

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

ЛІВАКОВСЬКИЙ ВОЛОДИМИР КОСТЯНТИНОВИЧ

Допускається до захисту:

В.о. завідувача кафедри
інформаційних технологій
канд. техн. наук, доцент

_____ Оксана ЗЕЛІНСЬКА

«_____» _____ 2025р.

**РОЗРОБКА ВЕБ-ПЛАТФОРМИ ДЛЯ ПІДБОРУ ХАРЧУВАННЯ З
ІНТЕГРАЦІЄЮ ШТУЧНОГО ІНТЕЛЕКТУ**

Спеціальність 122 Комп'ютерні науки
Кваліфікаційна (бакалаврська) робота

Керівник:

Павло РИМАР, старший викладач
кафедри інформаційних технологій

Оцінка: _____ / _____ / _____
(бали/за шкалою ЄКТС/за національною шкалою)

Голова ЕК: _____

Вінниця – 2025

АНОТАЦІЯ

Ліваковський В.К. Розробка веб-платформи для підбору харчування з інтеграцією штучного інтелекту. Спеціальність 122 «Комп'ютерні науки», освітня програма «Комп'ютерні науки». Донецький національний університет імені Василя Стуса, Вінниця 2025.

У кваліфікаційній (бакалаврській) роботі розглянуто процес створення веб додатку для персонального управління харчуванням. Запропоновано архітектуру системи, розроблено механізм розрахунку калорійної потреби користувача та інтеграція з GPT моделлю для генерації індивідуальних порад. Використані технології HTML, CSS, Bootstrap, Jinja2, Flask, MySQL, OpenAI API. Реалізовано перегляд профіля, збереження параметрів, та добовий раціон.

Ключові слова: веб додаток, харчування, штучний інтелект, рекомендації, Flask, GPT, MySQL, HTML, CSS, Bootstrap.

61 с., 12 рис., 1 табл., 32 джерела.

ABSTRACT

Livakovskiy V.K. Development of a web platform for nutrition selection with artificial intelligence integration. Specialty 122 «Computer Science», educational program «Computer Science». Vasyl Stus Donetsk National University, Vinnytsia 2025.

This bachelor's thesis explores the development of a web application for personal nutrition management. The proposed system includes user-specific calorie calculation and integration with the GPT model for generating personalized recommendations. Technologies used HTML, CSS, Bootstrap, Jinja2, Flask, MySQL, OpenAI API. The application features data storage, profile view and daily meal plan generation.

Keywords: web application, nutrition, artificial intelligence, recommendations, Flask, GPT, MySQL, HTML, CSS, Bootstrap.

61 p., 12 fig., 1 tbl., 32 sources.

ЗМІСТ

ВСТУП	5
РОЗДІЛ 1. АНАЛІЗ СУЧАСНИХ ПІДХОДІВ ДО РАЦІОНАЛЬНОГО ЦИФРОВИХ РІШЕНЬ У СФЕРІ ЗДОРОВ'Я.....	7
1.1. Постановка задачі та вибір архітектурного підходу	7
1.2 Аналіз існуючих рішень	10
1.3. Сучасні проблеми харчових звичок та виклики	15
1.4. Методи аналізу харчування, традиційні та цифрові підходи.....	15
1.5. Використання технологій BigData у харчових системах.....	18
1.6. Стандарти та ризики зберігання персональних даних у харчових додатках	19
Висновки до розділу	21
РОЗДІЛ 2. ВИБІР ТЕХНОЛОГІЙ ТА АЛГОРИТМІВ ДЛЯ СТВОРЕННЯ ВЕБ-ПЛАТФОРМИ ПЕРСОНАЛЬНОГО ХАРЧУВАННЯ.....	23
2.1. Загальні положення щодо вибору технологій.....	23
2.2. Технологія Flask через Jinja2	24
2.3. Технологія Bootstrap	25
2.4. Використання MySQL	26
2.5. Алгоритми розрахунку калорій та раціонів за формулою Міффіна-Сан Жеора.....	27
2.6. Інтеграція зі штучним інтелектом через OpenAI GPT	29
2.7. Інструменти розробки PyCharm	31
2.8. Система контролю версій Git і GitHub	31
2.9. Безпека і конфіденційність	33
Висновки до розділу	34
РОЗДІЛ 3. ПРОЄКТУВАННЯ, РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ХАРЧУВАННЯ	36
3.1. Планування структури розробленої системи	36

3.2. Опис основних модулів	38
3.3. Результати роботи системи, приклади та демонстрація	49
3.4. Тестування та перевірка працездатності системи	51
Висновки до розділу	53
ВИСНОВКИ.....	55
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	58



ВСТУП

Актуальність кваліфікаційної бакалаврської роботи полягає в тому що здорове харчування є ключовим чинником профілактики хронічних захворювань. За даними Всесвітньої організації охорони здоров'я, станом на 2023 рік понад 1.9 мільярда дорослих у світі мали надлишкову вагу, з них 650 мільйонів - ожиріння. Також у понад 40% населення спостерігається дефіцит вітаміну D. Такі показники свідчать про нагальну потребу у створенні ефективних інструментів для контролю харчування. Сучасні цифрові технології дозволяють реалізувати інноваційні рішення в цій сфері, особливо з використанням штучного інтелекту.

Мета дослідження. Розробка веб-платформи для управління харчуванням із використанням алгоритмів штучного інтелекту, яка дозволяє здійснювати персоналізований аналіз раціону та надавати рекомендації щодо його покращення.

Об'єкт дослідження. Процес управління харчуванням користувача засобами інформаційних технологій.

Предмет дослідження. Алгоритми аналізу харчового раціону та формування персоналізованих рекомендацій на основі даних про нутрієнти із застосуванням штучного інтелекту.

Завдання дослідження

1. Проаналізувати сучасний стан проблеми авторизації контролю харчування.
2. Визначити вимоги до веб-платформи з управлінням харчуванням.
3. Розробити структуру веб-застосунку та алгоритми обробки даних.
4. Розробити веб-застосунок.
5. Інтегрувати штучний інтелект для формування персоналізованих рекомендацій.
6. Провести тестування розробленої системи.

Теоретичне значення полягає в узагальненні сучасних підходів до аналізу харчування та застосуванні алгоритмів машинного навчання в дієтології. Дослідження сприяє розвитку підвищує розуміння взаємозв'язку між харчовою поведінкою та станом здоров'я на основі даних. Окрему увагу приділено питанням адаптивності алгоритмів у контексті змінних харчових звичок, що дозволяє закласти теоретичну основу для подальших досліджень у сфері e-health та створення гнучких експертних систем підтримки прийняття рішень у харчуванні.

Практичне значення полягає в тому що розроблена система може бути використана у практиці сімейної медицини, фітнес-індустрії та для особистого контролю харчування.

Апробація результатів дослідження. Основні результати дослідження опубліковано у науковій фаховій статті (фаховий журнал категорії Б):

Римар П.В., Ліваковський В.К. Розробка платформи для управління харчуванням з інтеграцією штучного інтелекту. *Вісник Східноукраїнського національного університету імені Володимира Даля* 2025.

Результати роботи обговорювалися на VI Всеукраїнській науково-практичній конференції здобувачів вищої освіти та молодих вчених «Прикладні інформаційні технології 2025» з публікацією тез доповідей:

Ліваковський В.К., Римар П.В. Розробка платформи для управління харчуванням з інтеграцією штучного інтелекту. *Прикладні інформаційні технології 2025*: Матеріали всеукр. науково-практ. конф. здобувачів вищ. освіти та молодих вчен., м. Вінниця, 22 трав. 2025р. 2025.

Структура кваліфікаційної (бакалаврської) роботи. Робота складається зі вступу, трьох розділів, висновків, списку використаних джерел в кількості 32 найменування, 12 рисунків, 1 таблиці. Загальний об'єм роботи 61 сторінка.

РОЗДІЛ 1

АНАЛІЗ СУЧАСНИХ ПІДХОДІВ ДО РАЦІОНАЛЬНОГО ЦИФРОВИХ РІШЕНЬ У СФЕРІ ЗДОРОВ'Я

1.1. Постановка задачі та вибір архітектурного підходу

У процесі розробки програмного забезпечення важливим етапом є чітке визначення мети, постановка задачі та обґрунтування вибору відповідної архітектури, що дозволить досягти необхідної функціональності, масштабованості та зручності у використанні. Відповідно до поставленої мети створення системи інтелектуального управління харчуванням із використанням елементів штучного інтелекту основними завданнями, які необхідно було реалізувати в рамках проєкту є:

- Забезпечення можливості вводити фізіологічні параметри, вік, вага, зріст, стать.
- Автоматичний розрахунок добової калорійної норми на основі відомих формул.
- Формування персоналізованих порад та плану харчування за допомогою великої мовної моделі OpenAI GPT.
- Збереження введених даних, результатів та історії змін у базі даних.
- Створення зручного та інтуїтивного інтерфейсу для взаємодії з користувачем.
- Захист персональних даних і базова автентифікація.

Після аналізу предметної області було виявлено, що більшість існуючих рішень у сфері харчування або не враховують індивідуальні особливості користувача, або обмежуються базовим розрахунком калорій без подальшої персоналізації. Саме тому у проєкті було обрано шлях поєднання класичних обчислювальних методів із сучасними досягненнями в галузі генеративного штучного інтелекту.

Обґрунтування вибору архітектури

З метою забезпечення гнучкості та розширюваності, для реалізації системи було обрано архітектурний підхід, який поєднує серверну логіку, реалізовану у фреймворку Flask, з шаблонізованими HTML-сторінками через механізм Jinja2. Такий підхід дозволяє обробляти запити, зберігати й аналізувати дані, а також забезпечує динамічну генерацію веб-сторінок на основі змінних. Фактично застосовується спрощена модель Model View Controller, далі MVC:

- Model – модулі models.py та база даних MySQL, що зберігає всі параметри користувачів, тренди, цілі
- View – HTML шаблони з Jinja2, що відповідають за вивід даних на інтерфейс.
- Controller – маршрути у routes.py, які обробляють логіку взаємодії, запити до API, валідацію, тощо.

Додатково до цієї структури інтегрується зовнішній API OpenAI для генерації текстових рекомендацій. Такий підхід дозволяє уникнути складної логіки на стороні користувача і централізувати обробку даних на сервері, що значно спрощує супровід і масштабування проєкту. Таким чином, обраний архітектурний підхід дозволяє поєднати простоту реалізації, наочність структури та гнучкість розширення, а також відповідає вимогам до створення сучасного веб-застосунку з елементами штучного інтелекту.

Загальні задачі, що необхідно розробити

Реалізація реєстрації та входу з базовою автентифікацією: створення профілю користувача з можливістю введення індивідуальних параметрів: вік, стать, вага, зріст, ціль(схуднення, підтримка, набір). Збереження введених даних у базі даних MySQL.

Алгоритми розрахунку добової калорійної потреби:

- Імплементація формули Міффіна-Сан Жеора з урахуванням базового метаболізму та коефіцієнта активності.

- Адаптація результатів залежно від обраної цілі: дефіцит, підтримка, надлишок калорій.

Побудова системи підрахунку калорій і нутрієнтів:

- Розробка механізму введення прийомів їжі та продуктів з обліком калорій, білків, жирів, вуглеводів.
- Підключення локальної або вбудованої бази продуктів харчування з можливістю ручного додавання.
- Обчислення денних підсумків і порівняння з цільовими значеннями.

Генерація персоналізованих рекомендацій:

- Створення модуля взаємодії з OpenAI GPT для генерації текстових порад на основі статистики споживання за день/тиждень, цілей користувача, прогресу тенденцій у вазі, змінах параметрів.
- Інтерфейс взаємодії з API та фільтрації вхідних даних для запитів.

Побудова системи збереження історії:

- Запис усіх змін параметрів, вага, цілі, харчування до окремої таблиці історії.
- Реалізація механізму перегляду попередніх записів за датами.

Інтерфейс користувача:

- Верстка сторінок профілю, дашборду, редактора прийомів їжі, результатів.
- Застосування шаблонів Jinja2 для динамічного виводу інформації
- Забезпечення адаптивності інтерфейсу та базової валідації форм.

Безпека та захист даних:

- Захист сторінок для зареєстрованих користувачів.
- Хешування паролів та безпечна взаємодія з базою даних.
- Дотримання базових принципів обробки персональних даних згідно з GDPR.

Тестування системи:

- Модульне тестування критичних функцій, розрахунок калорій, інтеграція з GPT, збереження в БД.

- Перевірка навантаження та помилок при взаємодії з користувачем.
- Акт перевірки відповідності функціоналу заявленим вимогам.

1.2 Аналіз існуючих рішень

Cronometer [28] – потужний інструмент для відстеження харчування та здоров'я, який надає користувачам можливість контролювати споживання макро та мікронутрієнтів. Основні функції включають:

- точний підрахунок поживних речовин: база даних Cronometer містить понад 1 мільйон перевірених продуктів, що забезпечує високу точність підрахунку калорій, вітамінів та мінералів;
- інтеграція з фітнес-пристроями: сервіс синхронізується з Apple Health, Apple Watch, Fitbit та іншими фітнес-трекерами, що дозволяє комплексно аналізувати активність та харчування;
- преміум функції (Cronometer Gold): наявність розширених можливостей, таких як імпорт рецептів, детальні графіки нутрієнтів та планування прийомів їжі;
- звіти та діаграми: детальний аналіз споживаних продуктів дозволяє користувачам коригувати раціон відповідно до рекомендацій.



