

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет
імені Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ
Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук, проф.
_____ Ю. П. Кондратенко
« ____ » _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

**ІНТЕЛЕКТУАЛЬНА СИСТЕМА ПІДТРИМКИ КЛІЄНТІВ
ОНЛАЙН-МАГАЗИНУ КОМП'ЮТЕРНОЇ ТЕХНІКИ**

Спеціальність 122 «Комп'ютерні науки»

122 – КРБ – 401.22010117

Виконав студент 4-го курсу, групи 401
_____ *А. І. Овчаренко*
« 20 » червня 2024 р.

Керівник: канд. пед. наук, доцент
_____ *Н. М. Болюбаш*
« 20 » червня 2024 р.

Миколаїв – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет ім. Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

Рівень вищої освіти бакалавр
Спеціальність 122 «Комп'ютерні науки»
(шифр і назва)
Галузь знань 12 «Інформаційні технології»
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри інтелектуальних
інформаційних систем, д-р. техн. наук, проф.
_____ Ю. П. Кондратенко
«____» _____ 2024 р.

З А В Д А Н Н Я

на виконання кваліфікаційної роботи

Видано студенту групи 401 факультету комп'ютерних наук Овчаренку Артему Ігоровичу.

1. Тема кваліфікаційної роботи «Інтелектуальна система підтримки клієнтів онлайн-магазину комп'ютерної техніки».

Керівник роботи Болюбаш Надія Миколаївна, канд. пед. наук, доцент.

Затв. наказом Ректора ЧНУ ім. Петра Могили від «28» грудня 2024 р. № 271

2. Строк представлення кваліфікаційної роботи студентом «20» червня 2024 р.

3. Вхідні (початкові) дані до роботи: предметна сфера підтримки клієнтів у сфері продажу комп'ютерної техніки, набір даних з типовими запитамі клієнтів.

Очікуваний результат: система підтримки клієнтів онлайн-магазину комп'ютерної техніки, яка автоматизує процес обробки запитів та надає їм рекомендації.

4. Перелік питань, що підлягають розробці (зміст пояснювальної записки):

– здійснення аналізу предметної сфери продажу комп'ютерної техніки та дослідження теоретичних засад підтримки клієнтів у онлайн-магазинах;

– обґрунтування вибору технологій і інструментальних засобів розробки системи;

– розробка та здійснення програмної реалізації системи підтримки клієнтів онлайн-магазину комп'ютерної техніки.

5. Перелік графічного матеріалу: презентація, рисунки, таблиці.

6. Завдання до спеціальної частини: Захист від іонізуючих випромінювань.

7. Консультанти розділів роботи

| Розділ | Прізвище, ініціали та посада консультанта | Підпис |
|------------------------------------|--|--------|
| Спеціальна частина з охорони праці | Алексєєва А. О., канд. техн. наук, доцент кафедри екології | |

Керівник роботи канд. пед. наук, доцент Болюбаш Н. М.
(наук. ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Завдання прийнято до виконання Овчаренко А. І.
(прізвище та ініціали)

_____ (підпис)

Дата видачі завдання « 14 » січня 2024 р.

КАЛЕНДАРНИЙ ПЛАН виконання кваліфікаційної роботи

Тема: Інтелектуальна система підтримки клієнтів онлайн-магазину комп'ютерної техніки

| № | Найменування роботи | Початок | Закінчення | Примітки |
|----|--|------------|------------|----------|
| 1 | Визначення керівника і теми КРБ. Подання заяви на затвердження теми КРБ | 10.11.2023 | 15.11.2023 | Виконано |
| 2 | Отримання завдання на виконання КРБ | 10.01.2024 | 15.01.2024 | Виконано |
| 3 | Складання календарного плану | 16.01.2024 | 29.01.2024 | Виконано |
| 4 | Огляд літератури за темою дослідження. Аналіз предметної сфери онлайн продажу комп'ютерної техніки, дослідження теоретичних засад підтримки клієнтів у онлайн-магазинах | 30.01.2024 | 17.02.2024 | Виконано |
| 5 | Вибір технологій та інструментальних засобів розробки системи | 18.02.2024 | 29.02.2024 | Виконано |
| 6 | Створення дизайну, проєктування та програмна реалізація, тестування | 1.03.2024 | 15.04.2024 | Виконано |
| 7 | Робота над розділами фахової частини КРБ | 16.04.2024 | 31.04.2024 | Виконано |
| 8 | Проходження переддипломної практики, збір та аналіз матеріалів, остаточне оформлення розділів фахової частини БКР | 29.04.2024 | 12.05.2024 | Виконано |
| 9 | Розробка спеціальної частини з охорони праці | 13.05.2024 | 25.05.2024 | Виконано |
| 10 | Обговорення отриманих результатів з керівником та попередній захист КРБ | 27.05.2024 | 29.05.2024 | Виконано |
| 11 | Корегування роботи за результатами попереднього захисту | 30.05.2024 | 6.06.2024 | Виконано |
| 12 | Другий попередній захист КРБ | 10.06.2024 | 10.06.2024 | Виконано |
| 12 | Остаточне оформлення пояснювальної записки та слайдів доповіді до захисту | 11.06.2024 | 12.06.2024 | Виконано |
| 13 | Подання рецензенту та рецензування КРБ | 13.06.2024 | 13.06.2024 | Виконано |
| 14 | Подання КРБ, її електронної копії та інших документів (відгуку, рецензії) до захисту | 17.06.2024 | 21.06.2024 | Виконано |
| 15 | Захист КРБ перед ЕК | 27.05.2024 | 27.05.2024 | Виконано |

Розробив студент Овчаренко А. І.
(прізвище та ініціали)

_____ (підпис)

Керівник канд. пед. наук, доцент Болюбаш Н. М.
(наук. ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

« 29 » 01 2024 р.

АНОТАЦІЯ

кваліфікаційної роботи бакалавра
студента групи 401 ЧНУ ім. Петра Могили
Овчаренко Артема Ігоровича

Тема: «Інтелектуальна система підтримки клієнтів онлайн-магазину комп'ютерної техніки»

Кваліфікаційна робота бакалавра спрямована на розробку та здійснення програмної реалізації інтелектуальної системи підтримки клієнтів онлайн-магазину комп'ютерної техніки. Що є актуальним в умовах цифрової трансформації усіх сфер сучасного суспільства та зростаючого попиту на комп'ютерну техніку, оскільки її вибір вимагає наявності знань про внутрішню будову та основні принципи функціонування комп'ютерів.

Об'єкт роботи – процес підтримки клієнтів онлайн-магазину.

Предмет роботи – програмні засоби, технології та методи штучного інтелекту для підтримки клієнтів онлайн-магазину комп'ютерної техніки.

Мета роботи – підвищення ефективності роботи онлайн-магазину комп'ютерної техніки шляхом розробки системи підтримки клієнтів на основі методів штучного інтелекту.

Структура кваліфікаційної роботи включає фахову та спеціальну частину з охорони праці. Фахова частина включає вступ, три розділи, висновки та додатки. У першому розділі розкрито теоретичні аспекти підтримки клієнтів у онлайн-магазинах комп'ютерної техніки. У другому розділі обґрунтовано вибір технологій і засобів розробки системи. У третьому розділі описано проектування та програмну реалізацію інтелектуальної системи підтримки клієнтів онлайн-магазину комп'ютерної техніки.

Кваліфікаційна робота бакалавра містить 66 сторінок (без додатків), 31 рисунок, 31 джерело та 2 додатки.

Ключові слова: генеративний штучний інтелект, обробка природної мови, персоналізовані рекомендації, комп'ютерна техніка, онлайн-магазин.

ABSTRACT

**for bachelor's qualification work
of a student of 401 group at Petro Mohyla Black Sea National University
Ovcharenko Artema**

Theme: «An intelligent customer support system for an online computer store»

The bachelor's qualification work is aimed at the development and implementation of the software implementation of the intelligent customer support system of the online store of computer equipment. What is relevant in the conditions of digital transformation of all spheres of modern society and the growing demand for computer equipment, since its choice requires knowledge of the internal structure and basic principles of computer operation.

Object of work – online store customer support process.

Subject of work – software tools, technologies and methods of artificial intelligence to support customers of an online computer store.

The purpose of this work is to improving the efficiency of the online store of computer equipment by developing a customer support system based on artificial intelligence methods.

The structure of the bachelor's work includes a professional and special part on labor protection. The professional part includes an introduction, three chapters, conclusions and appendices. The first chapter reveals the theoretical aspects of customer support in online computer stores. The second section substantiates the choice of technologies and system development tools. The third section describes the design and software implementation of an intelligent customer support system for an online computer store.

The bachelor qualification work contains 66 pages (without appendices), 31 figures, 31 sources and 2 appendices.

Keywords: generative artificial intelligence, natural language processing, personalized recommendations, computer equipment, online store.

ЗМІСТ

| | |
|---|----|
| ПЕРЕЛІК СКОРОЧЕНЬ..... | 4 |
| ВСТУП..... | 5 |
| 1 ТЕОРЕТИЧНІ ЗАСАДИ ПІДТРИМКИ КЛІЄНТІВ ОНЛАЙН-МАГАЗИНІВ КОМП'ЮТЕРНОЇ ТЕХНІКИ | 7 |
| 1.1 Характеристика сучасних персональних комп'ютерів та їх складових | 7 |
| 1.2 Сфера продажу комп'ютерної техніки | 13 |
| 1.3 Штучний інтелект у підтримці клієнтів інтернет-магазинів..... | 18 |
| 1.4 Постановка задачі..... | 23 |
| Висновки до розділу 1..... | 25 |
| 2 ТЕХНОЛОГІЇ ТА ЗАСОБИ РОЗРОБКИ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ..... | 26 |
| 2.1 Середовище розробки Visual Studio Code | 26 |
| 2.2 Засоби розробки інтерфейсу користувача..... | 27 |
| 2.3 JavaScript та бібліотека React..... | 30 |
| 2.4 Протокол зв'язку WebSocket | 34 |
| 2.5 Мова програмування Python та Python-бібліотеки..... | 35 |
| 2.6 Фреймворк Llamaindex..... | 38 |
| Висновки до розділу 2..... | 40 |
| 3 ПРОЄКТУВАННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ПІДТРИМКИ КЛІЄНТІВ ОНЛАЙН-МАГАЗИНУ КОМП'ЮТЕРНОЇ ТЕХНІКИ | 42 |
| 3.1 Основні етапи створення системи | 42 |
| 3.2 Проєктування та програмна розробка інтерфейсу інтернет-магазину | 43 |
| 3.3 Програмна реалізація та демонстрація підтримки клієнтів магазину за допомогою WebSocket-клієнта..... | 48 |
| 3.4 Інтеграція мовних моделей за допомогою методу RAG..... | 53 |
| 3.5 Тестування та демонстрація роботи чат-бота..... | 57 |
| Висновки до розділу 3..... | 61 |
| ВИСНОВКИ | 63 |

| | |
|--------------------------------------|----|
| ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ | 66 |
| ДОДАТОК А Код файлу logic.js | 69 |
| ДОДАТОК Б Код файлу ai_code.py | 74 |

ПЕРЕЛІК СКОРОЧЕНЬ

ШІ – штучний інтелект

AI – Artificial Intelligence

AMD – Advanced Micro Devices

IBM – International Business Machines

API – Application Programming Interface

CSS – Cascading Style Sheets

GPT – Generative Pre-Trained Transformer

HTML – HyperText Markup Language

IDE – Integrated Development Environment

ML – Machine Learning

NLP – Natural Language Processing

RAG – Retrieval-Augmented Generation

LLM – Large Language Model

ВСТУП

Актуальність. В умовах стрімкої динаміки розвитку сучасного ринку електронної комерції та зростаючого попиту на комп'ютерну техніку одним із ключових аспектів успішної діяльності магазинів комп'ютерної техніки є забезпечення якісної підтримки клієнтів онлайн. Перспективним напрямом підвищення ефективності онлайн продаж є використання сучасних методів та мовних моделей штучного інтелекту для забезпечення кваліфікованого персоналізованого підходу у наданні відповідної підтримки клієнтам із урахуванням їх потреб та наявних знань про принципи функціонування комп'ютерної техніки.

Сектор обслуговування клієнтів онлайн-магазинів переживає революцію, пов'язану із використанням штучного інтелекту. Компанії все частіше вдаються до використання штучного інтелекту для обслуговування клієнтів з метою розширення можливостей підтримки. У сьогоднішніх реаліях це не просто тенденція, а стратегічна трансформація, яка встановлює нові орієнтири у сфері взаємодії з клієнтами та їх підтримки.

Інтеграція технологій розмовного ШІ у сферу електронної комерції в операціях обслуговування клієнтів автоматизує та покращує взаємодію клієнтів з інтернет-магазинами шляхом персоналізації взаємодії та оптимізації процесу обслуговування клієнтів, роблячи його більш ефективним та результативним. Сфера онлайн продажу комп'ютерної техніки має свою специфіку, яка обумовлена необхідністю врахування спеціальних знань стосовно комп'ютерної техніки при наданні підтримки. Виявлення можливостей використання генеративного штучного інтелекту для вирішення цієї проблеми не є дослідженим у повній мірі.

Це обумовило **мету роботи**, яка полягає у підвищенні ефективності роботи онлайн-магазину комп'ютерної техніки шляхом розробки системи підтримки клієнтів на основі методів штучного інтелекту.

Відповідно до поставленої мети було сформульовано **завдання**:

- здійснити аналіз предметної сфери продажу комп'ютерної техніки та дослідити теоретичні засади підтримки клієнтів у онлайн-магазинах;
- обґрунтувати вибір технологій та інструментальних засобів розробки системи з використанням методів штучного інтелекту;
- розробити та здійснити програмну реалізацію системи підтримки клієнтів онлайн-магазину комп'ютерної техніки.

Об'єкт роботи – процес підтримки клієнтів онлайн-магазину.

Предмет роботи – програмні засоби, технології та методи штучного інтелекту для підтримки клієнтів онлайн-магазину комп'ютерної техніки.

Методологічною основою дослідження є загальнонаукові та статистично-аналітичні методи, які дозволили комплексно вивчити предмет та об'єкт дослідження, дослідити основні підходи до застосування методів та моделей ШІ для підтримки клієнтів онлайн-магазину комп'ютерної техніки.

Практичне значення отриманих результатів полягає в тому, що використання розробленої системи дозволить оптимізувати процес обслуговування клієнтів, покращити комунікацію з ними, автоматизувати відповіді на запитання і запити та підвищити загальну якість обслуговування у онлайн торгівлі комп'ютерною технікою.

Структура кваліфікаційної роботи. Відповідно до мети, завдань і предмета дослідження, бакалаврська робота містить основну та спеціальну частини. Основна частина роботи складається із вступу, трьох розділів, висновку, списку використаних джерел та 2 додатків. Загальний обсяг роботи – 85 сторінок, із них основного тексту основної частини – 66 сторінок, спеціальної – 17 сторінок. Кількість використаних джерел – 26.

1 ТЕОРЕТИЧНІ ЗАСАДИ ПІДТРИМКИ КЛІЄНТІВ ОНЛАЙН-МАГАЗИНІВ КОМП'ЮТЕРНОЇ ТЕХНІКИ

1.1 Характеристика сучасних персональних комп'ютерів та їх складових

Базова апаратна конфігурація комп'ютера, яка складається з комплектуючих пристроїв, що входять до його складу, обумовлює характер програмного забезпечення, яке може бути встановлене на комп'ютері та відповідно задач, які можна буде вирішувати. Характеристики апаратних засобів впливають на споживчі характеристики комп'ютера.

Проектування комп'ютера на основі принципу відкритої архітектури дає можливість при виборі обладнання підбирати різні конфігурації комп'ютера відповідно до матеріальних можливостей та тих задач, які будуть на ньому розв'язуватися. Принцип відкритої архітектури полягає в наступному [1]:

- регламентуються і стандартизуються лише опис принципу дії комп'ютера і його конфігурація – певна сукупність апаратних засобів і з'єднань між ними. Таким чином, комп'ютер можна збирати з окремих вузлів і деталей, розроблених і виготовлених незалежними фірмами-виробниками;
- комп'ютер легко розширюється і модернізується за рахунок наявності внутрішніх розширювальних гнізд, в які можна вставляти різні пристрої, що задовольняють заданому стандарту, і тим самим встановлювати конфігурацію машини відповідно до необхідних вимог.

Розрізняються наступні види персональні комп'ютери: стаціонарні – настільні ПК, мобільні – ноутбуки та нетбуки, планшетні – iPad. Смартфони сьогодні суміщають функціональність мобільного телефона та кишенькового ПК.

Розглянемо склад та призначення основних блоків персонального комп'ютера. Будь-який комп'ютер має набір обов'язкових та додаткових пристроїв, які визначають його функціональність. Пристрої комп'ютера можуть бути внутрішніми та зовнішніми.

До основних внутрішніх пристроїв ПК відносять: процесор, оперативну пам'ять, шину та материнську плату. Важливу роль у складі сучасного комп'ютера відіграють зовнішні запам'ятовуючі пристрої, до яких відносять диски HDD та SSD, призначених для довготривалого збереження інформації, оптичні диски та flash-пам'ять. Оптичні компакт-диски (CD-R, CD-RW, DVD-RW(R)), призначені для збереження незмінної інформації, поступово витісняються flash-накопичувачами.

До зовнішніх також відносять пристрої введення, виведення інформації та зв'язку з іншими комп'ютерами. До основних пристроїв введення інформації відносять: клавіатуру, мишу, сканер, цифрову камеру та мікрофон. До основних пристроїв виведення інформації відносять: монітор, звукові колонки, принтер. Зображення на екрані монітору визначається відеосигналом, який надходить від спеціального пристрою – відеоадаптера. Відеоадаптер керує роботою монітору та виконує функції відеоконтролера, відеопроцесора і відеопам'яті, може бути реалізований у вигляді окремої плати або інтегрований у материнську плату. До пристроїв, що забезпечують зв'язок з іншими комп'ютерами, відносять модем та мережеву плату

У настільного персонального комп'ютера пристрої конструктивно розміщуються у системному блоці або приєднуються до роз'ємів материнської плати та портів введення-виведення. Ноутбуки та нетбуки є переносними комп'ютерами, у яких монітор, клавіатура, цифрова камера та мікрофон вмонтовані у корпус, а одночасно з мишею є можливість користуватися вбудованим маніпулятором тачпад. Принципова схема роботи усіх типів персональних комп'ютерів є єдиною (рис. 1.1).

До основних комплектуючих персонального комп'ютера необхідно віднести наступні: процесор; оперативна пам'ять; материнська плата; відеоадаптер; диск (HDD, SSD); блок живлення; система охолодження; звукова карта; пристрої введення клавіатура та миша; системний блок; монітор.

