

ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ

Навчально-науковий інститут денної освіти
Форма навчання денна
Кафедра комп'ютерних наук та інформаційних технологій

Допускається до захисту

Завідувач кафедри _____ Олена ОЛЬХОВСЬКА
(підпис)

«___» _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА

на тему
«РОЗРОБКА ДОВІДКОВОГО ЧАТ БОТУ ДЛЯ СТУДЕНТІВ»

зі спеціальності 122 «Комп'ютерні науки»

освітня програма «Комп'ютерні науки»

ступеня бакалавр

Виконавець роботи Шустваль Альона Станіславівна
_____ «___» _____ 2024 р.
(підпис)

Науковий керівник к.пед.н., доцент Кошова Оксана Петрівна
_____ «___» _____ 2024 р.
(підпис)

Рецензент _____

ПОЛТАВА 2024

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	3
ВСТУП	4
1. ПОСТАНОВКА ЗАДАЧІ.....	6
1.1. Змістовна постановка	6
1.2. Модель задачі та її характеристики	7
2. ІНФОРМАЦІЙНИЙ ОГЛЯД.....	9
2.1. Аналіз платформ для створення чат-ботів	9
2.2. Аналіз існуючих рішень у сфері довідкових чат-ботів для освітніх закладів.....	14
2.3. Вивчення технологій, які використовуються при розробці чат-ботів	19
2.4. Необхідність та актуальність теми роботи.....	21
3. ТЕОРЕТИЧНА ЧАСТИНА	23
3.1. Алгоритм роботи чат-боту	23
3.2. Блок-схема алгоритму роботи чат-боту.....	30
3.3. Обґрунтування вибору програмного забезпечення та методів	32
4. ПРАКТИЧНА ЧАСТИНА	38
4.1. Опис процесу створення та редагування чат-боту	38
4.2. Опис процесу програмної реалізації чат-боту	45
4.3. Опис роботи довідкового чат-боту	57
ВИСНОВКИ.....	63
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	65
ДОДАТОК А. КОД ЧАБ-БОТУ	65

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

Умовні позначення, символи, скорочення, терміни	Пояснення умовних позначень, скорочень, символів
BotFather	чат-бот Telegram, який дозволяє створювати власних чат-ботів
Python	інтерпретована об'єктно-орієнтована мова програмування високого рівня із суворою динамічною типізацією
pyTelegramBotAPI	бібліотека для створення боту для Telegram
чат-бот	автоматизований багатофункціональний помічник, який може збирати та показувати інформацію підписникам за їхнім запитом згідно зі заздалегідь підготовленим сценарієм

ВСТУП

Розробка довідкового чат-боту для студентів залишається вкрай актуальною та значущою темою. Сучасний світ освіти постійно змінюється, і чат-боти можуть бути використані для швидкого оновлення та надання актуальної інформації щодо нових курсів, методик, інструментів навчання.

Чат-боти можуть стати ефективним інструментом для забезпечення студентам зручного та доступного доступу до необхідної інформації. Через чат-інтерфейс студенти можуть легко отримати відповіді на питання щодо розкладу, навчальних матеріалів, термінів здачі завдань та іншої корисної інформації. Чат-боти можуть допомогти оптимізувати рутинні адміністративні процеси, такі як реєстрація на курси, підписка на заходи та отримання іншої інформації, що часто вимагає великої кількості часу та зусиль.

Розвиток технологій штучного інтелекту дозволяє створювати більш розумні та відзначені функціями чат-боти, здатні взаємодіяти зі студентами більш ефективно.

У випадку кризових ситуацій чат-бот може надати студентам необхідну підтримку та інформацію, що може полегшити стресові ситуації.

Загалом, розробка довідкового чат-боту для студентів відповідає сучасним вимогам ефективної та інноваційної освіти, забезпечуючи зручність, доступність та персоналізований підхід до кожного студента.

Метою роботи є розробка довідкового чат-боту для студентів спрямована на створення ефективного інструмента, який полегшить доступ студентів до інформації.

Об'єктом розробки є довідковий чат-бот, призначений для використання студентами в навчальних і адміністративних цілях.

Предметом розробки є елементи, які входять до складу чат-боту: чат-інтерфейс, модулі питань і відповідей, інтеграція з іншими системами.

Головне завдання – створення імплементованого та інтуїтивно зрозумілого чат-боту, який надає студентам швидкий та зручний доступ до різноманітної інформації, такої як:

- інформація щодо вступу;
- навчання;
- оплата навчання, гуртожитку тощо;
- отримання довідки, документів;
- контакти кафедр, викладачів;
- проходження практики;
- інші аспекти, що полегшують академічне життя.

Перелік використаних методів: чат-бот Telegram BotFather, який дозволяє створювати власних чат-ботів, інтерпретована об'єктно-орієнтована мова програмування високого рівня із суворою динамічною типізацією Python, бібліотека для створення боту для Telegram `pyTelegramBotAPI`.

Робота складається з чотирьох розділів. У першому розділі описано змістовну постановку, модель задачі та її характеристики. У другому розділі проведено аналіз платформ для створення чат-ботів, існуючих рішень у сфері довідкових чат-ботів для освітніх закладів, розглянуто технології, які використовуються при розробці чат-ботів, необхідність та актуальність теми роботи. У третьому розділі вказано алгоритм роботи чат-боту, блок-схему алгоритму роботи чат-боту, обґрунтування вибору програмного забезпечення та методів. У четвертому розділі описано процес створення та редагування чат-боту, процес програмної реалізації чат-боту, роботу довідкового чат-боту.

Обсяг пояснювальної записки: 73 стор., в т.ч. основна частина - 57 стор., джерела - 10 назв.

1. ПОСТАНОВКА ЗАДАЧІ

1.1. Змістовна постановка

Постановка задачі охоплює важливі аспекти розробки та впровадження довідкового чат-боту для студентів, спрямовуючи зусилля на створення високоякісного та ефективного інструменту, який відповідає вимогам освітнього середовища. [1]

1. Аналіз вимог:

- Визначення основних потреб та вимог студентів до чат-боту.
- Збір вимог від адміністраторів та фахівців.

2. Проектування інтерфейсу:

- Розробка зручного та легкого для використання чат-інтерфейсу.
- Проектування структури діалогів та визначення ключових функцій.

3. Розробка модулів:

- Створення модулів для обробки питань та надання відповідей.
- Розробка системи для інтеграції чат-боту.

4. Інтеграція з іншими системами:

- Робота над інтеграцією чат-боту з іншими інформаційними системами університету (розклад, кабінет студента, тощо).

5. Розробка функцій адміністрування:

- Створення інтерфейсу для адміністраторів для оновлення та управління інформацією в чат-боті.

6. Тестування та валідація:

- Проведення тестів для перевірки коректності роботи чат-боту.
- Збір відгуків від користувачів та внесення необхідних корекцій.

7. Впровадження та підтримка:

- Введення чат-боту в експлуатацію.
- Забезпечення технічної підтримки та можливості оновлення системи.

8. Оцінка ефективності:

- Здійснення аналізу використання чат-боту студентами та адміністраторами.
- Визначення позитивних змін у доступі до інформації та ефективності адміністративних процесів.

9. Документація:

- Підготовка технічної та користувацької документації для користувачів та адміністраторів.

1.2. Модель задачі та її характеристики

Задачі, які розробка довідкового чат-боту для студентів може вирішувати, включають, але не обмежуються наступними аспектами:

1. Інформація щодо вступу:

Надання інформації про вимоги до вступу, терміни подачі документів, списки необхідних документів, процедури вступу, вступні випробування тощо.

2. Навчання:

Доступ до розкладу занять, перегляд інформації про предмети та викладачів, матеріали лекцій і практичних занять, рекомендації щодо успішного навчання.

3. Технічна підтримка дистанційного навчання:

Вирішення проблем з платформами дистанційного навчання, інструкції з використання віртуальних інструментів для навчання, допомога з відновленням доступу до електронних ресурсів тощо.

4. Оплата навчання, гуртожитку тощо:

Інформація про процедуру оплати навчання, гуртожитку, наявність стипендій та інших фінансових питань.

5. Отримання довідки/документів:

Видання довідок про навчання, студентські квитки, академічні довідки та інші документи.

6. Контакти кафедр/викладачів:

Пошук та надання контактної інформації про викладачів та кафедри, можливість запису на консультації.

7. Проходження практики:

Інформація про можливості проходження практики, узгодження місць практики, подання необхідних документів.

8. Контакти всіх відділів/підрозділів:

Забезпечення доступу до контактної інформації всіх відділів та підрозділів університету для рішення різних адміністративних питань.

Створення довідкового чат-боту для студентів виявляється необхідною та корисною ініціативою, що спрямована на полегшення їхнього навчального процесу та підвищення загального рівня зручності та доступності інформації. Чат-бот, який здатний надавати швидку та точну інформацію щодо вступу, навчання, технічної підтримки, фінансових питань, отримання документів, контактів викладачів та інших аспектів студентського життя, забезпечить студентам зручність, ефективність та підтримку протягом їхнього навчального шляху. Такий інструмент сприятиме покращенню загальної навчальної атмосфери та підвищенню задоволення студентів від навчання.

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Аналіз платформ для створення чат-ботів

Microsoft Bot Framework надає інструменти для створення розумних чат-ботів, в тому числі для освітніх закладів (рис. 2.1). Забезпечує інтеграцію з Microsoft Teams та іншими платформами.

Приклад використання: створення чат-боту для підтримки студентів під час дистанційного навчання. Бот може відповідати на питання щодо викладання онлайн, надавати інформацію про доступ до ресурсів та допомагати з технічними проблемами. [2]

Позитивні аспекти:

- Зручний інтерфейс розробки та підтримка багатьох мов програмування.
- Інтеграція зі стеком Microsoft, що полегшує роботу з іншими продуктами компанії.

Негативні аспекти:

- Залежність від екосистеми Microsoft, що може обмежувати можливості для користувачів із іншими платформами.

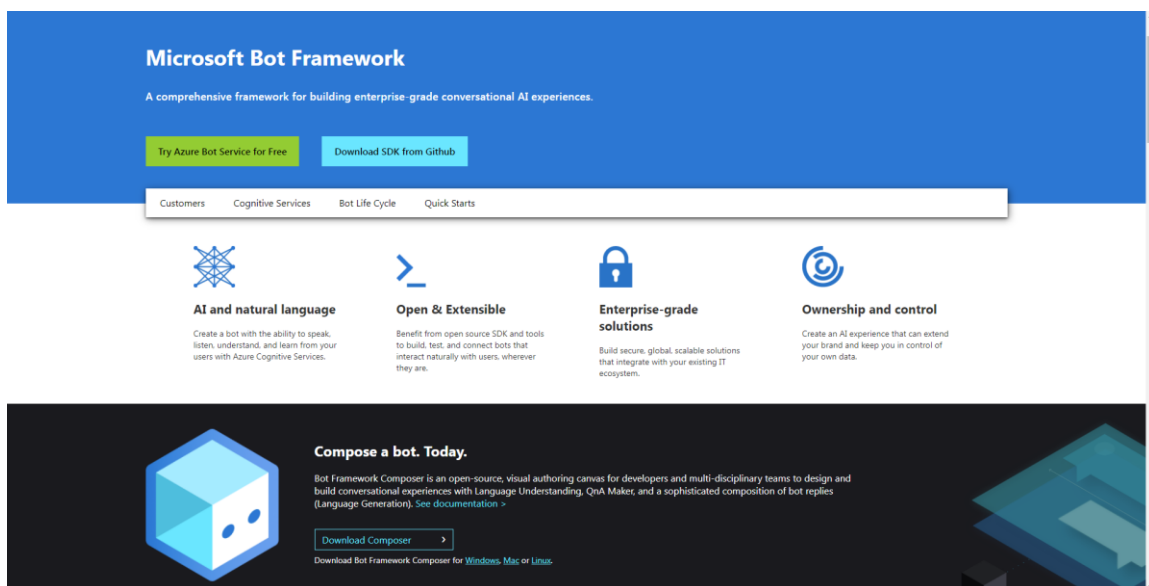


Рисунок 2.1 – Сайт Microsoft Bot Framework

Botpress - це відкритий кодовий фреймворк для створення чат-ботів. Надає розширюваність та можливості кастомізації (рис. 2.2).

Приклад використання: розробка чат-боту для нових студентів з метою надання інформації про кампус, житло, бібліотеку тощо. Бот може відповідати на загальні питання новачків та надавати корисну інформацію. [3]

Позитивні аспекти:

- Велика гнучкість та розширюваність завдяки відкритому коду.
- Наявність графічного інтерфейсу для налаштування боту без програмування.

Негативні аспекти:

- Потребує базових знань програмування для повного використання можливостей фреймворку.

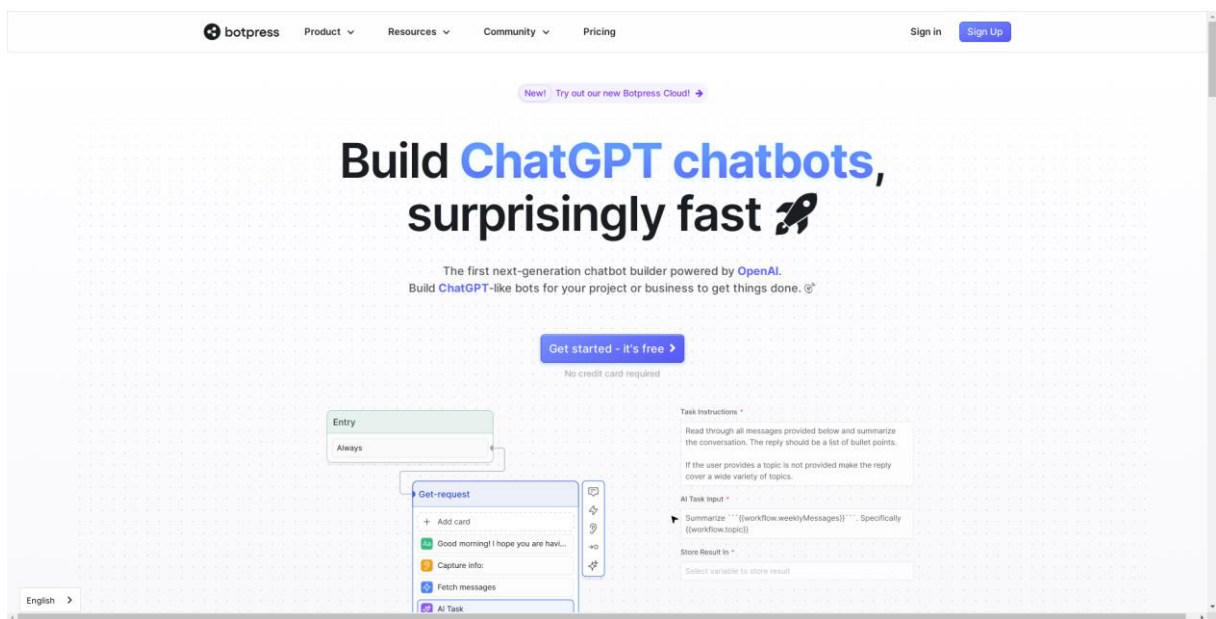


Рисунок 2.2 – Сайт Botpress

Wit.ai, придбаний Facebook, пропонує інтеграцію з месенджерами та дозволяє швидко створювати чат-ботів за допомогою природної мови (рис. 2.3).

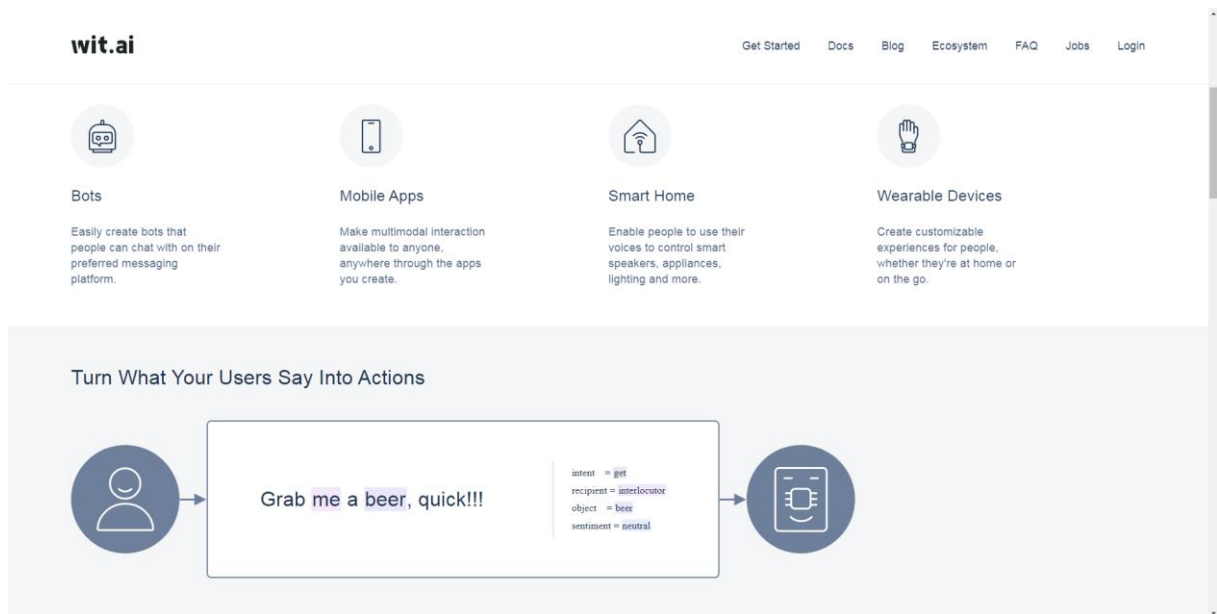


Рисунок 2.3 – Сайт Wit.ai

Приклад використання: розробка чат-боту для підтримки мовлення та навчання англійської мови. Бот може виправляти граматичні помилки, надавати відповіді на англійських, а також допомагати вибудовувати правильні речення. [4]

Позитивні аспекти:

- Навчання за допомогою природної мови для розпізнавання інтентів.
- Легкість інтеграції з різними платформами.

Негативні аспекти:

- Може вимагати додаткових інтеграцій для повноцінного функціоналу чат-боту.

SendPulse - це комплексна платформа для автоматизації маркетингу, яка, серед інших можливостей, надає інструменти для створення чат-ботів. Платформа спрямована на полегшення взаємодії з клієнтами, у тому числі у сфері освіти (рис. 2.4).

