

ХЕРСОНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ
КАФЕДРА ПРОГРАМНИХ ЗАСОБІВ І ТЕХНОЛОГІЙ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи

бакалавра

(освітньо-кваліфікаційний рівень)

на тему «Розробка інтелектуального чат-бота для
автоматизації взаємодії з користувачами Інтернет-магазину»

Виконав: здобувач 4 року навчання
групи 4ПР1

напряму підготовки (спеціальності)
121-Інженерія програмного забезпечення
(шифр і назва напряму підготовки, спеціальності)

Агеєнко Андрій Федорович
(підпис, прізвище та ініціали)

Керівник Огнєва Оксана Євгенівна
(підпис, прізвище та ініціали)

Рецензент к.т.н., доцент Корніловська Н.В.
(прізвище та ініціали)

Нормоконтролер Комісаров О. С.
(підпис, прізвище та ініціали)

ХЕРСОНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Інститут, факультет, відділення Інформаційних технологій та дизайну
Кафедра, циклова комісія Програмних засобів і технологій
Освітньо-кваліфікаційний рівень бакалавр
Освітньо-професійна підготовка Програмна інженерія
(шифр і назва)
Спеціальність 121-Інженерія програмного забезпечення
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри програмних засобів і технологій

О.Є. Огнєва
« » 2026 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Агєєнка Андрія Федоровича

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Розробка інтелектуального чат-бота для автоматизації взаємодії з користувачами Інтернет-магазину

керівник проєкту (роботи) Огнєва Оксана Євгенівна, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від «16» січня 2026 року №82-с

2. Строк подання здобувачем проєкту(роботи) 10.06.2026

3. Вихідні дані до проєкту (роботи) літературні та періодичні джерела, матеріали переддипломної практики

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
Аналіз предметної області, проєктування інтелектуального чат-бота для автоматизації взаємодії з користувачами Інтернет-магазину, розробка інтелектуального чат-бота для автоматизації взаємодії з користувачами Інтернет-магазину, тестування та впровадження інтелектуального чат-бота для автоматизації взаємодії з користувачами Інтернет-магазину

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
13 таблиць, 36 рисунків

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 14.01.2026

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів проєкту (роботи)	Примітка
1	Отримання завдання	14.01.2026	Виконано
2	Підбір літератури	17.01.2026-21.01.2026	Виконано
3	Аналіз предметної області	22.01.2026-26.01.2026	Виконано
4	Розробка та обґрунтування завдання	27.01.2026-12.02.2026	Виконано
5	Розробка концептуальної моделі	13.02.2026-19.02.2026	Виконано
6	Моделювання та проєктування системи	20.02.2026-02.03.2026	Виконано
7	Розробка чат-бота	03.03.2026-15.04.2026	Виконано
8	Тестування чат-бота	16.04.2026-20.04.2026	Виконано
9	Оформлення пояснювальної записки	21.04.2026-02.05.2026	Виконано
10	Захист кваліфікаційної роботи	24.06.2026	Виконано

Здобувач _____ *Агєєнко А.Ф.*
(підпис) (прізвище та ініціали)

Керівник проєкту (роботи) _____ *Огнєва О.Є.*
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Кваліфікаційна робота бакалавра: 77 сторінок, 36 рисунків, 13 таблиць, 3 додатки, 30 джерел.

Мета роботи:

Розробити інтелектуальний чат-бот для автоматизації взаємодії з користувачами Інтернет-магазину. Чат-бот повинен забезпечувати обробку запитів користувачів у режимі реального часу, надання консультацій та рекомендацій товарів, а також допомогу у виконанні основних дій в Інтернет-магазині. Реалізація такого рішення дозволить підвищити швидкість обслуговування клієнтів, зменшити навантаження на персонал та покращити якість користувацького досвіду.

Об'єкт дослідження:

Технології проєктування та розробки веб-додатків та віджетів для них. У роботі досліджуються сучасні підходи до створення інтерактивних веб-систем, використання клієнт-серверної архітектури та інтеграції чат-ботів на основі штучного інтелекту у веб-застосунки. Особлива увага приділяється технологіям обробки природної мови та організації взаємодії користувача з програмною системою.

Предмет дослідження:

Проєктування та розробка інтелектуального чат-бота для автоматизації взаємодії з користувачами Інтернет-магазину. Предметом дослідження є методи аналізу текстових запитів користувачів, механізми генерації відповідей та способи інтеграції чат-бота з базою даних товарів Інтернет-магазину. Також розглядаються питання реалізації рекомендаційної системи та підтримки актуальності інформації про товари.

Інструменти та засоби розробки:

Для розробки чат-бота було використано Node.js, Express, React, GPT-4.1 nano та Visual Studio Code. Node.js і Express застосовувалися для створення серверної частини системи та API, а React використовувався для реалізації

інтерактивного інтерфейсу користувача. Модель GPT-4.1 nano забезпечує аналіз природної мови, розпізнавання намірів користувача та генерацію релевантних відповідей у межах предметної області Інтернет-магазину.

Результати роботи:

Результатом роботи є інтелектуальний чат-бот, інтегрований у систему Інтернет-магазину та здатний автоматизувати процес взаємодії з користувачами. Розроблена система дозволяє здійснювати пошук товарів, надавати рекомендації, консультувати користувачів щодо роботи магазину та підтримувати діалог у режимі реального часу. Проведене тестування підтвердило працездатність, надійність та зручність використання розробленого програмного продукту.

Новизна рішення:

Розроблено інтелектуальний чат-бот для Інтернет-магазину, який забезпечує автоматизовану обробку запитів користувачів та надання рекомендацій товарів у режимі реального часу. Особливістю рішення є інтеграція з базою даних Інтернет-магазину, що дозволяє враховувати зміни у каталозі товарів та підтримувати актуальність відповідей без ручного оновлення даних. Запропонований підхід може бути використаний для автоматизації взаємодії з користувачами і в інших веб-системах електронної комерції.

КЛЮЧОВІ СЛОВА: ВЕБ-САЙТ, ВЕБ-ДОДАТОК, ПРОГРАМНА СИСТЕМА, ЧАТ-БОТ, КОРИСТУВАЦЬКА ПІДТРИМКА, ШТУЧНИЙ ІНТЕЛЕКТ, ІНТЕРНЕТ-МАГАЗИН.

АНОТАЦІЯ

Кваліфікаційна робота бакалавра включає такі структурні частини: вступ, чотири розділи, висновки, список використаних джерел та додатки.

Перший розділ «Аналіз предметної області» містить огляд предметної області. Розділ описує ключові процеси, що відбуваються у межах Інтернет-магазину. Також у розділі наведено мету – перейти від моделі «ЯК Є» до моделі «ЯК ПОВИННО БУТИ» та постановку завдання, виконання якого зможе досягти цієї мети.

Другий розділ «Проектування інтелектуального чат-бота для автоматизації взаємодії з користувачами Інтернет-магазину» складається з таких підрозділів: «Основні вимоги до чат-бота», «Розробка проєктних специфікацій», «Розробка поведінкових діаграм», «Розробка діаграм потоків даних», «Проектування архітектури чат-бота», «Розробка структурних UML-діаграм» та «Висновки до розділу». Після детального аналізу предметної області у було спроектовано весь необхідний функціонал чат-бота, який відповідає сучасним критеріям розробки, також змодельовані процеси, які виникають від час взаємодії користувача з чат-ботом та чат-бота з сервером. Розроблено основні вимоги до чат-бота, проєктні специфікації, побудовані та описані діаграми для моделювання, описано логіку чат-бота.

Третій розділ «Розробка інтелектуального чат-бота для автоматизації взаємодії з користувачами Інтернет-магазину» складається з трьох підрозділів: «Обґрунтування вибору інструментальних засобів розробки», «Розробка чат-бота», «Висновки до розділу». У цьому розділі було розглянуто інструментальні засоби розробки та обрано ті, що найкраще виконують поставлене завдання. React було обрано завдяки можливості оновлювати зміст веб-сторінки в реальному часі у відповідь на дії користувача, Node.js та Express було обрано для кращої сумісності з Інтернет-магазином, до якого інтегрується чат-бот, а GPT-4.1 nano було обрано як дешеву але ефективну мовну модель,

що зможе обробляти запити користувачів та генерувати відповіді на них. Після цього було проведено розробку чат-бота за допомогою обраних інструментів.

Четвертий розділ «Тестування та впровадження інтелектуального чат-бота для автоматизації взаємодії з користувачами Інтернет-магазину» складається з чотирьох підрозділів: «Тестування чат-бота», «Робота користувача з чат-ботом», «Встановлення та експериментальне застосування чат-бота», «Висновки до розділу». Розроблений чат-бот пройшов ряд тестових випробувань на предмет виявлення помилок у проектуванні та реалізації. Перевірка підтвердила відповідність вимогам функціональності, надійності, зручності використання. Також було описано процеси встановлення та використання чат-бота кінцевим користувачем.

ABSTRACT

The bachelor's thesis consists of the following sections: an introduction, four chapters, conclusions, a list of references, and appendices.

The first chapter, “Analysis of the Subject Area,” provides an overview of the subject area. The chapter describes the key processes occurring within an online store. It also outlines the goal—to transition from the “AS IS” model to the “AS IT SHOULD BE” model—and sets forth the task whose completion will achieve this goal.

The second section, “Designing an Intelligent Chatbot to Automate Interaction with Online Store Users,” consists of the following subsections: “Basic Requirements for the Chatbot,” “Development of Project Specifications,” “Development of Behavior Diagrams,” “Development of Data Flow Diagrams,” “Design of the Chatbot Architecture,” “Development of Structural UML Diagrams,” and “Conclusions to the Section.” Following a detailed analysis of the subject area, all necessary chatbot functionality was designed in accordance with modern development criteria, and the processes arising from user interactions with the chatbot and between the chatbot and the server were modeled. The basic requirements for the chatbot and project specifications were developed, modeling diagrams were constructed and described, and the chatbot’s logic was outlined.

The third chapter, “Development of an Intelligent Chatbot to Automate Interaction with Online Store Users,” consists of three sections: “Justification for the Choice of Development Tools,” “Chatbot Development,” and “Conclusions to the Chapter.” In this chapter, development tools were reviewed, and those best suited to the task at hand were selected. React was chosen for its ability to update web page content in real time in response to user actions; Node.js and Express were selected for better compatibility with the online store into which the chatbot is integrated; and GPT-4.1 nano was chosen as a cost-effective yet efficient language model capable of processing user queries and generating responses. Following this, the chatbot was developed using the selected tools.

The fourth chapter, “Testing and Implementation of an Intelligent Chatbot for Automating Interaction with Online Store Users,” consists of four subsections: “Chatbot Testing,” “User Interaction with the Chatbot,” “Installation and Experimental Application of Chatbot,” and “Conclusions to the Chapter.” The developed chatbot underwent a series of tests to identify errors in design and implementation. The testing confirmed compliance with requirements for functionality, reliability, and ease of use. The processes for installing and using the chatbot by the end user were also described.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	12
ВСТУП	13
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	16
1.1. Опис та загальна характеристика предметної області	16
1.2. Моделювання та аналіз бізнес-процесів предметної області	19
1.3. Постановка задачі проєктування	26
1.4. Висновки до розділу 1	28
РОЗДІЛ 2. ПРОЄКТУВАННЯ ІНТЕЛЕКТУАЛЬНОГО ЧАТ-БОТА ДЛЯ АВТОМАТИЗАЦІЇ ВЗАЄМОДІЇ З КОРИСТУВАЧАМИ ІНТЕРНЕТ- МАГАЗИНУ	29
2.1. Основні вимоги до чат-бота	29
2.2. Розробка проєктних специфікацій	32
2.3. Розробка поведінкових діаграм	35
2.4. Розробка діаграм потоків даних	40
2.5. Проєктування архітектури чат-бота	42
2.6. Розробка структурних UML-діаграм	42
2.7. Висновки до розділу 2	44
РОЗДІЛ 3. РОЗРОБКА ІНТЕЛЕКТУАЛЬНОГО ЧАТ-БОТА ДЛЯ АВТОМАТИЗАЦІЇ ВЗАЄМОДІЇ З КОРИСТУВАЧАМИ ІНТЕРНЕТ- МАГАЗИНУ	46
3.1. Обґрунтування вибору інструментальних засобів розробки	46
3.2. Розробка чат-бота	48
3.3. Висновки до розділу 3	50
РОЗДІЛ 4. ТЕСТУВАННЯ ТА ВПРОВАДЖЕННЯ ІНТЕЛЕКТУАЛЬНОГО ЧАТ-БОТА ДЛЯ АВТОМАТИЗАЦІЇ ВЗАЄМОДІЇ З КОРИСТУВАЧАМИ ІНТЕРНЕТ-МАГАЗИНУ	52
4.1. Тестування чат-бота	52
4.2. Робота користувача з чат-ботом	62
4.3. Встановлення та експериментальне застосування чат-бота	65
4.4. Висновки до розділу 4	69

ВИСНОВКИ.....	70
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	72
ДОДАТОК А.....	75
ДОДАТОК Б	76

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

БД – бази даних.

ПЗ – програмне забезпечення.

ПС – програмна система.

СУБД – системи управління базами даних.

API – прикладний програмний інтерфейс (інтерфейс програмування застосунків, інтерфейс прикладного програмування).

CSS – це спеціальна мова стилю сторінок, що використовується для опису їхнього зовнішнього вигляду.

HTML – стандартизована мова розмітки документів для перегляду веб-сторінок у браузері.

HTTP – протокол передачі даних, що використовується в комп'ютерних мережах.

JSON – це текстовий формат обміну даними між комп'ютерами.

PHP – скриптова мова програмування, була створена для генерації HTML-сторінок на стороні веб-сервера.

NLP – алгоритми обробки природної мови.

ШІ – штучний інтелект.

LLM – велика мовна модель.

JS – мова програмування JavaScript.

REST – архітектурний стиль взаємодії компонентів веб-систем.

JWT – JSON Web Token, стандарт для передачі даних авторизації.

CRUD – базові операції з даними (Create, Read, Update, Delete).

ВСТУП

Підтримка користувачів передбачає допомогу користувачам з проблемами та питаннями щодо товару або послуги шляхом надання вказівок, рішень та порад. Ефективна взаємодія між користувачем і системою є ключовою складовою успішного функціонування електронної комерції, так як вона впливає на рівень задоволеності клієнтів, швидкість обслуговування та конверсію [28]. Актуальним є впровадження інтелектуальних систем, здатних автоматизувати цей процес, забезпечуючи швидку обробку запитів, персоналізацію рекомендацій та підтримку користувача.

Таким чином, для Інтернет-магазину важливим є забезпечення своєчасної та релевантної відповіді на запити користувачів, що може бути досягнуто за допомогою інтелектуальних чат-ботів. Вони дозволяють автоматизувати обробку звернень, зменшити навантаження на персонал та підвищити ефективність обслуговування клієнтів.

Інтелектуальний чат-бот – програмна система, що використовує методи штучного інтелекту для аналізу запитів користувачів та формування відповідей у межах Інтернет-магазину. До таких дій належать пошук товарів, надання рекомендацій та інші функції, що спрямовані на покращення користувацького досвіду. Чат-бот може функціонувати на основі попередньо визначених сценаріїв або із застосуванням сучасних технологій обробки природної мови.

Стрімкий розвиток інформаційних технологій та електронної комерції сприяв значному зростанню кількості користувачів і навантаженню на служби підтримки. Це, у свою чергу, ускладнює процес взаємодії та підвищує вимоги до швидкості і якості обслуговування. Використання чат-ботів дозволяє вирішити ці проблеми шляхом автоматизації простих процесів, зменшення впливу людського фактора та підвищення рівня персоналізації сервісу.

Актуальність теми зумовлена стрімким розвитком електронної комерції та зростанням вимог користувачів до швидкості, зручності й персоналізації обслуговування в Інтернет-магазинах. Збільшення кількості запитів і обсягів

даних ускладнює процес взаємодії між користувачем і системою, що підвищує навантаження на служби підтримки та знижує ефективність традиційних підходів. У цих умовах впровадження чат-ботів є доцільним рішенням, оскільки вони дозволяють автоматизувати обробку звернень, забезпечити оперативний зворотний зв'язок і підвищити рівень персоналізації сервісу, що загалом сприяє покращенню якості обслуговування та конкурентоспроможності Інтернет-магазину.

Об'єкт дослідження – технології проектування та розробки веб-додатків та віджетів для них.

Предмет дослідження – проектування та розробка інтелектуального чат-бота для автоматизації взаємодії з користувачами Інтернет-магазину.

Методи дослідження – аналіз інформації про існуючі веб-ресурси електронної комерції та системи автоматизації підтримки користувачів, зокрема, їх структурні та функціональні особливості; теоретичний аналіз і узагальнення інформаційних джерел з тематики розробки веб-додатків та інтелектуальних чат-ботів; а також методи моделювання і проектування, що застосовуються для створення баз даних, архітектури системи та окремих модулів веб-додатку, які забезпечать ефективну інтеграцію чат-бота з Інтернет-магазином.

Мета кваліфікаційної роботи бакалавра – розробка інтелектуального чат-бота для автоматизації взаємодії з користувачами Інтернет-магазину, що забезпечує обробку запитів у режимі реального часу, надання релевантних рекомендацій товарів та підвищення ефективності обслуговування клієнтів.

Основним завданням кваліфікаційної роботи бакалавра на тему «Розробка інтелектуального чат-бота для автоматизації взаємодії з користувачами Інтернет-магазину» є аналіз технологій та розробка модуля чат-бота на основі отриманих знань та навичок.

Для досягнення поставленої мети необхідно вирішити такі **задачі**:

- проаналізувати сучасні підходи та існуючі рішення у сфері електронної комерції;

- визначити вимоги до функціональності та структури системи;
- спроектувати архітектуру чат-бота та інтеграцію з Інтернет-магазином;
- реалізувати основні функції чат-бота, включаючи обробку запитів від користувача з пошуком необхідної інформації у базах даних;
- провести тестування та оцінку ефективності розробленого рішення.

Наукова новизна одержаних результатів:

- розроблено інтелектуальний чат-бот для Інтернет-магазину, який забезпечує автоматизовану обробку запитів користувачів та надання рекомендацій товарів у режимі реального часу, причому система дозволяє враховувати зміни в каталозі товарів і забезпечує актуальність відповідей без необхідності ручного оновлення даних;
- показано, що розроблений чат-бот має не лише теоретичне, а й практичне значення, а запропонований підхід може бути застосований до інших веб-систем, де необхідна автоматизація взаємодії з користувачами та підтримка актуальності інформації.

