МІНІСТЕРСТВО ФІНАНСІВ УКРАЇНИ
ДЕРЖАВНИЙ ПОДАТКОВИЙ УНІВЕРСИТЕТ

ФАКУЛЬТЕТ ФІНАНСІВ ТА ЦИФРОВИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ

КВАЛІФІКАЦІЙНА РОБОТА
на тему:

РОЗРОБКА ТЕЛЕГРАМ-БОТУ РОЗКЛАДУ НАВЧАЛЬНИХ ЗАНЯТЬ

Здобувача вищої освіти першого (бакалаврського) рівня,
4 курсу, групи КБз-19-1, спеціальності 122 «Комп'ютерні науки»,
освітня програма «Інформаційні управляючі системи і технології (за галузями)»

_____ Вереса Андрія Ігоровича

Науковий керівник _____ Ратушняк Тетяна Володимирівна, к.ф.-м.н, доцент

Керівник освітньо-професійної програми _____ Федотова - Півень Ірина Миколаївна, к.т.н, доцент

Допущено до захисту за рішенням кафедри
протокол від «01» червня 2023 № 14

Завідувач кафедри _____ Філоненко Михайло Миколайович, к.ф.-м.н., доцент

Ірпінь – 2023

Реєстрація № 34 06.06.2023

Результати перевірки до захисту 01.06.2023

Результат захисту 5 (відмінно) 16.06.2023

Голова комісії: В.М. Франчук

Члени комісії:

М.М. Філоненко

О.В.Ніжегородцев

Є.В. Котух

БЛАНК ОЦІНЮВАННЯ ВИПУСКНОЇ РОБОТИ

Об'єкти контролю	Діапазон оцінки в балах	Фактична кількість балів	Примітки	Дата, П.І.Б.: керівника, голови комісії членів комісії,секретаря, (підписи)
ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ КОМІСІЄЮ ПО ЗАХИСТУ				
1. Зміст випускної роботи (якість теоретичного матеріалу, його зв'язок з практичним, вміння представляти матеріал в логічній послідовності, наявність висновків та пропозицій)	від 0 до 30	29		16.06.2023 Франчук В.М. Філоненко М.М Ніжегородцев В.О. Котух Є.В. Суховецька Л.П.
2. Дотримання вимог щодо оформлення	від 0 до 10	8		
3. Вчасне виконання роботи на всіх етапах у встановлені кафедрою терміни	від 0 до 5	4		
4. Рецензент	від 0 до 5	4		
РАЗОМ БАЛІВ	від 0 до 50	45		

Оцінювання кваліфікаційної роботи науковим керівником - 45 балів.

ПЕРЕВЕДЕННЯ ОЦІНКИ ЗА 100-БАЛЬНОЮ ШКАЛОЮ ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалоюЄКТС	Оцінка за національною шкалою	Дата	П.І.Б.: голови, членів комісії, секретаря, (підпис)
90	A	5 (відмінно)	16.06.2023	Франчук В.М. Філоненко М.М Ніжегородцев В.О. Котух Є.В. Суховецька Л.П.

ПРАВИЛА ПЕРЕВЕДЕННЯ ОЦІНОК ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою
90-100	A	5 (відмінно)
80-89	B	4 (добре)
70-79	C	
60-69	D	3 (задовільно)
50-59	E	
35-49	FX	2 (незадовільно)

Факультет фінансів та цифрових технологій
Кафедра комп'ютерних та інформаційних технологій і систем
Спеціальність 122 «Комп'ютерні науки»
Освітня програма «Інформаційні управляючі системи і технології (за галузями)»

Затверджую
Завідувач кафедри
_____ Філоненко М. М
«04» 05 2022

Завдання на випускну кваліфікаційну (дипломну) роботу здобувачу вищої освіти
Вересу Андрію Ігоровичу
(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної (дипломної) роботи Розробка телеграм-боту розкладу навчальних занять
Затверджена наказом ректора від «04» травня 2023 № 515

2. Строк здачі закінченої роботи «01» червня 2023

3. Цільова установка: Дослідження сучасних технологій розробки чат-ботів для месенджерів та розробка чат-бота розкладу навчальних занять для Telegram-месенджера.
Мета роботи: Розробити чат-бот розкладу навчальних занять для Telegram-месенджера
Об'єкт дослідження: Технології розробки чат-ботів для месенджерів.
Предмет дослідження: Технології розробки чат-ботів для месенджера Telegram.
База дослідження: ТОВ «Віста»

4. Вихідні дані до роботи: Інформаційні дані та процеси модуля «Розклад» автоматизованої системи управління Державного податкового університету

5. Перелік графічного матеріалу: Бізнес-класифікація чат-бот додатків, Технічна класифікація чат-ботів, Приклад чат-бота в форматі віджету, приклади ботів, Приклад дизайну розкладу на сайті університету, Способи взаємодії з ботом, Поле пошуку @BotFather, Запуск активації боту, Початок роботи із ботом BotFather та доступні команди, Додавання імені боту, Налаштування аватарки, Встановлення команд, Приклад коду та синтаксису, Ініціалізація токену, Структура база даних, Наповнена база даних, Початок роботи з ботом, Результат розгортання чат-бота

6. Зміст випускної кваліфікаційної (дипломної) роботи

Вступ
Розділ 1. Аналіз предметної області. Теоретичні положення розробки чат-ботів
1.1. Постановка проблеми
1.2. Загальна концепція чат-боту
1.3. Сучасні месенджери
1.4. Аналіз конкурентних систем оповіщення про розклад занять
1.5. Деталізація завдань розробки
Розділ 2. Програмна реалізація бота
2.1. Обгрунтування інструментарію розробки
2.2. Вибір засобів розробки чат-боту та бібліотек
2.3. Розробка чат-бота
Розділ 3. Практичне застосування чат-боту
3.1. Тестування чат-бота
3.2. Приклади застосування та операційний аналіз
3.3. Розміщення чат-бота на хмарному сервері Heroku
Висновки
Список використаних джерел

7. Календарний план виконаної роботи

№ п/п	Назва етапів кваліфікаційноїроботи	Строк виконання етапів роботи	
		За планом	Факти
1	2	3	4
1	Вибір теми, підбір літературних джерел, формування структури кваліфікаційної роботи	До 15.11.2022	15.11.2
2	**І розділ** (розкрити сутність і стан дослідження наукової проблематики роботи, після чого навести її загальну характеристику; огляд літератури за темою і вибір конкретних напрямів досліджень)	06.02.2023	06.02.2
3	**ІІ розділ** (аналіз експериментальної частини про методики досліджень)	31.03.2023	31.03.2
4	**ІІІ розділ** (проведені теоретичні і (або) експериментальні дослідження та проектні рішення)	05.05.2023	05.05.2
5	**Висновки**	16.05.2022	16.05.2
6	Доопрацювання роботи з урахуванням зауважень наукового керівника	01.06.2023	01.06.2
7	Отримання відгуку керівника	08.06.2023	08.06.2
8	Отримання зовнішньої рецензії	11.06.2023	11.06.2

8. Дата видачі завдання «16» листопада 2022

9. Керівник випускної кваліфікаційної (дипломної) роботи _____ Ратушняк Т.В.
(підпис) (прізвище, ініціали)

10. Керівник освітньо-професійної програми _____ Федотова-Півень І.М
(підпис) (прізвище, ініціали)

11. Завдання до виконання отримав здобувач вищої освіти _____ Верес А.І.

АНОТАЦІЯ

У кваліфікаційній роботі вивчається взаємодія користувача з існуючими системами обміну миттєвими повідомленнями. Подано огляд ключових факторів взаємодії та алгоритмів для чат-ботів. Описує основні принципи створення чат-бота. Представлено актуальність розробки Telegram бота та його застосування в студентському житті. Розглядаються проблеми проєкту, які вирішуються шляхом розробки чат-бота. Детально описано практичну частину роботи, результати роботи, обґрунтовано вибір використаних методів.

Ключові слова: телеграм бот, розклад занять, інформаційні технології, розробка продукту.

Рік виконання кваліфікаційної роботи: 2023

Рік захисту роботи: 2023

ANNOTATION

In the qualifying work, user interaction with existing instant messaging systems is studied. An overview of the key interaction factors and algorithms for chatbots is provided. Describes the basic principles of creating a chatbot. The relevance of Telegram bot development and its application in student life is presented. The problems of the project are considered, which are solved by developing a chatbot. The practical part of the work, the results of the work are described in detail, the choice of the used methods is justified.

Key words: telegram bot, occupation layout, information technologies, product development.

Year of completion: 2023

Year of thesis defense: 2023

ЗМІСТ

ВСТУП **5**

РОЗДІЛ 1. АНАЛІЗ ОБ'ЄКТУ ДОСЛІДЖЕННЯ **7**

1.1. Постановка проблеми *7*

1.2. Загальна концепція чат-боту та тенденції **8**

1.3. Сучасні месенджери **10**

1.4. Аналіз конкурентних систем оповіщення про розклад занять **13**

1.5. Деталізація завдань розробки **15**

РОЗДІЛ 2. ПРОГРАМНА РЕАЛІЗАЦІЯ БОТА **20**

2.1 Обгрунтування інструментарію розробки *20*

2.2 Вибір засобів розробки чат-боту та бібліотек *28*

2.3 Розробка чат-бота *34*

РОЗДІЛ 3. ПРАКТИЧНЕ ЗАСТОСУВАННЯ ЧАТ-БОТУ **40**

3.1. Тестування чат-бота *40*

3.2. Приклади застосування та операційний аналіз *45*

3.3. Розміщення чат-бота на хмарному сервері Heroku *45*

ВИСНОВКИ **47**

ВИКОРИСТАНА ЛІТЕРАТУРА **49**

ВСТУП

Системи віртуального спілкування з клієнтами на основі штучного інтелекту наразі набувають широкого поширення. Ці системи об'єднує швидкість доступу до інформації, можливість для багатьох клієнтів одночасно працювати в системі, обмінюватися інформацією, взаємодіяти для вирішення будь-яких проблем, спілкуватися з клієнтами та бізнес-партнерами, виконувати аналітичні завдання. Пошук і збір необхідної інформації Одним із варіантів віртуальних комунікаційних систем, які включають елементи штучного інтелекту, є чат-боти [1].

Чат-бот представляє собою програму, що виконує дії, які завчасно прописуються в алгоритмі з можливістю імітувати справжню розмову. Чат-боти не вимагають довгострокового налаштування, великий об’єм пам’яті. Користувачу не потрібно витрачати купу часу, щоб відкрити браузер, зайти в певний додаток, а достатньо відкрити зручний для себе месенджер (наприклад, Viber, Telegram, Facebook Messenger, WhatsApp) та перейти безпосередньо до діалогу з ботом. До речі, саме з інтенсивним збільшенням чисельності користувачів месенджерів пов’язане загальне зростання інтересу до чат-ботів [1].

На сьогодні існує багато чат-ботів у різних сферах людської діяльності. Наприклад, NovaPoshtaBot – неофіційний бот «Нової пошти», який відстежує відправлення, допомагає знайти найближче відділення і повідомляє про зміну статусу посилки; Weatherman_bot – бот, який розповідає про погоду та за його допомогою можна подивитися прогноз на день або погодинно. Поки мало охоплена така діяльність як робота з розкладом, наприклад сеансів фільмів в кінотеатрах, розклад руху поїздів і літаків, розклад занять в школі і у вищих навчальних закладах [2].

Актуальність роботи диктує ситуація на ринку інформаційних технологій.З кожним роком його поповнюють нові програмні продукти, веб-сервіси та мобільні додатки, за допомогою яких можна побудувати не тільки комунікації, ай сприяти зміцненню бренду університету, як освітнього центру. Чат-бот дозволить

спростити щоденні рутинні завдання студента, такі як отримання інформації про розклад, дзвінки, зміни розкладу та інше. Головною перевагою серед інших методів отримання інформації є те, що всі можливості поєднуються на платформі єдиного месенджера.

Мета полягала в тому, щоб створити чат-бот для факультету ІТ та прикладних технологій, щоб студенти могли легко знаходити інформацію та відповідь на запитання за мінімальний час.

Об'єктом дослідження є розробка веб-сайту з використанням Python та Telegram API.

Предмет дослідження - сучасні системи, представлені чат-ботами, які вважаються найкращими за своїм наповненням, швидкістю та з використанням AI.

Практичне значення: чат-бот буде використовуватися протягом періоду навчання для популяризації університету серед абітурієнтів, налаштування підтримки чат-бота та робота з ним зменшить велику кількість запитань студентів навчального закладу.

Структура роботи: У вступі наведено актуальність розробки телеграм бота та застосування його в студентському житті. Перший розділ курсової роботи присвячений проблемам проєкту, які вирішує розробка чат-боту. Другий розділ акцентує увагу на алгоритмах, покладених в основу чат-ботів. Описано принципи, які є основними для створення чат-ботів. У третьому розділі було розглянуто детальний опис практичної частини роботи. У висновку описаний результат проведеної роботи, а також обґрунтовано вибір використаних технологій.

