

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Державний університет Житомирська політехніка

Кафедра комп'ютерних наук

КУРСОВА РОБОТА (ПОЯСНЮВАЛЬНА ЗАПИСКА)

з дисципліни: «Мови інтелектуального аналізу даних»
на тему:
«Розробка telegram-бота для перевірки рівня знань на співбесіді»

студентки I курсу групи ІСТ-21-1
спеціальності 126 «Інформаційні системи
та технології»

Городецька Вікторія Валеріївна
(прізвище, ім'я та по-батькові)

Керівник: ст. викл. каф. КН Марчук Г. В.

Дата захисту: "___" _____ 2022 р.

Національна шкала _____

Кількість балів: _____

Оцінка: ECTS _____

Члени комісії:

(підпис)

(прізвище та ініціали)

(підпис)

(прізвище та ініціали)

(підпис)

(прізвище та ініціали)

Житомир – 2022

Державний університет Житомирська політехніка
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки
Освітній рівень: магістр
Спеціальність 126 «Інформаційні системи та технології»

«ЗАТВЕРДЖУЮ»

Зав. кафедри

_____Марчук Г.В.
“___” _____ 2022р.

ЗАВДАННЯ
НА КУРСОВУ РОБОТУ СТУДЕНТУ
Городецькій Вікторії Валеріївні

- Тема роботи: Розробка telegram-бота для перевірки рівня знань на співбесіді.
- Строк подання студентом: “___” _____ 2022р.
- Вихідні дані до роботи: програма тестування в месенджері Telegram.
- Зміст розрахунково-пояснювальної записки(перелік питань, які підлягають розробці)
 - Аналіз проблематики, методів та засобів вирішення задач
 - Проектування та розробка програмного забезпечення
 - Опис роботи з програмним додатком та його тестування
- Перелік графічного матеріалу(з точним зазначенням обов'язкових креслень)
 - Презентація PowerPoint
 - Репозиторій з повним кодом проекту:
<https://drive.google.com/file/d/1UfK71OtCvnjGQS78ttA7qkul8I4QkdSF/view?usp=sharing>
- Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посади консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Марчук Г. В., ст. викладач каф. КН		

- Дата видачі завдання “1” _____ 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів курсової роботи	Строк виконання етапів роботи	Примітки
1	Постановка задачі	25 березня 2022- 30 березня 2022	
2	Аналіз теоретичних матеріалів та вивчення предметної області	1 квітня 2022 - 5 квітня 2022	
3	Вибір методів та засобів реалізації задачі	6 квітня 2022- 24 квітня 2022	
4	Реалізація проекту	27 квітня 2022 - 2 травня 2022	
5	Налагодження та перевірка програми	3 травня 2022 - 9 травня 2022	
6	Оформлення пояснювальної записки	10 травня 2022 - 20 травня 2022	
7	Захист	31 травня 2022	

Студент

(підпис)

Городецька В.В.

(прізвище та ініціали)

Керівник роботи

(підпис)

Марчук Г.В.

(прізвище та ініціали)

РЕФЕРАТ

Завданням на курсового проекту (роботи) було створення telegram-бота для перевірки рівня знань на співбесіді.

Пояснювальна записка до курсової роботи на тему «розробка telegram-бота для перевірки рівня знань на співбесіді» складається з переліку умовних скорочень, вступу, трьох розділів, висновків, списку використаної літератури.

Текстова частина викладена на 27 сторінках друкованого тексту.

Пояснювальна записка містить 10 рисунків та 2 таблиці. Список використаних джерел містить 10 найменувань і займає 1 сторінку. Загальний обсяг роботи – 33 сторінки.

Ключові слова: АНАЛІЗ, ЧАТ-БОТ, ДОСЛІДЖЕННЯ, КОРИСТУВАЧ, МЕССЕНДЖЕР, ТЕСТУВАННЯ, БАЗА ДАНИХ, ДАНІ, TELEGRAM, PYTHON, MONGODB

					Житомирська політехніка 22.126.8.000		
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>			
Розробив		<i>Городецька В.В.</i>			Пояснювальна записка	<i>Літ.</i>	<i>Арк.</i>
Керівник		<i>Марчук Г.В.</i>					
Реценз						4	36
Н. контр.		<i>Морозов А.В.</i>				ФІКТ, Гр. ІСТ-21	
Затверд.							

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	6
ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ	9
1.1 Аналіз задачі, засобів та методів її вирішення	9
1.2 Мова програмування Python	10
1.3 База даних MongoDB	11
1.4 Середовище розробки	13
Висновки до першого розділу	15
РОЗДІЛ 2 ПРАКТИЧНА РЕАЛІЗАЦІЯ АЛГОРИТМІВ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ	16
2.1 Характеристика джерела даних для проведення аналізу	16
2.2 Створення Telegram бота та реалізація програмного коду	18
Висновки до другого розділу	24
РОЗДІЛ 3 ТЕСТУВАННЯ ТА АНАЛІЗ РОЗРОБЛЕНОЇ СИСТЕМИ	25
3.1 Тестування Telegram бота	25
Висновки до третього розділу	29
ВИСНОВКИ	30
СПИСОК ЛІТЕРАТУРИ	31

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ПЗ – Програмне забезпечення.

API – Application Programming Interface (Прикладний програмний інтерфейс).

СКБД – система керування базами даних.

БД – база даних.

JSON – JavaScript Object Notation (Текстовий формат обміну даними між комп'ютерами).

SQL – Structured Query Language (Мова структурованих запитів).

OS – Operating System (Операційна система).

CVS – Concurrent Versions System (Система одночасних версій)

		Городецька В.В			Житомирська політехніка 22.126.8.000	Арк.
		Марчук Г.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

ВСТУП

У сучасному світі мобільні пристрої розвинені як величезна інфраструктура, тому є великий попит на мобільні додатки. Особливо велику популярність мають соціальні мережі та месенджери. Одним із самих популярних таких месенджерів є Telegram, який віддалено нагадує соціальну мережу, завдяки своїй платформі. Тому раціональним лишається бути у тренді, користуючись новинками першими.

