

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА  
ДИЗАЙНУ

Факультет мехатроніки та комп'ютерних технологій  
Кафедра комп'ютерних наук та технологій

**ДИПЛОМНА БАКАЛАВРСЬКА РОБОТА**

на тему

Розробка програмного забезпечення в задачах комунікації освітнього процесу

Виконав: студент групи БІТск-19  
спеціальності  
122 Комп'ютерні науки

Варгач Олег Станіславович

Науковий керівник Демківська Т. І.

Рецензент д.т.н., проф. Чупринка В.І.

Київ 2021

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ  
ТА ДИЗАЙНУ**

Факультет            мехатроніки та комп'ютерних технологій  
Кафедра             комп'ютерних наук та технологій  
Спеціальність      122 Комп'ютерні науки  
Освітня програма   Комп'ютерні науки

**ЗАТВЕРДЖУЮ**

Завідувач кафедри  
комп'ютерних наук та технологій  
\_\_\_\_\_ В.Ю.Щербань  
(підпис)  
« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_ р.

**ЗАВДАННЯ**

**НА ДИПЛОМНУ БАКАЛАВРСЬКУ РОБОТУ**

студенту  
Варгачу Олегу Станіславовичу

Тема роботи        Розробка програмного забезпечення в задачах комунікації  
освітнього процесу

- Науковий керівник роботи Демківська Тетяна Іванівна  
затверджені наказом КНУТД від “15” березня 2021 року № 75-уч
2. Строк подання студентом дипломної роботи 10 .06.2021 р.
  3. Вихідні дані до дипломної роботи (проекту) Розробки кафедри комп'ютерних наук та технологій, рекомендована література, додатки
  4. Зміст дипломної бакалаврської роботи: 1) Аналіз предметної області, 2) Визначення призначення та функціоналу програмної реалізації; дослідження технологій та засобів розробки, 3) Розробка програмної реалізації та її тестування
  5. Дата видачі завдання 15.03.2021 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної бакалаврської роботи	Терміни виконання етапів	Примітка про виконання
1	Вступ	03.05.2021	
2	Розділ 1 Аналіз предметної області	08.05.2021	
3	Розділ 2 Проектування програмної реалізації	15.05.2021	
4	Розділ 3 Розробка чат бота	28.05.2021	
5	Висновки	30.05.2021	
6	Оформлення дипломної бакалаврської роботи	01.06.2021	
7	Здача дипломної бакалаврської роботи на кафедрі для рецензування (за 14 днів до захисту)	02.06.2021	
8	Перевірка дипломної бакалаврської роботи на наявність ознак плагіату (за 10 днів до захисту)	06.06.2021	
9	Подання дипломної бакалаврської роботи на затвердження завідувачу кафедри (за 7 днів до захисту)	10.06.2021	

Студент

\_\_\_\_\_

( підпис )

О.С. Варгач

Науковий керівник  
роботи

\_\_\_\_\_

( підпис )

Т.І. Демківська

Рецензент

\_\_\_\_\_

( підпис )

В.І. Чупринка

## АНОТАЦІЯ

**Варгач Олег Станіславович. Тема Розробка програмного забезпечення в задачах комунікації освітнього процесу**

Пояснювальна записка до дипломної бакалаврської роботи складається з трьох розділів та містить 3 таблиці, 2 додатки, 31 рисунок та 11 джерел.

Об'єкт дослідження: чат-бот, як вирішення задач комунікації в освітньому процесі.

Мета дипломного проекту: розроблення чат-бота в месенджері Telegram, для вирішення задач комунікації під час освітнього процесу, дослідження методів взаємодії з даними, для швидкого і комфортного перегляду їх користувачем, отримання зворотного зв'язку від користувача та збереження отриманих даних в базу даних, інтуїтивно зрозумілий інтерфейс.

Методи розробки: мова програмування Python, інтегроване середовище розробки PyCharm, HTTP-інтерфейс TelegramBotAPI.

У першому розділі був проведений аналіз предметної області, огляд існуючих аналогів.

У другому розділі визначено призначення та функціонал програмної реалізації; дослідження технологій та засобів розробки.

У третьому розділі описана розробка програмної реалізації та її тестування.

У додатках наведено: текст програми та її опис.

**КЛЮЧОВІ СЛОВА: МЕСЕНДЖЕР, ЧАТ-БОТ, ОСВІТНІЙ ПРОЦЕС, ЗВОТНИЙ ЗВ'ЯЗОК, БАЗА ДАНИХ.**

## ANNOTATION

### **Varhach Oleh Stanislavovich. Topic Software development in the problems of communication of the educational process**

The explanatory note to the bachelor's thesis consists of three sections and contains 3 tables, 2 appendices, 31 figures and 11 sources.

Object of research: chatbot, as a solution to communication problems in the educational process.

The purpose of the diploma project: development of a chatbot in the Telegram messenger, for solving communication problems during the educational process, research of methods of interaction with data, for fast and comfortable viewing by the user, receiving feedback from the user and saving the received data in a database, intuitive interface.

Development methods: Python programming language, PyCharm integrated development environment, TelegramBotAPI HTTP interface.

In the first section the analysis of the subject area, the review of existing analogues was carried out.

The second section defines the purpose and functionality of the software implementation; research of technologies and means of development.

The third section describes the development of software implementation and its testing.

The appendices contain: the text of the program and its description.

**KEY WORDS: MESSAGE, CHAT BOT, EDUCATIONAL PROCESS, FEEDBACK, DATABASE.**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ

Факультет мехатроніки та комп'ютерних технологій

Кафедра комп'ютерних наук та технологій

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

дипломної бакалаврської роботи

на тему

Розробка програмного забезпечення в задачах комунікації освітнього процесу

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

БД	База даних
URL ( <i>Uniform Resource Locator</i> )	Визначник місцезнаходження сайту в мережі Інтернет
IDE ( <i>Integrated development environment</i> )	Інтегроване середовище розробки
API ( <i>Application programming interface</i> )	Програмний інтерфейс
MTProto ( <i>Mobile Telegram Protocol</i> )	Криптографічний протокол, який використовується у Telegram для шифрування переписки користувачів
OS ( <i>Operating System</i> )	Операційна система
HTTP ( <i>HyperText Transfer Protocol</i> )	Протокол передачі даних, що використовується в комп'ютерних мережах
SQL ( <i>Structured Query Language</i> )	Мова структурованих запитів

## ЗМІСТ

ВСТУП .....	
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	
1.1. Огляд месенджера Telegram.....	
1.2. Загальні відомості про чат-боти.....	
1.3. Огляд існуючих аналогів.....	
1.3.1 RailwayBot.....	
1.3.2 NovaPoshtaBot.....	
1.3.3. Просто погода.....	
РОЗДІЛ 2. ПРОЕКТУВАННЯ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ .....	
2.1 Призначення чат-бота.....	
2.2 Опис вимог до чат-боту.....	
2.3 Проектування функціональних можливостей чат-бота.....	
2.4 Вибір засобів розробки.....	
2.4.1 Python.....	
2.4.2 Telegram Bot API.....	
2.5 Вибір середовища розробки.....	
2.6 База даних SQLite.....	
РОЗДІЛ 3. РОЗРОБКА ЧАТ-БОТА .....	
3.1. Реєстрація чат-бота для Telegram Bot API.....	
3.2. Реалізація програмної частини.....	
3.2.1. Створення перших повідомлень.....	
3.2.2. Розробка головного меню та його складових.....	
3.2.3. Підключення бази даних.....	
ВИСНОВКИ.....	
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	
ДОДАТОК А. ТЕКСТ ПРОГРАМИ.....	
ДОДАТОК Б. ІНТЕРФЕЙС ЧАТ-БОТА.....	



## ВСТУП

Сучасний освітній процес відрізняється своєю технологічністю. Все більше різних сучасних технологічних рішень дозволяють полегшити процес комунікації та отримання інформації. Впровадження нових інформаційних джерел в освітню галузь є невід'ємною частиною сучасного прогресу.

І в наш час коли розвиток Інтернету дійшов до того, що доступ до нього можна здійснити з будь-якого мобільного пристрою, а соціальні мережі та месенджери набули великої популярності, що робить їх чудовим джерелом швидкого і простого отримання та обміну інформації. Люди обирають, доступність, простоту та зручність використання, а саме ці ознаки вигідно відрізняють месенджер Telegram від всіх інших, для використання його можливостей у якості інструменту комунікації під час освітнього процесу.

Месенджер Telegram є платформою для такої програми як чат-бот. Даний вид програми є затребуваним в наш час, адже чат-бот є дуже простим і зручним у використанні, а його можливості його обмежені тільки задумами розробника. Багато людей вже користуються ними для вирішення різних задач, починаючи від пересилання повідомлень до оплати товарів в інтернет магазинах, адже поріг входження в користування чат-ботами досить низький. Взаємодія ж чат-ботом виглядає, як звичайна переписка з людиною. Все це робить Telegram чат-бота чудовим інструментом для вирішення задач комунікації в освітньому процесі.

Метою дипломного бакалаврського проекту розроблення чат-бота в месенджері Telegram, для вирішення задач комунікації під час освітнього процесу.

Чат бот є простим і постійним помічником, для вирішення поставлених перед ним задач. В освітньому процесі він дозволить спростити пошук необхідної інформації користувачем, а також автоматизує збір і збереження вхідної інформації.

Виходячи з поставленої мети необхідно виконати наступні завдання:

- виконати аналіз предметної області;
- виконати огляд існуючих аналогів;
- виконати вибір середовища та технологій розробки;

- створити чат-бота в месенджері Telegram та провести його тестування.

Дана програмна реалізація має перспективу користуватись попитом через свою зручність, доступність та простоту використання для звичайного користувача.

У майбутній перспективі можливе вдосконалення та покращення створеного програмного продукту.

## РОЗДІЛ 1

### АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

#### 1.1. Огляд месенджера Telegram

Telegram - це міжплатформний додаток, розроблений з використанням мови програмування C ++, що дозволяє обмінюватися повідомленнями та файлами у більшості форматів. Він використовує спеціально розроблену серверну частину із закритим кодом і працює на серверах у Німеччині та США. Telegram вирізняється серед своїх конкурентів такими перевагами:

- Конфіденційність – усі чати зашифровані, і повідомлення буде видалено через деякий час. Існує також поняття "секретний чат", повідомлення якого взагалі не зберігаються на сервері;
- Швидкість – швидкість передачі інформації набагато швидша, ніж аналогічні;
- Надійність – сервери Telegram розташовані по всьому світу, що значно покращує гнучкість;
- Відкритість – використовуйте відкритий протокол MTProto та API, безкоштовно для всіх;
- Відсутність передплати та реклами;
- Обмеження розміру повідомлень та вкладень не існує.

Функцією месенджера є секретний чат та спеціально розроблений протокол шифрування MTProto (рис. 1.1). Секретний чат корисний людям, які потребують найбезпечнішого спілкування.

На додаток до використання основного протоколу, усі повідомлення шифруються ключем за принципом "пристрій до пристрою". Ніхто, включаючи працівників Telegram, не може перехопити та розшифрувати повідомлення. Повідомлення з таких чатів не зберігаються на сервері програми, тому їх не можна пересилати. Однак завжди існує ризик витоку абсолютно секретної інформації через скріншоти інтерфейсу Telegram з відкритими секретними функціями чату.

Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Варгач О.С.			Розробка програмного забезпечення в задачах комунікації освітнього процесу	Літ.	Арк.	Аркушів
Перевір.		Демківська Т.І.						
Реценз.		Чупринка В.І.						
Н. Контр.		Щербань В.Ю.				<b>КНУТД. ФМКТ</b>		
Затверд.		Щербань В.Ю.						
					РОЗДІЛ 1.АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ			

## MTPROTO 2.0, part I

Cloud chats (server-client encryption)

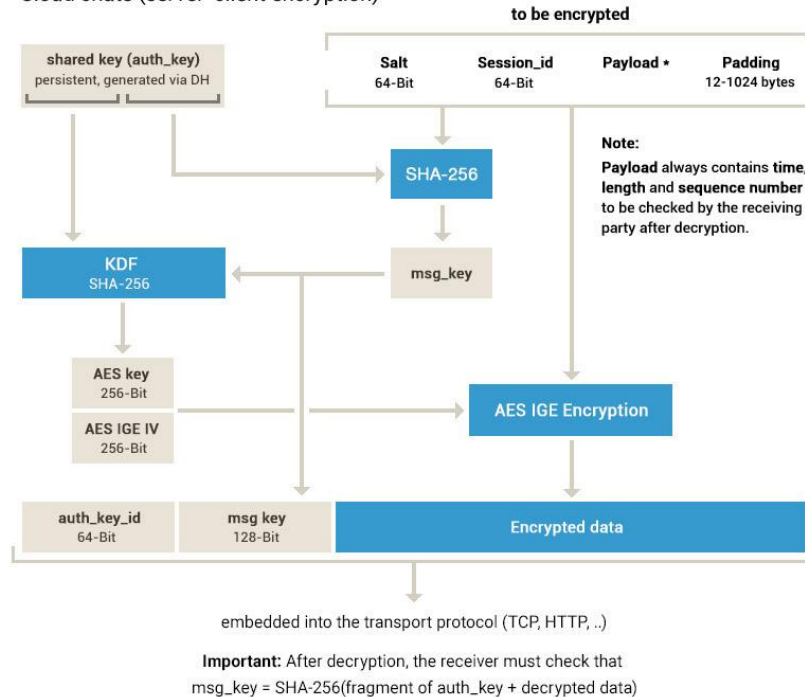
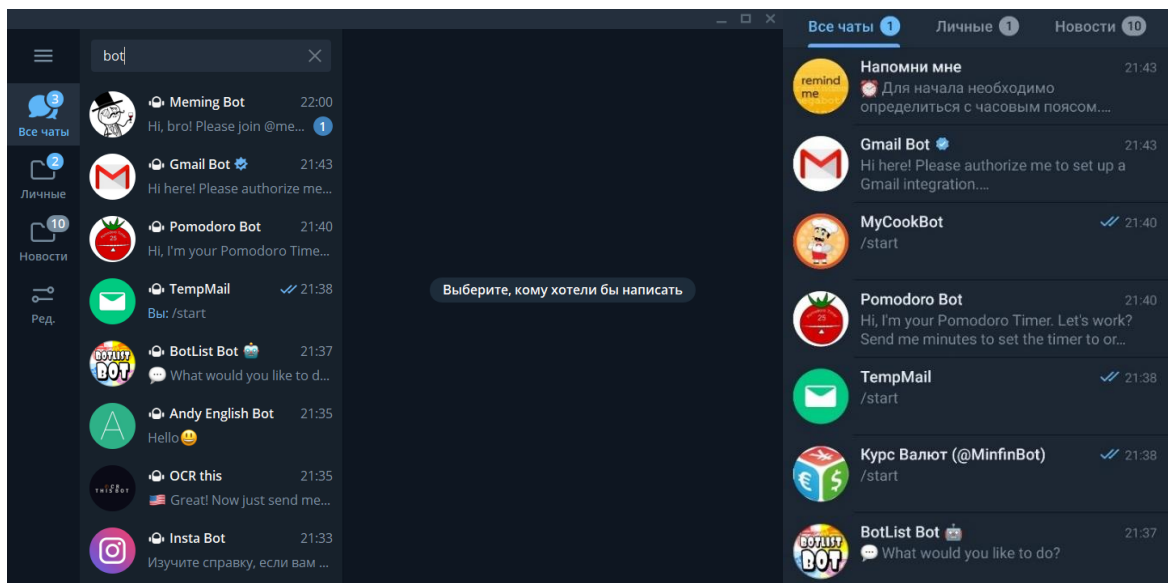


Рис.1.1. Схема роботи протоколу MTProto

Додаток можна використовувати на всіх популярних платформах, таких як OS Android, iOS, Windows, Linux тощо (рис.1.2). Завдяки своїй широкій популярності, Telegram має у своєму складі близько 18 мов, включаючи українську.



а)

б)

Рис 1.2. Інтерфейс Telegram на: а) OS Windows; б) OS Android

						Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

## 1.2. Загальні відомості про чат-боти

Останнім часом месенджери та соціальні мережі все частіше використовують такі послуги, як чат-боти. Перевіряючи інформацію про них з різних джерел, можна сказати, що чат-боти тепер стали незамінною частиною повсякденного життя – від спілкування до розваг, від консультацій лікарів до замовлення товарів та послуг.

У будь-якому випадку, незалежно від платформи, чат-бот – це програма, яка отримує інформацію від користувачів для формування правильних і логічних відповідей. Згідно з опитуванням стану чат-ботів, вони мають багато переваг:

- 24/7 підтримка клієнтів;
- надзвичайна швидкість відгуку і простота;
- ефективна взаємодія з клієнтами;
- отримання відповіді на прості запитання;
- уніфікований, зрозумілий інтерфейс;
- можливість одночасно працювати з великою кількістю клієнтів.

