ДЕРЖАВНИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

Кафедра комп'ютерних наук та інформаційних технологій

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЕКТ

на тему:

«Алгоритм та структура чат-бота для електронної черги Центру надання адміністративних послуг»

Студента 4 курсу, 8 групи,
спеціальності
122 «Комп'ютерні науки» _____ *підпис студента* Колб Анастасія Юріївна

Науковий керівник
кандидат педагогічних наук, доцент _____ *підпис керівника* Базурін Віталій Миколайович

Гарант освітньої програми
кандидат технічних наук, професор _____ *підпис керівника* Демідов Павло Георгійович

Київ 2023

Державний торговельно-економічний університет

Факультет інформаційних технологій
Кафедра комп'ютерних наук та інформаційних систем
Спеціальність 122 «Комп'ютерні науки»

Затверджую
Зав. кафедри _____Пурський О.І.
«12» грудня 2022р.

Завдання
на випускну кваліфікаційну роботу (проект) студентці

Колб Анастасії Юріївні
(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи (проекту)
«Алгоритм та структура чат-бота для електронної черги Центру надання адміністративних послуг»
Затверджена наказом ректора від *«09» грудня 2022 р. № 3332*
2. Строк здачі студентом закінченої роботи *30 травня 2023 року*
3. Цільова установка та вихідні дані до роботи
Мета роботи: розробка чат-бота електронної черги Центру надання адміністративних послуг
Об'єкт дослідження: особливості обміну інформацією у чат-ботах.
Предмет дослідження: алгоритм та структура чат-бота
4. Перелік графічного матеріалу_____
_____
_____
_____

5. Консультанти по роботі із зазначенням розділів, за якими здійснюється консультування:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Базурін В.М	15.12.2022 р.	15.12.2022 р.
2	Базурін В.М	15.12.2022 р.	15.12.2022 р.
3	Базурін В.М	15.12.2022 р.	15.12.2022 р.

6. Зміст випускної кваліфікаційної роботи (проекту) (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. АНАЛІТИЧНЕ ДОСЛІДЖЕННЯ СПЕЦИФІКИ ЗАСТОСУВАННЯ ЧАТ-БОТІВ ДЛЯ ОБМІНУ ДАНИМИ МІЖ КОРИСТУВАЧАМИ

1.1. Аналіз стану проблеми

1.2. Особливості обміну даними в мережі Інтернет

1.3 Огляд існуючих програмних рішень

РОЗДІЛ 2. РОЗРОБКА МОДЕЛІ ЧАТ-БОТА

2.1 Загальна концепція

2.2 Моделі обміну даними

2.3. Алгоритми обміну даними

РОЗДІЛ 3. РОЗРОБКА ЧАТ-БОТУ

3.1 Розробка інформаційно-логічної моделі

3.2 Технологія використання чат-боту

3.3 Програмна реалізація чат-боту та його тестування

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТОК

7. Календарний план виконання роботи

№ Пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	фактично
1	2	3	4
1	*Вибір теми випускної кваліфікаційної роботи*	*04.10.2022*	*04.10.2022*
2	*Розробка та затвердження завдання на випускну кваліфікаційну роботу*	*15.12.2022*	*15.12.2022*
3	*Вступ*	*03.02.2023*	*03.02.2023*
4	*РОЗДІЛ 1. Аналітичне дослідження специфіки застосування чат-ботів для обміну даними між користувачами*	*28.02.2023*	*28.02.2023*
5	*РОЗДІЛ 2. Розробка моделі чат-бота*	*06.04.2023*	*06.04.2023*
6	*РОЗДІЛ 3. . Розробка чат-боту*	*12.05.2023*	*12.05.2023*
7	*Висновки*	*15.05.2023*	*15.05.2023*
8	*Здача випускної кваліфікаційної роботи на кафедру науковому керівнику*	*30.05.2023*	*30.05.2023*
9	*Попередній захист випускної кваліфікаційної роботи*	*31.05.2023 -01.06.2023*	*31.05.2023 -01.06.2023*

11	*Виправлення зауважень, зовнішнє рецензування випускної кваліфікаційної роботи*	*02.06.2023*	*02.06.2023*
12	*Представлення готової зшитої випускної кваліфікаційної роботи на кафедру*	*05.06.2023*	*05.06.2023*
13	*Публічний захист випускної кваліфікаційної роботи*	*За розкладом роботи ЕК*	

8. Дата видачі завдання «15» грудня 2022 р.

Керівник випускної кваліфікаційної роботи (проекту)
Базурін В.М.
(прізвище, ініціали, підпис)

Гарант освітньої програми Демідов П.Г.
(прізвище, ініціали, підпис)

Завдання прийняв студент-дипломник Колб А.Ю.
(прізвище, ініціали, підпис)

9. Відгук керівника випускної кваліфікаційної роботи (проекту)

Керівник випускної кваліфікаційної роботи (проекту)
_____ 30.05.2023 р.
(підпис, дата)

10. Висновок про випускну кваліфікаційну роботу

Випускна кваліфікаційна робота студента _____
(прізвище, ініціали)
може бути допущена до захисту в екзаменаційній комісії.

Гарант освітньої програми_____Демідов П.Г.____
(підпис, прізвище, ініціали)

Завідувач кафедри _____Пурський О.І.___
(підпис, прізвище, ініціали)

«_____»_____2023 р.

Анотація

Ця випускна кваліфікаційна робота присвячена розробці та впровадженню системи чат-ботів для управління електронною чергою в Центрі надання адміністративних послуг. Метою є автоматизація процесу обробки запитів користувачів, надання інформації в режимі реального часу та оптимізація системи керування чергами. Дослідження показує алгоритм і структуру чат-бота, що включають методи обробки природної мови та ефективне управління даними. Запропоноване рішення має потенціал для покращення взаємодії з користувачем, скорочення часу очікування та підвищення загальної ефективності надання адміністративних послуг.

Ключові слова: чат-бот, електронна черга, адміністративні послуги, алгоритм, структура, обробка природної мови, досвід користувача.

Anotation

This graduation qualification work is devoted to the development and implementation of a chatbot system for electronic queue management in the Center for the provision of administrative services. The goal is to automate the process of processing user requests, provide real-time information, and optimize the queue management system. The study shows the algorithm and structure of the chatbot, which includes natural language processing methods and efficient data management. The proposed solution has the potential to improve interaction with the user, reduce waiting time and increase the overall efficiency of the provision of administrative services.

Keywords: chatbot, electronic queue, administrative services, algorithm, structure, natural language processing, user experience.

Список умовних скорочень

ЦНАП – Центр надання адміністративних послуг

ШІ – Штучний інтелект

NLP – Natural Language Processing

API – Application programming interface

ЗМІСТ

ВСТУП

Цифровий світ невпинно розвивається з кожним роком на протязі останнього десятиліття. Завдяки появі телефонів, персональних комп'ютерів та інших пристроїв, підтримуючих інтернет, питання цифрової трансформації набуло значного відтінку в житті людини та в роботі державних установ [2]. В нашій країні був започаткований дуже популярний цифровий проект - Дія, яким користуються більше як 14 млн українців. У застосунку є різні документи та послуги, якими можна скористатися будь-де та будь-коли. Від подачі річної декларації для ФОП до сплати штрафів та генерації COVID-сертифікатів. З початку військового стану в застосунок були додані послуги для перерахування допомоги, а також чат-бот в застосунку Телеграм, який дає можливість повідомити де знаходиться ворог [4].

На сайтах центів надання адміністративних послуг також відбулася цифрова трансформація. Центри надання адміністративних послуг зазвичай надають такий перелік послуг:

– реєстрація / зняття з реєстрації проживання фізичної особи;

– нотаріальні послуги (посвідчення/скасування заповіту);

– Державна реєстрація/скасування державної реєстрації права власності на нерухоме майно;

– реєстрація, скасування реєстрації бізнесу (фізичні та юридичні особи);

– земельні питання (надання відомостей з Державного земельного кадастру, оренда земляних ділянок, права/ припинення прав власності на земельну ділянку);

– державна реєстрація народження дитини, видача посвідчень багатодітних батьків;

– допомога для осіб, постраждалих під час проведення антитерористичної операції та яким встановлено статус учасника бойових дій або особи з інвалідністю внаслідок війни, або учасника війни [10].

Цифровізація адміністративних послуг являє собою адміністративні послуги, які надаються заявникам в електронній формі за допомогою інтернет-технологій. Отримати електронні адміністративні послуги можливо через офіційні вебсайти ЦНАПів у розділах «Реєстр послуг», де наведений вичерпний перелік

адміністративних послуг, які можна отримати онлайн. Потрібно звернути увагу на те, що процес отримання електронних адміністративних послуг через вебсайти ЦНАПів відбувається за допомогою використання кваліфікованого електронного підпису [5].

Один із варіантів цифровізації адміністративних послуг полягає у використанні чат-ботів. Чат-боти є загальнодоступними та використовуються в будь-яких месенджерах чи на будь-яких сайтах. Вони використовуються для різних цілей, допомагають вирішити проблему просто, хоча раніше для цього потребувалося більше часу.

Наразі чат-боти працюють на основі механізмів, керованих правилами, або механізмів штучного інтелекту (AI), які взаємодіють з користувачами переважно через текстовий інтерфейс.

З появою голосових технологій, такі компанії як Google, Apple і Amazon представили голосових асистентів. Apple випустила Siri, яка використовується на iPhone, iPad і macOS. Google запустив Google Home, а Amazon запустив Alexa, які фізично є пристроями для вашого дому чи офісу, які можуть допомогти вам із різними завданнями.

Технологія, що лежить в основі чат-ботів, базується на технології, подібній до голосових помічників. Усі голосові системи мають додаткову складність перетворення мови на текст для будь-якої комп'ютерної програми для роботи [28].

Чат-боти почали масово розповсюджуватися в Україні після того, як Президент України підписав закон про Національну програму інформатизації. Цей закон передбачає створення та функціонування Єдиної інформаційної системи обліку Національної програми інформатизації (ЄІСОНПІ). Вона призначена для формування, контролю, моніторингу, обробки (приймання, обліку, опрацювання) та зберігання програм, завдань, проєктів і робіт Національної програми інформатизації та матеріалів до них. Документ розширює спрямування програми на вирішення низки основних завдань. Зокрема, щодо забезпечення розвитку інформаційного суспільства, застосування інформаційних і цифрових технологій у

державному управлінні та суспільно-економічних відносинах, подолання цифрової нерівності.

Крім того, сприяє безпеці інформаційної діяльності та кіберзахисту, шляхом побудови, розвитку, інтеграції та використання сучасних інформаційних систем, комунікаційних мереж, інформаційних ресурсів й інформаційних технологій. Закон сприяє інтеграції України у світовий інформаційний простір. За допомогою цього закону стали впроваджуватися чат-боти не тільки в бізнес, а й для нашої країни. Наприклад, за період воєнного стану стали активно розроблятися чат-боти для перегляду повітряних тривог, чат-бот, за допомогою якого можна надавати інформацію про переміщення диверсантів та військової ворожої техніки. Також стали актуальними та масово поширилися чат-боти для сайтів надання адміністративних послуг та інших міжнародних установах [3].

Ця тема є дуже **актуальною** та знаковою у сучасну цифрову епоху, оскільки зростаючий попит на адміністративні послуги в поєднанні з потребою в ефективному управлінні чергами створює нагальну потребу в автоматизованих рішеннях, які можуть покращити надання послуг і покращити загальну взаємодію з користувачем.

Тема дипломної роботи: «Алгоритм та структура чат-бота для електронної черги Центру надання адміністративних послуг»

Мета: розробка чат-бота для електронної черги Центру надання адміністративних послуг

Об'єкт дослідження: особливості обміну інформацією в чат-ботах

Предмет дослідження: алгоритм та структура чат-бота

Завдання дослідження:

1) проаналізувати стан розроблення проблеми;
2) проаналізувати основні методи (технології) передачі даних у мережі Інтернет, а також існуючі програмні рішення щодо створення чат-ботів;
3) розробити загальну концепцію чат-бота;
4) розробити модель і алгоритми обміну даними у чат-боті;

5) розробити інформаційно-логічну модель чат-бота;
6) розробити алгоритм і описати технологію використання чат-бота у роботі Центру надання адміністративних послуг.

Методи дослідження: Задля теоретичної основи використовується теоретичний аналіз, тобто досліджується огляд наявної літератури та досліджень, проведених відомими науковцями у сфері розробки чат-ботів та оцінки систем надання адміністративних послуг. Для практичного вирішення поставлених задач використовувалися такі методи:

- теоретичний аналіз (розділ 1)
- математичне моделювання (розділ 2)
- методи теорії баз даних (питання 3.1)
- алгоритмічне програмування (питання 3.2)

Практичне значення отриманих результатів. Отримані результати мають практичне значення для ефективного управління адміністративними процесами. Практичний сенс отриманих результатів щодо алгоритму та структури чат-бота для електронної черги полягає в підвищенні адміністративної ефективності, розширеному користувальницькому досвіді, оптимізації ресурсів, доступності 24/7 та прийнятті рішень на основі даних. Застосування цієї моделі чат-бота сприяє ефективному управлінню електронною чергою, що призводить до покращення адміністративних процесів, підвищення рівня задоволеності клієнтів та оптимізації надання послуг у Центрі надання адміністративних послуг.

Структура та обсяг випускної кваліфікаційної роботи. Випускна кваліфікаційна робота складається із вступу, трьох розділів, висновків, списку використаних джерел із 39 найменувань, додатків і містить 58 сторінок основного тексту, 13 рисунків і 1 таблицю.

РОЗДІЛ 1.
АНАЛІТИЧНЕ ДОСЛІДЖЕННЯ СПЕЦИФІКИ ЗАСТОСУВАННЯ ЧАТ-БОТІВ ДЛЯ ОБМІНУ ДАНИМИ МІЖ КОРИСТУВАЧАМИ

1.1. Аналіз стану проблеми

Незважаючи на те, що чат-бот, судячи з слова, здається нещодавнім, він існує відтоді, коли люди почали взаємодіяти з комп'ютерами. Перший в історії чат-бот був представлений ще до того, як був розроблений перший персональний комп'ютер. Його звали Елізою і цей чат-бот був розроблений в Лабораторії штучного інтелекту Массачусетського технологічного інституту Джозефом Вейзенбаумом у 1966 році.

Еліза виконувала функції психотерапевта для людей. Вона перевіряла ключові слова у введених користувачами повідомленнях і запустила правила перетворення виходу. Ця конкретна методологія генерації відповідей все ще широко використовується при створенні чат-ботів. Після Елізи почав функціонувати ще один чат-бот під ім'ям Паррі, який був написана психіатром Кеннетом Колбі, який тоді працював у Стенфордському університеті, намагаючись симулювати людину з параноїдною шизофренією. A.L.I.C.E, або просто Alicebot, був розроблений Річардом Воллесом у 1995 і був натхненний Елізою.

У першому десятилітті 21 століття SmarterChild був створений ActiveBuddy. Це була перша спроба створити чат-бота, який міг би надати користувачу більш корисну інформацію, таку як інформація про акції, спортивні результати, цитати з фільмів та багато іншого. Пізніше його придбала Microsoft у 2007 році. SmarterChild є попередником Siri від Apple і S Voice від Samsung.

Siri — це інтелектуальний особистий помічник, розроблений SRI і пізніше прийнятий Apple у свою iOS 5 для iPhone. Це було невід'ємною частиною екосистеми iOS. Siri дозволяє користувачам брати участь у випадкових розмовах під час надання корисної інформації про погоду, акції та квитки в кіно. Технічні

гіганти Samsung і Google, слідуючи за Apple, розробили свої власні помічники AI, S Voice і Google Allo відповідно.

Є також голосові домашні помічники, такі як Amazon Alexa та Google Home, які є іншим представником чат-ботів [30].

Підсумовуючи, концепція чат-ботів існує з 1960-х років, але лише наприкінці 1990-х і на початку 2000-х років технологія просунулася достатньо для створення практичних додатків для чат-ботів. Відтоді чат-боти стають дедалі популярнішими та вдосконаленими, завдяки поширенню штучного інтелекту та технологій обробки природної мови, які створюють розширені можливості чат-ботів [32].

