

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Рівненський державний гуманітарний університет
Кафедра інформаційних технологій та моделювання**

**КВАЛІФІКАЦІЙНА РОБОТА
НА ТЕМУ:
«Чатбот з використанням штучного
інтелекту для вступників ФМІ РДГУ»**

Виконав:

здобувач 4 курсу
спеціальності 122 Комп'ютерні
науки

Федорчук Ілля Олександрович

Науковий керівник:

викл. Гаврюсєв С.М.

Рівне – 2026

ЗМІСТ

ВСТУП

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ

1.1 Особливості використання чатботів у сфері освіти

1.1.1 Роль чатботів для вступників закладів вищої освіти

1.1.2 Основні вимоги до інформаційного чатбота

1.2 Аналіз сучасних платформ та сервісів

1.2.1 Порівняння існуючих чатботів

1.2.2 Переваги використання штучного інтелекту

РОЗДІЛ 2. ТЕХНОЛОГІЇ РОЗРОБКИ ЧАТБОТА

2.1 Огляд Telegram Bot API

2.2 Використання мови JavaScript та Python

2.2.1 Використання штучного інтелекту у чатботі

2.2.2 Переваги хмарних технологій

2.3 Формат JSON та обмін даними

2.4 Використання Webhooks

2.5 Безпека та захист даних

РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ СИСТЕМИ

3.1 Архітектура чатбота

3.2 Взаємодія користувача з ботом

3.3 Реалізація функціоналу чатбота

3.4 Інтеграція з AI-моделлю

3.5 Тестування системи

ВСТУП

Актуальність дослідження

У сучасному інформаційному суспільстві цифрові технології активно впроваджуються в усі сфери діяльності людини, зокрема й у систему освіти. Заклади вищої освіти постійно вдосконалюють способи взаємодії зі студентами та абітурієнтами, використовуючи сучасні засоби комунікації. Особливо актуальним є питання швидкого та зручного надання інформації вступникам, адже саме від доступності та зрозумілості інформації часто залежить вибір майбутнього місця навчання.

Під час вступної кампанії абітурієнти стикаються з необхідністю отримання великої кількості різноманітної інформації: відомостей про спеціальності, умови вступу, прохідні бали, перелік необхідних документів, вартість навчання, наявність бюджетних місць, особливості освітнього процесу та студентського життя. Значна частина запитань, які ставлять вступники, є однотипними, проте працівники приймальної комісії змушені витратити багато часу на постійне консультування. У період активної вступної кампанії це створює додаткове навантаження на адміністрацію факультету та може призводити до затримок у наданні відповідей.

Одним із сучасних способів вирішення даної проблеми є використання чатботів із застосуванням технологій штучного інтелекту. Чатбот являє собою програмну систему, здатну автоматично взаємодіяти з користувачем у форматі текстового діалогу. Завдяки використанню елементів штучного інтелекту така система може аналізувати зміст запитів користувача, розпізнавати ключові слова та формувати логічні й зрозумілі відповіді. Це дозволяє значно покращити якість інформаційної підтримки

та забезпечити швидке отримання необхідної інформації у будь-який час доби.

Використання чатботів у сфері освіти стає дедалі популярнішим, оскільки вони забезпечують автоматизацію рутинних процесів, підвищують ефективність комунікації та покращують взаємодію між закладом освіти та вступниками. Крім того, чатботи дозволяють скоротити час очікування відповіді, зменшити навантаження на працівників приймальної комісії та зробити процес отримання інформації більш комфортним для користувачів.

Для реалізації системи в даній роботі було обрано месенджер Telegram, який є одним із найпопулярніших засобів онлайн-спілкування серед молоді та студентів. Telegram має простий і зрозумілий інтерфейс, підтримує швидкий обмін повідомленнями та надає відкритий програмний інтерфейс Telegram Bot API для створення власних ботів. Завдяки цьому розробка чатбота не потребує створення окремого мобільного застосунку чи вебсайту, а користувачі можуть взаємодіяти із системою через звичний для них месенджер.

Актуальність теми також обумовлена стрімким розвитком технологій штучного інтелекту. Сучасні AI-сервіси дозволяють створювати більш «розумні» системи, здатні підтримувати природну мову спілкування, аналізувати запити користувачів та формувати інформативні відповіді. Інтеграція штучного інтелекту у чатботи значно розширює їх функціональні можливості та робить взаємодію з користувачем більш природною та ефективною.

Таким чином, розробка чатбота для вступників факультету математики та інформатики РДГУ є актуальною задачею, що спрямована на вдосконалення інформаційної підтримки абітурієнтів, автоматизацію процесу консультування та впровадження сучасних інформаційних технологій у діяльність навчального закладу.

Метою дослідження

Метою даної курсової роботи є розробка чатбота з використанням технологій штучного інтелекту для вступників факультету математики та інформатики РДГУ, який забезпечить автоматизоване надання консультацій та швидкий доступ до необхідної інформації.

Завдання дослідження

Для досягнення поставленої мети необхідно виконати такі завдання:

1. Провести аналіз предметної області та дослідити особливості інформаційної підтримки вступників.
2. Проаналізувати існуючі аналоги чатботів та визначити їх переваги й недоліки.
3. Визначити функціональні та нефункціональні вимоги до системи.
4. Обґрунтувати вибір технологій і засобів розробки чатбота.
5. Спроектувати архітектуру системи та структуру взаємодії компонентів.
6. Реалізувати програмний прототип чатбота з використанням Telegram Bot API та AI-сервісів.
7. Провести тестування роботи системи та оцінити ефективність її функціонування.
8. Проаналізувати результати роботи чатбота та можливості його подальшого вдосконалення.

Об'єкт дослідження

Об'єктом дослідження є процес автоматизації інформаційної підтримки вступників закладу вищої освіти.

Предмет дослідження

Предметом дослідження є методи, засоби та технології створення чатботів із використанням штучного інтелекту для автоматизації консультацій та обробки текстових запитів користувачів.

Методи дослідження

У процесі виконання курсової роботи було використано методи аналізу наукової та технічної літератури, порівняння сучасних програмних рішень, методи об'єктно-орієнтованого програмування, аналізу API та тестування програмного забезпечення. Також використовувалися методи моделювання програмних систем та аналізу ефективності роботи чатбота.

Практичне значення роботи

Практичне значення роботи полягає у створенні функціонального прототипу чатбота, який може використовуватись факультетом математики та інформатики РДГУ для автоматизації процесу консультування вступників. Розроблена система дозволяє швидко надавати відповіді на поширені запитання, покращує взаємодію між факультетом та абітурієнтами, а також демонструє можливості застосування технологій штучного інтелекту у сфері освіти.

Структура роботи

Курсова робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків. У першому розділі розглядається аналіз предметної області та існуючих рішень. У другому розділі описується проектування та вибір технологій розробки чатбота. Третій розділ присвячений практичній реалізації системи, тестуванню її функціоналу та аналізу отриманих результатів.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ

1.1 Особливості використання чатботів у сфері освіти

У сучасних умовах розвитку інформаційних технологій та цифрової трансформації суспільства система освіти активно впроваджує новітні програмні рішення для автоматизації навчального процесу та покращення комунікації між закладами освіти й користувачами. Одним із найбільш перспективних напрямів розвитку цифрових сервісів є використання чатботів, які дозволяють автоматизувати процес взаємодії з користувачами та забезпечувати швидке отримання необхідної інформації.