Рисунок 1.1 – Логотип Cronometer

MyFitnessPal [27] є одним з найпопулярніших додатків для підрахування калорій і моніторингу фізичної активності. Основні можливості включають:

- глобальна база продуктів: містить мільйони записів продуктів, що дозволяє швидко знаходити необхідну інформацію про калорійність і склад;
- підрахунок калорій та макронутрієнтів: дає змогу контролювати співвідношення білків, жирів і вуглеводів у раціоні;
- інтеграція з розумними пристроями: підтримує синхронізацію з багатьма фітнес-трекерами, такими як Garmin, Fitbit, Samsung Health;
- соціальні можливості: користувачі можуть ділитися своїми досягненнями та отримувати підтримку від спільноти;
- Недоліком MyFitnessPal є наявність реклами у безкоштовній версії, а також платний доступ до розширених функцій.

The logo for MyFitnessPal, featuring the brand name in a bold, blue, sans-serif font. The logo is centered and overlaid on a faint, circular watermark of a university seal.

Рисунок 1.2 – Логотип MyFitnessPal

FatSecret [29] це безкоштовний додаток для підрахування калорій, який відзначається простим інтерфейсом і доступними функціями:

- щоденник харчування: дозволяє легко записувати прийоми їжі та стежити за їхньою калорійністю;
- сканер штрих-кодів: швидке введення інформації про продукти шляхом сканування їхніх штрих-кодів;
- відстеження фізичної активності: дає змогу записувати витрачені калорії та тренування;
- інтеграції з пристроями: синхронізується з популярними фітнес-трекерами;

- FatSecret хоч і є безкоштовним додатком, але йому бракує функцій для аналізу нутрієнтів.



Рисунок 1.3 – Логотип FatSecret

LifeSum [26] поєднує функції трекера харчування з рекомендаціями щодо здорового способу життя. Його функціонал:

- персоналізовані плани харчування: програма пропонує дієти відповідно до цілей користувача;
- відстеження споживання їжі та води: допомагає користувачам контролювати щоденний баланс;
- інтеграція з фітнес-додатками: синхронізується з Google Fit, Apple Health та іншими сервісами;
- рецепти та корисні поради: пропонує готові страви та рекомендації щодо здорового харчування.

LifeSum є хорошим варіантом для тих, хто прагне не тільки вести щоденник харчування, а й отримувати індивідуальні рекомендації.

Аналіз існуючих рішень показує, що більшість популярних додатків зосереджені на підрахунку калорій та базовому аналізі нутрієнтів, проте мають певні обмеження. Наприклад, Cronometer пропонує детальний аналіз мікроелементів, але вимагає преміум-доступу для розширених можливостей. MyFitnessPal має найбільшу базу продуктів, але безкоштовна версія містить рекламу. FatSecret є безкоштовним та зручним у використанні, але не надає

глибокого аналізу поживних речовин. Lifesum орієнтований на персоналізовані рекомендації, що є корисним для користувачів, які хочуть не тільки контролювати калорії, а й слідувати певним дієтичним планам.

Окрім популярних зарубіжних платформ, таких як Cronometer, MyFitnessPal, Google Fit та інших, важливо також звернути увагу на вітчизняні розробки у сфері контролю харчування, здоров'я та фізичної активності. Українські сервіси мають ряд переваг, зокрема локалізований контент, адаптацію до національних кулінарних традицій і доступність українською мовою. Нижче розглянуто кілька прикладів таких систем.

Health24 – це український інформаційний медичний портал, який пропонує користувачам статті, поради лікарів, відео та новини, пов'язані зі здоров'ям, харчуванням і способом життя[6]. Платформа не є класичною системою для підрахунку калорій чи складання раціону, проте містить численні рекомендації щодо здорового харчування та поради дієтологів[4].

Переваги:

- українська мова інтерфейсу та контенту;
- актуальні рекомендації, адаптовані до українського контексту;
- можливість отримати інформацію про харчові звички, дієти, продукти;
- постійне оновлення статей і консультації з експертами.

Недоліки:

- відсутність інтерактивного функціоналу для обліку харчування;
- платформа не дозволяє автоматично підраховувати калорії чи БЖВ;
- більше орієнтована на читача, ніж на активного користувача.



Рисунок 1.4 – Логотип Health24

Wellness Studio – це українське програмне забезпечення, розроблене для фахівців у сфері дієтології, фітнесу та медицини. Воно дозволяє створювати індивідуальні плани харчування, контролювати фізичні показники клієнтів, вести облік калорій та макронутрієнтів. Хоча ця система орієнтована переважно на професіоналів, вона може бути використана як інструмент для персонального моніторингу здоров'я.

Переваги:

- потужний набір функцій для аналізу харчування та фізіологічних показників;
- орієнтація на українські продукти, можливість локалізованих розрахунків;
- може бути використана як база для професійної консультації користувачів.

Недоліки:

- високий поріг входу для звичайного користувача;
- відсутність мобільного додатку та зручного веб-інтерфейсу для щоденного використання;
- призначена здебільшого для медичних чи фітнес-установ, а не для самостійного використання.



Рисунок 1.5 – Логотип Wellness Studio

Таким чином, у рамках цього дослідження передбачається розробка веб-платформи, яка поєднуватиме точність підрахунку нутрієнтів, гнучкість у

використанні та можливість отримання персоналізованих рекомендацій без необхідності придбання преміум-доступу.

1.3. Сучасні проблеми харчових звичок та виклики

Основні проблеми, що виникають через недостатній контроль харчування є ожиріння та метаболічний синдром – надмірне споживання калорій без контролю їхньої якості призводить до порушень обміну речовин. [14] Дефіцит нутрієнтів – неправильний підбір продуктів може спричиняти нестачу важливих мінералів та вітамінів. Хронічні захворювання – незбалансоване харчування пов'язане з ризиком серцево-судинних хвороб, діабет 2-го типу, остеопорозу. Психологічні аспекти – неконтрольоване споживання їжі часто пов'язане з розладами харчової поведінки. [10] З огляду на ці проблеми, розробка автоматизованих інструментів для моніторингу харчування, підрахунку калорій та аналізу є актуальним завданням, яке може допомогти у вирішенні глобальних питань, пов'язаних зі здоров'ям населення. Вивчення наукових джерел з питань харчування, аналізу нутрієнтів та автоматизації моніторингу раціону показує [1], що протягом останніх десятиліть було запропоновано чимало методів для оцінки харчових звичок та створення персоналізованих рекомендацій. [9]

1.4. Методи аналізу харчування, традиційні та цифрові підходи

Основними підходами до аналізу раціону та контролю нутрієнтів є статистичні методи оцінки харчування які в свою чергу базуються на анкетуванні та оцінці середньої спожитої кількості калорій і нутрієнтів, також часто використовуються у великих дослідженнях для оцінки впливу харчових звичок на стан здоров'я. Основним недоліком є те що суб'єктивність відповідей може досить неточно відображати реальність у розрахунках. [3]

Також біохімічний підхід є одним з найпопулярніших методів аналізу який базується на лабораторних аналізах, наприклад аналіз крові, сечі або інших біологічних матеріалів для визначення рівня нутрієнтів в організмі.

Найточніший метод, але дорогий та не завжди доступний для регулярного використання.[2]

Цифрові технології та алгоритми машинного навчання також допомагають при аналізі, основними перевагами є те, що для аналізу використовуються автоматизовані методи для аналізу даних про харчування. Підхід ґрунтується на обробці великих обсягів інформації та пошуку закономірностей у харчових звичках. Досить важлива гнучкість та можливість персоналізації рекомендації. У наукових працях останніх років акцент робиться на поєднанні технологій штучного інтелекту та нутриціології. [15]

Дослідження, проведене Smith et al [22] демонструє, що використання алгоритмів машинного навчання дозволяє підвищити точність оцінки харчового раціону на 35% у порівнянні з класичними методами. Інше дослідження Lee et al [16] показало, що впровадження персоналізованих рекомендацій за основі аналізу історій харчування користувача дозволяє знизити дефіцит основних нутрієнтів на 20% у тестовій групі. Дослідження Wilson et al [17] описує алгоритми підрахунку енергетичного балансу на основі віку, маси тіла та рівня активності людини, що доводить те що математичні моделі можуть зменшити похибку оцінки калорійності на 15-20%, але не враховують індивідуальні особливості метаболізму.

Значною перевагою використання технологій штучного інтелекту є те, що завдяки великим базам даних з різною інформацією щодо їжі та її енергетичної цінності алгоритму вдається точно аналізувати харчовий баланс користувача. Також втілення штучного інтелекту в системи відслідковування здоров'я надає приблизний висновок щодо фізичного здоров'я користувача що значно знижує рівень виникнення різних хвороб пов'язаних з харчуванням. Дослідження Chen et al [18] описує технологію розпізнавання продуктів за зображеннями за допомогою конволюційних нейронних мереж. Результати дослідження показують, що система автоматично визначає калорійність страв та має точність 92,5%, що значно перевищує традиційні методи підрахунку калорій.

Впровадження штучного інтелекту в автоматизацію моніторингу харчування дозволить побудувати різні системи глибокого аналізу харчових звичок, нейронні мережі зможуть аналізувати калорійність та нутрієнтів. Використання колаборативної фільтрації для створення персоналізованих рекомендацій на основі харчових уподобань. Аналізувати часові ряди для прогнозування змін у метаболізмі та рівні нутрієнтів. Аналіз наукових досліджень свідчить про те, що традиційні підходи до моніторингу харчування мають значні обмеження. Використання машинного навчання та автоматизованих алгоритмів дозволяє значно підвищити точність оцінки калорійності їжі, передбачати ризики дефіциту нутрієнтів та надавати персоналізовані рекомендації. [12]

Результати наукових праць підтверджують необхідність створення веб-додатку, який буде поєднувати сучасні алгоритми аналізу харчування, глибоку персоналізацію та високу точність оцінки раціону. Тому розробка інтелектуальних систем для моніторингу харчування та контролю нутрієнтів є важливим напрямом сучасних досліджень у сферах дієтології, медицини, інформаційних технологій та штучного інтелекту. Вивчення проблематики дозволяє не лише створити програмні рішення, що допоможуть людям слідкувати за своїм харчуванням, але й розширити наукові підходи до персоналізованого аналізу раціону. Теоретична значимість даного дослідження полягає в аналізі існуючих підходів до оцінки харчових звичок, виявленні їхніх недоліків та обґрунтуванні використання машинного навчання для підвищення точності аналізу харчового раціону.

В результаті виконання даного проекту варто зазначити взаємодію різних галузей роботи таких як медицина, нутрициологія, дієтологія та інформаційні технології.

1.5. Використання технологій BigData у харчових системах

Однією з ключових тенденцій сучасної науки є використання Big Data у медичних та нутриціологічних дослідженнях. [23] Основні можливості Big Data дозволяють збирати та аналізувати величезні масиви даних про харчові звички мільйонів користувачів. Виявлення кореляцій між раціоном та розвитком певних захворювань. Розробка персоналізованих рекомендацій на основі статистичних закономірностей у поведінці користувачів. Big Data щодня отримує джерела даних у харчуванні завдяки веб-платформам, інтернет ресурсам, традиційним методам опитування, наукових досліджень які в свою чергу дозволяють покращувати систему розробки персоналізованих рекомендацій. Основні методи аналізу Big data застосовують різні алгоритми, одним з яких є машинне навчання та глибокі нейронні мережі, які використовуються для передбачення дефіциту нутрієнтів та розпізнавання продуктів у раціоні та генерації рекомендації на основі вподобань користувача. [11] У дослідженні Lee нейромережі збільшили точність розрахунку калорійності продуктів до 93%. Також важливо зазначити що алгоритми кластеризації групують користувачів за харчовими звичками, дозволяючи створювати персоналізовані плани харчування. Дослідження Williams кластеризації допомогло виділити 5 основних типів харчування серед 1 мільйона користувачів. Аналіз часових рядів дозволяє відстежувати зміни у раціоні користувачів та їхній вплив на метаболізм що дозволяє використати аналіз у передбаченні ефективності дієти.

На сьогодні Big Data активно використовується у медицині, дієтології та розробці веб-додатків. Прикладом використання є Google Health який прогнозує на основі Big Data ризики серцево-судинних захворювань, виходячи з даних про раціон пацієнтів. [25]



Рисунок 1.6 – Логотип Google Health

Дослідження Brown et al [20] довело, що використання Big Data в мобільних додатках дозволило на 30% знизити ризик надмірної ваги серед активних користувачів. Використання Big Data також має обмеження та виклики у харчовій сфері. Попри величезний потенціал, використання Big Data має труднощі, такі як проблеми з конфіденційністю, збір персональних даних користувачів вимагає дотримання стандартів безпеки. Неточність даних, отриманні дані з веб-платформ або інтернет-ресурсів досить часто містять помилки через суб'єктивність введених даних. Обмежена інтерпретація результатів через великий обсяг інформації не завжди дозволяє чітко оприлюднювати висновок, що потребує додаткового ретельного аналізу. Робота з BigData потребує потужних серверів та складних обчислювальних систем. [30]

Таким чином, інтеграція Big Data у систему моніторингу харчування дозволяє підвищити якість аналізу раціону та точність прогнозування потенційних проблем. [21]

1.6. Стандарти та ризики зберігання персональних даних у харчових додатках

Важливо зазначити, що розробка веб-додатку повинна дотримуватись основних правил безпеки при зборі та обробці даних користувачів. Бази даних веб-додатків та інтернет ресурсів зберігають дані про користувачів, дані про споживані продукти, калорійність, персоналізовані рекомендації та навіть стан

здоров'я. Уразливість системи безпеки може призвести до несанкціонованого доступу та використання цієї інформації сторонніми особами.

