ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І. І. МЕЧНИКОВА
(повне найменування закладу вищої освіти)

Факультет математики, фізики та інформаційних технологій
(повне найменування факультету)

Кафедра механіки, автоматизації та інформаційних технологій
(повна назва кафедри)

Кваліфікаційна робота

на здобуття ступеня вищої освіти «бакалавр»

«Розробка веб-платформи для планування та моніторингу тренувального процесу. Аналітичний модуль тренера для управління тренувальними програмами»
(тема кваліфікаційної роботи українською мовою)
«Development of a web platform for training planning and monitoring. Analytical module for coaches to manage training programs»
(тема кваліфікаційної роботи англійською мовою)

Виконав: здобувач денної форми навчання
спеціальності 122 Комп'ютерні науки
Освітня програма Комп'ютерні науки
Могилевський Тимур Олексійович
(прізвище, ім'я, по-батькові здобувача)

Керівник к.т.н., доцент Іщенко О.В.
(науковий ступінь, вчене звання, прізвище та ініціали, підпис)

Рецензент к.ф.-м.н., доцент Рачинська А.Л.
(науковий ступінь, вчене звання, прізвище та ініціали)

Рекомендовано до захисту:

Протокол засідання кафедри №___ від «___»_____ 2026 р.

Завідувач кафедри

_____ _Алла РАЧИНСЬКА_
(підпис) (прізвище, ініціали)

Захищено на засіданні ЕК № ___ протокол № __ від «___» ____ 2026 р.
Оцінка _____/_____/_____
(за національною шкалою, шкалою ECTS, бали)

Голова ЕК

_____ _Микола МАЛАКСІАНО _
(підпис) (прізвище, ініціали)

Одеса - 2026

АНОТАЦІЯ

Кваліфікаційну роботу присвячено розробці веб-платформи для планування, виконання та моніторингу тренувального процесу з акцентом на аналітичний модуль тренера. У роботі розглянуто проблему фрагментованого збирання тренувальних даних, коли план тренування, фактичне виконання, показники активного часу, калорійності, повторень, ваги, дистанції та стану користувача зберігаються в різних середовищах і не дають тренеру швидкої картини прогресу спортсмена. Запропонована платформа поєднує створення шаблонів програм, призначення тренувань, фіксацію результатів вправ і формування аналітичних показників для ухвалення тренером рішень щодо корекції програми.

Практична цінність роботи полягає в тому, що система не обмежується календарем або довідником вправ, а формує основу для аналізу даних: порівняння планових і фактичних показників, оцінювання регулярності тренувань, візуалізацію навантаження за періодами та виявлення відхилень від програми. У реалізованому проєкті використано стек Next.js, React, PostgreSQL, серверні API-маршрути та бібліотеку Chart.js для побудови статистичних візуалізацій.

ABSTRACT

This thesis is devoted to the development of a web platform for planning, executing, and monitoring the training process, with a focus on the coach's analytical module. The thesis addresses the problem of fragmented training data collection, where the training plan, actual performance, metrics such as active time, calories burned, repetitions, weight, distance, and the user's condition are stored in different environments and do not provide the coach with a quick overview of the athlete's progress. The proposed platform combines the creation of program templates, the assignment of workouts, the recording of exercise results, and the generation of analytical metrics to enable the coach to make decisions regarding program adjustments.

The practical value of this work lies in the fact that the system is not limited to a calendar or exercise guide, but forms the basis for data analysis: comparing planned and actual metrics, assessing training regularity, visualizing workload by period, and identifying deviations from the program. The implemented project uses the Next.js stack, React, PostgreSQL, server-side API routes, and the Chart.js library to build statistical visualizations.

ЗМІСТ

ВСТУП 4

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 7

1.1 Особливості організації тренувального процесу 7

1.2 Роль тренера у плануванні та моніторингу тренувань 9

1.3 Аналіз потреб спортсменів і тренерів у веб-платформі 11

1.4 Огляд існуючих рішень для планування тренувань і спортивної аналітики 14

1.5 Аналіз тренувальних даних як основи для прийняття рішень тренером .. 16

1.6 Вимоги до аналітичного модуля тренера 18

2 ПРОЄКТУВАННЯ АРХІТЕКТУРИ СИСТЕМИ 20

2.1 Загальна архітектура веб-платформи 20

2.2 Системний аналіз веб-платформи 21

2.3 Проєктування ролей користувачів: спортсмен, тренер, адміністратор 25

2.4 Проєктування модуля управління тренувальними програмами 26

2.5 Проєктування модуля планування тренувань 28

2.6 Проєктування модуля моніторингу виконання тренувань 29

2.7 Проєктування бази даних тренувального процесу 30

2.8 Проєктування потоків даних між клієнтською частиною, сервером і БД 33

2.9 Проєктування аналітичного модуля тренера 35

3 РЕАЛІЗАЦІЯ ТА РЕЗУЛЬТАТИ РОБОТИ 38

3.1 Реалізація веб-платформи та її основних модулів 38

3.2 Реалізація авторизації та рольового доступу користувачів 39

3.3 Реалізація створення та редагування тренувальних програм 42

3.4 Реалізація планування тренувань у календарі 44
3.5 Результати роботи програмного продукту та приклади екранних форм .. 48
4 ТЕСТУВАННЯ СИСТЕМИ 50
4.1 Мета, підхід і критерії тестування системи 50
4.2 Тестування сценаріїв авторизації та роботи з профілем користувача 51
4.3 Тестування створення тренувань 56
4.4 Тестування календаря тренувань 60
4.5 Тестування аналітичного модуля 62
ВИСНОВОК 65
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 66
ДОДАТОК А. ДОВІДКА ПРО ВПРОВАДЖЕННЯ 68

ВСТУП

Сучасний тренувальний процес дедалі більше залежить від даних. Для тренера вже недостатньо лише скласти програму вправ і контролювати факт відвідування занять: необхідно бачити, як спортсмен виконує заплановані блоки, які вправи дають стабільний прогрес, де виникає перевантаження, які показники змінюються з тижня в тиждень і які рішення потрібно прийняти для корекції програми. Особливо актуальним це стає для індивідуальних і малих групових занять, де тренер одночасно веде декількох спортсменів, а дані про тренування часто залишаються у нотатках, таблицях, месенджерах або різних мобільних застосунках.

Актуальність теми полягає у необхідності створення єдиного програмного середовища, яке поєднує планування тренувань, фіксацію фактичних результатів і аналітичну підтримку тренера. Тренувальна програма має розглядатися не як статичний список вправ, а як керований процес, у якому кожне виконане тренування поповнює базу даних і може бути використане для подальшого аналізу. Такий підхід дозволяє оцінювати не тільки факт виконання, а й якість навантаження: активний час, загальний час, калорійність, обсяг силових вправ, параметри кардіонавантаження, регулярність занять, динаміку результатів та відповідність фактичного виконання плану.

Розроблювана веб-платформа орієнтована на взаємодію тренера і спортсмена. Тренер створює шаблони програм, формує блоки та вправи, призначає тренування на конкретну дату і конкретному користувачу. Спортсмен виконує тренування в інтерфейсі системи, а під час виконання фіксуються результати окремих вправ та підсумкові дані тренування. На основі цих даних формується аналітичний шар, який має стати головним фокусом захисту роботи: саме він показує, що платформа є не лише інструментом планування, а й системою підтримки тренерських рішень.

Метою роботи є оптимізація комунікації між тренером і клієнтами-спортсменами та систематизація статистики тренувального процесу, аналізу результатів тренувань та відстеження динаміки прогресу спортсменів.

Для досягнення поставленої мети необхідно вирішити такі задачі:

1. Проаналізувати предметну область, типові сценарії роботи тренера та спортсмена, а також роль даних у керуванні тренувальними програмами.
2. Дослідити існуючі програмні рішення для планування, моніторингу та аналізу тренувань і визначити їхні сильні сторони та обмеження з позиції тренера.
3. Спроєктувати структуру даних платформи, яка дозволяє пов’язати користувача, тренувальну програму, блоки, вправи, розклад і фактичні результати.
4. Реалізувати основні функціональні модулі веб-платформи: автентифікацію, ролі користувачів, створення програм, призначення тренувань, виконання тренування та збереження результатів.
5. Розробити аналітичний модуль тренера, який формує показники активності, регулярності, план-факт виконання та динаміки навантаження.
6. Забезпечити візуальне подання тренувальних даних у формі графіків, таблиць та агрегованих індикаторів, придатних для швидкого прийняття рішень.
7. Перевірити коректність роботи системи на прикладах створення програм, виконання тренувань і відображення статистики.

У роботі використано методи системного аналізу, моделювання даних, проєктування реляційних баз даних, об’єктно-орієнтованого та компонентного підходу до розробки веб-застосунків, а також методи агрегування та візуалізації даних. Для реалізації використано Next.js і React для клієнтської частини, API-маршрути для серверної логіки, PostgreSQL для зберігання даних і Chart.js для побудови статистичних графіків.

Практичне значення роботи полягає у можливості використання платформи як основи для супроводу тренувального процесу. Система може допомогти тренеру швидше оцінювати стан виконання програм, знаходити проблемні місця у навантаженні, бачити динаміку активності спортсменів і приймати рішення на основі фактичних даних, а не лише суб’єктивних спостережень.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Особливості організації тренувального процесу

Тренувальний процес є систематизованою та цілеспрямованою діяльністю, спрямованою на розвиток фізичних якостей спортсмена, підвищення рівня його працездатності, формування правильної техніки виконання вправ, покращення функціонального стану організму та досягнення визначених спортивних або оздоровчих цілей. Він охоплює не лише безпосереднє виконання фізичних вправ, а й планування, контроль, аналіз результатів та корекцію тренувальної програми відповідно до стану й прогресу спортсмена.

Ефективність тренувального процесу залежить від багатьох чинників. До них належать індивідуальні особливості спортсмена, рівень його фізичної підготовки, поставлена мета, регулярність занять, правильність добору вправ, адекватність навантаження, якість відновлення та своєчасність аналізу результатів. Тренування не може розглядатися лише як набір окремих вправ, оскільки для досягнення стабільного результату необхідна цілісна система, у якій кожне заняття є частиною загальної програми підготовки.

Організація тренувального процесу передбачає послідовне виконання кількох етапів. Спочатку визначається мета тренувань: розвиток сили, витривалості, гнучкості, зниження маси тіла, набір м’язової маси, підтримання загальної фізичної форми або підготовка до змагань. Після цього тренер добирає відповідні вправи, встановлює параметри навантаження, формує розклад занять, визначає тривалість тренувального циклу та способи контролю результатів. У процесі виконання програми фіксуються фактичні показники тренувань, які в подальшому використовуються для аналізу динаміки та прийняття рішень щодо корекції навантаження.

До основних параметрів тренування належать частота занять, інтенсивність, тривалість, тип вправ, кількість підходів і повторень, робоча вага, час відпочинку між підходами, загальний обсяг навантаження та темп

прогресії. Кожен із цих параметрів впливає на результат тренувального процесу. Сучасні рекомендації з тестування та призначення вправ мають спиратися на доказові стандарти й актуальні практики [2].

Важливою особливістю тренувального процесу є його циклічність. Тренувальна програма не є сталою структурою, яка залишається незмінною протягом усього періоду підготовки. Вона повинна адаптуватися до поточного стану спортсмена, рівня його підготовленості, реакції організму на навантаження, регулярності відвідування занять та отриманих результатів. Якщо спортсмен демонструє стабільний прогрес, тренер може поступово збільшувати складність програми, підвищувати інтенсивність, додавати нові вправи або змінювати структуру тренувань. Для подальшої адаптації до конкретних тренувальних цілей необхідні прогресивні протоколи силового тренування [4]. Якщо ж результати погіршуються, спортсмен пропускає заняття або виникають ознаки перевантаження, програма потребує перегляду.

Окрему роль в організації тренувального процесу відіграє принцип індивідуалізації. Навіть за однакової мети різні спортсмени можуть потребувати різного підходу до побудови програми. Це пов’язано з відмінностями у фізичній підготовці, віці, стані здоров’я, попередньому досвіді тренувань, режимі дня та доступному обладнанні. Тому ефективна система організації тренувань повинна дозволяти тренеру налаштовувати програми під конкретного спортсмена, а не використовувати універсальні шаблони без урахування індивідуальних особливостей.

Ще однією важливою складовою є контроль виконання тренувань. План тренування має практичну цінність лише тоді, коли тренер може оцінити, наскільки спортсмен його виконує. Для цього необхідно фіксувати завершені тренування, виконані вправи, кількість підходів і повторень, використану вагу, активний час, суб’єктивні відчуття спортсмена та інші показники. Без систематичного збору цих даних тренер фактично не має повної картини тренувального процесу і змушений приймати рішення на основі неповної або неструктурованої інформації.

У традиційному підході значна частина інформації про тренування може зберігатися у паперових щоденниках, електронних таблицях, нотатках або неструктурованих повідомленнях між тренером і спортсменом. Такий спосіб ведення даних має низку недоліків. Інформація може дублюватися, втрачатися або бути неповною. Тренеру складно швидко порівнювати результати за різні періоди, оцінювати динаміку прогресу, виявляти пропуски занять і визначати, які саме елементи програми потребують корекції. Крім того, при роботі з кількома спортсменами навантаження на тренера значно зростає, оскільки кожного підопічного потрібно контролювати окремо.

1.2 Роль тренера у плануванні та моніторингу тренувань

Тренер є ключовою фігурою в організації тренувального процесу, оскільки саме він відповідає за постановку цілей, побудову тренувальної програми, добір вправ, визначення навантаження, контроль виконання та аналіз результатів спортсмена. Його діяльність не обмежується лише складанням плану тренувань. Тренер виконує роль організатора, аналітика, консультанта та координатора, який супроводжує спортсмена на всіх етапах тренувального процесу.

На етапі планування тренер визначає структуру програми відповідно до мети спортсмена, рівня його фізичної підготовки, доступного обладнання, графіка занять, стану здоров’я та індивідуальних обмежень. Наприклад, для спортсмена, який прагне збільшити м’язову масу, програма може містити більший акцент на силові вправи, поступове збільшення робочої ваги та контроль тренувального обсягу. Для клієнта, метою якого є покращення загальної фізичної форми або зниження маси тіла, важливими можуть бути регулярність занять, поєднання силових і кардіонавантажень, активний час тренування та загальна витрата енергії.

Тренувальна програма може містити різні блоки вправ, кількість підходів, повторень, робочу вагу, тривалість виконання, час відпочинку, тип

активності, рівень інтенсивності та інші параметри. Для тренера важливо мати можливість швидко створювати нові програми, редагувати вже існуючі, використовувати шаблони, адаптувати їх під конкретного спортсмена та призначати тренування на певні дні. Це особливо актуально у випадках, коли тренер працює не з одним клієнтом, а з групою спортсменів, кожен із яких має власну мету, графік і рівень підготовки.

Окрім планування, важливою функцією тренера є моніторинг виконання тренувань. Після призначення програми тренеру необхідно розуміти, чи виконує спортсмен заплановані заняття, які результати він показує, чи дотримується визначеної частоти тренувань і чи не виникають ознаки перевантаження або зниження мотивації. Для цього аналізуються такі показники, як кількість завершених тренувань, активний час, обсяг виконаної роботи, спалені калорії, регулярність занять, тренувальна частота, виконання плану та баланс навантаження на різні групи м’язів.

Моніторинг дозволяє тренеру бачити не лише окреме тренування, а й загальну тенденцію розвитку спортсмена. Наприклад, якщо спортсмен протягом кількох тижнів стабільно виконує програму і поступово покращує результати, це може свідчити про правильність обраного навантаження. Якщо ж спостерігається зниження активності, часті пропуски занять або падіння тренувального обсягу, тренер може зробити висновок про необхідність перегляду програми, зміни інтенсивності або додаткової комунікації зі спортсменом.

Важливим аспектом роботи тренера є своєчасна корекція тренувального плану. Здоровим дорослим рекомендовано виконувати прогресивне силове тренування для покращення м’язової функції, розміру м’язів і фізичної працездатності [3]. Тренувальна програма повинна змінюватися відповідно до фактичних результатів спортсмена. Якщо навантаження є занадто легким, воно не забезпечує достатнього стимулу для розвитку. Якщо ж навантаження є надмірним, це може призвести до перевтоми, зниження результативності або втрати мотивації. Тренувальне навантаження пов’язане з ризиком травм,

захворювань і болючості, тому воно потребує систематичного контролю [5]. Саме тому тренер повинен мати доступ до об’єктивних даних, які дозволяють оцінити ефективність програми і прийняти обґрунтоване рішення щодо її зміни.

Окреме значення має комунікація між тренером і спортсменом. У процесі тренувань спортсмен може мати запитання щодо техніки виконання вправ, самопочуття, складності програми або необхідності зміни навантаження. Тренеру, зі свого боку, важливо отримувати від спортсмена зворотний зв’язок і швидко реагувати на нього. Якщо комунікація відбувається через різні канали — месенджери, електронну пошту, усні домовленості або окремі таблиці, — частина інформації може втрачатися. Наявність єдиної платформи дозволяє зробити взаємодію більш структурованою та пов’язаною безпосередньо з тренувальними даними.