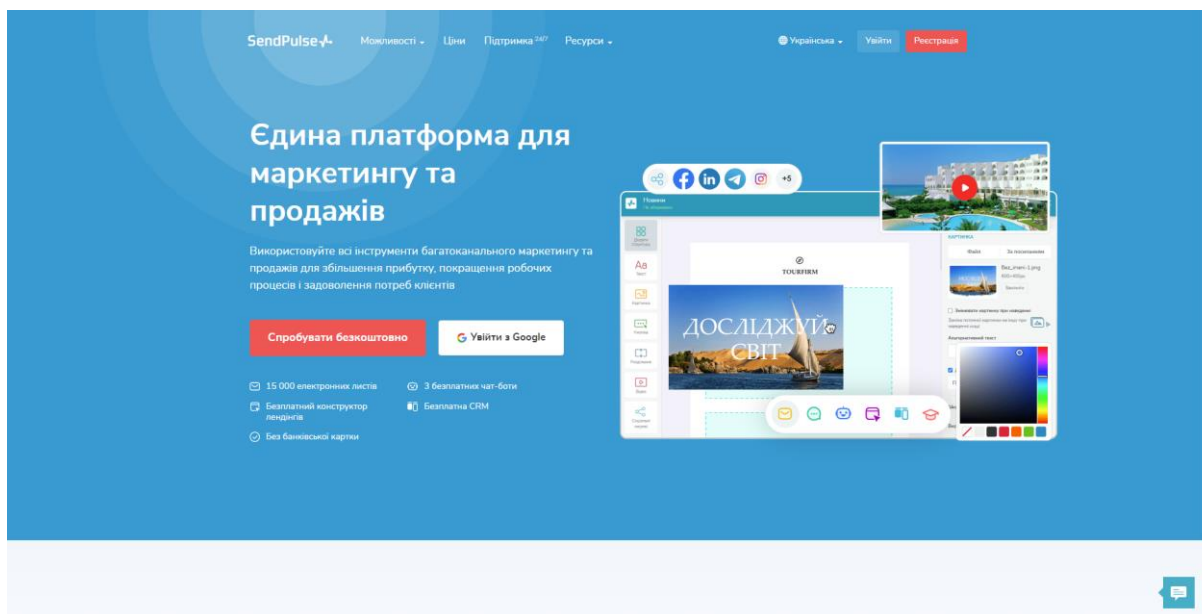


Рисунок 2.4 – Сайт SendPulse

Приклад використання: впровадження чат-боту для надання інформації студентам про події в університеті та допомоги з питаннями щодо навчання. Бот може відповідати на питання про розклад, події, допомагати вирішувати адміністративні питання та надавати загальну підтримку. [5]

Позитивні аспекти:

- Інтуїтивний інтерфейс для створення чат-ботів без необхідності великих технічних навичок.
- Зручна можливість інтеграції з іншими інструментами маркетингового стеку.
- Можливість розширення та масштабування функціоналу чат-боту з ростом потреб освітнього закладу.

Негативні аспекти:

- Може бути обмеженою для проєктів, які вимагають великої налаштованості та складного логіку.
- Вартість використання SendPulse може бути високою для деяких освітніх закладів, зокрема для повного спектру їхніх послуг.

IBM Watson Assistant for Education - це інструмент, який використовує технологію штучного інтелекту для створення чат-ботів для освітніх закладів. Він забезпечує можливість персоналізованої комунікації та взаємодії зі студентами (рис. 2.5).

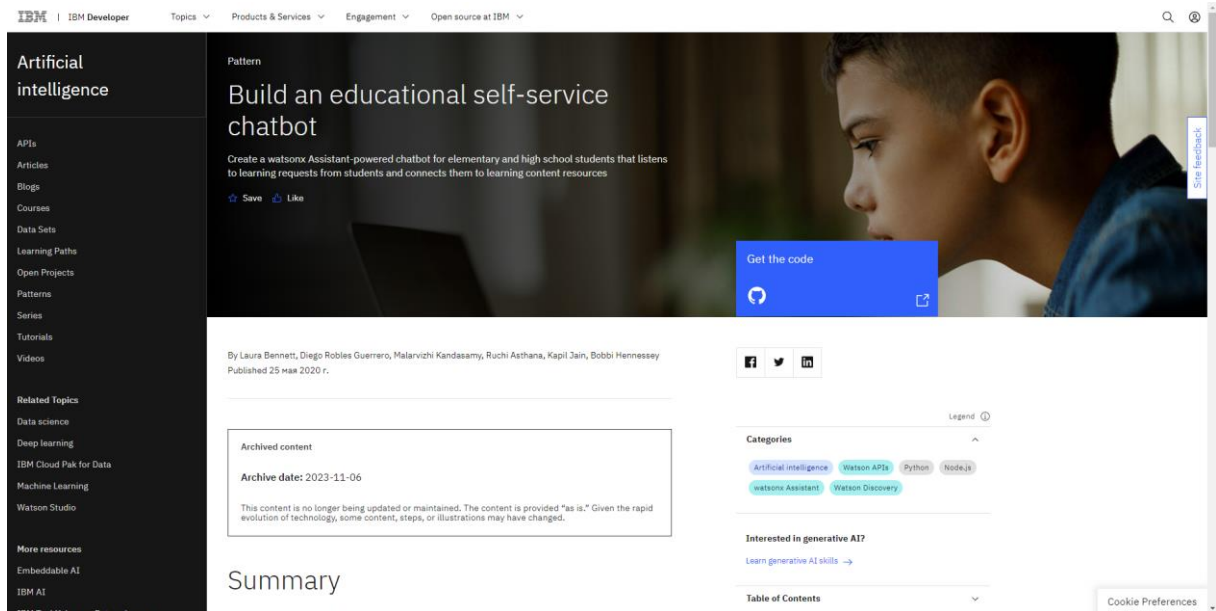


Рисунок 2.5 – Сайт IBM Watson Assistant for Education

Позитивні аспекти:

- Великий набір інструментів для налаштування та персоналізації чат-боту під потреби конкретного навчального закладу.
- Здатність інтеграції з іншими системами.

Негативні аспекти:

- Висока складність конфігурації для неспеціалізованих користувачів.
- Вартість використання послуг IBM Watson може бути високою для деяких освітніх закладів.

Google Dialogflow for Education - інструмент для створення розумних чат-ботів, які можуть взаємодіяти зі студентами та надавати допомогу у навчальному процесі (рис. 2.6). [6]

Позитивні аспекти:

- Легкість використання та інтуїтивний інтерфейс для створення різноманітних діалогів та відповідей.
- Підтримка інтеграції з різними платформами, такими як Facebook Messenger, Slack, тощо.

Негативні аспекти:

- Обмежена можливість адаптації для вирішення конкретних завдань університету.
- Вартість використання великої кількості функцій може бути високою.

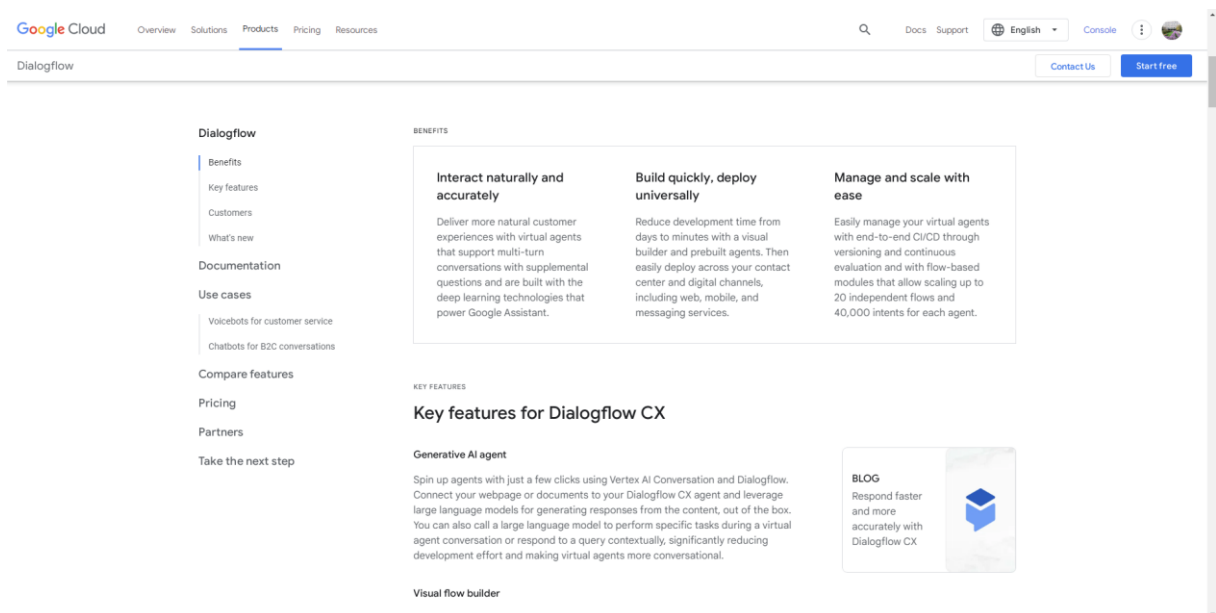


Рисунок 2.6 – Сайт Google Dialogflow

2.2. Аналіз існуючих рішень у сфері довідкових чат-ботів для освітніх закладів

Загальний аналіз того, які функціональності можуть бути реалізовані в чат-ботах для освітніх цілей:

- бот для відповідей на запитання;

- бот для підтримки навчання;
- чат-бот для адміністративних послуг;
- орієнтаційний бот для новачків;
- бот для залучення до заходів та клубів;
- чат-бот для зарахування та вступу.

Бот для відповідей на запитання – надання студентам можливості отримати швидкі та точні відповіді на запитання щодо розкладу, викладачів, адміністративних послуг тощо.

Приклад: студент може запитати боту: «Які заняття у мене завтра?» або «Як зв'язатися з відділом реєстрації?»

Бот для підтримки навчання – допомога студентам з навчанням, наприклад, надання пояснень до певних тем, виконання тестів, наведення рекомендацій для підготовки.

Приклад: студент може запитати: «Як вирішити цей математичний розрахунок?» або «Є які-небудь рекомендації для вивчення історії?»

Чат-бот для адміністративних послуг – спрощення процесів, пов'язаних з адміністративною роботою, такими як реєстрація на курси, оплата навчання, заявки на стипендії тощо.

Приклад: студент може запитати: «Як подати заявку на стипендію?» або «Як оплатити вартість навчання?»

Орієнтаційний бот для новачків – допомога новим студентам орієнтуватися в університетському середовищі, забезпечення інформацією про відділи, послуги та ресурси.

Приклад: студент може запитати: «Де розташована бібліотека?» або «Які спортивні заходи відбуваються цього тижня?»

Бот для залучення до заходів та клубів – інформування студентів про події, зустрічі та можливості для участі в студентських клубах та інших заходах.

Приклад: студент може запитати: «Як приєднатися до клубу програмістів?» або «Який захід сьогодні відбудеться?»

Чат-бот для зарахування та вступу – надання інформації про вступний процес, необхідні документи, терміни та відповіді на запитання про зарахування.

Приклад: абітурієнт може запитати: «Які документи потрібні для вступу?» або «Які терміни подачі документів?»

Ці приклади вказують на різноманітні можливості використання чат-ботів у сфері освіти для полегшення комунікації та підтримки студентів та абітурієнтів.

Університет ПУЕТ створив зручну Telegram-групу під назвою «Загальний чат ПУЕТ». Ця група призначена для спрощення взаємодії та спілкування всіх учасників університетської спільноти (рис. 2.7). [7]

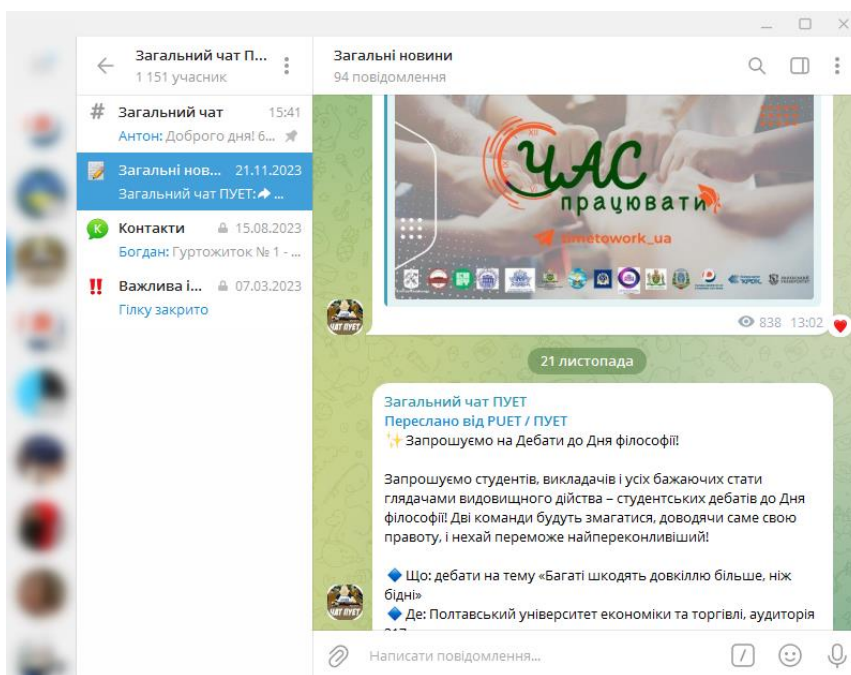


Рисунок 2.7 – Telegram-група «Загальний чат ПУЕТ»

У групі реалізовані різні гілки, які надають користувачам можливість швидко знаходити та обговорювати потрібну інформацію. Зокрема:

- Загальний чат – ця гілка призначена для вільного спілкування між студентами, викладачами та іншими членами університетської

громади. Тут можна обговорювати різноманітні теми, ділитися досвідом та задавати питання.

- Загальні новини – у цій гілці представлені офіційні новини та оголошення університету. Студенти можуть швидко отримувати інформацію про важливі події, оновлення та інші аспекти життя університетської спільноти.

- Контакти – гілка містить корисні контактні дані відділів та служб університету. Це дозволяє студентам швидко знаходити необхідні контакти для звернення за допомогою чи отримання інформації з різних сфер діяльності університету.

Такий підхід до організації чат-групи в Telegram дозволяє університету ПУЕТ забезпечити зручну та ефективну комунікацію всередині своєї спільноти, надаючи різні функціональні гілки для різноманітних потреб користувачів.

Для зручності та ефективності вступного процесу був розроблений спеціальний чат-бот, який стає невід'ємним помічником для абітурієнтів на кожному етапі їхнього вступного шляху. Цей чат-бот висвітлює різні аспекти вступу та надає інформацію, що сприяє прийняттю обдуманих рішень (рис. 2.8).

Серед основних розділів чат-боту виділяються: [7]

- Навчання – розділ, який детально описує освітні програми, факультети та можливості для студентів університету ПУЕТ.

- Про ПУЕТ – інформація про історію та цінності університету, його місію та особливості, які роблять його привабливим для майбутніх студентів.

- Вступна кампанія 2023 – актуальні новини та події, пов'язані з вступною кампанією поточного року, щоб абітурієнти завжди були в курсі подій.

- ЗНО (НМТ) – інформація про вимоги до ЗНО та конкретні вимірювання для напрямів.

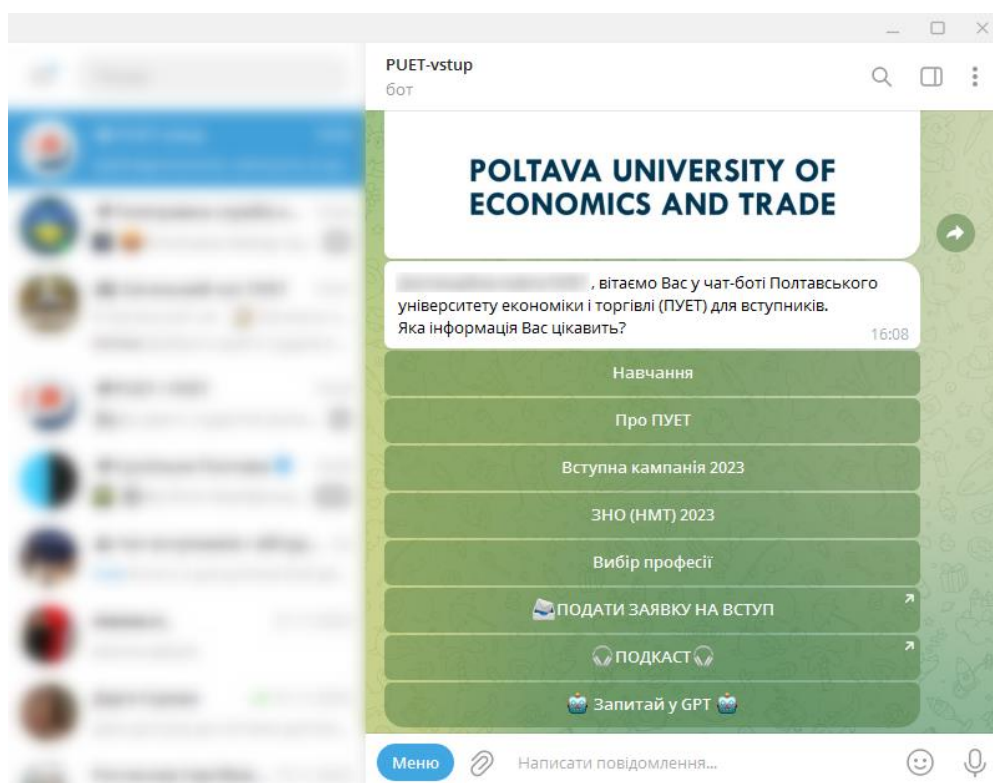


Рисунок 2.8 – Чат-бот ПУЕТ для вступників

- Вибір професії – поради та рекомендації стосовно вибору професії, підходящої індивідуальним інтересам та здібностям абітурієнта.
- Подання заявки на вступ – посилання для успішного подання заявки на вступ до університету.
- Генератор мотиваційного листа – інструмент для створення власного мотиваційного листа, який допомагає абітурієнтам висловити свої мети та мрії.
- Тренажер (НМТ) – розділ, який пропонує тренажер для підготовки до незалежного оцінювання з історії, математики та української мови.
- Контакти приймальної комісії – контактна інформація для отримання допомоги та консультацій від приймальної комісії університету.

- Подкасти – доступ до аудіо-матеріалів, які допомагають абітурієнтам отримати додаткову інформацію та враження про університет.

Цей чат-бот створений з метою забезпечити абітурієнтам все необхідне для успішного вступу, спрощуючи інформаційний доступ та допомагаючи зробити правильний вибір.

2.3. Вивчення технологій, які використовуються при розробці чат-ботів

Розробка чат-ботів вимагає глибокого розуміння різних технологій та інструментів, які допомагають створювати ефективні та функціональні рішення. Ось детальне вивчення ключових технологій, що використовуються при розробці чат-ботів: [3,4]

1. Мови програмування.

Python, JavaScript, Java, C#: ці мови програмування часто використовуються для створення чат-ботів. Python зазвичай відзначається своєю простотою та широкою підтримкою, а JavaScript використовується для веб-ботів.

2. Фреймворки для розробки чат-ботів.

Microsoft Bot Framework, Dialogflow, Botpress: ці фреймворки надають інструменти та ресурси для створення, розгортання та управління чат-ботами. Вони також можуть мати вбудовані інтеграції з популярними месенджерами.

3. Natural Language Processing (NLP).

NLTK, SpaCy, Rasa NLU: використання NLP дозволяє ботам розуміти та обробляти природну мову. Це включає в себе розпізнавання інтентів, аналіз настрою, визначення ключових слів та інше.

4. Методи Machine Learning (ML).

TensorFlow, PyTorch: в ML використовують для навчання чат-ботів розпізнавати та адаптуватися до нових паттернів взаємодії. Вони можуть покращувати відповіді боту та забезпечувати більшу розуміючість.