РОЗДІЛ 1. АНАЛІЗ ОБ'ЄКТУ ДОСЛІДЖЕННЯ

1.1. Постановка проблеми

У вищих навчальних закладах розклади відіграють важливу роль у повсякденному житті студентів і співробітників, повідомляючи про час і місце навчання. Коли розклад створюється в динамічному середовищі, такому як школа, в ньому можуть відбуватися зміни. Виникає питання, як оперативно інформувати учнів про розклад та його зміни. Наразі система інформування про розклад є традиційною і розміщується на паперових носіях: на дошках оголошень для учнів, у кабінетах для вчителів або на веб-сайті навчального закладу. Студентам не завжди зручно користуватися паперовими носіями, які не доступні 24 години на добу. Крім того, фотографії, зроблені мобільними телефонами, незручні тим, що ними важко маніпулювати, легко загубити фотографію розкладу серед багатьох фотографій на мобільному телефоні, а також студенти не можуть отримувати сповіщення про зміни в розкладі. Доцільно розробити систему сповіщень з використанням поширених онлайн-додатків та сервісів, щоб забезпечити швидке отримання студентами інформації про розклад. Як зазначено в *We Are Social і Hootsuite* [3], частка користувачі Інтернету, які використовують мобільні телефони, і частка користувачів мобільних месенджерів зростає в усьому світі. Відповідно до дослідження [4] компанії мобільного маркетингу LEAD9 та Київського міжнародного інституту соціології, найпопулярнішими месенджерами серед українців є Viber, Facebook, Skype та Telegram. Розробка систем пошуку розкладу з використанням Facebook та Skype є месенджерів не є доцільним, оскільки лише невелика кількість студентів користується цими месенджерами на своїх мобільних телефонах. Тому доцільніше використовувати функціонал таких месенджерів, як Viber та Telegram.

1.2. Загальна концепція чат-боту та тенденції

Чат-боти використовуються в багатьох сферах послуг як постачальники розмовної інформації. Ці чат-боти розгортаються як у державному, так і в приватному секторах. Індійські залізниці, одна з найбільших залізничних

компаній у світі, використовує чат-бота під назвою ask DISHA для надання інформації в розмовній формі. У приватному секторі послуги чат-ботів значно зросли. Просування товарів, демонстрація продукції та обслуговування клієнтів. Навіть у банківському секторі чат-боти надають банківські послуги та відповідають на запитання. У соціальних мережах чат-боти відповідають на будь-які повідомлення без участі людини. Чат-боти отримують вхідні дані у вигляді мовлення, перетворюють їх на текст і генерують відповіді у формі текст-мовлення [5].

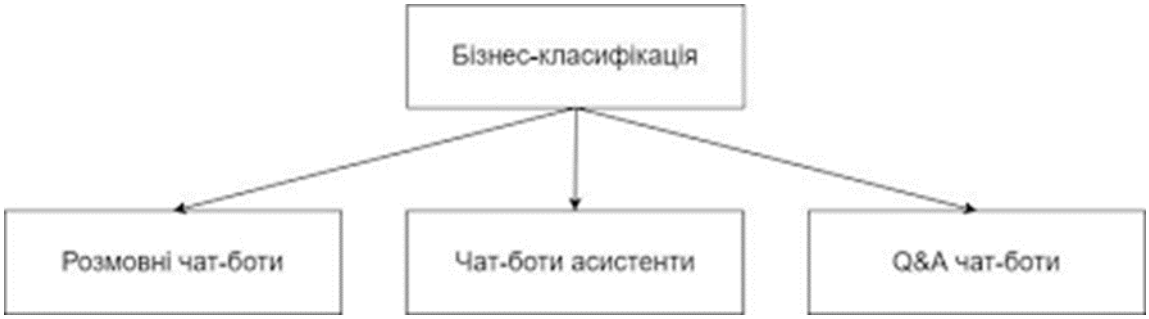


Рис.1.1 – Бізнес-класифікація чат-бот додатків

Багато є ботів для охорони здоров’я та консультування людей, які потребують консультації як хворі на цукровий діабет, алекогольнозалежні [6], з психологічними проблемами [6, 7], люди похилого віку, які страждають на втрату пам'яті (деменція)[8].

Чат-бот – це запрограмований комп’ютер, який використовує автоматичну відповідь за допомогою стилю мультимедіа, тобто тексту, зображень, посилань, відео тощо. Чат-боти доступні у двох варіантах [9]:

1. На основі правил,

2. На основі AI

Чат-боти на основі правил залежать від правил і умов. Це не є надійним у всіх випадках, коли користувачі задають запитання, що питання не мають умов чи правил. Другий тип чат-ботів підхід на основі штучного інтелекту або наміру — це машинне навчання. Чат-бот навчається на основі обробки природної мови (NLP) із наборами даних, які є діалоговими вікнами перетворення виділити

комбінацію розмови, включаючи наміри, контекст, сутність [9]. Багато сучасних інструментів, які можна використовувати для цього підходу, наприклад IBM Watson, Api. Як правило, в основі реалізація Chatbot вимагає відповідних шаблонів введення даних користувача та створення певної відповіді. На даний момент існує багато способів розробити чат-бот без кодування, яке робить розвиток чат-бота досить простим, зручним і швидким процесом.



Рис 1.2 – Технічна класифікація чат-ботів

Чат-боти - це програми, які працюють у месенджері (системі обміну миттєвими повідомленнями) і можуть відповідати на запитання або ставити власні запитання. Водночас чат-боти можуть виступати в ролі цифрових асистентів, виконуючи команди всередині месенджера, надаючи рекомендації або здійснюючи певний пошук за заданими користувачем параметрами. Чат-боти також можуть бути частиною інтерфейсу веб-сайту і бути безпосередньо доступними для користувачів через віджети (*рис. 1.3*).

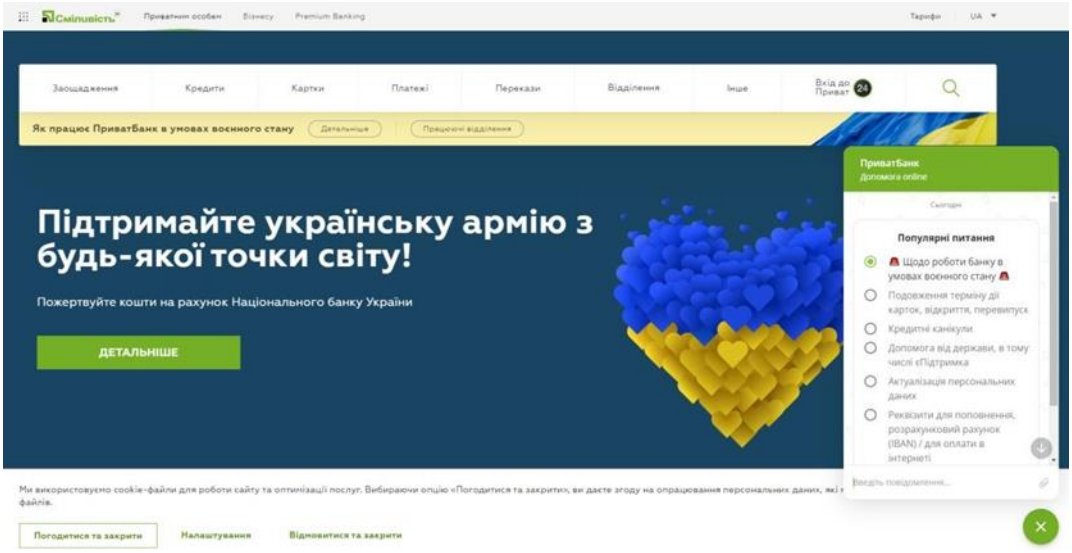


Рис. 1.4 – Приклад чат-бота в форматі віджету

Таким чином, чат-боти допомагають користувачам з'ясувати їхні потреби та задовольнити їх, використовуючи найбільш природну форму спілкування, мову, яку розуміє користувач, тобто взаємодіючи вербально або невербально з людиною через онлайн-середовище, пропонуючи рішення та вирішуючи проблеми користувача. Чат-боти - це програми, які допомагають користувачам вирішувати їхні проблеми, пропонуючи рішення шляхом вербальної або невербальної взаємодії з людиною в онлайн-середовищі. Чат-боти можуть автоматизувати завдання, які можна розділити на інші сегменти, такі як підтримка клієнтів, опитування та планування зустрічей. Наприклад, чат-боти успішно допомагають користувачам вирішувати проблеми, пов'язані з конкретними брендами, відповідати на поширені запитання та надавати клієнтам інформацію про продукти та послуги. Крім того, боти можуть допомагати клієнтам у різних ситуаціях, таких як бронювання, перевірка наявності товару або допомога з планами подорожей.

1.3. Сучасні месенджери

Ймовірно причиною великої популярності месенджерів є зміни у сфері мобільного Інтернету: високі швидкості, нижчі ціни та широке використання смартфонів. Згідно з дослідженням statista.com станом на квітень 2020 р. кількість активних користувачів у найпопулярніших представлені месенджери. Явний лідер є WhatsApp із загальною кількістю користувачів близько 1,5 мільярда. За ним іде Facebook Messenger з 1,3 мільярда активних користувачів на місяць. Очевидно, усі месенджери в цьому списку відрізняються [10].

Однак у них є спільні особливості, які привели до їх успіху. Майже кожен одна з них має зручний, сучасний і зрозумілий користувацький інтерфейс. Кожен з месенджерів також підтримує групові чати. Деякі з них підтримують канали частково копіювати функціональність чатів, за винятком того, що тільки

користувачі з відповідними правами доступу мають можливість публікувати повідомлення.

Майже всі месенджери пропонують голосові дзвінки. Порівняйте WhatsApp, найбільший месенджер за кількістю активних користувачів, з Telegram, месенджером, що розвивається найшвидше. З січня 2015 року по грудень 2017 року відбулося стрімке зростання активної користувацької бази WhatsApp. За цей період кількість активних користувачів подвоїлася з 700 мільйонів до 1,5 мільярда за лічені місяці. Це означає, що кожна п'ята людина на планеті користувалася WhatsApp щомісяця. Telegram є найпопулярнішим месенджером у світі; з лютого 2016 року по грудень 2017 року кількість активних користувачів зросла на 80% (*рис. 1.5*) [11].

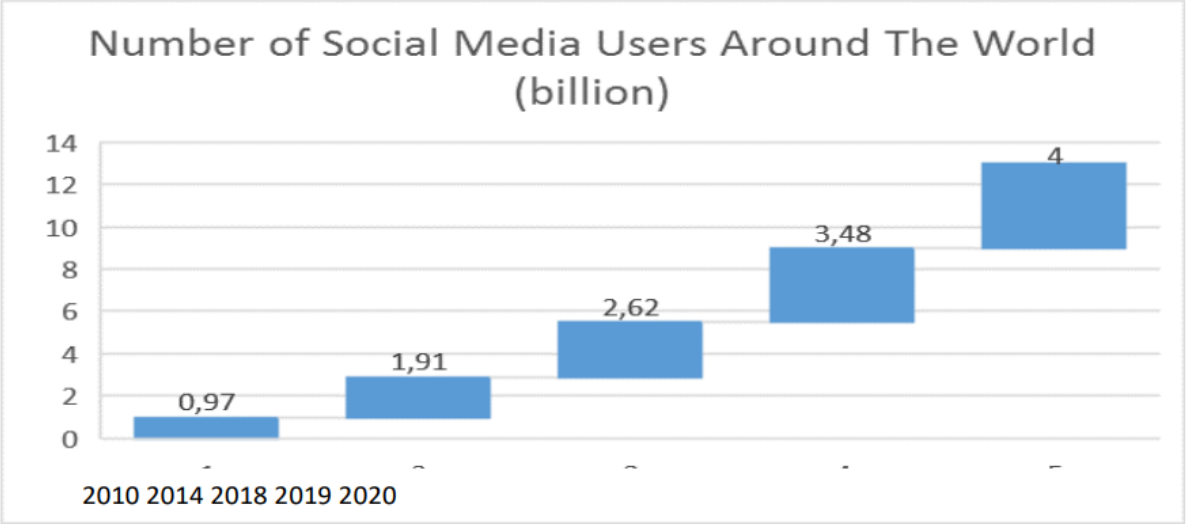


Рис. 1.5 Глобальні соціальні мережі за кількістю користувачів 2020

За той самий період у WhatsApp цей показник зріс на 50%. Telegram можна назвати месенджером, що найшвидше зростає: станом на квітень 2020 року він уже перетнув позначку в 200 мільйонів активних користувачів. Завдяки своїй постійній присутності в житті користувачів соціальні медіа мають вирішальний соціальний вплив (рис. 1.6). Варто також зазначити, що Telegram пропонує секретні чати, під час яких майже неможливо отримати доступ до контенту. Крім того, порівняно з WhatsApp, Telegram працює швидше і може передавати будь-які файли, тоді як WhatsApp дозволяє користувачеві зайти лише на один пристрій, а

згідно з умовами надання послуг додатку, користувача можуть заблокувати за часту зміну пристрою.

