

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ Д
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет автоматизації інформаційних
технологій

Кафедра інформаційних технологій

(назва випускової кафедри)

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВР**

на тему:

**Web-додаток для комп'ютерного моніторингу та аналізу
спортивних досягнень користувача**

Даллакян Анастасія Олександрівна

Київ 2025 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій

(кафедра)

ЗАТВЕРДЖУЮ

Завідувачка кафедри ІТ

д.т.н., доцент Гончаренко Т.А.

„___” _____ 2025 року

**КВАЛІФІКАЦІЙНА РОБОТА
ЗДОБУВАЧА СТУПЕНЯ ВИЩОЇ ОСВІТИ БАКАЛАВР**

на тему:

**«Web-додаток для комп'ютерного моніторингу та обробки
результатів досягнень учасників спортивних заходів»**

Я як здобувач вищої освіти КНУБА розумію і підтримую політику закладу з академічної доброчесності. Я не надавав(-ла) і не одержував(-ла) недозволену допомогу під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

Здобувач

Даллакян Анастасія Олександрівна

122 «Комп'ютерні науки»

(спеціальність)

Інформаційні управляючі системи і технології

(освітня програма)

Групи КН-21

Керівник Рябчун Ю.В.

(прізвище та ініціали)

Доктор філософії

(вчене звання, науковий ступінь)

Рецензент к.т.н., доц. Доля О.В.

(Прізвище та ініціали)

Ідентичність підтверджую

Київ, 2025 р.

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БУДІВНИЦТВА І АРХІТЕКТУРИ

| | |
|--------------------|---|
| Факультет: | Автоматизації інформаційних технологій |
| Випускова кафедра: | Інформаційних технологій |
| Освітній ступінь: | Бакалавр |
| Спеціальність: | Комп'ютерні науки |
| Освітня програма: | Інформаційні управляючі системи та технології |

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТ

Тетяна ГОНЧАРЕНКО

„___” _____ 2025 року

ЗАВДАННЯ

ДО ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВР

| | |
|---|----------------------------------|
| | Даллакян Анастасії Олександрівні |
| 1. Тема роботи Web-додаток для комп'ютерного моніторингу та обробки результатів досягнень учасників спортивних заходів затверджена наказом ректора КНУБА № 235/23/25 від «14» лютого 2025 року | |
| 2. Керівник роботи | Рябчун Юлія Володимирівна, PhD |

3. Строк подання Здобувачем роботи до захисту 26/05/2025

4. Зміст пояснювальної записки за розділами:

1. Аналіз предметної області та постановка задачі

2. Проектування та розробка web-додатка для комп'ютерного моніторингу та аналізу спортивних досягнень користувача

3. Впровадження та реалізація

4. Аналіз отриманих результатів та напрями подальшого розвитку системи

5. Графічний матеріал за розділами:

Р.1 Аналіз предметної області та постановка задачі

Р.2 Проєктування та розробка web-додатка для комп'ютерного моніторингу та аналізу спортивних досягнень користувача

Р.3 Впровадження та реалізація

Р.4 Аналіз отриманих результатів та напрями подальшого розвитку системи

6. Календарний план виконання роботи:

| Види робіт та їх зміст | Дата виконання |
|---|----------------|
| Розділ 1 Аналіз предметної області та постановка задачі | 28.01.2025 |
| Розділ 2 Проєктування та розробка web-додатка для комп'ютерного моніторингу та аналізу спортивних досягнень користувача | 14.04.2025 |
| Розділ 3 Впровадження та реалізація | 15.05.2025 |
| Розділ 4 Аналіз отриманих результатів та напрями подальшого розвитку системи | 15.05.2025 |
| Остаточне оформлення роботи | 24.05.2025 |
| Направлення роботи для перевірки на плагіат | 24.05.2025 |
| Попередній захист роботи на випусковій кафедрі | 26.05.2025 |
| Направлення роботи на рецензування | 26.05.2025 |

7. Консультанти розділів атестаційної випускної роботи

| Розділ | Прізвище, ініціали та посада консультанта | Перевірив | |
|----------|---|------------|--------|
| | | дата | підпис |
| Розділ 1 | Рябчун Ю.В., доц.каф.ІТ | 28.01.2025 | |
| Розділ 2 | Рябчун Ю.В., доц.каф.ІТ | 14.04.2025 | |
| Розділ 3 | Рябчун Ю.В., доц.каф.ІТ | 15.05.2025 | |
| Розділ 4 | Рябчун Ю.В., доц.каф.ІТ | 24.05.2025 | |

8. Дата видачі завдання листопад 2025 р.

| | | | |
|--------------|----------|--|------------------------|
| Зав. кафедри | | | Гончаренко Т.А. |
| | (підпис) | | (прізвище та ініціали) |
| Керівники | | | Рябчун Ю.В. |
| | (підпис) | | (прізвище та ініціали) |
| Здобувач | | | Даллакян А.О. |
| | (підпис) | | (прізвище та ініціали) |

АНОТАЦІЯ

Даллакян А.О. Web-додаток для комп'ютерного моніторингу та обробки результатів досягнень учасників спортивних заходів.

Атестаційна випускна робота бакалавра за спеціальністю 122 «Комп'ютерні науки», освітня програма «Інформаційні управляючі системи і технології». – Київський національний університет будівництва та архітектури. – Київ, 2025.

Дана дипломна робота присвячена розробці web-додатку для комп'ютерного моніторингу та обробки результатів досягнень учасників спортивних заходів. У роботі розглядаються етапи проектування та реалізації основних функціональних можливостей додатку, таких як управління профілями учасників, реєстрація результатів, аналіз статистики та формування звітів. Експериментальна частина роботи включає тестування системи, оцінку її продуктивності, зручності використання та точності обробки даних. На основі отриманих результатів сформовані висновки та рекомендації щодо вдосконалення системи, розширення її функціональності та інтеграції з іншими платформами.

Робота викладена на 79 аркушах, 2 таблиці, 39 рисунків, список використаної літератури із 30 найменувань.

Ключові слова: web-додаток, моніторинг, спортивні заходи, обробка результатів, статистика, HTML, CSS, JavaScript, Node.js, MongoDB.

SUMMARY

Dallakian A. Development of a web application for computer monitoring and processing of the achievements of sports event participants. Bachelor's thesis in the specialty: 122 "Computer Science," specialization: "Information Management Systems and Technologies." – Kyiv National University of Construction and Architecture. – Kyiv, 2025.

This thesis is dedicated to the development of a web application for computer monitoring and processing the achievements of sports event participants. The work covers the design and implementation of key functionalities, including participant profile management, result registration, statistical analysis, and report generation. The experimental part includes system testing, evaluation of its performance, usability, and data processing accuracy. Based on the obtained results, conclusions and recommendations are made for improving the system, expanding its functionality, and integrating it with other platforms.

The work is set out on 79 pages, 2 tables, 39 figures, a list of references of 30 titles.

Keywords: web application, monitoring, sports events, results processing, statistics, HTML, CSS, JavaScript, Node.js, MongoDB.

ЗМІСТ

Помилка! Закладку не визначено.

| | |
|--|----|
| ВСТУП | 10 |
| Розділ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ | 12 |
| 1.1 Постановка та аналіз проблеми | 12 |
| 1.2 Характеристика спортивних заходів як об’єкта моніторингу | 13 |
| 1.3 Аналіз потреб користувачів на прикладі готових рішень..... | 14 |
| 1.4 Визначення вимог до системи та концептуальна модель проекту | 19 |
| 1.5 Постановка задач та очікувані результати | 24 |
| Розділ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА WEB-ДОДАТКА ДЛЯ КОМП’ЮТЕРНОГО МОНІТОРИНГУ ТА АНАЛІЗУ СПОРТИВНИХ ДОСЯГНЕНЬ КОРИСТУВАЧА | 28 |
| 2.1. Вибір технологічного стеку | 28 |
| 2.2. Архітектура системи | 31 |
| 2.3. Проєктування бази даних | 36 |
| 2.4. Функціональні можливості веб-додатку..... | 42 |
| РОЗДІЛ 3. ВПРОВАДЖЕННЯ ТА РЕАЛІЗАЦІЯ..... | 48 |
| 3.1 Загальний опис реалізації..... | 48 |
| 3.2 Реалізація клієнтської частини | 50 |
| 3.3 Переваги розробленого web додатку | 61 |
| РОЗДІЛ 4. АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ ТА НАПРЯМИ ПОДАЛЬШОГО РОЗВИТКУ СИСТЕМИ..... | 64 |
| 4.1. Аналіз ефективності створеного програмного забезпечення | 64 |
| 4.2. Перспективи масштабування та інтеграції з іншими системами..... | 69 |
| 4.3. Пропозиції щодо майбутніх удосконалень..... | 70 |
| 4.4. Висновки щодо ефективності розробки та перспектив розвитку | 72 |

| | |
|---------------------------------|----|
| ВИСНОВКИ..... | 74 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... | 76 |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПЗ – програмне забезпечення;

БД – база даних;

OS (Operation System) – операційна система;

UML (Unified Modelling Language) – універсальна мова моделювання;

ERD (entity-relationship diagram, ER-діаграма) – діаграма сутність-зв'язок;

Java – об'єктно-орієнтована мова програмування;

SQLite – компактна вбудована реляційна база даних;

Android Studio – інтегроване середовище розробки (IDE) для платформи

Android.

ВСТУП

Сучасний підхід до моніторингу спортивних досягнень потребує інноваційних рішень, які дозволяють користувачам ефективно відстежувати, аналізувати та покращувати свої результати. Люди, які займаються спортом, тренери та аналітики стикаються з викликами, пов'язаними із забезпеченням точного обліку, аналізу даних та оцінки прогресу. Традиційні методи, які базуються на ручному введенні результатів або використовують паперові журнали, є неефективними, не гарантують необхідної швидкості та точності обробки інформації.

Автоматизація моніторингу спортивних досягнень відкриває нові можливості для аналізу фізичної активності, встановлення персональних цілей та об'єктивного оцінювання прогресу користувачів. Веб-технології не дозволяють створювати інтерактивні платформи, які забезпечують доступ до даних у реальному часі, візуалізацію результатів, генерацію статистики та інтеграцію з іншими сервісами й пристроями.

Дана дипломна робота присвячена розробці веб-додатку, яка дозволить автоматизувати процеси моніторингу та аналізу спортивних досягнень людини. Основна мета розробки – створення зручного інструменту для користувачів, які прагнуть контролювати свої результати, аналізувати фізичну активність і підвищувати ефективність тренувань. Запропонування рішень та обробку даних, побудову збереження статистики, візуалізацію показників і створення адаптивного інтерфейсу для взаємодії з користувачами.

Метою роботи є створення програмного забезпечення, яке дозволяє користувачам вести облік спортивних досягнень, відстежувати зміни у фізичних показниках та покращувати процес аналізу тренувань. У ході дослідження буде проведено аналіз вимог до системи, розроблено її архітектуру, реалізовано основний функціонал та оцінено ефективність запропонованого рішення.

Завдання дослідження:

1. Провести аналіз існуючих веб-систем для моніторингу спортивних досягнень, дослідити їхні переваги та недоліки.

2. Розробити архітектуру веб-додатку для моніторингу та аналізу спортивних досягнень користувачів.

3. Реалізувати функціональні модулі веб-додатку, зокрема інтерфейс користувача, систему зберігання даних та аналітичні інструменти.

4. Виконати впровадження та тестування веб-додатку в модельованих умовах.

5. Провести аналіз ефективності розробленого рішення, оцінити його вплив на якість обробки спортивних даних та зручність роботи користувачів.

Об'єкт дослідження – процес моніторингу та аналізу спортивних досягнень людини.

Предмет дослідження – веб-додаток для комп'ютерного моніторингу та аналізу спортивних досягнень користувача.

Розділ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

Сучасний розвиток інформаційних технологій відкриває нові можливості для оптимізації процесів у різних сферах, включаючи спорт та фізичну активність. Зростання популярності здорового способу життя, регулярних тренувань та персонального моніторингу фізичних показників потребує впровадження інноваційних інструментів для збору, аналізу та збереження даних про спортивні досягнення людини.

Значення якісної системи моніторингу спортивних досягнень полягає не лише у підвищенні точності та прозорості обліку результатів, але й у створенні зручного середовища для користувачів, які прагнуть аналізувати свій прогрес. Водночас існуючі рішення часто не враховують специфічні потреби користувачів, мають обмежений функціонал або не відповідають сучасним вимогам щодо зручності, гнучкості та інтеграції з іншими сервісами й пристроями (фітнес-трекерами, мобільними застосунками тощо).

Аналіз предметної області дозволяє виявити недоліки існуючих рішень, визначити основні вимоги до системи та забезпечити її відповідність очікуванням користувачів. Це є основою для розробки ефективного програмного забезпечення, яке вирішує проблему обліку, аналізу та візуалізації спортивних досягнень людини.

У цьому розділі проведено аналіз предметної області, вивчено наявні системи моніторингу фізичної активності, визначено їхні недоліки, а також сформульовано основну задачу, яку має вирішити розроблений веб-додаток.

1.1 Постановка та аналіз проблеми

У сучасному світі, де фізична активність та спорт є важливими складовими здорового способу життя, потреба в автоматизації процесів моніторингу спортивних досягнень людини стає все більш актуальною. Це пов'язано зі зростанням популярності індивідуальних тренувань, підвищенням вимог до аналізу фізичних показників та необхідністю точного контролю власних спортивних результатів.

Попри наявність програмного забезпечення для обліку спортивних досягнень, існує низка проблем:

- Розрізненість рішень: багато платформ мають обмежену функціональність або орієнтовані лише на конкретні види спорту.
- Низька інтерактивність: відсутність інструментів для створення звітів у реальному часі або персоналізації даних.
- Недостатня автоматизація: більшість процесів, як-от введення даних про тренування чи аналіз результатів, виконуються вручну.
- Відсутність інтеграції: наявні системи рідко взаємодіють із фітнес-трекерами, мобільними застосунками чи іншими сервісами для збору та обробки спортивних даних.
- Низька доступність: обмежені можливості для перегляду та аналізу особистих результатів у зручному форматі.

Таким чином, проблема полягає у створенні інтегрованої системи, яка автоматизує процеси моніторингу, обробки й аналізу спортивних досягнень людини. Вона має бути зручною у використанні, забезпечувати безпеку даних та інтеграцію з сучасними технологіями.

Далі в розділі буде визначено вимоги до веб-додатку, сформовано цілі та задачі, які необхідно вирішити для досягнення поставленої мети.

1.2 Характеристика спортивних заходів як об'єкта моніторингу

Спортивні заходи - це форма організації рухової активності, яка є комплексною. Основна мета заходу – систематичне оновлення фізичних якостей, підвищення рівня здоров'я та досягнення спортивних рекордів, а також забезпечення соціального аспекту. У контексті інформаційної системи або веб-додатку спортивні події є об'єктом моніторингу та потребують структурованого способу збору, обробки та аналізу даних.

Основними особливостями моніторингу спортивних подій, які можуть бути:

- Тип заходу (індивідуальний чи командний, тренування чи змагання);

- Тривалість та інтенсивність (у хвиликах, годинах, з урахуванням фізичного навантаження);
- Фізіологічні параметри учасників (пульс, витрачені калорії, кроки, пульс тощо);
- Розташування (на вулиці чи в приміщенні, геолокаційні координати);
- Кількість учасників та їх особисті характеристики (вік, стать, рівень підготовки);
- Мета заходу (змагання, реабілітація, вага).

1.3 Аналіз потреб користувачів на прикладі готових рішень

У сучасному світі все більше людей використовують мобільні додатки для підтримки фізичної активності, моніторингу здоров'я та досягнень у спорті. Для того, щоб розробити ефективний веб-додаток, який буде корисним для широкої аудиторії, важливо враховувати досвід існуючих рішень. У цьому підрозділі буде проаналізовано кілька популярних додатків, що активно використовуються в Україні протягом останніх 5 років, і виокремимо потреби користувачів, а також недоліки цих додатків, що можуть бути враховані при розробці нового продукту.

MyFitnessPal - це комплексна програма для фітнесу та харчування, яка слугує особистим помічником у здоров'ї [1]. Це допомагає користувачам встановлювати та відстежувати цілі щодо втрати ваги або досягнення, підраховувати калорії та вести харчовий щоденник (рис 1.1.).

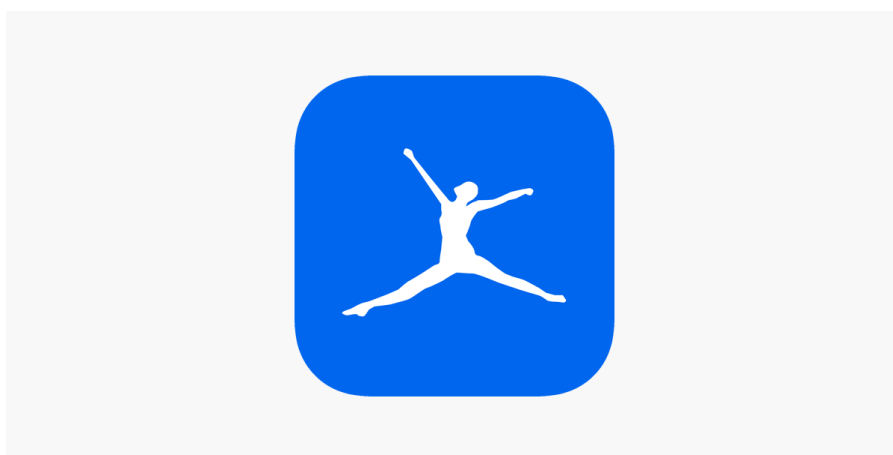


Рис. 1.1. Додаток MyFitnessPal

Цільова аудиторія: Спортсмени, люди, які бажають покращити фізичну форму, дієтисти, люди, що хочуть схуднути.