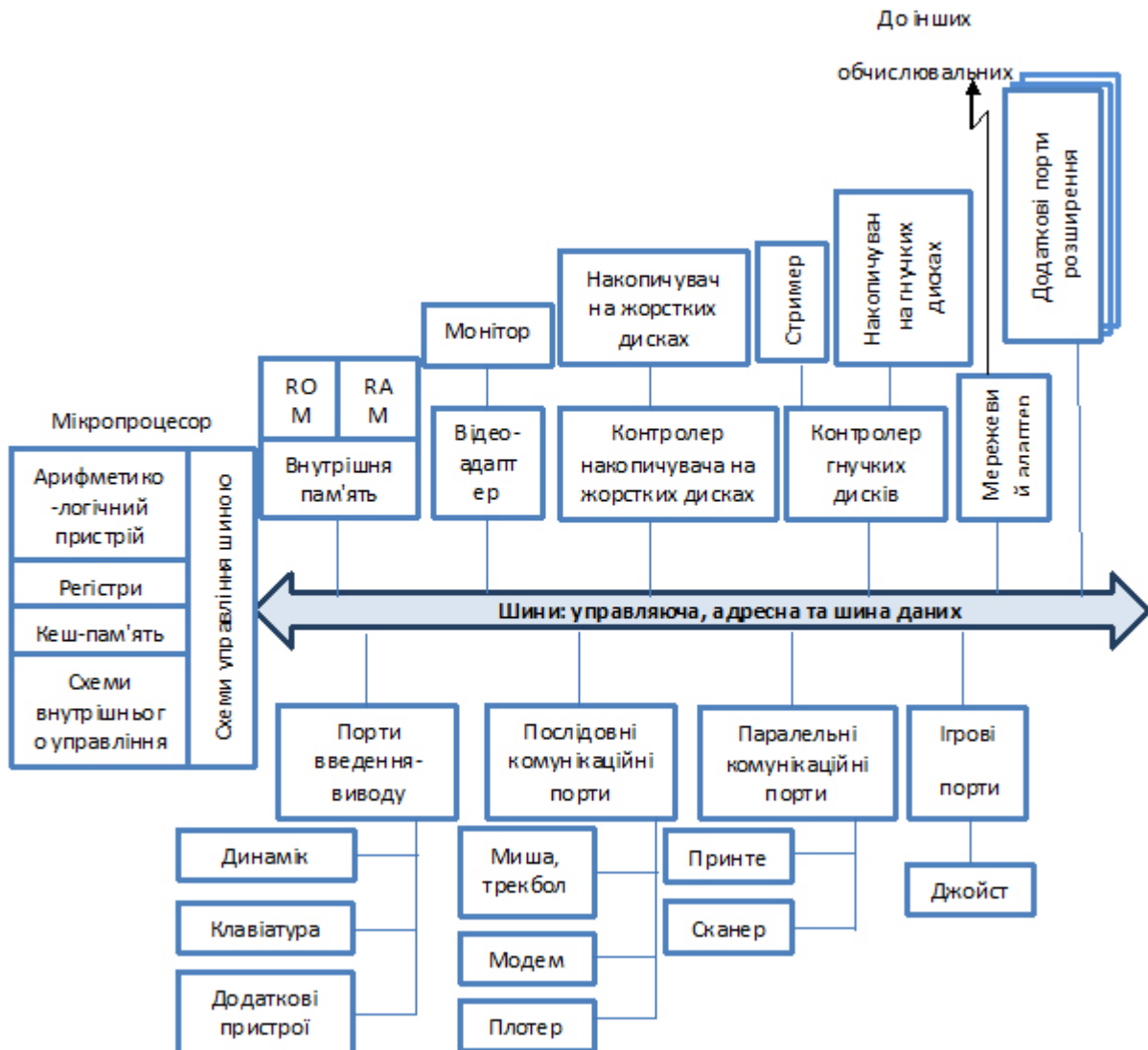


Рисунок 1.1 – Загальна схема персонального комп'ютера

З основних комплектуючих [2] при підборі конфігурації персонального комп'ютера його функціональність визначають наступні: процесор; оперативна пам'ять; материнська плата; відеоадаптер; диск (HDD, SSD); блок живлення; система охолодження; системний блок. Характеристики саме цих пристроїв можуть розрізнятися суттєво, а їх підбір дозволяє отримувати різні за функціональністю комплектації персонального комп'ютера. При виборі конфігурації персонального комп'ютера слід враховувати усі нюанси – від виробника до характеристики кожної комплектуючої.

Процесор зазвичай обирається одним із перших у черзі компонентів. Будучи реалізованим у вигляді ВІС, він характеризується багатьма параметрами та властивостями, які можуть суттєво відрізнятися у різних виробників. Найбільш практично важливими та функціональними характеристиками процесора є наступні:

- тактова частота процесора (ГГц);
- частота системної шини (МГц);
- кількість ядер;
- кількість потоків;
- розрядність мікропроцесора.

Швидкодія процесора суттєво залежить від тактової частоти та кількості ядер. Характеристики процесорів різних виробників представлені рисунку 1.2.

| | BOOST BASE | CORES/ THREADS | GAMECACHE | PCIe SUPPORT | SUGGESTED PRICE (USD) | | BOOST BASE | CORES/ THREADS | SMARTCACHE | PCIe SUPPORT | SUGGESTED PRICE (USD) |
|--|---------------|-------------------|-----------|-----------------|--------------------------|------------------|---------------|-------------------|------------|-----------------|--------------------------|
| Ryzen™ 9 3900X | 4.6 3.8 | 12 / 24 | 70MB | 4.0 | \$499 | Core i9-9900K(F) | 5.0 3.6 | 8/16 | 16MB | 3.0 | \$488 |
| Ryzen™ 7 3800X | 4.5 3.9 | 8 / 16 | 36MB | 4.0 | \$399 | Core i7-9700K(F) | 4.9 3.6 | 8/8 | 12MB | 3.0 | \$374 |
| Ryzen™ 7 3700X | 4.4 3.6 | 8 / 16 | 36MB | 4.0 | \$329 | | | | | | |
| Ryzen™ 5 3600X | 4.4 3.8 | 6 / 12 | 35MB | 4.0 | \$249 | Core i5-9600K(F) | 4.6 3.7 | 6/6 | 9MB | 3.0 | \$262 |
| Ryzen™ 5 3600 | 4.2 3.6 | 6 / 12 | 35MB | 4.0 | \$199 | | | | | | |
| Ryzen™ 5 3400G with Radeon™ RX Vega 11 Graphics | 4.2 3.7 | 4 / 8 | 6MB | 3.0 | \$149 | Core i5-9400 | 4.1 2.9 | 6/6 | 9MB | 3.0 | \$182 |
| Ryzen™ 3 3200G with Radeon™ Vega 8 Graphics | 4.0 3.6 | 4 / 4 | 6MB | 3.0 | \$99 | Core i3-9100 | 4.2 3.6 | 4/4 | 6MB | 3.0 | \$122 |

Рисунок 1.2 – Характеристики процесорів

Материнська плата обирається під обраний центральний процесор, оскільки тип роз'єму для приєднання процесора на материнській платі має відповідати роз'єму обраного процесора. При виборі материнської плати необхідно звертати увагу на такі її характеристики:

- роз'єм процесора;

- підтримка пам'яті: кількість слот для приєднання модулів пам'яті і типи підтримуваних модулів пам'яті та їх частота;
- наявність чи відсутність певних інтегрованих контролерів (відеоадаптера та інших);
- кількість слот розширення PCI;
- форм-фактор плати;
- частота шини;
- кількість та тип роз'ємів для дисків (SATA та інші);
- наявність інших роз'ємів (USB та інші).

Таким чином, вибір материнської плати обумовлює подальшу конфігурацію комп'ютера.

Наступним важливим компонентом, який впливає на швидкість та потужність комп'ютера, є оперативна пам'ять. Конструктивно пам'ять реалізована у вигляді модулів, які вставляють у відповідні слоти материнської пам'яті та мають такі характеристики:

- тип модуля пам'яті та роз'ємів (DDR3, DDR4, DDR5);
- тактова частота пам'яті (МГц, ГГц);
- частота шини;
- обсяг модуля пам'яті.

Вибір материнської плати обумовлює форм-фактор системного блоку для настільного ПК. Корпус потрібно обирати в першу чергу за сумісністю по форм-фактору з блоком живлення та материнською платою, а в вже в другу чергу за його системою охолодження та зовнішнім виглядом. Слід зазначити, що обирати корпус без систем охолодження або провітрювання не дуже гарна думка, тому що інші компоненти при роботі можуть перегріватись саме через такий корпус. Основні характеристики: форм-фактор, охолодження, відсіки, порти, розміри.

Чим потужніші комплектуючі, тим потужнішим має бути охолодження та блок живлення настільного ПК. Найважливіший компонент, на якому тримаються

усі інші – блок живлення. Через нього живлення подається до усіх компонентів, тому спочатку треба визначитися з усіма іншими компонентами, а тільки потім обирати потрібний блок живлення. Також необхідно визначити формфактор корпусу, та перевірити роз'єми та їх кількість. Основні характеристики: формфактор, потужність, роз'єми живлення, сертифікат.

Відеоадаптер є одним із основних компонентів але потрібно зазначити, що відеоадаптер може бути інтегрований до материнської плати. Також слід перевірити розміри та з'єднати їх із форм-фактором корпусу. Основні характеристики: обсяг і тип пам'яті, її частота.

В сучасних комп'ютерних конфігураціях використовуються жорсткі диски двох типів: HDD – традиційні магнітні диски і SSD – більш сучасні твердотільні диски електронного типу. Кожному з видів властиві свої переваги і недоліки. Твердотільні диски мають більш високі показники швидкості, ніж їхні старші побратими, але поступаються їм в ємності (співвідношення ємність / вартість у SSD дисків значно нижче, ніж у HDD). Ще один істотний недолік твердотільних дисків – нездатність зберігати інформацію протягом тривалого часу та висока вартість. Прискорює втрату інформації дисками даного виду відключення живлення, підвищена температура, ступінь зношеності накопичувача.

Що стосується HDD, то можна сказати, що вони дуже надійні та співвідношення ціна-ємність набагато переважає SSD але ця перевага перекривається замалою швидкістю читання файлів, тому можна зробити висновок, що ці диски використовують для зберігання великої кількості інформації, в той час як SSD використовують для роботи із файлами та додатками. Основні характеристики: тип пам'яті, обсяг, інтерфейс, швидкість читання та запису.

Виходячи з характеристик дисків різного типу, фахівці по збірці ПК пропонують оптимальне рішення – використання при побудові конфігурації дисків обох типів, на один з яких (HDD-диск) покладається завдання зберігання даних, а інший забезпечить швидку роботу операційної системи.

Для стаціонарних ПК, ноутбуків та нетбуків комплектуючі мають різні формфактори. Ноутбуки і нетбуки характеризуються максимальною мобільністю. Тому вони мають популярність серед людей, яким необхідно часто мати комп'ютер при собі. Стаціонарні ж ПК не мають такої мобільності, однак вони є більш потужними. Тому вони затребувані користувачами, які цього потребують: геймерів, проектувальників, людей, які працюють у режимі багатозадачності.

1.2 Сфера продажу комп'ютерної техніки

Сфера продажу комп'ютерної техніки є однією з найбільш динамічних галузей на сучасному ринку, яка швидко розвивається. Вона охоплює широкий спектр продуктів, включаючи настільні комп'ютери, ноутбуки, планшети, комплектуючі, периферійні пристрої та аксесуари. комп'ютерна техніка стає все більш затребуваною.

На комп'ютерному ринку нашої країни можна придбати продукцію іноземних компаній IBM, Hewlett Packard, Compaq, Dell, DEC, Apple, Acer та інших, комп'ютери яких імпортуються та продаються у готовому вигляді. Проте значну частину ринку складають комп'ютери, конфігурація яких підібрана за потребами користувачів.

При виборі комп'ютеру необхідно виходити з того, для вирішення яких задач він буде використовуватися: для офісної роботи, домашнього перегляду фільмів, роботи з документами, ігор або ж відеомонтажу. За призначенням ПК можна розділити на такі типи комп'ютерів [3]:

- ігрові: мають потужну відеосистему та інші характеристики ПК, забезпечують комфортну роботу з комп'ютерними іграми (звідси їх назва), підходять для роботи з графікою, аудіоматеріалами, монтажу відео. Є найбільш потужними та дорогими;

- мультимедійні: призначені для прослуховування музики, перегляду відео, невимогливих ігор. Менш вимогливі до потужності ПК;

– офісні: призначені для повсякденних робочих завдань, таких, як набір тексту, відправка електронних листів, ведення електронних розрахунків, бухгалтерії, здійснення відео нарад по Skype або Zoom тощо. Такий ПК досить часто підходить користувачам і для дому. Є менш вимогливими до швидкодії та потужності ПК й більш дешевими.

При виборі комплектуючих для офісних збірок підійде чотирьох або навіть двоядерний процесор, якщо кількість потоків дорівнює хоча б 4. Для мультимедійних домашніх ПК чотирьох ядер також достатньо, а для ігрових краще мати процесор з 6-ма або 8-ма ядрами. Це дає можливість отримати достатній приріст продуктивності при роботі з графікою та відео.

Сьогодні на ринку представлені в основному процесори фірми Intel та AMD, продукція яких має свої переваги і недоліки, тому вибір на користь одного або іншого здійснюється в кожному випадку індивідуально, залежно від потреб [4].

Відносно вартості сучасні процесори можна розбити на нижче перераховані групи:

- процесори вартістю до 50\$ цілком підходять для офісних задач та пошуку інтернет-ресурсів. Сюди відносяться гібриди процесорів з материнськими платами, які за рахунок високої інтеграції забезпечують збірку недорогих, але надійних ПК;
- серед процесорів вартістю в діапазоні 50-100\$ більша перевага у процесорів марки AMD, оскільки їх використання дозволяє збирати відносно недорогі комп'ютери, здатні підтримувати ігри на налаштуваннях початкового рівня;
- у категорії цін 100-180\$ вигідно відрізняються процесори лінійки AMD Ryzen, оптимально поєднуючи продуктивність, енергоефективність та доступні ціни. Процесори даного типу можуть підтримувати ігри, що вимагають великої потужності, на налаштуваннях початкового або навіть середнього рівня. Що стосується процесорів Intel даної цінової категорії, то слід вибрати пристрої 8-го покоління, що характеризуються оптимальним співвідношенням вартості з продуктивністю пристрою;

– у категорії процесорів вартістю понад 180\$ слід віддати перевагу продукції Intel, які на однакових характеристиках забезпечують FPS до 10% вище, ніж AMD процесори і вигідно поєднують продуктивність і вартість. З процесорів AMD даної цінової категорії заслуговують на увагу лінійки Ryzen 5 і 7;

– найпотужніші на сьогодні процесори Intel i9 і AMD Ryzen Threadripper. Вони володіють високою вартістю, яка далеко не завжди виправданою. Їх, як правило, купують для найпотужніших ігрових ПК гравці-професіонали. В інших випадках більш важливо дотримуватися балансу між продуктивністю основних компонентів: процесора і відеокарти, дотримання якого цілком можливо при використанні процесорів Intel 7-го покоління або AMD 5-го покоління.

Материнські плати на ринок комп'ютерної техніки надходять від декількох виробників. Фахівці рекомендують вибирати плати від Asus Gigabyte, msi, asrock мотивуючи вибір підвищеною надійністю продукції даних виробників при тому, що ціни їх відповідають вартості аналогів від інших виробників. Материнські плати даних виробників відрізняються високою якістю, при цьому вони регулярно випускають оновлення з метою усунення недоліків і підтримки нового обладнання.

Вибираючи модуль пам'яті необхідно також враховувати частоту, на якій вона працює. Знову ж таки, це не є суттєвим для офісних та мультимедійних комп'ютерів. Однак у випадку ігрових комп'ютерів це дає істотний приріст продуктивності.

Відеокарта для офісного ПК може бути і інтегрованою у материнську плату. Для мультимедійних збірок цього також буде достатньо. Ігрові комп'ютери вимагають потужних дорогих відеокарт. Досить часто в побудові комп'ютерної конфігурації використовуються гібридні процесори з вбудованою відеокартою. До 2016 року оптимальним варіантом для таких конфігурацій були чотириядерні гібридні AMD процесори, на основі яких можна було створити відносно недорогі універсальні комп'ютерні системи, які характеризуються досить хорошою продуктивністю, як самого процесора, так і вбудованої в нього відеокарти, а також економним енергоспоживанням і невисоким шумовим рівнем.

Але особливістю будь-якої вбудованої відеокарти є використання ними оперативної пам'яті ПК, що значно обмежує їх швидкість. Тому будь-який геймерський комп'ютер потребує оснащення дискретною відеокартою достатньої потужності.

Найбільш важливою характеристикою при виборі відеокарти для ігрового комп'ютера є використовуваний нею графічний процесор (GPU), на який покладається виконання всіх обчислювальних операцій. Що стосується графічної пам'яті, то вона не бере участі в обробці даних, а тільки зберігає їх, тому її ємність менш значима.

Дуже важливий вплив на коректність роботи відеокарти має правильно організоване охолодження, тому при виборі пристрою слід поцікавитися системою охолодження. Щоб виключити передчасний вихід відеокарти з ладу, не слід вибирати пристрої з пасивним охолодженням, хоча вони і характеризуються тихою роботою. Щоб забезпечити тиху роботу відеокарти активного типу, слід вибирати пристрій, оснащений радіатором і кулером максимального розміру. Взагалі, якщо вибирати за критерієм надійності, то тут лідирують відеокарти Asus, Msi характеризуються грамотною конструкцією, надійною елементною базою і якісною системою охолодження.

Якщо аналізувати якість модулів оперативної пам'яті від різних виробників, то лідерами будуть такі бренди, як Samsung, Kingston, Hynix, Crcial, goodram продукція яких відрізняється високою надійністю.

Модулі пам'яті зі стандартною частотою представлені двома типами: DDR3, здатний розвивати частоту до 1866 МГц, і більш сучасний - DDR4, що працює на частоті до 2666 МГц. Їх вартість мало відрізняється, тому рекомендується купувати більш сучасний модуль DDR4, тим більше що материнські плати, розроблені під процесори нового покоління, з модулями DDR3 несумісні. Хоча при складанні недорогих комп'ютерів, на які не покладаються супер-завдання, DDR3 можна використовувати.

Об'єм оперативної пам'яті на швидкість роботи комп'ютерної системи не впливає, тому гнатися за модулями високої ємності не має сенсу. Для комп'ютерів офісного призначення цілком достатньо оперативної пам'яті об'ємом 8 Гб. Для універсальних варіантів підбираємо пам'ять об'ємом 8-16 Гб, що дозволить завантажувати і відтворювати ігри. Для ігрових комп'ютерів обсяг модуля оперативної пам'яті повинен бути більшим за 16 Гб. У професійних ігрових комп'ютерах, що працюють на високій потужності, встановлюють пам'ять ємністю 32 Гб.

При збірці комп'ютера одному модулю оперативної пам'яті високої ємності слід віддати перевагу два модуля меншого обсягу. Таке рішення при побудові конфігурації забезпечить можливість роботи ПК в двоканальному режимі доступу до пам'яті, забезпечуючи більш високу продуктивність. Особливо істотний вплив двоканальний режим має на продуктивність відеокарти, вбудованої в процесор.

В цілому сфера продажу комп'ютерної техніки постійно змінюється під впливом нових технологій, змін споживчих уподобань і глобальних економічних тенденцій. Зупинимося на характеристиці основних тенденцій.

1. Зростання попиту на мобільні пристрої. Мобільні пристрої, такі як ноутбуки та планшети, стають все більш популярними завдяки їх портативності та зручності. Це особливо актуально у контексті зростання кількості віддалених робіт і онлайн-освіти, що збільшили попит на портативні пристрої.

2. Розвиток ігрового сегменту. Ігрові комп'ютери та аксесуари, такі як потужні відеокарти, спеціалізовані клавіатури, миші та монітори, користуються великим попитом серед геймерів. Зростання популярності кіберспорту і стрімінгових платформ також стимулює цей сегмент ринку.

3. Впровадження нових технологій. Інновації, такі як штучний інтелект, машинне навчання, віртуальна та доповнена реальність, сприяють розвитку нових продуктів і рішень у сфері комп'ютерної техніки. Ці технології підвищують продуктивність пристроїв і створюють нові можливості для користувачів.

4. Екологічна свідомість. Зростає попит на екологічно чисті та енергоефективні продукти. Виробники активно працюють над зменшенням впливу на навколишнє середовище, пропонуючи продукти з перероблених матеріалів та знижуючи енергоспоживання.