Одним із головних плюсів для бізнесу та організаційної роботи в Телеграмі є так звані боти. Більшість сучасних компаній, організацій та онлайн-магазинів додали у свою діяльність Telegram-бота для вирішення певних задач, такі як взнати ту чи іншу інформацію, або зробити замовлення. Для розробника ПЗ це гарна можливість збільшити діапазон своєї роботи. Адже боти створені щоб спростити деякі задачі для малого та великого бізнесу, і не тільки. Гарним прикладом є створення бота для тестування людей на співбесіді, завдяки ньому можна скоротити час, який був би витрачений на опитування для перевірки знань. Telegram є на всіх мобільних платформах: Android, iOS, Windows Phone та на десктопах: Windows, Linux/Ubuntu, MacOS. Користувач може користуватись Телеграмом на всіх цих операційних системах під одним користувачем і персональні дані будуть синхронізуватись.

Актуальність теми ботів в наш час є настільки велика, що генеральні директори крупних ІТ-компаній в світі вже запевнили своїх користувачів, що найближчим часом в їхніх продуктах також з'явиться можливість користування так званими ботами.

Мета курсової роботи: розробити мобільний чат-бот для перевірки знань на співбесіді.

Для того, щоб досягти мети даного дипломного проекту, необхідно реалізувати наступні поставлені задачі:

- провести детальний аналіз актуальності та необхідності створення чат-бота для широкого загалу користувачів;
- виконати реалізацію функціоналу чат-бота;

		Городецька В.В.			Житомирська політехніка 22.126.8.000	Арк.
		Марчук Г.В.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

– провести тестування розробленого продукту.

Об’єкт дослідження: чат-бот який дозволяє перевірити рівень знань на співбесіді .

Предмет дослідження: методи і засоби розробки чат-бота для перевірки рівня знань на співбесіді.

Практичне значення: розроблений чат-бот буде використовуватись під час проведення співбесіди, для тестування претендента на посаду, на результати якого в подальшому буде опиратись менеджер по набору персоналу. Дане тестування дозволить оцінити рівень знань претендента, щоб отримати бачення про його рівень кваліфікації.

Методи досліджень, технічні та програмні засоби: використання мови програмування Python для написання чат-боту для месенджера Telegram, використовуючи Telegram Bot API та системи керування базами даних MongoDB у середовищі PyCharm

		Городецька В.В			Житомирська політехніка 22.126.8.000	Арк.
		Марчук Г.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		8

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Аналіз задачі, засобів та методів її вирішення

Будь-яка співбесіда — стрес і хвилювання для претендента: окрім відмінного резюме та вміння триматися, завжди є так званий людський фактор. Зазвичай на співбесіді задаються заздалегідь підготовленні питання, для отримання бачення рівня кваліфікації.

Все частіше у великих компаніях усі можливі процеси автоматизують. Проведення тестування — не виняток. Згідно з прогнозами міжнародної аудиторсько-консалтингової компанії, до 2022 року 7 з 10 співробітників працюватимуть поза офісом, а найматимуть їх боти та програми штучного інтелекту. В Україні вже теж є компанії, де тестуванням кандидата займаються роботи — спеціально написані програми.

Для більшої схожості з людиною їх називають, даючи співзвучні з людськими імена. Бот оцінює рівень знань, які повинні відповідати вимогам вакансії.

Для автоматизації процесу тестування було вирішено створити чат-бота на платформі Telegram, який в режимі реального часу буде оцінювати рівень знань претендента, на результат якого в майбутньому буде опиратися HR-менеджер.

Робот або боти — це спеціальні акаунти в Telegram, які можуть автоматично обробляти і відправляти повідомлення. Вони створюються програмістами і працюють у них на сервері.

Боти можуть виконувати практично будь-які завдання, які може робити кожен користувач акаунту Telegram з онлайн-сервісами. Роботи можуть вчити, розважати, шукати, транслювати, нагадувати, з'єднувати та підключатися до інтернету речей. По суті, боти — це такий собі зручний для людини інтерфейс роботи з різноманітними веб-службами.

		Городецька В.В.			Житомирська політехніка 22.126.8.000	Арк.
		Марчук Г.В.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

Усе, що потрібно для роботи з ботами, – акаунт у Telegram. Для вас взаємодія з ними виглядатиме як спілкування в чаті, різниця лише в тому, що на іншому кінці не людина, а програма з початками штучного інтелекту.

У роботів немає статусів «онлайн» і «був у мережі», натомість відображається напис «робот». Окрім цього, боти не можуть самі почати спілкування. Обраного бота треба спершу додати в групу або першим почати з ним діалог. Для цього можна використовувати посилання виду telegram.me/<ім'я бота> або пошук за іменем користувача. Знайти ботів просто, оскільки в них ім'я завжди закінчується на «bot», наприклад, @MyPersonalWikiBot.

1.2. Мова програмування Python

Для реалізації поставленої задачі було обрано мову програмування Python.

Python – багатоцільова мова програмування, яка дозволяє писати код, що добре читається. Відносний лаконізм мови Python дозволяє створити програму, яка буде набагато коротше свого аналога, написаного на іншій мові.

Python - багатоплатформова мова програмування. Це означає, що програми на Python можна запускати в різних операційних системах без будь-яких змін.

Ще однією перевагою Python є його стандартна бібліотека, яка встановлюється разом з Python і містить готові інструменти для роботи з операційною системою, веб-сторінками, базами даних, різними форматами даних, для побудови графічного інтерфейсу програм тощо.

Програми, написані на мові програмування Python, можуть бути як невеликими скриптами, так і складними системами.

Один з можливих недоліків Python - швидкість виконання коду. Python не є компільованою мовою. Код на Python спочатку компілюється у внутрішній байт-код, який потім виконується інтерпретатором Python. У більшості випадків при використанні Python виходять програми повільніші в порівнянні з такими мовами, як C.

