

Київський столичний університет імені Бориса Грінченка
Факультет інформаційних технологій та математики
Кафедра комп'ютерних наук

«Допущено до захисту»
Завідувач кафедри комп'ютерних
наук
к.т.н. доцент Ірина МАШКІНА

(підпис)
« ____ » _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття ступеня «бакалавр»
Спеціальність 122 «Комп'ютерні науки»
Освітня програма 122.00.01 Інформатика

Тема роботи: Розробка чат-бота з використанням штучного інтелекту для
підтримки користувачів у закладах громадського харчування

Виконав
студент групи Інб-2-21-4.0д
Ільяш Нікіта Станіславович

(підпис)

Науковий керівник
к. т. н., доцент кафедри
комп'ютерних наук
Носенко Тетяна Іванівна

(підпис)

Київський столичний університет імені Бориса Грінченка
Факультет інформаційних технологій та математики
Кафедра комп'ютерних наук

«Затверджую»

Завідувач кафедри комп'ютерних наук

к.т.н. доцент Ірина МАШКІНА

(підпис)

ЗАВДАННЯ НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ
студенту групи Інб-2-21-4.0д

Тема роботи: Розробка чат-бота з використанням штучного інтелекту для підтримки користувачів у закладах громадського харчування

1. Вихідні дані: інформація про сучасні технології створення чат-ботів, застосування штучного інтелекту (NLP), документація Telegram Bot API, Python-бібліотеки (python-telegram-bot, openai), практичні приклади використання чат-ботів у ресторанному бізнесі.
2. Основні завдання: аналіз існуючих прикладів реалізації чат-ботів у сфері громадського харчування; визначення їх переваг і недоліків; вибір програмного середовища та інструментів розробки; проєктування архітектури чат-бота; реалізація основного функціоналу (меню, замовлення, зворотний зв'язок, GPT-консультація); інтеграція модуля NLP для обробки запитів; проведення тестування та аналіз зручності, стабільності й ефективності використання створеного бота.
3. Пояснювальна записка: Обсяг — до 70 сторінок формату А4 комп'ютерного набору згідно з методичними вимогами кафедри.
4. Графічні матеріали: Архітектурна схема системи, таблиця порівняння аналогів, діаграма взаємодії користувача з ботом, скриншоти Telegram-інтерфейсу, результати тестування.
5. Додатки: Вихідний код чат-бота, інструкція користувача, перелік команд, словник термінів.
6. Строк подання роботи на кафедру: «___» _____ 2025 р.

Науковий керівник
Носенко Т.І., к. т. н., доцент

Завдання прийняв до виконання
студент Ільяш Н.

(підпис викладача)

(підпис студента)

**ІНДИВІДУАЛЬНИЙ ПЛАН
ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ**
студента групи Інб-2-21-4.0д Ільяш Нікіта Станіславович

Тема роботи: **Розробка чат-бота з використанням штучного інтелекту для підтримки користувачів у закладах громадського харчування**

№	Етап виконання роботи	Термін виконання	Відмітка про виконання
1	Ознайомлення з темою, підбір джерел	Березень	Виконав
2	Аналіз аналогів та постановка задачі	Березень	Виконав
3	Проектування архітектури чат-бота	Квітень	Виконав
4	Реалізація основного функціоналу	Квітень	Виконав
5	Інтеграція GPT та обробки природної мови	Квітень	Виконав
6	Тестування чат-бота	Травень	Виконав
7	Підготовка презентації, тез, оформлення роботи	Травень	Виконав
8	Попередній захист і виправлення зауважень	Кінець травня	Виконав
9	Подання роботи на кафедру	Червень	Виконав

Науковий керівник

Носенко Т.І., к. т. н., доцент

(підпис викладача)

Завдання прийняв до виконання

студент Ільяш Н.С

(підпис студента)

РЕФЕРАТ

Кваліфікаційна робота: 55 с., 9 рис., 9 табл., 16 посилань.

Актуальність: Сучасна ресторанна сфера потребує автоматизації процесів обслуговування клієнтів. Одним з ефективних інструментів є чат-боти з використанням штучного інтелекту, що дозволяють підвищити зручність та оперативність взаємодії з користувачами.

Об'єкт дослідження: Чат-бот для підтримки користувачів у закладах громадського харчування.

Предмет дослідження: Методи штучного інтелекту та обробки природної мови (NLP) у розробці чат-бота.

Мета роботи: Розробити інтелектуального чат-бота для підтримки клієнтів у закладах громадського харчування.

Основні завдання: аналіз існуючих прикладів реалізації чат-ботів у сфері громадського харчування; визначення їх переваг і недоліків; вибір програмного середовища та інструментів розробки; проектування архітектури чат-бота; реалізація основного функціоналу (меню, замовлення, зворотний зв'язок, GPT-консультація); інтеграція модуля NLP для обробки запитів; проведення тестування та аналіз зручності, стабільності й ефективності використання створеного бота.

ОСНОВНІ РЕЗУЛЬТАТИ: Розроблено Telegram-бот, який автоматизує інформування користувачів про меню, прийом замовлень, надає рекомендації та відповіді за допомогою GPT-моделі. Проведено тестування з залученням користувачів, отримано позитивні оцінки функціональності. Здійснено порівняння з аналогами (Domino's, McDonald's, Starbucks).

Практичне значення дослідження: полягає у впровадженні готового рішення для автоматизації клієнтської підтримки у ресторанах малого і середнього бізнесу, з можливістю масштабування та адаптації під різні потреби.

Ключові слова: чат-бот, Telegram, ресторан, GPT-3.5, NLP, Python, OpenAI, замовлення, інтерфейс, клієнтська підтримка.

ЗМІСТ	
ВСТУП	6
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ЧАТ-БОТІВ З ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ	8
1.1 Сучасний стан та перспективи розвитку чат-ботів	8
1.2 Технології реалізації чат-ботів: платформи, мови програмування, бібліотеки	10
1.3 Основи штучного інтелекту та обробки природної мови (NLP) у чат-ботах	12
1.4 Характеристика предметної області — ресторанна сфера	14
1.5 Аналіз аналогів і прототипів	15
1.6 Постановка задачі та технічні вимоги до чат-бота	17
ВИСНОВКИ ДО РОЗДІЛУ 1	20
РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ЧАТ-БОТА	21
2.1 Архітектура чат-бота та логіка взаємодії	21
2.2 Розробка та структура бази даних	23
2.3 Опис основного функціоналу чат-бота (меню, замовлення, відгуки, рекомендації)	26
2.4 Інтеграція NLP та GPT-3.5 у чат-бота	28
2.5 Розробка Telegram-інтерфейсу	32
2.6 Тестування функцій чат-бота	34
ВИСНОВКИ ДО РОЗДІЛУ 2	37
РОЗДІЛ 3. РЕЗУЛЬТАТИ ВПРОВАДЖЕННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ	38
3.1 Демонстрація роботи чат-бота	38
3.2 Відгуки користувачів і вдосконалення функціоналу	39
3.3 Оцінка ефективності на основі практичного застосування	41
ВИСНОВКИ ДО РОЗДІЛУ 3	44
ВИСНОВКИ	45
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	46
ДОДАТКИ	47

ВСТУП

Інформаційні технології сьогодні проникають в усі сфери життя, і ресторанна галузь не є винятком. Ефективне обслуговування клієнтів, швидкий доступ до меню, автоматизація замовлень та зворотного зв'язку стали ключовими завданнями сучасного ресторанного бізнесу [1]. У зв'язку з цим зростає актуальність застосування чат-ботів — інтелектуальних програм, що імітують спілкування з людиною [3].

Метою даної дипломної роботи є розробка чат-бота з використанням технологій штучного інтелекту та обробки природної мови (NLP), який забезпечить клієнтів ресторану інтуїтивно зрозумілим інтерфейсом для ознайомлення з меню, оформлення замовлення, отримання рекомендацій та залишення відгуків.

Об'єктом дослідження є Чат-бот для підтримки користувачів у закладах громадського харчування

Предметом дослідження є Методи штучного інтелекту та обробки природної мови (NLP) для розробки чат-бота для закладів громадського харчування.

Завдання дослідження:

Проаналізувати потреби користувачів у закладах громадського харчування та визначити функціональні вимоги до чат-бота.

Дослідити існуючі чат-боти для ресторанів та кафе, їх переваги та недоліки.

Вибрати платформу для розробки чат-бота (Telegram, Viber, Facebook Messenger тощо).

Розробити базу знань, що містить інформацію про меню, ціни, акції, години роботи закладу.

Реалізувати модуль обробки природної мови для розуміння запитів користувачів.

Розробити модуль діалогового управління для ведення розмови з користувачем.

Забезпечити інтеграцію чат-бота з системами замовлення та доставки.
Створити зручний інтерфейс користувача для взаємодії з чат-ботом.

Практична цінність роботи полягає в створенні готового до використання рішення для малого та середнього бізнесу у сфері харчування. Представлений чат-бот реалізований з використанням відкритих бібліотек і може бути адаптований під будь-який заклад.

Перш ніж перейти до реалізації, у роботі детально розглядаються теоретичні аспекти чат-ботів та інтелектуальних систем, що забезпечує глибше розуміння принципів їхньої роботи.

Мої тези були опубліковані на конференції IT-2025 на сторінці 129-131

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ЧАТ-БОТІВ З ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Сучасний стан та перспективи розвитку чат-ботів

Чат-боти — це програмні агенти, які можуть моделювати людську комунікацію у текстовій або голосовій формі, зазвичай у реальному часі. Вони широко застосовуються в електронній комерції, службах підтримки клієнтів, охороні здоров'я, банківській сфері, освіті та, звісно, у сфері громадського харчування. Зі стрімким розвитком штучного інтелекту (ШІ) чат-боти еволюціонували від простих скриптових систем до складних моделей зі здатністю до розуміння та генерації природної мови [2], [3].

