

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет інформаційних технологій
Кафедра прикладних інформаційних систем

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
БАКАЛАВРА
НА ТЕМУ**

«Чат-бот для надання навчальних консультацій»

Галузь знань **12 «Інформаційні технології»**

Спеціальність **122 «Комп'ютерні науки»**

Освітня програма «**Прикладне програмування**»

Освітній рівень: бакалавр

Виконав: студент 4 курсу, групи ПП-41

Грищенко Б.А.

(прізвище та ініціали)

Керівник Ващіліна О.В.

(прізвище та ініціали)

к.ф.-м.н., доц.

(науковий ступінь, звання)

Унікальність тексту – 91%

Випускна кваліфікаційна робота бакалавра допущена до захисту
рішенням кафедри *прикладних інформаційних систем*
Протокол № 14 від 23.05.2023 р.

зав. кафедри  Плескач В.Л.

Київ – 2023

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ БАКАЛАВРА

Номер	Назва етапів кваліфікаційної роботи бакалавра	Термін виконання етапів кваліфікаційної роботи бакалавра	Відмітка про виконання
1.	Вибір теми та наукового керівника кваліфікаційної роботи бакалавра	14.10.2022	виконано
2.	Видача завдання кваліфікаційної роботи бакалавра	24.10.2022	заява
3.	Настановча групова співбесіда з питань кваліфікаційної роботи бакалавра	31.10.2022	виконано
4.	Затвердження плану кваліфікаційної роботи бакалавра	01.11.2022	виконано
5.	Підбір та вивчення літературних та інших джерел з теми дослідження	08.11.2022	виконано
6.	Підготовка і подання науковому керівнику першого варіанту I розділу роботи	21.12.2022	виконано
7.	Підготовка і подання науковому керівнику першого варіанту II розділу роботи	31.01.2023	виконано
8.	Підготовка і подання науковому керівнику першого варіанту III розділу роботи	30.03.2023	виконано
9.	Подання роботи у першому варіанті	28.04.2023	виконано
10.	Оформлення пояснювальної записки кваліфікаційної роботи бакалавра	03.05.2023	виконано
11.	Подання кваліфікаційної роботи бакалавра на попередній захист	22.05.2023	виконано
12.	Врахування зауважень керівника і подання роботи в остаточному варіанті (з відповідним висновком про допуск) на кафедрі	26.05.2023	виконано
13.	Затвердження роботи в цілому (підготовка письмового відгуку керівника, письмова рецензія на бакалаврську роботу)	12.06.2023	виконано
14.	Захист кваліфікаційної роботи бакалавра	26.06.2023	виконано

Здобувач вищої освіти _____



(підпис)

Керівник _____



(підпис)

ВІДОМІСТЬ КВАЛІФІКАЦІЙНОЇ РОБОТИ БАКАЛАВРА

Складові частини дипломної роботи	Обсяг, арк.
Титульний аркуш	1
Календарний план дипломної роботи	1
Відомість дипломної роботи	1
Анотація	1
Анотація (іноземною мовою-англійською)	1
Зміст	1
Перелік скорочень, умовних позначень, термінів	1
Вступ	1
РОЗДІЛ 1	8
РОЗДІЛ 2	10
РОЗДІЛ 3	5
Висновки	2
Перелік використаних джерел	3
Додатки	3

	ПІБ	Підп.	Дата	ДП ХХХХ 00.000.00		
Розробн.	Грищенко Б.А.		26.06.2023	Відомість дипломної роботи добре (85)	Лист	Листів
Керівн.	Ваціліна О.В.		26.06.2023			52
Н/контр.	Макаренко С.А.		26.06.2023			
Зав.каф.	Плескач В.Л.		26.06.2023			

АНОТАЦІЯ

Дипломна робота: 52 с., 5 рис., 1 табл., 36 джерел, 1 дод.

Ця дипломна робота присвячена проектуванню та розробленню чат-бота для надання навчальних консультацій.

Метою дипломної роботи є організація ефективного надання навчальних консультацій на основі телеграм-боту

Для досягнення поставленої мети треба вирішити такі **завдання**:

- Дослідити теоретичні аспекти чат-ботів та їх роль у навчальному процесі.
- Аналізувати сучасні технології та алгоритми, які використовуються для реалізації чат-ботів.
- Розробити та реалізувати програмний код чат-бота з використанням мови програмування Python та фреймворку Telegram Bot API.
- Провести тестування та оцінку ефективності розробленого чат-бота за допомогою показників, таких як час відповіді, задоволення користувачів та якість наданої інформації.

Об'єкт дослідження.

Процеси надання навчальних консультацій з використанням чат-боту

Предмет дослідження.

Технічні, організаційні засади, принципи, концептуальні підходи до побудови чат-бота для надання навчальних консультацій.

Методи дослідження.

Аналіз літературних джерел, вивчення сучасних технологій та практик розробки чат-ботів, розробка та реалізація програмного коду, тестування та оцінка ефективності.

Ключові слова: чат-бот, Python, Telegram.

ANNOTATION

Thesis: 52 pages, 5 figures, 1 tables, 36 references, 3 appendices.

This thesis is dedicated to the design and development of a chatbot for providing educational consultations.

The purpose of this thesis is to develop software that automates the process of educational consultations through the use of a chatbot.

To achieve this goal you need to solve the following **tasks**:

- Investigate the theoretical aspects of chatbots and their role in the educational process.
- Analyze modern technologies and algorithms used for chatbot implementation.
- Design and implement the chatbot's software code using the Python programming language and the Telegram Bot API framework.
- Conduct testing and evaluate the effectiveness of the developed chatbot using metrics such as response time, user satisfaction, and information quality.

Object of study:

The process of providing educational consultations.

Subject of study:

Software-technical foundations and principles of developing chatbots for educational consultations.

Research Methods:

Analysis of literary sources, exploration of modern technologies and development practices for chatbots, software code development and implementation, testing and evaluation of effectiveness.

Keywords: chatbot, Python, Telegram.

Зміст

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ

ВСТУП

РОЗДІЛ 1 ТЕОРЕТИЧНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТЕМИ ДОСЛІДЖЕННЯ

- 1.1 Загальні тенденції у сфері освітніх консультацій
- 1.2 Електронна пошта та форуми як засоби отримання навчальних консультацій
- 1.3. Виклики та обмеження традиційних методів

РОЗДІЛ 2 АНАЛІЗ ПРОГРАМНО-ТЕХНІЧНИХ ЗАСОБІВ ПОБУДОВИ ЧАТ-БОТІВ

- 2.1. Чат-боти: визначення та принцип роботи
- 2.2 Технологічні аспекти роботи чат-ботів
- 2.3. Технологічні аспекти розробки чат-боту
- 2.4 Популярні бібліотеки для створення чат-ботів

РОЗДІЛ 3 ПРОЕКТУВАННЯ, РОЗРОБКА ТА ВПРОВАДЖЕННЯ ЧАТ-БОТУ ДЛЯ НАДАННЯ НАВЧАЛЬНИХ КОНСУЛЬТАЦІЙ

- 3.1 Алгоритм надання навчальних консультацій чат-ботом
- 3.2 Використання Chat GPT API для збільшення гнучкості системи
- 3.3 Інструкція для користування ботом “Your Assistant”

ВИСНОВОК

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТОК

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ

NLP - Natural Language Processing

ML - Machine Learning

БД - База даних

URL - Uniform Resource Locator

ВСТУП

Чат-бот (або просто бот) - це програмне забезпечення, призначене для автоматизованої взаємодії з користувачем через чат-інтерфейс. Чат-боти можуть використовуватися для вирішення різноманітних завдань, включаючи відповіді на запитання, надання інформації, виконання операцій та інше.

Чат-боти можуть бути запрограмовані для розуміння текстових повідомлень користувачів та надсилання відповідей на основі попередньо заданих правил та алгоритмів, або вони можуть використовувати штучний інтелект та машинне навчання для вирішення складних завдань та взаємодії з користувачем більш природним способом.

Чат-боти можуть бути використані в різних сферах, включаючи бізнес, комунікацію з клієнтами, здоров'я, освіту, розваги та багато іншого. Вони можуть бути доступні через платформи миттєвих повідомлень, веб-сайти, мобільні додатки та інші інтерфейси.

Актуальність теми цієї бакалаврської роботи зумовлено зростанням популярності онлайн-навчання, необхідності швидко надавати консультації індивідуальним підходом для кожного студента та загальним покращенням навчального процесу.

Мета бакалаврської роботи - організація ефективного надання навчальних консультацій на основі телеграм-боту.

Об'єктом дослідження бакалаврської роботи є процеси надання навчальних консультацій з використанням чат-боту.

Предметом дослідження бакалаврської роботи є технічні, організаційні засади, принципи, концептуальні підходи до побудови чат-бота для надання навчальних консультацій.

У процесі виконання бакалаврської роботи за основу було взято мову програмування Python, необхідні бібліотеки для цієї мови програмування та їх документацію.

Дипломна робота складається зі вступу, трьох розділів, висновку і списку використаних джерел.

РОЗДІЛ 1 ТЕОРЕТИЧНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТЕМИ ДОСЛІДЖЕННЯ

1.1 Загальні тенденції у сфері освітніх консультацій

За останні роки зростає попит на навчальні консультації. Це пов'язано зі збільшеним доступом до освіти завдяки Інтернету та технологіям. Освітні платформи, курси онлайн та електронні ресурси надають більше можливостей для навчання, але часто вимагають додаткової підтримки та консультацій. Індивідуалізація навчання та розширення вибору також створюють потребу у персоналізованій підтримці. Складні предмети та конкурентне навчання також вимагають додаткових пояснень та підтримки. Розвиток технологій, таких як штучний інтелект, сприяє створенню нових інструментів для навчання та автоматизації консультацій. Все це призводить до зростання запитів на навчальні консультації та потреби в ефективних засобах отримання консультацій для студентів та навчальних закладів.

У сучасному світі, де технології швидко розвиваються, використання інформаційних систем стає все більш поширеним у різних сферах діяльності. Освітній сектор також не залишається осторонь цих змін і активно впроваджує нові технології для поліпшення процесу навчання та консультацій студентів. Одним з таких інструментів є чат-бот, який може стати ефективним засобом надання навчальних консультацій.

Переваги навчальних консультацій для студентів можуть бути різноманітними і впливати на їх академічний успіх та особистий розвиток. Ось деякі з них:

- **Індивідуальний підхід:** Індивідуальний підхід до студентів є надзвичайно важливим і має значний вплив на їхню академічну успішність, мотивацію та загальний розвиток. Цей підхід дозволяє врахувати унікальні потреби, інтереси та рівень вмінь кожного студента. Викладач може створити навчальну програму, яка оптимально відповідає їхнім потребам. Індивідуальний підхід допомагає розвивати конкретні навички і здібності

кожного студента, а також забезпечує додаткову мотивацію до навчання. Він сприяє розширенню можливостей студентів і допомагає їм розвиватися відповідно до їхніх унікальних потреб. Індивідуальний підхід також сприяє побудові глибших відносин між викладачем і студентом, створюючи довіру та позитивне навчальне середовище. В цілому, індивідуальний підхід допомагає забезпечити ефективніше навчання, розвиток студентів та досягнення їхніх особистих цілей.