Очікується, що ринок комп'ютерної техніки продовжить зростати у найближчі роки. Основними драйверами цього зростання будуть:

- інновації та технологічний прогрес: подальший розвиток технологій штучного інтелекту, Інтернету речей (IoT), 5G та інших новітніх технологій буде стимулювати попит на нові пристрої та рішення;

- зростання кіберспорту і геймінгу: популярність кіберспорту продовжить зростати, що сприятиме підвищенню попиту на високопродуктивну ігрову техніку та аксесуари;

- зростання екологічної свідомості: виробники будуть приділяти більше уваги розробці екологічно чистих продуктів, що відповідають зростаючому попиту споживачів на стійкі рішення;

- підтримка віддаленої роботи: віддалена робота, яка стала нормою для багатьох компаній, збереже свою актуальність, що стимулюватиме попит на ноутбуки, периферійні пристрої та інструменти для співпраці.

Узагальнюючи, сфера продажу комп'ютерної техніки є однією з найдинамічніших і найбільш перспективних галузей, яка постійно розвивається під впливом нових технологій та змін у споживчих уподобаннях. Високий попит на новітні технологічні рішення забезпечує стабільний розвиток і відкриває нові можливості для гравців на цьому ринку.

1.3 Штучний інтелект у підтримці клієнтів інтернет-магазинів

Протягом останніх років спостерігається експоненційний ріст обсягів електронної торгівлі. З розвитком штучного інтелекту ландшафт електронної комерції обіцяє значно змінитись у найближчі роки. Експонентне зростання

електронної комерції ще більше прискориться через розвиток штучного інтелекту. Завдяки цій технології, що моделює когнітивні процеси людини, інтернет-магазини мають намір змінити купівельну поведінку споживачів.

Щороку у світі відбувається покупок на більш ніж 22 трильйони доларів. В даний час на онлайн-транзакції припадає 11% світової торгівлі. Азіатсько-Тихоокеанський регіон є найбільшим ринком електронної комерції. Де вже в 2016 році обсяг продажів перевищив 1 трильйон доларів. За ним йдуть Європа та Північна Америка, де цей вид торгівлі також став звичайним явищем. Американці витрачають більше одного трильйона доларів на онлайн-купівлі різних предметів повсякденного побуту (одягу, електроніки, товарів для догляду за дітьми, товарів для тварин тощо).

Штучний інтелект – це система, яка використовує та аналізує зібрані дані та на їх основі приймає рішення про ту чи іншу дію [5]. Це стало можливим, серед іншого, завдяки машинному навчанню та когнітивному API. Таким чином, штучний інтелект пропонує користувачам задовольняти свої потреби на основі того, що він уже знає (або дізнався) завдяки іншим користувачам.

Сьогодні все більше сайтів електронної комерції використовують чат-ботів для покращення якості обслуговування клієнтів. Для роздрібних продавців розгортання цих розмовних агентів пов'язане з бажанням персоналізувати взаємодію з користувачами. Деякі програми використовують штучний інтелект для розробки віртуального помічника з покупок. Ґрунтуючись на смаках та бюджеті покупця, ШІ направляє клієнта до товарів, які йому підходять. Крім того, помічник може залишати відгуки та розміщувати нові замовлення. Таким чином, штучний інтелект представлений як засіб створення цифрового досвіду покупця, ідентичного тому, що пропонується у магазинах.

Наведемо кілька прикладів популярних хороших ботів, які використовуються сьогодні у електронній комерції [6].

1. *Пошукові боти*, або павуки – це боти пошукових систем, які сканують та індексують веб-сторінки в Інтернеті. Вони допомагають пошуковим системам

покращувати якість пошуку, отримуючи дані для розуміння структури та релевантності веб-контенту.

2. *Скрейпери* або пошукові веб-скрейпери сканують певний контент в Інтернеті та завантажують його (рис. 1.3). Наприклад, підприємства електронної комерції використовують роботів-скрейперів для відстеження реальних цін на товари на різних торгових майданчиках. Маркетологи використовують скрейперів з можливостями обробки природної мови, щоб аналізувати настрої у соціальних мережах.

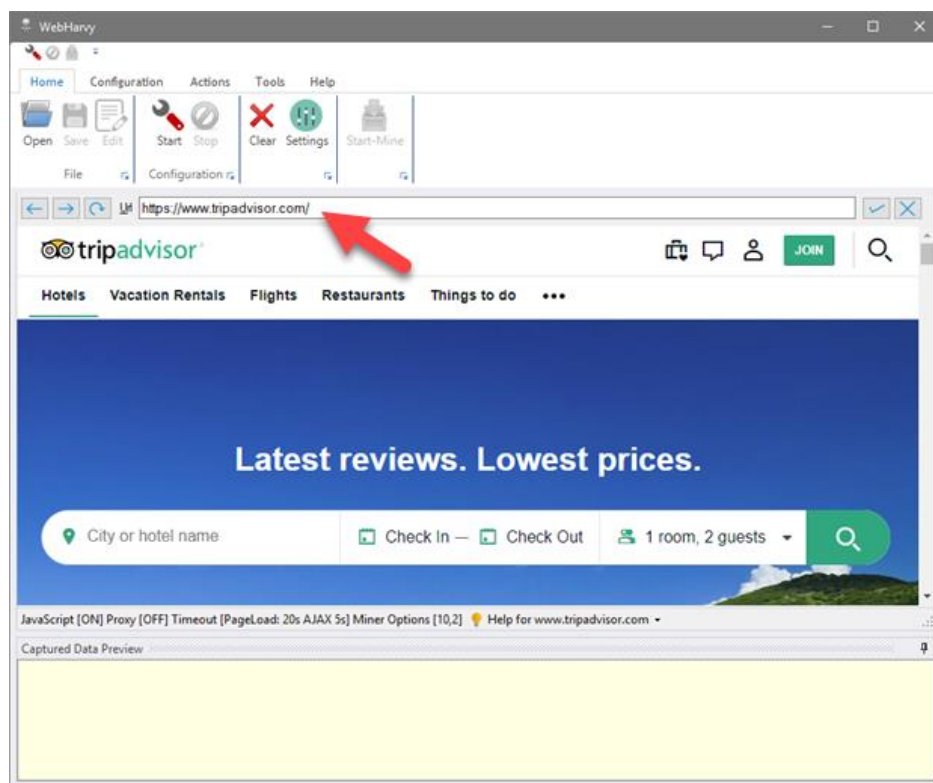


Рисунок 1.3 – WebHarvy – програма-скрейпер

3. *Торгові боти* сканують ціни на товари на кількох сайтах, щоб допомогти покупцям знайти найкращі пропозиції. Торговий бот також може надсилати персональні рекомендації у додатках миттєвих повідомлень.

4. *Моніторингові боти* обмежують вашу схильність до інцидентів безпеки, постійно скануючи системи на наявність помилок і шкідливих програм. Вони попереджають вас про незвичайну веб-активність, збираючи та аналізуючи дані

про взаємодію користувачів та веб-трафік. Деякі моніторингові роботи можуть також працювати разом з іншими ботами, наприклад, чат-ботами, щоб переконатися, що ті працюють так, як задумано.

5. *Транзакційні боти* стежать за тим, щоб платіжні реквізити були в порядку до завершення транзакцій на сайтах електронної комерції. Вони перевіряють реквізити кредитної картки та точність особистих даних під час оформлення замовлення. Ці роботи оснащені високонадійними функціями для захисту конфіденційних фінансових даних.

6. *Чат-боти* (рис. 1.4). Чат-боти імітують людську розмову за допомогою технологій штучного інтелекту та машинного навчання. Вони можуть відповісти на запити від імені служби підтримки клієнтів. Високоінтелектуальні чат-боти, такі як Amazon Alexa можуть природно спілкуватися з людьми.

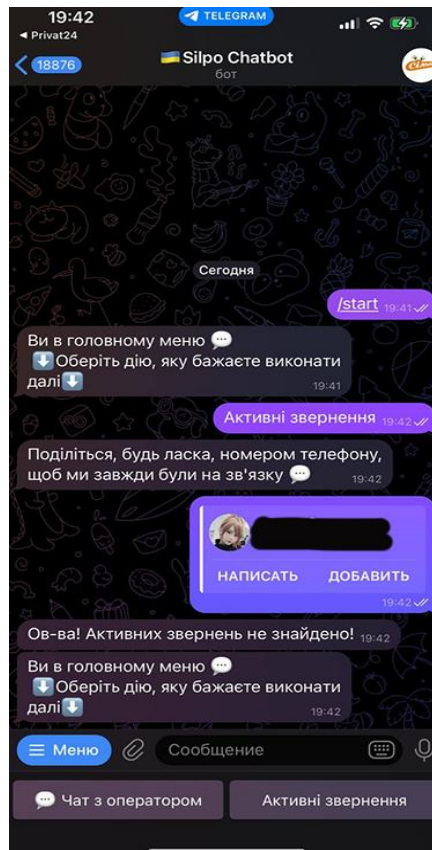


Рисунок 1.4 – Приклад спілкування з чат-ботом «Сільпо»

Більш просунуті роботи – це віртуальні помічники. Ці системи вивчають відповіді, раніше дані іншими користувачами, забезпечуючи більш точний та персоналізований досвід. Найкраща відповідність продуктів і послуг переваг потенційного клієнта є синонімом збільшення продажів.

Актуальність інтернет-ботів в наш час полягає в їхньому потенціалі оптимізувати та полегшувати сферу електронної комерції та онлайн продажу [7]. Ось деякі аспекти, які підкреслюють їхню актуальність:

- ефективність та продуктивність: Чат-боти дозволяють автоматизувати рутинні та повторювані завдання, звільняючи людей від монотонної роботи і забезпечуючи більш ефективне використання їхнього часу. Вони можуть виконувати завдання швидше і надійніше, ніж це можливо для людини;

- покращення обслуговування клієнтів: Чат-боти стають невід'ємною частиною клієнтського обслуговування, забезпечуючи негайну відповідь на запитання та надаючи підтримку у будь-який час доби. Вони допомагають зменшити час очікування для клієнтів і покращують загальний досвід спілкування з компанією;

- аналітика та ринкові дослідження: Чат-боти можуть автоматично збирати, аналізувати та інтерпретувати великі обсяги даних з Інтернету. Це дозволяє компаніям отримувати цінні інсайди щодо своєї аудиторії, конкурентів та ринкових умов;

- застосування в різних сферах: Чат-боти застосовуються в різних галузях, таких як фінанси, медицина, освіта, маркетинг, торгівля та багато інших. Вони можуть бути корисними як для великих корпорацій, так і для малого бізнесу та індивідуальних користувачів;

- розвиток технологій штучного інтелекту: Завдяки постійному розвитку технологій штучного інтелекту, інтернет-боти стають все більш інтелектуальними та здатними до складніших завдань, таких як розпізнавання мови, аналіз тональності текстів, рекомендації тощо.

Інструмент ChatGPT, заснований на штучному інтелекті, став центральною темою у галузі технологій. Google вже готується дати відсіч і оголосила про швидкий випуск власної програми Bard. Використання ШІ дозволяє автоматизувати процеси, спростити планування та заощадити час. В галузі електронної комерції ШІ може особливо допомогти персоналізувати повідомлення, призначені для конкретних клієнтів, чи то у сфері обслуговування клієнтів, чи під час маркетингових кампаній.

ChatGPT – це система від OpenAI, яка генерує відповіді на запитання користувачів – вона працює за принципом чат-бота. Однак ChatGPT не є звичайним чат-ботом [8]. Він має великий потенціал для цифрового маркетингу та SEO. Цей інструмент особливо здатний створювати цілі тексти на задану тему з контенту, доступного в Інтернеті, пропонувати список тем на основі ключових слів, перекладати тексти, отримувати статистику або допомагати створювати метатеги заголовків та описів.

Таким чином, чат-боти вже є дуже поширеним рішенням інтернет-магазинів. Оскільки вони доступні клієнтам 24/7, вони оптимізують процес обслуговування клієнтів. Вони використовують певні питання та адаптують відповіді до питань, сформульованих клієнтами. Їхнє впровадження на сайті економить час і не вимагає великої кількості співробітників в обслуговуванні клієнтів.

При використанні ШІ у інтернет-магазинах неупередженість і швидкість роботи механізмів чат-ботів позитивно позначаються на досвіді користувача, який майже відразу отримує відповідь на свої питання. Надаючи клієнтам персональну пропозицію або контент, заснований на їх перевагах або попередніх покупках, вони з більшою ймовірністю будуть і подалі робити замовлення у інтернет-магазині.

1.4 Постановка задачі

Розробка інтелектуальної системи підтримки клієнтів онлайн магазину комп'ютерної техніки в умовах цифрової трансформації усіх сфер сучасного

суспільства та зростаючого попиту на комп'ютерну техніку дозволяє покращити та персоналізувати взаємодію з клієнтами, роблячи їх обслуговування більш ефективним та результативним.

Об'єкт роботи – процес підтримки клієнтів онлайн-магазину.

Предмет роботи – програмні засоби, технології та методи штучного інтелекту для підтримки клієнтів онлайн-магазину комп'ютерної техніки.

Мета роботи – підвищення ефективності роботи онлайн-магазину комп'ютерної техніки шляхом розробки системи підтримки клієнтів на основі методів штучного інтелекту.

Для досягнення поставленої мети було поставлено такі **завдання**:

- здійснити аналіз предметної сфери продажу комп'ютерної техніки та дослідити теоретичні засади підтримки клієнтів у онлайн-магазинах;
- обґрунтувати вибір технологій та інструментальних засобів розробки системи з використанням методів штучного інтелекту;
- розробити та здійснити програмну реалізацію системи підтримки клієнтів онлайн-магазину комп'ютерної техніки.

Розроблювана інтелектуальна система підтримки клієнтів онлайн-магазину комп'ютерної техніки у вигляді віртуального помічника повинна мати наступні функціональні можливості:

- розуміти запити клієнтів, сформульовані природною мовою у вигляді текстових повідомлень та генерувати відповіді, які відповідають на поставлені запитання;
- надавати клієнтам персоналізовані рекомендації: вичерпну інформацію про наявні у магазині комп'ютерні комплектуючі та їх характеристики, спираючись на дані з бази даних магазину та зовнішніх джерел;
- надавати допомогу у вирішенні проблем, що виникають при підборі комп'ютерних комплектуючих з урахуванням потрібних для клієнта характеристик – помічник може надати клієнту поради та запропонувати альтернативні варіанти вирішення проблем.

Ці вимоги є основою для подальшого проєктування та реалізації системи.

Висновки до розділу 1

У першому розділі було проведено аналіз сфери онлайн продажу комп'ютерної техніки та досліджено потенціал використання штучного інтелекту для покращення взаємодії з клієнтами інтернет-магазинів. Було розглянуто основні проблеми, що виникають при підтримці клієнтів у сфері електронної комерції, специфіку онлайн продажу комп'ютерної техніки, а також проаналізовані існуючі методи та рішення до використання ботів для підтримки клієнтів на основі ШІ. На основі отриманих даних запропоновано концепцію розробки інтелектуальної системи, спрямованої на підвищення ефективності та якості обслуговування клієнтів у сфері онлайн продажу комп'ютерної техніки. Передбачено створення чат-бота віртуального помічника, який спрямований на покращення якості обслуговування клієнтів, підвищення ефективності бізнес-процесів, збільшення продажі за рахунок задоволення потреб споживачів.

2 ТЕХНОЛОГІЇ ТА ЗАСОБИ РОЗРОБКИ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ

2.1 Середовище розробки Visual Studio Code

Visual Studio Code (рис. 2.1) [9] – це безкоштовний, легкий та потужний редактор коду, розроблений компанією Microsoft. Він підтримує широкий спектр мов програмування та технологій, надаючи багату екосистему розширень та інструментів для розробників.

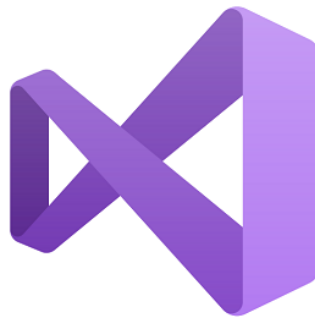


Рисунок 2.1 – Логотип середовища розробки Visual Studio

Однією з ключових особливостей VS Code є його кроссплатформенність: він підтримується на Windows, macOS та Linux, пропонуючи однаковий інтерфейс та функціональність на всіх платформах. Це робить його зручним вибором для розробників, які працюють у різних операційних системах.

VS Code має великий каталог розширень, доступний через вбудований Marketplace. Ці розширення дозволяють додавати підтримку різних мов, інструментів та інтеграцій, значно розширюючи можливості редактора. Користувачі можуть тонко налаштовувати редактор через JSON файли конфігурації, що дозволяє адаптувати його під свої конкретні потреби.

Редактор надає вбудовану підтримку систем контролю версій, таких як Git. Зручний інтерфейс дозволяє виконувати основні операції з Git прямо з редактора, такі як commit, pull, push, branch та merge. Це спрощує процес управління версіями та робить роботу з репозиторіями зручнішою.

Налагодження VS Code підтримує безліч мов програмування. Розробники можуть налаштовувати конфігурації налагодження, ставити точки зупинки, крок за кроком проходити код, переглядати змінні та виклики стека. Це робить процес налагодження ефективним та зручним.

Однією з найпотужніших функцій VS Code є інтелектуальне підсвічування синтаксису та автодоповнення. Підтримка IntelliSense забезпечує автодоповнення коду, пропозиції щодо завершення, інформацію про параметри функцій та документацію щодо API. Це значно прискорює процес написання коду та зменшує кількість помилок.

Вбудований термінал VS Code дозволяє виконувати команди оболонки прямо з редактора. Підтримка кількох терміналів та їх налаштування роблять роботу з командним рядком зручнішим та ефективнішим.

Функція Live Share дозволяє працювати над кодом у реальному часі. Розробники можуть спільно редагувати та налагоджувати код, обмінюватися терміналом та серверами. Це робить VS Code чудовим інструментом для командної роботи та парного програмування.

VS Code підтримує Language Server Protocol (LSP), що дозволяє додавати інтелектуальні функції для різних мов програмування. Це дозволяє розробникам створювати сервери мови для покращення підсвічування синтаксису, автодоповнення та інших функцій, що робить роботу з різними мовами більш продуктивною.

2.2 Засоби розробки інтерфейсу користувача

HTML (HyperText Markup Language) є основою для створення веб-сторінок та веб-додатків (рис. 2.2). Це мова розмітки, яка використовується для структурування та відображення контенту у браузерях. HTML визначає структуру сторінки за допомогою елементів або тегів, які оточують різні частини контенту та визначають їхню роль і значення [10].



Рисунок 2.2 – Логотип гіпертекстової розмітки HTML

Основною одиницею розмітки в HTML є елемент. Елемент складається з відкриваючого тегу, контенту та закриваючого тегу. Відкриваючий тег містить ім'я елемента, яке вказує браузеру, як повинен відобразитися контент, а закриваючий тег вказує кінець цього контенту. Наприклад, `<p>` - відкриваючий тег для абзацу, а `</p>` - закриваючий тег. HTML також дозволяє вставляти атрибути до елементів, які надають додаткову інформацію про елемент. Наприклад, атрибут `src` використовується для визначення джерела зображення в елементі ``, а атрибут `href` використовується для визначення посилання в елементі `<a>`.

HTML5 включає нові мультимедійні теги `<video>` та `<audio>`, які дозволяють вбудовувати відео та аудіо без необхідності у використанні сторонніх плагінів. Це значно спрощує додавання мультимедійного контенту на веб-сторінки. Тег `<canvas>` забезпечує можливість малювання 2D графіки за допомогою JavaScript, відкриваючи широкі можливості для створення інтерактивних ігор та анімацій.

Зростання використання мобільних пристроїв для перегляду веб-сторінок робить адаптивний дизайн обов'язковим. HTML і CSS дозволяють створювати адаптивні веб-сайти, які зручно переглядати на різних пристроях, включаючи смартфони і планшети. Це включає в себе використання медіа-запитів у CSS для налаштування стилів в залежності від розміру екрану.