Практичне значення одержаних результатів. Розроблений чат-бот розвиває теорію проєктування та розробки веб-додатків та віджетів для них та сприяє покращенню взаємодії користувачів з веб-ресурсами електронної комерції з використанням чат-ботів на основі штучного інтелекту.

Структура: Кваліфікаційна робота бакалавра складається з 77 сторінок, 36 рисунків, 13 таблиць, 3 додатки, 30 джерел.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Опис та загальна характеристика предметної області

У сфері сучасної електронної комерції стає все більш поширеним використання технологій штучного інтелекту. ШІ – технологія, що базується на методах обробки мови та машинного навчання і дозволяє чат-боту розуміти наміри користувача, аналізувати контекст запиту та генерувати відповіді у зрозумілій для людини формі.

Такі технології реалізуються у вигляді інтелектуальних чат-ботів, які здатні виконувати різноманітні функції, пов'язані з підтримкою процесу покупок. Традиційні чат-боти базувалися на жорстко заданих сценаріях. Вони відповідали лише на підготовлені питання підготовленими відповідями, у інших випадках користувач отримував іншу шаблонну відповідь про нездатність бота надати допомогу. Сучасні системи можуть адаптуватися та навчатися, що значно підвищує ефективність їх використання у бізнес-процесах [23].

За оцінками Gartner, провідної дослідницької компанії США, близько 80% підприємств електронної комерції вже використовують або планують впровадження чат-ботів на основі штучного інтелекту, що свідчить про високу актуальність і перспективність даного напрямку розвитку [15]. Це зумовлено тим, що застосування штучного інтелекту дозволяє суттєво покращити користувацький досвід, підвищити рівень конверсії та оптимізувати витрати підприємства на підтримку користувачів.

Функціональні можливості чат-ботів у сфері електронної комерції сучасності значно розширилися порівняно з інформаційними системами минулих часів. Зокрема, такі системи здатні надавати персоналізовані рекомендації товарів на основі аналізу поведінки та запитів користувача, надавати інформацію про характеристики товарів, аналізувати відгуки та надавати допомогу з навігацією по додатку та зі здійсненням різних дій,

наприклад, авторизація, реєстрація нового облікового запису, редагування даних акаунта, покупка товарів онлайн, перегляд історії замовлень тощо.

Тобто такі боти можуть автоматизувати обробку простих та поширених запитів користувачів, надавати консультації, формувати індивідуальні пропозиції та навіть ініціювати взаємодію з клієнтом, наприклад, у випадку покинутого кошика або повторного відвідування сайту. Такий підхід дозволяє значно підвищити обсяг продажів.

Однією з переваг використання розмовного штучного інтелекту є можливість забезпечення цілодобової взаємодії з користувачами без залучення персоналу. Також це дозволяє зменшити навантаження на співробітників з відділу підтримки користувачів, що є перевагою як для великого бізнесу з крупними потоками користувачів, так і для малого бізнесу, де за користувацьку підтримку відповідає кілька співробітників або взагалі такого відділу немає і персонал має виконувати кілька ролей одночасно. Можливість зменшення навантаження на персонал підвищує конкурентоспроможність магазину.

Суттєвою перевагою сучасних чат-ботів є їх інтеграція з різними інформаційними системами підприємств, такими як CRM-системи, системи управління замовленнями, бази даних товарів та аналітичні платформи. Це означає, що бот зможе надавати точні відповіді користувачу про товари, використовуючи реальну інформацію з бази даних.

Також варто приділити увагу універсальності сучасних рішень, яка передбачає можливість використання чат-ботів не лише на веб-сайті, але й у месенджерах, соціальних мережах та інших додатках, де є можливість інтегрувати чат-бота. Це дозволяє значно розширити аудиторію користувачів і підвищити доступність сервісу. Значна частина користувачів здійснює покупки саме під впливом соціальних мереж [3]. Соціальні мережі також дають можливість малому бізнесу набрати клієнтів, навіть одна рекламна кампанія, яку, до речі, можна легко провести за допомогою вбудованих інструментів, наприклад, Facebook Ads, може перетворити магазин з маловідомого на популярний за лічені дні.

Інноваційним напрямом розвитку є використання мультимодальних можливостей чат-ботів, зокрема підтримка пошуку товарів за зображеннями. Такий підхід дозволяє користувачам знаходити необхідні товари без формування текстового запиту, що є перевагою у випадках, коли складно описати характеристики продукту словами або коли треба дати детальний опис товару швидко (за кілька секунд). Детальний текстовий опис може забрати кілька хвилин часу, які могли були б використані на більш корисні дії, включаючи замовлення товару. Кілька хвилин можуть здаватися чимось незначним, але для великих магазинів зі значною кількістю клієнтів такі затримки означають великі втрати потенціального доходу.

Але треба не забувати, що ефективність функціонування чат-бота значною мірою залежить від якості його проєктування, від побудови сценаріїв взаємодії та логіки обробки запитів. Незважаючи на використання великих мовних моделей, важливим залишається питання розробки структури діалогу, що враховує особливості поведінки користувачів та специфіку бізнес-процесів Інтернет-магазину. Також навіть сучасні LLM (мовні моделі) можуть допускати помилки, плутати та вигадувати інформацію, особливо якщо розробник дає моделі складні дані та недостатньо детальні інструкції.

Застосування розмовного штучного інтелекту також дозволяє здійснювати збір та аналіз даних про взаємодію користувачів із системою. Отримані дані можуть використовуватися для вдосконалення бізнес-процесів, оптимізації асортименту, підвищення ефективності маркетингових кампаній та покращення якості обслуговування клієнтів. Але деяку інформацію збирати та аналізувати заборонено відповідно до статті 182 Кримінального кодексу України [4]. До такої інформації належить державна таємниця або конфіденційні дані. Такі порушення караються штрафами, виправними роботами або обмеженням волі. Обсяг покарання залежить від заподіяної шкоди. Крім того, такі порушення можуть завдати непоправну репутаційну шкоду.

Отже, для предметної області, що розглядається, є актуальним впровадження технологій штучного інтелекту та чат-ботів на його базі. Це дозволить автоматизувати процес взаємодії з користувачами, знизити навантаження на персонал, підвищити ефективність бізнес-процесів, забезпечити персоналізований підхід до кожного клієнта та створити новий рівень користувацького досвіду.

1.2. Моделювання та аналіз бізнес-процесів предметної області

Ефективність функціонування Інтернет-магазину залежить від визначення, аналізу та оптимізації бізнес-процесів, що будуть відбуватися у цьому магазині. Перед розробкою чат-бота необхідно дослідити існуючі процеси предметної області, визначити їх структуру, взаємозв'язки, учасників. Це дозволить виявити слабкі місця в системі та проблеми, які можна вирішити розробкою та впровадженням чат-бота у веб-додаток

Предметною областю даної роботи є сфера е-комерції, а саме Інтернет-магазин «Survivalist Shop», що дає можливість придбати товари різних категорій через веб-застосунок онлайн. Основними об'єктами предметної області є користувачі, товари, замовлення, відгуки та фінансові операції. Бізнес-процеси пов'язують ці об'єкти та формують основу функціоналу онлайн-магазину.

Загальна характеристика бізнес-процесів

Бізнес-процеси Інтернет-магазину можна умовно поділити на кілька основних груп:

- процеси взаємодії з користувачем;
- процеси управління товарами;
- процеси обробки замовлень;
- процеси управління користувачами;
- процеси фінансових операцій;
- процеси збору та аналізу даних.

Кожен із цих процесів включає набір операцій, що виконуються акторами (учасники системи) або автоматично серверною частиною.

Було проведено узагальнення інформації про діяльність користувачів (гостьових та зареєстрованих), адміністраторів та системи для того, щоб сформувати цілісну модель бізнес-процесів.

Бізнес-процес перегляду каталогу товарів

Одним із базових процесів є перегляд каталогу товарів (табл. 1.1), тому що він є першим етапом взаємодії користувача з додатком. Каталог доступний для зареєстрованих та незареєстрованих користувачів.

Таблиця 1.1

Характеристика бізнес-процесу перегляду каталогу

Назва характеристики	Значення характеристики
Назва бізнес-процесу	Перегляд каталогу товарів
Вхідна подія	Перехід на головну сторінку
Основні учасники	Гість, користувач
Вхідні дані	Дані про товари з бази
Вихідні дані	Список товарів
Вихідна подія	Відображення каталогу
Клієнт	Користувач

Бізнес-процес перегляду товару

Наступним етапом є перегляд товару (табл. 1.2), який дозволяє користувачу отримати повну інформацію перед покупкою. Користувач має клікнути на товар зі списку, відкриється сторінка цього товару з повною інформацією про цей товар, відгуками та кнопкою покупки.

Таблиця 1.2**Характеристика бізнес-процесу перегляду товару**

Назва характеристики	Значення характеристики
Назва бізнес-процесу	Перегляд товару
Вхідна подія	Вибір товару
Основні учасники	Користувач
Вхідні дані	ID товару
Вихідні дані	Дані товару, відгуки, кнопка «Купити»
Вихідна подія	Відображення сторінки товару
Клієнт	Користувач

Цей процес є критично важливим, оскільки безпосередньо впливає на рішення користувача щодо покупки. Детальний опис товару та значна кількість реальних відгуків має значні шанси на завоювання довіри користувача та подальше замовлення.

Бізнес-процес реєстрації та авторизації

Для доступу до повного функціоналу системи користувач повинен пройти процедуру реєстрації та авторизації (табл. 1.3).

Таблиця 1.3**Характеристика бізнес-процесу реєстрації**

Назва характеристики	Значення характеристики
Назва бізнес-процесу	Реєстрація користувача
Вхідна подія	Заповнення форми
Основні учасники	Користувач
Вхідні дані	Email, пароль, ім'я

Вихідні дані	Новий обліковий запис
Вихідна подія	Успішна реєстрація
Клієнт	Користувач

Авторизація передбачає перевірку введених даних та надання доступу до системи через JWT-токен, що є компактим, ефективним та безпечним аналогом звичайних сесій [24]. Цей процес забезпечує ідентифікацію користувача та контроль доступу до різних функцій системи. Наприклад, звичайний користувач матиме доступ до базового функціоналу, а адміністратор матиме доступ до інструментів управління користувачами, товарами та перегляду аналітичних даних.

Бізнес-процес управління кошиком

Кошик є проміжним етапом між вибором товару та оформленням замовлення (табл. 1.4). Процес включає додавання, зміну кількості та видалення товарів із кошика. Система автоматично обчислює загальну вартість замовлення.

Таблиця 1.4

Характеристика бізнес-процесу роботи з кошиком

Назва характеристики	Значення характеристики
Назва бізнес-процесу	Управління кошиком
Вхідна подія	Додавання товару
Основні учасники	Користувач
Вхідні дані	ID товару, кількість
Вихідні дані	Оновлений кошик
Вихідна подія	Зміна стану кошика
Клієнт	Користувач

Бізнес-процес оформлення замовлення

Цей процес (табл. 1.5) є ключовим для Інтернет-магазину, оскільки саме він завершує взаємодію користувача з системою.

Таблиця 1.5

Характеристика бізнес-процесу оформлення замовлення

Назва характеристики	Значення характеристики
Назва бізнес-процесу	Оформлення замовлення
Вхідна подія	Натискання кнопки «Купити»
Основні учасники	Користувач, система
Вхідні дані	Кошик, баланс
Вихідні дані	Замовлення
Вихідна подія	Замовлення створено
Клієнт	Користувач

У процесі оформлення замовлення система перевіряє баланс користувача, списує кошти та створює запис у базі даних замовлень.

Бізнес-процес роботи з відгуками

Відгуки (табл. 1.6) є важливим елементом зворотного зв'язку та впливають на довіру користувачів до товарів та магазину в цілому.

Таблиця 1.6

Характеристика бізнес-процесу відгуків

Назва характеристики	Значення характеристики
Назва бізнес-процесу	Додавання відгуку
Вхідна подія	Відправка форми
Основні учасники	Користувач

Продовження таблиці 1.6

Вхідні дані	Текст, рейтинг
Вихідні дані	Новий відгук
Вихідна подія	Відгук додано
Клієнт	Користувач

Користувач відкриває форму відгуку, вводить текст, обирає рейтинг, натискає кнопку створення відгуку. Система створює новий запис в базі даних відгуків, відгук з'являється на сторінці товару.

Бізнес-процес управління товарами (лише для адміністратора)

Адміністратор відповідає за підтримку актуальності каталогу (табл. 1.7). Він обирає товар, входить в режим редагування та змінює параметри товару. Або адміністратор може додати новий товар у спеціальній формі. Також передбачена можливість видалення товару. Результат – оновлений каталог товарів.

Таблиця 1.7

Характеристика бізнес-процесу управління товарами

Назва характеристики	Значення характеристики
Назва бізнес-процесу	Управління товарами
Вхідна подія	Дія адміністратора
Основні учасники	Адміністратор
Вхідні дані	Дані товару
Вихідні дані	Оновлений каталог
Вихідна подія	Зміна даних
Клієнт	Система

Бізнес-процес роботи з балансом та промокодами

Цей процес (табл. 1.8) забезпечує фінансову взаємодію користувача із системою. Користувач натискає на кнопку «Баланс», вводить промокод у спеціальному текстовому полі та отримує додаткову валюту. Використані промокоди система заносить до бази даних, щоб запобігти повторному використанню промокодів.

Таблиця 1.8

Характеристика бізнес-процесу поповнення балансу

Назва характеристики	Значення характеристики
Назва бізнес-процесу	Поповнення балансу
Вхідна подія	Введення промокоду
Основні учасники	Користувач
Вхідні дані	Промокод
Вихідні дані	Новий баланс
Вихідна подія	Баланс оновлено
Клієнт	Користувач

Аналіз взаємодії бізнес-процесів

Усі розглянуті бізнес-процеси взаємопов'язані між собою та формують єдину систему. Наприклад, процес перегляду товару переходить у процес додавання до кошика, який, у свою чергу, веде до оформлення замовлення. Аналогічно, процес авторизації є передумовою для більшості інших процесів.

Система побудована за принципом клієнт-серверної взаємодії, де клієнт ініціює запити, а сервер обробляє їх та повертає результати.

Під час аналізу було виявлено ряд проблем:

- відсутність цілодобової підтримки користувачів;
- складність пошуку товарів без інтелектуальних підказок;

- відсутність персоналізованих рекомендацій для клієнтів;
- відсутність автоматизованої обробки типових запитів;
- складність збору та аналізу запитів користувачів;
- відсутність каналу комунікації з клієнтом;

У результаті проведеного аналізу було визначено основні бізнес-процеси Інтернет-магазину. Побудована модель дозволяє описати діяльність системи, враховуючи взаємодію користувачів, адміністраторів та системи. Виявлені проблеми свідчать про необхідність впровадження чат-бота на основі штучного інтелекту, який дозволить автоматизувати процес користувацької підтримки, підвищити якість обслуговування та ефективність функціонування Інтернет-магазину.

1.3. Постановка задачі проєктування

Проведений аналіз бізнес-процесів Інтернет-магазину дозволив виявити ряд проблем, що негативно впливають на ефективність функціонування системи. Існуюча модель організації взаємодії з користувачами має недостатній рівень автоматизації, не може забезпечити цілодобову користувацьку підтримку без зайвого навантаження на персонал.

Саме тому модель бізнес-процесів «ЯК Є» не відповідає вимогам е-комерції, де головними факторами успіху є швидкість обслуговування, доступність та індивідуальний підхід до кожного клієнта. Існує необхідність переходу до нової моделі «ЯК ПОВИННО БУТИ», яка передбачає впровадження інформаційних технологій для автоматизації процесів обробки інформації.

Чат-бот на основі штучного інтелекту є однією з таких технологій, так як він може консультувати користувачів веб-додатку у режимі реального часу без необхідності залучення персоналу у будь-який час доби.

Актуальність теми даної роботи зумовлена необхідністю розширення функціоналу Інтернет-магазину за рахунок впровадження програмних систем на основі штучного інтелекту, що підвищать швидкість та якість

обслуговування, зменшать навантаження на співробітників та забезпечать цілодобову підтримку користувачів. Таке покращення позитивно вплине на конкурентоспроможність магазину та підвищить його позицію на ринку електронної комерції.

Метою даної роботи є розробка інтелектуального чат-бота для Інтернет-магазину, що зможе оптимізувати розглянуті бізнес-процеси шляхом автоматизації обробки запитів користувачів.

Для досягнення мети необхідно вирішити такі задачі:

- провести детальний аналіз предметної області Інтернет-магазину;
- визначити основні бізнес-процеси, що підлягають автоматизації;
- дослідити сучасні підходи до розробки чат-ботів та технології штучного інтелекту;
- сформулювати функціональні та нефункціональні вимоги до системи;
- розробити структуру та архітектуру чат-бота;
- спроектувати алгоритми обробки природної мови користувачів;
- реалізувати механізми інтеграції з базами даних Інтернет-магазину;
- забезпечити можливість пошуку товарів та надання рекомендацій;
- реалізувати обробку типових сценаріїв взаємодії (консультація, оформлення замовлення, отримання інформації);
- провести тестування та оцінку ефективності розробленої системи.

У межах проєктування необхідно автоматизувати ряд задач, пов'язаних зі зберіганням та обробкою інформації, а саме: обробку текстових запитів користувачів, аналіз змісту повідомлень, пошук інформації про товари, формування відповідей у зрозумілому для клієнта вигляді.

Особливу увагу слід приділити забезпеченню персоналізації обслуговування, що передбачає врахування індивідуальних потреб користувачів, їхніх попередніх запитів та поведінки, що підвищить ефективність рекомендацій та ймовірність здійснення покупок клієнтом.

Результатом виконання даної роботи має стати чат-бот, що інтегрується в інформаційну систему Інтернет-магазину. Впровадження такого рішення

дозволить перейти від поточної моделі «ЯК Є» до вдосконаленої моделі «ЯК ПОВИННО БУТИ» з автоматизованою обробкою запитів, безперервною підтримкою користувачів та підвищеним рівнем сервісу. Очікується, що використання чат-бота дозволить скоротити час обробки запитів, зменшити навантаження на персонал, підвищити рівень задоволеності клієнтів та загальну ефективність роботи Інтернет-магазину.

1.4. Висновки до розділу 1

У розділі 1 була розглянута бізнес-система Інтернет-магазину, яка потребує вдосконалення. Було проведено аналіз предметної області та розглянуто актуальні напрями розвитку, наприклад, інтеграцію технологій штучного інтелекту в онлайн-магазин, що автоматизує процес користувацької підтримки та покращить ефективність магазину.

Було проведено аналіз бізнес-процесів системи: визначено основних учасників, вхідні та вихідні дані, та зв'язки між цими процесами. Бізнес-процеси мають недоліки, що впливають на ефективність роботи магазину і які можна усунути інтеграцією чат-бота на основі штучного інтелекту.

