

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Сумський державний університет**

**Факультет електроніки та інформаційних технологій**

**Кафедра комп'ютерних наук**

«До захисту допущено»

В.о. завідувача кафедри

Оксана ШОВКОПЛЯС  
(підпис)

05 грудня 2024 р.

**КВАЛІФІКАЦІЙНА РОБОТА**

**на здобуття освітнього ступеня магістр**

зі спеціальності 122 «Комп'ютерні науки»

освітньо-професійної програми «Інформатика»

на тему: «Інформаційна технологія створення системи розпізнавання голосу для формування навичок міжособистісної комунікації»

здобувача групи ІН.м - 33 Папіжука Данила Олександровича

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

Данило ПАПІЖУК

\_\_\_\_\_  
(підпис)

Керівник

доцент кафедри КН,

кандидат фізико-математичних наук,

доцент

Оксана ШОВКОПЛЯС

\_\_\_\_\_  
(підпис)

**Суми – 2024**

**Сумський державний університет**  
Факультет електроніки та інформаційних технологій  
Кафедра комп'ютерних наук

«Затверджую»

В.о. завідувача кафедри

Оксана ШОВКОПЛЯС

(підпис)

**ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ**

**на здобуття освітнього ступеня магістра**

зі спеціальності 122 «Комп'ютерні науки», освітньо-професійної програми «Інформатика»  
здобувача групи ІН.м-33 Папіжука Данила Олександровича

1. Тема роботи: «Інформаційна технологія створення системи розпізнавання голосу для формування навичок міжособистісної комунікації»  
затверджую наказом по СумДУ від «03» грудня 2024 р. № 1257-VI
2. Термін здачі здобувачем кваліфікаційної роботи до 09 грудня 2024 року
3. Вхідні дані до кваліфікаційної роботи \_\_\_\_\_
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)  
1) Аналіз предметної області, формування та визначення мети дослідження.  
2) Огляд технологій, що задіяні для створення голосових систем. 3) Розроблення системи міжособистісних зв'язків. 4) Аналіз отриманих результатів.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) \_\_\_\_\_
6. Консультанти до проєкту (роботи), із значенням розділів проєкту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання « \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

Завдання прийняв до виконання \_\_\_\_\_  
(підпис)

Керівник \_\_\_\_\_  
(підпис)

**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	<i>Аналіз предметної області, формування та визначення мети дослідження.</i>		
2	<i>Розгляд технологій, що використовуються для голосових систем.</i>		
3	<i>Розроблення системи міжособистісних зв'язків.</i>		
4	<i>Аналіз отриманих результатів</i>		
5	<i>Оформлення пояснювальної записки до кваліфікаційної роботи</i>		

Здобувач вищої освіти \_\_\_\_\_  
(підпис)

Керівник \_\_\_\_\_  
(підпис)

## АНОТАЦІЯ

**Записка:** 65 стор., 14 рис., 7 табл., 2 додатки, 56 використаних джерел.

**Обґрунтування актуальності теми роботи** – Тема кваліфікаційної роботи є актуальною, оскільки вона зосереджена на дослідженні комунікативних звичок суб'єктів, впровадженні голосових помічників в цей процес та використання його для побутових потреб.

**Об'єкт дослідження** – процес визначення особливостей міжособистісних зв'язків.

**Предмет дослідження** – методи використання голосових помічників для диверсифікації окремих процесів.

**Мета роботи** – розробка голосового помічника для міжособистісної комунікації.

**Методи дослідження** – використання інформаційно-комунікаційних технологій для проектування, побудови та розробки голосового помічника, теорії соціальної комунікації для розширення роботи помічника.

**Результати** – створено голосовий помічник для реалізації загальних комунікаційних потреб. Можливість задавати питання, обговорювати події, автоматизувати використання власних систем: ноутбук, комп'ютер, телефон, та використання в щоденних справах.

ГОЛОСОВИЙ ПОМІЧНИК, ШТУЧНИЙ ІНТЕЛЕКТ, КОМУНІКАЦІЯ,  
ІНКЛЮЗИВНІСТЬ, RYTHON, ПСИХОЛОГІЯ

## ЗМІСТ

ВСТУП .....	5
1 ТЕОРЕТИЧНІ ВІДОМОСТІ.....	6
1.1 Технологічна складова .....	6
1.2 Соціальна складова.....	9
1.3 Інклюзія у використанні.....	15
1.4 Фітнес помічник .....	17
1.5 Способи реалізації.....	26
1.6 Постановка задачі .....	32
2 ВИБІР МЕТОДІВ РЕАЛІЗАЦІЇ.....	33
2.1 Початкові кроки .....	33
2.2 Інструменти для розробки .....	34
2.3 Нефункціональні вимоги .....	34
2.4 Функціональні вимоги .....	37
3 ТЕХНІЧНІ ВІДОМОСТІ.....	39
3.1 Скелет програми .....	39
3.2 Озвучення тексту .....	41
3.3 Виконання команд.....	42
3.4 Ключове слово .....	43
3.5 Історія розмови .....	43
3.6 Штучний інтелект.....	44
3.7 Безпекові заходи.....	44
3.8 Алгоритм роботи.....	45
ВИСНОВКИ.....	46
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	47
ДОДАТОК А ЛІСТИНГ ПРОГРАМНОГО КОДУ.....	51
ДОДАТОК Б СТИЛІ ТА АНІМАЦІЯ .....	60

## ВСТУП

**Актуальність.** Ця робота спрямована на розробку інструментів, призначених для покращення навичок міжособистісного спілкування. Створення голосового асистента для цієї мети й використовує технології для полегшення взаємодії з користувачем, пропонуючи доступний і адаптований підхід до розвитку навичок ефективного автоматизованого спілкування.

**Об'єкт дослідження.** Процес визначення аспектів міжособистісних комунікацій.

**Предмет дослідження.** Методологія розробки голосових помічників для покращення навичок взаємодії.

**Гіпотеза.** Формування міжособистісних комунікацій можна досягти шляхом впровадження інформаційної системи, що реалізує функціональність голосового асистенту в поєднанні зі штучним інтелектом.

**Наукова новизна.** Згадана в роботі технічна складова дозволить покращити розуміння принципів і способів використання голосових помічників, дозволяючи впроваджувати їх, як серед інженерних потреб, так і потреб міжособистісної взаємодії.

**Апробація матеріалів роботи.** Основні результати роботи оприлюднені та обговорені на міжнародній науково-технічній конференції студентів та молодих вчених «Інформатика, математика, автоматика» (ІМА – 2024).

**Структура.** Дана робота складається зі вступу, теоретичних відомостей, вибір методу реалізації поставленої задачі, технічних відомостей, висновків, списку використаних джерел та додатків.

**Зв'язок роботи з науковою темою.** Кваліфікаційна робота виконана на кафедрі комп'ютерних наук та пов'язана з виконанням науково-дослідної роботи № 0120U103071 «Моделі та методи інформаційних технологій для аналізу та синтезу структурних, інформаційних і функціональних моделей об'єктів і процесів, що автоматизуються» (2020-2024).

# 1 ТЕОРЕТИЧНІ ВІДОМОСТІ

## 1.1 Технологічна складова

Інформаційно-комунікаційні технології (ІКТ) – це широка галузь, що постійно розвивається, яка охоплює розробку, управління та впровадження цифрових рішень. Однією з головних цілей ІТ у цій сфері є полегшення взаємодії в соціальних, освітніх чи професійних цілях. Інтеграція ІТ з повсякденними засобами комунікації трансформувала не лише спосіб нашої взаємодії, але й те, як ми вчимося та вдосконалюємо міжособистісні навички [1].

Зокрема, технології, що покращують розпізнавання голосу, обробку мови та точність відповідей, відкрили нові можливості для опосередкованої міжособистісної взаємодії. Ці технології ґрунтуються на теоріях штучного інтелекту та обробки природної мови, надалі ШІ та ОПМ відповідно. Вони дозволяють перетворювати голос на текст у реальному часі, аналізувати розмови та виокремлювати емоційні нюанси з мовлення. Перехід від текстового до голосового технологічного ландшафту є значним кроком вперед в еволюції комунікації. Поява голосових помічників зробила їх надзвичайно цінним інструментом, не лише засобом підвищення продуктивності, але й інструментом для розвитку навичок різної форми [2].

Голосовий асистент, або голосовий помічник (ГП) – це програмне рішення на основі алгоритмів ШІ та ОПМ, яке дозволяє користувачам взаємодіяти з пристроями за допомогою голосу [3]. Технологія, що лежить в основі асистентів, побудована на декількох теоретичних і технічних рівнях, які в сукупності полегшують обробку усних команд і генерування відповідей. З детальним описом рівнів можна ознайомитись, звернувши увагу на рисунок 1.1.

Розвиток технології голосових помічників ґрунтується на низці теоретичних засад (табл. 1.1), у тому числі тих, що стосуються взаємодії людини з комп'ютером, когнітивних наук і навіть соціальної психології [4–6].

<b>Теорія розпізнання мовлення</b>	Розпізнавання мовлення є початковим етапом взаємодії з будь-яким голосовим помічником. Від системи вимагається точне розпізнавання вимовлених слів, що становить значну проблему через варіації акцентів, вимови та швидкості мовлення. Ця галузь значною мірою покладається на методи обробки сигналів і машинного навчання (МН), щоб відрізнити слова і фрази від фонового шуму та інших форм перешкод.
<b>Обробка природної мови</b>	Після того, як розмовна мова ідентифікована, алгоритми обробки природної мови (ОПМ) переходять до інтерпретації значення слів. ОПМ охоплює ряд підпроцесів, включаючи токенизацію, синтаксичний та розпізнавання намірів. ОПМ ґрунтується на лінгвістичній теорії, яка вивчає синтаксис, семантику та контекст для виведення значення.
<b>Машинне навчання та глибоке навчання</b>	Застосування МН та ГН має величезне значення для розробки голосових помічників, здатних до безперервного навчання та адаптації. Завдяки використанню рекурентних нейронних мереж та трансформаційних моделей, голосові помічники покращують свої можливості інтерпретації складних команд, виявлення емоцій користувача та генерування відповідей, що імітують природну взаємодію з людиною. ГН полегшує здатність голосового асистента вдосконалювати свої відповіді з часом, тим самим адаптуючись до конкретних вимог і вподобань окремих користувачів.
<b>Перетворення тексту в мовлення</b>	Після того, як асистент визначив відповідь, використовується технологія перетворення тексту в мовлення для перетворення тексту в чульне мовлення. Цей процес залежить від використання функції просодії, яка охоплює ритм, висоту та інтонацію мовлення, щоб досягти природного звучання. Складні алгоритми перетворення дозволяють голосовим асистентам використовувати різні тони і стилі, залежно від контексту взаємодії. Це надає взаємодії певної індивідуальності.

Рисунок 1.1 – Теоретичні та технічні рівні [4–6]

Розуміння механіки людської комунікації та когнітивної обробки має вирішальне значення для розробки голосових помічників, які можуть ефективно підтримувати та покращувати міжособистісні навички.

Таблиця 1.1 – Теоретичні засади [4–6]

<b>Взаємодія людини та комп'ютера</b>	<b>Теорія соціальної присутності</b>	<b>Лінгвістична теорія та прагматика</b>
<b>1</b>	<b>2</b>	<b>3</b>
Основна для розробки голосових помічників, які сприяють інтуїтивно зрозумілій, орієнтованій на користувача та контекстно-чутливій взаємодії.	Відображає, якою мірою цифровий об'єкт сприймається як «присутній» у розмові, для створення більш особистих та зрозумілих помічників.	Голосові асистенти повинні бути здатні розуміти мову в буквальному сенсі, а також інтерпретувати приховані значення та емоції.

Продовження таблиці 1.1

1	2	3
Завдяки створенню зручного інтерфейсу, в якому пріоритетами є простота, доступність і легкість використання, голосові асистенти можуть сприяти більш безперешкодному спілкуванню між людьми і пристроями.	Імітуючи відчуття присутності, помічники можуть створювати інтерактивний і захоплюючий досвід, імітуючи міжособистісну динаміку.	Застосування прагматичної теорії, сприяє розробці асистентів, здатних брати участь у більш нюансованих розмовах, тим самим сприяючи більш динамічній та адаптивній взаємодії.

Незважаючи на стрімкий розвиток голосових цифрових помічників та їх значну поширеність, їх використання викликає значні занепокоєння з точки зору безпеки. Однією з головних проблем є конфіденційність даних користувачів [7]. Запис і обробка голосових команд голосовими помічниками може викликати занепокоєння щодо потенційного витоку інформації, яка стосується особистого життя людини. Крім того, існує ризик несанкціонованого доступу до записів, що може мати наслідки для конфіденційності користувача. Слід зазначити ймовірність використання голосових помічників для атак на безпеку даних, таких як імітація голосу з метою отримання несанкціонованого доступу. Дослідники в цій галузі визначили п'ять потенційних сценаріїв атак на системи:

- Backdoor – метод обходу звичайних процедур автентифікації для отримання несанкціонованого доступу до системи.
- Voice squatting/voice masquerading – Під час атаки з маскуванню голосу зловмисник подовжує час взаємодії з запущеним додатком, не повідомляючи про це користувачеві.
- Adversarial – тип кібератаки, який має на меті обдурити моделі машинного навчання, вводячи в них зловмисні дані.



- Hidden command – передбачає вбудовування команд в аудіо або інші носії, які непомітні для людини.
- DolphinAttack – технологія, яка використовує голосові помічники, такі як Siri, Alexa або Google Assistant, за допомогою ультразвукових частот, які не чутні для людини [8].

Розробники та користувачі повинні бути вкрай уважними до цих питань та вживати ефективних заходів безпеки для забезпечення захисту особистих даних та виключення можливості використання голосових помічників для шкідливих намірів.

