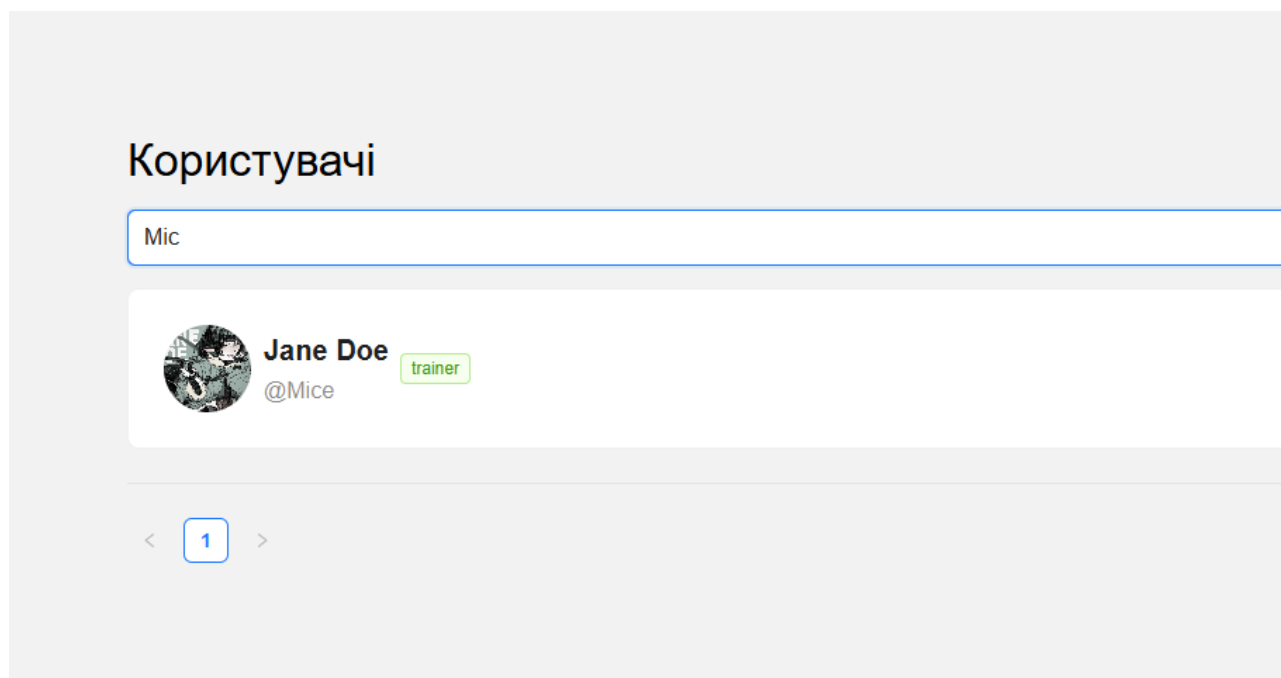


Рис. 3.48 – Результат пошуку користувачів

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		65

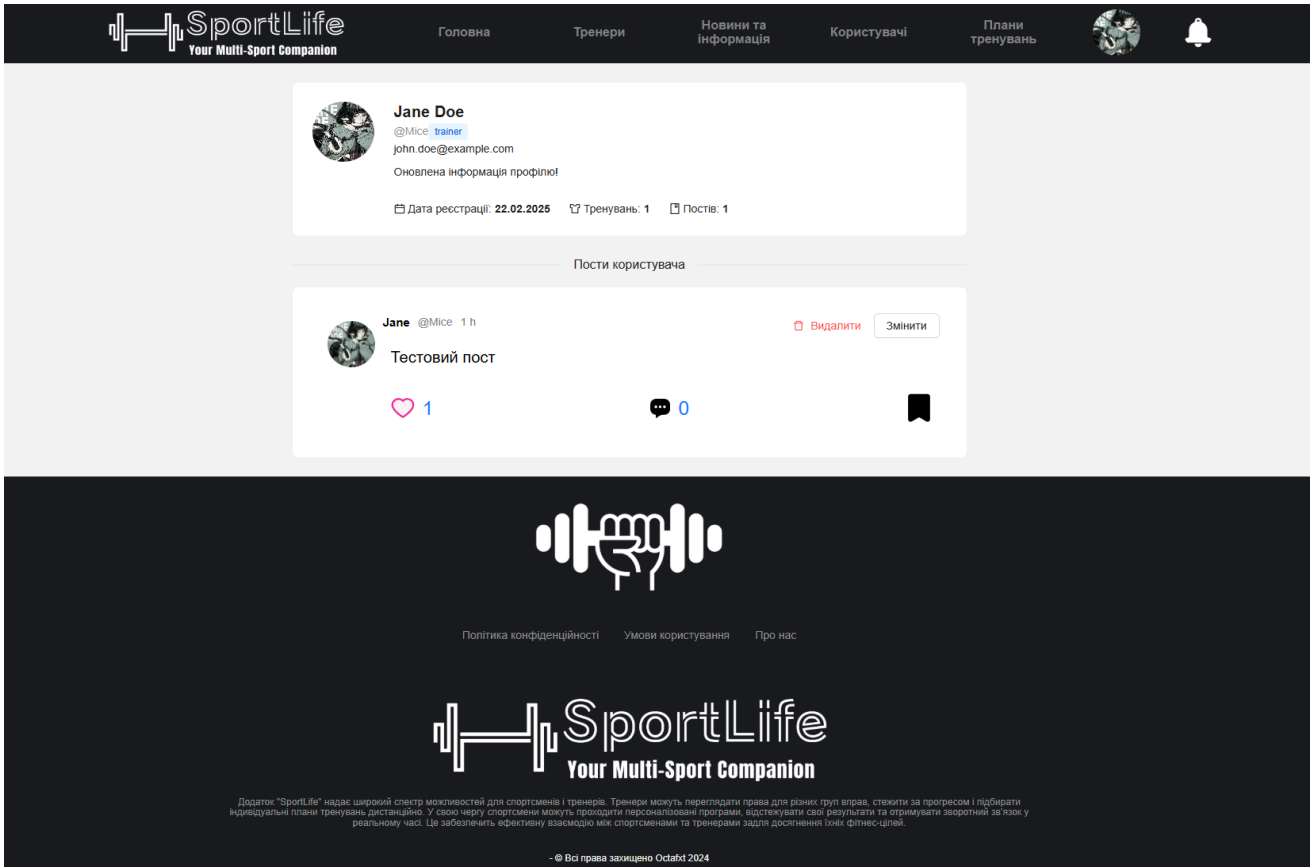


Рис. 3.49 – Сторінка профілю самого користувача

Далі йде головна сторінка для тренувань. Вона має назву "Плани тренувань", на цій сторінці присутній фільтр який дозволяє користувачам швидко зорієнтуватися у доступних варіантах(рис. 3.50 – 3.51). Також на сторінці присутній елемент команди тренерів (рис. 3.36). Ще кінці сторінки є невеликий блоком в якому присутні декілька відповіді на питання (рис. 3.52).

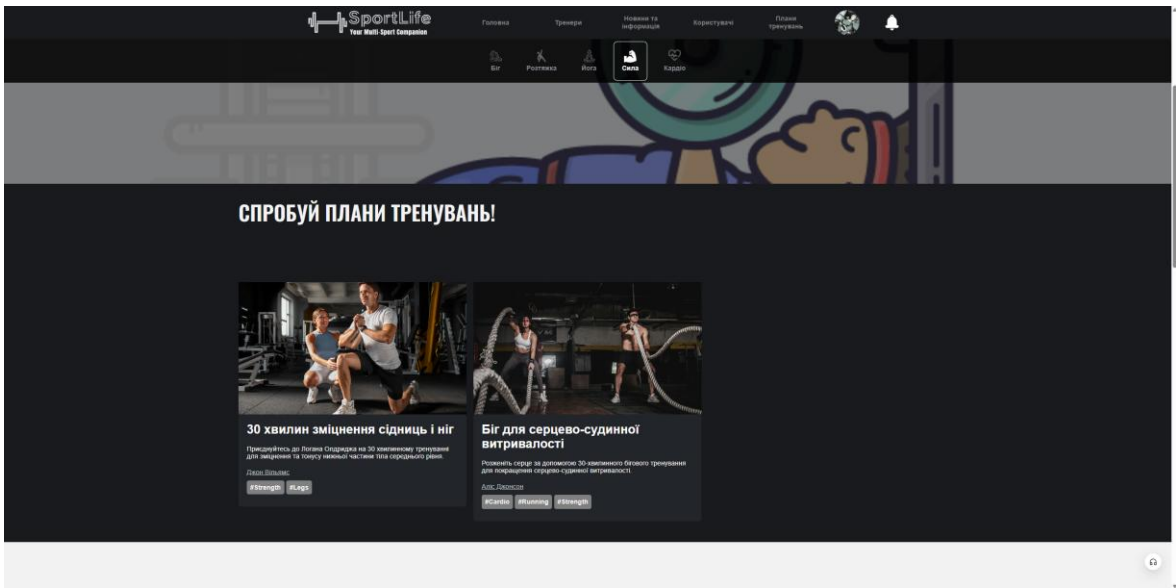


Рис. 3.50 – Сторінка плану тренувань

		Левкович О.О.			ІІЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				66
Змн.	Арк.	№ докум.	Підпис	Дата		

