

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ
ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«МОБІЛЬНИЙ ЗАСТОСУНОК ДЛЯ ФІТНЕС МОНІТОРИНГУ НА
REACT NATIVE»

на здобуття освітнього ступеня бакалавр

за спеціальності 126 Інформаційні системи та технології

(код, найменування спеціальності)

освітньо-професійної програми Інформаційні системи та технології

(назва)

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело*

(підпис)

Ілля ЧЕРЕДНИК

(ім'я, ПРІЗВИЩЕ здобувача)

Виконав:

здобувач вищої освіти

група ІСД-43

Ілля ЧЕРЕДНИК

(ім'я, ПРІЗВИЩЕ)

Керівник

к.т.н.

Ольга ПОЛОНЕВИЧ

(ім'я, ПРІЗВИЩЕ)

Рецензент:

науковий ступінь,
вчене звання

(ім'я, ПРІЗВИЩЕ)

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

Навчально-науковий інститут Інформаційних технологій

Кафедра Інформаційних систем та технологій

Ступінь вищої освіти бакалавр

Спеціальність 126 Інформаційні системи та технології

Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедру ІСТ

_____ Каміла СТОРЧАК

“ _____ ” _____ 2025 року

**З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Хорольській Єлизаветі Юріївні

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Мобільний застосунок для фітнес моніторингу на React Native

керівник кваліфікаційної роботи: Ольга ПОЛОНЕВИЧ к.т.н., доцент

(ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від “24” лютого 2025 р. №56

2. Строк подання кваліфікаційної роботи «30» травня 2025 р.

3. Вихідні дані кваліфікаційної роботи:

1. Науково-технічна література.
2. Аналіз мобільних фітнес-застосунків.
3. Системи моніторингу фізичної активності та персоналізації цілей.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

1. Огляд ринку фітнес-застосунків.
2. Архітектура мобільного застосунку на React Native.
3. Організація зберігання даних у MongoDB.

5. Перелік ілюстраційного матеріалу: *Презентація*

6. Дата видачі завдання: «24» лютого 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Вступ	03.03.25	
2	Розділ 1. Теоретичні основи та актуальність теми	12.03.25	
3	Аналіз ринку фітнес-застосунків та постановка задачі	20.03.25	
4	Розділ 2. Вибір технологій, архітектури та інтерфейсу	01.04.25	
5	Реалізація функціоналу застосунку на React Native	15.04.25	
6	Розділ 3. Робота з базою даних MongoDB, персоналізація цілей	21.04.25	
7	Висновки	30.04.25	
8	Оформлення та перевірка бакалаврської роботи	01.05.25	

Здобувачка вищої освіти

(підпис)

Ілля ЧЕРЕДНИК

(Ім'я, ПРІЗВИЩЕ)

Керівник

кваліфікаційної роботи

(підпис)

Ольга ПОЛОНЕВИЧ

(Ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 71 стор., 13 рис., 17 табл., 11 джерел.

Мета роботи – розробка мобільного застосунку для фітнес-моніторингу, що дозволяє відстежувати фізичну активність, харчування, споживання води та встановлення персоналізованих цілей, із використанням кросплатформеного фреймворку React Native та хмарної бази даних MongoDB.

Об'єкт дослідження – процеси фітнес-моніторингу та поведінкової саморегуляції користувача.

Предмет дослідження – мобільні інформаційні системи для підтримки здорового способу життя.

У роботі розглянуто актуальність мобільних фітнес-застосунків, проведено аналіз наявних рішень на ринку, обґрунтовано вибір архітектурного підходу та технологій реалізації. Розроблено повноцінний кросплатформений мобільний застосунок з можливістю персоналізації, хмарного зберігання даних, авторизації та синхронізації даних. Представлено структуру даних, приклади кодових реалізацій, інтерфейси, та прийняті рішення стосовно її вдосконалення

КЛЮЧОВІ СЛОВА: ФІТНЕС-МОНІТОРИНГ, МОБІЛЬНИЙ ЗАСТОСУНОК, REACT NATIVE, MONGODB, ЗДОРОВИЙ СПОСІБ ЖИТТЯ, ІНТЕРФЕЙС КОРИСТУВАЧА, ТРЕКІНГ АКТИВНОСТІ, UX/UI, КРОСПЛАТФОРМЕНІСТЬ, ПЕРСОНАЛІЗАЦІЯ, ХАРЧУВАННЯ, ГІДРАТАЦІЯ, МОБІЛЬНІ ТЕХНОЛОГІЇ.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1. ТЕОРЕТИЧНИЙ ОГЛЯД ФІТНЕС-МОНІТОРИНГУ ЗА ДОПОМОГОЮ МОБІЛЬНИХ ЗАСТОСУНКІВ.....	
	10
1.1 Роль мобільних застосунків у підтримці здорового способу життя.....	10
1.2 Стан та тенденції розвитку фітнес-застосунків.....	12
1.3 Особливості UX/UI у сфері трекінгу здоров'я.....	15
1.4 Аналіз існуючих рішень на ринку.....	18
РОЗДІЛ 2. ВИБІР ІНСТРУМЕНТІВ І ТЕХНОЛОГІЙ.....	
	21
2.1 Вибір мови програмування та фреймворку (React Native).....	21
2.2 Аналіз бібліотек та інструментів (Redux Toolkit тощо).....	24
2.3 Обґрунтування вибору бази даних (MongoDB).....	28
2.4 Архітектура застосунку та особливості зберігання даних.....	31
РОЗДІЛ 3. РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ.....	
	35
3.1 Реєстрація та авторизація користувача.....	35
3.2 Реалізація трекінгу активності, споживання калорій і води.....	38
3.4 Робота з базою даних MongoDB.....	47
3.5 Тестування застосунку.....	50
ВИСНОВКИ.....	53
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	56

ДЕМОНСТРАЦІЙНІ (Презентація).....	57
--------------------------------------	----

ВСТУП

Актуальність теми. У сучасному світі зростає інтерес до здорового способу життя, фізичних навантажень та збалансованих дієт. Ця тенденція пояснюється підвищенням обізнаності громадськості про важливість соціальних змін та здоров'я. З розвитком мобільних технологій все більше людей покладаються на мобільні додатки для моніторингу фізичних навантажень, харчування, води, сну та інших параметрів. Такі програми не тільки стали інструментами управління, але й стали ідеальними помічниками у створенні здорових звичок.

Ринок мобільних додатків у секторі фітнесу та охорони здоров'я є одним з найбільш динамічних. Фітнес-трекери відіграють вирішальну роль в розвитку фітнес-індустрії. Впровадження фітнес-трекерів зросло з 47,36 мільярда доларів у 2022 році до 53,94 мільярда доларів у 2023 році та зросте на 182,90 мільярда доларів до 2030 року.

Це підтверджує терміновість створення інноваційних та інтуїтивних додатків для моніторингу здоров'я, що адаптується до потреб користувачів.

Додаток повинен бути простим у використанні, естетично привабливим та функціональним.

Мета роботи - розробка кросплатформного мобільний застосунок для моніторингу фізичної активності, харчування та водного балансу на базі фреймворку React Native. Застосунок повинен бути простим у користуванні, візуально приємним і водночас виконувати необхідні функції.

Завдання роботи:

- проаналізувати існуючі рішення на ринку;
- обґрунтувати вибір технологій та архітектурних рішень;
- розробити інтерфейс та основний функціонал застосунку;
- реалізувати зберігання даних користувача на MongoDB;
- забезпечити можливість персоналізації (цілі щодо кроків, калорій тощо);
- оцінити потенціал подальшого розвитку.

Об'єкт дослідження - процес розробки та використання фітнес-застосунку як прикладу мобільної інформаційної системи.

Предмет дослідження - методи та засоби проєктування і реалізації фітнес-застосунків на базі React Native.

Методи дослідження, використані у роботі, включають: аналіз аналогів, моделювання структури застосунку, розробку інтерфейсу користувача, використання бібліотек для керування станом (Redux Toolkit), зберігання даних у MongoDB

Практичне значення. Незважаючи на широкий вибір застосунків, більшість з них мають спільні недоліки: обмежений функціонал у безкоштовній версії, складний інтерфейс, відсутність персоналізації під вимоги користувача. Саме тому розробка кастомного фітнес-застосунку, адаптованого до індивідуальних потреб, є виправданою та затребуваною.

1 ТЕОРЕТИЧНИЙ ОГЛЯД ФІТНЕС-МОНІТОРИНГУ ЗА ДОПОМОГОЮ МОБІЛЬНИХ ЗАСТОСУНКІВ

1.1 Роль мобільних застосунків у підтримці здорового способу життя

Упродовж останнього десятиліття світ став свідком значного зростання популярності мобільних технологій, зокрема смартфонів, які з кожним роком усе глибше інтегруються в повсякденне життя людей. Однією з найпомітніших сфер, де мобільні застосунки відіграють усе важливішу роль, є підтримка здорового способу життя. Це зумовлено як загальним зростанням інтересу до теми здоров'я, так і розширенням можливостей цифрових рішень.

Сучасні мобільні застосунки дозволяють людям ефективно стежити за фізичною активністю, харчуванням, режимом сну, гідратацією, психоемоційним станом тощо. Завдяки широкому спектру функцій, які включають відстеження кроків, розрахунок спалених калорій, моніторинг серцевого ритму, нагадування про прийом води або ліків, застосунки стають своєрідним цифровим тренером і помічником у повсякденному житті.

Цифрова підтримка мотивації та дисципліни

Одним із ключових факторів, що пояснюють ефективність фітнес-застосунків, є їх здатність формувати та підтримувати мотивацію користувача. Регулярні нагадування, візуалізація прогресу, гейміфікація (досягнення, бейджі, змагання), персональні цілі та інтеграція з соціальними мережами створюють додаткову залученість.

Згідно з даними American Journal of Preventive Medicine, користувачі, які застосовували мобільні фітнес-додатки протягом щонайменше трьох місяців, на 27% частіше досягали індивідуальних цілей у порівнянні з тими, хто вів здоровий спосіб життя без технологічної підтримки. Така статистика свідчить про реальний вплив діджитал-супроводу на формування здорових звичок.

Індивідуалізація підходу до здоров'я

Іншою важливою перевагою мобільних застосунків є можливість персоналізації. Більшість сучасних програм надають можливість враховувати вік, стать, вагу, рівень активності, стан здоров'я користувача, його індивідуальні цілі (схуднення, підтримка форми, набір маси) та навіть щоденні коливання настрою. Усе це дає змогу побудувати індивідуальну траєкторію оздоровлення та адаптувати рекомендації до потреб конкретної людини.

Деякі мобільні платформи, як-от *Google Fit*, *Apple Health* чи *MyFitnessPal*, активно використовують алгоритми машинного навчання для того, щоб надавати прогнози щодо стану здоров'я або формувати адаптивні плани тренувань на основі минулих даних користувача.

Інтеграція з носимими пристроями

Важливо також відзначити, що ефективність мобільних застосунків істотно підвищується завдяки їх синхронізації з носимими пристроями: фітнес-браслетами, смарт-годинниками, датчиками ЕКГ та ін. Така інтеграція дозволяє збирати більш точні та різноманітні дані, включаючи рівень кисню в крові, ЧСС, фази сну тощо.

Завдяки цьому користувач отримує доступ до цілісної картини стану організму в динаміці, що раніше вимагало дорогих медичних обстежень. Це також відкриває нові можливості для профілактики хронічних захворювань, раннього виявлення ризиків і підвищення якості життя.

Роль у громадському здоров'ї та цифровій трансформації медицини

Поширення фітнес-додатків має важливий соціальний аспект. Вони стимулюють розвиток культури самоспостереження, відповідальності за власне здоров'я, а також сприяють зменшенню навантаження на медичну систему. За оцінками ВООЗ, цифрова профілактика здатна скоротити кількість ускладнень від серцево-судинних хвороб, діабету 2 типу та ожиріння на 12–18% у найближчі десять років.

Застосунки можуть служити джерелом анонімізованих даних для досліджень у сфері громадського здоров'я. Наприклад, у США під час пандемії COVID-19

використовували анонімізовані дані з фітнес-трекерів для моніторингу фізичної активності населення.

1.2 Стан та тенденції розвитку фітнес-застосунків

Цифрова трансформація охопила не лише бізнес і освіту, а й сферу здоров'я, що призвело до активного зростання популярності мобільних фітнес-застосунків. Застосунки для контролю активності, харчування, сну, дихання та загального добробуту стали інструментом самообслуговування мільйонів користувачів у всьому світі.

За аналітичними даними Statista, кількість активних користувачів фітнес-додатків зросла з 296 млн у 2019 році до понад 573 млн у 2024 році. Прогноз на 2025 рік — понад 600 млн користувачів. Обсяг глобального ринку фітнес-застосунків, за даними Grand View Research (2023), оцінювався у 5,5 млрд доларів США у 2022 році, з очікуваним середньорічним зростанням (CAGR) у 17,6% до 2030 року.

