

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
«ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність:

123 – «Комп'ютерна інженерія»

Освітня програма:

«Обслуговування комп'ютерних систем і мереж»

Група: 4КС-56

Дипломний проект

студента денної форми навчання
КС 56.04.000.00 ДП ПЗ

**ГАВРИЛЮКА
ОЛЕГА
СЕРГІЙОВИЧА**

м. Одеса
2023 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «Одеський технічний фаховий коледж ОНТУ»

Спеціальність 123 – «Комп'ютерна інженерія»

Освітня програма «Обслуговування комп'ютерних систем і мереж»

Група 4КС-56

ПОЯСНЮВАЛЬНА ЗАПИСКА

До дипломного проекту (роботи) на тему: Створення чат-бота для тестування контингенту здобувачів освіти відділення комп'ютерних систем

Проектний матеріал складається з пояснювальної записки на 70 сторінках та графічного матеріалу на 15 аркушах (слайдах).

Дипломник  (Гаврилюк О.С.)

Керівник проекту  (Скорнякова О.В.)

Консультанти:

з економічної частини  (Копайгородська Т.Г.)

з охорони праці  (Чорновол Н.І.)

за дотриманням вимог ЄСКД  (Петрашова В.І.)

старший консультант  (Кривченко Ю.В.)

До захисту допущений

Голова циклової комісії  (Кривченко Ю.В.)

Завідувач відділенням  (Скорнякова О.В.)

Захист « 19 » серпня 2022 р.

Протокол ДКК № 1

Оцінка ДКК 5 (відмінно)

Секретар ДКК 

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ВСП «Одеський технічний фаховий коледж ОНТУ»

Відділення комп'ютерних систем Комісія КТ і Ш

Спеціальність 123 "Комп'ютерна інженерія"

ЗАТВЕРДЖУЮ:

Заст. дир. з НВР 

Беркань І.В.

" " 20 р.

ЗАВДАННЯ

на дипломний проект (роботу)

Здобувачу освіти Гаврилюку Олегу Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Створення чат-бота для тестування контингенту здобувачів освіти відділення комп'ютерних систем

затверджена наказом по коледжу від " 17 " 10 2022 р. № 235-A2-ОД

2. Термін задачі здобувачем освіти закінченого проекту (роботи) 10.06.2023 р.

3. Вихідні дані до проекту (роботи)

Мова програмування Python, бібліотеки: aiogram, ruyongong та asyncio. Платформа месенджеру Telegram, BotFather. База даних MongoDB. Текстовий редактор VS Code від Microsoft.

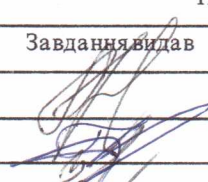
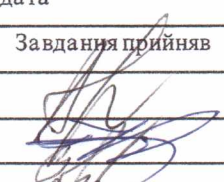
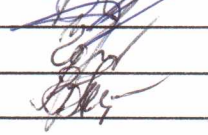
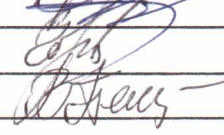
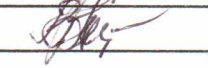
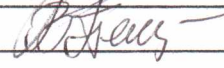


4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)

1. Технологічний розділ. 2. Економічний розділ 3. Охорона праці. Висновки. Список використаних джерел. Додатки.

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

Слайд 1 – Титульний слайд(тема, виконавець, керівник). Слайд 2 – Вступ. Слайд 3 - Розберемо детальніше: чому саме telegram? Слайд 4 – 5 - Telegram Bot API та Telegram API. Слайд 6 - Бібліотека aiogram. Слайд 7 - База даних Mongoddb. Слайд 8 - Реалізація чат-бота. Слайд 9 – Розробка програми. Слайд 10 - Бібліотеки та запуск. Слайд 11 - Обробники команд. Слайд 12 - Отримання та відправка тестів. Слайд 13 - Інтерація з користувачем. Слайд 14 - Отже, перейдемо безпосередньо до самого чат-боту у telegram. Слайд 15 – Дякую за Увагу!

6. Консультанти по проекту (роботі), із зазначенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Розділ 1-3	Скорнякова О.В.		
Економ.розділ	Копайгородська Т.Г.		
Охорона праці	Чорновол Н.І.		
Нормоконтроль	Петрашова В.І.		

7. Дата видачі завдання _____

Керівник Скорнякова О.В.


(підпис)

Завдання прийняв до виконання _____


(підпис)

КАЛЕНДАРНИЙ ПЛАН

Пор. №	Назва етапів дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1.	Огляд літератури. Огляд існуючих рішень	20.02.2023	виконав
2.	Формування кінцевого завдання на розробку. Вступ.	01.03.2023	виконав
3.	Аналітичний огляд. Огляд існуючих рішень	20.03.2023	виконав
4.	Конструкторський розділ. Вибір елементної бази	10.04.2023	виконав
5.	Розробка алгоритму та управляючої програми	17.04.2023	виконав
* 6.	Економічний розділ. Проведення розрахунків щодо економічної доцільності розробки	01.05.2023	виконав
7.	Виконання розділу «Охорона праці»	15.05.2023	виконав
8.	Виконання графічної частини дипломного проекту	22.05.2023	виконав
9.	Підготовка до попереднього захисту, підготовка до захисту	01.06.2023	виконав
10.	Підготовка доповіді та презентації для захисту	10.06.2023	виконав
11.	Отримання рецензії, відповіді на зауваження рецензента	до 10.06.2023	виконав
12.	Захист роботи	до 30.06.2023	

Дипломник _____


(підпис)

Керівник проекту _____


(підпис)

ЗМІСТ

ВСТУП	7
1 ТЕХНОЛОГІЧНИЙ РОЗДІЛ	9
1.1 Загальні відомості про чат-боти	9
1.2 Вибір засобів реалізації чат-бота	13
1.3 Реалізація чат-бота	19
1.4 Розробка програми	22
1.5 Запуск та перевірка функціоналу бота	34
2 ЕКОНОМІЧНИЙ РОЗДІЛ	38
2.1 Резюме	38
2.2 Визначення трудомісткості розробки програмного забезпечення	38
2.3 Розрахунок ціни програмного продукту	41
3 ОХОРОНА ПРАЦІ	44
3.1 Аналіз небезпечних та шкідливих чинників, що впливають на працівника	44
3.2 Розробка заходів з охорони праці	45
3.2.1 Виробниче приміщення	45
3.2.2 Мікроклімат робочої зони працівників, вентиляція	46
3.2.3 Шум	47
3.2.4 Освітлення робочого місця	47
3.2.5 Електробезпека	48
3.2.6 Організація робочого місця користувача ПК	48
3.2.7 Пожежна безпека при роботі з ВДТ	49
ВИСНОВКИ	52
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	53
ДОДАТОК А - КОД ПРОГРАМИ	55
ДОДАТОК Б - ТЕСТИ	60
ДОДАТОК В - СЛАЙДИ ПРЕЗЕНТАЦІЇ	70

ВСТУП

Месенджери з моменту їх створення проникають в наше життя дуже стрімко. Це відбувається тому, що вони дозволяють швидко отримувати та давати інформацію.

У зв'язку з цим почали розвиватися чат-боти, які дозволяють зекономити час на пошук потрібного запиту. Одним із таких видів чат-ботів являє собою TelegramBot.

Робот-помічник, готовий виконати будь-яке рутинне заняття, або спеціальний програмний код, який виконує певні команди користувача. Вся переписка з ним ведеться через звичайний чат. Ви даєте боту команди, які він готовий виконувати цілодобово. Його основне завдання відповісти на питання користувача, згідно своїй програмі. Боти допомагають, економлять багато часу і керувати ними дуже просто.

На сьогодні чат-боти Телеграм можуть:

- проводити навчання;
- розважати і грати з вами;
- працювати пошуковими системами в інтернеті;
- скачувати текстову інформацію, відео або аудіо;
- бути звичайною нагадувачем;
- брати участь у групових чатах, припустимо, для узгодження часу зустрічі, оптимальної для всіх учасників;
- коментувати потрібні статті;
- використовуватися для управління розумним будинком і ін.

Іншими словами, вони, як посередники між людиною і численними веб-службами. Їх великий плюс – це загальна оболонка, тепер всередині Телеграм знаходиться вся інформація, яку люди звикли шукати через Яндекс і Гугл. На мій погляд, безсумнівний плюс в економії часу за рахунок зменшення кількості

					КС 56. 04 000. 00 ДП ПЗ	Аркуш
						7
Зм.	Аркуш	№ докум.	Підпис	Дата		

додатків в гаджеті. Люди завжди хочуть отримувати все, що їм потрібно, в одному вікні. Тепер немає необхідності в установці зайвих додатків для погоди, новин і т. п., що було особливо критично для малопотужного смартфона. Тепер все це замінює Телеграм з можливістю установки необхідних робіт.

Тема мого дипломного проекту - створення чат-бота для тестування контингенту здобувачів освіти відділення комп'ютерних систем.

Об'єктом дипломного проекту є створення чат-боту.

Завдання, які потрібно реалізувати:

- створити асинхронного бота для месенджеру Telegram.
- під'єднати його до бази даних.
- ввести у бота тести для тестування контингенту здобувачів освіти.

У першому розділі дипломного проекту

- здійснюється аналітичний огляд існуючих рішень - чат боти, їх відомості, поширення, платформи впровадження;
- обираються: мова програмування, месенджер під який буде створено бота, база даних;
- визначаються: тести для впровадження в бота, які і будуть тестувати контингент здобувачів освіти.

У другому розділі визначаються питання економічної доцільності розробки. У третьому – питання охорони праці.

					<i>КС 56. 04 000. 00 ДП ПЗ</i>	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		8

1 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

1.1 Загальні відомості про чат-боти

Нині чат-боти набирають великої популярності завдяки широкому використанню в багатьох сферах життєдіяльності людини. Ці віртуальні агенти можуть повідомити погоду, курс валют, записати клієнта на певну дату та навіть бути альтернативою онлайн бібліотеки. Статистика ринку чат-ботів показує, що одна з причин, чому ця технологія стає все більш і більш популярною, полягає в тому, що чат-боти можуть відповісти на більшість запитань, які можуть поставити їм користувачі. Для складніших питань все ще важливо мати кваліфікованих фахівців служби підтримки клієнтів. Але для щоденних питань служба чат-ботів зменшує витрати та пришвидшує час відгуку.

На цей момент існує велика класифікація чат-ботів: по платформі впровадження, технології розробки, способу спілкування з користувачами і функціональності[1]

Але найпоширенішими залишається поділ з точки зору реалізації:

- Бізнес класифікація яка полягає у сферах застосування чат-ботів
- Технічна класифікація полягає в основі створення чат-боту.

Бізнес класифікація:

- Продажі. Бот допомагає підібрати клієнту відповідний товар, відповідає на питання, розповідає про знижки та підводить клієнта до здійснення покупки.
- Чат-бот асистент. Інтеграція з системами-аналітики, CRM і бухгалтерією дає йому можливість складати звіти, аналізувати дані, заповнювати форми та ін.
- Обслуговування клієнтів. Забронювати таксі, забронювати квитки, номер в готелі і столик в ресторані, оформити доставку продуктів - все це можна зробити за допомогою чат-бота.
- Медіа. З їх допомогою користувачі знаходяться в курсі останніх новин або вибирають і читають тільки цікаві для них розділи.

					КС 56. 04 000. 00 ДП ПЗ	Аркуш
						9
Зм.	Аркуш	№ докум.	Підпис	Дата		

– Особистий помічник. Нагадуватиме про зустрічі, підкаже прогноз погоди та навіть знайде рецепт будь-якої страви.

За технічною класифікацією чат-боти поділяються на прості, розумні з підтримкою штучного інтелекту та гібридні.

Простий бот – це бот, який відповідає на запитання на основі заздалегідь встановленого вибору інтегрованих відповідей.[2] Простих ботів також називаються ботами дерева прийняття рішень, як впливає з назви, вони використовують ряд визначених правил та нагадують текстову систему IVR (система попередньо записаних голосових повідомлень, що виконує функцію маршрутизації дзвінків всередині контакт-центру). Такі боти не роблять жодних висновків з попередніх взаємодій та найкраще підходять для прямолінійних діалогів. Простий бот зазвичай будується на базі кнопок, тому він ідеально підходить для опитування, підтримки продажів і практично для будь-якого простого завдання автоматизації процесів, де сценарії спілкування чітко визначені.

Статистика використання та залучення чат-бота:

- За словами 87.2% респондентів, споживачі мають нейтральний або позитивний досвід роботи з чат-ботами.
- Споживачі використовують чат-ботів з різних причин, найпоширенішою з яких є отримання швидкої відповіді.
- Одна третина клієнтів хочуть зробити бронювання за допомогою чат-бота.
- Сполучені Штати мають найбільше споживачів чат-ботів (36%), далі йдуть Індія (11%) і Німеччина (4%).
- Керівники рівня С складають 41% тих, хто ініціює онлайн-чат, чати з організаціями.
- Зараз клієнти зацікавлені в допоміжних інструментах штучного інтелекту (AI) на рівні 27%.

					КС 56. 04 000. 00 ДП ПЗ	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		10

- У надзвичайних ситуаціях 37% людей використовують бота служби підтримки клієнтів, щоб отримати швидку відповідь. [3]
- Майже половина жінок-покупців воліють спілкуватися за допомогою чат-ботів під час здійснення онлайн-покупок, однак лише 36.81% чоловіків це роблять.

Статистика переваг чат-бота:

- Час відгуку чат-ботів швидший, ніж у людей.
- За словами 68 відсотків респондентів, споживачам подобаються чат-боти, тому що вони швидко відповідають.
- Споживачі також віддають перевагу думці, що чат-бот може допомогти їм у неробочий час і направити їх до живої людини.
- Найважливішою особливістю чат-бота, на думку користувачів, є доступність сервісу 24 години на добу, сім днів на тиждень.
- Доведено, що чат-боти зменшують витрати на оперативну підтримку клієнтів до 30%.

Платформи впровадження чат-ботів:

Як вже було сказано раніше, чат-ботів можливо інтегрувати у будь-який веб сервіс. Але, як не дивно найбільшої популярності вони здобули завдяки месенджерам. Це все завдяки тому що, з кожним роком люди все частіше віддають перевагу смартфонам. [4]

Telegram – кросплатформний месенджер із відкритим кодом, функціями телефонного зв'язку по протоколу IP, що дозволяє обмінюватися текстовими, голосовими та відеоповідомленнями, фотографіями та файлами багатьох форматів. [5]

Станом на березень 2023 року офіційні клієнти для Telegram охоплюють: мобільні додатки для Android і iOS / iPadOS ; настільні додатки для Windows , Linux і macOS ; веб-додаток

					КС 56. 04 000. 00 ДП ПЗ	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		11

Використовуючи месенджер Телеграм користувачі отримують не тільки платформу для спілкування, але і багато інших можливостей, завдяки, яким месенджер набув великої популярності.

З основних можна перерахувати:

- спілкуватись з користувачами по всьому світу, та здійснювати аудіо та відеодзвінки
- відправляти пересилати і отримувати файли різних форматів
- створювати групові та засекречені чати
- знайти людину за псевдонімом, не знаючи номера телефону
- відправляти знайдені в пошукових системах фото без збереження в пам'яті пристрою
- користуватись хмарним сховищем даних. будь-які файли, обсяг і кількість яких не обмежені, зберігаються на віддаленому сервері.
- можливість безплатно та без інтернету слухати музику. для цього спеціально був розроблений медіа плеєр.

Месенджер має низку переваг. Серед основних:

- висока швидкість роботи - не дивлячись на розташування серверів в різних частинах планети, передача повідомлення займе частки секунди.
- автоматичне додавання списку користувачів сервісу з телефонної книги.
- простий початок роботи. для реєстрації потрібен тільки номер телефону.
- всі функції та можливості безплатні та без реклами.
- інноваційність – сервіс постійно оновлюється. його можливості розширюються як з урахуванням пропозицій конкурентів, так і власними розробками, які не мають аналогів.
- високий рівень безпеки і захисту від злому.
- необмеженість: кількість відправлених в день повідомлень, обсяги завантажуваних в хмару файлів, часу використання.

					КС 56. 04 000. 00 ДП ПЗ	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		12

- сумісність з усіма основними операційними системами, включаючи мобільні.
- конфіденційність. усі чати, які існують у месенджері – зашифровані. останній пункт можна віднести одночасно як до переваг, так і до недоліків.

Важливо розуміти, що всі повідомлення в Telegram завжди надійно зашифровані. Різниця між секретним і звичайним чатом тільки в методі шифрування: клієнт-клієнт для секретних чатів, клієнт-сервер / сервер-клієнт для звичайних. Завдяки цьому, користувачі мають доступ до свого листування в звичайних чатах з будь-якого пристрою без втрати приватності.

У секретних чатах це не працює – він прив’язаний до конкретного пристрою. Вся історія повідомлень зберігається на одному пристрої і доступна тільки на ньому.

1.2 Вибір засобів реалізації чат-бота

Оскільки чат-боти набрали вже досить великої популярності, це призвело до того, що з’явилося безліч рішень для їх створення. Створити чат-бота можна двома способами: за допомогою мови програмування; за допомогою конструктора на сервісі.