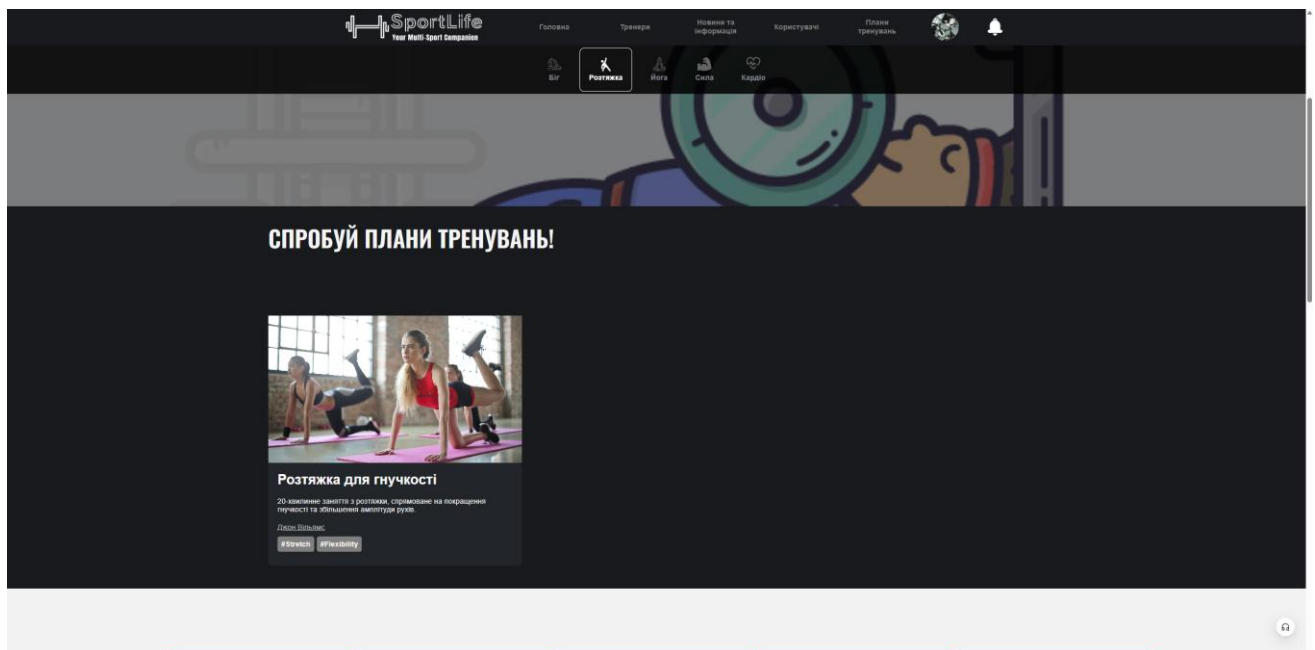


Рис. 3.51 – Приклад фільтру на списку планів тренувань

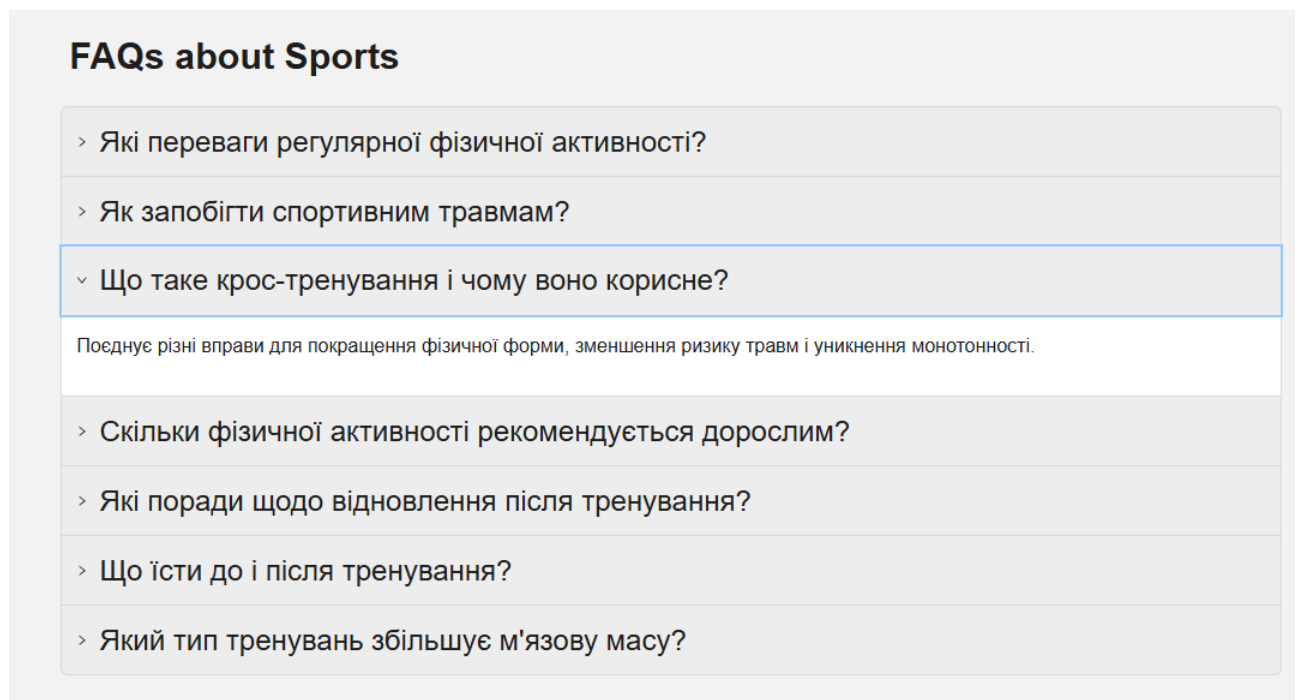


Рис. 3.52 – Відповіді на питання

Якщо користувач натискає на обраний план тренувань зі списку, система перенаправляє його на окрему сторінку з розширеною інформацією про цей план. На цій сторінці передбачено наявність кнопки для створення власної кімнати та збереження плану у вкладку “Мої збереження” на сторінці “Аккаунт” (рис. 3.53). У разі натискання на цю кнопку, автоматично створюється персональна кімната, після чого користувача перенаправляє у відповідний розділ інтерфейсу (рис. 3.54).

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		67

У межах створеної кімнати користувач отримує додаткові функції: можливість надсилати запрошення іншим користувачам, почати тренування, задати статус камері або покинути кімнату. Якщо запросити користувача, після підтвердження запрошений користувач з'явиться над власником кімнати (рис. 3.55).