Сучасні освітні заклади постійно стикаються з великою кількістю інформаційних запитів від студентів, викладачів та абітурієнтів. Особливо актуальною ця проблема стає під час вступної кампанії, коли приймальні комісії змушені щоденно обробляти десятки або навіть сотні однотипних звернень. Значна частина таких запитів стосується спеціальностей, умов вступу, необхідних документів, вартості навчання, термінів подачі заяв та інших організаційних питань.

У зв'язку з цим виникає потреба у використанні автоматизованих систем підтримки користувачів, здатних швидко та ефективно надавати відповіді на поширені запитання. Одним із таких рішень є чатботи — спеціалізовані програмні системи, що взаємодіють із користувачем у форматі текстового діалогу. Вони дозволяють автоматизувати процес консультування та значно зменшити навантаження на працівників закладу освіти.

Освітні чатботи активно використовуються університетами та коледжами для вирішення різних задач. Вони можуть надавати довідкову інформацію, допомагати у навігації навчальними ресурсами, інформувати про розклад занять, повідомляти про важливі події та відповідати на типові запитання користувачів. Крім того, сучасні чатботи можуть інтегруватися із системами дистанційного навчання, електронними журналами та іншими інформаційними сервісами навчального закладу.

Важливою перевагою чатботів є можливість цілодобової роботи. На відміну від працівників приймальної комісії або технічної підтримки, чатбот здатний безперервно обробляти повідомлення користувачів незалежно від часу доби чи кількості одночасних звернень. Це особливо важливо для вступників, які можуть шукати інформацію у вечірній час або у вихідні дні.

Ще однією суттєвою перевагою є висока швидкість надання відповідей. Користувач отримує інформацію практично миттєво, що значно підвищує комфорт взаємодії із системою та позитивно впливає на загальне враження від роботи із закладом освіти.

Окремої уваги заслуговує використання технологій хмарних обчислень у роботі чатботів. Завдяки хмарним сервісам система може працювати стабільно, незалежно від навантаження, а також забезпечувати доступ до сервісу великій кількості користувачів одночасно. Хмарні технології дозволяють зменшити витрати на підтримку серверної інфраструктури та забезпечують зручне адміністрування системи.

Таким чином, використання чатботів у сфері освіти є ефективним інструментом автоматизації інформаційної підтримки користувачів, що дозволяє підвищити якість комунікації, скоротити навантаження на персонал закладу освіти та впроваджувати сучасні цифрові технології у навчальний процес.

1.1.1 Роль чатботів для вступників закладів вищої освіти

Процес вступу до закладу вищої освіти є одним із найважливіших етапів для абітурієнтів. У цей період вступники потребують швидкого доступу до великого обсягу актуальної інформації, пов'язаної з умовами вступу, вибором спеціальності та організаційними питаннями. Через значну кількість інформації абітурієнти часто стикаються з труднощами під час пошуку необхідних даних на офіційних сайтах університетів або в соціальних мережах.

У більшості випадків вступники звертаються до приймальної комісії із типовими питаннями, які повторюються щороку. Це створює значне навантаження на працівників факультету та потребує додаткових ресурсів для організації консультаційної підтримки. Саме тому використання чатботів для автоматизації процесу консультування є актуальним та ефективним рішенням.

Чатбот для вступників виконує роль електронного консультанта, який автоматично надає відповіді на поширені запитання користувачів. Завдяки цьому вступники можуть швидко отримати необхідну інформацію без необхідності телефонувати до приймальної комісії або очікувати відповіді електронною поштою.

Найбільш важливою для вступників є інформація щодо:

- спеціальностей та освітніх програм;
- умов вступу;
- конкурсних предметів;
- переліку необхідних документів;
- термінів подання заяв;
- вартості навчання;
- наявності бюджетних місць;
- контактної інформації факультету;
- можливостей проживання у гуртожитку;
- особливостей студентського життя.

Використання чатбота дозволяє забезпечити оперативне надання актуальної інформації у зручному форматі. Крім того, система може швидко оновлювати дані у разі змін правил вступу або інших організаційних питань.

Важливою перевагою є також доступність сервісу через популярні месенджери. Більшість сучасної молоді активно використовує Telegram у повсякденному житті, тому взаємодія із чатботом не потребує встановлення додаткового програмного забезпечення чи проходження складної реєстрації.

Сучасні чатботи із використанням штучного інтелекту здатні аналізувати природну мову користувача та формувати більш точні відповіді навіть у випадку, якщо запит сформульований у довільній формі. Це дозволяє зробити спілкування із системою більш природним та комфортним.

Таким чином, чатботи для вступників є важливим елементом цифровізації закладів вищої освіти та дозволяють значно покращити процес інформаційної підтримки абітурієнтів.

1.1.2 Основні вимоги до інформаційного чатбота

Під час розробки інформаційного чатбота для вступників необхідно враховувати низку функціональних та технічних вимог, які забезпечують стабільну та ефективну роботу системи. Від правильного визначення вимог залежить зручність використання чатбота, його продуктивність та якість взаємодії з користувачами.

Однією з головних вимог є швидка обробка повідомлень. Система повинна миттєво реагувати на запити користувачів та забезпечувати оперативне надання відповідей. Затримки в роботі чатбота можуть негативно впливати на якість взаємодії та створювати незручності для користувачів.

Важливою вимогою є підтримка української мови. Оскільки система орієнтована на українських вступників, чатбот повинен коректно обробляти текстові повідомлення українською мовою та формувати зрозумілі відповіді.

Ще однією важливою складовою є інтеграція з Telegram. Даний месенджер є одним із найпопулярніших серед студентської молоді та має відкритий програмний інтерфейс Telegram Bot API, який дозволяє створювати функціональних чатботів із підтримкою Webhooks та інтеграцією із зовнішніми сервісами.

Оскільки тема роботи пов'язана з технологіями хмарних обчислень, важливою вимогою є стабільна робота системи у хмарному середовищі. Використання хмарних платформ дозволяє забезпечити цілодобову доступність чатбота, автоматичне масштабування системи та зменшення витрат на обслуговування серверного обладнання.

Особливу увагу було приділено можливості використання технологій штучного інтелекту. AI дозволяє чатботу аналізувати зміст повідомлень користувача, розпізнавати наміри та формувати більш природні відповіді. Завдяки цьому система може працювати не лише за шаблонними сценаріями, а й підтримувати більш гнучке спілкування.

Також важливою вимогою є простий та зрозумілий інтерфейс користувача. Чатбот повинен бути максимально простим у використанні навіть для людей без технічних навичок. Всі повідомлення та команди мають бути логічними, короткими та зрозумілими.

Крім основних вимог, система повинна відповідати таким характеристикам:

- стабільність роботи;
- безпечність передачі даних;
- можливість масштабування;
- простота адміністрування;
- підтримка оновлення функціоналу;
- зручність інтеграції із зовнішніми AI-сервісами.