Загалом чат-бот, написаний певною мовою програмування, являється серверним додатком, в якому функції чату працюють через власний API. Для того, щоб створити такого бота необхідна певна інфраструктура: хостинг, сервер (фізичний або хмарний) та база даних. Можливості таких чат-ботів обмежуються лише можливостями платформи, на яку вони інтегруються.

Натомість чат-бот, який створюється власноруч, за допомогою конструктора, обмежується особливостям сервісу на якому він створюється.

Перед тим як створити чат-бота необхідно зрозуміти як саме відбувається взаємодія між користувачем та сервером. [6]

					КС 56. 04 000. 00 ДП ПЗ	Аркуш
						13
Зм.	Аркуш	№ докум.	Підпис	Дата		

Повідомлення, команди і запити, надіслані користувачами, передаються на програмне забезпечення, запущене на серверах розробників.

Сервер Telegram, який являється посередницьким та анонімним, обробляє шифрування і здійснює зворотний зв'язок між користувачем і утилітою.

Взаємодія між користувачем і ботом виглядає наступним чином:

1. Користувач надсилає боту команду
2. Бот передає команду на сервер
3. Програма на сервері оброблює отриманий від бота запит
4. Сервер віддає відповідь боту
5. Бот виводить відповідь користувачеві на вікні керування.

І цей цикл повторюється раз за разом під час взаємодії з будь-яким телеграм-ботом. Спілкування відбувається за допомогою простого HTTPS-інтерфейсу, який є спрощеною версією API Telegram. Інакше цей інтерфейс можна назвати програмним каталогом або бот-алгоритмом. Схема обміну даними між користувачем і Telegram ботом наведена на рис. 1.1.

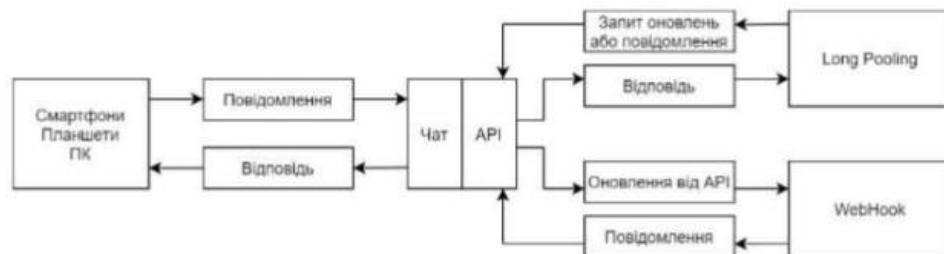


Рисунок 1.1 - Схема обміну даними між користувачем і Telegram ботом

Ось деякі функції, які Телеграм надає для ботів:

1. Відправка повідомлень: Боти можуть надсилати текстові повідомлення, фотографії, відео, аудіо, документи та інші типи файлів користувачам.
2. Клавіатури: Боти можуть надсилати клавіатури з кнопками для швидкого вибору відповідей або дій. Це може бути звичайна клавіатура або онлайн-клавіатура, яка відображається безпосередньо у чаті.

3. Команди: боти можуть реагувати на команди, які користувачі вводять у чаті. Команди починаються зі знака "/", наприклад, "/start" або "/help".
4. Запити зі сторонніх сервісів: боти можуть взаємодіяти зі сторонніми сервісами, такими як дані, API, або інші веб-сервіси, щоб отримувати та надсилати інформацію.
5. Обробка природної мови (NLP): Боти можуть використовувати сервіси NLP для розуміння запитів користувачів та формування відповідей.
6. Вебхуки: Вебхуки дозволяють ботам отримувати оновлення про нові повідомлення та інші події в режимі реального часу, без необхідності постійного опитування сервера Телеграм.
7. Інтеграція з іншими платформами: Боти можуть бути інтегровані з іншими платформами, такими як системи управління навчанням (LMS), CRM, або інші веб-сервіси.

В залежності від необхідної функціональності, створення чат-бота в Telegram досить просто виконується за допомогою використання однієї з серверних мов програмувань: Ruby, Python, PHP, Node.JS.

Серед цих мов, що дозволяють швидко і безпроблемно виконати створення бота, Python є одним з найбільш популярних рішень. В основному це пов'язано з широкими можливостями, доступними як при використанні стандартних бібліотек, так і з застосуванням вже готових варіантів, таких як TelegramBotAPI, розрахованих на роботу безпосередньо з Telegram.

Python - це мова програмування яка допомагає підвищити продуктивність розробника, а не коду, який він пише. Шляхом простоти коду, подальший супровід програм, написаних на Python, стає легше і приємніше в порівнянні з Java або C++. Ця мова програмування являється універсальною завдяки багатій 24 стандартній бібліотеці, тому широко використовується у різних областях: веб-розробці, машинному навчанні, розробці ігор, комп'ютерній безпеці тощо. [7]

Інтерпретатор Python практично був реалізований на всіх платформах і операційних системах, що є безперечною перевагою. Першою такою мовою була

					<i>КС 56. 04 000. 00 ДП ПЗ</i>	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		15

C, однак типи даних цієї мови на різних машинах займають різну кількість пам'яті, а це представляє деяку перешкоду при написанні великих програм. Також, велике значення надається розширюваності мови. Як казав сам розробник Python, кожен програміст може вдосконалювати цю мову й ділитися своїми напрацюваннями з іншими. Інтерпретатор був розроблений мовою програмування C і вихідний код доступний для будь-яких змін. Його з легкістю можна додати у свою програму і використовувати як вбудовану оболонку.

Visual Studio Code, який також зазвичай називають VS Code - це кросплатформенний редактор вихідного коду, створений Microsoft із Electron Framework для Windows, Linux і macOS. Функції включають підтримку налагодження, підсвічування синтаксису, інтелектуальне завершення коду, фрагменти, рефакторинг коду та вбудований Git. Користувачі можуть змінювати тему, комбінації клавіш, параметри та встановлювати розширення, які додають функціональність [8].

Telegram Bot API та Telegram API

Усе почалося з того, що Микола Дуров (власник і творець Telegram) спільно з командою програмістів створив криптографічний протокол. Його рушій задіяв комбінацію симетричного шифрування AES, протокол Діффі-Хеллмана для обміну ключами шифрування між клієнтами і низку хеш-функцій. На основі цього протоколу було побудовано MTProto - механізм, що дозволяє користувачам сьогодні використовувати Telegram-месенджери. [9]

Наразі є два основних інструменти API, за допомогою яких можна задіяти сервіси Telegram - Telegram Bot API і Telegram API. Перший служить для розробки чат-ботів, другий дає змогу робити повністю кастомні Telegram-клієнти. Розробникам також доступна відкрита бібліотека TDLib (Telegram Database Library), за допомогою якої можна створювати свою версію месенжера з унікальними опціями (як наприклад, Telegram X, побудований саме на TDLib). Telegram Bot API є надбудовою над Telegram API, тому користуватися Bot API можна без знань про механізм використовуваного протоколу MTProto.

					КС 56. 04 000. 00 ДП ПЗ	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		16

Для його роботи задіяний проміжний сервер з HTTPS-інтерфейсом, який шифрує трафік і забезпечує зв'язок з Telegram API. Bot API дозволяє легко створювати програми, які використовують інтерфейс Telegram для виконання коду на локальному сервері. Користувачі можуть взаємодіяти з ботами, надсилаючи їм повідомлення, команди та вбудовані запити.

Принцип роботи будь-якого бота полягає в тому, що він перманентно надсилає запити на сервер і регулярно отримує оновлення. Отримувати їх можна двома способами. По-перше, можна використовувати вебхуки, коли сервер робить зворотний виклик на вказаний URL. А по-друге, можна просто "закидати" запитами Telegram, отримуючи постійні відповіді. Отже, Bot API - це інтерфейс на основі HTTP, створений для розробників, захоплених створенням ботів для Telegram.

Бібліотека aiogram

Бібліотека aiogram - це високорівнева бібліотека для розробки ботів для месенджера Telegram з використанням мови програмування Python. Вона надає зручний і простий інтерфейс для взаємодії з Telegram Bot API та дозволяє легко створювати потужні та функціональні боти.

Отримуючи доступ до Telegram Bot API, aiogram дозволяє виконувати такі дії, як отримання оновлень з чатів, відправка повідомлень, керування клавіатурами та іншими елементами інтерфейсу, робота з мультимедійними файлами, робота з клавішами швидкого доступу та інше. [10]

Бібліотека aiogram надає розширені можливості для обробки різних типів повідомлень та подій, включаючи текстові повідомлення, зображення, відео, аудіо, документи, геолокацію, контакти, статуси набору тексту і багато іншого. Вона також підтримує роботу з клавіатурами, меню, інлайн-кнопками та іншими елементами інтерфейсу для забезпечення зручності користувачів.

aiogram пропонує асинхронну модель програмування з використанням `async/await`, що дозволяє зручно виконувати асинхронні запити до Telegram Bot API та обробляти одночасно багато взаємодій з користувачами.

					КС 56. 04 000. 00 ДП ПЗ	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		17

Загалом, бібліотека aiogram є потужним інструментом для розробки ботів для Telegram з використанням Python, що дозволяє зосередитися на логіці бота, спрощує взаємодію з Telegram Bot API та надає широкі можливості для створення різноманітних функцій та взаємодії з користувачами.

База даних MongoDB

MongoDB підтримує зберігання документів в JSON-подібному форматі, має досить гнучку мову для формування запитів, може створювати індекси для різних збережених атрибутів, ефективно забезпечує зберігання великих бінарних об'єктів, підтримує журналювання операцій зі зміни і додавання даних в БД, може працювати відповідно до парадигми Map/Reduce, підтримує реплікацію і побудову відмовостійких конфігурацій. У MongoDB є вбудовані засоби із забезпечення шардінгу (розподіл набору даних по серверах на основі певного ключа), комбінуючи який з реплікацією даних можна побудувати горизонтально масштабований кластер зберігання, в якому відсутня єдина точка відмови (збій будь-якого вузла не позначається на роботі БД), підтримується автоматичне відновлення після збою і перенесення навантаження з вузла, який вийшов з ладу. Розширення кластера або перетворення одного сервера на кластер проводиться без зупинки роботи БД простим додаванням нових машин. [11]

Отже, стає зрозумілим, що Telegram бот, має власний інтерфейс взаємодії Bot API, який створений на основі HTTP протоколу. Також, що HTTP – це протокол прикладного рівня, по якому відбувається обмін даними. У чат-боті дані передаються у форматі JSON. Для взаємодії з Bot API можна використовувати будь-яку мову програмування, але найзручнішою виявилась Python, завдяки простоті написання коду та широкому вибору бібліотек, які розраховані на роботу безпосередньо з Telegram. Та для збереження результатів тестів, інформації про абітурієнта який проходив тест, буде використовуватися база даних MongoDB.

					КС 56. 04 000. 00 ДП ПЗ	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		18

1.3 Реалізація чат-бота

Для того щоб почати розробку чат-боту потрібно безпосередньо створити цього самого бота у додатку Телеграм. Для того щоб створити бота у Телеграмі потрібно звернутися до ще одного бота який називається BotFather. [12]

Знаходимо його у самому додатку Телеграму по запиту – BotFather. Натискаємо кнопку старт.

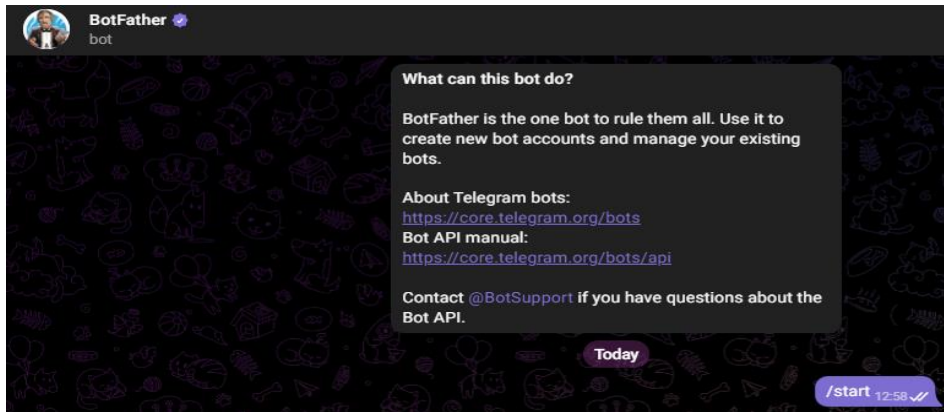


Рисунок 1.2 - Телеграмбот – BotFather

Дивимось, які функції та команди нам надає цей бот. Мені потрібна частина зі створенням бота та його перше налаштування таке як: Ім'я, опис, фото.

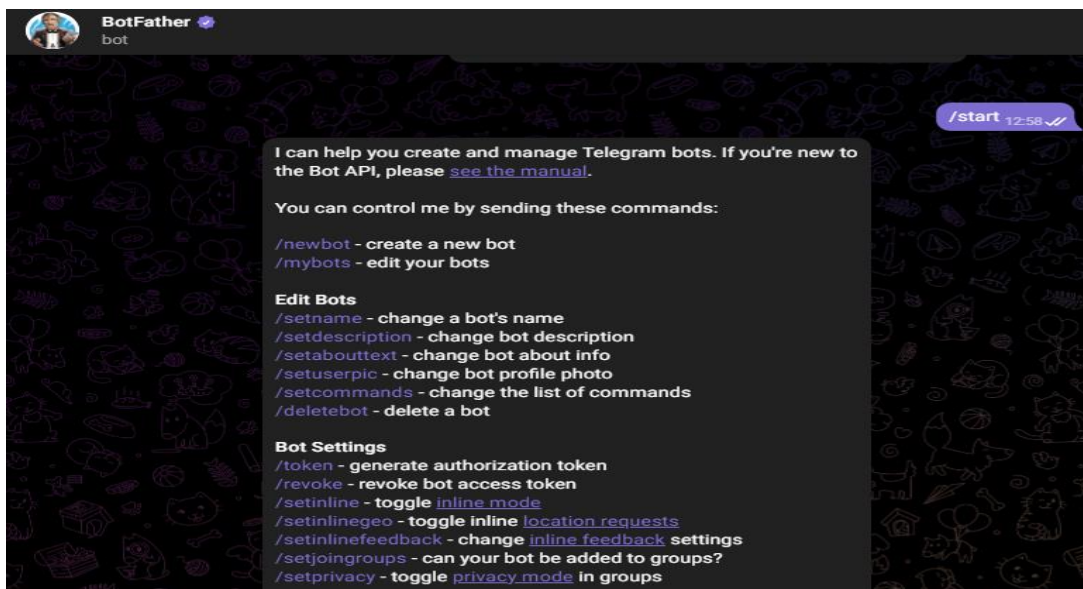


Рисунок 1.3 - Налаштування бота

Вводимо потрібні команди:

`/newbot` – створити нового бота;

При створенні бота BotFather запропонує вигадати йому ім'я, я назву його

					КС 56. 04 000. 00 ДП ПЗ	Аркуш
						19
Зм.	Аркуш	№ докум.	Підпис	Дата		

TestBot.

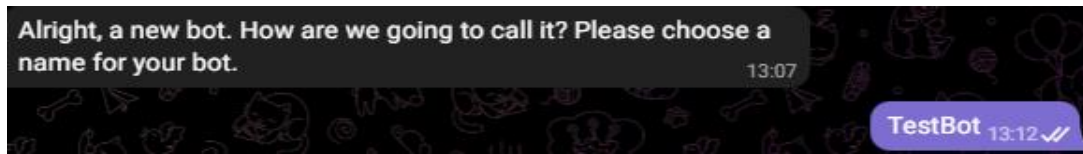


Рисунок 1.4 - Ім'я бота

Після вводу ім'я, BotFather запропонує створити Ім'я для посилання на бота, яке потрібно буде вводити щоб знайти його у Телеграмі. Я назву його OTFK_TestBot. Важливо щоб приписка Bot була в кінці.

Після цього BotFather створить бота, та надасть мені його токен. Токен - це ключ, який складається з набору символів, та використовується для зв'язку webhook-у та серверу месенджера telegram. Для безпеки захисту бота на рисунку я його замажу.