- Розширення розуміння матеріалу - розуміння матеріалу є надзвичайно важливим для студента, оскільки воно є основою успішного навчання і освоєння нових знань. Якщо студент не розуміє матеріалу, важко очікувати, що він зможе використовувати його в практичних ситуаціях або виконувати завдання на високому рівні. Розуміння матеріалу дозволяє студентам засвоювати нові концепції, взаємозв'язки та принципи, розвивати навички аналізу, синтезу та критичного мислення. Це сприяє їхній впевненості в собі і мотивації до навчання, оскільки успіх у засвоєнні матеріалу надихає студентів на подальші досягнення. Важливо забезпечувати чіткість, доступність та активну підтримку студентів у процесі навчання для розвитку їхнього розуміння матеріалу.

- Вирішення проблем та усунення помилок - важливо швидко усувати проблеми та помилки під час навчання, оскільки це сприяє ефективному засвоєнню матеріалу і забезпечує поступове покращення навичок студента. Якщо проблеми та помилки не вирішуються швидко, вони можуть накопичуватись і ставати перешкодою для подальшого розвитку. Швидке усунення проблем дозволяє студентам зберегти мотивацію та впевненість у своїх можливостях, а також позитивно впливає на їхню самооцінку. Крім того, швидке реагування на помилки допомагає попереджувати накопичення неправильних навичок та неправильних уявлень, що можуть виникнути при невірному розумінні матеріалу. Це сприяє більш ефективному та стабільному навчанню студентів.

- Мотивація та підтримка - мотивація та підтримка є надзвичайно важливими елементами для студентів у процесі навчання. Мотивація є внутрішньою силою, яка підтримує і стимулює студента до досягнення навчальних цілей. Вона допомагає студентам зосередитись, зберегти зацікавленість та наполегливість у засвоєнні нових знань та навичок.

Підтримка також відіграє важливу роль у розвитку студентів. Вона може бути як академічною, так і емоційною. Академічна підтримка включає наявність доступних ресурсів, допомогу вирішенні навчальних питань та відповіді на запитання. Емоційна підтримка передбачає наявність позитивної атмосфери, підтримку від викладачів та одногрупників, а також можливість обговорювати труднощі та виклики, з якими стикаються студенти.

Мотивація та підтримка сприяють успішному навчанню та досягненню цілей студентами. Вони допомагають студентам подолати виклики, зберегти позитивний настрій та переконання у своїх здібностях. Крім того, мотивація та підтримка сприяють формуванню самодисципліни, самоорганізації та відповідальності студентів за своє навчання. Вони створюють сприятливу навчальну атмосферу, де студенти відчують себе підтриманими та мають можливість розвиватися на повну потужність.

- Розвиток навичок самостійної роботи - Навички самостійної роботи є надзвичайно важливими для студентів. Вони включають у себе здатність ефективно організовувати свій час, самостійно поставити й виконати навчальні цілі, здійснювати пошук і аналіз інформації, розвивати критичне мислення та самоконтроль.

Навички самостійної роботи дозволяють студентам бути активними учасниками навчального процесу, незалежно вчитися і розвиватися. Вони дають можливість глибше засвоювати матеріал, здобувати нові знання та навички навіть поза межами університету чи навчального закладу.

Вміння самостійно працювати допомагають студентам стати більш організованими, самодисциплінованими та впевненими в своїх можливостях.

Вони здатні самостійно шукати джерела інформації, вивчати нові теми, готуватися до занять, виконувати домашні завдання та проекти.

Крім того, навички самостійної роботи мають велике значення для подальшого кар'єрного розвитку студентів. Роботодавці шукають фахівців, які можуть працювати незалежно, приймати рішення, вирішувати проблеми та бути самомотивованими. Студенти з розвинутими навичками самостійної роботи мають перевагу на ринку праці, оскільки вони можуть швидше адаптуватися до нових умов і вимог.

Тому, вивчення та розвиток навичок самостійної роботи є важливою складовою навчання студентів. Вони допомагають створити основу для успіху, самореалізації та досягнення поставлених цілей як у навчанні, так і в майбутній професійній діяльності.

Загалом, навчальні консультації допомагають студентам отримати індивідуальну увагу, підтримку та додаткові ресурси для покращення їхнього навчання. Це сприяє зростанню їхнього розуміння, розвитку навичок та досягненню академічних цілей.

1.2 Електронна пошта та форуми як засоби отримання навчальних консультацій

Електронна пошта та форуми є двома популярними засобами комунікації, які використовуються в освітньому середовищі для спілкування між викладачами та студентами. Кожен з цих засобів має свої переваги та може використовуватися в різних контекстах.

Електронна пошта є одним з основних засобів комунікації в освітніх установах. Електронна пошта є досить простим та зручним засобом комунікації. Вона дозволяє студентам та викладачам обмінюватися повідомленнями, додавати вкладення (такі як документи, презентації тощо) і зберігати записи розмов. Вона дозволяє викладачам та студентам спілкуватися безпосередньо один з одним. Це дозволяє налагодити персоналізоване спілкування, де можна звертатися до кожної особи окремо, враховуючи її потреби та індивідуальність. Важливим є можливість зберігати всю історію спілкування між викладачами та студентами. Це корисно для подальшого посилення на важливу інформацію, таку як питання, відповіді, завдання тощо. Загалом студенти та викладачі можуть комунікувати у зручний для них час. Вони можуть відповідати на повідомлення в зручний для них момент, що сприяє більш ефективній комунікації.

Форуми є онлайн-платформами, де користувачі можуть обговорювати теми, задавати питання та ділитися інформацією. Вони створюють сприятливе середовище для спілкування та обміну ідеями між студентами та викладачами. Учасники можуть створити віртуальну спільноту, де можна навчатися один від одного, ділитися своїм досвідом та розвивати критичне мислення. Форуми надають можливість студентам виразити свою думку та погляди на певну тему. Вони можуть задавати запитання, коментувати думки інших учасників та сприяти глибшому розумінню навчального матеріалу. Перевагою є можливість студентів та викладачів брати участь в обговореннях у зручний для них час. Вони можуть задавати питання, отримувати відповіді та приєднуватися до обговорень, коли це найбільш зручно для них. Важливим є можливість

зберігати всю інформацію, яка була опублікована на платформі. Це дозволяє студентам повертатися до обговорень, шукати відповіді на свої питання та матеріалів, які були розміщені раніше.

Комбінація електронної пошти та форумів може створити ефективну комунікаційну платформу в освітньому середовищі. Використовуючи ці засоби, викладачі та студенти можуть спілкуватися, обмінюватися інформацією та надавати підтримку один одному, сприяючи більш ефективному навчанню та зростанню знань.

Існує кілька популярних електронних поштових служб та форумів, які можна використовувати для отримання навчальних консультацій. Ось кілька з них:

1. Електронні поштові сервіси:

- Gmail
- Outlook
- Yahoo Mail

2. Форуми та спільноти:

Stack Overflow є одним з найбільш популярних веб-сайтів для програмістів та розробників програмного забезпечення. Це онлайн-спільнота, де програмісти можуть задавати питання, отримувати відповіді та обговорювати проблеми, пов'язані з програмуванням та розробкою програмного забезпечення.

Stack Overflow дозволяє користувачам створювати облікові записи і задавати питання у форматі форуму. Інші користувачі можуть відповідати на ці питання, надаючи свої власні рішення та поради. Відповіді оцінюються спільнотою, і найкращі відповіді отримують більше голосів та вищий рейтинг.

Основною метою Stack Overflow є створення бібліотеки знань, де програмісти можуть знайти відповіді на свої питання та проблеми. Велика кількість питань та відповідей, які вже присутні на Stack Overflow, робить його важливим ресурсом для пошуку рішень на різних етапах програмування.

Stack Overflow також має строгі правила та стандарти, що стосуються відповідей і коментарів користувачів. Це допомагає забезпечити якість вмісту та зберегти цінність ресурсу як надійного джерела інформації для програмістів.

Узагалі, Stack Overflow є незамінним інструментом для програмістів, де вони можуть спілкуватися, ділитися знаннями та отримувати допомогу вирішення проблем, з якими вони зіштовхуються під час розробки програмного забезпечення.

Quora є популярною онлайн-платформою для обміну знаннями, де користувачі можуть задавати питання та отримувати відповіді від інших користувачів, які мають експертні знання або досвід у відповідних галузях.

На відміну від Stack Overflow, який спеціалізується на програмуванні та технічних питаннях, Quora охоплює більш широкий спектр тематик. Користувачі можуть задавати питання з різних областей, таких як наука, технології, мистецтво, культура, бізнес, освіта, психологія та багато інших.

Quora дозволяє користувачам створювати облікові записи та відповідати на питання, що цікавлять їх. Відповіді можуть бути надані як індивідуальними користувачами, так і фахівцями з відповідної галузі, які мають впевненість та досвід відповідно до заданого питання.

Одна з унікальних рис Quora полягає у можливості створення профілів користувачів, де вони можуть вказувати свої професійні навички, досвід та освіту. Це допомагає іншим користувачам оцінити вірогідність та якість відповідей.

Quora також надає можливість користувачам створювати тематичні групи та обговорювати конкретні теми. Це сприяє формуванню спільноти експертів та зацікавлених осіб, які можуть активно обмінюватися знаннями та досвідом.

Загалом, Quora є цікавою та корисною платформою для отримання відповідей на різноманітні питання, спілкування з експертами та ділових контактів, а також для обміну знаннями та поглиблення розуміння різних тематик.

Reddit є соціальною мережею та веб-сайтом, який дозволяє користувачам обговорювати, ділитися контентом та взаємодіяти на різноманітні теми. Це одна з найбільш популярних платформ спільнотового веб-сайту, яка охоплює широкий спектр інтересів та тематик.

Reddit працює на основі системи піднесення (upvoting) та пониження (downvoting) вмісту користувачами. Користувачі можуть створювати "пости" (публікації) на різні теми та вступати в дискусії, а також спільно формувати вміст шляхом голосування за публікації, коментарі та інші вмістові елементи.

Особливістю Reddit є наявність великої кількості "субреддітів" (subreddits), що є окремими підфорумами або групами, спеціалізованими на конкретних тематиках. Користувачі можуть підписуватися на ці субреддіти та отримувати актуальні публікації та обговорення на ці теми.

Reddit став популярним серед різноманітних спільнот, включаючи геймерів, технологічних ентузіастів, кінофілів, науковців, фотографів, музикантів, політичних активістів та багатьох інших. Ця платформа надає можливість обговорювати питання, задавати питання, ділитися цікавим вмістом, вступати в дискусії та взаємодіяти з іншими користувачами.

Reddit також відомий своєю відкритістю та анонімністю. Користувачі можуть використовувати псевдоніми та виступати анонімно, що дозволяє їм вільно висловлювати свої думки та обговорювати різні теми без прив'язки до своєї особистості.