Втім, сучасні комп'ютери мають таку обчислювальну потужність, що для

		Городецька В.В.			Житомирська політехніка 22.126.8.000	Арк.
		Марчук Г.В.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

більшості застосунків швидкість розробки важливіша швидкості виконання, а програми на Python зазвичай пишуться набагато швидше.

Окрім того, Python легко розширюється модулями, написаними на C або C++. Такі модулі можуть використовуватися для виконання частин програми, що створюють інтенсивне навантаження на процесор. В даній курсовій роботі було використано бібліотеки pyTelegramBotAPI та pymongo.

PyTelegramBotAPI хороша та легка бібліотека для створення бота на Python на платформі Telegram

1.3. База даних MongoDB

Для зберігання, аналізу та обробки даних було обрано систему керування базами даних MongoDB

MongoDB – це база даних, яка є документо-орієнтована з відкритим вихідним кодом і вона не вимагає опису схем таблиць. Головні риси та можливості:

- підтримка відмовостійкості і масштабованості;
- асинхронна реплікація, набір реплік і розподілу бази даних на вузли;
- JSON-подібна схема зберігання даних;
- ефективне зберігання великих об'єктів, адміністративний інтерфейс, серверні функції, Map / Reduce і інше;
- використання Javascript в якості мови для формування запитів;
- профілювання запитів;
- широкий набір атомарних операцій над даними (умовний пошук, складна вставка / оновлення тощо);
- різні типи даних (в тому числі підтримка масивів); підтримка індексів (B-Tree);
- повнотекстовий пошук, в тому числі українською мовою, з підтримкою морфології;
- журнал операцій, що змінюють дані в базі даних.

		Городецька В.В.			Житомирська політехніка 22.126.8.000	Арк.
		Марчук Г.В.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

MongoDB пропонує документо-орієнтовану модель даних на відміну від реляційних баз даних, завдяки цьому вона працює в рази швидше, має кращу масштабованість та її легше використовувати. Однак, враховуючи всі недоліки традиційних баз даних і переваги MongoDB, важливим розумінням є те, що завдання бувають різні і методи їх вирішення бувають також різні.

В певній ситуації MongoDB дійсно здатна поліпшити продуктивність програми, для прикладу візьмемо ситуацію, що існує потреба зберігати складні за структурою дані. В іншій же ситуації краще буде використовувати традиційні реляційні бази даних.

MongoDB може представляти не тільки єдину базу даних, що знаходиться на одному фізичному сервері. Функціональність MongoDB дозволяє розташувати декілька баз даних на декількох фізичних серверах, і ці бази даних зможуть легко обмінюватися даними і зберігати цілісність.

Ось кілька важливих властивостей цієї бази даних:

- формат даних в MongoDB. Одним з популярних стандартів обміну даними та їх зберігання є JSON (JavaScript Object Notation). JSON ефективно описує складні за структурою дані. Спосіб зберігання даних в MongoDB в цьому плані схожий на JSON, хоча формально JSON не використовується. Для зберігання в MongoDB застосовується формат, який називається BSON або скорочення від binary JSON. BSON дозволяє працювати з даними швидше: швидше виконується пошук і обробка. Хоча треба зазначити, що BSON на відміну від зберігання даних в форматі JSON має невеликий недолік: в цілому дані в JSON- форматі займають менше місця, ніж в форматі BSON, з іншого боку, даний недолік з лишком окупається швидкістю;
- кросплатформеність. MongoDB написана на C ++, тому її легко перенести на різні платформи. MongoDB може бути розгорнута на платформах Windows, Linux, MacOS, Solaris. Можна також завантажити вихідний код і самому скомпілювати MongoDB, але

		Городецька В.В.			Житомирська політехніка 22.126.8.000	Арк.
		Марчук Г.В.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

- рекомендується використовувати бібліотеки з офіційного сайту;
- колекції. Якщо в традиційному світі SQL є таблиці, то в світі MongoDB є колекції. І якщо в реляційних БД таблиці зберігають однотипні жорстко структуровані об'єкти, то в колекції можуть містити найрізноманітніші об'єкти, що мають різну структуру і різний набір властивостей;
 - реплікація. Система зберігання даних в MongoDB представляє набір реплік. У цьому наборі є основний вузол, а також може бути набір вторинних вузлів. Всі вторинні вузли зберігають цілісність і автоматично оновлюються разом з оновленням головного вузла. І якщо основний вузол з якихось причин виходить з ладу, то один з вторинних вузлів стає головним;
 - простота у використанні. Відсутність жорсткої схеми бази даних і в зв'язку з цим потреби при щонайменшій зміні концепції зберігання даних створювати знову цю схему значно полегшують роботу з базами даних MongoDB і подальшим їх масштабуванням. Крім того, економиться час розробників. Їм більше не треба думати про нове створення бази даних і витратити час на побудову складних запитів;

1.4. Середовище розробки

PyCharm — інтегроване середовище розробки для мови програмування Python. Надає засоби для аналізу коду, графічний зневаджувач, інструмент для запуску юніт-тестів і підтримує веброзробку на Django. PyCharm розроблена на основі IntelliJ IDEA.

PyCharm працює під операційними системами Windows, Mac OS X і Linux.

		Городецька В.В			Житомирська політехніка 22.126.8.000	Арк.
		Марчук Г.В.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

Можливості:

- Статичний аналіз коду, підсвічування синтаксису і помилок;
- навігація серед проектів і сирцевого коду: відображення файлової структури проекту, швидкий перехід між файлами, класами і методами;
- рефакторинг: перейменування, витяг методу, введення змінної, введення константи, підняття і опускання методу тощо;
- інструменти для веброзробки з використанням фреймворку Django
- вбудований зневаджувач для Python;
- вбудовані інструменти для юніт-тестування;
- розробка з використанням Google App Engine;
- підтримка систем контролю версій: загальний користувацький інтерфейс для Mercurial, Git, Subversion, Perforce і CVS з підтримкою списків змін та злиття.