Таким чином, визначення функціональних та технічних вимог є важливим етапом проєктування чатбота та безпосередньо впливає на ефективність роботи системи.

1.2 Аналіз сучасних платформ та сервісів

Сьогодні існує велика кількість платформ та сервісів для створення чатботів, які дозволяють реалізовувати системи різного рівня складності. Сучасні інструменти розробки підтримують інтеграцію із месенджерами, AI-моделями, хмарними сервісами та базами даних.

До найбільш популярних платформ належать:

- Telegram Bot API;
- Dialogflow;
- Microsoft Bot Framework;
- ManyChat;
- Botpress;
- Rasa та інші.

Кожна із платформ має власні особливості, переваги та недоліки. Деякі сервіси орієнтовані на створення комерційних чатботів, інші — на корпоративні рішення або інтеграцію зі складними AI-системами.

Dialogflow від компанії Google є потужною платформою для створення AI-чатботів із підтримкою обробки природної мови. Сервіс дозволяє створювати складні сценарії взаємодії, проте потребує додаткового налаштування та має певні обмеження у безкоштовній версії.

Microsoft Bot Framework орієнтований переважно на корпоративні системи та інтеграцію із сервісами Microsoft. Платформа має широкий функціонал, однак є більш складною для початківців.

ManyChat та аналогічні сервіси здебільшого використовуються у сфері маркетингу та електронної комерції. Їх безкоштовний функціонал часто є обмеженим, що ускладнює використання для повноцінних освітніх проєктів.

Для реалізації чатбота у даній роботі було обрано Telegram Bot API. Дане рішення забезпечує оптимальне поєднання простоти, функціональності та доступності. Telegram API є безкоштовним, добре документованим та підтримує велику кількість бібліотек для різних мов програмування.

1.2.1 Порівняння існуючих чатботів

Аналіз існуючих рішень показав, що більшість готових чатботів мають ряд недоліків та обмежень. Частина платформ потребує оформлення платної підписки для використання розширених функцій, інші вимагають складного налаштування або мають обмежені можливості інтеграції із зовнішніми сервісами.

Telegram Bot API має низку суттєвих переваг:

- безкоштовне використання;
- підтримка Webhooks;
- висока швидкість роботи;
- велика кількість бібліотек для розробки;
- можливість інтеграції з AI-моделями;
- підтримка хмарного розгортання;
- простота адміністрування;
- популярність серед молоді.

Ще однією важливою перевагою є підтримка хмарних технологій. Telegram-бот може бути розгорнутий на хмарних сервісах, що дозволяє забезпечити стабільну роботу системи без необхідності використання власного серверного обладнання.

У результаті проведеного аналізу було визначено, що Telegram Bot API є найбільш доцільним рішенням для створення чатбота для вступників ФМІ РДГУ.

1.2.2 Переваги використання штучного інтелекту

Використання технологій штучного інтелекту є одним із ключових напрямів розвитку сучасних інформаційних систем. На відміну від звичайних чатботів, які працюють лише за попередньо визначеними сценаріями, AI-чатботи здатні аналізувати природну мову користувача та формувати більш гнучкі відповіді.

Штучний інтелект дозволяє:

- аналізувати текстові запити користувачів;
- розпізнавати наміри та ключові слова;
- генерувати більш природні відповіді;
- підтримувати діалог у довільній формі;
- зменшувати кількість шаблонних сценаріїв;
- покращувати якість взаємодії із системою.

Завдяки AI користувач може ставити запитання у різних формулюваннях, а система буде намагатися правильно зрозуміти зміст повідомлення та надати відповідь. Це значно покращує зручність використання чатбота та робить спілкування більш природним.

Використання AI також дозволяє підвищити рівень автоматизації системи. Чатбот може обробляти складніші запити без необхідності створення великої кількості жорстко прописаних сценаріїв.

Крім того, технології штучного інтелекту добре поєднуються із хмарними обчисленнями. AI-моделі можуть працювати через хмарні API-сервіси, що дозволяє зменшити навантаження на локальну систему та забезпечити високу продуктивність роботи чатбота.

Таким чином, використання штучного інтелекту значно підвищує ефективність роботи інформаційного чатбота та дозволяє створити сучасну систему підтримки вступників закладу вищої освіти.

РОЗДІЛ 2. ТЕХНОЛОГІЇ РОЗРОБКИ ЧАТБОТА

2.1 Огляд Telegram Bot API

Telegram Bot API — це спеціалізований програмний інтерфейс, який надається месенджером Telegram для створення та керування чатботами. Даний API дозволяє розробникам створювати автоматизовані системи взаємодії з користувачами, які можуть отримувати повідомлення, обробляти команди, надсилати текстову інформацію, файли, кнопки та інші елементи інтерфейсу.

Telegram Bot API працює за принципом обміну HTTP-запитами між сервером Telegram та серверною частиною чатбота. Після створення бота користувач отримує унікальний API-токен, який використовується для авторизації запитів та взаємодії із сервером Telegram.

Однією з головних переваг Telegram Bot API є його простота та доступність. Для роботи з API не потрібно створювати складну серверну інфраструктуру, а документація Telegram містить велику кількість прикладів та описів методів. Завдяки цьому Telegram Bot API є популярним рішенням як для навчальних проєктів, так і для створення повноцінних комерційних систем.

Важливою особливістю Telegram Bot API є підтримка різних типів повідомлень. Чатбот може працювати не лише з текстом, а й із фотографіями, відео, документами, геолокацією та інтерактивними кнопками. Це дозволяє створювати більш функціональні та зручні системи взаємодії з користувачами.

У процесі роботи чатбот постійно взаємодіє із серверами Telegram через API-запити. Для цього використовуються спеціальні методи, кожен із яких виконує певну функцію.

Основними методами Telegram Bot API є:

- **sendMessage** — використовується для надсилання текстових повідомлень користувачу;
- **getUpdates** — дозволяє отримувати нові повідомлення від користувачів;
- **setWebhook** — використовується для налаштування автоматичної передачі повідомлень на сервер чатбота.

Метод **sendMessage** є одним із найважливіших, оскільки саме він забезпечує відправлення відповідей користувачу. За допомогою цього методу бот може надсилати звичайний текст, форматовані повідомлення, кнопки та інші елементи інтерфейсу.

Метод **getUpdates** працює за принципом періодичного опитування серверів Telegram. Сервер чатбота регулярно перевіряє наявність нових повідомлень та отримує їх у форматі JSON. Даний підхід є простим у реалізації, проте може створювати додаткове навантаження на сервер.

Більш сучасним способом роботи є використання Webhooks через метод **setWebhook**. У цьому випадку Telegram автоматично надсилає інформацію про нові повідомлення на вказану адресу сервера. Такий підхід дозволяє забезпечити миттєву реакцію системи та зменшити кількість зайвих запитів.

Ще однією перевагою Telegram Bot API є підтримка інтеграції із зовнішніми сервісами та AI-моделями. Завдяки цьому чатбот може взаємодіяти із хмарними платформами, базами даних, системами штучного інтелекту та іншими програмними компонентами.