За сферою застосування чат-боти поділяються на:

- р2р – для особистого спілкування;
- b2c – підтримка на корпоративному сайті та в мобільних додатках клієнтів компанії.

Також чат-боти застосовуються в різних сферах життя:

- Бізнес. Чат-боти в цій сфері мають найбільший функціонал і вирішують велику кількість завдань, такі як: продажі, онлайн консультування, рекламні комунікації.
- Отримання інформації. Такі чат-боти використовуються для надання інформації в будь-якій сфері, будь-то новини чи відомості про якийсь заклад.
- Навчання. Такі чат-боти надають необхідні навчальні матеріали, невеликі за обсягом тексти з необхідної дисципліни, оцінювання знань тестуванням.

Тому можна сказати, що роль чат-ботів зростає з кожним днем. Це тому, що люди проводять більшу частину часу на своїх смартфонах, перевіряючи електронні листи та повідомлення.

									Арк.
Змн.	Арк.	№ докум.	Підпис	Дата					

Бот - це програма, яка виконує різні завдання для користувачів соціальних мереж або месенджерів. Завжди є прості речі, на які ви не хочете витратити час. Тут завжди можуть прийти на допомогу роботи. Це схоже на звичайний чат, але спілкування відбувається не з людьми, а з програмою. Якщо це робот для пошуку квитків, він може забронювати квитки на поїзд. Не потрібно вручну вводити дані - зазвичай розмова з роботом складається з попередньо зарезервованих команд, схожих на кнопки (рис. 1.3).

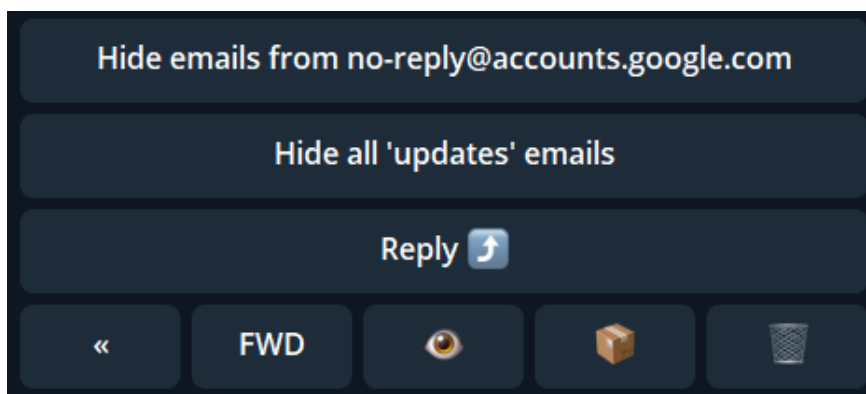


Рис. 1.3. Вид кнопок для керування чат-ботом.

Найбільша перевага чат-ботів полягає в тому, що ними дуже легко користуватися. Вони створені на основі соціальних мереж та месенджерів, а це означає, що вимоги чат-ботів не перевищать вимог передбачуваної ними платформи. Найголовніше - це підключення до Інтернету та справний пристрій. Навіть якщо ви не можете отримати доступ до Інтернету, але листування з ботом не видаляється, користувач все одно може переглянути бота та отримати необхідну інформацію (якщо кеш програми збережено).

Щоб знайти відповідний чат-бот для Telegram, зазвичай використовують такі методи:

- Інтернет-пошук;
- Пошук в Telegram по псевдоніму чат-бота.

						Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

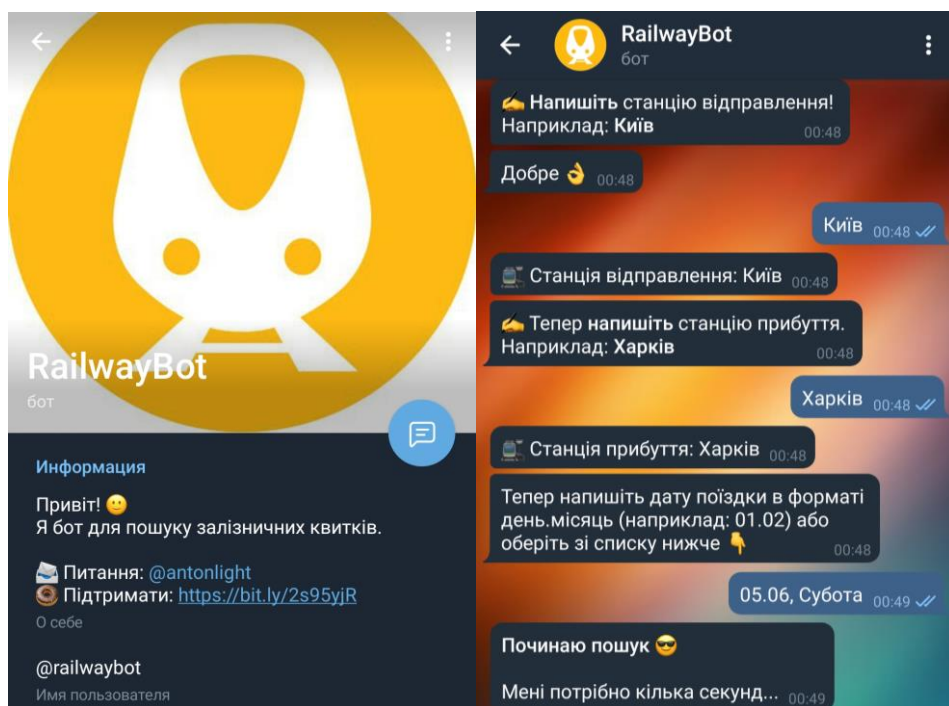
### 1.3. Огляд існуючих аналогів

Перед розробкою програмної реалізації, необхідно виконати огляд існуючих чат-ботів для уникнення помилок вже створених ботів та зрозуміти загальну концепцію їх використання та створити чат-бот з оригінальним набором функцій.

Під час пошуку чат-ботів не було знайдено повних аналогів функціоналу яких збігався б з метою даної дипломної бакалаврської роботи. Оскільки розроблювальний чат-бот в більшій мірі буде інформаційним, тому і розглянемо приклади інформаційних чат-ботів.

#### 1.3.1. RailwayBot

Чат-бот, який використовується для моніторингу та пошуку квитків на поїзди в Україні (рис. 1.4), є хорошим рішенням для тих, хто не встигає вчасно придбати проїзні квитки. Він швидко реагує на обрані користувачем запити: від місця до часу відправлення. Якщо квитків на певну дату немає, помічник надає можливість створити відстежування, щоб повідомити користувача про їх наявність.



а)

б)

Рис.1.4. Опис RailwayBot: а) інформація про RailwayBot; б) функціонал чат-бота RailwayBot.

						Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

### 1.3.2. NovaPoshtaBot

Це неофіційний чат-бот "Нової пошти", який відстежує відвантаження, знаходить найближче відділення та повідомляє про зміни статусу відправлення (рис. 1.5). Якщо ви поділитесь номером телефону з роботом, він зможе надати більш детальний опис вантажу. Під час вдосконалення цього помічника єдине, на що потрібно звернути увагу, це захист персональних даних користувачів.

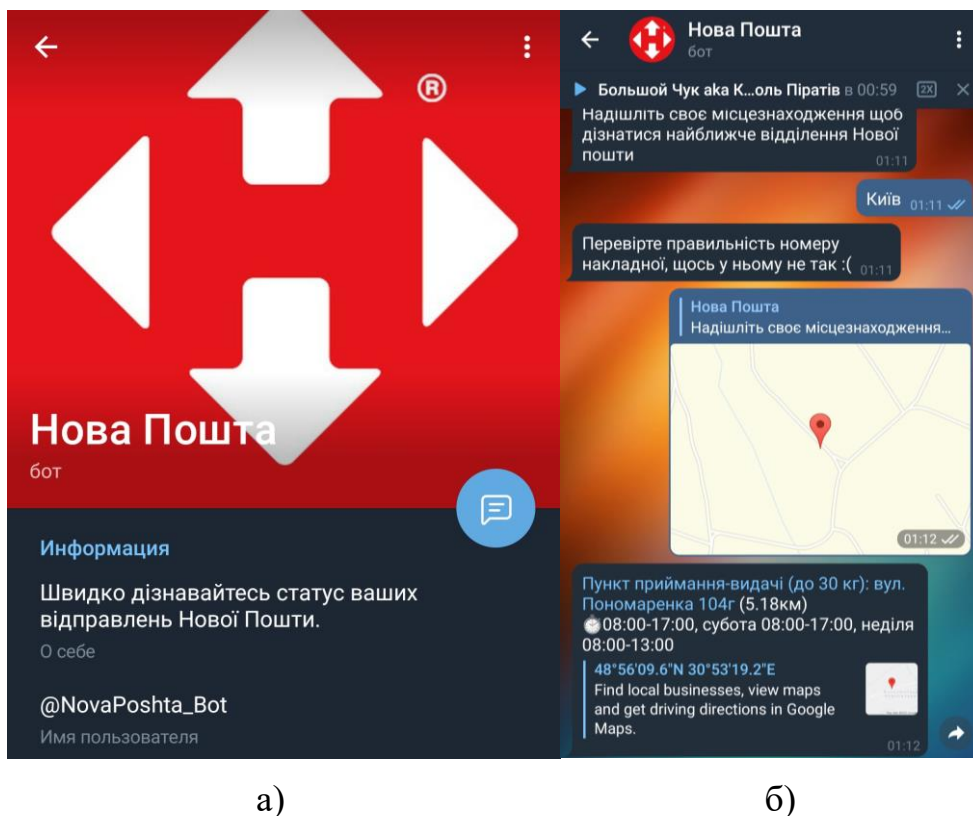


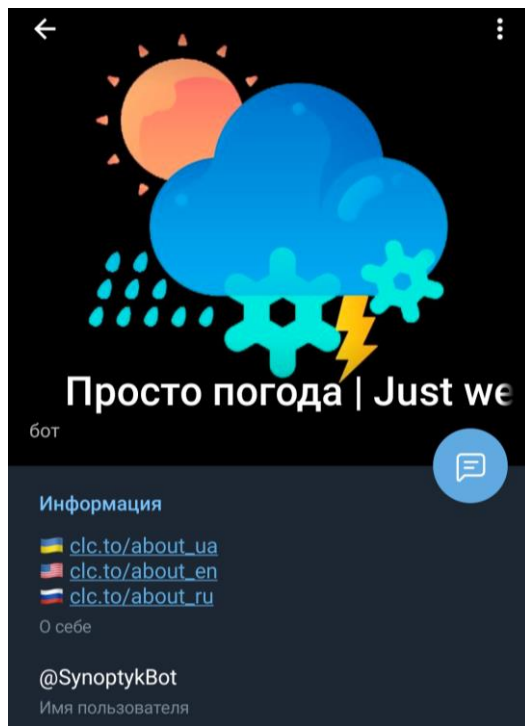
Рис. 1.5. Опис NovaPoshtaBot: а) інформація про NovaPoshtaBot; б) функціонал чат-бота NovaPoshtaBot.

### 1.3.3. Просто погода

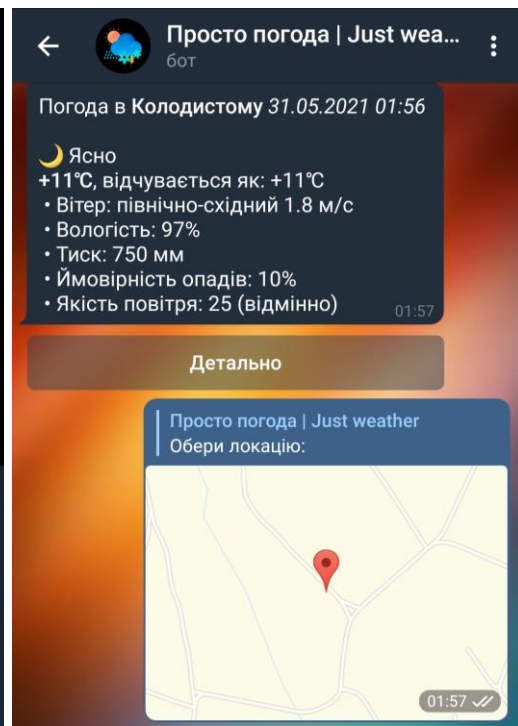
Чат-бот для відстежування поточного прогнозу погоди та на 7 днів та з автоматичним сповіщенням про зміну погоди. Можна додати до п'яти відстежуваних геолокацій та налаштувати сповіщення для кожної з них (рис. 1.6).

						Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		





а)



б)

Рис. 1.6. Опис Просто погода: а) інформація про Просто погода; б) функціонал чат-бота Просто погода.

						Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

## РОЗДІЛ 2

### ПРОЕКТУВАННЯ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

#### 2.1. Призначення чат-бота

Умовним замовником чат-бота буде Київський національний університет технологій та дизайну. Університет має сайт де можна знайти всю необхідну інформацію та соціальні мережі для розміщення новин, але для швидкого і більш зручного способу пошуку користувачем інформації та взаємодії з нею в освітньому процесі краще підійде чат-бот.

Тому, під час розробки необхідно, щоб майбутній чат-бот університету вирішив наступні задачі:

- забезпечити студентів та майбутніх вступників інформацією про університет (опис університету, навчання, інформація для абітурієнтів, наукова робота університету);
- можливість перегляду на карті розташування університету та гуртожитків;
- забезпечення зворотного зв'язку;
- можливість записатися через бот на різні заходи;
- можливість розсилки через бот інформації та матеріалів;
- можливість проведення опитування серед користувачів.
- 

#### 2.2. Опис вимог до чат-боту

Перш за все, чат-бот повинен мати можливість працювати на всіх версіях Telegram та всіх пристроях, які його підтримують; Повністю відповідати поставленим вимогам, тобто виконувати всі вищезазначені функції. Чат-бот повинен мати зручний інтерфейс, мінімум тексту та найкориснішу інформацію. Користувачі повинні швидко зрозуміти, як користуватися ботом.

Розроблений чат-бот повинен забезпечувати надійність, безпеку та цілісність даних, що вводяться користувачем.

Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Варгач О.С.			Розробка програмного забезпечення в задачах комунікації освітнього процесу	Літ.	Арк.	Аркуші
Перевір.		Демківська Т.І.						
Реценз.		Чупринка В.І.						
Н. Контр.		Щербань В.Ю.						
Затверд.		Щербань В.Ю.						
<b>РОЗДІ 2. ПРОЕКТУВАННЯ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ</b>						<b>КНУТД, ФМКТ</b>		

Інформація, надана асистентом, повинна бути точною та відповідною.

Якщо чат-бот допустить помилку або будь-яку небажану поведінку, користувач повинен мати можливість зв'язатися з адміністратором. Будь-який пристрій, на якому встановлена програма Telegram і може отримати доступ до Інтернету, є єдиною вимогою, яка повинна бути виконана для використання чат-бота.

### **2.3. Проектування функціональних можливостей чат-бота**

Розроблюваний в даній бакалаврській дипломній роботі чат-бот, має наступні функціональні можливості:

- взаємодія з чат-ботом за допомогою кнопок та навігація за допомогою головного меню;
- відображення інформації необхідної для користувача при виконанні запитів;
- інтуїтивно зрозумілий та приємний інтерфейс;
- робота на всіх пристроях, що мають доступ до Інтернету та підтримують месенджер Telegram;
- швидке надання відповідей на запити користувача.

Робота з чат-ботом починається з командою /start і перше, що побачить користувач це опис і перелік можливостей чат-бота, потім стає доступним весь функціонал.

Кожний розділ з меню відповідає за різні функції, розглянемо їх:

- «Довідка»: даний пункт відображає інформацію з описом і переліком розділів головного меню для кращого орієнтування під час роботи з чат-ботом.
- «Головне меню»: включає в себе перелік підрозділів:
  - «Університет»: надає інформацію на вибір про Київський національний університет технологій та дизайну, Профком, видавничий відділ, науково-технічну бібліотеку, доступ до публічної інформації, платні послуги, контакти та розміщення.

									Арк.
Змн.	Арк.	№ докум.	Підпис	Дата					

- «Студенту»: в цьому розділі можна знайти інформацію, корисну вже для студентів університету, їхні права та обов'язки, профспілкова організація, гуртожитки, студентський будинок моди, центр праці та кар'єри, студентський парламент, оплата послуг та інші.
  - «Вступнику»: цей розділ надає всю необхідну інформацію для майбутніх абітурієнтів Київського національного університету технологій та дизайну, правила прийому, перелік спеціальностей, умови прийому, приймальну комісію, програми фахових вступних іспитів та інші.
  - «Наука та інновації»: в цьому розділі користувач знайде інформацію про наукову роботу в університеті, наукова та інноваційна діяльність, науково дослідна частина, інституційний репозиторій та інші.
  - «Корисні посилання»: включає в себе список посилань на сайт університету та сервіси, які необхідні під час навчання, а також посилання на соціальні мережі університету.
- «Зв'язатись з нами»: через цей пункт користувач зможе зв'язатись з відповідальним за чат бот і задати своє питання.
  - «Ми на карті»: цей пункт допоможе користувачеві подивитись розміщення університету та гуртожитків не виходячи з додатку та прокласти маршрут до них.