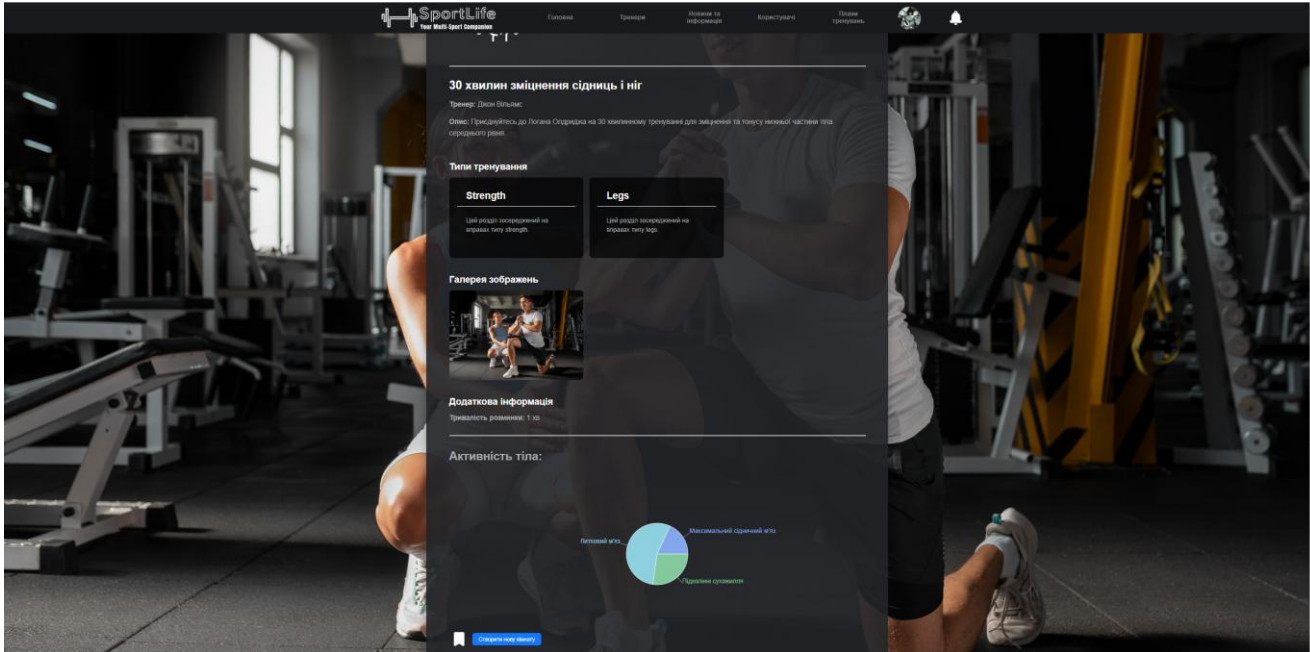


Рис. 3.53 – Сторінка інформації плану тренувань

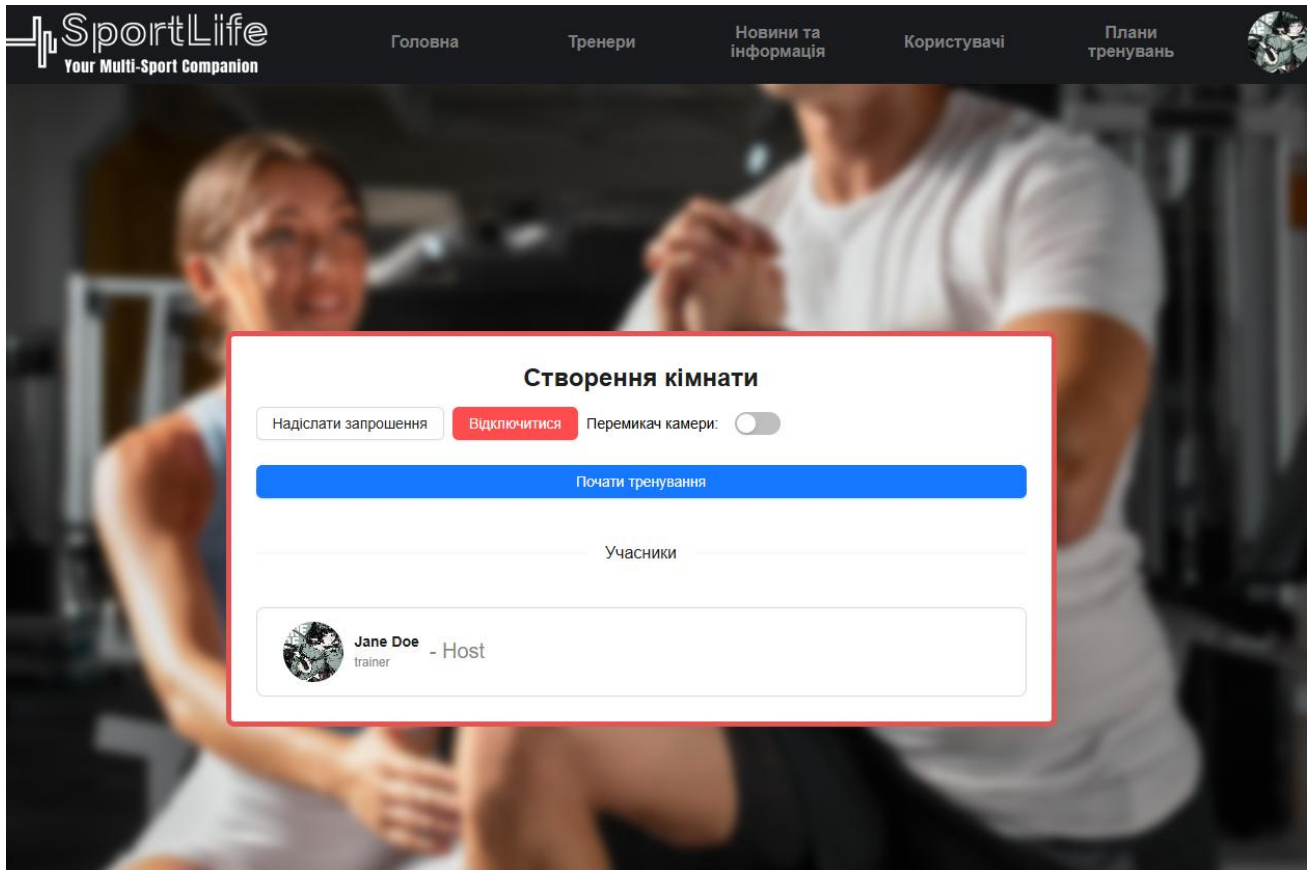


Рис. 3.54 – Сторінка кімнати

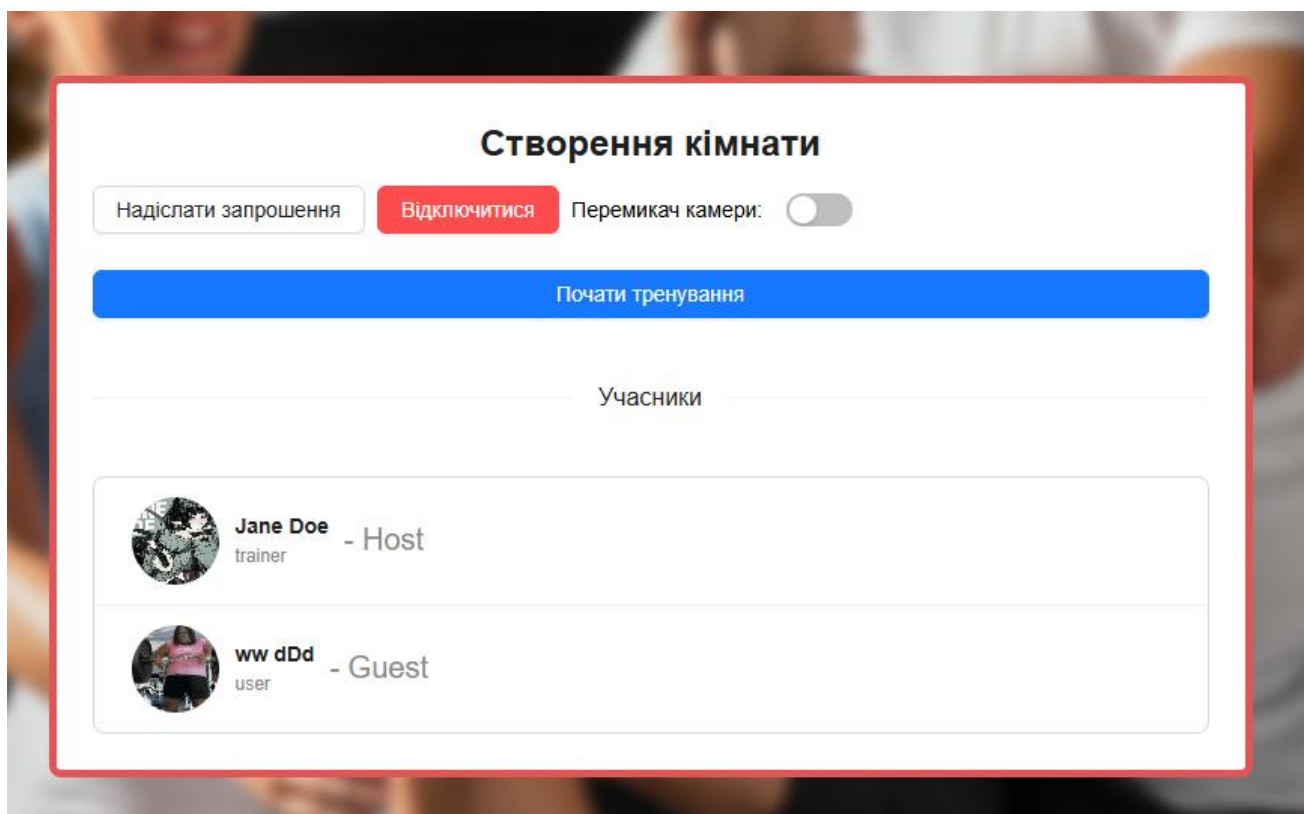


Рис. 3.55 – Приклад запрошення гостя

Якщо почати тренування тоді всіх людей в кімнаті перенаправить на сторінку проходження плану тренувань (рис. 3.56), в якій можна побачити картки з детальною інформацією про вправи з якою користувач може мати можливість як мога краще та точніше виконувати самі вправи, а також фотографію або гіфку для перегляду правильного положення під час виконання яке також допоможе у правильному виконанні самої вправи.

Також внизу знаходиться кнопка для завершення вправи та переходу до наступної, у протилежному куті знаходиться статус вправи, який може мати одне з трьох значень, “В процесі”, “Завершено” або “Очікується”. Якщо користувач переходить до наступної вправи тоді статус попередньої вправи становиться на статус “Завершено” а наступна в статус “В процесі”.

У верхньому лівому куті екрана знаходиться вікно з камерою, яке можна перетягнути в будь-яке зручне місце, в цьому вікні знаходиться камера власника кімнати яку він може вимкнути на сторінці кімнати.

Також внизу самого екрана знаходиться невелика інформація про деталі самої кімнати.

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		69



Рис. 3.56 – Сторінка проходження плану тренувань

Якщо натиснути кнопку “Наступна вправа”, тоді наступна вправа розблокується, а попередня отримає в статус “Завершено” (рис. 3.57). Також змінюються данні відсотку проходження у верхньому меню, аналогічно у нижньому під назвод “Вправи”. При зміні любых даних, гості підключені до кімнати отримують тіж самі оновлення що у власника кімнати.

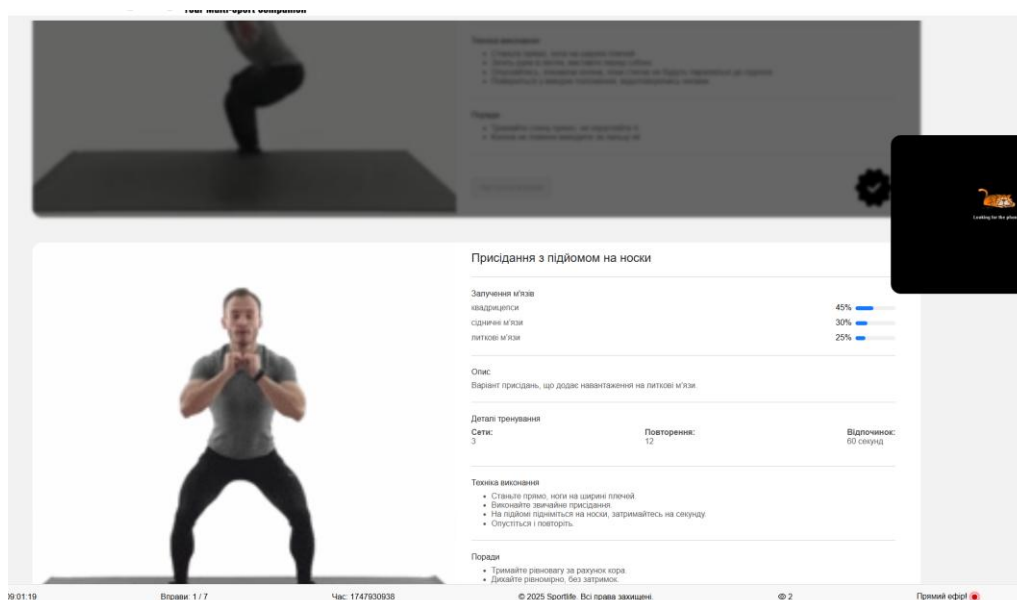


Рис. 3.57 – Приклад завершення вправи

Після проходження всіх зазначених вправ, в кінці розблокується кнопка “Завершити тренування” (рис. 3.58). При натисканні на кнопку завершення власника кімнати направляє на результат проходження (рис. 3.59), а всіх гостей на

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		70

головну. Проходжений план зберігається у історію (рис. 3.29) і до статистики додаються нові дані (рис. 3.30).

