

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА
Навчально-науковий інститут
«Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці**

ОСВІТНЬО ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»

галузь знань 12 «Інформаційні технології»

спеціальність 122 «Комп'ютерні науки»

Форма навчання: заочна

КВАЛІФІКАЦІЙНИЙ БАКАЛАВРСЬКИЙ ПРОЕКТ

на тему

**РОЗРОБКА WEB-ЗАСТОСУНКУ З ВІДСТЕЖЕННЯ
ХАРЧОВОЇ ПОВЕДІНКИ І ФІЗИЧНОЇ АКТИВНОСТІ**

здобувача *Лебедєв Ростислав Дмитрович*

Науковий керівник:

к.е.н., доцент

_____Денісова О.О.

Проект допущений до захисту

**перед екзаменаційною комісією з
атестації здобувачів вищої освіти**

завідувач кафедри:

к.е.н., доцент.

_____Тішков Б.О.

Київ 2024

Міністерство освіти і науки України
Київський національний економічний університет імені Вадима Гетьмана
Навчально-науковий інститут «Інститут інформаційних технологій в економіці»

Кафедра інформаційних систем в економіці

ОСВІТНЬОПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»

галузь знань 12 «Інформаційні технології»

спеціальність 122 «Комп'ютерні науки»

ПОГОДЖЕНО:

Керівник проектної групи(гарант)
освітньо-професійної програми

_____Іванченко Г.Ф.

“ _____ ” _____ 2024 р.

ЗАТВЕРДЖУЮ:

Завідувач кафедри

_____Тішков Б.О.

“ _____ ” _____ 2024 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

здобувача вищої освіти *Лебедев Ростислав Дмитрович*

очної (денної) форми навчання

на підготовку кваліфікаційного бакалаврського проекту
на тему: «Розробка web-застосунку з відстеження харчової поведінки і фізичної активності»

Тему затверджено наказом ректора Університету від «__» _____ 2024 р. № 332-ст.

Кваліфікаційний бакалаврський проект виконується на матеріалах наукових публікацій, інтернет-ресурсів

План кваліфікаційного бакалаврського проекту

Розділ I Характеристика та аналіз предметної галузі

Розділ II Розробка вимог і моделювання інформаційної системи

Розділ III Проектування та реалізація компонентів системи

Об'єкт дослідження Розробка web-застосунку з відстеження харчової поведінки і фізичної активності

Предмет дослідження теоретичні, методичні та практичні аспекти проектування web-застосування, яке дозволяє відслідковувати раціон харчування та фізичні навантаження.

Мета кваліфікаційного бакалаврського проекту систематизувати та узагальнити сучасний досвід у галузі проектування web-застосувань для відслідковування раціону харчування та фізичних навантажень та розробити власний прототип такого web-застосування.

Конкретні завдання, які студент повинен виконати для досягнення поставленої мети:

У розділі I Описати предметну галузь щодо відслідковування раціону харчування та фізичних навантажень. Навести характеристику об'єкта дослідження. Подати результати аналізу літературних джерел та існуючих варіантів інформаційних систем для відслідковування раціону харчування та фізичних навантажень.

У розділі II Визначити бізнес-вимоги, функціональні та нефункціональні вимоги до проєктованої системи та подати їх у формі діаграм і проєктної специфікації, розроблених з використанням CASE-засобів та мови SysML. Подати постановку та алгоритм розв'язання задачі відслідковування раціону харчування та фізичних навантажень. Розробити моделі системи в специфікації мови UML.

У розділі III Спроектувати базу даних та елементи інформаційного забезпечення web-застосування для відслідковування раціону харчування та фізичних навантажень. Обгрунтувати вибір та розташування комплексу технічних засобів на об'єкті управління. Описати структуру та складові програмного забезпечення ІС. Висвітлити результати реалізації web-застосування.

**Завдання підготував
науковий керівник _____**

Денісова Ольга Олександрівна

“ _____ ” _____ 2024 р.

**Завдання одержав
студент _____**

Лебедєв Ростислав Дмитрович

“ _____ ” _____ 2024 р.

Відгук
про кваліфікаційний бакалаврський проект
здобувача навчально-наукового інституту
«Інститут інформаційних технологій в економіці»
освітньо-професійної програми
«Комп'ютерні науки»

Лебедєв Ростислав Дмитрович

на тему

**«Розробка web-застосунку з відстеження харчової поведінки і
фізичної активності»**

1. **Актуальність теми:** тема проекту є вкрай актуальною у зв'язку з зростаючою популярністю здорового способу життя та необхідністю моніторингу харчування і фізичної активності. Веб-застосування такого типу може значно покращити контроль та управління раціоном харчування та фізичними навантаженнями для широкого кола користувачів.
2. **Позитивні риси кваліфікаційного бакалаврського проекту:** автор продемонстрував високий рівень технічних знань та вмінь у сфері веб-розробки. Проект відзначається чітко визначеними вимогами, продуманою архітектурою, якісним дизайном та функціональністю. Реалізовані функції відповідають потребам користувачів, а інтерфейс є зручним та інтуїтивно зрозумілим.
3. **Наявність самостійних розробок автора:** автор самостійно розробив та впровадив ключові компоненти веб-застосування, включаючи дизайн бази даних, алгоритми обробки даних та функціональні модулі системи.
4. **Цінність теоретичних висновків та практичних рекомендацій:** теоретичні висновки та практичні рекомендації мають високу цінність для подальшого розвитку аналогічних систем.
5. **Наявність недоліків:** деякі аспекти проекту можуть бути покращені, зокрема, детальніший опис безпекових заходів та захисту даних користувачів. Було б корисно також додати функції персоналізації та інтеграції з іншими сервісами.

6. Загальна оцінка кваліфікаційного бакалаврського проекту та його допущення до захисту перед ЕК: кваліфікаційний бакалаврський проект Лебедева Ростислава Дмитровича заслуговує на високу оцінку та рекомендується до захисту перед ЕК. Автор продемонстрував глибоке розуміння предметної області та високий рівень професійної підготовки.

Науковий керівник

(підпис)

доцент, к.е.н. Денісова О.О.

“ ____ ” _____ 20__ р.

Рецензія
на кваліфікаційний бакалаврський проект
Лебедєва Ростислава Дмитровича
тема
«Проектування інформаційної системи інтернет-магазину одягу»

1. **Актуальність теми кваліфікаційного бакалаврського проекту і доцільність його розроблення:** тема проекту "Проектування інформаційної системи інтернет-магазину одягу" є надзвичайно актуальною в сучасному світі, де онлайн-торгівля швидко розвивається та набуває все більшої популярності. Розробка такої системи є доцільною, оскільки вона може значно полегшити процес купівлі-продажу одягу, підвищити ефективність роботи інтернет-магазинів та покращити обслуговування клієнтів.
2. **Якість проведеного дослідження:** автор продемонстрував глибоке розуміння предметної області та високий рівень дослідницьких навичок. Виконано детальний аналіз існуючих інформаційних систем для інтернет-магазинів, проаналізовано вимоги користувачів та ринку. Дослідження проводилось з урахуванням сучасних технологій та тенденцій у сфері електронної комерції. Результати дослідження стали основою для визначення ключових функцій та вимог до розробки інформаційної системи інтернет-магазину одягу.
3. **Позитивні риси кваліфікаційного бакалаврського проекту:** автор виявив високий рівень професійних компетенцій у галузі розробки інформаційних систем. Проект має чітку структуру, детально описані етапи проектування та реалізації системи, включаючи аналіз вимог, створення прототипу, розробку функціональності та дизайну, а також тестування і впровадження. Система має добре структуровану архітектуру та ефективне використання сучасних програмних інструментів і технологій.
4. **Зауваження:** деякі аспекти проекту можуть бути покращені. Наприклад, більш детальний опис алгоритмів і методів, використаних для розробки інформаційної системи, допоміг би краще зрозуміти технічні аспекти

проекту. Також варто приділити більше уваги питанням безпеки та захисту даних клієнтів у контексті розробки системи інтернет-магазину.

- 5. Практична значимість висновків і рекомендацій:** висновки та рекомендації, які випливають з кваліфікаційного бакалаврського проекту, мають високу практичну значимість для сфери електронної комерції. Розроблена інформаційна система може значно покращити ефективність роботи інтернет-магазинів, підвищити рівень обслуговування клієнтів та забезпечити зручність використання. Рекомендації щодо додаткових функціональних можливостей, оптимізації продуктивності та поліпшення користувацького досвіду можуть бути цінними для майбутніх розробників.

Місце роботи та посада рецензента “SPACE MARKETING”, Project manager
Науковий ступінь, учене звання (за наявності) _____
(підпис, ПІБ)

Місце печатки організації, де працює рецензент

АНОТАЦІЯ

У данному проєкті детально розглянуто розробку web-застосунку з відстеження харчової поведінки і фізичної активності. Проєкт має на меті поліпшення здоров'я та фізичної форми користувачів шляхом систематизації та аналізу їхнього режиму харчування та активності, що дозволить користувачам приймати обґрунтовані рішення щодо свого способу життя.

У роботі представлено аналіз предметної області, визначено основні проблеми та потреби користувачів, а також сформульовано бізнес-функціональні та нефункціональні вимоги до системи. Зокрема, розроблено інтуїтивно зрозумілий інтерфейс для введення та збереження інформації про прийом їжі, виконані фізичні вправи та інші важливі параметри, що впливають на здоров'я. Окрім цього, створено функціонал для аналізу отриманих даних, який надає корисні рекомендації користувачам на основі їхніх індивідуальних даних і цілей.

Реалізація проєкту включає в себе декілька ключових етапів: розробку детального проєкту системи, яка включає архітектуру, моделі даних та алгоритми, тестування та верифікацію рішень для забезпечення їхньої коректності та ефективності, а також прототипування для перевірки зручності та ефективності користувацького інтерфейсу. Особлива увага приділяється питанню безпеки та захисту персональних даних користувачів.

Проєкт також включає проведення експериментів для оцінки впливу використання системи на здоров'я та фізичну форму користувачів, що дозволить зробити обґрунтовані висновки щодо ефективності запропонованого рішення. У підсумку, web-застосування спрямоване на створення комплексного інструменту для підтримки здорового способу життя, що робить його надзвичайно актуальним у сучасному суспільстві, де здоров'я та фізична активність набувають все більшого значення.

РЕФЕРАТ

Обсяг проекту складає 92 сторінок, включаючи ілюстрації та таблиці. В роботі використано 21 літературних джерел.

Ключові слова: web-застосування, web-застосунку з відстеження харчової поведінки і фізичної активності, аналіз даних, рекомендації, здоров'я.

Реферат відображає предметну галузь розробка web-застосунку з відстеження харчової поведінки і фізичної активності. Мета роботи полягає в поліпшенні здоров'я та фізичної форми користувачів за допомогою систематизації та аналізу їхнього раціону харчування та активності.

Отримані результати включають розробку функціоналу для збору та аналізу даних, а також надання рекомендацій користувачам на основі цих даних. Проект демонструє новаторський підхід до проблеми контролю за здоров'ям через використання сучасних технологій.

Ступінь упровадження проекту визначається його готовністю до реалізації в життя. Проект може бути використаний у сферах здорового способу життя, фітнесу та медицини.

Основні конструктивні та техніко-експлуатаційні характеристики інформаційної системи включають зручний інтерфейс користувача, високу швидкість обробки даних та надійність функціонування.

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1 ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ	5
1.1 Характеристика предметної галузі та об'єкта дослідження.....	5
1.2 Аналіз літературних джерел та практичного досвіду використання ІС і технологій в предметній галузі.....	7
РОЗДІЛ 2 РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ ПІДСИСТЕМИ	15
2.1 Аналіз і специфікація вимог до інформаційної підсистеми.....	15
2.2 Постановка задачі та алгоритм розв'язання задачі.....	24
2.3 Моделювання інформаційної підсистеми	28
РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ.....	43
3.1. Інформаційне забезпечення	43
3.2 Технічне забезпечення	45
3.3 Програмне забезпечення	49
3.4 Результати реалізації інформаційної підсистеми	51
ВИСНОВКИ	61
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	63

ВСТУП

В сучасному світі здоров'я та фізична активність стають все більшою пріоритетною для багатьох людей. Зростання усвідомленості про важливість здорового способу життя призводить до зростання популярності різноманітних засобів та методів контролю за раціоном харчування та фізичною активністю. У цьому контексті обрана тема дослідження - розробка web-застосунку з відстеження харчової поведінки і фізичної активності - стає актуальною та значущою.

Цільовою аудиторією розробленого web-застосування є люди, які бажають контролювати свій раціон харчування та фізичну активність для підтримки здоров'я, досягнення спортивних цілей або покращення фізичної форми. Підвищена увага до здоров'я та спорту в сучасному суспільстві створює підґрунтя для успішного впровадження такого web-застосування, оскільки воно відповідає потребам цільової аудиторії.

Існуючі підходи та інформаційні системи для відслідковування раціону харчування та фізичних навантажень часто мають обмежену функціональність або не враховують сучасні тенденції у галузі здорового способу життя. Саме тому створення нового web-застосування, що відповідає сучасним вимогам та потребам користувачів, є необхідним кроком у вдосконаленні інструментів контролю за здоров'ям та фізичною формою.

Загальний обсяг проблеми, актуальність теми та потреба у вдосконаленні інструментів контролю за здоров'ям та фізичною активністю обґрунтовують необхідність розробки web-застосування для відслідковування раціону харчування та фізичних навантажень.

У світі швидкого темпу життя та поширення нових технологій, забезпечення здорового способу життя стає складнішим завданням для багатьох людей. Відсутність свідомого контролю над харчуванням та фізичною активністю може призвести до різних проблем зі здоров'ям, таких як надмірна вага, серцево-судинні захворювання, діабет та інші хвороби, які

можна уникнути або пом'якшити шляхом правильного раціону та активного способу життя.

Сучасні інформаційні технології можуть стати справжнім союзником у здійсненні контролю за здоров'ям. Розробка web-застосунку з відстеження харчової поведінки і фізичної активності може надати користувачам зручний та доступний інструмент для моніторингу їхнього харчування та активності, а також допомогти у встановленні та досягненні особистих цілей щодо здоров'я та фізичної форми.

Крім того, web-застосування може бути корисним інструментом для медичних працівників та дієтологів у веденні консультацій з пацієнтами та підтримці їх у процесі досягнення здорового способу життя.

Отже, обґрунтовуючи актуальність теми та необхідність створення інформаційної системи для відслідковування раціону харчування та фізичних навантажень, слід звернути увагу на широкий спектр можливостей, які вона може надати для покращення якості життя та забезпечення здоров'я.

РОЗДІЛ 1 ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Характеристика предметної галузі та об'єкта дослідження

Предметна галузь проекту "Розробка web-застосунку з відстеження харчової поведінки і фізичної активності" стосується управління здоров'ям і фітнесом за допомогою сучасних інформаційних технологій. Ця галузь є однією з найдинамічніших у світі, оскільки значна частина населення все більше зосереджується на підтримці здорового способу життя, що включає збалансоване харчування та регулярні фізичні вправи.

Головний об'єкт дослідження у цьому проекті - процес моніторингу та управління раціоном харчування і фізичними навантаженнями користувачів через web-застосунок. Цей процес охоплює різні етапи, такі як:

- Введення даних: Користувачі вводять інформацію про споживану їжу та виконані фізичні вправи.
- Аналіз даних: Система обробляє введені дані, аналізує їх та генерує корисні інсайти.
- Надання рекомендацій: Web-застосунок пропонує персоналізовані рекомендації щодо харчування та фізичної активності.

Ключовими аспектами дослідження є:

- Аналіз технічних вимог: Визначення необхідних функцій та характеристик web-застосунку.
- Проектування структури: Розробка бази даних, визначення основних компонентів системи та їх взаємодії.
- Розробка інтерфейсу: Створення зручного та інтуїтивно зрозумілого користувацького інтерфейсу.
- Безпека та конфіденційність: Забезпечення захисту персональних даних користувачів.

Характеристика предметної галузі та об'єкта дослідження включає кілька важливих аспектів:

- Зростаючий інтерес до здорового способу життя: Люди все більше усвідомлюють значення збалансованого харчування та регулярної фізичної активності для підтримки здоров'я.
- Розширення ринку цифрових рішень для здоров'я: Ринок програм та сервісів для моніторингу здоров'я швидко зростає, відповідаючи на потреби сучасних користувачів.
- Технологічний прогрес: Інформаційні технології дозволяють створювати більш точні та зручні інструменти для моніторингу здоров'я.
- Зручність використання: Користувачі шукають прості у використанні застосунки, які дозволяють легко відстежувати свій раціон та фізичну активність.
- Мотивація користувачів: Ефективні інструменти моніторингу допомагають користувачам залишатися мотивованими та досягати своїх цілей у сфері здоров'я.

Переваги проекту "Розробка web-застосунку з відстеження харчової поведінки і фізичної активності" включають:

- Інтуїтивний інтерфейс: Простий та зрозумілий дизайн, що забезпечує легке користування.
- Широка функціональність: Інструменти для детального моніторингу харчування та фізичної активності.
- Надійність та безпека: Захист персональних даних користувачів.
- Можливості розширення: Перспективи для додавання нових функцій та інтеграції з іншими сервісами.

Ці характеристики демонструють значення та перспективи проекту, який має потенціал значно покращити якість життя користувачів за

допомогою зручного та ефективного web-застосунку для моніторингу здоров'я.

1.2 Аналіз літературних джерел та практичного досвіду використання ІС і технологій в предметній галузі

Аналіз літературних джерел та дослідження практичного досвіду у сфері проектування розробки web-застосунку з відстеження харчової поведінки і фізичної активності є важливим етапом, що дозволяє зрозуміти сучасні тенденції, новітні технології та кращі практики. Основними джерелами інформації у цій галузі є наукові статті, книги, практичний досвід розробників та онлайн-ресурси.

Наукові статті та журнали надають глибокі уявлення про останні дослідження та інновації. Зокрема, у галузі проектування web-застосунків активно вивчаються такі технології, як штучний інтелект, машинне навчання та хмарні обчислення. Ці технології дозволяють створювати більш ефективні та інтерактивні web-застосунки, що допомагають користувачам краще стежити за своїм раціоном та фізичною активністю. Наприклад, використання алгоритмів машинного навчання для аналізу даних про харчування та тренування дозволяє створювати персоналізовані рекомендації, що враховують індивідуальні особливості користувачів.

Книги та підручники є цінним джерелом структурованих знань про технічні аспекти розробки web-застосунків. Вони описують різні етапи розробки, включаючи проектування архітектури системи, вибір технологій, розробку користувацького інтерфейсу та тестування. Крім того, книги часто містять приклади реальних проектів, що дозволяє розробникам отримати практичні навички та зрозуміти, як застосовувати теоретичні знання на практиці.

Практичний досвід розробників є безцінним джерелом знань про конкретні випадки використання технологій, усунення помилок та

оптимізацію процесів. Web-розробники діляться своїм досвідом через блоги, форуми та соціальні мережі, що дозволяє іншим фахівцям уникати поширених помилок та використовувати перевірені рішення. Наприклад, у випадку розробки web-застосунку для відстеження раціону харчування важливо враховувати аспекти зручності використання та інтеграції з іншими сервісами, такими як фітнес-трекери та додатки для моніторингу здоров'я.

Онлайн-ресурси та спільноти відіграють важливу роль у поширенні знань та спільного обговорення актуальних тем. Web-сайти, блоги та форуми дозволяють розробникам ділитися своїми досягненнями, обговорювати проблеми та знаходити рішення. Наприклад, спільноти на платформах, таких як Stack Overflow та GitHub, є важливими ресурсами для отримання допомоги та консультацій від інших розробників. Вони також надають доступ до великої кількості зразків коду та готових рішень, що сприяє швидкому впровадженню нових технологій та методів.

Використання інформаційних систем та технологій

У процесі розробки web-застосунку з відстеження харчової поведінки і фізичної активності використовуються різноманітні інформаційні системи (ІС) та інструменти, що допомагають забезпечити ефективність та якість розробки.

- Системи керування проектами: Інструменти, такі як Jira, Trello або Asana, дозволяють організувати робочий процес, відстежувати виконання завдань та контролювати терміни. Вони сприяють ефективній комунікації між членами команди та забезпечують чітке планування та управління проектом.
- Інструменти для прототипування та дизайну: Adobe XD, Figma та Sketch дозволяють створювати інтерактивні макети web-застосунків, визначати структуру сторінок та взаємодію з користувачем. Це допомагає розробникам та користувачам чітко візуалізувати кінцевий результат та внести необхідні зміни на ранніх етапах розробки.