5. Бази даних.

MongoDB, MySQL, SQLite: чат-боти можуть використовувати бази даних для зберігання користувацької інформації, історії чату та інших даних.

6. APIs та Webhooks.

RESTful API, WebSockets: для інтеграції з іншими системами та послугами, чат-боти використовують APIs та Webhooks.

7. Frontend та UI/UX.

React, Vue, Angular: якщо чат-бот використовується на веб-сайті, важливо враховувати фронтенд технології для створення зручного та естетичного інтерфейсу.

8. Security та Authentication.

OAuth, JWT: для захисту конфіденційності та автентифікації користувачів використовуються технології такі як OAuth та JWT.

9. Cloud Services.

AWS, Azure, Google Cloud: розгортання чат-ботів у хмарних сервісах дозволяє швидко масштабувати та забезпечувати доступність.

10. Continuous Integration/Continuous Deployment (CI/CD).

Jenkins, Travis CI: для автоматизації та оптимізації процесу розгортання та оновлення чат-ботів.

Вивчення цих технологій допомагає розробникам створювати чат-боти, які ефективно взаємодіють з користувачами та відповідають на їхні потреби.

2.4. Необхідність та актуальність теми роботи

Необхідність та актуальність теми роботи «Розробка довідкового чат-боту для студентів» обумовлені кількома факторами:

1. Зростання обсягу інформації: сучасні студенти стикаються з величезним обсягом інформації, пов'язаною з навчанням, адміністративними питаннями та життєвими потребами. Чат-бот може допомогти їм швидко та ефективно знаходити потрібну інформацію без витрати часу на пошук.

2. Підвищення ефективності комунікації: чат-боти можуть стати універсальними інструментами для забезпечення швидкого та зручного спілкування між студентами та університетськими службами. Це спростить процеси запитань і відповідей, підтримки та координації.

3. Підтримка дистанційного навчання: зростання популярності дистанційного навчання підкреслює потребу у зручних та доступних інструментах для отримання допомоги та інформації від університету, що може бути надано чат-ботом.

4. Підвищення конкурентоспроможності університету: університети, які інвестують у розробку технологій для полегшення студентського життя, демонструють свою інноваційність та готовність до впровадження сучасних рішень. Це може позитивно вплинути на привабливість університету для майбутніх студентів.

5. Підтримка: довідковий чат-бот може надати підтримку, враховуючи індивідуальні потреби та запити студентів. Це дозволить створити більш дружнє та ефективне середовище навчання, де кожен студент може отримати інформацію та допомогу, яка відповідає його потребам.

6. Підвищення доступності: чат-бот може бути доступним 24/7, що дозволить студентам отримати потрібну інформацію в будь-який час,

навіть під час нічних годин або у вихідні дні. Це особливо важливо для студентів з різних часових поясів або зі змінним робочим графіком.

7. Підтримка адаптивності: чат-бот може швидко адаптуватися до змін у навчальному процесі, адміністративних процедурах та вимогах, що забезпечить стабільну та надійну підтримку студентів навіть у змінних умовах.

Отже, розробка довідкового чат-боту для студентів є необхідною та актуальною, оскільки вона відповідає потребам сучасного університетського середовища, сприяючи покращенню доступності, ефективності та персоналізації навчання.

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Алгоритм роботи чат-боту

Чат-бот вітає студента та запитує, яку категорію інформації він шукає: «Привіт! Як я можу вам сьогодні допомогти? Виберіть категорію:». Пропонуються такі пункти: [2, 5]

1. Інформація щодо вступу;
2. Навчання;
3. Технічна підтримка дистанційного навчання;
4. Оплата навчання, гуртожитку тощо;
5. Отримання довідки/документів;
6. Контакти кафедр/викладачів;
7. Проходження практики;
8. Контакти всіх відділів/підрозділів;
9. Докладніше про кожну категорію.

Після отримання потрібної інформації, чат-бот запитує, чи є ще які-небудь питання чи як він може допомогти далі: «Сподіваюсь ви знайшли інформацію, що вас цікавить. Якщо ні, то подивіться Докладніше про кожну категорію – відобразиться короткий опис і типові запитання».

1. Інформація щодо вступу.

Короткий опис: посилання на сайт вступника.

Типові запитання: «Які вимоги до вступу?», «Як подати документи?», «Коли відбудеться вступна кампанія?».

2. Навчання.

Короткий опис: деталі щодо навчального процесу, розкладу занять та інших аспектів навчання.

Типові запитання: «Які предмети входять до вибіркових?», «Коли проходить сесія?», «Де можна переглянути розклад?».

3. Технічна підтримка дистанційного навчання.

Короткий опис: інструкції щодо логіну/паролю, дистанційних курсів, відеозв'язку тощо під час дистанційного навчання.

Типові запитання: «Як підключитися до онлайн лекцій?», «Що робити коли забув логін і пароль?», «Чому виникли проблеми з дистанційним курсом?».

4. Оплата навчання, гуртожитку тощо.

Короткий опис: інформація про фінансові питання, способи оплати та інші деталі.

Типові запитання: «Як відбувається оплата за навчання?», «Як оплатити проживання в гуртожитку?», «Які реквізити на оплату?».

5. Отримання довідки/документів.

Короткий опис: інструкції щодо отримання довідок та документів від університету.

Типові запитання: «Як отримати довідку про навчання?», «Як отримати студентський квиток?», «Як отримати дублікати документів про освіту?».

6. Контакти кафедр/викладачів.

Короткий опис: інформація про контакти викладачів та кафедр для звернення з питаннями чи консультаціями.

Типові запитання: «Які контакти викладача з дисципліни?», «Як зв'язатися з кафедрою для консультації?».

7. Проходження практики.

Короткий опис: посилання на центр зв'язків з виробництвом.

Типові запитання: «Як організовано проходження практики?», «Як знайти місце для стажування?».

8. Контакти всіх відділів/підрозділів.

Короткий опис: посилання на організаційну структуру університету, де зазначені всі контакти.

При виборі категорії виводяться відповідні пункти або виконується прописана дія.

1. Інформація щодо вступу.

Пункти:

1.1.Галузі знань та спеціальності – посилання на сайт

<https://vstup.puet.edu.ua/#specs>;

1.2.Все про ЗНО/НМТ – посилання на сайт

<https://vstup.puet.edu.ua/pidgotovka-do-zno-2-2/>;

1.3.Бакалаврат:

1.3.1. На базі повної загальної середньої освіти – посилання на

сайт <https://vstup.puet.edu.ua/bakalavr-pislya-zagalnoosvitnoyi-shkoly/>;

1.3.2. На базі диплома фахового мол. бакалавра, мол. бакалавра, мол. спеціаліста – посилання на сайт

<https://vstup.puet.edu.ua/na-bazi-dyplomu-molodshogo-spetsialista/>.

1.4.Магістратура – посилання на сайт

<https://vstup.puet.edu.ua/magistr/>;

1.5.Друга вища освіта (Магістратура) – посилання на сайт

<https://vstup.puet.edu.ua/vstupna-kampaniya/suchasna-vyshha-osvita-bez-zno/>;

1.6.Аспірантура / Докторантура – посилання на сайт

<https://vstup.puet.edu.ua/aspirantura-doktorantura/>;

1.7.Військова підготовка – посилання на сайт

<https://vstup.puet.edu.ua/906-2/>.

2. Навчання.

Пункти:

2.1.Питання щодо навчання, сесії:

2.1.1. Денна форма навчання – повідомлення:

«Навчально-науковий інститут денної освіти

Кабінет. 236, 237

Телефон: +38 (066) 474-45-79

Детальніше:<http://puet.edu.ua/institutes/navchalno-naukovyj-institut-dennoyi-osvity/>»

2.1.2. Заочна форма навчання – повідомлення:

«Навчально-науковий інститут заочно-дистанційного навчання

Кабінет. 222, 222а, 217

Телефон: +38 (097) 062-28-66

Пошта: dekanatzaocnyj@gmail.com

Детальніше:<http://puet.edu.ua/institutes/navchalno-naukovyj-institut-zaочно-dystanczijного-navchannya/>»

2.2. Вибіркові дисципліни:

2.2.1. Загальна інформація – посилання на сайт

<http://puet.edu.ua/vibirkovi-disciplini/>;

2.2.2. Вибір/зміна вибірових дисциплін – повідомлення:

«Навчально-науковий центр забезпечення якості вищої освіти

Кабінет. 203

Телефон: (0532) 56-05-49

Пошта: vuod@puet.edu.ua»

2.3. Розклад занять – посилання на сайт

<https://schedule.puet.edu.ua/>;

2.4. Персональний кабінет – посилання на сайт

<https://cabinet.puet.edu.ua/>;

2.5. Сайт дистанційного навчання – посилання на сайт

<https://el.puet.edu.ua/>;

2.6. Навчальні плани освітніх програм – посилання на сайт

<http://puet.edu.ua/osvita/organizaciya-osvitnogo-procesu/navchalni-plany-osvitnih-program/>;

2.7. Освітні програми – посилання на сайт

<https://puet.edu.ua/osvita/organizaciya-osvitnogo-procesu/osvitno-profesijni-programy/>;

2.8. Графіки освітнього процесу – посилання на сайт

<http://puet.edu.ua/osvita/organizaciya-osvitnogo-procesu/grafiky-osvitnogo-proczesu/>.

3. Технічна підтримка дистанційного навчання.

Пункти:

3.1. Інструкції по роботі в системі – посилання на сайт

<https://el.puet.edu.ua/help/>;

3.2. Головний центр дистанційного навчання – посилання на сайт

<https://el.puet.edu.ua/>;

3.3. Персональний кабінет – посилання на сайт

<https://cabinet.puet.edu.ua/>;

3.4. Втрата/зміна логіну і пароллю – повідомлення:

«З питань технічної підтримки сайту:

Кабінет. 409

Телефон: +38 (099) 740-56-88

Пошта: elarning.puet@gmail.com

Детальніше: <https://el.puet.edu.ua/contact-us/>»

3.5. Проблеми з дистанційним курсом/відеозв'язком – повідомлення:

«З питань технічної підтримки сайту:

Кабінет. 409

Телефон: +38 (099) 740-56-88

Пошта: elarning.puet@gmail.com

Детальніше: <https://el.puet.edu.ua/contact-us/>»

4. Оплата навчання, гуртожитку тощо.

4.1. Інформація щодо оплати – повідомлення:

«Фінансово-економічне управління

З питань розрахунків за навчання звертатись у кім. 100 або за телефонами: +38 (0532) 56-10-29; +38 (095) 848-24-70; +38 (068) 242-52-30

З питань розрахунків за проживання у гуртожитках, виплати стипендій звертатись у кім. 206 або за телефоном +38 (0532) 56-08-50

Пошта: puetglavbuh@gmail.com

Детальніше: <http://puet.edu.ua/other-divisions/finansovo-ekonomiche-upravlinnya/>»

4.2. Реквізити для оплати/Privat24 – посилання на сайт <https://next.privat24.ua/payments/form/%7B%22token%22:%221b69e41a3abb2d30cf5b6851636eeaebe9hx3tqg%22%7D>.

5. Отримання довідки/документів.

5.1. Довідка про навчання:

5.1.1. Денна форма навчання – повідомлення:

«Навчально-науковий інститут денної освіти

Кабінет. 236, 237

Телефон: +38 (066) 474-45-79

Детальніше: <http://puet.edu.ua/institutes/navchalno-naukovyj-instytut-dennoyi-osvity/>»

5.1.2. Заочна форма навчання – повідомлення:

«Навчально-науковий інститут заочно-дистанційного навчання

Кабінет. 222, 222а, 217

Телефон: +38 (097) 062-28-66

Пошта: dekanatzaocnyj@gmail.com

Детальніше: <http://puet.edu.ua/institutes/navchalno-naukovyj-instytut-zaochno-dystancijnogo-navchannya/>»

5.2. Довідка у ТЦК та СП – повідомлення:

«Відділ кадрів

Кабінет. 320

Телефон: +38 (0532) 56-05-28

Пошта: ok@puet.edu.ua

Детальніше: <http://puet.edu.ua/other-divisions/viddil-kadriv/>»

5.3. Студентський квиток, дублікати документів про освіту:

5.3.1. Денна форма навчання – повідомлення:

«Відділ студентської документації

Кабінет. 342

Телефон: +38 (0532) 50-92-06

Пошта: vsd@puet.edu.ua

Детальніше: <http://puet.edu.ua/other-divisions/viddil-studentskoyi-dokumentacziyi/>»

5.3.2. Заочна форма навчання – повідомлення:

«Навчально-науковий інститут заочно-дистанційного навчання

Кабінет. 222, 222а, 217

Телефон: +38 (097) 062-28-66

Пошта: dekanatzaocnyj@gmail.com

Детальніше: <http://puet.edu.ua/institutes/navchalno-naukovyj-instytut-zaочно-dystanczijного-navchannya/>»

6. Контакти кафедр/викладачів – посилання на сайт

<http://puet.edu.ua/kafedry/>.

7. Проходження практики – повідомлення:

«Центр зв'язків з виробництвом

Кабінет. 219

Телефон: +38 (0532) 52-09-32

Пошта: czv@puet.edu.ua

Детальніше:<http://puet.edu.ua/other-divisions/czentr-zvyazkiv-z-vyrobnyctvom/>»

8. Контакти всіх відділів/підрозділів – посилання на сайт <http://puet.edu.ua/organizaciyna-struktura/>.

3.2. Блок-схема алгоритму роботи чат-боту

Блок-схема алгоритму роботи чат-боту містить розділи, що надають інформацію щодо вступу, процесу навчання, технічної підтримки дистанційного навчання, оплати навчання та гуртожитку, отримання довідок/документів, контактів кафедр/викладачів, проходження практики та контактів відділів університету.

Нижче подано блок-схему роботи алгоритму:

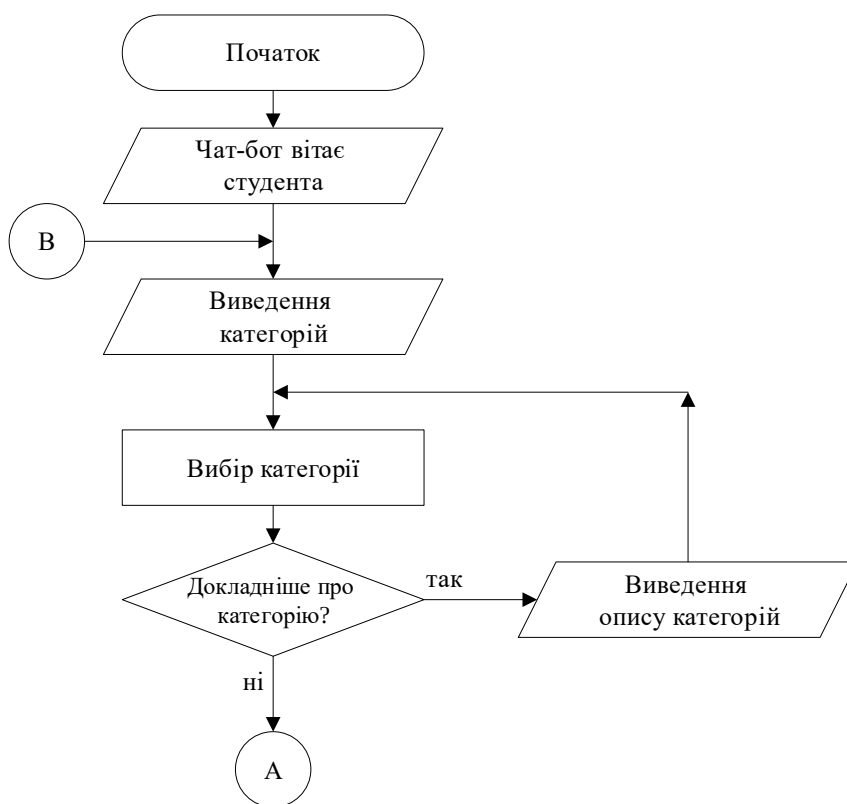


Рисунок 3.1 – Блок-схема алгоритму роботи чат-боту

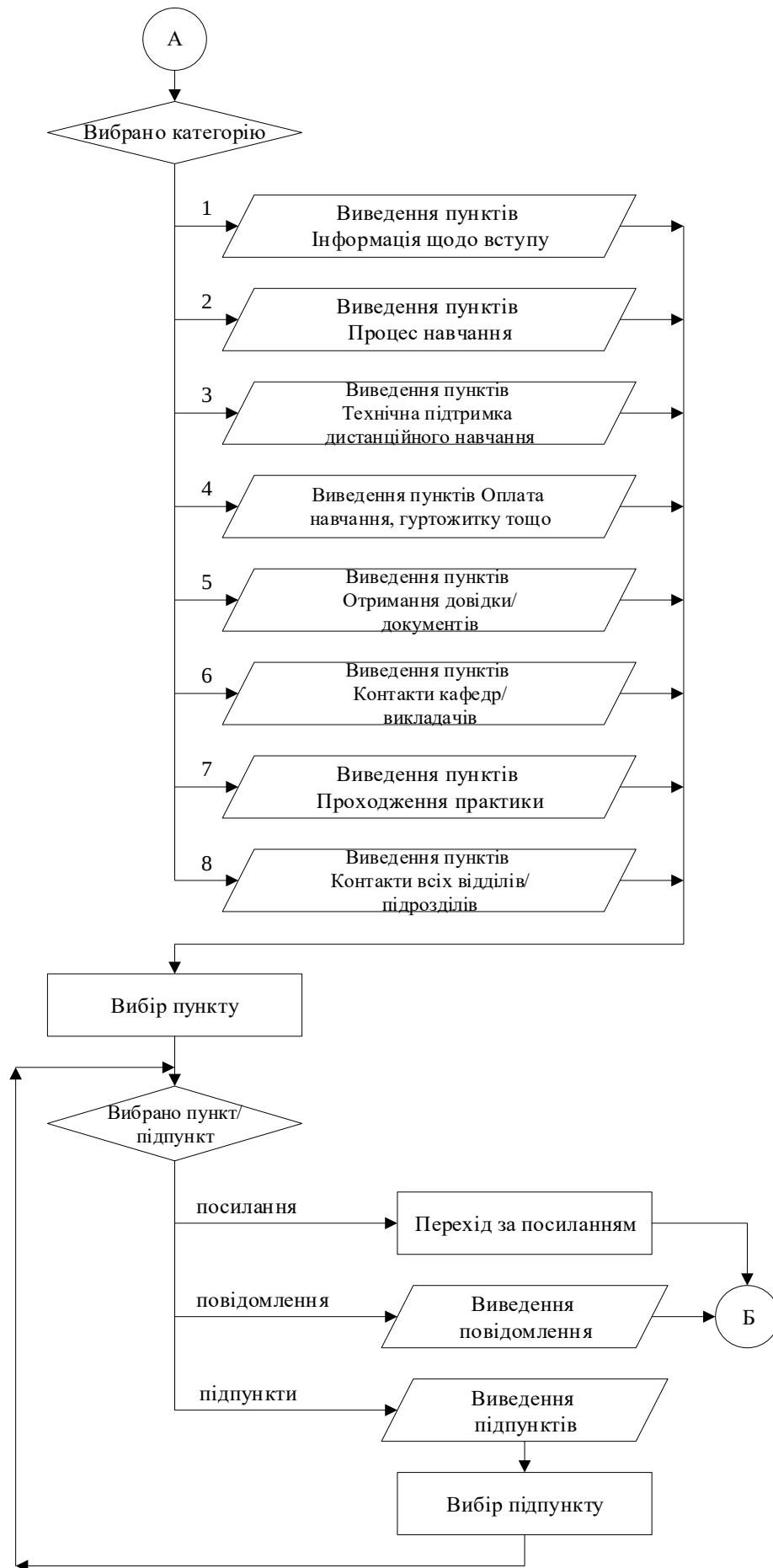


Рисунок 3.2 – Продовження блок-схеми алгоритму роботи чат-боту

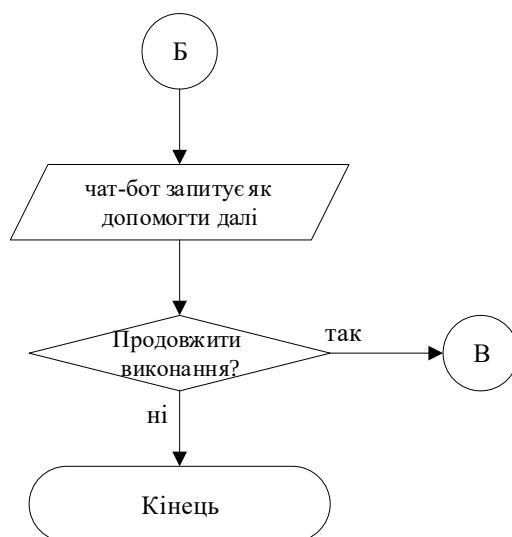


Рисунок 3.3 – Продовження блок-схеми алгоритму роботи чат-боту

3.3. Обґрунтування вибору програмного забезпечення та методів

Telegram BotFather - це офіційний бот Telegram, який дозволяє користувачам створювати та керувати своїми власними чат-ботами в Telegram. Ось деякі ключові функції та можливості Telegram BotFather:

- Створення нового бота:

BotFather дозволяє користувачам створювати нові чат-боти, надаючи їм унікальні імена та ідентифікатори.