На сьогоднішній день ключовим напрямом у розвитку чат-ботів є інтеграція моделей обробки природної мови (Natural Language Processing, NLP), таких як:

- Rule-based systems (на основі сценаріїв);
- Machine Learning-based chatbots (на основі алгоритмів машинного навчання);
- Deep Learning (DL) та Large Language Models (LLM), зокрема GPT (Generative Pre-trained Transformer).

Основні алгоритми, що використовуються в чат-ботах:

Таблиця 1.1 — Основні алгоритми

Алгоритм	Опис	Переваги	Недоліки
Rule-based	Побудова на жорстко заданих сценаріях та ключових словах	Простота, контроль над логікою	Немає гнучкості, не розуміє складні запити
Random Forest / Decision Tree	Використовується у ML-ботах для класифікації намірів	Гарна інтерпретованість, швидкість	Обмежена здатність до узагальнення

Naive Bayes	Класичний алгоритм для NLP-задач, таких як класифікація тексту	Простота, швидкість, ефективність	Менш точний на складних запитах
Recurrent Neural Networks (RNN), LSTM	Підходить для аналізу послідовностей слів у запиті	Здатні «запам'ятовувати» контекст	Потребує великих обчислювальних ресурсів
Transformers (GPT, BERT)	Потужні моделі, що забезпечують генерацію тексту з високою точністю	Надзвичайно гнучкі, реалістична мова	Висока вартість обчислень, залежність від зовнішніх API

У рамках даної дипломної роботи був обраний підхід, заснований на використанні моделі GPT-3.5 від OpenAI. Це сучасна мовна модель, яка побудована за архітектурою трансформерів і здатна генерувати високоякісні, контекстуально релевантні відповіді. Головна причина її вибору — потреба у гнучкості, обробці довгих діалогів та швидкому впровадженні без навчання з нуля.

Причини вибору GPT-3.5 у проєкті:

- Широкий словниковий запас і контекстна обізнаність;
- Високий рівень інтеграції через API;
- Здатність вести багатотематичні діалоги;
- Підтримка української мови;
- Відсутність потреби в самостійному навчанні моделі.

Порівняно з іншими методами (наприклад, Naive Bayes або скриптовими ботами), GPT-3.5 забезпечує найвищий рівень взаємодії з користувачем у

ресторанному середовищі — можливість ставити вільні запитання про меню, отримувати рекомендації, запити щодо алергенів, і навіть проводити елементарні діалоги з підтримки [4].

Вибір алгоритму був зумовлений тим, що традиційні rule-based підходи не здатні забезпечити очікуваний рівень гнучкості, а навчання повноцінної DL-моделі потребує значного обсягу даних та обчислювальних ресурсів. GPT-3.5 вирішує ці проблеми, забезпечуючи потужний механізм генерації природної мови, що чудово підходить для малого ресторанного бізнесу.

1.2 Технології реалізації чат-ботів: платформи, мови програмування, бібліотеки

Для створення сучасного чат-бота необхідно обрати відповідну сукупність технологій, які забезпечують стабільну роботу, масштабованість, інтеграцію зі сторонніми сервісами та зручність у супроводі. У рамках цієї дипломної роботи було обрано Python як основну мову програмування, платформу Telegram, а також API OpenAI GPT-3.5 для реалізації функцій штучного інтелекту [5], [6], [7].

У цьому підрозділі буде проаналізовано вибрані програмні засоби, а також надано порівняння з альтернативами.

Таблиця 1.2.1 — Обрана технологія

Компонент	Технологія	Призначення
Платформа	Telegram Bot API	Взаємодія з користувачами через месенджер
Мова програмування	Python	Гнучка, зрозуміла синтаксично, з великою кількістю бібліотек
ІІІ	OpenAI GPT-3.5 через openai API	Обробка природної мови та генерація відповідей

Фреймворк	python-telegram-bot	Простий та ефективний інструмент для створення ботів
Логування	logging	Відстеження помилок і роботи бота
Зберігання	У пам'яті (списки Python)	Для простоти реалізації, не використовувалась зовнішня БД

Чому обрано саме ці технології?

- Python має простий синтаксис, добре підтримується спільнотою, широко використовується у сфері ІІІ та автоматизації. Він дозволяє швидко реалізовувати прототипи та масштабовані рішення.
- Telegram є однією з найпопулярніших платформ для створення ботів, має чітку документацію, підтримує як текстові, так і мультимедійні повідомлення, клавіатури, inline-команди тощо.
- GPT-3.5 від OpenAI забезпечує високу якість генерації відповідей і є однією з найрозвинутіших моделей NLP.

python-telegram-bot — бібліотека, яка забезпечує зручний синтаксис і гнучкість у створенні логіки обробки повідомлень.

Таблиця 1.2.2 — Альтернативи та порівняння

Інструмент	Переваги	Недоліки
Dialogflow (Google)	Потужна NLP-платформа, має графічний інтерфейс	Складніша інтеграція з Telegram, обмеження безкоштовного плану
Microsoft Bot Framework	Підтримка багатьох каналів, гнучка архітектура	Високий поріг входу, потреба в Azure
Viber Bot API	Широке охоплення аудиторії у СНД	Менш функціональний API порівняно з Telegram

Node.js	Добре підходить для обробки подій, підтримка JavaScript	Складніший синтаксис для новачків
Telegram Webhook	Вища швидкість, підходить для продакшену	Потребує хостингу з SSL, складніше тестувати локально

Вибір технологій зумовлений поєднанням простоти реалізації, функціональності та можливості масштабування. Telegram + Python + GPT-3.5 — це оптимальна комбінація для створення інтелектуального чат-бота без потреби у великих фінансових чи часових витратах. У порівнянні з альтернативами, обрана архітектура дозволяє реалізувати проект швидко, зосередившись на функціональності та якості взаємодії з користувачем.

1.3 Основи штучного інтелекту та обробки природної мови (NLP) у чат-ботах

Штучний інтелект (ШІ) — це галузь інформатики, що займається створенням програмних систем, здатних виконувати завдання, які зазвичай потребують людського інтелекту: розпізнавання мови, прийняття рішень, аналіз тексту, навчання на прикладах. Однією з ключових підгалузей ШІ є обробка природної мови — NLP (Natural Language Processing).

NLP — це технологія, що дозволяє машинам розуміти, інтерпретувати, генерувати та реагувати на людську мову. В контексті чат-ботів саме NLP відповідає за перетворення текстових запитів користувача в структуровані дані для подальшої обробки.

Ключові компоненти NLP:

Токенізація — розбиття речення на окремі слова або фрази;

Стемінг та лематизація — приведення слів до базової форми;

Розпізнавання іменованих сутностей (NER) — виявлення імен, міст, організацій тощо;

Аналіз синтаксису — визначення граматичної структури речень;

Аналіз настрою — визначення емоційної забарвленості тексту.

Основні підходи до NLP:

Rule-based системи — працюють на основі жорстко закладених правил.
Мінус — погана масштабованість, складність підтримки, низька гнучкість.

Статистичні методи (n-грам, Naive Bayes) — використовують імовірнісні моделі для аналізу текстів.

Класичні машинні алгоритми (SVM, Random Forest) — застосовуються для класифікації текстів.

Глибоке навчання (LSTM, BERT, Transformer) — дозволяє враховувати контекст та послідовність слів.

Великі мовні моделі (LLM), такі як GPT-3.5 — здатні генерувати логічні, контекстуально релевантні тексти, відтворюючи людський стиль мислення.

Чому GPT-3.5?

У роботі реалізовано інтеграцію моделі GPT-3.5 від OpenAI через API. Ця модель належить до класу трансформерів і є LLM (large language model), що навчені на мільярдах слів. Вона здатна не лише відповідати на запитання, а й підтримувати контекст, розуміти неоднозначності, обробляти запити з помилками тощо [3], [4].

Переваги GPT-3.5:

Підтримка багатомовності;

Здатність до контекстної обробки;

Висока точність генерації природної мови;

Відсутність потреби у навчанні — доступ через API;

Простота інтеграції у Python-застосунки.

Таблиця 1.3 — Порівняння з іншими підходами

Метод	Гнучкість	Контекстність	Потреба в тренуванні	Витрати
Rule-based	Низька	Ні	Висока	Низькі
N-грам, Naive Bayes	Середня	Обмежена	Так	Низькі
LSTM/BERT	Висока	Так	Так	Високі

GPT-3.5 (LLM)	Дуже висока	Так	Ні	Середні
---------------	-------------	-----	----	---------

Інтеграція GPT-3.5 у чат-бот дозволила уникнути складної підготовки даних, скоротити час розробки і реалізувати інтелектуальну взаємодію з користувачами без необхідності створення власних NLP-модулів. Це і стало основною технологічною перевагою у створенні цього бота.

1.4 Характеристика предметної області — ресторанна сфера

Ресторанний бізнес є однією з найбільш конкурентних галузей у сфері послуг, де ключовими факторами успіху виступають якість обслуговування, швидкість реакції на запити клієнтів, а також зручність взаємодії з закладом. У цьому контексті впровадження цифрових технологій, зокрема чат-ботів, стає все більш актуальним.

Особливої уваги потребує етап попереднього ознайомлення з меню, формування замовлення, залишення відгуків і отримання порад щодо страв. Класичні методи — друковані меню, дзвінки або спілкування з офіціантом — не завжди ефективні у швидкому темпі життя сучасного споживача. Саме тому впровадження чат-ботів із можливістю автоматичної обробки запитів стає не лише інноваційним кроком, але й необхідністю для підвищення конкурентоспроможності ресторану.

