

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»**

Спеціальність: 123 «Комп'ютерна інженерія»

Освітньо-професійна програма: «Комп'ютерна інженерія»

Група: 2БКС-29

КВАЛІФІКАЦІЙНА РОБОТА

**здобувача освіти денної форми навчання
БКС.29.02.000.КРБ**

***БОРИСКОВА
ДАНИЛА ЄВГЕНОВИЧА***

**м. Одеса
2025 р.**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 123 «Комп'ютерна інженерія»

Освітньо-професійна програма: «Комп'ютерна інженерія»

Група: 2БКС-29

ПОЯСНЮВАЛЬНА ЗАПИСКА


До кваліфікаційної роботи бакалавра на тему: «Створення інформаційного чат-боту для абітурієнтів коледжу»

Проектний матеріал складається з пояснювальної записки на 68 сторінках та графічного (презентаційного) матеріалу на 11 аркушах (слайдах)

Виконавець  (Борисков Д.С.)

Керівник проекту  (Іванова Л.В.)

Консультанти:

з розділу охорони праці та техніки безпеки  (Чорновол Н.І.)

з нормоконтролю  (Петрашова В.І.)

старший консультант  (Кривченко Ю.В.)

До захисту допущений

Завідувач кафедри  (Іванова Л.В.)

Завідувач відділенням  (Краснокутська К.Г.)

Захист «27» 06 2025 р.

Протокол ЕК № 2

Оцінка ЕК 5 (відмінно) / 90

Секретар ЕК 

АНОТАЦІЯ

Метою даної роботи є розробка інформаційного Telegram-бота для абітурієнтів ВСП «Одеський технічний фаховий коледж ОНТУ», який дозволяє швидко отримувати актуальну інформацію щодо вступу, сприяє автоматизації відповідей на типові запитання та покращує комунікацію між навчальним закладом та вступниками.

Вивчено закономірності використання месенджерів у молодіжному середовищі, зокрема Telegram як найбільш популярної платформи серед цільової аудиторії; досліджено існуючі рішення у сфері освітніх чат-ботів, виявлено їх функціональні можливості, переваги та недоліки.


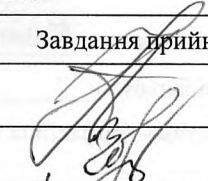
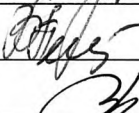


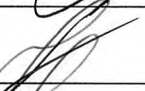
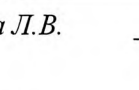
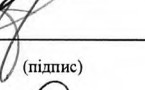
Отримані кількісні та якісні показники, що підтверджують доцільність створення Telegram-бота як зручного інструменту для абітурієнтів. Проведено технічний аналіз структури бота, реалізовано функціональні блоки для надання тематичної інформації, включаючи навчальні програми, терміни подачі документів, умови вступу та інше.

Створено повнофункціональний Telegram-бот з інтерактивним меню на основі кнопок, який забезпечує швидкий доступ до інформації без потреби в навігації сайтом. Реалізацію виконано з використанням мови програмування Python, бібліотеки aiogram, а також дотриманням сучасних підходів до структурування коду та зручності користувача.

Розглянуто питання з охорони праці та техніки безпеки.

Ключові слова: Telegram-бот, Python, aiogram, чат-бот, абітурієнт.

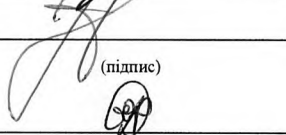
6. Консультанти по кваліфікаційній роботі, із зазначенням розділів, що їх стосуються

Розділ	Консультант	ПІДПИС	
		Завдання видав	Завдання прийняв
Основний розділ	Іванова Л.В.		
Розділ охорони праці	Чорновол Н.І.		
Нормоконтроль	Петрашова В.І.		
Старший консультант	Кривченко Ю.В.		

7. Дата видачі завдання _____


Керівник роботи Іванова Л.В.

Завдання прийняв до виконання _____


(підпис)

КАЛЕНДАРНИЙ ПЛАН

Пор. №	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1.	Аналіз технічного завдання та пошук джерел інформації	4.05.2025	
2.	Формування функціональних вимог до ПЗ	8.05.2025	
3.	Визначення технічного завдання на розробку	10.05.2025	
4.	Вибір технологій для реалізації чат-боту	12.05.2025	
5.	Розробка архітектури чат-боту	15.05.2025	
6.	Розробка алгоритмів роботи чат-боту	18.05.2025	
7.	Розробка елементів дизайну	22.05.2025	
8.	Розробка елементів інтерфейсу	24.05.2025	
9.	Створення макету чат-боту	26.05.2025	
10.	Створення прототипу	28.05.2025	
11.	Тестування ПЗ	31.05.2025	
12.	Тестування виконання вимог користувача	2.06.2025	
13.	Перевірка роботи на плагіат	6.06.2025	
14.	Підготовка до малого захисту	10.06.2025	
15.	Розробка питань з охорони праці	12.06.2025	
16.	Підготовка роботи до захисту та тестування ПЗ	15.06.2025	
17.	Оформлення слайдів мультимедійної презентації	18.06.2025	

Здобувач освіти  _____

(підпис)

Керівник роботи  _____

(підпис)

ЗМІСТ

Вступ	7
1 Основний розділ	8
1.1 Аналіз предметної області. Мета і обґрунтування проекту	8
1.1.1 Аналіз існуючих рішень	8
1.1.2 Технічне завдання	14
1.2 Обґрунтування вибору технологій і платформ для створення інформаційних чат-ботів.....	16
1.2.1 Визначення платформи, на якій буде реалізований чат-бот	17
1.2.2 Вибір основних технологій розробки	21
1.2.2.1 Вибір мови програмування	21
1.2.2.2 Вибір бібліотек для реалізації дипломного проекту	23
1.2.2.3 Вибір середовища розробки	24
1.2.2.4 Система контролю версій	24
1.3 Архітектура боту	25
1.3.1 Структура запитів та команд	25
1.3.2 Структура файлу для збереження повідомлень	27
1.3.3 Структура коренової папки, зберігання токена і глобальних змінних	27
1.4 Опис етапів розробки і функціональності проекту	29
1.4.1 Перший етап створення. Робота з BotFather	29
1.4.2 Встановлення усього необхідного ПЗ та залежностей	32
1.4.3 Створення структури проекту та робота над кодом	33
1.5 Тестування розробленого бота	39
1.5.1 Тестування дій користувача	39
1.5.2 Тестування дій адміністратора	45
2 Розділ охорони праці та техніки безпеки	49
2.1 Аналіз шкідливих та небезпечних факторів виробничого середовища	49
2.2 Вимоги до приміщення	50
2.3 Освітлення приміщення	50

					БКС 29.02.000.00 КРБ ПЗ	Арк.
						5
Ізм.	Лист	№ докум.	Підпис	Дата		

2.4 Вентиляція та мікроклімат	51
2.5 Пожежна безпека	52
Висновки	54
Перелік використаних інформаційних джерел	56
Додаток А. Фрагмент коду головного алгоритму мовою Python чат-боту для абітурієнтів коледжу	58
Додаток Б. Слайди мультимедійної презентації	63

					<i>БКС 29.02.000.00 КРБ ПЗ</i>	Арк.
						6
<i>Ізм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

ВСТУП

У сучасному цифровому суспільстві роль інформаційних технологій стрімко зростає, охоплюючи всі сфери діяльності людини, зокрема й освіти. З кожним роком збільшується кількість вступників, які прагнуть отримати актуальну та структуровану інформацію про умови вступу до закладів фахової передвищої освіти. У той же час, працівники приймальних комісій змушені відповідати на десятки типових запитів щодня, що створює додаткове навантаження і потребує автоматизації процесів.

Одним з ефективних рішень цієї проблеми є впровадження чат-ботів. Такі програмні рішення забезпечують оперативний доступ до потрібної інформації, працюють цілодобово та не потребують постійної присутності оператора. Вони є ефективними, економічно доцільними та доступними в реалізації навіть для невеликих навчальних закладів.

Особливої популярності в Україні набуває месенджер Telegram. Згідно з аналітичними даними на 2024–2025 роки, понад 80% української молоді віком від 14 до 25 років щоденно користуються цією платформою. Це робить Telegram ідеальним середовищем для розгортання освітніх ботів, орієнтованих на школярів, випускників коледжів та інших потенційних вступників.

Актуальність цієї теми зумовлена потребою підвищення ефективності інформаційного супроводу вступної кампанії, створення цифрових сервісів, зручних для молоді, а також необхідністю зменшення навантаження на адміністративний персонал коледжу. Реалізація Telegram-бота дозволяє уніфікувати відповіді на поширені запити, зробити доступ до інформації простим та інтуїтивно зрозумілим.

Таким чином, даний дипломний проєкт має не лише прикладне значення для самого навчального закладу, але й демонструє можливості використання сучасних інформаційних технологій у сфері освіти, що є важливим напрямом цифрової трансформації суспільства.

					БКС 29.02.000.00 КРБ ПЗ	Арк.
						7
Ізм.	Лист	№ докум.	Підпис	Дата		

1 ОСНОВНИЙ РОЗДІЛ

1.1 Аналіз предметної області. Мета і обґрунтування проекту

Перед початком розробки будь-якого програмного продукту важливо ознайомитися з уже наявними реалізаціями у вибраній сфері. Це дозволяє не лише краще зрозуміти поточний рівень розвитку технологій, а й виявити недоліки, які можна уникнути у власному рішенні, а також запозичити ефективні ідеї чи підходи.

У межах цього підрозділу було проведено огляд актуальних Telegram-ботів, створених для потреб абітурієнтів в Україні. Основна мета аналізу — визначити, які функції вже реалізовані у подібних рішеннях, наскільки ефективно вони працюють, які технології використовуються, та чи можуть вони бути адаптовані або вдосконалені у рамках розробки власного чат-бота.

Також досліджуються підходи, які застосовуються різними освітніми закладами до інформування вступників, у тому числі з використанням Telegram-каналів, веб-сайтів і ботів. Звертається увага на переваги та недоліки кожного формату, а також на відповідність запитам молодіжної аудиторії, яка є основною цільовою групою проекту.

Такий аналіз дозволяє сформулювати чітке розуміння ринкової ситуації, актуальності тематики та унікальності обраного напрямку, а також надає основу для обґрунтованого проєктного рішення.

1.1.1 Аналіз існуючих рішень

У цьому підрозділі описується аналіз функціональності чат-ботів українських ресурсів та навчальних закладів для розуміння актуальності і унікальності обраної тематики.

На українському ринку вже є декілька готових рішень, що наочно демонструють успішне застосування Telegram-ботів для абітурієнтів, наприклад “ОсвітаВсім” (OsvitaVsim). Це офіційний бот проекту ЄДЕБО/Інфоресурс, що доступний у Telegram. Він реалізує концепцію «єдиного вікна» для вступника:

					БКС 29.02.001.00 КРБ ПЗ	Арк.
						8
Ізм.	Лист	№ докум.	Підпис	Дата		

проводить людину через вибір спеціальності, надає ключову інформацію (дати вступної кампанії, відеовізуалізації процесу, поради, нормативні документи тощо). Розробники відзначають, що бот «не нав'язує спеціальність» і лише допомагає сформувати переліки можливих напрямків з урахуванням результатів ЗНО. Це демонструє існування робочого рішення, яке частково перекриває наші цілі і показує архітектуру «меню + інформаційний контент».

У боті є наступні функції:

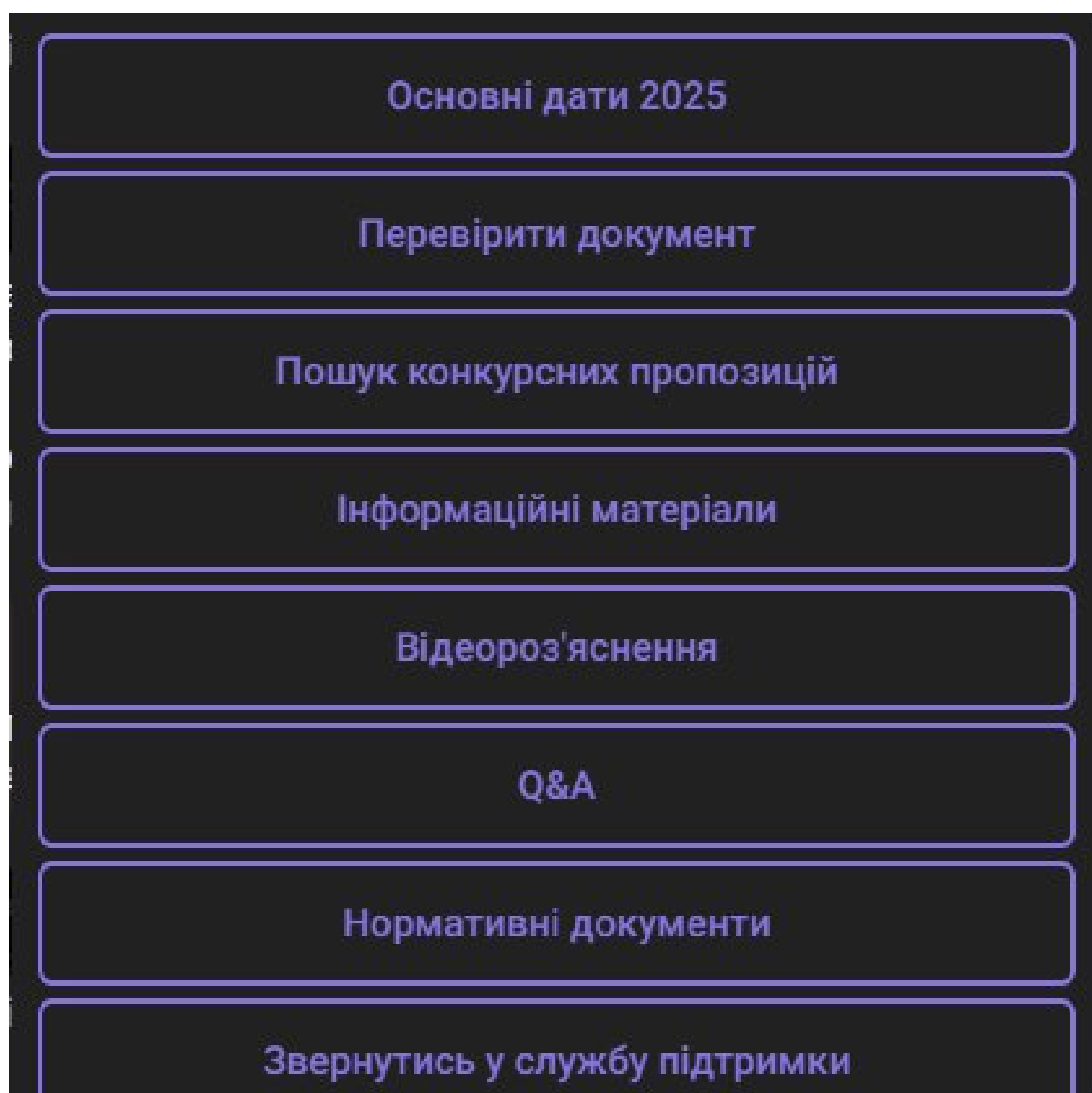


Рисунок 1.1. Меню бота “ОсвітаВсім”

В цілому бот працює, виконує поставлені перед собою задачі, однак не представляє собою бота конкретного навчального закладу.

					БКС 29.02.001.00 КРБ ПЗ	Арк.
						9
Ізм.	Лист	№ докум.	Підпис	Дата		

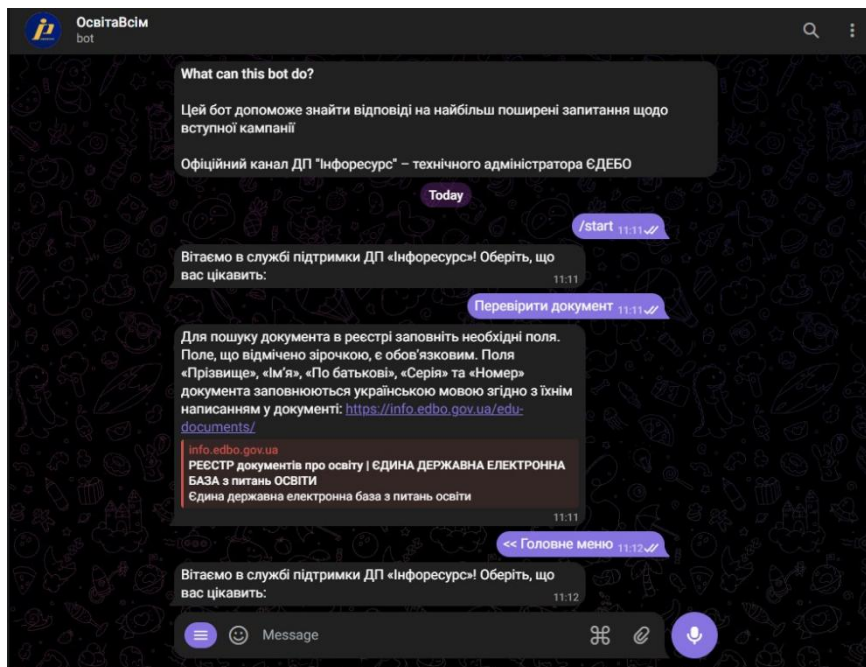


Рисунок 1.2. Демонстрація роботи бота “ОсвітаВсім”

“Вступ Helper”, ще один волонтерський бот, запущений у 2024 році, орієнтований на українських абітурієнтів. Він надає калькулятор конкурсу та розрахунок ймовірності вступу на бюджет до конкретних ВНЗ, а також відповідає на запитання про дати, пільги, пріоритети тощо. Мета проекту — знизити стрес вступників і спростити вступну кампанію. Приклад цього рішення може ілюструвати потребу в динамічних розрахунках всередині бота (математика конкурсів) та зручній подачі складних даних.

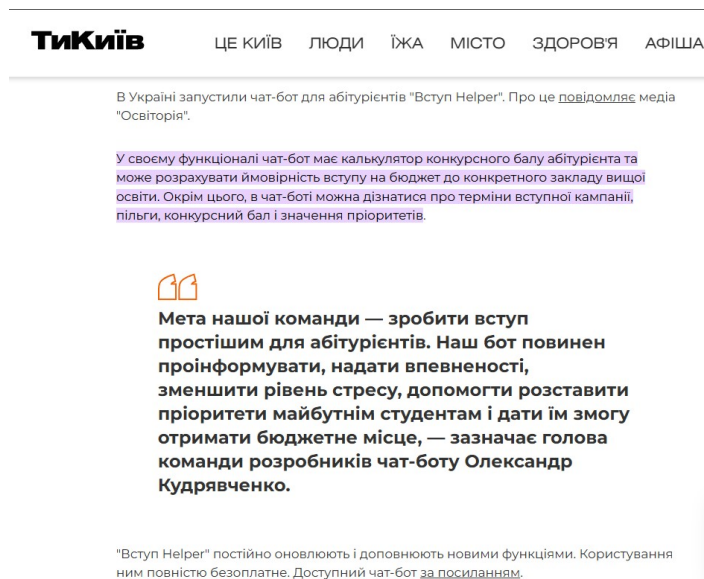


Рисунок 1.3. Стаття про “Вступ Helper”

					БКС 29.02.001.00 КРБ ПЗ	Арк.
						10
Ізм.	Лист	№ докум.	Підпис	Дата		

Заявлений функціонал дуже потужний, це не просто відповіді на питання, а взаємодії, якісь калькуляції і так далі, однак є нюанс - бот зараз не працює:

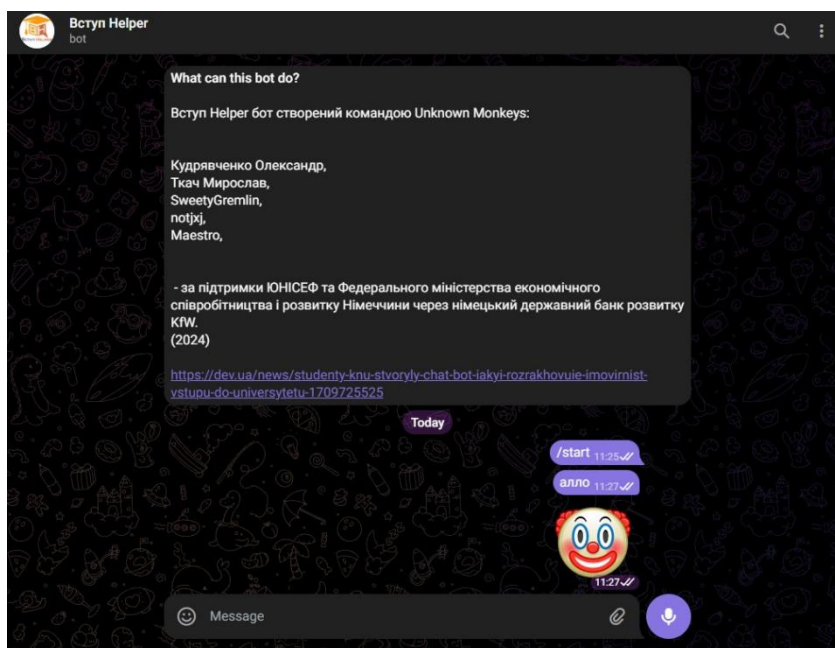


Рисунок 1.4. Демонстрація непрацездатності бота “Вступ Helper”

Перейдемо до більш прямих аналогів. Деякі навчальні заклади роблять телеграм-канали, а не боти, котрі повинні допомагати своїм абітурієнтам вступати, однак, недоліком телеграм каналів є незручність у пошуку інформації, коли у боті можна “запитати” у будь який час, не витрачаючи час на пошук.