- Налаштування параметрів бота:

Користувачі можуть налаштовувати різні параметри свого чат-боту, такі як його опис, аватар, команди, вітання, повідомлення про помилки тощо.

- Створення команд:

BotFather дозволяє користувачам додавати різноманітні команди для свого чат-боту, які можуть виконувати певні дії при введенні користувачем спеціальних ключових слів або команд.

- Управління правами доступу:

Користувачі можуть налаштовувати права доступу для свого чат-боту, визначаючи, хто має право взаємодіяти з ботом та які функції доступні для користувачів.

- Надання API ключа:

BotFather надає користувачам API ключ, який можна використовувати для інтеграції чат-боту з іншими сервісами або додатками.

- Управління статистикою та налаштуваннями:

BotFather також дозволяє користувачам переглядати статистику використання свого чат-боту, налаштовувати повідомлення про помилки, відключати чи включати бота та інші параметри керування.

У цілому, Telegram BotFather - це потужний інструмент, який робить процес створення та управління чат-ботами в Telegram простим та зручним.

Python є однією з популярних мов програмування для розробки чат-ботів завдяки своїй простоті, багатий виборі бібліотек та фреймворків для обробки природної мови (NLP) і можливості швидкої розробки програм (рис. 3.4).

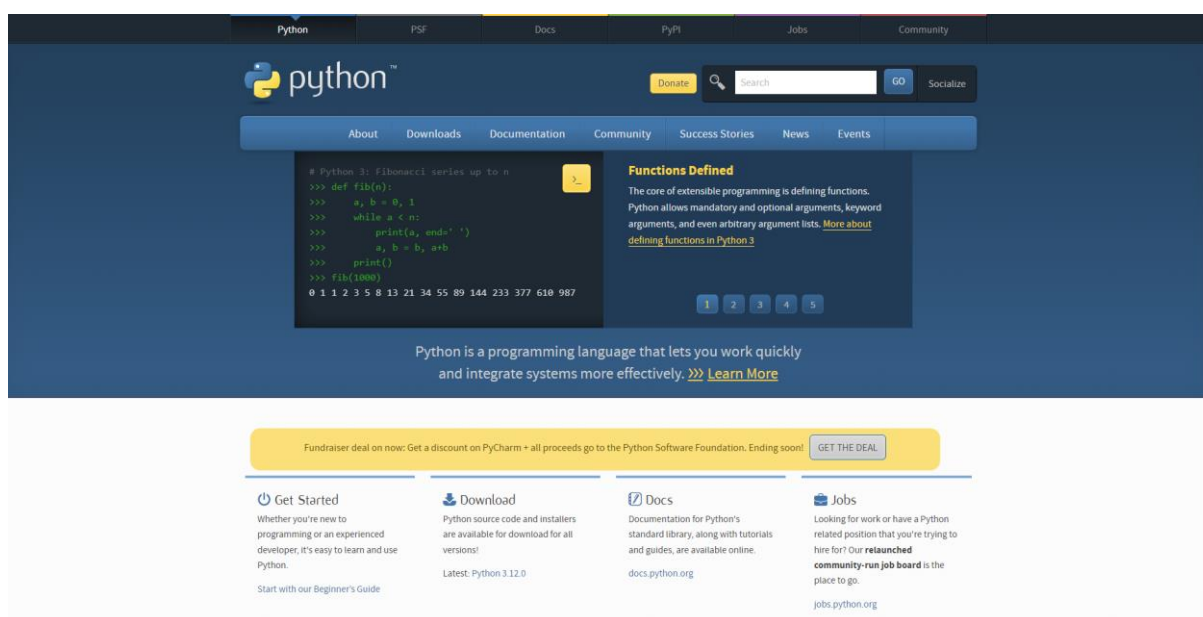


Рисунок 3.4 – Сайт Python

Ось кілька аспектів використання Python для створення чат-боту:

- Бібліотеки для роботи з NLP:

NLTK (Natural Language Toolkit) – це бібліотека для обробки природної мови, яка надає інструменти для роботи з текстом, включаючи токенизацію, лематизацію, аналіз тональності та інше.

sраСу – це бібліотека для обробки природної мови, яка дозволяє вам виконувати високошвидкісні аналізи тексту, такі як розпізнавання іменованих сутностей та синтаксичний аналіз.

- Фреймворки для створення чат-ботів:

ChatterBot – це бібліотека, яка дозволяє створювати прості чат-боти з використанням машинного навчання. Вона забезпечує можливість використання власних наборів даних для навчання чат-боту.

Rasa – це відкритий фреймворк для розробки розумних чат-ботів. Він має вбудовані моделі для розпізнавання мови і обробки природної мови.

- Веб-фреймворки:

Flask або Django – ці фреймворки для веб-розробки можна використовувати для створення веб-інтерфейсу для чат-боту. Flask є легковаговим і ідеально підходить для простих застосунків, тоді як Django надасть більше функціоналу для складніших застосунків.

- API для чат-платформ:

Telegram Bot API, Facebook Messenger API – якщо ви хочете створити чат-боту для певної платформи, такої як Telegram чи Facebook Messenger, використання відповідних API відповідаючих платформ є важливим. Для Telegram це python-telegram-bot, а для Facebook Messenger - pyMessenger.

- Обробка запитань і відповідей:

Моделі машинного навчання – використовуючи бібліотеки, такі як scikit-learn або TensorFlow, можна навчити моделі машинного навчання для кращого розпізнавання інтентів та створення більш ефективних відповідей на запитання.

- Сховище даних:

SQLite, MongoDB – для зберігання інформації про користувачів, історію чату та інші дані вам може знадобитися локальна база даних, така як SQLite або нереляційна база даних, наприклад, MongoDB.

- Веб-сервер:

uWSGI, Gunicorn – якщо ви використовуєте Flask або Django, вам слід використовувати веб-сервери, такі як uWSGI чи Gunicorn для обслуговування запитів. [6, 7]

Наведені інструменти дозволяють легко розпочати розробку чат-боту з використанням Python, незалежно від рівня ваших навичок в програмуванні чи розробці чат-ботів.

pyTelegramBotAPI - це бібліотека для роботи з API Telegram Bot в середовищі Python (рис. 3.5). За допомогою цієї бібліотеки ви можете легко створювати і взаємодіяти з чат-ботом для Telegram. [13]

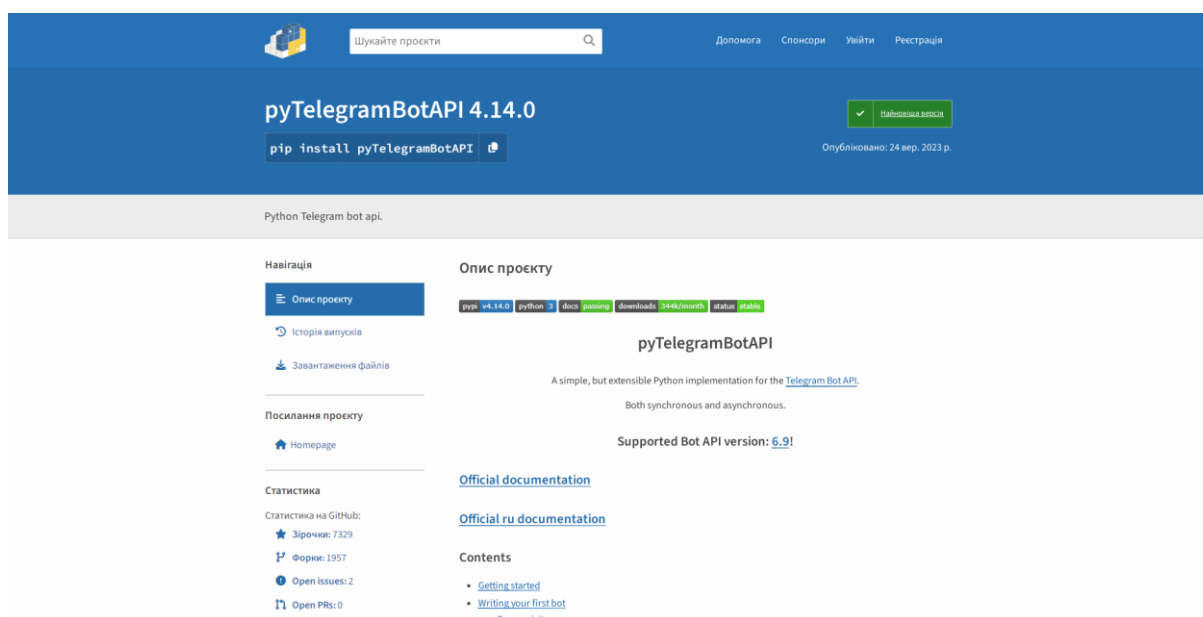


Рисунок 3.5 – Сайт pyTelegramBotAPI

Нижче наведено основні етапи використання pyTelegramBotAPI для створення простого чат-боту:

- Встановіть pyTelegramBotAPI, використовуючи pip:

```
pip install pyTelegramBotAPI
```

- Створіть новий файл Python та введіть наступний код для базового чат-боту:

```
import telebot
```

```
# Замініть 'YOUR_BOT_TOKEN' на фактичний токен вашого боту
bot = telebot.TeleBot('YOUR_BOT_TOKEN')
```

```
@bot.message_handler(func=lambda message: True)
```

```
def echo_all(message):
```

```
    bot.reply_to(message, message.text)
```

```
bot.polling()
```

- Запуск чат-боту:
 - Запустіть ваш Python-скрипт.
 - Ваш чат-бот готовий до роботи.
- Взаємодія з чат-ботом:
 - Перейдіть в чат з вашим чат-ботом в Telegram.
 - Надішліть будь-яке повідомлення.
 - Чат-бот поверне вам те саме повідомлення.
- Реалізація власної функціональності:
 - Додайте власні обробники повідомлень для створення специфічної функціональності чат-боту.
 - Наприклад, для відповіді на певні команди чи обробки зображень.
- Розгортання на сервері:
 - Якщо ви плануєте використовувати чат-бот у продакшні, розгорніть його на сервері з постійною доступністю. Для цього можна використовувати сервіси, такі як Heroku, AWS, чи будь-які інші хмарні рішення.

- Збереження конфіденційної інформації:

- Не розголошуйте свій токен боту та інші конфіденційні дані. Забезпечте безпеку доступу до вашого коду та сервера.

Ці кроки дозволяють вам створити базовий чат-бот для Telegram за допомогою `pyTelegramBotAPI`. При більш складних сценаріях вам може знадобитися докладніше вивчення документації та додаткових функцій цієї бібліотеки.

4. ПРАКТИЧНА ЧАСТИНА

4.1. Опис процесу створення та редагування чат-боту

Чат-бот — це автоматизований багатофункціональний помічник, який може збирати та показувати інформацію підписникам за їхнім запитом згідно зі заздалегідь підготовленим сценарієм. [14]

Основні етапи розробки чат-боту виглядати наступним чином:

1. Налаштування середовища розробки: включає встановлення Python, вибір IDE, реєстрацію бота на платформі Telegram, і так далі.
2. Створення логіки бота: включає розробку функціональності, яка буде реагувати на повідомлення від користувачів, обробляти їх та відправляти відповіді.
3. Тестування: важливий етап, коли ви перевіряєте, чи працює ваш бот наочно і відповідає на повідомлення коректно. Тестування також може включати в себе виявлення та усунення помилок.
4. Розгортання: після успішного тестування ви можете розгорнути свого бота на відповідному сервері, щоб він був доступний для користувачів.
5. Підтримка та вдосконалення: після розгортання ви можете отримувати зворотний зв'язок від користувачів і вдосконалювати бота, додавати новий функціонал, а також виправляти помилки.

Першим кроком потрібно відкрити месенджер Telegram, увійти до облікового запису або створити новий. Ввести у полі пошуку @BotFather і вибрати бот. У офіційного боту Telegram стоятиме синій знак підтвердження, біля імені у вигляді галочки (рис. 4.1).

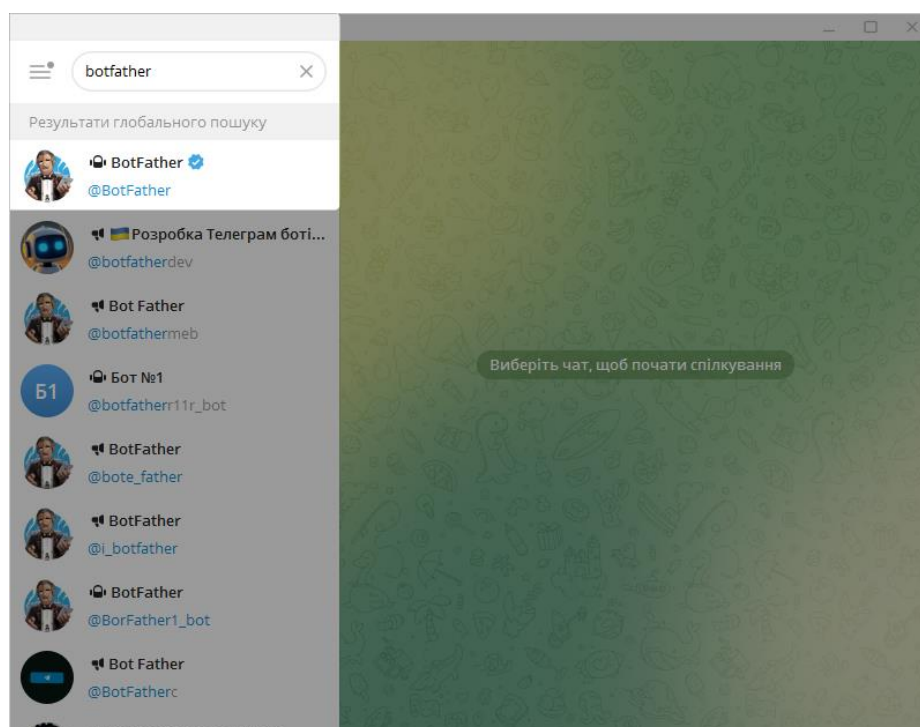


Рисунок 4.1 – Пошук @BotFather

Потім необхідно натиснути «Запустити» для активації боту BotFather (рис. 4.2). У відповідь відобразиться список команд з управління ботом.

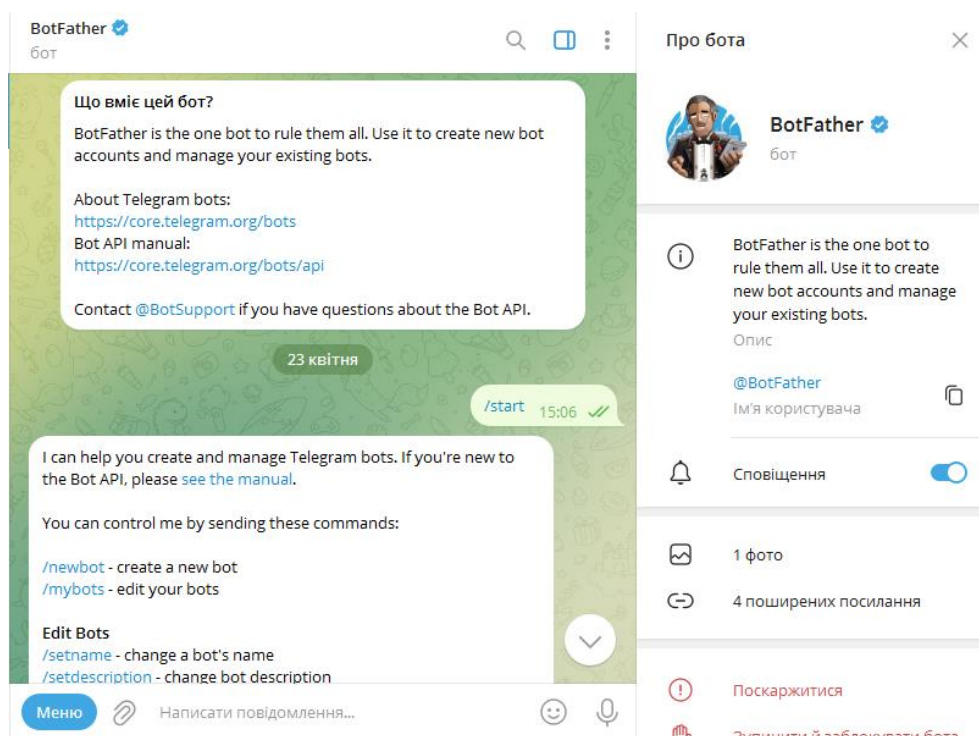


Рисунок 4.2 – Запуск боту BotFather

Тепер слід вибрати або надрукувати команду `/newbot` (рис. 4.3).