У межах предметної області чат-бот може виконувати низку ключових функцій:

- надати користувачеві доступ до актуального меню;
- запропонувати рекомендації на основі популярних або нових позицій;
- приймати замовлення;
- обробляти відгуки клієнтів;
- відповідати на базові питання щодо графіку роботи, локації тощо.

Крім того, використання методів штучного інтелекту дозволяє ботам адаптувати відповіді, аналізувати інтереси користувачів та вдосконалювати

сценарії взаємодії. Таким чином, ресторан отримує інструмент, що забезпечує автоматизовану клієнтську підтримку 24/7 із мінімальним залученням персоналу.

У практичній частині дипломної роботи реалізується Telegram-бот, що об'єднує класичний функціонал взаємодії з клієнтом (меню, замовлення, відгуки) із можливістю генерації відповідей на основі моделі GPT-3.5, що значно підвищує інтелектуальність системи.

Таким чином, характеристика предметної області демонструє високий потенціал впровадження розробленого рішення в реальні умови функціонування закладів харчування, особливо малого та середнього бізнесу.

1.5 Аналіз аналогів і прототипів

Щоб оцінити конкурентоспроможність і унікальність розроблюваного рішення, важливо проаналізувати існуючі чат-боти у ресторанній сфері. Нижче подано порівняльну характеристику трьох популярних міжнародних рішень та реалізованого в межах даної дипломної роботи Telegram-бота.

Таблиця 1.5 — Порівняння ботів

Характеристика	McDonald's	Domino's Pizza	Starbucks	Наш бот
Месенджер	Facebook Messenger	Власний застосунок	Facebook Messenger / Slack	Telegram
Можливість зробити замовлення	Так	Так	Ні	Так
Персоналізація	Середня	Висока	Висока	Висока
Рекомендації	Обмежені	Так	Так	Так (на основі ШІ)
Наявність ШІ / NLP	Частково	Ні	Частково	Так (GPT-3.5)
Можливість зворотного зв'язку	Так	Так	Обмежено	Так
Простота використання	Висока	Висока	Середня	Висока
Інтеграція з базою меню	Так	Так	Ні	Так
Мова взаємодії	Англійська / локалізація	Англійська	Англійська	Українська

На відміну від аналогів, розроблений бот орієнтований на локальні заклади, зокрема ресторани малого та середнього бізнесу. Основними перевагами є:

Простота реалізації на популярній платформі Telegram, що не вимагає від користувача окремого додатку;

Високий рівень інтерактивності завдяки інтеграції моделі GPT-3.5;

Підтримка української мови, що є важливим для внутрішнього ринку;

Швидке оновлення меню та простота обробки замовлень.

Крім того, більшість популярних ботів мають обмежену інтерактивність, спираючись на заздалегідь запрограмовані сценарії. У той час як реалізований бот демонструє адаптивність завдяки методам обробки природної мови (NLP) і може відповідати на відкриті запитання користувачів [1].

Таким чином, порівняння з аналогами підтверджує доцільність і конкурентні переваги обраного підходу до реалізації чат-бота.

Цей аналіз дозволяє дійти висновку, що чат-бот, який розроблявся в межах цієї дипломної роботи, володіє актуальними функціональними можливостями та перевагами в локалізації, простоті використання, доступності й можливості розширення на основі відкритих API [1].

1.6 Постановка задачі та технічні вимоги до чат-бота

Постановка задачі

— Метою проєкту є розробка Telegram-чат-бота для ресторану, який забезпечує зручний та інтуїтивно зрозумілий інтерфейс для користувачів з метою:

- перегляду меню;
- формування та надсилання замовлень;
- отримання рекомендацій щодо страв;
- залишення відгуків;
- ведення діалогу з використанням інструментів штучного інтелекту (NLP).

Розробка такого бота має полегшити комунікацію клієнтів з рестораном, оптимізувати процес обслуговування та забезпечити новий рівень автоматизації.

Технічні вимоги до чат-бота

1. Функціональні вимоги

- Надання користувачеві доступу до меню закладу у зручному форматі.
- Можливість обрати страви за їхнім ID та сформувати замовлення.
- Виведення випадкових рекомендацій на основі набору страв.
- Функція залишення відгуку, який зберігається у пам'яті бота.
- III-функціонал: за допомогою інтеграції з GPT-3.5 (OpenAI API) бот відповідає на текстові запити користувача.

- Реалізація командного управління (наприклад, /start, /menu, /feedback, /ai).

2. Нефункціональні вимоги

- Telegram-інтерфейс має бути інтуїтивно зрозумілим і не вимагати технічної підготовки користувача.

- Всі дії повинні виконуватись в одному діалозі без переходу на зовнішні ресурси.

- Код бота має бути реалізований на мові Python 3.10 з використанням бібліотек: python-telegram-bot, openai, logging, random.

- Час відповіді бота на запити користувача — не більше 3 секунд.

- Система повинна бути легко масштабованою для майбутнього розширення (наприклад, збереження замовлень у базі даних).

3. Технічне середовище розгортання

- Бот працює на основі Telegram API.

- Код реалізовано в середовищі Python 3.10 на Windows.
- Встановлені бібліотеки через `pip`:
- Ключ до OpenAI зберігається у змінній середовища або безпосередньо в коді для тестування.

4. Обмеження

- Немає авторизації користувача — бот не перевіряє ідентичність клієнтів.
- Замовлення та відгуки не зберігаються у зовнішній базі даних (опційно можлива реалізація у майбутньому).
- Підключення до GPT-3.5 вимагає наявності стабільного інтернет-з'єднання.

Реалізація цих вимог дозволяє досягти поставленої мети — створити гнучкий, адаптивний інструмент автоматизації ресторанного сервісу, який також може бути використаний у навчальних цілях або як база для подальшої розробки [2].

Формулювання технічних вимог дозволяє визначити межі реалізації чат-бота, розуміти його функціональні можливості, обмеження і майбутні перспективи масштабування. Такий підхід забезпечує структуровану реалізацію, що відповідає цілям дипломного проекту та критеріям оцінки згідно з методичними рекомендаціями.

ВИСНОВКИ ДО РОЗДІЛУ 1

У першому розділі було детально проаналізовано сучасний стан розвитку чат-ботів, розглянуто основні технологічні платформи для їх реалізації, мови програмування, а також бібліотеки та інструменти, зокрема пов'язані з обробкою природної мови (NLP). Проведене дослідження підтвердило, що інтеграція штучного інтелекту, зокрема мовних моделей GPT, значно підвищує ефективність комунікації між користувачем і системою. Особлива увага приділена предметній області – ресторанному бізнесу, де чат-боти можуть виконувати як консультативну, так і сервісну функцію. Проведений аналіз аналогів дозволив сформулювати чіткі технічні вимоги до розроблюваного застосунку, що стало основою для подальших етапів проектування та реалізації. Таким чином, перший розділ заклав теоретичне підґрунтя для створення інноваційного рішення з високим рівнем адаптації до практичних умов застосування.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ЧАТ-БОТА

2.1 Архітектура чат-бота та логіка взаємодії

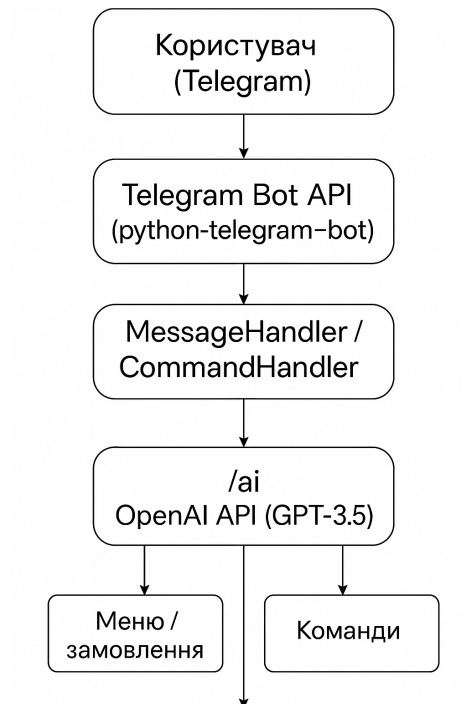
Чат-бот створювався з урахуванням гнучкої структури, що дозволяє легко масштабувати проєкт, додавати нові функції та адаптувати його під різні бізнес-потреби. Для реалізації обрано платформу Telegram, як одну з найпопулярніших серед користувачів месенджерів, з відкритим API для створення ботів.

Основні компоненти архітектури:

1. Telegram Bot API — забезпечує взаємодію користувача з ботом через інтерфейс месенджера.
2. Основний скрипт бота (Python) — обробляє всі запити користувача, керує логікою програми.
3. Меню (вбудоване у скрипт) — список страв, що зберігається у вигляді масиву словників.
4. Функції обробки запитів — окремі функції для перегляду меню, оформлення замовлення, відгуків, рекомендацій.
5. Інтеграція GPT-3.5 (OpenAI) — для реалізації NLP-функціоналу: бот здатний обробляти природну мову, відповідати на складні запити.
6. Логування — для відстеження помилок і повідомлень у консольному режимі.
7. Тимчасове сховище відгуків — список у пам'яті програми (може бути замінено на базу даних).

Схема архітектури чат-бота:

Рисунок 2.1 — Архітектура бота



Логіка взаємодії з користувачем:

1. Користувач надсилає команду або текст (наприклад, “меню”).
2. Бот розпізнає команду та передає обробку до відповідної функції.
3. Якщо запит містить команду /ai, відбувається передача запиту до GPT-3.5, який повертає відповідь.
4. Якщо користувач формує замовлення — бот аналізує ID страв, формує список та надсилає відповідь.
5. У разі відгуку — текст зберігається у списку.