1.3 Аналіз потреб спортсменів і тренерів у веб-платформі

Потреби спортсменів і тренерів у веб-платформі мають різну спрямованість, однак вони пов’язані спільною метою — зробити тренувальний процес більш структурованим, контрольованим і результативним. Спортсмен зацікавлений у зручному доступі до тренувальної програми, зрозумілому відображенні вправ і можливості фіксувати власні результати. Тренер, у свою чергу, потребує інструментів для створення програм, керування спортсменами, моніторингу виконання тренувань, аналізу статистики та корекції навантаження.

Для спортсмена важливо, щоб система була простою та зрозумілою у використанні. Після авторизації він повинен мати доступ до призначених тренувань, календаря занять, опису вправ, кількості підходів, повторень, робочої ваги або часу виконання. Під час тренування система має допомагати спортсмену послідовно проходити програму, відображати поточну вправу, показувати прогрес виконання та надавати можливість вносити фактичні

результати. Це дозволяє зменшити залежність від паперових записів, окремих файлів або повідомлень від тренера.

Важливою потребою спортсмена є можливість бачити власний прогрес. Користувач повинен мати доступ до історії виконаних тренувань, основних показників активності, кількості завершених занять, активного часу, обсягу навантаження та інших результатів. Візуалізація прогресу підвищує мотивацію, оскільки спортсмен може бачити зміни у своїх показниках і краще розуміти, як його дії впливають на досягнення поставленої мети. Крім того, наявність історії тренувань дозволяє спортсмену більш усвідомлено ставитися до процесу підготовки.

Для тренера основною потребою є ефективне керування тренувальними програмами. Система повинна надавати можливість створювати вправи, формувати тренувальні плани, використовувати шаблони, призначати програми конкретним спортсменам і редагувати їх у разі потреби. При цьому важливо, щоб тренер міг швидко адаптувати програму під індивідуальні особливості спортсмена, не витрачаючи багато часу на повторне створення однотипних тренувань.

Не менш важливою є потреба тренера в моніторингу виконання тренувань. Тренер повинен бачити, які тренування були виконані, які були пропущені, скільки часу спортсмен витратив на заняття, який обсяг роботи виконав і як змінюються його показники з часом. Наявність такої інформації дозволяє оперативно реагувати на зміни у тренувальному процесі. Наприклад, якщо спортсмен регулярно пропускає заняття, тренер може змінити графік або провести додаткову консультацію. Якщо спортсмен виконує програму, але не демонструє прогресу, це може бути підставою для зміни навантаження або структури вправ.

Особливо важливою для тренера є аналітична складова платформи. Простого збереження інформації про тренування недостатньо, оскільки велика кількість даних сама по собі не забезпечує ефективного управління процесом. Система повинна перетворювати ці дані на зрозумілі показники: частоту

тренувань, регулярність, активний час, сумарний тренувальний обсяг, виконання плану, динаміку результатів і баланс навантаження на різні групи м’язів. Такі показники допомагають тренеру швидко оцінити ефективність програми та приймати рішення на основі фактичних даних, а не лише суб’єктивних вражень.

Окремою потребою обох сторін є покращення комунікації. У традиційному підході тренер і спортсмен часто обмінюються інформацією через різні канали: месенджери, телефонні дзвінки, електронну пошту або окремі документи. Це може призводити до втрати важливих даних, непорозумінь або складнощів у відстеженні змін програми. Веб-платформа дозволяє об’єднати комунікацію з тренувальним процесом, щоб обговорення, результати та зміни програми були пов’язані між собою в межах однієї системи.

Також важливою вимогою до веб-платформи є централізоване зберігання даних. Уся інформація про спортсменів, програми, тренування, вправи, результати та статистику повинна зберігатися в єдиній системі. Це спрощує доступ до даних, зменшує ризик їх втрати та дозволяє швидко формувати загальну картину тренувального процесу. Для тренера це особливо важливо при роботі з кількома клієнтами, оскільки платформа дозволяє структурувати інформацію і швидко переходити від даних одного спортсмена до іншого.

З погляду функціональності, веб-платформа повинна поєднувати кілька основних модулів: модуль керування користувачами, модуль створення тренувальних програм, календар тренувань, інтерфейс виконання тренування, систему фіксації результатів, аналітичний модуль і засоби комунікації. Така структура дозволяє забезпечити повний цикл роботи: від створення програми до оцінки її ефективності.

1.4 Огляд існуючих рішень для планування тренувань і спортивної аналітики

На ринку існує низка програмних рішень для організації тренувального процесу, серед яких можна виділити ABC Trainerize, TrainHeroic, TrueCoach, Everfit та Garmin Connect. Ці системи мають спільну мету - спростити планування тренувань, контроль виконання вправ і відстеження прогресу спортсменів або клієнтів, однак кожна з них реалізує ці завдання по-різному та орієнтується на дещо різні категорії користувачів.

ABC Trainerize є однією з популярних платформ для персональних тренерів, фітнес-студій і онлайн-коучингу. Вона надає можливість створювати індивідуальні тренувальні програми, формувати бібліотеки вправ із відео-демонстраціями, відстежувати прогрес клієнтів і підтримувати комунікацію між тренером та спортсменом. Окремою перевагою платформи є підтримка роботи з харчуванням, звичками та загальним способом життя клієнта.

TrainHeroic більше орієнтована на силову підготовку, спортивні команди та тренерів, які працюють із групами спортсменів. Платформа дозволяє створювати календарі тренувань, призначати програми окремим спортсменам або командам, контролювати виконання вправ і збирати зворотний зв’язок щодо результатів тренування. Значну увагу приділено роботі з показниками сили, обсягом навантаження, повтореннями, вагою та динамікою прогресу.

TrueCoach також позиціонується як інструмент для персональних тренерів і онлайн-коучів. Вона підтримує створення тренувальних планів, призначення вправ, перегляд виконаних тренувань, обмін повідомленнями та відстеження окремих метрик клієнта. Перевагою TrueCoach є зручна взаємодія між тренером і клієнтом: спортсмен може отримувати завдання, вносити результати, залишати коментарі або відео виконання вправ, а тренер — оперативно переглядати ці дані. Водночас основний акцент платформи зроблено на комунікації та адмініструванні клієнтів, тоді як глибока аналітика

тренувального процесу та автоматизована оцінка навантаження можуть бути обмеженими або потребувати додаткової ручної обробки.

Everfit є сучасною платформою для тренерів, що поєднує конструктор тренувань, календар занять, бібліотеку вправ, контроль виконання програм і відстеження прогресу клієнтів. Система дозволяє тренеру створювати шаблони програм, призначати їх користувачам, переглядати історію тренувань і підтримувати комунікацію через мобільний застосунок.

Garmin Connect суттєво відрізняється від попередніх рішень, оскільки орієнтований насамперед на користувачів спортивних пристроїв Garmin. Носимі трекери активності є доступним інструментом для підвищення фізичної активності та моніторингу пов’язаних показників здоров’я [9]. Основна функціональність цієї системи пов’язана зі збором даних із годинників, велокомп’ютерів та інших фітнес-пристроїв. Платформа дозволяє переглядати статистику активності, маршрути, пульс, темп, дистанцію, калорії, сон, рівень стресу та інші фізіологічні показники. Garmin Connect є потужним інструментом для самостійного моніторингу активності спортсменом, однак він меншою мірою орієнтований на централізоване управління тренувальними програмами тренером. Також його використання значною мірою залежить від наявності сумісних пристроїв, що може обмежувати застосування системи в межах ширшої групи користувачів.

Частина існуючих систем більше орієнтована на самого спортсмена, який самостійно фіксує активність, переглядає статистику та оцінює власний прогрес. У таких рішеннях тренер не завжди має достатньо зручних інструментів для комплексного аналізу групи спортсменів, порівняння планових і фактичних показників, виявлення пропусків у тренувальному процесі або своєчасної корекції навантаження. В інших платформах, навпаки, добре реалізоване планування тренувань, але аналітична складова може зводитися переважно до перегляду історії виконання вправ без узагальнення даних у вигляді показників, необхідних для прийняття тренерських рішень.

Попри наявність великої кількості готових програмних продуктів, залишається актуальною розробка веб-платформи, яка поєднує планування тренувань, моніторинг їх виконання та аналітичний модуль тренера. Основна увага в такій системі має приділятися не лише збереженню тренувальних даних, а й їх практичному використанню для управління тренувальними програмами. Зокрема, важливими є оцінка регулярності занять, частоти тренувань, активного часу, тренувального обсягу, виконання плану та балансу навантаження.

1.5 Аналіз тренувальних даних як основи для прийняття рішень тренером

У сучасному тренувальному процесі дані відіграють важливу роль, оскільки дозволяють тренеру оцінювати не лише факт виконання тренування, а й якість, регулярність та динаміку роботи спортсмена. Якщо тренувальна програма існує окремо від результатів її виконання, тренер змушений приймати рішення переважно на основі суб’єктивних спостережень. Натомість структуроване збереження тренувальних даних дає змогу бачити реальну картину навантаження та своєчасно коригувати програму/ Оптимальну прогресію тренувань спортсменів можна підтримувати за допомогою надійних і валідних інструментів моніторингу [1].

У межах веб-платформи тренувальні дані можна поділити на декілька груп(див. табл. 1.1). До першої належать дані користувача: роль у системі, фізичні параметри, контакт із тренером або спортсменом. До другої групи належать дані планування: тренувальні програми, блоки вправ, кількість підходів, повторень, робоча вага, тривалість, тип активності та дата призначення тренування. Третю групу формують фактичні результати: виконані вправи, активний час, загальний час тренування, використана вага, кількість повторень, витрачені калорії та інші показники. Саме поєднання цих груп даних створює основу для аналітики.

Для тренера важливо бачити не ізольовані цифри, а зв’язок між планом і результатом. Наприклад, якщо спортсмен регулярно виконує тренування, але обсяг навантаження не зростає, це може свідчити про недостатню прогресію програми. Якщо активний час зменшується, а кількість пропущених тренувань збільшується, тренер може переглянути інтенсивність або частоту занять. Якщо навантаження розподілене нерівномірно між м’язовими групами, виникає потреба змінити структуру програми.

Таблиця 1.1 – Види тренувальних даних та їх використання тренером

Вид тренувальних даних	Приклад даних	Рішення, яке може прийняти тренер
Дані планування	Дата тренування, програма, блоки, вправи	Сформувати або змінити тренувальний план
Дані виконання	Активний час, повторення, вага, кількість підходів	Оцінити фактичне навантаження спортсмена
Дані регулярності	Кількість тренувань за період, частота занять	Виявити пропуски або нестабільність графіка
Дані прогресу	Зміна обсягу ваги, тривалості, калорій	Визначити позитивну або негативну динаміку
Дані м’язового балансу	Навантаження за групами м’язів	Скоригувати програму для рівномірного розвитку

Таким чином, тренувальні дані є не просто результатом роботи системи, а джерелом управлінських рішень для тренера. На їх основі можна визначати ефективність програми, змінювати навантаження, контролювати регулярність занять і планувати подальший розвиток спортсмена. Тижневий обсяг силового тренування має дозозалежний зв’язок зі збільшенням м’язової маси [7]. Саме

тому аналітичний модуль є важливою складовою веб-платформи, а не додатковою другорядною функцією.

1.6 Вимоги до аналітичного модуля тренера

Аналітичний модуль тренера є частиною веб-платформи, яка відповідає за обробку результатів тренувань і подання їх у вигляді зрозумілих показників. Його основне призначення полягає в тому, щоб допомогти тренеру оцінити стан тренувального процесу, виявити зміни в активності спортсмена та прийняти рішення щодо подальшого коригування програми.

Модуль повинен надавати тренеру можливість обрати спортсмена та період аналізу. Це важливо, оскільки тренер може працювати з декількома спортсменами, а оцінка результатів має сенс лише в межах конкретного проміжку часу. Наприклад, аналіз за два тижні дозволяє оцінити короткострокову регулярність, а аналіз за один або три місяці краще показує загальну динаміку тренувального процесу.

Ключовими показниками аналітичного модуля мають бути кількість виконаних тренувань, активний час, витрачені калорії, частота тренувань, регулярність занять та обсяг виконаної роботи. Різні призначення силового тренування можуть бути ефективними для розвитку сили та гіпертрофії порівняно з відсутністю тренувань [8]. Для силових вправ важливим показником є тренувальний обсяг, який може розраховуватися на основі ваги, кількості повторень і підходів. Також доцільно відображати навантаження за групами м’язів, оскільки це допомагає тренеру бачити, чи не перевантажується певна частина тіла та чи не ігноруються інші м’язові групи (див. табл. 1.2).

Окремою вимогою є зрозумілість інтерфейсу. Аналітичний модуль не повинен перевантажувати тренера великою кількістю сирих даних. Його завдання — показати найважливіші показники у зручному вигляді: картки метрик, графіки, діаграми та короткі порівняння з попереднім періодом. Це дозволяє швидко оцінити ситуацію без необхідності вручну аналізувати

таблиці результатів. Суб’єктивні показники самопочуття спортсмена можуть бути корисними для моніторингу його реакції на тренування [6].

Також модуль повинен коректно працювати з неповними даними. Якщо за обраний період спортсмен не виконував тренування, система має відобразити відсутність результатів, а не створювати помилкові висновки. Якщо частина показників відсутня, аналітика повинна враховувати це під час розрахунків.

Таблиця 1.2 – Вимоги до аналітичного модуля тренера

Вимога до модуля	Призначення
Вибір спортсмена	Дає тренеру змогу аналізувати результати конкретного підопічного
Вибір періоду аналізу	Дозволяє оцінювати короткострокову та довгострокову динаміку
Розрахунок основних показників	Формує загальне уявлення про активність спортсмена
Порівняння з попереднім періодом	Допомагає визначити позитивну або негативну тенденцію
Візуалізація даних	Спрощує сприйняття статистики тренером
Аналіз м’язового балансу	Дозволяє коригувати структуру тренувальної програми
Захист доступу до даних	Обмежує перегляд аналітики відповідно до ролі користувача

2 ПРОЄКТУВАННЯ АРХІТЕКТУРИ СИСТЕМИ

2.1 Загальна архітектура веб-платформи

Веб-платформа для планування та моніторингу тренувального процесу проєктується як клієнт-серверна система, у якій користувач взаємодіє із застосунком через веб-інтерфейс, а обробка даних, авторизація, збереження результатів і формування аналітики виконуються на серверній частині. Такий підхід дозволяє використовувати платформу з різних пристроїв без встановлення окремого програмного забезпечення.

Загальна архітектура системи складається з трьох основних рівнів: клієнтського інтерфейсу, серверної логіки та бази даних. Клієнтська частина відповідає за відображення сторінок, форм, календаря, тренувального процесу та аналітичних панелей. Серверна частина приймає запити від клієнта, перевіряє права доступу, виконує операції з даними та повертає результат у зручному для інтерфейсу форматі. База даних зберігає користувачів, тренувальні програми, вправи, результати виконання та дані, необхідні для аналітики.

Окреме місце в архітектурі займає аналітичний модуль. Він не є ізольованою частиною системи, а працює на основі даних, які накопичуються під час планування і виконання тренувань. Після завершення тренування система зберігає фактичні результати: активний час, кількість повторень, вагу, калорії та інші показники. Надалі ці дані використовуються для розрахунку статистики, порівняння періодів і візуалізації прогресу спортсмена для тренера.

У межах такої архітектури користувач працює з єдиним веб-інтерфейсом, але фактично кожна дія проходить через декілька рівнів системи(див. табл. 2.1). Наприклад, коли тренер створює тренувальну програму, дані спочатку вводяться у формі, потім передаються на сервер, перевіряються та зберігаються в базі даних. Коли спортсмен виконує

тренування, система фіксує результати вправ, після чого ці дані стають доступними для аналітичного модуля.

Таблиця 2.1 – Основні рівні архітектури веб-платформи

Рівень системи	Призначення
Клієнтський рівень	Відображення інтерфейсу, форм, календаря, тренувань і аналітики
Серверний рівень	Обробка запитів, авторизація, перевірка ролей, бізнес-логіка
Рівень бази даних	Збереження користувачів, програм, тренувань, вправ і результатів
Аналітичний рівень	Розрахунок показників, агрегація даних, підготовка статистики

2.2 Системний аналіз веб-платформи

Об’єктом системного аналізу є веб-платформа, яка забезпечує взаємодію між спортсменом, тренером та адміністратором. Предметом аналізу є інформаційні потоки, функціональні зв’язки між модулями, структура даних, ролі користувачів та механізми перетворення первинних тренувальних даних в аналітичні показники. Основною метою системи є підвищення керованості тренувального процесу за рахунок централізованого зберігання даних, автоматизації планування, фіксації фактичних результатів і формування статистики для тренера(див. рис. 2.1).

З позиції системного підходу веб-платформу можна подати як систему, що має вхідні дані, процеси обробки, вихідні результати та зворотний зв’язок. До вхідних даних належать облікові записи користувачів, тренувальні програми, вправи, параметри навантаження, розклад занять і фактичні результати виконання тренувань. Процеси обробки включають авторизацію

користувачів, перевірку ролей доступу, створення програм, призначення тренувань, збереження результатів вправ, агрегацію статистичних даних і побудову аналітичних показників. Вихідними результатами є календар тренувань, історія виконання, статистичні картки, графіки активності, показники регулярності, план-факт виконання та дані для прийняття тренером рішень щодо корекції програми.

