

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: Аналіз та вдосконалення інформаційного чат-боту ТНТУ засобами Python та Aiogram

Виконав: студент VI курсу, групи СНМ-61
спеціальності 122 Комп'ютерні науки
(шифр і назва спеціальності)

(підпис)

Шишкіна В.О.

(прізвище та ініціали)

Керівник

(підпис)

Козбур Г.В.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Никитюк В.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

Яворська Є.Б.

(прізвище та ініціали)

Тернопіль
2023

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Боднарчук І.О.
(підпис) (прізвище та ініціали)

«__» _____ 2023 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Магістр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студентці Шишкіній Вікторії Олександрівні
(прізвище, ім'я, по батькові)

1. Тема роботи Аналіз та вдосконалення інформаційного чат-боту ТНТУ засобами Python та Aiogram.

Керівник роботи Козбур Галина Володимирівна, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «24» листопада 2023 року № 4/7-1099

2. Термін подання студентом завершеної роботи _____ грудня 2023р.

3. Вихідні дані до роботи Наукові публікації

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Аналітичний огляд сучасного стану використання чат-ботів. 1.1. Актуальність використання чат-ботів. 1.2. Сфери використання чат-ботів. 1.3. Чат-боти зі штучним інтелектом. 1.4. Приклади найпопулярніших чат-ботів у різних соціальних мережах.

1.5. Висновок до першого розділу. 2. Підходи до розробки чат-ботів та аналіз активності користувачів. 2.1. Сценарій розробки чат-бота. 2.2. Прикладне використання бібліотеки pyTelegramBotAPI. 2.3. Порівняльний аналіз бібліотек для розробки чат-бота у месенджері. Telegram. 2.4. Дослідження активності користувачів Інформаційного чат-бота ТНТУ.

2.5. Висновок до другого розділу. 3. Реалізація оновленої версії інформаційного чат-бота ТНТУ. 3.1. Вдосконалення інформаційного чат-бота ТНТУ із застосуванням бібліотеки Aiogram. 3.2. Реалізація запису аналітичних даних активності користувачів. 3.3. Доповнення функціоналу Інформаційного чат-бота ТНТУ. 3.4. Висновок до третього розділу 4. Охорона праці та безпека в надзвичайних ситуаціях. 4.1. Охорона праці. 4.2. Безпека в надзвичайних ситуаціях. 4.3. Висновок до четвертого розділу. Висновки. Перелік джерел. Додатки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Сенчишин В.С., доцент		
Безпека в надзвичайних ситуаціях	Клепчик В.М., ст. викладач, проректор		

7. Дата видачі завдання 24 листопада 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	25.11.2023	Виконано
2.	Підбір наукових джерел про аналіз та вдосконалення інформаційного чат-боту ТНТУ засобами Python та Aiogram	26.11.2023-28.11.2023	Виконано
3.	Опрацювання наукових публікацій та збір даних по аналізу та вдосконаленню інформаційного чат-боту ТНТУ засобами Python та Aiogram	29.11.2023-01.12.2023	Виконано
4.	Виконання дослідження згідно мети кваліфікаційної роботи	02.12.2023-04.12.2023	Виконано
5.	Оформлення розділу «Аналітичний огляд сучасного стану використання чат-ботів»	05.12.2023-07.12.2023	Виконано
6.	Оформлення розділу «Підходи до розробки чат-ботів та аналіз активності користувачів»	08.12.2023-10.12.2023	Виконано
7.	Оформлення розділу «Реалізація оновленої версії інформаційного чат-бота ТНТУ»	11.12.2023-13.12.2023	Виконано
8.	Виконання завдання до підрозділу «Охорона праці»	14.12.2023-15.12.2023	Виконано
9.	Виконання завдання до підрозділу «Безпека в надзвичайних ситуаціях»	16.12.2023-17.12.2023	Виконано
10.	Оформлення кваліфікаційної роботи	18.12.2023-19.12.2023	Виконано
11.	Нормоконтроль	19.12.2023-20.12.2023	Виконано
12.	Перевірка на плагіат	21.12.2023	Виконано
13.	Попередній захист кваліфікаційної роботи	22.12.2023	Виконано
14.	Захист кваліфікаційної роботи	28.12.2023	

Студент

_____ (підпис)

Шишкіна В.О.

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

Козбур Г.В.

_____ (прізвище та ініціали)

АНОТАЦІЯ

Аналіз та вдосконалення Інформаційного чат-боту ТНТУ засобами Python та Aiogram // Кваліфікаційна робота освітнього рівня «Магістр» // Шишкіна Вікторія Олександрівна // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНм-61 // Тернопіль, 2023 // С.81, рис. – 17, табл. – 0, кресл. – 0, додат. – 3, бібліогр. – 61.

Ключові слова: чат-бот, аналіз, Telegram, Python, користувач, повідомлення, месенджер, Aiogram.

Кваліфікаційна робота присвячена аналізу та вдосконаленню інформаційного чат-боту для студентів ТНТУ засобами мови програмування Python та AIOGram.

Метою роботи є покращення інформаційного чат-боту для студентів ТНТУ та його аналіз, основне завдання якого допомога студентам у навчальному процесі та спрощення процесу адаптації.

В першому розділі кваліфікаційної роботи розглянуто актуальність використання чат-ботів, сфери використання та чат-боти зі штучним інтелектом. Також, було наведено приклади найпопулярніших чат-ботів у різних соціальних мережах.

В другому розділі кваліфікаційної роботи описано процес створення сценарію розробки чат-бота, наведено прикладне використання бібліотеки pyTelegramBotAPI, проведено порівняльний аналіз бібліотек для розробки чат-бота у месенджері Telegram та дослідження активності користувачів Інформаційного чат-бота ТНТУ.

В третьому розділі кваліфікаційної роботи проведено вдосконалення Інформаційного чат-бота ТНТУ із застосуванням бібліотеки AIOGram,

реалізацію запису аналітичних даних активності користувачів та доповнення функціональності Інформаційного чат-бота ТНТУ.

В четвертому розділі розглядаються питання значення адаптації в трудовому процесі, психофізіологічного розвантаження для працівників, соціального значення охорони праці та структуру цивільного захисту ТНТУ імені Івана Пулюя.

ANNOTATION

Analysis and improvement of TNTU Information chatbot by means of Python and Aiogram // Qualification work degree «Master» // Shyshkina Victoria // Ternopil Ivan Pul'uj National Technical University, Department of Computer Information Systems and Software Engineering, Department of Computer Science // Ternopil, 2023 // P. – 81, Fig. – 17, Tables – 0, Annexes. – 0, References –61 .

Key words: chat-bot, analysis, Telegram, Python, user, message, messenger, Aiogram.

The qualification work is devoted to the analysis and improvement of an information chatbot for students of TNTU using Python and AIOGram.

The aim of the work is to improve the information chatbot for students of TNTU and analyze it, the main task of which is to help students in the educational process and simplify the adaptation process.

The first chapter of the qualification work discusses the relevance of using chatbots, areas of use, and chatbots with artificial intelligence. It also provides examples of the most popular chatbots on various social networks.

The second section of the qualification work describes the process of creating a chatbot development scenario, example of the use of the pyTelegramBotAPI library, a comparative analysis of libraries for developing a chatbot in the Telegram messenger and a research of user activity of the TNTU Information Chatbot.

The third chapter of the qualification work is devoted to the improvement of the Information Chatbot of TNTU using the AIOGram library, the implementation of recording analytical data of user activity and the addition of functionality to the Information Chatbot of TNTU.

The fourth chapter discusses the importance of adaptation in the labor process, psychophysiological relief for employees, the social significance of labor protection, and the structure of civil protection of the TNTU.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ШІ – Штучний інтелект.

AI (англ. Artificial intelligence) – Штучний інтелект.

API (англ. Application Programming Interface) – Програмний інтерфейс застосунку.

HTML (англ. HyperText Markup Language) – Мова розмітки гіпертексту.

HTTP (англ. HyperText Transfer Protocol) – протокол передачі гіпертексту.

NLP (англ. Natural Language Processing) – Нейролінгвістичне програмування.

UI (англ. User Interface) – Інтерфейс користувача.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД СУЧАСНОГО СТАНУ ВИКОРИСТАННЯ ЧАТ-БОТІВ.....	12
1.1 Актуальність використання чат-ботів.....	12
1.2 Сфери використання чат-ботів.....	13
1.3 Чат-боти зі штучним інтелектом	15
1.4 Приклади найпопулярніших чат-ботів у різних соціальних мережах	23
1.5 Висновок до першого розділу	30
РОЗДІЛ 2. ПІДХОДИ ДО РОЗРОБКИ ЧАТ-БОТІВ ТА АНАЛІЗ АКТИВНОСТІ КОРИСТУВАЧІВ.....	31
2.1 Сценарій розробки чат-бота	31
2.2 Прикладне використання бібліотеки pyTelegramBotAPI.....	35
2.3 Порівняльний аналіз бібліотек для розробки чат-бота у месенджері Telegram	39
2.4 Дослідження активності користувачів Інформаційного чат-бота ТНТУ	41
2.5 Висновок до другого розділу	45
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ОНОВЛЕНОЇ ВЕРСІЇ ІНФОРМАЦІЙНОГО ЧАТ- БОТА ТНТУ	46
3.1 Вдосконалення інформаційного чат-бота ТНТУ із застосуванням бібліотеки AIUGram	46
3.2 Реалізація запису аналітичних даних активності користувачів	58
3.3 Доповнення функціоналу Інформаційного чат-бота ТНТУ	62
3.4 Висновок до третього розділу	65
РОЗДІЛ 4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	66
4.1 Охорона праці	66
4.1.1 Значення адаптації в трудовому процесі	66

4.1.2 Психофізіологічне розвантаження для працівників	68
4.1.3 Соціальне значення охорони праці	69
4.2 Безпека в надзвичайних ситуаціях	70
4.2.1 Структура цивільного захисту ТНТУ імені Івана Пулюя. Її аналіз	70
4.3 Висновок до четвертого розділу	73
ВИСНОВКИ.....	74
ПЕРЕЛІК ДЖЕРЕЛ.....	76
ДОДАТКИ	

ВСТУП

Актуальність теми. У сучасному світі використання чат-ботів стає все більш актуальним та необхідним засобом взаємодії з користувачами. Чат-боти виконують різноманітні завдання: відповіді на поширені запитання, обробка даних, допомога користувачу у виборі, пошук даних, технічна підтримка, проведення фінансових операцій. [1]

В сучасному вищому освітньому контексті, де інновації та технологічний прогрес відіграють ключову роль, оптимізація інформаційного чат-боту університету стає стратегічним завданням.

Актуальність теми «Аналіз та вдосконалення Інформаційного чат-боту ТНТУ засобами Python та Aiogram» спричинена активним розвитком цифрових технологій та високою конкуренцією в сфері вищої освіти. Використання Python та Aiogram в якості інструментів для реалізації цієї мети дозволяє поєднати потужні можливості мови програмування та високофункціональність бібліотеки для створення ефективного та інтуїтивно зрозумілого інтерфейсу чат-боту.

Детальний аналіз інформаційного потоку та оптимізація відповідей чат-боту покликані сприяти покращенню якості обслуговування користувачів та забезпечити високий рівень задоволення від взаємодії з університетським чат-ботом.

Швидкість та ефективність обслуговування користувачів є ключовими факторами взаємодії, інформаційні чат-боти в університетах в свою чергу забезпечують швидкий та зручний доступ до необхідної інформації. [2] Відсутність адаптивності та оптимізації чат-бота може призвести до втрати конкурентоспроможності та недостатньої задоволеності користувачів.

Мета і задачі дослідження. Метою даної кваліфікаційної роботи освітнього рівня «Магістр» є аналіз роботи інформаційного чат-боту для студентів ТНТУ та його покращення.

Для досягнення цієї мети необхідно було виконати такі завдання:

- проаналізувати актуальність використання чат-ботів;
- опрацювати сфери використання чат-ботів;
- дослідити чат-ботів із штучним інтелектом;
- розглянути приклади найпопулярніших чат-ботів у різних соціальних мережах;
- опрацювати сценарій розробки чат-бота;
- проаналізувати прикладне використання бібліотеки pyTelegramBotAPI;
- провести порівняльний аналіз бібліотек для розробки чат-бота у месенджері Telegram;
- дослідити активність користувачів Інформаційного чат-бота ТНТУ;
- провести вдосконалення Інформаційного чат-бота ТНТУ із застосуванням бібліотеки AIOGram;
- виконати реалізацію запису аналітичних даних активності користувачів.

Об’єкт дослідження процеси збирання та опрацювання даних щодо розробки та вдосконалення чат-ботів.

Предмет дослідження методи аналітичного опрацювання даних щодо вдосконалення чат-ботів, збирання та опрацювання користувацьких даних.

Наукова новизна одержаних результатів кваліфікаційної роботи полягає у тому, що отримало подальший розвиток дослідження та вдосконалення чат-ботів, призначених для використання у освітніх закладах.

Практичне значення одержаних результатів. Практичне значення вміщає в себе аналіз та вдосконалення інформаційного чат-боту ТНТУ, основним призначенням якого є підвищення ефективності та конкурентоспроможності чат-бота.

Завдяки розширенню функціональності та вдосконаленню інтерактивності чат-боту стимулюється залучення та утримання уваги студентів, викладачів і інших учасників університетської спільноти.

Апробація результатів магістерської роботи. Основні результати проведених досліджень обговорювались на VI Міжнародній студентській

науково - технічній конференції "Природничі та гуманітарні науки. Актуальні питання" Тернопільського національного технічного університету імені Івана Пулюя (м. Тернопіль, 2023 р.) та авторському праві на офіційний чат-бот Тернопільського національного технічного університету імені Івана Пулюя.

Публікації. Основні результати кваліфікаційної роботи опубліковано у праці конференції (Див. додаток Б).

Структура й обсяг кваліфікаційної роботи. Кваліфікаційна робота складається зі вступу, чотирьох розділів, висновків, списку літератури з 61 найменувань та 3 додатків. Загальний обсяг кваліфікаційної роботи складає 81 сторінку, з них 54 сторінки основного тексту, який містить 17 рисунків.

РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД СУЧАСНОГО СТАНУ ВИКОРИСТАННЯ ЧАТ-БОТІВ

1.1 Актуальність використання чат-ботів

У наш час чат-боти набирають все більшої популярності у різних сферах. Впровадження чат-ботів значно спрощує життя працівників різних фірм. Чат-боти допомагають працівникам скоротити час розмови із клієнтами щодо найчастіших запитань та зосередитися на більш складних завданнях. [3]

Наприклад, коли клієнти телефонують запитати ціну послуги і таких дзвінків протягом дня надходить дуже багато, то набагато простіше щоб на це питання відповів чат-бот. Звісно таку інформацію можна розмістити і на веб-сайті компанії або інших ресурсах, але за допомогою бота необхідну інформацію буде набагато простіше знайти у порівнянні з веб-сайтом, де міститься дуже багато інформації. Тобто чат-боти не тільки скорочують час працівників на вирішення питань але й користувачів. [4]

Чат-боти працюють у режимі реального часу, що є важливо. Люди отримують відповідь одразу, а не очікують, поки з'явиться працівник, який зможе відповісти на їхні запитання. У багатьох сферах це є досить важливим фактором, особливо коли людині потрібно швидко щось дізнатися, бо у випадку довгого очікування клієнт може перейти до конкурентів.

Підтримка зв'язку з клієнтами цілодобово та без вихідних є також вагомим перевагою чат-бота. У людей можуть виникнути запитання навіть на вихідних або пізньої ночі. [5]

Важливим також є те, що чат-бот створює середовище спілкування, а не проводить пошук самостійно витрачаючи на це більше часу та зусиль. Для клієнтів такий формат є більш зрозумілим та дружелюбним, наче це відповідає реальна людина по той бік екрану.

Також, є люди-інтроверти, для яких дзвінок безпосередньо до людини є ситуацією за межами комфорту, тому їм простіше знайти відповіді на свої запитання онлайн. Текстовий формат є ідеальним варіантом через те, що не супроводжується різними соціальними стресами, немає потреби обдумувати речення та інтонацію. Деякі варіанти чат-ботів також включають в себе анонімність, через що людина себе відчуває безпечніше.

У освітньому процесі також зростає популярність використання чат-ботів. Чат-боти працюють індивідуально з кожным учнем, і відстежують їх прогрес. Вони здатні адаптуватися до рівня знань учня і надавати додаткові матеріали відповідно до потреб. Кожен учень задає власний темп і за допомогою цього досягає кращих результатів. Учні менше залежать від вчителя і здатні отримати пояснення на своє запитання миттєво. [6]

Чат-боти доступні на різних платформах, наприклад мобільних пристроях, тому користуватися ними можна в будь-який комфортний час і у будь-якому місці. Вони надають зворотній зв'язок, вказують на помилки та надають поради для покращення. Також, є чат-боти, які допомагають людям впоратись зі стресом, справлятися із тривожністю та іншими емоційними проблемами, що можуть впливати на якість життя.

1.2 Сфери використання чат-ботів

Чат-боти на сьогоднішній день охопили більшість сфер, продовжують розвиватися та набирати популярності. Їх використання особливо помітне у сфері послуг та маркетингу як потужний інструмент, за допомогою якого виконується розсилка рекламних оголошень. [7]

У клієнтському обслуговуванні боти є назамінними помічниками. За допомогою чат-ботів сервіси легко відповідають на запитання клієнтів, надають інформацію про послуги та проводять опитування щодо якості обслуговування.

Хорошим рішенням є впровадження чат-ботів для купівлі та бронювання квитків [8]. Під час подорожей туристи можуть забронювати готель, квитки на

літак, квитки на потяг і весь необхідний їм транспорт. За допомогою таких ботів клієнти моніторять квитки на конкретний потяг, зупинки потяга, час відправки, ціну, обирають місце у потязі. Боти повідомляють коли з'являється потрібний квиток.

Боти можуть допомогти обрати товар, відслідкувати замовлення, надати рекомендації, повідомити про акції та знижки, оформити замовлення. Це все є швидкою взаємодією із клієнтами, боти справляються із такими завданнями не гірше працівників.[9]

Слід відзначити використання чат-ботів у банківській справі. Все більше набирає обертів використання чат-ботів у онлайн банкінгу [10]. Чат-боти надають інформацію про баланс рахунку, банківські послуги, допомагають у вирішенні питань, детально описують послуги банку. Можна навіть створювати банківську карту онлайн та забрати у відділенні банку, також і заблокувати у випадку викрадення або втрати.

Чат-боти також можна використовувати у медичній сфері. Наприклад для запису до лікаря, дізнатися графік роботи та графік конкретного лікаря, надання інформації про симптоми та нагадування про прийом ліків. Важливо те що можна проводити онлайн консультації, що особливо поширено у психології та допомагає підтримувати психічне здоров'я.

Використання технології автоматизованого обслуговування у державних установах було б дуже доречним. На сьогоднішній час державні установи асоціюються у людей із вічними чергами, багатьма документами та витраченим часом. У більшості випадків людям доводиться приходити у держані установи по декілька разів. Цей процес можна спростити звичайною автоматизацією. Чат-бот відповідав б на найбільш поширені запитання, одразу надавав перелік документів що необхідні для процедури, повідомляв про графік роботи та можливі труднощі.

У навчальних закладах чат-боти також є досить корисними. За допомогою них можна дізнатися графік навчання, розклад занять, які документи потрібні для вступу, реквізити оплати, розміщення корпусів та

гуртожитків. В загальному для студентів цей спосіб пошуку інформації є більш комфортним аніж пошук по сайтах та дзвінки до навчального закладу.

Також, чат-боти допомагають у збиранні статистики. За допомогою них можна відслідковувати дії користувачів, інформацію про них та на основі зібраних даних виконати аналіз продукту і запропонувати можливе покращення. [11]

1.3 Чат-боти зі штучним інтелектом

Чат-бот зі штучним інтелектом є програмним забезпеченням, яке взаємодіє із користувачами за допомогою написання тексту. Це комп'ютерна програма, яка імітує людське спілкування. Їх використовують у різноманітних ситуаціях, від обслуговування клієнтів до продажів. Наприклад, штучний інтелект використовують на сайтах електронної комерції, де вони відповідають на питання про певний товар та користувачі отримують миттєву відповідь. [12]

Інфраструктура бізнесу все більше зростає та за останнє десятиліття безліч трендів та технологій підвищили конкурентність настільки, що майже кожна компанія вимушена боротися за свої кошти. Ключовими факторами зростання бізнесу є технології AI, 5G, IoT та інші. [13][14]

Чат-боти або AI є одними з технологій, які найбільше вплинули на інфраструктуру бізнесу. Більшість людей стикалися із чат-ботами на веб-сайтах компаній, наприклад як із міні боксом внизу вікна. Веб-сайти та різні додатки інтегрують чат-боти зі штучним інтелектом, щоб без участі людини відповідати на запити користувачів, що забезпечує доступне обслуговування клієнтів. Попит на чат-боти зі штучним інтелектом все більше зростає, вони можуть бути корисними для бізнесу у багатьох відношеннях.

Зі всім набором функцій, які включає в себе чат-бот зі штучним інтелектом, вони можуть видаватись незрозумілими чи магічними. Протягом років люди були зачаровані та водночас налякані штучним інтелектом. Багато сюжетів транслювали катастрофи, які могли б статися, якби машини мали

занадто багато свободи та інтелекту. Художні фільми демонстрували, як машини досягають людського або надлюдського інтелекту та загрожують існуванню людей, які їх створили. [15]

Завдяки ідеям, висунутим у науково-фантастичних фільмах, багато людей не розуміють суть роботи штучного інтелекту і розглядають його як загрозу. Наприклад, від деяких людей можна почути думку на рахунок того, що машини заберуть їх роботу. Також, певні програми неправильно класифікують та відносять їх до штучного інтелекту, яскравим прикладом цього є саме чат-боти.

Зараз чат-боти вже не є чимось новим, особливо з появою найпопулярнішого бота зі штучним інтелектом ChatGPT. ChatGPT – це програма, яка використовує потужність штучного інтелекту, інші технології та дані, для того щоб надавати користувачам відповіді, схожі на людські.

Чат-бот інтерпретує звернення користувачів природною мовою та генерує належні відповіді на них. Вони застосовуються в різних сферах – інформаційний пошук, обслуговування клієнтів, віртуальні асистенти та інше. Найкращі приклади чат-ботів на основі штучного інтелекту – Slush, Cortana, Siri тощо. Найбільш просунутими є ChatGPT, Google Bard, Jasper тощо.

Чат-боти зі штучним інтелектом – це чат-боти, які натреновані вести розмови, схожі до людських, використовуючи обробку природної мови – Natural Language Processing (NLP). [16] За допомогою NLP вони здатні інтерпретувати людську мову так, як вона написана, що дозволяє їм діяти більш-менш самостійно. Можна сказати, що програмне забезпечення чат-бота зі штучним інтелектом може розуміти мову за попередньо запрограмованими командами і відповідати на основі наявних даних. Це дозволяє відвідувачам спілкуватися формулюючи свої цілі власними словами.

Більше того, чат-боти навчаються з своїх розмов, тому з часом вони зможуть адаптувати свої відповіді до різних шаблонів та нових ситуацій. Це означає, що їх можна використовувати у різних сферах, наприклад, для аналізу почуттів клієнта або прогнозування того, що саме користувач хоче знайти на сайті.

Робота чат-бота зі штучним інтелектом базується на численних технологіях і компонентах. Існує дві основні категорії, на які поділяються компоненти чат-ботів: компоненти інтерфейсу користувача та функціональні компоненти.

Все, що пов'язано з тим, що користувач бачить і відчуває, відноситься до компонентів інтерфейсу. Компоненти інтерфейсу містять такі умови:

- інтерфейс користувача повинен мати всі візуальні компоненти, такі як кнопки, текстові поля, перемикачі тощо;
- враховується те, як все відбувається на екрані, наприклад, навігація, анімація тощо – тобто все, що покращує сприйняття веб-сайту;
- розмовний дизайн, що зосереджений на розробці того, як чат-бот буде взаємодіяти або спілкуватися з користувачами. Елементами розмовного дизайну є потік і сценарії. Частина потоку включає контекст, сутності та наміри, які визначають, що саме скаже чат-бот. Частина сценарію розробляє індивідуальність чат-бота (наприклад, те, як чат-бот щось говорить). Іншими словами дизайн розмови включає в себе визначення шляхів розмови, відповідей і шрифтів.

Функціональні компоненти чат-бота зі штучним інтелектом набагато складніші, ніж компоненти інтерфейсу. Функціональні компоненти включають в себе:

- обробку природної мови – найважливішу частину чат-бота зі штучним інтелектом. AI та NLP допомагають чат-боту розуміти вхідні дані користувача, незалежно від того, чи це текст, чи голос;
- навчання з даних – алгоритми машинного навчання, які гарантують, що чат-бот навчається на даних. Це відбувається за допомогою контрольованого та неконтрольованого навчання;
- базу знань, або інформаційне сховище, яке містить інформацію, необхідну чат-боту для відповідей на запити користувачів. Це може бути інформація про продукт, поширені запитання тощо;

– управління діалогом – функціональну складову чат-бота зі штучним інтелектом. Керування діалогом перевіряє хід розмови, враховуючи контекст, наміри та відповіді.