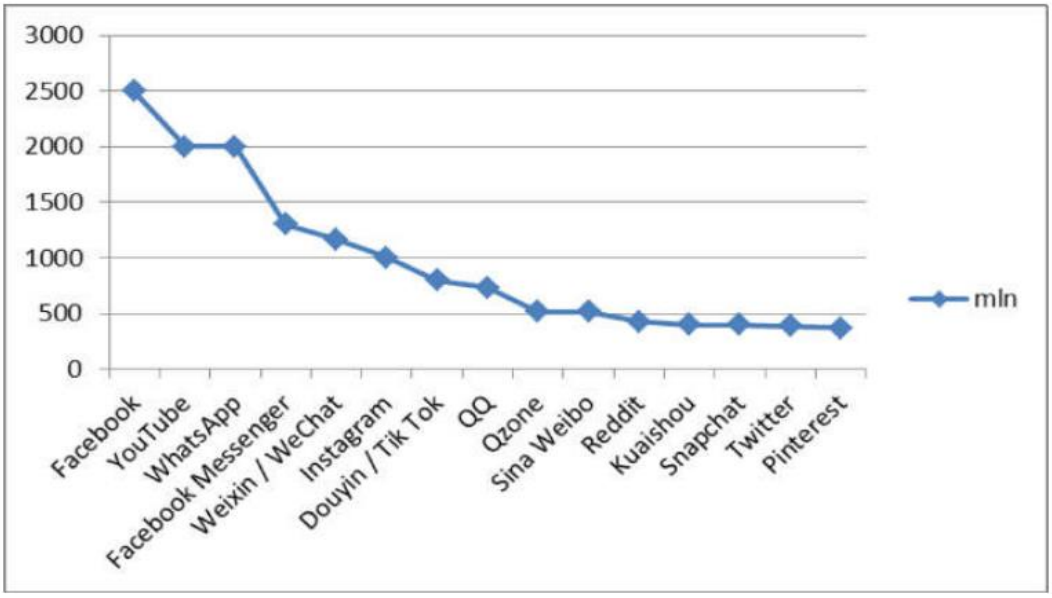


Рис. 1.6 Найпопулярніші соціальні мережі світу станом на квітень 2020 року, за кількістю активних користувачів (у мільйонах)

За останні кілька років традиційні телефонні дзвінки стали менш популярними, ніж обмін миттєвими повідомленнями або онлайн-послуги. Крім того, багато людей і компаній віддають перевагу тексту або мультимедіа повідомлення через дзвінки. Месенджери стали важливою частиною сучасного онлайн-спілкування. Користувачі віддають перевагу зручному, дешевому і швидкому зв'язоку, тому вони вибрати месенджери. Telegram має найвищий темп зростання кількості активних користувачів, а отже й перспективи для зростання аудиторії для розробленого сервісу [10].

1.4. Аналіз конкурентних систем оповіщення про розклад занять

Чат-бот університету буде розроблений українською мовою, тому при аналізі існуючих рішень були обрані найпопулярніші та найбільш затребувані боти в українських компаніях та продуктах. Серед найпопулярніших функцій, які можуть виконувати боти, - спілкування та взаємодія з іншими мобільними додатками та сервісами.

Прикладом такого бота є RailwayBot (*рис. 1.7*), який допомагає моніторити, шукати та купувати залізничні квитки. За допомогою цього бота ви можете налаштувати моніторинг, вказавши конкретний поїзд, дату, час і тип місця. Коли квитки стають доступними, бот негайно надсилає повідомлення. Алгоритм схожий на купівлю квитків на сайті uz.booking.gov.ua. Бот реалізований на основі кнопок для введення пунктів відправлення та прибуття [12].

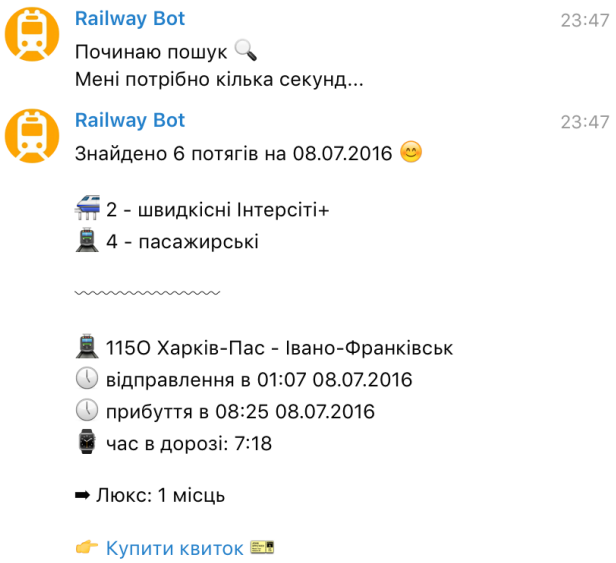


Рис. 1.7 – RailwayBot

PrivatBankBot (*рис. 1.8*) - це бот від ПриватБанку, який спрощує платіжну систему. Він дозволяє користувачам легко надсилати гроші без необхідності відкривати додаток. Щоб надіслати гроші іншому користувачеві, у нього також має бути активований бот [13].

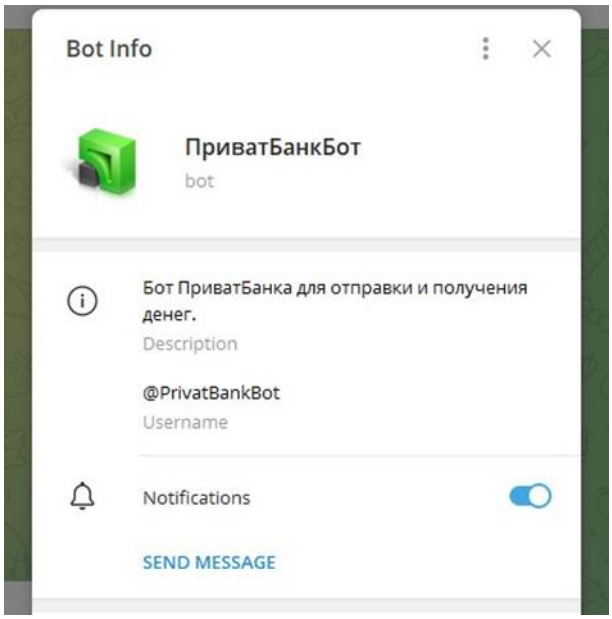


Рис. 1.8 – PrivatBankBot

NovaPoshtaBot (*рис. 1.9*) — неофіційний бот «Нової пошти», який відстежує відправлення, допомагає знайти найближче відділення і повідомляє про зміну статусу посилки.

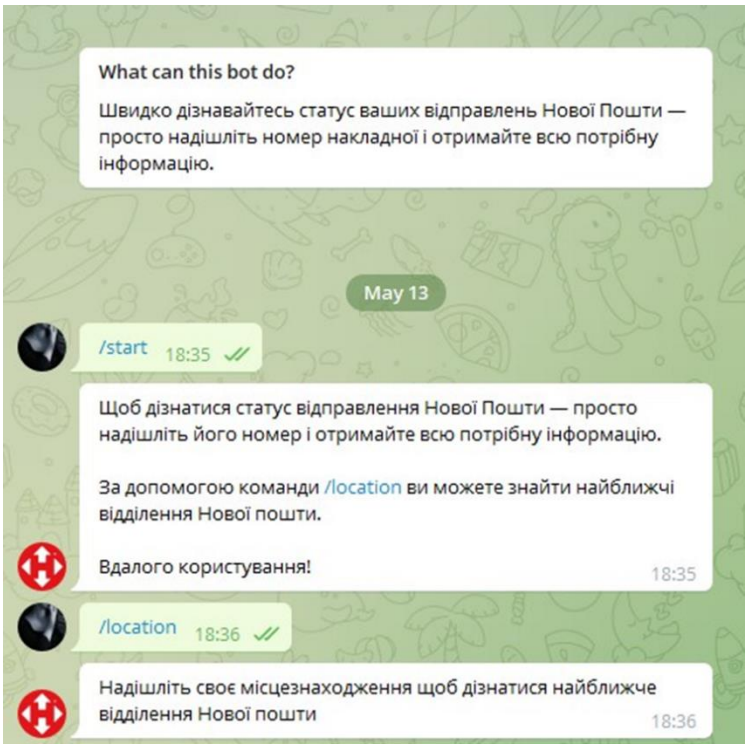


Рис. 1.9 – NovaPoshtaBot

Більшість ресторанів та кафе мають власних ботів для замовлення та доставки через популярний месенджер. Проаналізувавши готові рішення,

ключовою вимогою була швидкість роботи бота. Якщо затримка перевищує кілька секунд, боти втрачають перевагу над пошуком у Google або відкриттям додатку; боти зі штучним інтелектом мають перевагу над скриптовими ботами, оскільки вони ближчі до живого спілкування. Водночас гібридні боти більш зручні для користувача. Пропозиція готових варіантів спрощує комунікацію і вимагає менше дій від клієнта.

1.5. Деталізація завдань розробки

Основною метою кваліфікаційної (дипломної) роботи було створення чат-бота з розкладом занять для інформування студентів за допомогою месенджера Telegram. Перед початком проектування та розробки Telegram-бота необхідно було визначити та описати основні вимоги. У попередньому розділі було проаналізовано цільову область та вивчено існуючі подібні продукти, щоб переконатися в доцільності розробки.

Сучасні методи викладання та надання інформації кінцевим користувачам постійно потребують вдосконалення в усіх сферах життя - від продуктових магазинів до великих державних компаній: те, що було актуальним п'ять років тому, вже не користується популярністю або має меншу читацьку аудиторію (рис. 1.10). Для того, щоб донести до студентів актуальну інформацію за короткий проміжок часу, наприклад, про зміни у розкладі, важливі події (спонтанна здача крові, дні народження університету) тощо, необхідне негайне інформування. Єдиний спосіб донести інформацію до студентів не завжди зручний і часто займає більше часу, ніж це відбувається насправді. Тому одним із головних викликів у цій сфері є оновлення способу подачі інформації на сучасних комунікаційних платформах у суспільстві [14].

Номер пари	Початок заняття	Закінчення заняття
1 зміна		
1 пара	8-30	9-50
2 пара	10-10	11-30
3 пара	11-50	13-10
Перерва з 13.10-14.00 Санітарна обробка приміщень		
2 зміна		
4 пара	14-00	15-20
5 пара	15-35	16-55
6 пара	17-10	18-30
7 пара	18-45	20-05
Санітарна обробка приміщень		

Рис. 1.10 – Приклад формату розкладу на сайту університету

Чим більше студентів реєструється та навчається, тим важливіше колективно охопити всю аудиторію за короткий час. Студенти - це дорослі, досить активні учні, які використовують у повсякденному житті всі останні інновації, в тому числі й месенджер Telegram. Як правило, не всі постійно слідкують за сайтом свого університету та додатковими офіційними джерелами, а всі необхідні та актуальні дані потребують централізації. Тому метою даного дослідження є створення проекту інформаційної підтримки студентів на базі месенджера Telegram. Розроблена система повинна забезпечити зручний сервіс для студентів, які навчаються в університеті.

Виходячи з вищезазначених цілей, слід розробити Telegram-бота, який

а) бути в курсі змін у розкладах та форматі, в якому ці дані зберігаються

б) зберігав отримані дані у спеціально розробленій базі даних, оптимізованій для зберігання таких даних

в) зберігати дані кожного користувача та розклад кожної підгрупи у визначеному порядку

г) відстежувати поточний час та налаштовувати сповіщення для користувачів бота

є) отримувати інформацію про розклад дзвінків;

Оскільки розклад дзвінків - це параметр, який рідко змінюється, зберігання інформації про дзвінки в базі даних позбавляє від необхідності витрачати зайвий час на її пошук знову і знову. Те ж саме стосується інформації про розклад занять, оскільки щоразу ця інформація підтягується з сайту університету і дані оновлюються.

Телеграм-чат - це можливість миттєвого обміну повідомленнями між учасниками конференц-зв'язку, а телеграм-канал - це односторонній спосіб отримання, де інформація надсилається від імені адміністратора. Telegram-боти - це також учасники конференції, що складаються виключно з учасників і ботів, які надають інформацію за запитом користувача; Telegram-боти призначені для отримання інформації шляхом вибору певного шляху з дерева ланцюжків запитань і відповідей. У цьому проєкті об'єктом дослідження є Telegram-боти. Це пов'язано з тим, що Telegram-боти наразі є одним з найпоширеніших способів отримання будь-якої інформації, починаючи від текстової та закінчуючи відеоматеріалами.

