

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО»**

Навчально-науковий інститут атомної та теплової енергетики
Кафедра інженерії програмного забезпечення в енергетиці

ДО ЗАХИСТУ ДОПУЩЕНО

В.о. завідувача кафедри
_____ Олександр КОВАЛЬ

«___» _____ 2025 р.

Дипломна робота

**на здобуття ступеня бакалавра
за освітньо-професійною програмою «Інженерія програмного забезпечення
інтелектуальних кібер-фізичних систем в енергетиці»
спеціальності 121 Інженерія програмного забезпечення**

**на тему: «Інтелектуальний чат-бот для інформаційного супроводу
абітурієнтів під час вступної кампанії»**

Виконала:

студентка IV курсу, групи ТВ-13

Поліщук Вікторія Володимирівна

(прізвище, ім'я, по батькові)

_____ (підпис)

Керівник:

Старший викладач, PhD, Колумбет В. П.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Консультант з першого розділу:

Доцент, к.ф-м.н., доц. Свинчук О. В.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Рецензент:

_____ (посада, науковий ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Засвідчую, що у цій дипломній роботі
немає запозичень із праць інших авторів
без відповідних посилань.

Студентка _____
(підпис)

Київ – 2025

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Навчально-науковий інститут атомної та теплової енергетики
Кафедра інженерії програмного забезпечення в енергетиці
Рівень вищої освіти перший (бакалаврський)
Спеціальність 121 Інженерія програмного забезпечення
Освітньо-професійна програма «Інженерія програмного забезпечення
інтелектуальних кібер-фізичних систем в енергетиці»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

_____ Олександр КОВАЛЬ

«__» _____ 2025р.

ЗАВДАННЯ

на дипломну роботу студенту

_____ Поліщук Вікторії Володимирівні

(прізвище, ім'я, по батькові)

1. Тема роботи: Інтелектуальний чат-бот для інформаційного супроводу абітурієнтів під час вступної кампанії

керівник роботи Колумбет Вадим Петрович, PhD

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «__» _____ 2025р. № _____

2. Строк подання студентом роботи «__» _____ 2025р.

3. Вихідні дані до роботи: мова програмування Python, середовище розробки PyCharm, бібліотека discord.py, хмарна база даних Cloud Firestore, фреймворк для вебзастосунків Streamlit.

4. Зміст (дипломної роботи) пояснювальної записки (перелік завдань, які потрібно розробити): проаналізувати літературні джерела та існуючі рішення; спроектувати чат-бота на платформі Discord; створити механізм обробки природної мови; розробити функції отримання даних з офіційних джерел; виконати тестування системи.

5. Перелік ілюстративного матеріалу: діаграма потоків даних, діаграма варіантів використання, діаграма компонентів, скріншоти взаємодії з чат-ботом, інтерфейс вебзастосунку.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Свинчук О. В., доцент		

7. Дата видачі завдання «31» жовтня 2024р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Строки виконання етапів роботи	Примітка
1	Отримання завдання	30.10.2024	
2	Дослідження предметної області	31.10.2024 – 01.11.2024	
3	Дослідження існуючих рішень	02.11.2024 – 01.12.2024	
4	Постановка вимог до проектування системи	02.12.2024 – 12.01.2025	
5	Розробка програмного продукту	13.01.2025 – 11.05.2025	
6	Тестування	12.05.2025 – 15.05.2025	
7	Захист програмного продукту	12.05.2025 – 15.05.2025	
8	Оформлення дипломної роботи	19.05.2025 – 01.06.2025	
9	Передзахист	02.06.2025 – 06.06.2025	
10	Захист	16.06.2025 – 27.06.2025	

Студент

_____ (підпис)

Вікторія ПОЛІЩУК

_____ (ім'я, прізвище)

Керівник роботи

_____ (підпис)

Вадим КОЛУМБЕТ

_____ (ім'я, прізвище)

РЕФЕРАТ

Структура та обсяг дипломної роботи. Робота містить 59 сторінок, 29 рисунків, 3 додатки та 17 посилань.

Метою роботи є розробка інтелектуального чат-бота для інформаційного супроводу абітурієнтів під час вступної кампанії, який забезпечить оперативне й точне надання інформації та допоможе у вирішенні актуальних питань.

Для досягнення поставленої мети виконано такі завдання:

- проаналізовано літературні джерела, офіційні сайти освітніх установ та наявні рішення;
- визначено основні функціональні та нефункціональні вимоги;
- розроблено архітектуру та реалізовано програмну систему у вигляді Discord-бота;
- реалізовано підтримку діалогової взаємодії та надання інформації з офіційних сайтів;
- виконано тестування та виправлення неполадок.

Практичне значення одержаних результатів полягає в створенні інструменту для освітніх закладів, що дозволяє автоматизувати надання інформації абітурієнтам, зменшити навантаження на приймальні комісії та підвищити зручність доступу до актуальної інформації. Розроблена система є масштабованою, легкою у використанні та підтримує розширення функціональності.

Апробація результатів дипломної роботи

1. Поліщук В.В., Свинчук О.В., Колумбет В.П., Бандурка О.І. Чат-боти в інформаційному супроводі абітурієнтів під час вступної кампанії. *Матеріали XXV Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів «Стан, досягнення та перспективи інформаційних систем і технологій»*. Одеса, 17-18 квітня 2025 р. С. 67-69.

URL: https://ontu.edu.ua/download/konfi/2025/Conference_abstract-IT-2025.pdf

Ключові слова: вступна кампанія, чат-бот, обробка природної мови, Python.

ABSTRACT

Structure and scope of the thesis. The work contains 59 pages, 29 figures, 3 appendices and 17 references.

The purpose of the work is to develop an intelligent chatbot for information support of applicants during the admission campaign, which will ensure prompt and accurate provision of information and help in resolving urgent issues.

To achieve the set goal, the following tasks were performed:

- literary sources, official websites of educational institutions and existing solutions were analyzed;
- the main functional and non-functional requirements were determined;
- the architecture was developed and a software system in the form of a Discord bot was implemented;
- support for dialogic interaction and provision of information from official websites was implemented;
- testing and troubleshooting were performed.

The practical value of the results obtained is in creating a tool for educational institutions that allows automating the provision of information to applicants, reducing the burden on admissions committees and increasing the convenience of access to relevant information. The developed system is scalable, easy to use and supports the expansion of functionality.

Approbation of the results of the thesis

1. Polishchuk V.V., Svinchuk O.V., Kolumbet V.P., Bandurka O.I. Chatbots in the information support of applicants during the entrance campaign. *Materials of the XXV All-Ukrainian Scientific and Technical Conference of Young Scientists, Postgraduate Students and Students "State, Achievements and Prospects of Information Systems and Technologies"*. Odesa, April 17-18, 2025. P. 67-69.

URL: https://ontu.edu.ua/download/konfi/2025/Conference_abstract-IT-2025.pdf

Keywords: entrance campaign, chatbot, natural language processing, Python.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	8
ВСТУП.....	9
1 ПОСТАНОВКА ЗАВДАННЯ СТВОРЕННЯ ІНТЕЛЕКТУАЛЬНОГО ЧАТ-БОТУ	11
1.1 Постановка завдання.....	11
1.2 Визначення користувачів системи.....	13
Висновки до розділу 1.....	14
2 АНАЛІЗ ІСНУЮЧИХ СИСТЕМ	15
2.1 Офіційні вебсайти університетів	15
2.2 Інформаційні системи для вступників	16
2.3 Спеціалізовані форуми та спільноти	18
Висновки до розділу 2.....	19
3 ЗАСОБИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	20
3.1 Вимоги до системи.....	20
3.2 Програмне середовище.....	21
3.3 Мова програмування та бібліотеки	22
3.3.1 SpaCy та Py morphology	23
3.3.2 Discord.py.....	24
3.3.3 BeautifulSoup	25
3.3.4 Streamlit.....	25
3.3.5 FPDF та XlsxWriter	26
3.4 Cloud Firestore	27
Висновки до розділу 3.....	28

4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ	29
4.1 Архітектура системи	29
4.1.1 Діаграма компонентів	30
4.1.2 Діаграма варіантів використання.....	32
4.1.3 Діаграма потоків даних.....	35
4.2 Програмна реалізація	37
4.2.1 Discord-бот	37
4.2.2 Обробка запитів користувача.....	38
4.2.3 Зчитування даних із сайтів	39
4.2.4 Cloud Firestore	39
4.2.5 Вебінтерфейс для адміністрування.....	40
4.2.6 Формування файлів	41
Висновки до розділу 4.....	41
5 РОБОТА КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ.....	43
5.1 Взаємодія з ботом.....	43
5.2 Взаємодія із вебзастосунком для адміністрування.....	53
Висновки до розділу 5.....	56
ВИСНОВКИ.....	57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	58
ДОДАТОК А Лістинг розробленої системи.....	60
ДОДАТОК Б Апробація	65
ДОДАТОК В Презентація	69

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД	База даних
ОП	Освітня програма
ЄДЕБО	Єдина державна електронна база з питань освіти
IDE	(англ. Integrated Development Environment) – інтегроване середовище розробки
NLP	(англ. Natural Language Processing) – обробка природної мови
API	(англ. Application Programming Interface) – інтерфейс взаємодії застосунків
Discord	Платформа для обміну текстових, голосових чи відео-повідомлень між користувачами

ВСТУП

Щороку тисячі абітурієнтів стикаються з необхідністю вибору майбутньої професії та навчального закладу. Цей процес є одним із найважливіших етапів у житті людини, адже саме від нього значною мірою залежить подальший професійний та особистий розвиток. Разом з тим, він супроводжується великим обсягом інформації, яку потрібно зібрати, проаналізувати і засвоїти. Зокрема, абітурієнтам потрібно ознайомитися з численними спеціальностями, освітніми програмами, умовами вступу, вимогами до документів, а також з датами подачі заяв на навчання [1].

У сучасних умовах цифровізації більшість інформації про вступ доступна переважно в електронному форматі, зокрема на офіційних вебсайтах університетів, у нормативних документах та в інформаційних системах, таких як ЄДЕБО. Це значно спрощує доступ до необхідної інформації та дозволяє скоротити витрати ресурсів закладів вищої освіти, що потрібні для управління процесом вступу [2]. Однак через великий потік даних та складність навігації багато абітурієнтів стикаються з труднощами. Вони змушені витратити багато часу на пошук необхідних даних, переглядаючи сторінки офіційних сайтів університетів або звертаючись до приймальних комісій, які в період вступної кампанії не завжди можуть надати швидку відповідь.

Сучасні технології відкривають нові можливості для автоматизації процесу інформаційного супроводу абітурієнтів. Одним із ефективних рішень є використання чат-ботів – програм, які імітують живе спілкування з користувачами та можуть швидко надавати необхідні відповіді на їх запити. За останні роки вони зарекомендували себе як ефективний інструмент в електронній комерції, технічній підтримці та наданні державних послуг. Тому більшість людей зараз віддають перевагу чат-ботам, а не традиційним пошуковим системам, оскільки вони можуть дати те, чого не можуть пошукові системи, – прямі відповіді в більш персоналізованій формі [3]. Завдяки інтерактивній формі комунікації та здатності

обробляти запити у режимі реального часу, чат-боти набули популярності серед молоді та можуть значно спростити пошук інформації, зробивши його більш структурованим і зручним.

Важливим також є те, що сучасні боти можуть бути інтегровані в популярні платформи комунікації – такі як Discord, WhatsApp чи Facebook Messenger. Це робить їх ще більш доступними та привабливими для цільової аудиторії, адже абітурієнти можуть отримати консультації у звичному для них середовищі без потреб переходити на сторонні сайти та вчитувати там усю інформацію, чи завантажувати окремі додатки.

Такі системи вже активно застосовуються у багатьох сферах діяльності: від підтримки клієнтів у бізнесі до надання послуг в державних установах, і тепер набувають популярності в освітніх установах, де здатні значно полегшити процеси, пов'язані з наданням консультацій, інформацій та відповідей на питання користувачів.

У зв'язку з цим стає очевидною актуальність даної роботи, а саме розробка інтелектуального чат-бота, який допомагатиме абітурієнтам швидко знаходити потрібну інформацію у режимі звичайного спілкування. Такий підхід дозволить не лише оптимізувати процес інформаційного супроводу вступників, зменшити кількість повторних звернень та знизити навантаження на приймальні комісії, а й підвищити загальний рівень інформованості потенційних студентів.

1 ПОСТАНОВКА ЗАВДАННЯ СТВОРЕННЯ ІНТЕЛЕКТУАЛЬНОГО ЧАТ-БОТУ

Одним із ключових викликів у сфері інформаційного супроводу абітурієнтів є забезпечення доступу до актуальних та структурованих даних про вступну кампанію в максимально зручній та швидкий спосіб. Попри наявність інформації у відкритих джерелах, а саме на офіційних сторінках сайтів чи новинних статтях, її фрагментованість та складність пошуку можуть створювати додаткові перешкоди для вступників.

Основна проблема полягає не лише у великій кількості інформації, а й у необхідності швидкого її отримання відповідно до індивідуальних запитів користувачів. Багато абітурієнтів потребують персоналізованих відповідей, які враховують їхні конкретні умови, наприклад бажану спеціальність, умови вступу, терміни подачі документів, рівень підготовки тощо. Більшість абітурієнтів не мають достатнього досвіду чи навичок для оперативного пошуку та систематизації такої інформації. Це створює ризики втрати можливостей та шансів для вступників через необізнаність або запізнілі дії.

У зв'язку з цим виникає необхідність створення інструменту, який виконуватиме роль помічника і надаватиме користувачам персоналізовану, структуровану й актуальну інформацію у зручній та зрозумілій формі – через діалогову взаємодію. Найбільш зручним рішенням для такого завдання є використання чат-бота, що працює на популярних для молоді платформах, наприклад, Discord чи WhatsApp.

1.1 Постановка завдання

Метою дипломної роботи є створення інтелектуального чат-бота, який виступатиме ефективним інструментом для автоматизації процесу надання інформації абітурієнтам, зменшуючи необхідність ручного пошуку даних та

зробивши його максимально швидким і зрозумілим. Це сприятиме підвищенню обізнаності вступників та полегшить взаємодію із системою освіти.

Для досягнення мети необхідно виконати такі завдання:

- провести аналіз існуючих рішень та офіційних сайтів освітніх установ;
- визначити основні вимоги:
 - розробити функціонал для інтерактивного надання офіційної інформації про умови вступу та вимоги до абітурієнтів;
 - реалізувати систему відповідей на часті запитання;
 - підтримувати діалогову взаємодію для можливості уточнювати та деталізувати відповіді залежно від потреб користувача;
 - розробити механізм персоналізації інформації залежно від спеціальності та індивідуальних запитів абітурієнта;
 - забезпечити можливість нагадування про важливі дати та події, пов'язані з вступною кампанією;
 - реалізувати зручний для користувачів інтерфейс, враховуючи інтуїтивність та простоту використання.
- розробити архітектуру системи та відобразити її у вигляді діаграм;
- реалізувати програмну систему у вигляді Discord-бота;
- виконати тестування та виправлення неполадок.

Розроблений інтелектуальний чат-бот дозволить:

- централізовано збирати та впорядкувати дані про спеціальності, вступні вимоги та процес подачі документів;
- забезпечувати швидкий і зручний доступ до цієї інформації через діалоговий інтерфейс;
- підтримуватиме діалогову модель, що дозволяє уточнювати деталі запиту та рекомендувати корисні матеріали;
- реалізовуватиме механізми розпізнавання користувацьких запитів та формування відповідей у зрозумілій та лаконічній формі;

- забезпечуватиме доступність у популярних месенджерах для широкого кола користувачів, що зробить сервіс зручним для молоді.

Така система може бути корисною не лише у межах однієї освітньої установи, а й масштабуватися для використання в регіональних або національних інформаційних кампаніях, що стосуються вступу до закладів освіти. Її можна інтегрувати у зовнішні комунікаційні платформи, наприклад Discord, де вже існують активні спільноти абітурієнтів. Крім того, її можна адаптувати до різних спеціальностей, освітніх програм чи напрямів підготовки, оновлюючи інформацію залежно від змін у вступній кампанії.

Завдяки цьому система може ефективно застосовуватися у серверах приймальних комісій закладів вищої освіти, на профорієнтаційних заходах, виставках або днях відкритих дверей, де абітурієнти можуть у зручному форматі отримувати відповіді на запитання.

1.2 Визначення користувачів системи

Система орієнтована на взаємодію з двома основними категоріями користувачів: Discord-користувачі та адміністратори.

Користувачам системи на платформі Discord можуть виступати абітурієнти, їхні батьки, викладачі та всі, кому цікаво дізнатися про актуальну інформацію щодо процесу вступу. Це основні кінцеві користувачі чат-бота. Вони звертаються до системи для отримання актуальної, структурованої та персоналізованої інформації, що стосується вступної кампанії. Серед їхніх типових потреб – пошук даних про спеціальності, умови вступу, терміни подачі заяв, перелік необхідних документів, відповіді на часті запитання, гранти тощо. Для них важливим є зручність та легкість у використанні, миттєві відповіді та можливість уточнювати запити у діалозі. Чат-бот зможе забезпечити ці функції через інтерактивну взаємодію в середовищі Discord.

Адміністраторами системи виступають представники освітнього закладу, а саме – відповідальні особи від приймальної комісії, які мають доступ до інтерфейсу управління даними. Їхнім основним завданням є оновлення та коригування інформації, яка надає чат-бот користувачам.

З метою забезпечення зручності та ефективності цієї роботи доцільно створити окремий вебінтерфейс, який не вимагатиме значних ресурсів і часу для обслуговування, а також буде простим у використанні з можливістю зручного оновлення інформації. Через такий інтерфейс адміністратор зможе переглядати, додавати, редагувати або видаляти освітні програми та інформацію про них. Оскільки обсяг даних може бути значним, важливо забезпечити легку навігацію по записах. Тому необхідним також є реалізувати підхід фільтрації, пошуку та сортування, що дозволить покращити взаємодію з даними для адміністратора шляхом швидкого доступу до конкретних освітніх програм. Такий підхід дозволить підтримувати інформацію в актуальному стані без потреб змінювати код чат-бота.

Висновки до розділу 1

У даному розділі проаналізовано актуальні проблеми, пов'язані з інформаційним супроводом абітурієнтів під час вступної кампанії. Визначено, що основні труднощі полягають у фрагментованості, складності пошуку та недостатній персоналізації доступної інформації. Це знижує ефективність комунікації між вступниками та освітніми закладами, особливо в умовах великої кількості даних та обмеженого часу.

Сформульовано мету дипломної роботи – створення інтелектуального чат-бота для інформаційного супроводу абітурієнтів щодо процесів вступу. Визначено конкретні завдання, які необхідно виконати для досягнення цієї мети, зокрема встановлення вимог, розробка архітектури та програмна реалізація системи.

Також визначено основні категорії користувачів системи: Discord-користувачі та адміністратори.

2 АНАЛІЗ ІСНУЮЧИХ СИСТЕМ

Інформаційна підтримка абітурієнтів є важливим завданням як для навчальних закладів, так і для державних органів, що регулюють процес вступу. Сьогодні існує кілька основних джерел отримання інформації про спеціальності, умови вступу та освітні програми: офіційні вебсайти університетів, централізовані інформаційні системи та спеціалізовані форуми. Кожен із цих підходів має свої особливості, переваги та недоліки, які впливають на зручність та ефективність пошуку інформації для абітурієнтів.

2.1 Офіційні вебсайти університетів

Офіційні вебресурси університетів є основним джерелом достовірної інформації про освітні програми, умови вступу, терміни подачі документів та особливості навчання в закладі [4]. Як правило, кожен університет має власний сайт, де розміщується розділ для абітурієнтів із необхідними матеріалами, контактними даними приймальної комісії та посиланнями на нормативні документи (рисунк 2.1).