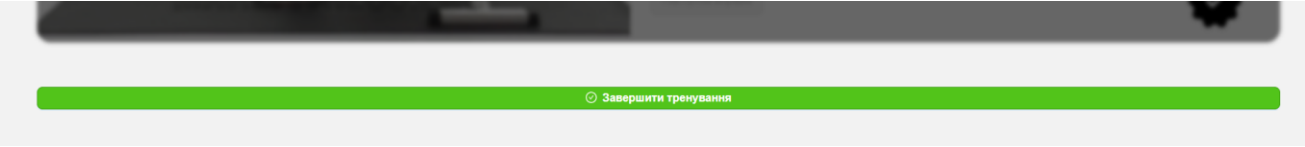


Рис. 3.58 – Кнопка завершення тренування

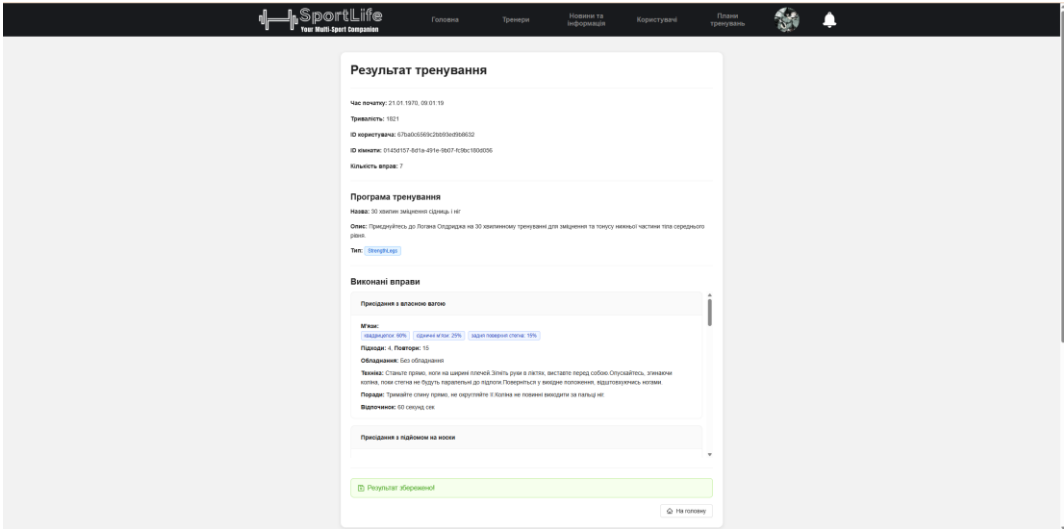


Рис. 3.59 – Сторінка результатів плану тренувань

При натисканні на елемент чату (рис. 3.14) у нижній частині екрана відкриється меню з доступними чатами (рис. 3.60). У цьому меню користувач має можливість переглядати список чатів, перемикатися між ними, створювати нові, а також надсилати повідомлення в обраний чат.

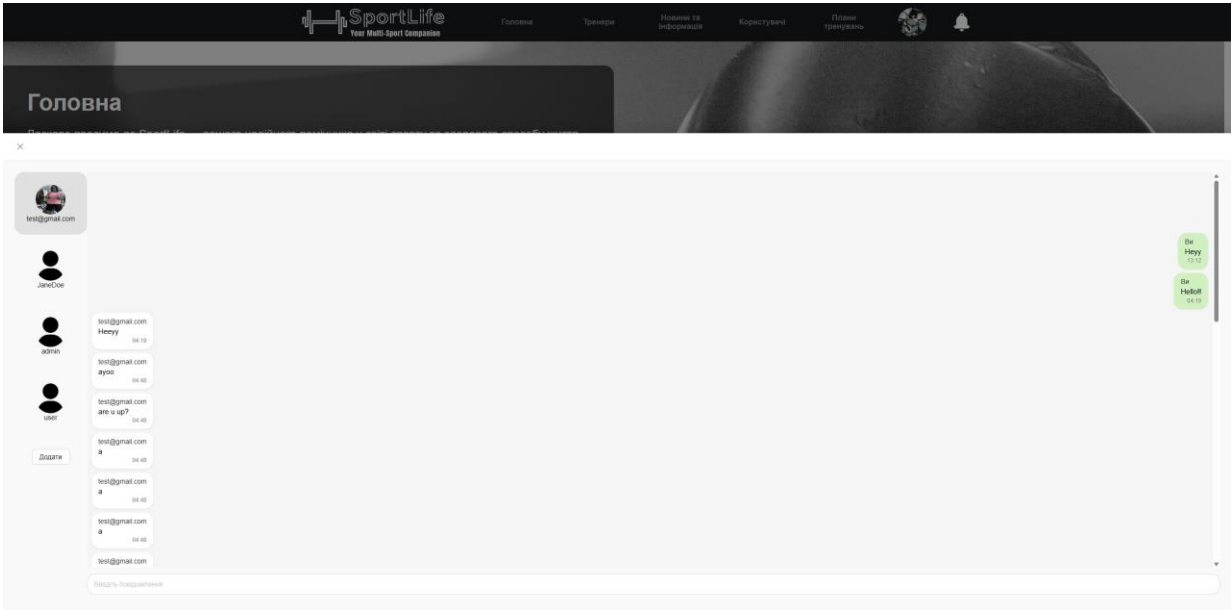


Рис. 3.60 – Відкрите меню чату

Якщо чат закритий і хтось напише користувачу повідомлення, тоді воно висвітлиться у верхньому правому куті екрану (рис. 3.61). Також продемонструємо приклад форми листа заяви зміни ролі на рис. 3.62, також повідомлення запрошення у кімнату який наведено на рис. 3.63.

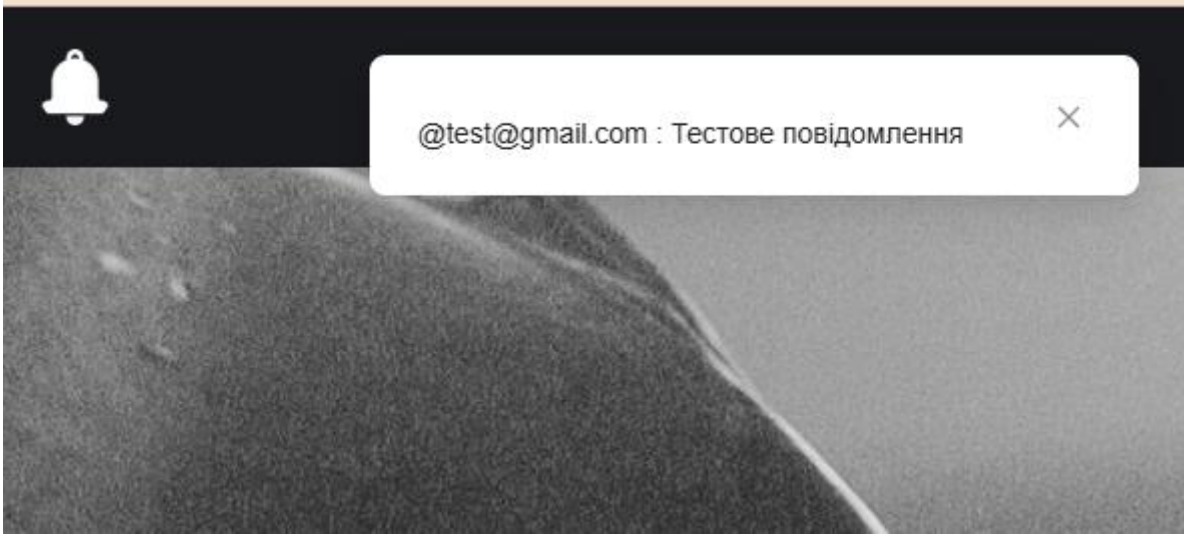


Рис. 3.61 – Повідомлення від іншого користувача

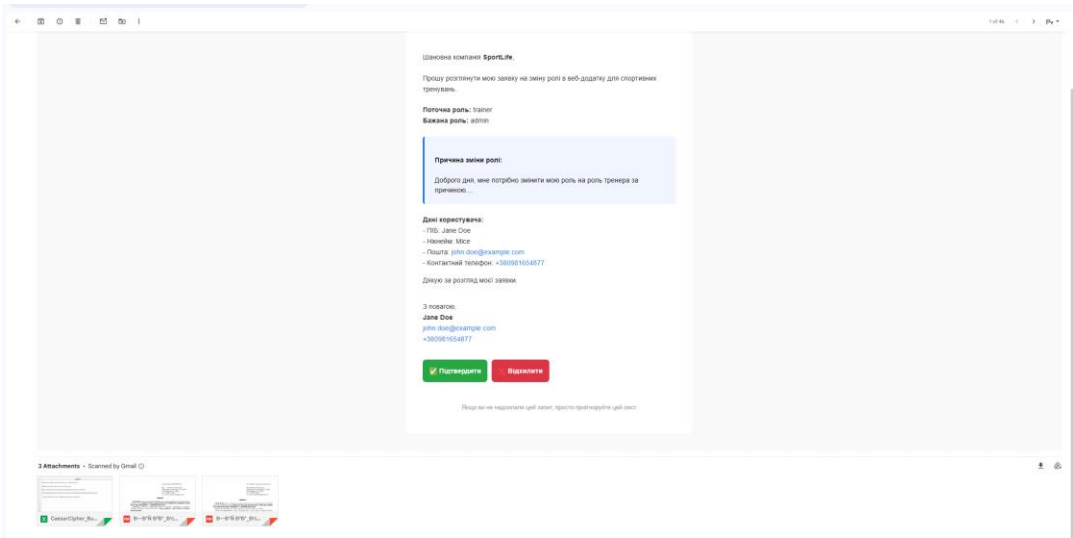


Рис. 3.62 – Приклад листу заявки зміни ролі на корпоративній пошті

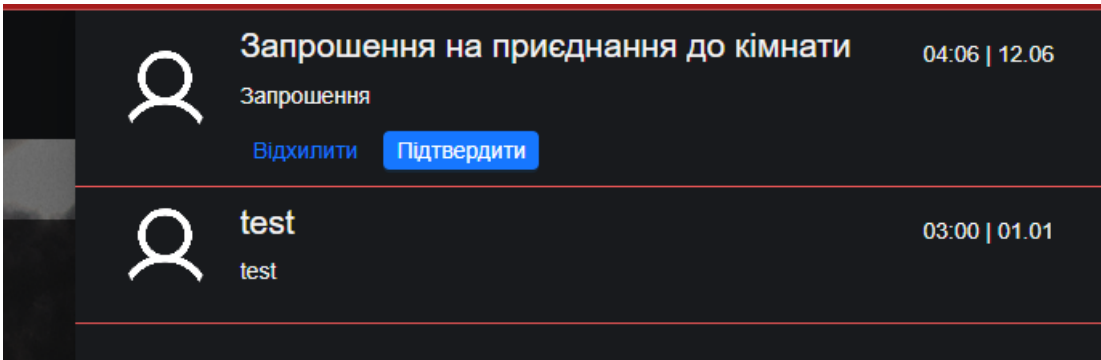


Рис. 3.63 – Приклад повідомлення

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				72
Змн.	Арк.	№ докум.	Підпис	Дата		

3.3 Тестування роботи програмного комплексу

Тестування вебдодатку це одна з важливих етапів у розробці проєкту, цей етап потрібен для того, щоб перевірити вебдодаток на працездатність, тобто при етапі тестування, бажано як умога сильніше зменшити відсоток зтикнення користувачу додатку з критичною помилкою.