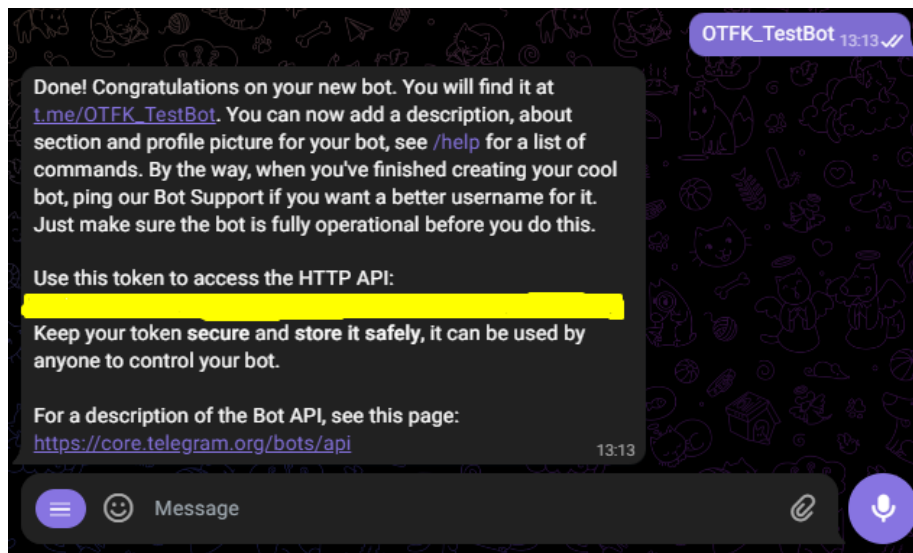


Рисунок 1.5 - Токен бота

Тепер нам потрібно налаштувати інформацію про бота, додати опис що може робити цей бот та поставити аватар.

Щоб додати інформацію про бота, яка буде з'являтися у вікні до запуску бота, потрібна команда /setdescription. Напишу, що Бот створений для тестування контингенту здобувачів освіти відділення комп'ютерних систем ВСП ОТФК ОНТУ.

					КС 56. 04 000. 00 ДП ПЗ	Аркуш
						20
Зм.	Аркуш	№ докум.	Підпис	Дата		

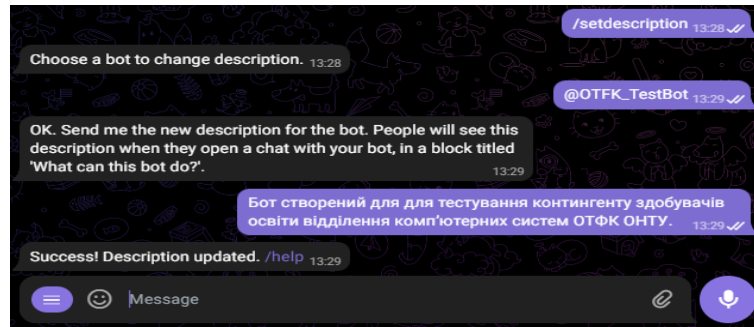


Рисунок 1.6 - Опис бота

Потім, щоб додати інформацію про бота в графу «Інформація про бота», вводимо команду /setabouttext. Напишу, що це бот студента коледжу створений для допомоги по проведенню тестування здобувачів коледжу.

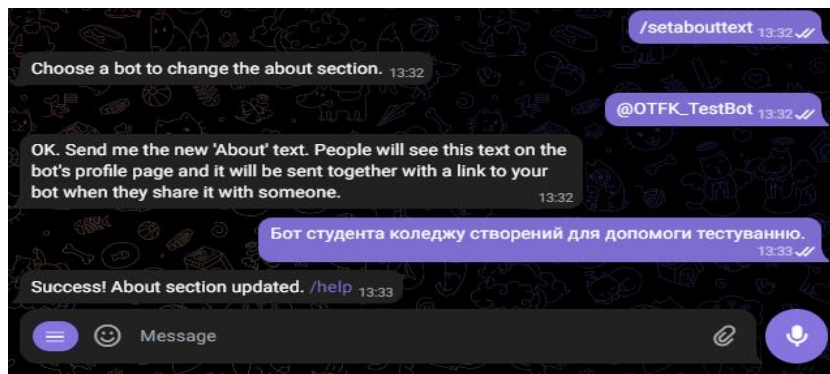


Рисунок 1.7 - Графа «Інформація про бота»

Залишилось тільки додати аватар боту, для цього пропишемо команду /setuserpic. Виберу фото в Інтернеті яке мені до вподоби.

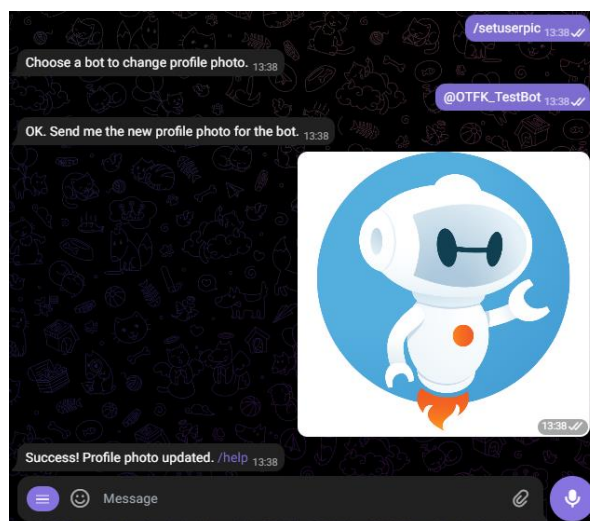


Рисунок 1.8 - Аватар бота

					<i>КС 56. 04 000. 00 ДП ПЗ</i>	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		21

Тепер налаштувавши бота перейдемо безпосередньо до самого бота та подивимось, чи застосувались всі налаштування.

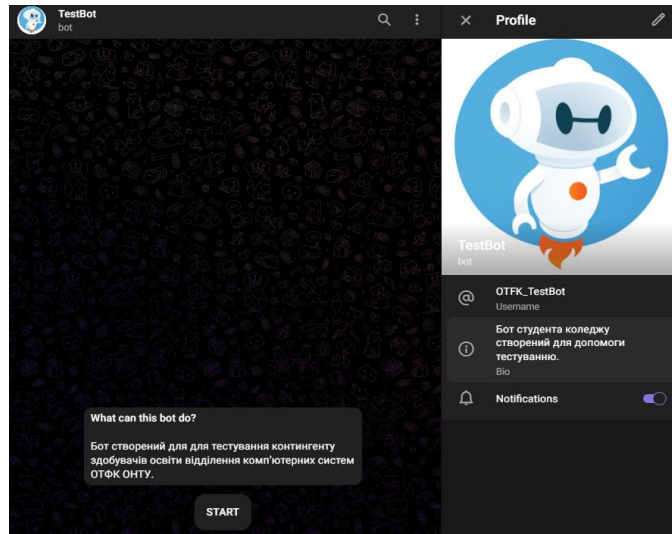


Рисунок 1.9 - Перевірка готового бота створеного BotFather

1.4 Розробка програми

Після формування кінцевого завдання на розробку та створення бота у Telegram перейдемо до реалізації системи. Для розробки був обраний текстовий редактор VS Code від Microsoft. Для початку встановимо мову програмування Python, перейдемо до офіційного сайту python.org/downloads, та завантажимо файл встановлення. Після цього згідно інструкції встановимо мову програмування Python. Створимо папку проєкта TelegramBot. Через VS Code відкриваємо цю папку, та додаємо головний python файл проєкту `testbot.py`.

Тепер нам необхідно встановити всі необхідні залежності, бібліотеки які ми будемо використовувати. Переходимо до вбудованого терміналу у Vs Code, та через команду пакетного менеджера `pip install` встановлюємо всі бібліотеки. `pip install aiogram, pymongo`.

Для початку імпортуємо до нашої програми всі необхідні бібліотеки з якими ми будемо працювати:

					КС 56. 04 000. 00 ДП ПЗ	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		22

```

1  from aiogram import Bot, types
2  from aiogram.dispatcher import Dispatcher
3
4  from aiogram.utils.exceptions import MessageCantBeEdited, MessageNotModified
5
6  from pymongo.collation import Collation
7
8  from scripts.settings import API_TOKEN, DATABASE, TESTS
9
10 from typing import Union
11 import asyncio
12

```

Рисунок 1.10 - Бібліотеки та імпорт з них

Створимо підготовчі кроки перед запуском Telegram бота.

```

14  # Отримання поточного циклу подій
15  loop = asyncio.get_event_loop()
16
17  # Створення екземплярів Bot та Dispatcher
18  bot = Bot(token=API_TOKEN, loop=loop)
19  dispatcher = Dispatcher(bot=bot)
20
21  # Отримання колекції користувачів з бази даних
22  users: Collation = DATABASE.Users
23

```

Рисунок 1.11 - Підготовчі кроки

Ці частини коду відповідають за деякі підготовчі кроки перед запуском Telegram бота:

`loop = asyncio.get_event_loop()`: Отримання поточного циклу подій (event loop). Це потрібно для асинхронної обробки подій та виконання асинхронних функцій у програмі.

Створення екземплярів Bot та Dispatcher: `bot = Bot(token=API_TOKEN, loop=loop)` створює об'єкт Bot з використанням токена API_TOKEN і прив'язує його до поточного циклу подій loop. `dispatcher = Dispatcher(bot=bot)` створює об'єкт Dispatcher і прив'язує його до створеного об'єкта Bot. Dispatcher відповідає за обробку повідомлень та подій бота.

`users: Collation = DATABASE.Users`: Отримання колекції користувачів з бази даних. Здається, що DATABASE є об'єктом бази даних, а Users - колекцією

Основна функція цього обробника полягає в обробці повідомлень, які користувачі відправляють боту. Нижче наведений розбір кожного кроку в кодї:

`user: dict = users.find_one({'_id': message.from_user.id})` - отримання інформації про користувача з бази даних. В цьому випадку, використовується метод `find_one` для пошуку користувача за `_id`, який взятий з властивості `from_user.id` об'єкта `message`. Отримані дані про користувача зберігаються у змінній `user`.

Перевіряється наявність значення `name` у `user`:

Якщо `user['name']` є `None`, то це означає, що користувач ще не ввів своє ім'я. В цьому випадку:

Виконується оновлення запису про користувача в базі даних з використанням методу `update_one`. Властивість `name` встановлюється на значення `message.text`, яке відповідає тексту повідомлення, яке користувач надіслав боту.

Відправляється повідомлення боту до чату з використанням методу `bot.send_message`. В тексті повідомлення вказується користувачевий чат `chat_id` та текст повідомлення 'Введіть свою фамілію:'.

Якщо `user['name']` не є `None`, переходимо до наступного кроку.

Перевіряється наявність значення `surname` у `user`:

Якщо `user['surname']` є `None`, це означає, що користувач вже ввів своє ім'я, але ще не ввів своє прізвище. В цьому випадку - виконується оновлення запису про користувача в базі даних, аналогічно попередньому кроці, проте властивість `surname` встановлюється на значення `message.text`.

Викликається функція `send_tests`, якій передається об'єкт `message` в якості параметра.

Якщо `user['surname']` не є `None`, переходимо до наступного кроку.

Викликається функція `send_tests`, якій передається об'єкт `message` в якості параметра. Цей крок виконується, коли користувач вже ввів як своє ім'я, так і прізвище.

					<i>КС 56. 04 000. 00 ДП ПЗ</i>	Аркуш
						26
Зм.	Аркуш	№ докум.	Підпис	Дата		

Наступним кроком буде відправка списку тестів до користувача.

```
37 # Відправка списку тестів
38 async def send_tests(message: types.Message) -> None:
39     keyboard = types.InlineKeyboardMarkup()
40
41     for test in TESTS:
42         keyboard.add(types.InlineKeyboardButton(text=test['name'], callback_data=test['id']))
43
44     try:
45         await message.edit_text(text='Виберіть тест:', reply_markup=keyboard)
46     except MessageCantBeEdited:
47         await bot.send_message(chat_id=message.chat.id, text='Виберіть тест:', reply_markup=keyboard)
48
```

Рисунок 1.18 - Відправка список тестів

Розглянемо кожен крок в коді:

Створення об'єкту keyboard типу InlineKeyboardMarkup. Цей об'єкт використовується для створення клавіатури з інлайн-кнопками, які користувач може натиснути.

Цикл for test in TESTS перебирає список тестів TESTS і додає для кожного тесту інлайн-кнопку до клавіатури. Кожна кнопка має текст, який визначається значенням ключа 'name' у словнику test, а також callback_data, яке визначається значенням ключа 'id' у словнику test. Ці значення можуть використовуватися для ідентифікації тесту, коли користувач натисне на кнопку.

Блок try-ехсепт виконує спробу змінити текст повідомлення message на 'Виберіть тест:' та встановити клавіатуру keyboard з використанням методу edit_text. Якщо повідомлення не може бути відредаговане (наприклад, якщо воно не було створене раніше), виникає виняток MessageCantBeEdited, і виконується блок ехсепт. У блоку ехсепт викликається функція bot.send_message, яка відправляє нове повідомлення з текстом 'Виберіть тест:' та клавіатурою keyboard.

```
63 # Отримання тесту за його ідентифікатором
64 async def get_test(test_id: int) -> Union[dict, None]:
65     for test in TESTS:
66         if test['id'] == test_id:
67             return test
68     return None
```

Рисунок 1.19 - Отримання тесту за ідентифікатором

Ця частина коду виконує функцію отримання тесту за його ідентифікатором. Розглянемо кожен крок в коді:

Функція `get_test` приймає параметр `test_id`, який представляє ідентифікатор тесту.

Цикл `for test in TESTS` перебирає список тестів `TESTS`.

Умовний оператор `if test['id'] == test_id` порівнює значення ключа `'id'` в словнику `test` з переданим `test_id`.

Якщо знайдений тест має ідентифікатор, що співпадає з `test_id`, то повертається сам тест, використовуючи оператор `return test`. Таким чином, функція повертає словник, який представляє знайдений тест.

Якщо жодного тесту зі співпадаючим ідентифікатором не знайдено, виконується оператор `return None`. Це означає, що функція поверне `None`, що може вказувати на те, що тест з таким ідентифікатором не існує або не був знайдений.

```
92 # Обробник запитів зворотнього виклику
93 @dispatcher.callback_query_handler()
94 async def callback_queries(callback_query: types.CallbackQuery) -> None:
95     test_data = callback_query.data.split(':')
96     test = await get_test(test_id=int(test_data[0]))
97
```

Рисунок 1.20 - Обробник запитів

Дана частина коду відповідає за обробку запитів зворотного виклику, які виникають при натисканні на інлайн-кнопки у клавіатурі.

Функція `callback_queries` є обробником запитів зворотного виклику. Вона приймає параметр `callback_query`, який є об'єктом типу `types.CallbackQuery` і представляє зворотний виклик.

Змінна `test_data` створюється шляхом розбиття рядка `callback_query.data` за допомогою методу `split(':')`. Рядок `callback_query.data` містить дані, які були встановлені у властивості `callback_data` кнопки. У даному випадку, значення розділяються знаком `':'`. Як результат, змінна `test_data` міститиме список значень.

Викликається функція `get_test` з параметром `test_id`, який береться з першого елемента списку `test_data` і перетворюється у ціле число за допомогою `int(test_data[0])`. Ця функція використовується для отримання тесту зі списку тестів за його ідентифікатором. Знайдений тест зберігається у змінній `test`.

Отже, цей код використовується для обробки запитів зворотного виклику, які виникають при натисканні на інлайн-кнопки. Він отримує ідентифікатор тесту з `callback_query.data`, викликає функцію `get_test` для отримання знайденого тесту і зберігає його у змінній `test`. Далі можуть бути виконані подальші дії з отриманим тестом.

```
70 # Відправлення питання тесту і кнопок для відповіді
71 async def send_question(callback_query: types.CallbackQuery, test: dict, test_question_id: int) -> None:
72     keyboard = types.InlineKeyboardMarkup()
73
74     if 'buttons' in test:
75         for test_button_id, test_button in test['buttons'].items():
76             keyboard.add(
77                 types.InlineKeyboardButton(text=test_button['text'], callback_data=f"{test['id']}:{test_question_id}:{test_button_id}")
78             )
79
80         await callback_query.message.edit_text(text=test['questions'][test_question_id], reply_markup=keyboard)
81     else:
82         for test_button_id, test_button in test['questions'][test_question_id]['buttons'].items():
83             keyboard.add(
84                 types.InlineKeyboardButton(text=test_button['text'], callback_data=f"{test['id']}:{test_question_id}:{test_button_id}")
85             )
86
87     try:
88         await callback_query.message.edit_text(text=test['questions'][test_question_id]['question'], reply_markup=keyboard)
89     except MessageNotModified:
90         await callback_query.message.edit_text(text=test['questions'][test_question_id]['question'], reply_markup=keyboard)
91
```

Рисунок 1.21 - Питання та кнопки для відповіді

Далі розглянемо частину коду яка відправляє питання з вибраного тесту та кнопки для відповіді:

Функція `send_question` отримує параметри:

`callback_query`: об'єкт типу `types.CallbackQuery`, який представляє зворотний виклик від кнопки.

`test`: словник, який представляє тест, з якого буде взяте питання та кнопки для відповіді.

`test_question_id`: індекс питання у списку питань тесту.

Створюється об'єкт `keyboard` типу `InlineKeyboardMarkup`, який використовується для створення клавіатури з інлайн-кнопками для вибору відповідей.

Умовний оператор `if 'buttons' in test` перевіряє наявність ключа `'buttons'` у словнику `test`. Якщо ключ `'buttons'` присутній, то виконується код у першому блоку `if`. Якщо ключ `'buttons'` відсутній, виконується код у блоку `else`.

У першому блоку `if` перебираються кнопки в `test['buttons']` за допомогою циклу `for test_button_id, test_button in test['buttons'].items()`. Для кожної кнопки створюється інлайн-кнопка з текстом `test_button['text']` і `callback_data` у вигляді рядка `"{test['id']}:{test_question_id}:{test_button_id}"`. Ці значення можуть використовуватися для ідентифікації вибраної відповіді.

