

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління  
(повна назва)

Кафедра електронних обчислювальних машин  
(повна назва)

**КВАЛІФІКАЦІЙНА РОБОТА**  
**Пояснювальна записка**

Рівень вищої освіти другий (магістерський)

Інтегрована система штучного інтелекту  
в телеграм-бот для вивчення англійської мови

(тема)

Виконав:

студент II курсу, групи СПМ-22-1  
Реука К. О.  
(прізвище, ініціали)

Спеціальність 123 «Комп'ютерна інженерія»  
(код і повна назва спеціальності)

Тип програми освітньо-професійна  
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування  
(повна назва освітньої програми)

Керівник: доцент Ільїна І.В.  
(посада, прізвище, ініціали)

Допускається до захисту

В.о. зав. кафедри ЕОМ

(підпис)

(прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ комп'ютерної інженерії та управління \_\_\_\_\_

Кафедра \_\_\_\_\_ електронних обчислювальних машин \_\_\_\_\_

Рівень вищої освіти \_\_\_\_\_ другий (магістерський) \_\_\_\_\_

Спеціальність \_\_\_\_\_ 123 – Комп'ютерна інженерія \_\_\_\_\_  
(код і повна назва)

Тип програми \_\_\_\_\_ освітньо-професійна \_\_\_\_\_  
(освітньо-професійна або освітньо-наукова)

Освітня програма \_\_\_\_\_ Комп'ютерна інженерія \_\_\_\_\_  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

“ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ**

**НА КВАЛІФІКАЦІЙНУ РОБОТУ**

студенту \_\_\_\_\_ Реуці Кирилу Олеговичу \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи Інтегрована система штучного інтелекту в телеграм-бот для вивчення англійської мови

затверджена наказом по університету від “ \_\_\_\_\_ ” \_\_\_\_\_ 2023 р. № \_\_\_\_\_ 561 Ст

2. Термін подання студентом роботи до екзаменаційної комісії \_\_\_\_\_ 15 січня 2024 р.

3. Вхідні дані до роботи \_\_\_\_\_

1) документація Node.js;

2) документація фреймворку Telegraf;

3) документація REST API;

4) інтегроване середовище: Microsoft Visual Studio Code.

4. Перелік питань, що потрібно опрацювати у роботі \_\_\_\_\_

1) аналіз предметної області

2) аналіз використаних технологій

3) програмна реалізація

4) інструкція користувача

5) висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Слайд-презентація – 9 слайдів

---

---

---

---

---

---

---

---

---

---

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1 )

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз проблеми та огляд існуючих рішень		
2	Вибір технології розробки та інструментальних засобів		
3	Розробка алгоритмічного забезпечення		
4	Розробка програмних модулів		
5	Відлагодження програмних модулів		
6	Оформлення матеріалів кваліфікаційної роботи		
7	Подання кваліфікаційної роботи керівникові та її попередній захист		
8	Подання кваліфікаційної роботи на рецензування		

Дата видачі завдання 06 листопада 2023 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_  
(підпис)

доцент Ільїна І.В.  
(посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 84 с., 28 рис., 6 табл., 2 дод., 15 джерел.

ТЕЛЕГРАМ-БОТ, ЧАТ-БОТ, ІНТЕРНЕТ, СЦЕНА, СЕРВЕР, КЛІЄНТ, TELEGRAM, TELEGRAF, БАЗА ДАНИХ, ДОКУМЕНТ.

Метою кваліфікаційної роботи є створення телеграм-боту з інтегрованою системою штучного інтелекту для вивчення англійської мови. Головною особливістю бота є можливість спілкування використовуючи голосові повідомлення.

У ході виконання кваліфікаційної роботи досліджені існуючі рішення, їх переваги, недоліки та сформовано функціонал розробленого застосунку.

Для вирішення поставленого завдання використовується фреймворк Telegraf разом із Node.js для серверної частини.

## ABSTRACT

Bachelor's thesis: 84 pages, 28 figures, 6 tables, 2 appendices, 15 sources.

TELEGRAM BOT, CHAT BOT, INTERNET, SCENE, SERVER, CLIENT,  
TELEGRAM, TELEGRAF, DATABASE, DOCUMENT.

The purpose of the qualification work is to create a Telegram bot with an integrated artificial intelligence system for learning the English language. The main feature of the bot is the ability to communicate using voice messages.

During the execution of the qualification work, existing solutions have been investigated, their advantages and disadvantages have been identified, and the functionality of the developed application has been formulated.

To address the set task, the Telegraf framework is used in conjunction with Node.js for the server-side.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	8
ВСТУП .....	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	11
1.1 Основні поняття чат-боту .....	11
1.1.1 Основні завдання телеграм-ботів.....	11
1.1.2 Порівняння з мобільними додатками.....	11
1.2 Причини вивчення англійської мови .....	12
1.3 Проблеми у вивченні англійської мови.....	12
1.4 Допомога телеграм-ботів у вивченні англійської мови.....	12
1.4.1 Вивчення слів та лексики.....	12
1.4.2 Граматичні вправи .....	12
1.4.3 Аудіопрактика .....	12
1.4.4 Чат-боти для спілкування .....	12
1.4.5 Використання ботів для навчання в грі.....	12
1.4.6 Можливість отримувати консультації в будь-який час.....	12
1.4.7 Персоналізація навчання.....	12
1.5 Постановка задачі.....	<b>2Помилка! Закладку не визначено.</b>
1.5.1 Опис функціоналу бота.....	<b>2Помилка! Закладку не визначено.</b>
1.6 Очікувані результати .....	22
1.7 Методи реалізації .....	22
1.8 Аналіз існуючих рішень .....	22
1.8.1 LinguaLeo .....	23
1.8.2 DuoLingo .....	24
1.8.3 Підсумки аналізу існуючих рішень .....	24
2 АНАЛІЗ ВИКОРИСТОВУВАНИХ ТЕХНОЛОГІЙ.....	26
2.1 Огляд засобів розробки .....	25

2.2 Основні поняття штучного інтелекту .....	21
2.2.1 Natural Language Processing .....	21
2.2.2 Machine Learning.....	20
2.2.3 Технологія GPT.....	28
2.2.4 ChatGPT .....	29
2.3 JavaScript.....	26
2.4 Node.js .....	32
2.5 Управління залежностями за допомогою NPM.....	33
2.6 Telegram Bot API.....	34
2.7 Telegraf.....	30
2.8 Система контролю версій та стратегія розгалуження.....	32
2.9 Visual Studio Code .....	33
2.10 Docker.....	34
2.11 Axios .....	35
3 ПРОГРАМНА РЕАЛІЗАЦІЯ.....	37
3.1 Архітектура проекту.....	37
3.2 Використання OpenAI моделей.....	46
3.2.1 GPT-3.5 Turbo.....	46
3.2.2 Whisper-1 .....	46
3.2 Реєстрація чат-бота для Telegram Bot API .....	42
3.3 Розміщення чат-боту на платформі Heroku .....	51
4 ІНСТРУКЦІЯ КОРИСТУВАЧА .....	55
ВИСНОВКИ.....	60
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	62
ДОДАТОК А ГРАФІЧНИЙ МАТЕРІАЛ КВАЛІФІКАЦІЙНОЇ РОБОТИ .....	69
ДОДАТОК Б ПРОГРАМНА РЕАЛІЗАЦІЯ.....	71

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І  
ТЕРМІНІВ

БД – база даних

API – Application program interface

HTTP – HyperText Transfer Protocol

NPM – менеджер пакетів Node.JS

ORM – Object Relation Mapping

REST – Representational State Transfer

URL – Uniform Resource Locator

## ВСТУП

У наш час повсякденне існування людини нерозривно переплетено з комп'ютерними технологіями. З урахуванням стрімкого розвитку світу та постійного появи нових ідей та можливостей, вплив ІТ-світу на інші галузі та наше щоденне життя є надзвичайно суттєвим. Ще недавно активно використовувались комп'ютерні програми, додатки та веб-сайти, але зараз чітким лідером є чат-боти.

В месенджері telegram чат-боти виступають як важливий інструмент, сприяючи автоматизації рутинних процесів. Вони здатні автоматично обробляти та відправляти повідомлення, виконуючи різноманітні завдання, доступні для кожного користувача telegram. Творці месенджера telegram відзначаються особливою підтримкою чат-ботів, надаючи розробникам широкі можливості для творчості у цьому напрямі. Це дало незалежним програмістам можливість створювати інтерактивні інтерфейси з широким спектром можливостей.

Зараз ця тематика залишається дуже актуальною і популярною, оскільки кожна людина витрачає значний час в інтернеті, зокрема в соціальних мережах і месенджерах. Взаємодія через повідомлення часто зручніша для людей, ніж телефонні дзвінки. Такий попит пояснює поширену практику використання чат-ботів для бронювання квитків, замовлення їжі, отримання новин, проведення тестів на знання і багато іншого, особливо у сфері онлайн.

Цей взаємодіючий ландшафт інформаційних технологій постійно зростає, а нові технології виходять на передній план, розширюючи можливості використання чат-ботів. Ще більше інноваційних функцій та додаткових можливостей надають користувачам можливість насолоджуватися високотехнологічним середовищем.

Продовжуючи думку про роль чат-ботів, можна відзначити, що вони стають не лише інструментом ефективності, але й джерелом розваг та розвитку. Розширені можливості штучного інтелекту дозволяють створювати чат-ботів, які можуть адаптуватися до індивідуальних потреб користувачів, роблячи

взаємодію з ними більш персоналізованою та задовільною.

Однак разом із зростанням функціональності чат-ботів з'являються нові виклики, пов'язані з безпекою та приватністю, що потребує удосконалення заходів захисту даних. Постійне вдосконалення алгоритмів та впровадження нових технологій стають необхідністю для забезпечення безпеки та конфіденційності користувачів у світі, де використання чат-ботів стає все більш важливим і поширеним аспектом нашого життя.

З розвитком технологій чат-боти стають не лише інструментом зручної комунікації, але й ключовим елементом у сфері бізнесу. Вони впроваджуються в різноманітні галузі, починаючи від обслуговування клієнтів і закінчуючи веденням маркетингових кампаній.

Чат-боти можуть автоматизувати процеси обробки замовлень, відповідати на запитання покупців та надавати персоналізовані рекомендації. Важливою перевагою є їхній раціональний робочий час та здатність обробляти багато запитань одночасно, поліпшуючи ефективність бізнес-процесів. Однак з використанням чат-ботів постає виклик у забезпеченні високого рівня інтерактивності та якості обслуговування, аби забезпечити задоволення клієнтів та підтримувати позитивний імідж компанії.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Основні поняття телеграм-боту

Телеграм-боти представляють собою автоматизовані програми, створені для взаємодії з користувачами в месенджері Telegram. Їхнє функціональне призначення далеко не обмежується простим обміном текстовими повідомленнями. Ці боти можуть виконувати низку завдань та виконувати функції, що полегшують та розширюють життя користувачів, і це робить їх надзвичайно важливими в сучасному світі зв'язку та автоматизації [1].

Телеграм-боти стали надзвичайно популярними серед користувачів Telegram завдяки їхній унікальній можливості розширювати функціональність месенджера та надавати користувачам нові способи взаємодії з інформацією та сервісами. Вони використовуються як в особистих цілях, так і в бізнесі для автоматизації процесів обслуговування клієнтів та надання інформації.

Телеграм-боти можуть бути створені розробниками або компаніями за допомогою спеціальних інструментів, наданих Telegram, і вони розвиваються з кожним днем, надаючи користувачам все більше можливостей для зручної та цікавої комунікації в месенджері.

Telegram має декілька переваг [2]:

- безліч серверів по всьому світу, що дозволяє досягти високої швидкості та безпеки. Завдяки цьому у Telegram найшвидша доставка повідомлень у порівнянні з іншими конкурентами;
- відкритий протокол, що дозволяє стороннім розробникам використовувати напрацювання команди Telegram;
- конфіденційність – всі чати являються зашифрованими, а повідомлення знищуються через деякий час;
- практично повна відсутність обмежень за розміром чату та файлів, що відправляються. В одному повідомленні можна надсилати скільки завгодно

файлів розміром до 1,5 ГБ;

- немає підписок і реклами;
- групові чати до 30000 учасників, а також тематичні канали на кшталт груп у соціальних мережах;
- легкість у використанні, мінімум споживаних ресурсів пристрою, а також повна кросплатформність;
- самовидалення повідомлення. Можна вказати проміжок часу, коли повідомлення буде видалено автоматично;
- синхронізувати повідомлення між усіма пристроями.

### 1.1.1 Основні завдання телеграм-ботів

Завдання, які на даний час виконують телеграм-боти:

- відповіді на запитання: Телеграм-боти можуть відповідати на запитання користувачів на різні теми, надаючи інформацію або рекомендації. Наприклад, бот може надавати прогноз погоди, перекладати тексти, або навіть надавати консультації з різних галузей;
- надсилання інформації: Телеграм-боти можуть розсилати повідомлення з важливою інформацією або новинами. Це дозволяє користувачам отримувати оновлення безпосередньо в месенджері;
- автоматизація завдань: Боти можуть виконувати рутинні завдання за запитом користувача. Наприклад, вони можуть реєструвати замовлення, планувати події, або нагадувати про деякі дії;
- спілкування з іншими сервісами: Телеграм-боти можуть інтегруватися з іншими онлайн-сервісами і взаємодіяти з ними. Наприклад, бот може здійснювати платежі, перевіряти статус доставки товарів або бронювати квитки;
- розважальні функції: В деяких випадках телеграм-боти створені для розваги. Вони можуть пропонувати ігри, вікторини, гумористичний контент або створювати меми;
- служба підтримки: Багато компаній використовують телеграм-ботів для

надання підтримки клієнтам. Боти можуть відповідати на запитання про товари та послуги, надавати інструкції з використання продукту або надавати контактну інформацію для спілкування з живими операторами.

Ці завдання лише кілька прикладів того, що можуть робити телеграм-боти. Вони можуть бути створені для будь-якої галузі та виконувати різноманітні функції в залежності від потреби користувачів і бізнес-сценаріїв.

### 1.1.2 Порівняння з мобільними додатками

Порівнюючи телеграм-боти і мобільні додатки, слід зазначити, що обидва мають свої переваги та недоліки. Телеграм-боти підходять для швидких запитів та послуг, вони зручні для використання на поточний момент. Мобільні додатки, натомість, можуть надавати більше функцій та інтерфейс, призначений для різних завдань. Вибір залежить від конкретних потреб користувача.

Таблиця 1.1 – Порівняння телеграм ботів та мобільних додатків

Критерії	Телеграм-боти	Мобільні додатки
Доступність	+	-
Невелика кількість ресурсів	+	-
Інтеграція з іншими системами	+	+/-

### 1.2 Причини вивчення англійської мови

Вивчення англійської мови в сучасному світі вважається важливим і незамінним навичкою з багатьох причин. Англійська є мовою міжнародного спілкування, вона відкриває доступ до світових можливостей, сприяє робочому та особистому росту. Знання англійської полегшує доступ до академічних ресурсів та наукової інформації. Ця мова є ключем до розуміння світової культури через фільми, музику, літературу та інше. Вона також полегшує

подорожі та спілкування з місцевими жителями, допомагає розуміти технології та інновації, сприяє міжкультурному розумінню, розвитку когнітивних навичок та кар'єрному зростанню. Знання англійської мови надає соціальні переваги та сприяє побудові нових соціальних зв'язків.

### 1.3 Проблеми у вивченні англійської мови

Одна з найбільших проблем, з якими стикаються люди під час вивчення англійської мови, полягає у тому, що вони можуть навчитися граматиці, збагатити словниковий запас і вивчити правила вимови, але водночас відчувати відсутність можливості практикувати ці навички в реальних розмовних ситуаціях. Ця проблема особливо актуальна для тих, хто не має доступу до носіїв мови або англомовних оточуючих.

Однією з ключових складнощів є відсутність знайомих, з якими можна вільно спілкуватися на англійській мові. Регулярне розмовляння з носіями мови або іншими студентами допомагає вдосконалювати навички говоріння і навіть засвоювати нові мовні вирази та виправляти помилки.

Без практики розмовного спілкування, будь-який вивчений матеріал дуже швидко може виходити з пам'яті. Якщо ви не маєте можливості використовувати англійську мову у реальних ситуаціях, то навіть найкраща освіта і величезний словниковий запас можуть бути непродуктивними. Пам'ять потребує постійного нагадування, а говоріння з іншими - це відмінний спосіб зробити це ефективним.

