

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

До захисту допущено:

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 2025 р.

Дипломний проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Інформаційне забезпечення
робототехнічних систем»
спеціальності 126 «Інформаційні системи та технології»
на тему: «Інформаційні системи та технології»
на тему: «Інтелектуальна робототехнічна система для персоналізованого
підбору харчових добавок»

Виконав:

студент IV курсу, групи ІК-12

Новіков Нікіта Максимович _____

Керівник:

асистент кафедри ІСТ

Коломоєць Сергій Олексійович _____

Рецензент:

професор каф. ШІ, д.т.н., професор

Чумаченко Олена Іллівна _____

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____

Київ – 2025 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційне забезпечення робототехнічних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 2025 р.

ЗАВДАННЯ
на дипломний проєкт студенту
Новікову Никіті Максимовичу

1. Тема проєкту «Інтелектуальна робототехнічна система для персоналізованого підбору харчових добавок», керівник проєкту асистент кафедри ІСТ Коломиець С.О., затверджені наказом по університету від «23» травня 2025 р. № 1705-с
2. Термін подання студентом проєкту: «9» червня 2025р.
3. Вихідні дані до проєкту: Рекомендації з точністю $\geq 90\%$, час відповіді AI ≤ 30 с, сумісність із ESP та WS2812, підтримка 9 LED, , OLED-дисплей 128×64, Wi-Fi зв'язок 2.4 ГГц, об'єм бази знань ≥ 10 тис. записів, підтримка до 7 днів.
4. Зміст пояснювальної записки:
 - 1) Аналіз предметної області та обґрунтування підходу до персоналізації підбору добавок
 - 2) Проєктування інтелектуальної системи із врахуванням сучасних технологій та AI-методів
 - 3) Реалізація програмних і апаратних компонентів, опис інтерфейсу та логіки роботи
 - 4) Оцінка ефективності рішень, тестування системи та формування висновків

5. Перелік графічного матеріалу :

- 1) Діаграма архітектури та взаємодії модулів
- 2) ER-діаграми
- 3) Діаграма прецидентів
- 4) Схема підключення компонентів до ESP8266
- 5) Макети вебінтерфейсу (UI)

6. Дата видачі завдання : «14» березня 2025р.

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Аналіз предметної області, формулювання задачі	20.04.2025	
2	Вибір архітектури, визначення технологій	27.04.2025	
3	Розробка структури бази даних, макетування інтерфейсу	04.05.2025	
4	Створення backend та API, автентифікація	11.05.2025	
5	Інтеграція frontend та backend, навчання AI-моделі	18.05.2025	
6	3D-моделювання корпусу пристрою, підбір компонентів	25.05.2025	
7	Реалізація ESP8266, підключення OLED, LED	01.06.2025	
8	Тестування, оформлення ПЗ, підготовка до захисту	09.06.2025	

Студент

Нікіта НОВІКОВ

Керівник

Сергій КОЛОМОЄЦЬ

АНОТАЦІЯ

Інтелектуальна робототехнічна система для персоналізованого підбору харчових добавок.

Дипломний проєкт містить 70 сторінок пояснювальної записки, 38 рисунків, 1 таблиці, 5 креслеників, 4 додатка та 17 інформаційні джерела.

Ключові слова: харчові добавки, штучний інтелект, робототехнічна система, персоналізація, рекомендує система, LLM, векторна база знань, Vue.js, ESP8266, OLED-дисплей.

Об'єкт дослідження — система персоналізованого підбору харчових добавок.

Мета роботи — створити інтелектуальну систему, яка аналізує індивідуальні параметри користувача (вік, стать, симптоми тощо) і формує список рекомендованих харчових добавок з графіком їх прийому, реалізованим через фізичний робототехнічний модуль.

У роботі застосовано методи машинного навчання, технології LLM (Large Language Models), локальний RAG (Retrieval-Augmented Generation), обробка наукових джерел PubMed, а також вебтехнології (Vue.js, Node.js, MongoDB) і мікроконтролер ESP8266 з OLED-екраном.

Результатом є повністю функціональний прототип інтелектуальної системи з вебінтерфейсом, внутрішньою базою даних, модулем персоналізованих AI-рекомендацій та апаратною реалізацією графіка прийому у вигляді світлодіодної підсвітки коробки з сапліментами.

Розробку може бути використано у сфері здоров'я, нутриціології, а також у smart-медицини або споживчої електроніки.

SUMMARY

Intelligent Robotic System for Personalized Dietary Supplement Selection.

The thesis project contains 70 pages of explanatory note, 38 figures, 1 tables, 5 technical drawings, 4 appendices, and 17 information sources.

Keywords: dietary supplements, artificial intelligence, robotic system, personalization, recommendation system, LLM, vector knowledge base, Vue.js, ESP8266, OLED display.

The object of the research is a system for personalized dietary supplement selection.

The goal of the project is to develop an intelligent system that analyzes a user's individual parameters (age, gender, symptoms, etc.) and generates a personalized list of recommended supplements with a schedule for intake, implemented through a physical robotic module.

The work utilizes machine learning methods, LLM technologies (Large Language Models), local RAG (Retrieval-Augmented Generation), scientific sources from PubMed, as well as web technologies (Vue.js, Node.js, MongoDB) and an ESP8266 microcontroller with an OLED display.

The result is a fully functional prototype of an intelligent system with a web interface, internal database, personalized AI recommendation module, and a hardware schedule display via LED lighting in a supplement pillbox.

The development can be applied in healthcare, nutrition, smart medicine, or consumer electronics.

[illegible]

№ рядка	Формат	Позначення	Найменування	Кіл. аркушів	№ екз.	Примітка
29						
30	A3	ІК12.200БАК.006 С1	Інтелектуальна робототехнічна	4		
31			система для персоналізованого			
32			підбору харчових добавок.			
33			Схема таблетниці			
34						
35						
36						
37						
38						
39						
40						
41						
42						
43						
44						
45						
46						
47						
48						
49						
50						
51						
52						
53						
54						
55						
56						
57						
Дата					ІК12.200БАК.006 ТП	
Зм.	Лист	№ докум.	Підпис	Дата		
					Арк.	
					2	

**Пояснювальна записка
до дипломного проєкту
на тему: «Інтелектуальна робототехнічна система
для персоналізованого підбору
харчових добавок»**

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ..... 4

ВСТУП..... 5

1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ..... 7

1.1 Постановка задачі..... 7

1.2 Мотивація та аналіз задачі 8

1.3 Аналіз предметної області..... 10

1.4 Обране середовище 11

1.5 Вибір архітектури..... 13

2 РОЗРОБКА ІНТЕЛЕКТУАЛЬНОГО МОДУЛЯ..... 15

2.1 Підхід до персоналізації в підборі харчових добавок 15

2.2 Збір і підготовка даних з PubMed 16

2.3 Побудова локальної LLM-моделі для рекомендацій..... 18

2.4 Інтеграція LLM-моделі з векторною базою знань (RAG-підхід) 19

2.5 Оцінка результатів рекомендацій 20

3 РОЗРОБКА ВЕБ-СИСТЕМИ ТА ІНТЕРАКТИВНОГО ІНТЕРФЕЙСУ 23

3.1 Архітектура фронтенду (Vue.js)..... 23

3.2 Структура бекенду (Node.js + MongoDB + JWT)..... 26

3.3 Авторизація та профіль користувача..... 28

3.4 Інтерфейс підбору, додавання та перегляду добавок 30

3.5 Сторінка рекомендацій на основі AI 31

3.6 Сторінка створення персональних рекомендацій 35

3.7 Побудова розкладу прийому добавок 36

3.8 База даних..... 36

3.9 Механізм збереження даних користувача 38

					ІК12.200БАК.006 ПЗ			
Зм.	Лист	№ докум.	Підпис		Інтелектуальна робототехнічна система для персоналізованого підбору харчових добавок Пояснювальна записка			
Розробив	Новіков Н.М.							
Перевірив	Коломоець С.О							
Затв.								
					Літ.	Арк.	Аркушів	
					Т		2	70
					КПІ ім. Ігоря Сікорського Група ІК-12			

4. ІНТЕГРАЦІЯ З РОБОТОТЕХНІЧНИМ МОДУЛЕМ	41
4.1 Обґрунтування використання фізичного пристрою	41
4.2 Алгоритм управління робототехнічним модулем.....	42
4.3 Підключення та конфігурація елементів пристрою.....	45
4.4 Реалізація фізичного корпусу пристрою на основі 3D-моделі	47
5. ЕКСПЕРИМЕНТАЛЬНА ОЦІНКА ТА ТЕСТУВАННЯ	51
5.1 Комплексне тестування функціональності системи	51
5.2 Оцінка якості рекомендацій	54
5.3 Реалізація Адмін панелі	59
5.4 Стрес-тестування роботи системи	61
5.5 Можливості масштабування системи	63
ВИСНОВКИ	66
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	69

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

AI (Artificial Intelligence) — Штучний інтелект.

LLM (Large Language Model) — Велика мовна модель.

RAG (Retrieval-Augmented Generation) — Генерація з підключенням до бази знань.

JWT (JSON Web Token) — Токен авторизації.

ESP (Espressif Microcontroller) — Мікроконтролер Wemos D1 mini.

WS2812 (RGB LED) — Адресний світлодіод.

OLED (Organic LED) — Компактний дисплей.

Auth0 (Authentication Service) — Сервіс автентифікації.

MongoDB (Database) — NoSQL база даних.

Vue.js (Frontend Framework) — Фреймворк інтерфейсу.

Node.js (Backend Runtime) — Серверне середовище.

Wi-Fi (Wireless Connection) — Бездротовий зв'язок.

API (Application Interface) — Інтерфейс обміну даними.

CRUD (Create, Read, Update, Delete) — Основні дії з даними.

					ІК12.200БАК.006 ПЗ	Арк.
						4
Зм.	Лист	№ докум.	Підпис	Дата		

ВСТУП

У сучасному світі все більше людей прагнуть слідкувати за власним здоров'ям та покращувати якість життя шляхом вживання харчових добавок. Водночас стрімке зростання кількості доступних саплементів, відмінностей у їх складі, ефектах та сферах застосування призводить до плутанини навіть серед освічених споживачів. Відсутність персоналізованих порад, що базуються на наукових дослідженнях, часто стає причиною неефективного або навіть шкідливого вживання добавок. Саме тому зростає потреба у системах, здатних надати індивідуальні рекомендації на основі актуального стану здоров'я користувача, його фізичних параметрів, мети та супутніх симптомів.

Штучний інтелект і, зокрема, великі мовні моделі відкривають нові можливості в цій сфері. Їх здатність аналізувати великі обсяги текстових медичних даних, таких як наукові статті з бази PubMed, дозволяє створювати рішення, що пропонують обґрунтовані та релевантні поради у сфері нутриціології. Окрім того, інтеграція таких інтелектуальних систем із вебінтерфейсами забезпечує зручну взаємодію з користувачем, а додаткове підключення робототехнічних модулів для фізичної видачі підібраних добавок відкриває шлях до створення повноцінних автономних станцій здоров'я майбутнього.

У цьому дипломному проєкті описується створення інтелектуальної робототехнічної системи для персоналізованого підбору харчових добавок, яка поєднує в собі інструменти штучного інтелекту, сучасні вебтехнології та базу даних користувача. Система аналізує дані про самопочуття, вік, вагу, стать, цілі користувача і формує індивідуальні рекомендації щодо саплементів. Модель донавчена на наукових публікаціях із авторитетних джерел, а результат її роботи візуалізується на вебсайті, де користувач може додати рекомендовані добавки до особистого списку. Додатково розроблено графік прийому та модуль для контролю виконання рекомендацій.

Результатом реалізації є повноцінна інтелектуальна система, яка враховує потреби конкретної особи, зменшує ризики невірної вибору, автоматизує рутинні процеси та ілюструє потенціал міждисциплінарного підходу в медицині майбутнього. Актуальність цієї теми обумовлена загальносвітовою тенденцією до персоналізації, розвитком технологій AI, а також необхідністю пошуку ефективних способів підтримки здоров'я в умовах високих навантажень та стресу. У межах роботи здійснено повний цикл — від збору даних і побудови моделі до створення програмного забезпечення і інтеграції з фізичним пристроєм. Усе це доводить практичну цінність та інноваційність запропонованого підходу.

					ІК12.200БАК.006 ПЗ	Арк.
						6
Зм.	Лист	№ докум.	Підпис	Дата		

1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Постановка задачі

Сучасне суспільство все більше звертає увагу на якість життя, профілактику захворювань та підтримку здоров'я. Одним із важливих інструментів у цьому напрямі стали харчові добавки, які дозволяють компенсувати нестачу поживних речовин, підтримати роботу організму за умов підвищених навантажень або хронічних проблем. Водночас, зростання кількості доступних добавок ускладнило процес їхнього вибору: користувачі часто стикаються з відсутністю індивідуальних рекомендацій, недостатнім рівнем знань про вплив різних речовин та плутаниною в маркетингових описах препаратів. Виникає потреба у створенні системи, яка б допомагала формувати обґрунтовані рішення щодо вибору саплементів на основі наукових даних та індивідуальних характеристик людини.

У цьому контексті дипломний проєкт присвячений розробці інтелектуальної робототехнічної системи, яка дозволяє здійснювати персоналізований підбір харчових добавок. Основна мета полягає в тому, щоб поєднати можливості штучного інтелекту, вебтехнологій і медичної науки для формування рекомендацій, адаптованих до конкретного користувача. Система має приймати на вхід дані про вік, вагу, ріст, стать, поточне самопочуття, симптоми, цілі користувача та на основі аналізу медичних досліджень пропонувати відповідні саплементи. Додатково передбачається можливість інтерактивної взаємодії з системою: користувач може зберігати свої добавки, створювати графік їх прийому та редагувати профіль.

Технічно проєкт реалізується у вигляді вебзастосунку, де реалізовано автентифікацію, профіль користувача, аналітичну систему рекомендацій, управління базою добавок та механізм візуального відображення розкладу прийому. Також застосовується штучний інтелект, натренований на текстах з бази даних PubMed, для генерації відповідей та підбору добавок з урахуванням останніх наукових даних.

Додатково створюється апаратна частина на базі ESP8266 з OLED-дисплеєм і LED-індикаторами для наочності виконання рекомендацій.

Таким чином, сформульована задача охоплює створення комплексної системи, що інтегрує сучасні інструменти інтелектуальної обробки інформації, зручний користувацький інтерфейс та індивідуальний підхід до підтримки здоров'я.

1.2 Мотивація та аналіз задачі

Проблема збереження здоров'я та підвищення якості життя дедалі більше стає предметом уваги сучасного суспільства, особливо в умовах хронічного стресу, погіршення екології, малорухливого способу життя, незбалансованого харчування та зростання психоемоційного навантаження. Люди все частіше звертаються до харчових добавок як до інструменту профілактики, відновлення сил або підвищення працездатності. Проте ринок сапліментів є надзвичайно насиченим і хаотичним: існують тисячі найменувань продуктів, з різними механізмами дії, ступенем ефективності та рівнем наукової перевіреності. У такій ситуації користувачам надзвичайно складно зробити обґрунтований вибір.

Більшість існуючих інструментів для вибору добавок базуються або на стандартних анкетах, або на маркетингових підходах, які не враховують реального фізіологічного стану конкретної особи. Вони ігнорують такі важливі аспекти, як вік, стать, поточне самопочуття, анамнез, спосіб життя, рівень стресу, хронічні симптоми чи цілі користувача — наприклад, покращення сну, зниження тривожності, підвищення імунітету чи концентрації уваги. Через це споживачі або повністю відмовляються від добавок, або приймають їх наосліп, що потенційно шкодить здоров'ю.

Ще одним викликом є величезний обсяг інформації. Щороку публікуються тисячі наукових статей, які стосуються дії окремих нутрієнтів або харчових добавок на організм. Ці дослідження мають різний рівень доказовості, часто є

складними для інтерпретації неспеціалістом і не охоплюються жодним доступним користувачу сервісом.