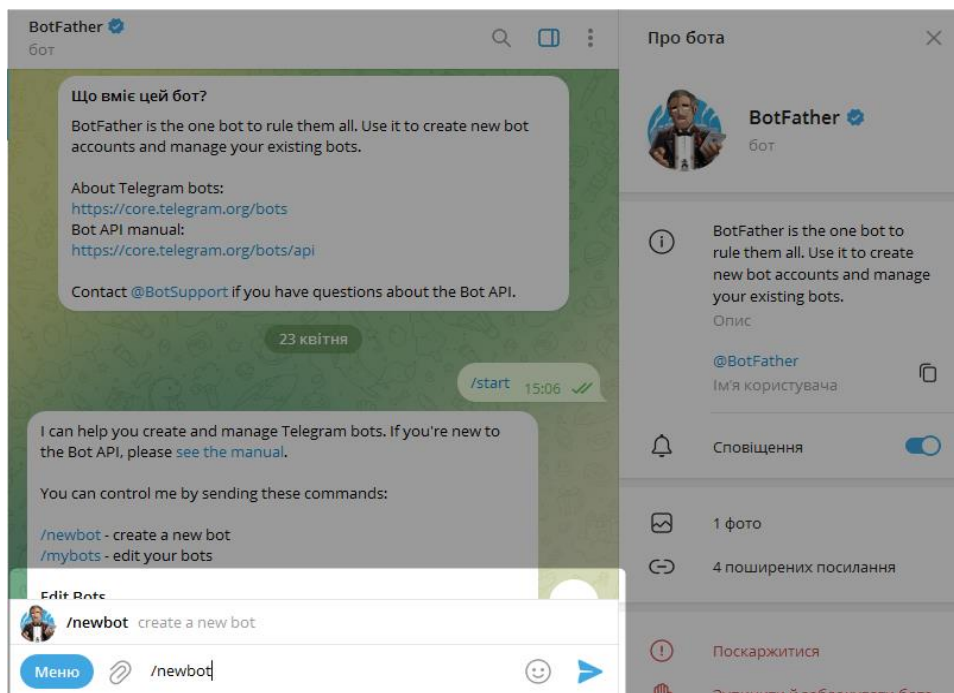


Рисунок 4.3 – Створення власного боту

Наступним кроком потрібно дати ім'я боту — клієнти побачать це ім'я під час спілкування з ботом. І нікнейм боту — за ним можна буде знайти боту в Telegram. Нікнейм має бути унікальним, не повторювати існуючі в базі та закінчуватися словом «bot» (рис. 4.4).

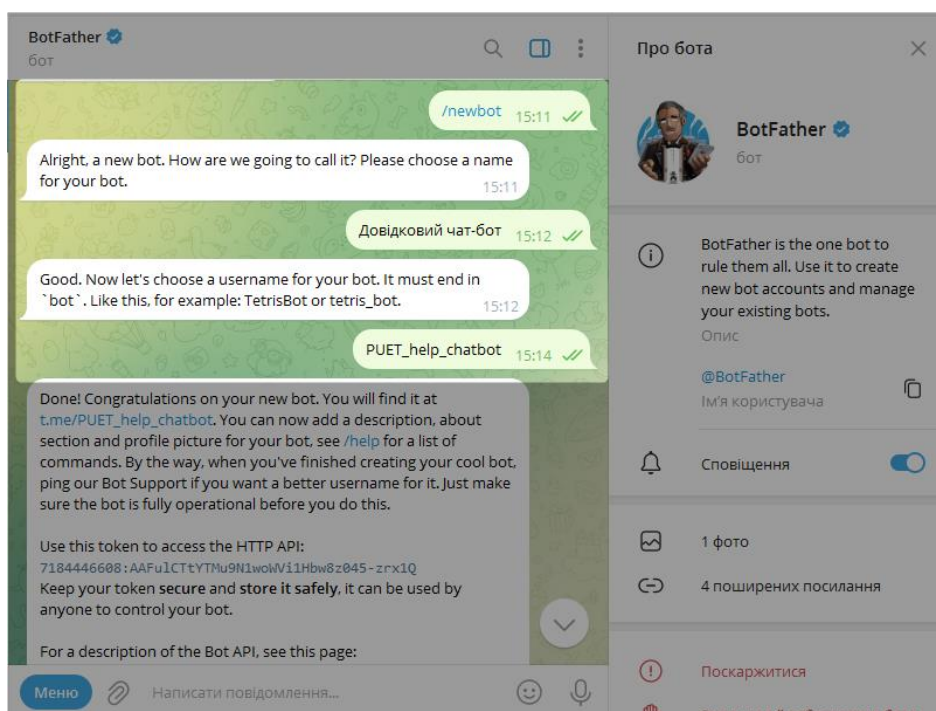


Рисунок 4.4 – Введення імені і нікнейму

Після цього бот буде створено. Виведеться повідомлення з посиланням на боту `t.me/<нікнейм_боту>`, рекомендації щодо налаштування аватарки, опис боту та список команд для його налаштування.

Для редагування боту знадобиться токен. Необхідно скопіювати значення токена та перейти до середовища розробки (рис. 4.5).

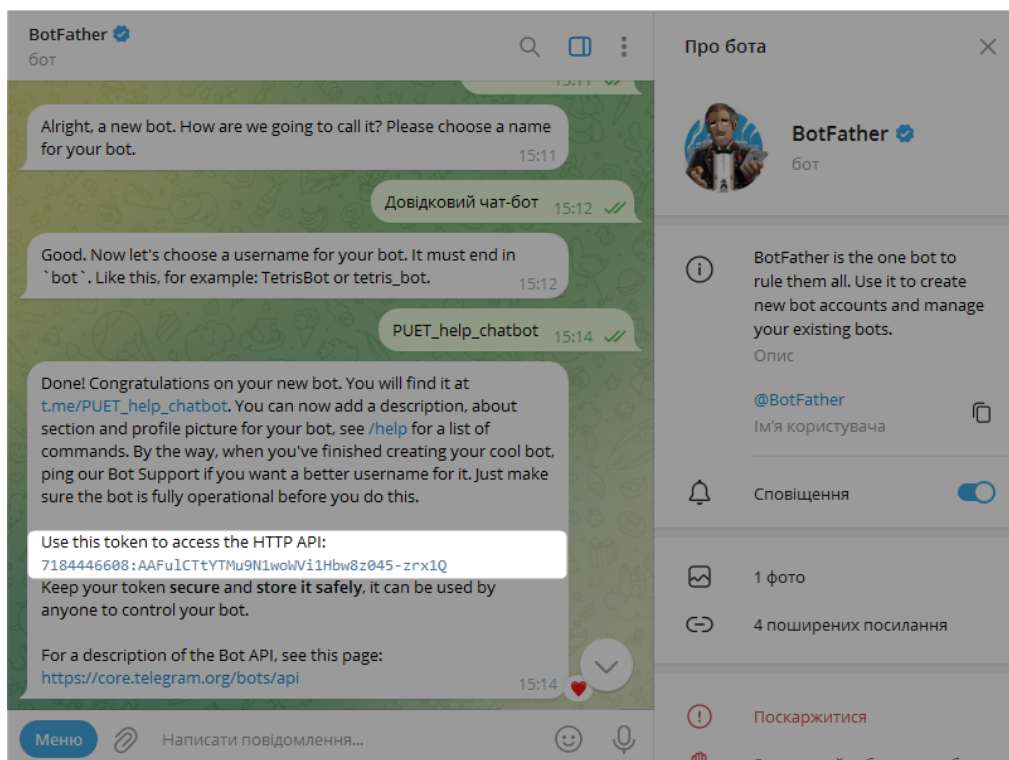


Рисунок 4.5 – Значення токена

Для знаходження ключа для вже існуючого робота потрібно перейти до боту @BotFather і ввести команду `/token` (рис. 4.6).

Потім відобразяться кнопки зі створеними ботами. Необхідно вибрати бот, який потрібно підключити. Тепер слід скопіювати значення токена (рис. 4.7).

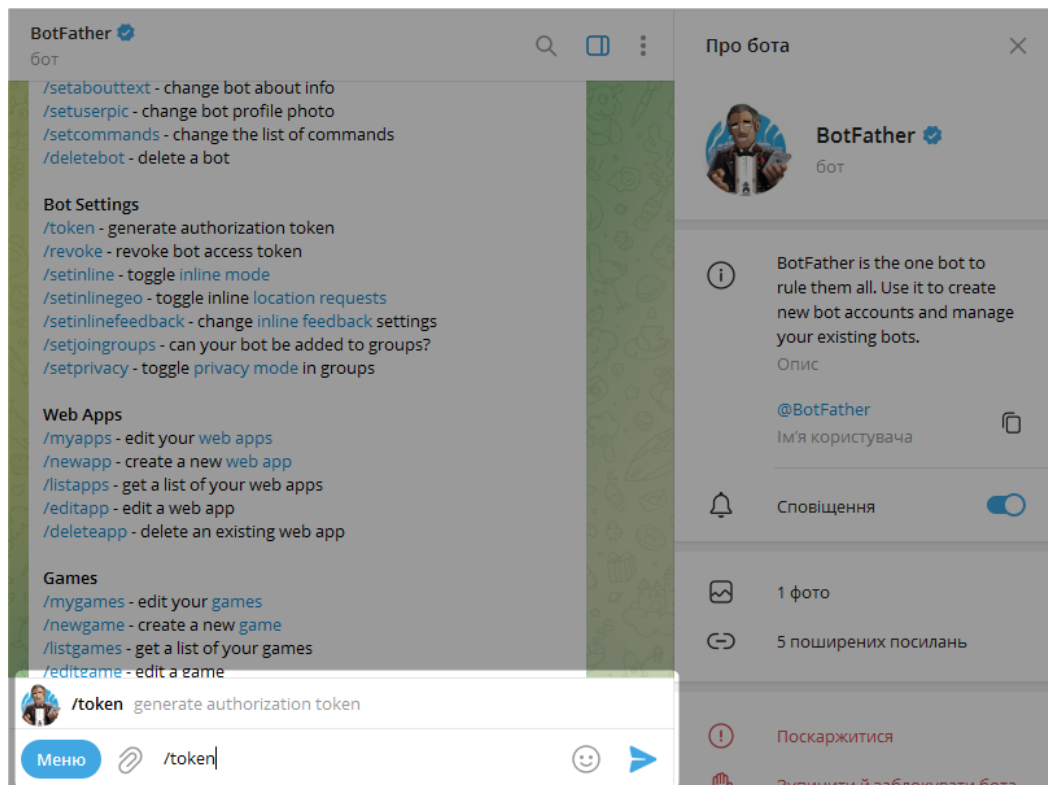


Рисунок 4.6 – Введення команди /token

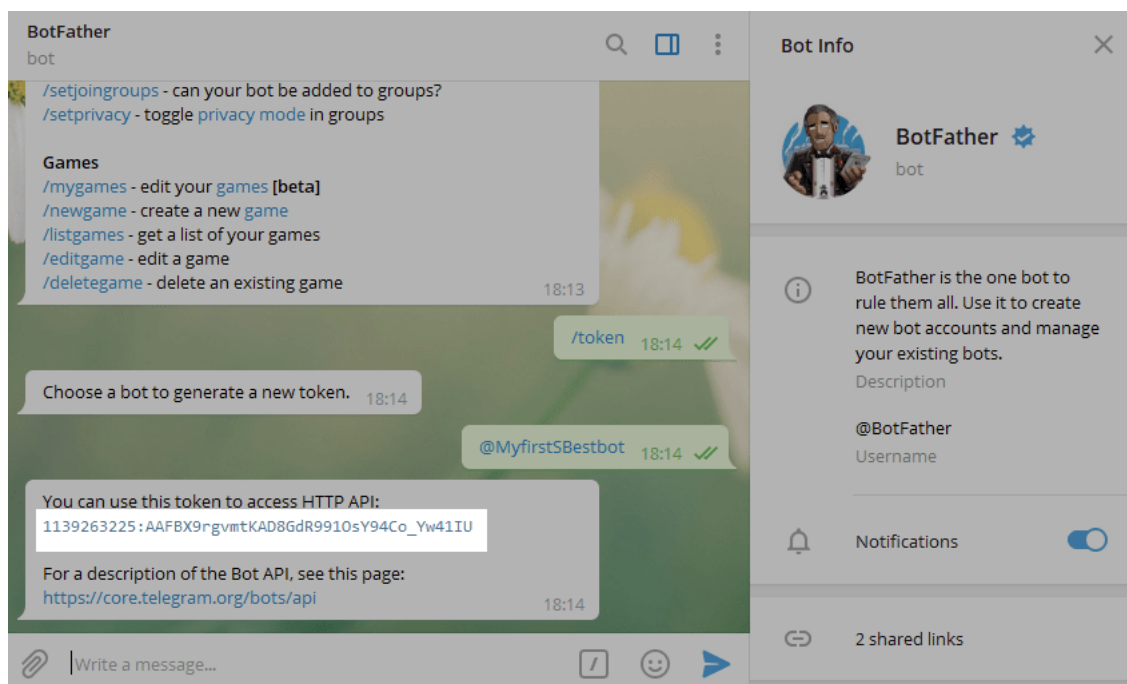


Рисунок 4.7 – Значення точена для використання

Наступним кроком можна налаштувати створений чат-бот.

Щоб встановити зображення профілю боту потрібно написати команду /setuserpic, вибрати бот і завантажити картинку (рис. 4.8).

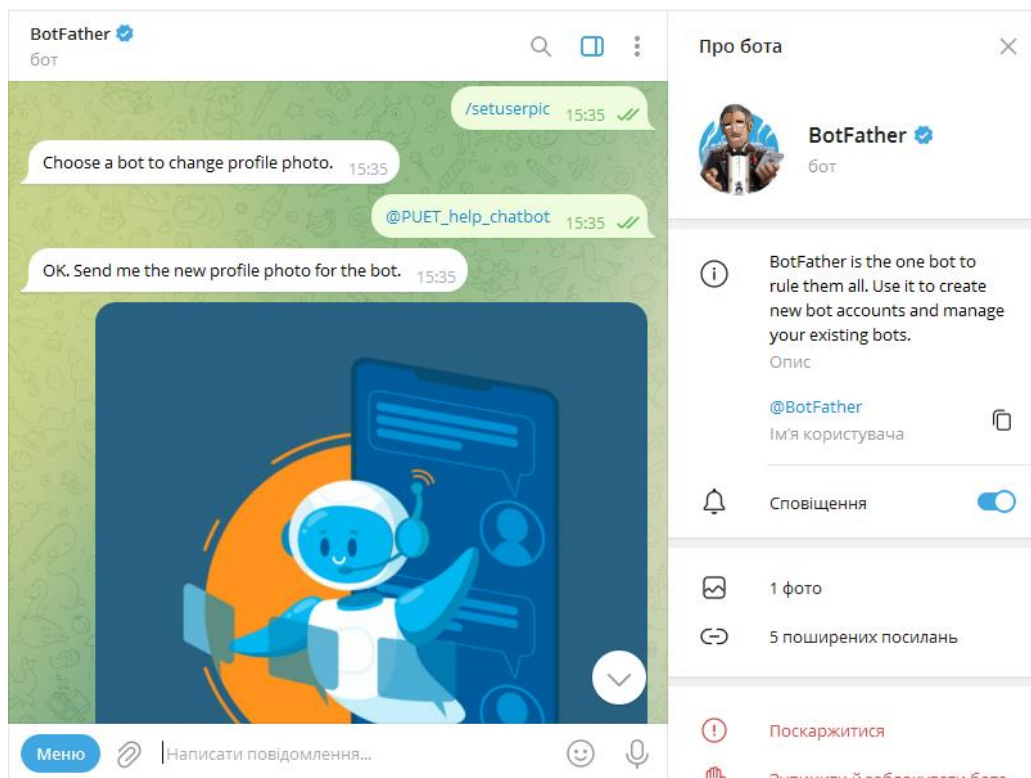


Рисунок 4.8 – Встановлення зображення профілю бота

Також вказуємо новий текст «Про бота». Люди побачать цей текст на сторінці профілю бота, і він буде надісланий разом із посиланням на бота, коли вони поділяться ним з кимось. Для цього використовуємо команду /setabouttext. Слід зауважити, що максимальна кількість символів – 120. (рис. 4.9)

Необхідно ще вказати новий опис для бота. Люди побачать цей опис, коли відкриють чат із ботом, у блоці під назвою «Що може робити цей бот?». Потрібно ввести команду /setdescription, вибрати бот і написати новий опис (рис. 4.10).

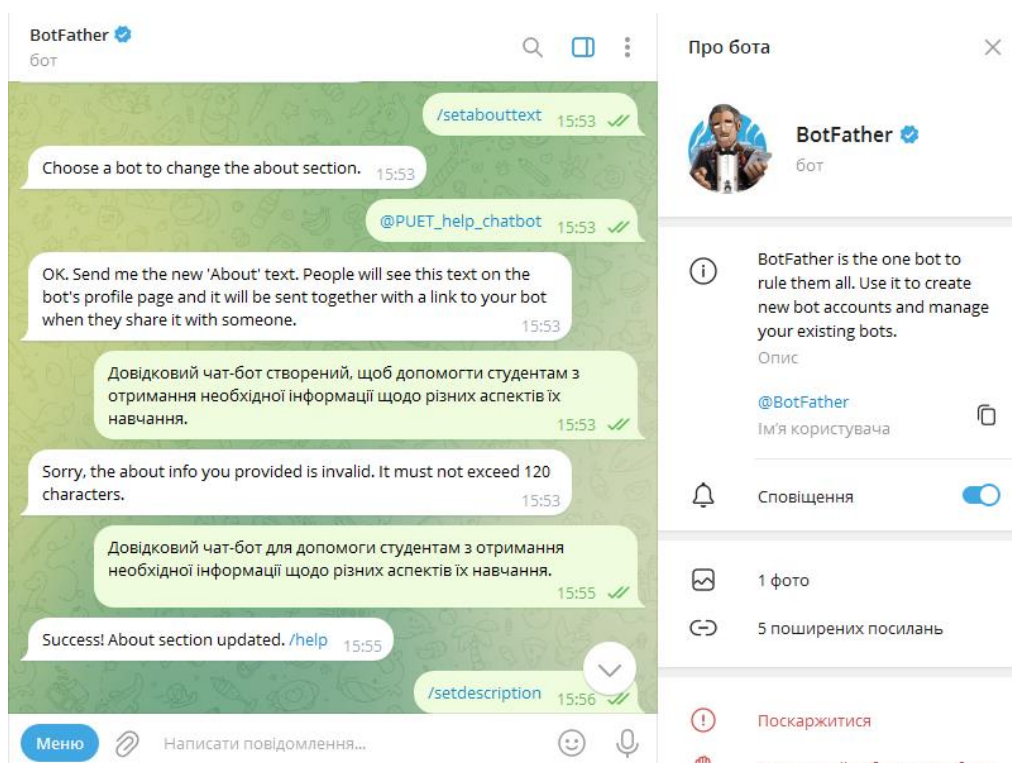


Рисунок 4.9 – Встановлення тексту «Про бота»

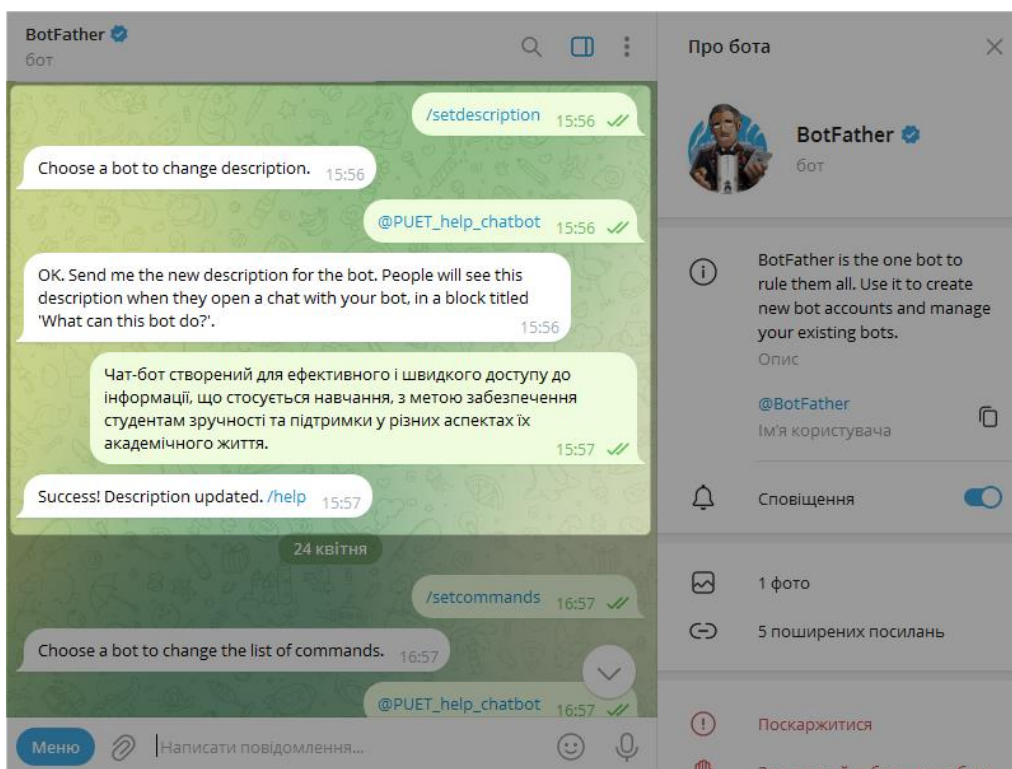


Рисунок 4.10 – Встановлення опису для бота

Ще одним кроком для зручності користування ботом можна вказати команди бота. Для цього необхідно ввести `/setcommands` та по вказаній

інструкції написати власні команди. Слід використовувати `/empty`, щоб список залишити список команд пустим (рис. 4.11).