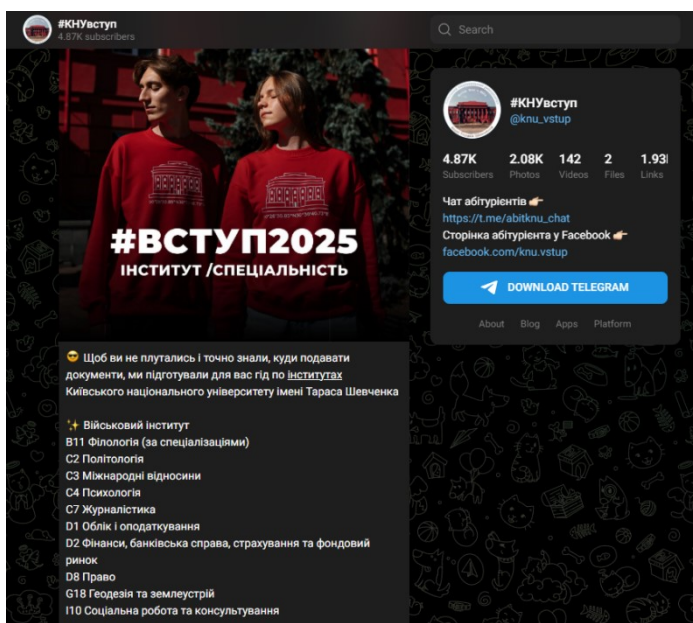


Рисунок 1.5. Телеграм канал вступу до КНУ

					БКС 29.02.001.00 КРБ ПЗ	Арк.
						11
Ізм.	Лист	№ докум.	Підпис	Дата		

Аналогом телеграм-канал можна не вважати, тому перейдемо до розглянення інших найдених мною рішень. Окремі ВНЗ розробляють власні боти для вступників. Наприклад, у Львівському університеті запустили Telegram-бот “IntroductoryLNU_bot”, який відповідає на найпоширеніші запитання абітурієнтів. У ході його дослідження виявилось, що він не працює:

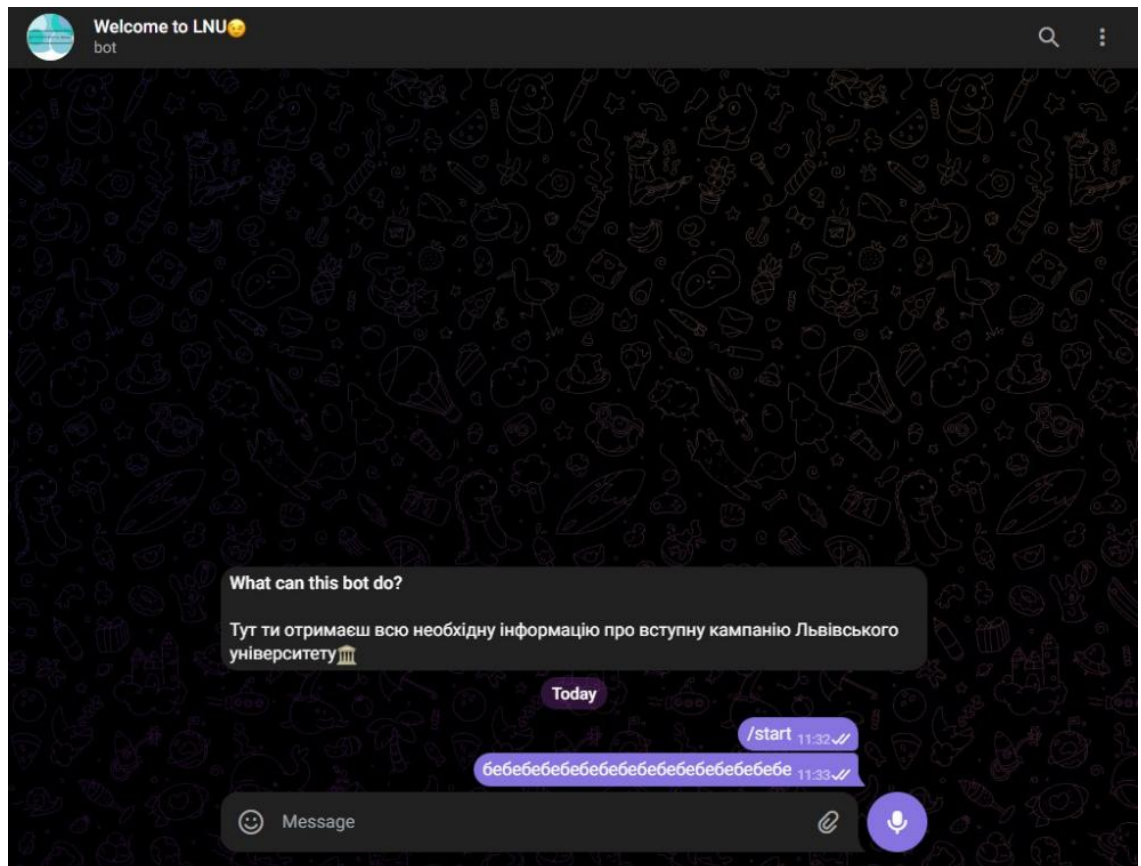


Рисунок 1.6. Телеграм бот вступу до ЛНУ

Однак, з сайту ЛНУ можна дізнатись, які функції передбачались у цьому боті:

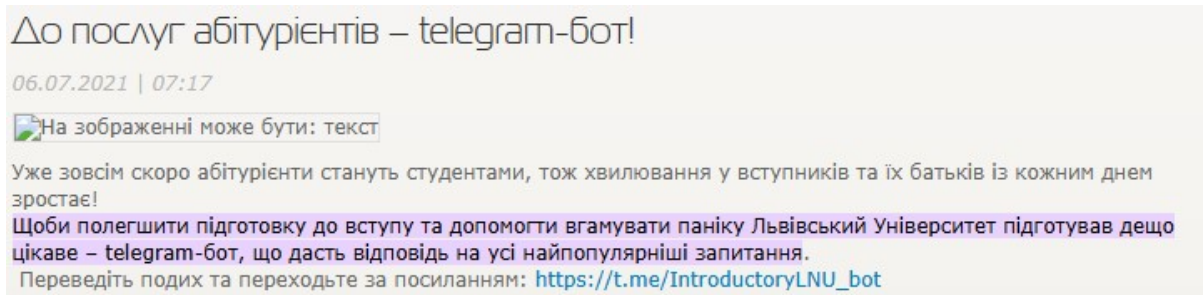


Рисунок 1.7. Стаття на сайті ЛНУ про запуск телеграм-боту для абітурієнтів

					БКС 29.02.001.00 КРБ ПЗ	Арк.
						12
Ізм.	Лист	№ докум.	Підпис	Дата		

З повідомлення можна зрозуміти, що бот повинен був відповідати на найпоширеніші запитання. Саме таку модель ми і примінемо для реалізації чат-боту для абітурієнтів ОТФК ОНТУ, доступна інформація у ПРАЦЕЗДАТНОМУ боті зробить коледж більш привабливим для вступу серед молоді.

На додачу до ботів, альтернативними каналами є веб-портали (зазвичай громіздкі для молоді) і офіційні канали у соцмережах або Viber. Однак Telegram переважає для нашого демографічного профілю, тому інші платформи менш актуальні.

Існуючі рішення показують, що необхідні функції можна реалізувати на базі Telegram-бота. Є успішні приклади, що Telegram-боти для вступу працюють і популярні серед освітніх інституцій (OsvitaVsim, Vstup Helper, університетські боти). Ці сервіси вже охоплюють більшість функцій, означених у нашому технічному завданні. Ми можемо вивчити їхній функціонал і архітектуру при розробці власного бота.

Проведений аналіз дозволяє виокремити кілька основних переваг та недоліків існуючих Telegram-ботів, орієнтованих на абітурієнтів. Серед переваг існуючих рішень:

- **Доступність:** Telegram-боти не потребують встановлення окремих додатків, що особливо зручно для молоді аудиторії.
- **Мобільність:** більшість функцій доступні 24/7 у смартфоні, що дозволяє отримати інформацію у будь-який момент.
- **Інтерактивність:** деякі боти, зокрема “Вступ Helper”, реалізують механізми калькуляції балів, формування рекомендацій та введення параметрів користувачем.
- **Офіційність:** наявність державного бота (“ОсвітаВсім”) демонструє довіру до такого формату з боку офіційних органів.

Недоліки виявлених рішень:

- **Низька надійність:** як показує приклад ботів ЛНУ та “Вступ Helper”, багато проєктів втрачають працездатність або не оновлюються після запуску.

					БКС 29.02.001.00 КРБ ПЗ	Арк.
						13
Ізм.	Лист	№ докум.	Підпис	Дата		

- Відсутність локалізації: більшість рішень є загальнодержавними або регіональними, і не враховують специфіку конкретного закладу.
- Складний інтерфейс або навігація: деякі боти не мають логічної побудови меню або надмірно перевантажені інформацією.

З огляду на проведений аналіз, можна зробити висновок, що створення Telegram-бота, орієнтованого виключно на потреби абітурієнтів одного навчального закладу, є обґрунтованим та актуальним. Такий бот дозволяє уникнути недоліків масових рішень, залишаючи при цьому всі ключові переваги формату: мобільність, швидкий доступ до інформації, зручність для молоді. Саме тому реалізація чат-бота для ОТФК ОНТУ, який містить актуальну, структуровану та офіційну інформацію, є цілком логічною відповіддю на виклики вступної кампанії сучасності.

Бот такого типу не тільки підвищує імідж навчального закладу як сучасного та технологічно підкованого, але й сприяє залученню абітурієнтів, для яких Telegram є основною платформою комунікації.

1.1.2 Технічне завдання

Метою є створення зручного, інформативного та доступного Telegram-бота, який забезпечить абітурієнтів актуальною інформацією про вступ до коледжу, зменшить навантаження на працівників приймальної комісії та сприятиме цифровізації освітнього процесу.

Цілі:

- Створити Telegram-бота, що працює в режимі 24/7.
- Забезпечити простий доступ до актуальної вступної інформації.
- Автоматизувати відповіді на типові запитання абітурієнтів.
- Зробити бота зручним для молодшої цільової аудиторії (14–19 років).

Завдання бота:

- Надання вичерпної інформації про умови вступу.

					БКС 29.02.001.00 КРБ ПЗ	Арк.
						14
<i>Ізм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

- Автоматизоване виведення актуальних дат та термінів.
- Пояснення процедур вступу (пільги, пріоритети, етапи).
- Формування переліку необхідних документів.
- Надання даних про спеціальності та конкурсні предмети.
- Відображення вартості навчання та обсягу державного замовлення.
- Інформування про підготовчі курси, іспити, розклади тощо.
- Реалізація інтуїтивної кнопкової навігації, що дозволяє швидко повернутися до головного меню з будь-якої точки діалогу.

Функціональні вимоги:

- Робота через Telegram Bot API.
- Обробка основних текстових команд: /start, /help, вибір кнопок.
- Структуровані кнопкові меню з можливістю навігації.
- Можливість змінювати вміст відповідей через окремий файл (наприклад, messages.py), без втручання в логіку.
- Надання посилань на офіційні документи.

Нефункціональні вимоги:

- Доступність 24/7.
- Швидкий час відповіді на запити користувача.
- Масштабованість: підтримка великої кількості одночасних користувачів.
- Безпека: використання захищених з'єднань Telegram, захист токена бота.
- Зручність використання: мінімалістичний, зрозумілий інтерфейс.
- Підтримка україномовного контенту та навігації.

Технології реалізації:

- Мова програмування: Python
- Фреймворк для Telegram-бота: Aiogram 3.x
- Середовище розробки: Visual Studio Code
- Система контролю версій: Git

					БКС 29.02.001.00 КРБ ПЗ	Арк.
						15
Ізм.	Лист	№ докум.	Підпис	Дата		

- Платформа розгортання: Telegram

Вимоги до програмного забезпечення:

- Python 3.10 або вище
- Встановлені бібліотеки: aiogram, dotenv, logging, інші залежності проєкту
- Операційна система: Windows / Linux
- Наявність Telegram-акаунту для тестування

Вимоги до апаратного забезпечення:

- ПК або ноутбук з мінімум 4 ГБ оперативної пам'яті
- Підключення до Інтернету
- Встановлений текстовий редактор (рекомендовано VS Code)
- Telegram-клієнт (мобільна або десктопна версія)

Вимоги до програмного забезпечення користувача:

- Встановлений застосунок Telegram Може бути як мобільна версія (Android/iOS), так і десктопна (Windows/Linux/macOS) або веб-версія. Рекомендовано використовувати актуальну версію клієнта для підтримки кнопочних меню, інлайн-інтерфейсу та мультимедійного контенту.
- Стабільне підключення до Інтернету. Для коректної роботи бота необхідний доступ до мережі — мобільного інтернету або Wi-Fi.
- Обліковий запис Telegram. Користувач повинен бути зареєстрованим у Telegram.
- Підтримка української мови в інтерфейсі Telegram (рекомендовано). Це полегшить сприйняття меню та відповідей бота.

1.2 Обґрунтування вибору технологій і платформ для створення інформаційних чат-ботів.

Після визначення мети, завдань та загальної ідеї дипломного проєкту наступним кроком є вибір технологічної основи для його реалізації. Правильний

					БКС 29.02.001.00 КРБ ПЗ	Арк.
						16
Ізм.	Лист	№ докум.	Підпис	Дата		

підбір платформи, мови програмування, бібліотек і середовища розробки є критично важливим для досягнення бажаного результату з урахуванням потреб користувачів, специфіки проєкту та ресурсів розробника.

Цей розділ присвячено обґрунтованому вибору платформи для запуску бота, а також аналізу і порівнянню різних технологій, які могли б бути використані під час реалізації проєкту. Зокрема, розглянуто переваги та недоліки найпопулярніших месенджерів, що підтримують інтеграцію з ботами, після чого обрано оптимальний варіант відповідно до цільової аудиторії — абітурієнтів українських навчальних закладів.

Далі аналізуються мови програмування, спеціалізовані бібліотеки для розробки Telegram-ботів, середовища розробки та інструменти для контролю версій. Такий підхід дозволяє створити надійну технічну основу, яка забезпечить зручну реалізацію функціоналу, масштабованість і подальший розвиток проєкту.

1.2.1 Визначення платформи, на якій буде реалізований чат-бот

У процесі розробки інформаційного чат-бота для абітурієнтів важливо було дослідити існуючі платформи, що підтримують інтеграцію ботів, та проаналізувати їх відповідність завданням, аудиторії й технічним можливостям. Особлива увага приділяється популярним месенджерам, які активно використовуються молоддю в Україні.

Telegram є одним із найпоширеніших месенджерів серед молоді в Україні. Платформа пропонує відкритий та зручний Bot API, який дозволяє створювати чат-боти будь-якої складності без потреби у складній реєстрації чи перевірці. Telegram-боти підтримують інтерактивні кнопки, меню, інлайн-команди, медіавміст, групові чати, та навіть логіку обробки callback-запитів. Окрім того, Telegram відзначається високою швидкістю роботи, безпекою, а також активною спільнотою розробників і великою кількістю супровідної документації.

Viber широко використовується в Україні, особливо серед старшої аудиторії та в державному секторі. Платформа також підтримує створення ботів через Viber Public Account API, що дозволяє реалізовувати інтерактивну комунікацію з

					БКС 29.02.001.00 КРБ ПЗ	Арк.
						17
Ізм.	Лист	№ докум.	Підпис	Дата		

користувачами. Проте функціональність Viber-ботів є дещо обмеженою у порівнянні з Telegram. Зокрема, реалізація складної логіки, багаторівневих меню або взаємодії з базами даних вимагає додаткових зусиль. Крім того, візуальна реалізація елементів інтерфейсу є менш гнучкою, а сам процес модерації бота займає більше часу.

Facebook Messenger має функціонал для створення ботів через Graph API, але процес налаштування є значно складнішим, оскільки вимагає створення окремої бізнес-сторінки, модерації та підтвердження профілю. Хоча бот може взаємодіяти з користувачем через шаблони повідомлень, quick replies і меню, платформа орієнтована більше на англomовний ринок. У контексті України популярність Facebook Messenger поступово зменшується, особливо серед молоді, яка дедалі частіше обирає альтернативні платформи, такі як Telegram або Discord. Таким чином, Facebook Messenger не є пріоритетною платформою для реалізації чат-бота, орієнтованого на абітурієнтів.

WhatsApp пропонує WhatsApp Business API для створення ботів, однак ця можливість доступна лише верифікованим компаніям, потребує складної процедури підключення та є частково платною. Крім того, функціональність ботів у WhatsApp є обмеженою, особливо у плані інтерфейсу та інтерактивних елементів.

Discord активно використовується в середовищі геймерів, молоді, програмістів та ІТ-спільноти. Його бот-платформа є надзвичайно потужною, а розробка може здійснюватися з використанням мов Python, JavaScript, Go тощо. Боти можуть відповідати на команди, автоматизувати процеси в чатах, модерацію та навіть обробку подій на голосових каналах. Однак Discord потребує наявності спеціального «сервера», де працюватиме бот. Це обмежує сферу використання й ускладнює доступ для звичайного користувача.

Slack — професійний месенджер, орієнтований на командну співпрацю в організаціях та бізнес-середовищі. Платформа має відкритий API і можливість створення інтерактивних ботів, але вона використовується здебільшого у внутрішньому корпоративному спілкуванні. Популярність серед молоді в Україні практично відсутня, тому Slack не розглядається як релевантна платформа у

					БКС 29.02.001.00 КРБ ПЗ	Арк.
						18
Ізм.	Лист	№ докум.	Підпис	Дата		

межах цього проекту.

Для того, щоб з залишившихся Telegram, Viber та WhatsApp вибрати, на якій платформі реалізувати бота, треба звернутися до статистики. Який з цих месенджерів популярніший у вікових категоріях абітурієнтів, а саме 14-15, та 18-19 років? Соціологічні опитування однозначно вказують: Telegram – лідер серед молоді. Так, дослідження USAID/CISR показало, що серед підлітків 13–15 років Telegram найпопулярніший – його щоденно чи кілька разів на день використовують близько 82% респондентів. У тій же групі наступний за популярністю TikTok (78%) та YouTube (63%):

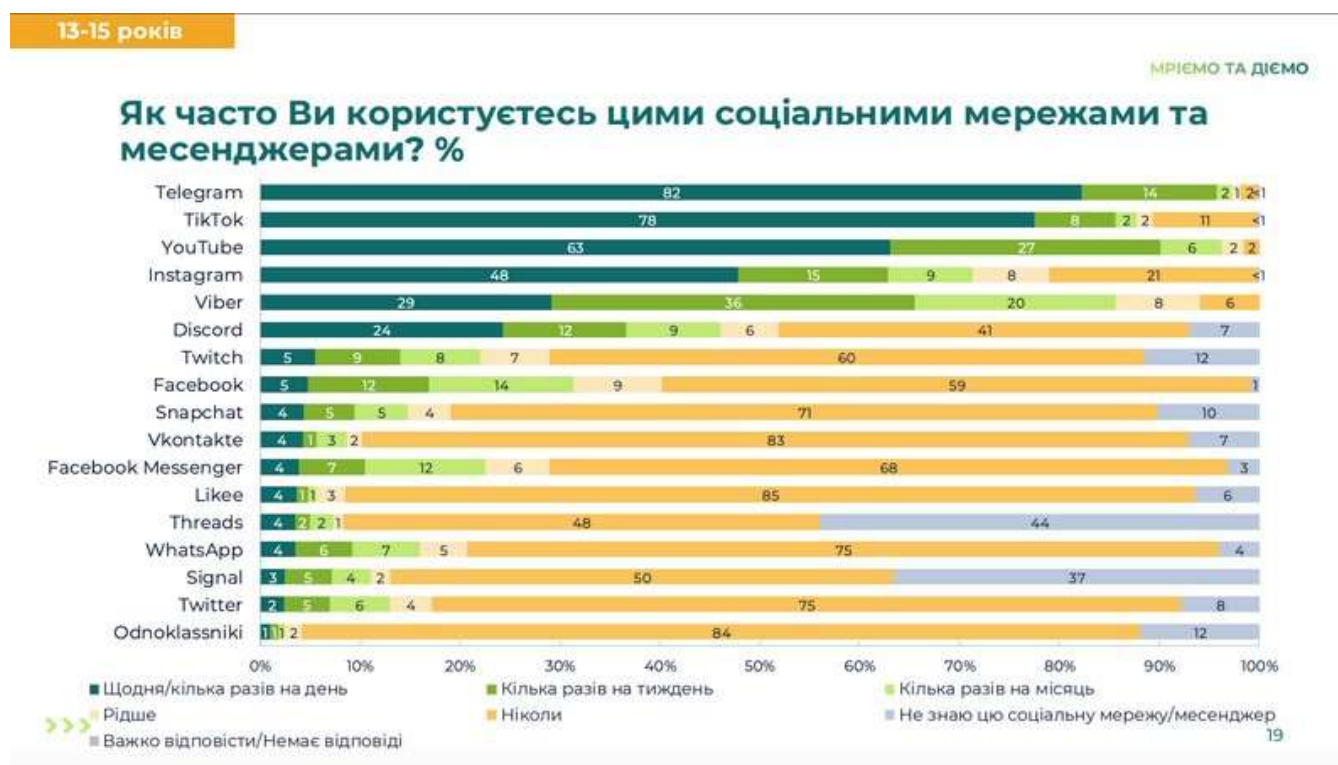


Рисунок 1.8. Результати опитування у категорії підлітків віком від 13 до 15 років

Схожа картина у старшій віковій групі: серед молоді 16–35 років 86% використовують Telegram щодня, тоді як YouTube – 70%, Instagram – 66%, Viber – 51%. Інші згадані месенджери використовують ще менше, а саме Discord –24%, WhatsApp – 4%, Facebook Messenger – 5%. Тобто понад три чверті молодих людей віддають перевагу Telegram над іншими месенджерами і соцмережами:

Як часто Ви користуєтесь цими соціальними мережами та месенджерами? %

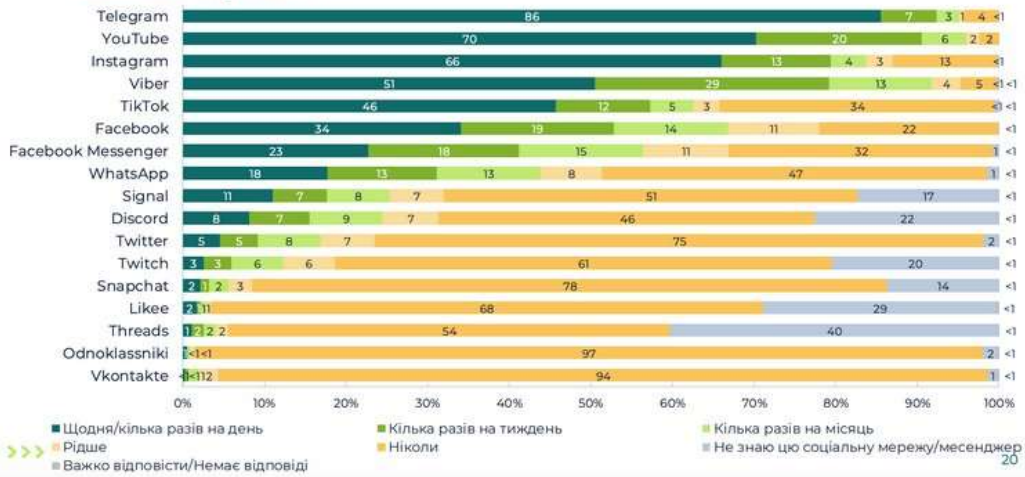


Рисунок 1.9. Результати опитування у категорії молоді віком від 16 до 35 років

Аналогічні результати бачимо й у контексті отримання новин: опитування громадян України показує, що Telegram утримує першість в медіаспоживанні. У 2024 році 78,1% опитаних взагалі зазначали Telegram серед платформ для новин (проти 59,5% YouTube та 44,6% Facebook). Серед молоді 18–39 років близько 90% читають новини саме через Telegram. Оскільки старша частина абітурієнтів (18–19 років) належить до цієї категорії, можна очікувати, що аналогічний рівень користування притаманний і їм.