Дизайн і структура HTML-документа впливають на сприйняття безпеки сайту користувачами. Наявність SSL-сертифіката, чіткі політики конфіденційності та захисту даних, а також надійна інформація про оплату і доставку створюють відчуття безпеки у користувачів. Сучасний та професійний вигляд сайту підвищує

довіру до магазину і зменшує побоювання щодо шахрайства або проблем з оплатою.

HTML є фундаментом для створення веб-сторінок. Від простих сайтів до складних веб-додатків, HTML забезпечує основу для структурування та представлення контенту. Розуміння основних принципів та можливостей HTML, включаючи семантичну розмітку та нові функції HTML5, є критично важливим для ефективної веб-розробки. Інвестування в професійний, інтуїтивно зрозумілий і привабливий дизайн допомагає виділитися серед конкурентів, залучити і утримати клієнтів, а також забезпечити стабільне зростання продажів.

CSS (Cascading Style Sheets)– це мова стилів, яка використовується для задання зовнішнього вигляду веб-сторінок (рис. 2.3). Вона дозволяє веб-розробникам визначати різні аспекти візуального представлення елементів HTML, такі як кольори, шрифти, розміри, відступи, рамки та позиціонування. CSS відокремлює структуру та вміст сторінки від її вигляду, що дозволяє змінювати оформлення сторінки без зміни її вмісту [11].



Рисунок 2.3 – Логотип CSS.

Однією з ключових особливостей CSS є його каскадна природа. Це означає, що стилі можуть бути задані на різних рівнях (наприклад, в середині HTML-тегів, в блоках `<style>`, у зовнішніх CSS-файлах), і вони будуть застосовуватися в порядку каскаду та згідно з правилами спадкування, що дозволяє зручно керувати оформленням сторінки. CSS використовує селектори для вибору елементів HTML, до яких будуть застосовані стилі. Селектори можуть бути простими (наприклад,

назва елемента або класу) або складними (наприклад, комбінація класів, псевдокласів або атрибутів). Це дозволяє точно визначати, які елементи потребують зміни стилів та як саме ці зміни повинні виглядати.

Крім того, CSS підтримує різні методи розміщення елементів на сторінці, включаючи блочну модель, рядкову модель, гнучкі та сіткові системи. Це дозволяє створювати різноманітні макети сторінок з використанням різних типів контейнерів та позиціонування елементів.

Узагальнюючи, CSS є важливою складовою веб-розробки, яка дозволяє створювати привабливий та зручний візуальний дизайн для веб-сторінок. Вона відокремлює структуру та зміст від представлення, що сприяє покращенню модульності та підтримці.

2.3 JavaScript та бібліотека React

JavaScript (рис. 2.4) – це високорівнева, інтерпретована мова програмування, яка використовується для створення динамічного контенту на веб-сторінках [12]. Вона виконується безпосередньо у браузері користувача, що дозволяє додавати інтерактивність до веб-сайтів. JavaScript є однією з трьох основних технологій веб-розробки, разом з HTML і CSS, і відіграє ключову роль у реалізації клієнтської частини веб-застосунків.



Рисунок 2.4 – Логотип JavaScript

JavaScript є мовою сценаріїв, що означає, що його код виконується рядок за рядком в реальному часі. Основними будівельними блоками JavaScript є змінні, оператори, функції та об'єкти. Змінні використовуються для зберігання даних, а функції – для виконання певних завдань. JavaScript підтримує об'єктно-орієнтований підхід, що дозволяє розробникам створювати складні структури даних та функціональні компоненти.

Однією з особливостей JavaScript є його динамічність та гнучкість. Він дозволяє додавати нові елементи до сторінки, змінювати властивості та стилі елементів, обробляти події, взаємодіяти з користувачем та виконувати багато інших дій у реальному часі без перезавантаження сторінки.

JavaScript є мовою з об'єктно-орієнтованим підходом до програмування. Вона підтримує об'єктно-орієнтовані концепції, такі як класи, об'єкти, наслідування, інкапсуляція та поліморфізм [13]. Це дозволяє створювати складні програми шляхом організації коду у відповідності до об'єктно-орієнтованих принципів. JavaScript також підтримує функціональне програмування. Він дозволяє використовувати функції як об'єкти першого класу, передавати їх як аргументи, повертати з інших функцій та зберігати в змінних. Це відкриває широкі можливості для створення складних функціональних програм та реалізації різних парадигм програмування.

JavaScript тісно інтегрується з HTML і CSS, що дозволяє змінювати DOM (Document Object Model) — структуру веб-сторінки. Використовуючи методи, такі як `getElementById`, `querySelector` і `addEventListener`, розробники можуть динамічно змінювати вміст та стиль елементів, а також обробляти події, такі як натискання кнопок або зміна вмісту форм.

JavaScript має багату екосистему бібліотек та фреймворків, які значно спрощують розробку веб-додатків. Серед найбільш популярних бібліотек — `jQuery`, яка спрощує роботу з DOM і подіями. Веб-фреймворки, такі як `React`, `Angular` і `Vue.js`, забезпечують структурований підхід до розробки складних веб-

додатків, дозволяючи створювати компонентні системи, керувати станом додатка і оптимізувати продуктивність.

Завдяки Node.js, JavaScript використовується не тільки на стороні клієнта, але й на сервері. Node.js – це середовище розробки JavaScript, яке дозволяє створювати швидкі і масштабовані серверні додатки. Node.js базується на движку V8 від Google і використовує неблокуючу архітектуру вводу-виводу, що дозволяє обробляти велику кількість одночасних запитів. Це робить його ідеальним для розробки реального часу додатків, таких як чати, онлайн-ігри і системи обробки транзакцій.

JavaScript підтримує асинхронне програмування, що дозволяє виконувати кілька завдань одночасно, не блокуючи головний потік виконання. Це особливо важливо для веб-додатків, які часто виконують асинхронні операції, такі як запити до серверу або робота з файлами. Основними механізмами асинхронного програмування є колбеки, проміси та `async/await`.

Загалом, JavaScript є потужною та універсальною мовою програмування, яка використовується для реалізації різноманітних функцій на веб-сторінках. Вона дозволяє створювати динамічні та інтерактивні веб-додатки, що забезпечує зручне та привабливе користувацьке досвід.

React – це бібліотека JavaScript для розробки інтерфейсів користувача [14]. Вона була розроблена компанією Facebook і стала однією з найпопулярніших технологій у веб-розробці. Основна концепція React полягає у використанні компонентів, які є будівельними блоками для створення веб-інтерфейсів. Компоненти в React є невеликими, автономними та повторно використовуваними модулями, які можна збирати в складніші структури.

Однією з ключових особливостей React є віртуальний DOM (Document Object Model). React використовує віртуальний DOM для оптимізації роботи з реальним DOM. Він створює віртуальне представлення сторінки, яке зберігається у пам'яті, і порівнює його з реальним DOM для визначення мінімальних змін, які потрібно

зробити для оновлення сторінки. Це дозволяє реалізувати швидку та ефективну роботу з інтерфейсом, навіть у великих і складних додатках.

Ще однією важливою концепцією в React є односторонній потік даних. Дані в React передаються вниз по ієрархії компонентів зверху вниз, що дозволяє створювати простіше та прогнозованіше керовану структуру даних. Це робить код React більш читабельним та легшим у супроводі, оскільки кожен компонент відповідає лише за свої власні дані та відображення.

Ще однією важливою особливістю React є можливість використання JSX (JavaScript XML). JSX - це розширення JavaScript, яке дозволяє використовувати синтаксис, подібний до HTML, для опису структури інтерфейсу. Це робить код React більш зрозумілим та експресивним, особливо для великих та складних інтерфейсів.

Props — це вхідні дані для компонентів, які передаються від батьківського компонента до дочірнього. Вони є незмінними і забезпечують передачу даних вниз по дереву компонентів. State, з іншого боку, є внутрішнім станом компонента, який може змінюватися з часом. Зміни в стані компонента призводять до повторного рендерингу компонента, що дозволяє оновлювати інтерфейс у відповідь на дії користувача або інші події.

Компоненти React мають життєвий цикл, що складається з різних етапів: монтаж, оновлення та демонтаж [15]. Під час кожного з цих етапів викликаються спеціальні методи життєвого циклу, які дозволяють розробникам виконувати додаткові дії, такі як ініціалізація стану, завантаження даних або очищення ресурсів. Важливі методи життєвого циклу включають `componentDidMount`, `componentDidUpdate` та `componentWillUnmount`.

Для великих додатків управління станом може стати складним завданням. React забезпечує кілька підходів для цього, включаючи використання локального стану компонентів, контексту (Context API) для передачі даних через дерево компонентів та зовнішніх бібліотек, таких як Redux або MobX. Redux є популярним інструментом для керування глобальним станом додатка, що дозволяє

організувати стан у вигляді єдиного дерева і використовувати ред'юсери для обробки змін стану.

React підтримує серверний рендеринг (Server-Side Rendering, SSR) і статичну генерацію сайтів (Static Site Generation, SSG) завдяки таким фреймворкам, як Next.js. SSR дозволяє рендерити React-компоненти на сервері і відправляти готовий HTML-код на клієнт, що покращує SEO та швидкість завантаження сторінок. SSG дозволяє заздалегідь генерувати HTML для всіх сторінок під час збірки, що робить їх доступними миттєво під час запиту.

Узагальнюючи, React є потужною та популярною бібліотекою для розробки інтерфейсів користувача, яка дозволяє створювати ефективні, прогнозовані та легко супроводжувані веб-додатки. Він пропонує набір інструментів та підходів для розробки модульних та перевикористовуваних інтерфейсів, що робить його відмінним вибором для будь-якого розробника веб-додатків.

2.4 Протокол зв'язку WebSocket

WebSocket – це двонаправлений повнодуплексний протокол зв'язку між клієнтом та сервером [16]. На відміну від HTTP-протоколу, який працює за принципом «запит від клієнта – відповідь від сервера», у вебсокетах і сервер, і клієнт можуть надсилати один одному повідомлення (рис. 2.5). Кожна сторона комунікації здатна одночасно отримувати та відправляти дані.

У вебсокетах обмін повідомленнями відбувається через єдиний канал зв'язку. Він залишається відкритим протягом усієї комунікації, а за необхідності будь-яка зі сторін може його закрити. Основна перевага вебсокетів полягає в тому, що після встановлення з'єднання воно залишається відкритим, що дозволяє відправляти та отримувати дані з мінімальною затримкою. Це значно знижує накладні витрати у порівнянні з традиційними HTTP-запитами, які потребують встановлення нового з'єднання для кожного обміну даними.

Вебсокети знайшли широке застосування у інтерактивних вебзастосунках, які у режимі реального часу підтримують зв'язок [17]. Наприклад, вони використовуються в чат-додатках для миттєвого обміну повідомленнями між користувачами, в іграх для забезпечення низької затримки обміну даними, а також для відправки миттєвих сповіщень користувачам про нові події чи повідомлення.



Рисунок 2.5 – Порівняння WebSocket та HTTP

Крім того, вебсокети використовуються в інтерактивних вебзастосунках, які потребують постійного оновлення даних, таких як фінансові системи або додатки для моніторингу. Це робить їх важливим інструментом для розробників, що створюють застосунки, які потребують високої продуктивності і ефективності при обміні даними в реальному часі.

2.5 Мова програмування Python та Python-бібліотеки

Використання мови програмування Python та її бібліотек є ключовим для розробки застосунку для підтримки клієнтів інтернет-магазину.

Python – це мова програмування високого рівня, яка визнана другою за популярністю у світі [18]. Вона використовується для створення веб-застосунків,

та реалізації методів і моделей машинного навчання. Python широко застосовується для розв'язання завдань у таких компаніях, як Google, Instagram, Facebook, IBM, NASA, Dropbox, Netflix та інші. Програмісти цінують цю мову програмування через її простоту навчання, продуктивність та підтримку багатьох платформ.

Python є мовою об'єктно-орієнтованого програмування (ООП) [19]. Програмне забезпечення на Python оформлене у вигляді моделей, які можуть бути зібраними у пакети. Тип та структуру кожного об'єкта можна запитати під час виконання програми. Для кожного з об'єктів можна отримати всю інформацію щодо його внутрішньої структури. Окрім того:

- у мови логічний синтаксис, завдяки чому вихідний код легко читати та розуміти;
- гнучкість та масштабованість Python дозволяє адаптувати високорівневу логіку та розширяти складні застосунки, як тільки виникне така необхідність;
- розробка на Python у більшості випадків проходить швидше, ніж на інших мовах програмування;
- Python – інтерпретована мова програмування, це означає, що код можна написати у будь-якому текстовому файлі на будь-якій платформі, і потім успішно запустити;
- у Python – колосальна спільнота однодумців, будь-які складнощі конкретних розробників вирішуються колективно.

Проте є декілька особливостей, які можна віднести до недоліків. Це повільність (ця мова програмування хоч і універсальна, проте повільніша за інші), велика кількість ресурсів, необхідних для роботи та «прив'язаність» до системних бібліотек. Pandas – це бібліотека для роботи з даними на Python [20]. Вона спрощує життя аналітикам: де раніше використовувалося 10 рядків коду тепер вистачить однієї.

Бібліотека Pandas є однією з найпопулярніших і широко використовуваних бібліотек для роботи з даними у мовах програмування Python. Вона забезпечує потужні та гнучкі інструменти для аналізу, маніпуляції та візуалізації даних, що

робить її незамінною у сфері науки про дані, машинного навчання та бізнес-аналітики. Pandas надає структури даних на високому рівні, такі як DataFrame та Series, які дозволяють ефективно працювати з табличними даними, а також великий набір функцій для очищення, обробки та перетворення даних.

Однією з головних переваг Pandas є його здатність інтегруватися з іншими бібліотеками Python, такими як NumPy, Matplotlib і Scikit-Learn, що забезпечує зручний і комплексний підхід до аналізу даних. DataFrame, основна структура даних Pandas, дозволяє працювати з даними, що мають різні типи (числові, текстові, датовані), та надає широкий спектр методів для фільтрації, сортування, групування та агрегування даних. Це робить бібліотеку Pandas ідеальним вибором для обробки та аналізу великих обсягів даних.

Працюючи з Pandas, користувачі можуть легко імпортувати дані з різних джерел, таких як CSV, Excel, SQL-бази даних та інші. Крім того, бібліотека підтримує роботу з відсутніми даними, що є важливим аспектом при підготовці даних для аналізу або моделювання. Pandas також забезпечує ефективні методи для роботи з часовими рядами, що робить її корисною для аналізу фінансових даних, моніторингу та прогнозування.

В цілому, Pandas є потужним інструментом для кожного, хто працює з даними в Python. Завдяки її гнучкості, швидкодії та широкому набору функцій, вона допомагає спеціалістам ефективно аналізувати та маніпулювати даними, що призводить до більш точних та обґрунтованих рішень у різних галузях.

NumPy (англ. Numerical Python) – це одна з найкращих бібліотек Python із відкритим вихідним кодом, яка є універсальним стандартом для роботи з числовими даними в Python, що лежить в основі наукових екосистем Python і PyData [21]. Користувачі NumPy включають усіх, від починаючих програмістів до досвідчених дослідників, які займаються сучасними науковими та промисловими дослідженнями та розробками.

Бібліотека NumPy містить багатомірні масиви та матричні структури даних. NumPy можна використовувати для виконання безлічі математических операцій з

масивами. Вона додає в Python потужні структури даних, що гарантують ефективне вичислення з масивами та матрицями, і надає величезну бібліотеку високорівневих математичних функцій, які працюють з цими масивами та матрицями.

Ось кілька областей, де активно використовується NumPy:

- наукові та інженерні обчислення;
- машинне навчання та штучний інтелект;
- обробка даних;
- візуалізація даних;
- високопродуктивні обчислення.

Масив – це центральна структура даних бібліотеки NumPy. Масив є сіткою значень і містить інформацію про необроблені дані, про те, як знайти елемент і як його інтерпретувати. Він має сітку елементів, які можна індексувати різними способами. Всі елементи одного типу називають масивом «dtype».

Масиви NumPy швидші та компактніші, ніж списки Python. Масив споживає менше пам'яті (що дуже важливо для оптимізації швидкості обчислень або потужності, що використовується) і зручний у використанні. NumPy використовує набагато менше пам'яті для зберігання даних та надає механізм вказівки типів даних. Це дозволяє ще більше оптимізувати код. Масив може бути проіндексований кортежем невід'ємних цілих чисел, логічними значеннями, іншим масивом чи цілими числами.

2.6 Фреймворк Llamaindex

Llamaindex – це фреймворк, який надає зручні інструменти для роботи з великими моделями мови [22]. Він дозволяє індексувати великі обсяги текстових даних та виконувати за ними ефективний пошук. Це особливо корисно під час роботи з великими мовними моделями, які можуть генерувати текст на основі даного контексту.

Основна мета Llamaindex – полегшити роботу з мовними моделями, надавши зручні інструменти для індексування та пошуку. Це дозволяє розробникам зосередитися на створенні додатків та сервісів, які використовують мовні моделі, замість витратити час та ресурси на вирішення технічних проблем.

Llamaindex заснований на концепції Retrieval Augmented Generation – RAG. В основі RAG лежить ідея про те, що для генерації відповіді на запит можна використовувати не лише контекст запиту, а й додаткові дані, знайдені індексом. Це дозволяє генерувати більш точні та інформативні відповіді.

В рамках flashindex основні елементи RAG представлені наступними модулями:

- Data Connectors: модулі, які відповідають за завантаження та обробку даних, вони дозволяють завантажити дані, створити з них вузли та побудувати індекс;
- Retrievers: це модулі, які відповідають за пошук за індексом вони дозволяють виконати пошук за індексом та отримати найбільш релевантні вузли;
- Query Engines: це модулі, які відповідають за створення відповіді на запит. Вони використовують інформацію, отриману від retrievers, для створення відповіді.

У LlamaIndex є різні типи індексів, кожен із яких працює трохи по-різному. Наприклад, індекс векторного сховища VectorStoreIndex зберігає кожен вузол і відповідне векторне уявлення, і при пошуку повертає топ-k найбільш схожих вузлів. Інший тип індексу, індекс таблиці ключових слів KeywordTableIndex, отримує ключові слова з кожного вузла і будує відображення від кожного ключового слова до відповідних вузлів цього ключового слова. Під час пошуку він витягує релевантні ключові слова із запиту та шукає відповідні вузли.

При розробці системи підтримки інтернет-магазину комп'ютерної техніки було вирішено використовувати цей фреймворк для роботи з мовними моделями.

Висновки до розділу 2

У цьому розділі було розглянуто та обґрунтовано вибір технологій та інструментальних засобів розробки інтелектуальної системи підтримки клієнтів онлайн-магазину комп'ютерної техніки, застосування яких дозволяє створити віртуальний помічник – чат-бот для підтримки клієнтів при виборі комп'ютерних комплектуючих. Що є важливими інструментами для підвищення ефективності бізнесу та задоволення потреб споживачів.

У якості середовища розробки обрано IDE Visual Studio Code, яке забезпечує зручний і ефективний процес розробки завдяки підтримці різноманітних мов програмування та розширень і мову програмування Python, яка надає широкі можливості для розробки систем штучного інтелекту а також інтеграції з системами контролю версій, такими як Git. HTML та CSS використовуються для створення структури та стилізації веб-сторінок. HTML визначає структуру документа за допомогою елементів та атрибутів, тоді як CSS надає можливість задавати зовнішній вигляд цих елементів, включаючи кольори, шрифти, розміри та позиціонування. Використання цих технологій забезпечує створення адаптивних, естетично привабливих та зручних для користувача інтерфейсів.