На основі аналізу бізнес-процесів було виконано постановку задачі на проєктування. Визначено актуальність та мету роботи, задачі, необхідні для досягнення мети та очікуваний результат.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ІНТЕЛЕКТУАЛЬНОГО ЧАТ-БОТА ДЛЯ АВТОМАТИЗАЦІЇ ВЗАЄМОДІЇ З КОРИСТУВАЧАМИ ІНТЕРНЕТ-МАГАЗИНУ

2.1. Основні вимоги до чат-бота

Після проведеного аналізу предметної області та виявлених проблем функціонування бізнес-процесів сформовано основні вимоги до програмного продукту.

Вимоги до програмного продукту визначають його можливості, обмеження та критерії оцінки якості.

Функціональні вимоги - це перелік дій, які має виконувати програмна система, щоб задовольнити потреби користувачів та бізнес-цілі [8]. Однією з таких функцій є діалог з користувачем у режимі реального часу. Чат-бот має приймати запити від користувача, аналізувати їх та надавати релевантні відповіді. Він повинен надавати у відповідях інформацію про товари Інтернет-магазину, шукати їх за описом, назвою, ключовими словами.

Важливими функціями є надання рекомендацій та консультацій. Система повинна пропонувати найбільш релевантні товари, що можуть задовольнити потреби користувача. Також вона має відповідати на поширені питання щодо авторизації, замовлень, відгуків, налаштування облікового запису, кошика тощо.

Система повинна очищувати історію діалогу за потреби з боку користувача. Так користувач зможе контролювати контекст розмови, видаляти конфіденційні дані (якщо такі дані потрапили до діалогу) та знижувати навантаження на систему. Довгі розмови негативно впливають на швидкодію як бота так і пристрою та знижують точність відповідей. Також кнопка очищення діалогу допомагає розробнику, що може перевіряти сценарії багато разів без їх змішування (контекст одного сценарію не буде накладатися на контекст іншого).

Якщо функціональні вимоги визначають, що має робити програма, то нефункціональні вимоги визначають, наскільки добре вона має це робити [1]. Однією з таких вимог є швидкодія системи. Час відповіді чат-бота на запит користувача має бути мінімальним (менше 3-5 секунд) щоб користувач мав позитивний та продуктивний досвід користування. Також система має бути надійною, щоб мати можливість обробляти багато запитів одночасно без збоїв та втрати продуктивності.

Масштабованість також є однією з таких вимог. Чат-бот має бути готовим до розширення та інтеграції в інші додатки, наприклад, в Telegram у якості бота.

Чат-бот має відповідати вимогам безпеки (враховувати вимоги стандарту ISO/IEC 27001), забезпечувати захист персональних даних користувачів, запобігати несанкціонованому доступу до цих даних та зберігати конфіденційність інформації [18].

Також програмна система має бути гнучкою, щоб у адміністрації була можливість коригувати її поведінку (наприклад, через зміну інструкцій, моделі та інших параметрів) та надавати актуальні дані про товари з бази даних.

Необхідною є і зручність використання. Інтерфейс має бути простим і зрозумілим, а відповіді - корисними для користувача.

Чат-бот має працювати у сучасних браузерах, наприклад Google Chrome, MS Edge, Opera.

Актор – хтось або щось, що взаємодіє із системою через варіанти використання, але не належить до неї. Це може бути людина, пристрій або навіть інша система, що може стати джерелом дії на систему, що проєктується. Правильне визначення акторів допомагає уникнути непорозумінь у вимогах та наочно демонструє їх роль у роботі системи.

Для проєктованого чат-бота було визначено таких акторів:

- Користувач Інтернет-магазину. Користувач може задавати чат-боту питання щодо товарів та роботи з Інтернет-магазином: авторизації,

реєстрації, покупки, кошику, профілю, відгуків тощо. Також користувач може очистити історію діалогу.

- Адміністратор системи. Адміністратор регулює поведінку чат-бота, надає йому найновішу інформацію про товари та контролює роботу системи. Важливий актор, так як від нього залежить швидкість та точність відповідей чат-бота, а це впливає на задоволеність клієнтів та їх бажання користуватись послугами Інтернет-магазину.

Для формалізації вимог була розроблена діаграма варіантів використання (рис.2.1) для визначення функціональних вимог майбутнього додатка. Подана діаграма розмежовує програмну систему і зовнішніх акторів та так само відображає, як вони взаємодіють з нею.

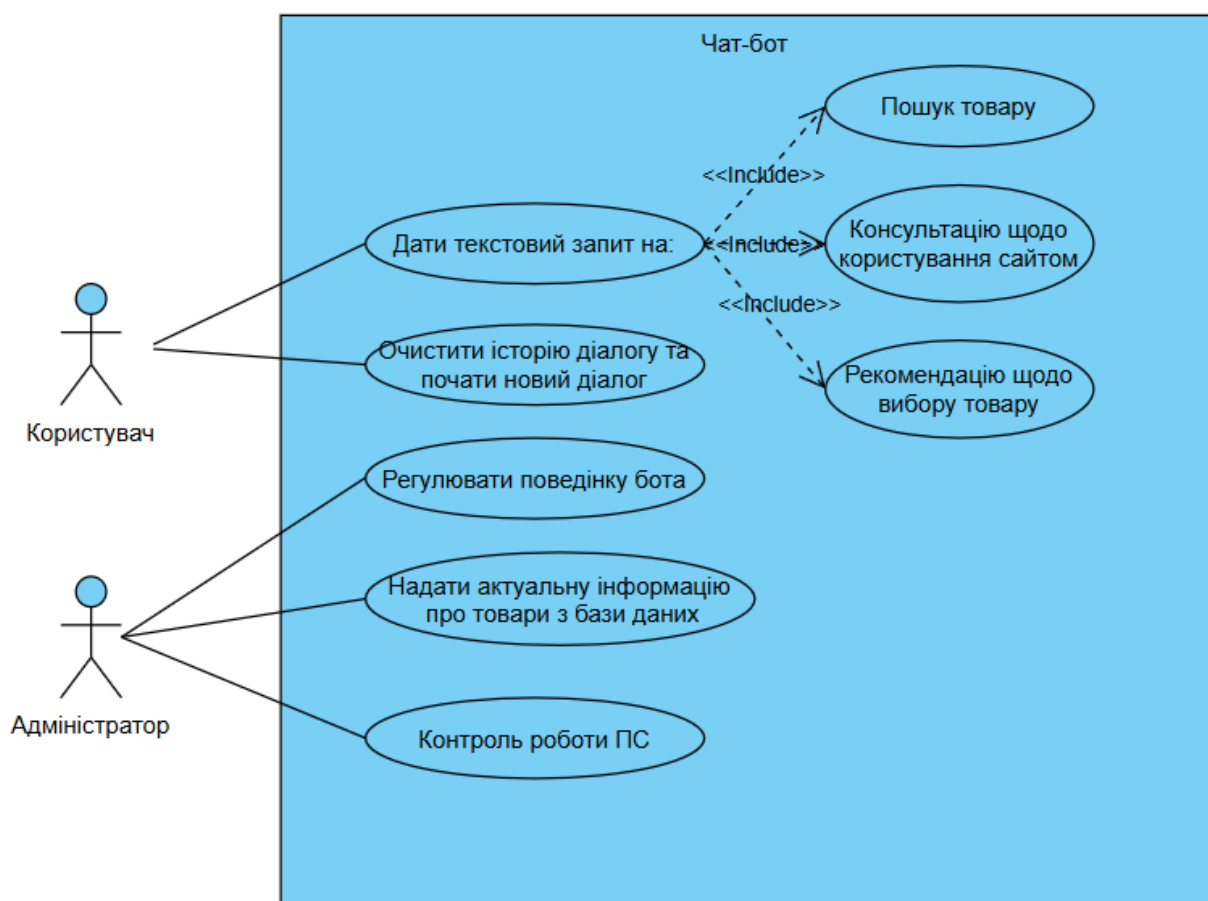


Рисунок 2.1. Діаграма варіантів використання

У користувача є можливість дати запит боту текстом у спеціальному полі віджета чат-бота. Він може задавати питання щодо товару, його опису, рейтингу, відгуків. Також він може запитати чат-бота надати йому рекомендацію, наприклад, кращий інструмент для роботи у копальні або набір для походів. Якщо у користувача виникають труднощі з навігацією по сайту або з процесами авторизації, реєстрації, замовлення, роботи з кошиком та іншими функціями магазину, чат-бот може проконсультувати користувача і дати покрокову інструкцію, що є зрозумілою і простою навіть для людини, що не є обізнаною у сфері ІТ.

Створена діаграма відображає весь необхідний функціонал чат-бота та через які варіанти використання виконується взаємодія з програмною системою.

2.2. Розробка проєктних специфікацій

Розробка проєктних специфікацій є важливим етапом у процесі проєктування та розробки програмного забезпечення. Вони формалізують та розширюють нефункціональні та функціональні вимоги. Ці специфікації також є основою для подальшого проєктування та розробки програмного продукту. Саме на ці специфікації спираються тестувальники у процесі тестування продукту для того, щоб перевірити, наскільки кінцевий продукт відповідає вимогам замовника.

Для чат-бота, що проєктується, було розроблено такі специфікації:

- Функціональні специфікації (таблиця 2.1);
- Нефункціональні специфікації (таблиця 2.2);
- Специфікації інтерфейсу (таблиця 2.3);
- Специфікації взаємодії (таблиця 2.4).

Таблиця 2.1

Функціональні специфікації

№	Назва функції	Опис	Вхідні дані	Вихідні дані
1	Обробка повідомлення користувача	Прийом та первинний аналіз тексту	Текст повідомлення	Структурований запит
2	Визначення наміру	Визначення цілі запиту користувача	Текст повідомлення	Намір користувача
3	Генерація відповіді	Формування відповіді	Намір, дані системи	Текст відповіді
4	Пошук товарів через чат	Пошук товарів за запитом	Ключові слова, опис товару	Список товарів
5	Рекомендація товарів	Персоналізовані рекомендації	Історія, запит	Рекомендовані товари
6	Консультація користувача	Надання допомоги	Запит користувача	Відповідь/пояснення
8	Обробка помилкових запитів	Реакція на запит	Некоректний текст	Повідомлення про некоректний запит з боку користувача
10	Контекст діалогу	Збереження контексту	Попередні повідомлення	Актуальний стан діалогу

Таблиця 2.2

Нефункціональні специфікації

№	Категорія	Опис
1	Продуктивність	Час відповіді не більше 3 секунд або 5 секунд в умовах високого навантаження
2	Точність	Бот коректно визначає намір користувача
3	Масштабованість	Здатність до розширення та інтеграцій
4	Надійність	Безперервна робота без збоїв
5	Безпека	Захист API та даних користувача
6	Конфіденційність	Захист персональних даних
7	Доступність	Підтримка популярних браузерів: Google Chrome, MS Edge, Opera
8	Зручність	Простий текстовий інтерфейс
10	Інтеграція	Взаємодія з API магазину

Таблиця 2.3

Специфікації інтерфейсу

Екран	Елемент	Опис
Чат-інтерфейс	Вікно діалогу	Відображення повідомлень
Поле введення	Введення тексту користувачем	
Кнопка “Надіслати”	Відправка повідомлення	
Індикатор набору	Показ відповіді бота	
Повідомлення бота	Відповіді системи	
Повідомлення користувача	Запити користувача	

Специфікації взаємодії

№	Роль	Дія	Вплив
1	User	Написати повідомлення	Передача запиту боту
2	User	Отримати відповідь	Відображення відповіді
3	User	Пошук товару	Отримання списку товарів за запитом
4	User	Запит рекомендацій	Персоналізовані товари
5	User	Очистити діалог	Повне очищення діалогу
6	User	Уточнення запиту	Оновлення контексту
7	System	Аналіз запиту	Визначення наміру
8	System	Генерація відповіді	Формування відповіді
9	System	Обробка помилки	Альтернативна відповідь

Таким чином, було сформовано повний набір проєктних специфікацій, що визначають: функції чат-бота, якість та швидкість виконання цих функцій, інтерфейс та взаємодії між актором та системою. Отримані специфікації можна використовувати для наступних етапів роботи, а саме: проєктування, розробки та тестування чат-бота.

2.3. Розробка поведінкових діаграм

Нотація у форматі UML є одним з найпопулярніших стандартів розробки діаграм. UML може запропонувати широкий вибір діаграм, що є корисними у різних галузях діяльності людини, не лише у сфері програмної інженерії. UML діаграми використовуються у сферах: HR-менеджменту, бізнес-аналітиці, контролю якості, корпоративних фінансів тощо [2].

Динамічна поведінка системи визначається поведінковими діаграмами. Ці діаграми відображають процеси та взаємодії між частинами системи та з зовнішніми акторами на різних рівнях. Від подачі високорівневих концептів у вигляді діаграми варіантів використання до низькорівневої декомпозиції конкретних процесів у діаграмах послідовності та діяльності [9].

Діаграма діяльності описує потік подій, через які проходить система для досягнення цілі або завершення завдання. Така діаграма може зображувати як систему в цілому, так і окремі бізнес-процеси, що відбуваються в ній.

На рисунку 2.2 зображено діаграму діяльності для діалогу між чат-ботом та користувачем. Користувач вводить повідомлення, бот його отримує та генерує відповідь користувачу. Але якщо запит не є зрозумілим або не належить до предметної області (у контексті цієї роботи, це Інтернет-магазин), то бот повідомить користувача про це.

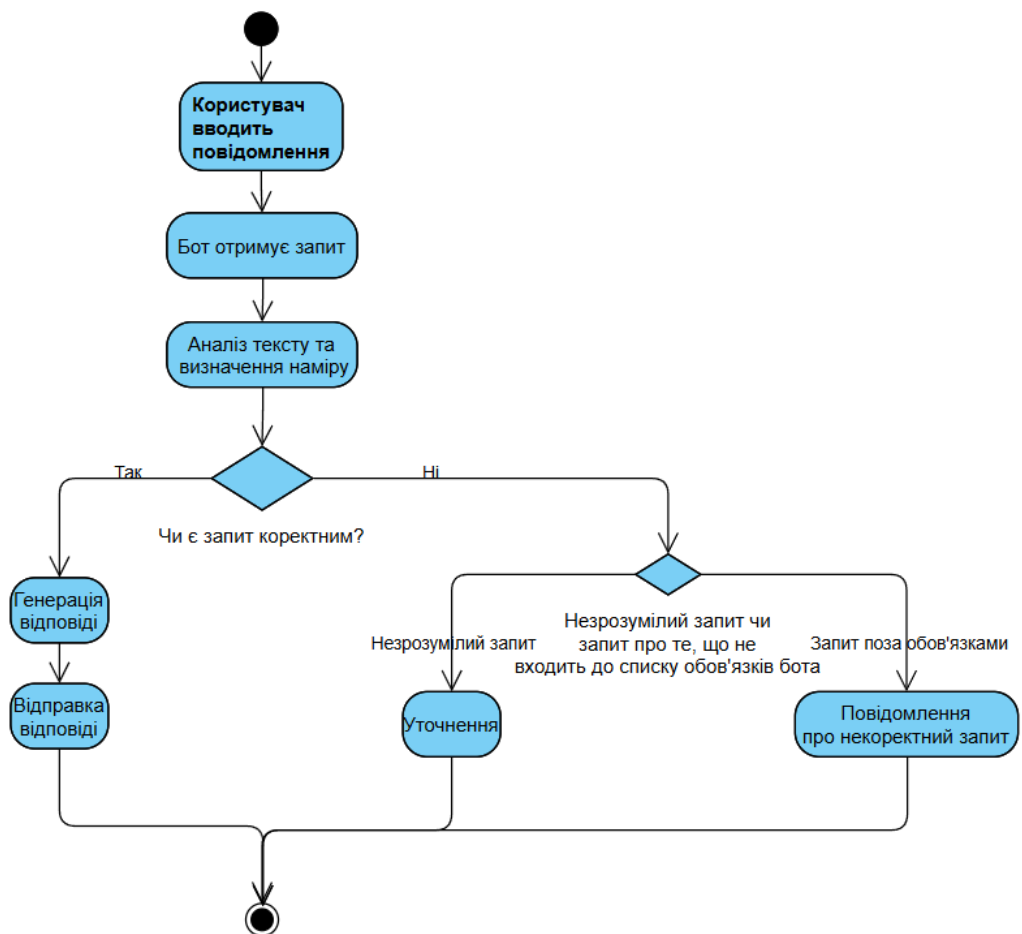


Рисунок 2.2. Діаграма діяльності для процесу діалогу

На рисунку 2.3 зображено процес пошуку конкретного товару за запитом від користувача. Бот розпізнає намір пошуку з повідомлення, формує запит до бази даних товарів. Якщо шуканий товар існує, то бот генерує та надсилає повідомлення про те, що такий товар існує, та дає користувачу інформацію про нього. У іншому випадку бот повідомить користувача про те, що такий товар відсутній.