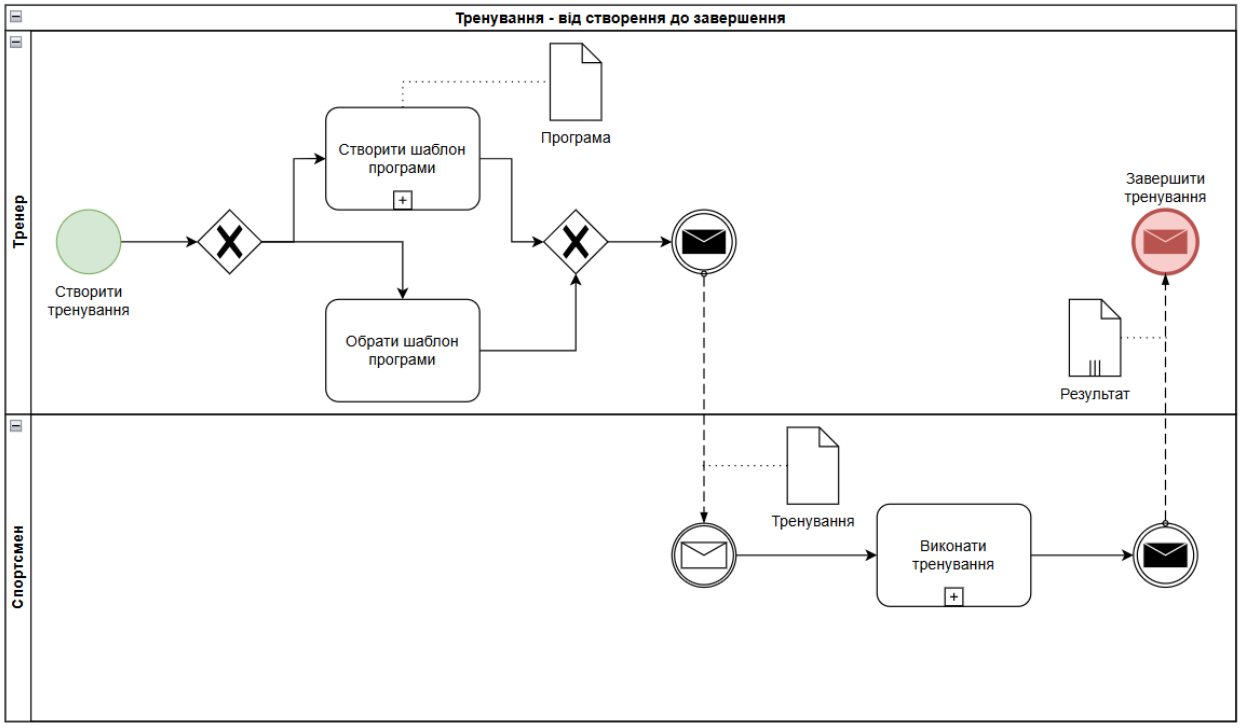


Рисунок. 2.1 – BPMN-діаграма процесу тренування.

Основними підсистемами веб-платформи є підсистема керування користувачами, підсистема тренувальних програм, підсистема планування, підсистема виконання тренування, підсистема збереження результатів, аналітична підсистема та підсистема адміністрування довідкових даних(див рис. 2.2). Кожна з них виконує окрему функцію, однак ефективність системи досягається саме завдяки їх взаємодії. Наприклад, модуль планування використовує дані з модуля тренувальних програм, модуль виконання тренування створює фактичні результати, а аналітичний модуль використовує ці результати для формування статистики.

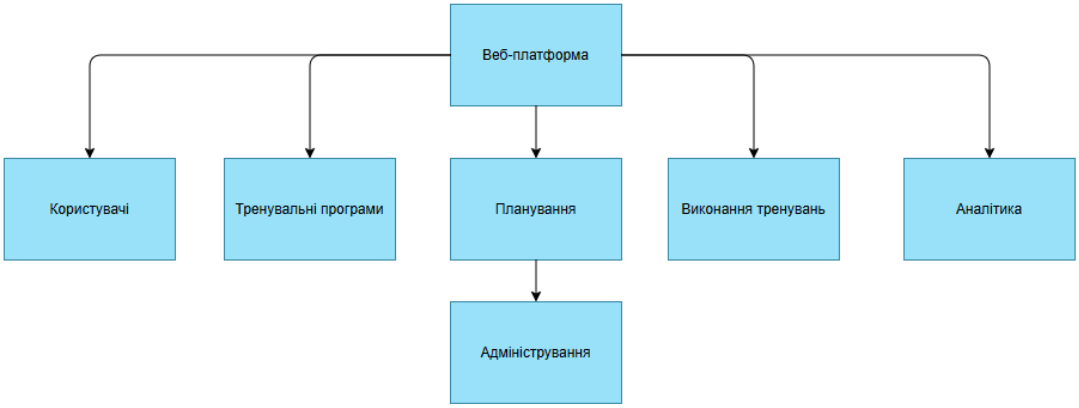


Рисунок 2.2 – Основні підсистеми.

Особливу роль у системі відіграє зворотний зв’язок. Після виконання тренування спортсменом система зберігає фактичні показники: активний час, кількість підходів і повторень, використану вагу, дистанцію, калорії та інші параметри. Ці дані надходять до аналітичного модуля, де перетворюються на узагальнені показники. На основі отриманої інформації тренер може оцінити, чи відповідає фактичне виконання запланованому навантаженню, чи дотримується спортсмен регулярності занять, чи спостерігається прогрес або, навпаки, зниження активності. Таким чином, система реалізує замкнений цикл управління тренувальним процесом: планування — виконання — аналіз — корекція.

З погляду користувачів система має три основні зовнішні ролі: спортсмена, тренера та адміністратора(див. табл. 2.2). Спортсмен взаємодіє із системою переважно через календар, екран тренування та форми введення результатів. Тренер використовує платформу для створення програм, призначення тренувань, перегляду виконання та аналізу статистики. Адміністратор забезпечує підтримку довідкової інформації, необхідної для коректної роботи системи: вправ, типів активностей, обладнання, м’язових груп та інших базових сутностей. Такий розподіл ролей дозволяє обмежити

доступ до функцій відповідно до завдань користувача та підвищити безпеку роботи з даними.

Таблиця 2.2 – Ролі користувачів веб-платформи

Роль	Основні можливості
Спортсмен	Перегляд призначених тренувань, виконання вправ, збереження результатів
Тренер	Створення програм, призначення тренувань, моніторинг спортсменів, перегляд аналітики
Адміністратор	Керування довідниками вправ, обладнання, активностей і м’язових груп

Інформаційні потоки системи можна поділити на декілька основних напрямів. Перший потік пов’язаний з автентифікацією та авторизацією користувачів. Другий потік охоплює створення тренувальних програм і збереження їх структури в базі даних. Третій потік відповідає за планування тренувань і прив’язку програми до конкретного спортсмена та дати. Четвертий потік формується під час виконання тренування, коли спортсмен вносить фактичні результати. П’ятий потік пов’язаний з аналітичною обробкою даних, коли система агрегує результати та формує показники для тренера.

З системної точки зору важливо, щоб усі модулі працювали з єдиною базою даних і не дублювали інформацію. Це дозволяє уникнути розбіжностей між плановими та фактичними даними, спрощує підтримку системи та забезпечує цілісність інформації. Наприклад, тренувальна програма створюється один раз, після чого може бути призначена різним спортсменам або на різні дати. Фактичні результати не змінюють сам шаблон програми, а зберігаються окремо як результат конкретного виконання. Такий підхід

дозволяє одночасно зберігати планову структуру тренування та історію реального виконання.

У результаті можна визначити, що розроблювана веб-платформа є багаторівневою інформаційною системою з чітким розподілом функцій між клієнтською частиною, серверною логікою, базою даних та аналітичним модулем. Її основна цінність полягає у поєднанні планування, виконання та аналізу тренувань у межах одного середовища.

2.3 Проєктування ролей користувачів: спортсмен, тренер, адміністратор

Для коректної роботи веб-платформи необхідно розмежувати можливості користувачів відповідно до їх ролей(див. рис. 2.3). У системі передбачено три основні ролі: спортсмен, тренер та адміністратор. Такий поділ дозволяє обмежити доступ до функцій, які не потрібні певному користувачу, і водночас забезпечити тренеру необхідні інструменти для управління тренувальним процесом.

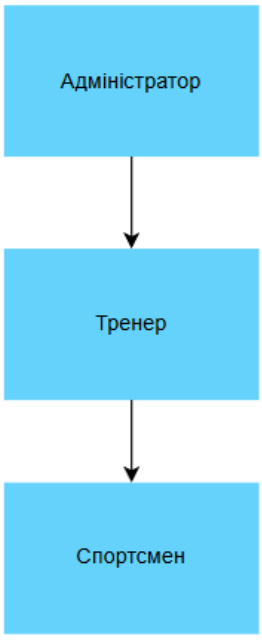


Рисунок 2.3 – Ієрархія ролей користувачів веб-платформи.

Спортсмен є користувачем, який виконує призначені тренування. Його основні дії пов’язані з переглядом календаря, запуском тренування, проходженням вправ і збереженням фактичних результатів. Для спортсмена важливо, щоб інтерфейс був простим і не вимагав додаткових налаштувань під час тренування.

Тренер має ширші можливості. Він створює тренувальні програми, формує шаблони, призначає тренування спортсменам і переглядає результати їх виконання. Саме тренер є основним користувачем аналітичного модуля, оскільки йому потрібно бачити динаміку активності, регулярність занять, обсяг навантаження та баланс роботи за м’язовими групами.

Адміністратор відповідає за підтримку довідкової інформації системи. До такої інформації належать вправи, типи вправ, типи активностей, обладнання, групи м’язів, типи підходів та інші сутності, на основі яких тренер створює програми. Адміністратор не втручається безпосередньо у тренувальний процес спортсмена, але забезпечує коректність базових даних платформи.

Проєктування ролей також впливає на структуру доступу до даних. Спортсмен повинен бачити власні тренування та результати. Тренер повинен мати доступ до даних спортсменів, з якими він взаємодіє. Адміністратор отримує доступ до службових розділів, необхідних для підтримки системи.

2.4 Проєктування модуля управління тренувальними програмами

Модуль управління тренувальними програмами є одним із центральних елементів веб-платформи, оскільки саме через нього тренер формує основу майбутніх тренувань. Тренувальна програма розглядається не як простий список вправ, а як структурований набір блоків, у яких визначаються тип активності, порядок виконання, кількість підходів, час, повторення, вага та інші параметри.

Під час проєктування модуля важливо забезпечити тренеру можливість створювати як нові програми, так і програми на основі вже існуючих шаблонів. Це зменшує кількість повторної роботи, оскільки тренер часто використовує схожі структури тренувань для різних спортсменів, змінюючи лише окремі вправи або параметри навантаження.

Структура тренувальної програми складається з декількох рівнів(див. рис. 2.4). На верхньому рівні знаходиться сама програма, яка має назву, опис, автора та статус. Нижче розташовуються блоки програми. Блок може відповідати окремій частині тренування, наприклад розминці, силовому блоку або кардіо. Усередині блоку розміщуються вправи з конкретними параметрами виконання(див. табл. 2.3).

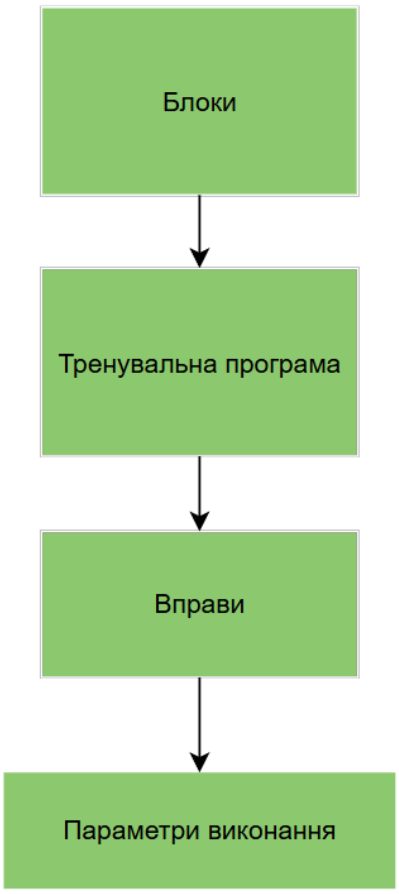


Рисунок 2.4 - Структура тренувальної програми.

Таблиця 2.3 – Структура тренувальної програми

Елемент програми	Зміст
Програма	Назва, опис, автор, статус, дата створення
Блок програми	Тип активності, позиція в програмі, кількість підходів, час
Вправа	Назва вправи, повторення, вага, час, дистанція або інші параметри
Довідкові дані	Типи вправ, обладнання, групи м’язів, типи активностей

Для зручності тренера модуль повинен підтримувати додавання, редагування та видалення блоків і вправ, зміну їх порядку, вибір вправ із каталогу та збереження готової програми.

2.5 Проєктування модуля планування тренувань

Модуль планування тренувань призначений для того, щоб перетворити створену тренувальну програму на конкретне заплановане заняття для спортсмена. Якщо модуль управління програмами відповідає на питання “що потрібно виконати”, то модуль планування відповідає на питання “хто, коли і за якою програмою буде тренуватися”(див. табл. 2.4).

Таблиця 2.4 – Дані, необхідні для планування тренування

Дані	Призначення
Спортсмен	Визначає користувача, якому призначено тренування
Тренер	Визначає автора призначення
Програма	Містить структуру вправ і параметри виконання
Дата і час	Визначають місце тренування в календарі
Статус виконання	Показує, чи було тренування завершене

Основною сутністю цього модуля є заплановане тренування. Воно містить посилання на тренувальну програму, спортсмена, якому призначене тренування, тренера, який його створив, дату та час виконання. Завдяки цьому одна й та сама програма може бути використана багато разів для різних спортсменів або для різних дат.

Планування тренування починається з вибору спортсмена. Для цього тренер повинен мати список користувачів, з якими він має зв’язок у системі. Далі тренер обирає підготовлену програму або створює нову, після чого задає дату та час тренування. Після збереження тренування відображається у календарі спортсмена.

Календар є важливою частиною модуля планування, оскільки дозволяє бачити тренувальний процес у часовій послідовності. Через календар спортсмен може швидко знайти найближче тренування, а тренер — оцінити розклад занять. Якщо тренування вже виконане, система може відобразити це окремим станом, щоб відрізнити завершені заняття від запланованих.

Проєктування цього модуля також пов’язане з подальшою аналітикою. Коли тренування заплановане, система має очікувану структуру навантаження. Коли спортсмен його виконує, з’являються фактичні результати. Порівняння плану та виконання створює основу для моніторингу: тренер бачить, які тренування були завершені, як часто спортсмен тренується і які результати накопичуються з часом.

2.6 Проєктування модуля моніторингу виконання тренувань

Модуль моніторингу виконання тренувань проєктується як частина системи, що поєднує заплановану тренувальну програму з фактичними результатами її проходження спортсменом. Його основне завдання полягає не тільки у відображенні тренування, а й у фіксації процесу виконання: які вправи були виконані, скільки часу зайняло тренування, які параметри навантаження були використані та чи було тренування завершене.

Для спортсмена цей модуль повинен виглядати як зручний інтерфейс проходження тренування. Після вибору запланованого заняття система відкриває структуру програми: блоки, вправи, кількість підходів, повторень, вагу, час та інші параметри. Спортсмен бачить поточну вправу, чергу наступних вправ і загальний прогрес. Це дозволяє виконувати тренування послідовно, без необхідності додатково переглядати повний план або уточнювати порядок дій у тренера.

Важливою частиною модуля є фіксація виконання окремих вправ. Після завершення підходу або вправи система зберігає результат у базі даних. До таких результатів можуть належати час виконання, кількість повторень, робоча вага, дистанція, швидкість, навантаження тренажера, витрачені калорії та інші параметри залежно від типу активності. Таким чином, кожне тренування поступово перетворюється з плану на набір фактичних даних.

Для тренера модуль моніторингу має інше значення. Він дозволяє бачити, чи виконує спортсмен призначені тренування, наскільки регулярно він тренується і які результати накопичуються з часом. Тренер може оцінювати не лише завершення тренування, а й зміст виконаної роботи: активний час, обсяг навантаження, кількість вправ і загальний стан тренувальної активності.

Проєктування цього модуля безпосередньо пов’язане з подальшим аналізом даних. Якщо система лише показує тренування, але не зберігає результати, тренер не отримує основи для прийняття рішень. Тому моніторинг виконання повинен завершуватися створенням запису про тренування та записів про виконані вправи. Саме ці дані надалі використовуються в аналітичному модулі.

2.7 Проєктування бази даних тренувального процесу

База даних є основою веб-платформи, оскільки саме в ній зберігаються користувачі, тренувальні програми, заплановані заняття, довідники вправ і фактичні результати виконання(див. табл 2.5). Під час проєктування бази

даних важливо було забезпечити не тільки збереження інформації, а й логічний зв’язок між плануванням, виконанням і подальшою аналітикою.