		Городецька В.В.			Житомирська політехніка 22.126.8.000	Арк.
		Марчук Г.В.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновки до першого розділу

Отже, в даному розділі було проведено аналіз задачі, засобів та методів її вирішення, в результаті якого визначено необхідність створення чат-бота для перевірки рівня знань на співбесіді. Визначено мову програмування, джерело даних та середовище розробки проекту.

		Городецька В.В			Житомирська політехніка 22.126.8.000	Арк.
		Марчук Г.В.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2 ПРАКТИЧНА РЕАЛІЗАЦІЯ АЛГОРИТМІВ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ

2.1 Характеристика джерела даних для проведення аналізу

В якості джерела даних було використано систему керування базами даних MongoDB.

В ході проектування БД в системі керування базами даних MongoDB створено наступні колекції:

- Bot
- Users

Для збереження даних про питання та відповіді тесту була призначена колекція Bot (таблиця 2.1 та рис. 2.1).

Таблиця 2.1

Колекція Bot

Назва	Тип даних	Опис
id	Int32	Ідентифікатор запитання
text	String	Текст запитання
answers	Array	Масив даних з відповідями
correct	Int32	Правильна відповідь

Дані у колекцію User генеруються автоматично при старті, та змінюються впродовж усього тестування (таблиця 2.2 та рис. 2.2).

Колекція Users

Назва	Тип даних	Опис
chat_id	Int32	Ідентифікатор чату (користувача)
is_passing	Boolean	Статус тестування (проходить)
is_passed	Boolean	Статус тестування (пройдено)
question_index	Int32	Номер запитання на даний момент проходження тестування
answers	Array	Масив з обраними відповідями

```

_id: ObjectId("62dfd768cb2b8ed515f2593f")
id: 0
text: "Скільки різних груп крові у людини?"
✓ answers: Array
  0: "1"
  1: "2"
  2: "3"
  3: "4"
  4: "5"
  5: "6"
correct: 3

```

Рис. 2.1. Візуалізація даних колекції Bot в MongoDB

```

_id: ObjectId("62e3dcac0428971e9a0f341b")
chat_id: 608881332
is_passing: true
is_passed: false
question_index: 1
✓ answers: Array
  0: 0

```

Рис. 2.2. Візуалізація даних колекції Users в MongoDB

		Городецька В.В.			Житомирська політехніка 22.126.8.000	Арк.
		Марчук Г.В.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

2.2 Створення Telegram бота та реалізація програмного коду

Перед написання програмного коду за допомогою офіційного бота Telegram @BotFather потрібно створити нового бота, та отримати його API Токен (рис. 2.3).

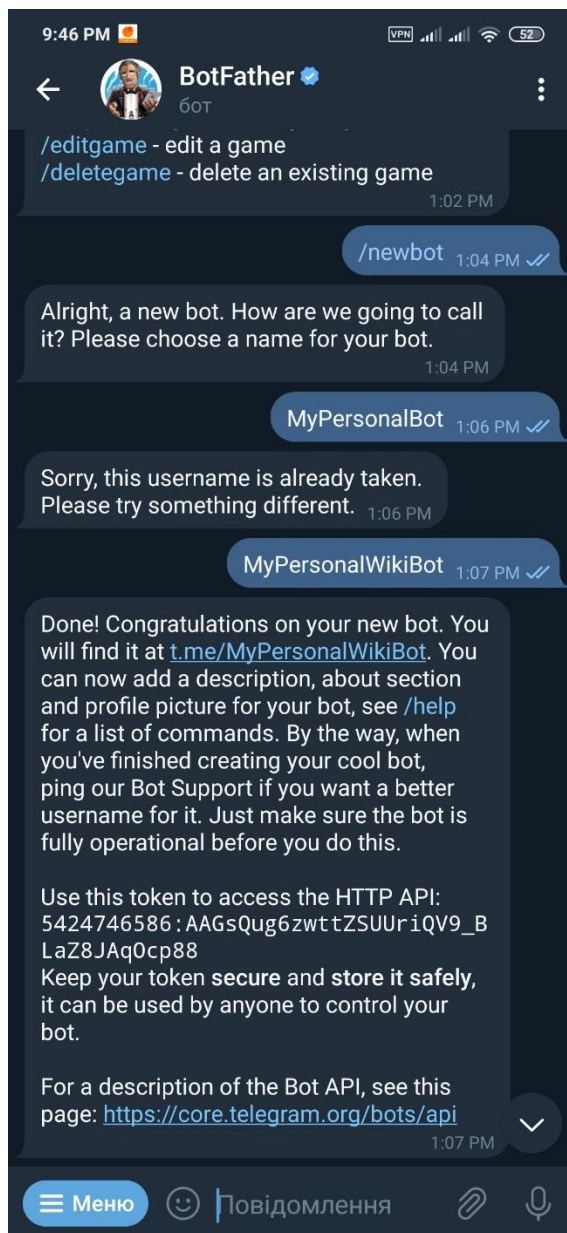


Рис. 2.3. Створення бота

		Городецька В.В.			Житомирська політехніка 22.126.8.000	Арк.
		Марчук Г.В.				18
Змн.	Арк.	№ докум.	Підпис	Дата		

Початком написання коду є підключення бібліотек `pyTelegramBotAPI` та `pymongo`. Фрагмент коду підключення бібліотек:

```
import telebot
from pymongo import MongoClient
```

Далі було виконано підключення API токена бота. Фрагмент коду підключення токена:

```
bot = telebot.TeleBot("5424746586:AAGX3cqvuxfO8II_cdXe_NsWJPfswSNRxl")
```