У зв'язку з великою кількістю вхідних даних було вирішено розділити проект на етапи. Першим етапом було визначення структури веб-порталу, тобто способу отримання інформації користувачами. Перший етап полягав у визначенні структури веб-порталу, тобто типу структури, яка б дозволила користувачам здійснювати більш лінійну навігацію з головної сторінки, мережеву структуру, яка б дозволила користувачам переходити зі сторінки на сторінку без суворого порядку, або ієрархічну структуру, де на головній сторінці міститься коротка інформація про структурні підрозділи. Другий крок - вибір моделі побудови веб-сайту, тобто моделі водоспаду, спіралі або MSF (Microsoft Solutions Framework). Заключним етапом визначаються технології розробки web-порталу, а саме технічна частина back-end і користувальницька froent-end.

Основні елементи інтерфейсу чат-бота універсальні і притаманні будь-якому додатку для обміну повідомленнями; як описано в документації Telegram API, Telegram-боти можуть взаємодіяти з сервером двома способами:

1. getUpdates - тягнути: бот завжди звертається до сервера Telegram і перевіряє наявність нових повідомлень;

2. setWebhook - push: при отриманні нового повідомлення сервер Telegram надсилає його боту.

Різниця між цими двома методами проілюстрована на *рис. 1.11*.

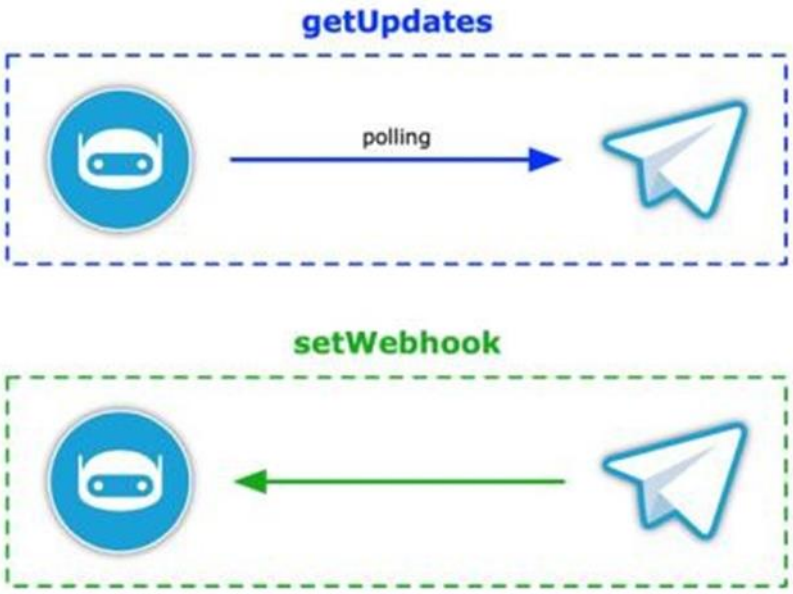


Рис. 1.11 – Способи взаємодії з ботом

Другий спосіб (setWebhook), безумовно, більш прийнятний для всіх учасників процесу. Для нього потрібен веб-сервер або його еквівалент, оскільки повідомлення з Telegram потрібно отримувати на стороні бота. Також потрібен SSL-сертифікат, щоб гарантувати, що комунікація відбувається через захищений HTTPS-канал. Це необхідна і невід'ємна частина обміну через веб-хук.

Програмне забезпечення включає в себе веб-сервер і Telegram-бот. Основні ідеї, реалізовані в додатку, включають

1. обробка повідомлень користувачів в режимі реального часу;

2. негайне представлення користувачеві можливих варіантів відповідей; та

3. обробка рішення користувача в режимі реального часу та надсилання відповіді;

Веб-сервер, в свою чергу, складається з серверного програмного забезпечення Dennou Bot. Програмне забезпечення веб-сервера використовується

для двох основних завдань: зв'язок з базою даних та виконання команд, які запитує користувач.

РОЗДІЛ 2. ПРОГРАМНА РЕАЛІЗАЦІЯ БОТА

2.1 Обгрунтування інструментарію розробки

Для розробки програми необхідно спершу визначити певні засоби та інструменти які використовуватимуться.

До таких можна віднести:

- Мова програмування;
- Середовище розробки;
- Використані інструменти розробки;
- Система контролю версій.

У цьому розділі буде проаналізовано можливості вибору таких засобів та наведено приклад розробки бота за їх допомогою.

API (Application Programming Interface) - це набір правил, протоколів та інструментів, які дозволяють програмам взаємодіяти між собою. API визначає, які функції та операції доступні для виклику, які дані можна передавати та які формати даних використовуються для обміну інформацією.

API може бути реалізовано в різних формах, але одним з найпоширеніших є веб-сервіси API. Веб-сервіси API дозволяють програмам взаємодіяти через мережу Інтернет. Це може включати використання протоколу HTTP для обміну запитами та відповідями. Наприклад, клієнтська програма може надсилати HTTP-запити на сервер API, що надає певні послуги або дані. Сервер оброблює ці запити, виконує необхідні операції та повертає відповіді з результатами клієнту.

API визначає також формати даних, які передаються між програмами. Часто використовуються формати, такі як JSON (JavaScript Object Notation) або XML (eXtensible Markup Language), щоб структурувати дані і забезпечити їх зрозумілість для обох сторін.

Публічні API надаються розробниками для використання сторонніми програмами. Це дозволяє розробникам створювати додатки, розширення та інтеграції, які використовують функціональність або дані, що надаються через API. Наприклад, соціальні мережі надають API, які дозволяють стороннім

розробникам отримувати доступ до профілів користувачів, публікувати повідомлення або отримувати повідомлення через ці платформи.

Приватні API можуть бути використані в межах однієї організації або проекту для забезпечення внутрішньої комунікації між компонентами програмного забезпечення. Це дозволяє різним частинам системи взаємодіяти між собою, обмінюватися даними та виконувати спільні операції.

API дозволяють розробникам створювати нові програми, використовуючи функціональність та дані, що надаються через API. Вони спрощують розробку програмного забезпечення, оскільки розробники можуть використовувати готові функції та компоненти, які надаються через API, замість того, щоб писати їх з нуля. Це дозволяє заощаджувати час і зусилля, а також забезпечує однорідність та стандартизацію взаємодії між програмами.

Узагальнюючи, API є важливим інструментом у світі програмування, який дозволяє програмам взаємодіяти між собою, обмінюватися даними та функціональністю. Вони розширюють можливості програмного забезпечення, дозволяючи створювати розширені та інтегровані програми та сервіси.

Telegram API (Application Programming Interface) - це набір інструментів та функцій, що надаються Telegram для розробників з метою створення програм, які взаємодіють з платформою Telegram.

Telegram API надає розробникам можливість здійснювати різноманітні дії, такі як надсилання повідомлень, управління групами та каналами, робота з контактами, отримання інформації про користувачів і багато іншого. Всі ці функції можуть бути використані для створення власних ботів або додатків, що інтегруються з Telegram.

Telegram API базується на протоколі MTProto (Telegram Messenger Protocol), який є шифрованим та забезпечує безпеку комунікації між клієнтами та серверами Telegram. API може бути використаний на різних платформах, включаючи веб, мобільні пристрої та настільні комп'ютери.

Для використання Telegram API розробник повинен отримати токен доступу (API token) від Telegram. Цей токен ідентифікує ваш додаток і дозволяє вам

взаємодіяти з Telegram API.

Основні можливості Telegram API включають:

- Надсилання повідомлень: ви можете відправляти текстові повідомлення, медіафайли (зображення, відео, аудіо), документи та інші типи повідомлень.
- Управління групами та каналами: ви можете створювати, редагувати та видаляти групи та канали, керувати їхніми учасниками, отримувати повідомлення з груп та каналів.
- Робота з контактами: ви можете отримувати інформацію про користувачів Telegram, включаючи їхній ідентифікатор, ім'я, фото та інші дані.
- Боти: ви можете створювати та налаштовувати ботів Telegram, які можуть автоматично відповідати на повідомлення, обробляти команди користувачів та виконувати різноманітні дії.
- Інтеграція зі сторонніми сервісами: ви можете використовувати Telegram API для інтеграції з іншими сервісами, такими як CRM-системи, платіжні шлюзи, сервіси нагадувань і т.д.

Telegram API дозволяє розробникам створювати різноманітні додатки та сервіси, які використовують можливості Telegram для комунікації, спілкування та автоматизації різних задач. Це робить його потужним інструментом для розробників, які хочуть інтегрувати Telegram у свої проекти.

BotFather - це найпростіший спосіб реєструвати, налаштовувати та керувати іншими Telegram-ботами. Простий у використанні і не вимагає спеціальних навичок, BotFather дозволяє реєструвати необмежену кількість нових ботів. Єдиною умовою для реєстрації нового бота є його унікальне ім'я користувача [15].

BotFather - ідеальне рішення для тих, хто не розуміється на програмуванні і не хоче довіряти створення та управління ботами стороннім людям або третім особам.

Відкрийте месенджер Telegram, увійдіть у свій обліковий запис або створіть новий.

Як створити нового бота в Telegram [16].

КРОК 1. Введіть у поле пошуку @BotFather і виберіть роботу.

В офіційному бота Telegram буде стояти синій підтверджуючий знак возле імені у вигляді галочки.

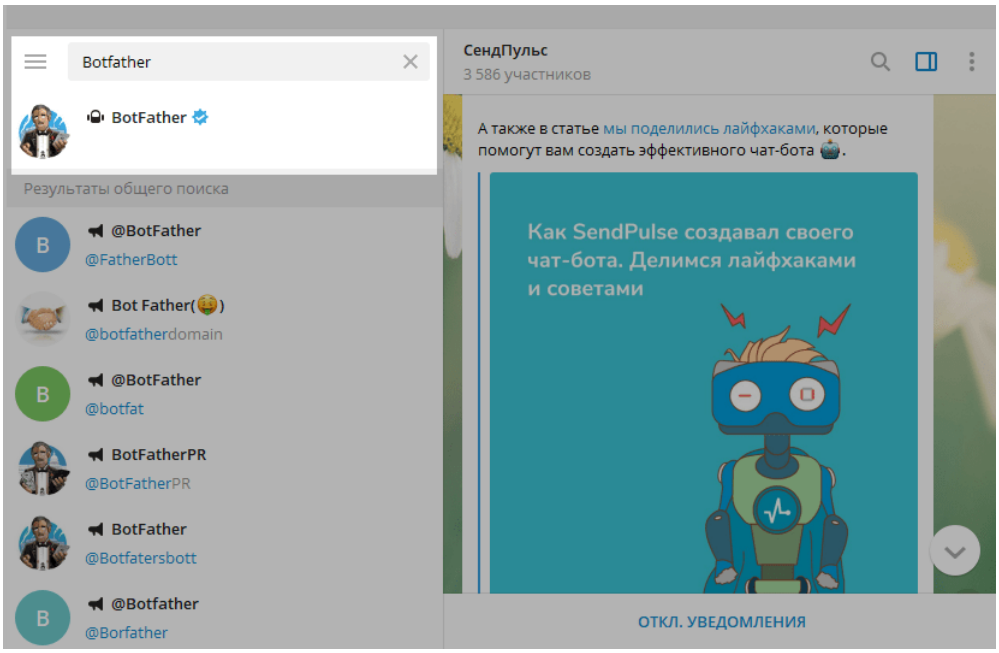


Рис. 2.1 - Поле пошуку @BotFather

КРОК 2. Натисніть «Запустити» для активації бота BotFather. У відповідь ви отримаєте список команд для керування ботами.