Таблиця 2.5 – Основні групи таблиць бази даних тренувального процесу

Група таблиць	Приклади сутностей	Призначення
Користувачі	Users, UserContacts	Збереження профілів, ролей і зв’язків між користувачами
Програми	ProgramsTemplates, ProgramTemplateBlocks, BlockTemplateExercises, ExercisesTemplates	Опис структури тренувальних програм
Планування	WorkoutsSchedule	Призначення програми спортсмену на конкретну дату
Результати	WorkoutResults, ExerciseResults	Збереження фактичного виконання тренувань і вправ
Довідники	ExercisesCatalog, Exercises, ActivitiesTypes, MusclesTypes, Equipments	Базові дані для створення програм і аналітики

У центрі структури даних знаходиться користувач. Користувач може бути спортсменом, тренером або адміністратором. Для взаємодії між тренером і спортсменом передбачені контакти, які дозволяють визначити, з якими спортсменами працює тренер. Це важливо для призначення тренувань і перегляду аналітики, оскільки тренер повинен бачити дані лише тих спортсменів, з якими він пов’язаний у системі.

Тренувальна програма зберігається як окрема сутність. Вона складається з блоків, а кожен блок містить вправи з конкретними параметрами виконання.

Така структура дозволяє будувати тренування не як плоский список, а як логічно впорядковану програму. Наприклад, програма може містити блок розминки, силовий блок і блок кардіо, а кожен із них матиме власні вправи та параметри.

Окремо зберігаються заплановані тренування. Заплановане тренування пов’язує спортсмена, тренера, дату, час і конкретну тренувальну програму. Коли спортсмен виконує тренування, система створює записи результатів: загальний результат тренування та результати окремих вправ.

База даних проєктується як реляційна структура, у якій планові та фактичні дані пов’язані між собою. Це дає можливість відстежувати повний шлях тренування: від створення програми тренером до виконання спортсменом і подальшого аналізу результатів(див. рис. 2.3, 2.4).

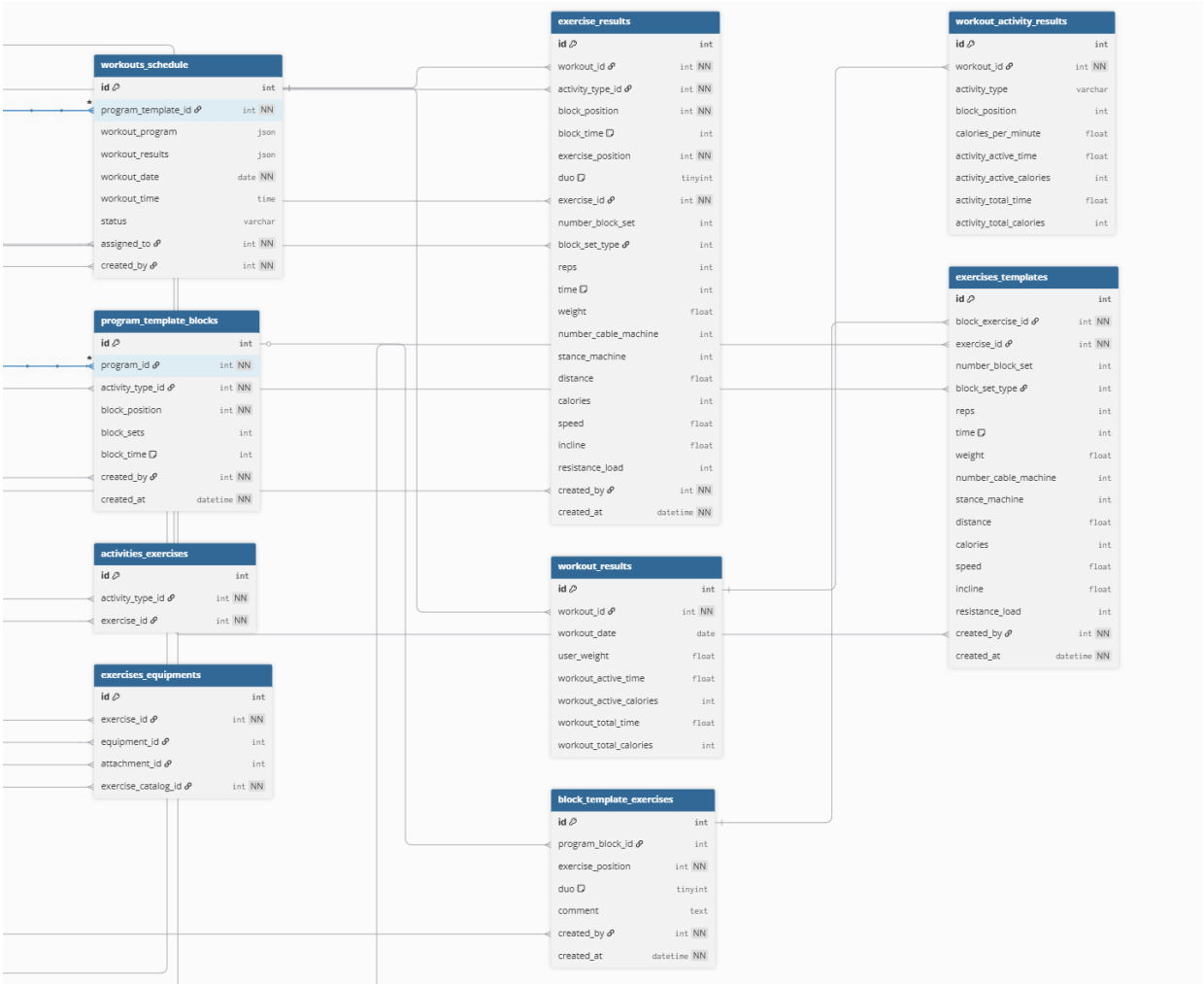


Рисунок 2.3 – Схема бд. Перша частина.

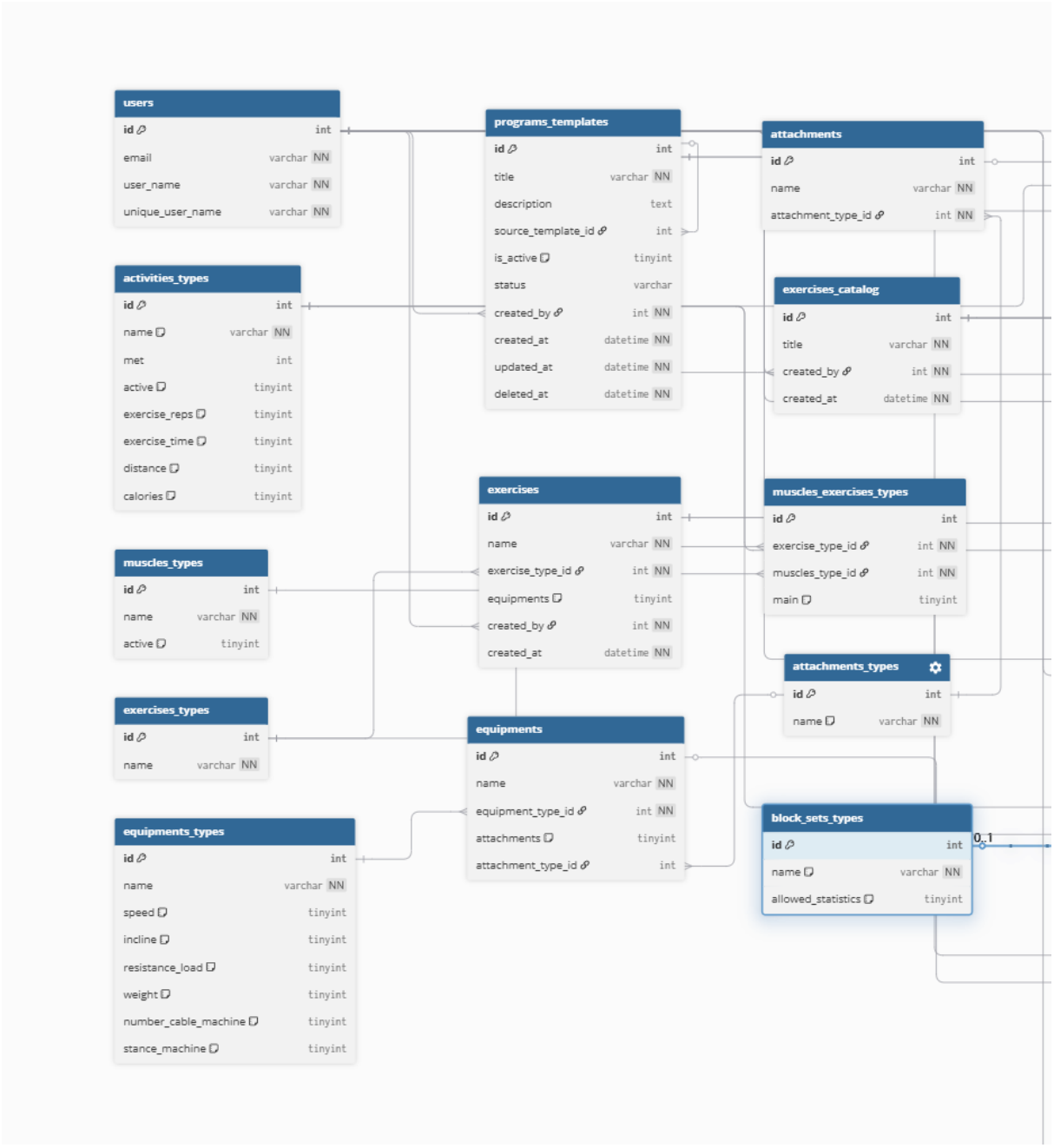


Рисунок 2.4 – Схема бд. Друга частина.

2.8 Проєктування потоків даних між клієнтською частиною, сервером і БД

Потоки даних у веб-платформі проєктуються таким чином, щоб кожна дія користувача проходила через клієнтський інтерфейс, серверну обробку та базу даних. Клієнтська частина відповідає за взаємодію з користувачем: форми, кнопки, календар, тренувальний екран та аналітичні панелі. Серверна

частина перевіряє запити, виконує бізнес-логіку, контролює доступ відповідно до ролі користувача та звертається до бази даних(див. табл. 2.6).

Таблиця 2.6 – Основні потоки даних у веб-платформі

Потік даних	Джерело	Результат
Авторизація	Форма входу	Токен доступу та профіль користувача
Створення програми	Конструктор тренувань	Збережена програма з блоками і вправами
Планування тренування	Форма призначення	Запис тренування в календарі
Виконання тренування	Екран проходження вправ	Результати вправ і завершеного тренування
Формування аналітики	Запит тренера	Агреговані показники за обраний період

Першим важливим потоком є авторизація. Користувач вводить облікові дані у веб-інтерфейсі, після чого вони передаються на сервер. Сервер перевіряє користувача, формує токени доступу та повертає їх клієнтській частині. Надалі ці токени використовуються для виконання захищених запитів. Завдяки цьому система може визначити, хто саме виконує дію: спортсмен, тренер або адміністратор.

Потік створення тренувальної програми починається з дій тренера у конструкторі програми. Тренер додає блоки, вибирає вправи, задає параметри навантаження і зберігає програму. Клієнтська частина передає сформовану структуру на сервер, а сервер записує її в базу даних як програму, блоки та вправи. У цьому випадку важливо зберегти правильну ієрархію: програма повинна бути пов’язана з блоками, а блоки — з вправами.

Потік планування тренування пов’язує готову програму з конкретним спортсменом і датою. Тренер обирає спортсмена, програму, дату та час, після чого система створює запис у календарі. Коли спортсмен відкриває календар, клієнтська частина запитує у сервера тренування за обраний період, а сервер отримує ці дані з бази даних і повертає їх у вигляді списку або календарної мапи.

Найбільш важливим для аналітики є потік виконання тренування. Під час проходження тренування спортсмен поступово завершує вправи, а система передає результати на сервер. Сервер зберігає результати окремих вправ, розраховує додаткові показники, а після завершення тренування формує загальний запис результату. Надалі ці дані стають основою для статистики та візуалізації.

Інтерфейс лише відображає дані та передає дії користувача, а основна перевірка, розрахунки та збереження виконуються на сервері.

2.9 Проєктування аналітичного модуля тренера

Аналітичний модуль тренера проєктується як інструмент підтримки прийняття рішень, який використовує накопичені результати тренувань і перетворює їх на зрозумілі показники(див. табл. 2.7). На відміну від звичайної історії тренувань, аналітичний модуль повинен не просто показувати список виконаних занять, а узагальнювати інформацію за певний період і допомагати тренеру оцінити ефективність тренувальної програми.

Основною логікою модуля є аналіз даних конкретного спортсмена. Тренер повинен мати можливість обрати підопічного та період аналізу. Після цього система отримує з бази даних результати тренувань і вправ, агрегує їх та формує набір метрик. Такий підхід дозволяє тренеру швидко перейти від загального списку спортсменів до оцінки стану одного конкретного користувача.

Таблиця 2.7 - Аналітичні показники модуля тренера

Аналітичний показник	Що показує тренеру
Кількість тренувань	Загальну активність спортсмена за період
Активний час	Скільки часу спортсмен фактично виконував вправи
Калорії	Приблизні енергетичні витрати під час тренувань
Частота тренувань	Середню кількість занять на тиждень
Регулярність	Стабільність тренувального процесу
Обсяг навантаження	Сумарну виконану роботу у вправах з вагою
М’язовий баланс	Розподіл навантаження між групами м’язів

До основних показників аналітичного модуля належать кількість тренувань, активний час, витрачені калорії, частота тренувань, регулярність і загальний обсяг навантаження. Для силових вправ обсяг може визначатися на основі ваги, кількості повторень і підходів. Це дає можливість оцінити не тільки факт тренування, а й реальний рівень виконаної роботи.

Окремим напрямом аналітики є аналіз м’язового балансу. Оскільки вправи пов’язані з типами вправ і групами м’язів, система може визначати, які м’язові групи отримували навантаження протягом обраного періоду. Для тренера це важливо, оскільки нерівномірний розподіл навантаження може свідчити про недоліки в програмі або потребу в її корекції.

Проєктування модуля передбачає також порівняння поточного періоду з попереднім. Це дозволяє тренеру бачити динаміку: чи збільшилася кількість тренувань, чи зростає активний час, чи не зменшується обсяг навантаження. Саме порівняння робить аналітику практично корисною, оскільки окреме значення без контексту не завжди дозволяє зробити висновок.

Візуальна частина модуля повинна бути простою для сприйняття. Основні показники доцільно подавати у вигляді карток метрик, а складніші

дані — у вигляді графіків або діаграм. Наприклад, м’язовий баланс можна показати радарною діаграмою, а зміну обсягу навантаження — графіком або стовпчиковою діаграмою. Це дозволяє тренеру швидко оцінити ситуацію без ручного перегляду всіх тренувань.

Аналітичний модуль не замінює тренера, а надає йому інформаційну основу для рішень. Остаточну корекцію програми тренер виконує самостійно, враховуючи цілі спортсмена, його самопочуття, рівень підготовки та результати, які показує система. Саме тому модуль проєктується як допоміжний інструмент управління тренувальними програмами, а не як автоматична система прийняття рішень.

3 РЕАЛІЗАЦІЯ ТА РЕЗУЛЬТАТИ РОБОТИ

3.1 Реалізація веб-платформи та її основних модулів

Веб-платформа реалізована як сучасний клієнт-серверний застосунок на базі Next.js, React та TypeScript. Такий підхід дозволив поєднати інтерфейс користувача, серверні API-маршрути та роботу з базою даних в одній структурі проєкту. Основна логіка платформи поділена на приватну частину для авторизованих користувачів, API-рівень для обробки запитів та серверний шар для роботи з PostgreSQL(див. табл. 3.1).

Після входу користувач потрапляє до приватного кабінету з боковим меню, через яке доступні основні розділи: тренування, календар, статистика та профіль. Така структура зроблена не як набір окремих сторінок, а як єдиний робочий простір спортсмена або тренера. Користувач не переходить між випадковими екранами, а працює в логіці тренувального процесу: переглядає програми, планує заняття, виконує тренування, а потім аналізує результати.

Таблиця 3.1 – модулі программи та їх призначення

Модуль	Основне призначення
Модуль тренувань	Перегляд програм, створення шаблонів, запуск тренування
Модуль календаря	Планування занять за датою та часом
Модуль статистики	Аналіз виконаних тренувань, активності, калорій, обсягу навантаження
Профіль користувача	Збереження фізичних параметрів, які використовуються в розрахунках
Контакти	Зв’язок між тренером і спортсменом
Адміністративний модуль	Керування довідниками вправ, обладнання, груп м’язів і типів активності

Особливу роль у системі відіграє аналітичний модуль. Він не просто показує список виконаних тренувань, а перетворює зібрані дані на показники, зручні для тренера: кількість тренувань за період, активний час, витрачені калорії, частоту занять, регулярність і тренувальний обсяг. Для візуалізації використовуються графіки та окремі картки метрик, що допомагає швидко оцінити динаміку спортсмена.

На серверній стороні всі сутності зберігаються в PostgreSQL. Для роботи з таблицями використовується узагальнений CRUD-шар та метадані, де описані поля, типи даних і обов'язковість значень. Завдяки цьому створення користувачів, тренувальних програм, блоків, вправ і запланованих тренувань виконується за єдиним принципом. Це спрощує підтримку системи та зменшує кількість дублювання в коді.

3.2 Реалізація авторизації та рольового доступу користувачів

Авторизація у веб-платформі реалізована через власний механізм JWT-токенів. JWT є компактним URL-безпечним способом передавання тверджень між двома сторонами [10]. Після реєстрації або входу користувач отримує два токени: accessToken і refreshToken. accessToken використовується для більшості запитів до приватної частини системи, а refreshToken потрібен для оновлення сесії, якщо основний токен застарів. Генерація і перевірка токенів винесені в серверні модулі авторизації, де формується JWT-підпис на основі секретного ключа з environment-змінних.