Таким чином, Telegram Bot API є ефективним та гнучким інструментом для створення сучасних чатботів із підтримкою хмарних технологій та елементів штучного інтелекту.

2.2 Використання мови JavaScript та Python

Для розробки сучасних чатботів можуть використовуватись різні мови програмування, серед яких найбільш популярними є JavaScript та Python. Обидві мови мають значну кількість бібліотек і засобів для роботи з Telegram Bot API, обробки HTTP-запитів та інтеграції із сервісами штучного інтелекту.

JavaScript є однією з найпоширеніших мов програмування у сфері веброзробки. Вона активно використовується для створення серверної логіки, вебзастосунків та автоматизованих систем. Однією з переваг JavaScript є можливість роботи у хмарному середовищі Google Apps Script, що дозволяє створювати серверні застосунки без необхідності налаштування окремого вебсервера.

Python також є популярною мовою програмування для створення чатботів та AI-систем. Вона має велику кількість бібліотек для машинного навчання, обробки тексту та інтеграції із зовнішніми сервісами. Завдяки простому синтаксису Python часто використовується для створення систем штучного інтелекту та автоматизації обробки даних.

У даній роботі основна логіка чатбота реалізована за допомогою мови JavaScript у середовищі Google Apps Script. Дане рішення було обране через простоту інтеграції із сервісами Google, підтримку Webhooks та можливість швидкого розгортання застосунку у хмарному середовищі.

Google Apps Script дозволяє створювати серверні функції, які автоматично обробляють HTTP-запити від Telegram API. Крім того, сервіс підтримує інтеграцію з Google Sheets, що може використовуватись для зберігання інформації або ведення журналів роботи чатбота.

Використання JavaScript у поєднанні із хмарною платформою дозволяє:

- швидко розробляти функціонал чатбота;
- автоматизувати обробку повідомлень;
- інтегрувати AI-сервіси;
- працювати без окремого серверного обладнання;
- спростувати підтримку та оновлення системи.

Таким чином, використання JavaScript та хмарних технологій є ефективним рішенням для створення чатбота для вступників ФМІ РДГУ.

2.2.1 Використання штучного інтелекту у чатботі

Однією з ключових особливостей сучасних чатботів є використання технологій штучного інтелекту. На відміну від звичайних сценарних ботів, які працюють лише за заздалегідь прописаними правилами, AI-чатботи здатні аналізувати текстові повідомлення користувачів та формувати більш гнучкі відповіді.

У даній системі AI-модель використовується для аналізу повідомлень користувачів та генерації відповідей у природній формі. Це дозволяє забезпечити більш комфортне та «живе» спілкування між користувачем і системою.

Штучний інтелект дозволяє чатботу:

- аналізувати текстові запити;
- розпізнавати ключові слова;
- визначати наміри користувача;
- генерувати логічні відповіді;
- підтримувати діалог у довільній формі.

Завдяки AI користувач може формулювати запитання різними способами, а система буде намагатися правильно визначити зміст повідомлення та надати відповідь. Це значно покращує якість взаємодії та зменшує кількість жорстко прописаних сценаріїв.

Ще однією перевагою використання AI є можливість масштабування функціоналу системи. У майбутньому чатбот може бути доповнений

підтримкою голосових повідомлень, автоматичним перекладом тексту або персоналізованими рекомендаціями для вступників.

AI-модель взаємодіє із чатботом через API-запити. Серверна частина надсилає текст повідомлення користувача до AI-сервісу, після чого отримує сформовану відповідь та передає її назад у Telegram.

Таким чином, використання технологій штучного інтелекту дозволяє значно підвищити ефективність чатбота та зробити взаємодію із системою більш природною та зручною.

2.2.2 Переваги хмарних технологій

Сучасні інформаційні системи все частіше використовують технології хмарних обчислень, які дозволяють розміщувати програмні сервіси у віддаленому середовищі без необхідності використання власного серверного обладнання. Хмарні технології забезпечують високу доступність, масштабованість та зручність адміністрування програмних систем.

Для реалізації чатбота у даній роботі було використано платформу Google Apps Script, яка є хмарним середовищем для створення та виконання JavaScript-застосунків. Даний сервіс дозволяє розробляти серверну логіку без необхідності налаштування окремого вебсервера.

Google Apps Script забезпечує:

- serverless-архітектуру;
- автоматичне HTTPS-з'єднання;
- інтеграцію з Google Sheets;
- швидке розгортання застосунку;
- автоматичне масштабування системи;
- підтримку роботи через Webhooks.

Однією з головних переваг serverless-підходу є відсутність необхідності самостійного адміністрування серверів. Усі процеси

виконуються у хмарному середовищі Google, що значно спрощує підтримку та оновлення системи.

Автоматичне HTTPS-з'єднання забезпечує безпечну передачу даних між Telegram API та серверною частиною чатбота. Це особливо важливо для захисту інформації користувачів та стабільної роботи Webhooks.

Інтеграція з Google Sheets дозволяє використовувати електронні таблиці як просту базу даних для збереження інформації, статистики або журналів роботи системи.

Хмарні технології також забезпечують високу стабільність роботи чатбота. Система може одночасно обробляти велику кількість запитів без значного зниження продуктивності.

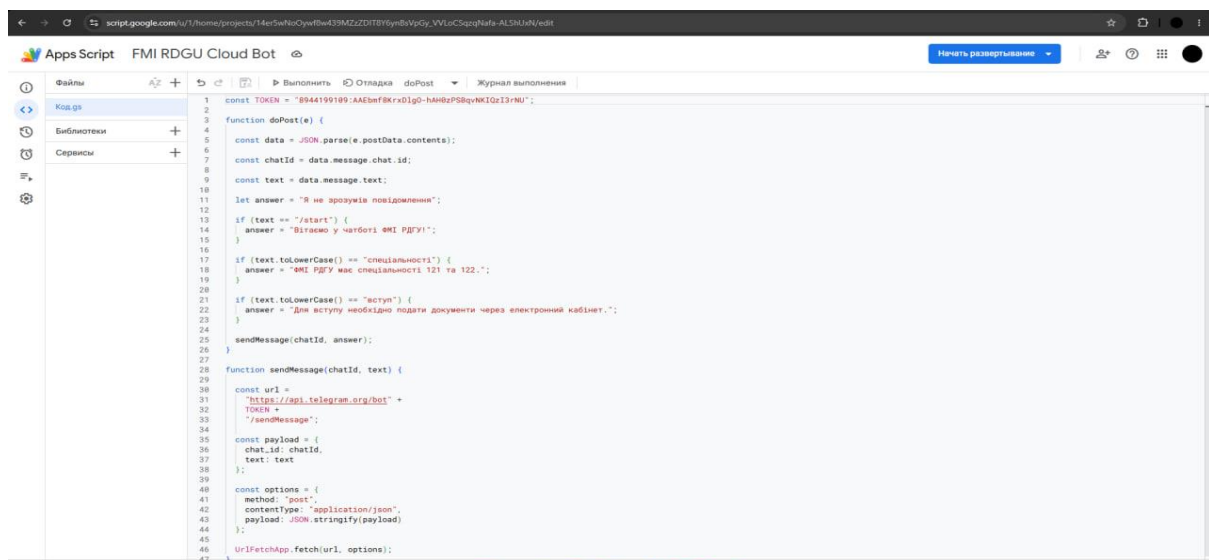


Рисунок 2.1 – Хмарне середовище Google Apps Script

Таким чином, використання хмарних технологій дозволяє створити стабільну, масштабовану та зручну систему підтримки вступників.