Адміністратори чат-боту мають доступ по команді /adm до режиму адміністрування чат-ботом. Цей режим включає в себе такі пункти:

- «Захід»: за допомогою цього пункту, адміністратор може створити розсилку всім користувачам чат-боту з назвою заходу, який буде відбуватись в університеті. Після чого в чат-боті з'явиться повідомлення з описом заходу та кнопкою «Брати участь», за допомогою якої користувач зможе зареєструватись на нього. Для цього користувачеві необхідно буде ввести своє ім'я, прізвище, номер групи, якщо вже навчається, номер телефону та примітку за бажанням.
- «Розсилка»: дозволяє створити розсилку інформації для користувачів чат-боту з прикріпленням документу чи фото.

										Арк.
Змн.	Арк.	№ докум.	Підпис	Дата						

- «Опитування»: створює опитування серед користувачів бота. Адміністратор вказує суть опитування, а також перелік відповідей. Після вибору відповіді вона записується в базу даних.
- «Підрахувати відповіді»: дозволяє адміністратору підрахувати кількість відповідей конкретного опитування. Необхідно вказати назву опитування де ви збираєтесь підрахувати відповідь, а потім назву самого варіанту після чого ви отримаєте кількість відповідей за цей варіант.

Закінчити роботу з чат-ботом можливо за допомогою вбудованих кнопок, а повністю припинити його роботу, видаливши його можна за допомогою вбудованих функцій месенджера Telegram.

На рис. 2.1 зображена діаграма варіантів використання, на якій можна побачити відносини між користувачем, чат ботом, адміністратором та БД та варіанти взаємодії між ними.

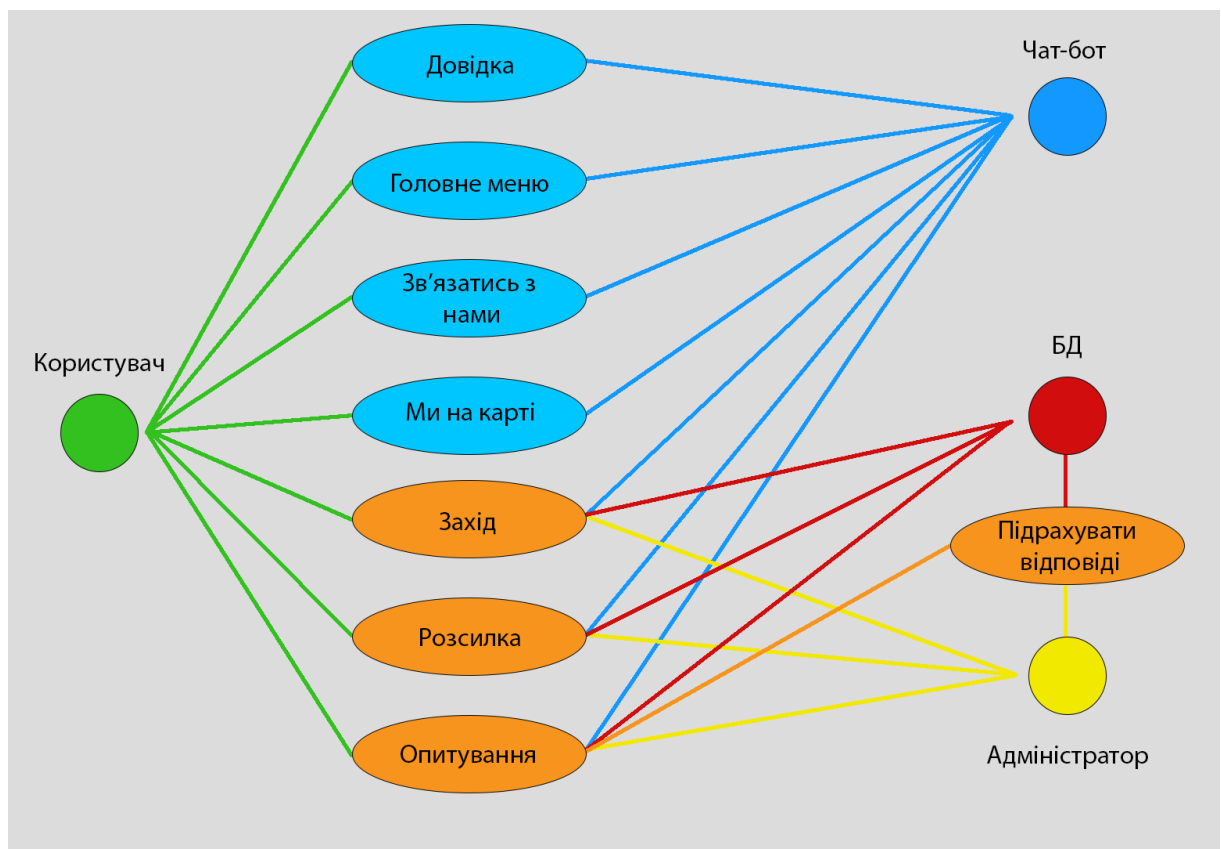


Рис. 2.1. Діаграма варіантів використання

						Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

## 2.4 Вибір засобів розробки

### 2.4.1 Python

Одним з найважливіших завдань під час створення дипломного проекту є правильний вибір засобів розробки. В результаті аналізу мов програмування, які зазвичай використовуються для створення чат-ботів, визначено, що Python є найбільш вдалим варіантом для даного проекту.

Python – інтерпретована, високорівнева, інтерактивна, об'єктно-орієнтована скриптова мова програмування (рис.2.2). Вона розроблена як мова, що «легко читається»; часто використовує в якості ключових слів слова англійської мови.

Вихідний код на Python НЕ компілюється в машинний код, а виконується безпосередньо за допомогою спеціальної програми-інтерпретатора. Користувач може писати код прямо в оболонці інтерпретатора і вводити нові команди у міру того, як виконуються попередні. Інтерпретатор починає свою роботу у верхній частині файлу, перекладає перший рядок на машинну мову, а потім виконує його. Цей процес повторюється до кінця файлу. На рисунку 2.3 зображено як працює інтерпретатор.



Рис. 2.2. Логотип мови програмування Python

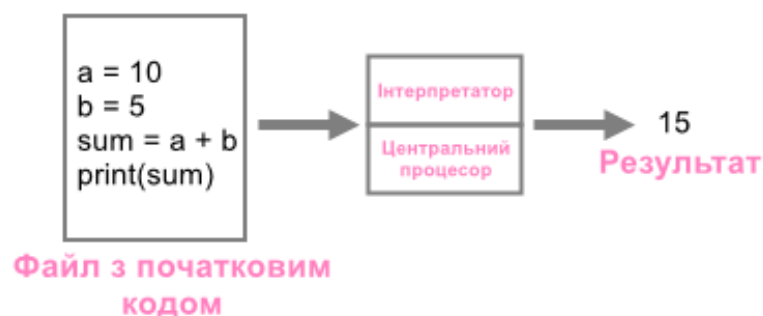


Рис.2.3. Виконання програми інтерпретатором

						Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

Розробка мови Python припадає на кінець 80-их років співробітником Голландського Національного Дослідницького Інституту Математики та Інформатики Гвідо ван Россумом.

Мова програмування Python увібрала у себе риси багатьох популярних в той час мов: Algol-68, C, C ++, Modula-3 ABC, SmallTalk, і це далеко не повний перелік. Python підтримує принципи ООП, а саме інкапсуляцію коду в особливі структури, іменовані об'єктами.

Версія 1.0 з'явилася в 1994 році, 2.0 в 2000-му, а 3.0 в 2008-му році. Підтримка Python'a здійснюється командою розробників все того ж інституту, при цьому за ван Россумом залишилося право вирішального голосу в питаннях розвитку мови. При розробці дипломного проекту використовувалася найсучасніша версія.

#### Особливості мови Python:

- *Легка для вивчення:* у мови мало ключових слів, проста структура і чітко визначений синтаксис. Завдяки цьому навчитися основам Python'a можливо за досить короткий час.
- *Легко читається:* блоки коду в Python виділяються за допомогою відступів, що спільно із ключовими словами, взятими з англійської мови, значно полегшують читання коду.
- *Легка в обслуговуванні:* однією з причин широкої популярності Python'a є простота обслуговування коду, написаного на цій мові.
- *Широка стандартна бібліотека:* наявність широкої крос-платформної бібліотеки є ще однією сильною стороною цієї мови програмування.
- *Наявність інтерактивного режиму:* дозволяє швидко тестувати потрібні ділянки коду.
- *Портативність:* Python без проблем запускається на різних платформах, при цьому зберігає однаковий інтерфейс, незалежно від того, на якому комп'ютері працює користувач.
- *Можливість розширення:* за необхідності в Python можна впроваджувати низькорівневі модулі, написані на інших мовах програмування, для найбільш гнучкого вирішення поставлених завдань.

						Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

- *Робота з базами даних:* в стандартній бібліотеці Python можна знайти модулі для роботи з більшістю комерційних баз даних.
- *Створення GUI (Графічного інтерфейсу користувача):* на Python можливе створення GUI додатків, які працюватимуть незалежно від типу операційної системи.

## 2.4.2 Telegram Bot API

Telegram Bot API - це HTTP-інтерфейс для роботи з ботами у Telegram. По суті бот – це спеціальний обліковий запис, створений для автоматичної відправки та обробки повідомлень.

Для роботи з Telegram Bot API вивчено документацію, в якій описані всі методи і параметри, що передаються. Існує два протилежних за логікою способи отримання оновлень від бота:

- `getUpdates`(використовуючи `long polling`) використовується для отримання оновлень через `long polling`. Логіка полягає у періодичному опитуванні серверів Telegram на предмет наявності нової інформації. З'єднання відкривається на нетривалий час і всі оновлення одразу відправляються боту. Відповідь повертається у вигляді масиву об'єктів Update. Параметри методу наведені у таблиці 2.1.

Табл. 2.1.

Параметри методу `getUpdates`

Параметри	Тип	Обов'язковий	Опис
<code>offset</code>	Integer	Ні	Дозволяє переглядати нові повідомлення, ігноруючи старі.
<code>limit</code>	Integer	Ні	Обмежує кількість елементів масиву Update. Приймає значення від 1 до 100.
<code>timeout</code>	Integer	Ні	Таймаут в секундах для <code>long polling</code> . За замовчуванням 0, тобто короткий запит.

									Арк.
Змн.	Арк.	№ докум.	Підпис	Дата					



- setWebhook необхідний для задання URL вебхука, на який бот відправлятиме оновлення. Мається на увазі, що тепер якщо в чат приходить повідомлення, то
- Telegram сам говорить про це. Кожний раз при отриманні оновлення на цю адресу відправлятиметься HTTPS POST із JSON-об'єктом Update. При невдалому запиті до сервера спроба повториться декілька разів. Для більшої безпеки рекомендується включити токен в URL вебхука, наприклад, наступним чином: `https://yourwebhookserver.com/ <token>`. Для того щоб отримати токен (власний ключ авторизації) необхідно написати спеціальному боту @BotFather. Оскільки ніхто сторонній не знає особистий токен користувача, запити до вебхуку посилає саме Telegram. Параметри методу наведені в таблиці 2.2.

Табл. 2.2.

Параметри методу setWebhook

Параметри	Тип	Обов'язковий	Опис
url	String	Ні	HTTP url для відправки запитів. Щоб видалили вебхук, треба відправити порожній рядок.
certificate	InputFile	Ні	Завантаження публічного ключа для перевірки кореневого сертифікату.

Принцип взаємодії чат-бота та користувача зображено на рис. 2.4.

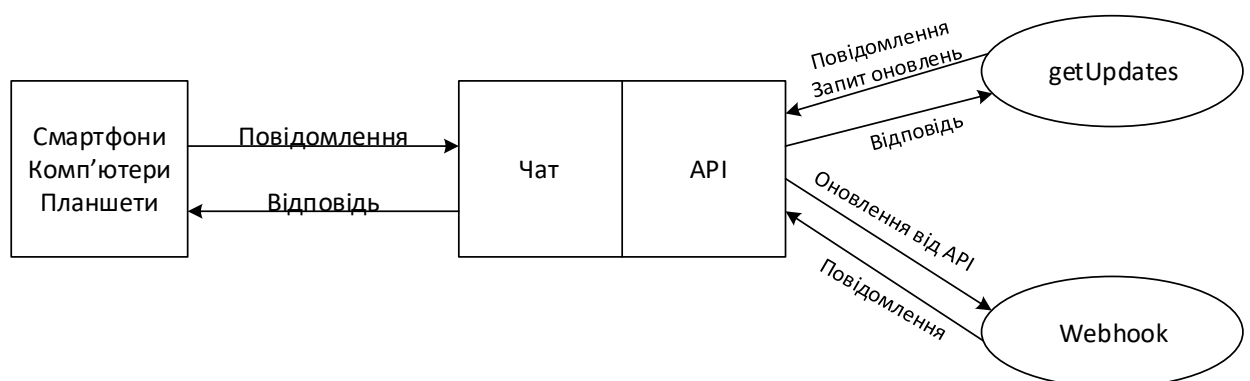


Рис.2.4. Принцип роботи чат-бота на платформі Telegram

						Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

Приклади інших доступних для Telegram Bot API методів описані нижче:

- sendMessage – метод для відправки текстових повідомлень;
- sendLocation – метод для відправки точки на карті;
- editMessageText – метод для редагування текстових повідомлень, відправлених ботом чи за допомогою бота;
- sendAudio – метод для відправки аудіофайлів, якщо бажаєте, щоб клієнти Telegram відображали їх в музичному плеєрі;
- sendContact – метод для відправки телефонних контактів.

Допускаються POST і GET запити. Існує 4 способи для передачі параметрів в Bot API:

- Запит в URL
- application / x-www-form-urlencoded;
- application / json (не прийнятний для завантаження файлів);
- multipart / form-data (для завантаження файлів).

## 2.5 Вибір середовища розробки

PyCharm – це інтегроване середовище розробки для Python, яке має повний комплект засобів, необхідних для ефективного програмування на Python. Перша версія вийшла у 2010 році. Інтерфейс PyCharm можна побачити на рис. 2.5.

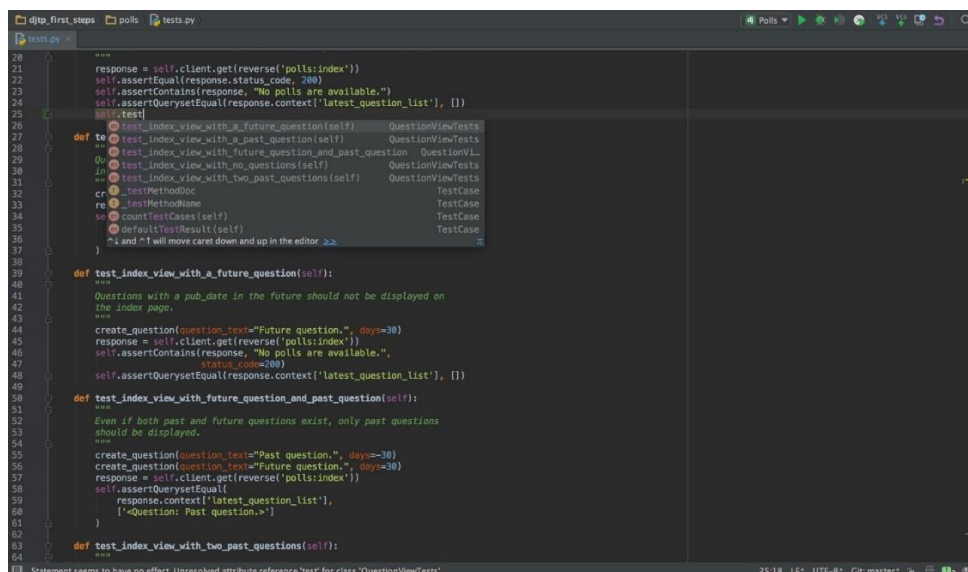


Рис. 2.5. Інтерфейс PyCharm IDE

					Арк.
Змн.	Арк.	№ докум.	Підпис	Дата	

Вбудований відладчик допомагає при налагодженні та тестуванні. Можна використовувати командний рядок для роботи в середовищі. PyCharm також допоможе виявити та оптимізувати неоптимальний програмний код, який виконується дуже довго. Є можливість використовувати віддалений інтерпретатор. Є інтегрований інструмент, що використовується для створення віртуального середовища і його налаштування для розробки Vagrant та розвитку в середовищі, що підтримує контейнерезацію Docker.

PyCharm підтримує модульне тестування, ви можете перевірити, чи правильно працюють код, модулі та різні частини програми.

Під час створення доступу до бази даних середовище розробки надає інструменти для створення запитів SQL до Oracle, SQL Server, PostgreSQL, MySQL та інших видів баз даних. Ви можете писати код SQL та редагувати його вид та переглядати дані на які прийшов запит за його допомогою.