Таким чином, одним із способів подолання цієї проблеми може бути використання інтерактивних засобів навчання, таких як телеграм бот для вивчення англійської мови. Цей бот може створити можливість для студентів практикувати розмовні навички та переглядати вивчений матеріал у формі спілкування, що робить процес вивчення більш ефективним і наближеним до реальних розмовних ситуацій.

## 1.4 Допомога телеграм-ботів у вивченні англійської мови

Вивчення іноземної мови, зокрема англійської, може бути важким завданням, але сучасні технології і інтернет роблять цей процес значно ефективнішим і цікавішим. Телеграм-боти є однією з інноваційних інструментів, які допомагають у вивченні англійської мови.

### 1.4.1 Вивчення слів та лексики

Багато телеграм-ботів спеціалізуються на розвитку лексичних навичок. Вони надсилають користувачам щоденні слова та фрази разом із перекладом та прикладами вживання. Це допомагає постійно розширювати словниковий запас та засвоювати нові вирази. Користувачі можуть навчати і запам'ятовувати нові слова в зручній для них спосіб.

Телеграм-боти можуть пропонувати користувачам систематичні словники та набори слів за темами. Наприклад, ви можете вибрати словник зі специфічними словами для подорожей, бізнесу, англійської кухні тощо. Це дозволяє фокусуватися на словах і виразах, які є корисними в певних сферах життя або роботи.

Телеграм-боти можуть включати вправи та тести на запам'ятовування слів. Вони можуть запитувати користувача переклад або визначення слів, а також перевіряти рівень їх знань. Це допомагає закріпити нові слова та вирази в пам'яті.

Телеграм-боти часто надсилають слова та фрази через певні інтервали часу, щоб забезпечити систематичне повторення. Це важливо для закріплення нових слів та їхнього використання в реальних розмовах.

Багато телеграм-ботів дозволяють користувачам відстежувати свій прогрес у вивченні слів та лексики. Вони надають статистику про кількість вивчених слів, результати тестів та інформацію про покращення навичок.

### 1.4.2 Граматичні вправи

Навички граматики є важливим елементом вивчення будь-якої мови, включаючи англійську. Телеграм-боти можуть допомогти користувачам поліпшити свої граматичні навички через різні граматичні вправи та завдання.

Телеграм-боти також можуть надавати граматичні вправи для вдосконалення навичок граматики. Вони можуть включати тестування на вживання часів, правильний порядок слів у реченнях та інші аспекти граматики. Це допомагає користувачам вивчити правила та практикувати їх у реальних виразах.

Багато телеграм-ботів пропонують вправи, спрямовані на вивчення і вдосконалення граматичних часів, таких як Present Simple, Past Continuous, або Future Perfect. Вправи можуть включати в себе вибір правильного часу для заповнення пропусків у реченнях, а також створення власних речень за вказаними правилами часів.

Деякі телеграм-боти надають вправи, які допомагають користувачам розуміти і використовувати правильний порядок слів у реченнях. Це особливо важливо для тих, хто має труднощі зі складанням логічних речень.

Вивчення англійської граматики також включає в себе знання про правильну будову речень та використання пунктуації. Телеграм-боти можуть надсилати користувачам завдання на переписування речень або виправлення граматичних помилок у вже існуючих реченнях.

Деякі телеграм-боти мають можливість проводити тести на граматику, які допомагають користувачам визначити свій рівень знань і знайти слабкі місця у граматичній підготовці. Після завершення тесту користувачі можуть отримати докладні результати та рекомендації щодо подальших кроків.

Граматичні вправи можуть надсилатися користувачам регулярно, створюючи систематичний підхід до навчання. Постійне повторення допомагає закріпити правила та покращити навички.

Деякі телеграм-боти можуть адаптувати граматичні вправи до рівня знань

та потреб кожного користувача. Це дозволяє кожному навчати граматику на своєму власному рівні та в темах, які їх цікавлять.

Одна із переваг використання телеграм-ботів полягає в тому, що вони надають миттєві відповіді та пояснення на запитання користувачів. Це допомагає розуміти граматичні правила та виправляти помилки швидко.

### 1.4.3 Аудіопрактика

Аудіопрактика є ключовою частиною вивчення будь-якої мови, оскільки вона допомагає розвивати навички аудіювання, вимови та розуміння мови. Телеграм-боти можуть надавати користувачам можливість вдосконалити ці аспекти через різні аудіозаписи та завдання.

Деякі телеграм-боти можуть надсилати аудіозаписи з вимовою слів і фраз. Це корисно для тих, хто бажає покращити навички аудіювання і вимови. Слухання англійської мови в різних ситуаціях допомагає зрозуміти носіїв мови краще і навчити вимовляти слова правильно.

Також вони можуть надсилати користувачам аудіо книги та аудіозаписи текстів книг, статей або новин. Це розширює слуховий словник та допомагає розуміти реальний контекст використання слів та фраз.

Можуть підписувати користувачів на розсилки новин або подкастів англійською мовою. Це допомагає слухати актуальну інформацію та розмови на різноманітні теми, розвиваючи навички аудіювання та розуміння.

Деякі телеграм-боти мають функцію слухання та відповіді. Вони надсилають користувачам аудіозаписи з питаннями або завданнями, на які користувачі повинні відповісти в усній формі. Це допомагає розвивати навички відповіді на запитання та покращує усну взаємодію.

Деякі телеграм-боти співпрацюють з різними навчальними ресурсами, надаючи користувачам можливість вивчати англійську мову за допомогою аудіо уроків та відео уроків. Це може бути важливою частиною навчання, оскільки вона дозволяє бачити та слухати реальних викладачів.

#### 1.4.4 Чат-боти для спілкування

Ще одним корисним аспектом телеграм-ботів є можливість спілкуватися з ними в текстовому форматі. Деякі чат-боти призначені для розмови і відповідають на питання користувачів англійською мовою. Це може бути корисно для практики реального спілкування та вдосконалення навичок розмовної мови.

Чат-боти створюють можливість для користувачів практикувати свої навички спілкування в реальному часі. Користувачі можуть почати чат з ботом і спілкуватися з ним, вживаючи різноманітні слова і вирази англійською мовою. Боти можуть надавати відповіді та коригувати помилки, допомагаючи користувачам розвивати навички розмовної мови.

Деякі чат-боти можуть спеціалізуватися на різних темах, дозволяючи користувачам практикувати мову в контексті, що їх цікавить. Наприклад, можна вибрати бота, який працює на тему подорожей, бізнесу, гастрономії або будь-якої іншої галузі, яка вас цікавить.

#### 1.4.5 Використання ботів для навчання в грі

Використання ботів для навчання в грі - це цікавий та захопливий спосіб покращити навички англійської мови. Боти можуть впроваджувати навчальні елементи в ігри та розваги, що робить процес навчання більш привабливим та ефективним.

Багато телеграм-ботів розроблені у формі ігор. Користувачі можуть грати у різні лінгвістичні гри, де вони повинні відповідати на питання, перекладати слова або розв'язувати кросворди. Це розважає і стимулює вивчення англійської мови.

Деякі боти створюють рольові ігри, в яких користувачі грають в ролі персонажів та взаємодіють з іншими персонажами у віртуальному світі. Це надає можливість використовувати англійську мову для спілкування, прийняття

рішень та вирішення завдань.

Використання ботів для навчання в грі робить процес вивчення більш цікавим та забавним. Користувачі можуть відчувати, що вони грають, а не просто навчаються, що може бути додатковою мотивацією для вивчення.

#### 1.4.6 Можливість отримувати консультації будь-який час

Один із переваг використання телеграм-ботів полягає в тому, що вони доступні цілодобово. Користувачі можуть задавати питання та отримувати відповіді в будь-який час, коли це для них зручно. Це особливо корисно для тих, хто вивчає англійську мову у вільний від роботи час.

#### 1.4.7 Персоналізація навчання

Поєднання навчання та персоналізації, особливо у вивченні англійської мови, може зробити процес більш ефективним та захопливим. Персоналізація навчання вигідна, оскільки вона дозволяє користувачам вивчати мову відповідно до їхніх індивідуальних потреб, рівня знань і стилю навчання.

Деякі телеграм-боти мають можливість персоналізації навчання, адаптуючи вправи та матеріали до рівня знань користувача. Це допомагає кожному вивчати мову відповідно до своїх потреб та темпу.

Першим кроком до персоналізації навчання є аналіз потреб користувача. Телеграм-боти можуть запитувати користувачів про їхні мети вивчення мови, рівень знань, обрані теми та інші індивідуальні фактори, що впливають на навчання.

На основі аналізу, боти можуть підбирати відповідний контент для користувачів. Це може включати в себе словники, вправи, граматичні завдання, текстовий чи аудіо-контент відповідно до індивідуальних потреб користувача.

Персоналізовані боти можуть створювати індивідуальні навчальні плани для користувачів. Ці плани можуть включати в себе розклади вправ, завдання на

певний період часу, рекомендації для самонавчання та інші дії, які допомагають досягти конкретних цілей.

## 1.5 Постановка задачі

Проект передбачає розробку та реалізацію інтерактивного телеграм бота, призначеного для вивчення англійської мови. Бот буде виступати в ролі розмовного партнера, надаючи користувачам можливість взаємодії, покращення навичок використання англійської мови та надання знань з граматики.

### 1.5.1 Опис функціоналу бота

При розробці телеграм боту повинні виконатися такі функціональні вимоги, як:

- бот повинен мати можливість вести розмову з користувачем на різні теми, використовуючи відповідний лексичний спектр для різних рівнів володіння мовою;
- бот повинен мати можливість аналізувати речення користувачів на предмет граматичних помилок та надавати відповідні корекції;
- бот може надавати користувачам приклади вживання слів та фраз у різних контекстах, щоб допомогти їм розуміти, як правильно використовувати слова.

## 1.6 Очікувані результати

В результаті проведення дослідження очікуються такі результати:

- розробка та реалізація телеграм бота для вивчення англійської мови;
- покращення навичок користувачів у використанні англійської мови на різних рівнях складності;
- автоматична перевірка граматичних помилок та надання відповідних

корекцій;

- надання користувачам інформації та пояснень щодо граматичних тем та правил англійської мови.

### 1.7 Методи реалізації

Для досягнення цих цілей передбачається використання мов програмування, які підтримують створення телеграм ботів (наприклад, JavaScript та бібліотека Telegraf). Розробка інтерфейсу для взаємодії з користувачами буде включати в себе аналіз тексту та генерацію відповідей на основі навчальних даних та алгоритмів штучного інтелекту.

Ця робота передбачає аналіз та тестування результатів взаємодії з користувачами для оцінки ефективності бота та коригування його функціоналу за необхідності.

### 1.9 Аналіз існуючих рішень

Для визначення переліку необхідного функціоналу ми провели аналіз популярних існуючих рішень в Інтернет-просторі на сучасний момент. Ці рішення демонструють схожість за базовим функціоналом, оскільки вони включають основні можливості для ефективного обліку фінансів. Однак кожен сервіс відрізняється від інших завдяки своїм унікальним функціям. У нашій роботі детально розглянуті найпопулярніші на сьогоднішній день сервіси з даної області, визначаючи їхні особливості та переваги.

#### 1.9.1 LinguaLeo

LinguaLeo – це освітня платформа, доступна для використання на різних пристроях та операційних системах, включаючи iOS, Android та веб-версію. Орієнтована на вивчення іноземних мов, вона пропонує користувачам низку

інструментів для покращення навичок вивчення мов.

Платформа дозволяє вивчати слова та вирази в контексті, використовуючи тексти з реальних джерел та відеоматеріали. Окрім цього, LinguaLeo пропонує інтерактивні вправи, тестування та можливість ведення особистого словника. Однією з ключових особливостей є гейміфікація процесу вивчення, що робить навчання більш захоплюючим.

Незважаючи на ці переваги, LinguaLeo має деякі недоліки. Зокрема, деякі користувачі вказують на обмежену кількість безкоштовного контенту та неякісний переклад деяких слів чи фраз. Також, для повного доступу до всіх функцій іноді потрібна підписка, що може виявитися неприємною для деяких користувачів. Однак, не зважаючи на це, LinguaLeo залишається популярним інструментом для тих, хто бажає покращити свої навички вивчення іноземних мов.

### 1.9.2 Duolingo

Duolingo - це онлайн-платформа для вивчення мов, доступна на різних платформах, включаючи iOS, Android та веб-версію. Завдяки своїй простоті та гейміфікованому підходу, Duolingo завоював популярність серед користувачів всього світу.

Платформа пропонує короткі уроки з різних тем, використовуючи зображення, звук та текст для закріплення знань. Duolingo також використовує гейміфікацію, де користувачі можуть змагатися з друзями та отримувати віртуальні нагороди.

Однак, серед недоліків Duolingo можна відзначити обмежену глибину вивчення мови, особливо в порівнянні з більш традиційними методами. Деякі користувачі вказують на відсутність глибокого розуміння граматики та культурних аспектів мови. Крім того, при продовженому використанні може виникнути відчуття монотонності та повторюваності вправ.

Деякі функції та розширені можливості також доступні лише за певною

плату, що може викликати незадоволення серед користувачів. Незважаючи на ці недоліки, Duolingo залишається популярним серед тих, хто шукає простий та цікавий спосіб вивчення мов.

### 1.9.3 Підсумки аналізу існуючих рішень

Розглянутим програмним рішенням для вивчення англійської мови притаманні такі найбільш поширені недоліки:

- обмежена кількість безкоштовного контенту та якість перекладу;
- недостатня підтримка ОС;
- відсутність синхронізації між різними пристроями;
- відчуття монотонності та повторюваності вправ при тривалому використанні..

Результати аналізу існуючих рішень представлені в таблиці 1.1.

Таблиця 1.1 – Порівняння існуючих рішень

Критерії	LinguaLeo	Duolingo
Переваги	Вивчення мов з використанням реальних контекстів та відеоматеріалів	Гейміфікований підхід для зроблення навчання цікавим
	Можливість синхронізації даних між пристроями	Простота використання та доступність на різних платформах
	Експорт даних та можливість створювати власні категорії	Змагання з друзями та отримання віртуальних нагород
Недоліки	Обмежена кількість безкоштовного контенту та якість перекладу	Обмежена глибина вивчення мови та відсутність глибокого

		розуміння граматики та культурних аспектів
	Не завжди задовільний рівень перекладу слів та фраз	Відчуття монотонності та повторюваності вправ при тривалому використанні
	Певні функції та розширені можливості доступні лише за певною платою	Деякі функції доступні лише за певною платою

Розглянутим аналогам, описаним в таблиці, відповідають такі програмні рішення:

- LinguaLeo;
- Duolingo.

## 2 АНАЛІЗ ВИКОРИСТОВУВАНИХ ТЕХНОЛОГІЙ

### 2.1 Огляд засобів розробки

Технологічний стек - це арсенал інструментів, що використовуються при розробці проектів, включаючи мови програмування, компілятори, фреймворки, системи управління базами даних і інше. Від обраного технологічного стеку залежить продуктивність, ресурси, надійність та безпека програмного забезпечення.

Реалізація функціоналу вивчення англійської мови, як описано у вихідній задачі, відбувається через використання Телеграм-боту. Для створення таких ботів найпоширенішими є мови програмування Python, PHP та Node.js.

У даному випадку використовується Telegram Bot API фреймворк для Node.js, а саме - Telegraf. Ця технологія дозволяє швидко та ефективно створювати зручні та функціональні Телеграм-боти.

Використання Visual Studio Code для написання коду на Node.js спрощує розробку завдяки своєму інтуїтивно зрозумілому інтерфейсу та багатьом корисним функціям. Встановлення цього середовища автоматично додає необхідні інструменти для розробки програм, забезпечуючи зручний робочий процес.

Git, як система контролю версій, взята використанням для зберігання проекту на GitHub, що дозволяє команді легко співпрацювати, вносити зміни та відстежувати історію розробки. GitHub став не тільки платформою для зберігання коду, але й спільним простором для обговорення іншими членами команди, ведення питань та вирішення завдань.

Важливо пам'ятати, що вибір технологічного стеку та інструментів визначається конкретними потребами проекту і ваших здібностей. Послідовне дотримання кращих практик у розробці програмного забезпечення сприяє ефективній та стабільній роботі створеного продукту.