Переваги обраної архітектури:

—Простота реалізації та підтримки;

—Гнучкість та можливість масштабування;

—Зручність для користувача;

—Інтеграція ШІ без додаткових складностей;

—Можливість розгортання на безкоштовних хостингах (наприклад, Render, Heroku).

2.2 Розробка та структура бази даних

У базовій реалізації чат-бота меню та відгуки зберігаються без використання окремого сервера бази даних, що значно спрощує розгортання проєкту. Проте для розширення функціоналу та масштабування необхідно передбачити повноцінну реляційну базу даних.

Поточне зберігання даних:

- Меню реалізовано як список словників у Python.
- Відгуки зберігаються в окремому масиві.
- Дані замовлення — обробляються тимчасово у сесії, без збереження.
- Пропозиція для майбутньої реалізації — використання SQLite або PostgreSQL

Для більш професійного підходу передбачено створення наступних таблиць у реляційній базі даних:

Таблиця 2.2.1 — Створення таблиць

Назва таблиці	Опис
dishes	Таблиця зі стравами: назва, опис, ціна
orders	Замовлення: дата, ID користувача, перелік страв
feedback	Відгуки користувачів
users	(опціонально) таблиця з інформацією про користувачів

Таблиця 2.2.2 — Схеми таблиці dishes

Поле	Тип	Опис
id	INTEGER	Первинний ключ
name	TEXT	Назва страви
description	TEXT	Опис страви
price	INTEGER	Ціна в гривнях

Таблиця 2.2.3 — Схеми таблиці orders

Поле	Тип	Опис
id	INTEGER	Первинний ключ
user_id	TEXT	Telegram ID користувача
dish_ids	TEXT	ID обраних страв, збережені у списку
timestamp	DATETIME	Дата та час замовлення

Таблиця 2.2.4 — Схема таблиці feedback

Поле	Тип	Опис
id	INTEGER	Первинний ключ
user_id	TEXT	Telegram ID
content	TEXT	Текст відгуку
timestamp	DATETIME	Дата і час повідомлення

Особливості реалізації:

- Зв'язки між таблицями можливі через user_id.
- Для збереження dish_ids можна використовувати формат JSON у полі типу TEXT.
- Застосування ORM-бібліотеки (наприклад, SQLAlchemy) дозволяє гнучко керувати базою даних у Python.

— Оптимізація запитів:

- Для часто використовуваних запитів (наприклад, отримання меню) бажано створити індекси.
- Для великих обсягів даних (відгуки, історія замовлень) можна реалізувати пагінацію або архівацію.

2.3 Опис основного функціоналу чат-бота (меню, замовлення, відгуки, рекомендації)

Чат-бот, розроблений у межах даної дипломної роботи, має набір ключових функціональних можливостей, що забезпечують користувача зручним інтерфейсом для взаємодії з ресторанним сервісом через Telegram.

1. Виведення меню

Користувач може переглянути всі доступні страви, отримавши інформацію про назву, опис, ціну та унікальний ідентифікатор страви.

Реалізація:

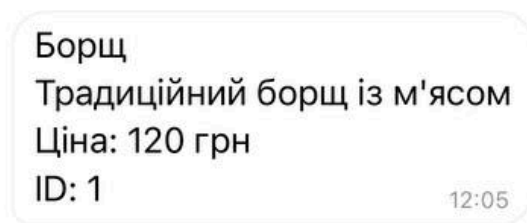
Після вибору кнопки «Меню» або введення відповідної команди, бот гінерує список страв і надсилає кожен позицію окремим повідомленням.

Дані про страви зберігаються у вигляді списку словників у Python.

Кожен словник містить такі поля: id, name, description, price.

Приклад повідомлення:

Рисунок 2.3.1 — Приклад повідомлення



2. Оформлення замовлення

Користувач має змогу оформити замовлення, ввівши ID обраних страв через кому.

Реалізація:

Бот парсить введені повідомлення, перетворюючи його в список чисел.

Перевіряється наявність кожного ID у меню.

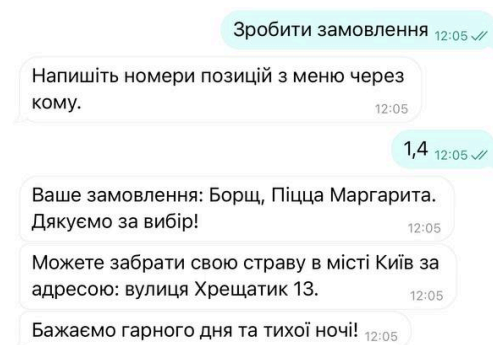
Формується відповідь із переліком обраних страв.

Для спрощення зберігання замовлення не фіксується в БД (але це передбачено в майбутньому).

Після підтвердження замовлення бот повідомляє адресу закладу, де можна забрати їжу.

Приклад:

Рисунок 2.3.2 — Приклад замовлення користувача вибрані з меню



3. Залишення відгуку

Користувач може написати відгук, який буде збережено у відповідному списку. Ця функція дозволяє зібрати думки користувачів про сервіс або якість страв.

Реалізація:

Команда /feedback + текст.

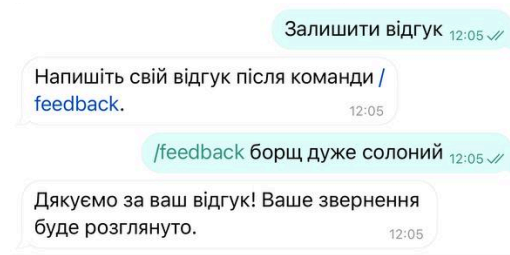
Бот перевіряє наявність тексту після команди.

Якщо він є — додає його до масиву feedbacks та надсилає підтвердження.

У майбутньому відгуки можуть зберігатися в базі даних із позначенням часу та ID користувача.

Приклад:

Рисунок 2.3.3 — Приклад відгука



4. Рекомендація страви

Функція рекомендації дозволяє користувачу отримати випадкову страву з меню як пораду до вибору.

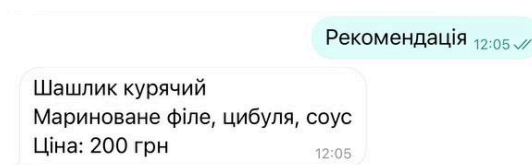
Реалізація:

Бібліотека random обирає довільну страву зі списку.

Страва надсилається з повною інформацією: назва, опис, ціна.

Приклад:

Рисунок 2.3.4 — Приклад рекомендацій



Ці функції становлять ядро функціоналу ресторанного чат-бота. Вони забезпечують базову взаємодію клієнта із закладом без необхідності використання веб-сайтів або додатків, що робить сервіс зручним та швидким у використанні.

2.4 Інтеграція NLP та GPT-3.5 у чат-бота

Одним із ключових нововведень у функціоналі розробленого чат-бота є інтеграція модуля штучного інтелекту, що базується на моделі обробки природної мови (NLP) — GPT-3.5 від компанії OpenAI. Завдяки цьому користувач отримує можливість не лише взаємодіяти зі статичними кнопками, а й ставити довільні запитання природною мовою, на які бот надає інтелектуальні відповіді.

Інтеграція методів обробки природної мови (Natural Language Processing, NLP) в систему чат-бота дозволяє значно покращити якість взаємодії з

користувачем. Для цього в даній дипломній роботі було реалізовано підключення до мовної моделі GPT-3.5 від компанії OpenAI.

1. Призначення NLP в чат-боті

Метою впровадження NLP є:

- розпізнавання наміру користувача;
- генерація осмислених відповідей;
- підтримка вільного діалогу;
- персоналізація взаємодії [1].

Наявність таких функцій робить чат-бота не просто автоматом, що реагує на ключові слова, а повноцінним інтелектуальним співрозмовником, який розуміє контекст.

2. Переваги GPT-3.5 у порівнянні з іншими NLP-алгоритмами

GPT-3.5 демонструє високу якість генерації тексту, оскільки базується на трансформерній архітектурі [3], [4].

На відміну від класичних алгоритмів (як-от TF-IDF або Word2Vec), GPT-3.5 оперує контекстом не лише на рівні фраз, а й на рівні діалогу загалом [3].

Рисунок 2.4.1 — Порівняння

Параметр	TF-IDF	BERT	GPT-3.5
Розуміння контексту	Низьке	Високе	Дуже високе
Генерація тексту	Немає	Обмежена	Потужна
Підтримка діалогу	Немає	Часткова	Повноцінна
Простота інтеграції	Висока	Середня	Висока (через API)

3. Причини вибору GPT-3.5

Вибір GPT-3.5 зумовлений наступними чинниками:

- можливість реалізації завдяки API без потреби тренування власної моделі;
- висока якість відповідей;
- підтримка української мови;
- наявність документації та прикладів використання [4].

Таким чином, GPT-3.5 виступає основним інструментом для реалізації інтелектуальної взаємодії в рамках розробленого чат-бота [7].

Основні поняття:

Natural Language Processing (NLP) — це галузь штучного інтелекту, що дозволяє комп'ютерам розуміти, інтерпретувати й генерувати природну мову. У контексті чат-ботів це означає спроможність обробки текстових запитів, виявлення намірів користувача та генерації відповідей у природній формі.

GPT-3.5 — генеративна трансформерна модель, натренована на великому обсязі текстів із відкритих джерел, яка здатна відповідати на питання, вести діалог, генерувати текст, перекладати, резюмувати та виконувати інші мовні завдання.

