

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Інформаційна система для вивчення іноземних мов
Назва теми


КВРІСТ 101107.01.07 ПЗ
Шифр


Галузь знань 12 «Інформаційні технології»
Шифр, назва

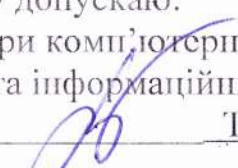
Спеціальність 126 «Інформаційні системи та технології»
Шифр, назва

Освітня програма «Інформаційні системи та технології»
Назва

Виконав: студент IV курсу, група ІСТ-20-1  Р. В. Недашківський
Підпис Ініціали, прізвище

Керівник  П. Г. Періда
Підпис, дата Ініціали, прізвище

Нормоконтролер  С. М. Лисенко
Підпис, дата Ініціали, прізвище

До захисту допускаю:
Зав. кафедри комп'ютерної
інженерії та інформаційних
систем  Т. О. Говоруценко
Підпис Ініціали, прізвище

«19» червня 2024 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 126 ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

Освітня програма «ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О.Говорущенко

“ 10 ” 09 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Недашківському Роману Вадимовичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Інформаційна система для вивчення іноземних мов

Керівник проекту (роботи) Регіда П. Г.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.02.2024 р. № 5

2. Строк подання студентом проекту (роботи) на кафедрі 06.06.2024 р.

3. Вихідні дані до проекту (роботи) Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Інформаційна система для вивчення іноземних мов та огляд існуючих інформаційних систем

Вибір мови програмування та вибір компонентів для реалізації чат-бота

Реалізація та демонстрація інформаційної системи для вивчення іноземних мов

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)





Блок-схема алгоритму роботи чат-боту

Діаграма взаємодії користувача та розробника з чат-ботом

Діаграма функціонування команд з чат-ботом

Повний код чат-бота на Python

1. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Лисенко С.М., д.т.н., проф.		
Антиплагіат	Пічепорук А.О., доцент кафедри КПС		

2. Дата видачі завдання «10» 01 2024 р.

КАЛЕНДАРНИЙ ПЛАН

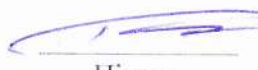
№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2024	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2024	виконано
3	Робота над розділом 1 – Інформаційна система для вивчення іноземних мов та огляд існуючих інформаційних систем	01.03.2024	виконано
4	Робота над розділом 2 – Вибір мови програмування та вибір компонентів для реалізації чат-бота	01.04.2024	виконано
5	Робота над розділом 3 – Реалізація та демонстрація інформаційної системи для вивчення іноземних мов	29.04.2024	виконано
6	Оформлення пояснювальної записки згідно вимог	27.05.2024	виконано
7	Попередній захист ВКР	30.05.2024	виконано
8	Захист ВКР на засіданні ЕК	Червень 2024 року	

Студент

Керівник роботи


Підпис

Р. В. Недапківський
Ініціали, прізвище


Підпис

П. Г. Пегіда
Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Інформаційна система для вивчення іноземних мов».

Автор роботи: Недашківський Роман Вадимович. Керівник роботи: Регіда Павло Геннадійович.

Пояснювальна записка: 58 с., 26 рис., 1 табл., 4 дод., 61 джерел.,

Графічна частина: 3 креслення.

НАВЧАННЯ, ІНФОРМАЦІЙНА СИСТЕМА, ЧАТ-БОТ, РЕАЛІЗАЦІЯ,
БАЗА ДАНИХ

Мета кваліфікаційної роботи - розробити та впровадити інформаційну систему у вигляді бота для поліпшення процесу вивчення іноземної мови. Розроблений бот має забезпечити легкий доступ до вивчення нових слів за допомогою різних інструментів які допоможуть людям ефективно та цікаво вивчити мову в будь-який зручний для них час та місце.

Об'єктом дослідження є функціонал чат-бота у застосунку telegram для вивчення іноземних мов

Предметом дослідження є оцінка можливостей функціоналу бота для вивчення іноземних мов

Під час проведення даного дослідження використано метод систематичного огляду літератури для вивчення і аналізу предметної області даного дослідження з текстових джерел інформації.






Підпис студента

06.06.2024

Дата

ЗМІСТ

ВСТУП.....	4
1 ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ ВИВЧЕННЯ ІНОЗЕМНИХ МОВ ТА ОГЛЯД ІСНУЮЧИХ ІНФОРМАЦІЙНИХ СИСТЕМ.....	6
1.1 Аналіз предметної області і виявлення наявних проблем і завдань	6
1.2 Порівняльний аналіз переваг та недоліків існуючих рішень.....	9
1.2.1 Принципи роботи Duolingo.....	11
1.2.2 Функціонал Rosetta Stone.....	13
1.2.3 Онлайн-платформа Babbel.....	15
1.2.4 Аналіз функціоналу Coursera.....	17
1.3 Постановка задачі для подальшої роботи	20
1.4 Висновок.....	21
2 ВИБІР МОВИ ПРОГРАМУВАННЯ ТА ВИБІР КОМПОНЕНТІВ ДЛЯ РЕАЛІЗАЦІЇ ЧАТ-БОТА.....	23
2.1 Вибір мови програмування.....	23
2.2 Бібліотеки для функціонування чат-бота.....	25
2.3 База даних SQLite та створення словника	29
2.4 Dictionary API та її взаємодія з проектом.....	33
2.5 Висновок.....	38
3 РЕАЛІЗАЦІЯ ТА ДЕМОНСТРАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ВИВЧЕННЯ ІНОЗЕМНИХ МОВ	40
3.1 Формування графічного інтерфейсу.....	40
3.2 Взаємодія процесів в чат-боті	42
3.3 Демонстрація роботи чат-бота	47

КВРІСТ 101107.01.07 ІІЗ								
Зм	Арк.	Надодкум.	Підпис	Дата	Інформаційна система для вивчення іноземних мов. Пояснювальна записка	Літера	Аркуш	Аркушів
Виконав		Меданківський Р. Регіда П.Г.				п	2	58
Н.контр.		Лисенко М.С.		19.08	ХНУ ІСТ-20-1			
Затвер.		Говорушенко						

3.4 Висновок	57
ВИСНОВОК.....	60
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	62
Додаток А Копія креслення «Блок-схема алгоритму роботи чат-боту»	68
Додаток Б Копія креслення «Діаграма взаємодії користувача та розробника»..	69
Додаток В Копія креслення «Діаграма функціонування команд чат-бота»	70
Додаток Г Повний код чат-бота на Python	71

ВСТУП

Інформаційні системи в наш час відіграють важливу роль у процесі вивчення іноземних мов, зокрема англійської. У цій роботі, буде представлено систему у вигляді чат-бота, інтегрованого в месенджер Telegram, який зможе допомогти користувачам вивчати англійську мову. Чат-бот був написаний на мові програмування Python і забезпечує інтерактивне навчання, дозволяючи користувачам практикувати мовні навички, отримувати миттєві відповіді на запити. Основна мета цієї інформаційної системи - зробити процес вивчення англійської мови ефективнішим і доступнішим для широкого кола користувачів. Чат-боти в Telegram стають все більш популярними завдяки своїй зручності та доступності. Telegram, як один із найбільш використовуваних месенджерів в Україні, пропонує широкий спектр можливостей для створення інтерактивних чат-ботів, що можуть виконувати різні функції, включаючи навчання іноземних мов. Вивчення мов за допомогою чат-ботів має кілька значних переваг.

Однією з найкращих переваг чат-ботів у Telegram для вивчення іноземних мов є їхня доступність 24/7, що дозволяє користувачам навчатися в будь-який зручний для них час. Це означає, що користувачі не обмежені розкладом занять або часом роботи викладача, а можуть займатися вивченням мови тоді, коли їм зручно, наприклад, у вільні хвилини під час обідньої перерви, поїздки на роботу чи навіть пізно ввечері. Така гнучкість особливо цінна для зайнятих людей, які часто не можуть виділити час на регулярні заняття. Завдяки постійному доступу до навчальних матеріалів і можливості практикувати мовні навички в будь-який момент, чат-боти значно підвищують ефективність навчання і допомагають користувачам досягати кращих результатів у вивченні англійської мови.

Вибір мови Python для розробки чат-бота в Telegram обумовлений її простотою, гнучкістю та потужними можливостями. Python відомий своїм читабельним синтаксисом, що спрощує процес написання коду та знижує ймовірність помилок, особливо для початківців. Крім того, Python має широкий

					КвРІСТ 101107.01.07 ПЗ	Арк.
						4
Зм.	Арк.	№докум.	Підпис	Дата		

спектр бібліотек і фреймворків, таких як aiogram, які спеціально призначені для створення Telegram-ботів. Ці інструменти забезпечують простий і зручний інтерфейс для інтеграції з Telegram API.

Завдяки своїй універсальності, Python також підтримує інтеграцію з іншими системами та сервісами, що відкриває додаткові можливості для розширення функціональності чат-бота. Володіння англійською мовою відкриває широкі можливості для професійного та особистісного зростання, включаючи доступ до міжнародної освіти, перспективи кар'єрного росту в транснаціональних компаніях, а також свободу подорожувати та спілкуватися з людьми з різних країн. Вибір теми дипломної роботи, присвяченої розробці інформаційних систем вивчення англійської мови за допомогою чат-ботів, обґрунтований необхідністю практичних та ефективних методів навчання. Використовуючи такі сучасні технології, як чат-боти для обміну миттєвими повідомленнями, можна створювати інноваційні освітні інструменти, які роблять процес вивчення мови доступним для широкого кола користувачів, враховуючи індивідуальні потреби та темп життя. Поставлена мета досягається розв'язанням такої основної задачі: розробка чат-боту для Telegram, що буде застосовувати різні методи вивчення іноземних мов. Об'єктом дослідження є процес вивчення іноземних мов в чат-боті у Telegram Предметом дослідження є чат-бот у Telegram для вивчення іноземних мов.

Для досягнення поставленої мети використовуються методи високорівневого програмування на Python, застосування різноманітних бібліотек, використання API, засоби роботи із базою даних. Практичне значення є чат-бот для Telegram, що буде застосовувати різні методи вивчення іноземних мов.

					КвРІСТ 101107.01.07 ПЗ	Арк.
						5
Зм.	Арк.	№докум.	Підпис	Дата		

1 ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ ВИВЧЕННЯ ІНОЗЕМНИХ МОВ ТА ОГЛЯД ІСНУЮЧИХ ІНФОРМАЦІЙНИХ СИСТЕМ

1.1 Аналіз предметної області і виявлення наявних проблем і завдань

Аналіз предметної області інформаційних систем навчання іноземних мов виявив кілька важливих проблем і викликів. По-перше, максимальний ефект від вивчення мови може бути не досягнутий через відсутність індивідуалізації та персоналізації в навчальному процесі. По -друге, є доступний та цікавий інтерактивний ресурс, який допомагає студентам залучати студентів до активної освіти. По -третє, метод задоволення останніх вимог до вивчення іноземних мов та необхідність постійних оновлень та вдосконалення матеріалів [1-2]. Розробка інформаційної системи має вирішити ці проблеми та задовольнити вимоги користувачів у практичний та ефективний спосіб.

Існує багато різних додатків для вивчення англійської мови, таких як Duolingo, Rosetta Stone, Babbel та Coursera. Кожен з них пропонує різні методики і підходи до навчання, але вони часто мають обмежений набір функцій, не завжди інтегрованих в одному додатку. Основні проблеми, з якими стикаються користувачі, включають відсутність комплексного підходу до вивчення лексики, недостатню інтерактивність, обмежені можливості для практики вимови та контекстного використання слів. Це створює завдання розробити рішення, яке б об'єднало найкращі функціональні можливості існуючих додатків і запропонувало більш ефективний та інтерактивний інструмент для вивчення англійської мови.

Крім того, аналіз галузі інформаційної системи навчання іноземних мов вимагав врахування різноманітності методів навчання та рівнів володіння мовою студентів. Іншою проблемою є відсутність можливості контролювати прогрес учнів і адаптувати навчальні матеріали до потреб і здібностей учнів. Тому, щоб забезпечити ефективну та персоналізовану іноземну освіту, ці аспекти необхідно враховувати при проектуванні інформаційних систем. Крім того, відсутність

					КвРІСТ 101107.01.07 ПЗ	Арк.
						6
Зм.	Арк.	№докум.	Підпис	Дата		

можливостей надати учням змістовний особистий зворотний зв'язок ускладнює учням самооцінку та вдосконалення своїх мовних навичок. Ще одним ключовим викликом є необхідність інтеграції інформаційних систем з існуючими програмами та стандартами для забезпечення сумісності та відповідності вимогам освітнього сайту. При розробці інформаційних систем для вивчення іноземних мов врахування цих аспектів є запорукою успішного впровадження та використання [3-4].

Історія вивчення іноземних мов налічує тисячі років і відображає різноманітні культурні, соціальні та політичні контексти. З давніх часів люди вивчали мови інших народів, щоб полегшити спілкування, торгівлю, дипломатію та культурний обмін. У середні віки іноземні мови вивчали переважно релігійні установи та вчені для перекладу релігійних текстів і спілкування з іншими країнами. З розвитком торгівлі, подорожей і заснуванням перших університетів вивчення іноземних мов стало більш поширеним [5-6]. У сучасний час, з появою масової освіти та глобалізацією Інтернету, вивчення іноземних мов стало більш доступним та поширеним серед широкої громадськості та фахівців у різних галузях. Сьогодні вивчення іноземних мов є важливою складовою освіти та розвитку особистості, сприяючи розумінню та спілкуванню в міжнародному контексті.

Сучасні методи вивчення іноземних мов відображають сучасні тенденції педагогічної науки та використовують передові підходи та технології.

Одним із найпопулярніших методів є метод комунікації, який спрямований на розвиток комунікативних навичок через спілкування та вирішення реальних життєвих проблем і ситуацій. Крім того, активні методи навчання підкреслюють використання інтерактивних технологій та ігрових підходів для стимулювання навчання. Інші методи, такі як метод проектів, спрямовані на розвиток практичних навичок за допомогою конкретних проектів або завдань. Інтернет-інструменти та платформи, такі як відеоуроки, онлайн-курси, мобільні додатки та інтерактивні вправи, дають учням можливість вивчати мову в будь-який час,

					КвРІСТ 101107.01.07 ПЗ	Арк.
						7
Зм.	Арк.	№докум.	Підпис	Дата		

будь-де та за власним розкладом. Таким чином, сучасні методи вивчення іноземної мови поєднують традиційні підходи з інноваційними технологіями, щоб забезпечити ефективне та захоплююче навчання [7].

Рисунок 1.1 показує як влаштоване сучасні підходи для вивчення іноземних мов.



Рисунок 1.1 – Діаграма вивчення іноземних мов за допомогою сучасних технологій

Вивчення іноземних мов є важливою складовою розвитку особистості та ключовим елементом міжнародного спілкування в сучасному світі. Сучасні методи навчання, які поєднують традиційні підходи з інноваційними технологіями, надають студентам широкі можливості для ефективного і цікавого засвоєння мовного матеріалу [8]. Відкритість до вивчення нових мов сприяє культурному обміну, розумінню та толерантності між народами, що робить

вивчення іноземних мов актуальним та важливим аспектом освіти та розвитку суспільства.

1.2 Порівняльний аналіз переваг та недоліків існуючих рішень

Порівняльний аналіз існуючих ІР-рішень для вивчення іноземних мов виявляє різні підходи, їх сильні та слабкі сторони. Мобільні програми корисні для навчання в дорозі, але можуть бути обмежені у функціональності та глибині вмісту. Веб-платформи та онлайн-курси надають доступ до ширшого спектру навчальних матеріалів і можливість взаємодіяти з іншими студентами, але може знадобитися стабільне підключення до Інтернету. Програмне забезпечення для настільних ПК може пропонувати розширені функції для персоналізації процесу навчання, але може бути обмежено географічною доступністю та вимагати інсталяції на певному пристрої [9-10]. Важливо зважити ці переваги та недоліки відповідно до потреб і умов користувача, щоб ефективно вибрати найбільш прийнятне рішення.

Існують різні підходи до вивчення іноземної мови, кожен з яких має свої сильні та слабкі сторони. Наприклад, мобільні програми можуть бути зручними та доступними для користувачів у будь-який час і будь-де, але їх функціональність може бути обмеженою порівняно з більш просунутими платформами. Веб-платформи та онлайн-курси пропонують широкий вибір навчальних матеріалів і можливостей для взаємодії з іншими студентами, але може знадобитися стабільне підключення до Інтернету [11]. ІТ -програмне забезпечення, з іншого боку, може забезпечити більше індивідуального тренувального потенціалу, але обмежене географічною доступністю і вимагає встановлення на конкретних пристроях. Вибір чудової інтелектуальної власності для вивчення іноземних мов залежить від конкретних потреб та умов користувача. Основними перевагами вивчення іноземних мов за допомогою інформаційних систем є:

					КвРІСТ 101107.01.07 ПЗ	Арк.
						9
Зм.	Арк.	№докум.	Підпис	Дата		

- Гнучкий і легкий доступ до навчальних матеріалів у будь-який час і в будь-якому місці.

- Розширений вибір інтерактивних ресурсів, включаючи відеоуроки, аудіофайли, вправи та ігри, щоб допомогти вам зрозуміти та запам'ятати вміст.

- Можливість індивідуалізації навчального процесу з урахуванням особливостей і потреб кожного учня.

- Забезпечувати зворотний зв'язок і оцінювати прогрес студентів, дозволяючи відстежувати та підтримувати їх прогрес.

Недоліками використання інформаційних систем для вивчення іноземних мов є:

- Відсутність особистого контакту з викладачами може ускладнити отримання додаткової підтримки чи роз'яснень.

- Потенційні технічні проблеми, такі як погане підключення до Інтернету або несправності програмного забезпечення, можуть порушити нормальне навчання.

- Почуття ізоляції та відсутності мотивації в учнів, яким бракує особистої взаємодії з однокласниками та вчителями.