#### Переваги:

- наявність редактору коду зі всіма корисними функціями: підсвічування синтаксису, автоматичне форматування і т. ін.;
- PyCharm дозволяє перевірити версію інтерпретатора мови на сумісність, а також використовувати шаблони коду;
- можливість переглядати документацію прямо у вікні редактора (для елементів) чи в браузері (для зовнішньої документації);
- можливість швидко проводити рефакторинг коду;
- утиліта підтримує всі свіжі версії Django, а також IronPython, Jython, Cython, PyPy wxPython, PyQt, PyGTK тощо.
- можливість проведення інтегрованого Unit тестування та використання інтерактивних консолей для Python, Django, SSH, та БД.
- наявність великої колекції плагінів, його можна використовувати у поєднанні з різними трекерами такими як JIRA, Youtrack, Lighthouse, Redmine, Trac і т. ін.
- PyCharm - кросс-платформне середовище розробки: доступні версії на Linux, Windows та Mac OS.

									Арк.
Змн.	Арк.	№ докум.	Підпис	Дата					

## 2.6. База даних SQLite

Для виконання всіх поставлених задач, чат-бот повинен не тільки володіти інформацією та відображати її для користувача, а й зберігати введену інформацію користувачем. У даній дипломній бакалаврській роботі необхідно зберігати інформацію про людей, які взяли участь в заході та інформацію про тих хто взяв участь в опитуванні та їхній вибір. Тут і виникає потреба у тому, щоб вибрати найбільш зручний та сучасний спосіб для зберігання великого обсягу особистої інформації. В результаті аналізу цього питання, було обрано SQLite, компактна вбудована СУБД, вихідний код якої був переданий в суспільне надбання. Але перед початком роботи з нею, необхідно ознайомитись з основними поняттями.

База даних - це централізоване сховище даних для зберігання, доступу, первинної обробки та отримання інформації. Це набір впорядкованих логічно взаємопов'язаних даних, якими можна спільно користуватися та призначених для задоволення інформаційних потреб користувачів.

База даних повинна містити:

- незалежність даних;
- відкритий доступ до даних;
- підтримка операцій, забезпечених відповідними властивостями;
- гарантування, що немає несправностей;
- робота з багатьма користувачами одночасно.

SQLite - це бібліотека в процесі розробки, що реалізує самодостатнє, без серверне, транзакційне ядро SQL-бази даних. Код SQLite є загальнодоступним, тому його можна використовувати безкоштовно для будь-яких цілей, комерційних або приватних. Це також найпоширеніша база даних у світі, і підтримується великою кількістю різних програм.

Як файлова база даних вона забезпечує чудовий набір інструментів, які допоможуть легше обробляти будь-який тип даних (порівняно з серверними базами даних).

Коли додатки використовують SQLite, вони використовують функціональні та прямі виклики до файлів, що містять дані (наприклад, бази даних SQLite), для

									Арк.
Змн.	Арк.	№ докум.	Підпис	Дата					

зв'язку, замість того, щоб використовувати деякі інтерфейси для підвищення швидкості роботи та продуктивності.

Найбільш важливі особливості, які зробили бібліотеку SQLite популярною, це:

- SQLite не використовує парадигму клієнт-сервер, тобто його движок не є окремо працюючим процесом;
- в якості протоколу обміну використовуються виклики API функцій бібліотеки SQLite;
- декілька процесів чи потоків одночасно можуть зчитувати дані з однієї бази;
- SQLite можна використовувати як на вбудованих системах так і на виділених машинах з великими масивами даних;
- Підтримує динамічне типізування даних.
- Вся база даних зберігається в одному файлі для зручності переміщення.
- SQLite використовує SQL, деякі функції (RIGHT, OUTER, JOIN, FOR, EACH, STATEMENT) опущені, але є деякі нові функції.
- Дуже підходить для розробки та навіть тестування: розширення найбільш необхідне на етапі розробки. SQLite має багатий набір функцій, який може забезпечити достатньо можливостей.

Недоліки:

- Відсутність контролю користувача: Розширені бази даних дозволяють користувачам управляти взаємозв'язками в таблицях на основі дозволів, але SQLite не має такої функції.
- Ніякі додаткові налаштування неможливі: знову ж таки, SQLite не може збільшити свою продуктивність більше, а ні ж це було задумано з самого початку.

Коли варто використовувати базу даних SQLite:

- Вбудовані програми: Усі портативні програми не призначені для розширення - наприклад, локальні однокористувацькі програми, мобільні програми чи ігри.

									Арк.
Змн.	Арк.	№ докум.	Підпис	Дата					

- Система доступу до диска: У більшості випадків програми, які часто виконують операції прямого читання / запису на дисках, можуть бути перетворені в SQLite для кращої продуктивності.
- Тестування: Ідеально підходить для більшості програм, які є частиною функції тестування бізнес-логіки.

									Арк.
Змн.	Арк.	№ докум.	Підпис	Дата					

## РОЗДІЛ 3. РОЗРОБКА ЧАТ-БОТА

### 3.1. Реєстрація чат-бота для Telegram Bot API

Перший крок – реєстрація чат-бота, що розробляється, у Telegram за допомогою головного бота «BotFather». Він призначений для створення нових ботів та керування ними. Його дуже просто знайти, використовуючи пошук Telegram. Після початку роботи з ним за допомогою команди «/start» зі списку команд потрібно обрати «/newbot», що передбачає створення власного бота. Після цього варто ввести назву, яка міститиме в собі слова «Bot» чи «\_bot». Якщо назву введено правильно, то «BotFather» видає токен (спеціальний набір символів для доступу до HTTP API Telegram Bot) та URL-адресу для доступу до чат-бота. Процес реєстрації на рис. 3.1.

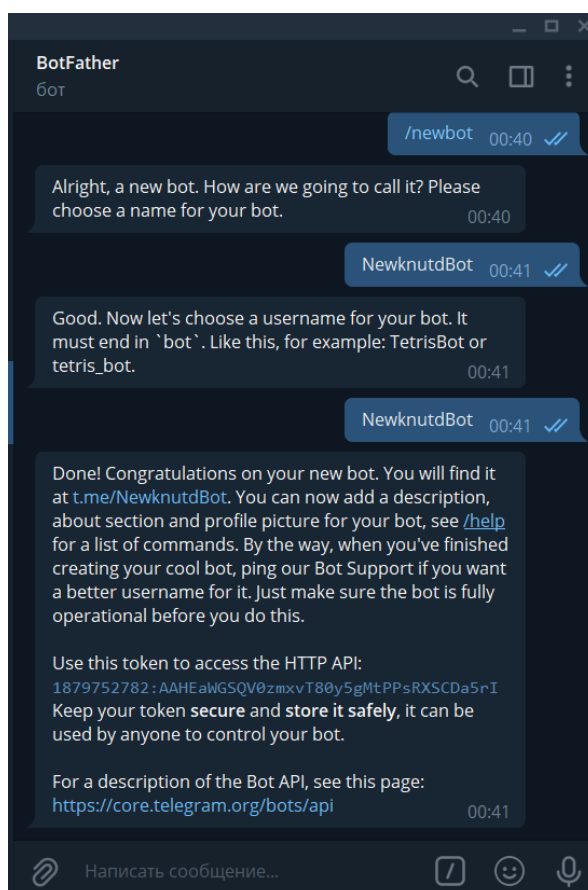


Рис. 3.1. Реєстрація бота у Telegram

Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Варгач О.С.			Розробка програмного забезпечення в задачах комунікації освітнього процесу	Лім.	Арк.	Аркуші
Перевір.		Демківська Т.І.						
Реценз.		Чупринка В.І.						
Н. Контр.		Щербань В.Ю.				КНУТД, ФМКТ		
Затверд.		Щербань В.Ю.						

Для редагування параметрів чат-бота, наприклад, зображення або опису, існує ряд команд (таблиця 3.1).

Табл. 3.1.

Команда	Опис
/setname	Змінює існуюче ім'я
/setdescription	Встановлює текст, який відображається при першому відкритті чат-бота
/setabouttext	Встановлює текст у поле «Про чат-бот»
/setuserpic	Встановлює обране зображення
/setcommands	Дозволяє створити список доступних команд
/deletebot	Видаляє обраного чат-бота

Після того, як ви отримали токен бота для керування від Telegram можна починати розробляти програмну частину.

### 3.2. Реалізація програмної частини

Проект складається з файлів, які зображені на рис. 3.2, що утворюють програмну реалізацію чат-бота на мові програмування Python.

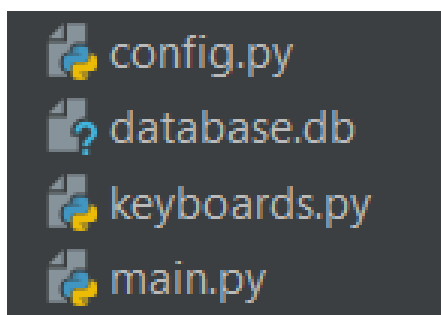


Рис. 3.2. Файлова структура чат-бота

Розглянемо кожен файл проекту детальніше:

									Арк.
Змн.	Арк.	№ докум.	Підпис	Дата					



- config.py – містить інформацію з токеном бота та ідентифікатор адміністратора;
- database.db – файл бази даних;
- keyboards.py – файл в якому створені кнопки чат-бота;
- main.py – головний файл програми, в якому прописана логіка роботи.

Спочатку треба підключити до проекту всі необхідні бібліотеки та модулі необхідні для розробки (рис. 3.3).

```
import sqlite3
from sqlite3 import Error
import keyboards as kb
from aiogram import Bot, types
from aiogram.dispatcher import Dispatcher
from aiogram.utils import executor, exceptions
from aiogram.dispatcher.filters import Text
from aiogram.types import ReplyKeyboardMarkup, KeyboardButton, InlineKeyboardMarkup, InlineKeyboardButton, ReplyKeyboardRemove, message
from aiogram.contrib.fsm_storage.memory import MemoryStorage
from aiogram.dispatcher import FSMContext
from aiogram.dispatcher.filters.state import StatesGroup, State
from aiogram.dispatcher import FSMContext
from aiogram.contrib.fsm_storage.memory import MemoryStorage
from aiogram.types import CallbackQuery
import random
```

Рис. 3.3. Підключення бібліотек та модулів до проекту

### 3.2.1. Створення перших повідомлень

Для початку роботи чат-бота потрібно відправити йому команду «/start», що буде першою командою боту для користувача після вибору діалогу з ним.

Щоб бот сприймав цю команду та починав працювати після неї, створено декоратор.

Декоратор – це функція, яка дозволяє змінити функціональність іншої функції у деякому напрямку, при цьому не змінюючи її код. Приклад декоратора можна побачити на рис. 3.4. Він виступає в ролі «слухача», який обробляє вхідні повідомлення.

```
@dp.message_handler(commands=['start'])
```

Рис. 3.4. Декоратор для початку роботи бота

						Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

За допомогою методу *send\_message* помічник відправить у чат перше повідомлення з описом чат-бота та переліком пунктів головного меню (рис. 3.5).

```
bot.send_message(message.chat.id, text = 'Привіт! Даний бот створений для швидкого пошуку основної інформації п  
' та вступникам, а також для комунікації та прийняття участі в житті університету. '  
' Ви можете швидко знайти потрібну інформацію в розділах:\n1. Університет\n2. Студенту\n3. Вступнику\n4. '  
' Наука та інновації\n5. Корисні посилання\nПриємного користування!', reply_markup = kb.menu)
```

Рис. 3.5. Створення першого повідомлення

Параметри методу *ReplyKeyboardMarkup* для створення клавіатури вказують на ширину рядків (*row\_width*) та на те, що клавіатура змінюватиме свій розмір в залежності від кількості кнопок для кращого її сприйняття користувачем (*resize\_keyboard=True*). На рис. 3.6 зображений програмний код для створення першої клавіатури.

```
menu = types.ReplyKeyboardMarkup(resize_keyboard = True, row_width = 2)  
faq = types.KeyboardButton(text = 'Довідка')  
main = types.KeyboardButton(text = 'Головне меню')  
supp = types.KeyboardButton(text = "Зв'язатись з нами")  
map = types.KeyboardButton('Ми на карті')
```

Рис. 3.6. Створення клавіатури.

### 3.2.2. Розробка головного меню та його складових

При запуску чат-бота користувача зустріне клавіатура з наступними кнопками:

- «Довідка»;
- «Головне меню»;
- «Зв'язатись з нами»;
- «Ми на карті».

В свою чергу пункт «Головне меню» включає в себе:

- «Університет»;
- «Студенту»;
- «Вступнику»;

									Арк.
Змн.	Арк.	№ докум.	Підпис	Дата					

- «Наука та інновації»;
- «Корисні посилання».

Дана клавіатура створюється аналогічно до першої (див. рис. 3.6) і з'являється після натиснення кнопки «Головне меню». Де потім користувач зможе обрати необхідний йому пункт для отримання інформації (рис. 3.7).

Кожен підрозділ головного меню містить інформацію на відповідну тематику і після вибору того чи іншого пункту з'являється перелік інформації, яка вміщується там. На рис. 3.8 зображений перелік інформації пункту меню «Університет».

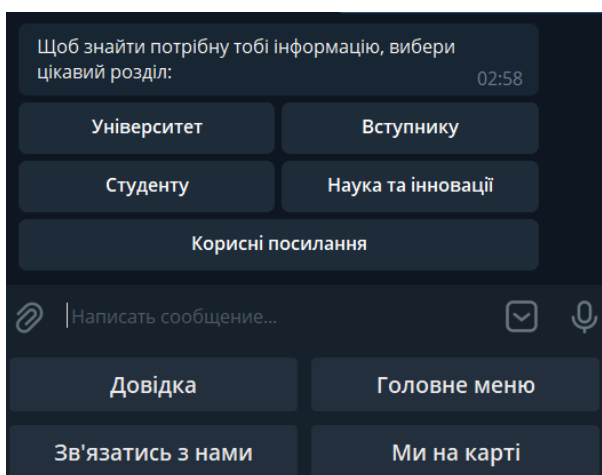


Рис. 3.7. Пункти «Головного меню»

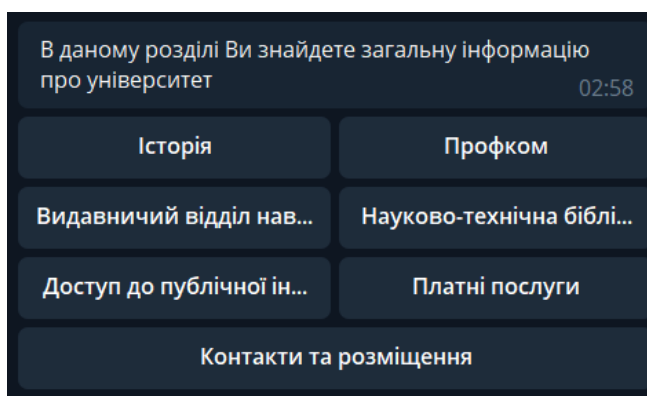


Рис. 3.8. Інформація пункту меню «Університет»

Дана вбудована клавіатура, так само, як і та, що описана вище створюється за допомогою методу *InlineKeyboardButton*. Вона буде знаходитись в чаті діалогу

						Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

нижче тексту з повідомленнями. Частина коду де показано створення такої вбудованої клавіатури зображено на рис. 3.9. Взаємодія з іншими пунктами «Головного меню» відбувається аналогічно до вищеописаної.

```
categories = types.InlineKeyboardMarkup(resize_keyboard=True, row_width=2)
p1 = types.InlineKeyboardButton(text='Університет', callback_data='struct')
p2 = types.InlineKeyboardButton(text='Вступнику', callback_data='abit')
p3 = types.InlineKeyboardButton(text='Студенту', callback_data='stud')
p4 = types.InlineKeyboardButton(text='Наука та інновації', callback_data='nau')
p5 = types.InlineKeyboardButton(text='Корисні посилання', callback_data='kor')
```

Рис. 3.9. Код програми зі створенням вбудованої клавіатури.

Пункт меню «Зв'язатись з нами» виконує перенаправлення користувача на контакт адміністратора для того щоб можна було йому написати. Реалізація цього в програмному коді відбувається за допомогою параметра *url* методу *InlineKeyboardButton* і зображено на рис. 3.10.

```
supp = types.InlineKeyboardMarkup(resize_keyboard=True)
write = types.InlineKeyboardButton(text="Зв'язатись з нами", url='t.me/varg_ol')
supp.add(write)
```

Рис. 3.10. Реалізація можливостей кнопки «Зв'язатись з нами»

За допомогою пункту «Ми на карті» можна подивитись розташування університету, а також його гуртожитків на вбудованій в Telegram карті рис. 3.11. та 3.12.