Згідно ч. 1 ст. 12 Закону України «Про адміністративні послуги» центр надання адміністративних послуг – це постійно діючий робочий орган або структурний підрозділ виконавчого органу міської, селищної ради …, в якому надаються адміністративні послуги через адміністратора шляхом його взаємодії з суб'єктами надання адміністративних послуг [9].

Цифровізація послуг ЦНАПу була поступовим процесом, який розпочався із впровадження систем електронного документообігу та запуску онлайн-порталів різноманітних державних послуг.

У 2016 році ЦНАП запустив систему електронної черги, щоб упорядкувати процес отримання послуг у центрах. Система дозволяє користувачам планувати зустрічі та стежити за своїм місцем у черзі за допомогою мобільного додатку або веб-інтерфейсу. Ці зусилля з оцифрування допомогли скоротити час очікування та підвищити загальну ефективність центрів.

Що стосується розвитку чат-ботів у ЦНАПі, то коли саме центри почали впроваджувати чат-боти, незрозуміло. Однак, враховуючи зростаючу тенденцію впровадження чат-ботів у різних галузях, цілком імовірно, що ЦНАП почав досліджувати рішення чат-ботів останніми роками для подальшого вдосконалення своїх цифрових послуг.

Обмін даними між Центром надання адміністративних послуг і користувачами може здійснюватися за допомогою таких технологій:

- Інформаційно-комунікаційні технології(ІКТ) - це роль уніфікованих технологій та інтеграцію телекомунікацій (телефонних ліній та бездротових з'єднань), комп'ютерів, підпрограмного забезпечення, програмного забезпечення, накопичувальних та аудіовізуальних систем, які дозволяють користувачам створювати, одержувати доступ, зберігати, передавати та змінювати інформацію. Процес надання адміністративних послуг повністю базується на цій технології.

- Мобільні технології. Деякі державні установи спостерігають повсюдне поширення мобільних технологій, їх вплив, а також збільшення кількості користувачів, що здійснюють вхід на веб-сайти з мобільних пристроїв. Ці технології дозволяють користувачам отримувати доступ до широкого спектру послуг і програм, спілкуватися з іншими та отримувати доступ до інформації з будь-якого місця та в будь-який час.

- Технологія електронного обміну даними - EDI (Electronic Data Interchange) - дозволяє передавати інформацію з однієї комп’ютерної мережі в іншу. EDI дозволяє підприємствам обмінюватися такими документами, як замовлення на купівлю, рахунки-фактури та повідомлення про доставку, у швидкий, точний та економічно ефективний спосіб без необхідності паперових документів або ручного введення даних.

Технологія EDI використовує стандартизований набір форматів повідомлень і протоколів зв’язку, щоб забезпечити безперебійний обмін даними між різними комп’ютерними системами .

Класичне визначення чат-бота – це комп’ютерна програма, яка обробляє природну мову введення від користувача та генерує розумні та відносні відповіді, які потім надсилаються назад користувачу [7].

Чат-боти - це не просто інструменти для доступу користувачів до потрібної інформації (хоча вони кращі, на відміну від FAQ на сторінці), а й можливість просувати свій бренд. Тобто, наприклад, якщо це салон краси, то бот буде ввічливий, акуратний. Якщо це чат-бот для лікарні, то він буде інформативним, корисним [32]. Оскільки для написання чат-ботів використовується штучний інтелект, який не

помиляється і значно економить час людини, у ці технології вкладуються великі гроші і коло поширення цих технологій є досить широким.

За призначенням виділяють такі чат-боти: консультанти, розваги, чат-боти для бізнесу та освіти. В нашому житті вони відіграють значну роль, і на це є кілька причин:

– як комп'ютер, вони прекрасно справляються з обчислювальними операціями і проводять аналіз баз даних за секунди, видаючи оптимальне рішення;

– автоматизує рутинну діяльність;

– як помічник бот мало чим відрізняється від людини, при цьому працює в рази швидше;

– чат-бот швидко шукає. Це дуже важливо при роботі з великими обсягами даних;

– чат-боти швидко удосконалюються і в деяких областях здатні повністю замінити людину.

Існує багато чат-ботів з різними функціями та можливостями. Ось деякі поширені типи чат-ботів і короткий аналіз їхніх сильних і слабких сторін:

Чат-боти на основі правил: Чат-бот на основі правил, також відомий як чат-бот дерева рішень, — це тип чат-бота, який дотримується набору попередньо визначених правил, щоб визначити свої відповіді на введення користувача. Чат-боти на основі правил зазвичай використовують серію операторів if-then, щоб зіставити введені користувачем дані з попередньо визначеною відповіддю. Наприклад, якщо користувач запитує у чат-бота погоду, у чат-бота може бути правило, згідно з яким якщо введення містить слово «погода», він повинен відповісти поточним прогнозом погоди для місця розташування користувача.

Ось деякі особливості, які мають чат-боти на основі правил:

Попередньо визначені правила: чат-боти на основі правил покладаються на заздалегідь визначені правила для визначення своїх відповідей. Ці правила зазвичай створюють розробники чат-бота на основі типів запитань або запитів, які очікується отримати від чат-бота.

Обмежений обсяг: чат-боти на основі правил зазвичай розроблені для обробки певного набору завдань або запитань і не дуже підходять для обробки складних або відкритих запитів.

Проста логіка: логіка чат-ботів на основі правил зазвичай проста, і вони не використовують машинне навчання чи інші передові технології для створення відповідей.

Легко розробити: чат-ботів на основі правил відносно легко розробити, оскільки вони не потребують складних алгоритмів чи складних моделей ШІ.

Обмежена гнучкість: оскільки чат-боти, засновані на правилах, покладаються на заздалегідь визначені правила, вони не дуже гнучкі, і їм може бути важко зрозуміти вхідні дані користувача, які виходять за межі їхніх попередньо визначених правил.

Незважаючи на свої обмеження, чат-боти на основі правил все ще широко використовуються сьогодні, особливо для таких завдань, як підтримка клієнтів, планування та пошук інформації. Вони також часто використовуються як відправна точка для більш просунутої розробки чат-ботів, слугуючи простим прототипом, який можна розширювати та покращувати з часом [23].

Інший тип чат-ботів - це чат-боти на основі пошуку: Чат-боти на основі пошуку, також відомі як чат-боти на основі відповідей, є типом чат-ботів, які генерують відповіді шляхом отримання вже існуючих відповідей із бази даних або бази знань. Вони призначені для зіставлення введених даних користувача з уже існуючими відповідями на основі подібності між введеними даними та доступними відповідями в базі даних.

Ось деякі з ключових функцій пошукових чат-ботів:

База даних відповідей: чат-боти на основі пошуку покладаються на базу даних уже існуючих відповідей, щоб створити свої відповіді. База даних може бути набором попередньо визначених відповідей або текстом, на якому чат-бот пройшов навчання.

Алгоритм відповідності: чат-боти на основі пошуку використовують алгоритм відповідності, щоб порівняти введені користувачем дані з доступними

відповідями в базі даних. Алгоритм може використовувати такі методи, як косинусна подібність або TF-IDF, щоб визначити найкращу відповідну відповідь.

Обмежений контекст: чат-боти на основі пошуку обмежені наявними відповідями в базі даних, що означає, що їм може бути складно обробляти запити, які виходять за межі попередньо визначеного обсягу.

Немає можливості навчання: чат-боти на основі пошуку не мають можливості навчатися на нових введеннях користувачів або оновлювати свої відповіді з часом.

Масштабований: Чат-боти на основі пошуку є масштабованими, оскільки вони можуть швидко отримувати відповіді з великої бази даних уже існуючих відповідей.

Незважаючи на свої обмеження, чат-боти на основі пошуку широко використовуються для таких завдань, як підтримка клієнтів, онлайн-магазини та поширені запитання. Вони часто використовуються в поєднанні з базованими на правилах і генеративними чат-ботами, щоб забезпечити більш комплексний і ефективний досвід чат-ботів [36].

Генеративні чат-боти: Генеративні чат-боти – це тип чат-ботів, які використовують алгоритми машинного навчання, щоб генерувати відповіді на дані користувача. На відміну від чат-ботів на основі правил і пошуку, генеративні чат-боти не покладаються на вже існуючі відповіді чи правила для створення своїх відповідей. Натомість вони використовують методи глибокого навчання, такі як обробка природної мови (NLP) і моделі послідовності до послідовності, щоб генерувати відповіді на основі контексту та вмісту введених користувачем даних.

Ось деякі з ключових особливостей генеративних чат-ботів:

Алгоритми машинного навчання: генеративні чат-боти використовують алгоритми машинного навчання, щоб генерувати відповіді на основі контексту та вмісту введення користувача. Ці алгоритми можуть включати рекурентні нейронні мережі (RNN), трансформатори та інші моделі глибокого навчання.

Здатність до навчання: генеративні чат-боти мають здатність з часом навчатися на нових введеннях користувачів, що означає, що вони можуть покращувати свої відповіді та адаптуватися до нових ситуацій.

Людські розмови: генеративні чат-боти створені для імітації людських розмов і можуть розуміти нюанси мови та контексту, щоб генерувати відповіді, що звучать природно.

Керовані даними: генеративним чат-ботам потрібні великі обсяги даних для навчання моделей машинного навчання, що може бути проблемою для деяких програм.

Складність: генеративні чат-боти є складнішими, ніж чат-боти на основі правил і пошукових запитів, і для роботи вимагають більш досконалих методів машинного навчання та інфраструктури.

Генеративні чат-боти добре підходять для завдань, які потребують природного звучання відповідей з урахуванням контексту, таких як обслуговування клієнтів, персональні помічники та вивчення мови. Вони також використовуються для дослідницьких цілей у сфері обробки природної мови та ШІ.

Помічники зі штучним інтелектом: помічники зі штучним інтелектом, такі як Siri та Alexa, є вдосконаленими чат-ботами, які можуть виконувати широкий спектр завдань, наприклад планувати зустрічі, замовляти їжу та керувати розумними пристроями. Ці чат-боти дуже інтелектуальні та можуть надавати персоналізований досвід на основі даних користувачів. Однак вони вимагають значної обчислювальної потужності і можуть не підходити для невеликих програм [28].

Загалом, тип чат-бота, який найкраще підходить для конкретної програми, залежатиме від її конкретних вимог і кількості доступних ресурсів. Чат-боти на основі правил і пошуку можуть бути корисними для невеликих програм, тоді як генеративні чат-боти та помічники зі штучним інтелектом можуть бути більш придатними для великих і складніших програм.

Чат-боти привернули увагу оточуючих за допомогою 2 додатків: Facebook і Slack. Усі популярні повідомлення платформи надають розробникам величезну

споживчу базу, яку можна використовувати для надання численних сервісів через чат.

Найбільша перевага використання інтерфейсу на основі чату, порівняно з мобільними додатками, сайтами - це те, що мобільні додатки дають можливість користувачу передати свої наміри природньою мовою, якою б вони розмовляли зі своїми друзями. Отже, для того, щоб розробнику запровадити природню мову для чат боту потрібно розібрати текст на зрозумілі частини, які бот повинен зрозуміти та згенерувати відповіді на цей текст. Але це буде займати час для користувача писати багато тексту. Тому платформи обміну повідомленнями представили різні інтерфейси користувача, дозволили дізнаватися певні питання шляхом натискання однієї кнопки.

Основною метою для дипломної роботи є створення чат-боту для електронної черги Центру надання адміністративних послуг. Не так давно люди, які приходили в державні установи стикалися з проблемою в вигляді довгих черг. Навіть було таке, що простоявши в черзі цілий день людина так і не вирішила своє питання. Для цього створили чат-бот, який може відповісти на питання, яке не потребує того, щоб створювати чергу. Ця тема є актуальною, оскільки ми цінуємо свій час і не хочемо витрачати його для того, щоб простояти в черзі. Для цього буде доречно створити чат-бот, де людина зможе побачити вільний час в державній установі та обрати зручний для неї день та час, коли потрібно вирішити питання.

Задля того, щоб усвідомити структуру чат-боту спочатку потрібно усвідомити архітектуру для створення них. В програмуванні найбільш поширеними мовами програмування для написання чат-ботів, які відповідають на запити є: Java, Python, PHP. Зазвичай, чат-боти починають розмову з привітання або запитання, потім користувач обирає варіанти відповідей з пропонованих і в кінці отримує відповідь на певне питання.

Базову структуру чат-боту складають наступні пункти:

1. Обробка природньої мови:(Natural Language Processing (NLP)): це можливість чат-бота розуміти та оброблять людську мову. Обробка природньої

мови допомагає чат-боту визначити намір повідомлення користувача і підібрати найбільш точне вирішення певного питання [17].

2. Розпізнавання намірів: ця функція дозволяє чат-боту зрозуміти, які наміри має користувач спілкуючись з ним. В свою чергу, для розуміння намірів користувача чат-боти використовують алгоритми та моделі машинного навчання.
3. Управління діалогом: це процес керування частинами діалогу між користувачем і чат-ботом. Чат-бот використовує попередньо визначені правила та дерева рішень, щоб визначити, яку відповідь надати на основі повідомлення користувача. Користувач може захотіти змінити чи додати щось до смс. За допомогою цієї функції чат-бот повинен «запам'ятати» намір, який він виявляє на початкових етапах розмови, а потім інтерпретувати пов'язані об'єкти, виявлені в наступних запитах, як належні до цього наміру.
4. База знань: чат-бот може використовувати базу знань для зберігання інформації, яку він може використовувати для відповідей на запити користувачів. База знань чат-бота дозволяє структурувати вміст навколо тем із поширеними запитаннями або набором даних, спрямованих на надання інформації, і враховує різні способи, якими люди запитують ту саму інформацію.
5. Генерація природної мови (Natural Language Generation (NLG)): ця функція дозволяє чат-боту генерувати відповіді, схожі на людську мову. NLG використовує алгоритми та моделі машинного навчання, щоб генерувати відповіді, які є природними та відповідають контексту розмови. З NLG ми беремо структуровані дані з серверних і кінцевих автоматів і перетворюємо їх на неструктуровані дані. Розмовний вихід людською мовою [17].
6. Розпізнавання контексту: ця функція дозволяє чат-боту розуміти контекст розмови та надавати відповідні відповіді. Наприклад, якщо користувач задає одне й те саме запитання двічі, чат-бот повинен мати можливість зрозуміти, що контекст однаковий, і надати однакову відповідь. Якщо користувач буде далі продовжувати діалог за цією темою, то чат-бот буде надавати інформацію, пов'язану з однією темою.

7. Обробка помилок: чат-боти повинні вміти витончено обробляти помилки та несподівані ситуації. Тобто, якщо користувач некоректно спілкується з ботом, задає питання, на які чат-бот не може надати відповідь, або якісь технічні проблеми з ботом, то чат-бот повинен коректно обробити помилки
8. Інтеграція: чат-ботам часто потрібно інтегруватися з іншими системами, такими як бази даних або API, щоб надавати найкращі відповіді. Ця інтеграція може допомогти чат-боту надавати більш точну та актуальну інформацію.

Задля того, щоб підключити чат-бота до веб-сайту, зазвичай потрібно створити кінцеву точку API на сервері чат-бота, з яким веб-сайт може спілкуватися. Кінцева точка API – це URL-адреса, на яку веб-сайт може надсилати запити для взаємодії з чат-ботом. Коли користувач надсилає повідомлення чат-боту через веб-сайт, веб-сайт надсилає запит HTTP POST до кінцевої точки API на сервері чат-бота. Запит POST містить повідомлення користувача в тілі запиту.

Сервер чат-бота отримує запит POST, витягує повідомлення користувача з тіла запиту та передає його чат-боту для обробки. Чат-бот генерує відповідь на повідомлення користувача, яку сервер чат-бота надсилає назад на веб-сайт у тілі відповіді HTTP. Веб-сайт отримує відповідь чат-бота та відображає її користувачеві в інтерфейсі чату. Цей процес триває, коли користувач надсилає додаткові повідомлення, а чат-бот генерує відповіді на ці повідомлення. Специфіка взаємодії сервера чат-бота та веб-сайту може відрізнятися залежно від архітектури чат-бота та веб-сайту. Наприклад, сервер чат-бота може використовувати такий веб-фреймворк, як Flask або Django, для реалізації кінцевої точки API, або він може використовувати протокол обміну повідомленнями, такий як WebSocket або MQTT, для зв’язку з веб-сайтом. Подібним чином веб-сайт може використовувати віджет чату або спеціальний інтерфейс для відображення відповідей чат-бота користувачу [12].