Відсутність живого усного спілкування може обмежити можливості для розвитку навичок мовлення та міжособистісного спілкування. Використання інформаційних систем у вивченні мови має такі переваги, як гнучкість і розширений доступ до навчальних ресурсів, але також і недоліки, такі як відсутність особистого контакту з викладачем мови та можливі технічні проблеми. Важливо збалансувати ці фактори та вибрати підхід до вивчення мови, який найкраще відповідає потребам і ситуації кожного учня.

Зараз існує багато інформаційних систем для вивчення англійської мови, що пропонують різні можливості для вивчення та розвитку мовних навичок. Одним із найпопулярніших прикладів є Duolingo, який надає користувачам інтерактивні вправи, відеоуроки та групові вправи для вивчення мов. Ресурси, такі як Rosetta Stone і Babbel [12-13], спеціалізуються на інтерактивних уроках англійської мови,

					КвРІСТ 101107.01.07 ПЗ	Арк. 10
Зм.	Арк.	№докум.	Підпис	Дата		

які допомагають учням вчитися через реальні ситуації розмови та практичні запитання. Такі школи, як Coursera та Udey, пропонують професійні курси англійської мови, які включають відеоуроки, інтерактивні вправи, можливості для обговорення з викладачами та однолітками тощо.

Кожна з цих інформаційних систем має унікальні особливості та підходи до вивчення мови, що дозволяє учням вибрати метод, який їм найбільше підходить.

1.2.1 Принцип роботи Duolingo

Найвідомішим застосунком для вивчення іноземних мов є Duolingo. Ця платформа є одна з найпопулярніших інформаційних систем для вивчення англійської мови, якою користуються мільйони людей у всьому світі. Ця платформа пропонує інтерактивні вправи, відеоуроки та групові завдання для ефективного навчання.

Duolingo відомий своєю зручністю та доступністю, що дозволяє користувачам вивчати мови будь-коли та будь-де за допомогою мобільного додатку та веб-платформи. Платформа пропонує широкий спектр вправ, що охоплюють різні аспекти мови, від граматики до лексики та вимови. Крім того, Duolingo використовує групові вправи та різні рівні складності, щоб дозволити користувачам вивчати мови у власному темпі та на основі їхніх потреб і рівня навичок [14-16]. Ця платформа є ефективним інструментом для самостійного вивчення англійської мови, який дозволяє учням розвивати свої розмовні навички у зручний для них спосіб. По-перше, вони повинні були зрозуміти, що саме відбувається.

Було виявлено, що в міру додавання нових функціональних можливостей продуктивність візуалізації програми знижувалася на 5-10% щомісяця. Фактично, один особливо громіздкий випуск збільшив кількість збоїв на 10%, уповільнив рендеринг кадрів на 25%, а уроки починалися на 70% повільніше на пристроях початкового рівня Одним із перших ключових висновків команди було те, що

					КвРІСТ 101107.01.07 ПЗ	Арк. 11
Зм.	Арк.	№докум.	Підпис	Дата		

використовує методи активного навчання та гейміфікації, щоб забезпечити цікаве та захоплююче навчання. Кожен користувач починає з базового рівня та отримує початкову граматику, лексику та практику вимови. Поступово користувач переходить до більш складних рівнів, а вправи стають більш різноманітними та вимогливими. Основною особливістю Duolingo є персоналізація навчання. Система враховує прогрес і помилки кожного користувача та надає вправи, адаптовані до потреб користувача та рівня володіння мовою.

На платформі також використовується система миттєвого зворотного зв'язку, тому користувачі можуть одразу побачити свої помилки та отримати рекомендації щодо подальшого навчання. Такий принцип роботи сприяє прогресивному та ефективному засвоєнню англійської мови користувачами різного рівня підготовки. Duolingo став найвідомішим додатком для вивчення іноземних мов у світі [18]. Їхнє прагнення до досконалості додатків – створення передових освітніх програм без шкоди для доступності.

1.2.2 Функціонал Rosetta Stone

Rosetta Stone — це комп'ютерна програма та веб-платформа для вивчення мови, розроблена Rosetta Stone, Inc. У цій інформаційній системі використовується багаторівневий метод поглибленого вивчення мови, який називається методом «природного занурення».

Основний принцип цього методу полягає в тому, що ви вивчаєте мову, розуміючи її в реальних життєвих ситуаціях через повсякденне спілкування, так само, як ви вивчаєте рідну мову в дитинстві. Rosetta Stone надає користувачам широкий спектр вправ і завдань, включаючи аудіо та відео, групові вправи та інтерактивні уроки. Платформа також дозволяє користувачам ефективно розвивати всі свої мовні навички, використовуючи різні методи вивчення мови, такі як аудіювання, читання, письмо та усне мовлення [19]. Rosetta Stone використовується в програмах самостійного навчання, а також в освітніх і

					КвРІСТ 101107.01.07 ПЗ	Арк. 13
Зм.	Арк.	№докум.	Підпис	Дата		

рідну мову. Цей метод сприяє розвитку навичок експресії, розуміння, читання та письма в природному середовищі.

Різниця між Rosetta Stone і Duolingo полягає в їхніх методах навчання та підходах до вивчення мови. Duolingo використовує більш структурований підхід до навчання, використовуючи певний набір вправ і завдань і зосереджуючись на інтерактивності та гейміфікації. Користувачі Duolingo насамперед навчаються за допомогою коротких вправ та ігор, заробляючи бали для досягнення нових рівнів. Rosetta Stone зосереджується на навчанні в мовних середовищах, які імітують реальні ситуації спілкування та сприяють засвоєнню природної мови без необхідності перекладу.

1.2.3 Онлайн-платформа Babbel

Babbel – це інтерактивна онлайн-платформа для вивчення мов, яка надає користувачам доступ до широкого спектру курсів з різних мов, включаючи англійську. Основний принцип дії Babbel полягає у наданні користувачам можливості вивчати мову шляхом виконання різних вправ та завдань, орієнтованих на реальні комунікації. Платформа пропонує інтерактивні вправи з читання, письма, аудіо та відео, які допомагають розвивати різні аспекти мови. Однією з головних особливостей Babbel є орієнтація на практичне використання мови в реальних ситуаціях, що сприяє ефективному навчанню та розвитку вокальних навичок. Платформа також забезпечує зручний інтерфейс користувача та персоналізовані можливості навчання, за допомогою яких кожен студент може вибрати курс.

Ці функції роблять Babbel популярним інструментом для самостійного вивчення мови, яким користуються як початківці, так і просунуті мовці. Платформа Babbel розроблена з використанням різних технологій, таких як веб, JavaScript тощо. Він надає користувачам можливість вивчати мови за допомогою різноманітних вправ, аудіо- та відеоматеріалів та інтерактивних завдань. Babbel —

					КвРІСТ 101107.01.07 ПЗ	Арк. 15
Зм.	Арк.	№докум.	Підпис	Дата		

популярний інструмент для самостійного вивчення мови, яким користуються початківці та просунуті учні в усьому світі. Babbel також використовує методи навчання, засновані на ефективних підходах до навчання, щоб допомогти користувачам ефективно вивчати мови та зберігати те, що вони вивчили. Babbel використовує різні бази даних для зберігання інформації про вас, ваш прогрес у вивченні мови, вправи та завдання, які ви виконали, та інші дані, необхідні для процесу навчання. Ці бази даних можуть використовувати різні технології, включаючи реляційні бази даних (наприклад, MySQL, PostgreSQL), бази даних NoSQL (наприклад, MongoDB) або гібридні рішення, які поєднують реляційний і NoSQL підходи до зберігання даних. Вибір конкретної бази даних залежить від потреб і вимог проекту, а також від швидкості та масштабованості, які забезпечує база даних. Рисунок 1.4 показує переваги застосунка Babbel перед іншими застосунками для вивчення іноземних мов.

	Babbel App	Other Premium Learning Apps	Freemium Apps
Speech Recognition Technology	✓	✓	✓
Games To Mix Up Your Learning	✓	✓	—
Courses Designed By Language Experts	✓	✓	—
Live Video Classes With Expert Teachers	✓	✓	—
Explicit Grammar Help	✓	✓	—
No Ads	✓	✓	—
In-App Podcasts To Supplement Lessons	✓	—	—
Learning Tailored To Your Native Language	✓	—	—
Culture Bites For Speaking Like A Native	✓	—	—

Рисунок 1.4 – Переваги застосунка Babbel перед іншими платними та безкоштовними застосунками

Перевагою Babbel є його ефективність і простота використання. Платформа пропонує структуровані курси різними мовами, що дозволяє користувачам вивчати мови різного рівня складності, від початкового до просунутого. Вправи та завдання Babbel зосереджені на реальних ситуаціях спілкування та сприяють швидкому й ефективному засвоєнню мови.

Платформа також пропонує персоналізоване навчання, де кожен користувач може вибрати курси, які відповідають його потребам і мовним навичкам. Крім того, Babbel має зручний інтерфейс і може використовуватися на різних пристроях, що дозволяє вивчати мови в будь-який час і в будь-якому місці. Ці переваги роблять Babbel популярним інструментом для самостійного вивчення мови як для початківців, так і для досвідчених лінгвістів [21].

1.2.4 Аналіз функціоналу Coursera

Coursera – це навчальна онлайн-платформа, яка співпрацює з університетами та організаціями по всьому світу, щоб надати користувачам доступ до різноманітних курсів у різних областях навчання, включаючи англійську мову [22-24]. Coursera пропонує безкоштовні та платні курси різного рівня складності – від початківців до просунутих. Платформа не лише надає вам можливість вивчати мову за допомогою відеоуроків, текстових матеріалів, вправ і тестів, але також дозволяє вам спілкуватися з викладачами та іншими студентами через форуми та онлайн-спільноти. Coursera також пропонує сертифікати та дипломи про завершення курсу, що дозволяє користувачам отримати підтвердження своїх знань і навичок у вивченні англійської мови. Coursera працює як платформа для навчання та розробки курсів, використовуючи різноманітні технології та мови програмування. Це включає такі веб-технології, як HTML, CSS і JavaScript для інтерфейсу користувача та взаємодії з користувачем. Крім того, бізнес-логіка та серверна частина платформи можуть використовувати

					КвРІСТ 101107.01.07 ПЗ	Арк. 17
Зм.	Арк.	№докум.	Підпис	Дата		

англійську, що дає користувачам можливість вивчати мови за допомогою відеокурсів, текстових матеріалів, вправ і тестів, а також можливість спілкуватися з викладачами та іншими. Спілкуйтеся зі студентами через онлайн-форуми та спільноти. Цей широкий набір функцій і партнерство з провідними навчальними закладами роблять Coursera чудовим вибором для тих, хто хоче отримати якісну освіту від експертів у різних галузях [28].

Для порівняння функціоналу додатків була створена таблиця 1.1.

Таблиця 1.1 – Порівняння функціоналу додатків

Застосунок	Інтерактивні уроки	Геймфікація	Соціальні функції	Офлайн режим	Вимова
Duolingo	+	+	+	-	+
Rosseta stone	+	-	-	+	+
Babbel	+	-	-	+	+
Coursera	+	-	+	+	+

Висновок по таблиці додатків для вивчення мов. Згідно з таблицею видно, що кожен застосунок має свої унікальні переваги та недоліки. Duolingo виділяється своєю гейміфікацією та безкоштовним доступом, хоча йому бракує офлайн режиму та детальних граматичних пояснень.

Rosetta Stone відома своїм методом повного занурення, чудовими можливостями для вимови та офлайн доступом, але вона має високу вартість і обмежені соціальні функції. Babbel пропонує структуровані курси, детальні граматичні пояснення та підтримку вимови, однак підписка на нього також є дорогою [29]. Coursera забезпечує різні рівні складності, багатий аудіо та відео контент, підтримку спільноти та офлайн доступ, але має відносно високу вартість.

Усі додатки підтримують інтерактивні уроки та різні рівні адаптивності до знань користувача, що робить їх корисними для різних рівнів володіння мовою. Також усі чотири додатки пропонують підтримку різних мов, що робить їх універсальними інструментами для вивчення мов [30].

Таким чином, вибір конкретного додатку залежить від індивідуальних потреб користувача: чи то інтерактивність, офлайн доступ та гейміфікація, чи то потреба в глибокому зануренні та якісній вимові, чи то детальні граматичні пояснення та підтримка спільноти.

1.3 Постановка задачі для подальшої роботи

Необхідно розробити Telegram чат-бота на мові Python з використанням бібліотеки aiogram, який допоможе користувачам у вивченні нових слів. Основною метою цього бота є забезпечення комплексного підходу до вивчення лексики, роблячи процес навчання більш ефективним та інтерактивним.

Цей бот буде надавати користувачам широкий спектр функціональних можливостей для покращення їх мовних навичок. Він зможе надавати визначення слів разом з їх транскрипціями, що допоможе користувачам краще зрозуміти вимову та значення нових слів.

Крім того, бот надаватиме приклади використання слів у реченнях, що дозволить користувачам побачити, як ці слова використовуються в контексті, та покращити їхні навички побудови речень. Однією з ключових функцій буде можливість знаходити синоніми до введених слів, що допоможе користувачам розширити свій словниковий запас та знайти більш відповідні слова для різних контекстів. Додатково, бот надаватиме можливість прослухати аудіофайли з правильною вимовою слів, що є важливим аспектом у вивченні нової мови. Вимова слів буде озвучена носіями мови або високоякісними текст-до-мови (TTS) сервісами, що забезпечить точність та автентичність вимови.

					КвРІСТ 101107.01.07 ПЗ	Арк. 20
Зм.	Арк.	№докум.	Підпис	Дата		

Використовуючи можливості мови програмування Python та бібліотеки aiogram, цей бот буде швидким, надійним та зручним у використанні. Бібліотека aiogram дозволяє легко інтегрувати функції Telegram у Python проекти, що забезпечить високу швидкість відповіді та стабільність роботи бота. Крім того, Python, як мова програмування, надає широкі можливості для обробки даних та інтеграції з різними API сервісами, що розширить функціонал бота та зробить його більш гнучким та потужним.

Завдяки такому комплексному підходу до вивчення лексики, користувачі зможуть значно покращити свої мовні навички, збагатити свій словниковий запас та удосконалити вимову. Бот стане надійним помічником у процесі вивчення нових слів, роблячи цей процес цікавим та інтерактивним.

Цілями роботи є:

- Аналіз потреб користувачів і визначення вимог до IS.
- Розробка концепцій ІБ та функціональних вимог.
- Проектування інтерфейсу користувача та структури бази даних.
- Впровадження програмного забезпечення на основі певних вимог.
- Тестування ІР на функціональність і відповідність вимогам.
- Впровадження інформаційної системи та навчання користувачів.
- Підтримка та розвиток системи з урахуванням мінливих потреб користувачів.

1.4 Висновок

У першому розділі було проведено детальне дослідження існуючих методів та інструментів для вивчення англійської мови, зокрема за допомогою цифрових технологій а саме додатків. Аналіз показав, що більшість доступних рішень мають обмежений функціонал або складний інтерфейс, що ускладнює їх використання для користувачів а також обмежений функціонал у безкоштовних версіях додатків. Основні проблеми включали в себе недостатню інтерактивність,

					КвРІСТ 101107.01.07 ПЗ	Арк. 21
Зм.	Арк.	№докум.	Підпис	Дата		

обмежений доступ до актуальної інформації, відсутність аудіо супроводу для правильної вимови та труднощі в додаванні нових слів до навчальних матеріалів. Враховуючи ці недоліки, були визначені ключові завдання для нового чат-бота: створити зручний та функціональний Telegram бот, який забезпечить користувачів комплексним набором інструментів для вивчення англійської мови. Бот повинен підтримувати визначення слів, транскрипції, приклади використання, синоніми та аудіовимову, а також надавати можливість додавати нові слова до бази даних. Це дозволить створити інтерактивне та ефективне середовище для самостійного вивчення мови, враховуючи сучасні потреби та очікування користувачів.

Повна функціональність чат-бота буде доступна усім користувачам безкоштовно, з необмеженою кількістю запитів чат-бота та влюбий для них зручний час.

					КвРІСТ 101107.01.07 ПЗ	Арк.
						22
Зм.	Арк.	№докум.	Підпис	Дата		

2 ВИБІР МОВИ ПРОГРАМУВАННЯ ТА ВИБІР КОМПОНЕНТІВ ДЛЯ РЕАЛІЗАЦІЇ ЧАТ-БОТА

2.1 Вибір мови програмування

Найпопулярнішою мовою для створення чат-ботів є мова програмування Python, тим самим, вибір мови програмування був очевидний. Python [30-32] є мовою програмування яка є однією з найпопулярніших мов програмування у світі завдяки своїй простоті, гнучкості та потужним функціям. Python має зрозумілий синтаксис, що робить його доступним як для початківців, так і для досвідчених розробників. Python підтримує кілька парадигм програмування, включаючи об'єктно-орієнтоване, процедурне та функціональне програмування, що робить його універсальним для різних типів завдань. Крім того, Python має велику та активну спільноту, яка допомагає розробити багату екосистему бібліотек та структур, таких як Django для веб -розробки, TensorFlow для машинного навчання та Aiogram для створення чатів у Telegram. Завдяки цим функціям Python є чудовим вибором для розробки сучасних програмних рішень, таких як інтерактивні системи навчання [33].

Ще одна перевага Python у широкій підтримці та активному розвитку спеціалізованих інструментів для конкретної галузі. Python має велику кількість бібліотек, які спеціалізуються на взаємодії з Telegram API та дозволяють ефективно впроваджувати різноманітний функціонал для чат-ботів. Більше того, Python активно використовується в галузі машинного навчання, що відкриває широкі можливості для розширення можливостей бота, наприклад, за допомогою інтелектуальних систем відповідей або аналізу тексту користувача для персоналізованих рекомендацій.