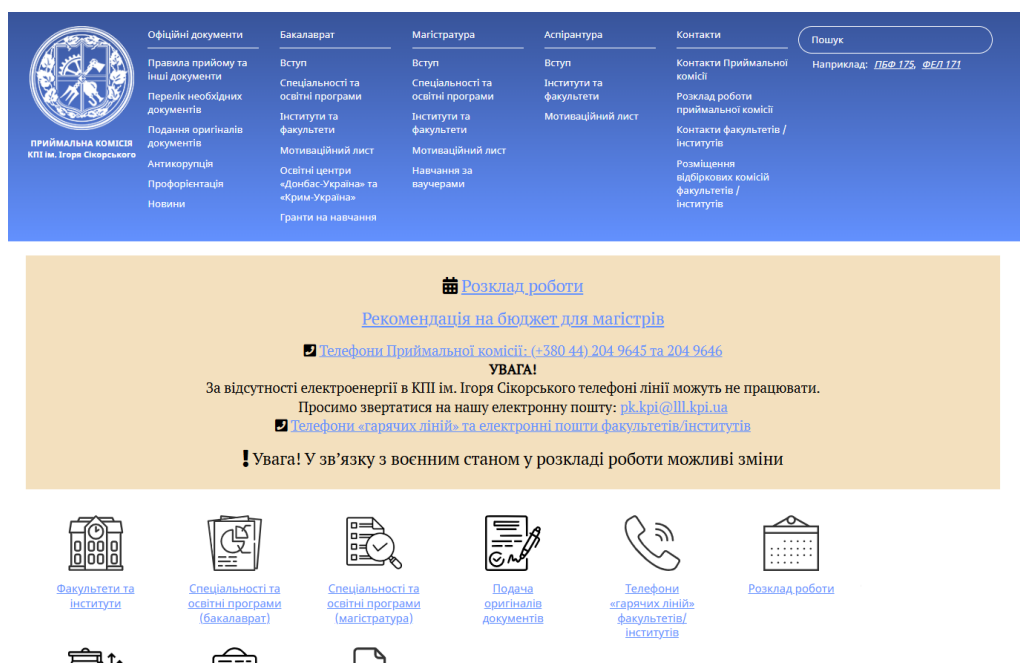


Рисунок 2.1 – Сторінка офіційного сайту приймальної комісії КПІ

Попри те, що вебсайти університетів містять найбільш точну та актуальну інформацію, їх використання має певні труднощі. Зокрема, структура таких сайтів може бути складною, що ускладнює швидкий пошук необхідних відомостей. Інформація нерідко представлена у вигляді великих текстових документів або PDF-файлів, які потребують додаткового опрацювання з боку абітурієнта. Крім того, відсутність інтерактивного формату подачі інформації змушує користувача самостійно аналізувати й систематизувати знайдені дані.

Перевагами такого способу є:

- достовірність та актуальність інформації;
- докладний опис навчальних програм та вимог до вступу;
- офіційне джерело даних.

Недоліками такого способу є:

- складна навігація та перевантаженість інформацією;
- відсутність персоналізації, тобто користувачеві потрібно самостійно фільтрувати дані;
- відсутність інтерактивності та можливості діалогу.

2.2 Інформаційні системи для вступників

Інформаційні системи для абітурієнтів, такі як Єдина державна електронна база з питань освіти (ЄДЕБО), є централізованими ресурсами, що містять дані про всі українські університети та їхні конкурсні пропозиції [5]. Такі платформи дозволяють переглядати умови вступу, необхідні сертифікати ЗНО/НМТ, конкурсні предмети та прохідні бали минулих років (рисунок 2.2).

Основною перевагою подібних систем є можливість швидкого пошуку інформації за різними критеріями. Наприклад, абітурієнт може обрати спеціальність, університет або регіон навчання та отримати перелік актуальних пропозицій. Крім того, на таких платформах часто доступна статистика вступної кампанії минулих років, що допомагає оцінити шанси на зарахування.

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
БАКАЛАВР (основа вступу - повна загальна середня освіта)

Спеціальність	121 ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	КОНКУРСНІ ПРЕДМЕТИ	
Назва пропозиції	Інститут атомної та теплової енергетики, 121 Інженерія програмного забезпечення, Бакалавр, денна форма	1	0.3 НМТ Українська мова ≥ 100
Код конкурсної пропозиції	1294929	2	0.5 НМТ Математика ≥ 100
Освітня програма	Інженерія програмного забезпечення інтелектуальних кіберфізичних систем в енергетиці	3	0.2 НМТ Історія України ≥ 100
Факультет	Навчально-науковий інститут атомної та теплової енергетики	0.2 НМТ Українська література В ≥ 100	0.3 НМТ Іноземна мова В ≥ 100
Форма навчання	Денна	4	0.2 НМТ Біологія В ≥ 100
Курс зарахування	1 курс	0.2 НМТ Географія В ≥ 100	0.4 НМТ Фізика В ≥ 100
Вид пропозиції	Відкрита, з пріоритетом	0.2 НМТ Хімія В ≥ 100	5
Термін навчання	01.09.2024 – 30.06.2028 (Зр 10м)	5	0.3 ЗНО Українська мова ≥ 100
Термін подачі заяв	19.07.2024 – 31.07.2024	6	0.5 ЗНО Математика ≥ 100
Ліцензійний обсяг	120	0.3 ЗНО Іноземна мова В ≥ 100	0.2 ЗНО Історія України В ≥ 100
Макс. обсяг держзамовлення	90	7	0.2 ЗНО Біологія В ≥ 100
Регіональний коефіцієнт	1	0.2 ЗНО Географія В ≥ 100	0.4 ЗНО Фізика В ≥ 100
Вартість навчання за рік (контракт)	54900	0.2 ЗНО Хімія В ≥ 100	0
Загальна вартість за повний термін навчання	219600	0	Мотиваційний лист
		Вимоги до мотиваційного листа	

СТАТИСТИКА ЗАЯВ			
365	Подано заяв	360	Допущено до конкурсу
160.1	Сер. бал	122.6	Мін. бал
54	Рекомендовано на загальних підставах	50	Зараховано на бюджет
158.5	Мін. бал рекомендованих	144.5	Мін. бал зарахованих на бюджет
279	Заяв на бюджет	190.1	Макс. бал
12	Зараховано на контракт	139.6	Мін. бал зарахованих на контракт

[🏠](#) [☰ Список вступників](#)

Рисунок 2.2 – Конкурсні пропозиції в ЄДЕБО

Однак, незважаючи на зручність доступу до великого обсягу даних, інформаційні системи мають певні обмеження. Їхній інтерфейс нерідко є перевантаженим, що ускладнює навігацію. Крім того, такі платформи не надають персоналізованих відповідей та не можуть враховувати індивідуальні потреби користувача.

Переваги:

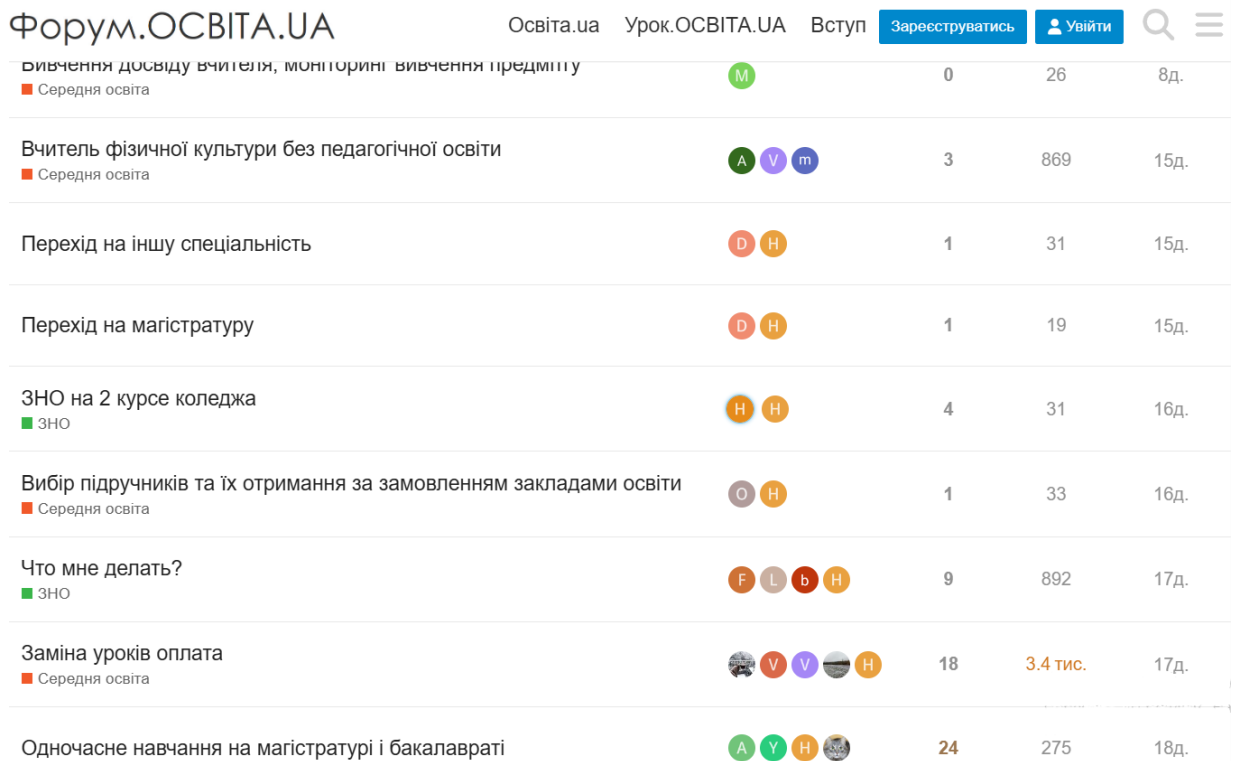
- зібрана інформація про всі університети в одному місці;
- можливість порівняння спеціальностей та конкурсних умов;
- доступ до статистики попередніх років.

Недоліки:

- складний інтерфейс, який потребує часу на освоєння;
- відсутність інтерактивної взаємодії та можливості ведення діалогу.

2.3 Спеціалізовані форуми та спільноти

Окрім офіційних джерел, абітурієнти часто шукають інформацію у спеціалізованих онлайн-спільнотах та на форумах. Прикладами таких ресурсів є освітній форум Освіта.ua (рисунок 2.3) [6], тематичні групи у соціальних мережах (наприклад, у Facebook) та дискусійні майданчики на студентських сайтах.



Тема	Категорія	Аватари	Вступ	Зареєструватись	Увійти	Пошук	Меню
Вивчення досвіду вчителя, моніторинг вивчення предмету	Середня освіта	M	0	26	8д.		
Вчитель фізичної культури без педагогічної освіти	Середня освіта	A V m	3	869	15д.		
Перехід на іншу спеціальність		D H	1	31	15д.		
Перехід на магістратуру		D H	1	19	15д.		
ЗНО на 2 курсе коледжа	ЗНО	H H	4	31	16д.		
Вибір підручників та їх отримання за замовленням закладами освіти	Середня освіта	O H	1	33	16д.		
Что мне делать?	ЗНО	F L b H	9	892	17д.		
Заміна уроків оплата	Середня освіта	V V H	18	3.4 тис.	17д.		
Одночасне навчання на магістратурі і бакалавраті		A Y H	24	275	18д.		

Рисунок 2.3 – Головна сторінка форуму Освіта.UA

Основною перевагою подібних платформ є можливість отримання відповідей на конкретні запитання від інших користувачів, включаючи студентів та випускників, які вже пройшли процес вступу. В таких спільнотах часто ведуться активні обговорення різних аспектів навчання та вступної кампанії, що допомагає отримати реальний досвід і практичні поради.

Однак інформація, надана на форумах, не завжди є достовірною. Оскільки відповіді можуть залишати будь-які користувачі, існує ризик отримання неточної або застарілої інформації. Крім того, структура форумів зазвичай не передбачає

швидкого пошуку даних – користувачеві доводиться переглядати численні коментарі або створювати власні запити, очікуючи на відповіді.

Переваги:

- можливість отримати актуальні відповіді від реальних людей;
- обговорення конкретних випадків та обмін досвідом;
- просте пояснення складної інформації від студентів та абітурієнтів.

Недоліки:

- відсутність гарантій достовірності інформації;
- велика кількість повторюваних запитань, що ускладнює пошук;
- відсутність централізованої бази знань.

Висновки до розділу 2

У даному розділі було проаналізовано основні джерела отримання інформації про вступну кампанію. Для розгляду використовувалися офіційні сайти вищих навчальних закладів, інформаційна система ЄДЕБО та спеціалізований форум Освіта.UA.

У результаті проведеного аналізу було виявлено, що кожне з наявних рішень виконує важливу роль у процесі вступної кампанії, оскільки містить достатню кількість необхідної для абітурієнта інформації. Проте кожна з цих систем також має свої недоліки, що значною мірою зменшують зручність та популярність їхнього використання. Особливо актуальними такі проблеми постають серед молоді, що в нас час потребують точної, актуальної та швидкої відповіді у зручному та інтерактивному форматі в комфортному для них середовищі.

3 ЗАСОБИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Розробка програмного забезпечення потребує використання спеціалізованих інструментів, які забезпечують ефективне створення та налагодження системи. Для реалізації інтелектуального чат-бота, який надає інформаційну підтримку абітурієнтам, було використано сучасні засоби та інструменти, що забезпечують зручність, легкість та масштабованість розробки, а також дозволяють здійснювати інтеграцію системи із зовнішніми сервісами. Даний розділ описує середовище розробки, засоби та інструменти розробки програмного забезпечення, що забезпечують зручне програмування, обробку природної мови, інтеграцію з базою даних та представлення інформації користувачам.

3.1 Вимоги до системи

Перед розробкою системи необхідно сформулювати низку функціональних та нефункціональних вимог до розроблюваного програмного забезпечення.

До функціональних вимог належить:

- інформування щодо процесу вступу, необхідних документів та вимог, графіку прийому приймальної комісії, контактів для зв'язку, доступних спеціальностей та освітніх програм, вступних іспитів та підготовка до них, корисних посилань та відповіді на типові запитання від абітурієнтів;
- пошук інформації на офіційних ресурсах, її структурування та виведення користувачу в зручному читабельному вигляді;
- надсилання зображень-інструкцій, що містять ключову інформацію щодо вступних процесів;
- надання посилань на офіційні джерела;
- обробка природної мови для розуміння запитів користувача за ключовими словами та фразами;

- реалізація багатокрокових діалогів для уточнення запиту користувача;
- інтерактивна взаємодія з користувачем завдяки slash-командам, кнопкам та випадним меню;
- виведення інформативного повідомлення у випадку нерозпізаного запиту користувача;
- нагадування про важливі дати та події, пов'язані з вступною кампанією;
- підключення до хмарної бази даних для зберігання та обробки даних;
- забезпечення адміністративного управління даними в БД для їхнього систематичного оновлення;
- перевірка правильності внесених даних перед записом до БД.

Нефункціональні вимоги:

- система має бути інтегрованою у популярну для абітурієнтів платформу Discord;
- інтерфейс взаємодії з системою повинен бути інтуїтивно зрозумілим для користувачів;
- система має стабільно функціонувати в режиму реального часу, забезпечуючи надання відповідей у будь-який час доби;
- програмне забезпечення має бути масштабованим та відкритим до подальшого розширення функціональних можливостей.

3.2 Програмне середовище

PyCharm – це повноцінне інтегроване середовище розробки (IDE) від компанії JetBrains, спеціалізоване на створенні програмного забезпечення мовою Python [7]. Воно забезпечує широкий набір інструментів для ефективної розробки, налагодження та тестування коду.

PyCharm пропонує інтелектуальне автодоповнення коду, що адаптується до стилю програміста, контексту проекту та правил іменування, а також підтримку

сучасних інструментів рефакторингу, перевірку відповідності PEP 8, миттєвий аналіз помилок та рекомендації щодо покращення коду.

Однією з його ключових переваг є зручна навігація по проєкту – розробник може швидко переміщатися до визначень змінних, функцій, класів або файлів, та переглядати всі місця їх використання в проєкті.

PyCharm також інтегрується з системами контролю версій, такими як Git та GitHub, має вбудований термінал, менеджери пакетів, підтримку роботи з Docker-контейнерами та можливість розгортання у віддаленому середовищі.

Усе це робить його ідеальним середовищем для повноцінної професійної розробки, дозволяючи зосередитися безпосередньо на реалізації функціоналу системи.

3.3 Мова програмування та бібліотеки

Для реалізації програмної частини проєкту обрано мову програмування Python, яка є однією з найпопулярніших мов. Вона вирізняється простим синтаксисом, високим рівнем абстракції та великим обсягом бібліотек для вирішення найрізноманітніших завдань — від аналізу даних до створення вебзастосунків і систем штучного інтелекту.

Python — це мова програмування, яка дозволяє швидко працювати та ефективніше інтегрувати системи [8]. Завдяки цьому вона стала особливо поширеною в освітніх та наукових проєктах, а також у сфері розробки чат-ботів, де важливо швидко створювати прототипи, підключати зовнішні сервіси та обробляти природну мову. Широка екосистема бібліотек дозволяє легко розробляти багатофункціональні застосунки без потреб писати складний код.

3.3.1 SpaCy та PyMorphy3

Для обробки природної мови (NLP) використано бібліотеки spaCy та PyMorphy3, які забезпечують аналіз та розпізнавання запитів користувачів.

SpaCy є однією з провідних бібліотек для обробки природної мови, яка підтримує багатомовні моделі, в тому числі українську мову [9]. Вона дозволяє швидко реалізувати як базову, так і розширену NLP-аналітику, надаючи такі інструменти:

- токенізація – розбиття тексту на окремі слова чи фрази;
- лематизація – перетворення слів до початкової форми;
- визначення частин мови – іменник, дієслово, прикметник тощо;
- виділення іменованих сутностей – імена людей, організації, дати, локації;
- аналіз залежностей між словами – визначення синтаксичних зв'язків.

SpaCy також забезпечує високу продуктивність, тому дозволяє обробляти великі обсяги текстів із мінімальними витратами часу.

PyMorphy3 – це морфологічний аналізатор для української мови, який дозволяє більш детально аналізувати граматичні форми слів [10]. Ця бібліотека надає можливість:

- визначити для кожного слова рід, число, відмінок, час, особу тощо;
- проводити лематизацію з урахуванням контексту;
- отримувати всі можливі граматичні варіанти слова.

PyMorphy3 дозволяє розпізнавати граматичні характеристики слів, що є особливо корисним при реалізації логіки розуміння запитів користувача.

Обидві бібліотеки активно підтримуються спільнотою, мають відкритий вихідний код та добре документовані, що спрощує інтеграцію в проєкт. Їхнє використання дозволить покращити якість взаємодії чат-бота з користувачем та побудувати гнучку систему обробки вхідних запитів, яка правильно розпізнає текст користувача, навіть якщо запит сформульований не зовсім стандартно.

3.3.2 Discord.py

Для розробки чат-бота на платформі Discord доцільним є використання бібліотеки Discord.py. Вона є офіційним асинхронним API-інтерфейсом для Python, що забезпечує повний доступ до функціональності Discord [11]. Вона є найбільш популярним інструментом для розробки ботів і широко використовується у навчальних та комерційних проєктах.

Бібліотека Discord.py базується на асинхронному програмуванні (async/await), що забезпечує ефективну обробку значної кількості подій у режимі реального часу без блокування виконання програм. Такий підхід особливо корисний для чат-ботів, які мають обробляти десятки запитів одночасно, надсилати відповіді, реагувати на взаємодію з інтерактивними елементами та динамічно оновлювати повідомлення.

Серед ключових можливостей бібліотеки:

- обробка подій: реагування на вхідні повідомлення, підключення чи вихід користувачів, натискання кнопок тощо;
- створення команд: підтримка сучасних slash-команд (/вступ, /спеціальності тощо), які сприяють швидкості користуванню;
- інтерактивність: можливість створювати випадні меню, інтерактивні кнопки, а також модальні вікна для введення даних користувачами;
- можливість роботи з вбудованими повідомленнями: форматування відповіді у вигляді приємних карток із заголовками, списками, посиланнями та зображеннями;
- гнучка система прав і доступів: дозволяє обмежити доступ до адміністративних команд лише для певних користувачів або ролей.

Discord.py легка у використанні завдяки об'єктно-орієнтованому дизайну, що спрощує створення команд, обробку повідомлень та події в режимі реального часу.

3.3.3 BeautifulSoup

Бібліотека BeautifulSoup призначена для розбору (парсингу) HTML- та XML-документів [12]. Вона дозволяє легко аналізувати, знаходити та витягувати потрібні елементи зі структури вебсторінок, що є корисним для обробки зовнішніх джерел інформації. Завдяки простому інтерфейсу BeautifulSoup може застосовуватися для реалізації веб-скрейпінгу, а саме – автоматизованого збору даних з офіційних сайтів закладів вищої освіти або інших ресурсів з актуальною інформацією про вступну кампанію.