Для забезпечення динамічності та інтерактивності хостосунку магазину обрано JavaScript та бібліотеку React. JavaScript та бібліотека React відіграють ключову роль у реалізації динамічних функцій та інтерактивності на веб-сторінках. JavaScript дозволяє додавати нові елементи до сторінки, змінювати властивості та стилі елементів, обробляти події та взаємодіяти з користувачем у реальному часі. React забезпечує ефективне створення компонентів користувацького інтерфейсу та їх повторне використання, що спрощує розробку складних додатків та підвищує їх продуктивність.

Python та його бібліотеки, такі як NumPy, Pandas, та фреймворк LlamaIndex, були використані для обробки даних та реалізації алгоритмів штучного інтелекту. NumPy надає потужні інструменти для роботи з багатовимірними масивами та

виконанням високопродуктивних математичних операцій. Pandas забезпечує зручний спосіб обробки та аналізу табличних даних, включаючи фільтрацію, агрегацію та візуалізацію. LlamaIndex використовується для створення індексів та пошуку по великим обсягам текстових даних, що дозволяє ефективно реалізувати функції інтелектуального пошуку та рекомендацій.

Вибір технологій та інструментальних засобів, розглянутий у цьому розділі, забезпечує основу для подальшої розробки та успішної реалізації інтелектуальної системи підтримки клієнтів онлайн-магазину комп'ютерної техніки.

3 ПРОЄКТУВАННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ПІДТРИМКИ КЛІЄНТІВ ОНЛАЙН-МАГАЗИНУ КОМП'ЮТЕРНОЇ ТЕХНІКИ

3.1 Основні етапи створення системи

Проєктування інтелектуальної системи підтримки клієнтів онлайн-магазину комп'ютерної техніки включало декілька основних етапів. На першому етапі було визначено вимоги до системи, що включають функціональні можливості, такі як розуміння запитів клієнтів, сформульованих природною мовою у вигляді текстових повідомлень та генерація відповіді на поставлені запитання. А також надання клієнтам вичерпної інформації про наявні у магазині комплектуючі та їх характеристики. І надання допомоги клієнтам у вирішенні проблем, які виникають при підборі комп'ютерних комплектуючих у вигляді порад та пропозицій альтернативних варіантів.

На другому етапі було розроблено користувацький інтерфейс, який включає в себе структурні компоненти вебсторінки інтернет-магазину, створені за допомогою HTML, та візуальні стилі, реалізовані за допомогою CSS. Для додання інтерактивності та динамічності використовувались JavaScript та бібліотека React, що дозволило забезпечити зручну та ефективну взаємодію користувачів із системою.

Третій етап передбачав проєктування та програмування інтелектуальної складової системи. Використання бібліотек Python, таких як NumPy, Pandas та фреймворку LlamaIndex, дозволило ефективно обробляти дані та забезпечувати належне функціонування інтелектуальних алгоритмів системи [23]. Це дало змогу надавати користувачам персоналізовані рекомендації та швидко відповідати на їхні запити.

На останньому етапі було здійснено тестування роботи чат-бота, який інтегрується в систему для забезпечення підтримки клієнтів у реальному часі. Це

забезпечило тестування функціональних можливостей системи та її готовність до використання в реальних умовах роботи онлайн-магазину комп'ютерної техніки.

Таким чином, реалізація інтелектуальної системи підтримки клієнтів включала комплексний підхід до проектування, що охоплює аналіз вимог, розробку користувацького інтерфейсу, програмування інтелектуальних компонентів та тестування системи в умовах реального часу.

3.2 Проектування та програмна розробка інтерфейсу інтернет-магазину

Основна сторінка розробленої системи в загальній проекції буде мати такий вигляд (рис. 3.1, рис. 3.2): шапка сторінки (header), презентація сайту, фільтрація товарів, перелік товарів та нижня частина (footer).

Програмна реалізація здійснювалася у середовищі розробки Visual Studio Code з використанням мови програмування JavaScript та бібліотеки React. Основним компонентом системи є App.js який дозволяє функціонувати з товарами та виводить їх на екран.

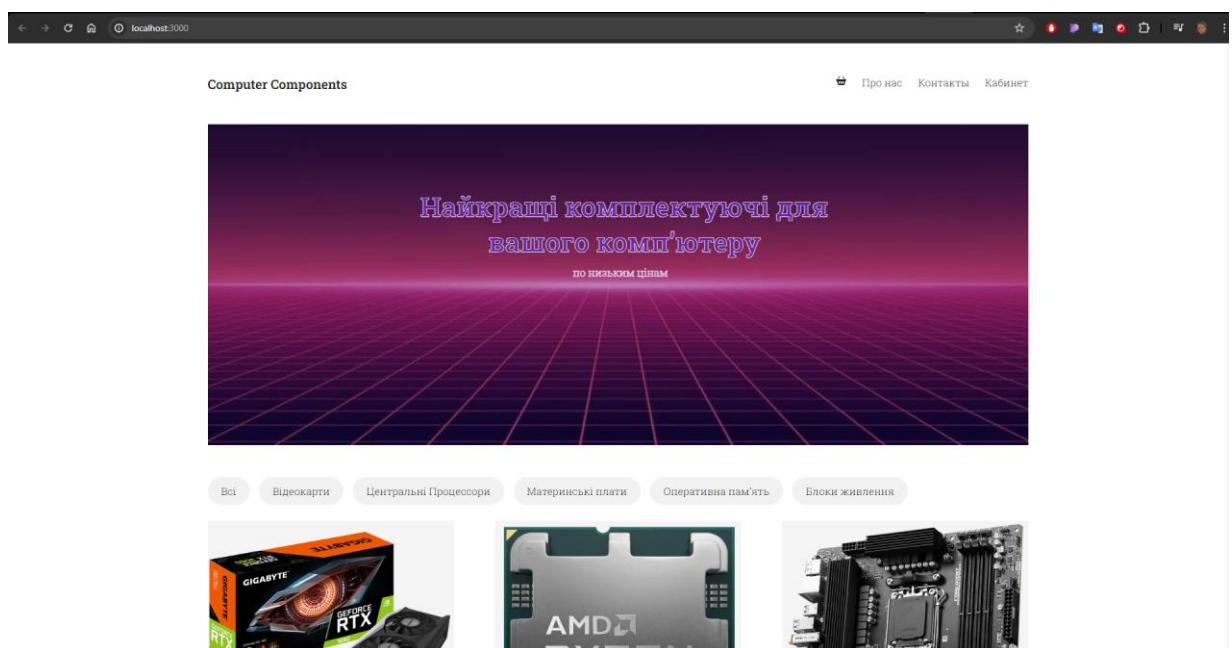


Рисунок 3.1 – Загальний вигляд початкової сторінки магазину

Фільтрація товарів в інтернет-магазині необхідна для покращення користувацького досвіду, оскільки вона дозволяє покупцям швидко знаходити продукти, що відповідають їхнім конкретним потребам та критеріям, таким як ціна, бренд, розмір, колір, рейтинг та інші характеристики. Це зменшує час пошуку, підвищує задоволеність клієнтів і може збільшити конверсію, оскільки користувачі легше знаходять і купують потрібні їм товари.

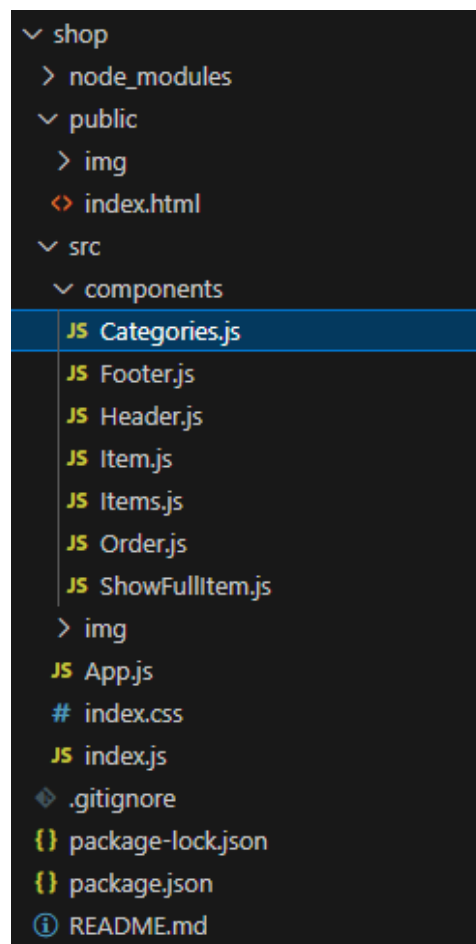


Рисунок 3.2 – Загальна структура вебзастосунку

У розробленому веб-застосунку фільтрація реалізовано таким чином: при натисканні на кнопку вибору конкретного компоненту/частини комп'ютера, зникають усі товари окрім вибраного нами типу. Наприклад при натисканні на кнопку «Відеокарти», на екрані виведуться усі відеокарти в наявності (рис. 3.3).

Фільтрація товарів відбувається у компоненті `Categories.js`. У ньому зберігаються кнопки при натисканні на котрі показуються товари з певної категорії та обробник натискання котрий містить анонімну функцію, яка містить метод `chooseCategory` (рис. 3.4). В головному файлі `App.js` міститься метод `chooseCategory`, котрий містить в собі вбудований метод `filter` та фільтрує товари по їх унікальному номеру, або іншим словом `id`.

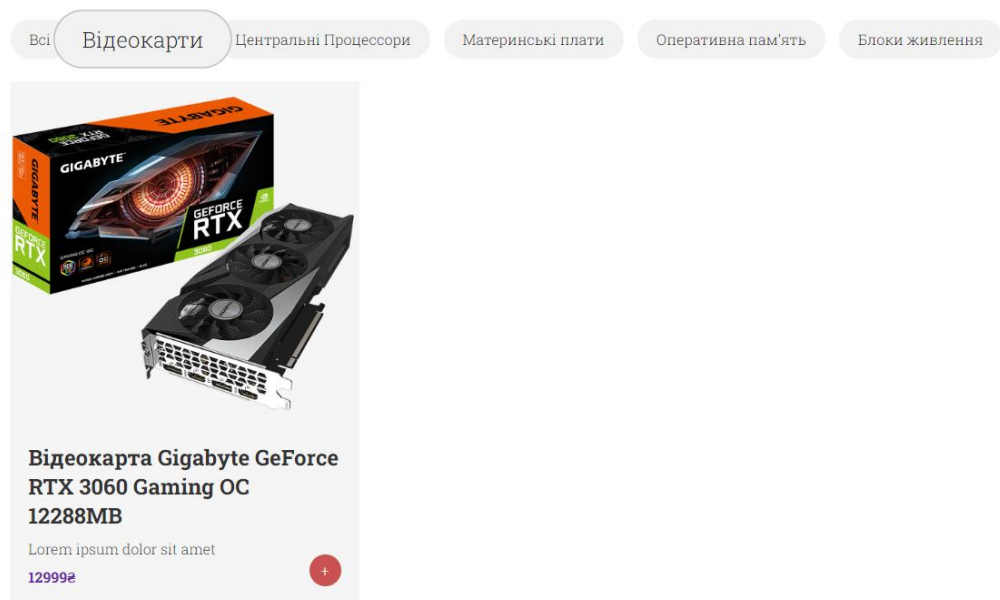


Рисунок 3.3 – Зразок використання фільтрації на прикладі натискання на кнопку «Відеокарти»

```
chooseCategory(category) {  
  if( category === 'all' ) {  
    this.setState({currentItems:this.state.items})  
    return  
  }  
  this.setState({  
    currentItems: this.state.items.filter(el => el.category === category)  
  })  
}
```

Рисунок 3.4 – Метод `chooseCategory` для здійснення фільтрації товарів

Кошик. Кошик в інтернет-магазині є ключовою функцією, яка дозволяє користувачам зручно збирати вибрані товари перед покупкою, переглядати їхній загальний кошт, коригувати кількість або видаляти непотрібні позиції (рис. 3.5). Це забезпечує спрощений і організований процес покупки, сприяє більшій залученості клієнтів, полегшує управління замовленнями і підвищує ймовірність завершення транзакції, тим самим збільшуючи конверсію і задоволеність клієнтів.

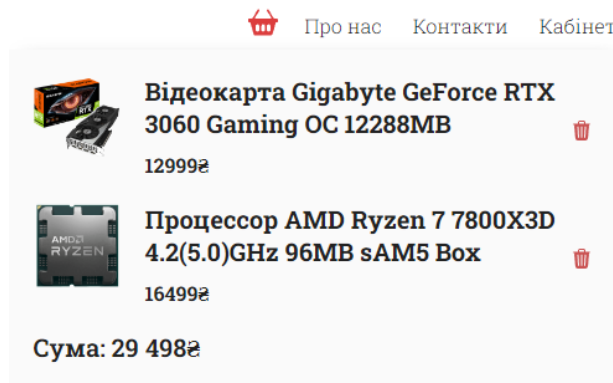


Рисунок 3.5 – Загальний вигляд кошика і відображення товарів

Додавання товарів у кошик відбувається за рахунок методу `addToOrder` (рис. 3.6). Разом з товаром створюється спеціальна кнопка, при натисканні на котру викликається метод `addToOrder`, та він додає товар у кошик. Компонент `Order.js` буде виводити товари на екран. Також в кошику буде рахуватись сума всіх товарів котрі туди було додано.

```
addToOrder(item) {
  let isInArray = false
  this.state.orders.forEach(el => {
    if(el.id === item.id)
      isInArray = true
  })
  if(!isInArray)
    this.setState({orders: [...this.state.orders, item] })
}
```

Рисунок 3.6 – Метод `addToOrder`

Відкриття корзини відбувається при натисканні на відповідну кнопку. Видалення товарів відбувається в методі `deleteOrder` (рис. 3.7). При натисканні на відповідну кнопку котра знаходиться поряд із товаром у кошику викликається цей самий метод і товар прибирається з кошику.

```
deleteOrder(id){  
  this.setState({orders: this.state.orders.filter(el => el.id !== id)})  
}
```

Рисунок 3.7 – Метод `deleteOrder`

Сторінка з товарами. Виведення кожного товару окремо в інтернет-магазині важливе для забезпечення повної інформації користувачам про кожну позицію (рис. 3.8). Це дозволяє покупцям детально розглянути характеристики, властивості, фотографії та іншу важливу інформацію про товар перед придбанням. Окремий вигляд для кожного товару допомагає збільшити прозорість, довіру та комфорт користувачів під час прийняття рішення про покупку, що може позитивно вплинути на їхню участь у магазині та зростання конверсії.

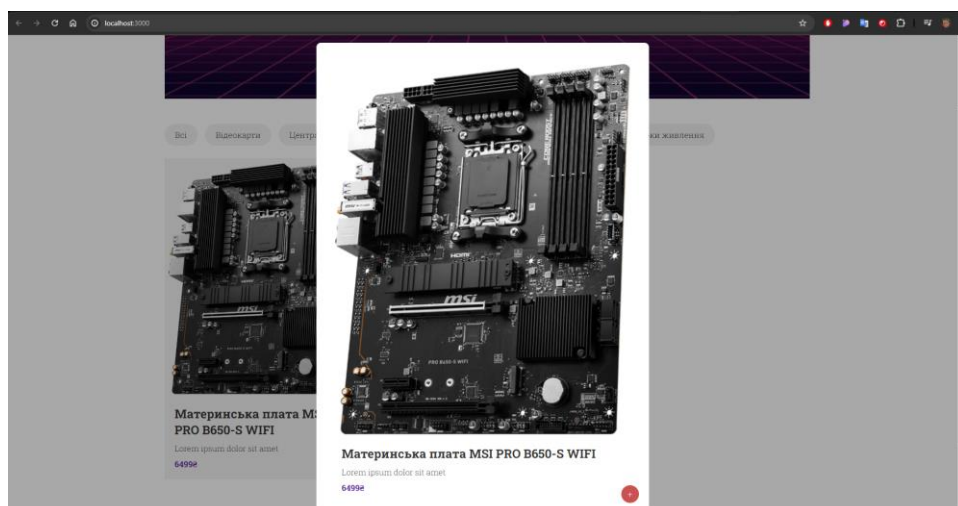


Рисунок 3.8 – Відображення товару окремою сторінкою

Відкриття товару відбувається при натисканні на картинку з самим товаром (рис. 3.9). Відбувається це за рахунок методу `onShowItem`. Саме відображення вмикається в компоненті `ShowFullItem.js`.

```
onShowItem(item) {  
  this.setState({fullItem: item})  
  this.setState({ShowFullItem: !this.state.ShowFullItem})  
}
```

Рисунок 3.9 – Метод `onShowItem`

3.3 Програмна реалізація та демонстрація підтримки клієнтів магазину за допомогою WebSocket-клієнта

Для роботи системи на початку було створено WebSocket-клієнт в якому відбувається спілкування користувача з системою. З нього будуть відправлятися дані на сервер, оброблятися там, генерувати відповідь, яка буде надсилатися до клієнта та виводитися на екран користувача. Спілкування та відповідь буде генеруватися із врахуванням потреб клієнта й архітектури комп'ютерної техніки та характеристик необхідних компонентів. Система буде задавати питання користувачеві і в залежності від його відповідей користувача запропонує потрібну йому комп'ютерну запчастину.

Так наприклад, коли користувач заходить на сторінку він вітається с ботом, потім бот задає питання що потрібно користувачу, користувач обирає один із варіантів котрі йому запропоновані, після цього бот задає йому ще одне питання яке уточнює потрібні критерії, і якщо боту потрібно він задає ще одне питання щоб вивести потрібну деталь.

Окремим файлом `data.js` міститься інформація про товари (рис. 3.10).

```

91 const powerSupplyModels = {
92   'modular': {
93     '500-750': ['Modular PSU 500W-750W Model A', 'Modular PSU 500W-750W Model B'],
94     '751-1000': ['Modular PSU 751W-1000W Model C', 'Modular PSU 751W-1000W Model D'],
95     '1000-1500': ['Modular PSU 1000W-1500W Model E', 'Modular PSU 1000W-1500W Model F']
96   },
97   'non-modular': {
98     '500-750': ['Non-Modular PSU 500W-750W Model A', 'Non-Modular PSU 500W-750W Model B'],
99     '751-1000': ['Non-Modular PSU 751W-1000W Model C', 'Non-Modular PSU 751W-1000W Model D'],
100    '1000-1500': ['Non-Modular PSU 1000W-1500W Model E', 'Non-Modular PSU 1000W-1500W Model F']
101  },
102  'all': {
103    '500-750': ['Modular PSU 500W-750W Model A', 'Modular PSU 500W-750W Model B', 'Non-Modular PSU 500W-750W Model A', 'Non-Modular PSU 500W-750W Model B'],
104    '751-1000': ['Modular PSU 751W-1000W Model C', 'Modular PSU 751W-1000W Model D', 'Non-Modular PSU 751W-1000W Model C', 'Non-Modular PSU 751W-1000W Model D'],
105    '1000-1500': ['Modular PSU 1000W-1500W Model E', 'Modular PSU 1000W-1500W Model F', 'Non-Modular PSU 1000W-1500W Model E', 'Non-Modular PSU 1000W-1500W Model F']
106  }
107 };
108
109 module.exports = {
110   videoCardModels,
111   processorModels,
112   ramModels,
113   powerSupplyModels
114 };
115

```

Рисунок 3.10 – Вміст файлу data.js

Наприклад, при виборі відеокарти повідомлення користувача обробляються на таких етапах: вибір виробника відеокарти, вибір цінового діапазону для доступних моделей відеокарт та її характеристик (рис. 3.11).