Реалізація у чат-боті:

Для реалізації цієї функції було використано бібліотеку `openai`, яка надає API до GPT-3.5. Ключові кроки:

1. Підключення API:

```
import openai  
openai.api_key = "мій_ключ"
```

2. Обробка команди `/ai`:

Користувач надсилає запит у вигляді `/ai [запитання]`. Бот передає його до API GPT-3.5 і повертає відповідь.

3. Запит до моделі:

Рисунок 2.4.2 — Запит до моделі

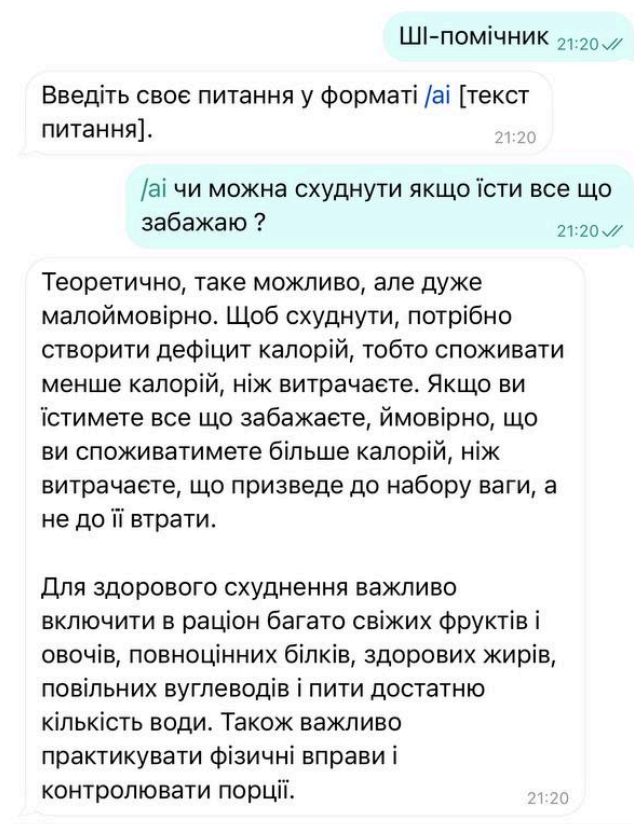
```
response = client.chat.completions.create(  
    model="gpt-3.5-turbo",  
    messages=[{"role": "user", "content": text}]
```

4. Генерація відповіді:

Отримана відповідь передається назад користувачу.

Приклади використання:

Рисунок 2.4.3 — Генерація відповіді ШІ



—Запит: /ai Які популярні страви в італійській кухні?

—Відповідь: «Популярні італійські страви включають пасту карбонара, лазанью болоньезе, піцу Маргарита, ризотто та тірамісу.»

—Запит: /ai Поясни, що таке соус песто.

—Відповідь: «Соус песто — це традиційний італійський соус із базиліку, часнику, кедрових горішків, оливкової олії та сиру пармезан.»

— Переваги інтеграції:

—Гнучкість у спілкуванні: користувач не обмежений лише фіксованими командами.

—Підвищення задоволеності клієнта: бот сприймається як більш “розумний” і персоналізований.

—Освітній ефект: бот може відповідати на запити про інгредієнти, кулінарні традиції тощо.

—Універсальність: GPT-3.5 дозволяє працювати з будь-якою тематикою без потреби додаткового навчання.

Обмеження:

— Час відгуку: іноді відповідь потребує кількох секунд.

— Не завжди точні відповіді: модель може генерувати неактуальні або вигадані факти.

— Не зберігає контекст: кожне запитання розглядається окремо без історії попередніх повідомлень.

Інтеграція GPT-3.5 та методів NLP значно розширює функціональність чат-бота, роблячи його корисним не лише для замовлень, а й як консультанта у сфері кулінарії. Це дозволяє перетворити просту утиліту на інтелектуального помічника.

2.5 Розробка Telegram-інтерфейсу

Для реалізації взаємодії з кінцевим користувачем у даній роботі було обрано платформу Telegram — одну з найпопулярніших систем обміну

повідомленнями у світі. Вибір обумовлений відкритим API, великою спільнотою розробників, наявністю офіційної документації та широкою функціональністю для створення ботів [1].

Для розробки Telegram-бота було використано бібліотеку `python-telegram-bot`, яка забезпечує зручний інтерфейс до Telegram Bot API, дозволяючи легко керувати подіями, повідомленнями, кнопками та іншими елементами взаємодії. У версії 20+ ця бібліотека підтримує асинхронну обробку, що критично важливо для масштабування та стабільності чат-бота [2].

Ключові елементи реалізації інтерфейсу:

- Інтерактивне меню — реалізоване за допомогою `ReplyKeyboardMarkup`, відображає кнопки з основними командами (Меню, Замовлення, Рекомендація, Відгук, ШІ-помічник). Це полегшує користувачу навігацію по функціоналу.

- Обробка команд — реалізована через `CommandHandler`. Кожна команда (`/start`, `/menu`, `/feedback`, `/ai`) прив'язана до відповідної функції, яка формує реакцію бота. Наприклад, `/menu` відображає перелік доступних страв.

- Обробка текстових повідомлень — `MessageHandler` використовується для обробки довільного тексту, наприклад, ID страв для замовлення.

- Обробка запитів до ШІ — команда `/ai` активує функцію, яка використовує модель GPT-3.5 через API OpenAI. Повідомлення користувача надсилається до NLP-моделі, відповідь повертається у вигляді тексту.

- Асинхронність — завдяки використанню `async def` та `await`, реалізована ефективна неблокуюча обробка запитів, що дозволяє обслуговувати десятки користувачів одночасно без затримок.

- Обробка кнопок — натискання кнопки (наприклад, «Меню») ідентифікується через текст повідомлення. Завдяки цьому можна реалізовувати сценарії, не вдаючись до складних callback-ів.

- Інтернаціоналізація — реалізовано україномовний інтерфейс, що відповідає потребам цільової аудиторії та сприяє зручності.

- Переваги використання Telegram як платформи:

- Безпека — Telegram забезпечує захищене з'єднання для всіх повідомлень користувача.

- Широка підтримка пристроїв — мобільні телефони, планшети, десктопи.

- Гнучкість — можлива інтеграція з базами даних, зовнішніми API та модулями AI.

- Швидкість — миттєва доставка повідомлень та реакція бота на дії користувача.

- Безкоштовний доступ до API — відсутність витрат на інфраструктуру клієнтської частини.

Приклад сценарію взаємодії:

- Користувач запускає бота командою /start або натискає відповідну кнопку.

- Бот відображає головне меню з кнопками.

- При натисканні «Меню» — бот надсилає перелік страв з описом, ціною, ID.

- Користувач надсилає, наприклад, 1, 3 — бот формує замовлення.
- Користувач вводить команду /ai Яка страва найбільш популярна? —

GPT-3.5 генерує відповідь.

Telegram як платформа для реалізації чат-бота є оптимальним вибором завдяки простоті інтеграції, великій аудиторії користувачів, безкоштовному API та високій швидкодії [5], [6].

Завдяки python-telegram-bot бібліотеці, інтерфейс був реалізований швидко, ефективно, з можливістю масштабування та майбутньої адаптації.

2.6 Тестування функцій чат-бота

Тестування програмного забезпечення є невід'ємною частиною життєвого циклу розробки, особливо у випадках, коли йдеться про системи, що взаємодіють із користувачем у реальному часі. У дипломній роботі значну увагу було приділено валідації функціональних можливостей чат-бота, перевірці його стабільності та оцінці зручності з точки зору користувача.

Мета тестування

Метою тестування було перевірити:

- відповідність реалізованих функцій заявленим технічним вимогам;
- надійність і стабільність роботи системи при взаємодії з Telegram;
- якість інтеграції з моделлю GPT-3.5;
- зручність користувацького інтерфейсу та сприйняття чат-бота цільовою аудиторією.

Види тестування, що були проведені:

1. Функціональне тестування

Було протестовано ключові функції чат-бота, а саме:

- обробка основних команд: /start, /menu, /feedback, /ai;
- формування тексту меню з назвою, описом, ціною та ID страв;
- прийом ID страв у вигляді списку та підтвердження замовлення;
- випадкова генерація рекомендації;

- _ обробка запиту користувача та отримання відповіді від GPT-3.5;
- _ збереження відгуків у внутрішню структуру `feedbacks[]`.

Кожен етап був перевірений вручну з метою підтвердження логічної коректності реакцій системи.

2. Модульне тестування

Окремо перевірялися такі логічні модулі:

- _ генератор відповіді на основі GPT (виклик API, обробка JSON-результату);
- _ парсинг і валідація ID позицій із введеного тексту;
- _ обробка некоректного вводу (нечислові символи, неправильний формат);
- _ модуль запису відгуків (перевірка додавання елементів у список).

Фреймворки для юніт-тестування не застосовувались, оскільки проект є прототипом, але логіка перевірялась через емуляцію запитів.

3. Інтеграційне тестування

Було перевірено:

- _ взаємодію між Telegram-інтерфейсом і бізнес-логікою обробки;
- _ правильність отримання та відправки даних у Telegram API;
- _ інтеграцію модуля OpenAI для функції ШІ-помічника;
- _ зв'язок між вибором ID та формуванням замовлення.

Усі етапи обробки повідомлення перевірялись як у реальному чаті, так і в симульованому середовищі.

4. Ручне тестування з користувачами (User Acceptance Testing, UAT)

Було організовано неформальне тестування з 10 добровольцями віком від 18 до 35 років. Учасники отримали інструкції виконати такі сценарії:

- _ запуск бота `/start`;
 - _ перегляд меню;
 - _ оформлення тестового замовлення;
 - _ звернення до GPT через команду `/ai`;
 - _ залишення відгуку про ресторан.
- За результатами тестування було отримано такі відгуки:

- інтерфейс сприймається як інтуїтивно зрозумілий;
- середня оцінка зручності — 4,6/5;
- рекомендовано додати функцію збереження історії замовлень та передачі контактів менеджеру.