В створеному вебдодатку дистанційни тренувань - серверної сторони REST API самостійна перевіряє всі запити на наявність потрібних даних і правильність їхньої структури. Тобто, якщо у якомусь випадку – наприклад, дані не передані або некоректні – система не виконає дію, і поверне відповідне повідомлення про помилку.

Для прикладу проведемо тестування кімнати плану тренувань, декілька можливих ситуацій які можуть викликати помилку в з'єднанні користувачів.

Підключення користувача до кімнати.

тестування: Власник кімнати надсилає запрошення користувачу та очікую його приєднання. В той самий час запрошений користувач отримує повідомлення (рис. 3.63). Якщо гість приймає запрошення тоді його перенаправляє в кімнату, та додається до списку користувачів (рис. 3.64).

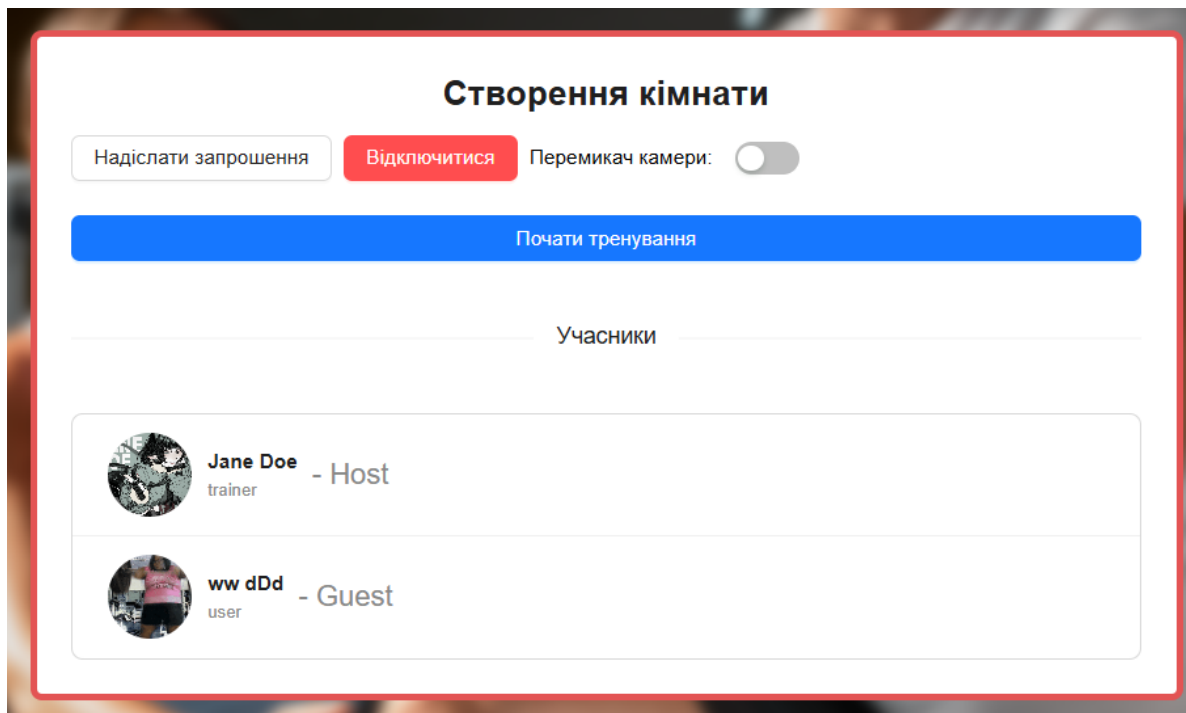


Рис. 3.64 – Приклад від власника - підключення до кімнати

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		73

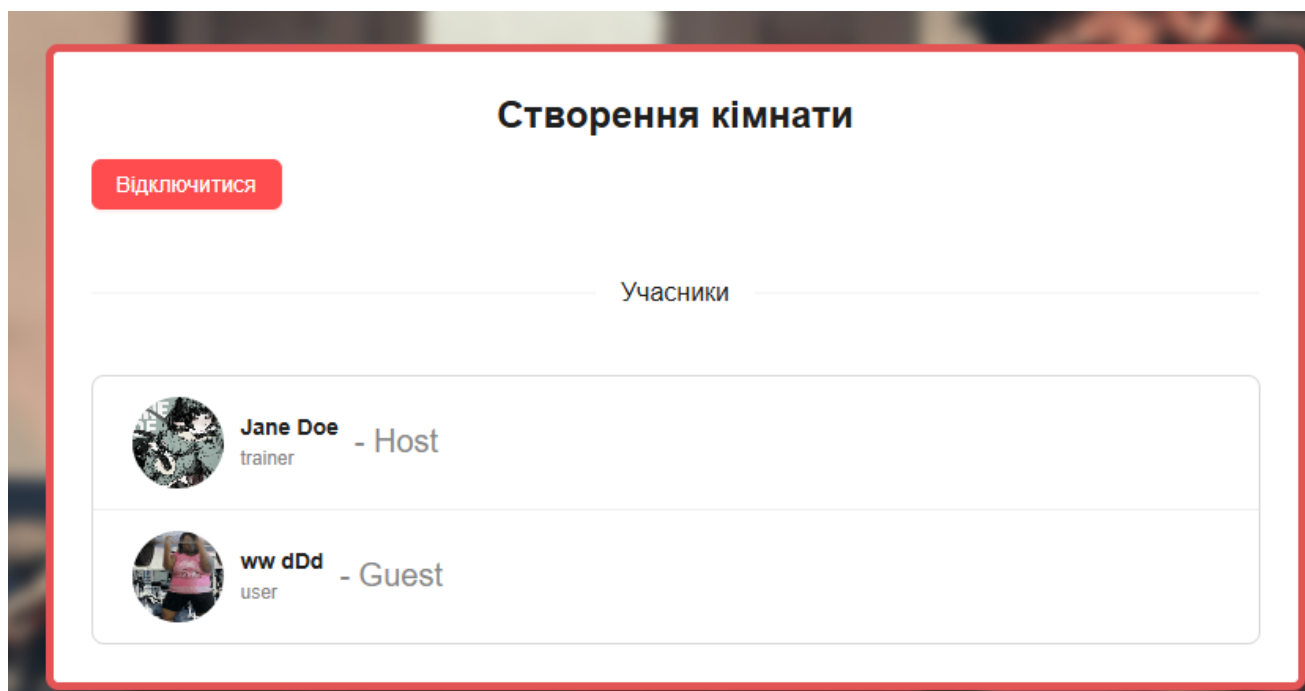


Рис. 3.65 – Приклад від гостю – підключення до кімнати
результат: Як і очікувалось гість додається без помилок та отримує поточні дані, які виводяться у список “Учасники” в обох користувачів.

Відключення від кімнати.

тестування: У разі відключення власника від кімнати, кімната закривається і всі гості втрачають підключення з повідомленнями “Кімнату закрито – власник вийшов” (рис. 3.66).

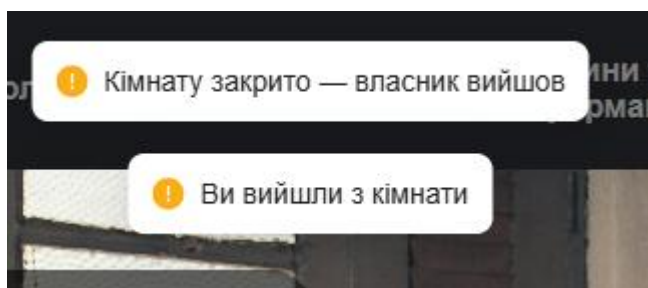


Рис. 3.66 – Повідомлення гостя при виході з кімнати

Якщо гість вийде з сторінки кімнати тоді з’явиться одне повідомлення “Ви вийшли з кімнати”, також він зникне з списку ”Учасників”. Якщо власник залишає сторінку кімнати, це еквівалентно натисканню кнопки “Відключитися”.

У разі закриття вебдодатку або браузера власником, кімната не закривається та зі списку “Учасників” власник не зникає. Підключення фактично не розривається, і якщо сторінку відкрити знову, власник повторно підключається до

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		74

кімнати та продовжить взаємодію без втрат функціональності або критичної помилки. В разі гостя, ситуація аналогічна але власник може продовжувати взаємодію з кімнатою.

результат: Як і очікувалось, якщо виходить власник – кімната закривається, якщо виходить гість – він зникає зі списку. Під час цих дій помилок не виникало. Але якщо власник або гість закриває сторінку він не зникає зі списку, це може зазначити багом.

Приєднання кількох користувачів до кімнати.

тестування: Спочатку створимо кімнату та надішлемо 4 запрошення користувачам (рис. 3.67).