1.2. **Особливості обміну даними в мережі Інтернет**

В даний час найбільшою і найпопулярнішою в світі являється глобальна мережа Internet (від лат. Inter – між. і net - мережа).

Internet – це глобальна інформаційна мережа, яка об'єднує велику кількість регіональних мереж і водночас мільйони комп'ютерів у всіх кінцях планети з метою обміну даними та доступу до інформаційних і технологічних ресурсів.

Інтернет надає платформу для обміну даними між комп'ютерами, серверами та іншими пристроями та розроблений як надійний із резервною інфраструктурою та механізмами відновлення після збоїв, щоб гарантувати постійну доступність даних.

На сьогодення комп'ютерні системи стали все більше взаємопов'язані. Технологія, що лежить в основі мереж обміну даними, розвинулася до точки, коли взаємозв'язок системи вже є нормою. Сучасні мережі більш здатні обробляти великі обсяги даних на найвищих швидкостях. Однак час від часу можливі збої та простої, тому важливо мати плани резервного копіювання [8].

Обмін даних - це процес спільного використання або обміну даними між двома чи більше комп'ютерними системами. Тобто, незалежно від будь-яких обставин, між мережевими системами має бути встановлене з'єднання, яке має бути заздалегіть продумане.

Деякі з ключових особливостей обміну даними в Інтернеті включають:

1. Доступність
2. Швидкість
3. Безпека
4. Надійність
5. Стандарти
6. Маштабованість

Зазвичай, обмін даними здійснюється між різними, зацікавленими в цьому сторонами. Обмін даними надає доступ до точок даних з усього світу для стимулювання маркетингової діяльності та відображення реклами на основі отриманих даних. Тобто, можна взяти застарілі дані, чим користувач цікавився раніше і використовувати цю інформацію для проведення маркетингових компаній. Глобальний

обмін даними Lotame (LDX) є одним із наймасштабніших засобів обміну даними, що забезпечує миттєвий доступ до тисяч сегментів аудиторії, створених із незліченних джерел даних з усього світу.

Для обміну даними в інтернеті потрібно мати достатню швидкість, щоб передавати дані на ваш пристрій з веб-сервера через маршрутизатор. Мегабіт на секунду (Мбіт/с) є стандартним показником швидкості Інтернету. Кількість Мбіт/с вимірює швидкість, з якою інформація завантажується/завантажується з Інтернету на різні пристрої.

Інтернет дозволяє обмінюватися даними на високій швидкості, що дає змогу швидко передавати великі обсяги даних. Це важливо для додатків, які потребують обміну даними в реальному часі, таких як відеоконференції та онлайн-ігри [26].

Необхідна швидкість Інтернету для чат-бота залежатиме від низки факторів, таких як складність чат-бота, кількість користувачів і тип даних, якими обмінюються. Загалом чат-ботам не потрібна висока швидкість Інтернету, оскільки вони зазвичай обмінюються невеликими обсягами даних.

Для чат-бота на основі правил або пошуку, який обмінюється простими текстовими повідомленнями, мінімальна швидкість Інтернету в 1 Мбіт/с має бути достатньою для роботи з великою кількістю користувачів. Однак, якщо чат-бот призначений для обміну мультимедійними файлами, такими як зображення чи відео, може знадобитися швидше з'єднання з Інтернетом, щоб забезпечити швидку та безперебійну передачу даних [16].

Безпечний обмін даними надзвичайно важливий для захисту конфіденційної інформації від несанкціонованого доступу, розголошення або зміни. Розглянемо різноманітні протоколи безпеки, які в основному може використовувати чат-бот, щоб забезпечити безпечне та надійне спілкування між ботом і користувачем. Нижче наведено деякі з найпоширеніших протоколів безпеки для чат-ботів:

SSL/TLS: чат-боти, розміщені на веб-сайтах або в веб-додатках, можуть використовувати шифрування SSL/TLS для захисту зв'язку між ботом і користувачем. SSL/TLS шифрує дані, які передаються між користувачем і чат-

ботом, гарантуючи, що зв'язок безпечний і не може бути перехоплений або змінений неавторизованими третіми сторонами [19].

OAuth: OAuth – це протокол авторизації, який використовується для забезпечення безпечного доступу до ресурсів в Інтернеті. Чат-боти, які потребують автентифікації користувача, можуть використовувати OAuth для автентифікації користувача та безпечного доступу до ресурсів користувача [21].

Веб-токени JSON (JWT): JWT — це безпечний метод передачі інформації між сторонами як об'єкт JSON. Чат-боти можуть використовувати JWT для безпечної передачі інформації між користувачем і ботом, гарантуючи автентифікацію та безпеку зв'язку [15].

Двофакторна автентифікація (2FA): чат-боти можуть використовувати 2FA, щоб забезпечити додатковий рівень безпеки для автентифікації користувачів. За допомогою 2FA користувачі повинні надати дві форми ідентифікації, перш ніж вони зможуть отримати доступ до чат-бота, гарантуючи, що лише авторизовані користувачі зможуть отримати доступ до бота [1].

Вибір протоколу безпеки залежить від конкретного випадку використання та необхідного рівня безпеки.

Технологія шифрування відіграє вирішальну роль у забезпеченні безпечного спілкування між чат-ботами та користувачами. Технології шифрування є ключовим компонентом безпечного спілкування, і є кілька технологій шифрування, які можуть використовуватися чат-ботами для забезпечення безпечного спілкування з користувачами. Найбільш поширеними технологіями шифрування, які використовують чат-боти є:

Шифрування SSL/TLS;

Наскрізне шифрування: наскрізне шифрування (E2EE) — це метод шифрування зв'язку між двома сторонами. Чат-боти можуть використовувати E2EE для шифрування повідомлень і гарантувати, що лише призначений одержувач зможе прочитати повідомлення.

Інфраструктура відкритих ключів (PKI): PKI — це система технологій і

процедур, які використовують цифрові сертифікати для перевірки ідентичності користувачів і забезпечення безпечного зв'язку. Чат-боти можуть використовувати PKI для встановлення безпечного каналу зв'язку з користувачем.

Алгоритми дайджесту повідомлень (MD): алгоритми MD використовуються для шифрування повідомлень і перевірки їх цілісності. Чат-боти можуть використовувати алгоритми MD для шифрування повідомлень і гарантувати, що вони не були підроблені під час передачі.

Розширений стандарт шифрування (AES): AES — це широко поширений алгоритм шифрування, який використовується для захисту конфіденційних даних. Чат-боти можуть використовувати AES для шифрування повідомлень і забезпечення їх безпеки під час передачі.

Вибір технології шифрування залежить від виду та призначення чат-боту.

Також, важливо встановити та дотримуватися стандартів для розробки та розгортання чат-ботів, щоб забезпечити їх ефективність, надійність і безпеку. Наведу приклади різних стандартів, яких можна дотримуватися під час розробки та розгортання чат-ботів.

Ось деякі з найпоширеніших стандартів для чат-ботів:

Розуміння природної мови: розуміння природної мови (NLU) ;

Конфіденційність і безпека: чат-боти повинні дотримуватися стандартів і вказівок щодо конфіденційності та безпеки даних, таких як GDPR (Загальний регламент захисту даних) і HIPAA (Закон про перенесення та підзвітність медичного страхування).

Взаємодія з користувачем: чат-боти повинні відповідати стандартам взаємодії з користувачем (UX), щоб гарантувати, що вони прості у використанні та забезпечують позитивний досвід користувача. Це включає впевнення, що відповіді чат-бота є чіткими та легкими для розуміння, а також що чат-бот реагує та не змушує користувачів чекати надто довго.

Доступність: чат-боти повинні відповідати стандартам доступності, щоб гарантувати, що вони доступні для користувачів з обмеженими можливостями. Це включає надання альтернативного тексту для зображень і забезпечення того, що

чат-бот можна використовувати з допоміжними технологіями, такими як програми зчитування з екрана.

Взаємодія: чат-боти повинні дотримуватися стандартів взаємодії, щоб гарантувати, що вони можуть спілкуватися з іншими системами та службами. Це включає використання відкритих стандартів, таких як RESTful API та JSON.

Тестування та забезпечення якості: чат-боти повинні дотримуватися стандартів тестування та забезпечення якості, щоб переконатися, що вони працюють правильно та надають точні відповіді користувачам.

Дотримання цих стандартів може допомогти забезпечити ефективність, зручність і безпеку чат-ботів [34].

Для створення та розгортання чат-ботів розробники покладаються на різноманітні інтернет-протоколи, які дозволяють спілкуватися між ботом і користувачем. Ці протоколи включають HTTP, REST і WebSocket, серед інших.

Ось деякі з найбільш часто використовуваних інтернет-протоколів для чат-ботів:

HTTP і HTTPS: HTTP (протокол передачі гіпертексту) і HTTPS (захищений HTTP) є основними протоколами, які використовуються для обміну даними між веб-серверами та клієнтами. Чат-боти, інтегровані у веб-сайти або веб-додатки, можуть використовувати ці протоколи для зв’язку із сервером.

WebSocket: WebSocket — це протокол, який забезпечує повнодуплексний канал зв’язку через одне з’єднання TCP. Чат-боти можуть використовувати WebSocket для підтримки постійного з’єднання з сервером і отримання оновлень у реальному часі.

MQTT: MQTT (Message Queuing Telemetry Transport) — це легкий протокол обміну повідомленнями, який зазвичай використовується для пристроїв IoT (Інтернет речей). Чат-боти можуть використовувати MQTT для обміну повідомленнями та даними з іншими пристроями та службами.

SMTP і IMAP: SMTP (простий протокол передачі пошти) і IMAP (протокол доступу до повідомлень Інтернету) — це протоколи електронної пошти, які можна використовувати для надсилання й отримання електронних листів. Чат-боти

можуть використовувати ці протоколи для надсилання та отримання електронних листів від імені користувача.

RESTful API: REST (Representational State Transfer) — це архітектурний стиль, який забезпечує стандарт для створення веб-служб. Чат-боти можуть використовувати RESTful API для зв'язку з іншими системами та службами та обміну даними.

GraphQL: GraphQL — це мова запитів і середовище виконання, що забезпечує більш ефективну та гнучку альтернативу RESTful API. Чат-боти можуть використовувати GraphQL для запиту та отримання певних даних із сервера.

Інтернет розроблений таким чином, щоб бути масштабованим, щоб вмістити зростаючу кількість користувачів і пристроїв. Це досягається за допомогою розподілених архітектур, балансування навантаження та інших методів, які гарантують, що обмін даними можна підтримувати, навіть якщо обсяг даних і користувачів зростає.

Чат-боти можна розробляти та розгортати за допомогою різних технологій, фреймворків і мов програмування. Ось деякі з найбільш часто використовуваних технологій для розробки чат-ботів:

Обробка природної мови (NLP)

Машинне навчання: машинне навчання – це технологія, яка дозволяє чат-ботам навчатися та покращувати свої відповіді на основі взаємодії з користувачем. Чат-боти можуть використовувати машинне навчання, щоб ідентифікувати шаблони поведінки користувачів і з часом покращувати їхні відповіді.

Хмарні обчислення: Хмарні обчислення забезпечують інфраструктуру та ресурси, необхідні для роботи чат-ботів у масштабі. Чат-боти можна розміщувати на хмарних платформах, таких як AWS, Microsoft Azure та Google Cloud.

Платформи обміну повідомленнями: платформи обміну повідомленнями, такі як Facebook Messenger, WhatsApp і Slack, надають розробникам чат-ботів інструменти та API, необхідні для створення та розгортання чат-ботів на цих платформах.

Фреймворки веб-розробки: чат-боти, інтегровані у веб-сайти або веб-додатки, можна розробляти за допомогою фреймворків веб-розробки, таких як Django, Flask і Express.

API та SDK: API та SDK надають розробникам чат-ботів інструменти, необхідні для інтеграції чат-ботів з іншими програмами та службами. Приклади API та SDK включають Dialogflow API, Wit.ai API та Microsoft Bot Framework SDK [14].

Обмін даними в Інтернеті стає можливим завдяки використанню різноманітних протоколів, які полегшують зв'язок між різними пристроями та мережами. Найбільш часто використовувані протоколи включають HTTP, TCP/IP, FTP і SMTP, серед інших. Кожен протокол має свої унікальні особливості та переваги, які роблять його придатним для конкретних програм.

Крім того, розробка та розгортання чат-ботів в Інтернеті вимагають використання спеціальних протоколів, які забезпечують спілкування між ботом і його користувачами. Ці протоколи включають серед інших REST, WebSocket і MQTT. Кожен протокол має свої сильні та слабкі сторони, які визначають його придатність для конкретних додатків чат-ботів.

Таким чином, правильний вибір і використання Інтернет-протоколів є критично важливими для забезпечення безпечного та ефективного обміну даними в Інтернеті. Розробники та мережеві адміністратори повинні добре розуміти ці протоколи та їхні функції, щоб забезпечити оптимальну продуктивність і безпеку своїх програм. Оскільки Інтернет продовжує розвиватися, з'являтимуться нові протоколи, тому важливо бути в курсі останніх подій у цій галузі.

Підсумовуючи, обмін даними в Інтернеті змінив спосіб спілкування, співпраці та ведення бізнесу, зробивши його швидшим, ефективнішим і доступнішим для всіх. Обмін даними в Інтернеті передбачає передачу цифрової інформації з одного комп'ютера на інший за допомогою різних протоколів і технологій. Однією з головних особливостей обміну даними в Інтернеті є його децентралізація. Інтернет складається з мережі взаємопов'язаних комп'ютерів і серверів, які спілкуються один з одним для обміну даними. Це означає, що дані

можна надсилати та отримувати з будь-якої точки світу, і немає центральних точок контролю. Оскільки технології продовжують розвиватися, ми можемо очікувати ще більше інноваційних способів обміну даними в Інтернеті.

1.3. Огляд існуючих програмних рішень

Чат-боти стають все більш популярними як інструмент для бізнесу для покращення взаємодії з клієнтами, скорочення часу відповіді та підвищення ефективності. Як наслідок, зараз існує кілька доступних програмних рішень для створення чат-ботів, починаючи від бібліотек і фреймворків з відкритим кодом до хмарних платформ і інтерфейсів перетягування. Ці рішення надають розробникам інструменти та ресурси, необхідні для швидкого та легкого створення чат-ботів, не вимагаючи великих технічних знань чи досвіду.

У цьому підрозділі ми надамо огляд деяких існуючих програмних рішень для створення чат-ботів, включаючи їхні функції, переваги та обмеження.

Одним із популярних варіантів програмних рішень є Dialogflow, хмарна платформа розробки, яка використовує комбінацію машинного навчання та методів NLP для обробки даних користувача, розуміння намірів користувача та створення відповідних відповідей. Раніше відомий як API.ai, Dialogflow був придбаний Google у 2016 році і з тих пір став однією з найпопулярніших платформ для розробки чат-ботів на ринку.

Найкраще підходить для створення Assistant, віртуального друга для смартфонів Android, iOS і Windows Phone. Продукт від Google дозволяє розробникам створювати текстові та голосові розмовні інтерфейси для відповідей на проблеми клієнтів різними мовами.

Dialogflow надає візуальний інтерфейс для створення чат-ботів, тож розробники можуть легко створювати та керувати потоками розмов, визначати наміри та сутності та тестувати свої чат-боти в режимі реального часу. Платформа також взаємодіє з широким спектром платформ обміну повідомленнями, такими як Facebook

Messenger, Slack і Google Assistant, дозволяючи розробникам легко розгортати свої чат-боти в кількох каналах [37].