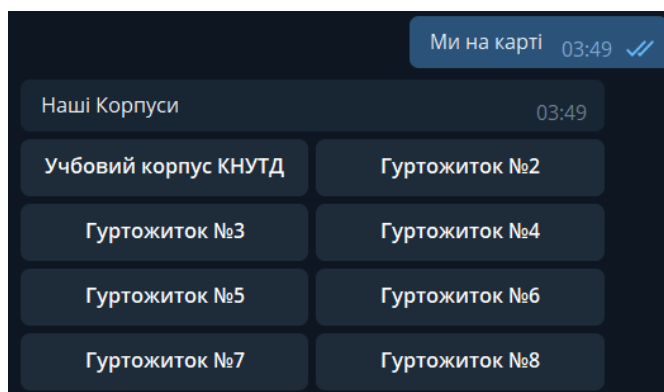


Рис. 3.11. Відображення пунктів меню «Ми на карті»

						Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

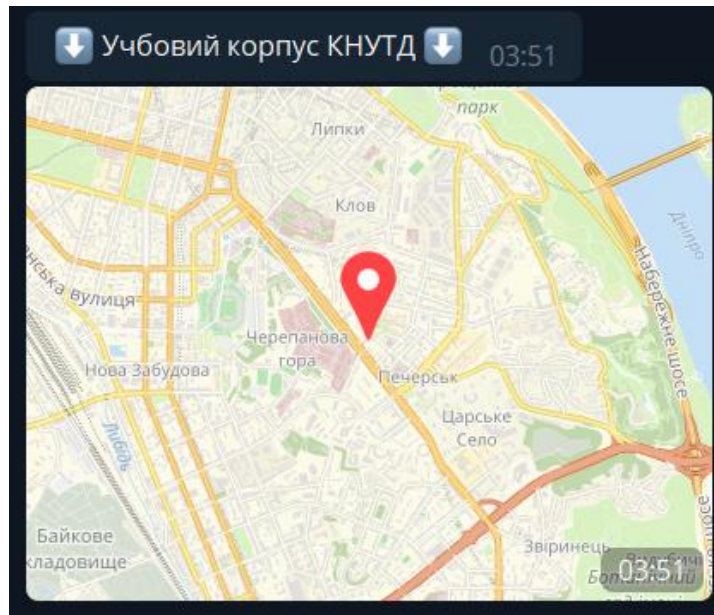


Рис. 3.12. Відображення розташування на карті

Для прокладання маршруту будуть вже використовуватися встановлені карти на вашому пристрої, наприклад Google Maps на смартфоні.

При введенні в чат-бот команди /adm адміністратору відкривається список додаткових команд (рис. 3.13).

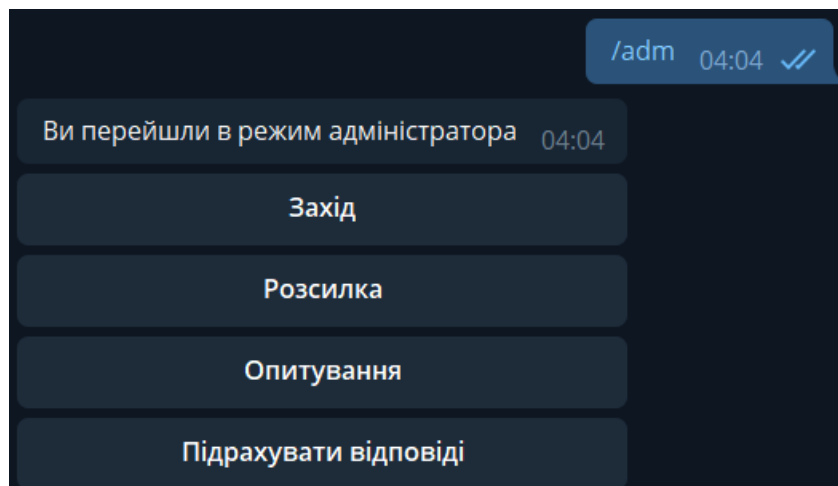


Рис. 3.13. Список команд для адміністратора

Реалізовано це за допомогою даного програмного коду (рис. 3.14) де перемінна *admins* містить id адміністратора в Telegram і він має доступ до цього режиму.

						Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

```

if message.text == '/adm' and (message.from_user.id in admins):
    await bot.send_message(message.chat.id, text = ' Ви перейшли в режим адміністратора', reply_markup=kb.admin)

```

Рис. 3.14. Код переходу в режим адміністратора

### 3.2.3. Підключення бази даних

Для того щоб зберігати дані користувачів та виконувати подальші дії з ними до проекту була підключена база даних SQLite3. Вона необхідна для збереження введених даних при реєстрації на заходи та збереженні відповіді від користувача при опитуванні.

Для під'єднання до бази даних використовується метод *connect()*.

У файлі *main.py* написаний код для з'єднання з базою даних (рис. 3.15). За допомогою методу *cursor()* виконується запит до БД та отримується результат від неї. Тобто за його допомогою створюється порожня БД, записуються та редагуються дані.

```

with sqlite3.connect("database.db") as conn:
    cursor = conn.cursor()
    cursor.execute("""
    CREATE TABLE IF NOT EXISTS users
    (user INTEGER, name TEXT)
    """)
    cursor.execute("""
    CREATE TABLE IF NOT EXISTS ievent
    (ievent TEXT, username TEXT, name TEXT, surname TEXT, groupe TEXT, number INT, dop TEXT)
    """)
    cursor.execute("""
    CREATE TABLE IF NOT EXISTS answers
    (question TEXT, username TEXT, answer TEXT)
    """)

```

Рис. 3.15. Програмний код підключення до бази даних.

За допомогою стрічки *CREATE TABLE IF NOT EXIST* виконується запит чи існує таблиця, якщо так то перевірить чи нічого не змінилось. Також вона допомагає при повторному підключенні до БД.

						Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

SQL-команди виконують за допомогою методу *execute()*. Заповнювати таблиці також можна за допомогою цього методу.

За допомогою цих методів були створені таблиці:

- *users*, яка має поля: *user* та *name*.
- *ivent*, яка має поля: *ivent*, *username*, *name*, *surname*, *groupe*, *number*, *dop*.
- *answers*, яка має поля: *question*, *username* та *answer*.

При першому запуску чат-бота і виборі команди */start*, користувач буде записаний в базу даних для подальшої роботи з цими даними для інших таблиць (рис. 3.16).

```
@dp.message_handler(commands=['start'])
async def send_welcome(message: types.Message):
    cursor.execute("select * from users where user = ?", (message.from_user.id,))
    usr = cursor.fetchone()
    if not usr:
        cursor.execute("INSERT INTO users values (:user, :name);",
            {'user': message.from_user.id,
             'name': message.from_user.first_name})
        conn.commit()
```

Рис. 3.16. Запис нових користувачів до бази даних.

Для розсилки «Заходу» відбувається перегляд в таблиці *users* активних користувачів чат-ботом і потім відправлення всім введених даних від адміністратора

```
global kad
kad = message.text
await state.update_data(kad=kad)
await state.finish()
cursor.execute("SELECT user FROM users")
userbase = []
while True:
    row = cursor.fetchone()
    if row == None:
        break
    userbase.append(row)
if len(userbase) > 1:
    for z in range(len(userbase)):
        await bot.send_message(userbase[z][0], text=f"{kad}" ,reply_markup=_kb.active)
else:
    await bot.send_message(userbase[0], f"{kad}" ,reply_markup=_kb.active)
await bot.send_message(message.chat.id, text=_ "Розсилка заходу була успішно виконана" ,reply_markup=_kb.menu)
await state.finish()
```

Рис. 3.17. Програмний код для розсилки «Заходу» користувачам чат-бота.

						Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

Після того, як користувач натисне взяти участь у заході, йому необхідно буде ввести свої дані після чого вони запишуться в БД (рис. 3.18).

```
await bot.send_message(message.chat.id, text=... 'Ви взяли участь.\nВідповідь була записана')
data = await state.get_data()
name = data.get('name')
surname = data.get('surname')
groupe = data.get('groupe')
number = data.get('number')
cursor.execute("INSERT INTO ivent values (:ivent, :username, :name, :surname, :groupe, :number, :dop);" ...,
               {'ivent': kad,
                'username': message.from_user.username,
                'name': name,
                'surname': surname,
                'groupe': groupe,
                'number': number,
                'dop': dop})
conn.commit()
await state.finish()
```

Рис. 3.18. Запис даних користувача до бази даних.

Розсилка опитування виконується аналогічним способом, що був описаний вище.

Для того щоб переглянути кількість відповідей з опитування відбувається запит в базу даних по назві опитування та назві варіанту відповіді, після чого в чат бот надсилається необхідну відповідь (рис. 3.19). Якщо значення не були знайдені в базі даних, то адміністратор отримає відповідне повідомлення.

```
await bot.send_message(message.chat.id, text=... 'Введіть порядковий номер відповіді який бажаєте підрахувати')
await cont.answer.set()

@dps.message_handler(state=cont.answer)
async def konkname(message: types.Message, state: FSMContext):
    if message.text == 'Повернутись назад':
        await bot.send_message(message.chat.id, text=... 'Ви були повернуті в адмінку', reply_markup=kb.admin)
        await state.finish()
        return
    answer = message.text
    data = await state.get_data()
    quest = data.get('nazva')
    await state.update_data(answer=answer)
    try:
        allans = cursor.execute(f"SELECT Count(*) FROM answers WHERE answer = {answer} and question = '{quest}'").fetchone()[0]
        await bot.send_message(message.chat.id, text=... f'Кількість відповідей: {allans}')
    except:
        await bot.send_message(message.chat.id, text=... 'Опитувань з такою назвою/відповіддю немає', reply_markup=... kb.menu)
    await state.finish()
```

Рис. 3.19. Програмний код для отримання кількості відповідей опитування.

						Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		



Для збереження результатів в таблицях використовується команда *commit*. Вона зберігає всі транзакції в базі даних з моменту останньої команди *commit*. Записані дані в базу даних мають наступний вигляд (рис. 3.20).

rowid	ivent	username	name	surname	groupe	number	dop
Click here to define a filter							
1	День відкритих дверей	varg_ol	Олег	Варгач	БІТск-19	38096969696	----
2	День відкритих дверей	ivanov	Іван	Іванов	БІТ21	38095959595	Зателефонуйте мені

Рис. 3.20. Вигляд даних записаних у базу даних

										Арк.
Змн.	Арк.	№ докум.	Підпис	Дата						

## ВИСНОВКИ

Результатом виконання дипломної бакалаврської роботи є Telegram чат-бот, який вирішує задачі комунікації в освітньому процесі.

В першому розділі була досліджена предметна область, а саме платформа Telegram, загальні принципи чат-ботів та принцип їх роботи. Розглянуті існуючі чат-боти для порівняння та встановлення функціональності власного розроблювального чат-бота в Telegram.

У другому розділі описано призначення розроблювального чат-бота, вимоги його функціональності та задач, які повинен виконувати. Також важливими пунктами є вибір мови програмування та середовища розробки, бази даних та огляд інтерфейсу Telegram Bot API. Вибір цих всіх складових впливає на роботу проекту і кінцевий результат.

У третьому розділі описані етапи розробки чат-бота, починаючи від його створення в Telegram і закінчуючи підключенням і взаємодією з ним бази даних.

Завдяки правильному відбору всіх аспектів, необхідних для отримання високоякісних та сучасних програмних продуктів, було розроблено чат-бот, який повністю відповідає описаним вимогам та особливостям. Бот може працювати безперебійно і може приймати запити від багатьох користувачів одночасно.

Він є унікальним помічником під час навчального процесу і без сумніву полегшує пошук інформації, а також швидкість її пошуку, а також надає функції для комунікації з користувачами, зберігання введеної інформації для її подальшої обробки.

Даний програмний продукт має перспективу для розвитку і розширення свого функціоналу, що дозволить оптимізувати і полегшити у вирішення ще більшої кількості задач комунікації в освітньому процесі.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. A Byte of Python / Chitlur Swaroop., 2020. – 158 с. – (2.0.2).
2. PyCharm [Електронний ресурс] – Режим доступу до ресурсу: <https://www.jetbrains.com/ru-ru/pycharm/>.
3. pyTelegramBotAPI [Електронний ресурс]-Режим доступу до ресурсу: <https://github.com/eternnoir/pyTelegramBotAPI>
4. Python 3. Самое необходимое. / В. Прохоренко, В. Дронов. – Санкт-Петербург: БХВ-Петербург, 2016. – 445 с.
5. SQLite, MySQL и PostgreSQL: сравниваем популярные реляционные СУБД [Електронний ресурс] – Режим доступу до ресурсу: <https://tproger.ru/translations/sqlite-mysql-postgresql-comparison/>.  
[stud/uk/nurse/and/05.Принципи%20побудови%20та%20сфери%20застосувань%20баз%20даних.htm](https://stud.uk/nurse/and/05.Принципи%20побудови%20та%20сфери%20застосувань%20баз%20даних.htm)
6. Telegram Bot API [Електронний ресурс] – Режим доступу до ресурсу: <https://core.telegram.org/bots/api>
7. Telegram FAQ [Електронний ресурс] – Режим доступу до ресурсу: <https://telegram.org/>.
8. Welcome to aiogram's documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.aiogram.dev/en/latest/>
9. Как создать Telegram бота с помощью Python [Електронний ресурс] – Режим доступу: <https://medium.com/nuances-of-programming/>
10. Комунікації в організаціях [Електронний ресурс] – Режим доступу до ресурсу: <https://ru.osvita.ua/vnz/reports/management/15040/>.
11. Можливості використання чат-ботів в освітній сфері [Електронний ресурс] / Юлія Сергіївна Чаплінська – Режим доступу до ресурсу: [http://mediaosvita.org.ua/wp-content/uploads/2020/07/СНaplinska\\_YUS\\_Vykorystannya\\_chat-botiv\\_v\\_osviti.pdf](http://mediaosvita.org.ua/wp-content/uploads/2020/07/СНaplinska_YUS_Vykorystannya_chat-botiv_v_osviti.pdf).
12. Описание продукта JetBrains PyCharm [Електронний ресурс] – Режим доступу до ресурсу: <https://itpro.ua/product/jetbrains-pycharm/?tab=description>.
13. Офіційний сайт SQLite [Електронний ресурс] – Режим доступу до ресурсу:

14. Офіційний сайт КНУТД [Електронний ресурс] – Режим доступу до ресурсу: <https://www.knutd.edu.ua/>
15. Пишем Telegram-ботов на Python (v2) [Електронний ресурс] – Режим доступу до ресурсу: <https://mastergroosha.github.io/telegram-tutorial-2/>.
16. Пишем ботов для Telegram на языке Python [Електронний ресурс] – Режим доступу до ресурсу: <https://mastergroosha.github.io/telegram-tutorial/>
17. Принципи побудови та сфери застосувань баз даних [Електронний ресурс] – Режим доступу до ресурсу: [tdmuv.com/kafedra/internal/informatika/classes\\_](http://tdmuv.com/kafedra/internal/informatika/classes_)
18. Принципи проектування програм [Електронний ресурс] – Режим доступу до ресурсу: [https://allref.com.ua/uk/skachaty/Principi\\_proektuvannya\\_program](https://allref.com.ua/uk/skachaty/Principi_proektuvannya_program).
19. Путівник мовою програмування Python [Електронний ресурс] / Олександр Мізюк. – 2020. – Режим доступу до ресурсу: <https://pythonguide.rozh2sch.org.ua/>.
20. Работа с базами данных SQLite [Електронний ресурс] – Режим доступу до ресурсу: [oxygensoftware.zendesk.com/hc/ru/articles/115000087304-Работа-с-базами-данных-SQLite](https://oxygensoftware.zendesk.com/hc/ru/articles/115000087304-Работа-с-базами-данных-SQLite)
21. Самоучитель Python [Електронний ресурс] – Режим доступу до ресурсу: <https://pythonworld.ru/samouchitel-python>
22. Создание базы данных и таблиц [Електронний ресурс] – Режим доступу до ресурсу: <https://younglinux.info/sqlite/create>.
23. Соціальні мережі та месенджери в Україні [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <http://rb.com.ua/uk/blog-uk/omnibus-uk/socialni-merezhi-ta-mesendzheri-v-ukraini/>.
24. Телеграм бот для поддержки своими руками [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://habr.com/ru/post/539766/>.
25. Телеграм-бот на Python [Електронний ресурс] – Режим доступу до ресурсу: <https://thecode.media/python-bot/>.
26. Чат-бот - что это такое, как работает, виды, преимущества и недостатки, как создать [Електронний ресурс] – Режим доступу до ресурсу: <http://kak-bog.ru/chat-bot-cto-eto-takoe-kak-rabotaet-vidy-preimushchestva>