Основні функції:

- Відстеження харчування.
- Визначення кількості спожитих калорій та макронутрієнтів.
- Інтеграція з іншими фітнес-додатками та пристроями.
- Спостереження за прогресом і досягненнями.

Потреби користувачів:

- Бажання мати простий інтерфейс для моніторингу харчування та тренувань.
- Наявність функцій для зручного введення їжі та аналізу раціону.
- Інтеграція з іншими платформами (наприклад, Apple Health або Google Fit).
- Можливість встановлення індивідуальних цілей, таких як контроль за ваговими показниками або покращення спортивних результатів.

Недоліки:

- Обмеженість у налаштуванні тренувальних програм: Додаток не надає детальних планів тренувань, а також не адаптується до специфічних типів фізичних активностей.

- Залишкова складність для новачків: Інтерфейс додатку може бути складним для користувачів, які тільки починають працювати над своїм здоров'ям, через велику кількість опцій і функцій.

Strava - це мобільний застосунок і сайт для відстеження спортивної активності через GPS [2]. Девіз сервісу: «соціальна мережа для спортсменів» (рис 1.2.).

Цільова аудиторія: Бігуни, велосипедисти, спортсменки та любителі активного способу життя.

Основні функції:

- Відстеження спортивних активностей (біг, велосипед, плавання).
- Аналіз тренувань за різними метриками (час, швидкість, пульс, каденс, тощо).
- Можливість змагання з іншими користувачами та порівняння результатів.
- Інтеграція з іншими пристроями, такими як смарт-годинники та фітнес-трекери.



Рис 1.2. Додаток Strava

Потреби користувачів:

- Бажання мати детальні графіки та аналітику про свої тренування.
- Підтримка можливості змагань із друзями та спільнотами.
- Інтерфейс, що надає точні дані для покращення ефективності тренувань.
- Персоналізація функцій для різних рівнів підготовки.

Недоліки:

- Орієнтація лише на специфічні види активностей: Додаток зосереджений здебільшого на бігу, велоспорті та плаванні. Для осіб, що займаються силовими вправами або кардіонавантаженнями, функціонал не повністю відповідає потребам.

- Проблеми з конфіденційністю: Деякі користувачі повідомляють про проблеми з обміну особистими даними, адже на платформі легко порівнювати свої результати з іншими користувачами.

FatSecret [3] - це мобільний додаток, веб-сайт і API, які допомагають людям досягати цілей щодо схуднення, знаходити точну інформацію про харчування та використовувати програми для глобального здоров'я (рис 1.3.).

Цільова аудиторія: Люди, які хочуть контролювати своє харчування, схуднути або підтримувати фізичну форму.

Основні функції:

- Ведення щоденника харчування і моніторинг калорій.

- Велика база даних з продуктами харчування та їхніми калорійними значеннями.
- Відстеження фізичної активності та калорій, витрачених під час тренувань.
- Спільнота користувачів для підтримки та мотивації.



Рис. 1.3. Додаток FatSecret

Потреби користувачів:

- Легкий доступ до великої бази даних з продуктами харчування.
- Можливість створення індивідуальних дієт і контролю споживаних калорій.
- Підтримка здорового способу життя через регулярні тренування та відстеження фізичних досягнень.

Недоліки:

- Невелика увага до тренувань: Основна функція програми зосереджена на харчуванні, але вона має обмежену підтримку для тренувань, особливо для людей, які займаються силовими вправами або хочуть мати детальні тренувальні плани.

Headspace [4] - це популярний додаток для медитації, спорту, врівноваження який використовується в 190 країнах світу, має понад 70 мільйонів користувачів і понад 600 000 відгуків (рис. 1.4.).

Цільова аудиторія: Люди, які шукають методи зменшення стресу та покращення психоемоційного стану.

Основні функції:

- Медитація для релаксації, зменшення стресу та поліпшення сну.
- Програми для розвитку емоційної стійкості та психологічного здоров'я.
- Можливість інтеграції з іншими фітнес-додатками для комплексного здоров'я.

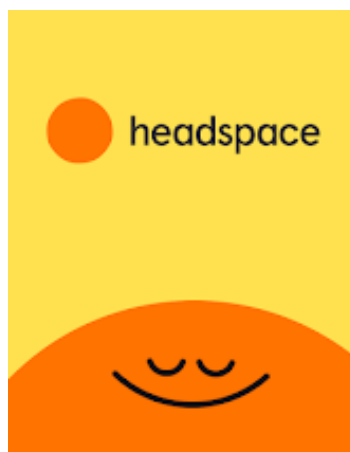


Рис 1.4. Додаток Headspace

Потреби користувачів:

- Психологічна підтримка через медитативні техніки.
- Легкість у використанні для початківців та досвідчених користувачів.
- Покращення якості сну та боротьба зі стресом.

Недоліки:

- Обмежена фізична активність: Headspace практично не фокусується на фізичній активності, а тільки на психологічній підтримці, тому не є ефективним інструментом для комплексного фізичного відновлення.

- Відсутність інтерфейсу для відстеження фізичних досягнень: Програма не надає можливості для запису тренувань або досягнень, що важливо для користувачів, які прагнуть поєднати психологічне здоров'я з фізичним прогресом.

Аналіз показав, що існуючі додатки задовольняють основні потреби користувачів у відстеженні харчування, фізичних досягнень та психологічного стану, але мають суттєві обмеження. Для нашого веб-додатку важливо:

1. Розширити функціонал для включення повної адаптації під індивідуальні потреби користувачів, включаючи реабілітаційні програми та тренування для відновлення після фізичних травм.

2. Покращити персоналізацію планів тренувань для різних категорій користувачів, від новачків до досвідчених спортсменів.

3. Враховувати психологічні аспекти здоров'я та включати методи психологічної підтримки, що допоможе користувачам знижувати стрес і покращувати загальний емоційний стан.

Цей підхід дозволить створити інструмент, який буде корисний як для звичайних користувачів, так і для осіб, які потребують реабілітації після інтенсивних навантажень або травм.

1.4 Визначення вимог до системи та концептуальна модель проекту

Розробка веб-додатка для комп'ютерного моніторингу та аналізу спортивних досягнень користувача потребує детального визначення вимог до системи, щоб забезпечити її функціональність, надійність та зручність для кінцевих користувачів. Це включає як загальні вимоги до функцій і можливостей додатка, так і специфічні потреби певних груп користувачів. Зокрема, для військових буде розроблена можливість адаптації програми для відновлення після фізичних навантажень, хоча акцент в основному робиться на звичайних користувачах.

Цей підрозділ зосереджується на визначенні функціональних та нефункціональних вимог, а також на представленні концептуальної моделі проекту, яка відображає основні компоненти системи та їх взаємодії.

1. Функціональні вимоги до системи

1.1. Реєстрація та авторизація користувачів

- Користувачі повинні мати можливість реєструватися за допомогою електронної пошти або через соціальні мережі (Google, Facebook тощо) [5];
- Авторизація повинна бути безпечною та захищеною, з можливістю відновлення паролю для користувачів;
- Для військових передбачена можливість підвищеного рівня безпеки даних.

1.2. Персоналізація профілю користувача

- Кожен користувач має можливість створити профіль, де будуть зазначені основні фізичні параметри (вага, зріст, рівень фізичної підготовки) та цілі (покращення фізичних показників, відновлення після травм тощо).

- Програма пропонуватиме персоналізовані рекомендації з тренувань, харчування та відновлення, в залежності від введених даних.

1.3. Відстеження фізичної активності

- Користувачі можуть вводити свої тренування вручну або використовувати інтеграцію з фітнес-пристроями (смарт-годинники, фітнес-браслети).

- Система повинна автоматично обчислювати витрачені калорії, час тренувань, рівень інтенсивності та інші показники фізичної активності.

- Користувачам будуть доступні графіки та статистика для аналізу прогресу.

1.4. Моніторинг харчування

- Система повинна дозволяти користувачам відстежувати харчування, вводячи спожиті продукти та порції.

- Користувачі отримуватимуть рекомендації щодо балансу білків, вуглеводів, жирів та калорійності раціону.

- Для військових будуть розроблені спеціальні харчові рекомендації, орієнтовані на потреби в енергетичному відновленні після інтенсивних фізичних навантажень.

1.5. Персоналізовані тренувальні програми

- Додаток має пропонувати тренувальні програми, адаптовані до рівня фізичної підготовки користувача.

- Для кожного користувача буде сформовано план тренувань, що включає різні типи активності (сила, кардіо, гнучкість).

- Веб-додаток також може враховувати наявність травм чи інших обмежень (наприклад, для військових користувачів після бойових завдань).

1.6. Соціальні функції та мотивація

- Користувачі можуть створювати групи для взаємодії, обміну досвідом, досягненням цілей та обговорення результатів.

- Мотиваційні системи, такі як нагороди, бали та досягнення, будуть використовуватися для підтримки користувачів у процесі досягнення цілей.

- Можливість порівнювати свої результати з іншими користувачами, а також отримувати поради від професіоналів.

1.7. Інтеграція з іншими платформами та пристроями

- Веб-додаток має бути сумісним з іншими популярними платформами для моніторингу здоров'я та фізичної активності, такими як Google Fit (рис. 1.5.) [6], Apple Health (рис. 1.6.) [7], Fitbit (рис 1.7.) [8].

- Здійснюватиметься інтеграція з фітнес-пристроями для автоматичного збору даних про активність та здоров'я користувача.



Рис 1.5. Платформа Google Fit



Рис 1.6. Платформа Apple Health



Рис 1.7. Платформа Fitbit

2. Нефункціональні вимоги до системи

2.1. Безпека даних

- Система повинна мати високий рівень захисту персональних даних користувачів, включаючи шифрування та використання надійних методів аутентифікації.

- Зберігання чутливих даних (медичні показники, історія тренувань) має відповідати стандартам безпеки та конфіденційності.

2.2. Швидкодія та надійність

- Система повинна бути швидкою і забезпечувати безперебійний доступ до даних, навіть при високих навантаженнях.

- Розробка додатку повинна включати механізми відновлення даних у разі неполадок чи втрати з'єднання.

2.3. Масштабованість

- Веб-додаток має бути здатним масштабуватися, підтримуючи велику кількість користувачів одночасно, зберігаючи високу продуктивність та швидкодію.

2.4. Мобільність та доступність

- Додаток має бути адаптований для мобільних пристроїв (смартфонів та планшетів), щоб користувачі могли зручно відслідковувати свої тренування та харчування з будь-якої точки.

- Інтерфейс повинен бути адаптивним і зручним для різних типів екранів.

2.5. Мультимовність

- Платформа повинна підтримувати кілька мов, зокрема українську, англійську та інші популярні мови, для забезпечення доступності додатку для міжнародної аудиторії.

2.6. Сумісність

- Веб-додаток повинен працювати на всіх сучасних браузерах і підтримувати різні операційні системи; Операційна система (ОС) — це програмне забезпечення, яке виконує управління апаратними ресурсами комп'ютера та надає інтерфейс для взаємодії користувача з комп'ютером. ОС є основою для роботи всіх інших програм. Історія розвитку операційних систем є багатогранною і відображає еволюцію комп'ютерних технологій. [9] (Windows, macOS, Android, iOS) (рис. 1.8.).



shutterstock.com · 2390974209

Рис. 1.8. Операційні системи

3. Концептуальна модель проекту

Концептуальна модель проекту включає основні компоненти, які взаємодіють для забезпечення функціональності додатку:

1. Користувацький інтерфейс (UI) [10] :

UI (User Interface) — це користувацький інтерфейс. Цим терміном описують те, як продукт виглядає: кольори, дизайн, анімація, контент, форми (кнопки, мітки, поля для вводу) тощо. Інтерфейс повинен бути простим і зручним, щоб користувачі могли швидко виконувати основні функції: вводити тренування, харчування, перевіряти статистику прогресу.

2. Модуль збору даних:

Збір інформації про активність користувачів здійснюється через інтеграцію з фітнес-пристроями або ручний ввід даних.

3. Модуль аналізу даних:

Аналізує фізичні досягнення користувача, порівнює з попередніми результатами та надає рекомендації для досягнення наступного рівня.

4. Модуль харчування:

Дозволяє користувачам вводити дані про спожиті продукти, а також автоматично обчислювати їх калорійність та макроелементи.

5. Модуль тренувальних програм:

Розробляє індивідуальні плани тренувань, які враховують рівень фізичної підготовки та цілі користувача.

6. Інтеграція з іншими платформами:

Система повинна бути інтегрована з іншими платформами, такими як Google Fit, Apple Health для отримання даних про активність.

7. Модуль безпеки:

Забезпечує конфіденційність особистих даних користувачів через використання найсучасніших методів захисту даних.

Визначення вимог до системи та концептуальна модель проекту забезпечують чітке уявлення про майбутній веб-додаток. Врахування потреб користувачів, у тому

числі військових, дозволить розробити ефективну та безпечну платформу для моніторингу фізичної активності, харчування та прогресу користувачів.

1.5 Постановка задач та очікувані результати

Постановка задач є важливою складовою етапу розробки веб-додатка, оскільки саме від цього етапу залежить структура і функціональність майбутнього продукту. Метою цього підрозділу є чітке визначення основних задач, які повинні бути виконані в межах проекту, а також формулювання очікуваних результатів від їх виконання. Задачі охоплюють всі етапи розробки, від проектування та дизайну до інтеграції, тестування та запуску. Кожна задача має бути забезпечена ясным результатом, який можна виміряти та перевірити.

1. Постановка задач

Для досягнення мети створення веб-додатка для моніторингу та обробки фізичних результатів учасників спортивних заходів, включаючи потреби користувачів, проект передбачає реалізацію низки важливих задач. Кожна з них має конкретні цілі та вимоги.

1.1. Дизайн і концепція користувацького інтерфейсу

Задача: Розробити інтуїтивно зрозумілий, доступний і привабливий інтерфейс для користувачів. Це включає в себе створення макетів і прототипів для мобільних та веб-версій додатка, щоб забезпечити зручну навігацію, а також привабливий зовнішній вигляд.

Очікуваний результат: Виготовлений прототип інтерфейсу, перевірений через тестування на реальних користувачах, з урахуванням їх зауважень. Має бути досягнута максимальна зручність в користуванні і відсутність складних елементів, що можуть ускладнити використання додатка.

1.2. Інтеграція з фітнес-платформами та мобільними пристроями

Задача: Реалізувати інтеграцію з популярними фітнес-платформами (Google Fit, Apple Health, Fitbit), щоб користувачі могли автоматично отримувати дані про фізичну активність.

Очікуваний результат: Можливість користувачам додавати свої дані про активність, що зберігаються на інших платформах, автоматично без необхідності ручного введення. Інтеграція повинна бути безшовною, забезпечувати коректний обмін даними та безпеку.

1.3. Створення персоналізованих тренувальних планів

Задача: Розробити систему, яка на основі особистих даних користувача (вік, фізичний стан, цілі, рівень підготовки) генеруватиме індивідуальні тренувальні програми.

Очікуваний результат: Кожен користувач отримає персоналізовані рекомендації, які будуть враховувати всі необхідні фактори для максимального ефекту від тренувань, зокрема адаптивні рекомендації щодо інтенсивності тренувань для тих, хто відновлюється після травм або боїв.

1.4. Відстежування та моніторинг результатів

Задача: Створити функціонал для моніторингу фізичних результатів користувачів, який автоматично збирає дані про виконані вправи, калорії, час тренувань, а також порівнює ці результати з попередніми досягненнями.

Очікуваний результат: Користувачі будуть мати змогу відстежувати свій прогрес у вигляді графіків та звітів, що дозволить їм ефективно коригувати свої тренування для досягнення бажаних результатів.

1.5. Моніторинг харчування та дієт

Задача: Створити функціонал для моніторингу харчування, який дасть змогу користувачам реєструвати свої прийоми їжі та відслідковувати кількість спожитих калорій, білків, жирів і вуглеводів.

Очікуваний результат: Користувачі зможуть отримувати рекомендації щодо харчування, а також бачити аналіз свого раціону, що допоможе досягнути фізичних цілей, будь то схуднення, набір м'язової маси або відновлення після тренувань.

1.6. Безпека даних

Задача: Розробити систему безпеки даних користувачів, яка включатиме шифрування, захист від несанкціонованого доступу, а також відповідність нормам конфіденційності, таким як GDPR.

Очікуваний результат: Повний захист особистих даних користувачів, включаючи медичні показники та інші чутливі дані, забезпечуючи відповідність нормативним вимогам і безпеку.