## 2.2 Основні поняття штучного інтелекту

Штучний інтелект - це галузь комп'ютерних наук, яка займається створенням систем та програм, які можуть виконувати завдання, які, зазвичай, потребували б інтелекту людини. ШІ використовує різноманітні методи та техніки, включаючи машинне навчання, обробку природної мови, комп'ютерне зорове сприйняття та багато інших [3].

Зазвичай, телеграм боти використовують лише декілька технологій штучного інтелекту:

- методи обробки природної мови (Natural Language Processing);
- машинного навчання (Machine Learning).

### 2.2.1 Natural Language Processing

Обробка природної мови (Natural Language Processing, NLP) - це галузь штучного інтелекту, яка займається розробкою комп'ютерних систем для аналізу, розуміння та генерації людської мови. NLP дозволяє комп'ютерам працювати з текстом, текстовими документами, інтерпретувати інтенції та виконувати завдання, пов'язані з природною мовою.

Телеграм-бот, який використовує NLP, взаємодіє з цими компонентами для аналізу та розуміння текстових повідомлень, які надходять від користувачів. Наприклад:

- розпізнавання інтенцій: бот використовує NLP для визначення інтенцій користувача, тобто для розуміння того, якого роду дії користувач хоче від бота (наприклад, переклад тексту, відповідь на питання, пошук інформації тощо);
- витягання ключової інформації: бот може використовувати NLP для визначення важливої інформації з тексту, наприклад, ідентифікація дат, місць або інших важливих даних у користувача;
- генерація відповідей: з використанням NLP, бот може генерувати текстові відповіді, які зрозумілі та природні для користувача;

- розуміння запитів користувачів: бот аналізує текстові запити користувачів для взаємодії з ними, розуміючи їхні потреби та надаючи відповіді на запити.

Використання NLP дозволяє створити інтелегентного та ефективного телеграм-бота, який може ефективно комунікувати з користувачами і надавати відповіді на їхні запити.

### 2.2.2 Machine learning

Машинне навчання (Machine Learning, ML) - це галузь штучного інтелекту, яка зосереджена на розробці алгоритмів та моделей, які дають комп'ютерам можливість вчитися з даних та виконувати завдання без явного програмування.

Телеграм-боти використовують машинне навчання [4]:

- визначення інтенцій користувачів: боти використовують методи машинного навчання для визначення інтенцій та витягання ключової інформації з текстових запитів користувачів, щоб розуміти, які дії користувач хоче здійснити;

- автоматизована відповідь на запити: моделі машинного навчання навчають ботів розуміти та генерувати текстові відповіді на запити користувачів, забезпечуючи більш інтелегентну комунікацію;

- семантичний аналіз і взаємодія з текстом: боти можуть використовувати машинне навчання для аналізу семантики тексту, зокрема для сентиментного аналізу (визначення емоційного тону) або виразних перекладів тексту;

- покращення відповідей на основі навчання на льоту: моделі можуть навчатися на льоту, вдосконалюючи відповіді бота на запити користувачів на основі зворотного зв'язку та нових даних;

- рекомендаційні системи: боти можуть використовувати моделі машинного навчання для рекомендацій, наприклад, рекомендації контенту, продуктів або послуг на основі історії взаємодії користувача.

Машинне навчання дозволяє телеграм-ботам робити взаємодію з

користувачами більш ефективно, адаптувати відповіді до потреб користувачів та використовувати інтелегентні аналітичні та обчислювальні можливості для поліпшення якості обслуговування.

### 2.2.3 Технологія GPT

Технологія GPT (Generative Pre-trained Transformer) - [8] це сучасна модель глибокого навчання, яка використовує трансформерну архітектуру. Головна її ідея полягає в тому, щоб передньо навчити модель на величезних об'ємах тексту перед тим, як вона буде використовуватися для різних завдань, таких як відповіді на запитання, генерація текстів тощо.

Моделі GPT використовують механізм уваги для ефективної обробки послідовностей, а попереднє навчання включає в себе аналіз великої кількості різноманітних текстових даних. Основна перевага полягає в тому, що після попереднього навчання модель може застосовуватися до різних завдань без необхідності повторного навчання.

Моделі GPT генерують тексти, враховуючи попередні знання, що дозволяє їм створювати природні та змістовні висловлювання. Зазвичай ці моделі мають велику кількість параметрів, що дозволяє їм засвоювати широкий спектр мовних особливостей.

Універсальність моделей GPT проявляється в їхній здатності вирішувати різноманітні завдання обробки природної мови, такі як відповіді на питання, переклад текстів, генерація контенту тощо. Оновлені версії, такі як GPT-4, покращують ці можливості та можуть включати нові функціональності, такі як обробка зображень.

Таблиця 2.1 – Версії GPT від OpenAI

Назва	Використання	Тренувальні дані	Дата випуску
GPT-1	Загальне	BookCorpus: 4,5 ГБ тексту із 7000 невиданих книг різних	11 червня 2018



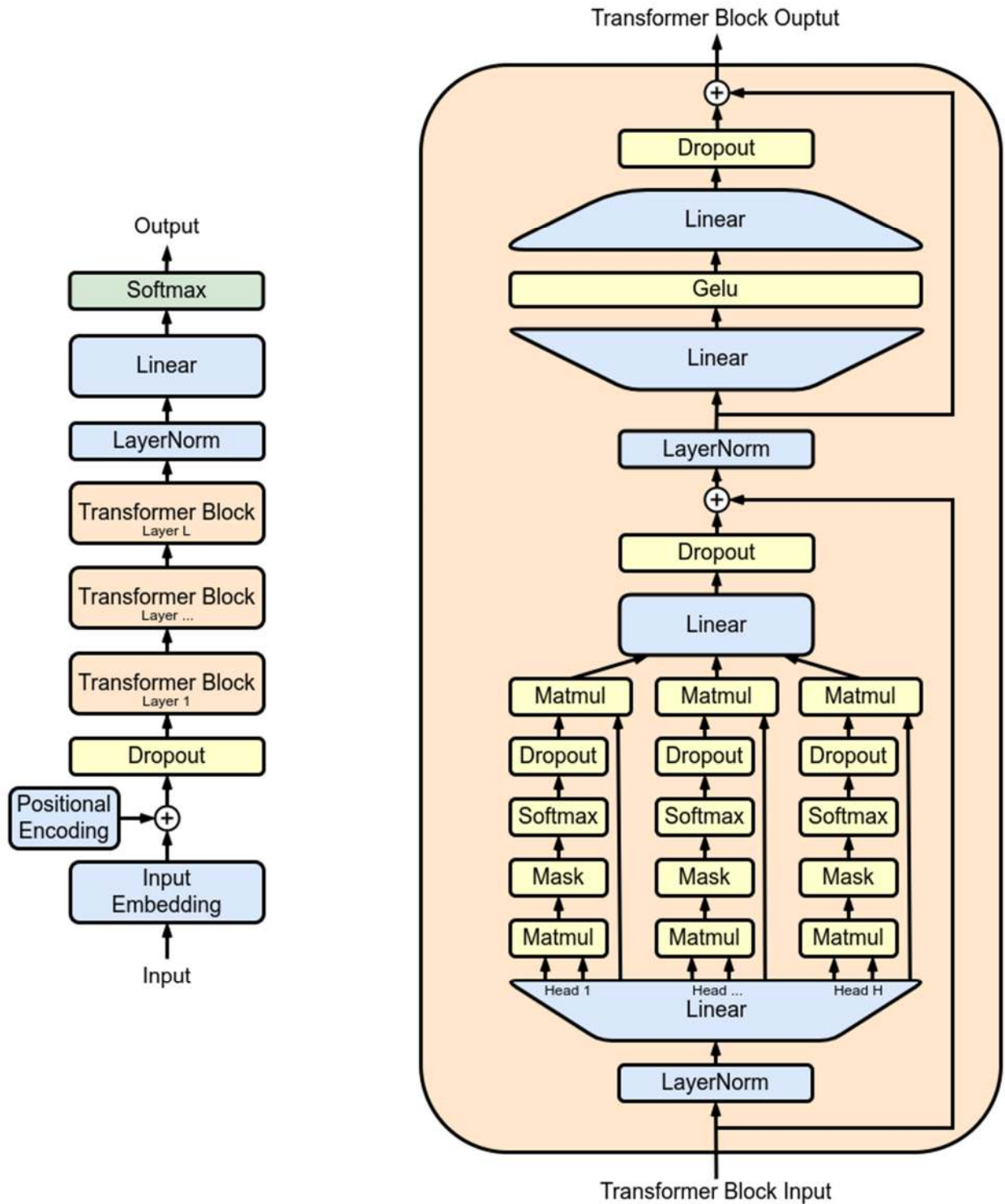


Рисунок 2.1 – Оригінальна мовна модель GPT

#### 2.2.4 ChatGPT

ChatGPT (від англ. Generative Pre-trained Transformer, що означає "генеративний передньо-навчений трансформер") - це інтерактивний чат-бот,

створений компанією OpenAI [7]. Цей бот використовує глибоке навчання і штучний інтелект для того, щоб взаємодіяти з користувачем у вигляді текстового діалогу. Однією з ключових особливостей ChatGPT є його здатність працювати з природною мовою, тобто з різними запитаннями і командами, які користувач може вводити у звичайному мовленні.

Цей бот базується на технології генеративного передньо-навченого трансформера (Generative Pre-trained Transformer, GPT). Тобто, він був попередньо навчений на великій кількості різноманітних текстових даних перед тим, як надати змогу користувачам взаємодіяти з ним у реальному часі. Ця попередня навчання включає в себе різні типи текстів, що дозволяє боту розуміти та відповідати на різноманітні питання та вирази.

ChatGPT може відповідати на запитання, роз'яснювати концепції, генерувати тексти та взагалі проводити розмову. Важливою особливістю цього бота є його універсальність і можливість працювати з багатьма мовами, включаючи українську.

OpenAI регулярно випускає нові версії своєї технології, наприклад, GPT-4, що може мати покращені функції та можливості порівняно з попередніми версіями.

### 2.3 JavaScript

JavaScript –це мова програмування з великою гнучкістю, оскільки вона підтримує різні парадигми, такі як об'єктно-орієнтований, імперативний та функціональний стилі. Вона представляє собою реалізацію специфікації ECMAScript, що забезпечує стандартизацію мови [12].

JavaScript широко використовується в якості вбудованої мови для програмного доступу до об'єктів програм. Особливо активно вона застосовується у браузерах, де вона використовується для написання сценаріїв, що надають інтерактивність веб-сторінкам.

Серед ключових архітектурних особливостей JavaScript можна виділити

динамічну та слабку типізацію, автоматичне керування пам'яттю, прототипне програмування та функції як об'єкти першого класу.

Хоча JavaScript є об'єктно-орієнтованою мовою, використання прототипування вносить свої відмінності в роботу з об'єктами, що відрізняються від традиційних клас-орієнтованих мов. Крім того, властивості функціональних мов, такі як функції як об'єкти першого класу, об'єкти як списки, каррінг, анонімні функції та замикання, додають мові додаткової гнучкості.

## 2.4 Node.js

Node.js (або просто Node) – це не просто платформа для виконання JavaScript-коду на сервері, це справжня еволюція для розробників. Розглянемо деякі ключові аспекти цієї потужної платформи, яка визначає сучасний підхід до веб-розробки та програмування загалом.

Node.js базується на відкритому вихідному коді та використовує двигун Chrome V8 для виконання JavaScript-коду. Вона виступає як ідеальна платформа для написання серверного коду для веб-застосунків, динамічних веб-сторінок та навіть програм командного рядка. Її основні переваги полягають у подієво-керованій моделі з неблокуючими операціями введення-виводу, що забезпечує велику ефективність та легкість в розробці.

До появи Node.js, JavaScript використовувався переважно в браузері. Однак завдяки цій платформі можна використовувати JavaScript не лише на клієнтській стороні, але й на сервері. Його роль узагальнюється, оскільки Node.js перетворює спеціалізовану скриптову мову JavaScript на загальнопризначену, що дозволяє писати різноманітні комп'ютерні програми.

Node.js забезпечує єдиною мовою JavaScript розробку як на стороні клієнта (Frontend), так і на сервері (Backend). Це не тільки спрощує розробку, але й робить його ідеальним інструментом для створення програм реального часу, які базуються на подіях.

Платформа користується популярністю серед різних категорій розробників

– від fronted-розробників до backend-розробників. Це відмінне рішення для написання програм, які працюють на різних операційних системах, зокрема Linux, OS X та Windows. Вона також використовується для створення API та крос-платформних програм, які синхронізують дані в реальному часі та взаємодіють з мобільними пристроями.

Не менш важливою областю використання Node.js є розробка Internet of Things (IoT), де платформа виступає ключовим інструментом управління пристроями та створення серверів, які можуть обробляти велику кількість запитів одночасно.

Серед основних факторів популярності Node.js виділяються його висока швидкість виконання, універсальність та гнучкість, наявність великої кількості модулів та бібліотек, а також використання двигуна Chrome V8. Всі ці фактори роблять Node.js не лише потужною платформою для розробки веб-застосунків, а й ключовим інструментом для інновацій та постійного росту в області програмування.

## 2.5 Управління залежностями за допомогою NPM

NPM (Node Package Manager) – це не лише невід'ємна частина платформи NodeJS, але і важливий фактор, що забезпечує популярність NodeJS в розробницькому середовищі. Цей потужний менеджер пакетів формує обширну та динамічну екосистему, що визначає величезний успіх платформи. NPM виступає в ролі вірного помічника, спрощуючи вирішення різноманітних завдань завдяки можливості встановлення різноманітних пакетів, розроблених іншими учасниками спільноти [11].

Цей пакетний менеджер несе на собі низку важливих завдань, які роблять його незамінним інструментом у процесі розробки. Наприклад, він дозволяє запускати пакети безпосередньо, без необхідності завантаження, що зберігає час і прискорює розробку. Крім того, NPM обмежує доступ до коду для конкретних розробників, надаючи можливість адаптувати пакетний код під конкретні

потреби. Зручно вести управління декількома версіями коду та його залежностей, а також завантажувати автономні інструменти, готові до використання в будь-який момент.

Однією з ключових функцій NPM є можливість знаходження інших розробників, що працюють над аналогічними проблемами та проектами. Це створює сприятливий ґрунт для обміну досвідом та знаходження оптимальних рішень. Крім того, NPM надає користувачам можливість легко ділитися своїм кодом з іншими розробниками в будь-якій точці світу, сприяючи колективному розвитку та співпраці в програмній галузі.

## 2.6 Telegram Bot API

Telegram Bot API представляє собою комплекс інтерфейсів програмування застосунків (API), створених Telegram, які дають змогу взаємодіяти з сервісами Telegram Bot та інтегрувати їх з іншими платформами. Цей набір інтерфейсів відкриває безмежні можливості для розробників створювати сторонні застосунки та покращувати чи розширювати функціональність ботів.

Використання Telegram Bot API дозволяє легко створювати програми, які взаємодіють з Telegram за допомогою власного інтерфейсу, виконуючи код на локальному чи віддаленому сервері. Користувачі мають змогу взаємодіяти з ботами, відправляючи їм повідомлення, команди та викликаючи вбудовані запити.

Заснований на принципі постійного надсилання запитів на сервер та регулярного отримання оновлень, Telegram Bot API функціонує через два основних механізми. По-перше, використання вебхуків, де сервер здійснює зворотний дзвінок на зазначений URL. По-друге, можливість «закидування» запитамі безпосередньо до Telegram, отримуючи постійні відповіді. Важливо зазначити, що веб-хуки в Telegram працюють виключно через HTTPS (див. рисунок 2.2). Ці два методи надають розробникам гнучкість та можливість вибору оптимального підходу залежно від конкретних потреб та умов взаємодії.