Загалом, Reddit є динамічною та різносторонньою платформою, яка сприяє обміну ідеями, знаннями та думками в широкому спектрі тематик, а також формуванню спільнот і взаємодії між користувачами з усього світу.

Таблиця 1.2 — Порівнянн електронної пошти та форуму

Відмінності	Електронна пошта	Форум
Тип комунікації	Одне до одного	Багато до багатьох
Формат повідомлень	Текст, вкладення	Текст, зображення, відео
Приватність	Приватні розмови	Публічні обговорення
Синхронність	Асинхронна	Асинхронна
Архівація	Зберігається в особистому поштовому ящику	Інформація доступна для всіх учасників
Форматування повідомлення	Простий текст, HTML формат, вкладення	Більш гнучкий, можливість додавати різні медіа елементи

Ці інструменти надають можливість звернутися до широкої спільноти користувачів і отримати різноманітні відповіді та консультації щодо навчання та інших тем. Завжди пам'ятайте, що відповіді, отримані на цих платформах, можуть бути надані різними людьми, і рекомендується перевіряти достовірність і релевантність інформації, яку ви отримуєте.

1.3. Виклики та обмеження традиційних методів

Часові обмеження та доступність викладачів можуть бути важливими факторами у проведенні навчальних консультацій. Оскільки викладачі зайняті багатьма обов'язками, їхній час може бути обмеженим. Однак, існують кілька способів, якими викладачі можуть забезпечити доступність для студентів.

- Заздалегідь заплановані консультації: Викладачі можуть встановлювати певний графік консультацій, коли вони доступні для студентів. Це може включати регулярні консультації під час визначеного раніше дня або години, або навіть заздалегідь заплановані індивідуальні зустрічі. Це дозволяє студентам знати, коли вони можуть звертатися до викладача для отримання консультацій.

- Викладачі можуть використовувати електронні засоби комунікації, такі як електронна пошта або онлайн-платформи, для доступності у позаурочний час. Вони можуть відповідати на електронні повідомлення студентів або приймати запити на консультації через ці канали. Це дозволяє студентам звертатися до викладача з питаннями або проблемами у зручний для них час, навіть якщо викладач не доступний особисто.

- Викладачі можуть використовувати систему запису на консультації, де студенти можуть запланувати свій час для зустрічі з викладачем. Це допомагає організувати час і розподілити його між студентами, забезпечуючи більш ефективне використання часу викладача та забезпечуючи студентам доступ до консультацій у встановлений час.

- Викладачі можуть використовувати віртуальні засоби комунікації, такі як відеоконференції або чат-платформи, для проведення консультацій. Це дозволяє викладачам та студентам зустрітися в онлайн-середовищі, навіть якщо вони знаходяться в різних місцях. Віртуальні консультації можуть бути зручним способом отримати консультацію без необхідності фізичного присутності.

Важливо, щоб викладачі були свідомі та відповідальні щодо своїх обов'язків та доступності для студентів. Це допоможе створити позитивне навчальне середовище та сприяти ефективному навчанню та розвитку студентів.

Загалом завдання та обов'язки викладачів щодо навчальних консультацій можуть бути різноманітними:

- Надання додаткової інформації: Викладачі повинні бути готові надати додаткову інформацію студентам під час консультацій. Це можуть бути пояснення навчального матеріалу, додаткові приклади або ресурси, що допоможуть студентам зрозуміти складні концепції або вирішити проблемні завдання.

- Відповіді на запитання: Одним із основних завдань викладача під час навчальних консультацій є відповідь на запитання студентів. Викладач

повинен бути готовим відповісти на запитання з різних аспектів навчального матеріалу, пояснити складні поняття та надати додаткові роз'яснення.

- **Допомога у вирішенні проблем:** Викладачі повинні бути готові допомогти студентам у вирішенні проблем, з якими вони зіткнулися у процесі навчання. Це може включати вирішення труднощів у розумінні матеріалу, пояснення правильного підходу до вирішення задач або надання рекомендацій з покращення навичок.

- **Стимулювання самостійності:** Викладачі повинні сприяти розвитку навичок самостійної роботи у студентів. Вони можуть допомогти студентам розвинути вміння самостійно аналізувати проблеми, шукати відповіді та розв'язувати завдання. Це може включати спрямування студентів на відповідні ресурси та навчання їх розуміти, як самостійно навчатися.

- **Встановлення чітких очікувань:** Викладачі повинні встановити чіткі очікування стосовно навчальних консультацій, таких як тривалість, доступність та вимоги щодо підготовки. Це допоможе студентам зрозуміти, що вони можуть очікувати від консультацій та як підготуватися до них.

- **Стимулювання активності та співпраці:** Викладачі повинні стимулювати активну участь студентів під час навчальних консультацій. Це може включати постановку запитань, проведення дискусій та сприяння співпраці між студентами.

- **Оцінка та зворотній зв'язок:** Викладачі повинні оцінювати рівень розуміння студентів та надавати їм зворотній зв'язок щодо їхнього прогресу. Це допоможе студентам зрозуміти свої сильні та слабкі сторони та скерувати свої зусилля на покращення.

Ці завдання та обов'язки викладачів під час навчальних консультацій спрямовані на підтримку студентів, сприяння їхньому академічному успіху та розвитку навчальних навичок. Вони допомагають створити сприятливу навчальну середовище та підтримують індивідуальні потреби студентів у процесі навчання.

Ефективна комунікація та розуміння потреб студентів є ключовими аспектами успішних навчальних консультацій. Вчителі повинні мати здатність налагоджувати ефективний зв'язок зі своїми студентами, здатність слухати та розуміти їхні потреби. Викладачі повинні бути уважними до того, що говорять студенти. Це означає надавати їм можливість висловлюватися, задавати питання та показувати зацікавленість у їхніх проблемах чи запитаннях. Активне слухання допомагає викладачам краще розуміти потреби студентів та знаходити ефективні рішення. Необхідно використовувати різні канали комунікації, які відповідають потребам різних студентів. Дехто може відчувати себе комфортніше у візуальному середовищі, тому використання відео або ілюстрацій може бути корисним. Інші можуть віддавати перевагу письмовій комунікації через електронну пошту або форуми. Важливо забезпечити доступ до різних способів комунікації, щоб задовольнити потреби всіх. Викладачі повинні створити атмосферу, в якій учні почуваються комфортно, вони відно висловлюють свої думки та питання. Важливо показати, що їхня думка цінна та важлива. Викладачі повинні бути гнучкими та готовими адаптуватися до потреб студентів. Кожен студент може мати унікальні потреби та способи сприйняття інформації. Важливо забезпечити індивідуальний підхід до кожного, розуміти його особливості та надавати належну підтримку. Не останню роль займає регулярний збір фідбеку від студентів, це є важливим елементом в оцінці ефективності комунікації та розуміння потреб. Викладачі можуть проводити анкетування, співбесіди або інші форми збору думок та пропозицій від студентів. Це допомагає викладачам удосконалювати свою роботу та відповідати потребам студентів.

Ефективна комунікація та розуміння потреб студентів є взаємозалежними процесами. Чим краще викладачі розуміють потреби студентів та здатні ефективно спілкуватися з ними, тим більше можливостей для навчального успіху та задоволеності студентів.

РОЗДІЛ 2 АНАЛІЗ ПРОГРАМНО-ТЕХНІЧНИХ ЗАСОБІВ ПОБУДОВИ ЧАТ-БОТІВ

2.1. Чат-боти: визначення та принцип роботи

Чат-бот - це програмне забезпечення, яке здатне автоматично взаємодіяти з користувачем через текстовий інтерфейс, імітуючи розмову з людиною. Він використовує штучний інтелект та алгоритми обробки мови для розпізнавання запитів та надання відповідей, що релевантні для користувача.

У навчальному процесі чат-боти можуть стати цінними джерелами інформації для студентів. Вони можуть відповідати на запитання про конкретні теми, надавати вказівки, пояснення та допомагати в засвоєнні нового матеріалу. Чат-боти можуть також надавати рекомендації щодо джерел додаткової інформації, книжок або онлайн-ресурсів. Вони можуть створювати тестові завдання та запитання, щоб перевірити знання студентів, можуть оцінювати відповіді та надавати негайну зворотну інформацію стосовно правильності відповідей. Це може бути корисною функцією для самоперевірки та контролю знань. Чат-боти можуть допомагати студентам у виконанні завдань та контрольних робіт. Вони можуть надавати підказки, пропонувати методичні матеріали, відповідати на запитання щодо завдань та допомагати студентам зрозуміти, як краще виконати роботу. Це сприяє активному самостійному навчанню та розвитку аналітичних навичок.

Деякі чат-боти можуть навчатися на основі взаємодії з користувачами, що дозволяє їм надавати персоналізовані рекомендації та матеріали. Вони можуть аналізувати історію запитів студента та рекомендувати додаткові ресурси, підручники, відео або статті, які відповідають їхнім інтересам та потребам. Це дозволяє студентам навчатися в зручному для них темпі та отримувати персоналізовану підтримку. Також бот може включати в себе функції тестування та оцінювання. Він може створювати тести, запитувати студентів питання з різних тем та надавати миттєвий фідбек про правильність відповідей. Це допомагає студентам вправлятися в засвоєнні матеріалу, перевіряти свої

знання та визначати області, які потребують додаткової уваги. Ще одним можливим застосуванням може бути використання елементи гейміфікації, такі як бали або досягнення, для мотивації студентів до активної навчальної діяльності. Вони можуть створювати стимули для самоосвіти, спонукати студентів до виконання завдань, досліджень або участі в додаткових навчальних активностях.

Чат-боти можуть функціонувати як особисті помічники для студентів, які надають поради та рекомендації щодо організації навчання. Вони можуть нагадувати про важливі дати та події, допомагати у плануванні навчального графіка, пропонувати стратегії ефективного вивчення та надавати підтримку у розв'язанні проблем, з якими студенти можуть зіткнутися під час навчання. Чат-боти можуть мати базу даних з відповідями на часті запитання студентів, що допомагає ефективно та швидко розповісти про основні питання, що цікавлять студентів. Вони можуть відповідати на запитання про розклад занять, правила та процедури навчання, доступні ресурси та багато іншого. Крім того можуть також служити підтримкою викладачам, допомагаючи відповідати на запитання студентів, аналізувати та обробляти дані про виконання завдань, надавати статистику про навчальний процес. Вони можуть автоматизувати деякі адміністративні процеси та забезпечувати швидку комунікацію між викладачами та студентами.

Чат-боти можуть бути ефективними інструментами підтримки навчання, полегшуючи доступ до інформації, забезпечуючи інтерактивну комунікацію та надаючи персоналізовану підтримку. Вони допомагають зробити навчальний процес більш ефективним, доступним та зацікавлюючим для студентів.

Важливим елементом при впровадженні та використанні чат-бота в навчальному процесі є зворотній зв'язок від студентів, викладачів та адміністрації. Кожна з цих груп має свої унікальні перспективи та потреби, які можуть бути враховані для поліпшення функціоналу та ефективності чат-бота. Ось як може бути забезпечений зворотній зв'язок з кожною зі сторін:

1. Зворотній зв'язок від студентів:

- Опитування та анкетування: Студентам можуть бути надані опитування або анкети, щоб зібрати їхні враження, думки та пропозиції щодо використання чат-бота. Це може включати оцінку зручності, корисності, якості відповідей та загального задоволення від використання чат-бота.