- Системи керування контентом (CMS): WordPress, Drupal та Joomla дозволяють легко створювати, редагувати та оновлювати вміст web-застосунку без необхідності глибоких знань програмування. Це спрощує управління контентом та забезпечує гнучкість у змінах.
- Системи для версіонування коду: Git та SVN забезпечують ефективне управління версіями програмного забезпечення та дозволяють спільно працювати над кодом. Вони забезпечують можливість відстежувати зміни, об'єднувати різні гілки розробки та відновлювати попередні версії у разі потреби.
- Інструменти для автоматизованого тестування: Selenium та Cypress дозволяють автоматизувати тестування функціональності web-застосунку, перевіряти відповідність специфікаціям та виявляти помилки на ранніх етапах розробки. Це допомагає підвищити якість програмного забезпечення та забезпечити його надійність.

Етапи розробки web-застосунку

Процес розробки web-застосунку з відстеження харчової поведінки і фізичної активності включає кілька ключових етапів:

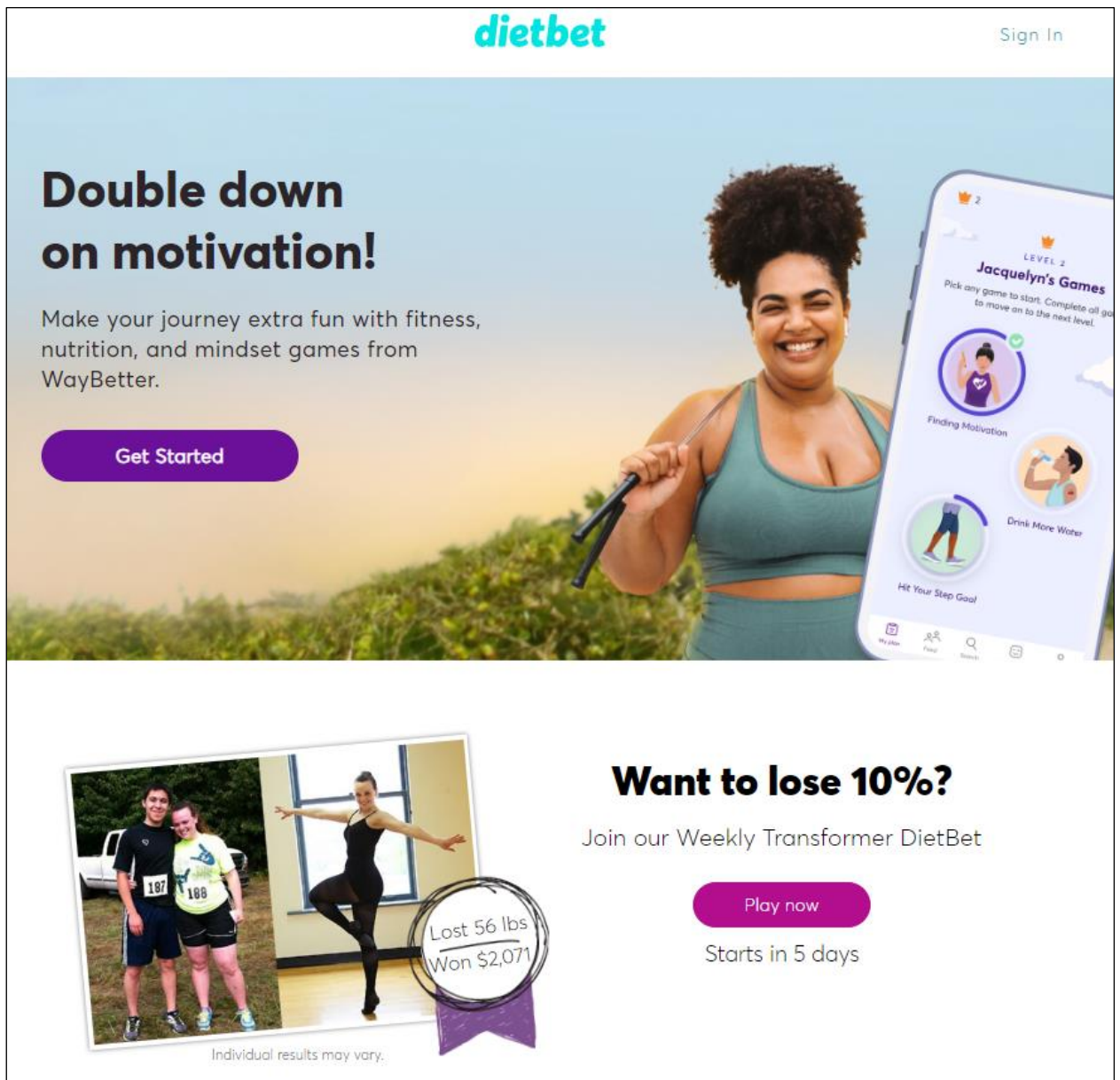
- Вивчення вимог клієнта: Глибоке дослідження та аналіз вимог клієнта дозволяє визначити необхідні функції та можливості web-застосунку, його вигляд та інтеграції з іншими системами.
- Розробка інформаційної архітектури: Визначення структури сторінок, навігації та взаємозв'язків між розділами допомагає організувати контент та забезпечити зручність використання.
- UI/UX-дизайн: Створення зручного та привабливого інтерфейсу користувача сприяє поліпшенню користувацького досвіду та робить web-застосунок більш зрозумілим та привабливим.
- Вибір технологій: Вибір відповідних технологій для фронтенду, бекенду, баз даних та API дозволяє забезпечити продуктивність, безпеку та масштабованість web-застосунку.

- Розробка прототипів та MVP: Створення прототипів та мінімально життєздатного продукту (MVP) дозволяє отримати зворотний зв'язок від користувачів та внести необхідні зміни на ранніх етапах розробки.
- Адаптивний дизайн: Забезпечення оптимального вигляду та функціональності web-застосунку на різних пристроях є важливим для зручності користувачів.
- Тестування та налагодження: Регулярне тестування та усунення помилок дозволяє підвищити якість програмного забезпечення та забезпечити його надійність.
- Документування: Створення зрозумілої документації допомагає підтримувати високу якість програмного забезпечення та забезпечує легкість його підтримки та масштабування.
- Запуск та підтримка: Підготовка web-застосунку до запуску, налаштування хостингу, деплоймент та моніторинг продуктивності є важливими для успішного запуску проекту.
- Зворотний зв'язок та аналітика: Використання інструментів для збору аналітичних даних дозволяє отримати інформацію про поведінку користувачів та виявити проблемні місця у застосунку.

Аналіз літературних джерел та практичного досвіду дозволяє забезпечити високоякісне та інноваційне проектування web-застосунків для відстеження раціону харчування та фізичних наван

Нижче наведено декілька практик, які можуть бути застосовані при розробці web-застосування для відслідковування раціону харчування та фізичних навантажень:

DietBet [<https://www.dietbet.com/>] - це онлайн-платформа, яка мотивує людей втрачати вагу та підтримувати здоровий спосіб життя через систему ставок на досягнення своїх цілей щодо втрати ваги. Користувачі можуть приєднуватися до різних груп та брати участь у викликах, що стимулює їх досягати поставлених цілей. Платформа пропонує інструменти для моніторингу харчування та фізичної активності, а також підтримує спільноту, де учасники можуть обмінюватися порадами та надавати підтримку один одному.



The image shows the DietBet website landing page. At the top, the 'dietbet' logo is in teal, and a 'Sign In' link is in the top right. The main section features a woman with curly hair in a teal sports top, holding a jump rope. To her left, the text reads 'Double down on motivation!' followed by 'Make your journey extra fun with fitness, nutrition, and mindset games from WayBetter.' Below this is a purple 'Get Started' button. To the right of the woman is a large smartphone displaying the app interface, which includes a 'LEVEL 2' badge, 'Jacquelyn's Games', and tasks like 'Finding Motivation', 'Drink More Water', and 'Hit Your Step Goal'. The bottom section has a photo of two men in athletic wear with bib numbers 187 and 188, and a woman in a black leotard performing a dance move. A circular badge over the photo says 'Lost 56 lbs' and 'Won \$2,071'. Below the photo is the text 'Individual results may vary.' To the right of the photo, the text reads 'Want to lose 10%?' followed by 'Join our Weekly Transformer DietBet'. Below this is a purple 'Play now' button and the text 'Starts in 5 days'.

Рисунок - 1.2.1: Головна сторінка порталу DietBet

Lose It! [<https://www.loseit.com/>] є популярним web-застосунком, який надає користувачам інструменти для відстеження споживаних калорій і фізичної активності. Користувачі можуть встановлювати особисті цілі щодо ваги та отримувати рекомендації по щоденному калорійному бюджету. Додаток також дозволяє сканувати штрих-коди продуктів для зручності введення даних та має велику базу продуктів харчування і страв.

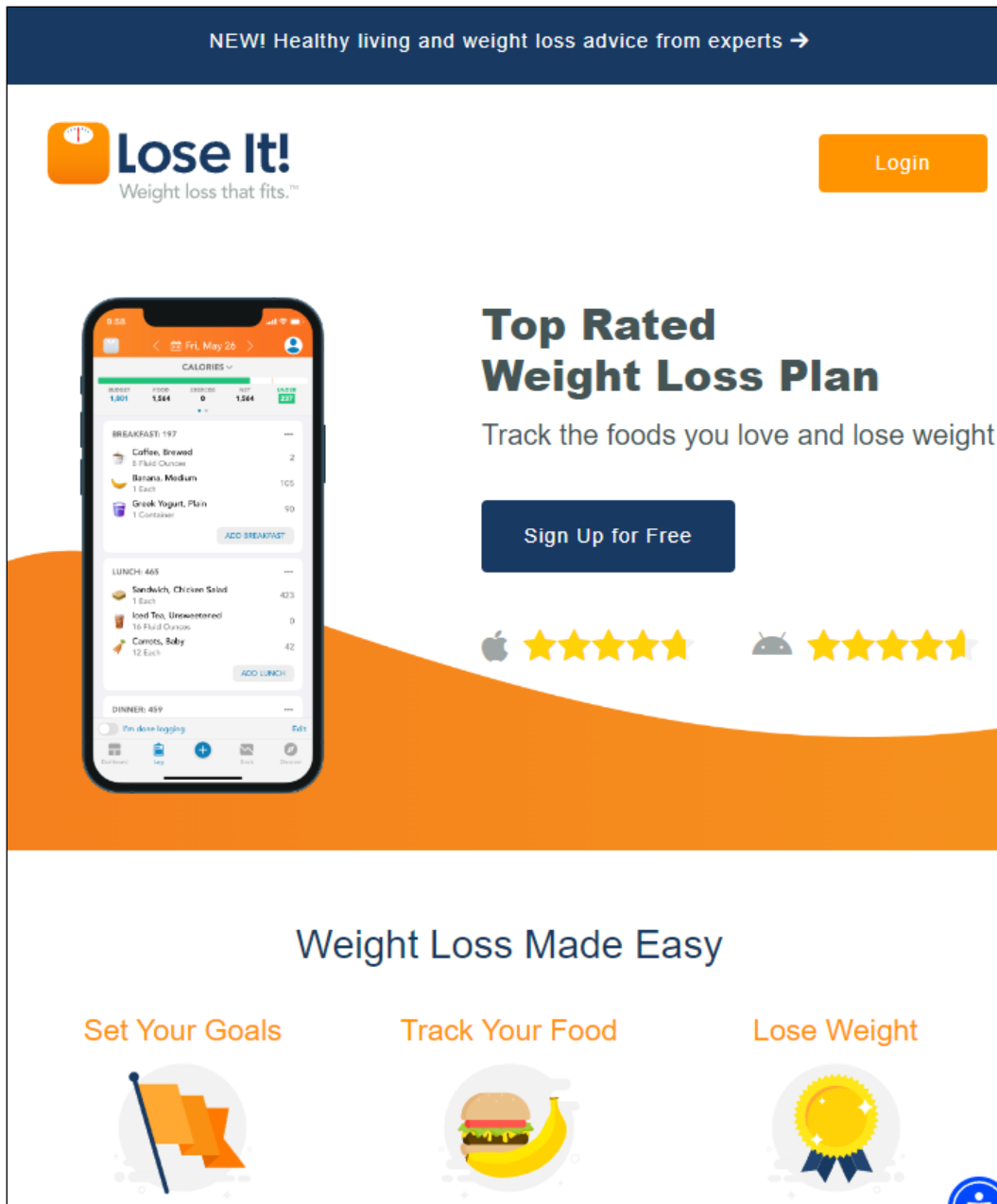


Рисунок - 1.2.2: Головна сторінка порталу Lose It!

Cronometer [<https://cronometer.com/>] - це web-застосунок, орієнтований на детальне відстеження харчування та здоров'я. Цей сервіс дозволяє користувачам реєструвати споживану їжу, вітаміни, мікроелементи та фізичну активність. Cronometer надає можливість налаштування під різні дієтичні режими і підтримує інтеграцію з іншими фітнес-трекерами.

Cronometer

Eat smarter. Live better.

Track your diet, exercise, and health data.

Sign Up - It's Free!

Eat Smarter. Live Better.

Develop healthy habits

Count your calories, ensure you're meeting nutrient targets, and see your progress over time.

Track up to 84 nutrients and other compounds

Log your meals and track all your macro and micronutrients.

Log meals, exercise and health metrics

Plus, you can create custom foods, recipes, exercises and metrics.

Get valuable health reports and charts

Learn how nutrients and metrics

Custom diet settings

Set weight, macro and nutrient targets to

Рисунок - 1.2.3: Головна сторінка порталу Cronometer

Таблиця 1.2.1 - Відмінності платформ

Характеристика	DietBet	Lose It	Cronometer
Тип платформи	Платформа для ставок на втрату ваги	Web-застосування для відслідковування калорій	Додаток для відслідковування харчування
Ціль користувачів	Втрата ваги та підтримка здорового способу життя	Контроль калорій та фізичної активності	Детальне відслідковування харчування та здоров'я
Функціональність	Ставки на втрату ваги, спільнота, підтримка	Відслідковування калорій, аналіз харчування	Запис харчування, вправ та здоров'я
Мотиваційна система	Використання ставок для мотивації	Спільнота, нагороди за досягнення цілей	Індивідуальні цілі, аналітика
Інтеграція	Зовнішні фітнес-трекери, платформи соціальних мереж	Фітнес-трекери, платформи соціальних мереж	Фітнес-трекери, харчування
Аналітика	Аналіз результатів ставок	Графіки, діаграми, звіти про прогрес	Детальний аналіз харчування, звіти
Цінова політика	Підписка, плата за участь у викликах	Безкоштовна базова версія, преміум підписка	Безкоштовна базова версія, преміум підписка
Користувацький інтерфейс	Простий, орієнтований на соціальну взаємодію	Інтуїтивний, зручний для користувачів	Інформативний, орієнтований на деталі
Підтримка користувачів	Форум, підтримка спільноти	FAQ, онлайн підтримка, форуми	FAQ, онлайн підтримка, блоги

Ця таблиця допомагає порівняти різні аспекти трьох платформ, які спрямовані на відслідковування раціону харчування та фізичних навантажень.

РОЗДІЛ 2 РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ ПІДСИСТЕМИ

2.1 Аналіз і специфікація вимог до інформаційної підсистеми

У цьому розділі визначено вимоги до розробки web-застосунку з відстеження харчової поведінки і фізичної активності. Вимоги до системи є ключовими аспектами, які визначають функціональність, надійність та якість кінцевого продукту. Цей розділ поділяється на три основні підрозділи: бізнес-вимоги, функціональні вимоги та нефункціональні вимоги.

Бізнес-вимоги відображають загальні цілі та задачі системи, які вона повинна виконувати для задоволення потреб користувачів та зацікавлених сторін. Вони визначають, які функції та можливості повинні бути реалізовані, щоб система була корисною та ефективною для своїх користувачів. Ці вимоги враховують потреби різних категорій користувачів, таких як звичайні користувачі, які вводять свої дані про харчування та фізичну активність, і адміністратори, які керують системою та забезпечують її безперебійну роботу.

Функціональні вимоги деталізують конкретні дії та процеси, які повинні бути реалізовані в системі для забезпечення її основних функцій. Вони описують, як система буде працювати, які інтерфейси будуть надані користувачам, і які операції будуть доступні. Функціональні вимоги охоплюють такі аспекти, як реєстрація та автентифікація користувачів, введення даних про харчування та фізичні навантаження, аналіз даних та формування рекомендацій, а також перегляд калорійності продуктів.

Нефункціональні вимоги визначають атрибути та характеристики системи, які впливають на її продуктивність, безпеку, надійність та зручність використання. Вони включають вимоги до швидкості обробки даних, доступності системи, захисту даних користувачів, масштабованості, зручності інтерфейсу та підтримки користувачів. Ці вимоги важливі для забезпечення високої якості продукту та задоволення очікувань користувачів щодо його роботи.

Докладне визначення вимог до системи дозволяє розробникам створити продукт, який відповідає очікуванням користувачів, забезпечує необхідну функціональність та підтримує високий рівень якості. Це також допомагає уникнути непорозумінь та забезпечує узгодженість між командою розробників та замовником.

Бізнес-вимоги до системи

- Реєстрація та автентифікація користувачів:
 - Система повинна надавати можливість користувачам створювати нові облікові записи.
 - Система повинна забезпечувати безпечний вхід в обліковий запис користувача через логін і пароль.
 - Підтримка функції відновлення паролю для користувачів.
- Введення даних:
 - Користувачі повинні мати можливість вводити інформацію про своє харчування, включаючи тип продукту, кількість спожитих калорій та час прийому їжі.
 - Користувачі повинні мати можливість вводити дані про свої фізичні навантаження, включаючи тип активності, тривалість та інтенсивність.
- Аналіз даних:
 - Система повинна аналізувати введені дані про харчування та фізичні навантаження.
 - На основі аналізу даних система повинна надавати рекомендації щодо поліпшення раціону харчування та фізичної активності.
- Перегляд калорійності продуктів:
 - Користувачі повинні мати доступ до бази даних продуктів із зазначенням їх калорійності та інших поживних характеристик.

- Система повинна надавати можливість пошуку продуктів у базі даних.
- Надання звітів та рекомендацій:
 - Система повинна надавати користувачам регулярні звіти про їх харчування та фізичні навантаження.
 - Система повинна генерувати індивідуальні рекомендації щодо покращення раціону харчування та фізичної активності.
- Підтримка користувачів:
 - Система повинна надавати технічну підтримку користувачам через відповідний інтерфейс або контактну форму.
 - Адміністратори системи повинні мати доступ до інструментів для управління користувацькими обліковими записами та вирішення технічних проблем.

Група стейкхолдерів	Пріоритет цілей
Замовник	Якісна реалізація проекту; Точність збору та аналіз вимог клієнта; Відповідність фінансових затрат клієнту; Строки виконання проекту; Досягнення максимального рівня прибутку
Розробник	Якісне програмне забезпечення; Виконання проекту у відведений термін; Розвиток професійних навичок; Максимальне використання ресурсів проекту; Задоволення потреб клієнту в розробленому ПЗ;
Команда проекту	Взаємодія між розробниками та замовниками; Ефективне використання ресурсів; Відповідність результатів вимогам клієнта; Задоволення умов проекту
Кінцевий користувач	Функціональність програмного забезпечення; Легкість використання та навігація сайту; Безпека та конфіденційність персональних даних;

	Доступність та швидкість роботи сайту; Якість наданих послуг.
--	--

Таблиця 2.1.1: Пріоритет цілей

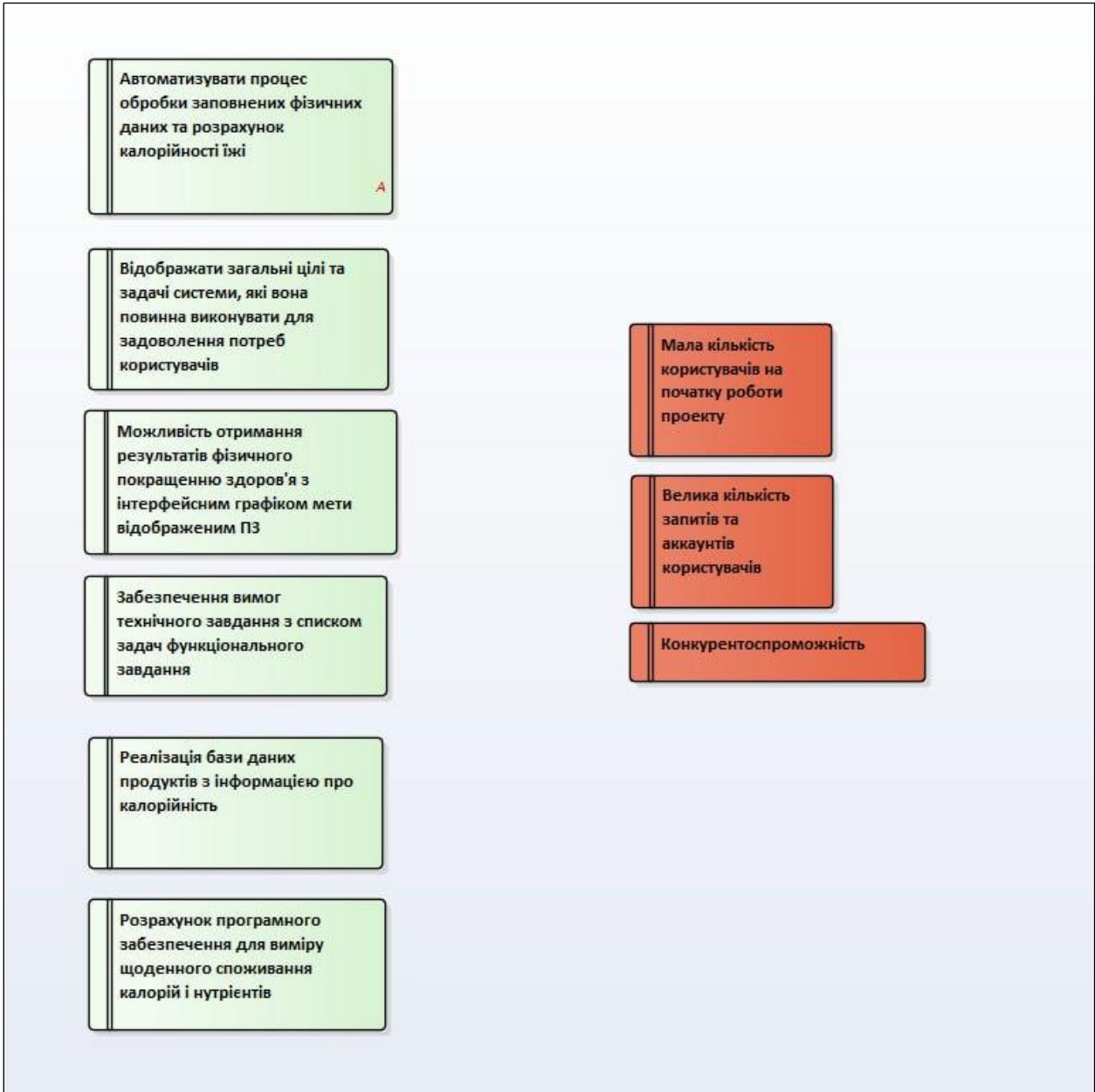


Рисунок 2.1.1: Бізнес вимоги

Функціональні вимоги до системи

Функціональні вимоги визначають конкретні функції, які система повинна виконувати.