Потрібно розуміти, що не всі чат-боти використовують штучний інтелект. Деякі з них керуються правилами, забезпечуючи чітко структурований діалог на основі введення користувачем інформації. Вони здатні розпізнавати запитання та коментарі лише тоді, коли вони відповідають ключовим словам, обраним розробниками чат-бота. З іншого боку, бот зі штучним інтелектом використовує машинне навчання та обробку природної мови, щоб реагувати на людську поведінку, навіть якщо вона відхиляється від попередньо запрограмованого сценарію.

Найпростіший вид чат-бота – це скриптовий чат-бот, який працює, відповідаючи на запити користувачів згідно встановлених правил, чат-бот зі штучним інтелектом працює дещо інакше. Звичайні чат-боти не здатні навчатися або адаптуватися, вони мають заздалегідь визначений перелік відповідей, які вони можуть використовувати на основі ключових слів, що містяться в запитанні клієнта. Чат-бот зі штучним інтелектом комбінує машинне навчання та NLP, щоб зрозуміти запит користувача та відповісти більш натуральною мовою.

На відміну від чат-ботів, що керуються правилами, чат-боти зі штучним інтелектом навчаються за допомогою алгоритмів машинного навчання, щоб вони могли коректно відповідати на відкриті запитання [17]. Налаштування таких ботів займає більше часу, але вони можуть навчатися на взаємодії користувачів та покращувати свої відповіді. Вони можуть дізнатися набагато більше про користувачів продукту і ефективно застосовувати ці знання без постороннього втручання. Це також дає торговим представникам багато інформації про покупців, щоб вони могли краще персоналізувати свої власні розмови.

На жаль, чат-боти часто продаються як штучний інтелект, що призводить до величезної плутанини для бізнесу. Хоча чат-боти мають місце на ринку для

обробки елементарних запитань, та помилково плутати їх зі справжнім ШІ, адже чим складнішим стає запит, тим менш успішним є чат-бот.

Крім того, через нездатність чат-ботів навчатися з часом, бот не зробить висновків зі своєї помилки і не зробить краще наступного разу. Замість цього він продовжуватиме пропонувати ті самі відповіді, доки людина, яка запрограмувала цього бота, не додасть до його списку більш складні відповіді. [18]

Можна сказати, що чат-боти зі штучним інтелектом звичайно більш корисні для бізнесу в швидкозмінних галузях, де типи запитань, які задає користувач, можуть змінюватися. Чат-бот зі штучним інтелектом може використовувати попередні розмови, документацію від служби підтримки та будь-які інші дані для визначення мови, контексту та намірів. [19]

У порівнянні з чат-ботом, що базується на правилах, чат-бот зі штучним інтелектом має кілька додаткових переваг. До них відносяться:

- Доступність – використовуючи чат-бота зі штучним інтелектом замість людини, компаніям стає набагато простіше забезпечувати доступність 24/7. Щоразу, коли споживач звертається із запитанням, чат-бот готовий надати миттєву відповідь. Спираючись на інтегровані системи та бази даних, він дійсно може допомогти клієнту;

- Адаптивність – незалежно від того, чи йдеться про обслуговування клієнтів або підтримку персоналу, генерацію лідів або допомогу в продажах, чат-боти зі штучним інтелектом мають безліч застосувань;

- Дешева вартість – одним з головних чинників використання чат-ботів зі штучним інтелектом є їхня доступність. Загальна вартість володіння низька порівняно з наймом додаткових співробітників;

- Клієнтський досвід – збираючи безліч додаткових даних, чат-боти зі штучним інтелектом можуть значно покращити клієнтський досвід. Сприяючи більшій кількості взаємодій, вони можуть спростити шлях клієнта, забезпечити персоналізацію в режимі реального часу і дати компаніям можливість збирати дані, які можна використовувати для поліпшення продуктів і послуг. Завдяки

добре налаштованому, вдосконаленому боту зі штучним інтелектом клієнт може пройти весь шлях клієнта в діалогах. [20]

З іншого боку, для коректного використання чат-бот зі штучним інтелектом повинен мати доступ до величезної бази розмовних даних. Саме тому чат-боти зі штучним інтелектом проходять період навчання, під час якого програміст навчає його розуміти контекст слів людини. Саме це розуміння дозволяє чат-боту відповідати на складні запитання в природній, розмовній манері.

Хоча звичайні чат-боти не такі гнучкі, як їхні аналоги зі штучним інтелектом, вони мають свої переваги. Якщо клієнт бажає створити передбачуваний, контрольований сценарій діалогу, чат-боти на основі правил дозволять спрямувати аудиторію до конкретних цілей – розмова з людиною, завантаження контенту або підписки на демо-версію.

Однак чат-боти на основі правил не запрограмовані реагувати на зміну мови. Якщо відвідувач зайде на сайт і запитає щось, на що розробники не налаштували відповідь, чат-бот не зможе надати відповідь.

Необхідно переконатися, що клієнт обирає рішення, яке передбачає певне розуміння термінів і знань, характерних для цільової галузі. Загальний чат-бот не може бути одразу готовий до використання, тому потрібно врахувати кількість часу, необхідного для підготовки бота до роботи.

Розробка чат-ботів зі штучним інтелектом вимагає досить великої кількості часу та зусиль. Створити чат-бот із ШІ – це справа не одного дня, та вимагає точного аналізу та уваги до деталей. Сама розробка проводиться в декілька етапів (див. рисунок 1.1).

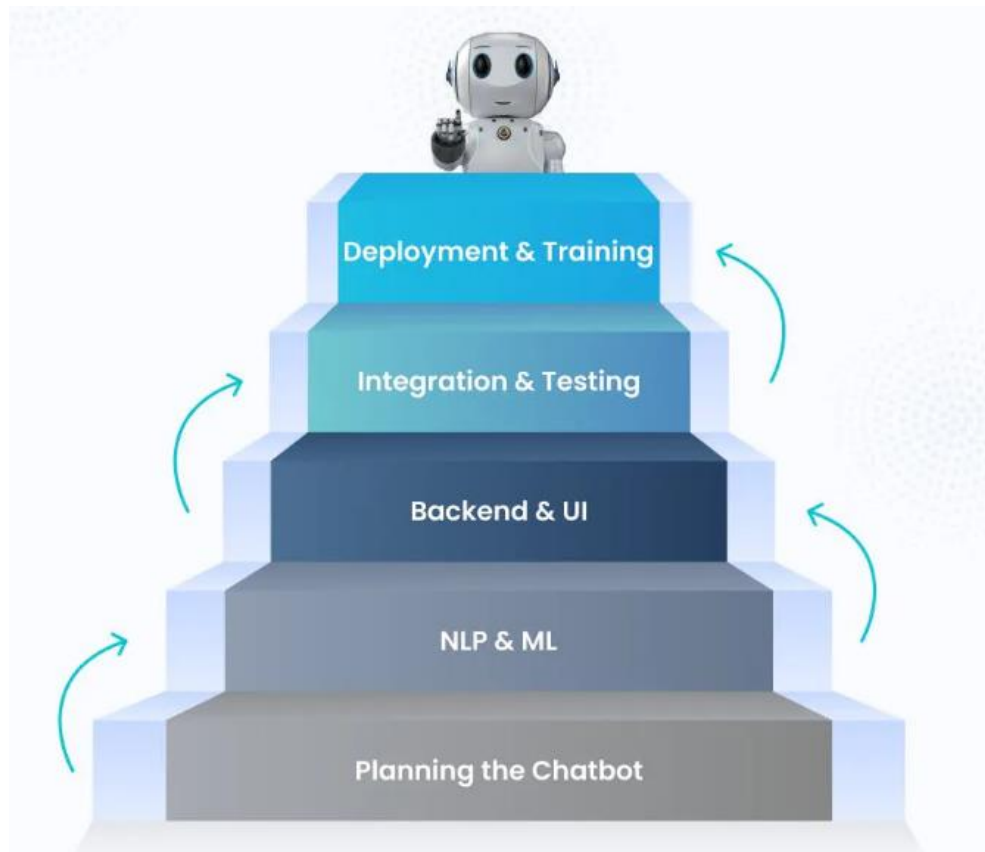


Рисунок 1.1 – Етапи створення чат-бота з ШІ

Розглянемо етапи розробки чат-бота зі штучним інтелектом: [21]

1. Планування чат-бота. Визначається його мета, сфера застосування, інструменти, які будуть використовуватися, функції тощо. Потрібно з'ясувати, на які питання він відповідатиме і як він відповідатиме на них. Крім того, потрібно знати про діалоговий потік чат-бота для забезпечення кращого UX. Можна використовувати різні мови програмування, такі як Python або Javascript, а також фреймворки, такі як Dialogflow, IBM Watson Assistant, Amazon Lex тощо.

2. NLP та ML. Обробка природної мови та машинне навчання є основою технології штучного інтелекту. Співпраця цих двох технологій робить чат-бота придатним для використання. NLP гарантує, що чат-бот правильно інтерпретує запити користувача. На противагу цьому, машинне навчання – це технологія, яка дозволяє чат-боту вчитися з часом, вивчаючи та аналізуючи дані. Такі методи, як нейронні мережі, дерева рішень і навчання з підкріпленням, можуть

бути використані для реалізації машинного навчання в чат-боті зі штучним інтелектом.

3. Backend та UI. Внутрішня частина чат-бота або backend – це та частина, де знаходяться всі функціональні можливості. Вона відповідає за отримання запиту, його обробку та генерування відповіді. Інша частина – розробка користувацького інтерфейсу або UI. Інтерфейс повинен бути візуально привабливим та інтерактивним, щоб утримувати увагу користувача. Можна повністю розробити новий інтерфейс або інтегрувати його з такими платформами, як Facebook, Telegram або Slack.

4. Інтеграція та тестування. Інтеграція чат-бота передбачає його поєднання з іншими системами, такими як CRM, системами email-маркетингу, електронної комерції тощо. Можна інтегрувати чат-бота з цими системами, а також з такими технологіями, як NLP, розпізнавання голосу, аналіз настроїв тощо, щоб надати йому необхідну функціональність. Тестування гарантує, що відповіді чат-бота будуть доречними і не будуть написані з помилками.

5. Впровадження та спостереження. Після впровадження чат-бота на публічне використання необхідно спостерігати за його роботою та продуктивністю. Крім того, можна використовувати інструмент аналітики для відстеження та аналізу критичних параметрів, таких як продуктивність, взаємодія з користувачами, вузькі місця тощо.

Машинне навчання зараз передбачає використання нейронних мереж. Теоретично, нейронні мережі імітують нейрони людського мозку, що дозволяє ШІ вчитися так само, як це робить людина. Алгоритми отримують великі обсяги інформації, включаючи вхідні та вихідні дані. В даному випадку вхідною інформацією є запитання клієнтів, а вихідною – відповіді на ці запитання, що історично надавалися агентами з обслуговування клієнтів. Ця інформація включає електронні листи, стенограми чатів і всі відповідні метадані.

1.4 Приклади найпопулярніших чат-ботів у різних соціальних мережах

Завдяки стрімкому розвитку технологій штучного інтелекту та розширенню можливостей чат-платформ, чат-боти стають невід'ємною частиною нашого життя. [22] Розглянемо кілька зразків, які користуються популярністю серед користувачів.

Першою соціальною мережею яку ми розглянемо є Viber – популярний месенджер, де користувачі можуть обмінюватися повідомленнями, дзвінками, мультимедіа та іншим. В Україні Viber є досить поширеним додатком, який встановлений майже на кожному телефоні, найчастіше його використовує покоління середнього та старшого віку.

У Viber чат-боти розробляються в основному за допомогою конструктора. Є можливість створити бота по шаблону або з нуля під власні потреби. [23] Viber-бот – це інструмент для автоматизації різних бізнес-процесів, від реагування на стандартні запитання до реєстрації на події чи отримання картки постійного клієнта. Спілкування з ботом є можливим завдяки передбаченим швидким відповідям, кнопкам та простому меню. Крім того, розмір, розташування та дизайн окремих елементів меню можна відредагувати відповідно до потреб.

Чат-боти у Viber можуть як відповідати на запитання згідно сценарію так і на більш розгорнуті запитання за допомогою штучного інтелекту GPT-3. Доступним є і чат з оператором, де є можливість поділитися фото, відео або файлом. Можна обирати товар як на веб-сайті, переглядаючи пропозиції, додаючи у кошик та оплачуючи прямо у додатку. Крім того, у додатку можна реєструватися, переглядати історію замовлень, статус замовлення, бонуси та переходити на сайт компанії.

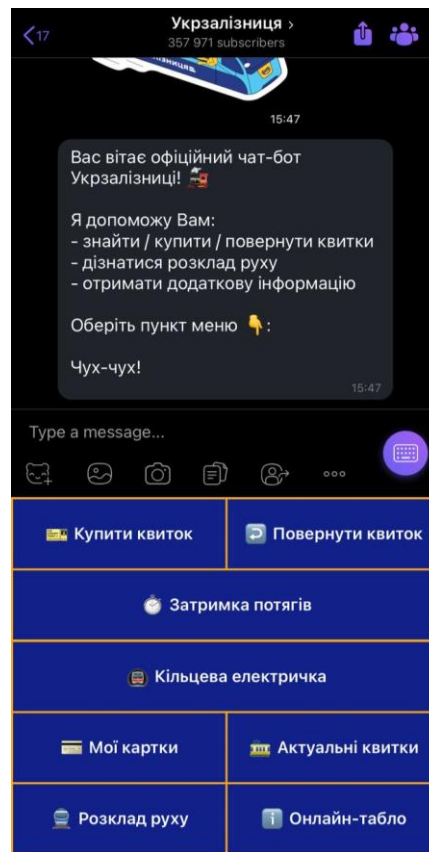


Рисунок 1.2 – Чат-бот Укрзалізниця

Розглянемо для прикладу чат-бот Укрзалізниця у Viber (див. рис. 1.2). У боті використовуються кнопки, за допомогою яких ми дізнаємося інформацію що нам необхідно. За допомогою бота можна знаходити, купляти та повертати квитки, переглядати розклад руху, дізнаватися про затримку потягів та отримувати додаткову інформацію.

Наступною соціальною мережею є Facebook, тобто Messenger за допомогою якого проходить комунікація у Facebook. Facebook є найбільшою соціальною мережею у світі та налічує мільярди користувачів. Він надає широкий спектр інструментів для бізнесу, наприклад, рекламні можливості, створення сторінок компаній та взаємодію з користувачами через чат-боти. Розширені можливості API дозволяють розробникам створювати різноманітні додатки та сервіси, що інтегруються з платформою. [24]

Для створення чат-боту у Facebook можна скористатися двома способами – написання коду або використання конструкторів. У Facebook є спеціальний розділ, де перераховані акредитовані сервіси для роботи з платформою.

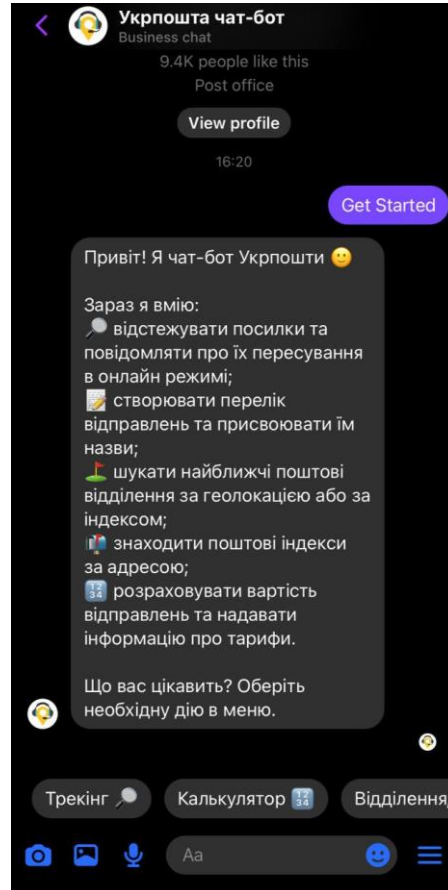


Рисунок 1.3 – Чат-бот Укрпошти

На рисунку 1.3 можна побачити чат-бот Укрпошти у Messenger [25]. Для початку спілкування потрібно натиснути на кнопку «Почати», після чого надходить повідомлення від користувача та чат-бот розпочинає свою роботу. За допомогою бота можна відстежувати посилки, знаходити найближчі поштові відділення, знаходити поштові індекси, дізнаватися тарифи, розраховувати вартість та отримати допомогу у вирішенні запитань.

Далі поговоримо про чат-боти у Telegram. Цей месенджер має широкий функціонал, високу безпеку та відкритість щодо розробників. Telegram має відкрите API, що дозволяє розробникам створювати різноманітні додатки, боти та сервіси, розширюючи можливості платформи. Під час повномасштабного

вторгнення його популярність зростає ще більше, також він є досить популярним у молодого покоління.

Чат-боти у Telegram можна створити написанням власного коду або за допомогою використання конструкторів. Вибір між конструктором та написанням власного коду залежить від конкретних потреб, навичок та вимог проекту.

Для створення спочатку необхідно зареєструвати чат-бота у боті, який контролює інші боти у Telegram – BotFather [26]. У цьому боті ви обираєте назву чат-бота, зображення, опис та найголовніше – бот надає вам ключ за допомогою якого розробник має доступ до HTTP API бота. Цей ключ є унікальним, – будь-хто, отримавши ключ зможе стати адміністратором бота.

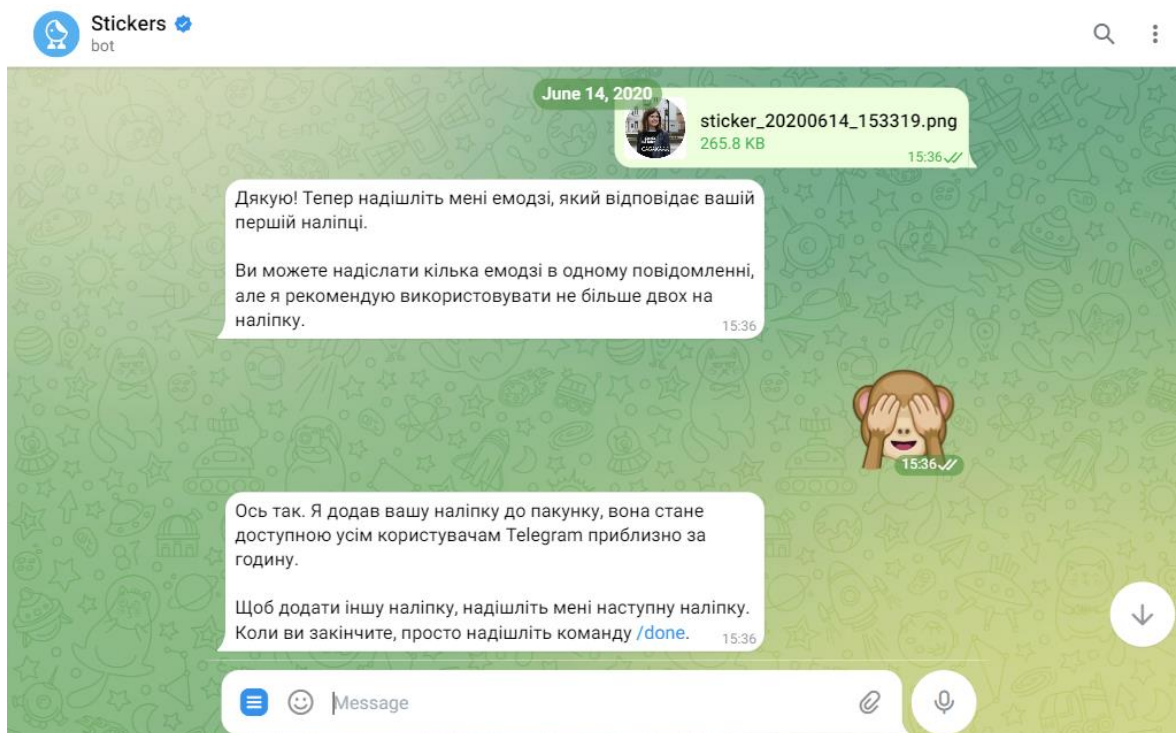


Рисунок 1.4 – Чат-бот для створення стікерів у Telegram

На рисунку 1.4 ми бачимо чат-бот для створення стікерів у Telegram. Цей чат-бот створює стікери, відеостікери, анімовані стікери, маски із файлів, які ви йому надсилаєте, також він може надавати статистику використання ваших стікерів. На початку роботи із цим ботом вам надсилається інформаційне

повідомлення із тим що вміє цей бот. Для запуску роботи вам необхідно обрати функцію /start та бот надішле повідомлення де ви зможете вибрати потрібну вам функцію із всіх.

Також, не можна випускати з уваги чат-боти на власних веб-сайтах. Зараз часто можна побачити на деяких сайтах міні-бокс із чат-ботом внизу вікна. Для того щоб створити такий чат-бот, можна так само використати два способи: написання коду та використання конструкторів. Використання конструкторів в цьому випадку є більш популярним.

Спочатку потрібно обрати платформу для створення, далі необхідно встановити віджет на свій сайт, цей етап залежно від платформи може виглядати по-різному. Зазвичай, потрібно скопіювати код віджета та додати його до сайту. Після чого, за допомогою редактора розробник адаптує наявний шаблон або створює власний варіант поведінки для чат-бота і завантажує його. [27]

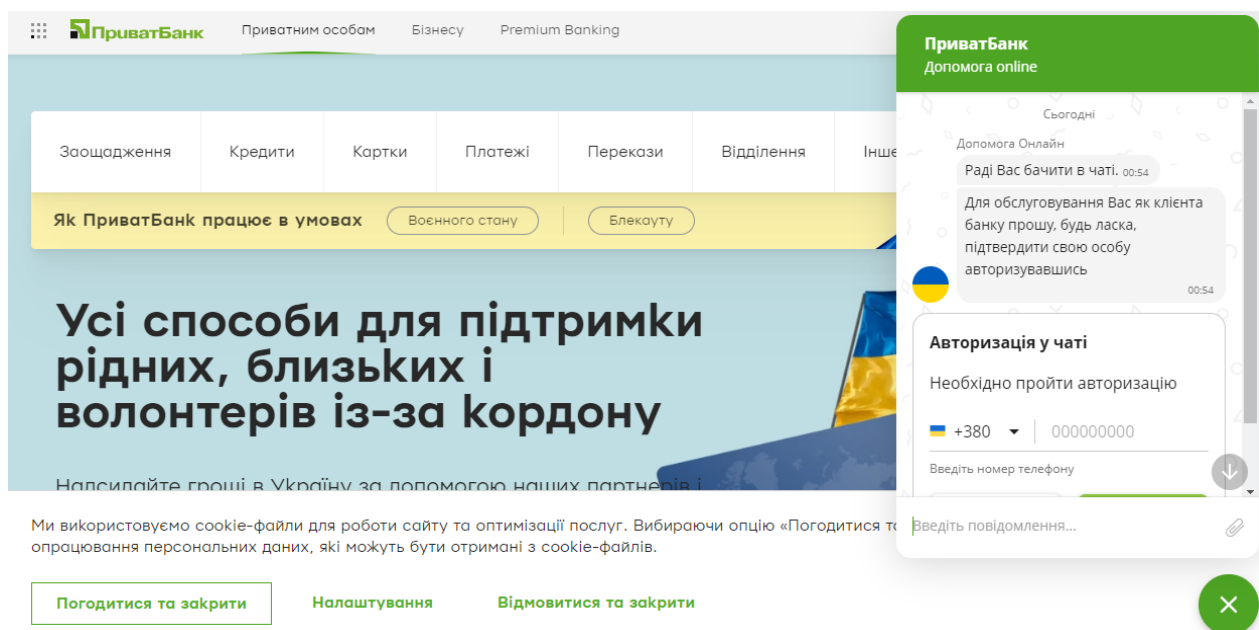


Рисунок 1.5 – Чат-бот на сайті ПриватБанку

На рисунку 1.5 зображений чат-бот на сайті ПриватБанк. Цей чат-бот допомагає користувачам у вирішенні їх запитань. Для початку необхідно авторизуватися у чаті за допомогою номеру телефону та введення паролю до

Приват24. Після чого можна описати ваше питання самостійно або обрати зі списку найпопулярніших питань, що саме вас цікавить. Чат-бот надає інформацію стосовно роботи банку під час воєнного стану, подовження терміну дії, відкриття та перевипуск карток, електронного цифрового підпису, віртуальної карти та інших питань.

Наступними ми розглянемо чат-ботів зі штучним інтелектом. Новітні дослідження говорять нам, що чат-боти зі штучним інтелектом в змозі конкурувати з людиною та можуть її навіть перевершити в креативних завданнях.