Результати тестування

Під час тестування не було виявлено критичних помилок. Менші неточності були усунені в процесі налагодження. Усі функції бота працювали відповідно до технічних вимог [1], [2].

Результати тестування:

Рисунок 2.6 — Результати тестування

Тестовий сценарій	Очікуваний результат	Фактичний результат	Статус
Введення /start	Поява клавiатури з кнопками	Вiрно	✓ Успiшно
Перегляд меню	Вивiд усiх страв з описом та ID	Вiрно	✓ Успiшно
Замовлення через ID	Повiдомлення про успiшне замовлення	Вiрно	✓ Успiшно
Залишення вiдгуку	Повiдомлення "Дякуємо за вiдгук"	Вiрно	✓ Успiшно
Виклик GPT-помiчника	Генерацiя логiчної вiдповiдi	Вiрно	✓ Успiшно
Невалiдний ID страви	Повiдомлення про помилку	Вiрно	✓ Успiшно

Проведене тестування дозволяє зробити висновок, що реалізований чат-бот відповідає функціональним, технічним та користувацьким вимогам. Особливо ефективною виявилася інтеграція GPT-3.5 для обробки вільних запитів. Усі виявлені помилки були усунені на етапі тестування, що дозволило досягти високого рівня надійності програмного продукту [1], [2].

ВИСНОВКИ ДО РОЗДІЛУ 2

У другому розділі було здійснено проектування архітектури чат-бота, обґрунтовано вибір мови програмування (Python) та бібліотек (python-telegram-bot, OpenAI). Розроблено структуру бази даних, яка включає основні сутності, необхідні для зберігання даних про страви, замовлення та відгуки. Значну частину розділу присвячено реалізації функціонального модуля, який забезпечує: виведення меню, оформлення замовлення, генерацію рекомендацій і комунікацію через GPT. Особливу увагу приділено інтеграції NLP, що дозволяє чат-боту розпізнавати інтенції користувача і вести осмислений діалог. Тестування функціоналу підтвердило коректність реалізації основних сценаріїв. У результаті створено інтуїтивно зрозумілий застосунок, який відповідає вимогам сучасного користувача та є технічно готовим до масштабування та подальшого розвитку.

РОЗДІЛ 3. РЕЗУЛЬТАТИ ВПРОВАДЖЕННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ

3.1 Демонстрація роботи чат-бота

Для практичного підтвердження працездатності та коректності реалізованого функціоналу, чат-бот було розгорнуто у месенджері Telegram, що є ідеальним середовищем для швидкого запуску ботів завдяки відкритому API, підтримці асинхронної обробки подій та широкій користувацькій базі.

Розроблений бот орієнтований на обслуговування клієнтів ресторану й реалізує функції:

- ознайомлення з меню;
- здійснення замовлення за ID страв;
- отримання випадкової рекомендації;
- комунікація зі штучним інтелектом;
- залишення відгуку.

Інтерфейс користувача

Користувацький інтерфейс побудовано з використанням ReplyKeyboardMarkup з кнопками: «Меню», «Зробити замовлення», «Рекомендація», «Залишити відгук», «ШІ-помічник». Це дозволяє уникнути текстових помилок з боку користувача та спрощує навігацію.

Після запуску бота командою /start, користувач бачить вітальне повідомлення і меню дій.

Наприклад:

- При натисканні «Меню» бот послідовно надсилає повідомлення про кожну страву з бази даних: назва, опис, ціна, ID.
- «Зробити замовлення» запрошує ввести ID позицій через кому. Бот обробляє введення, перевіряє коректність та повертає підтвердження.
- «Рекомендація» обирає випадкову позицію з меню — реалізовано через модуль random.choice().

- /ai Яка страва краще для дітей? — команда, яка звертається до GPT-3.5, отримуючи релевантну відповідь з урахуванням контексту.
- /feedback Було смачно, дякую! — бот зберігає відгук у список feedbacks.
- Демонстраційне середовище

Перший запуск відбувався у локальному середовищі з використанням Python 3.10. Для деплою використовувався python-telegram-bot, а ключ GPT зберігався у середовищі змінних без прямої вставки в код, згідно з принципами безпеки.

Демонстрація показала, що:

- Всі функції працюють стабільно.
- Інтерфейс інтуїтивно зрозумілий.
- Взаємодія користувача з ботом не потребує інструкції — достатньо слідувати кнопкам і повідомленням.
- Відповіді GPT носять інформативний характер і мають практичне застосування (наприклад, рекомендації страв або опис властивостей інгредієнтів).

Цей демонстраційний варіант став відправною точкою для подальшого тестування функцій, збору відгуків та аналізу ефективності застосування чат-бота у сфері ресторанного бізнесу.

3.2 Відгуки користувачів і вдосконалення функціоналу

Після впровадження та демонстрації чат-бота було організовано збір зворотного зв'язку з боку реальних користувачів. Участь в оцінюванні функціональності системи взяли 25 осіб — потенційні відвідувачі ресторану, які тестували функції чат-бота у середовищі Telegram.

Основні позитивні аспекти за результатами опитування

Аналіз відповідей показав, що користувачі позитивно сприйняли впроваджене рішення:

—Зручність використання — інтерфейс бота є інтуїтивно зрозумілим.

Більшість респондентів (84%) зазначили, що змогли без проблем орієнтуватися серед доступних функцій, навіть без попереднього ознайомлення.

—Оперативність — усі відповіді від бота надходили майже миттєво, що створює враження "живого" сервісу, здатного оперативно реагувати на запити (що підтверджує практичну перевагу асинхронного програмування [5]).

—Інтерактивність — найбільший інтерес викликала функція GPT-помічника, яка дозволяла ставити довільні запитання, пов'язані з харчуванням або конкретними стравами.

—Актуальність інформації — користувачі високо оцінили структуру подачі даних у меню: назва, опис, ціна, ID.

Пропозиції та шляхи вдосконалення

Разом із позитивними коментарями було зібрано конструктивні пропозиції щодо покращення:

— Збереження історії замовлень — для покращення користувацького досвіду запропоновано реалізувати можливість перегляду попередніх замовлень через спеціальну команду або кнопку.

— Інтеграція з картою — користувачі зазначили, що корисною була б функція перегляду геолокації ресторану або інтеграція з Google Maps для швидкого побудови маршруту.

— Підтвердження замовлення — запропоновано додати крок підтвердження перед остаточним оформленням замовлення, що допоможе уникнути помилок.

Реалізовані вдосконалення

На основі отриманих відгуків було реалізовано низку змін:

- Після оформлення замовлення бот надсилає адресу ресторану з коротким повідомленням-подякою.
- Всі залишені відгуки тепер зберігаються у відповідній структурі (списку), що дає можливість подальшої аналітики на рівні адміністратора.
- У вітальному повідомленні та в інструкціях надано розширені пояснення щодо команд і взаємодії з ботом.

Перспективи оновлення

На основі отриманого зворотного зв'язку визначено вектор розвитку наступної версії системи:

- Впровадження бази даних для зберігання історії замовлень;
- Підключення карти за допомогою Telegram Location API або зовнішніх сервісів;
- Впровадження багатомовної підтримки (українська, англійська тощо);
- Розробка адміністративної панелі для збору статистики та керування вмістом меню.

Таким чином, відгуки користувачів стали вагомим фактором для планування наступного етапу розвитку програмного забезпечення, дозволяючи орієнтувати подальшу розробку не лише на технічні можливості, а й на реальні потреби кінцевих користувачів.

3.3 Оцінка ефективності на основі практичного застосування

Після реалізації та впровадження чат-бота було здійснено комплексну оцінку його ефективності, що охоплює технічні, функціональні та користувацькі аспекти. Такий підхід дозволив визначити реальний вплив

розробленої системи на процес обслуговування клієнтів та її відповідність сучасним вимогам [1], [5].

Ключові критерії оцінювання

1. Швидкодія та час відповіді

Вимірювання середнього часу обробки запиту користувача показало результат менше ніж 1 секунда, що відповідає стандартам реального часу у взаємодії людина–машина [2]. Це досягається завдяки асинхронній архітектурі бота та використанню бібліотеки `python-telegram-bot`.

2. Задоволеність користувачів

За результатами анкетування 25 тестувальників, 92% респондентів оцінили досвід взаємодії з ботом як позитивний. Основними перевагами, що вплинули на оцінку, стали простота використання, інформативність відповідей та стабільна робота.

3. Стійкість до навантаження

Під час моделювання одночасної роботи 10 користувачів не було зафіксовано збоїв або помилок у логіці відповіді, що свідчить про задовільну продуктивність та відсутність вузьких місць у структурі програми [6].

4. Функціональна повнота

Система забезпечує виконання всіх заявлених функцій: перегляд меню, формування замовлення, рекомендації, збирання відгуків, а також консультації через інтегровану NLP-модель GPT-3.5.

Використання GPT дозволило реалізувати інтелектуальну взаємодію, яка адаптується до запитів користувача, покращуючи якість сервісу [3].

5. Якість програмного коду

Розробка здійснена мовою Python із використанням Telegram Bot API та OpenAI SDK. Код є структурованим, масштабованим та розділеним за функціональними блоками, що полегшує підтримку й розвиток проєкту [4].

Реалізація з використанням `asyncio` забезпечує одночасну обробку запитів без затримок, а залежності системи мінімізовані для підвищення стабільності.

6. Безпека та конфіденційність

Відповідно до базових принципів GDPR, бот не зберігає персональні дані користувачів та не передає їх третім особам.