Іншим варіантом програмних рішень є Microsoft Bot Framework . Це комплексний набір інструментів і служб для створення та розгортання чат-ботів і розмовних агентів у кількох каналах. Широкий набір інструментів розробки, готових шаблонів і інтеграцій спрощує розробникам створення та розгортання чат-ботів на широкому діапазоні каналів і платформ. Платформа також надає розширені можливості ШІ та машинного навчання завдяки інтеграції з Azure Cognitive Services, що дозволяє розробникам легко додавати розширені функції до своїх чат-ботів. Фреймворк включає ряд інструментів розробки, включаючи Bot Builder SDK, Bot Framework Emulator і Bot Framework Composer, а також низку готових шаблонів, конекторів та інтеграцій [35].

Деякі з ключових функцій Microsoft Bot Framework включають:

1. Bot Builder SDK: Bot Builder SDK — це потужний інструмент для створення чат-ботів і розмовних агентів за допомогою широкого діапазону мов програмування, включаючи C#, Node.js і Python .
2. Емулятор Bot Framework: емулятор Bot Framework — це настільна програма, яка дозволяє розробникам тестувати та налагоджувати свої чат-боти перед розгортанням їх у робочих середовищах.
3. Bot Framework Composer: Bot Framework Composer — це інструмент візуальної розробки, який дозволяє розробникам створювати та керувати потоками розмов за допомогою інтерфейсу перетягування.
4. Попередньо створені шаблони: Microsoft Bot Framework містить низку готових шаблонів для створення поширених типів чат-ботів, наприклад ботів служби підтримки клієнтів, ботів електронної комерції тощо.
5. Інтеграція каналів: Bot Framework включає широкий спектр інтеграцій із популярними платформами обміну повідомленнями, такими як Facebook Messenger, Skype і Slack, а також голосовими помічниками, такими як Alexa та Google Assistant.

6. Когнітивні служби: Microsoft Bot Framework інтегрується з Azure Cognitive Services, дозволяючи розробникам додавати розширені функції, такі як розуміння природної мови, аналіз настроїв і розпізнавання мовлення до своїх чат-ботів.

Недоліком Microsoft Bot Framework є те, що деякі користувачі вважають платформу дещо складною та непосильною, особливо для початківців або розробників із обмеженим досвідом створення чат-ботів .

Ще одним програмним рішенням для створення чат-боту є Botpress. Це потужна та гнучка платформа з відкритим вихідним кодом для створення та розгортання чат-ботів і розмовних агентів. Він надає ряд інструментів і функцій для спрощення процесу розробки, включаючи візуальний редактор потоку, потужний механізм NLU та інтеграцію з популярними платформами обміну повідомленнями та API.

Деякі з ключових функцій Botpress включають:

1. Візуальний редактор потоків: Botpress надає візуальний редактор потоків, який дозволяє розробникам створювати та керувати потоками розмов за допомогою інтерфейсу перетягування. Це спрощує створення складних робочих процесів і керування розмовами вашого чат-бота.
2. Розуміння природної мови (NLU): Botpress містить потужний механізм NLU, який дозволяє розробникам додавати можливості обробки природної мови до своїх чат-ботів. Це дозволяє чат-боту розуміти та інтерпретувати введені користувачем дані, що робить його більш розумним і чуйним.
3. Інтеграція: Botpress інтегрується з широким спектром платформ обміну повідомленнями, такими як Facebook Messenger, Telegram і Slack, а також API, такими як Twilio і WhatsApp.
4. Відкритий код: Botpress — це платформа з відкритим кодом, що означає, що розробники можуть отримати доступ до вихідного коду та змінити його відповідно до своїх потреб. Це забезпечує більшу гнучкість і налаштування. Той факт, що він є відкритим кодом, також означає, що розробники можуть налаштувати та розширити платформу відповідно до своїх потреб.

5. Аналітика: Botpress надає аналітику та показники для відстеження ефективності вашого чат-бота та моніторингу залучення користувачів. Ці дані можна використовувати для оптимізації чат-бота та покращення його продуктивності з часом [24].

Tars — це також програмне рішення для побудови чат-бота. Tars — це хмарна платформа чат-ботів, яка дозволяє користувачам створювати та розгортати чат-боти для широкого спектру випадків використання, включаючи підтримку клієнтів, створення потенційних клієнтів і продажі. Він пропонує ряд функцій та інструментів для спрощення процесу розробки чат-бота, включаючи візуальний конструктор перетягування, інтеграцію з популярними платформами обміну повідомленнями та обробку природної мови (NLP) на основі ШІ.

Tars надає візуальний інтерфейс для створення робочих процесів чат-бота та керування ними. Це дозволяє легко розробляти складні потоки розмов без необхідності писати код. Ця платформа використовує NLP на основі штучного інтелекту, щоб розуміти та інтерпретувати введені користувачем дані, дозволяючи йому розумно відповідати на запити та запити користувачів. Tars інтегрується з широким спектром платформ обміну повідомленнями, включаючи Facebook Messenger, WhatsApp і Slack, що дозволяє користувачам розгортати свої чат-боти в кількох каналах. Платформа Tars інтегрується з низкою інструментів і платформ сторонніх розробників, зокрема Zapier, Google Sheets і Mailchimp.

Загалом Tars є потужною та зручною платформою для створення чат-ботів. Його візуальний конструктор спрощує створення інтелектуальних чат-ботів без будь-якого досвіду програмування, а його інтеграція з популярними платформами обміну повідомленнями та інструментами сторонніх розробників спрощує розгортання та керування чат-ботами в кількох каналах. Платформа також пропонує надійні інструменти аналітики та звітності, що дозволяє користувачам оптимізувати свої чат-боти для максимального залучення та конверсій [25].

Після того, як ми розглянули платформи за допомогою яких ми можемо створити чат-бот, далі ми розглянемо який тип чат-боту найкраще підходить для застосування в ЦНАП. Для цього ми склали таблицю 1.3, де дально можна

розглянути типи чат-ботів, їх плюси та мінуси та наскільки вони придатні для застосування в ЦНАП:

Табл. 1.3

«Який тип чат-бота найбільше підходить для ЦНАП»

Тип чат-бота	**Плюси**	**Мінуси**	**Підходить для ЦНАПу?**
Чат-боти на основі правил(rule-based chatbot)	Прості в налаштуванні та обслуговуванні, можуть ефективно обробляти прості запити, економічно ефективні	Обмежена здатність обробляти складні запити чи розмови, вимагає створення великих правил, може швидко застаріти	Так, якщо для обробки ЦНАП потрібен простий чат-бот, рутинні завдання та запити
Спеціальні чат-боти(custom chatbot)	Пристосовані до конкретних бізнес-потреб і вимог користувачів, можуть забезпечити персоналізований і привабливий досвід для користувачів, підвищують ефективність і підтримку клієнтів	Вимагає досвіду програмування, дизайну та НЛП, може бути дорогим в розробці	Так, якщо для ЦНАП потрібен чат-бот, який відображає індивідуальність та відповідає конкретним потребамбізнесу та вимогам користувачів
Чат-боти на основі штучного)	можуть ефективно обробляти	Може бути дорогим у розробці та	Так, якщо ЦНАПу потрібен

Продовження Табл. 1.3

Тип чат-бота	Плюси	Мінуси	Підходить для ЦНАПу?
інтелекту(AI-powered Chatbots)	складні запити та розмови, постійно навчатися на основі взаємодії, можуть надавати персоналізовані відповіді	підтримці, вимагає великих наборів даних і можливостей NLP, може бути складніше підтримувати	чат-бот, який може навчатися на основі взаємодії та надавати персоналізовані відповіді

Проаналізувавши ці типи чат-ботів та зробивши детальний огляд кожного з них, в подальшому для створення чат-боту для запису в чергу в ЦНАП доречно буде розробити Спеціальний чат-бот (custom chatbot), тому що він може використовуватися згідно індивідуальних потреб і надає консультацію в реальному часі.

У цьому підрозділі ми розглянули програмні рішення, які можна застосувати для створення чат-боту. Загалом вибір програмного рішення залежатиме від конкретних потреб і вимог чат-бота ЦНАПу, а також від таких факторів, як бюджет, технічна експертиза та бажані функції та інтеграції. Перш ніж прийняти рішення, важливо оцінити кілька варіантів і порівняти їхні характеристики та ціни. Після вибору відповідного програмного рішення ЦНАП може створити індивідуальний чат-бот, який може покращити взаємодію з користувачем, оптимізувати роботу та надати цінну інформацію про поведінку та вподобання користувачів. Ми також зробили таблицю, в якій оцінили різні типи чат-ботів та обрали найкращий тип.

РОЗДІЛ 2.
РОЗРОБКА МОДЕЛІ ЧАТ-БОТА

2.1. Загальна концепція

Розробка спеціальної моделі чат-бота для Центру надання адміністративних послуг включає низку етапів, зокрема визначення обсягу та призначення чат-бота, проектування потоку розмов, вибір відповідного програмного рішення, а також тестування та доопрацювання чат-бота. Для початку створимо загальну концепцію моделі чат-боту.

Перше, що потрібно зробити — це визначити сферу застосування та мету чат-бота. Оскільки основним завданням для ЦНАП є створення чат боту для запису в чергу, то спеціальний чат-бот буде виконувати саме цю функцію. Основна мета спеціального чат-бота в ЦНАП – покращити взаємодію з користувачем, забезпечивши розмовний та інтуїтивно зрозумілий інтерфейс, який може допомогти користувачам легше та ефективніше орієнтуватися на платформі. Чат-бот може допомогти користувачам записатися в чергу в певний місяць і в певне число. Обсяг спеціального чат-бота в ЦНАПі може змінюватися залежно від конкретних потреб і цілей платформи.

Загалом, спеціальна модель чат-бота для ЦНАП може стати потужним інструментом для покращення взаємодії з користувачем і оптимізації операцій на платформі. Забезпечуючи розмовний та інтуїтивно зрозумілий інтерфейс, чат-бот може допомогти користувачам легше та ефективніше орієнтуватися на платформі, а також надати персональну і швидку допомогу допомогу користувачеві.

Далі в загальну концепцію входить розробка розмовного потоку, тобто визначити яку структуру діалогу буде містити чат-бот.

Проектування розмовного потоку для власної моделі чат-бота для ЦНАП передбачає створення структурованого та інтуїтивно зрозумілого діалогу між користувачем і чат-ботом. Нижче наведено загальний приклад потоку розмов для власної моделі чат-бота для ЦНАП:

Привітання та вступ: Чат-бот вітає користувача та представляється, пояснюючи своє призначення та можливості.

Ідентифікація намірів користувача: Чат-бот запитує користувача, з чим йому потрібна допомога на платформі ЦНАП, запропонує користувачеві кілька варіантів на вибір

Збір інформації: Чат-бот запитує у користувача відповідну інформацію для виконання запиту, наприклад, дату та час, коли він хоче замовити послугу.

Надання послуг або інформації: Чат-бот надає користувачу необхідну послугу або інформацію на основі його введення та переваг

Підтвердження та відгук: Чат-бот підтверджує запит користувача та надає короткий опис послуги або наданої інформації.

До побачення: Чат-бот дякує користувачеві за використання платформи ЦНАП та сервісів чат-бота.

Потік розмов для спеціальної моделі чат-бота для ЦНАП має бути розроблений таким чином, щоб забезпечити безперебійний та інтуїтивно зрозумілий досвід для користувача, а також гарантувати, що чат-бот може ефективно виконувати запити та потреби користувача.

Вибір відповідної мови програмування для спеціальної моделі чат-бота для ЦНАП залежить від різних факторів, таких як складність чат-бота, наявні ресурси та досвід, а також бажані функції та функціональність. Ось кілька мов програмування, які зазвичай використовуються для створення чат-ботів:

Python: Python — це широко поширена мова програмування, яка має багато бібліотек і фреймворків для обробки природної мови та машинного навчання. Він відомий своєю простотою та легкістю використання, що робить його популярним вибором для розробки чат-ботів. Python має кілька бібліотек, таких як NLTK, spaCy і TextBlob, які можна використовувати для аналізу тексту та аналізу настроїв.

JavaScript: JavaScript — популярна мова програмування, яка часто використовується для створення чат-ботів для веб- і мобільних додатків. Він має кілька фреймворків, таких як Node.js, React.js і Angular.js, які можна використовувати для

створення чат-ботів. Крім того, доступно кілька бібліотек для обробки природної мови та машинного навчання [38].

C#: C# — це мова програмування, яка часто використовується для створення чат-ботів на Microsoft Bot Framework. Він має кілька бібліотек, як-от Microsoft Bot Builder SDK, які можна використовувати для створення чат-ботів із такими функціями, як обробка природної мови, аналіз настроїв тощо [11].

Java: Java — це широко поширена мова програмування, яка має кілька бібліотек і фреймворків, таких як Stanford CoreNLP, Apache OpenNLP і LingPipe, які можна використовувати для обробки природної мови та машинного навчання. Java часто використовується для створення чат-ботів для корпоративних програм [13].

Зрештою, для розробки чат-боту я обрала Python. Приведемо приклад кроків додання на вебсайт чат-боту, написаного на мові програмування Python:

1. Вибір веб-фреймворку. Першим кроком є вибір веб-фреймворку, який ви хочете використовувати для створення свого веб-сайту. До популярних веб-фреймворків на Python належать Flask, Django та Pyramid.
2. Створення чат-бота: для створення чат-бота можна використовувати бібліотеку розробки чат-бота на Python, наприклад ChatterBot. Вам потрібно буде спроектувати потік розмови, створити наміри та налаштувати відповіді. Етапи створення чат-бота залежатимуть від обраної вами бібліотеки розробки чат-бота.
3. Створення веб-програми: використовуючи обрану вами веб-платформу, створіть веб-програму, яка зможе спілкуватися з чат-ботом. Вам потрібно буде налаштувати кінцеву точку API, щоб отримувати та надсилати повідомлення чат-боту.
4. Потрібно інтегрувати чат-бот у веб-програму: інтегруйте код чат-бота з веб-програмою, викликавши API чат-бота з кінцевої точки веб-програми. Можна використовувати бібліотеку запитів у Python, щоб надсилати HTTP запити до API чат-бота.

5. Розгорніть веб-додаток: розгорніть веб-додаток на веб-сервері або хмарній платформі, наприклад Heroku або AWS Elastic Beanstalk, де до нього зможуть отримати доступ користувачі.
6. Підключення веб-програми до свого веб-сайту: щоб підключити веб-програму до свого веб-сайту, буде доцільне використання iframe або можна вставити веб-програму на сторінку свого веб-сайту. Ви також можете створити спеціальний віджет або вікно чату, яке інтегрує веб-програму.
7. Перевірте чат-бота: після підключення веб-програми до веб-сайту вам потрібно протестувати чат-бота, щоб переконатися, що він працює правильно. Ви можете зробити це, відкривши веб-сайт і протестувавши потік розмови чат-бота.

Важливо переконатися, що бібліотека розробки чат-бота та обраний веб-фреймворк сумісні між собою.

Також, важливо ретельно протестувати чат-бота та вдосконалити його на основі відгуків користувачів і показників ефективності.

Отже, ми розглянули загальну концепцію та проаналізували етапи для створення спеціальної моделі чат-боту. Дотримуючись цих кроків, ЦНАП може створити чат-бота, який відповідає його конкретним потребам і цілям. Загалом розробка спеціальної моделі чат-бота для ЦНАП може допомогти покращити взаємодію з користувачем, пришвидчити запис в чергу.

2.2. Моделі обміну даними

При розробці спеціальної моделі чат-бота для ЦНАП моделі обміну даними є важливими для забезпечення зв’язку між чат-ботом і системами ЦНАП. Ось деякі моделі обміну даними, які можна використовувати:

REST API — широко використовувана модель обміну даними для веб-додатків. REST означає Representational State Transfer, і це архітектурний стиль, який використовує протоколи HTTP для взаємодії з ресурсами, такими як бази даних або інші веб-служби. Це дозволяє системам TsNAP надсилати та отримувати дані в

стандартизованому форматі, такому як JSON або XML, через кінцеву точку URL. Чат-бот може надсилати API-запити до систем ЦНАПу для отримання або оновлення даних.