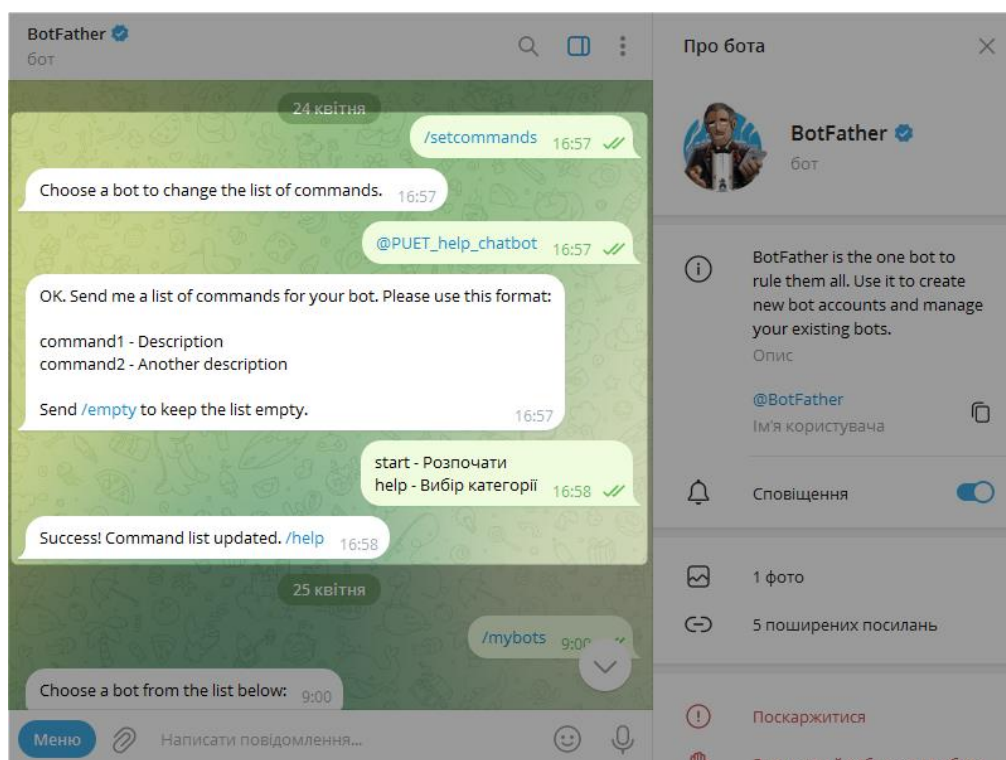


Рисунок 4.11 – Встановлення списку команд бота

4.2. Опис процесу програмної реалізації чат-боту

Для створення чат-боту за допомогою `pyTelegramBotAPI` із вказаною структурою, спершу слід встановити бібліотеку. [15]

```
pip install pyTelegramBotAPI
```

Далі можна написати код для обробки повідомлень та відправлення відповідей (див. Додаток А).

Спочатку потрібно встановити зв'язок між Python-скриптом і платформою Telegram за допомогою бібліотеки Telebot, використовуючи токен бота. Після встановлення цього зв'язку можна розробляти

функціонал чат-боту, який буде взаємодіяти з користувачами на платформі Telegram.

```
import telebot
from telebot import types
import time
```

```
bot = telebot.TeleBot('7184446608:AAFulCTtYTMu9N1woWVi1Hbw8z045-zrx1Q')
```

`import telebot` – Це імпортує бібліотеку Telebot, яка надає можливість створювати чат-боти для платформи Telegram за допомогою Python.

`from telebot import types` – Це імпортує модуль types з бібліотеки Telebot, який містить типи даних для створення різноманітних об'єктів, таких як клавіатури чи інлайн-відповіді.

`import time` – Це імпортує стандартний модуль часу, який може бути використаний для роботи з часом в програмі.

`bot = telebot.TeleBot('TOKEN')` – Ця строка створює екземпляр класу TeleBot з заданим токеном. Токен - це унікальний ідентифікатор бота на платформі Telegram.

Визначаємо декоратор з назвою `print_after_execution`, який додає функціонал до існуючих функцій. Декоратор використовується для виведення певного повідомлення після виконання команди, яка обернена даним декоратором.

```
def print_after_execution(func):
    def wrapper(*args, **kwargs):
        result = func(*args, **kwargs)
        message = args[0] if args else kwargs.get('message') # Отримуємо перше
повідомлення з аргументів функції
        chat_id = message.chat.id
        time.sleep(10)
        response = f"Сподіваюсь ви знайшли інформацію, що вас цікавить.\nЯкщо
ні, то подивіться <b>Докладніше про кожну категорію</b> /help – відобразиться
короткий опис і типові запитання."
        bot.send_message(chat_id, response, parse_mode="HTML")
        return result
    return wrapper
```

Давайте розберемо кожну частину коду:

`def print_after_execution(func)` – Це оголошення декоратора, який отримує функцію `func` як вхідний аргумент.

`def wrapper(*args, **kwargs)` – Ця вкладена функція визначається в межах декоратора. Вона приймає будь-яку кількість позиційних (`*args`) та ключових (`**kwargs`) аргументів. У цьому випадку, вона отримує аргументи команди, яку обертає декоратор.

`result = func(*args, **kwargs)` – Викликається оригінальна функція, яка була передана декоратору. Результат виклику зберігається у змінній `result`.

`message = args[0] if args else kwargs.get('message')` – Цей рядок отримує повідомлення, що було передано функції, яку декоруємо. Вона перевіряє, чи є аргументи (`args`): якщо так, вона бере перший аргумент як повідомлення. В іншому випадку, вона шукає ключ `'message'` в словнику `kwargs`.

`chat_id = message.chat.id` – Отримується `chat_id` з повідомлення, яке було отримано.

`time.sleep(10)` – Ця команда затримує виконання далі на 10 секунд. Це може бути використано для затримки відправлення відповіді на певний час.

`bot.send_message(chat_id, response, parse_mode="HTML")` – Відправляється відповідне повідомлення з використанням `bot.send_message` методу. Повідомлення містить текст `response`, а також вказано, що текст повинен розумітися як HTML за допомогою `parse_mode="HTML"`.

`return result` – Повертається результат виклику оригінальної функції.

Встановлюємо обробник повідомлень для команд `"/start"` і `"/help"` у чат-боті, використовуючи бібліотеку `Telebot`. Додавши декоратор `@print_after_execution`, вказується боту виконати певну дію після виконання цієї функції.

```

@bot.message_handler(commands=['start', 'help'])
@print_after_execution
def handle_start(message):
    markup = types.InlineKeyboardMarkup(row_width=1)
    item_1 = types.InlineKeyboardButton("Інформація щодо вступу",
callback_data="vstup")
    item_2 = types.InlineKeyboardButton("Навчання", callback_data="navchannya")
    item_3 = types.InlineKeyboardButton("Технічна підтримка дистанційного
навчання", callback_data="tehpitrymka")
    item_4 = types.InlineKeyboardButton("Оплата навчання, гуртожитку тощо",
callback_data="opлата")
    item_5 = types.InlineKeyboardButton("Отримання довідки/документів",
callback_data="dovidka")
    item_6 = types.InlineKeyboardButton("Контакти кафедр/викладачів",
url="http://puet.edu.ua/kafedry/")
    item_7 = types.InlineKeyboardButton("Проходження практики",
callback_data="praktyka")
    item_8 = types.InlineKeyboardButton("Контакти всіх відділів/підрозділів",
url="http://puet.edu.ua/organizaciyna-struktura/")
    item_9 = types.InlineKeyboardButton("Докладніше про кожену категорію",
callback_data="detali")

    markup.add(item_1, item_2, item_3, item_4, item_5, item_6, item_7, item_8,
item_9)

    if message.text == r'/start':
        bot.send_message(message.chat.id, "Привіт! Як я можу вам сьогодні
допомогти?")
        bot.send_message(message.chat.id, "Виберіть категорію:",
reply_markup=markup)

```

Розберемо кожену частину коду:

```
@bot.message_handler(commands=['start', 'help'])
```

```
@print_after_execution
```

```
def handle_start(message):
```

Це встановлює обробник повідомлень для команд `/start` і `/help`.
При отриманні будь-якої з цих команд бот викличе функцію `handle_start`.

```
markup = types.InlineKeyboardMarkup(row_width=1)
```

Тут створюється клавіатура з одним рядком кнопок, використовуючи `types.InlineKeyboardMarkup`.

```
item_1 = types.InlineKeyboardButton("Інформація щодо вступу",
callback_data="vstup")
```

...

Створюються кнопки клавіатури з різними назвами із відповідними `callback_data`, які використовуються для ідентифікації натискань кнопок.

```
markup.add(item_1, item_2, ..., item_9)
```

Усі створені кнопки додаються до клавіатури.

```
if message.text == r'/start':
```

```
    bot.send_message(message.chat.id, "Привіт! Як я можу вам сьогодні допомогти?")
```

Ця умова перевіряє, чи команда була `"/start"`, і відправляє привітальне повідомлення.

```
    bot.send_message(message.chat.id, "Виберіть категорію:",
reply_markup=markup)
```

Це відправляє повідомлення з текстом `"Виберіть категорію:"` та прикріпленою клавіатурою `markup`.

Наступним кроком потрібно створити обробники для натискання кнопок.

Розберемо обробник для натискання кнопки з категорією `"Докладніше про кожну категорію"`.

```
@bot.callback_query_handler(func=lambda callback: callback.data == "detali")
def handle_detali(callback):
    bot.send_message(callback.message.chat.id, textHTML, parse_mode="HTML")
```

Цей код створює обробник для натискання кнопки з `callback_data` `"detali"`. Коли користувач натискає цю кнопку, викликається функція `handle_detali`. В цій функції відправляється повідомлення з HTML-форматованим текстом змінної `textHTML`. Параметр `parse_mode="HTML"` вказує Telegram на обробку тексту як HTML для відображення тексту з форматкуванням.

Змінна `textHTML` містить розгорнуту інформацію про кожну категорію в HTML-форматі, зокрема назви категорій, опис та типові запитання до кожної категорії (див. Додаток А).

Обробник для натискання кнопки з категорією "Інформація щодо вступу":

```
@bot.callback_query_handler(func=lambda callback: callback.data == "vstup")
def handle_vstup(callback):
    markup = types.InlineKeyboardMarkup(row_width=1)
    item_1 = types.InlineKeyboardButton("Галузі знань та спеціальності",
url="https://vstup.puet.edu.ua/#specs")
    item_2 = types.InlineKeyboardButton("Все про ЗНО/НМТ",
url="https://vstup.puet.edu.ua/pidgotovka-do-zno-2-2/")
    item_3 = types.InlineKeyboardButton("Бакалаврат", callback_data="bakalavrat")
    item_4 = types.InlineKeyboardButton("Магістратура",
url="https://vstup.puet.edu.ua/magistr/")
    item_5 = types.InlineKeyboardButton("Друга вища освіта (Магістратура)",
url="https://vstup.puet.edu.ua/vstupna-kampaniya/suchasna-vyshha-osvita-bez-zno/")
    item_6 = types.InlineKeyboardButton("Аспірантура / Докторантура",
url="https://vstup.puet.edu.ua/aspirantura-doktorantura/")
    item_7 = types.InlineKeyboardButton("Військова підготовка",
url="https://vstup.puet.edu.ua/906-2/")

    markup.add(item_1, item_2, item_3, item_4, item_5, item_6, item_7)

    bot.send_message(callback.message.chat.id, "Інформація щодо вступу:",
reply_markup=markup)
```

Код створює обробник для натискання кнопки з `callback_data` "vstup". Коли користувач натискає цю кнопку, викликається функція `handle_vstup`. У цій функції створюється клавіатура з кнопками, які містять посилання на різні сторінки веб-сайту, пов'язані з інформацією щодо вступу.

Кнопки, які посилаються на веб-сторінки, створюються за допомогою `types.InlineKeyboardButton` з вказанням параметру `url` для кожної кнопки. Кнопки, які викликають подальші дії в чат-боті, створюються з вказанням параметру `callback_data`.

Після створення клавіатури відправляється повідомлення з текстом "Інформація щодо вступу:", прикріпленою до якого є створена клавіатура.

Обробник для натискання кнопки з категорією "Бакалаврат":

```
@bot.callback_query_handler(func=lambda callback: callback.data == "bakalavrat")
def handle_bakalavrat(callback):
    markup = types.InlineKeyboardMarkup(row_width=1)
    item_1 = types.InlineKeyboardButton("На базі повної загальної середньої освіти", url="https://vstup.puet.edu.ua/bakalavr-pislya-zagalnoosvitnoyi-shkoly/")
```

```

        item_2 = types.InlineKeyboardButton("На базі диплома фахового мол.  
бакалавра, мол. бакалавра, мол. спеціаліста", url="https://vstup.puet.edu.ua/na-bazi-  
dyplomu-molodshogo-spetsialista/")

        markup.add(item_1, item_2)

        bot.send_message(callback.message.chat.id, "Бакалаврат:",
reply_markup=markup)

```

Кнопки створюються за допомогою `types.InlineKeyboardButton` з вказанням параметру `url` для кожної кнопки, яка веде на відповідну сторінку веб-сайту.

Обробник для натискання кнопки з категорією "Навчання":

```

@bot.callback_query_handler(func=lambda callback: callback.data == "navchannya")
def handle_navchannya(callback):
    markup = types.InlineKeyboardMarkup(row_width=1)
    item_1 = types.InlineKeyboardButton("Питання щодо навчання, сесії",
callback_data="navchannya_pytannya")
    item_2 = types.InlineKeyboardButton("Вибіркові дисципліни",
callback_data="navchannya_vybirkovi")
    item_3 = types.InlineKeyboardButton("Розклад занять",
url="https://schedule.puet.edu.ua/")
    item_4 = types.InlineKeyboardButton("Персональний кабінет",
url="https://cabinet.puet.edu.ua/")
    item_5 = types.InlineKeyboardButton("Сайт дистанційного навчання",
url="https://el.puet.edu.ua/")
    item_6 = types.InlineKeyboardButton("Навчальні плани освітніх програм",
url="http://puet.edu.ua/osvita/organizaciya-osvitnogo-procesu/navchalni-plany-osvitnih-  
program/")
    item_7 = types.InlineKeyboardButton("Освітні програми",
url="https://puet.edu.ua/osvita/organizaciya-osvitnogo-procesu/osvitno-profesijni-  
programy/")
    item_8 = types.InlineKeyboardButton("Графіки освітнього процесу",
url="http://puet.edu.ua/osvita/organizaciya-osvitnogo-procesu/grafiky-osvitnogo-procesu/")

    markup.add(item_1, item_2, item_3, item_4, item_5, item_6, item_7, item_8)
    bot.send_message(callback.message.chat.id, "Навчання:", reply_markup=markup)

```

Обробник для категорії "Питання щодо навчання, сесії":

```

@bot.callback_query_handler(func=lambda callback: callback.data ==
"navchannya_pytannya")
def handle_navchannya_pytannya(callback):
    markup = types.InlineKeyboardMarkup()
    item_1 = types.InlineKeyboardButton("Денна форма навчання",
callback_data="navchannya_pytannya_denna")

```

```

        item_2 = types.InlineKeyboardButton("Заочна форма навчання",
callback_data="navchannya_pytannya_zaochna")

        markup.add(item_1, item_2)

        bot.send_message(callback.message.chat.id, "Питання щодо навчання, сесії",
reply_markup=markup)

```

Обробник для натискання кнопки з `callback_data` "navchannya_pytannya_denna". Коли користувач натискає цю кнопку, викликається функція `handle_navchannya_pytannya_denna`. У цій функції створюється клавіатура з однією кнопкою.

```

@bot.callback_query_handler(func=lambda callback: callback.data ==
"navchannya_pytannya_denna")
def handle_navchannya_pytannya_denna(callback):
    markup = types.InlineKeyboardMarkup()
    markup.add(types.InlineKeyboardButton("Детальніше",
url="http://puet.edu.ua/institutes/navchalno-naukovyj-institut-dennoyi-osvity/"))

    bot.send_message(callback.message.chat.id, "Денна форма навчання:\n"+
        "Навчально-науковий інститут денної освіти\n"+
        "Кабінет. 236, 237\n"+
        "Телефон: +38 (066) 474-45-79",
        reply_markup=markup)

```

Після створення клавіатури відправляється повідомлення з текстом, що містить інформацію про денну форму навчання та контактні дані, а також кнопку "Детальніше", яка веде на веб-сторінку інституту денної освіти.