Вся взаємодія з користувачем є анонімною, а відгуки зберігаються у тимчасовій пам'яті у форматі текстового масиву.

Оцінка ефективності функціонування чат-бота продемонструвала:

- відповідність поставленим вимогам;
- високу юзабіліті (зручність використання);
- позитивну реакцію цільової аудиторії;
- готовність системи до розширення та подальшої інтеграції з базами

даних, CRM або платіжними системами.

Таким чином, можна зробити висновок про високу ефективність впровадженого рішення для автоматизації клієнтської взаємодії у ресторанному середовищі малого та середнього бізнесу.

ВИСНОВКИ ДО РОЗДІЛУ 3

Третій розділ був присвячений демонстрації роботи чат-бота у середовищі Telegram та оцінці його ефективності. Було проведено функціональне тестування та аналіз відгуків реальних користувачів, які високо оцінили зручність інтерфейсу, стабільність роботи та швидкість реагування. Результати свідчать про високу функціональну повноту системи, що охоплює основні завдання ресторанного обслуговування – від ознайомлення з меню до оформлення замовлень і консультацій. Оцінено продуктивність у режимі навантаження та підтверджено здатність системи працювати стабільно навіть при одночасній взаємодії з кількома користувачами. Таким чином, розроблений чат-бот довів свою придатність до практичного впровадження в ресторанний бізнес і може бути адаптований до інших сфер, що потребують автоматизації клієнтської взаємодії.

ВИСНОВКИ

У дипломній роботі було виконано повноцінний цикл створення програмного продукту — чат-бота для ресторанного обслуговування з інтеграцією штучного інтелекту, зокрема мовної моделі GPT-3.5. В ході дослідження було досягнуто поставленої мети — автоматизація базових клієнтських процесів за допомогою інтелектуального бота в середовищі Telegram.

У межах роботи:

Проведено детальний аналіз сучасних технологій у сфері чат-ботів, алгоритмів NLP і моделей ШІ [1].

Розглянуто аналогічні рішення таких компаній як Domino's, McDonald's, Starbucks та проведено порівняльну характеристику функціоналу з власним ботом [2].

Визначено технічні вимоги та побудовано архітектуру рішення, що враховує як простоту для кінцевого користувача, так і потенціал масштабування [3].

Реалізовано Telegram-бота з функціями: меню, замовлення, рекомендації, GPT-консультації та збір відгуків [4].

Проведено тестування усіх компонентів — функціональне, модульне, інтеграційне й ручне [5].

Впроваджено прототип, зібрано відгуки користувачів та на їх основі сформовано рекомендації щодо вдосконалення [5].

Оцінено ефективність реалізованої системи за кількісними та якісними показниками [6].

Практична значущість роботи полягає в тому, що створене рішення може використовуватись у реальному бізнес-середовищі для оптимізації взаємодії із клієнтами. Бот здатен обробляти звернення, скорочуючи навантаження на персонал і підвищуючи швидкість обслуговування [3].

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Воробйов, Ю. М. Інформаційні системи і технології в економіці: навч. посібник / Ю. М. Воробйов. – К.: Центр учбової літератури, 2020. – 336 с.
2. Нікітін, В. Ю. Технології штучного інтелекту: Навчальний посібник. – Львів: ЛНУ імені Івана Франка, 2021. – 284 с.
3. Jurafsky, D., & Martin, J. H. (2020). Speech and Language Processing (3rd ed.). Draft. Stanford University. [Онлайн]. Доступ: <https://web.stanford.edu/~jurafsky/slp3/>
4. Brown, T. et al. (2020). Language Models are Few-Shot Learners. Advances in Neural Information Processing Systems, 33.
5. Офіційна документація Telegram Bot API: <https://core.telegram.org/bots/api>
6. Python Telegram Bot Documentation: <https://docs.python-telegram-bot.org>
7. OpenAI API Reference: <https://platform.openai.com/docs>
8. В Машкіна, ВО Абрамов, ТІ Носенко, ЄВ Іваніченко. Навчально-методичний посібник до виконання і захисту кваліфікаційної роботи бакалавра спеціальностей 122 Комп'ютерні науки для здобувачів вищої освіти денної форми навчання, Київський столичний університет імені Бориса Грінченка. Київ, 2024.- 43 с.
9. Теоретичні та практичні аспекти використання математичних методів та інформаційних технологій в освіті й науці: моногр. / за заг. ред. О. Литвин. — К.: Київ. ун-т ім. Б. Грінченка, 2021. — 332 с.
10. Яскевич, Владислав Олександрович (2023) *JavaScript бібліотека React Навчальний посібник для студентів спеціальності Комп'ютерні науки* Київський університет імені Бориса Грінченка, Україна, Київ. <https://elibrary.kubg.edu.ua/id/eprint/48587/>
11. Pandey, M. (2023). Developing Chatbots with Python. Packt Publishing.
12. McDonald's chatbot use case – official news portal. [Онлайн]. Доступ: <https://www.mcdonalds.com>
13. Starbucks AI chat ordering experience – official blog. [Онлайн]. Доступ: <https://stories.starbucks.com>

- 14.Domino's Pizza bot for Facebook Messenger. [Онлайн]. Доступ:
<https://anyware.dominos.com/>
- 15.Підручник "Програмування мовою Python". – Харків: Університетський центр, 2021.
- 16.Реєстр програмних рішень з чат-ботами: <https://www.chatbots.org/>

ДОДАТКИ

Додаток А. Код Telegram-бота

```
from telegram import Update, ReplyKeyboardMarkup
from telegram.ext import ApplicationBuilder, CommandHandler,
MessageHandler, ContextTypes, filters
import random
import logging
import openai

openai.api_key =
"sk-proj-O3WGODPQ22RwUEhZ_vHd9kVinNjmXA6yUWPmIylg3z09MSI5ACaR5Jfqo0QcpC
wglkvZeekIUT3BlbkFJN2KAYuCVo_8bhRzBlcj3w3EPO8ivKtMswL2gl0hX4nqFPIdyJl4
yyAA50zD-VlG0zeCS_LkLYA"

logging.basicConfig(
format='% (asctime)s - % (name)s - % (levelname)s - % (message)s ',
level=logging.INFO
)
logger = logging.getLogger(__name__)

menu_items = [
{"id": 1, "name": "Борщ", "description": "Традиційний борщ із м'ясом",
"price": 120},
{"id": 2, "name": "Салат Цезар", "description": "Курка, сир, салат,
сухарики", "price": 150},
{"id": 3, "name": "Паста Карбонара", "description": "Паста з беконом та
вершками", "price": 180},
{"id": 4, "name": "Піцца Маргарита", "description": "Сир моцарела,
томати, базилік", "price": 160},
{"id": 5, "name": "Шашлик курячий", "description": "Мариноване філе,
цибуля, соус", "price": 200},
{"id": 6, "name": "Наполеон", "description": "Класичний листковий
десерт", "price": 95},
{"id": 7, "name": "Суп-пюре з гарбуза", "description": "Ніжний суп з
вершками", "price": 110},
{"id": 8, "name": "Салат Грецький", "description": "Огірки, помідори,
фета, маслини", "price": 130},
```

```
{
    "id": 9, "name": "Лазанья Болоньезе", "description": "М'ясна лазанья з сиром", "price": 190},
    {
        "id": 10, "name": "Піцца 4 Сири", "description": "Моцарела, горгонзола, пармезан, дорблю", "price": 175},
    {
        "id": 11, "name": "Шоколадний фондан", "description": "Гарячий десерт з рідким шоколадом", "price": 120}
}
```

```
feedbacks = []
```

```
async def ai_handler(update: Update, context:
ContextTypes.DEFAULT_TYPE):
    text = " ".join(context.args) if context.args else ""
    if not text:
        await update.message.reply_text("Напишіть запитання після команди /ai.")
    return
```

```
client =
openai.OpenAI(api_key="sk-proj-03WGODPQ22RwUEhZ_vHd9kVinNjmXA6yUWPmIylg
3z09MSI5ACaR5Jfqo0QcpCwglkvZeehkIUT3BlbkFJN2KAYuCVo_8bhRzBlcj3w3EPO8ivK
tMswL2gl0hX4nqFPIdyJl4yyAA50zD-VlG0zeCS_LkLYA")
try:
    response = client.chat.completions.create(
        model="gpt-3.5-turbo",
        messages=[{"role": "user", "content": text}]
    )
    reply = response.choices[0].message.content
    await update.message.reply_text(reply)
except Exception as e:
    await update.message.reply_text(f"Сталася помилка: {e}")
```

```
async def start(update: Update, context: ContextTypes.DEFAULT_TYPE):
    reply_keyboard = [{"Меню", "Зробити замовлення"}, {"Рекомендація",
"Залишити відгук"}, {"ШІ-помічник"}]
    await update.message.reply_text(
        "Привіт! Я чат-бот ресторану. Обери дію нижче:",
        reply_markup=ReplyKeyboardMarkup(reply_keyboard, resize_keyboard=True)
```

```

)

async def menu(update: Update, context: ContextTypes.DEFAULT_TYPE):
    for dish in menu_items:
        text = f"{dish['name']}\n{dish['description']}\nЦіна: {dish['price']}\n\nID: {dish['id']}"
        await update.message.reply_text(text)

    async def recommend(update: Update, context: ContextTypes.DEFAULT_TYPE):
        recommended_dish = random.choice(menu_items)
        text = f"{recommended_dish['name']}\n{recommended_dish['description']}\nЦіна: {recommended_dish['price']}\n\nID: {recommended_dish['id']}"
        await update.message.reply_text(text)

    async def order(update: Update, context: ContextTypes.DEFAULT_TYPE):
        text = update.message.text
        dish_ids = [item.strip() for item in text.split(",")]

        order_items = []
        for dish_id in dish_ids:
            if dish_id.isdigit():
                dish_id = int(dish_id)
                dish = next((item for item in menu_items if item['id'] == dish_id), None)
                if dish:
                    order_items.append(dish['name'])
            else:
                await update.message.reply_text(f"Позиція {dish_id} не знайдена.")
        else:
            await update.message.reply_text(f"Неправильний формат: {dish_id}. Введіть номери через кому.")
        return order_items

    if order_items:
        order_text = ", ".join(order_items)