Ось як може працювати модель обміну даними REST API для чат-бота в ЦНАП:

1. Чат-бот надсилає HTTP-запит до кінцевої точки TsNAP із параметрами для отримання або оновлення певних даних.
2. Системи ЦНАП приймають запит, обробляють дані та надсилають відповідь HTTP, як правило, у форматі JSON або XML.
3. Чат-бот аналізує відповідь і використовує її, щоб надати відповідну відповідь користувачеві.

Однією з переваг використання REST API як моделі обміну даними є те, що це стандартизований протокол, який широко використовується у веб-сервісах, що дозволяє легко інтегрувати чат-бот із системами ЦНАП. Крім того, REST API є масштабованим і може обробляти велику кількість запитів, що робить його придатним для чат-бота, який, як очікується, обробляє велику кількість користувачів [22].

Однак розробка REST API може потребувати значних ресурсів як з точки зору часу розробки, так і з точки зору інфраструктури. Також, це може вимагати додаткових заходів безпеки для запобігання несанкціонованому доступу до систем ЦНАПу.

Webhooks — ще одна широко використовувана модель обміну даними для чат-ботів, яка передбачає спілкування в реальному часі між чат-ботом і системами ЦНАП. За допомогою Webhooks системи системи ЦНАП можуть надсилати повідомлення чат-боту, щойно відбуваються нові події, без необхідності чат-боту постійно опитувати нові дані. Чат-бот може обробляти дані та відповідати відповідним чином.

Основний принцип роботи Webhooks для чат-бота в ЦНАП містить таку послідовність:

1. Системи ЦНАП надсилають вебхук до кінцевої точки URL-адреси, вказаної чат-ботом, із інформацією про подію, наприклад розміщення нового замовлення.
2. Чат-бот отримує вебхук і обробляє дані, відповідно генеруючи відповідь користувачеві.
3. Чат-бот надсилає відповідь HTTP назад на вебхук, щоб підтвердити, що він отримав і обробив дані.

Основною перевагою в використанні Webhooks як моделі обміну даними є те, що він дозволяє спілкуватися в реальному часі, тому чат-бот може реагувати на події, щойно вони відбуваються. Це може бути особливо корисно для завдань, що потребують часу, наприклад, підтримка клієнтів або обробка замовлень.

Але, недоліками є те, що розробка моделі обміну даними Webhooks може вимагати додаткових налаштувань і конфігурацій порівняно з іншими моделями обміну даними, і може бути складнішим оброблення помилок або винятків, які можуть виникнути під час обміну даними. Крім того, чат-боту може знадобитися застосувати додаткові заходи безпеки, такі як перевірка автентичності вхідних вебхуків, щоб запобігти несанкціонованому доступу.

Ще одна модель обміну даними, яка зазвичай використовується для чат-ботів для полегшення зв’язку між чат-ботом і зовнішніми системами, такими як ЦНАП є API обміну повідомленнями. API обміну повідомленнями, такі як API Facebook Messenger, API WhatsApp Business або API Telegram Bot, дозволяють чат-боту надсилати та отримувати повідомлення через платформу обміну повідомленнями. Системи ЦНАПу можуть інтегруватися з платформою обміну повідомленнями, щоб надсилати та отримувати повідомлення за допомогою чат-бота.

Ось основне представлення роботи моделі обміну даними Messaging API для чат-бота в ЦНАП:

1. Розробник чат-бота реєструє чат-бота на платформі обміну повідомленнями або додатку для обміну повідомленнями, який використовується системами ЦНАПу.

2. Коли системам ЦНАП потрібно надіслати повідомлення чат-боту, вони надсилають повідомлення на платформу обміну повідомленнями або додаток для обміну повідомленнями, який пересилає його чат-боту.
3. Чат-бот отримує повідомлення та обробляє його, відповідно генеруючи відповідь користувачеві.
4. Чат-бот надсилає повідомлення-відповідь назад на платформу або додаток для обміну повідомленнями, які пересилають його до систем ЦНАПу.

API обміну повідомленнями є доступною платформою обміну повідомленнями та програми обміну повідомленнями, що робить їх доступними для великої бази користувачів. Більшість користувачів вже знайомі з платформами та програмами обміну повідомленнями, що полегшує їм взаємодію з чат-ботами через це середовище. API обміну повідомленнями можна використовувати для створення чат-ботів, які працюють на кількох платформах і програмах обміну повідомленнями.

Головна функція для цієї моделі обміну даними є вкладення медіафайлів, які можуть використовуватися чат-ботами для покращення взаємодії з користувачем.

Однак під час використання API обміну повідомленнями слід враховувати деякі обмеження:

Складність інтеграції: розробники чат-ботів повинні інтегруватися зі сторонніми службами обміну повідомленнями, що може ускладнити розробку.

Обмеження платформи: кожна платформа обміну повідомленнями або програма може мати власні обмеження та вимоги, як-от форматування повідомлень або автентифікація, яких необхідно дотримуватися.

Обмежена функціональність: платформи та програми обміну повідомленнями можуть обмежувати типи повідомлень, які можна надсилати та отримувати чат-ботами, що може обмежувати функціональність чат-бота.

Інтеграція бази даних є ключовим аспектом розробки спеціального чат-бота, оскільки вона дозволяє чат-боту зберігати та отримувати дані, що стосуються взаємодії користувачів. База даних може забезпечити централізоване розташування для зберігання та доступу до налаштувань користувача, даних сеансу та іншої

інформації, яку можна використовувати для персоналізації взаємодії з користувачем і покращення продуктивності чат-бота.

Під час інтеграції бази даних у власну модель чат-бота для ЦНАП важливо вибрати систему баз даних, яка відповідає конкретним вимогам проекту, таким як масштабованість, продуктивність і безпека даних. Деякі популярні системи баз даних для розробки чат-ботів включають:

1. MySQL. Це популярна система керування реляційними базами даних із відкритим вихідним кодом, яка широко використовується для веб-додатків, зокрема для чат-ботів. Він відомий своєю масштабованістю, надійністю та продуктивністю, що робить його підходящим вибором для розробки індивідуальних чат-ботів у ЦНАПі. MySQL розроблено для високої масштабованості, що дозволяє чат-ботам швидко й ефективно зберігати та отримувати великі обсяги даних. Він також надає розширені функції для керування даними, наприклад підтримку транзакцій, збережених процедур і тригерів. Ці функції можуть допомогти забезпечити цілісність даних чат-бота та покращити продуктивність. Крім того, MySQL є високонадійним із вбудованими механізмами резервного копіювання та відновлення даних. Це може допомогти запобігти втраті даних у разі збою системи або іншої несподіваної події. MySQL також легко інтегрувати з мовами програмування, які зазвичай використовуються для розробки чат-ботів, такими як Python, за допомогою бібліотек і API.
2. MongoDB — це популярна документно-орієнтована система керування базами даних NoSQL, яка часто використовується для веб-додатків, зокрема чат-ботів. Він відомий своєю гнучкістю, масштабованістю та здатністю обробляти великі обсяги неструктурованих даних, що робить його підходящим вибором для розробки власних чат-ботів у ЦНАП.

Однією з головних переваг MongoDB є його здатність обробляти неструктуровані дані, такі як текстові повідомлення, зображення та інший мультимедійний вміст, який важко зберігати в традиційних реляційних базах даних. MongoDB зберігає дані в гнучких документах, схожих на JSON, які можна легко змінювати та

оновлювати за потреби. MongoDB також має високу масштабованість із вбудованими механізмами шардингу та реплікації. Це дозволяє чат-ботам обробляти великі обсяги даних і трафіку, зберігаючи високу продуктивність і доступність.

3. PostgreSQL- це система керування об'єктно-реляційною базою даних з відкритим вихідним кодом, яка пропонує розширені функції для контролю цілісності даних і паралельності. PostgreSQL відомий своєю продуктивністю та масштабованістю, що робить його популярним вибором для розробки чат-ботів. Однією з головних переваг PostgreSQL є його підтримка розширених функцій SQL, таких як розширені операції з'єднання, підзапити та віконні функції. Це полегшує написання складних запитів, які можна використовувати для отримання та аналізу даних із розмов чат-бота.

PostgreSQL також має широкі можливості налаштування з широким набором розширень і плагінів, які можна використовувати для додавання додаткових функцій і оптимізації продуктивності. Це може бути особливо корисним при розробці чат-ботів, яким потрібно обробляти великі обсяги даних і трафіку. Крім того, PostgreSQL є високонадійним із вбудованими механізмами резервного копіювання та реплікації. Це гарантує, що дані чат-бота завжди доступні, навіть у разі апаратних збоїв чи інших проблем.

Під час інтеграції бази даних у власну модель чат-бота для ЦНАП важливо розробити схему бази даних для ефективного зберігання та отримання необхідних даних. Чат-бот також має включати обробку помилок і перевірку даних, щоб забезпечити цілісність бази даних і запобігти потенційній вразливості безпеки.

Підсумовуючи, існує кілька доступних моделей обміну даними для розробки власних чат-ботів у Python для ЦНАП, включаючи REST API, веб-хуки та API обміну повідомленнями. Кожна з цих моделей має свої переваги та недоліки, і вибір залежатиме від конкретних потреб та вимог чат-бота. Коли справа доходить до інтеграції бази даних, існує кілька доступних варіантів, включаючи MySQL, MongoDB і PostgreSQL. Кожна з цих баз даних має свої сильні та слабкі сторони.

Вибір моделі обміну даними залежить від конкретних вимог ЦНАПу та чат-бота. Важливо вибрати модель обміну даними, яка є надійною, ефективною та

безпечною. Крім того, важливо враховувати масштабованість моделі обміну даними, щоб переконатися, що вона може обробляти великий обсяг запитів, оскільки чат-бот стає все більш популярним.

2.3. Алгоритми обміну даними

Розробка власної моделі чат-бота для ЦНАП також передбачає створення алгоритмів обміну даними, які дозволяють чат-боту отримувати та обробляти дані з платформи та баз даних ЦНАП.

Алгоритми обробки природної мови (NLP) є основою розробки інтелектуальних чат-ботів, які можуть розуміти та генерувати людську мову. Ці алгоритми дозволяють чат-ботам обробляти та інтерпретувати текстові дані, витягувати значення та генерувати відповіді. Існують деякі алгоритми НЛП, які найчастіше використовуються в розробці чат-ботів. По-перше, це токенізація (процес поділу тексту на окремі токени, наприклад слова чи підслова). Він формує основу завдань НЛП, розділяючи текст на значущі одиниці для подальшого аналізу. Такожм є теги POS, які призначають граматичні теги кожному токену в реченні, наприклад іменникам, дієсловам, прикметникам тощо. Це допомагає зрозуміти синтаксичну структуру речень і може допомогти в таких завданнях, як розпізнавання іменованих об’єктів або розбір речення. Алгоритми розпізнавання іменованих об’єктів NER ідентифікують і класифікують іменовані об’єкти в тексті, такі як імена осіб, організацій, місця розташування, дати тощо. Це може бути корисним для отримання релевантної інформації з введених користувачем даних або надання відповідей з урахуванням контексту. Алгоритми розпізнавання наміру визначають намір або мету введення користувача. Вони прив’язують запити користувачів до конкретних дій або відповідей, що дозволяє чат-боту розуміти наміри користувача та надавати відповідну інформацію або виконувати потрібні завдання. Після отримання намірів користувача існують алгоритми відповідей на запитання, які дозволяють чат-ботам розуміти запитання користувачів і надавати точні відповіді. Вони включають такі методи, як

пошук інформації, ранжування документів і розуміння уривків, щоб ідентифікувати відповідну інформацію та створити відповіді.

Ми навели лише кілька прикладів алгоритмів NLP, які використовуються в розробці чат-ботів. Інші алгоритми, які можна використовувати для покращення продуктивності чат-бота з часом шляхом аналізу взаємодії користувачів і відгуків є машинне навчання (ML). Вивчаючи поведінку користувачів, алгоритми машинного навчання можуть допомогти чат-боту надавати користувачам більш персоналізовану та відповідну інформацію. Ці алгоритми можуть бути застосовані до різних завдань, таких як розуміння мови, керування діалогом і формування відповіді. Наведемо деякі алгоритми ML, які найчастіше використовуються в розробці чат-ботів:

Контрольоване навчання: Алгоритми контрольованого навчання навчаються з позначених навчальних даних, де вхідні дані поєднуються з відповідними цільовими виходами. Ці алгоритми можна використовувати для таких завдань, як розпізнавання намірів, розпізнавання іменованих об’єктів і аналіз настроїв.

Навчання з підкріпленням: алгоритми навчання з підкріпленням навчаються через взаємодію з навколишнім середовищем, отримуючи зворотний зв’язок у формі винагород або покарань. Ці алгоритми можна використовувати для керування діалогом, щоб приймати рішення про те, як реагувати на введення користувача на основі очікуваних винагород.

Моделі послідовності: моделі послідовності, такі як рекурентні нейронні мережі (RNN) і трансформаторні моделі, зазвичай використовуються для створення відповідей у чат-ботах. Ці моделі вчаться відображати вхідні послідовності (запити користувача) у вихідні послідовності (відповіді чат-бота) і навчаються з використанням великих обсягів парних даних.

Навчання передачі: навчання передачі дозволяє чат-ботам використовувати попередньо підготовлені моделі на великих наборах даних, щоб підвищити ефективність виконання конкретних завдань.

Також чат-бот використовує алгоритми аналізу настроїв, які дозволяють йому аналізувати відгуки та настрої користувачів. Це дозволяє надавати відповідні відповіді та адаптувати свою взаємодію до емоційного стану користувача. Це може допомогти покращити взаємодію з користувачами та створити довіру та взаєморозуміння з користувачами. Алгоритми аналізу настроїв на основі правил використовують попередньо визначені правила та шаблони для визначення настроїв. Ці правила можуть базуватися на ключових словах, лінгвістичних моделях або граматичних структурах, пов'язаних із позитивними чи негативними настроями. Підходи, засновані на правилах, є відносно простими та доступними для тлумачення, але можуть потребувати ручних зусиль для створення та підтримки правил. Також алгоритми покладаються на лексикони або словники настроїв, які містять слова чи фрази, позначені мітками настрою. Кожному слову в тексті присвоюється оцінка настрою на основі лексикону, і загальний настрій обчислюється шляхом узагальнення цих оцінок.

Алгоритми машинного навчання, наприклад контрольоване навчання, може навчити класифікувати текст за різними категоріями настрою на основі позначених навчальних даних. Ці алгоритми вивчають шаблони та особливості з даних, щоб передбачити настрій невидимого тексту. Поширені алгоритми машинного навчання, які використовуються для аналізу настроїв, включають опорні векторні машини (SVM), наївні байєсівські алгоритми, випадкові ліси та рекурентні нейронні мережі (RNN).

Вибір алгоритму аналізу настрою залежить від таких факторів, як доступні позначені дані, обчислювальні ресурси, вимоги до точності та потреби конкретного застосування. Часто буває корисно поекспериментувати й оцінити кілька алгоритмів, щоб знайти найбільш прийнятний підхід для даного контексту.

Також, для чат-ботів можна використовувати механізми рекомендацій, щоб пропонувати відповідні послуги чи інформацію користувачам на основі їхніх уподобань та історії. Аналізуючи дані та поведінку користувачів, механізми рекомендацій можуть допомогти чат-боту надавати персоналізовану та цінну допомогу користувачам. В розробці чат-ботів використовуються методи механізму рекомендації, такі як спільна фільтрація (аналізує поведінку та вподобання користувачів для створення рекомендацій), фільтрування на основі вмісту (аналізується вміст або характеристики елементів, якими користувач виявив інтерес, і рекомендуються схожі елементи), рекомендації на основі знань інформації про користувачів, гібридна фільтрація (поєднує численні методи рекомендацій, такі як спільна фільтрація та фільтрація на основі вмісту, щоб використовувати переваги кожного підходу)

Впроваджуючи систему рекомендацій для чат-бота, важливо враховувати такі фактори, як доступні дані, масштабованість, продуктивність у реальному часі та конкретні вимоги програми. Для створення ефективної та персоналізованої системи рекомендацій у чат-боті може знадобитися поєднання алгоритмів, методів і методів обробки даних.

Алгоритми на основі правил: алгоритми на основі правил можна використовувати для визначення конкретних правил і дій, яких чат-бот повинен виконувати на основі введених користувачами чи конкретних подій на платформі ЦНАП. Це може допомогти автоматизувати певні процеси та підвищити ефективність.