Такий різноманітний та потужний інструментарій дозволяє швидко реалізувати складний функціонал і забезпечує високий рівень якості та ефективності чат- бота. Порівнюючи код у Python з кодом в інших мовах

					КвРІСТ 101107.01.07 ПЗ	Арк.
						23
Зм.	Арк.	№докум.	Підпис	Дата		

програмування, Видні чіткі переваги у його зручності та компактності у зрівняні з іншими мовами програмування [34]. Python відомий своєю простотою та лаконічністю, що дозволяє зменшити кількість рядків коду, не втрачаючи при цьому функціональності. Це особливо важливо для розробки чат-ботів, де кожен рядок коду має значення, а збільшення його кількості може призвести до складностей у розумінні та обслуговуванні проекту.

Python ставить акцент на читабельність коду, що робить його більш зрозумілим для програмістів. Чітка структура та інтуїтивний синтаксис сприяють тому, щоб код був легко читабельний та підтримуваний. Підключення Telegram бота до проекту на Python за допомогою токена бота - це швидкий та простий процес, який забезпечує легку інтеграцію бота зі створеним програмним кодом. Після отримання токена бота від BotFather у Telegram, достатньо лише кількох рядків коду, щоб запустити та налаштувати зв'язок між ботом і проектом на Python [35]. Використовуючи бібліотеки для роботи з Telegram, можна створити об'єкт бота, використовуючи отриманий токен, і визначити реакцію бота на різні команди та повідомлення в Telegram (рис. 2.1).

Цей простий процес дозволяє швидко почати роботу з чат-ботом та швидко інтегрувати його у будь-який проект на Python [36-37].

JAVA	VS	PYTHON
<pre>public class JavaPyramid 1 { public static void main(String[] args) { for(int i=1; i<=5; i++) { for(int j=0; j<i; j++) { System.out.print(" * "); } //generate a new line System.out.println(" "); } } }</pre>		<pre>def create_pyramid (rows): for i in range (rows): print ('*' * (i+1))</pre>

Рисунок 2.1 – Порівняння міні-програми, яка малює піраміду, на Java і Python.

Створення команд для Telegram бота на Python - це простий та зрозумілий процес, який дозволяє легко реалізувати різноманітний функціонал бота. За допомогою бібліотек, таких як aiogram або python-telegram-bot, можна визначити обробники для різних команд і повідомлень, які отримує бот. Наприклад, за допомогою декораторів у Python можна вказати, яку функцію викликати при отриманні певної команди в Telegram [38-39]. Це дозволяє легко створювати та організовувати код, реагуючи на різні команди користувачів з Telegram. При цьому, Python надає широкий спектр можливостей для обробки даних, роботи з базами даних, виконання операцій над текстом та іншими завданнями, що робить створення команд для Telegram бота максимально гнучким та зручним.

Підводячи висновки про мову програмування можна сказати що Python є найкращим вибором для розробки чат-бота в Telegram через його простоту, ефективність і гнучкість. Завдяки чіткому синтаксису, потужним бібліотекам і багатим ресурсам можна швидко створювати функціональні та надійні чат- боти, які забезпечують високоякісну взаємодію з користувачем. Python — це не лише інструмент програмування, а й ефективний інструмент для впровадження креативності та інновацій у індустрію чат-ботів.

Усі унікальні переваги Python, включаючи простоту, читабельність і широкий вибір бібліотек, роблять його неперевершеним інструментом для розробки чат- ботів у Telegram. Використання Python спрощує процес розробки та підтримки чат-ботів, дозволяючи розробникам швидко впроваджувати нові функції та покращувати наявні, забезпечуючи високу якість взаємодії з користувачем і звільняючи ресурси розробника [40].

2.2 Бібліотеки для функціонування чат-бота

Бібліотеки, такі як aiogram та python-telegram-api, є важливими інструментами для розробки чат-ботів у Telegram на Python. Aiogram надає

					КвРІСТ 101107.01.07 ПЗ	Арк.
						25
Зм.	Арк.	№докум.	Підпис	Дата		

зручний та потужний інтерфейс для взаємодії з Telegram API, дозволяючи створювати чат-ботів з різноманітним функціоналом, від простих відповідей до складних інтерактивних операцій. З іншого боку, бібліотека python-telegram-api також дозволяє легко взаємодіяти з Telegram API, але забезпечує більш прямий доступ до методів API для тих, хто шукає більш гнучкий підхід до розробки [41]. Обидві бібліотеки дозволяють ефективно створювати та налаштовувати чат-ботів у Telegram, забезпечуючи розробникам широкі можливості для реалізації будь-яких ідей та завдань.

Aiogram – це асинхронний фреймворк для створення ботів Telegram на Python. Він заснований на бібліотеці Asyncio, яка дозволяє писати асинхронний код, здатний виконувати багато операцій одночасно, не блокуючи основний потік. Aiogram надає простий і зрозумілий API для роботи з Telegram Bot API, що робить процес розробки швидким і зручним [42-43].

Однією з головних переваг Aiogram є його асинхронність, що дозволяє значно підвищити продуктивність бота. asyncio дозволяє обробляти декілька запитів користувачів одночасно. Це важливо для ботів, які очікують великої кількості взаємодій. Це робить Aiogram ідеальним вибором для створення програм із високим навантаженням. Крім того, Aiogram має добре структуровану документацію та безліч прикладів, що допомагає швидко розібратися з основними концепціями та розпочати розробку. Бібліотека підтримує різні типи оновлень від Telegram, включаючи повідомлення, команди, кнопки callback та інші. Це дозволяє гнучко розробляти роботів із різними функціями [44].

Python-Telegram-Bot — це потужна та добре підтримувана бібліотека для розробки чат-ботів Telegram мовою програмування Python [45].

Ця бібліотека надає зручний інтерфейс для взаємодії з API Telegram, дозволяючи розробникам легко створювати, налаштовувати та розгортати чат-боти в Telegram.

					КВРІСТ 101107.01.07 ПЗ	Арк.
						26
Зм.	Арк.	№докум.	Підпис	Дата		

Python-Telegram-Bot пропонує розробникам широкий спектр функцій, включаючи можливість обробки різних типів повідомлень, роботи з клавіатурами та кнопками, а також керування груповими чатами та підписниками.

Крім того, бібліотека має добре задокументований API та активну спільноту розробників, що робить процес розробки чат-бота швидким та ефективним.

Python-Telegram-Bot також підтримує різні функції Telegram API, включаючи надсилання повідомлень, роботу з медіа-файлами, використання онлайн-режиму тощо.

Це дозволяє розробникам створювати різноманітні та потужні чат-боти, які відповідають різноманітним потребам користувачів.

Також важливим є підключення до проекту Бібліотеки Requests.

Вона є однією з найпопулярніших бібліотек для здійснення HTTP-запитів у Python. Вона забезпечує зручний та простий інтерфейс для взаємодії з веб-ресурсами, включаючи відправку та отримання даних через HTTP-протокол [46-47].

За допомогою бібліотеки Requests можна легко виконувати різні види запитів, такі як GET, POST, PUT, DELETE тощо, та обробляти їхні відповіді.

Вона також підтримує роботу з параметрами запитів, заголовками, файлами, кукісами та іншими елементами HTTP-запитів, що робить її досить гнучкою для різних сценаріїв взаємодії з веб-ресурсами.

Бібліотека Requests особливо корисна для розробки чат-ботів, оскільки вона дозволяє ботам взаємодіяти з зовнішніми веб-сервісами та API, наприклад, для отримання даних або відправки повідомлень.

Вона допомагає забезпечити ефективну та надійну комунікацію між ботом та зовнішніми ресурсами, що розширює можливості функціоналу чат-бота та робить його більш інтерактивним і корисним для користувачів [48].

Принцип роботи команд чат-бота подано на рис. 2.2.

					КВРІСТ 101107.01.07 ПЗ	Арк.
						27
Зм.	Арк.	№докум.	Підпис	Дата		

2.3 База даних SQLite та створення словника

База даних SQLite дозволяє розширити функціонал Telegram бота, забезпечуючи зберігання та обробку великої кількості інформації. Для проекту було використано реляційну базу даних, яка дозволила організувати та зберігати дані про слова які користувачі можуть вносити в базу даних. Це забезпечує персоналізовані взаємодії з кожним користувачем, підтримуючи історію своєї діяльності та прогрес мови [49]. Завдяки базам даних бот надає більш точну та відповідну відповідь, аналізує дії користувача та оптимізує процес навчання. Інтеграція в проект бази даних може легко підвищити функції бота, додавати нові можливості та покращити існуючі можливості.

Була вибрана вбудована база даних SQLite яка є чудовим рішенням для Telegram бота, оскільки вона легко інтегрується з Python і забезпечує просте та ефективно зберігання даних. SQLite є легковаговою реляційною базою даних, яка не вимагає окремого серверного програмного забезпечення, що робить її ідеальною для невеликих та середніх проектів. Вона підтримує стандартний SQL, що дозволяє виконувати складні запити та маніпулювати даними. Використання SQLite в проекті дозволило організувати зберігання інформації про користувачів, їхні запити та результати навчання, забезпечуючи швидкий доступ та обробку даних. Крім того, простота налаштування та використання SQLite значно прискорила процес розробки та дозволила зосередитися на створенні основного функціоналу Telegram бота.

Функціонування бази даних у чат-боті забезпечує зручне зберігання та використання слів для перекладу. Користувач може додавати нові слова до бази даних за допомогою команди /add, спочатку вказуючи англійське слово, а потім його український переклад. Ці дані зберігаються у базі даних SQLite, забезпечуючи надійне зберігання та швидкий доступ до інформації. Коли користувач потребує перекладу слова, він може скористатися командою /translate, і бот миттєво знаходить відповідний переклад у базі даних та надає його

					КвРІСТ 101107.01.07 ПЗ	Арк.
						29
Зм.	Арк.	№докум.	Підпис	Дата		

користувачу [50]. Такий підхід забезпечує ефективне та зручне керування словником, дозволяючи користувачам легко додавати нові слова та отримувати переклади за запитом.

Функціонування бази даних у чат-боті дозволяє будь-якому користувачу вносити нові слова з перекладом, які зберігаються у системі. Це забезпечує колективне наповнення бази даних, роблячи її постійно зростаючою та актуальною. Всі додані слова зберігаються в базі даних SQLite, що гарантує надійне та швидке зберігання інформації. Завдяки цьому, кожне нове слово, введене користувачем, зберігається і стає доступним для всіх інших користувачів бота, що значно підвищує ефективність та інтерактивність процесу вивчення мови. Цей підхід дозволяє створити спільну базу знань, яка сприяє обміну лінгвістичною інформацією та збагаченню словникового запасу всіх користувачів [51].

Важливо зауважити, що всі слова в базу даних слід додавати, починаючи з великої літери. Це забезпечує коректний пошук і уникнення проблем із пошуком перекладів. Якщо користувач вводить слово з маленької літери, під час пошуку воно може не знайтися через невідповідність регістру. Таким чином, для забезпечення точності та коректності роботи бота, користувачам необхідно дотримуватись цього правила при додаванні нових слів до бази даних, що сприятиме правильному та швидкому знаходженню перекладів у подальшому.

Колекція бази даних, реалізована за допомогою SQLite, містить англійські слова та їх переклад українською мовою. Колекція являється ключовим елементом функціонування чат-бота, забезпечуючи швидкий і зручний доступ до необхідної лексичної інформації. Структура бази даних включає таблиці, де кожне слово англійською мовою відповідає своєму перекладу українською, що дозволяє користувачам легко отримувати потрібні переклади у режимі реального часу. Такий підхід до зберігання даних гарантує точність і актуальність інформації, надаючи користувачам можливість ефективно вивчати нові слова та покращувати свої мовні навички. [52-53] Проект використовує лише одну

					КвРІСТ 101107.01.07 ПЗ	Арк. 30
Зм.	Арк.	№докум.	Підпис	Дата		

колекція для бази даних, а саме - words. Вона складається з двох полів: “english” та “ukrainian”, які відповідно зберігають англійські слова та їхні українські переклади.

Інтеграція бази даних з чат-ботом здійснюється за допомогою бібліотеки SQLite, що дозволяє ефективно управляти збереженням і доступом до лексичних даних [54]. Бібліотека SQLite, яка використовується в цьому проекті, є потужним і легким у використанні інструментом для роботи з базами даних. SQLite забезпечує повноцінну реляційну базу даних у вигляді компактного бібліотечного файлу, що робить її ідеальною для вбудованих додатків, таких як чат-боти. Вона не потребує окремого серверного програмного забезпечення, що значно спрощує її інтеграцію та використання.

SQLite підтримує всі стандартні операції бази даних, такі як створення, читання, оновлення і видалення записів, що дозволяє ефективно зберігати і управляти даними.

Висока продуктивність, надійність і простота використання роблять цю бібліотеку ідеальним вибором для забезпечення швидкого доступу до лексичних ресурсів та інших даних, необхідних для функціонування чат-бота.

Завдяки SQLite, чат-бот може надавати користувачам точні і швидкі відповіді на їхні запити, покращуючи загальний досвід взаємодії з системою. [55-56].

У проекті для запису даних до бази даних SQLite використовується SQL-запит “INSERT”, який виконується за допомогою методу “execute”. Цей метод дозволяє інтегрувати нові записи у відповідні таблиці бази даних.

Процес виглядає наступним чином: спочатку створюється SQL-запит у вигляді рядка, де визначається таблиця та поля, в які будуть додані нові дані. Далі, за допомогою методу “execute” з бібліотеки SQLite, цей запит виконується з передачею необхідних значень для відповідних полів.

Наприклад, “cursor.execute(“INSERT INTO words (english_word, ukrainian_translation) VALUES (?, ?)”, (english_word, ukrainian_translation))” додає

					КвРІСТ 101107.01.07 ПЗ	Арк. 31
Зм.	Арк.	№докум.	Підпис	Дата		

нове слово з його перекладом до таблиці words.db. Використання методу “execute” забезпечує надійний та ефективний спосіб додавання нових даних до бази даних, що є критичним для підтримки актуальності та повноти лексичних ресурсів, які використовує чат-бот.

Метод “cursor.execute” використовується для виконання різноманітних SQL-запитів. Спочатку він застосовується для створення таблиці “words.db” з полями “id”, “english” та “ukrainian”, де “id” є первинним ключем з автоматичним збільшенням. Потім цей метод використовується у функції “add_word” для вставки нових записів у таблицю “words.db”, приймаючи англійське слово та його український переклад.

Метод “conn.commit” застосовується для збереження змін у базі даних після виконання команд вставки або створення таблиці.

У функції “get_translation” метод “cursor.execute” виконує запит на вибірку українського перекладу для заданого англійського слова, а метод “cursor.fetchone” повертає перший результат цього запиту. Ці методи забезпечують ефективну взаємодію з базою даних, дозволяючи створювати структури для зберігання даних, додавати нові записи та здійснювати пошук необхідної інформації.

База даних SQLite, яка використовується у проекті, є корисним інструментом для зберігання та управління лексичними даними. Вона дозволяє ефективно створювати таблиці, додавати нові дані та швидко виконувати запити для отримання необхідної інформації.

Використання SQLite у поєднанні з Python дозволяє забезпечити стабільність, швидкість і надійність функціонування чат-бота, забезпечуючи при цьому простоту установки та налаштування.

SQLite допомагає чат-боту перейти на кращий рівень для підвищення якості вивчення іноземних мов, завдяки швидкому доступу до перекладу англійських слів українською.

2.4 Dictionary API та її взаємодія з проектом

Dictionary API - це інтерфейс програмування застосунків, який надає доступ до словникової бази даних, що містить визначення, синоніми, антоніми та інші лінгвістичні дані для слів. Цей API дозволяє розробникам інтегрувати словниковий функціонал у свої програми та додатки, щоб забезпечити користувачам можливість швидко та зручно знаходити визначення та іншу інформацію про слова. Використання API Dictionary API може бути корисним для створення навчальних додатків, перекладачів, редакторів тексту та інших програм, які потребують доступу до лексичних даних (рис. 2.3).

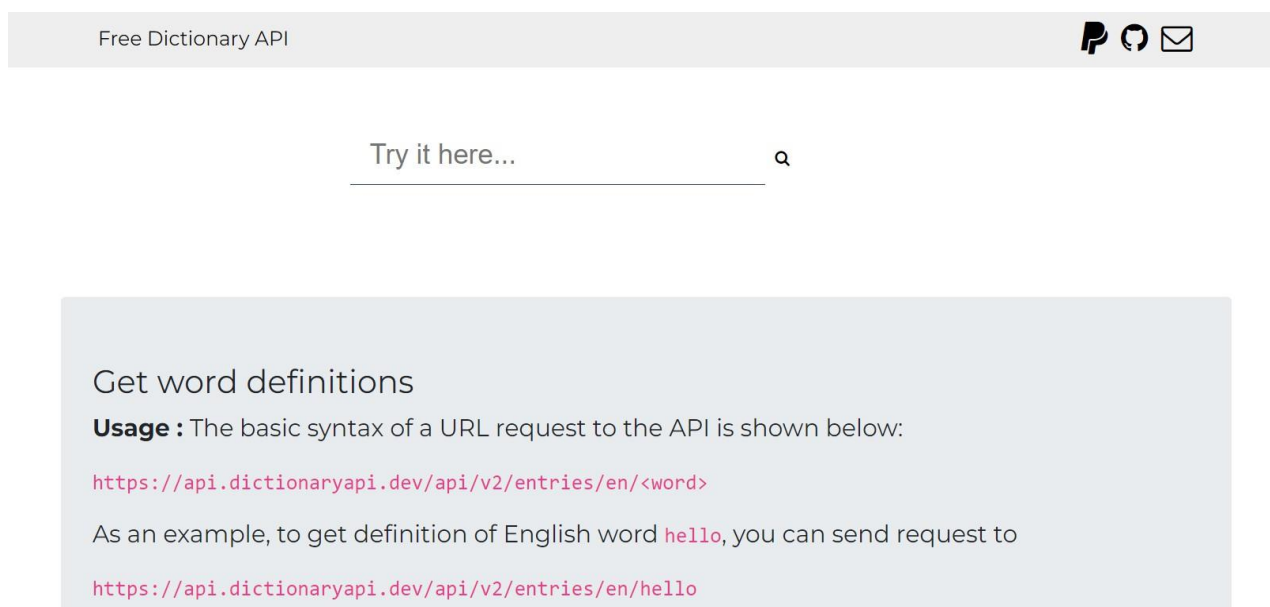


Рисунок 2.3 – Вигляд головної сторінки Dictionary API

Це API відіграє важливу роль у проекті, оскільки дозволяє забезпечити користувачам доступ до широкого спектру лінгвістичних даних та функціоналу, пов'язаного зі словниковими операціями. За допомогою цього API чат-бот зможе надавати користувачам можливість отримувати визначення слів, перевіряти синоніми та антоніми, а також вивчати різноманітні лінгвістичні дані, не виходячи з додатку. Це робить проект більш цікавим та корисним для

					КВРІСТ 101107.01.07 ПЗ	Арк. 33
Зм.	Арк.	№докум.	Підпис	Дата		

користувачів, допомагаючи їм розширити свій словниковий запас та покращити навички мови.