Проводився експеримент, в якому чат-боти із ШІ проявили себе ефективно в таких творчих завданнях, як мозковий штурм для альтернативного використання звичайних предметів. Експеримент оцінює дивергентне мислення, яке доволі часто зв'язане з творчістю, при такому оцінюванні враховуються такі параметри як послідовність, гнучкість, оригінальність та удосконалення. Чат-боти отримали вищі бали ніж люди, зокрема, проявившись краще за оригінальністю та креативністю. [28]

З листопада 2022 року всім відомий чат-бот зі штучним інтелектом – ChatGPT. У перші дні, як він став доступним, майже кожна людина спробувала задавати питання цьому чат-боту. Деякі люди розчарувалися відповідями бота, оскільки вони не завжди були коректними, а інші навпаки захопилися тим, як працює цей бот. [29]

Для використання ChatGPT потрібно спершу зареєструватися, це можна зробити за допомогою наявного облікового запису у Google або ж створити новий. Перед початком спілкування розробники надають інформацію про переваги та недоліки бота. Далі все залежить від правильно сформованого запитання. Також варто враховувати, що бот надає більш відповідний текст, якщо з ним спілкуватися англійською мовою. Звичайно, він може відповідати також іншими мовами, зокрема українською, але розроблявся та навчався він саме англійською мовою.

ChatGPT пам'ятає весь діалог з людиною, що дозволяє йому більш коректно розуміти, в якій саме формі людина хоче отримати відповідь. Він може пояснювати речі більш складними формулюваннями або простими словами. Також цей бот володіє креативністю, добре працює із текстом та навіть може генерувати код. Слід розуміти, що ChatGPT також часто помиляється, він не володіє всією інформацією. Чат-бот навчався на англійськомовних даних, не є підключеним до інтернету та цей продукт постійно вдосконалюється.

На рисунку 1.6 показана взаємодія із чат-ботом ChatGPT. Можемо побачити, що бот описав способи, за допомогою яких можна відкрити пляшку води.

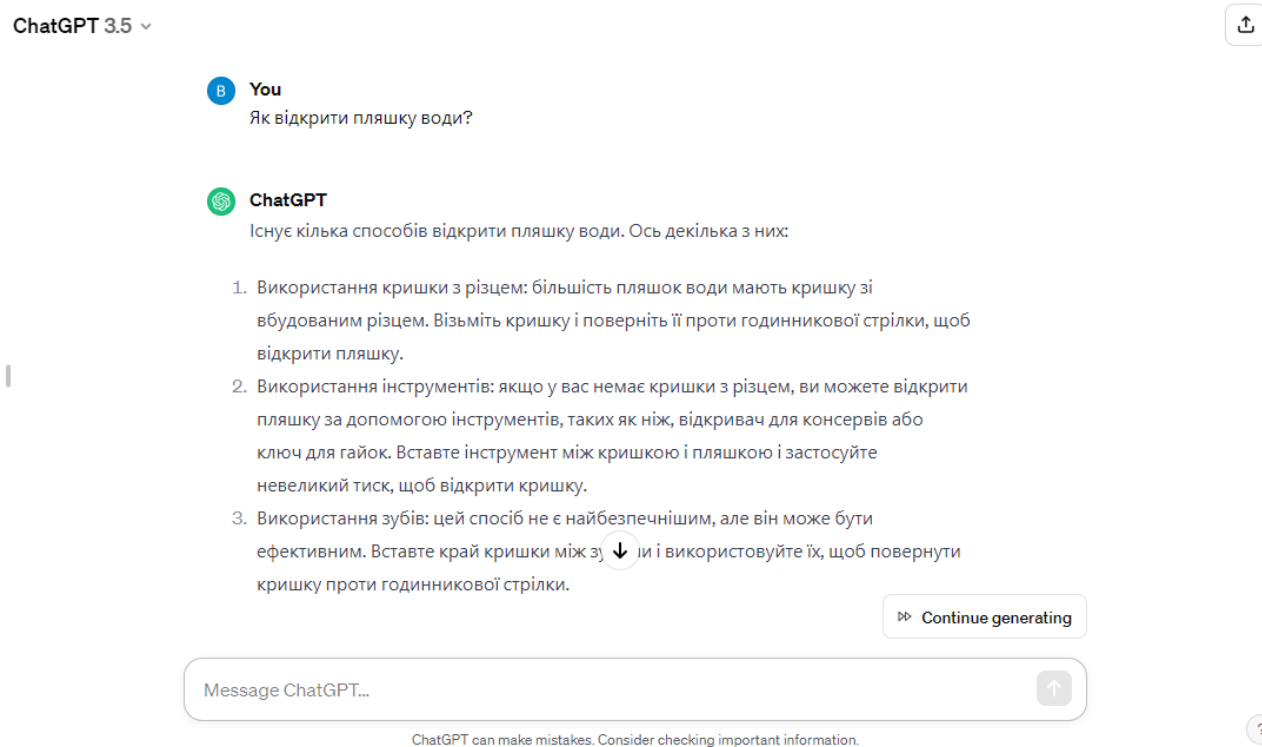


Рисунок 1.6 – Діалог із ChatGPT

На достовірність інформації цього чат-бота хоча й не можна повноцінно розраховувати, але він може стати незамінним помічником при правильному використанні.

1.5 Висновок до першого розділу

У першому розділі кваліфікаційної роботи було розглянуто актуальність використання чат-ботів, сфери використання та чат-боти зі штучним інтелектом. Крім того, було наведено приклади найпопулярніших чат-ботів у різних соціальних мережах.

Таким чином було визначено, що чат-боти широко використовуються у різних сферах та месенджерах і залишаються актуальними в сучасному світі. Слід відзначити, що чат-боти зі штучним інтелектом досягли високого рівня розвитку і продовжують вражати своїми можливостями. Тим не менше, розробка чат-ботів зі штучним інтелектом вимагає високих навичок, а процес виготовлення великої кількості часу та зусиль. Отож, чат-боти залишаються інноваційним і важливим інструментом, але їх вдосконалення потребує великого вкладу в технічні знання та ресурси.

РОЗДІЛ 2. ПІДХОДИ ДО РОЗРОБКИ ЧАТ-БОТІВ ТА АНАЛІЗ АКТИВНОСТІ КОРИСТУВАЧІВ

2.1 Сценарій розробки чат-бота

Для створення чат-бота потрібно спочатку скласти сценарій взаємодії користувача з чат-ботом. Сценарій є важливим елементом при створенні, саме сценарій визначає на які запитання бот може відповідати, які команди розуміє, і яким чином він реагує на ситуації.

Розглянемо основні етапи створення сценарію для розробки чат-бота: [30]

1. Визначення мети. Перед створенням необхідно визначити для чого саме і чому користувач може використати чат-бота. Крім того, необхідно визначити, як саме чат-бот може допомогти користувачеві у досягненні мети.

2. Аналіз реальних прикладів використання. Після визначення мети створення чат-боту необхідно проаналізувати потенційних конкурентів. Переглянувши існуючі чат-боти потенційних конкурентів можна визначити, що саме необхідно врахувати при створенні власного чат-бота, оцінити слабкі місця конкурентів та визначити можливі способи покращення цих недоліків, за наявності більш детальної інформації про функціонування чат-бота конкурентів є можливість оцінити потенційну цільову аудиторію та охоплення.

3. Створення сценарію використання. Необхідно розробити сценарій взаємодії користувача з ботом у якому будуть визначатися ключові елементи розвитку діалогу. Визначитися яка основна інформація буде відображатися та яким чином. Створити сценарій можна способом візуального зображення меню у вигляді схеми, щоб чітко представити, яким чином користувачі будуть взаємодіяти з ботом та які опції будуть доступні у головному меню. Це допоможе зрозуміти, що саме повинен містити готовий бот. Приклад створеного сценарію наведено на рисунку 2.1.

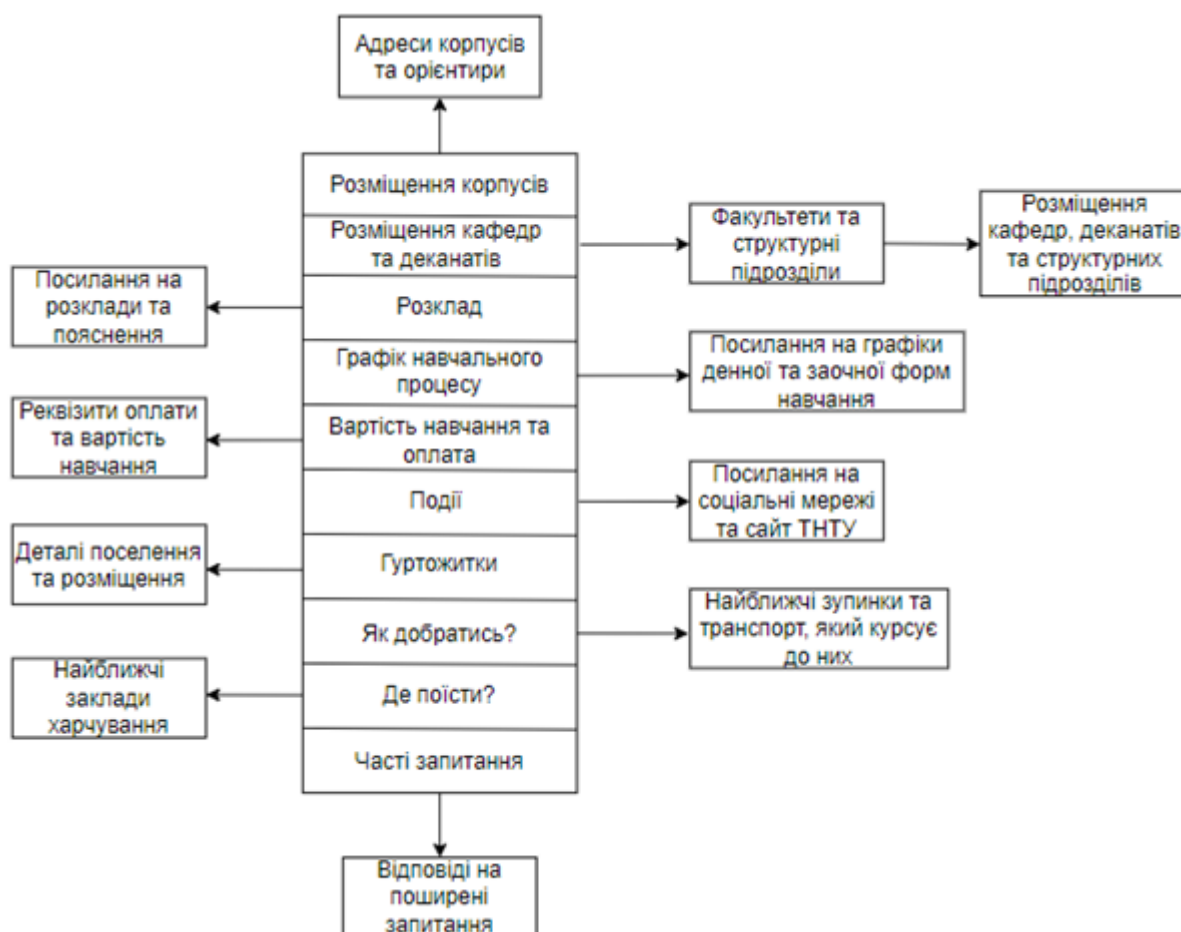


Рисунок 2.1 – Сценарій використання чат-бота

4. Визначення середовища. Потрібно обрати де саме чат-бот буде взаємодіяти із користувачами. Бот може бути розміщений у соціальній мережі або месенджері, на власному сайті та навіть можна створити окремий сайт для чат-бота.

5. Визначення способу розробки. Існують різні способи розробки чат-ботів. Наприклад, зазвичай чат-ботів розробляють з використанням конструктора, мінімальним редагуванням шаблону або написання програмного коду чат-бота з нуля. Вибір способу розробки залежить від наявних ресурсів, навичок персоналу, бажаного функціоналу та деталей розробки. [31]

6. Визначення цільової аудиторії. Визначення цільової аудиторії є досить важливим етапом у створенні чат-бота, адже саме користувачі будуть взаємодіяти із ним. Потрібно розуміти, хто саме є клієнтом, хто саме буде

надавати перевагу розробленому продукту, де користувачам комфортніше спілкуватися із ботом, які їх очікування тощо.

7. Вибір особистості для бота та вид спілкування. Для створення вигляду спілкування з реальною людиною потрібно зробити таким чином, щоб чат-бот поведився як людина, що надає інформацію. Є можливість обрати ім'я для чат бота та манеру спілкування. Потрібно продумати як він буде представляти себе, які запитання він може запитувати, як він буде відповідати у випадку питань які які він не розуміє. Чат-бот який просто викидає інформацію не є цікавим співбесідником. Також, якщо цільова сфера використання має складні терміни, потрібно упростити їх для звичайних користувачів.

8. Тестування та впровадження. Після розробки чат-бота його необхідно протестувати для того, щоб зрозуміти, чи все коректно працює. Потрібно перевірити всі можливості чат-бота, зрозуміти, де є несправності та виправити їх. Для початку рекомендується надати доступ до бота декільком людям, які змогли б перевірити його роботу та вказати на те, що їм не подобається і потім, після врахування слабких місць, запускати його на ширшу аудиторію. [32]

Звичайно, потрібно зрозуміти, що саме буде відбуватися, коли користувач натискає будь-яку кнопку у бота або задає запитання. Які переходи є у створеному чат-боті, як вони відбуваються, що можна спростити, а що вимагає більш детальної інформації.

Також, можна додати аватар для бота, щоб створити емоційний контакт із користувачами. Створений аватар фактично є відображенням бренду, його місії та цінностей, і за допомогою зображення зміцнюється лояльність користувачів і чат-бот виділяється серед інших.

При створенні також надається перевага чат-ботам із кнопками тому, що при введенні вручну запитання, користувач може стомитися та йому просто набридне. На противагу цьому можна використати загальні назви кнопок, за допомогою яких можна задовільнити більшість питань користувачів. [33]

Для більш дружньої взаємодії можна додати емодзі до тексту. Таким чином, бот стає більш людським і з ним приємніше спілкуватися. Необхідно

використовувати емодзі за допомогою асоціації їх із текстом, проте в оптимальній кількості.

Виникають ситуації, коли пояснити інформацію текстом набагато складніше та об'ємніше. Великий обсяг тексту може відштовхнути людину, зазвичай користувачі не хочуть бачити у відповідь текст, що розтягнувся на весь екран. В таких ситуаціях, потрібно не боятися використовувати зображення, деколи їх використання просто необхідне та надає краще взаєморозуміння з користувачами.

Початок створення чат-бота визначає ключові моменти, що визначатимуть його подальший розвиток та успіх у взаємодії з користувачами. Важливо належним чином розробити концепцію та функціонал чат-бота, враховуючи його мету та цільову аудиторію. Необхідно детально проаналізувати потреби користувачів та передбачити можливі сценарії взаємодії, забезпечуючи інтуїтивний та легкий процес спілкування.

Також необхідно враховувати технічні аспекти створення чат-бота. Вибір технологій, платформи для розгортання, архітектури системи та забезпечення безпеки даних – усі ці складові визначають ефективність та стабільність чат-бота [34]. Визначення структури та логіки взаємодії з базою даних також є одним з ключових аспектів, спрямованим на забезпечення ефективного зберігання та обробки інформації.

Таким чином, на старті розробки чат-бота слід обдумано інтегрувати всі аспекти, враховуючи як функціональні, так і технічні вимоги, для створення продукту, який буде не лише відповідати потребам користувачів, але й бути готовим до майбутніх викликів та розширень.

Створення сценарію розробки для чат-бота є критичним етапом, оскільки від нього залежить успішність та ефективність всього процесу створення. Це дозволяє команді чітко визначити стратегію та орієнтуватися на конкретних цілях, які необхідно досягти. Сценарій є основою для розробки та визначення функціоналу чат-бота, допомагаючи уникнути випадковості та концентруючи зусилля на ключових аспектах проекту.

Крім того, створення сценарію сприяє оптимізації користувацького досвіду, оскільки це дозволяє уникнути плутанини та забезпечити логічний порядок взаємодії з чат-ботом. Користувачам стає зручніше та простіше взаємодіяти з ботом, що впливає на їхню задоволеність та лояльність.

Узгоджений сценарій розробки також сприяє ефективному використанню технічних ресурсів. Розробники можуть чітко розуміти завдання та вимоги, що дозволяє раціонально розподіляти зусилля та уникати зайвих витрат часу та енергії. [35]

Не останнє місце важливості сценарію є в підтримці майбутньої інтеграції та розширення. Він створює основу для подальшого розвитку проєкту, дозволяючи легко впроваджувати нові функції та розширювати функціонал, що є важливим для динамічних змін у вимогах та умовах використання.

2.2 Прикладне використання бібліотеки pyTelegramBotAPI

Бібліотека pyTelegramBotAPI – є популярною бібліотекою для розробки чат-ботів. Розробники надають перевагу цій бібліотеці через добре прописану документацію та хорошу взаємодію з Telegram API. [36]

Для початку нам потрібно встановити бібліотеку pyTelegramBotAPI, тому у командному рядку потрібно написати `pip3 install pytelegrambotapi`.

Далі імпортуємо бібліотеки та вписуємо токен який отримали для боту у BotFather:

```
import telebot
bot = telebot.TeleBot('token')
```

За допомогою цієї бібліотеки можна використовувати два види кнопок: `inline` та `keyboard`. [37]

Для використання кнопок імпортуємо типи:

```
from telebot import types
```

Для inline кнопок використовується метод `InlineKeyboardMarkup`. [38]

Приклад використання методу `InlineKeyboardMarkup` наведено у лістингу 2.1.

Лістинг 2.1 – Використання методу `InlineKeyboardMarkup`

```
@bot.message_handler(commands = ['start'])
def url(message):
    markup = types.InlineKeyboardMarkup()
    btn1 = types.InlineKeyboardButton(text='Сайт ТНТУ', url='
https://tntu.edu.ua/?p=uk/main ')
    markup.add(btn1)
    bot.send_message(message.from_user.id, "Для переходу на
сайт ТНТУ, натисніть кнопку", reply_markup = markup)
```

Для keyboard кнопок використовується метод `ReplyKeyboardMarkup`.

Приклад використання методу `ReplyKeyboardMarkup` наведено у лістингу 2.2.

Лістинг 2.2 – Використання методу `ReplyKeyboardMarkup`

```
@bot.message_handler(commands=['start'])
def start(message):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    btn1 = types.KeyboardButton("ФІС")
    btn2 = types.KeyboardButton('ФПТ')
    markup.add(btn1, btn2)
    bot.send_message(message.from_user.id, "Оберіть
факультет", reply_markup=markup)
```

Гіперпосилання створюється за допомогою запису цільового слова у квадратних дужках, а посилання – в круглих. У кінці методу надсилання повідомлення додаємо `parse_mode='Markdown'`:

```
bot.send_message(message.from_user.id, 'Розклад за ' +
'[посиланням] (https://tntu.edu.ua/?p=uk/schedule/)',
parse_mode='Markdown')
```

Додаємо стартову команду та відповідне привітальне повідомлення.

Додавання стартової команди та привітального повідомлення наведено у лістингу 2.3.

Лістинг 2.3 – Додавання стартової команди та привітального повідомлення

```
@bot.message_handler(commands=['start'])
def start(message):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    btn1 = types.KeyboardButton("Вітання")
    markup.add(btn1)
    bot.send_message(message.from_user.id, "Привіт! Я чат-бот
ТНТУ", reply_markup=markup)
```

Опишемо реакцію бота на натискання кнопок користувачем.

Програмний код обробки натискання кнопок у чат-боті наведено у лістингу 2.4.

Лістинг 2.4 – Обробка натискання кнопок

```
@bot.message_handler(content_types=['text'])
def get_text_messages(message):

if message.text == 'Вітання':
    markup=types.ReplyKeyboardMarkup(resize_keyboard=True)
    btn1 = types.KeyboardButton('Розклад')
    btn2 = types.KeyboardButton('Розміщення корпусів')
    btn3 = types.KeyboardButton('Гуртожитки')
    markup.add(btn1, btn2, btn3)
```

```

bot.send_message(message.from_user.id, 'Оберіть що саме
вас цікавить', reply_markup=markup)
elif message.text == 'Розклад':
    bot.send_message(message.from_user.id, 'Для того, щоб
орієнтуватися у розкладі необхідно знати декілька правил, розклад
знаходить за ' + '[посиланням] (https://
https://tntu.edu.ua/?p=uk/schedule/)', parse_mode='Markdown')
elif message.text == 'Розміщення корпусів':
    bot.send_message(message.from_user.id, 'У ТНТУ є 11
корпусів', parse_mode='Markdown')
elif message.text == 'Гуртожитки':
    bot.send_message(message.from_user.id, 'У ТНТУ є 3
гуртожитки', parse_mode='Markdown')

```

Вигляд розробленого чат-бота з використанням бібліотека руTelegramBotAPI зображено на рисунку 2.2.



Рисунок 2.2 – Вигляд чат-бота

Аналогічним принципом було виконано розробку Інформаційного чат-бота ТНТУ. В процесі експлуатації було виявлено критичні недоліки використання даної бібліотеки, тому було прийнято рішення про вдосконалення чат-бота з використанням іншої бібліотеки.

2.3 Порівняльний аналіз бібліотек для розробки чат-бота у месенджері Telegram

Спочатку розглянемо бібліотеку `pyTelegramBotAPI`, її ми використовували для написання чат-бота. Ця бібліотека є однією з найпопулярніших для створення чат-ботів у Telegram. Вона є відомою завдяки широкому функціоналу та зручному інтерфейсу.

Бот, написаний на цій бібліотеці, може оперативно реагувати на повідомлення, також він підтримує різні типи повідомлень, такі як голосові, текстові, мультимедіа тощо. [39]

Чат-бот написаний на `pyTelegramBotAPI`, легко налаштовується та має можливість розширення його функціональності. Також, у бібліотеці є добре прописана документація, що значно полегшує роботу із цією бібліотекою.

Проте у цієї бібліотеки є деякі недоліки, такі як обмеженість асинхронності та можлива неактивність оновлень. Для того, щоб зробити нашого чат-бота більш асинхронним та зручним для користувачів було прийнято рішення переписати програмний код чат-бота використовуючи іншу бібліотеку.

Перш за все, необхідно порівняти існуючі варіанти бібліотек для написання чат-ботів у месенджері Telegram та обрати найбільш відповідний до наших потреб варіант. [40]

Бібліотка `python-telegram-bot` є простою у використанні та з можливістю створення чат-ботів різних типів. Вона досить легко встановлюється та налаштовується. Завдяки цій бібліотеці можна створювати розширені сценарії для відповіді на повідомлення користувачів. [41]

Однак бібліотека `python-telegram-bot` має обмежену продуктивність, особливо при великій кількості користувачів. Оновлення фреймворку також досить часто порушували роботу бота. Ця бібліотека також має проблеми з асинхронністю, довгі запити та обчислення спричиняють затримки у чат-боті.

З цього можемо зробити висновок що бібліотека `python-telegram-bot` не підходить для нашого чат-бота, перехід на цю бібліотеку здійснювати недоцільно.

Наступною бібліотекою, яку ми будемо розглядати, є `twxbotapi`, вона є ідеальним вибором для новачків у розробці через свою простоту. У цій бібліотеці можна легко інтегрувати готові функції та діалоги у чат-бота.

В `twxbotapi` швидко створюється базовий функціонал та потім можливе розширення за потребою.

Недоліками у `twxbotapi` є обмежена асинхронність та недостатня функціональність. Ця бібліотека не має таких розширених можливостей, які є необхідними у деяких випадках.

Отже, `twxbotapi` також не найкраще рішення в якості заміни бібліотеки, через те що основним критерієм у пошуку нової бібліотеки є покращення асинхронності.

Наступна бібліотека – `telepot`, відзначається простотою та високим рівнем абстракції. Для використання цієї бібліотеки не потрібне глибоке розуміння Telegram API. Підтримка асинхронності є однією з переваг цієї бібліотеки, також у бібліотеки є досить детальна документація. [42]

Однак, високий рівень абстракції обмежує деякі розширені функції Telegram API, які можуть бути доступні за допомогою інших бібліотек. Крім того, бібліотека має певний навчальний поріг через те, що вона створена для розробників із базовими знаннями Python.

Розглянемо бібліотеку `AIOGram`. Ця бібліотека є асинхронною та набула своєї популярності завдяки ефективності та здатності обробляти велику кількість запитів одночасно. [43]

AIOGram є досить швидкою та може працювати в умовах високого навантаження. При обробці багатьох запитів інші операції не блокуються, тому вона є досить популярним рішенням.