Він оновлює стан користувача та відповідно реагує на введені дані, щоб надати потрібну інформацію.

```

} else if (currentState.step === 'awaiting_video_manufacturer') {
  if (message.includes('amd') || message.includes('nvidia') || message.includes('пропустити')) {
    const manufacturer = message.includes('amd') ? 'amd' : (message.includes('nvidia') ? 'nvidia' : 'all');
    state[userId] = { step: 'awaiting_video_price_range', manufacturer: manufacturer };
    return `Напишіть діапазон цін для відеокарт (наприклад 0-1000, 1000-2000, 2000-3000) або напишіть "усі" цінові категорії";
  } else {
    return `Будь ласка, введіть дійсного виробника (AMD, Nvidia) або напишіть "пропустити", щоб побачити всі варіанти.`;
  }
} else if (currentState.step === 'awaiting_video_price_range') {
  const range = message.match(/(\d+\-\d+)/);
  const manufacturer = currentState.manufacturer;
  if (range && videoCardModels[manufacturer] && videoCardModels[manufacturer][range[0]]) {
    state[userId] = { step: 'default' };
    return `Доступні відеокарти для ${manufacturer.toUpperCase()} в діапазоні ${range[0]}: ${videoCardModels[manufacturer][range[0]].join(', ')}`;
  } else if (message.includes('усі цінові категорії')) {
    state[userId] = { step: 'default' };
    return `Доступні відеокарти для ${manufacturer.toUpperCase()} в усіх цінових категоріях: ${Object.values(videoCardModels[manufacturer]).flat().join(', ')}`;
  } else {
    return `Будь ласка, введіть дійсний діапазон цін для відеокарт (наприклад 0-1000, 1000-2000, 2000-3000) або напишіть "усі" цінові категорії";
  }
}

```

Рисунок 3.11 – Цикл обробки повідомлень користувача

На рисунку 3.12 наведено фрагмент коду, як відбувається обробка повідомлень користувача на прикладі вибору відеокарт. У додатку А код обробки повідомлень наведено у повному обсязі. Нижче описано ці етапи.

```

} else if (currentState.step === 'awaiting_video_manufacturer') {
  if (message.includes('amd') || message.includes('nvidia') || message.includes('пропустити')) {
    const manufacturer = message.includes('amd') ? 'amd' : (message.includes('nvidia') ? 'nvidia' : 'all');
    state[userId] = { step: 'awaiting_video_price_range', manufacturer: manufacturer };
    return `Напишіть діапазон цін для відеокарт (наприклад 0-1000, 1000-2000, 2000-3000) або напишіть "усі цінові категорії";`;
  } else {
    return `Будь ласка, введіть дійсного виробника (AMD, Nvidia) або напишіть "пропустити", щоб побачити всі варіанти.`;
  }
} else if (currentState.step === 'awaiting_video_price_range') {
  const range = message.match(/\d+-\d+/);
  const manufacturer = currentState.manufacturer;
  if (range && videoCardModels[manufacturer] && videoCardModels[manufacturer][range[0]]) {
    state[userId] = { step: 'default' };
    return `Доступні відеокарти для ${manufacturer.toUpperCase()} в діапазоні ${range[0]}: ${videoCardModels[manufacturer][range[0]].join(', ')}`;
  } else if (message.includes('усі цінові категорії')) {
    state[userId] = { step: 'default' };
    return `Доступні відеокарти для ${manufacturer.toUpperCase()} в усіх цінових категоріях: ${Object.values(videoCardModels[manufacturer]).flat().join(', ')}`;
  } else {
    return `Будь ласка, введіть дійсний діапазон цін для відеокарт (наприклад 0-1000, 1000-2000, 2000-3000) або напишіть "усі цінові категорії";`;
  }
}

```

Рисунок 3.12 – Цикл обробки повідомлень користувача

1. Перевірка на етап «awaiting_video_manufacturer» (очікування вибору виробника відеокарти):

- якщо поточний стан користувача currentState.step є awaiting_video_manufacturer, то перевіряється, чи містить повідомлення користувача (message) слова «amd», «nvidia» або «пропустити»;
- якщо повідомлення містить «amd», встановлюється виробник як amd, якщо містить «nvidia», встановлюється виробник як «nvidia». Якщо повідомлення містить «пропустити», виробник встановлюється як «all»;
- після встановлення виробника, стан користувача state[userId] змінюється на awaiting_video_price_range, а також зберігається вибраний виробник;
- користувачеві надсилається відповідь із проханням вказати діапазон цін для відеокарт або написати «усі цінові категорії»;
- якщо ж повідомлення не містить жодного з цих ключових слів, користувачеві надсилається повідомлення з проханням ввести дійсного виробника;

2. Перевірка на етап «awaiting_video_price_range» (очікування вибору цінового діапазону):

- якщо поточний стан користувача currentState.step є awaiting_video_price_range, відбувається спроба знайти в повідомленні

користувача діапазон цін за допомогою регулярного виразу
(message.match(/^\d+-\d+/));

– якщо діапазон цін знайдено і є відповідні моделі відеокарт для вибраного раніше виробника в цьому діапазоні videoCardModels[manufacturer][range[0]], то стан користувача змінюється на default і користувачеві надсилається список доступних відеокарт у вказаному діапазоні;

– якщо ж повідомлення містить «усі цінові категорії», користувачеві надсилається список всіх доступних відеокарт для вибраного виробника у всіх цінових категоріях, а стан змінюється на default;

– якщо введений діапазон цін некоректний або не знайдено відповідних моделей, користувачеві надсилається прохання ввести дійсний діапазон цін або написати «усі цінові категорії».

Таким чином, цей код обробляє повідомлення користувача на двох етапах: вибір виробника відеокарти та вибір цінового діапазону для доступних моделей відеокарт. Він оновлює стан користувача та відповідно реагує на введені дані, щоб надати потрібну інформацію.

Коли користувач заходить на сторінку спілкування, він бачить у центрі екрану чат, з лівої частини якого є кнопка повернення до магазину, знизу екрана дві кнопки: кнопка відправлення повідомлення і кнопка очистки чату (рис. 3.13).

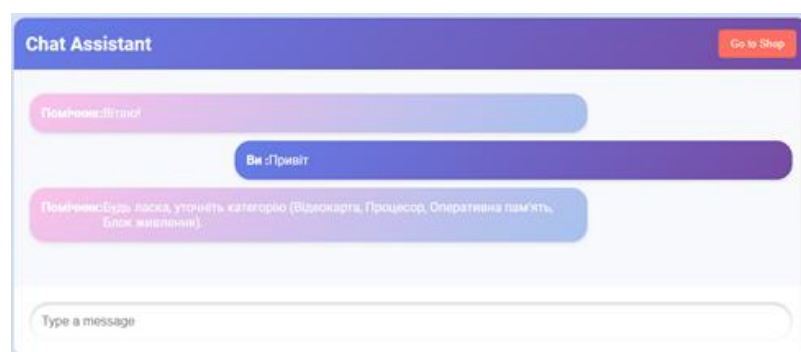


Рисунок 3.13 – Загальний вигляд сторінки чату

Бот-помічник задає питання, користувач пишемо що його цікавить, і отримуємо необхідну інформацію (рис. 3.14). У випадку, якщо користувач не правильно вводить дані, бот буде просити його правильно ввести дані (рис. 3.15).

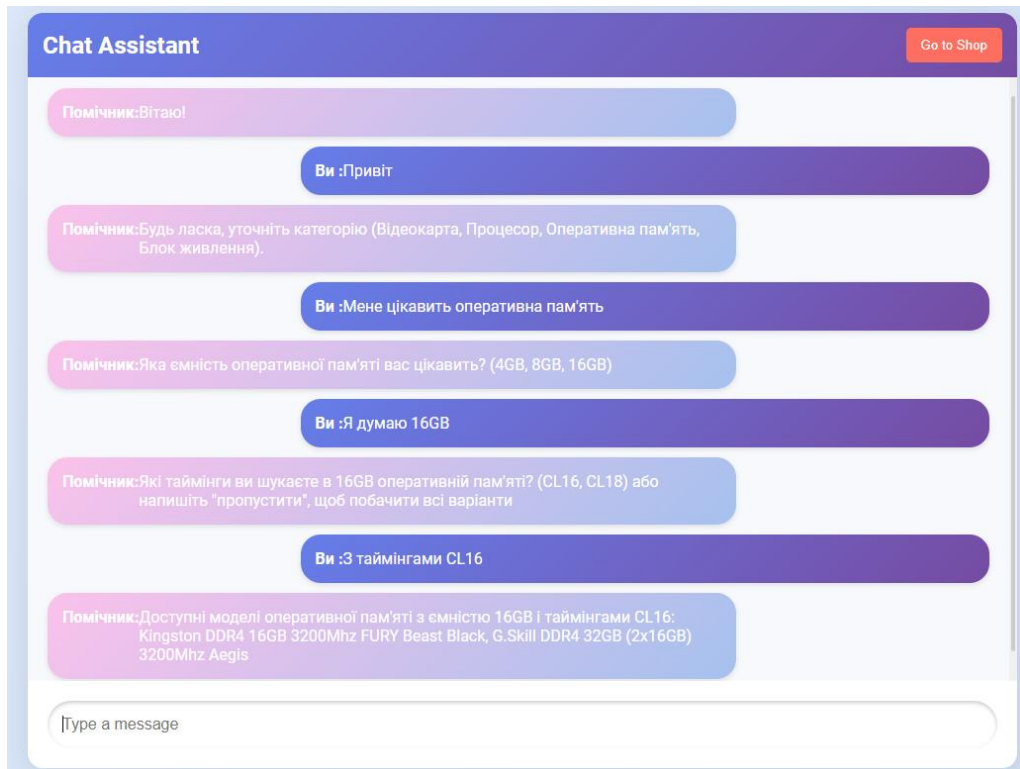


Рисунок 3.14 – Приклад пошуку оперативної пам'яті на 16 гігабайт

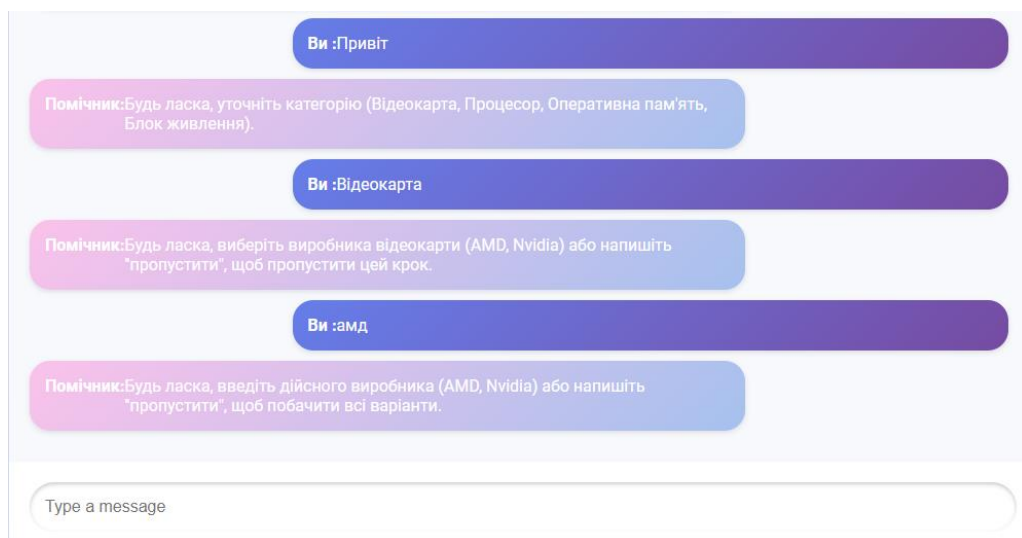


Рисунок 3.15 – Приклад діалогу з ботом

3.4 Інтеграція мовних моделей за допомогою методу RAG

RAG (англ. Retrieval Augmented Generation) – це метод роботи з великими мовними моделями, коли користувач задає запитання, а розробник програмно до цього питання «підмішує» додаткову інформацію з будь-яких зовнішніх джерел і подає текст на вхідній мовній моделі [24]. Іншими словами, додається в контекст запиту до мовної моделі додаткова інформація, на основі якої мовна модель може дати вам більш повну і точну відповідь (рис. 3.16).

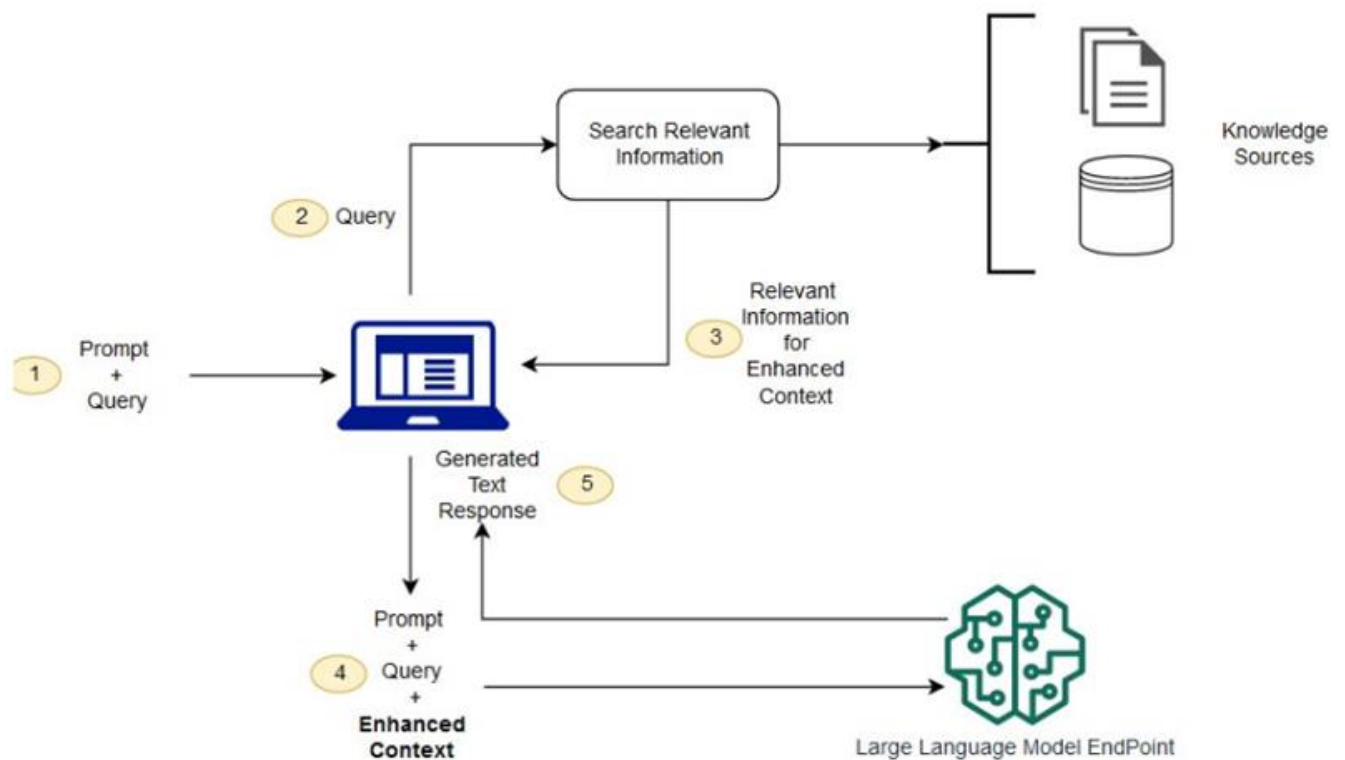


Рисунок 3.16 – Концепт використання RAG з LLM

У нашому випадку застосунок реалізує взаємодію користувача з Large Language Model (LLM) через веб-інтерфейс із використанням методу Retrieval-Augmented Generation (RAG) [25]. Коли користувач вводить запит на вебсторінку, запит відправляється на сервер через WebSocket. На сервері запит обробляється, використовуючи попередньо завантажені дані, і передаються моделі LLM для генерації відповіді.

Метод RAG дозволяє програмі шукати релевантну інформацію із зовнішніх джерел і використовувати її для покращення відповіді, створеної моделлю LLM. Відповідь моделі повертається на веб-сторінку і відображається користувачеві в режимі реального часу. Таким чином, програма ефективно поєднує можливості пошуку інформації та генеративні можливості LLM, забезпечуючи інтерактивну та точну взаємодію з користувачем.

Вебзастосунок створено на базі *FastAPI*, який використовує Jinja2 для шаблонізації HTML, обслуговує статичні файли, дозволяє крос-доменні запити (CORS) для фронтенду на React, і обробляє WebSocket-з'єднання (рис. 3.17) [26].

```
from fastapi import FastAPI, Request, WebSocket
from fastapi.templating import Jinja2Templates
from fastapi.staticfiles import StaticFiles
from fastapi.middleware.cors import CORSMiddleware
from ai_code import request_ai_with_tool

import pandas as pd

app = FastAPI()

app.add_middleware(
    CORSMiddleware,
    allow_origins=["http://localhost:3000"],
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)

templates = Jinja2Templates(directory="templates")

app.mount("/static", StaticFiles(directory="static"), name="static")

@app.get("/")
async def get(request: Request):
    return templates.TemplateResponse("index.html", {"request": request})

@app.websocket("/ws")
async def websocket_endpoint(websocket: WebSocket):
    await websocket.accept()
    while True:
        data = await websocket.receive_text()
        print(data)
        response = request_ai_with_tool(data)
        await websocket.send_text(str(response))
```

Рисунок 3.17 – Створення веб-додатку на базі FastAPI

Файл *ai_code.py* (додаток Б). Далі було створено агента на основі методики ReAct (англ. Reasoning and Acting) для взаємодії з різними інструментами, зокрема для виконання запитів до таблиць Pandas та створення нотаток. Він використовує FastAPI, Jinja2, WebSocket, та інтеграцію з OpenAI для реалізації таких функцій:

- завантаження змінних середовища: Використовується `'dotenv'` для завантаження конфігураційних змінних з файлу `'.env'`;
- завантаження даних: CSV-файл `'components.csv'` завантажується у DataFrame за допомогою Pandas;
- створення Query Engine: Використовується `'PandasQueryEngine'` для виконання запитів до завантаженого DataFrame. Налаштовуються підказки для запитів;
- налаштування інструментів: Інструменти для агента включають `'note_engine'` та `'QueryEngineTool'`, який працює з `'PandasQueryEngine'` для запитів до даних компонентів;
- створення агента ContextAwareReActAgent: Цей агент розширює базовий `'ReActAgent'`, додаючи можливість зберігати історію запитів та відповідей. Він будує контекст з попередніх запитів для покращення обробки нових запитів;
- налаштування мовної моделі: використовується мовна модель GPT-3.5 від OpenAI;
- функція для обробки запитів: Функція `'request_ai_with_tool'` приймає промпт, передає його агенту для обробки та повертає результат.

Цей код дозволяє агенту обробляти запити користувачів, взаємодіяти з даними в DataFrame, отримувати інформацію про компоненти зі складу та створювати нотатки, використовуючи контекст попередніх запитів.

Файл *prompts.py*. Цей код налаштовує шаблони підказок (prompts) для агента, який допомагає користувачам працювати з Pandas DataFrame, конвертуючи запити у виконуваний Python-код. Основні функції та компоненти коду такі:

- інструкції для генерації коду – `instruction_str`: містить інструкції для перетворення запитів користувачів у виконуваний Python-код за допомогою `Pandas`, які наголошують на створенні виразу, який може бути виконаний за допомогою функції `eval()`;
- шаблон підказки для нових запитів – `new_prompt`: створюється об'єкт `PromptTemplate`, який включає текст підказки для агента, що надає контекст про `DataFrame`, його структуру (вивід `df.head()`), інструкції для створення правильного виразу та сам запит користувача;
- контекст для агента – `context`: описує основну мету агента, яка полягає у наданні користувачам точної інформації про компоненти на складі;
- у підсумку, цей код налаштовує інструкції та шаблони, які використовуватимуться агентом для обробки запитів користувачів, допомагаючи їм отримувати необхідну інформацію з `DataFrame` за допомогою правильних виразів на Python, що виконуються через `eval()`.