1.7. Тестування та забезпечення якості

Задача: Провести тестування всіх функцій додатка, включаючи навігацію, роботу з базою даних, функціонування інтеграцій з іншими платформами, а також загальну стабільність роботи програми на різних пристроях.

Очікуваний результат: Після тестування додаток буде стабільно працювати, а всі знайдені помилки та недоліки будуть виправлені. Завдяки тестуванню буде забезпечено правильну роботу на всіх можливих платформах (Windows, Android, iOS).

2. Очікувані результати проекту

Очікувані результати проекту відображають бажану ефективність роботи системи в реальних умовах використання:

2.1. Зручний інтерфейс: Інтерфейс, що забезпечує легкий доступ до всіх функцій додатка, зручну навігацію та оптимізований під різні розміри екранів. Веб- та мобільні версії повинні мати однакову функціональність і зручність для користувача.

2.2. Персоналізація: Створення індивідуальних тренувальних програм і дієт, які враховують фізичний стан та особисті цілі користувача. Користувач буде отримувати налаштування, що відповідають його поточному рівню фізичної підготовки та іншим параметрам.

3. Інтеграція: Система має підтримувати інтеграцію з популярними фітнес-платформами, дозволяючи користувачам зручно імпортувати свої дані про фізичну активність з інших додатків.

4. Безпека: Система має відповідати найвищим стандартам безпеки, забезпечуючи захист від несанкціонованого доступу до персональних і медичних даних користувачів.

5. Висока надійність: Програма повинна стабільно працювати на різних пристроях та в умовах високого навантаження. Всі функції мають працювати без збоїв, забезпечуючи ефективну обробку даних користувачів.

6. Аналіз і звіти: Користувачі повинні мати можливість отримувати точну і зрозумілу інформацію про свій прогрес, аналіз тренувань і харчування, що дозволить їм коригувати свої дії для досягнення більш ефективних результатів.

Розділ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА WEB-ДОДАТКА ДЛЯ КОМП'ЮТЕРНОГО МОНІТОРИНГУ ТА АНАЛІЗУ СПОРТИВНИХ ДОСЯГНЕНЬ КОРИСТУВАЧА

Розробка веб-додатку для моніторингу та аналізу спортивних досягнень користувача потребує комплексного підходу, що включає визначення вимог до системи, вибір відповідного технологічного стеку, проєктування архітектури та бази даних, а також реалізацію функціональних можливостей.

У цьому розділі розглянуто ключові етапи створення веб-додатку: обґрунтовано вибір технологій, описано архітектуру системи та механізми її роботи, сформовано модель бази даних, що дозволяє ефективно зберігати та обробляти спортивні результати користувачів. Також детально розглянуто основні функціональні можливості веб-додатку, зокрема процес реєстрації та автентифікації користувачів, ведення тренувань, аналіз спортивних показників та візуалізацію зібраних даних.

Розроблена система покликана автоматизувати процеси обліку та аналізу спортивних досягнень, забезпечити зручний доступ до даних та інтеграцію з сучасними технологіями, що підвищує ефективність користування та допомагає користувачам контролювати власний прогрес у спорті.

2.1. Вибір технологічного стеку

Вибір відповідного стека програмування – це важливий крок під час розроблення програмного забезпечення [11]. Правильний підхід до вибору мов програмування, фреймворків, бази даних та інших технологій безпосередньо впливає на продуктивність, гнучкість та масштабованість системи.

Для реалізації даного проєкту було вирішено використовувати **Node.js** для серверної частини, **Express.js** як фреймворк для управління маршрутизацією та обробки HTTP-запитів, **MySQL** для збереження та обробки даних, а також **HTML**, **CSS** і **JavaScript** для клієнтської частини.

Обґрунтування вибору серверної частини: Node.js та Express.js

Node.js, асинхронне подієве середовище виконання JavaScript, створене будувати масштабовані мережеві застосунки [12]. Є однією з найпопулярніших технологій для створення веб-додатків завдяки своїй асинхронній природі та високій продуктивності. Він побудований на рушії **V8**, який забезпечує швидке виконання JavaScript-коду, що особливо важливо для роботи з великою кількістю одночасних запитів.

Основні причини вибору Node.js:

1. **Асинхронність та неблокуючий ввід-вивід.** Це дає можливість обробляти велику кількість одночасних запитів, що критично для веб-додатків, які працюють у режимі реального часу.
2. **Гнучкість.** Node.js дозволяє легко масштабувати систему у майбутньому, додаючи нові функціональні можливості без значних змін у коді.
3. **Велика екосистема пакетів.** Завдяки **npm (Node Package Manager)** можна швидко інтегрувати різні бібліотеки, що прискорює розробку.

Щоб спростити роботу із серверною частиною, було обрано **Express.js** [13] - один із найпопулярніших фреймворків для Node.js. Його використання дозволяє легко керувати маршрутами, обробляти HTTP-запити та відповіді, а також інтегрувати middleware для додаткових функцій.

Переваги Express.js:

- Легка та мінімалістична структура, що дає можливість швидко налаштувати сервер.
- Гнучка система маршрутизації, яка допомагає організувати логіку веб-додатку.
- Простота у роботі з API-запитами, що важливо для взаємодії з клієнтською частиною.

Обґрунтування вибору бази даних: MySQL

Для збереження, обробки та управління даними було обрано **MySQL** [14] - реляційну систему керування базами даних (СКБД), яка є однією з найбільш надійних та широко використовуваних технологій у веб-розробці.

Причини вибору MySQL:

1. **Структурованість даних.** Оскільки у додатку передбачено роботу з обліковими записами користувачів, тренуваннями та аналітикою, реляційна модель є найкращим варіантом для організації взаємопов'язаних даних.

2. **Продуктивність.** MySQL забезпечує швидку обробку запитів та оптимізовану роботу з великими наборами даних.

3. **Захист та надійність.** Доступні інструменти для резервного копіювання та управління доступом забезпечують безпеку збережених даних.

4. **Гнучка система запитів.** Використання SQL-запитів дозволяє виконувати складні вибірки, фільтрування та аналіз інформації.

Для взаємодії з MySQL у Node.js буде використовуватися **бібліотека MySQL2**, яка підтримує асинхронні запити та підключення до бази даних через **Promise API**, що підвищує продуктивність.

Обґрунтування вибору клієнтської частини: HTML, CSS та JavaScript

Оскільки веб-додаток має бути зручним та інтуїтивно зрозумілим для користувачів, для створення інтерфейсу було обрано стандартний набір технологій:

- **HTML (HyperText Markup Language)** використовується для структури веб-сторінок, що визначає розташування основних елементів додатку.
- **CSS (Cascading Style Sheets)** застосовується для оформлення зовнішнього вигляду та забезпечення адаптивного дизайну, що дозволить користувачам комфортно працювати на різних пристроях.
- **JavaScript (Vanilla JS)** використовується для створення інтерактивності, обробки подій та взаємодії з сервером через **AJAX або Fetch API**.

Такий підхід дозволяє забезпечити швидке завантаження сторінок, плавний досвід роботи користувачів та простоту в подальшому розширенні функціоналу. Вибір **Node.js + Express.js** як серверного середовища дозволяє створити продуктивний, масштабований та зручний у підтримці веб-додаток. **MySQL** гарантує стабільне збереження даних та швидку обробку запитів, а використання **HTML, CSS та JavaScript** забезпечує комфортний користувацький досвід. У сукупності цей технологічний стек відповідає вимогам проєкту, дозволяючи ефективно реалізувати моніторинг та аналіз спортивних досягнень користувачів.

2.2. Архітектура системи

Проектування архітектури веб-додатку є одним із ключових етапів розробки, оскільки саме від цього залежить стабільність, продуктивність і масштабованість системи. У нашому випадку веб-додаток має забезпечувати зручний облік та аналіз спортивних досягнень користувачів, обробляти велику кількість даних та підтримувати багатокористувацький доступ. Виходячи з цих вимог, було вирішено використовувати **клієнт-серверну архітектуру**.

Клієнт-серверну архітектуру можна означити, як концепцію інформаційної мережі в якій основна частина її ресурсів зосереджена в серверах, обслуговуючих своїх клієнтів. Така архітектура визначає такі типи компонентів [15]:

- набір **серверів**, які надають інформацію або інші послуги програмам, які звертаються до них;
- набір **клієнтів**, які використовують сервіси, що надаються серверами;
- **мережа**, яка забезпечує взаємодію між клієнтами та серверами.

Загальна схема взаємодії компонентів. Вся система складається з трьох основних частин:

- **Клієнтська частина (Frontend)** [16] – це клієнтська (видима) сторона інтерфейсу. Тобто все, що браузер виводить на екран, і з чим на сторінці сайту взаємодіє користувач: кнопки, текст, зображення, таблиці, стрілки, банери та інші візуальні елементи інтернет-ресурсу
- **Серверна частина (Backend)** [17] – це серверна частина сайту/додатку, прихована від очей користувача. Вона відповідає за логіку обробки запитів, збереження та отримання даних, управління обліковими записами, авторизацію, взаємодію з зовнішніми API тощо.
- **База даних (MySQL)** – зберігає всі необхідні дані, включаючи інформацію про користувачів, їхні тренування, досягнення та результати аналізу фізичних показників.

Процес взаємодії між цими компонентами виглядає так:

1. Користувач заходить на веб-додаток через браузер.
2. Вводить свої дані для авторизації, після чого система перевіряє правильність введених даних.

3. Після успішного входу він може додавати або переглядати свої спортивні досягнення.
4. Всі дії користувача (наприклад, додавання нового тренування) формують **HTTP-запити**, які надсилаються на сервер.
5. Сервер обробляє ці запити, звертається до бази даних, виконує необхідні операції та повертає відповідь у вигляді **JSON-даних**.
6. Клієнтська частина отримує ці дані та відображає їх у зручному для користувача форматі.

REST API: структура запитів та відповідей. Для забезпечення ефективного зв'язку між клієнтом і сервером було реалізовано **REST API**, яке дозволяє здійснювати обмін даними у стандартизованому форматі. Сучасна розробка програмного забезпечення (ПЗ) безперервно пов'язана з таким поняттям як інтерфейс прикладного програмування. Річ у тім, що так званий API є тією ключовою ланкою, яка дає змогу різним системам ефективно взаємодіяти між собою [18].

Основні маршрути API:

- **POST /register** – реєстрація нового користувача.
- **POST /login** – автентифікація користувача.
- **GET /workouts** – отримання списку тренувань користувача.
- **POST /workouts** – додавання нового тренування.
- **GET /analytics** – отримання статистичних даних та аналітики.

Кожен запит має свою **структуру**. Наприклад, при додаванні нового тренування користувач відправляє запит (рис.2.1):

```
{  
  "userId": 12,  
  "exercise": "Біг",  
  "duration": 30,  
  "caloriesBurned": 250  
}
```

Рисунок 2.1. Фрагмент запиту

Сервер обробляє цей запит, записує дані у базу і надсилає відповідь (рис.2.2)
(рис. 2.3):

```
{  
  "message": "Тренування успішно додано!",  
  "workoutId": 45  
}
```

Рисунок 2.2. Фрагмент відповіді

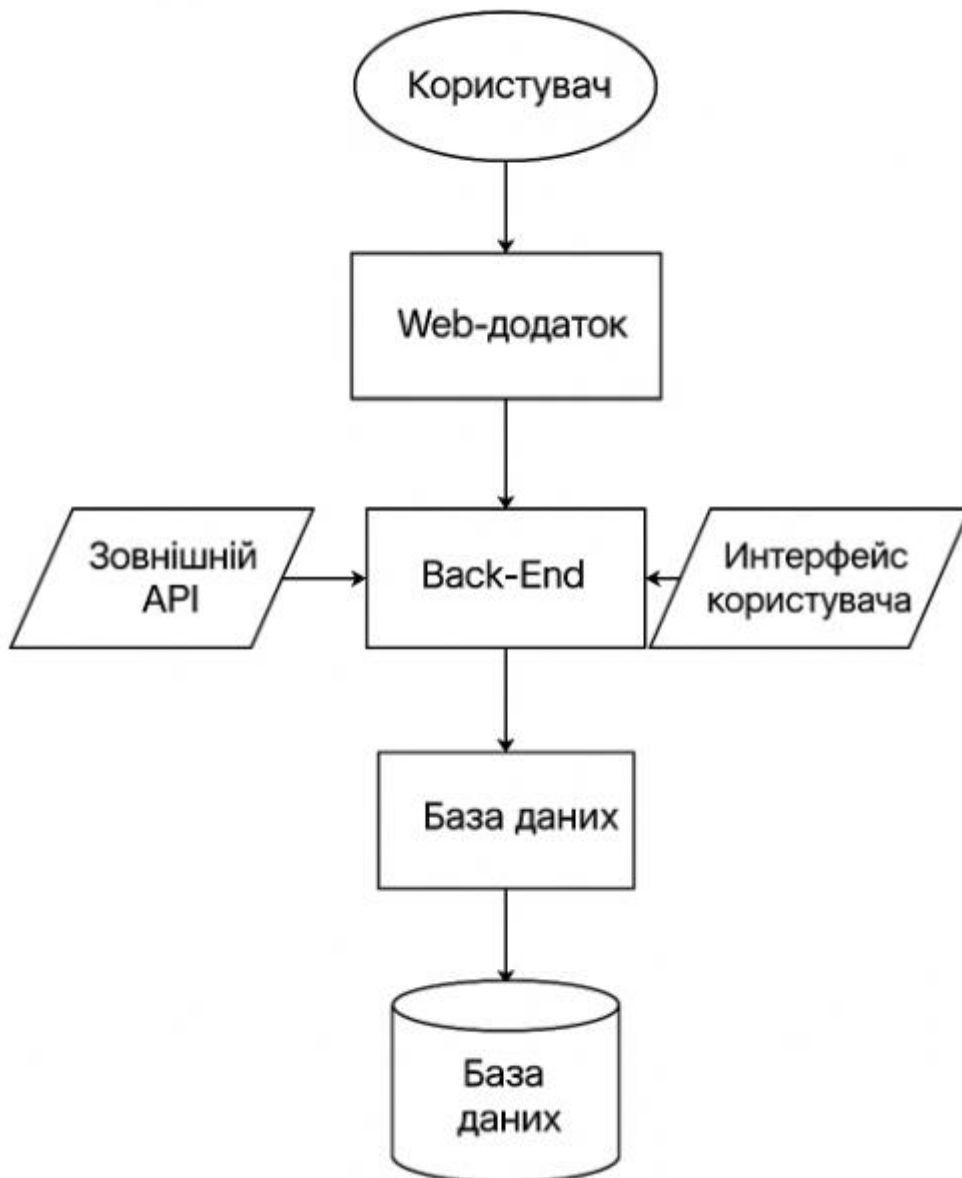


Рис.2.3 Схема системної архітектури

Це дозволяє додатку працювати динамічно і в реальному часі обробляти інформацію без потреби оновлення сторінки.

Організація серверної частини. Для кращої підтримуваності коду серверна частина структурована за модульним принципом. У кодї додатку можна виокремити кілька основних модулів:

- **routes/** – містить маршрути для роботи з API-запитами.
- **controllers/** – містить логіку обробки запитів (наприклад, додавання тренування або створення статистики).
- **models/** – описує структуру збереження даних та взаємодію з базою даних.
- **config/** – містить налаштування підключення до бази даних та інші конфігурації.

Така структура дозволяє легко розширювати додаток та підтримувати його в майбутньому.

Взаємодія між сервером та базою даних. Сервер взаємодіє з MySQL через бібліотеку **MySQL2**, яка дозволяє виконувати асинхронні SQL-запити. Приклад запиту на отримання всіх тренувань користувача виглядає так (рис.2.4), (рис 2.5):

```
const db = require('../config/db');

const getWorkouts = async (userId) => {
  const [rows] = await db.execute(
    "SELECT * FROM workouts WHERE user_id = ?",
    [userId]
  );
  return rows;
};
```

Рисунок 2.4. Фрагмент запиту на отримання всіх тренувань

Забезпечення масштабованості. Архітектура додатку передбачає можливість масштабування у майбутньому. Якщо кількість користувачів збільшиться, можна розширити серверну інфраструктуру, додавши балансувальник навантаження, або використати кешування даних (наприклад, Redis [19]) для прискорення обробки запитів.



Рисунок 2.5. Діаграма архітектури веб-додатка

У даному підрозділі розглянуто структуру додатку, основні компоненти та їхню взаємодію. Описано підхід до побудови клієнт-серверної архітектури, розподілення логіки між фронтендом та бекендом, взаємодію з базою даних. Також описано використання REST API або GraphQL для комунікації між компонентами.

2.3. Проєктування бази даних

Проєктування бази даних є одним із ключових етапів створення веб-додатку, оскільки саме тут зберігатиметься вся необхідна інформація про користувачів, їхні тренування, досягнення та аналітичні дані. Для зберігання даних було обрано **реляційну базу даних MySQL**, яка забезпечує структуроване збереження інформації, швидкі SQL-запити та можливість виконання складних аналітичних операцій.

У даному обґрунтовано вибір системи керування базами даних (SQL або NoSQL), її переваги та обмеження.