У бібліотеки AIOGram є можливість створювати власні та складні сценарії. Бот може відповідати на запити користувачів зважаючи на їх попередні дії. Вона є досить гнучкою та легко інтегрується з іншими сервісами та системами. Крім того, ця бібліотека дає змогу використовувати зовнішні API для виконання завдань. [44]

AIOGram добре працює із базами даних, тому вона є хорошим варіантом для розробки ботів, які потребують зберігання даних користувачів та інших даних.

З недоліків бібліотеки AIOGram можна виділити обмежену документацію та складність в зв'язку з використанням асинхронного Python.

Отже, можна сказати, що AIOGram підходить для нашого чат-бота, оскільки переваги цієї бібліотеки вирішують проблеми, які виникли під час експлуатації Інформаційного чат-бота ТНТУ.[45]

2.4 Дослідження активності користувачів Інформаційного чат-бота ТНТУ

Аналітика є одним із основних способів збирання та обробки різноманітних даних. За допомогою ретельного та якісного аналізу даних адміністратор будь-якої системи має змогу виділити найбільш ефективні способи покращення користувацького досвіду та вирішення загальних проблем у використанні продукту.

Таким чином, аналітика є основним елементом, який використовується під час створення стратегії розвитку. На основі аналізу даних є можливість опрацювати вплив глобальних трендів на активність користувачів, оптимізувати сценарії та функціональні процеси. Крім того, на основі використання звітів аналітики створюються припущення щодо причин

Проаналізуємо, у які дні користувачі найбільше взаємодіяли з чат-ботом натискаючи кнопки. На рисунку 2.4 зображено графік частоти взаємодії користувачів з чат-ботом за днями.



Рисунок 2.4 – Частота звернень користувачів за днями тижня

Проаналізувавши рисунок 2.4 можемо зробити висновок, що користувачі найчастіше взаємодіяли з чат-ботом у п'ятницю та понеділок, а найменше – у неділю.

Крім того, опрацюємо дані щодо годин звернення користувачів до Інформаційного чат-бота ТНТУ. Результат проведеного дослідження зображено на рисунку 2.5.

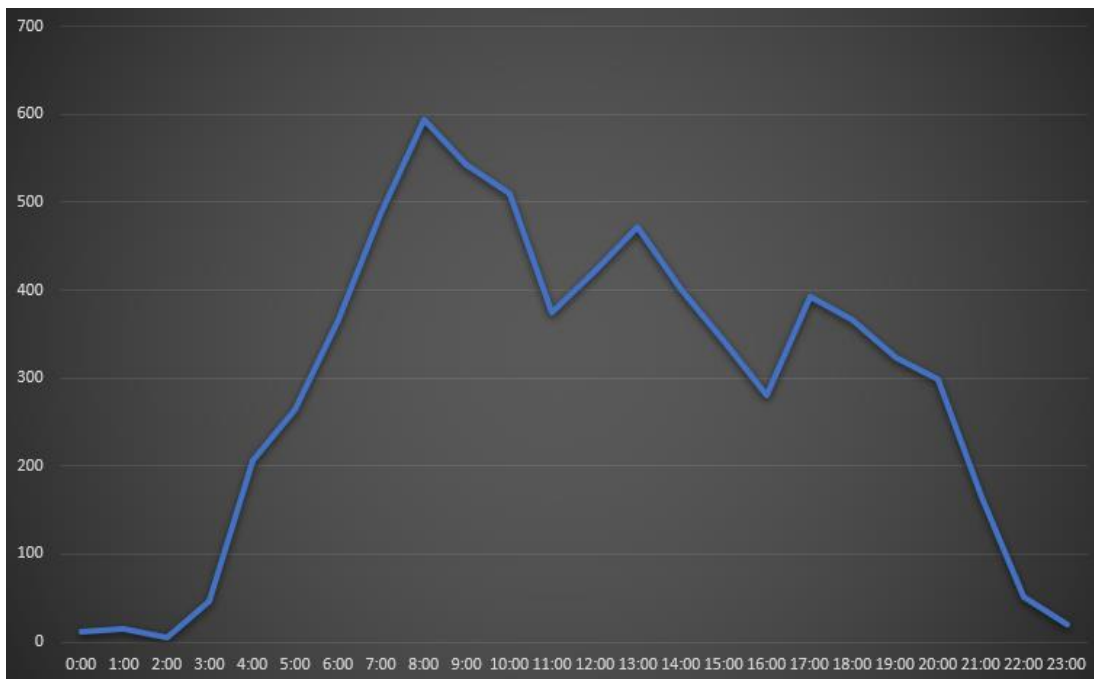


Рисунок 2.5 – Години звернення користувачів

Оглянувши результати дослідження, яке зображено на рисунку 2.5, можемо спостерігати пік активності користувачів о 8:00-10:00 та 13:00.

Також створимо графік з даними, які відображають кількість натискань на кнопки за листопад. Графік кількості натискань на кнопки зображено на рисунку 2.6.

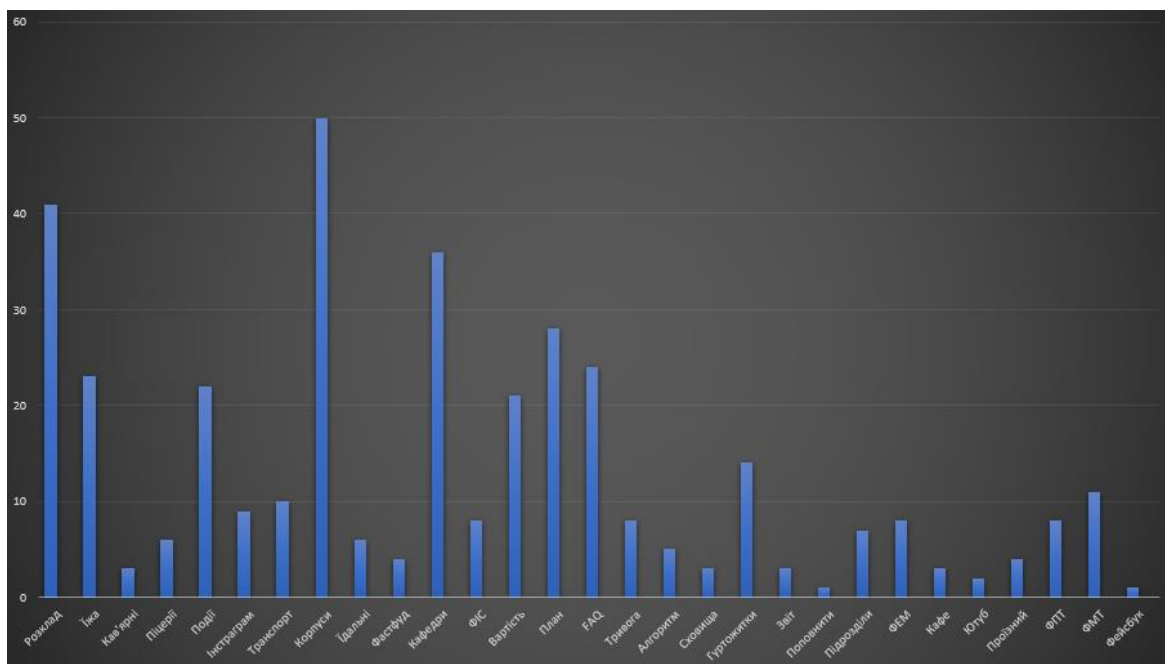


Рисунок 2.6 – Кількість натискань на кнопки

З рисунку 2.6 можемо зробити висновок, що користувачі найчастіше користуються кнопками «Розміщення корпусів» та «Розклад».

2.5 Висновок до другого розділу

В другому розділі кваліфікаційної роботи було описано процес створення сценарію розробки чат-бота, наведено прикладне використання бібліотеки `pyTelegramBotAPI`, проведено порівняльний аналіз бібліотек для розробки чат-бота у месенджері Telegram та обрано одну з них для подальшого покращення чат-бота. Крім того, проведено дослідження активності користувачів Інформаційного чат-бота ТНТУ та наведено графіки активності користувачів.

Під час дослідження активності користувачів бота було визначено дні та години, коли взаємодія з чат-ботом є найбільш інтенсивною. Також було встановлено, які кнопки найчастіше натискалися користувачами та в який період часу приєднувалося найбільше користувачів. Ці висновки дозволяють зрозуміти популярність та ефективність окремих функцій чат-бота, що є цінною інформацією для подальшої оптимізації та вдосконалення взаємодії з користувачами.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ОНОВЛЕНОЇ ВЕРСІЇ ІНФОРМАЦІЙНОГО ЧАТ-БОТА ТНТУ

3.1 Вдосконалення інформаційного чат-бота ТНТУ із застосуванням бібліотеки AIOGram

Проаналізувавши швидкість відповіді інформаційного чат-бота ТНТУ та тривалість затримки між відповідями, було прийнято рішення замінити бібліотеку `pytelegrambotapi` на іншу асинхронну бібліотеку.

На основі аналізу, виконаного у пункті 2.3, було визначено, що найкращою альтернативою є бібліотека `AIOGram`, оскільки вона характеризується такими перевагами: [47]

- Асинхронність. Бібліотека розроблена з врахуванням асинхронного програмування, за допомогою чого можна реалізувати ефективну обробку великої кількості одночасних запитів;

- Обробка повідомлень. За допомогою цієї бібліотеки можна використати потужні засоби обробки різноманітних повідомлень, подій, фотографій та велику кількість інших типів даних;

- Інтерактивні клавіатури. Дана бібліотека дозволяє розробнику використовувати `inline` та `keyboard` клавіатури для забезпечення більш комфортного використання чат-бота користувачами;

- Інтеграція з API. Бібліотека `AIOGram` забезпечує інтеграцію з різноманітними сервісами: мікросервіси, бази даних, веб-сайти та інші зовнішні ресурси.

Перш за все, особливістю асинхронного програмування є те, що функції повинні бути асинхронними, а для кожного виклику методів необхідно використовувати ключове слово `await`. [48]

```
async def cmd_test1(message: types.Message):
    await message.reply("Bulka")
```

Необхідно використовувати ключове слово `await`, оскільки дані виклики повертають корутини. Корутини є одним із елементів конкурентного програмування, схожим на систему потоків та процесів.

Конкурентність, яка базується на потоках використовується у Python за допомогою модуля `threading`. Такий вид конкурентності використовується для обробки блокуючих операцій вводу/виводу. Прикладами таких операцій є зчитування і запис файлів, робота з пристроями та сокетамі.

Конкурентність, яка базується на процесах використовується у Python за допомогою модуля `multiprocessing`. Цей вид конкурентності використовується для задач, виконання яких прямопропорційно залежить від продуктивності центрального процесора системи. Яскравим прикладом таких задач є важкі обчислювальні задачі.

Корутини є альтернативою до системи з потоків і процесів. Власне корутини надає середовище виконання мови Python, а підтримкою цього процесу займається модуль `asuncio`. Вони використовуються для виконання не блокуючих операцій вводу/виводу з використанням сокетів та підпроцесів.

Корутини в Python реалізують корпоративний підхід до багатозадачності. Корутини є функціями, виконання яких можна призупиняти та відновлювати. Виконання таких функцій призупиняється за допомогою використання ключового слова `await`, а відновлення роботи функції починається після остаточного розрахунку задач.

Основною особливістю корутин є те, що вони є легкими, тобто використовують меншу кількість ресурсів обчислювальної системи та швидше виконуються.

Також, у `AIOGram` використовується концепція хендлерів аналогічно до бібліотеки `pytelegrambotapi`. Єдиною відмінністю використання хендлерів у `AIOGram` є те, що хендлерами керує диспетчер. [49]

Диспетчер реєструє функції-обробники обмежуючи кількість подій, які викликають цю функцію за допомогою фільтрів. Після надходження нової події

диспетчер обирає необхідну функцію, яка буде використовуватись для обробки, виконавши пошук по заданим фільтрам.

За допомогою диспетчера розробник може виконувати наступні дії:

- обробляти вхідні події;
- відфільтровувати вхідні події до обробки конкретним декоратором;
- змінювати подію та відповідні дані;
- розподіляти функціональність бота між різними декораторами,

модулями та пакетами.

Приклад використання диспетчера наведено у лістингу 3.1.

Лістинг 3.1 – Приклад використання диспетчера

```
dp = Dispatcher()

@dp.message()
async def message_handler(message: types.Message) -> None:
    await SendMessage(chat_id=message.from_user.id,
text=message.text)
```

Для реєстрації обробника повідомлень розробник може використати один з двох методів:

1. Використавши декоратор на функції.

```
@dp.message_handler(commands="Decorator")
```

2. Викликати метод реєстрації диспетчера після виконання функції.

```
dp.register_message_handler(cmd_test, commands="Register")
```

Як правило, при розробці чат-ботів використовують метод зі встановленням декоратора на функції. Конструктивно декоратор в Python являє собою функцію, аргументами якої є інша функція. В загальному, декоратор

використовується для надання додаткового функціоналу певній функції без зміни вмісту.

Приклад використання декоратора наведено у лістингу 3.2.

Лістинг 3.2 – Приклад використання декоратора

```
def decor_with_return(fn):
    def wrapper(args, **kwargs):
        print("Run method: " + str(fn.name))
        return fn(args, **kwargs)
    return wrapper
```

Крім того, бібліотека AIOGram дозволяє розробнику зручно обробляти помилки, які пов'язані не з програмним кодом, а є наслідком зовнішніх подій. Аналогічно до конструкції try...except, бібліотека AIOGram містить спеціальний хендлер для обробки помилок.

Бібліотека містить наступні методи, які використовуються для обробки помилок:

- BotBlocked: обробка події, коли користувач заблокував чат-бота;
- BotKicked: ситуація, коли чат-бота було видалено з бесіди (для чат-ботів, які були створені для користувацьких груп);
- TypeOfFileMismatch: використовується для випадків, коли користувач надсилає файл не цільового формату;
- CantRestrictSelf: ситуація, коли чат-бот намагається видалити самого себе з користувацької групи через спеціальну команду від користувача (для чат-ботів, які були створені для користувацьких груп);
- CantParseUrl: використовується для випадків, коли користувач надсилає не дійсне посилання на веб-ресурс;
- FileisTooVig: обробка події, коли користувач надсилає файл, обсяг якого не передбачений функцією чат-бота.

У лістингу 3.3 наведено приклад використання хендлера `BotBlocked`, який відпрацює при умові, коли користувач заблокує чат-бота у момент надсилання повідомлення.

Лістинг 3.3 – Використання хендлера `BotBlocked`

```
from aiogram.utils.exceptions import BotBlocked

@dp.errors_handler(exception=BotBlocked)
async def error_bot_blocked(update: types.Update, exception:
BotBlocked):
    print(f"О ні!\nМене заблокували :( :{update}\nCritical
error!!!: {exception}")

    return True
```

Також `AIOGram` розширює розробникам можливості використання стандартного API Telegram. Наприклад, API месенджеру Telegram має метод `bot.send_message()`. У свою чергу, бібліотека `AIOGram` доповнює цей метод власними методами `message.answer()` та `message.reply()`.

Зручність використання цих методів полягає у тому, що вони мають меншу кількість обов'язкових полів, а також дозволяють по-різному відповідати на повідомлення користувача. Метод `message.answer()` лише відправляє повідомлення у чат звичайним способом, а метод `message.reply()` відповідає на конкретне повідомлення користувача пересилаючи його.

Приклад використання методів `message.answer()` та `message.reply()` зображено на рисунку 3.1.

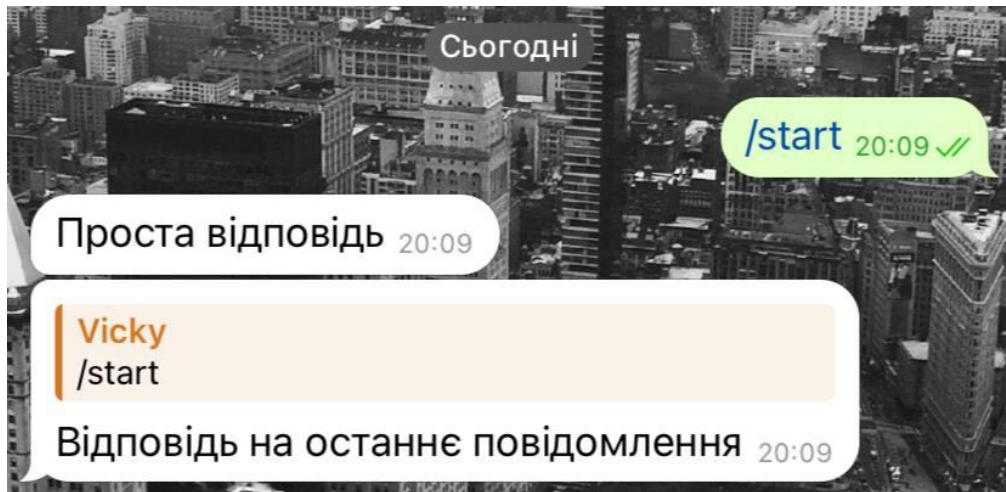


Рисунок 3.1 – Використання методів відповіді користувачу

Найбільш важливою дією чат-ботів є обробка надісланих користувачу повідомлень. Для стилістичного оформлення повідомлень бібліотека AIOGram може використати 3 різних способи розмітки тексту для надання бажаного стилю: [50]

– HTML: є найбільш поширеним способом розмітки тексту серед веб-застосунків (використовує розмітку `bold`, `<i>italic</i>`, `<u>underline</u>`, `<s>strikethrough</s>`, `spoiler`, `inline URL`)

– Markdown: є застарілою «оновленою» версією розмітки HTML, яка наразі майже не використовується оскільки її призначення – зворотня сумісність (використовує розмітку `*bold text*`, `_italic text_`, `[inline URL] (http://www.example.com/)`);

– Markdown V2: лідер у використанні при розробці чат-ботів – покращена версія Markdown (використовує розмітку `*bold *text*`, `_italic *text_`, `__underline__`, `~strikethrough~`, `||spoiler||`)

Для визначення необхідного стилю розмітки тексту використовується параметр `parse_mode`. Є декілька варіантів використання цього параметру: розробник може призначити один стиль розмітки повідомлень на всі

повідомлення чат-бота, або визначати `parse_mode` для кожного окремого методу.

Наведемо приклад встановлення параметру `parse_mode` на всі повідомлення чат-бота.

```
bot = Bot(token=config2.TOKEN, parse_mode=types.ParseMode.HTML)
```

Крім того, існує більш динамічний метод розмітки повідомлень. Для цього використовується модуль `aiogram.utils`, який надає розробнику можливість підтримувати всі варіанти способів розмітки тексту.

Таким чином, розробник може використати метод `hbold()`, де префікс `h` означає, що цільове повідомлення необхідно використати з HTML стилем розмітки.

Для розробки чат-бота було використано значну кількість різноманітних бібліотек. Для їхнього використання необхідно було імпортувати відповідні модулі на початку програмного коду. Імпортування необхідних модулів наведео у лістингу 3.4.

Лістинг 3.4 – Імпортування необхідних модулів

```
import config
import logging
import os.path
from aiogram import Bot, Dispatcher, types
from aiogram.utils import executor
from datetime import datetime
```

Кожний з імпортованих модулів має своє призначення:

- Файл `config` містить приватний ключ бота, за допомогою якого відбувається безпосереднє керування чат-ботом. Ключ спеціально виноситься в окремий файл, оскільки ключ необхідно ретельно зберігати та унеможливити

доступ третіх осіб до нього, адже використовуючи його будь-яка особа має можливість змінити функціонал, назву, наповнення а також ініціювати повне видалення чат-бота.

– За допомогою модуля `logging` визначаються об'єкти, які будуть використовуватись для реєстрації бажаних та небажаних подій під час виконання програмного коду. Крім того, цей модуль має можливість відображати події різних рівнів: `Debug`, `Info`, `Warning`, `Error`, `Critical`. В цілому, логування подій є одним з найбільш важливих елементів під час розробки та експлуатації будь-якого програмного забезпечення.

– `os.path` є модулем, який дає можливість широкої взаємодії з файлами та використання імені шляху до файлів з системи адміністратора. Найбільш поширеною є функція `open(path, params)`, яка дозволяє знайти файл у файловій системі адміністратора та використати його відповідно до вказаних параметрів: 'r' (зчитування), 'w' (запис), 'x' (створення), 'a' (дописування), 'b' (бінарний), 't' (текстовий), '+' (оновлення). [51]

– З бібліотеки `aiogram` було імпортовано декілька модулів: `Bot`, який використовується для створення екземпляру нового чат-бота; `Dispatcher`, який дозволяє розробнику реєструвати функції-обробники обмежуючи кількість подій; `types`, який дозволяє взаємодіяти з будь-яким необхідним об'єктом, наприклад, `Chat`, `User`, `Message`, `KeyboardButton`, `File`, `Update`, `WebhookInfo`, `CallbackQuery` тощо.

– Модуль `executor` з `aiogram.utils` використовується для запуску, створення нескінченного циклу роботи чат-бота, а також для встановлення параметрів роботи чат-бота.

– Бібліотека `datetime` постачає розробнику класи, які використовуються для керування датами та часом. Найбільш поширеним є використання функції `datetime.now()`, яка дозволяє записати поточний системий час адміністратора. Крім того, доволі поширеною є конвертація годин різних часових поясів за допомогою бібліотеки `datetime`.

Перш за все, у момент коли користувач вперше зверається до чат-бота надсилається команда /start. Таким чином, чат-бот обробляє цю команду та надсилає користувачу відповідне привітальне повідомлення. Програмний код привітального повідомлення наведено у лістингу 3.5.

Лістинг 3.5 – Програмний код привітального повідомлення, яке надсилається у відповідь на команду /start

```
@dp.message_handler(commands=['start'])
async def welcome(message: types.Message):
    await message.answer("Привіт, {0.first_name}
".format(message.from_user)+hi+"! Я - інформаційний чат-бот ТНТУ
"+demon+". З радістю допоможу розібратися у навчальному процесі!"
)

    markup=types.InlineKeyboardMarkup(row_width=1)
    shedule=types.InlineKeyboardButton(text="Розклад "+paper,
callback_data="sh")
    placement=types.InlineKeyboardButton(text="Розміщення
корпусів "+building, callback_data="pp")
    events=types.InlineKeyboardButton(text="Події "+event,
callback_data="ev")
    curriculum=types.InlineKeyboardButton(text="Графік
навчального процесу "+calendar, callback_data="cu")
    air=types.InlineKeyboardButton(text="Що робити під час
тривоги? "+red, callback_data="ai")
    questions=types.InlineKeyboardButton(text="Часті запитання
"+shock, callback_data="qu")
    markup.add(placement, кафедра, shedule, curriculum, price,
events, dormitory, transport, food, air, questions)
    await message.answer("Виберіть те, що вас цікавить "+look,
reply_markup=markup)
```

На початку лістингу вказано декоратор диспетчера, у якому реалізовано обробку повідомлення-команди початку спілкування користувача з чат-ботом.

У випадку, якщо чат-бот отримає та обробить команду start, користувачу буде надіслано повідомлення, яке знаходиться у асинхронній функції welcome. У свою чергу, функція welcome використовує об'єкт класу types.Message, що дозволяє надсилати повідомлення.

У функції welcome описано надсилання повідомлення message.answer, у якому міститься привітальний текст з логіном користувача.

Для використання inline кнопок використовується функція InlineKeyboardMarkup, у якій спершу збираються необхідні кнопки, а потім прикріплюються до конкретного повідомлення.

Кожна окрема кнопка має параметр callback_data, за допомогою якої виконується перенаправлення користувачеві відповідного повідомлення.

Наприклад, при натисканні кнопки «Як добратися?» чат-бот отримує callback_data="tr". Програмний код обробки callback_data="tr" наведено у лістингу 3.6.

Лістинг 3.6 – Програмний код обробки параметру callback_data="tr"

```
@dp.callback_query_handler(text="tr")
async def callback_query(call: types.CallbackQuery):
    backmarkup=types.InlineKeyboardMarkup(row_width=1)
    back=types.InlineKeyboardButton(text="Головне меню "+bback, callback_data="back")
    backmarkup.add(back)
    img=open("table2.png", 'rb')
    await call.message.answer("Найближчі до корпусів зупинки, маршрути автобусів "+abus+" (Авт) і тролейбусів "+trolleybus+" (Тр).")
    await call.message.answer_photo(img)
    await call.message.answer("Більш детально: http://detransport.com.ua/", reply_markup=backmarkup)
    await call.answer()
    button_stats(call)
```


На початку лістингу виконується обробка Callback повідомлення за допомогою хендлеру `callback_query_handler`, який очікує на параметр `text="tr"`), після чого користувачеві надсилається відповідне повідомлення.

Крім того, функція `callback_query` тепер приймає об'єкт класу `types.CallbackQuery`, оскільки у відповідь на натискання будь-якої кнопки користувачем надсилається Callback повідомлення до чат-бота.

Також у відповідь на параметр `callback_data="tr"` користувачеві надсилається фотографія за допомогою методу `call.message.answer_photo`.

За допомогою бібліотеки `os.path` було використано метод `open()`, який зчитує необхідну фотографію з сервера та передає її у метод `call.message.answer_photo`.