- Функція реєстрації:
 - Користувачі можуть зареєструвати новий обліковий запис, вказавши необхідні дані (ім'я, електронна пошта, пароль).
 - Система повинна перевіряти унікальність електронної пошти перед створенням облікового запису.
- Функція автентифікації:
 - Користувачі можуть входити до системи, використовуючи свої облікові дані (електронна пошта і пароль).
 - Система повинна реалізовувати механізм скидання паролю через електронну пошту.
- Функція введення даних про харчування:
 - Користувачі можуть вводити дані про кожний прийом їжі (тип продукту, кількість, калорійність).
 - Система повинна зберігати введені дані у базі даних.
- Функція введення даних про фізичні навантаження:
 - Користувачі можуть вводити дані про кожну фізичну активність (тип активності, тривалість, інтенсивність).
 - Система повинна зберігати введені дані у базі даних.
- Функція аналізу даних:
 - Система повинна аналізувати дані про харчування і фізичні навантаження, щоб оцінити загальний стан користувача.
 - На основі аналізу система повинна формувати рекомендації.
- Функція перегляду калорійності продуктів:
 - Користувачі можуть переглядати інформацію про калорійність різних продуктів через зручний інтерфейс пошуку.

- Система повинна підтримувати можливість фільтрації продуктів за різними критеріями.
- Функція надання звітів та рекомендацій:
 - Система повинна надавати користувачам звіти про їхній прогрес за обраний період (день, тиждень, місяць).
 - Система повинна генерувати індивідуальні рекомендації на основі аналізу даних.
- Функція підтримки користувачів:
 - Користувачі можуть звертатися за технічною підтримкою через спеціальну форму.
 - Адміністратори можуть відповідати на запити користувачів та керувати їхніми обліковими записами.

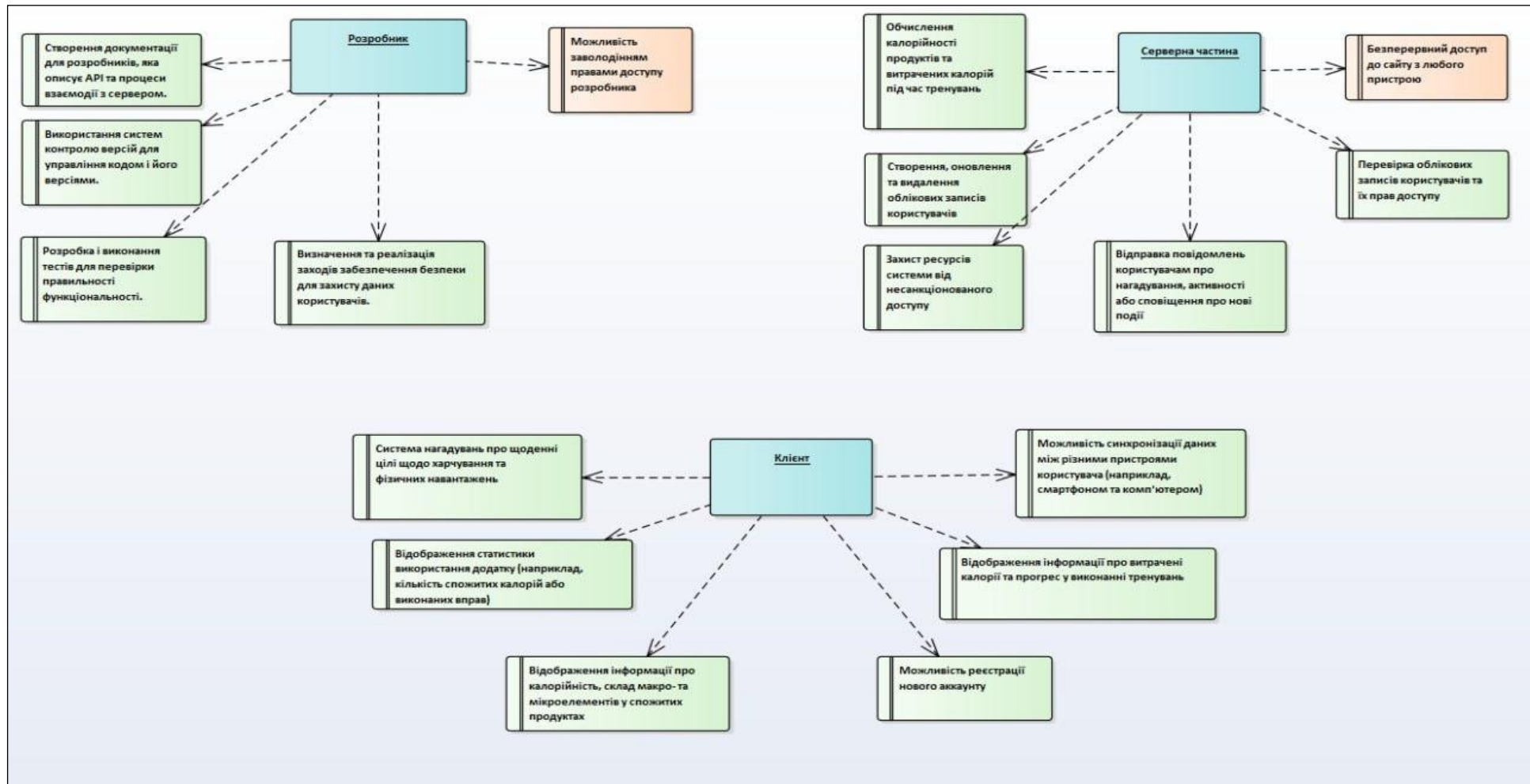


Рисунок 2.1.2: Функціональні вимоги

Нефункціональні вимоги до системи

Нефункціональні вимоги визначають характеристики, які система повинна мати для забезпечення належного рівня якості.

- **Продуктивність:**
 - Система повинна обробляти запити користувачів у режимі реального часу без значних затримок.
 - Час відповіді на запит не повинен перевищувати 2 секунд.
- **Надійність:**
 - Система повинна бути доступною не менше 99% часу на місяць.
 - Дані користувачів повинні регулярно зберігатися та резервуватися.
- **Безпека:**
 - Система повинна використовувати SSL/TLS для захисту даних, що передаються між користувачем та сервером.
 - Паролі користувачів повинні зберігатися у зашифрованому вигляді.

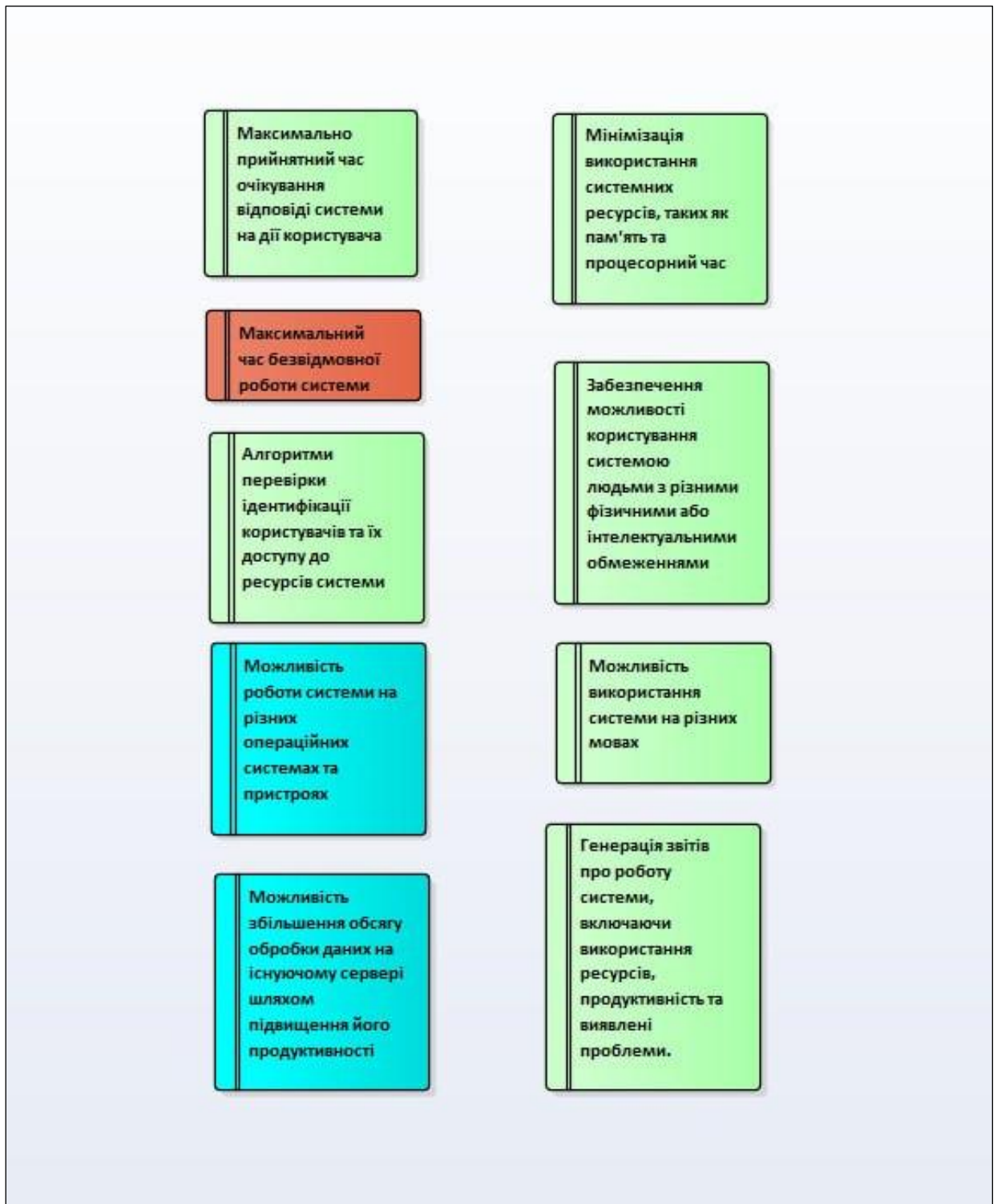


Рисунок 2.1.2: Нефункціональні вимоги

2.2 Постановка задачі та алгоритм розв'язання задачі

Постановка задачі:

Розробка web-застосунку з відстеження харчової поведінки і фізичної активності.

Характеристика задачі:

Задача полягає в проектуванні інформаційної системи web-застосунку з відстеження харчової поведінки і фізичної активності. Ця система призначена для автоматизації процесу запису та аналізу даних про споживану їжу та фізичну активність користувачів.

Призначення і техніко-економічна сутність задачі:

Основне призначення цієї інформаційної системи полягає в спрощенні та автоматизації процесу відслідковування харчування та фізичних навантажень. Вона дозволяє користувачам вводити свої дані про споживану їжу та фізичну активність, а також отримувати звіти про прогрес.

Ця інформація передається до бази даних для подальшого зберігання та обробки. Інформація про стан раціону та фізичні навантаження доступна для перегляду та аналізу, що сприяє зручному робочому процесу та ефективній комунікації між користувачами та адміністраторами сайту.

Об'єкти, за управлінням яких розв'язується задача:

Користувачі: вводять дані про своє харчування та фізичну активність, отримують інформацію про стан свого прогресу.

Адміністратори системи: відповідають за підтримку та обробку даних користувачів, ведення бази даних користувачів та їх діяльності.

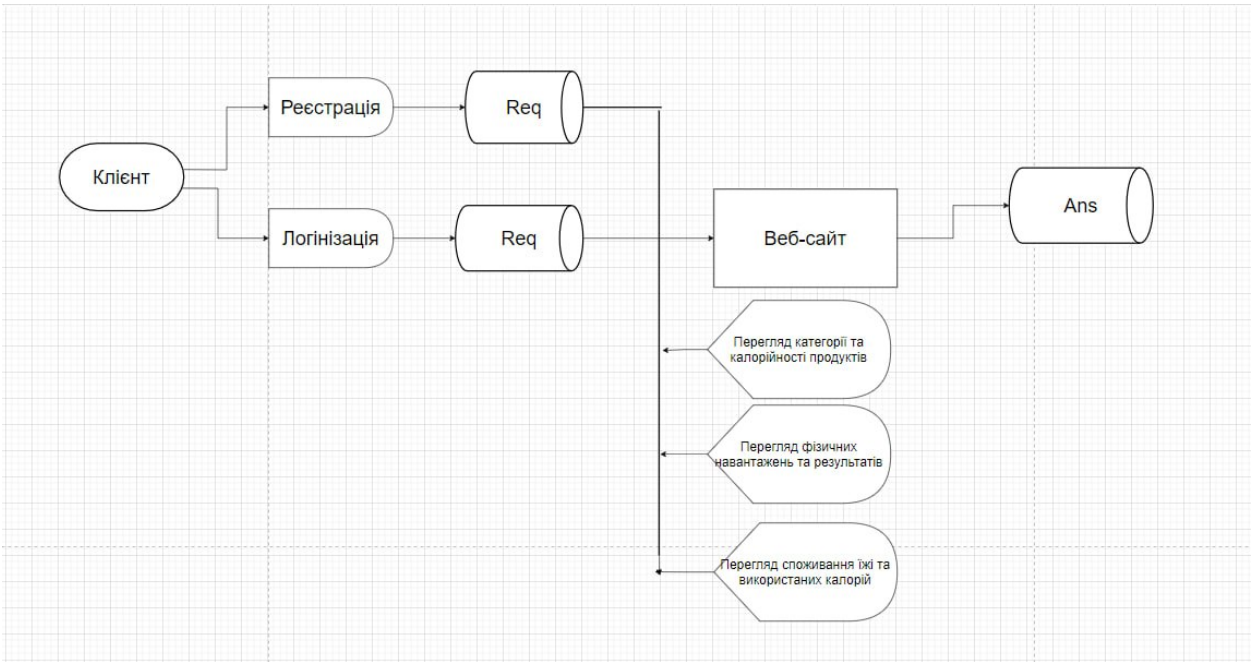


Рисунок 2.2.1 – Інформаційна модель задачі

Призначення й використання вихідної інформації:

Вихідна інформація використовується для керування процесами обслуговування, забезпечення ефективного моніторингу та аналізу даних, відстеження стану прогресу та звітності перед користувачами.

Таблиця 2.2: Вихідні дані системи

Таблиця 2.2.1: Вихідні дані системи

п/п	Назва вихідного повідомлення	Ідентифікатор	Форма подання	Періодичність видання	Термін видання і допустимий час затримки	Користувачі інформації
1	Статус раціону	ID001	електронне повідомлення	негайно	в момент зміни статусу	користувачі
2	Звіт про фізичні навантаження	ID002	web-сторінка	щоденно	до 19:00	користувачі
3	Підтвердження введення даних про харчування	ID003	електронне повідомлення	негайно	негайно	користувачі

4	Оновлення даних про фізичну активність	ID004	електронне повідомлення	при кожному оновленні	негайно	користувачі
5	Запит на уточнення даних	ID005	електронне повідомлення	при необхідності	негайно	користувачі
6	Завершення дня	ID006	електронне повідомлення	щоденно	до 22:00	користувачі

Вхідна інформація:

Вхідна інформація використовується для створення та оновлення записів про користувачів та їх діяльність.

Нижче наведено таблицю, в якій перелічені вхідні повідомлення.

Таблиця 2.2.2: Вхідні дані системи

п/п	Назва вхідного повідомлення	Ідентифікатор	Форма подання	Термін і частота надходження	Джерело
1	Введення даних про харчування	ID101	електронне повідомлення	Щоденно	Від користувачів
2	Введення даних про фізичну активність	ID102	web-форма	Щоденно	Від користувачів
3	Запит на консультацію щодо раціону	ID103	електронне повідомлення	За потребою	Від користувачів
4	Запит на зміну даних про активність	ID104	web-форма	За потребою	Від користувачів
5	Запит на аналіз харчування	ID105	електронне повідомлення	За потребою	Від користувачів
6	Запит на звіт про фізичні навантаження	ID106	web-форма	За потребою	Від користувачів

Алгоритм розв'язання задачі:

Використовувана інформація:

- **База даних користувачів:** зберігає інформацію про користувачів, їх харчування та фізичну активність;
- **Розклад фізичних навантажень:** містить дані про фізичну активність кожного користувача.

Результати розв'язання:

- **Статус раціону:** містить інформацію про поточний стан кожного користувача;
- **Звіт про фізичні навантаження:** містить інформацію про виконані фізичні активності та їх результати.

Алгоритм розв'язання задачі на ЕОМ:

- Отримати вхідні дані про харчування та фізичну активність користувача.
- Перевірити наявність запису про користувача в базі даних. Якщо запис відсутній, додати новий запис.
- Зберегти інформацію про харчування та фізичну активність в базу даних.
- Оновити статус раціону та фізичних навантажень, зберегти зміни в базі даних.
- Згенерувати та надіслати користувачу повідомлення про стан раціону та фізичної активності.
- Після завершення дня зберегти інформацію про результати та оновити статус.
- Згенерувати та надіслати користувачу звіт про виконані роботи.
- Завершити роботу.

Цей алгоритм забезпечує ефективне керування процесами відслідковування раціону харчування та фізичних навантажень, забезпечує користувачів необхідною інформацією про їх стан та прогрес. Його виконання включає отримання, збереження, оновлення та передачу даних

між різними компонентами системи, такими як web-сайт та база даних MySQL. Виконання цього проекту дозволить забезпечити зручний та ефективний процес відслідковування раціону харчування та фізичних навантажень через web-сайт, а також швидку та надійну обробку даних. Такий підхід сприятиме вдосконаленню бізнес-процесів та задоволенню потреб користувачів.

2.3 Моделювання інформаційної підсистеми

Проектування системи є критично важливим етапом розробки web-застосунку з відстеження харчової поведінки і фізичної активності. На цьому етапі визначається архітектура системи, її компоненти, взаємозв'язки між ними, а також розробляється користувацький інтерфейс, який забезпечує зручність та ефективність взаємодії користувачів із системою. Основна мета проектування полягає в створенні комплексного, ефективного та надійного продукту, який задовольняє всі вимоги бізнесу та користувачів.

Процес проектування системи включає декілька ключових аспектів:

Аналіз вимог та розробка архітектури системи

На початковому етапі проектування необхідно детально проаналізувати всі зібрані вимоги, включаючи бізнес-вимоги, функціональні та нефункціональні вимоги. Це дозволяє створити загальне уявлення про систему, її основні компоненти та їх взаємодію.

Архітектура системи визначає загальну структуру, включаючи серверну частину, базу даних, користувацький інтерфейс, а також зовнішні сервіси та API, які будуть використовуватися. Важливо врахувати масштабованість, надійність та безпеку системи при проектуванні архітектури.