У 2021 році стався масштабний витік даних з платформи MyFitnessPal, унаслідок якого хакери отримали доступ до 150 мільйонів акаунтів, включаючи харчові звички, паролі та email-адреси користувачів, такі інциденти підкреслюють необхідність надійного шифрування та анонімізації персональних даних. Деякі компанії можуть передавати дані користувачів третім сторонам без їхньої згоди, наприклад фармацевтичним компаніям або маркетинговим агентствам. Аналіз споживчих звичок без явної згоди користувачів порушує право на приватність та суперечить міжнародним стандартам. Також важливо розуміти ризики кіберзлочинності, хакери можуть використовувати фішингові атаки або взломи баз даних щоб отримати доступ до важливої інформації та продати її на чорному ринку. Такі дані можуть бути використані для маніпуляції харчовими звичками користувачів, підробки рекомендацій або навіть для фальсифікації медичних показників.

Міжнародні стандарти та регулювання захисту даних забезпечують безпеку персональних даних, у багатьох країнах діють регуляторні акти, які визначають правила збору, зберігання та обробки конфіденційної інформації. General Data Protection Regulation використовується Європейським Союзом, регламент встановлює жорсткі вимоги до збору персональних даних, включаючи право користувача на забуття, тобто можливість повного видалення всіх своїх даних із системи, вимагаючи чіткої згоди перед обробкою будь-якої персональної інформації. Health Insurance Portability and Accountability Act використовується Сполученими Штатами Америки. Закон США, що регулює зберігання та обробку медичних даних, включаючи нутриціологічну інформацію. Вимагає шифрування даних пацієнтів та обмеження доступу до них, які в свою чергу змушують веб-додатки та інтернет-ресурс чітко визначати межу між обробкою даних та розповсюдженням їх. California Consumer Privacy Act також використовується Сполученими Штатами Америки, захищаючи права

споживачів у Каліфорнії щодо їх персональних даних. Дозволяє користувачам дізнаватися, які дані зберігаються та вимагати їхнього видалення. ISO/IEC 27001 [13] – це міжнародний стандарт для захисту інформаційних систем та забезпечення безпеки персональних даних, розроблений Міжнародною організацією зі стандартизації ISO та Міжнародною електротехнічною комісією ІЕС. Його мета полягає в тому, щоб встановити ефективні практики управління безпекою інформаційних систем, захисту даних та запобігання кіберзагрозам. [19] Стандарт регламентує управління ризиками у сфері інформаційної безпеки, що особливо важливо для організацій, які працюють із конфіденційними даними користувачів, включаючи сервіси моніторингу харчування. Основні принципи роботи ISO/IEC 27001 полягають в конфіденційності, забезпечення того що доступ до даних мають тільки уповноважені особи. Захист інформації від змін, модифікацій або зловмисного редагування та забезпечення безперервного доступу до даних для користувачів, які мають відповідні вимоги.

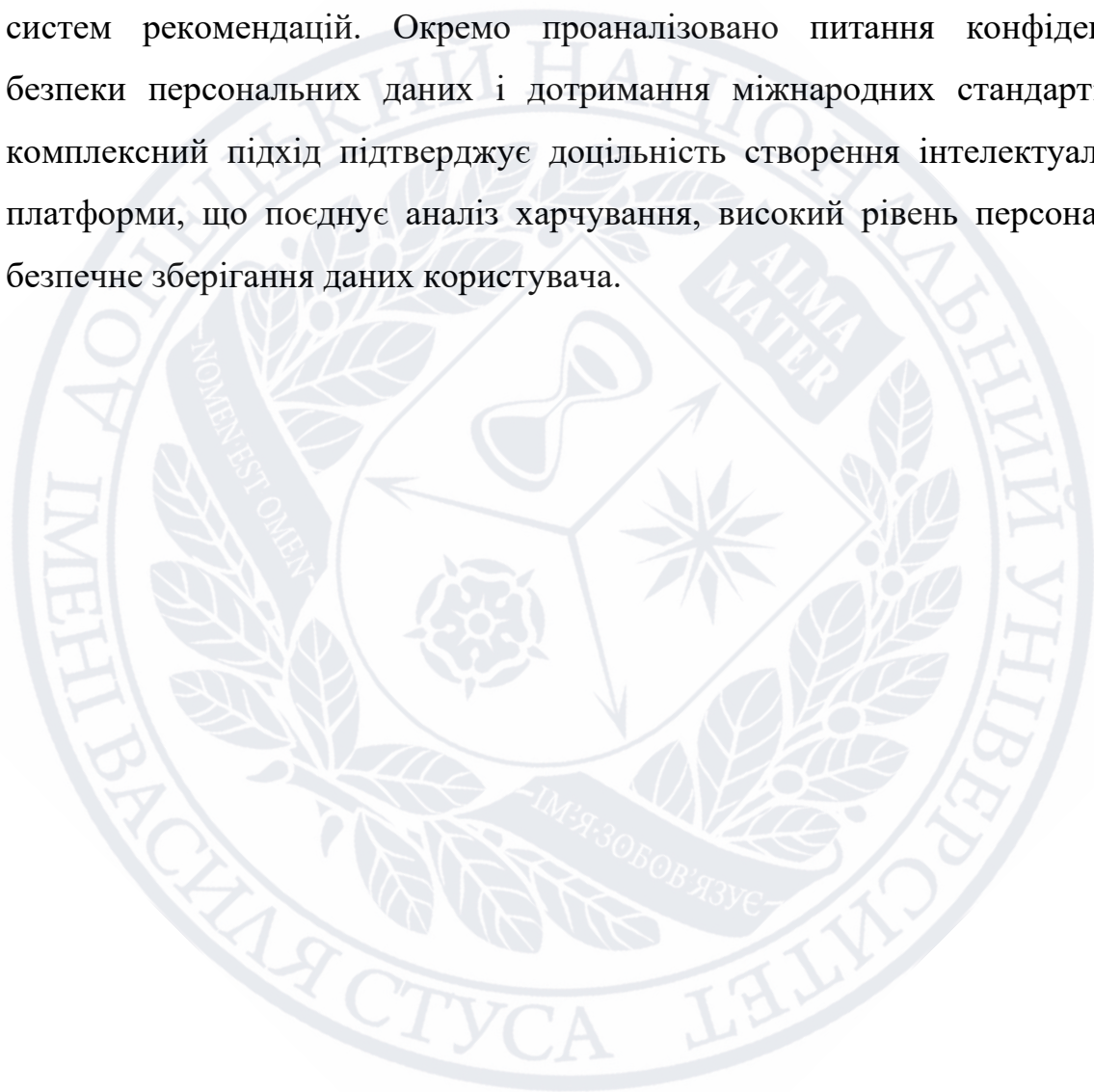
Серед міжнародних ініціатив, що підтримують розвиток медичних ІТ рішень в Україні, варто відзначити проєкт USAID HealthLink, що акцентує увагу на доступності медичних послуг, у тому числі через розвиток електронної взаємодії з користувачем. [7]

Для платформ аналізу харчування та нутрієнтів ISO/IEC 27001 є ключовим стандартом, що забезпечує безпеку персональних даних користувачів. Таким чином впровадження стандарту у розробку веб-додатку для моніторингу харчування забезпечить високий рівень безпеки, відповідність міжнародним нормам та надійний захист персональних даних користувачів.

Висновки до розділу

В першому розділі було розглянуто теоретичні та практичні аспекти, що формують підґрунтя для розробки системи управління харчуванням із застосування технологій штучного інтелекту. Оцінено сучасний стан проблеми раціонального харчування, проаналізовано причини порушень харчових звичок

і визначено актуальність створення цифрових інструментів для їх моніторингу. Особливу увагу приділено існуючим рішенням, як зарубіжним так і українським. Визначено, що більшість сервісів зосереджені на базовому підрахунку калорій або надання медичних рекомендацій, не забезпечуючи глибокого рівня персоналізації. Також акцентовано увагу на потенціалі використання BigData та алгоритмів машинного навчання для підвищення точності та адаптивності систем рекомендацій. Окремо проаналізовано питання конфіденційності, безпеки персональних даних і дотримання міжнародних стандартів. Такий комплексний підхід підтверджує доцільність створення інтелектуальної веб-платформи, що поєднує аналіз харчування, високий рівень персоналізації та безпечне зберігання даних користувача.



РОЗДІЛ 2

ВИБІР ТЕХНОЛОГІЙ ТА АЛГОРИТМІВ ДЛЯ СТВОРЕННЯ ВЕБ-ПЛАТФОРМИ ПЕРСОНАЛЬНОГО ХАРЧУВАННЯ

2.1. Загальні положення щодо вибору технологій

Процес створення сучасного веб-додатку, орієнтованого на зручну взаємодію з користувачем, ефективну обробку даних та підтримку персоналізованих функцій, неможливий без ретельно підбраного технологічного стеку. На початковому етапі розробки системи управління харчуванням було проаналізовано різні технічні рішення, зважаючи на такі чинники, як зручність реалізації, сумісність компонентів, масштабованість, безпека, швидкодія, а також відповідність функціональним вимогам проєкту. На відміну від комплексних Single Page Application із розділеною клієнтською та серверною частинами, у даному проєкті обрано більш лаконічно архітектуру одностороннього проєкта на основі Flask, де інтерфейс формується безпосередньо за допомогою шаблонів Jinja2. Такий підхід дозволяє зменшити кількість зовнішніх залежностей, спростити структуру додатку та зосередитися на реалізації основної функціональності, основними технологіями є Flask із шаблонізатором Jinja2, як основна серверна логіка та формування HTML сторінок. Bootstrap для побудови візуально привабливого й адаптивного інтерфейсу. MySQL, як система збереження структурованої інформації. OpenAI API для створення моделі штучного інтелекту, який формує рекомендації на основі введених користувачем параметрів. Pycharm як основне середовище розробки. Git та GitHub для організації контролю версій, фіксації змін та ведення історії змін проєкту. Кожен із зазначених інструментів має свою чітку роль у системі, доповнюючи інші компоненти та забезпечуючи цілісну, зручну і стабільну роботу платформи. Структура веб-додатку повинна розроблятися із урахуванням базових принципів безпечної архітектури веб сервісів, зокрема поділу на логічні рівні, обмеження прав доступу та ізоляції бізнес логіки.

2.2. Технологія Flask через Jinja2

Організація серверної логіки додатку є ключовим компонентом проєкту, адже саме на цьому рівні здійснюється обробка даних користувача, взаємодія з базою даних, формування запитів до штучного інтелекту та генерування динамічного контенту. Для реалізації цих функцій було обрано фреймворк Flask, який поєднує простоту, гнучкість і достатню функціональність для побудови повноцінного веб-застосунку. Flask є легким веб-фреймворком для мови Python, який дозволяє створювати веб-додатки без необхідності використовувати складну або надлишкову інфраструктуру. Його особливістю є мінімалістичний підхід, повний контроль над структурою проєкту, вибором компонентів, маршрутизацією та логікою. Це дає можливість створити максимально адаптований під вимоги продукту серверний застосунок без перезавантаження зайвими залежностями.

Flask виконує низку важливих функцій:

- прийом та обробка запитів від користувача через методи GET і POST;
- зчитування, обробка й валідація введених даних;
- логічне формування результатів;
- формування запитів до OpenAI API;
- передача отриманих відповідей у шаблони HTML через вбудований механізм шаблонізації Jinja2.

Шаблонізатор Jinja2, який є невід'ємною частиною Flask, дозволяє впроваджувати динамічні елементи у статистичні HTML сторінки. Завдяки цьому кожна сторінка наповнюється конкретним змістом залежно від введених параметрів. Ще однією перевагою є простота роботи з URL маршрутами за допомогою декораторів route, реалізовано чітку структуру логіки, кожна дія користувача це окремий маршрут, пов'язаний з певною функцією у коді.

Крім того у Flask реалізовано підтримку роботи з формами, сесіями, cookie-файлами, авторизацією та зовнішніми API. Це дозволяє не лише створювати базові інформаційні сторінки, але й формувати повноцінні веб-

сервіси. Обрана технологія виявилась оптимальною для поставлених завдань, забезпечуючи легку підтримку, зрозуміло структуру та швидкий розвиток функціоналу без потреби залучення важковагових фреймворків.

2.3. Технологія Bootstrap

Щоб забезпечити платформі сучасний вигляд, адаптивність до різних типів пристроїв і високий рівень зручності для користувача, у проєкті використано CSS-фреймворк Bootstrap. Цей інструмент дозволяє швидко й ефективно створювати візуально привабливі інтерфейси без необхідності розробляти стилі з нуля. Завдяки багатій бібліотеці готових компонентів і гнучкій системі сіток Bootstrap став одним з основних елементів клієнтської частини веб додатку. На відміну від ручного CSS-оформлення, яке вимагає значних зусиль для досягнення єдиного стилю та сумісності з мобільними пристроями, використання Bootstrap дозволяє:

- створювати форми, таблиці, кнопки, повідомлення та інші елементи за допомогою попередньо стилізованих класів;
- швидко реалізувати адаптивну сітку, яка автоматично підлаштовує компонування сторінки під ширину екрану;
- застосувати вбудовані класи для відступів, кольорів, розмірів та інших параметрів;
- інтегрувати в проєкт UI-компоненти, які відповідають сучасним трендам дизайну.