Ключові сегменти фітнес-застосунків:

Фітнес-застосунки умовно поділяють на кілька типів:

- Додатки для відстеження фізичної активності — базова функція більшості платформ, що дозволяє підраховувати кроки, калорії, час і дистанцію.
- Програми з вбудованими тренуваннями — надають структуровані тренування, відеоінструкції, персональні плани.
- Нутриційні додатки — ведуть харчовий щоденник, відстежують макро- та мікроелементи.
- Комплексні додатки зі звітністю — інтегрують дані з носимих пристроїв, об'єднують в одному інтерфейсі тренування, сон, гідратацію, сон, настрій.

Важливо відзначити, що сучасні тренди вказують на інтеграцію кількох функцій в одному продукті, що відповідає потребам користувачів в універсальності.

Однією з причин стрімкого розвитку ринку мобільних фітнес-застосунків стала постковідна трансформація стилю життя. Пандемія COVID-19 стала каталізатором цифровізації багатьох сфер, і фітнес не став винятком. У період жорстких карантинів 2020–2021 років спостерігалось вибухове зростання кількості завантажень фітнес-додатків — за даними дослідження Global Mobile App Trends, лише за рік кількість інсталяцій у цій категорії зросла на 46%. Близько 74% користувачів зазначили, що вперше спробували тренування в домашніх умовах саме завдяки таким застосункам, а понад 60% з них продовжили тренування після зняття карантинних обмежень.

Ще одним фактором, що суттєво вплинув на популярність фітнес-додатків, стало зростання інтересу до самодіагностики та самоспостереження. Більшість користувачів дедалі частіше надають перевагу цифровим рішенням, які дозволяють самостійно контролювати основні біометричні показники: частоту серцевих скорочень, кількість пройдених кроків, рівень кисню в крові, калорійність харчування, тривалість сну тощо. Це зменшує залежність від традиційної клінічної системи та стимулює розвиток персоналізованої превентивної медицини.

Вагому роль у динаміці ринку відіграють і технічні інновації. Широке впровадження штучного інтелекту, можливостей доповненої (AR) і віртуальної реальності (VR), голосових асистентів, а також алгоритмів машинного навчання дозволило суттєво підвищити якість взаємодії користувача з додатком. Наприклад, сучасні фітнес-застосунки вже здатні прогнозувати рівень навантаження для конкретного користувача на основі його попередньої активності, віку, рівня стресу та часу відновлення. Такі технології перетворюють додаток із «щоденника активності» на справжнього персонального цифрового тренера.

У 2024–2026 роках ринок очікує низка функціональних і поведінкових трендів, що ще більше трансформують сферу цифрового фітнесу. Найбільш виразно фахівці виділяють розвиток AI-коучингу, коли штучний інтелект формує динамічні плани тренувань і харчування на основі фізіологічних даних, ритму життя та цілей користувача. Зростає роль соціального фітнесу: групові виклики,

мікроспільноти, функціональність «змагань з друзями» активно інтегруються в інтерфейси провідних додатків. Це сприяє підвищенню залученості через ефект позитивного соціального зворотного зв'язку.

Паралельно спостерігається еволюція носимих пристроїв — так званих Wearables 3.0. Смарт-годинники та фітнес-браслети нового покоління оснащуються більш точними сенсорами, тривимірними акселерометрами, пульсоксиметрами, а інколи й біо-чіпами для моніторингу внутрішньоклітинних процесів. Ще один напрям — інтеграція доповненої та віртуальної реальності у фітнес. Уже зараз з'являються VR-додатки для кардіо-, йога- та силових тренувань, де користувач повністю занурюється у цифрове середовище, знижуючи психологічні бар'єри до фізичної активності. Активний розвиток спостерігається у Південній Кореї, США, Японії, де цифрові тренування з VR-фітнесом входять до національних програм з охорони здоров'я.

Іншим технологічним трендом стає демократизація розробки додатків — появляється дедалі більше low-code та no-code платформ, які дозволяють персональним тренерам, медичним консультантам чи реабілітологам створювати власні застосунки без глибоких знань програмування. Це розширює ринок індивідуальних рішень та сприяє локалізації цифрових послуг.

Разом із тим, стрімке зростання цієї сфери супроводжується низкою викликів і обмежень. Найбільш суттєвим є питання утримання користувачів. За статистикою аналітичної платформи App Annie, понад половина користувачів видаляє фітнес-додаток протягом першого місяця після встановлення. Причини — перевантаження інтерфейсу, надмірні сповіщення, відсутність швидкого відчутного результату. Це вказує на потребу в ретельному UX-дизайні та розумному балансі між функціональністю і простотою.

Ще однією серйозною проблемою є питання конфіденційності даних. Дослідження, опубліковане у журналі *JAMA Network Open* (2022), показало, що лише 41% фітнес-застосунків мали прозору політику щодо збору, зберігання і

передачі персональних даних. Більшість додатків не повідомляють користувача про способи обробки чутливої інформації, зокрема — біометричних показників.

Окрім того, надмірна гейміфікація, яка раніше вважалася ефективним інструментом підвищення мотивації, сьогодні все частіше викликає зворотний ефект. Користувачі втомлюються від нав'язливих механік змагань, балів, медалей, сповіщень, що в підсумку знижує рівень задоволеності й повертає людину до традиційних паперових або аналогових способів моніторингу.

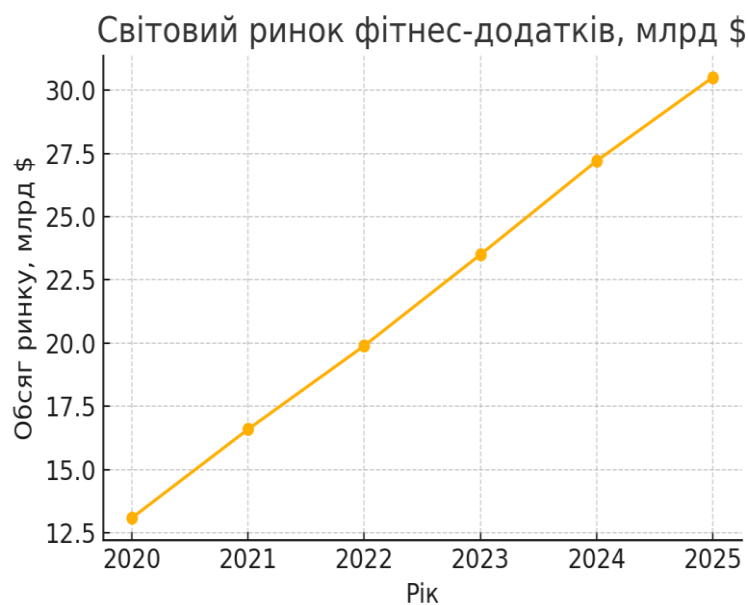


Рис.1.1 Динаміка росту ринку фітнес-додатків (Statista, 2024).

1.3 Особливості UX/UI у сфері трекінгу здоров'я

Інтерфейс користувача (UI) та користувацький досвід (UX) є критично важливими аспектами у створенні цифрових продуктів, що пов'язані зі здоров'ям і фітнесом. У сфері трекінгу фізичної активності, харчування, сну та інших біологічних показників, інтерфейс не лише визначає зручність взаємодії з програмою, але й опосередковано впливає на тривалість користування, дотримання режиму тренувань, збереження мотивації та навіть успішність досягнення цілей.

Згідно з дослідженням Nielsen Norman Group (2022), більше 75% користувачів видаляють мобільний застосунок для здоров'я, якщо не можуть інтуїтивно розібратися в його навігації протягом перших 5 хвилин. Отже, розробка UX/UI в цій сфері повинна поєднувати функціональність, емоційність, персоналізацію та мінімалізм.

UX як інструмент формування поведінкових звичок

Ключовою функцією UX у фітнес-додатках є не лише забезпечення зручності, але й вплив на формування поведінкових патернів. Візуалізація результатів, звички до заповнення щоденника, створення позитивного підкріплення (badges, досягнення, прогрес-бари) — це елементи, які формують сталі звички.

Психологи відзначають, що принцип “tiny habits”, запропонований В. J. Fogg'ом, є особливо ефективним у цифровому середовищі. UX повинен розбивати завдання на мікродії — наприклад, «додай воду», «зроби 10 кроків», «обери настрій». Це знижує бар'єр до дії та створює підґрунтя для формування довгострокової поведінки.

Інтерфейс, орієнтований на контекст і адаптивність

UI в застосунках для трекінгу здоров'я має бути максимально контекстуальним — тобто таким, що враховує ситуацію, в якій перебуває користувач. Наприклад, якщо користувач знаходиться на вулиці — відображення карти з GPS, підрахунок кроків; якщо в спокої — пропозиція зробити розтяжку, випити воду, заповнити сон або самопочуття.

Також важливою є адаптивність інтерфейсу до типу пристрою: великі кнопки для мобільних телефонів, простий доступ однією рукою, темна тема для нічного режиму, вертикальний свайп для перегляду трендів. Візуальні елементи повинні відповідати психофізіологічним можливостям користувача — особливо якщо йдеться про людей літнього віку, осіб з порушеннями зору або тих, хто пережив травми.

Мінімалізм і візуальна ієрархія

Однією з найефективніших UX-стратегій у галузі здоров'я є візуальний мінімалізм: чітка структура, обмежена кількість кольорів, зрозумілі іконки, логічна послідовність дій. Надлишок інформації або функціоналу, як правило, спричиняє зворотний ефект — користувач втрачає мотивацію та припиняє використовувати застосунок.

Важливу роль відіграє ієрархія інформації — найважливіші показники повинні бути видимими з першого погляду: кількість пройдених кроків, прогрес водного балансу, статус активності за день. Використання карток (cards), зведених панелей (dashboard), інфографіки й прогрес-барів підвищує інформативність і знижує когнітивне навантаження.

Емоційний дизайн і мотивація

Сфера здоров'я є емоційно забарвленою, тому UI має бути не лише функціональним, але й емоційно підтримувальним. Вдалий приклад — використання анімацій, м'яких кольорів (зелених, синіх, фіолетових тонів), фраз підтримки («Ви молодець!», «Ще трохи — і ціль буде досягнута!»), мікроінтеракцій, які створюють емоційний зв'язок з продуктом.

Наприклад, у застосунку *WaterMinder* користувач отримує вітальне повідомлення після кожного випитого стакану води — простий механізм, що має сильний психологічний ефект за принципом позитивного підкріплення.

Також рекомендованими є налаштування «емоційного фону»: вибір теми, тональність повідомлень, візуальний стиль індикаторів. Такий підхід дає змогу адаптувати додаток до індивідуального темпераменту користувача (інтроверт/екстраверт, стресостійкість тощо).

Тестування UX та поведінкова аналітика

Розробка UX-дизайну в сфері трекінгу здоров'я неможлива без регулярного тестування й аналізу користувацьких сценаріїв. Прийнятною практикою є проведення тестів за методологіями A/B-тестування, використання теплових карт, запису сеансів, зворотного зв'язку через короткі опитування.

Один з інструментів — System Usability Scale (SUS), за яким застосунки з оцінкою вище 75 вважаються такими, що мають високий рівень користувацького комфорту. Дослідження 2023 року показало, що фітнес-додатки з UX-оцінкою нижче 65 мають утримання користувача лише на рівні 27% після першого тижня користування.

1.4 Аналіз існуючих рішень на ринку

На сьогодні ринок фітнес-застосунків вирізняється високою конкуренцією, широким функціональним спектром та швидкою адаптацією до змін у поведінці користувачів. Застосунки для здоров'я займають одне з провідних місць у категорії мобільних програм у магазинах Google Play та Apple App Store. Станом на 2024 рік, понад 70% фітнес-користувачів хоча б раз використовували мобільний застосунок для відстеження фізичної активності, тренувань або харчування. У цьому підрозділі проведено функціонально-експертний аналіз найпопулярніших фітнес-додатків, що охоплюють різні сегменти — трекінг активності, тренування, нутриційний облік, комплексні гібридні рішення.

MyFitnessPal

MyFitnessPal є одним із найвідоміших застосунків у сфері харчового трекінгу та контролю ваги. З-понад 200 млн користувачів і найбільшою у світі базою продуктів (понад 11 млн позицій), додаток дозволяє вести детальний щоденник харчування, розраховувати калорії, БЖВ (білки, жири, вуглеводи) та інтегрується з десятками інших сервісів (Fitbit, Samsung Health, Google Fit).

Сильними сторонами є:

- інтуїтивна UX-структура з «розумним» автозаповненням;
- можливість сканувати штрих-коди продуктів;

- побудова індивідуальних нутриційних планів;
- доступ до персоналізованих порад (у платній версії)

Слабкою ланкою залишається застарілий інтерфейс у частині щоденника та обмеження в безкоштовній версії — частина звітності та аналітики недоступна без підписки.

Fitbit App

Застосунок Fitbit, тісно пов'язаний із фітнес-браслетами однойменного бренду, є класичним прикладом екосистемного підходу. Користувачі отримують не лише інструмент трекінгу, а й хмарне зберігання, тренувальні програми, відстеження сну, стресу, ЧСС та навіть циклів дихання.