Проектування бази даних

Проектування бази даних є важливим кроком, оскільки правильна структура даних забезпечує ефективне зберігання, обробку та доступ до інформації. Необхідно визначити всі сутності (наприклад, користувачі,

продукти харчування, фізичні навантаження), їх атрибути та зв'язки між ними. Це може бути реалізовано за допомогою діаграм сутність-зв'язок (ERD), які наочно відображають структуру даних.

Розробка серверної частини

Серверна частина системи відповідає за обробку запитів від клієнтської частини, взаємодію з базою даних та іншими сервісами. На цьому етапі визначаються основні модулі та сервіси, які будуть реалізовані, а також технології та фреймворки, що будуть використовуватися для розробки (наприклад, phpMyAdmin, PHP).

Важливо врахувати принципи розподіленої архітектури, такі як мікросервіси, щоб забезпечити гнучкість та масштабованість системи. Кожен мікросервіс повинен відповідати за конкретну функціональність і взаємодіяти з іншими мікросервісами через чітко визначені API.

Розробка користувацького інтерфейсу

Користувацький інтерфейс (UI) є важливою складовою системи, оскільки саме через нього користувачі взаємодіють із системою. На цьому етапі проектуються макети інтерфейсу, які відображають розташування елементів, навігацію та основні функціональні можливості. Важливо забезпечити зручність та інтуїтивність інтерфейсу, щоб користувачі могли легко виконувати необхідні дії.

Для створення прототипів інтерфейсу можуть використовуватися спеціалізовані інструменти, такі як Figma, Sketch або Adobe XD. Крім того, важливо врахувати принципи адаптивного дизайну, щоб забезпечити коректне відображення інтерфейсу на різних пристроях (настільних комп'ютерах, планшетах, смартфонах).

Забезпечення безпеки системи

Безпека є одним з ключових аспектів проектування будь-якої інформаційної системи. На цьому етапі необхідно врахувати всі потенційні

загрози та розробити відповідні заходи безпеки. Це може включати аутентифікацію та авторизацію користувачів, захист від SQL-ін'єкцій, забезпечення конфіденційності та цілісності даних, а також регулярне проведення аудиту безпеки.

Тестування та верифікація

Проектування системи також включає розробку планів тестування для перевірки функціональності, продуктивності, безпеки та зручності використання системи. Тестування проводиться на всіх етапах розробки, щоб виявити та усунути можливі помилки на ранніх стадіях. Важливо також провести верифікацію, щоб переконатися, що система відповідає всім вимогам та очікуванням користувачів.

Таким чином, проектування системи є комплексним процесом, який включає детальний аналіз вимог, розробку архітектури, проектування бази даних, реалізацію серверної та клієнтської частин, забезпечення безпеки та проведення тестування. Кожен з цих етапів є критично важливим для створення ефективної, надійної та зручної системи, яка задовольняє потреби користувачів та бізнесу.

Моделювання поведінки системи є важливим етапом у процесі розробки web-застосування для відслідковування раціону харчування та фізичних навантажень. На цьому етапі розробники описують, як система реагує на різні події, які дії виконуються у відповідь на певні події, а також які зміни станів відбуваються у системі в процесі її роботи. Моделювання поведінки системи допомагає забезпечити чітке розуміння того, як система функціонує, а також ідентифікувати можливі проблеми на ранніх стадіях розробки.

Основні інструменти моделювання поведінки системи включають:

Діаграми діяльності (Activity Diagrams)

Діаграми діяльності відображають потік дій та процесів у системі. Вони показують, які дії виконуються, в якому порядку і як ці дії взаємодіють одна з

одною. Для нашого web-застосування діаграми діяльності можуть включати такі процеси, як реєстрація користувачів, введення даних про харчування, відстеження фізичних навантажень та генерація звітів.

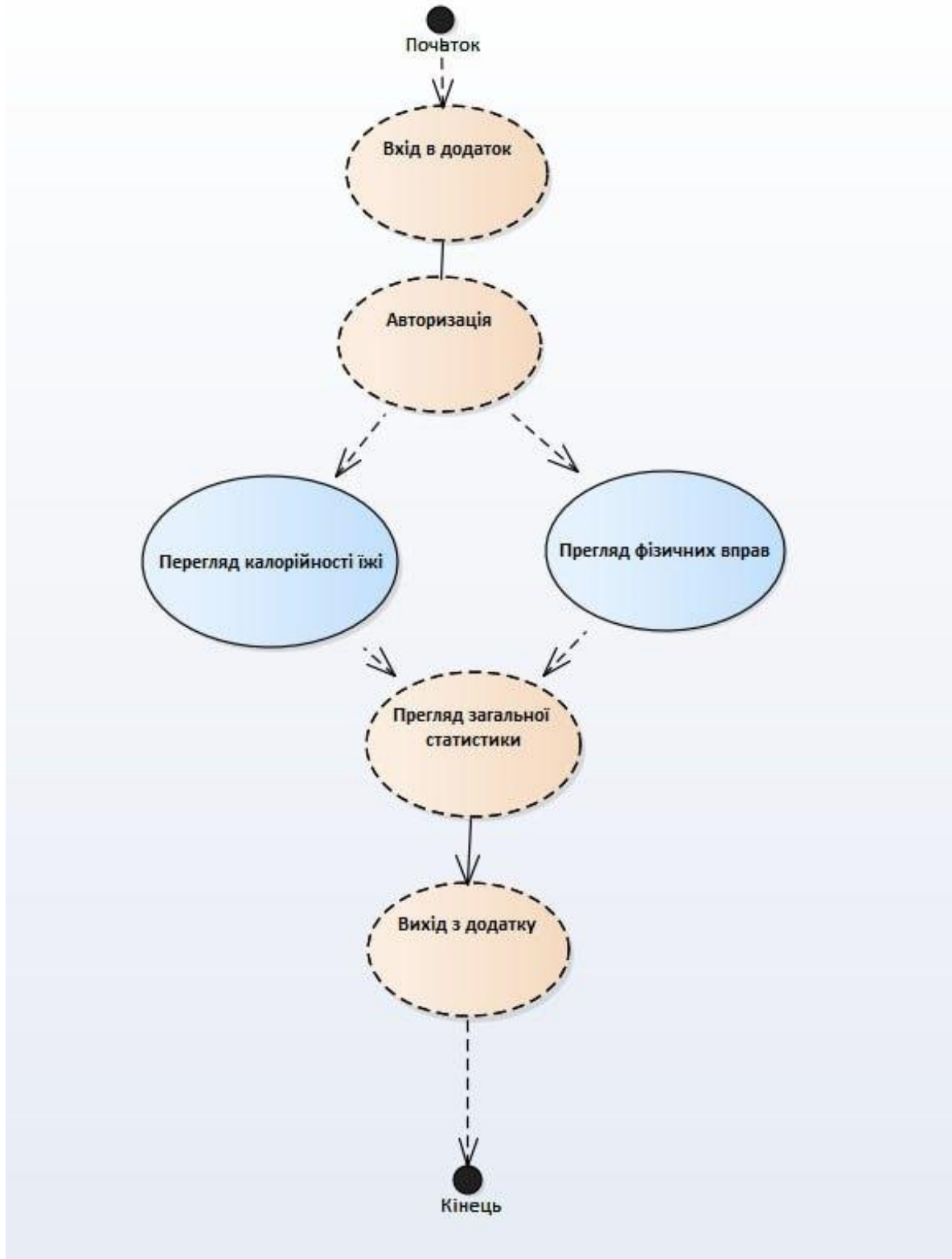


Рисунок 2.3.1: діаграми діяльності

Діаграми станів (State Diagrams)

Діаграми станів відображають різні стани, через які проходять об'єкти в системі, а також події, які викликають зміну станів. Вони є корисними для моделювання поведінки системи на більш детальному рівні, особливо для відстеження станів окремих компонентів або сутностей.

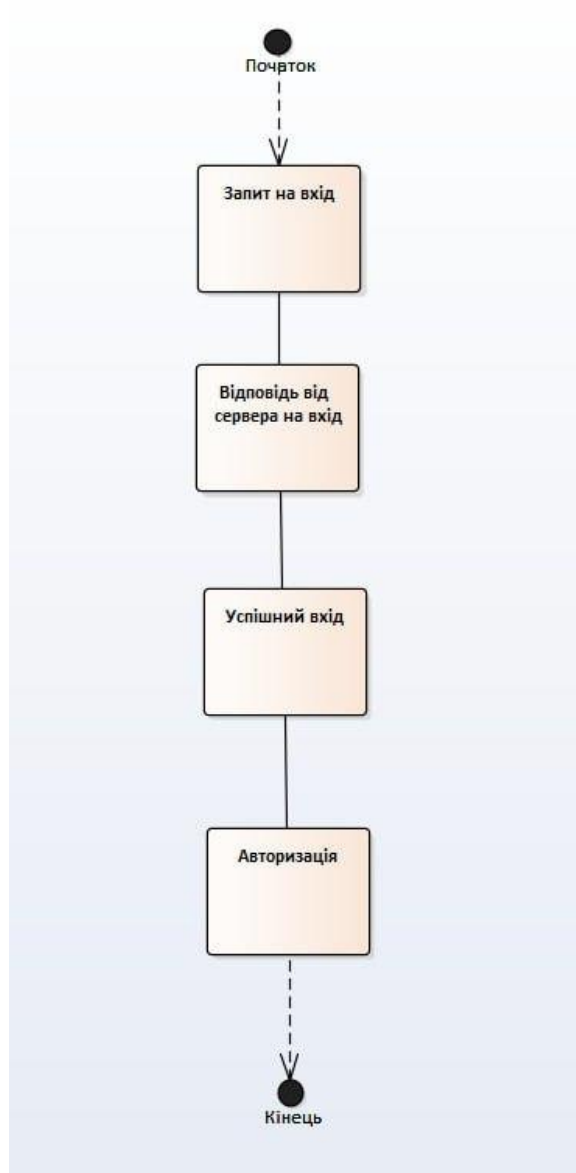


Рисунок 2.3.2: діаграми станів

Діаграми прецедентів (Use Case Diagrams)

Діаграми прецедентів відображають взаємодію між користувачами (акторами) і системою, визначаючи функціональні вимоги та сценарії використання.

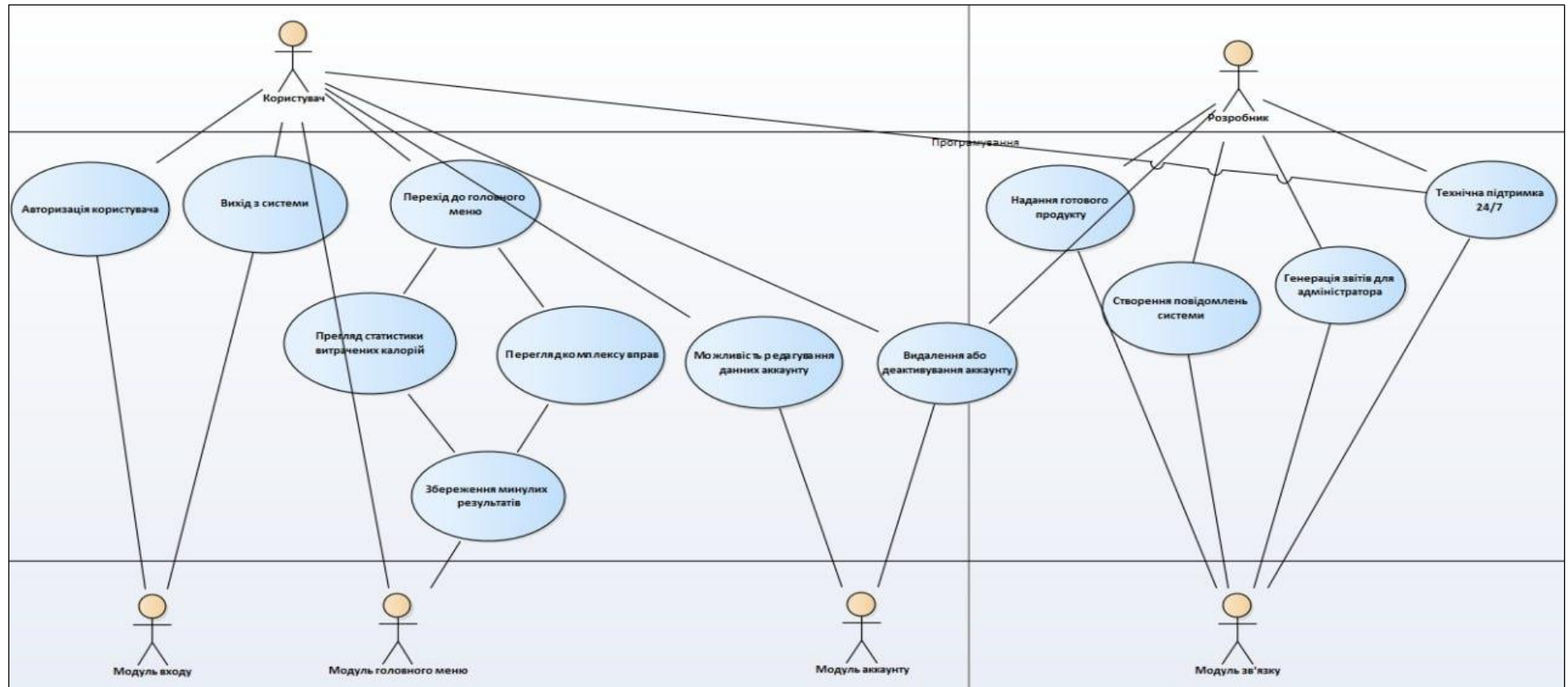


Рисунок 2.3.3 діаграма прецедентів

Діаграми послідовностей (Sequence Diagrams)

Діаграми послідовностей ілюструють взаємодію між різними об'єктами системи у вигляді послідовності обміну повідомленнями. Вони показують, як об'єкти взаємодіють один з одним протягом виконання певної функції або процесу.

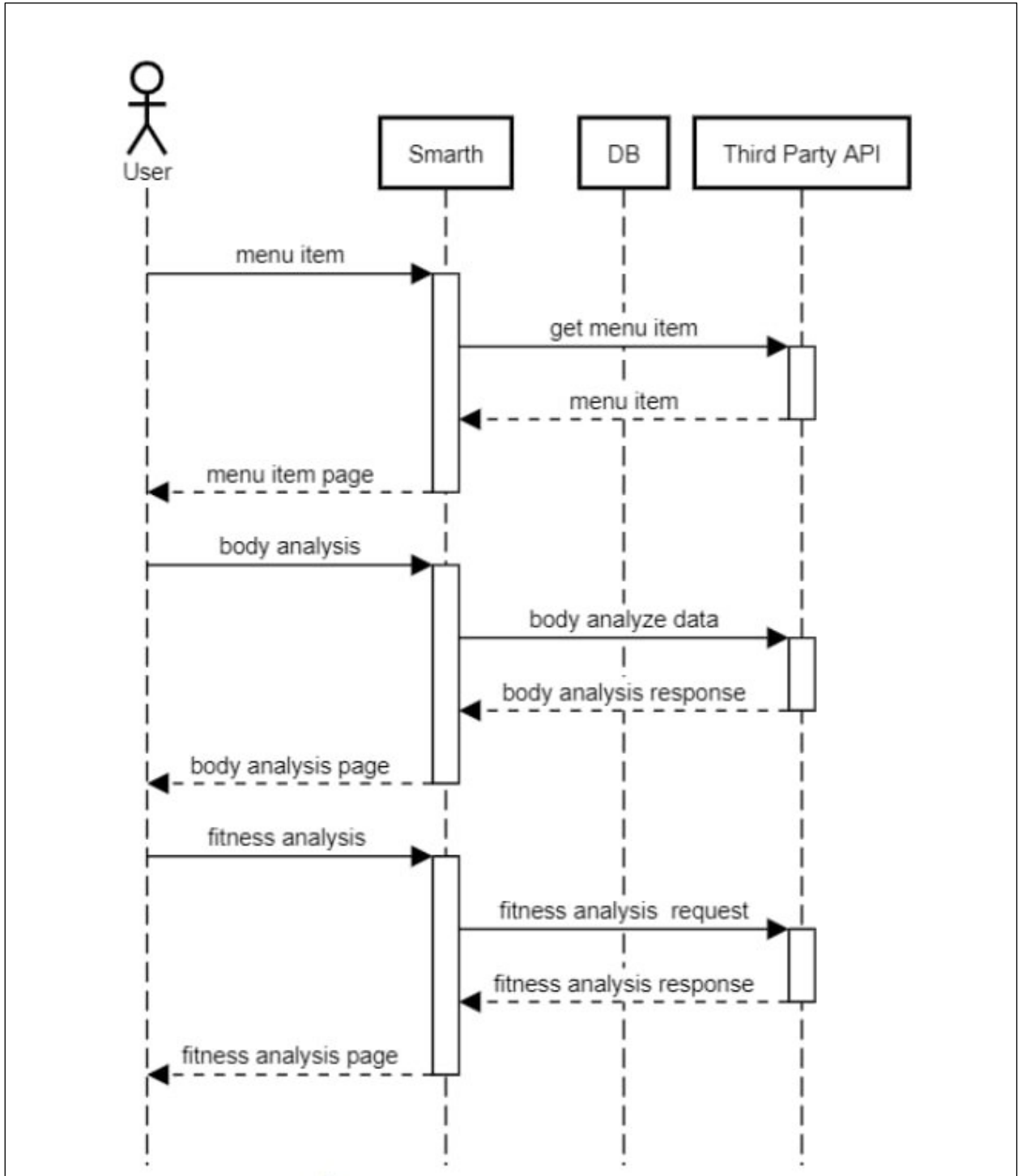


Рисунок 2.3.4 діаграма послідовностей

Моделювання поведінки системи є важливим інструментом для розуміння того, як система функціонує та взаємодіє з користувачами і зовнішніми системами. Використовуючи різні види діаграм, розробники можуть виявити та вирішити потенційні проблеми на ранніх етапах розробки, що дозволяє створити надійний, зручний та ефективний продукт.

Моделювання структури системи

Моделювання структури системи є важливим етапом у процесі розробки web-застосунку з відстеження харчової поведінки і фізичної активності. На цьому етапі створюється детальна статична модель системи, яка визначає її внутрішню структуру, компоненти та взаємозв'язки між ними. Це допомагає розробникам і архітекторам отримати чітке уявлення про будову системи, що забезпечує ефективне планування та реалізацію проекту.

Діаграми визначення блоків (Block Definition Diagrams)

Діаграми визначення блоків (BDD) у SysML використовуються для подання внутрішньої структури системи у вигляді ієрархії блоків різних типів, включаючи технічне забезпечення, програмне забезпечення та інші компоненти.

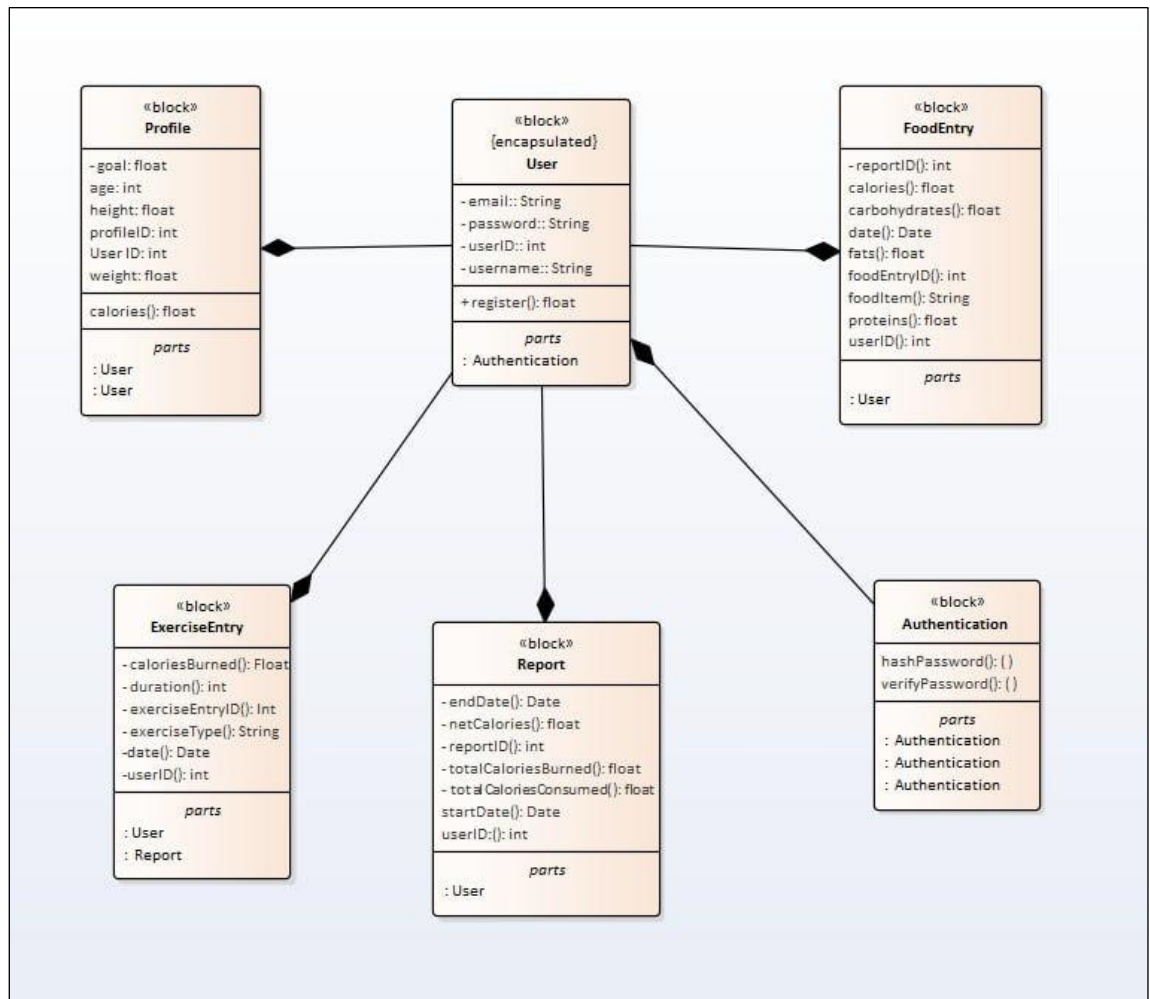


Рисунок 2.3.5 діаграми визначення блоків

Діаграми внутрішніх блоків (Internal Block Diagrams)

Діаграми внутрішніх блоків (IBD) у SysML використовуються для подання блоків як частин системи, їх з'єднань між собою безпосередньо або через порти, а також потоків, що існують між частинами та портами. Це дозволяє детально описати, як компоненти системи взаємодіють між собою.