2.3.1 Основні сутності та їхні атрибути

База даних містить кілька основних сутностей, які відповідають ключовим елементам веб-додатку:

1. **Користувачі (users)** – зберігає інформацію про користувачів системи.
2. **Тренування (workouts)** – фіксує кожне тренування користувача.
3. **Досягнення (achievements)** – відображає персональні досягнення користувачів.
4. **Аналітика (analytics)** – містить статистичні дані та розрахунки на основі тренувань.
5. **Категорії вправ (exercise_categories)** – допомагає класифікувати типи тренувань.

Таблиця users (Користувачі). Ця таблиця містить інформацію про кожного користувача веб-додатку. Вона є центральною, оскільки інші таблиці будуть пов'язані з нею через зовнішні ключі.

Основні атрибути (рис 2.6):

- id – унікальний ідентифікатор користувача.
- name – ім'я користувача.
- email – електронна пошта, яка використовується для входу.
- password – зашифрований пароль.
- created_at – дата реєстрації користувача.

```

CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    password VARCHAR(255) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

```

Рис 2.6. Основні атрибути. Користувачі

Таблиця workouts (Тренування). Фіксує інформацію про всі тренування користувачів, включаючи тип вправ, тривалість та витрачені калорії.

Основні атрибути (рис 2.7):

- id – унікальний ідентифікатор тренування.
- user_id – посилання на користувача, який виконав тренування.
- exercise_name – назва вправи.
- duration – тривалість тренування (у хвилинах).
- calories_burned – кількість спалених калорій.
- date – дата та час виконання тренування.

```

CREATE TABLE workouts (
    id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    exercise_name VARCHAR(100) NOT NULL,
    duration INT NOT NULL,
    calories_burned INT NOT NULL,
    date DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
);

```

Рис 2.7. Головні атрибути. Тренування

Таблиця achievements (Досягнення). Відображає особисті спортивні досягнення користувачів на основі їхньої активності.

Основні атрибути (рис 2.8):

- id – унікальний ідентифікатор досягнення.
- user_id – посилання на користувача.
- achievement_name – назва досягнення.
- description – опис досягнення.
- achieved_at – дата отримання.

```
CREATE TABLE achievements (
    id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    achievement_name VARCHAR(150) NOT NULL,
    description TEXT NOT NULL,
    achieved_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
);
```

Рис 2.8. Головні атрибути. Досягнення

Таблиця analytics (Аналітика). Зберігає зведену статистику тренувань користувача, допомагаючи аналізувати ефективність занять.

Основні атрибути (рис 2.9):

- id – унікальний ідентифікатор запису.
- user_id – посилання на користувача.
- total_workouts – загальна кількість тренувань.
- total_calories_burned – загальна кількість спалених калорій.
- avg_duration – середня тривалість тренувань.
- last_updated – дата останнього оновлення статистики.

```

CREATE TABLE analytics (
  id INT AUTO_INCREMENT PRIMARY KEY,
  user_id INT NOT NULL,
  total_workouts INT DEFAULT 0,
  total_calories_burned INT DEFAULT 0,
  avg_duration FLOAT DEFAULT 0.0,
  last_updated TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
);

```

Рис 2.9. Головні атрибути. Аналітика

Таблиця `exercise_categories` (Категорії вправ). Ця таблиця допомагає структурувати вправи за певними категоріями (кардіо, силові, йога тощо).

Основні атрибути (рис 2.10):

- `id` – унікальний ідентифікатор категорії.
- `category_name` – назва категорії.

```

CREATE TABLE exercise_categories (
  id INT AUTO_INCREMENT PRIMARY KEY,
  category_name VARCHAR(100) UNIQUE NOT NULL
);

```

Рис 2.10. Головні атрибути. Категорії вправ

У підрозділі описано основні **таблиці бази даних**, їхні ключові атрибути та типи даних. Визначено структуру для збереження інформації про **користувачів**, **тренування**, **досягнення**, **аналітику** тощо.

2.3.2 Зв'язки між таблицями (ER-діаграма)

Дана **ER-діаграма (Entity-Relationship Diagram)** (рис. 2.11) відображає взаємозв'язки між основними сутностями веб-додатку для моніторингу та аналізу спортивних досягнень користувачів.

У базі даних передбачено п'ять ключових таблиць, які зберігають інформацію про користувачів, їхні тренування, аналітику, досягнення та категорії вправ.

Основні сутності та їхні атрибути

1.1. Users (Користувачі)

- Центральна таблиця, яка містить дані про зареєстрованих користувачів.
- Має **унікальний ідентифікатор id** для кожного користувача.
- Містить `name` (ім'я), `email` (унікальна електронна пошта) та `password` (захищений пароль).
- Поле `created_at` фіксує дату реєстрації користувача.

Зв'язки:

- **1:M (один до багатьох)** → **Workouts** (один користувач може мати багато тренувань).
- **1:M** → **Achievements** (один користувач може мати багато досягнень).
- **1:1** → **Analytics** (один користувач має лише один запис з аналітикою).

1.2. Workouts (Тренування)

- Таблиця зберігає всі тренування користувачів.
- Поле `user_id` є зовнішнім ключем, який посилається на `id` у таблиці `Users`.
- Кожен запис містить `exercise_name` (назву вправи), `duration` (тривалість у хвиликах), `calories_burned` (спалені калорії) та `date` (дату тренування).

Зв'язки:

- **M:1 (багато до одного)** → **Users** (кожне тренування належить конкретному користувачеві).
- **M:1** → **ExerciseCategories** (кожне тренування може бути віднесене до певної категорії вправ).

1.3. Achievements (Досягнення)

- Зберігає персональні спортивні досягнення користувачів.
- Поле `achievement_name` містить назву досягнення, а `description` – його опис.
- Поле `achieved_at` фіксує дату отримання досягнення.

Зв'язки:

- **M:1** → **Users** (кожне досягнення належить конкретному користувачеві).

1.4. Analytics (Аналітика)

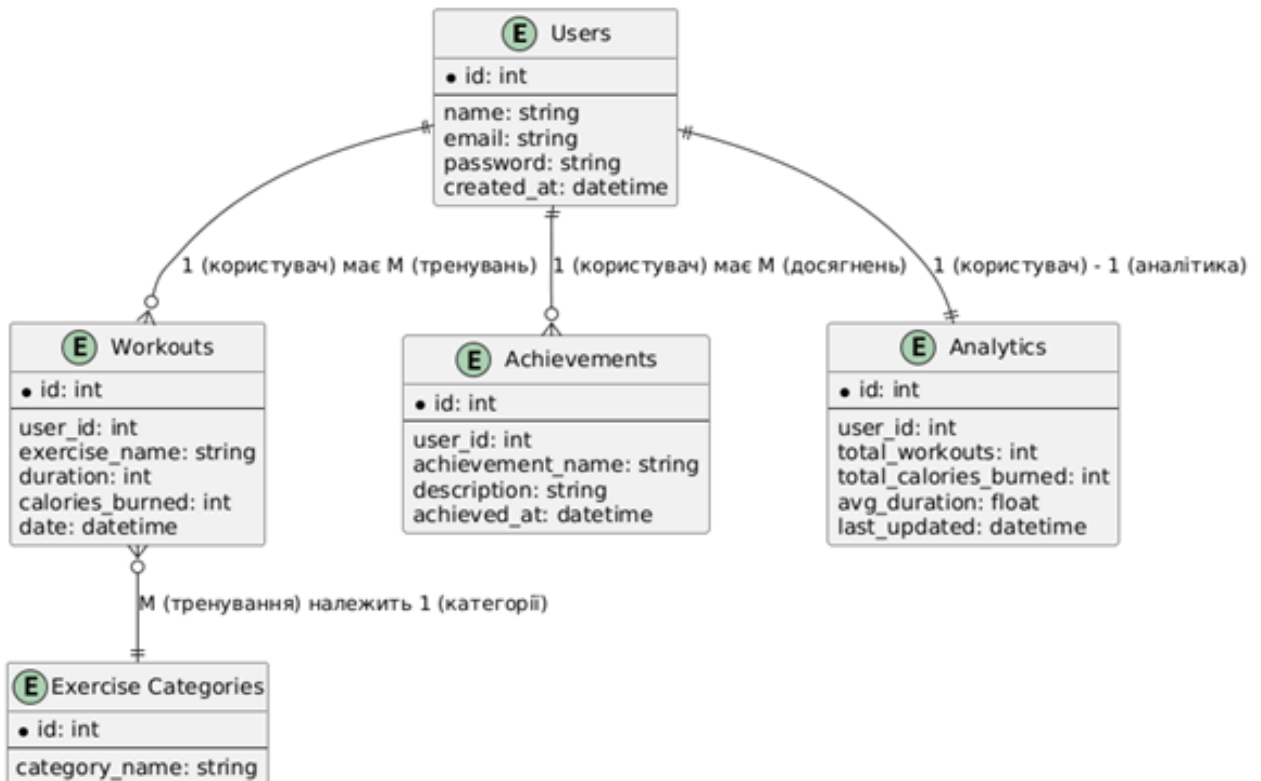


Рисунок 2.11. ER-діаграма відображає взаємозв'язки між основними сутностями веб-додатку для моніторингу та аналізу спортивних досягнень користувачів

- Містить аналітичні дані користувача, такі як `total_workouts` (загальна кількість тренувань), `total_calories_burned` (загальна кількість спалених калорій) та `avg_duration` (середня тривалість тренувань).
- Поле `last_updated` зберігає дату останнього оновлення аналітики.

Зв'язки:

- **1:1** → **Users** (кожен користувач має один запис у таблиці аналітики).

1.5. ExerciseCategories (Категорії вправ)

- Використовується для класифікації вправ (наприклад, "кардіо", "силові вправи", "йога").
- Поле `category_name` містить назву категорії.

Зв'язки:

- **1:M** → **Workouts** (кожне тренування належить певній категорії).

Подано ER-діаграму бази даних із детальним описом зв'язків між сутностями. Розглянуто типи зв'язків (один-до-багатьох, багато-до-багатьох) та механізми забезпечення цілісності даних.



Рис 2.12. Відображення сутності діаграми

2.4. Функціональні можливості веб-додатку

Функціональність веб-додатку для моніторингу та аналізу спортивних досягнень користувачів спрямована на спрощення ведення обліку тренувань, аналізу результатів та побудови персоналізованих рекомендацій. В основі системи лежать чотири ключові можливості: **реєстрація та автентифікація користувачів, управління тренуваннями, автоматичний аналіз спортивних показників і візуалізація аналітики у вигляді звітів** (рис 2.13).

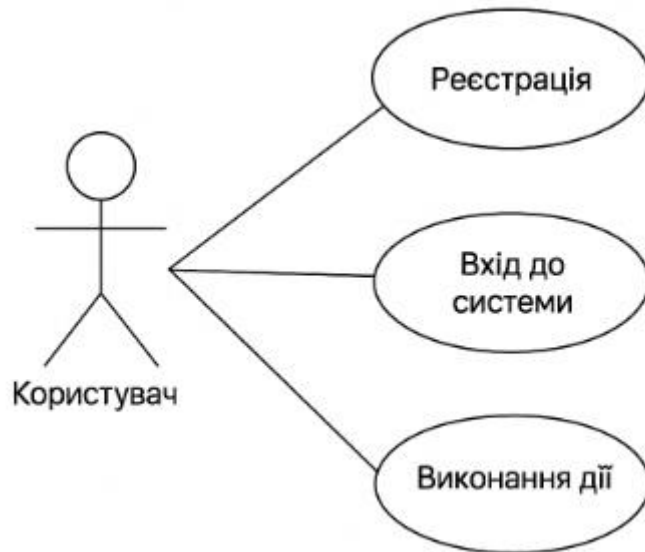


Рис 2.13. Діаграма випадків використання для користувачів системи

Описано набір основних функцій, які реалізовані в додатку, та їхній вплив на користувацький досвід.

2.4.1 Реєстрація та автентифікація користувачів

Для роботи з додатком користувач повинен пройти процес **реєстрації та автентифікації**. Це необхідно для забезпечення персонального доступу до даних та захисту інформації.

Процес реєстрації:

1. Користувач вводить ім'я, email та пароль.
2. Додаток перевіряє, чи унікальний email.
3. Пароль хешується для безпечного зберігання.
4. Дані зберігаються в базі даних, після чого користувач отримує доступ до особистого кабінету.

Процес входу (автентифікація):

1. Користувач вводить email та пароль.
2. Додаток перевіряє відповідність паролю, використовуючи хешування.

SQL-запити для реєстрації:

```
INSERT INTO users (name, email, password)
VALUES ('Іван Іваненко', 'ivan@email.com', 'захешований_пароль');
```

Node.js (Express) код для обробки реєстрації (рис 2.14):

```
const bcrypt = require('bcrypt');
const db = require('../config/db');

const registerUser = async (req, res) => {
  const { name, email, password } = req.body;
  const hashedPassword = await bcrypt.hash(password, 10);

  try {
    await db.execute("INSERT INTO users (name, email, password) VALUES (?, ?, ?)",
      [name, email, hashedPassword]);
    res.json({ message: "Користувач успішно зареєстрований!" });
  } catch (error) {
    res.status(500).json({ error: "Помилка при реєстрації" });
  }
};
```

Рис 2.14. Node.js (Express) код для обробки реєстрації

Розглянуто процес створення облікового запису, входу в систему, відновлення пароля. Описано впроваджені механізми безпеки (JWT, OAuth, двофакторна аутентифікація).

2.4.2 Додавання, редагування та перегляд тренувань

Користувач має можливість додавати, редагувати та переглядати свої тренування. Це ключова функція, яка дозволяє зберігати дані про фізичні навантаження та їхню ефективність.

Додавання тренування:

- Користувач вводить назву вправи, її тривалість та інші параметри.
- Дані передаються на сервер та записуються у базу.

Редагування тренування:

- Користувач може змінити дані про тренування (наприклад, уточнити витрачені калорії).

Перегляд тренувань:

- У розділі "Мої тренування" користувач може бачити список своїх записів, відсортованих за датою.

SQL-запит для додавання тренування (рис 2.15):

```
INSERT INTO workouts (user_id, exercise_name, duration, calories_burned, date)
VALUES (1, 'Bir', 30, 250, NOW());
```

Рис 2.15. SQL-запит для додавання тренування

Node.js (Express) для додавання тренування (рис 2.16):

```
const addWorkout = async (req, res) => {
  const { user_id, exercise_name, duration, calories_burned } = req.body;

  try {
    await db.execute("INSERT INTO workouts (user_id, exercise_name, duration, calories_burned, date) VALUES (?, ?, ?, ?, NOW())",
      [user_id, exercise_name, duration, calories_burned]);
    res.json({ message: "Тренування додано успішно!" });
  } catch (error) {
    res.status(500).json({ error: "Помилка при додаванні тренування" });
  }
};
```

Рис 2.16. Node.js (Express) для додавання тренування

Описано функціонал, що дозволяє користувачам **вносити дані про тренування**, переглядати історію активності, додавати персональні параметри (наприклад, масу тіла, кількість повторень).

2.4.3 Автоматичний аналіз спортивних показників

Щоб допомогти користувачам ефективніше досягати своїх цілей, система автоматично аналізує їхні результати та пропонує відповідну аналітику.

Основні аналітичні параметри:

- Загальна кількість тренувань – скільки разів користувач тренувався за певний період.
- Середня тривалість тренування – середній час виконання вправ.
- Загальна кількість спалених калорій – допомагає оцінити прогрес у спалюванні калорій.

Як працює аналіз:

1. Користувач додає дані про тренування.
2. Сервер виконує SQL-запит для підрахунку загальних показників.
3. Дані зберігаються у таблиці analytics та оновлюються після кожного нового тренування.

SQL-запит для підрахунку аналітики:

```
UPDATE analytics
SET total_workouts = (SELECT COUNT(*) FROM workouts WHERE user_id = 1),
    total_calories_burned = (SELECT SUM(calories_burned) FROM workouts WHERE user_id = 1),
    avg_duration = (SELECT AVG(duration) FROM workouts WHERE user_id = 1),
    last_updated = NOW()
WHERE user_id = 1;
```

Node.js (Express) код для оновлення аналітики (рис 2.17):

```
const updateAnalytics = async (user_id) => {
  await db.execute(`
    UPDATE analytics
    SET total_workouts = (SELECT COUNT(*) FROM workouts WHERE user_id = ?),
        total_calories_burned = (SELECT SUM(calories_burned) FROM workouts WHERE user_id = ?),
        avg_duration = (SELECT AVG(duration) FROM workouts WHERE user_id = ?),
        last_updated = NOW()
    WHERE user_id = ?`,
    [user_id, user_id, user_id, user_id]);
};
```

Рис 2.17. Node.js (Express) код для оновлення аналітики

Розглянуто **методи обробки даних**, які використовуються для аналізу фізичних показників користувачів. Описано алгоритми підрахунку прогресу, визначення ефективності тренувань та рекомендації на основі зібраних даних.

2.4.4 Генерація звітів та візуалізація даних

Щоб користувачі могли краще аналізувати свої досягнення, система генерує графіки та звіти.

Що включає звітність:

- Графіки змін тренувань – як змінюється кількість тренувань у часі.