Наступним кроком підключаємо БД `MongoDB`, отримуємо колекції та підраховуємо кількість питань, створивши клас `DataBase` в якому будуть функції для керування БД. Фрагмент коду наведено нижче:

```
class DataBase:
    def __init__(self):
        cluster = MongoClient("mongodb+srv://user:12345@cluster0.0dsh8.mongodb.net/?retryWrites=true&w=majority")
        self.db = cluster["WikiBot"]
        self.users = self.db["Users"]
        self.questions = self.db["Bot"]
        self.questions_count = len(list(self.questions.find({})))
```

Ідентифікація користувача буде відбуватися по `chat_id`. Створюємо функцію яка повертає користувача по `chat_id`, якщо користувача немає в БД, створиться новий. Фрагмент коду наведено нижче:

```
user = self.users.find_one({"chat_id": chat_id})
    if user is not None:
        return user
    user = {
        "chat_id": chat_id,
        "is_passing": False,
        "is_passed": False,
        "question_index": None,
        "answers": []
    }
    self.users.insert_one(user)
    return user
```

		Городецька В.В.			Житомирська політехніка 22.126.8.000	Арк.
		Марчук Г.В.				19
Змн.	Арк.	№ докум.	Підпис	Дата		

Створюємо функцію яка буде змінювати параметри користувача по chat_id. Фрагмент коду наведено нижче:

```
def set_user(self, chat_id, update):  
    self.users.update_one({"chat_id": chat_id}, {"$set": update})
```

Створюємо функцію, яка буде повертати питання по його id. Фрагмент коду наведено нижче:

```
def get_question(self, index):  
    return self.questions.find_one({"id": index})
```

```
db = DataBase()
```

Створюємо функцію яка буде викликатися командою /start, отримаємо користувача, якщо користувач проходив тест, тобто якщо параметр is_passed=true, тоді буде відправлено повідомлення, що другий раз тест пройти неможливо. Якщо користувач проходить тест в даний момент, тобто параметри is_passed=false та is_passing=true, нічого відправлятися не буде. Фрагмент коду наведено нижче:

```
@bot.message_handler(commands=["start"])  
def start(message):  
    user = db.get_user(message.chat.id)  
    if user["is_passed"]:  
        bot.send_message(message.from_user.wid, "Ви вже пройшли це  
тестування. Вдруге пройти неможливо")  
        return  
    if user["is_passing"]:  
        return  
    db.set_user(message.chat.id, {"question_index": 0, "is_passing": True})  
    user = db.get_user(message.chat.id)  
    post = get_question_message(user)  
    if post is not None:  
        bot.send_message(message.from_user.id, post["text"],  
reply_markup=post["keyboard"])
```

Створюємо функцію яка буде повертати текст і клавіатуру питання. Якщо користувач пройшов тест, тобто коли question_index буде дорівнювати кількості запитань, буде рахуватися відсоток правильних відповідей, в інших випадках

		Городецька В.В.			Житомирська політехніка 22.126.8.000	Арк.
		Марчук Г.В.				20
Змн.	Арк.	№ докум.	Підпис	Дата		

створюється пуста клавіатура. Проходимось по усіх відповідях, та додаємо необхідні кнопки в клавіатуру. Оновлюємо параметри користувача Створюємо змінну з текстом повідомлення, далі повернемо словник з текстом і клавіатурою. Фрагмент коду наведено нижче:

```
def get_question_message(user):
    if user["question_index"] == db.questions_count:
        count = 0
        for question_index, question in enumerate(db.questions.find({})):
            if question["correct"] == user["answers"][question_index]:
                count += 1
        percents = round(100 * count / db.questions_count)

        if percents < 40:
            smile = "😞"
        elif percents < 60:
            smile = "😐"
        elif percents < 90:
            smile = "😊"
        else:
            smile = "👍"

        text = f"Ви відповіли правильно на {percents}% питань {smile}"
        db.set_user(user["chat_id"], {"is_passed": True, "is_passing": False})
        return {
            "text": text,
            "keyboard": None
        }

    question = db.get_question(user["question_index"])
    if question is None:
        return

    keyboard = telebot.types.InlineKeyboardMarkup()
    for answer_index, answer in enumerate(question["answers"]):
        keyboard.row(telebot.types.InlineKeyboardButton(f"{chr(answer_index +
97))} {answer}",
            callback_data=f"?ans&{answer_index}"))
```

		Городецька В.В.			Житомирська політехніка 22.126.8.000	Арк.
		Марчук Г.В.				21
Змн.	Арк.	№ докум.	Підпис	Дата		

```

text = f"Питання №{user['question_index'] + 1}\n\n{question['text']}"
return {
    "text": text,
    "keyboard": keyboard
}

```

Створюємо функцію, яка буде викликатися коли користувач натиснув на відповідь, в ній перевіряються параметри `is_passed` та `is_passing`, далі додаємо відповідь користувача в його відповіді. Фрагмент коду наведено нижче:

```

@bot.callback_query_handler(func=lambda query: query.data.startswith("?ans"))
def answered(query):
    user = db.get_user(query.message.chat.id)

    if user["is_passed"] or not user["is_passing"]:
        return
    user["answers"].append(int(query.data.split("&")[1]))
    db.set_user(query.message.chat.id, {"answers": user["answers"]})
    post = get_answered_message(user)
    if post is not None:
        bot.edit_message_text(post["text"], query.message.chat.id, query.message.id,
                               reply_markup=post["keyboard"])

```

Далі створюємо функцію, яка буде повертати що потрібно відповісти користувачу, в ній створюємо змінну з текстом, в змінну додаємо номер і текст питання. Проходимось по усіх відповідях, галочкою відмітимо правильну відповідь, хрестиком неправильну Створимо кнопку «Далі». Фрагмент коду наведено нижче:

```

def get_answered_message(user):
    question = db.get_question(user["question_index"])
    text = f"Питання №{user['question_index'] + 1}\n\n{question['text']}\n"
    for answer_index, answer in enumerate(question["answers"]):
        text += f"{chr(answer_index + 97)} {answer}"
        if answer_index == question["correct"]:
            text += " ✓"
        elif answer_index == user["answers"][-1]:
            text += " ✗"