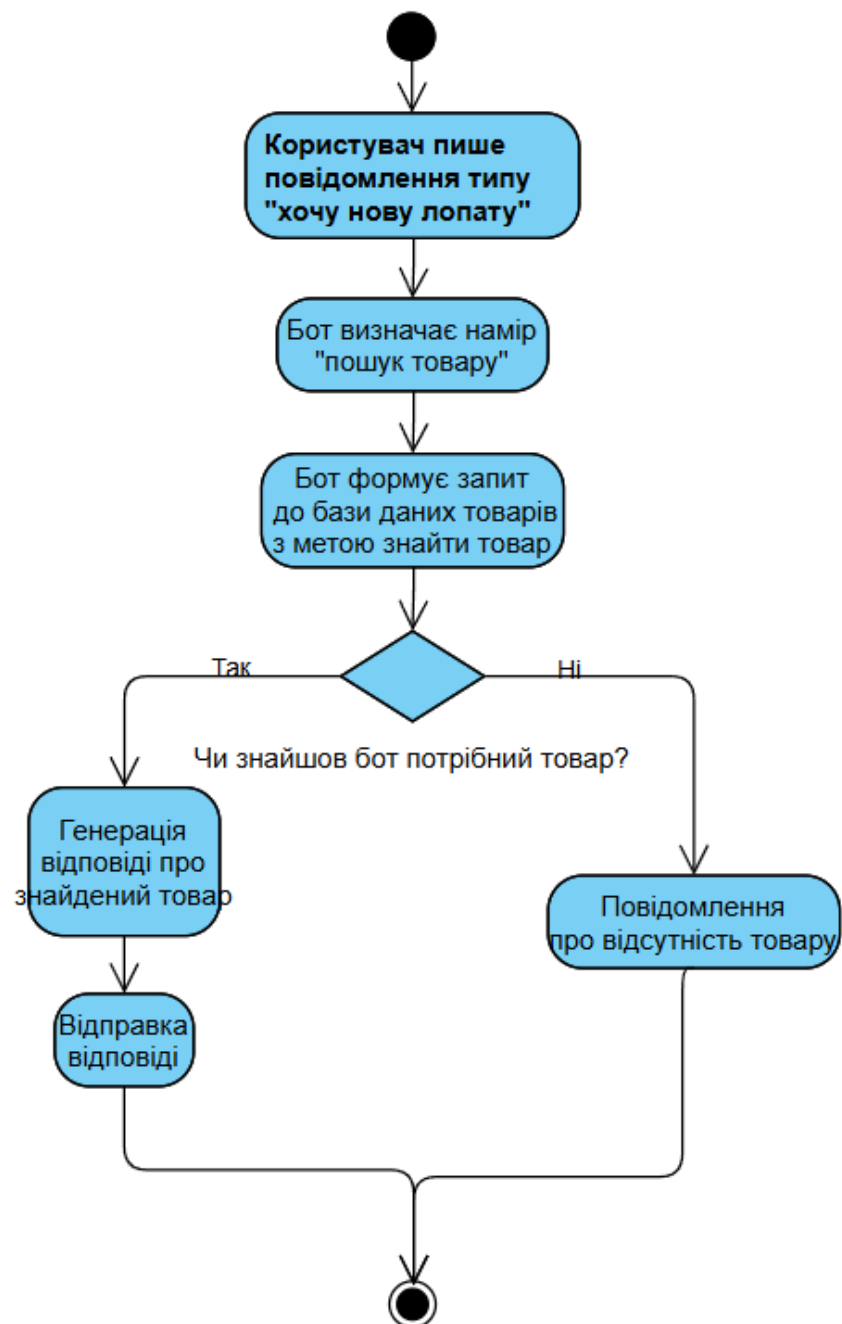


Рисунок 2.3. Діаграма діяльності для пошуку товарів

Діаграми послідовності показують потік інформації, що відбувається між елементами в системі для досягнення мети. Вони моделюють серії подій та потоків інформації, що мають відбутися в межах одного бізнес-процесу. Використовується для того, щоб гарантувати правильну передачу інформації між акторами або сутностями в правильний час.

На рисунку 2.4 зображено процес обробки повідомлення користувача чат-ботом. Користувач вводить текстове повідомлення, яке передається через інтерфейс користувача до контролера чат-бота. Далі повідомлення обробляється за допомогою NLP-модуля, який визначає намір користувача та витягує ключові параметри запиту. Якщо намір розпізнано коректно, бот формує відповідний запит до серверної частини системи для отримання необхідних даних. Після отримання результату формується відповідь, яка передається користувачу. У випадку, якщо намір не було розпізнано, бот генерує повідомлення з проханням уточнити запит або повідомляє про помилку.

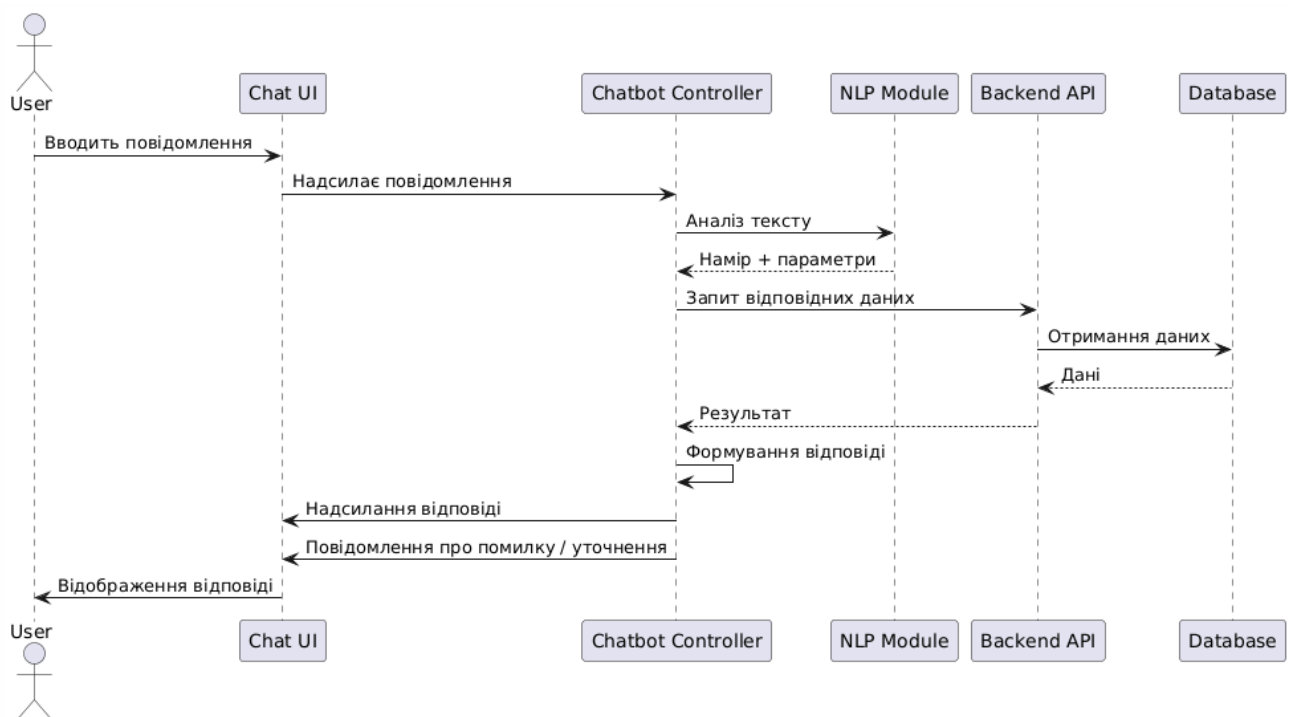


Рисунок 2.4. Діаграма послідовності для обробки повідомлень

На рисунку 2.5 зображено процес надання рекомендацій користувачу чат-ботом. Користувач надсилає повідомлення із запитом на рекомендації, яке передається до чат-бота через інтерфейс. Далі запит обробляється NLP-модулем, який визначає намір користувача та виділяє ключові слова. На основі отриманих параметрів бот формує критерії пошуку та звертається до бази даних товарів через серверну частину системи. Якщо відповідні товари знайдено, бот формує список рекомендацій та надсилає його користувачу. У випадку відсутності відповідних товарів бот повідомляє користувача про неможливість сформулювати рекомендації за заданим запитом.

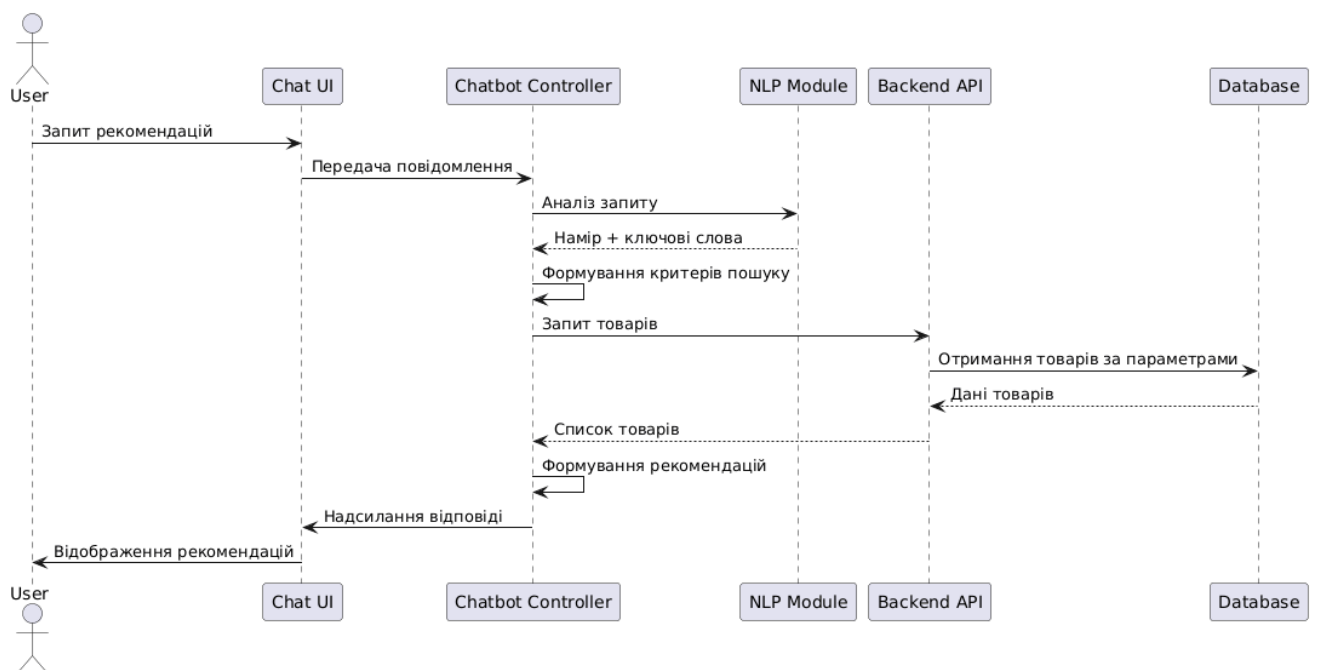


Рисунок 2.5. Діаграма послідовності для формування рекомендацій

У цьому підрозділі було визначено та наведено UML-діаграми, що відображають поведінкові особливості проєктованого чат-бота. Розроблено діаграми діяльності та діаграми послідовності. Створені діаграми наочно відображають динамічні елементи системи та моделюють бізнес-процеси, які необхідно розробити.

2.4. Розробка діаграм потоків даних

Діаграми потоків даних відображають потоки інформації в межах одного бізнес-процесу або системи. Ці діаграми використовуються для огляду системи на різних рівнях. Діаграма 0 рівня відображає всю систему як 1 процес та показує її взаємодію із зовнішніми сутностями. Діаграма 1 рівня розбиває систему на ключові бізнес-процеси, відображає внутрішні потоки даних та сховища даних. Діаграма 2 рівня розбиває конкретні бізнес-процеси для більш детального аналізу. Особливість цього типу діаграм полягає в тому, що декомпозицію можна проводити до моменту, коли не залишиться процесів, які можна розбити [30].

На рисунку 2.6 зображено діаграму потоків даних нульового рівня (контекстна діаграма). Для обробки повідомлення користувача чат-бот отримує текстовий запит від користувача. Користувач надсилає повідомлення, яке передається до чат-бота для подальшої обробки. Чат-бот аналізує отриманий запит та формує запит до бази даних товарів для отримання інформації про товари. База даних потім передає дані про товари, після чого чат-бот формує остаточну відповідь та надсилає її користувачу.

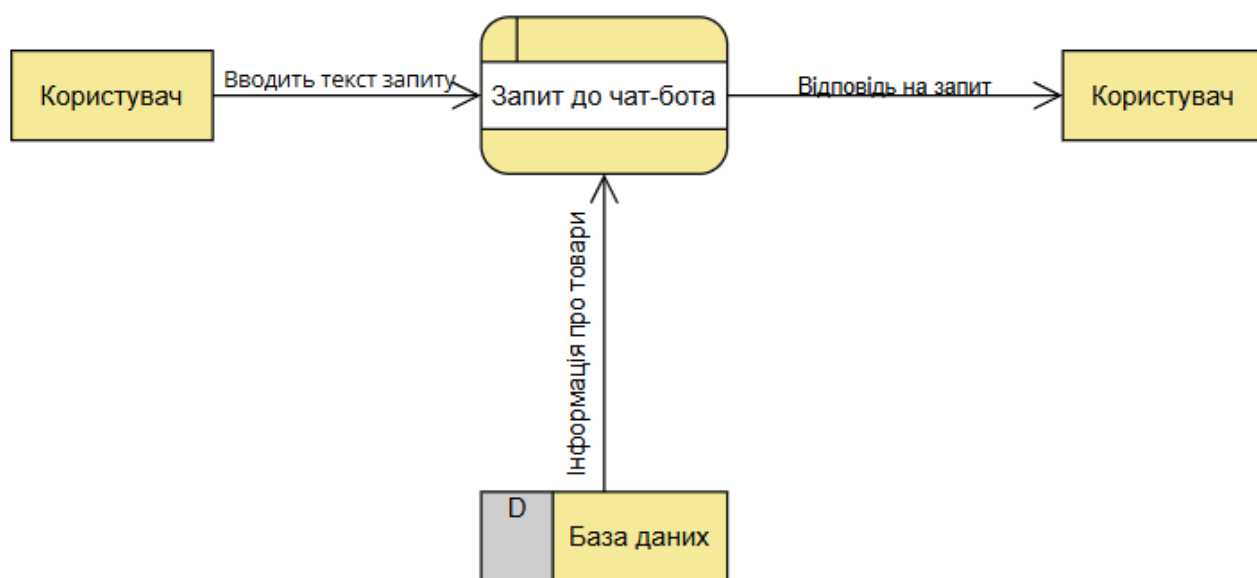


Рисунок 2.6. Діаграма потоків даних 0 рівня

На рисунку 2.7 зображено діаграму потоків даних першого рівня, що проводить декомпозицію запитів до чат-бота. Для обробки повідомлення користувача система виконує декілька послідовних процесів. Користувач надсилає повідомлення, яке надходить до процесу обробки повідомлення. Далі текст передається до модуля NLP-аналізу, де визначається намір користувача та виділяються ключові параметри запиту. Після цього отримані дані передаються до процесу формування відповіді. Якщо для формування відповіді необхідна інформація про товари, система надсилає запит до бази даних товарів. База даних повертає відповідні дані, на основі яких формується остаточна відповідь. Після цього чат-бот надсилає сформовану відповідь користувачу.

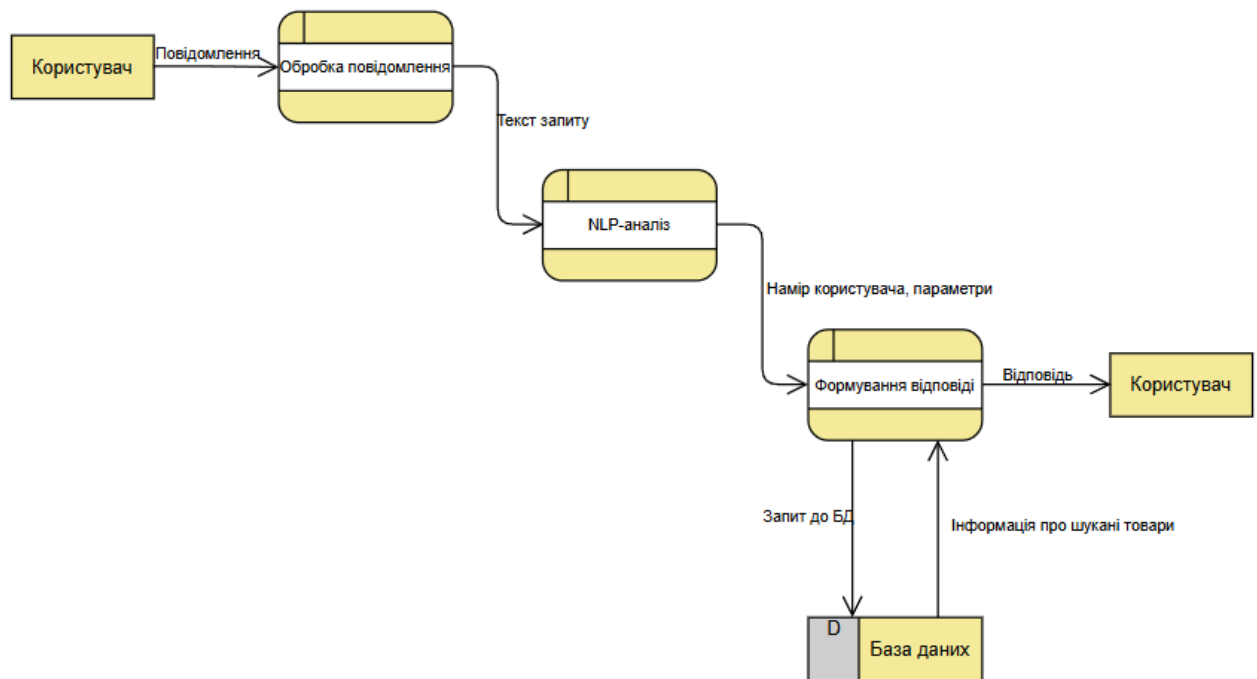


Рисунок 2.7. Діаграма потоків даних 1 рівня

У цьому підрозділі були наведені моделі процесів у нотації DFD, що відповідають моделі «ЯК ПОВИННО БУТИ». Модель бізнес-процесу було зображено у вигляді пари IDEF0-DFD1, де діаграма нульового рівня відображає бізнес процес, а діаграма першого рівня – розбиває його на більш дрібні процеси.

2.5. Проєктування архітектури чат-бота

На основі отриманих раніше моделей та проєктних специфікацій було сформовано абстракції архітектурного рівня, визначено архітектурні шари та виконано декомпозицію чат-бота на компоненти та програмні модулі. Прийняті під час проєктування рішення відображено у нотації ArchiMate (додаток Г).

ArchiMate – незалежна та відкрита мова для опису архітектури на різних рівнях. Є технічним стандартом The Open Group та заснована на стандарті IEEE 1471 (стандарт опису архітектури програмного забезпечення). ArchiMate надає всі необхідні інструменти для опису, аналізу та візуалізації зв'язків між бізнес-доменами у зручному та однозначному вигляді [22].

Користувач взаємодіє з системою через бізнес-процеси. Обробка бізнес-процесів виконується програмними модулями. Спочатку текстовий запит передається з інтерфейсу в контролер, що координує обробку запитів. Для аналізу тексту використовується модуль обробки природної мови. Він визначає наміри користувача та виділяє параметри для пошуку товарів. Ці дані передаються контролеру і він з отриманими параметрами звертається до модуля доступу до бази даних, якщо є потреба знайти товари. Потім контролер отримує потрібну інформацію та надає відповідь користувачу через інтерфейс. Інфраструктурний рівень відповідає за роботу системи в середовищі виконання. Клієнтською частиною виступає веб-браузер. Сервер відповідає за виконання бізнес-логіки. Природню мову обробляє модель штучного інтелекту. А дані про товари зберігаються у базі даних.

У цьому підрозділі було спроектовано крос-шарову архітектуру чат-бота у нотації ArchiMate. Було визначено шари архітектури, модулі та залежності між компонентами в системі, оцінено складність програмного проєкту. Розроблена архітектура відповідає проєктним специфікаціям.