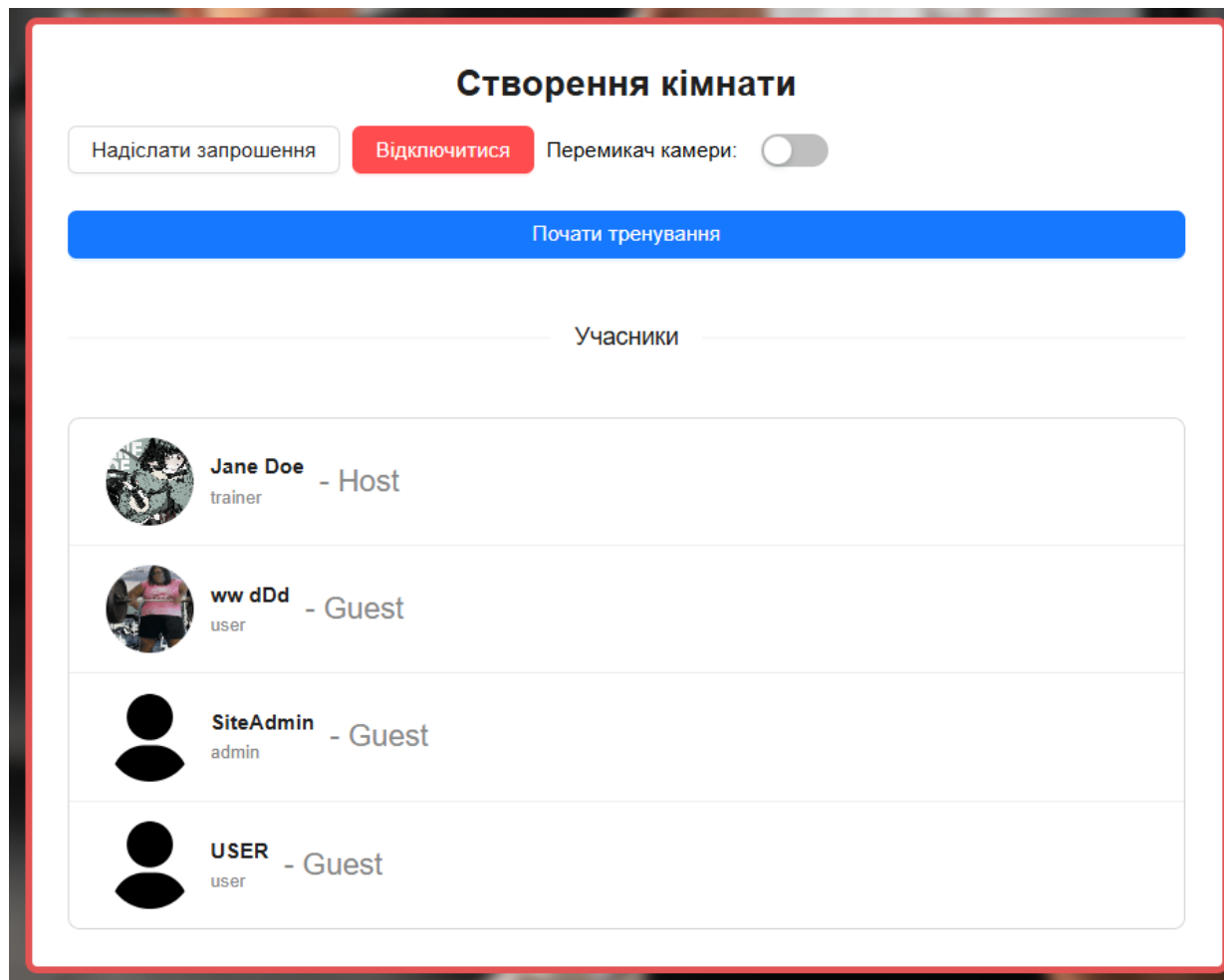


Рис. 3.67 – Декілька користувачів у кімнаті

Якщо спробувати почати тренування, тоді всіх хто знаходиться в кімнаті надішле до сторінки проходження (рис. 3.68).

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		75

Якщо власник перейде до кімнати, тоді зміни в проходженні збережуться і всіх користувачів перекине (рис. 3.69).

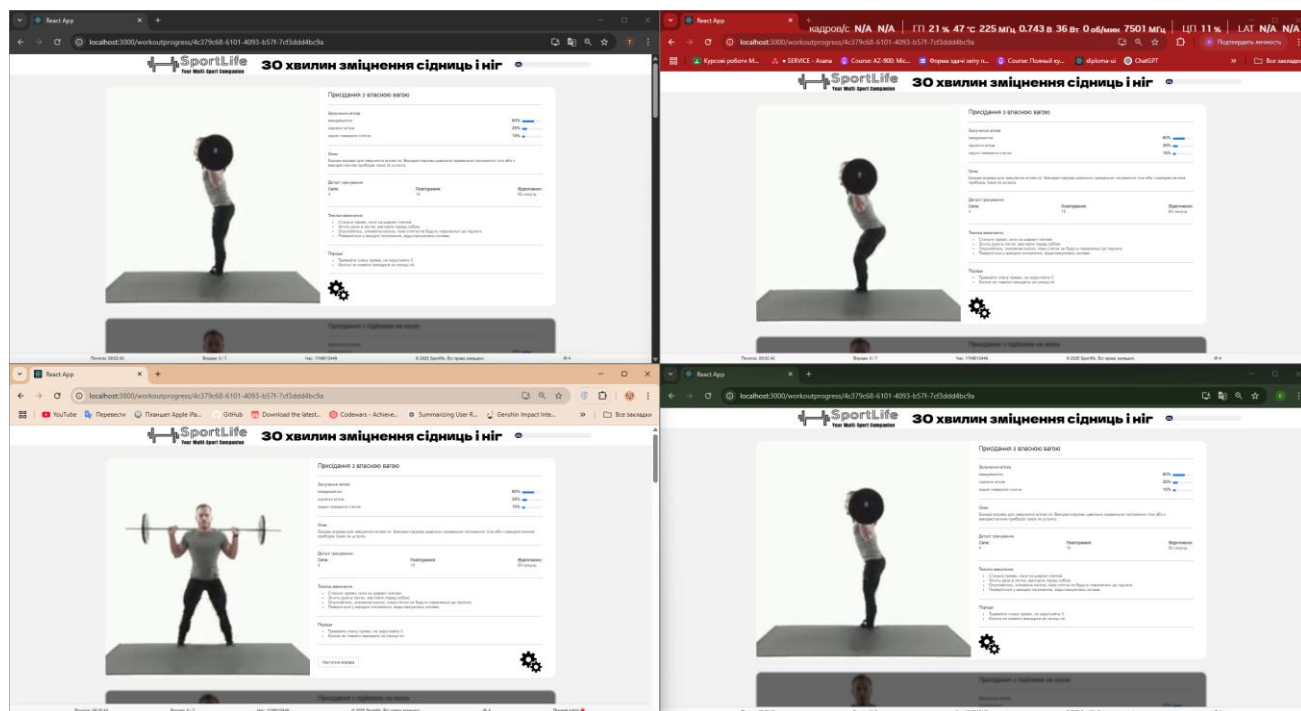


Рис. 3.68 – Приклад декількох підключень

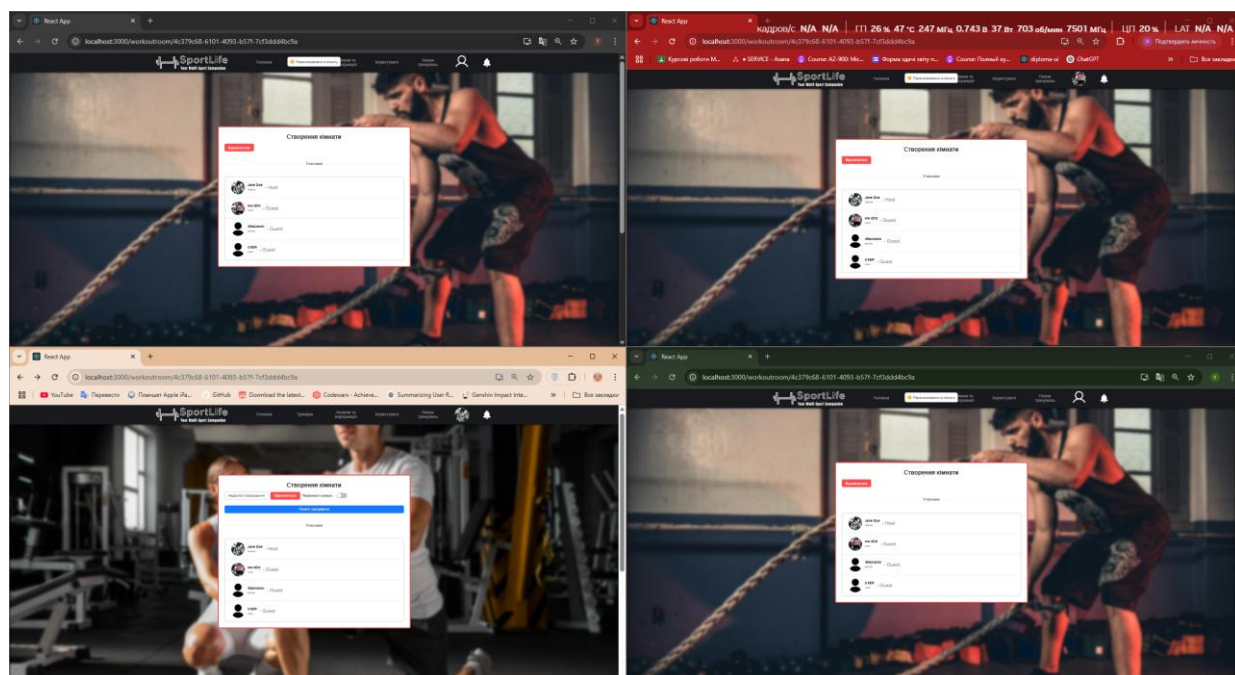


Рис. 3.69 – Приклад виходу з проходження

результат: Як і очікувалось, гості додалися до кімнати, та при початку тренування всіх перенаправило на саму сторінку. Коли власник перейшов назад до кімнати, всіх перенаправило з ним. Всі дії пройшли успішно без помилок або багів.

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк. 76
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновок до третього розділу

В данному розділі було розглянуто процес встановлення, налаштування та тестування тренувального вебдодатку, а також проаналізовано структуру його інтерфейсу. Під час розробки було розроблено діаграму розгортання яку було продемонстровано у першій частині. Додаток вимагає мінімального налаштування перед запуском, а базові параметри дозволяють швидко розгорнути додаток для тестування або роботи.

Інтерфейс вебдодатку побудований з урахуванням проаналізованих аналогів, виявлені потреби користувачів – з чіткою структурою, інформативними блоками, навігаційними меню, візуально представленими тренерами та формою підписки на новини. Структура інтерфесу тренувального додатку сприяє не лише зручності, а й підвищує загальну привабливість платформи.