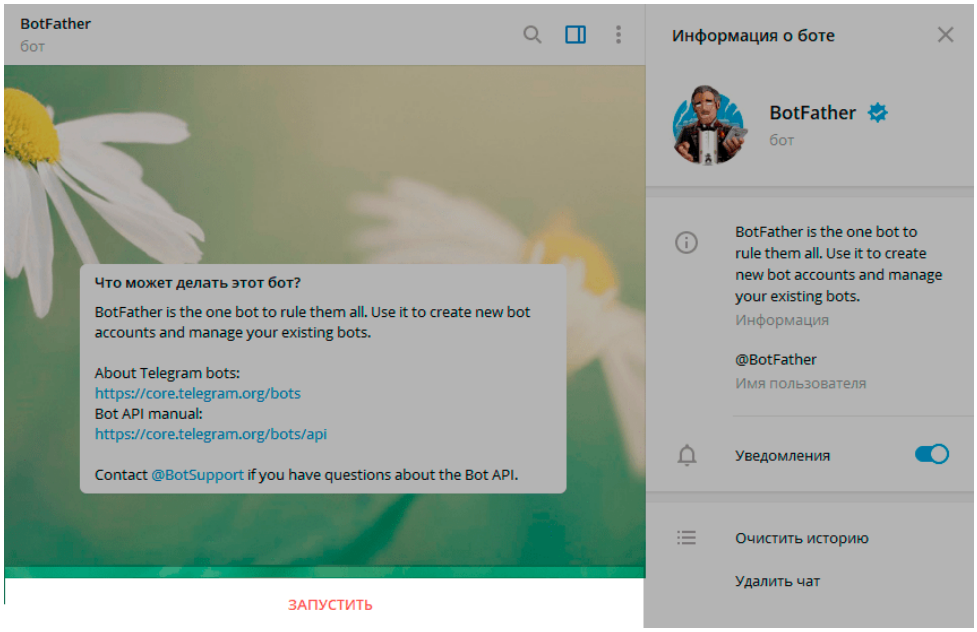


Рис. 2.2 – Запуск активації боту

КРОК 3. Виберіть або надрукуйте та виправте команду /newbot.

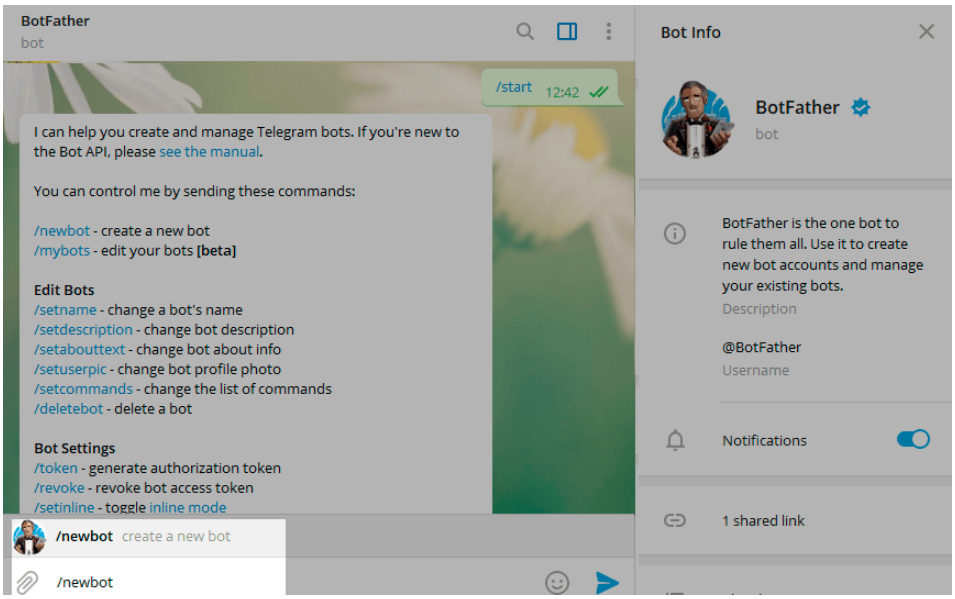


Рисунок 2.3 – Початок роботи із ботом BotFather та доступні команди

КРОК 4. Дайте боту ім'я - клієнти будуть пам'ятати це ім'я при спілкуванні з ботом. І нікнейм бота - за ним його можна буде знайти в Telegram. Нікнейм повинен бути унікальним, не збігатися з існуючим в базі даних і закінчуватися на слово "бот".

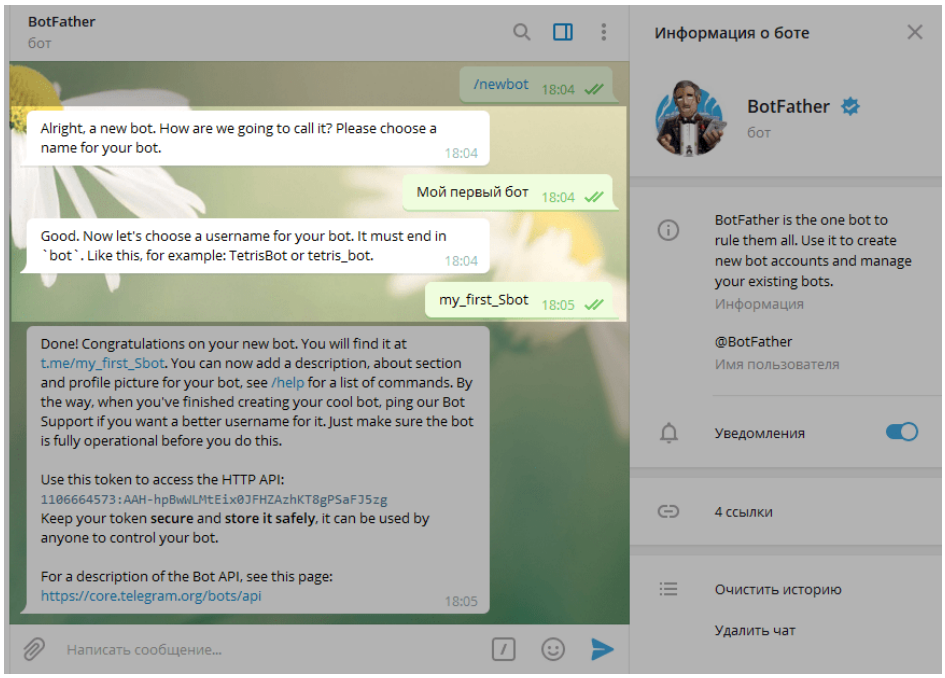


Рисунок 2.4 – Додавання імені боту

Після вибору відповідного імені бот буде створений. Ви отримаєте повідомлення з посиланням на бота t.me/<нікнейм_бота>, рекомендаціями щодо

налаштування аватарки, описом бота та переліком команд для його налаштування.

Для підключення бота до SendPulse вам знадобиться токен. Скопіюйте значення токена і перейдіть до останнього кроку.

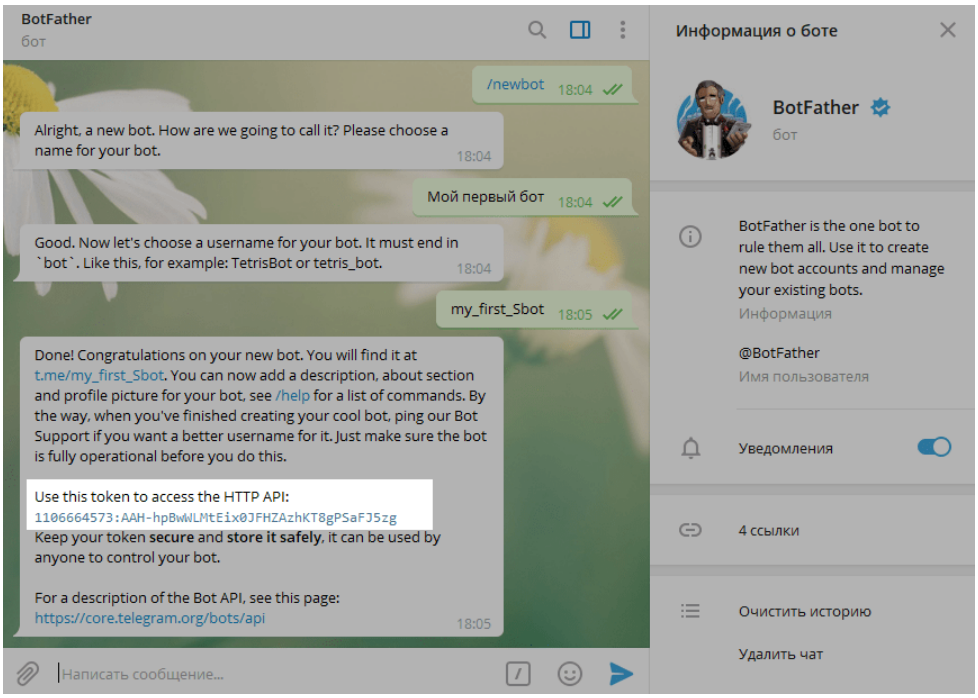


Рисунок 2.5 – Налаштування аватарки

Наприклад, якщо вам більше не подобається ім'я робота, ви можете змінити його за допомогою команди /setname.

Ви можете встановити аватарку за допомогою команди /setuserpic, змінити або додати короткий опис за допомогою команди /setdescription, змінити інформацію у профілі за допомогою команди /setabouttext або змінити ім'я за допомогою команди /setname. На цьому налаштування зовнішнього вигляду робота завершено, і ви можете користуватися його функціями. Щоб задати список команд, введіть /setcommands у вікні чату.

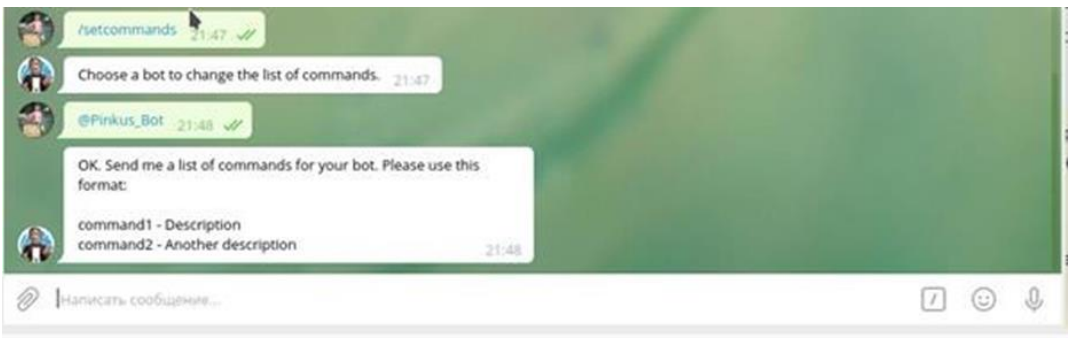


Рисунок 2.6 – Встановлення команд

Команди вводяться без скісної риски у форматі "команда1 - опис команди". Нижче наведено приклади основних команд та їх розшифрування:

/newbot - зареєструвати нового робота.

/mybots - редагування моїх [бета] ботів.

/setname - змінити ім'я бота;

/setdescription - змінити опис бота;

/setabouttext - змінити інформацію про бота; /setaboutbot - змінити інформацію про бота; /setaboutbot - змінити інформацію про бота

/setuserpic - змінити картинку профілю бота;

/setcommands - змінити список команд; /setcommands - змінити список команд; /setcommands - змінити список команд.

/deletebot - видалити бота. /setbot - налаштувати параметри бота.

/token - згенерувати токен аутентифікації; /token - згенерувати токен аутентифікації; /token - згенерувати токен аутентифікації;

/revoke - відкликати токен доступу бота;

/setinline - увімкнути вбудований-режим (ви можете звернутися до бота безпосередньо з будь-якого каналу, групи або чату, написавши ім'я бота в полі повідомлення) [17].

/setinlinegeo - вмикає запит на визначення місцезнаходження під час використання бота у вбудованому-режимі [17].

/setinlinefeedback - зміна налаштувань зворотного зв'язку (збір статистики про команди, що надсилаються найчастіше)[17].

/setjoingroups - визначає можливість додавання ваших роботів до груп;

/setprivacy - вмикає або вимикає режим конфіденційності в групах

Ігри.

/mygames - редагування власних ігор (бета-версія)[18];

/newgame - створити нову гру.

/listgames - отримати список ваших ігор;

/editgame - редагувати ваші ігри;

/deletegame - видалити існуючу гру.

Таким чином можна додати будь-які необхідні команди та завершити налаштування.

Мільйони людей обирають Telegram через його швидкість. Щоб залишатися конкурентоспроможними в цьому середовищі, боти теж мають бути швидкими. Щоб допомогти розробникам підтримувати ботів в актуальному стані, @BotFather надсилає сповіщення про статус, коли щось йде не так.

Telegram перевіряє кількість відповідей і коефіцієнт конверсії запитів/відповідей популярних ботів (300 запитів на хвилину: це значення може змінюватися в майбутньому). У разі незвично низьких показників ви отримаєте повідомлення від @BotFather.

Перш ніж обрати інструмент для розробки, варто ознайомитися з інтерфейсом прикладного програмування (API), який месенджер надає розробникам. Наразі існує два основних API-інструменти, за допомогою яких можна використовувати сервіси Telegram.

- Telegram Bot API та Telegram API. Перший використовується для розробки чат-ботів, а другий дозволяє створювати повністю кастомізовані клієнти Telegram. Розробники також мають доступ до бібліотеки з відкритим вихідним кодом TDLib (Telegram Database Library), за допомогою якої можна створювати версії месенджера з власними опціями (наприклад, Telegram X, побудований на TDLib та ін.) Оскільки Telegram Bot API є надбудовою над Telegram API, Bot API можна використовувати без будь-яких знань про те, як працює протокол MTProto.