## 1.2 Соціальна складова

Процес міжособистісного спілкування формується під впливом безлічі соціальних, психологічних і біологічних факторів. З появою інформаційних технологій, зокрема систем, що реагують на голос, методи, які використовуються для розвитку комунікативних навичок і налагодження зв'язків з іншими людьми, зазнають значної трансформації. Глибше розуміння механізмів, що лежать в основі соціальної взаємодії, когнітивних процесів і навіть біологічних впливів, може сприяти розробці та використанню цифрових інструментів, які підтримують ефективну та емпатичну комунікацію. Це має важливе значення в таких сферах, як вивчення мови, соціальна активність та емоційний інтелект, де технології можуть імітувати аспекти людської взаємодії, дозволяючи користувачам практикувати і вдосконалювати свої навички в доступному середовищі [9,10].

Вивчення міжособистісної комунікації ґрунтується на теоріях соціальної взаємодії, які досліджують взаємодію між людьми в різних контекстах [9]. Три основні теорії, а саме теорія соціальної присутності, теорія соціального навчання та теорія комунікативного пристосування, є особливо важливими для з'ясування того, як технології можуть полегшити та покращити міжособистісні навички:

- Теорія соціальної присутності: Розроблена Шортом, Вільямсом і Крісті (1976), досліджує вплив відчуття «присутності» на ефективність комунікації. Коли люди сприймають свого співрозмовника як «присутнього», їхні відповіді стають більш безпосередніми, цікавими та особистими. Таке відчуття присутності особливо важливе в цифровій комунікації, оскільки воно визначає сприйняття близькості та ефективності взаємодії. У голосових системах розробка відповідей, що імітують людську участь і присутність, може допомогти користувачам відчувати зв'язок і розуміння, тим самим підтримуючи практику розмовних навичок у віртуальному середовищі [10].
- Теорія соціального навчання: Запропонована Альбертом Бандурою, стверджує, що люди набувають нових навичок, спостерігаючи та наслідуючи інших, особливо в соціальному контексті. Цифрові інструменти, розроблені з голосовими відповідями, можуть виступати в ролі моделей, демонструючи відповідні моделі розмови, словниковий запас і реакції. Імітуючи взаємодію в реальному житті, ці системи можуть допомогти людям практикувати та засвоювати ефективну комунікативну поведінку, таку як активне слухання та емпатія, які мають вирішальне значення для особистих і професійних стосунків [9].
- Теорія комунікативної акомодатії: Підхід, запропонований Говардом Джайлзом, наголошує на тому, як люди змінюють свою мову, тон голосу і мову тіла, щоб підлаштуватися під свого співрозмовника. У контексті технологічних засобів комунікації адаптивні реакції, які підлаштовуються під тон і стиль користувача, можуть посилити відчуття персоналізації та підтримки у взаємодії. Коли система імітує ці налаштування, вона не тільки створює більш привабливий досвід, але й моделює адаптивні навички спілкування, які користувачі можуть практикувати і застосовувати в реальних ситуаціях [11].

На додаток до соціальних впливів, комунікація також формується під впливом когнітивних та емоційних факторів, які розглядаються через призму психологічних теорій. До них належать теорія когнітивного навантаження, теорія прив'язаності та теорія емоційного інтелекту, які в сукупності дають уявлення про те, як люди беруть участь у змістовних обмінах і як технології можуть підтримати цей процес [12].

Теорія когнітивного навантаження запропонована Джоном Веллером, стосується розумових зусиль, необхідних для обробки інформації та реагування на неї. У контексті комунікації використання складних або незнайомих лінгвістичних конвенцій може спричинити підвищене когнітивне навантаження, тим самим перешкоджаючи безперешкодній взаємодії. Впровадження технологій, які пропонують спрощені та структуровані відповіді, полегшує це когнітивне навантаження, дозволяючи користувачам зосередити свою увагу на розвитку впевненості в розмові та розумінні. Крім того, ці системи можуть сприяти поступовому підвищенню складності відповідно до зростаючого комфорту користувача, тим самим полегшуючи процес поступового розвитку навичок [12].

Теорія прив'язаності розроблена Джоном Боулбі, досліджує вплив ранніх емоційних зв'язків на міжособистісну взаємодію та стосунки. У спілкуванні люди з безпечним стилем прив'язаності, як правило, демонструють більший комфорт у міжособистісній взаємодії. І навпаки, люди з менш безпечним стилем прив'язаності можуть знайти технологію зручною платформою для взаємодії, оскільки вона пропонує середовище з низькими ставками, де вони можуть експериментувати з новими техніками розмови, не боячись осуду. Це може бути особливо корисно для подолання соціальної тривожності та розвитку комфорту в міжособистісному спілкуванні [12].

Теорія емоційного інтелекту популяризована Деніелом Гоулманом, охоплює здатність розпізнавати, розуміти та керувати власними емоціями, а також співпереживати іншим. Емоційний інтелект є вирішальним фактором

ефективної міжособистісної комунікації, оскільки дозволяє людям реагувати з чутливістю та розумінням. Цифрові інструменти, які включають аналіз настроїв та емпатичні реакції, можуть допомогти користувачам практикувати емпатію та емоційну регуляцію, сприяючи кращому усвідомленню емоційних сигналів у спілкуванні [13].

Також відомо, що біологія відіграє невід'ємну роль у нашій здатності взаємодіяти з іншими, впливаючи на нашу здатність сприймати соціальні сигнали та реагувати на них (табл. 1.2) [14]. Нейропластичність і лімбічна система, зокрема, роблять значний внесок у наші комунікативні здібності. Розуміння цих біологічних факторів проливає світло на те, як технології можуть сприяти розвитку комунікативних навичок. Нейропластичність можна визначити як здатність мозку до реорганізації шляхом формування нових нейронних зв'язків. Доведено, що участь у повторюваних змістовних розмовах, навіть за допомогою цифрової системи, може зміцнити нейронні шляхи, пов'язані з обробкою мови, емпатією та соціальною взаємодією. Технології, розроблені для забезпечення позитивного підкріплення, пропонування різноманітних сценаріїв розмов і поступового ускладнення, можуть допомогти користувачам зміцнити ці шляхи, тим самим покращуючи їхню розмовну вільність з часом [15].

Лімбічна система, як сукупність структур у головному мозку, відіграє ключову роль в емоційній регуляції та соціальних зв'язках. Мигдалеподібне тіло, складова лімбічної системи, має особливе значення у виявленні емоційних сигналів і реагуванні на них. Розвиваючи голосову взаємодію, що імітує емоційну реакцію, технології можуть полегшити користувачам здатність розпізнавати та інтерпретувати емоції з більшою точністю. З часом ця практика дозволяє людям розвивати емоційну обізнаність і реагувати на реальні емоційні сигнали з підвищеною чутливістю [16,17].

Дзеркальні нейрони – це клітини мозку, які стають активними, коли людина спостерігає, як інша людина виконує якусь дію, тим самим уможлиблюючи імітацію та емпатію. Цифрові системи, які моделюють розмовну поведінку,

наприклад, модуляцію тону або ввічливі вирази, можуть стимулювати активність дзеркальних нейронів, що, в свою чергу, сприяє емпатії та заохочує користувачів імітувати цю поведінку. Ця здатність має фундаментальне значення для зміцнення соціальних норм і встановлення взаєморозуміння у спілкуванні, оскільки користувачі засвоюють ці моделі через багаторазову взаємодію [18].

Таблиця 1.2 – Огляд теорій комунікації та їх застосування в технологіях [14]

Теоретичний аспект	Теорія	Опис	Застосування
1	2	3	4
Соціальний	Теорія соціальної присутності	Підкреслює важливість відчутної присутності у взаємодії.	Імітація присутності, щоб зробити віртуальну взаємодію більш особистою та цікавою.
	Теорія соціального навчання	Припускає, що навички набуваються через спостереження та імітацію.	Надання розмовних моделей для користувачів для практики ефективної комунікації.
	Теорія комунікативної акомодатії	Фокусується на адаптації мовлення та поведінки, щоб відповідати іншим.	Створення адаптивних відповідей для персоналізованого залучення користувачів.
Психологічний	Теорія когнітивного навантаження	Вивчає розумові зусилля, пов'язані з обробкою інформації.	Пропонує чіткі, структуровані взаємодії для зменшення когнітивного навантаження.

Продовження таблиці 1.2

1	2	3	4
Психологічний	Теорія прив'язаності	Досліджує, як ранні зв'язки впливають на комфорт у взаємодії.	Платформи з низькими ставками дозволяють користувачам будувати довіру в спілкуванні.
	Теорія емоційного інтелекту	Наголошує на розпізнаванні та управлінні емоціями у себе та інших.	Підтримка практики емпатії за допомогою відповідей, адаптованих до емоційних сигналів користувачів.
Біологічний	Нейропластичність	Описує здатність мозку утворювати нові зв'язки через практику.	Повторні взаємодії зміцнюють нейронні шляхи, пов'язані з комунікацією.
	Лімбічна система	Задіяні в емоційній обробці та соціальних зв'язках.	Підвищення емоційної чутливості у взаємодії для імітації реальних життєвих ситуацій.
	Дзеркальні нейрони	Увімкніть імітацію та емпатію, імітуючи спостережувану поведінку.	Моделювання розмовної поведінки для зміцнення соціальних норм та емпатії.

Інтеграція соціальних, психологічних і біологічних теорій в інформаційні технології забезпечує комплексний підхід до розробки комунікаційних інструментів, які покращують міжособистісні навички. Імітуючи ключові

аспекти людської взаємодії, цифрові системи можуть запропонувати користувачам ефективну платформу для практики та розвитку навичок у сприятливому середовищі з низьким рівнем ризику.

### 1.3 Інклюзія у використанні

Використання голосових помічників викликає неоднозначну реакцію у людей різних вікових груп. Для підлітків вони часто слугують засобом розваги та оперативного доступу до інформації. Вони більш схильні досліджувати нові можливості та функції цих помічників. Однією з перешкод для використання цих систем стала глобальна пандемія, спричинена вірусом SARS-CoV-2. Це складні часи, як з економічної точки зору, так і з точки зору глобальної охорони здоров'я, а також з точки зору значних змін у способі життя людей [19,20]. Більшість людей були змушені перейти на дистанційне навчання та роботу, що також вплинуло на дітей [21]. У цей період зросла зацікавленість дітей технологіями, зокрема голосовими помічниками [22–24].

На робочому місці дорослі користуються голосовими помічниками, щоб підвищити ефективність роботи, полегшити планування та організацію завдань, а також спростити різні процеси. Високо цінується простота використання та ефективність виконання завдань [25]. Як наслідок, у секторі роздрібно́ї торгівлі спостерігається поширення голосових помічників, які стають вирішальним елементом у покращенні споживчого досвіду [26]. Впровадження голосових помічників у роздрібній торгівлі має на меті надати споживачам додаткову підтримку протягом усього процесу здійснення покупок. Такі пристрої надають споживачеві можливість отримувати інформацію про товари, порівнювати ціни, здійснювати покупки або замовлення онлайн, і все це лише за допомогою голосових команд. Очевидно, що допомога, надана колегою або голосовим помічником, може викликати чіткі почуття задоволення і позитивного впливу. Коли колега пропонує допомогу, це може зміцнити міжособистісні зв'язки і сприяти взаєморозумінню в команді. Голосовий асистент, надаючи швидку і

точну інформацію, сприяє швидкому виконанню завдань, підвищуючи особистий комфорт і продуктивність. Обидва сценарії можуть викликати позитивні емоції та сприяти загальному задоволенню роботою або повсякденними завданнями [27].

Люди похилого віку сприймають голосових помічників як засіб, що покращує їхнє розуміння технологій. Голосові помічники можуть полегшити виконання повсякденних завдань для людей похилого віку, хоча їхнє особливе сприйняття технологій іноді може перешкоджати їхньому використанню. Тим не менш, вони можуть бути корисними у виконанні повсякденних завдань та отриманні необхідної інформації [28]. Люди похилого віку також потребують допомоги зі здоров'ям, що вимагає своєчасної, простої та нетривіальної допомоги. Для цього добре підходять голосові асистенти, які можуть аналізувати стан здоров'я користувача на основі звукових хвиль [29].

Тим не менш, очевидно, що не всі люди користуються голосовими помічниками на постійній основі. Мотивація до використання голосових помічників походить з багатьох джерел і залежить від низки факторів [30,31]. По-перше, вони стають невід'ємним компонентом нашого сучасного світу, що швидко змінюється, де ефективність і швидкий пошук інформації мають першорядне значення. Здатність голосових помічників надавати миттєвий доступ до інформації полегшує пошук потрібних даних і виконання завдань. Крім того, здатність цих технологій спрощувати повсякденну діяльність і підвищувати зручність сприяє їхній зростаючій популярності. Голосові асистенти стають все більш популярними для виконання різноманітних завдань, включно зі складанням розкладу та замовленням товарів. Зручність та адаптивність до вподобань користувачів роблять їх привабливими для широких верств населення, що сприяє їхній інтеграції в повсякденне життя [23,32,33].

Універсальність та ефективність голосових помічників у виконанні завдань є вагомим фактором їхньої популярності. Для багатьох вони стали надійним інструментом для спрощення повсякденних завдань і прискорення рутинних



процесів, пов'язаних з пошуком інформації або операціями [34]. Однак деякі користувачі ставляться до голосових помічників з обережністю або негативно через побоювання, пов'язані з конфіденційністю даних і потенційними ризиками їх розголошення [35]. Тому важливо розуміти, що точність розпізнавання голосових команд або відповідей може бути обмеженою.

Етичні наслідки використання голосових помічників стають дедалі актуальнішими у світлі їхньої дедалі більшої поширеності. Однією з найважливіших проблем є питання алгоритмічної упередженості та справедливості. Алгоритми, які лежать в основі голосових помічників, можуть містити упередження або відображати дискримінацію, що виникає через неналежне навчання моделей на неправильних або неусвідомлених даних. Це може призвести до некоректної обробки запитів користувачів або надання упереджених відповідей, що порушує принципи справедливості та рівності. Важливо також підкреслити, що етичні проблеми поширюються і на прозорість того, як голосові помічники збирають і використовують дані користувачів [36].