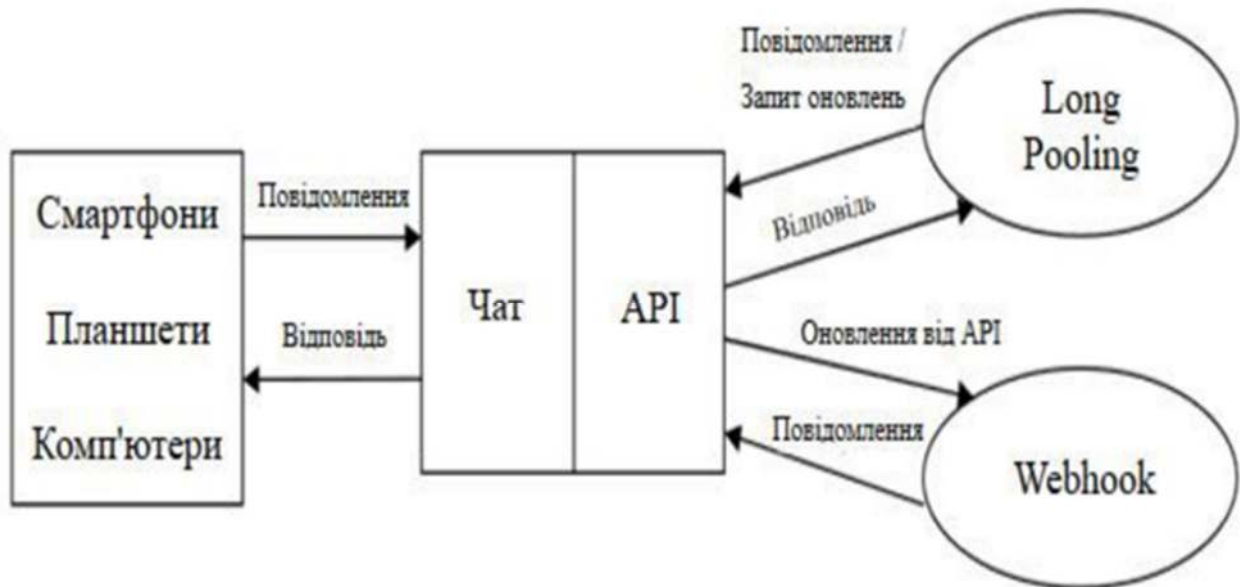


Рисунок 2.2 – Принцип роботи чат-бота на платформі Telegram

## 2.7 Telegraf

Telegraf – це потужний фреймворк, який служить оболонкою для ефективної взаємодії з Telegram Bot API, а його використання спрощує та мінімізує написання коду, роблячи розробку ботів більш зручною та продуктивною. Цей фреймворк об'єднує всі необхідні методи API у класі Telegraf, що робить його важливим інструментом для розробників, які створюють Telegram-ботів.

Основні функції, які входять до арсеналу фреймворку Telegraf, вражають своєю різноманітністю та потужністю. Перш за все, він повністю підтримує Telegram Bot API версії 5.0, забезпечуючи розробникам доступ до найновіших можливостей та функцій платформи.

Однією з інноваційних функцій є вбудована підтримка платіжної платформи Telegram, що дозволяє ботам легко інтегрувати функції оплати та електронної комерції в свої сервіси. Крім того, фреймворк підтримує ігри HTML5, розширюючи можливості розробників у створенні інтерактивного

контенту для користувачів.

Зазначимо також, що Telegraf пропонує Inline режим, який дозволяє ботам взаємодіяти з користувачами прямо в чатах, спрощуючи введення команд та отримання відповідей. Ще однією важливою характеристикою є сумісність з різними платформами, такими як Firebase, Glitch, Heroku та AWS, що надає розробникам широкі можливості розгортання та використання фреймворку.

Telegraf демонструє вражаючий рівень гнучкості, підтримуючи різні протоколи для веб-хуків, такі як http, https, fastify, Connect.js та express.js. Це робить його відмінним вибором для розробників з різними потребами та уподобаннями у використанні веб-серверів.

Крім того, Telegraf підтримує TypeScript типізацію, що додає додатковий рівень структурованості та надійності до процесу розробки, сприяючи створенню більш стійких та масштабованих ботів. Таким чином, фреймворк Telegraf виступає важливим інструментом для розробників, які бажають створювати потужні та інноваційні Telegram-боти з ефективним та чистим кодом.

## 2.8 Система контролю версій та стратегія розгалуження

Під час процесу розробки невід'ємно використовувати засоби контролю версій, і наразі одним з найбільш поширених і ефективних є Git. Цей інструмент надає безліч можливостей для ведення роботи над проектом та співпраці з іншими розробниками.

Git використовується не лише для збереження історії змін коду проекту, але й для організації спільної роботи. Одним із ключових аспектів використання Git є його можливість створення різних гілок, що дозволяє поділити розробку на окремі етапи та завдання [13].

Існує кілька основних моделей розгалуження в Git, які розробники можуть використовувати в залежності від особливостей проекту та вимог співпраці [5]. Однією з таких моделей є Central Workflow, де центральний репозиторій

використовується як єдина точка входу для всіх змін у проекті. Гілка розробки за умовчанням називається головною гілкою (main), і до неї надсилаються всі коміти змін.

Ще однією моделлю є Developer Branch Workflow, де розробники створюють власні гілки для роботи над окремими функціональностями. Featured Branch Workflow передбачає наявність головної гілки та власних гілок для розробки окремих функціональностей, які потім інтегруються в головну гілку.

Issue Branch Workflow передбачає створення розробниками гілок для вирішення конкретних завдань у проектному трекері, створюючи окрему гілку для кожного завдання. Patch Workflow включає додавання змін до репозиторію разом з патчем.

У нашому проекті ми вибрали модель розгалуження Central Workflow, оскільки вона ідеально підходить для самостійних проектів, де немає потреби у створенні багатьох гілок окрім головної. Ця модель передбачає здійснення комітів після кожного етапу виконаної роботи, що надає можливість легко повертатися до попередніх версій і визначати прогрес проекту на кожному етапі.



Рисунок 2.3 – Модель розгалуження Central Workflow

## 2.9 Visual Studio Code

Visual Studio Code (VS Code) – це не просто редактор вихідного коду; це потужний і водночас легкий інструмент, доступний для використання на платформах Windows, MacOS та Linux. Забезпечуючи підтримку для таких мов програмування, як JavaScript, TypeScript і Node.js, а також розширеної екосистеми додаткових модулів для інших мов (таких як C++, C#, Java, Python, PHP та Go), VS Code визначається своєю універсальністю та гнучкістю.

Інтеграція з інструментами збірки та створення сценаріїв у VS Code сприяє

швидкому та зручному щоденному робочому процесу розробника. Підтримка Git вбудована в редактор, що дозволяє вам здійснювати контроль версій безпосередньо в робочому середовищі.

Однією з вагомих переваг цього інструменту є можливість налаштування будь-якої функції відповідно до ваших особистих потреб у автоматизації. Крім того, VS Code підтримує можливість встановлення необмеженої кількості сторонніх розширень, розширюючи його базовий функціонал.

Редактор також включає багатофункціональну вбудовану підтримку розробки на Node.js з використанням JavaScript і TypeScript, використовуючи ті ж базові технології, що й Visual Studio. Зокрема, VS Code володіє інструментами для розробки веб-технологій, таких як JSX/React, HTML, CSS, SCSS, Less та JSON.

Завдяки можливостям розширення, VS Code стає ще більш потужним інструментом. Розширення розширюють функціонал і додають нові можливості, забезпечуючи більш гнучкий та розширений досвід розробки. Після випуску цих функцій, розширення VS Code можна використовувати під час роботи з порталами Power Apps, що робить його важливим інструментом для розробників, які працюють в цьому середовищі.

Однією з головних особливостей VS Code є можливість використовувати сніпети. Сніпет - це короткий фрагмент коду чи тексту, який використовується для автоматизації і прискорення написання коду. Використання сніпетів є потужним інструментом для підвищення продуктивності розробників, допомагаючи автоматизувати рутинні операції та прискорюючи процес розробки.

## 2.10 Docker

Docker – це відкрите програмне забезпечення, яке надає засоби для автоматизації розгортання та управління контейнеризованими додатками. Контейнери у Docker включають в себе програмний код, залежності,

конфігураційні файли та інші елементи, необхідні для виконання додатка. Ця технологія ізолює додатки від операційної системи та середовища виконання, забезпечуючи консистентність та незалежність від середовища [10].

Основними перевагами Docker є:

- легкість розгортання: контейнери можна легко переносити між різними середовищами, що полегшує розгортання додатків на різних платформах;
- ізоляція: контейнери забезпечують відокремлене середовище виконання для додатків, що дозволяє уникнути конфліктів між різними програмами та їхніми залежностями;
- ефективність ресурсів: Docker дозволяє ефективно використовувати ресурси системи, оскільки кожен контейнер має своє власне ізольоване середовище, але використовує спільний ядро операційної системи;
- швидкість розгортання: контейнери можуть бути створені та запуснені за кілька секунд, що дозволяє швидко реагувати на зміни та вимоги в розробці.

Docker став популярним інструментом в галузі розробки програмного забезпечення та інфраструктури завдяки своїй простоті використання та потужним можливостям контейнеризації.

Одним із практичних використань Docker є запуск та управління віртуальними контейнерами, які містять Telegram-боти. Це надає унікальну можливість легкої ізоляції та розгортання ботів у стандартизованому середовищі. Процес запуску Telegram-ботів у Docker-контейнерах сприяє збереженню їхньої незалежності, підвищуючи ефективність та еластичність програмних рішень.

## 2.11 Axios

Axios – це бібліотека для виконання HTTP-запитів в середовищі JavaScript, яка дозволяє здійснювати асинхронні запити до сервера та обробляти їхні відповіді. Axios є однією з популярних бібліотек для роботи з HTTP-запитами і широко використовується як у браузерях, так і в середовищі Node.js [15].

Основні особливості Axios включають:

- простота використання: Axios надає простий та зрозумілий інтерфейс для виконання різних типів HTTP-запитів, таких як GET, POST, PUT, DELETE тощо;
- асинхронність: Axios використовує асинхронний підхід для виконання запитів, що дозволяє здійснювати запити без блокування виконання інших операцій;
- підтримка обіцянь: Axios повертає обіцянки (Promises), що дозволяє зручно використовувати синтаксис обіцянь або async/await для обробки відповідей;
- Можливість встановлення перехоплювачів (interceptors): Axios дозволяє вам встановлювати перехоплювачі для обробки та зміни запитів та відповідей до їхнього відправлення або після отримання;
- Автоматичне перетворення даних: Axios автоматично перетворює дані в формат JSON, щоб спростити роботу з даними, які передаються чи отримуються з сервера.

## 3 ПРОГРАМНА РЕАЛІЗАЦІЯ

### 3.1 Архітектура проекту

Робота телеграм бота ґрунтується на моделі взаємодії клієнт-сервер, що забезпечує розділення функціоналу та навантаження між клієнтською та серверною частинами.

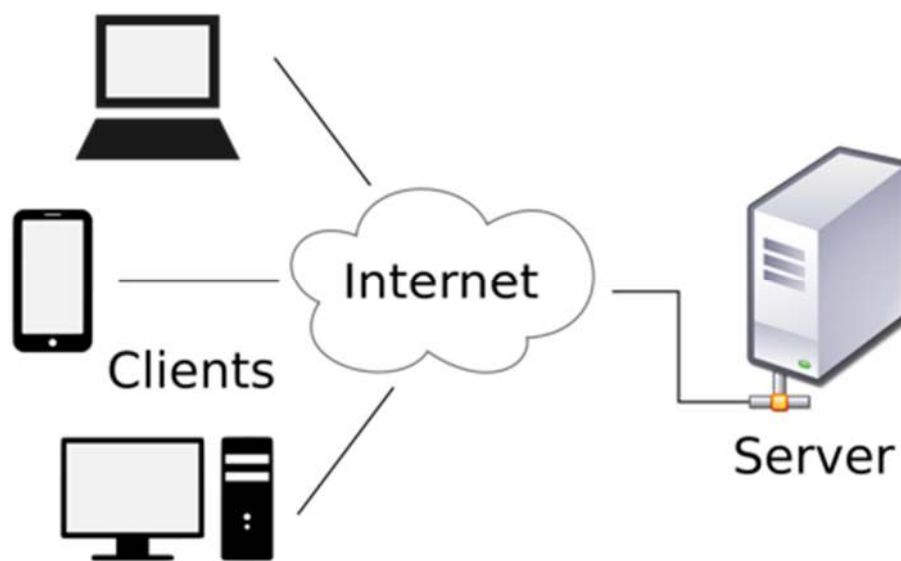


Рисунок 3.1 – Клієнт-серверна архітектура загального програмного засобу

У рамках даного проекту серверна частина була реалізована за допомогою технології Node.js, в той час як відповідальність за клієнтську частину лежить на платформі Telegram. Бекенд частина була створена у форматі REST API та визначається рядом ключових параметрів:

а) Client-server (клієнт-серверна архітектура): цей підхід передбачає відокремлення інтерфейсу від зберігання даних. Такий підхід поліпшує переносимість інтерфейсу між різними платформами та забезпечує більшу гнучкість системи.

б) Stateless (відсутність збереження стану): у цій архітектурі кожен запит містить усю необхідну інформацію для розуміння та обробки запиту, і не використовує будь-якого збереженого стану на сервері. Замість цього, інформація про стан зберігається на стороні клієнта, що дозволяє підтримувати систему без стану та сприяє кращій масштабованості та швидкості відповіді.

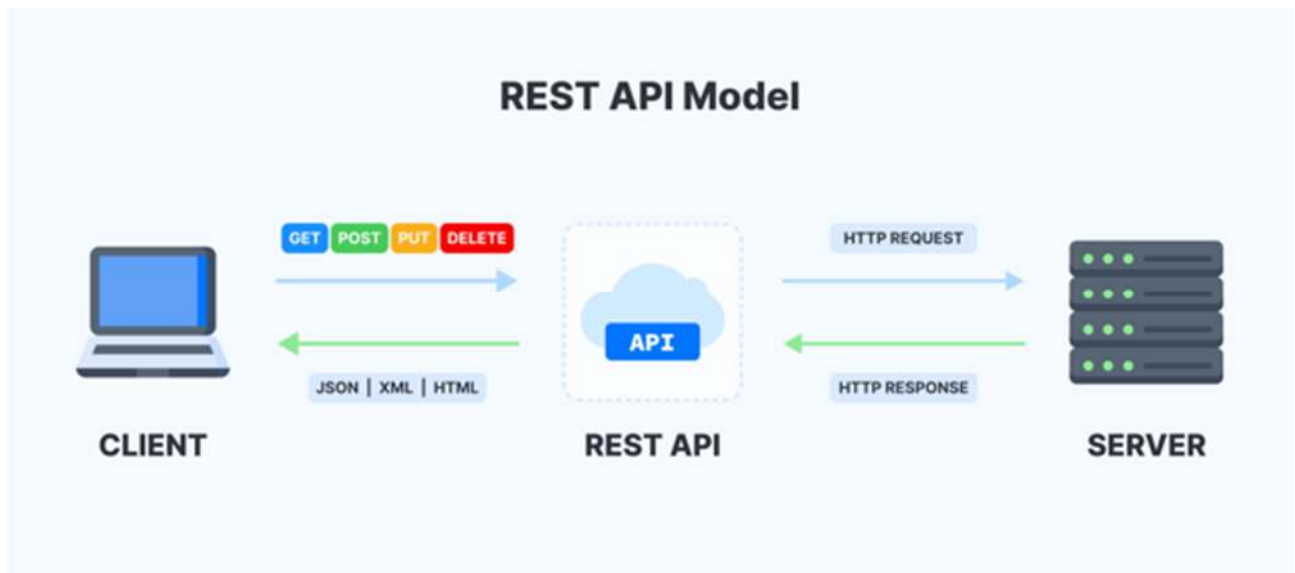


Рисунок 3.2 – Протокол REST API

У архітектурі REST сервер не зберігає жодної інформації про стан операції, залишаючи усі дані сесій на стороні клієнта. Це вказує на те, що два різні запити від одного клієнта не повинні взаємодіяти або впливати один на одного. Таким чином, клієнт повинен передавати всю необхідну інформацію для виконання дії відразу. Цей підхід сприяє ефективній економії часу та ресурсів і спрощує масштабування сервера.

REST, або Representational State Transfer, є архітектурою, що визначає принципи побудови розподілених гіпермедіа систем, зокрема, світової павутини (World Wide Web). Вона використовує універсальні методи обробки та передачі стану ресурсів по протоколу HTTP.

Архітектура REST не є прив'язаною до конкретних технологій та протоколів, але в сучасному Інтернеті побудова RESTful API майже завжди

включає використання HTTP і популярних форматів представлення ресурсів, таких як JSON або, менш популярний сьогодні, XML.