Функціонал цієї API дозволяє користувачам отримувати визначення значень слів, приклади їх використання в реченнях, а також здійснювати пошук синонімів. Ще однією важливою можливістю, яку надає це API, є можливість отримання аудіофайлів з промовою конкретних слів, що робить чат-бот більш інтерактивним та корисним для користувачів, які хочуть не лише знаходити визначення слів, але й вивчати їх вимову та використання в реальних ситуаціях [60].

Визначення слів - Команда надає користувачам можливість одержувати визначення та транскрипцію англійських слів. Після виклику команди /definition, користувач може відправити слово, для якого він бажає отримати визначення. Бот обробляє це слово та намагається знайти відповідне визначення та транскрипцію. Якщо слово знайдено, бот надсилає користувачеві визначення та транскрипцію. Ця функція корисна для тих, хто вивчає англійську мову та потребує швидкого доступу до перекладу та визначення слів.

Приклад використання слова - Команда /example надає користувачам можливість отримати приклад використання заданого слова в англійській мові. Після активації цієї команди користувач може відправити слово, для якого він бажає отримати приклад вживання. Бот обробляє запит та намагається знайти контекст, в якому це слово може бути використане. Після успішного пошуку бот надсилає користувачеві приклад використання слова в реальних англомовних реченнях. Ця функція допомагає користувачам краще зрозуміти контекст та ситуації, в яких можуть вживатися англійські слова, що сприяє покращенню їхнього розуміння мови.

Команда /audio в Telegram боті дозволяє користувачам отримати аудіовимову конкретного англійського слова, що сприяє покращенню їхньої вимови. Коли користувач вводить команду /audio разом із потрібним словом, бот відправляє запит до Dictionary API для отримання даних про це слово. API

повертає JSON-відповідь, що містить посилання на аудіофайл з правильною вимовою.

Команда пошуку синонімів - Команда /synonimus дозволяє користувачам швидко знаходити альтернативні слова з подібним значенням для вже введеного слова. Ця функція особливо корисна для тих, хто хоче розширити свій словниковий запас або знайти більш різноманітні варіанти вираження своїх думок. Команда пошуку синонімів сприяє покращенню мовленнєвих навичок та допомагає збагатити мову виразного засобу, що робить проект ще більш корисним для користувачів, які прагнуть вдосконалити своє володіння мовою.

Взаємодія з API є ключовою частиною функціоналу Telegram бота. Бот використовує методи цього API для отримання різноманітної інформації про слова, яку подальше надає користувачам. Для цього бот використовує різні HTTP-запити до API, такі як запити GET для отримання визначень, синонімів, прикладів використання та аудіофайлів з промовою. Після отримання відповіді від API бот обробляє отримані дані та формує відповідь у зручному форматі для користувача, який відправляється у чат. Цей процес відбувається автоматично при виклику відповідних команд користувачем у чаті з ботом, що забезпечує швидке та зручне отримання необхідної лінгвістичної інформації безпосередньо у месенджері Telegram (рис. 2.4).

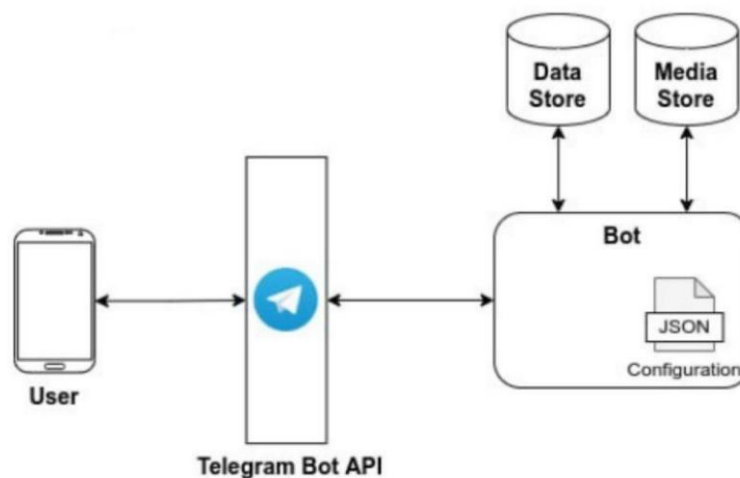


Рисунок 2.4 – Взаємодія JSON з Чат-Ботом

У процесі взаємодії Telegram бота з Dictionary API використовується формат даних JSON (JavaScript Object Notation). Коли бот надсилає запит до API, він отримує відповідь у вигляді JSON-об'єкту, який містить інформацію про слово, таку як його визначення, транскрипція, приклади використання в реченнях, синоніми та посилання на аудіофайли з вимовою. Ці дані бот аналізує та витягує необхідну інформацію для подальшої відправки користувачеві. Використання JSON забезпечує зручність обміну даними та їх обробки, оскільки цей формат є легким для читання і роботи з ним за допомогою стандартних бібліотек Python, таких як json. Завдяки цьому бот може швидко і ефективно взаємодіяти з API, забезпечуючи користувачів актуальною і точною інформацією.

API - це важлива складова у функціонуванні Telegram бота, яка дозволяє користувачам отримувати лінгвістичну інформацію про слова безпосередньо у чаті. Завдяки цьому API, бот може автоматично отримувати визначення, приклади використання, синоніми та навіть аудіофайли з промовою для введених слів. JSON дозволяє легко парсити та обробляти дані за допомогою вбудованих бібліотек Python, таких як json.

Це спрощує інтеграцію API з ботом, роблячи процес отримання та відправлення інформації швидким і надійним. Завдяки використанню JSON, проект отримує значну гнучкість і масштабованість, що дозволяє легко додавати нові функціональні можливості та підтримувати актуальність даних для користувачів. Всі ці компоненти роблять чат-бот корисним інструментом для вивчення мови та розширення словникового запасу. Загалом, використання Dictionary API покращує користувацький досвід та функціональність Telegram бота.

Найкращим вибором середовища розробки для чат-бота став PyCharm. PyCharm надає всі необхідні інструменти для ефективної роботи з ними. PyCharm, як потужне середовище розробки (IDE) для Python, пропонує інтеграцію з різноманітними API, що дозволяє розробникам легко підключатися до зовнішніх сервісів, отримувати та обробляти дані. Використовуючи вбудовані функції

					КвРІСТ 101107.01.07 ПЗ	Арк. 36
Зм.	Арк.	№докум.	Підпис	Дата		

PyCharm, такі як автоматичне завершення коду, інтерактивний відлагоджувач та зручне управління пакетами, розробники можуть швидко інтегрувати API у свої проекти. PyCharm також підтримує роботу з популярними бібліотеками для запитів до API, такими як requests та aiohttp, що спрощує процес створення запитів та обробки відповідей. Завдяки цим можливостям, розробка додатків, які взаємодіють з API, стає більш продуктивною та зручною (рис. 2.5).

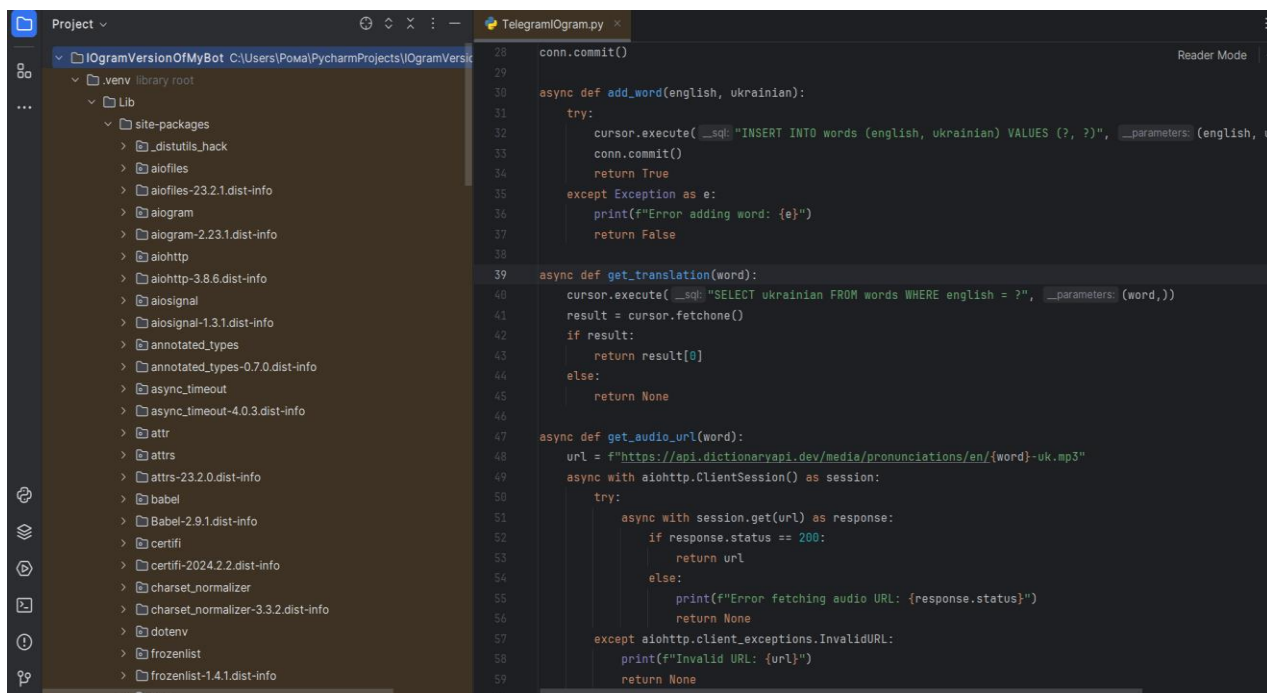


Рисунок 2.5 – Інтерфейс PyCharm

Вибір PyCharm замість Visual Studio для проекту полягає з декількох причин. PyCharm спеціалізується на розробці з використанням мови Python і надає широкий спектр інструментів, які значно підвищують продуктивність. Це середовище розробки має потужний інтерфейс для автоматичного завершення коду, інтеграцію з системами контролю версій, такими як Git, та підтримку для написання і відлагодження тестів. Крім того, PyCharm пропонує відмінні можливості для роботи з віртуальними оточеннями та управління залежностями проекту. Інструменти аналізу коду та інтеграція з різноманітними бібліотеками роблять розробку більш ефективною та зручною. Visual Studio, хоча і є потужним

інструментом для багатьох мов програмування, не забезпечує такої ж спеціалізованої підтримки та оптимізації для Python, як PyCharm. Тому вибір на користь PyCharm дозволяє зосередитися на розробці функціоналу проекту без зайвих труднощів, що значно спрощує процес створення якісного програмного забезпечення. Крім того, PyCharm підтримує роботу з віртуальними оточеннями та має зручний інтерфейс для управління залежностями проекту.

Інструменти аналізу коду допомагають виявляти потенційні проблеми та оптимізувати код, а інтеграція з популярними фреймворками та бібліотеками, такими як Django, Flask та NumPy, дозволяє швидко розпочати розробку складних проектів. Завдяки цим можливостям, PyCharm є ефективним і зручним інструментом для розробки на Python, забезпечуючи високу продуктивність і якість програмного забезпечення.

2.5 Висновок

При завершенні вибору компонентів та мови програмування для розробки чат-бота потрібно підвести підсумки. Мова програмування Python була обрана як основна мова програмування завдяки своїй простоті, читабельності та великій кількості доступних бібліотек, які полегшують інтеграцію різних функцій. Використання бібліотеки aiogram для взаємодії з Telegram API дозволяє швидко та ефективно створювати бота з багатим функціоналом. PyCharm було обрано як середовище розробки завдяки його потужним інструментам для автозавершення коду, відлагодження та управління залежностями, що забезпечує зручність і продуктивність роботи. Додатково, інтеграція з Dictionary API дозволить надавати користувачам визначення слів, транскрипції, приклади використання, синоніми та аудіовимову, що зробить процес вивчення нових слів ефективнішим та інтерактивнішим.

Такий комплексний підхід до вибору мови програмування та компонентів забезпечить створення швидкого, надійного та функціонального чат-бота для

					КВРІСТ 101107.01.07 ПЗ	Арк.
						38
Зм.	Арк.	№докум.	Підпис	Дата		

вивчення лексики. Це гарантує високу якість та користувацьке задоволення від використання бота.

Використання бази даних SQLite для чат-бота є важливим компонентом, що забезпечує зберігання та управління даними користувачів. SQLite обрана за її легкість у використанні, невеликий розмір та вбудовану підтримку в Python. Вона дозволяє зберігати інформацію про користувачів, їх прогрес у вивченні нових слів, історію запитів та інші важливі дані, що покращує взаємодію користувачів з ботом. Завдяки своїй простоті та ефективності, SQLite не вимагає налаштування серверів, що робить її ідеальним рішенням для невеликих та середніх проектів, як чат-бот. Це дозволяє розробникам легко інтегрувати базу даних у застосунок, забезпечуючи швидкий доступ до даних та високу продуктивність. Використання SQLite сприяє надійній та безпечній роботі чат-бота, забезпечуючи збереження і доступність даних користувачів у будь-який час.

Після вибору основних компонентів та мови програмування настав час приступати до реалізації функціоналу чат-бота. З використанням Python та бібліотеки aiogram для інтеграції з Telegram API, а також бази даних SQLite для ефективного зберігання даних, створення ботів стає більш структурованим і зрозумілим процесом. На цьому етапі важливо визначити всі необхідні функції бота, такі як надання визначень слів, транскрипцій, прикладів використання, синонімів та аудіовимови.

Створення гнучкої архітектури та забезпечення надійної роботи бота допоможе забезпечити його довготривалу ефективність та корисність для користувачів. Крім того, варто розробити зручний і інтуїтивно зрозумілий інтерфейс взаємодії з користувачем. Планування і початок реалізації функціоналу дозволять забезпечити ефективну роботу бота та його відповідність поставленим завданням, роблячи процес вивчення нових слів для користувачів легким і цікавим.

3 РЕАЛІЗАЦІЯ ТА ДЕМОНСТРАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ВИВЧЕННЯ ІНОЗЕМНИХ МОВ

3.1 Формування графічного інтерфейсу

Формування графічного інтерфейсу для Telegram бота для вивчення іноземних мов має на меті забезпечити інтуїтивно зрозумілу та зручну взаємодію користувачів з ботом.

Хоча Telegram боти зазвичай використовують текстовий інтерфейс, важливо приділити увагу організації та подачі інформації, щоб користувачі могли легко додавати нові слова, отримувати визначення, приклади використання, синоніми та аудіофайли з вимовою.

Використання кнопок, меню та інших інтерактивних елементів у повідомленнях бота може значно покращити взаємодію з користувачем і зробити процес навчання більш структурованим і цікавим.

Створення чіткого, лаконічного та візуально привабливого інтерфейсу допоможе користувачам ефективніше взаємодіяти з вашим ботом і досягати своїх навчальних цілей.

Меню з вибором команд допомагає користувачам отримати швидкий доступ до всіх команд які доступні в чат-боті.

Це меню дозволяє користувачам швидко і легко використовувати такі команди, як /add для додавання нових слів із перекладами, /translate для пошуку перекладів, /definition для отримання визначень, /example для перегляду прикладів використання в реченні, /synonyms для пошуку синонімів, та /audio для прослуховування аудіофайлів з вимовою.

Інтерактивні кнопки та меню роблять взаємодію з ботом більш інтуїтивною та зрозумілою, допомагаючи користувачам ефективно вивчати мову.

					КВРІСТ 101107.01.07 ПЗ	Арк.
						40
Зм.	Арк.	№докум.	Підпис	Дата		

Зручне та організоване меню дозволяє максимально використовувати можливості бота, полегшуючи процес навчання та забезпечуючи швидкий доступ до всіх необхідних функцій (рис. 3.1).

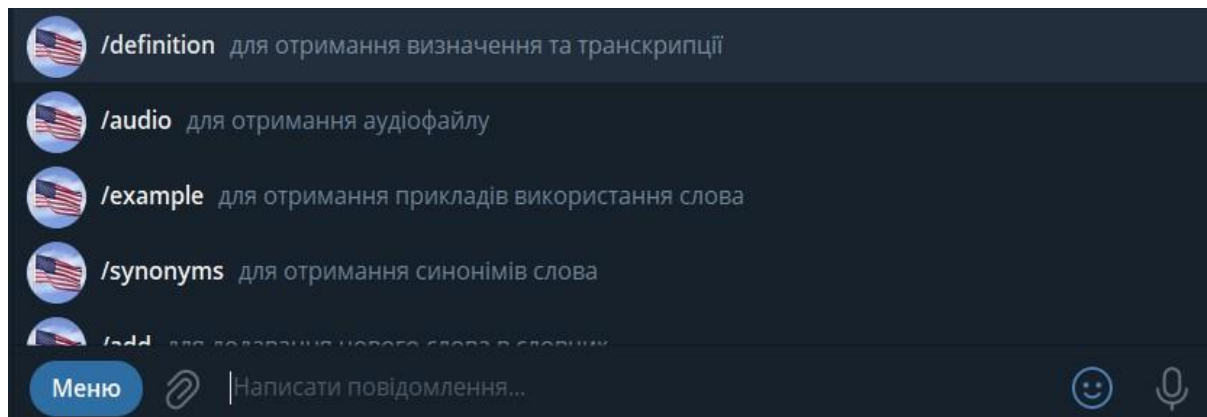


Рисунок 3.1 – Вигляд меню з доступними командами

Вікно з командами може бути викликане за допомогою кнопки меню або при введенні символу "/", що робить взаємодію з ботом ще більш інтуїтивною та зручною. Таке організоване меню допомагає користувачам максимально використовувати можливості бота, полегшуючи процес навчання та забезпечуючи швидкий доступ до всіх необхідних функцій (рис. 3.2).