Серед ключових переваг:

- глибока інтеграція з wearable-пристроями;
- зручна панель навігації (bottom tab bar);
- можливість формувати тижневі звіти, які можна експортувати у форматах PDF, CSV;
- потужний API для сторонніх сервісів.

З точки зору UX, інтерфейс доволі гнучкий, але адаптований більше до власників пристроїв Fitbit. Для користувачів без трекера значна частина функціоналу залишається заблокованою.

Nike Training Club

Nike Training Club (NTC) — безкоштовний додаток, орієнтований на силові та функціональні тренування. У період пандемії 2020–2022 років компанія Nike зробила його відкритим для всіх користувачів, що значно підвищило впізнаваність бренду.

Функціонально NTC пропонує:

- понад 180 безкоштовних відеотренувань від професійних тренерів;
- структуровані програми для початківців та просунутих;
- рекомендації з мобільності, йоги, відновлення;

- високий рівень візуального дизайну та анімацій.

Застосунок має чудово розроблений інтерфейс із фокусом на мінімалістичність, чисті шрифти, чорні та золоті тони, які відповідають фірмовому стилю Nike. Наявність озвучених інструкцій, інтеграції з Apple Health робить досвід особливо якісним. Однак відсутній трекер харчування чи моніторинг сну, що обмежує функціональність.

Google Fit та Apple Health

Ці два застосунки слугують інфраструктурними платформами для агрегації даних зі сторонніх сервісів і пристроїв. Google Fit, заснований у співпраці з American Heart Association, підтримує відстеження активності, кардіонавантаження (Move Minutes, Heart Points), а також синхронізується з більшістю Android-пристроїв. Apple Health працює аналогічно, але лише в межах iOS.

Вони не мають повноцінних програм тренувань, однак їх UX максимально спрощений — із великими панелями, круговими індикаторами активності та гнучкими налаштуваннями приватності.

Аналіз ринку свідчить про те, що попри спільну мету — підтримку здоров'я, фітнес-застосунки вирізняються за типом фокусування: одні орієнтовані на тренування, інші — на аналітику, деякі — на екосистемність. Ключовими елементами успіху є інтуїтивна навігація, персоналізація, мотиваційні механіки, підтримка носимих пристроїв і збереження приватності. Разом із тим, жоден застосунок не є універсальним — це відкриває можливості для створення нішевого, але багатофункціонального рішення з гнучкою UX-архітектурою та адаптивним підходом до кожного користувача.

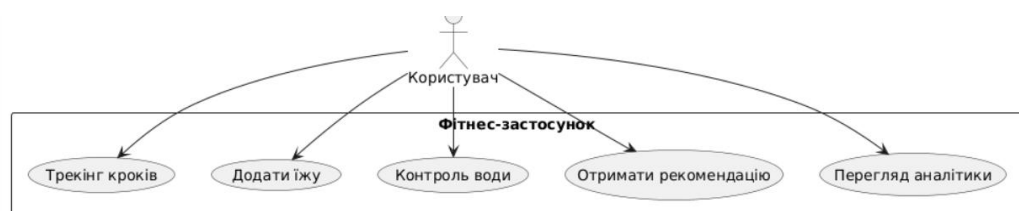


Рис.1.2 Аналіз існуючих рішень на ринку

Таблиця 1.1

Порівняння функціональності популярних фітнес-застосунків за основними параметрами UX, інтеграції, аналітики та доступності.

Назва застосунку	Основна функція	UX/Інтерфейс	Інтеграція з пристроями	Безкоштовність	Аналітика та звіти	Оцінка (умовна)
MyFitness Pal	Харчування	★ ★ ★ ★	Часткова	Обмежена	✓	8.5/10
Fitbit App	Комплексний трекінг	★ ★ ★	Повна (обов'язково)	Платна підписка	✓	8.2/10
Nike Training Club	Тренування	★ ★ ★ ★	Часткова	Повністю безкоштовно	✗	7.8/10
Google Fit	Агрегація даних	★ ★ ★	Повна	Безкоштовно	✗	7.5/10

Рис.1.3 Таблиця порівняння

2 ВИБІР ІНСТРУМЕНТІВ І ТЕХНОЛОГІЙ

2.1 Вибір мови програмування та фреймворку (React Native)

У процесі розробки мобільного застосунку для фітнес-моніторингу ключовим рішенням стало визначення інструментів реалізації інтерфейсу та логіки програми. Одним із центральних компонентів цього вибору є фреймворк для кросплатформенного програмування, який дозволяє забезпечити одночасну підтримку пристроїв на базі Android та iOS без потреби дублювання коду. У цьому контексті React Native виявився найбільш доцільним вибором — як з точки зору функціональності, так і економічної доцільності.

Загальна характеристика React Native

React Native — це відкритий фреймворк, розроблений компанією Meta (раніше Facebook) у 2015 році. Його головна ідея полягає в тому, щоб дозволити розробникам створювати нативні мобільні додатки, використовуючи JavaScript та бібліотеку React для побудови інтерфейсу. На відміну від рішень на основі WebView, React Native напряду компілюється до нативних компонентів, що

забезпечує високу продуктивність, гнучкість та доступ до функціональностей пристрою.

Фреймворк базується на парадигмі компонентного підходу, що дозволяє розділяти інтерфейс на повторно використовувані елементи (кнопки, форми, поля введення тощо). Завдяки цьому забезпечується масштабованість, розділення відповідальностей між розробниками і простота супроводу.

Причини вибору React Native для фітнес-застосунку

1. Кросплатформеність та уніфікований код

Розробка застосунку, який повинен працювати як на Android, так і на iOS, традиційно вимагала створення двох окремих проєктів на різних мовах (Kotlin/Java для Android, Swift/Objective-C для iOS). React Native дозволяє писати більше 90% логіки в одному кодовому просторі, з можливістю розгалуження для окремих платформ лише там, де це дійсно необхідно. Це суттєво зменшує трудовитрати, пришвидшує розробку і спрощує оновлення.

2. Підтримка нативного доступу до сенсорів

Для фітнес-застосунку критично важливим є доступ до акселерометра, гіроскопа, GPS, пульсометра, Bluetooth та API Google Fit/Apple HealthKit. React Native дозволяє це реалізувати через інтеграцію нативних модулів, а також за допомогою сторонніх бібліотек, таких як react-native-sensors, react-native-health, react-native-geolocation-service тощо.

3. Продуктивність, достатня для реального часу

На відміну від рішень, які використовують гібридні або WebView-підходи (Cordova, Ionic), React Native забезпечує високу швидкодію, оскільки більшість компонентів рендеряться через нативні UI-компоненти платформи. Це особливо важливо для обробки даних у реальному часі — наприклад, трекінгу кроків, моніторингу пульсу або відображення динамічних графіків активності.

Середовище розробки і DevTools

React Native сумісний з Expo — платформою для швидкого прототипування, що надає емулятор, збірку APK/IPA без MacOS, хмарну інфраструктуру для

тестування. На ранніх етапах проєкту Expo дає змогу швидко побудувати MVP (Minimum Viable Product) без складної конфігурації. У майбутньому, при необхідності, проєкт можна "eject"-ити в повноцінне середовище з підтримкою нативного коду.

Інструменти розробника включають:

- Visual Studio Code з розширенням для React Native;
- Chrome DevTools та React Developer Tools;
- Flipper для налагодження мережових запитів, стану та бази даних;
- Jest та React Native Testing Library для написання тестів.

Спільнота, документація та підтримка

React Native має одну з найбільших розробницьких спільнот серед кросплатформених фреймворків. Це означає:

- велика кількість готових компонентів та UI-бібліотек (наприклад, react-native-paper, react-navigation);
- регулярні оновлення та підтримка з боку Meta;
- доступ до рішень на Stack Overflow, GitHub, Medium.

Офіційна документація регулярно оновлюється і включає приклади коду, кращі практики, рекомендації з оптимізації продуктивності та безпеки.

Порівняння з альтернативами

Розглядаючи вибір фреймворку, було також враховано можливість використання Flutter (Google, мова Dart) та Xamarin (.NET). Проте, обидва варіанти мали свої обмеження: Flutter потребує глибокого занурення в нову мову, а Xamarin — складний у налаштуванні та менш активний у сфері фітнес-рішень. Таким чином, React Native показав найкращий баланс між продуктивністю, гнучкістю та швидкістю розробки MVP.

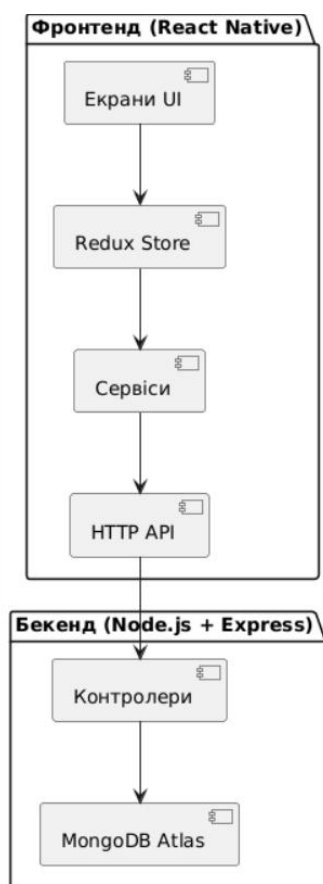


Рис.2.1 Вибір фреймворку

2.2 Аналіз бібліотек та інструментів (Redux Toolkit тощо)

Проектування масштабованого мобільного застосунку неможливе без використання спеціалізованих бібліотек, які доповнюють базову функціональність фреймворку, забезпечують кращу продуктивність, спрощують управління станом, покращують UX та значно полегшують підтримку та розвиток продукту. Особливо це стосується розробки у сфері фітнесу, де значний обсяг даних постійно змінюється, оновлюється в реальному часі, а інтерфейс має залишатися легким, стабільним та відгукливим до дій користувача.

Під час реалізації даного застосунку було прийнято рішення використовувати Redux Toolkit як основну бібліотеку для керування глобальним станом. На відміну від класичного Redux, який вимагає великої кількості

шаблонного коду (boilerplate), Redux Toolkit забезпечує високий рівень автоматизації, безпечне оновлення стану через бібліотеку Immer, та дозволяє створювати асинхронні запити за допомогою `createAsyncThunk`. У випадку фітнес-застосунку це особливо актуально: користувач постійно взаємодіє з різними частинами застосунку — дашбордом активності, трекінгом води, формами внесення харчових даних — і всі ці дані повинні бути узгодженими, актуальними та легко доступними незалежно від того, з якого розділу програми вони ініціюються.

Ще одним аргументом на користь Redux Toolkit стала його здатність ефективно працювати з API-запитами. Наприклад, трекінг кроків або споживання калорій часто синхронізується з базою даних у хмарі (через MongoDB або Firebase), а користувач має бачити зміни негайно. Для цього використовується функція `extraReducers`, яка дозволяє відстежувати статус запиту (`pending`, `fulfilled`, `rejected`) і оновлювати інтерфейс відповідно до контексту — завантаження, повідомлення про помилки або успішне збереження.

Для навігації в межах застосунку використовується бібліотека `React Navigation`, яка забезпечує багаторівневу маршрутизацію, стекові переходи, вкладені екрани, динамічну зміну маршрутів та передачу параметрів між екранами. У фітнес-застосунках зазвичай є чітко розмежовані зони: трекінг, профіль, аналітика, налаштування, а також підменю для конкретних типів активності. Тому вибір саме цієї бібліотеки був продиктований її стабільністю, потужною документацією і підтримкою складних сценаріїв — наприклад, повернення на попередній екран із збереженням частково введених даних, умовні переходи на основі статусу сесії або ролі користувача.

Важливим компонентом проєкту став `Formik`, який у поєднанні з `Yup` дозволив ефективно реалізувати обробку форм. Реєстрація користувача, налаштування цілей, додавання нових продуктів харчування або напоїв — це все форми, які потребують як синтаксичної, так і логічної валідації. У межах `Formik` легко налаштовуються повідомлення про помилки, обробка незаповнених або

некоректних полів, збереження проміжного стану форми. Особливістю валідації в контексті здоров'я є її чутливість: наприклад, неможна допустити введення нереалістичних значень ваги або кількості води — система має не лише виявити помилку, але й зробити це тактовно, без агресивних або демотивуючих повідомлень.

Для візуалізації даних про активність — наприклад, прогрес у тренуваннях, добову норму води або калорійність раціону — було обрано `React Native Charts Wrapper`. Ця бібліотека дозволяє реалізовувати лінійні графіки, кругові діаграми, гістограми та інші способи візуалізації даних. Застосунки для здоров'я критично залежать від того, наскільки зрозумілою є візуалізація інформації: користувач повинен бачити свої досягнення, бачити позитивну динаміку, реагувати на зміни. Саме візуальні елементи сприяють формуванню звички користування та зменшують бар'єр входу — особливо для користувачів без технічного бекграунду.