Тому постає потреба в інструменті, який би міг обробляти такі дані автоматично, витягувати з них найрелевантнішу інформацію, фільтрувати за достовірністю та пов'язувати із запитами користувача.

У зв'язку з цим виникає ідея використати сучасні можливості штучного інтелекту — зокрема великі мовні моделі (LLM), здатні працювати з неструктурованими текстовими даними, розуміти контекст, витягувати знання з наукових джерел і генерувати висновки у зручному для користувача форматі. Завдяки донавчанню таких моделей на базі медичних публікацій (наприклад, з PubMed) можна побудувати експертну систему, яка буде давати якісні рекомендації на основі сучасних наукових доказів, а не загальних порад з інтернету.

Крім суто програмної частини, з'являється запит на автоматизацію повного циклу — від аналізу до фізичної дії. Уявлення про те, що система не лише рекомендує добавки, а й дозволяє автоматизовано видавати їх у відповідності до персонального графіка, відкриває новий рівень зручності та персоналізованої турботи про здоров'я. Це особливо актуально для людей з великою зайнятістю, похилого віку або хронічними станами, яким важливо не пропустити прийом.

Таким чином, мотивація даної роботи ґрунтується на кількох взаємопов'язаних факторах: зростаючій потребі в індивідуальному підході до здоров'я, складності прийняття обґрунтованих рішень на ринку сапліментів, відсутності зручних і надійних сервісів персоналізованих рекомендацій, а також появі технічної можливості реалізувати подібну систему на основі локальної AI-моделі та недорогого роботизованого модуля. Це не лише дає користувачеві зручність, а й формує нову культуру відповідального ставлення до здоров'я через цифрові технології.

1.3 Аналіз предметної області

Предметна область, у межах якої реалізується дана система, поєднує кілька міждисциплінарних напрямів: інтелектуальні інформаційні системи, персоналізована медицина, біомедичні знання, обробка природної мови (NLP), веброзробка та робототехніка. В її основі лежить прагнення до автоматизації процесу підбору харчових добавок з урахуванням індивідуальних особливостей користувача та сучасних наукових даних. Сучасна медицина дедалі більше рухається в бік персоналізованих підходів, які враховують не лише загальні показники здоров'я, а й спосіб життя, генетику, поведінкові фактори, рівень стресу, харчування та навіть мікробіом. У той же час ринок харчових добавок не встигає адаптуватися під цю концепцію, залишаючи користувачів сам на сам з хаотичними списками препаратів і відсутністю якісної навігації.

Зростаюча кількість наукових публікацій з тематики нутрицевтики, функціональних речовин, впливу вітамінів, мінералів та рослинних екстрактів на фізіологічні процеси створює новий пласт знань, який практично недоступний для широкого загалу через свою складність і обсяг. Простий користувач, навіть за наявності вільного часу та мотивації, не здатен вручну опрацювати сотні публікацій, порівняти їхню достовірність, зробити логічні висновки та прив'язати їх до власного стану. Саме тут виникає потреба у використанні LLM — моделей, здатних «читати» ці тексти, виявляти релевантні фрагменти, витягувати ключову інформацію та пов'язувати її з індивідуальними запитами.

Предметна область також охоплює процес збору вхідних даних від користувача: демографічних (вік, стать, вага, зріст), симптоматичних (втома, тривожність, безсоння, проблеми з травленням тощо), а також декларативних —

тобто цілей, які користувач хоче досягти (підвищити продуктивність, зміцнити імунітет, покращити сон, зменшити тривогу тощо). Усі ці фактори є складовими персонального профілю і формують запит до системи, який надалі обробляється інтелектуальним модулем.

Крім того, розглядається й візуальна складова предметної області — спосіб, у який користувач взаємодіє з системою. Сучасний користувач очікує не лише точних рекомендацій, але й приємного та зрозумілого інтерфейсу. Тому важливою є реалізація UX/UI-дизайну, який дозволяє інтуїтивно працювати з добавками: переглядати інформацію у вигляді карток, додавати їх до свого списку, редагувати, створювати розклад прийому, відслідковувати стан здоров'я тощо.

Крім того, розглядається й візуальна складова предметної області — спосіб, у який користувач взаємодіє з системою. Сучасний користувач очікує не лише точних рекомендацій, але й приємного та зрозумілого інтерфейсу. Тому важливою є реалізація UX/UI-дизайну, який дозволяє інтуїтивно працювати з добавками: переглядати інформацію у вигляді карток, додавати їх до свого списку, редагувати, створювати розклад прийому, відслідковувати стан здоров'я тощо.

Ще одним аспектом предметної області є зв'язок з фізичним світом — реалізація розумної коробки або пристрою, який може підсвічувати або видавати сапліменти згідно з персональним графіком. Це не лише підвищує дисципліну користувача, а й створює елемент автоматизації в побуті, подібно до "розумного будинку". Таким чином, предметна область охоплює й IoT-компонент: збереження даних у мікроконтролері, відображення інформації на OLED-дисплеї, роботу з LED-індикаторами, управління через кнопки та з'єднання по Wi-Fi.

Таким чином, аналіз предметної області свідчить про багатовимірність завдання: тут поєднуються знання з медицини, програмування, UX-дизайну, штучного інтелекту, вебтехнологій та апаратної реалізації. Саме така комплексність дозволяє реалізувати інноваційний продукт, який не лише генерує рекомендації, але й надає повний цикл підтримки користувача — від діагностики до фізичної дії.

1.4 Обране середовище

Для реалізації інтелектуальної системи персоналізованого підбору харчових добавок було обрано сучасне, перевірене та збалансоване середовище розробки,

яке дозволяє поєднати вебінтерфейс, штучний інтелект, роботу з базою даних та інтеграцію з фізичним пристроєм. Ключовими критеріями при виборі середовища були: відкритість і доступність інструментів, гнучкість масштабування, активна спільнота підтримки, наявність бібліотек для швидкої реалізації складних функцій, а також можливість роботи в локальному середовищі без прив’язки до хмарних сервісів.

Клієнтська частина (фронтенд) реалізована за допомогою фреймворку Vue.js, що дозволяє створювати швидкий, реактивний і адаптивний інтерфейс. Його компонента структура забезпечує зручне управління станом, повторне використання блоків та легку інтеграцію з бекендом. Для стилізації використовується Tailwind CSS, який дозволяє забезпечити єдиний стиль і швидке оформлення візуальних компонентів інтерфейсу.

Серверна частина (бекенд) розроблена з використанням Node.js з Express.js як HTTP-сервером. Такий підхід забезпечує асинхронність обробки запитів, високу швидкодію, а також зручну реалізацію REST API для зв’язку з клієнтською частиною. Усі дані користувачів, включно з інформацією про добавки, графік прийому, профіль та інші параметри, зберігаються в MongoDB — документно-орієнтованій NoSQL базі даних, яка добре підходить для зберігання гнучких та вкладених структур, властивих персоналізованим застосункам. Для взаємодії з базою даних використовується бібліотека Mongoose, яка дозволяє описувати схеми, проводити валідацію та реалізовувати складну логіку обробки даних.

Аутентифікація користувачів реалізована за допомогою JWT (JSON Web Token) — надійного способу створення токенів доступу, які зберігаються на клієнті та передаються до сервера при кожному запиті. Це дозволяє реалізувати захищену авторизацію без постійного зберігання сесій на сервері, що робить систему масштабованою та зручною в обслуговуванні.

Модуль штучного інтелекту реалізований локально за допомогою моделі medllama2:7b, розгорнутої в середовищі Ollama. Такий підхід дозволяє забезпечити повну автономність роботи системи без залежності від зовнішніх API, уникнути передачі чутливих даних до сторонніх серверів, а також контролювати ресурси

моделі, швидкість відповіді та донавчання. Модель працює у зв'язці з векторною базою знань, сформованою на основі наукових статей з PubMed, що дозволяє реалізувати RAG-підхід до генерації відповідей.

Для реалізації візуального плану прийому добавок і взаємодії з фізичним пристроєм використано мікроконтролер Wemos D1 mini (ESP8266), що підтримує підключення до Wi-Fi, роботу з OLED-екраном, LED-індикаторами (WS2812), а також двома кнопками керування. Комунікація між вебзастосунком і пристроєм здійснюється через HTTP-запити у форматі JSON, що забезпечує просту та уніфіковану структуру даних.

Таким чином, вибране середовище є повністю кросплатформеним, побудоване на відкритих технологіях, що дозволяє легко розгорнути систему як локально, так і в хмарі, а також у подальшому масштабувати її до мобільного застосунку, публічного сервісу чи інтеграції з електронною медичною картою користувача.

1.5 Вибір архітектури

Архітектура системи сформована на основі принципів модульності, масштабованості та взаємної ізольованості компонентів. Обрано багаторівневу клієнт-серверну архітектуру, де кожен рівень відповідає за окрему функціональність і може розвиватися незалежно. Основна мета архітектурного підходу — забезпечити ефективну обробку даних, швидку взаємодію з користувачем, підтримку персоналізації й можливість інтеграції з фізичними пристроями.

На верхньому рівні архітектури знаходиться фронтенд — інтерфейс користувача, реалізований як односторінковий додаток (SPA). Він відповідає за збирання введених даних, візуалізацію результатів, управління профілем, відображення рекомендацій і графіка прийому добавок.

Наступний рівень — це серверна частина (backend), яка виступає логічним центром системи. Тут зосереджені маршрути, контролери, сервіси та обробка

					ІК12.200БАК.006 ПЗ	Арк.
						13
Зм.	Лист	№ докум.	Підпис	Дата		

авторизації. Сервер відповідає за зберігання даних у базі, формування запитів до AI-модуля, перевірку прав доступу, обробку запитів користувача та зв'язок з фізичним пристроєм. Сервер працює як посередник між користувачем і модулем інтелекту, реалізуючи бізнес-логіку системи.

Висновок до розділу 1

У цьому розділі було проаналізовано сучасний стан сфери персоналізованого підбору харчових добавок, виявлено ключові проблеми — відсутність індивідуального підходу, перевантаження інформацією та нестачу інтегрованих інструментів підтримки користувача. Було обґрунтовано потребу в інтелектуальній системі, здатній поєднати штучний інтелект, вебтехнології та робототехнічні рішення задля створення персоналізованих рекомендацій на основі наукових джерел.

Описано мотивацію до розробки системи, що враховує сучасні виклики у сфері здоров'я, а також наведено аналіз предметної області, яка охоплює міждисциплінарний підхід: від NLP і UX/UI до IoT. Обґрунтовано вибір середовища розробки, інструментів та архітектури, що забезпечують автономність, масштабованість і ефективність взаємодії з користувачем.

Таким чином, створення подібної системи є своєчасним і актуальним кроком на перетині медичної науки, цифрових технологій та індивідуального турботливого підходу до здоров'я людини.

2 РОЗРОБКА ІНТЕЛЕКТУАЛЬНОГО МОДУЛЯ

2.1 Підхід до персоналізації в підборі харчових добавок

Персоналізація — це ключовий тренд сучасної медицини та здорового способу життя. Замість універсальних підходів дедалі більше уваги приділяється створенню індивідуальних стратегій для кожної людини, з урахуванням її унікальних фізіологічних особливостей, способу життя, цілей, віку, статі, та наявних симптомів. У сфері харчових добавок цей принцип стає особливо актуальним, оскільки ефективність впливу нутрієнтів значною мірою залежить від конкретного контексту організму.

У традиційних підходах до підбору сапліментів користувачі найчастіше орієнтуються на загальні рекомендації, поради знайомих, або ж маркетингові матеріали. Це створює високий ризик нерелевантного вибору, перевищення дозування або вживання непотрібних речовин. Такий підхід не лише неефективний, а іноді й шкідливий. Водночас, у науковій спільноті вже давно накопичено достатньо знань про те, як саме різні вітаміни, мінерали та біоактивні речовини впливають на організм у тих чи інших умовах.

У цьому дипломному проєкті персоналізація реалізована як багаторівневий процес, що починається з формування індивідуального профілю користувача. Користувач вводить базову інформацію про себе — вік, стать, ріст, вагу, а також відповідає на кілька запитань про самопочуття, наявні проблеми (наприклад, сонливість, дратівливість, втрата енергії) і цілі (покращити концентрацію, нормалізувати сон, знизити тривожність тощо). Ці дані дозволяють побудувати чіткий контекст, який служить основою для рекомендацій.

Особливу увагу приділено дизайну цього опитування: інтерфейс побудовано таким чином, щоб знизити бар'єр входу і зробити процес простим, інтуїтивним і навіть приємним для користувача. Кожна відповідь формує частину профілю, який надалі передається до модуля штучного інтелекту. Таким чином, персоналізація не зводиться до простого фільтрування, а включає семантичну інтерпретацію потреб користувача на основі його природної мови або вибору параметрів.

На основі зібраних даних генерується запит до великої мовної моделі, яка враховує зазначені індивідуальні фактори і формує список релевантних харчових добавок, кожна з яких супроводжується обґрунтуванням. У цьому полягає принципова відмінність від класичних рекомендаційних систем: модель не просто витягує інформацію зі статичних таблиць, а формує рекомендації на основі контекстного розуміння стану людини та знань, отриманих з наукових джерел [7].

Результатом такого підходу є поєднання високої інформативності та зручності: користувач отримує не просто список, а логічно структуровану відповідь, адаптовану до його потреб. Це підвищує довіру до системи, мотивацію дотримуватись рекомендацій і сприяє довгостроковому формуванню здорових звичок. Персоналізація, реалізована в системі, стає не лише інструментом покращення здоров'я, а й новим способом комунікації між людиною та наукою.

2.2 Збір і підготовка даних з PubMed

Для створення дійсно корисного та обґрунтованого інтелектуального модуля важливо, щоб усі рекомендації, які він генерує, базувалися не на загальних знаннях або припущеннях, а на науково підтверджених фактах. У рамках цього проєкту джерелом таких фактів стала одна з найбільших та найавторитетніших медичних баз у світі — PubMed [1]. Цей ресурс надає доступ до мільйонів рецензованих наукових публікацій, що охоплюють медицину, нутрицевтику, біологію, фармакологію та інші суміжні галузі.

Збір даних з PubMed розпочався зі створення списку найбільш поширених та актуальних харчових добавок, таких як магній, мелатонін, омега-3, вітамін D, залізо, цинк, ашваганда, бакопа, колаген тощо. Для кожного з цих елементів формувалися ключові запити англійською мовою, спрямовані на пошук статей, які містять опис впливу добавки на конкретний стан або симптом (наприклад, "magnesium AND sleep quality" або "ashwagandha AND stress").

Для автоматизації цього процесу використовувались або API PubMed, або власні скрипти з вебскрейпінгу, які дозволяли отримати такі метадані: назва статті, анотація (абстракт), дата публікації, DOI та посилання на повний текст, якщо він відкритий.

Зібрані дані проходили первинну перевірку та очищення — прибирались технічні символи, зайві HTML-теги, дублі, а також малозначущі статті без опису механізмів дії добавки. Після очищення проводився етап семантичної фільтрації: відбиралися лише ті статті, в яких чітко описано ефекти сапліменту, дослідження за участю людей (клінічні випробування) або інші переконливі емпіричні дані. Для цього використовувався метод ключових словосполучень і ручна валідація частини корпусу.

Також було введено вагову систему оцінки джерел, де перевага надавалась:

- 1) публікаціям не старше 10 років;
- 2) рецензованим статтям у журналах з імпакт-фактором;
- 3) дослідженням, що включають результати з вимірюваними показниками (рівень кортизолу, зміни сну, ЧСС тощо);
- 4) статтям з чітко сформульованими висновками.

У результаті сформовано корпус текстів, який виконує роль основного джерела знань для великої мовної моделі. Його призначення — надати моделі якісне, перевірене й актуальне контекстне підґрунтя для генерації медичних рекомендацій.