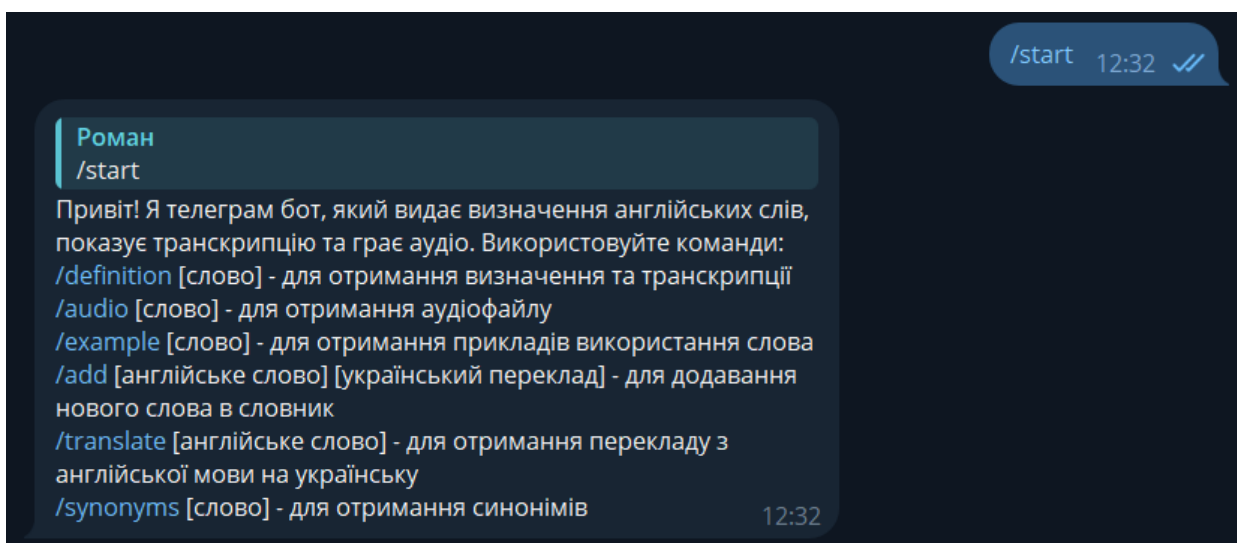


Рисунок 3.2 – Зміст команди /start

Команда /start є першою і найважливішою командою, яка вітає користувача та надає базову інформацію про те, як правильно взаємодіяти з ботом і подавати запити. Після виклику команди /start, бот надсилає повідомлення в якому розписано як правильно подавати запит для кожної команди в чат-боті. Команда /start автоматично надсилається після першого запуску бота, в подальшому цю команду можна все ще викликати в любий зручний час. Це повідомлення не лише вітає користувача, але й надає чіткі інструкції щодо використання основних команд бота. Завдяки команді /start, користувачі швидко дізнаються, як подавати правильні запити, що допомагає уникнути помилок та забезпечує ефективну взаємодію з ботом.

У висновку варто зазначити що формування графічного інтерфейсу для Telegram бота для вивчення іноземних мов було представлено опис головних елементів інтерфейсу, включаючи меню з переліком команд, команду /start Розглянуто структуру та функціонал меню з урахуванням доступних команд, які користувач може викликати через нього. Детально розглянуто роботу першої команди /start, яка вітає користувача та надає інструкції щодо правильного використання бота.

Підкреслено важливість зрозумілого та доступного інтерфейсу для користувачів, що дозволяє їм ефективно користуватися функціоналом бота та зручно отримувати необхідну інформацію. За допомогою правильно сформованого графічного інтерфейсу користувачі можуть швидко орієнтуватися та використовувати доступні функції бота для вивчення іноземних мов.

3.2 Взаємодія процесів в чат-боті

Взаємодія процесів в Telegram боті для вивчення іноземних мов відбувається через кілька ключових етапів. Користувач відправляє команду через Telegram, яка пересилається до бота через Telegram API. Бот отримує цю команду і обробляє її, виконуючи відповідні дії. Якщо команда стосується додавання

					КвРІСТ 101107.01.07 ПЗ	Арк. 42
Зм.	Арк.	№докум.	Підпис	Дата		

показує послідовність цих дій: від отримання команди від користувача, запиту до бази даних, отримання результатів і повернення їх користувачеві.

Така взаємодія забезпечує ефективне управління даними та швидкий доступ до необхідної інформації, підвищуючи зручність користування ботом (рис. 3.4).

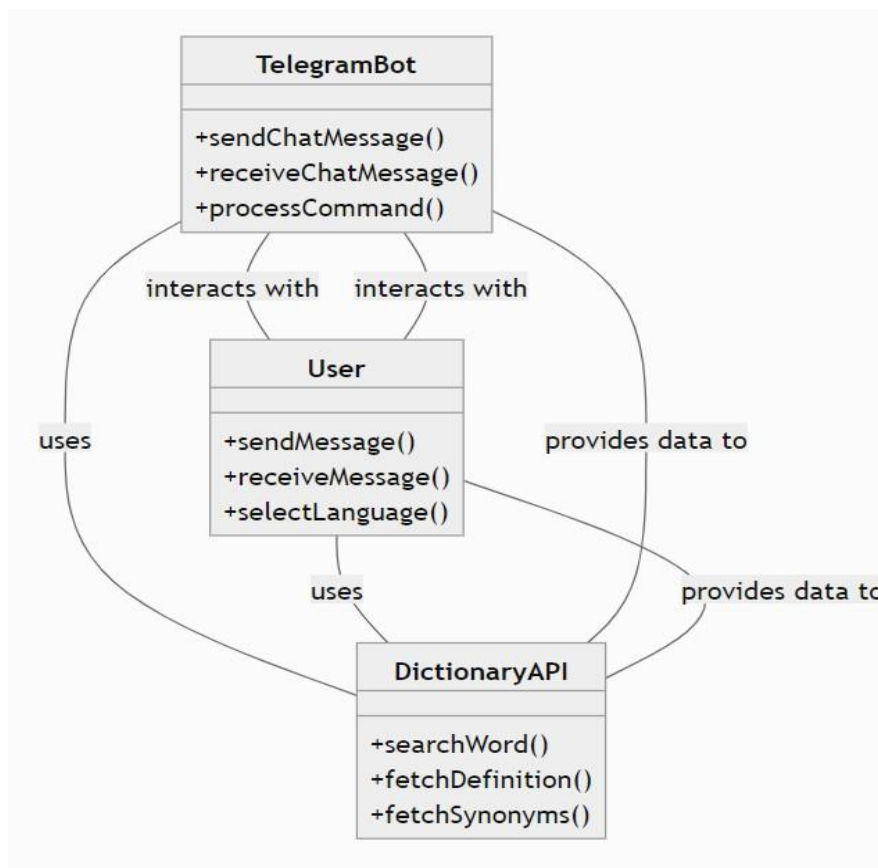


Рисунок 3.4 – Діаграма взаємодії бота з Dictionary API

Діаграма взаємодії бота з Dictionary API демонструє, як бот отримує лексичні дані для користувачів. Коли користувач відправляє запит, наприклад, на визначення слова або аудіо вимову, бот спочатку обробляє команду та формує відповідний запит до Dictionary API. API повертає JSON-відповідь, що містить потрібну інформацію, таку як визначення слова, транскрипція, приклади використання в реченні, синоніми та аудіо файли з вимовою. Бот аналізує отриману відповідь і відправляє користувачеві відповідне повідомлення з результатами. Така взаємодія дозволяє боту забезпечувати користувачів

миттєво реагувати на дії користувачів і знижує повідомлення. Це дозволяє миттєво реагувати на дії користувачів і знижує навантаження на сервер, але потребує наявності публічно доступного та захищеного серверу для обробки запитів.

Long Polling відіграє важливу роль в проєкті, забезпечуючи стабільне та надійне отримання оновлень від сервера Telegram. Цей метод дозволяє боту робити періодичні запити до сервера Telegram для перевірки нових повідомлень. Хоча це може призводити до невеликих затримок, Long Polling має низку переваг: він простий у налаштуванні, не потребує додаткових ресурсів і може працювати без наявності публічно доступного сервера, що робить його ідеальним вибором для проєкту. Використання Long Polling гарантує, що бот завжди зможе відповідати на запити користувачів, зберігаючи при цьому високу стабільність і надійність роботи.

Підводячи підсумки варто зазначити що, взаємодія процесів у чат-боті для вивчення іноземних мов є складним і добре організованим процесом, який забезпечує ефективне виконання команд користувачів. Бот обробляє введені команди, взаємодіє з базою даних для зберігання і отримання інформації, а також використовує зовнішні API для надання додаткових функцій, таких як, наприклад, визначення слів та пошук синонімів. Коли користувач вводить команду, бот аналізує її та визначає необхідну дію. Для команд, пов'язаних із базою даних, бот звертається до бази даних SQLite для зберігання нових слів або отримання існуючих перекладів.

Для команд, що потребують зовнішніх ресурсів, таких як аудіо вимова чи визначення слів, бот звертається до Dictionary API, отримуючи необхідні дані та відправляючи їх користувачу. Ця взаємодія між компонентами забезпечує комплексний і зручний функціонал чат-бота, який може ефективно допомагати користувачам у вивченні англійської мови. Кожен компонент, від бази даних до зовнішніх API, відіграє важливу роль у створенні інтерактивного та корисного інструменту для навчання.

					КвРІСТ 101107.01.07 ПЗ	Арк. 46
Зм.	Арк.	№докум.	Підпис	Дата		

3.3 Демонстрація роботи чат-бота

Під час демонстрації роботи чат-бота буде продемонстрована поведінка чат-бота, призначеного для вивчення англійської мови.

Мета - показати ефективність та корисність цього бота в процесі навчання іноземних мов. Під час запуску чат-бота для вивчення іноземних мов, перший крок полягає в активації бота в Telegram-платформі. Це досягається шляхом створення спеціального бота через Telegram BotFather і отримання унікального токена доступу, який ідентифікує бота в системі. Після цього бот може бути запуснений на сервері, використовуючи мову програмування Python та бібліотеку Aiogram для взаємодії з API Telegram. Після успішного запуску бот буде готовий приймати запити від користувачів та надавати їм необхідну інформацію та сервіси для вивчення іноземних мов.

BotFather – це офіційний інструмент для створення та керування ботами в Telegram. Він дозволяє розробникам легко створювати нових ботів, налаштовувати їхні параметри, такі як ім'я, опис та аватар, а також отримувати токени доступу для інтеграції бота з API Telegram. За допомогою BotFather можна також керувати командами бота, додавати нові функції та налаштовувати інтерактивні можливості. Цей інструмент значно спрощує процес створення ботів та їх подальше адміністрування, надаючи розробникам усі необхідні ресурси для швидкого запуску і підтримки їхніх проектів.

Процес створення бота за допомогою BotFather у Telegram є простим і зручним. Спочатку потрібно знайти BotFather у пошуку Telegram і почати з ним чат, використовуючи команду /start. Далі, слід використати команду /newbot, щоб створити нового бота. BotFather попросить користувача вказати ім'я бота та унікальне ім'я користувача (username), яке повинно закінчуватися на 'bot'. Після цього користувач отримає токен доступу, який необхідно зберегти, оскільки він використовується для інтеграції чат-бота з API Telegram. Щоб використовувати

					КвРІСТ 101107.01.07 ПЗ	Арк.
						47
Зм.	Арк.	№докум.	Підпис	Дата		

цей токен, треба налаштувати свого бота, додавати команди, інтегрувати його з різними сервісами та розширювати його функціональні можливості (рис. 3.6).

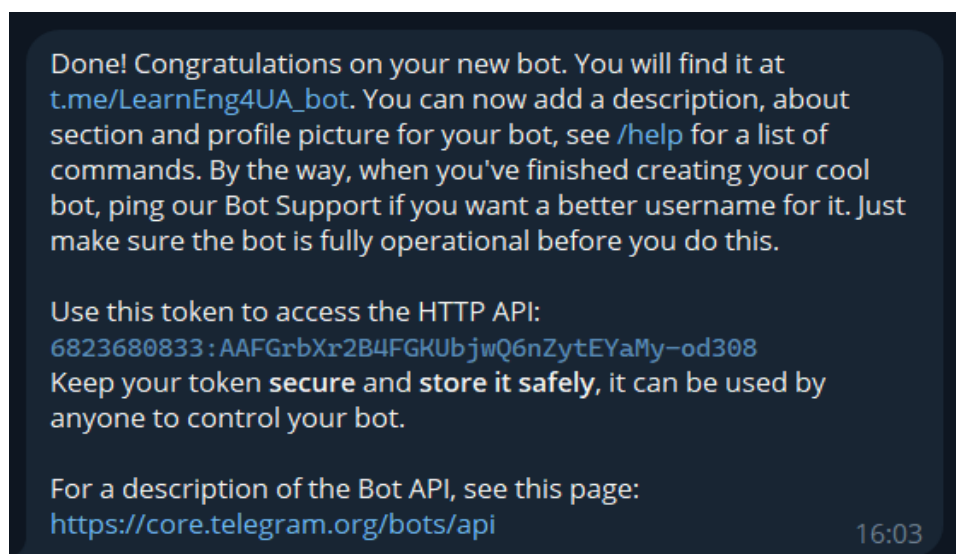


Рисунок 3.6 – Повідомлення про успішне створення чат-бота

Після створення бота за допомогою BotFather та отримання токена доступу, наступним кроком було встановлення зображення профілю бота, додання опису та короткої інформації. За допомогою команд `/setuserpic`, `/setdescription` та `/setabouttext` у BotFather були завантажені відповідні зображення та текстові дані. Зображення профілю допомагає користувачам легко ідентифікувати бота серед інших контактів. Опис бота містить детальну інформацію про його функції та можливості, що допомагає користувачам зрозуміти, як саме вони можуть взаємодіяти з ботом. Коротка інформація, додана за допомогою команди `/setabouttext`, відображається при відкритті чату з ботом і дає користувачам швидкий огляд основних можливостей та цілей бота.

Наступним кроком після налаштування зображення, опису та короткої інформації було додавання команд до бота через BotFather. Команди допомагають користувачам зрозуміти, як взаємодіяти з ботом і які функції він підтримує. Для додавання команд було використано команду `/setcommands` у BotFather. Ця

					КвРІСТ 101107.01.07 ПЗ	Арк.
						48
Зм.	Арк.	№докум.	Підпис	Дата		

команда дозволяє визначити список команд разом з їх короткими описами, що будуть відображатися у вбудованому меню бота. Наприклад, команди можуть включати '/define - отримати визначення слова', '/translate - перекласти слово або фразу', '/synonyms - знайти синоніми для слова' тощо. Це допомагає зробити взаємодію з ботом інтуїтивно зрозумілою та зручною для користувачів (рис. 3.7).



Рисунок 3.7 – Вигляд Чат-Бота

Початкова сторінка бота з інформацією в Telegram надає користувачам швидкий і зручний доступ до основної інформації про бота. Тут відображається фото профілю бота, яке може бути логотипом або іншим зображенням, що асоціюється з ботом. «Юзернейм» бота, починаючи зі знака "@", допомагає користувачам легко знайти бота у пошуку. Опис бота містить коротку інформацію про його функціональність та цілі. Картинка на титульній сторінці, якщо додана, доповнює візуальний образ бота, роблячи його більш привабливим і впізнаваним. Разом ці елементи створюють перше враження про бота та стимулюють користувачів розпочати взаємодію з ним (рис. 3.8).

					КвРІСТ 101107.01.07 ПЗ	Арк. 49
Зм.	Арк.	Недокум.	Підпис	Дата		

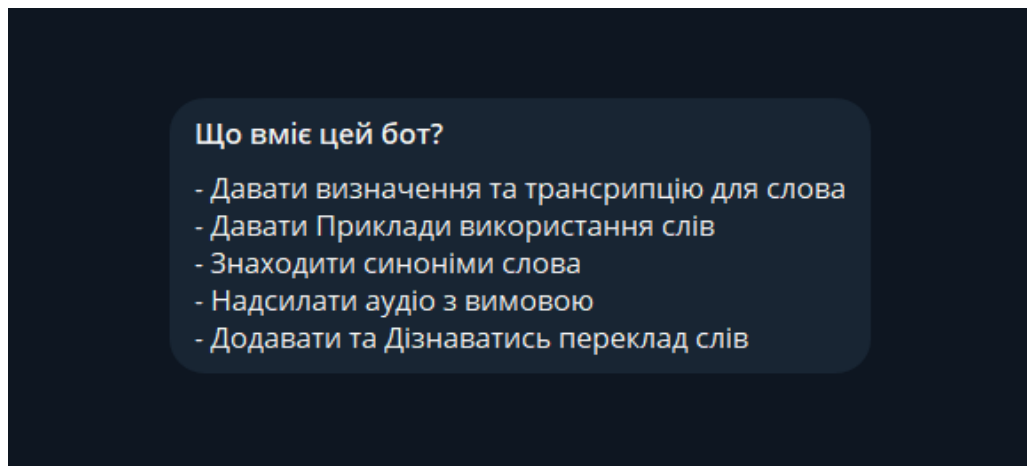


Рисунок 3.8 – Коротка інформація про функціонал чат-бота

При першому відкритті чату з чат-ботом у Telegram користувач бачить вітальне повідомлення, яке містить короткий опис функцій бота. Це повідомлення призначене для ознайомлення користувача з можливостями та командами бота. В ньому зазначено, що бот може надавати визначення слів, відтворювати вимову, показувати приклади використання слів, знаходити синоніми, зберігати дані в базу даних та здійснювати переклади. Це допомагає користувачу швидко зрозуміти, як саме можна використовувати бот і які команди доступні для взаємодії з ним (рис. 3.9).