Вміст DataSet components.csv. `DataSet components.csv`, взятий з Kaggle, містить інформацію про різноманітні компоненти комп'ютерної техніки. Він має 2301 запис і 12 стовпців. Ось структура даних, яка міститься у файлі:

- `article_number`: унікальний номер статті для кожного компонента, це цілочисельне поле;
- `name`: назва компонента, представлена у вигляді рядка тексту;
- `brand_name`: назва бренду компонента, також представлена у вигляді рядка тексту;
- `category`: категорія, до якої належить компонент (наприклад, «Hard Drives», «RAM Memory», «Towers» тощо), це рядкове поле, але для деяких записів може бути порожнім;
- `product_number`: номер продукту або модельний номер компонента, це рядкове поле.

- price: ціна компонента, представлена у вигляді числа з плаваючою точкою;
- score: оцінка компонента, також представлена у вигляді числа з плаваючою точкою;
- reviews: кількість відгуків про компонент, це цілочисельне поле;
- amount: кількість компонентів, доступних на складі, це цілочисельне поле;
- memory: об'єм пам'яті компонента, представлений у вигляді рядка тексту, це поле заповнене не для всіх записів;
- power: потужність компонента, представлена у вигляді рядка тексту, це поле заповнене не для всіх записів;
- frequency: частота компонента, представлена у вигляді рядка тексту, це поле заповнене не для всіх записів.

Дані в цьому файлі структуровані так, що можна легко фільтрувати та сортувати компоненти за різними характеристиками, такими як ціна, кількість відгуків, бренд, категорія тощо.

Це робить файл корисним для різних аналітичних завдань та управління запасами.

При роботі віртуального-помічника, інтегрованого з мовною моделлю, інформація, яка надається по запитам клієнта, узгоджується з наявністю комплектуючих на складі інтернет-магазину.

3.5 Тестування та демонстрація роботи чат-бота

За основу було взято дизайн бота створеного за допомогою WebSocket-клієнта.

Віртуальний бот-помічник може працювати з історією запитів користувача (рис. 3.18).

Також бот може адаптуватися до мови користувача, і відповідати йому на запити мовою, котрою йому надсилають запити (рис. 3.19).

З ботом також можна вести діалог на порівняння комп'ютерних комплектуючих, на інші теми він буде відповідати що не може дати відповіді на запити користувача. Тобто бот строго налаштований на відповіді лише по темі комп'ютерних комплектуючих (рис. 3.20).

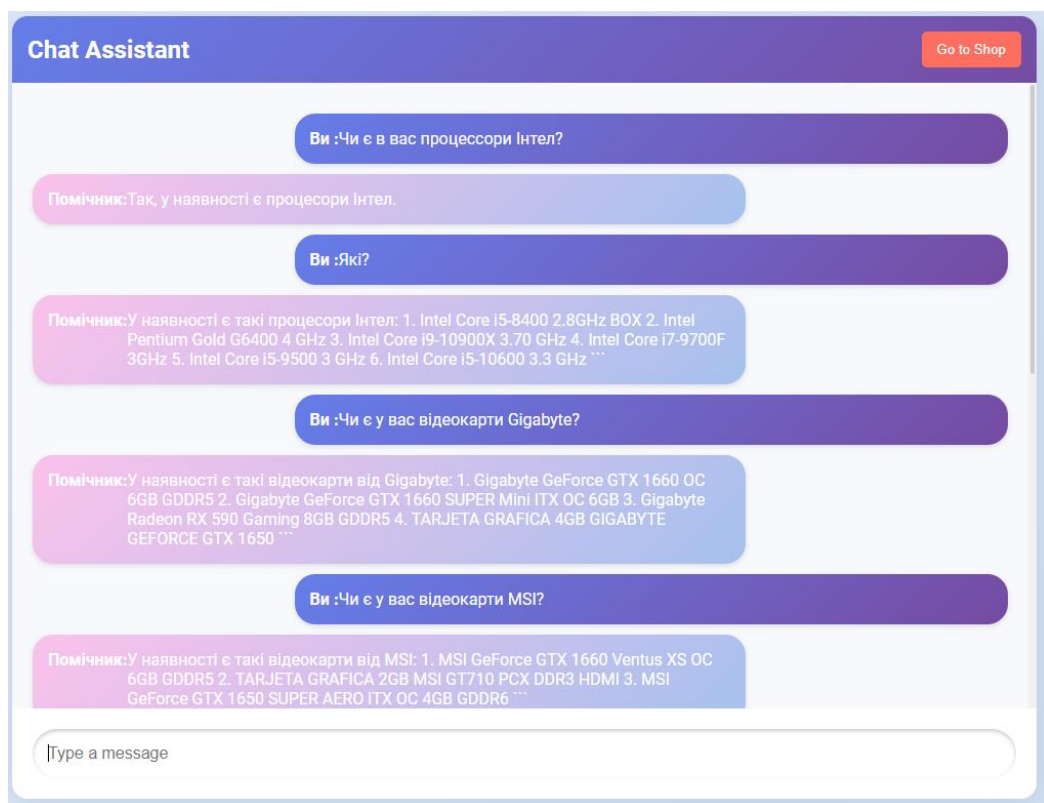


Рисунок 3.18 – Приклад роботи боту та зв'язку з історією запитів

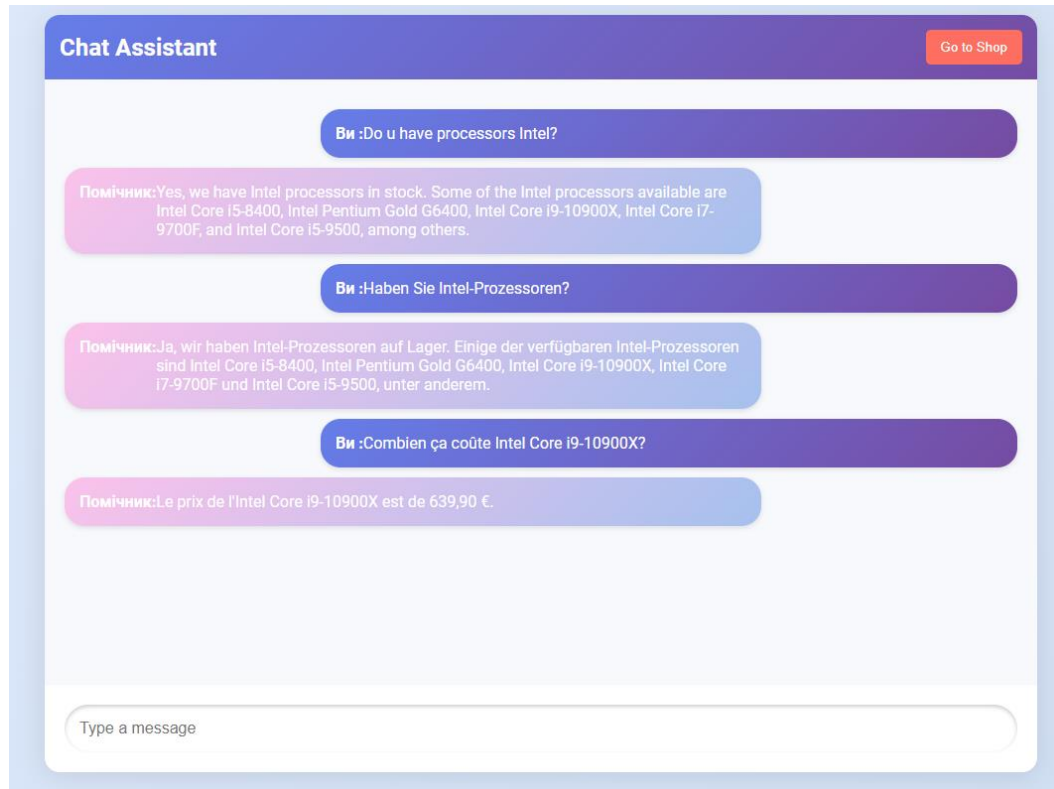


Рисунок 3.19 – Приклад адаптування бота до мови користувача

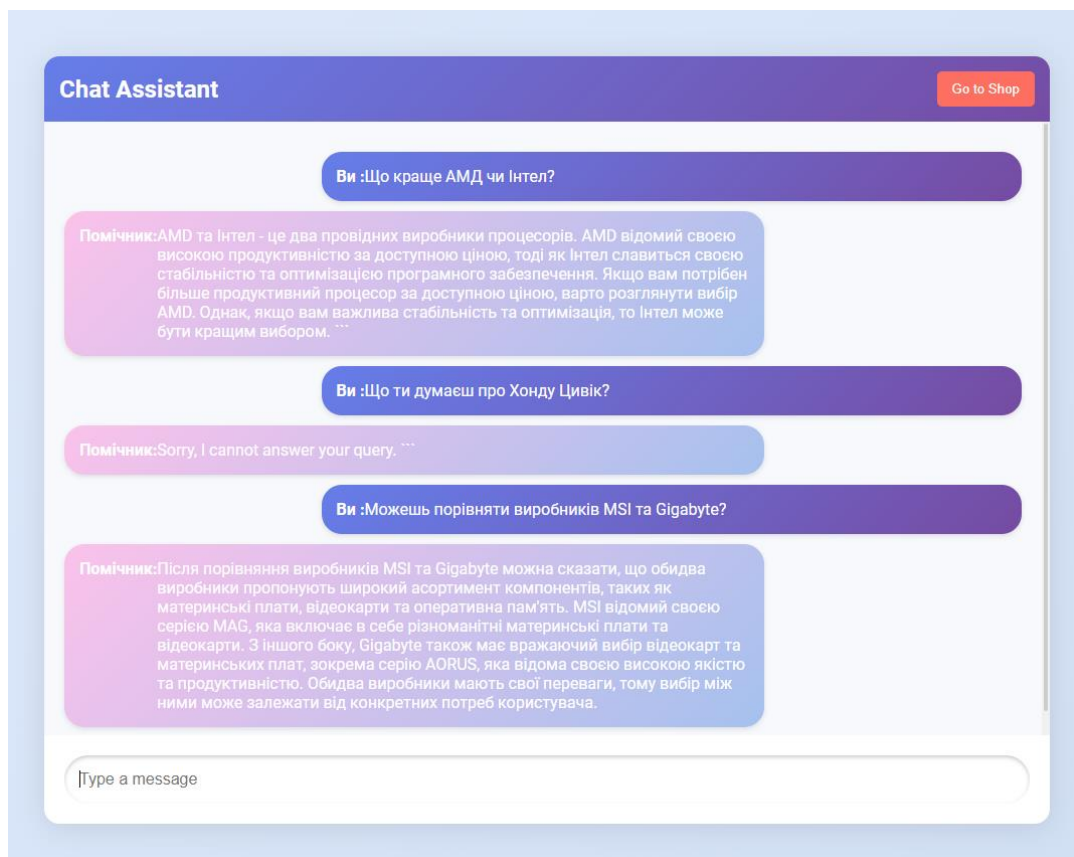


Рисунок 3.20 – Зразок діалогу з ботом

Коли користувач задає питання «Що краще?» або «Можеш порівняти один бренд та другий?», бот починає в будь-якому разі порівнювати ці компоненти.

На прикладі запиту на рисунку 3.21 бот перебирає усі компоненти виробників MSI та Gigabyte та потім формує відповідь на основі отриманої інформації.

У віртуального помічника також можна дізнатись про різні характеристики деяких комплектуючих, наприклад про таймінги оперативної пам'яті, що це таке, та які краще таймінги обирати. (рис. 3.22).

```

Observation:  article_number      name brand_name      category  product_number  price  score  reviews  amount  memory  power  frequency
40           1117      MSI Mag Z390 Tomahawk  MSI  Motherboards  911-7B18-001  154,90000  9,62  163  7  NaN  NaN  NaN
50           3261      MSI B550M PRO-VDH WIFI  MSI  Motherboards  911-7C95-09S  129,91000  8,00  7  3  NaN  NaN  NaN
61           7623  TARJETA GRAFICA 2GB MSI GT710 PCX DDR3 HDMI  MSI  Graphics Cards  GT  48,74000  0,00  0  7  NaN  NaN  NaN
85           1291      MSI H310M PRO-VDH PLUS  MSI  Motherboards  911-7C09-001  58,99000  8,92  24  7  NaN  NaN  NaN
121          2758      MSI B450-A Pro Max      MSI  Motherboards  911-7B86-022  91,98000  9,27  186  12  NaN  NaN  NaN
...         ...         ...         ...         ...         ...         ...         ...         ...         ...         ...         ...
2194         39      MSI GeForce GTX 1660 Ventus XS OC 6GB GDDR5  MSI  Graphics Cards  912-V379-014  230,90000  9,53  90  13  NaN  NaN  NaN
2199         4059      MSI MPG Z490 GAMING EDGE WIFI  MSI  Motherboards  911-7C79-005  219,91000  9,50  40  4  NaN  NaN  NaN
2232         6194      MSI MAG B460 Torpedo    MSI  Motherboards  911-7C81-020  104,89000  0,00  0  13  NaN  NaN  NaN
2250         5053  MSI GeForce GTX 1650 SUPER AERO ITX OC 4GB GDDR6  MSI  Graphics Cards  912-V809-3295  161,46281  9,43  7  0  NaN  NaN  NaN
2288         4256  MSI MPG Gungnir 110M Cristal Templado USB 3.2 ...  MSI  Towers  MPG GUNGNIR 110M  89,99000  9,33  36  18  NaN  NaN  NaN

[97 rows x 12 columns]
Thought: I need to use a tool to help me answer the question.
Action: component_data
Action Input: {'input': 'Gigabyte'}
> Pandas Instructions:
...
df[df['brand_name'].str.contains('Gigabyte', case=False)]
...
> Pandas Output:
  article_number      name brand_name      category  product_number  price  score  reviews  amount  memory  power  frequency
33           2273  Gigabyte AORUS RGB DDR4 3200 PC4-25600 2x8GB C...  Gigabyte  RAM Memory  GP-ARS16G32  129,00000  0,00  0  16  8GB  NaN  NaN
35           6524      Gigabyte B450 Gaming X      Gigabyte  Motherboards  B450 Gaming X  84,990000  9,14  44  9  NaN  NaN  NaN
46           2261      Gigabyte B550 GAMING X V2    Gigabyte  Motherboards  GAB550M02-00-G  134,90000  0,00  0  19  NaN  NaN  NaN
54           41      Gigabyte GeForce GTX 1660 OC 6GB GDDR5  Gigabyte  Graphics Cards  GV-N1660OC-6GD  208,876033  9,30  327  20  6GB  NaN  NaN
78           2586      Gigabyte X570 Aorus Pro      Gigabyte  Motherboards  X570 AORUS PRO  259,900000  0,00  34  11  NaN  NaN  NaN
...         ...         ...         ...         ...         ...         ...         ...         ...         ...         ...         ...
2172         1412  Gigabyte GeForce GTX 1660 SUPER Mini ITX OC 6G...  Gigabyte  Graphics Cards  GV-N1665IXOC-6GD  229,900000  9,67  6  13  NaN  NaN  NaN
2175         3870      Gigabyte Radeon RX 590 Gaming 8GB GDDR5  Gigabyte  Graphics Cards  GV-RX590GAMING-8GD  219,890000  8,75  48  1  NaN  NaN  NaN
2186         7639  TARJETA GRAFICA 4GB GIGABYTE GEFORCE GTX 1650 ...  GIGABYTE  Graphics Cards  GV-N1655OC-4GD  190,650000  0,00  0  1  NaN  NaN  NaN
2203         3066      Gigabyte Ga-B250-FinTech      Gigabyte  Motherboards  GA-B250-FinTech  99,990000  8,59  27  12  NaN  NaN  NaN
2252         2271  Gigabyte AORUS RGB DDR4 3600 PC4-28800 2x8GB C...  Gigabyte  RAM Memory  GP-AR36C18S8K2H416RD  123,636364  0,00  0  5  8GB  NaN  NaN

```

Рисунок 3.21 – Приклад порівняння ботом компонентів обох виробників для формування відповіді

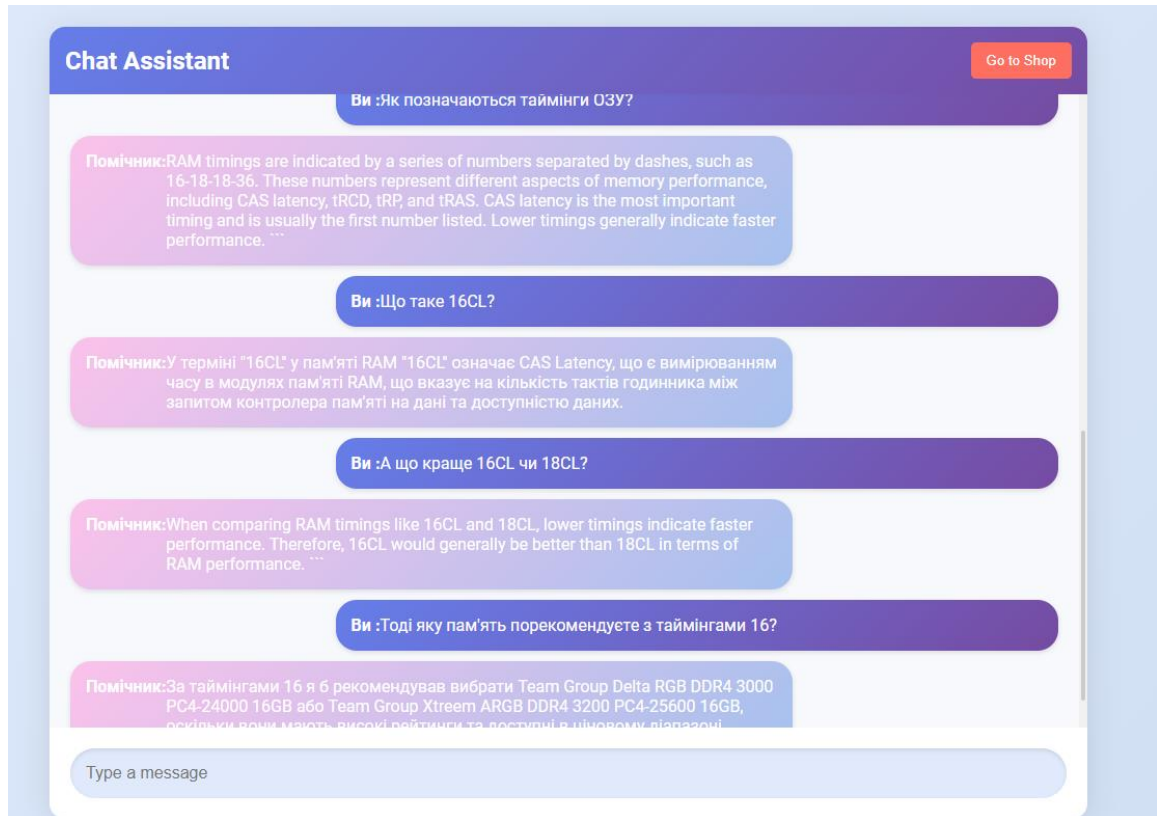


Рисунок 3.22 – Приклад отримання інформації про характеристики оперативної пам'яті

Висновки до розділу 3

У даному розділі було розглянуто процес проектування та програмної реалізації інтелектуальної системи підтримки клієнтів онлайн-магазину комп'ютерної техніки. Розроблено структуру інтерфейсу користувача та функціональність системи, а також здійснено її програмну реалізацію. У системі було реалізовано механізми обробки запитів користувачів, надання рекомендацій та забезпечення швидкої відповіді на запити клієнтів.