Корпус структуровано у форматі, зручному для індексації та обробки — у вигляді пар «запит — відповідь» або просто релевантних фрагментів, які можна швидко знаходити у векторній базі.

Таким чином, система, заснована на таких даних, має значну перевагу перед шаблонними або "вмонтованими" моделями. Вона працює не як чорна скринька, а як гібридна аналітична платформа, що поєднує потужність LLM з підтвердженою інформацією з реальних наукових публікацій [7].

2.3 Побудова локальної LLM-моделі для рекомендацій

Основою інтелектуального модуля системи виступає велика мовна модель (LLM — Large Language Model), спеціально адаптована для роботи з медичними запитам. На відміну від традиційних рекомендаційних алгоритмів, які оперують статичними правилами або обмеженими шаблонами, LLM-модель здатна контекстно інтерпретувати запит користувача, формулювати складні логічні зв'язки між симптомами й нутрієнтами, та генерувати персоналізовані відповіді, що базуються на наукових даних.

У межах цього проєкту було обрано модель medllama2:7b, яка демонструє високу якість обробки запитів у сфері охорони здоров'я. Модель розгорнуто локально за допомогою середовища Ollama Runtime [9], що дозволяє запускати її на персональному комп'ютері або сервері без потреби у постійному підключенні до зовнішніх API.

Такий підхід забезпечує декілька критичних переваг.

а) конфіденційність. Усі персональні дані користувача — такі як симптоми, вік, вага, індивідуальні цілі — обробляються локально і не передаються третім сторонам. Це особливо важливо в медичній сфері, де безпека інформації має вирішальне значення.

б) гнучкість конфігурації. Модель може бути донавчена або модифікована відповідно до специфіки задачі. Це дозволяє адаптувати відповіді під українськомовний інтерфейс, налаштувати стиль спілкування, зменшити ризик «галюцинацій» і краще враховувати векторну базу знань.

в) оперативність і автономність. Відсутність залежності від інтернету чи сторонніх API знижує час очікування відповіді (в середньому 15–30 секунд) і дозволяє використовувати систему навіть у закритих або офлайн-середовищах.

Технічно модель працює у зв'язці з попередньо зібраними даними з PubMed. Користувачський запит — сформований на основі його фізіологічних параметрів і вибору в опитуванні — перетворюється у prompt, що передається до LLM. Якщо використовується RAG [7]-підхід, то разом із запитом передається також

релевантний контекст — фрагменти з наукових статей. Модель на цій основі формує текст, який включає назви добавок, пояснення їхньої дії, показання до застосування і, за можливості, посилання на джерела.

Формат відповіді також стандартизовано. Кожен результат включає назву сапліменту, короткий опис механізму дії, а також кнопку для додавання до профілю. У випадку сумнівного або недостатнього запиту модель може повернути запитання для уточнення, або обережну відповідь із попередженням.

Важливо, що така модель є відкритою для подальшого розвитку: за бажанням її можна перенести на хмарну інфраструктуру, інтегрувати із системою медичних записів або розширити новими мовними доменами. Поточна реалізація вже демонструє практичну придатність, надійність і здатність до персоналізованої підтримки користувача в обґрунтованому виборі харчових добавок.

2.4 Інтеграція LLM-моделі з векторною базою знань (RAG-підхід)

Щоб забезпечити не лише гнучке генеративне мислення мовної моделі, але й фактологічну обґрунтованість відповідей, у проєкті реалізовано підхід Retrieval-Augmented Generation (RAG). Це сучасна архітектура, яка поєднує переваги великих мовних моделей із цілеспрямованим пошуком у базі знань, що значно підвищує точність і достовірність результатів. Особливо це актуально у сфері здоров'я, де недостовірна рекомендація може мати серйозні наслідки.

Суть RAG-підходу полягає в тому, що перед тим як LLM-модель сформує відповідь, система спочатку виконує пошук релевантної інформації в окремій векторній базі знань. У нашому випадку ця база складається з попередньо зібраних, очищених і структурованих фрагментів текстів з наукових статей PubMed. Кожен текстовий фрагмент перетворюється на вектор у багатовимірному просторі за допомогою медичних ембеддінгів, після чого зберігається у векторній базі, збудованій на основі FAISS або аналогічного механізму (Transformers [14]).

Коли користувач формує запит — через опитування або вручну — система створює векторне представлення цього запиту і виконує пошук найближчих

фрагментів у векторній базі. Зазвичай вибирається топ-3 або топ-5 найбільш релевантних результатів, які передаються до LLM як частина input prompt. Таким чином, відповідь генерується не "з нуля", а в контексті перевіреної наукової інформації, що суттєво знижує ймовірність "галюцинацій".

Цей підхід дозволяє не лише формувати рекомендації, а й аргументувати їх. У відповіді модель може посилається на реальні дослідження, наводити цитати або висновки зі статей. Це сприяє підвищенню довіри до системи, робить її експертною в очах користувача і відповідає вимогам прозорості в медичних інтерфейсах.

У системі також передбачено механізм оновлення бази знань: нові статті можуть регулярно додаватися, що дозволяє модулю залишатися актуальним без потреби повного перенавчання моделі. Завдяки цьому інтелектуальний агент адаптується до появи нових наукових відкриттів або змін у підходах до нутрицевтичного лікування.

2.5 Оцінка результатів рекомендацій

Після побудови інтелектуального модуля та інтеграції його з базою знань постає питання оцінки якості сформованих рекомендацій. Оскільки система працює в медично чутливому контексті, основна увага приділялася не лише технічній стабільності, а й точності, безпечності, пояснюваності та релевантності отриманих порад.

Оцінювання проводилось шляхом моделювання реальних сценаріїв використання. Було сформовано понад 30 тестових запитів, що охоплюють найпоширеніші запити користувачів: порушення сну, втома, зниження концентрації, підвищений рівень стресу, потреба в покращенні роботи шлунково-кишкового тракту тощо. Запити подавались у різних формах — як через формалізовану анкету, так і у вигляді відкритого запиту в стилі natural language input.

У відповідь система генерувала список сапліментів, кожен з яких супроводжувався описом дії, можливим механізмом впливу та — в окремих випадках — згадкою джерела, з якого було отримано інформацію. Усі рекомендації були проаналізовані з точки зору:

- чи є запропонована добавка актуальною для заявленого стану;
- чи не містить відповідь вигаданих або небезпечних речовин;
- наскільки зрозуміло подано пояснення дії добавки;
- чи відчутна користувачу логіка зв'язку між симптомом і порадою.

У переважній більшості випадків відповіді були логічними, обґрунтованими та зрозумілими. В окремих випадках система повертала нейтральну відповідь або просила уточнення, якщо інформації було недостатньо — що є ознакою коректного поводження моделі в умовах невизначеності. Також було протестовано поведінку системи при наданні суперечливих або неповних даних, і вона не видавала небезпечних або випадкових рекомендацій, а натомість формулювала загальні підходи до покращення стану або пропонувала звернутись до спеціаліста.

Додатково оцінювався користувацький досвід: інтеграція результатів у вебінтерфейс була інтуїтивною та зрозумілою, користувачі могли одразу додавати сапліменти до особистого списку, бачити підсвітку вже обраних елементів, а також формувати на їх основі розклад прийому. Усі ці деталі підвищують зручність використання і створюють повноцінну екосистему персонального оздоровлення.

Таким чином, за результатами тестування було зроблено висновок, що інтелектуальний модуль здатен ефективно формувати персоналізовані, релевантні та безпечні рекомендації. Його відповіді є логічними, прозорими для користувача і базуються на реальних даних. Це свідчить про успішність реалізованого підходу та придатність модуля до практичного застосування в реальних умовах.

Висновок до розділу 2

У другому розділі дипломного проєкту було детально розглянуто побудову інтелектуального модуля для персоналізованого підбору харчових добавок.

					ІК12.200БАК.006 ПЗ	Арк.
						21
Зм.	Лист	№ докум.	Підпис	Дата		

Ключовим елементом стала реалізація принципу персоналізації на основі індивідуальних фізіологічних параметрів, самопочуття та цілей користувача. На відміну від шаблонних підходів, система адаптується до конкретного контексту кожної людини, забезпечуючи високу релевантність рекомендацій.

Основу знань модуля становить корпус наукових статей з PubMed, ретельно відібраних та структурованих з урахуванням надійності, актуальності та доказової бази. Завдяки цьому система оперує не загальними фразами, а фактологічно обґрунтованими висновками, що значно підвищує довіру до неї.

Технічно модуль побудовано на базі великої мовної моделі medllama2:7b, розгорнутої локально за допомогою Ollama Runtime [9]. Її інтеграція з векторною базою знань реалізована за RAG-підходом, що поєднує генеративні можливості моделі з пошуком у перевірених джерелах. Це дозволяє уникати "галюцинацій" та підвищує точність відповідей. Оцінювання якості показало, що система демонструє логічність, обґрунтованість і безпечність рекомендацій, навіть у випадках часткової або неоднозначної інформації.

Таким чином, розроблений інтелектуальний модуль є надійним, адаптивним і готовим до практичного використання. Він не лише генерує персоналізовані рекомендації, а й обґрунтовує їх з позиції сучасної науки, створюючи інтелектуальний міст між користувачем та світовими медичними знаннями.

3 РОЗРОБКА ВЕБ-СИСТЕМИ ТА ІНТЕРАКТИВНОГО ІНТЕРФЕЙСУ

3.1 Архітектура фронтенду (Vue.js)

Для побудови користувацького інтерфейсу інтелектуальної системи було використано сучасний JavaScript-фреймворк Vue.js [2], який забезпечує високу продуктивність, реактивність даних та зручну модульну структуру. Основною перевагою Vue.js є можливість створювати масштабовані інтерфейси шляхом компонування незалежних функціональних блоків.

Архітектура фронтенду побудована за принципом Single Page Application (SPA). Це дозволяє завантажити інтерфейс один раз, а далі оновлювати тільки необхідні частини сторінки без повного перезавантаження. У результаті покращується продуктивність і зручність користування.

Система складається з таких основних компонентів:

1) Навігаційний хедер. Він включає логотип, посилання на ключові розділи сайту (рекомендації, мій профіль, підбір добавок, графік прийому), а також кнопки логіну та реєстрації або профілю та виходу — залежно від статусу користувача. Приклад зовнішнього вигляду наведено на рисунку 3.1 не авторизований користувач та рисунку 3.2 авторизований користувач.

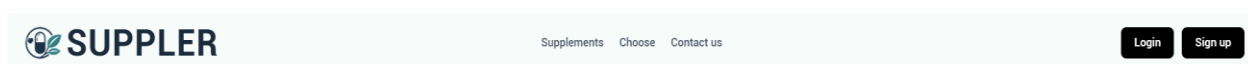


Рисунок 3.1 — Не авторизований користувач

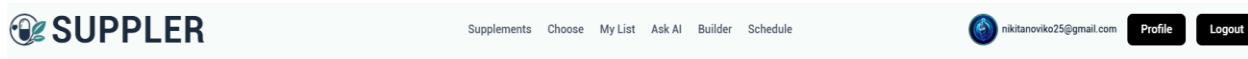


Рисунок 3.2 — Авторизований користувач

Головна сторінка. Вона вітає користувача, інформує про можливості системи та мотивує розпочати взаємодію з інтелектуальним помічником. Інтерфейс реалізовано у приємному, світлому стилі з мінімалістичним оформленням (рисунок 3.3 головна сторінка).

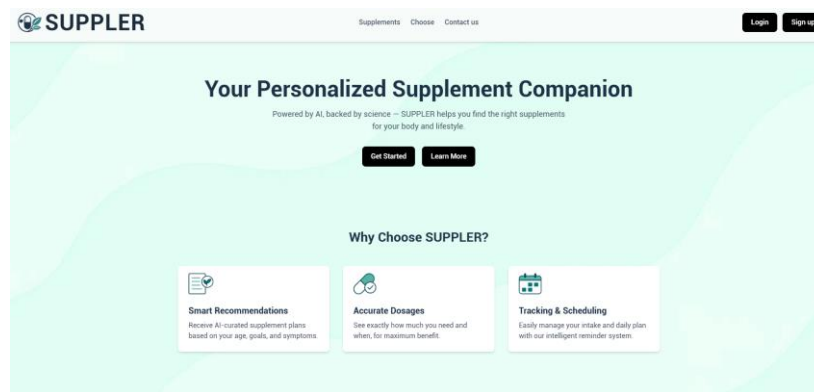


Рисунок 3.3 — Головна сторінка

Сторінка логіну та реєстрації. Реалізована за допомогою сервісу Auth0 [6], що дозволяє здійснювати автентифікацію без необхідності власної реалізації механізмів безпеки. Вона містить форми введення email/пароля або опцію входу через соціальні мережі

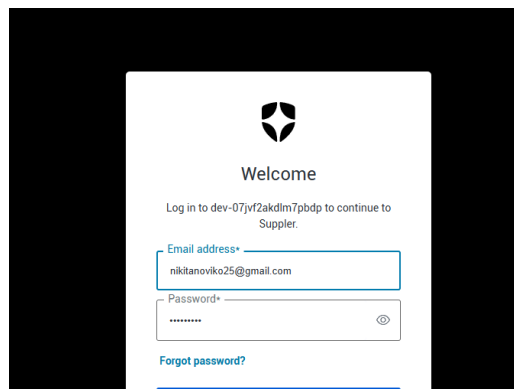


Рисунок 3.4 — Логін і Регистрація через Auth0

Сторінка профілю. Дає змогу редагувати базові персональні дані (вік, стать, вага), змінювати ім'я, а також завантажувати власний аватар. Усі зміни зберігаються до бази даних у реальному часі на рисунку 3.5 ми можемо бачити реалізацію профілю.

Рисунок 3.5 — Сторінка профілю

Інтерфейс чату з ІІІ. У вигляді текстового поля для запитів реалізацію діалогово вікна можна побачити на рисунку 3.6, нижче якого відображаються результати у форматі карток із назвою сапліменту, коротким описом та кнопкою "додати до мого списку".

Рисунок 3.6 — Інтерфейс чату з ІІІ

Розділ «Мої добавки». Це персональний перелік обраних користувачем сапліментів. Інтерфейс дозволяє переглядати опис кожного, редагувати або видаляти записи. Приклад демонстрації можна побачити на рисунку 3.7

Рисунок 3.7 — Розділ «Мої добавки»

Сторінка розкладу прийому. Представлена у вигляді інтерактивного календаря з розподілом на день і ніч для кожного дня тижня. Користувач може додавати сапліменти, які він приймає в конкретний день, і позначати їх виконання. Для того щоб відмітити що саплімент був прийнятий необхідно лише натиснути на відповідний , як показано на рисунку 3.8

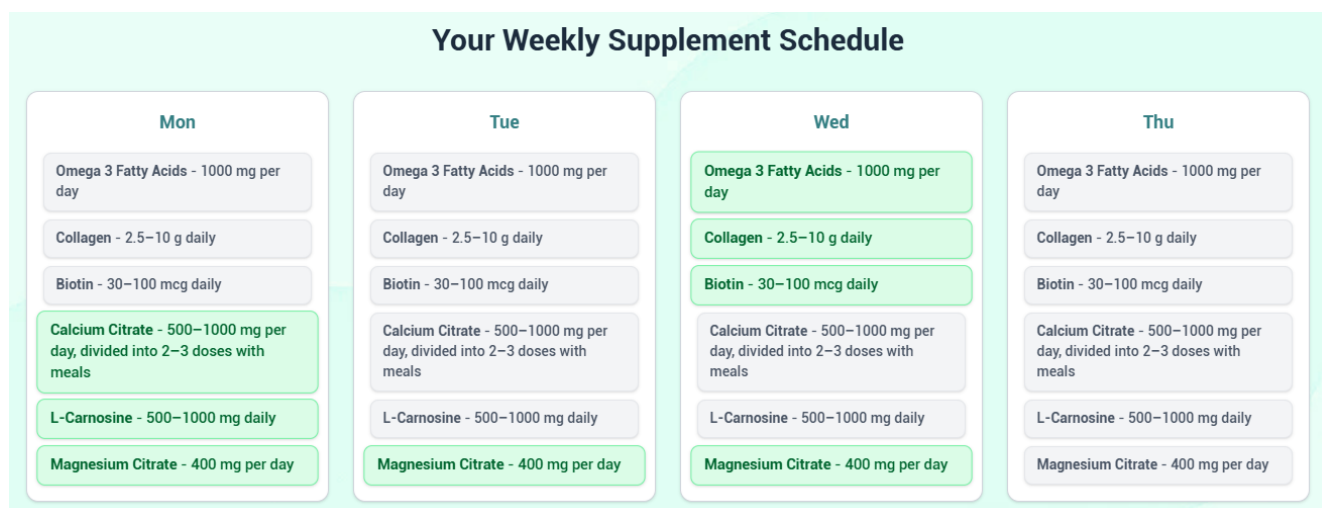


Рисунок 3.8 — Сторінка розкладу прийому

Для стилізації інтерфейсу було використано Tailwind CSS[12], що дозволило швидко реалізувати адаптивний дизайн. Також були застосовані transition-ефекти, які забезпечують плавні анімації при наведенні курсора або зміні станів елементів.