- Динаміка спалювання калорій – загальна статистика за період.
- Порівняльний аналіз – наприклад, як користувач прогресує у бігових або силових вправах.

Звіти можна переглядати в особистому кабінеті. Вони будуть формуватися на основі запитів до бази даних, які збирають статистичні показники.

Node.js код для отримання статистики тренувань (рис 2.18):

```
const getStatistics = async (req, res) => {
  const { user_id } = req.params;

  const [rows] = await db.execute(
    "SELECT date, SUM(calories_burned) as total_calories FROM workouts WHERE user_id = ? GROUP BY date",
    [user_id]
  );

  res.json(rows);
};
```

Рис 2.18. Node.js код для отримання статистики тренувань

Функціональні можливості додатку охоплюють авторизацію, управління тренуваннями, аналіз спортивних показників та візуалізацію даних. Вони забезпечують зручний інструмент для відстеження прогресу та підвищення ефективності тренувань.

Таким чином, було розроблено гнучку та масштабовану систему, яка дозволяє ефективно вести облік тренувань, аналізувати спортивні досягнення та покращувати процес контролю фізичної активності. Реалізована структура дозволяє легко розширювати функціонал у майбутньому, інтегрувати нові можливості та адаптувати систему під потреби користувачів.

РОЗДІЛ 3. ВПРОВАДЖЕННЯ ТА РЕАЛІЗАЦІЯ

У цьому розділі розглядається безпосередній процес впровадження та реалізації запропонованого web-додатку, призначеного для моніторингу та обробки спортивних досягнень учасників. Відповідно до визначених у попередніх розділах функціональних вимог і технічних характеристик, у цьому етапі дослідження особлива увага приділяється практичній реалізації проекту, що охоплює архітектуру системи, вибір технологій, розробку інтерфейсу користувача, реалізацію основних модулів та перевірку працездатності.

Метою даного етапу є створення ефективного, зручного у користуванні та функціонального рішення, яке відповідатиме вимогам користувачів і забезпечуватиме якісну обробку й аналіз результатів спортивної активності. Реалізація відбувається з дотриманням принципів модульності, масштабованості та адаптивності, що дає змогу у подальшому розширювати систему або інтегрувати її з іншими сервісами.

3.1 Загальний опис реалізації

Розробка веб-додатку, що орієнтований на моніторинг фізичної активності та аналіз спортивних досягнень користувача, вимагала поетапного підходу до впровадження. Основним завданням на цьому етапі було забезпечити стабільну, інтуїтивно зрозумілу та функціонально завершену систему, яка працюватиме у середовищі реального часу. Особливу увагу приділено тому, щоб усі реалізовані компоненти працювали як єдине ціле, утворюючи завершений цифровий продукт.

У технічному контексті реалізація системи базувалась на модульному принципі — усі елементи, починаючи від авторизації користувача до виводу графічної аналітики, були спроектовані як незалежні, проте взаємопов'язані блоки. Це дозволило досягти високої гнучкості у процесі інтеграції та тестування. Упровадження реалізовувалося в середовищі локального серверного хостингу, що дозволило відтестувати всі вузли перед розгортанням на продуктивному сервері.

Кожен функціональний блок додатку — від обробки HTTP-запитів до оновлення графічних інтерфейсів — проходив тестування за сценаріями, які максимально імітували дії реальних користувачів. Такий підхід дозволив не лише виявити критичні точки, але й оптимізувати взаємодію між модулями: наприклад, спрощено обробку помилок при передачі некоректних даних, а також прискорено час відповіді сервера при великій кількості одночасних запитів.

Загальна архітектура впровадження передбачала мінімальне навантаження на клієнтську сторону. Основна обробка даних реалізована на стороні серверу за допомогою Node.js із застосуванням Express.js. Запити до бази даних, організованої на основі MySQL, реалізовувались через ORM-рівень, що дозволяє уникнути безпосередньої роботи із SQL-операціями всередині бізнес-логіки додатку. Це не лише підвищує безпеку системи, але й сприяє зменшенню кількості помилок при обробці запитів.

Користувацький інтерфейс був інтегрований поступово, з урахуванням кросплатформенності та адаптивного дизайну. При проєктуванні UI було застосовано принципи компонентного підходу: кожен інтерфейсний блок, наприклад, форма авторизації, блок обліку тренувань, сторінка аналітики чи модуль калькулятора ІМТ, реалізований у вигляді самостійного компонента. Це дозволяє гнучко змінювати логіку та зовнішній вигляд без необхідності втручання у структуру усього застосунку.

На особливу увагу заслуговує механізм динамічного оновлення інтерфейсу, який дозволяє користувачу миттєво бачити результат введених даних. Наприклад, після додавання нового тренування або обчислення ІМТ, відповідні блоки оновлюються в режимі реального часу без перезавантаження сторінки. Це стало можливим завдяки використанню асинхронної взаємодії між фронтендом та сервером, що реалізована через Fetch API.

Кінцева реалізація додатку забезпечила повноцінне функціонування системи на всіх основних етапах користувацької взаємодії: реєстрація, вхід, внесення даних, обчислення аналітики, відображення звітів та збереження обраного контенту. Система протестована на відповідність функціональним вимогам та готова до розгортання у середовищі з високим навантаженням (рис 3.1).

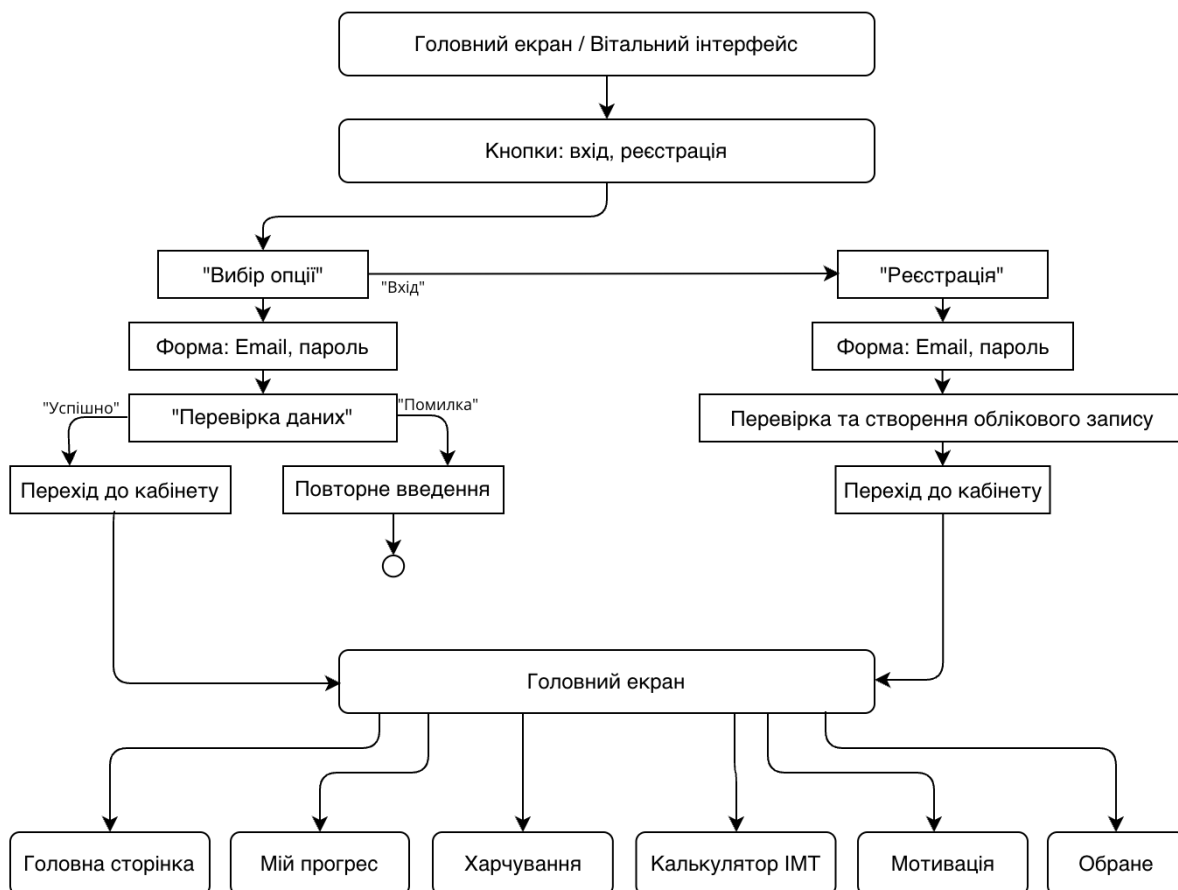


Рис 3.1. Діаграма користувацького потоку в клієнтській частині веб-додатку

3.2 Реалізація клієнтської частини

Клієнтська частина веб-додатку реалізована з дотриманням сучасних стандартів у сфері веб-розробки та мобільного UX-дизайну. Основна мета даного модуля — забезпечити користувача зручним, інтуїтивно зрозумілим інтерфейсом для щоденного використання. Реалізація включає декілька тематичних розділів, кожен із яких виконує окрему функцію, що відповідає загальній меті системи — моніторингу фізичної активності, харчування та мотивації користувача.

Кожен із функціональних блоків представлений у вигляді окремого екрану або вкладки, реалізованої з використанням стандартного HTML/CSS-фреймворку, а також JavaScript-логіки для обробки подій, запитів до серверної частини та динамічного оновлення контенту.

1. Форма реєстрації/входу

При першому вході користувач має можливість зареєструватися, вказавши свої особисті дані: ім'я, електронну пошту. У разі наявного облікового запису — здійснюється авторизація (рис 3.2). Інтерфейс забезпечує перевірку правильності введених даних, а також виводить повідомлення про помилки (наприклад, “невірний email” або “пароль надто короткий”).

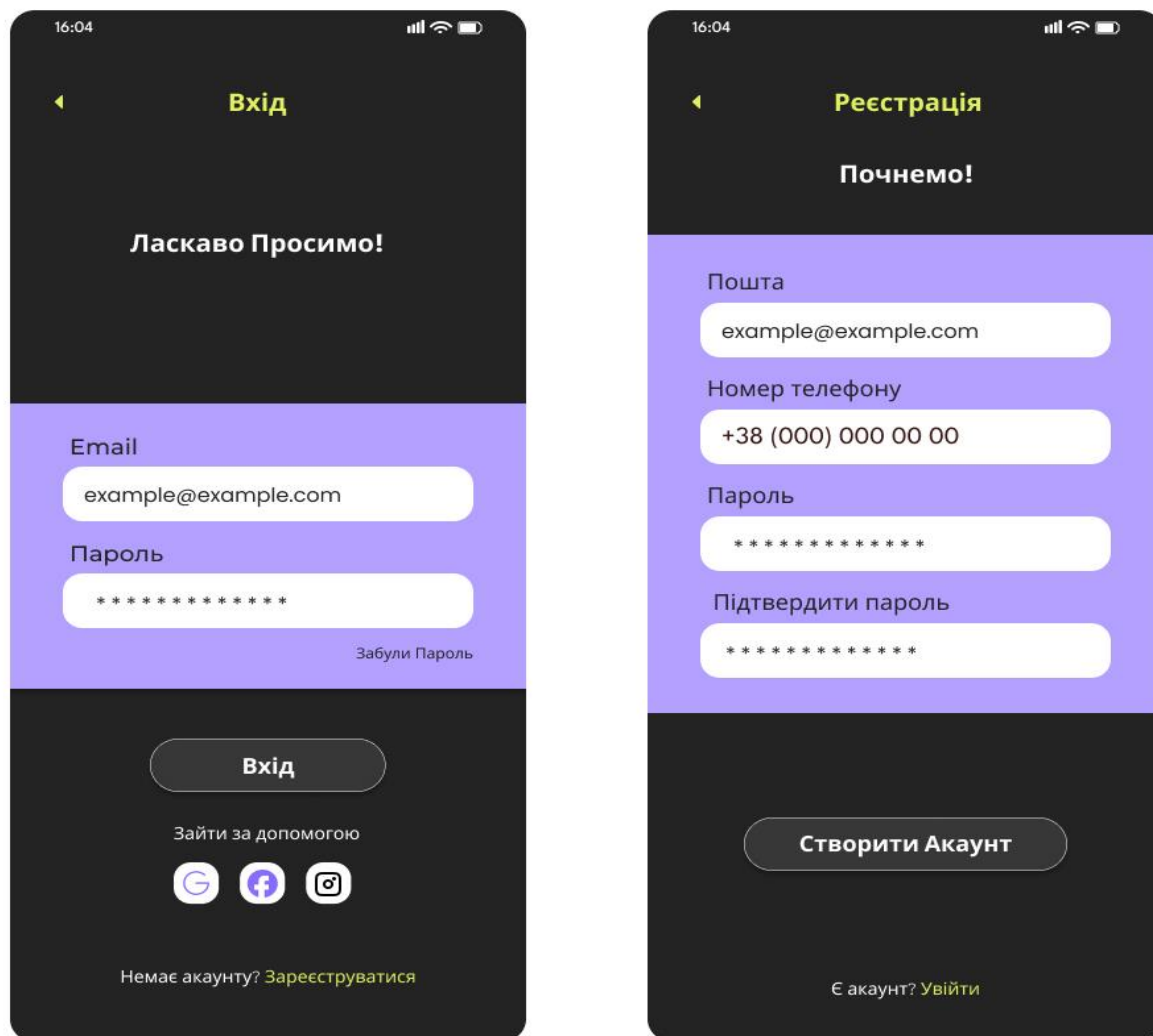


Рис 3.2. Форма реєстрації/входу (мобільна версія)

2. Головна сторінка після входу

Після входу до системи користувач потрапляє на головну сторінку веб-додатка (рис 3.3). Інтерфейс адаптований під мобільні пристрої.

У верхній частині відображається привітання користувача за ім'ям, а також коротке мотиваційне повідомлення. Нижче розташовано панель навігації з розділами:

- Активності
- Мій прогрес
- Харчування
- Спільнота

Центральну частину займають:

- Блок рекомендацій – показує персоналізовані вправи з назвою, тривалістю та витраченими калоріями.
- Тижневий челендж – окрема секція з візуальним акцентом, що мотивує користувача на регулярні тренування.
- Статті – добірка корисних матеріалів про харчування, фізичну активність та здоровий спосіб життя.

У нижній частині екрана знаходиться навігаційна панель, що забезпечує швидкий доступ до основних розділів додатку.

3. Розділ «Мій прогрес»

У модулі «Мій прогрес» реалізовано зручний інтерфейс для перегляду фізичної активності користувача та аналізу динаміки досягнень. Цей розділ містить дві вкладки: Журнал та Графік, між якими можна перемикатися для перегляду даних у різних форматах.

Інтерфейс «Журнал» (Рис 3.4)

У вкладці Журнал відображається календар, що дозволяє обрати конкретну дату для перегляду активностей. Нижче календаря представлено список тренувань за обраний день із зазначенням типу тренування, тривалості та часу проведення. Для кожного запису зазначено:

- Назву тренування (наприклад, Upper Body Workout);
- Частина тіла, на яку спрямоване тренування;
- Час початку та тривалість заняття.

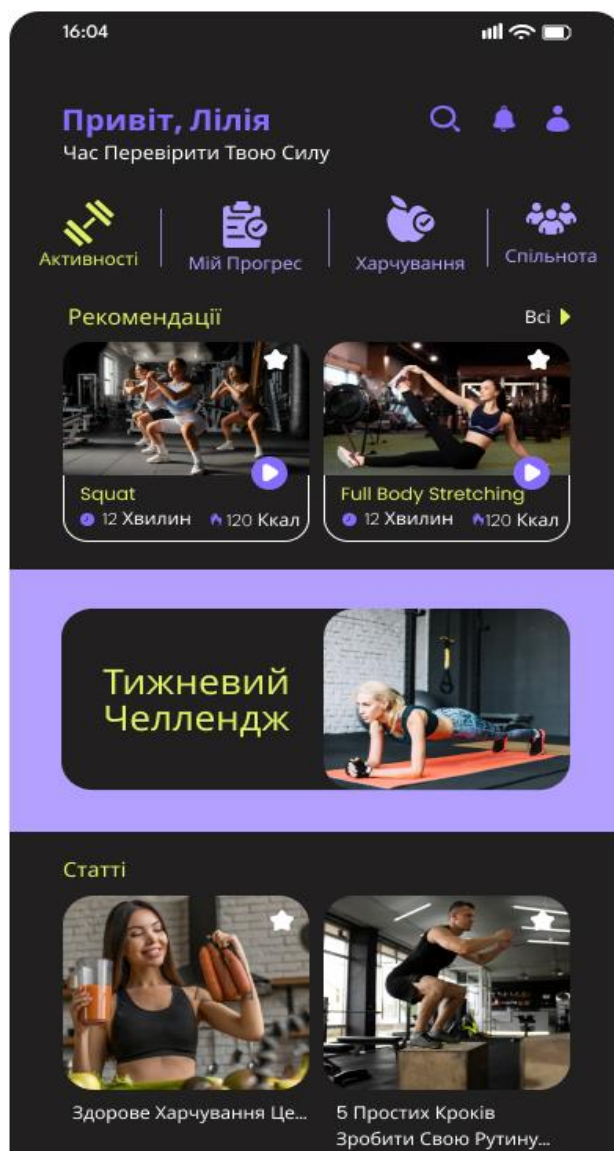


Рис 3.3. Головна сторінка після входу (мобільна версія)

Цей підрозділ допомагає користувачеві відстежувати власні активності у форматі щоденника.