Серед основних можливостей BeautifulSoup:

- навігація по DOM-структурі: пошук елементів за тегами, класами, ідентифікаторами, атрибутами або комбінацією умов;
- витягання тексту або атрибутів: зручне та швидке отримання текстового вмісту елементів або значення їхніх атрибутів, наприклад, витягти посилання з href;
- модифікація дерева документа: окрім читання, є можливість редагування структури HTML – додавання, зміни або видалення елементів;
- обробка некоректного або «брудного» HTML-коду: здатність розпізнавати і коректно обробляти навіть HTML-код із синтаксичними помилками чи невалідною розміткою/

BeautifulSoup дозволяє динамічно оновлювати інформацію, що забезпечує надання актуальних даних із сайтів.

3.3.4 Streamlit

Streamlit — це високорівневий Python-фреймворк з відкритим вихідним кодом, призначений для створення простих та інтерактивних вебінтерфейсів. Завдяки мінімалістичному синтаксису, Streamlit дозволяє швидко перетворити Python-скрипти на повноцінні вебзастосунки без потреби в знаннях HTML, CSS чи JavaScript [13].

Streamlit побудований такий чином, що кожне оновлення інтерфейсу відбувається в реальному часі під час запуску скрипта, що забезпечує високу динамічність взаємодії з користувачем. Завдяки декларативному стилю програмування, інтерфейс описується простими словами, наприклад `st.button()`, `st.selectbox()` тощо.

Основними перевагами Streamlit є:

- простота використання: розробник може створити повноцінний вебінтерфейс лише за допомогою Python, без залучення front-end технологій;
- швидкість розробки: мінімалістичний API дозволяє зменшити час на реалізацію функціональних прототипів;
- інтерактивність: Streamlit підтримує елементи взаємодії в реальному часі, такі як кнопки, випадаючі списки, форми тощо, що робить інтерфейси зручними для користувача;
- інтеграція з бібліотеками Python: легко поєднується з іншими бібліотеками Python, наприклад, для обробки текстових даних чи зображень;
- швидке розгортання: застосунки на Streamlit можна швидко й легко розгорнути на локальному сервері або в хмарі;

Завдяки цих характеристикам, Streamlit можна використати для створення адміністративної панелі керування даними у БД Cloud Firestore.

3.3.5 FPDF та XlsxWriter

Бібліотеки FPDF та XlsxWriter застосовуються для генерації документів у форматах PDF та Excel відповідно. Вони надають можливість автоматизовано формувати звіти, таблиці, інструкції чи інші документи з наданих даних.

FPDF – це легка бібліотека для створення PDF-файлів без потреб у зовнішніх залежностях [14]. Вона надає такі можливості:

- створювати документи з довільним розташуванням тексту, зображень та графіки;

- додавати таблиці, заголовки, колонтитули та сторінкову навігацію;
- встановлювати різні шрифти, стилі та кольори;
- працювати з багатосторінковими документами.

XlsxWriter – бібліотека для створення файлів Excel [15]. Вона надає такі можливості:

- створення робочих аркушів з багатьма таблицями;
- широкі можливості форматування комірок (кольори, шрифти, стилі меж тощо);
- вставку формул, діаграм та графіків;
- додавання умовного форматування.

Обидві бібліотеки підтримують роботу в автономному режимі, мають простий API та детальну документацію, що дозволяє швидко впровадити в проєкти, які потребують автоматичного створення документів у форматах PDF та Excel.

3.4 Cloud Firestore

Cloud Firestore – це гнучка, масштабована хмарна NoSQL база даних, розроблена компанією Google у межах екосистеми Firebase [16]. Вона забезпечує зберігання, синхронізацію та обмін даними у реальному часі між клієнтськими додатками та серверною частиною. Cloud Firestore добре підходить для розробки мобільних пристроїв, веб-сайтів та серверів від Firebase завдяки простій інтеграції з інтерфейсами на різних платформах.

Ключовими перевагами Cloud Firestore є:

- підтримка в реальному часі: система підтримує слухачі змін у режимі реального часу, що дозволяє миттєво оновлювати дані;
- автоматична синхронізація: зміни синхронізуються між усіма підключеними клієнтами;
- гнучка структура: дані організуються у вигляді колекцій та документів, що дозволяє зберігати вкладені об'єкти та списки у зручному форматі;

- маштабованість: база даних адаптована для високих навантажень і підтримує маштабування як у межах однієї програми, так і для глобальних проєктів;
- офлайн-доступ: мобільні додатки та вебзастосунки можуть працювати з базою даних в режимі офлайн, а внесені зміни автоматично синхронізуються після відновлення з'єднання;
- безпека: використання правил безпеки дозволяє керувати доступом до даних на основі автентифікації користувача та його ролі;
- інтеграція з іншими сервісами: база даних легко інтегрується з іншими сервісами через зручне API.

Завдяки поєднанню простоти, високої продуктивності та широкого функціоналу, Cloud Firestore є чудовим рішенням для розробки застосунків, що потребують централізоване зберігання даних.

Висновки до розділу 3

У даному розділі було визначено основні вимоги до програмного забезпечення інтелектуального чат-бота, зокрема щодо функціональності, зручності використання та інтеграції з популярною платформою Discord. Розглянуто програмне середовище для розробки такого чат-бота, яке забезпечить ефективну роботу над проєктом. Також обґрунтовано вибір мови програмування Python та відповідних бібліотек, які дозволяють реалізувати діалогову логіку, обробку природної мови, розпізнавання намірів і взаємодію з користувачем. Окрему увагу приділено хмарному сервісу Cloud Firestore, який використовується як надійне та маштабоване сховище даних, що забезпечує зберігання та оперативне оновлення структурованої інформації, необхідної для коректного функціонування системи.

4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

Процес реалізації інтелектуального чат-бота складається з розробки архітектури системи та написання програмних модулів. Програмна структура побудована на основі логічного поділу функцій, що дозволяє досягти гнучкості, зручності та розширюваності для підтримки проєкту. Окрему увагу приділено реалізації команд взаємодії, формуванню відповідей у різних форматах, а також використанню зовнішніх джерел інформації, а саме офіційних сайтів. Усі компоненти інтегруються в єдину систему, що забезпечує швидкий цілодобовий доступ до важливої вступної інформації та ефективно реагує на запити користувачів.

4.1 Архітектура системи

Архітектура даної системи складається з кількох основних компонентів: (рисунок 4.1)

- Discord-бот, що є основним інтерфейсом взаємодії з користувачами. Через цей канал абітурієнти формулюють свої запити, а бот надає відповіді у зручному та лаконічному форматі. Бо доступний на платформі Discord, оскільки вона є популярною серед цільової аудиторії – молоді;
- модуль обробки природної мови, що відповідає за розуміння змісту запитів. Саме цей компонент забезпечує інтелектуальність системи, дозволяючи сприймати користувацький ввід та варіативні запити, а також визначати його наміри;
- модуль зчитування даних з офіційних сторінок сайтів, який забезпечує отримання актуальної інформації про вступну кампанію. Даний компонент має вигляд скриптів та окремих функцій, які автоматично збирають, оновлюють та структуровано зберігають інформацію з офіційних сайтів, зокрема вищих навчальних закладів та державних платформ;

- база даних Cloud Firestore виконує функцію централізованого сховища структурованої інформації, а саме освітніх програм та посилань на сайти кафедр. Використання Firestore забезпечує маштабованість, гнучкість структури даних та зручну інтеграцію з іншими модулями через REST API;
- вебінтерфейс для адміністрування, що забезпечує можливість ручного керування вмістом БД, зокрема додавання, редагування або видалення освітніх програм. Даний інтерфейс дозволяє адміністратору системи працювати з інформацією без потреби прямого доступу до коду чи бази;
- модуль для формування звітів дозволяє генерувати документи у форматах PDF та Excel, що може бути корисним для аналізу інформації та розповсюдження даних.

4.1.1 Діаграма компонентів

Діаграма компонентів (component diagram) є одним із видів діаграм у нотації UML, що дозволяє відобразити внутрішню структуру програмної системи у вигляді окремих компонентів та зв'язків між ними. Вона ілюструє, з яких основних частин складається програмне забезпечення, як ці частини взаємодіють між собою, які інтерфейси реалізують, а також які залежності між компонентами існують.

На рисунку 4.1 продемонстровано діаграму компонентів для розроблюваної системи.

Як можна побачити на рисунку 4.1, система складається з двох користувацьких інтерфейсів та хмарної бази даних Cloud Firestore. Інтерфейс Discord містить в собі такі компоненти:

- Discord-bot: основний елемент, що надає взаємозв'язок користувачам із системою за допомогою тексту та команд;
- NLP: даний компонент відповідає за обробку користувацького вводу та визначення його намірів, що дозволить отримати й надати бажану йому інформацію;

- Parser: даний модуль призначений для отримання інформації із зовнішніх джерел, таких як офіційні сайти державних установ, та відповідна обробка цих даних для подальшого використання у системі.

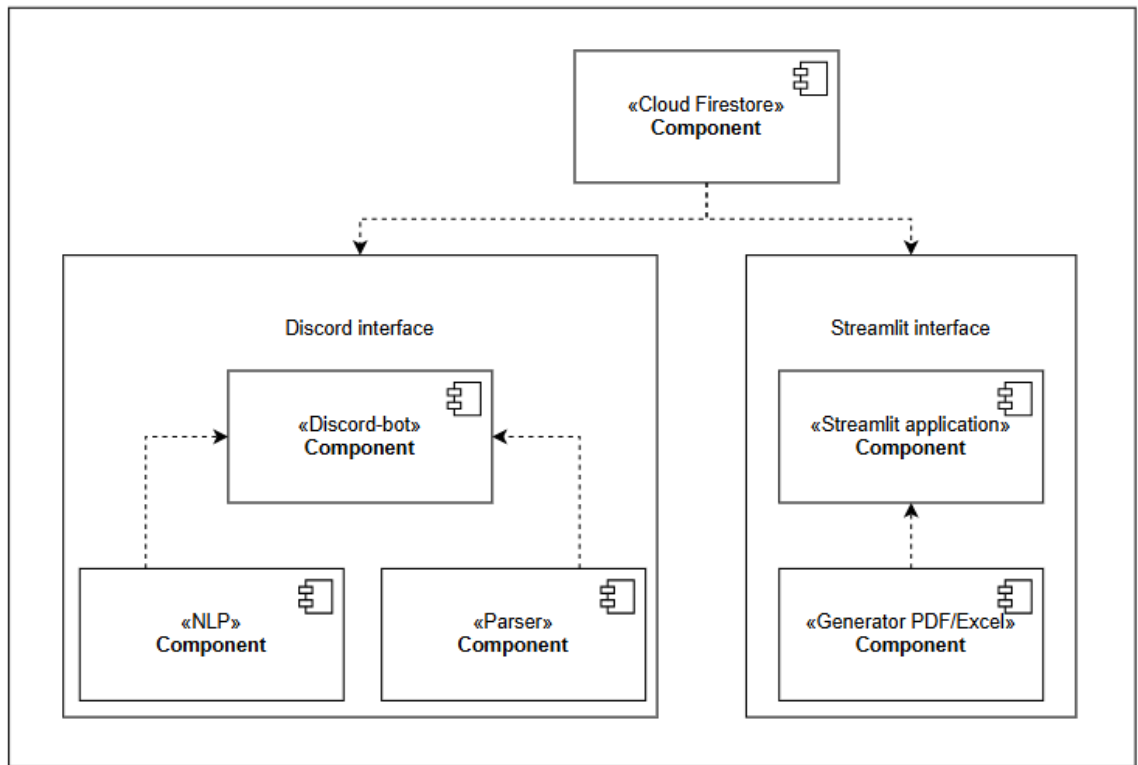


Рисунок 4.1 – Діаграма компонентів

Інтерфейс вебзастосунку представлений компонентом на базі Streamlit, який реалізує необхідний функціонал для зручного управління даними, що зберігаються у базі даних.

Cloud Firestore виступає як хмарне сховище, де зберігається структурована інформація про освітні програми. Цей компонент забезпечує доступ до даних для чат-бота під час обробки запитів користувачів. Оновлення, редагування чи додавання інформації відбувається через адміністративну частину системи – вебінтерфейс, що взаємодіє з Firestore.

4.1.2 Діаграма варіантів використання

Use Case diagram або діаграма варіантів (сценаріїв) використання є узагальненою моделлю відношення між акторами (користувачами) та прецедентами в системі. Кожен варіант використання визначає певний набір дій, який виконує система під час діалогу з актором. У нотації такої діаграми, актори представлені у вигляді чоловічків, а функції та дії – у вигляді кружечків, всередині яких записана назва виконуваної опції.

Для даної системи виділено дві підсистеми: Discord-бот для взаємодії з користувачем та вебзастосунок для адміністратора. Кожна з них реалізує окремі сценарії використання, тому для наочності та кращого розуміння було створено дві відповідні діаграми прецедентів (Use case), які відображають функціональні можливості кожної з підсистем.

На рисунку 4.2 представлено діаграму використання для користувача.

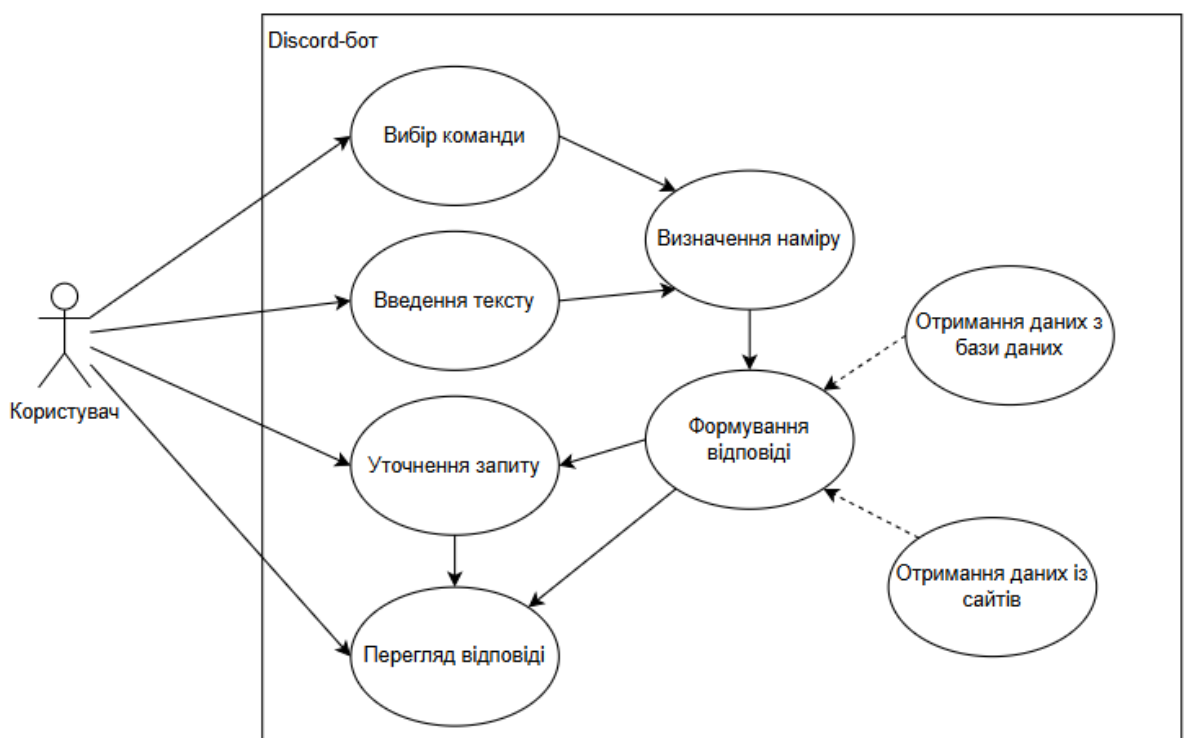


Рисунок 4.2 – Діаграма варіантів використання для користувача

Головним актором у даній системі є користувач, який взаємодіє із системою через інтерфейс та отримує потрібну йому інформацію. Для нього передбачено наступні ключові сценарії:

- вибір команди з меню вкладки Команди, що дозволяє здійснити конкретний та швидкий запит до системи;
- введення тексту дозволяє користувачеві спілкуватися з ботом природною мовою та здійснювати більш наповнені запити;
- уточнення запитів дозволяє отримати конкретні дані за потреби шляхом діалогового спілкування;
- перегляд відповідей передбачає отримання користувачем бажаної інформації у вигляді тексту, картинок та посилань.

Discord-бот в свою чергу виконує наступні дії, залежно від користувацького вибору:

- визначає намір користувача завдяки обробці природної мови, виділяючи ключові слова та порівнюючи їх із шаблонними словами та виразами;
- залежно від визначеного наміру, формує відповідь користувачу в зручному й читабельному для нього вигляді, використовуючи такі елементи як марковані списки, виділення тексту напівжирним шрифтом, а також графічні позначення у вигляді емодзі, що чудово підходять для візуального підсилення змісту повідомлення;
- отримує дані з хмарної БД Cloud Firestore, а саме перелік освітніх програм та відповідні посилання на сайти кафедр, що можуть бути використані при формуванні відповіді;
- отримує дані з офіційних сайтів, таких як кафедра інженерії програмного забезпечення в енергетиці, приймальна комісія НТУУ «КПІ імені Ігоря Сікорського» та український центр оцінювання якості освіти, забезпечуючи основну інформаційну наповненість бота.

Наступним елементом для детального ознайомлення та проектування є адміністративний вебзастосунок, діаграму прецедентів для якого можна переглянути на рисунку 4.3.

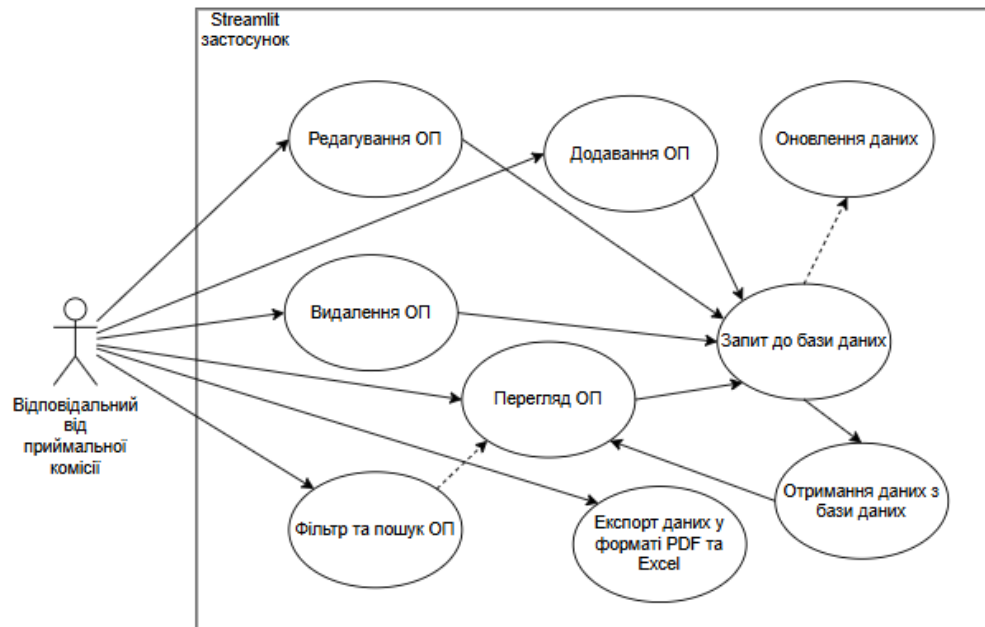


Рисунок 4.3 – Діаграма варіантів використання для адміністративного вебзастосунка

В даній діаграмі головним актором представлено відповідального від приймальної комісії, який наповнює базу даних. Серед основних доступних для нього дій є:

- додавання нової освітньої програм;
- редагування існуючої ОП;
- видалення обраної ОП з додатковим уточненням перед виконанням дії, щоб не втратити дані при помилковому кліку;
- перегляд повного списку освітніх програм;
- фільтр та пошук для виведення лише необхідних ОП, що дозволить виконувати швидко оновлення даних;
- завантаження даних у форматах PDF та Excel, що дозволить аналізувати та передавати інформацію у стандартному вигляді.