#### **1.4 Фітнес помічник**

Поява голосових систем на основі штучного інтелекту призвела до революційних змін у фітнес-індустрії, зробивши персоналізовані тренування доступними для широкого кола користувачів. У сфері фітнесу ці системи виконують функцію тренера, пропонуючи вказівки, зворотний зв'язок у режимі реального часу і мотивацію за допомогою обробки природної мови, забезпечуючи таким чином діалоговий і підтримуючий досвід. Поява голосових помічників не тільки зробила фітнес-підтримку більш доступною, але й змінила підхід до фізичних вправ, особливо для тих, хто вважає за краще тренуватися вдома або не має можливості робити це з тренером. Голосова система на основі штучного інтелекту пропонує динамічний і зручний підхід до фітнесу. Вона реагує на голосові команди, надає інструкції та адаптує рекомендації на основі зворотного зв'язку з користувачем. Такі системи призначені для допомоги

користувачам у виконанні тренувань, моніторингу ефективності та надання негайного зворотного зв'язку – все це за допомогою простої голосової взаємодії. Ця можливість особливо корисна для користувачів, яким може бути складно самостійно орієнтуватися в складних фітнес-програмах [37].

Голосові фітнес-системи дуже ефективно надають вказівки в режимі реального часу під час тренувань, забезпечуючи безпечне та ефективне виконання вправ [12]. Ці системи часто включають в себе датчики або поєднуються з пристроями, здатними відстежувати рух, такими як смартфони, фітнес-браслети або смарт-колонки, оснащені акселерометрами. Аналізуючи дані користувача, голосовий асистент може надавати негайні вказівки щодо форми і техніки виконання вправ, тим самим знижуючи ризик травм і максимізуючи ефективність кожної вправи. Наприклад, під час випадку або присідання голосовий асистент може надавати вказівки, пов'язані з формою, такі як «Тримайте спину прямо» або «Переконайтеся, що коліна не виходять за пальці ніг», допомагаючи користувачеві зробити необхідні корективи, щоб запобігти перенапруженню. Ці нагадування особливо корисні для початківців або людей, які тренуються самостійно і можуть не знати про свою поставу або техніку виконання вправ.

Однією з найпомітніших характеристик голосових систем на основі штучного інтелекту у сфері фітнесу є їхня здатність персоналізувати режими тренувань відповідно до конкретного рівня фізичної підготовки, особистих уподобань та бажаних результатів [2]. Голосовий асистент здатний генерувати індивідуальний план на основі аналізу даних користувача, включаючи вік, тип фігури і фітнес-цілі. У міру того, як користувач прогресує, система здатна безперервно коригувати програми, щоб збільшити інтенсивність або змінити вправи на основі результатів, відгуків та будь-яких показників, які користувач повідомляє самостійно, таких як вага або рівень енергії. Персоналізований план може включати різну тривалість тренувань, типи вправ і рівні складності. Наприклад, користувач, який займається силовими тренуваннями, може

отримати індивідуальну комбінацію вправ з власною вагою та вправ з обтяженнями, тоді як людині, яка прагне розвинути витривалість, може бути запланований режим, що складається з кардіотренувань з поступовим збільшенням інтенсивності. Під час кожного заняття голосовий асистент може надавати нагадування на кшталт «Ви досягли середини шляху до своєї тижневої мети» або «У вас є ще один підйом, щоб виконати сьогоднішню ціль», які сприяють почуттю досягнення і мотивують користувачів не зупинятися на досягнутому [38].

Підтримка постійної мотивації є складним аспектом регулярних тренувань, особливо при самотійних спробах. Голосовий асистент забезпечує мотивацію та залучення завдяки своєчасному заохоченню, персоналізованому зворотному зв'язку та цілеспрямованим завданням, перетворюючи фітнес із самотнього заняття на інтерактивний досвід [39]. Це досягається кількома способами:

- Позитивне підкріплення та заохочення: Голосові фітнес-помічники часто використовують позитивне підкріплення, щоб мотивувати користувачів під час тренувань. Після складного тренування асистент може запропонувати похвалу, наприклад, «Гарна робота на цьому підході!» або «Ви робите чудовий прогрес!». Надання позитивного зворотного зв'язку підвищує впевненість користувача в собі і збільшує задоволення від тренування, що, в свою чергу, сприяє довгостроковій прихильності до фітнесу.
- Підзвітність та відстеження прогресу: Моніторинг прогресу є фундаментальним аспектом фітнесу, оскільки він дозволяє користувачам залишатися в курсі своїх досягнень і областей, які потребують вдосконалення. Системи, що реагують на голос, особливо добре записують дані, пов'язані з тренуваннями, зокрема кількість виконаних підходів, час, витрачений на виконання кожного підходу, кількість виконаних повторень і кількість спалених калорій. Надання цієї інформації зміцнює прихильність користувачів до свого фітнес-режиму і заохочує їх дотримуватися послідовності. Наприклад, голосовий асистент може

сказати: «Цього тижня ви виконали три тренування, що означає значний прогрес у досягненні тижневої мети!» або «Вам залишилося лише одне тренування до досягнення місячної мети». Такий підхід виховує почуття відповідальності та зв'язку з фітнес-цілями, тим самим посилюючи почуття досягнення і прихильності до своїх рутинних занять.

- Постановка цілей та виклики: Ще одна важлива функція, яка слугує для підтримання залученості користувача у його фітнес-подорожі. Голосовий асистент може допомогти користувачам у встановленні досяжних короткострокових і довгострокових цілей, які можуть бути засновані на вазі, результативності або часі. Такі завдання, як «Виконати п'ять тренувань цього тижня» або «Збільшити час виконання планки на 30 секунд до кінця місяця», дають користувачам чіткі цілі, над досягненням яких потрібно працювати. Деякі вдосконалені голосові системи використовують методи гейміфікації, додаючи елемент задоволення у фітнес-подорож, пропонуючи користувачам віртуальні бейджики, бали або графіки прогресу в знак визнання досягнення цих цілей. Ці стимули можуть мати значний вплив на мотивацію користувача, додаючи азарту і відчутної винагороди під час тренувань.

Ефективність голосових систем у сфері фітнесу залежить від інтеграції декількох базових технологій штучного інтелекту, включаючи обробку природної мови, машинне навчання та аналіз даних [40,41]. У таблиці 1.3 розглянуто конкретний внесок кожної з цих технологій у покращення фітнес-досвіду.

Таблиця 1.3 – Внесок технологій [40,41]

Технологія	Функція у фітнес-консультації	Приклади застосування
1	2	3
Обробка природної мови	Перекладає голосові команди, запити та відгуки користувачів.	Отримує команди тренування, відповідає на запитання користувача та надає мотиваційні підказки.

Продовження таблиці 1.3

1	2	3
Машинне навчання	Адаптує тренувальні програми та відгуки на основі прогресу та вподобань користувача.	Налаштовує плани тренувань, визначає бажані стилі тренувань і відстежує цілі.
Аналітика даних	Аналізує метрики тренувань, щоб дати уявлення про продуктивність і прогрес.	Відстежує спалені калорії, відстежує завершені тренування, налаштовує рутинні програми для прогресу.

Поєднання згаданих технологій призводить до створення системи, яка здатна розуміти, навчатися та адаптуватися до конкретного фітнес-подорожі кожного користувача. Це дозволяє надавати рекомендації, які сприймаються як інтуїтивно зрозумілі та персоналізовані [41,42].

Голосові системи у фітнесі можуть застосовуватися в широкому діапазоні вправ, від серцево-судинних до силових, що робить їх універсальними інструментами для досягнення різних фітнес-цілей:

- Що стосується серцево-судинних вправ, асистент може надавати користувачам підказки щодо темпу та дихання, допомагаючи їм досягти оптимальної продуктивності. Наприклад, під час бігу асистент може запропонувати збільшити темп на наступну хвилину або, навпаки, підтримувати постійний ритм, щоб зберегти енергію.
- Що стосується силових тренувань, голосовий асистент може допомагати користувачам підтримувати належну форму і пропонувати інтервали відпочинку. Наприклад, він може надавати вказівки на кшталт «Під час цього підйому вмикайте м'язи пресу» або «Відпочиньте 30 секунд перед наступним підходом», що імітує допоміжні вказівки, які зазвичай надає персональний тренер у реальному часі.
- Система розроблена для полегшення тренування гнучкості та мобільності. Вона надає вербальні підказки, наприклад, інструкції утримувати розтяжку

протягом 20 секунд або глибоко дихати і розслабитися під час розтяжки, що підвищує здатність користувача виконувати ці рухи безпечно і ефективно. Система також допомагає користувачам виконувати вправи на розтяжку, пози йоги та вправи на рухливість.

Окрім підвищення ефективності тренувань, голосові фітнес-системи надають психологічні переваги, які сприяють довготривалому дотриманню фітнес-режимів. Вони сприяють формуванню звичок, розвивають позитивне мислення і роблять фітнес доступним для широких верств населення, надаючи експертні рекомендації без необхідності залучення фізичного тренера. Для тих, хто прагне досягти успіху у фітнесі, дуже важливо встановити послідовність у своїй рутині. ГП може допомогти користувачам у цьому, оскільки він дозволяє їм створювати і дотримуватися регулярного розкладу. Нагадування і можливості планування тренувань допомагають тримати користувачів підзвітними, заохочуючи їх інтегрувати фізичні вправи у своє повсякденне життя. Наприклад, щоденні нагадування на кшталт «Готові до сьогоднішнього заняття?» або «Пам'ятайте, що для досягнення прогресу достатньо 20-хвилинного тренування» сприяють формуванню звички до регулярних занять фітнесом. Для людей, які можуть відчувати дискомфорт у спортзалі або не мають доступу до нього, голосова фітнес-система забезпечує інклюзивне, приватне і сприятливе середовище. Дозволяючи користувачам займатися вдома або за власним розкладом, ці системи розширюють спектр доступних можливостей для фітнесу, підвищуючи загальну доступність фізичного здоров'я.

Оскільки голосова технологія штучного інтелекту продовжує розвиватися, майбутні вдосконалення можуть ще більше оптимізувати взаємодію з користувачами та результати [43]:

- Покращене розпізнавання емоцій: З появою нових розробок в обробці природної мови перспективні системи можуть бути здатні розпізнавати емоційні нюанси в голосі користувача, тим самим дозволяючи

віртуальному помічнику модулювати тон і відповіді таким чином, щоб викликати співчуття і підтримку під час складних взаємодій.

- Покращена персоналізація завдяки інтеграції з натільними пристроями: Поєднання голосових систем з носяться технологіями полегшує надання зворотного зв'язку в режимі реального часу за такими показниками, як частота серцевих скорочень, артеріальний тиск і рівень навантажень, тим самим підвищуючи точність рекомендацій, що надаються користувачеві відповідно до його фізичного стану.
- Потенціал майбутніх голосових систем, що реагують на голос, включатимуть соціальні функції є сферою, що викликає значний інтерес. Такі функції могли б полегшити участь користувачів у групових змаганнях, дозволити їм ділитися прогресом з друзями і полегшити взаємодію зі спільнотою користувачів. Це покращить соціальний вимір до самостійних занять фітнесом.

Згадані аспекти були випробувані на Міжнародній науково-практичній конференції «Інформаційні технології і автоматизація – 2024» [44]. В рамках дослідження було опитано 47 респондентів щодо залучення можливої системи. Дослідження охоплювали низку питань, таких як «Види спорту, якими займаються респонденти», «Тренуєтесь з тренером чи ні», «Використовуєте додатки» та інші (рис. 1.2 – рис. 1.8).

Чим саме ви займаєтесь?

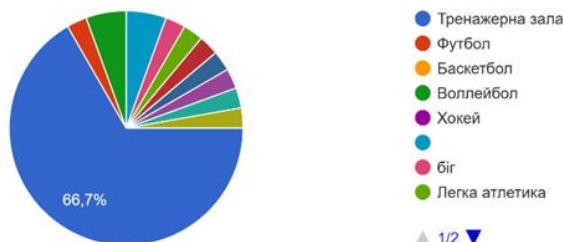


Рисунок 1.2 – Аналіз діяльності

Чи займаєтесь ви з тренером?

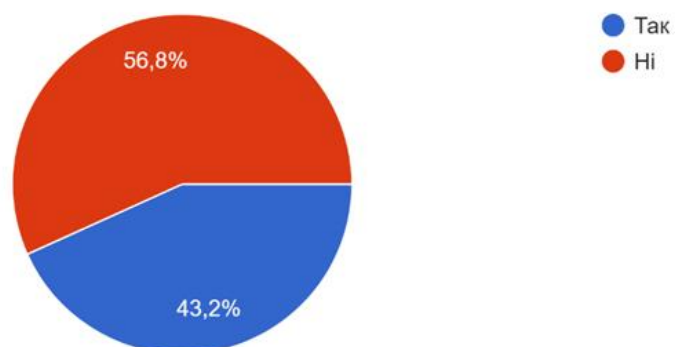


Рисунок 1.3 – Аналіз використання тренера

Чи користуєтесь спортивними добавками?

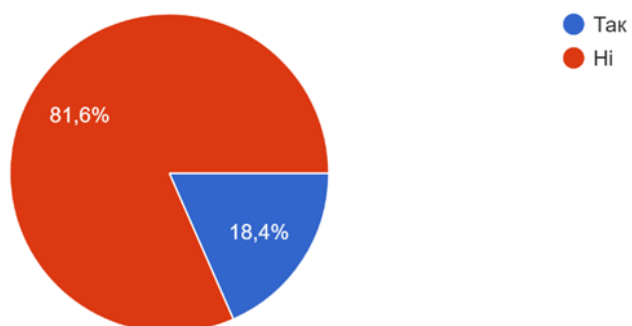


Рисунок 1.4 – Аналіз використання добавків

Чи використовували б ви інтегрованого помічника для своїх спортивних цілей?