У вкладці Графік реалізовано візуалізацію прогресу за допомогою стовпчикової діаграми. Користувач може побачити кількість зроблених кроків у різні місяці. Для деталізації показників нижче виводиться інформація про активність у конкретні дні, зокрема:

- Дата;
- Кількість кроків;
- Загальний час активності.

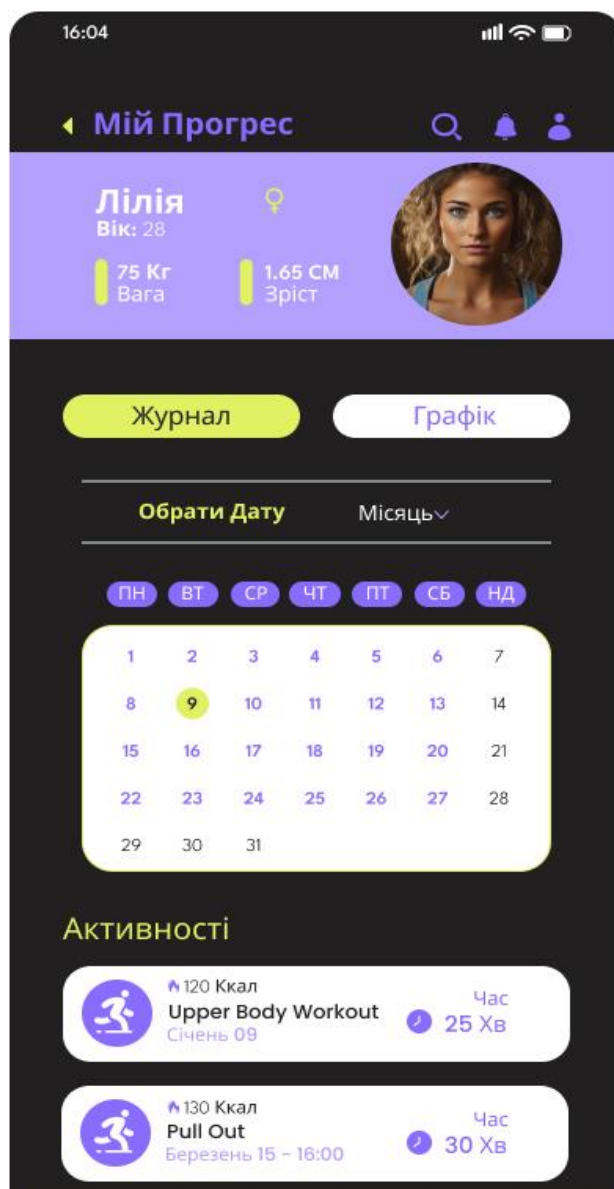


Рис 3.4. Розділ «Мій прогрес» Інтерфейс «Журнал»

Інтерфейс «Графік» (Рис 3.5)

Цей інтерфейс дозволяє користувачеві оцінити загальну динаміку прогресу, визначити найактивніші дні та місяці, а також скоригувати свої тренування відповідно до результатів.

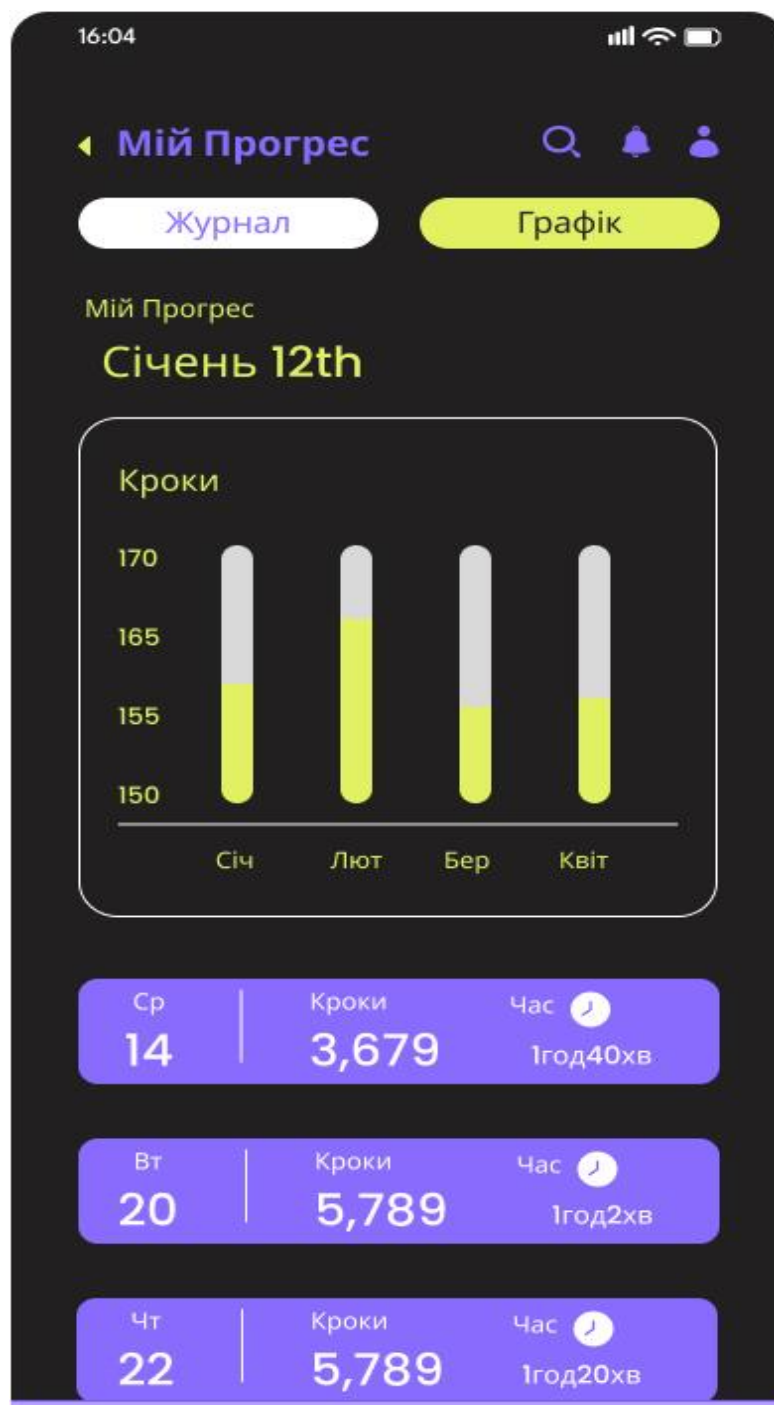


Рис 3.5. Розділ «Мій прогрес» Інтерфейс «Графік»

4. Розділ «Харчування»

Головний екран модуля харчового планування відображає користувачеві сітку категорій харчування, серед яких: «Схуднення», «Підтримка ваги», «Набір ваги» (Рис 3.6). Для кожної категорії використано такі підрозділи як «Рецепт дня»,

«Рекомендуємо», «Рецепти для тебе», а також високоякісні ілюстрації, що асоціюються зі змістом, що значно покращує візуальне сприйняття інформації.

При натисканні на картку здійснюється виклик API-запиту до серверної частини з параметрами вибраної категорії, що дозволяє вивести рекомендовані страви або дієти на основі вподобань користувача.



Рис 3.6. Головний екран вибору категорій харчування

5. Калькулятор індексу маси тіла (ІМТ)

Наступним елементом інтерфейсу є екран калькулятора ваги, де користувач може ввести власні антропометричні дані — вік, зріст у сантиметрах та вагу в кілограмах (Рис 3.7).

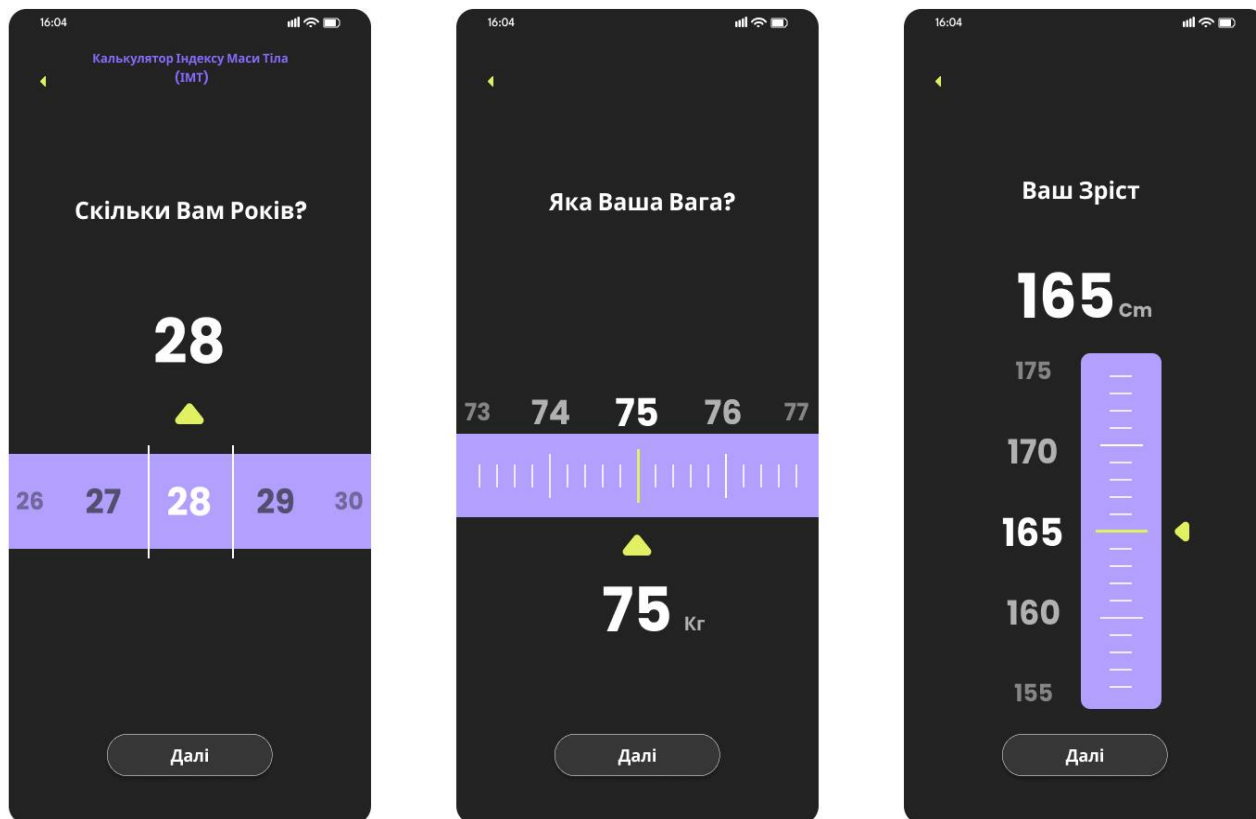


Рис 3.7. Калькулятор ІМТ

Після заповнення обох полів натисканням кнопки "Далі" здійснюється обчислення ІМТ за формулою:

$$ІМТ = \text{вага (кг)} / (\text{зріст (м)})^2$$

Результат обчислення відображається у центрі екрана в стилізованому круглому елементі, який додатково супроводжується вербальною інтерпретацією результату ("дефіцит маси", "норма", "надмірна вага") та змінює кольорову палітру залежно від результату: синій — нейтральне значення, зелений — норма, червоний — надлишок (Рис 3.8).

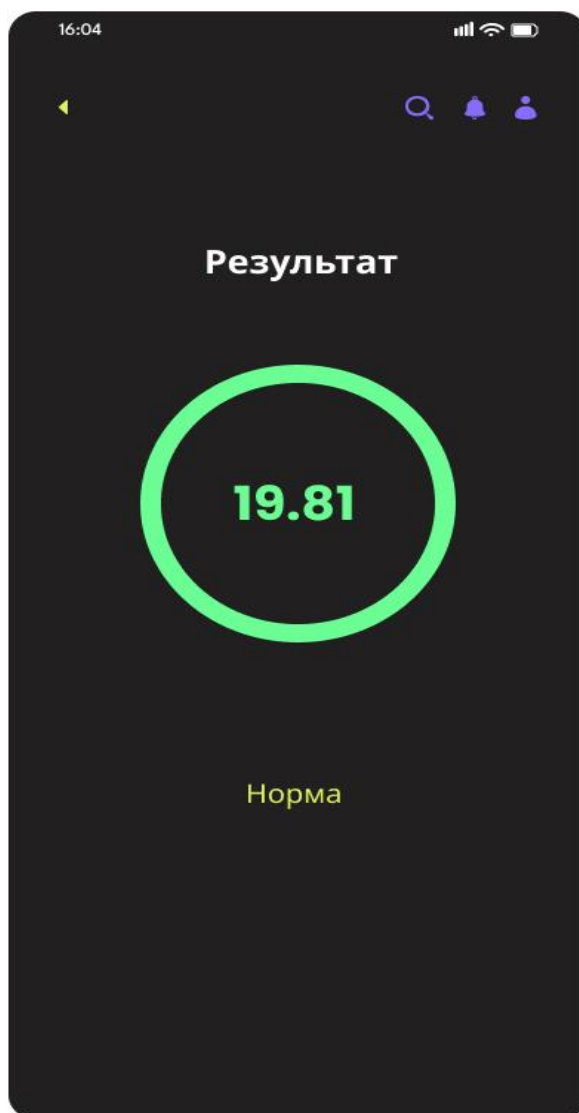


Рис 3.8. Обчислений результат: ІМТ у межах норми

6. Мотиваційний екран

Окрема вкладка присвячена відображенню **мотиваційних цитат**, які щоденно оновлюються. У реалізації використано механізм вибору випадкової цитати з попередньо визначеного масиву.

Дана функція покликана підтримувати моральну складову користувача, сприяючи регулярності виконання тренувань і формуванню позитивного емоційного фону (Рис 3.9).

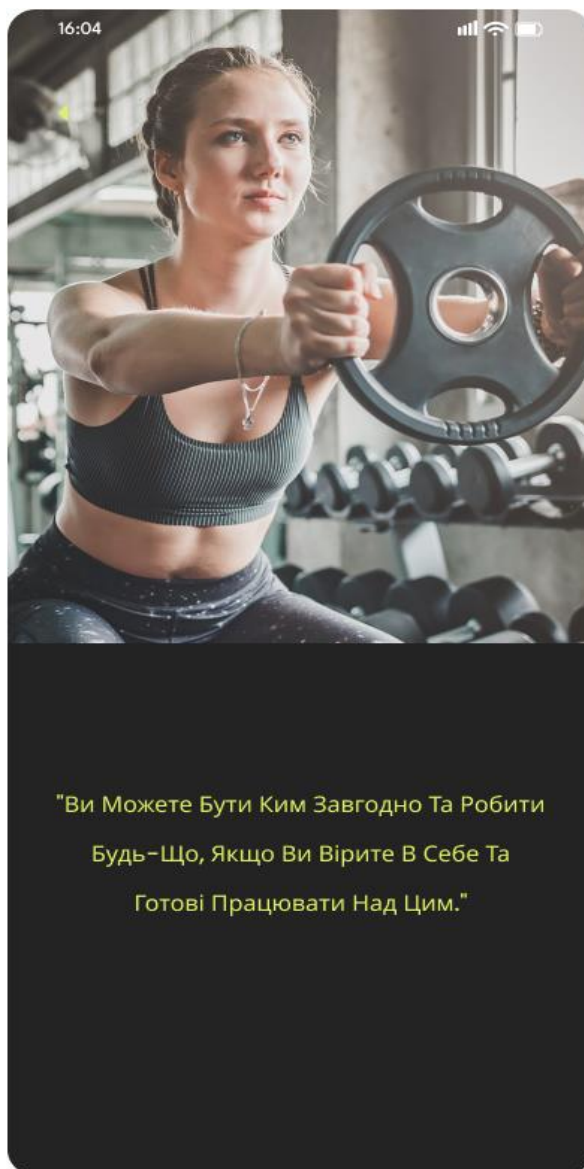


Рис 3.9. Мотиваційний екран

7. Розділ «Обране» для персоналізованого збереження контенту

Механізм збереження контенту в обране є одним з найзручніших засобів організації персонального простору користувача. Вкладка «Обране» складається з трьох логічних блоків: «Тренування», «Блог», «Харчування» (Рис 3.10). У кожному блоці виводяться відповідні елементи, які користувач позначив як важливі. Реалізовано функціонал видалення та динамічного оновлення елементів у режимі реального часу.

Візуально кожен обраний елемент представлено у вигляді картки з мініатюрою зображення та назвою.