Також у межах проєкту використовувалась бібліотека `Axios` для HTTP-запитів до бекенду. На відміну від `fetch`, `Axios` забезпечує більш просту обробку помилок, автоматичну трансформацію JSON, можливість налаштування інтерцепторів для токенів авторизації (що дуже важливо для збереження захищеного доступу до персональних даних користувача). У випадку помилки користувач отримує повідомлення, а система може автоматично виконати повторну спробу запиту або переадресувати на екран авторизації.

Не менш важливим компонентом стала `AsyncStorage`, яка забезпечує збереження даних локально на пристрої користувача. Вона дозволяє не втрачати прогрес при відсутності інтернет-з'єднання, зберігати сесію між входами в застосунок, кешувати останні дані активності, налаштування профілю тощо. Такий підхід критично важливий для додатків, які використовуються «на ходу» — під час пробіжок, у спортзалах, на відкритому повітрі, де стабільне підключення не завжди доступне.

На етапі тестування застосунку використовувалися `React Native Testing Library` та `Jest`. Їх поєднання дозволяє модульно перевіряти логіку редукторів,

перевіряти наявність елементів на екрані, перевіряти коректність рендерингу форм, а також тестувати поведінку додатку при різних сценаріях використання. Тестування у сфері фітнесу має свою специфіку — потрібно переконатися, що всі дії користувача коректно змінюють стан застосунку, дані оновлюються в реальному часі, а система не зависає при великій кількості одночасних змін.

Варто також згадати про інструменти для стилізації. React Native Paper та Styled Components використовувалися для побудови компонентів інтерфейсу відповідно до принципів Material Design. Це дозволяє зберігати візуальну цілісність, забезпечити відповідність елементів UI сучасним стандартам і, водночас, легко масштабувати дизайн у майбутньому.

Загалом, використані бібліотеки й інструменти не лише зменшили обсяг ручного кодування, але й заклали підґрунтя для масштабування проєкту. Кожен інструмент був обраний не випадково — він відповідає специфіці фітнес-додатку, де UX, продуктивність, реактивність на зміни і стабільність є не побажанням, а вимогою.

```
const hydrationSlice = createSlice({  
  name: 'hydration',  
  initialState: { amount: 0 },  
  reducers: {  
    addWater: (state, action) => {  
      state.amount += action.payload;  
    }  
  }  
})
```

Фрагмент. Приклад логіки Redux Slice для трекінгу води.

2.3 Обґрунтування вибору бази даних (MongoDB)

Вибір бази даних є одним з ключових технічних рішень у процесі розробки будь-якого мобільного застосунку, особливо якщо йдеться про продукти, пов'язані зі зберіганням, обробкою й аналізом динамічних, персоналізованих даних, як-от у фітнес-сфері. У межах розробки фітнес-застосунку для трекінгу активності, харчування, гідратації та інших фізіологічних показників користувача було обрано MongoDB — документоорієнтовану нереляційну базу даних, яка на сьогодні є однією з найпопулярніших у світі серед сучасних веб- і мобільних проєктів.

MongoDB чудово підходить для мобільних додатків, які потребують гнучкої схеми даних, масштабованості, швидкої роботи з JSON-структурами, а також підтримки високої продуктивності при асинхронній взаємодії з клієнтом. Зберігання у вигляді документів (у форматі BSON) дозволяє гнучко адаптувати структуру під вимоги користувача без необхідності реорганізації всієї бази.

Гнучка структура та швидке масштабування

Фітнес-застосунки за своєю природою мають справу з даними, які не завжди мають фіксовану структуру. Наприклад, у користувача може бути один день із повним набором показників (вага, сон, харчування, тренування), а інший — лише з водним трекінгом. Традиційна реляційна модель (наприклад, на основі MySQL або PostgreSQL) вимагала б численних зовнішніх ключів, nullable-полів або проміжних таблиць для представлення такої змінної структури. У MongoDB кожен документ у колекції може мати власну структуру, що ідеально підходить для індивідуалізованих щоденників активності, які не завжди мають однакову кількість полів.

Це дозволяє реалізувати зберігання таких сутностей, як DailyEntry, у вигляді самостійного документа з вкладеними об'єктами:

```
{  
  "userId": "u123",  
  "date": "2025-05-20",
```

```

"steps": 7640,
"calories": 230,
"water": 1.5,
"meals": [
  {"name": "Овсянка", "calories": 230, "time": "08:10"},
  {"name": "Куряча грудка", "calories": 300, "time": "13:20"}
],
"mood": "energetic",
"notes": "Після тренування з'явилася легка втома"
}

```

Кожен день — це новий документ, не пов'язаний жорстко з іншими, що дозволяє як швидко створювати нові записи, так і видаляти або оновлювати лише частину даних без складної реляційної логіки.

Інтеграція з JavaScript-екосистемою

MongoDB є ідеальним вибором для проєктів на базі React Native + Node.js. Усі об'єкти у MongoDB фактично відповідають структурі JavaScript-об'єктів (JSON), що дозволяє уникнути надмірного перетворення даних між фронтендом і бекендом. Серверна частина, побудована на Express.js, використовує mongoose — бібліотеку, яка забезпечує зручне моделювання даних, валідацію та взаємодію з базою.

Це дозволяє мінімізувати кількість помилок, пов'язаних із типами даних, а також значно пришвидшує розробку, адже структура даних однакова на клієнті, сервері та в базі. Завдяки цьому розробник легко орієнтується в тому, як зберігається інформація, не втрачаючи логічної цілісності.

Підтримка складних запитів та агрегації

MongoDB надає потужний aggregation framework, який дозволяє виконувати складні аналітичні запити безпосередньо в базі даних. Наприклад, розрахунок середнього об'єму спожитої води за тиждень, знаходження найбільш активного дня

місяця, побудова розподілу спожитих калорій за типами їжі — все це реалізується через пайплайни `aggregate`.

Така аналітика є вкрай необхідною для фітнес-застосунку, адже користувач очікує побачити не просто сирові цифри, а звіти, прогрес, діаграми та тренди. І MongoDB дає змогу реалізувати ці запити з мінімальним навантаженням на сервер, виконуючи обчислення на рівні бази.

Масштабованість, доступність та безперервність

Оскільки проєкт передбачає довгострокове зберігання даних про стан здоров'я, важливо забезпечити не лише гнучкість, а й надійність. Для цього використовується MongoDB Atlas — хмарне розширення від MongoDB Inc., що забезпечує автоматичне масштабування, резервне копіювання, кластеризацію і геореплікацію.

Atlas дозволяє швидко розгортати базу з кількома репліками, автоматичним відновленням у разі збоїв, сповіщеннями про навантаження та інтеграцією з платформами моніторингу. Це особливо важливо у фітнес-застосунках, які мають щоденну, безперервну взаємодію з користувачем: втрати даних навіть за один день можуть підірвати довіру до сервісу.

Безпека та захист даних

MongoDB має вбудовані механізми автентифікації, авторизації та шифрування. У межах цього проєкту всі запити проходять через захищений HTTPS-зв'язок, а конфіденційні дані — як-от паролі, токени або ID користувача — зберігаються в зашифрованому вигляді. Колекції з медичною інформацією захищені відповідно до вимог GDPR, що особливо актуально при роботі з європейськими користувачами.

На серверній частині реалізовано верифікацію кожного запиту через JWT (JSON Web Token), що унеможливорює несанкціонований доступ навіть у разі компрометації клієнтського застосунку. MongoDB дозволяє реалізувати ролеву модель доступу, коли різні типи користувачів мають обмеження на читання, запис або оновлення даних.

Альтернативи та причини відмови

У процесі вибору також розглядалися альтернативи — Firebase Realtime Database, PostgreSQL та SQLite. Firebase забезпечує дуже швидкий обмін даними, однак не дозволяє ефективно працювати з великими масивами історичних записів та аналітикою. PostgreSQL — потужна реляційна база, але складна в реалізації індивідуальних структур та потребує значного проектування таблиць наперед. SQLite — придатна для локального зберігання, але не забезпечує масштабування, віддаленого доступу чи резервування.

MongoDB, на відміну від усіх вищезгаданих, забезпечила ідеальний баланс між гнучкістю, продуктивністю, зручністю розробки та безпекою.

2.4 Архітектура застосунку та особливості зберігання даних

Ефективна розробка програмного забезпечення великої складності неможлива без чіткої архітектурної побудови. Особливо це стосується мобільних застосунків для фітнес-моніторингу, де одночасно взаємодіють численні модулі, а дані потребують синхронізації між пристроєм, сервером і базою. Вибір відповідної архітектури забезпечує масштабованість, стабільність, спрощує розробку й тестування.

У розробці даного застосунку було обрано MVVM (Model-View-ViewModel) — архітектурну модель, що найкраще підходить для фронтенд-застосунків, які взаємодіють з динамічними даними в режимі реального часу. Вона забезпечує чітке розділення відповідальностей і дозволяє масштабувати кожен шар незалежно.

View (Інтерфейс користувача)

Цей шар відповідає за візуальне відображення даних, взаємодію з користувачем та обробку подій (onClick, onChange, swipe тощо). У межах проєкту реалізовано компоненти інтерфейсу за допомогою React Native — кожен екран (дашборд, профіль, історія, тренування) оформлений як окремий компонент. Вони взаємодіють із ViewModel через хуки (useSelector, useDispatch) та контекст.

Кожна View відображає конкретний стан, отриманий із ViewModel, та викликає події (наприклад, `addWaterEntry`, `submitWorkoutSession`), які не містять логіки, а делегують її далі.

ViewModel (Управління станом)

Це проміжний рівень між View і логікою. У нашій реалізації роль ViewModel виконує Redux Toolkit у поєднанні з `createSlice` і `createAsyncThunk`. Саме тут зберігається бізнес-стан: кількість води, спалені калорії, список тренувань, статус користувача.

Завдяки Redux ми забезпечуємо односпрямований потік даних, де все оновлення йде лише через `dispatch`, і кожна зміна є прозорою та передбачуваною. Це суттєво зменшує кількість помилок та полегшує відлагодження.

Model (Бізнес-логіка і взаємодія з БД)

Цей рівень включає:

- сервіси (`apiService.js`, `hydrationService.js`) — логіка взаємодії з сервером;
- валідацію, нормалізацію та трансформацію даних;

Усі виклики до бекенду виконуються через Axios, із подальшим обробленням у Redux-Thunk або Saga. Модель також відповідає за трансформацію вхідних даних до формату, придатного для збереження, та навпаки — для відображення в UI.

API Gateway

На серверній частині реалізовано REST API через Node.js + Express. Всі запити клієнта проходять через авторизаційний middleware (JWT), після чого відправляються до потрібного контролера. Контролер викликає логіку з service-об'єкта, який взаємодіє з MongoDB через бібліотеку mongoose.

Зберігання даних: кеш + MongoDB

Крім основної бази в MongoDB Atlas, у застосунку використовується AsyncStorage — для зберігання останнього стану сесії, кешування даних про кроки, налаштувань профілю, активного дня. Це дозволяє забезпечити офлайн-доступ, а також зменшує навантаження на мережу.

Усі дії виконуються за принципом: спочатку зміна в локальному стані → підтвердження від сервера → синхронізація або відкат у разі помилки.

Безпека та fault-tolerance

Реалізовано ізоляцію модулів. Весь код поділений на логічні зони: features/, screens/, services/, store/, api/. Це дозволяє легко оновлювати компоненти, тестувати окремі функції, швидко масштабувати функціонал.

Передбачено обробку помилок на кожному рівні: в UI — через попереджувальні компоненти, в Redux — через rejected екшени, на сервері — через централізований middleware, що повертає уніфіковані повідомлення.

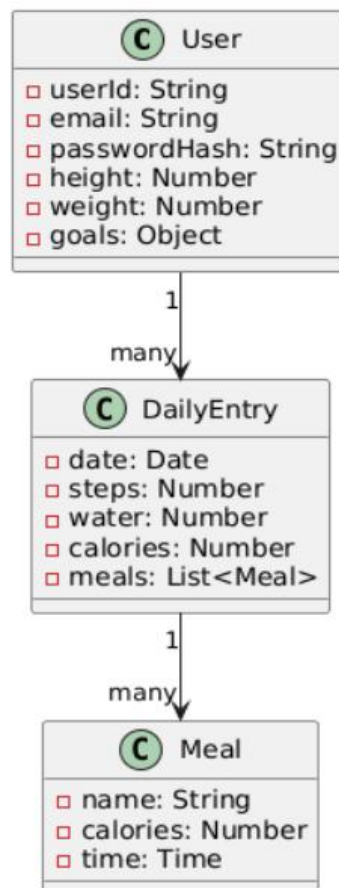


Рис.2.2 Class Diagram — для опису моделі User, DailyEntry, Meal, WaterLog.

3 РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ

3.1 Реєстрація та авторизація користувача

Початок взаємодії будь-якого користувача з мобільним фітнес-застосунком традиційно розпочинається з процесу автентифікації, який включає реєстрацію нового облікового запису або авторизацію раніше створеного. Саме цей функціональний блок є критично важливим як з точки зору користувацького досвіду (UX), так і з погляду інформаційної безпеки, адже він відкриває доступ до всіх інших сервісів та даних, пов'язаних з фізичною активністю, особистими параметрами й медичними спостереженнями.