Потім з використанням методу `edit_text` виконується спроба змінити текст повідомлення `callback_query.message` на питання з `test['questions'][test_question_id]` та встановити клавіатуру `keyboard`. Якщо текст повідомлення не може бути відредагований (наприклад, якщо воно не було створене раніше), виникає виняток `MessageNotModified`, і виконується блок `except`. У блоку `except` викликається функція `edit_text`, яка відправляє нове повідомлення з питання `'test['questions'][test_question_id]` та клавіатурою `keyboard`.

У блоку `else` перебираються кнопки в `test['questions'][test_question_id]['buttons']` за допомогою циклу `for test_button_id, test_button in test['questions'][test_question_id]['buttons'].items()`. Для кожної кнопки створюється інлайн-кнопка з текстом `test_button['text']` і `callback_data` у вигляді рядка `"{test['id']}:{test_question_id}:{test_button_id}"`.

Знову виконується спроба змінити текст повідомлення `callback_query.message` на питання з `test['questions'][test_question_id]['question']` та встановити клавіатуру `keyboard`. Якщо текст повідомлення не може бути відредагований, виникає виняток `MessageNotModified`, і виконується блок `except`, який відправляє нове повідомлення з питанням `test['questions'][test_question_id]['question']` та клавіатурою `keyboard`.

Отже, цей код відправляє користувачеві питання з тесту та кнопки для відповіді. Залежно від наявності кнопок в `test`, відповідний блок коду буде

					<i>КС 56. 04 000. 00 ДП ПЗ</i>	Аркуш
						30
Зм.	Аркуш	№ докум.	Підпис	Дата		

виконано для створення кнопок та відправлення питання разом з клавіатурою. Далі я поділю частину коду яка відповідає за обробку різних сценаріїв, які виникають після отримання тесту в результаті запиту зворотного виклику.

```
98     # Якщо тест існує, то виконуємо код нижче
99     if test is not None:
100         if len(test_data) == 1:
101             keyboard = types.InlineKeyboardMarkup()
102             keyboard.add(
103                 types.InlineKeyboardButton(text='Почати', callback_data=f"{test['id']}:start"),
104                 types.InlineKeyboardButton(text='Назад', callback_data='0')
105             )
106
107             await callback_query.message.edit_text(
108                 text=f"*Назва*: {test['name']}\n*Опис*: {test['description']}",
109                 parse_mode='Markdown',
110                 reply_markup=keyboard
111             )
```

Рисунок 1.22 - Обробка запиту зворотного виклику, якщо тест існує

Ця частина коду відповідає за обробку запиту зворотного виклику, якщо тест існує і натиснута кнопка для початку тесту.

Перевіряється, чи тест існує, шляхом перевірки, чи змінна `test` не є `None`.

У блоку `if len(test_data) == 1`, перевіряється довжина списку `test_data`. Якщо довжина дорівнює 1, це означає, що було натиснуто кнопку "Почати" для початку тесту.

Виконується наступний код, який встановлює клавіатуру з кнопками "Почати" та "Назад" і оновлює повідомлення з назвою та описом тесту. Клавіатура створюється за допомогою `types.InlineKeyboardMarkup()`. Використовуються кнопки типу `types.InlineKeyboardButton` з текстом "Почати" та "Назад" та відповідними значеннями `callback_data`.

Повідомлення оновлюється за допомогою `callback_query.message.edit_text()`. Встановлюється текст повідомлення, який містить назву та опис тесту з використанням форматування `Markdown`. Клавіатура також додається до повідомлення за допомогою параметру `reply_markup`.

```

112 elif len(test_data) == 2 and test_data[1] == 'start':
113     await send_question(callback_query=callback_query, test=test, test_question_id=1)
114 elif len(test_data) == 3:
115     test_question_id = int(test_data[1])
116     test_button_id = int(test_data[2])
117
118     users.update_one(
119         {
120             '_id': callback_query.from_user.id,
121         },
122         {
123             '$set': {
124                 f'tests.test_{test[id]}.{test_question_id}': {
125                     'question': test['questions'][test_question_id] if 'buttons' in test else test['questions'][test_question_id]['question'],
126                     'answer': test['buttons'][test_button_id]['text'] if 'buttons' in test else test['questions'][test_question_id]['buttons'][test_button_id]['text'],
127                     'score': test['buttons'][test_button_id]['score'] if 'buttons' in test else test['questions'][test_question_id]['buttons'][test_button_id]['score'],
128                 },
129             },
130         },
131     )

```

Рисунок 1.23 - Обробку запиту зворотного виклику, коли вибрана відповідь на питання тесту

Ця частина коду відповідає за обробку запиту зворотного виклику, коли вибрана відповідь на питання тесту.

Умова `elif len(test_data) == 3` перевіряє, чи довжина списку `test_data` дорівнює 3. Це означає, що було вибрано відповідь на питання тесту.

Значення `test_question_id` та `test_button_id` отримуються з елементів списку `test_data`.

Виконується оновлення інформації про користувача в базі даних. За допомогою `users.update_one()`, виконується оновлення документа з використанням операції `$set`. Встановлюються наступні значення:

`f'tests.test_{test[id]}.{test_question_id}'` - шлях до полів в документі, які представляють питання та відповідь на тест.

`'question', 'answer', 'score'` - поля, які оновлюються залежно від наявності кнопок в тесті. Якщо кнопки присутні, використовуються відповідні значення з `test['buttons']`, інакше використовуються значення з `test['questions']`.

Таким чином, інформація про відповідь на питання тесту зберігається в базі даних для подальшого аналізу та обробки.

Останні дві частини коду відповідають за відправку наступного питання тесту.

```

133     try:
134         await send_question(callback_query=callback_query, test=test, test_question_id=test_question_id + 1)
135     except KeyError:
136         keyboard = types.InlineKeyboardMarkup()
137         keyboard.add(
138             types.InlineKeyboardButton(text='Назад', callback_data='0')
139         )
140
141         user = dict = users.find_one({'_id': callback_query.from_user.id})
142         test_questions = user['tests'][f'test_{test['id']}']
143
144         test_total_score = 0
145
146         for test_question_id in test_questions:
147             test_total_score += test_questions[test_question_id]['score']
148
149         for test_score_info in test['scores_info']:
150             min_score, max_score = test_score_info.split('-')
151
152             if int(min_score) <= test_total_score <= int(max_score):
153                 test_total_score_info = test['scores_info'][test_score_info]
154                 break
155
156         await callback_query.message.edit_text(text=f"""\
157 Дякую за те, що пройшли тест!
158
159 Ваш бал: {test_total_score}
160 {test_total_score_info}
161 """).replace(' ', ''),
162         reply_markup=keyboard
163     )
164 else:
165     await send_tests(callback_query.message)
166

```

Рисунок 1.24 - Обробку подальших питань тесту та відображення результатів після завершення тесту

Ця частина коду відповідає за обробку подальших питань тесту та відображення результатів після завершення тесту. В блоку try викликається функція `send_question()`, передаючи їй параметри `callback_query`, `test` та `test_question_id + 1`. Це означає, що відправляється наступне питання тесту для користувача. Якщо виникає помилка `KeyError`, означає, що всі питання тесту вже були відповідані.

У блоку `except KeyError`, створюється клавіатура з однією кнопкою "Назад" та обчислюється оцінка за тест. Для цього отримується інформація про відповіді користувача з бази даних, розраховується загальний бал тесту та знаходиться відповідна інформація про оцінку зі списку `test['scores_info']`.

Повідомлення оновлюється за допомогою `callback_query.message.edit_text()`. Встановлюється текст повідомлення, який містить інформацію про завершення тесту, включаючи загальний бал та оцінку. Клавіатура з кнопкою "Назад" також додається до повідомлення. Якщо тест не існує, викликається функція `send_tests()`, передаючи їй параметр `callback_query.message`. Це означає, що буде відправлено список тестів користувачу.

1.5 Запуск та перевірка функціоналу бота

Запускаємо бота у редакторі VS Code, запускаємо базу даних на локальному хості, та переходимо в телеграм. Для початку, напишемо команду /start. Бот привітався, звернувшись до мене через мій псевдонім, та запросив у мене ім'я. Після того як я ввів своє ім'я, бот запросив моє прізвище.

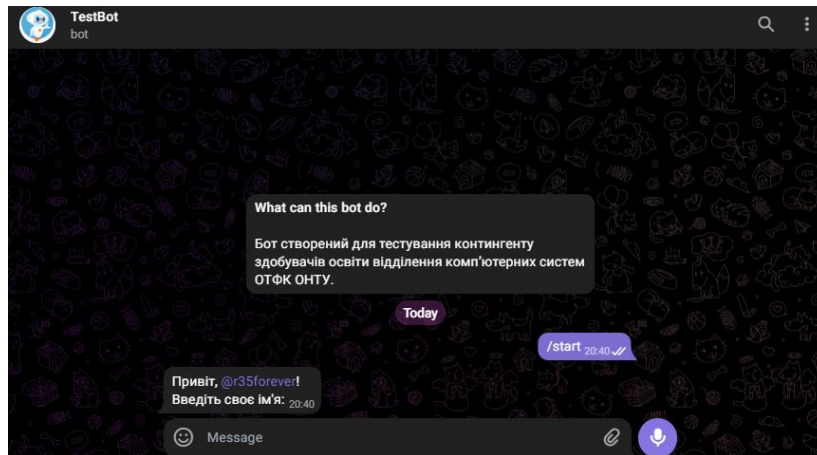


Рисунок 1.25 - Запуск бота

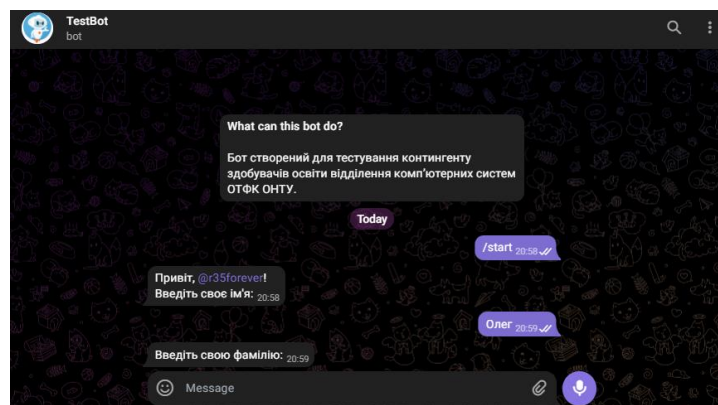


Рисунок 1.26 - Введення імені

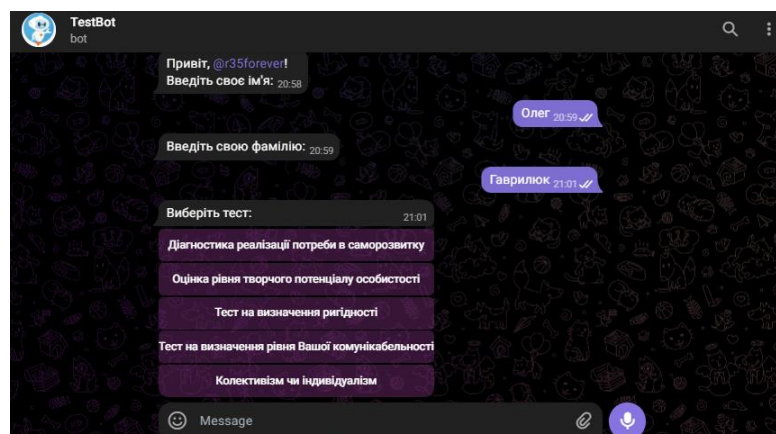


Рисунок 1.27 - Введення прізвища

Після вводу прізвища бот надав мені список тестів, та записав мене у базу даних. Виберу один з тестів.

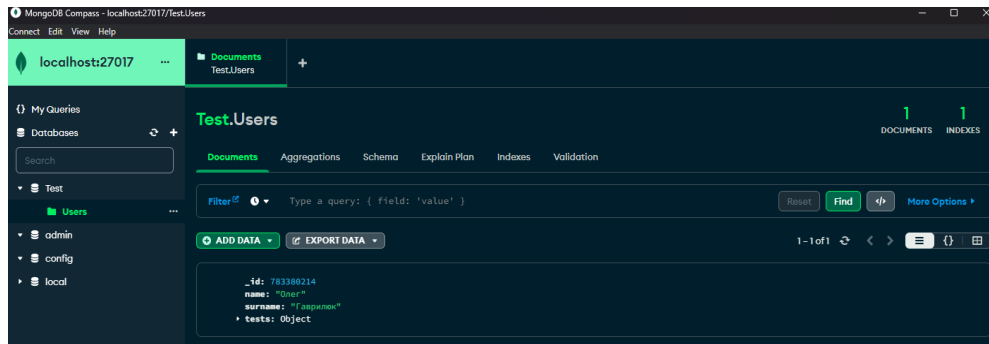


Рисунок 1.28 - Запис у базі даних

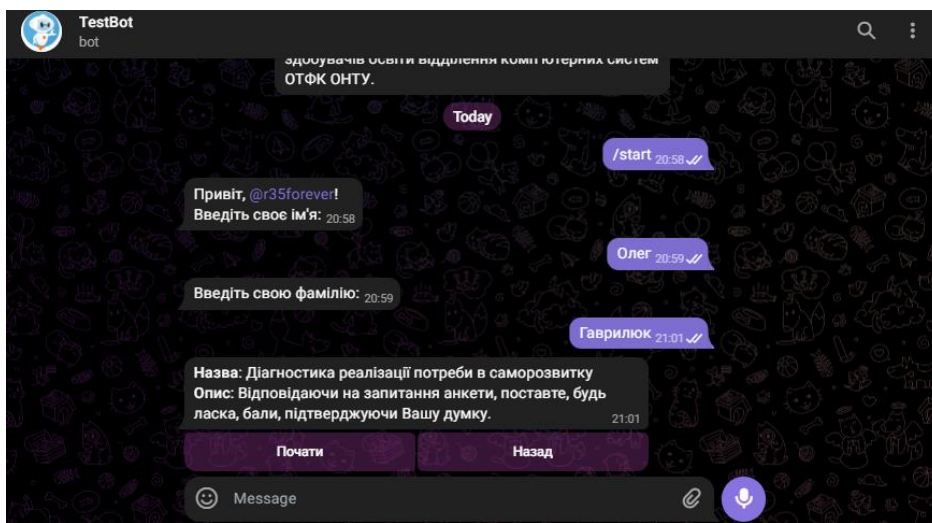


Рисунок 1.29 - Вибір тесту

Як ми побачили зник вибір тестів після натискання на тест, та з'явився опис тесту. Натиснемо «Почати».

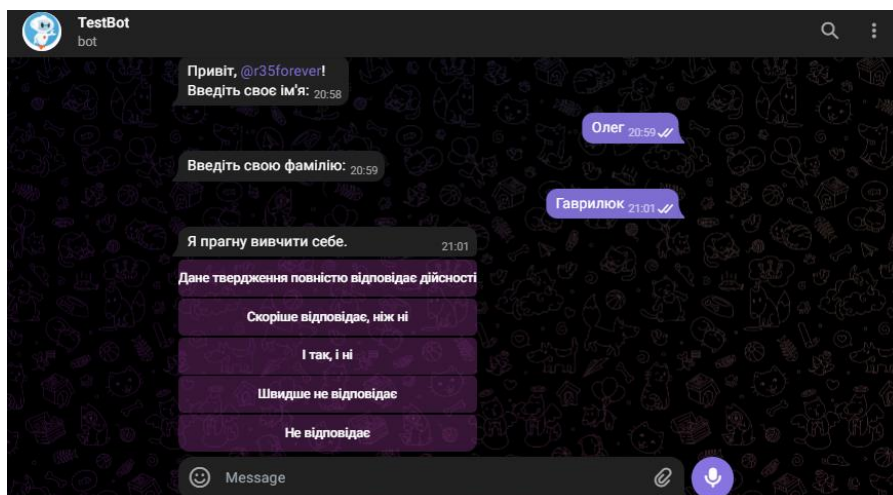


Рисунок 1.30 - Запитання тесту

Знову же, зник опис тесту, та з'явилося перше запитання тесту та кнопки для відповіді. Відповім на всі запитання.

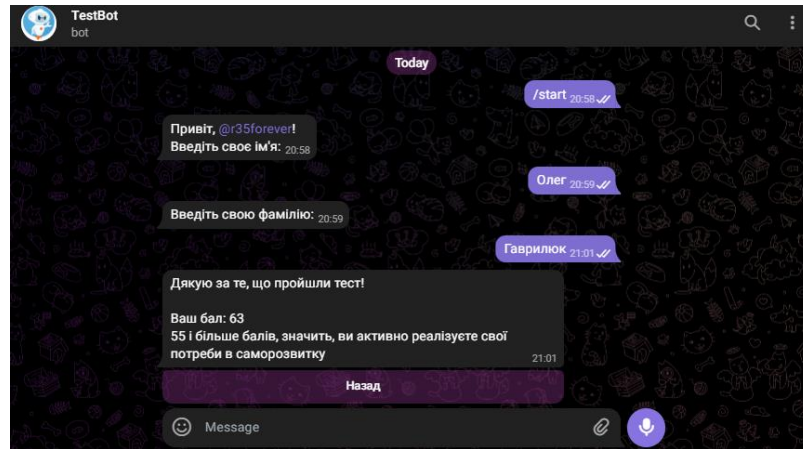


Рисунок 1.31 - Результат тесту

Як ми бачимо, після усіх запитань, бот вивів мій результат та кнопку «Назад». А ще бот записав у базу даних мої відповіді на запитання. Натиснемо на кнопку.