З погляду RESTful-сервісу, операція вважається ідемпотентною, якщо клієнти можуть робити однаковий виклик неодноразово з однаковим результатом на сервері. Це означає, що створення великої кількості ідентичних запитів викликає той самий ефект, що й один запит. Важливо зауважити, що хоча ідемпотентні операції дають однаковий результат на сервері, відповідь сама по собі може не бути однаковою, оскільки стан ресурсу може змінюватися між запитами.

Персоналізуючи архітектуру під цей телеграм-бот, отримуємо таку схему (рисунок 3.3)

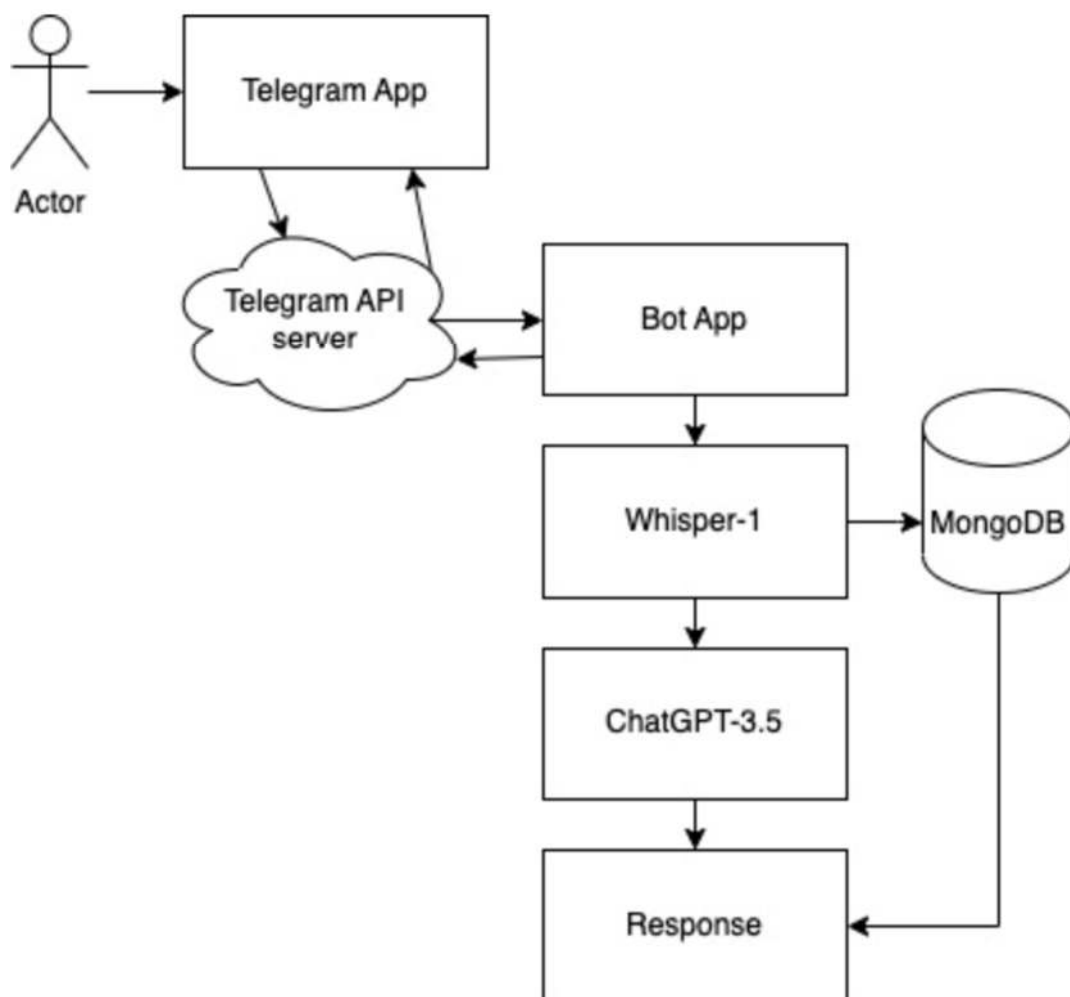


Рисунок 3.3 – Архітектура даного Телеграм-боту

Коли користувач відправляє повідомлення чат-боту, бот відправляє запит на Telegram API Server. Для цього бот використовує long polling (постіне опитування, чи не сталася якась нова подія на сервері). Тобто бот постійно запитує через HTTP у Telegram API Server оновлення і коли з'являється нова подія, ми отримуємо її.

Щоб надати можливість кільком користувачам використовувати бота без перешкод між собою, необхідно створювати унікальні сесії для кожного із них. Дані щодо цих сесій зберігаються у базі даних.

У випадках, коли користувачу потрібно виконати якусь дію за допомогою відправки декількох повідомлень, наприклад діалог с ботом, отримання інформації щодо правильності тексту, де потрібно вказати речення, слова, голосові повідомлення, ми використовуємо сцени. Для даного чат-боту було реалізовано декілька сцен (таблиця 3.1), увійти в кожную з цих сцен можна за допомогою кнопок у головному меню (рисунок 3.4).

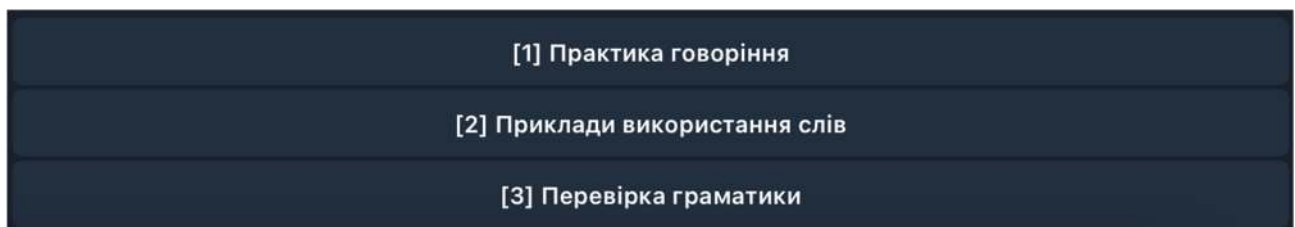


Рисунок 3.4 – Кнопки головного меню

Таблиця 3.1 – Реалізовані сцени для чат-боту

Сцена	Опис
checkGrammarScene	Сцена для перевірки правильності граматики у тексті. Складається з двох рівнів. На першому рівні потрібно ввести речення, на останньому рівні реалізовано метод перевірки граматики у тексті.
dialogueScene	Сцена для діалогу з ботом. Складається з чотирьох рівнів. На першому рівні потрібно обрати рівень знань

	англійської мови, на другому обрати тему діалогу зі списку, на третьому випадковим образом з бази даних витягнеться питання, на основі обраного рівня англійської та теми, на останньому буде трансформовано голосове повідомлення у текст, далі буде отримано відповідь від OpenAI з порадами, виправленням помилок та буде виведена інформація про використані слова та їх синоніми.
continueDialogueScene	Сцена для продовження діалогу з ботом на вже вибрану тему та з тим самим рівнем англійської. Складається з двох рівнів. Напершому рівні з бази даних витягнеться питання, на основі обраного рівня англійської мови та теми, на останньому буде трансформовано голосове повідомлення у текст, далі буде отримано відповідь від OpenAI з порадами, виправленням помилок та буде виведена інформація про використані слова та їх синоніми.
wordExamplesScene	Сцена для отримання інформації стосовно використання слів та словосполучень. Складається з двох рівнів. На першому рівні потрібно ввести слово або словосполучення, на останньому рівні реалізовано метод отримання варіантів використання даного слова.

Було реалізовано команду «/menu».

При введенні команди «/menu» визивається метод `showMainMenu()`, який показує кнопки головного меню (лістинг 3.1).

Лістинг 3.1 – Метод показу кнопок головного меню

```
async function showMainMenu(ctx) {
  await ctx.reply(
```

```

"Виберіть потрібну опцію у меню:",
Markup.keyboard([
  ["[1] Практика говоріння"],
  ["[2] Приклади використання слів"],
  ["[3] Перевірка граматики"],
]).oneTime().resize().extra()
)
}

```

Файлова структура проекту виглядає наступним чином (рисунок 3.5).

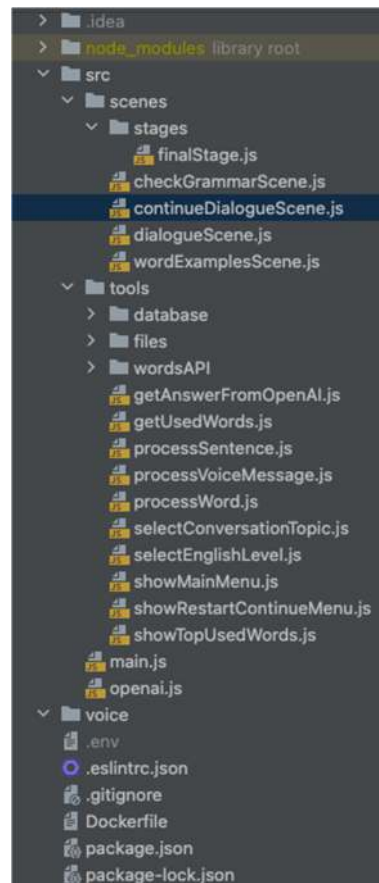


Рисунок 3.5 – Файлова структура проекту

## 3.2 Використання OpenAI моделей

В даному телеграм-боті реалізовано можливість отримувати пояснення щодо граматичних питань, також питань пов'язаних з використанням іноземних слів у правильних ситуаціях. Для цього використовується GPT-3.5 Turbo модель.

Ще одна реалізована можливість у боті - це спілкування з моделлю GPT-3.5 Turbo за допомогою голосу. Для переведення аудіо файлу у текст використовується Whisper-1 модель.

### 3.2.1 GPT-3.5 Turbo

GPT-3.5 Turbo –це сімейство моделей, яке підходить для використання, не пов'язаного з чат-ботами. Модель може обробляти 4 тисячі токенів одночасно, що вдвічі більше, ніж можливості попередніх версій. OpenAI заявила, що перші тестувальники використовували на 90% коротші підказки після підготовки GPT-3.5 до точно налаштованих інструкцій [9].

Для взаємодії з моделями OpenAI використовується OpenAI API. Щоб отримати відповідь від GPT-3.5 Turbo моделі використовуємо наступний метод (лістинг 3.2), який буде повертати відповідь моделі.

#### Лістинг 3.2 – Отримання відповіді моделі

```

async function chat(messages) {
  const chatCompletion = await openai.chat.completions.create({
    model: "gpt-3.5-turbo",
    messages: [{ role: "user", content: `${messages}` }],
  })

  return chatCompletion.choices[0].message
}

```

Перед початком взаємодії потрібно провести попередні налаштування, описавши що ми очікуємо у відповідях від моделі, формат, правила та інше (лістинг 3.3)

#### Лістинг 3.3 – Попередні налаштування

```

async function getAnswerFromOpenAI(messages) {
  const answerRequirements =
    "Imagine that you are a real person. " +
    "Please, don't tell 'as an AI' and something about AI. " +
    "Comment the idea in the next message. Answer length is 2-
3 sentences."
  const answerResponse = await
chat(answerRequirements.concat(messages))

  const suggestionRequirements =
    "Check grammar, give possible suggestions for improvements
in terms of English language." +
    "How we can rephrase the same text but make it more
clear." +
    "Explain my every mistake." +
    "Maximum answer length is 5 sentences." +
    "The following sentences:"
  const userAnswer = messages.slice(1)
  const suggestionResponse = await
chat(suggestionRequirements.concat(userAnswer))
  return { answerResponse, suggestionResponse}
}

```

Таким чином ми задаємо попередній контекст для діалогу, який модель буде враховувати при кожній відповіді.

### 3.2.2 Whisper-1

Whisper – це система автоматичного визнання мовлення (ASR), навчена на 680 000 годинах багатомовних та багатозадачних даних, зібраних з мережі. Використання такого великого та різноманітного набору даних призводить до покращеної стійкості до акцентів, фонового шуму та технічної мови. Крім того, це дозволяє транскрипцію на кількох мовах, а також переклад з цих мов на

англійську [14].

Whisper працює на архітектурі нейромережі Transformer, яка включає кодер і декодер. Звук розбивається на 30-секундні відрізки, які перетворюються у лог-Мел-спектрограму і передаються кодеру. Потім дані направляються до декодера, який передбачає текстове представлення.

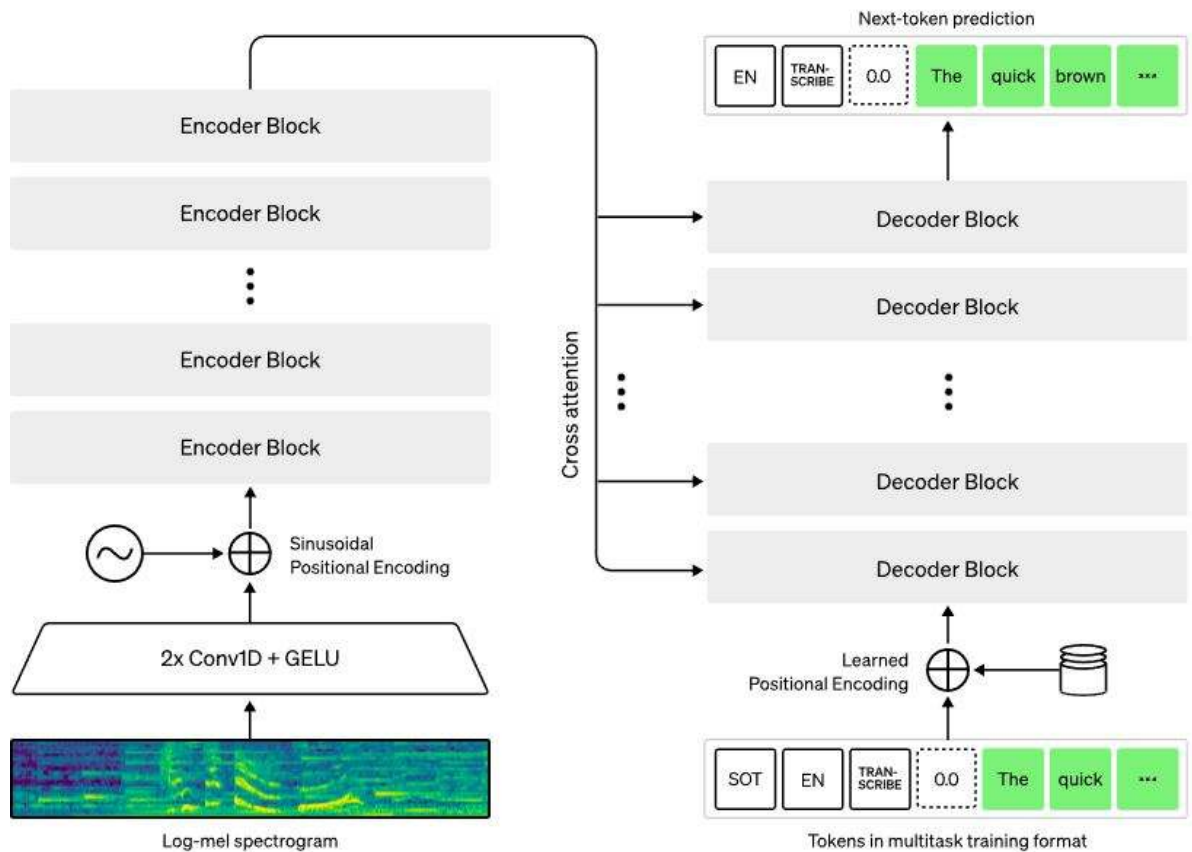


Рисунок 3.6 – Архітектура моделі Whisper-1

В даному випадку ми використовуємо цю модель для переведення голосних повідомлень від користувача у текст (лістинг 3.4).

Лістинг 3.4 – Трансформування голосового повідомлення у текст

```
async function transcription(filepath) {
    const response = await
openai.audio.transcriptions.create({
```

```
        model: "whisper-1",  
        file: createReadStream(filepath),  
    })  
  
    return response.text  
}
```

### 3.2 Реєстрація чат-бота для Telegram Bot API

Спочатку для розробки бота потрібно зробити його реєстрацію у спеціального чат-бота «BotFather».

BotFather – це асистент, який допомагає користувачам керувати ботами (рисунок 3.7).