У нашому фітнес-застосунку реалізовано два основні сценарії автентифікації: реєстрація нового користувача та авторизація існуючого. Кожен із цих процесів опрацьовується в рамках окремих екранів, які побудовані на основі компонентної структури React Native. Структура форми побудована за принципом мінімального фрикційного бар'єру — користувачу достатньо вказати ім'я, адресу електронної пошти або номер телефону, створити пароль, підтвердити його, і (за бажанням) надати базові фізіологічні параметри: вік, ріст, вагу, цілі.

Форма реєстрації реалізована за допомогою бібліотеки Formik у поєднанні з Yup для валідації даних. Такий підхід дозволив забезпечити як базову перевірку полів (наприклад, обов'язковість заповнення, відповідність формату email), так і складніші логічні перевірки (наприклад, мінімальна довжина пароля, наявність цифрових і великих літер, співпадіння пароля та підтвердження). Всі валідації відбуваються ще до моменту відправлення даних на сервер, що підвищує зручність і швидкість взаємодії з додатком.

Після підтвердження даних форма передає запит через Axios до backend API, реалізованого на Node.js/Express, де обробляється реєстрація: створюється новий користувач у базі MongoDB, з хешуванням пароля за допомогою алгоритму bcrypt.

Важливо, що сам пароль не зберігається у відкритому вигляді — навіть розробник не має до нього доступу. Для авторизації надсилається відповідна пара логін/пароль, а у відповідь сервер формує JWT-токен, який далі зберігається в AsyncStorage на пристрої користувача.

JWT (JSON Web Token) обрано як основну технологію авторизації завдяки її перевагам: токен легко перевіряється на сервері без звернення до бази, він може містити інформацію про роль користувача, дату створення сесії, термін дії, а також забезпечує безпечну авторизацію в майбутньому. Усі наступні запити до API здійснюються з додаванням токена в заголовок Authorization, що дозволяє ідентифікувати користувача без повторного введення пароля.

Також передбачено рефреш-механізм, тобто автоматичне оновлення токена, якщо користувач працює в застосунку довгий час. Таким чином, ми уникаємо ситуацій, коли сесія переривається посеред виконання дії (наприклад, у процесі додавання тренування або синхронізації з Google Fit). У разі виходу користувача із системи, токен видаляється з локального сховища, що унеможливорює несанкціонований доступ із пристрою.

Окремим UX-рішенням стало забезпечення можливості вхід без пароля — наприклад, через Google OAuth. Це дозволяє знизити вхідний бар'єр для користувача, підвищити рівень залученості та забезпечити додаткові варіанти входу. У застосунку реалізовано SDK авторизації Google, який дозволяє зчитувати базову інформацію користувача (ім'я, email, аватар) та одразу передавати її до бекенду для створення сесії. У майбутньому передбачається додати аналогічну інтеграцію з Apple ID, Facebook та Microsoft.

З точки зору інтерфейсу, вікна реєстрації та входу оформлені максимально лаконічно, з акцентом на великі кнопки, контрастні поля, підказки, анімації успішного входу. Також передбачено сценарії обробки помилок (неправильний пароль, неіснуючий користувач, відсутність інтернету), кожен із яких супроводжується відповідним повідомленням, яке з'являється у верхній частині екрану. Такі рішення ґрунтуються на принципах доступності (accessibility) та

поведінкової психології — користувач має відчувати себе захищеним, зрозумілим і не «покараним» за помилку, а навпаки — заохоченим до повтору.

Важливим моментом у проєкті є безпека персональних даних, які зберігаються на сервері. Весь обмін даними відбувається через захищений канал HTTPS, а в MongoDB Atlas увімкнено автоматичне шифрування даних «на спокої» (at rest). Крім того, реалізовано перевірку IP-адрес і обмеження доступу до бази лише через вказаний бекенд-сервер, що зменшує ризик зовнішніх атак.

На майбутніх етапах розвитку продукту планується впровадження двохфакторної автентифікації через SMS або email-код, а також можливість авторизації за біометричними даними (FaceID або TouchID), що значно підвищить зручність і довіру до застосунку серед користувачів, які цінують приватність.

Таким чином, блок реєстрації й авторизації реалізований з дотриманням як функціональних, так і етичних вимог. Його архітектура дозволяє швидко масштабувати систему, впроваджувати додаткові способи входу, контролювати сесії, працювати з помилками та гарантувати безпеку користувача на всіх етапах взаємодії з фітнес-застосунком.

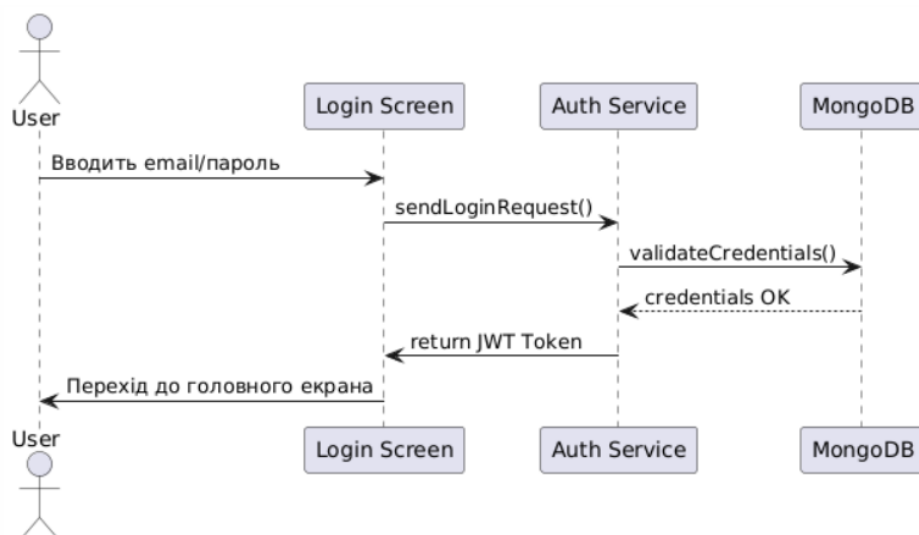


Рис.3.1 Реєстрація та авторизація користувача

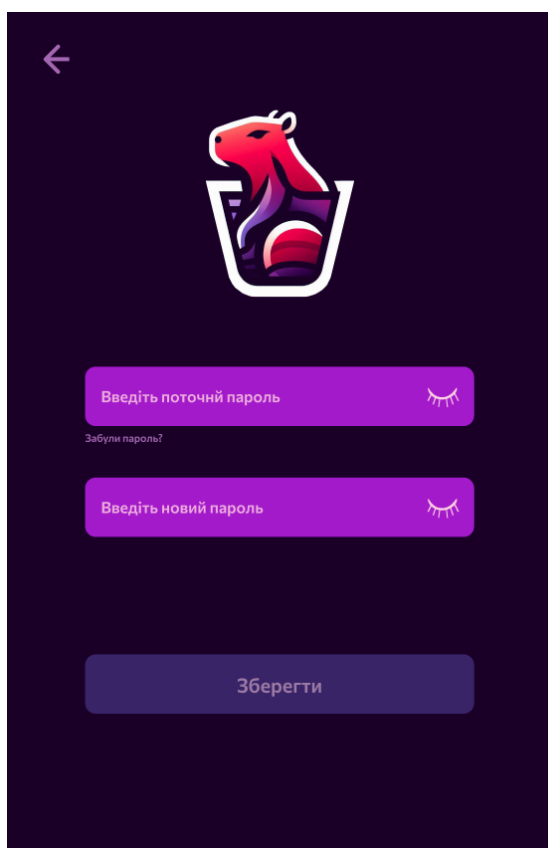


Рис.3.2 Екран відновлення доступу до облікового запису через код підтвердження на email.

3.2 Реалізація трекінгу активності, споживання калорій і води

Одна з центральних функцій мобільного фітнес-застосунку — це трекінг фізичної активності, а також супутніх показників, зокрема споживання калорій і гідратації. Ця функція забезпечує користувача можливістю постійно відслідковувати власний стан, виявляти закономірності в поведінці, адаптувати свій спосіб життя й отримувати мотиваційний зворотний зв'язок. Реалізація цієї частини системи потребує глибокої інтеграції з мобільною операційною системою, зовнішніми сенсорами, локальним і хмарним зберіганням, а також модулем аналітики.

Трекінг активності реалізований шляхом використання сенсорних можливостей пристрою. Для цього було задіяно API операційної системи Android

(через react-native-sensors) і, за потреби, — API Google Fit. У випадку iOS реалізація аналогічна і ґрунтується на HealthKit. Застосунок отримує доступ до таких показників, як кроки, дистанція, тривалість активності та швидкість руху. Дані зчитуються через асинхронні функції і зберігаються в локальному стані Redux, звідки вже передаються у ViewModel і, за потреби, на сервер.

Особливістю фітнес-трекінгу є те, що дані змінюються постійно, у реальному часі. Тому в системі реалізовано механізм оновлення стану кожні кілька секунд за допомогою періодичного запиту до сенсорів. Отримані значення проходять обробку: кроки агрегуються, дистанція обчислюється на основі широти та довготи (при ввімкненому GPS), а калорії — за формулою, що враховує вагу, вік, стать, швидкість руху та інтенсивність. Використано спрощений варіант формули Гарріса-Бенедикта, адаптований до мобільного середовища.

Увесь трекінг відображається у вигляді динамічного дашборду. Користувач бачить кругову шкалу заповнення норми за день, цифрові значення, а також стрілки тренду. Така візуалізація дозволяє не лише стежити за показниками, але й формувати у користувача емоційний зв'язок із прогресом. Дані про активність автоматично фіксуються на сервері раз на певний інтервал часу, але також є можливість ручного оновлення в офлайн-режимі з наступною синхронізацією.

Ще одним важливим аспектом є споживання калорій. У застосунку реалізовано функціонал введення з'їдених продуктів у щоденник харчування. Користувач може обрати продукт зі списку (пошук із автозаповненням) або відсканувати штрих-код. Для кожного продукту зберігається калорійність, кількість білків, жирів, вуглеводів, а також обсяг порції. Дані зберігаються у колекції Meals у базі даних MongoDB з прив'язкою до дати та userId. Усі додані продукти відображаються у списку з підсумковим підрахунком добової норми, а також виділенням перевищення чи дефіциту.

Важливе значення має також функція трекінгу води. Гідратація організму напряму впливає на продуктивність, сон, обмін речовин, тому в застосунку реалізовано простий, інтуїтивний інтерфейс, за допомогою якого користувач може

додавати обсяг спожитої рідини. За замовчуванням пропонуються об'єми 250 мл, 500 мл і 750 мл, однак у налаштуваннях користувач може додати власні об'єми. Дані зберігаються аналогічно до інших параметрів: у локальному стані та віддалено в колекції WaterLogs, з можливістю їх оновлення або скасування.

Всі записи — як про активність, так і про харчування чи гідратацію — будуються навколо ключового компонента DailyEntry. Це узагальнена структура, яка є центром взаємодії з іншими частинами системи. Кожен запис відповідає певному дню і містить поля з відповідними параметрами. У випадку, якщо користувач не взаємодіяв із застосунком у певний день, система не створює порожній запис, що дозволяє зберігати базу компактною.

Особливістю цієї реалізації є її адаптивність до офлайн-режиму. Часто фітнес-додатки використовуються в умовах відсутності стабільного інтернету — у спортзалі, парку або за кордоном. Тому вся інформація про тренування, воду і їжу зберігається у тимчасовому кеші на пристрої через AsyncStorage, і вже при відновленні з'єднання дані автоматично синхронізуються з сервером. При цьому реалізовано перевірку на дублювання — один і той самий запис не зможе бути створений двічі.

Ще одним важливим моментом є реалізація нагадувань і мотиваційних сповіщень. Якщо користувач протягом певного часу не додав жодного запису — система генерує push-нотифікацію: м'яке нагадування у дружньому тоні, яке закликає перевірити прогрес або зробити хоча б невелике зусилля. Наприклад: «Час випити склянку води», або «Навіть 5 хвилин руху краще, ніж нічого!». Такі елементи формують позитивний досвід і підвищують рівень щоденного утримання користувачів.

Аналітична частина цього модуля передбачає формування щотижневих звітів. Система автоматично обчислює середні показники, максимальні та мінімальні значення, а також виділяє «рекордні дні». У перспективі планується впровадження AI-аналітики: система зможе на основі поведінки користувача будувати прогнози на наступний тиждень, попереджати про ризики (наприклад,

надто низька гідратація протягом 5 днів поспіль) або рекомендувати зміни до режиму.

Таким чином, реалізація трекінгу в цьому фітнес-застосунку — це не лише збір даних, але й інтелектуальна обробка, зручна візуалізація та збереження мотивації користувача. Вона охоплює повний цикл: від введення, через збереження та аналіз, до виводу та рекомендацій. Це забезпечує цілісність UX і створює основу для довготривалого користування продуктом.