У лістингу 3.7 наведено програмний код обробки `callback_data="ter"`

Лістинг 3.7 – Програмний код обробки `callback_data="ter"`

```
@dp.callback_query_handler(text="ter")
async def callback_query(call: types.CallbackQuery):
    backmarkup=types.InlineKeyboardMarkup(row_width=1)
    back=types.InlineKeyboardButton(text="Головне          меню
"+back, callback_data="back")
    backmarkup.add(back)
    await          call.message.answer("Як          отримати          карту
тернополянина? "+hands+"\nІнформацію про перелік документів та
пункти видачі можна отримати тут: https://fayna-karta.te.ua/ ",
reply_markup=backmarkup)
    await call.answer()
    button_stats(call)
```

Для запуску та виконання нескінченного циклу чат-бота використовується метод `start_polling` з модуля `executor` бібліотеки `aiogram`.

```
if __name__ == "__main__":
    executor.start_polling(dp, skip_updates=True)
```

Після виконання програмного коду користувач може відразу почати користуватись чат-ботом. Перевіримо, чи справно працює чат-бот на практичному прикладі.

На рисунку 3.1 зображено привітальне повідомлення Інформаційного чат-бота ТНТУ.

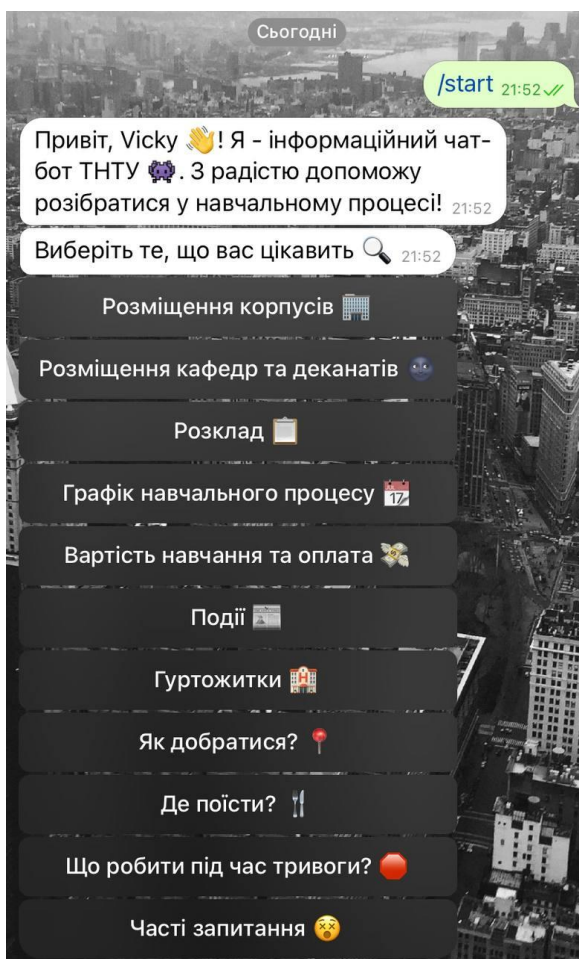


Рисунок 3.1 – Привітальне повідомлення чат-бота

Крім того, перевіримо функціонування inline кнопок привітального повідомлення чат-бота.

На рисунку 3.2 зображено відповідь чат-бота на натискання кнопки «Як добратись?».



Рисунок 3.2 – Повідомлення на кнопці “Як добратись?”

Таким чином, ми впевнились, що чат-бот повністю функціонує та виконує поставлені до нього завдання.

3.2 Реалізація запису аналітичних даних активності користувачів

Існують різні варіанти збору даних користувачів. Розглянемо найпоширеніші: [52]

1. Опитування. Це так звані анкети, які бувають як у фізичному так і у цифровому варіанті. Наприклад, ви можете проводити збір відгуків користувачів після заходу, щоб дізнатися чим вони задоволені та що потрібно змінити. Цифрові опитування мають змогу збирати дані у більших масштабах, а

також вони можуть взагалі бути безкоштовними, залежно від інструменту який ви оберете.

2. Інтерв'ю. Такий метод збору даних проводиться один на один або у групах, можна використовувати обидва варіанти. За допомогою інтерв'ю можна збирати відгуки людей з конкретної цільової аудиторії про нові введення. Також можна побачити, як люди взаємодіють з продуктом у режимі реального часу і записувати їхні реакції та відповіді на запитання. Завдяки цьому можна отримати дані про те, що у продукті потрібно покращити. Звичайно вони можуть забрати багато часу та грошей, а для того щоб не займатися цим самостійно, можна звернутися до спеціальних людей які проводять інтерв'ю від вашого імені.

3. Спостереження. Завдяки відвертості цей метод є досить корисним, хоча і доволі складним. Для спостереження можна використовувати сторонні інструменти для запису переходів користувачів на сайті або спостерігати за взаємодією користувачів з бета-версією сайту чи продукту. Цей метод є менш доступним аніж інші, та він дає можливість одразу побачити, як користувачі взаємодіють з продуктом або сайтом.

4. Трекінг дій. На веб-сайтах існує два основні інструменти для збору даних – пікселі та файли cookies. Вони надають змогу відстежувати користувачів на різних сайтах і збирають інформацію про зацікавлення користувача та його взаємодії. Вставка пікселів часто безкоштовна і відносно проста в налаштуванні. Впровадження файлів cookie може бути платним, оскільки якість даних, які ви отримаєте, буде найвищою. Після встановлення пікселів і файлів cookie вони збирають дані самостійно і не потребують особливого обслуговування.

5. Форми. Заповнення форм використовується для якісного збору даних про користувачів, наприклад збираються демографічні дані або контактна інформація. Форми є недорогими та простими у налаштуванні, а також їх можна використовувати для управління контентом або реєстраціями, таких як вебінари та електронні розсилки.

6. Моніторинг соціальних мереж. Такий метод є досить доступним способом відстежувати дані про інтереси вашої аудиторії. Різноманітні соціальні мережі часто мають вже вбудовану аналітику, але існують також сторонні соціальні платформи, які надають більш детальну інформацію отриману з різних каналів. Цей варіант надає інформацію, наприклад про те які дописи найбільше цікавлять вашу аудиторію.

Для збору аналітичних даних активності користувачів Інформаційного чат-боту ТНТУ було обрано метод трекінгу дій користувача, але використовуючи не пікселі чи файли cookies, а вбудувавши програмний код, який буде збирати інформацію про дії користувачів у режимі реального часу.

У програмному коді Інформаційного чат-боту ТНТУ створимо дві функції, які будуть збирати дані щодо активності користувачів у двох контрольних точках: натискання команди /start та взаємодія з кнопками.

При першому зверненні до чат-бота користувач повинен спершу надіслати команду /start натиснувши на кнопку «Почати» у діалозі з чат-ботом. Ця дія є обов'язковою, тому гарним рішенням є прив'язати до цього місця збір аналітичних даних.

Програмний код збору аналітичних даних появи команди /start наведено у лістингу 3.8.

Лістинг 3.8 – Збір даних появи команди /start

```
file = open("tntu_start_stats.txt", "a")
    now = datetime.now()
    file.write(message.from_user.username + " " +
now.strftime("%d/%m/%Y, %H:%M") + "\n")
    file.close()
```

Таким чином, при натисканні на кнопку «Почати» до чат-бота надсилається команда start та виконується програмний код з лістингу 3.8. Цей програмний код відкриває файл tntu_start_stats, записує до нього нову стрічку у

форматі [«Ім'я користувача» «Дата та година звернення»] та закриває файл `tntu_start_stats`.

Аналітичні дані записуються у файл формату `.txt`, що надає можливість перегляду цього файлу з практично будь-якої системи, а також можливість конвертувати дані в потрібний формат, наприклад, `.csv`.

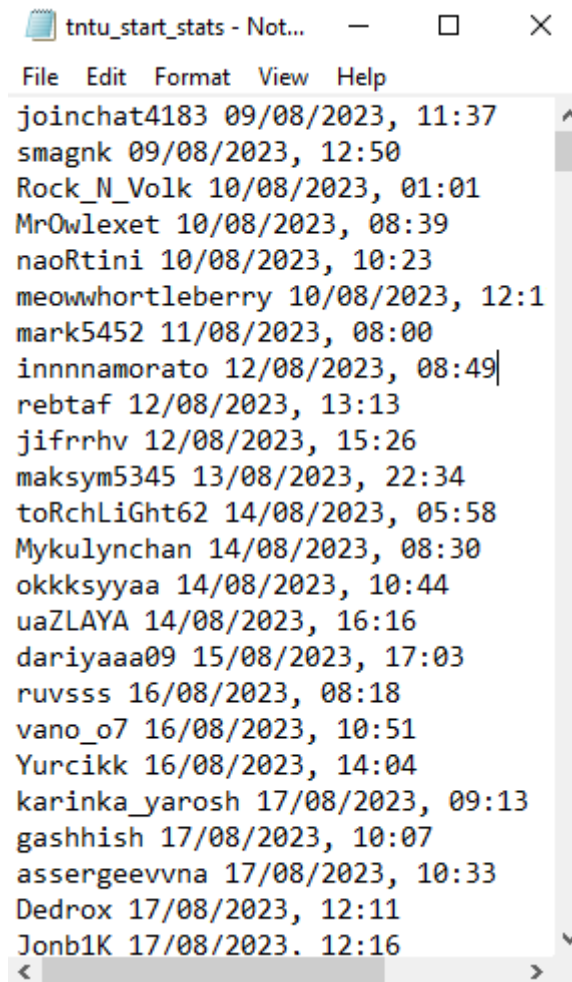
Також був доданий програмний код для запису подій взаємодії користувачів з кнопками. Програмний код запису подій взаємодії користувачів з кнопками наведено у лістингу 3.9.

Лістинг 3.9 – Програмний код запису подій взаємодії користувачів з кнопками

```
def button_stats(call):
    file = open("tntu_buttons_stats.txt", "a")
    now = datetime.now()
    file.write(str(call.message.chat.username) + " " +
now.strftime("%d/%m/%Y, %H:%M") + ", " + str(call.data) + "\n")
    file.close()
```

У лістингу 3.9 наведено програмний код, який при натисканні кнопки користувачем відкриває файл `tntu_buttons_stats`, записує до нього нову стрічку у форматі [«Ім'я користувача» «Дата та година звернення» «Натиснута кнопка»]

На рисунку 3.3 зображено вміст текстового файлу `tntu_start_stats.txt`.



```

tntu_start_stats - Not...
File Edit Format View Help
joinchat4183 09/08/2023, 11:37
smagnk 09/08/2023, 12:50
Rock_N_Volk 10/08/2023, 01:01
MrOwlexet 10/08/2023, 08:39
naoRtini 10/08/2023, 10:23
meowhortleberry 10/08/2023, 12:1
mark5452 11/08/2023, 08:00
innnamorato 12/08/2023, 08:49|
rebtaf 12/08/2023, 13:13
jifrrhv 12/08/2023, 15:26
maksym5345 13/08/2023, 22:34
toRchLiGht62 14/08/2023, 05:58
Mykulynchan 14/08/2023, 08:30
okkksyyaa 14/08/2023, 10:44
uaZLAYA 14/08/2023, 16:16
dariyaaa09 15/08/2023, 17:03
ruvsss 16/08/2023, 08:18
vano_o7 16/08/2023, 10:51
Yurcikk 16/08/2023, 14:04
karinka_yarosh 17/08/2023, 09:13
gashhish 17/08/2023, 10:07
assergeevvna 17/08/2023, 10:33
Dedrox 17/08/2023, 12:11
Jonb1K 17/08/2023. 12:16

```

Рисунок 3.3 – Вміст текстового файлу tntu_start_stats.txt

Протестувавши збір аналітичних даних було зроблено висновок, що функції збору даних працюють належним чином та відображають актуальні дані щодо активності користувачів.

3.3 Доповнення функціоналу Інформаційного чат-бота ТНТУ

Зважаючи на ситуацію в країні, важливим питанням стало інформування студентів та викладацький склад щодо дій під час повітряної тривоги. Для виконання цих цілей було запропоновано додати новий розділ до Інформаційного чат-бота ТНТУ, що дозволить забезпечити більш широкий доступ до життєво необхідної інформації.

Таким чином, було створено інструкцію для поводження під час тривоги, а саме детально описано алгоритм дій та розміщення сховищ поблизу корпусів університету.

Для створення нового блоку інформації щодо оповіщення студентів та викладацького складу університету було розроблено структуру діалогу, яка зображена на рисунку 3.4.

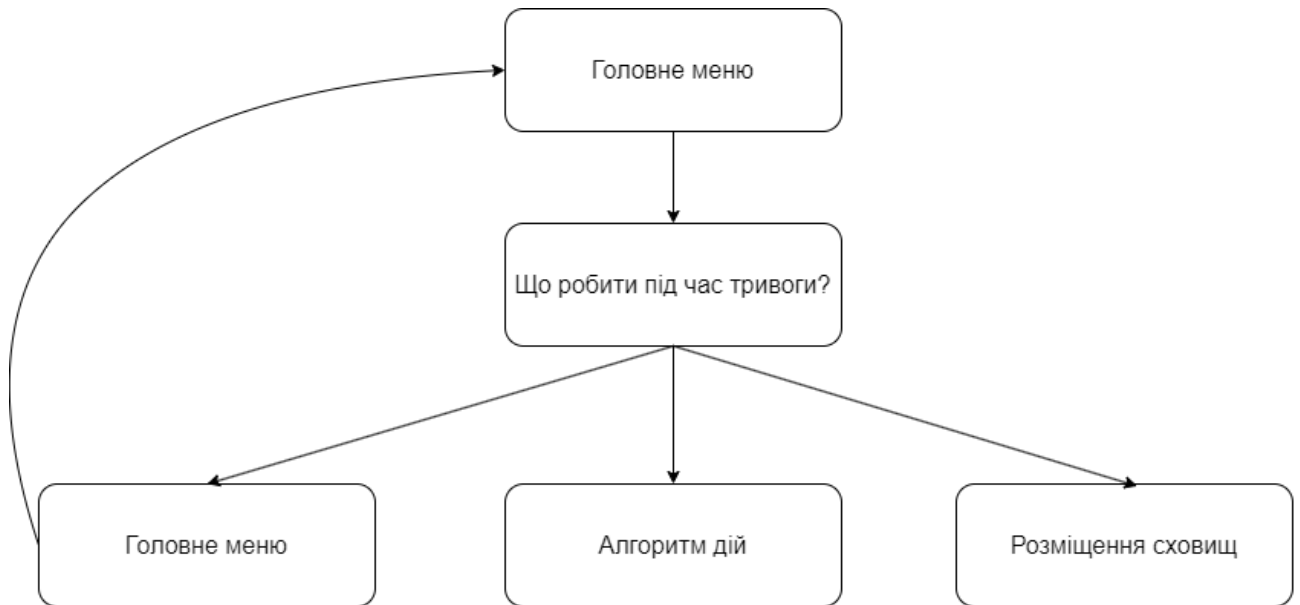


Рисунок 3.4 – Структура діалогу «Що робити під час тривоги?»

Для додавання нового блоку інформації до чат-бота необхідно було розробити програмний код з відповідною інформацією та посиланнями на кнопки.

У лістингу 3.10 наведено програмний код обробки натискання на кнопку «Що робити під час тривоги?».

Лістинг 3.10 – Обробка кнопки «Що робити під час тривоги?»

```

@dp.callback_query_handler(text="ai")
async def callback_query(call: types.CallbackQuery):
    backmarkup=types.InlineKeyboardMarkup(row_width=1)
    bunk=types.InlineKeyboardButton(text="Розміщення сховищ
"+bun, callback_data="bunker")
  
```



```

    algo=types.InlineKeyboardButton(text="Алгоритм дій "+alg,
callback_data="algoritm")
    back=types.InlineKeyboardButton(text="Головне меню "+bback,
callback_data="back")
    backmarkup.add(bunk, algo, back)
    await call.message.answer("Виберіть те, що вас цікавить "+look,
reply_markup=backmarkup)
    await call.answer()
    button_stats(call)

```

Таким чином, внаслідок імплементатії наведеного програмного коду до чат-бота користувачі зможуть побачити новий блок інформації з головного меню. На рисунку 3.5 зображено результат додавання нового блоку інформації до Інформаційного чат-бота ТНТУ.



Рисунок 3.5 – Діалог «Що робити під час тривоги?»

На рисунку 3.5 можна побачити реакцію та повідомлення чат-бота на вибір кнопки «Алгоритм дій» у розділі «Що робити під час тривоги?».

3.4 Висновок до третього розділу

В третьому розділі кваліфікаційної роботи було проведено вдосконалення Інформаційного чат-бота ТНТУ із застосуванням бібліотеки AIOGram, наведено реалізацію запису аналітичних даних активності користувачів та здійснено доповнення функціоналу Інформаційного чат-бота ТНТУ.

Вдосконалення чат-бота за допомогою бібліотеки AIOGram дозволило зробити бот асинхронним, підвищуючи його ефективність та швидкодію відгуків. Крім того, AIOGram надає ряд додаткових можливостей, які сприяють поліпшенню функціоналу та зручності використання.

Додатково, впроваджено нову функцію у вигляді кнопки для ситуацій тривоги, що стає особливо актуальним у контексті воєнного стану в Україні. Ця функція дозволяє користувачам швидко отримати необхідну інформацію та інструкції, підвищуючи зручність взаємодії з чат-ботом у критичних ситуаціях.

РОЗДІЛ 4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Охорона праці

4.1.1 Значення адаптації в трудовому процесі

Працівник нормально здійснює трудову діяльність лише тоді, коли умови зовнішнього середовища відповідають оптимальним. Якщо вони стають несприятливими та на протидію їм організм людини включає спеціальний механізм, який зберігає постійність внутрішнього середовища, або змінює його в межах допустимого. Такий механізм називається адаптацією.[53]

Адаптація в трудовій діяльності поділяється на фізіологічну, психічну, соціальну та професійну:

– фізіологічна адаптація – це сукупність фізіологічних реакцій, які є в основі пристосування організму до змін оточуючого середовища і направлені на збереження відносної постійності його внутрішнього середовища. Суть механізму адаптації полягає у змінах меж чутливості аналізаторів, розширенні діапазону фізіологічних резервів організму та зміні в певних межах параметрів фізіологічних функцій (підвищується стійкість організму до холоду, тепла, недостачі кисню, змін барометричного тиску);

– психічна адаптація – це процес встановлення оптимальної відповідності особистості до оточуючого середовища в процесі діяльності. Психічна адаптація в процесі праці залежить від психічних властивостей працівника, його психічного стану, психологічних реакцій на стреси, що виникають на роботі, кваліфікації та культури людини, особливостей професійної діяльності, конкурентних умов праці ;

– соціальна адаптація – це пристосування працюючої людини до системи відносин у робочому колективі з його нормами, правилами, традиціями, ціннісними орієнтаціями. При несприятливому протіканні соціальної адаптації підвищується рівень стресу на роботі, наслідки якого позначаються на

поведінці працівника та можуть призвести до міжособових конфліктів, нещасних випадків;

– професійна адаптація – це адаптація до трудової діяльності з усіма її складовими і адаптація до робочого місця, знарядь та засобів праці, об'єктів та предметів праці, особливостей технологічного процесу, головних параметрів роботи. Професійна адаптація виражається у розвитку стійкого позитивного ставлення працівника до своєї професії, певного рівня оволодіння ним специфічними навичками та вміннями у формуванні необхідних для якісного виконання роботи властивостей. Виявляється в ознайомленні з професійною роботою, придбанні навичок професійної майстерності, вправності, достатніх для якісного виконання функціональних обов'язків і творчості в праці.

Для розробки оптимальних програм адаптації важливо враховувати фактори, які стосуються специфіки організації з одного боку і категорії нових співробітників, на яких програма орієнтована – з іншого. До специфіки організації відносяться особливості чи сфера її діяльності. А співробітники умовно розділені на молодих фахівців, що мають в основному тільки теоретичні знання про робочий процес, і професіоналів з практичними знаннями і суттєвим досвідом роботи в інших організаціях.[54]

Показником успішно проведеного процесу адаптації персоналу і введення його на посаду є успішне виконання роботи. Наскільки ефективними виявилися відбір кандидатів і проходження ними програми адаптації, можна визначити з низки простих критеріїв.

Критерії адаптації для працівників:

- виконання норм виробітку;
- виконання змінно-добових завдань;
- опанування робочим місцем відповідно до технічних умов;
- дотримання трудової дисципліни;
- здатність влитися в колектив;
- дотримання ділової філософії підприємства.

Кожен із розглянутих видів адаптації впливає на працездатність та здоров'я працівника, формує у нього певний рівень чутливості та стійкості до психоемоційних перевантажень, внаслідок розвитку яких може істотно змінитися надійність професійної діяльності.

4.1.2 Психофізіологічне розвантаження для працівників

Праця офісних працівників, у тому числі розробника чат-боту характеризується тривалою багатогодинною працею в одноманітному напруженому положенні, малою руховою активністю при значних динамічних навантаженнях кістково-м'язового апарату кистей рук.[55]

Тривала понаднормова робота за комп'ютеризованим робочим місцем призводить до значного навантаження зорового аналізатора та опорно-рухового апарату, що зумовлює високий рівень нервово-психічного перевантаження, знижує функціональну активність центральної нервової системи, призводить до розвитку втоми, стресу, а також до емоційного вигорання.

Виділяються три періоди, що відповідають фазам відновлювального процесу[56]:

– перший період – абстрагування працівників від виробничої обстановки, який відповідає фазі залишкового збудження. Лунають повільна мелодійна музика, пташиний спів. Обравши зручну позу, працівники адаптуються і психологічно готуються до наступних періодів;

– другий – заспокоєння. Відповідає фазі відновлювального гальмування. Пропонується показ фотослайдів із зображеннями квітучого луку, березового гаю, гладенької поверхні ставка. Через навушники транслюється спокійна музика, а на її фоні негучно, повільно висловлюються заспокійливі формули аутогенного тренування. Як функціональне освітлення застосовують зелене світло. Яскравість світла має поступово знижуватись протягом періоду, а наприкінці його світло вимикається зовсім на одну - дві хвилини. Екран теж гасне;

– третій період – активізація, який відповідає фазі підвищеної збудженості. На початку періоду світло вимкнене, через певний час на екрані з'являється червона пряма, розміри і яскравість якої поступово збільшуються. Наприкінці періоду лунає бадьора музика. Вимовляються тричі мобілізуючі формули аутогенного тренування, яким мають передувати глибоке вдихання та довге глибоке видихання.

Після сеансів психофізіологічного розвантаження у працівників зменшується відчуття втоми, з'являються бадьорість, гарний настрій та покращується їх загальний стан.

4.1.3 Соціальне значення охорони праці

Соціальне значення охорони праці полягає в сприянні росту ефективності суспільного виробництва шляхом безперервного вдосконалення і поліпшення умов праці, підвищення їх безпеки, зниження виробничого травматизму і професійних захворювань.

Зростання продуктивності праці відбувається в результаті збільшення фонду робочого часу завдяки скороченню внутрішньо-змінних простоїв шляхом ліквідації мікротравм або зниження їх кількості, а також завдяки запобіганню передчасного стомлення шляхом раціоналізації і покращення умов праці та введенню оптимальних режимів праці і відпочинку та інших заходів, які сприяють підвищенню ефективності використання робочого часу.

Збереження трудових ресурсів і підвищення професійної активності працюючих відбувається завдяки покращенню стану здоров'я і подовженню середньої тривалості життя шляхом покращення умов праці, що супроводжується високою трудовою активністю і підвищенням виробничого стажу. Підвищується професійний рівень також завдяки зростанню кваліфікації і майстерності.[57]

Соціальне значення охорони праці проявляється, насамперед, у впливі на зміну наступних основних показників, які характеризують рівень розвитку суспільного виробництва:

– зростання продуктивності праці в результаті збільшення фонду робочого часу за рахунок скорочення цілодобових втрат робочого часу в результаті зниження рівня або ліквідації тимчасової непрацездатності через виробничий травматизм, професійну й загальну захворюваність. Цей показник має важливе значення для виробництв, де кожна травма в даний час, за даними статистики, супроводжується втратою працездатності в середньому більш ніж на 26 днів;

– збереження трудових ресурсів і підвищення професійної активності працюючих за рахунок можливості використання залишкової трудової активності, великого практичного досвіду й професійних знань пенсіонерів по старості й інвалідів на доступних для них роботах і забезпеченні відповідних їх фізичним можливостям умов праці.

Таким чином, визначено, що соціальне значення охорони праці є досить важливим оскільки завдяки цьому зберігається висока ефективність підприємств в цілому.