2.3 Формат JSON та обмін даними

Для передачі інформації між Telegram API, серверною частиною чатбота та AI-сервісами використовується формат JSON (JavaScript Object Notation). JSON є одним із найпоширеніших форматів обміну даними у сучасних вебзастосунках та API-системах.

Основною перевагою JSON є простота структури та зручність обробки даних у різних мовах програмування. Формат використовує структуру «ключ–значення», що дозволяє легко зчитувати та передавати інформацію між різними компонентами системи.

Telegram API надсилає повідомлення користувачів у форматі JSON. Сервер чатбота отримує ці дані, аналізує текст повідомлення та формує відповідь.

Приклад JSON-об'єкта:

```
{  
  "message": {  
    "chat": {  
      "id": 123456789  
    },  
    "text": "/start"  
  }  
}
```

У даному прикладі:

- **chat.id** містить ідентифікатор користувача;
- **text** містить текст повідомлення;
- **message** описує структуру отриманого повідомлення.

JSON також використовується для взаємодії із AI-моделями та зовнішніми API-сервісами. Завдяки універсальності формату система може легко інтегруватися із різними платформами та хмарними сервісами.

Таким чином, використання JSON забезпечує ефективний та зручний обмін даними між компонентами інформаційної системи.

2.4 Використання Webhooks

Webhook — це механізм автоматичної передачі даних між серверами через HTTP-запити. У Telegram Bot API Webhook використовується для автоматичного надсилання повідомлень на сервер чатбота після отримання нового повідомлення від користувача.

На відміну від методу **getUpdates**, який працює за принципом постійного опитування серверів Telegram, Webhooks дозволяють значно зменшити навантаження на систему та забезпечити миттєву реакцію чатбота.

Після налаштування Webhook Telegram автоматично передає всі нові повідомлення на вказану HTTPS-адресу сервера. Серверна частина отримує JSON-об'єкт із даними повідомлення, обробляє його та надсилає відповідь користувачу.

Використання Webhooks забезпечує:

- миттєву реакцію системи;
- менше навантаження на сервер;
- стабільну роботу чатбота;
- швидшу обробку повідомлень;
- ефективніше використання ресурсів.

Для роботи Webhooks необхідно використовувати HTTPS-з'єднання, оскільки Telegram приймає лише захищені адреси серверів. Google Apps

Script автоматично підтримує HTTPS, що значно спрощує процес налаштування.

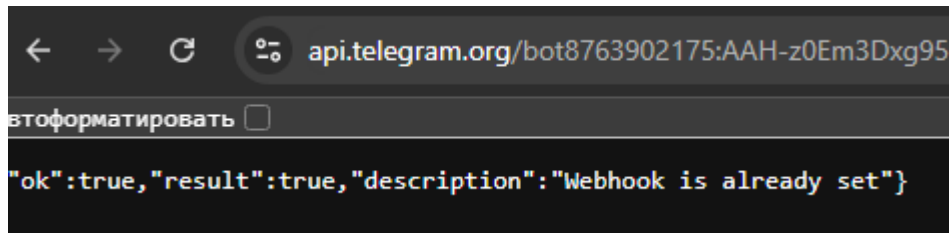


Рисунок 2.4 – Налаштування Webhook для Telegram-бота

Таким чином, використання Webhooks є більш сучасним та ефективним способом взаємодії між Telegram API та серверною частиною чатбота.

2.5 Безпека та захист даних

Безпека інформаційної системи є важливою складовою розробки чатбота, особливо при використанні хмарних технологій та зовнішніх API-сервісів. Захист даних дозволяє забезпечити стабільну роботу системи та запобігти несанкціонованому доступу до інформації.

Для захисту даних у системі використовуються:

- HTTPS-з'єднання;
- обмеження доступу до Google Apps Script;
- приховування API-токенів;
- перевірка вхідних запитів;
- контроль доступу до сервісів.

HTTPS-з'єднання забезпечує шифрування даних під час передачі між Telegram API та серверною частиною чатбота. Це дозволяє захистити інформацію від перехоплення сторонніми особами.

API-токени є важливою частиною системи авторизації, тому вони повинні зберігатися у захищеному вигляді та не публікуватися у відкритому доступі. У разі компрометації токена сторонні користувачі можуть отримати доступ до керування ботом.

Для додаткового захисту використовуються механізми перевірки вхідних запитів та обмеження доступу до адміністративних функцій системи.

Хмарне середовище Google Apps Script також забезпечує базові механізми захисту, автоматичне HTTPS-з'єднання та контроль доступу до проєкту.

Таким чином, використання сучасних механізмів захисту даних дозволяє забезпечити безпечну та стабільну роботу чатбота у хмарному середовищі.

РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ СИСТЕМИ

3.1 Архітектура чатбота

Проектування архітектури системи є одним із найважливіших етапів розробки чатбота, оскільки саме від структури програмного рішення залежить стабільність, швидкодія та можливість подальшого масштабування системи. Під час створення чатбота для вступників ФМІ РДГУ було обрано архітектуру, побудовану на основі хмарних технологій та взаємодії декількох сервісів.

Архітектура системи складається з трьох основних компонентів:

- Telegram Bot API;
- Google Apps Script;
- Google Sheets.

Кожен із компонентів виконує окрему функцію та забезпечує взаємодію між користувачем і серверною частиною системи.

Telegram Bot API використовується як основний інтерфейс взаємодії з користувачем. Саме через Telegram користувач надсилає повідомлення, отримує відповіді від системи та взаємодіє із функціоналом чатбота. Telegram забезпечує швидку доставку повідомлень, підтримку інтерактивних елементів та стабільну роботу системи.

Google Apps Script використовується як серверна частина чатбота. У даному середовищі реалізована основна логіка системи, обробка повідомлень, взаємодія із Telegram API та AI-моделлю. Перевагою використання Google Apps Script є можливість роботи у хмарному середовищі без необхідності налаштування окремого сервера.

Google Sheets використовується для збереження даних та ведення журналів роботи системи. За допомогою електронних таблиць можна зберігати інформацію про користувачів, історію повідомлень або статистику роботи чатбота. Даний підхід є простим у реалізації та не потребує використання складних систем керування базами даних.

Принцип роботи системи полягає у наступному:

1. Користувач надсилає повідомлення чатботу у Telegram.
2. Telegram Bot API передає дані через Webhook у Google Apps Script.
3. Серверна логіка аналізує повідомлення користувача.
4. У разі необхідності система звертається до AI-моделі.
5. Після обробки формується відповідь.
6. Telegram API надсилає відповідь користувачу.