- 27.** Чат-боти для навчання [Електронний ресурс] – Режим доступу до ресурсу:  
<http://teach-hub.com/chat-boty-dlia-navchannia-ohliad-na-ypopuliarnishykh-ta-osoblyvosti-vykorystannia/>.
- 28.** Чат-боти як нове покоління каналів комунікації [Електронний ресурс] / О. С. Курилець – Режим доступу до ресурсу:  
<https://core.ac.uk/download/pdf/197267488.pdf>.
- 29.** Чат-боты: обзор и состояние технологий в отрасли [Електронний ресурс] – Режим доступу до ресурсу: <http://nlrx.net/archives/425>
- 30.** 42 полезных бота для Telegram [Електронний ресурс] – Режим доступу до ресурсу: <https://l-a-b-a.com/blog/1973-42-poleznyh-bota-dlya-telegram>

## ДОДАТОК А. ТЕКСТ ПРОГРАМИ

Файл config.py:

```
ТОК = '1834702380:AAF1RNSUo37H1Nr2nWBOAr53xyFZjP0rV2U'
```

```
ADMINS = [463203436]
```

Файл keyboards.py

```
from aiogram import types
from aiogram.types import ReplyKeyboardMarkup, KeyboardButton,
InlineKeyboardMarkup, InlineKeyboardButton, ReplyKeyboardRemove
menu = types.ReplyKeyboardMarkup(resize_keyboard = True,row_width = 2)
faq = types.KeyboardButton(text = 'Довідка')
main = types.KeyboardButton(text = 'Головне меню')
supp = types.KeyboardButton(text = "Зв'язатись з нами")
map = types.KeyboardButton('Ми на карті')
menu.add(faq,main,supp,map)
map = types.InlineKeyboardMarkup(resize_keyboard = True,row_width = 2)
bud1 = types.InlineKeyboardButton(text = 'Учбовий корпус КНУТД',callback_data =
'bud1')
bud2 = types.InlineKeyboardButton(text = 'Гуртожиток №2',callback_data = 'bud2')
bud3 = types.InlineKeyboardButton(text = 'Гуртожиток №3',callback_data = 'bud3')
bud4 = types.InlineKeyboardButton(text = 'Гуртожиток №4',callback_data = 'bud4')
bud5 = types.InlineKeyboardButton(text = 'Гуртожиток №5',callback_data = 'bud5')
bud6 = types.InlineKeyboardButton(text = 'Гуртожиток №6',callback_data = 'bud6')
bud7 = types.InlineKeyboardButton(text = 'Гуртожиток №7',callback_data = 'bud7')
bud8 = types.InlineKeyboardButton(text = 'Гуртожиток №8',callback_data = 'bud8')
map.add(bud1,bud2,bud3,bud4,bud5,bud6,bud7,bud8)
supp = types.InlineKeyboardMarkup(resize_keyboard = True)
write = types.InlineKeyboardButton(text = "Зв'язатись з нами",url = 't.me/varg_ol')
supp.add(write)
categories = types.InlineKeyboardMarkup(resize_keyboard = True,row_width = 2)
p1 = types.InlineKeyboardButton(text = 'Університет',callback_data = 'struct')
p2 = types.InlineKeyboardButton(text = 'Вступнику',callback_data = 'abit')
p3 = types.InlineKeyboardButton(text = 'Студенту',callback_data = 'stud')
p4 = types.InlineKeyboardButton(text = 'Наука та інновації',callback_data = 'nau')
p5 = types.InlineKeyboardButton(text = 'Корисні посилання',callback_data = 'kor')
categories.add(p1,p2,p3,p4,p5)
admin = types.InlineKeyboardMarkup(resize_keyboard = True,row_width = 1)
opros = types.InlineKeyboardButton(text = 'Опитування',callback_data = 'opr')
konkurs = types.InlineKeyboardButton(text = 'Захід',callback_data = 'kon')
rassylka = types.InlineKeyboardButton(text = 'Розсилка',callback_data = 'ras')
count = types.InlineKeyboardButton(text = 'Підрахувати відповіді',callback_data =
'con')
admin.add(konkurs,rassylka,opros,count)
studtype = types.InlineKeyboardMarkup(resize_keyboard = True,row_width = 2)
g1 = types.InlineKeyboardButton(text = "Права та обов'язки студента",callback_data
```

```
= 'g1')
g2 = types.InlineKeyboardButton(text = 'Центр праці та кар'єри',callback_data = 'g2')
g3 = types.InlineKeyboardButton(text = 'Первинна профспілкова організація
студентів КНУТД',callback_data = 'g3')
g4 = types.InlineKeyboardButton(text = 'Студентський Парламент',callback_data =
'g4')
g5 = types.InlineKeyboardButton(text = 'Гуртожитки',callback_data = 'g5')
g6 = types.InlineKeyboardButton(text = 'Центр культури та мистецтв',callback_data =
'g6')
g7 = types.InlineKeyboardButton(text = 'Студентський будинок моди',callback_data
= 'g7')
g8 = types.InlineKeyboardButton(text = 'Всеукраїнські студентські
олімпіади',callback_data = 'g8')
g9 = types.InlineKeyboardButton(text = 'Оплата послуг через ПриватБанк
',callback_data = 'g9')
studtype.add(g1,g2,g3,g4,g5,g6,g7,g8,g9)
nautype = types.InlineKeyboardMarkup(resize_keyboard = True,row_width = 2)
k1 = types.InlineKeyboardButton(text = "Наука та інноваційна
діяльність",callback_data = 'k1')
k2 = types.InlineKeyboardButton(text = 'Інституційний репозиторій ',callback_data =
'k2')
k3 = types.InlineKeyboardButton(text = 'Науково-дослідна частина',callback_data =
'k3')
k4 = types.InlineKeyboardButton(text = 'Відділ докторантури і аспірантури
',callback_data = 'k4')
k5 = types.InlineKeyboardButton(text = 'Спеціалізовані вчені ради',callback_data =
'k5')
k6 = types.InlineKeyboardButton(text = 'Наукові та мистецькі школи',callback_data =
'k6')
k7 = types.InlineKeyboardButton(text = 'Всеукраїнський конкурс студентських
наукових робіт',callback_data = 'k7')
k8 = types.InlineKeyboardButton(text = 'Наукове товариство студентів та
аспірантів',callback_data = 'k8')
nautype.add(k1,k2,k3,k4,k5,k6,k7,k8)
abitype = types.InlineKeyboardMarkup(resize_keyboard = True,row_width = 2)
s1 = types.InlineKeyboardButton(text = 'Правила прийому до КНУТД в 2021
р.',callback_data = 's1')
s2 = types.InlineKeyboardButton(text = 'Приймальна комісія',callback_data = 's2')
s3 = types.InlineKeyboardButton(text = 'Перелік спеціальностей і освітніх
програм',callback_data = 's3')
s4 = types.InlineKeyboardButton(text = 'Сторінки освітніх програм ',callback_data =
's4')
s5 = types.InlineKeyboardButton(text = 'Військова підготовка',callback_data = 's5')
s6 = types.InlineKeyboardButton(text = 'English for Exam',callback_data = 's6')
s7 = types.InlineKeyboardButton(text = 'Підготовка до ЗНО та вступних
```

```
випробувань',callback_data = 's7')
s8 = types.InlineKeyboardButton(text = 'Заняття в червні – липні.',callback_data =
's8')
s9 = types.InlineKeyboardButton(text = 'Умови прийому на навчання до закладів
вищої освіти України ',callback_data = 's9')
s10 = types.InlineKeyboardButton(text = 'Програми фахових вступних
випробувань',callback_data = 's10')
s11 = types.InlineKeyboardButton(text = 'Інформаційний пакет ЄКТС',callback_data
= 's11')
s12 = types.InlineKeyboardButton(text = 'Обсяг держ. замовлення на підготовку
фахівців з вищою освітою',callback_data = 's12')
s13 = types.InlineKeyboardButton(text = 'Про захист персональних
даних',callback_data = 's13')
s14 = types.InlineKeyboardButton(text = 'Вступна кампанія',callback_data = 's14')
abitype.add(s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12,s13,s14)
structype = types.InlineKeyboardMarkup(resize_keyboard = True,row_width = 2)
et1 = types.InlineKeyboardButton(text = 'Історія',callback_data = 'hist')
et2 = types.InlineKeyboardButton(text = 'Профком',callback_data = 'prof')
et3 = types.InlineKeyboardButton(text = 'Видавничий відділ навчально-наукових
матеріалів',callback_data = 'vid')
et4 = types.InlineKeyboardButton(text = 'Науково-технічна бібліотека
КНУТД',callback_data = 'nauk')
et5 = types.InlineKeyboardButton(text = 'Доступ до публічної
інформації',callback_data = 'dost')
et6 = types.InlineKeyboardButton(text = 'Платні послуги',callback_data = 'plat')
et7 = types.InlineKeyboardButton(text = 'Контакти та розміщення',callback_data =
'kont')
structype.add(et1,et2,et3,et4,et5,et6,et7)
back = types.InlineKeyboardMarkup(resize_keyboard = True)
nazad = types.InlineKeyboardButton(text = 'Повернутися в Головне
меню',callback_data = 'back')
back.add(nazad)
exback = types.InlineKeyboardMarkup(resize_keyboard = True)
nazad = types.InlineKeyboardButton(text = 'Повернутися в Головне
меню',callback_data = 'exback')
exback.add(nazad)
entertype = types.InlineKeyboardMarkup(resize_keyboard = True,row_width = 2)
et1 = types.InlineKeyboardButton(text = 'Что такое бакалавриат',callback_data = "")
et2 = types.InlineKeyboardButton(text = 'Что такое магистратура',callback_data = "")
et3 = types.InlineKeyboardButton(text = 'Контакты приём комиссии',callback_data =
"")
et4 = types.InlineKeyboardButton(text = 'Направление подготовки',callback_data = "")
et5 = types.InlineKeyboardButton(text = 'Помощь поступающим',callback_data = "")
et6 = types.InlineKeyboardButton(text = 'Информация об общежитии',callback_data
= "")
```



```

et7 = types.InlineKeyboardButton(text = 'Необходимые документы',callback_data = "")
et8 = types.InlineKeyboardButton(text = 'Где посмотреть стоимость
обучения?',callback_data = "")
et9 = types.InlineKeyboardButton(text = 'Назад',callback_data = "")
entertype.add(et1,et2,et3,et4,et5)
admexit = types.ReplyKeyboardMarkup(resize_keyboard=True)
nazad = types.KeyboardButton(text = 'Вернуться назад')
admexit.add(nazad)
active = types.InlineKeyboardMarkup(resize_keyboard = True)
acti = types.InlineKeyboardButton(text = 'Брати участь',callback_data = 'act')
active.add(acti)
rasstypes = types.InlineKeyboardMarkup(resize_keyboard = True,row_width = 1)
text = types.InlineKeyboardButton(text = 'Текст',callback_data = 'tx')
photo = types.InlineKeyboardButton(text = 'Текст + Фото',callback_data = 'ph')
doc = types.InlineKeyboardButton(text = 'Текст + Документ',callback_data = 'doc')
rasstypes.add(text,photo,doc)

```

Файл main.py

```

import sqlite3
from sqlite3 import Error
import keyboards as kb
from aiogram import Bot, types
from aiogram.dispatcher import Dispatcher
from aiogram.utils import executor, exceptions
from aiogram.dispatcher.filters import Text
from aiogram.types import ReplyKeyboardMarkup, KeyboardButton,
InlineKeyboardMarkup, InlineKeyboardButton, ReplyKeyboardRemove, message
from aiogram.contrib.fsm_storage.memory import MemoryStorage
from aiogram.dispatcher import FSMContext
from aiogram.dispatcher.filters.state import StatesGroup, State
from aiogram.dispatcher import FSMContext
from aiogram.contrib.fsm_storage.memory import MemoryStorage
from aiogram.types import CallbackQuery
import random
from config import ADMINS,TOK
with sqlite3.connect("database.db") as conn:
    cursor = conn.cursor()
    cursor.execute("""
CREATE TABLE IF NOT EXISTS users
(user INTEGER, name TEXT)
""")
    cursor.execute("""
CREATE TABLE IF NOT EXISTS ivent
(ivent TEXT,username TEXT,name TEXT, surname TEXT,groupe TEXT,number
INT,dop TEXT)
""")

```

```

cursor.execute("""
CREATE TABLE IF NOT EXISTS answers
(question TEXT,username TEXT,answer TEXT)
""")
TOKEN = TOK
bot = Bot(token=TOKEN)
dp = Dispatcher(bot,storage = MemoryStorage())
admins = ADMINS
class cont(StatesGroup):
    nazva = State()
    answer = State()
class opros(StatesGroup):
    question = State()
    count = State()
class admad(StatesGroup):
    ad = State()
class kadmad(StatesGroup):
    kad = State()
class konkurs(StatesGroup):
    name = State()
    dop = State()
class konkurs(StatesGroup):
    name = State()
    surname = State()
    groupe = State()
    number = State()
    dop = State()
class getph(StatesGroup):
    photo = State()
    phtext = State()
class getdoc(StatesGroup):
    doc = State()
    doctext = State()
@dp.message_handler(commands=['start'])
async def send_welcome(message: types.Message):
    cursor.execute("select * from users where user = ?",(message.from_user.id,))
    usr = cursor.fetchone()
    if not usr:
        cursor.execute("INSERT INTO users values (:user, :name);" ,
            {'user':message.from_user.id,
            'name':message.from_user.first_name})
        conn.commit()
    await bot.send_message(message.chat.id,text = 'Привіт! Даний бот створений для
швидкого пошуку основної інформації про університет студентам '
'та вступникам, а також для комунікації та прийняття участі в житті

```

університету. '

'Ви можете швидко знайти потрібну інформацію в розділах:\n1. Університет\n2. Студенту\n3. Вступнику\n4. '

'Наука та інновації\n5. Корисні посилання\nПриємного користування!',reply\_markup = kb.menu)

@dp.message\_handler(content\_types = ['text'])

async def get\_text(message: types.message,state: FSMContext):

if message.text == 'Довідка':

await bot.send\_message(message.chat.id,text = ""Привіт! Даний бот створений для швидкого пошуку основної інформації про університет студентам та вступникам, а також для комунікації та прийняття участі в житті університету. Ви можете швидко знайти потрібну інформацію в розділах:\n1. Університет\n2. Студенту\n3. Вступнику\n4. Наука та інновації\n5. Корисні посилання\nПриємного користування!""")

if message.text == 'Головне меню':

await bot.send\_message(message.chat.id,text = 'Щоб знайти потрібну тобі інформацію, вибери цікавий розділ:',reply\_markup = kb.categories)

if message.text == '/adm' and (message.from\_user.id in admins):

await bot.send\_message(message.chat.id,text = ' Ви перейшли в режим адміністратора',reply\_markup=kb.admin)

if message.text == "Зв'язатись з нами":

await bot.send\_message(message.chat.id,text = 'Напишіть нам, що вас цікавить',reply\_markup=kb.supr)

if message.text == 'Вернуться назад':

await bot.send\_message(message.chat.id,text = 'Ви були повернуті в Головне меню',reply\_markup=kb.menu)

await state.finish()

return

if message.text == 'Ми на карті':

await bot.send\_message(message.chat.id,text = 'Наші Корпуси',reply\_markup=kb.map)

@dp.message\_handler(state=admad.ad)

async def rass(message: types.Message, state: FSMContext):

if message.text == 'Вернуться назад':

await bot.send\_message(message.chat.id,text = 'Ви були повернуті в адмінку',reply\_markup=kb.admin)

await state.finish()

return

ad = message.text

await state.update\_data(ad=ad)

await state.finish()

cursor.execute("SELECT user FROM users")

userbase = []

while True:

row = cursor.fetchone()

```

    if row == None:
        break
    userbase.append(row)
if len(userbase) > 1:
    for z in range(len(userbase)):
        await bot.send_message(userbase[z][0],text = f" {ad} ")
else:
    await bot.send_message(userbase[0], f" {ad} ")
    await bot.send_message(message.chat.id,text = 'Розсилка опитування була успішно виконана',reply_markup = kb.menu)
    await state.finish()
@dp.message_handler(state=kadmad.kad)
async def konk(message: types.Message, state: FSMContext):
    if message.text == 'Повернутись назад':
        await bot.send_message(message.chat.id,text = 'Ви були повернуті в адмінку',reply_markup=kb.admin)
        await state.finish()
        return
global kad
kad = message.text
await state.update_data(kad=kad)
await state.finish()
cursor.execute("SELECT user FROM users")
userbase = []
while True:
    row = cursor.fetchone()
    if row == None:
        break
    userbase.append(row)
if len(userbase) > 1:
    for z in range(len(userbase)):
        await bot.send_message(userbase[z][0],text = f" {kad} ",reply_markup = kb.active)
else:
    await bot.send_message(userbase[0], f" {kad} ",reply_markup = kb.active)
    await bot.send_message(message.chat.id,text = 'Розсилка заходу була успішно виконана',reply_markup = kb.menu)
    await state.finish()
@dp.message_handler(state=konkurs.name)
async def konkname(message: types.Message, state: FSMContext):
    if message.text == 'Повернутись назад':
        await bot.send_message(message.chat.id,text = 'Ви відмовились від участі в заході.',reply_markup=kb.menu)
        await state.finish()
        return