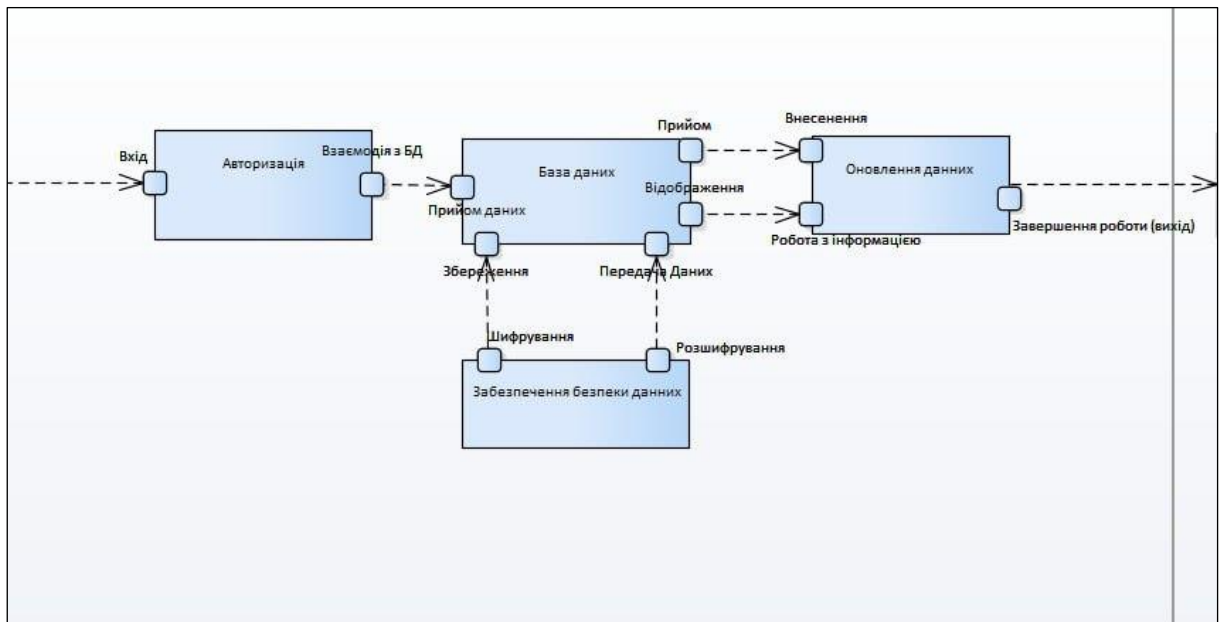


Рисунок 2.3.6 діаграма внутрішніх блоків

Діаграми класів (Class Diagrams) для інформаційних систем

У випадку, коли тема курсового проекту передбачає проектування інформаційної системи, у цьому підрозділі наводяться діаграми класів UML для моделювання статичної структури класів системи і зв'язків між ними. Діаграми класів допомагають визначити сутності, їх атрибути, методи та.

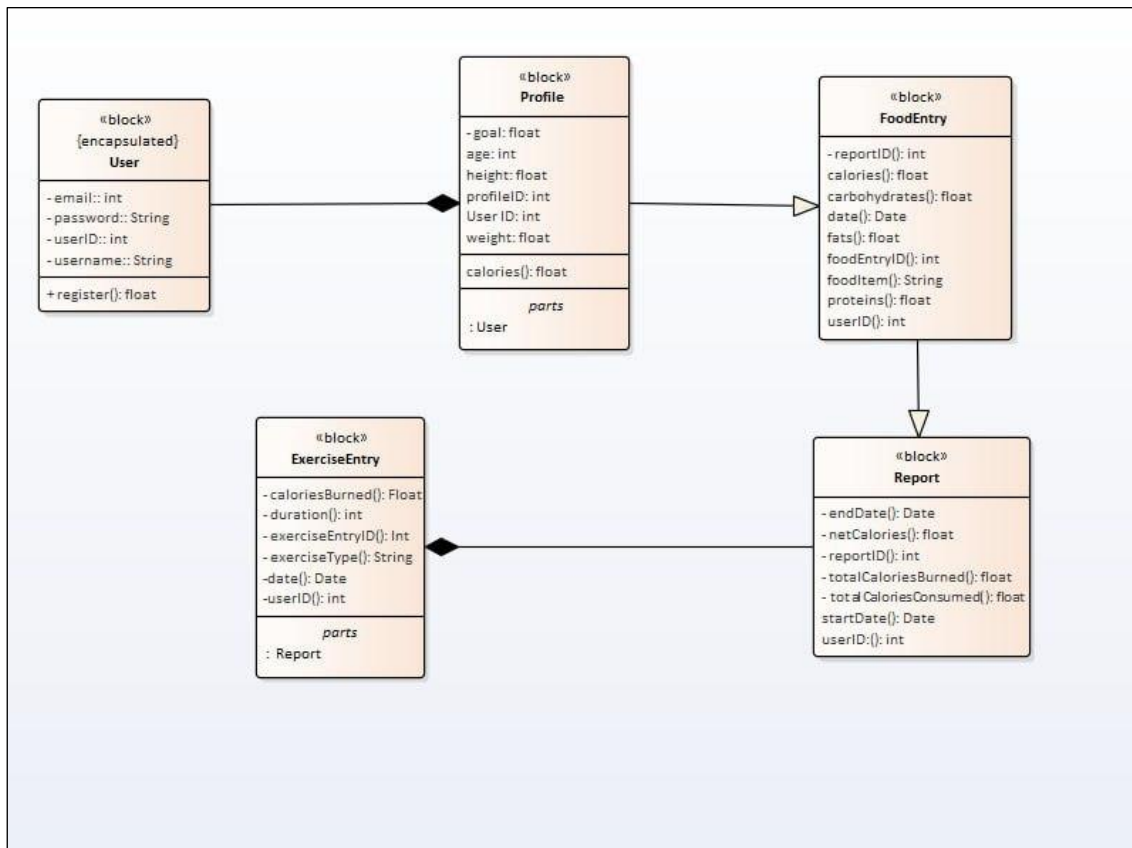


Рисунок 2.3.7 діаграми класів

Пояснення основних класів стереотипів

- **Класи-сутності:** використовуються для моделювання даних і поведінки з довгим життєвим циклом. Прикладом може бути клас "User", який зберігає інформацію про користувачів системи.
- **Граничні класи:** моделюють інтерфейси системи на високому рівні. Наприклад, клас "ClientApp" відповідає за взаємодію користувача з системою через інтерфейс.
- **Класи керування (класи-менеджери):** створюються для моделювання послідовної поведінки одного або кількох прецедентів і координації подій, що реалізують закладену в ці прецеденти поведінку. Наприклад, клас "WebServer" відповідає за обробку запитів і координацію взаємодії між клієнтським застосунком та базою даних.

Таким чином, моделювання структури системи включає створення діаграм визначення блоків, діаграм внутрішніх блоків та діаграм класів. Це дозволяє розробникам детально описати всі компоненти системи, їх властивості, методи та взаємозв'язки, що забезпечує прозорість, зрозумілість та ефективність процесу розробки.

Розробка тест-кейсів

Тестування системи є важливою складовою процесу розробки програмного забезпечення. Основна мета тестування полягає в тому, щоб забезпечити якість та надійність системи, виявити та усунути помилки та дефекти перед її введенням в експлуатацію. Для того аби забезпечити повноту та ефективність тестування, перед його проведенням формуються тест кейси.

Розроблені тест-кейси дозволяють виконувати тестування в стандартизованому та повторюваному форматі. Це допомагає виявляти більше помилок та забезпечувати високу якість тестування. Також вони дозволяють оцінити покриття програмного забезпечення функціональними вимогами, що допомагає виявляти прогалини та забезпечувати повноту тестування.

В процесі розробки даної системи засобами представленими в Enterprise Architect було розроблено тест кейси. Для зручності вони об'єднані в test suits, відповідно до того який функціонал вони покривають. Розроблені тести наведено на рисунках нижче.

Таблиця 2.3.1 Таблиця даних тест-кейсів

Дія	Очікуваний результат	Результат тесту (пройшов/непройшов/ заблоковано)
Завантажити головну сторінку після збереження даних у LocalStorage	Дані після перезавантаження сторінки автоматично завантажуються, і сторінка відкривається в тому стані, в якому вона була на момент збереження.	Тест пройшов

Дія	Очікуваний результат	Результат тесту (пройшов/непройшов/ заблоковано)
Введення різних типів символів у поля реєстрації	Система повинна допустити використання різних типів символів (a-Z, 0-9, !@#\$%^&*()_').	Тест непройшов
Зайти на сторінку реєстрації	При першому заході на сторінку користувачеві пропонується ввести особисті дані (ім'я, електронна пошта, пароль).	Тест пройшов
Зайти на сторінку логіна після реєстрації	При повторному заході на сторінку після реєстрації користувач повинен потрапити на сторінку входу.	Тест пройшов
Перегляд калорійності продуктів з різною кількістю даних	Всі продукти з їхніми калорійностями повинні відображатися коректно, незалежно від кількості даних.	Тест пройшов
Додавання фізичних навантажень	Поля для введення типу активності, тривалості та кількості спалених калорій повинні коректно зберігатися та відображатися.	Тест пройшов
Видалення фізичних навантажень зі списку	Після видалення останнього запису фізичної активності, список повинен бути порожнім.	Тест пройшов
Введення тексту в числові поля (наприклад, тривалість активності)	Система повинна видавати помилку про некоректно введені числові дані.	Тест пройшов
Налаштування профілю користувача	Всі зміни в налаштуваннях профілю повинні коректно зберігатися та відображатися при наступних входах у систему.	Тест пройшов
Перегляд графіків фізичних навантажень за різні періоди	Графіки повинні коректно відображати дані за обраний період (день, тиждень, місяць).	Тест пройшов
Додавання нового продукту в раціон	Поля для введення назви продукту, кількості та калорійності повинні коректно зберігатися та відображатися у списку продуктів.	Тест пройшов

Дія	Очікуваний результат	Результат тесту (пройшов/непройшов/ заблоковано)
Видалення продукту зі списку раціону	Після видалення продукту зі списку, він більше не повинен відображатися в списку продуктів.	Тест пройшов
Зміна налаштувань інтерфейсу (наприклад, тема оформлення)	Всі зміни в налаштуваннях інтерфейсу повинні коректно зберігатися та застосовуватися після перезавантаження сторінки.	Тест не пройшов

Трасування вимог до системи

Верифікація проектних рішень здійснюється шляхом трасування вимог до системи, визначених у розділі 2, та їх співвіднесення з проектними компонентами і артефактами. Надається діаграма трасування, яка включає вимоги, прецеденти, варіанти тестування та елементи користувацького інтерфейсу з відповідними зв'язками між ними. У додатку додається матриця взаємозв'язків, яка доповнює діаграму трасування.

РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ

3.1. Інформаційне забезпечення

В процесі розробки web-застосунку з відстеження харчової поведінки і фізичної активності, було обрано СКБД MySQL з використанням phpMyAdmin як основу для створення інформаційного сховища даних. MySQL є реляційною базою даних, що забезпечує високу продуктивність, надійність та зручність у використанні. phpMyAdmin є web-застосунком, який надає інтуїтивно зрозумілий інтерфейс для адміністрування MySQL баз даних.

Цей вибір зроблений з метою забезпечення структурованого зберігання даних, швидкого доступу до них, а також надання розробникам зручних інструментів для роботи з даними. MySQL разом з phpMyAdmin може бути використана для проекту web-сайту для відслідковування раціону харчування та фізичних навантажень з наступними аргументами:

- **Структурованість та надійність:** MySQL забезпечує структуроване зберігання даних, що важливо для точного відслідковування раціону та фізичної активності. Реляційна модель даних дозволяє легко зв'язувати різні таблиці та проводити складні запити.
- **Продуктивність:** MySQL є високопродуктивною СКБД і надає швидкий доступ до даних. Вона використовує індекси для прискорення пошуку і запитів до бази даних. Також, вона підтримує розподілені операції читання та запису, що забезпечує розподілення навантаження і покращує продуктивність в системах з великою кількістю одночасних запитів.
- **Масштабованість:** MySQL підтримує горизонтальне масштабування, що дозволяє розширювати базу даних на багато серверів для обробки великих обсягів даних.

- **Безпека:** MySQL надає розширені засоби безпеки, включаючи контроль доступу та шифрування даних, що забезпечує захист особистої інформації користувачів.
- **Зручність адміністрування:** phpMyAdmin надає зручний графічний інтерфейс для адміністрування MySQL баз даних, що дозволяє легко створювати, модифікувати та управляти базами даних, таблицями та запитами.
- **Широка підтримка та екосистема:** MySQL має велику спільноту користувачів, що сприяє наявності документації, підтримки та відповідей на питання. Вона також має багату екосистему інструментів і бібліотек для розширення можливостей СКБД.

Враховуючи ці аргументи, MySQL разом з phpMyAdmin є гарним вибором для проекту web-сайту для відслідковування раціону харчування та фізичних навантажень. Вони надають структурованість, надійність, продуктивність, масштабованість та зручність адміністрування, що дозволяє ефективно зберігати, керувати та отримувати доступ до даних про користувачів, їх раціон та фізичну активність.

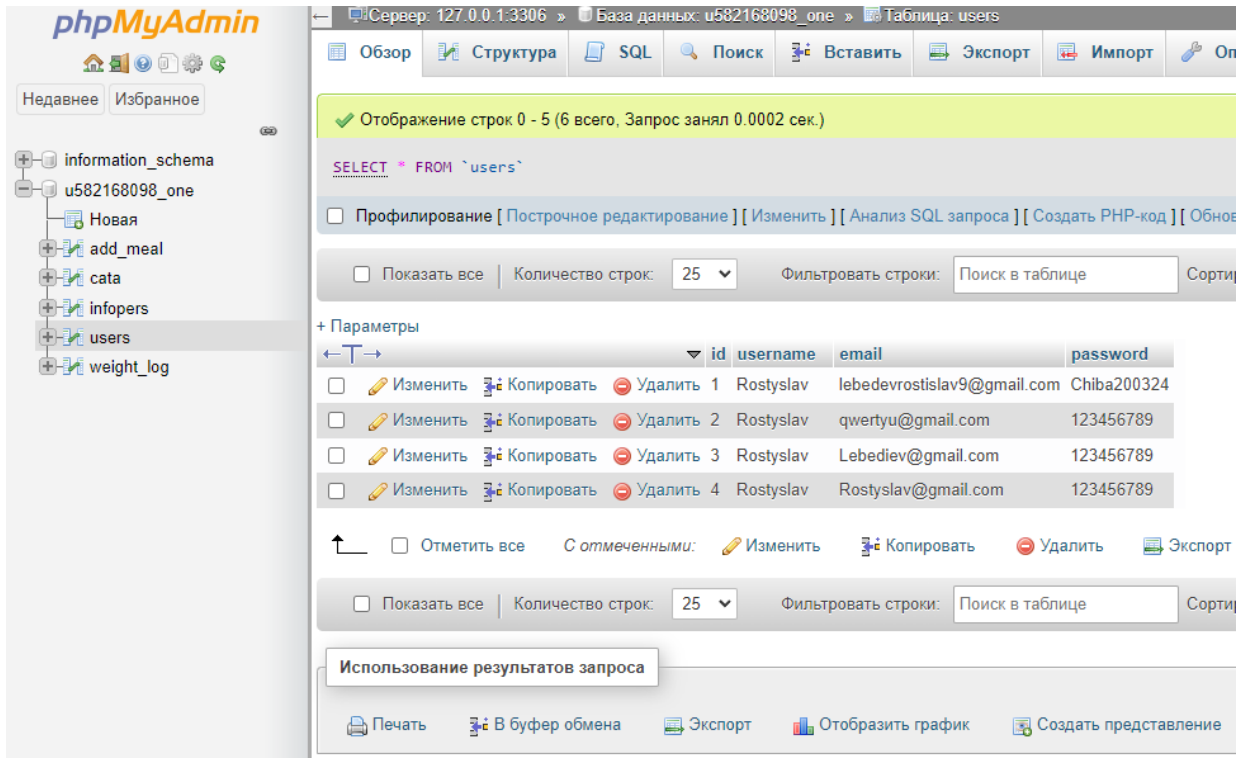


Рисунок 3.1.1 - Елементи масиву користувачів БД

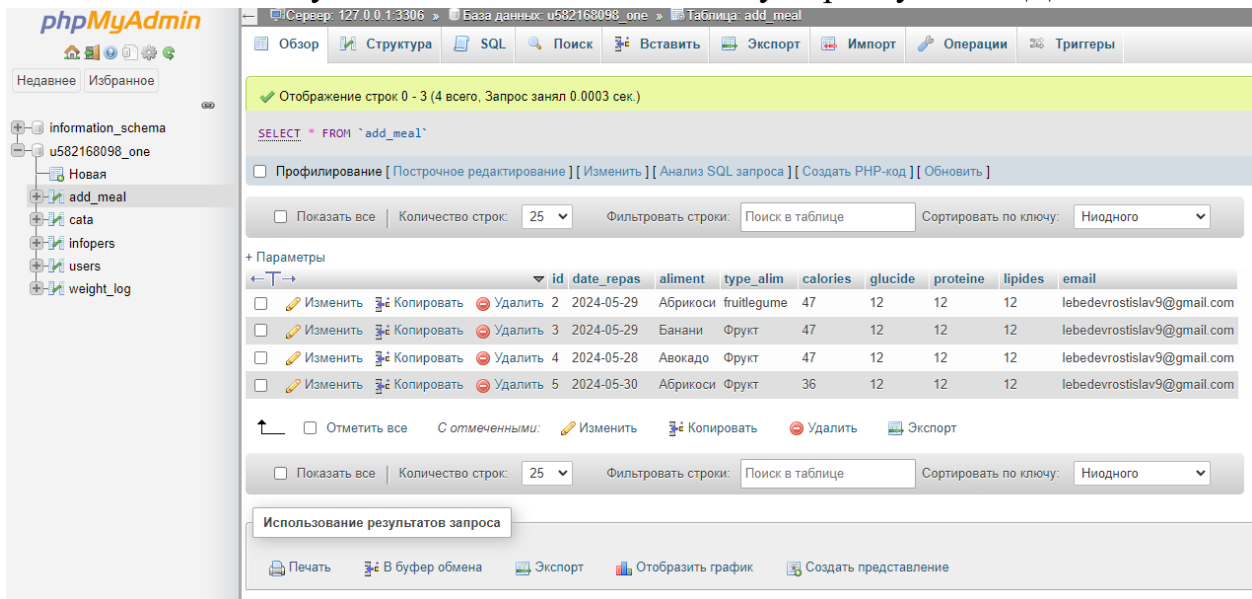


Рисунок 3.1.2 - Елементи масиву вибраного раціону БД

3.2 Технічне забезпечення

Технічне забезпечення для web-сайту, який забезпечує відслідковування раціону харчування та фізичних навантажень, може бути наступним.