```

		Городецька В.В.			Житомирська політехніка 22.126.8.000	Арк.
		Марчук Г.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		22

```

        text += "\n"

    keyboard = telebot.types.InlineKeyboardMarkup()
    keyboard.row(telebot.types.InlineKeyboardButton("Далі", callback_data="?next"))
    return {
        "text": text,
        "keyboard": keyboard
    }

```

Створюємо функцію, яка буде викликатися при натисненні кнопки «Далі», в ній збільшуємо індекс питання на 1, та відправляємо текст питання. Фрагмент коду наведено нижче:

```

@bot.callback_query_handler(func=lambda query: query.data == "?next")
def next(query):
    user = db.get_user(query.message.chat.id)

    if user["is_passed"] or not user["is_passing"]:
        return

    user["question_index"] += 1
    db.set_user(query.message.chat.id, {"question_index": user["question_index"]})
    post = get_question_message(user)
    if post is not None:
        bot.edit_message_text(post["text"], query.message.chat.id, query.message.id,
                               reply_markup=post["keyboard"])

```

		Городецька В.В			Житомирська політехніка 22.126.8.000	Арк.
		Марчук Г.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		23

Висновки до другого розділу

Отже, в даному розділі було описано типи даних та у якому вигляді вони зберігаються. Створено telegram-бота за допомогою офіційного бота додатку Telegram @BotFather, реалізовано та детально описано програмний код, який було написано на багатоцільовій мові програмування Python, з використанням таких бібліотек як pyTelegramBotAPI та pymongo.

		Городецька В.В			Житомирська політехніка 22.126.8.000	Арк.
		Марчук Г.В.				24
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3 ТЕСТУВАННЯ ТА АНАЛІЗ РОЗРОБЛЕНОЇ СИСТЕМИ

3.1 Тестування Telegram бота

Тестування бота здійснювалося вручну в додатку Telegram. Для початку тестування потрібно відкрити чат з ботом, знайшовши його в пошуку за нікнеймом @MyPersonalWikiBot або перейшовши за посиланням яке було видане ботом @BotFather. Процес пошуку та відкритий чат з ботом зображено на рис. 3.1 та рис 3.2.



Рис. 3.1 Пошук telegram бота

		Городецька В.В.			Житомирська політехніка 22.126.8.000	Арк.
		Марчук Г.В.				25
Змн.	Арк.	№ докум.	Підпис	Дата		

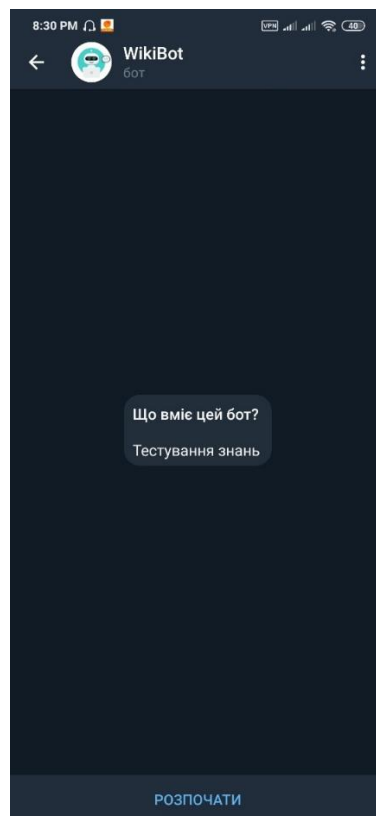


Рис. 3.2 Вікно чату з ботом

Для того щоб запустити бота, потрібно натиснути клавішу «Розпочати», або відправити команду /start. Після запуску бота, одразу з'явиться перше запитання (рис. 3.3).

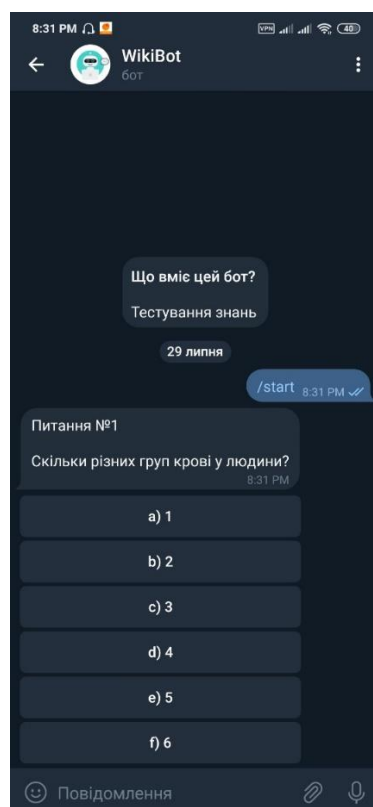


Рис. 3.3 Запуск бота

		Городецька В.В			Житомирська політехніка 22.126.8.000	Арк.
		Марчук Г.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		26

Після вибору відповіді, бот замінює повідомлення на нове, в якому відображаються дані відповіді, і кнопка «Далі», щоб перейти до наступного питання. Галочкою позначається правильна відповідь, хрестиком неправильна (рис.3.4 та рис.3.5).