Обробник для натискання кнопки з `callback_data` "navchannya_pytannya_zaochna". Коли користувач натискає цю кнопку, викликається функція `handle_navchannya_pytannya_zaochna`. У цій функції створюється клавіатура з однією кнопкою, яка містить посилання на ресурси, пов'язані з заочною формою навчання.

```

@bot.callback_query_handler(func=lambda callback: callback.data ==
"navchannya_pytannya_zaochna")
def handle_navchannya_pytannya_zaochna(callback):
    markup = types.InlineKeyboardMarkup()

```

```
markup.add(types.InlineKeyboardButton("Детальніше",
url="http://puet.edu.ua/institutes/navchalno-naukovyj-institut-zaochno-dystancijnogo-
navchannya/"))
```

```
bot.send_message(callback.message.chat.id, "Заочна форма навчання:\n"+
"Навчально-науковий інститут заочно-дистанційного навчання\n"+
"Кабінет. 222, 222а, 217\n"+
"Телефон: +38 (097) 062-28-66\n"+
"Пошта: dekanatzaocnyj@gmail.com",
reply_markup=markup)
```

Обробник для категорії "Вибіркові дисципліни":

```
@bot.callback_query_handler(func=lambda callback: callback.data ==
"navchannya_vybirkovi")
def handle_navchannya_vybirkovi(callback):
    markup = types.InlineKeyboardMarkup()
    item_1 = types.InlineKeyboardButton("Загальна інформація",
url="http://puet.edu.ua/vibirkovi-disciplini/")
    item_2 = types.InlineKeyboardButton("Вибір/зміна вибірових дисциплін",
callback_data="navchannya_vybir_dyscyplin")
```

```
markup.add(item_1, item_2)
```

```
bot.send_message(callback.message.chat.id, "Вибіркові дисципліни:",
reply_markup=markup)
```

```
@bot.callback_query_handler(func=lambda callback: callback.data ==
"navchannya_vybir_dyscyplin")
def handle_navchannya_vybir_dyscyplin(callback):
    bot.send_message(callback.message.chat.id, "Вибір/зміна вибірових
дисциплін\n"+
"Навчально-науковий центр забезпечення якості вищої освіти\n"+
"Кабінет. 203\n"+
"Телефон: (0532) 56-05-49\n"+
"Пошта: vuod@puet.edu.ua")
```

Обробник для категорії "Технічна підтримка дистанційного навчання":

```
@bot.callback_query_handler(func=lambda callback: callback.data ==
"tehpitrymka")
def handle_tehpidtrymka(callback):
    markup = types.InlineKeyboardMarkup(row_width=1)
    item_1 = types.InlineKeyboardButton("Інструкції по роботі в системі",
url="https://el.puet.edu.ua/help/")
    item_2 = types.InlineKeyboardButton("Головний центр дистанційного
навчання", url="https://el.puet.edu.ua/")
    item_3 = types.InlineKeyboardButton("Персональний кабінет",
url="https://cabinet.puet.edu.ua/")
```

```

        item_4 = types.InlineKeyboardButton("Втрата/зміна логіну і паролю",
callback_data="tehpидtrymka_kontakty")
        item_5 = types.InlineKeyboardButton("Проблеми з дистанційним
курсом/відеозв'язком", callback_data="tehpидtrymka_kontakty")

        markup.add(item_1, item_2, item_3, item_4, item_5)

        bot.send_message(callback.message.chat.id, "Технічна підтримка дистанційного
навчання:", reply_markup=markup)

        @bot.callback_query_handler(func=lambda callback: callback.data ==
"tehpидtrymka_kontakty")
        def handle_tehpidtrymka_kontakty(callback):
            markup = types.InlineKeyboardMarkup()
            markup.add(types.InlineKeyboardButton("Детальніше",
url="https://el.puet.edu.ua/contact-us/"))

            bot.send_message(callback.message.chat.id, "3 питань технічної підтримки
сайту:\n"+
                "Кабінет. 409\n"+
                "Телефон: +38 (099) 740-56-88\n"+
                "Пошта: elearning.puet@gmail.com",
                reply_markup=markup)

```

Обробник для категорії "Оплата навчання, гуртожитку тощо":

```

        @bot.callback_query_handler(func=lambda callback: callback.data == "oplata")
        def handle_oplata(callback):
            markup = types.InlineKeyboardMarkup(row_width=1)
            item_1 = types.InlineKeyboardButton("Інформація щодо оплати",
callback_data="oplata_detali")
            item_2 = types.InlineKeyboardButton("Реквізити для оплати/Privat24",
url="https://next.privat24.ua/payments/form/%7B%22token%22:%221b69e41a3abb2d30cf5b
6851636eeaebe9hx3tqg%22%7D")

            markup.add(item_1, item_2)

            bot.send_message(callback.message.chat.id, "Оплата навчання, гуртожитку
тощо:", reply_markup=markup)

            @bot.callback_query_handler(func=lambda callback: callback.data ==
"oplata_detali")
            def handle_oplata_detali(callback):
                markup = types.InlineKeyboardMarkup()
                markup.add(types.InlineKeyboardButton("Детальніше",
url="http://puet.edu.ua/other-divisions/finansovo-ekonomiche-upravlinnya/"))

                bot.send_message(callback.message.chat.id, "Фінансово-економічне
управління:\n"+

```

```

"3 питань розрахунків за навчання звертатись у кім. 100 або за
телефонами: +38 (0532) 56-10-29; +38 (095) 848-24-70; +38 (068) 242-52-30\n"+
"3 питань розрахунків за проживання у гуртожитках, виплати
стипендій звертатись у кім. 206 або за телефоном +38 (0532) 56-08-50\n"+
"Пошта: puetglavbuh@gmail.com",
reply_markup=markup)

```

Обробник для категорії "Отримання довідки/документів":

```

@bot.callback_query_handler(func=lambda callback: callback.data == "dovidka")
def handle_dovidka(callback):
    markup = types.InlineKeyboardMarkup(row_width=1)
    item_1 = types.InlineKeyboardButton("Довідка про навчання",
callback_data="dovidka_navchannya")
    item_2 = types.InlineKeyboardButton("Довідка у ТЦК та СП",
callback_data="dovidka_tck")
    item_3 = types.InlineKeyboardButton("Студентський квиток, дублікати
документів про освіту", callback_data="dovidka_dokumenty")

    markup.add(item_1, item_2, item_3)

    bot.send_message(callback.message.chat.id, "Отримання довідки/документів:",
reply_markup=markup)

```

Обробник для категорії "Довідка про навчання":

```

@bot.callback_query_handler(func=lambda callback: callback.data ==
"dovidka_navchannya")
def handle_dovidka_navchannya(callback):
    markup = types.InlineKeyboardMarkup()
    item_1 = types.InlineKeyboardButton("Денна форма навчання",
callback_data="navchannya_pytannya_denna")
    item_2 = types.InlineKeyboardButton("Заочна форма навчання",
callback_data="navchannya_pytannya_zaochna")

    markup.add(item_1, item_2)

    bot.send_message(callback.message.chat.id, "Довідка про навчання:",
reply_markup=markup)

```

Обробник для категорії "Довідка у ТЦК та СП":

```

@bot.callback_query_handler(func=lambda callback: callback.data == "dovidka_tck")
def handle_dovidka_tck(callback):
    markup = types.InlineKeyboardMarkup()
    markup.add(types.InlineKeyboardButton("Детальніше",
url="http://puet.edu.ua/other-divisions/viddil-kadriv/"))

```

```

bot.send_message(callback.message.chat.id, "Довідка у ТЦК та СП:\n"+
    "Відділ кадрів\n"+
    "Кабінет. 320\n"+
    "Телефон: +38 (0532) 56-05-28\n"+
    "Пошта: ok@puet.edu.ua",
    reply_markup=markup)

```

Обробник для категорії "Студентський квиток, дублікати документів про освіту":

```

@bot.callback_query_handler(func=lambda callback: callback.data ==
"dovidka_dokumenty")
def handle_dovidka_dokumenty(callback):
    markup = types.InlineKeyboardMarkup()
    item_1 = types.InlineKeyboardButton("Денна форма навчання",
callback_data="dovidka_dokumenty_denna")
    item_2 = types.InlineKeyboardButton("Заочна форма навчання",
callback_data="navchannya_pytannya_zaochna")

    markup.add(item_1, item_2)

    bot.send_message(callback.message.chat.id, "Студентський квиток, дублікати
документів про освіту:", reply_markup=markup)

@bot.callback_query_handler(func=lambda callback: callback.data ==
"dovidka_dokumenty_denna")
def handle_dovidka_dokumenty_denna(callback):
    markup = types.InlineKeyboardMarkup()
    markup.add(types.InlineKeyboardButton("Детальніше",
url="http://puet.edu.ua/other-divisions/viddil-studentskoyi-dokumentacziyi/"))

    bot.send_message(callback.message.chat.id, "Денна форма навчання:\n"+
    "Відділ студентської документації\n"+
    "Кабінет. 342\n"+
    "Телефон: +38 (0532) 50-92-06\n"+
    "Пошта: vsd@puet.edu.ua",
    reply_markup=markup)

```

Обробник для категорії "Проходження практики":

```

@bot.callback_query_handler(func=lambda callback: callback.data == "praktyka")
def handle_praktyka(callback):
    markup = types.InlineKeyboardMarkup()
    markup.add(types.InlineKeyboardButton("Детальніше",
url="http://puet.edu.ua/other-divisions/czentr-zvyazkiv-z-vyrobnycztvom/"))

    bot.send_message(callback.message.chat.id, "Проходження практики:\n"+
    "Центр зв'язків з виробництвом\n"+
    "Кабінет. 219\n"+
    "Телефон: +38 (0532) 52-09-32\n"+

```

```
"Пошта: czv@puet.edu.ua",
reply_markup=markup)
```

Метод `bot.infinity_polling()` розпочинає безкінечний процес отримання оновлень від Telegram та обробки їх у чат-боті. Після запуску цієї функції бот буде постійно очікувати нових повідомлень та викликів від користувачів.

```
bot.infinity_polling()
```

Тому викликаємо цей метод після того, як вже визначили всі обробники та налаштували бота. Цей метод запускає процес взаємодії бота з користувачами, тож після його виклику бот буде працювати неперервно, обробляючи всі вхідні повідомлення та запити.

Після запуску цього скрипту бот буде доступний у Telegram. Далі відбувається його тестування, вибираючи категорії та перевіряючи взаємодію з кнопками та посиланнями.

4.3. Опис роботи довідкового чат-боту

Користувач відкриває чат з чат-ботом. Це може бути через пошук бота в Телеграмі та натискання кнопки «Старт», або ж користувач може запустити бота за допомогою ім'я користувача чи посилання на бота (рис. 4.12).

Після запуску бота, користувач отримує привітальне повідомлення та може вибрати категорію, з якою він хоче отримати додаткову інформацію або отримати допомогу (рис. 4.13). Це можна також зробити, ввівши команду `/start`.

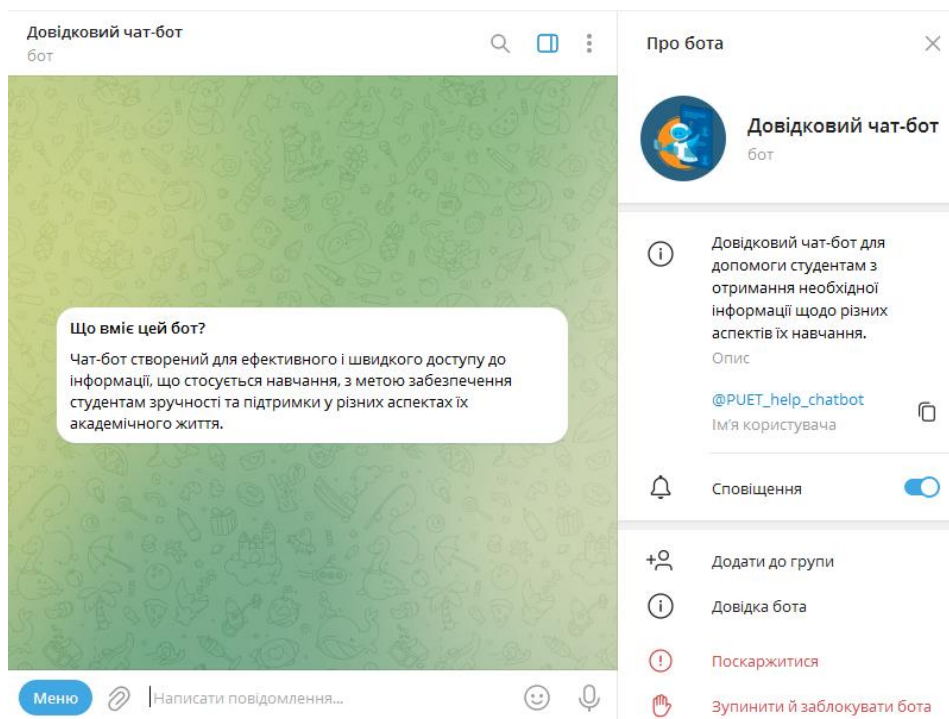


Рисунок 4.12 – Запуск чат-боту

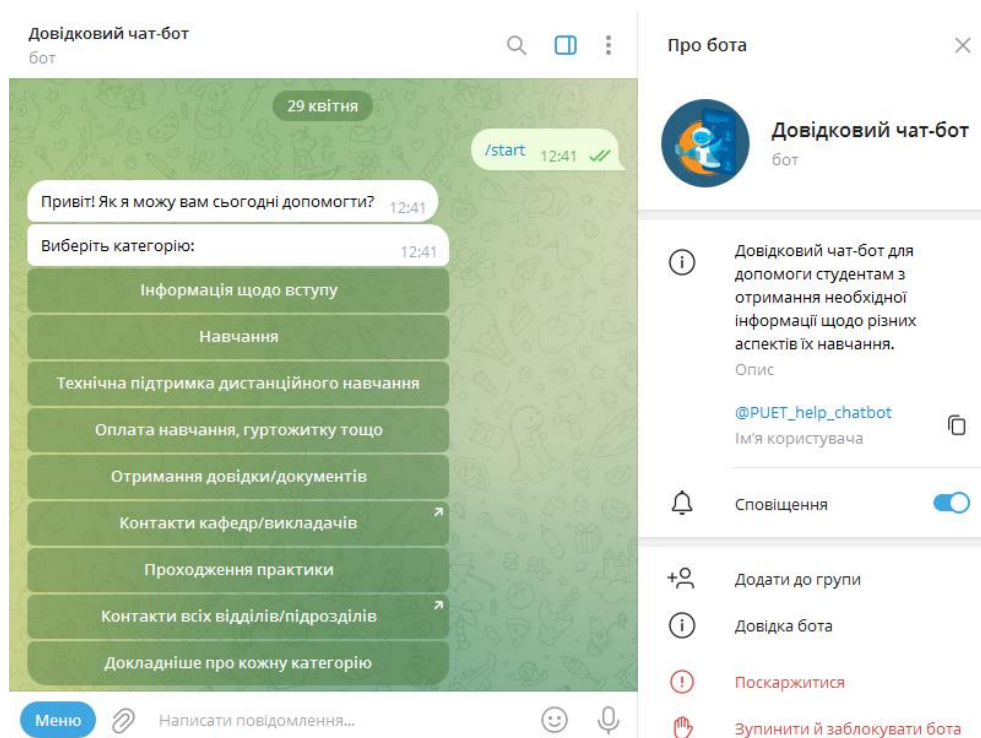


Рисунок 4.13 – Привітальне повідомлення

Після вибору категорії користувач може мати дві опції:

- Перехід за посиланням: якщо користувач натискає на кнопку з посиланням, пов'язане з певною категорією (наприклад, Контакти

кафедр/викладачів), він буде перенаправлений на веб-сторінку або інший ресурс з докладнішою інформацією (рис. 4.14).

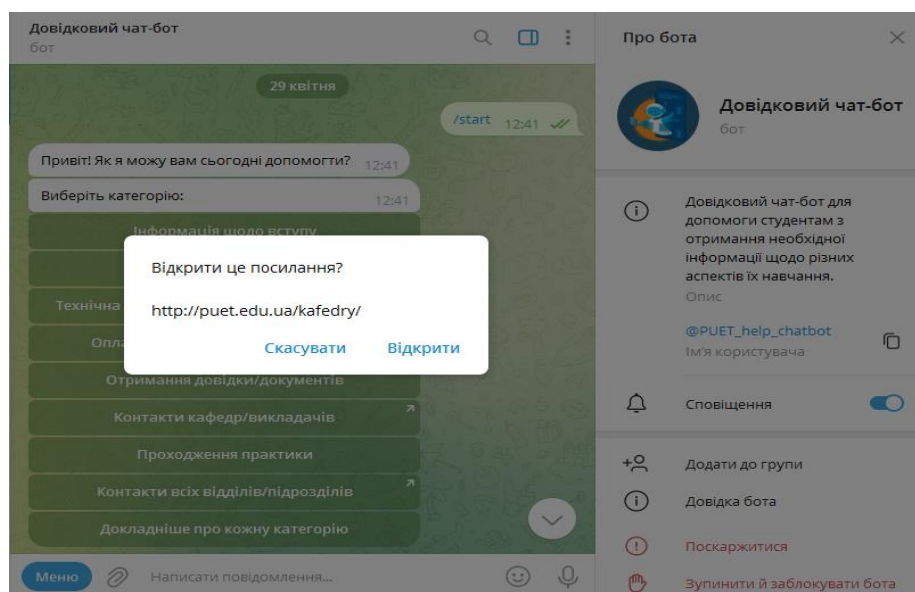


Рисунок 4.14 – Перехід за посиланням

- Вибір пункту категорії: іншою опцією може бути вибір певного пункту з обраної категорії. Наприклад, у категорії «Навчання» містяться власні пункти. Користувач може вибрати один з цих пунктів, щоб отримати більше інформації (рис. 4.15).

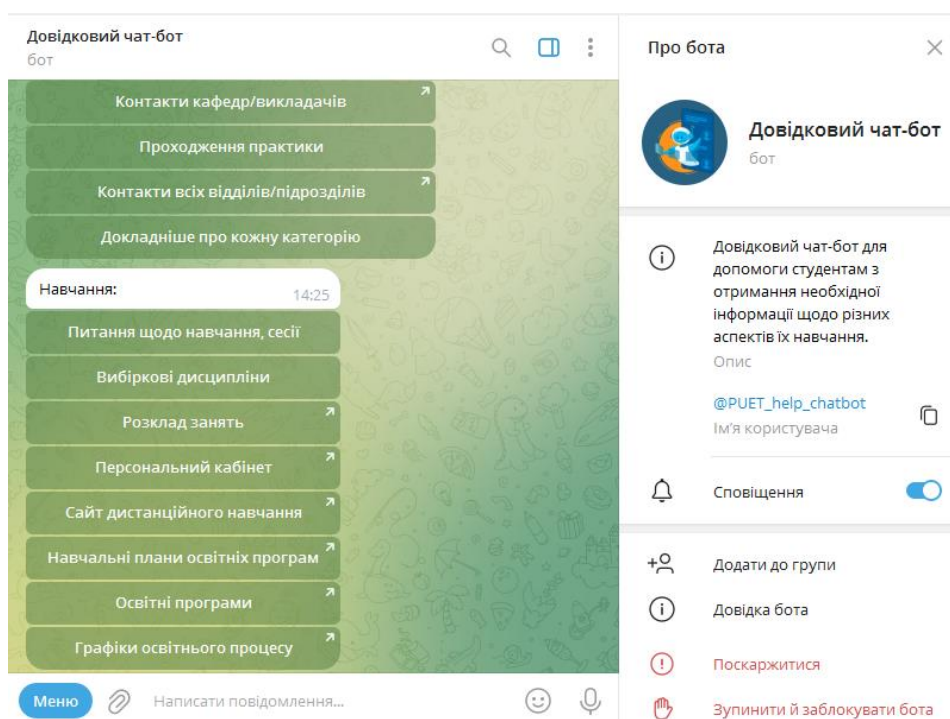


Рисунок 4.15 – Пункти в категорії «Навчання»

В кожному пункті будь-якої категорії можуть бути власні підпункти. Наприклад, при виборі «Питання щодо навчання, сесії» ще уточнюється форма навчання (рис. 4.16).