Натомість вебзастосунок виконує такі операції:

- виконує запит до БД – залежно від обраної користувачем дії формується відповідний запит з певними параметрами та надсилається до хмарної БД, використовуючи REST API;
- оновлює дані – після виконання певних запитів до БД, а саме додавання, редагування та видалення, виконується оновлення інформації;
- отримує дані з БД – після надсилання запиту до БД, програма отримує запитовані дані та виконує їхню обробку, після чого виводить користувачеві для перегляду. Якщо потрібна інформація у базі відсутня, програма інформує користувача про це.

4.1.3 Діаграма потоків даних

DFD diagram або діаграму потоків даних використовують для візуалізації процесів обробки даних (структурне проектування). На відміну від інших нотацій, DFD diagram візуально показує всі процеси з погляду даних.

Основні особливості та призначення DFD включають наступне:

- візуалізація потоків даних: DFD дозволяє представити потоки даних у системі, включаючи вхідні, вихідні та внутрішні потоки.
- рівні деталей: DFD може бути створена на різних рівнях деталей, від загального огляду до докладного розгляду окремих компонентів системи.
- прозорість: DFD надає можливість відстежувати шляхи переміщення даних і визначати, як різні частини системи взаємодіють між собою.
- призначення аналізу та проектування: DFD використовується для аналізу і проектування інформаційних систем.
- зручність для комунікації: DFD є ефективним інструментом комунікації між бізнес-аналітиками, програмістами, інженерами та іншими учасниками проекту.

Для візуального представлення розроблюваної системи створено DFD-діаграму, яку можна переглянути на рисунку 4.4.

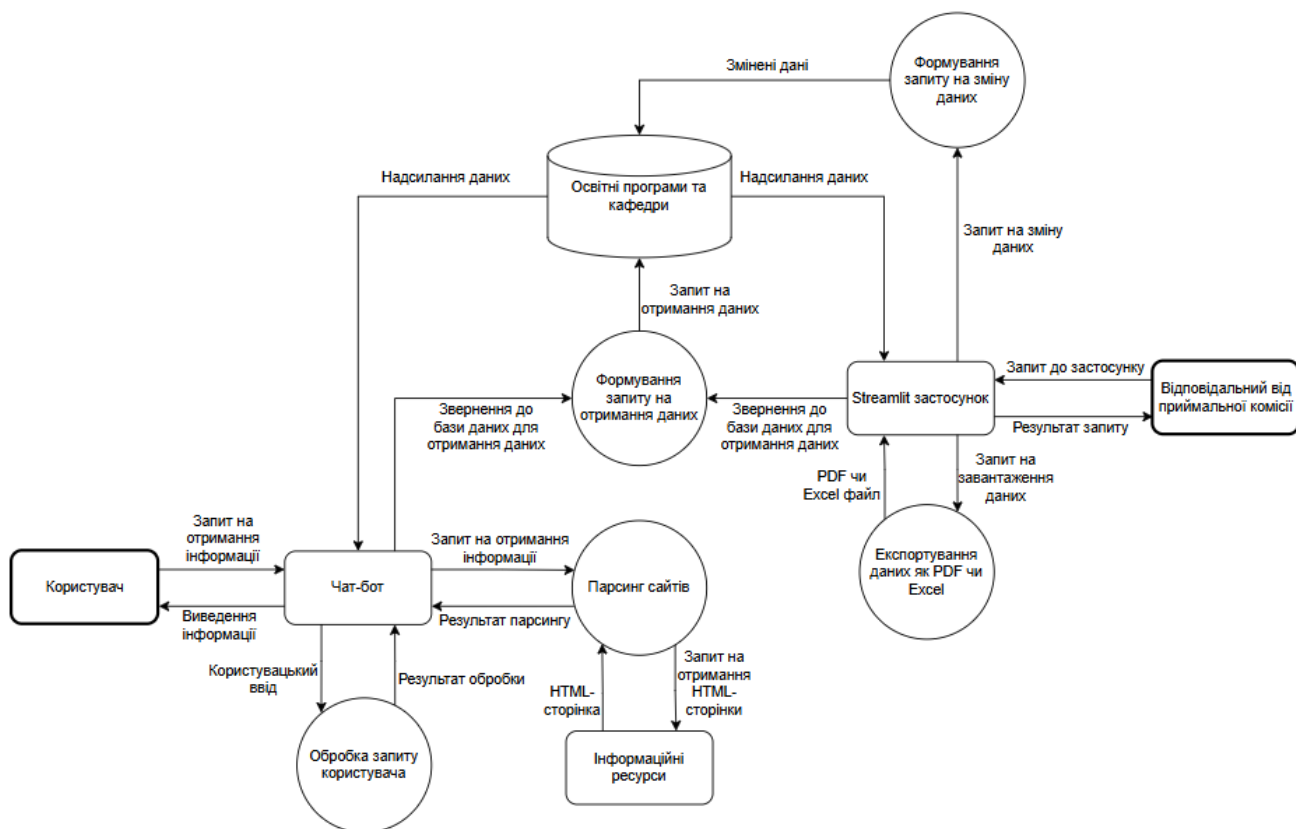


Рисунок 4.4 – Діаграма потоків даних

Основними компонентами діаграми потоків даних в даній системі є:

- користувач: виконує запит та отримує інформацію;
- чат-бот: інтерфейс взаємодії користувача із системою, що приймає користувацький ввід, виконує запити для пошуку бажаних даних та надає їх користувачу;
 - Streamlit застосунок: надає можливість отримати та оновити інформацію з бази даних;
 - відповідальний від приймальної комісії: взаємодіє через інтерфейс вебзастосунка для роботи з БД;
 - освітні програми та кафедри: хмарна БД, що зберігає та надає інформацію про освітні програми та відповідні сайти кафедр;
 - інформаційні ресурси: джерело достовірної та актуальної інформації щодо вступної кампанії.

Основними процесами є:

- обробка запитів користувача: отримує користувацький ввід та визначає його намір;
- парсинг сайтів: здійснює запити на сторінки сайтів та витягує необхідні текст, картинки та посилання;
- експортування даних: виконує експорт даних з БД у форматах PDF та Excel;
- формування запитів до БД: підготовка запитів до бази даних через механізм REST API.

4.2 Програмна реалізація

Програмна частина системи складається з окремих логічно взаємопов'язаних модулів, кожен із яких виконує чітко визначену функцію. Основу взаємодії з користувачем становить Discord-бот, реалізований із використанням відповідного API. Для обробки користувацького вводу застосовуються методи обробки природної мови, що дозволяють аналізувати зміст тексту та визначити намір користувача. Інформаційна складова забезпечується шляхом парсингу сайтів та через інтеграцію з базою даних Cloud Firestore, у якій зберігаються структуровані дані про освітні програми.

Модульна організація дозволяє розширювати функціонал за потреби без порушення цілісності системи.

4.2.1 Discord-бот

Одним з головних елементів даної системи є чат-бот, що інтегрований із платформою Discord. Цей компонент виконує роль інтелектуального помічника, який надає абітурієнтам необхідну інформацію в інтерактивному форматі. Він реалізований за допомогою бібліотеки discord.py, що забезпечує зручну взаємодію

з Discord API та дозволяє легко обробляти повідомлення чи команди. Основними функціональними можливостями бота є:

- виведення інформації щодо вступу;
- надання списку освітніх програм та спеціальностей за ключовими словами або кодом спеціальності;
- отримання детальної інформації про конкретні спеціальності та освітні програми;
- формування відповідей на часті запитання за допомогою випадного меню;
- надсилання зображень-інструкцій щодо необхідних для вступу документів та вимог до мотиваційного листа;
- виведення корисних посилань на додаткові ресурси, що містять демоваріанти іспитів, тести минулих років, онлайн-тренажерів;
- надання інформації щодо структури вступних тестів та їх програм за ключовими словами;
- нагадування про важливі дати пов'язані із процесом вступної кампанії;
- інтерактивна взаємодія з користувачем через slash-команди та повідомлення з кнопками;
- підтримка багатокрокових діалогів для уточнення деталей запиту користувача.

Завдяки цьому, бот забезпечує швидкий доступ до важливої для вступників інформації, з можливістю детальніше ознайомитися з нею на офіційних сторінках сайтів.

4.2.2 Обробка запитів користувача

Основна взаємодія користувача з ботом відбувається шляхом введення запитів природною мовою. Такі запити спочатку проходять попередню обробку у вигляді очищення від зайвих пробілів, нормалізації тексту, тобто переведення його

до стандартної форми, токенизації та лематизації. Після цього система визначає намір користувача на основі ключових слів та шаблонів. Залежно від наміру, запускається відповідна функція чи процес, наприклад, отримання переліку спеціальностей із офіційного сайту, інформації щодо вступу чи відповідь на часте запитання.

4.2.3 Зчитування даних із сайтів

Оскільки чат-бот надає достовірну інформацію з офіційних джерел, постає завдання отримання даних зі сторінок вебсайтів. Для цього використано бібліотеку BeautifulSoup, яка дозволяє здійснювати аналіз HTML-сторінок та видобувати необхідну текстову інформацію, посилання або зображення. Вона застосовується для отримання актуального списку спеціальностей та освітніх програм, процесу вступу та вимог, графіку роботи приймальної комісії та контактів для зв'язку, інформації про гранти, корисних посилань та тестів минулих років для підготовки до іспитів. Після зчитування із сайтів, дані проходять попередню обробку та зберігаються у певному структурному вигляді (наприклад списку). Це дозволяє зручно використовувати їх при подальших діях системи, наприклад при формуванні відповіді користувачу.

4.2.4 Cloud Firestore

База даних Firestore використовується як централізоване сховище інформації про освітні програми. Вона складається з колекції programs, що містить список документів, які представляють освітні програми. Кожен документ містить кілька полів: назва освітньої програми, посилання на сайт кафедри та назва інституту чи факультету, на якому цю освітню програму викладають. Доступ до бази даних здійснюється за допомогою REST API, що дозволяє виконувати операції читання, створення, оновлення та видалення документів безпосередньо з програми без

потреби в окремому SDK. Для запитів використовуються унікальні ідентифікатори документів, що дозволяють звертатися до конкретних освітніх програм для їхньої модифікації.

4.2.5 Вебінтерфейс для адміністрування

Чат-бот використовує базу даних лише для отримання необхідної інформації. Проте дані в ній потребують оновлення з певним часом, тому для зручності управління цими даними, було створено вебзастосунок за допомогою інструменту Streamlit. Він дозволяє адміністратору безпосередньо взаємодіяти з базою даних Firestore через зручний інтерфейс, не вдаючись до написання коду чи скриптів. Даний застосунок складається з кількох функціональних блоків:

- перегляд даних – виведення списку усіх освітніх програм, що записані в базі даних, з можливістю пошуку за назвою та сортуванням за інститутом чи факультетом для зручного перегляду та пошуку конкретних програм;
- редагування наявних освітніх програм – переглядаючи списки, адміністратор одразу може внести необхідні зміни щодо конкретних даних та відправити їх безпосередньо в базу даних;
- видалення освітніх програм – адміністратор може видалити непотрібні дані з підтвердженням такої дії, після чого будуть внесені зміни до хмарної бази даних;
- додавання освітніх програм – передбачено форму введення даних, що дозволяє створювати новий запис. При підтвердженні виконується валідація внесених даних, а саме чи поля не порожні та чи містить коректне посилання, і лише після того відправляються в базу даних, якщо було успішно пройдено валідацію;
- завантаження даних – передбачена можливість отримання даних у зручних для подальшого використання форматах, а саме PDF та Excel.

Даний застосунок можна легко розгорнути на локальному сервері, до того ж він не потребує складних налаштувань.

4.2.6 Формування файлів

У вебінтерфейсі передбачена можливість завантаження файлів у певних форматах, що дозволяє зручно зберігати, аналізувати та передавати структуровану інформацію про освітні програми. Для цього було створено функції, що створюють файли, заповнюють їх даними та зберігають у байтах для передачі у завантажувальні кнопки.

Для створення PDF-документів використовується бібліотека `fpdf`, що дозволяє генерувати документи з потрібною структурою та форматуванням. Спочатку дані перетворюються у тип `DataFrame`, після чого записуються у PDF-файл у вигляді таблиці, з трьома стовпцями, що відповідають полям з бази даних. Для побудови таблиці малюються відповідні лінії, для яких, за допомогою математичних обчислень, визначаються початкові точки. Враховується також перенесення слів на новий рядок, якщо вони не поміщаються в стовпець, та перенесення всього рядка на нову сторінку, якщо він не поміщається на одну.

Формування Excel-файлу відбувається з використанням `xlsxwriter`, яка дозволяє задавати стиль, створювати таблиці, формувати заголовки, додавати автофільтри та фіксовані рядки. Як і для PDF, Excel-документ складається з трьох стовпців, що відповідають полям з бази даних.

Висновки до розділу 4

У даному розділі детально описано процес реалізації програмної системи – від побудови архітектури у вигляді діаграм до розробки функціональних модулів.

Для кращого розуміння структури та взаємодії елементів системи було створено діаграми прецедентів (Use case), які відображають основні сценарії

використання вебзастосунка для адміністратора та чат-бота залежно від вибору певних дій чи введенні варіативних запитів, а також діаграму потоків даних (DFD), що демонструє логіку обробки запитів, передачу інформації між модулями та взаємодію з базою даних. Крім цього побудовано діаграму компонентів, що дає змогу краще зрозуміти архітектуру системи, а саме з яких вона елементів складається та як ці компоненти взаємозв'язані між собою.

Структура системи побудована на модульному підході, що забезпечує її масштабованість, зручність супроводу та можливість подальшого розширення функціоналу.

Кожен компонент системи виконує визначену роль: Discord-бот реалізовує діалоговий інтерфейс, модулі обробки природної мови та парсингу забезпечують коректне розпізнавання запитів та динамічне оновлення даних відповідно, а база даних відповідає за централізоване зберігання інформації. Адміністративний вебінтерфейс дозволяє легко керувати вмістом БД, а модуль генерації звітів підвищує зручність перегляду даних та їх розповсюдження.

Реалізована система демонструє ефективну інтеграцію сучасних технологій та сервісів для вирішення завдання автоматизованого надання актуальної інформації в інтерактивному форматі.

5 РОБОТА КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ

Зручний, приємний та легкий у користуванні інтерфейс є основою програмної системи, що взаємодіє з користувачами. Тому важливим було створення простого та зрозумілого механізму отримання потрібної інформації шляхом використання інтерактивних елементів та запитів природною мовою.

5.1 Взаємодія з ботом

Інтелектуальний чат-бот інтегрований у платформу Discord, що створює звичне середовище спілкування для молоді. Для початку роботи достатньо додати програму (бота) до своїх програм. На рисунку 5.1 можна побачити вікно додавання бота до своїх програм.

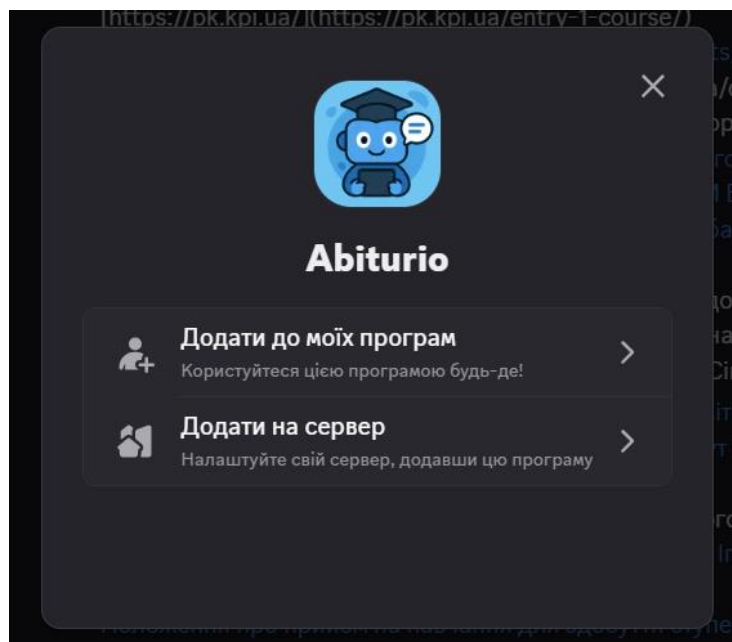


Рисунок 5.1 – Додавання бота до своїх програм

Потім для користувача відкриється ще одне вікно, де потрібно натиснути кнопку «Авторизувати» (рисунок 5.2). Після цього бот готовий до використання та чекатиме на повідомлення від користувача.

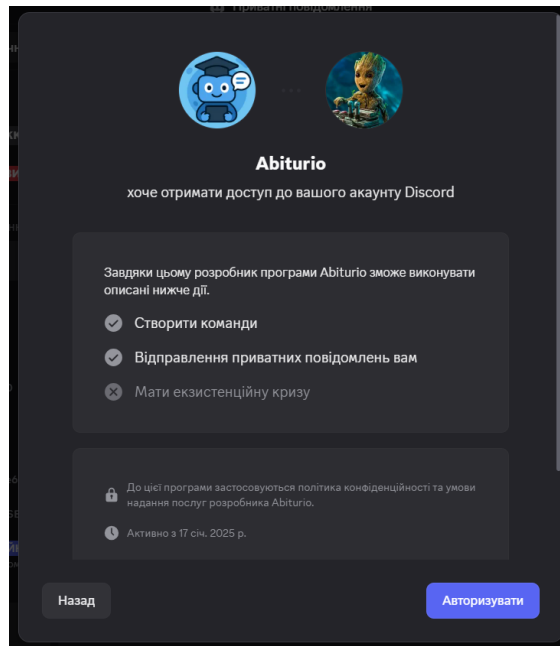


Рисунок 5.2 – Вікно авторизації бота

Користувач може взаємодіяти з ботом за допомогою текстових повідомлень або через slash-команди (наприклад `/вступ` чи `/документи`), які автоматично підказуються інтерфейсом Discord. На рисунку 5.3 представлено перелік доступних команд бота.

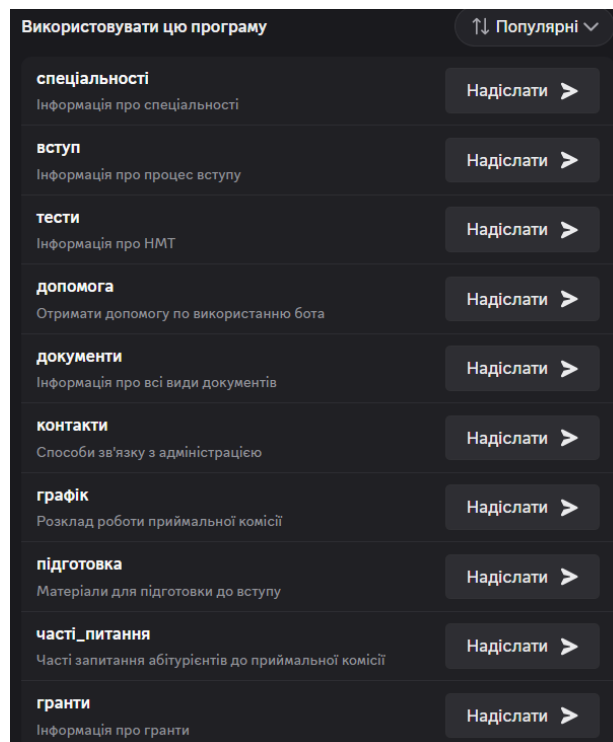


Рисунок 5.3 – Доступні команди чат-бота

На рисунку 5.4 продемонстровано взаємодію з чат-ботом за допомогою текстових повідомлень.