СОЦІАЛЬНІ МЕРЕЖІ, ЯКІ ОПИТАНІ ВИКОРИСТОВУВАЛИ ДЛЯ ОТРИМАННЯ НОВИН ПРЯТОМ ОСТАННІХ ДВОХ МІСЯЦІВ

Можливі декілька варіантів відповіді

2022 рік 2023 рік 2024 рік

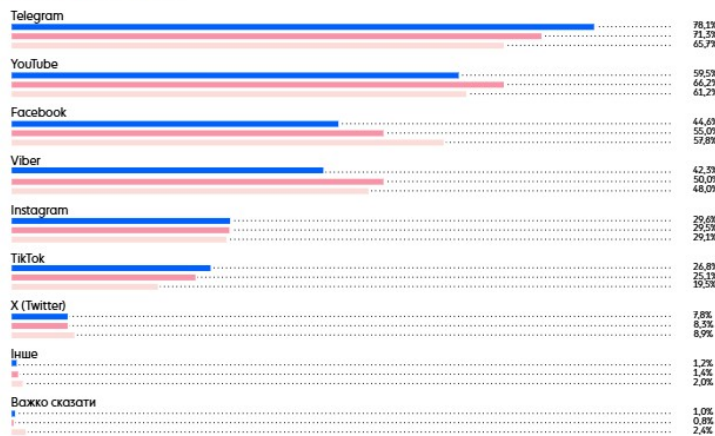


Рисунок 1.10. Результати опитування людей в іншому дослідженні

Ізм.	Лист	№ докум.	Підпис	Дата

Причини популярності Telegram у молоді (зокрема 14–19 років) пов’язані з особливостями платформи: безпека та анонімність, підтримка ботів та каналів з офіційною інформацією. Також Telegram дозволяє створювати тематичні групи й канали для вступників, а бот-інтерфейс є знайомим і зручним (доступний мобільний застосунок і десктоп-версія). У підсумку, цифри говорять самі за себе: телефони абітурієнтів вже “насичені” Telegram. Це робить Telegram ідеальною платформою для реалізації нашого проєкту.

1.2.2 Вибір основних технологій розробки

Після визначення платформи для реалізації чат-бота постає питання вибору технологічного стеку, що забезпечить ефективну розробку, підтримку та подальший розвиток системи. Правильно підібрана мова програмування, бібліотека для інтеграції з Telegram, середовище розробки та система контролю версій — це не лише питання зручності розробника, а й запорука стабільної роботи бота, швидкого реагування на запити користувачів і можливості розширення функціоналу в майбутньому.

У цьому підрозділі буде розглянуто ключові аспекти вибору технологій, які найкраще відповідають потребам проєкту. Також буде наведено обґрунтування доцільності використання Python та фреймворку Aiogram, а також супутніх інструментів, що сприяють ефективній реалізації Telegram-бота.

1.2.2.1 Вибір мови програмування

У сучасному середовищі розробки Telegram-ботів використовується низка мов програмування, серед яких найпопулярнішими є Python, JavaScript (зокрема Node.js), Go, Java та PHP. Кожна з цих мов має свої переваги та недоліки, і вибір конкретної залежить від завдань проєкту, рівня складності системи, а також досвіду розробника. Для розробки чат-бота в межах дипломного проєкту доцільно обрати мову, яка поєднує в собі простоту, гнучкість та активну підтримку з боку спільноти.

Python посідає особливе місце серед мов, що використовуються для

					<i>БКС 29.02.001.00 КРБ ПЗ</i>	Арк.
						21
<i>Ізм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

створення ботів, завдяки своїй читабельності, лаконічності синтаксису та доступності для початківців. Ця мова добре підходить для швидкої розробки прототипів і водночас дає змогу будувати складні, масштабовані системи. Однією з головних переваг Python у сфері Telegram-розробки є бібліотека Aiogram — асинхронний фреймворк нового покоління, що забезпечує зручну архітектуру, просту обробку подій, підтримку машини станів (FSM) та ефективну роботу з інтерфейсами бота. Aiogram дозволяє швидко реалізувати бот із багатоступеневою логікою, динамічними меню, inline-кнопками, а також інтеграцією з базами даних і зовнішніми API.

Іншою популярною платформою для створення Telegram-ботів є Node.js — середовище виконання JavaScript на сервері. Воно надає змогу писати асинхронний код та обробляти велику кількість одночасних запитів. Бібліотека Telegraf.js є основним інструментом для реалізації ботів у Node.js і підтримує широкий спектр функціональності. Проте для новачків ця технологія може бути складнішою, оскільки асинхронне програмування в JavaScript вимагає глибшого розуміння event loop та promises. Крім того, синтаксис JavaScript є менш інтуїтивно зрозумілим для студентів, які тільки починають знайомитися з бекенд-розробкою.

Go — це ще одна мова, яка забезпечує надзвичайно високу продуктивність і низьке споживання ресурсів. Її обирають для створення мікросервісів та навантажених систем. Telegram SDK для Go також існує, однак документація менш розвинена, а спільнота — менш численна. Розробка на Go вимагає глибших знань, суворішого контролю над помилками та не настільки гнучка для реалізації швидких MVP-рішень у порівнянні з Python.

PHP, хоча й залишається популярною у веб-розробці, дедалі рідше використовується для створення Telegram-ботів. Це пов'язано з її обмеженими можливостями у сфері асинхронного програмування та слабкою підтримкою сучасної архітектури. Хоча Telegram-боти на PHP можливі (наприклад, з бібліотекою MadelineProto), вони не забезпечують тієї гнучкості та зручності, що доступні у Python або Node.js.

З огляду на всі ці фактори, мова Python була обрана як основна для

					БКС 29.02.001.00 КРБ ПЗ	Арк.
						22
Ізм.	Лист	№ докум.	Підпис	Дата		

реалізації дипломного проекту. Вона забезпечує найкращий баланс між простотою, функціональністю та можливостями масштабування. У результаті Python виявляється не лише зручним і доступним для реалізації Telegram-бота, але й технічно виправданим вибором для навчального проекту, спрямованого на надання практичної інформації абітурієнтам освітнього закладу.

1.2.2.2 Вибір бібліотек для реалізації дипломного проекту

Серед доступних фреймворків для Telegram-ботів на Python ми зосереджуємось на Aiogram. Документація Aiogram прямо описує його як «сучасний і повністю асинхронний фреймворк для Telegram Bot API». Також плюсом можна виділити наявність документації українською мовою:

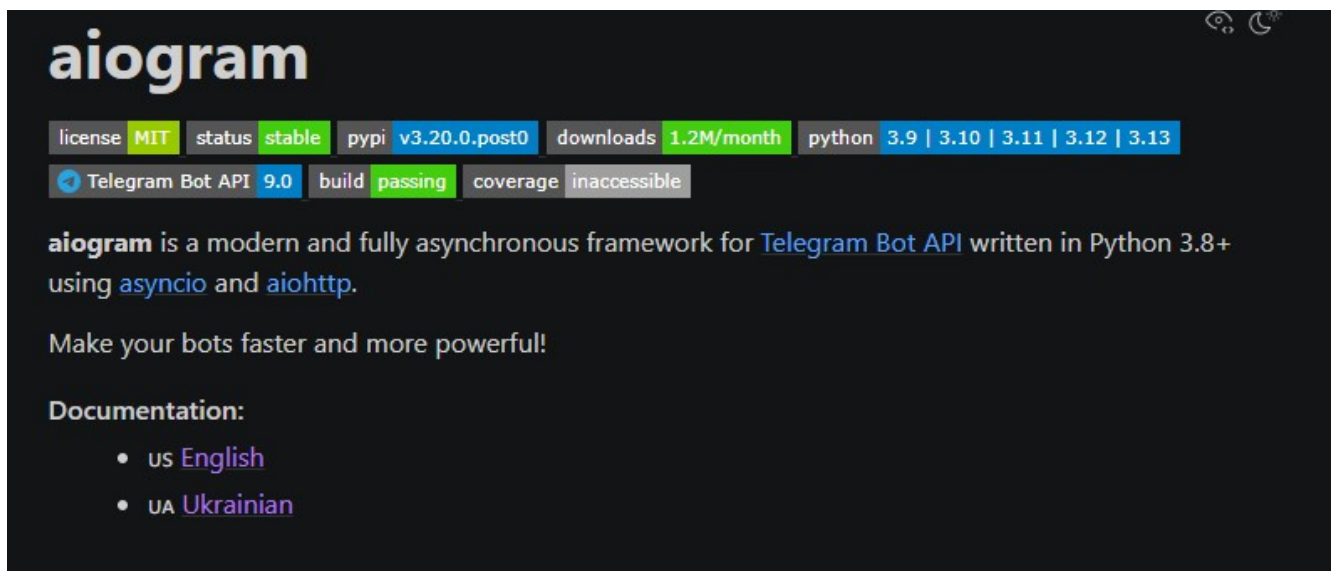


Рисунок 1.11. Демонстрація доступності документації англійською та українською мовами

Асинхронність (через asyncio) дозволяє боту паралельно обробляти багато запитів, що критично для великої аудиторії абітурієнтів. У статтях підкреслюється, що Aiogram 3.x – це потужний і гнучкий фреймворк, який дозволяє швидко реалізовувати складні сценарії взаємодії користувачів, зосереджуючись на логіці бота, а не на технічних деталях. Також Aiogram має вбудовані засоби роботи зі станами (Finite State Machine), клавіатурами, middleware тощо, що спрощує реалізацію діалогів.

					БКС 29.02.001.00 КРБ ПЗ	Арк.
						23
Ізм.	Лист	№ докум.	Підпис	Дата		

В альтернативу можна розглядати, скажімо, python-telegram-bot чи інші мови/фреймворки, але Aiogram вирізняється своєю сучасністю: він одразу спроектований під асинхронний код і активно підтримується. Як відзначає документація, «робіть свої боти швидшими і потужнішими» завдяки Aiogram. Python же, в свою чергу, дозволить легко інтегрувати додаткові модулі (наприклад, парсинг розкладу, NLP для аналізу запитів, AI-модулі для складніших відповідей), оскільки має багато відповідних бібліотек.

1.2.2.3 Вибір середовища розробки

Для розробки Telegram-бота в межах проекту було обрано середовище Visual Studio Code (VS Code) — один із найпопулярніших текстових редакторів серед розробників. Його головними перевагами є швидкодія, розширюваність та зручність у використанні. Завдяки великій кількості безкоштовних розширень (наприклад, для Python, Aiogram, роботи з Git, форматування коду тощо), VS Code забезпечує повноцінне середовище для написання, налагодження та тестування програмного забезпечення.

На відміну від важких IDE (як-от PyCharm або Visual Studio), VS Code споживає менше системних ресурсів і водночас пропонує гнучкі можливості під будь-який стек технологій. Інтерфейс редактора інтуїтивно зрозумілий, підтримує підсвічування синтаксису, автодоповнення, відладку коду та інтеграцію з терміналом, що робить його ідеальним вибором як для досвідчених розробників, так і для студентів.

1.2.2.4 Система контролю версій

Під час розробки програмного забезпечення важливо мати змогу відслідковувати зміни в коді, повертатися до попередніх версій і працювати з різними гілками проекту. Для цього використовується система контролю версій Git, яка стала галузевим стандартом.

Git дозволяє ефективно управляти історією розробки, створювати контрольні точки (коміти), вносити зміни без ризику втрати даних та, за потреби,

					БКС 29.02.001.00 КРБ ПЗ	Арк.
						24
Ізм.	Лист	№ докум.	Підпис	Дата		

об'єднувати зміни з кількох джерел. У поєднанні з платформами на зразок GitHub або GitLab він також забезпечує можливість командної роботи над кодом. У межах дипломного проєкту Git використовувався для збереження версій бота, створення резервних копій, а також для зручності під час тестування нових функцій.

1.3 Архітектура боту

Telegram-бот, створений у межах цього проєкту, реалізований за класичною архітектурною моделлю "клієнт–сервер" із використанням асинхронного фреймворку Aiogram, який дозволяє ефективно обробляти велику кількість запитів одночасно, не блокуючи виконання програми.

1.3.1 Структура запитів та команд

У боті треба буде реалізувати наступні команди/кнопки з відповідями на питання:

- documents - Документи, необхідні для вступу
- dates - Терміни прийому документів
- vacancies - Кількість вакантних місць
- subjects - Перелік конкурсних предметів
- cost - Вартість навчання
- state_order - Обсяг державного замовлення
- certificates - Сертифікати
- results - Результати вступних іспитів
- lists - Рейтингові списки
- cabinet - Інструкція для електронного кабінету
- online_docs - Подання документів онлайн
- specialties - Наші спеціальності
- programs - Освітньо-професійні програми
- courses - Підготовчі курси
- exams_programs - Програми вступних випробувань

					БКС 29.02.001.00 КРБ ПЗ	Арк.
						25
Ізм.	Лист	№ докум.	Підпис	Дата		

- schedule_9 - Розклад для вступників на базі 9 класу
- schedule_11 - Розклад для вступників на базі 11 класу

Команда після відправки буде проходити наступний шлях:

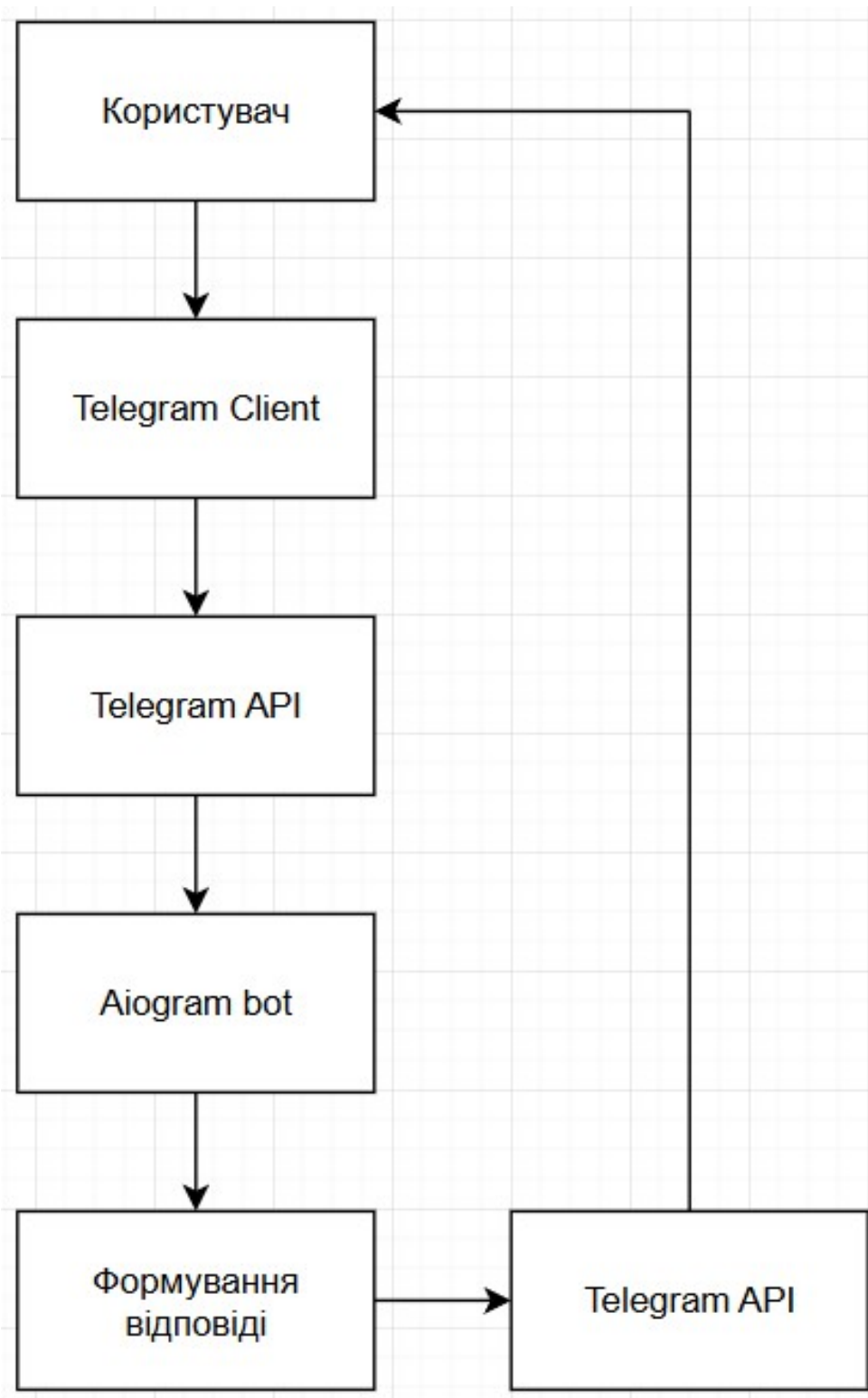


Рисунок 1.12. Візуальна схема шляху запиту

Ізм.	Лист	№ докум.	Підпис	Дата

1.3.2 Структура файлу для збереження повідомлень

У процесі розробки Telegram-бота було прийнято рішення винести текстові відповіді бота в окремий файл. Це рішення базується на принципах розділення логіки та контенту, що є гарною практикою при створенні масштабованих програм.

Такий файл виконує роль словника повідомлень — він містить лише текстові шаблони, які бот використовує для відповіді на запити користувачів. Наприклад, привітання, інструкції, підказки, відповіді на команди /documents, /subjects, /schedule_9 тощо.

Файл має формат .py і структурований у вигляді змінних, де кожна змінна відповідає певному повідомленню. Наприклад:

```
dates = "" <b>Терміни прийому документів</b>
```

ТЕКСТ ПОВІДОМЛЕННЯ

Докладніше: [https://otfk.od.ua/applicant/terms/""](https://otfk.od.ua/applicant/terms/)

1.3.3 Структура кореної папки, зберігання токена і глобальних змінних

Після створення, у проекту вийшла така структура:

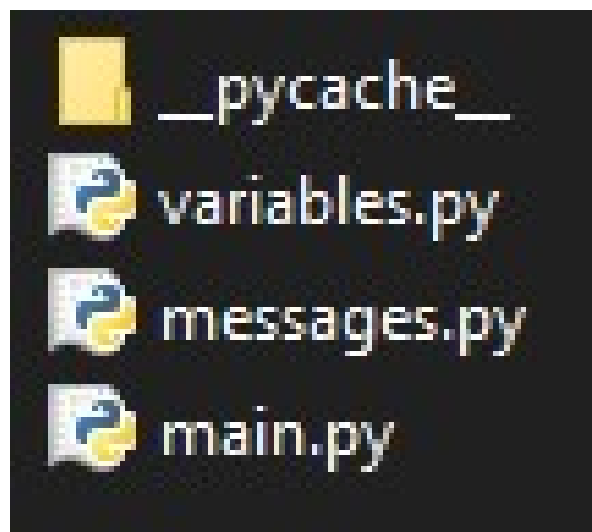


Рисунок 1.13. Структура кореної папки проекту

					БКС 29.02.001.00 КРБ ПЗ	Арк.
						27
Ізм.	Лист	№ докум.	Підпис	Дата		

У типовій структурі невеликого Telegram-бота, реалізованого з використанням мови програмування Python та бібліотеки Aiogram, доцільно виокремити кілька основних файлів, кожен з яких виконує конкретну функцію в архітектурі застосунку. Такий підхід відповідає принципам чистого коду, модульності та спрощує підтримку і масштабування проєкту. Розглянемо призначення ключових файлів, що входять до складу проєкту.

У процесі розробки програмного забезпечення мовою Python автоматично створюється спеціальна службова директорія з назвою `__pycache__`. Її основним призначенням є збереження зкомпільованих байт-кодів Python-модулів у вигляді файлів із розширенням `.pyc`. Це дозволяє прискорити запуск застосунку, зменшити навантаження на систему та підвищити загальну ефективність виконання програми.

Під час першого запуску або імпорту модуля Python-інтерпретатор компілює вихідний файл `.py` у проміжне представлення — байт-код. Отриманий результат зберігається у папці `__pycache__`. При наступному запуску програма використовує вже зкомпільовані `.pyc`-файли, що значно скорочує час завантаження модулів, особливо в проєктах із великою кількістю залежностей.

«`main.py`» це головний файл проєкту, з якого запускається бот. Саме тут створюється екземпляр бота, диспетчер повідомлень (Dispatcher), реєструються обробники команд, підключаються необхідні модулі, а також ініціюється цикл обробки оновлень (polling). Структурно цей файл відповідає за логіку запуску бота, взаємодію з Telegram API та маршрутизацію вхідних повідомлень.

«`messages.py`» файл призначений виключно для зберігання текстового контенту, який надсилається користувачеві під час взаємодії з ботом. Зазвичай сюди виносять вітальні повідомлення, інструкції, описи команд, шаблони відповідей тощо. Такий підхід дозволяє уникати дублювання коду, спрощує редагування текстів та дає змогу швидко адаптувати бота під різні мовні версії або оновлення змісту. Використання окремого модуля для текстів сприяє логічному розділенню коду на функціональні блоки: логіка — у `main.py`, контент — у `messages.py`.