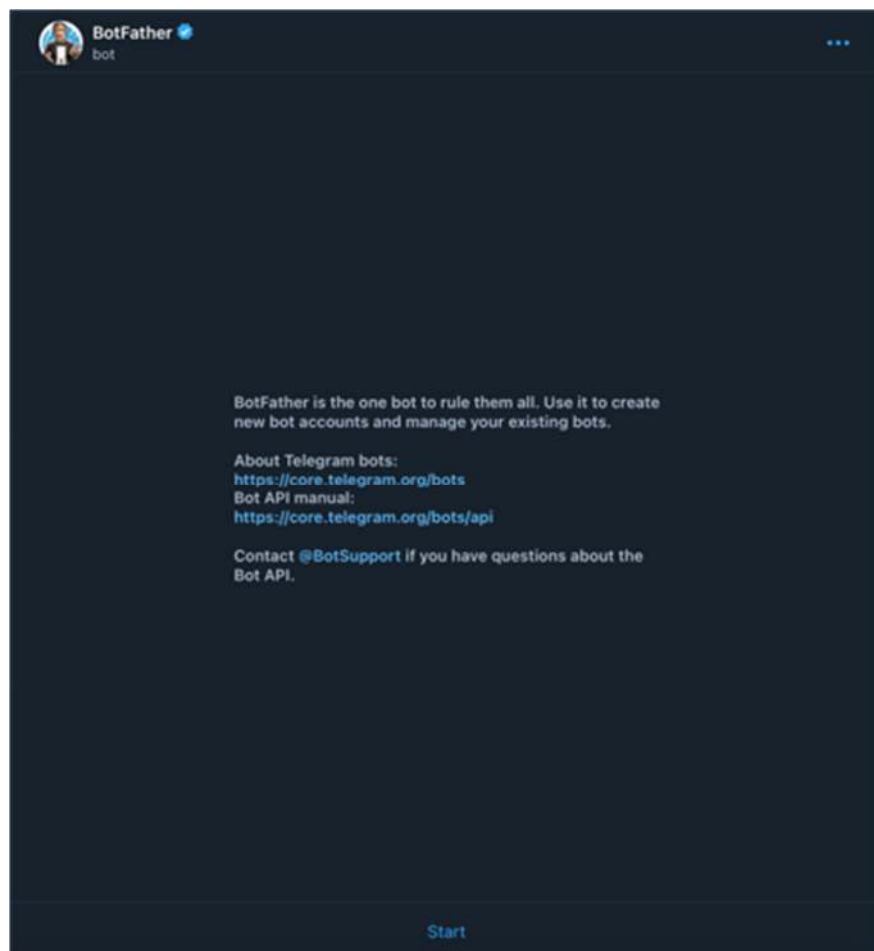


Рисунок 3.7 – Телеграм-бот «BotFather»

Спочатку треба запустити «BotFather» командою «/start». Далі ми бачимо всі доступні команди для створення та керування власними чат-ботами (рисунок 3.8).

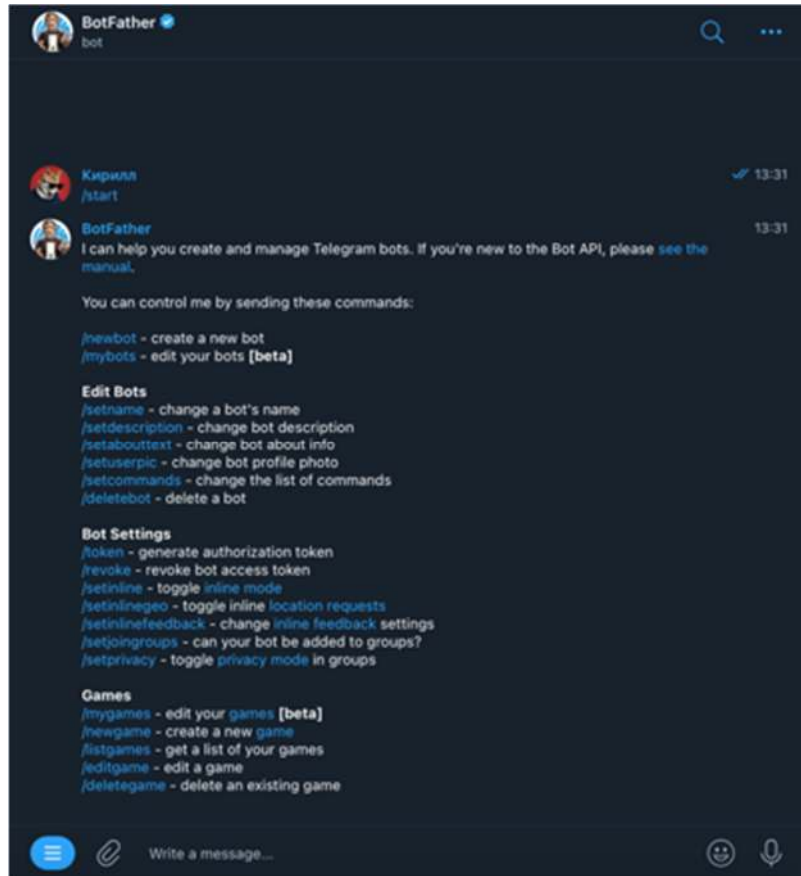


Рисунок 3.8 – Команди Телеграм-бота «BotFather»

Для розпочатку реєстрації, скористайтеся командою `/newbot`. Після цього вам потрібно вказати назву та юзернейм для вашого бота. Важливою умовою є те, що юзернейм має закінчуватися на `"Bot"` або `"_bot"`.

При правильному введенні даних, "BotFather" видасть унікальний токен для вашого бота (спеціальний символний набір для доступу до HTTP API Telegram Bot) та URL-адресу для вашого чат-бота (рисунок 3.9).

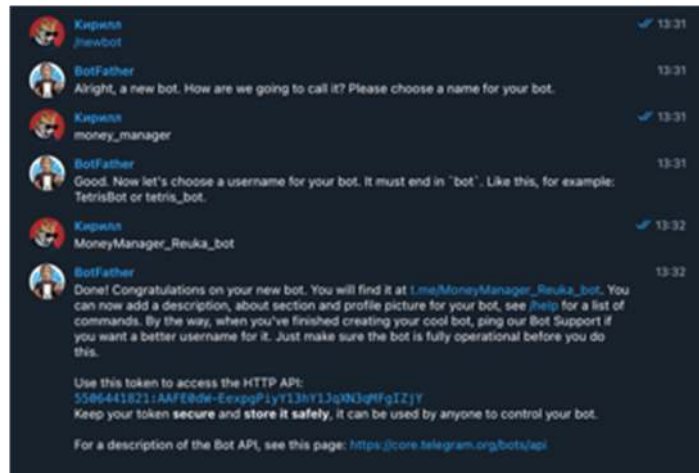


Рисунок 3.9 – Отримання токена та URL адреси чат-бота

Також ми маємо можливість встановити додаткові параметри, наприклад іконку, опис, інформацію про чат-бот та інші (Таблиця 3.2).

Таблиця 3.2 – Доступні основні команди для власного чат-бота

Команда	Опис
/setname	Змінити вказане ім'я бота
/setdescription	Текст, який буде зображено при першому при першому відкритті
/setabouttext	Текст, який буде вказано у поле «Про чат-бот»
/setuserpic	Встановлює обране зображення
/setcommands	Створює список доступних команд для використання
/deletebot	Видаляє обраного чат-бота

Скористуємося деякими з цих команд, щоб налаштувати нашого чат-бота.

Використаємо команду «/setdescription» щоб вказати текст, який буде зображено при першому відкритті бота (рисунок 3.10).

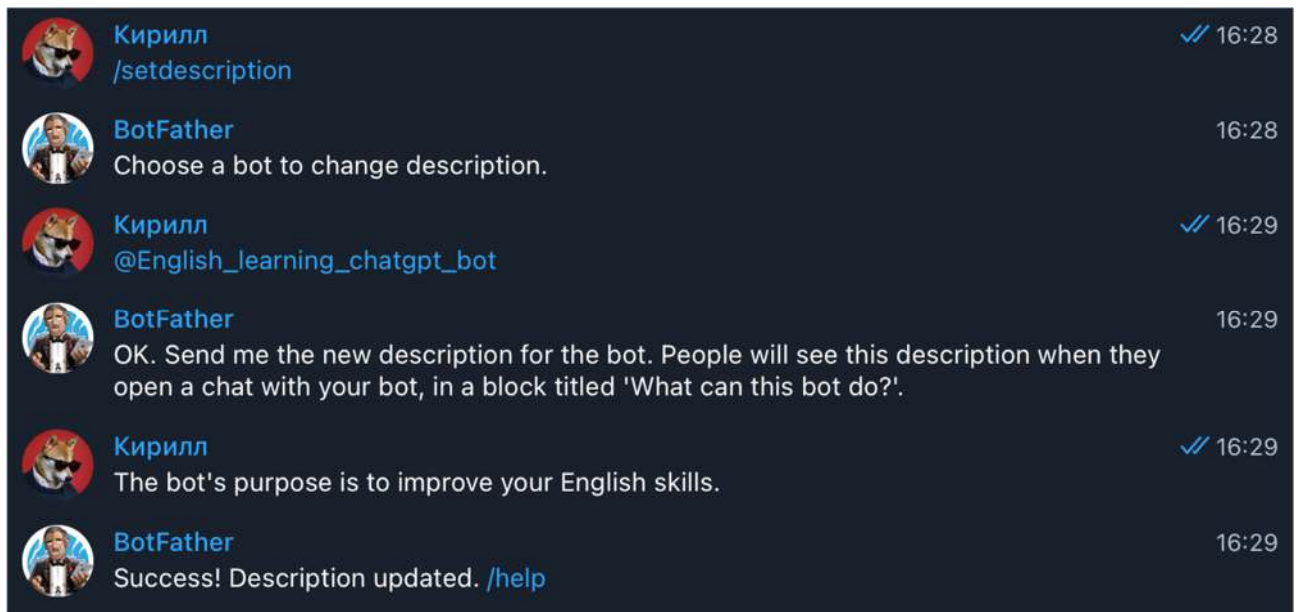


Рисунок 3.10 – Використання команди «/setdescription»

Використаємо команду «/setuserpic» щоб вказати картинку бота (рисунок 3.11).



Рисунок 3.11 – Використання команди «/setuserpic»

Використаємо команду «/setcommands» щоб вказати команди, які будуть доступні у боті (рисунок 3.12).

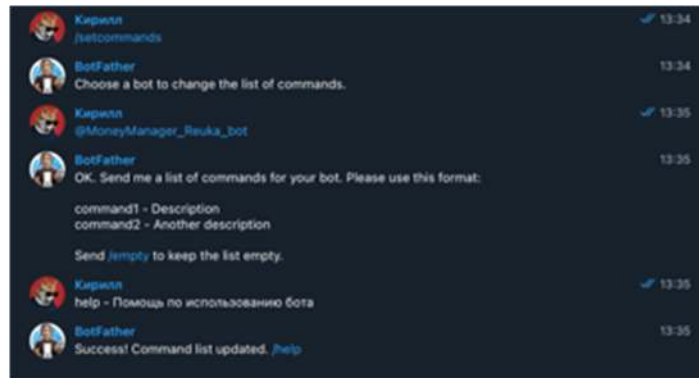


Рисунок 3.12 – Використання команди «/setcommands»

Використаємо команду «/setabouttext» щоб вказати текст, який буде зображено при першому відкритті бота (рисунок 3.13).

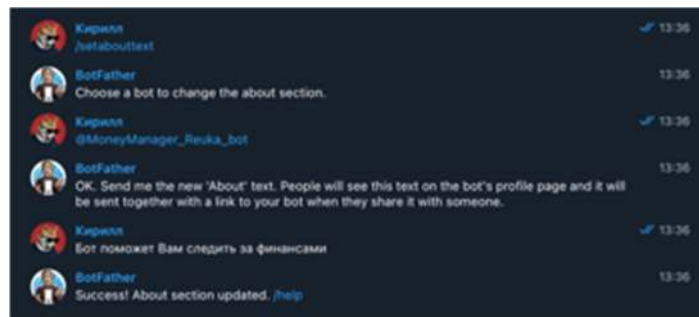


Рисунок 3.13 – Використання команди «/setabouttext»

Окрім основних команд для зміни параметрів чат-боту існують команди, які дозволяють виводити незмінні параметри та привласнювати значення, які вказані в таблиці 3.3.

Таблиця 3.3 – Доступні основні команди для власного чат-бота

Команда	Опис
1	2
/token	Виводить токен боту
/revoke	Скидає токен боту

## Продовження таблиці 3.3

1	2
/setinline	Вмикає або вимикає inline режим
/setinlinegeo	Вмикає або вимикає можливість передачі локації з інших чатів
/setjoingroup	Вмикає або вимикає можливість додавання боту до інших чатів
/setprivacy	Вмикає або вимикає режим конфіденційності

## 3.3 Розміщення чат-боту на платформі Heroku

Heroku – це платформа, яка надає розробникам готове середовище для розгортання веб-проектів. Це, здебільшого, спеціалізований хостинг, на якому вже налаштовані всі необхідні послуги, такі як мови програмування, бази даних і т.д. Користувачеві залишається тільки використовувати команду `git push` для відправлення коду свого додатка на платформу, після чого він стає доступним в Інтернеті за доменом 3-го рівня (хоча можна також додати власний домен) [6].

Крім того, у Heroku є чудова можливість - наявність безкоштовного тарифного плану, який, навіть з обмеженнями, відмінно підходить для тестування проектів.

Платформа підтримує різні мови та середовища, такі як Python, PHP, NodeJS, Java, Go та інші. І все це без будь-якої установки - просто пишете код і деплоїте його на сервіс.

Отже, перш ніж розпочати, вам слід зареєструватися на Heroku (рисунок 3.14).

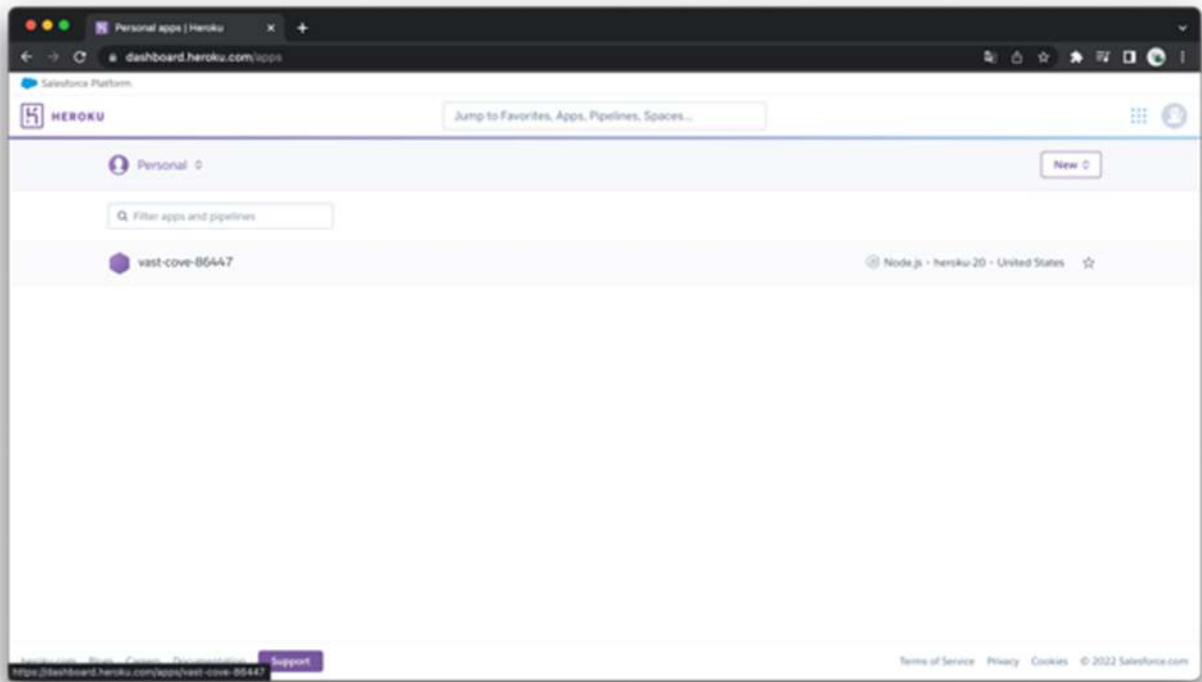


Рисунок 3.14 – Головна сторінка платформи Нероку

Далі, необхідно встановити інструмент командного рядка Нероку, відомий як Нероку CLI. За допомогою цього інструменту ми будемо розгортати програму на платформі.