3.2 Структура бекенду (Node.js + MongoDB + JWT)

Серверна частина системи побудована на основі сучасного технологічного стеку, що поєднує продуктивність, масштабованість та безпеку. Для розробки обрано Node.js — асинхронне середовище виконання JavaScript, яке чудово підходить для побудови високонавантажених вебзастосунків [3]. У рамках проєкту реалізовано повноцінний сервер на базі Express.js, який обробляє всі запити, пов'язані з реєстрацією та автентифікацією користувачів, обробкою персональних даних, формуванням графіка прийому добавок, додаванням нових сапліментів, взаємодією з фізичним пристроєм і взаємозв'язком із фронтендом.

Усі користувацькі дані, включно з особистими профілями, списками добавок, конфігураціями комірок для підсвітки, графіками прийому та іншими параметрами, зберігаються у базі даних MongoDB — документно-орієнтованій СУБД, яка забезпечує високу гнучкість у збереженні вкладених структур [4]. Робота з базою організована через ORM Mongoose, яка дозволяє створювати гнучкі схеми, накладати обмеження, автоматично валідувати поля та спрощувати взаємодію з колекціями для візуалізації основних сценаріїв взаємодії користувачів із системою розроблено діаграму прецедентів ІК12.200БАК.006 Д1, яка демонструє ролі користувача, адміністратора та ШІ-модуля, а також відповідні дії, які кожен з них може виконувати в межах платформи..

У системі реалізовано механізм захищеної аутентифікації за допомогою JWT (JSON Web Token) [5]. Після входу користувач отримує пару токенів: accessToken (діє обмежений час) і refreshToken (для автоматичного оновлення сесії). Усі запити до захищених маршрутів перевіряються на наявність дійсного accessToken у заголовку Authorization. Сервер обробляє ці токени у спеціальному middleware, розшифровує їх за допомогою секретного ключа та надає доступ лише автентифікованим користувачам.

Взаємодія з фронтом реалізована у вигляді REST API [16]. Для кожного ресурсу створено відповідні ендпоінти: користувач може авторизуватись, отримати свої дані, оновити профіль, додати або видалити саплімент, зберегти розклад, змінити відповідність між добавками та LED-комірками, а також надіслати готовий графік на мікроконтролер. Уся взаємодія відбувається у форматі JSON, що забезпечує універсальність та простоту обробки.

Один із ключових елементів бекенду — це структура збереження персональних даних у базі. Кожен користувач має:

- унікальний email та зашифрований пароль;
- ім'я та аватар для профілю;
- масив supplementList з ObjectId обраних сапліментів;
- структуру schedule, яка включає день і ніч для кожного дня тижня;

- об'єкт `pillAssignments`, де кожна ячейка LED-пристрою пов'язана з конкретною добавкою;
- часові мітки створення й оновлення облікового запису.

Сервер також відповідає за зв'язок із фізичним пристроєм — мікроконтролером Wemos D1 Mini [11], який керує підсвіткою світлодіодів і відображенням розкладу на OLED-екрані. Запит до ESP8266 містить усю структуру прийому сапліментів по днях тижня, що дозволяє пристрою працювати автономно навіть після відключення від сервера. Передача даних здійснюється через спеціальний ендпоінт, який приймає повний JSON-графік та підтверджує збереження налаштувань у EEPROM пристрою.

Обробка помилок організована централізовано. Усі виняткові ситуації, включно з неправильними даними, недоступними маршрутами, помилками бази даних або спробами доступу без авторизації, відловлюються і перетворюються на чіткі повідомлення для фронтенду. Це забезпечує стабільну роботу системи та якісний зворотний зв'язок для користувача.

Таким чином, бекенд системи виступає потужним ядром усього застосунку, яке з'єднує інтерфейс, базу знань, мікроконтролер і персональні дані. Архітектура побудована так, щоб бути легко масштабованою, розширюваною і безпечною. У подальшому вона може доповнюватися зовнішніми API, аналітикою, мобільним застосунком або модулями телемедицини без суттєвих змін у її основі.

3.3 Авторизація та профіль користувача

У системі реалізовано повноцінну функціональність авторизації користувача на базі сервісу Auth0 [6], що дозволяє забезпечити безпечний, масштабований та стандартизований механізм автентифікації. Це є критично важливою частиною проекту, оскільки персоналізація рекомендацій, формування графіків, керування LED-пристроєм і збереження історії користувача напряду залежать від його авторизованого доступу.

Після переходу на сторінку входу користувач вводить електронну пошту та пароль. Усі дані надсилаються до сервісу Auth0, де обробляються відповідно до сучасних протоколів безпеки. У разі успішного входу система отримує два токени: `accessToken` для виконання запитів до захищених ресурсів та `refreshToken` для автоматичного подовження сесії без повторного введення облікових даних.

Ці токени зберігаються на стороні клієнта в безпечному сховищі браузера. Кожен запит до backend-серверу (реалізованого на Node.js [3]) містить `accessToken` у заголовку `Authorization`, що дає змогу системі перевірити автентичність користувача. При завершенні терміну дії токена frontend-частина автоматично викликає механізм оновлення на базі `refreshToken`.

Особистий профіль користувача містить основні параметри: ім'я, email, аватар (із можливістю редагування), список обраних сапліментів та індивідуальний розклад прийому. Усі дані зберігаються в базі MongoDB [4], а оновлення відбувається в режимі реального часу.

Особливу увагу приділено безпеці зберігання облікових даних: паролі користувачів на стороні Auth0 зберігаються у вигляді хешів, що відповідає сучасним стандартам, зокрема через використання алгоритму `bcrypt`. На сервері застосовано `middleware authMiddleware`, який перевіряє дійсність токенів перед доступом до будь-якого маршруту з персональними даними.

Після входу користувач автоматично перенаправляється на домашню сторінку або на сторінку редагування графіка, якщо вже існує налаштована коробка з комірками. При зміні профілю (наприклад, нового імені або зображення) оновлення даних застосовується в реальному часі.

Використання Auth0 як авторизаційної платформи спростило реалізацію безпечної автентифікації, дозволило уникнути ручної роботи з паролями, пришвидшило розгортання системи та забезпечило її відповідність промисловим стандартам безпеки. Це є важливою перевагою у дипломному проєкті, що демонструє вміння працювати з сучасними хмарними сервісами.

3.4 Інтерфейс підбору, додавання та перегляду добавок

Однією з ключових функцій інтелектуальної системи є модуль, що забезпечує взаємодію користувача з рекомендованими харчовими добавками. Саме через цей інтерфейс відбувається не лише ознайомлення з результатами, які сформував ІІІ-модуль, а й подальше керування списком сапліментів. Для реалізації даного розділу інтерфейсу використано фреймворк Vue.js [2], що дозволяє створювати гнучкі та масштабовані інтерфейси із сучасним UI-дизайном.

Інтерфейс побудований за принципами зручності та наочності. Після проходження опитування і входу в систему користувач бачить сформований для нього список рекомендованих добавок. Кожна добавка відображається у вигляді

окремого блока-картки з наступними елементами:

- 1) назва добавки — короткий заголовок;
- 2) опис дії — пояснення того, як саме діє добавка, з науковим підґрунтям (PubMed [1]);
- 3) показання до застосування — основні симптоми чи стани, при яких вона ефективна;
- 4) дозування — якщо доступно, виводиться рекомендована доза згідно з джерелами;
- 5) посилання на джерело — якщо присутнє, показується гіперпосилання на наукову статтю або опис дослідження.

У нижній частині кожного блоку знаходяться дві основні кнопки:

- 1) add — додає добавку до особистого списку (при натисканні колір елемента змінюється);
- 2) remove — видаляє раніше додану добавку з профілю користувача.

Щоб зробити взаємодію більш зрозумілою, додано візуальні ефекти: добавки, які вже було обрано, підсвічуються зеленим кольором (рис. 3.10), а неактивні залишаються сірими (рис. 3.9). Це дозволяє з першого погляду зорієнтуватися у статусі кожного елемента, навіть при великій кількості позицій.

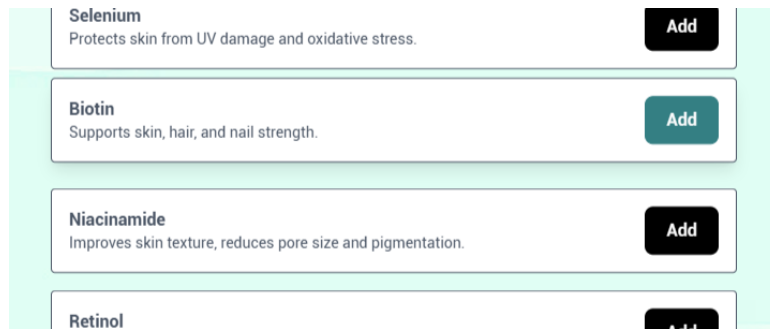


Рисунок 3.9 — Не доданий

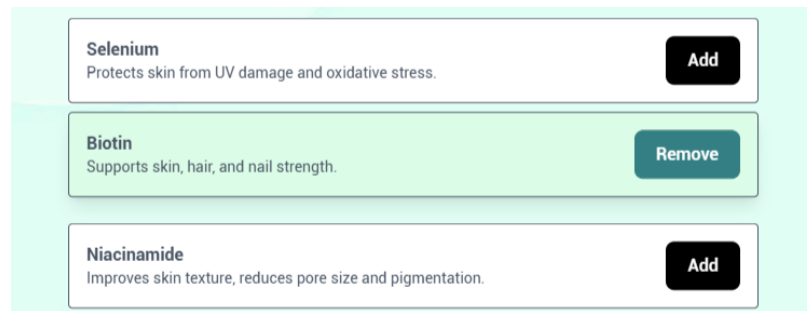


Рисунок 3.10 — Доданий

Крім базової взаємодії, передбачено модальне вікно з детальним описом кожного елементу. При натисканні на блок відкривається розширена картка, де подано додаткову інформацію: біохімічні властивості, взаємодія з іншими речовинами, дані з досліджень, ефективність для окремих вікових груп тощо. Такий підхід дозволяє не просто додавати елементи у список, а й приймати рішення на основі обґрунтованих знань — що особливо важливо в медичному контексті.

З технічного боку, взаємодія реалізована через REST API. Кнопки "Add" та "Remove" здійснюють запити POST і DELETE відповідно, а оновлений список одразу синхронізується з базою даних MongoDB як здійснюється передача можна відповідно побачити на Діаграма послідовностей ІК12.200БАК.006 Д2 . Уся передача даних здійснюється у форматі JSON, що дозволяє ефективно обробляти їх як на сервері, так і на клієнті.

3.5 Сторінка рекомендацій на основі AI

Центральним елементом користувацького інтерфейсу інтелектуальної системи є сторінка рекомендацій на основі AI, яка реалізує логіку генерації

індивідуальних порад щодо прийому харчових добавок. Вона служить основною точкою контакту між користувачем та штучним інтелектом і дає змогу отримати науково обґрунтовані пропозиції на основі особистих параметрів, зазначених у короткому опитуванні.

AI-модуль розгорнуто локально за допомогою платформи Ollama, яка забезпечує зручне розгортання великих мовних моделей у персональному середовищі. В основі модуля використовується модель medllama2:7b — це модифікація LLaMA2, натренована спеціально для роботи в медичному контексті [8]. Модель працює у форматі RAG (Retrieval-Augmented Generation) [10], що дозволяє поєднати генеративні можливості LLM із пошуком по зовнішній базі знань.

Для цього за допомогою бібліотеки LlamaIndex [9] було побудовано векторну базу знань, сформовану з наукових статей з PubMed. Документи обробляються, розбиваються на фрагменти, кожному з яких присвоюється ембедінг (векторне представлення). У момент запиту ці вектори використовуються для пошуку релевантної інформації до питання користувача, після чого результат додається у prompt для LLM.

Після введення тексту формується запит, який надсилається до локального AI-модуля. Обробка запиту здійснюється за допомогою мовної моделі medllama2:7b, розгорнутої на персональному комп'ютері в середовищі Ollama [7]. Модель використовує попередньо підготовлену базу знань, зібрану з PubMed, у рамках RAG-архітектури, що дозволяє включати до відповіді фактичні дані з наукових статей.

Кожна рекомендація виводиться у вигляді окремої інформаційної картки, яка містить:

- назву добавки;
- короткий опис її дії на організм;
- обґрунтування призначення, з можливим посиланням на відповідне дослідження;

— кнопку «Add», яка дозволяє миттєво додати добавку до профілю користувача.

Приклад візуального відображення результату який надає AI наведено нижче на рисунку 3.11

Ask the AI for supplement advice

Що приймати при тривожності та поганому сні?

Ask AI

Melatonin
This hormone helps regulate sleep-wake cycles and can improve sleep quality. **Add**

Ashwagandha Root
This adaptogen may help reduce stress and anxiety. **Add**

Magnesium Citrate
This mineral supports nerve function and may promote relaxation. **Remove**

Рисунок 3.11 — Приклад запиту

Коли користувач додає добавку, картка змінює свій вигляд — підсвічується зеленим кольором, що дозволяє швидко зорієнтуватися, які елементи вже обрано. Ця візуальна ідентифікація дозволяє уникнути дублювання, не потребуючи переходу на інші сторінки.

Обробка стану запиту, а також відображення результатів і повідомлень про помилки реалізовані через реактивну логіку, що забезпечує швидку та зрозумілу взаємодію з користувачем. Вся логіка побудована навколо підходу natural language interface — користувач може сформулювати запит у довільній формі (наприклад: "Відчуваю втому, хочу покращити концентрацію") — і отримати персоналізовану відповідь.

Важливо, що навіть при частково заповненому профілі (наприклад, лише вік і вага) система все одно здатна сформулювати відповідь. У випадках, коли AI-модуль тимчасово недоступний (через локальний перегрів, вимкнення сервера або

обмеження ресурсів), користувач бачить відповідне повідомлення про помилку з порадою повторити запит пізніше.

Формування запиту відбувається у форматі `prompt`, який об'єднує дані з опитування в один структурований текст. Цей `prompt` надсилається через HTTP-запит [17] на окремий backend-сервер, що працює на порту 5001 приклад запиту до локального AI-модуля можна побачити на рисунку 3.12. Відповідь повертається у вигляді об'єкта JSON, який перетворюється на серію візуальних карток безпосередньо у клієнті.

```
try {
  await fetchUserProfile()
  await fetchUserSupplements()
  const fullPrompt = `
  Age: ${userProfile.value.age || 'unknown'}
  Weight: ${userProfile.value.weight || 'unknown'}
  Height: ${userProfile.value.height || 'unknown'}
  Question: ${userMessage.value}
  const res = await fetch('http://localhost:5001/chat', {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify({ message: fullPrompt })
```

Рисунок 3.12 — Код запиту

Окрему увагу приділено масштабованості: поточна версія працює локально, але передбачено можливість переносу моделі на окремий сервер або у хмару, що дозволить обробляти запити швидше та підтримувати більшу кількість одночасних користувачів.