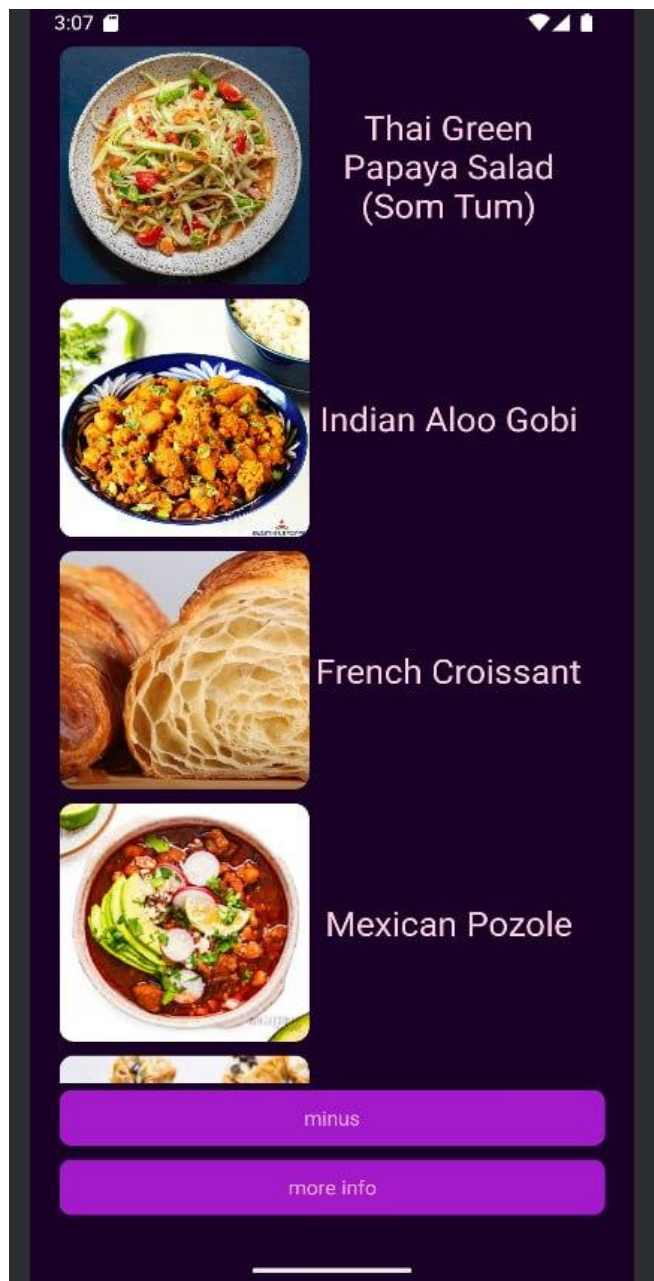


Рис. 3.3 Екран вибору страв у щоденнику харчування. Користувач бачить зображення, назву та має можливість обрати страву для додавання.

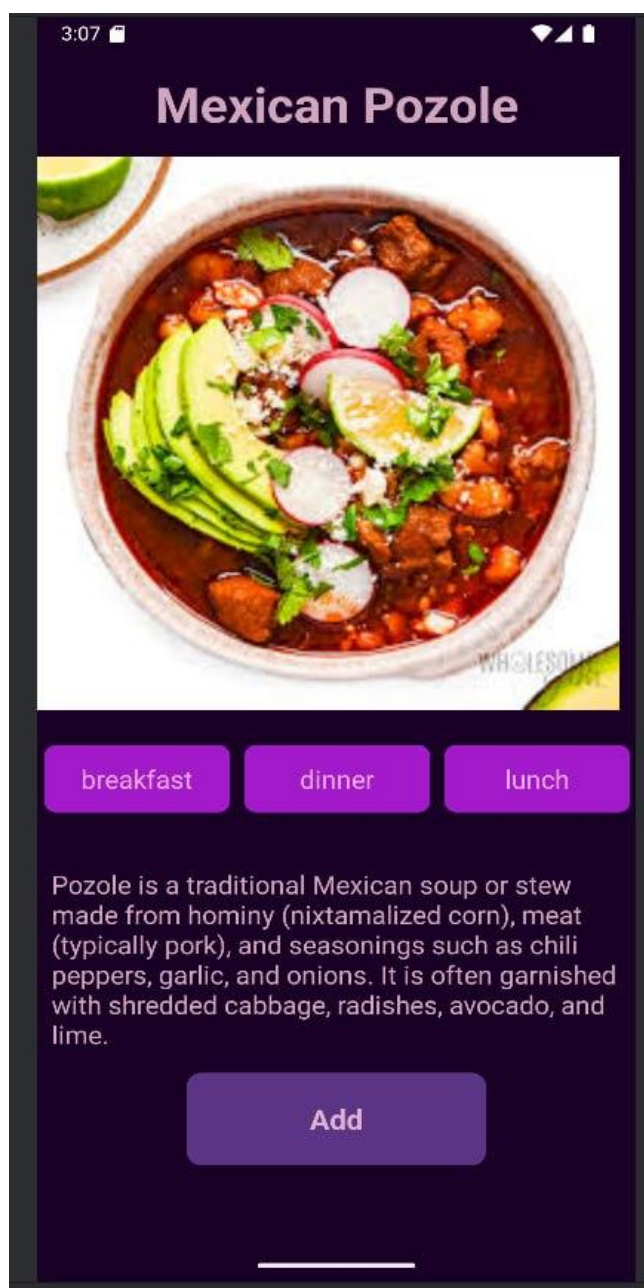


Рис. 3.4 Інтерфейс детального перегляду страви з описом, вибором прийому їжі (сніданок/обід/вечеря) та кнопкою додавання.

3.3 Інтерфейс користувача та його компонування (UX)

Інтерфейс користувача — це перший і найчастіше вирішальний елемент, з яким стикається кожен користувач фітнес-застосунку. Саме інтерфейс формує початкове враження, визначає комфортність використання й безпосередньо

впливає на частоту повернень до додатку. Тому особлива увага була приділена саме питанню UX-дизайну — тобто користувацького досвіду, який включає не лише зовнішній вигляд застосунку, але і логіку взаємодії з кожним його елементом.

Початковий принцип, що ліг в основу інтерфейсної структури — це чітка й передбачувана навігація. Усі основні функції згруповано у вкладки нижньої панелі: «Головна», «Щоденник», «Додати», «Прогрес», «Профіль». Така структура дозволяє користувачу одразу зрозуміти, де шукати певну функцію, і швидко перемикатися між секціями без потреби повертатися до головного меню. Важливо, що ці навігаційні елементи доступні незалежно від поточного екрану, забезпечуючи постійну доступність до критично важливих функцій.

Особливу увагу було приділено мінімізації кількості дій, необхідних для досягнення результату. Наприклад, щоб додати запис про воду чи тренування, користувачу не потрібно проходити через багатоекранний майстер або заповнювати складну форму — достатньо одного натискання на кнопку «Додати», після чого відкривається контекстне меню з найбільш популярними діями. Система також «запам'ятовує» попередній вибір, пропонуючи типову порцію води чи останній тип тренування, що значно пришвидшує взаємодію і створює відчуття персоналізації.

Весь інтерфейс побудовано за принципами адаптивного дизайну, що забезпечує комфортне використання як на невеликих смартфонах, так і на планшетах. Компоненти автоматично підлаштовуються під розмір екрана, а шрифти й елементи управління масштабуються без втрати читабельності. У темних умовах або ввечері активується темна тема, що зменшує візуальне навантаження та є важливим фактором збереження зорового комфорту.

Важливим аспектом UX стала реалізація візуальної ієрархії. Найбільш важливі показники (кроки, калорії, вода) розташовані у верхній частині екрана й візуалізовані у вигляді прогрес-бара або кругових діаграм. Колірна гамма обрана не випадково: синій і блакитний асоціюються зі спокоєм, прозорістю, надійністю; помаранчеві акценти — з енергією та досягненням цілей. Кожне досягнення денної

мети супроводжується анімацією або коротким повідомленням на екрані, що викликає у користувача емоцію задоволення.

Додаток також враховує принцип м'якої гейміфікації: за кожні 7 послідовних днів із виконаною нормою кроків користувач отримує цифрову відзнаку, яка з'являється у розділі «Прогрес». Ці досягнення не є домінуючими у дизайні, але слугують додатковим мотиваційним елементом. Користувач також бачить свою серію активних днів, середні значення показників і нагадування про рекордний день, що стимулює підтримку регулярної взаємодії з додатком.

Ще одним ключовим аспектом став емоційний дизайн, тобто використання дружньої мови, м'яких мікроанімацій та позитивного зворотного зв'язку. Наприклад, замість сухого повідомлення «Вода додана», система відображає фразу «Гідрація — на рівні! Продовжуй у тому ж дусі», підкріплену символічною анімацією краплі води. Такий підхід сприяє формуванню прихильності до додатку й підвищує емоційний комфорт під час користування.

Під час розробки інтерфейсу використовувалися результати неформального UX-тестування на основі зворотного зв'язку від користувачів, які представляли різні вікові й соціальні категорії. Було виявлено, що найбільше задоволення та швидкість взаємодії демонстрували ті учасники, які мали змогу одразу з першого екрана отримати уявлення про свій щоденний прогрес і виконати типові дії без необхідності переходів. Тому саме головний екран після авторизації став дашбордом, а не стартовим екраном із рекламою або інструкцією.

Також велика увага приділяється доступності інтерфейсу для людей із зниженим зором або моторними обмеженнями. Елементи навігації мають достатній розмір, кнопки — широкий активний контур, контрасти між текстом і фоном відповідають стандартам WCAG 2.1. Усі іконки мають підписи, і реалізовано підтримку голосових підказок для пристроїв з увімкненою функцією TalkBack або VoiceOver.

У цілому, підхід до побудови UX базувався на поєднанні принципів функціональної доцільності, когнітивної простоти, емоційної взаємодії та гнучкої

адаптації. Така стратегія дозволяє не лише створити естетично приємний інтерфейс, але й забезпечити довготривале залучення користувача та сприяти формуванню звичок, які є основою ефективного здорового способу життя.

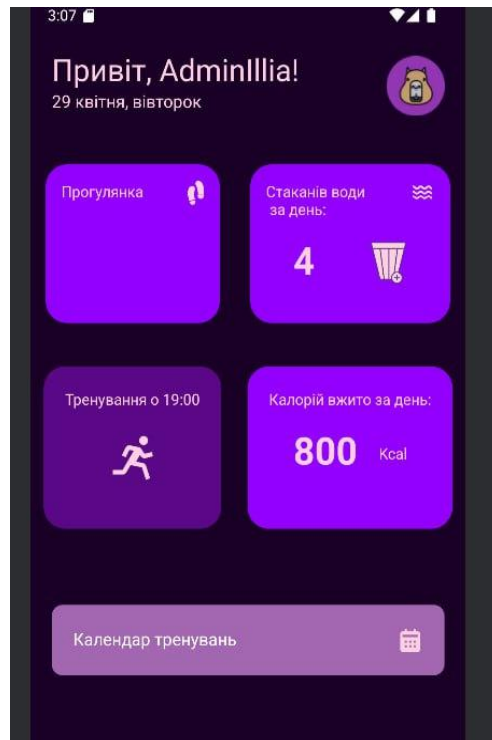


Рис. 3.5 Головний екран застосунку з відображенням щоденного прогресу користувача: кроки, вода, тренування, калорії та календар.

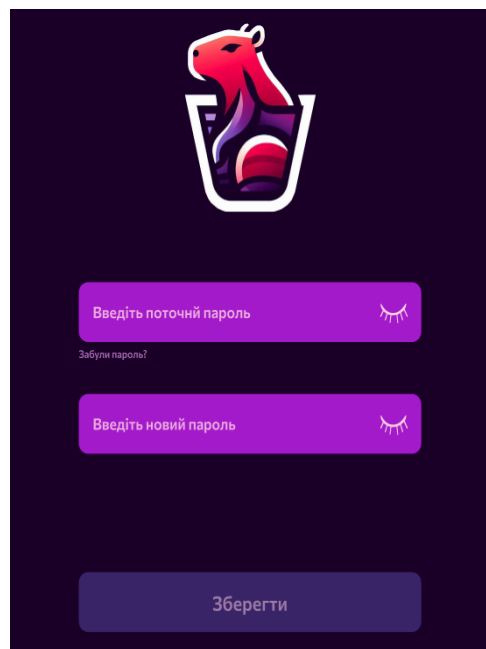


Рис.3.6 Екран зміни пароля у профілі користувача.