Після успішної реєстрації на платформі та встановлення необхідного інструментарію, ви можете перейти на головну сторінку і створити нову програму. Інтерфейс є інтуїтивно зрозумілим, навіть якщо він англійською мовою

Створення програми виконується на головній сторінці (рисунок 3.15). При цьому відкриється діалогове вікно, де вам потрібно вказати ім'я програми та обрати регіон його розташування. Після цього, натиснувши кнопку "Create App", ви автоматично перейдете на вкладку "Deploy" (розгортання), де надається інструкція щодо розміщення програми на платформі.

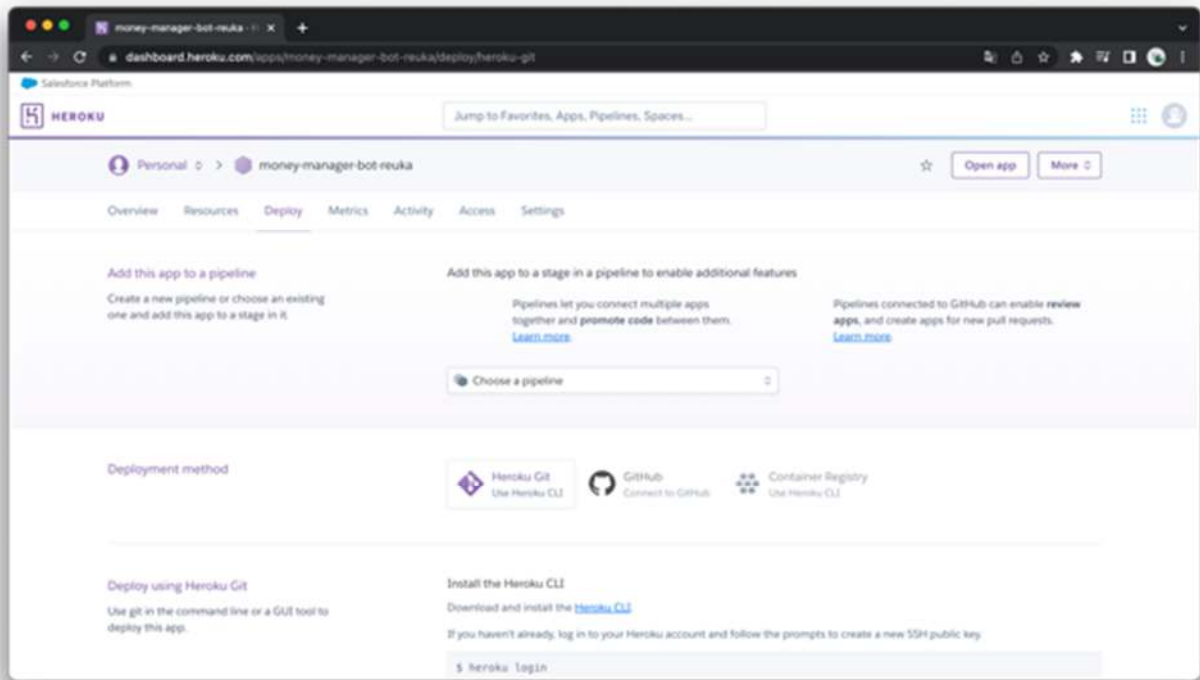


Рисунок 3.15 – Сторінка розгортання чат-боту на платформі Нероку

Існує декілька варіантів як ми можемо розмістити чат-бот на платформі Нероку. В даному випадку я буду використовувати GitHub. Для цього потрібно авторизуватись через GitHub та вказати назву репозиторію (рисунок 3.16).

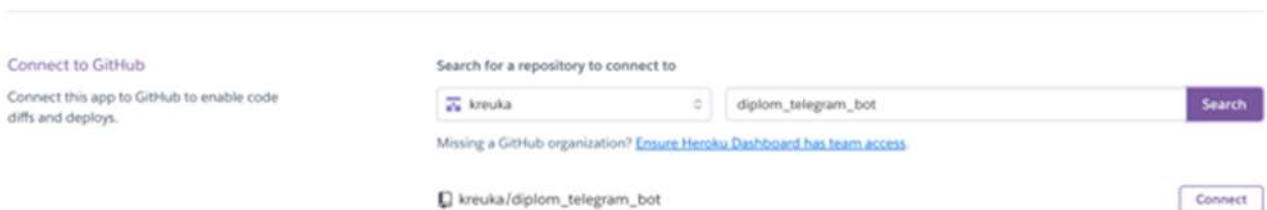


Рисунок 3.16 – Під'єднання репозиторію до платформи Нероку

Після цього нам залишається вибрати гілку, яка буде розгортатися, та налаштувати параметри автоматичного розгортання при новому коміті, та натиснути кнопку Deploy (рисунок 3.17).

Після цього нашим чат-ботом можна буде користуватися у будь-який час.

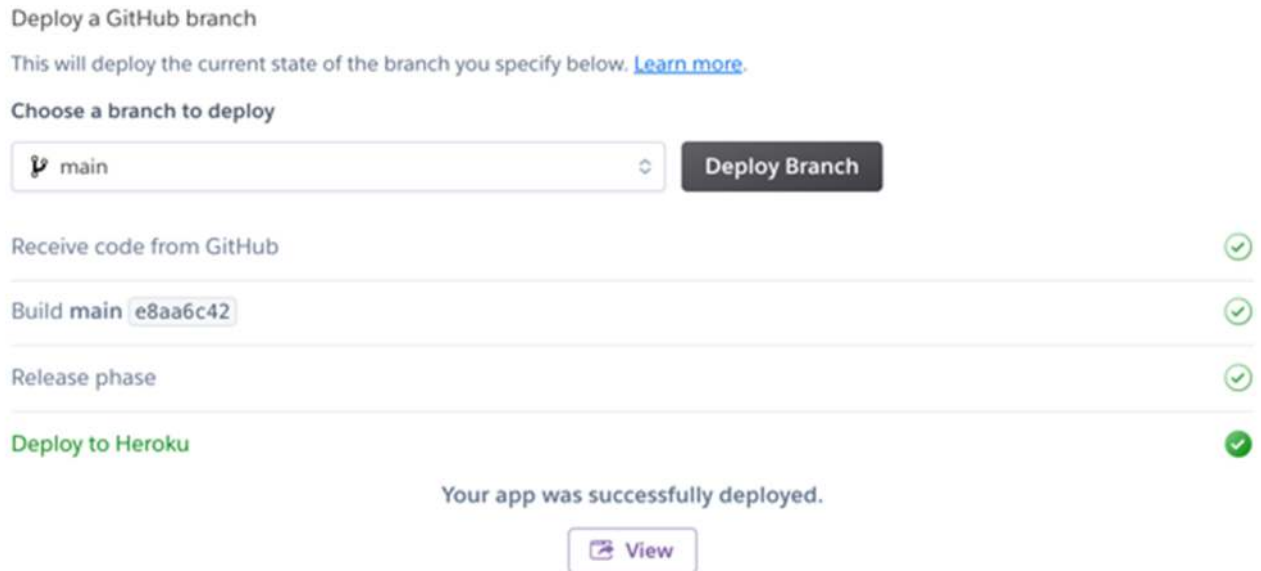


Рисунок 3.17 – Результат розгортання чат-боту на платформу Heroku

## 4 ІНСТРУКЦІЯ КОРИСТУВАЧА

Застосунок складається з наступних функціональних секцій, розбитих на сцени:

- початковий стан чату (рисунок 4.1);
- діалог;
- перевірка граматики;
- отримання інформації стосовно правильного використання слів та словосполучень.

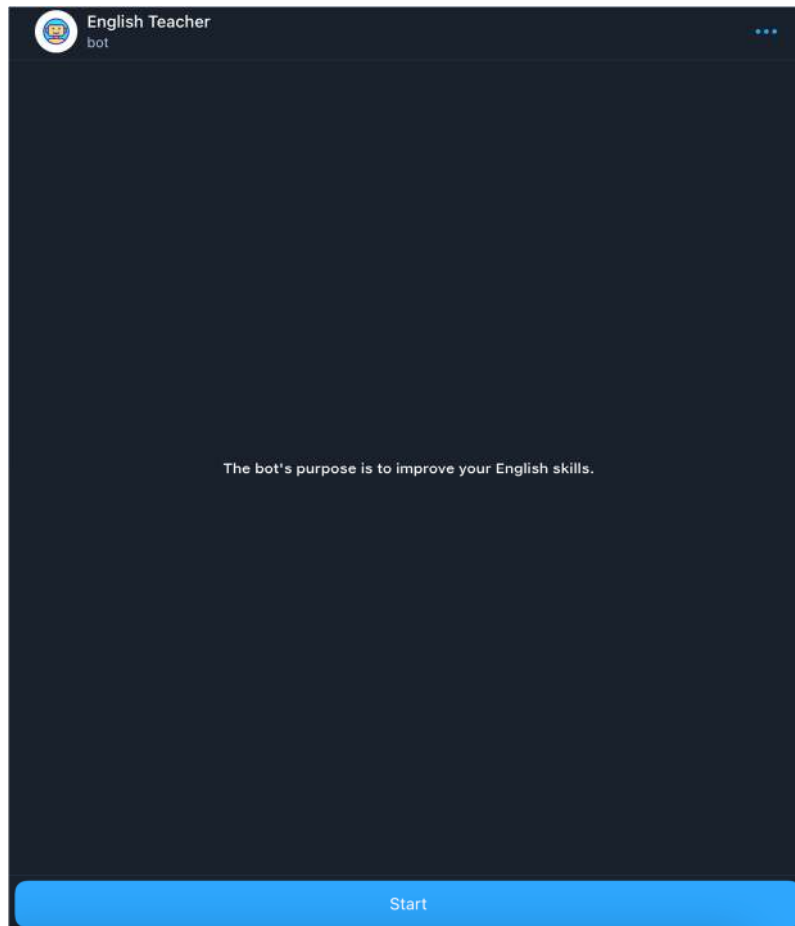


Рисунок 4.1 – Початковий стан чату

При натисканні на кнопку «Start» користувачу відображається головне меню з усіма доступними командами (рисунок 4.2).

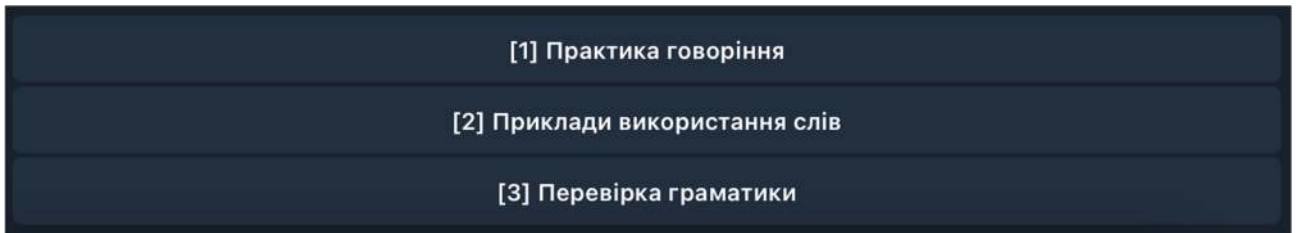


Рисунок 4.2 – Головне меню чат-боту

Після натискання кнопки «Практика говоріння», користувач входить у режим Діалогу. У цьому режимі користувачу потрібно вказати свій поточний рівень знань з англійської мови та обрати тему для спілкування. (рисунок 4.3).

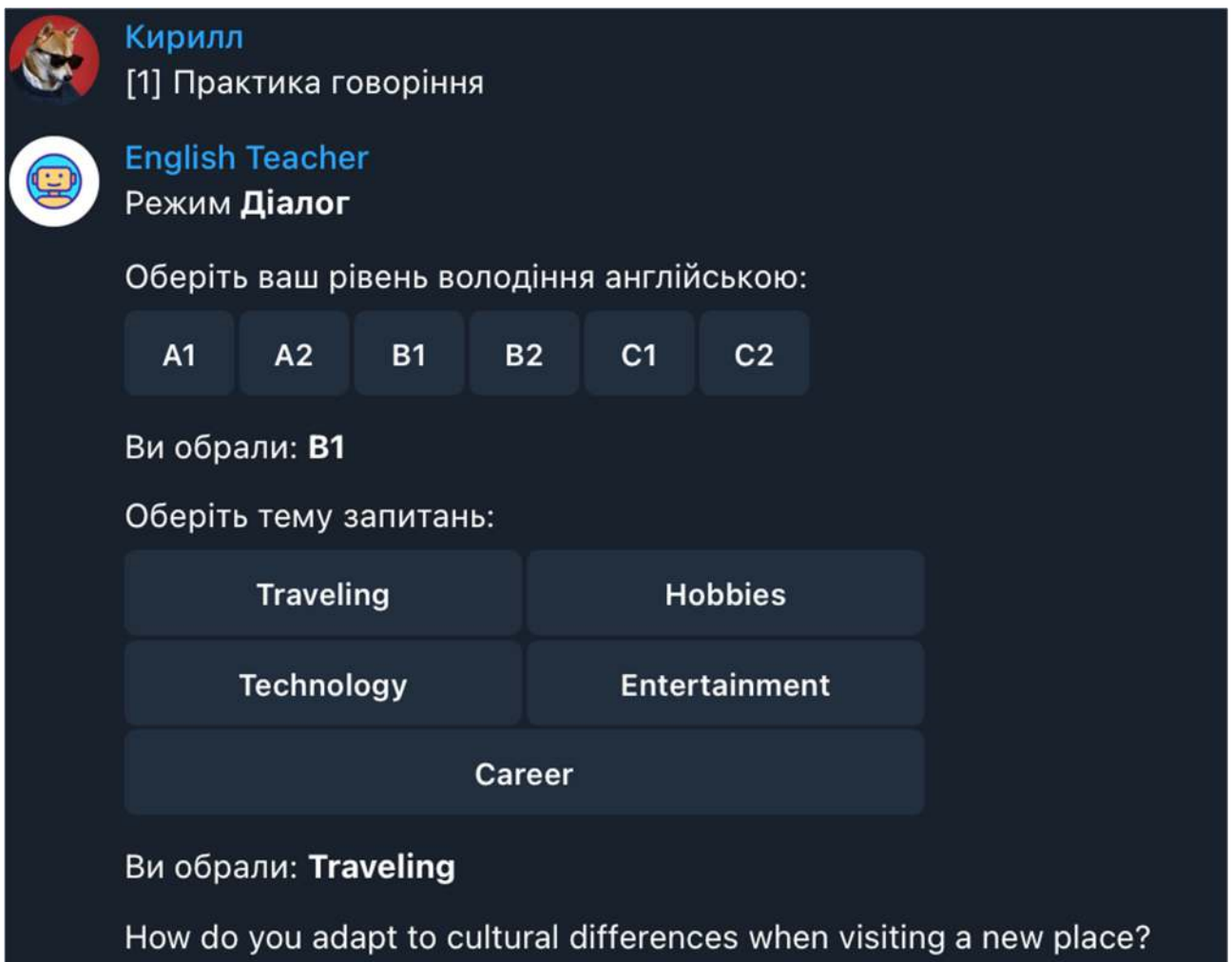


Рисунок 4.3 – Отримання питання базованого на рівні англійської мови та теми

Зараз користувачу потрібно відповісти на це питання голосом. Після цього з'явиться повідомлення з виправленими помилками, рекомендаціями, варіантами покращити сказане (рисунок 4.4).