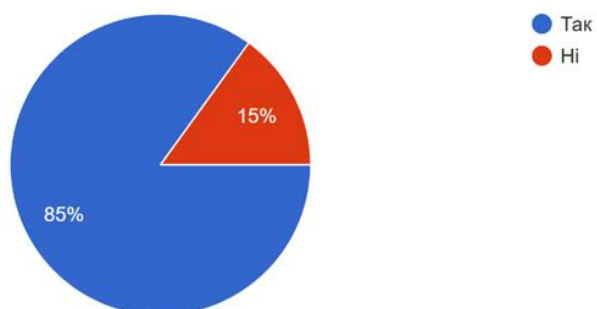


Рисунок 1.5 – Аналіз використання ГП



Чи були б ви готові платити підписку за цей сервіс?

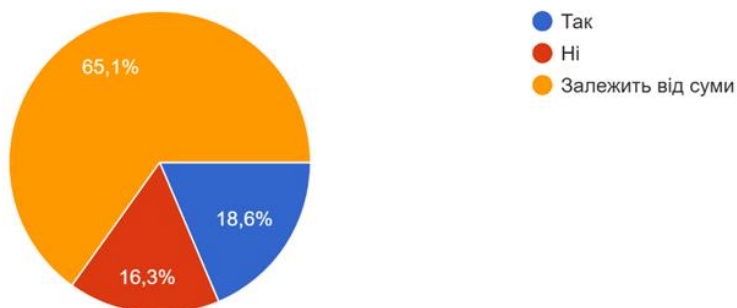


Рисунок 1.6 – Аналіз сервісного збору

Кому б ви віддали перевагу?

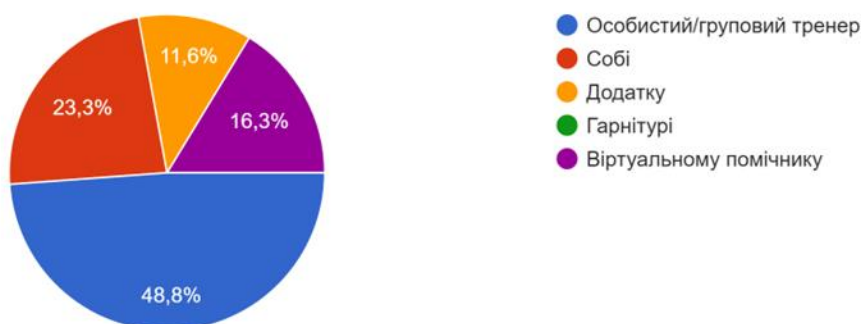


Рисунок 1.7 – Аналіз вподобань

Які запитання вас турбують стосовно "віртуального помічника"

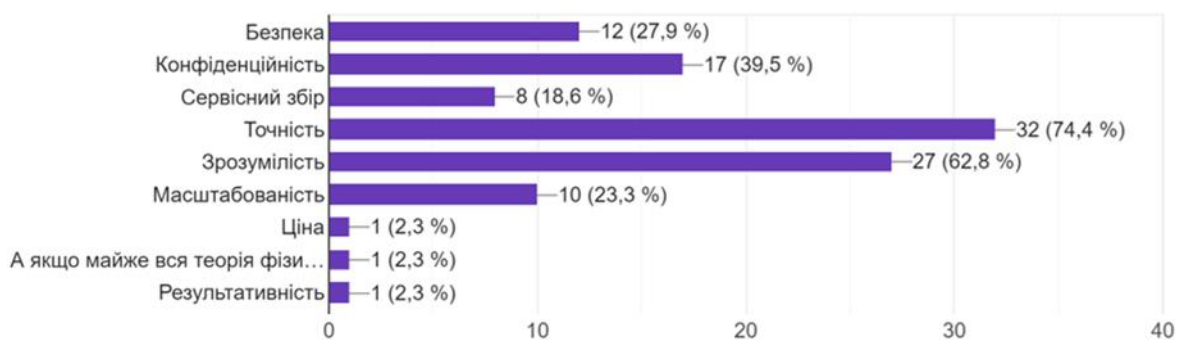


Рисунок 1.8 – Аналіз питань

Відповідно до наведених діаграм та експертизи, можна виокремити головні тези. Користувачі звикли до традиційного підходу через використання особистого тренера. Така поведінка не дивна. Проте, звернувши увагу на рисунок 1.6 та рисунок 1.8, можна помітити, що користувачеві все ж таки цікавить можливість використання ГП. Особливо, якщо розкрити відповіді на питання котрі зазначені на рисунку 1.8, дозволить підвищити популярність використання даного додатку в сфері фітнесу. Доречно буде зазначити, що ті ж самі питання породжують теми для покращення даного дослідження та спонукання до розвитку майбутніх робіт з даної сфери.

### **1.5 Способи реалізації**

Створення голосового асистента вимагає міждисциплінарного підходу, що поєднує штучний інтелект, розробку програмного забезпечення, дизайн користувацького досвіду та обробку природної мови. Основна мета такого асистента – розуміти, обробляти і реагувати на введення користувача точно і природньо, що вимагає складної технічної бази і ретельно розроблених інтерфейсів [45].

Важливим компонентом будь-якої системи з підтримкою мовлення є добре продуманий інтерфейс, який забезпечує безперебійну, інтуїтивно зрозумілу та зручну для користувача роботу. Дизайн інтерфейсу, як візуальний, так і звуковий, відіграє важливу роль у тому, як користувачі взаємодіють з асистентом, впливаючи на зручність використання, доступність і загальну задоволеність. На початкових етапах розробки дизайн користувацького досвіду (UX) в першу чергу спрямований на створення безперешкодного розмовного процесу. Враховуючи значну залежність голосових помічників від аудіовзаємодії, першочерговим завданням є розробка розмовного UX (табл. 1.4). Розглянемо деталі створення:

- Відображення процесів у вигляді діалогу: Цей процес передбачає розробку потенційних вхідних даних користувача і відповідних реакцій системи. Програмні інструменти, такі як Voiceflow і Dialogflow, можуть бути

використані для розробки розмовних потоків з умовними відповідями, тим самим покращуючи користувацький досвід завдяки більш природній взаємодії.

- У контексті користувацьких інтерфейсів візуальні елементи мають особливе значення в додатках, які мають візуальний компонент, таких як мобільне або комп'ютерне програмне забезпечення. Дизайн іконок, бульбашок відгуку та анімації слід ретельно продумувати, оскільки вони надають користувачам візуальний зворотній зв'язок. Популярними інструментами для розробки цих елементів є Figma, Sketch та Adobe XD.

Простий користувацький інтерфейс може включати анімований значок мікрофона, який вказує на те, що асистент слухає, або голосові бульбашки, що відображають відповіді асистента в текстовій формі. Крім того, візуальні підказки допомагають користувачам визначити, коли асистент обробляє інформацію або відповідає, тим самим підвищуючи взаємодію.

Голосовий інтерфейс користувача є основою будь-якого голосового асистента. Інструменти для розробки інтерфейсу, такі як Amazon Alexa Skills Kit або Google Dialogflow, допомагають розробникам проєктувати взаємодію на основі обробки природної мови. Ці інструменти пропонують інтегровані можливості розпізнавання та синтезу мови, забезпечуючи необхідну функціональність для обробки голосових вхідних і вихідних даних.

Таблиця 1.4 – Порівняння інструментів UX

<b>Інструмент</b>	<b>Мета</b>	<b>Використання</b>
<b>1</b>	<b>2</b>	<b>3</b>
Voiceflow	Розмовний дизайн і картування потоків	Створення складних діалогових потоків і сценаріїв тестування
Dialogflow	Розуміння природної мови	Побудова взаємодії на основі «розуміння» для природних розмов

## Продовження таблиці 1.4

1	2	3
Adobe XD / Figma	Дизайн інтерфейсу для візуальних компонентів	Розробка візуальних аспектів, таких як кнопки та анімація зворотного зв'язку

Роль мов програмування у створенні голосових асистентів має першорядне значення, починаючи від розробки бекенду і закінчуючи інтеграцією фронтенду. Серед найпоширеніших мов – Python, JavaScript і Java, кожна з яких має свої переваги в реалізації конкретних функцій голосових асистентів (табл. 1.5).

Python є найкращою мовою для розробки голосових помічників завдяки своїй простоті, універсальності та великим бібліотекам. Такі бібліотеки, як NLTK, SpaCy та Transformers від Hugging Face, надають розширені функції обробки природної мови, що дозволяє розробникам реалізовувати складні мовні моделі для розуміння та генерування відповідей, подібних до людських. Бібліотеки машинного навчання Python, включаючи TensorFlow і PyTorch, полегшують створення та інтеграцію моделей глибокого навчання, які мають вирішальне значення для здатності асистента розуміти і генерувати відповіді.

JavaScript, зокрема у формі Node.js, є загальноживаною мовою для розробки голосових помічників, які працюють на вебплатформах. Node.js надає асинхронні можливості, які полегшують обробку декількох запитів користувачів одночасно. Крім того, вона дозволяє легко інтегруватися з елементами інтерфейсу, що робить її доречним вибором для створення крос-платформних голосових помічників, які функціонують у вебдодатках.

Java – ще одна мова, яка використовується для розробки голосових помічників, зокрема для додатків для Android. Враховуючи широке використання Android, сумісність Java з екосистемою Android дозволяє розробникам розгорнути голосові помічники як мобільні додатки, тим самим розширюючи їх охоплення та функціональність.

Таблиця 1.5 – Порівняння мов програмування

Мова	Функціональність	Бібліотеки/фреймворки
Python	Інтеграція моделей обробки природної мови та машинного навчання	NLTK, SpaCy, TensorFlow, PyTorch
JavaScript (Node.js)	Вебінтеграція та реагування в режимі реального часу	Express.js, Watson SDK, Dialogflow API
Java	Додатки для голосових помічників на базі Android	Android SDK, Google Assistant API

Моделі ШІ є основним компонентом будь-якого голосового асистента, полегшуючи інтерпретацію, обробку та генерування відповідей [5,45]. Голосові помічники використовують моделі обробки природної мови, розпізнавання мови та синтез мови, щоб розуміти розмовну мову і надавати зв'язні, схожі на людські відповіді. Обробка природної мови є основним елементом у розробці голосових помічників, що дозволяє системі обробляти людську мову і надавати зрозумілі відповіді (табл. 1.6) [46]. Найпоширеніші моделі ОПМ включають:

- Модель Bidirectional Encoder Representations from Transformers (BERT) була розроблена компанією Google і широко використовується завдяки здатності розуміти контекст слів у реченні. Це дозволяє асистенту інтерпретувати складні запити з високим ступенем точності [47].
- Generative Pre-trained Transformer (GPT), розроблений OpenAI, являє собою передову модель генерації тексту з людськими характеристиками. Вона дозволяє асистенту генерувати відповіді, які є одночасно зв'язними і розмовними, тим самим підвищуючи природність взаємодії.
- Rasa – це фреймворк машинного навчання з відкритим вихідним кодом, спеціально розроблений для розмовного ШІ. Він підходить для побудови спеціальних моделей ОПМ, пристосованих до конкретних випадків використання. Це дозволяє тонко налаштовувати моделі для

підвищення точності голосових помічників.

Щоб забезпечити точну обробку голосового введення, голосові помічники використовують технології автоматичного розпізнавання мови (АРМ) і перетворення тексту в мову (ПТМ). Інтерфейси прикладного програмування (API) Google Speech-to-Text і Text-to-Speech забезпечують високоякісне АРМ і ПТМ, дозволяючи цифровим асистентам транскрибувати вимовлені слова в текст і перетворювати текст назад в синтезовану мову [48]. Amazon Polly – це сервіс перетворення тексту, який переводить текст на реалістичну мову. Це дозволяє розробникам налаштовувати голос помічника для різних мов і акцентів. Microsoft Azure Cognitive Services – це низка пропозицій АРМ та ПТМ від Microsoft. Вони надають надійні інструменти для розпізнавання та синтезу мовлення, які інтегруються з численними платформами.

Таблиця 1.6 – Порівняння моделей ШІ [46]

<b>Модель/Сервіс</b>	<b>Функціональність</b>	<b>Особливості</b>
BERT	Контекстуальне розуміння обробки природньої мови	Висока точність для розуміння мови
GPT	Генерування діалогових відповідей	Створює текстові відповіді, схожі на людські
Google Speech-to-Text	Розпізнавання мови	Транскрипція в режимі реального часу, багатомовна підтримка
Amazon Polly	Перетворення тексту в мовлення	Природне звучання, настроюванні голоси

На додаток до своїх основних функцій, голосові асистенти здатні надавати цілий ряд додаткових можливостей, які можна використовувати для підвищення залученості користувачів і надання їм більш універсального досвіду. Контекстна пам'ять дозволяє асистенту пригадувати нещодавні розмови або вподобання користувача, тим самим сприяючи більш персоналізованій взаємодії. Модулі

пам'яті можуть бути побудовані з використанням баз даних, таких як MongoDB або Redis, що дозволяє асистенту зберігати інформацію тимчасово або постійно, відповідно до дозволів користувача та налаштувань конфіденційності даних. Щоб задовольнити різноманітні потреби користувачів, багато голосових помічників включають мультимодальну взаємодію, коли звукові підказки поєднуються з візуальною інформацією. Це особливо корисно в сценаріях, де візуальний зворотний зв'язок може доповнювати слухові інструкції, наприклад, навігацію або вказівки щодо складних завдань. Мультимодальні фреймворки, такі як Google's Actions від Google, дозволяють розробникам створювати асистентів, які взаємодіють з голосом, текстом і екраном. Поява штучного інтелекту прискорила створення систем, здатних розпізнавати емоційні стани, що дозволяє цифровим асистентам оцінювати тон користувача і відповідно змінювати свої реакції. Розвиток емоційного інтелекту часто спирається на аналіз настроїв і розпізнавання тону, що дозволяє цифровим асистентам надавати емпатичні відповіді.