```

```

name = message.text
await state.update_data(name=name)
await bot.send_message(message.chat.id,text = 'Введіть ваше прізвище')
await konkurs.surname.set()
@dp.message_handler(state=konkurs.surname)
async def konkname(message: types.Message, state: FSMContext):
    if message.text == 'Повернутись назад':
        await bot.send_message(message.chat.id,text = 'Ви відмовились від участі в заході.',reply_markup=kb.menu)
        await state.finish()
        return
    surname = message.text
    await state.update_data(surname=surname)
    await bot.send_message(message.chat.id,text = 'Введіть номер вашої групи')
    await konkurs.groupe.set()
@dp.message_handler(state=konkurs.groupe)
async def konkname(message: types.Message, state: FSMContext):
    if message.text == 'Вернуться назад':
        await bot.send_message(message.chat.id,text = 'Ви відмовились від участі в заході.',reply_markup=kb.menu)
        await state.finish()
        return
    groupe = message.text
    await state.update_data(groupe=groupe)
    await bot.send_message(message.chat.id,text = 'Введіть ваш номер телефону')
    await konkurs.number.set()
@dp.message_handler(state=konkurs.number)
async def konkname(message: types.Message, state: FSMContext):
    if message.text == 'Вернуться назад':
        await bot.send_message(message.chat.id,text = 'Ви відмовились від участі в заході.',reply_markup=kb.menu)
        await state.finish()
        return
    number = message.text
    await state.update_data(number=number)
    await bot.send_message(message.chat.id,text = 'Ваша примітка (за бажанням)')
    await konkurs.dop.set()
@dp.message_handler(state=konkurs.dop)
async def konkname(message: types.Message, state: FSMContext):
    if message.text == 'Вернуться назад':
        await bot.send_message(message.chat.id,text = 'Ви відмовились від участі в заході.',reply_markup=kb.menu)
        await state.finish()
        return
    dop = message.text

```

```

await state.update_data(dop=dop)
await bot.send_message(message.chat.id,text = 'Ви взяли участь.\nВідповідь була
записана')
data = await state.get_data()
name = data.get('name')
surname = data.get('surname')
groupe = data.get('groupe')
number = data.get('number')
cursor.execute("INSERT INTO ivent values (:ivent, :username, :name, :surname,
:groupe, :number, :dop);" ,
    {'ivent': kad,
    'username': message.from_user.username,
    'name': name,
    'surname': surname,
    'groupe': groupe,
    'number': number,
    'dop': dop})
conn.commit()
await state.finish()
@dp.message_handler(state=opros.question)
async def konknae(message: types.Message, state: FSMContext):
    if message.text == 'Повернутись назад':
        await bot.send_message(message.chat.id,text = 'Ви відмовились від участі в
заході.',reply_markup=kb.menu)
        await state.finish()
        return
    question = message.text
    await state.update_data(question=question)
    await bot.send_message(message.chat.id,text = 'Введіть відповіді в наступному
форматі:\nвідповідь1, відповідь2, відповідь3...')
    await opros.count.set()
@dp.message_handler(state=opros.count)
async def kome(message: types.Message, state: FSMContext):
    global count
    count = message.text
    data = await state.get_data()
    global que
    que = data.get('question')
    await state.update_data(count=count)
    cursor.execute("SELECT user FROM users")
    userbase = []
    while True:
        row = cursor.fetchone()
        if row == None:
            break

```

```

    userbase.append(row)
if len(userbase) > 1:
    for z in range(len(userbase)):
        await bot.send_message(userbase[z][0],text = f'{que}',reply_markup =
answer_inline())
    else:
        await bot.send_message(userbase[0], text = f'{que}',reply_markup =
answer_inline())
        await bot.send_message(message.chat.id,text = 'Розсилка опитування була успішно
виконана',reply_markup = kb.menu)
        await state.finish()
def answer_inline():
    answer_inline = types.ReplyKeyboardRemove()
    answer_inline = types.InlineKeyboardMarkup()
    answers = count.split(",") # Массив с ответами
    i = 1
    for answer in answers:
        answer_inline.add(types.InlineKeyboardButton(answer,
callback_data=f'answer_number_{i}'))
        i += 1
    return answer_inline
@dp.message_handler(state=getph.phtext)
async def konkname(message: types.Message, state: FSMContext):
    if message.text == 'Вернуться назад':
        await bot.send_message(message.chat.id,text = 'Ви були повернуті в
адмінку',reply_markup=kb.admin)
        await state.finish()
        return
    phtext = message.text
    await state.update_data(phtext=phtext)
    await bot.send_message(message.chat.id,text = 'Завантажте фото')
    await getph.photo.set()
    @dp.message_handler(state=getph.photo,content_types=['photo'])
async def konkname(message: types.Message, state: FSMContext):
    if message.text == 'Вернуться назад':
        await bot.send_message(message.chat.id,text = 'Ви були повернуті в
адмінку',reply_markup=kb.admin)
        await state.finish()
        return
    name = random.randint(1,99999)
    await message.photo[-1].download(f'{name}.jpg')
    data = await state.get_data()
    tx = data.get('phtext')
    photo = open(f'C:\\Users\\s27_s\\Desktop\\quest-bot\\{name}.jpg', 'rb')
    cursor.execute("SELECT user FROM users")

```

```

userbase = []
while True:
    row = cursor.fetchone()
    if row == None:
        break
    userbase.append(row)
if len(userbase) > 1:
    for z in range(len(userbase)):
        with open(f'C:\\Users\\s27_s\\Desktop\\quest-bot\\{name}.jpg', 'rb') as photo:
            await bot.send_photo(userbase[z][0],photo,caption = f" {tx}")
        await bot.send_message(message.chat.id,text = 'Розсилка опитування була успішно
виконана',reply_markup = kb.menu)
        await state.finish()
@dp.message_handler(state=getdoc.doctext)
async def konkname(message: types.Message, state: FSMContext):
    if message.text == 'Вернуться назад':
        await bot.send_message(message.chat.id,text = 'Ви були повернуті в
адмінку',reply_markup=kb.admin)
        await state.finish()
        return
    doctext = message.text
    await state.update_data(doctext=doctext)
    await bot.send_message(message.chat.id,text = 'Завантажте файл')
    await getdoc.doc.set()
@dp.message_handler(state=getdoc.doc,content_types=['document'])
async def konkname(message: types.Message, state: FSMContext):
    if message.text == 'Вернуться назад':
        await bot.send_message(message.chat.id,text = 'Ви були повернуті в
адмінку',reply_markup=kb.admin)
        await state.finish()
        return
    chat_id = message.chat.id
    name = random.randint(1,99999)
    file_info = await bot.get_file(message.document.file_id)
    downloaded_file = await bot.download_file(file_info.file_path)
    src = 'C:\\Users\\s27_s\\Desktop\\quest-bot' + message.document.file_name
    with open(src, 'wb') as new_file:
        new_file.write(downloaded_file.getvalue())
    data = await state.get_data()
    tex = data.get('doctext')
    cursor.execute("SELECT user FROM users")
    userbase = []
    while True:
        row = cursor.fetchone()
        if row == None:

```



```

        break
    userbase.append(row)
if len(userbase) > 1:
    for z in range(len(userbase)):
        await bot.send_document(userbase[z][0], downloaded_file.getvalue(),caption =
tex)
        await bot.send_message(message.chat.id,text = 'Розсилка опитування була успішно
виконана',reply_markup = kb.menu)
        await state.finish()
        @dp.message_handler(state=cont.nazva)
async def konkname(message: types.Message, state: FSMContext):
    if message.text == 'Вернутися назад':
        await bot.send_message(message.chat.id,text = 'Ви були повернуті в
адмінку',reply_markup=kb.admin)
        await state.finish()
        return
    nazva = message.text
    await state.update_data(nazva=nazva)
    await bot.send_message(message.chat.id,text = 'Введіть порядковий номер
відповіді який бажаєте підрахувати')
    await cont.answer.set()
    @dp.message_handler(state=cont.answer)
async def konkname(message: types.Message, state: FSMContext):
    if message.text == 'Повернутись назад':
        await bot.send_message(message.chat.id,text = 'Ви були повернуті в
адмінку',reply_markup=kb.admin)
        await state.finish()
        return
    answer = message.text
    data = await state.get_data()
    quest = data.get('nazva')
    await state.update_data(answer=answer)
    try:
        allans = cursor.execute(f"SELECT Count(*) FROM answers WHERE answer =
{answer} and question = '{quest}").fetchone()[0]
        await bot.send_message(message.chat.id,text = f'Кількість відповідей: {allans}')
    except:
        await bot.send_message(message.chat.id,text = 'Опитувань з такою
назвою/відповіддю немає',reply_markup = kb.menu)
        await state.finish()
    @dp.callback_query_handler(lambda c: c.data)
async def ans(call: CallbackQuery,state: FSMContext):
    if call.data == 'bud1':
        await bot.send_message(call.message.chat.id,text = ' Учбовий корпус КНУТД ')
        await bot.send_location(call.message.chat.id,50.4311243,30.5360318)

```

```

if call.data == 'bud2':
    await bot.send_message(call.message.chat.id,text = ' Гуртожиток №2 ')
    await bot.send_location(call.message.chat.id,50.4142509,30.542162)
if call.data == 'bud3':
    await bot.send_message(call.message.chat.id,text = ' Гуртожиток №3 ')
    await bot.send_location(call.message.chat.id,50.4088274,30.5495272)
if call.data == 'bud4':
    await bot.send_message(call.message.chat.id,text = ' Гуртожиток №4 ')
    await bot.send_location(call.message.chat.id,50.408062,30.5500097)
if call.data == 'bud5':
    await bot.send_message(call.message.chat.id,text = ' Гуртожиток №5 ')
    await bot.send_location(call.message.chat.id,50.4330688,30.5356504)
if call.data == 'bud6':
    await bot.send_message(call.message.chat.id,text = ' Гуртожиток №6 ')
    await bot.send_location(call.message.chat.id,50.4205641,30.5275092)
if call.data == 'bud7':
    await bot.send_message(call.message.chat.id,text = ' Гуртожиток №7 ')
    await bot.send_location(call.message.chat.id,50.43019,30.5461563)
if call.data == 'bud8':
    await bot.send_message(call.message.chat.id,text = ' Гуртожиток №8 ')
    await bot.send_location(call.message.chat.id,50.5055811,30.4797678)
if call.data == 'con':
    await bot.send_message(call.message.chat.id,text = 'Введіть назву опитування за
яким бажаєте підрахувати відповіді',reply_markup = kb.admexit)
    await cont.nazva.set()
    if "answer_number" in call.data:
        await bot.send_message(call.message.chat.id,text = 'Ваша відповідь
записана',reply_markup = kb.admexit)
        usr = cursor.fetchone()
        if not usr:
            cursor.execute("INSERT INTO answers values ( :question,:username,
:answer);" ,
                {'question':que,
                'username': call.from_user.username,
                'answer': str(call.data).strip("answer_number_")})
            conn.commit()
    if call.data == 'opr':
        await bot.send_message(call.message.chat.id,text = 'Введіть назву
опитування',reply_markup = kb.admexit)
        await opos.question.set()
    if call.data == 'ras':
        await bot.send_message(call.message.chat.id,text = 'Виберіть тип
розсилки',reply_markup = kb.rasstypes)
    if call.data == 'tx':
        await bot.send_message(call.message.chat.id,text = 'Введіть текст який бажаєте

```

```

відправити',reply_markup = kb.admexit)
    await admad.ad.set()
    if call.data == 'ph':
        await bot.send_message(call.message.chat.id,text = 'Введіть текст',reply_markup
= kb.admexit)
        await getph.phtext.set()
    if call.data == 'doc':
        await bot.send_message(call.message.chat.id,text = 'Введіть текст',reply_markup
= kb.admexit)
        await getdoc.doctext.set()
    if call.data == 'kon':
        await bot.send_message(call.message.chat.id,text = 'Опишіть захід',reply_markup
= kb.admexit)
        await kadmad.kad.set()
    if call.data == 'act':
        await bot.send_message(call.message.chat.id,text = "Введіть ваше ім'я")
        await konkurs.name.set()
    if call.data == 'back':
        await bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id, text=f"Щоб знайти потрібну тобі інформацію,
вибери цікавий розділ:",reply_markup = kb.categories)
        if call.data == 'exback':
            await bot.send_message(call.message.chat.id,text = 'Щоб знайти потрібну тобі
інформацію, вибери цікавий розділ:',reply_markup = kb.categories)
        if call.data == 'struct':
            await bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id, text=f"В даному розділі Ви знайдете загальну
інформацію про університет",reply_markup = kb.structure)
        if call.data == 'hist':
            await bot.send_message(call.message.chat.id,text = 'Київський національний
університет технологій та дизайну ....
.\nДетальніше: https://knutd.edu.ua/university/history/')
            await bot.send_photo(call.message.chat.id,photo =
'https://imgur.com/a/2OXfc2J',reply_markup = kb.exback)
        if call.data == 'prof':
            await bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id, text=f"*Адреса:* 01011 м. Київ, вул.
Немировича-Данченка, 2, навчальний корпус 1, кімн. 1-0311\n\n*Телефон для
довідок:* (044) 256-29-62, (044) 284-83-08\n\n*e-mail:*
profsoyus@knutd.com.ua\n\n*Детальніше:*
https://knutd.edu.ua/university/profkom/",reply_markup = kb.back,parse_mode =
'Markdown')
        if call.data == 'vid':
            await bot.edit_message_text(chat_id=call.message.chat.id,

```

```

message_id=call.message.message_id, text=f"*Видавничий відділ навчально-
наукових матеріалів ....
/\n",reply_markup = kb.back,parse_mode="Markdown")
    if call.data == 'nauk':
        await bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id, text=f"Бібліотека КНУТД ....
...бібліотеки.\n\n*Адреса:* 01011 Київ, вул. Немировича-Данченка, 2, корпус 1,
к. 1-0275\n\n*Телефон для довідок:* +38044-256-29-02\n\n*e-mail:*
biblioteca@knutd.edu.ua\n\n*Сайт бібліотеки:*
http://lib.knutd.edu.ua/\n\n*Електронний каталог:*
http://biblio.knutd.edu.ua/catalog\n\n*Інституційний репозитарій:*
http://er.knutd.edu.ua/\n\n*Детальніше:*
https://knutd.edu.ua/university/library/",reply_markup = kb.back)
    if call.data == 'dost':
        await bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id, text=f"" Київський національний
університет...
/""",reply_markup = kb.back)
    if call.data == 'plat':
        await bot.send_message(call.message.chat.id,text = ""1) Положення про
надання платних послуг КНУТД (https://knutd.edu.ua/files/pay/polozh-pro-nadannya-
platnykh-poslug-KNUTD.pdf)
2) Оплата послуг через ПриватБанк
(https://next.privat24.ua/payments/form/%7B%22token%22:%22072ccac2-c067-4c2c-
a256-533d1d22%22%7D?lang)
3) Вартість проживання у гуртожитку КНУТД.....
4) Платні послуги науково-технічної бібліотеки
""")
        await bot.send_photo(call.message.chat.id,photo =
'https://imgur.com/a/tvgJKzP',caption = "",reply_markup = kb.exback)
    if call.data == 'kont':
        await bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id, text=f""Адреса: 01011 Київ, вул. ....
""",reply_markup = kb.back)
    if call.data == 'abit':
        await bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id, text=f"Тут Ви знайдете основну інформацію,
яка вам знадобиться при вступі до КНУТД",reply_markup = kb.abitype)
    if call.data == 's1':
        await bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id,
text=f"https://knutd.edu.ua/admissions_main/admissions_rules/",reply_markup =
kb.back,parse_mode = 'Markdown')
    if call.data == 's2':
        await bot.edit_message_text(chat_id=call.message.chat.id,

```

message\_id=call.message.message\_id, text=f"Відповідальний секретар: Власюк Тетяна Миколаївна\n\nАдреса: м. Київ, вул. Немировича-Данченка, 2, навчальний корпус 4, 2 поверх.\n\nТелефон для довідок: +38044-256-29-75\n\n\*e-mail: rk@knutd.edu.ua\n\n\*Додатково:

[https://knutd.edu.ua/admissions\\_main/admissions/](https://knutd.edu.ua/admissions_main/admissions/)",reply\_markup = kb.back)

if call.data == 's3':

await bot.edit\_message\_text(chat\_id=call.message.chat.id, message\_id=call.message.message\_id, text=f"[https://knutd.edu.ua/ekts/perelik-program/](\"https://knutd.edu.ua/ekts/perelik-program/\")",reply\_markup = kb.back,parse\_mode = 'Markdown')

if call.data == 's4':

await bot.edit\_message\_text(chat\_id=call.message.chat.id, message\_id=call.message.message\_id, text=f"[https://knutd.edu.ua/admissions\\_main/obradi-profesiju/567/](\"https://knutd.edu.ua/admissions_main/obradi-profesiju/567/\")",reply\_markup = kb.back)

if call.data == 's5':

await bot.send\_message(call.message.chat.id,text = f"Дотримуючись Державної програми ....