2.6. Розробка структурних UML-діаграм

Структурні UML-діаграми відображають статичну архітектуру системи, елементи, що утворюють її структуру та зв'язки між ними. Вони фокусуються

на організацію та зв'язки між такими елементами: класи, інтерфейси, вузли тощо. Завдяки цьому структурні діаграми допомагають команді розробників зрозуміти побудову системи, її підтримку та масштабування.

Діаграма компонентів є видом структурних діаграм і відображає компоненти системи та як вони пов'язані одне з одним. Має багато застосувань у сфері розробки ПЗ, наприклад, аналіз вимог, програмна документація, генерація коду, підтримка та розширення готового програмного застосунку [11].

Переваги діаграм компонентів: візуалізація системної архітектури; модульність; покращення комунікації між менеджерами проєкту, проєктувальниками, розробниками та тестувальниками; масштабованість; застосування ключових принципів проєктування: інкапсуляція та зв'язування. На рисунку 2.9 зображена діаграма компонентів.

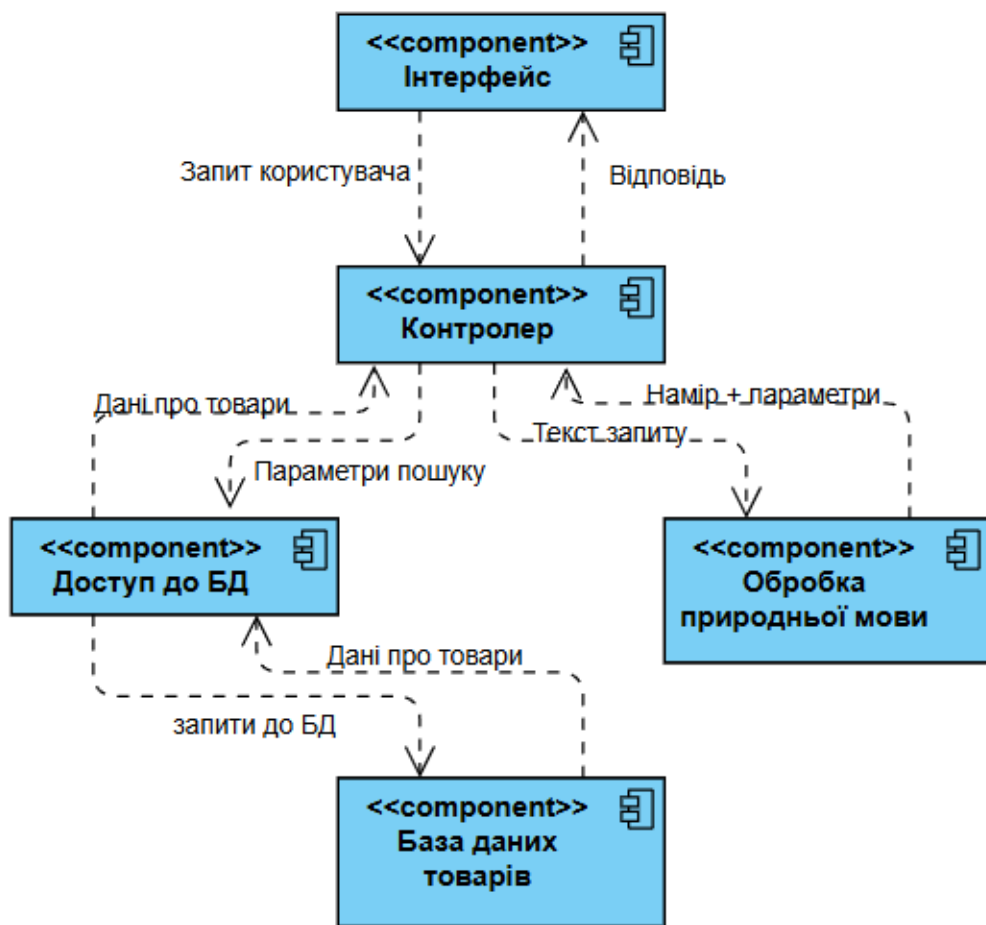


Рисунок 2.8. Діаграма компонентів

Діаграма розгортання показує конфігурацію вузлів та компоненти цих вузлів. Використовується для моделювання фізичних аспектів об'єктно-орієнтованої системи. Їх використовують для відображення статичного вигляду розгортання системи (топології апаратного забезпечення). Вони моделюють апаратне забезпечення для реалізації програмної системи та зв'язки між різними елементами цього забезпечення. Такі діаграми є корисними для документування розгортання програмних компонентів [29]. На рисунку 2.10 зображено діаграму розгортання.

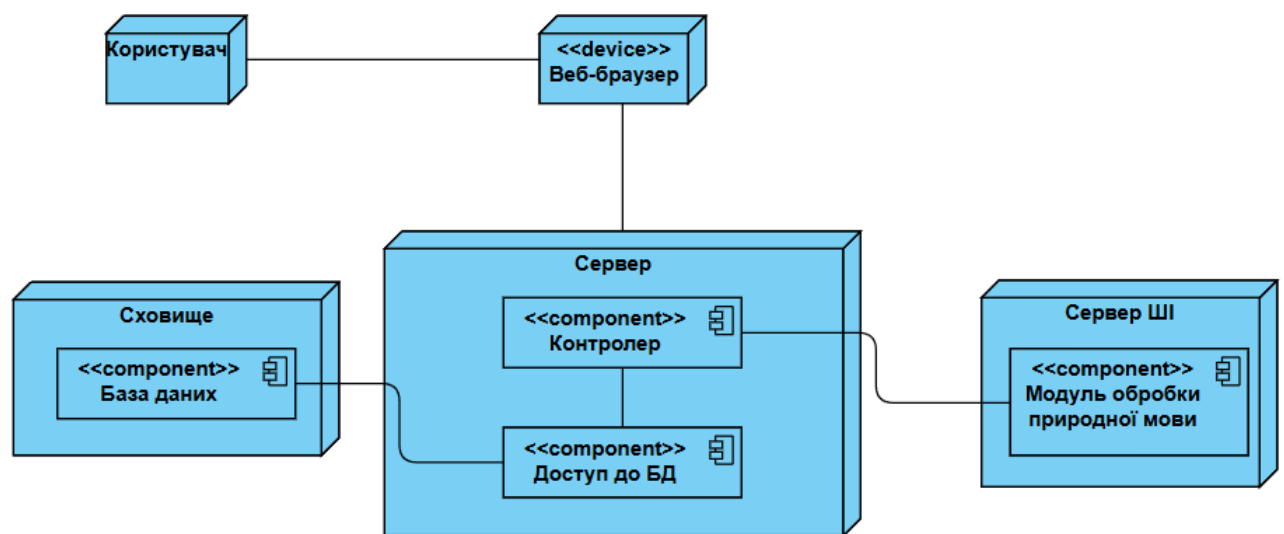


Рисунок 2.9. Діаграма розгортання

У цьому підрозділі було розроблено структурні діаграми у нотації UML. Діаграма компонентів відображає структурні зв'язки між компонентами програмного забезпечення, а діаграма розгортання відображає апаратне забезпечення необхідне для функціонування системи. Розроблені діаграми дають детальне уявлення про внутрішню організацію проєктованого чат-бота.

2.7. Висновки до розділу 2

У розділі 2 було проведено проєктування інтелектуального чат-бота для онлайн-магазину. Було визначено функціональні та нефункціональні вимоги,

визначено ключових акторів та наведено діаграму варіантів використання. Потім було розроблено проєктні специфікації для формалізації вимог. На основі вимог та специфікацій було розроблено UML-діаграми та архітектуру програмного продукту у нотації ArchiMate. Створені діаграми дають уявлення про ключові аспекти проєктованого додатку: динамічну поведінку системи, потік даних в межах цієї системи, статичну архітектуру та архітектурні шари системи. Отримані результати проєктування можна використовувати для подальшої розробки чат-бота.

РОЗДІЛ 3. РОЗРОБКА ІНТЕЛЕКТУАЛЬНОГО ЧАТ-БОТА ДЛЯ АВТОМАТИЗАЦІЇ ВЗАЄМОДІЇ З КОРИСТУВАЧАМИ ІНТЕРНЕТ-МАГАЗИНУ

3.1. Обґрунтування вибору інструментальних засобів розробки

Для розробки чат-бота Інтернет-магазину було розглянуто такі інструментальні засоби:

Node.js – безкоштовна платформа з відкритим кодом для розробки веб-додатків, скриптів та навіть веб-серверів. Node.js дав можливість виконувати JavaScript-скрипти на сервері та повертати результат користувачеві. Раніше такої можливості не було і скрипти виконувалися саме у браузері користувача. Працює на рушії JavaScript Google V8. Ключова перевага Node.js – асинхронний ввід/вивід, що покращує комунікацію та масштабованість веб-додатків, що проводять багато операцій типу «ввід/вивід». Наприклад, програми для комунікації в реальному часі та браузерні ігри. Також ця платформа має менеджер пакетів (npm, node package manager), що дозволяє розробнику швидко та зручно розширити можливості свого веб-застосунку [19].

NestJS є одним із таких пакетів. Це фреймворк на основі TypeScript, що поєднує елементи об'єктно-орієнтованого програмування, функціонального програмування та функціонального реактивного програмування. Nest підтримує інші потужні фреймворки, такі як Express та Fastify. Це дає розробнику свободу вибору інструментів для розробки веб-додатків [12].

Express це мінімалістичний фреймворк, що орієнтований на гнучкість та простоту без негативного впливу на можливості Node.js. Має набір інструментів, що дозволяють швидко та легко створити API. За потреби розробник може розширити цей набір шляхом встановлення додаткових middleware-модулів [13].

Fastify також є легким фреймворком з підтримкою JSON Schema, можливістю обробляти до 30 тисяч запитів за секунду, логуванням та

підтримкою TypeScript. За допомогою плагінів можна розширити базовий функціонал [14].

GPT – сімейство великих мовних моделей, створених компанією OpenAI. Ці моделі здатні розпізнавати природну мову, обробляти її та надавати відповіді, генерувати текст, код. Підтримують більшість мов, включаючи українську. Є можливість гнучко налаштувати моделі під різні сценарії та інтегрувати їх в мобільні та веб-додатки. Ключова перевага – широкий вибір моделей – від дешевих до дорогих [20]. Із недоліків – можливі «галюцинації» – вигадкування інформації [17].

Claude – мовна модель, що розроблена компанією Anthropic. У цій моделі акцент розміщено на безпечності. У порівнянні з іншими моделями, Claude рідше вигадує інформацію та надає більш точні відповіді. Але функціонал в неї обмежений, а сама модель менш гнучка, особливо у творчих задачах [10].

Gemini – сімейство моделей від Google, що відрізняється від інших моделей інтеграцією з екосистемою Google та оптимізацією для досягнення продуктивності та швидкості. Швидкість має свою ціну – менш стабільна якість відповідей та не дуже розвинений контроль генерації [16].

React – фреймворк для розробки інтерфейсів нативних та веб-додатків. Інтерфейс можна створити за допомогою окремих React-компонентів. Використовує розширений синтаксис Javascript – JSX. React можна інтегрувати в існуючий веб-додаток без необхідності його переписувати з нуля. React автоматично оновлює елементи інтерфейсу у відповідь на дію користувача без необхідності перезапускати сторінку або залучати спеціальний скрипт [21].

Іншим фреймворком для розробки користувацьких інтерфейсів є Vue. Створюється поверх HTML, CSS та JavaScript і використовує компоненти, що дають можливість створювати інтерфейси будь-якої складності. Також Vue відстежує зміни станів JavaScript та оновлює DOM у разі змін. Однією з переваг Vue є гнучкість – його можна використати для різних задач: покращити статичний HTML, вбудувати Vue-елементи на існуючі сторінки та для створення додатків на різних платформах [26].

Angular – фреймворк, що підтримується компанією Google. Головна перевага Angular - широкий вибір інструментів, що спрощують процес розробки та задають чітку структуру проєкту. Є інструменти для створення інтерфейсу, API для роботи з даними та вбудовані бібліотеки, що відповідають за маршрутизацію, форми та HTTP-запити. Масштабованість є ще одною перевагою цього фреймворку. Можна проводити масштабування по ходу росту проєкту та команди розробників [27].

Після розгляду вищевказаних інструментів було остаточно вибрано засоби розробки чат-бота для Інтернет-магазину.

Для бекенд-частини обрано Node.js з фреймворком Express. Чат-бот буде впроваджено в існуючий Інтернет-магазин, тому важливо, щоб інтеграція була плавною та не порушувала його роботу. Express є кращим кандидатом, так як він не впливає на створену раніше бізнес-логіку.

Для фронтенд-частини React є кращим вибором завдяки можливості оновлювати зміст сторінки у відповідь на дії користувача. Користувач може вести діалог з чат-ботом в реальному часі без необхідності перезавантажувати сторінку щоб отримати відповідь або поставити інше запитання.

У якості мовної моделі на основі штучного інтелекту було обрано GPT-4.1 nano. Хоча ця модель має гірші здібності, ніж моделі п'ятої серії, вона може виконувати всі функції, визначені вимогами, а саме: пошук товарів, надання рекомендацій та консультація щодо сайту. Головною перевагою є ціна - GPT-4.1 nano: \$0.80 за мільйон токенів проти \$22.50 за мільйон токенів моделі GPT-5.4.

3.2. Розробка чат-бота

Перед розробкою чат-бота необхідно розробити UML-діаграму класів. Ця діаграма точно зображує структуру чат-бота: сутності, їх атрибути, функції та зв'язки з іншими сутностями [5]. Діаграму класів наведено в Додатку В.

Сутність ChatWidget буде `.jsx` компонентом, що відповідатиме за відображення вікна чат-бота користувачу. Також цей віджет буде приймати текстові запити та надавати відповіді на них.

Server буде містити ендпоінти для того, щоб передавати відповіді у віджет або запити до AIService. Також сервер має ендпоінти для доступу до баз даних (така можливість була присутня у веб-додатку Інтернет-магазину).

AIService відповідає за підключення API моделі штучного інтелекту. Генерація відповіді, аналіз тексту запиту, передача інструкції до ШІ-моделі – всі ці функції будуть виконуватись саме цим модулем.

DatabaseService буде проміжною сутністю між бізнес-логікою та базами даних. Сюди входять запити на отримання списків товарів та відгуків.

Item та Review є сутностями, що представляють товари та відгуки. Вони мають свої атрибути, які потім будуть відображатися користувачу у вікні чат-бота. Якщо користувач захоче знайти певний товар, чат-бот покаже йому цей товар та ці атрибути – назва, ціна, опис, категорія тощо.

Діаграма класів надала уявлення щодо структури чат-бота та функцій, які треба розробити.

Інтеграція з API штучного інтелекту була першим етапом розробки. Для спрощення підтримки та розширення чат-бота цю логіку було винесено у окремий модуль `aiService.js`. Функція `generateReply(message, items, reviews)` надає моделі штучного інтелекту запит користувача, системні інструкції (роль моделі у магазині та правила спілкування) та дані про товари та відгуки. Відповідь на текстовий запит користувача є результатом цієї функції.

Файл `server.js` є ядром системи та відповідає за обробку HTTP-запитів. Сюди було додано ендпоінт `POST /api/chat`, що приймає запит користувача, викликає функцію `generateReply()` та повертає відповідь. Ендпоінт `GET /api/test-ai` було реалізовано для перевірки підключення до ШІ-моделі у консолі без використання користувацького інтерфейсу.

Інтерфейс користувача та виклик ендпоінта `POST /api/chat` реалізовано у компоненті `ChatWidget.jsx`. Компонент відповідає за вікно чат-бота,

автопрокрутку повідомлень, індикатор генерації відповіді та кнопку, що дає можливість очистити історію діалогу.

У процесі розробки виникали помилки пов'язані з CORS, взаємодією з базами даних та обробкою помилок. Вони були вирішені правильним налаштуванням CORS та виправленням логіки доступу до даних.

У результаті створено чат-бот, вбудований в Інтернет-магазин, здатний допомагати користувачу в режимі реального часу. Роботу чат-бота зображено на рисунку 3.2. Екранні форми розробленого додатку наведено в додатку А.

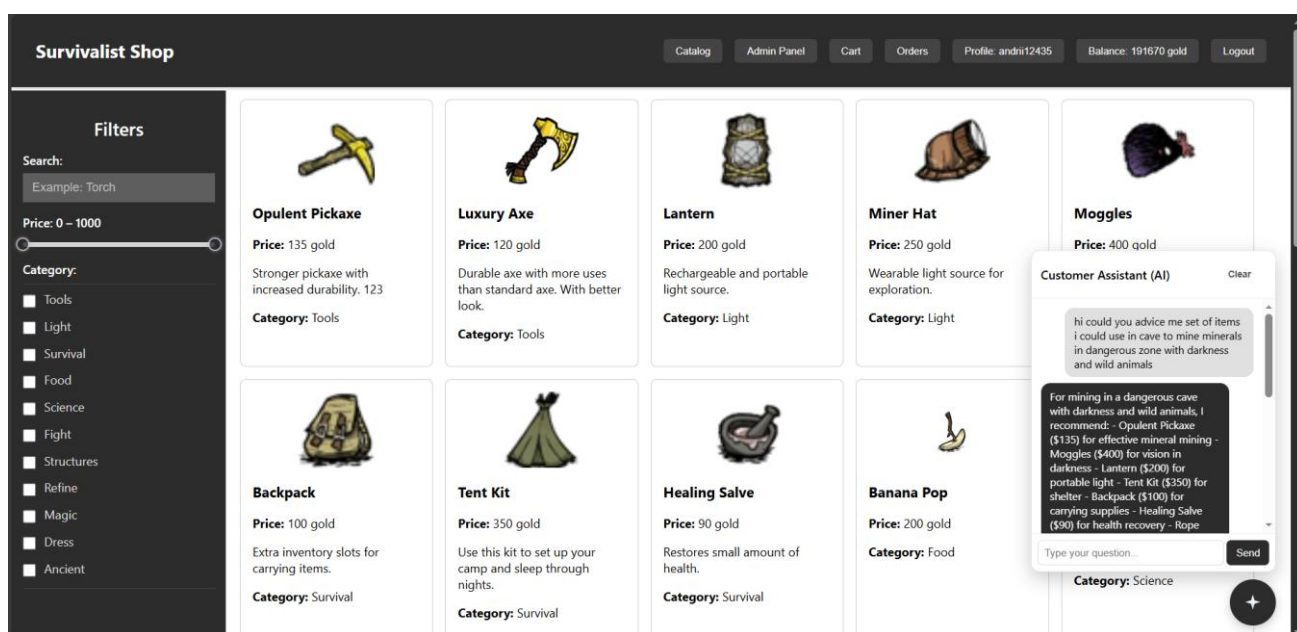


Рисунок 3.1. Робота чат-бота Інтернет-магазину

3.3. Висновки до розділу 3

У розділі 3 було проведено розробку інтелектуального чат-бота для онлайн-магазину. Було розглянуто інструментальні засоби, що можуть виконати поставлену задачу. Node.js з фреймворком Express обрано для бекенд-частини, React було обрано для фронтенд-частини, а у якості мовної моделі кращим вибором стала модель GPT-4.1 nano. Ці інструменти було використано у процесі розробки. Цей процес включав у себе розробку модуля aiService.js, створення нових ендпоінтів у серверному файлі server.js та розробку компонента ChatWidget.jsx. Також було проведено покращення якості

відповідей за допомогою виправлень запитів до баз даних та інструкцій для моделі штучного інтелекту. Результат – чат-бот на основі штучного інтелекту, що допомагає користувачам Інтернет-магазину.