Використовуйте проміжний сервер з інтерфейсом HTTPS, який шифрує трафік і взаємодіє з Telegram API. API бота дозволяє легко створювати додатки, які виконують код на локальному сервері за допомогою інтерфейсу Telegram. Користувачі можуть взаємодіяти з ботом, надсилаючи йому повідомлення, команди та вбудовані запити.

Принцип роботи будь-якого бота полягає в постійному надсиланні запитів на сервер і регулярному отриманні оновлень. Отримувати їх можна двома способами. По-перше, можна використовувати веб-хук, якщо сервер виконує зворотний дзвінок на задану URL-адресу. По-друге, ви можете просто "закинути"

запит у Telegram і періодично отримувати відповідь[18].

Зверніть увагу, що ви можете отримувати сповіщення про нові повідомлення лише один раз, наприклад, на роботі. Тому, якщо дані вашого чату дуже важливі, ви повинні самостійно зберігати список чатів та історію старих повідомлень. Якщо ви випадково зітрете/втратите цю інформацію, її неможливо буде відновити.

На відміну від Bot API, який дозволяє отримувати оновлення лише один раз, Telegram API дозволяє обійти це обмеження, якщо ви використовуєте кілька клієнтів. В такому випадку робот буде отримувати всі оновлення на кожному запущеному клієнті. Крім того, Bot API не може надсилати сповіщення всім користувачам одночасно[18].

2.2 Вибір засобів розробки чат-боту та бібліотек

Для свого проєкту я обрав мову програмування Python.

Простота вивчення та використання Python є однією з його ключових переваг. Ось кілька аспектів, які роблять Python легким для новачків та досвідчених розробників:

Зрозумілий синтаксис: Синтаксис Python дуже зрозумілий та логічний. Код на Python дуже близький до англійської мови, що робить його більш зрозумілим для розробників. Відсутність скобок для визначення блоків коду та використання відступів для відокремлення блоків робить код на Python дуже читабельним. Приклад коду на Python, що виводить привітання (рис. 2.7).

```
name = "John"
print("Hello, " + name + "!")
```

Рис. 2.7 – Приклад коду та синтаксису

Мало написаного коду: Python відомий своєю здатністю вирішувати завдання за допомогою меншої кількості коду порівняно з іншими мовами програмування. Це полегшує розуміння коду та зменшує його обсяг, що в свою чергу сприяє більшій продуктивності розробників. Приклад коду на Python, що

обчислює суму перших 10 натуральних чисел (рис. 2.8).

```
total = sum(range(1, 11))
print(total)
```

Рис. 2.8 – Приклад коду який займає мало місця

Багато вбудованих функцій та бібліотек: Python має велику кількість вбудованих функцій та бібліотек, які полегшують виконання різноманітних завдань. Наприклад, вбудована функція len() дозволяє отримати довжину списку або рядка, а модуль math надає функціональність для математичних операцій. Використання цих вбудованих функцій та бібліотек дозволяє розробникам швидко вирішувати завдання без необхідності писати власний код з нуля.

Багата екосистема: Python має велику кількість сторонніх бібліотек та фреймворків, які розширюють функціональність мови і спрощують розробку різних типів проектів. Наприклад, для веб-розробки можна використовувати фреймворки, такі як Django або Flask, а для аналізу даних - бібліотеки, такі як NumPy або Pandas. Для розробки телеграм ботів можна використовувати популярні фреймворки, такі як python-telegram-bot або telebot, які надають зручні інтерфейси для взаємодії з Telegram API.. Ця багата екосистема дозволяє розробникам швидко реалізовувати складні функції та розв'язувати різноманітні задачі.

Документація та спільнота: Python має високоякісну документацію, яка детально описує синтаксис, функціонал. У разі виникнення питань або потреби в допомозі, у розробників Python є доступ до широкої спільноти, яка завжди готова допомогти. На форумах, стековерфлоу, чат-каналах та інших ресурсах ви зможете знайти відповіді на свої питання, приклади коду та поради від інших розробників. Це дозволяє ефективно вирішувати проблеми та швидко рухатись у розробці проєкту.

Широке застосування: Python застосовується в різних галузях, таких як веб-розробка, наукові обчислення, машинне навчання, автоматизація, інтернет

вещей та багато інших. Його простота та багатий інструментарій дозволяють розробникам використовувати Python для різних типів проектів і задач.

Після вибору мови потрібно обрати місце в якому буде розроблятися телеграм-бот. Переглянувши декілька варіантів я зупинився на Visual Studio.

Visual Studio - це інтегроване середовище розробки (IDE) від компанії Microsoft, яке надає потужні інструменти для розробки програмного забезпечення. Це середовище підтримує різні мови програмування, включаючи Python. Ось кілька причин, чому варто використовувати Visual Studio для розробки на мові Python:

Повноцінне інтегроване середовище розробки: Visual Studio надає повноцінне середовище розробки з широким спектром функцій. Воно включає редактор коду з підсвічуванням синтаксису, автодоповненням, перевіркою помилок та відлагодженням. За допомогою Visual Studio можна легко створювати, змінювати та налагоджувати Python-програми.

Підтримка популярних фреймворків: Visual Studio має підтримку популярних фреймворків Python, таких як Django, Flask, PyTorch, TensorFlow та інші. Воно надає зручні інструменти для створення та розробки проєктів на основі цих фреймворків.

Багатофункціональний відлагоджувач: Visual Studio має потужний відлагоджувач, який дозволяє розробникам ефективно відлагоджувати свій Python-код. Воно підтримує точкову зупинку, відстеження змінних, виклик функцій та багато інших функцій, які спрощують процес відлагодження та полегшують виявлення помилок.

Широкий вибір розширень: Visual Studio має велику кількість розширень, які розширюють його функціональність для розробки на мові Python. Наприклад, існують розширення для підтримки спеціальних функцій Python, таких як аналіз даних, машинне навчання, веб-розробка тощо. Це дозволяє розробникам налаштовувати Visual Studio під свої потреби та отримувати додаткові можливості для розробки на Python.

Інтеграція з іншими інструментами: Visual Studio має здатність

інтегруватися з іншими популярними інструментами для розробки, такими як Git для контролю версій, Docker для контейнеризації, Azure для хмарного розгортання та багато інших. Це спрощує роботу з цими інструментами та полегшує розробку і впровадження проєктів на Python.

В цілому, Visual Studio є потужним інструментом для розробки на мові Python. Вона надає зручне та повноцінне середовище розробки, підтримку популярних фреймворків, ефективний відлагоджувач та широкі можливості розширення. Це робить Visual Studio привабливим вибором для розробників Python, незалежно від їхнього рівня досвіду.

Тепер варто обрати бібліотеки для створення чат-бота.

Бібліотека telebot. Одна з найпопулярніших бібліотек для розробки телеграм-ботів з використанням мови програмування Python. Вона надає зручний інтерфейс для взаємодії з API Telegram і дозволяє розробникам легко створювати та керувати функціональністю своїх ботів.

Ось кілька ключових особливостей та можливостей бібліотеки telebot:

Простота використання. Бібліотека telebot пропонує зрозумілий та логічний інтерфейс, що дозволяє швидко розпочати роботу з розробкою телеграм ботів. Вона має чітку документацію та приклади коду, що полегшує розуміння та використання різних функцій бібліотеки.

Обробка повідомлень. Бібліотека telebot дозволяє отримувати та обробляти повідомлення від користувачів Telegram. Ви можете створювати обробники повідомлень, що реагують на текстові повідомлення, команди, клавіатуру та інші події. Це дозволяє створювати інтерактивних телеграм ботів з різними функціями.

Відправка повідомлень. За допомогою бібліотеки telebot ви можете відправляти повідомлення, зображення, аудіо, відео, документи та інші медіафайли користувачам Telegram. Вона надає зручні методи для відправки повідомлень з включенням розмітки, клавіатур та інших елементів.

Робота з клавіатурами. Бібліотека telebot підтримує створення і використання клавіатур для взаємодії з користувачами. Ви можете створювати кнопки, розміщувати їх у вигляді рядків та стовпців, налаштовувати обробку

натискань кнопок. Це дозволяє створювати інтерактивні меню та спрощує навігацію користувачів в телеграм боті.

Асинхронна підтримка. Бібліотека telebot підтримує асинхронну роботу з використанням асинхронних фреймворків, таких як asyncio. Це дозволяє створювати ефективні та швидкодіючі телеграм боти з можливістю обробки багатьох запитів одночасно.

Розширені можливості. Бібліотека telebot підтримує багато інших функцій та можливостей, таких як робота з клавішами відповіді, пересилання повідомлень, редагування повідомлень, робота з чатами та групами, зчитування даних користувачів та багато іншого. Це дозволяє створювати потужні телеграм боти з різноманітними функціями.

Бібліотека telebot є потужним інструментом для розробки телеграм ботів на мові програмування Python. Вона пропонує зручний та простий інтерфейс, що дозволяє швидко розгортати ботів та створювати різноманітні функції для взаємодії з користувачами Telegram.

Для розуміння розглянемо встановлення бібліотеки та базові команди для налаштування телеграм-бота.

Встановлення. Для встановлення бібліотеки telebot, можна використовувати менеджер пакетів pip. Просто виконайте команду *pip install pytelegrambotapi* у терміналі або командному рядку.

Ініціалізація бота. Після встановлення бібліотеки telebot, необхідно створити об'єкт бота із використанням токену Telegram. Токен був створений раніше через бота BotFather у Telegram. Приклад ініціалізації (рис. 2.9).

```
import telebot

bot = telebot.TeleBot("YOUR_TELEGRAM_TOKEN")
```

Рис. 2.9 – Ініціалізація токену

Запуск бота. Після написання всіх необхідних обробників і функціоналу, потрібно запустити бота за допомогою команди bot.polling().

Бібліотека SQLite3. Для того, щоб зберігати наші дані з розкладом занять та іншим, була обранабаза даних SQLite3.

SQLite3 – це компактна інтегрована реляційна база даних. Вихідний код бібліотеки було передано на громадське надбання. Це чисто реляційна база даних. Типи даних, що підтримуються:

- NULL: значення NULL.
- INTEGER: ціле число зі знаком, що зберігається в 1, 2, 3, 4, 6 або 8 байтів.
- REAL: число з плаваючою комою, що зберігається в 8-байтовому формат IEEE.
- TEXT: текстовий рядок, закодований UTF-8, UTF-16BE або UTF-16LE.
- BLOB: тип даних, що зберігаються в тій самій формі, в якій вони були отримані.

Переваги:

- Файлова: вся база даних зберігається в одному файлі, що полегшує переміщення.
- Стандартизована: SQLite використовує SQL.
- Також чудово підходить для розробки та тестування: на етапі розробки більшість людей потребують масштабованого рішення. Багатофункціональний SQLite забезпечує більш ніж достатню функціональність і може бути настільки простим, як робота з одним файлом і пов'язаними з ним бібліотеками.

Недоліки:

- Відсутність управління користувачами: просунуті бази даних мають можливість для користувачів керувати 33 зв'язками між таблицями відповідно до їхніх дозволів, але SQLite не має такої функціональності.
- Неможливість додаткової конфігурації: SQLite не може бути більш продуктивним. У цьому розділі ми ознайомилися з базами даних, необхідними для чат-ботів, і пояснили їхні концепції, переваги та недоліки.

2.3 Розробка чат-бота

Перед початком розробки самого бота необхідно створити базу даних. Вона буде складатися з таблиці rozklad (рис. 2.10).

Та з наступних колонок:

- *group_name* – назва групи (TEXT);
- *day* – день тижня (TEXT);
- *sub* – предмет (TEXT);
- *starting* – початок заняття (TEXT);
- *duraction* – тривалість заняття (INTEGER);
- *teacher* – викладач (TEXT).