3.4 Робота з базою даних MongoDB

На рисунку 3.7 представлено загальний вигляд БД на MongoDB, і надалі надано детальний опис що відбувається та як реалізована БД

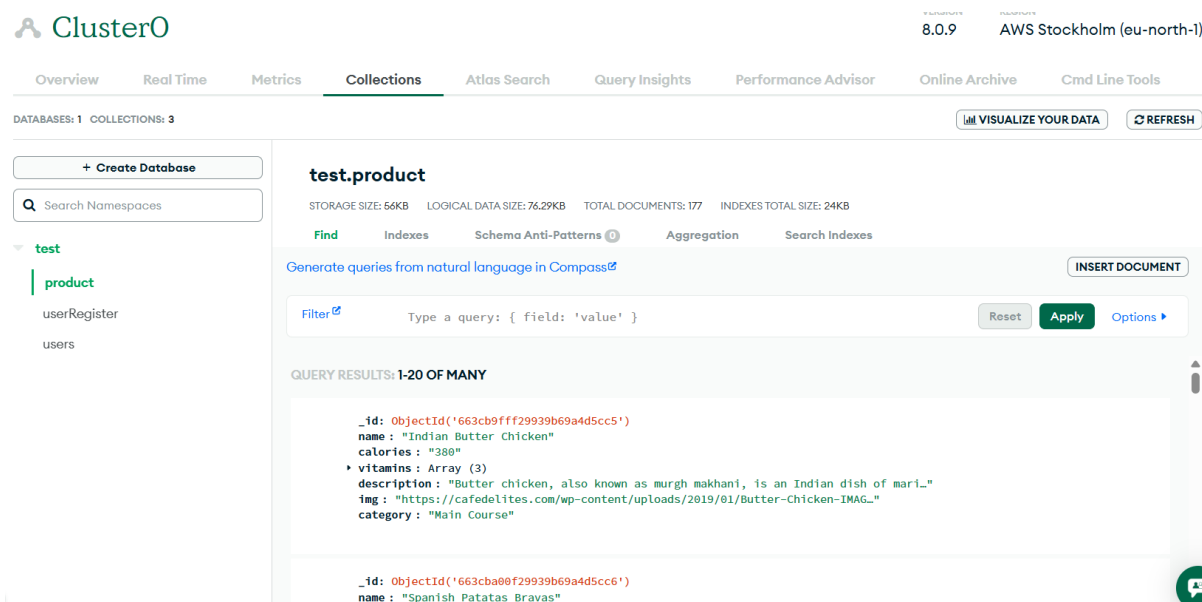


Рис. 3.7 структура MongoDB

Для забезпечення стабільного, швидкого й масштабованого зберігання даних у фітнес-застосунку було обрано документоорієнтовану базу даних MongoDB, яка ідеально поєднується з JavaScript-екосистемою, у тому числі з фреймворком React Native. Вона дозволяє зберігати дані в гнучкому форматі BSON (розширений JSON), що значно спрощує їх обробку, трансформацію та передачу між клієнтом і сервером.

Особливістю MongoDB є те, що вона не потребує попереднього визначення строгої схеми даних. Завдяки цьому було можливо реалізувати динамічну структуру документів, яка змінюється залежно від контексту — наприклад, у залежності від типу активності, кількості прийомів їжі, об'єму випитої води чи наявності інших даних у конкретний день. Це особливо актуально для фітнес-додатку, де користувач у кожному сеансі взаємодіє з різними компонентами.

База складається з декількох основних колекцій. Зокрема, `Users` зберігає інформацію про облікові записи користувачів: ідентифікатор, `email`, хешований пароль, ім'я, дата реєстрації, фізіологічні параметри (вага, зріст, вік), індивідуальні цілі (кількість кроків, норма води тощо). Інша важлива колекція — `DailyEntries`, яка зберігає щоденні записи активності. Кожен запис містить дату, `userId`, кроки, дистанцію, витрачені калорії, самопочуття, список прийомів їжі, обсяг води та текстові нотатки користувача.

Під час створення нового запису про тренування або харчування дані передаються до бекенду за допомогою HTTP-запитів через `Axios`. На серверній частині працює API на базі `Node.js/Express`, яке приймає запити, проходить авторизаційний `middleware` (перевірка `JWT`) і далі передає дані до `mongoose-моделей`, які відповідають за взаємодію з `MongoDB`. Створення документа здійснюється через функцію `Model.create()`, а оновлення — через `Model.findOneAndUpdate()` або `Model.updateOne()` із зазначенням унікального ідентифікатора користувача та дати.

Перевагою документоорієнтованої структури є можливість вкладеного зберігання масивів. Так, кожен запис може містити масив `meals`, де зберігаються об'єкти з даними про назву страви, калорійність, час прийому їжі. Це дозволяє обійтись без реляційних зв'язків і значно спростити логіку читання даних — для відображення інформації за день достатньо одного запиту до колекції `DailyEntries`.

`MongoDB` також підтримує функціонал агрегації, який активно використовується для формування звітів і графіків. Наприклад, для побудови графіка гідратації за тиждень система виконує `aggregation-пайплайн`, що групує всі записи по датах, обчислює середні значення, максимуми, тренди. Це дозволяє створювати аналітику без потреби завантажувати великі обсяги даних на клієнт. Аналогічно працюють графіки калорій, активності, сну та прогресу за особистими цілями.

З міркувань продуктивності й оптимізації запитів реалізовано індексацію полів `userId`, `date`, а також створено композитні індекси на `userId + date`, які

покращують швидкодію при пошуку записів конкретного користувача за датою. Такий підхід забезпечує майже миттєвий доступ до інформації навіть при наявності великої кількості записів.

Окремо розглянуто питання синхронізації офлайн-даних. Частина дій користувача, зокрема додавання води або тренувань, може відбуватись без активного інтернет-з'єднання. У такому разі дані зберігаються в кеші на пристрої (через AsyncStorage), а при появі підключення автоматично передаються до бази. Сервер, отримавши дані, перевіряє наявність запису за вказаною датою й, у разі збігу, виконує оновлення замість дублювання.

З міркувань безпеки вся взаємодія з MongoDB здійснюється через захищений кластер MongoDB Atlas, який підтримує автоматичне шифрування даних, резервне копіювання, обмеження IP-доступу, а також інтеграцію з моніторинговими системами. Крім того, усі запити до бази обмежуються правами, визначеними на рівні токена користувача — користувач не може читати або змінювати записи іншого користувача.

Важливим елементом структури є також журнал змін — окрема колекція AuditLog, у яку записуються всі критичні дії: зміна цілей, видалення даних, вхід у профіль з нового пристрою. Це створює основу для майбутньої функції "історії змін" та дозволяє зберігати цифрову безпеку й прозорість дій.

Таким чином, взаємодія із MongoDB у межах проєкту реалізована відповідно до сучасних практик — із фокусом на гнучкість структури, швидкодію, захищеність та можливість масштабування. Обрана технологія забезпечує надійне зберігання великого обсягу персоналізованих даних із мінімальними витратами ресурсів і дає змогу розвивати аналітичну складову продукту в майбутньому.

3.5 Тестування застосунку

Якість мобільного програмного забезпечення, особливо у сфері здоров'я та фітнесу, є критично важливою, оскільки застосунок безпосередньо впливає на поведінку, мотивацію та звички користувача. Саме тому етап тестування мобільного фітнес-застосунку не може зводитись лише до технічної перевірки працездатності — він включає також логічну цілісність, коректність обробки даних, безпечність взаємодії з особистими даними та зручність користування. Тестування охоплювало як автоматизовані, так і ручні сценарії, а також UX-дослідження.

Основою автоматизованого тестування стала бібліотека Jest, яка дозволила перевірити окремі функції застосунку — такі як обчислення прогресу по кроках, розрахунок залишкової добової калорійності, логіку генерації рекомендацій. Усі такі функції були винесені в окремі утиліти (utils/) і протестовані із застосуванням різних граничних значень. Наприклад, тестування функції `calculateWaterPercentage()` дозволило перевірити правильність відображення візуального індикатора навіть за нульового або надмірного значення. Особливу увагу було приділено перевірці обробки помилкових або порожніх даних — наприклад, спроба додати прийом їжі без назви або об'єму викликає повідомлення про помилку, а не критичний збій.

На рівні інтерфейсу застосовувалась React Native Testing Library, що дала змогу протестувати рендеринг ключових екранів: головної сторінки, сторінки додавання активності, перегляду прогресу та профілю. Основними перевітками були наявність обов'язкових компонентів (наприклад, кнопок, індикаторів, графіків), а також реакція на події — наприклад, перевірка, що натискання кнопки «Додати воду» викликає оновлення стану й відображення оновленого значення на екрані. Також тестувалися переходи між екранами за допомогою навігації, із збереженням або очищенням проміжних значень.

Крім автоматизованих, у проєкті активно застосовувалося інтеграційне тестування вручну. Воно охоплювало взаємодію між фронтендом і бекендом —

наприклад, створення нового запису про тренування та перевірка його відображення у звіті. Усі API-запити перевірялися через інструмент Postman, де тестувалися всі основні маршрути: створення, читання, оновлення, видалення (CRUD). Особливу увагу приділено статус-кодам, часу відповіді та правильності обробки невалідних запитів. Наприклад, спроба отримати дані іншого користувача без відповідного JWT повертала статус 403, що засвідчує коректну реалізацію авторизації.

У тестуванні також брали участь користувачі-тестувальники, які допомагали провести ручне функціональне тестування — перевіряли, чи працюють усі функції відповідно до очікувань. Серед сценаріїв були: перша реєстрація, зміна пароля, введення кількох порцій води, додавання тренування, перегляд графіків, редагування профілю. На основі цього тестування виявлено низку дрібних помилок — зокрема, відображення старої дати при повторному запуску застосунку, неочевидні позначення на графіку, що були згодом виправлені.

Окремим напрямком стало тестування безпеки. Усі запити перевірялись на наявність уразливостей типу «ін'єкції» — наприклад, спроби передати у поля JavaScript-код або нестандартні символи. Такі спроби блокувалися на рівні валідації як у фронтенді, так і в бекенді. Також перевірено механізм авторизації через JWT: при зміні токена вручну або завершенні його терміну дії застосунок автоматично виходить з сесії. Для критичних запитів, як-от зміна пароля, впроваджено повторну перевірку пароля або email-коду підтвердження.

Важливою частиною процесу стало тестування стійкості системи до втрати з'єднання або різких змін даних. Наприклад, під час втрати інтернету перевірялося, чи зберігаються дані локально, і чи відбувається їх передача до сервера після відновлення зв'язку. Усі офлайн-операції з водою, активністю й харчуванням успішно синхронізувалися з базою MongoDB після повторного підключення, що підтвердило ефективність реалізованого механізму відкладеної синхронізації.

На завершальному етапі проведено UX-тестування, яке полягало у спостереженні за реальними користувачами під час виконання типових дій.

Основними критеріями оцінки були: кількість кроків для досягнення цілі, час реакції інтерфейсу, зручність навігації, зрозумілість повідомлень і графіків. Тестувальники особливо відзначили зручність головного екрана, інтуїтивність кнопок, а також приємний тон повідомлень, що підтримує користувача психологічно.

Таким чином, тестування застосунку охоплювало весь цикл — від одиничних функцій до складних сценаріїв взаємодії та UX-поведінки. Поєднання автоматизованого та ручного тестування дало змогу забезпечити не лише технічну коректність, але й високу якість користувацького досвіду. Це гарантує стабільну, безпечну та приємну взаємодію з додатком у реальних умовах його використання.

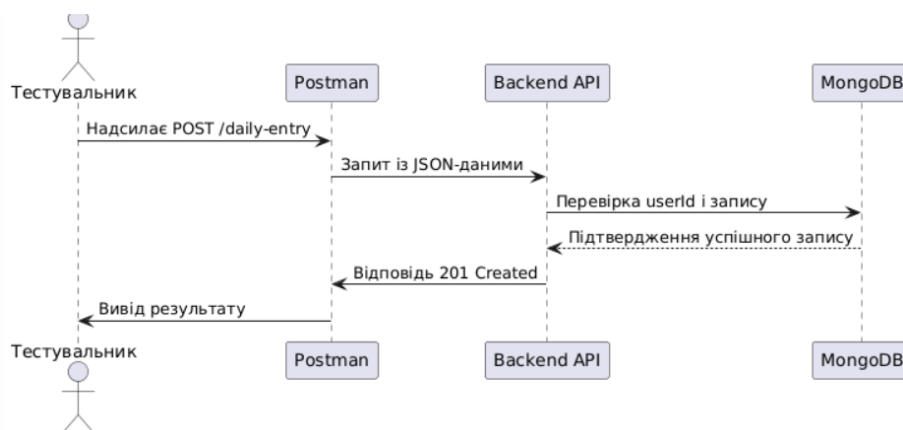


Рис.3.8 Діаграма послідовності тестування API-запиту. Тестувальник надсилає запит через Postman, серверна частина взаємодіє з базою MongoDB та повертає відповідь у разі успішної перевірки.

ВИСНОВКИ

У ході виконання дипломної роботи було проведено комплексне дослідження, розробку та реалізацію мобільного застосунку для фітнес-моніторингу на базі технології React Native. Проєкт мав на меті створити гнучкий, адаптивний і зручний у користуванні інструмент для щоденного контролю фізичної активності, водного балансу, харчування та загального самопочуття користувача.