Загалом, дозволяючи чат-боту отримувати та обробляти дані з платформи та сервісів ЦНАП, ці алгоритми можуть допомогти покращити взаємодію з користувачем, оптимізувати операції та надати цінну інформацію про поведінку та вподобання користувачів. При виборі алгоритму обміну даними для ЦНАП важливо враховувати такі фактори, як доступні варіанти інтеграції, вимоги до безпеки, потреби в спілкуванні в реальному часі та масштабованість. Вибір алгоритму має відповідати можливостям та інфраструктурі ЦНАП, забезпечуючи ефективний та надійний обмін даними між чат-ботом і системою. Отже, добре розроблений і реалізований алгоритм обміну даними є ключовим для забезпечення безперебійного зв’язку

та інтеграції між чат-ботом і ЦНАП, підвищення ефективності чат-бота в наданні відповідної інформації та послуг користувачам.

РОЗДІЛ 3.
РОЗРОБКА ЧАТ-БОТУ

3.1. Розробка інформаційно-логічної моделі

В цьому підрозділі ми побудуємо інформаційно-логічну модель розробки персонального чат-боту. Оскільки до появи чат-ботів в Центрах надання адміністративних послуг були черги і люди були неспроможні записатися на той час, коли хочуть, то зараз це стало простіше. Основна мета, яку буде виконувати чат-бот - це запис людей в чергу в ЦНАП. Для того, щоб записатися в чергу потрібно буде обрати місяць, день і час прийому. Після цього бот надасть унікальний номер запису в ЦНАП з яким можна піти до адміністрації. Оскільки ЦНАП має обмежений робочий день, то саме чат-бот зможе надавати запис в чергу поза робочим днем. Також, важливим є те, що чат-бот сам вказує години, на котрі можна буде записатися, враховуючи графік і обідні перерви.

Бот розробляється для адміністративного бізнесу і основною цільовою аудиторією для користування ботами є люди від 14 до 70 років, оскільки старші люди можуть бути не навченими користуватися світовою мережею Інтернет. Було створено загальний запис в черзі без визначення конкретної цілі. Середній час, який призначений для одного клієнта — 30 хв.

Основна структура роботи чат-бота містить таку послідовність:

1. Введення привітання користувачем
2. Привітання від чат-бота
3. Користувач вводить інформацію про те, що хоче записатися на прийом в електронну чергу
4. Чат-бот запитує на який місяць хоче записатися користувач надаючи перелік запропонованих місяців(можна записатися тільки до Жовтня)
5. Далі користувач прописує в поле вводу місяць, на який хоче записатися.

6. Чат-бот запитує про день, в який користувач хотів би записатися
7. Користувач вводить день
8. Після цього чат-бот пропонує наявний вільний час для тієї дати, що вводив раніше користувач
9. Користувач обирає час
10. Чат-бот надає персональний номер, з яким потрібно прийти в ЦНАП
11. Користувач дякує за надані послуги
12. Користувач прощається
13. Чат-бот прощається

Ми навели основний алгоритм перебігу розмови чат-боту з користувачем для запису в електронну чергу.

Мова написання чат-боту — англійська, оскільки вона використовується з деяких основних причин, таких як глобальність застосування (майже всі розуміють англійську, можна охопити ширшу аудиторію)

Через те, що існує велика кількість ресурсів, інструментів і бібліотек, доступних для обробки природної мови (NLP) і машинного навчання спеціально для англійської мови, то саме написання чат-боту на англійській мові полегшує розробку, використовуючи наявні набори даних і попередньо навчені мовні моделі.

Розробка чат-боту на англійській відбувалася з тих міркувань, що багато підприємств працюють англійською мовою або мають глобальну присутність, де англійська є основною мовою спілкування.

Даний чат-бот може бути інтегрованим на сайт ЦНАП використовуючи певні бібліотеки та інтерфейс сайту. В моїй роботі цей чат-бот інтегрований на local host, оскільки моїм основним завданням було показати функції чат-боту та дослідити алгоритм дії.

Підсумовуючи, розробка інформаційно-логічної моделі для чат-бота є вирішальним кроком у створенні ефективної та інтелектуальної розмовної системи. Інформаційно-логічна модель охоплює організацію, обробку та потік інформації в

системі чат-бота для забезпечення точного розуміння введених користувачем даних і генерування відповідних відповідей.

Важливо постійно тестувати та вдосконалювати інформаційно-логічну модель, збираючи відгуки користувачів і вносячи вдосконалення для підвищення продуктивності чат-бота та взаємодії з користувачем. Зрештою, розробка інформаційно-логічної моделі для чат-бота вимагає поєднання обробки природної мови, машинного навчання та методів керування діалогами, щоб забезпечити безперебійну та змістовну взаємодію між чат-ботом і користувачами.

3.2 Технологія використання чат-боту

Розробка чат-ботів передбачає створення інтелектуальної програми, здатної брати участь у розмовах з користувачами. Для розробки чат-боту для запису в електронну чергу і його інтеграцію з сайтом ми використовували такі мови програмування:

Основна мова, яка використовувалася для створення чат-боту — це Python, яка використовує різні бібліотеки, які ми описали нижче. Також для того, щоб чат інтегрувався з сайтом, ми використовували мову frontend розробки — JavaScript та HTML, для придання боту стилю – CSS(Cascading Style Sheets).

Для розробки чат-боту використовувалися такі технології та інструменти:

1. NLTK (Natural Language Toolkit) - це популярна бібліотека в Python для роботи з даними людської мови. NLTK надає широкий спектр функціональних можливостей та інструментів для таких завдань, як токенізація(допомагає розбивати текстові дані на значущі одиниці для подальшого аналізу), створення коренів (скорочують слова до їх коренів), тегування, аналіз, семантичне обґрунтування тощо. NLTK містить

класифікатори, які можна навчити на даних із мітками, щоб автоматично класифікувати текст за попередньо визначеними класами або мітками.

NLTK широко використовується в дослідженнях обробки природної мови (NLP), освітніх установах і галузевих додатках. Широкий набір функцій і ресурсів робить його потужним інструментом для роботи з текстовими даними.

2. NumPy — це фундаментальна бібліотека Python для чисельних обчислень. Він розшифровується як Numerical Python і забезпечує підтримку великих багатовимірних масивів і матриць разом із набором математичних функцій для ефективної роботи з цими масивами. NumPy представляє нову структуру даних під назвою ndarray (n-вимірний масив), яка дозволяє ефективно зберігати однорідні дані та маніпулювати ними. Він забезпечує швидкі та ефективні операції над великими масивами, такі як поелементні операції, нарізка, зміна форми та індексування. NumPy є базовою бібліотекою для багатьох наукових і пов'язаних з даними бібліотек у Python. Ефективне зберігання та обробка даних у масивах NumPy також сприяє підвищенню продуктивності числових обчислень.

   Масиви NumPy використовують переваги безперервного розподілу пам'яті, що забезпечує ефективне використання пам'яті. Порівняно зі списками Python, масиви NumPy споживають менше пам'яті та забезпечують швидший доступ до елементів, що робить їх придатними для роботи з великими наборами даних.

3. JSON, що означає JavaScript Object Notation, є легким форматом обміну даними, який використовується в розробці чат-бота за допомогою Python. Він забезпечує простий і зрозумілий спосіб зберігання та обміну даними між різними системами. JSON використовується для зберігання структурованих даних, необхідних для роботи чат-бота. Це включає зберігання намірів, шаблонів, відповідей та інших даних, пов'язаних із навчанням і поведінкою чат-бота. Файл JSON забезпечує зручний формат для організації та зберігання цих даних, що полегшує їх читання та оновлення. Для даного чат- боту JSON

використовується для написання намірів (запити клієнтів і можливі відповіді боту).

4. У контексті розробки чат-ботів «torch» відноситься до бібліотеки PyTorch, яка є популярною системою машинного навчання з відкритим кодом для Python. PyTorch надає ряд функціональних можливостей для побудови та навчання нейронних мереж (neural etwork), які можна використовувати при розробці моделей чат-ботів. PyTorch пропонує різні інструменти та модулі, корисні для завдань NLP (обробка природньої мови), таких як токенізація, моделювання послідовності та генерація мови. Ці можливості є важливими для обробки та розуміння введення користувачами, створення відповідних відповідей та ефективного керування розмовами.

Також PyTorch надає гнучку та інтуїтивно зрозумілу структуру для розробки та впровадження архітектур нейронних мереж. Він пропонує широкий спектр попередньо створених шарів, функцій активації та алгоритмів оптимізації, які можна використовувати для створення основних компонентів моделі чат-бота, наприклад архітектури кодера-декодера або моделі трансформатора.

PyTorch підтримує прискорення графічного процесора, що дозволяє моделям чат-ботів використовувати обчислювальну потужність графічних процесорів для швидшого навчання та висновків. Це особливо корисно під час роботи з великомасштабними моделями чат-ботів або обробки значного обсягу розмовних даних.

5. Flask — популярна веб-платформа для створення веб-додатків і API на Python. Він забезпечує простий і легкий підхід до веб-розробки, дозволяючи розробникам швидко та легко створювати веб-додатки. Flask відіграє вирішальну роль у розробці спеціального чат-бота, надаючи веб-фреймворк, який дозволяє створювати програму на стороні сервера та обробляти HTTP-запити та відповіді, підтримує використання шаблонів Jinja2, які дозволяють відокремити логіку презентації від логіки програми. Шаблони допомагають створювати динамічні HTML-сторінки, поєднуючи статичний вміст із динамічними даними.

Flask містить функцію створення URL-адрес, яка дозволяє генерувати URL-адреси динамічно на основі визначених маршрутів і пов’язаних з ними функцій перегляду. Це полегшує створення зв’язків між різними частинами програми.

Тестування та аналітика. Тестування має вирішальне значення для забезпечення належної роботи чат-бота та надання точних відповідей. Інструменти аналітики можуть допомогти відстежувати взаємодію користувачів, вимірювати продуктивність і збирати статистичні дані для покращення чат-бота з часом.

Загалом розробка чат-бота вимагає поєднання NLP, машинного навчання, керування діалогами, розробки бекенда, інтеграції та дизайну інтерфейсу користувача для створення ефективного та інтелектуального розмовного агента, а також створення стилю та сайту для бота.

3.3 Програмна реалізація чат-боту та його тестування

В цьому розділі ми розглянемо поетапне створення чат-боту та протестуємо його роботу. Перед початком створення чат-боту спочатку потрібно створити папку, в якому будуть міститься файли роботи завантажити. В цю папку ми завантажимо бібліотеки, які будемо застосовувати в подальшому для розробки.

Для завантаження бібліотек потрібно зайти в Термінал (запуск від імені адміністратора), перейти до папки, в якій буде майбутня програма за допомогою такої команди: cd Name of document, де Name of document - це назва папки, в яку потрібно перейти.

Після цього напишемо команди для завантаження бібліотек: pip install Flask torch torchvision nltk. Після успішного встановлення в Терміналі повинен бути такий текст:

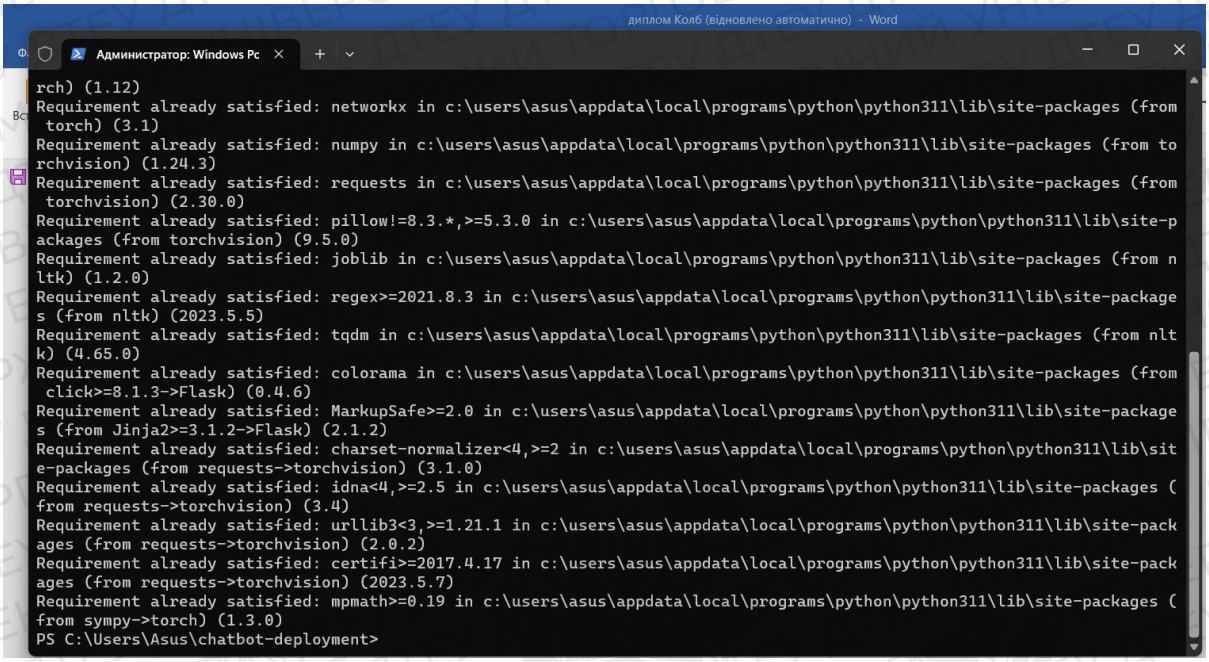

Рис.3.1. Завантаження бібліотек в папку з майбутнім чат-ботом

Далі завантажимо пакети nltk. Вводимо в термінал:

```
import nltk
nltk.download
```

Після цього потрібно перейти в редактор коду та створити формат файлу .json, в якому ми напишемо наміри для чат-боту (рис. 3.2).

Рис.3.2. Наміри, які використовуються в чат-боті

Ці наміри, які наведені в файлі будуть відображатися безпосередньо при взаємодії користувача та чат-бота. "tag"-це загальна назва одного списку, в якому є теги "patterns", які позначають основні речення, запити для користувача. Тег "responses" використовує написані речення для відповіді на запит користувача.

Далі ми скористаємося документацією по бібліотекам NLTK і numpy. Опис кожного компонента я навела в вигляді коментарів на рис 3.3

Рис.3.3. Написання коду з використанням бібліотек NLTK і numpy

Далі створимо ще один файл, формат якого .py, код якого призначений для навчання чат-бота за допомогою моделі нейронної мережі, реалізованою за допомогою PyTorch. На початку написання коду ми імпортуємо вказані бібліотеки, а також завантажимо наміри з файлу .json. Відбувається обробка даних, токенізація шаблону та додавання слів до списку, теги сортуються. Також в коді позначаються знаки чи слова, які будуть ігноруватися(рис. 3.4)

Рис.3.4. Написання коду для завантаження намірів

В наступному файлі ми реалізуємо модель нейронної мережі за допомогою PyTorch. Клас NeuralNet визначається як підклас nn.Module, який є базовим класом для всіх модулів нейронної мережі в PyTorch.

Всередині ініціалізатора клас визначає три лінійні шари (11, 12, 13) за допомогою модуля nn.Linear. Перший лінійний шар приймає input_size як вхідний розмір і hidden_size як вихідний розмір. Два інших лінійних шару мають hidden_size як вхідний, так і вихідний розміри. Клас також визначає функцію активації ReLU за допомогою модуля nn.ReLU.

Цей код визначає модель нейронної мережі з трьома лінійними рівнями та функціями активації ReLU. Він утворює базову структуру нейронної мережі і може використовуватися для різних завдань шляхом налаштування розміру вхідних даних, прихованого розміру та кількості класів.