РОЗДІЛ 4. ТЕСТУВАННЯ ТА ВПРОВАДЖЕННЯ ІНТЕЛЕКТУАЛЬНОГО ЧАТ-БОТА ДЛЯ АВТОМАТИЗАЦІЇ ВЗАЄМОДІЇ З КОРИСТУВАЧАМИ ІНТЕРНЕТ-МАГАЗИНУ

4.1. Тестування чат-бота

Тестування програмного забезпечення проводиться з метою виявлення дефектів, помилок, недоліків перед випуском його на ринок або в експлуатацію. Також тестування проводиться, щоб дізнатися, чи відповідає розроблений продукт функціональним та нефункціональним вимогам та специфікаціям. Тестування забезпечує високу якість і надійність програмної системи.

Для опису всього обсягу робіт з тестування використовується тест-план. Він відповідає за: врегулювання процесів тестування, пріоритезацію завдань, планування ресурсів та облік програмного забезпечення та людських ресурсів [7].

На підставі сукупності вимог, проєктних специфікацій та архітектури чат-бота було розроблено тест-план:

1. Об'єкт тестування

Об'єктом тестування є чат-бот інтегрований у веб-додаток Інтернет-магазину для автоматизації користувацької підтримки.

Основні функціональні блоки системи:

- користувацький інтерфейс;
- серверна частина для обробки HTTP-запитів;
- модуль взаємодії зі штучним інтелектом;

Тестування проводиться на персональному комп'ютері з операційною системою Windows 10 та браузерами Google Chrome та Microsoft Edge.

2. Стратегія тестування

Тестування проводиться з використанням декількох підходів, а саме:

- модульне тестування;
- тестування інтерфейсу користувача;
- інтеграційне тестування;

- системне тестування.

3. Етапи тестування

Тестування можна поділити на 3 етапи:

- підготовка (налаштування тестового середовища, формування тест-сценаріїв);
- тестування (перевірка можливостей системи та їх відповідності до вимог);
- аналіз результатів;

4. Критерії початку тестування

Тестування можна розпочинати, якщо виконані такі умови: серверна частина працює і обробляє запити користувача, інтерфейс доступний у браузері, реалізовано основний функціонал та підготовлено тестові сценарії

5. Критерії завершення тестування

Тестування можна вважати завершеним, якщо: відсутні критичні помилки, тестові сценарії виконані, програмна система відповідає вимогам, протягом місяця не виникають нові критичні помилки.

6. Тестове середовище

Тестування проводиться у такому середовищі:

- операційна система Windows 10;
- Node.js сервер;
- MySQL база даних;
- веб-браузери Google Chrome, Microsoft Edge;
- локальний сервер у якості мережевого середовища.

7. Засоби тестування

Для тестування застосовуються інструменти розробника DevTools та Postman.

8. Ризики

Під час тестування можуть виникнути такі ризики: нестабільна робота сервісу, затримки відповідей, помилки при обробці користувацьких запитів. Ці ризики можна уникнути за допомогою резервних відповідей, обмежень обсягу контексту та обробки помилок на сервері.

У результаті розроблений тест-план дозволяє системно перевірити працездатність програмного додатку та підтвердити його відповідність визначеним вимогам. Застосування описаної стратегії тестування забезпечує достатній рівень якості системи перед її використанням.

Модульне тестування полягає у окремому тестуванні окремих модулів програмної системи. Таке тестування дозволяє перевірити, чи поведуться модулі належним чином без залежностей від інших компонентів системи. Зовнішні ресурси імітуються, щоб тестувалася саме логіка конкретного модуля [6].

Відповідно до архітектури розробленого додатку тестування проводиться для:

- модуля взаємодії зі штучним інтелектом (aiService.js);
- серверного модуля обробки запитів (server.js);
- клієнтського компоненту інтерфейсу (ChatWidget.jsx).

Для перевірки серверної частини використано ручні HTTP-запити, аналіз логів сервера та тестові виклики функцій. Для клієнтської частини перевірялася поведінка інтерфейсу при різних сценаріях. У таблиці 4.1 наведено тестові сценарії, за якими проводилося модульне тестування.

Таблиця 4.1

Тестові сценарії

№	Модуль	Опис тесту	Вхідні дані	Очікуваний результат	Фактичний результат	Статус
1	aiService.js	Генерація відповіді на стандартний запит	message: "Який товар найкращий?", items, reviews	Згенерована змістовна відповідь	Отримано коректну відповідь	Пройдено
2	aiService.js	Обробка пустого запиту	message: ""	Повідомлення про некоректний запит	Отримано відповідь без збою	Пройдено

Продовження таблиці 4.1

3	server.js (/api/chat)	Обробка стандартного запиту	JSON з текстом повідомлення	Повернення JSON з відповіддю	Відповідь отримано	Пройдено
4	server.js (/api/chat)	Обробка порожнього запиту	message: ""	Повідомлення про помилку	Отримано помилку	Пройдено
5	ChatWidget.js x	Відправка повідомлення	введений текст	Повідомлення відображається у чаті	Відображається коректно	Пройдено
6	ChatWidget.js x	Відображення відповіді бота	відповідь сервера	Повідомлення бота у чаті	Відображається	Пройдено
7	ChatWidget.js x	Індикатор генерації	запит користувача	Відображення індикатора	Працює коректно	Пройдено
8	ChatWidget.js x	Очищення історії чату	натискання кнопки	Чат очищується	Працює коректно	Пройдено

Спочатку було протестовано модуль aiService.js, а саме функцію generateReply(message, items, reviews). Для цього треба створити файл test.js, що буде використовувати цю функцію з тестовими даними (лістинг 4.1).

Лістинг 4.1 – Код модульного тесту aiService.js:

```
const { generateReply } = require("../aiService");
async function test() {
  const message = "Який товар найкращий?";
  const items = [
    { name: "Ліхтар", price: 200, description: "Яскравий ліхтар" },
    { name: "Ніж", price: 150, description: "Туристичний ніж" },
  ];
  const reviews = [
    { username: "Andrii", rating: 5, review_text: "Дуже хороший товар" },
    { username: "Ivan", rating: 4, review_text: "Нормально" },
  ];
  const reply = await generateReply(message, items, reviews);

  console.log("AI reply:", reply);
}
test();
```

За допомогою Node цей код треба запустити в командному рядку. Тестовий код використає функцію з підставними даними і поверне результат. На рисунку 4.1 зображено процес тестування модуля роботи зі штучним інтелектом.

```
D:\xampp\htdocs\ds-shop>node server/test.js
[dotenv@17.3.1] injecting env (1) from server\.env -- tip: run anywhere with `dotenvx run --yourcommand`
AI reply: Ви можете переглянути або придбати Ліхтар за 200 золота, який є яскравим і корисним для освітлення. Також є Ніж за 150 золота, ідеальний для туристичних поїздок. Якщо потрібно, могу допомогти з додаванням товарів у кошик або оформленням замовлення.

D:\xampp\htdocs\ds-shop>node server/test.js
[dotenv@17.3.1] injecting env (1) from server\.env -- tip: auth for agents: https://vestauth.com
AI reply: Обидва товари мають позитивні відгуки. Ліхтар яскравий і хороший для освітлення, а ніж зручний для туристичних цілей. Вибір залежить від ваших потреб.

D:\xampp\htdocs\ds-shop>
```

Рисунок 4.1. Тестування generateReply(message, items, reviews) з пустим та стандартним запитом

Тестування показало, що модуль aiService.js працює належним чином та правильно обробляє тестові дані.

Далі було протестовано ендпоінт api/chat серверної частини (server.js). Для цього використано Postman, так як цей інструмент дозволяє тестувати API прямо у вікні браузера та надає дані про результати тестування у зручному форматі. На рисунках 4.2 та 4.3 зображено тестування серверної частини.

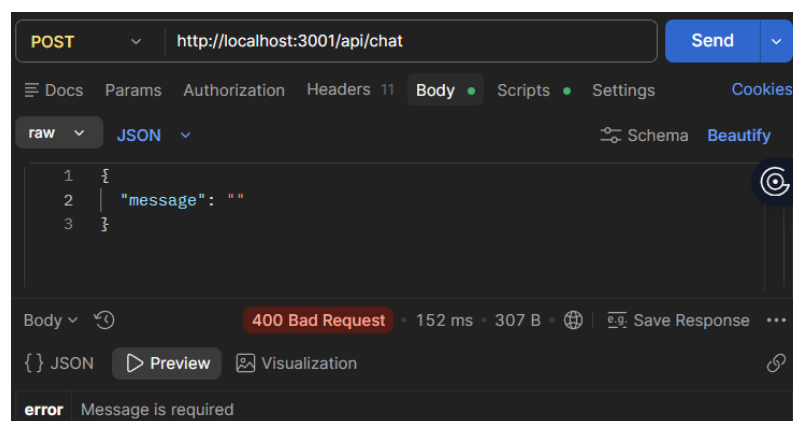


Рисунок 4.2. Тестування api/chat з пустим запитом

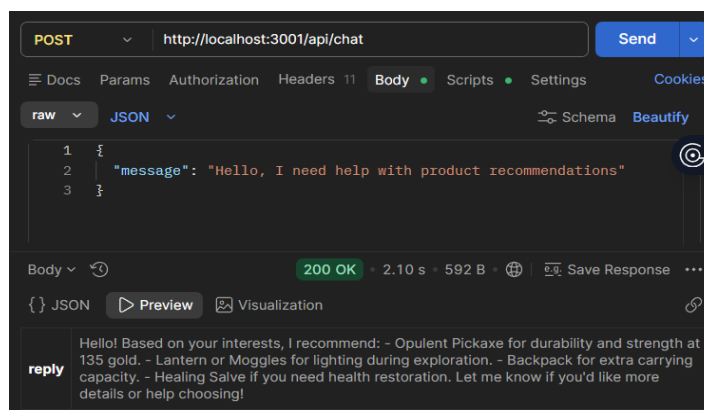


Рисунок 4.3. Тестування api/chat з стандартним запитом

Тестування показало, що звичайні повідомлення користувача обробляються належним чином. Але у випадку відправки пустого повідомлення отримано повідомлення про відсутність тексту, що свідчить про наявність механізму, що запобігає надсиланню пустих повідомлень.

Модуль інтерфейсу (компонент ChatWidget.jsx) було протестовано шляхом перевірки його відповідності до вимог та специфікацій інтерфейсу. Вікно має складатися з: секції діалогу, поле введення, кнопка «надіслати повідомлення», кнопки очистки діалогу та повідомлень чат-бота та користувача. Також при генерації відповіді має відображатися спеціальний індикатор. Інтерфейс чат-бота зображено на рисунках 4.4 та 4.5.

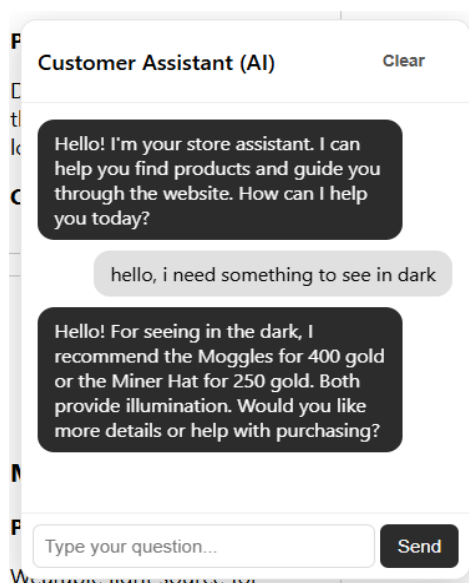


Рисунок 4.4. Інтерфейс чат-бота

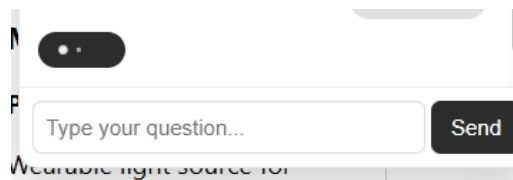


Рисунок 4.5. Індикатор генерації тексту

Тестування інтерфейсу підтвердило його працездатність та відповідність специфікаціям інтерфейсу. Усі елементи працюють належним чином.

У результаті проведеного модульного тестування було перевірено основні компоненти програмного додатку: модуль взаємодії зі штучним інтелектом, серверну частину та клієнтський інтерфейс. Тестування виконувалось ізольовано для кожного модуля.

Наступним етапом є інтеграційне та системне тестування. Їх мета – перевірка взаємодії між модулями в системі, а також оцінка роботи додатку відповідно до вимог.

Так як система складається з кількох взаємопов’язаних частин, важливо провести інтеграційне тестування. У його межах перевірялись такі зв’язки: взаємодія інтерфейсу користувача з сервером через HTTP-запити, передача даних з серверу до модуля ШІ, повернення відповіді від модуля ШІ до сервера а потім до інтерфейсу. Тестування проводилося за допомогою DevTools. На рисунках 4.6, 4.7, 4.8, 4.9 зображено тестування процесу діалогу – від передачі інтерфейсом повідомлення до сервера, до повернення відповіді від сервера до інтерфейсу.

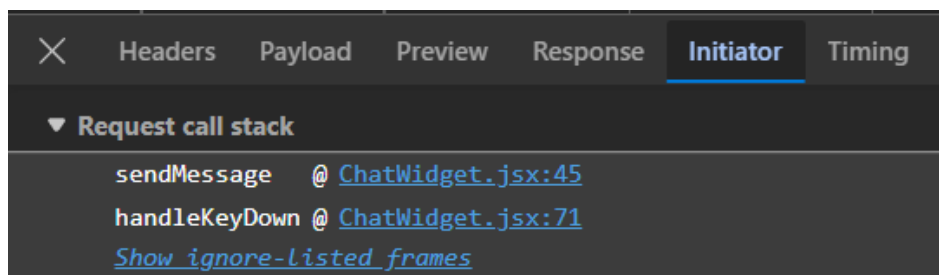


Рисунок 4.6. Компонент інтерфейсу

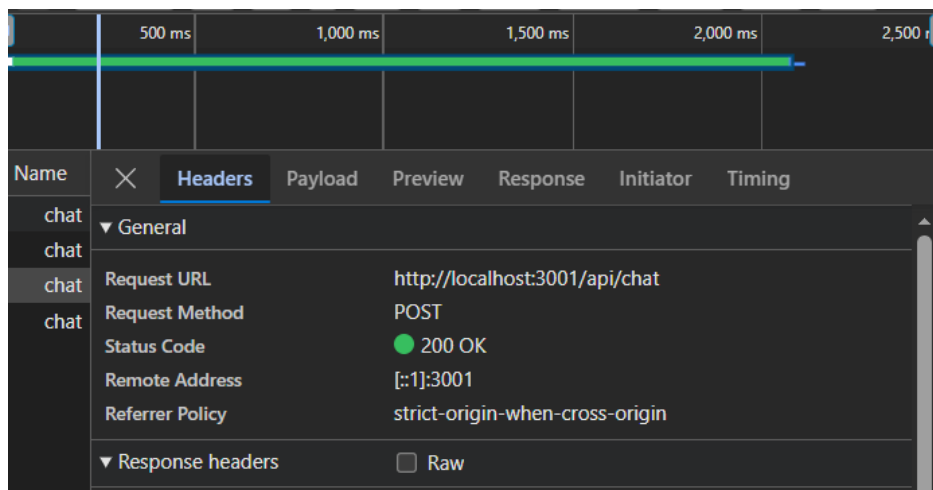


Рисунок 4.7. Виклик ендпоінта `api/chat` на сервері

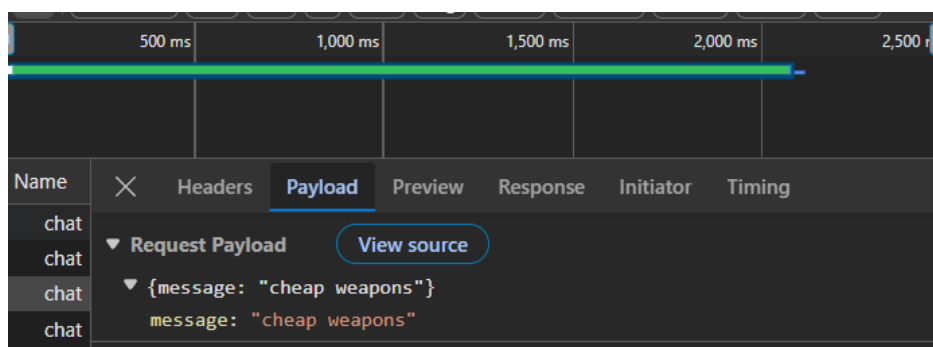


Рисунок 4.8. Текст, що передається до модуля ШІ

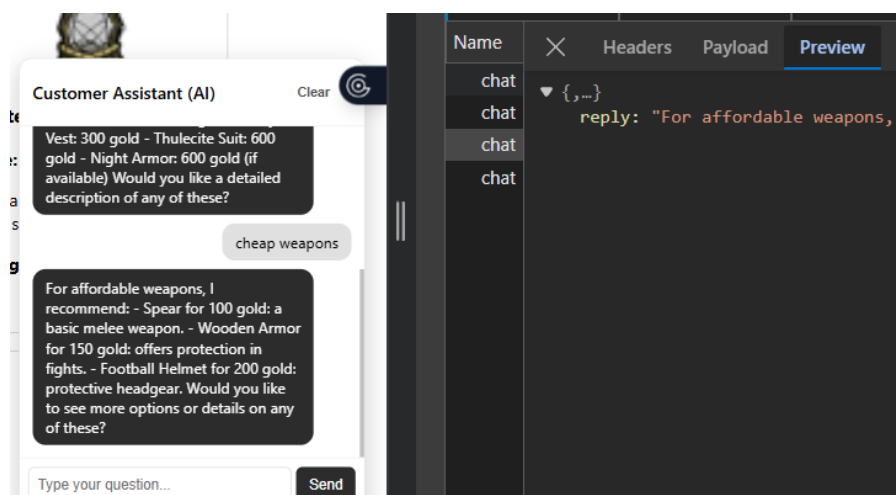


Рисунок 4.9. Відповідь передається користувачу

Результати інтеграційного тестування свідчать про те, що модулі пов'язані належним чином, інформація передається без втрат або спотворень і відповіді формуються відповідно до запитів користувача.