На клієнтській стороні токени зберігаються в localStorage. Перед відкриттям приватних сторінок виконується функція перевірки авторизації: вона бере accessToken, надсилає запит на /api/auth і отримує профіль користувача. Якщо токен недійсний або прострочений, система пробує виконати оновлення через /api/refresh. Якщо оновлення не вдалося, токени очищуються, а користувач перенаправляється на сторінку входу.

У приватній частині застосунку використовується UserContext, куди записується профіль поточного користувача. Це дозволяє різним компонентам отримувати роль, ідентифікатор користувача та інші дані без повторних запитів. Наприклад, модулі тренувань, календаря й аналітики перевіряють роль користувача і на основі цього показують або приховують окремі дії.

У системі передбачено три ролі. Вони описані у вигляді enum, де спортсмен має значення 0, тренер — 1, адміністратор — 2. Завдяки числовій ієрархії перевірка тренерських прав виконується просто: якщо значення ролі >= значення ролі тренера, користувач може створювати програми, призначати тренування та переглядати розширену аналітику. Адміністративні можливості доступні тільки тоді, коли роль дорівнює admin.

Серверна перевірка доступу реалізована через функцію-обгортку над api шляхами. Вона автоматично створює підключення до бази даних, перевіряє Authorization-заголовок, дістає ідентифікатор користувача з токена та, якщо потрібно, додатково перевіряє права тренера або адміністратора. Тому обмеження доступу працює не тільки на рівні інтерфейсу, а й на рівні API(див. табл. 3.2).

Таблиця 3.2 – API – шляхи системи авторізації

API-шлях	Метод	Призначення
/api/sign-up	POST	Реєструє нового користувача, створює запис у таблиці Users і повертає пару токенів.
/api/login	POST	Перевіряє email і пароль користувача, після успішного входу

API-шлях	Метод	Призначення
		повертає accessToken і re freshToken.
/api/auth	GET	Перевіряє поточний accessToken і повертає дані авторизованого користувача.
/api/refresh	POST	Приймає refresh_token, перевіряє його та видає нову пару токенів.
/api/users/[id]	PATCH	Оновлює дані профілю користувача: ім’я, email, зріст, вагу, дату народження та інші персональні поля.
/api/channel	POST	Змінює роль користувача, наприклад переводить звичайного спортсмена у роль тренера.
/api/exercises, /api/equipments, /api/muscles-types	GET/POST/PATCH/ DELETE	Приклад адміністративних маршрутів, доступ до яких обмежується роллю admin.

3.3 Реалізація створення та редагування тренувальних програм

Створення тренувальних програм реалізовано через компонент конструктора програми (див. табл. 3.3). До нього тренер може перейти з розділу тренувань або під час створення нового запланованого заняття, якщо потрібного шаблону ще немає. У коді цей сценарій винесений в окремий компонент, який працює з локальним станом форми, списком блоків і списком вправ.

Програма складається з основної інформації та вкладеної структури блоків. На верхньому рівні зберігається назва, опис, статус, активність програми, автор створення та дати створення/оновлення. Нижче йдуть блоки програми. Кожен блок має тип активності, позицію в програмі, кількість сетів або часову тривалість. Усередині блоку знаходяться вправи, а для кожної вправи зберігаються параметри навантаження: повторення, вага, час, дистанція, швидкість, калорії, нахил, опір та інші поля.

Редагування програми відбувається на рівні робочого стану перед збереженням у базу даних. Тренер може додати новий блок, змінити тип активності, переключити метрику між сетами і часом, додати вправи з каталогу або видалити зайві елементи.

Після натискання кнопки збереження клієнт надсилає на сервер весь шаблон програми разом із блоками та вправами. Для цього використовується спеціальний маршрут /api/programs-templates/submit. На сервері збереження виконується транзакційно: спочатку створюється запис у ProgramsTemplates, потім додаються блоки в ProgramTemplateBlocks, після цього створюються записи у BlockTemplateExercises та ExercisesTemplates. Якщо під час збереження виникає помилка, виконується ROLLBACK, і частково створена програма не потрапляє в базу.

Таблиця 3.3 – API тренувальних програм

API-шлях	Метод	Призначення
/api/programs-templates	GET	Повертає список шаблонів програм користувача. Якщо передано blocks=true, повертає програми разом із блоками та вправами.
/api/programs-templates	POST	Створює базовий запис шаблону програми без повної вкладеної структури.
/api/programs-templates/[id]	GET	Повертає одну програму за ідентифікатором разом із блоками, вправами, типами активності й обладнанням.
/api/programs-templates/submit	POST	Основний маршрут збереження повної програми: шаблон, блоки, вправи та параметри вправ.
/api/program-template-blocks	POST	Створює окремий блок програми, якщо потрібно працювати з блоками напряму.
/api/exercises-catalog	GET	Завантажує каталог вправ, з якого тренер обирає вправи для програми.
/api/activities-types	GET	Повертає типи активностей, які використовуються для блоків програми і розрахунку калорій.

Дані програми зберігаються у кількох пов’язаних таблицях. ProgramsTemplates відповідає за сам шаблон, ProgramTemplateBlocks – за

блоки та їх порядок, BlockTemplateExercises - за зв’язок блоку з вправами, а ExercisesTemplates - за параметри конкретної вправи.

3.4 Реалізація планування тренувань у календарі

Планування тренувань реалізовано через створення записів у таблиці WorkoutsSchedule(див. табл. 3.4). Тренер спочатку обирає спортсмена зі списку контактів, потім вибирає шаблон тренувальної програми, задає дату та час. Після збереження система створює заплановане тренування, у якому фіксується, хто його створив, кому воно призначене і за якою програмою має виконуватися.

Таблиця 3.4 – API тренувань

API-шлях	Метод	Призначення
/api/workouts-schedule	POST	Створює заплановане тренування на основі шаблону програми, дати, часу та виконавця.
/api/calendar?start_date=...&end_date=...	GET	Повертає тренування користувача за вибраний діапазон дат для відображення в календарі.
/api/trainings/[id]	GET	Завантажує конкретне заплановане тренування для подальшого проходження.
/api/trainings/results/[id]	GET	Перевіряє, чи є збережені результати для конкретного тренування.

Форма створення тренування використовує дані з кількох джерел. Список доступних програм завантажується з /api/programs-templates?userId=...&blocks=true, а список спортсменів формується через контакти тренера. Це зроблено для того, щоб тренер міг призначати заняття тільки тим користувачам, з якими він пов’язаний у системі.

Після створення тренування воно відображається в календарі. Календар будує місячну сітку, додаючи дні попереднього та наступного місяця для рівного відображення. Коли користувач відкриває календар або перемикає місяць, клієнтська частина визначає перший і останній видимий день та надсилає запит на /api/calendar. Сервер повертає тренування, згруповані за датами.

Запит до бази отримує записи з WorkoutsSchedule, приєднує назву програми з ProgramsTemplates і перевіряє, чи є для тренування результат у WorkoutResults. Якщо результат існує, календар може позначити тренування як завершене.

У правій частині календаря користувач бачить тренування на вибрану дату. Якщо заняття ще не виконане, доступна дія для переходу до проходження тренування. Якщо результат уже є, користувач може перейти до перегляду результатів. Для тренера додатково показується кнопка створення нового тренування, тому календар працює не лише як перегляд розкладу, а й як центр планування тренувального процесу.

3.5 Реалізація процесу проходження тренування та фіксації результатів

Проходження тренування реалізовано як окремий інтерактивний сценарій. Коли користувач переходить до виконання заняття, сторінка отримує id тренування з параметрів URL і надсилає запит до /api/trainings/[id]. Сервер знаходить запис у WorkoutsSchedule, підтягує відповідний шаблон програми, блоки та вправи, після чого повертає клієнту готову структуру тренування(див. табл. 3.5).

На клієнтській стороні блоки програми перетворюються на послідовну чергу вправ. Для кожної вправи зберігається її позиція у блоці, позиція в загальній черзі, кількість сетів, параметри навантаження та тип активності. Користувач бачить поточну вправу, кількість повторень, вагу, блок тренування, прогрес і список наступних вправ.

Таблиця 3.5 – API процесу тренування

API-шлях	Метод	Призначення
/api/trainings/[id]	GET	Завантажує тренування разом із програмою, блоками та вправами.
/api/trainings/[id]	PATCH	Зберігає результат виконаної вправи в ExerciseResults.
/api/trainings/[id]	POST	Завершує тренування і створює загальний запис у WorkoutResults.
/api/trainings/results/[id]	GET	Повертає загальний результат тренування та список виконаних вправ.
/api/trainings/results	GET	Повертає історію результатів користувача або спортсмена, якщо запит виконує тренер.

У процесі виконання працює таймер. Він рахує час поточної вправи та загальний активний час тренування. Після натискання кнопки «Сет виконано» система або збільшує кількість виконаних сетів, або завершує поточну вправу і переходить до наступної. Коли вправа завершена, клієнт надсилає PATCH-запит до /api/trainings/[id], де передає фактичні параметри виконання.

На сервері результат окремої вправи зберігається в таблиці ExerciseResults. Перед збереженням система розраховує калорії за активним часом, типом активності та вагою користувача. Для цього

використовується MET-показник із таблиці типів активності. Завдяки цьому результат містить не тільки введені або взяті з програми параметри, а й розрахунковий показник енерговитрат.

Коли тренування завершується, клієнт надсилає POST-запит до /api/trainings/[id]. Сервер збирає всі результати вправ, рахує сумарний активний час, калорії, загальний час тренування та створює запис у WorkoutResults. Якщо користувач відкриває тренування повторно, система перевіряє вже збережені результати через /api/trainings/results/[id] і визначає, з якої вправи продовжити.

3.6 Реалізація аналітичного модуля обробки тренувальних даних

Аналітичний модуль побудований на основі даних, які накопичуються під час виконання тренувань(див. табл. 3.6). Він використовує записи з WorkoutsSchedule, WorkoutResults, ExerciseResults, а також довідники вправ, груп м’язів і типів активності. Завдяки цьому система аналізує не тільки факт виконання занять, а й реальне навантаження спортсмена.

Таблиця 3.6 – API аналітичного модуля

API-шлях	Метод	Призначення
/api/analytics/overview?userId=...&period=...	GET	Повертає метрики за вибраний період і попередній.
/api/analytics/muscle-balance?userId=...&period=...	GET	Повертає розподіл навантаження.
/api/trainings/results	GET	Повертає список завершених тренувань.
/api/trainings/results?id=...	GET	Повертає результати спортсмена.
/api/contacts?userId=...&type=contacts	GET	Повертає список контактів.

Таблиця 3.7 – Таблиця формул розрахунків

Показник	Як розраховується або що показує
Кількість тренувань	Кількість записів у WorkoutResults за вибраний період.
Активний час	Сума workout_active_time за всі завершені тренування.
Калорії	Сума workout_active_calories.
Частота тренувань	Кількість тренувань, поділена на кількість тижнів у періоді.
Регулярність	Відсоток тижнів, у яких було хоча б одне тренування.
Обсяг ваги	Сума weight × reps × number_block_set за результатами вправ.
Баланс м’язів	Групування вправ за м’язовими групами через довідники вправ і м’язів.

Окремо реалізовано аналіз балансу груп м’язів. Маршрут /api/analytics/muscle-balance групує результати вправ за м’язовими групами та повертає кількість вправ, повторень і сумарний обсяг ваги. На клієнтській стороні ці дані відображаються у вигляді радарної діаграми. Для тренера це зручно, бо можна швидко помітити перекіс у навантаженні.

3.5 Результати роботи програмного продукту та приклади екранних форм

У результаті реалізації було створено веб-платформу, яка охоплює повний цикл роботи з тренувальним процесом. Користувач може зареєструватися, заповнити профіль, отримати роль спортсмена або тренера, створювати тренувальні програми, планувати заняття в календарі, проходити тренування та аналізувати результати(див. табл. 3.8).

Таблиця 3.8 – Основні форми застосунку

Екранна форма	Призначення
Сторінка входу та реєстрації	Авторизація користувача і створення облікового запису.
Кабінет тренувань	Перегляд програм, бібліотеки вправ і тренерських дій.
Конструктор програми	Створення блоків, додавання вправ і налаштування параметрів.
Створення тренування	Призначення програми спортсмену на дату й час.
Календар	Перегляд запланованих і виконаних тренувань.
Проходження тренування	Виконання вправ, робота таймера, фіксація сетів і результатів.
Аналітика	Перегляд метрик, трендів, графіків і балансу груп м’язів.
Профіль	Збереження параметрів користувача.

Для спортсмена система працює як особистий кабінет тренувань. Він бачить призначені заняття, може перейти до виконання тренування, фіксувати результати та переглядати статистику. Для тренера платформа виконує роль робочого інструмента: він створює програми, призначає їх спортсменам і контролює прогрес через аналітичний модуль.

4 ТЕСТУВАННЯ СИСТЕМИ

4.1 Мета, підхід і критерії тестування системи

Метою тестування розробленої системи є перевірка коректності її роботи, відповідності функціональним вимогам, стабільності основних сценаріїв використання та зручності взаємодії користувачів із програмним продуктом. Особлива увага приділялася перевірці сценаріїв авторизації, роботи з профілем користувача, розмежування доступу між ролями спортсмена та тренера, створення програм тренувань, роботи календаря та аналітичного модуля.

Тестування проводилося за функціональним підходом, тобто перевірялися основні можливості системи з погляду кінцевого користувача. Для кожного модуля було визначено набір тестових сценаріїв, очікуваний результат та фактичну поведінку системи. Також виконувалася перевірка обробки помилкових або неповних даних, щоб переконатися, що система не допускає некоректних операцій і повідомляє користувача про помилки у зрозумілій формі.

Основними критеріями успішного тестування були:

1. Коректне виконання основних функцій системи.
2. Відповідність результату дій користувача очікуваній поведінці.
3. Захист сторінок і функцій від несанкціонованого доступу.
4. Правильне розмежування можливостей для ролей спортсмена та тренера.
5. Збереження та відображення актуальних даних.
6. Відсутність критичних помилок під час виконання базових сценаріїв.

4.2 Тестування сценаріїв авторизації та роботи з профілем користувача

На цьому етапі перевірялася робота механізмів реєстрації, входу в систему, виходу з облікового запису та редагування профілю користувача. Авторизація є одним із ключових модулів системи, оскільки саме вона забезпечує ідентифікацію користувача та відкриває доступ до персоналізованих функцій.

При запуску програми у перший раз нас зустрічає сторінка логіну(див. рис. 4.1). Щоб перейти на сторінку реєстрації треба клікнути на кнопку “Зареєструватись”

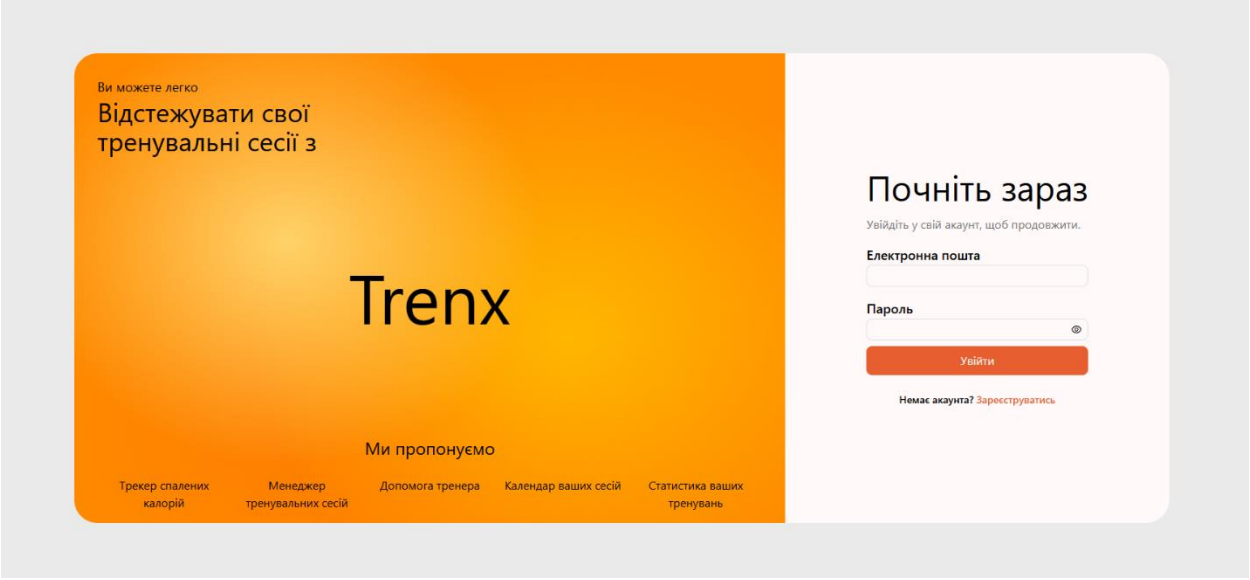


Рисунок 4.1 – Сторінка логіну.

При переході на сторінку реєстрації (див. рис. 4.2) бачимо поля що необхідно заповнити: Ім'я, Ім'я користувача, Електронна пошта, Пароль та Повторіть пароль. Заповняємо їх відповідними даними.