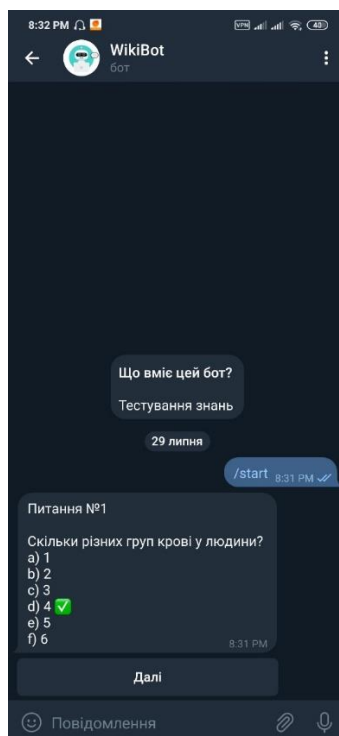


Рис. 3.4 Повідомлення про правильну відповідь

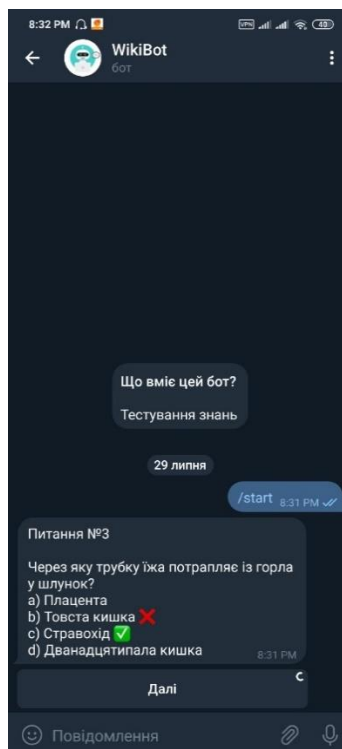


Рис. 3.4 Повідомлення про неправильну відповідь

		Городецька В.В			Житомирська політехніка 22.126.8.000	Арк.
		Марчук Г.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		27

Після проходження усього тесту, з'являється повідомлення з кількістю правильних відповідей у відсотках (рис. 3.5)

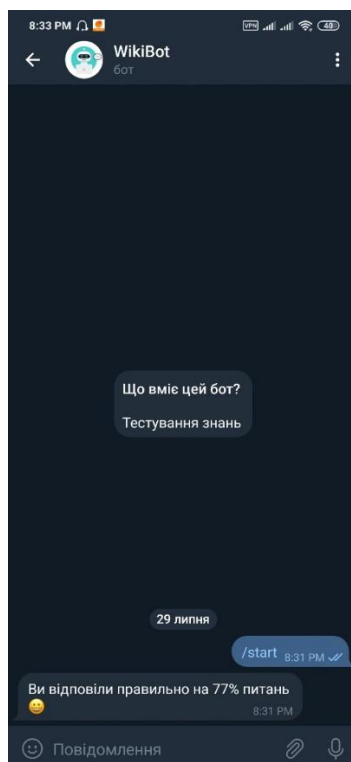


Рис. 3.5 Результат проходження тесту

При повторній спробі пройти тест з'явиться повідомлення «Ви вже пройшли це тестування. Вдруге пройти неможливо» (рис. 3.6).

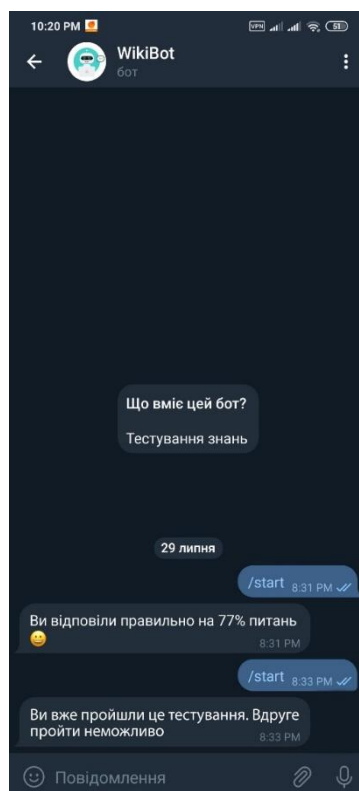


Рис. 3.6 Повторна спроба пройти тестування

		Городецька В.В.			Житомирська політехніка 22.126.8.000	Арк.
		Марчук Г.В.				28
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновки до третього розділу

Отже, в даному розділі було проведено тестування розробленого telegram-бота. Проведено аналіз різних сценаріїв, наприклад проведення повторного тестування для перевірки знань.

		Городецька В.В			Житомирська політехніка 22.126.8.000	Арк.
		Марчук Г.В.				29
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

В ході виконання дипломної роботи було проведено аналіз задачі, засобів та методів її вирішення, в результаті якого визначено необхідність створення чат-бота для перевірки рівня знань на співбесіді. В результаті дослідження було обрано мову програмування Python, джерело даних MongoDB та середовище розробки проекту PyCharm.

Було описано типи даних та у якому вигляді вони зберігаються. Створено telegram-бота за допомогою офіційного бота додатку Telegram @BotFather, реалізовано та детально описано програмний код, який було написано на багатоцільовій мові програмування Python, з використанням таких бібліотек як pyTelegramBotAPI та pymongo.

Результатом проведеного дослідження стало створення telegram-бота для перевірки рівня знань на співбесіді. Було проведено тестування розробленого telegram-бота. Проведено аналіз різних сценаріїв, наприклад проведення повторного тестування для перевірки знань.

		Городецька В.В.			Житомирська політехніка 22.126.8.000	Арк.
		Марчук Г.В.				30
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ЛІТЕРАТУРИ

1. Бенгфорт Б. Прикладной анализ текстовых данных на Python. Машинное обучение и создание приложений обработки естественного языка / Б. Бенгфорт, Р. Билбро, Т. Охеда. – СПб: Питер, 2019. – 368 с.
2. Бизли, Дэвид М. Язык программирования Python. Справочник. - К.: ДиаСофт, 2000. - 336 с. - ISBN 966-7393-54-2, ISBN 0-7357-0901-7
3. Дэвидсон-Пайлон К. Вероятностное программирование на Python: байесовский вывод и алгоритмы / Кэмерон Дэвидсон-Пайлон. – СПб: Питер, 2019. – 256 с.
4. И.А. Хахаев Практикум по алгоритмизации и программированию на Python. Учебник. - М.: Альт Линукс, 2010. - 126 с. - (Библиотека ALT Linux). - ISBN 978-5-905167-02-7
5. Марк Лутц. Программирование на Python, 4-е видання, II том - Переклад з англійської. - СПб.: Символ-Плюс, 2011
6. Марк Саммерфилд. Программирование на Python 3. Подробное руководство. - Переклад з англійської. - СПб.: Символ-Плюс, 2009. - 608 с - ISBN 978-5-93286-161-5
7. Самоучитель Python [Електронний ресурс] : Режим доступу: <https://pythonworld.ru/samouchitel-python>
8. JSON[Електронний ресурс]. – Режим доступу: <https://www.json.org/json-ru.html>
9. Telegram Bot API [Електронний ресурс] – Режим доступу: <https://core.telegram.org/bots/api>
10. Створювання бота. Telebot [Електронний ресурс]. – Режим доступу: <https://medium.com/@plug2up/%D1%81%D0%BE%D0%B7%D0%B4%D0%B0%D0%B5%D0%BC%D1%82%D0%B5%D0%BB%D0%B5%D0%B3%D1%80%D0%B0%D0%BC%D0%B1%D0%BE%D1%82%D0%B0-2c8377d4d48>