Далі було проведено системне тестування шляхом перевірки додатку як єдиного цілого. Для цього було змодельовано кілька тестових сценаріїв взаємодії користувача з чат-ботом, а саме:

- Запит рекомендації;
- Пошук товару;
- Пустий запит;
- Кілька запитів одночасно;
- Помилка з боку сервера моделі III;

Опис системного середовища, у якому проводилось тестування наведено у додатку Б.

На рисунку 4.10 відображено процес рекомендації товару. Користувач просить рекомендацію для дослідження печери. А чат-бот надає відповідь, спираючись на інформацію про товари магазину.

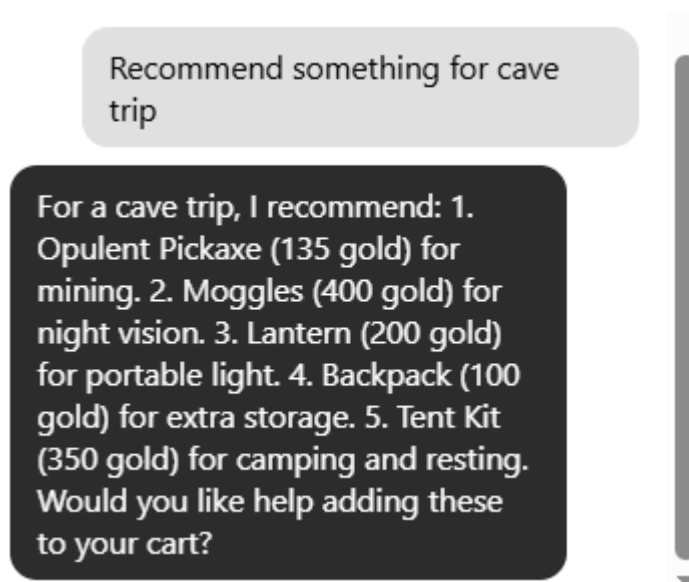


Рисунок 4.10. Рекомендація

На рисунку 4.11 відображено процес пошуку товару. Користувач запитує про сокиру, а чат-бот шукає сокиру у списку товарів і дає інформацію про знайдений товар.

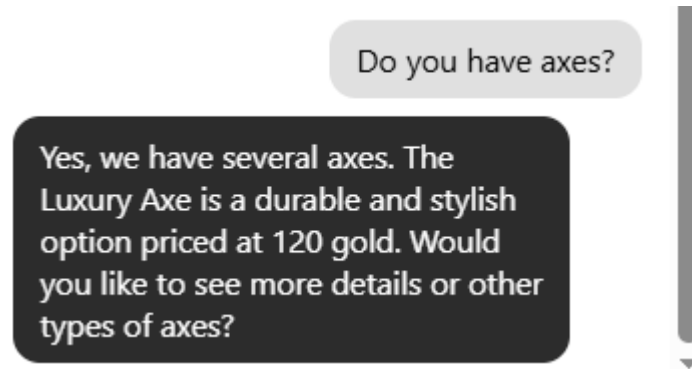


Рисунок 4.11. Пошук товару

Пусті запити та кілька запитів одночасно надіслати неможливо. Кнопка ігнорує дії користувача. Необхідно ввести повідомлення або дочекатися відповіді чат-бота. У таких випадках, система працює належним чином, адже пусті запити, або запити під час генерації відповіді можуть порушити хід діалогу, призвести до втрат контексту та неточних відповідей.

Останній сценарій тестування, ситуація, коли сервер ШІ недоступний, зображений на рисунку 4.12. Для цього було проведено симуляцію помилки шляхом від'єднання від мережі Інтернет (фактично додаток втрачає доступ до ШІ-моделі). У такому випадку чат-бот повідомить користувача про те, що ШІ наразі недоступний.

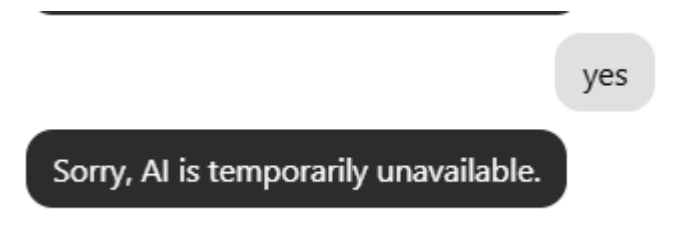


Рисунок 4.12. ШІ недоступний

Результати системного тестування свідчать про те, що система в цілому працює належним чином та правильно реагує при типових сценаріях взаємодії з користувачем.

Після проведення всіх етапів тестування можна зробити висновок, що чат-бот є працездатним та відповідає всім вимогам та проєктним специфікаціям. Він готовий для подальшого впровадження та експлуатації.

4.2. Робота користувача з чат-ботом

Користування додатком не потребує спеціальної підготовки та може бути здійснене будь-яким користувачем, який має базові навички роботи з веб-браузером.

Для початку користувачу необхідно відкрити веб-додаток Інтернет-магазину. Після завантаження сторінки, користувач побачить інтерфейс. У правому нижньому куті (рисунок 4.13) є кнопка, яка відкриває віджет чат-бота. Також цією кнопкою його можна згорнути за потреби.

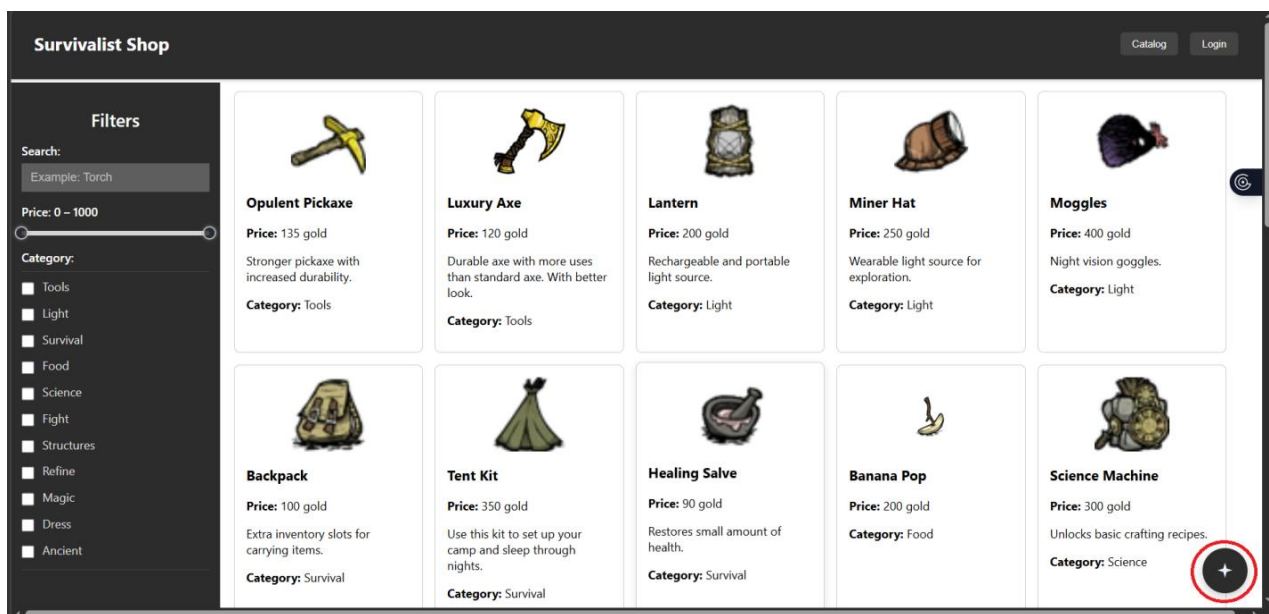
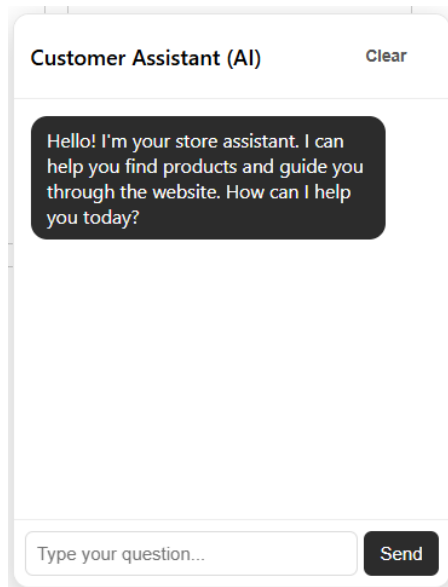


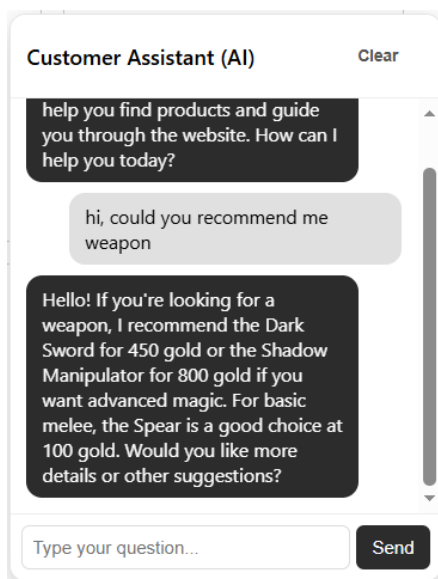
Рисунок 4.13. Інтерфейс Інтернет-магазину

При відкритті віджета, користувач побачить інтерфейс чат-бота: його назву, перше повідомлення чат-бота, поле для запиту, кнопку «Надіслати» та кнопку очищення діалогу (рисуюнок 4.14).



Рисуюнок 4.14. Інтерфейс чат-бота

Для ініціації діалогу користувач має ввести повідомлення у текстове поле та натиснути кнопку «Надіслати». Бот отримає повідомлення та дасть відповідь. На рисунку 4.15 зображено надання рекомендації після запиту користувача.



Рисуюнок 4.15. Чат-бот надає рекомендацію

Однією з переваг чат-ботів на основі штучного інтелекту є підтримка більшості мов. Чат-бот може перейти на мову запиту у реальному часі. Але іноді він помиляється при перекладі. На рисунку 4.16 зображено процес пошуку товару українською мовою.

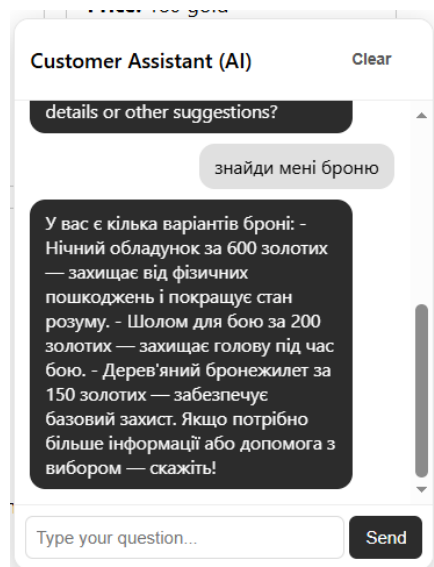


Рисунок 4.16. Пошук товару та відповідь мовою запиту

Якщо у користувача виникають питання, щодо використання самого веб-додатку (наприклад, користувач не може оформити замовлення), він може звернутися до чат-бота. На рисунку 4.17 зображено процес надання детальної консультації щодо оформлення замовлення.

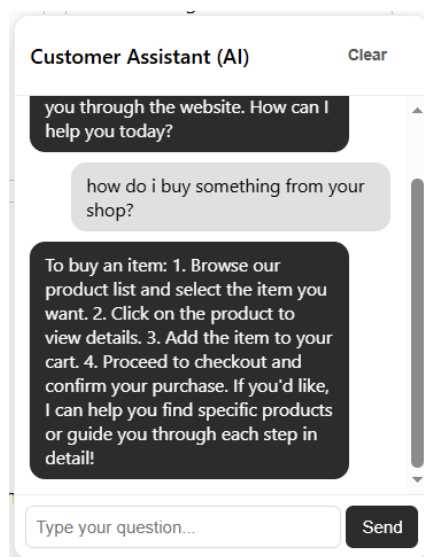


Рисунок 4.17. Надання консультації по сайту

Якщо користувач бажає очистити діалог, він має натиснути на кнопку «Очистити» у верхньому правому куті вікна чат-бота. Діалог очиститься і користувач зможе розпочати нову розмову з новим контекстом. На рисунку 4.18 зображено результат очищення діалогу та виділено місце розташування кнопки «Очистити».

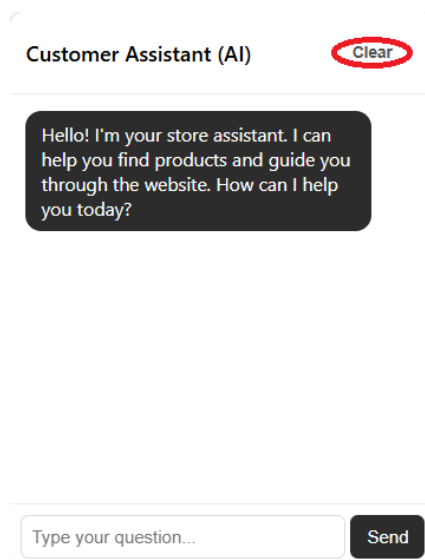


Рисунок 4.18. Очищений діалог

У цьому підрозділі було наведено опис роботи кінцевого користувача з розробленим чат-ботом у вигляді короткого керівництва користувача. Описано всі варіанти взаємодії користувача з чат-ботом: від розгортання віджета до різноманітних запитів та очищення діалогу.

4.3. Встановлення та експериментальне застосування чат-бота

Для встановлення додатку необхідно спочатку архів, що містить сам додаток та структуру баз даних (<https://shorturl.at/uaE4j>). На рисунку 4.19 зображено зміст завантаженого архіву.

 data.rar	20 368	20 368
 ds-shop.rar	110 051 651	110 051 651

Рисунок 4.19. Завантажений архів з додатком

Далі необхідно завантажити та встановити Node.js та XAMPP, якщо вони не були встановлені на пристрої. Node.js необхідний для запуску сервера та роботи API, XAMPP відповідає за роботу React-частини, зв'язок з Node.js та роботу баз даних (через phpMyAdmin). На рисунках 4.20 та 4.21 зображено веб-сторінки та посилання, через які можна завантажити Node.js (<https://shorturl.at/P4Oev>) та XAMPP (<https://shorturl.at/mUDoC>).

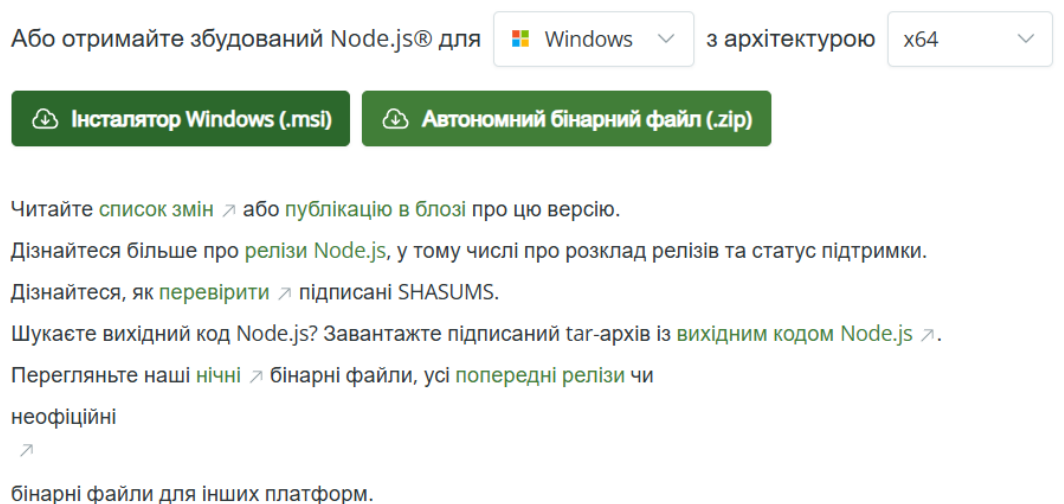


Рисунок 4.20. Сторінка завантаження Node.js

XAMPP is an easy to install Apache distribution containing MariaDB, PHP, and Perl. Just download and start the installer. It's that easy. Installers created using [InstallBuilder](#).

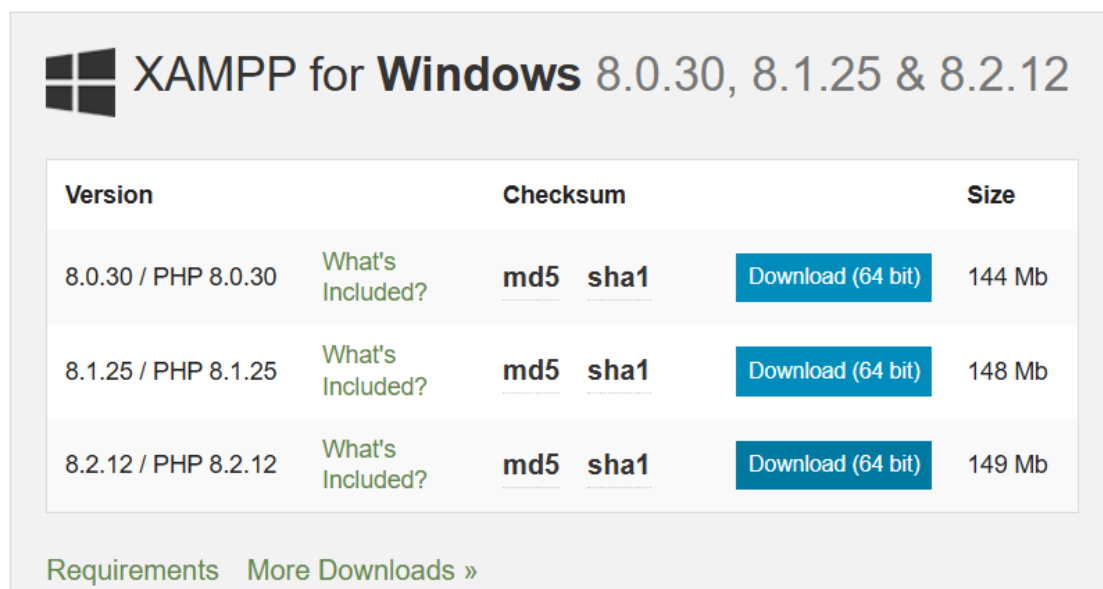


Рисунок 4.21. Сторінка завантаження XAMPP

Після завантаження необхідно встановити програми, дотримуючись інструкцій у програмах-інстальаторах. Після встановлення Node.js нічого робити не потрібно, цей додаток автоматично налаштований належним чином і не потребує інших дій з боку користувача. Після встановлення ХАМРР буде створена відповідна папка, куди треба буде вставити зміст раніше завантаженого архіву. Зміст архіву data.rar необхідно розпакувати в папку хампп/mysql/data, а зміст архіву ds-shop.rar – в папку хампп/htdocs. На рисунку 4.22 зображено ці папки.

htdocs	19.04.2026 12:23
img	18.04.2026 16:04
install	18.04.2026 16:11
licenses	18.04.2026 16:04
locale	18.04.2026 16:04
mailoutput	18.04.2026 16:04
mailtodisk	18.04.2026 16:04
MercuryMail	18.04.2026 16:11
mysql	18.04.2026 16:05

Рисунок 4.22. Папки хампп/htdocs та хампп/mysql

Тепер треба запустити ХАМРР Control Panel. Тут можна увімкнути потрібні для роботи додатку сервіси – Apache та MySQL. Інтерфейс панелі керування зображено на рисунку 4.23.