Використання такої архітектури дозволяє забезпечити:

- швидку обробку повідомлень;
- стабільну роботу системи;
- простоту масштабування;
- інтеграцію із хмарними сервісами;
- підтримку AI-технологій;
- мінімальні витрати на серверну інфраструктуру.

Окремою перевагою є serverless-підхід, який використовується у Google Apps Script. У цьому випадку розробнику не потрібно самостійно адмініструвати сервери або налаштовувати складну інфраструктуру. Усі процеси виконуються у хмарному середовищі Google.

Таким чином, обрана архітектура забезпечує ефективну взаємодію між користувачем та системою, дозволяє інтегрувати AI-функціонал і відповідає сучасним вимогам до хмарних інформаційних систем.

3.2 Взаємодія користувача з ботом

Одним із важливих етапів проектування системи є організація взаємодії користувача із чатботом. Інтерфейс системи повинен бути максимально простим, зрозумілим та зручним для використання навіть користувачами без технічних навичок.

Взаємодія із системою здійснюється через месенджер Telegram. Для початку роботи користувач знаходить чатбота у Telegram та запускає його за допомогою команди `/start`. Після цього система автоматично надсилає привітальне повідомлення та список доступних команд або функцій.

Привітальне повідомлення містить коротку інформацію про можливості чатбота та допомагає користувачу швидко ознайомитися з функціоналом системи. Це особливо важливо для нових користувачів, які вперше взаємодіють із чатботом.

Основними функціями чатбота є:

- надання інформації про спеціальності;
- консультація щодо умов вступу;
- надання контактної інформації факультету;
- відповіді на типові запитання;
- інформування про вступну кампанію;
- підтримка діалогу у текстовому форматі.

Користувач може взаємодіяти із системою як через команди, так і за допомогою звичайних текстових повідомлень. Завдяки інтеграції AI-моделі чатбот здатний аналізувати природну мову та відповідати навіть на запити, сформульовані у довільній формі.

Під час отримання повідомлення система виконує такі дії:

1. Отримує повідомлення від Telegram API.
2. Аналізує текст запиту.
3. Визначає тип команди або намір користувача.
4. Формує відповідь.
5. Надсилає повідомлення назад користувачу.

Важливою особливістю системи є швидкість реагування на запити. Завдяки використанню Webhooks та хмарних технологій користувач отримує відповідь практично миттєво.

Крім того, використання Telegram забезпечує:

- зручний мобільний інтерфейс;
- підтримку кнопок та меню;
- швидку доставку повідомлень;
- підтримку мультимедійних файлів;
- доступність сервісу на різних пристроях.

Таким чином, взаємодія користувача із чатботом організована у максимально простому та зручному форматі, що дозволяє забезпечити ефективну інформаційну підтримку вступників.

3.3 Реалізація функціоналу чатбота

Реалізація функціоналу чатбота здійснювалася за допомогою мови програмування JavaScript у середовищі Google Apps Script. Основна логіка системи побудована на обробці HTTP-запитів, які надходять від Telegram Bot API через Webhooks.

Після отримання повідомлення серверна частина системи аналізує JSON-об'єкт, визначає текст повідомлення та виконує необхідну логіку обробки запиту.

Основною функцією системи є функція `doPost(e)`, яка автоматично викликається після отримання нового повідомлення від Telegram.

Приклад фрагмента коду:

```
function doPost(e) {  
  
    const data = JSON.parse(e.postData.contents);  
  
    const chatId = data.message.chat.id;  
  
    const text = data.message.text;  
  
    let reply = "Невідома команда";  
  
    if (text == "/start") {  
        reply = "Вітаємо у чатботі ФМІ РДГУ!";  
    }  
}
```

```
sendMessage(chatId, reply);
```

```
}
```

У даному фрагменті коду:

- `JSON.parse()` використовується для обробки JSON-даних;
- `chat.id` містить ідентифікатор користувача;
- `text` містить текст повідомлення;
- змінна `reply` використовується для формування відповіді.

Якщо користувач вводить команду `/start`, система надсилає привітальне повідомлення. У разі отримання інших повідомлень чатбот виконує аналіз тексту та формує відповідь залежно від змісту запиту.

Для надсилення повідомлень використовується окрема функція `sendMessage()`, яка взаємодіє із Telegram Bot API через HTTP-запити.

Основна логіка чатбота включає:

- обробку команд користувача;
- генерацію відповідей;
- взаємодію з AI-моделлю;
- обробку помилок;
- інтеграцію з Google Sheets;
- логування повідомлень.

Перевагою використання Google Apps Script є швидке розгортання системи та автоматична підтримка HTTPS-з'єднання. Це дозволяє значно спростити процес розробки та тестування чатбота.

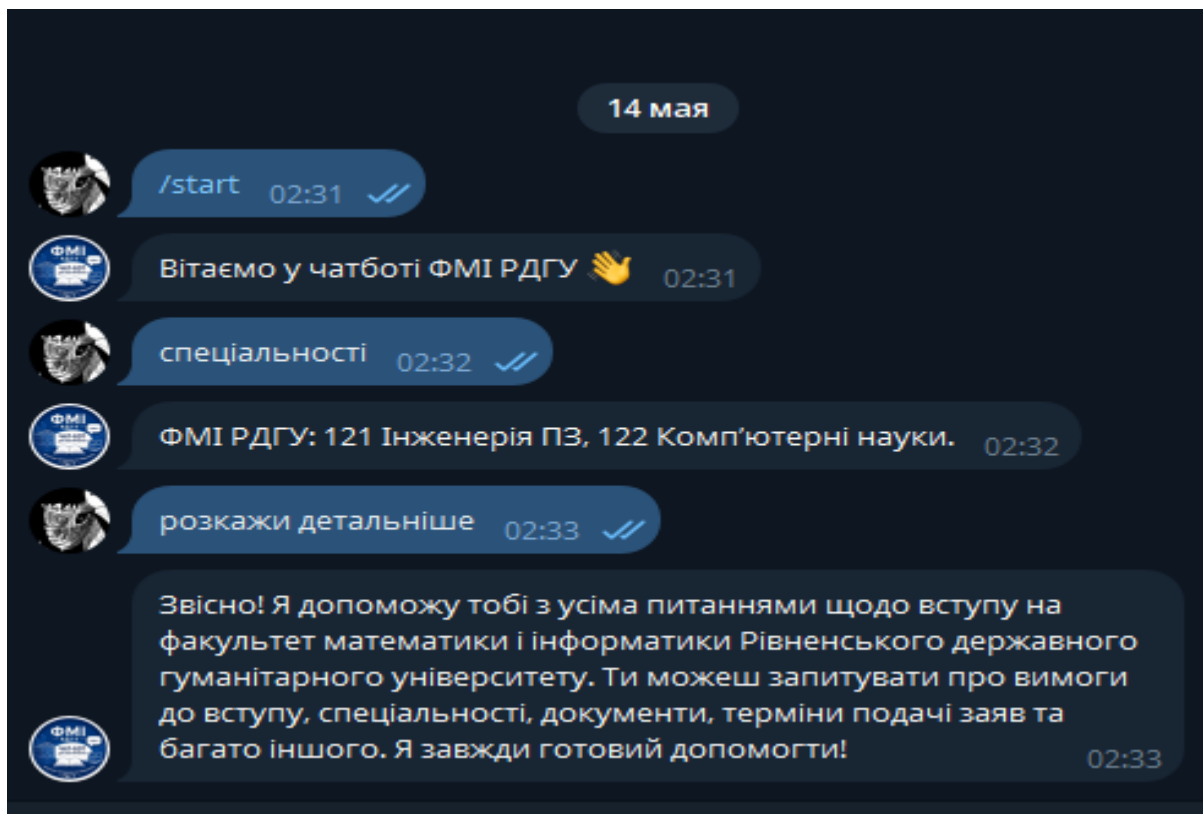


Рисунок 3.1 – Приклад взаємодії користувача з чатботом

Таким чином, реалізація функціоналу чатбота забезпечує стабільну обробку повідомлень та підтримку основних можливостей інформаційної системи.