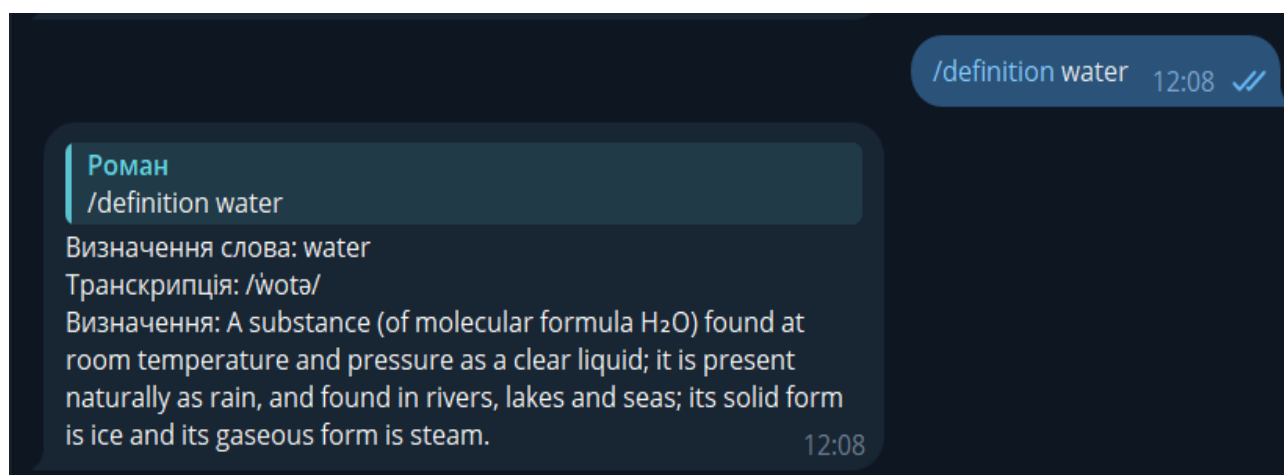


Рисунок 3.9 – Функціонал команди /definition

Визначення слів. Команда надає користувачам можливість одержувати визначення та транскрипцію англійських слів. Після виклику команди /definition, користувач може відправити слово, для якого він бажає отримати визначення. Бот обробляє це слово та намагається знайти відповідне визначення та транскрипцію. Якщо слово знайдено, бот надсилає користувачеві визначення та транскрипцію. Ця функція корисна для тих, хто вивчає англійську мову та потребує швидкого доступу до перекладу та визначення слів (рис. 3.10).

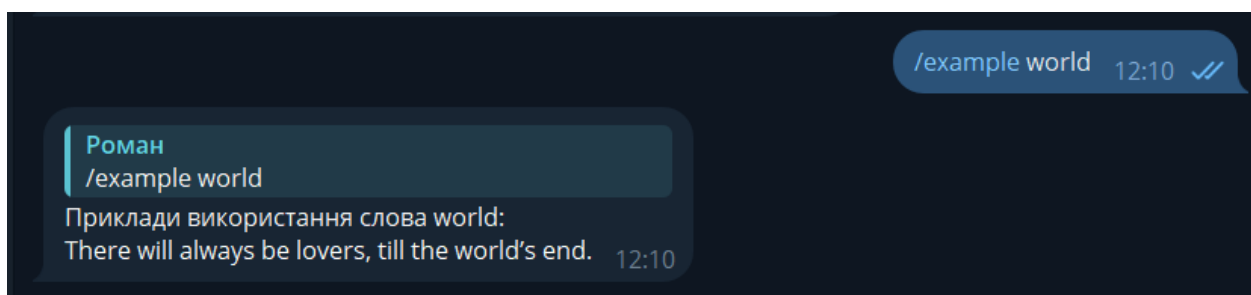


Рисунок 3.10 – Функціонал команди /example

Приклад використання слова. Команда /example надає користувачам можливість отримати приклад використання заданого слова в англійській мові. Після активації цієї команди користувач може відправити слово, для якого він бажає отримати приклад вживання.

Бот обробляє запит та намагається знайти контекст, в якому це слово може бути використане.

Після успішного пошуку бот надсилає користувачеві приклад використання слова в реальних англомовних реченнях. Ця функція допомагає користувачам краще зрозуміти контекст та ситуації, в яких можуть вживатися англійські слова, що сприяє покращенню їхнього розуміння мови.

Прослухати вимову слова - Команда /audio в Telegram боті забезпечує користувачам можливість отримати вимову конкретного англійського слова (рис. 3.11).

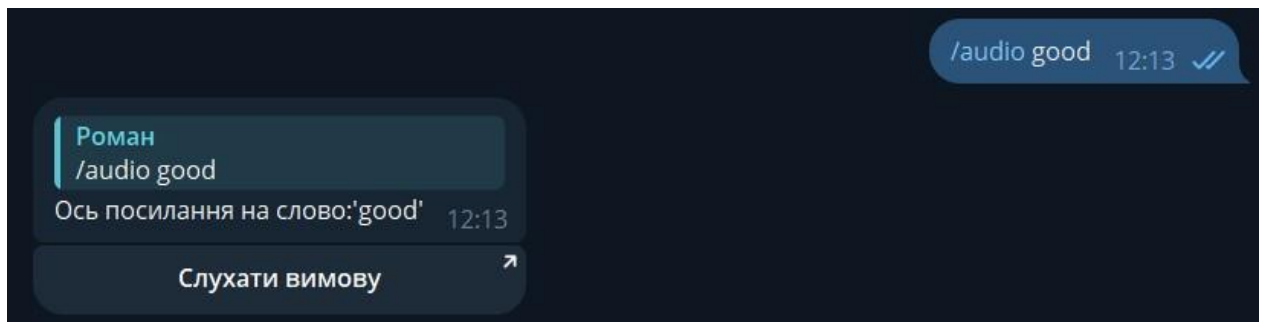


Рисунок 3.11 – Функціонал команди /audio

Користувач вводить команду /audio разом зі словом, вимову якого він хоче почути. Бот обробляє запит, звертається до Dictionary API, отримує посилання на аудіофайл із правильною вимовою та відправляє це посилання користувачу. Повідомлення бота містить текст "Ось посилання на слово 'слово яке задав користувач':, а також інтерактивну кнопку "Слухати вимову", яка дозволяє користувачу негайно перейти до прослуховування аудіофайлу. Така функціональність допомагає користувачам удосконалювати свою вимову, прослуховуючи правильні зразки вимови слів (рис. 3.12).

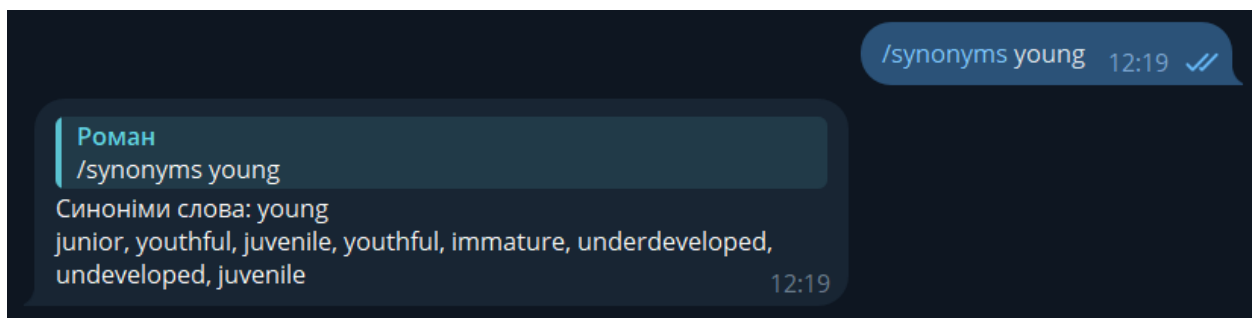


Рисунок 3.12 – Функціонал команди /synonyms

Дізнатись синоніми слова – Команда /synonyms надає користувачам можливість отримати синоніми для англійського слова, яке вони вказали. Користувачі вводять англійське слово, за яким вони бажають отримати синоніми, і бот надає їм список інших слів з аналогічним або схожим значенням. Ця функція допомагає розширити словниковий запас користувачів та збагатити їхнє

розуміння англійської мови, дозволяючи їм використовувати різноманітні слова для вираження одного й того ж поняття або ідеї. Такий підхід сприяє розвитку мовленнєвих навичок та покращенню загального рівня знань англійської мови у користувачі (рис. 3.13)в.

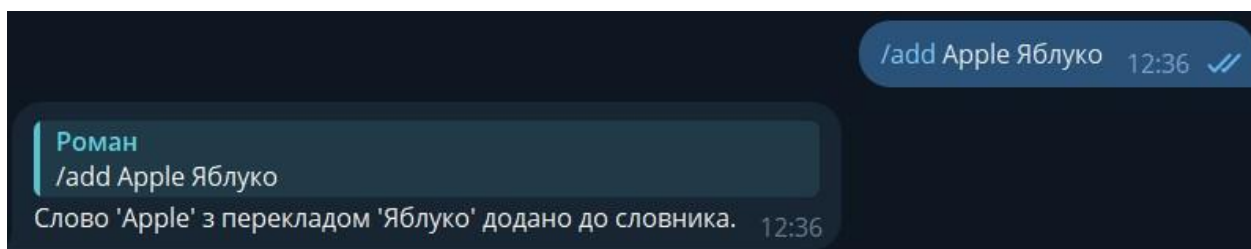


Рисунок 3.13 – Функціонал команди `/add`

Додавання нових слів у базу даних – Команда `/add` є важливим функціоналом чат-бота, оскільки вона дозволяє користувачам активно брати участь у розширенні бази даних бота.

Користувачі можуть ввести англійське слово разом із відповідним українським перекладом, і бот автоматично додає цю пару слів до бази даних.

Ця можливість не лише дозволяє користувачам активно взаємодіяти з ботом, але й допомагає у поповненні словника та розширенні його функціоналу, щоб надавати більш повну та різноманітну інформацію користувачам (рисунок 3.14).

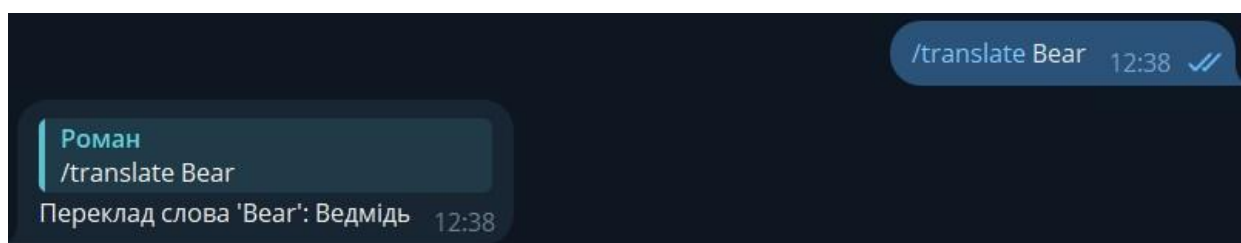


Рисунок 3.14 – Функціонал команди `/translate`

Переклад слів. Команда /translate дозволяє користувачам отримувати переклад англійських слів на українську мову, використовуючи інформацію, яка міститься в базі даних бота. Користувачі вводять англійське слово, за яким вони бажають отримати переклад, і бот надає їм відповідний український еквівалент, який зберігається в базі даних. Ця функція дозволяє користувачам швидко та зручно перекладати слова, не виходячи за межі чат-бота, і використовувати отриманий переклад для розуміння і використання англійських слів у контексті української мови.

Можна підсумувати що було детально описано процес створення та налаштування Telegram бота за допомогою BotFather, а також продемонстровано функціонал його команд. Першим кроком було створення бота через BotFather, де був отриманий унікальний токен доступу, необхідний для інтеграції з API Telegram.

Далі були виконані налаштування зображення профілю, опису та короткої інформації про бота, щоб користувачі могли легко ідентифікувати бота та розуміти його основні функції. Особлива увага була приділена додаванню команд через BotFather, що дозволило зробити бота більш інтуїтивно зрозумілим і зручним у використанні. Окрім опису процесу налаштування, була продемонстрована робота кожної з команд, що показало ефективність і корисність бота в реальних умовах. Таким чином, було продемонстровано не лише технічні аспекти створення та налаштування бота, але було показано практичну цінність та функціональність для кінцевих користувачів.

Функціонування команд бота було ретельно продемонстровано для показу їх практичної користі та ефективності.

Команда /definition дозволяє користувачам отримувати точні визначення слів, надаючи їм чітке розуміння їх значення.

Команда /translate забезпечує можливість перекладу слів або фраз на різні мови, що є надзвичайно корисним для користувачів, які вивчають нові мови або потребують швидкого перекладу.

					КвРІСТ 101107.01.07 ПЗ	Арк.
						54
Зм.	Арк.	№докум.	Підпис	Дата		

Команда `/synonyms` дозволяє знаходити синоніми до введених слів, що допомагає розширити словниковий запас та знайти більш відповідні слова для різних контекстів.

Команда `/audio` надає можливість прослухати вимову слова, що допомагає покращити вимовні навички. Крім того, команда `/example` показує приклади використання слів у реченнях, що допомагає користувачам краще зрозуміти, як слово використовується на практиці. Всі ці команди інтегруються з базою даних, що дозволяє зберігати важливі дані та забезпечує зручний доступ до необхідної інформації. Завдяки цьому, користувачі можуть швидко і ефективно взаємодіяти з ботом, отримуючи всю необхідну мовну підтримку.

PyCharm відіграє важливу роль у розробці та демонстрації чат-бота, забезпечуючи зручне і потужне середовище для кодування, тестування та відлагодження. Завдяки своїм інструментам для автозавершення коду, інтегрованому відлагоджувачу та підтримці віртуальних оточень, PyCharm значно спрощує процес написання та оптимізації коду. Під час демонстрації чат-бота PyCharm дозволяє швидко знаходити і виправляти помилки, а також тестувати різні функції бота в реальному часі. Його зручний інтерфейс та багатофункціональність роблять процес розробки інтуїтивно зрозумілим і ефективним, що є ключовим для створення якісного програмного забезпечення. Крім того, PyCharm підтримує інтеграцію з системами контролю версій, що дозволяє відстежувати зміни в коді та спільно працювати над проектом. Усе це робить PyCharm незамінним інструментом для успішної розробки та демонстрації функціональних можливостей чат-бота.

Одним з найважливіших та цікавих аспектів у розробці та розширенні функціональності чат-бота було створення команди для відтворення аудіофайлів. Ця функція надає користувачам можливість прослуховувати вимову англійських слів за допомогою спеціально створеного лінка, що значно покращує їхнє засвоєння та розуміння правильної вимови. Впровадження цієї функції вимагало інтеграції зі словниковим API для отримання відповідних аудіофайлів та розробки

					КвРІСТ 101107.01.07 ПЗ	Арк.
						55
Зм.	Арк.	№докум.	Підпис	Дата		

механізму генерації посилань, які надсилаються користувачеві для прослуховування. Основним акцентом вимови був обраний британський акцент, Тому що саме британський акцент є одним із самих розповсюджених акцентів для вивчення англійської мови. Це рішення не лише розширює можливості чат-бота, але й зробило процес вивчення англійської мови більш інтерактивним і ефективним для кінцевих користувачів.

Користувачі мають можливість скористатися командою /audio, щоб отримати лінк з вимовою конкретного слова. Ця функція чат-бота дозволяє користувачам не лише прочитати визначення слова, а й почути його коректну вимову. Після введення команди бот формує спеціальне посилання, яке забезпечує доступ до аудіофайлу з правильною вимовою слова, що значно полегшує процес вивчення та запам'ятовування нових слів (рис. 3.15).

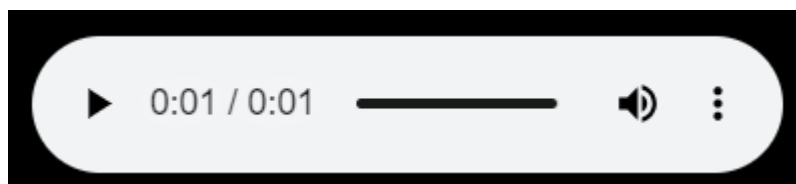


Рисунок 3.15 – Вигляд плеєра для прослуховування аудіофайла

Лінк представляє собою URL-адресу, що надає можливість отримати аудіофайл з правильною вимовою англійського слова в британському варіанті. Використання цього URL-адреси дозволяє чат-боту надсилати користувачам зручний доступ до аудіофайлів, які містять вимову конкретних слів. Посилання побудоване таким чином, що параметр «word» може бути замінений на будь-яке слово англійської мови, що дозволяє динамічно генерувати посилання для будь-якого запиту на вимову.

Цей механізм дозволяє швидко і зручно отримувати аудіофайли формату MP3 з вимовою, що можна відтворити на будь-якому пристрої або програмі. Користувачі можуть без зусиль слухати правильну англійську вимову слів, що допомагає поліпшити їх навички аудіювання та відтворення мовленнєвих звуків.

Використання цього посилання забезпечує ефективний інструмент для навчання та вдосконалення вимови, що робить процес вивчення англійської мови більш доступним і цікавим для користувачів чат-бота. Розвиток чат-бота обіцяє бути значним завдяки його потужній базі даних і вбудованому API.

Наявність розширеної бази даних слів і фраз дозволяє боту не лише надавати визначення та синоніми для запитаних слів, але й забезпечує доступ до аудіофайлів з вимовою кожного слова.

Це створює унікальну можливість для користувачів не лише читати, а й слухати правильну вимову, що покращує їхнє засвоєння англійської мови.