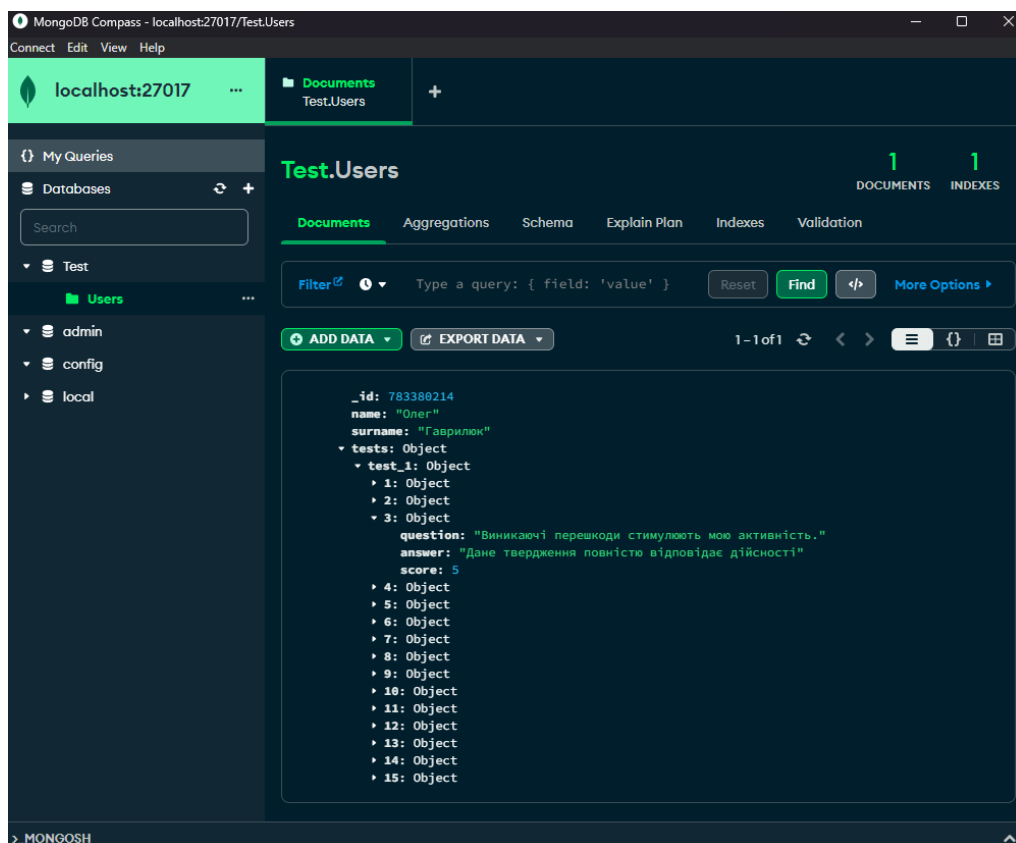


Рисунок 1.32 - Запис у базу даних відповідей на запитання

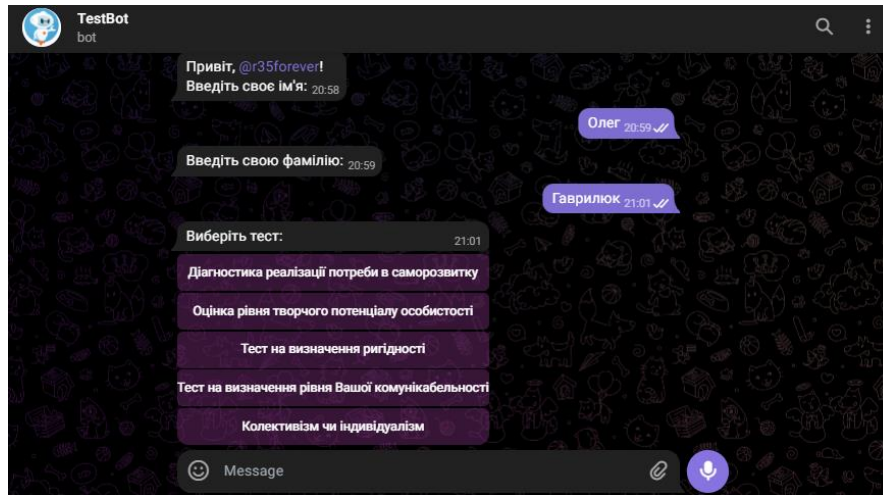


Рисунок 1.33 - Реакція на кнопку «Назад»

Бот вивів нас назад на вибір тесту, і ми можемо розпочати інший тест, або пере пройти той самий.

Отже, в даному розділі було пройдено багато етапів, таких як: створення чат-боту через BotFather, написання коду для його роботи та результат і тестування завершеного бота.

Результати роботи свідчать про успішну реалізацію поставлених завдань. Бот створений і успішно функціонує. Написаний код на Python дозволяє забезпечувати правильну роботу бота та його функціональні можливості.

Тестування бота підтвердило його коректну роботу, а також виявило деякі можливі покращення, які можуть бути внесені в подальшій розробці. Перший запуск бота був успішним, і він готовий до подальшого використання здобувачами освіти.

2 ЕКОНОМІЧНИЙ РОЗДІЛ

2.1 Резюме

В даному дипломному проєкті розроблений чат-бот на основі соціальної мережі Telegram. Цей чат-бот створений для тестування контингенту здобувачів освіти відділення комп'ютерних систем. Наразі цей бот може проводити психологічні тести для здобувачів освіти. На платформі Telegram він створений для того щоб не було проблем з легким та швидким доступом до нього. Ця соціальна мережа є майже у всіх на телефоні. Побудований на мові програмування Python, тому що ця мова наразі є одна з найпопулярніших та найвикористовуваниших мов програмування, вона славиться легкою читабельністю коду та своєю переносимістю.

Ефективність кожного програмного продукту визначається його якістю та ефективністю процесу розробки. Якість ПП визначається наступними складовими: з точки зору користувача; з позиції використання ресурсів; виконання вимог до програмного забезпечення.

Оцінка якості програмного продукту з точки зору користувача визначається необхідним на стадії функціонування розміром оперативної пам'яті ЕОТ, витратами машинного часу, пропускнуною спроможністю каналів передачі даних.

Оцінка якості програмного продукту включає визначення трудомісткості і вартості його створення.

2.2 Визначення трудомісткості розробки програмного забезпечення

Тривалість розробки програмного продукту залежить від його обсягу, трудомісткості розробки, кваліфікації виконавців, а також планових термінів, визначених умовами ринку. Методом структурної аналогії по відповідних каталогах аналогів програмного забезпечення визначається обсяг програмних засобів, у тисячах умовних машинних команд програми аналога.[13]

					<i>КС 56. 04 000. 00 ДП ПЗ</i>	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		38

Таблиця 2.1 - Каталог аналогів

Найменування ПП	Обсяг функції ПП – V_0 , усл. машинних командах.
1. ПП автоматизованих розрахунків	1300 – 8600
3. ПП введення інформації	1060 – 5750
4. ПП оптимізації розрахунків	1300 – 4200

У таблиці 2.1 представлені аналоги програмного забезпечення, функції яких, у більшому або меншому ступені, виконує розроблений програмний продукт. Для нашого варіанта виділено сірим кольором.

Вибравши аналог ПП, що містить V_0 в умовних машинних командах, трудомісткості визначати на основі табл.2.2.

Таблиця 2.2 – Обсяг ПП

Обсяг ПП, тис.умов. машинних команд	Норма часу, люд/год
1.00	229
2.00	244
3.00	262

На підставі отриманого значення, по довіднику, визначається укрупнена норма часу на розробку аналога програмного забезпечення (коректується поправочним коефіцієнтом враховуючої умови розробки ПП, тобто в умовах комп'ютера, $K_k=0,7 \div 0,8$): $T^a = 229 \times 0,8 = 183,2$ (люд/годин).

Трудомісткість програмного продукту визначається по кожному етапу розробки окремо на підставі трудомісткості аналога з урахуванням складності розробки, ступеня новизни і ступеня використання в розробці стандартних модулів на підставі формул:

$$T_{T3} = T^a p \times L_1 \times K_H \quad (2.1)$$

$$T_{TII} = T^a p \times L_2 \times K_H \quad (2.2)$$

$$T_{PII} = T^a p \times L_3 \times K_H \times K_T \quad (2.3)$$

Для розрахунку необхідні наступні коефіцієнти:

L_i – питома вага i -го етапу розробки (див. табл. 2.3.);

K_n – поправочний коефіцієнт, що враховує ступінь новизни (див. табл. 2.4.);

K_t – поправочний коефіцієнт, що враховує ступінь використання в розробці типових програм (див. табл. 2.5.).

Таблиця 2.3 - Значення питомих коефіцієнтів трудомісткості стадії в загальній трудомісткості розробки ПП

Код стадії	Ступінь новизни		
	А	Б	В
ТЗ (L_1)	0,15	0,12	0,12
ТП (L_2)	0,16	0,15	0,11
РП (L_3)	0,55	0,58	0,61

Для нашого варіанта виділено сірим кольором.

Таблиця 2.4 - Значення поправочного коефіцієнта, що враховує ступінь новизни

Код ступеня новизни	Ступінь новизни	Значення K_n
А	Принципово нові ПП	1,75 – 1,2
Б	ПП – розвиток визначеного параметричного ряду	1,0 – 0,8
В	ПП маючий аналог	0,7

Для нашого варіанта виділено сірим кольором.

Таблиця 2.5 - Значення коефіцієнта ступеня використання в розробці типових програм

Ступінь охоплення реалізованих функцій розроблювального ПП типовими програмами, %	Значення K_t
60 і вище	0,6
40-60	0,7
20-40	0,8
До 20	0,9

Для нашого варіанта виділено сірим кольором.

Тепер розраховуємо трудомісткість по кожному етапу окремо:

Трудомісткість технічного завдання

$$T_{ТЗ} = T_a * L_1 * K_n = 183.2 * 0,12 * 0,9 = 19.8 \text{ (люд/годин)} \quad (2.1)$$

Трудомісткість розробки технічного проекту

$$T_{\text{ТП}} = T^a * L_2 * K_H = 183.2 * 0,15 * 0,9 = 24.7 \text{ (люд/годин)} \quad (2.2)$$

Трудомісткість розробки робочого проекту

$$T_{\text{РП}} = T^a * L_3 * K_H * K_T = 183.2 * 0,58 * 0,9 * 0,7 = 67 \text{ (люд/годин)} \quad (2.3)$$

Для подальших розрахунків визначили кількість папера, витраченого на кожен етап: технічне завдання $N_{\text{ТЗ}} = 2$ (стр), розробка ТП $N_{\text{ТП}} = 10$ (стр), розробка робочого проекту $N_{\text{РП}} = 15$ (стр), пояснювальна записка відповідно $N_{\text{ПЗ}} = 23$ (стр) Розрахунок зведений у таблицю 2.6. [14]

Таблиця 2.6. Розрахунок трудомісткості ПП

Найменування етапів	Розрахунок, годин.		
	2	3	4
1.Розробка ТЗ	$T_{\text{РТЗ}} = 19.8$	$T_{\text{КК}} = 0,7 * N_{\text{ТЗ}} = 0,7 * 2 = 1.4$	$T_{\text{НК}} = 0,15 * N_{\text{ТЗ}} = 0,15 * 2 = 0,3$
2.Розробка ТП	$T_{\text{РТП}} = 24.7$	$T_{\text{КК}} = 0,7 * N_{\text{ТП}} = 0,7 * 10 = 7$	$T_{\text{НК}} = 0,15 * N_{\text{ТП}} = 0,15 * 10 = 1,5$
3.Розробка РП	$T_{\text{РРП}} = 67$	$T_{\text{КК}} = 0,7 * N_{\text{РП}} = 0,7 * 15 = 10,5$	$T_{\text{НК}} = 0,15 * N_{\text{РП}} = 0,15 * 15 = 1,8$
4.Розробка ПЗ	$T_{\text{ПЗ}} = 1,5 * N_{\text{ПЗ}} = 34,5$	$T_{\text{КК}} = 0,7 * N_{\text{ПЗ}} = 0,7 * 23 = 16,1$	$T_{\text{НК}} = 0,15 * N_{\text{ПЗ}} = 0,15 * 23 = 2,76$
Усього, в т.ч.:	$\Sigma T = 187,4$		
- на розробку	$\Sigma T_p = 146$		
- контроль керівника		$\Sigma T_{\text{КК}} = 35$	
- нормоконтроль			$\Sigma T_{\text{НК}} = 6,4$

2.3 Розрахунок ціни програмного продукту

У цьому розділі для визначення ціни розраховуємо основну заробітну плату виконавців, матеріальні витрати, вартість машино – години і витрати на розробку ПО. Розрахунок основної заробітної плати виконавців приведений у таблиці 4.6. Відповідно до статті 8 «Закону про Державний бюджет України на

2023» встановлено мінімальну заробітну плату у місячному розмірі з 1 січня 2023 року - 6700 гривень; мінімальну погодинну тарифну ставку – 40.46 грн.

Таблиця 2.7 - Розрахунок основної заробітної плати виконавців

Найменування робіт	Трудомісткість робіт, години	Погодинна тарифна ставка, грн.	Розрахунок, грн.
1.Розробка ПП	Таб.2.5 $\Sigma T_p=146$	40,46	5910
2.Контроль керівника	$\Sigma T_{kk}=35$	70.00	2450
3.Нормоконтроль	$\Sigma T_{нк}=6,4$	70.00	450
Усього	-	-	$\Sigma Z_o= 8810$

Зробимо розрахунок матеріальних витрат на розробку ПП. Розрахунок зведемо в таблицю 2.8

Таблиця 2.8 - Розрахунок матеріальних витрат на розробку ПО

Найменування матеріальних витрат	Тип, модель	Кількість	Ціна одиниці, грн.	Вартість, грн.
Папір	Лист А4	71	1,5	$V_{mi}=106,5$
Транспортні витрати (10%)				$V_{тр_з} = 0,1 \times V_{m1} = 10,65$
Усього				$V_m = V_{mi} + V_{тр_з} = 117,1$

На підставі отриманих даних по окремих статтях витрат складена калькуляція планової собівартості в цілому ПП за формою, приведеною в таблиці 2.9. Розмір прибутку, що включається в ціну, визначаємо по наступній формулі:

$$П = (C_{пов} * P) / 100\% = 15119 * 10\% / 100\% = 1511,9 \quad (2.9)$$

Де P – плановий рівень рентабельності (10-15%).

Оптова ціна (кошторисна вартість) визначається по формулі:

$$C_o = C_{пов} + П = 15119 + 1511,9 = 16630,9 \quad (2.5)$$

Податок на додану вартість визначаємо по наступній формулі:

					КС 56. 04 000. 00 ДП ПЗ	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		42

$$\text{ПДВ} = 0,2 * \text{Ц}_0 = 0,2 * 16630,9 = 3326,2 \quad (2.6)$$

Таблиця 2.9 - Розрахунок статей витрат планової собівартості

Стаття витрат	Значення, грн.	Формула розрахунку
1. Матеріали	117,1	V_m (див. табл. 2.8)
2. Основна заробітна плата	8810	Z_o (див. табл. 2.6)
3. Додаткова заробітна плата	1321	$Z_d = 0,15 \times Z_o = 0,15 * 8810$
4. Відрахування до єдиного фонду соціального внеску	2228	$V_{\text{е.с.в.}} = 0,22 \times (Z_o + Z_d) =$ $= 0,22 * (8810 + 1321)$
5. Накладні витрати	2643	$V_{\text{нак.}} = 0,3 \times Z_o = 0,3 * 8810$
6. Повна собівартість	15119	$C_{\text{пов.}} = V_m + Z_o + Z_d + V_{\text{е.с.в.}} + V_{\text{нак.}} =$ $= 117,1 + 8810 + 1321 + 2228 + 2643$

Виходячи з отриманих даних, ціна реалізації розробленого програмного продукту на основі наступної формули, становитиме:

$$\text{Ц}_p = \text{Ц}_0 + \text{ПДВ} = 16630,9 + 3326,2 = 19957 \quad (2.7)$$

3 ОХОРОНА ПРАЦІ

У галузі охорони праці є Закон України «Про охорону праці», дія якого поширюється на всі підприємства, установи і організації незалежно від форм власності та видів їх діяльності, на усіх громадян, які працюють на цих підприємствах. Власник або уповноважені ним органи зобов'язані дбати про умови праці працівників, полегшувати їх, оздоровляти навколишнє середовище, дбати про виконання правил безпеки і інструкцій по техніці безпеки. Працівники також повинні відповідально ставитись до охорони праці, знати та виконувати вимоги, визначені нормативною документацією.