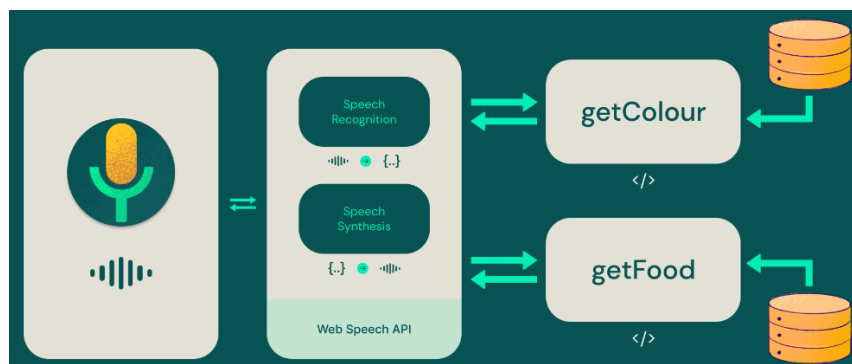


Рисунок 1.9 – Візуалізація залучення інструментів [49]

Розробка голосового асистента на основі штучного інтелекту вимагає залучення достатньої кількості інструментів, технологій і принципів дизайну, щоб гарантувати його ефективність, зручність і здатність розуміти і взаємодіяти з користувачами (рис. 1.9). Від початкової концептуалізації компонентів користувацького інтерфейсу до вибору мов програмування та моделей штучного інтелекту – кожен елемент ретельно підбирається, щоб підвищити здатність асистента обробляти людську мову та реагувати на неї.

## 1.6 Постановка задачі

Метою цієї роботи є розробка інформаційної системи, керованої штучним інтелектом, яка заохочує формуванню навичок міжособистісної комунікації. Аби досягти відповідного результату, потрібно дотриматись наступних кроків:

- 1) розробити дизайн, який орієнтований на досвід користувача і спрямований на створення інтуїтивно зрозумілого і привабливого інтерфейсу, що сприяє безперешкодній взаємодії;
- 2) визначити функціональні вимоги, що окреслюють основні операції, які система повинна виконувати, включаючи обробку команд, генерування відповідей і набуття знань через взаємодію;
- 3) визначити нефункціональні вимоги, які гарантують, що система володіє необхідними якостями, такими як швидкість, надійність і простота використання;
- 4) розробити інтерфейс, що включає в себе створення інтерактивних компонентів, які слугують для покращення користувацького досвіду та полегшення комунікації.;
- 5) обирати мову програмування (як Python або JavaScript), через її можливості штучного інтелекту та потужну бібліотечну підтримку;
- 6) визначення відповідних бібліотек;
- 7) вибір моделі ШІ, яка гарантує, що система здатна точно інтерпретувати мову користувача і реагувати на неї;
- 8) вирішити питання безпеки даних

Завдяки ретельному виконанню кожного з цих етапів буде створено адаптивну, безпечну та зручну для користувача систему, здатну сприяти формуванню навичок міжособистісної взаємодії.



## 2 ВИБІР МЕТОДІВ РЕАЛІЗАЦІЇ

### 2.1 Початкові кроки

При розробці системи зі штучним інтелектом, яка покращує взаємодію з користувачем, вкрай важливим є структурований, багатоетапний підхід (рис. 2.1). Процес починається з етапу проєктування, на якому потреби та вподобання користувачів формують основу адаптивного та інтуїтивно зрозумілого інтерфейсу. Після етапу проєктування реалізація інтерфейсу передбачає створення візуальних та інтерактивних елементів, які полегшують комунікацію. Сюди входить включення аудіо-візуального зворотного зв'язку, що сприяє безперешкодному користувацькому досвіду. Етап написання коду передбачає розробку можливостей розпізнавання мови, що дозволяє системі обробляти і реагувати на усні вхідні дані. Згодом створюється функціонал голосових команд, що дозволяє системі інтерпретувати та виконувати конкретні голосові вказівки. Крім того, розробляється функція реагування на ключові слова, що дозволяє системі виявляти ключові терміни і діяти на них, що підвищує залученість користувачів. Інтеграція моделі штучного інтелекту, наприклад, моделі обробки природної мови, наділяє систему здатністю розуміти і генерувати відповіді, аналогічні людським. За необхідності, розпізнавання обличчя також може бути включене для подальшої персоналізації взаємодії, розпізнавання окремих користувачів і пропонування індивідуальних відповідей. У сукупності ці етапи призводять до створення надійної інтерактивної системи, яка здатна обробляти складні взаємодії з приємним інтерфейсом.

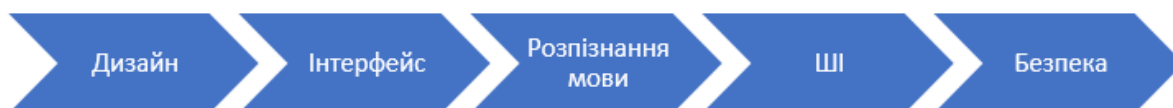


Рисунок 2.1 – Процеси, потрібні для створення ГП

## 2.2 Інструменти для розробки

Створення голосового помічника, з використанням штучного інтелекту, вимагає використання багатогранного набору інструментів, кожен з яких виконує свою роль протягом усього процесу розробки. Figma використовується на етапі проектування, дозволяючи розробникам створювати візуально привабливий та інтуїтивно зрозумілий інтерфейс користувача [45]. Тим часом HTML і CSS забезпечують структуру і стилізацію, необхідні для веброзробки інтерфейсу. JavaScript (JS) дозволяє вбудовувати в інтерфейс динамічний контент і адаптивні взаємодії з користувачем, тим самим підвищуючи інтерактивність системи.

З точки зору внутрішньої частини, Python є кращою мовою програмування завдяки своїй універсальності та великим бібліотекам, включаючи EEL, яка слугує мостом між Python та вебтехнологіями, що дозволяє додаткам на Python мати користувацькі інтерфейси на основі HTML/JS [50,51]. OpenCV полегшує інтеграцію можливостей комп'ютерного зору, таких як розпізнавання зображень або облич, що може покращити персоналізацію та функції безпеки [52]. Для того, щоб полегшити інтеграцію функцій ШІ, бібліотека «Hugging Face» пропонує низку заздалегідь підготовлених мовних моделей, що полегшує інтеграцію складних функцій ОПМ [53]. SQLite3, з іншого боку, слугує легким, але надійним рішенням для баз даних, ефективно керуючи даними користувача та конфігураціями системи [54]. Коли ці інструменти використовуються разом, вони створюють надійну основу для розробки гнучких та інтерактивних систем штучного інтелекту, здатних вирішувати складні завдання, зберігаючи при цьому дружній інтерфейс.

## 2.3 Нефункціональні вимоги

При розробці ГП слід попередньо визначити нефункціональні вимоги, оскільки вони гарантують ефективність, безпеку і задоволеність користувачів. Заходи безпеки вживаються для захисту системи від несанкціонованого доступу, тим самим забезпечуючи захист даних користувачів і взаємодії [55]. Вимоги до

продуктивності гарантують, що система працює ефективно і якісно при очікуваному користувацькому навантаженні без зниження продуктивності, тим самим підтримуючи безперебійну роботу навіть за наявності високих обсягів трафіку. Крім того, система повинна бути спроектована таким чином, щоб її можна було масштабувати, тобто розширювати її можливості за потреби. Доступність є фундаментальною вимогою, яка гарантує, що система буде доступна для користувачів у будь-який час. Крім того, обслуговування є пріоритетом, що гарантує, що система легко оновлюється, легко усуває несправності і може бути вдосконалена з плином часу.

Переносимість дозволяє використовувати систему на різних платформах з мінімальними змінами, тим самим підвищуючи її доступність. Надійність має першорядне значення для послідовного виконання вимог користувачів і підтримки довіри до функціональності системи. Концепція юзабіліті охоплює розробку інтуїтивно зрозумілої системи, яка дозволяє користувачам легко орієнтуватися в ній і взаємодіяти з нею. Для забезпечення сумісності система повинна легко інтегруватися з іншими технологіями або інструментами, з якими вона може взаємодіяти. У сукупності ці нефункціональні вимоги слугують для посилення стійкості системи, покращення користувацького досвіду та адаптивності.

Розглянемо визначені нефункціональні вимоги до майбутньої системи:

- що стосується безпеки, то повинні бути виконані наступні вимоги: необхідно, щоб система використовувала надійні протоколи аутентифікації та шифрування для запобігання несанкціонованому доступу та захисту даних користувачів.
- з точки зору продуктивності, система повинна бути здатна обробляти очікуване навантаження користувачів і надавати відповіді протягом визначеного часу, з мінімальною затримкою, незалежно від обсягу трафіку. Важливо, щоб система ефективно працювала при очікуваному навантаженні користувачів, надаючи відповіді в зазначені терміни з

- мінімальною затримкою, незалежно від обсягу трафіку.
- Система повинна бути спроектована таким чином, щоб підтримувати масштабовану інфраструктуру, що дозволить їй динамічно розширювати або зменшувати пропускну здатність, таким чином дозволяючи їй пристосовуватися до різної кількості користувачів та операційних вимог.
  - Система повинна бути доступною для використання в будь-який час, щоб забезпечити постійну доступність для користувачів. Важливо, щоб система підтримувала цільовий рівень безвідмовної роботи на рівні 99,9% або вище для забезпечення безперебійної доступності.
  - Система повинна бути спроектована таким чином, щоб полегшити її обслуговування. Кодова база та архітектура системи повинні бути модульними та добре задокументованими, що сприятиме простому оновленню, виправленню помилок та розширенню функцій без необхідності значних переробок.
  - Переносимість визначається як можливість перенесення системи з одного середовища в інше без значних змін. Система повинна бути розроблена таким чином, щоб її можна було розгортати на різних платформах (наприклад, Windows, macOS, Linux) з мінімальними змінами, тим самим забезпечуючи гнучкість розгортання.
  - Система повинна бути надійною, для постійно надання точних та надійних результатів, тим самим мінімізуючи системні помилки та час простою для збереження довіри користувачів.
  - Система повинна мати інтуїтивно зрозумілий користувацький інтерфейс і просту навігацію, що дозволить користувачам швидко освоїти і використовувати її з мінімальною підготовкою.
  - Сумісність: Система повинна підтримувати інтеграцію з іншими програмними та апаратними платформами, включаючи сумісність з API, базами даних та сторонніми сервісами.

## 2.4 Функціональні вимоги

Функціональні вимоги до голосового асистента визначають конкретні можливості, які він повинен мати, щоб забезпечити безперебійну та ефективну роботу користувача. ГП повинен бути оснащений високоточним розпізнаванням мови, що дозволяє надійно фіксувати і перетворювати усні команди в текст [55].

Використання обробки природної мови дозволяє пристрою розуміти та інтерпретувати наміри користувача, полегшуючи багатомовність, коли це необхідно. Основні функціональні можливості включають виконання голосових команд для таких завдань, як встановлення нагадувань, відтворення музики або управління пристроями розумного будинку. Кожна команда виконується з мінімальною затримкою, що дозволяє швидко та ефективно виконувати завдання. Функція ПТМ є важливим компонентом, оскільки дозволяє асистенту відповідати чіткою, природною мовою, тим самим підвищуючи залученість користувача. Крім того, функція збереження контексту дозволяє асистенту зберігати контекст розмови під час декількох обмінів, тим самим сприяючи більш людській взаємодії. Реалізація активації за ключовим словом гарантує, що голосовий асистент реагує тільки тоді, коли його викликають, тим самим зберігаючи конфіденційність користувача. Щоб розширити його корисність, асистент повинен підтримувати інтеграцію зі сторонніми додатками, такими як календарі або платформи для обміну повідомленнями, а також включати засоби контролю конфіденційності даних, які дозволяють користувачам безпечно керувати своїми особистими даними. У сукупності ці функціональні вимоги гарантують, що голосовий асистент буде чуйним, інтуїтивно зрозумілим і адаптованим до потреб користувача.

Нижче наведено основні функціональні вимоги до голосового асистента:

- Система повинна бути здатна точно розпізнавати і перетворювати розмовну мову в текст з точністю не менше 90% за нормальних умов. Така функціональність забезпечує ефективне розуміння команд користувача.

- ГП повинен вміти інтерпретувати команди користувача природною мовою, визначати наміри користувача та витягувати відповідну інформацію з усного введення. Система повинна підтримувати команди на декількох мовах, з оновленнями, заснованими на лінгвістичних покращеннях.
- Система повинна бути здатна розпізнавати і виконувати певні команди, такі як встановлення нагадувань, надання прогнозу погоди і здійснення телефонних дзвінків. Вкрай важливо, щоб кожна команда виконувалася в межах тимчасового вікна обробки для забезпечення своєчасної реакції.
- Система має бути персоналізованою відповідно до вподобань користувача. Асистент повинен бути здатним навчатися та адаптуватися до індивідуальних уподобань користувача з часом, що дозволить йому надавати індивідуальні рекомендації та відповіді на основі попередніх взаємодій.
- Система ПТМ повинна бути здатна відтворювати природну мову у відповідь на введення користувачем. Система повинна відповідати користувачам мовою, яка сприймається як природна, перетворюючи текстові відповіді на чіткі та зрозумілі голосові повідомлення. Користувачеві повинні бути доступні різноманітні голосові опції на вибір.
- Система повинна бути здатна реагувати на певні ключові слова або команди. Вкрай важливо, щоб ГП був запрограмований на прослуховування певного слова активації (наприклад, «Jarvis») і вмикався лише тоді, коли це слово буде виявлено. Це гарантує конфіденційність користувача і запобігає ненавмисним реакціям.
- Система повинна мати можливість зберігати контекст розмови під час декількох обмінів, що забезпечує більш природний, безперервний потік взаємодії.