Крім того, реалізовано механізм накопичення прикладів запитів, що дозволяє донавчати модель у майбутньому для ще точніших рекомендацій. Це відкриває шлях до створення повноцінного інтелектуального агента, здатного адаптуватися до потреб користувача в динаміці.

Загалом, сторінка рекомендацій виступає не лише результатом взаємодії з AI, а й платформою для глибокої, науково обґрунтованої персоналізації. Вона формує міст між користувачем і знанням, роблячи процес вибору добавок прозорим, обґрунтованим та зручним.

					ІК12.200БАК.006 ПЗ	Арк.
						34
Зм.	Лист	№ докум.	Підпис	Дата		

3.6 Сторінка створення персональних рекомендацій

Окремий розділ вебінтерфейсу присвячений швидкому формуванню персоналізованих рекомендацій без потреби ручного введення запитів. Ця сторінка побудована у вигляді інтерактивного анкетування, яке дозволяє користувачеві вибрати найбільш релевантні варіанти стану здоров'я, наявні проблеми та очікувані результати — наприклад, покращення сну, зниження тривожності або підтримка суглобів.

Анкета містить також базові фізіологічні параметри (вік, вага, зріст), що дозволяє штучному інтелекту краще адаптувати відповіді. Структура блоків побудована таким чином, щоб користувач не відволікався на навігацію, а зосередився виключно на виборі варіантів.

Після натискання кнопки "Get My Recommendations" дані автоматично перетворюються у структурований запит, який надсилається до локального AI-модуля. Система, використовуючи модель medllama2:7b, аналізує вхідні параметри й генерує список добавок, релевантних саме для цього профілю.

Як і у минулих розділах кожна картка оснащена кнопкою "Add", що дозволяє додати добавку до особистого списку користувача без потреби оновлення сторінки. Якщо добавка вже була збережена, її фон автоматично змінюється на зелений, що запобігає дублюванню та надає візуальний зворотний зв'язок приклад запиту можна побачити на рисунку 3.13.

Let's get to know you

How do you feel lately?

Tired Energetic Stressed Relaxed Unfocused

Any issues you'd like to address?

Sleep problems Low energy Anxiety Digestive issues

Your health goals

Better focus Skin health Immune boost Stress support

Age 21 Weight (kg) 70 Height (cm) 180

Get My Recommendations

Рисунок 3.13 — Візуальний приклад результату після опитуванн

Уся логіка взаємодії інкапсульована в одному компоненті Vue, що робить сторінку масштабованою і придатною до повторного використання в інших сценаріях — наприклад, для дітей, спортсменів або літніх людей.

Таким чином, сторінка «Build Your Own Recommendations» виконує функцію швидкого старту для нових користувачів, спрощує доступ до рекомендацій, і водночас демонструє персоналізовану цінність системи вже на етапі першої взаємодії. Це підвищує залучення, довіру та бажання користуватись продуктом у довгостроковій перспективі.

3.7 Побудова розкладу прийому добавок

Ефективність харчових добавок значною мірою залежить не лише від їх складу, але й від регулярності та часу прийому. Тому важливим компонентом інтелектуальної системи є можливість формування персонального графіка прийому саплементів, адаптованого до потреб кожного користувача.

На реалізованій сторінці «Weekly Intake Schedule» користувач бачить візуально структуровану сітку з 7 днів тижня, кожен з яких містить список саплементів, доданих до його профілю.

Кожна клітинка розкладу є інтерактивною. При натисканні на назву саплементу користувач може позначити або скасувати факт прийому (відмічається зміною кольору). Ця інформація автоматично зберігається в базі даних через API, створюючи індивідуальний журнал прийому. Завдяки цьому реалізується механізм тривалого моніторингу дотримання режиму, що у майбутньому може бути використано для аналізу ефективності тих чи інших добавок.

3.8 База даних

Для зберігання всієї інформації, що використовується в системі, застосовується документно-орієнтована база даних MongoDB [4], розгорнута локально або у хмарі. Вона організована у вигляді колекцій, кожна з яких

					ІК12.200БАК.006 ПЗ	Арк.
						36
Зм.	Лист	№ докум.	Підпис	Дата		

відповідає окремому типу сутностей. Обрана структура дозволяє зберігати вкладені об'єкти, гнучко масштабуватись та легко обробляти персоналізовані дані без складного реляційного моделювання. База даних має назву `supplerdb` і включає наступні основні колекції:

1) `users` — містить профілі користувачів: ім'я, email, аватар, хешований пароль, токени доступу, список обраних добавок, графік прийому, відповідність добавок до LED-комірок та іншу персональну інформацію. Дані захищено відповідно до вимог приватності.

2) `supplements` — глобальна колекція доступних харчових добавок. Тут зберігаються назва, опис, категорія дії, рекомендоване дозування, ідентифікатор і статус активності. Всі елементи доступні для вибору в рекомендаціях та вручну через інтерфейс користувача або адміністратора.

3) `pillboxes` — відображає налаштування фізичної коробки користувача. Зберігає відповідність між сапліментами (за `ObjectId`) і комітками (за номером), що дозволяє коректно підсвічувати LED на мікроконтролері відповідно до обраного розкладу. Ця колекція фактично пов'язує цифрову частину з фізичним пристроєм.

4) `intakelogs` — колекція, що фіксує історію прийому добавок. Для кожного користувача зберігаються записи про те, коли і яку добавку було запропоновано до прийому, які LED-комірки були активовані, а також чи було підтверджено прийом. Це дозволяє реалізувати трекінг та оцінювати дотримання режиму в перспективі.

Усі колекції мають індекси для оптимізації запитів, а також перевірку у форматі `Mongoose-схем` для валідації вхідних даних. Така структура дозволяє легко масштабувати систему, додавати нові функції (наприклад, відстеження стану здоров'я, журнал змін, збереження чат-запитів). Більш зрозумілу архітектуру можна побачити на ER Діаграмі ІК12.200БАК.006 ДЗ

Завдяки використанню `MongoDB`, проєкт отримує високу швидкість роботи з даними, стабільність, а також гнучкість у персоналізації — кожен користувач має незалежну конфігурацію, яку можна зберігати, оновлювати або переносити між пристроями.

3.9 Механізм збереження даних користувача

Одним із ключових компонентів вебсистеми є реалізація стабільного та безпечного механізму збереження даних користувача. Оскільки платформа забезпечує персоналізований підбір харчових добавок, формування індивідуального графіка прийому та управління профілем, важливою вимогою є збереження всіх змін і станів у базі даних без втрат або помилок.

Для реалізації системи збереження використано базу даних MongoDB, яка завдяки своїй гнучкій документо-орієнтованій структурі дозволяє зберігати об'єкти у вигляді JSON-документів. Кожен користувач має власний документ у колекції `users`, де фіксується унікальний ідентифікатор, ім'я, електронна пошта, список обраних добавок, графік прийому (`schedule`), а також дані профілю — вік, зріст, вага, цілі.

На серверній частині реалізовано RESTful API з такими маршрутами, як:

- GET `/user/me` — отримання поточного користувача,
- PATCH `/user/update` — оновлення профілю,
- POST `/user/pillbox` — збереження нової конфігурації добавок,
- POST `/user/schedule` — збереження або оновлення графіка прийому.

Кожен запит до API супроводжується JWT-токеном авторизації, який перевіряється на сервері, що гарантує доступ лише до власних даних користувача. На фронтенді (Vue.js) реалізовано взаємодію з бекендом за допомогою `fetch` або `axios`, з обробкою помилок і повідомленнями про успішне збереження приклад запиту для оновлення аватару можна побачити на рисунку 3.14.

Наприклад, при створенні нового розкладу прийому добавок, дані, обрані користувачем у графічному інтерфейсі, перетворюються на масив об'єктів, де кожен день тижня має окремо масив `day_leds` та `night_leds`. Цей масив надсилається на сервер у вигляді JSON, де зберігається у відповідному полі бази даних, пов'язаному з ID користувача.

```

router.put('/profile', authMiddleware, upload.single('avatar'), async (req, res)
=> {
  const auth0Id = req.auth.sub
  const { age, weight, gender } = req.body
  const update = { age, weight, gender }

  if (req.file) {
    update.picture = `uploads/${req.file.filename}`
  }

  const user = await User.findOneAndUpdate({ auth0Id }, update, { new: true })
  res.json({ message: 'Profile updated', user })
})

```

Рисунок 3.14 — Запит на оновлення аватару

Збереження реалізовано з урахуванням перевірки правильності структури даних (валідація) та повторної відправки при виникненні помилки.

Таким чином, користувач може бути впевнений, що його профіль, налаштування й обрані сапліменти залишаються актуальними навіть після перезапуску сторінки чи повторного входу в систему.

У майбутньому передбачається додаткове розширення механізмів збереження: збереження історії змін, логів дій, аналітики запитів до AI та резервного копіювання даних у хмарне сховище для підвищення надійності. Такий підхід забезпечує масштабованість, безпечність і гнучкість роботи з персональними даними у межах інтелектуальної системи.

Висновок до розділу 3

У третьому розділі дипломного проєкту було розглянуто архітектуру та реалізацію вебсистеми для персоналізованого підбору харчових добавок. Сформовано повноцінний функціональний фронтенд на базі Vue.js, що забезпечує сучасний, інтуїтивно зрозумілий інтерфейс користувача з можливістю інтерактивного опитування, перегляду рекомендацій на основі AI, керування особистим профілем та графіком прийому добавок.

Серверна частина реалізована на Node.js з використанням Express.js, MongoDB та JWT-автентифікації, що забезпечило надійність, масштабованість та захист персональних даних. Інтеграція з Auth0 дозволила підвищити рівень безпеки авторизації та пришвидшити розгортання системи. Значну увагу приділено збереженню даних користувачів, а також взаємодії з мікроконтролером Wemos через REST API для управління фізичним пристроєм.

Кожен елемент системи — від підбору добавок до відображення прийому по днях — розроблено з урахуванням зручності, адаптивності та наукової обґрунтованості. Інтерфейс взаємодії з AI-модулем для генерації рекомендацій на основі даних користувача, зберігає результати у базі, дозволяє створювати графіки та контролювати прийом у режимі реального часу.

Таким чином, розділ 3 підтверджує технічну реалізованість і комплексний характер запропонованої системи, що поєднує сучасні вебтехнології, штучний інтелект та інтеграцію з фізичним пристроєм у єдину інтелектуальну екосистему підтримки здоров'я.

4. ІНТЕГРАЦІЯ З РОБОТОТЕХНІЧНИМ МОДУЛЕМ

4.1 Обґрунтування використання фізичного пристрою

Використання фізичного пристрою у структурі інтелектуальної системи персоналізованого підбору харчових добавок має як функціональне, так і концептуальне обґрунтування. Ідея полягає не лише в тому, щоб сформулювати рекомендації у цифровому вигляді, а й у тому, щоб перенести результат взаємодії з системою у фізичне середовище, де користувач отримає нагадування або інструкцію щодо прийому добавок у зручній і зрозумілій формі. Це дозволяє створити повноцінний замкнутий цикл — від аналізу стану до конкретної дії в реальному світі.

Однією з ключових проблем при вживанні харчових добавок є регулярність і дисципліна. Навіть за наявності правильно підбраного плану багато користувачів забувають приймати препарати вчасно, плутають ранкові та вечірні дози, або втрачають фокус через однотипність упаковок. Саме тому в системі реалізовано фізичний модуль — пристрій, який сигналізує про час прийому конкретних добавок шляхом підсвічування відповідних комірок LED-індикаторами [14].

На відміну від класичних органайзерів, де користувач повинен самостійно пам'ятати графік, заповнювати комірки вручну і звірятись із інструкцією, розумна коробка автоматизує ці процеси. Вона отримує графік із вебінтерфейсу у вигляді структурованого JSON-файлу, зберігає його у внутрішній EEPROM-пам'яті мікроконтролера [13] і виконує підсвічування відповідно до поточного дня та режиму (день/ніч). Користувачу достатньо лише подивитися, яка комірка світиться, і прийняти добавку без зайвих дій.

Крім функціонального навантаження, пристрій виконує й мотиваційну роль: він формує звичку дотримуватись режиму прийому, візуалізує користь від проходження циклу, а також створює відчуття залученості. Це особливо актуально для людей з високим рівнем навантаження, літніх осіб, людей з когнітивними порушеннями або для тих, хто просто прагне мінімізувати ризик забути прийом.

Таким чином, пристрій виступає як елемент підвищення якості медичної дисципліни.

Технічно пристрій реалізований на базі мікроконтролера ESP8266 (модель Wemos D1 mini), який підтримує з'єднання з Wi-Fi мережею [11] та може приймати команди через HTTP-запити. Він керує 9 адресними світлодіодами WS2812 [14], OLED-дисплеєм для виводу текстової інформації [15] та двома кнопками, які дозволяють вручну перемикає день тижня або режим прийому. Така конфігурація дозволяє пристрою працювати як у напівавтономному, так і у повністю автономному режимі, навіть без активного підключення до вебінтерфейсу. Для більшого розуміння була створена структурна схема ІК12.200БАК.006 Д4.

На відміну від систем, які працюють лише в цифровому середовищі, запропонований підхід передбачає реальну фізичну взаємодію. Це суттєво підвищує ефективність використання, дозволяє адаптувати систему до побутових умов, а також забезпечує унікальність рішення в контексті персонального здоров'я. Інтеграція фізичного елемента — не просто зручність, а ключова відмінність проєкту від існуючих аналогів.

4.2 Алгоритм управління робототехнічним модулем

У системі реалізовано повноцінний алгоритм керування робототехнічним модулем, що виконує функцію інтелектуального нагадування про прийом харчових добавок відповідно до персоналізованого графіка. Модуль працює на базі мікроконтролера Wemos D1 mini з мікропроцесором ESP8266, який має вбудовану підтримку Wi-Fi і достатній обсяг пам'яті для локального збереження інформації. Пристрій керує світлодіодами WS2812 (індивідуальна адресація) [14], OLED-дисплеєм для виводу поточного статусу, а також двома фізичними кнопками для зміни режиму та вибору дня тижня.

Управління модулем організовано на основі попередньо сформованого графіка, який надсилається з клієнтської частини системи у вигляді JSON-структури. Ця структура включає масив із семи днів тижня, кожен з яких має два

масиви — day та night, що містять індекси ячеек, які мають бути підсвічені у відповідному режимі. Наприклад, об'єкт `days[0].day = [2, 4]` означає, що у понеділок вдень повинні загорітись ячейки під номерами 2 та 4.

Після отримання графіка через POST-запит за допомогою Wi-Fi, мікроконтролер зберігає цю інформацію у власну EEPROM-пам'ять [13]. Це дозволяє зберегти дані навіть після вимкнення живлення або перезавантаження пристрою. При ввімкненні модуль завантажує збережені дані та активує відповідні світлодіоди згідно з поточним днем і режимом. Користувач може перемикає дні кнопкою "Day", а режим день/ніч — кнопкою "Mode". Відповідний текст (наприклад, "Monday — Day") виводиться на OLED-дисплей.

У кожному циклі оновлення програма перевіряє:

- 1) номер активного дня (0–6),
- 2) активний режим (`isDay = true/false`),
- 3) чи існують у цьому дні відповідні комірки,
- 4) які саме LED потрібно підсвітити, і які залишити вимкненими.