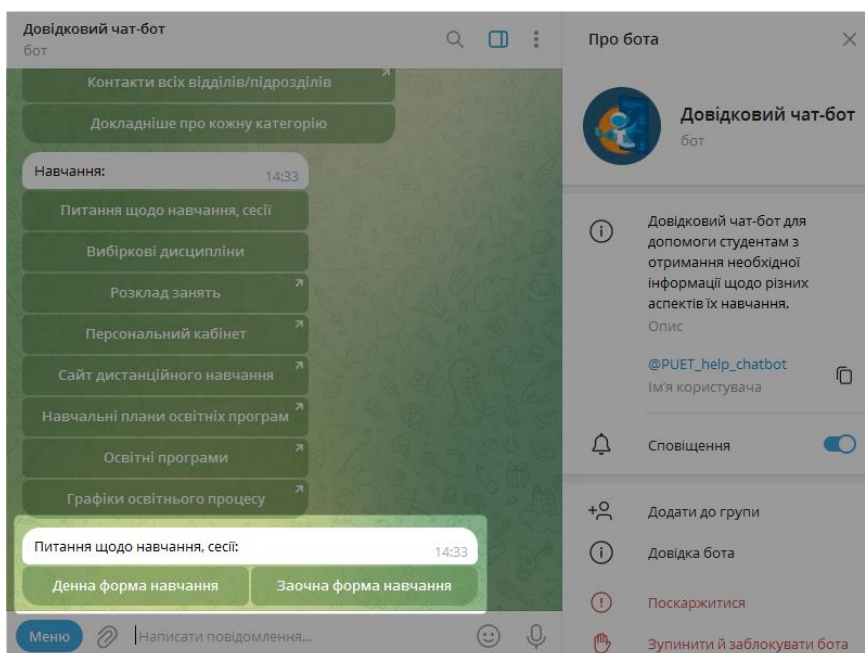


Рисунок 4.16 – Пункт «Питання щодо навчання, сесії»

Після вибору певного пункту категорії, чат-бот може відправити користувачеві додаткову інформацію – це текстове повідомлення з поясненням або інструкцією, які можуть бути корисними для користувача (рис. 4.17).

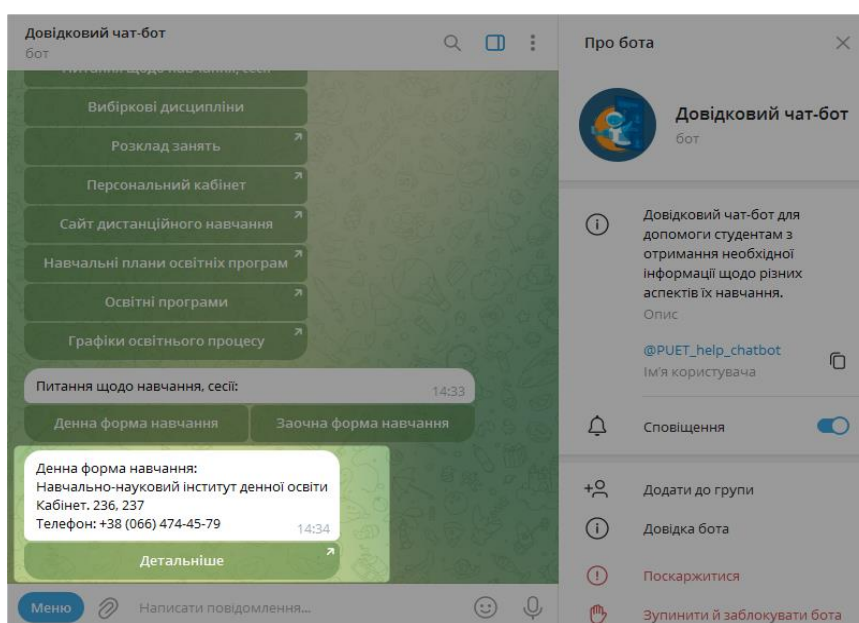


Рисунок 4.17 – Виведення повідомлення

Кінцевим результатом в кожній категорії є перехід за посиланням або виведення повідомлення.

Через деякий час після роботи чат-боту відображається відповідне повідомлення з уточненнями (рис. 4.18).

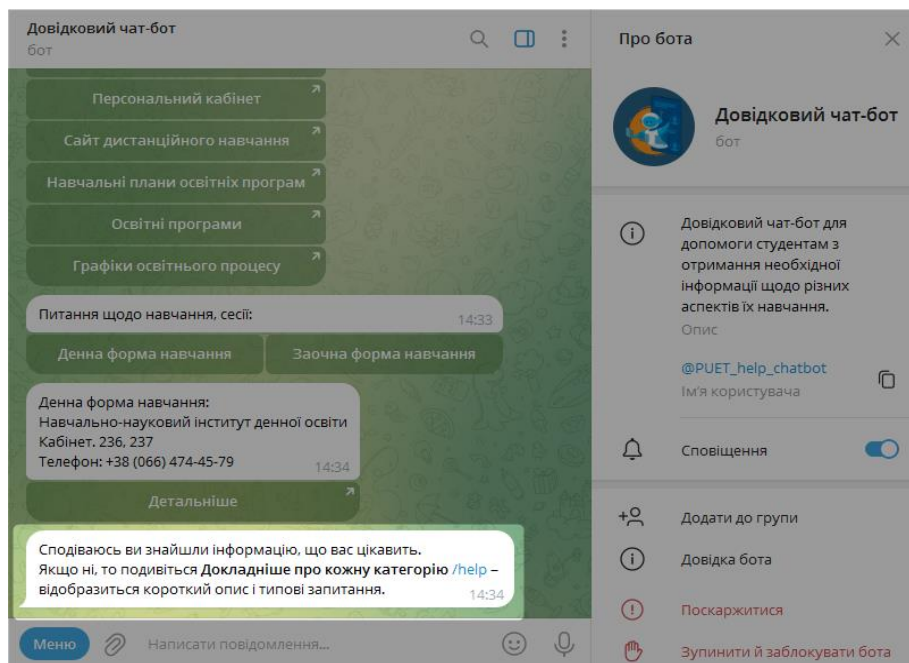


Рисунок 4.18 – Повідомлення з уточненнями

Щоб знову вивести категорії необхідно написати команду /help або натиснути на неї в повідомленні, або вибрати її в меню (рис. 4.19).

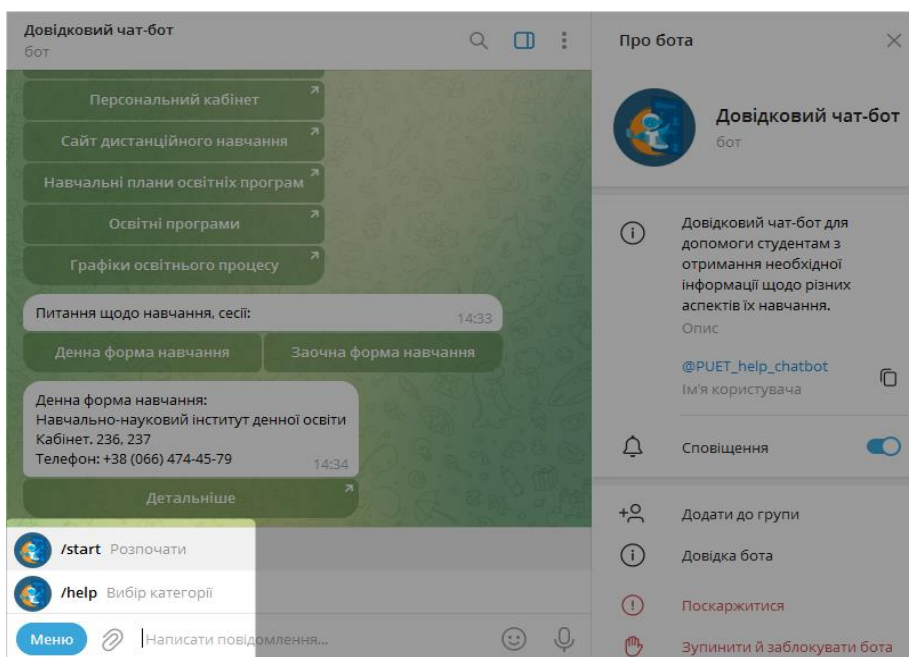


Рисунок 4.19 – Меню чат-боту

Якщо виникають проблеми з обранням категорії, то слід натиснути «Докладніше про кожну категорію» (рис. 4.20).

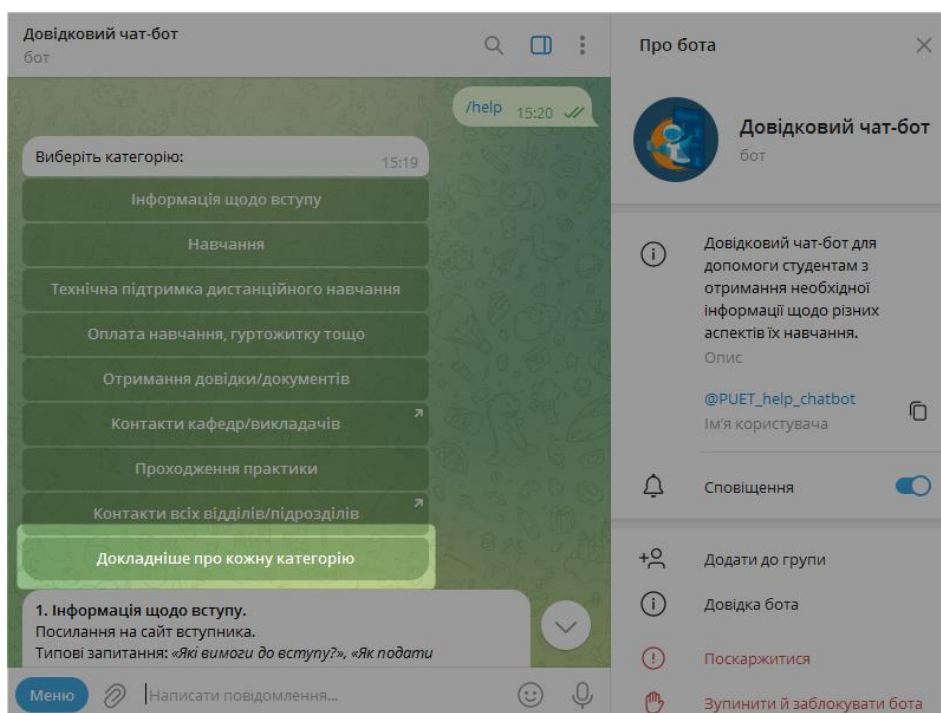


Рисунок 4.20 – Кнопка «Докладніше про кожну категорію»

Виведеться повідомлення з коротким описом кожної категорії і типовими запитаннями (рис. 4.21).

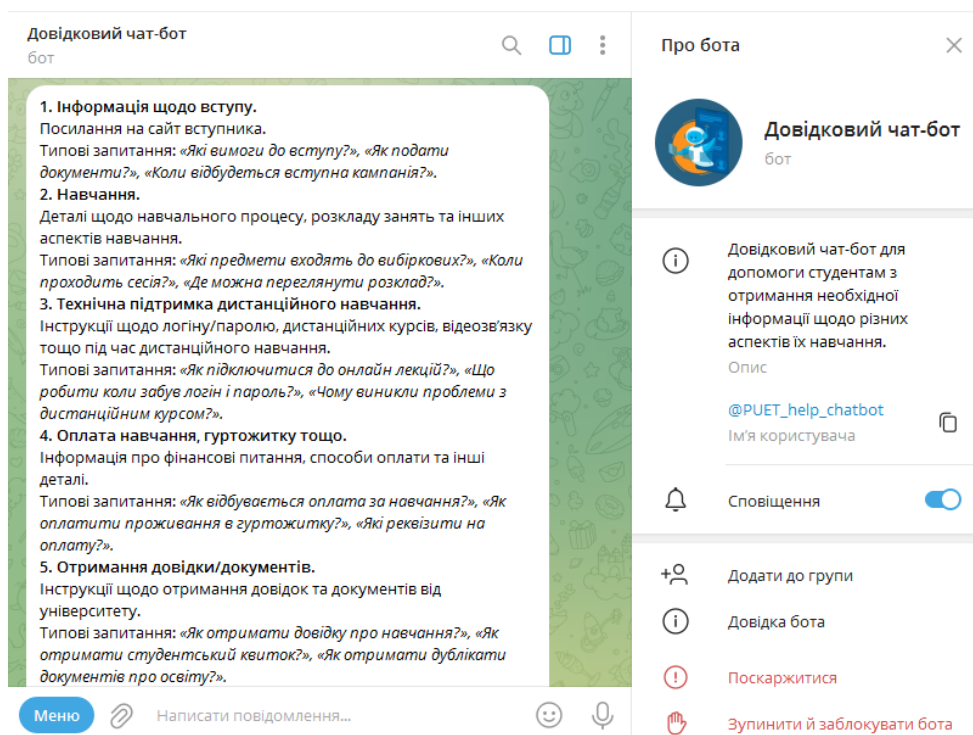


Рисунок 4.21 – Повідомлення з описом кожної категорії

ВИСНОВКИ

Робота, спрямована на розробку довідкового чат-боту для студентів, представляє собою комплексний підхід до покращення інформаційної підтримки в освітньому середовищі. У рамках теоретичної частини була поставлена задача з розробки зручного та інтуїтивно зрозумілого інструменту, який забезпечує студентів актуальною інформацією щодо різних аспектів їхнього університетського життя.

Аналіз платформ для створення чат-ботів показав, що існує багато різноманітних інструментів з різними функціональними можливостями. Однак вибір програмного забезпечення та методів розробки був здійснений з урахуванням потреб роботи, забезпечуючи оптимальну ефективність та можливість подальшого розвитку.

Аналіз існуючих рішень у сфері довідкових чат-ботів для освітніх закладів дозволив виокремити ключові функції, які слід враховувати при розробці. Це включає інформацію щодо:

1. Вступ;
2. Навчання;
3. Технічна підтримка дистанційного навчання;
4. Оплата навчання, гуртожитку тощо;
5. Отримання довідки/документів;
6. Контакти кафедр/викладачів;
7. Проходження практики;
8. Контакти всіх відділів/підрозділів;

Вивчення технологій, які використовуються при розробці чат-ботів, виявилось ключовим етапом. Обрано платформу pyTelegramBotAPI для створення чат-боту з розгалуженою структурою, що дозволяє надати різноманітну інформацію через інтуїтивно зрозумілі кнопки.

Алгоритм роботи чат-боту та його блок-схема були розроблені так, щоб користувач міг швидко та зручно отримати необхідну інформацію з

різних категорій. Кожна категорія представлена кнопкою, яка викликає відповідний обробник запитань.

Загальний висновок полягає в тому, що розроблений чат-бот є потужним інструментом для полегшення навчального процесу студентів та надання їм необхідної інформації. Поєднання теоретичних засад, аналізу платформ, практичної реалізації та вибору оптимальних технологій робить його ефективним інструментом для освітніх закладів.

Створений чат-бот вирішує важливі завдання, пов'язані з підтримкою студентів в університетському середовищі. Це дозволяє студентам ефективно взаємодіяти з університетом, отримуючи вчасну та актуальну інформацію.

Розроблений алгоритм роботи чат-боту, разом із зручною блок-схемою, забезпечує легку навігацію та доступ до різних категорій інформації. Кожен обробник відповідає за конкретну тематичну групу запитань, що спрощує взаємодію з користувачем.

Вибір платформи pyTelegramBotAPI виявився обґрунтованим, оскільки вона забезпечує зручний інтерфейс для роботи з Telegram API та можливості для розширення функціоналу чат-боту в майбутньому.

У цілому, розроблена робота є важливим інструментом для покращення комунікації між університетом та студентами. Вона сприяє підвищенню доступності інформації, оптимізації процесів та покращенню загального досвіду студентів у вищому навчальному закладі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Провотар О.І. Особливості та проблеми віртуального спілкування за допомогою чат-ботів / О.І. Провотар, Х.А. Ключко // Наукові праці ВНТУ: Інформаційні технології та комп'ютерна техніка. – 2013. – № 3. – 6 с.
2. Basics of the Microsoft Bot Framework [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/uk-ua/azure/bot-service/bot-builder-basics?view=azure-bot-service-4.0>
3. Botpress: Потужна Платформа для Розробки Чат-Ботів // Блог TradeMinister. –TradeMinister. 13 жовтня 2023 р. – Режим доступу: <https://trademinister.net/blogs/shopify/botpress>
4. Офіційний сайт Wit.ai [Електронний ресурс]. – Режим доступу: <https://wit.ai/>
5. Як налаштувати Telegram чат-бота, створювати розсилки та автоматизовані відповіді. Чат-бот Telegram [Електронний ресурс] // Сайт SendPulse. – Режим доступу: <http://puet.edu.ua/>
6. Андрощук М. В. Порівняння сервісів для створення питально-відповідальних систем / Андрощук М. В., Кирієнко О. В. // Наукові записки НаУКМА. Комп'ютерні науки. - 2020. - Т. 3. - С. 132-137.
7. Офіційний сайт ПУЕТ [Електронний ресурс] // Сайт Полтавського університету економіки і торгівлі. – Режим доступу: <http://puet.edu.ua/>
8. Левикін В.М. Візуальні мови та середовища розробки застосувань: Навч. посібник / В.М. Левикін, М.С. Кудрявцева. – Харків: ХНУРЕ, 2011. – 168 с.
9. Кундік К.В. Розробка веб-сервісу інтернет оголошень / К.В. Кундік, А.В. Гречко. – К.: Києво-Могилян. акад., 2019. – 19 с.
10. Головний центр дистанційного навчання [Електронний ресурс] // Сайт дистанційного навчання, Полтавський університет економіки і торгівлі. – Режим доступу: <https://el.puet.edu.ua/>

11. The Python Tutorial [Електронний ресурс] .– Режим доступу:
<https://docs.python.org/3/tutorial/index.html>
12. Ерік Маттес Пришвидшений курс Python / Переклад : Ольга Белова. –
Видавництво Старого Лева, 2021. – 600 с.
13. pyTelegramBotAPI 4.14.0 [Електронний ресурс] .– Режим доступу:
<https://pypi.org/project/pyTelegramBotAPI/>
14. Як створити чат-бот у Telegram [Електронний ресурс] .– Режим
доступу: <https://sendpulse.ua/knowledge-base/chatbot/telegram/create-telegram-chatbot>
15. Ashutosh Krishna Як створити телеграм-бота за допомогою Python /
Перекладач: Anastasiia Buievych [Електронний ресурс] .– Режим
доступу: <https://www.freecodecamp.org/ukrainian/news/yak-stvoryty-telehram-bota-za-dopomohoyu-python/>

ДОДАТОК А. КОД ЧАБ-БОТУ