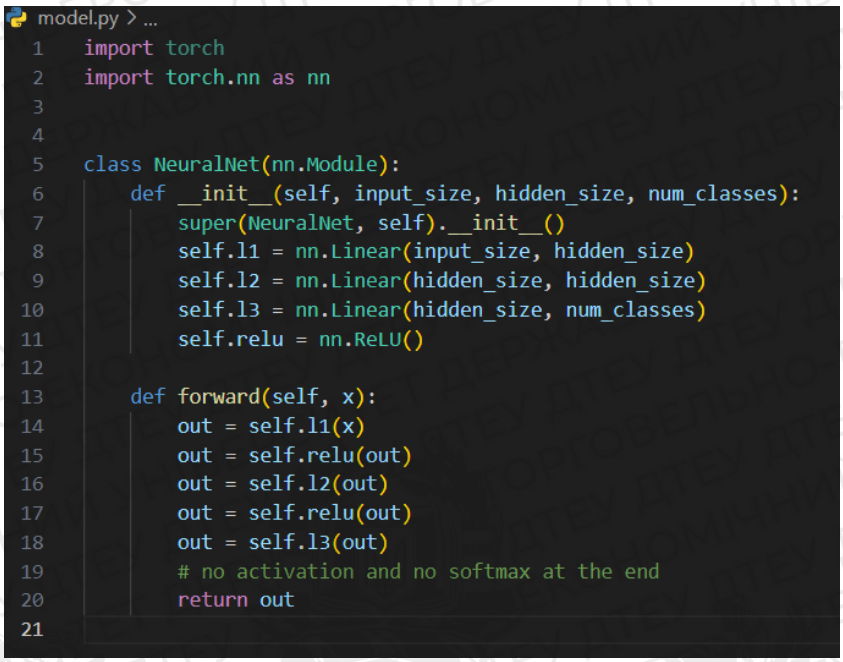

Рис.3.5. Написання коду реалізації моделі нейронної мережі

Для роботи чат-бота в консолі залишилося реалізувати фактичний чат, який буде взаємодіяти з користувачем в консолі. Код читає файл .json, який містить попередньо визначені наміри для чат-бота. Він також завантажує дані навченої моделі з файлу data.pth, який включає input_size, hidden_size, output_size, all_words, теги та model_state. Модель використовується для прогнозування вхідного тензора, і виходить прогнозований індекс класу. Якщо ймовірність передбаченого класу перевищує 0,75, вибирається та повертається випадкова відповідь із відповідного наміру. Якщо ймовірність нижча, повертається відповідь за замовчуванням "Я не розумію...".

Код містить цикл while, який дозволяє користувачеві взаємодіяти з чат-ботом. Користувач може ввести повідомлення, яке передається у функцію get_response для отримання відповіді від чат-бота. Потім відповідь друкується.

```
chat.py > ...
model = NeuralNet(input_size, hidden_size, output_size).to(device)
model.load_state_dict(model_state)
model.eval()

bot_name = "Sam"


def get_response(msg):
    sentence = tokenize(msg)
    X = bag_of_words(sentence, all_words)
    X = X.reshape(1, X.shape[0])
    X = torch.from_numpy(X).to(device)

    output = model(X)
    _, predicted = torch.max(output, dim=1)

    tag = tags[predicted.item()]

    probs = torch.softmax(output, dim=1)
    prob = probs[0][predicted.item()]
    if prob.item() > 0.75:
        for intent in intents['intents']:
            if tag == intent["tag"]:
                return random.choice(intent['responses'])

    return "I do not understand..."


if __name__ == "__main__":
    print("Let's chat! (type 'quit' to exit)")
    while True:
        # sentence = "do you use credit cards?"
        sentence = input("You: ")
        if sentence == "quit":
            break

        resp = get_response(sentence)
```

Рис.3.6. Код реалізації чат-бота з намірами

Оскільки завданням випускної кваліфікаційної роботи є створення чат-бота на сайті для ЦНАП, тому потрібно цей чат-бот інтегрувати, використовуючи Flask та JavaScript.

Наданий нижче код (рис. 3.7) — це програма Flask, яка встановлює кінцеву точку API для створення прогнозів за допомогою чат-бота. Код імпортує клас Flask із модуля flask, функції render_template і request, а також функцію jsonify із модуля flask, створює екземпляр програми Flask з назвою __name__. Використовується конфігурація спільного використання ресурсів між джерелами (CORS), щоб увімкнути обмін ресурсами між джерелами, дозволяючи запитам із різних доменів отримувати доступ до API.

Також код визначає маршрут під назвою «/predict» за допомогою декоратора @app.post. Цей маршрут очікує корисне навантаження JSON із полем

«повідомлення», що містить введений користувачем текст. Далі відбувається обробка маршуту, створюється і повертається відповідь.

```python
app.py > ...
from flask import Flask, render_template, request, jsonify
from flask_cors import CORS
from chat import get_response

app = Flask(__name__)
CORS(app)


@app.post("/predict")
def predict():
    text = request.get_json().get("message")
    # TODO: check if text is valid
    response = get_response(text)
    message = {"answer": response}
    return jsonify(message)


if __name__ == "__main__":
    app.run(debug=True)

```

Рис.3.7. Код встановлення Flask

Основою для того, щоб чатбот з’явився на сайті є HTML код, який містить структуру вікна чату. CSS створює стиль для боту. Детальніше зображено на рисунках 3.8 і 3.9.

```html
    <title>Chatbot</title>
  </head>
  <body>
    <div class="container">
      <div class="chatbox">
        <div class="chatbox__support">
          <div class="chatbox__header">
            <div class="chatbox__image--header">
              <img
                src="https://img.icons8.com/color/48/000000/circled-user-female-skin-type-5--v1.png"
                alt="image"
              />
            </div>
            <div class="chatbox__content--header">
              <h4 class="chatbox__heading--header">Chat support</h4>
              <p class="chatbox__description--header">
                Hi. My name is Sam. How can I help you?
              </p>
            </div>
          </div>
          <div class="chatbox__messages">
            <div></div>
          </div>
          <div class="chatbox__footer">
            <input type="text" placeholder="Write a message..." />
            <button class="chatbox__send--footer send__button">Send</button>
          </div>
        </div>
        <div class="chatbox__button">
          <button><img src="./images/chatbox-icon.svg" /></button>
        </div>
      </div>
    </div>
    <script src="./app.js"></script>
  </body>
</html>
```

Рис.3.8. HTML код

```css
*,
html {
  --primaryGradient: linear-gradient(93.12deg, #fb59c2 0.52%, #510b45 100%);
  --secondaryGradient: linear-gradient(
    268.91deg,
    #fb59c2 -2.14%,
    #510b45 99.69%
  );
  --primaryBoxShadow: 0px 10px 15px rgba(0, 0, 0, 0.1);
  --secondaryBoxShadow: 0px -10px 15px rgba(0, 0, 0, 0.1);
  --primary: #d53a87;
}

/* CHATBOX
=============== */
.chatbox {
  position: absolute;
  bottom: 30px;
  right: 30px;
}

/* CONTENT IS CLOSE */
.chatbox__support {
  display: flex;
  flex-direction: column;
  background: #eee;
  width: 300px;
  height: 350px;
  z-index: -123456;
  opacity: 0;
  transition: all 0.5s ease-in-out;
}
```

Рис.3.9. CSS стиль для чат-боту

Все, що нам залишилося зробити перед тестуванням боту- реалізувати код JavaScript. Наданий код є класом JavaScript під назвою Chatbox, який керує функціями інтерфейсу чат-бокса. Конструктор ініціалізує об'єкт args посиланнями на елементи DOM, такі як кнопка відкриття, вікно чату та кнопка надсилання. Метод відображення встановлює прослуховувачі подій для кнопки відкриття та кнопки надсилання. Він відстежує події натискання кнопки «Відкрити», щоб перемикати стан вікна чату між активним і неактивним. Він також відстежує події натискання кнопки надсилання та події натискання клавіші Enter у полі введення, щоб обробити надсилання повідомлень. В кінці код створює екземпляр класу Chatbox і викликає метод відображення для налаштування прослуховувачів подій та ініціалізації інтерфейсу користувача chatbox.

```
39    onSendButton(chatbox) {
40      var textField = chatbox.querySelector('input')
41      let text1 = textField.value
42      if (text1 === '') {
43        return
44      }
45
46      let msg1 = { name: 'User', message: text1 }
47      this.messages.push(msg1)
48
49      fetch('http://127.0.0.1:5000/predict', {
50        method: 'POST',
51        body: JSON.stringify({ message: text1 }),
52        mode: 'cors',
53        headers: {
54          'Content-Type': 'application/json',
55        },
56      })
57        .then((r) => r.json())
58        .then((r) => {
59          let msg2 = { name: 'Sam', message: r.answer }
60          this.messages.push(msg2)
61          this.updateChatText(chatbox)
62          textField.value = ''
63        })
64        .catch((error) => {
65          console.error('Error:', error)
66          this.updateChatText(chatbox)
67          textField.value = ''
68        })
69    }
70
71    updateChatText(chatbox) {
72      var html = ''
73      this.messages
74        .slice()
75        .reverse()
```

Рис.3.10. Реалізація коду JavaScript

Тестування чат-боту:

1. Привітання та прохання записатися в чергу:

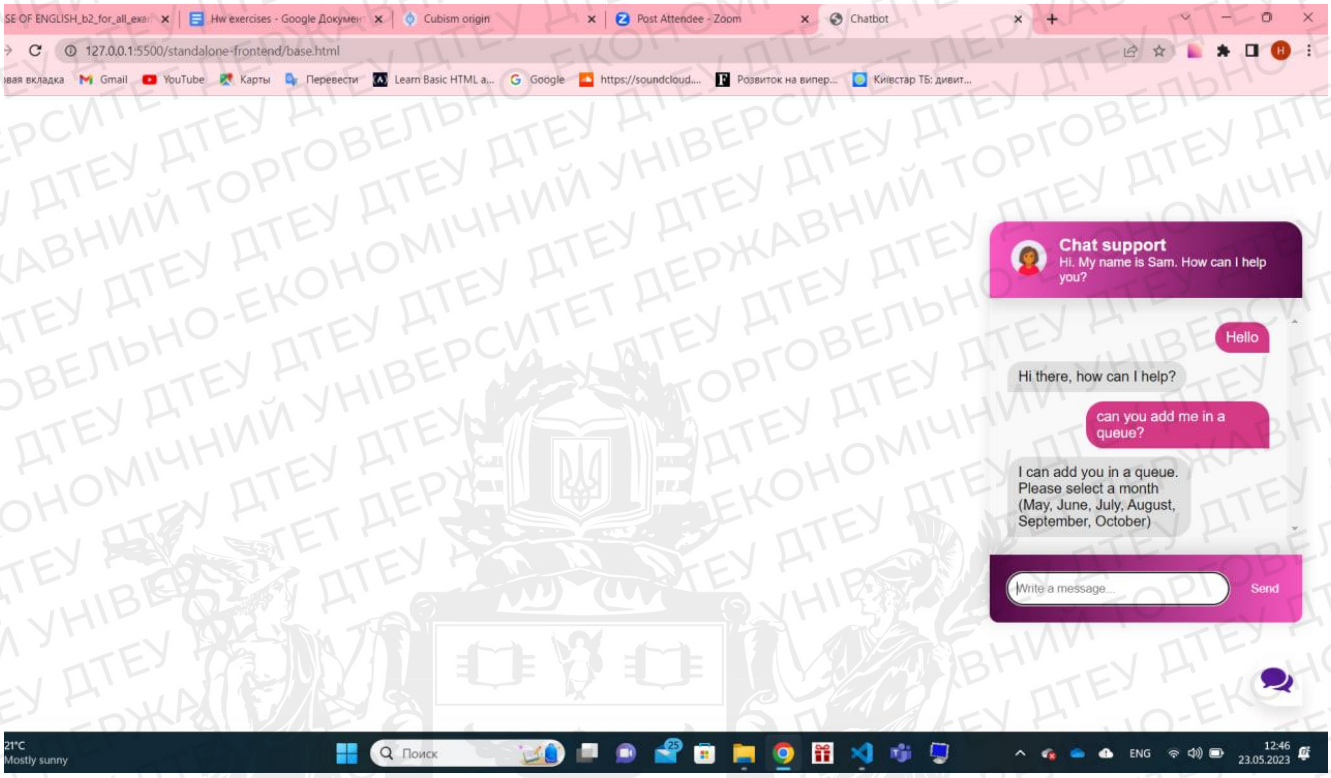

Рис.3.11. Тестування чат-боту

2. Вибір місяця та часу реєстрації

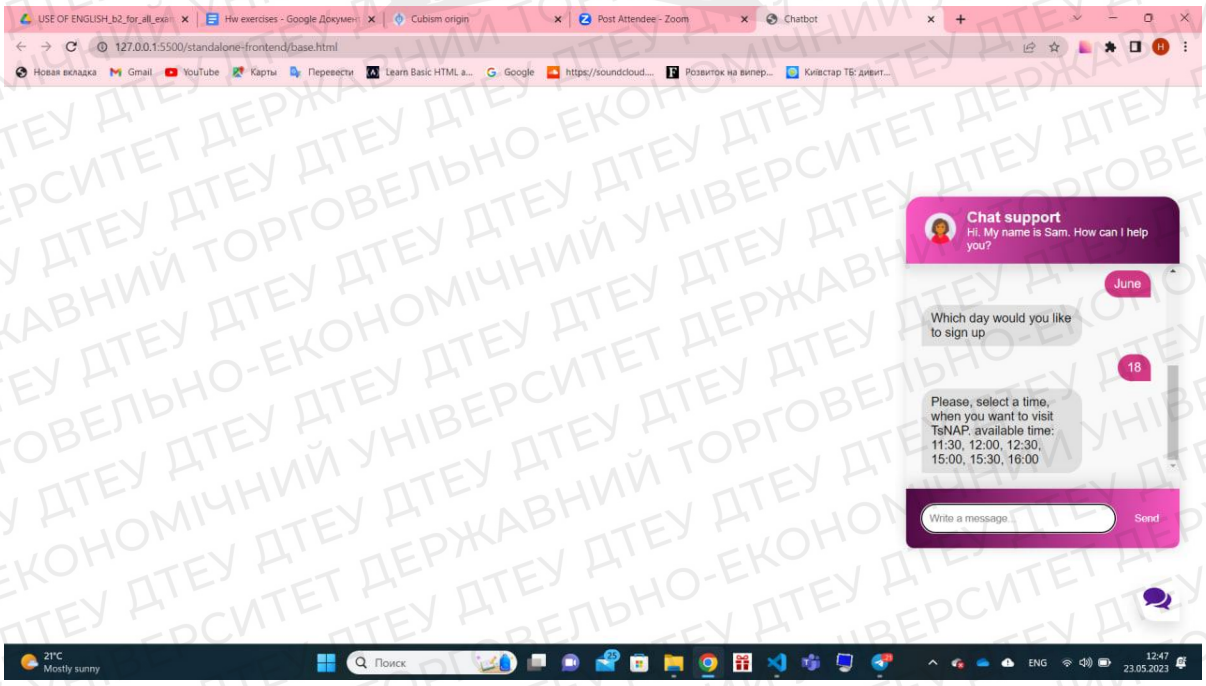

Рис.3.12. Тестування чат-боту(місце, час)