Інтеграція з цим API дозволяє боту стати динамічним інструментом для навчання, що не лише допомагає розширювати словниковий запас, але й забезпечує більш глибоке розуміння й використання мовних концепцій у реальному часі.

Такий підхід робить процес вивчення англійської мови більш ефективним та інтерактивним для кожного користувача. Шляхом постійного розширення бази даних слів і фраз бот може стати ще більш універсальним і корисним інструментом для користувачів з різними потребами і рівнями знань.

3.4 Висновок

Підсумовуючи висновок про реалізацію інформаційної системи для вивчення іноземних мов за допомогою чат-бота треба виділити основні аспекти реалізації та демонстрації функціоналу чат-бота.

Графічний інтерфейс бота був створений з урахуванням зручності та інтуїтивного сприйняття користувачами. Він включає в себе чітко відображені елементи управління, такі як кнопки та текстові поля, що спрощує навігацію та взаємодію з ботом.

Графічний інтерфейс чат-бота реалізований з використанням стандартних функцій і можливостей месенджера Telegram, що забезпечує його сучасний

					КвРІСТ 101107.01.07 ПЗ	Арк.
						57
Зм.	Арк.	№докум.	Підпис	Дата		

широкого спектру послуг, включаючи визначення слів з транскрипціями, надання прикладів використання у реченнях, синонімів та можливість прослуховувати аудіофайли з правильною вимовою.

Під час демонстрації користувачі мали можливість переконатися у відмінності та корисності цих функцій, що підтверджує важливість і практичну цінність чат-бота в контексті вивчення іноземних мов.

					КВРІСТ 101107.01.07 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		59

ВИСНОВОК

У завершенні роботи над розробкою чат-бота можна підвести загальний висновок по всій роботі над створенням інформаційної системи для вивчення іноземних мов. Першою задачею стояло проведення аналізу предметної області, де були виявлені ключові проблеми та завдання, пов'язані з вивченням англійської мови. Для цього було проведено порівняльний аналіз існуючих рішень та платформ, визначивши їх переваги та недоліки. Особлива увага була приділена зручності використання, швидкості та доступності навчальних ресурсів, а також їхній ефективності у навчанні та підвищенні мовної компетентності. Результати аналізу послужили основою для розробки чат-бота, спрямованого на вдосконалення процесу вивчення англійської мови з використанням технологій програмування. Вибір мови програмування Python для реалізації проекту Telegram бота для вивчення англійської мови виявилось вдалим та правильним рішенням. Після успішного завершення проекту стало очевидним, що Python дозволив реалізувати всі поставлені завдання з максимальною ефективністю та швидкістю. Його проста синтаксична структура та багата екосистема бібліотек дозволили швидко розробляти та впроваджувати новий функціонал, забезпечуючи при цьому високу якість та стабільність програмного продукту. Python також виявився досить простим для розуміння та навчання, що зробило його ідеальним вибором для розробки проекту, спрямованого на навчання та розвиток. В цілому, вибір Python підтвердив свою правильність та допоміг досягти успіху у реалізації проекту та досягненні його цілей.

При розробці чат-бота були використані різні бібліотеки, що значно спростило та прискорило процес розробки. Зокрема, бібліотека aiogram виявляється необхідною для роботи з Telegram Bot API, надаючи зручний інтерфейс для взаємодії з ботом і обробки повідомлень користувачів. Крім того, за допомогою бібліотеки запитів тепер можна надсилати HTTP-запити до зовнішніх служб, таких як Dictionary API, для отримання лінгвістичної інформації про слова.

					КвРІСТ 101107.01.07 ПЗ	Арк. 60
Зм.	Арк.	№докум.	Підпис	Дата		

Використовуючи ці бібліотеки, ми значно спростили та оптимізували розробку та роботу нашого Telegram-бота, що дозволило нам зосередитися на вирішенні основних завдань нашого проекту та покращенні взаємодії з користувачем.

Dictionary API виявився надзвичайно корисним інструментом для функціонування чат-бота. Використання цього API дозволило чат-боту отримувати різноманітну лінгвістичну інформацію про слова, включаючи визначення, приклади використання в реченнях, синоніми та навіть аудіофайли з вимовою. Це зробило бота повноцінним інструментом для вивчення мови.

SQLite знадобився чат-боту для зберігання інформації, що забезпечує зручний і ефективний спосіб управління даними. SQLite дозволяє легко зберігати, оновлювати та отримувати дані без необхідності налаштування складних серверних систем.

Результат розробки Telegram бота для вивчення англійської мови показав, що обрані технології та методи були правильними та ефективними. Бот успішно інтегрувався з Dictionary API для отримання визначень, транскрипцій, прикладів використання слів, синонімів та аудіофайлів з правильною вимовою. Використання мови програмування Python та бібліотек, таких як AIogram та Requests, дозволило створити гнучку та надійну систему. Бот має простий та зрозумілий інтерфейс, що забезпечує зручну взаємодію з користувачем через команди меню. Кожен користувач може легко додавати нові слова до бази даних, отримувати переклади та аудіовимови, що значно покращує процес вивчення англійської мови. Загалом, проект досяг своїх цілей, створивши корисний та функціональний інструмент для мовного навчання, який може бути надалі розширений та вдосконалений.

					КвРІСТ 101107.01.07 ПЗ	Арк.
						61
Зм.	Арк.	№докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Malinova A., Ivanova V. Automation of Electronic Testing Examination in English Language using DeTC. *Scientific Works*. 2012. Vol. 38. Book 3. pp. 59–68.
2. Akhyak and A. Indramawan. Improving the Students' English Speaking Competence through Storytelling: Study in Pangeran Diponegoro Islamic College (STAI) of Nganjuk, East Java, and Indonesia. *International Journal of Language and Literature*. 2013. Vol. 1. No. 2. pp. 18–24.
3. Level of proficiency in english and other foreign languages in ukraine: results of quantitative sociological research conducted in december 2022 - january 2023 – 2023. URL:<https://www.kiis.com.ua/?Lang=eng&cat=reports&id=1210&page=1>.
4. English with Lucy. URL: <https://englishwithlucy.com/> (дата звернення: 28.03.2024).
5. Matt D'avella. URL:<https://www.mattdavella.com> / (дата звернення: 28.03.2024).
6. Goel S. How does Duolingo work & make money: Business Model & Strategy. URL:<https://thestrategystory.com/2022/11/12/how-does-duolingo-work-make-money-business-model-strategy/> (дата звернення: 28.03.2024).
7. Duolingo. URL: <https://www.duolingo.com/> (дата звернення: 23.10.2023).
8. Петрик М.Р., Петрик О.Ю. Моделювання програмного забезпечення: науково-методичний посібник. Тернопіль: Вид-во ТНТУ імені Івана Пулюя, 2015. 200 с.
9. Kryskova, S. Хмарні сервіси для вивчення іноземної мови. *English in Interdisciplinary Context of Life-long Education*. 2019. P. 292.
10. Серов Ю., Соломон А. Специфіка використання YouTube-каналів як бази знань для ефективного вивчення іноземної мови. *Вісник Книжкової палати*. 2018. (6), pp. 46–48.
11. Творошенко І.С. Технології прийняття рішень в інформаційних

					КВРІСТ 101107.01.07 ПЗ	Арк. 62
Зм.	Арк.	№докум.	Підпис	Дата		

системах: навч. посібник. Харків: ХНУРЕ, 2021. 120 с.

12. Демчук С.В., Гаврилюк Р.О., Самусько Б.М., Кошельник Д.С. Розробка застосунку для вивчення іноземних слів "English for you" (Doctoral dissertation, ВНТУ). 2016.

13. Álvarez Valencia J.A. Language views on social networking sites for language learning: the case of Busuu. *Computer Assisted Language Learning*. 2016. Vol. 29. No. 5. pp. 853–867.

14. Pham X.L., Nguyen T.H., Chen G.D. Research through the app store: Understanding participant behavior on a mobile English learning app. *Journal of Educational Computing Research*. 2018. Vol. 57. No. 7. pp. 1076–1098.

15. Godwin-Jones R. Future Directions in Informal Language Learning. *The Handbook of Informal Language Learning*. 2019. pp. 457–470.

16. Жеманова О.А. Використання нетрадиційних методів у вивченні іноземних мов. *Актуальні проблеми навчання іноземних студентів*. 2019. Р. 40.

17. Кирич Б.М. Дослідження методів та засобів реалізації web-сервісів для вивчення іноземних мов (Doctoral dissertation, Тернопільський національний технічний університет імені Івана Пулюя). 2017.

18. Андрєєва А.Ю. Особливості та складнощі перекладу науково-технічних текстів. *Радіоелектроніка та молодь у XXI столітті: тези доповідей 22-го Міжнародного молодіжного форуму* (Харків, 16–18 квітня 2018 р.). Харків: ХНУРЕ, 2018. Т. 7. pp. 5–6.

19. Іванова М.С. Edutainment як спосіб поліпшення вивчення іноземної мови. *Національний авіаційний університет "Політ. Сучасні проблеми науки"*. 2021. Р. 42.

20. 8 найкращих додатків для вивчення англійської мови. URL: <https://www.fluentu.com/blog/english/>.

21. Гороль П.К. Мультимедійні засоби навчання: навч. посібник. Вінниця: Планер, 2010. 486 с.

22. Бех П.О., Биркун Л.В. Концепція викладання іноземних мов в Україні.

					КвРІСТ 101107.01.07 ПЗ	Арк.
						63
Зм.	Арк.	№докум.	Підпис	Дата		

28.03.2024).

45. GINO Basics. URL: <https://pythongino.org/docs/en/1.1b2/tutorials/tutorial.html> (дата звернення: 28.03.2024).

46. Using Chatbots in E-Learning. URL: <https://chatbotlife.com/using-chatbots-in-e-learning-dacde1d245e5> (дата звернення: 28.03.2024).

47. Python Requests. URL: <https://2.python-requests.org/en/master/> (дата звернення: 28.03.2024)..

48. Most Popular Global Mobile Messenger Apps. URL: <https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps/> (дата звернення: 28.03.2024).

49. A History of Instant Messaging and Chat. URL: <https://www.maize.io/news/lizshemaria-history-of-instant-messaging/> (дата звернення: 28.03.2024).

50. Pysox: Leveraging the Audio Signal Processing Power of Sox in Python. URL: <https://s18798.pcdn.co/ismir2016/wp-content/uploads/sites/2294/2016/08/bittner-pysox.pdf> (дата звернення: 28.03.2024).

51. Xintian Han. A Mathematical Introduction to Reinforcement Learning. URL: <https://cims.nyu.edu/~donev/Teaching/WrittenOral/Projects/XintianHan-WrittenAndOral.pdf>. (дата звернення: 28.03.2024).

52. What is SQLite? URL: <https://www.sqlite.org/index.html> (дата звернення: 28.03.2024).

53. How to Use Chatbots for Education and Learning. URL: <https://www.chatcompose.com/chatbot-learning.html> (дата звернення: 28.03.2024).

54. Functional Modelling: IDEF0. URL: <https://www.threesl.com/pages/reference/diagrams/idef0-diagram.php> (дата звернення: 28.03.2024).

55. Bots: An Introduction for Developers. URL: <https://core.telegram.org/bots> (дата звернення: 28.03.2024).

56. Responsibility Assignment Matrix (RAM) Goals.

										КвРІСТ 101107.01.07 ПЗ	Арк. 66
Зм.	Арк.	№докум.	Підпис	Дата							

[URL:https://acqnotes.com/acqnote/careerfields/responsibility-assignment-matrix](https://acqnotes.com/acqnote/careerfields/responsibility-assignment-matrix) (дата звернення: 28.03.2024).

57. Computer Networking and Internet Protocols. URL: <https://www.cse.wustl.edu/~jain/tutorials/ftp/airtel.pdf> (дата звернення: 28.03.2024).

58. Exploring Chatbots for News. URL: <https://medium.com/nos-digital/exploring-chatbots-for-news-7f12d226c53d> (дата звернення: 28.03.2024).

59. Gasic M., Breslin C., Henderson M., Kim D., Szummer M., Thomson B., Tsiakoulis P., Young S.J. On-line policy optimisation of Bayesian spoken dialogue systems via human interaction. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2013. pp. 8367–8371.

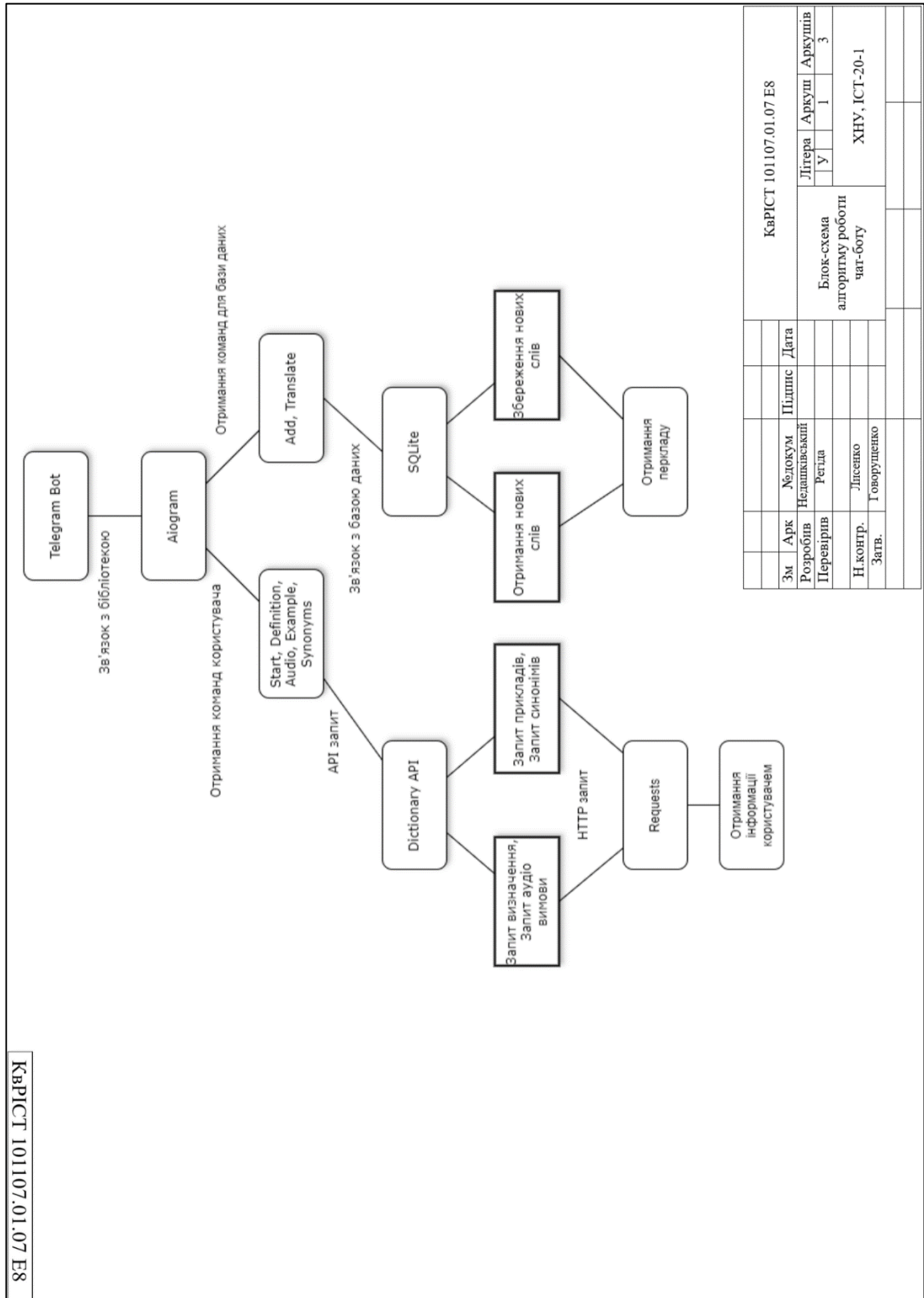
60. Chatbots for Learning: A Review of Educational Chatbots for the Facebook Messenger. URL: <https://www.sciencedirect.com/science/article/pii/S0360131520300622?pes=vor> (дата звернення: 28.03.2024).