- Фокус-групи та інтерв'ю: Можна створити фокус-групи або провести інтерв'ю зі студентами, щоб отримати докладнішу інформацію про їхні враження та думки про чат-бота. Це може допомогти зрозуміти їхні потреби, проблеми та пропозиції щодо поліпшення системи.

2. Зворотній зв'язок від викладачів:

- Зустрічі та обговорення: Адміністрація може організовувати зустрічі з викладачами для обговорення їхнього досвіду використання чат-бота та збору відгуків. Викладачі можуть поділитися своїми спостереженнями, проблемами, які виникають у процесі, а також запропонувати ідеї щодо поліпшення та розширення функціоналу чат-бота.

- Зворотній зв'язок у вигляді повідомлень: Викладачі можуть мати можливість надсилати свої пропозиції, питання та проблеми пов'язані з чат-ботом через спеціальний інтерфейс адміністратора чат-бота. Адміністрація може вивчати ці повідомлення та надавати необхідну підтримку та відповіді на запитання.

3. Зворотній зв'язок від адміністрації:

- Моніторинг та аналітика: Адміністрація може здійснювати моніторинг та аналіз використання чат-бота, включаючи кількість взаємодій, типи запитань, час реакції та інші метрики. Це дозволяє виявити тенденції та зрозуміти, як ефективно використовується чат-бот, а також виявити проблемні аспекти, які потребують уваги.

- Зворотній зв'язок з викладачами та студентами: Адміністрація може активно спілкуватися з викладачами та студентами, слухати їхні побажання, пропозиції та проблеми, і намагатися вирішувати їх шляхом вдосконалення чат-бота.

Зворотній зв'язок від студентів, викладачів та адміністрації дозволяє отримати цінні дані та пропозиції для вдосконалення чат-бота та його використання в навчальному процесі. Це сприяє покращенню якості та ефективності навчання, а також задоволенню потреб усіх учасників навчального процесу.

Цілодобова доступність та швидкість відповідей є однією з ключових переваг чат-ботів у навчальних консультаціях. Фактори, які впливають на цілодобову доступність та швидкість відповідей чат-ботів:

- Автоматизована взаємодія - чат-боти забезпечують автоматизовану взаємодію з користувачами, що дозволяє їм працювати цілодобово без прив'язки до реального часу викладача. Користувачі можуть задавати питання та отримувати відповіді в будь-який зручний для них час, що забезпечує недоступність викладача або обмеження часових зон.

- Швидкість обробки запитів - чат-боти працюють на основі комп'ютерних алгоритмів та штучного інтелекту, що дозволяє їм швидко обробляти запити користувачів. Вони можуть миттєво аналізувати текстові запити, розпізнавати інтент та надавати релевантні відповіді без затримок.

- Персоналізованість - чат-боти можуть мати можливість навчання на основі взаємодії з користувачами. Вони можуть аналізувати історію запитів та надавати персоналізовані відповіді, враховуючи індивідуальні потреби та стилі навчання кожного студента. Це дозволяє надати швидку та точну підтримку, що відповідає конкретним потребам кожного студента.

- Автоматичні системи нагадувань - чат-боти можуть мати вбудовані системи нагадувань та сповіщень, які надсилають студентам повідомлення про важливі дати, дедлайни, події та іншу релевантну інформацію. Це допомагає студентам бути в курсі навчального процесу і пам'ятати про важливі події.

- Миттєва доступність - чат-боти можуть бути доступні на різних платформах, таких як веб-сайти, мобільні додатки або месенджери. Це дозволяє студентам мати постійний доступ до чат-бота з будь-якого пристрою з підключенням до Інтернету.

Завдяки цілодобовій доступності та швидкості відповідей, чат-боти забезпечують студентам негайну підтримку та допомогу в будь-який час, що є особливо важливим у навчальному процесі, де можуть виникати термінові запитання та проблеми, які потребують швидкого вирішення.

Враховуючи ці підходи до оцінки, можна отримати зрозуміння ефективності чат-бота в навчальному процесі та визначити його вплив на студентів. Результати оцінки можуть служити основою для подальших поліпшень та розробки чат-ботів, що краще задовольнятимуть потреби студентів і сприятимуть їхньому навчанню.

2.2 Технологічні аспекти роботи чат-ботів

Робота чат-ботів базується на використанні різних технологій та алгоритмів, які дозволяють їм розпізнавати та обробляти запити користувачів, а також надавати відповіді. Ось декілька ключових технологій та алгоритмів, що використовуються в чат-ботах:

- Обробка природної мови (Natural Language Processing, NLP): NLP є ключовою технологією для розпізнавання та розуміння текстових запитів користувачів. Вона включає в себе різні компоненти, такі як токенізація (розбиття тексту на окремі слова або фрази), лематизація (перетворення слова до його базової форми), синтаксичний аналіз (визначення структури речення) та семантичний аналіз (розуміння смислу запиту).

- Машинне навчання (Machine Learning, ML): Машинне навчання використовується для навчання чат-ботів розпізнавати та класифікувати текстові запити користувачів. Це може бути здійснено за допомогою навчання на основі набору даних з позначками, де модель навчається розпізнавати шаблони та зв'язки між вхідними запитом та відповідями.

- Алгоритми розпізнавання намірів (Intent Recognition): Ці алгоритми допомагають визначити намір користувача за його текстовим запитом. Вони допомагають відокремити запити на конкретні дії або питання, щоб відповісти на них правильно та ефективно.

- **Дерева рішень (Decision Trees):** Дерева рішень використовуються для моделювання процесу прийняття рішень в чат-боті. Вони дозволяють створювати логічні правила, які визначають потрібні кроки або відповіді в залежності від характеристик запиту користувача.
- **Алгоритми генерування відповідей (Response Generation):** Ці алгоритми використовуються для генерації відповідей на запити користувачів. Вони можуть бути засновані на правилах, шаблонах або використовувати глибокі нейронні мережі для створення текстових відповідей, які звучать природним чином.
- **Бази знань (Knowledge Bases):** Бази знань використовуються для зберігання інформації та фактів, що використовуються чат-ботами для надання точних відповідей. Вони можуть містити фактичну інформацію, процедури, правила, розклади та іншу релевантну інформацію, яка може бути запитана користувачами.

Ці технології та алгоритми спільно працюють, щоб забезпечити ефективну та розуміну взаємодію з користувачами. Вони дозволяють чат-ботам розуміти запити користувачів, генерувати адекватні відповіді та надавати цінну інформацію для навчання та консультацій.

2.3. Технологічні аспекти розробки чат-боту

При виборі платформи та інструментів для створення чат-боту в навчальному контексті, необхідно враховувати декілька факторів. Ось деякі з них:

1. **Цільова аудиторія:** Розуміння потреб та характеристик цільової аудиторії допоможе визначити, які платформи та інструменти найкраще підходять для їхнього залучення. Наприклад, якщо більшість студентів використовують мобільні пристрої, слід розглянути платформи, які підтримують мобільні додатки.

2. **Функціональні можливості:** Визначення потреб та вимог до функціональності чат-бота допоможе вибрати платформу, яка надає необхідні

інструменти та можливості. Наприклад, якщо потрібна інтеграція з існуючими системами керування навчанням або можливість автоматичної обробки запитань та відповідей, потрібно знайти платформу з відповідними функціями.

3. Технічні вимоги та навички: Враховуйте свої технічні можливості та навички при виборі платформи. Деякі платформи можуть мати складну настройку або вимагати програмування, тоді як інші можуть бути більш інтуїтивно зрозумілими та мають візуальний редактор для створення чат-ботів.

4. Надійність та підтримка: Важливо враховувати надійність платформи та рівень підтримки, яку вони надають. Переконайтеся, що платформа має стабільну роботу, безпекові заходи та широкий спектр документації та підтримки.

5. Вартість: Розгляньте вартість використання платформи або інструменту. Деякі платформи можуть мати безкоштовну версію з обмеженими функціями, але з додатковими платними планами. При оцінці вартості також враховуйте масштаб проекту та бюджет.



Рисунок 2.1 - Лого Python

Python - це інтерпретована, високорівнева мова програмування загального призначення. Вона була створена Гвідо ван Россумом і була випущена вперше у 1991 році. Python має простий і зрозумілий синтаксис, що робить його дуже популярним серед початківців у програмуванні, а також серед професіоналів.

Основні особливості Python:

1. Простий синтаксис: Python має простий і зрозумілий синтаксис, який дозволяє програмістам легко вчитися і розуміти мову. Він нагадує англійську мову, що робить код більш зрозумілим і читабельним.

2. Інтерпретованість: Python є інтерпретованою мовою, що означає, що програми Python виконуються рядок за рядком. Це дозволяє швидше розробляти програми, оскільки немає необхідності компілювати код перед його виконанням.

3. Розширюваність: Python має багато сторонніх бібліотек і модулів, які дозволяють розширити його функціональність. Це дозволяє програмістам використовувати готові рішення і прискорює розробку програм.

4. Крос-платформеність: Python підтримується на багатьох операційних системах, таких як Windows, macOS, Linux, а також має порти на мобільні платформи, такі як Android і iOS.

5. Велика спільнота: Python має велику і активну спільноту програмістів, яка створює багато ресурсів, бібліотек і фреймворків. Це робить Python дуже популярним і зручним для використання.

Python використовується для різних цілей, включаючи веб-розробку, наукові обчислення, аналіз даних, штучний інтелект, автоматизацію задач, розробку ігор та багато іншого.

PyCharm - це інтегроване середовище розробки (IDE) для мови програмування Python, розроблене компанією JetBrains. Воно надає широкий набір інструментів та функціональностей, спрямованих на полегшення розробки, тестування та налагодження програм на Python.

Основні особливості PyCharm включають:

1. Створення проектів: PyCharm дозволяє створювати та керувати проектами Python. Ви можете вибрати тип проекту, налаштувати середовище виконання та управляти залежностями.

2. Редактор коду: PyCharm має потужний редактор коду з функціями автодоповнення, підсвічування синтаксису, перевірки помилок та багато

іншого. Він підтримує також використання інструментів форматування коду для підтримки стилю програмування.

3. Відладка: PyCharm надає вбудовану підтримку для відлагодження програм. Ви можете встановлювати точки зупину, виконувати код крок за кроком, спостерігати змінні та стежити за виконанням програми.

4. Управління залежностями та пакетами: PyCharm допомагає встановлювати, оновлювати та керувати залежностями Python за допомогою інтегрованого менеджера пакетів, такого як pip або Anaconda.

5. Інструменти для тестування: PyCharm надає інтеграцію з різними фреймворками для тестування Python, такими як pytest або unittest. Ви можете запускати тести, аналізувати результати тестування та оцінювати покриття коду тестами.

6. Контроль версій: PyCharm інтегрується з різними системами керування версіями, такими як Git, Mercurial або Subversion, що дозволяє зручно керувати історією розробки вашого проекту.