Дипломним проектом передбачається розробка чат-боту на платформі Telegram та мові програмування Python. Тому для розгляду беремо робоче місце програміста.[15]

3.1 Аналіз небезпечних та шкідливих чинників, що впливають на працівника

Безпечні умови праці на підприємстві досягаються за рахунок забезпечення безпеки виробничих процесів, які обґрунтовані і прийняті в технологічній частині дипломного проекту.

Для установлення можливого впливу на здоров'я користувачів ВДТ виробничих чинників має значення ряд якісних характеристик робочого середовища. Це середовище у приміщеннях (офісах) в основному характеризується такими фізичними параметрами, як температура, вологість та електричний опір підлоги. Фізико-хімічні показники включають інформацію про вміст у повітрі іонів та різноманітних забруднювачів, а також деякі інші якісні характеристики середовища.

					КС 56. 04 000. 00 ДП ПЗ	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		44

3.2 Розробка заходів з охорони праці

3.2.1 Виробниче приміщення

Площу приміщень, в яких розташовують персональні комп'ютери, визначають згідно з чинними нормативними документами з розрахунку на одне робоче місце, обладнане ПК:

- площа – не менше 6,0 кв.м,
- обсяг – не менше 20,0 куб.м, з урахуванням максимальної кількості осіб, які одночасно працюють у зміні.

Робочі місця з ПК заборонено облаштовувати у підвальних або цокольних приміщеннях будинків.

В обладнанні приміщень забороняється використання полімерних матеріалів (деревинно-стружкові плити, шпалери, що миються, рулонні синтетичні матеріали, шаруватий паперовий пластик тощо), що виділяють у повітря шкідливі хімічні речовини. Покриття підлоги повинно бути матовим, а поверхня – рівною, неслизькою, з антистатичними властивостями. Заземлені конструкції, що знаходяться в приміщеннях (батареї опалення, водопровідні труби, кабелі із заземленим відкритим екраном тощо), мають бути надійно захищені діелектричними щитками або сітками від випадкового дотику.

Особливу увагу необхідно приділити колірній гармонії офісних приміщень. Колір є засобом створення психологічного комфорту та підвищення продуктивності праці. Найбільш сприятливі для нервової системи світлі, пастельні тони – зеленувато-блакитний, ясно-сірий, золотавий. Яскраві, контрастні поєднання (синій і жовтогарячий, червоний і фіолетовий) викликають втому, роздратування.

В приміщенні є вікна, які орієнтовані на південь . Тому основні кольори інтер'єру – стіни світло-блакитного кольору; підлога-зеленого. У приміщеннях, де здійснюється робота з комп'ютерами, щодня має проводитися вологе прибирання з метою недопущення запиленості підлоги та меблів.

					КС 56. 04 000. 00 ДП ПЗ	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		45

Також в цих приміщеннях повинні бути медичні аптечки першої допомоги та система автоматичної пожежної сигналізації з димовими пожежними сповіщувачами та переносними вуглекислотними вогнегасниками з розрахунку 2 шт. на кожні 20 кв.м площі приміщення. Підходи до засобів пожежогасіння повинні бути вільними.

Всі вказані вимоги до приміщень де використовується робота з ПК в дипломному проєкті виконані.

3.2.2 Мікроклімат робочої зони працівників, вентиляція

Оптимальні та допустимі мікрокліматичні параметри у приміщеннях повинні враховувати специфіку технологічного процесу при використанні комп'ютерів. Згідно з діючими у нашій країні нормативними документами (ДСанПіН 3.3.2-007-98

В приміщенні з ПК дотримуються оптимальних значень мікроклімату. Їх параметри відображені в таблиці 3.1

Таблиця 3.1 - Норми мікроклімату для приміщень з ПК

Порароку	Категорія робіт	Температура повітря, °С, не більше	Відносна вологість повітря, %	Швидкість руху повітря, м/с
Холодна	легка -1а	22-24	40-60	0,1
	легка -1б	21-23	40-60	0,1
Тепла	легка -1а	23-25	40-60	0,1
	легка -1б	22-24	40-60	0,2

Рівні позитивних і негативних іонів у повітрі приміщень з ПК мають відповідати санітарно-гігієнічним нормам №2152–80 (табл.3.2).

Таблиця 3.2 - Рівні іонізації повітря приміщень при роботі на ПК

Рівні	Кількість іонів в 1 см ³ повітря	
	n+	n-
Мінімально необхідні	400	600
Оптимальні	1500-3000	3000-5000
Максимально допустимі	50000	50000

Нормалізуючий вплив на склад повітря робочої зони справляють примусова вентиляція, захисні екрани (оснащені заземленням) та застосування іонізаторів.

3.2.3 Шум

Джерелами шуму при роботі з комп'ютерною технікою є жорсткий диск, вентилятор блока живлення мережі, вентилятор розташований на процесорі, швидкісні CD-ROM, механічні сканери, пересувні механічні частини принтера. При роботі матричних голкових принтерів шум виникає при переміщенні головки принтеру і в процесі удару голок головки по паперу. При роботі вентиляційної системи ПК, яка забезпечує оптимальний температурний режим електронних блоків ПК і вмонтована в задню панель, створюється аеродинамічний шум. Окрім того діють і інші зовнішні джерела шуму, не пов'язані з роботою ПК

Для офісів та приміщень, обладнаних персональними комп'ютерами або технікою для бізнесу допустимий рівень шуму цілодобово - 50 дБА, а максимальний- 65 дБА

3.2.4 Освітлення робочого місця

Освітлення повинно бути достатнім, рівномірним і відповідати вимогам здоров'я та безпеки.

Використання природного світла найбільш природне та комфортне для очей. У разі потреби додатково використовують штучне освітлення, такі як стелеві світильники, настільні лампи або світлодіодні панелі. Необхідно забезпечити рівномірне розподілення світла по всій робочій зоні. При цьому потрібно уникати сильних тіней, відблисків або надмірного контрасту. Світло повинно бути розподілене рівномірно на робочій поверхні.

Важливо налаштувати яскравість освітлення таким чином, щоб було зручно читати текст на моніторі та документах без напруження очей. Занадто

					КС 56. 04 000. 00 ДП ПЗ	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		47

яскраве освітлення може створювати блиск на моніторі, тоді як занадто тьмяне освітлення може призводити до напруження очей. Кольорова температура впливає на загальний настрій та концентра

3.2.5 Електробезпека

Персональні комп'ютери, периферійні пристрої, інше устаткування (апарати управління, контрольно-вимірювальні прилади, світильники), електропроводи та кабелі за виконанням і ступенем захисту відповідають класу зони, мають апаратуру захисту від струму короткого замикання та інших аварійних режимів.

Лінія електромережі для живлення персональних комп'ютерів і периферійних пристроїв виконана як окрема групова трипровідна мережа шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Персональні комп'ютери і периферійні пристрої повинні підключатися до електромережі тільки за допомогою справних штепсельних з'єднань і електророзеток заводського виготовлення.

При розміщенні в приміщенні до п'яти персональних комп'ютерів і периферійних пристроїв допускається прокладання трипровідникового захищеного проводу або кабелю в оболонці з негорючого чи важкогорючого матеріалу по периметру приміщення без металевих труб та гнучких металевих рукавів.[16]

3.2.6 Організація робочого місця користувача ПК

Обладнання і організація робочого місця з ВДТ мають забезпечувати відповідність конструкцій всіх елементів робочого місця та їх взаємного розташування, ергономічним вимогам, з урахуванням характеру і особливостей трудової діяльності (ДСанПіН 3.3.2.-007-98).

					КС 56. 04 000. 00 ДП ПЗ	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		48

Робочі місця повинні бути розташовані на відстані не менше ніж 1 м. від стіни з вікном, Відстань між бічними поверхнями комп'ютерів має бути не меншою за 1,2м, відстань між тильною поверхнею одного комп'ютера та екраном іншого не повинна бути меншою 2,5 м, прохід між рядами робочих місць має бути не меншим 1 м.

Конструкція робочого місця й взаємне розташування всіх його елементів (сидіння, органи керування, засобу відображення інформації) відповідають антропометричним, фізіологічним і психологічним вимогам, а також характеру роботи.

Конструкція робочих меблів дає можливість забезпечувати можливість індивідуального регулювання їх відповідно до потреб працівника для підтримки зручної пози. Робочий стіл повинен бути пофарбований матовою фарбою. Дисплей розташований так, що його верхній край перебуває на рівні очей, на відстані близько 70 см, що укладається в припустимі рамки від 60 до 90 см. Частота мерехтіння екрана дорівнює 100 Гц, що відповідає умові більше 70 Гц.

Для зниження нервово-емоційного напруження, стомлювання, поліпшення мозкового кровообігу, подолання несприятливих наслідків гіподинамії, запобігання втомі доцільно впроваджувати виконання комплексу вправ, які наведені у Державних санітарних правилах і нормах роботи з візуальними терміналами електронно-обчислювальних машин ДСанПіН 3.3.2.007-98.

3.2.7 Пожежна безпека при роботі з ВДТ

Пожежна безпека - стан об'єкта, при якому виключається можливість пожежі, а в разі його виникнення запобігається вплив на людей небезпечних факторів пожежі і забезпечується захист матеріальних цінностей.

Пожежна безпека забезпечується системою запобігання пожежі і системою пожежного захисту. У всіх службових приміщеннях обов'язково повинен бути "План евакуації людей при пожежі", що регламентує дії персоналу у разі виникнення вогнища загоряння і в якому зазначено місця розташування

					КС 56. 04 000. 00 ДП ПЗ	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		49

пожежної техніки. Пожежі становлять особливу небезпеку, тому що пов'язані з великими матеріальними втратами.

У сучасних ПК дуже висока щільність розміщення елементів електронних схем. У безпосередній близькості один від одного розташовуються сполучні дроти, кабелі. При протіканні по них електричного струму виділяється значна кількість теплоти. При цьому можливо оплавлення ізоляції. Для відведення надлишкової теплоти від ПК служать системи вентиляції та кондиціонування повітря. При постійному дії ці системи представляють собою додаткову пожежну небезпеку.

Горючими компонентами є: будівельні матеріали для акустичної і естетичної обробки приміщень, перегородки, двері, підлоги, перфокарти і перфострічки, ізоляція кабелів і ін.

Джерелами запалювання можуть бути електронні схеми від ПК, прилади, застосовувані для технічного обслуговування, пристрої електроживлення, кондиціонування повітря, де внаслідок різних порушень утворюються перегріті елементи, електричні іскри та дуги, здатні викликати загоряння горючих матеріалів.

До засобів гасіння пожежі відносяться внутрішні пожежні водопроводи (крани - ПК), вогнегасники (вуглекислотні та порошкові), сухий пісок тощо.

В будівлях пожежні крани встановлюють в коридорах, на майданчиках сходових кліток. Кожний пожежний кран укомплектований пожежним рукавом і розміщений у відповідних ящиках, які знаходяться на висоті 1,35 м від полу.

Для гасіння пожеж на початкових стадіях широко застосовуються вогнегасники.

У виробничих приміщеннях це головним чином вуглекислотні вогнегасники, достоїнством яких є висока ефективність гасіння пожежі, збереження електричного устаткування. Розташовують вогнегасники на видних місцях, на висоті не більше як 1,5 м від полу.

					<i>КС 56. 04 000. 00 ДП ПЗ</i>	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		50

Вогнегасники повинні бути розташовані на вертикальній перегородці або стіні, на спеціальних кронштейнах або в пожежних шафах. Інструкції та правила щодо їх використання, нанесені на корпус вогнегасника, повинні бути повернені назовні і добре видно в екстрених ситуаціях. Механізми запуску і двері пожежних шаф повинні мати захисні пломби.



Рисунок 3.3 – Засоби пожежогасіння

Виробничі приміщення мають запасні виходи. Двері повинні мати освітлений надпис «Запасний вихід». План евакуації вивіщується на видному місці у основного виходу із приміщення.

					КС 56. 04 000. 00 ДП ПЗ	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		51

ВИСНОВКИ

Виконаний дипломний проект спрямований на створення чат-боту у Telegram для тестування контингенту здобувачів освіти відділення комп'ютерних систем. У процесі роботи було проведено дослідження та розробку моделі чат-боту, а також реалізовано необхідні алгоритми та програмне забезпечення для його функціонування.

Метою проекту було розробити ефективний та зручний інструмент для проведення тестування здобувачів освіти відділення комп'ютерних систем. У результаті виконаної роботи були досягнуті такі цілі:

1. Розроблено архітектуру чат-боту, яка дозволяє забезпечити зручне та логічно організоване взаємодію з користувачами.
2. Реалізовано функціональні можливості чат-боту, такі як надання запитань, прийом відповідей, оцінювання та збереження результатів тестування.
3. Розроблено алгоритми обробки запитань та оцінювання відповідей, що дозволяють надати об'єктивну оцінку знань здобувачів освіти.
4. Виконано розробку програмного забезпечення для реалізації чат-боту у Telegram та його інтеграції з необхідними інструментами.

Використання запропонованого чат-боту покращить якість процесу тестування контингенту здобувачів освіти відділення комп'ютерних систем. Він забезпечить зручний та швидкий доступ до тестових завдань, автоматичне оцінювання результатів та збереження їх для подальшого аналізу. Це дозволить ефективніше вимірювати рівень знань студентів і вчасно виявляти їхні проблемні місця.

Таким чином, можна зробити висновок, що виконаний проект повністю відповідає поставленим завданням і сприятиме покращенню процесу тестування контингенту здобувачів освіти відділення комп'ютерних систем.

					<i>КС 56. 04 000. 00 ДП ПЗ</i>	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		52

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Що таке чат-бот: секрети використання та основні переваги для бізнесу [Електронний ресурс] - <https://helpcrunch.com/blog/uk/shcho-take-chat-bot/>
2. Боти в телеграм: що це таке і як вони працюють [Електронний ресурс] - <https://107.com.ua/poradu/boti-v-telegram-sho-ce-take-%d1%96-iak-voni-pracuut/>
3. Приголомшлива статистика чат ботів у 2023 році: переваги, тенденції [Електронний ресурс] - <https://bloggersideas.com/uk/chatbot-statistics/>
4. Названо найпопулярніші месенджери серед українців – УНІАН [Електронний ресурс] - <https://www.unian.ua/economics/telecom/opituvannya-pokazalo-yakiy-z-mobilnih-dodatki-ye-naybilsh-populyarnim-v-ukrajinciv-novini-11614792.html>
5. Телеграм - вікіпедія [Електронний ресурс] - <https://uk.wikipedia.org/wiki/Telegram>
6. Всё о чем должен знать разработчик Телеграм-ботов [Електронний ресурс] - <https://habr.com/ru/articles/543676/>
7. Python – Вікіпедія [Електронний ресурс] - <https://ru.wikipedia.org/wiki/Python>
8. Visual Studio Code – Вікіпедія [Електронний ресурс] - https://uk.wikipedia.org/wiki/Visual_Studio_Code
9. Telegram Bot AIP и Telegram API [Електронний ресурс] - <https://highload.today/telegram-api/>
10. Как начать работу с aiogram: установка и первые шаги [Електронний ресурс] - <https://aiogram.ru/?p=112>
11. MongoDB – Вікіпедія [Електронний ресурс] - <https://uk.wikipedia.org/wiki/MongoDB>
12. Как сделать бота в телеграмме | Telegram Bot [Електронний ресурс] - https://www.youtube.com/watch?v=tq8VrdBfRik&t=144s&ab_channel=Likeus

13. Економіка підприємства: навчальний посібник / О.І. Лисак, Л.О. Андрєєва, Л.О. Болтянська. -Мелітополь: Люкс, 2020. -272с.
14. Економіка підприємства: навч. Посіб. / [І.М. Посохов, В.Г. Дюжев, С.В. Сусліков, К.О. Тимофєєва]; М-во освіти і науки України, Нац. Техн. ун-т. «Харків. Політехн. ін-т». – Харків: «ХПІ», 2016. – 380 с.
15. Основи охорони праці. навчально-методичний посібник для студентів вищих закладів педагогічного напрямку / [Укладачі: В.І. Кошель, Г.П. Сав'юк, Б.С. Дзундза] – Івано-Франківськ: НАІР, 2020. –182 с.
16. Козяр М.М., Щедрий Я.І., Станіславчук О.В. Основи охорони праці, безпеки життєдіяльності та цивільного захисту населення: Навч.посіб. -К.: Кондор, 2012.