					КВРІСТ 101107.01.07 ПЗ	Арк. 67
Зм.	Арк.	№докум.	Підпис	Дата		

ДОДАТОК А

(обов'язковий)

Копія креслення «Блок-схема алгоритму роботи чат-боту»

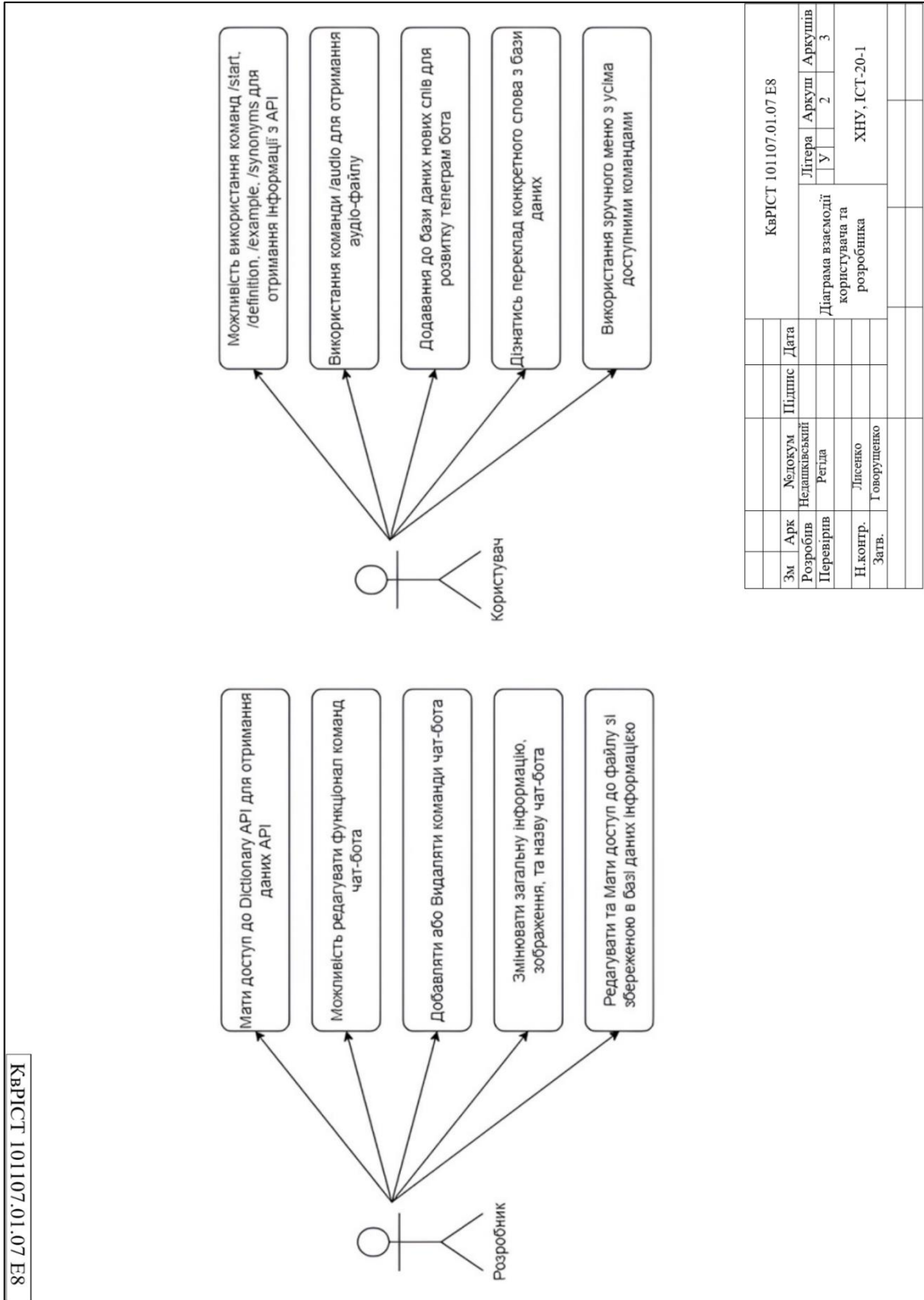


КВРІСТ 101107.01.07 Е8

КВРІСТ 101107.01.07 Е8									
Зм	Арк	Модокум	Підпис	Дата					
Розробив	Недальський	Регда			Літера	Аркуш	Аркушів		
Перевірив					У	1	3		
Н. контр.	Лісенко				Блок-схема алгоритму роботи чат-боту				
Загв.	Говоруленко				ХНУ, ІСТ-20-1				

Додаток Б (обов'язковий)

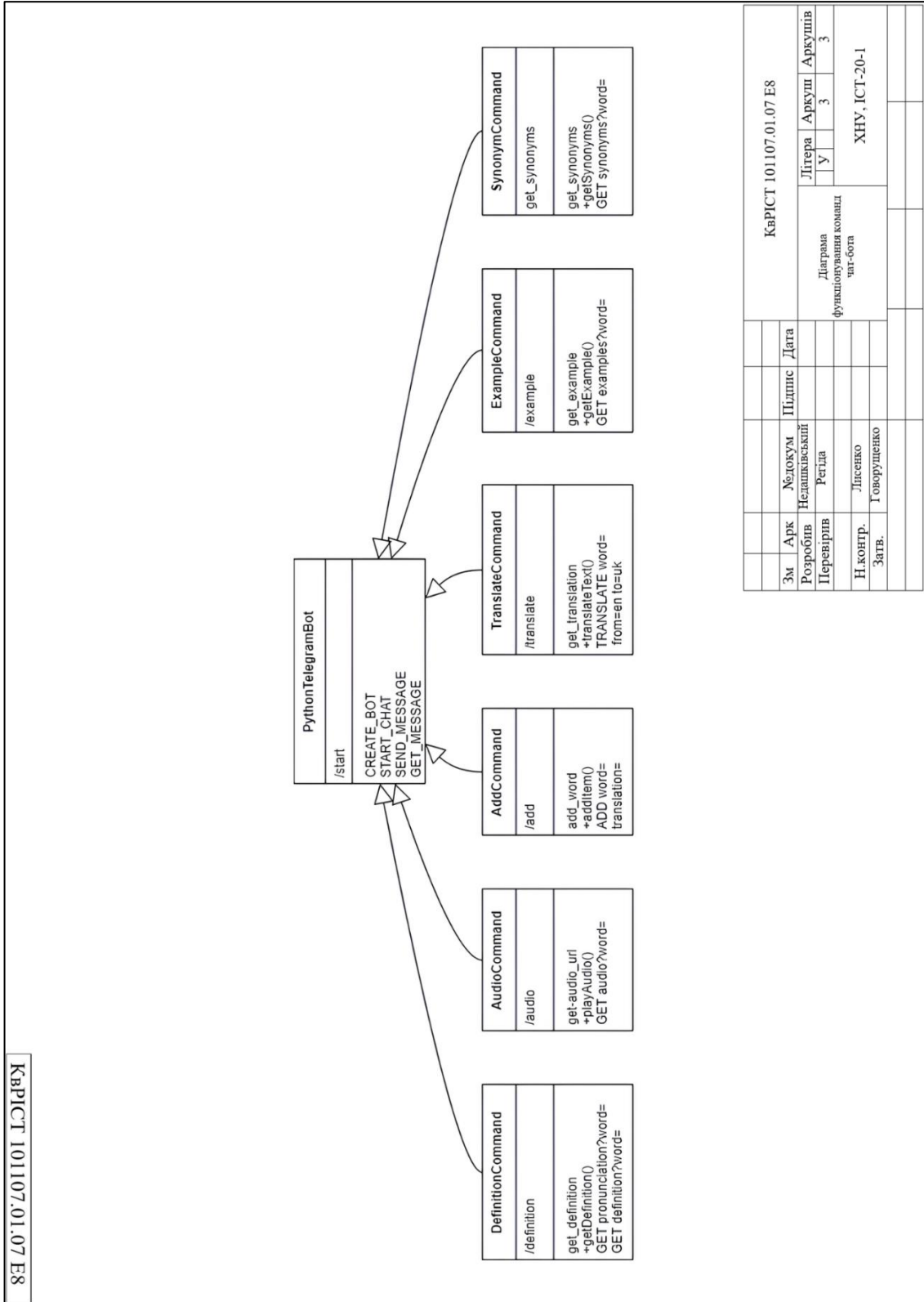
Копія креслення «Діаграма взаємодії користувача та розробника»



Додаток В

(обов'язковий)

Копія креслення «Діаграма функціонування команд чат-бота»



КвРІСТ 101107.01.07 E8

КвРІСТ 101107.01.07 E8					
Зм	Арк	Недокум	Підпис	Дата	
Розробив	Арк	Недаквіський			
Перевірив	Регіда				
Н.контр.	Лісенко				
Затв.	Говоруценко				
Діаграма функціонування команд чат-бота					
Літера					
У					
Аркуш					
3					
3					
ХНУ, ІСТ-20-1					

Додаток Г

Повний код чат-бота на Python

```
import aiohttp
import asyncio
import requests
import random
from aiogram import Bot, Dispatcher, types
from aiogram.types import InputFile
from aiogram.utils import executor
from dotenv import load_dotenv
import os
import sqlite3

load_dotenv()
TOKEN = "6823680833:AAFGrbXr2B4FGKUbJwQ6nZyteYamy-od308"

bot = Bot(token=TOKEN)
dp = Dispatcher(bot)

conn = sqlite3.connect('words.db')
cursor = conn.cursor()

cursor.execute('''
    CREATE TABLE IF NOT EXISTS words (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        english TEXT NOT NULL,
        ukrainian TEXT NOT NULL
    )
''')
conn.commit()

async def add_word(english, ukrainian):
    try:
        cursor.execute("INSERT INTO words (english, ukrainian) VALUES (?, ?)",
            (english, ukrainian))
        conn.commit()
        return True
    except Exception as e:
        print(f"Error adding word: {e}")
        return False
```

```

async def get_translation(word):
    cursor.execute("SELECT ukrainian FROM words WHERE english = ?", (word,))
    result = cursor.fetchone()
    if result:
        return result[0]
    else:
        return None

async def get_audio_url(word):
    URL:= f"https://api.dictionaryapi.dev/media/pronunciations/en/{word}-uk.mp3"
    async with aiohttp.ClientSession() as session:
        try:
            async with session.get(url) as response:
                if response.status == 200:
                    return url
                else:
                    print(f"Error fetching audio URL: {response.status}")
                    return None
        except aiohttp.client_exceptions.InvalidURL:
            print(f"Invalid URL: {url}")
            return None

async def get_definition(word):
    URL:= f"https://api.dictionaryapi.dev/api/v2/entries/en/{word}"
    async with aiohttp.ClientSession() as session:
        try:
            async with session.get(url) as response:
                if response.status == 200:
                    data = await response.json()
                    definition = data[0]["meanings"][0]["definitions"][0]["definition"]
                    pronunciation = data[0]["phonetics"][0]["text"]
                    return definition, pronunciation
                else:
                    print(f"Error fetching definition: {response.status}")
                    return None, None
        except aiohttp.client_exceptions.InvalidURL:
            print(f"Invalid URL: {url}")
            return None, None

async def get_examples(word):
    URL:= f"https://api.dictionaryapi.dev/api/v2/entries/en/{word}"

```

```

async with aiohttp.ClientSession() as session:
    try:
        async with session.get(url) as response:
            if response.status == 200:
                data = await response.json()
                examples = data[0]["meanings"][0]["definitions"][0]["example"]
                return examples
            else:
                print(f"Error fetching examples: {response.status}")
                return None
    except aiohttp.client_exceptions.InvalidURL:
        print(f"Invalid URL: {url}")
        return None

```

```

async def get_synonyms(word):
    URL:= f"https://api.dictionaryapi.dev/api/v2/entries/en/{word}"
    async with aiohttp.ClientSession() as session:
        try:
            async with session.get(url) as response:
                if response.status == 200:
                    data = await response.json()
                    synonyms = []
                    for meaning in data[0]["meanings"]:
                        if "synonyms" in meaning:
                            synonyms.extend(meaning["synonyms"])
                    return synonyms
                else:
                    print(f"Error fetching synonyms: {response.status}")
                    return []
        except aiohttp.client_exceptions.InvalidURL:
            print(f"Invalid URL: {url}")
            return []

```

```

@dp.message_handler(commands=["start"])
async def start(message: types.Message):
    await message.reply("Привіт! Я телеграм бот, який видає визначення англійських
слів, показує транскрипцію та грає аудіо. "
        "Використовуйте команди: \n"
        "/definition [слово] - для отримання визначення та транскрипції\n"
        "/audio [слово] - для отримання аудіофайлу\n")

```

```
"/example [слово] - для отримання прикладів використання слова\n"  
"/add [англійське слово] [український переклад] - для додавання  
нового слова в словник\n"  
"/translate [англійське слово] - для отримання перекладу з  
англійської мови на українську\n"  
"/synonyms [слово] - для отримання синонімів\n"  
)
```

```
@dp.message_handler(commands=["definition"])
```

```
async def handle_definition(message: types.Message):
```

```
    word = message.text.split()[1]
```

```
    definition, pronunciation = await get_definition(word)
```

```
    if definition:
```

```
        await message.reply(f"Визначення слова: {word}\nТранскрипція:  
{pronunciation}\nВизначення: {definition}")
```

```
    else:
```

```
        await message.reply(f"Не вдалося знайти визначення слова {word}")
```

```
@dp.message_handler(commands=["audio"])
```

```
async def handle_audio(message: types.Message):
```

```
    word = message.text.split()[1]
```

```
    audio_URL:= await get_audio_url(word)
```

```
    if audio_url:
```

```
        await message.reply(f"Ось посилання на слово: '{word}'",
```

```
            reply_markup=types.InlineKeyboardMarkup().add(
```

```
                types.InlineKeyboardButton(text="Слухати вимову", url=audio_url)
```

```
            ))
```

```
    else:
```

```
        await message.reply(f"Не вдалося знайти аудіо для слова '{word}'")
```

```
@dp.message_handler(commands=["example"])
```

```
async def handle_example(message: types.Message):
```

```
    word = message.text.split()[1]
```

```
    examples = await get_examples(word)
```

```
    if examples:
```

```
        await message.reply(f"Приклади використання слова {word}:\n{examples}")
```

```
    else:
```

```
        await message.reply(f"Не вдалося знайти приклади використання слова {word}")
```

```
@dp.message_handler(commands=["synonyms"])
```

```

async def handle_synonyms(message: types.Message):
    word = message.text.split()[1]
    synonyms = await get_synonyms(word)
    if synonyms:
        await message.reply(f"Синоніми слова: {word}\n{' '.join(synonyms)}")
    else:
        await message.reply(f"Не вдалося знайти синоніми слова {word}")

@dp.message_handler(commands=["add"])
async def handle_add_word(message: types.Message):
    try:
        english_word = message.text.split()[1]
        ukrainian_translation = message.text.split()[2]
        success = await add_word(english_word, ukrainian_translation)
        if success:
            await message.reply(f"Слово '{english_word}' з перекладом '{ukrainian_translation}' додано до словника.")
        else:
            await message.reply(f"Виникла помилка при додаванні слова.")
    except IndexError:
        await message.reply(f"Неправильний формат команди. Використовуйте '/add [англійське слово] [український переклад]'")

@dp.message_handler(commands=["translate"])
    async def handle_translate(message: types.Message):
        try:
            english_word = message.text.split()[1]
            translation = await get_translation(english_word)
            if translation:
                await message.reply(f"Переклад слова '{english_word}': {translation}")
            else:
                await message.reply(f"Слово '{english_word}' не знайдено в словнику.")
        except IndexError:
            await message.reply(f"Неправильний формат команди. Використовуйте '/translate [англійське слово]'")

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

```

Ім'я користувача:
Кафедра КІ

ID перевірки:
1016375299

Дата перевірки:
19.06.2024 14:01:47 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
19.06.2024 14:03:35 EEST

ID користувача:
100005591

Назва документа: Недашківський_Інформаційна система для вивчення іноземних мов

Кількість сторінок: 64 Кількість слів: 13384 Кількість символів: 102901 Розмір файлу: 1.16 MB ID файлу: 1016183184

2.44% Схожість

Найбільша схожість: 0.78% з джерелом з Бібліотеки (ID файлу: 1014915260)

2.07% Джерела з Інтернету 136 Сторінка 66

1.7% Джерела з Бібліотеки 128 Сторінка 67

0% Цитат

Цитати 1 Сторінка 68

Посилання 1 Сторінка 68

0% Вилучень

Немає вилучених джерел

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 1.0%

Словник перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 8%

ID: 131539 Назва: БКР Інформаційна система для вивчення іноземних мов Додано в БД: 2024-06-19 Автора: Р. В. Недашківський Керівник: П. Г. Регіда Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	92968	688	1584 (2%)	23 (3%)
	Джерело плагіату			

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Недашківський Роман Вадимович

Тема: Інформаційні системи для вивчення іноземних мов

Спеціальність: 126 «Інформаційні системи та технології»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 58

1. Короткий зміст роботи та прийнятих рішень: Мета кваліфікаційної роботи - розробити та впровадити інформаційну систему у вигляді чат-боту для вивчення іноземної мови.
2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.
3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: В першому розділі кваліфікаційної роботи проведено дослідження предметної області, а саме: проаналізовані інформаційні системи для вивчення іноземних мов, проведено порівняння існуючих інформаційних систем між собою. Також виконано постановку задачі дослідження. В другому розділі роботи представлено обґрунтування вибору мови програмування Python, представлено використання бази даних SQLite та Dictionary API для реалізації чат-боту та його функціональних особливостей. В третьому розділі було представлено ключові аспекти реалізації та демонстрацію розробленого чат-боту для вивчення іноземних мов, через використання відповідних команд чат-боту.
4. Позитивні сторони роботи: Завдяки використанню Dictionary API, чат-бот сприяє покращенню процесу вивчення нових слів, застосовуючи різні навчальні підходи.
5. Негативні сторони роботи: В проєкті не реалізовані засоби ідентифікації користувачів, через що неможливо зберігати прогрес вивчення нових слів.

6. Оцінка графічного оформлення та пояснювальної записки роботи:
Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному технічному рівні.

8. Інші зауваження: _____

9. Оцінка дипломної роботи: Розглянувши позитивні та негативні сторони представленої дипломної роботи вважаю, що робота заслуговує оцінки «задовільно», 3.5 (D).

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) _____

Стасюк Максим Васильович С, викладач кафедри
хімії та біології, доцент

"15" травня 2024 р.

 (підпис)

Завідувачу кафедри КПС
д-р.техн.наук, проф. Говорущенко Т. О.

Недашківського Романа Вадимовича

ПБ здобувача вищої освіти

ФІТ, 4 курсу, групи ІСТ-20-1

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

3 червня 2024 року



РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Інформаційна система для вивчення іноземних мов

Автор: Недашківський Роман Вадимович

Спеціальність: 126– Інформаційні системи та технології

Освітня програма: освітньо-професійна

Науковий керівник: Регіда Павло Геннадійович, старший викладач

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

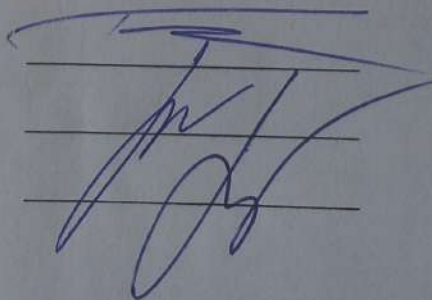
- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) схожість із джерелами в інтернеті складає 2.07% і адресується на 136 першоджерел;
- 4) схожість із джерелами в бібліотеці складає 1.7% і адресується на 128 першоджерел;

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 2.44% і адресується до 264 першоджерела, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КІС



П. Г. Регіда

Є.Г. Гнатчук

Т. О. Говорущенко