4.2 Безпека в надзвичайних ситуаціях

4.2.1 Структура цивільного захисту ТНТУ імені Івана Пулюя. Її аналіз

Цивільний захист – це система заходів та дій, спрямованих на захист життя, здоров'я, майна та навколишнього середовища людей в умовах надзвичайних ситуацій, таких як природні катастрофи, техногенні аварії, воєнні конфлікти, епідемії, терористичні атаки та інші загрози безпеці. [58]

Таким чином, цивільний захист є одним з найбільш важливих структурних підрозділів Тернопільського національного технічного університету імені Івана Пулюя. Розвиток структури цивільного захисту та

планування заходів є невід'ємною частиною загальної стратегії університету щодо управління надзвичайними ситуаціями та запобігання можливим небезпекам.

Цивільний захист Тернопільського національного технічного університету регулюється наступними документами:

- Інструкція з техногенної безпеки, ЦЗ та дій у надзвичайних ситуаціях; [59]
- Інструкція №321 керівника структурного підрозділу; [60]
- Інструкція №322 Щодо дії персоналу при надзвичайних ситуаціях; [61]
- Пам'ятка «Робота в воєнний час»; [62]
- Правила поведінки при виявленні небезпечних предметів. [63]

У перелічених регулюючих документах чітко вказано порядок дій в надзвичайних ситуаціях під час війни, у разі виникнення ситуацій природного та техногенного характеру, дії в разі аварії на системах життєзабезпечення, порядок дій керівника підрозділу цивільного захисту під час різних режимів а також регулювальні сигнали, які використовуються для оповіщення.

Для більш детального розуміння роботи відділу цивільного захисту необхідно проаналізувати його структуру. Структура цивільного захисту Тернопільського національного технічного університету включає такі основні складові:

- Керівництво цивільного захисту. Включає в себе керівника цивільного захисту та його заступників. Керівник цивільного захисту координує всі аспекти цивільного захисту університету, приймає стратегічні рішення та відповідає за розробку та впровадження планів евакуації та захисту. У свою чергу, заступники керівника допомагають у керівництві та можуть взяти на себе обов'язки в разі відсутності керівника;
- Цивільний захист на рівні факультетів. Відповідальними за цю складову є координатори та групи цивільного захисту. Вони організують та контролюють впровадження планів цивільного захисту на своєму факультеті, а

також забезпечують підтримку та організацію заходів цивільного захисту у своїх сферах відповідальності;

– Планування цивільного захисту. Включає штаб та комітет цивільного захисту. Основними обов'язками є розробка, актуалізація та підтримка планів цивільного захисту. Крім того, в зону їхньої відповідальності входить організація тренування та навчання для персоналу та студентів;

– Матеріальне та медичне забезпечення. Технічний персонал відповідає за інженерно-технічні аспекти цивільного захисту, включаючи стан споруд, обладнання для захисту та рятування. В свою чергу, медичний персонал забезпечує медичну допомогу, організовує роботу лікувальних пунктів та готує медичний персонал до надзвичайних ситуацій;

– Ланка інформації та зв'язку. Цей елемент структури цивільного захисту ТНТУ забезпечує роботу засобів зв'язку, інформаційну підтримку під час екстрених ситуацій, встановлює та підтримує взаємодію із місцевими службами цивільного захисту, пожежною охороною і іншими рятувальними службами.

Ця структура цивільного захисту в Тернопільському національному технічному університеті гарантує готовність та швидку реакцію на надзвичайні ситуації, сприяє безпеці всіх членів університетської спільноти та також забезпечує належну організацію та координацію дій під час надзвичайних подій. Структура цивільного захисту в університеті також забезпечує взаємодію з місцевими органами влади, службами екстреної допомоги та іншими інституціями, що дозволяє забезпечити взаємопідтримку та взаємодію в умовах надзвичайних ситуацій.

Отже, наявна структура цивільного захисту Тернопільського національного технічного університету імені Івана Пулюя демонструє високий рівень готовності, організаційної підготовки та відповідальності університетської спільноти перед можливими небезпеками та надзвичайними ситуаціями.

4.3 Висновок до четвертого розділу

В третьому розділі кваліфікаційної роботи описано питання з охорони праці та безпеки в надзвичайних ситуаціях.

Визначено, що адаптація в трудовому процесі поділяється на чотири основні частини: фізіологічну, психічну, соціальну та професійну. Значенням успішно проведеної адаптації є підвищення результативності виконання роботи працівником.

Психофізіологічне розвантаження має три періоди, що відповідають фазам відновлюваного процесу: абстрагування працівників від виробничої обстановки, заспокоєння та активізація.

Соціальне значення охорони праці проявляється у впливі на зміну основних показників, які характеризують рівень розвитку суспільного виробництва.

Виявлено, що структура цивільного захисту в Тернопільському національному технічному університеті належно організована та функціонує з метою гарантування готовності та оперативної реакції на надзвичайні ситуації. Ця структура не лише сприяє безпеці всіх членів університетської спільноти, але й забезпечує належну організацію та координацію дій під час надзвичайних подій, роблячи акцент на ефективному управлінні та підтримці університетського середовища в умовах ризику та непередбачених обставин.

ВИСНОВКИ

Під час виконання цієї кваліфікаційної роботи освітнього рівня «Магістр» було проведено аналіз Інформаційного чат-бота ТНТУ та його вдосконалення за допомогою мови програмування Python та бібліотеки AIOGram.

В першому розділі кваліфікаційної роботи:

- проаналізовано актуальність використання чат-ботів;
- розглянуто сфери використання чат-ботів;
- подано чат-ботів із штучним інтелектом;
- наведено приклади найпопулярніших чат-ботів у різних соціальних мережах.

В другому розділі кваліфікаційної роботи:

- опрацьовано сценарій розробки чат-бота;
- проаналізовано прикладне використання бібліотеки pyTelegramBotAPI;
- наведено порівняльний аналіз бібліотек для розробки чат-бота у месенджері Telegram;
- висвітлено дослідження активності користувачів Інформаційного чат-бота ТНТУ.

В третьому розділі кваліфікаційної роботи:

- проведено вдосконалення Інформаційного чат-бота ТНТУ із застосуванням бібліотеки AIOGram;
- виконано реалізацію запису аналітичних даних активності користувачів.
- здійснено доповнення функціоналу Інформаційного чат-бота ТНТУ.

У розділі «Охорона праці та безпека в надзвичайних ситуаціях» розглянуто питання значення адаптації у трудовому процесі, психофізіологічні розвантаження для працівників, соціальне значення охорони праці та структуру цивільного захисту ТНТУ імені Івана Пулюя.

Результатом виконання цієї кваліфікаційної роботи є вдосконалення та аналіз інформаційного чат-бота для ТНТУ за допомогою якого студенти

можуть оперативно знайти інформацію щодо навчання. Дослідження виявилось важливим кроком у напрямку покращення інформаційного чат-бота ТНТУ. Застосування мови програмування Python та бібліотеки Aiogram дозволили ефективно аналізувати та вдосконалювати функціонал інформаційного чат-боту, забезпечуючи його більшу адаптивність, ефективність та зручність для користувачів.

Запроваджені покращення дозволили підняти якість обслуговування та забезпечити більш високий рівень задоволення від користування інформаційним чат-ботом ТНТУ.

Результати цього дослідження свідчать про важливість постійного вдосконалення інформаційних систем у вищих навчальних закладах. Підвищення якості та ефективності таких систем є актуальним завданням, що сприяє покращенню загального досвіду користувачів та забезпечує відповідність університетського інформаційного сервісу сучасним стандартам та вимогам.

ПЕРЕЛІК ДЖЕРЕЛ

- 1 Чат-боти: вигоди та ризики застосування [Електронний ресурс] – Режим доступу до ресурсу: <https://conf.ztu.edu.ua/wp-content/uploads/2021/11/118.pdf>.
- 2 Фактори взаємодії користувачами з чат-ботом [Електронний ресурс] – Режим доступу до ресурсу: https://gerabot.com/article/vazhlivi_faktori_vzamodii_z_koristuvachami_chatbota.
- 3 Чому чат-боти стали популярними [Електронний ресурс] – Режим доступу до ресурсу: <https://valmaxdigital.com.ua/ua/blog/why-chat-bots-became-popular-and-how-to-use-them-for-your-business/>.
- 4 Переваги використання чат-ботів [Електронний ресурс] – Режим доступу до ресурсу: <https://gincore.net/uk/blog/preimuschestva-prodazh-cherez-chat-bot>.
- 5 Чат-бот: плюси на мінуси [Електронний ресурс] – Режим доступу до ресурсу: <https://ukr-bot.com/plyusi-ta-minusi-chat-botiv/>.
- 6 Чат-боти в освітньому процесі [Електронний ресурс] – Режим доступу до ресурсу: <https://mon.gov.ua/ua/osvita/cifrova-osvita/osvitnij-chat-bot>.
- 7 Використання чат-ботів у сфері послуг [Електронний ресурс] – Режим доступу до ресурсу: https://gerabot.com/article/chatbot_v_sfere_uslug_kak_effektivnij_sposob_povisheniya_effektivnosti_biznesa.
- 8 Чат-бот для бронювання квитків [Електронний ресурс] – Режим доступу до ресурсу: <https://railwaybot.com/uk/>.
- 9 Ріст популярності чат-ботів [Електронний ресурс] – Режим доступу до ресурсу: <https://forklog.com/sp/chatbots/>.
- 10 Чат-боти у банківській сфері [Електронний ресурс] – Режим доступу до ресурсу: <https://aiconference.com.ua/uk/news/chat-boti-v-bankovskoy-sfere-kak-virtualnie-sobesedniki-ispolzuyutsya-v-bankah-93750>.

11 Збір статистики за допомогою чат-ботів [Електронний ресурс] – Режим доступу до ресурсу: <https://webpromoeexperts.net/ua/blog/telegram-botidlya-internet-marketologa/>.

12 Використання чат-ботів зі штучним інтелектом [Електронний ресурс] – Режим доступу до ресурсу: <https://www.intuz.com/blog/guide-to-building-an-ai-chatbot>.

13 I. Strutynska, H. Kozbur, L. Dmytrotsa, O. Sorokivska, L. Melnyk, R. Grytseliak. Regarding to the Concept of Small and Medium-Sized Enterprises Digitalization in Ukraine: Problems and Solutions. IEEE Deggendorf, Germany, 2021, pp.276–279.

14 I. Strutynska, L. Dmytrotsa, H. Kozbur, L. Melnyk, R. Sherstiuk: The Unification of Approaches to Measuring the Digital Maturity of Business Structures (International and Domestic Approaches). ICTERI 2021: Kherson, Ukraine, September 28 - October 2, 2021. CEUR Workshop Proceedings 3013, CEUR-WS.org 2021, pp.10-23.

15 Стереотипи щодо чат-ботів [Електронний ресурс] – Режим доступу до ресурсу: <https://www.business.com/articles/chabots-practical-ai/>.

16 NLP у чат-ботах [Електронний ресурс] – Режим доступу до ресурсу: <https://www.cm.com/glossary/what-is-ai-chatbot/>.

17 Особливості використання чат-ботів зі штучним інтелектом [Електронний ресурс] – <https://www.ranktracker.com/uk/blog/how-ai-powered-chatbots-are-transforming-customer-support/>.

18 Недоліки чат-ботів, побудованих на основі штучного інтелекту [Електронний ресурс] – Режим доступу до ресурсу: <https://www.drift.com/learn/chatbot/ai-chatbots/>.

19 Покращення чат-ботів на основі клієнтського досвіду [Електронний ресурс] – Режим доступу до ресурсу: <https://www.gms-worldwide.com/uk/blog/how-can-banking-ai-chatbots-improve-cx-and-customer-retention/>.

20 Плюси використання чат-ботів [Електронний ресурс] – Режим доступу до ресурсу: <https://ideadigital.agency/blog/top-10-prichin-chomu-chat-boti-u-sotsmerezah-potribni-kozhnomu/>.

21 Етапи створення чат-бота зі штучним інтелектом [Електронний ресурс] – Режим доступу до ресурсу: <https://www.uptech.team/blog/how-to-build-an-ai-chatbot>.

22 Приклади чат-ботів у месенджерах [Електронний ресурс] – Режим доступу до ресурсу: <https://prometheus.org.ua/blog/ukrainiantelegrambots/>.

23 Створення чат-бота у Viber [Електронний ресурс] – Режим доступу до ресурсу: <https://sendpulse.ua/blog/viber-chatbots-guide>.

24 Чат-боти у месенджері Facebook [Електронний ресурс] – Режим доступу до ресурсу: <https://uaateam.agency/blog/nalashtuvannya-chat-bota-v-facebook/>.

25 Чат-бот Укрпошти у Facebook [Електронний ресурс] – Режим доступу до ресурсу: <https://www.facebook.com/ukrposhta.chatbot>.

26 Створення чат-бота у месенджері Telegram [Електронний ресурс] – Режим доступу до ресурсу: <https://helpcrunch.com/blog/uk/chat-bot-dla-telegram/>.

27 Чат-боти на платформах веб-сайтів [Електронний ресурс] – Режим доступу до ресурсу: <https://helpcrunch.com/blog/uk/yak-stvoryty-chat-bota-dlia-saitu/>.

28 Дослідження креативності чат-ботів [Електронний ресурс] – Режим доступу до ресурсу: <https://life.pravda.com.ua/society/2023/09/17/256584/>.

29 Поява чат-бота ChatGPT [Електронний ресурс] – Режим доступу до ресурсу: <https://www.epravda.com.ua/publications/2023/02/21/697273/>.

30 Етапи створення чат-бота [Електронний ресурс] – Режим доступу до ресурсу: <https://engage.sinch.com/blog/design-chatbot-company-tips/>.

31 Способи розробки чат-ботів [Електронний ресурс] – Режим доступу до ресурсу: <https://evergreens.com.ua/ru/articles/create-bot.html>.

32 Тестування чат-ботів [Електронний ресурс] – Режим доступу до ресурсу: <https://www.chatcompose.com/uk/testchatbots.html>.

33 Сценарій чат-бота [Електронний ресурс] – Режим доступу до ресурсу: <https://youtogift.com/ua/blog/scenarij-cat-bota-instrukcia-po-sozdaniu-primery>.

34 Технічні аспекти створення чат-ботів [Електронний ресурс] – Режим доступу до ресурсу: <https://aweb.ua/blog/uk/shho-take-chat-boti-i-yak-stvoriti-chat-bota-dlya-sajtu/>.

35 Створення сценарію для чат-боту [Електронний ресурс] – Режим доступу до ресурсу: <https://bot.konveier.com/poleznoye/kak-sostavit-stsenarij-dialoga-dlya-chat-bota/>.

36 Бібліотека `pyTelegramBotAPI` [Електронний ресурс] – Режим доступу до ресурсу: <https://www.freecodecamp.org/ukrainian/news/yak-stvoryty-telehram-bota-za-dopomohoyu-python/>.

37 Види кнопок у Telegram [Електронний ресурс] – Режим доступу до ресурсу: <https://mastergroosha.github.io/telegram-tutorial-2/buttons/>.

38 Документація бібліотеки `pyTelegramBotAPI` [Електронний ресурс] – Режим доступу до ресурсу: <https://pypi.org/project/pyTelegramBotAPI/>.

39 Функції чат-ботів [Електронний ресурс] – Режим доступу до ресурсу: <https://www.carrotquest.io/chatbot/chatbot-types/>.

40 Бібліотеки для створення чат-ботів [Електронний ресурс] – Режим доступу до ресурсу: <https://kandi.openweaver.com/collections/python/python-telegram-bot>.

41 Створення чат-бота за допомогою бібліотеки `python-telegram-bot` [Електронний ресурс] – Режим доступу до ресурсу: <https://spacelab.ua/articles/yak-stvoriti-telegram-bota-na-python/>.

42 Документація до бібліотеки `telepot` [Електронний ресурс] – Режим доступу до ресурсу: <https://telepot.readthedocs.io/en/latest/>.

43 Огляд бібліотеки `AIOGram` [Електронний ресурс] – Режим доступу до ресурсу: <https://drukarnia.com.ua/articles/telegram-boti-na-python-oglyad-p-yati-naikrashikh-freimvorkiv-bibliotek-L7UA7>.

44 Документація `AIOGram` [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.aiogram.dev/en/latest/>.

45 Створення інформаційного чат-боту для студентів ТНТУ засобами мови програмування Python та Telegram API [Електронний ресурс] – Режим доступу до ресурсу: https://elartu.tntu.edu.ua/bitstream/lib/38294/1/KRB_Shyshkina_V_O.pdf

46 Важливі метрики аналітики [Електронний ресурс] – Режим доступу до ресурсу: <https://www.marketing-ua.com/article/10-golovnih-metrik-dlya-analitiki-mobilnogo-dodatka/>.

47 Переваги бібліотеки AIOGram [Електронний ресурс] – Режим доступу до ресурсу: <https://medium.com/@galperovich/build-a-telegram-chatbot-with-any-ai-model-under-the-hood-62f9a8675d81>.

48 Використання асинхронного програмування [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.python.org/uk/3/library/asyncio-dev.html>.

49 Використання хендлерів у AIOGram [Електронний ресурс] – Режим доступу до ресурсу: <https://mastergroosha.github.io/aiogram-3-guide/>.

50 Стилзація тексту Telegram [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/sandbox/170069/>.

51 Біблотека os.path [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.python.org/3/library/os.path.html>.

52 Методи збору даних користувачів [Електронний ресурс] – Режим доступу до ресурсу: <https://online.hbs.edu/blog/post/data-collection-methods>.

53 Банько Н.А. Управління персоналом / Н.А. Банько, Б.А. Карташов, Н.С. Яшин. Част. I: Навч. посіб. Волгоград. Вид. ВолГТУ. 2006. 96 с.

54 Значення адаптації в трудовому процесі [Електронний ресурс] – Режим доступу до ресурсу: <https://library.if.ua/book/9/921.html>.

55 Захарчук К. Емоційне вигорання чи втома? / К. Захарчук // Науково-виробничий журнал “Охорона Праці”. – 2021. – №1. С. 48-51.

56 Психофізіологічне перевантаження на роботі як чинник виробничого травматизму [Електронний ресурс] – Режим доступу до ресурсу: <https://bit.ly/3xiV6SK>.

57 Жидецький В. Ц. Основи охорони праці / В. Ц. Жидецький, В. С. Джигирей, О. В. Мельников. – Львів: Афіша, 2000. – 350 с.

58 Що таке цивільний захист? [Електронний ресурс] – Режим доступу до ресурсу: <http://old.voz.gov.ua/material/74>.

59 Інструкція з техногенної безпеки, ЦЗ та дій у надзвичайних ситуаціях [Електронний ресурс] – Режим доступу до ресурсу: https://tntu.edu.ua/storage/pages/00000907/Instrukcia_z_tehnogennoi_bezpeky.pdf.

60 Інструкція № 321 керівника структурного підрозділу [Електронний ресурс] – Режим доступу до ресурсу: https://tntu.edu.ua/storage/pages/00000907/Instrukcia321_kerivnyka_struct_pidrozdilu.pdf.

61 Інструкція № 322 Щодо дії персоналу при надзвичайних ситуацій [Електронний ресурс] – Режим доступу до ресурсу: https://tntu.edu.ua/storage/pages/00000907/Instrukcia322_diy_personalu.pdf.

62 Пам'ятка «Робота в воєнний час» [Електронний ресурс] – Режим доступу до ресурсу: https://tntu.edu.ua/storage/pages/00000907/Pamiatka_Robota_v_voennyu_chas.pdf.

63 Правила поведінки при виявленні небезпечних предметів [Електронний ресурс] – Режим доступу до ресурсу: https://tntu.edu.ua/storage/pages/00000907/Pravyla_povedinky_pry_vyavlenni_nebezpechnyh_predmetiv.pdf.

ДОДАТКИ

Програмний код чат-боту

```
import config
import logging
import os.path
from aiogram import Bot, Dispatcher, types
from aiogram.utils import executor
from datetime import datetime

downarrow=u'\U00002B07'
look=u'\U0001F50D'
wrench=u'\U0001F527'
computer=u'\U0001F4BB'
gmark=u'\U00002705'
mark=u'\U00002611'
danger=u'\U000026A0'
electric=u'\U000026A1'
spagetti=u'\U0001F35D'
cookie=u'\U0001F36A'
coffee=u'\U00002615'
plate=u'\U0001F372'
hamburger=u'\U0001F354'
eat=u'\U0001F374'
pan=u'\U0001F373'
ball=u'\U000026BD'
swim=u'\U0001F3CA'
hostel=u'\U0001F3E8'
building=u'\U0001F3E2'
hi=u'\U0001F44B'
ok=u'\U0001F44C'
cool=u'\U0001F44D'
demon=u'\U0001F47E'
dance=u'\U0001F483'
hearts=u'\U0001F495'
light=u'\U0001F4A1'
star=u'\U0001F4AB'
sto=u'\U0001F4AF'
money=u'\U0001F4B8'
stonks=u'\U0001F4C8'
calendar=u'\U0001F4C6'
maps=u'\U0001F4CD'
paper=u'\U0001F4CB'
book=u'\U0001F4D6'
books=u'\U0001F4DA'
printer=u'\U0001F4E0'
```

```
news=u'\U0001F4E2'  
skrepka=u'\U0001F517'  
bback=u'\U0001F519'  
fire=u'\U0001F525'  
sphere=u'\U0001F52E'  
euro=u'\U0001F4B6'  
moon=u'\U0001F31A'  
trolleybus=u'\U0001F68E'  
bus=u'\U0001F68D'  
hands=u'\U0001F64C'  
shock=u'\U0001F635'  
smile=u'\U0001F60A'  
tipok=u'\U0001F60F'  
calm=u'\U0001F60C'  
tongue=u'\U0001F60B'  
laught=u'\U0001F604'  
kapla=u'\U0001F605'  
teeth=u'\U0001F601'  
abus=u'\U0001F68C'  
hotdog=u'\U0001F32D'  
event=u'\U0001F4F0'  
winking=u'\U0001F609'  
pizzas=u'\U0001F355'  
strilka=u'\U000021A9'  
dama=u'\U0001F469'  
swims=u'\U0001F3CA'  
red=u'\U0001F6D1'  
alg=u'\U0001F4CB'  
bun=u'\U0001F3E0'
```

```
bot = Bot(token=config2.TOKEN,  
parse_mode=types.ParseMode.HTML)  
dp = Dispatcher(bot)
```

```
logging.basicConfig(level=logging.INFO)
```

```
def button_stats(call):  
    file = open("tntu_buttons_stats.txt", "a")  
    now = datetime.now()  
    file.write(str(call.message.chat.username) + " " +  
now.strftime("%d/%m/%Y, %H:%M") + ", " + str(call.data) + "\n")  
    file.close()
```

```
@dp.message_handler(commands=['start'])  
async def welcome(message: types.Message):
```

```

        await message.answer("Привіт, {0.first_name}
".format(message.from_user)+hi+"! Я - інформаційний чат-бот
ТНТУ "+demon+". З радістю допоможу розібратися у навчальному
процесі!" )
        markup=types.InlineKeyboardMarkup(row_width=1)
        shedule=types.InlineKeyboardButton(text="Розклад "+paper,
callback_data="sh")
        placement=types.InlineKeyboardButton(text="Розміщення
корпусів "+building, callback_data="pp")
        events=types.InlineKeyboardButton(text="Події "+event,
callback_data="ev")
        curriculum=types.InlineKeyboardButton(text="Графік
навчального процесу "+calendar, callback_data="cu")
        price=types.InlineKeyboardButton(text="Вартість навчання
та оплата "+money, callback_data="pr")
        кафедра=types.InlineKeyboardButton(text="Розміщення кафедр
та деканатів "+moon, callback_data="ca")
        dormitory=types.InlineKeyboardButton(text="Гуртожитки
"+hostel, callback_data="do")
        transport=types.InlineKeyboardButton(text="Як добратися?
"+maps, callback_data="tr")
        food=types.InlineKeyboardButton(text="Де поїсти? "+eat,
callback_data="fo")
        air=types.InlineKeyboardButton(text="Що робити під час
тривоги? "+red, callback_data="ai")
        questions=types.InlineKeyboardButton(text="Часті запитання
"+shock, callback_data="qu")
        markup.add( placement, кафедра, shedule, curriculum, price,
events, dormitory, transport, food, air, questions)
        await message.answer("Виберіть те, що вас цікавить "+look,
reply_markup=markup)
        file = open("tntu_start_stats.txt", "a")
        now = datetime.now()
        file.write(message.from_user.username + " " +
now.strftime("%d/%m/%Y, %H:%M") + "\n")
        file.close()

@dp.message_handler(content_types=['text'])

@dp.callback_query_handler(text="sh")
async def callback_query(call: types.CallbackQuery):
    shmarkup=types.InlineKeyboardMarkup(row_width=2)
    fis=types.InlineKeyboardButton(text="ФІС "+computer,
url="https://tntu.edu.ua/?p=uk/schedule&s=fis")
    fem=types.InlineKeyboardButton(text="ФЕМ "+stonks,
url="https://tntu.edu.ua/?p=uk/schedule&s=fem")