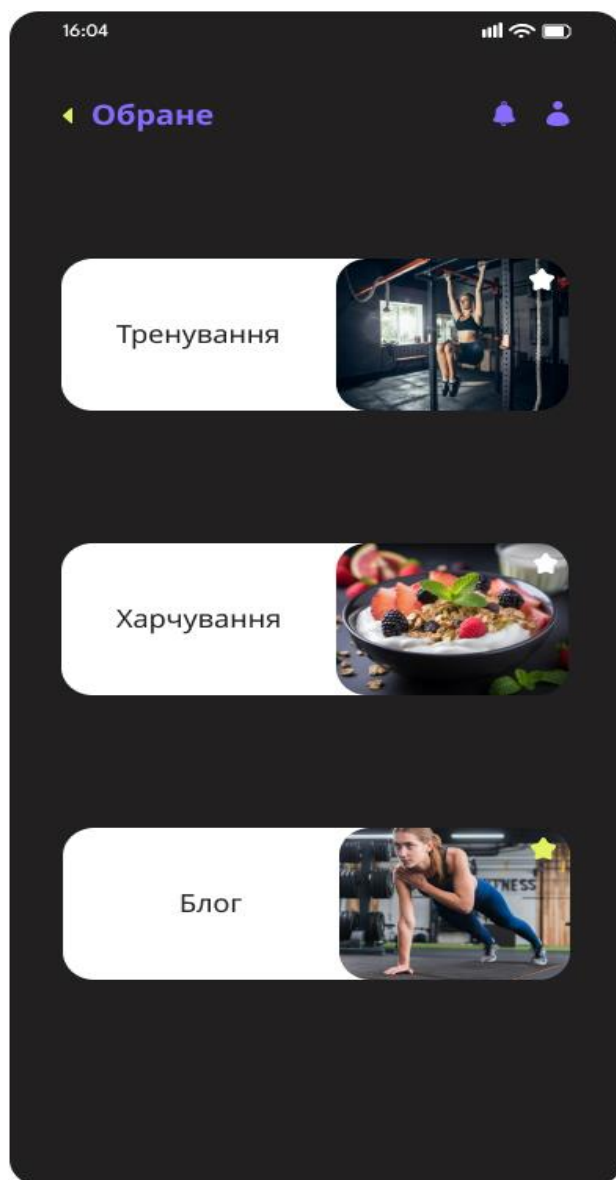


Рис 3.10. Відображення персонального списку обраного користувача

Таблиця відповідності UI-модулів функціональним цілям (Таб 3.1):

Таблиця 3.1.

Відповідність функціональних блоків клієнтського інтерфейсу цілям застосунку

| Назва модуля | Призначення | Тип взаємодії з користувачем |
|------------------------|---|--|
| Мій прогрес | Контроль прогресу | Вибір дати, перемикання між вкладками, натискання на елементи активності |
| Категорії харчування | Навігація та вибір плану харчування | Картки з класифікацією, що ведуть до деталей |
| Калькулятор ІМТ | Обчислення фізичного стану тіла | Введення параметрів, натискання кнопки |
| Мотиваційний екран | Психологічна підтримка користувача | Пасивне сприйняття, оновлення контенту |
| Обране | Збереження та швидкий доступ до улюблених елементів | Додавання, видалення, перелік |
| Навігація між екранами | Перехід між функціональними модулями | Кнопки, таби, адаптивна верстка |

3.3 Переваги розробленого web додатку

У процесі розробки та впровадження веб-додатку було досягнуто низки ключових переваг, які вирізняють систему серед аналогічних рішень у сфері моніторингу фізичної активності та здорового способу життя. Переваги стосуються як функціонального наповнення, так і технологічної реалізації, користувацького досвіду та гнучкості подальшого розвитку системи.

1. Інтуїтивно зрозумілий інтерфейс

Користувацький інтерфейс побудований з урахуванням сучасних принципів UI/UX-дизайну. Навігація реалізована у вигляді зрозумілих вкладок, іконок та кнопок, що дозволяє навіть недосвідченим користувачам легко орієнтуватися в додатку. Окремі модулі, такі як Мій прогрес, калькулятор ІМТ, мотиваційний блок

або перегляд харчування, не потребують додаткових інструкцій і доступні з першого використання.

2. Персоналізація користувацького досвіду

Додаток зберігає вподобання користувача (вибрані дієти, тренування, статті) та формує індивідуальні рекомендації. Це дозволяє кожному користувачеві створити власний інформаційний простір і повертатися до нього без необхідності повторного пошуку.

3. Автоматичний аналіз фізичного стану

Модуль ІМТ забезпечує миттєвий аналіз параметрів тіла користувача із виведенням результату у візуальній формі. Це дозволяє оперативно оцінити свій фізичний стан та коригувати план дій — обрати відповідну дієту, режим тренувань або звернути увагу на динаміку.

4. Мотиваційна підтримка

Система не обмежується сухою статистикою — користувач отримує емоційне підкріплення у вигляді мотиваційних повідомлень. Це сприяє регулярному використанню додатку, знижує ризик «вигорання» під час фізичних навантажень та підвищує загальний рівень задоволеності застосунком.

5. Технологічна гнучкість та масштабованість

Використання Node.js, Express.js та MySQL забезпечує високу швидкодію системи та можливість масштабування при збільшенні кількості користувачів. Компонентна структура дозволяє безперешкодно інтегрувати нові модулі: наприклад, розділ «Пульсометрія», підключення трекера активності або чат з тренером.

6. Кросплатформенність і мобільна адаптивність

Інтерфейс повністю адаптований до використання на мобільних пристроях: усі кнопки масштабуються, елементи автоматично перебудовуються, забезпечується зручність натискання навіть на невеликих екранах. Додаток тестувався на Android/iOS смартфонах, планшетах та десктопах з різними розширеннями екрана.

Таблиця 3.2

Таблиця порівняння функціональності з типовим рішенням

| Ознака | Розроблений веб-додаток | Типовий фітнес-трекер |
|-----------------------------------|------------------------------|-------------------------------------|
| Інтерфейс | Адаптивний, структурований | Складна або перевантажена навігація |
| Аналіз фізичних даних | Калькулятор ІМТ, аналітика | Обмежена або відсутня статистика |
| Персоналізація контенту | Збереження обраного, історія | Фіксований шаблон |
| Мотиваційні елементи | Динамічні повідомлення | Зазвичай відсутні |
| Можливість розширення функціоналу | Висока, модульна архітектура | Закрита система |
| Платформа | Web + мобільна адаптація | Лише мобільний застосунок |

РОЗДІЛ 4. АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ ТА НАПРЯМИ ПОДАЛЬШОГО РОЗВИТКУ СИСТЕМИ

У цьому розділі здійснюється оцінка результатів реалізації web-додатку для комп'ютерного моніторингу та обробки досягнень учасників спортивних заходів. Після завершення процесу розробки та впровадження системи важливим етапом є аналіз її ефективності, зручності використання, відповідності поставленим завданням і технічним вимогам.

Основна увага приділяється аналізу практичних результатів функціонування програмного забезпечення, а також визначенню можливих напрямів його удосконалення та масштабування. Оцінюється потенціал інтеграції системи з іншими цифровими платформами, що може забезпечити її гнучкість, мобільність і підвищену цінність для кінцевого користувача.

Крім того, в цьому розділі розглядаються ідеї щодо покращення функціональності, інтерфейсу та користувацького досвіду. Окремий підпункт присвячено узагальненню висновків щодо досягнутих результатів, а також перспективам подальшого розвитку розробленого продукту.

4.1. Аналіз ефективності створеного програмного забезпечення

Ефективність програмного забезпечення визначається здатністю реалізованої системи вирішувати поставлені задачі, відповідати вимогам користувачів та забезпечувати стабільну, безпечну та зручну взаємодію. У процесі реалізації веб-додатку для комп'ютерного моніторингу та обробки результатів досягнень учасників спортивних заходів основними критеріями ефективності виступали: відповідність функціональним вимогам, стабільність роботи системи, зручність інтерфейсу, швидкодія, безпека та позитивний досвід користувача.

Відповідність функціональним вимогам. На етапі аналізу предметної області та постановки задач було сформульовано перелік функцій, які система повинна реалізовувати. Серед них — реєстрація та автентифікація користувачів, введення та

редагування тренувань, ведення статистики, автоматичний аналіз результатів, формування звітів, а також інтеграція з зовнішніми платформами. Під час реалізації кожна з зазначених функцій була впроваджена відповідно до специфікації:

Форма реєстрації дозволяє створювати облікові записи з валідацією даних.

Функціонал тренувань реалізовано через CRUD-операції (create-read-update-delete), що забезпечує повний життєвий цикл введення фізичної активності.

Модуль аналітики надає користувачу зведену інформацію щодо прогресу (загальна кількість тренувань, витрачені калорії, середня тривалість).

Генерація звітів реалізована через графіки та таблиці, які автоматично оновлюються залежно від введених даних.

Інтеграція з фітнес-платформами (Google Fit, Apple Health) закладена в архітектурі через API.

Таким чином, усі ключові функціональні вимоги виконані, що підтверджує функціональну повноту рішення.

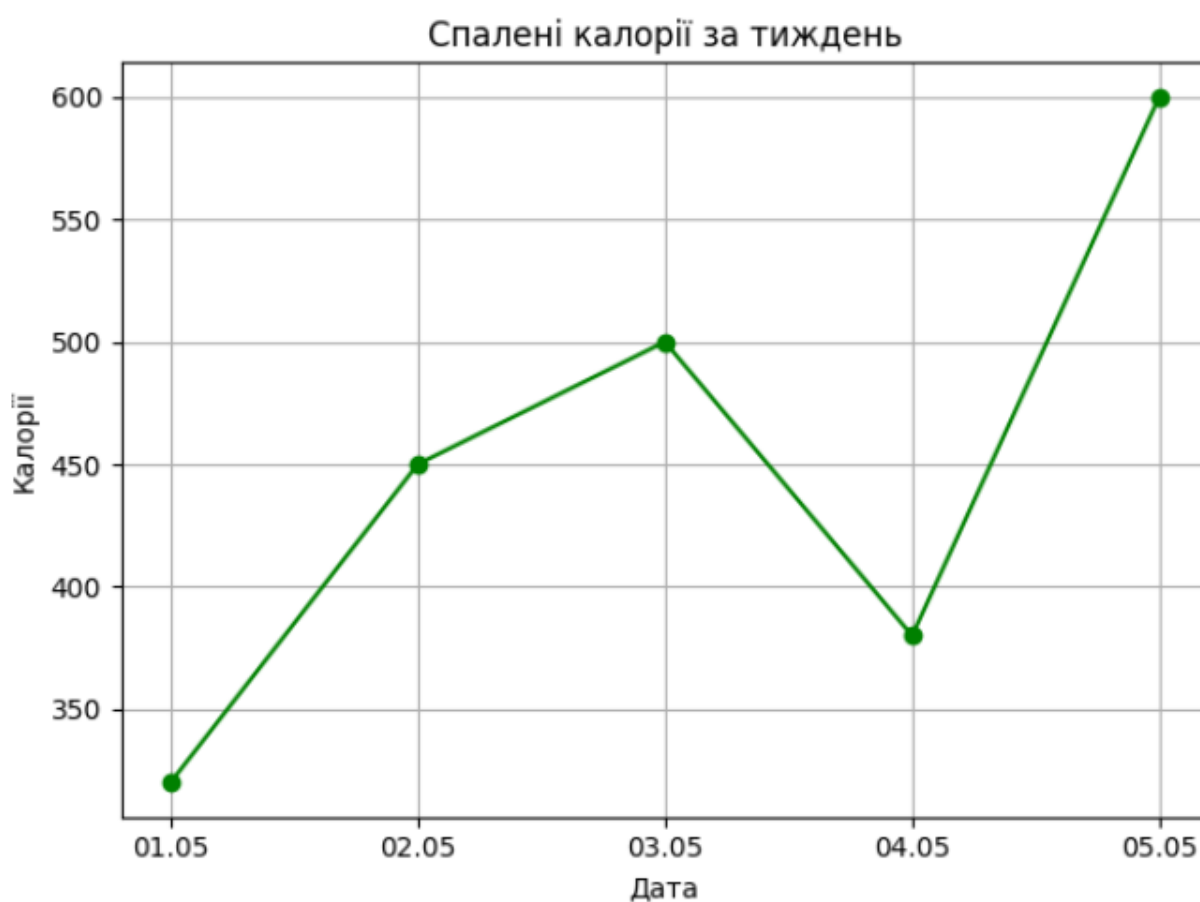


Рис. 4.1 Приклад вигляду графіка спалених калорій



Рис. 4.2 Приклад вигляду різних типів тренувань



Рис 4.3. Приклад відображення кількості тренувань протягом тижня

Стабільність та швидкодія системи. Для оцінки стабільності системи проводилося тестування додатку в умовах імітації навантаження — одночасного використання системи кількома користувачами. Було змодельовано до 100 одночасних підключень через емулятор запитів, що дозволило перевірити обробку запитів на додавання тренувань, вхід у систему та перегляд статистики. У результаті система не зазнала збоїв, помилок обробки чи втрати даних.

Час відповіді сервера на запити становив у середньому 800–1100 мс, що є прийнятним показником для веб-додатку без застосування кешування. В майбутньому, за потреби масштабування, можливе впровадження Redis або аналогічних рішень для оптимізації.

Також були протестовані ситуації втрати з'єднання з інтернетом, під час яких система адекватно обробляла помилки та зберігала дані після повторного підключення.

Оцінка зручності використання. Користувацький інтерфейс був протестований на групі з 10 користувачів, що включали як досвідчених користувачів фітнес-додатків, так і новачків. Було виявлено, що:

- 9 з 10 користувачів оцінили інтерфейс як «зрозумілий» або «дуже зручний»;
- 8 з 10 вказали, що можуть виконати основні дії (дати тренування, переглянути статистику) без інструкцій;

Усі користувачі позитивно оцінили візуальне відображення результатів — графіки, колірну схему та адаптивність під мобільні пристрої.

Інтерфейс системи має інтуїтивну навігацію, логічно згруповані функції, передбачувану взаємодію з елементами та адаптивний дизайн, що дозволяє працювати як на десктопах, так і на мобільних пристроях.

Захищеність даних. У процесі розробки було реалізовано механізми захисту персональних даних:

- Паролі зберігаються у хешованому вигляді (алгоритм bcrypt);
- Для автентифікації використовується JWT-токенізація, що забезпечує безпечний обмін даними між клієнтом і сервером;
- Реалізована перевірка прав доступу до ресурсів (авторизовані запити);

- Сервер підтримує CORS та HTTPS-з'єднання.

Таким чином, реалізовані базові вимоги щодо інформаційної безпеки, а також закладено потенціал для сертифікації відповідно до стандартів GDPR.

Рівень досягнення поставлених цілей. Порівняння результатів з очікуваними цілями, сформульованими в розділі 1.4, демонструє:

- Реалізовано персоналізовану систему тренувань;
- Здійснено базову інтеграцію з фітнес-платформами;
- Здійснюється облік і аналіз результатів у зручному форматі;
- Забезпечено стабільність роботи і позитивний користувацький досвід.

Це свідчить про досягнення як функціональної, так і практичної ефективності веб-додатку.

Інтеграція з зовнішніми платформами. Важливою складовою ефективності веб-додатку є можливість інтеграції з популярними фітнес-платформами та пристроями для збору фізичних даних. В рамках розробки було реалізовано базову інтеграцію з такими сервісами, як Google Fit та Apple Health, що дозволяє користувачам автоматично імпортувати інформацію про тренування та фізичну активність. Це суттєво підвищує зручність роботи з додатком, зменшує навантаження на користувача при введенні даних і сприяє більш точному моніторингу результатів.

Забезпечення безпеки та конфіденційності даних. Одним із ключових критеріїв ефективності системи є надійність захисту персональних даних користувачів. В розробленому додатку застосовані сучасні методи безпеки, включно з хешуванням паролів (алгоритм bcrypt), використанням JWT-токенів для автентифікації та авторизації, а також захищеними каналами передачі даних (HTTPS). Ці заходи гарантують конфіденційність інформації, що зберігається та обробляється в системі.

Оцінка результатів впровадження. Проведений аналіз реалізованої системи показує, що запропоноване рішення дозволяє ефективно виконувати поставлені завдання з моніторингу спортивних досягнень. Веб-додаток забезпечує коректне збереження, обробку та візуалізацію даних, що підтверджує високу якість розробки.

4.2. Перспективи масштабування та інтеграції з іншими системами

Розроблений веб-додаток має гнучку архітектуру, що передбачає можливість масштабування та розширення функціональних можливостей у майбутньому. Використання технологічного стеку Node.js для серверної частини і MySQL для збереження даних забезпечує базові умови для ефективної роботи навіть при збільшенні кількості користувачів. Асинхронна обробка запитів і модульна організація коду дозволяють легко адаптувати систему під зростаюче навантаження.

У процесі масштабування можна розглядати кілька напрямків. Перш за все, це горизонтальне масштабування серверної частини шляхом додавання додаткових серверів і використання балансувальників навантаження, що дозволить рівномірно розподілити вхідні запити і уникнути перевантаження окремих вузлів. Окрім цього, можливе вертикальне масштабування — збільшення ресурсів апаратної частини серверів, що особливо актуально для обробки великих обсягів даних або складних аналітичних запитів.