					<i>КС 56. 04 000. 00 ДП ПЗ</i>	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		54

```
from aiogram import Bot, types
from aiogram.dispatcher import Dispatcher

from aiogram.utils.exceptions import MessageCantBeEdited, MessageNotModified

from pymongo.collation import Collation

from scripts.settings import API_TOKEN, DATABASE, TESTS

from typing import Union
import asyncio

# Отримання поточного циклу подій
loop = asyncio.get_event_loop()

# Створення екземплярів Bot та Dispatcher
bot = Bot(token=API_TOKEN, loop=loop)
dispatcher = Dispatcher(bot=bot)

# Отримання колекції користувачів з бази даних
users: Collation = DATABASE.Users

# Обробник команди /start
@dispatcher.message_handler(commands=['start'])
async def send_welcome(message: types.Message) -> None:
    await bot.send_message(
        chat_id=message.chat.id,
        text=f'Привіт, @ {message.from_user.username}!\nВведіть своє ім\'я:'
    )

    # Якщо користувача не знайдено в базі даних, то додаємо його
    if not users.find_one({'_id': message.from_user.id}):
        users.insert_one({'_id': message.from_user.id, 'name': None, 'surname': None,
            'tests': {}})

# Відправка списку тестів
async def send_tests(message: types.Message) -> None:
    keyboard = types.InlineKeyboardMarkup()

    for test in TESTS:
        keyboard.add(types.InlineKeyboardButton(text=test['name'],
            callback_data=test['id']))
```

```

try:
    await message.edit_text(text='Виберіть тест:', reply_markup=keyboard)
except MessageCantBeEdited:
    await bot.send_message(chat_id=message.chat.id, text='Виберіть тест:

reply_markup=keyboard)

# Обробник усіх повідомлень, крім команди /start
@dispatcher.message_handler()
async def messages(message: types.Message) -> None:
    user: dict = users.find_one({'_id': message.from_user.id})

    if user['name'] is None:
        users.update_one({'_id': message.from_user.id}, {'$set': {'name':
message.text}})
        await bot.send_message(chat_id=message.chat.id, text='Введіть свою
фамілію:')
    elif user['surname'] is None:
        users.update_one({'_id': message.from_user.id}, {'$set': {'surname':
message.text}})
        await send_tests(message=message)
    else:
        await send_tests(message=message)

# Отримання тесту за його ідентифікатором
async def get_test(test_id: int) -> Union[dict, None]:
    for test in TESTS:
        if test['id'] == test_id:
            return test
    return None

# Відправлення питання тесту і кнопок для відповіді
async def send_question(callback_query: types.CallbackQuery, test: dict,
test_question_id: int) -> None:
    keyboard = types.InlineKeyboardMarkup()

    if 'buttons' in test:
        for test_button_id, test_button in test['buttons'].items():
            keyboard.add(
                types.InlineKeyboardButton(text=test_button['text'],
callback_data=f"{test['id']}:{test_question_id}:{test_button_id}")
            )

        await callback_query.message.edit_text(text=test['questions'][test_question_id],
reply_markup=keyboard)

```

```

else:
    for test_button_id, test_button in
test['questions'][test_question_id]['buttons'].items():
        keyboard.add(
            types.InlineKeyboardButton(text=test_button['text'],
callback_data=f" {test['id']}:{test_question_id}:{test_button_id}")
        )

try:
    await
callback_query.message.edit_text(text=test['questions'][test_question_id]['question'],
reply_markup=keyboard)
except MessageNotModified:
    await
callback_query.message.edit_text(text=test['questions'][test_question_id]['question'],
reply_markup=keyboard)

# Обробник запитів зворотнього виклику
@dispatcher.callback_query_handler()
async def callback_queries(callback_query: types.CallbackQuery) -> None:
    test_data = callback_query.data.split(':')
    test = await get_test(test_id=int(test_data[0]))

    # Якщо тест існує, то виконуємо код нижче
    if test is not None:
        if len(test_data) == 1:
            keyboard = types.InlineKeyboardMarkup()
            keyboard.add(
                types.InlineKeyboardButton(text='Почати',
callback_data=f" {test['id']}:{start}"),
                types.InlineKeyboardButton(text='Назад', callback_data='0')
            )

            await callback_query.message.edit_text(
                text=f"*Назва*: {test['name']}\n*Опис*: {test['description']}",
                parse_mode='Markdown',
                reply_markup=keyboard
            )
        elif len(test_data) == 2 and test_data[1] == 'start':
            await send_question(callback_query=callback_query, test=test,
test_question_id=1)
        elif len(test_data) == 3:
            test_question_id = int(test_data[1])
            test_button_id = int(test_data[2])

```

```

users.update_one(
    {
        '_id': callback_query.from_user.id,
    },
    {
        '$set': {
            f'tests.test_{test['id']}. {test_question_id}': {
                'question': test['questions'][test_question_id]['question'],
                'answer': test['buttons'][test_button_id]['text'] if 'buttons' in test else
test['questions'][test_question_id]['buttons'][test_button_id]['text'],
                'score': test['buttons'][test_button_id]['score'] if 'buttons' in test else
test['questions'][test_question_id]['buttons'][test_button_id]['score'],
            },
        },
    )

```

```

try:
    await send_question(callback_query=callback_query, test=test,
test_question_id=test_question_id + 1)
except KeyError:
    keyboard = types.InlineKeyboardMarkup()
    keyboard.add(
        types.InlineKeyboardButton(text='Назад', callback_data='0')
    )

```

```

user: dict = users.find_one({'_id': callback_query.from_user.id})
test_questions = user['tests'][f'test_{test['id']}']

```

```

test_total_score = 0

```

```

for test_question_id in test_questions:
    test_total_score += test_questions[test_question_id]['score']

```

```

for test_score_info in test['scores_info']:
    min_score, max_score = test_score_info.split('-')

```

```

    if int(min_score) <= test_total_score <= int(max_score):
        test_total_score_info = test['scores_info'][test_score_info]
        break

```

```

await callback_query.message.edit_text(text=f"""\
Дякую за те, що пройшли тест!

```

```
        Ваш бал: {test_total_score}
        {test_total_score_info}
        """.replace(' ', ''),
        reply_markup=keyboard
    )
else:
    await send_tests(callback_query.message)

# Запуск Telegram бота
async def start() -> None:
    await dispatcher.skip_updates()
    await dispatcher.start_polling()

# Якщо скрипт був запущений безпосередньо напямую, то запускаємо Telegram
бота
if __name__ == '__main__':
    try:
        loop.run_until_complete(start())
    except KeyboardInterrupt:
        loop.stop()
```

ДОДАТОК Б. Тести та опитувальники, які використані в проекті

Діагностика реалізації потреби в саморозвитку (І. Нікішина)

Інструкція. Відповідаючи на запитання анкети, поставте, будь ласка, бали, підтверджуючи Вашу думку: 5 - якщо дане твердження повністю відповідає дійсності; 4 - скоріше відповідає, ніж ні; 3 - і так, і ні; 2 - швидше не відповідає; 1 - не відповідає.

Тестовий матеріал.

- 1) Я прагну вивчити себе.
- 2) Я залишаю час для розвитку, якби не був зайнятий справами.
- 3) Виникаючі перешкоди стимулюють мою активність.
- 4) Я шукаю зворотній зв'язок, оскільки це допомагає мені пізнати й оцінити себе.
- 5) Я рефлексую свою діяльність, виділяючи для цього спеціальний час.
- 6) Я аналізую свої почуття і досвід.
- 7) Я багато читаю.
- 8) Я широко дискутую з питань, які мене цікавлять.
- 9) Я вірю у свої можливості.
- 10) Я прагну бути відкритою для інших людиною.
- 11) Я усвідомлюю вплив, який чинять на мене оточуючі люди.
- 12) Я керую своїм професійним розвитком і отримую позитивні результати.
- 13) Я отримую задоволення від освоєння нового.
- 14) Зростаюча відповідальність не лякає мене.
- 15) Я позитивно ставився б до просування по службі.

Обробка результатів. Підрахуйте загальну суму балів.

Ключ та інтерпретація. 55 і більше балів, значить, ви активно реалізуєте свої потреби в саморозвитку; діапазон від 36 до 54 балів свідчить про те, що у вас відсутня сформована система саморозвитку; результат від 15 до 35 балів дає підставу вважати, що саморозвиток на даний момент зупинився.

Оцінка рівня творчого потенціалу особистості (рекомендована методика Т. Морозовою)

Інструкція. Методика дозволяє визначити самооцінку особових якостей або частоту їх прояву, які і характеризують рівень розвитку творчого потенціалу особистості. За 9-бальною шкалою оцініть кожне з 18 тверджень.

Тестовий матеріал.

- 1) Як часто розпочату справу вам вдається довести до логічного кінця?
- 2) Якщо всіх людей подумки розділити на логіків та генераторів ідей, то якою мірою ви є генератором ідей?
- 3) Якою мірою ви відносите себе до людей рішучих?
- 4) Якою мірою ваш кінцевий «продукт», ваше творіння найчастіше відрізняється від початкового проєкту, задуму?
- 5) Наскільки ви здатні проявити вимогливість і наполегливість, щоб люди, які обіцяли вам щось, виконали обіцяне?
- 6) Як часто вам доводиться виступати з критичними відомостями на чийсь адресу?
- 7) Як часто рішення виникаючих у вас проблем залежить від вашої енергії та наполегливості?
- 8) Який відсоток людей у вашому колективі найчастіше підтримують вас, ваші ініціативи та пропозиції? (1 бал - близько 10%)
- 9) Як часто у вас буває оптимістичний і веселий настрій?
- 10) Якщо всі проблеми, які вам доводилося вирішувати за останній рік, умовно поділити на теоретичні та практичні, то яка серед них питома вага практичних?
- 11) Як часто вам доводилося відстоювати свої принципи, переконання?
- 12) Якою мірою ваша товарицькість, комунікабельність сприяє вирішенню життєво важливих для вас проблем?

13) Як часто у вас виникають ситуації, коли головну відповідальність за вирішення найбільш складних проблем і справ в колективі вам доводиться брати на себе?

14) Як часто і в якій мірі ваші ідеї, проєкти вдавалося втілювати в життя?

15) Як часто вам вдається, проявивши винахідливість і навіть підприємливість, хоч у чомусь випередити своїх суперників по роботі чи навчанню?

16) Як багато людей серед ваших друзів і близьких, які вважають вас людиною вихованою і інтелігентною?

17) Як часто вам у житті доводилося робити щось таке, що було сприйнято навіть вашими друзями як несподіванка, як принципово нова для вас справа?

18) Як часто вам доводилося докорінно реформувати своє життя або знаходити принципово нові підходи у вирішенні старих проблем?

Обробка та інтерпретація результатів.

На основі сумарного числа набраних балів визначте рівень вашого творчого потенціалу.

- 18-69 – низький рівень
- 70-129 – середній рівень
- 130-162 – високий рівень.

Тест на визначення ригідності (мобільності) (за Я. Поколовою)

Інструкція. Тест включає кілька тверджень, що вимагають однозначної відповіді («так» або «ні»). Кожна відповідь, що співпадає з ключем, дає один бал.

Тестовий матеріал.

1) Корисно читати книги, в яких містяться думки, протилежні моїм власним.

2) Я дратуюсь, коли мене відволікають від важливої роботи

- 3) Свята потрібно відзначати з родичами.
- 4) Я можу бути у дружніх стосунках з людьми, чії вчинки не схвалюю.
- 5) У грі я прагну вигравати.
- 6) Коли я запізнююся куди-небудь, я не в змозі думати ні про що інше, крім як доїхати швидше.
- 7) Мені важче зосередитися, ніж іншим.
- 8) Я багато часу приділяю тому, щоб всі речі лежали на своїх місцях.
- 9) Я дуже напружено працюю.
- 10) Непристойні жарти нерідко викликають у мене сміх.
- 11) Впевнений, що за моєю спиною про мене говорять.
- 12) Мене легко перемогти.
- 13) Я волію ходити відомими маршрутами.
- 14) Все своє життя я суворо дотримуюся принципів, що засновані на почутті боргу.
- 15) Часом мої думки проносяться швидше, ніж я встигаю їх висловити.
- 16) Буває, що чиясь безглузда помилка викликає у мене сміх.
- 17) Буває, що мені в голову приходять погані слова, часто навіть лайки, і я не можу їх позбутися.
- 18) Я впевнений, що в мою відсутність, про мене говорять.
- 19) Я спокійно виходжу з дому, не турбуючись про те, замкнені двері, чи вимкнене світло, газ тощо
- 20) Найважче для мене в будь-якій справі – це початок.
- 21) Я практично завжди стримую свої обіцянки.
- 22) Не можна строго засуджувати людину, яка порушує формальні правила.
- 23) Мені часто доводилося виконувати розпорядження людей, які знають менше ніж я.
- 24) Я не завжди кажу правду.
- 25) Мені важко зосередитися на якомусь одному завданні.

- 26) Дехто налаштований проти мене.
- 27) Я люблю доводити почате до кінця.
- 28) Я завжди намагаюся не відкладати на завтра те, що можна зробити сьогодні.
- 29) Коли я йду або їду по вулиці, то часто помічаю зміни в навколишньому середовищі – підстрижені куці, нові рекламні щити і т. д.
- 30) Іноді я так наполягаю на своєму, що люди втрачають терпіння.
- 31) Іноді знайомі жартують над моєю акуратністю і педантичністю.
- 32) Якщо я не правий, я не серджуся.
- 33) Зазвичай мене насторожують люди, які ставляться до мене дружелюбніше, ніж я очікував.
- 34) Мені важко відволіктися від початої роботи навіть ненадовго.
- 35) Коли я бачу, що мене не розуміють, я легко відмовляюся від наміру довести що-небудь.
- 36) У важкі моменти я вмію подбати про інших.
- 37) У мене тяга до зміни місць, і я щасливий, коли блукаю де-небудь або подорожую.
- 38) Мені нелегко перейти на нову справу, але потім, розібравшись, я впораюсь краще за інших.
- 39) Мені подобається детально вивчати те, чим я займаюся.
- 40) Мати або батько змушували підкорятися мене навіть тоді, коли я вважав це нерозумним.
- 41) Я вмію бути спокійним і навіть трохи байдужими до чужого нещастя.
- 42) Я легко перемикаюся з однієї справи на іншу.
- 43) З усіх думок щодо спірного питання лише одне дійсно є вірним.
- 44) Я люблю доводити свої вміння та навички до автоматизму.
- 45) Мене легко захопити новими витівками.
- 46) Я намагаюся домогтися свого всупереч обставинам.

47) Під час монотонної роботи я мимоволі починаю змінювати спосіб дії, навіть якщо це іноді погіршує результат.

48) Люди часом заздять моєму терпінню і скрупульозності.

49) На вулиці, в транспорті я часто роздивляюся оточуючих людей.

50) Якщо б люди не були налаштовані проти мене, я б досяг більшого.

Обробка результатів.

По одному балу за відповідь «ні»: 1, 4, 6, 8, 9, 11, 13, 14, 18, 20, 23, 26, 27, 30, 31, 33, 34, 38, 39, 40, 43, 44, 46, 48, 50 та по одному балу за відповіді «так»: 2, 3, 5, 7, 10, 12, 15, 16, 17, 19, 21, 22, 24, 25, 28, 29, 32, 35, 37, 41, 42, 45, 47, 49.

0 – 17 балів – особистість зовсім не відрізняється ригідністю, добре пристосовується, гнучка в поведінці, дуже сприйнятлива до нового; такий низький індекс властивий найчастіше юному віку, а також людям, що переживають стадію великих змін у своєму житті;

18 – 35 балів – особистість, що пристосовується до нових умов під тиском обставин; здатна сприймати нове;

28 – 40 – особистість, безумовно ригідна, відрізняється жорсткими настановами, категорична у судженнях, важко сприймає нове і важко перебудовується, і лише під тиском обставин.

Тест на визначення рівня Вашої комунікабельності

(рекомендовано В. Ряховським)

Інструкція. Для визначення Вашого рівня комунікативності слід відповісти на запропоновані нижче запитання. Варіанти відповідей: «так», «ні», «іноді».

Тестовий матеріал.

1) Вас чекає ординарна чи ділова зустріч. Вас вибиває з колії її очікування?

2) Чи відкладаєте Ви візит до лікаря до останнього моменту?

3) Чи викликає у Вас ніяковість і невдоволення доручення виступити з доповіддю, повідомленням, інформацією на будь-якій нараді, зборах чи іншому подібному заході?

4) Вам пропонують поїхати у відрядження туди, де Ви ніколи не були. Чи докладете Ви максимум зусиль, щоб уникнути цього відрядження?

5) Чи люблять Ви ділитися своїми переживаннями з кимось?

6) Чи дратує Вас, якщо незнайома людина на вулиці звертається до Вас із проханням (показати дорогу, назвати час, відповісти на якесь запитання)?

7) Чи вірите Ви, що існує проблема батьків і дітей, що людям різних поколінь важко розуміти один одного?

8) Чи соромитеся Ви нагадати знайомому, що він забув Вам повернути невелику суму грошей, яку позичив кілька місяців тому?

9) У ресторані чи в їдальні Вам подали недоброякісну страву. Чи промовчите Ви, лише сердито відсунувши тарілку?

10) Залишившись наодинці з незнайомою людиною, чи Ви вступите з нею в бесіду і відчуєте себе обтяженим(ою), якщо першою заговорить вона. Чи так це?

11) Чи жахаєтеся Ви великої черги? Віддасте перевагу відмові від свого наміру, чи станете у хвіст черги і будете знемагати в очікуванні?

12) Чи боїтеся Ви брати участь у будь-якій комісії з розгляду конфліктної ситуації?