		Городецкая В.В.			Житомирська політехніка 22.126.8.000	Арк.
		Марчук Г.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		31

ДОДАТОК А

		Городецька В.В			Житомирська політехніка 22.126.8.000	Арк.
		Марчук Г.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		32

Програмний код створеного додатку:

```
import telebot

from pymongo import MongoClient

bot = telebot.TeleBot("5424746586:AAGX-3cqvuXfO8II_cdXe_NsWJPfswSNRxl")

class DataBase:

    def __init__(self):

        cluster = MongoClient("mongodb+srv://user:12345@cluster0.0dsh8.mongodb.net/?retryWrites=true&w=majority")

        self.db = cluster["WikiBot"]
        self.users = self.db["Users"]
        self.questions = self.db["Bot"]
        self.questions_count = len(list(self.questions.find({ })))

    def get_user(self, chat_id):

        user = self.users.find_one({"chat_id": chat_id})

        if user is not None:

            return user

        user = {

            "chat_id": chat_id,

            "is_passing": False,

            "is_passed": False,

            "question_index": None,

            "answers": []

        }

        self.users.insert_one(user)

        return user

    def set_user(self, chat_id, update):

        self.users.update_one({"chat_id": chat_id}, {"$set": update})

    def get_question(self, index):

        return self.questions.find_one({"id": index})

db = DataBase()

@bot.message_handler(commands=["start"])

def start(message):

    user = db.get_user(message.chat.id)

    if user["is_passed"]:
```

		Городецька В.В.			Житомирська політехніка 22.126.8.000	Арк.
		Марчук Г.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		33

```

        bot.send_message(message.from_user.wid, "Ви вже пройшли це
тестування. Вдруге пройти неможливо")

        return

    if user["is_passing"]:
        return

    db.set_user(message.chat.id, {"question_index": 0, "is_passing": True})
    user = db.get_user(message.chat.id)
    post = get_question_message(user)
    if post is not None:
        bot.send_message(message.from_user.id, post["text"],
reply_markup=post["keyboard"])

@bot.callback_query_handler(func=lambda query: query.data.startswith("?ans"))
def answered(query):
    user = db.get_user(query.message.chat.id)
    if user["is_passed"] or not user["is_passing"]:
        return
    user["answers"].append(int(query.data.split("&")[1]))
    db.set_user(query.message.chat.id, {"answers": user["answers"]})
    post = get_answered_message(user)
    if post is not None:
        bot.edit_message_text(post["text"], query.message.chat.id, query.message.id,
reply_markup=post["keyboard"])

@bot.callback_query_handler(func=lambda query: query.data == "?next")
def next(query):
    user = db.get_user(query.message.chat.id)
    if user["is_passed"] or not user["is_passing"]:
        return
    user["question_index"] += 1
    db.set_user(query.message.chat.id, {"question_index": user["question_index"]})
    post = get_question_message(user)
    if post is not None:
        bot.edit_message_text(post["text"], query.message.chat.id, query.message.id,
reply_markup=post["keyboard"])

def get_question_message(user):
    if user["question_index"] == db.questions_count:

```

		Городецька В.В			Житомирська політехніка 22.126.8.000	Арк.
		Марчук Г.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		34

```

count = 0
for question_index, question in enumerate(db.questions.find({})):
    if question["correct"] == user["answers"][question_index]:
        count += 1
percents = round(100 * count / db.questions_count)
if percents < 40:
    smile = "😞"
elif percents < 60:
    smile = "😐"
elif percents < 90:
    smile = "😄"
else:
    smile = "😍"
text = f"Ви відповіли правильно на {percents}% питань {smile}"
db.set_user(user["chat_id"], {"is_passed": True, "is_passing": False})
return {
    "text": text,
    "keyboard": None
}

question = db.get_question(user["question_index"])
if question is None:
    return
keyboard = telebot.types.InlineKeyboardMarkup()
for answer_index, answer in enumerate(question["answers"]):
    keyboard.row(telebot.types.InlineKeyboardButton(f"{chr(answer_index +
97)) {answer}", callback_data=f"?ans&{answer_index}"))
text = f"Питання №{user['question_index'] + 1}\n\n{question['text']}"
return {
    "text": text,
    "keyboard": keyboard
}

def get_answered_message(user):
    question = db.get_question(user["question_index"])
    text = f"Питання №{user['question_index'] + 1}\n\n{question['text']}\n"

```

		Городецька В.В.			Житомирська політехніка 22.126.8.000	Арк.
		Марчук Г.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		35

```

for answer_index, answer in enumerate(question["answers"]):
    text += f" {chr(answer_index + 97)} {answer}"
    if answer_index == question["correct"]:
        text += " ✓"
    elif answer_index == user["answers"][-1]:
        text += " ✗"
    text += "\n"

keyboard = telebot.types.InlineKeyboardMarkup()
keyboard.row(telebot.types.InlineKeyboardButton("Далі", callback_data="?next"))
return {
    "text": text,
    "keyboard": keyboard
}
bot.polling()

```

		Городецька В.В			Житомирська політехніка 22.126.8.000	Арк.
		Марчук Г.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		36