## 3 ТЕХНІЧНІ ВІДОМОСТІ

### 3.1 Скелет програми

Процес розробки починається з визначення інструменту де писати код, майбутньої архітектури файлів та необхідних бібліотек. Для реалізації програмного коду був обраний Visual Studio Code. Це безкоштовний редактор коду з відкритим вихідним кодом, розроблений компанією Microsoft. Редактор пропонує ряд функцій, включаючи налагодження та інтеграцію з Git, як для невеликих проєктів, так і для складних додатків [56]. Архітектура проєкту (рис. 3.1) окреслює основні компоненти, їхні взаємозв'язки і те, як вони взаємодіють, щоб відповідати вимогам проєкту. По суті, вона слугує планом, який спрямовує розробку, забезпечуючи відповідність заявленим цілям. Модуль «engine» слугує місцем серверної частини, де зберігаються функціональні елементи: конфігурації, основні команди, база даних, константи та інше. Модуль «www» відповідає за збереження клієнтської сторони: головний інтерфейс, стилі, функції та звукові сигнали.

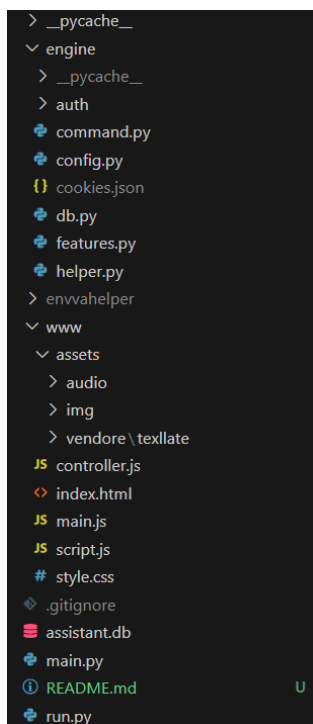


Рисунок 3.1 – Структура проєкту

Головною точкою входу до програми є файли `run.py` та `main.py`. Де визначаються запущені процеси та виконання основних команд, відповідно. Також попередньо треба створити віртуальне середовище, аби інтерпретатор Python міг скопіювати код та виконати його. Це реалізовується за допомоги команди `'python -m venv *name*'`, де `name` – назва віртуального середовища. Слід зазначити, аби відповідний код правильно працював, потрібно залучити підключення належних бібліотек (табл. 3.1). Як зазначалося в минулих розділах, бібліотеки дозволяють використовувати завчасно створені методи та класи для реалізації необхідного функціоналу.

Таблиця 3.1 – Опис бібліотек

Сторона	Бібліотека/фреймворк	Опис
1	2	3
Клієнтська	Bootstrap	Надає заздалегідь розроблені компоненти HTML, CSS та JavaScript для швидкого та ефективного створення адаптивних додатків.
	Jquery	Спрощує маніпуляції з HTML-документами, обробку подій та анімацію.
	particles.js	Надає доступ до кастомних анімацій для покращення візуальної складової.
	textillate.js	Надає доступ до кастомних текстових варіацій.
	siriwave.js	Дозволяє впровадити анімацію хвилі як на IOS.
Серверна	EEL	Забезпечує міст між клієнтською та серверною частиною.
	playsound	Відповідає за відтворення звуку.
	pytsx3	Відтворює ПТМ функціональність.
	SpeechRecognition	Бібліотека для виконання розпізнавання мови.



## Продовження таблиці 3.1

1	2	3
Серверна	PyAudio	Відтворювати та записувати аудіо на різних платформах
	pywhatkit	Найпопулярніша бібліотека для автоматизації WhatsApp та YouTube
	sqlite3	Забезпечує легку дискову базу даних, яка не потребує окремого серверного процесу
	pyportcupine	Високоточний і легкий механізм розпізнавання слів, який прокидається під час прослуховування.
	hugchat	HuggingChat Python API, розширюваний для чат-ботів тощо.
	opencv-python	Готові пакети OpenCV для Python, що працюють лише на CPU.
	opencv-contrib-python	Допоміжна бібліотека для OpenCV.

### 3.2 Озвучення тексту

Одним з головних елементів при створенні ГП для покращення міжособистісних зв'язків є озвучення отриманого тексту. Дана функціональність досягається в методі «speak» (ДОДАТОК А – `comman.py`). Використовуючи бібліотеку для реалізації ПТМ, ми можемо задати модель для відтворення голосу, стиль озвучення, частоту та інше (рис. 3.2). Отримуючи запит, а саме текст, з клієнтської сторони, метод створює модель для перетворення тексту в мову, зазначає частоту вимови і відображає отримані дані на інтерфейс.



```

1 engine = pyttsx3.init('sapi5')
2 engine.setProperty('rate', 174)

```

Рисунок 3.2 – Налаштування моделі ПТМ

### 3.3 Виконання команд

Наступним ключовим етапом є отримання можливості відтворити озвучені команди. За дану функціональність відповідатимуть наступні методи: 'take\_command' та 'allCommands' (ДОДАТОК А – command.py) – отримання вхідних даних, а також 'openCommand' 'playYoutube' (ДОДАТОК А – features.py) – виконання заданої команди. Перші файли використовують бібліотеку для розпізнання мови SpeechRecognition. Головним моментом з налаштування є створення мосту для отримання вхідного голосу «r.listen(source, 10, 10)», де другий параметр відповідає за час отримання даних, а 3й параметр – за те, скільки секунд буде збережено як інформація. Також, обрання обробника голосу, для цього було використано реалізацію від Google «query = r.recognize\_google(audio)». Далі, зазначений механізм використовується для визначення, яка команда повинна бути реалізована. В системі реалізовано три варіанти: відкрити, програти на YouTube та обговорення зі ШІ. Основний список команд налаштовується за з'єднанням з базою даних SQLite3, де зберігається інформація про голосову команду та шлях до файлу/сервісу (рис. 3.3).



```

1 import sqlite3
2 from config import DATABASE_NAME
3
4 conn = sqlite3.connect(DATABASE_NAME)
5 cursor = conn.cursor()
6
7 # query = "CREATE TABLE IF NOT EXISTS sys_command(id integer primary key, name VARCHAR(100), path VARCHAR(1000))"
8 # cursor.execute(query)

```

Рисунок 3.3 – З'єднання з базою даних

В залежності від того, яке ключове слово для команди прийшло, система обирає відповідний метод. Якщо перший варіант – відкриває відповідну програму з бази даних, якщо другий – запускає відео на YouTube, останній – звертається до ШІ по допомогу.

### 3.4 Ключове слово

Більшість сучасних голосових помічників працюють за допомогою системи активації за ключовими словами. У цьому випадку наша мета – навчити систему реагувати на задані ключові слова. Бібліотека Porcupine дозволяє створювати розпізнавання ключових слів за допомогою команди типу `‘create(keywords=['jarvis', 'alexa'])’`. Слід зазначити, що вищезгадані ключові слова можна налаштовувати, однак створення власних ключових слів потребує підписки та додаткового навчання моделі.

При виявленні ключового слова, система налаштована так, щоб реагувати на нього натисканням комбінації клавіш «Win + J». Ця дія активує попередньо налаштовану подію, яка відкриває модальне вікно асистента та ініціює основний алгоритм (див. ДОДАТОК А – `main.js`). На цьому етапі асистент починає активне прослуховування для подальших голосових команд.

### 3.5 Історія розмови

Створення історії чату в голосових помічниках має важливе значення для покращення взаємодії з користувачем, персоналізації відповідей і підвищення обізнаності про контекст. Для легкої взаємодії, історія чату була виконана за допомоги використання CSS та JS. Методи `‘senderText’` та `‘receiverText’` відповідають за відображення та ховання вікна сесії (ДОДАТОК А – `controller.js`). Історія зберігається в поточному сеансі, щоб зафіксувати розмови як безперервні потоки. Таким чином, можна повернутися до попередніх запитань або інструкцій в межах одного сеансу, що підвищує контекстну точність. Питання і відповіді в чаті підсвічені окремими стилями. Розширення функціоналу передбачає залучення бази даних для зберігання інформації про сеанс, користувача та задані питання.

### 3.6 Штучний інтелект

Інтеграція моделі ШІ з голосовим асистентом полегшує надання інтелектуальних, контекстно-залежних відповідей завдяки включенню передових можливостей машинного навчання в операційну структуру додатку. Цей зв'язок зазвичай передбачає використання інтерфейсів прикладного програмування для полегшення обміну даними між моделлю ШІ та голосовим асистентом. Голосовий помічник збирає вхідні дані користувача за допомогою розпізнавання мови, перетворює їх на текст і надсилає моделі ШІ для обробки. Безкоштовна модель від HuggingFace інтерпретує вхідні дані, формулює відповідну відповідь і передає її назад голосовому помічнику, який потім перетворює її на розмовну мову (ДОДАТОК А – features.py: chatbot). Для цього потрібно увійти в обліковий профіль, або створити його. Скопіювавши файли Cookie, передати їх в відповідний файл cookies.json. Після цього використовуючи модель штучного інтелекту, віртуальний помічник почне надавати більш детальні відповіді, обробляти складні запити і навіть вчитися на попередніх взаємодіях, тим самим покращуючи користувацький досвід, персоналізуючи його і підвищуючи точність.

### 3.7 Безпекові заходи

Ключовим етапом є залучення безпеки до майбутнього додатку. Серед варіацій, був обраний відомий процес ідентифікації обличчя користувача. Використовуючи можливості розпізнавання облич OpenCV, голосовий помічник може ідентифікувати конкретних користувачів і активуватися лише тоді, коли виявлено авторизовані обличчя. Цей процес включає в себе захоплення кадрів відео в реальному часі (рис 3.1 – auth/sample.py), виявлення облич (рис 3.1 – auth/recognize.py) і порівняння їх з попередньо навченою моделлю розпізнаних облич (рис 3.1 – auth/trainer.py). Під час запуску програми, при виявленні збігу голосовий помічник автоматично запускається, таким чином гарантуючи, що система реагує тільки на перевірених користувачів (ДОДАТОК А – main.py).

Такий підхід не лише додає рівень безпеки, але й уможливорює персоналізовану взаємодію, оскільки голосовий помічник може коригувати свої реакції на основі користувацьких уподобань, що зберігаються в його пам'яті. Таким чином, використання розпізнавання обличчя як тригера активації може підвищити інтуїтивність, безпеку та персоналізацію голосових помічників.

### 3.8 Алгоритм роботи

У програмі користувач може зробити запит до голосового помічника як голосом, так і текстом, а також переглянути історію поточного чату. Запит до помічника можна ініціювати натисканням кнопки в головному меню або вимовивши ключове слово. Після активації відповідний запит передається до бази даних, де програма шукає потрібну команду. Якщо команда не знайдена в базі, на запит відповідає штучний інтелект, формуючи відповідь у реальному часі. Результатом виконання алгоритму є відповідь від ГП, яка відображається користувачеві, забезпечуючи швидкий та зручний доступ до інформації (рис. 3.4).

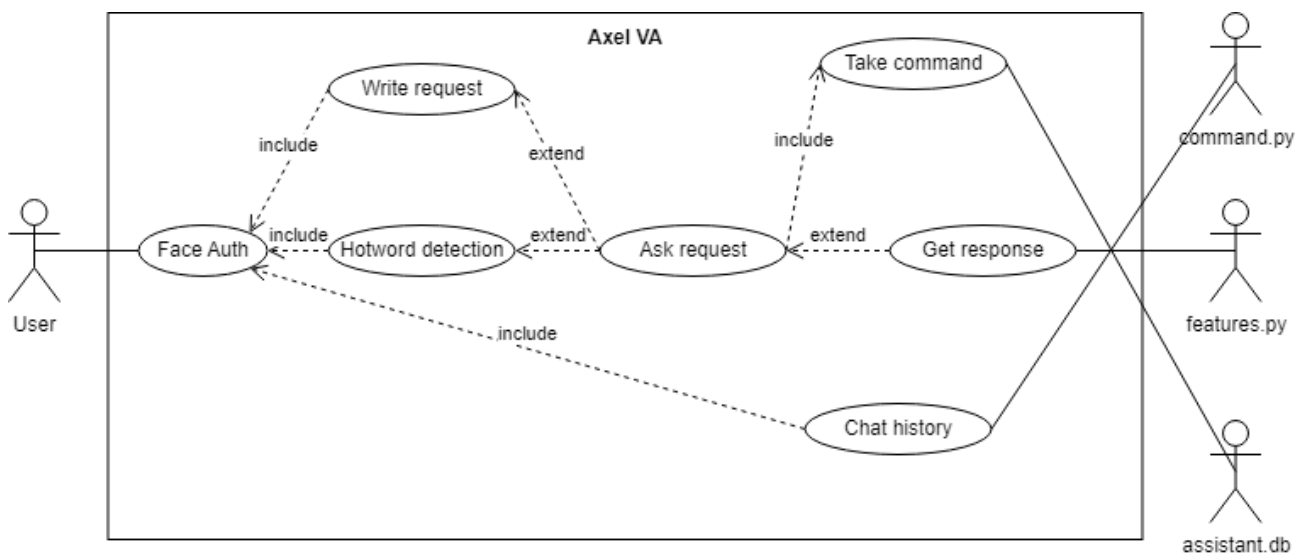


Рисунок 3.4 – Потік виконання програми

Докладнішу інформацію про реалізацію програми можна знайти в моєму репозиторії на [GitHub](#).

## ВИСНОВКИ

У ході роботи було розроблено інформаційну систему "Axel" для вдосконалення навичок міжособистісної комунікації.