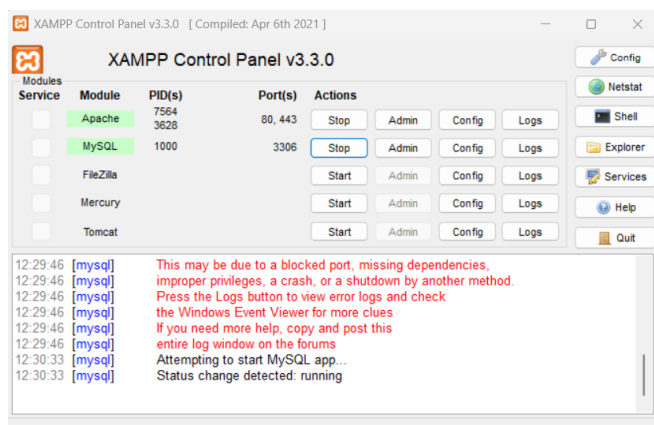
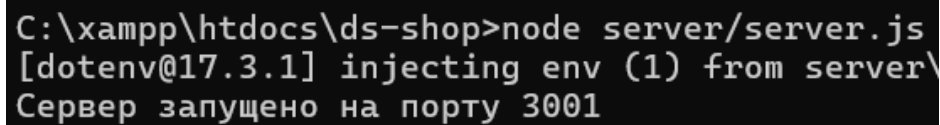


Рисунок 4.23. Інтерфейс ХАМРР Control Panel

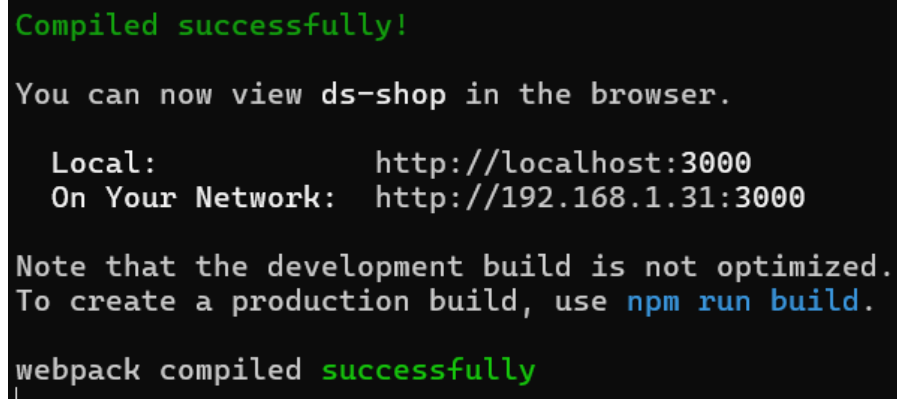
Далі необхідно запуснути сервер Node.js та додаток React.js. Обидві дії необхідно виконати через командний рядок (натиснути Win + R, у новому вікні написати cmd.exe та натиснути «Виконати»). За допомогою команди cd C:/xampp/htdocs/ds-shop треба перейти у директорію проєкту. Команда node server/server.js запусить сервер.



```
C:\xampp\htdocs\ds-shop>node server/server.js
[dotenv@17.3.1] injecting env (1) from server\
Сервер запущено на порту 3001
```

Рисунок 4.24. Сервер Node.js запущено

Для запуску React-додатку необхідно відкрити нове вікно командного рядка та перейти директорію проєкту. Команда npm start запусить додаток.



```
Compiled successfully!

You can now view ds-shop in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.1.31:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

Рисунок 4.25. Результат запуску додатку

У браузері, обраному за замовчуванням, відкриється веб-сторінка додатку магазину. Разом з доступом до основних функцій Інтернет-магазину, користувач отримає доступ і до розробленого чат-бота. Інтерфейс додатку з розгорнутим віджетом чат-бота зображено на рисунку 4.26.

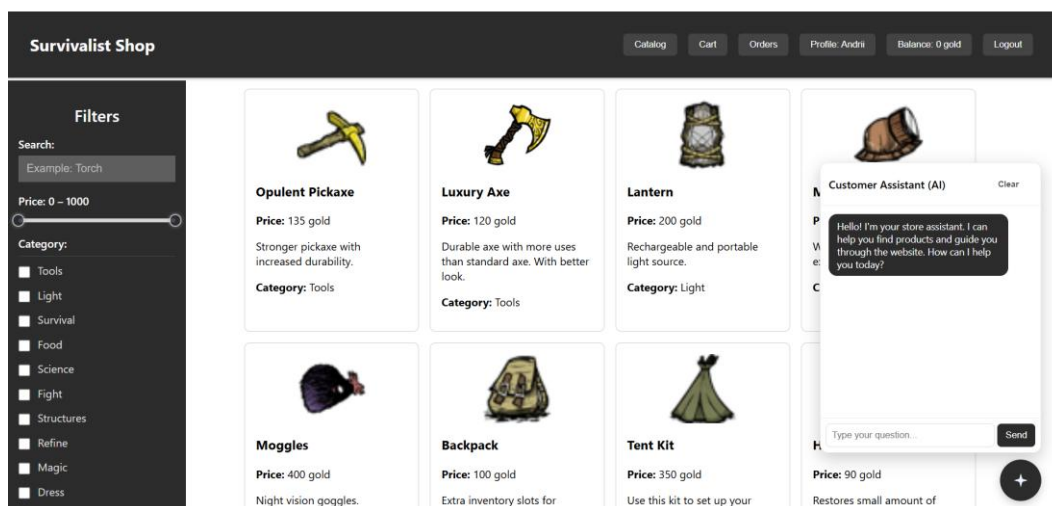


Рисунок 4.26. Інтерфейс Інтернет-магазину та чат-бота

Після цих дій додаток є встановленим та готовим до подальшого використання.

4.4. Висновки до розділу 4

У розділі 4 було описано процеси тестування, експлуатації та встановлення розробленого чат-бота. Було розроблено тест-план, тестові сценарії, проведено поетапне тестування. Після проходження всіх етапів тестування було зроблено висновок, що чат-бот є працездатним та відповідає всім вимогам та проєктним специфікаціям.

Далі було описано роботу користувача з розробленим чат-ботом у вигляді керівництва користувача, щоб у нього була можливість працювати з додатком навіть без спеціальної підготовки. Були розглянуті та описані всі ключові бізнес-процеси.

Після цього було описано процес встановлювання додатку на комп'ютер кінцевого користувача. Було наведено всі необхідні програми, що треба встановити. Потім було проведено процес імпорту проєкту із завантаженого .jar архіву. І після цього додаток було запущено за допомогою командного рядка та панелі керування XAMPP.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи бакалавра було створено інтелектуальний чат-бот для автоматизації взаємодії з користувачами Інтернет-магазину.

Було проведено аналіз сучасних підходів та існуючих рішень у сфері е-комерції. Проаналізовано предметну область – Інтернет-магазин, розглянуто основні бізнес-процеси, що відбуваються в межах цієї області. На основі цього аналізу було виконано постановку задачі на проектування.

Аналіз предметної області став основою для подальшого визначення функціональних і нефункціональних вимог. Ці вимоги були формалізовані у вигляді проєктних специфікацій: функціональних специфікацій, нефункціональних специфікацій та специфікацій інтерфейсу. Далі було описано різні аспекти проєктованої системи. Динамічну поведінку системи відображено через діаграми діяльності та послідовності. Потоки даних у межах системи або окремих модулів відображено через діаграми потоків даних. Апаратне забезпечення та зв'язки між його компонентами було відображено через діаграми розгортання та компонентів. Архітектуру чат-бота та її шари було спроектовано та відображено у нотації ArchiMate.

На основі результатів проектування реалізовано основні функції чат-бота. Node.js з фреймворком Express використано для бекенд-частини, React – для фронтенд-частини, а у якості мовної моделі кращим вибором стала модель GPT-4.1 папо. Результатом розробки став чат-бот на основі штучного інтелекту, що допомагає користувачам Інтернет-магазину.

Для перевірки працездатності чат-бота та його відповідності поставленим вимогам проведено тестування. Чат-бот пройшов ряд випробувань: модульне тестування, тестування інтерфейсу, інтеграційне та системне тестування. Розроблений чат-бот є працездатним та повністю відповідає вимогам та проєктним специфікаціями.

У результаті виконання роботи досягнуто поставленої мети – розроблено інтелектуальний чат-бот для автоматизації взаємодії з користувачами Інтернет-магазину, який підвищує ефективність обслуговування клієнтів та може бути використаний у реальних веб-системах електронної комерції. Також є потенціал до модернізації – код дозволяє перейти на новіші моделі та розширити функціонал чат-бота.

Повний вихідний код веб-застосунку доступний на GitHub (<https://shorturl.at/LYadh>) та Google Drive (<https://shorturl.at/uaE4j>).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Визначення нефункціональних вимог URL: <https://tinyurl.com/mr2f4ay7> (дата звернення: 27.01.2026)
2. Інструкція, як будувати UML-діаграми | DOU URL: <https://tinyurl.com/mr3vybjw> (дата звернення: 22.02.2026)
3. Справжній вплив соціальних мереж на комерцію: ключові тенденції та показники 2025 року URL: <https://shorturl.at/D0zFu> (дата звернення: 23.01.2026)
4. Стаття 182. Порухення недоторканності приватного життя - Кримінальний кодекс України URL: <https://shorturl.at/QJ6ve> (дата звернення: 25.01.2026)
5. Що таке UML Class Diagram, включаючи UML Class Diagram Maker URL: <https://tinyurl.com/4ehahwm7> (дата звернення: 25.02.2026)
6. Що таке модульне тестування? URL: <https://tinyurl.com/k4ettyz9> (дата звернення: 16.04.2026)
7. Що таке тест-план і з чого складається. URL: <https://tinyurl.com/mr98hn3y> (дата звернення: 16.04.2026)
8. Що таке функціональні вимоги: приклади та шаблони - Visure Solutions URL: <https://tinyurl.com/yr455tff> (дата звернення: 27.01.2026)
9. Blog - UML overview - where and why each UML diagram is used URL: <https://tinyurl.com/5e2jytxa> (дата звернення: 23.02.2026)
10. Claude Platform | Claude URL: <https://claude.com/platform/api> (дата звернення: 04.03.2026)
11. Component Based Diagram - Unified Modeling Language (UML) – GeeksforGeeks URL: <https://tinyurl.com/5xfhfzvz> (дата звернення: 28.02.2026)
12. Documentation | NestJS - A progressive Node.js framework URL: <https://docs.nestjs.com/> (дата звернення: 05.03.2026)
13. Express - Node.js web application framework URL: <https://expressjs.com/> (дата звернення: 04.03.2026)

14. Fast and low overhead web framework, for Node.js | Fastify URL: <https://fastify.dev/> (дата звернення: 03.03.2026)
15. Gartner Predicts Chatbots Will Become a Primary Customer Service Channel Within Five Years URL: <https://shorturl.at/rCkEo> (дата звернення: 23.01.2026)
16. Gemini API | Google AI for Developers URL: <https://tinyurl.com/yfr4j3rv> (дата звернення: 04.03.2026)
17. Introducing GPT-4.1 in the API URL: <https://openai.com/index/gpt-4-1/> (дата звернення: 04.03.2026)
18. ISO/IEC 27001:2022 - Information security management systems URL: <https://tinyurl.com/3ehsmuwp> (дата звернення: 24.01.2026)
19. Node.js – Run JavaScript Everywhere URL: <https://nodejs.org/en> (дата звернення: 05.03.2026)
20. Pricing | OpenAI API URL: <https://developers.openai.com/api/docs/pricing> (дата звернення: 06.03.2026)
21. React URL: <https://react.dev/> (дата звернення: 05.03.2026)
22. The ArchiMate® Enterprise Architecture Modeling Language URL: <https://tinyurl.com/mr4x7cfp> (дата звернення: 27.02.2026)
23. Top 6 AI Chatbots for E-Commerce in 2026 URL: <https://shorturl.at/se0wy> (дата звернення: 25.01.2026)
24. Understanding JWT Authentication: Benefits and Limitations | by alficodes | Medium URL: <https://rb.gy/r56sj7> (дата звернення: 16.03.2026)
25. Visual Studio Code – The open-source AI code editor URL: <https://code.visualstudio.com/> (дата звернення: 10.03.2026)
26. Vue.js - The Progressive JavaScript Framework | Vue.js URL: <https://vuejs.org/> (дата звернення: 09.03.2026)
27. What is Angular? • Angular URL: <https://angular.dev/overview> (дата звернення: 08.03.2026)
28. What is customer support? URL: <https://shorturl.at/RWd2U> (дата звернення: 23.01.2026)

29. What is Deployment Diagram? URL: <https://tinyurl.com/355un8mc> (дата звернення: 01.03.2026)
30. What is DFD (Data Flow Diagram) – GeeksforGeeks URL: <https://tinyurl.com/4st27953> (дата звернення: 26.02.2026)

Опис системного середовища в якому проводилось тестування

Апаратне забезпечення

- Процесор: AMD Radeon A6-7310
- Оперативна пам'ять: 4 GB
- SSD: 512 GB
- Графічний адаптер: інтегрований (React не використовує GPU)

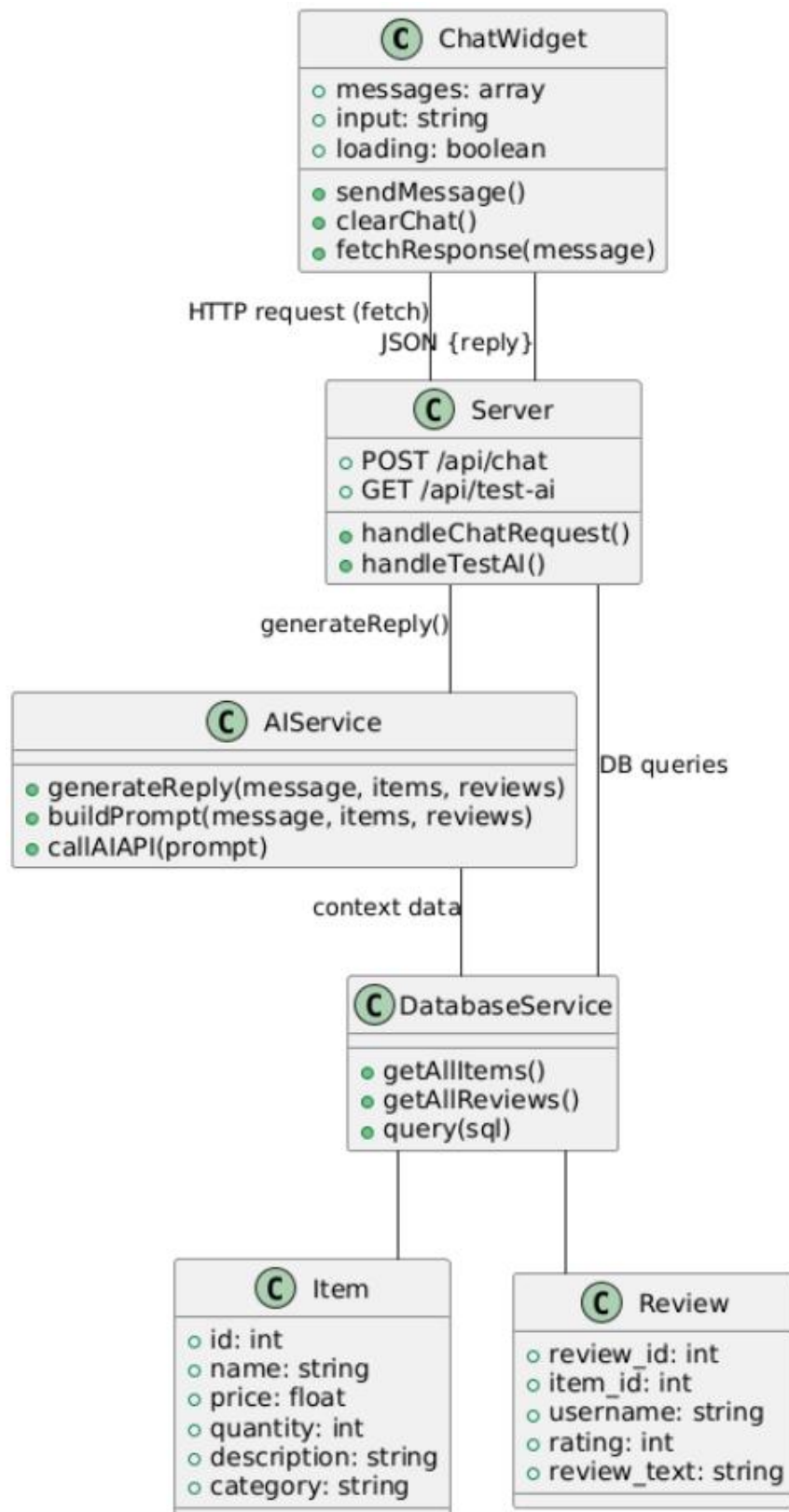
Програмне забезпечення

- Операційна система: Windows 10 (64-bit)
- Node.js: v18.17.1
- npm: 9.0
- MySQL Server: v8.0.1
- phpMyAdmin для перегляду БД
- React: React 18, додаток запущено через npm start
- Браузер: Google Chrome, Microsoft Edge
- Сервер: запуск через node server/server.js

Середовище розгортання

- Backend і frontend запущені локально на одному ПК
- Сервер: <http://localhost:3001>

Діаграма класів



Кросшарова архітектура у нотації Archimate