Bootstrap виконує важливі функції для оформлення всіх основних сторінок і форм, адаптації інтерфейсу до мобільних пристроїв без необхідності написання медіазапитів вручну, формування однакового стилю для всіх блоків, що забезпечує єдине сприйняття системи. Важливо зазначити, що Bootstrap не лише полегшує розробку, а й забезпечує відповідність базовим вимогами до UX/UI, доступність елементів, контрастність, масштабованість шрифтів, зручність натискання на мобільних екранах тощо.

2.4. Використання MySQL

Зберігання, обробка та доступ до структурованої інформації є критично важливими компонентами будь якого веб-додатку. У межах створеної системи управління харчуванням облік користувацьких даних, результатів аналізу, історії запитів та параметрів розрахунків здійснюється з використанням реляційної системи керування базами даних MySQL. MySQL – це одна з найпопулярніших систем керування базами даних з відкритим вихідним кодом, яка відзначається високою продуктивністю, надійністю та широкою підтримкою як у спільноті розробників так і на рівні документації. Вибір саме цієї системи був обумовлений ключовими факторами, сумісність з Python підтримка через бібліотеки, зокрема `mysql-connector-python` та `SQLAlchemy`, що забезпечує зручну інтеграцію з серверною логікою на Flask. Надійність та швидкодія MySQL дозволяє обробляти велику кількість одночасних запитів, що робить її стабільною основою для веб-додатків. Структурованість даних дає можливість визначення чітких полів, типів даних, зовнішніх ключів і зв'язків між таблицями, що дозволяє ефективно моделювати предметну область. Гнучкість та масштабованість бази даних дозволяє легко адаптуватись до змін у бізнес-логіці, підтримка нормалізації та модифікації структури.

У рамках реалізованої системи база даних містить кілька логічних блоків інформації, що відображають основні компоненти функціонування програми, насамперед зберігання облікових записів, персональних даних, параметрів активності, можливої авторизації. Раціони харчування, перелік згенерованих раціонів або запропонованих меню, які пов'язані з конкретними користувачами. Історія запитів, що дозволяє перегляди дані про взаємодію з системою, які рекомендації формувались та на основі яких параметрів. Сесії та статистика, що дозволяє вільно моніторити зміни у статистці користувача.

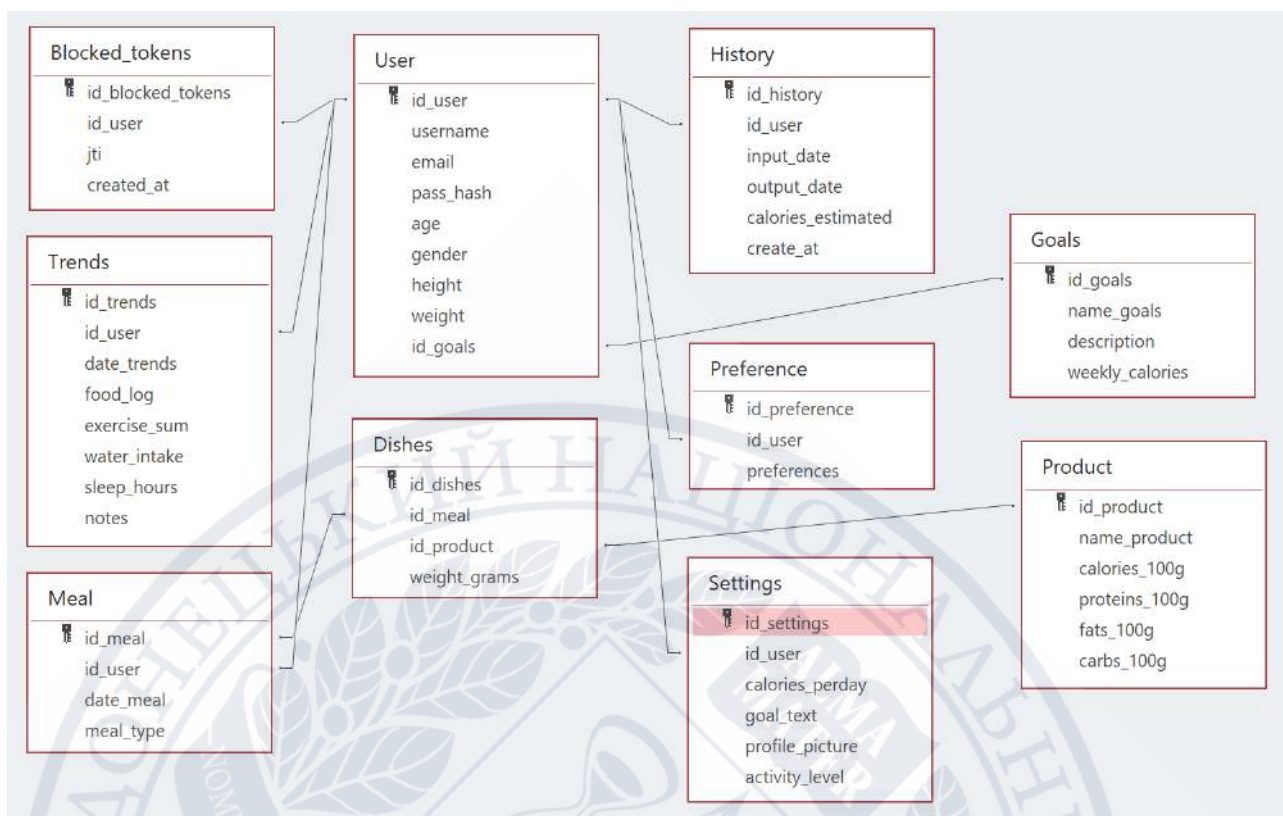


Рисунок 2.1 – Схема бази даних платформи управління харчування

2.5. Алгоритми розрахунку калорій та раціонів за формулою Міффіна-Сан Жеора

Одним із ключових завдань платформи управління харчуванням є забезпечення персоналізованих розрахунків добової потреби користувача в калоріях та макронутрієнтів, білків, жирів, вуглеводів. Для досягнення цієї мети було обрано використання класичних і перевірених формул, що дозволяють адаптувати рекомендації під індивідуальні параметри людини. [8]

Алгоритм розрахунку базового рівня метаболізму, далі BMR, визначає кількість калорій, яку організм людини втрачає у стані спокою для підтримки основних фізіологічних функцій. У розроблювальній системі для розрахунку BMR використовується формула Міффіна-Сан Жеора, яка вважається однією з найбільш точних сучасних методик. Формула для розрахунку BMR має вигляд:

для чоловіків

$$BMR = 10 * \text{вага(кг)} + 6,25 * \text{зріст(см)} - 5 * \text{вік(роки)} + 5 \quad (2.1)$$

для жінок

$$BMR = 10 * \text{вага(кг)} + 6,25 * \text{зріст(см)} - 5 * \text{вік(роки)} - 161 \quad (2.2)$$

Цей алгоритм обирається завдяки точності, адаптивності до різних категорій користувачів та широкому визнанню в дієтології. Корекція BMR за рівнем фізичної активності коригується відповідно до рівня фізичної активності користувача. Залежно від стилю життя та щоденних навантажень використовується відповідний коефіцієнт:

- малорухливий спосіб життя 1,2;
- невеликі фізичні навантаження 1,375;
- помірна активність 1,55;
- висока активність 1,725;
- дуже висока активність 1,9.

Таким чином визначається загальна добова потреба в калоріях яка використовується для подальших розрахунків.

Система дозволяє враховувати ціль користувача, збереження поточної ваги, зниження ваги зі створенням дефіциту калорій або набір маси, створення профіциту калорій. На основі цільового рівня калорій система також розраховує орієнтовну норму споживання макронутрієнтів в пропорціях, білки 15-20% від загальної енергетичної цінності, жири 25-30%, вуглеводи залишок енергії 50-60%. Такі пропорції відповідають загальноприйнятим рекомендаціям для здорового харчування та можуть бути адаптовані від специфічних потреб користувача.

Для складання щоденного меню система використовує вибір продуктів, які відповідають обмеженням за калорійністю, баланс макронутрієнтів відповідно до встановлених цілей, формування раціонів на основі простих, доступних продуктів харчування, можливість варіативності страв для збереження мотивації

користувача. Таким чином, розрахунок калорійності та складання раціонів у системі базуються на науково обґрунтованих формулах і підходах, що забезпечує високий рівень точності та персоналізації рекомендацій.

Перевагами обраної методики є те, що формула Міффіна-Сан Жеора широко використовується в медицині та спортивній дієтології. Універсальність дозволяє адаптуватись під різні категорії користувачів. Алгоритм легко впроваджується у серверну логіку без необхідності складних обчислювальних процесів. Розрахунки можуть виконуватись миттєво після введення даних користувачем.

2.6. Інтеграція зі штучним інтелектом через OpenAI GPT

У сучасних веб-додатках використання елементів штучного інтелекту значно підвищує рівень персоналізації сервісу, покращує якість взаємодії з користувачем та забезпечує гнучке формування контенту. У межах реалізованої платформи управління харчуванням для генерації індивідуальних рекомендацій створено модель на основі OpenAI API [24], сучасного програмного інтерфейсу, що надає можливість природної обробки мови, зокрема Generative Pre-Trained Transformer, далі GPT. Основна функція OpenAI API полягає в генерації персоналізованих порад і описів раціонів на основі параметрів введених користувачем. Запити OpenAI API формуються динамічно відповідно до введених користувачем даних.

Переваги використання моделі на основі OpenAI API дають змогу реалізувати функціональність, яка була б надзвичайно складною або затратною при самостійному створенні ШІ. Серед ключових переваг відзначається висока якість результатів, модель здатна генерувати зв'язний, граматично правильний текст із врахуванням запиту користувача. Контекстна обробка інформації: моделі на основі GPT можуть аналізувати структуру запиту, логічно узгоджувати числові та текстові параметри. Гнучкість: система легко адаптується до нових сценаріїв або типів запитів без необхідності навчання нової моделі. Миттєва

генерація відповідей, швидкість обробки запиту становить частки секунди, що дозволяє забезпечити комфортну взаємодію для кінцевого користувача.

Інтеграція OpenAI API реалізована безпосередньо в межах серверного застосунку, побудованого на Flask. Вся логіка відбувається в рамках одного веб-проєкту, де і серверна і клієнтська частина організовані на основі шаблонів HTML з Jinja2. Користувацький інтерфейс представлений HTML формами, які рендеряться за допомогою Flask. Взаємодія з GPT відбувається при надсиланні форми на сервер. Користувач заповнює форму з параметрами, Flask приймає ці дані через маршрут POST у відповідному представленні, на основі введених значень формується текстовий запит до моделі, запит передається у запиті до API з використанням бібліотеки openai, отриманий результат обробляється Flask і вставляється у відповідний шаблон для відображення користувачу. Таким чином, уся логіка повністю обробляється на стороні сервера, а відповіді штучного інтелекту виводяться безпосередньо у HTML шаблонах, що формуються на сервері. Це дозволяє уникнути потреби в окремій клієнтській логіці та забезпечує просту архітектуру проєкту.

Слід підкреслити, що розгляд варіанту створення власного алгоритму генерації порад або навіть тренування ML-моделі є малоефективним з погляду затрат часу, ресурсів і обсягу необхідних навчальних даних. Навчання навіть базової нейромережі вимагає великих обчислювальних потужностей і ретельно підготовлених датасетів. Крім того, забезпечення якості генерованого контенту без використання великомасштабних моделей на практиці є складним завданням. Інтеграція OpenAI GPT дозволила досягти високого рівня персоналізації та гнучкості при мінімальних витратах ресурсів. Такий підхід не лише спростив реалізацію функціоналу, але й надав можливість у майбутньому масштабувати систему або додавати нові типи запитів без істотних змін у програмному коді.

2.7. Інструменти розробки PyCharm

Вибір ефективного середовища розробки програмного забезпечення є важливою складовою успішного створення будь-якого програмного продукту. Основним інструментом розробки було обрано PyCharm – інтегроване середовище для програмування мовою Python, розроблене компанією JetBrains. Середовище PyCharm поєднує в собі багатий функціонал, оптимізований саме для Python розробки. Його зручний інтерфейс, інструменти аналізу коду та можливості розширення дозволяють значно прискорити процес написання, перевірки та налагодження логіки веб-додатку. Важливо також, що середовище підтримує повноцінну роботу з фреймворком Flask. Ключові функції PyCharm:

- аналіз помилок у реальному часі;
- зручне керування віртуальними середовищами;
- потужний візуальний дебаггер;
- інтеграція з системами контролю версій Git.

Крім базового функціоналу, PyCharm також має розширення для роботи з базами даних, HTML шаблонами та API запитамі, що дозволяє реалізувати повний цикл розробки від написання логіки обробки форм до тестування взаємодії з зовнішніми сервісами. Однією з переваг PyCharm у межах розробки проєкту є можливість зосередитися виключно на логіці, не витрачаючи час на технічне налагодження процесів. Наявність автоматичної перевірки коду, порад щодо рефакторингу та гнучкого керування середовищами створює комфортні умови для продуктивної розробки.