Під час виконання кваліфікаційної роботи були здійснені такі завдання:

- створено дизайн, орієнтований на користувацький досвід, з інтуїтивно зрозумілим і привабливим інтерфейсом, що сприяє легкій взаємодії;
- сформульовано функціональні вимоги, що визначають ключові операції системи, такі як обробка команд, генерація відповідей та набуття знань через взаємодію;
- встановлено нефункціональні вимоги, які забезпечують необхідні якості системи: швидкодію, надійність та зручність використання;
- створено інтерфейс з інтерактивними компонентами, що поліпшують користувацький досвід та сприяють ефективній комунікації;
- обрана мова програмування Python завдяки якій реалізовано ГП;
- підібрано відповідні бібліотеки;
- обрано безкоштовну модель ШІ, яка дозволяє системі точно розпізнавати та інтерпретувати мову користувача;
- забезпечено вирішення питань безпеки даних.

Створення система покращення міжособистісних комунікацій дозволяє робити, писати та переглядати поточні запити. Додаток розроблений з урахуванням актуальних, на сьогоднішній день, особливостей дизайну користувацького інтерфейсу. Впроваджуючи зручний користувацький досвід та легку інтеграцію з повсякденними завданнями. Від автоматизації роботи за комп'ютером, до визначення необхідної інформації на базі штучного інтелекту.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Guo H. et al. Effectiveness of information and communication technology (ICT) for addictive behaviors: An umbrella review of systematic reviews and meta-analysis of randomized controlled trials // *Comput Human Behav.* Pergamon, 2023. Vol. 147. P. 107843.
2. Heaton J. Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning // *Genet Program Evolvable Mach.* Springer Nature, 2018. Vol. 19, № 1–2. P. 305–307.
3. Russell S.J.. et al. Artificial intelligence : a modern approach. Pearson, 2021. 1115 p.
4. Wired for speech: How voice activates and advances the human-computer relationship. [Electronic resource]. URL: <https://psycnet.apa.org/record/2005-11771-000> (accessed: 11.11.2024).
5. Jurafsky D., Martin J.H. Speech and Language Processing An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models Third Edition draft Summary of Contents.
6. Searle J.R. Speech Acts: An Essay in the Philosophy of Language. Cambridge University Press, 1969.
7. Li J. et al. Security and privacy problems in voice assistant applications: A survey // *Comput Secur.* Elsevier Ltd, 2023. Vol. 134. P. 103448.
8. Zhang G. et al. DolphinAttack: Inaudible voice commands // *Proceedings of the ACM Conference on Computer and Communications Security.* Association for Computing Machinery, 2017. P. 103–117.
9. Social learning theory. [Electronic resource]. URL: <https://psycnet.apa.org/record/1979-05015-000> (accessed: 11.11.2024).
10. Parker E.B. et al. The social psychology of telecommunications // *Contemp Sociol.* SAGE Publications, 1976. Vol. 7, № 1. P. 32.
11. Dragojevic M., Gasiorek J., Giles H. Communication Accommodation Theory // *The International Encyclopedia of Interpersonal Communication.* Wiley, 2015. P. 1–21.
12. Sweller J. Cognitive load during problem solving: Effects on learning // *Cogn Sci.* No longer published by Elsevier, 1988. Vol. 12, № 2. P. 257–285.
13. Perloff R. Daniel Goleman's Emotional intelligence: Why it can matter more than IQ. // *The Psychologist-Manager Journal.* 1997. Vol. 1, № 1. P. 21–22.
14. The brain that changes itself: Stories of personal triumph from the frontiers of brain science. [Electronic resource]. URL: <https://psycnet.apa.org/record/2006-23192-000> (accessed: 11.11.2024).
15. The cognitive neurosciences, 5th ed. [Electronic resource]. URL: <https://psycnet.apa.org/record/2015-20745-000> (accessed: 11.11.2024).
16. The developing mind: How relationships and the brain interact to shape who we are, 2nd ed. [Electronic resource]. URL: <https://psycnet.apa.org/record/2012-12726-000> (accessed: 11.11.2024).

17. LeDoux J. Cognitive and emotional interactions in the brain: The mysterious underpinnings of emotional life. Simon & Schuster, 1996. P. 384.
18. Martin A. Mirroring People: The New Science of How We Connect With Others. // J Am Acad Child Adolesc Psychiatry. Elsevier BV, 2008. Vol. 47, № 12. P. 1462–1463.
19. Steinsbekk S. et al. The new social landscape: Relationships among social media use, social skills, and offline friendships from age 10–18 years // Comput Human Behav. Pergamon, 2024. Vol. 156. P. 108235.
20. Cummings J.J., Shore Ingber A. Distinguishing social virtual reality: Comparing communication channels across perceived social affordances, privacy, and trust // Comput Human Behav. Pergamon, 2024. Vol. 161. P. 108427.
21. Д.О. Папіжук, О.А. Шовкопляс. Система розпізнання голосових команд для адаптації навчального процесу до динамічних змін у освіті // Інформатика, математика, автоматика (ІМА - 2024) : матеріали та програма міжнародної наукової конференції молодих учених, м. Суми-Астана, 22–26 квітня 2024 року. Суми: Сумський державний університет, 2024. P. 85.
22. Menon D., Shilpa K. “Hey, Alexa” “Hey, Siri”, “OK Google” ....” exploring teenagers’ interaction with artificial intelligence (AI)-enabled voice assistants during the COVID-19 pandemic // Int J Child Comput Interact. Elsevier BV, 2023. Vol. 38. P. 100622.
23. Molinillo S. et al. Impact of perceived value on intention to use voice assistants: The moderating effects of personal innovativeness and experience // Psychol Mark. John Wiley and Sons Inc, 2023. Vol. 40, № 11. P. 2272–2290.
24. Derakhshan A., Teo T., Khazaie S. Investigating the usefulness of artificial intelligence-driven robots in developing empathy for English for medical purposes communication: The role-play of Asian and African students // Comput Human Behav. Pergamon, 2025. Vol. 162. P. 108416.
25. She R. et al. The double-edged sword effect of social networking use intensity on problematic social networking use among college students: The role of social skills and social anxiety // Comput Human Behav. Pergamon, 2023. Vol. 140. P. 107555.
26. Schmitz A., Quiñones M., Díaz-Martín A.M. Alexa, Give Me a Hand...On GenZ’ Smart Speaker Acceptance in a Retail Environment. 2024. P. 49–58.
27. Ossadnik J., Muehlfeld K., Goerke L. Man or machine – or something in between? Social responses to voice assistants at work and their effects on job satisfaction // Comput Human Behav. Elsevier Ltd, 2023. Vol. 149. P. 107919.
28. Chen C. et al. Screen or No Screen? Lessons Learnt from a Real-World Deployment Study of Using Voice Assistants With and Without Touchscreen for Older Adults // The 25th International ACM SIGACCESS Conference on Computers and Accessibility. New York, NY, USA: ACM, 2023. P. 1–21.



29. Brewer R. et al. An Empirical Study of Older Adult's Voice Assistant Use for Health Information Seeking // *ACM Trans Interact Intell Syst.* 2022. Vol. 12, № 2. P. 1–32.
30. Wu Q. et al. Why communication matters? The roles of patient-provider communication and social media use in cancer survivors' meaning in life // *Comput Human Behav.* Pergamon, 2024. Vol. 156. P. 108218.
31. Janson A. How to leverage anthropomorphism for chatbot service interfaces: The interplay of communication style and personification // *Comput Human Behav.* Pergamon, 2023. Vol. 149. P. 107954.
32. Malodia S. et al. Why Do People Use Artificial Intelligence (AI)-Enabled Voice Assistants? // *IEEE Trans Eng Manag.* 2024. Vol. 71. P. 491–505.
33. Zhou P., Xie Y., Liang C. How to increase consumers' continued use intention of artificial intelligence voice assistants? The role of anthropomorphic features // *Electronic Markets.* 2023. Vol. 33, № 1. P. 60.
34. Hernández Ortega B.I., Lucia-Palacios L. Trust in word of voice communication: why consumers adhere to purchase recommendations made by smart voice assistants // *Marketing Intelligence & Planning.* Emerald Publishing, 2023. Vol. 41, № 8. P. 1093–1120.
35. Jain S. et al. Impact of irritation and negative emotions on the performance of voice assistants: Netting dissatisfied customers' perspectives // *Int J Inf Manage.* 2023. Vol. 72. P. 102662.
36. Seymour W. et al. A Systematic Review of Ethical Concerns with Voice Assistants // *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society.* New York, NY, USA: ACM, 2023. P. 131–145.
37. Zeiler M.D., Fergus R. Visualizing and Understanding Convolutional Networks // *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics).* Springer Verlag, 2013. Vol. 8689 LNCS, № PART 1. P. 818–833.
38. *Contexts of Accommodation* // *Contexts of Accommodation.* Cambridge University Press, 1991.
39. Goldstein E. Bruce. *Cognitive psychology : connecting mind, research and everyday experience.* Cengage learning, 2015. P. 464.
40. Pulvermüller F., Fadiga L. Active perception: sensorimotor circuits as a cortical basis for language // *Nature Reviews Neuroscience* 2010 11:5. Nature Publishing Group, 2010. Vol. 11, № 5. P. 351–360.
41. Gollwitzer P.M. Implementation intentions: Strong effects of simple plans // *American Psychologist.* American Psychological Association Inc., 1999. Vol. 54, № 7. P. 493–503.
42. Lecun Y., Bengio Y., Hinton G. Deep learning // *Nature* 2015 521:7553. Nature Publishing Group, 2015. Vol. 521, № 7553. P. 436–444.
43. Preckel K., Kanske P., Singer T. On the interaction of social affect and cognition: empathy, compassion and theory of mind // *Curr Opin Behav Sci.* Elsevier, 2018. Vol. 19. P. 1–6.

44. Д.О. Папіжук, О.А. Шовкопляс. Голосові помічники зі штучним інтелектом як персональні фітнес-тренери // «Інформаційні технології і автоматизація – 2024»: матеріали XVII міжнародної науково-практичної конференції, м. Одеса, 31 жовтня - 1 листопада 2024 р. Одеса: Одеський національний технологічний університет, 2024. Р. 672–674.
45. Building a Decoupled Voice Assistant [Electronic resource]. URL: <https://www.crio.do/blog/build-a-voice-assistant/> (accessed: 10.11.2024).
46. Vaswani A. et al. Attention Is All You Need // Adv Neural Inf Process Syst. Neural information processing systems foundation, 2017. Vol. 2017-December. P. 5999–6009.
47. Devlin J. et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding // NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference. Association for Computational Linguistics (ACL), 2018. Vol. 1. P. 4171–4186.
48. Speech-to-Text documentation | Cloud Speech-to-Text Documentation | Google Cloud [Electronic resource]. URL: <https://cloud.google.com/speech-to-text/docs/> (accessed: 11.11.2024).
49. Latha B. et al. Hand Gesture and Voice Assistants // E3S Web of Conferences. EDP Sciences, 2023. Vol. 399. P. 04050.
50. Eel in Python - Javatpoint [Electronic resource]. URL: <https://www.javatpoint.com/eel-in-python> (accessed: 10.11.2024).
51. What is Python? How the Interpreter Works and How to Write “Hello World” in Python [Electronic resource]. URL: <https://www.freecodecamp.org/news/what-is-python-beginners-guide/> (accessed: 10.11.2024).
52. What is OpenCV Library? - GeeksforGeeks [Electronic resource]. URL: <https://www.geeksforgeeks.org/opencv-overview/> (accessed: 10.11.2024).
53. What is Hugging Face and Transformers [Electronic resource]. URL: <https://www.geeksforgeeks.org/hugging-face-transformers/> (accessed: 10.11.2024).
54. Introduction to SQLite - GeeksforGeeks [Electronic resource]. URL: <https://www.geeksforgeeks.org/introduction-to-sqlite/> (accessed: 10.11.2024).
55. Functional vs. Non Functional Requirements - GeeksforGeeks [Electronic resource]. URL: <https://www.geeksforgeeks.org/functional-vs-non-functional-requirements/> (accessed: 11.11.2024).
56. Why Visual Studio Code? [Electronic resource]. URL: <https://code.visualstudio.com/Docs/editor/whyvscode> (accessed: 10.11.2024).