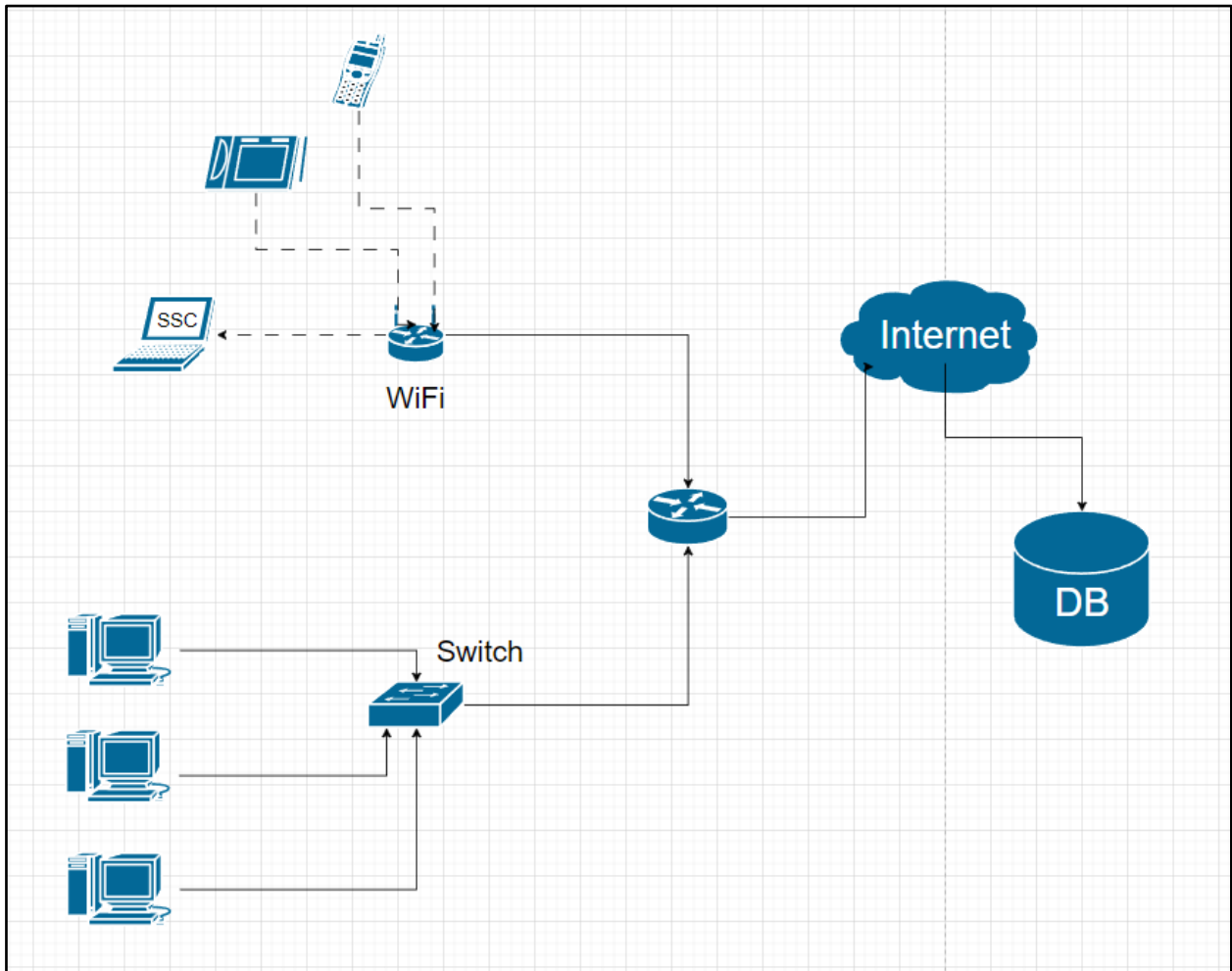


Рисунок 3.2.1 – Організація комп'ютерної мережі

Віддалений сервер

- Розміщення ядра системи, бази даних та необхідних файлів: Віддалений сервер є основним місцем для зберігання та обробки даних, включаючи інформацію про користувачів, їх раціон харчування та фізичну активність.
- Оренда серверу: Використання орендованого серверу дозволяє економити витрати на обладнання та підтримку, забезпечуючи при цьому високу доступність та надійність.
- Доступність: Сервер забезпечує зручний доступ користувачів до системи через web-сайт з будь-якого пристрою з підключенням до інтернету.

Комплекс технічних засобів

- Потреби різних відділів: Структура комплексу технічних засобів враховує потреби різних відділів, таких як розробка, тестування та підтримка.
- Хостинг-провайдер: Використовується хостинг-провайдер з ємністю твердотілого NVMe накопичувача розміром 100 ГБ та пропускнуою здатністю до 1 Гбіт.

Вибір ЕОМ і серверів

- Робочі станції: Використовуються робочі станції серії "Artline Business" з процесорами Intel Core i7, 16 ГБ оперативної пам'яті та 500 ГБ на твердотілому накопичувачі.
- Сервери баз даних: Обираються сервери з високою продуктивністю та надійністю для ефективної роботи системи.

Технічні засоби в обчислювальній мережі

- Датчики та збір даних: Використовуються різноманітні датчики для збору та аналізу потрібних даних про фізичну активність користувачів.
- Вибір периферійних технічних засобів
- Системи відеоспостереження: Встановлюються системи відеоспостереження для забезпечення безпеки даних та моніторингу роботи системи.

Акустичні системи:

- Використовуються для відтворення звукових сповіщень, що покращує взаємодію з системою.
- Світлодіодні інформаційні панелі: Встановлюються для відображення інформації, що сприяє зручній навігації та спілкуванню з системою.

Опис автоматизованого робочого місця (АРМ)

- Системні вимоги: Використовуються ЕОМ з мінімальними системними вимогами, такими як Intel Core i5, 8 ГБ оперативної пам'яті та 256 ГБ на жорсткому диску.

- Периферійні засоби: Використовуються монітори, клавіатури, принтери та інші периферійні засоби, що забезпечують комфортні умови роботи для користувачів.
- Використання такого технічного забезпечення для web-сайту, що відслідковує раціон харчування та фізичні навантаження, спрямоване на забезпечення надійної та продуктивної роботи системи, а також зручного доступу користувачів до неї.

Протокол HTTP



Рисунок 3.2.2 – робота протоколу HTTP

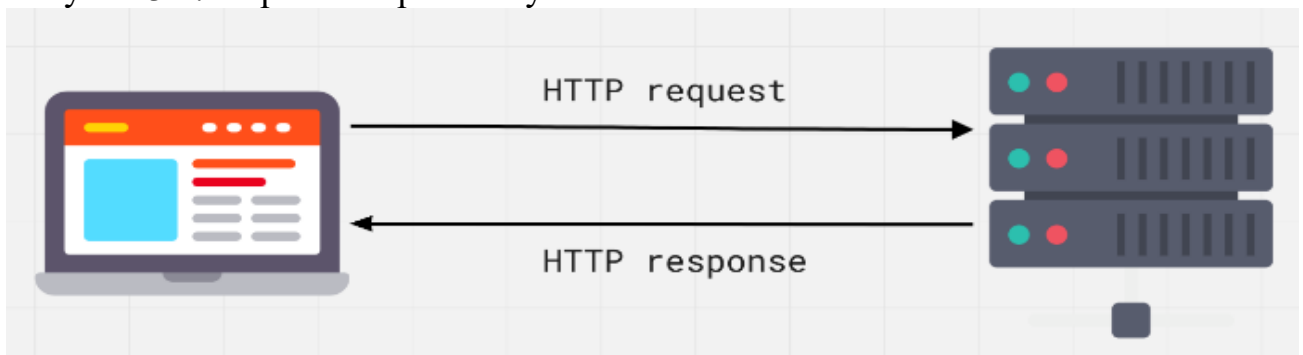


Рисунок 3.2.3 – робота протоколу HTTP

Протокол HTTP базується на структурі клієнт-сервер і моделі запит-відповідь, де клієнтський додаток ініціює з'єднання, формує запит і надсилає його на сервер, після чого сервер обробляє цей запит, формує відповідь і передає її

назад клієнту. Зв'язок між ними здійснюється за допомогою черги чергуючихся HTTP-запитів і HTTP-відповідей.

3.3 Програмне забезпечення

Розділ із програмним забезпеченням для проекту ІС "розробки web-застосунку з відстеження харчової поведінки і фізичної активності" включає різноманітні компоненти, що допомагають забезпечити його ефективну та стабільну роботу.

Системне програмне забезпечення, яке використовується у цьому проєкті, включає такі складові:

- **Операційна система (ОС)** - для розробки та експлуатації web-сайту використовується операційна система Windows. Вона забезпечує надійну та стабільну роботу системи, а також сумісність з іншими необхідними програмами та інструментами, що є важливим аспектом для забезпечення продуктивної розробки та безперебійної роботи web-додатку.
- **Draw.IO:** - це інструмент для створення діаграм та схем, який використовується для документації та візуалізації концепцій системи. З його допомогою можна створювати графічні представлення архітектури проєкту, взаємозв'язків компонентів та інших важливих аспектів системи, що дозволяє зрозуміло представляти та комунікувати структуру проєкту всім учасникам.
- **Figma:** - це інструмент для дизайну інтерфейсів, який використовувався для створення макетів web-сайту. Він надає можливість детально візуалізувати та протестувати дизайн елементів інтерфейсу перед їхнім реалізацією, сприяючи ефективній розробці та вдосконаленню користувацького досвіду.
- **HTML (HyperText Markup Language)** - це стандартна мова розмітки для створення web-сторінок та web-додатків. Вона використовується для визначення

структури та вмісту web-сторінок за допомогою елементів та тегів. Основні компоненти HTML включають:

- **CSS (Cascading Style Sheets)** - це мова стилів, яка використовується для опису вигляду та форматування документа, написаного на HTML. CSS визначає, як HTML-елементи повинні відображатися на екрані, папері або в інших медіа. Основні компоненти CSS включають:

- **JavaScript** - це мова програмування, яка використовується для створення динамічного та інтерактивного контенту на web-сторінках. JavaScript дозволяє змінювати HTML та CSS на стороні клієнта, реагувати на події користувача, виконувати запити до серверу та багато іншого. Основні компоненти JavaScript включають:

- **PHP (Hypertext Preprocessor)** - це серверна мова програмування, яка використовується для створення динамічних web-сторінок. Вона дозволяє взаємодіяти з базами даних, обробляти форми, генерувати контент на сервері та багато іншого.

- **MySQL** - це система керування базами даних (СКБД), яка використовує мову структурованих запитів (SQL) для керування даними. MySQL є однією з найпопулярніших баз даних і часто використовується в поєднанні з PHP для створення web-додатків.

- **phpMyAdmin** - це web-інтерфейс для адміністрування MySQL, написаний на PHP. Він дозволяє користувачам взаємодіяти з базами даних

3.4 Результати реалізації інформаційної підсистеми

Моделювання користувацького інтерфейсу (UI) є важливим етапом розробки web-застосунку з відстеження харчової поведінки і фізичної активності. На цьому етапі створюється візуальне уявлення того, як користувачі взаємодіятимуть із системою, які елементи інтерфейсу будуть використані, та яким чином буде організована навігація між різними екранами.

Розробка структури інтерфейсу у вигляді ієрархії екранів

Ієрархія екранів є основою для проектування користувацького інтерфейсу. Вона визначає, як різні екрани взаємопов'язані між собою і яким чином користувач переходить від одного екрану до іншого. Ієрархія екранів допомагає зрозуміти загальну структуру інтерфейсу і забезпечити логічний та інтуїтивно зрозумілий шлях для користувачів.

Приклад ієрархії екранів:

- Головний екран
- Екран реєстрації/логіна
- Екран домашньої сторінки
- Екран перегляду калорійності продуктів
- Екран перегляду раціону
- Екран додавання фізичних навантажень
- Екран графіку перегляду фізичних навантажень
- Екран налаштувань
- Екран профілю користувача

2. Опис призначення кожної екранної форми

Кожна екранна форма має своє призначення і відповідає за певну функціональність у системі. Важливо детально описати, що відображається на

кожному екрані, які дії може виконувати користувач, і як ці дії впливають на дані системи.

Опису призначення екранних форм:

- **Головний екран:** Початковий екран застосунку, що містить навігаційне меню та основні опції для користувача.
- **Екран реєстрації/логіна:** Дозволяє новим користувачам зареєструватися, а існуючим користувачам — увійти в систему.
- **Екран домашньої сторінки:** Центральний екран, з якого користувач може перейти до додавання продуктів, перегляду раціону, додавання фізичних навантажень та перегляду фізичних навантажень.
- **Екран перегляду калорійності продуктів:** дозволяє користувачеві переглянути та вибрати їжу та калорійність даної їжі для свого раціону.
- **Екран перегляду раціону:** Відображає поточний раціон користувача, включаючи всі додані продукти та їхні калорійні значення.
- **Екран додавання фізичних навантажень:** Дозволяє користувачу додавати нові фізичні вправи, вказуючи тривалість та кількість спалених калорій.
- **Екран графіку перегляду фізичних навантажень:** Відображає всі фізичні вправи, додані користувачем, та їхній вплив на загальну кількість спалених калорій.
- **Екран налаштувань:** Містить опції для налаштування параметрів застосунку.
- **Екран профілю користувача:** Відображає інформацію про користувача, включаючи його персональні дані та статистику використання.

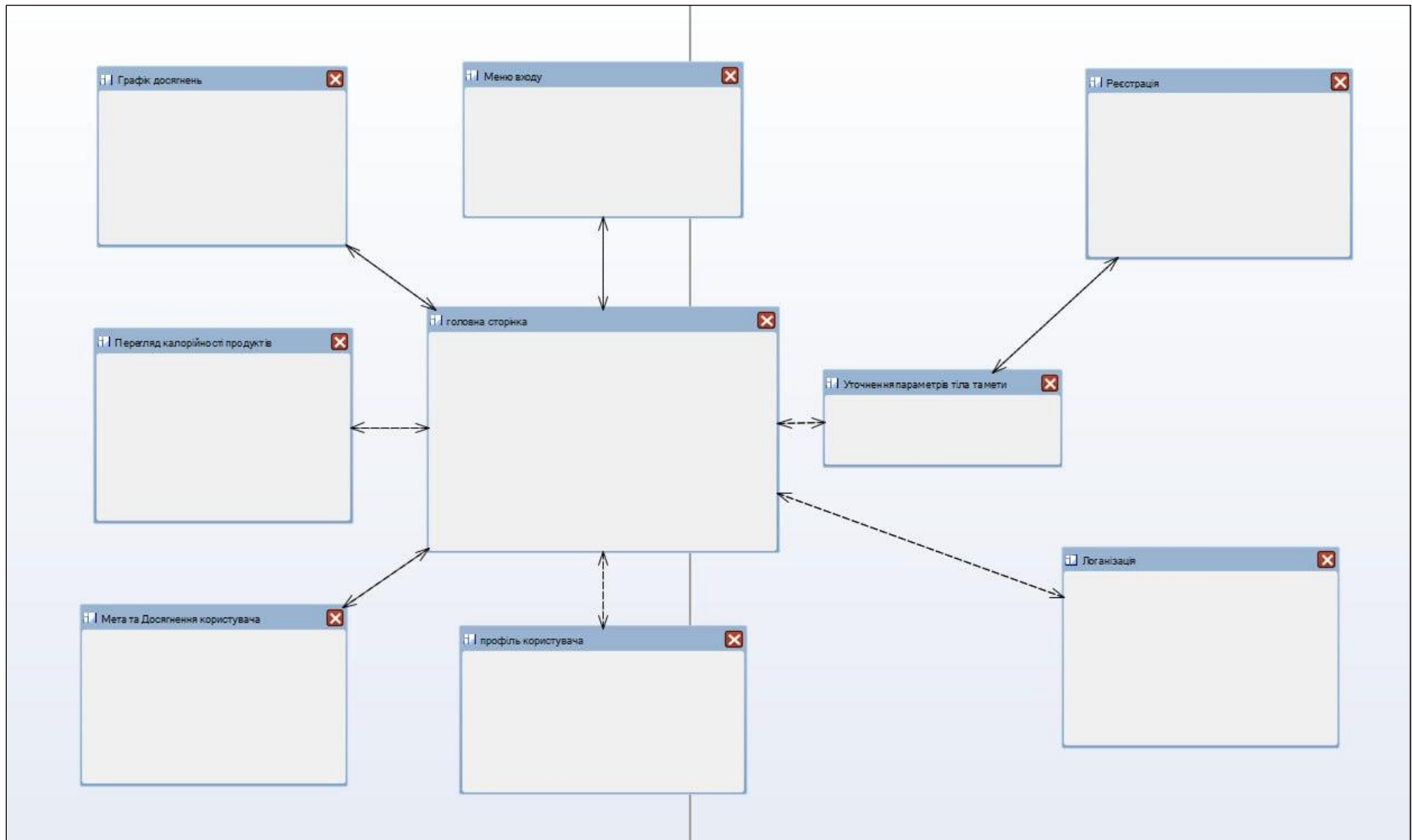


Рисунок 3.4.1 - Діаграма UI системи – ієрархія екранних форм

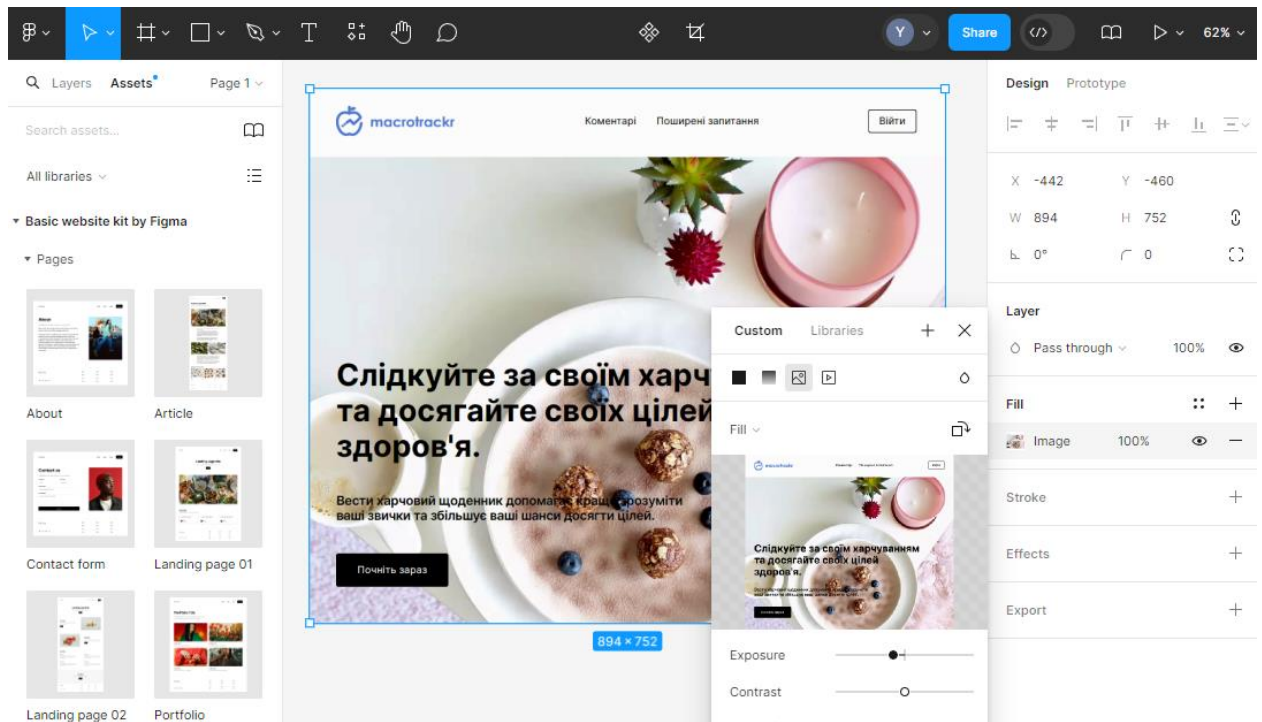


Рисунок 3.4.2 - UI макет figma головного екрану

Приклад прототипу екрану головного екрану

- Форма головного екрану:
- Вітальний текст або зображення.
- Меню навігації для доступу до основних функцій (реєстрація, логін, коментарів, запитань тощо).
- Кнопка для швидкого переходу до екрану реєстрації/логіна.
- Інформаційні повідомлення або новини.