Важливим аспектом масштабованості є оптимізація роботи бази даних. Застосування індексації, а також використання кешування на рівні сервера (наприклад, за допомогою Redis) можуть значно покращити час відповіді на запити, особливо у випадках інтенсивного користування додатком. У перспективі можливе застосування розподілених баз даних або шардингу, що дозволить ефективно працювати з великими обсягами даних та численною аудиторією.

Що стосується інтеграції, веб-додаток вже зараз підтримує основні функції імпорту даних із популярних фітнес-платформ, таких як Google Fit та Apple Health, що значно підвищує зручність користування і точність моніторингу. Проте, існують значні можливості для розширення інтеграцій з іншими сервісами та пристроями. Планується включення підтримки платформ Fitbit та Samsung Health, що дозволить залучити ширшу аудиторію користувачів із різними типами пристроїв. Важливим напрямом є також розвиток інтеграцій з медичними інформаційними системами та реабілітаційними центрами, що відкриває перспективи для професійного використання веб-додатку.

Для забезпечення більш тісної взаємодії користувачів із системою планується розробка мобільних застосунків або прогресивного веб-додатку (PWA), що дозволить отримувати доступ до функціоналу з будь-яких мобільних пристроїв, а також розширить аудиторію і збільшить зручність використання.

Також перспективним є впровадження API для сторонніх розробників, що відкриє можливість створення додаткових сервісів та розширень на основі даних користувачів, а також інтеграції з іншими програмними продуктами, які можуть доповнити функціонал або підвищити цінність додатку.

Важливим фактором подальшого розвитку є забезпечення безпеки даних при масштабуванні та інтеграції. Веб-додаток використовує сучасні стандарти захисту персональних даних, такі як хешування паролів, використання JWT для безпечної аутентифікації та авторизації, а також передачу даних через захищені канали. Ці заходи мають бути збережені і посилені з розвитком системи, щоб гарантувати конфіденційність та захист інформації навіть при збільшеному навантаженні та розширенні інтеграцій.

Таким чином, розроблений веб-додаток має добре спроектовану основу для масштабування і розширення функціональності, що дозволяє йому адаптуватися до змінних потреб користувачів і технологічного середовища. Перспективи інтеграції з популярними платформами і пристроями, а також можливість впровадження нових сервісів роблять цей продукт конкурентоспроможним та актуальним для широкого кола користувачів.

4.3. Пропозиції щодо майбутніх удосконалень

Розроблений веб-додаток має значний потенціал для подальшого розвитку та вдосконалення, що дозволить розширити функціональність, покращити користувацький досвід і забезпечити більшу адаптивність до потреб різних категорій користувачів.

Перш за все, одним із важливих напрямків є поліпшення користувацького інтерфейсу. Незважаючи на вже реалізований адаптивний дизайн, подальша

оптимізація інтерфейсу, впровадження темної теми, розширення налаштувань персоналізації дозволить зробити використання додатку ще більш комфортним і привабливим для користувачів з різними вподобаннями та потребами. Також актуальним є впровадження більш інтуїтивної навігації та покращення візуалізації звітів, що допоможе користувачам швидше отримувати необхідну інформацію та приймати рішення на її основі.

Другим важливим напрямом є розробка мобільної версії додатку або прогресивного веб-додатку (PWA). З огляду на широке використання мобільних пристроїв, наявність повноцінного мобільного клієнта сприятиме збільшенню аудиторії, забезпечить зручність користування та дозволить отримувати миттєві сповіщення і нагадування, що підвищить рівень залученості користувачів.

Ще однією перспективною можливістю є впровадження систем штучного інтелекту та машинного навчання для персоналізації рекомендацій і аналізу тренувань. Такий підхід дозволить на основі накопичених даних прогнозувати можливі результати, виявляти закономірності у фізичних показниках і формувати індивідуальні плани тренувань і відновлення. Це особливо важливо для користувачів із особливими потребами, включаючи спортсменів, які проходять реабілітацію після травм, або військовослужбовців, що потребують специфічної підтримки.

Іншим напрямом є розширення соціальних функцій платформи. Введення можливості створення груп за інтересами, проведення змагань між користувачами, обміну досвідом та підтримки мотивації через рейтинги і нагороди створить більш інтерактивне і залучене співтовариство, що сприятиме регулярному використанню додатку і підвищенню ефективності тренувань.

Також важливим є подальший розвиток інтеграції з різноманітними пристроями та платформами, розширення API для сторонніх розробників, що дозволить створювати додаткові сервіси та розширення, а також підвищить універсальність системи.

З погляду безпеки, планується впровадження додаткових рівнів захисту, зокрема двофакторної автентифікації, моніторингу підозрілої активності, а також регулярних оновлень системи для усунення можливих вразливостей.

В цілому, наведені пропозиції щодо удосконалень формують чітку стратегію розвитку веб-додатку, що забезпечить його відповідність сучасним технологічним тенденціям та потребам користувачів, підвищить конкурентоспроможність і дозволить охопити ширшу аудиторію.

4.4. Висновки щодо ефективності розробки та перспектив розвитку

Проведений аналіз реалізованого веб-додатку для комп'ютерного моніторингу та обробки спортивних досягнень підтверджує, що розроблена система відповідає основним вимогам, визначеним у постановці задачі, і забезпечує широкий функціонал, необхідний для комплексного контролю фізичної активності користувачів. Реалізовані модулі реєстрації, автентифікації, ведення тренувань, аналізу спортивних показників та генерації звітів працюють стабільно, забезпечують високу продуктивність і позитивний користувацький досвід.

Завдяки вибору сучасного технологічного стеку та архітектурних рішень система має хороші показники швидкодії і надійності. Асинхронна обробка запитів на сервері, оптимізована робота з базою даних і модульна структура коду забезпечують масштабованість і гнучкість системи, що дозволить у майбутньому ефективно реагувати на збільшення навантаження та потреби користувачів.

Забезпечення безпеки персональних даних та конфіденційності інформації є одним із пріоритетних напрямків розробки, реалізовані механізми захисту відповідають сучасним стандартам, що підвищує рівень довіри користувачів і відкриває можливості для застосування системи у професійних та медичних сферах.

Аналіз можливостей подальшого розвитку системи показує, що веб-додаток має значний потенціал для вдосконалення. Розширення інтеграції з різноманітними платформами і пристроями дозволить збільшити аудиторію користувачів і підвищити точність моніторингу. Розробка мобільної версії, впровадження штучного інтелекту для персоналізації тренувальних планів і розширення соціальних функцій зробить систему більш привабливою і конкурентоспроможною.

Пропоновані удосконалення спрямовані на покращення користувацького досвіду, розширення функціональних можливостей і забезпечення високого рівня безпеки, що створює міцну основу для комерційного застосування і масштабування продукту.

Отже, реалізований веб-додаток є ефективним і перспективним рішенням для моніторингу спортивних досягнень, здатним адаптуватися до потреб користувачів і сучасних технологічних вимог. Його подальший розвиток та вдосконалення забезпечать довготривалу актуальність та успішність на ринку.

ВИСНОВКИ

У межах даної кваліфікаційної роботи було розроблено web-додаток для комп'ютерного моніторингу та аналізу спортивних досягнень користувача. Проведене дослідження підтвердило актуальність теми в умовах зростання популярності здорового способу життя, самостійних тренувань та потреби в персоналізованому контролі фізичних показників.

В результаті виконання поставлених задач було досягнуто наступних результатів:

- Проведено аналіз предметної області, вивчено наявні програмні рішення та виявлено їхні недоліки.
- Сформовано вимоги до системи. Було запропоновано функціональність для відстеження активності, тренувань, харчування, а також систему аналітики та звітності.
- Розроблено архітектуру системи на основі клієнт-серверної моделі, обґрунтовано вибір технологій: Node.js, Express.js, MySQL, HTML, CSS, JavaScript. Створено REST API для ефективної взаємодії між клієнтом і сервером.
- Реалізовано ключові функціональні модулі, серед яких: реєстрація та автентифікація користувачів, введення та перегляд тренувань, журнал і графік активності, візуалізація статистики, перегляд персональних параметрів.
- Забезпечено захист персональних даних, що є особливо важливим для користувачів, які вводять медичну або приватну інформацію.
- Проведено тестування системи, яке показало стабільність роботи додатку, швидкість обробки даних і зручність інтерфейсу. Отримані результати підтверджують ефективність розробленого рішення.
- Надано рекомендації щодо подальшого розвитку, зокрема: розширення функціональності (наприклад, впровадження трекінгу харчування), інтеграція з фітнес-пристроями, додавання психологічних аспектів для комплексної підтримки користувачів, впровадження мобільної версії.

У підсумку, розроблений web-додаток є сучасним, функціональним інструментом, який дозволяє користувачам ефективно відстежувати свої спортивні

досягнення, аналізувати прогрес та підвищувати результативність тренувань. Робота має як практичне, так і теоретичне значення, та може бути основою для подальших досліджень або вдосконалення в рамках розробки програмного забезпечення у сфері цифрового здоров'я.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мобільний додаток MyFitnessPal. – URL:

<https://www.morningdough.com/uk/ai-tools/my-fitness-pal/>

2. Мобільний додаток Strava. – URL:

[https://uk.wikipedia.org/wiki/Strava#:~:text=str%C3%A4va%20%E2%80%94%20%20%AB%D0%BF%D1%80%D0%B0%D0%B3%D0%BD%D1%83%D1%82%D0%B8%C2%BB\)%20%E2%80%94,%D0%B1%D1%96%D0%B3%20%D1%96%20%D1%97%D0%B7%D0%B4%D0%B0%20%D0%BD%D0%B0%20%D0%B2%D0%B5%D0%BB%D0%BE%D1%81%D0%B8%D0%BF%D0%B5%D0%B4%D1%96.](https://uk.wikipedia.org/wiki/Strava#:~:text=str%C3%A4va%20%E2%80%94%20%20%AB%D0%BF%D1%80%D0%B0%D0%B3%D0%BD%D1%83%D1%82%D0%B8%C2%BB)%20%E2%80%94,%D0%B1%D1%96%D0%B3%20%D1%96%20%D1%97%D0%B7%D0%B4%D0%B0%20%D0%BD%D0%B0%20%D0%B2%D0%B5%D0%BB%D0%BE%D1%81%D0%B8%D0%BF%D0%B5%D0%B4%D1%96.)

3. Мобільний додаток FatSecret. – URL:

<https://www.fatsecret.com/Home.aspx?pa=privd&l=uk>

4. Мобільний додаток Headspace. – URL:

<https://drukarnia.com.ua/articles/estetichni-ta-korisni-mobilni-zastosunki-sho-dopomozhut-pokrashiti-mentalne-zdorov-ye-v0LcW>

5. Посібник О.В. Виноградова, Н.М. Недопако «Маркетинг у соціальних мережах». - 2022.

6. Платформа Google Fit. – URL:

https://www.google.com/intl/en_us/fit/

7. Платформа Apple Health. – URL:

<https://www.apple.com/health/>

8. Платформа Fitbit. – URL:

https://store.google.com/regionpicker?hl=en-US&utm_source=fitbit_redirect&utm_medium=google_ooo&utm_campaign=category&pli=1

9. Історія Операційних Систем. – URL:

<https://vseosvita.ua/library/istoriia-operatsiinykh-system-875527.html>

10. Користувачький інтерфейс (UI) . – URL:

[https://prjctr.com/mag/uxui-questions#:~:text=UI%20\(User%20Interface\)%20%E2%80%94%D1%86%D0%B5,Experience\)%20%E2%80%94%D1%86%D0%B5%20%D0%BA%D0%BE%D1%80%D0](https://prjctr.com/mag/uxui-questions#:~:text=UI%20(User%20Interface)%20%E2%80%94%D1%86%D0%B5,Experience)%20%E2%80%94%D1%86%D0%B5%20%D0%BA%D0%BE%D1%80%D0)

[%B8%D1%81%D1%82%D1%83%D0%B2%D0%B0%D1%86%D1%8C%D0%BA%D0%B8%D0%B9%20%D0%B4%D0%BE%D1%81%D0%B2%D1%96%D0%B4.](#)

11. Вибір технологічного стеку. – URL:

<https://foxminded.ua/stek-u-prohramuvanni/>

12. Node.js. – URL:

<https://nodejs.org/uk/about>

13. Express.js. – URL:

<https://expressjs.com/>

14. MySQL. – URL:

https://scalegrid.io/mysql/?kw=&cpn=21514540474&utm_term=&utm_campaign=Re

[dis+-](#)

[+REMARKETING&utm_source=adwords&utm_medium=ppc&hsa_mt=&hsa_ad=&hsa_ne](#)

[t=adwords&hsa_src=x&hsa_kw=&hsa_tgt=&hsa_cam=21514540474&hsa_acc=242843520](#)

[6&hsa_ver=3&hsa_grp=&gad_source=1&gad_campaignid=21510648143&gbraid=0AAAA](#)

[ADusSkxCXEVI55TwX1LCwoVDb_g1x&gclid=CjwKCAjw56DBBhAkEiwAaFsG-](#)

[hypMFVfM9Vq2D1Z4EL-ASDQ8LSRvD3XAzit4BrkHk0ILUYRSD-](#)

[G7hoCLj4QAvD_BwE](#)

15. Архітектура системи. – URL:

<https://medium.com/@IvanZmerzlyi/%D0%BA%D0%BB%D1%96%D1%94%D0%B>

[D%D1%82-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80%D0%BD%D0%B0-](#)

[%D0%B0%D1%80%D1%85%D1%96%D1%82%D0%B5%D0%BA%D1%82%D1%83%D](#)

[1%80%D0%B0-%D1%82%D0%B0-%D1%80%D0%BE%D0%BB%D1%96-](#)

[%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80%D1%96%D0%B2-](#)

[9893d8048229](#)

16. Клієнтська частина (Frontend) . – URL:

<https://gl.ua/blog/shcho-potribno-znaty-shchob-staty-frontend->

[developer#:~:text=Front%2Dend%20%E2%80%94%D1%86%D0%B5%20%D0%BA%](#)

[D0%BB%D1%96%D1%94%D0%BD%D1%82%D1%81%D1%8C%D0%BA%D0%B0%20](#)

[\(,%D1%96%D0%BD%D1%88%D1%96%20%D0%B2%D1%96%D0%B7%D1%83%D0%](#)

[B0%D0%BB%D1%8C%D0%BD%D1%96%20%D0%B5%D0%BB%D0%B5%D0%BC%](#)

[D0%B5%D0%BD%D1%82%D0%B8%20%D1%96%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82%2D%D1%80%D0%B5%D1%81%D1%83%D1%80%D1%81%D1%83.](#)

17. Серверна частина (Backend) . – URL:

<https://wezom.com.ua/ua/blog/chto-takoe-back-end-razrabotka#:~:text=Back%2Dend%20%E2%80%94%D1%86%D0%B5%20%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80%D0%BD%D0%B0%20%D1%87%D0%B0%D1%81%D1%82%D0%B8%D0%BD%D0%B0,%D0%B2%D0%B7%D0%B0%D1%94%D0%BC%D0%BE%D0%B4%D1%96%D1%8E%20%D0%B7%20%D0%B7%D0%BE%D0%B2%D0%BD%D1%96%D1%88%D0%BD%D1%96%D0%BC%D0%B8%20API%20%D1%82%D0%BE%D1%89%D0%BE.>

18. REST API. – URL:

<https://goit.global/ua/articles/rest-api-shcho-tse-iak-pratsiuie-perevahy-i-prykklady/>

19. Redis. – URL:

<https://uk.wikipedia.org/wiki/Redis>

20. MDN Web Docs. *HTML, CSS, and JavaScript documentation*. Mozilla Developer Network. . – URL:

<https://developer.mozilla.org>

21. W3Schools. *JavaScript, HTML, CSS tutorials*. – URL:

<https://www.w3schools.com>

22. Node.js Documentation. *Node.js v20+ API reference*. – URL:

<https://nodejs.org/en/docs>

23. Express.js Guide. *Getting started with Express 4.x*. – URL:

<https://expressjs.com/en/starter/installing.html>

24. MySQL Documentation. *MySQL 8.0 Reference Manual*. Oracle Corporation.

- URL:

<https://dev.mysql.com/doc>

25. Карпенко, О. С. (2022). *Розробка клієнт-серверного додатку засобами Node.js та Express.js*. // Вісник Київського національного університету імені Тараса Шевченка. Серія: Прикладна інформатика.

26. Гриценко, М. В. (2021). *Особливості побудови веб-інтерфейсів з адаптивним дизайном.* // Наукові праці ОНАХТ, № 2(70), с. 45–51.
27. Khare, R., & Sharma, R. (2023). *Efficient Health Monitoring Web App using MERN stack with Node.js and MongoDB.* // International Journal of Advanced Research in Computer Science, Vol. 14(1), pp. 56–63.
28. Smashing Magazine. (2023). *Designing For Mobile: Best Practices For Mobile-First Design.* – URL:
<https://www.smashingmagazine.com/2023/01/mobile-design-principles>
29. Гончаренко І. І. (2020). *Сучасні технології веб-програмування: практикум.* – К.: КПІ ім. Ігоря Сікорського, 108 с.
30. Postman API Network. *REST API design best practices.* – URL:
<https://learning.postman.com/docs>