					БКС 29.02.001.00 КРБ ПЗ	Арк.
						28
Ізм.	Лист	№ докум.	Підпис	Дата		

У модулі «variables.py» зберігаються глобальні змінні та конфігураційні параметри, зокрема:

- токен доступу до Telegram Bot API (TOKEN);
- налаштування проксі або логування (за потреби);
- шляхи до ресурсів (файлів, баз даних тощо);
- інші константи, які використовуються у проєкті.

Такий підхід дозволяє централізовано керувати змінними середовища без необхідності вносити зміни в основний код бота, що підвищує гнучкість та безпеку проєкту.

1.4 Опис етапів розробки і функціональності проєкту

У цьому розділі розглядається повна послідовність етапів, що були виконані під час створення Telegram-бота, призначеного для надання інформаційної підтримки абітурієнтам коледжу. Детально описуються дії, починаючи зі створення пустого бота та завершуючи тестуванням функціоналу готового бота в умовах використання. Також представлено опис функціональних можливостей програмного рішення, його логіка взаємодії з користувачем, перелік доступних команд і відповідних реакцій системи.

Основною метою даного етапу стало створення ефективного, простого у використанні та інформаційно насиченого інструменту для потенційних вступників.

У межах цього розділу буде детально проаналізовано ключові кроки розробки, включаючи реєстрацію бота через сервіс BotFather, побудову файлової структури проєкту, реалізацію функціональних блоків, а також проведення тестування працездатності застосунку.

1.4.1 Перший етап створення. Робота з BotFather

Процес створення Telegram-бота починається не з програмування, а з реєстрації бота у Telegram через офіційного сервісного бота під назвою BotFather. BotFather — це спеціальний бот, наданий Telegram, який дозволяє створювати

					БКС 29.02.001.00 КРБ ПЗ	Арк.
						29
Ізм.	Лист	№ докум.	Підпис	Дата		

нових ботів, налаштовувати їхній профіль, отримувати токен доступу та змінювати параметри. Основні кроки:

1. Користувач відкриває чат із BotFather за посиланням: <https://t.me/botfather>.
2. У діалозі вводиться команда /start.
3. Далі обирається команда /newbot.
4. Після цього BotFather просить вказати ім'я бота (відображається користувачам у профілі).
5. Потім потрібно ввести унікальний username, що має закінчуватися на bot.
6. Після успішного створення BotFather надає HTTP API Token — це унікальний ключ, який необхідний для програмного доступу до Telegram Bot API.
7. Цей токен є конфіденційним і зберігається в окремому конфігураційному файлі variables.py.

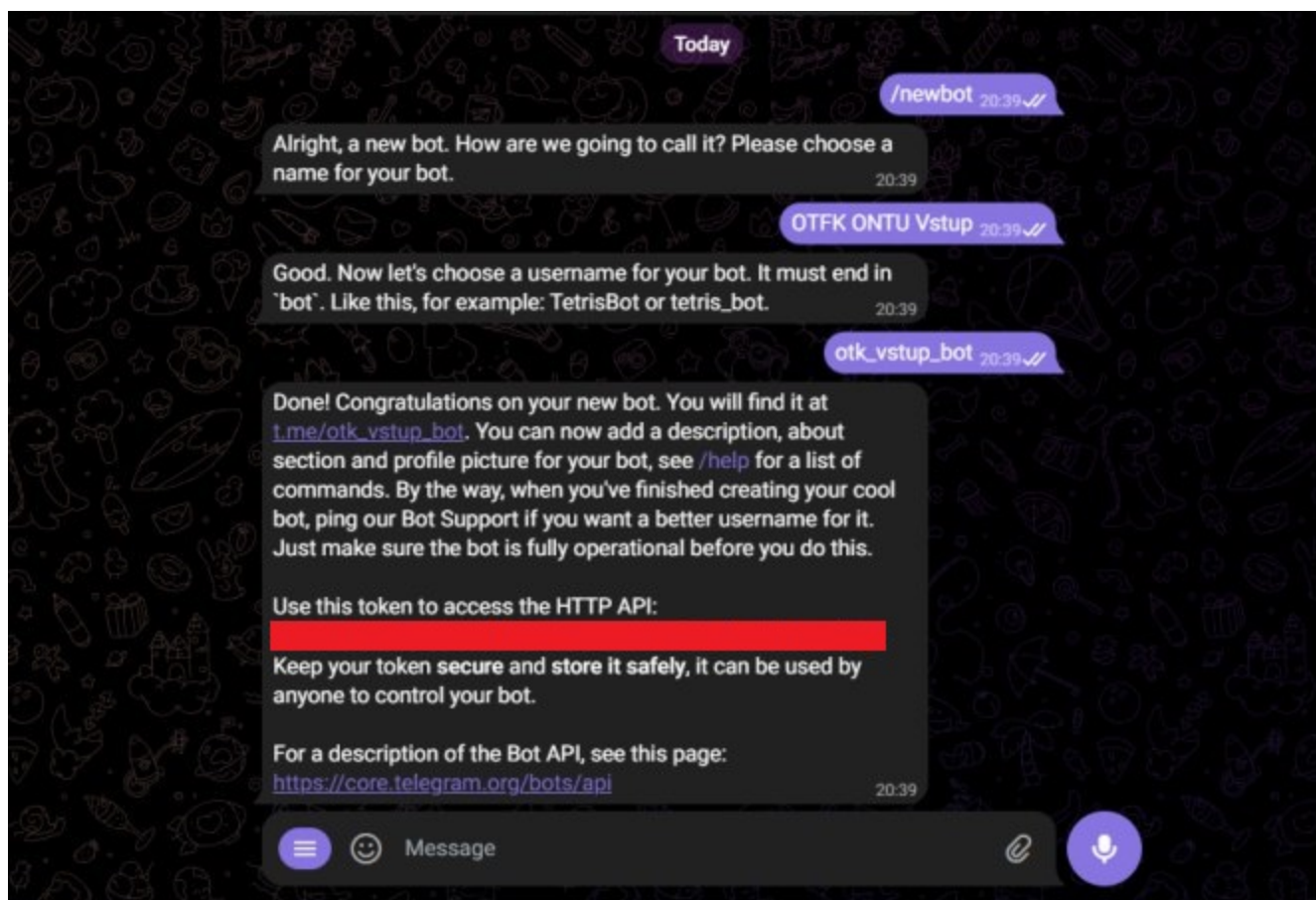


Рисунок 1.14. Процес створення бота

Після цього було прийнято рішення вибрати зображення для “аватару” бота, для цього я перейшов на офіційний сайт ОТФК, знайшов герб нашого коледжу та почав процес зміни, який виглядав так:

									Арк.
									30
Ізм.	Лист	№ докум.	Підпис	Дата					

БКС 29.02.001.00 КРБ ПЗ



Рисунок 1.15. Вибір боту для змін налаштувань

Я вибрав новостворений бот, потім зайшов в його налаштування та вибрав Edit Botpic:

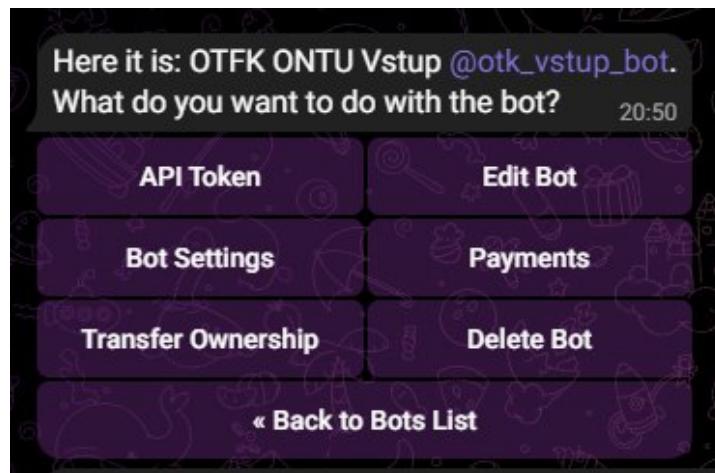


Рисунок 1.16. Дії, які можна зробити з ботом

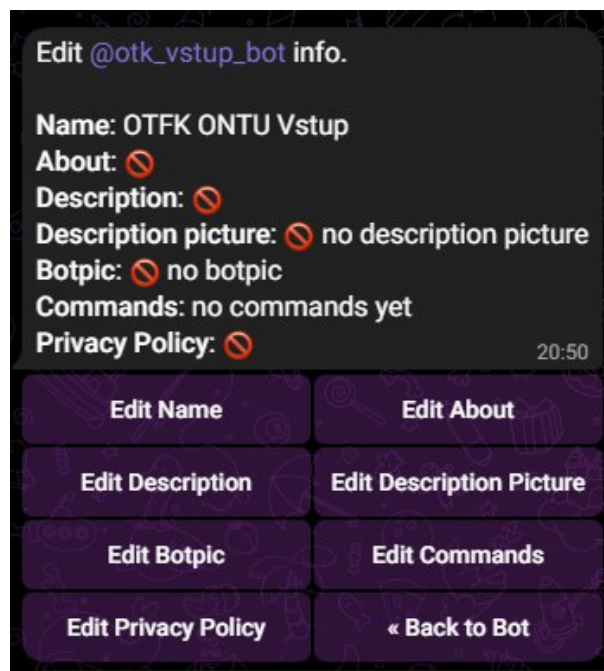


Рисунок 1.17. Налаштування боту

Далі залишилось тільки скинути фото і налаштування закінчено:

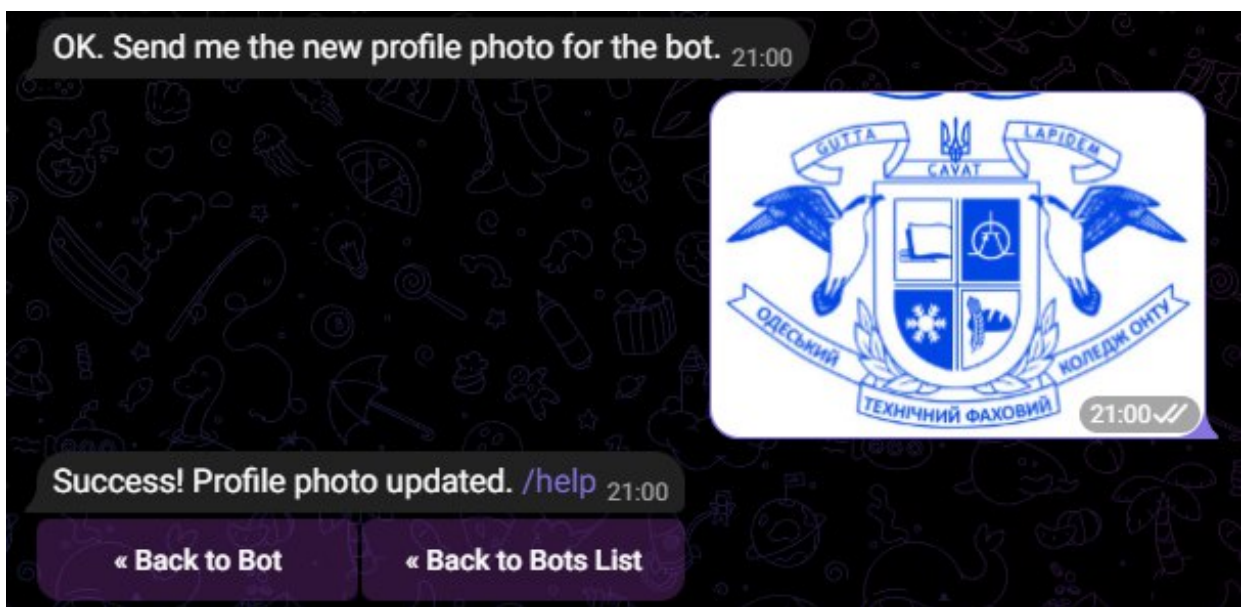


Рисунок 1.18. Завантаження фото боту

Тепер перевіримо, чи примінилися зміни:



Рисунок 1.19. Що видно користувачу у Bot Info

1.4.2 Встановлення усього необхідного ПЗ та залежностей

Для встановлення Python необхідно:

1. Перейти на офіційний сайт мови програмування Python: <https://www.python.org>.
2. Завантажити останню стабільну версію для операційної системи користувача (Windows, macOS або Linux).

3. Запустити інсталяційний файл. Обов'язково потрібно поставити позначку навпроти пункту “Add Python to PATH” — це дозволить запускати Python з командного рядка.

Далі встановлюємо Visual Studio Code:

1. Перейти на сайт: <https://code.visualstudio.com>.
2. Завантажити відповідний інсталятор для своєї операційної системи.
3. Запустити інсталяційний файл та обрати стандартні налаштування.

Після встановлення встановлюю наступні розширення (extensions) для роботи з Python:

1. Python — офіційне розширення від Microsoft для підтримки Python (автодоповнення, запуск скриптів, відлагодження).
2. Pylance — для покращеного аналізу коду.
3. GitLens — для інтеграції з Git та перегляду історії змін.

Далі встановлюємо необхідні бібліотеки, це робиться за допомогою командного рядка. Встановити треба `asuncio` та `aiogram`, це робиться командами “`pip install aiogram`” та “`pip install asuncio`”

1.4.3 Створення структури проекту та робота над кодом

На початковому етапі розробки чат-бота для абітурієнтів Одеського технічного фахового коледжу ОНТУ було створено базову структуру програмного проекту. Ця структура складалася з кількох окремих модулів, кожен з яких виконує певну функціональну роль. Такий підхід дозволяє реалізувати принцип розділення логіки, покращити зручність супроводу та масштабування коду. Структура файлів описувалась у пункті 1.3.3

Початкові повідомлення для користувачів були зібрані з офіційного веб-сайту коледжу — <https://otfk.od.ua>, або написані з нуля на основі інформації з PDF-сканів документів коледжу. Вони були структуровані та перенесені у файл

перебуває користувач, він завжди зможе легко орієнтуватися та за потреби почати «з нуля». Таким чином, реалізація логіки повернення до головного меню є критичним елементом загального користувацького досвіду.

Загальна логічна структура майбутнього меню буде являти собою ієрархічну систему, яка включатиме головне меню з доступом до основних розділів бота, таких як інформація про спеціальності, документи для вступу, терміни прийому, вартість навчання тощо. Кожен з цих розділів у свою чергу може містити підменю, що деталізують інформацію. Структура має бути гнучкою, розширюваною та побудованою таким чином, щоб забезпечити мінімальну кількість дій для отримання необхідних даних.

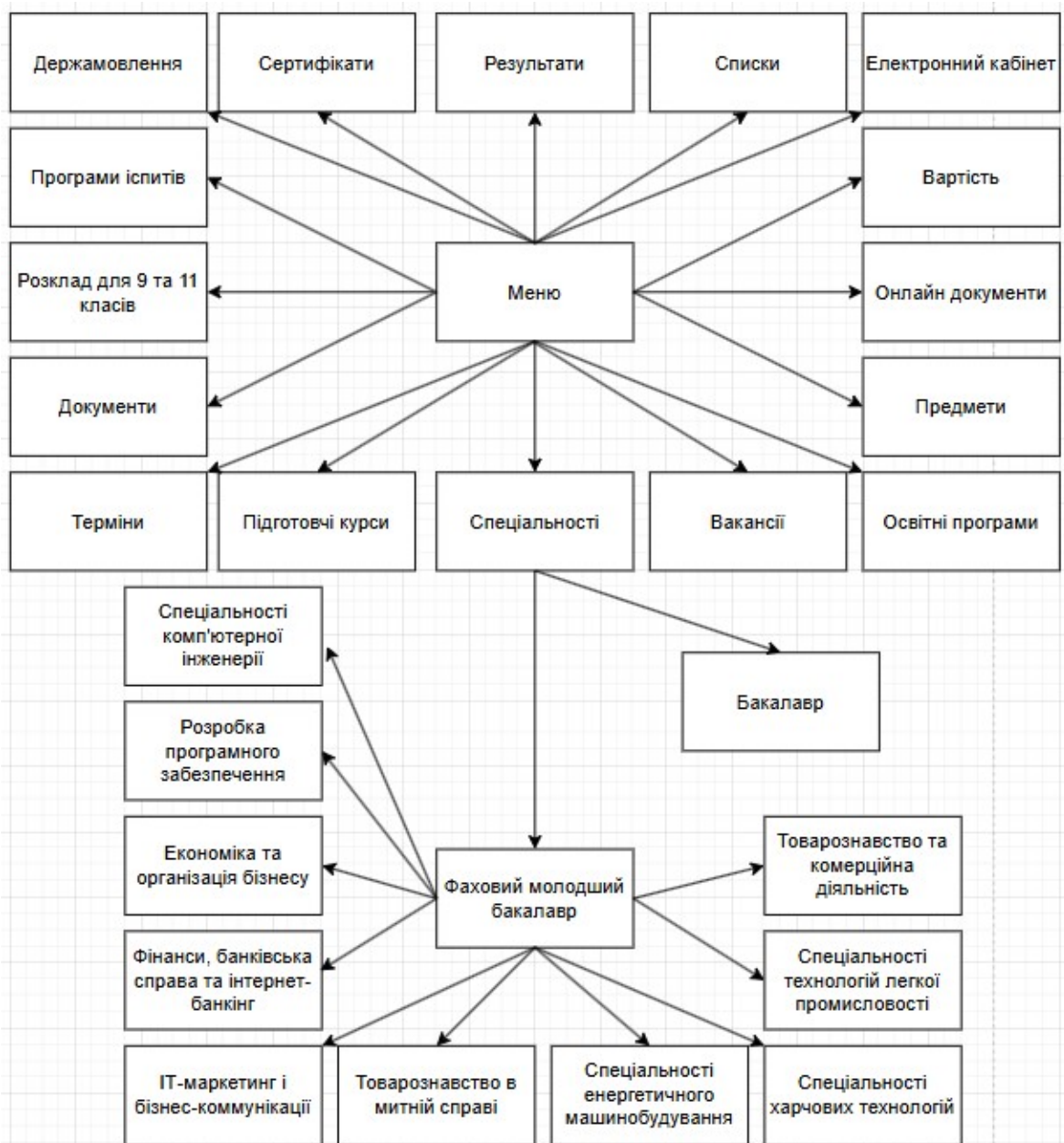


Рисунок 1.24. Структура головного меню

Ізм.	Лист	№ докум.	Підпис	Дата

текст повідомлення треба було брати з файлу messages.py, ось приклад реалізованих функцій:

```
67 # Обробка основних запитів
68 @dp.callback_query(F.data == "documents")
69 async def cb_documents(callback: types.CallbackQuery):
70     await callback.message.answer(messages.documents, parse_mode='HTML', reply_markup=back_button)
71     await callback.answer()
72
73 @dp.callback_query(F.data == "dates")
74 async def cb_dates(callback: types.CallbackQuery):
75     await callback.message.answer(messages.dates, parse_mode='HTML', reply_markup=back_button)
76     await callback.answer()
77
78 @dp.callback_query(F.data == "vacancies")
79 async def cb_vacancies(callback: types.CallbackQuery):
80     await callback.message.answer(messages.vacancies, parse_mode='HTML', reply_markup=back_button)
81     await callback.answer()
82
83 @dp.callback_query(F.data == "subjects")
84 async def cb_subjects(callback: types.CallbackQuery):
85     await callback.message.answer(messages.subjects, parse_mode='HTML', reply_markup=back_button)
86     await callback.answer()
87
88 @dp.callback_query(F.data == "cost")
89 async def cb_cost(callback: types.CallbackQuery):
90     await callback.message.answer(messages.cost, parse_mode='HTML', reply_markup=back_button)
91     await callback.answer()
92
93 @dp.callback_query(F.data == "state_order")
94 async def cb_state_order(callback: types.CallbackQuery):
95     await callback.message.answer(messages.state_order, parse_mode='HTML', reply_markup=back_button)
96     await callback.answer()
97
98 @dp.callback_query(F.data == "certificates")
99 async def cb_certificates(callback: types.CallbackQuery):
100     await callback.message.answer(messages.certificates, parse_mode='HTML', reply_markup=back_button)
101     await callback.answer()
102
```

Рисунок 1.27. Однотипні функції відправки повідомлень на команду

Завершальним етапом розробки базової структури Telegram-бота є реалізація запуску асинхронного циклу обробки повідомлень. Це реалізовано через основну асинхронну функцію main() та спеціальну конструкцію if `__name__ == "__main__"`:

```
132 async def main():
133     await dp.start_polling(bot)
```

Рисунок 1.28. Запуск процесу polling

У цьому фрагменті коду створюється асинхронна функція main(), яка запускає процес polling — постійне опитування серверів Telegram на предмет нових подій (повідомлень від користувачів). Метод dp.start_polling(bot) ініціює цикл обробки повідомлень, використовуючи об'єкт бота bot, створений раніше.

									Арк.
									38
Ізм.	Лист	№ докум.	Підпис	Дата					

Асинхронність дозволяє програмі не блокувати виконання, а працювати ефективно навіть при великій кількості одночасних запитів.

```
135 if __name__ == "__main__":
136     asyncio.run(main())
```

Рисунок 1.29. Перевірка запуску файлу

Цей блок коду перевіряє, чи виконується файл безпосередньо (тобто, як головний модуль програми), і в такому випадку запускає асинхронну функцію `main()` через метод `asyncio.run(...)`. Цей механізм є стандартним способом запуску асинхронного коду в Python 3.7+.

Використання `asyncio` є необхідним, оскільки бібліотека `aiogram` побудована на асинхронній моделі виконання, яка забезпечує високу продуктивність та швидку реакцію на дії користувача без блокування інших частин програми.