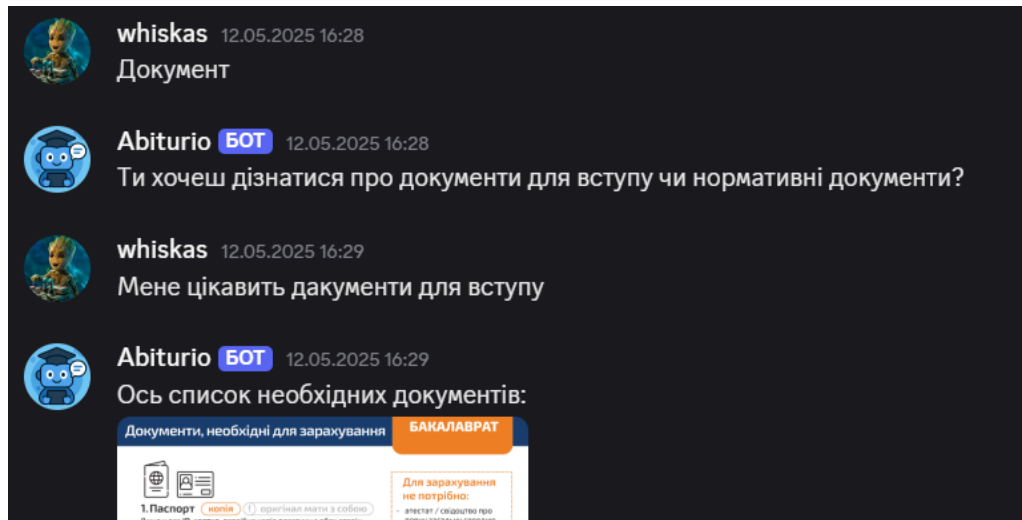


Рисунок 5.4 Взаємодія з ботом текстовими повідомленнями

Після введення тексту чи вибору команди користувач отримує структуровану відповідь у вигляді тексту, зображень чи інтерактивних елементів – кнопок та випадаючих списків (рисунки 5.5 та 5.6), а у разі нерозпізнання запиту отримує інформативне повідомлення про це (рисунок 5.7).

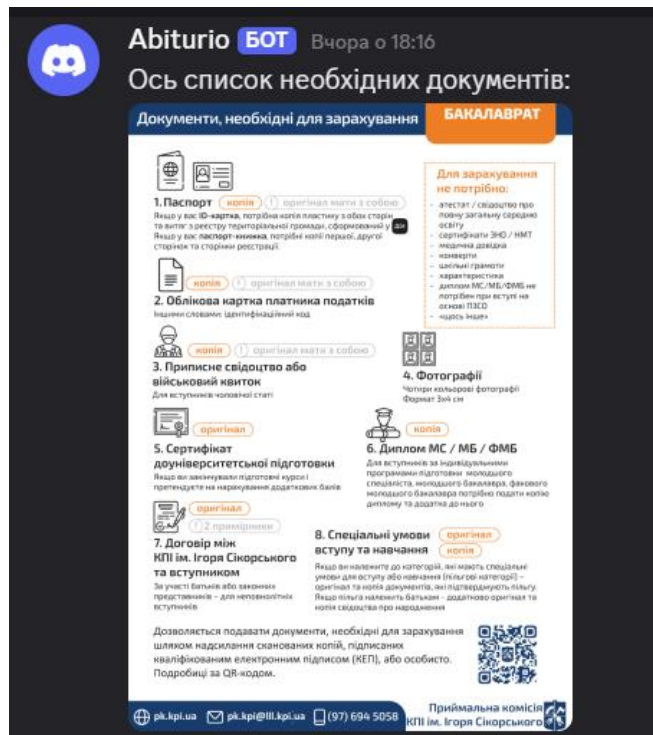


Рисунок 5.5 – Відповідь бота у вигляді тексту та зображень

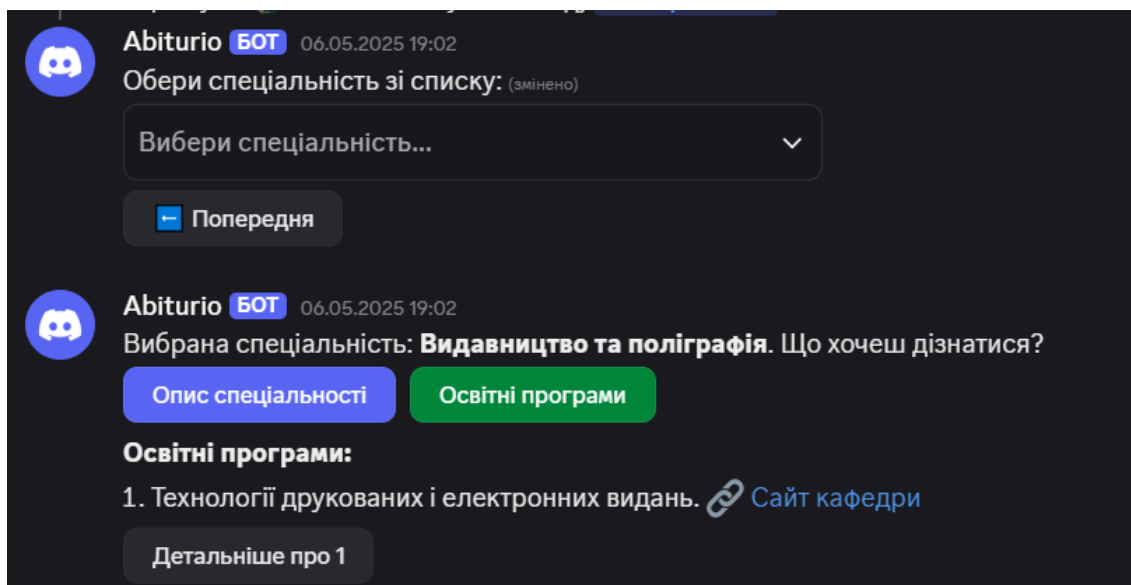


Рисунок 5.6 – Відповідь бота у вигляді кнопок

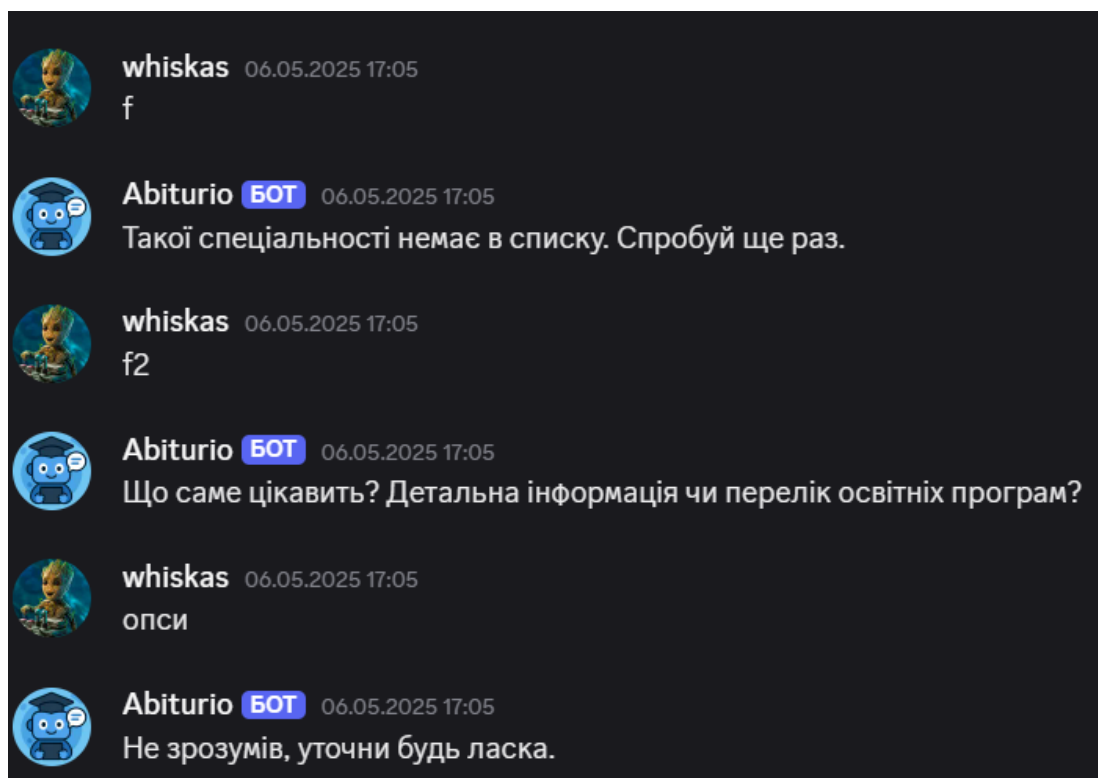


Рисунок 5.7 – Повідомлення про нерозпізнаний запит

Крім того, чат-бот нагадує користувачам про важливі дати пов'язані із процесом вступної кампанії, надсилаючи їм сповіщення за день до настання цієї дати (рисунок 5.8).

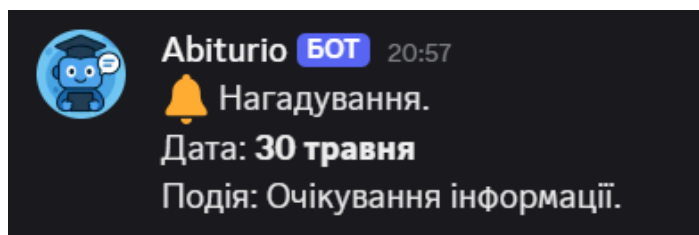


Рисунок 5.8 – Нагадування про важливі дати

Основними сценаріями використання є:

- ознайомлення з поняттям вступу та інструкцією: бот надає загальну інформацію про етапи, вимоги та терміни вступної кампанії, а також надає корисну інструкцію у вигляді гайду щодо процесу вступу (рисунок 5.9);



Рисунок 5.9 – Путівник по вступу

- перегляд спеціальностей: користувач може переглянути повний список наявних спеціальностей, а також дізнатися детальну інформацію щодо кожної з них за допомогою коду чи назви спеціальності (рисунок 5.10);

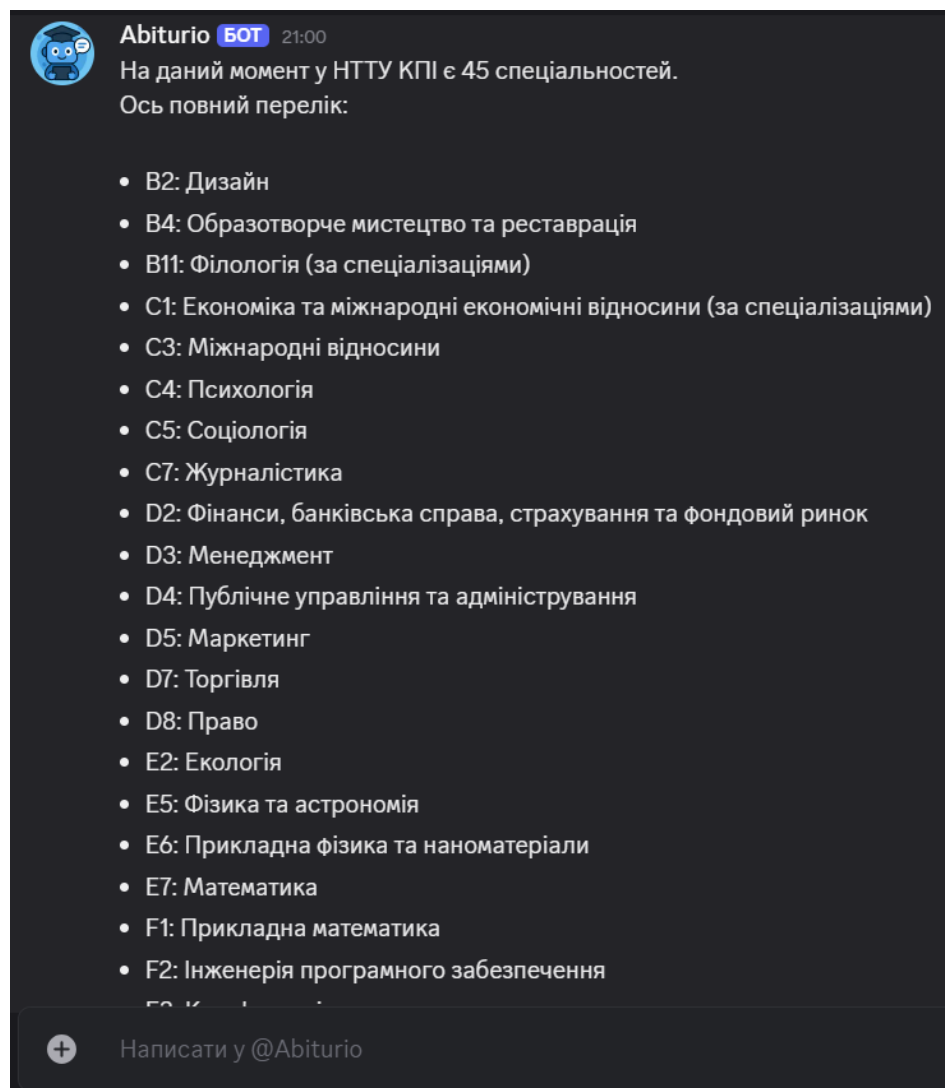


Рисунок 5.10 – Перелік спеціальностей

- перегляд освітніх програм: у межах кожної спеціальності є перелік доступних освітніх програм, про які користувач також може дізнатися більше, або перейти на офіційний сайт кафедри для детального ознайомлення (рисунок 5.11);
- отримання інструкцій у вигляді зображень: користувач може отримати необхідну інформацію щодо документів для вступу, вимог до мотиваційного листа та інформації про гранти у вигляді одного чи кількох зображень, в яких коротко та лаконічно пояснено ключові моменти, на які варто звернути увагу вступнику (рисунок 5.5);

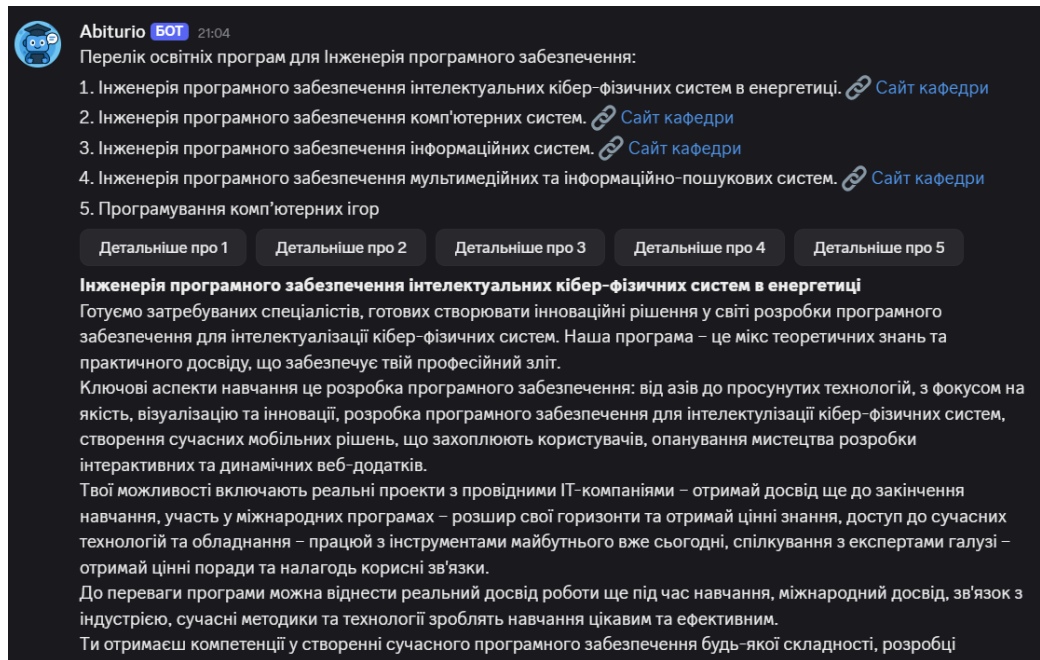


Рисунок 5.11 – Перелік освітніх програм за обраною спеціальністю

- часті запитання: для користувача відкрита можливість ознайомитися з переліком частих запитань обраної категорії та отримати відповідь на будь-яке з них, обравши його з випадного меню або шляхом введення тексту запитання чи номера (рисунок 5.12);

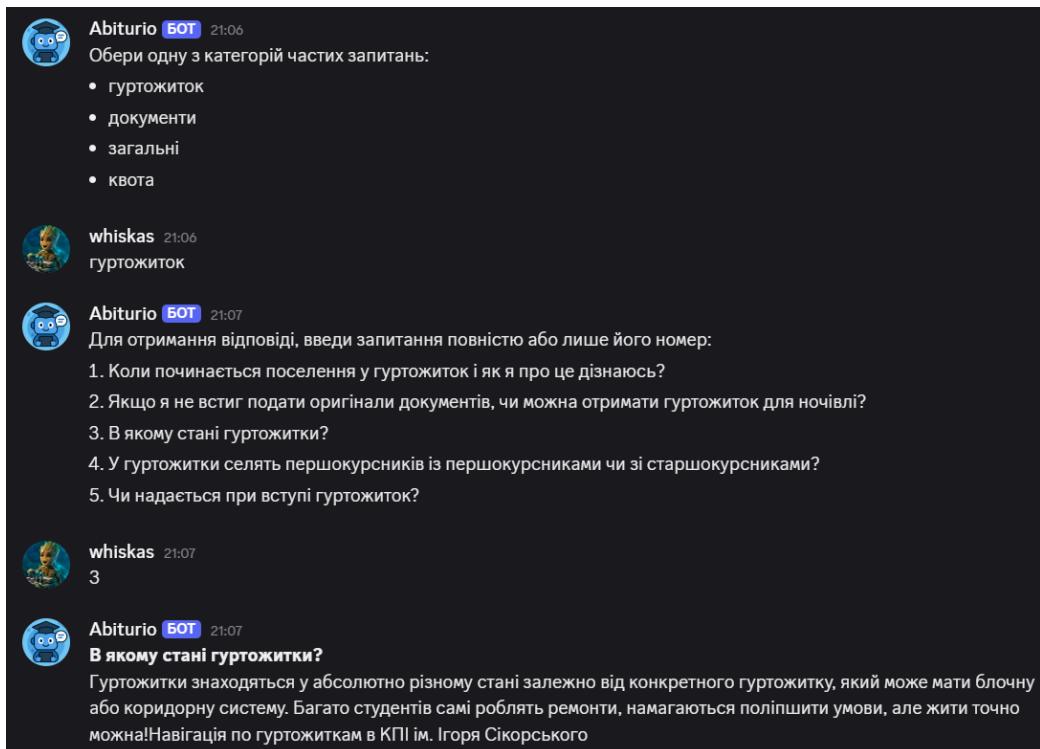


Рисунок 5.12 – Часті запитання

- графік роботи та контакти: користувач може отримати інформацію щодо графіку роботи приймальної комісії та контактів для зв'язку, а саме номери телефонів та електронні пошти (рисунки 5.13 та 5.14);