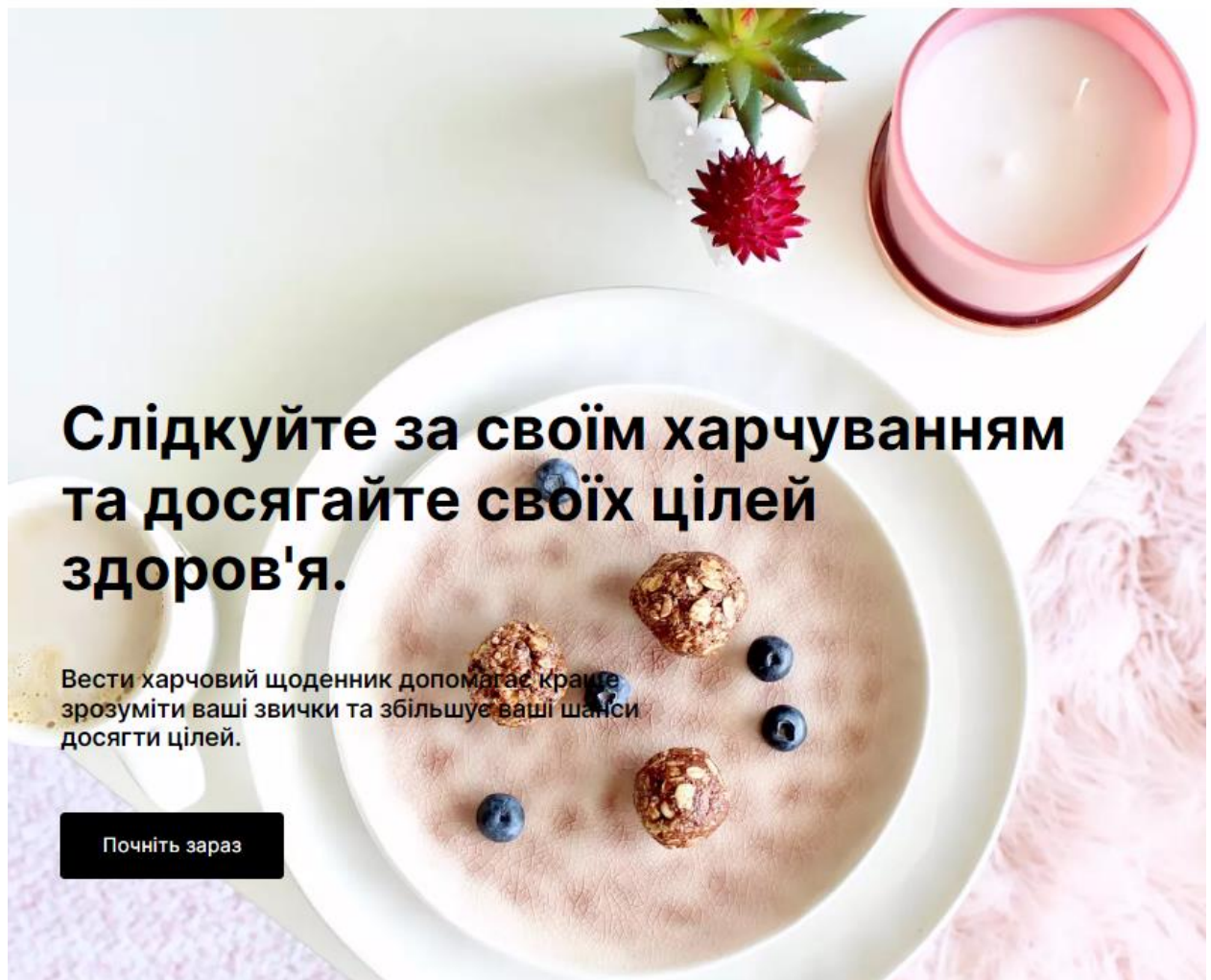
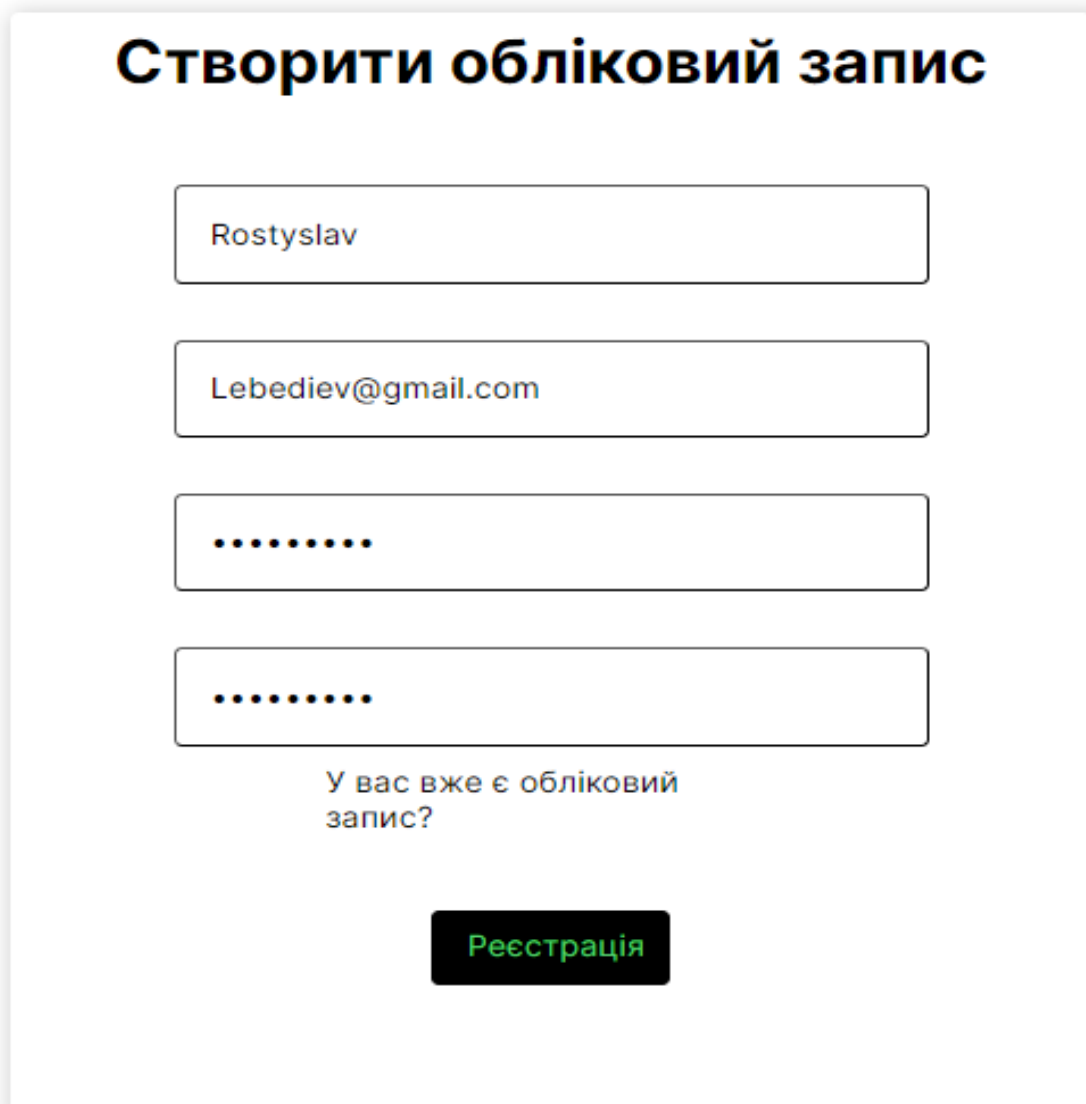


Рисунок 3.4.3 екран головний меню

Приклад прототипу екрану реєстрації

Форма реєстрації:

- Поля для введення імені, електронної пошти, паролю та підтвердження паролю.
- Кнопка для завершення реєстрації.



The image shows a registration form prototype with a white background and a subtle drop shadow. At the top, the title "Створити обліковий запис" (Create account) is displayed in a bold, black, sans-serif font. Below the title are four input fields, each with a thin black border. The first field contains the text "Rostyslav". The second field contains the email address "Lebediev@gmail.com". The third and fourth fields contain ten black dots, representing a password and its confirmation. Below these fields, the text "У вас вже є обліковий запис?" (Do you already have an account?) is centered. At the bottom, there is a black rectangular button with the word "Реєстрація" (Registration) in green text.

Створити обліковий запис

Rostyslav

Lebediev@gmail.com

.....

.....

У вас вже є обліковий запис?

Реєстрація

Рисунок 3.4.4 екран поля реєстрації

Приклад прототипу екрану логіна

Форма логіна:

- Поля для введення електронної пошти та паролю.
- Кнопка для входу в систему.
- Посилання на форму створення нового аккаунту.

The image shows a login form prototype on a white background. At the top, the word "Увійти" (Login) is displayed in a large, bold, black font. Below it, there are two input fields. The first field is a white rectangle with a thin black border, containing the email address "Rostyslav@gmail.com". The second field is a light blue rectangle with a thin black border, containing ten black dots representing a password. Below the password field, the text "У вас немає облікового запису?" (Do you not have an account?) is centered in a smaller black font. At the bottom, there is a black rectangular button with the word "Підключення" (Connection) written in green text.

Рисунок 3.4.5 екран поля логонізації

Приклад прототипу екрану домашньої сторінки

- Форма домашньої сторінки:
- Вітальний текст з іменем користувача.
- Кнопки для переходу до різних розділів (перегляд раціону, фізичні навантаження, профіль, налаштування тощо).
- • Список досягнень або статистика активності.
- Останні оновлення або активності користувача.
- Кнопка для виходу з профілю.

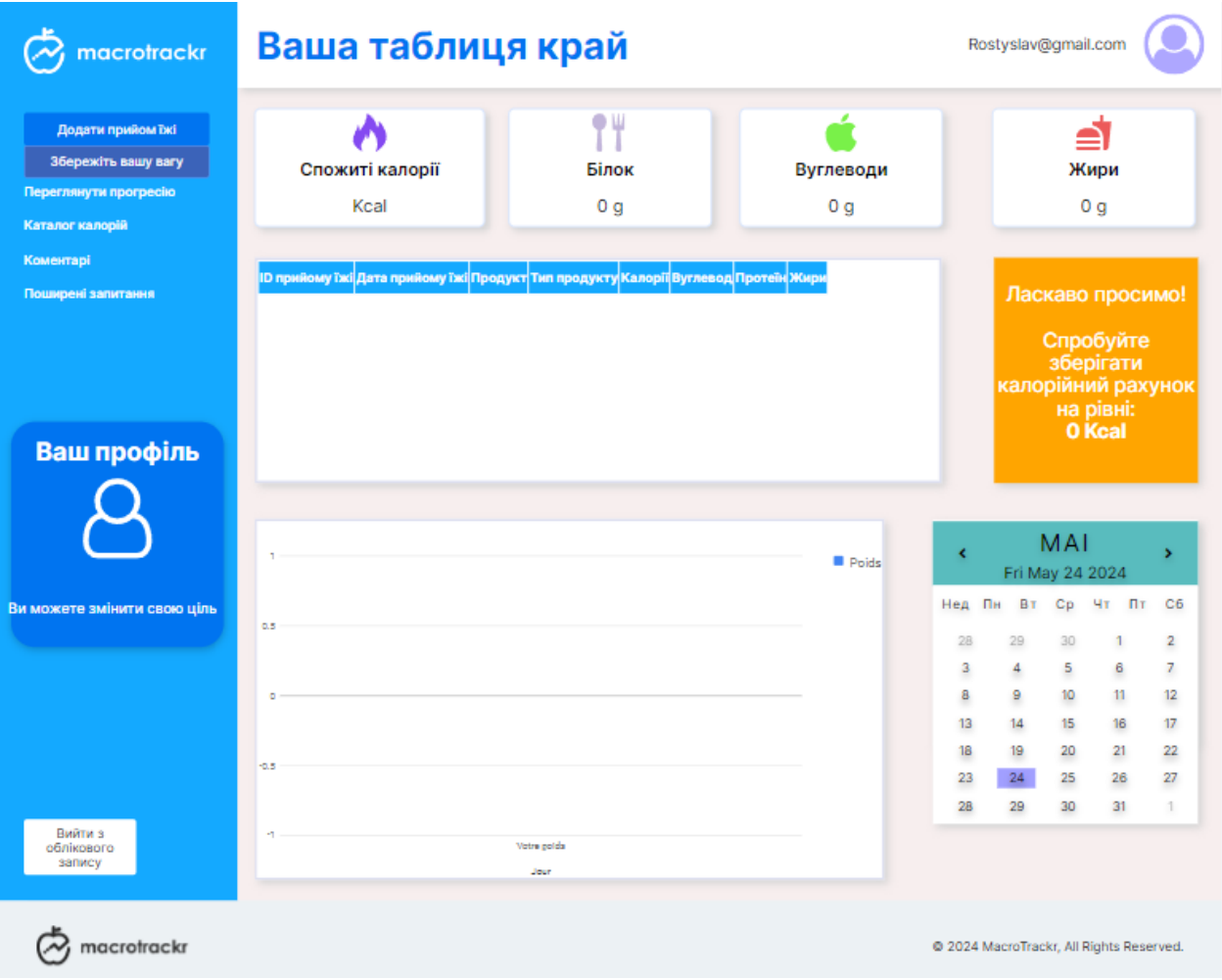


Рисунок 3.4.6 екран домашньої сторінки

Приклад прототипу екрану перегляду калорійності продуктів

- Форма перегляду калорійності продуктів:
- Поле для введення назви продукту для пошуку.
- Список продуктів з їх калорійністю та поживними характеристиками.
- Фільтри для сортування продуктів за різними параметрами (назва, калорійність тощо).
- Кнопка для повернення до попереднього екрану.

Кількість калорій у продуктах

Введіть назву для пошуку конкретного продукту:

Ікра

(Не знайшли те, що шукали? - Введіть свій власний персоналізований прийом їжі тут)

Продукт	Порція	Калорії	Протеїн	Вуглеводи	Жири
Ікра білуги зерниста	100g	237 Kcal	27.2 g	0.0 g	14.2 g
Ікра горбуші зерниста	100g	230 Kcal	31.2 g	0.0 g	11.7 g
Ікра кети зерниста	100g	251 Kcal	31.6 g	0.0 g	13.8 g
Ікра мінтая пробійна	100g	131 Kcal	28.4 g	0.0 g	1.9 g
Ікра осетрова зерниста	100g	203 Kcal	28.0 g	0.0 g	9.7 g
Ікра осетрова паюсна	100g	236 Kcal	36.0 g	0.0 g	10.2 g
Ікра севрюги зерниста	100g	221 Kcal	28.4 g	0.0 g	11.9 g
Ікра тріски	100g	115 Kcal	24.0 g	0.0 g	0.2 g
Балик осетровий	100g	194 Kcal	20.4 g	0.0 g	12.5 g
Баранина відварена	100g	243 Kcal	22.0 g	0.0 g	17.2 g
Баранина тушкована, консерва	100g	191 Kcal	17.4 g	0.2 g	13.4 g
Баранина, лопатка	100g	284 Kcal	15.6 g	0.0 g	25.0 g
Баранина, окіст	100g	232 Kcal	18.0 g	0.0 g	18.0 g
Бефстроганов	100g	220 Kcal	19.5 g	3.7 g	14.3 g
Биточки яловичі парові	100g	194 Kcal	14.0 g	8.2 g	11.6 g
Білуга бланшована	100g	234 Kcal	23.3 g	0.0 g	15.6 g
Білуга свіжа	100g	147 Kcal	21.0 g	0.0 g	0.7 g
Біфштекс яловичий	100g	214 Kcal	28.8 g	0.0 g	11.0 g

Рисунок 3.4.7 екран перегляду калорійності продуктів

Приклад прототипу екрану налаштувань профілю

- Форма налаштувань (стать, вік, зріст, вага, активні щодня, ціль і т.д.)
- Кнопка для збереження змін.
- Кнопка для скасування змін.

Розрахувати вашу калорійну норму

Для точного обчислення нам потрібна деяка базова інформація про вас

Яка ваша стать?



☐ Чоловік



☐ Жінка

Скільки вам років?

Вік

Якого ви зросту?

Cm

Скільки ви важите?

Kg

Наскільки ви активні щодня?



Не дуже активний
Ви проводите більшу частину дня у сидінному положенні.
(наприклад: офісна робота)



Помірно активний
Ви проводите багато часу у стоячому положенні протягом дня.
(наприклад: вчитель)



Активний
Ви проводите значну частину дня займаючись фізичною активністю.
(наприклад: офісiant)



Дуже активний
Ви проводите значну частину дня виконуючи дуже фізичні активності.
(наприклад: кур'єр на велосипеді)

Яка ваша ціль?



Схуднути
Ваша нова дієта допоможе вам схуднути швидше.



Підтримувати вагу
Ваша нова дієта дозволить вам підтримувати свою поточну вагу.



Набрати вагу
Ваша нова дієта дозволить вам мати калорійний надлишок і набрати вагу.

Розрахувати вашу денну калорійну норму

Будь ласка, переконайтеся, що введені вами значення є правильними :

Ваша калорійна потреба

Зберегти

Рисунок 3.4.8 екран налаштувань профілю

ВИСНОВКИ

У процесі виконання дипломної роботи було створено web-застосунок, призначений для моніторингу раціону харчування та фізичної активності користувачів. Основною метою проекту було розроблення зручного та функціонального інструменту, який дозволяє відслідковувати споживання калорій та фізичні навантаження, а також отримувати візуалізацію даних у вигляді графіків та звітів.

У рамках розробки проекту було проведено наступні кроки:

- **Аналіз вимог:** Були визначені та документовані функціональні та нефункціональні вимоги до системи. Відбувся детальний аналіз існуючих аналогів, що дозволив виокремити унікальні особливості проекту.
- **Проектування системи:** Розроблено структуру бази даних та визначено основні компоненти системи та їх взаємодію. Створено моделі та діаграми, що ілюструють архітектуру та логіку роботи web-застосунку.
- **Розробка користувацького інтерфейсу:** Створено прототипи екранних форм та визначено структуру інтерфейсу. Була забезпечена зручність та інтуїтивність користувацького досвіду.
- **Тестування та верифікація:** Проведено модульні, інтеграційні, системні та приймальні тести, а також тести-сценарії. Всі тести були підтверджені успішністю, що дозволило перевірити відповідність реалізації заявленим вимогам.

Проект відрізняється наступними особливостями та перевагами прийнятих рішень:

- **Інтуїтивний інтерфейс:** Простий та зрозумілий інтерфейс, що забезпечує зручність використання для користувачів будь-якого рівня технічної підготовки.

- Розширена функціональність: Великий набір функцій для моніторингу харчування та фізичної активності, включаючи введення та збереження даних, генерування звітів та графіків.
- Тестування та верифікація: Глибоке тестування модулів системи забезпечило високу якість кінцевого продукту.

З метою подальшого розвитку та впровадження у практику рекомендується:

- Розширення функціоналу: Додавання нових функцій для більш детального моніторингу здоров'я користувачів, таких як відстеження споживання макро- та мікроелементів.
- Мобільна версія: Розробка мобільної версії застосунку для забезпечення доступу до системи з будь-якого пристрою.
- Інтеграція з іншими сервісами: Можливість інтеграції з іншими сервісами для автоматичного отримання даних про фізичну активність та харчування.

Виконана дипломна робота показує успішну реалізацію проекту від аналізу вимог до створення прототипу та його тестування. Розроблений web-застосунок має значний потенціал для подальшого розвитку та впровадження у практику, що сприятиме покращенню якості життя користувачів.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. W3Schools [Електронний ресурс] – Режим доступу до ресурсу: <https://www.w3schools.com/>.
2. MDN Web Docs (JavaScript) [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/ru/docs/Web/JavaScript>.
3. Bootstrap [Електронний ресурс] – Режим доступу до ресурсу: <https://getbootstrap.com/>.
4. php.net [Електронний ресурс] – Режим доступу до ресурсу: <https://www.php.net/>.
5. MySQL Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://dev.mysql.com/doc/>.
6. phpMyAdmin [Електронний ресурс] – Режим доступу до ресурсу: <https://www.phpmyadmin.net/>.
7. Stack Overflow [Електронний ресурс] – Режим доступу до ресурсу: <https://stackoverflow.com/>.
8. CSS-Tricks [Електронний ресурс] – Режим доступу до ресурсу: <https://css-tricks.com/>.
9. JavaScript.info [Електронний ресурс] – Режим доступу до ресурсу: <https://javascript.info/>.
10. Mozilla Developer Network (MDN) [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/>.
11. HTML Dog [Електронний ресурс] – Режим доступу до ресурсу: <https://htmldog.com/>.
12. TutorialsPoint [Електронний ресурс] – Режим доступу до ресурсу: <https://www.tutorialspoint.com/>.
13. PHP The Right Way [Електронний ресурс] – Режим доступу до ресурсу: <https://phptherightway.com/>.
14. MySQL Tutorial [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mysqltutorial.org/>.