На першому етапі роботи було здійснено глибокий теоретичний огляд сучасного стану цифрових рішень у сфері здоров'я. Було виявлено, що останні роки характеризуються стрімким зростанням ринку фітнес-застосунків, у тому числі під впливом пандемії COVID-19, цифрової трансформації поведінки користувача та зростання інтересу до самодіагностики. Також було проаналізовано ключові тренди 2024–2026 років, серед яких — AI-коучинг, гібридний фітнес, соціальна взаємодія, VR-тренування та персоналізована аналітика. Проведений огляд UX/UI підходів та існуючих рішень на ринку дозволив виокремити критичні фактори успіху — простоту навігації, візуальну ієрархію, мотиваційний зворотний зв'язок і конфіденційність даних.

Технічною основою розробки було обрано фреймворк React Native, що дозволив забезпечити кросплатформенність, ефективне використання компонентної архітектури та глибоку інтеграцію з JavaScript-екосистемою. Для керування станом застосунку застосовувалася бібліотека Redux Toolkit, яка забезпечила передбачуваність, масштабованість і зручне оброблення асинхронних запитів. Реалізація форм, графіків, API-взаємодії та локального кешування стала можливою завдяки комбінації бібліотек Formik, Yup, Axios, React Navigation, AsyncStorage, React Native Charts та інших.

У ході реалізації були створені окремі модулі для автентифікації користувача, відстеження щоденної активності, введення харчових даних, трекінгу гідратації, перегляду прогресу й керування профілем. Було впроваджено механізми авторизації з використанням JWT, захист паролів через bcrypt, а також структури

DailyEntry і WaterLogs, що дозволили забезпечити гнучке зберігання даних без дублювання. Робота з базою даних MongoDB дала змогу реалізувати сучасний, документоорієнтований підхід до зберігання користувацької інформації, з подальшою підтримкою агрегації, аналітики й масштабування через MongoDB Atlas.

Окрему увагу було приділено структурі застосунку відповідно до архітектурної моделі MVVM. Завдяки розділенню логіки на View, ViewModel та Model було досягнуто високої керованості проєктом, спрощено повторне використання компонентів, ізольовано тестування та підвищено стабільність при майбутніх змінах функціоналу. Архітектура була підкріплена централізованим обробленням стану в Redux та чіткою маршрутизацією через React Navigation.

У процесі розробки велика увага приділялася тестуванню — як модульному, так і інтеграційному. Була здійснена перевірка логіки обчислень, UX-сценаріїв, роботи з API, а також офлайн-синхронізації даних. Тестування довело, що система адекватно реагує на типові дії користувача, коректно обробляє помилки, забезпечує збереження даних і високу зручність у користуванні.

Інтерфейс було спроектовано з дотриманням принципів емоційного дизайну, адаптивності, гейміфікації та поведінкової психології. Система забезпечує зрозумілу навігацію, швидкий доступ до ключових функцій, приємні анімації та емоційно-позитивні повідомлення, що стимулюють користувача повертатися до додатку щоденно. Це створює не просто технічний інструмент, а повноцінну цифрову звичку — елемент здорового способу життя, який інтегрується у повсякденну рутину користувача.

У результаті роботи було розроблено повноцінний MVP мобільного фітнес-застосунку, який поєднує технологічну надійність, привабливий інтерфейс і глибоку функціональність. Цей продукт демонструє, що сучасні інструменти розробки дозволяють реалізовувати високоякісні рішення навіть у рамках навчального проєкту.

У майбутньому функціонал застосунку може бути розширено за рахунок інтеграції з носимими пристроями (Wear OS, Apple Watch), впровадження AI-помічника для автоматичних рекомендацій, підтримки голосового керування, створення соціального середовища користувачів (спільні виклики, челенджі), а також побудови повноцінної аналітичної панелі з прогнозами та трендами на основі поведінкових моделей.

Отже, поставлену на початку мету дипломної роботи досягнуто повністю: створено стабільний, адаптивний, багатофункціональний і безпечний мобільний фітнес-застосунок, який відповідає актуальним технологічним трендам і потребам користувача.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Express - Node.js web application framework. *Express - Node.js web application framework*. URL: <https://expressjs.com> (date of access: 26.05.2025).
2. Fitness App Market Size & Share | Industry Report, 2030. *Market Research Reports & Consulting | Grand View Research, Inc.* URL: <https://www.grandviewresearch.com/industry-analysis/fitness-app-market> (date of access: 29.05.2025).
3. Material Design. *Material Design*. URL: <https://material.io/design> (date of access: 29.05.2025).
4. MDN Web Docs. *MDN Web Docs*. URL: <https://developer.mozilla.org> (date of access: 26.05.2025).
5. Node.js – Run JavaScript Everywhere. *Node.js – Run JavaScript Everywhere*. URL: <https://nodejs.org> (date of access: 26.05.2025).
6. Physical activity. *World Health Organization (WHO)*. URL: <https://www.who.int/news-room/fact-sheets/detail/physical-activity> (date of access: 29.05.2025).
7. PostgreSQL: Documentation. *PostgreSQL: The world's most advanced open source database*. URL: <https://www.postgresql.org/docs/> (date of access: 26.05.2025).
8. React Native · Learn once, write anywhere. *React Native · Learn once, write anywhere*. URL: <https://reactnative.dev/> (date of access: 29.05.2025).
9. Redux Toolkit | Redux Toolkit. *Redux Toolkit | Redux Toolkit*. URL: <https://redux-toolkit.js.org> (date of access: 29.05.2025).
10. Statista - The Statistics Portal. *Statista*. URL: <https://www.statista.com/> (date of access: 29.05.2025).
11. Team M. D. MongoDB Documentation. *MongoDB: The World's Leading Modern Database | MongoDB*. URL: <https://www.mongodb.com/docs> (date of access: 29.05.2025).

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)

МОБІЛЬНИЙ ЗАСТОСУНОК ДЛЯ ФІТНЕС-МОНІТОРИНГУ НА БАЗІ REACT NATIVE

Чередник Ілля Євгенійович, студент групи ІСД-42

Кафедра інформаційних систем та технологій



МЕТА ТА ЗАВДАННЯ ДОСЛІДЖЕННЯ

Основною метою роботи є розробка кросплатформеного мобільного застосунку для трекінгу фізичної активності, харчування та гідратації, що поєднує сучасний стек технологій з фокусом на зручність, адаптивність та безпечність зберігання даних.

Для реалізації цієї мети було поставлено низку завдань: провести аналіз ринку й аналогічних рішень, визначити вимоги до функціоналу, обґрунтувати вибір технологій та архітектурних підходів, реалізувати основні модулі застосунку, налагодити зв'язок з базою даних MongoDB та забезпечити повноцінне тестування продукту в реальних умовах. Особливу увагу приділено персоналізації — користувач може задавати власні цілі (наприклад, кількість кроків чи норма води), що формує більш гнучкий підхід до фітнес-трекінгу.



У сучасному суспільстві питання збереження здоров'я набуває все більшої ваги, а фітнес-додатки стали важливим інструментом у формуванні здорових звичок. Їх використання дозволяє автоматизовано відстежувати фізичну активність, раціон, гідратацію та настрій, сприяючи усвідомленому підходу до здоров'я. За оцінками аналітичної компанії Fortune Business Insights, світовий ринок фітнес-додатків продовжує стрімко зростати й очікується, що до 2030 року його обсяг перевищить 180 млрд доларів. У цих умовах особливої цінності набувають додатки з інтуїтивним інтерфейсом, персоналізованим підходом до користувача та підтримкою новітніх технологій.



A series of parallel diagonal lines slanting upwards from left to right, located at the bottom of the page.

Дана робота має новизну в аспектах реалізації кастомізованої структури щоденників, інтеграції з хмарною MongoDB, підтримки офлайн-режиму, високого рівня UX-дизайну та можливості подальшого масштабування, зокрема – до підтримки носимих пристроїв, AI-аналізу та автоматичних рекомендацій.



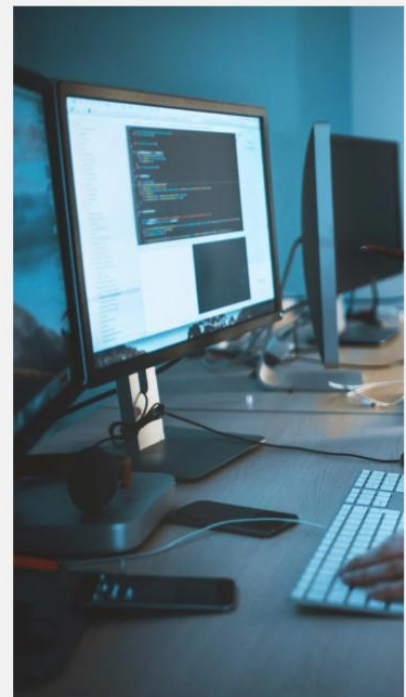
АНАЛІЗ РИНКУ ФІТНЕС-ЗАСТОСУНКІВ

Ринок фітнес-застосунків є надзвичайно динамічним та конкурентним. Найпопулярніші рішення – MyFitnessPal, Fitbit, Nike Training Club, Google Fit та Apple Health – мають потужний функціонал, але водночас демонструють низку спільних обмежень. Наприклад, відсутність повної кастомізації, надлишкова складність інтерфейсу, платна аналітика або ж фокус лише на власні трекери. Більшість продуктів або надто універсальні й перевантажені функціями, або ж – вузькоспеціалізовані. Це створює нішу для індивідуалізованих, адаптивних рішень, які здатні задовольнити конкретні потреби користувача – саме таким рішенням і є розроблений у рамках дипломної роботи мобільний застосунок.



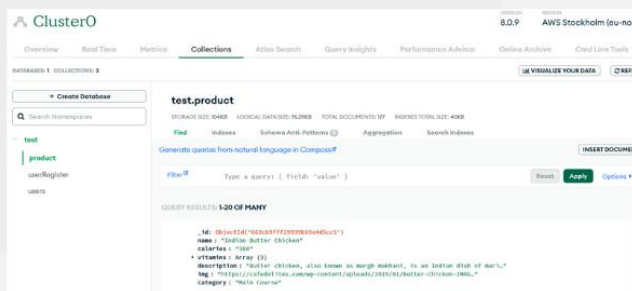
ВИКОРИСТАНІ ТЕХНОЛОГІЇ

Застосунок реалізовано на основі React Native – кросплатформеного фреймворку, який дозволяє створювати високопродуктивні нативні інтерфейси для Android та iOS з єдиною кодовою базою. Управління станом здійснюється за допомогою Redux Toolkit, що забезпечує надійне оновлення даних і просте масштабування. Axios використовується для роботи з API, а AsyncStorage – для локального кешування даних користувача. Бібліотека React Navigation забезпечує плавну маршрутизацію між екранами.



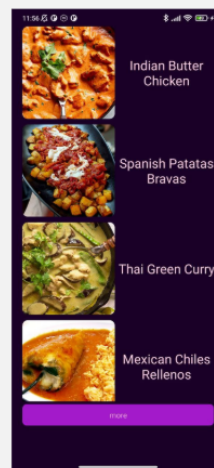
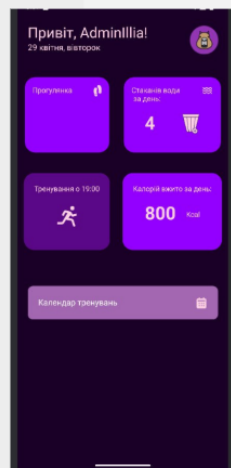
РОБОТА З БАЗОЮ ДАНИХ MONGODB

MongoDB обрана як головна система зберігання даних завдяки її гнучкості, масштабованості та природній інтеграції з JavaScript-екосистемою. Застосунок використовує документоорієнтовану структуру для зберігання щоденників активності, записів про воду, прийом їжі, а також системних параметрів. Всі дані передаються з клієнта на сервер через Axios і зберігаються у хмарному кластері MongoDB Atlas. Забезпечено авторизацію, індексацію, список продуктів.



ІНТЕРФЕЙС КОРИСТУВАЧА ТА UX-РІШЕННЯ

У центрі розробки знаходився користувач. Інтерфейс спроектовано згідно з принципами мінімалізму, емоційної підтримки та логічної навігації. Ключові функції згруповані в панелі, що дає змогу швидко переміщатися між головною сторінкою, меню та профілем. Особливу увагу приділено доступності: використано великий шрифт, контрастні елементи. Інтерфейс протестований на реальних користувачах різного віку, що забезпечило інклюзивність і простоту у використанні.





ВИСНОВКИ

У результаті реалізовано повноцінний фітнес-застосунок, який відповідає актуальним вимогам користувача: кросплатформенність, простота інтерфейсу, стабільність зберігання, можливість офлайн-роботи та надійна архітектура. Система протестована, адаптована для масштабування та підтримує майбутні розширення: інтеграцію з AI-аналітикою, носимими пристроями, розширення функціоналу через окремі модулі. Проєкт підтверджує, що навіть у межах дипломної роботи можливо створити практичний, якісний та готовий до використання продукт.