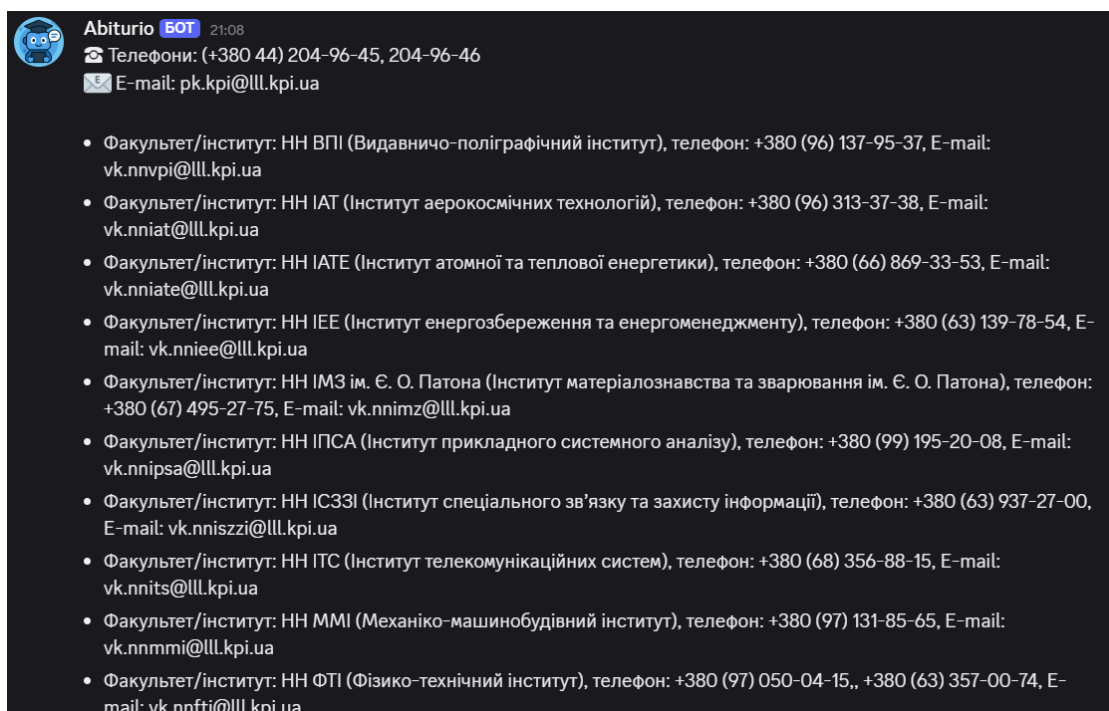


Рисунок 5.13 – Контакти приймальних комісій інститутів/факультетів

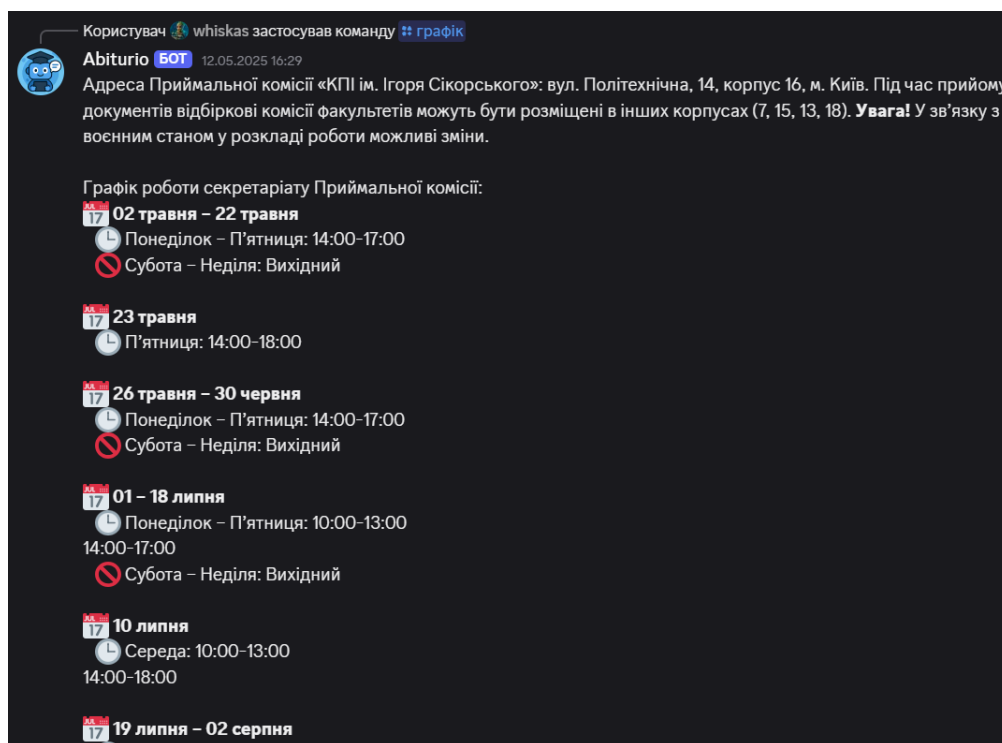


Рисунок 5.14 – Графік роботи приймальної комісії

- тести: чат-бот надає як загальну інформацію щодо тестування, так і більш детальну для кожного предмета. для користувача є можливість переглянути програму обраного тесту та його складники (рисунок 5.15);

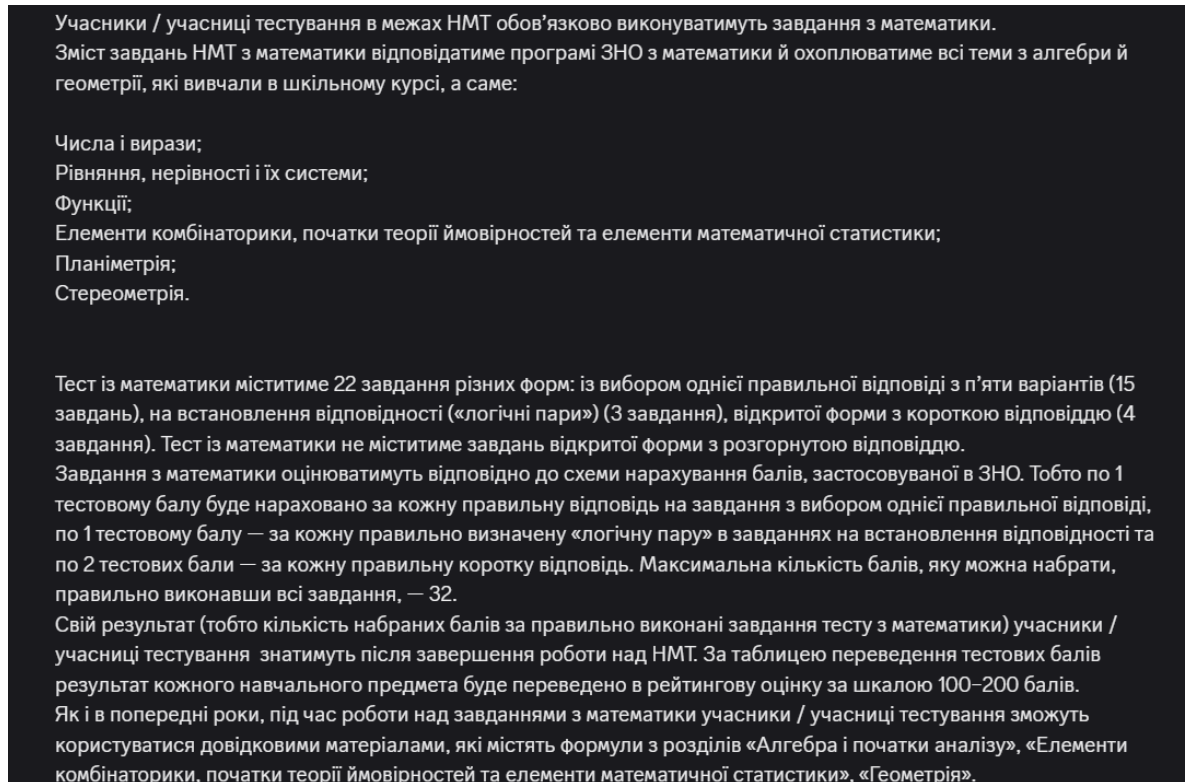


Рисунок 5.15 – Інформація про тест з математики

- підготовка: не менш важливим етапом у тестуванні є підготовка до нього, тому користувач може отримати перелік корисних посилань для тривалої підготовки до іспитів та посилання на демоваріанти і тести минулих років, щоб перевірити свої навички (рисунок 5.16);

- посилання на офіційні сайти: при надсиланні більшості повідомлень, бот також надає кнопку-посилання на офіційний ресурс для детального ознайомлення з інформацією (рисунок 5.17).

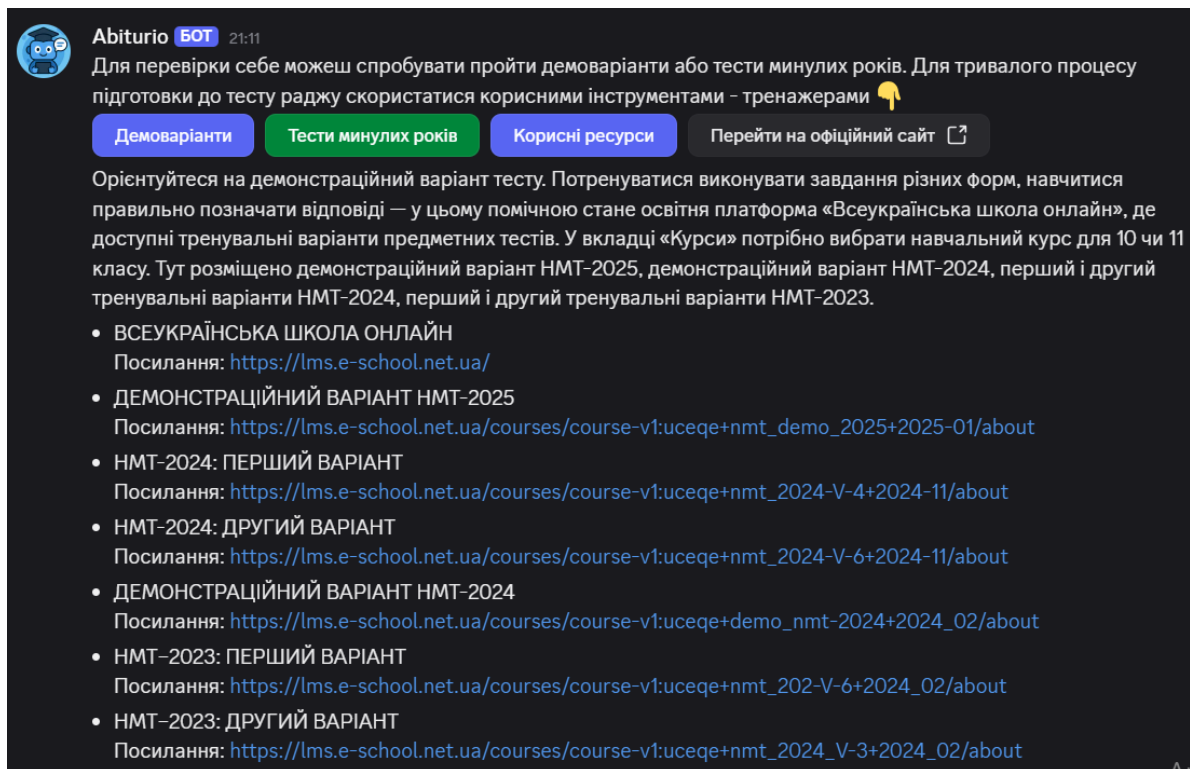


Рисунок 5.16 – Матеріали для підготовки до тестування

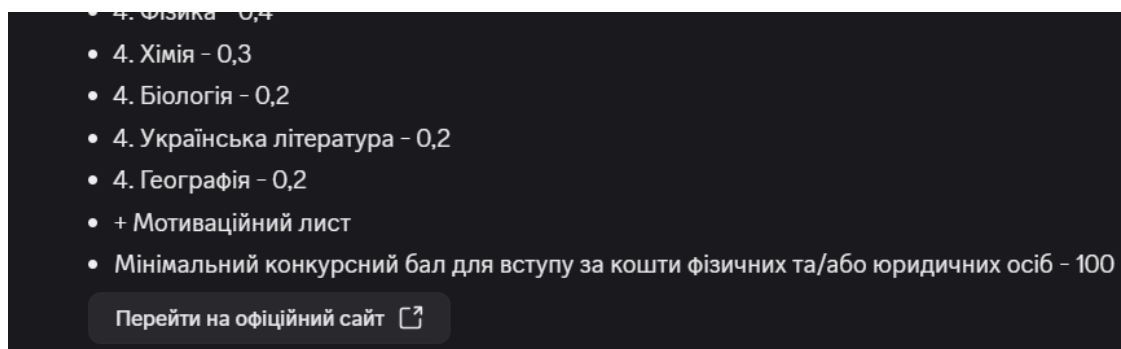


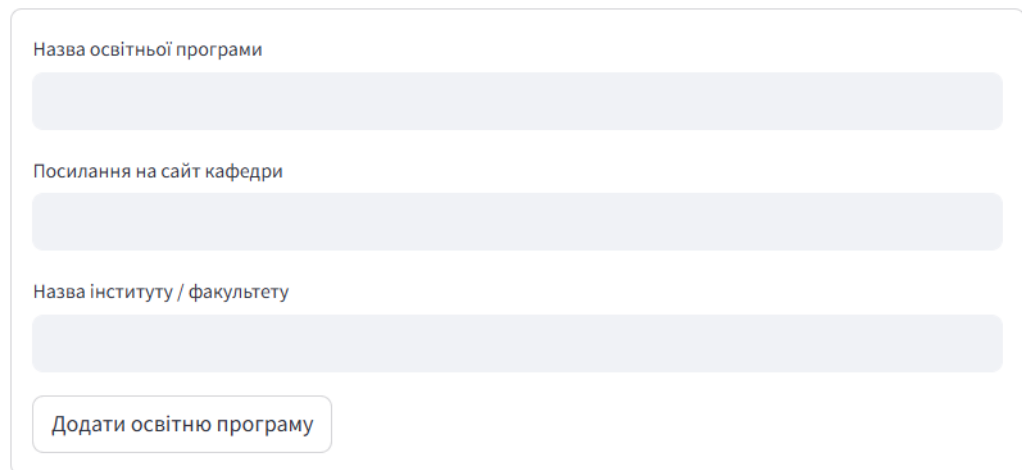
Рисунок 5.17 – Кнопка-посилання для переходу на офіційну сторінку

Даний чат-бот реалізовує як підтримку інтерактивних елементів, так і обробку природної мови, що робить його зручним у використанні для осіб з різними вподобаннями. Він надає користувачам широкий спектр інформації, пов'язаної з вступною кампанією: від подачі необхідних документів до наявних спеціальностей, від поняття вступу до структури кожного з тестів. Це є особливо корисним для абітурієнтів, які не завжди здатні знайти й обробити таку інформацію на офіційних сайтах чи в інших джерелах.

5.2 Взаємодія із вебзастосунком для адміністрування

Іншим модулем системи є вебзастосунок, створений за допомогою інструменту Streamlit. Він дозволяє управляти даними в базі даних Cloud Firestore в зручному інтерфейсі без будь-яких знань написання коду чи спеціалізованого навчання користування таким застосунком. Інтерфейс інтуїтивно-зрозумілий і легкий у навігації та передбачає повний перелік необхідних функцій для управління даних про освітні програми. Його реалізацію можна переглянути на рисунках 5.18 та 5.19.

Освітні програми



Назва освітньої програми

Посилання на сайт кафедри

Назва інституту / факультету

Додати освітню програму

Фільтрація та пошук

Рисунок 5.18 – Додавання нової освітньої програми

У формі створення нової освітньої програми, адміністратор вносить дані: назву програми, посилання на сайт кафедри та назву інституту чи факультету. Після чого натискає на кнопку «Додати освітню програму» та отримує повідомлення про успішне додавання даних до бази або повідомлення про помилку конкретних внесених даних. На рисунку 5.20 можна побачити некоректно внесений текст посилання та повідомлення від програми, що вказує на цю проблему.

У розділі «Фільтрація та пошук» користувач може здійснити пошук серед освітніх програм, або відфільтрувати їх за назвою інституту чи факультету. За бажанням, можна здійснити сортування списку за алфавітом.

Фільтрація та пошук

Пошук за назвою

Фільтр за інститутом / факультетом

Усі ▼

Сортувати за назвою (А-Я)

Список освітніх програм

Клінічна інженерія ▼

Технології захисту навколишнього середовища та гуманітарне розмінвання ▼

Технології друкованих і електронних видань ▼

Рисунок 5.19 – Перегляд наявних освітніх програм

Освітні програми

Назва освітньої програми

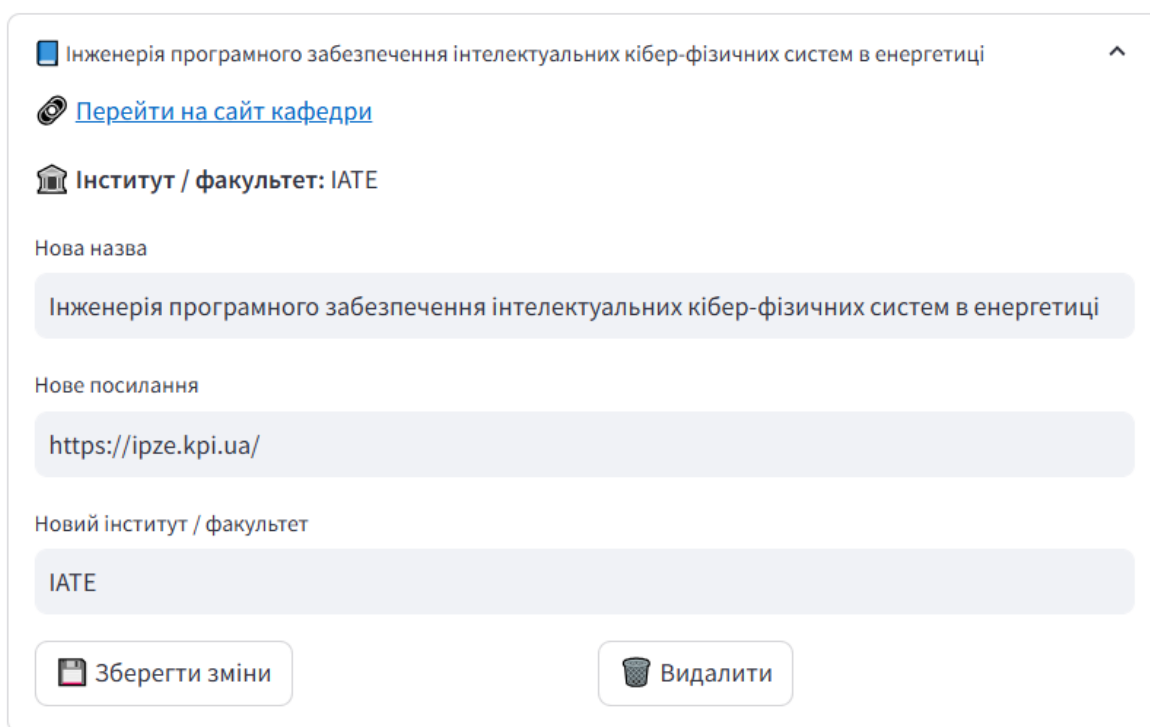
Посилання на сайт кафедри

Назва інституту / факультету

Будь ласка, введіть коректне посилання на сайт кафедри (наприклад, <https://example.com>).

Рисунок 5.20 – Повідомлення про невірно введені посилання

Розділ «Список освітніх програм» виводить перелік освітніх програм, що є у базі даних, з урахуванням значень попереднього розділу фільтрації та пошуку. Він дозволяє не лише переглядати список програм, а й вносити зміни до певних полів та зберегти їх за допомогою кнопки збереження (рисунок 5.21).



Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці

[Перейти на сайт кафедри](#)

Інститут / факультет: IATE

Нова назва

Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці

Нове посилання

<https://ipze.kpi.ua/>

Новий інститут / факультет

IATE

Зберегти зміни

Видалити

Рисунок 5.21 – Розширений перегляд конкретної спеціальності

Окрім редагування даних, користувач також може видалити певну освітню програму за допомогою кнопки «Видалити», після натискання на яку буде виведено повідомлення для підтвердження видалення даної програми у разі помилкового натискання на кнопку.

Крім того застосунок надає можливість адміністратору завантажити як повний список освітніх програм, так і з вказаними фільтрами. Для цього є кнопки «Експортувати як PDF» та «Експортувати як Excel», що дозволяють користувачу завантажити файли у форматі PDF та Excel відповідно. Після завантаження користувач отримує повідомлення про отримані файли (рисунок 5.22) та може переглянути їхній вміст.

Нещодавня історія завантажень ×

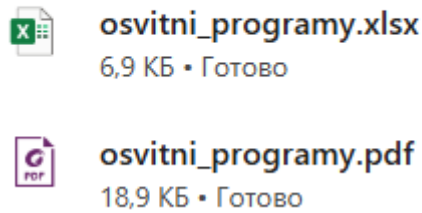


Рисунок 5.22 – Сповіщення про завантажені файли

Такі документи можуть використовуватися для зберігання, аналізування та передавання структурованої інформації про освітні програми.

Висновки до розділу 5

У даному розділі було продемонстровано взаємодію користувача із системою, а саме початок роботи, сценарії використання та результати запитів.

Реалізований чат-бот демонструє високий рівень зручності у взаємодії з користувачем. Завдяки використанню сучасних елементів інтерфейсу – таких як slash-команди, кнопки та випадаючі списки – користувач може легко орієнтуватися у функціоналі бота та отримувати потрібну інформацію без зайвих зусиль. Також бот відкритий до спілкування природної мови, що робить його адаптивним під різні потреби користувачів.

Окрім взаємодії з боку кінцевого користувача, важливу роль у функціонування системи відіграє вебзастосунок для адміністратора. Завдяки інтуїтивно-зрозумілому інтерфейсу він дає змогу швидко переглядати, додавати, редагувати та видаляти освітні програми, а також формувати звітні документи у форматах PDF та Excel. Такий підхід дозволяє спростити оновлення бази даних на сервері та знизити ризик помилок, пов'язаних з ручним оновленням даних.