```

```

        fpt=types.InlineKeyboardButton(text="ФПТ           "+light,
url="https://tntu.edu.ua/?p=uk/schedule&s=fpt")
        fmt=types.InlineKeyboardButton(text="ФМТ           "+wrench,
url="https://tntu.edu.ua/?p=uk/schedule&s=fmt")
        back=types.InlineKeyboardButton(text="Головне           меню
"+bback, callback_data="back")
        shmarkup.add(fis, fem, fpt, fmt, back )
        shmarkup2=types.InlineKeyboardMarkup(row_width=1)
        shedulebut=types.InlineKeyboardButton(text="Розклад
дзвінків           та           тижнів           "+calendar,
url="https://nv.tntu.edu.ua/info.html")
        shmarkup2.add(shedulebut)
        await call.message.answer("Для того, щоб орієнтуватися у
розкладі необхідно знати декілька правил "+sphere+". \nГрупа
іноді може бути поділена на дві підгрупи, а пари можуть бути
поділені по першому та другому тижню. Більш детальне пояснення
знаходиться на зображенні "+winking+". \nДля того, щоб студент
орієнтувався, над розкладом зазвичай зазначається поточний
тиждень "+book+".", reply_markup=shmarkup2)
        img=open("shedule1.jpg", 'rb')
        await call.message.answer_photo(img)
        await call.message.answer("Оберіть факультет "+calm,
reply_markup=shmarkup)
        await call.answer()
        button_stats(call)

@dp.callback_query_handler(text="pp")
async def callback_query(call: types.CallbackQuery):
    backmarkup=types.InlineKeyboardMarkup(row_width=1)
    back=types.InlineKeyboardButton(text="Головне           меню
"+bback, callback_data="back")
    backmarkup.add(back)
    img=open("table1.png", 'rb')
    await call.message.answer_photo(img)
    await call.message.answer("Детальніше можна побачити
розміщення на карті за посиланням:
https://www.google.com/maps/d/viewer?mid=1CYr1ELkD1Pv6nbimq3kz
_uv6a1ClLJpK&ll", reply_markup=backmarkup)
    await call.answer()
    button_stats(call)

@dp.callback_query_handler(text="ev")
async def callback_query(call: types.CallbackQuery):
    evmarkup=types.InlineKeyboardMarkup(row_width=1)
    instagram=types.InlineKeyboardButton(text="ТНТУ           В
Instagram "+fire, callback_data="inst")

```

```

    youtube=types.InlineKeyboardButton(text="ТНТУ в Youtube
"+fire, callback_data="you")
    facebook=types.InlineKeyboardButton(text="ТНТУ в Facebook
"+fire, callback_data="face")
    tntu=types.InlineKeyboardButton(text="Сайт ТНТУ "+hearts,
url="https://tntu.edu.ua/?p=uk/main")
    back=types.InlineKeyboardButton(text="Головне меню
"+bback, callback_data="back")
    evmarkup.add(tntu, instagram, facebook, youtube, back)
    await call.message.answer("Тут ви можете знайти інформацію
про конференції, вебінари, зустрічі, посвяти, культурні заходи
та інші події нашого університету. "+calm,
reply_markup=evmarkup)
    await call.answer()
    button_stats(call)

```

```

@dp.callback_query_handler(text="inst")
async def callback_query(call: types.CallbackQuery):
    instmarkup=types.InlineKeyboardMarkup(row_width=2)
    tntuin=types.InlineKeyboardButton(text="ТНТУ",
url="https://instagram.com/tntu.edu.ua?igshid=YmMyMTA2M2Y=")
    tntutvin=types.InlineKeyboardButton(text="ТНТУ TV",
url="https://instagram.com/tntutv?igshid=YmMyMTA2M2Y=")
    studradain=types.InlineKeyboardButton(text="Студрада",
url="https://instagram.com/studradatntu?igshid=YmMyMTA2M2Y=")
    projectin=types.InlineKeyboardButton(text="Проектний
відділ",
url="https://instagram.com/project_tntu?igshid=YmMyMTA2M2Y=")
    fisin=types.InlineKeyboardButton(text="ФІС",
url="https://instagram.com/fis_tntu_?igshid=YmMyMTA2M2Y=")
    fptin=types.InlineKeyboardButton(text="ФПТ",
url="https://instagram.com/fpt_tntu?igshid=YmMyMTA2M2Y=")
    femin=types.InlineKeyboardButton(text="ФЕМ",
url="https://instagram.com/fem.tntu?igshid=YmMyMTA2M2Y=")
    fmtin=types.InlineKeyboardButton(text="ФМТ",
url="https://instagram.com/fmt_tntu?igshid=YmMyMTA2M2Y=")
    intin=types.InlineKeyboardButton(text="International",
url="https://instagram.com/i.tntu?igshid=YmMyMTA2M2Y=")
    knin=types.InlineKeyboardButton(text="Кафедра КН",
url="https://instagram.com/kn.tntu?igshid=YmMyMTA2M2Y=")
    kbin=types.InlineKeyboardButton(text="Кафедра КБ",
url="https://instagram.com/cybertntu?igshid=YmMyMTA2M2Y=")
    kiin=types.InlineKeyboardButton(text="Кафедра КІ",
url="https://instagram.com/kaf.ki.tntu?igshid=YmMyMTA2M2Y=")

```



```

        hbin=types.InlineKeyboardButton(text="Кафедра          ХБ",
url="https://instagram.com/kafedra_hb.tntu?igshid=YmMyMTA2M2Y="
")
        mkin=types.InlineKeyboardButton(text="Кафедра          МК",
url="https://instagram.com/prommarketingtntu?igshid=YmMyMTA2M2
Y=")
        viin=types.InlineKeyboardButton(text="Кафедра          ВІ",
url="https://instagram.com/kafedra.vi.tntu?igshid=YmMyMTA2M2Y="
")
        bmin=types.InlineKeyboardButton(text="Кафедра          БМ",
url="https://instagram.com/kafedra.bm.tntu?igshid=YmMyMTA2M2Y="
")
        btin=types.InlineKeyboardButton(text="Кафедра          БТ",
url="https://instagram.com/kafedra.bt.tntu?igshid=YmMyMTA2M2Y="
")
        efin=types.InlineKeyboardButton(text="Кафедра          ЕФ",
url="https://instagram.com/kafedraeif?igshid=YmMyMTA2M2Y=")
        back=types.InlineKeyboardButton(text="Головне          меню
"+bback, callback_data="back")
        cont=types.InlineKeyboardButton(text="Продовжити          тут
"+strilka, callback_data="ev")
        instmarkup.add(tntuin, tntutvin, studradain, projectin,
fisin, fptin, femin, fmtin, intin, knin, kbin, kiin, hbin,
mkin, viin, bmin, btin, efin, cont, back)
        await call.message.answer("Інстаграм сторінки ТНТУ, деяких
кафедр та факультетів "+laught, reply_markup=instmarkup)
        await call.answer()
        button_stats(call)

@dp.callback_query_handler(text="face")
async def callback_query(call: types.CallbackQuery):
    famarkup=types.InlineKeyboardMarkup(row_width=2)
    tntufa=types.InlineKeyboardButton(text="TNTU",
url="https://www.facebook.com/tntu.ip/")
    tntu2fa=types.InlineKeyboardButton(text="ТНТУ",
url="https://www.facebook.com/profile.php?id=100028447081552")
    polifa=types.InlineKeyboardButton(text="Політехнік",
url="https://www.facebook.com/profile.php?id=100026108259140")
    studfa=types.InlineKeyboardButton(text="Студрада",
url="https://www.facebook.com/groups/462312780776570/?ref=shar
e")
    tntutvfa=types.InlineKeyboardButton(text="ТНТУ          TV",
url="https://www.facebook.com/1tntu/")
    knfa=types.InlineKeyboardButton(text="Кафедра          КН",
url="https://www.facebook.com/kn.tntu/")

```

vmfa=types.InlineKeyboardButton (text="Кафедра BM",
url="https://www.facebook.com/vm.tntu/")
ktfa=types.InlineKeyboardButton (text="Кафедра KT",
url="https://www.facebook.com/kaf.kt.tntu/")
hbfa=types.InlineKeyboardButton (text="Кафедра XB",
url="https://www.facebook.com/hb.tntu/")
uifa=types.InlineKeyboardButton (text="Кафедра YI",
url="https://www.facebook.com/ui.tntu")
avfa=types.InlineKeyboardButton (text="Кафедра AB",
url="https://www.facebook.com/kaf.av.tntu/")
bmfa=types.InlineKeyboardButton (text="Кафедра BM",
url="https://www.facebook.com/kafedra.bm.tntu/")
mkfa=types.InlineKeyboardButton (text="Кафедра MK",
url="https://www.facebook.com/PromMarketingTNTU/")
kifa=types.InlineKeyboardButton (text="Кафедра KC",
url="https://www.facebook.com/ki.tntu/")
pifa=types.InlineKeyboardButton (text="Кафедра ПИ",
url="https://www.facebook.com/pi.tntu")
kbfa=types.InlineKeyboardButton (text="Кафедра KB",
url="https://www.facebook.com/cybertntu")
amfa=types.InlineKeyboardButton (text="Кафедра AM",
url="https://www.facebook.com/AM.TNTU")
mtfa=types.InlineKeyboardButton (text="Кафедра MT",
url="https://www.facebook.com/KafMTTNTU")
vifa=types.InlineKeyboardButton (text="Кафедра BI",
url="https://www.facebook.com/kafvi")
ohfa=types.InlineKeyboardButton (text="Кафедра OX",
url="https://www.facebook.com/KafOHT")
thfa=types.InlineKeyboardButton (text="Кафедра TX",
url="https://www.facebook.com/kaf.th.tntu")
btfa=types.InlineKeyboardButton (text="Кафедра BT",
url="https://www.facebook.com/kafedra.bt.tntu")
eifa=types.InlineKeyboardButton (text="Кафедра EI",
url="https://www.facebook.com/kaf.ei.tntu")
pvfa=types.InlineKeyboardButton (text="Кафедра ПВ",
url="https://www.facebook.com/kafedra.pv")
rtfa=types.InlineKeyboardButton (text="Кафедра PT",
url="https://www.facebook.com/kaf.rt.tntu")
mmfa=types.InlineKeyboardButton (text="Кафедра MM",
url="https://www.facebook.com/kaf.imm")
mnfa=types.InlineKeyboardButton (text="Кафедра MH",
url="https://www.facebook.com/mn.tntu/?ref=page_internal")
fzfa=types.InlineKeyboardButton (text="Кафедра ФЗ",
url="https://www.facebook.com/PhysicsDeptTNTU")
oafa=types.InlineKeyboardButton (text="Кафедра OA",
url="https://www.facebook.com/kafbo")

```

        effa=types.InlineKeyboardButton(text="Кафедра          ЕФ",
url="https://www.facebook.com/finance.tntu")
        befa=types.InlineKeyboardButton(text="Кафедра          БЕ",
url="https://www.facebook.com/ec.cyb.tntu")
        mafa=types.InlineKeyboardButton(text="Кафедра          МА",
url="https://www.facebook.com/tntumv")
        psfa=types.InlineKeyboardButton(text="Кафедра          ПС",
url="https://www.facebook.com/kaf.tntups")
        yzfa=types.InlineKeyboardButton(text="Кафедра          УЗ",
url="https://www.facebook.com/kaf.fil.tntu")
        mpfa=types.InlineKeyboardButton(text="Кафедра          МП",
url="https://www.facebook.com/kafmptntu")
        proffa=types.InlineKeyboardButton(text="Профспілка  ТНТУ",
url="https://www.facebook.com/tntuprofspilka")
        back=types.InlineKeyboardButton(text="Головне          меню
"+bback, callback_data="back")
        cont=types.InlineKeyboardButton(text="Продовжити      тут
"+strilka, callback_data="ev")
        famarkup.add(tntufa, polifa, studfa, tntutvfa, knfa, kifa,
pifa, kbfa, ktfa, vmfa, hbfa, uifa, avfa, bmfa, mkfa, amfa,
mtfa, vifa, ohfa, thfa, btfa, eifa, pvfa, rtfa, mmfa, mnfa,
fzfa, oafa, effa, befa, mafa, psfa, yzfa, mpfa, tntu2fa,
proffa, cont, back)
        await call.message.answer("Сторінки у Фейсбук ТНТУ та усіх
кафедр "+kapla, reply_markup=famarkup)
        await call.answer()
        button_stats(call)

@dp.callback_query_handler(text="you")
async def callback_query(call: types.CallbackQuery):
    youmarkup=types.InlineKeyboardMarkup(row_width=1)
    tntutvyou=types.InlineKeyboardButton(text="ТНТУ          ТВ",
url="https://youtube.com/c/TNTUTV")
    tntuyou=types.InlineKeyboardButton(text="ТНТУ",
url="https://youtube.com/c/%D0%A2%D0%9D%D0%A2%D0%A3%D1%96%D0%B
C%D0%86%D0%B2%D0%B0%D0%BD%D0%B0%D0%9F%D1%83%D0%BB%D1%8E%D1%8F"
)
    back=types.InlineKeyboardButton(text="Головне          меню
"+bback, callback_data="back")
    cont=types.InlineKeyboardButton(text="Продовжити      тут
"+strilka, callback_data="ev")
    youmarkup.add(tntutvyou, tntuyou, cont, back)
    await call.message.answer("Youtube канали "+demon+cool,
reply_markup=youmarkup)
    await call.answer()
    button_stats(call)

```

```

@dp.callback_query_handler(text="cu")
async def callback_query(call: types.CallbackQuery):
    cumarkup=types.InlineKeyboardMarkup(row_width=1)
    denn=types.InlineKeyboardButton(text="Графік денної форми
навчання", url="https://nv.tntu.edu.ua/files/graph_den.pdf")
    zao=types.InlineKeyboardButton(text="Графік заочної форми
навчання", url="https://nv.tntu.edu.ua/files/graph_zao.pdf")
    back=types.InlineKeyboardButton(text="Головне          меню
"+bback, callback_data="back")
    cumarkup.add(denn, zao, back)
    await call.message.answer("Оберіть, який графік
навчального процесу ви хочете побачити "+downarrow,
reply_markup=cumarkup)
    await call.answer()
    button_stats(call)

```

```

@dp.callback_query_handler(text="pr")
async def callback_query(call: types.CallbackQuery):
    prmarkup=types.InlineKeyboardMarkup(row_width=2)
    cost=types.InlineKeyboardButton(text="Вартість навчання",
url="https://tntu.edu.ua/?p=uk/admission/prices")
    paymentinfo=types.InlineKeyboardButton(text="Реквізити
оплати", url="https://tntu.edu.ua/?p=uk/about/payment")
    back=types.InlineKeyboardButton(text="Головне          меню
"+bback, callback_data="back")
    prmarkup.add(cost, paymentinfo, back)
    await call.message.answer("Що саме вас цікавить? "+euro,
reply_markup=prmarkup)
    await call.answer()
    button_stats(call)

```

```

@dp.callback_query_handler(text="ca")
async def callback_query(call: types.CallbackQuery):
    dekm Markup=types.InlineKeyboardMarkup(row_width=2)
    fisdek=types.InlineKeyboardButton(text="ФІС          "+computer,
callback_data="fisd")
    fptdek=types.InlineKeyboardButton(text="ФПТ          "+light,
callback_data="fptd")
    fmtdek=types.InlineKeyboardButton(text="ФМТ          "+wrench,
callback_data="fmtd")
    femdek=types.InlineKeyboardButton(text="ФЕМ          "+stonks,
callback_data="femd")
    buhdek=types.InlineKeyboardButton(text="Структурні
підрозділи "+star, callback_data="und")
    dekm Markup.add(fisdek, fptdek, fmtdek, femdek, buhdek)

```

```
    await call.message.answer("Тут зібрана інформація про  
розміщення університетських структур, кафедр та деканатів.  
"+light, reply_markup=dekmarkup)  
    await call.answer()  
    button_stats(call)
```

```
@dp.callback_query_handler(text="do")  
async def callback_query(call: types.CallbackQuery):  
    domarkup=types.InlineKeyboardMarkup(row_width=2)  
    back=types.InlineKeyboardButton(text="Головне          меню  
"+bback, callback_data="back")  
    address=types.InlineKeyboardButton(text="Детальніша  
інформація          "+maps,  
url="https://tntu.edu.ua/?p=uk/structure/students/sm/hostels")  
    settlement=types.InlineKeyboardButton(text="Деталі  
поселення          "+gmark,  
url="https://tntu.edu.ua/?p=uk/structure/students/sm/settle-  
in")  
    domarkup.add(address, settlement, back)  
    await call.message.answer("Гуртожиток №1 - вул.  
Шептицького, 13 \nГуртожиток №2 - вул. Замонастирська, 18  
\nГуртожиток №3 - вул. Тарнавського, 7а \n\nКарта:  
https://www.google.com/maps/d/edit?mid=1nd6nSKAzHObI3WtbpmiBZE  
p1LB_8p6Jd&usp=sharing ", reply_markup=domarkup)  
    await call.answer()  
    button_stats(call)
```

```
@dp.callback_query_handler(text="tr")  
async def callback_query(call: types.CallbackQuery):  
    backmarkup=types.InlineKeyboardMarkup(row_width=1)  
    back=types.InlineKeyboardButton(text="Головне          меню  
"+bback, callback_data="back")  
    backmarkup.add(back)  
    img=open("table2.png", 'rb')  
    await call.message.answer("Найближчі до корпусів зупинки,  
маршрути автобусів "+abus+" (Авт) і тролейбусів "+trolleybus+"  
(Тр).")  
    await call.message.answer_photo(img)  
    await call.message.answer("Більш          детально:  
http://detransport.com.ua/", reply_markup=backmarkup)  
    await call.answer()  
    button_stats(call)
```

```
@dp.callback_query_handler(text="fo")  
async def callback_query(call: types.CallbackQuery):  
    backmarkup=types.InlineKeyboardMarkup(row_width=1)
```

```

        backmarkup2=types.InlineKeyboardMarkup(row_width=1)
        din=types.InlineKeyboardButton(text="Їдальні "+plate,
callback_data="dini")
        coffeeshop=types.InlineKeyboardButton(text="Кав`ярні
"+coffee, callback_data="coffe")
        cafe=types.InlineKeyboardButton(text="Паби, кафе та
ресторани "+spagetti, callback_data="caf")
        fastfood=types.InlineKeyboardButton(text="Фастфуд
"+hamburger, callback_data="fast")
        pizza=types.InlineKeyboardButton(text="Піцерії "+pizzas,
callback_data="pi")
        back=types.InlineKeyboardButton(text="Головне меню
"+bback, callback_data="back")
        backmarkup.add(din, coffeeshop, cafe, fastfood, pizza)
        backmarkup2.add(back)
        await call.message.answer("Тут ми підібрали заклади
поблизу ТНТУ, де можна перекусити "+tongue+pan,
reply_markup=backmarkup)
        await call.message.answer("Також навпроти 2-го корпусу є
магазин Сім23 який має власну випічку"+cookie+",
хотдоги"+hotdog+" та каву"+coffee+".",
reply_markup=backmarkup2)
        await call.answer()
        button_stats(call)

@dp.callback_query_handler(text="dini")
async def callback_query(call: types.CallbackQuery):
    backmarkup=types.InlineKeyboardMarkup(row_width=1)
    back=types.InlineKeyboardButton(text="Головне меню
"+bback, callback_data="back")
    cont=types.InlineKeyboardButton(text="Продовжити тут
"+strilka, callback_data="fo")
    backmarkup.add(cont, back)
    await call.message.answer("Їдальні "+plate+":\n1. Їдальня
ТНТУ, у дворі 2-го корпусу. \n2. ''Бурячок'', вул. Руська 52.
\n3. ''Пиріжкова'', вул. Руська 23. \n\nРозміщення на карті:
https://www.google.com/maps/d/edit?mid=1xBg3odJo08ttZ9AR8q48Kn2elQH\_BtEq&usp=sharing ", reply_markup=backmarkup)
    await call.answer()
    button_stats(call)

@dp.callback_query_handler(text="coffe")
async def callback_query(call: types.CallbackQuery):
    backmarkup=types.InlineKeyboardMarkup(row_width=1)
    back=types.InlineKeyboardButton(text="Головне меню
"+bback, callback_data="back")

```

```

        cont=types.InlineKeyboardButton(text="Продовжити тут
"+strilka, callback_data="fo")
        backmarkup.add(cont, back)
        await call.message.answer("Кав`ярні "+coffee+": \n1.
''BarBarista'', вул. Йосипа Сліпого 1/25. \n2. ''Zerno'', вул.
Митрополита Шептицького 13.\n3. ''Карма кава'', вул. Гетьмана
Сагайдачного 13.\n4. ''Ровер'', б-р Тараса Шевченка 10.\n5.
''Такава'', вул. Дмитра Січинського 4.\n6. ''Star Coffee'',
вул. Гетьмана Сагайдачного 8.\n\nРозміщення на карті:
https://www.google.com/maps/d/edit?mid=1ri26Lp5sa7f2Sf7qE-Y9EIT2vbIJ0vB\_&usp=sharing ", reply_markup=backmarkup)
        await call.answer()
        button_stats(call)

```

```

@dp.callback_query_handler(text="caf")
async def callback_query(call: types.CallbackQuery):
    backmarkup=types.InlineKeyboardMarkup(row_width=1)
    back=types.InlineKeyboardButton(text="Головне меню
"+bback, callback_data="back")
    cont=types.InlineKeyboardButton(text="Продовжити тут
"+strilka, callback_data="fo")
    backmarkup.add(cont, back)
    await call.message.answer("Паби, кафе,
ресторани"+spagetti+":\n1. ''Файне місто'', б-р Тараса
Шевченка 23. \n2. ''Клуб Файного міста'', вул. Олени
Кульчицької 1. \n3. ''Pasta Fresca'', б-р Тараса Шевченка
23.\n4. ''Пасаж Адлера 13'', б-р Тараса Шевченка 25. \n5. ''Le
Rock'', вул. Олени Кульчицької 2. \n6. ''Barbaresco'', вул.
Князя Острозького 14. \n7. ''Lampa'', вул. Грушевського 1.
\n8. ''DEPO'', вул. Богдана Хмельницького 10А.\n9. ''Lviv
Croissant'', вул. Валова 18. \n\nРозміщення на карті:
https://www.google.com/maps/d/edit?mid=1Q8A\_Jabr6vjJNVqJVlS8bLwNkU2oTrMJ&usp=sharing ", reply_markup=backmarkup)
    await call.answer()
    button_stats(call)

```

```

@dp.callback_query_handler(text="fast")
async def callback_query(call: types.CallbackQuery):
    backmarkup=types.InlineKeyboardMarkup(row_width=1)
    back=types.InlineKeyboardButton(text="Головне меню
"+bback, callback_data="back")
    cont=types.InlineKeyboardButton(text="Продовжити тут
"+strilka, callback_data="fo")
    backmarkup.add(cont, back)
    await call.message.answer("Фастфуд "+hamburger+": \n1.
''Алі-Баба'', вул. Руська 28. \n2. ''I love kebab'', вул.