- 15.Bootstrapious [Электронный ресурс] – Режим доступа до ресурсу: <https://bootstrapious.com/>.
- 16.DevDocs [Электронный ресурс] – Режим доступа до ресурсу: <https://devdocs.io/>.
- Codecademy [Электронный ресурс] – Режим доступа до ресурсу: <https://www.codecademy.com/>.
- 17.W3C Markup Validation Service [Электронный ресурс] – Режим доступа до ресурсу: <https://validator.w3.org/>.
- 18.CSS Gradient [Электронный ресурс] – Режим доступа до ресурсу: <https://cssgradient.io/>.
- 19.PHP.net (MySQLi Documentation) [Электронный ресурс] – Режим доступа до ресурсу: <https://www.php.net/manual/en/book.mysqli.php>.
- 20.MySQL Documentation (MySQLi API) [Электронный ресурс] – Режим доступа до ресурсу: <https://dev.mysql.com/doc/apis/en/>.
- 21.Stack Exchange [Электронный ресурс] – Режим доступа до ресурсу: <https://stackoverflow.com/>.

ДОДАТКИ

```
<?php
@include 'config.php';
session_start();

$email = $_SESSION['email'];

$resultp= mysqli_query($conn,"SELECT * FROM weight_log where
email='$email' order by day; ");


$query = "SELECT id, count(*) as number FROM users ORDER BY id";
$result = mysqli_query($conn, $query);


if(!isset($_SESSION['email'])){
    header('location:index.php');
}
?>

<!DOCTYPE html>

<html lang="uk">
<head>
    <title>Панель управління</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta charset="utf-8" />
    <style data-tag="reset-style-sheet">
        html { line-height: 1.15;}body { margin: 0;}* { box-sizing: border-
        box; border-width: 0; border-style: solid;}p,li,ul,pre,div,h1,h2,h3,h4,h5,h6
        { margin: 0; padding: 0;}button,input,optgroup,select,textarea { font-family:
        inherit; font-size: 100%; line-height: 1.15; margin: 0;}button,select { text-
        transform: none;}button,[type="button"],[type="reset"],[type="submit"] { -
        webkit-appearance: button;}button::-moz-focus-inner,[type="button"]::-moz-
```

```
focus-inner,[type="reset"]::-moz-focus-inner,[type="submit"]::-moz-focus-inner
{ border-style: none; padding: 0;}button:-moz-focus,[type="button"]:-moz-
focus,[type="reset"]:-moz-focus,[type="submit"]:-moz-focus { outline: 1px dotted
ButtonText;}a { color: inherit; text-decoration: inherit;}input { padding: 2px
4px;}img { display: block;}html { scroll-behavior: smooth }
```

```
</style>
```

```
<style data-tag="default-style-sheet">
```

```
html {
  font-family: Inter;
  font-size: 16px;
}
```

```
body {
  font-weight: 400;
  font-style:normal;
  text-decoration: none;
  text-transform: none;
  letter-spacing: normal;
  line-height: 1.15;
  color: var(--dl-color-gray-black);
  background-color: var(--dl-color-gray-white);
}
```

```
#table{

  table-layout: fixed;
  width: 100%;
  border-collapse: collapse;
  border: 3px solid var(--dl-color-primary-500);
```

```
}
```

```
#table td, #table th {  
    border: 1px solid #ddd;  
    padding: 20px;
```

```
}
```

```
#table tr:nth-child(even){background-color: #f2f2f2;}
```

```
#table tr:hover {background-color: #8FB7D3;}
```

```
#table th {  
    padding-top: 12px;  
    padding-bottom: 12px;  
    text-align: left;  
    background-color: var(--dl-color-primary-500);  
    color: white;  
}
```

```
tr{transition:all .25s ease-in-out}  
tr:hover{background-color:#3B93D6;cursor: pointer}  
td {  
    width: 200px;  
    height: 50px;  
    align-content: center;  
    text-align: center;  
    background-color: aliceblue;  
}
```

```
thead {  
  background-color: var(--dl-color-primary-500);  
  height: 40px;  
  color: #ffffff;  
}  
  
.main-dashboard-protein {  
  top: 133px;  
  flex: 0 0 auto;  
  left: 640px;  
  width: 256px;  
  height: 150px;  
  display: flex;  
  z-index: 1;  
  position: absolute;  
  box-shadow: 5px 5px 10px 0px #d4d4d4;  
  transition: 0.3s;  
  align-items: flex-start;  
  border-color: #DAE7FE;  
  border-style: solid;  
  border-width: 2px;  
  border-radius: var(--dl-radius-radius-radius8);  
  flex-direction: column;  
  background-color: #ffffff;  
}  
  
.main-dashboard-carbs {  
  top: 133px;  
  flex: 0 0 auto;  
  left: 930px;  
  width: 256px;
```

```
height: 150px;
display: flex;
z-index: 1;
position: absolute;
box-shadow: 5px 5px 10px 0px #d4d4d4;
transition: 0.3s;
align-items: flex-start;
border-color: #DAE7FE;
border-style: solid;
border-width: 2px;
border-radius: var(--dl-radius-radius-radius8);
flex-direction: column;
background-color: #ffffff;
}
```

```
.main-dashboard-fats {
top: 133px;
flex: 0 0 auto;
right: 33px;
width: 256px;
height: 150px;
display: flex;
z-index: 1;
position: absolute;
box-shadow: 5px 5px 10px 0px #d4d4d4;
transition: 0.3s;
align-items: flex-start;
border-color: #DAE7FE;
border-style: solid;
border-width: 2px;
```

```
border-radius: var(--dl-radius-radius-radius8);
flex-direction: column;
background-color: #ffffff;
}
```

```
</style>
```

```
<link
```

```
  rel="shortcut icon"
```

```
  href="/playground_assets/favicon-32x32.png"
```

```
  type="icon/png"
```

```
  sizes="32x32"
```

```
<link
```

```
  rel="stylesheet"
```

```
  href="https://fonts.googleapis.com/css2?family=Inter:wght@100;200;300;400;500;600;700;800;900&display=swap"
```

```
  data-tag="font"
```

```
<link rel="stylesheet" href="./style.css" />
```

```
</head>
```

```
<body>
```

```
<div>
```

```
  <link href="./main-dashboard.css" rel="stylesheet" />
```

```
<div class="main-dashboard-container">
```

```
  <svg viewBox="0 0 1024 1024" class="main-dashboard-icon">
```

```
    <path
```

```
      d="M512 0c282.857 0 512 229.143 512 512 0 281.143-228 512-512 512-283.429 0-512-230.286-512-512 0-282.857 229.143-512 512-512zM865.714 772c53.143-73.143 85.143-162.857 85.143-260 0-241.714-197.143-438.857-
```

438.857-438.857s-438.857 197.143-438.857 438.857c0 97.143 32 186.857 85.143
260 20.571-102.286 70.286-186.857 174.857-186.857 46.286 45.143 109.143
73.143 178.857 73.143s132.571-28 178.857-73.143c104.571 0 154.286 84.571
174.857 186.857zM731.429 402.286c0-121.143-98.286-219.429-219.429-
219.429s-219.429 98.286-219.429 219.429 98.286 219.429 219.429 219.429
219.429-98.286 219.429-219.429z"

></path>

</svg>

<?php echo \$_SESSION['email'];
?>

<div class="main-dashboard-meals">

<!-- <h1 class="main-dashboard-text01">Entrées de nourriture</h1>-->

<table class="table" value="<? \$table ?>">

<thead>

<tr>

<th scope="col">ID прийому їжі</th>

<th scope="col">Дата прийому їжі</th>

<th scope="col">Продукт</th>

<th scope="col">Тип продукту</th>

<th scope="col">Калорії</th>

<th scope="col">Вуглевод</th>

<th scope="col">Протеїн</th>

<th scope="col">Жири</th>

</tr>

</thead>

<tbody>

<?php

\$email=\$_SESSION['email'];

\$sql= "SELECT * FROM add_meal where date_repas=CURRENT_DATE()
and email='\$email'";

```
$result=mysqli_query($conn,$sql);
```

```
if(!('date_repas'==date('yyyy-mm-dd'))){
```

```
if($result){
```

```
while( $row=mysqli_fetch_assoc($result)){
```

```
    $id_repas=$row['id'];
```

```
    $date_repas=$row['date_repas'];
```

```
    $aliment=$row['aliment'];
```

```
    $type_alim=$row['type_alim'];
```

```
    $calories=$row['calories'];
```

```
    $glucide=$row['glucide'];
```

```
    $proteine=$row['proteine'];
```

```
    $lipides=$row['lipides'];
```

```
    echo'<tr>
```

```
        <th scope="row">'.$id_repas.'</th>
```

```
        <td>'.$date_repas.'</td>
```

```
        <td>'.$aliment.'</td>
```

```
        <td>'.$type_alim.'</td>
```

```
        <td>'.$calories.' Kcal</td>
```

```
        <td>'.$glucide.' g</td>
```

```
        <td>'.$proteine.' g</td>
```

```
        <td>'.$lipides.' g</td>
```

```
    </tr>';
```

```
}
```

```

    }
}

?>
</tbody>
</table>

</div>
<h1 class="main-dashboard-text02">
    <span>Ваша таблиця</span>
    <span></span>
    <span>край</span>
    <span></span>
</h1>
<div class="main-dashboard-welcome">
    <span class="main-dashboard-text07">
        <span class="main-dashboard-text08">
            Ласкаво просимо!<br>
        </span>
    <br />
    <span class="main-dashboard-text10">
        Спробуйте зберігати калорійний рахунок на рівні: <br>
        <b><?php
            $email=$_SESSION['email'];
            $sql="SELECT * FROM infopers WHERE email='$email'";
            $intake=mysqli_query($conn,$sql);
            if(mysqli_num_rows($intake) > 0){
                $row=mysqli_fetch_array($intake);

```

```
$apport=$row['calo'];
```

```
echo "$apport Kcal";
```

```
} else{
```

```
echo "0 Kcal";
```

```
}
```

```
?></b>
```

```
</span>
```

```
</span>
```

```
</div>
```

```
<a href="profile.php">
```

```
<div class="main-dashboard-profile">
```

```
<h1 class="main-dashboard-text11">
```

```
<span>Ваш профіль</span>
```

```
<span class="main-dashboard-text13"></span>
```

```
</h1>
```

```
<svg
```

```
viewBox="0 0 877.7142857142857 1024"
```

```
class="main-dashboard-icon02"
```

```
>
```

```
<path
```

```
d="M686.286 448c80.571 23.429 191.429 102.857 191.429 362.857 0
117.714-87.429 213.143-194.857 213.143h-488c-107.429 0-194.857-95.429-
194.857-213.143 0-260 110.857-339.429 191.429-362.857-28.571-45.143-45.143-
98.286-45.143-155.429 0-161.143 131.429-292.571 292.571-292.571s292.571
131.429 292.571 292.571c0 57.143-16.571 110.286-45.143 155.429zM438.857
```

73.143c-121.143 0-219.429 98.286-219.429 219.429s98.286 219.429 219.429
219.429 219.429-98.286 219.429-219.429-98.286-219.429-219.429-
219.429zM682.857 950.857c66.857 0 121.714-62.286 121.714-140 0-180-60.571-
292.571-173.714-298.286-51.429 45.143-118.286 72.571-192 72.571s-140.571-
27.429-192-72.571c-113.143 5.714-173.714 118.286-173.714 298.286 0 77.714
54.857 140 121.714 140h488z"

></path>

</svg>

Ви можете змінити свою ціль

</div>

<div id="columnchart_material" class="main-dashboard-progress">

<h1 class="main-dashboard-text15">Ваші досягнення</h1>

<script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>

<script type="text/javascript">

google.charts.load('current', {'packages':['bar']});

google.charts.setOnLoadCallback(drawChart);

function drawChart() {

var data = google.visualization.arrayToDataTable([

['Jour','Poids'],

['Votre poids',0],

<?php

if(mysqli_num_rows(\$resultp)>0){

while(\$row =mysqli_fetch_array(\$resultp)){

echo "[".\$row['day'].",".\$row['poids']."]";

}

```
}  
?>
```

```
]);
```

```
var options = {  
  chart: {  
    title: '',
```

```
  }  
};
```

```
var chart = new  
google.charts.Bar(document.getElementById('columnchart_material'));
```

```
chart.draw(data, google.charts.Bar.convertOptions(options));  
}  
</script>  
</div>
```

```
<link  
rel="stylesheet"  
href="https://cdnjs.cloudflare.com/ajax/libs/font-  
awesome/5.13.0/css/all.min.css"  
/>  
  
<link  
href="https://fonts.googleapis.com/css2?family=Quicksand:wght@300;400;50  
0;600;700&display=swap"
```

rel="stylesheet"

/>

<div class="main-dashboard-calendar">

<div class="calendar">

<div class="month">

<i class="fas fa-angle-left prev"></i>

<div class="date">

<h1></h1>

<p></p>

</div>

<i class="fas fa-angle-right next"></i>

</div>

<div class="weekdays">

<div>Нед</div>

<div>Пн</div>

<div>Вт</div>

<div>Ср</div>

<div>Чт</div>

<div>Пт</div>

<div>Сб</div>

</div>

<div class="days"></div>

</div>

</div>

<script src="calendar.js"></script>

<div class="main-dashboard-header"></div>

```

<div class="main-dashboard-calories">

    <svg viewBox="0 0 1024 1024" class="main-dashboard-icon04">

        <path

            d="M321.008 1024c-68.246-142.008-31.902-223.378 20.55-300.044
57.44-83.956 72.244-167.066 72.244-167.066s45.154 58.7 27.092
150.508c79.772-88.8 94.824-230.28 82.782-284.464 180.314 126.012 257.376
398.856 153.522 601.066 552.372-312.532 137.398-780.172 65.154-832.85
24.082 52.676 28.648 141.85-20 185.126-82.352-312.276-285.972-376.276-
285.972-376.276 24.082 161.044-87.296 337.144-194.696 468.73-3.774-64.216-
7.782-108.528-41.55-169.98-7.58 116.656-96.732 211.748-120.874 328.628-
32.702 158.286 24.496 274.18 241.748 396.622z"

            ></path>

        </svg>

        <h1 class="macros_title"><span>Спожиті калорії</span></h1>

        <span class="main-dashboard-text18 macros">

            <?php

                $email=$_SESSION['email'];

                $sql= "SELECT SUM(calories) as sum FROM add_meal where
date_repas=CURRENT_DATE() and email='$email'";

                $result=mysqli_query($conn,$sql);

                if(mysqli_num_rows($result) > 0){

                    while($row=mysqli_fetch_assoc($result)){

                        $output=$row['sum'];

                    }

                    echo "$output Kcal";

                } else{

                    echo "0 Kcal";

                }

            ?></span>

```

</div>

<!--macros----->

<div class="main-dashboard-protein">

<svg viewBox="0 0 1024 1024" class="main-dashboard-icon06 icon">

<path

d="M224 0c-106.040 0-192 100.288-192 224 0 105.924 63.022 194.666
147.706 217.998l-31.788 518.124c-2.154 35.132 24.882 63.878 60.082
63.878h32c35.2 0 62.236-28.746 60.082-63.878l-31.788-518.124c84.684-23.332
147.706-112.074 147.706-217.998 0-123.712-85.96-224-192-224zM869.334 0l-
53.334 320h-40l-26.666-320h-26.668l-26.666 320h-40l-53.334-320h-
26.666v416c0 17.672 14.326 32 32 32h83.338l-31.42 512.122c-2.154 35.132
24.882 63.878 60.082 63.878h32c35.2 0 62.236-28.746 60.082-63.878l-31.42-
512.122h83.338c17.674 0 32-14.328 32-32v-416h-26.666z"

></path>

</svg>

<h1 class="macros_title">Білок</h1>

<?php

```
$email=$_SESSION['email'];
```

```
$sql= "SELECT * FROM add_meal where  
date_repas=CURRENT_DATE() and email='$email';
```

```
$result=mysqli_query($conn,$sql);
```

```
if(mysqli_num_rows($result) > 0){
```

```
    $totalprote=0;
```

```
    while($row=mysqli_fetch_array($result)){
```

```
        $proteine=$row['proteine'];
```

```
        $totalprote= $totalprote + $proteine;
```

```
    }
```

```
    echo "$totalprote g";
```

```

    } else{
        echo "0 g";
    }
?></span>
</div>
<div class="main-dashboard-carbs">
    <svg viewBox="0 0 1024 1024" class="main-dashboard-icon08 icon">
        <path
            d="M791.498 544.092c-1.294-129.682 105.758-191.876 110.542-
194.966-60.152-88.020-153.85-100.078-187.242-101.472-79.742-8.074-155.596
46.948-196.066 46.948-40.368 0-102.818-45.754-168.952-44.552-86.916 1.292-
167.058 50.538-211.812 128.38-90.304 156.698-23.126 388.84 64.89 515.926
43.008 62.204 94.292 132.076 161.626 129.58 64.842-2.588 89.362-41.958
167.756-41.958s100.428 41.958 169.050 40.67c69.774-1.296 113.982-63.398
156.692-125.796 49.39-72.168 69.726-142.038 70.924-145.626-1.548-0.706-
136.060-52.236-137.408-207.134zM662.562 163.522c35.738-43.358 59.86-
103.512 53.28-163.522-51.478 2.096-113.878 34.29-150.81 77.55-33.142 38.376-
62.148 99.626-54.374 158.436 57.466 4.484 116.128-29.204 151.904-72.464z"
        ></path>
    </svg>
    <h1 class="macros_title"><span>Вуглеводи</span></h1>
    <span class="main-dashboard-text24 macros"><?php

    $email=$_SESSION['email'];

    $sql= "SELECT * FROM add_meal where
date_repas=CURRENT_DATE() and email='$email'";

    $result=mysqli_query($conn,$sql);

    if(mysqli_num_rows($result) > 0){

    $totalgluc=0;

    while($row=mysqli_fetch_assoc($result)){

        $glucide=$row['glucide'];

        $totalgluc= $totalgluc + $glucide;

```

```

    }
    echo "$totalgluc g";
} else{
    echo "0 g";
}

?></span>

</div>

<div class="main-dashboard-fats">

    <svg viewBox="0 0 1024 1024" class="main-dashboard-icon10 icon">

        <path

            d="M44 726h640v84h-640v-84zM684 640h-642q0-98 66-164t160-82
189 0 161 82 66 164zM42 938v-42h642v42q0 18-13 30t-31 12h-554q-18 0-31-
12t-13-30zM770 980v-342q0-126-104-226-62-62-182-96l-12-100h212v-
174h84v174h214l-72 702q-4 26-23 44t-45 18h-72z"

            ></path>

        </svg>

<h1 class="macros_title"><span>Жири</span></h1>

<span class="main-dashboard-text27 macros"><?php

    $email=$_SESSION['email'];

    $sql= "SELECT *FROM add_meal where
date_repas=CURRENT_DATE() and email='$email'";

    $result=mysqli_query($conn,$sql);

    if(mysqli_num_rows($result) > 0){

        $totallipide=0;

        while($row=mysqli_fetch_assoc($result)){

            $lipides=$row['lipides'];

            $totallipide= $totallipide + $lipides;

        }

```

```

        echo "$totallipide g";
    } else{
        echo "0 g";
    }
?></span>
</div>
<div class="main-dashboard-sidebar">
    <nav class="main-dashboard-nav">
        
        <a href="add-meal.php"><button class="main-dashboard-add-meal button
add">
            Додати прийом їжі
        </button></a>
        <a href="add-weight.php"><button class="main-dashboard-log-weight
button add">
            Збережіть вашу вагу
        </button></a>
        <span class="main-dashboard-text28 sider">
            <span></span>
            <a href="progress.php">Переглянути прогресію</a>
        </span>
        <span class="main-dashboard-text30 sider"><a href="catalogue-
calorique.php">Каталог калорій</a></span>
        <span class="main-dashboard-text31 sider"><a
href="about.php">Коментарі</a></span>

```

Поширені
запитання

</nav>

<button class="main-dashboard-logut button">Вийти
з облікового запису</button>

</div>

<footer class="main-dashboard-footer">

© 2024 MacroTrackr, All Rights Reserved.

</footer>

<div class="main-dashboard-background"></div>

</div>

</div>

</body>

</html>

Ім'я користувача:
Інформаційних систем в економіці Шкуратовська Те...

ID перевірки:
1016378637

Дата перевірки:
20.06.2024 17:31:36 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
20.06.2024 17:32:47 EEST

ID користувача:
100005745

Назва документа: Лебедєв Р.Д. ІН-401з_2

Кількість сторінок: 64 Кількість слів: 7326 Кількість символів: 58704 Розмір файлу: 5.25 MB ID файлу: 1016187202

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

25.7%
Схожість

Найбільша схожість: 17.8% з джерелом з Бібліотеки (ID файлу: 1015266987)

3.79% Джерела з Інтернету 329 Сторінка 66

25.1% Джерела з Бібліотеки 522 Сторінка 68

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 35

Підозріле форматування 16
сторінок