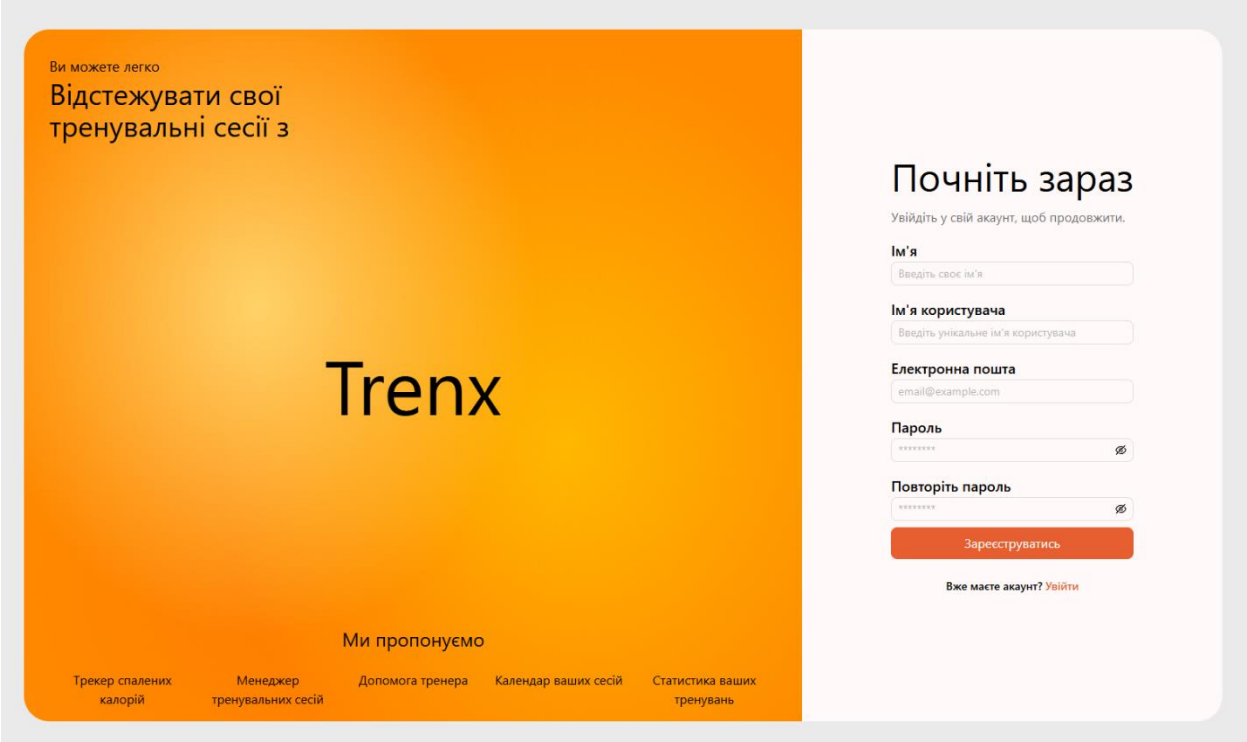


Рисунок 4.2 – Сторінка реєстрації.

Після натискання на кнопку “Зареєструватись” ми потрапляємо на сторінку з календарем. Це означає що реєстрація успішна та новий профіль користувача було створено. Щоб це перевірити виходимо з акаунта та переходимо на сторінку логіну(див. рис. 4.3). Вводимо електронну пошту та пароль і натискаємо на кнопку “Увійти”. Нас перекидує на сторінку календаря, що свядчить про успіх логіну.

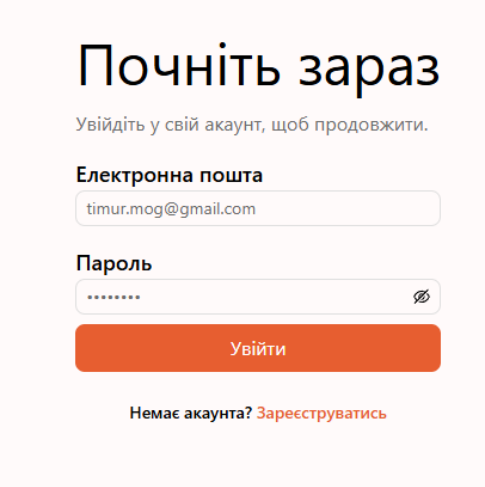


Рисунок 4.3 – Заповнена форма логіну.

Якщо ввести неправильні дані, то буде відображено повідомлення про помилку. Наприклад якщо ввести неправильний пароль або електронну пошту(див рис. 4.4, 4.5).

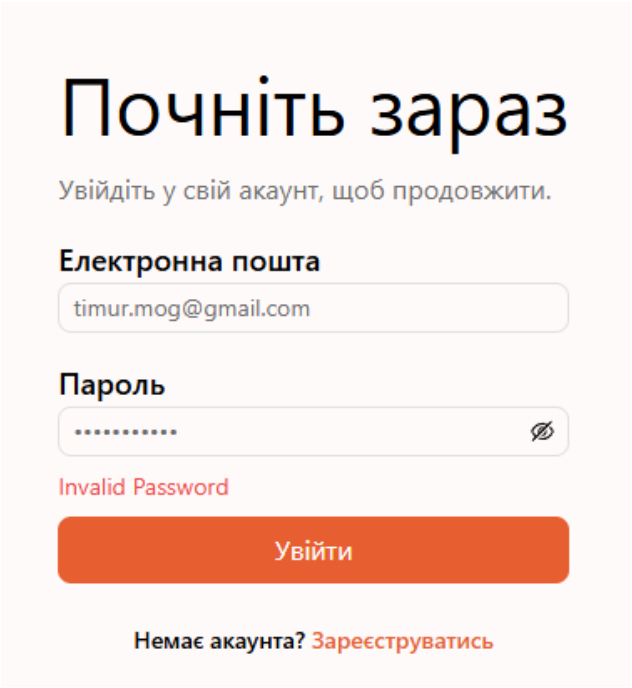


Рисунок 4.4 – Повідомлення про неправильний пароль.

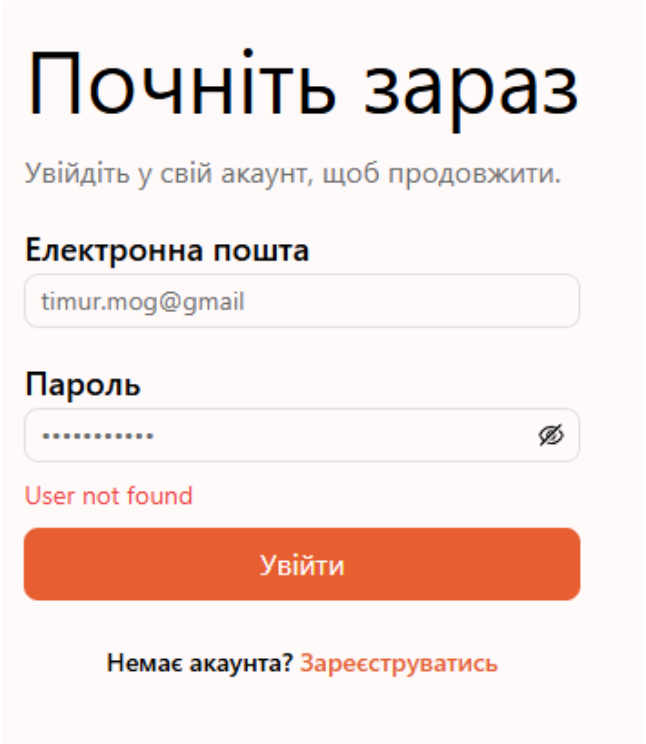


Рисунок 4.5 – Повідомлення про неправильну електронну пошту.

Далі треба протестувати профіль користувача(див. рис. 4.6). Щоб перейти на цю сторінку треба натиснути на кнопку “Профіль” у навігаційній панелі.

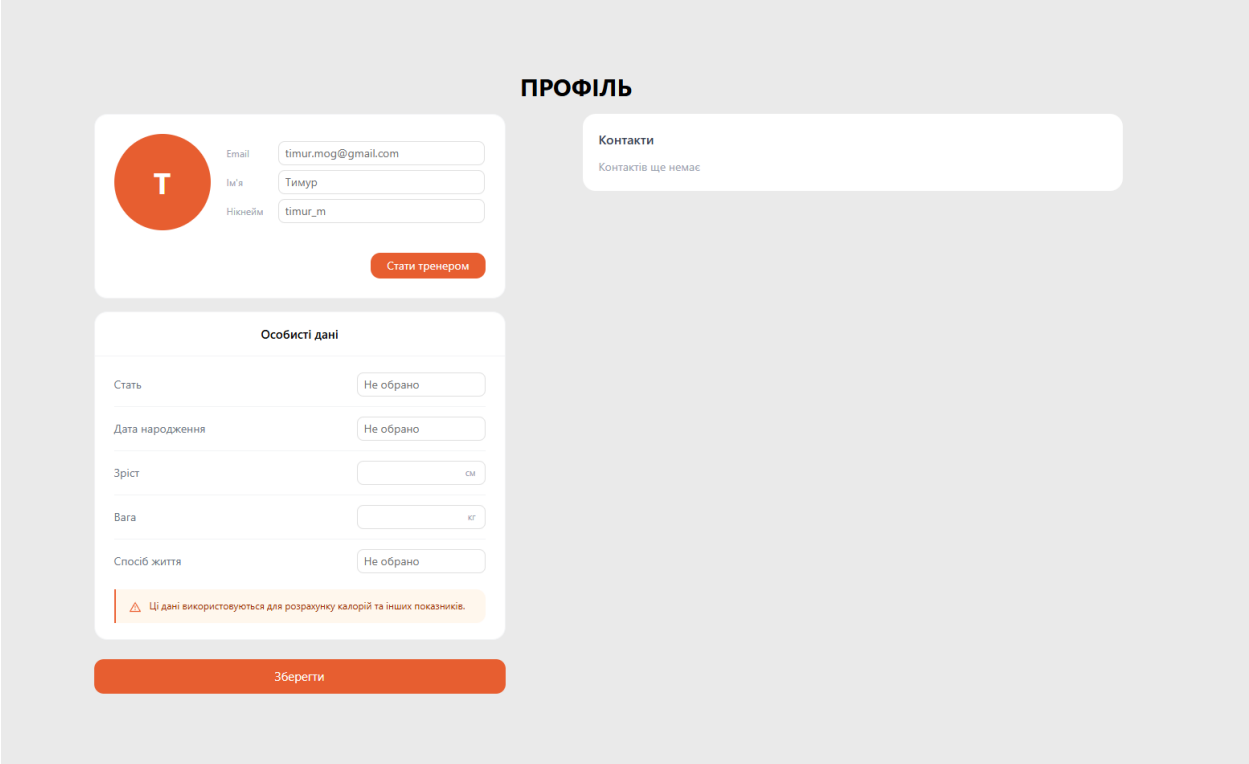


Рисунок 4.6 – Сторінка профіля користувача.

Для перевірки функціоналу, змінимо наступні поля: нікнейм, стать, дата народження, зріст, вага, спосіб життя(див. рис 4.7). Натискаємо на кнопку “Зберегти” та перезавантажуємо сторінку щоб перевірити.

Далі натиснемо на кнопку “Стати тренером”. Праворуч бачимо що в нас з’явилась можливість додавати інших користувачів до контактів(див. рис. 4.8). Вводимо електронну адресу та натискаємо “Запросити”, поки що у запрошення статус “Очікує”, статус зміниться або на “Прийнято”, або на “Відхилено” в залежності від відповіді користувача якому було надіслане запрошення.

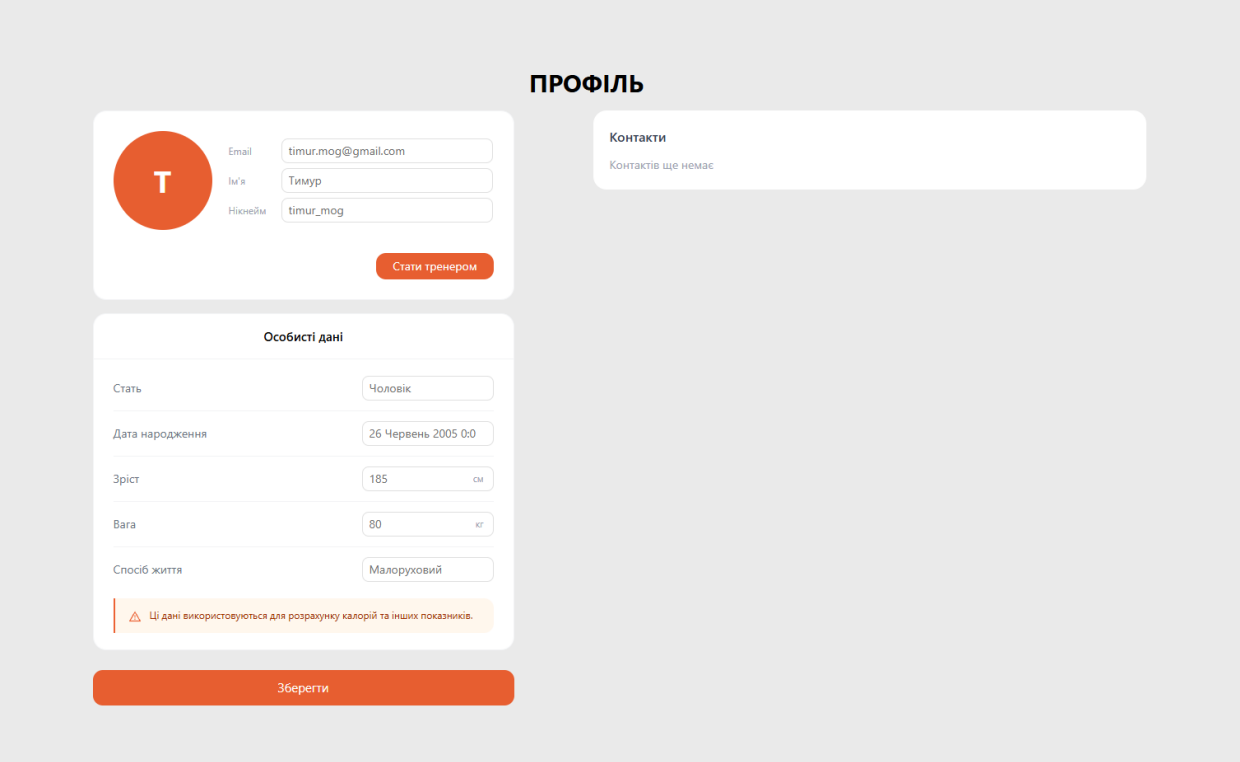


Рисунок 4.7 – Змінені поля профіля.

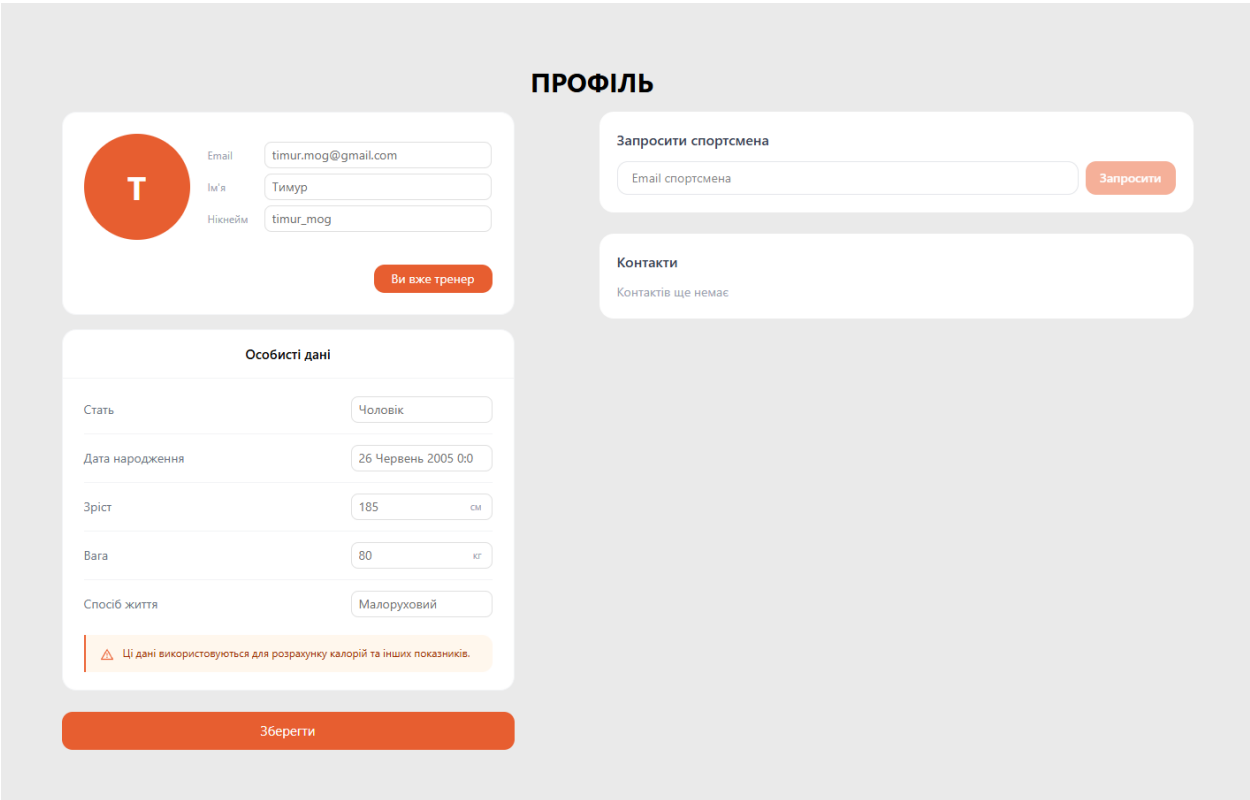


Рисунок 4.8 – Профіль тренера.