Світлодіоди підсвічуються індивідуально відповідно до комірки, якій було призначено конкретну добавку. Завдяки адресності WS2812, кожна LED-ячейка керується окремо. У разі відсутності даних на поточний день режим виводу вимикає всі LED, тим самим унеможливорюючи хибну активацію.

OLED-дисплей додатково використовується для комунікації з користувачем. Він показує день тижня, активний режим і коротке повідомлення (наприклад, "Night: take 3 supplements"), що робить взаємодію інтуїтивною і доступною навіть без застосунку [15].

Крім базової логіки, у коді передбачено захист від помилок: некоректні JSON-запити ігноруються, парсинг перевіряється на валідність, а структура графіка тестується перед записом у пам'ять. Це забезпечує стійку роботу навіть у випадку непередбачених збоїв у мережі чи при нестабільному з'єднанні з клієнтом.

На стороні таблеток реалізовано функцію, яка дозволяє одним натисканням кнопки сформувати й надіслати повний графік прийому харчових добавок на

пристрій. В основі — асинхронна функція `sendFullSchedule()`, яку можна побачити на рисунку 4.1.

```
async function sendFullSchedule() {
  try {
    const token = await getAccessTokenSilently()

    const userRes = await fetch('http://localhost:3000/user/me', {
      headers: { Authorization: `Bearer ${token}` }
    })
    const user = await userRes.json()
    const userId = user._id

    const scheduleRes = await fetch(`http://localhost:3000/user/pillbox/schedule-
public?userId=${userId}`)
    if (!scheduleRes.ok) throw new Error('✗ Failed to fetch schedule from DB')
    const scheduleData = await scheduleRes.json()

    const wemosRes = await fetch('http://192.168.1.38/schedule', {
      method: 'POST',
      headers: { 'Content-Type': 'application/json' },
      body: JSON.stringify(scheduleData)
    })

    if (!wemosRes.ok) throw new Error('✗ Failed to send to Wemos')
    const result = await wemosRes.json()

    console.log('☑ Sent to Wemos:', result)
    alert('☑ Schedule sent to Wemos!')

  } catch (err) {
    console.error('✗ Error sending schedule:', err)
    alert('✗ Failed to send schedule')
  }
}
```

Рисунок 4.1 — Відправка розкладу

У першій частині функції здійснюється отримання `accessToken` для автентифікації користувача, після чого робиться запит до маршруту `/user/me`, який повертає ідентифікатор користувача. Далі цей ідентифікатор використовується для запиту до серверного маршруту `/user/pillbox/schedule-public`, який повертає повну структуру графіка прийому — у форматі, що враховує розклад по днях тижня,

розділення на денний і нічний прийом, і відповідні комірки, які мають підсвічуватись.

Отриманий об'єкт `scheduleData` передається у форматі JSON на IP-адресу пристрою Wemos (у даному прикладі — 192.168.1.38) за маршрутом `/schedule`. Wemos приймає цей POST-запит, розпаковує структуру і зберігає її у пам'яті для подальшого використання. Таким чином, Wemos отримує повну, готову до виконання інструкцію, що саме й коли підсвічувати, а також на які LED-комірки впливати — без складної логіки на стороні пристрою .

4.3 Підключення та конфігурація елементів пристрою

Фізичний пристрій створено на базі мікроконтролера ESP8266 (Wemos D1 Mini), до якого підключено OLED-дисплей, 9 окремих світлодіодів WS2812, дві кнопки керування та USB-живлення. Уся конструкція інтегрується з вебдодатком через Wi-Fi.

OLED-дисплей з роздільною здатністю 128×64 працює по інтерфейсу I2C, займає порти D1 (SCL) та D2 (SDA) та відображає день тижня і активний режим прийому — день або ніч. Це дозволяє користувачу швидко орієнтуватися в поточному статусі без потреби звертатись до телефону чи комп'ютера.

Основним елементом індикації є дев'ять індивідуальних RGB-світлодіодів типу WS2812, які з'єднані в один цифровий ланцюг і підключені до піну D4. Кожен світлодіод відповідає певній комірці в коробці для добавок, і підсвічується згідно з графіком, збереженим у пам'яті пристрою. Розклад завантажується через JSON-запит із вебінтерфейсу та зберігається в EEPROM, що гарантує його збереження після вимкнення живлення.

Дві фізичні кнопки, розміщені на пін D5 та D6, дозволяють вручну перемикати день тижня та режим (день/ніч). Це забезпечує зручне керування без доступу до Wi-Fi чи смартфона. Реакція на натискання реалізована з урахуванням `debounce-фільтрації`.

Живлення пристрою зазвичай подається через USB (наприклад, від адаптера або PowerBank), однак його архітектура дозволяє організувати повністю автономну роботу від акумулятора. Зокрема, підтримується підключення літій-полімерної батареї 3.7V з підзарядкою через вбудований micro-USB, а також можливість підключення модуля живлення TP4056 із захистом. За умови оптимізації яскравості LED і контролю споживання енергії, пристрій здатен працювати без підзарядки протягом кількох діб.

Електрична схема яку можна побачити на рисунку 4.2 включає захисний резистор 330 Ом на лінії даних між D4 і першим WS2812, а також конденсатор на 1000 мкФ для стабілізації напруги. Усі компоненти працюють від 5V, а логіка керування забезпечує безпечну і стабільну роботу в реальних умовах.

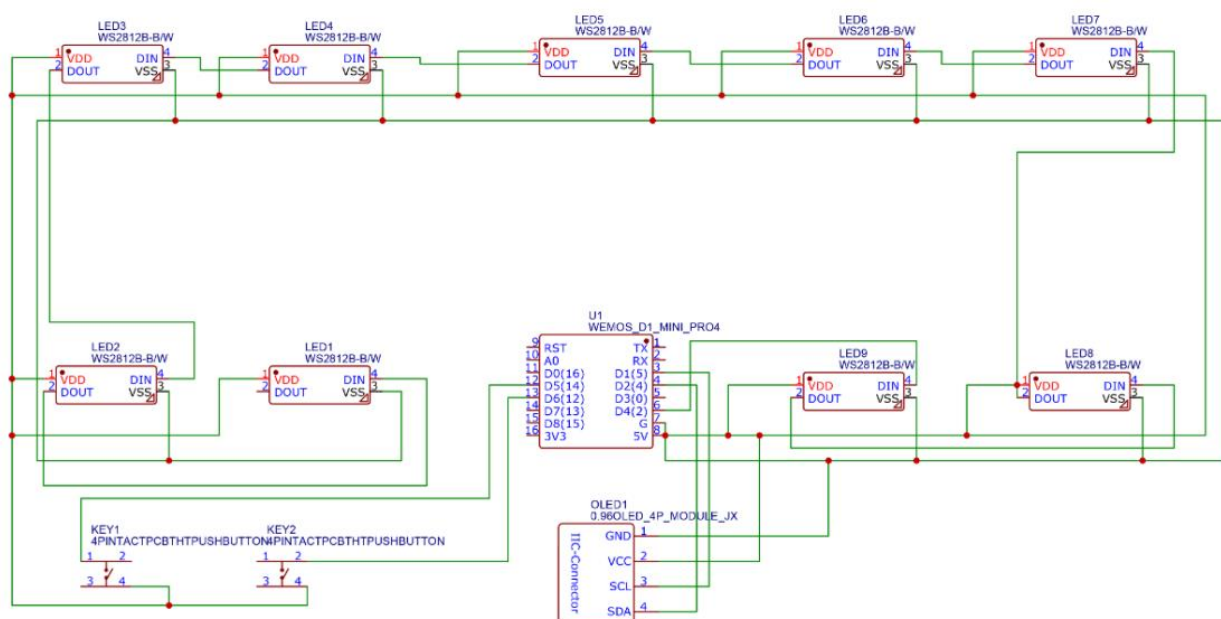


Рисунок 4.2 — Електрична схема

Конфігурація обрана з урахуванням компактності, надійності та можливості масштабування: у майбутньому легко додати більше світлодіодів, вбудовану RTC-синхронізацію або сенсори, не змінюючи базову логіку підключення.

4.4 Реалізація фізичного корпусу пристрою на основі 3D-моделі

Для забезпечення зручного розміщення всіх апаратних компонентів та створення цілісного готового пристрою було розроблено та реалізовано фізичний корпус на основі 3D-моделі, призначений спеціально для роботи системи персоналізованого підбору харчових добавок. Корпус спроектовано індивідуально, із урахуванням усіх вимог до ергономіки, розміщення світлодіодів, екрана, кнопок і комірок під капсули. На кресниках ІК12.200БАК.006 С1 зображено загальний вигляд 3D-моделі корпусу.

Модель створена у середовищі тривимірного моделювання Fusion 360, після чого роздрукована на FDM-принтері з PLA-пластику. Корпус складається з одного цілісного блоку з 9 чітко відокремленими комірками, розміщеними у два ряди: верхній і нижній. Кожна комірка відповідає одному з 9 окремих WS2812-світлодіодів, що встановлюються знизу, підсвічуючи відповідні відсіки згідно з активним графіком (рисунок 4.3).



Рисунок 4.3 — Корпус прототипу

У нижній частині корпусу вбудовано кольорову вставку з PLA жовтого кольору з отворами під OLED-дисплей і дві кнопки керування. Це дозволило функціонально відокремити зону взаємодії користувача й зробити пристрій більш

візуально структурованим (рисунок 4.4). В отвори легко вставляються готові компоненти без додаткової механічної обробки після друку.



Рисунок 4.4 — Закрита кришка

Бічні стінки корпусу обладнані замковими засувками, які дозволяють зафіксувати верхню кришку або кришку з електронною платою. Це забезпечує зручність у зборці, технічному обслуговуванні та надійне утримання конструкції без використання гвинтів (рисунок 4.5).

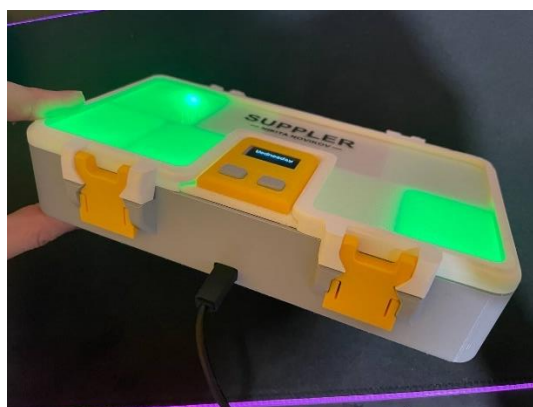


Рисунок 4.5 — Конструкція замків

Конструкція передбачає місце для розміщення плати ESP8266 (Wemos D1 Mini), підключення micro-USB кабелю для живлення та, за потреби, літій-полімерного акумулятора. Така архітектура дозволяє перейти на повністю автономний режим роботи без постійного підключення до джерела енергії.

Приклад розміщення електроніки всередині корпусу показано на рисунку 4.6. також на рисунку 4.7 можна побачити окремо розібрану коробку

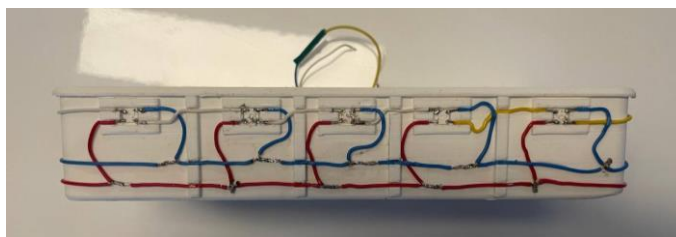


Рисунок 4.6 — Підключення світодіодів



Рисунок 4.7 — Коробка окремо

Усі запити до коробки виконані через веб інтерфейс , як на рисунку 4.8 ми можемо відправляти готовий графік до ESP8266

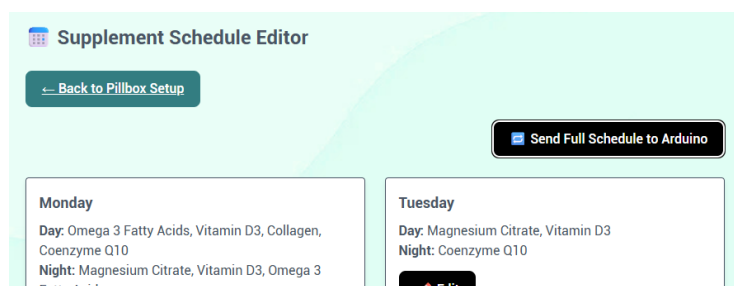


Рисунок 4.8 — Відправка даних розкладу

Реальна модель була роздрукована, зібрана, протестована та використовується у демонстраційному прототипі. Такий корпус не лише

забезпечує зручне розміщення компонентів, а й формує завершений вигляд пристрою, готового до повсякденного використання.

Висновок до розділу 4

У цьому розділі було детально розглянуто фізичну та програмну реалізацію робототехнічного модуля, що виконує функції інтелектуального нагадування про прийом харчових добавок. Запропоноване рішення поєднує можливості IoT-пристрою на базі ESP8266 з вебінтерфейсом, що дозволяє зручно та ефективно управляти персоналізованим графіком прийому сапліментів.

Розроблена апаратна частина забезпечує інтуїтивно зрозумілий механізм взаємодії з користувачем — завдяки OLED-дисплею, кнопкам перемикання та LED-індикації, користувач отримує чіткі й візуальні сигнали щодо необхідності прийому добавок у певний час. Пристрій має здатність працювати в автономному режимі, зберігати графік у вбудованій пам'яті та реагувати на зміну днів і режимів без постійного з'єднання з сервером.

Крім того, було впроваджено повний цикл обміну даними між клієнтською частиною системи та пристроєм через Wi-Fi, з реалізацією надійного механізму передачі графіка у форматі JSON. Було розроблено й виготовлено індивідуальний 3D-корпус, що об'єднав усі компоненти в зручну конструкцію, готову до реального використання.

Таким чином, модуль є не лише допоміжним елементом системи, а її ключовою частиною, яка переносить результати роботи штучного інтелекту в фізичну площину, забезпечуючи користувача максимально зручним, наочним і технологічно просунутим інструментом для підтримки здоров'я.

5. ЕКСПЕРИМЕНТАЛЬНА ОЦІНКА ТА ТЕСТУВАННЯ

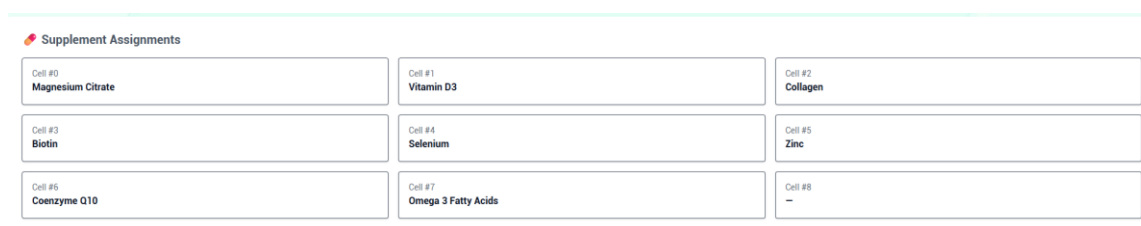
5.1 Комплексне тестування функціональності системи

Після завершення розробки інтелектуального модуля, вебінтерфейсу та фізичного пристрою було проведено повноцінне тестування всієї системи з метою перевірки її стабільності, зручності використання та відповідності поставленим функціональним вимогам. Тестування охоплювало як програмну, так і апаратну частину рішення в умовах максимально наближених до реального використання.

Перш за все було перевірено інтелектуальний модуль, який відповідає за персоналізовані рекомендації. Користувач вводив дані про себе: вік, стать, поточне самопочуття, проблеми зі сном, стрес або концентрацію. Модель LLM, донавчена на текстах із PubMed, формувала змістовні та логічні відповіді, пропонуючи актуальні харчові добавки згідно з науковими джерелами.

Рекомендації були осмисленими, релевантними до вхідних параметрів і доповнювались коротким поясненням впливу кожного сапліменту. Це підтвердило ефективність використання LLM у форматі локального агента рекомендацій.