ВИСНОВКИ

У процесі написання дипломної роботі було ознайомлено з літературними джерелами та офіційними сайтами для вступної кампанії. Розглянуто наявні рішення та визначено їх основні переваги й недоліки. Це дозволило проаналізувати особливості інформаційної підтримки абітурієнтів, визначити основні функціональні та нефункціональні вимоги до системи та обрати оптимальні засоби та інструменти для реалізації програмного забезпечення з урахуванням специфіки поставленої задачі.

В результаті було розроблено інтелектуального чат-бота для платформи Discord, який надає користувачам інформацію щодо процесу вступу, зокрема: перелік спеціальностей, деталі освітніх програм, відповіді на часті запитання, посилання на навчальні матеріали та офіційні сайти, інструкції та вимоги до документів. Крім того, було реалізовано підтримку багатокрокових діалогів для уточнення користувацького запиту та інформування у випадку нерозуміння. Також впроваджено інтерактивну взаємодію через slash-команди, випадні меню та кнопки. Забезпечено зберігання даних про освітні програми та відповідні посилання на сайти кафедр у хмарній базі Cloud Firestore. У ході проєкту було також розроблено вебзастосунок для управління даними та генерації звітних файлів у форматах PDF та Excel.

Створення такої системи здійснювалася у середовищі розробки Pycharm з використанням мови програмування Python. Для реалізації було використано низку програмних модулів та інструментів, а саме discord.py, spaCy та rymorphy3, BeautifulSoup, fpdf та XlsxWriter, Streamlit та Cloud Firestore.

Даний продукт є зручним у використанні та при потребі може бути розширеним новими функціями чи модулями або інтегрованим у інші платформи.

Дана робота апробована на Всеукраїнській науково-технічній конференції молодих вчених, аспірантів та студентів «Стан, досягнення та перспективи інформаційних систем і технологій [17].

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Зуб Х. В., Жежнич П. І. Аналіз ефективності вступної кампанії закладів вищої освіти України та способів її підвищення шляхом впровадження інформаційних технологій. *Вісник ВПІ*. 2022. Вип. 3. С. 52–59.
<https://doi.org/10.31649/1997-9266-2022-162-3-52-59>
2. Макоєдова В. О. Інформатизація процесів вступної кампанії в закладах вищої освіти. *Технічні науки та технології*. 2023. № 1(31). С.90-97.
[https://doi.org/10.25140/2411-5363-2023-1\(31\)-90-97](https://doi.org/10.25140/2411-5363-2023-1(31)-90-97)
3. Популярність чат-ботів: веб-сайт. URL:
<https://www.techradar.com/pro/the-beginning-of-the-end-more-of-us-are-moving-away-from-google-towards-tiktok-and-ai-chatbots-as-research-reveals-that-the-golden-era-of-search-engines-may-well-be-over> (дата звернення 08.05.2025)
4. Офіційний сайт приймальної комісії КПП. URL: <https://pk.kpi.ua/> (дата звернення 20.05.2025)
5. Єдина державна електронна база з питань освіти. URL:
<https://vstup.edbo.gov.ua/> (дата звернення 20.05.2025)
6. Форум Освіта.UA. URL: <https://forum.osvita.ua/> (дата звернення 10.05.2025)
7. Програмне середовище PyCharm. URL:
<https://www.jetbrains.com/pycharm/> (дата звернення 23.05.2025)
8. Мова програмування Python. URL: <https://www.python.org/> (дата звернення 23.05.2025)
9. Бібліотека SpaCy. URL: <https://spacy.io/> (дата звернення 24.05.2025)
10. Бібліотека Rymorphy3. URL: <https://pypi.org/project/rymorphy3/> (дата звернення 24.05.2025)
11. Бібліотека discord.py. URL: <https://discordpy.readthedocs.io/en/stable/> (дата звернення 24.05.2025)

12. Бібліотека BeautifulSoup. URL: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/> (дата звернення 25.05.2025)
13. Фреймворк Streamlit. URL: <https://streamlit.io/> (дата звернення 27.05.2025)
14. Бібліотека FPDF. URL: <https://www.fpdf.org/> (дата звернення 28.05.2025)
15. Бібліотека XlsxWriter. URL: <https://xlsxwriter.readthedocs.io/> (дата звернення 28.05.2025)
16. База даних Cloud Firestore. URL: <https://firebase.google.com/docs/firestore> (дата звернення 28.05.2025)
17. Поліщук В.В., Свинчук О.В., Колумбет В.П., Бандурка О.І. Чат-боти в інформаційному супроводі абітурієнтів під час вступної кампанії. *Матеріали XXV Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів «Стан, досягнення та перспективи інформаційних систем і технологій»*. Одеса, 17-18 квітня 2025 р. С. 67-69.

ДОДАТОК А Лістинг розробленої системи

```
# Обробка повідомлень користувача
@bot.event
async def on_message(message):
    if message.author == bot.user:
        return
    user_id = message.author.id
    user_input = message.content.lower()
    key_words = nlp_process(message)
    normalized_words = normalize_words(key_words)
    # Початок нової розмови
    if user_id not in conversations:
        # Привітання
        if any(word in user_input for word in greeting):
            response = f"Привіт👋 Чим я можу тобі сьогодні допомогти?"
            await send_message(message.channel, response)
        # Допомога
        elif any(word in helps for word in normalized_words):
            response = f"З чим саме тобі допомогти? Спеціальності, документи,
тести, підготовка?"
            await send_message(message.channel, response)
        # Дякую
        elif any(word in thanks for word in user_input) or "дякую" in user_input
or "спасибі" in user_input:
            response = f"Завжди радий допомогти 😊" # звертайся
            await send_message(message.channel, response)
        # Підготовка до тестів
        elif any(word in trainee for word in normalized_words):
            await send_message(message.channel, "Для перевірки себе можеш
спробувати пройти демоваріанти або тести"
                                " минулих років. Для тривалого
процесу підготовки до тесту раджу"
                                " скористатися корисними
інструментами - тренажерами 🖱",
                                view=PreparingView())
        # Гайд по вступу
        elif any(word in guide for word in normalized_words) or "гайд" in
user_input: # гайд по вступі
            response, images = ipze_info_admission("https://ipze.kpi.ua/admission-
based-on-certificates-zno-and-nmt/")
            if images:
                await send_images_from_urls(message.channel, images)
                await send_message(message.channel, remove_duplicate_text(response),
WebsiteView("https://ipze.kpi.ua/admission-based-
on-certificates-zno-and-nmt/"))
            # Правила прийому та зразки документів
            elif any(word in norm_docs for word in normalized_words) or "правила" in
user_input:
                response1, images = ipze_info_admission("https://ipze.kpi.ua/legal-
regulations-and-reference-materials"
                                                        "-for-undergraduate-
admission/")
                response2, images2 = ipze_info_admission("https://ipze.kpi.ua/sample-
documents-for-admission/")
                if images or images2:
                    await send_images_from_urls(message.channel,
images.extend(images2))
                response = remove_duplicate_text(response1)
```

```

        response += remove_duplicate_text(response2)
        await send_message(message.channel, response)
    # Вступ
    elif any(word in admission for word in normalized_words):
        # Якщо введено документи для вступу
        if "документ" in normalized_words:
            docs = get_docs_needed("https://pk.kpi.ua/zayava-docs/")
            await send_images_from_urls(message.channel, [docs[0]], "Ось
спісок необхідних документів:")
        else:
            response = "Тебе цікавить поняття вступу чи гайд?"
            await send_message(message.channel, response)
            conversations[user_id] = {"state": "awaiting_admission_selection"}
    # Контакти приймальної комісії
    elif any(word in contacts for word in normalized_words):
        response = f"☎ Телефони: (+380 44) 204-96-45, 204-96-46\n✉ E-mail:
pk.kpi@l111.kpi.ua"
        faculty_contacts = parse_faculty_contacts()
        for contact in faculty_contacts:
            response += f"\n - Факультет/інститут: {contact['faculty']},
телефон:" \
                f" {contact['phone']}, E-mail: {contact['email']}"
            await send_message(message.channel, response)
    # Графік роботи приймальної комісії
    elif any(word in schedule_pk for word in key_words):
        schedule = parse_schedule_table()
        formatted_schedule = format_schedule_for_discord(schedule)
        await message.channel.send(formatted_schedule)
    # Перелік необхідних документів для вступу
    elif any(word in list_documents for word in normalized_words):
        await message.channel.send("Ти хочеш дізнатися про документи для
вступу чи нормативні документи?")
        conversations[user_id] = {"state": "awaiting_type_of_docs_selection"}
    # Мотиваційний лист
    elif any(word in motivation_letter for word in key_words):
        docs = get_docs_needed("https://pk.kpi.ua/motivation-letter/")
        await send_images_from_urls(message.channel, docs, "Ось все що тобі
порібно знати про мотиваційний лист:")
    # Тести
    elif any(word in nmt for word in normalized_words) or "нмт" in user_input
or "зно" in user_input:
        if "програма" in normalized_words:
            intro, programs = get_nmt_programs_info()
            response = f"{intro}\n"
            response += "\n".join([f"- {program['title']}\n Посилання:
{program['url']}]" for program in programs])
            await message.channel.send(response)
        else:
            paragraphs, subjects = parse_nmt_page()
            # Якщо користувач вводить нмт та предмет, то одразу інформація про
цей предмет
            matched_subject = None
            for subject in subjects:
                if subject["name"].lower() in user_input:
                    matched_subject = subject
                    break
            if matched_subject:
                texts, images = parse_subject_test(matched_subject['link'])
                if images:
                    await send_images_from_urls(message.channel, images)
                if texts:

```

```

        await send_message(message.channel, '\n'.join(texts),
WebsiteView("https://testportal.gov.ua/"
"pidgotovka-do-nmt-2025/"))
        del conversations[user_id]
    else:
        if paragraphs:
            await send_message(message.channel, '\n'.join(paragraphs))
        if subjects:
            subjects_message = "\n".join([f"- {subject['name']}" for
subject in subjects])
            await message.channel.send(subjects_message)
            await message.channel.send("Який предмет обираєш для
детального ознайомлення? Введи назву предмета"
" у тому відмінку, що я надав вище
👉")
            conversations[user_id] = {"state":
"awaiting_subject_selection_nmt"}
            # Часті запитання
            elif any(word in often_asked for word in key_words):
                categories = "\n".join([f"- {cat}" for cat in
question_category_key.keys()])
                conversations[user_id] = {"state": "choosing_category_faq"}
                await message.channel.send(f"Обери одну з категорій частих
запитань:\n{categories}")
            # Гранти
            elif any(word in grants for word in normalized_words):
                image_urls = parse_grants()
                await send_images_from_urls(message.channel, image_urls, "Ось
інформація про гранти:")
            else:
                specialties = parse_specialties()
                sented = False
                # Перевіряємо, чи користувач одразу запитує про конкретну
спеціальність
                for specialty in specialties:
                    if specialty['full_name'].lower() in user_input or
specialty['short_name'].lower() in user_input:
                        try:
                            description, programs =
parse_specialty_page(specialty['more_info_link'])
                            if "програма" in normalized_words:
                                # Отримуємо дані з бази даних
                                firestore_programs = await
fetch_programs_from_firestore()

                                response = f"Перелік освітніх програм для
{specialty['full_name']}:\n"
                                for i, program in enumerate(programs):
                                    url =
firestore_programs.get(program['name'].lower())
                                    if url:
                                        response += f"{i}. {program['name']}. 🌐 [Сайт
кафедри] ({url})\n"
                                    else:
                                        response += f"{i}. {program['name']}\n"
                                await send_message(message.channel, response,
ProgramsView(programs))
                                sented = True
                            elif "інформація" in normalized_words or "опис" in
normalized_words:
                                response = description

```

```

        response += "\nНеобхідні предмети для вступу
(бюджет):\n"
        for item in specialty['requirements_budget']:
            response += f" - {item}\n"

        if specialty['requirements_contract']:
            # чи потрібно? хіба туди предмети не треба
            response += "\nНеобхідні предмети для вступу
(контракт):\n"
            for item in specialty['requirements_contract']:
                response += f" - {item}\n"

                response += f"\nДетальніше:
{specialty['more_info_link']}"
                await send_message(message.channel, response,
WebsiteView(specialty['more_info_link']))
                sent = True
            else:
                response = "Ви хочете дізнатися детальну інформацію чи
перелік освітніх програм?"
                await send_message(message.channel, response)
                sent = True
                conversations[user_id] = {"state":
"awaiting_specialty_detail", "specialty": specialty}
            except TypeError:
                await message.channel.send("На жаль посилання на цю
спеціальність не дійсне 😞")
                if not sent:
                    if any(word in available_programs for word in normalized_words):
                        response = f"На даний момент у НТТУ КПІ є {len(specialties)}
спеціальностей.\n"
                        response += "Ось повний перелік:"
                        for faculty in specialties:
                            response += f"\n - {faculty['short_name']}:
{faculty['full_name']}"
                            response += "\nКожна спеціальність має свій перелік освітніх
програм. Про яку з них хочеш" \
                                " дізнатись більше? Можеш обрати номер (наприклад
G7) або ввести назву повністю."
                            await send_message(message.channel, response)
                            conversations[user_id] = {"state":
"awaiting_specialty_selection"}
                        else:
                            await send_message(message.channel, "Не зрозумів твого запиту
😞, спробуй іншими словами запитати.")
                            await bot.process_commands(message)

# Власна функція для надсилання повідомлень
async def send_message(channel, message, view=None):
    max_length = 2000 # Limit Discord
    if len(message) <= max_length:
        await channel.send(message, view=view, suppress_embeds=True)
        return
    chunks = split_text(message)
    for i, chunk in enumerate(chunks):
        if i == len(chunks) - 1:
            await channel.send(chunk, view=view, suppress_embeds=True)
        else:
            await channel.send(chunk, suppress_embeds=True)

# Функція для надсилання зображень

```

```

async def send_images_from_urls(channel, urls, text=None):
    files = []
    async with aiohttp.ClientSession() as session:
        for url in urls:
            async with session.get(url) as resp:
                if resp.status == 200:
                    img_data = await resp.read()
                    filename = url.split("/")[-1]
                    files.append(discord.File(BytesIO(img_data),
filename=filename))
    if not files:
        await channel.send("Не вдалося завантажити зображення.")
        return
    if len(files) <= 10:
        await channel.send(content=text or "", files=files)
    else:
        for i in range(0, len(files), 10):
            chunk = files[i:i + 10]
            await channel.send(content=text if i == 0 and text else "",
files=chunk)

# Нормалізація слів - перетворення їх до базового стану
def normalize_words(words):
    normalized = []
    for word in words:
        doc = nlp(word)
        lemma = doc[0].lemma_
        lemma_corrected = morph.parse(lemma)[0].normal_form
        normalized.append(lemma_corrected)
    return normalized

# Аналізуємо текст за допомогою spaCy
def nlp_process(message):
    doc = nlp(message.content.lower())
    key_phrases = []
    temp_phrase = []
    for token in doc:
        if (token.pos_ == 'NOUN' or token.pos_ == 'ADJ') and token.dep_ !=
'punct':
            temp_phrase.append(token.text)
        else:
            if temp_phrase:
                key_phrases.append(" ".join(temp_phrase))
                temp_phrase = []
            if temp_phrase:
                key_phrases.append(" ".join(temp_phrase))
    keywords = [token.text.lower() for token in doc if token.pos_ in ['NOUN',
'VERB'] and not token.is_stop]
    all_keywords = keywords + key_phrases
    return all_keywords

# Запуск бота
bot.run(BOT_TOKEN)

```

ДОДАТОК Б Апробація

Поліщук В.В., Свинчук О.В., Колумбет В.П., Бандурка О.І. Чат-боти в інформаційному супроводі абітурієнтів під час вступної кампанії. *Матеріали XXV Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів «Стан, досягнення та перспективи інформаційних систем і технологій»*. Одеса, 17-18 квітня 2025 р. С. 67-69.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Одеський національний технологічний університет
ННІ Комп'ютерної інженерії, автоматизації робототехніки та
програмування ім. П.М.Платонова
Національний технічний університет України
«Київський політехнічний інститут»
Університет Інформатики і прикладних знань м. Лодзь, Польща

XXV Всеукраїнська науково-технічна конференція
молодих вчених, аспірантів та студентів

«СТАН, ДОСЯГНЕННЯ І ПЕРСПЕКТИВИ
ІНФОРМАЦІЙНИХ СИСТЕМ І
ТЕХНОЛОГІЙ»

Матеріали конференції



Одеса

17-18 квітня 2025 р.

ЧАТ-БОТИ В ІНФОРМАЦІЙНОМУ СУПРОВОДІ АБІТУРІЄНТІВ ПІД ЧАС ВСТУПНОЇ КАМПАНІЇ

ПОЛІЩУК В.В., СВИНЧУК О.В., КОЛУМБЕТ В.П., БАНДУРКА О.І.
(vickypolishchuk04@gmail.com)

Національний технічний університет України «Київський політехнічний інститут імені
Ігоря Сікорського» (Україна)

У роботі розглянуто актуальність застосування чат-ботів для інформаційного супроводу абітурієнтів під час процесу вступної кампанії в закладах вищої освіти. Проаналізовано основні переваги та обмеження чат-ботів, а також розглянуто напрями їх подальшого розвитку в контексті цифрової трансформації освітнього процесу.

Постановка проблеми. Щороку тисячі абітурієнтів стикаються з необхідністю вибору майбутньої професії та навчального закладу. Цей процес супроводжується великим обсягом

67

Матеріали конференції «Стан, досягнення і перспективи інформаційних систем і технологій»

інформації, яку потрібно проаналізувати: спеціальності, освітні програми, умови вступу, вимоги до документів та дати подачі заяв [1]. У сучасних умовах цифровізації інформація про вступ доступна переважно в електронному форматі, зокрема на офіційних вебсайтах університетів, у нормативних документах та в інформаційних системах, таких як ЄДЕБО. Це дозволяє скоротити витрати ресурсів закладів вищої освіти, що потрібні для управління процесом вступу [2]. Однак через великий потік даних та складність навігації багато абітурієнтів стикаються з труднощами. Вони змушені витрачати багато часу на пошук необхідних даних, переглядаючи офіційні сайти університетів або звертаючись до приймальних комісій, які в період вступної кампанії можуть бути перевантажені.

Перелік завдань. Метою дослідження є комплексний аналіз можливостей застосування чат-ботів як ефективного інструменту інформаційного супроводу абітурієнтів у процесі вступної кампанії, з акцентом на особливості їх інтеграції у вищу освіту, а також оцінка потенціалу цих технологій щодо підвищення доступності, оперативності та персоналізації надання освітньої інформації.

Основна частина. Сучасні технології відкривають нові можливості для автоматизації процесу інформаційного супроводу абітурієнтів. Одним із ефективних рішень є використання чат-ботів – програм, які імітують живе спілкування та можуть швидко надавати користувачам необхідні відповіді. Більшість людей зараз віддають перевагу чат-ботам, а не традиційним пошуковим системам, оскільки вони можуть дати те, чого не можуть пошукові системи, а саме прями відповіді в більш персоналізованій формі [3].

Основними перевагами чат-ботів є швидка обробка запитів, одночасна взаємодія з багатьма користувачами та цілодобова доступність [4]. Вони оперативно надають відповіді на типові запитання, що знижує навантаження на приймальні комісії, особливо в період вступної кампанії, коли традиційні канали комунікації перевантажені.

Чат-боти також дозволяють здійснювати персоналізовану взаємодію з користувачем. Абітурієнт може задати питання, що стосується конкретної спеціальності, освітньої програми або конкурсного балу, та отримати відповідь з урахуванням його інтересів та

Основна частина. Сучасні технології відкривають нові можливості для автоматизації процесу інформаційного супроводу абітурієнтів. Одним із ефективних рішень є використання чат-ботів – програм, які імітують живе спілкування та можуть швидко надавати користувачам необхідні відповіді. Більшість людей зараз віддають перевагу чат-ботам, а не традиційним пошуковим системам, оскільки вони можуть дати те, чого не можуть пошукові системи, а саме прямі відповіді в більш персоналізованій формі [3].