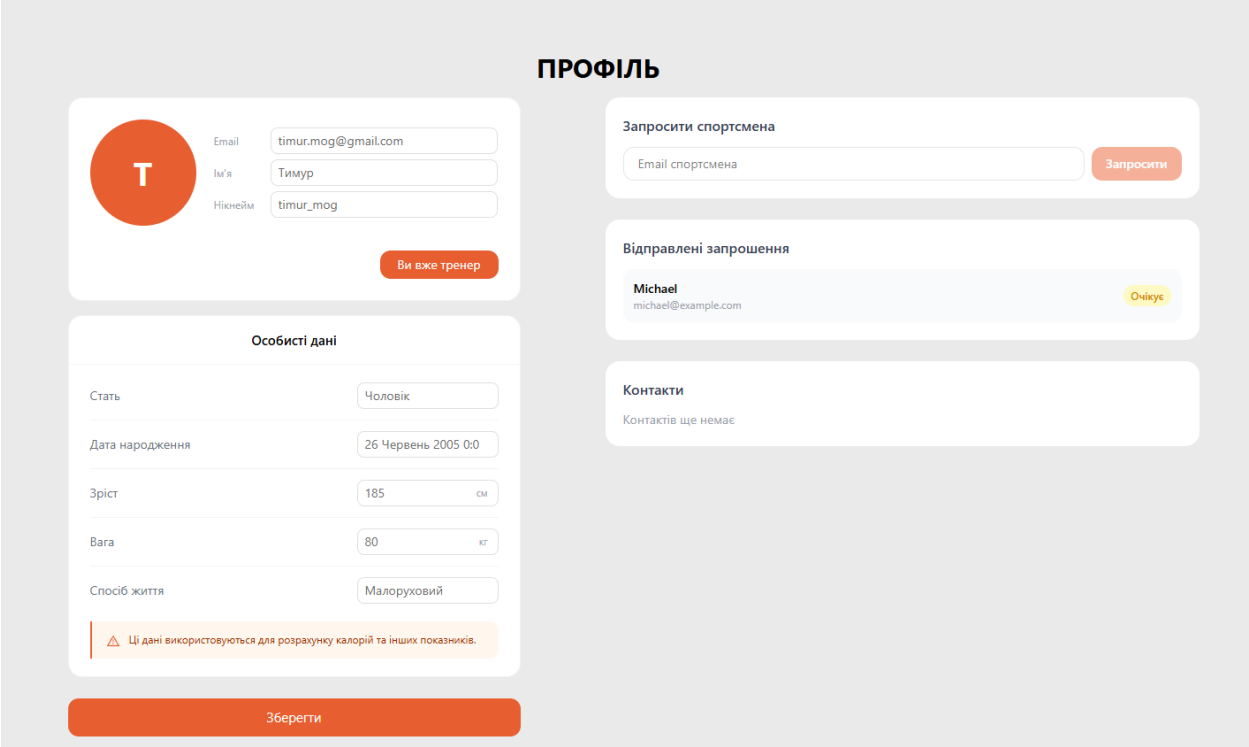


Рисунок 4.9 – Відправлене запрошення.

Під час тестування було перевірено, що користувач може створити обліковий запис, увійти до системи з правильними даними, або отримати повідомлення про помилку при неправильному введенні електронної пошти або пароля.

У результаті тестування встановлено, що модуль авторизації працює коректно. Система не дозволяє отримати доступ до захищених сторінок без входу в обліковий запис, а дані профілю користувача зберігаються та оновлюються відповідно до виконаних дій.

4.3 Тестування створення тренувань

Модуль створення тренувань призначений для формування індивідуальних або загальних планів занять. У межах тестування перевірялася можливість створення програми, додавання вправ, зазначення кількості підходів, повторень, тривалості або інших параметрів тренування.

Щоб створити тренування треба перейти на сторінку тренувань натиснувши на панелі навігації кнопку “Тренування”. Далі треба натиснути плюс у правому нижньому куті екрану(див. рис.4.10).



Рисунок 4.10 – Кнопка для створення тренувань.

Бачимо сторінку для створення тренувань(див. рис. 4.11). Натискаємо кнопку “Створити програму”. Та потрапляємо на сторінку “Створення програми”(див. рис. 4.12).

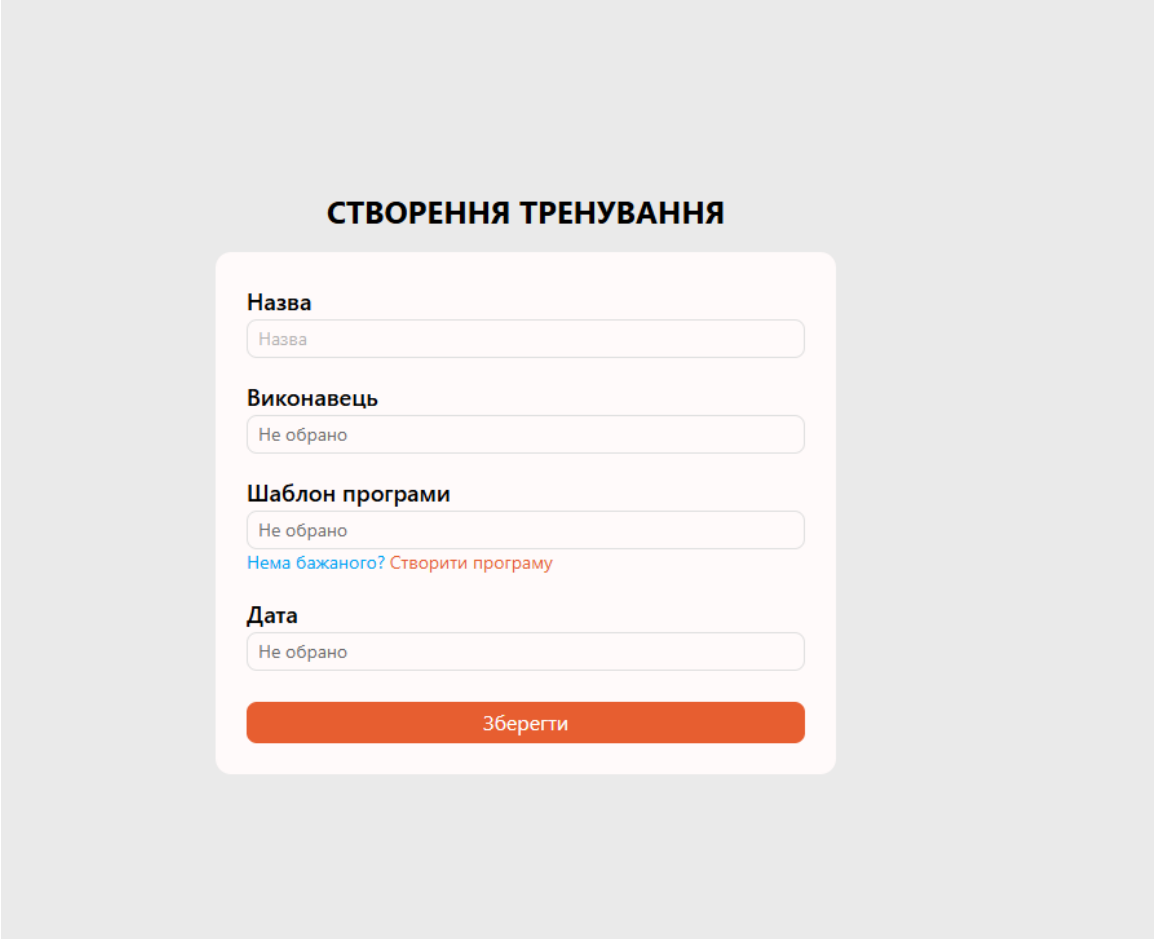


Рисунок 4.11 – Сторінка “Створення тренування”.

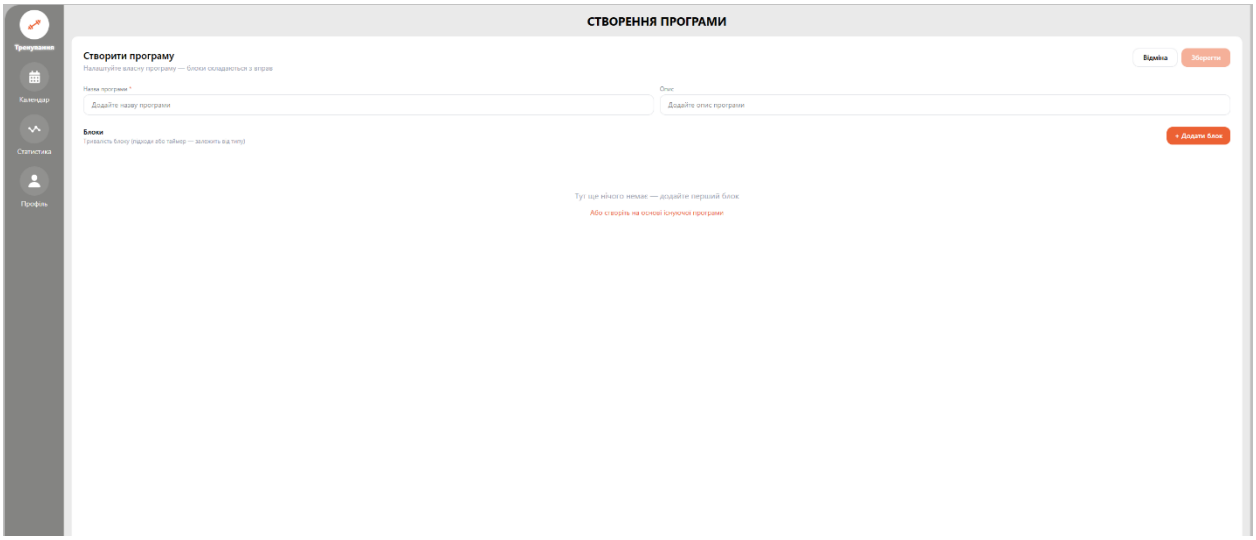


Рисунок 4.12 – Сторінка “Створення програми”.

Заповнимо назву тренування та створимо три блоки: розминка, силова та кардіо(див. рис. 4.13). До розминки додаємо 5хв планки(див. рис. 4.14). До другого – Французський жим, Жим лежачи та Розведення гантелей лежачи(див. рис. 4.15). У блок кардіо додано 15хв на дорожці(див. рис. 4.16). Натискаємо кнопку “Зберегти” та нас повертає на сторінку “Створення тренувань”.

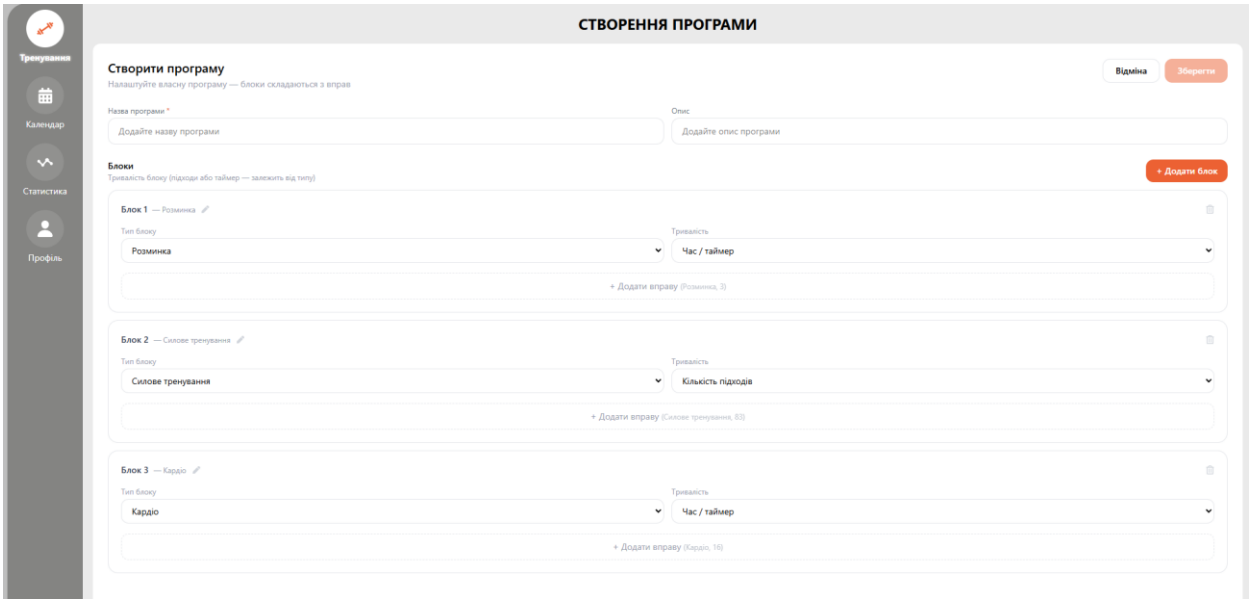


Рисунок 4.13 – Створені тренувальні блоки.

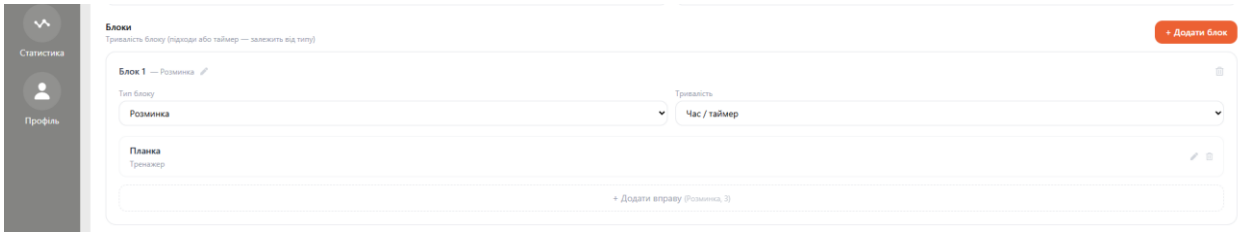


Рисунок 4.14 – Блок “Розминка”.

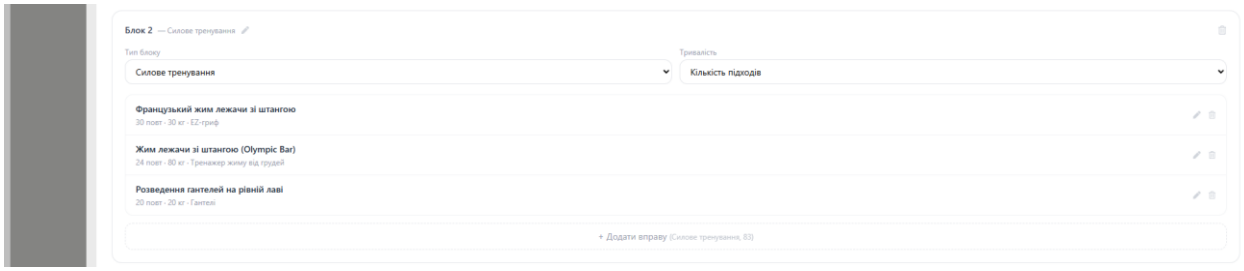


Рисунок 4.15 – Блок “Силова”.

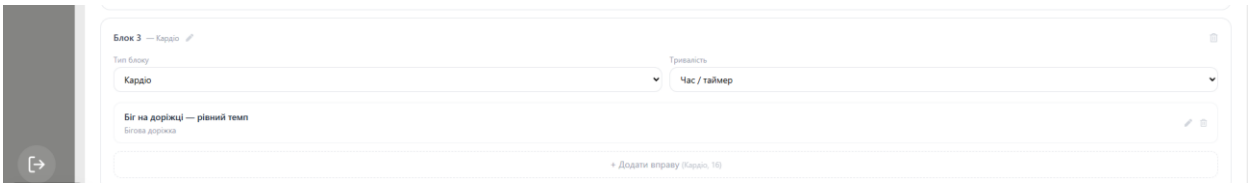


Рисунок 4.16 – Блок “Кардіо”.

Заповняємо всі необхідні поля, обираємо зі списку програм нову створену тренувальну програму, виконавцем ставимо спортсмена зі списку контактів та ставимо тренування на 18:00 наступного дня(див. рис. 4.17). Натискаємо кнопку “Зберегти”. Api-запит повертає статус 200, тренування успішно створене.