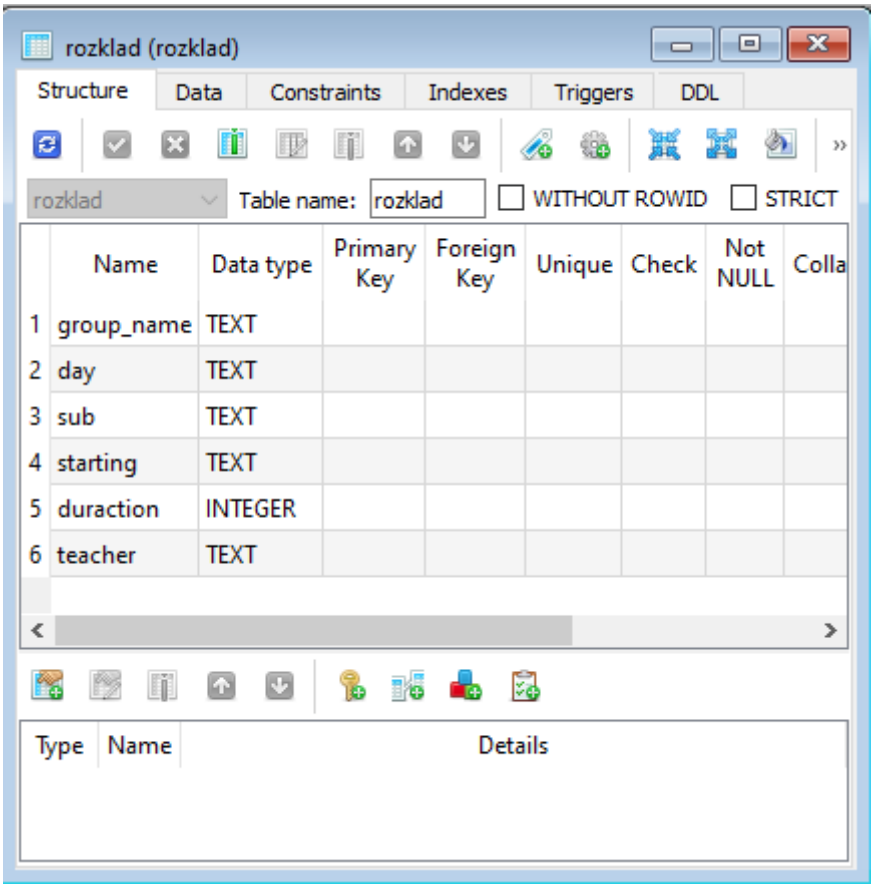


Рис. 2.10 – Структура база даних

База даних наповнюється даними, заповнюється кожна колонка відповідними даними.

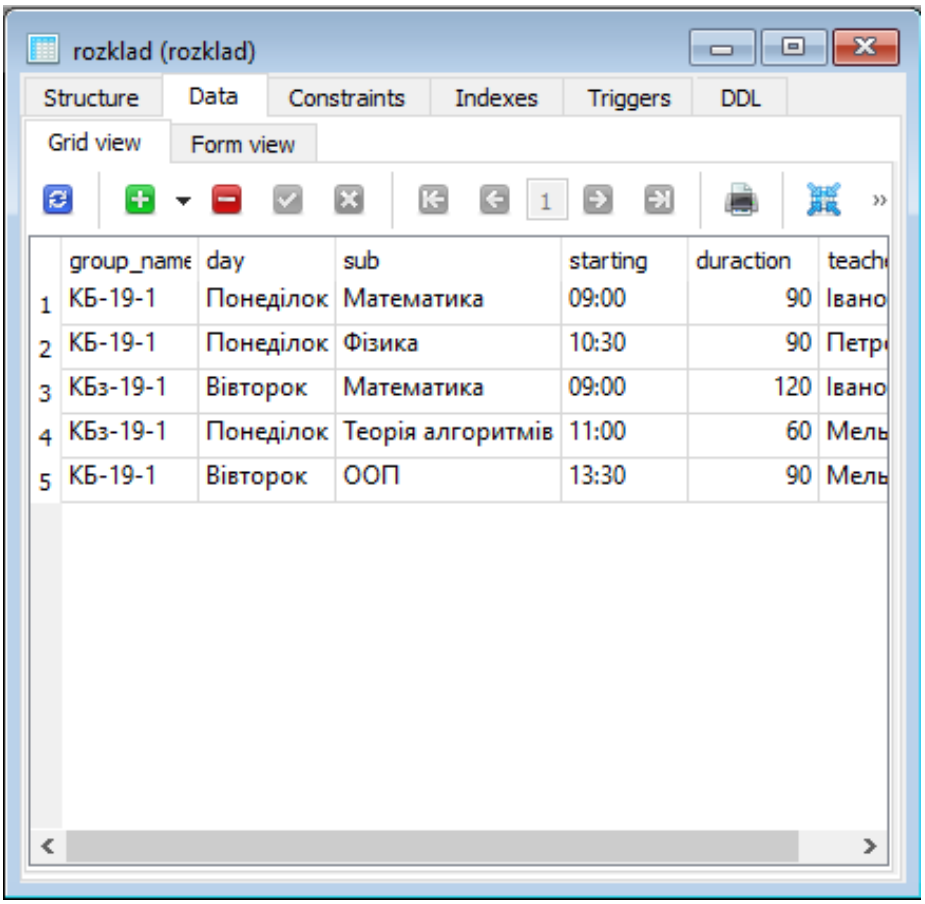


Рис. 2.11 – Наповнена база даних

Ця база даних буде використовуватись в самому боті у якості демонстрації. При можливості підключення бази даних розкладу від університету бот зможе виводити актуальну інформацію.

Розглянемо код створеного чат-бота. Для початку необхідно підключити бібліотеки telebot, config, sqlite3 (лістинг 2.1).

Лістинг 2.1 – Імпортування бібліотек

```
import telebot
from telebot import types
import config
import sqlite3
```

Далі необхідно підключити токен який був створений в BotFather, та підключити створену базу даних (лістинг 2.2).

Лістинг 2.2 – Підключення токену та бази даних

```
bot = telebot.TeleBot(config.Token)
conn = sqlite3.connect('rozklad.db', check_same_thread=False)
cursor = conn.cursor()
```

Далі створюється обробник команди /start. Він виводить повідомлення «Привіт, для початку роботи обери курс» та створює спеціальну клавіатуру, яка має дві кнопки з вибором групи (лістинг 2.3).

Лістинг 2.3 – Обробник команди /start

```
@bot.message_handler(commands=['start'])
def start(message):
    markup = types.ReplyKeyboardMarkup(resize_keyboard = True)
    item1 = types.KeyboardButton('КБ-19-1')
    item2 = types.KeyboardButton('КБз-19-1')
    markup.add(item1,item2)
    bot.send_message(message.chat.id, "Привіт, для початку роботи обери курс",
reply_markup = markup)
```

Наступним кроком бот перевіряє, яку кнопку було натиснуто та робить вибірку по обраній групі. У випадку невідомої команди, бот виводить відповідне повідомлення. У результаті успішної команди виводить розклад користувачу (Лістинг 2.4).

Лістинг 2.4 – Обробник вибору групи

```
@bot.message_handler(content_types=['text'])
def menu(message):
    if message.chat.type == 'private':
        if message.text == 'КБ-19-1':
            msg_course = message.text
            cursor.execute(f"SELECT * FROM rozklad WHERE group_name =
'{msg_course}'")
            rozklad = cursor.fetchall()
```

```
        result = "Розклад:\n"
        for row in rozklad:
            result += f"День: {row[1]}\n"
            result += f"Предмет: {row[2]}\n"
            result += f"Початок: {row[3]}\n"
            result += f"Тривалість: {row[4]} хвилин\n"
            result += f"Викладач: {row[5]}\n\n"
        bot.send_message(message.chat.id, result)
    elif message.text == 'КБз-19-1':
        msg_course = message.text
        cursor.execute(f"SELECT * FROM rozklad WHERE group_name =
'{msg_course}'")
        rozklad = cursor.fetchall()
        result = "Розклад:\n"
        for row in rozklad:
            result += f"День: {row[1]}\n"
            result += f"Предмет: {row[2]}\n"
            result += f"Початок: {row[3]}\n"
            result += f"Тривалість: {row[4]} хвилин\n"
            result += f"Викладач: {row[5]}\n\n"
        bot.send_message(message.chat.id, result)
    else:
        bot.send_message(message.chat.id, "Невідома команда")
```

Остання команда запускає бота (лістинг 2.5).

Лістинг 2.5 – Запуск бота

```
bot.polling()
```

Отже, бот був розроблений. Він може виводити розклад в залежності від групи. Крім того, бот має зручну клавіатуру, яка полегшує

взаємодію з користувачами. Замість того, щоб вводити команди або параметри вручну, користувачі можуть просто натиснути на кнопки на клавіатурі, що розташована в інтерфейсі бота. Це робить взаємодію з ботом більш інтуїтивно зрозумілою та легкою, зменшуючи ризик помилок та полегшуючи використання.

РОЗДІЛ 3. ПРАКТИЧНЕ ЗАСТОСУВАННЯ ЧАТ-БОТУ

3.1. Тестування чат-бота

Тестування телеграм-бота - це процес перевірки його функціональності, надійності та якості перед випуском.

Функціональне тестування - перевірка, чи працюють всі функції бота правильно і відповідають очікуванням. Наприклад, перевірка, чи відповідає бот на правильні команди, виконує запити користувачів або надсилає повідомлення.

Тестування взаємодії - перевірка взаємодії бота з користувачами. Наприклад, перевірка, чи відповідає бот на запити, які містять текст, зображення, аудіо або відео.

Тестування на різних платформах - перевірка, чи працює бот на різних платформах, таких як мобільні пристрої, комп'ютери або планшети.

Щоб ближче познайомитися з результатами практичної частини, розглянемо кілька сценаріїв взаємодії чат-бота з користувачем.

В описі чат-бота коротко описано його функціонал. Образ бота базується на дизайні, розробленому для конкретного університету (чи факультету). Бота можна знайти, набравши його ім'я в пошуку Telegram, ось так виглядає бот коли його вперше відкрити (*рис. 3.1*).

Для початку роботи з ботом користувач має натиснути на кнопку "Розпочати".



Рис. 3.1 – Початок роботи з ботом

Після натиску на кнопку «Розпочати» або при введені команди /start. Можна побачити повідомлення від бота, та поява зручної клавіатури з назвами груп (*рис. 3.2*). Можливість отримувати зворотній зв'язок від людей та ботів є дуже важливою функцією. Не всі питання можна легко вирішити шляхом пошуку та вивчення.



Рис. 3.2 – Команда /start

При натиску на кнопку «КБ-19-1» бот запам’ятовує обрану групу та готується вивести потрібну інформацію (рис. 3.3).

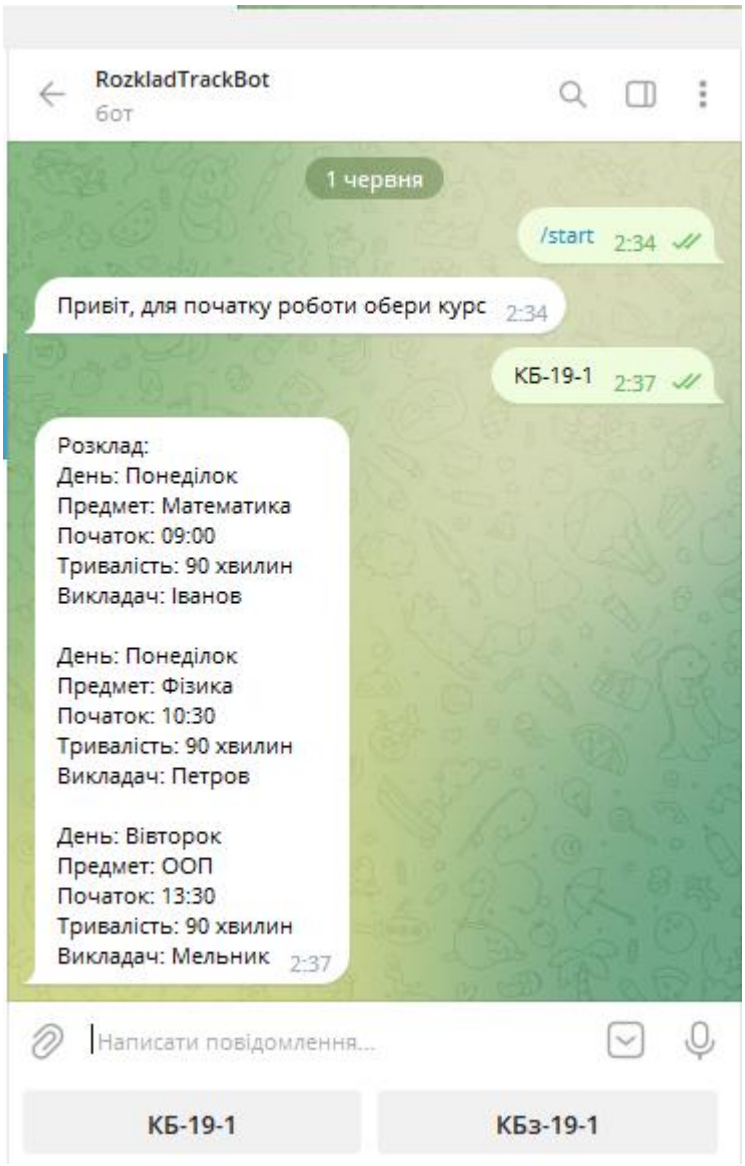


Рис. 3.3 – Розклад для групи КБ-19-1

При натиску на кнопку «КБз-19-1» бот запам’ятовує обрану групу та готується вивести потрібну інформацію (рис. 3.4).