```

```
Руська 49. \n3. 'Merci', вул. Сагайдачного 2. \n4. 'Chicken HUT', вул. Пирогова 2А. \n\nРозміщення на карті: https://www.google.com/maps/d/edit?mid=1eaQepeiGnULE_vNDXisTW67vzot9OzqO&usp=sharing ", reply_markup=backmarkup)
```

```
    await call.answer()  
    button_stats(call)
```

```
@dp.callback_query_handler(text="pi")
```

```
async def callback_query(call: types.CallbackQuery):
```

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

```
    back=types.InlineKeyboardButton(text="Головне меню  
"+bback, callback_data="back")
```

```
    cont=types.InlineKeyboardButton(text="Продовжити тут  
"+strilka, callback_data="fo")
```

```
    backmarkup.add(cont, back)
```

```
    await call.message.answer("Піцерії "+pizzas+":\n1. 'Добра піца', вул. Богдана Хмельницького 17.\n2. 'Стара піцерія', вул. Шашкевича 1. \n3. 'Фламінго', вул. Анатолія Живова 7. \n4. 'La Pizza', вул. Чорновола 11. \n\nРозміщення на карті: https://www.google.com/maps/d/edit?mid=17DXGpswruSoJlXojRbcszGTWYQb-aHMb&usp=sharing ", reply_markup=backmarkup)
```

```
    await call.answer()  
    button_stats(call)
```

```
@dp.callback_query_handler(text="fisd")
```

```
async def callback_query(call: types.CallbackQuery):
```

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

```
    back=types.InlineKeyboardButton(text="Головне меню  
"+bback, callback_data="back")
```

```
    backmarkup.add(back)
```

```
    await call.message.answer("Факультет комп'ютерно-інформаційних систем та програмної інженерії (ФІС)\nДеканат - корпус № 1, к.607 "+hearts+"\nКафедра інформатики і математичного моделювання (ММ) - корпус № 2, к.62 \nКафедра кібербезпеки (КБ) - корпус № 1, к.804 \nКафедра комп'ютерних наук (КН) - корпус № 1, к.706 \nКафедра комп'ютерних систем та мереж (КС) - корпус № 1, к.604 \nКафедра математичних методів в інженерії (МН) - корпус № 1, к.810, 707 \nКафедра програмної інженерії (ПІ) - корпус №1, к.104 \nКафедра української та іноземних мов (УІ) - корпус № 1, к.307 \nКафедра фізики (ФЗ) - корпус № 2, к.66 ", reply_markup=backmarkup)
```

```
    await call.answer()  
    button_stats(call)
```

```
@dp.callback_query_handler(text="fptd")
```

```
async def callback_query(call: types.CallbackQuery):
```



```

    backmarkup=types.InlineKeyboardMarkup(row_width=1)
    back=types.InlineKeyboardButton(text="Головне меню"
    "+bback, callback_data="back")
    backmarkup.add(back)
    await call.message.answer("Факультет прикладних
    інформаційних технологій та електроінженерії (ФПТ)\nДеканат -
    корпус № 2, к.41 "+hearts+"\nКафедра автоматизації
    технологічних процесів і виробництв (АВ) - корпус №1, к.408,
    409 \nКафедра біотехнічних систем (БТ) - корпус №9, к.9-502
    \nКафедра вищої математики (ВМ) - корпус № 1, к.305 \nКафедра
    електричної інженерії (ЕІ) - корпус № 7, к.403 \nКафедра
    комп'ютерно-інтегрованих технологій (КТ) - корпус № 1, к.410
    \nКафедра приладів і контрольно-вимірювальних систем (ПВ) -
    корпус № 9, к.304 \nКафедра радіотехнічних систем (РТ) -
    корпус № 9, к.611 ", reply_markup=backmarkup)
    await call.answer()
    button_stats(call)

```

```

@dp.callback_query_handler(text="fmd")
async def callback_query(call: types.CallbackQuery):
    backmarkup=types.InlineKeyboardMarkup(row_width=1)
    back=types.InlineKeyboardButton(text="Головне меню"
    "+bback, callback_data="back")
    backmarkup.add(back)
    await call.message.answer("Факультет інженерії машин,
    споруд та технологій (ФМТ)\nДеканат - корпус № 2, к.46
    "+hearts+"\nКафедра автомобілів (АМ) - корпус № 9, к.201б
    \nКафедра будівельної механіки (БМ) - корпус № 2, к.47
    \nКафедра інжинірингу машинобудівних технологій (МТ) - корпус
    № 3, к.8 \nКафедра конструювання верстатів, інструментів та
    машин (ВІ) - корпус № 4 \nКафедра обладнання харчових
    технологій (ОХ) - корпус № 6, к.18 \nКафедра технічної
    механіки та сільськогосподарських машин (ТХ) - корпус № 2,
    к.76 \nКафедра фізичного виховання і спорту (ФС) - корпус № 10
    \nКафедра харчової біотехнології і хімії (ХБ) - корпус № 5,
    к.5", reply_markup=backmarkup)
    await call.answer()
    button_stats(call)

```

```

@dp.callback_query_handler(text="femd")
async def callback_query(call: types.CallbackQuery):
    backmarkup=types.InlineKeyboardMarkup(row_width=1)
    back=types.InlineKeyboardButton(text="Головне меню"
    "+bback, callback_data="back")
    backmarkup.add(back)

```

```

        await call.message.answer("Факультет економіки та менеджменту (ФЕМ)\nДеканат - корпус № 2, к.44 "+hearts+"\nКафедра бухгалтерського обліку та аудиту (ОА) - корпус № 5, к.20, 22 \nКафедра економіки та фінансів (ЕФ) - корпус № 10, к.204 \nКафедра економічної кібернетики (БЕ) - корпус № 5, к.7\nКафедра менеджменту та адміністрування (МА) - корпус № 7, к.215 \nКафедра промислового маркетингу (МК) - корпус № 10, к.209 \nКафедра психології (ПС) - корпус № 7, к.208 \nКафедра українознавства та філософії (УЗ) - корпус № 1, к.304, 303 \nКафедра управління інноваційною діяльністю та сферою послуг (МП) - корпус №7, к.305", reply_markup=backmarkup)
        await call.answer()
        button_stats(call)

```

```

@dp.callback_query_handler(text="und")
async def callback_query(call: types.CallbackQuery):
    backmarkup=types.InlineKeyboardMarkup(row_width=1)
    back=types.InlineKeyboardButton(text="Головне меню "+bback, callback_data="back")
    backmarkup.add(back)
    await call.message.answer("Ректорат - корпус №1, 2-й поверх "+hands+"\nСтудмістечко - корпус №8, к.21а \nПриймальна комісія - корпус №1, к.107 \nЧитальний зал - корпус №2, к.58\nДиректор бібліотеки - корпус №2, к.56\nЕлектронний читальний зал - корпус №2, к.54\nВійськово обліковий сектор - корпус №2, к.49\nБухгалтерія - корпус №1, 2-й поверх, телефон:(0352)51-97-08", reply_markup=backmarkup)
    await call.answer()
    button_stats(call)

```

```

@dp.callback_query_handler(text="qu")
async def callback_query(call: types.CallbackQuery):
    backmarkup=types.InlineKeyboardMarkup(row_width=1)
    zvit=types.InlineKeyboardButton(text="Де можна роздрукувати звіт? "+tipok, callback_data="zv")
    tern=types.InlineKeyboardButton(text="Як отримати карту тернополянина? "+hands, callback_data="ter")
    moneytern=types.InlineKeyboardButton(text="Де поповнити карту тернополянина поблизу ТНТУ? "+money, callback_data="mont")
    sports=types.InlineKeyboardButton(text="Спортивні секції "+ball, url="https://kaf-fv.tntu.edu.ua/html/sektsij.html")
    swimming=types.InlineKeyboardButton(text="Оздоровче плавання "+swims, url="https://kaf-fv.tntu.edu.ua/html/Ozdorovche%20plavannja.htm")

```

```

        arts=types.InlineKeyboardButton(text="Хто допоможе
організувати мистецький захід? "+dance,
url="https://tntu.edu.ua/?p=uk/about/pedagogical-work")
        back=types.InlineKeyboardButton(text="Головне меню
"+bback, callback_data="back")
        backmarkup.add(zvit, tern, moneytern, sports, swimming,
arts, back)
        await call.message.answer("Тут зібрані найбільш часті
запитання "+downarrow, reply_markup=backmarkup)
        await call.answer()
        button_stats(call)

```

```

@dp.callback_query_handler(text="zv")
async def callback_query(call: types.CallbackQuery):
    backmarkup=types.InlineKeyboardMarkup(row_width=1)
    back=types.InlineKeyboardButton(text="Головне меню
"+bback, callback_data="back")
    backmarkup.add(back)
    await call.message.answer("Де можна роздрукувати звіт?
"+tipok+"\n1. На 1-му поверсі 1-го корпусу біля ліфту сидить
прекрасна жіночка"+dama+", яка роздрукує файл з флешки
"+smile+"\n2. За магазином Сім23 є інтернет-кафе Крапка, де
роздрукують файл як з флешки, так і з соціальних мереж
(Telegram, Viber) а також зшиють курсові. "+skrepka+"\n3. На
1-му поверсі ЦУМу є відділ для друку та сканування документів.
"+printer, reply_markup=backmarkup)
    await call.answer()
    button_stats(call)

```

```

@dp.callback_query_handler(text="ter")
async def callback_query(call: types.CallbackQuery):
    backmarkup=types.InlineKeyboardMarkup(row_width=1)
    back=types.InlineKeyboardButton(text="Головне меню
"+bback, callback_data="back")
    backmarkup.add(back)
    await call.message.answer("Як отримати карту
тернополянина? "+hands+"\nІнформацію про перелік документів та
пункти видачі можна отримати тут: https://fayna-karta.te.ua/
", reply_markup=backmarkup)
    await call.answer()
    button_stats(call)

```

```

@dp.callback_query_handler(text="ai")
async def callback_query(call: types.CallbackQuery):
    backmarkup=types.InlineKeyboardMarkup(row_width=1)

```

```

        bunk=types.InlineKeyboardButton(text="Розміщення сховищ
"+bun, callback_data="bunker")
        algo=types.InlineKeyboardButton(text="Алгоритм дій "+alg,
callback_data="algorithm")
        back=types.InlineKeyboardButton(text="Головне меню
"+bback, callback_data="back")
        backmarkup.add(bunk, algo, back)
        await call.message.answer("Виберіть те, що вас цікавить
"+look, reply_markup=backmarkup)
        await call.answer()
        button_stats(call)

```

```

@dp.callback_query_handler(text="bunker")
async def callback_query(call: types.CallbackQuery):
    backmarkup=types.InlineKeyboardMarkup(row_width=1)
    back=types.InlineKeyboardButton(text="Головне меню
"+bback, callback_data="back")
    backmarkup.add(back)
    await call.message.answer("Перелік сховищ поблизу корпусів
ТНТУ: "+bun+"\n1. Корпуси № 1, 2 – підвальне приміщення
Корпусу №1.\n2. Корпуси № 3, 4 – підвальне приміщення корпусу
№4.\n3. Корпус № 5 – Тернопільська обласна філармонія
(Острозького, 11).\n4. Корпуси № 6, 8 – Обласний комунальний
центр НТТ школярів та молоді (Січових Стрільців, 13). \n5.
Корпус № 7 – 'Ватра' (Микулинецька, 46).\n6. Корпус № 9 – ТОВ
"Теркурій-2" (ТРЦ "Подoliaни"). \n7. Корпус № 10 – підвальне
приміщення. \n8. Гуртожиток № 1 – Тернопільський обласний
краєзнавчий музей (Героїв Євромайдану, 3). \n9. Гуртожиток № 2
– Тернопільський комбінат хлібопродуктів (Гайова, 21), ЗОШ №
13 (Острозького, 51).\n\nРозміщення на карті:
https://www.google.com/maps/d/edit?mid=1ToBXXvZV2awWhMnRoLHluxFvS9lcVBA&usp=sharing ", reply_markup=backmarkup)
    await call.answer()
    button_stats(call)

```

```

@dp.callback_query_handler(text="algorithm")
async def callback_query(call: types.CallbackQuery):
    backmarkup=types.InlineKeyboardMarkup(row_width=1)
    back=types.InlineKeyboardButton(text="Головне меню
"+bback, callback_data="back")
    backmarkup.add(back)
    await call.message.answer("Порядок дій при оголошенні
повітряної тривоги: "+hands+"\n1. Після трьох довгих дзвінків
викладач припиняє заняття.\n2. Студенти організовано за
правилом правостороннього руху переходять в укриття.\n3.
Працівники та студенти, які знаходяться на подвір'ї закладу,

```

самостійно переходять в укриття.\n4. Під час перебування в укритті зберігати спокій. Після відбою повернутись до навчання за розкладом.", reply_markup=backmarkup)

```
    await call.answer()
```

```
    button_stats(call)
```

```
@dp.callback_query_handler(text="mont")
```

```
async def callback_query(call: types.CallbackQuery):
```

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

```
    punkt=types.InlineKeyboardButton(text="Всі пункти поповнення "+maps, url="https://fayna-karta.te.ua/2022/05/10/%d0%b0%d0%b4%d1%80%d0%b5%d1%81%d0%b8-%d0%bf%d1%83%d0%bd%d0%ba%d1%82%d1%96%d0%b2-%d0%bf%d0%be%d0%bf%d0%be%d0%b2%d0%bd%d0%b5%d0%bd%d0%bd%d1%8f-%d0%b5%d0%bb%d0%b5%d0%ba%d1%82%d1%80%d0%be%d0%bd%d0%bd-2/")
```

```
    back=types.InlineKeyboardButton(text="Головне меню "+bback, callback_data="back")
```

```
    backmarkup.add(punkt, back)
```

```
    await call.message.answer("Де можна поповнити карту тернополянина поблизу ТНТУ? "+money+"\n1. Магазин "Урожай", вул. Руська, 17, зупинка "Кооперативний технікум"\n2. Кіоск "Мерсі", вул. Руська, 14 (Книгарня)\n3. Кіоск "Мерсі", Театральний майдан\n4. Магазин "Поляна", вул. С. Крушельницької, 1, зупинка "Вул. С. Крушельницької"\n5. Кіоск "Мерсі", вул. Пирогова, 16 (Міська лікарня №1)\n6. Кіоск "Мерсі", вул. Митрополита Шептицького, 3 (навпроти зуп. "вул. М. Шептицького") \n\nНа карті:
```

```
https://www.google.com/maps/d/edit?mid=1So-nK6HxX_-Ve36h-52WkkzS9wWxrNk_&usp=sharing ", reply_markup=backmarkup)
```

```
    await call.answer()
```

```
    button_stats(call)
```

```
@dp.callback_query_handler(text="back")
```

```
async def callback_query(call: types.CallbackQuery):
```

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

```
    shedule=types.InlineKeyboardButton(text="Розклад "+paper, callback_data="sh")
```

```
    placement=types.InlineKeyboardButton(text="Розміщення корпусів "+building, callback_data="pp")
```

```
    events=types.InlineKeyboardButton(text="Події "+event, callback_data="ev")
```

```
    curriculum=types.InlineKeyboardButton(text="Графік навчального процесу "+calendar, callback_data="cu")
```

```
    price=types.InlineKeyboardButton(text="Вартість навчання та оплата "+money, callback_data="pr")
```

```
        cafedra=types.InlineKeyboardButton(text="Розміщення кафедр  
та деканатів "+moon, callback_data="ca")  
        dormitory=types.InlineKeyboardButton(text="Гуртожитки  
"+hostel, callback_data="do")  
        transport=types.InlineKeyboardButton(text="Як добратися?  
"+maps, callback_data="tr")  
        food=types.InlineKeyboardButton(text="Де поїсти? "+eat,  
callback_data="fo")  
        air=types.InlineKeyboardButton(text="Що робити під час  
тривоги? "+red, callback_data="ai")  
        questions=types.InlineKeyboardButton(text="Часті запитання  
"+shock, callback_data="qu")  
        markup.add(placement, cafedra, shedule, curriculum, price,  
events, dormitory, transport, food, air, questions)  
        await call.message.answer("Виберіть те, що вас цікавить  
"+look, reply_markup=markup)  
        await call.answer()  
  
if __name__ == "__main__":  
    executor.start_polling(dp, skip_updates=True)
```

Теза наукової конференції

Міністерство освіти і науки України,
Тернопільський національний технічний університет
імені Івана Пулюя
Маріборський університет (Словенія)
Технічний університет в Кошице (Словаччина)
Каунаський технологічний університет (Литва)
Львівський національний університет
імені Івана Франка,
Гірничо-металургійна академія ім. Станіслава Сташиця (Польща)
Луцький національний технічний університет,
Чернівецький національний університет
імені Юрія Федьковича,
Вроцлавський економічний університет (Польща)
Університет технологій та економіки
імені Хелени Ходковської (Польща)
Донбаська державна машинобудівна академія



*Студентське наукове
товариство*



VI МІЖНАРОДНА
студентська науково - технічна конференція
"ПРИРОДНИЧІ ТА ГУМАНІТАРНІ
НАУКИ.

АКТУАЛЬНІ ПИТАННЯ"

27-28 квітня 2023 р.

(збірник тез конференції)

Тернопіль 2023

УДК 004

Шишкіна В. – ст. гр. СНм-51.

Тернопільський національний технічний університет імені Івана Пулюя

ОСНОВНІ ТЕНДЕНЦІЇ РОЗВИТКУ ЧАТ-БОТІВ

Науковий керівник: канд. тех. наук, доцент Козбур Г.

Shyshkina V.

Ternopil Ivan Pul'uj National Technical University

MAIN TRENDS IN THE DEVELOPMENT OF CHAT-BOTS

Supervisor: Ph.D., Assoc.Prof. Kozbur H.

Ключові слова: чат-бот, месенджер, машинне навчання, IoT

Key words: chatbot, messenger, machine learning, IoT

Чат-боти - це програми, які можуть автоматизувати взаємодію з користувачами через текстові повідомлення в месенджерах, таких як Telegram, Facebook Messenger, Viber, і багатьох інших. Розвиток чат-ботів є актуальною темою для досліджень, оскільки вони стають все більш популярними в різних сферах бізнесу та повсякденного життя.

В останні роки чат-боти стали популярними інструментами для взаємодії з користувачами. Нові можливості технологій штучного інтелекту та машинного навчання, а також зростаюча популярність голосових інтерфейсів забезпечують розвиток нових підходів до створення чат-ботів.

Однією з основних тенденцій є використання штучного інтелекту для поліпшення розуміння мови та відповідей чат-ботів. Технології машинного навчання та глибокого навчання дозволяють чат-ботам вчитися на основі взаємодії з користувачами, що дозволяє їм стати все більш ефективними та точними в своїх відповідях. Зокрема вони можуть виконувати завдання з організації зустрічей, покупки товарів, резервування квитків і готелів. Чат-боти також можуть використовуватися для збору та аналізу даних про користувачів, що дає можливість розуміти їхні потреби та бажання.

Ще одна важлива тенденція полягає у розвитку голосових інтерфейсів для чат-ботів. Голосові інтерфейси забезпечують зручну та швидку взаємодію з користувачами, зокрема в тих ситуаціях, коли неможливо або незручно використовувати клавіатуру чи екран. Однак, для забезпечення ефективної роботи таких чат-ботів, необхідно вирішувати складні завдання з обробки природньої мови та розуміння інтонацій користувачів, що залишає відкриті питання для подальшого дослідження[1].

Також можна відзначити зростаючу популярність чат-ботів для роботи з різними пристроями Інтернету Речей (IoT). За рахунок зростання популярності IoT є необхідність розширення функціональних можливостей чат-ботів, що забезпечують зручну та автоматизовану взаємодію з користувачами різних пристроїв IoT. Завдяки цьому, користувачам стає доступним широкий спектр функціональних можливостей, які можуть бути виконані за допомогою чат-ботів, таких як контроль освітлення в квартирі, регулювання температури, відкриття та закриття вікон, підключення до систем безпеки, збір та аналіз даних про стан пристроїв і багато іншого[2]. Однією з найважливіших переваг використання чат-ботів для пристроїв IoT є можливість контролювати їх дистанційно та віддалено. Однак, використання чат-ботів для пристроїв IoT також стикається з деякими викликами, такими як складність розуміння та інтерпретації команд від користувачів, складність взаємодії зі старими моделями пристроїв, та необхідність забезпечення високого рівня безпеки даних.

Зростаюча популярність чат-ботів як засобу взаємодії з користувачами свідчить про важливість їх ролі в розвитку бізнесу та покращенні досвіду взаємодії з різними платформами та сервісами. Особливим напрямком у цьому процесі є розвиток нових підходів до взаємодії з користувачами, зокрема, використання чат-ботів може допомогти створити унікальний та індивідуальний досвід для кожного користувача, що підвищить їх задоволення від взаємодії з брендом або платформою, а також сприяє збільшенню рівня лояльності користувачів[3]. Для досягнення цих цілей можуть бути використані різноманітні технології, такі як машинне навчання та аналітика даних, які дозволяють забезпечити індивідуальний підхід до кожного користувача та забезпечити йому унікальний досвід взаємодії з чат-ботом. Однак, для досягнення успіху у цій галузі, необхідно враховувати особливості цільової аудиторії, робити акцент на якість та швидкість відповідей, а також надавати можливості для подальшої персоналізації взаємодії з чат-ботом.

Такі чат-боти можуть бути використані для створення індивідуальних рекомендацій, персональних пропозицій та навіть для підтримки емоційного стану користувача.

Загалом, розвиток чат-ботів відкриває безліч можливостей для створення зручної та ефективної взаємодії з користувачами в різних галузях, таких як бізнес, медицина, освіта та багато інших.

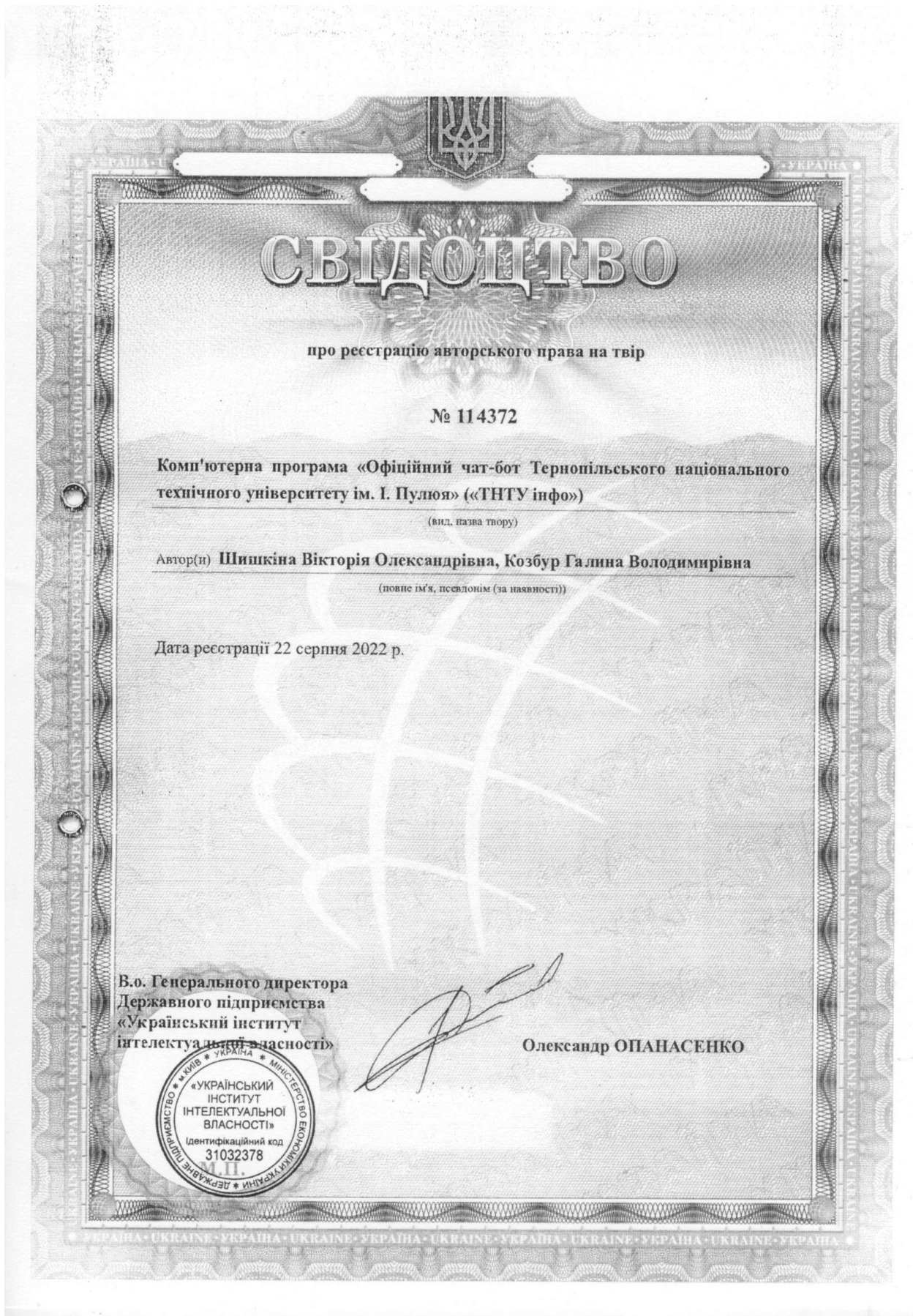
Однак, на думку деяких дослідників, існує потреба в подальшому дослідженні проблем взаємодії з користувачами, зокрема забезпечення безпеки та конфіденційності, а також забезпечення максимальної доступності чат-ботів для людей з різними потребами та обмеженнями.

Таким чином, можна зробити висновок, що розвиток чат-ботів є актуальною темою для досліджень та відкриває безліч можливостей для покращення взаємодії з користувачами в різних галузях.

Література:

1. Features of Chatbots and Voice Interfaces [Електронний ресурс] – Режим доступу до ресурсу: <https://www.merementor.com/blog/voice-interfaces-and-chatbots>.
2. Enhancing Smart Home Design with AI Models [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mdpi.com/1996-1073/16/6/2636>.
3. AI-based chatbots [Електронний ресурс] – Режим доступу до ресурсу: <https://link.springer.com/article/10.1007/s12525-020-00414-7>.

Авторське право



СВІДОЦТВО

про реєстрацію авторського права на твір

№ 114372

Комп'ютерна програма «Офіційний чат-бот Тернопільського національного технічного університету ім. І. Пулюя» («ТНТУ інфо»)

(вид, назва твору)

Автор(и) Шижкіна Вікторія Олександрівна, Козбур Галина Володимирівна

(повне ім'я, псевдонім (за наявності))

Дата реєстрації 22 серпня 2022 р.

В.о. Генерального директора
Державного підприємства
«Український інститут
інтелектуальної власності»

Олександр ОПАНАСЕНКО