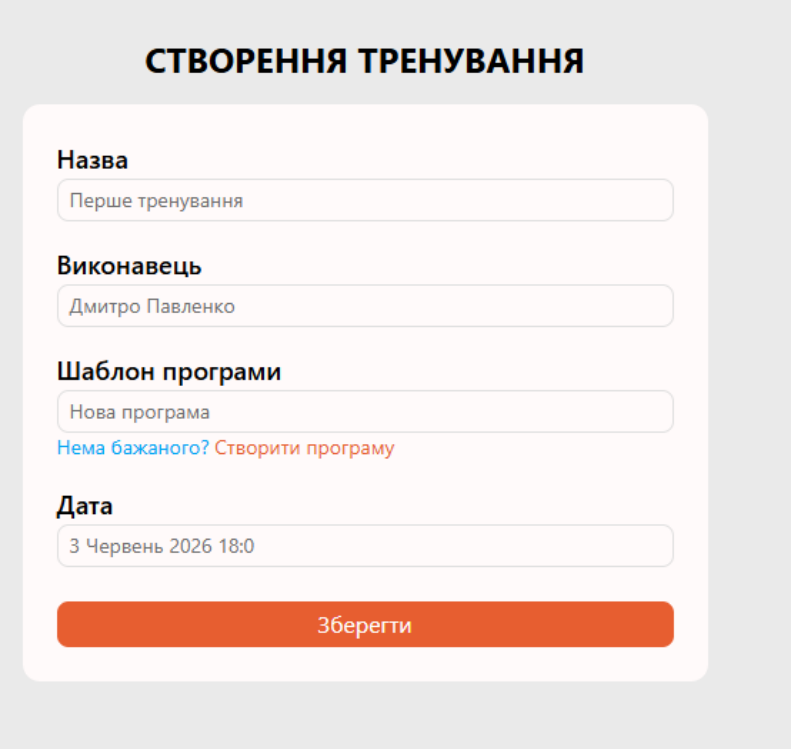


Рисунок 4.17 – Заповнена форма.

За результатами тестування підтверджено, що модуль створення програм тренувань працює стабільно. Дані зберігаються коректно, а система контролює правильність введення основної інформації.

4.4 Тестування календаря тренувань

Календар тренувань використовується для планування занять, перегляду розкладу та контролю виконання тренувальних програм. Тестування цього модуля було спрямоване на перевірку правильного відображення тренувань за датами.

Щоб побачити календарь тренувань треба натиснути на кнопку “Календар” у навігаційному меню. Тренер бачить всі свої тренування та тренування спортсменів у своїх контактах, бачимо наприклад тренування створене в минулому розділі(див. рис. 4.18).

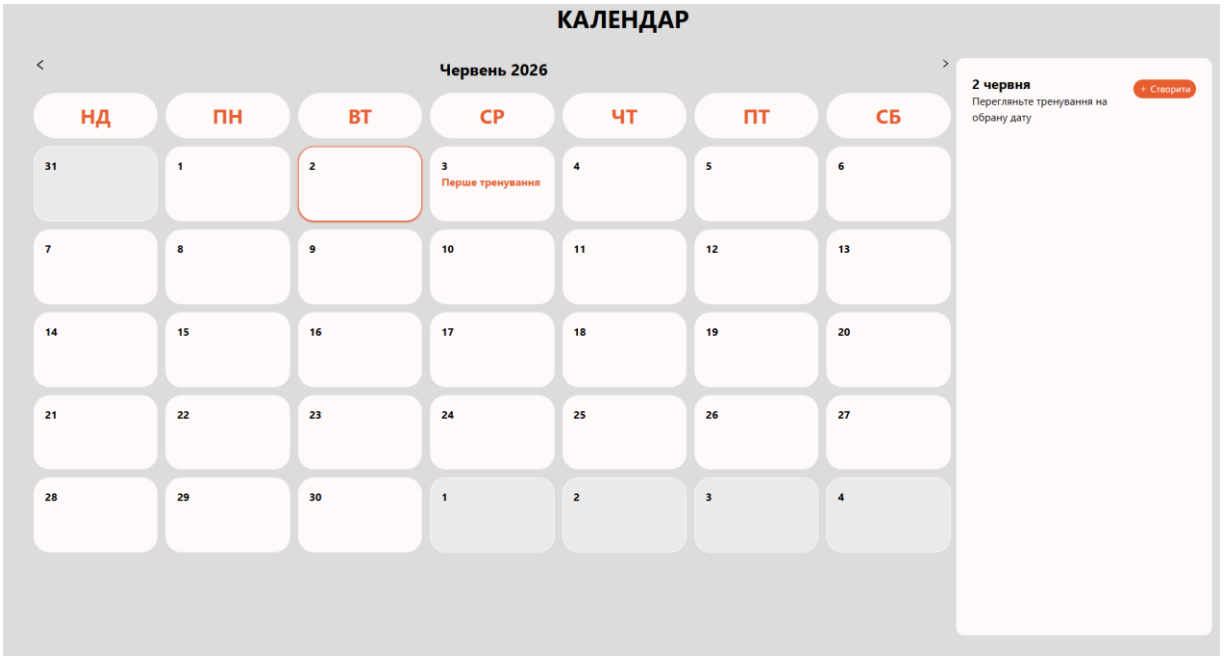


Рисунок 4.18 – Календар тренувань.

Якщо натиснути на 3 червня, можемо праворуч побачити, що на цей день є одне тренування на 18:00(див. рис. 4.19). Закреслені тренування позначаються закресленим текстом.

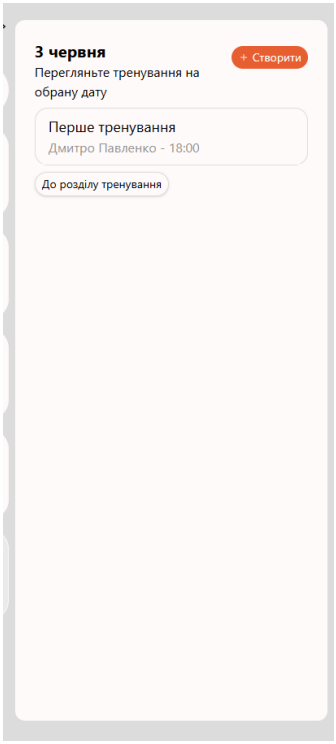


Рисунок 4.19 – Список тренувань за день.

Тестування показало, що календар коректно відображає заплановані тренування та дозволяє користувачам керувати розкладом.

4.5 Тестування аналітичного модуля

Аналітичний модуль призначений для відображення результатів тренувального процесу, оцінки прогресу спортсмена та формування загального уявлення про ефективність виконання програм. У межах тестування перевірялася правильність відображення статистичних даних, графіків, показників виконання тренувань та інших аналітичних елементів.

Щоб подивитись статистику, треба натиснути кнопку “Статистика” на боковій панелі. Тренер може бачити статистику будь кого зі спортсменів у себе в контактах(див. рис. 4.20).

Натискаємо на кнопку “Статистика вправ” та переходимо на відповідну сторінку(див. рис. 4.21). Бачимо список всіх вправ що робить спортсмен, персональні рекорди по вправі та історію виконання вправ. Також тренер може подивитись графік зміни ваги при виконанні певної вправи(див. рис. 4.22), якщо вона передбачає роботу з навантаженням.

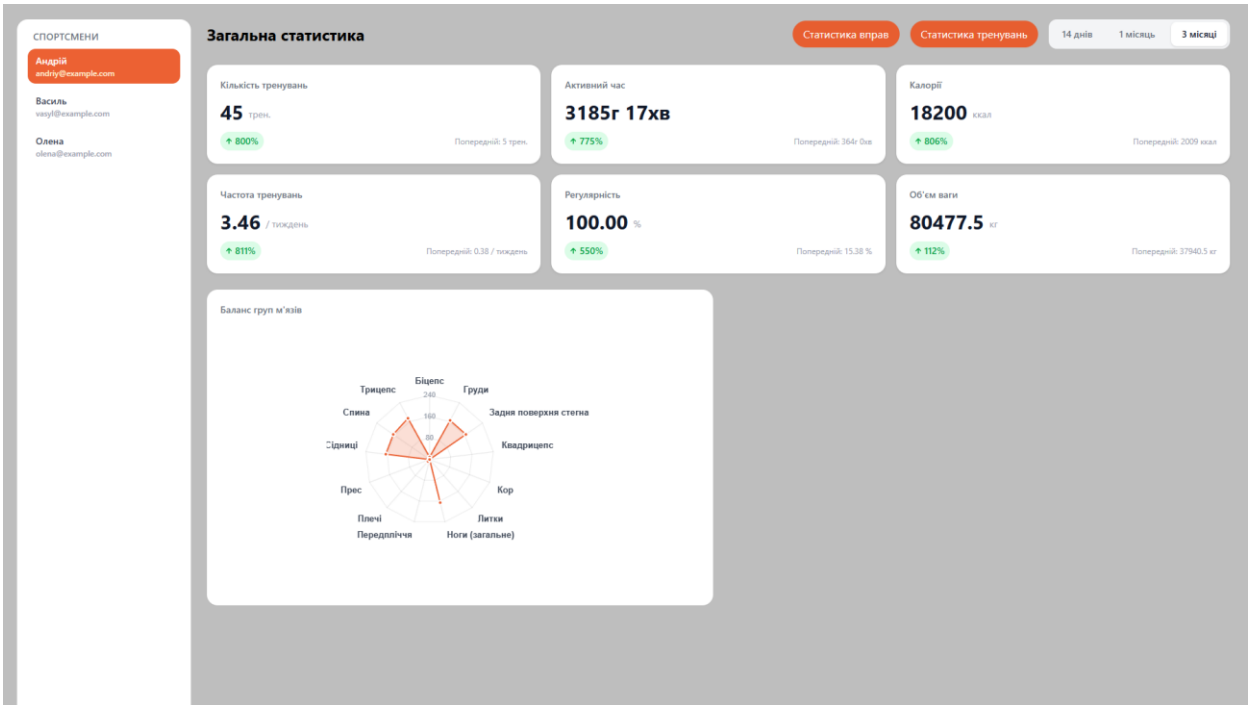


Рисунок 4.20 – Сторінка “Статистика”.



Рисунок 4.21 – Сторінка “Статистика вправ”.

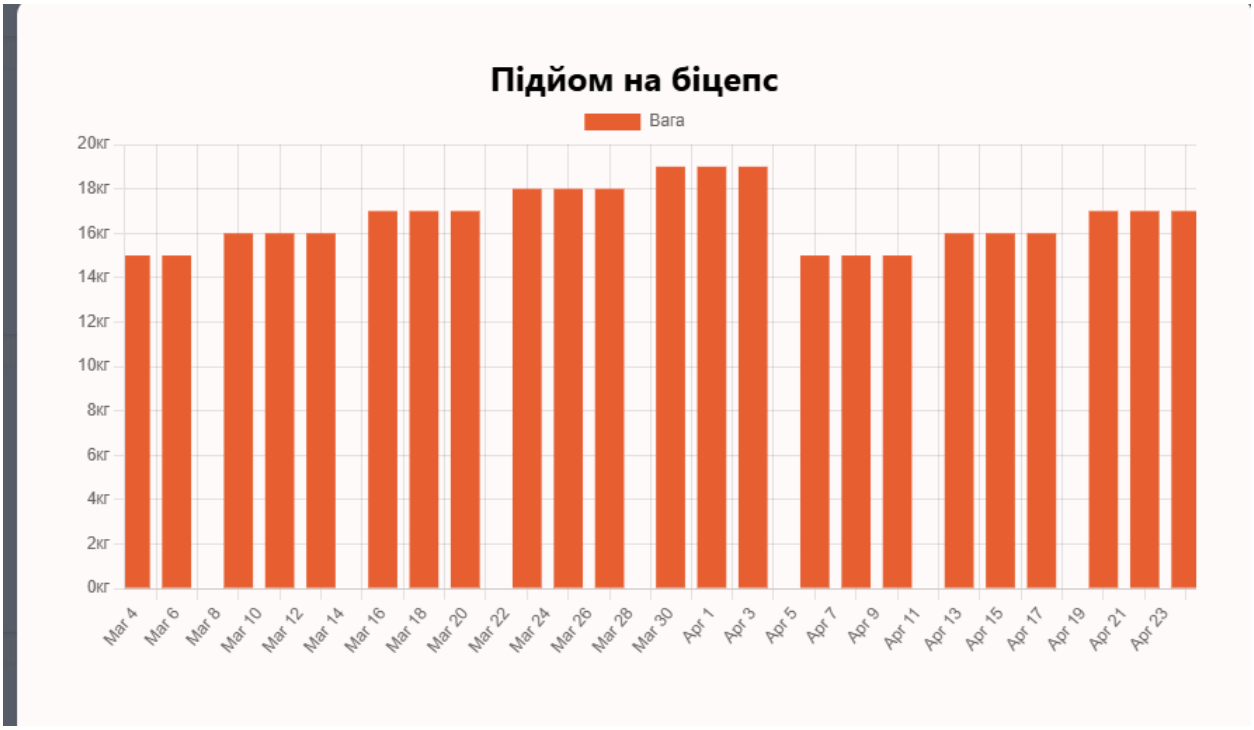


Рисунок 4.22 – Графік зміни ваги.

На сторінці “Статистика тренувань” тренер може відкрити діалогове вікно з двома графіками: зміна тривалості активного часу та часу відпочинку, а також графік зміни загальної ваги в роботі(див. рис. 4.23).

Проведене тестування підтвердило, що основні функціональні модулі системи працюють відповідно до визначених вимог, забезпечують коректну

взаємодію користувачів із системою та підтримують необхідний рівень надійності для практичного використання.



Рисунок 4.23 – Графіки зі сторінки “Статистика тренувань”.

ВИСНОВОК

У дипломній роботі було розроблено веб-платформу для планування та моніторингу тренувального процесу з акцентом на аналітичний модуль тренера. У ході виконання роботи було проаналізовано предметну область, визначено потреби спортсменів і тренерів, розглянуто існуючі підходи до організації тренувань та сформульовано вимоги до програмного продукту.

Було спроєктовано архітектуру системи, яка поєднує клієнтський інтерфейс, серверну логіку та реляційну базу даних. Структура даних забезпечує зв’язок між користувачами, ролями, тренувальними програмами, блоками вправ, розкладом занять і фактичними результатами виконання. Такий підхід дозволяє зберігати тренувальний процес не як набір окремих записів, а як цілісну систему даних, придатну для подальшого аналізу.

У системі реалізовано авторизацію користувачів, рольовий доступ, профіль користувача, модуль контактів між тренером і спортсменом, створення програм тренувань, календарне планування, проходження тренування та збереження результатів. Особливе значення має аналітичний модуль, який формує показники активності, регулярності, частоти тренувань, калорійності, обсягу силового навантаження та балансу навантаження за групами м’язів.

Розроблена платформа демонструє практичну цінність, оскільки допомагає тренеру швидше отримувати узагальнену картину стану спортсмена, аналізувати виконання програми та приймати рішення щодо корекції тренувального плану на основі фактичних даних. Таким чином, поставлена мета дипломної роботи була досягнута, а розроблений програмний продукт може бути використаний як основа для подальшого розвитку системи підтримки тренувального процесу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. McGuigan H., Hassmén P., Rosic N., Stevens C.J. Training monitoring methods used in the field by coaches and practitioners: A systematic review// International Journal of Sports Science & Coaching, 2020. – Том. 15, № 3. – с. 439–451.
2. American College of Sports Medicine. ACSM's Guidelines for Exercise Testing and Prescription// Philadelphia: Wolters Kluwer Health, 2025. – 12th ed. – 680 с.
3. Currier B.S., D'Souza A.C., Fiatarone Singh M.A., Lowisz C.V., Rawson E.S., Schoenfeld B.J. et al. American College of Sports Medicine Position Stand. Resistance Training Prescription for Muscle Function, Hypertrophy, and Physical Performance in Healthy Adults: An Overview of Reviews// Medicine & Science in Sports & Exercise, 2026. – Том. 58, № 4. – с. 851–872.
4. American College of Sports Medicine. Progression Models in Resistance Training for Healthy Adults// Medicine & Science in Sports & Exercise, 2009. – Том. 41, № 3. – с. 687–708.
5. Drew M.K., Finch C.F. The Relationship Between Training Load and Injury, Illness and Soreness: A Systematic and Literature Review// Sports Medicine, 2016. – Том. 46, № 6. – с. 861–883.
6. Saw A.E., Main L.C., Gastin P.B. Monitoring the Athlete Training Response: Subjective Self-Reported Measures Trump Commonly Used Objective Measures: A Systematic Review// British Journal of Sports Medicine, 2016. – Том. 50, № 5. – с. 281–291.
7. Schoenfeld B.J., Ogborn D., Krieger J.W. Dose-response Relationship Between Weekly Resistance Training Volume and Increases in Muscle Mass: A Systematic Review and Meta-analysis// Journal of Sports Sciences, 2017. – Том. 35, № 11. – с. 1073–1082.

8. Currier B.S., McLeod J.C., Banfield L., Beyene J., Welton N.J., D'Souza A.C. et al. Resistance Training Prescription for Muscle Strength and Hypertrophy in Healthy Adults: A Systematic Review and Bayesian Network Meta-analysis// British Journal of Sports Medicine, 2023. – Том. 57, № 18. – с. 1211–1220.
9. Ferguson T., Olds T., Curtis R., Blake H., Crozier A.J., Dankiw K. et al. Effectiveness of Wearable Activity Trackers to Increase Physical Activity and Improve Health: A Systematic Review of Systematic Reviews and Meta-analyses// The Lancet Digital Health, 2022. – Том. 4, № 8. – с. 615– 626.
10. Jones M., Bradley J., Sakimura N. JSON Web Token (JWT): RFC 7519// Internet Engineering Task Force, 2015. – URL: https://datatracker.ietf.org/doc/html/rfc7519

ДОДАТОК А. ДОВІДКА ПРО ВПРОВАДЖЕННЯ