7. Інструменти аналізу коду: PyCharm надає можливості аналізу коду для виявлення потенційних проблем, неправильних викликів функцій, недоступних атрибутів та інших помилок.

PyCharm доступний як безкоштовна Community Edition, яка має базовий набір функціональності, а також як платна Professional Edition, яка має розширений функціонал для комерційного використання та розробки великих проектів.

Бібліотеки для Python є надзвичайно важливими інструментами для програмістів і дослідників, які працюють з мовою програмування Python. Ось кілька причин, чому бібліотеки є необхідними:

- Розширення функціональності - бібліотеки додають нові функції та можливості до мови Python. Вони надають реалізацію різних алгоритмів, інструментів та рішень для широкого спектру завдань, включаючи обробку даних, машинне навчання, веб-розробку, наукові обчислення, візуалізацію та багато іншого.

- Економія часу та зусиль - використання готових бібліотек дозволяє уникнути необхідності писати код з нуля для кожної задачі. Бібліотеки зазвичай мають оптимізовані і випробувані реалізації, що дозволяє програмістам ефективно використовувати їх і скорочує час розробки.
- Спільнота та підтримка - бібліотеки мають активну спільноту користувачів, яка надає підтримку, документацію, приклади коду та відповіді на питання. Ви можете отримати допомогу, поради та ресурси від інших розробників, які вже використовують ці бібліотеки.
- Швидкість розробки - бібліотеки прискорюють процес розробки, оскільки вони надають готові інструменти та інтерфейси для вирішення конкретних завдань. Вам не потрібно займатися деталями низькорівневої реалізації алгоритмів, оскільки бібліотеки вже це забезпечують.
- Кросплатформеність - бібліотеки зазвичай працюють на різних платформах та операційних системах. Це дозволяє вам розробляти програми з використанням бібліотек на одній платформі та запускати їх на інших без необхідності переписування коду.

Усі ці переваги роблять бібліотеки важливими компонентами в екосистемі Python та допомагають розробникам створювати ефективні та потужні програми.

Використання бібліотек в Python вимагає кількох кроків:

1. Встановлення бібліотеки: Перш за все, потрібно встановити бібліотеку, яку ви хочете використовувати. Це можна зробити за допомогою менеджера пакетів `pip`, який входить у склад стандартного розподілу Python. Наприклад, для встановлення бібліотеки `NumPy` ви можете виконати команду `pip install numpy`.
2. Імпорт бібліотеки: Після встановлення ви повинні імпортувати бібліотеку у свій Python-код. Це дозволяє вам отримати доступ до функцій, класів та інших компонентів бібліотеки. Наприклад, для імпорту бібліотеки `NumPy` ви можете використовувати `import numpy`.

3. Використання функцій та класів бібліотеки: Після імпорту ви можете використовувати функції, класи та інші компоненти бібліотеки у своєму кодї. Це означає виклик функцій, створення об'єктів класів та виконання операцій, які бібліотека надає. Наприклад, якщо у вас є бібліотека NumPy, ви можете викликати функції для роботи з масивами або математичними операціями.

4. Конфігурація та налаштування: Більшість бібліотек мають параметри та налаштування, які можна змінювати для досягнення певного результату або пристосування до вашого використання. Вам може знадобитись ознайомитись з документацією бібліотеки, щоб дізнатись про доступні параметри та як їх налаштувати.

5. Обробка помилок: Під час використання бібліотек можуть виникати помилки або виключення. Важливо обробляти ці помилки, щоб ваша програма не завершилася аварійно. Ви можете використовувати блоки try-except для обробки виключень та забезпечення коректної поведінки програми.

Це загальний підхід до використання бібліотек в Python. Зверніть увагу, що кожна бібліотека може мати свої особливості та специфічний спосіб використання, тому рекомендується ознайомитись з документацією бібліотеки, яку ви плануєте використовувати.

2.4 Популярні бібліотеки для створення чат-ботів

Бібліотека **Flask** є одним з найпопулярніших фреймворків для веб-розробки на мові Python. Вона дозволяє легко створювати веб-додатки, включаючи чат-ботів. Основні особливості Flask:

- Легкість використання: Flask має простий і зрозумілий синтаксис, що дозволяє швидко розпочати розробку. Він не накладає жорстких обмежень і дозволяє програмістам вибирати, як організувати свій код і структурувати проект.

- **Маршрутизація:** Flask надає зручний механізм маршрутизації, що дозволяє визначати, які функції повинні виконуватися для кожного URL. Це дає можливість керувати поведінкою чат-бота в залежності від вхідного URL.
- **Інтеграція з шаблонами:** Flask підтримує використання шаблонів, що дозволяє розділити логіку програми від представлення. Ви можете використовувати шаблони для відображення веб-сторінок з відповідями чат-бота.
- **Розширюваність:** Flask має велику кількість розширень (extensions) із зручними функціями, такими як обробка форм, аутентифікація користувачів, робота з базами даних тощо. Це дозволяє розширити функціональність вашого чат-бота з використанням готових рішень.
- **Підтримка REST API:** Flask надає можливість створювати RESTful API, що дозволяє взаємодіяти з вашим чат-ботом за допомогою HTTP-запитів.
- **Розгортання:** Flask дозволяє легко розгортати веб-додатки на різних платформах і хмарних сервісах, таких як Heroku, AWS, GCP і т. д.

Flask є гнучким і потужним інструментом для розробки чат-ботів і забезпечує широкі можливості для налаштування та розширення.

Django - це повноцінний фреймворк для веб-розробки на мові Python. Він надає набір інструментів і бібліотек для швидкої розробки безпечних та масштабованих веб-додатків, включаючи чат-ботів. Основні особливості Django:

1. **Об'єктно-реляційна модель (ORM):** Django надає розширену ORM, яка дозволяє зручно працювати з базами даних і виконувати операції зберігання та отримання даних. Це полегшує роботу з базою даних для чат-бота.
2. **Автоматична адміністративна панель:** Django надає вбудовану адміністративну панель, яка дозволяє легко створювати та керувати адміністративним інтерфейсом для вашого чат-бота. Ви можете легко налаштувати моделі та адміністративні функції.

3. URL-маршрутизація: Django має механізм URL-маршрутизації, що дозволяє визначати, які функції або класи виконуватимуться для кожного URL. Це дозволяє легко налаштувати поведінку вашого чат-бота залежно від URL.

4. Шаблонізація: Django надає вбудовану систему шаблонів, яка дозволяє створювати веб-сторінки з використанням шаблонів і вставляти динамічні дані. Це корисно для створення відповідей чат-бота у вигляді веб-сторінок.

5. Безпека: Django має вбудовані механізми для захисту від загроз безпеки, таких як захист від CSRF-атак, аутентифікація користувачів, авторизація тощо. Ви можете легко захистити ваш чат-бот від потенційних загроз безпеки.

6. Розширюваність: Django має велику кількість розширень (packages) та сторонніх бібліотек, що дозволяють розширити функціональність вашого чат-бота з використанням готових рішень.

Django є потужним фреймворком, який дозволяє швидко розробляти складні веб-додатки, включаючи чат-ботів, забезпечуючи ряд важливих функцій і зручний інтерфейс розробки.

PyTorch - це бібліотека для машинного навчання та глибокого навчання на мові програмування Python. Вона надає зручні інструменти для створення, навчання і використання нейронних мереж. Основні особливості PyTorch:

1. Динамічний граф обчислень: Однією з ключових особливостей PyTorch є динамічний граф обчислень. Це означає, що граф обчислень формується під час виконання програми, що дозволяє більш гнучко визначати моделі та виконувати різноманітні обчислення.

2. Легкість використання: PyTorch має простий та зрозумілий інтерфейс, що робить його легким у використанні та експериментах з моделями глибокого навчання. Вона надає багато вбудованих функцій для швидкого створення та навчання моделей.

3. GPU-підтримка: PyTorch має вбудовану підтримку використання графічних процесорів (GPU) для прискорення обчислень. Це дозволяє значно покращити швидкодію навчання моделей глибокого навчання.

4. Багатошарова архітектура: PyTorch надає можливість створювати складні багатошарові моделі глибокого навчання з використанням різних типів шарів та активаційних функцій.

5. Переносність: PyTorch може працювати на різних платформах, включаючи Windows, macOS та Linux. Вона також підтримує інтеграцію з іншими бібліотеками машинного навчання, такими як NumPy.

6. Спільнота та ресурси: PyTorch має активну спільноту користувачів, яка надає підтримку та поширює навчальні матеріали та приклади використання.

PyTorch є популярною бібліотекою серед дослідників і практиків глибокого навчання, і вона дозволяє вам ефективно працювати з нейронними мережами та виконувати різні завдання у сфері машинного навчання.

Бібліотека **Rasa** є популярним інструментом для розробки чат-ботів з використанням природної мови (Natural Language Understanding, NLU) та діалогових систем. Rasa надає повний набір інструментів для побудови розумних чат-ботів з можливістю розпізнавання та розуміння намірів користувача та взаємодії з ним.

Основні особливості бібліотеки Rasa:

1. Розуміння мови (NLU): Rasa має вбудований модуль NLU, який дозволяє розпізнавати наміри користувача та екстрагувати параметри з введеної мови. Це дозволяє визначати, яку дію або відповідь повинен здійснити чат-бот на підставі вхідного повідомлення.

2. Менеджмент діалогів: Rasa надає можливості для управління діалогом з користувачем. Ви можете визначати сценарії, правила та логіку діалогу, щоб забезпечити природну та плавну взаємодію з чат-ботом.

3. Тренування моделей: Rasa надає інструменти для тренування моделей NLU та діалогових моделей. Ви можете навчати моделі на основі

вхідних даних, вдосконалювати їх і налаштовувати для кращої розпізнаваності намірів та більш точної взаємодії з користувачем.

4. Розподілена архітектура: Rasa дозволяє розподіляти та масштабувати чат-бота на різних серверах, що полегшує обробку великого обсягу запитів та підтримує підвищену доступність.

5. Підтримка мови Python: Rasa написана на мові Python і використовується з іншими популярними бібліотеками Python, що дозволяє вам комбінувати її з іншими інструментами та фреймворками Python для створення повноцінних рішень чат-ботів.

Rasa надає гнучкість та розширюваність для розробки чат-ботів, забезпечуючи розуміння мови, управління діалогами та інші важливі функції, що допомагають побудувати потужні та інтелектуальні чат-боти.

Бібліотека ChatterBot є популярним інструментом для розробки чат-ботів з використанням штучного інтелекту (AI) для генерації відповідей на основі попередньо навчених моделей. ChatterBot надає простий інтерфейс для створення, навчання та використання чат-ботів з метою спілкування з користувачами.

Основні особливості бібліотеки ChatterBot:

1. Генерація відповідей: ChatterBot використовує навчену модель для генерації відповідей на основі вхідного повідомлення користувача. Вона використовує методи штучного інтелекту, такі як генеративні моделі, статистичні аналізи та правила, для визначення найбільш відповідного тексту відповіді.