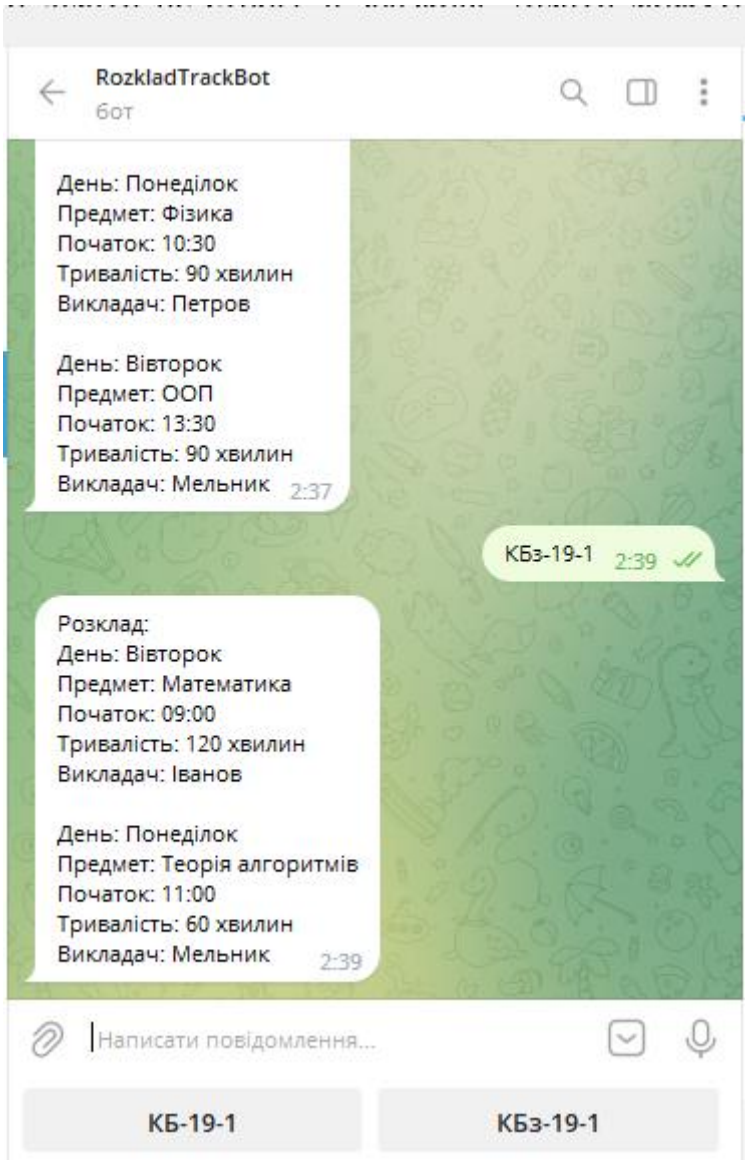


Рис. 3.3 – Розклад для групи КБ-19-1

Отже, в результаті тестування бот успішно виконує всі свої основні функції без помилок і збоїв. Він правильно розпізнає команди користувачів і надає відповіді відповідно до своїх налаштувань. Бот повністю задовольняє встановлені вимоги, що стосуються його функцій, реакцій, інтерфейсу та іншого функціоналу. Він відповідає всім вимогам, визначеним у специфікації проекту. Бот працює стабільно та надійно протягом тривалого часу без збоїв або критичних помилок. Бот відповідає на запити користувачів швидко та точно, надаючи зрозумілі та корисні відповіді. Якщо бот має графічний інтерфейс, то він має привабливий та зрозумілий дизайн, зручний для навігації та використання.

3.2. Приклади застосування та операційний аналіз

Чат-бот не може отримати старі повідомлення чату. Бот не може отримати список всіх користувачів. Бот може отримувати лише інформацію про оновлення. Це основна складність у розробці бота.

Він може отримувати інформацію про нові повідомлення з роботи, але тільки один раз. Потрібно зберігати список чатів, старих повідомлень (якщо це потрібно з якихось причин) тощо. Якщо ця інформація буде випадково стерта/загублена, ви ніколи не отримаєте її знову.

За допомогою Telegram API бот може робити трохи більше: отримувати повідомлення за ID, отримувати список учасників групи тощо. Оцінювати правильність відповідей.

Звичайно, існують обмеження на запити до сервера: про це йдеться в FAQ по ботам на сайті Telegram:

- Не більше одного повідомлення на секунду в один чат,
- Не більше 30 повідомлень за секунду взагалі,
- Не більше 20 повідомлень за хвилину в одну групу.

Ці ліміти не строгі, а зразкові. Ліміти можуть бути збільшені для великих роботів через підтримку.

3.3. Розміщення чат-бота на хмарному сервері Heroku

Щоб забезпечити безперебійний доступ до чат-бота після завершення розробки, його потрібно розмістити на віддаленому сервері. З цієї причини хмарною платформою було обрано Heroku [19].

Хостинг на хмарній PaaS-платформі Heroku схожий на роботу з розподіленою системою контролю версій, відомою як Git. Існує три способи її розгортання:

– За допомогою Heroku Git та реєстру контейнерів за допомогою Heroku CLI;

– Підключення облікового запису GitHub з автоматичним розгортанням;

– Підключення облікового запису Dropbox з автоматичним розгортанням.

Для розгортання чат-бота були виконані наступні кроки:

– Зареєструватися на хмарній SaaP-платформі Heroku Cloud;

– Завантажте та встановіть – Heroku CLI;

– Увійдіть до свого облікового запису Heroku за допомогою команди 'heroku login' в терміналі ОС або вбудованому IDE;

– Клонований репозиторій з віддаленого Git-сервера Heroku на локальну машину за допомогою команди heroku git:clone -a <APP_NAME>;

– Зафіксовані зміни в коді за допомогою команд git add . та «git commit -am <COMMIT_NAME>»;

Всі зафіксовані зміни були відправлені на віддалений сервер хмарних обчислень Heroku командою «git push heroku master».

Після того, як всі вищезазначені команди будуть виконані у відповідній послідовності, автоматично розпочнеться передача даних на віддалений сервер Heroku. Якщо розгортання пройшло успішно, на терміналі відобразиться довідкова інформація про стан програми та режим доступу (*Рис. 3.4*).



Рис. 3.4 – Результат розгортання чат-бота

ВИСНОВКИ

У даній дипломній роботі було проведено дослідження та розробку телеграм бота для створення розкладу занять в університеті. Основною метою роботи було спростити процес отримання розкладу для студентів та викладачів, шляхом автоматизації цього процесу за допомогою телеграм бота.

Під час роботи було проведено аналіз поточних систем розкладу занять університету, виявлено проблеми та недоліки, пов'язані з відсутністю зручностей в отриманні розкладу та його актуальності. На основі цього були визначені вимоги до телеграм бота, який мав би вирішити ці проблеми та покращити процес отримання розкладу.

Пройдемось по всім розділам дипломної роботи:

Перший розділ цієї дипломної роботи присвячений проектній проблемі, яку вирішує розробка чат-ботів. Детально описано існуючі типи ботів, їх розповсюдження та використання. Розглянуто приклади успішних рішень та способи їх реалізації. Перша глава також присвячена алгоритмам, що лежать в основі чат-ботів. Пояснюються принципи, що лежать в основі створення чат-ботів. Проведено аналіз для вибору алгоритму, який буде використаний для створення. За допомогою обраних принципів обговорюється, яких результатів можна досягти.

У другому розділі розглянуто поняття чат-бота, який буде використовуватися для розробки, представлено його функції та наведено класифікацію всіх чат-ботів, що існують на сьогоднішній день. Далі було обґрунтовано вибір програмного забезпечення для реалізації чат-бота, в результаті чого було обрано наступні інструменти: мови програмування Python, текстовий редактор Webstorm, месенджер Telegram та СУБД SQLite. А також були обрані наступні бібліотеки: telebot та sqlite3. У розділі описано, як було обрано середовище розробки та мову програмування. Розглянуто та описано процес розробки за допомогою коду.

У третьому розділі описано практичну частину по запуску та організації роботи чат – боту. Проведено тестування робото спроможності чат-бота Наведено скріншоти інтерфейсу та пояснення, як користуватися чат-ботом.

Результати експерименту показали, що розроблений телеграм-бот є ефективним інструментом для отримання розкладу занять університету. Він дозволяє студентам та викладачам швидко та зручно отримувати актуальну інформацію про розклад, уникати зайвих труднощів та затрат часу.

Отже, на основі проведених досліджень та розробки можна зробити висновок, що створення телеграм-бота для розкладу занять університету є важливим кроком у поліпшенні процесу отримання розкладу для студентів та викладачів. Цей бот може бути використаний як основний інструмент для отримання інформації про розклад та сприяти покращенню організації навчального процесу в університеті.

ВИКОРИСТАНА ЛІТЕРАТУРА

1. ТКАЧУК, В. О.; ЧАЛАХЕШАШВІЛІ, Д. Г. ОПТИМІЗАЦІЯ БІЗНЕС-ПРОЦЕСІВ ІЗ ЗАСТОСУВАННЯМ ЧАТ-БОТІВ. *Рекомендовано до друку науково-технічною радою Луцького національного технічного університету, протокол № 5 від 15.12. 2021 р.*, 2021, 526.

2. МЄЛЄНТЬЄВА, М. В.; ЛЕБЕДЄВА, О. Ю. Розробка чат-бота розкладу занять для месенджера TELEGRAM. 2018.

3. Kemp, Simon. "Digital in 2018: World's internet users pass the 4 billion mark." *We are social* 30 (2018).

4. Priadko, A. O., Osadcha, K. P., Kruhlyk, V. S., & Rakovych, V. A. "Development of a chatbot for informing students of the schedule" (2020).

5. Arun, K., A. Sri Nagesh, and P. Ganga. "A multi-model and AI-based collegebot management system (AICMS) for professional engineering colleges." *International Journal of Innovative Technology and Exploring Engineering* 8.9 (2019): 2910-2914.

6. Lee, Dongkeon, Kyo-Joong Oh, and Ho-Jin Choi. "The chatbot feels you-a counseling service using emotional response generation." *2017 IEEE international conference on big data and smart computing (BigComp)*. IEEE, 2017. DOI: 10.1109/BIGCOMP.2017.7881752

7. Rosruen, Nudtaporn, and Taweesak Samanchuen. "Chatbot utilization for medical consultant system." *2018 3rd technology innovation management and engineering science international conference (TIMES-iCON)*. IEEE, 2018. DOI: 10.1109/TIMES-iCON.2018.8621678

8. Tsai, Yin-Te and Wei-An Lin. "Design of an Intelligent Cognition Assistant for People with Cognitive Impairment." 2018 *IEEE 20th International Conference on High-Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/Smart City/DSS)* (2018): 1207-1212. DOI: 10.1109/HPCC/SmartCity/DSS.2018.00203

9. Pratt, E. "Artificial Intelligence and Chatbots in Technical Communication–A

Primer." *Intelligent Information Journal* (2017).

10. Kozlov, A. A., and A. V. Batishchev. "Telegram-bot kak prostoj i udobnyj sposob polucheniya informacii [Telegram bot as a simple and convenient way to get information]. Territory of Science. No. 5." *Territory of Science* 5 (2017).

11. Social networks, most popular in Ukraine and countries of the world 2020 *[Електронний ресурс] //Режим доступу: https://marketer.ua/ua/the-most-popular-social-networksin-the-world/*

12. Чатбот для пошуку та моніторингу залізничних квитків *[Електронний ресурс] //Режим доступу: https://railwaybot.com/uk/*

13. Чат-боти: від розваг до бізнес-рішень *[Електронний ресурс] //Режим доступу: https://privatbank.ua/udalenniy-banking/chat-boty*

14. Bykov, V. Yu. "Vidkryta osvita v Yedynomu informatsiinomu osvitnomu prostori [Open education in the Unified Information Educational Space]." *Pedagogical discourse* 7 (2010): 30-35.

15. Інструкція по роботі з BotFather ботом *[Електронний ресурс] //Режим доступу: https://botcreators.ru/blog/botfather-instrukciya/*

16. Як створити та підключити Telegram чат-бота *[Електронний ресурс] //Режим доступу: https://sendpulse.ua/ru/knowledge-base/chatbot/telegram/create-telegram-chatbot*

17. Inline Bots *[Електронний ресурс] //Режим доступу: https://core.telegram.org/bots/inline*

18. Gaming Platform *[Електронний ресурс] //Режим доступу: https://core.telegram.org/bots/games*

19. Heroku CLI шпаргалка *[Електронний ресурс] //Режим доступу: http://wp-blog.pp.ua/2018/06/09/heroku-cli-%D1%88%D0%BF%D0%B0%D1%80%D0%B3%D0%B0%D0%BB%D0%BA%D0%B0/*