Використовуючи методи генеративного штучного інтелекту, зокрема RAG (Retrieval-Augmented Generation), було створено систему, яка ефективно інтегрує механізми пошуку інформації з можливостями генеративних мовних моделей LLM. Це дозволяє надавати користувачам більш точні та релевантні відповіді на їхні запити. У розділі описано основні етапи розробки, включаючи проектування інтерфейсу користувача, програмування основних компонентів системи та

інтеграцію з серверами через WebSocket. Продемонстровано, як користувачі можуть взаємодіяти з системою через зручний вебінтерфейс, вводячи запити та отримуючи відповіді у режимі реального часу. Також було проведено тестування та налагодження системи, що підтвердило її здатність покращити обслуговування клієнтів і підвищити ефективність роботи онлайн-магазину.

Реалізована система дозволяє підвищити ефективність обслуговування клієнтів, зменшити час на обробку запитів та покращити загальний рівень задоволеності клієнтів. Це створює передумови для подальшого розвитку системи та її адаптації до нових вимог і умов ринку.

ВИСНОВКИ

Впровадження інтелектуальної системи підтримки клієнтів у онлайн-магазин комп'ютерної техніки є важливим кроком до підвищення ефективності обслуговування та задоволення клієнтів. Використання сучасних алгоритмів та підходів дозволяє автоматизувати багато аспектів взаємодії з клієнтами, забезпечуючи персоналізований та швидкий сервіс. Водночас, впровадження таких систем вимагає ретельного планування та постійного вдосконалення для забезпечення їхньої ефективності та безпеки.

Інтелектуальна система підтримки клієнтів може стати ключовим інструментом для підвищення конкурентоспроможності онлайн-магазину комп'ютерної техніки. Використання сучасних технологій, таких як натуральна обробка мови, машинне навчання та системи рекомендацій, дозволяє створити ефективну, швидку та персоналізовану взаємодію з клієнтами. Однак, для досягнення успіху необхідно ретельно планувати впровадження системи, забезпечувати її безпеку та постійно вдосконалювати. Таким чином, інтелектуальна система підтримки клієнтів не лише підвищить якість обслуговування, але й сприятиме довгостроковому зростанню та успіху бізнесу.

Проведене дослідження дозволяє зробити наступні висновки.

Аналіз сфери онлайн продажу комп'ютерної техніки та дослідження потенціалу використання штучного інтелекту є основою для покращення взаємодії з клієнтами інтернет-магазинів. Було розглянуто основні проблеми, що виникають при підтримці клієнтів у сфері електронної комерції, специфіку онлайн продажу комп'ютерної техніки, а також проаналізовані існуючі методи та рішення до використання ботів для підтримки клієнтів на основі ШІ. На основі отриманих даних запропоновано концепцію розробки інтелектуальної системи, спрямованої на підвищення ефективності та якості обслуговування клієнтів у сфері онлайн продажу комп'ютерної техніки. Створення чат-бота віртуального помічника, який спрямований на покращення якості обслуговування клієнтів, підвищення

ефективності бізнес-процесів, збільшення продажі за рахунок задоволення потреб споживачів.

У якості середовища розробки обрано IDE Visual Studio Code, яке забезпечує зручний і ефективний процес розробки завдяки підтримці різноманітних мов програмування та розширень і мову програмування Python, яка надає широкі можливості для розробки систем штучного інтелекту. Для розробки інтерфейсу користувача обрано HTML та CSS. Для забезпечення динамічності та інтерактивності хостосунку магазину обрано JavaScript та бібліотеку React. Протокол зв'язку WebSocket було задіяно для забезпечення взаємодії між клієнтом і сервером у режимі реального часу. Ці технології були обрані з метою створення ефективної, інтерактивної та зручної для користувачів системи.

Python та його бібліотеки, такі як NumPy, Pandas, та LlamaIndex, були використані для обробки даних та реалізації алгоритмів штучного інтелекту. NumPy надає потужні інструменти для роботи з багатовимірними масивами та виконанням високопродуктивних математичних операцій. Pandas забезпечує зручний спосіб обробки та аналізу табличних даних, включаючи фільтрацію, агрегацію та візуалізацію. LlamaIndex використовується для створення індексів та пошуку по великим обсягам текстових даних, що дозволяє ефективно реалізувати функції інтелектуального пошуку та рекомендацій.

Використовуючи методи генеративного штучного інтелекту, зокрема RAG (Retrieval-Augmented Generation), було створено систему, яка ефективно інтегрує механізми пошуку інформації з можливостями генеративних мовних моделей LLM. Вибір технологій та інструментальних засобів забезпечив основу для розробки та реалізації інтелектуальної системи підтримки клієнтів онлайн-магазину комп'ютерної техніки.

Здійснено проектування та програмну реалізацію системи підтримки клієнтів онлайн-магазину комп'ютерної техніки. У системі було реалізовано механізми обробки запитів користувачів, надання рекомендацій та забезпечення швидкої відповіді на запити клієнтів. Обробка запитів користувача здійснювалася з

використанням методу генеративного ШІ та великих мовних моделей. Це дозволяє підібрати компоненти для комп'ютерної техніки у відповідності з потребами клієнта та наявними у магазині товарами. Реалізована система дозволяє підвищити ефективність обслуговування клієнтів, зменшити час на обробку запитів та покращити загальний рівень задоволеності клієнтів. Це створює передумови для подальшого розвитку системи та її адаптації до нових вимог і умов ринку.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Загальні принципи архітектури комп'ютерів : вебсайт. URL: https://elearning.sumdu.edu.ua/free_content/lectured:1a259358378153792bb8645df287e86d790fc40d/20160903092057/44806/index.html (дата звернення 23.04.2024).
2. Будова персонального комп'ютера : вебсайт. URL: <https://informatics.dp.ua/budova-personalnoho-kompyutera/> (дата звернення 23.05.2024).
3. Різновиди комп'ютерів : вебсайт. URL: <https://ua5.org/device/2191-riznovydy-kompyuteriv.html> (дата звернення 24.03.2024).
4. Як вибрати процесор, щоб комп'ютер точно все потягнув : вебсайт. URL: <https://zhuk.ua/poradi-ta-rekomendatsii/yak-vybraty-protsesor-shchob-kompiuter-tochno-vse-potiahnuv/> (дата звернення 25.04.2024).
5. Що таке штучний інтелект: історія, види та складові : вебсайт. URL: <https://gigacloud.ua/blog/navchannja/scho-take-shtuchnij-intelekt-istorija-vidi-ta-skladovi> (дата звернення 25.05.2024).
6. Що таке бот? – Опис типів ботів : вебсайт. URL: <https://aws.amazon.com/ru/what-is/bot/> (дата звернення 26.04.2024).
7. Інтелектуальні системи підтримки клієнтів / за ред. І. В. Сидоренка. Київ: КНТЕУ, 2020. 400 с.
8. Що таке ChatGPT та як його використовувати? : вебсайт. URL: <https://blog.ithillel.ua/articles/what-is-chatgpt-and-how-to-use-it> (дата звернення 27.05.2024).
9. Visual Studio: IDE і редактор кода для розробників та груп : вебсайт. URL: <https://visualstudio.microsoft.com/ru/> (дата звернення 01.06.2024).
10. Фрейн Б. Responsive Web Design with HTML5 and CSS: Develop future-proof responsive websites using the latest HTML5 and CSS techniques, 3rd Edition, 336 с.

11. Дакетт Дж. HTML та CSS. Розробка та дизайн веб-сайтів : довідник. Ексмо, Україна, 2017, 480 с.
12. Фленаган Д. JavaScript повне керівництво; довідник по самому популярному мові програмування : довідник. Вид. 7-е, Україна, 2021, 722 с.
13. Сидоренко А. П., Іваненко О. К., Коваленко М. В. JavaScript: сучасний підхід до веб-розробки, Харків: Ранок, 2019. 320 с.
14. React – A JavaScript library for building user interfaces : вебсайт. URL: <https://legacy.reactjs.org/> (дата звернення 02.05.2024).
15. Тіленс Томас, React у дії : довідник. Print2Print, Україна, 2019, 368 с.
16. WebSockets: навіщо потрібні та як з ними працювати : вебсайт. URL: <https://proit.org.ua/websockets-navishcho-potribni-ta-iaak-z-nimi-pratsiuvati-2/> (дата звернення 05.05.2024).
17. Гнатенко В. С., Ткаченко А. В., Іванчук О. М. WebSockets: інтерактивні веб-додатки. Дніпро : Дніпрокнига, 2018. 280 с.
18. Luciano R. Fluent Python, 2nd Edition, O'Reilly Media, Inc., 2022. 1012 с.
19. Що таке мова програмування Python? : вебсайт. URL: <https://freehost.com.ua/ukr/faq/wiki/chto-takoe-jazik-programmirovanija-python/> (дата звернення 07.05.2024).
20. Пасхавер Б. Pandas в дії : довідник. Print2Print, Україна, 2022, 512 с.
21. Чому вам слід використовувати NumPy : вебсайт. URL: <https://devzone.org.ua/post/chomu-vam-slid-vykorystovuvaty-numpy> (дата звернення 03.03.2024).
22. Ковальчук О. В., Лисенко М. П., Савчук Ю. І. LlamaIndex : робота з базами даних : монографія. Київ: Видавництво Політехніка, 2022. 270 с.
23. Сучасні бібліотеки машинного навчання : вебсайт. URL: <https://travelscode.com/suchasni-biblioteki-mashinnogo-navchannya/> (дата звернення 09.04.2024).

24. Що таке RAG система та як її побудувати : вебсайт. URL: <https://fwdays.com/en/event/python-ds-fwdays-2024/review/what-is-a-rag-system-and-how-to-build-it> (дата звернення 1.04.2024).

25. Що таке велика мовна модель (Large Language Model, LLM)? : вебсайт. URL: <https://thetransmitted.com/adlucem/shho-take-velyka-movna-model-large-language-model-llm/> (дата звернення 11.05.2024).

26. Чому я обираю FastAPI: основні можливості та переваги фреймворку : вебсайт. URL: <https://dou.ua/forums/topic/37547/> (дата звернення 11.05.2024).

ДОДАТОК А

Код файлу logic.js

```
const { videoCardModels, processorModels, ramModels, powerSupplyModels } = require('./data');
let state = {};
function handleMessage(userId, message) {
  if (typeof message !== 'string') {
    return 'Неправильний формат повідомлення.';
  }
  message = message.toLowerCase();
  if (!state[userId]) {
    state[userId] = { step: 'default' };
  }
  const currentState = state[userId];
  if (currentState.step === 'default') {
    if (message.includes('відеокарта')) {
      state[userId] = { step: 'awaiting_video_manufacturer' };
      return 'Будь ласка, виберіть виробника відеокарти (AMD, Nvidia) або напишіть "пропустити", щоб пропустити цей крок.';
    } else if (message.includes('процесор')) {
      state[userId] = { step: 'awaiting_processor_manufacturer' };
      return 'Виберіть виробника процесора? (AMD, Intel, Uci)';
    } else if (message.includes('оперативна пам\`ять')) {
      state[userId] = { step: 'awaiting_ram_memory_capacity' };
      return 'Яка ємність оперативної пам\`яті вас цікавить? (4GB, 8GB, 16GB)';
    } else if (message.includes('блок живлення')) {
      state[userId] = { step: 'awaiting_psu_type' };
      return 'Будь ласка, виберіть тип блоку живлення (Модульний, Немодульний) або напишіть "пропустити", щоб пропустити цей крок.';
    } else {
      return 'Будь ласка, уточніть категорію (Відеокарта, Процесор, Оперативна пам\`ять, Блок живлення).';
    }
  }
}
```

```

} else if (currentState.step === 'awaiting_video_manufacturer') {
  if (message.includes('amd') || message.includes('nvidia') || message.includes('пропустити')) {
    const manufacturer = message.includes('amd') ? 'amd' : (message.includes('nvidia') ? 'nvidia' : 'all');
    state[userId] = { step: 'awaiting_video_price_range', manufacturer: manufacturer };
    return 'Напишіть діапазон цін для відеокарт (наприклад 0-1000, 1000-2000, 2000-3000) або напишіть "усі цінові категорії"';
  } else {
    return 'Будь ласка, введіть дійсного виробника (AMD, Nvidia) або напишіть "пропустити", щоб побачити всі варіанти.';
  }
} else if (currentState.step === 'awaiting_video_price_range') {
  const range = message.match(/^\d+-\d+$/);
  const manufacturer = currentState.manufacturer;
  if (range && videoCardModels[manufacturer] && videoCardModels[manufacturer][range[0]]) {
    state[userId] = { step: 'default' };
    return `Доступні відеокарти для ${manufacturer.toUpperCase()} в діапазоні ${range[0]}: ${videoCardModels[manufacturer][range[0]].join(', ')}`;
  } else if (message.includes('усі цінові категорії')) {
    state[userId] = { step: 'default' };
    return `Доступні відеокарти для ${manufacturer.toUpperCase()} в усіх цінових категоріях: ${Object.values(videoCardModels[manufacturer]).flat().join(', ')}`;
  } else {
    return 'Будь ласка, введіть дійсний діапазон цін для відеокарт (наприклад 0-1000, 1000-2000, 2000-3000) або напишіть "усі цінові категорії"';
  }
} else if (currentState.step === 'awaiting_processor_manufacturer') {
  if (message.includes('amd') || message.includes('intel') || message.includes('усі')) {
    const manufacturer = message.includes('amd') ? 'amd' : (message.includes('intel') ? 'intel' : 'all');
    state[userId] = { step: 'awaiting_processor_performance_level', manufacturer: manufacturer };
    return 'Будь ласка, вкажіть рівень продуктивності (низький, середній, високий)';
  } else {
    return 'Будь ласка, введіть дійсного виробника (AMD, Intel, Усі)';
  }
} else if (currentState.step === 'awaiting_processor_performance_level') {

```

```

if (message.includes('низький') || message.includes('середній') || message.includes('високий')) {
    const performanceLevel = message.includes('низький') ? 'low' : (message.includes('середній') ?
'medium' : 'high');

    state[userId] = { step: 'awaiting_processor_price_range', performanceLevel: performanceLevel };

    return 'Напишіть діапазон цін для процесорів (наприклад 0-1000, 1000-2000, 2000-3000) або
напишіть "усі цінові категорії"';

} else {

    return 'Будь ласка, введіть дійсний рівень продуктивності (низький, середній, високий);

}

} else if (currentState.step === 'awaiting_processor_price_range') {

    const range = message.match(/\d+-\d+/);

    const performanceLevel = currentState.performanceLevel;

    const manufacturer = currentState.manufacturer;

    if (range && processorModels[manufacturer] &&
processorModels[manufacturer][performanceLevel] &&
processorModels[manufacturer][performanceLevel][range[0]]) {

        state[userId] = { step: 'default' };

        return `Доступні процесори для ${manufacturer.toUpperCase()} ${performanceLevel}
продуктивності в діапазоні ${range[0]}:
${processorModels[manufacturer][performanceLevel][range[0]].join(', ')}`;

    } else if (message.includes('усі цінові категорії')) {

        state[userId] = { step: 'default' };

        return `Доступні процесори для ${manufacturer.toUpperCase()} ${performanceLevel}
продуктивності в усіх цінових категоріях:
${Object.values(processorModels[manufacturer][performanceLevel]).flat().join(', ')}`;

    } else {

        return 'Будь ласка, введіть дійсний діапазон цін для процесорів (наприклад 0-1000, 1000-
2000, 2000-3000) або напишіть "усі цінові категорії"';

    }

} else if (currentState.step === 'awaiting_ram_memory_capacity') {

    if (message.includes('4gb') || message.includes('8gb') || message.includes('16gb')) {

        const memoryCapacity = message.includes('4gb') ? '4gb' : (message.includes('8gb') ? '8gb' : '16gb');

        state[userId] = { step: 'awaiting_ram_timings', memoryCapacity: memoryCapacity };

        return `Які таймінги ви шукаєте в ${memoryCapacity.toUpperCase()} оперативній пам'яті?
(CL16, CL18) або напишіть "пропустити", щоб побачити всі варіанти`;

    } else {

```

```

return 'Будь ласка, введіть дійсну ємність оперативної пам\`яті (4GB, 8GB, 16GB);
}
} else if (currentState.step === 'awaiting_ram_timings') {
const memoryCapacity = currentState.memoryCapacity;
if (message.includes('cl16') || message.includes('cl18') || message.includes('пропустити')) {
const timings = message.includes('cl16') ? 'cl16' : (message.includes('cl18') ? 'cl18' : 'all');
state[userId] = { step: 'default' };
return `Доступні моделі оперативної пам\`яті з ємністю ${memoryCapacity.toUpperCase()} і
таймінгами ${timings === 'all' ? 'усі таймінги' : timings.toUpperCase()}:
${ramModels[memoryCapacity][timings].join(', ')}`;
} else {
return `Будь ласка, введіть дійсні таймінги для ${memoryCapacity.toUpperCase()}
оперативної пам\`яті (CL16, CL18) або напишіть "пропустити", щоб побачити всі варіанти`;
}
} else if (currentState.step === 'awaiting_psu_type') {
if (message.includes('модульний') || message.includes('немодульний') ||
message.includes('пропустити')) {
const type = message.includes('модульний') ? 'modular' : (message.includes('немодульний') ?
'non-modular' : 'all');
state[userId] = { step: 'awaiting_psu_power', psuType: type };
return 'Будь ласка, виберіть діапазон потужності для блоку живлення (500-750, 751-1000,
1000-1500)';
} else {
return 'Будь ласка, введіть дійсний тип блоку живлення (Модульний, Немодульний) або
напишіть "пропустити", щоб побачити всі варіанти';
}
} else if (currentState.step === 'awaiting_psu_power') {
const range = message.match(/\d+-\d+/);
const psuType = currentState.psuType;
if (range && powerSupplyModels[psuType] && powerSupplyModels[psuType][range[0]]) {
state[userId] = { step: 'default' };
return `Доступні ${psuType === 'all' ? 'блоки живлення' : `${psuType} блоки живлення`}:
${powerSupplyModels[psuType][range[0]].join(', ')}`;
} else {

```

```
    return 'Будь ласка, введіть дійсний діапазон потужності для блоків живлення (500-750, 751-1000, 1000-1500)';  
  }  
}  
return 'Вибачте, я не зрозумів. Можете перефразувати?';  
}  
module.exports = { handleMessage };
```

ДОДАТОК Б

Код файлу ai_code.py

```

from dotenv import load_dotenv
import os
import pandas as pd
from llama_index.query_engine import PandasQueryEngine
from prompts import new_prompt, instruction_str, context
from note_engine import note_engine
from llama_index.tools import QueryEngineTool, ToolMetadata
from llama_index.agent import ReActAgent
from llama_index.llms import OpenAI

load_dotenv()

component_path = os.path.join("data", "components.csv")
component_df = pd.read_csv(component_path)

component_query_engine = PandasQueryEngine(
    df=component_df, verbose=True, instruction_str=instruction_str
)
component_query_engine.update_prompts({"pandas_prompt": new_prompt})

tools = [
    note_engine,
    QueryEngineTool(
        query_engine=component_query_engine,
        metadata=ToolMetadata(
            name="component_data",
            description="this gives information about component at warehouse, you can search by
any field",
        ),
    ),
]

class ContextAwareReActAgent(ReActAgent):
    def __init__(self, tools, llm, verbose, context, max_iterations, memory=None):
        super().__init__(tools=tools, llm=llm, verbose=verbose, context=context,
max_iterations=max_iterations, memory=memory)
        self.history = []

    def query(self, prompt):
        full_prompt = self._build_full_prompt(prompt)
        response = super().query(full_prompt)
        self.history.append((prompt, response))
        return response

```

```
def _build_full_prompt(self, prompt):
    history_str = "\n".join([f"Q: {q}\nA: {a}" for q, a in self.history])
    return f"{history_str}\nQ: {prompt}"

llm = OpenAI(model="gpt-3.5-turbo")
agent = ContextAwareReActAgent(tools, llm=llm, verbose=True, context=context,
max_iterations=20, memory=None)

def request_ai_with_tool(prompt):
    result = agent.query("prompt: "+prompt + "Final task: Response in prompt language")
    return result
```