Отже, використання PyCharm дозволило забезпечити стабільну, зручну та безпомилкову роботу над усіма етапами проєкту, від початкового проєктування до фінального тестування та відлагодження.

2.8. Система контролю версій Git і GitHub

У розробці програмного забезпечення особливе значення має організація змін у коді, збереження історії правок, контроль стабільних версій та зручна

співпраця з іншими учасниками команди. Навіть у випадку індивідуального проєкту наявність системи контролю версій дозволяє забезпечити порядок, структурованість та захищеність процесу. Для цих цілей у даній роботі було використано комбінацію Git як локальний інструмент контролю версій і GitHub як хмарний сервіс для збереження та керування репозиторієм.

Git це розподілена система контролю версій, яка дозволяє зберігати всі зміни коду у вигляді окремих комітів, об'єднувати гілки розробки, повертатися до попередніх версій, а також виявляти конфлікти при паралельній роботі над одним і тим же кодом. Незважаючи на простоту у використанні, Git має потужну функціональність і вважається галузевим стандартом для більшості сучасних проєктів. Git використовувався для таких задач, фіксація кожного завершеного етапу розробки, ведення окремих гілок для тестування змін без ризику порушення стабільної версії, можливість перегляду історії змін, що спрощує відстеження помилок, зручний контроль за загальним обсягом роботи на різних етапах.

GitHub, у свою чергу, забезпечує зберігання репозиторію у віддаленому доступі. Це дозволило мати резервну копію проєкту у хмарі, відслідковувати прогрес із будь-якого пристрою, використовувати веб-інтерфейс для керування гілками, pull запитами та перегляду коду.

У поєднанні з PyCharm, який має вбудовану підтримку Git, весь процес контролю версій став інтуїтивно зрозумілим і технічно зручним. Після кожного завершеного функціонального блоку створювався окремий коміт із описом внесених змін, що дозволяло зберігати повну хронологію розвитку проєкту. Застосування Git і GitHub стало важливим компонентом організації роботи над системою, дозволяючи уникнути втрат коду, забезпечити гнучкість у тестуванні змін та сформуванню повноцінної історії розробки як доказ індивідуального внеску в результат.

2.9. Безпека і конфіденційність

Оскільки розроблювана система працює з чутливою інформацією, зокрема з персональними даними користувачів, таких як, вік, стать, зріст, вага, фізична активність, цілі щодо маси тіла, питання забезпечення їхньої безпеки має принципове значення. Незалежно від масштабу платформи чи кількості користувачів, захист даних є обов'язковою складовою будь якого відповідального цифрового продукту. У веб-застосунку захист персональних даних передбачено на кількох рівнях, валідація та обмеження введення даних що підтверджує що усі поля, що заповнюються користувачем, проходять первинну валідацію ще до збереження в базі даних, це дозволяє:

- запобігти введення некоректних або небезпечних значень по типу SQL ін'єкцій;
- контролювати допустимі діапазони числових значень, вік, вага, ріст тощо;
- гарантувати, що дані відповідають очікуваному формату, наприклад обмеження на типи введення у HTML формах.

У структурі бази даних застосовано типізацію полів та обмеження на довжину даних. Критично важливі поля, пароль користувача, обробляються із застосування хешування перед збереженням. Архітектура додатку передбачає зберігання лише тієї інформації, яка є критично необхідною для обчислення харчових рекомендацій. Жодні контактні дані, а також конфіденційна інформація, що дозволяє однозначно ідентифікувати особу, не збираються.

Формування відповідей, обробка даних та доступ до збережених результатів здійснюється лише через логіку Flask, яка обмежує можливість несанкціонованого доступу. Наприклад, дані не виводяться безпосередньо у шаблон, якщо вони не передані контролером із відповідною перевіркою. У розширенні системи до використання великою кількістю користувачів важливо реалізувати HTTPS з'єднання для захисту переданих даних, додати механізми автентифікації та авторизації, впровадити ролі користувачів таких як адміністратор, технічний адміністратор, звичайний користувач, зберігати

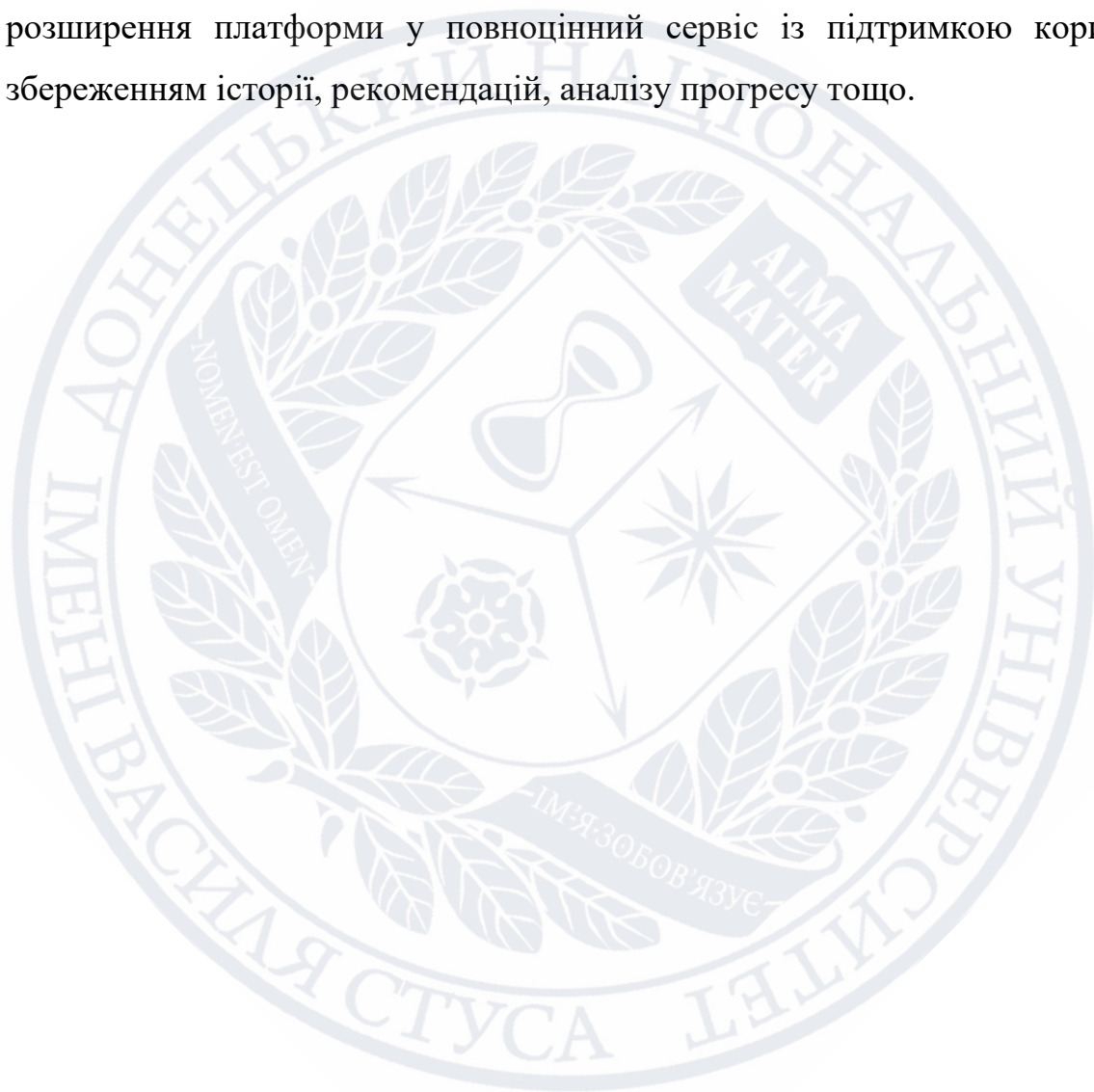
журнали доступу до критичних функцій. Також доцільно враховувати міжнародні вимоги щодо захисту даних, зокрема принципи GDPR, а також загальні рекомендації стандарту ISO/IEC 27001, які стосуються організаційної та технічної безпеки інформаційних систем.

Забезпечення безпеки даних користувачів це не лише технічне завдання, а й ознака відповідального ставлення до створення сервісів. Дотримання базових принципів захисту персональної інформації підвищує якість розробки та придатність сервіси до масштабування.

Висновки до розділу

В розділі було детально розглянуто набір інструментів, технологій і методів, що стали основою реалізованої системи управління харчуванням. Вибір кожного компонента обґрунтовано відповідно до поставлених цілей, обсягу функціональності, а також вимог до продуктивності, гнучкості та безпеки. Серверна частина застосунку побудована на основі Python фреймворку Flask, що забезпечує простоту організації логіки й повну інтеграцію з HTML шаблонами завдяки Jinja2. Для оформлення інтерфейсу використано Bootstrap, який дав змогу створити адаптивний та зручний у користуванні дизайн без надмірного навантаження на розробника. У якості бази даних обрано MySQL як стабільну й продуктивну реляційну СКБД, що дозволяє ефективно зберігати персональні дані та результати взаємодії з користувачем. Окрему увагу приділено інтеграції з OpenAI GPT, яка реалізує генерацію персоналізованих порад за допомогою моделей штучного інтелекту. Цей компонент значно розширює можливості системи, забезпечуючи якісну взаємодію із користувачем та підвищуючи рівень довіри до результатів. Також розглянуто алгоритмічну основу для розрахунку добової потреби в калоріях на основі формули Міффіна-Сан Жеора, що забезпечує науково обґрунтовану персоналізацію. Середовище розробки PyCharm та система контролю версій Git у поєднанні з GitHub забезпечили зручну організацію програмного коду, дозволили вести історію змін і спростили

керування проектом. Питання захисту персональних даних було враховано на кожному етапі, від обмеження введення, до логічного контролю доступу та рекомендацій для подальшого масштабування. Загалом, застосований технологічний стек є вдалим прикладом поєднання простоти реалізації та широких функціональних можливостей. Він дозволяє ефективно вирішувати поставлені завдання, а також слугує надійною основою для подальшого розширення платформи у повноцінний сервіс із підтримкою користувачів, збереженням історії, рекомендацій, аналізу прогресу тощо.



РОЗДІЛ 3

ПРОЄКТУВАННЯ, РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ХАРЧУВАННЯ

3.1. Планування структури розробленої системи

Після формалізації завдання та вибору архітектурної моделі наступним кроком стало планування логічної та фізичної структури програмного забезпечення. Важливою метою цього етапу є визначення модулів, відповідальних за зберігання, обробку та візуалізацію даних, а також організацію взаємодії між ними.

Структуру веб-застосунку було поділено на три функціональні частини:

- Серверна частина реалізована на мові Python із використанням фреймворку Flask. Забезпечує обробку запитів, бізнес логіку, обчислення, а також взаємодію з базою даних і зовнішнім API OpenAI.
- Клієнтська частина реалізована у вигляді HTML шаблонів із підтримкою шаблонізатора Jinja2. Відповідає за відображення даних, взаємодію з користувачем, форму введення параметрів, вивід результатів та рекомендацій.
- База даних реалізована за допомогою MySQL. Використовується для зберігання зареєстрованих користувачів, індивідуальних параметрів, історії змін, авторизаційних токенів і обраних цілей.

Такий розподіл компонентів дозволяє легко масштабувати окремі частини проєкту в разі потреби. Для ефективного проєктування інтерфейсу та взаємодії було виокремлено декілька ключових сценаріїв використання:

1) Реєстрація та вхід користувача:

- створення облікового запису;
- збереження авторизаційного токена;
- доступ до персоналізованих функцій.

2) Введення персональних параметрів:

- вага, зріст, вік, стать, ціль;
 - збереження в базу даних.
- 3) Автоматичний розрахунок калорійної норми:
- обчислення на основі формули Міффіна-Сан Жеора;
 - вивід результату на сторінку.
- 4) Генерація рекомендацій від GPT
- формування prompt запиту на основі введених параметрів;
 - отримання відповіді через API OpenAI;
 - відображення рекомендації у вигляді тексту.
- 5) Перегляд профілю та аналіз динаміки
- відображення змін у вазі, калоріях;
 - додавання нових записів;
 - перегляд трендів.

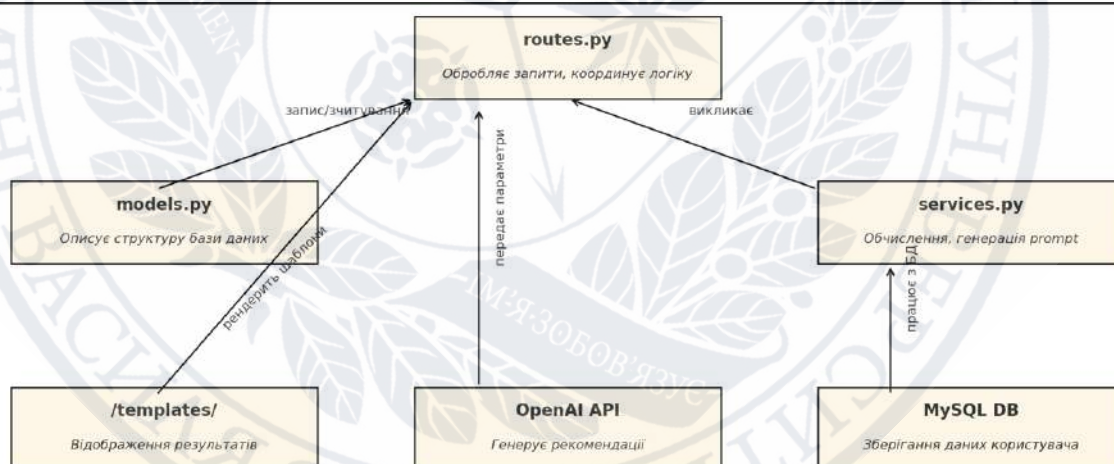


Рисунок 3.1 – UML діаграма взаємодії файлів

Проект організовано у вигляді структурованої файлової системи, що складається з основних елементів, templates – HTML шаблони з динамічним вмістом, static – стилі, шрифти, зображення, routes.py – маршрути та логіка обробки запитів, services.py – допоміжні обчислення, init.py – ініціалізація застосунку, config.py – змінні середовища, налаштування.