3.4 Інтеграція з AI-моделлю

Однією з ключових особливостей створеного чатбота є інтеграція із AI-моделлю для генерації відповідей користувачам. Використання технологій штучного інтелекту дозволяє зробити взаємодію із системою більш гнучкою та природною.

На відміну від звичайних сценарних чатботів, які працюють лише за заздалегідь визначеними правилами, AI-чатбот здатний аналізувати текст повідомлення користувача та формувати відповідь залежно від змісту запиту.

AI-модель використовується для:

- аналізу текстових повідомлень;
- розпізнавання намірів користувача;
- генерації природних відповідей;
- підтримки діалогу у довільній формі;
- покращення якості взаємодії із системою.

Під час роботи чатбот надсилає текст повідомлення користувача до AI-сервісу через API-запит. Після обробки запиту AI-модель формує відповідь та повертає її серверній частині системи. Далі відповідь передається користувачу через Telegram API.

Використання AI дозволяє:

- покращити якість відповідей;
- зменшити кількість жорстко заданих команд;
- забезпечити більш природне спілкування;
- автоматизувати обробку складних запитів;
- підвищити зручність використання системи.

Ще однією перевагою є можливість подальшого вдосконалення системи. У майбутньому AI-модель може бути навчена на додаткових даних факультету, що дозволить підвищити точність відповідей та розширити функціональні можливості чатбота.

Інтеграція AI також демонструє можливості поєднання технологій штучного інтелекту та хмарних обчислень у сучасних інформаційних системах.

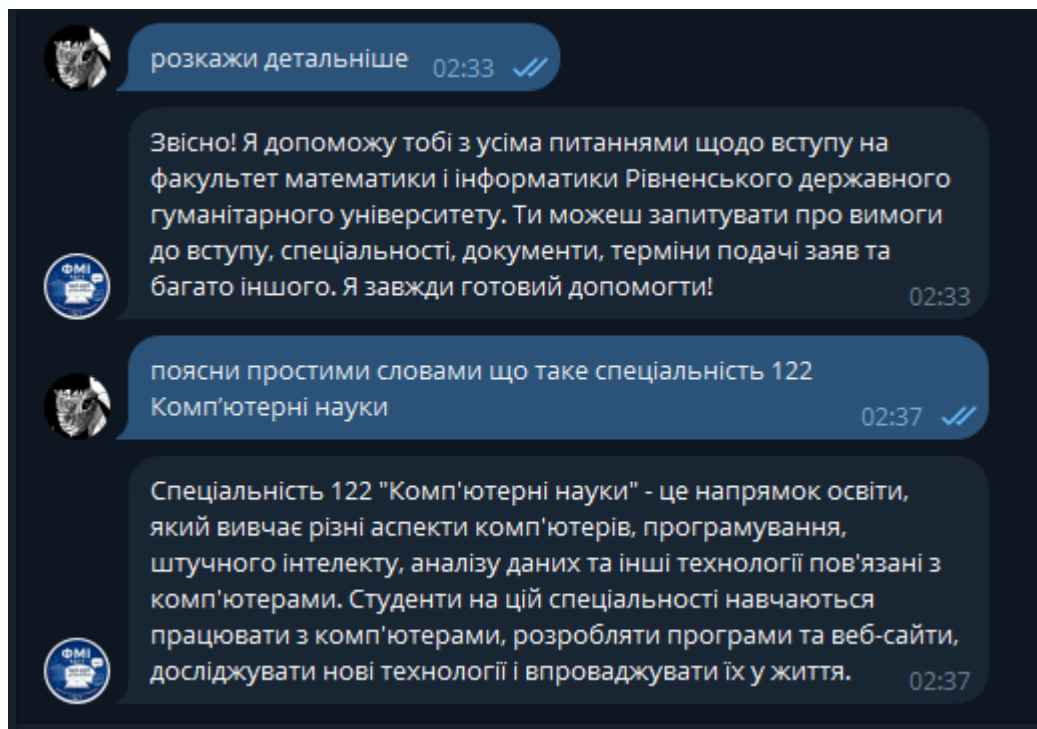


Рисунок 3.2 – Генерація відповідей за допомогою AI-моделі

Таким чином, використання AI-моделі значно покращує функціональність чатбота та забезпечує більш сучасний рівень взаємодії із користувачами.

3.5 Тестування системи

Після завершення розробки чатбота було проведено тестування системи для перевірки коректності роботи функціоналу, стабільності обробки повідомлень та правильності інтеграції із зовнішніми сервісами.

Тестування є важливим етапом розробки програмного забезпечення, оскільки дозволяє виявити помилки, оцінити продуктивність системи та перевірити відповідність функціоналу поставленим вимогам.

Під час тестування було перевірено:

- коректність обробки команд;
- швидкість відповіді чатбота;
- стабільність роботи Webhooks;
- правильність інтеграції з Telegram API;
- роботу AI-моделі;
- обробку помилкових запитів;
- стабільність роботи у хмарному середовищі.

Особлива увага приділялася перевірці швидкості роботи системи. Завдяки використанню Webhooks та Google Apps Script чатбот забезпечував швидке отримання та обробку повідомлень.

Також було протестовано роботу AI-функціоналу. Система успішно аналізувала текстові повідомлення користувачів та формувала логічні відповіді залежно від змісту запиту.

Під час тестування Webhooks було перевірено правильність надсилання HTTP-запитів між Telegram API та серверною частиною чатбота. Усі повідомлення успішно передавалися та оброблялися без значних затримок.

Крім того, тестування підтвердило стабільну роботу системи у хмарному середовищі Google Apps Script. Чатбот коректно працював із декількома користувачами одночасно та не потребував ручного адміністрування серверної інфраструктури.

Результати тестування показали:

- стабільну роботу системи;
- коректну обробку запитів;
- успішну інтеграцію з Telegram API;

- ефективну роботу AI-моделі;
- швидку реакцію чатбота на повідомлення користувачів.

Таким чином, проведене тестування підтвердило працездатність та ефективність розробленої інформаційної системи для підтримки вступників ФМІ РДГУ.