Основними перевагами чат-ботів є швидка обробка запитів, одночасна взаємодія з багатьма користувачами та цілодобова доступність [4]. Вони оперативно надають відповіді на типові запитання, що знижує навантаження на приймальні комісії, особливо в період вступної кампанії, коли традиційні канали комунікації перевантажені.

Чат-боти також дозволяють здійснювати персоналізовану взаємодію з користувачем. Абітурієнт може задати питання, що стосується конкретної спеціальності, освітньої програми або конкурсного балу, та отримати відповідь з урахуванням його інтересів та вибору. Наприклад, чат-бот для розрахунку конкурсного балу може приймати від користувача індивідуальні значення: результати тестування з обраних предметів, заклад освіти, до якого він планує вступати, та бажану спеціальність. На основі цих даних, чат-бот зможе обчислити конкурсний бал абітурієнта та його ймовірність вступу на бюджет. Такі функції значно підвищують практичну цінність чат-ботів для користувачів та сприяють загальній цифровізації процесу вступу.

Особливо актуальним є інтеграція чат-ботів у популярні месенджери, такими як WhatsApp чи Discord, з якими більшість вступників знайомі та активно використовують для особистого спілкування. Це визначає популярність даного підходу, що робить чат-бот зручним і швидким у використанні, а також підвищує рівень комфорту й довіри користувачів до таких інструментів та робить комунікацію більш природною.

Однак є і певні обмеження щодо використання чат-ботів. Вони не завжди здатні надати точну відповідь на специфічні запити, що вимагають складного аналізу або обробки нестандартної інформації. Також часто виникають проблеми з розумінням користувацького вводу, коли запит некоректно сформульований або містить граматичні помилки, що призводить до хибних відповідей або взагалі відсутності реакції на запит.

Іншим обмеженням чат-ботів є їх адаптація до нових ситуацій [5]. В сучасному світі інформація, що була корисною кілька днів тому, може бути неактуальною сьогодні. Тому

перед чат-ботами постає завдання постійно оновлювати свою базу даних та буди здатним реагувати на зміни в умовах чи процесах вступу. Це вимагає використання додаткових людських та технічних ресурсів для підтримки таких систем.

Важливою проблемою є обмеженість логіки у простіших реалізаціях: якщо чат-бот побудований лише на ключових словах і жорстких сценаріях, він швидко стає непридатним у випадку змін у структурі вступної кампанії, умовах подачі документів або термінології. Саме тому все частіше використовуються більш гнучкі рішення на основі методів машинного навчання (ML) та обробки природної мови (NLP), які дозволяють ботам краще розуміти запити й підтримувати складніші діалоги [6].

Ще одним аспектом є сприйняття чат-бота самими абітурієнтами. Попри популярність цифрових сервісів, деякі користувачі можуть не довіряти автоматизованим відповідям або сумніватися в їхній актуальності. Тому важливо, щоб чат-бот був не лише функціональним, а й надійним: містив посилання на офіційні джерела, використовував коректну термінологію та пропонував можливість зв'язку з живим консультантом у разі потреби.

Висновки. Дослідження підтверджує ефективність чат-ботів як інструменту інформаційного супроводу абітурієнтів. Вони підвищують доступність та оперативність отримання освітньої інформації, сприяють персоналізованій взаємодії та зменшують навантаження на приймальні комісії. Інтеграція в популярні цифрові платформи забезпечує зручність і поширеність їх використання серед молоді.

Для досягнення повної ефективності чат-боти мають регулярно оновлювати базу знань, адаптуватися до змін і використовувати сучасні технології обробки природної мови. Незважаючи на окремі обмеження, вони мають значний потенціал для подальшого розвитку у сфері вищої освіти.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Х. В. Зуб, П. І. Жежнич «Аналіз ефективності вступної кампанії закладів вищої освіти України та способів її підвищення шляхом впровадження інформаційних технологій», *Вісник ВПШ*, вип. 3, с. 52–59, 2022. <https://doi.org/10.31649/1997-9266-2022-162-3-52-59>
2. В. О. Макоедова «Інформатизація процесів вступної кампанії в закладах вищої освіти», *Технічні науки та технології*, № 1(31), с. 90-97, 2023. [https://doi.org/10.25140/2411-5363-2023-1\(31\)-90-97](https://doi.org/10.25140/2411-5363-2023-1(31)-90-97)
3. Популярність чат-ботів: веб-сайт. [Онлайн]. Доступно: <https://www.techradar.com/pro/the-beginning-of-the-end-more-of-us-are-moving-away-from-google-towards-tiktok-and-ai-chatbots-as-research-reveals-that-the-golden-era-of-search-engines-may-well-be-over>
4. В. Андрусак, Л. Гобил, Т. Ваврик «Оптимізація навчального процесу університету за допомогою чат-бота», *Інформаційні технології та суспільство*, №1 (12), с. 6-12, 2024. <https://doi.org/10.32689/maup.it.2024.1.1>
5. Я. В. Буй «Автоматизація обслуговування клієнтів: чат-боти та CRM-системи», *Ukrainian Journal of Computing Innovations*, № 1, 2024. <https://doi.org/10.5281/zenodo.14946527>
6. О. О. Nalyvaiko «Prospects of using neural networks in higher education of Ukraine», *ITLT*, vol. 97, no. 5, pp. 1–17, Oct. 2023. <https://doi.org/10.33407/itlt.v97i5.5322>



Навчально-науковий інститут атомної та теплової енергетики
Кафедра інженерії програмного забезпечення в енергетиці

Інтелектуальний чат-бот для інформаційного супроводу абітурієнтів під час вступної кампанії

Виконала: студентка IV курсу, групи ТВ-13,
Поліщук Вікторія Володимирівна

Керівник: старший викладач,
д.ф. Колумбет Вадим Петрович

Консультант: доцент, к.ф.-м.н.,
доц. Свинчук Ольга Василівна

Київ-2025

1/15



Актуальність теми

Актуальність теми визначається:

- складністю пошуку інформації;
- навантаженням на приймальні комісії;
- необхідністю автоматизації відповідей на типові запити;
- відсутність сучасних інструментів інформаційного супроводу вступників;
- популярністю месенджерів серед молоді.



2/15

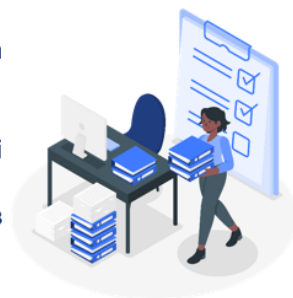


Постановка задачі

Мета: розробка інтелектуального чат-бота для інформаційного супроводу абітурієнтів під час вступної кампанії, який забезпечить оперативне й точне надання інформації та допоможе у вирішенні актуальних питань.

Завдання:

- проаналізувати літературні джерела, офіційні сайти освітніх установ та наявні рішення;
- визначити основні функціональні та нефункціональні вимоги;
- розробити архітектуру та реалізувати програмну систему у вигляді Discord-бота;
- реалізувати підтримку діалогової взаємодії та надання інформації з офіційних сайтів;
- виконати тестування та виправлення неполадок.



3/15



Аналіз предметної області

Критерій	Вебсайти	Інформаційні системи	Форуми
Актуальність даних	Офіційні дані	Офіційні дані	Не гарантована
Зручність пошуку	Складна навігація	Потребує багато параметрів	Довгий пошук
Інтерактивність	Обмежена	Статичні параметри	Живе спілкування
Швидкість отримання даних	Потрібен аналіз	Швидкий пошук	Очікування відповідей
Персоналізація	Відсутня	Відсутня	Обмежена

4/15



Вимоги до системи

Функціональні вимоги:

- надання довідкової інформації щодо процесу вступу;
- пошук та структуроване представлення інформації;
- надсилання зображень та посилань;
- розпізнавання природної мови та діалогова взаємодія;
- використання інтерактивних елементів (slash-команди, кнопки, випадні меню);
- повідомлення про нерозпізнаний запит;
- нагадування про важливі дати та події, пов'язані з вступною кампанією;
- робота з базою даних;
- адміністративне управління та валідація даних.

Нефункціональні вимоги:

- інтеграція з Discord;
- інтуїтивність інтерфейсу;
- цілодобова доступність та стабільність;
- масштабованість і можливість розширення.

5/15



Користувачі системи

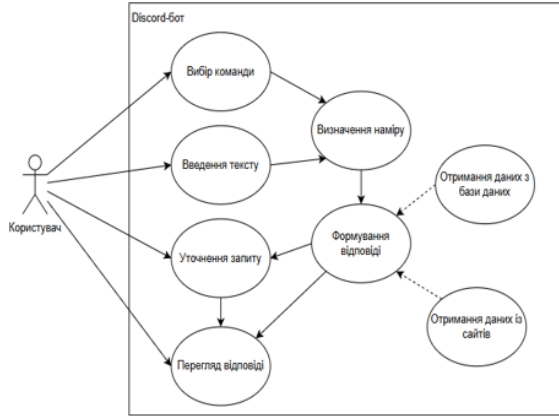


6/15

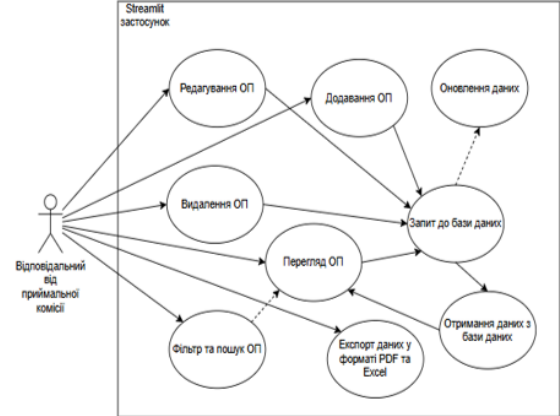


Діаграми варіантів використання

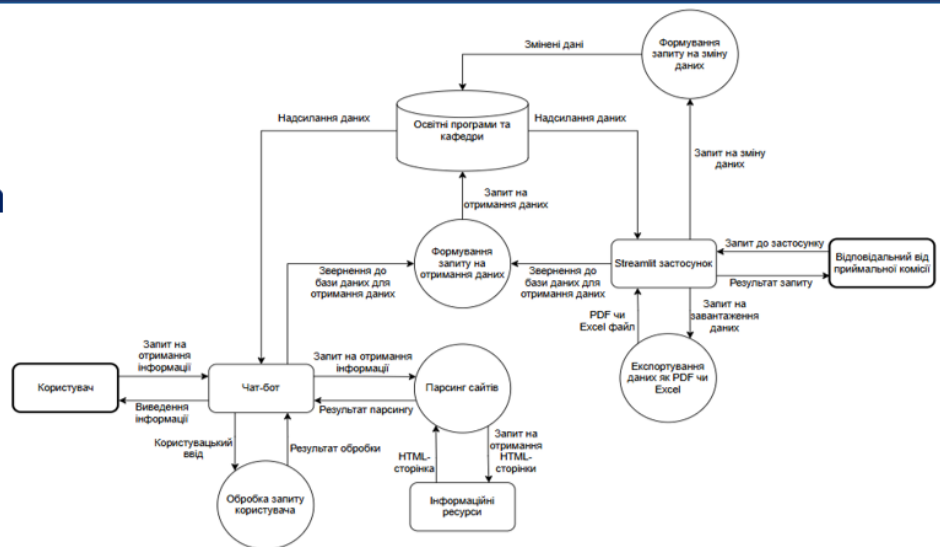
для користувачів Discord-бота



для користувачів вебзастосунку



Архітектура системи





Засоби розробки



PyCharm



Discord



Cloud Firestore



Python



SpaCy

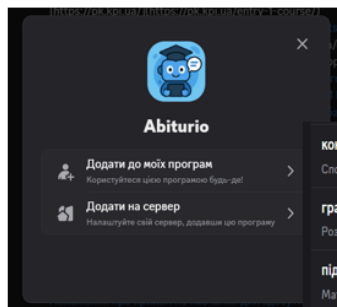


Streamlit

9/15



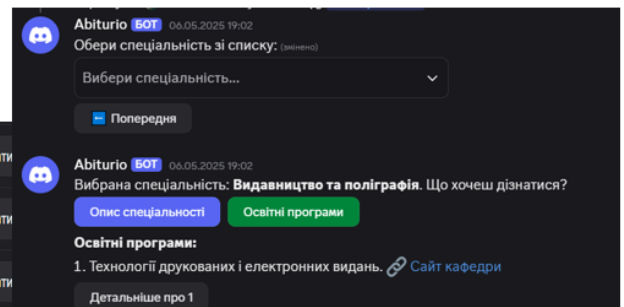
Взаємодія з ботом



Додавання бота до своїх програм

Перелік команд

контакти	Надіслати
Способи зв'язку з адміністрацією	
графік	Надіслати
Розклад роботи приймальної комісії	
підготовка	Надіслати
Матеріали для підготовки до вступу	
часті_питання	Надіслати >
Часті запитання абітурієнтів до приймальної комісії	
гранти	Надіслати >
Інформація про гранти	



Взаємодія інтерактивними елементами

10/15



Інформація від бота

Abiturio 20:57
Нагадування.
Дата: 30 травня
Подія: Очікування інформації.

Abiturio 06.05.2025 17:05
Що саме цікавить? Детальна інформація чи перелік освітніх програм?

whiskas 06.05.2025 17:05
опси

Abiturio 06.05.2025 17:05
Не зрозумів, уточни будь ласка.

Користувач whiskas застосував кожну з графік

Abiturio 02.05.2025 16:29
Адреса Приймальної комісії «КПІ ім. Ігори Сікорського» документи відірваної комісії вважувати повільно, були в основних станок у розкладі роботи кожної зміни.

Графік роботи секретаріату Приймальної комісії:

02 травня – 22 травня
Понеділок – П'ятниця: 14:00-17:00
Субота – Неділя: Вихідний

23 травня
П'ятниця: 14:00-18:00

26 травня – 30 червня
Понеділок – П'ятниця: 14:00-17:00
Субота – Неділя: Вихідний

01 – 18 липня
Понеділок – П'ятниця: 10:00-13:00
14:00-17:00
Субота – Неділя: Вихідний

19 липня
Середа: 10:00-13:00
14:00-18:00

19 липня – 02 серпня

Abiturio 20:48

КОНТАКТИ ВСТУПНИКА

Національний технічний університет України «Київський політехнічний інститут імені Ігори Сікорського»
Національний технічний університет України «Київський політехнічний інститут імені Ігори Сікорського»

Гайд для абітурієнта-2025: Як вступити без стресу
Підготуйте для вас головні лайфхаки з офіційного документу. Тримайте "шпатель"!

Топ-5 важливих дат (можливі зміни – слідкуйте за оновленнями)

- 3-10 липня – Реєстрація на творчі конкурси (для бюджету – до 10.07, для контракту – до 25.07).
- 19 липня – 1 серпня – Відео для подолання зава у СДЄ БО. Прорухайте час: останній день – 1 серпня до 18:00!
- 8 серпня – З'являється рейтинг сайту на Бюджет.
- 9 серпня (до 18:00) – Підтвердження пріоритетів для рекомендацій.
- 11 серпня – День X: офіційне зарахування на бюджет!
- Ваш електронний кабінет – на 90% готово!

Abiturio 20:07
Для отримання відповіді, введі запитання повністю або лише його номер:

1. Коли починається поселення у гуртожиток і як я про це дізнаюсь?
2. Якщо я не встиг подати оригінали документів, чи можна отримати гуртожиток для ночівлі?
3. В якому стані гуртожитки?
4. У гуртожитки світять першокурсники і першокурсники чи зі старшокурсниками?
5. Чи надається при вступі гуртожиток?

whiskas 20:07
3

Abiturio 20:07
В якому стані гуртожитки?
Гуртожитки знаходяться у абсолютно різному стані залежно від конкретного гуртожитку, який може мати блоню або коридорну систему. Багато студентів самі роблять ремонт, намагаються поліпшити умови, але жити точно можна! Навігація по гуртожитках в КПІ ім. Ігори Сікорського

11/15



Інформація від бота

Abiturio 21:11
Для перевірки себе можеш спробувати пройти демоверсії або тести минулих років. Для тривалого процесу підготовки до тесту раджу скористатися корисними інструментами – тренажерами

[Демоверсії](#) [Тести минулих років](#) [Корисні ресурси](#) [Перейти на офіційний сайт](#)

Орієнтуйтеся на демонстраційний варіант тесту. Потренуватися виконувати завдання різних форм, навчитися правильно позначати відповіді – у цьому помічно стане освітня платформа «Всеукраїнська школа онлайн», де доступні тренувальні варіанти предметних тестів. У вкладці «Курси» потрібно вибрати навчальний курс для 10 чи 11 класу. Тут розміщено демонстраційний варіант НМТ-2025, демонстраційний варіант НМТ-2024, перший і другий тренувальні варіанти НМТ-2024, перший і другий тренувальні варіанти НМТ-2023.

- **ВСЕУКРАЇНСЬКА ШКОЛА ОНЛАЙН**
Посилання: <https://lms.e-school.net.ua/>
- **ДЕМОНСТРАЦІЙНИЙ ВАРІАНТ НМТ-2025**
Посилання: https://lms.e-school.net.ua/courses/course-v1:uceqe+nmt_demo_2025+2025-01/about
- **НМТ-2024: ПЕРШИЙ ВАРІАНТ**
Посилання: https://lms.e-school.net.ua/courses/course-v1:uceqe+nmt_2024-V-4+2024-11/about
- **НМТ-2024: ДРУГИЙ ВАРІАНТ**
Посилання: https://lms.e-school.net.ua/courses/course-v1:uceqe+nmt_2024-V-6+2024-11/about
- **ДЕМОНСТРАЦІЙНИЙ ВАРІАНТ НМТ-2024**
Посилання: https://lms.e-school.net.ua/courses/course-v1:uceqe+demo_nmt_2024+2024_02/about
- **НМТ-2023: ПЕРШИЙ ВАРІАНТ**
Посилання: https://lms.e-school.net.ua/courses/course-v1:uceqe+nmt_2023-V-6+2024_02/about
- **НМТ-2023: ДРУГИЙ ВАРІАНТ**
Посилання: https://lms.e-school.net.ua/courses/course-v1:uceqe+nmt_2024_V-3+2024_02/about

12/15



Інтерфейс адміністративного вебзастосунок

Фільтрація та пошук

Пошук за назвою

Фільтр за інститутом / факультетом

Усі

Сортувати за назвою (А-Я)

Список освітніх програм

Експортувати як PDF

Експортувати як Excel

Київська інженерія

Технології захисту навколишнього середовища та гуманітарне розуміння

Технології друкування і електронні видання

Перегляд освітніх програм

Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці

[Перейти на сайт кафедри](#)

Інститут / факультет: ІАТЕ

Нова назва

Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці

Нове посилання

<https://ipze.kpi.ua/>

Новий інститут / факультет

ІАТЕ

Зберегти зміни

Видалити

Розширений перегляд освітньої програми

13/15



Висновки

В результаті роботи було розроблено інтелектуального чат-бота для інформаційного супроводу абітурієнтів під час вступної кампанії, який забезпечує оперативне й точне надання інформації та допомагає у вирішенні актуальних питань.

Виконано такі поставлені завдання:

- проаналізовано літературні джерела, офіційні сайти освітніх установ та наявні рішення;
- визначено основні функціональні та нефункціональні вимоги;
- розроблено архітектуру та реалізовано програмну систему у вигляді Discord-бота;
- реалізовано підтримку діалогової взаємодії та надання інформації з офіційних сайтів;
- виконано тестування та виправлення неполадок.

14/15



Апробація

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Одеський національний технологічний університет
ІНН Коми'ютерної інженерії, автоматизації, робототехніки та
програмування ім. П.М.Платонова
Національний технічний університет України
«Київський політехнічний інститут»
Університет Інформаційних І Гібридних знань м. Львів, Польща

XXV Всеукраїнська науково-технічна конференція
молодих вчених, аспірантів та студентів

«СТАН, ДОСЯГНЕННЯ І ПЕРСПЕКТИВИ
ІНФОРМАЦІЙНИХ СИСТЕМ І
ТЕХНОЛОГІЙ»

Матеріали конференції



Одеса

17-18 квітня 2025 р.

Поліщук В.В., Свинчук О.В., Колумбет В.П., Бандурка О.І. Чат-боти в інформаційному супроводі абітурієнтів під час вступної кампанії. Матеріали XXV Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів «Стан, досягнення та перспективи інформаційних систем і технологій». Одеса, 17-18 квітня 2025 р. С. 67-69.