Хоча фреймворк Flask не накладає жорсткої структури, логіка реалізована відповідно до принципів шаблону MVC. Це забезпечує зрозумілу модульність, спрощує розгортання, супровід і подальший розвиток системи.

3.2. Опис основних модулів

Розроблена система складається з ряду логічно ізольованих модулів, кожен із яких відповідає за окремий етап обробки даних, взаємодії з користувачем чи зовнішніми сервісами. Такий підхід дозволяє не лише спростити підтримку проєкту, а й забезпечити гнучке масштабування або адаптацію системи в майбутньому.

Модуль обробки вхідних параметрів routes.py.

Модуль routes.py є ключовим елементом серверної частини розробленого веб-додатку, побудованого з використанням фреймворку Flask. Його основна функція полягає в організації маршрутизації запитів, а також у забезпеченні зв'язку між фронтендом, модулями обробки логіки та шаблонізатором Jinja2. У межах даного модуля реалізовано два основних маршрути:

- GET запит, який відображає початкову форму введення користувацьких параметрів.
- POST запит, що виконується після надсилання форми та ініціює процес обробки введених даних.

Алгоритм роботи модуля включає такі етапи, як зчитування даних з HTML форми через об'єкт request.form, проведення базової валідації введених значень, перевірка на заповнення, відповідність очікуваному типу, обробка числових значень. Виклик функцій з модуля services.py для обчислення базового метаболізму та добової потреби, розрахунку макронутрієнтів, формування текстового запиту до GPT-модель, отримання відповіді від OpenAI API та передача результату в шаблон за допомогою функції render_template.

Таким чином, `routes.py` виконує функцію координатора між користувацьким введенням, бізнес-логікою та виведенням результатів на сторінці.

В процес розробки було враховано потенційні проблеми, пов'язані з обробкою некоректних даних. Наприклад, введення числових значень у форматі з комою призводило до помилки перетворення типу. Ця проблема була вирішена за рахунок автоматичної заміни коми на крапку та подальшого перетворення значення у тип `float`. Також було реалізовано обмеження на допустимі значення:

- мінімальний і максимальний поріг віку, ваги та зросту;
- обов'язковий вибір статі та цілі;
- виведення відповідних повідомлень про помилку у разі невідповідності вимогам.

Це дозволило забезпечити стабільну роботу додатку та підвищити зручність для кінцевого користувача. Модуль забезпечує ключову логіку маршрутизації в системі та відповідає за обробку всіх запитів користувача. Завдяки суворому поділу обов'язків і делегуванню обчислювальної логіки до окремих модулів, він зберігає компактність і підтримує принципи модульності, що позитивно впливає на супровід і масштабованість системи в цілому.

Модуль розрахунку калорійності `services.py`

Модуль `services.py` є функціональним ядром веб-додатку, відповідальним за реалізацію прикладної логіки. Саме в ньому зосереджено механізми обробки користувацьких даних, математичні обчислення, необхідні для визначення добової потреби в енергії та макронутрієнтах, а також формування текстових запитів до GPT моделі. Основні функції та їх призначення

1. Обчислення базового метаболізму – для обчислення використовується формула Міффіна-Сан Жеора, яка є однією з найпоширеніших і науково обґрунтованих у практиці дієтології. Формула враховує вагу, ріст, вік та стать користувача.

2. Розрахунок загальної добової потреби в калоріях – отримане значення *BMR* множиться на коефіцієнт фізичної активності, який вибирається відповідно до вибору користувача, від сидячого способу життя до дуже активного режиму. Це дозволяє враховувати індивідуальні енергетичні потреби.

$$TDEE = BMR * k \quad (3.1)$$

де k – коефіцієнт в діапазоні від 1,2 до 1,9.

3. Розподіл макронутрієнтів, на основі загальної калорійності – система розраховує рекомендовану кількість білків, жирів та вуглеводів. Наприклад, білки – 20% від загальної енергії, жири – 30%, вуглеводи – 50%. З урахуванням калорійної цінності кожного нутрієнта, 4ккал/г для білків і вуглеводів, 9ккал/г для жирів, обчислюється грамаж кожного з компонентів.
4. Формування текстового запиту до GPT моделі – на основі обчислених значень генерується інформативний prompt для OpenAI GPT. У запит включаються, вік, вага, стать, ціль користувача, підтримка, набір чи зниження ваги та рівень активності. Це дозволяє отримати контекстуально точну відповідь у вигляді персоналізованої рекомендації. Приклад сформованого звіту: «Простий денний раціон для жінки 28 років, вага 60 кг, ріст 165 см, мета зниження ваги, активність помірна. Поясни як це розраховується кількість калорій і розподіл БЖВ». На що модель відповість згідно до поставленого запиту.

На етапі розробки виникли наступні труднощі:

- обмеження на довжину prompt запиту, модель GPT має обмеження на кількість токенів, тому було прийнято рішення лаконізувати запити, уникаючи надмірної деталізації;
- нестабільність відповіді, при однакових параметрах відповідь могла змінюватись. Для часткової стабілізації було знижено температуру генерації до рекомендованого значення 0,7.

Модуль `services.py` виконує критично важливу роль у системі, вона забезпечує науково обґрунтовані розрахунки, що є основою для подальшої генерації порад. Завдяки чіткій структурі, відокремленню обчислень від маршрутизації та інтерфейсної частини, модуль легко масштабувати та підтримувати. Його логіка є достатньо універсальною, щоби за потреби адаптувати під інші цілі або змінити методики розрахунків.

Модуль роботи з базою даних `models.py`

Модуль `models.py` виконує функції зв'язувальної ланки між прикладною логікою веб-застосунку та системою збереження даних. У рамках проекту було обрано реляційну базу даних MySQL, яка забезпечує надійне, структуроване й масштабоване зберігання інформації про параметри користувача та результати обчислень. Модуль містить структуру таблиці, SQL логіку та ORM визначення для запису та подальшого використання цих даних.

Для реалізації бази даних створена таблиця яка містить поля:

1. ID – унікальний ідентифікатор запису, тип: ціле число, автоінкремент.
2. Стать користувача, тип: ціле число.
3. Вік, тип: ціле число.
4. Зріст, тип: ціле число.
5. Вага, тип: ціле число або з плаваючою крапкою.
6. Рівень фізично активності, тип: числовий код або рядкове представлення.
7. Ціль.
8. Розрахована калорійна потреба.
9. Грамова кількість білків, жирів, вуглеводів.
10. Текстова порада від GPT, тип: у вигляді текстового поля великої довжини.
11. Дата й час створення запису.

Така структура дозволяє відтворити історію обчислень користувача, проводити статистичний аналіз, реалізувати персональний кабінет чи збереження прогресу, забезпечити відповідність вимогами до нормалізації баз

даних. Взаємодія з базою даних здійснюється через SQL запити та ORM з використанням бібліотеки SQLAlchemy.

У модулі передбачено методи для:

- Запису нового запиту.
- Отримання останнього результату.
- Опціонального пошуку історії користувача.

Для збереження тексту GPT відповіді використовується тип TEXT та LONGTEXT, що дозволяє не обмежувати довжину генерацій моделі.

Під час проєктування структури бази даних особливу увагу було приділено забезпеченню цілісності даних, захисту від некоректних записів через типізацію та обмеження полів, розширюваності що передбачає можливість додавання додаткових полів у майбутньому.

Однією з початкових труднощів стало збереження великого обсягу текстової відповіді GPT. Деякі налаштування MySQL мали обмеження на довжину поля VARCHAR, тому було прийнято рішення використовувати тип TEXT, що усунуло обмеження без необхідності зміни загальної логіки збереження. Крім того при тестуванні було виявлено потребу у збереженні дати запиту. Додано поле типу DATETIME, яке автоматично ініціалізується при створенні нового запису.

Модуль models.py реалізує структуровану модель збереження всіх необхідних даних користувача, що дозволяє як формувати поточні відповіді, так і зберігати їх для подальшого використання. Його реалізація забезпечує підтримку вимог до безпеки, надійності та розширюваності, що є критично важливим при роботі з персоніфікованою інформацією в контексті здоров'я та харчування.

Модуль шаблонізації templates

Підкаталог templates є невід'ємною складовою фронтенд-частини веб-застосунку, створеного з використанням фреймворку Flask. У даній директорії розміщуються HTML шаблони, які реалізують користувацький інтерфейс і

відповідають за представлення результатів обчислень, здійснених серверною логікою. У межах архітектури Flask шаблони реалізовано за допомогою механізму Jinja2, потужного засобу для генерації HTML коду з можливістю вставки динамічного вмісту, умовних конструкцій і циклів.

У проєкті реалізовано декілька HTML шаблонів:

- login.html сторінка входу в систему. Містить форму з полями електронної пошти та пароля. Передбачено відображення повідомлень про помилки автентифікації.
- register.html сторінка створення нового облікового запису. Передбачає введення імені користувача, електронної пошти, пароля та підтвердження пароля. Валідація даних проводиться як на клієнтській, так і на серверній стороні.
- meal_plan.html шаблон для відображення згенерованого плану харчування на день, сформованого на основі параметрів користувача і результатів роботи моделі GPT. Включає заголовки прийомів їжі, страви та рекомендовані порції.
- analyze.html сторінка, призначена для виведення аналітичних даних, зведених рекомендацій, узагальнень або висновків.
- log_trends.html шаблон, який відображає динаміку змін параметрів користувача, ваги, калорійного споживання. Дані можуть виводитися у вигляді таблиці або графіків на основі збережених записів.
- profile.html сторінка особистого профілю користувача. Містить вивід його фізіологічних параметрів, індивідуальних цілей та завантаженого аватара. Передбачено функціонал для редагування даних і вибору цілей харчування.
- support.html сторінка з інформацією для зв'язку з розробниками, формою зворотного зв'язку або інструкціями щодо користування сервісом.

- `about.html` сторінка з описом платформи, її мети, реалізації та особливостей. Доступна для всіх користувачів, має структурований контент із прокруткою та оформленням.
- `features.html` шаблон з переліком можливостей системи у вигляді карток, кожен пункт містить посилання на детальніший опис.
- `confidens.html` сторінка політики конфіденційності, де наведено правила обробки персональних даних відповідно до чинного законодавства.

Розгалужена структура шаблонів у каталозі `templates` дозволяє досягти високого рівня інтерактивності, гнучкості та зручності користування системою. Застосування механізмів Jinja2 забезпечує динамічне оновлення вмісту сторінок залежно від введених параметрів, авторизаційного статусу та результатів обчислень. Такий підхід сприяє дотриманню принципів розділення логіки від представлення та полегшує супровід і розвиток веб-застосунку.

Модуль використання OpenAI API

Однією з інноваційних складових створеного веб-застосунку є використання потужної мовної моделі GPT, наданої компанією OpenAI. Інтеграція здійснюється через офіційне API з використанням відповідного Python SDK. Це дозволяє реалізувати генерацію персоналізованих текстових рекомендацій на основі фізіологічних параметрів користувача та обраної мети, підтримки ваги, зниження або набір[29].

Модуль інтеграції розміщується в складі `services.py` і включає наступні компоненти:

- Функцію формування текстового запиту `prompt` на основі вхідних даних.
- Виклик API OpenAI з передачею сформованого запиту.
- Отримання згенерованого тексту у вигляді відповіді моделі.
- Передача результату у HTML-шаблон для відображення на сторінці.

Виклик API відбувається шляхом використання бібліотеки `openai`, яка дозволяє вказувати модель, температуру генерації, максимальну кількість токенів та інші параметри.

Для досягнення релевантної відповіді було реалізовано автоматичне формування запиту у текстовій формі, який включає демографічні дані, рівень фізичної активності, обрану мету, контекст генерації.

При взаємодії з OpenAI API використовуються наступні параметри:

- Model GPT 3.5-turbo.
- Temperature = 0,7, рівень креативності моделі.
- Max_tokens = 800 – обмеження довжини відповіді.
- Stop=None – завершення генерування залежно від контексту.

Після отримання тексту від моделі виконується попередня обробка, видалення зайвих символів та подвійних пробілів. Обрізання або форматування під заданий тип повернення, передача значення у змінну шаблону через `render_template`.

В процесі інтеграції було виявлено низку технічних особливостей, при надто коротких або нечітко сформульованих запитах модель повертала узагальнену відповідь. У відповідях іноді містилися англійські фрази або інтернаціональні одиниці, що вимагало ручного пост оброблення. Через обмеження OpenAI API щодо кількості запитів на хвилину було додано обробку винятків та повідомлення про можливу затримку відповіді.