2. Навчання моделей: ChatterBot надає можливість навчати моделі на основі вхідних даних, що дозволяє персоналізувати та покращувати відповіді чат-бота. Ви можете навчати модель на власних даних чи використовувати попередньо навчені моделі для швидкого старту.

3. Конфігурування: ChatterBot дозволяє налаштовувати різні параметри моделі, такі як алгоритми генерації відповідей, міра подібності

тексту, чутливість до контексту та інші. Це дозволяє контролювати поведінку чат-бота та адаптувати його під ваші потреби.

4. Мовна підтримка: ChatterBot підтримує різні мови та має можливість роботи з текстом у міжнародних мовах. Ви можете налаштувати бота для спілкування в мові, яка вам потрібна.

5. Розширюваність: ChatterBot дозволяє розширювати його функціональність шляхом додавання власних модулів та розширень. Ви можете додавати нові алгоритми, фільтри, обробники та інші компоненти для покращення роботи бота.

ChatterBot є потужним інструментом для розробки чат-ботів, який дозволяє швидко створювати і налаштовувати моделі для взаємодії з користувачами на основі природної мови.

BotFather в Telegram - це спеціальний бот, створений для створення та керування іншими ботами в Telegram. Деякі з можливостей BotFather включають:

1. Створення нових ботів: BotFather надає можливість створювати нових ботів, вказуючи їх ім'я та ім'я користувача. Кожен новий бот отримує унікальний API-ключ, який використовується для взаємодії з Telegram API.

2. Керування ботами: BotFather дозволяє керувати налаштуваннями індивідуальних ботів, такими як зміна імені, імені користувача, опису та профілю. Ви також можете налаштувати фотографію профілю для свого бота.

3. Налаштування команд: BotFather дозволяє налаштувати команди для вашого бота. Ви можете визначити, які команди активують певні функції або відповідають на конкретні запити користувачів.

4. Отримання статистики: BotFather надає інформацію про статистику використання вашого бота, включаючи кількість користувачів, повідомлень та команд.

5. Захист API-ключа: BotFather надає можливість генерувати новий API-ключ для вашого бота, якщо поточний ключ був скомпрометований або втрачений.



Рисунок 2.2 - Лого Botfather в Telegram

BotFather є потужним інструментом для створення та керування ботами в Telegram, який дозволяє легко налаштовувати та керувати функціональністю вашого бота.

РОЗДІЛ 3 ПРОЕКТУВАННЯ, РОЗРОБКА ТА ВПРОВАДЖЕННЯ ЧАТ-БОТУ ДЛЯ НАДАННЯ НАВЧАЛЬНИХ КОНСУЛЬТАЦІЙ

3.1 Алгоритм надання навчальних консультацій чат-ботом

Алгоритм надання навчальних консультацій має наступний вигляд:
Вхід: Користувач починає спілкування з чат-ботом, наприклад, вводить привітання або запитання.

1. Привітання та ідентифікація: Чат-бот вітає користувача і, якщо необхідно, проводить процес ідентифікації, наприклад, попросивши користувача ввести свої дані або студентський номер.

2. Вибір послуг: Чат-бот надає перелік послуг, які доступні для навчальних консультацій, наприклад, питання щодо навчальних матеріалів, допомога у вирішенні завдань, роз'яснення понять тощо. Користувач може обрати одну зі списку або ввести своє власне питання.

3. Обробка запиту: Чат-бот аналізує запит користувача і спробує зрозуміти, яка конкретна допомога потрібна. Він може використовувати алгоритми обробки природної мови (Natural Language Processing - NLP) для розуміння та інтерпретації запитів.

4. Надання відповіді: Чат-бот надає відповідь на запит користувача, можливо, застосовуючи збережені попередньо створені відповіді або використовуючи базу знань. Відповідь може бути текстовою, аудіо- або відеофайлом, посиланням на веб-ресурс тощо.

5. Додаткова інтеракція: Якщо користувач має додаткові питання або потребує більш детальної інформації, чат-бот надає можливість продовжити спілкування. Він може задавати уточнюючі запитання, пропонувати додаткові ресурси або сприяти у розв'язанні проблеми.

6. Завершення консультації: Після вирішення питання чат-бот вітає користувача і пропонує додаткову допомогу або закриває сесію консультації.

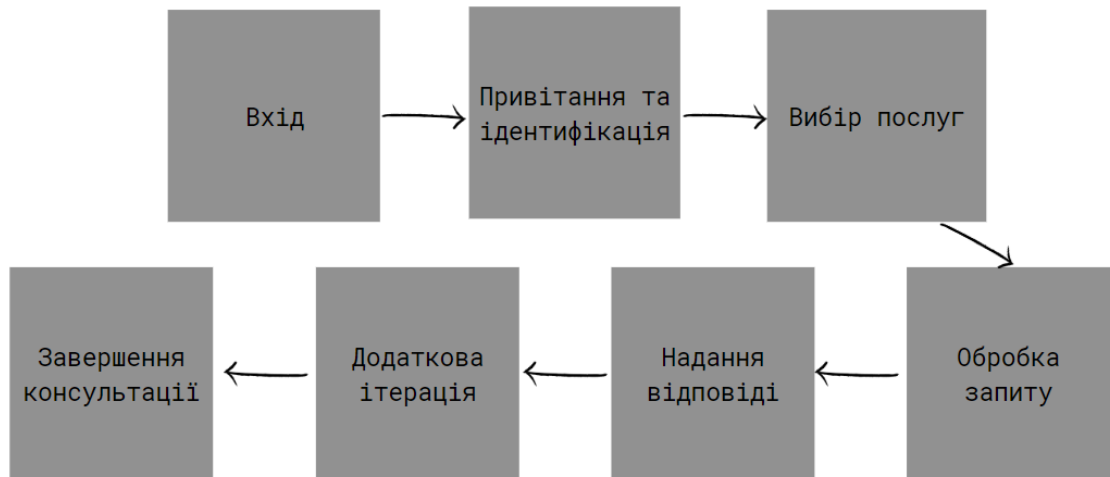


Рисунок 3.1 - Схема надання послуг чат-ботом

3.2 Використання Chat GPT API для збільшення гнучкості системи

Chat GPT API - це інтерфейс програмування застосунків, який надає доступ до моделі Chat GPT для використання у власних програмах або сервісах. Принцип роботи API полягає в тому, що ви можете відправляти HTTP-запити до сервера OpenAI, який містить модель Chat GPT, і отримувати відповіді в форматі JSON.

Для отримання доступу до Chat GPT API необхідно перейти на веб-сайт OpenAI за адресою <https://openai.com/>. Після реєстрації та входу в систему, треба ознайомитись з деталями API: переглянути документацію OpenAI, щоб отримати більше інформації про Chat GPT API, включаючи можливості, обмеження та цінову політику. Можна знайти докладну інформацію про використання API та те, як налаштувати ваші запити.

Для відправки запиту можна використовувати HTTP-клієнт (наприклад, Python бібліотеку requests) для відправки POST-запитів до URL API OpenAI, передаючи вміст запиту у форматі JSON. Запит повинен містити текстове повідомлення або декілька повідомлень, які представляють розмову з чат-ботом. Після відправки запиту до API OpenAI буде отримана відповідь у

форматі JSON, яка містить текстову відповідь чат-боту. Необхідно обробити цю відповідь у додатку або сервісі.

Після відправки запиту до Chat GPT API, він повертає відповідь у форматі JSON, далі необхідно обробити цю відповідь у своєму кодї, щоб отримати необхідні дані.

Основним полем, яке можна використовувати для отримання відповіді чат-боту, є поле "choices" або "messages" у відповіді JSON. Зазвичай, це масив об'єктів, які містять текстову відповідь, інформацію про автора або систему, а також інші метадані.

Код може витягти текстову відповідь з цього масиву і використовувати її за потреби. Наприклад, якщо використовується поле "choices", можна отримати перше повідомлення з масиву, яке містить відповідь чат-боту. Також можлива обробка інших полів відповіді, якщо вони необхідні для програми.

Важливо врахувати обмеження використання API, такі як ліміти кількості запитів на часовий проміжок або обмеження довжини текстового вводу. Докладну інформацію щодо використання API та обмежень можна знайти в документації OpenAI.

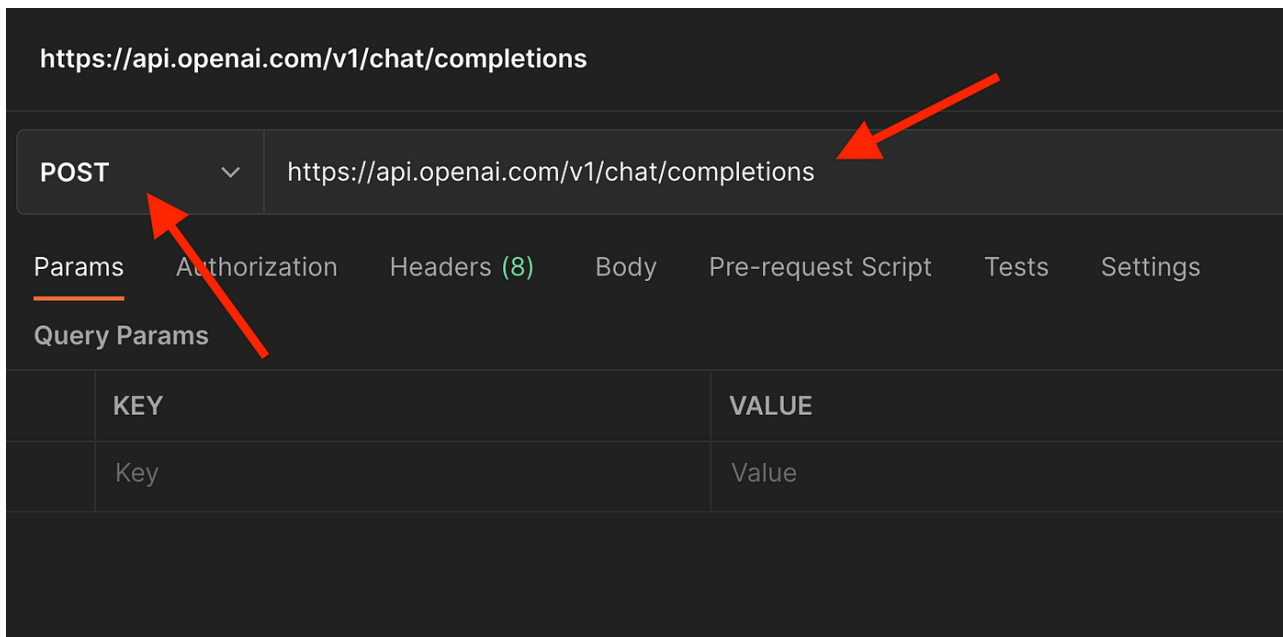


Рисунок 3.2 - Використання Chat GPT API в Postman

3.3 Інструкція для користування ботом “Your Assistant”

Чат-бот використовує Telegram API та OpenAI ChatGPT API для надання відповідей на запитання користувачів. Основний принцип дії бота наступний:

1. Спочатку імпортуються необхідні бібліотеки, встановлюються необхідні змінні, такі як токен для доступу до Telegram API та ключі для доступу до ChatGPT API.
2. Створюється екземпляр бота за допомогою класу “telebot.TeleBot” та визначаються обробники подій.
3. Обробник команди “/start” вітає користувача.
4. Обробник повідомлень (функція “handle_message”) отримує питання користувача та перевіряє наявність команд “/decrease” та “/default”. Якщо команда “/decrease” виконана, поріг подібності зменшується, а якщо команда “/default”, поріг встановлюється за замовчуванням.
5. За допомогою функції “get_similar_answer” отримується найбільш схожа відповідь з текстового файлу. Якщо відповідь існує, вона надсилається користувачеві.
6. Якщо немає відповіді для запитання користувача, бот повідомляє про це та зберігає невідоме питання в окремому файлі.
7. Виконується запит до ChatGPT API за допомогою функції “get_gpt_answer”, передаючи питання користувача. Отримана відповідь зберігається в файлі та надсилається користувачеві.
8. Функція “save_question” зберігає невідоме питання у файлі.
9. Функція “save_answer” зберігає питання та відповідь у файлі.
10. Функція “format_gpt_answer” форматує відповідь в форматі питання-відповіді.
11. Бот розпочинає роботу за допомогою методу “bot.polling()”, який запускає цикл отримання та обробки повідомлень від користувачів.