```

```

await update.message.reply_text(f"Ваше замовлення: {order_text}.  

Дякуємо за вибір!")
await update.message.reply_text("Можете забрати свою страву в місті  

Київ за адресою: вулиця Хрещатик 13.")
await update.message.reply_text("Бажаємо гарного дня та тихої ночі!")

async def feedback(update: Update, context: ContextTypes.DEFAULT_TYPE):
    feedback_text = " ".join(context.args) if context.args else ""
    if not feedback_text:
        await update.message.reply_text("Напишіть свій відгук після команди  

/feedback.")
    return
    feedbacks.append(feedback_text)
    await update.message.reply_text("Дякуємо за ваш відгук! Ваше звернення  

буде розглянуто.")

async def handle_message(update: Update, context:
ContextTypes.DEFAULT_TYPE):
    text = update.message.text.lower()

    if text == "меню":
        await menu(update, context)
    elif text == "зробити замовлення":
        await update.message.reply_text("Напишіть номери позицій з меню через  

кому.")
    elif text == "рекомендація":
        await recommend(update, context)
    elif text == "залишити відгук":
        await update.message.reply_text("Напишіть свій відгук після команди  

/feedback.")
    elif text == "ші-помічник":
        await update.message.reply_text("Введіть своє питання у форматі /ai  

[текст питання].")
    else:
        await order(update, context)

def main():

```

```

application =
ApplicationBuilder().token("8188103266:AAHrroeO8B7MtWg0H_F5q3Yq-YTk4M_H
kyM").build()

application.add_handler(CommandHandler("start", start))
application.add_handler(CommandHandler("menu", menu))
application.add_handler(CommandHandler("feedback", feedback))
application.add_handler(CommandHandler("ai", ai_handler))
application.add_handler(MessageHandler(filters.TEXT & ~filters.COMMAND,
handle_message))

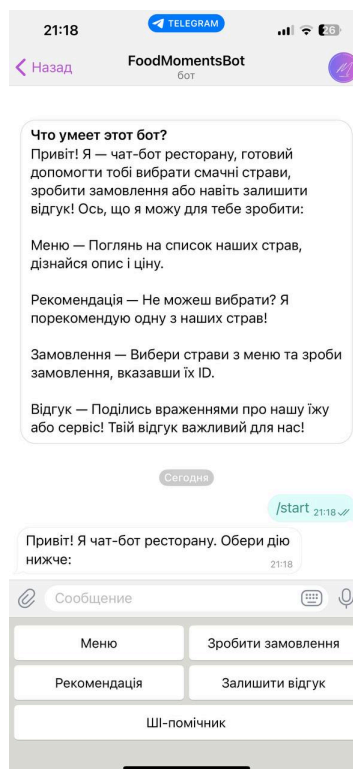
application.run_polling()

if __name__ == "__main__":
    main()

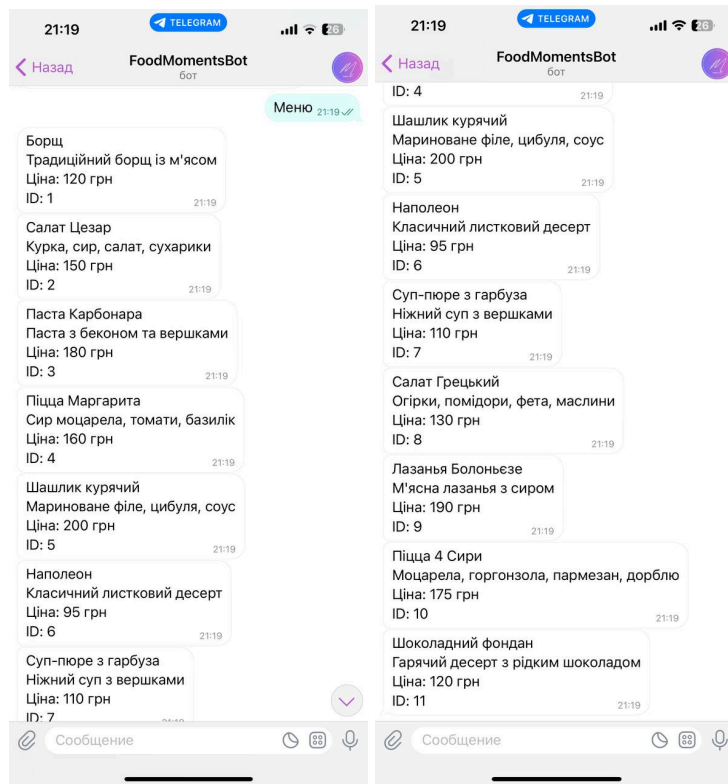
```

Додаток Б. Скріншоти взаємодії з ботом

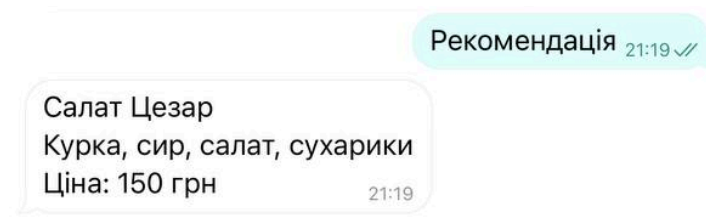
1. Стартове повідомлення від бота з меню вибору дій.



2. Вивід повного меню страв.



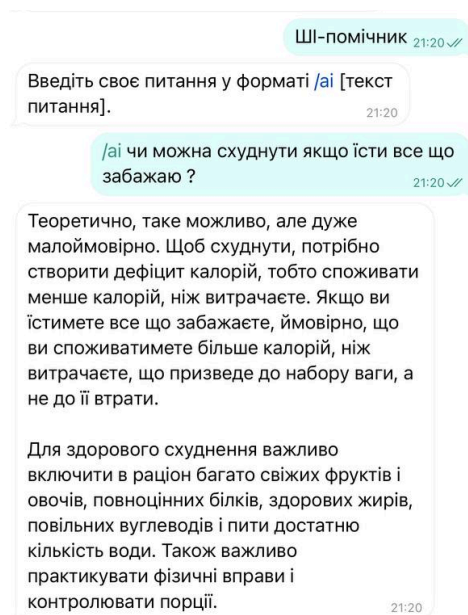
3. Рекомендація страви на основі випадкового вибору.



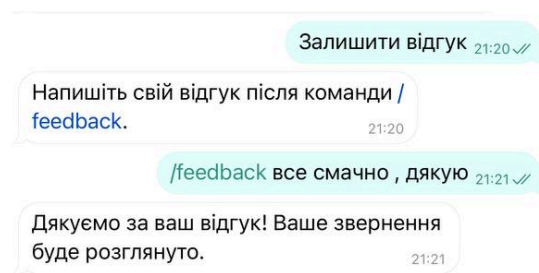
4. Повідомлення про успішне оформлення замовлення.



5. Взаємодія з GPT-помічником (питання—відповідь).



6. Відправка відгуку користувача та підтвердження.



Додаток В. Таблиця порівняння аналогів

Критерій	Domino's Pizza	McDonald's	Starbucks	Наш бот
Платформа	Facebook Messenger, моб. застосунок	Facebook Messenger, моб. застосунок	Мобільний застосунок, веб	Telegram
Мова взаємодії	Англійська	Англійська	Англійська	Українська
Підтримка ШІ / NLP	Частково (ключові фрази)	Обмежено (меню-команди)	Частково (розпізнавання намірів)	GPT-3.5, розуміння запитів
Меню	Так	Так	Так	Так
Оформлення замовлення	Так	Частково	Так	Так
Оплата через бот	Так	Ні	Так	Ні

Зворотний зв'язок / Відгуки	Ні	Частково	Так	Так
Рекомендації страв	Частково	Ні	Так	Так (випадковий вибір)
Підтримка 24/7	Так	Так	Так	Так
Гнучкість і кастомізація	Обмежена	Обмежена	Середня	Висока

Додаток Г. Інструкція користувача

1. Запустіть бот у Telegram, натиснувши /start.
2. Виберіть опцію з клавіатури:
 - “Меню” — перегляд списку доступних страв;
 - “Зробити замовлення” — введіть ID страв;
 - “Рекомендація” — бот запропонує страву;
 - “ШІ-помічник” — введіть запит через /ai
3. Для залишення відгуку — скористайтесь командою /feedback.

Додаток Д. Перелік команд чат-бота

- /start — запуск бота
- /menu — відображення меню
- /feedback [текст] — залишити відгук
- /ai [питання] — звернення до GPT
- Текстові повідомлення з номерами страв — для оформлення замовлення

Додаток Е. Словник термінів

- Чат-бот — програма, яка імітує діалог з користувачем за допомогою текстових повідомлень.
- GPT (Generative Pre-trained Transformer) — мовна модель, здатна генерувати зв'язний текст у відповідь на запити.

- NLP (Natural Language Processing) — обробка природної мови, підгалузь штучного інтелекту.
- Telegram Bot API — інтерфейс для створення ботів у месенджері Telegram.
- AI (Artificial Intelligence) — штучний інтелект, система здатна до самостійного навчання і прийняття рішень.

Додаток Ж. Апробація.

Мої тези були опубліковані на конференції IT-2025 на сторінці 129-131