Інтеграція зі штучним інтелектом на основі OpenAI GPT надала системі унікальної переваги – можливості формувати контекстуальні, гнучкі та персоналізовані рекомендації без потреби вручну програмувати вивідні шаблони. Це дозволяє значно покращити взаємодію з користувачем, підвищити інформативність результатів і розширити сферу застосування проєкту в майбутньому.

Модуль стилізації та використання Bootstrap

Каталог `static` є стандартною частиною структури веб-застосунку на Flask, призначеною для зберігання статистичних ресурсів, таких як каскадні таблиці стилів CSS, зображення та інші медіафайли. У межах реалізованої системи основну роль у візуальному оформленні відіграє CSS-фреймворк Bootstrap, який

забезпечує адаптивність, швидку верстку та уніфікований стиль всіх компонентів інтерфейсу. Каталог містить файл користувацьких стилів для доповнення або перевизначення базових класів Bootstrap. Папку з зображеннями, що використовуються в шапці, на сторінці профілю або у функціоналі генерації. Інтеграцію зовнішніх бібліотек, таких як шрифти або іконки, через CDN або підключення локальних копій.

У проєкті використовується Bootstrap 5, підключений через CDN у базовому шаблоні `base.html`. Цей фреймворк забезпечує:

1) Адаптивну сітку:

- усі сторінки використовують `container`, `row` і `col` для правильного масштабування на різних пристроях;
- блоки зміщуються та вирівнюються завдяки класам `d-flex`, `justify-content center`, `align-items center`.

2) Форми та кнопки:

- стилізація полів форми за допомогою класів `form-control`, `form-label`;
- кнопки виконані з використанням класів `btn`, `btn-primary`, `btn-outline`.

3) Алерти, картки, таблиці:

- сповіщення `alert-danger`, `alert-success` використовуються для відображення помилок або підтвердження;
- таблиці виводу даних, макронутрієнтів, логування змін, реалізовані через `table`, `table-striped`, `table-bordered`;
- блоки можливостей і функцій відображаються як `card` компоненти.

Попри використання Bootstrap, у проєкті реалізовано низку індивідуальних стилістичних рішень. Кольори кнопок та фонів змінені через перевизначення змінних CSS. Створено окремі стилі для блоків з текстом GPT, щоб виділити їх в інтерфейсі. Профіль користувача стилізовано з використанням аватару, підпису, відступів, роздільників.

Усі стилі структуруються централізовано, а нові класи додаються лише у `styles.css`, що забезпечує чистоту шаблонів і легкість підтримки. Завдяки

використанню Bootstrap, веб застосунок набув професійного зовнішнього вигляду з мінімальними витратами часу на розробку інтерфейсу. Поєднання готових компонентів із можливістю гнучкого налаштування забезпечує швидке розгортання, адаптивність під мобільні пристрої та високу зручність користування. Це дозволяє зосередити основні зусилля на функціональній частині системи, зберігаючи при цьому привабливий і сучасний дизайн.

Модуль логування трендів UserTrends i log_trends.html

Важливою складовою функціоналу платформи є фіксація змін у фізіологічних параметрах користувача, зокрема ваги та споживання калорій упродовж певного періоду часу. Це дозволяє відстежувати динаміку, аналізувати ефективність обраної стратегії харчування, а також забезпечити персоналізований підхід до формування майбутніх рекомендацій. Для реалізації цієї функціональності створено модуль логування трендів.

У базі даних реалізовано окрему таблицю UserTrends яка містить поля:

- Унікальний ідентифікатор запису.
- Зовнішній ключ, що посилається на таблицю User.
- Дата внесення запиту.
- Зафіксована вага користувача на конкретну дату.
- Спожиті калорії.

Кожен запис репрезентує одне вимірювання, внесене вручну або автоматично. Веб-інтерфейс реалізовано через шаблон log_trends.html. Сторінка містить:

- Форму для введення нових значень ваги та калорійності.
- Таблицю з попередніми записами.
- Можливість видалення або редагування окремих значень.

Дані виводять в хронологічному порядку, також можлива побудова графіків або інтеграцій з аналітичними модулями.

Фіксація змін дозволяє об'єктивно оцінювати наскільки користувач дотримується рекомендацій. Адаптувати дієтичні поради залежно від реакції

організму та надати індивідуальний аналіз тренду на сторінці `analyze.html`. Модуль `UserTrends` є важливим інструментом для довгострокового відстеження прогресу. Його використання посилює аналітичну складову системи та надає користувачеві можливість бачити об'єктивні зміни в результаті запропонованих дій.

Модуль маршрутизації та підтримка багатосторінковості

У веб застосунку реалізовано повноцінну підтримку багатосторінкової навігації, яка охоплює не лише основний функціонал, вхід, генерація рекомендацій, профіль, а й інформативні сторінки, опис проєкту, політика конфіденційності, можливості сервісу.

Принцип реалізації заключається в тому що на рівні бекенду маршрутизація реалізована у `routes.py` за допомогою декораторів `app.route`, які призначають функції обробники для кожного шляху. Таке рішення дозволяє забезпечити чітке розділення логіки обробки різних запитів, централізовано керувати переходами між сторінками, реалізувати логіку доступу до певних сторінок лише для авторизованих користувачів.

Основна структура меню розміщена у шаблоні `header.html`, який вбудовується в усі сторінки через `base.html`. Меню містить посилання на:

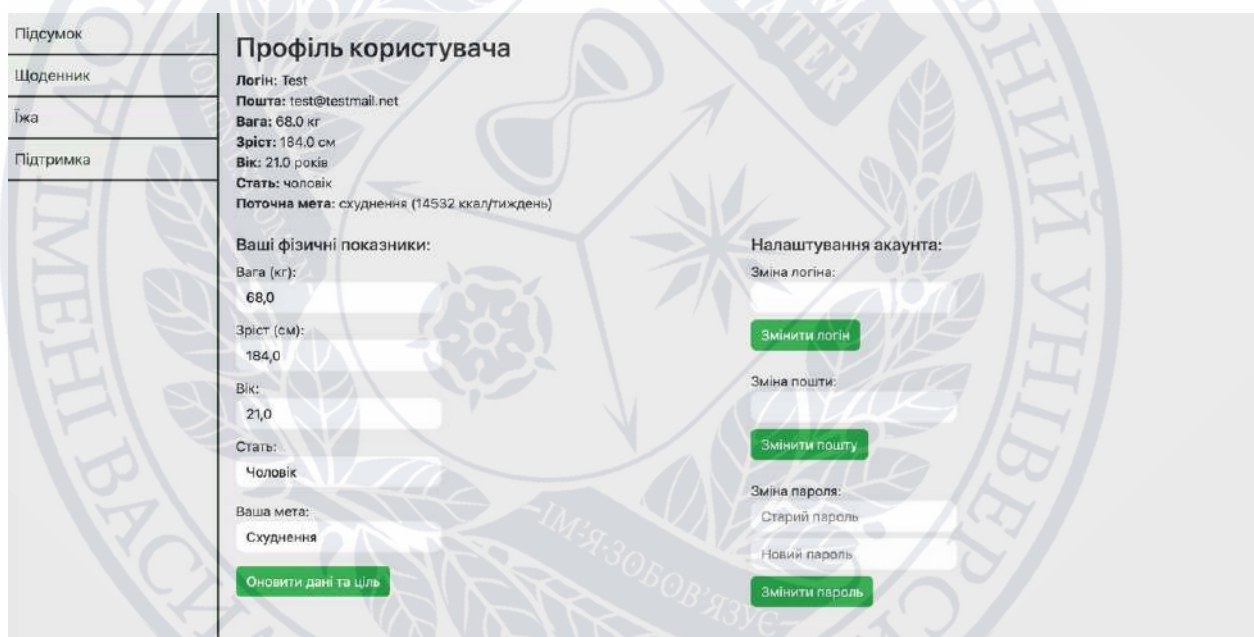
- Головна сторінка.
- Аналіз харчування.
- План харчування.
- Тренди.
- Профіль користувача.
- Інформативні сторінки.

Кожна сторінка розмічена з використанням `Bootstrap` сітки та адаптована до перегляду на мобільних пристроях. Перехід між сторінками виконується миттєво, без перенавантаження стану програми. Крім того реалізована логіка редиректу у разі неавторизованого доступу до захищених сторінок. Підтримка багатосторінковості та чітка маршрутизація забезпечують масштабованість веб

застосунку, гнучкість у розміщенні функціоналу, а також покращують користувацький досвід. Завдяки централізованому управлінню маршрутами система залишається організованою, модульною й придатною до подальшого розширення.

3.3. Результати роботи системи, приклади та демонстрація

Загальний сценарій використання системи користувачем: для початку користувач переходить на головну сторінку сервісу де реєструє особистий акаунт. Після успішної реєстрації користувач заповнює форму з полями, вік, вага, ріст, стать, рівень активності, ціль



Підсумок	Щоденник	Їжа	Підтримка
Профіль користувача Логін: Test Пошта: test@testmail.net Вага: 68.0 кг Зріст: 184.0 см Вік: 21.0 років Стать: чоловік Поточна мета: схуднення (14532 ккал/тиждень) Ваші фізичні показники: Вага (кг): 68,0 Зріст (см): 184,0 Вік: 21,0 Стать: Чоловік Ваша мета: Схуднення Оновити дані та ціль Налаштування акаунта: Зміна логіна: Змінити логін Зміна пошти: Змінити пошту Зміна пароля: Старий пароль Новий пароль Змінити пароль			

Рисунок 3.2 – Демонстрація форми з фізичними показниками користувача

Після чого користувач записує своє харчування, фізичну активність, воду та сон, штучний інтелект в реальному часі опрацьовує інформацію та готує аналіз для користувача.

Підсумок

Щоденник

Їжа

Підтримка

Їжа:

Вправи:

Кількість води (л):

Тривалість сну (год):

Нотатки:

Зберегти

Історія записів

Дата: 2025-05-07

Їжа: На сніданок ячмінна каша 100 грам, один банан, пів огірка, на обід піца з ковбасою та соусом, на вечерю стейк курячий та макарон з огірком.

Вправи: День спини в залі протягом 1 години

Вода: 2.5 л

Сон: 9.0 год

Нотатки: Почуваю себе добре

Видалити запис

Рисунок 3.3 – Демонстрація запису харчування, фізичної активності, води та сну

На сторінці результатів система виводить обчислену добову потребу в калоріях, рекомендований розподіл макронутрієнтів, згенеровану текстову пораду щодо харчування.

Підсумок

Щоденник

Їжа

Підтримка

Ваш профіль

Нора дослідила ваше харчування

Калорії: 1650 ккал; Білки: 81 г;
Вуглеводи: 137 г; Жири: 92 г.
Рекомендації: збалансоване співвідношення білків, жирів та вуглеводів, контроль калорій.

Вода

Середній прийом: 2.5 л
Важливо пити достатню кількість води для забезпечення нормальної роботи організму. 2,5 літра води на день є нормою для багатьох людей.

Сон

Середній сон: 9.0 год
Достатньо повноцінний сон.

Нора дослідила ваше фізичне життя

Важливо розвивати інші групи м'язів. Додай у тренування вправи для ніг, грудей, плечей та кори. Розглянь різноманітні вправи з гантелями, гірками та власною вагою. Розподіли час тренувань рівномірно між всіма групами м'язів.

Рисунок 3.4 – Демонстрація аналізу створеного штучним інтелектом

Також ключовою особливістю розробленої платформи є можливість отримати детальний денний раціон харчування, сформований на основі

індивідуальних параметрів користувача. Цей функціонал реалізовано шляхом інтеграції з моделлю, яка здатна адаптувати поради до конкретного користувача. Користувачу необхідно вказати свої вподобання після чого система автоматично формує текстовий запит на основі актуальної інформації користувача.

Рисунок 3.5 – Демонстрація створення раціону на день

3.4. Тестування та перевірка працездатності системи

Після завершення розробки веб-застосунку було проведено комплексне тестування з метою перевірки функціональності, надійності та коректності обробки даних. Особливу увагу приділено критичним ділянкам системи, модулям розрахунку калорійності, обміну з GPT API, валідації форм, збереженню даних і відображенню результатів.

Види проведеного тестування:

1) Функціональне тестування:

- мета: перевірити чи відповідає поведінка системи її вимогам;
- тестувалися: обробка запитів, логіка розрахунків, генерація раціону, відображення даних.

2) Модульне тестування:

- мета: перевірка окремих функцій;
- застосувалось до функцій з модуля services.py.

3) Тестування форм і валідації:

- перевірялися помилки користувача при введенні, відсутність значення, текст замість числа, некоректна вага тощо.

4) Тестування інтеграції з OpenAI:

- мета: перевірити стабільність відповідей, обробку API помилок, затримок або обмежень на кількість запитів.

5) Перевірка відображення результатів:

- перевірка правильного виводу даних у шаблонах, роботи Bootstrap компонентів, адаптивності інтерфейсу.