ВИСНОВКИ

У результаті виконання курсової роботи було проведено комплексний аналіз сучасних технологій створення чатботів та досліджено можливості використання штучного інтелекту у сфері освіти. У процесі роботи було розглянуто особливості автоматизації інформаційної підтримки вступників закладів вищої освіти та визначено актуальність використання чатботів для покращення комунікації між факультетом та абітурієнтами.

Під час дослідження було проаналізовано сучасні платформи та сервіси для створення чатботів, а також виконано порівняння існуючих рішень. Особливу увагу приділено Telegram Bot API як одному з найбільш зручних та ефективних інструментів для реалізації інформаційних систем такого типу. У роботі також було досліджено можливості використання хмарних технологій, зокрема середовища Google Apps Script, що забезпечує serverless-архітектуру, автоматичне HTTPS-з'єднання та швидке розгортання застосунку.

У ході виконання роботи було проаналізовано принципи використання форматів обміну даними JSON, механізмів Webhooks та методів інтеграції AI-моделей із Telegram-ботом. На основі проведеного аналізу було обґрунтовано вибір технологічного стеку для реалізації чатбота для вступників факультету математики та інформатики РДГУ.

У практичній частині роботи було спроектовано архітектуру інформаційної системи та реалізовано програмний прототип чатбота. Розроблена система побудована на основі взаємодії Telegram Bot API, Google Apps Script та Google Sheets. Така архітектура дозволила

забезпечити стабільну роботу чатбота у хмарному середовищі та підтримку взаємодії з користувачами у режимі реального часу.

Створений чатбот забезпечує:

- автоматичне надання інформації вступникам;
- швидку обробку запитів користувачів;
- підтримку взаємодії через месенджер Telegram;
- використання технологій штучного інтелекту;
- підтримку текстового діалогу;
- можливість масштабування функціоналу системи.

У роботі було реалізовано функціонал обробки повідомлень користувачів, генерації відповідей та інтеграції з AI-моделлю. Використання технологій штучного інтелекту дозволило зробити спілкування із системою більш природним та гнучким у порівнянні зі звичайними сценарними чатботами.

Окрему увагу було приділено питанням безпеки та захисту даних. Для забезпечення безпечної роботи системи використовувалися HTTPS-з'єднання, механізми перевірки запитів та приховування API-токенів.

Під час тестування було перевірено коректність обробки команд, швидкість відповіді чатбота, стабільність роботи Webhooks та правильність інтеграції із Telegram API та AI-сервісами. Результати тестування підтвердили працездатність системи, стабільну обробку запитів користувачів та можливість практичного використання чатбота у діяльності факультету.

Практичне значення роботи полягає у створенні готового програмного прототипу, який може бути використаний для автоматизації інформаційної підтримки вступників ФМІ РДГУ. Використання такої системи дозволяє зменшити навантаження на працівників приймальної комісії та забезпечити швидке отримання актуальної інформації абітурієнтами.

Подальший розвиток проєкту може включати:

- інтеграцію з базою даних університету;
- додавання голосових повідомлень;
- підтримку багатомовності;
- розширення AI-функціоналу;

- інтеграцію з вебсайтом університету;
- автоматичне оновлення інформації про вступну кампанію;
- використання більш сучасних AI-моделей.

Таким чином, поставлена мета роботи була досягнута, а розроблений чатбот продемонстрував ефективність використання технологій штучного інтелекту та хмарних обчислень для автоматизації інформаційної підтримки вступників закладу вищої освіти.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Telegram Bot API Documentation.
2. Google Apps Script Documentation.
3. JavaScript: The Definitive Guide. David Flanagan.
4. Python Documentation.
5. Google Sheets API Documentation.
6. ECMAScript 2023 Specification.
7. JSON Syntax Documentation.
8. OpenAI API Documentation.

.

Посилання на чатбота

https://t.me/fmi_rdgu_cloud1_bot

@fmi_rdgu_cloud1_bot

ДОДАТОК А

Приклад програмного коду чатбота

```
const TOKEN = "8763902175:AAH-  
z0Em3Dxg95H0fNuULZqkHxvMrRHWkCg";  
  
const OPENROUTER_KEY = "sk-or-v1-  
ca76e601dd14f9c7dd909a568e5829e146dc98ae95645281c465e9b31889ff05";  
  
function doGet() {  
  
    return ContentService.createTextOutput("OK");  
  
}  
  
function doPost(e) {  
  
    try {  
  
        const data = JSON.parse(e.postData.contents);  
  
        const chatId = data.message.chat.id;  
  
        const text = (data.message.text || "").trim();  
  
        const answer = handleMessage(text);  
  
        sendMessage(chatId, answer);  
  
    } catch (err) {
```

```
    Logger.log("ERROR: " + err);

}

}

// □ логіка бота

function handleMessage(text) {

    const t = text.toLowerCase();

    if (t === "/start") {

        return "Вітаємо у чатботі ФМІ РДГУ □";

    }

    if (t.includes("спеціальност")) {

        return "ФМІ РДГУ: 121 Інженерія ПЗ, 122 Комп'ютерні науки.";

    }

    if (t.includes("вступ")) {

        return "Для вступу потрібно подати документи через електронний кабінет.";

    }

}
```

```

}

return getOpenRouterAI(text);
}

// □ OPENROUTER AI

function getOpenRouterAI(text) {

    const url = "https://openrouter.ai/api/v1/chat/completions";

    const payload = {

        model: "openai/gpt-3.5-turbo", // можна змінити на інші моделі
        OpenRouter

        messages: [

            {

                role: "system",

                content: "Ти корисний чатбот для вступників ФМІ РДГУ. Відповідай
коротко і зрозуміло."

            },

            {

```

```
    role: "user",  
  
    content: text  
  }  
],  
  
  temperature: 0.7  
};  
  
const options = {  
  method: "post",  
  contentType: "application/json",  
  headers: {  
    Authorization: "Bearer " + OPENROUTER_KEY,  
  
    // □ ОБОБ'ЯЗКОВО для OpenRouter  
    "HTTP-Referer": "https://script.google.com",  
    "X-Title": "Telegram Bot FMI"  
  },  
  payload: JSON.stringify(payload),  
  muteHttpExceptions: true  
};
```

```
try {

    const response = UrlFetchApp.fetch(url, options);

    const code = response.getResponseCode();

    const body = response.getContentText();

    Logger.log("OpenRouter code: " + code);

    Logger.log("OpenRouter body: " + body);

    if (code !== 200) {

        return "❑ OpenRouter помилка (" + code + ")";

    }

    const json = JSON.parse(body);

    return json.choices?.[0]?.message?.content

        || "AI не зміг сформувати відповідь.";

} catch (e) {

    Logger.log(e);

    return "❑ AI тимчасово недоступний";

}
```

```

    }
}

// □ Telegram send

function sendMessage(chatId, text) {

    const url = "https://api.telegram.org/bot" + TOKEN + "/sendMessage";

    UrlFetchApp.fetch(url, {
        method: "post",
        contentType: "application/json",
        payload: JSON.stringify({
            chat_id: chatId,
            text: text
        })
    });
}

```