1.5 Тестування розробленого бота

Після завершення етапу розробки чат-бота було проведено тестування його функціоналу з метою перевірки стабільності роботи, відповідності логіки реакцій на команди та загального користувацького досвіду. Перевірка відбувалась у кілька етапів, починаючи з запуску бота через термінал розробника. Це дало можливість не лише переконатися у відсутності критичних помилок під час старту, але й одразу увімкнути режим логування для спостереження за подіями у реальному часі. У терміналі з'явився відповідний вивід, який свідчить про те, що бот активний і готовий до взаємодії з користувачем:

```
PS C:\Users\forcr> & C:/Users/forcr/AppData/Local/Programs/Python/Python311/python.exe d:/Projects/diplom/main.py
INFO:aiogram.dispatcher:Start polling
INFO:aiogram.dispatcher:Run polling for bot @otk_vstup_bot id=8006344610 - 'OTFK ONTU Vstup'
```

Рисунок 1.30. Старт боту та початок логування

1.5.1 Тестування дій користувача

Далі було відкрито Telegram-інтерфейс самого бота. Після переходу до чату, користувач повинен ініціалізувати взаємодію, натиснувши кнопку "START" або ввівши команду `/start`. Цей крок запускає основний сценарій, відповідальний за

					БКС 29.02.001.00 КРБ ПЗ	Арк.
						39
Ізм.	Лист	№ докум.	Підпис	Дата		

виведення початкового меню. У нашому випадку команда /start відпрацювала штатно — бот надіслав привітальне повідомлення з переліком можливих запитів, які він здатен обробити. Це означає, що початкова логіка функціонування реалізована вірно:

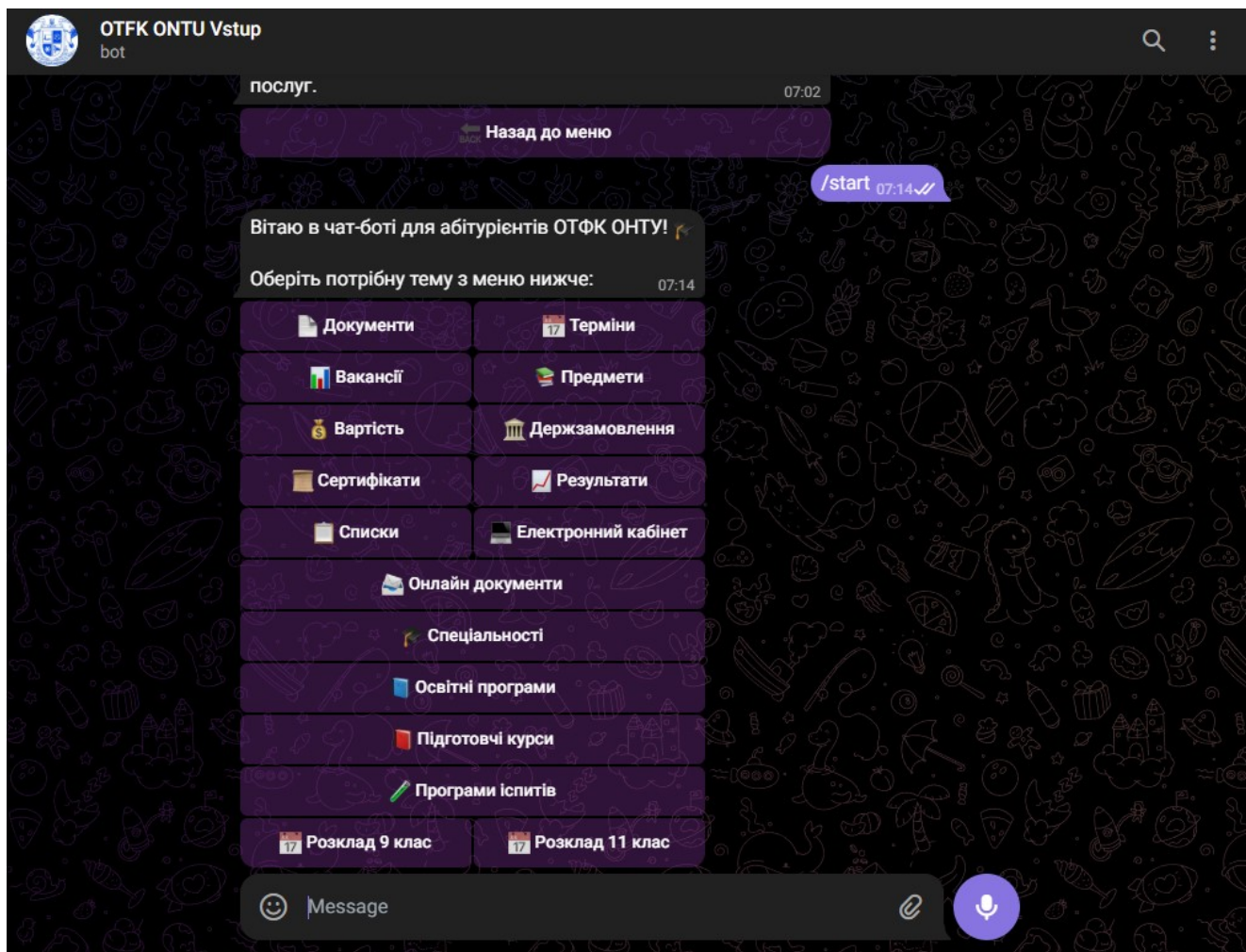


Рисунок 1.31. Реакція на команду “Старт”

На команду старт бот видав список того, що може підказати, а саме: документи, необхідні для вступу; терміни прийому документів; кількість вакантних місць; перелік конкурсних предметів; вартість навчання; обсяг державного замовлення; сертифікати; результати вступних іспитів; рейтингові списки; інструкція для електронного кабінету; подання документів онлайн; спеціальності коледжу; освітньо-професійні програми; підготовчі курси; програми вступних випробувань; розклад вступних іспитів для бази 9 класу; розклад вступних іспитів для бази 11 класу.

Припустимо, що ми хочемо дізнатися перелік необхідних для вступу документів, для цього натискаємо “Документи”:

					БКС 29.02.001.00 КРБ ПЗ	Арк.
						40
Ізм.	Лист	№ докум.	Підпис	Дата		

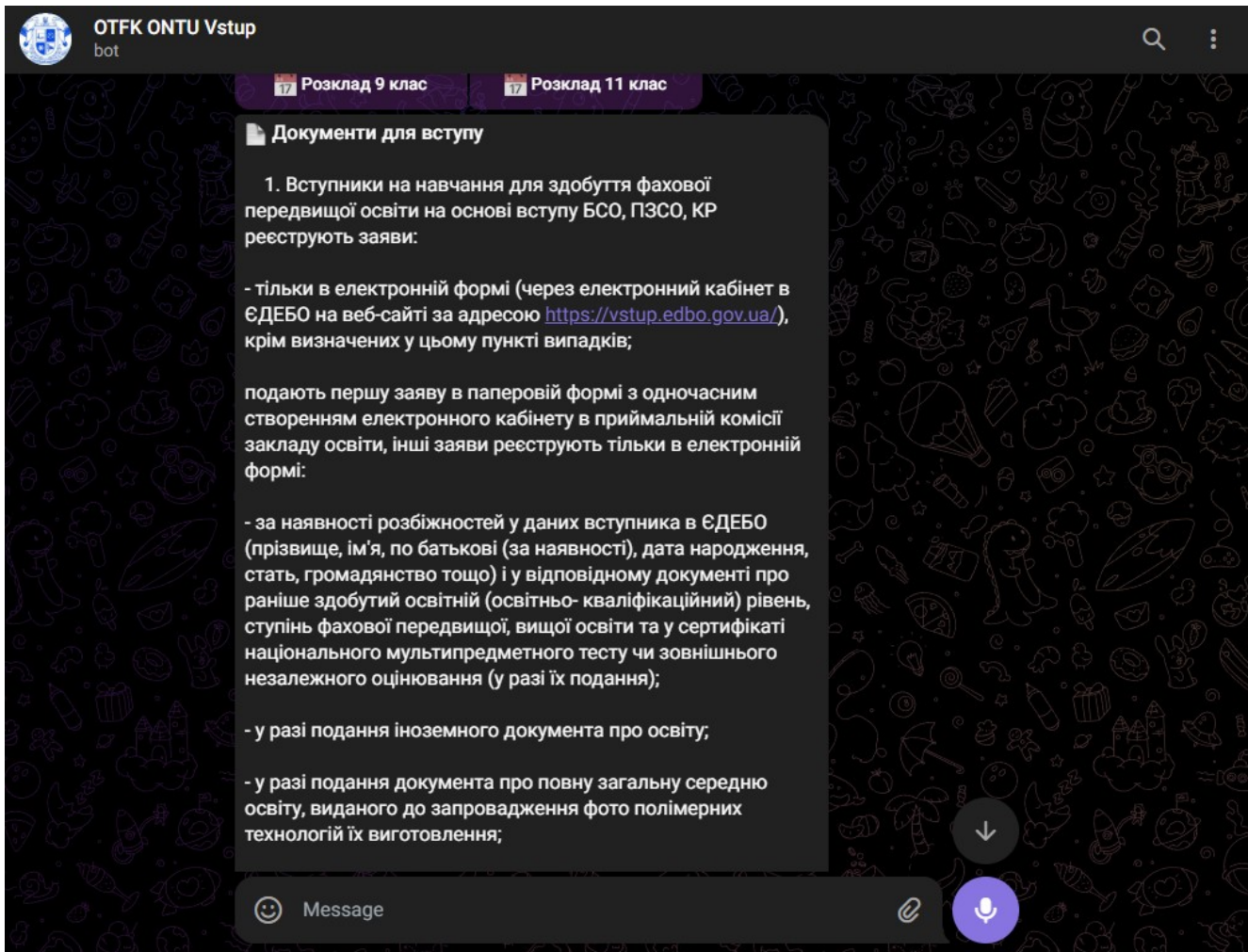


Рисунок 1.32. Перевірка команди /documents

Бот коректно відреагував на запит, надіславши список документів, що обов'язкові при поданні заяви до закладу освіти. Водночас варто зазначити, що усі дії користувача, зокрема натискання кнопок і запити, логуються в реальному часі у командному рядку. Це дуже корисно з точки зору адміністрування та моніторингу поведінки бота, а також дає змогу оперативно виявляти та усувати можливі помилки.:

```
PS C:\Users\forcr> & C:/Users/forcr/AppData/Local/Programs/Python/Python311/python.exe d:/Projects/diplom/main.py
INFO:aiogram.dispatcher:Start polling
INFO:aiogram.dispatcher:Run polling for bot @otk_vstup_bot id=8006344610 - 'OTFK ONTU Vstup'
INFO:aiogram.event:Update id=564463642 is handled. Duration 202 ms by bot id=8006344610
INFO:aiogram.event:Update id=564463643 is handled. Duration 125 ms by bot id=8006344610
```

Рисунок 1.33. Демонстрація логування

Для подальшої перевірки функціоналу протестуємо ще одну категорію запитів. Припустимо, користувач хоче дізнатися вартість навчання. Щоб це зробити, він повинен скористатись можливістю повернення до головного меню натиснувши кнопку “Назад до меню”, а потім вибрати пункт “Вартість”. Такий

підхід ілюструє наявність продуманої навігації, яка дозволяє з будь-якого місця в боті повернутися до початку:

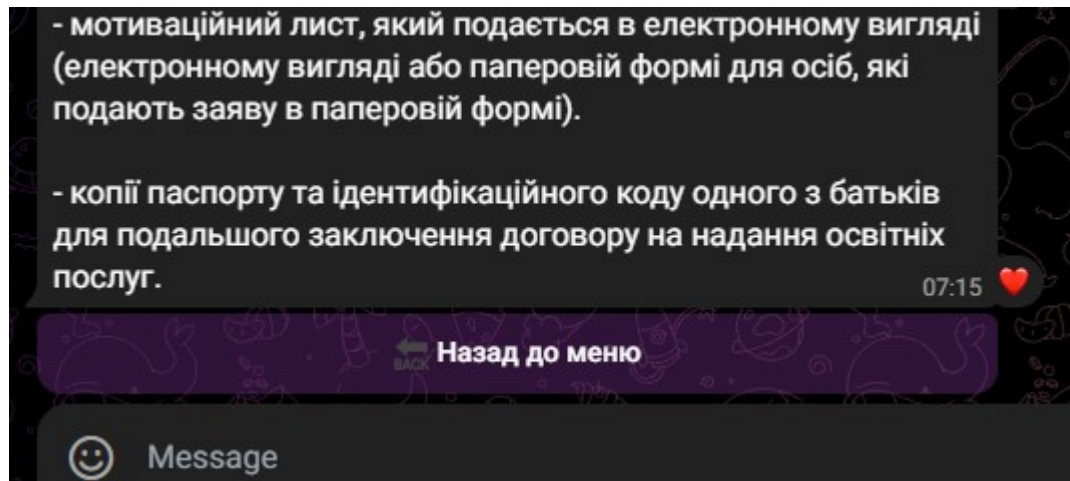


Рисунок 1.34. Демонстрація можливості повернутися в головне меню

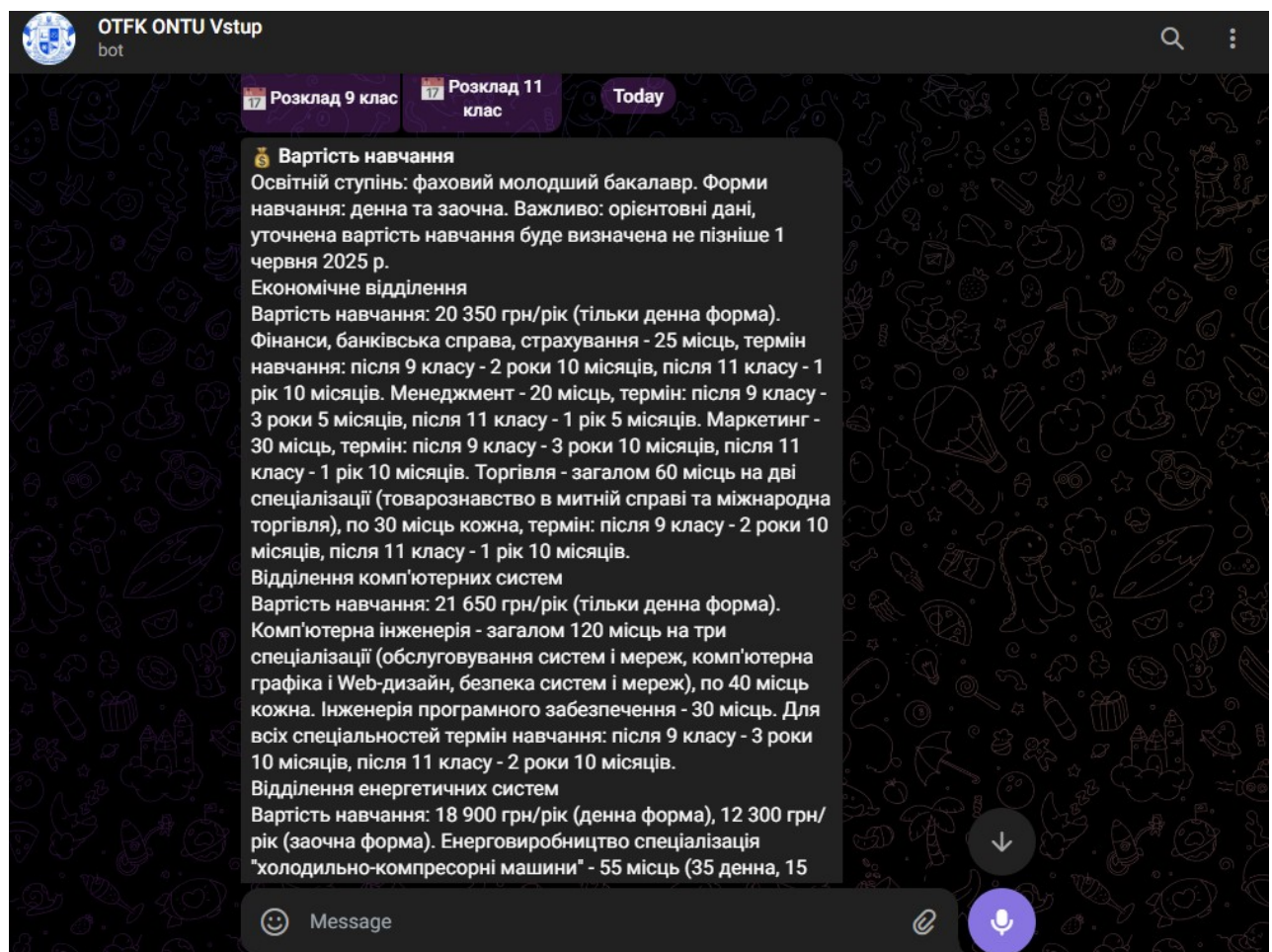


Рисунок 1.35. Відображення вартості

Відповідь на запит про вартість навчання надходить у вигляді структурованого текстового повідомлення. Звертаємо увагу на те, що дана інформація не була скопійована з текстового джерела, а формувалась вручну — першоджерело є сканом офіційного документа, тому для збереження

					БКС 29.02.001.00 КРБ ПЗ	Арк.
						42
Ізм.	Лист	№ докум.	Підпис	Дата		

достовірності наприкінці повідомлення було додано активне гіперпосилання на відповідний документ. Це дозволяє кожному охочому самостійно перевірити інформацію, ознайомившись із її офіційним джерелом.:

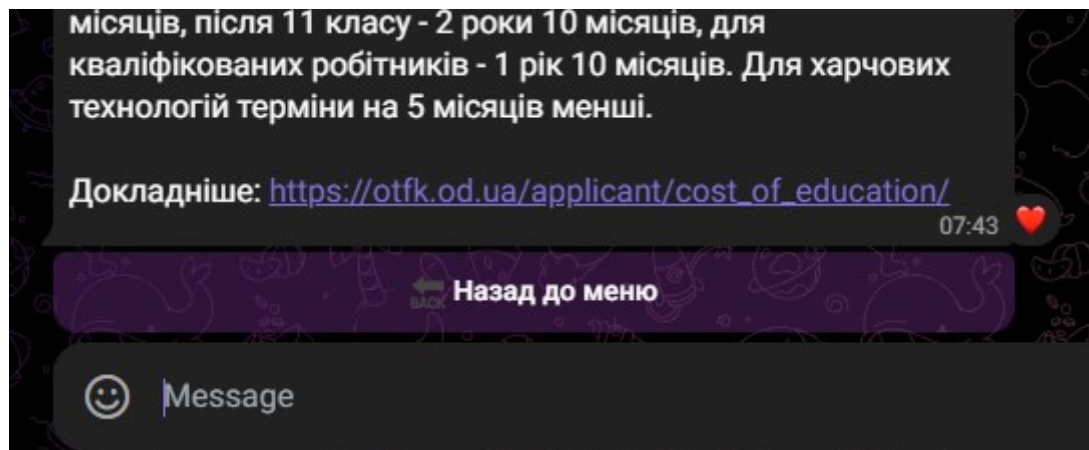


Рисунок 1.36. Демонстрація посилання в кінці

Натискання на гіперпосилання відкриває оригінал документа в новій вкладці, що забезпечує прозорість і підтвердження достовірності наданих ботом даних. Такий підхід не лише додає впевненості користувачеві, але й покращує довіру до інструменту в цілому:

cost_of_education_fmb.pdf 1 / 2 95%

Додаток 2
до Правил прийому 2025 р.

КОНКУРСНІ ПРОПОЗИЦІЇ
освітніх ступенів (освітньо-кваліфікаційних рівнів) та спеціальностей, за якими оголошується прийом на навчання,
ліцензовані обсяги та нормативні терміни навчання
Відокремлений структурний підрозділ «Одеський технічний фаховий коледж Одеського національного технологічного університету»
(назва закладу фахової передвищої освіти)
Освітній ступінь (освітньо-професійний рівень) – фаховий молодший бакалавр

Код	Назва	Код	Назва	Загальний ліцензований обсяг ВСЬОГО:	Освітньо-професійна програма /може повторювати назву спеціальності/	Ліцензовані обсяги			Нормативні терміни навчання		Вартість одного року навчання, грн. *	
						всього	у т.ч.		Денна форма навчання	Заочна форма навчання	Денна форма навчання	Заочна форма навчання
ЕКОНОМІЧНЕ ВІДДІЛЕННЯ												
D	Бізнес, адміністрування та право	D2	Фінанси, банківська справа, страхування та фондовий ринок	25	Фінанси, банківська справа та інтернет-банкінг	25	25	0	9 кл. – 2р.10 м. 11 кл. – 1р.10 м.	-	20350	-
D	Бізнес, адміністрування та право	D3	Менеджмент	20	Менеджмент та організація бізнесу	20	20	0	9 кл. – 3р.5 м. 11 кл. – 1р.5 м.	-	20350	-
D	Бізнес, адміністрування та право	D5	Маркетинг	30	IT-маркетинг та бізнес комунікації	30	30	0	9 кл. – 3р.10 м. 11 кл. – 1р.10 м.	-	20350	-
D	Бізнес, адміністрування та право	D7	Торгівля	60	Товарознавство в митній справі	30	30	-	9 кл. – 2р.10 м. 11 кл. – 1р.10 м.	-	20350	-
					Міжнародна торгівля та логістична діяльність	30	30	-	9 кл. – 2р.10 м. 11 кл. – 1р.10 м.	-	20350	-
ВІДДІЛЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ												
F	Інформаційні технології	F7	Комп'ютерна інженерія	120	Обслуговування комп'ютерних систем і мереж	40	40	0	9 кл. – 3р.10 м. 11 кл. – 2р.10 м.	-	21650	-
					Комп'ютерна графіка і Web-дизайн	40	40	0	9 кл. – 3р.10 м. 11 кл. – 2р.10 м.	-	21650	-
					Безпека комп'ютерних систем і мереж	40	40	0	9 кл. – 3р.10 м. 11 кл. – 2р.10 м.	-	21650	-
F	Інформаційні технології	F2	Інженерія програмного забезпечення	30	Розробка програмного забезпечення	30	30	0	9 кл. – 3р.10 м. 11 кл. – 2р.10 м.	-	21650	-

Рисунок 1.37. Демонстрація можливості перейти на оригінальний документ

					БКС 29.02.001.00 КРБ ПЗ	Арк. 43
Ізм.	Лист	№ докум.	Підпис	Дата		

Також, слід продемонструвати наяву гілок меню, яке реалізовано при виборі “Спеціальності”, перейдемо назад і натиснемо цю кнопку, тоді нам запропонують вибрати спочатку з двох варіантів:

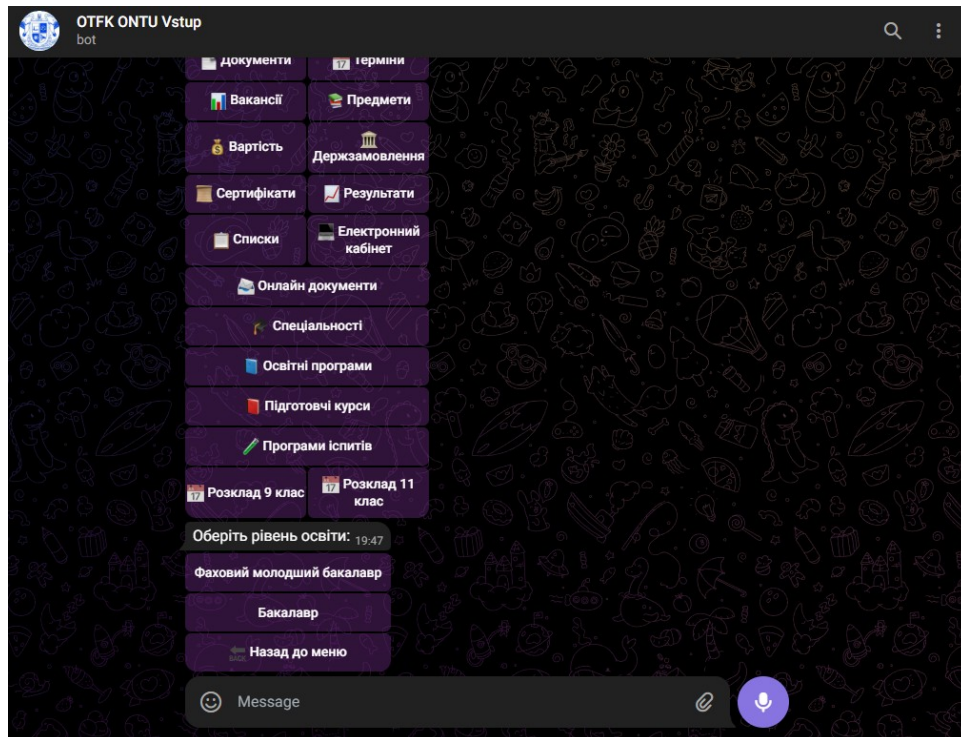


Рисунок 1.38. Демонстрація підменю вибору спеціальності

Натискаємо на “Фаховий молодший бакалавр”, після чого можемо вибрати якусь спеціальність:

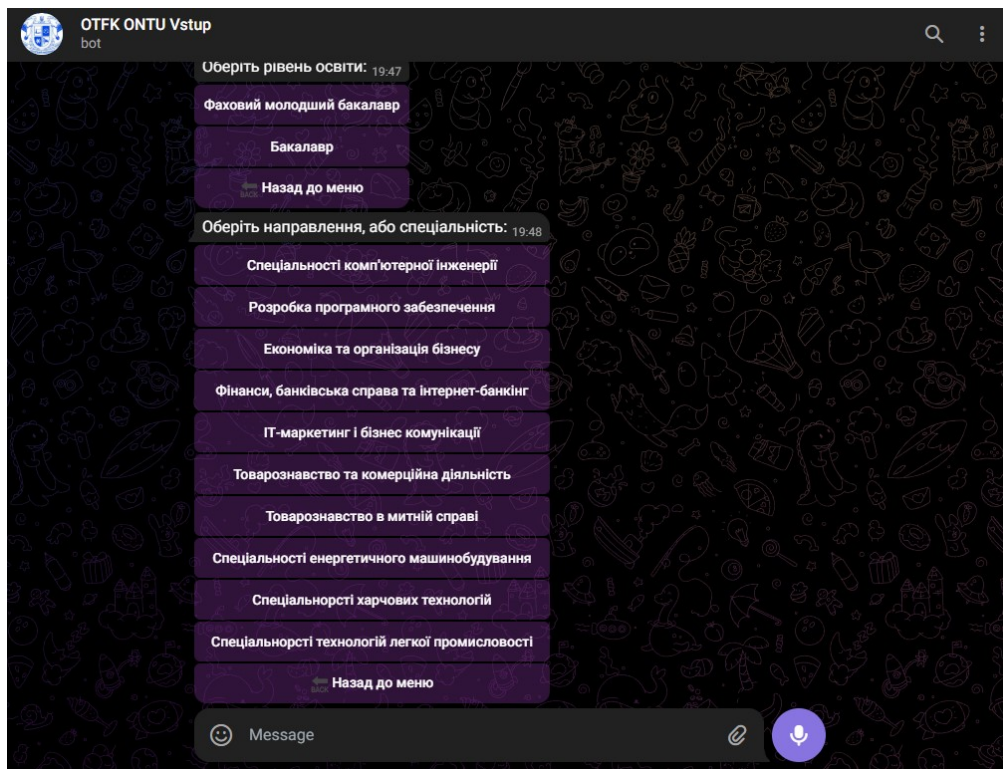


Рисунок 1.39. Спеціальності, дізнатись про які пропонує бот

Обираємо, наприклад “Товарознавство в митній справі”, бот видає наступну інформацію:

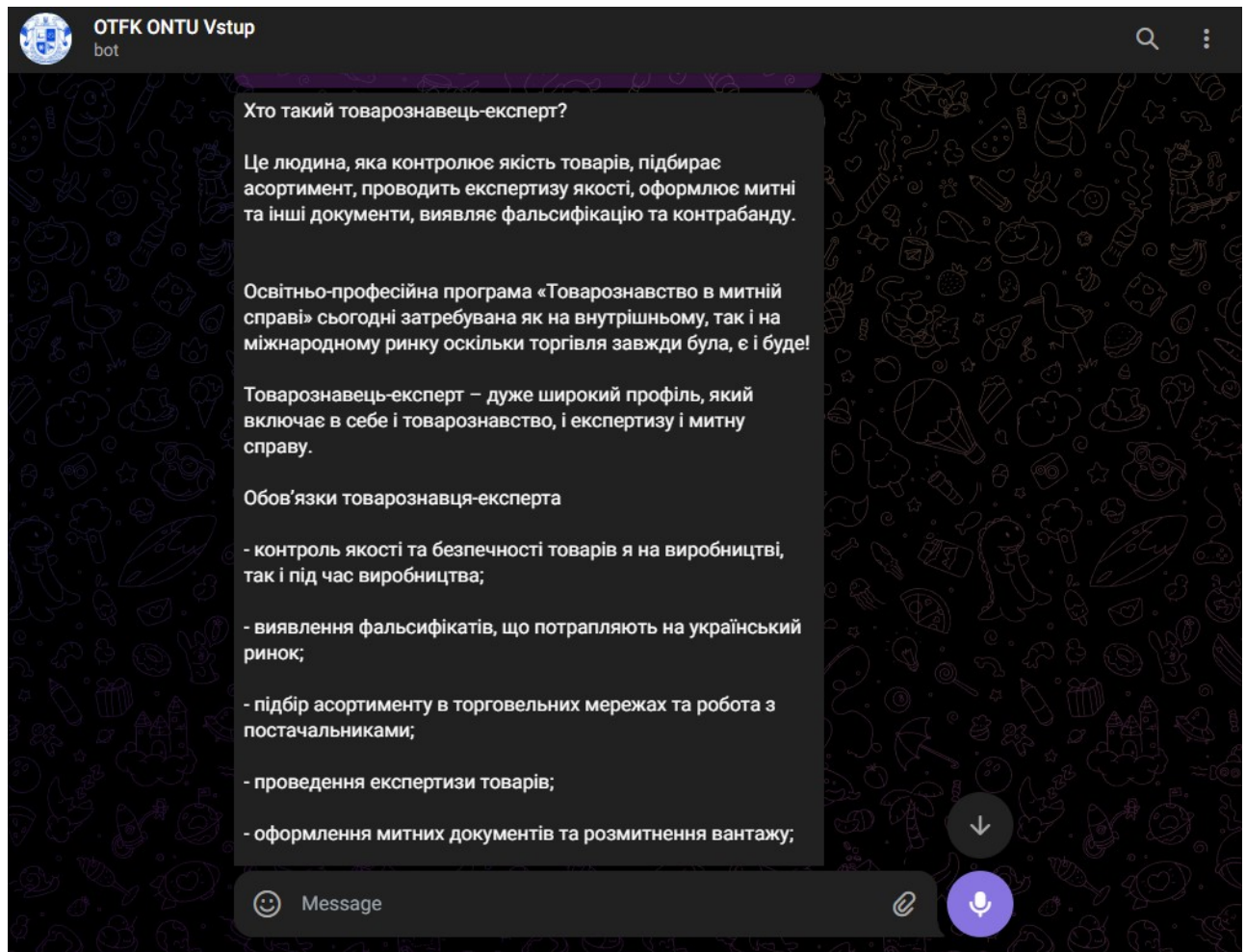


Рисунок 1.40. Інформація з боту про спеціальність

1.5.2 Тестування дій адміністратора

Далі було прийнято рішення протестувати можливість зміни інформації. Майбутньому адміністратору цього бота буде передан доступ тільки до файлу “messages.py”, для того, щоб він міг втручатися тільки у текст відповідей, не торкаючись логіки. Для того, щоб протестувати, чи все працює, ми припустемо, подивимося, що буде, якщо ми подивимося рейтингові списки. Звісно, що їх ще нема, тому у файлі написано:

```
196 lists = "" <b>Рейтингові списки</b>
197
198     Чекайте оприлюднення інформації""
199
```

Рисунок 1.41. Запис у “рейтингові списки”

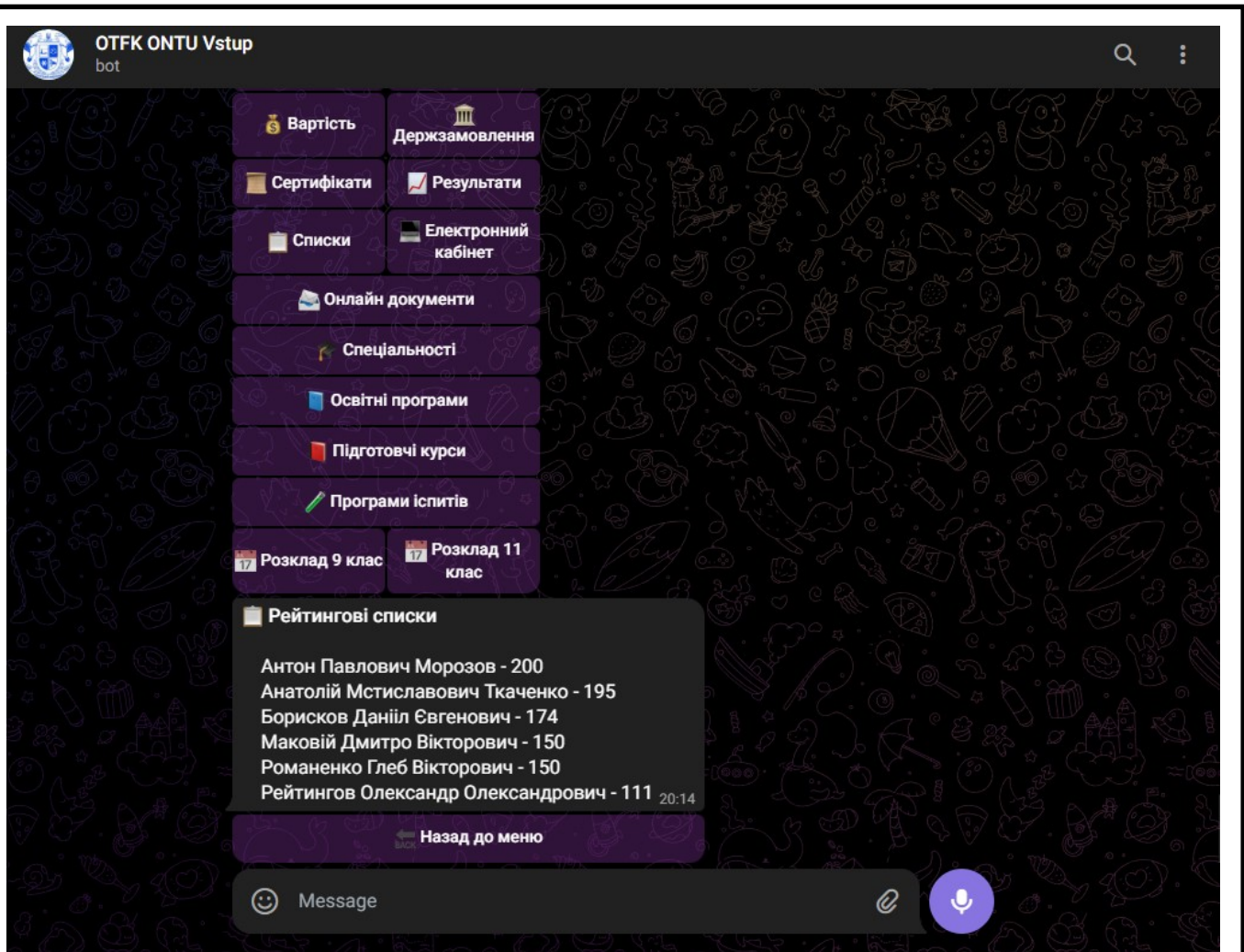


Рисунок 1.44. Зміни відображаються і у користувачів

Таким чином, у ході роботи було успішно реалізовано Telegram-бот, призначений для інформування абітурієнтів Одеського технічного фахового коледжу ОНТУ. Проєкт повністю відповідає початковому технічному завданню, передбачаючи автоматизоване надання вичерпної інформації, яка є необхідною під час вступної кампанії. Серед функціоналу, який був реалізований і протестований: надання переліку документів для вступу, терміни прийому, список спеціальностей, освітньо-професійні програми, розклад вступних іспитів, інформація про вартість навчання, рейтингові списки, державне замовлення, електронний кабінет та інші важливі аспекти.

Окрема увага була приділена зручності навігації: з будь-якого підменю можна повернутися до головного меню, що спрощує користування для абітурієнтів різного рівня цифрової грамотності.

Особливим досягненням проєкту стало розділення логіки програми та контентної частини. Усі текстові відповіді Telegram-бота були винесені у

					БКС 29.02.001.00 КРБ ПЗ	Арк.
						47
Ізм.	Лист	№ докум.	Підпис	Дата		

зовнішній файл messages.py, що дозволяє змінювати тексти повідомлень без втручання в основний програмний код. Це рішення було спеціально протестоване у ході розробки: адміністратор змінював текст відповіді на прикладі пункту “Рейтингові списки”, після чого перевірялась коректність оновленої відповіді у самому боті. Результати тестування довели, що система стабільно підхоплює нові тексти, що дає змогу швидко оновлювати інформацію в реальному часі без перезапуску логіки бота.

Цей підхід дає низку стратегічних переваг:

- Адміністративна простота — адміністратор, який не має технічної освіти, може самостійно змінювати відповіді бота в зручному файлі.
- Гнучкість — усі повідомлення можуть адаптуватися під нову вступну кампанію, змінюватися залежно від ситуації або актуалізуватися з року в рік без повторної розробки.
- Безпека — обмеження прав доступу до коду забезпечує, що адміністратор не зможе випадково пошкодити логіку роботи Telegram-бота.
- Масштабованість — така структура дозволяє у майбутньому реалізувати багатомовність або підключення до зовнішніх джерел даних, наприклад, бази даних або CMS

					БКС 29.02.001.00 КРБ ПЗ	Арк.
						48
Ізм.	Лист	№ докум.	Підпис	Дата		

2 РОЗДІЛ ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

У сучасних умовах охорона праці є ключовим елементом забезпечення безпечних і здорових умов на робочому місці, незалежно від сфери діяльності підприємства. Вона спрямована на запобігання нещасним випадкам, професійним захворюванням та іншим шкідливим наслідкам трудової діяльності. Застосування заходів з охорони праці дозволяє не лише зберегти життя і здоров'я працівників, а й підвищити ефективність праці, зменшити витрати, пов'язані з травматизмом та лікарняними.

Організація безпечних умов праці особливо важлива при роботі з екранними пристроями, що передбачає тривале перебування за комп'ютером. Такі умови регламентуються нормативно-правовими актами, зокрема НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» та НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин».

У цьому розділі буде проаналізовано основні небезпечні та шкідливі фактори, пов'язані з організацією робочого місця користувача персонального комп'ютера, розглянуто вимоги до приміщення, освітлення, мікроклімату, вентиляції, електробезпеки та пожежної безпеки, а також запропоновано заходи для забезпечення комфортних та безпечних умов праці.

2.1 Аналіз шкідливих та небезпечних факторів виробничого середовища

До шкідливих факторів під час роботи розробника належать: невідповідна температура в приміщенні (занадто висока або низька), надмірний шум, порушення рівня вологості та руху повітря. Також впливати можуть зміни іонного складу повітря, електромагнітне випромінювання від техніки, недостатнє або надмірне освітлення, наявність відблисків та мерехтіння екранів. Серед психофізичних факторів варто виділити перевантаження нервової системи,

					БКС 29.02.002.00 КРБ ПЗ	Арк.
						49
Ізм.	Лист	№ докум.	Підпис	Дата		

спричинені монотонною працею, постійною концентрацією уваги та емоційною напругою.

2.2 Вимоги до приміщення

Для комфортної та безпечної роботи з комп'ютерами площа приміщення на одну особу повинна бути не меншою за 6 м² — таку норму встановлено ДСанПіН 3.3.2.007-98. Відповідно до цього ж документа, рівень шуму на робочому місці не має перевищувати 50 децибел, а вібраційні впливи необхідно мінімізувати. Робоче місце повинно бути обладнане столом з можливістю регулювання висоти, а екран монітора має розміщуватись на відстані 50–70 см від очей. Згідно з ДБН В.2.5-23:2010, усі електроустановки в таких приміщеннях повинні відповідати вимогам електробезпеки: електропроводка — захищена, а вся техніка — належним чином заземлена.

2.3 Освітлення приміщення

Згідно з вимогами ДБН В.2.5-28:2006 "Природне і штучне освітлення", у приміщеннях, де здійснюється робота з персональними комп'ютерами, повинно бути організоване комбіноване освітлення — тобто поєднання природного та штучного. Це необхідно для створення сприятливих умов праці та зменшення втоми очей. Коефіцієнт природного освітлення (КПО) має становити не менше 1,5%, що забезпечує достатній рівень освітленості вдень завдяки проникненню сонячного світла через вікна. При цьому вікна повинні мати відповідну площу, бути чистими та не затінюватися сторонніми предметами.

Штучне освітлення, у свою чергу, повинно створювати належні умови для роботи у темний час доби або при недостатньому природному світлі. Нормативний рівень освітленості для робочої зони повинен становити від 300 до 500 люкс. Такі показники досягаються завдяки використанню загального та локального освітлення. Загальне освітлення реалізується за допомогою стельових

					БКС 29.02.002.00 КРБ ПЗ	Арк.
						50
Ізм.	Лист	№ докум.	Підпис	Дата		

світильників, розміщених рівномірно по всьому приміщенню, що дозволяє уникнути зон затемнення. Додатково використовується місцеве (індивідуальне) освітлення — настільні лампи або інші світильники, які встановлюються безпосередньо біля робочого місця для підсвічування монітора, клавіатури та документів. Таке поєднання джерел світла дозволяє уникнути перевантаження зору та підвищити комфортність роботи за комп'ютером.

Відповідно до ДСанПіН 3.3.2.007-98 "Державні санітарні норми та правила роботи з відеодисплейними терміналами електронно-обчислювальних машин", рівень освітленості на робочих місцях операторів ПК повинен відповідати наступним значенням:

Таблиця 2.1 - Рекомендовані рівні освітленості

Зона освітлення	Рекомендована освітленість (люкс)
Основне робоче місце	300-500
Зона для документів	500
Прохідні зони і коридори	150-200

Освітлення не повинно створювати відблисків на екранах моніторів, тому джерела світла слід розташовувати таким чином, щоб прямі промені не потрапляли на екран. Для цього використовуються світильники з розсіяним або відбивним світлом, а також екрани або шторки на вікнах для регулювання інтенсивності природного світла.

Джерела штучного світла повинні мати спектр, максимально наближений до природного денного світла, з колірною температурою 4000-5000 К.

Також важливо враховувати розташування робочих місць відносно джерел світла. Робочі місця повинні бути розташовані перпендикулярно до вікон, щоб уникнути відблисків, і на відстані не менше 1,5 м від стін з вікнами для забезпечення рівномірного освітлення.

2.4 Вентиляція та мікроклімат

Вентиляція та мікроклімат приміщень для операторів ПК мають відповідати вимогам ДСанПіН 3.3.2.007-98, забезпечуючи температурний режим від 18 до 24°C і відносну вологість повітря від 40 до 60%. Повітрообмін має гарантувати видалення шкідливих речовин та забезпечення свіжого повітря.

Таблиця 2.2 - Норми мікроклімату для приміщень з ВДТ ЕОМ та ПЕМ

Пора року	Категорія робіт	Температура повітря, С, не більше	Відносна вологість повітря %	Швидкість руху повітря, м/с
Холодна	Легка-1а	22-24	40-60	0,1
	Легка-1б	21-23	40-60	0,1
Тепла	Легка-1а	23-25	40-60	0,1
	Легка-1б	22-24	40-60	0,1

2.5 Пожежна безпека

Пожежна безпека являє собою стан, за якого створюються умови для мінімізації ймовірності виникнення пожежі, а також забезпечується захист життя і здоров'я людей, матеріальних цінностей та навколишнього середовища у разі її виникнення. Забезпечення пожежної безпеки досягається шляхом реалізації комплексу організаційних, профілактичних та технічних заходів, що відповідають чинним нормативним вимогам.

Однією з ключових вимог до протипожежного захисту є наявність евакуаційних шляхів і виходів. Відповідно до положень ДБН В.1.1-7:2016, такі шляхи повинні бути спроектовані так, щоб у випадку пожежі забезпечити оперативне і безпечне виведення людей з приміщень. Евакуаційні виходи мають бути завжди вільними, без перешкод, чітко позначеними світловими знаками, що видно навіть при поганій видимості. Важливо, щоб двері евакуаційних виходів відкривалися у напрямку виходу і не блокувалися меблями чи обладнанням.