У процесі тестування особливу увагу було приділено на перевірку стабільності роботи ключової функції вебдодатку, тобто підключенню користувачів до тренувальної кімнати, відключення від кімнати та перевірка на стабільність при деякої кількості користувачів в одній кімнаті. В кожній з ситуації було проведено аналіз, чіткі дії та результат в якому писалося можливі помилки або баги. Проведені ситуації оброблюються системою з урахуванням можливих помилок, а серверна логіка коректно реагує на невірні, невалідні чи неповні запити.

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		77

ВИСНОВКИ

Під час виконання кваліфікаційної роботи бакалавра було спроектовано та реалізовано інформаційний вебдодаток для дистанційних спортивних тренувань різних груп м'язів, також було проведено аналіз предметної області та порівняно з різними аналогами. На основі представлених аналогів було проаналізовано та визначено основні вимоги для створення сучасного тренувального застосунку. В процесі дослідження було розроблено архітектуру додатку основою якого складеться з сучасного стеку MERN, описані технології та їх переваги.

Проектування платформи включало аналіз варіантів використання програмного продукту, також проектування структури БД для зберігання інформації для тренувальних планів та детально описаних вправ, історії чатів, облікових записів користувачів, створення кімнат, також створення сповіщень та зберігання детальної інформації про команду тренерів, включаючи розроблення діаграми використання і ER-діаграми яка показує моделювання взаємодії між сутностями таблиць. Також було продемонстровано діаграми активності та розглянуто частину основного функціоналу, включаючи підключення до кімнати, оновлення даних для всіх користувачів кімнати, від'єднання від самої кімнати та завантаження файлів до хмарного сховища.

Також було проведено процес встановлення вебдодатку та надано діаграму розгортання. Наведено детальний опис інтерфейсу розробленого сайту, роботу між елементами додатку. Було проведено тестування кімнати планів тренувань, для виявлення можливих покращень. критичних помилок чи багів.

Отже, реалізований вебдодаток SportLife повністю відповідає поставленим цілям, це створення гнучкої, зручної та сучасної системи для дистанційного проведення тренувань, який не тільки забезпечує основний функціонал, але й має потенціал до подальшого доповнення, інтеграції нових сервісів і використання в реальних умовах спортивних клубів або індивідуальних програм підготовки.

		Левкович О.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		78

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Створення сайтів і додатків на MERN. ЧервинийХамелеон Веб Агенція. Створення та розробка сайтів. URL: <https://redchameleon.com.ua/ua/individualnaya-razrabotka/sozdanie-saytov-i-prilozheniy-na-mern/>.
2. Vivcharyk V., Ерам. Повний огляд REST: нюанси, поради, приклади. DOU. Форум. 19.09.2024. URL: <https://dou.ua/forums/topic/50364/>.
3. Myfitnesspal. URL: <https://www.myfitnesspal.com/>.
4. MERN Stack Explained. Mongodb. Resources. URL: <https://www.mongodb.com/resources/languages/mern-stack>.
5. Nike. Nike Training Club. URL: <https://www.nike.com/ntc-app>.
6. Peleton. URL: <https://www.onepeloton.com/>.
7. Cloudinary. About us. URL: <https://cloudinary.com/about>.
8. Що таке CSS. Css.in.ua. Статті. 04.09.2017. URL: https://css.in.ua/article/shcho-take-html_10.
9. Що таке html?. Css.in.ua. Статті. 19.02.2017. URL: https://css.in.ua/article/shcho-take-css_3.
10. Основи роботи з фреймворком Express.js. FoxmindEd. Блог. 31.01.2024. URL: <https://foxminded.ua/express-js/>.
11. Різниця між HTTP та HTTPS. HostIq Wiki. SSL-сертифікати. URL: <https://hostiq.ua/wiki/ukr/http-https/>.
12. Коди стану HTTP. HostIq Wiki. Хостинг. URL: <https://hostiq.ua/wiki/ukr/http-status-codes/>.
13. Що таке React JS? Як почати вивчати Реакт? Навички для react developer. Cases. Media. 11.07.2023. URL: https://cases.media/en/article/sho-take-react-js-yak-pochati-vivchati-reakt-navichki-dlya-react-developer?srsltid=AfmBOoqvnXhzej_jV_fRK-wjXlVkyhxt5fVDVFn9K-5c7KGQHJESw-4v.
14. Детальний огляд та розбір Node.js. WEZOM. Блог. 23.06.2023. URL: <https://wezom.com.ua/ua/blog/vse-chno-nuzhno-znat-o-nodejs>.

		Левкович О.О.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		79

- 15.Що таке JavaScript і для чого він потрібен. GoIT. IT Статті. 23.01.2025.
URL: <https://goit.global/ua/articles/shcho-take-javascript-i-dlia-choho-vin-potriben/>.
- 16.Розробка вебсайтів Technologies SOCKET.IO. BRANDER. Technologies.
URL: <https://brander.ua/technologies/socketio>.
- 17.Building an Internet-Connected Phone with PeerJS. Mdn web docks. WebRTC API.
URL: https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API/Build_a_phone_with_peerjs.
- 18.Що таке Visual Studio Code. Що таке Visual Studio Code. Програми. 25.02.2025. URL: <https://top-keis.com.ua/shho-take-visual-studio-code/>.
- 19.Postman – що це за інструмент і які його основні функції?. FoxmindEd. 27.05.2024. URL: <https://foxminded.ua/postman/>.
- 20.MongoDB Compass. Ukraine. MongoDB. URL: <https://www.ukraine.com.ua/wiki/mongodb/compass/>.
- 21.Erickson J. What Is the MERN Stack? Guide & Examples. Oracle. База даних. 30.01.2024. URL: <https://www.oracle.com/ua/database/mern-stack/>.
- 22.Gebben R. PeerJS. Daily Javascript. URL: <http://daily-javascript.com/articles/peerjs/>.
- 23.Андрєєв А. Що таке JSON. Усе про цей формат передачі даних в інтернеті. ApexDrive Блог. Блог. 22.07.2022. URL: <https://apix-drive.com/ua/blog/useful/scho-take-json>.
- 24.Сем В. SQL чи NoSQL – ось в чому питання. Альтернативна Наука. Програмування. URL: <https://alternativescience.net/programming/242-sql-chy-nosql-os-v-chomu-pytannya/>.
- 25.Твен С. Що таке MongoDB? вступ, Archітектура, функції та приклад. GURU99. 26.09.2024. URL: <https://www.guru99.com/uk/what-is-mongodb.html>.
- 26.JavaScript Guide. MDN Web Docs. JavaScript. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>.

		Левкович О.О.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		80

27. Getting started with React. MDN Web Docs. JavaScript frameworks and libraries. URL: https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Frameworks_libraries/React_getting_started.

28. HTTP. Node.js. Docs. URL: <https://nodejs.org/docs/latest/api/http.html>.

29. DNS. Node.js. Docs. URL: <https://nodejs.org/api/dns.html>.

30. Errors. Node.js. Docs. URL: <https://nodejs.org/api/errors.html>.

		Левкович О.О.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		81

ДОДАТКИ

		Левкович О.О.			ІІЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		82

Лістинг елементу повідомлення NotificationElement

```
import React, { useEffect, useState, useRef } from "react";
import { Avatar, Badge, Drawer, Space, Button } from "antd";
import {
  Notification,
  deleteNotification,
  setInvitedRoomId,
  trainerRequest,
} from "../function";
import axios from "axios";
import dayjs from "dayjs";
import "../style.css";
```

```
const NotificationElement = () => {
  const [open, setOpen] = useState(false);
  const [user, setUser] = useState(undefined);
  const [loading, setLoading] = useState(true);
  const [notifications, setNotifications] = useState([]);

  useEffect(() => {
    // Get current user data
    setLoading(true);
    axios
      .get("/currentuserdata")
      .then((response) => {
        setUser(response.data.user);
      })
      .catch((error) => console.error("Auth error", error))
      .finally(() => {
        setLoading(false);
      });
  }, []);

  const fetchNotifications = async () => {
    if (!user) return;
    setLoading(true);
    try {
```

		Левкович О.О.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		83

```

const response = await axios.get(`/allnotifications/${user._id}`);
setNotifications(response.data);
} catch (error) {
  console.error("Notification error", error);
} finally {
  setLoading(false);
}
};

useEffect(() => {
  if (!user) return;

  const intervalId = setInterval(async () => {
    try {
      const res = await axios.get(`/notification/${user._id}`);
      const data = res.data;

      if (data && data.title) {
        await axios.put(`/notification/${user._id}`, {
          notificationId: data._id,
        });
        Notification(
          data.message,
          data.title,
          data.type,
          data._id,
          data.url,
          data?.action
        );
      }
    } catch (error) {
      console.error("Notification error", error);
    }
  }, 5000);

  return () => clearInterval(intervalId);
}, [user]);

```

		Левкович О.О.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		84

```

const showDrawer = () => {
  fetchNotifications();
  setOpen(true);
};

const onClose = () => {
  setOpen(false);
};

return (
  <>
    <Avatar
      size={40}
      src="/img-pack/icons/bell.png"
      onClick={showDrawer}
      className="notification"
    />
    <Drawer
      onClose={onClose}
      closable={false}
      open={open}
      loading={loading}
      width={640}
      bodyStyle={{ padding: 0, backgroundColor: "rgb(24, 26, 29)" }}
    >
      {notifications.map((notification) => (
        <div key={notification._id} className="notification-item">
          <Avatar
            size={60}
            src={notification.avatar || "/img-pack/icons/user.png"}
          />
          <div className="notification-content">
            <div className="notification-header">
              <h2>{notification.title}</h2>
              <div>{`${dayjs(notification.date).format("HH:mm")}` | `${dayjs(
                notification.date
              )}.format("DD.MM")}`}</div>
            </div>

```

		Левкович О.О.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		85

```

<p
  style={{
    overflowWrap: "break-word",
    wordBreak: "break-word",
    whiteSpace: "normal",
  }}
>
  {notification.message}
</p>

{notification?.action && (
  <Space>
    <Button
      type="link"
      size="small"
      onClick={() => {
        if (notification?.action === "personalTrainerRequest")
          trainerRequest(
            notification?.fromWho,
            notification?.access,
            "reject"
          );
        setOpen(false);
        setLoading(true);

        fetchNotifications();
        deleteNotification(notification?._id);
      }}
    >
      Відхилити
    </Button>
    <Button
      type="primary"
      size="small"
      onClick={() => {
        if (notification?.action === "roomRequest")
          setInvitedRoomId(notification?.url);
        else if (
          notification?.action === "personalTrainerRequest"