Для старту роботи бота необхідно вказати команду “/start”, після чого чат-бот надішле повідомлення з привітанням, і ми можемо працювати з ним.

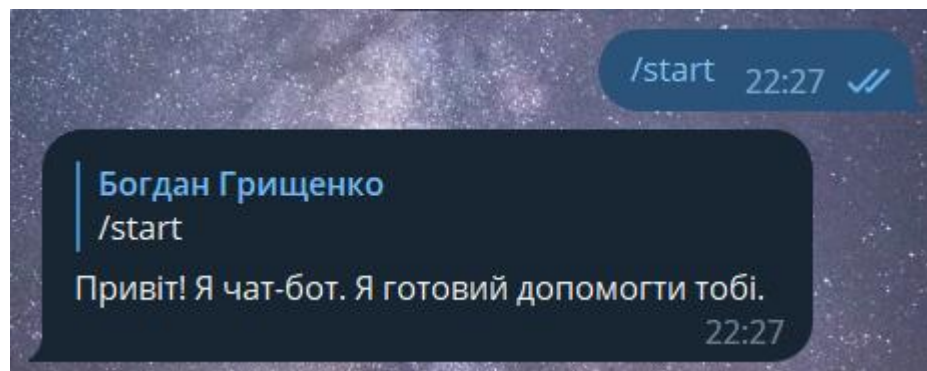


Рисунок 3.3 - Початок роботи з чат-ботом

Коли користувач задає питання, бот перевіряє файл на наявність відповіді і відповідає на питання з файлу. Якщо питання невідоме, воно зберігається в файлі в окремому файлі, для того щоб можна було проаналізувати запити чат-боту.

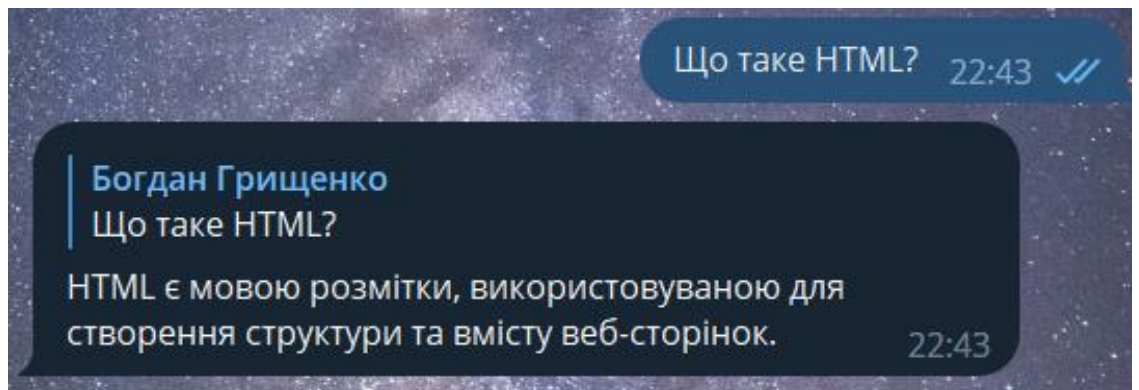


Рисунок 3.4 - Отримання відповіді від чат-боту

У тому випадку, якщо чат-бот отримав питання, відповідь на яке відсутнє у його БД, то відбувається звернення до Chat GPT API, потім його відповідь форматується, та додається до наявної БД, для того щоб в майбутньому миттєво надати інформацію.

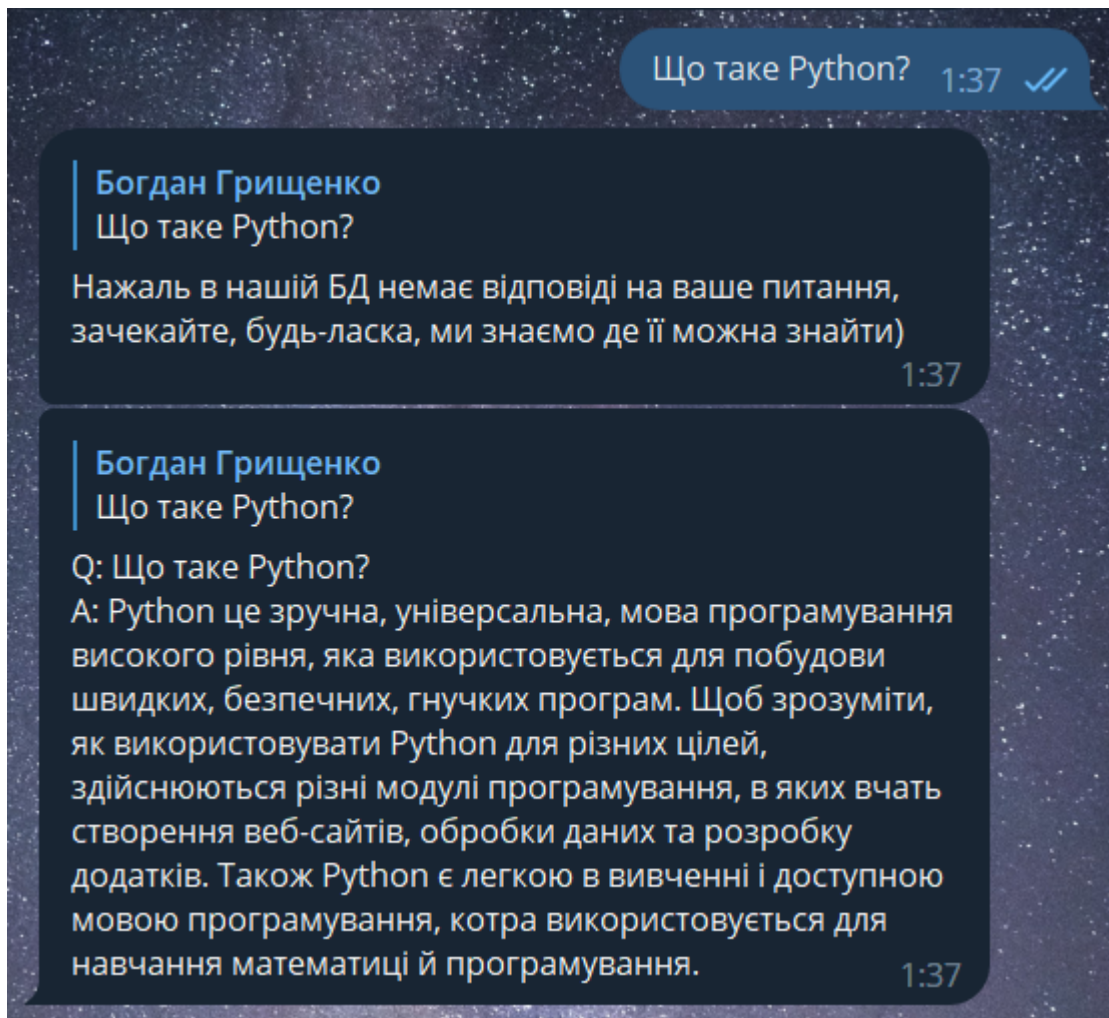


Рисунок 3.4 - Відповідь чат-боту на невідоме питання

ВИСНОВОК

У даній бакалаврській роботі була розглянута тема розробки чат-бота для надання навчальних консультацій у месенджері Telegram за допомогою мови програмування Python. Метою дослідження було створення ефективного інструменту, який допоможе студентам отримати швидкі та зручні консультації з навчальних питань у будь-який час.

У процесі розробки було проведено аналіз необхідності чат-боту для навчальних консультацій та виявлено ряд переваг, які він може принести студентам, викладачам та адміністрації навчального закладу. Серед цих переваг можна відзначити зручну та доступну комунікацію, підвищення ефективності навчання, покращення розуміння потреб студентів та забезпечення цілодобової підтримки.

Особлива увага приділялася вибору технологій та алгоритмів, що стоять за роботою чат-боту. Мова програмування Python була обрана через її простоту в освоєнні, велику спільноту розробників та широкі можливості. Для реалізації чат-боту використовувався фреймворк Telegram Bot API, який надавав зручний інтерфейс для роботи з Telegram.

В результаті роботи був розроблений чат-бот для навчальних консультацій, який був успішно інтегрований в Telegram. Чат-бот забезпечував студентам можливість задавати питання, отримувати відповіді та надання додаткових матеріалів або ресурсів. Він демонстрував високу швидкість відповідей та ефективну комунікацію.

Впровадження чат-боту для навчальних консультацій має значний вплив на процес навчання. Він допомагає покращити доступність та ефективність навчальної підтримки, сприяє підвищенню якості навчання та розумінню потреб студентів. Зворотній зв'язок від студентів, викладачів та адміністрації підтверджує позитивний вплив чат-боту на навчальний процес.