Також було протестовано вебінтерфейс користувача. Як показано на рисунку 5.1, у профілі можна переглянути збережений список добавок із відповідними комітками.



Supplement Assignments		
Cell #0 Magnesium Citrate	Cell #1 Vitamin D3	Cell #2 Collagen
Cell #3 Biotin	Cell #4 Selenium	Cell #5 Zinc
Cell #6 Coenzyme Q10	Cell #7 Omega 3 Fatty Acids	Cell #8 —

Рисунок 5.1 — Перегляд збереженої таблиці.

Функціонал додавання добавок, збереження графіка та інтеграція з пристроєм працював стабільно. На рисунку 5.2 наведено приклад відправки запиту на мікроконтролер через кнопку "Send Full Schedule".

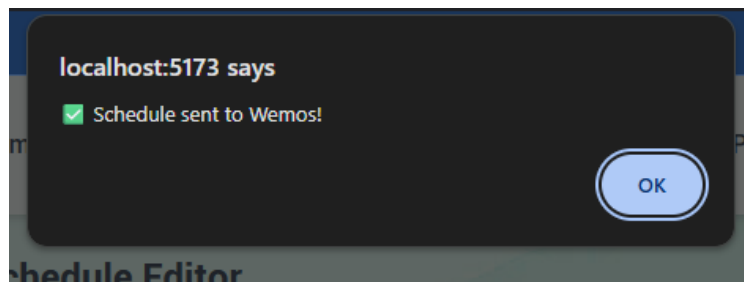


Рисунок 5.2 — Відправка запиту на мікроконтролер.

Фізичне тестування пристрою підтвердило його повну функціональність. На рисунку 5.3 зображено процес запуску пристрою: під час ініціалізації світлодіоди світяться жовтим, а після підключення до Wi-Fi — змінюють колір на зелений.



Рисунок 5.3 — Запуск пристрою.

Далі, як показано на рисунку 5.4, OLED-дисплей відображає поточний день і активний режим (день/ніч), що дозволяє швидко зорієнтуватися.



Рисунок 5.4 — Вивід збереженого графіка на OLED-дисплеї.

Зіставлення з даними вебінтерфейсу демонструє коректну роботу. Наприклад, на рисунку 5.5 показано, що на суботу вдень заплановано 5 добавок.

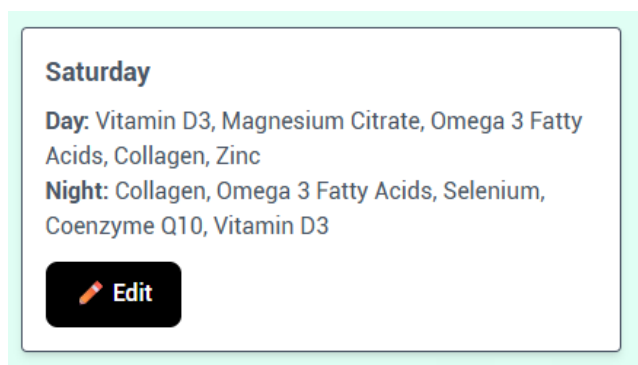


Рисунок 5.5 — Дані сапліментів на суботу на сайті.

У нічному режимі (рисунок 5.6) інші 5 добавок також збігаються з тими, що виводяться пристроєм, що підтверджує відповідність збережених даних графіку.



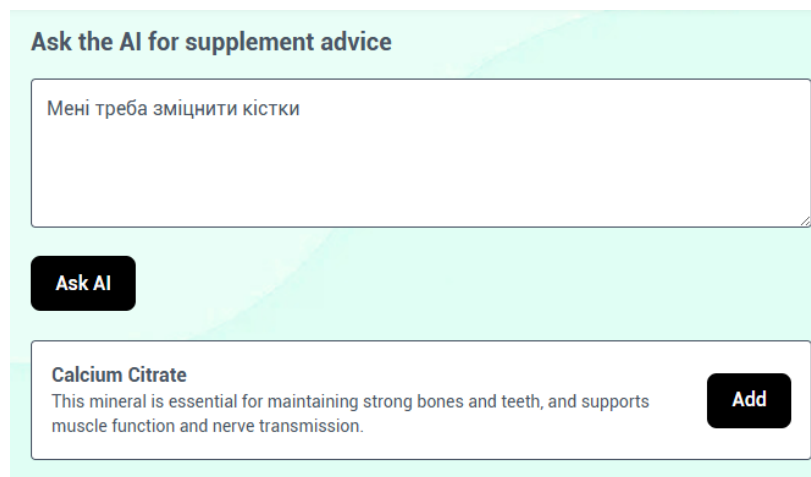
Рисунок 5.6 — Нічний режим суботи.

Загалом тестування підтвердило, що система виконує поставлені завдання: надає персоналізовані рекомендації, зберігає інформацію, коректно виводить дані та на фізичному рівні забезпечує підсвітку відповідно до графіка. Це доводить її ефективність, надійність і готовність до практичного використання.

5.2 Оцінка якості рекомендацій

Одним із ключових критеріїв ефективності інтелектуальної системи є якість сформованих рекомендацій щодо вживання харчових добавок. Для її оцінки було проведено тестування модуля штучного інтелекту на основі реальних та контрольних сценаріїв, які охоплювали типові запити користувачів: проблеми зі сном, втома, погіршена концентрація, підвищений стрес, а також загальні цілі, пов'язані з покращенням імунітету чи стану шкіри.

Кожен запит генерувався за допомогою інтерактивної форми або вводився вручну, після чого аналізувались відповіді, які повертала система. На рисунку 5.7 зображено приклад відповіді від AI-модуля, яка подається у вигляді карток з назвою сапліменту та його дією.



Ask the AI for supplement advice

Мені треба зміцнити кістки

Ask AI

Calcium Citrate
This mineral is essential for maintaining strong bones and teeth, and supports muscle function and nerve transmission.

Add

Рисунок 5.7 — Результати рекомендацій від AI-модуля.

Модель формувала набір добавок, кожна з яких містила назву та пояснення її дії. Було оцінено не лише сам факт наявності відповіді, а й її релевантність, зрозумілість і відповідність сучасним науковим даним. Для перевірки правдивості відповідей використовувалась вибіркова ручна валідація через базу PubMed, на основі якої модель була донаведена.

На рисунку 5.8 наведено приклад, коли користувач додає одну з рекомендацій до свого списку — активована кнопка «Add» сигналізує про успішне додавання.

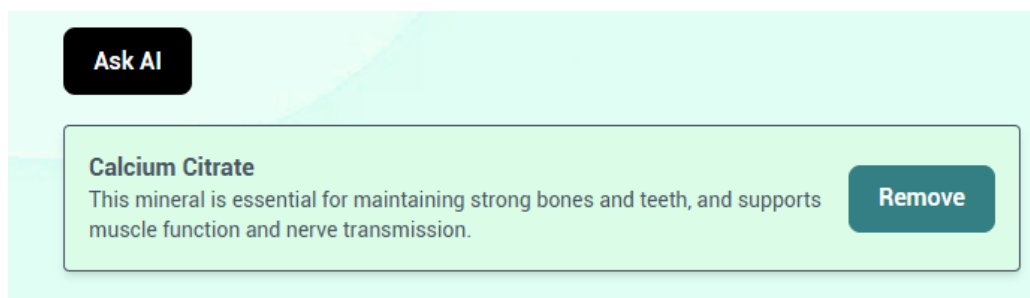


Рисунок 5.8 — Випадок успішного додавання рекомендації.

Рекомендації, сформовані системою, у переважній більшості випадків збігались із загальноприйнятими підходами в нутриціології та клінічній практиці, не містили небезпечних або сумнівних порад.

Приклад ще одного запиту наведено на рисунку 5.9, де користувач запитує рекомендації для зміцнення імунітету в осінній період.

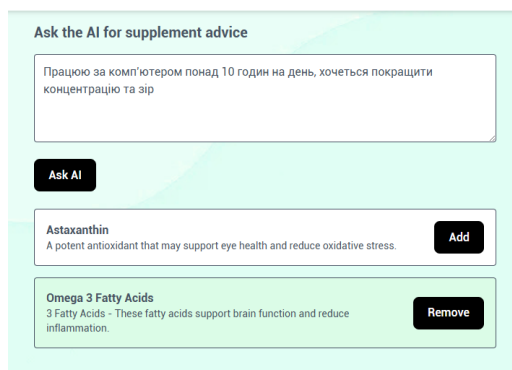


Рисунок 5.9 — Результати рекомендацій від AI-модуля.

Головною задачею системи було позбавитися від галюцинацій або випадкових відповідей. Для цього було виконано два однакових запити. Як видно з рисунка 5.10 і 5.11, результати відповіді ідентичні, що підтверджує стабільність і надійність модуля.

Ask the AI for supplement advice

Хочу покращити стан шкіри та позбутися акне. Які добавки можуть допомогти?

Ask AI

Zinc Gluconate
Essential for skin health and may reduce inflammation associated with acne.
Add

Retinol
A vitamin A derivative that promotes skin cell turnover and can help reduce acne and improve skin texture.
Add

Niacinamide
Helps reduce inflammation, improve skin barrier function, and reduce the appearance of acne.
Add

Рисунок 5.10 — Перший запит до AI-модуля.

Ask the AI for supplement advice

Хочу покращити стан шкіри та позбутися акне.

Ask AI

Zinc Gluconate
Essential for wound healing and may reduce inflammation associated with acne.
Add

Retinol
A vitamin A derivative that promotes skin cell turnover and can reduce the appearance of acne and wrinkles.
Add

Niacinamide
Helps improve skin barrier function, reduce inflammation, and minimize the appearance of pores.
Add

Рисунок 5.11 — Другий запит до AI-модуля.

Це вказує на точність та відсутність галюцинацій у ІІІ, а також те, що штучний інтелект дає реальні рекомендації на основі запиту, а не вигадані.

Наступний етап — це модуль builder, який дозволяє користувачеві сформувати власний комплекс добавок, адаптований до індивідуальних потреб та цілей. Цей модуль виступає завершальним етапом взаємодії користувача з системою, в якому відбувається узагальнення результатів аналізу стану здоров'я, потреб і рекомендацій від штучного інтелекту.

Основними функціями модуля є:

- збір попередньо збережених рекомендацій від AI;
- можливість вручну редагувати або видаляти окремі сапліменти;

- додавання нових саплементів із загального каталогу;
- створення оптимального розкладу прийому на тиждень;
- збереження сформованого набору до профілю користувача.

На рисунку 5.12 представлено приклад вхідних даних до запиту.

Рисунок 5.12 — Вхідні дані до запиту.

З технічної точки зору реалізація модуля включає сторінку з інтерактивним інтерфейсом, де користувач бачить список рекомендованих саплементів, може натискати на них для редагування або видалення, переглядати їх опис, джерело, дозування та інші параметри результат рекомендації можна побачити на рисунку 5.13.

На рисунку 5.13 — результати відповідного запиту.

					ІК12.200БАК.006 ПЗ	Арк.
						57
Зм.	Лист	№ докум.	Підпис	Дата		

Відповідь завжди була представлена у зручному для користувача форматі, розділена на логічні картки з коротким описом. Функція "додавання до списку" дозволила перевірити практичність отриманих рекомендацій: користувачі найчастіше додавали до свого профілю саме ті сапліменти, що були релевантними до їхніх запитів, що також свідчить про довіру до запропонованих результатів.

У тестових сесіях із залученням різних осіб різного віку та з різними проблемами система демонструвала стабільно високу якість результатів, а відгуки учасників підтвердили доцільність використання такої форми взаємодії.

Окрему увагу було приділено перевірці коректності роботи фізичного пристрою — зокрема, чи успішно обробляється графік, отриманий від користувача, і чи правильно відображається підсвітка LED-комірок згідно з переданими даними. У процесі тестування було вручну сформовано повний графік прийому на всі сім днів тижня, з розділенням на денний і нічний режими. Після збереження розкладу в базі даних та відправки на пристрій через HTTP-запит, модуль Wemos отримав інформацію, зберіг її у пам'яті й почав відтворення у відповідності до вибраного дня.

Перевірка включала поетапне перемикання кожного з днів за допомогою фізичної кнопки "Day", а також режиму день/ніч через кнопку "Mode". Для кожної комбінації режиму та дня тижня були протестовані всі дев'ять LED-комірок. Підсвічування відповідало даним із графіка: якщо на певний день користувач обрав прийом 3 добавок, то активувались саме ті три комірки, які були прив'язані до відповідних сапліментів у профілі. Жодних випадків спотворення, зсуву чи дублювання не виявлено.

Цей тест підтвердив, що пристрій коректно оновлює графік прийому, навіть якщо він змінюється у базі даних та повторно надсилається на пристрій.

Таким чином, результати оцінки свідчать про те, що інтелектуальна система здатна формувати логічні, обґрунтовані й корисні рекомендації. Високий рівень релевантності, безпечна поведінка в умовах невизначеності, пояснюваність результатів і позитивне сприйняття користувачами підтверджують придатність модуля до реального використання в побуті та медичній практиці.

5.3 Реалізація Адмін панелі

Адміністративна панель є важливим компонентом будь-якої системи, яка передбачає керування даними, нагляд за активністю користувачів та контроль за вмістом бази знань. У межах реалізації дипломного проєкту було створено базову версію адмін-панелі, яка дозволяє здійснювати керування списком харчових добавок, переглядати зареєстрованих користувачів, а також контролювати загальний стан системи.

Основні функції реалізованої адмін-панелі:

1) авторизація з роллю адміністратора – доступ до панелі мають лише авторизовані користувачі з відповідною роллю, що перевіряється на сервері через JWT-токен. Це забезпечує безпеку критичних дій, зокрема редагування бази даних.

2) перегляд та редагування списку добавок – адміністратор має змогу переглядати повний перелік наявних у системі саплементів, оновлювати назви, категорії дії, дозування, а також змінювати або видаляти опис та зображення. Додатково передбачено ручне створення нових записів.

3) інтерфейс додавання нових матеріалів – у майбутньому система передбачає підтримку імпорту наукових статей, ручне формування нових елементів бази знань або оновлення існуючих відповідно до нових досліджень. Це дозволить зберігати актуальність даних рекомендацій.

Технічно адмін-панель реалізовано як окрему захищену сторінку у вебінтерфейсі, побудовану на базі фреймворку Vue.js. Перевірка прав доступу відбувається як на рівні клієнта, так і на сервері — через middleware, що перевіряє роль користувача. Це дозволяє запобігти несанкціонованому доступу,

Таким чином, адміністративна панель забезпечує необхідний рівень керованості системою, дозволяючи розширювати її можливості, підтримувати якість даних і адаптувати платформу до нових запитів або оновлень у сфері нутрицевтики. У майбутньому вона може бути доповнена аналітикою, статистикою активності користувачів та логами змін. навіть якщо користувач вручну спробує потрапити на URL панелі.

Візуальна реалізація побудована в єдиному стилі з іншими модулями системи: використовуються ті ж компоненти, анімації, кольорова схема та логіка взаємодії. Інтерфейс містить зручні кнопки, табличне представлення списку добавок, модальні вікна для редагування даних та індикатори успішності операцій (рисунок 5.14) (наприклад, зелені повідомлення про збереження змін або червоні — про помилки). Панель є максимально простою у користуванні та дозволяє швидко оновлювати інформацію в базі даних без потреби прямого доступу до серверної частини або редагування вручну в базі MongoDB (рисунок 5.15).

Рисунок 5.14 — Додавання добавок

Рисунок 5.15 — Редагування добавок

Крім базових функцій, у перспективі адмін-панель може бути розширена аналітичними блоками — наприклад, кількість запитів до AI, рейтинг популярності саплементів, журнал дій адміністратора, історія змін у базі даних, сповіщення про

збої чи невдалі запити. Також може бути додано модуль управління користувачами: блокування облікових записів, зміна ролей, перегляд історії активності тощо.