Ще одним важливим елементом системи протипожежного захисту є система

виявлення пожежі та оповіщення. Відповідно до ДБН В.2.5-56:2014, у приміщеннях, де розміщені робочі місця з використанням комп'ютерної техніки, необхідно встановлювати автоматичні системи пожежної сигналізації, які здатні виявляти ознаки займання (зокрема дим чи аномально високу температуру) та негайно передавати сигнал тривоги. Системи оповіщення мають інформувати людей про необхідність евакуації, використовуючи звукові та світлові сигнали.

Окрему увагу слід приділяти засобам пожежогасіння. Згідно з вимогами ДБН В.2.5-56:2014, кожне приміщення повинне бути обладнане первинними засобами гасіння пожеж — вогнегасниками, пожежними кранами або пожежними ковдрами. Тип і кількість засобів визначається з урахуванням площі приміщення та категорії пожежної небезпеки. Наприклад, на кожні 100 квадратних метрів площі має бути встановлено принаймні один порошковий або вуглекислотний вогнегасник. Усі засоби пожежогасіння повинні розміщуватися у легкодоступних місцях та регулярно перевірятися на справність.

Таким чином, дотримання норм пожежної безпеки є обов'язковою умовою організації безпечного середовища в офісних чи освітніх приміщеннях, де здійснюється робота з комп'ютерами. Це дозволяє ефективно реагувати на надзвичайні ситуації та зменшує ризики для життя та здоров'я працівників.

					БКС 29.02.002.00 КРБ ПЗ	Арк.
						53
Ізм.	Лист	№ докум.	Підпис	Дата		

ВИСНОВКИ

У межах даного дипломного проєкту було успішно реалізовано Telegram-бота, що виконує функцію інформаційного помічника для абітурієнтів Одеського технічного фахового коледжу ОНТУ. Основною метою розробки було створення цифрового інструменту, який би надавав актуальну, структуровану та доступну інформацію про вступ до навчального закладу для молодшої аудиторії віком 14–19 років.

У ході проєкту було виконано повний цикл розробки програмного забезпечення: від аналізу цільової аудиторії, постановки задач та формування технічного завдання — до реалізації функціоналу, тестування і підготовки до розгортання. Основу технологічного стеку склала мова програмування Python у поєднанні з асинхронним фреймворком Aiogram, що забезпечило ефективну роботу з Telegram Bot API, гнучкість архітектури та масштабованість рішення.

Бот реалізує функціонал, що дозволяє користувачеві дізнатися:

- перелік необхідних для вступу документів;
- терміни прийому заяв та іспитів;
- перелік спеціальностей та конкурсних предметів;
- вартість навчання за кожною спеціальністю;
- обсяг державного замовлення;
- особливості електронного кабінету вступника;
- результати вступних випробувань та рейтингові списки.

Завдяки зручному кнопковому меню, користувач може швидко орієнтуватися у доступному функціоналі та повертатися до головного меню з будь-якого пункту. Це відповідає вимогам зручності та інтуїтивності, особливо важливим у роботі з підлітками. Крім того, всі повідомлення розділено на окремі файли, що дозволяє адміністратору редагувати відповіді без заглиблення у код або змінення логіки — достатньо лише відредагувати текст у файлі `messages.py`.

У рамках тестування були перевірені:

- коректність обробки команд та навігації;
- швидкодія бота при запуску;

					БКС 29.02.000.00 КРБ ПЗ	Арк.
						54
Ізм.	Лист	№ докум.	Підпис	Дата		

- зручність повернення до головного меню;
- можливість швидкої модифікації інформації адміністратором (наприклад, додавання актуальних рейтингових списків);
- стабільність логування дій у терміналі для моніторингу взаємодії користувачів.

Окрему увагу було приділено безпеці та структурі коду: усі токени збережено в окремому конфігураційному файлі, код модульний, з чітко розмежованими зонами відповідальності. Це відкриває шлях до масштабування проекту в майбутньому: додавання інтерактивних запитів, зв'язку з базами даних, впровадження персоналізованих відповідей на основі введених користувачем параметрів (наприклад, вибір спеціальностей за балами).

Проект також відповідає сучасним трендам цифровізації освіти. Враховуючи високу популярність Telegram серед молоді в Україні (82–86% у віці 13–35 років), саме ця платформа стала найбільш доречною для реалізації бота. Рішення виявилось не лише ефективним, а й зручним, доступним і готовим до інтеграції з освітнім середовищем.

Таким чином, можна зробити висновок, що реалізоване рішення повністю відповідає поставленим вимогам технічного завдання, виконує свою функцію та має потенціал до масштабування. Telegram-бот виконує роль надійного цифрового консультанта для абітурієнтів, полегшуючи їм процес ознайомлення з правилами вступу, підвищуючи інформованість та зменшуючи навантаження на адміністрацію навчального закладу.

Подальші перспективи розвитку включають:

- підключення повноцінної бази даних для зберігання історії запитів і персоналізації;
- створення адмін-панелі для зручного редагування повідомлень та графіку вступу;
- реалізацію мультимедійного контенту (зображень, відеоінструкцій, графіків тощо);
- інтеграцію з системами електронної подачі документів або інформування через API.

					БКС 29.02.000.00 КРБ ПЗ	Арк.
						55
<i>Ізм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Основи Python. Навчіться думати як програміст. / Аллен Б. Дауні. - "Манн, Іванов і Фербер", 2021 р. - 304 с.
2. Праворська Н.І. Інформатика та комп'ютерна техніка: Навчально-методичний посібник для студентів вищих навчальних закладів - Хмельницький, 2002 - 312с.
3. Простий Python. Сучасний стиль програмування. 2 видання. / Любанович Білл. - "Видавничий дім "Пітер"", 2020 р. - 592 с.
4. Національне опитування молоді. Хвиля 3 [Електронний ресурс] Режим доступу до ресурсу: <https://surl.li/fwetti> (дата звернення 20.05.2025)
5. Дослідження. Медіаспоживання українців [Електронний ресурс] Режим доступу до ресурсу: <https://surl.lu/suwhea> (дата звернення 20.05.2025)
6. Єдина державна електронна база з питань освіти [Електронний ресурс] Режим доступу до ресурсу: <https://info.edbo.gov.ua/> (дата звернення 20.05.2025)
4. ТиКиїв [Електронний ресурс] Режим доступу до ресурсу: <https://surl.li/atueyo> (дата звернення 20.05.2025)
5. Філологічний факультет Львівського Національного Університету [Електронний ресурс] Режим доступу до ресурсу: <https://surl.lu/sbllmh> (дата звернення 20.05.2025)
6. Aiogram Documentation [Електронний ресурс] Режим доступу до ресурсу: <https://docs.aiogram.dev/en/v3.20.0.post0/> (дата звернення 20.05.2025)
7. Python Documentation [Електронний ресурс] Режим доступу до ресурсу: <https://docs.python.org/uk/3.13/tutorial/index.html> (дата звернення 20.05.2025)
8. Visual Studio Code documentation [Електронний ресурс] Режим доступу до ресурсу: <https://code.visualstudio.com/docs> (дата звернення 20.05.2025)
9. Asyncio Documentation [Електронний ресурс] Режим доступу до ресурсу: <https://docs.python.org/3/library/asyncio.html> (дата звернення 20.05.2025)

					БКС 29.02.000.00 КРБ ПЗ	Арк.
						56
Ізм.	Лист	№ докум.	Підпис	Дата		

10. ДСанПіН 3.3.6.042-99 «Державні санітарні норми мікроклімату виробничих приміщень».

11 .ДСанПіН 2.3.6.037-99 «Державні санітарні норми виробничого шуму, ультразвуку та інфразвуку»

					БКС 29.02.000.00 КРБ ПЗ	Арк.
						57
Ізм.	Лист	№ докум.	Підпис	Дата		

ДОДАТОК А. Фрагмент коду головного алгоритму мовою Python чат-боту для абітурієнтів коледжу

```
import logging
import asyncio
from aiogram import Bot, Dispatcher, types, F
from aiogram.types import InlineKeyboardMarkup, InlineKeyboardButton
import messages, variables

logging.basicConfig(level=logging.INFO)
bot = Bot(token=variables.TOKEN)
dp = Dispatcher()

main_menu = InlineKeyboardMarkup(inline_keyboard=[
    [InlineKeyboardButton(text="📄 Документи", callback_data="documents"),
     InlineKeyboardButton(text="📅 Терміни", callback_data="dates")],
    [InlineKeyboardButton(text="📅 Вакансії", callback_data="vacancies"),
     InlineKeyboardButton(text="📚 Предмети", callback_data="subjects")],
    [InlineKeyboardButton(text="💰 Вартість", callback_data="cost"),
     InlineKeyboardButton(text="🏠 Держзамовлення", callback_data="state_order")],
    [InlineKeyboardButton(text="📄 Сертифікати", callback_data="certificates"),
     InlineKeyboardButton(text="📄 Результати", callback_data="results")],
    [InlineKeyboardButton(text="📄 Списки", callback_data="lists"),
     InlineKeyboardButton(text="📄 Електронний кабінет", callback_data="cabinet")],
    [InlineKeyboardButton(text="📄 Онлайн документи", callback_data="online_docs")],
    [InlineKeyboardButton(text="📄 Спеціальності", callback_data="specialties")],
    [InlineKeyboardButton(text="📄 Освітні програми", callback_data="programs")],
    [InlineKeyboardButton(text="📄 Підготовчі курси", callback_data="courses")],
    [InlineKeyboardButton(text="📄 Програми іспитів", callback_data="exams_programs")],
    [InlineKeyboardButton(text="📄 Розклад 9 клас", callback_data="schedule_9"),
     InlineKeyboardButton(text="📄 Розклад 11 клас", callback_data="schedule_11")],
])

back_button = InlineKeyboardMarkup(inline_keyboard=[
    [InlineKeyboardButton(text="🏠 Назад до меню", callback_data="back_to_menu")]
])

specialties_menu = InlineKeyboardMarkup(inline_keyboard=[
    [InlineKeyboardButton(text="Фаховий молодший бакалавр", callback_data="spec_1")],
    [InlineKeyboardButton(text="Бакалавр", callback_data="spec_2")],
    [InlineKeyboardButton(text="🏠 Назад до меню", callback_data="back_to_menu")]
])

ml_bac_menu = InlineKeyboardMarkup(inline_keyboard=[
    [InlineKeyboardButton(text="Спеціальності комп'ютерної інженерії", callback_data="bac_1")],
    [InlineKeyboardButton(text="Розробка програмного забезпечення", callback_data="bac_2")],
    [InlineKeyboardButton(text="Економіка та організація бізнесу", callback_data="bac_3")],
])
```

```

    [InlineKeyboardButton(text="Фінанси, банківська справа та інтернет-банкінг",
callback_data="bac_4")],
    [InlineKeyboardButton(text="ІТ-маркетинг і бізнес комунікації", callback_data="bac_5")],
    [InlineKeyboardButton(text="Товарознавство та комерційна діяльність",
callback_data="bac_6")],
    [InlineKeyboardButton(text="Товарознавство в митній справі", callback_data="bac_7")],
    [InlineKeyboardButton(text="Спеціальності енергетичного машинобудування",
callback_data="bac_8")],
    [InlineKeyboardButton(text="Спеціальності харчових технологій", callback_data="bac_9")],
    [InlineKeyboardButton(text="Спеціальності технологій легкої промисловості",
callback_data="bac_10")],
    [InlineKeyboardButton(text="⏪ Назад до меню", callback_data="back_to_menu")]
]
)

```

```

@dp.message(F.text == "/start")
async def send_welcome(message: types.Message):
    await message.answer(
        "Вітаю в чат-боті для абітурієнтів ОТФК ОНТУ! ⏪\n\n"
        "Оберіть потрібну тему з меню нижче:",
        reply_markup=main_menu
    )

```

```

@dp.callback_query(F.data == "documents")
async def cb_documents(callback: types.CallbackQuery):
    await callback.message.answer(messages.documents, parse_mode='HTML',
reply_markup=back_button)
    await callback.answer()

```

```

@dp.callback_query(F.data == "dates")
async def cb_dates(callback: types.CallbackQuery):
    await callback.message.answer(messages.dates, parse_mode='HTML', reply_markup=back_button)
    await callback.answer()

```

```

@dp.callback_query(F.data == "vacancies")
async def cb_vacancies(callback: types.CallbackQuery):
    await callback.message.answer(messages.vacancies, parse_mode='HTML',
reply_markup=back_button)
    await callback.answer()

```

```

@dp.callback_query(F.data == "subjects")
async def cb_subjects(callback: types.CallbackQuery):
    await callback.message.answer(messages.subjects, parse_mode='HTML',
reply_markup=back_button)
    await callback.answer()

```

```

@dp.callback_query(F.data == "cost")
async def cb_cost(callback: types.CallbackQuery):
    await callback.message.answer(messages.cost, parse_mode='HTML', reply_markup=back_button)
    await callback.answer()

```

```
@dp.callback_query(F.data == "state_order")
async def cb_state_order(callback: types.CallbackQuery):
    await callback.message.answer(messages.state_order, parse_mode='HTML',
reply_markup=back_button)
    await callback.answer()
```

```
@dp.callback_query(F.data == "certificates")
async def cb_certificates(callback: types.CallbackQuery):
    await callback.message.answer(messages.certificates, parse_mode='HTML',
reply_markup=back_button)
    await callback.answer()
```

```
@dp.callback_query(F.data == "results")
async def cb_results(callback: types.CallbackQuery):
    await callback.message.answer(messages.results, parse_mode='HTML', reply_markup=back_button)
    await callback.answer()
```

```
@dp.callback_query(F.data == "lists")
async def cb_lists(callback: types.CallbackQuery):
    await callback.message.answer(messages.lists, parse_mode='HTML', reply_markup=back_button)
    await callback.answer()
```

```
@dp.callback_query(F.data == "cabinet")
async def cb_cabinet(callback: types.CallbackQuery):
    await callback.message.answer(messages.cabinet, parse_mode='HTML',
reply_markup=back_button)
    await callback.answer()
```

```
@dp.callback_query(F.data == "online_docs")
async def cb_online_docs(callback: types.CallbackQuery):
    await callback.message.answer(messages.online_docs, parse_mode='HTML',
reply_markup=back_button)
    await callback.answer()
```

```
@dp.callback_query(F.data == "specialties")
async def cb_specialties(callback: types.CallbackQuery):
    await callback.message.answer("Оберіть рівень освіти:", reply_markup=specialties_menu)
    await callback.answer()
```

```
@dp.callback_query(F.data == "spec_1")
async def cb_spec_1(callback: types.CallbackQuery):
    await callback.message.answer("Оберіть направлення, або спеціальність:",
reply_markup=ml_bac_menu)
    await callback.answer()
```

```
@dp.callback_query(F.data == "bac_1")
async def cb_bac_1(callback: types.CallbackQuery):
    await callback.message.answer(messages.s_123_1, parse_mode='HTML')
    await callback.message.answer(messages.s_123_2, parse_mode='HTML')
```

```
    await callback.message.answer(messages.s_123_3, parse_mode='HTML',
reply_markup=back_button)
    await callback.answer()
```

```
@dp.callback_query(F.data == "bac_2")
async def cb_bac_2(callback: types.CallbackQuery):
    await callback.message.answer(messages.s_121, parse_mode='HTML', reply_markup=back_button)
    await callback.answer()
```

```
@dp.callback_query(F.data == "bac_3")
async def cb_bac_3(callback: types.CallbackQuery):
    await callback.message.answer(messages.s_051, parse_mode='HTML', reply_markup=back_button)
    await callback.answer()
```

```
@dp.callback_query(F.data == "bac_4")
async def cb_bac_4(callback: types.CallbackQuery):
    await callback.message.answer(messages.s_072, parse_mode='HTML', reply_markup=back_button)
    await callback.answer()
```

```
@dp.callback_query(F.data == "bac_5")
async def cb_bac_5(callback: types.CallbackQuery):
    await callback.message.answer(messages.s_IT, parse_mode='HTML', reply_markup=back_button)
    await callback.answer()
```

```
@dp.callback_query(F.data == "bac_6")
async def cb_bac_6(callback: types.CallbackQuery):
    await callback.message.answer(messages.s_tovar, parse_mode='HTML',
reply_markup=back_button)
    await callback.answer()
```

```
@dp.callback_query(F.data == "bac_7")
async def cb_bac_7(callback: types.CallbackQuery):
    await callback.message.answer(messages.s_076, parse_mode='HTML', reply_markup=back_button)
    await callback.answer()
```

```
@dp.callback_query(F.data == "bac_8")
async def cb_bac_8(callback: types.CallbackQuery):
    await callback.message.answer(messages.s_142_1, parse_mode='HTML'),
    await callback.message.answer(messages.s_142_2, parse_mode='HTML',
reply_markup=back_button)
    await callback.answer()
```

```
@dp.callback_query(F.data == "bac_9")
async def cb_bac_9(callback: types.CallbackQuery):
    await callback.message.answer(messages.s_181_1, parse_mode='HTML'),
    await callback.message.answer(messages.s_181_2, parse_mode='HTML',
reply_markup=back_button)
    await callback.answer()
```

```
@dp.callback_query(F.data == "bac_10")
```

```

async def cb_bac_10(callback: types.CallbackQuery):
    await callback.message.answer(messages.s_182_1, parse_mode='HTML'),
    await callback.message.answer(messages.s_182_2, parse_mode='HTML',
reply_markup=back_button)
    await callback.answer()

@dp.callback_query(F.data == "spec_2")
async def cb_spec_2(callback: types.CallbackQuery):
    await callback.message.answer(messages.s_b_en, parse_mode='HTML')
    await callback.message.answer(messages.s_b_comp_eng, parse_mode='HTML')
    await callback.message.answer(messages.s_b_econ, parse_mode='HTML',
reply_markup=back_button)
    await callback.answer()

@dp.callback_query(F.data == "programs")
async def cb_programs(callback: types.CallbackQuery):
    await callback.message.answer(messages.programs, parse_mode='HTML')
    await callback.message.answer(messages.programs2, parse_mode='HTML')
    await callback.message.answer(messages.programs3, parse_mode='HTML')
    await callback.message.answer(messages.programs4, parse_mode='HTML')
    await callback.message.answer(messages.programs5, parse_mode='HTML')
    await callback.message.answer(messages.programs6, parse_mode='HTML',
reply_markup=back_button)
    await callback.answer()

@dp.callback_query(F.data == "courses")
async def cb_courses(callback: types.CallbackQuery):
    await callback.message.answer(messages.courses, parse_mode='HTML',
reply_markup=back_button)
    await callback.answer()

@dp.callback_query(F.data == "exams_programs")
async def cb_exams_programs(callback: types.CallbackQuery):
    await callback.message.answer(messages.exams_programs, parse_mode='HTML',
reply_markup=back_button)
    await callback.answer()

@dp.callback_query(F.data == "back_to_menu")
async def cb_back(callback: types.CallbackQuery):
    await callback.message.answer("☐ Головне меню:", reply_markup=main_menu)
    await callback.answer()

# Запуск
async def main():
    await dp.start_polling(bot)

if __name__ == "__main__":
    asyncio.run(main())

```

ДОДАТОК Б. Слайди мультимедійної презентації



Дипломний проект

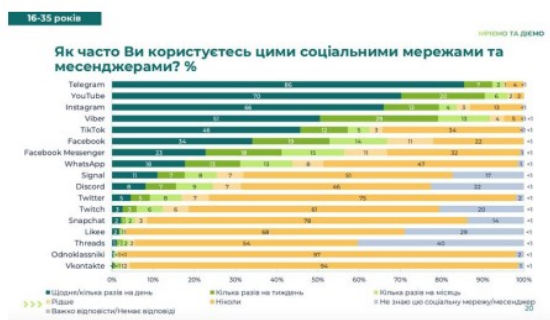
Розробка чат-бота для абітурієнтів коледжу ОТФК ОНТУ

Виконав:
Борисков Данііл Євгенович
Науковий керівник:
Іванова Лілія Вікторівна
Кафедра:
Комп'ютерних технологій

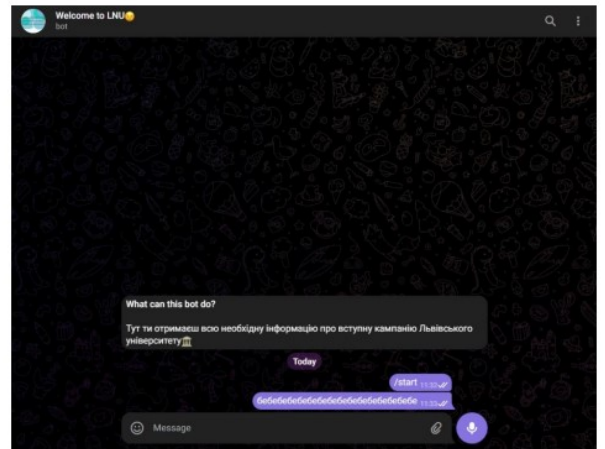
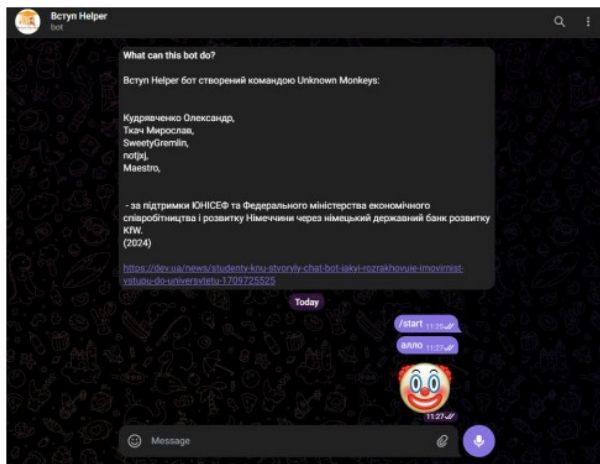
Одеса, 2025