3. Номер електронного запису та прощання

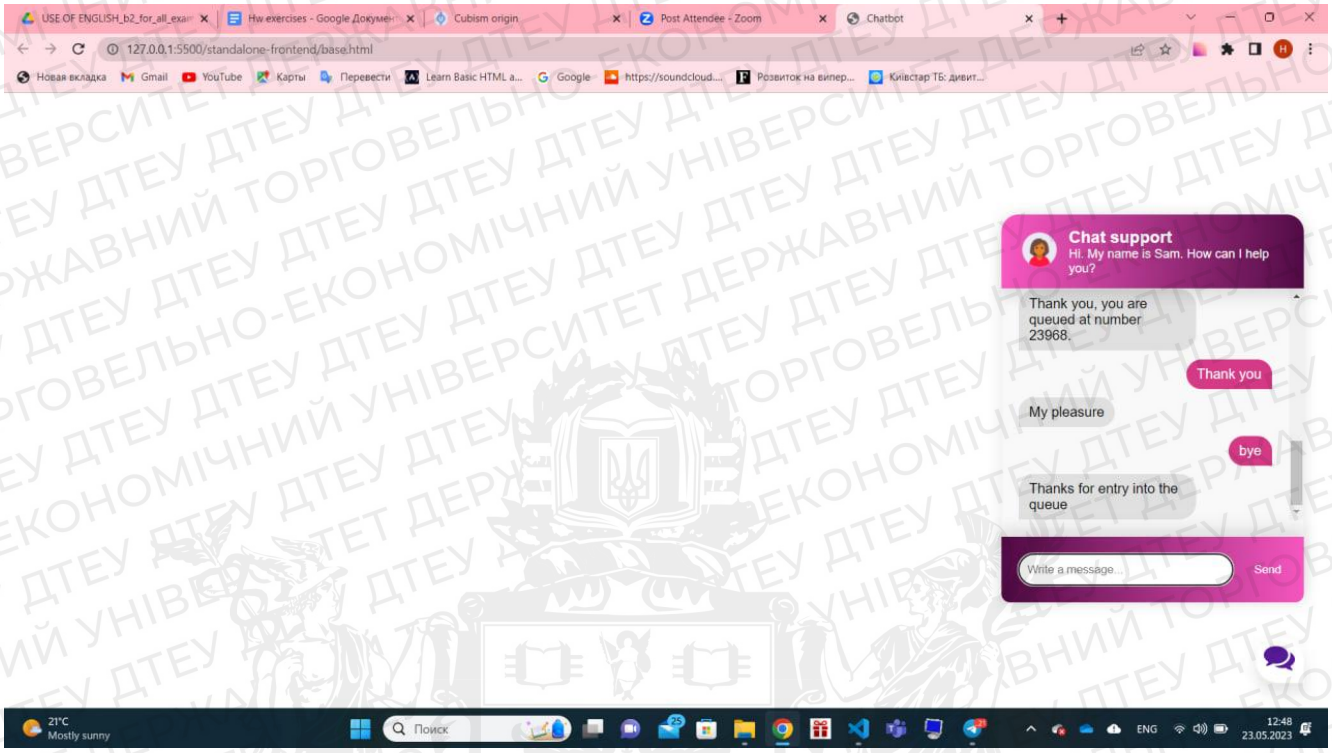

Рис.3.13. Тестування чат-боту(ел.запис, прощання)

У підсумку був створений чат-бот у вигляді асистента, який може допомогти у вирішенні питання запису в чергу. Для реалізації моделі чат-боту був задіяний Frontend (HTML, CSS, JavaScript). Для Backend був задіяний Python. Також була використана бібліотека Flask, яка допомогла взаємодіяти між Frontend та Backend.

ВИСНОВКИ

Випускна кваліфікаційна робота охоплює як теоретичні, так і практичні дослідження, спрямовані на алгоритм та розробку чат-боту для електронної черги в ЦНАП, кінцевою метою якого є зменшення черг в реальному житті та запис в чергу за допомогою боту. Результати досліджень послужили основою для створення сайту, який містить чат-бот у вигляді персонального консультанта для запису в чергу. Проведене дослідження дало суттєві **висновки**, які представлені таким чином:

1. Розробка чат-бота для електронної черги виявилася ефективним засобом покращення надання адміністративних послуг. Використовуючи сучасні інформаційні технології, чат-бот дав змогу регіональним органам управління отримувати вичерпну, достовірну та оперативну інформацію про стан черги та відповідно оптимізувати адміністративні процеси.
2. Дослідження підкреслило важливість впровадження сучасних інформаційних технологій у розробку чат-бота. Ця інтеграція забезпечила використання відповідних та надійних джерел даних, що призвело до отримання точної та актуальної інформації для прийняття обґрунтованих управлінських рішень щодо адміністративних послуг.
3. У дослідженні успішно розроблено методику визначення комплексного показника ефективності електронної черги. Крім того, була створена інформаційно-логічна модель автоматизованої системи, яка є основою для алгоритму та структури чат-бота. Такий методологічний підхід сприяв загальній ефективності та надійності системи чат-ботів.
4. Алгоритм і структуру чат-бота було перетворено на реалізацію програмного забезпечення. Система була спеціально створена для оцінки показників та оптимізації електронної черги Центру надання адміністративних послуг у режимі реального часу. Технологія, розроблена для використання чат-бота, забезпечила безперебійну інтеграцію та надала цінну інформацію про ефективне управління адміністративними процесами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Обліковий запис. Двоетапна аутентифікація (2FA). [Електронний ресурс] / ТОВ "Хостинг «Україна»" // Wiki документація. —2006-2023.—розд. 1.2.2. — Режим доступу: https://www.ukraine.com.ua/uk/wiki/account/security/otp/

2. Головне управління ДПС у Донецькій області. Навчально-організаційні заходи для представників територіальних громад [Електронний ресурс] // Державна податкова служба України. — Режим доступу: https://dn.tax.gov.ua/media-ark/news-ark/517067.html(опубліковано 18 жовтня 2021 о 14:04)

3. Закон України від 06.09.2012 № 5203 VI "Про адміністративні послуги" / Відомості Верховної Ради. — 2013. — № 32. —с. 409. — Стаття 12.

4. Івано-Франківська обласна державна адміністрація. Цифровізація важлива складова розвитку цифрових трансформацій України [Електронний ресурс] //Офіційний веб-портал Івано-Франківської обласної державної адміністрації. — Режим доступу: https://www.if.gov.ua/news/cifrovizaciya-vazhliva-skladova-rozvitku-cifrovih-transformacij-ukrayini (Опубліковано 11 лютого 2022 року о 15:28)

5. Краковська А. Є., Бабик М. К. Цифровізація адміністративних послуг в Україні: проблеми та перспективи розвитку // Науковий вісник Ужгородського Національного Університету. — 2022. — с. 333-334. — Режим доступу: http://visnyk-pravo.uzhnu.edu.ua/article/view/258976

6. Платформа для маркетингу та продажів SendPulse. Чат-бот [Електронний ресурс] // SendPulse. — 2019. — Режим доступу: https://sendpulse.ua/support/glossary/chatbot

7. Програми ЄС «U LEAD. Роль ЦНАП (центрів дія) у використанні е-послуг. Навчальний посібник для працівників ЦНАП (Центрів Дія). — с. 5, 14

8. Технічний фаховий коледж Луцького національного технічного університету. Лекція 32: Мережа Інтернет [Електронний ресурс] // LNTU. — Режим доступу: https://e-tk.lntu.edu.ua/mod/page/view.php?id=3589

9. Павло Кривенко. Штучний інтелект (AI): що це таке і чому це важливо? / Павло Кривенко // Центр досліджень армії, конверсії та роззброєння. — 2018.

10. Яготинська міська рада. Перелік адміністративних послуг [Електронний ресурс] // Яготинська міська рада. — додат. за 20.09.2022. — Режим доступу: https://yagotinmiska-rada.gov.ua/struktura-cnap-14-57-49-16-11-2021/

11. Ярош Є. В. Інформаційне суспільство: проблеми та перспективи. Програмні засоби розробки чат-ботів // VII Всеукраїнська науково-практична конференція. — С. 77-78.

12. Aman kumar. Building a Chatbot in Python (using chatterbot) and deploying it on web [Електронний ресурс] // Medium. — Опубліковано: Aug 3, 2020. — Режим доступу: https://medium.com/@kumaramanjha2901/building-a-chatbot-in-python-using-chatterbot-and-deploying-it-on-web-7a66871e1d9b

13. Ambika Choudhury. 7 Top NLP Libraries Java Developers Should Know In 2019 [Електронний ресурс] // Analytics India Magazine. — Опубліковано: June 5, 2019. — Режим доступу: https://analyticsindiamag.com/7-top-nlp-libraries-java-developers-should-know-in-2019/

14. Proceedings of the 13th Swecog conference: Poster presentations [Електронний ресурс] / [Anders Arwestrom Jansson, Anton Axelsson, Rebecca Andreasson, Erik Billing.]. // Skovde University Studies in Informatics 2017:2. — с. 28-30. — Режим доступу: https://mobidev.biz/blog/ai-virtual-assistant-technology-guide

15. Auth0 docs by Okta. Article: JSON Web Tokens [Електронний ресурс] // Auth0. - Режим доступу: https://auth0.com/docs/secure/tokens/json-web-tokens

16. Barbara Zito. Your Guide To Understanding Internet Speed [Електронний ресурс] / Barbara Zito, Samantha Allen // Forbes. — Режим доступу: https://www.forbes.com/home-improvement/home/all-about-internet-speed/

17. Cobus Greyling. Chatbots: Natural Language Generation In 7 Easy Steps | Medium [Електронний ресурс] // Medium. — Опубліковано: Oct 14, 2019. — Режим доступу: https://cobusgreyling.medium.com/chatbots-natural-language-generation-in-7-easy-steps-77887a6de249

18. Documentation. Natural Language Toolkit [Електронний ресурс] // NLTK. — Режим доступу: https://www.nltk.org/

19. GLOSSARY: CYBERSECURITY TERMS & DEFINITIONS. What is SSL/TLS Encryption? [Електронний ресурс] // F5. — Режим доступу: https://www.f5.com/glossary/ssl-tls-encryption

20. Google Cloud. Dialogflow documentation. Dialogflow Messenger [Електронний ресурс] // Google Cloud. — Режим доступу: https://cloud.google.com/dialogflow/es/docs/integrations/dialogflow-messenger

21. Harinder Singh Batra. Blog post. Use OAuth-based Authentication for SAP Conversational AI chatbots [Електронний ресурс] // SAP Blogs. — Опубліковано: December 2, 2020. — Режим доступу: https://blogs.sap.com/2020/12/02/why-and-how-to-migrate-your-chatbots-to-oauth-based-authentication/

22. IBM documentation. What is a REST API [Електронний ресурс] // IBM. - Режим доступу: https://www.ibm.com/topics/rest-apis

23. Indrek Vainu. Conversational Chatbots Maintaining Context Awareness – Complete Guide [Електронний ресурс] // AlphaChat. — Опубліковано: September 7, 2021. — Режим доступу: https://www.alphachat.ai/blog/conversational-chatbot)

24. Jenna Alburger. Rule-Based Chatbots vs. AI Chatbots: Key Differences [Електронний ресурс] // Hubtype Blog. — Опубліковано: October 25, 2018. — Режим доступу: https://www.hubtype.com/blog/rule-based-chatbots-vs-ai-chatbots

25. Joren Wouters. Chatbot software reviews. Botpress Review [Електронний ресурс] // Chatimize. — Опубліковано: Jun 4, 2021. — Режим доступу: https://chatimize.com/reviews/botpress/

26. Joren Wouters. Chatbot software reviews. Tars Review [Електронний ресурс] // Chatimize. — Опубліковано: Jul 3, 2020. — Режим доступу: https://chatimize.com/reviews/tars/

27. Krisette Capati. Data Exchange Across Networks: How Does It Work [Електронний ресурс] // iFaxApp Blog. — Режим доступу: https://www.ifaxapp.com/blog/data-exchange-networks

28. Microsoft documentation. Article: What is the Bot Framework SDK? [Електронний ресурс] // Microsoft Docs. — Опубліковано: 11/30/2022 — Режим доступу: https://learn.microsoft.com/en-us/azure/bot-service/bot-service-overview?view=azure-bot-service-4.0

29. Oisin Muldowney. CHATBOTS: An Introduction And Easy Guide To Making Your Own. / Oisin Muldowney. — First published 2017 by Curses & Magic, Dublin, Ireland Text © Curses & Magic —2017.— p. 1-5, p. 12-23

30. Meyer, Patrick. "Is Your Chatbot Accessible?" [Електронний ресурс]. // AI Plainenglish. — Опубліковано: Aug 6, 2021 — Режим доступу: https://ai.plainenglish.io/is-your-chatbot-accessible-6b89a5e300f1

31. Pavel Jiřík. The future of voice assistants. [Електронний ресурс] // Phonexia. — Опубліковано: April 25, 2022 — Режим доступу: https://www.phonexia.com/blog/the-future-of-voice-assistants/

32. Przemyslaw Kalka. Article. Fundamentals of Rule-based Chatbots. [Електронний ресурс] // Chatomizer. — Режим доступу: https://www.chatomiser.com/chatbot/rule-based

33. Rashid Khan Anik Das. Build Better Chatbots: A Complete Guide to Getting Started with Chatbots. / Rashid Khan Anik Das. — Apress published— 2018.— p. 1-3

34. Rob Guilfoyle. How Secure Are Chatbots? [Електронний ресурс] / Rob Guilfoyle. — Опубліковано: January 11, 2017 — Режим доступу: https://www.abe.ai/blog/how-secure-are-chatbots/

35. Sansone, S.-A. "Interoperability Standards - Digital Objects in Their Own Right." / Sansone, S.-A, Rocca-Serra, P. — Wellcome Trust, 2016. — Version 1. — С. 2-6.

36. Khan, S. M. A. "Post: Microsoft Azure Bot Service." [Електронний ресурс]. // Microsoft documentation. — Опубліковано: Jan 20, 2023 — Режим доступу: https://www.researchgate.net/publication/368661757_Microsoft_Azure_Bot_Service#pf6

37. Skoromets, S. "Tech guide: How to Build a Conversational Chatbot and Avoid If/Else Statements." [Електронний ресурс]. / Skoromets.S. — Опубліковано:

Oct 9, 2018. — Режим доступу: https://www.eliftech.com/insights/how-to-build-a-conversational-chatbot-and-avoid-if-else-statements/

38. Tudip blog. "What is Dialogflow? What are the uses and Benefits of Dialogflow?" [Електронний ресурс]. // Tudip blog. — Опубліковано: 25 May 2020. — Режим доступу: https://tudip.com/blog-post/what-is-dialogflow-what-are-the-uses-and-benefits-of-dialogflow/

39. Vivasoft. "10 Amazing Things You Can Build with JavaScript Programming Language." [Електронний ресурс]. // VivasoftLTD. — Режим доступу:https://www.vivasoftltd.com/amazing-things-you-can-build-with-javascript-programming-language/

ДОДАТОК

Програмний код запитів для чат-боту

```
{
 "intents": [
  {
   "tag": "greeting",
   "patterns": [
    "Good day",
    "Good evening",
    "Good morning",
    "Hi",
    "Hey",
    "How are you",
    "Is anyone there?",
    "Hello"
   ],
   "responses": [
    "Hi there, how can I help?",
    "Hello, can I help you to add in a queue"
   ]
  },
  {
   "tag": "goodbye",
   "patterns": ["Bye", "See you later", "Goodbye"],
   "responses": ["Have a nice day", "Thanks for entry into the queue"]
  },
  {
   "tag": "thanks",
   "patterns": ["Thanks", "Thank you", "That's helpful", "Thank's a lot!"],
```

```
  "responses": ["Happy to help!", "Any time!", "My pleasure"]
 },
 {
  "tag": "add in a queue",
  "patterns": [
   "can you put me in the queue?",
   "I would like to sign up",
   "I would like to sign up for the TsNAP queue"
  ],
  "responses": [
   "I can add you in a queue. Please select a month (May, June, July, August, September, October)"
  ]
 },
 {
  "tag": "chosen month",
  "patterns": ["May", "June", "July", "August", "September", "October"],
  "responses": ["Which day would you like to sign up"]
 },
 {
  "tag": "chose day",
  "patterns": [
   "1",
   "2",
   "3",
   "4",
   "5",
   "6",
   "7",
   "8",
```

```
      "9",
      "10",
      "11",
      "12",
      "13",
      "14",
      "15",
      "16",
      "17",
      "18",
      "19",
      "20",
      "21",
      "22",
      "23",
      "24",
      "25",
      "26",
      "27",
      "28",
      "29",
      "30",
      "31"
    ],
    "responses": [
      "Please, select a time, when you want to visit TsNAP. available time: 11:30, 12:00,
12:30, 15:00, 15:30, 16:00"
    ]
  },
  {
```

```
    "tag": "chosen time",
    "patterns": ["11:30", "12:00", "12:30", "15:00", "15:30", "16:00"],
    "responses": [
     "Thank you, you are queued at number 23968.",
     "Thank you, you are queued at number 29837.",
     "Thank you, you are queued at number 84647.",
     "Thank you, you are queued at number 95636."
    ]
   }
  ]
}
```