13) У Вас є власні індивідуальні критерії оцінки творів художньої літератури, мистецтва, культури; з чужими думками Ви не рахуєтеся. Чи так це?

14) Почувши в кулуарах висловлювання явно помилкової точки зору з відомого Вам питання, Ви віддасте перевагу мовчанню і не вступите в суперечку?

15) Чи викликає у Вас неприємне почуття будь-яке прохання допомогти розібратися в тому чи іншому службовому питанні або навчальній темі?

16) Вам краще викласти свою точку зору (міркування, оцінку) в

письмовій формі, ніж в усній?

Обробка результатів. Підрахуйте суму набраних Вами балів. За кожне «так» – 2 бали, «ні» – 0 бал, «іноді» – 1. За класифікатором визначте, до якої категорії комунікабельних людей Ви належите.

25 – 32 бали – Ви явно не комунікабельні, і це Ваша біда, так як страждаєте від цього більше всього Ви самі. Але й близьким людям з Вами нелегко! На Вас не можна покластися у справі, яка потребує колективних зусиль. Прагніть бути більш комунікабельним, контролюйте себе.

14 – 24 бали – нормальна комунікабельність.

до 13 балів – Ви досить комунікабельні (іноді навіть занадто).

Зверніть увагу! Якщо Ви набрали менше 3 балів – Ваша комунікабельність має хворобливий характер. Ви балакучі, багатослівні, втручаєтеся у справи, в яких зовсім не компетентні, запальні й образливі.

Колективізм чи індивідуалізм (рекомендовано О.Вем)

Інструкція. Для визначення Вашого рівня колективізму слід визначити прийнятний для Вас варіант відповіді: а, б чи в.

Тестовий матеріал.

1. У відділі, де ви працюєте, всі збирають кошти на подарунок співробітнику, ви:

а) здасте ту ж суму, що і всі;

в) не дасте грошей, тому що у вас своя думка, який подарунок треба дарувати;

с) не здасте, тому що привітаєте на словах.

2. Вас попросили допомогти підготувати корпоративне свято, ви:

а) обов'язково допоможете;

в) постараетесь відговоритися невідкладними справами;

с) допоможете, якщо вам дадуть додатковий вихідний.

3. Все безпосереднє керівництво пішло у відпустку, ви:

- а) залишаєтеся на роботі весь робочий час;
- в) намагаєтеся працювати за принципом «не бий лежачого»;
- с) взагалі рідко з'являєтеся на роботі.

4. Ваші підлеглі постійно спізнюються на роботу, ви:

- а) постараетесь роз'яснити ситуацію, волаючи до духу корпорації;
- в) серйозно поговорите, волаючи до дисципліну та порядок;
- с) будете змушувати співробітників писати пояснювальні.

5. Коли ви вчилися в школі і були не згодні в якихось питаннях з вчителями, то ви:

- а) підходили до вчителя і говорили тет-а-тет;
- в) не лізли на рожен;
- с) говорили правду.

6. Ви вважаєте, що найкращий спосіб впливу на людину - це:

- а) пояснення, переконання;
- в) ласка;
- с) командний тон голосу.

7. На роботі назріває конфлікт, ваші дії:

- а) спробуєте вирішити його до загострення;
- в) підете на загострення;
- с) напишете заяву про звільнення.

8. Ви зробили роботу за співробітника, який після бурхливої вечірки відлежувався вдома, а похвала шефа дісталася йому, ви будете:

- а) поблажливо і багатозначно посміхатися, оскільки знаєте собі ціну;
- в) скажете колезі, що він вам повинен «могорич»;
- с) підійдете до шефа і поясните, що це ви все зробили.

9. Якщо вас не влаштовує робота вашого колеги, то ви:

а) тактовно поговорите з ним, пояснюючи, що від цього залежить не тільки його добробут, а й добробут фірми;

- в) зробіть зауваження;
- с) розповісте про це начальнику.

10. У вашого колеги вийшов з ладу комп'ютер, а ви йдете у відпустку, ви:

- а) запропонуйте йому пересісти на ваше робоче місце;
- в) підкажете, до якого системного адміністратора можна звернутися;
- с) потисне плечима: адже це не ваші проблеми.

Ключ до тесту.

За кожну відповідь а) поставте 3 бали. За відповідь б) поставте 2 бали. За відповідь с) – 1 бал. Визначте ваш сумарний бал. Якщо Ви набрали:

23-30 балів - у Вас високий показник колективізму. Ви - справжній командний працівник, який поважає своїх колег, домагається поваги до себе і може легко організувати спільну діяльність. Все це відбувається тому, що ви в міру відповідальні і принципові. Тільки не тисніть на співробітників своїм авторитетом, і все буде добре.

15-22 бали - у Вас середній показник колективізму. Ви в міру дипломатичні і доброзичливі, але до роботи не відчуваєте підвищеного інтересу. Для вас робота не спосіб самореалізації, а можливість заробити грошей на життя. Ви - середньостатистичний працівник.

10-14 балів - у Вас низький показник колективізму. Довго на роботі такі особи не тримаються. У соціум Ви вливатися не бажаєте, колег не любите, шефа не поважаєте. З таким ставленням до життя вижити важко. Вам залишається тільки ... чекати звільнення або змінюватися.

РОЗРОБКА ПРОГРАМИ

Після формування чіткого завдання на розробку та створення бота у **Telegram** перейшли до реалізації системи. Для розробки були обрані текстовий редактор **VS Code** і **Python**. Створено папку проекту **TelegramBot**. Через **VS Code** відкриваємо цю папку та додаємо текстовий файл **main.py** за допомогою **code .**

Тепер нам необхідно встановити всі необхідні бібліотеки, об'єднані як функції **requirements.txt**. Переконаємося, що встановлено необхідні у **VS Code**, та через консоль гетього середовища **pip install** встановимо всі бібліотеки.

pip install telegram,urllib3

На наступних слайдах будуть показані фрагменти кодової програми.

Бібліотеки та запуск

```
from telegram import Bot, Update
from telegram.ext import Updater, CommandHandler, MessageHandler, Filters
import logging

# Enable logging
logging.basicConfig(format='%(asctime)s - %(message)s', level=logging.INFO)

# Create the bot
bot = Bot('123456789:ABCDEFghijklmnopqrstuvwxyz0123456789')

# Define a handler for the /start command
def start(update: Update, context: ContextTypes.DEFAULT_TYPE) -> None:
    """Send a message when the user /starts"""
    context.bot.sendMessage(update.message.chat_id, 'Hi!')

# Define a handler for the /help command
def help(update: Update, context: ContextTypes.DEFAULT_TYPE) -> None:
    """Send a message when the user /helps"""
    context.bot.sendMessage(update.message.chat_id, 'Help!')

# Add the handlers to the dispatcher
dispatcher = Updater(bot, use_context=True)
dispatcher.add_handler(CommandHandler('start', start))
dispatcher.add_handler(CommandHandler('help', help))

# Start the bot
dispatcher.start_polling()
```

Обробники команд

```
from telegram import Bot, Update
from telegram.ext import Updater, CommandHandler, MessageHandler, Filters
import logging

# Enable logging
logging.basicConfig(format='%(asctime)s - %(message)s', level=logging.INFO)

# Create the bot
bot = Bot('123456789:ABCDEFghijklmnopqrstuvwxyz0123456789')

# Define a handler for the /start command
def start(update: Update, context: ContextTypes.DEFAULT_TYPE) -> None:
    """Send a message when the user /starts"""
    context.bot.sendMessage(update.message.chat_id, 'Hi!')

# Define a handler for the /help command
def help(update: Update, context: ContextTypes.DEFAULT_TYPE) -> None:
    """Send a message when the user /helps"""
    context.bot.sendMessage(update.message.chat_id, 'Help!')

# Add the handlers to the dispatcher
dispatcher = Updater(bot, use_context=True)
dispatcher.add_handler(CommandHandler('start', start))
dispatcher.add_handler(CommandHandler('help', help))

# Start the bot
dispatcher.start_polling()
```

Отримання та відправка тестів

```
from telegram import Bot, Update
from telegram.ext import Updater, CommandHandler, MessageHandler, Filters
import logging

# Enable logging
logging.basicConfig(format='%(asctime)s - %(message)s', level=logging.INFO)

# Create the bot
bot = Bot('123456789:ABCDEFghijklmnopqrstuvwxyz0123456789')

# Define a handler for the /start command
def start(update: Update, context: ContextTypes.DEFAULT_TYPE) -> None:
    """Send a message when the user /starts"""
    context.bot.sendMessage(update.message.chat_id, 'Hi!')

# Define a handler for the /help command
def help(update: Update, context: ContextTypes.DEFAULT_TYPE) -> None:
    """Send a message when the user /helps"""
    context.bot.sendMessage(update.message.chat_id, 'Help!')

# Add the handlers to the dispatcher
dispatcher = Updater(bot, use_context=True)
dispatcher.add_handler(CommandHandler('start', start))
dispatcher.add_handler(CommandHandler('help', help))

# Start the bot
dispatcher.start_polling()
```

Інтеракція з користувачем

```
from telegram import Bot, Update
from telegram.ext import Updater, CommandHandler, MessageHandler, Filters
import logging

# Enable logging
logging.basicConfig(format='%(asctime)s - %(message)s', level=logging.INFO)

# Create the bot
bot = Bot('123456789:ABCDEFghijklmnopqrstuvwxyz0123456789')

# Define a handler for the /start command
def start(update: Update, context: ContextTypes.DEFAULT_TYPE) -> None:
    """Send a message when the user /starts"""
    context.bot.sendMessage(update.message.chat_id, 'Hi!')

# Define a handler for the /help command
def help(update: Update, context: ContextTypes.DEFAULT_TYPE) -> None:
    """Send a message when the user /helps"""
    context.bot.sendMessage(update.message.chat_id, 'Help!')

# Add the handlers to the dispatcher
dispatcher = Updater(bot, use_context=True)
dispatcher.add_handler(CommandHandler('start', start))
dispatcher.add_handler(CommandHandler('help', help))

# Start the bot
dispatcher.start_polling()
```

ОТЖЕ, ПЕРЕЙДЕМО БЕЗПОСЕРЕДНЬО ДО САМОГО ЧАТ-БОТУ У TELEGRAM

• Щоб знайти бота у коду у **Telegram** необхідно ввести - **@TG_TeleBot**



ДЯКУЮ ЗА УВАГУ! ВСЕ БУДЕ УКРАЇНА!



РЕЦЕНЗІЯ

на дипломний проект (роботу) здобувача (здобувачки) освіти
відділення комп'ютерних систем

Гаврилюка Олега Сергійовича

(прізвище, ім'я та по батькові)

Спеціальність **123 "Комп'ютерна інженерія"**

Освітня програма **Обслуговування комп'ютерних систем та мереж**

Керівник дипломного проекту (роботи) **Скорнякова Олена Володимирівна**

(прізвище, ім'я та по батькові)

Тема дипломного проекту (роботи):

**Створення чат-бота для тестування контингенту здобувачів освіти відділення
комп'ютерних систем**

Обсяг розрахунково-пояснювальної записки _____ сторінок

Обсяг графічної (презентаційної) частини _____ аркушів (слайдів)

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ)

а) **заключення про ступінь відповідності виконаного дипломного проекту (роботи) завданню**

Дипломний проект повністю відповідає завданню до дипломного проектування

б) характеристика виконання кожного розділу дипломного проекту (роботи) _____

Пояснювальна записка дипломного проекту виконана якісно, у повному обсязі. В дипломному проекті здобувачем проведено детальний аналіз існуючих рішень. Конкретизовано на основі проведеного аналізу вимоги до дипломного проекту, визначено завдання та технології, технічні рішення, що дозволяють реалізувати завдання дипломного проекту.

в) оцінка якості виконання пояснювальної записки та графічної частини дипломного проекту (роботи) _____

Презентаційні матеріали виконані якісно, демонстративно та відповідають вмісту теоретичного матеріалу

г) перелік позитивних якостей дипломного проекту (роботи) _____

Тематика дипломного проекту є актуальною. Серед позитивних якостей – детальний аналітичний огляд існуючих рішень, виважений підхід до реалізації завдань до дипломного проекту

д) основні недоліки дипломного проекту (роботи) _____

Серед недоліків роботи варто вказати невеликий перелік використаних літературних джерел. Варто було б в пояснювальній записці представити алгоритм обробки запитів від користувачів чат-бота

Оцінка розрахункової частини _____ *Відмінно*

Оцінка графічної частини _____ *Відмінно*

Загальна оцінка _____ *Відмінно*

Прізвище, ім'я, по батькові рецензента _____ **Васіліу Євген Вікторович**

Місце роботи і посада рецензента **Державний університет інтелектуальних технологій і зв'язку, д.т.н., проф. кафедри КБ та ТЗІ, декан факультету інформаційних технологій та кібербезпеки**

Підпис: _____ *[Signature]*

« 16 » _____ *серпень* 2023 р.



ПІДПИС ПОСВІДЧЕННЯ
НАЧАЛЬНИК ВІДДІЛУ
КАДРІВ ДУІТЗ

ВСП «Одеський технічний фаховий коледж ОНТУ»

ВІДГУК

Керівника на дипломний проект здобувача освіти

Гаврилюка Олега Сергійовича

(прізвище, ім'я та по батькові)

Спеціальність: 123 «Комп'ютерна інженерія»

Тема дипломного проекту: _____

Створення чат-бота для тестування контингенту здобувачів освіти відділення комп'ютерних систем

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ

а) Обсяг і якість виконання проекту (графічного матеріалу і розрахунково-пояснювальної записки): Пояснювальна записка дипломного проекту виконана якісно, у повному обсязі. В дипломному проекті здобувачем проведено огляд існуючих технологій та аналогів програмного продукту (чат-боту), що пропонується до створення, визначено структуру, розроблено програмну частину системи. Представлено функціонуючий програмний продукт, що дозволяє проводити тестування здобувачів. В дипломному проекті в останніх розділах проаналізовано питання економічної доцільності та охорони праці. Створено презентацію до захисту.

б) Самостійність роботи над проектом: Здобувач самостійно визначався з напрямом роботи, дослухався до рекомендацій керівника дипломного проекту, своєчасно надавав результати роботи, якісно виконував основні етапи роботи за вимогою керівника.

в) Теоретична підготовка випускника: _____

Теоретична підготовка випускника в цілому відповідає державним вимогам до фахівців відповідного рівня кваліфікації

г) Вміння розв'язувати виробничі і конструкторські питання на базі останніх досліджень науки і техніки, передових методів виробництва _____

В процесі роботи над дипломним проектом здобувач продемонстрував уміння використовувати останні досягнення науки та техніки в предметній галузі на підставі відповідної навчальної та науково-технічної літератури, достатньо впевнено користувався програмним забезпеченням при роботі над дипломним проектом та створенням презентації.

Оцінка розрахункової частини _____ відмінно _____

Оцінка графічної частини _____ відмінно _____

Загальна оцінка _____ відмінно _____

Прізвище, ім'я, по батькові Скорнякова Олена Володимирівна _____

Місце роботи і посада керівника дипломного проекту: к.пед.н., викладач-методист комісії КТ та ПІ ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» _____

Підпис _____
«12» _____ 2023 р.

**ДОЗВІЛ
НА РОЗМІЩЕННЯ
ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
В ЕЛЕКТРОННОМУ РЕПОЗИТАРІЇ ВСП «ОТФК ОНТУ»**

Ми, що нижче підписалися,

Гаврилюк Олег Сергійович,
здобувач освіти гр. 4КС-56, та

Скорнякова Олена Володимирівна,
керівник дипломного проекту,

не заперечуємо щодо розміщення електронного варіанту пояснювальної записки до випускної кваліфікаційної роботи молодшого спеціаліста на тему:

«Створення чат-бота для тестування контингенту здобувачів освіти відділення комп'ютерних систем» (автор роботи – Гаврилюк О.С., керівник роботи – Скорнякова О.В.)

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2023 році, у повному обсязі в електронному репозитарії ВСП «ОТФК ОНТУ» для вільного доступу через мережу Інтернет.

Несемо відповідальність за ідентичність електронного та друкованого варіантів випускної кваліфікаційної роботи, і даємо згоду на обробку персональних даних.

Виконавець



/ Гаврилюк О.С./

Керівник



/ Скорнякова О.В./

« 12 » червня 20 23 р.

Ім'я користувача:
Наталія Вікторівна Копусь

ID перевірки:
1015543579

Дата перевірки:
10.06.2023 18:53:26 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
10.06.2023 18:53:54 EEST

ID користувача:
100011688

Назва документа: 4КС-56 Гаврилюк Олег

Кількість сторінок: 47 Кількість слів: 8628 Кількість символів: 61757 Розмір файлу: 1.96 MB ID файлу: 1015196243

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

36% Схожість

Найбільша схожість: 14.4% з Інтернет-джерелом (<https://dspace.nau.edu.ua/bitstream/NAU/52055/1/%d0%a4%d0%9a%..>)

36% Джерела з Інтернету

1000

Сторінка 49

Не знайдено джерел з Бібліотеки

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

32

Підозріле форматування

8
сторінок