Опитування молоді стосовно використання соц. мереж та месенджерів



Непрацездатність аналогів

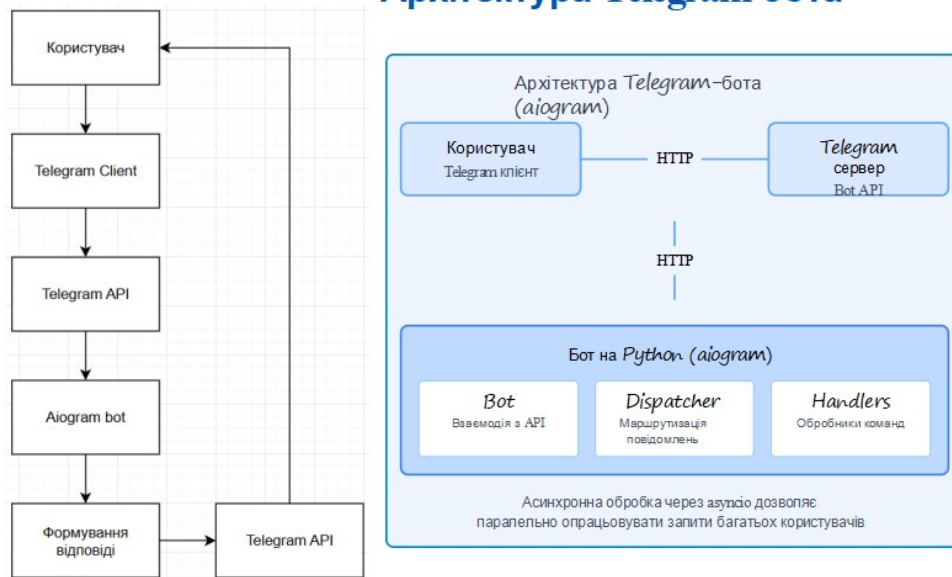


Відображена інформація про бота

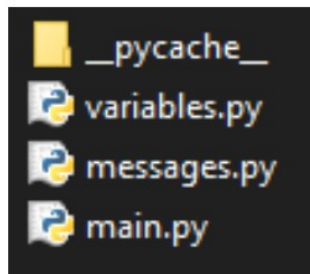




Архітектура Telegram-бота

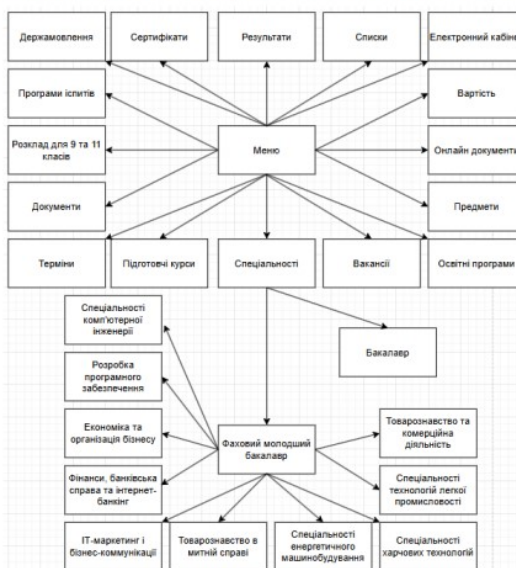


Структура файлів проекту





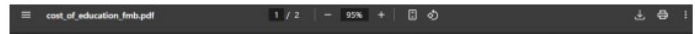
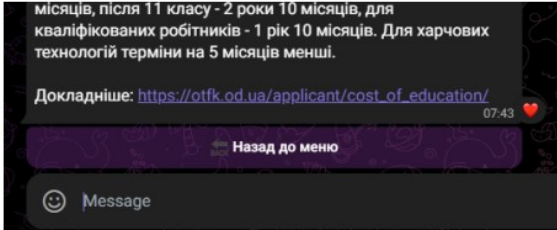
Функціональність, меню Telegram-бота



Логування за допомогою logging

```
PS C:\Users\forcr> & C:/Users/forcr/AppData/Local/Programs/Python/Python311/python.exe d:/Projects/diplom/main.py
INFO:aiogram.dispatcher:Start polling
INFO:aiogram.dispatcher:Run polling for bot @otk_vstup_bot id=8006344610 - 'OTFK ONTU Vstup'
INFO:aiogram.event:Update id=564463642 is handled. Duration 202 ms by bot id=8006344610
INFO:aiogram.event:Update id=564463643 is handled. Duration 125 ms by bot id=8006344610
```

Можливість перейти на оригінали документів



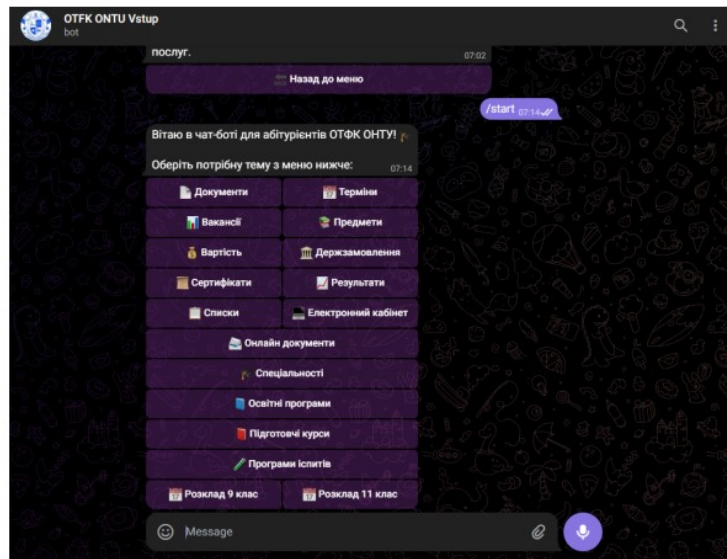
Додаток 2
до Правил прийому 2025 р.

КОНКУРСНІ ПРОПОЗИЦІЇ
освітніх студентів (освітньо-кваліфікаційні рівні) та спеціальностей, за якими оголошується прийом на навчання,
ліцензійованих об'єктів та організацій терміну навчання
Навчальний структурний підрозділ «Одеський державний фаховий коледж (Одеського національного технічного університету)»
(повна назва: Фаховий коледж) (повна назва: Фаховий коледж)

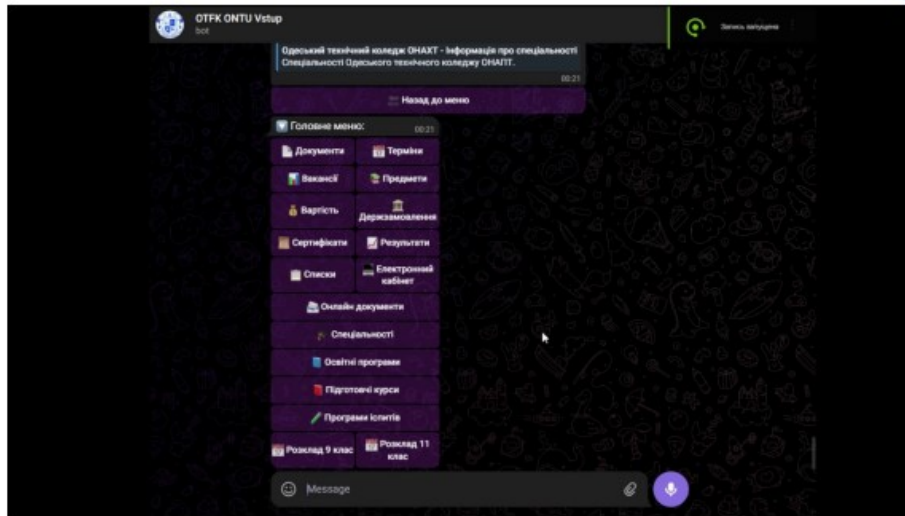
Освітній студент (освітньо-професійної рівня) – фаховий, аспірантський бакалавр

Клас	Назва	Спеціальність	Зв'язаний дипломний обов'язок	Освітньо-професійна програма (назва ліцензійованого навчального закладу)	Ліцензійовані об'єкти			Підрозділи терміну навчання			Вартість навчання, грн. *	
					всього	Ділова форма навчання	І.П.Н.	Ділова форма навчання	Зачесні форми навчання	Ділова форма навчання	Зачесні форми навчання	
ЕКОНОМІЧНІ ВІДДІЛЕННЯ												
D	Міжс. економічне та право	03	Фінанси, банківська справа, страхування та фінансовий ринок	25	25	0	9 кв. – 7:30 м 11 кв. – 7:30 м	-	-	-	20150	-
D	Міжс. економічне та право	03	Маркетинг	30	30	0	9 кв. – 7:30 м 11 кв. – 7:30 м	-	-	-	20150	-
D	Міжс. економічне та право	05	Менеджмент	30	30	0	9 кв. – 7:30 м 11 кв. – 7:30 м	-	-	-	20150	-
D	Міжс. економічне та право	09	Торговельно-сервісна	30	30	-	9 кв. – 7:30 м 11 кв. – 7:30 м	-	-	-	20150	-
D	Міжс. економічне та право	09	Торговельно-сервісна	30	30	-	9 кв. – 7:30 м 11 кв. – 7:30 м	-	-	-	20150	-
ВІДДІЛЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ												
F	Інформаційні технології	17	Комп'ютерна інженерія	120	40	40	6	9 кв. – 7:30 м 11 кв. – 7:30 м	-	-	21050	-
F	Інформаційні технології	17	Комп'ютерна інженерія	120	40	40	0	9 кв. – 7:30 м 11 кв. – 7:30 м	-	-	21050	-
F	Інформаційні технології	17	Комп'ютерна інженерія	120	40	40	0	9 кв. – 7:30 м 11 кв. – 7:30 м	-	-	21050	-
F	Інформаційні технології	17	Комп'ютерна інженерія	120	30	30	0	9 кв. – 7:30 м 11 кв. – 7:30 м	-	-	21050	-

Як виглядає головне меню



Демонстрація роботи бота



РЕЦЕНЗІЯ

на кваліфікаційну роботу здобувача (здобувачки) освіти
відділення комп'ютерних систем

Борискова Даніла Євгеновича

(прізвище, ім'я та по батькові)

Спеціальність 123 Комп'ютерна інженерія

ОПП Комп'ютерна інженерія

Керівник кваліфікаційної роботи Іванова Лілія Вікторівна

(прізвище, ім'я та по батькові)

Тема кваліфікаційної роботи «Створення інформаційного чат-боту для абітурієнтів коледжу»

Обсяг розрахунково-пояснювальної записки 54 сторінок

Обсяг графічної (презентаційної) частини 11 аркушів (слайдів)

ХАРАКТЕРИСТИКА КВАЛІФІКАЦІЙНОЇ РОБОТИ

а) заключення про ступінь відповідності виконаної кваліфікаційної роботи завданню кваліфікаційна робота у повному обсязі відповідає темі та завданню

б) характеристика виконання кожного розділу кваліфікаційної роботи

Основний розділ кваліфікаційної роботи складається з підрозділів: Чат-боти у бізнесі та в освіті. Програмне забезпечення для розробки чат-боту. Розробка інформаційного чат-боту.

в) оцінка якості виконання пояснювальної записки та графічної частини кваліфікаційної роботи

Пояснювальна записка виконана якісно, у достатньому обсязі, відповідно до індивідуального завдання та теми дипломного проекту, розділи пояснювальної записки відповідають етапам рішення завдання, поставленого у кваліфікаційній роботі.

Презентація виконана якісно, у достатньому обсязі. Презентація наочно демонструє результати роботи.

г) перелік позитивних якостей кваліфікаційної роботи _____
Автор обґрунтував вибір платформи (Telegram) на основі сучасної статистики використання месенджерів серед молоді. Бот реалізовано з використанням сучасного фреймворку Aiogram, що дозволяє забезпечити асинхронність, гнучкість і масштабованість. Чітко подано архітектуру проєкту, реалізовано розділення логіки та контенту, дотримано принципів модульності й чистого коду

д) основні недоліки кваліфікаційної роботи _____
Замало уваги приділено питанню забезпечення безпеки даних користувачів, зокрема, щодо зберігання токенів, шифрування або авторизації адміністратора. Також у роботі не згадується про можливості масштабування (наприклад, інтеграцію з базою даних або аналітику використання бота). Відсутність ергономічного дизайну. Відсутність оптимізації подання інформації

Оцінка розрахункової частини _____ *Відмінно*

Оцінка графічної частини _____ *Добре*

Загальна оцінка _____ *Добре*

Прізвище, ім'я, по батькові рецензента _____ *к.т.н. Шibaєва Наталя Олегівна*

Місце роботи і посада рецензента _____ *Національний університет «Одеська політехніка»,
доцент кафедри інформаційних технологій*

Підпис: _____

«13» _____ *2025* р.



Відокремлений структурний підрозділ
Одеський технічний фаховий коледж ОНАХТ

ВІДГУК

Керівника про кваліфікаційну роботу бакалавра

Борискова Данила Євгенович

(прізвище, ім'я та по батькові)

Освітньо-професійна програма «Комп'ютерна інженерія»

Спеціальність 123 «Комп'ютерна інженерія»

Тема кваліфікаційної роботи

«Створення інформаційного чат-боту для абітурієнтів коледжу»

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ)

а) Обсяг і якість виконання роботи (розрахунково-пояснювальної записки)

Пояснювальна записка виконана якісно, у достатньому обсязі, відповідно до індивідуального завдання та теми дипломного проекту, розділи пояснювальної записки відповідають етапам рішення завдання, поставленого у дипломному проекті

Презентація виконана якісно, у достатньому обсязі. Презентація наочно демонструє результати роботи.

б) Самостійність роботи над кваліфікаційною роботою

Студент самостійно обрала напрям та тематику кваліфікаційної роботи. Провів аналіз існуючих рішень і зробив необхідні висновки для реалізації проекту. Виявив навички самостійно опрацьовувати новий матеріал та виконувати пошук необхідної літератури та інших джерел інформації

в) Теоретична підготовка бакалавра _____

відповідає вимогам, що надаються до бакалавра зі спеціальності

«Комп'ютерна інженерія»

г) Вміння розв'язувати виробничі і конструкторські питання на базі останніх досліджень науки і техніки, передових методів виробництва _____

У роботі представлено результати розробки інформаційного Telegram-бота для абітурієнтів ВСП «Одеський технічний фаховий коледж ОНТУ», який дозволяє швидко отримувати актуальну інформацію щодо вступу, сприяє автоматизації відповідей на типові запитання та покращує комунікацію між навчальним закладом та вступниками. Вивчено закономірності використання месенджерів у молодіжному середовищі, зокрема Telegram як найбільш популярної платформи серед цільової аудиторії; досліджено існуючі рішення у сфері освітніх чат-ботів, виявлено їх функціональні можливості, переваги та недоліки

Загальна оцінка _____ 5(відмінно) _____

Прізвище, ім'я, по батькові _____ Іванова Лілія Вікторівна _____

Місто роботи і посада керівника проекту ВСП «Одеський технічний фаховий коледж ОНТУ» к.т.н., зав. кафедрою Комп'ютерної інженерії _____

Підпис _____

« 21 » 06. 2025р.

**ДОЗВІЛ
НА РОЗМІЩЕННЯ
ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
В ЕЛЕКТРОННОМУ РЕПОЗИТАРІЇ ВСП «ОТФК ОНТУ»**

Ми, що нижче підписалися,

Борисков Данііл Євгенович,
здобувач освіти гр. 2БКС-29, та

Іванова Лілія Вікторівна,
керівник дипломного проекту,

не заперечуємо щодо розміщення електронного варіанту пояснювальної записки до випускної кваліфікаційної роботи бакалавра на тему:

«Розробка інформаційного чат-боту для абітурієнтів коледжу» (автор роботи – Борисков Д.Є., керівник роботи – Іванова І.В.)

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2025 році, у повному обсязі в електронному репозитарії ВСП «ОТФК ОНТУ» для вільного доступу через мережу Інтернет.

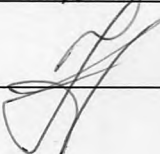
Несемо відповідальність за ідентичність електронного та друкованого варіантів випускної кваліфікаційної роботи, і даємо згоду на обробку персональних даних.

Виконавець



/ Борисков Д.Є. /

Керівник



/ Іванова Л.В. /

« _____ » _____ 20 ____ р.

ДОВІДКА

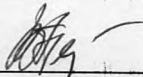
кафедри комп'ютерної інженерії
про допуск до захисту кваліфікаційної роботи
здобувача (здобувачки) освіти II курсу
відділення комп'ютерних систем групи 2БКС-29

Борискова Даніла Євгеновича

на тему Створення інформаційного чат-боту
для абітурієнтів коледжу

Висновок відповідальної особи за проведення нормоконтролю:

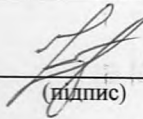
пояснювальна записка до кваліфікаційної роботи виконана з деякими
порушеннями ДСТУ та оформлена відповідно до вимог Положення про
дипломне проєктування


(підпис)

23.06.2025
(дата)

Петрашова В.І.
(П.І.Б.)

Висновок відповідальної особи за перевірку роботи на наявність академічного
плагиату згідно звіту про перевірку від 23.06.2025 р. значення коефіцієнту
подібності в роботі становить 9,38%, коефіцієнт цитування – 3,98%.


(підпис)

23.06.2025
(дата)

Краснокутська К.Г.
(П.І.Б.)

Попередня експертиза (малий захист) кваліфікаційної роботи

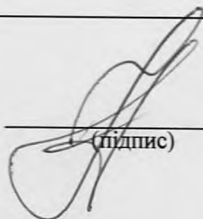
здобувача (здобувачки) освіти

Борискова Д.Є.
(П.І.Б.)

проведена « 23 » червня 2025 р.

Висновки Пояснювальна записка до кваліфікаційної роботи виконана у
повному обсязі. Випускна кваліфікаційна робота відповідає вимогам
Положення про дипломне проєктування та рекомендована до захисту.

Зав. кафедри КІ


(підпис)

Іванова Л.В.
(П.І.Б.)

Звіт подібності

метадані

Назва організації

Odesa Technical Professional College of Odesa National University of Technology

Заголовок

Створення інформаційного чат-боту для абітурієнтів коледжу

Автор

Науковий керівник / Експерт

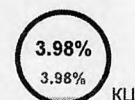
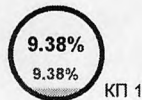
Борисков Даниїл Євгенович Іванова Лілія Вікторівна

підрозділ

Відокремлений структурний підрозділ "Одеський технічний фаховий коледж Одеського національного технологічного університету"

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



25

Довжина фрази для коефіцієнта подібності 2

10614

Кількість слів

83107

Кількість символів

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		4
Інтервали		0
Мікропробіли		0
Білі знаки		0
Парафрази (SmartMarks)		109

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз

Колір тексту

ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	https://card-file.ontu.edu.ua/bitstreams/7f031b5d-95bb-43c2-8b3f-0fa2499416c4/download	103 0.97 %
2	https://card-file.ontu.edu.ua/bitstreams/549ee9fe-7574-4ae5-b500-9fe2711f33e6/download	60 0.57 %
3	https://card-file.ontu.edu.ua/bitstreams/7f031b5d-95bb-43c2-8b3f-0fa2499416c4/download	32 0.30 %
4	https://card-file.ontu.edu.ua/bitstreams/0e6c3361-ffbf-4469-86a1-fe84a1fe21cd/download	30 0.28 %
5	https://card-file.ontu.edu.ua/bitstreams/7f031b5d-95bb-43c2-8b3f-0fa2499416c4/download	21 0.20 %

6	https://card-file.ontu.edu.ua/bitstreams/7f031b5d-95bb-43c2-8b3f-0fa2499416c4/download	21 0.20 %
7	Розроблення чат-боту для спеціалізованого фітнес-центру з медичної реабілітації 5/21/2025 Kharkiv National University of Economics named after S.Kuznets (KNUE) (KNUE)	19 0.18 %
8	https://card-file.ontu.edu.ua/bitstreams/7f031b5d-95bb-43c2-8b3f-0fa2499416c4/download	18 0.17 %
9	https://card-file.ontu.edu.ua/bitstreams/d42aac6d-ab01-4a74-b9cb-ced2a9eff719/download	17 0.16 %
10	https://card-file.ontu.edu.ua/server/api/core/bitstreams/21ac499a-a9e9-4137-810c-5f21a0318048/content	14 0.13 %

з домашньої бази даних (0.00 %)

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
------------------	-----------	--

з програми обміну базами даних (2.62 %)

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	Розроблення чат-боту для спеціалізованого фітнес-центру з медичної реабілітації 5/21/2025 Kharkiv National University of Economics named after S.Kuznets (KNUE) (KNUE)	222 (26) 2.09 %
2	bitstream_f655deb3-829a-44c1-878b-dcfbb7812d3a 12/6/2024 National Technical University "Kharkiv Polytechnic Institute" students papers (National Technical University "Kharkiv Polytechnic Institute" students papers)	14 (2) 0.13 %
3	Яковенко С. В._Готвянська А. С._1 12/10/2024 Dnipro State Agrarian and Economic University (Відділ внутрішнього аудиту і контролю якості освітньої діяльності)	11 (1) 0.10 %
4	Вінтоняк Роман_Лазарович 6/3/2025 Vasyl Stefanyk Precarpathian National University (VSPNU) (VSPNU)	10 (1) 0.09 %
5	IST_2024_b_05 8/20/2024 O.M.Beketov National University of Urban Economy in Kharkiv (O.M.Beketov National University of Urban Economy in Kharkiv)	9 (1) 0.08 %
6	Кваліфікаційна робота_бакалавра_Юр'єв_Д_Ю_ІТ_92_в_.docx 6/12/2023 Sumy State University (Кафедра інформаційних технологій)	7 (1) 0.07 %
7	2022_61260002_Chabak_Oleh_Valentynovych_98028 10/26/2024 National University "Lviv Politechnika" (National University Lviv Politechnika)	5 (1) 0.05 %

з Інтернету (6.76 %)

ПОРЯДКОВИЙ НОМЕР	ДЖЕРЕЛО URL	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	https://duikt.edu.ua/repozitorii/ipz/2024/%D0%9F%D0%94-41/%D0%9F%D0%94-41%20%D0%A2%D0%B8%D1%89%D0%B5%D0%BD%D0%BA%D0%BE%20%D0%90.%D0%92..pdf	286 (32) 2.69 %
2	https://card-file.ontu.edu.ua/bitstreams/7f031b5d-95bb-43c2-8b3f-0fa2499416c4/download	214 (8) 2.02 %
3	https://card-file.ontu.edu.ua/bitstreams/549ee9fe-7574-4ae5-b500-9fe2711f33e6/download	60 (1) 0.57 %
4	https://card-file.ontu.edu.ua/bitstreams/0e6c3361-ffbf-4469-86a1-fe84a1fe21cd/download	38 (2) 0.36 %