```

		Левкович О.О.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		86

```

    )
    trainerRequest(
      notification?.fromWho,
      notification?.access,
      "accept"
    );
    console.log(notification);
    fetchNotifications();
    deleteNotification(notification?._id);
  }}
>
  Підтвердити
</Button>
</Space>
)}
</div>
</div>
)}}
</Drawer>
</>
);
};

```

```
export default NotificationElement;
```

Додаток Б

Лістинг коду POST запиту посилення листів “/sendemail”

```

app.post("/sendemail", upload.array("files"), async (req, res) => {
  const { id, email, subject, note, role } = req.body;
  const files = req.files;
  const userData = JSON.parse(req.body.userData);

  if (!email || !subject || !note || !id || !files || files.length === 0) {
    return res.status(400).json({ message: "All fields are required" });
  }

  try {
    const attachments = files.map((file, index) => ({
      filename: file.originalname || `file-${index}`,

```

		Левкович О.О.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		87

```

content: file.buffer,
contentType: file.mimetype,
});

```

```

const baseUrl = process.env.BASE_SERVER_URL || "http://localhost:5000";
const linkAccept = `${baseUrl}/acceptchangerequest/${id}/${role}`;
const linkReject = `${baseUrl}/rejectchangerequest/${id}`;

```

```

const htmlContent = `
<div style="font-family: Arial, sans-serif; background-color: #f9f9f9; padding: 40px 20px;">
  <div style="
    max-width: 600px;
    margin: 0 auto;
    background: #ffffff;
    padding: 36px 40px;
    border-radius: 12px;
    box-shadow: 0 8px 24px rgba(0,0,0,0.1);
    color: #333333;
    line-height: 1.6;
    font-size: 16px;
  ">
    <p style="margin-bottom: 24px;">Шановна компанія <strong>SportLife</strong>,</p>

    <p style="margin-bottom: 24px;">
      Прошу розглянути мою заявку на зміну ролі в веб-додатку для спортивних тренувань.
    </p>

    <p style="margin-bottom: 16px;">
      <strong>Поточна роль:</strong> ${userData?.role}<br/>
      <strong>Бажана роль:</strong> ${role}
    </p>

    <div style="
      margin: 24px 0;
      padding: 20px 24px;
      background-color: #f1f5ff;
      border-left: 5px solid #2c7be5;
      border-radius: 6px;

```

		Левкович О.О.			ІІЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		88

```

color: #1a1a1a;
font-size: 16px;
line-height: 1.5;
white-space: pre-line;
">
<strong>Причина зміни полі:</strong><br/>
${note}
</div>

<p style="margin: 24px 0 16px 0;">
<strong>Дані користувача:</strong><br/>
- ПІБ: ${userData?.name} ${userData?.last_name}<br/>
- Нікнейм: ${userData?.username}<br/>
- Пошта: <a href="mailto:${
  userData?.email
}" style="color:#2c7be5; text-decoration:none;">${
userData?.email
}</a><br/>
- Контактний телефон: <a href="tel:${
  userData?.phone
}" style="color:#2c7be5; text-decoration:none;">${userData?.phone}</a>
</p>

<p style="margin-bottom: 36px;">
Дякую за розгляд моєї заявки.
</p>

<p style="margin-bottom: 0;">
З повагою,<br/>
<strong>${userData?.name} ${userData?.last_name}</strong><br/>
<a href="mailto:${
  userData?.email
}" style="color:#2c7be5; text-decoration:none;">${
userData?.email
}</a><br/>
<a href="tel:${
  userData?.phone
}" style="color:#2c7be5; text-decoration:none;">${userData?.phone}</a>

```

		Левкович О.О.			ІІЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		89

</p>

<div style="display: flex; gap: 10px; margin-top: 36px;">

<a href="{linkAccept}" style="

flex: 1;

text-align: center;

background-color: #28a745;

color: white !important;

text-decoration: none;

padding: 14px 15px;

font-weight: 600;

font-size: 16px;

border-radius: 8px;

box-shadow: 0 4px 12px rgba(40, 167, 69, 0.5);

display: inline-block;

transition: background-color 0.3s ease;

padding-inline: 15px;

margin-right: 10px;

" onmouseover="this.style.backgroundColor='#218838'"

onmouseout="this.style.backgroundColor='#28a745'">

 Підтвердити

{ " " }

<a href="{linkReject}" style="

flex: 1;

text-align: center;

background-color: #dc3545;

color: white !important;

text-decoration: none;

padding: 14px 15px;

font-weight: 600;

font-size: 16px;

border-radius: 8px;

box-shadow: 0 4px 12px rgba(220, 53, 69, 0.5);

display: inline-block;

transition: background-color 0.3s ease;

padding-inline: 15px;

		Левкович О.О.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		90

```
" onmouseover="this.style.backgroundColor='#c82333"
onmouseout="this.style.backgroundColor='#dc3545">
```

✗ Відхилити

```
</a>
```

```
</div>
```

```
<p style="font-size: 14px; color: #999999; margin-top: 48px; text-align: center;">
```

Якщо ви не надсилали цей запит, просто проігноруйте цей лист.

```
</p>
```

```
</div>
```

```
</div>
```

```
`;
```

```
await transporter.sendMail({
  from: `"SportLife" ${email}`,
  to: "sportlife.corporate.mail@gmail.com",
  subject: `Підтвердження`,
  text: note,
  html: htmlContent,
  replyTo: email,
  attachments,
});
```

```
// const findRequest = await Request.findOne({ userId: id });
// if (!findRequest) {
const user = await User.findById(id);
const newRequest = new Request({
  email: email,
  userId: user._id,
  requestReason: role,
  status: "pending",
});
```

```
if (role === "trainer") user.trainerRequestId = newRequest._id; // Save request id to user
await user.save(); // Save user with updated request id
await newRequest.save();
// }
```

		Левкович О.О.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		91

```

    res.json({ success: true, message: "Email sent successfully" });
  } catch (err) {
    console.error(err);
    res.status(500).json({ success: false, message: "Error sending mail" });
  }
});

```

Додаток В

ЛІСТИНГ коду методу joinRoom

```

socket.on("joinRoom", async ({ roomId, userId, peerId }) => {
  socket.join(roomId);
  const existingRoom = await Room.findOne({ roomId });
  if (!existingRoom) {
    const newRoom = new Room({
      roomId: roomId,
      createdAt: new Date(),
      data: { "": "1" },
      users: [userId],
      owner: userId,
    });
    await newRoom.save();
    const count = getRoomUserCount(roomId);
    io.to(roomId).emit("room-user-count", count);
    io.to(newRoom.roomId).emit("chatHistory", newRoom.users);
    io.to(newRoom.roomId).emit("roomOwner", newRoom.owner);
    socket.emit("receiveData", newRoom.data);
  } else {
    if (!existingRoom.users.includes(userId)) {
      existingRoom.users.push(userId);
      await existingRoom.save();
    }
    io.to(existingRoom.roomId).emit("chatHistory", existingRoom.users);
    io.to(existingRoom.roomId).emit("roomOwner", existingRoom.owner);
    socket.emit("receiveData", existingRoom.data);
  }
});

```

		Левкович О.О.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		92

Лістинг коду методу оновлення даних updateData

```

socket.on( "updateData", async ({ roomId, userId, data, currentWorkout, startTime = null,
finalTimeResult = null, status }) => {
    const existingRoom = await Room.findOne({ roomId });
    if (!existingRoom || !data) return;
    if (!existingRoom.owner.toString() === userId) {
        return;
    }
    existingRoom.data["exercises"] = data;
    existingRoom.data["startTime"] = startTime;
    existingRoom.data["finalTimeResult"] = finalTimeResult;
    existingRoom.data["status"] = status;
    existingRoom.data["currentWorkout"] = currentWorkout;
    existingRoom.markModified("data");
    await existingRoom.save();

    const count = getRoomUserCount(roomId);
    io.to(roomId).emit("room-user-count", count);
    io.to(roomId).emit("receiveData", existingRoom.data);
}
);

```

Лістинг коду методу для від'єднання disconnectData

```

socket.on("disconnectData", async ({ roomId, userId }) => {
    const existingRoom = await Room.findOne({ roomId });

    if (!existingRoom) return;

    if (existingRoom.owner.toString() === userId) {
        await Room.deleteOne({ roomId });

        console.log("owner disconnected, room deleted");
    }
}
);

```

		Левкович О.О.			ІІЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		93

```

    socket.broadcast.to(roomId).emit("roomClosed");

    return;
}
existingRoom.users = existingRoom.users.filter((user) => user !== userId);
await existingRoom.save();
const count = getRoomUserCount(roomId);
io.to(roomId).emit("room-user-count", count);

io.to(roomId).emit("chatHistory", existingRoom.users);

if (existingRoom.users.length === 0) {
    await Room.deleteOne({ roomId });
}
// * find and delete user from room if he disconnected
});

```

Додаток Д

ЛІСТИНГ КОДУ ДЛЯ ЗАВАНТАЖЕННЯ У ХМАРНЕ СХОВИЩЕ uploadFileToCloudinary

```

export const uploadFileToCloudinary = async (file) => {
    const formData = new FormData();
    // * Set data as file name
    const currentDate = Date.now();

    formData.append("file", file);
    formData.append("upload_preset", "SPORTLIFE");
    formData.append("public_id", `${currentDate}`);

    try {
        const response = await fetch(
            "https://api.cloudinary.com/v1_1/dnrRaamqm/image/upload",
            {
                method: "POST",
                body: formData,
            }
        );
        const data = await response.json();
    }
}

```

		Левкович О.О.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		94

```

if (data.secure_url) {
    return data.secure_url;
}
throw new Error("Error uploading image");
} catch (error) {
    console.error("Error uploading image:", error);
    throw error;
}
};

```

		Левкович О.О.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		95