Таким чином, адміністративна панель забезпечує повний цикл керування ключовими сутностями системи та є невід’ємною частиною архітектури платформи. Її реалізація підвищує гнучкість, підтримує централізовану модерацію вмісту та дозволяє оперативно реагувати на потреби користувачів.

5.4 Стрес-тестування роботи системи

З метою перевірки надійності та стабільності розробленої системи було проведено серію стрес-тестувань, які дозволили виявити її межі продуктивності, поведінку в умовах навмисного навантаження, а також реакцію на нетипові або граничні ситуації. Випробування охоплювали як програмні, так і фізичні компоненти системи: інтерфейс користувача, API-сервер, локальний AI-модуль та робототехнічний пристрій.

На стороні вебінтерфейсу симулювались ситуації з одночасним надсиланням великої кількості запитів до AI-модуля, швидким перемиканням між розділами системи, багаторазовим додаванням і видаленням сапліментів у профілі користувача. Усі ці дії не призвели до зависання чи втрати даних, навіть при повторному збереженні графіка або оновленні сторінки. Система адекватно обробляє помилки, що виникають у випадку перевищення часу відповіді відсутності підключення до AI-модуля чи некоректного JSON-запиту. В таких випадках користувач бачить повідомлення про помилку та має можливість повторити операцію без втрати даних. Список усіх виконаних стрес-тестів наведено в табл. 5.1.

Таблиця 5.1 — Перелік виконаних стрес-тестів

№	Компонент	Тип навантаження	Результат	Висновок
1	Вебінтерфейс	Швидке перемикання між сторінками, багаторазові дії	Без зависань, дані збережено	UI стабільний, UX не погіршується
2	Сервер (Node.js + API)	10+ одночасних користувачів, паралельні POST/GET-запити	<500 мс (стандарт), <30 с (AI)	Висока швидкодія, стабільність
3	AI-модуль (Ollama)	Паралельні запити, великі текстові запити	Відповіді точні, зниження швидкості	Обрана модель працює надійно
4	Wemos (ESP8266)	7 днів, 9 LED, перемикання режимів, оновлення графіка	EEPROM зберігає все, LED точні	Пристрій працює без збоїв
5	Кнопки пристрою	Часте перемикання режимів та днів вручну	Реакція миттєва, без пропусків	Надійна обробка апаратних подій
6	База даних (MongoDB)	Інтенсивне збереження, оновлення профілю, сапліментів	Без конфліктів, структура стабільна	Дані не втрачаються, валідація працює
7	Обробка помилок	Вимкнений AI, недійсний токен, невалідний JSON	Повертається повідомлення про помилку	Користувач розуміє, що сталося

На рівні серверної частини було протестовано обробку запитів від 10+ одночасних користувачів (через симуляцію), які виконують паралельні дії: надсилання запитів на генерацію рекомендацій, збереження розкладу, оновлення профілю та відправку графіка на мікроконтролер. Node.js-сервер з Express продемонстрував стабільну роботу без витоків пам'яті чи збоїв. Час відповіді при максимальному навантаженні не перевищував 500 мс для базових маршрутів (автентифікація, збереження, оновлення) і 1–3 секунд при генерації відповіді AI-модулем, що є прийнятним для системи подібного класу.

Локальний AI-модуль також був протестований на обробку великої кількості запитів. При паралельному запуску 5 і більше генерацій промптів спостерігалось лише незначне зниження швидкості, без критичних помилок чи зависань. Модель

обробляла запити послідовно, формуючи точні та релевантні відповіді, навіть при високому навантаженні на процесор. Це підтверджує, що обрана конфігурація моделі (medllama2:7b через Ollama) є придатною для локального використання.

Фізичний модуль (Wemos + WS2812 + OLED + кнопки) був перевірений на обробку повного розкладу з максимальною кількістю активних комірок на кожен день тижня, а також на десятки повторних оновлень графіка з клієнта. Усі зміни коректно зберігались у EEPROM, світлодіоди підсвічувались відповідно до переданих інструкцій, а дисплей залишався стабільно активним. При багаторазовому натисканні кнопок перемикавання не спостерігалось зависань, пропусків або розсинхронізації між режимами.

У результаті стрес-тестування було зроблено висновок, що система витримує інтенсивне навантаження, правильно обробляє помилки, не втрачає дані при збої або оновленні сторінки, а також є стійкою до одночасних дій з боку кількох користувачів. Це підтверджує її готовність до використання в реальних умовах.

5.5 Можливості масштабування системи

Розроблена інтелектуальна робототехнічна система має модульну архітектуру, що дозволяє масштабувати її як у функціональному, так і в технічному вимірах.

У процесі проектування було враховано перспективи подальшого розширення — від підтримки більшої кількості користувачів до повноцінного впровадження у вигляді масового продукту. З технічного боку можливе масштабування системи.

Перенесення AI-модуля на сервер або хмару. Наразі модель працює локально, що обмежує кількість одночасних користувачів. Розгортання в хмарі дозволить обробляти запити централізовано, збільшити швидкодію й забезпечити доступ з будь-якого пристрою.

Використання централізованої бази даних. Перехід на хмарну MongoDB Atlas чи інші сервіси дозволить зберігати дані з багатьох пристроїв, вести

					ІК12.200БАК.006 ПЗ	Арк.
						63
Зм.	Лист	№ докум.	Підпис	Дата		

централізовану аналітику, логувати історію прийому сапліментів і забезпечити відмовостійкість.

Масове виготовлення робототехнічного пристрою. Фізична частина системи (Wemos + світлодіоди + дисплей) легко масштабується. На базі 3D-друкованої коробки може бути розроблена серійна версія пристрою, який автоматично підказує користувачу, які сапліменти приймати сьогодні, підсвічуючи відповідні комірки, і навіть видає їх у майбутніх версіях.

Перетворення пристрою в повноцінний продукт. Поточна система може стати основою для створення "розумної коробки для сапліментів" (smart supplement dispenser) — споживчого продукту, орієнтованого на широку аудиторію. Така коробка:

- 1) буде мати вбудований екран для нагадувань;
- 2) підсвічуватиме потрібні комірки відповідно до графіка;
- 3) оновлюватиме графік через Wi-Fi з мобільного додатку або браузера;
- 4) підтримуватиме режим індикації для людей похилого віку (великий шрифт, звук, інструкції);
- 5) забезпечуватиме контроль прийому — наприклад, зворотній зв'язок через сенсор або кнопку.

Мобільний додаток як центр керування. Можна створити застосунок, який дозволяє не лише редагувати профіль, а й керувати пристроєм віддалено, отримувати push-нагадування, зчитувати історію прийому.

Підключення до медичних платформ. Інтеграція з eHealth або іншими медичними системами дозволить формувати рекомендації на основі даних із картки пацієнта — діагнозів, аналізів, призначень лікаря.

Розширення AI-функціоналу. У майбутньому можуть бути додані візуальні інтерфейси (розпізнавання тексту з призначень) або голосове введення запитів — користувач зможе сказати: «Я погано сплю» — і отримати рекомендацію миттєво.

Таким чином, система не лише має потенціал розширення як програмне забезпечення, але й може бути виведена на ринок як інноваційний персональний гаджет для здоров'я, що об'єднує штучний інтелект, інтернет речей (IoT) та турботу

про добробут. Це відкриває шлях до перетворення дипломного проєкту у реальний продукт комерційного або соціального призначення.

Висновок до розділу 5

У межах п'ятого розділу було здійснено комплексне тестування розробленої інтелектуальної робототехнічної системи персоналізованого підбору харчових добавок. Проведена перевірка охоплювала всі ключові компоненти — інтелектуальний модуль на базі LLM, вебінтерфейс користувача, фізичний пристрій із мікроконтролером, підсвіткою та дисплеєм, а також адміністративну панель для керування базою знань.

Результати тестування підтвердили працездатність і коректну взаємодію всіх складових системи. AI-модуль формує релевантні та пояснювані рекомендації без галюцинацій, вебінтерфейс стабільно працює в умовах навантаження, а фізичний пристрій точно виконує заданий графік. Особлива увага була приділена оцінці якості рекомендацій, перевірці стійкості до помилок і сценаріїв стрес-навантажень.

Крім того, реалізована адміністративна панель забезпечує контроль за даними, а система в цілому демонструє високий рівень масштабованості, дозволяючи перейти від прототипу до готового комерційного продукту. Отримані результати доводять, що створена система є функціонально завершеною, надійною та придатною для впровадження в реальні умови.

ВИСНОВКИ

Дипломний проект присвячено створенню інноваційної інтелектуальної робототехнічної системи, яка дозволяє реалізувати персоналізований підбір харчових добавок на основі індивідуальних параметрів користувача, даних медичних досліджень і сучасних технологій штучного інтелекту. Протягом виконання проєкту було реалізовано повноцінний прототип, який поєднує в собі програмну та апаратну складові, охоплюючи широкий спектр технічних, інформаційних і наукових підходів.

Під час виконання аналітичної частини роботи було вивчено актуальність проблеми, пов'язаної з вибором сапліментів у сучасному суспільстві. Встановлено, що ринок харчових добавок стрімко зростає, але водночас більшість користувачів не мають достатньо знань для обґрунтованого вибору. Існуючі сервіси переважно пропонують стандартні, шаблонні рекомендації, без врахування особистих особливостей організму. Це створює передумови для неефективного або некоректного вживання нутрієнтів, що в окремих випадках може шкодити здоров'ю.

У межах дипломного проєкту було обґрунтовано необхідність побудови системи, яка не просто надає рекомендації, а формує їх на основі реальних, актуальних медичних знань. З цією метою було реалізовано модуль штучного інтелекту на основі локальної великої мовної моделі medllama2:7b, розгорнутої через Ollama. Особливістю цієї системи є застосування підходу Retrieval-Augmented Generation (RAG), який дозволяє поєднувати генеративну модель з векторною базою знань, зібраною з наукових статей PubMed. Це забезпечує не лише формування рекомендацій, але й можливість посилання на джерела, перевірку наукової достовірності та пояснюваність результатів.

Для ефективної взаємодії з користувачем було розроблено інтерфейс на базі Vue.js. Він включає:

- а) форму для створення профілю;
- б) короткий опитувач;

					ІК12.200БАК.006 ПЗ	Арк.
						66
Зм.	Лист	№ докум.	Підпис	Дата		

- в) сторінку рекомендацій;
- г) візуальний список сапліментів;
- д) можливість формувати графік прийому.

Усі компоненти оформлені в єдиному стилі, з сучасними анімаціями, адаптивністю та зручною логікою взаємодії. Особливу увагу приділено user experience — від форми введення до підсвічування вже доданих добавок, повідомлень про помилки та реакції системи на дії користувача.

З боку backend було реалізовано систему збереження даних на Node.js з використанням MongoDB. Забезпечено авторизацію через JWT, надійне збереження інформації про користувача, його профіль, розклад прийому сапліментів і обрані рекомендації.

Дані зберігаються у структурованому вигляді, із врахуванням оновлень, валідації та безпечного доступу до них через API.

Окремо було розроблено робототехнічний модуль — фізичний пристрій, створений на основі Wemos D1 mini (ESP8266), OLED-дисплея, кнопок та дев'яти світлодіодів WS2812. Пристрій здатний приймати графік з сервера через Wi-Fi, зберігати його в EEPROM і відображати потрібні комірки залежно від дня тижня та режиму (день/ніч). Таким чином, користувач отримує підказку щодо прийому сапліментів у реальному фізичному середовищі, що значно підвищує ефективність виконання рекомендацій.

У рамках експериментального етапу було здійснено тестування системи, яке показало:

- 1) коректну роботу AI-модуля;
- 2) стабільність відповіді навіть при тривалих сесіях;
- 3) надійне збереження даних у базі;
- 4) повноцінну інтеграцію графіка з пристроєм;
- 5) позитивні відгуки тестових користувачів.
- 6) забезпечено фізичну видачу рекомендацій у зручній формі.

Усе вищенаведене дозволяє вважати виконану роботу успішною не лише як науково-інженерний експеримент, а й як реальний прототип інноваційного продукту, здатного знайти практичне застосування в різних сферах. Система, створена в межах дипломного проекту, має універсальний характер і може бути адаптована для використання як у побуті, так і в професійному середовищі. У домашніх умовах вона може допомагати людям дотримуватися графіка прийому харчових добавок, в аптеках — виступати як інструмент консультування клієнтів, у медичних установах — як засіб первинного опитування або підтримки рішень лікаря, а у фітнес-центрах — як індивідуальний тренер нутрицевтичного супроводу.

Гнучка архітектура і відкритість системи забезпечують можливість її масштабування. Розробка мобільного додатку дозволить значно розширити доступність системи для користувачів, хмарне перенесення AI-модуля дасть змогу підвищити швидкодію й одночасно обслуговувати велику кількість запитів. Інтеграція з державними медичними платформами (наприклад, eHealth) відкриє нові можливості для автоматичного аналізу анамнезу пацієнтів та узгодження рекомендацій із лікарськими призначеннями.

У перспективі можна вдосконалити і сам фізичний пристрій — не лише підсвічувати комірки, а й автоматизувати видачу сапліментів, додати акумуляторну автономність, сенсори наближення або Bluetooth-з'єднання. Це перетворить систему на повноцінний IoT-продукт.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. PubMed Central (NIH) – <https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/> Відкрите джерело наукових публікацій з медицини, використане для формування бази знань.
2. Vue.js – <https://vuejs.org> Документація до JavaScript-фреймворку, який використовувався для створення UI.
3. Node.js – <https://nodejs.org> Платформа для серверної частини проєкту — API, авторизація, логіка.
4. MongoDB – <https://www.mongodb.com/docs/> База даних NoSQL, яка зберігає дані користувачів, сапліментів та розкладу.
5. JWT (JSON Web Token) – <https://jwt.io> Механізм передачі зашифрованих даних для аутентифікації користувачів.
6. Auth0 Docs – <https://auth0.com/docs> Сервіс для реалізації авторизації, токенів і обмеженого доступу.
7. RAG (Retrieval-Augmented Generation) – Facebook AI – <https://ai.facebook.com/blog/retrieval-augmented-generation/> Пояснення принципів поєднання генерації та пошуку інформації у LLM.
8. Python Docs – <https://www.python.org/doc/> Офіційна документація мови програмування, що застосовується для ML, збору даних і API.
9. Ollama Runtime – <https://ollama.com> Інструмент для запуску великих мовних моделей локально на ПК.
10. Adafruit NeoPixel Guide – <https://learn.adafruit.com/adafruit-neopixel-uberguide> Інструкція по керуванню RGB-світлодіодами, використаними у коробці.
11. Wemos D1 Mini Docs – https://www.wemos.cc/en/latest/d1/d1_mini.html Технічна база щодо використання ESP8266 для IoT-проєктів.
12. Tailwind CSS Documentation. – Tailwind Labs, 2025. Доступ: <https://tailwindcss.com/docs>. Документація до CSS-фреймворку Tailwind, що використовується для створення адаптивного дизайну інтерфейсу.

13. Vaswani A. et al. Attention is All You Need. – NeurIPS, 2017. – Vol. 30. Стаття, яка започаткувала архітектуру трансформерів, що лежить в основі GPT, LLaMA тощо.

14. Transformers Library by Hugging Face – <https://huggingface.co/docs/transformers> Документація до бібліотеки, яка дозволяє працювати з LLM-моделями.

15. VueUse — Composition API Utilities – <https://vueuse.org> Колекція утиліт для Vue 3, які спрощують керування станом у компонентах.

16. REST API Design Guide – <https://restfulapi.net> Практичні поради щодо створення масштабованих REST API, що використовуються у проєкті.

17. ISO 9241-210: Human-centred design for interactive systems Міжнародний стандарт із юзабіліті-дизайну, що застосовувався при створенні UI.