Загалом, розробка та впровадження чат-боту для навчальних консультацій у месенджері Telegram мовою Python є перспективним напрямком розвитку освітніх технологій. Вона допомагає забезпечити доступність та

швидкість навчальної підтримки, покращує комунікацію та сприяє зростанню якості навчання. Чат-бот стає незамінним інструментом для студентів у процесі самоосвіти та навчання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Логінова, Ю. О. Використання чат-ботів у вищій школі: переваги та виклики. Вісник Харківського національного університету імені В. Н. Каразіна. Серія: Психологія, 73, 109-114.
2. Калініна, М. М. Роль чат-ботів у підвищенні якості освіти: перспективи та виклики. Педагогічна освіта: теорія і практика, 28, 78-85.
3. Петренко, І. О., & Ковальчук, А. М. Використання чат-ботів у навчальному процесі вищої школи: досвід та перспективи. Науковий вісник Міжнародного гуманітарного університету, 43(2), 157-162.
4. Кузьмін В . С., Григоренко, Т. М. Чат-боти в системі навчання вищого навчального закладу: можливості та перспективи використання. Інформаційні технології в освіті, 37, 61-70.
5. Черкасова, І. І. Використання чат-ботів у вищій школі: особливості та перспективи. Науковий вісник Івано-Франківського національного технічного університету нафти і газу, 2(47), 71-76.
6. Anderson, T., & Kanuka, H. E-Research: Methods, Strategies, and Issues. Allyn & Bacon.
7. Bickmore, T., & Picard, R. Establishing and maintaining long-term human-computer relationships. ACM Transactions on Computer-Human Interaction (TOCHI), 12(2), 293-327.
8. Chiu, T. K. F., & Churchill, D. Designing chatbots for customer service. In Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (pp. 2650-2656).
9. De Angeli, A., Johnson, G. I., & Coventry, L. The social and interactional dimensions of human-computer interfaces. International Journal of Human-Computer Studies, 60(3), 343-353.
10. Denecke, K. How to create a chatbot for health care: challenges and ethical considerations. JMIR Medical Informatics, 7(1), e11945.
11. Dennen, V. P. (Ed.). Theoretical underpinnings of distance education and online learning. Routledge.

- 12.D'Mello, S. K., & Graesser, A. AutoTutor and affective AutoTutor: Learning by talking with cognitively and emotionally intelligent computers that talk back. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 2(4), 23.
- 13.Dong, L., Cui, H., & Zhang, R. Design and implementation of an intelligent chatbot for E-Learning platforms. *IEEE Access*, 8, 95083-95093.
- 14.Følstad, A., & Brandtzæg, P. B. Chatbots and the new world of HCI. *Interactions*, 24(4), 38-42.
- 15.Gao, F., Luo, T., Zhang, K., Harrison, R., & Chang, V. Examining the influencing mechanism of social support on individual performance: a social network perspective. *Information Systems Frontiers*, 21(6), 1405-1417.
- 16.Greer, J. E., McCalla, G., & Cooke, J. Agents, agents everywhere! In *International Conference on Intelligent Tutoring Systems* (pp. 19-29). Springer, Cham.
- 17.Hone, K., & Graham, R. Towards a tool for the subjective assessment of speech system interfaces (SASSI). *Natural Language Engineering*, 6(3-4), 287-303.
- 18.Hu, X., Zhang, Y., Zhang, L., & Zhang, B. A survey on opinion mining and sentiment analysis: tasks, approaches and applications. *Knowledge-Based Systems*, 89, 14-46.
- 19.Johnson, W. L., Rickel, J. W., & Lester, J. C. Animated pedagogical agents: Face-to-face interaction in interactive learning environments. *International Journal of Artificial Intelligence in Education*, 11(1), 47-78.
- 20.Khalifa, M., & Shen, Y. Explaining the adoption of transactional B2C mobile commerce. *Journal of Enterprise Information Management*, 21(2), 110-124.
- 21.Kim, J. W., & Ko, E. Impacts of trust on consumer behavior in electronic commerce: empirical evidence from Korea. *Online Information Review*, 34(6), 935-951.
- 22.Koedinger, K. R., & Corbett, A. T. Cognitive tutors: Technology bringing learning science to the classroom. In *Handbook of educational psychology* (pp. 645- 656). Prentice Hall.

23. Lindsey Susan Aloia, Megan McTigue. Communication Research Reports Volume 36, 2019: книга. 126 – 135 с. URL: <https://www.tandfonline.com/doi/full/10.1080/08824096.2019.1590191?scroll=top&needAccess=true>.
24. Chudari, Mashudi, Fatihaturasyidah. Achieving ideal career 2020: публікація. URL: <https://journal.uny.ac.id/index.php/progcouns/article/view/34868/14744>.
25. Academic Support Harvard Extension School: веб-сайт. URL: <https://extension.harvard.edu/blog/5-strategies-for-academic-success-using-your-strengths/>.
26. Development of Learning Strategies Pearson: веб-сайт URL: <https://www.pearson.com/en-us/schools/products-services/academic-support-programs.html>.
27. Collaborative Learning Next Generation Learning Challenges: веб-сайт. URL: <https://www.nextgenlearning.org/articles/five-essential-systems-of-student-support>.
28. Консультація навчальна: веб-сайт. URL: https://uk.wikipedia.org/wiki/Консультація_навчальна.
29. Role of AI chatbots in education: веб-сайт. URL: <https://educationaltechnologyjournal.springeropen.com/articles/10.1186/s41239-023-00426-1>.
30. Educational Technology: веб-сайт. URL: https://en.wikipedia.org/wiki/Educational_technology.
31. Association for Educational Communications and Technology (AECT): веб-сайт. URL: <https://www.aect.org/>
32. International Society for Technology in Education (ISTE): веб-сайт. URL: <https://www.iste.org/>
33. Educause: веб-сайт. URL: <https://www.educause.edu/>
34. OpenAI ChatGPT. URL: <https://openai.com/chatgpt/>.
35. EdX: веб-сайт. URL: <https://www.edx.org/>.
36. Офіційний веб-сайт Python. URL: [Python.org](https://python.org)

- 37.Офіційний веб-сайт PyCharm. URL: <https://www.jetbrains.com/pycharm/>
- 38.PyCharm Community Edition: веб-сайт. URL
:<https://www.jetbrains.com/pycharm/download/#section=windows>
- 39.PyCharm Edu :веб-сайт. URL: <https://www.jetbrains.com/pycharm-edu/>
- 40.Офіційний веб-сайт Flask: веб-сайт. URL: <https://flask.palletsprojects.com/>
- 41.Flask Documentation :веб-сайт. URL:
<https://flask.palletsprojects.com/en/2.0.x/>
- 42.Flask на GitHub :веб-сайт. URL: <https://github.com/pallets/flask>
- 43.Офіційний веб-сайт Django :веб-сайт. URL: <https://www.djangoproject.com/>
- 44.Django на GitHub: веб-сайт. URL: <https://github.com/django/django>

ДОДАТОК

Лістинг коду

```
main.py ×
1 import telebot
2 import difflib
3 import requests
4
5 # Отримання токена вашого бота від BotFather
6 token = "6093145688:AAFq044c8KmAHdmZmbkA0I7oa0h1LtCEtoc"
7
8 # Шлях до текстового файлу, де зберігатимуться відповіді
9 file_path = "answers.txt"
10
11 # Шлях до текстового файлу, де зберігатимуться невідомі питання
12 unknown_file_path = "unknown_questions.txt"
13
14 # Адреса API для отримання відповідей від ChatGPT
15 gpt_api_url = "https://api.openai.com/v1/engines/text-davinci-003/completions"
16
17 # API ключ для доступу до ChatGPT
18 gpt_api_key = "sk-xPIkX7frqHYybAeKzrIT3BlbkFJhutoXceJYFuJtHlwohSs"
19
20 # Створення екземпляру бота
21 bot = telebot.TeleBot(token)
22
23 # Початковий поріг подібності питань
24 initial_cutoff = 1
25 # Приріст порогу подібності питань при кожній ітерації
26 cutoff_increment = 0.05
27
28 # Обробник команди /start
29 @bot.message_handler(commands=['start'])
30 def handle_start(message):
31     # Вітання та відповідь на команду /start
32     bot.reply_to(message, "Привіт! Я чат-бот. Я готовий допомогти тобі.")
33
34 # Обробник повідомлень
35 @bot.message_handler(func=lambda message: True)
36 def handle_message(message):
37     # Отримання питання користувача
38     question = message.text
39
```

```

40 # Перевірка команди /decrease
41 if question == '/decrease':
42     # Зміна порогу подібності
43     global initial_cutoff
44     initial_cutoff -= cutoff_increment
45     if initial_cutoff < 0:
46         initial_cutoff = 0
47
48     bot.reply_to(message, f"Новий поріг подібності: {initial_cutoff}")
49     return
50
51 # Перевірка команди /default
52 if question == '/default':
53     # Повернення значення порогу подібності до 0.6
54     initial_cutoff = 1
55
56     bot.reply_to(message, f"Поріг подібності встановлено за замовчуванням: {initial_cutoff}")
57     return
58
59 # Отримання найбільш схожої відповіді з файлу
60 answer = get_similar_answer(question)
61
62 if answer and answer != "Unknown":
63     # Відповідь на питання у форматі питання-відповіді
64     formatted_answer = f"Q: {question}\nA: {answer}"
65     bot.reply_to(message, formatted_answer)
66 else:
67     # Відповідь на невідоме питання
68     bot.reply_to(message, "Нажаль в нашій БД немає відповіді на ваше питання, зачекайте, будь-ласка, ми знаємо де її можна знайти")
69
70     # Збереження невідомого питання в файлі
71     save_question(question)
72
73     # Отримання відповіді від ChatGPT API
74     gpt_answer = get_gpt_answer(question)
75
76     if gpt_answer:
77         # Збереження відповіді в файлі
78         save_answer(question, gpt_answer)

```

```

80     # Форматування відповіді в форматі питання-відповіді
81     formatted_gpt_answer = format_gpt_answer(question, gpt_answer)
82     bot.reply_to(message, formatted_gpt_answer)
83
84     else:
85         bot.reply_to(message, "Вибач, відбулася помилка при отриманні відповіді від ChatGPT.")
86
87 1 usage
88 def get_similar_answer(question):
89     # Зчитування відповідей з файлу
90     with open(file_path, "r") as file:
91         lines = file.readlines()
92
93     # Пошук найбільш схожої відповіді за питанням
94     max_ratio = 0
95     similar_answer = None
96     for line in lines:
97         parts = line.strip().split("|")
98         if len(parts) == 2:
99             ratio = difflib.SequenceMatcher(None, question.lower(), parts[0].lower()).ratio()
100             if ratio >= initial_cutoff and ratio > max_ratio:
101                 max_ratio = ratio
102                 similar_answer = parts[1]
103
104     return similar_answer
105
106 return None

```

```

1 usage
106 def get_gpt_answer(question):
107     # Параметри запиту до ChatGPT API
108     headers = {
109         "Content-Type": "application/json",
110         "Authorization": f"Bearer {gpt_api_key}"
111     }
112     payload = {
113         "prompt": question,
114         "max_tokens": 1000
115     }
116
117     # Виклик API для отримання відповіді від ChatGPT
118     response = requests.post(gpt_api_url, json=payload, headers=headers)
119
120     if response.status_code == 200:
121         data = response.json()
122         answer = data["choices"][0]["text"].strip()
123         return answer
124
125     return None
1 usage
126 def save_question(question):
127     # Збереження невідомого питання в окремому файлі
128     with open(unknown_file_path, "a") as file:
129         file.write(f"{question}\n")
130
131     1 usage
132     def save_answer(question, answer):
133         # Збереження питання та відповіді у файлі
134         with open(file_path, "a") as file:
135             file.write(f"{question}|{answer}\n")
136
137     1 usage
138     def format_gpt_answer(question, answer):
139         # Форматування відповіді в форматі питання-відповіді
140         formatted_answer = f"Q: {question}\nA: {answer}"
141         return formatted_answer
142
143     # Запуск бота
144     bot.polling()

```