## ДОДАТОК А ЛІСТИНГ ПРОГРАМНОГО КОДУ

### A1 command.py

```
import pyttsx3
import speech_recognition as sr
import eel
import time

def speak(text):
    text = str(text)
    engine = pyttsx3.init('sapi5')
    engine.setProperty('rate', 174)
    eel.DisplayMessage(text)
    engine.say(text)
    eel.receiverText(text)
    engine.runAndWait()

def takeCommand():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        eel.DisplayMessage('Listening...')
        print("L")
        r.pause_threshold = 1
        r.adjust_for_ambient_noise(source)

        audio = r.listen(source, 10, 10)

    try:
        print("R")
        eel.DisplayMessage('Recognizing...')
        query = r.recognize_google(audio)
        print(f"User said: {query}")
        eel.DisplayMessage(query)
        time.sleep(2)
    except Exception as e:
        return ""

    return query.lower()

@eel.expose
def allCommands(message=1):

    if (message == 1):
        query = takeCommand()
        print(query)
        eel.senderText(query)
    else:
        query = message
        eel.senderText(query)
```

```

try:
    if 'open' in query:
        from engine.features import openCommand
        openCommand(query)
    elif "on youtube" in query:
        from engine.features import playYoutube
        playYoutube(query)
    else:
        from engine.features import chatBot
        chatBot(query)

except Exception as e:
    print('Error')
    print(e)

```

```
eel.ShowHood()
```

### **A2 features.py**

```

import os
import sqlite3
import struct
import time
import webbrowser
from hugchat import hugchat
from playsound import playsound
import pvporcupine
import pyaudio
from engine.command import speak
from engine.config import *
import eel
import pywhatkit as kit

from engine.helper import extractYoutubeTerm

conn = sqlite3.connect(DATABASE_NAME)
cursor = conn.cursor()

@eel.expose
def playAssistantSound():
    playsound(MUSIC_ROOT)

def executeQuery(query, app):
    cursor.execute(query, (app,))

    return cursor.fetchall()

def openCommand(query):
    query = query.replace(ASSISTANT_NAME, EMPTY)
    query = query.replace(OPEN, EMPTY)
    query.lower()

```

```

app_name = query.strip()

if app_name != EMPTY:
    try:
        results = executeQuery(SYS_COMMAND_QUERY, app_name)

        if len(results) != 0:
            speak(OPENING + query)
            os.startfile(results[0][0])

        elif len(results) == 0:
            results = executeQuery(WEB_COMMAND_QUERY,
app_name)

            if len(results) != 0:
                speak(OPENING + query)
                webbrowser.open(results[0][0])

            else:
                speak(OPENING + query)
                try:
                    os.system(START + SPACE + query)
                except:
                    speak(NOT_FOUND)
    except:
        speak(SMTH_WENT_WRONG)

def playYoutube(query):
    search_term = extractYoutubeTerm(query)
    speak(PLAYING + search_term + ON_YOUTUBE)
    kit.playonyt(search_term)

def hotword():
    print('hotword')
    porcupine=None
    paud=None
    audio_stream=None
    try:

        # pre trained keywords
        porcupine=pvporcupine.create(keywords=['jarvis','alexa'])
        paud=pyaudio.PyAudio()
        audio_stream=paud.open(
            rate=porcupine.sample_rate,
            channels=1,
            format=pyaudio.paInt16,
            input=True,
            frames_per_buffer=porcupine.frame_length
        )

```

```

# loop for streaming
while True:
    keyword=audio_stream.read(porcupine.frame_length)
    keyword=struct.unpack_from("h"*porcupine.frame_length,
keyword)

    # processing keyword comes from mic
    keyword_index=porcupine.process(keyword)

    # checking first keyword detected for not
    if keyword_index>=0:
        print("hotword detected")

        # pressing shortcut key win+j
        import pyautogui as autogui
        autogui.keyDown('win')
        autogui.press('j')
        time.sleep(2)
        autogui.keyUp('win')

except:
    if porcupine is not None:
        porcupine.delete()
    if audio_stream is not None:
        audio_stream.close()
    if paud is not None:
        paud.terminate()

def chatBot(query):
    user_input = query.lower()
    chatbot = hugchat.ChatBot(cookie_path='engine\\cookies.json')
    id = chatbot.new_conversation()
    chatbot.change_conversation(id)
    response = chatbot.chat(user_input)
    print(response)
    speak(response)
    return response

```

### **A3 helper.py**

```

import re

from engine.config import PATTERN

def extractYoutubeTerm(query):
    match = re.search(PATTERN, query, re.IGNORECASE)

    return match.group(1) if match else None

```

### **A4 main.py**

```

import os

```

```

import eel

from engine.features import *
from engine.command import *
from engine.auth import recognize

def start():
    eel.init("www")

    playAssistantSound()
    @eel.expose
    def init():
        eel.hideLoader()
        speak('Ready for Face Authentication')
        flag = recognize.AuthenticateFace()

        if (flag == 1):
            eel.hideFaceAuth()

            speak('Authentication is sucessful')
            eel.hideFaceAuthSuccess()

            speak('You are welcome, how can I help you?')
            eel.hideStart()

            playAssistantSound()
        else:
            speak('Authentication is failed')

        os.system('start msedge.exe -
app="http://localhost:8000/index.html"')

        eel.start('index.html', mode=None, host='localhost',
block=True)

```

### **A5 run.py**

```

import multiprocessing

def startAxel():
    print("Process 1 is running")
    from main import start
    start()

def listenHotword():
    print("Process 2 is running")
    from engine.features import hotword
    hotword()

if __name__ == '__main__':
    p1 = multiprocessing.Process(target=startAxel)
    p2 = multiprocessing.Process(target=listenHotword)

```

```

p1.start()
p2.start()
p1.join()

if p2.is_alive():
    p2.terminate()
    p2.join()

print("system stop")

```

### **A6 controller.js**

```

$(document).ready(function () {
    eel.expose(DisplayMessage)
    function DisplayMessage(message) {
        $(".siri-message li:first").text(message);
        $(".siri-messsage").textillate('start')
    }

    eel.expose(ShowHood)
    function ShowHood() {
        $("#Oval").attr("hidden", false);
        $("#SiriWave").attr("hidden", true);
    }

    eel.expose(senderText)
    function senderText(message) {
        var chatBox = document.getElementById("chat-canvas-body");
        if (message.trim() !== "") {
            chatBox.innerHTML += `<div class="row justify-content-
end mb-4">
            <div class = "width-size">
            <div class="sender_message">${message}</div>
            </div>`;

            chatBox.scrollTop = chatBox.scrollHeight;
        }
    }

    eel.expose(receiverText)
    function receiverText(message) {

        var chatBox = document.getElementById("chat-canvas-body");
        if (message.trim() !== "") {
            chatBox.innerHTML += `<div class="row justify-content-
start mb-4">
            <div class = "width-size">
            <div class="receiver_message">${message}</div>
            </div>
            </div>`;

            chatBox.scrollTop = chatBox.scrollHeight;
        }
    }
}

```



```

    }
}

eel.expose(hideLoader)
function hideLoader() {

    $("#Loader").attr("hidden", true);
    $("#FaceAuth").attr("hidden", false);

}

eel.expose(hideFaceAuth)
function hideFaceAuth() {

    $("#FaceAuth").attr("hidden", true);
    $("#FaceAuthSuccess").attr("hidden", false);

}

eel.expose(hideFaceAuthSuccess)
function hideFaceAuthSuccess() {

    $("#FaceAuthSuccess").attr("hidden", true);
    $("#HelloGreet").attr("hidden", false);

}

eel.expose(hideStart)
function hideStart() {

    $("#Start").attr("hidden", true);

    setTimeout(function () {
        $("#Oval").addClass("animate__animated
animate__zoomIn");

    }, 1000)
    setTimeout(function () {
        $("#Oval").attr("hidden", false);
    }, 1000)
}
});

```

### **A7 main.js**

```

$(document).ready(function () {

    eel.init() ()

    $('.text').textillate({
        loop: true,
        sync: true,
        in: {

```

```

        effect: "bounceIn",
    },
    out: {
        effect: "bounceOut",
    },
});

var siriWave = new SiriWave({
    container: document.getElementById("siri-container"),
    width: 800,
    height: 200,
    style: "ios9",
    amplitude: "1",
    speed: "0.30",
    autostart: true
});

$('.siri-message').textillate({
    loop: true,
    sync: true,
    in: {
        effect: "fadeInUp",
        sync: true
    },
    out: {
        effect: "fadeOutUp",
        sync: true
    },
});

$("#MicBtn").click(function (e) {
    eel.playAssistantSound()
    $("#Oval").attr("hidden", true);
    $("#SiriWave").attr("hidden", false);
    eel.allCommands();
});

function docKeyUp(e) {
    if (e.key === 'j' && e.metaKey){
        eel.playAssistantSound()
        $("#Oval").attr("hidden", true);
        $("#SiriWave").attr("hidden", false);
        eel.allCommands() ()
    }
}
document.addEventListener('keyup', docKeyUp, false);

function playAssistant(message) {
    if (message !== ''){

```

```

        $("#Oval").attr("hidden", true);
        $("#SiriWave").attr("hidden", false);
        eel.allCommands(message)
        $("#chatbox").val('');
        $("#MicBtn").attr("hidden", false);
        $("#SendBtn").attr("hidden", true);
    }
}

function showHideButton(message) {
    if (message.length == 0) {
        $("#MicBtn").attr("hidden", false);
        $("#SendBtn").attr("hidden", true);
    }
    else {
        $("#MicBtn").attr("hidden", true);
        $("#SendBtn").attr("hidden", false);
    }
}

$('#chatbox').keyup(function () {
    let message = $('#chatbox').val();
    showHideButton(message)
});

$('#SendBtn').click(function () {
    let message = $('#chatbox').val()
    playAssistant(message)
});

$('#chatbox').keypress(function (e) {
    key = e.which;
    if (key == 13) {
        let message = $('#chatbox').val();
        playAssistant(message)
    }
});
});

```

## ДОДАТОК Б СТИЛІ ТА АНІМАЦІЯ

```
body {
  background-color: black;
  overflow-y: hidden;
  overflow-x: hidden;
}

.square {
  position: relative;
  width: 400px;
  height: 400px;
  display: flex;
  justify-content: center;
  align-items: center;
}

.square span:nth-child(1) {
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background-image: radial-gradient(#6b72ff00 50%, #000dff3b 40%);
  box-shadow: 0 0 50px rgb(25, 0, 255), inset 0 0 50px rgb(25, 0,
255);
  border-radius: 38% 62% 63% 37% / 41% 44% 56% 59%;
  transition: 0.5s;
  animation: animate1 6s linear infinite;
}

.square span:nth-child(2) {
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background-image: radial-gradient(#6b72ff00 50%, #000dff3b 40%);
  box-shadow: 0 0 50px rgb(25, 0, 255), inset 0 0 50px rgb(25, 0,
255);
  border-radius: 38% 62% 63% 37% / 41% 44% 56% 59%;
  transition: 0.5s;
  animation: animate2 4s linear infinite;
}

.square span:nth-child(3) {
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
```

```

height: 100%;
background-image: radial-gradient(#6b72ff00 50%, #000dff3b 40%);
box-shadow: 0 0 50px rgb(25, 0, 255), inset 0 0 50px rgb(25, 0,
255);
border-radius: 38% 62% 63% 37% / 41% 44% 56% 59%;
transition: 0.5s;
animation: animate3 8s linear infinite;
}

@keyframes animate1 {
  0% {
    transform: rotate(0deg);
  }

  100% {
    transform: rotate(360deg);
  }
}

@keyframes animate2 {
  0% {
    transform: rotate(0deg);
  }

  100% {
    transform: rotate(360deg);
  }
}

@keyframes animate3 {
  0% {
    transform: rotate(0deg);
  }

  100% {
    transform: rotate(360deg);
  }
}

#TextInput {
  background-color: #181818a8;
  border-color: blue;
  box-shadow: 0 0 20px rgb(25, 0, 255),
  inset 0 0 0px rgb(25, 0, 255);
  border-radius: 8px;
  color: white;
  padding: 3px 0px 3px 20px;
  margin: 0px 20%;
}

.input-field {

```

```
background-color: transparent;
border: none;
width: 95%;
outline: none;
color: white;
font-family: cursive;
}

.glow-on-hover {
width: 35px;
height: 35px;
border: none;
outline: none;
color: #fff;
background: #111;
cursor: pointer;
position: relative;
z-index: 0;
border-radius: 10px;
padding: 0px;
margin-left: 10px;
}

.glow-on-hover:before {
content: '';
background: linear-gradient(45deg, #ff0000, #ff7300, #ffff00,
#48ff00, #00ffd5, #002bff, #7a00ff, #ff00c8, #ff0000);
position: absolute;
top: -2px;
left: -2px;
background-size: 400%;
z-index: -1;
filter: blur(5px);
width: calc(100% + 4px);
height: calc(100% + 4px);
animation: glowing 20s linear infinite;
opacity: 0;
transition: opacity .3s ease-in-out;
border-radius: 10px;
}

.glow-on-hover:active {
color: #181818a8
}

.glow-on-hover:active:after {
background: transparent;
}

.glow-on-hover:hover:before {
opacity: 1;
}
```

```
}

.glow-on-hover:after {
  z-index: -1;
  content: '';
  position: absolute;
  width: 100%;
  height: 100%;
  background: #111;
  left: 0;
  top: 0;
  border-radius: 10px;
}

@keyframes glowing {
  0% {
    background-position: 0 0;
  }

  50% {
    background-position: 400% 0;
  }

  100% {
    background-position: 0 0;
  }
}

.chat-canvas {
  background-color: #191919
}

.receiver_message {
  padding: 8px;
  border: 2px solid cyan;
  border-radius: 0px 15px 15px 20px;
  width: auto;
  color: white;
  background-color: #0dcaf014;
}

.sender_message {
  padding: 8px;
  border: 1px solid #0045ff;
  border-radius: 15px 15px 0px 20px;
  width: auto;
  color: white;
  background-color: #0045ff;
}

.width-size {
```

```

    max-width: 80%;
    width: auto;
}

.svg-frame {
    position: relative;
    width: 455px;
    height: 455px;
    transform-style: preserve-3d;
    display: flex;
    justify-content: center;
    align-items: center;
    animation: change-view 2s ease-in infinite;
}

@keyframes change-view {

    0%,
    50% {
        transform: rotate(-0deg) skew(00deg) translateX(calc(0 * var(-
-i))) translateY(calc(-0px * var(--i)));
    }

    70%,
    100% {
        transform: rotate(-80deg) skew(30deg) translateX(calc(45px *
var(--i))) translateY(calc(-35px * var(--i)));
    }
}

svg {
    position: absolute;
    transition: 0.5s;
    transform-origin: center;
    width: 450px;
    height: 450px;
    fill: none;
    animation: change-view 5s ease-in-out infinite alternate;
    filter: drop-shadow(0 0 12px #00aaff);
}

#big-centro,
#outter1,
#solo-lines,
#center,
#outter-center,
#bottom-dots,
#center-lines,
#squares,
#top-dots {
    transform-origin: center;
}

```



```
    animation: rotate 4s ease-in-out infinite alternate;
}

#big-centro {
  animation-delay: -1.5s;
}

#outter1 {
  animation-delay: -1.2s;
}

#center {
  animation-delay: -2.2s;
}

#bottom-dots,
#top-dots {
  animation-duration: 7s;
}

#center-lines,
#outter-center {
  animation-duration: 6s;
  animation-delay: -3s;
}

@keyframes rotate {
  to {
    transform: rotate(360deg);
  }
}
```