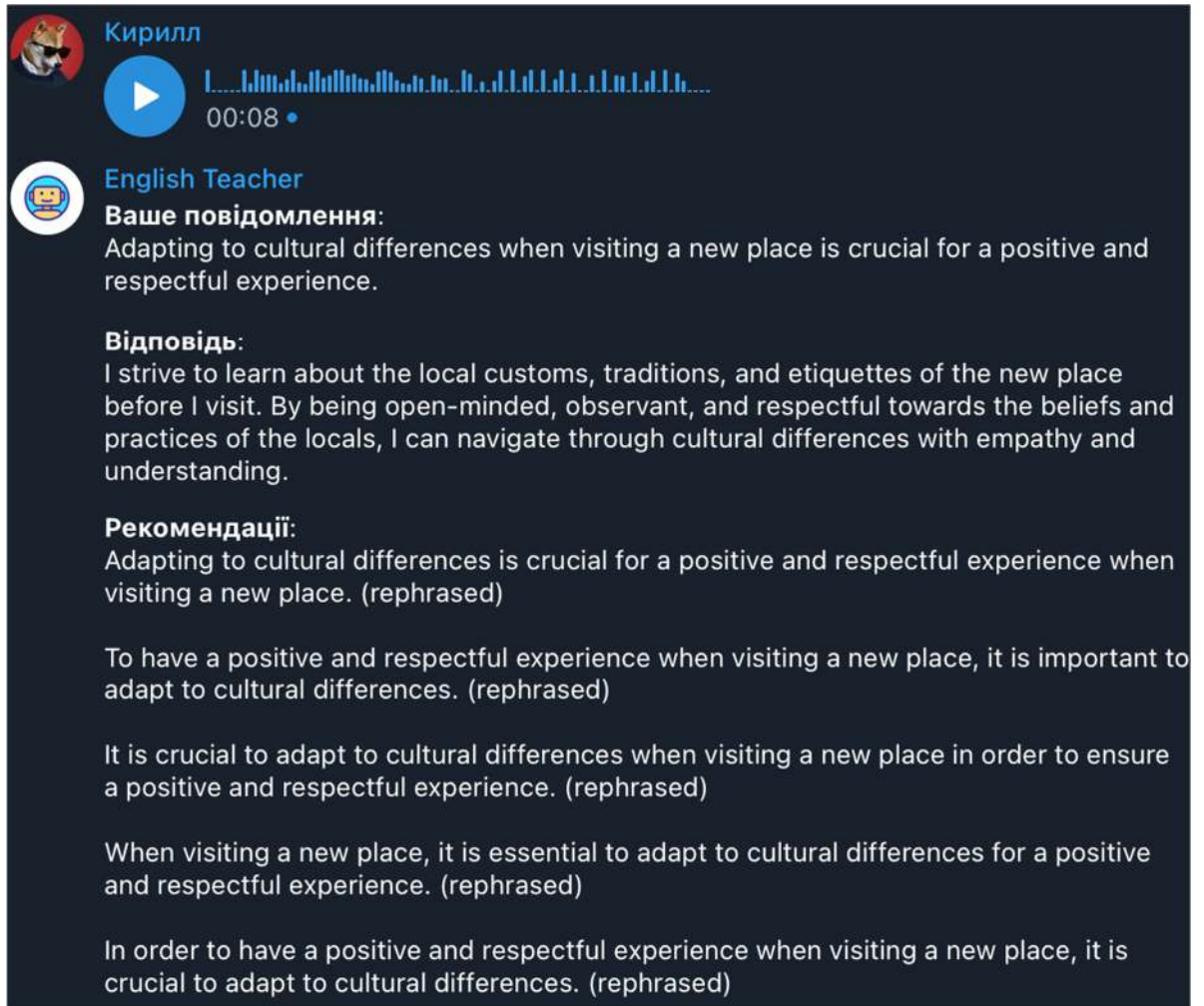


Рисунок 4.4 – Отримання повідомлення з виправленими помилками

Далі буде виведено слова, які користувач часто використовує. До кожного слова буде додано по три синоніми. Також будуть зображені кнопки щоб продовжити або почати діалог наново та кнопки головного меню (рисунок 4.5).

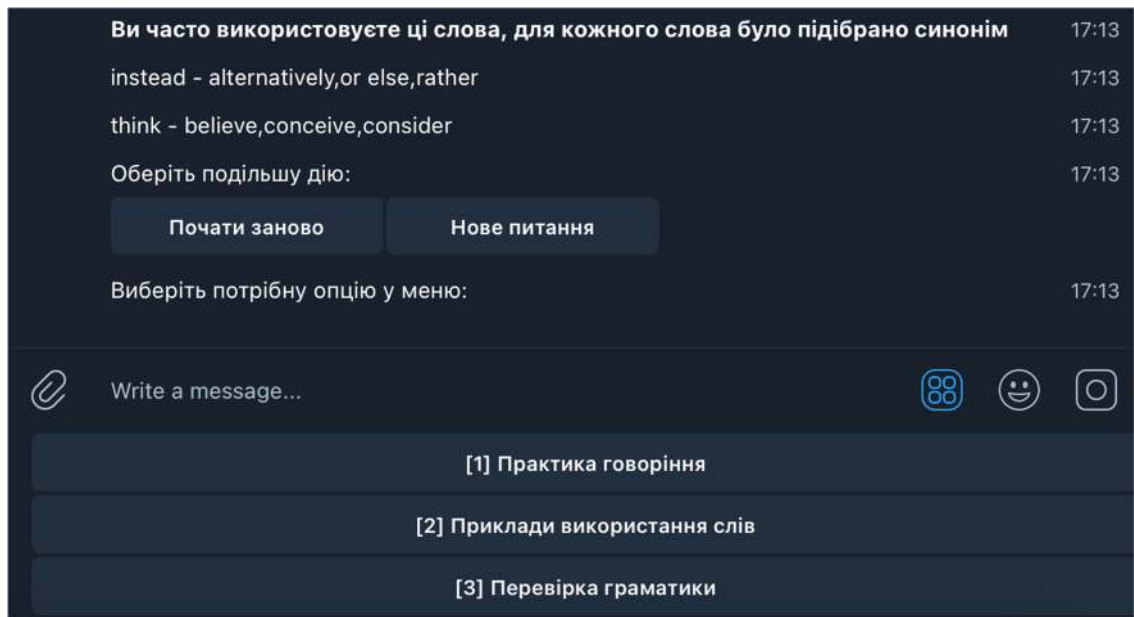


Рисунок 4.5 – Часто використовані слова

Після натискання кнопок «Приклади використання слів» користувачу потрібно ввести слово або словосполучення. Після цього він отримає повідомлення з прикладами використання цих слів або словосполучень у правильному контексті (рисунок 4.6).

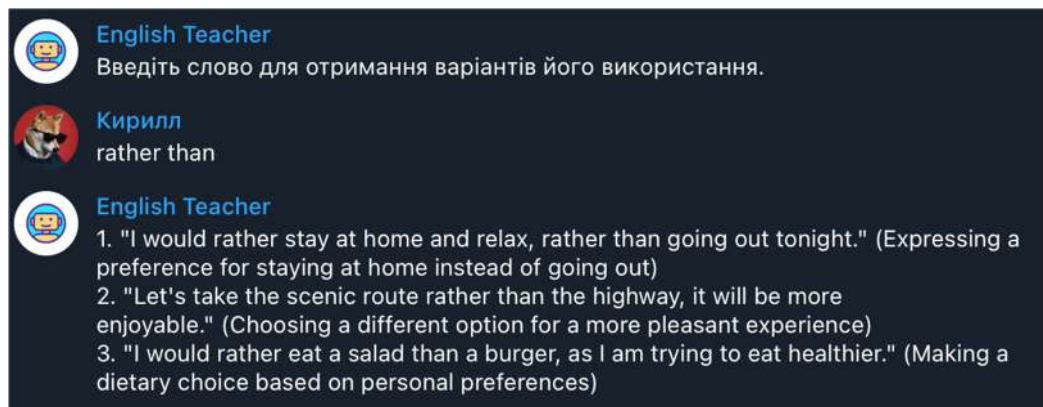


Рисунок 4.5 – Приклади використання слів

Після натискання кнопок «Перевірка граматики» користувачу потрібно ввести речення або невеликий об'єм тексту. Після цього він отримає повідомлення з виправленим текстом (рисунок 4.7).

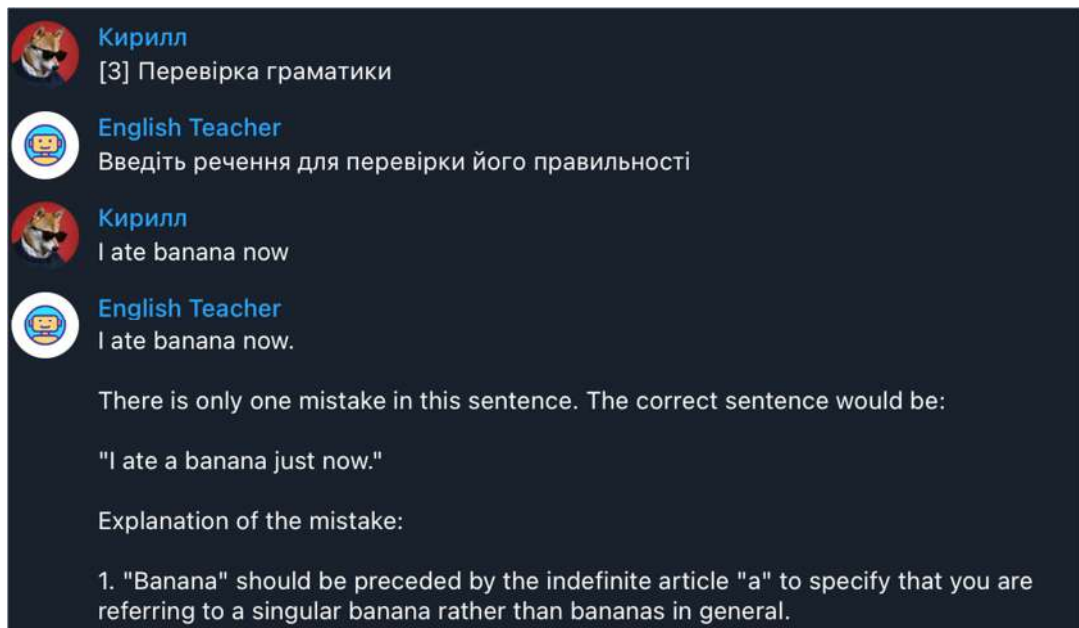


Рисунок 4.7 – Перевірка граматики

Таким чином, в результаті було створено чат-бот, якій виконує такі функції, а саме можливість відповідати на сгенеровані питання відповідно до рівня англійської мови голосом, отримувати поради стосовно покращення рівню мови.

## ВИСНОВКИ

Проведена робота присвячена створенню чат-боту з інтегрованою системою штучного інтелекту для ефективного вивчення англійської мови. Застосування чат-бота в цьому контексті спрямоване на спрощення процесу навчання та покращення якості вивчення англійської мови.

В ході виконання кваліфікаційної роботи було проведено глибоке дослідження предметної області, визначено ключові вимоги до системи та виявлені основні бізнес-процеси. Аналіз вимог охопив усі аспекти системи, включаючи функціонал, програмне та технічне забезпечення.

В рамках роботи було проведено дослідження різних технологій для побудови телеграм-бота. З результатів цього дослідження були вибрані ключові мови програмування та технології, такі як JavaScript, фреймворк Telegraf, платформа Node.js, Docker, OpenAI API та Telegram API.

Аналіз існуючих сервісів для вивчення англійської мови підтвердив існуючі обмеження багатьох з них, особливо при безкоштовному використанні. У відповідь на це було розроблено чат-бот на основі Telegram, який усунув визначені недоліки існуючих сервісів та об'єднав їх ключові можливості. Такий чат-бот є перспективним рішенням для вивчення англійської мови, надаючи ефективний та розширений функціонал.

Однією з ключових функціональних особливостей бота є можливість взаємодії з ним голосовими повідомленнями. Яку було реалізовано за допомогою використання штучного інтелекту, а саме GPT-3.5 Turbo та Whisper-1 моделей.

Ця функція дозволяє користувачам розвивати навички усної мови, а також забезпечує більш природний та інтуїтивний процес навчання. Голосові діалоги з ботом створюють атмосферу живого спілкування англійською мовою, що сприяє більш ефективному освоєнню мовних навичок.

Додатково в боті реалізована функція перевірки граматики в тексті. Це інструмент, що надає користувачам миттєвий зворотний зв'язок щодо

правильності граматичних конструкцій у їх повідомленнях. Такий аспект навчання сприяє формуванню грамотної мовної будови та підвищенню впевненості в власних мовних навичках.

Ще однією важливою можливістю телеграм-бота є функція надання прикладів використання різних слів в правильних ситуаціях. Цей інструмент допомагає користувачам освоювати не лише окремі слова, але й їх контекстуальне застосування, що значно збагачує словниковий запас та сприяє більш глибокому розумінню мови.

У результаті даного дослідження було продемонстровано, що розроблений телеграм-бот ефективно підтримує процес навчання англійської мови, забезпечуючи учням сучасні та інноваційні засоби для досягнення кращих результатів. Бот успішно інтегрує в себе технології голосової взаємодії, граматичної перевірки та контекстуального навчання, роблячи процес вивчення англійської мови більш цікавим, доступним та результативним.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Боты Telegram: всё, что вы хотели про них спросить [Електронний ресурс]. – Режим доступу : [www/ URL: http://mediatoolbox.ru/blog/boty-telegram-vsyo-chto-vyi-hoteli-pro-nih-sprosit /](http://mediatoolbox.ru/blog/boty-telegram-vsyo-chto-vyi-hoteli-pro-nih-sprosit/) – 10.11.2015 г. – Загол. з екрану.
2. Telegram [Електронний ресурс]. – Режим доступу : [www/ URL: https://ru.wikipedia.org/wiki/Telegram /](https://ru.wikipedia.org/wiki/Telegram/) – 20.03.2022 г. – Загол. з екрану.
3. Штучний інтелект [Електронний ресурс]. – Режим доступу : [www/ URL: https://osvita.diia.gov.ua/courses/artificial-intelligence](https://osvita.diia.gov.ua/courses/artificial-intelligence) – 20.03.2022 г. – Загол. з екрану.
4. Ільїна І., Реука К. Інтегрована система штучного інтелекту в телеграм-бот для вивчення англійської мови. проблеми інформатизації : Тези доп. одинадцятої міжнар. науково-техн. конф., м. Баку – Харків – Бельсько-Бяла, 16–17 листоп. 2023 р. 23. С. 14. [doi.org/10.32620/PI.23.t1](https://doi.org/10.32620/PI.23.t1).
5. Платформа Heroku [Електронний ресурс]. – Режим доступу : [www/ URL: https://ru.wikipedia.org/wiki/Heroku/](https://ru.wikipedia.org/wiki/Heroku/) – 02.06.2014 г. – Загол. з екрану.
6. Фреймворк Telegraf [Електронний ресурс]. – Режим доступу : [www/ URL: https://www.npmjs.com/package/telegraf/](https://www.npmjs.com/package/telegraf/) – 28.05.2022 г. – Загол. з екрану.
7. ChatGPT [Електронний ресурс]. – Режим доступу : [www/ URL: https://wikipedia.org/wiki/ChatGPT](https://wikipedia.org/wiki/ChatGPT) – 18.10.2023 г. – Загол. з екрану.
8. GPT [Електронний ресурс]. – Режим доступу : [www/ URL: https://wikipedia.org/wiki/Generative\\_pre-trained\\_transformer](https://wikipedia.org/wiki/Generative_pre-trained_transformer) – 19.11.2023 г. – Загол. з екрану.
9. GPT-3.5 Turbo [Електронний ресурс]. – Режим доступу : [www/ URL: https://openai.com/blog/gpt-3-5-turbo-fine-tuning-and-api-updates](https://openai.com/blog/gpt-3-5-turbo-fine-tuning-and-api-updates) – 04.03.2022 г. – Загол. з екрану.
10. Docker [Електронний ресурс]. – Режим доступу : [www/ URL: https://docs.docker.com/](https://docs.docker.com/) – Загол. з екрану.
11. NPM [Електронний ресурс]. – Режим доступу : [www/ URL: https://www.npmjs.com/](https://www.npmjs.com/)

<https://docs.npmjs.com/about-npm> – Загол. з екрану.

12. What is JavaScript? [Електронний ресурс]. – Режим доступу : [www/](http://www/) URL:[https://developer.mozilla.org/enUS/docs/Learn/JavaScript/First\\_steps/What\\_is\\_JavaScript](https://developer.mozilla.org/enUS/docs/Learn/JavaScript/First_steps/What_is_JavaScript) – Загол. з екрану.

13. Getting started – What is Git? [Електронний ресурс]. – Режим доступу : [www/](http://www/) URL: <https://git-scm.com/book/en/v2/Getting-Started-What-is-Git%3FGit> – Загол. з екрану.

14. Whisper [Електронний ресурс]. – Режим доступу : [www/](http://www/) URL: <https://platform.openai.com/docs/guides/speech-to-text> – Загол. з екрану.

15. Axios [Електронний ресурс]. – Режим доступу : [www/](http://www/) URL: <https://axios-http.com/docs/intro> – Загол. з екрану.