.....")

await bot.send\_photo(call.message.chat.id,photo = 'https://imgur.com/a/CME34TH',reply\_markup = kb.exback)

if call.data == 's6':

await bot.edit\_message\_text(chat\_id=call.message.chat.id, message\_id=call.message.message\_id, text=f"Вступникам на здобуття освітнього ступеня \"магістр\" ....

.....",reply\_markup = kb.back,parse\_mode = 'Markdown')

if call.data == 's7':

await bot.edit\_message\_text(chat\_id=call.message.chat.id, message\_id=call.message.message\_id, text=f"Центр доуніверситетської та індивідуальної освіти пропонує пройти підготовку до ЗНО за предметами:

1) українська мова та література

2) математика

3) хімія

4) біологія

5) фізика

6) історія України

7) іноземна мова

А також підготовку до вступних випробувань з живопису, рисунку та спецкомпозиції.

Форма проведення занять:

- вечірня (2-3 рази на тиждень з 16.00 до 19.00);
- вихідного дня (щосуботи з 9.00 до 15.00);
- дистанційна;
- заочна.

Початок занять з жовтня та лютого.

Короткотермінові курси (з 9.00 до 15.00)

```
""",reply_markup = kb.back,parse_mode = 'Markdown')
    if call.data == 's8':
        await bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id, text=f""""Відкрито Online реєстрацію на курси
центру доуніверситетської та індивідуальної освіти КНУТД:
https://goo.gl/forms/TbvMyPKp4QWc0yF52\nВипускники можуть отримати
додаткові бали при вступі на спеціальності університету, які потребують
особливої підтримки.\nПри цьому згідно з Умовами прийому до вищих
навчальних закладів України вага цих балів складає до 5% конкурсного
бала.\nДодатково: https://knutd.edu.ua/admissions\_main/zno/""",reply_markup =
kb.back,parse_mode = 'Markdown')
    if call.data == 's9':
        await bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id, text=f"https://knutd.edu.ua/files/pk/Nakaz-1274.pdf",reply_markup = kb.back,parse_mode = 'Markdown')
    if call.data == 's10':
        await bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id,
text=f"https://knutd.edu.ua/admissions\_main/exam\_programes/",reply_markup =
kb.back,parse_mode = 'Markdown')
    if call.data == 's11':
        await bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id, text=f"https://knutd.edu.ua/ekts/",reply_markup
= kb.back,parse_mode = 'Markdown')
    if call.data == 's12':
        await bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id, text=f""""1. Максимальний обсяг державного
замовлення для здобуття освітнього ступеня «Бакалавр» на базі повної загальної
середньої освіти: https://knutd.edu.ua/files/pk/obsiag\_state\_order/1.pdf\n2. Обсяг
державного замовлення на підготовку здобувачів вищої освіти за освітнім
ступенем «Бакалавр» : https://knutd.edu.ua/files/pk/obsiag\_state\_order/2\_2020.pdf\n3.
Обсяг державного замовлення на підготовку здобувачів вищої освіти за освітнім
ступенем «Магістр»»:
https://knutd.edu.ua/files/pk/obsiag\_state\_order/3.pdf""",reply_markup = kb.back)
    if call.data == 's13':
        await
bot.send_photo(call.message.chat.id,photo=f"https://imgur.com/a/9FZ0cGE",parse_mo
de = 'Markdown')
    if call.data == 's14':
        await bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id, text=f""""1) Рейтингові списки вступників:
https://knutd.edu.ua/admissions\_main/vstupna-kampania-2020/rating-lists/
2) Накази на зарахування: https://knutd.edu.ua/admissions\_main/vstupna-kampania-2020/nakazy-zarahovani/
""",reply_markup = kb.back,parse_mode = 'Markdown')
```

```
if call.data == 'kor':
    await bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id, text=f""Сайт університету:
https://knutd.edu.ua/
Модульне середовище освітнього процесу КНУТД:
https://msnp.knutd.edu.ua/index.php
Електронний журнал: https://stud.knutd.edu.ua/
Скринька довіри: https://knutd.edu.ua/dovira/
Facebook КНУТД: https://www.facebook.com/knutd.ua
Twitter КНУТД: https://twitter.com/knutd_ua
YouTube КНУТД: https://www.youtube.com/channel/UCiOoZirnyjdajKz7-eGWKIg
Instagram КНУТД: https://www.instagram.com/knutd.ua/
Linkedin КНУТД:
https://www.linkedin.com/school/kyivnationaluniversityoftechnologiesanddesignknutd/
Telegram КНУТД: https://t.me/KNUTD
Студентська Профспілка КНУТД: https://t.me/Knutd_Profkom
Instagram Профком КНУТД:
https://instagram.com/profkom_knutd?utm_medium=copy_link
""",reply_markup = kb.back)
    if call.data == 'k1':
        await bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id, text=f""Метою науково-інноваційної
політики Університету ....
""",reply_markup = kb.back)
    if call.data == 'k2':
        await bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id, text=f""Електронний архів Київського
національного університету технологій та дизайну наповнюється наступними
матеріалами: наукові публікації, автореферати дисертацій, монографії, тези
доповідей, матеріали конференцій.
Наукові публікації та магістерські роботи студентів розміщуються за
рекомендацією наукового керівника.
Детальніше: https://er.knutd.edu.ua/
""",reply_markup = kb.back)
    if call.data == 'k3':
        await bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id, text=f""Головним завданням відділу є....
....Детальніше: https://knutd.edu.ua/researchwork/rd_departments/
""",reply_markup = kb.back)
    if call.data == 'k4':
        await bot.send_message(call.message.chat.id,text = ""За рішенням Вченої ради
університету .....
....Адреса: 01011, м. Київ, вул. Немировича-Данченка, 2, корпус 1, к. 333
Тел./факс.: +38044-256-29-53
e-mail: aspirantura@knutd.edu.ua
```

Детальніше:

[https://knutd.edu.ua/researchwork/rd\\_departments/doctoral\\_studies\\_department/](https://knutd.edu.ua/researchwork/rd_departments/doctoral_studies_department/)

""")

```
await bot.send_photo(call.message.chat.id,photo =  
'https://imgur.com/a/sqz5qap',reply_markup=kb.exback)
```

```
if call.data == 'k5':
```

```
await bot.edit_message_text(chat_id=call.message.chat.id,  
message_id=call.message.message_id, text=f""""В КНУТД діє шість постійно діючих  
спеціалізованих вчених рад з правом прийняття до розгляду та проведення  
захисту дисертацій на здобуття наукового ступеня доктора та кандидата наук та  
разові спеціалізовані вчені ради з правом прийняття до розгляду та проведення  
захисту дисертації на здобуття ступеня доктора філософії.
```

Детальніше: [https://knutd.edu.ua/researchwork/academic\\_councils/](https://knutd.edu.ua/researchwork/academic_councils/)

```
""",reply_markup = kb.back)
```

```
if call.data == 'k6':
```

```
await bot.edit_message_text(chat_id=call.message.chat.id,  
message_id=call.message.message_id, text=f""""Наукові школи:
```

[https://knutd.edu.ua/researchwork/rd-art\\_schools/rd\\_schools/](https://knutd.edu.ua/researchwork/rd-art_schools/rd_schools/)

Мистецькі школи: [https://knutd.edu.ua/researchwork/rd-art\\_schools/art-schools/](https://knutd.edu.ua/researchwork/rd-art_schools/art-schools/)

```
""",reply_markup = kb.back)
```

```
if call.data == 'k7':
```

```
await bot.edit_message_text(chat_id=call.message.chat.id,  
message_id=call.message.message_id, text=f""""Відповідно до наказу Міністерства  
освіти і науки України від 24.11.2020 р. №1457 «Про проведення Всеукраїнського  
конкурсу студентських наукових робіт з галузей знань і спеціальностей у  
2020/2021 навчальному році», на базі Київського національного університету  
технологій та дизайну відбудеться другий тур Всеукраїнського конкурсу  
студентських наукових робіт з наступних галузей знань та спеціальностей:  
«Технології легкої промисловості», «Менеджмент інвестиційної та інноваційної  
діяльності», «Дизайн. Образотворче мистецтво, декоративне мистецтво,  
реставрація».
```

Детальніше: <https://knutd.edu.ua/researchwork/lp-contest/>

```
""",reply_markup = kb.back)
```

```
if call.data == 'k8':
```

```
await bot.send_message(call.message.chat.id,text = """"Наукове товариство....  
....популяризації досягнень сучасної науки.
```

""")

```
await bot.send_photo(call.message.chat.id,photo =  
'https://imgur.com/a/rTFmyuc',reply_markup=kb.exback)
```

```
if call.data == 'g1':
```

```
await bot.edit_message_text(chat_id=call.message.chat.id,  
message_id=call.message.message_id, text=f""""Відповідно до Закону України ...  
....Діяльність центру: https://knutd.edu.ua/students/job/dilo/
```

Підприємства-партнери КНУТД: <https://knutd.edu.ua/students/job/partnry/>



Наші випускники: <https://knutd.edu.ua/students/job/vypusk/>  
Законодавча база: <https://knutd.edu.ua/students/job/zakon/>  
Вакансії: <https://knutd.edu.ua/students/job/vakancij/>  
Адреса: 01011 м. Київ, вул. Немировича-Данченка, 2, навчальний корпус 1, ауд. 1-0246  
Телефони для довідок: +38044-280-07-74 (22-00 - внутрішній), +38097-30-90-470  
Email: [career\\_center@knutd.edu.ua](mailto:career_center@knutd.edu.ua)  
Facebook: <https://www.facebook.com/KNUTDcareercenter>

```
""",reply_markup = kb.back)
    if call.data == 'g3':
        await bot.send_message(call.message.chat.id,text = f""""Ти здобув почесне право
навчатися в одному із ....
....Детальніше: https://knutd.edu.ua/students/studprofcom/""")
        await bot.send_photo(call.message.chat.id,photo =
'https://imgur.com/a/C1o0akA',reply_markup=kb.exback)
    if call.data == 'g4':
        await bot.send_message(call.message.chat.id,text = f"Студентське
самоврядування ...
....Детальніше: https://knutd.edu.ua/students/studself-gov/""")
        await bot.send_photo(call.message.chat.id,photo =
'https://imgur.com/a/PU8v9sS',reply_markup=kb.exback)
    if call.data == 'g5':
        await bot.send_photo(call.message.chat.id,photo=f"https://imgur.com/a/I4b0ly4")
    if call.data == 'g6':
        await bot.send_message(call.message.chat.id,text = """"Білера О.В. Режисер,
актор театру і кіно ....
...., тощо на базі університету.
Детальніше: https://knutd.edu.ua/students/centrkm/
""")
        await bot.send_photo(call.message.chat.id,photo =
'https://imgur.com/a/g5FIN9Q',reply_markup=kb.exback)
    if call.data == 'g7':
        await bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id, text=f""""Діяльність Студентського будинку
моди ...
...Детальніше: https://knutd.edu.ua/students/fashionhouse/
""",reply_markup = kb.back)
    if call.data == 'g8':
        await bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id, text=f""""Відповідно до рішення ...
...Детальніше: https://knutd.edu.ua/students/vso/
""",reply_markup = kb.back)
    if call.data == 'g9':
        await bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id,
```

```
text=f"https://next.privat24.ua/payments/form/%7B%22token%22:%22072ccac2-c067-4c2c-a256-533d1d22%22%7D?lang",reply_markup = kb.back)
    if call.data == 'stud':
        await bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id, text=f"Тут Ви знайдете основну інформацію
необхідну під час навчання в КНУТД",reply_markup = kb.studtype)
    if call.data == 'nau':
        await bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id, text=f"Тут Ви знайдете основну інформацію
про наукову роботу КНУТД",reply_markup = kb.natype)
executor.start_polling(dp)
```

## ДОДАТОК Б. ІНТЕРФЕЙС ЧАТ-БОТА

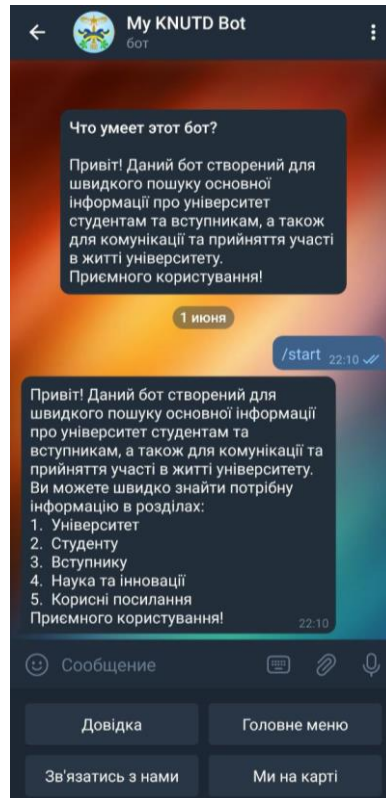
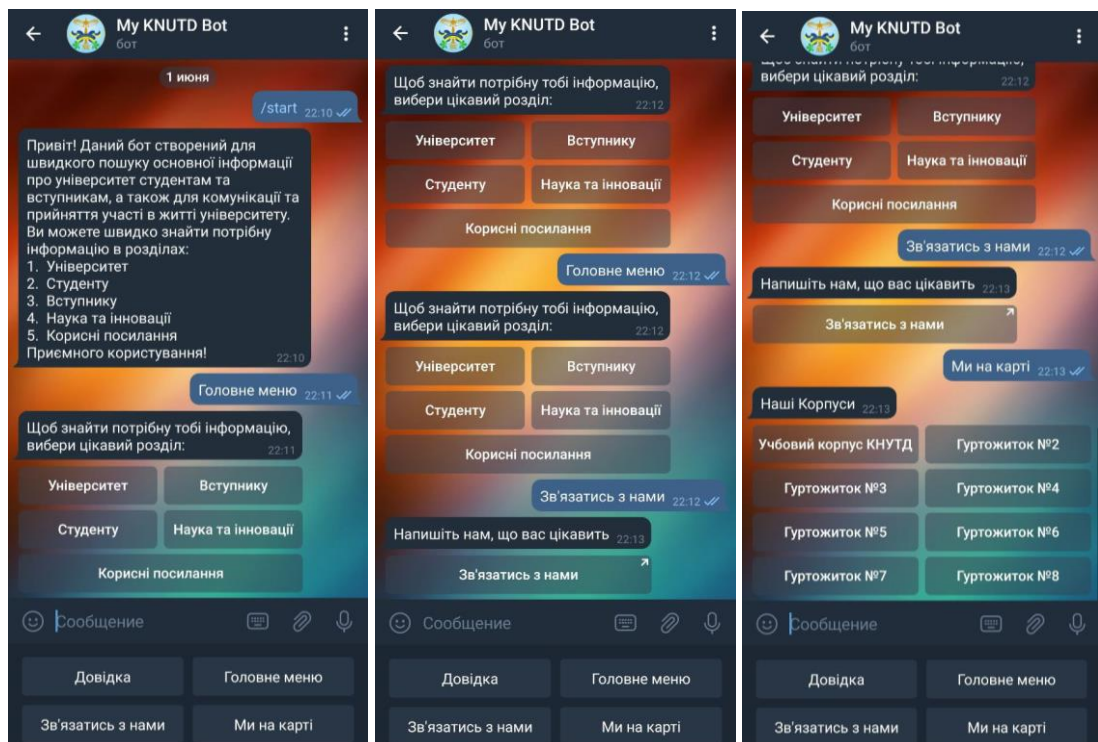


Рис. Б.1. Інтерфейс чат-бота на OS Android.



а)

б)

в)

Рис. Б.2. Розділи: а) «Головне меню»; б) «Зв'язатись з нами», в) «Ми на карті»

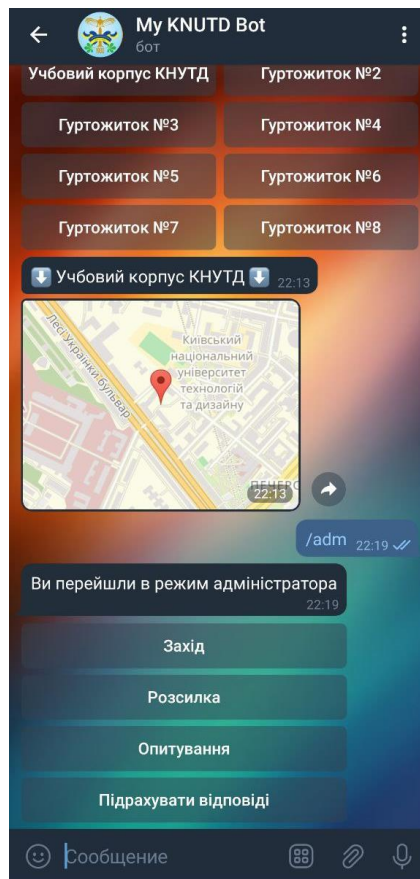


Рис. Б.3. Режим адміністратора