Таблиця 3.1 – Сценарії тестування та їх результати

№	Сценарій	Очікуваний результат	Результат тесту
1	Введення валідних параметрів	Розрахунок TDEE, формування меню	Успішно
2	Введення нечислового значення у полі «вага»	Виведення повідомлення про помилку	Успішно
3	Відправка запиту з незаповненими обов'язковими полями.	Блокування дії, попередження для користувача	Успішно
4	Перевищення ліміту prompt до моделі	Виведення повідомлення про помилку	Успішно
5	Надто низька вага	Попередження про недопустимі значення	Успішно
6	Відмова API, збій з'єднання або помилка «not response»	Виведення альтернативного повідомлення про недоступність	Успішно

Під час тестування було виявлено труднощі:

- недостатня перевірка числових значень – на ранньому етапі користувач міг ввести число з плаваючою комою, що призводило до помилки. Проблема вирішена автоматичною заміною коми на крапку;

- відсутність фідбеку при довгому очікуванні відповіді від моделі. Було додано індикатор завантаження та таймаут-затримку;
- некоректне відображення довгих текстів моделі. Було вирішено за допомогою блоків прокрутки в шаблоні та форматування тексту.

Крім функціонального тестування було перевірено відображення сторінок на різних типах пристроїв. Проведене тестування підтвердило стабільність, коректність і відповідність поведінки веб-застосунку заявленим вимогам. Усі критичні компоненти функціонують згідно очікувань, помилки обробляються, інтерфейс залишається зручним незалежно від платформи. Це дає підстави стверджувати про готовність системи до практичного використання.

Висновки до розділу

Було безпосередньо реалізовано та проаналізовано основні етапи створення веб платформи для управління харчуванням із використанням сучасних веб технологій та елементів штучного інтелекту. На початку сформульовано прикладну проблему, обґрунтовано доцільність створення нової системи, визначено її основні функціональні блоки та описано загальну архітектуру рішення. Усі компоненти платформи реалізовано у межах модульної структури, що відповідає вимогам до масштабованих веб систем.

Окремо розглянуто функціонування кожного модуля: маршрутизації, шаблонізації, стилізації, обробки логіки, інтеграції з OpenAI GPT, роботи з базою даних, форм введення параметрів, збереження історії змін тощо. Для підвищення наочності додано діаграми структури бази даних, приклади таблиць та елементів інтерфейсу.

Платформа була протестована за кількома сценаріями, включаючи модульне, функціональне та UX-тестування. За результатами перевірки встановлено коректну роботу всіх компонентів, передбачену реакцію на типові помилки користувача, адаптивність інтерфейсу та стабільну генерацію персоналізованих рекомендацій.

Таким чином, реалізований проєкт відповідає поставленим вимогам, забезпечує цільову функціональність, відзначається стійкістю та зручністю використання.



ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було досягнуто основної мети дослідження, розроблено інтелектуальну веб платформу для управління харчуванням, що дозволяє користувачам ефективно контролювати споживання калорій, отримувати персоналізовані рекомендації та формувати здорові харчові звички. Проєкт охопив усі ключові аспекти сучасних вимог до подібних систем: адаптивність інтерфейсу, науково обґрунтовані алгоритми розрахунків, інтеграцію з моделлю штучного інтелекту, безпеку персональних даних та зручність щоденного використання.

На початковому етапі було проаналізовано актуальність проблеми харчування в умовах сучасного ритму життя. Зібрані статистичні дані та результати медичних досліджень засвідчили, що значна частина населення страждає від наслідків незбалансованого раціону, надмірного або недостатнього споживання нутрієнтів, а також відсутності усвідомленого підходу до формування щоденного меню. Було визначено, що більшість існуючих цифрових рішень обмежуються лише базовим підрахунком калорій або вимірюванням ваги, при цьому не враховуючи індивідуальні потреби користувача, не надаючи гнучких рекомендацій або потребує платного доступу до розширених функцій.

У межах цієї роботи реалізовано повноцінний програмний продукт, який поєднує точність класичних формул розрахунку, зокрема формули Міффіна-Сан Жеора, з гнучкістю генеративної моделі GPT. Завдяки цьому, система здатна не лише розрахувати добову норму калорій, а й сформулювати текстову пораду та приклад формування раціону доступним навіть для людей без спеціальних знань у галузі дієтології.

Особливу увагу було приділено архітектурі веб застосунку. Обрано легкий і гнучкий стек технологій: python з фреймворком Flask, шаблонізатор Jinja2, база даних MySQL, фреймворк стилізації Bootstrap та API OpenAI. Така конфігурація

дозволила швидко впровадити як базові функції, введення параметрів, авторизацію, збереження історії, так і просунуті можливості, зокрема інтеграцію зі штучним інтелектом та динамічну генерацію персональних рекомендацій. Реалізовано адаптивний інтерфейс, що забезпечує зручну роботу з мобільних пристроїв, а також передбачено можливість масштабування системи в майбутньому.

У ході тестування платформи було перевірено її функціональність на всіх етапах, від введення даних до генерації відповідей моделлю. Усі компоненти системи успішно пройшли функціональне, модульне та UX-тестування. Система стабільно працює навіть за умов некоректного введення даних користувачем, має логіку валідації, інформує про помилки, а також дає можливість повторної генерації меню.

Проект також відповідає сучасним вимогам до безпеки, реалізовано хешування паролів, контроль введених значень, обмеження доступу до критичних даних. При подальшому масштабуванні система може бути адаптована під вимоги міжнародних стандартів захисту персональної інформації.

У результаті виконаної роботи було не лише досягнуто технічної реалізації системи, а й створено основу для її практичного використання. Сервіс може бути застосований у повсякденному житті для ведення харчового щоденника, підбору меню, самоконтролю або як прототип для масштабнішого продукту у сфері нутриціології чи медичної підтримки. Така система здатна підвищити поінформованість користувачів про власне харчування, поліпшити їхні звички та зробити процес турботи про здоров'я більш системним і зрозумілим.

Проект демонструє, що використання штучного інтелекту у сфері здоров'я не обов'язково потребує глибокої технічної або медичної підготовки з боку користувача. Завдяки інтерфейсу, який інтуїтивно зрозумілий, подібний до консультації з дієтологом, усього за кілька кроків. Це відкриває перспективи для

подальшого розвитку платформи як інструменту первинної цифрової допомоги в межах персонального здоров'я.

Виконана робота також демонструє практичне застосування знань, здобутих у процесі навчання. Проєкт об'єднав окремі дисципліни, програмування, бази даних, безпеку, штучний інтелект, веб розробку, в єдине цілісне рішення. Такий досвід має вагоме значення для подальшого професійного становлення розробки, аде він охоплює цикл створення, тестування та впровадження продукту.

Таким чином, виконане дослідження і програмна реалізація підтверджують доцільність розробки інтелектуального веб додатку для управління харчуванням. Отримані результати є не лише технічно цінними, а й соціально значущими, адже вони сприяють підвищенню якості життя користувачів через надання доступного інструменту для самоконтролю та підтримки здорового способу життя.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дні нутриціології та дієтології в Одесі. *МедЕксперт* : Науково-практ. конф., м. Одеса, 6 берез. 2025 р.
2. Здорове харчування від дитинства до довголіття: комплексний підхід, стан та перспективи. м. Київ, 12 жовт. 2023 р. Київ, 2023. С. 24–27.
3. Львівський національний медичний університет імені Данила Галицького. методичні рекомендації з нутриціології. URL: [https://new.meduniv.lviv.ua/uploads/repository/kaf/kaf_prophylactictox/02.Навчальні_програми/Student`s/Method.%20Nytriciologia%20Medicine.pdf](https://new.meduniv.lviv.ua/uploads/repository/kaf/kaf_prophylactictox/02.Навчальні_програми/Student`s/Method.%20Nytriciologia%20Medicine.pdf) (дата звернення: 21.04.2025).
4. Міністерство охорони здоров'я України. Порівнюйте свою їжу з «тарілкою здорового харчування». URL: <https://moz.gov.ua/uk/porivnjajte-svoju-izhu-z-tarilkoju-zdorovogo-harchuvannj> (дата звернення: 21.04.2025).
5. Міністерство охорони здоров'я України. рекомендації щодо здорового харчування. URL: <https://moz.gov.ua/uk/scho-i-jak-isti-schob-zhiti-dovgo-rekomendacii-moz> (дата звернення: 17.04.2025).
6. Міністерство охорони здоров'я України. як обрати ефективну та безпечну дієту: 5 порад. URL: <https://moz.gov.ua/uk/jak-obrati-efektivnu-ta-bezpechnu-dietu-5-zagalnih-porad> (дата звернення: 21.04.2025).
7. Національний університет біоресурсів і природокористування України. дні нутриціології та дієтології у Львові 2023. URL: <https://nubip.edu.ua/node/133288> (дата звернення: 20.04.2025).
8. Простий спосіб підрахунку калорій. як працює український ші-застосунок mealmate AI // mc.today. - 2024. URL: <https://mc.today/uk/prostij-sposib-pidrahunku-kalorij-yak-pratsyuye-ukrayinskij-shi-zastosunok-mealmate-ai/> (дата звернення: 22.04.2025).

9. Центр громадського здоров'я МОЗ України. Проєкт «Проздорове». URL: <https://phc.org.ua/promociya-zdorovya/proekt-prozdorove> (дата звернення: 18.04.2025).
10. Центр громадського здоров'я МОЗ України. Проєкт «П'ять кроків до здорового харчування». URL: <https://dp.cdc.gov.ua/news/uchast-du-dnipropetrovskyj-otskph-moz-v-proyekti-p-yat-kroktiv-do-zdorovogo-harchuvannya/> (дата звернення: 17.04.2025).
11. Штучний інтелект в харчовій промисловості / В. Ю. Бондаренко // researchgate. - 2023. URL: https://www.researchgate.net/publication/384481368_STUCNIJ_INTELEKT_V_HARCOVIJ_PROMISLOVOSTI (дата звернення: 22.04.2025).
12. Штучний інтелект є перспективним інструментом для отримання інформації про харчування // Національний університет біоресурсів і природокористування України - 2023. URL: <https://nubip.edu.ua/node/141833> (дата звернення: 21.04.2025).
13. ISO/IEC 27001:2022. Система управління інформаційною безпекою. URL: <https://oleg-dubetcky.medium.com/iso-iec-27001-2022-система-управління-інформаційною-безпекою-db24615a276d> (дата звернення: 25.04.2025).
14. Проєкт «Здорове харчування» - Центр громадського здоров'я МОЗ. URL: <https://phc.org.ua> (дата звернення: 10.03.2025).
15. Проєкт USAID HealthLink. Роль IT у формуванні раціону. URL: <https://healthlink.ua> (дата звернення: 10.03.2025).
16. Hojung Lee. Personalized nutrition via machine learning: Current state and future prospects. *Nutrients*. 2022. Vol. 14, no. 5. 1005.
17. George Wilson. Estimating Energy Balance: Methods and Challenges. *Journal of Nutrition Science*. 2021. Vol. 10. P. 1–12. URL: <https://www.cambridge.org/core/journals/journal-of-nutritional-science/article/estimating-energy-balance-methods-and->

- [challenges/31C7D53E9FAF4FCB82EDDD891E3D7B97](https://doi.org/10.1016/j.compbiomed.2020.103843) (date of access: 12.04.2025).
18. Yu Chen. Food recognition using convolutional neural networks. *Computers in Biology and Medicine*. 2020. Vol. 122. 103843. URL: <https://doi.org/10.1016/j.compbiomed.2020.103843> (date of access: 13.04.2025).
 19. ISO/IEC 27001:2022. Information Security Management Systems – Requirements. Replaces ISO/IEC 27001:2013 ; effective from 2022-10-25. Official edition. Geneva : ISO, 2022. 31 p.
 20. Laura Brown. Impact of Data-Driven Diet Apps on Weight Loss. *JMIR mHealth and uHealth*. 2020. Vol. 8, no. 3. e14830.
 21. Tuan Nguyen. Nutrient Recommendation Systems Based on AI. *Artificial Intelligence in Medicine*. 2023. Vol. 136. 102434.
 22. Smeets E. Improving dietary adherence with AI coaching. *International Journal of Behavioral Nutrition and Physical Activity*. 2021. Vol. 18, no. 1. P. 45.
 23. Rouskas K. The Influence of an AI-Driven Personalized Nutrition Program on the Human Gut Microbiome and Its Health Implications. *Nutrients*. 2025. Vol. 17, no. 7. P. 1260.
 24. OpenAI. GPT-4 Technical Report. 2023. URL: <https://openai.com/research/gpt-4> (date of access: 11.04.2025).
 25. Google Health. AI in clinical nutrition and disease prevention. URL: <https://health.google> (date of access: 11.04.2025).
 26. Lifesum App: Nutrition & Meal Planner. URL: <https://www.lifesum.com> (date of access: 07.04.2025).
 27. MyFitnessPal. Calorie Counter & Diet Tracker. URL: <https://www.myfitnesspal.com> (date of access: 07.04.2025).
 28. Cronometer. Nutrition Tracker App. URL: <https://cronometer.com> (date of access: 07.04.2025).
 29. FatSecret. Free Calorie Counter. URL: <https://www.fatsecret.com> (date of access: 07.04.2025).

30. Google Fit. Health Tracking App. URL: <https://www.google.com/fit> (date of access: 07.04.2025).
31. Римар П.В., Ліваковський В.К. Розробка платформи для управління харчуванням з інтеграцією штучного інтелекту. *Вісник Східноукраїнського національного університету імені Володимира Даля* 2025.
32. Ліваковський В.К., Римар П.В. Розробка платформи для управління харчуванням з інтеграцією штучного інтелекту. *Прикладні інформаційні технології 2025*: Матеріали всеукр. науково-практ. конф. здобувачів вищ. освіти та молодих вчен., м. Вінниця, 22 трав. 2025р. 2025.

