

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет кібербезпеки, комп'ютерної та програмної інженерії
Кафедра комп'ютерних інформаційних технологій

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри
_____ А.С. Савченко
« ____ » _____ 20 ____ р

ДИПЛОМНИЙ ПРОЕКТ

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СПУПЕНЯ «БАКАЛАВР»

Тема: «Чат-бот on-line співбесіди»

Виконавець: студентка УС-412 Козіна Юлія Володимирівна
(студент, група, прізвище, ім'я, по батькові)

Керівник: к. т. н., доцент Колісник Олена Василівна
(науковий ступень, вчене звання, прізвище, ім'я, по батькові)

Нормоконтролер: ст. викл. Шевченко О.П.
(П.І.Б.) (підпис)

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра комп'ютерних інформаційних технологій

Освітній ступінь: Бакалавр

Галузь знань, спеціальність, спеціалізація: 12 “Інформаційні технології”,
122 “Комп'ютерні науки”, “Інформаційні управляючі системи та технології”

ЗАТВЕРДЖУЮ

Завідувач кафедри

А.С. Савченко

“_____” _____ 2021 р.

ЗАВДАННЯ

на виконання дипломного проекту студента

Козіна Юлія Володимирівна

(прізвище, ім'я, по батькові)

1. Тема проекту: «Чат-бот on-line співбесіди» затверджена наказом ректора № 636/ст. від 22.04.2021р.

2. Термін виконання роботи: з 10.05.2021 по 20.06.2021р.

3. Вихідні дані до роботи: розробка та написання програмного коду для додатку Telegram за допомогою мови програмування Python – чат-бота для інформування кандидатів та проходження онлайн співбесіди.

4. Зміст пояснювальної записки (перелік питань, що підлягають розробці): вступ, огляд предметного середовища і аналогів, визначення призначення чат-бота та його функцій; вибір засобів розробки та їх характеристики; налаштування та розробка чат-боту, підключення бази даних; висновки.

5. Перелік обов'язкового графічного матеріалу: слайди презентації доповіді у PowerPoint.

КАЛЕНДАРНИЙ ПЛАН

	Етапи виконання дипломної роботи	Термін виконання етапів	Примітка
1	Огляд предметного середовища та аналіз аналогів. Написання 1 розділу та узгодження з керівником.	10.04.21 – 17.04.21	
2	Визначення мети і функціоналу чат-бота; аналіз і вибір засобів розробки. Написання 2 розділу та узгодження з керівником.	18.04.21 – 28.04.21	
3	Розробка чат-боту, перевірка функціонування. Написання 3 розділу та узгодження з керівником.	29.04.21 – 19.05.21	
4	Внесення правок та редагування	20.05.21 – 23.05.21	
5	Завершення оформлення пояснювальної записки	24.05.21	
6	Проходження перевірки на атиплагіат та нормоконтроль	25.05.21 – 31.05.21	
7	Підготовка презентації, доповіді	01.05.21 – 05.05.21	
8	Підготовка до захисту та передзахист дипломного проекту	06.06.21 – 14.06.21	

Студент

(*Козіна Ю.В.*)

Керівник дипломної роботи

(*Колісник О.В.*)

РЕФЕРАТ

Пояснювальна записка до дипломного проекту «Чат-бот on-line співбесіди» містить: 60 сторінок, 52 рисунки, 3 таблиці, 12 літературних джерел.

Об'єкт дослідження: чат-бот для маркетингового агентства “Restoria”.

Предмет дослідження: система для маркетингового агентства “Restoria” для проведення співбесіди online.

Мета роботи: розробити програму – бот в Telegram для маркетингового агентства “Restoria” для проведення співбесіди online з можливістю запису інформації, що додали користувачі, у базу даних MySQL.

Методи дослідження, технічні та програмні засоби: використання мови програмування Python для написання чат-боту для месенджера Telegram, використовуючи Telegram Bot API та реляційної бази даних MySQL у середовищі PyCharm.

Отримані результати та їх новизна: створено чат-бот для інформування потенційних кандидатів про можливі вакансії та проходження онлайн співбесіди. Продукт є актуальним на сьогоднішній день – більшість сфер робіт переходять на віддалене співробітництво.

МЕСЕНДЖЕР, ДОДАТОК, РОЗРОБКА, ЧАТ-БОТ, СПІВБЕСІДА, PYTHON, БАЗА ДАНИХ, MYSQL, TELEGRAM API, PYCHARM.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ.....	6
ВСТУП.....	7
РОЗДІЛ 1 ОГЛЯД ПРЕДМЕТНОГО СЕРЕДОВИЩА.....	9
1.1. Месенджер Telegram	9
1.2. Основна інформація про чат-боти	12
1.3. Аналіз діючих чат-ботів.....	15
1.3.1. OpenDataUa	16
1.3.2. SaveEcoBot	17
1.3.3. Andy English Bot	18
1.3.4. Skeddy	19
РОЗДІЛ 2 ПЛАНУВАННЯ ЧАТ-БОТА.....	21
2.1. Задача чат-бота.....	21
2.2. Вимоги до чат-бота.....	22
2.3. Створення функціоналу чат-бота.....	22
2.4. Характеристика обраних засобів розробки.....	24
2.4.1. Мова програмування Python.....	24
2.4.2. Документація Telegram Api.....	27
2.5. Характеристика середовища розробки Pycharm.....	31
2.6. Характеристика бази даних MySQL	32
РОЗДІЛ 3 СТВОРЕННЯ ЧАТ-БОТА	34
3.1. Налаштування чат-бота.....	34
3.2. Розробка програмної частини.....	35
3.2.1. Створення вітального повідомлення та головного меню.....	37
3.3. Приєднання бази даних	53
ВИСНОВКИ.....	56
СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ.....	57

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

API	Application programming interface
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
JSON	JavaScript Object Notation
MTProto	Mobile Telegram Protocol
PEP	Python Enhancement Proposal
PHP	Hypertext Preprocessor
URL	Uniform Resource Locator
БД	База даних
СУБД	Система управління базами даних
IT	Інформаційні технології

ВСТУП

У сучасному світі повсякденне життя людини тісно пов'язано з комп'ютерними технологіями. Ось чому все популярнішими стають соціальні мережі та користуються великим попитом месенджери. Одним з ведучих - вважається саме Telegram. Тому що він поєднує в собі такі характеристики як зручність, простота і надійність у використанні, сучасні налаштування для вільного та безпечного спілкування - що найчастіше вибирають користувачі. Одним з головних інструментів Telegram є якраз чат-боти, за допомогою яких здійснюється автоматизація рутинних процесів. Дана тематика цілком актуальна і користується великим попитом, так як щодня кожна людина проводить досить великий час в інтернеті, зокрема соціальних мережах і месенджерах. Найчастіше людині зручніше спілкуватися за допомогою повідомлень, ніж дзвінків. Це пояснює активне користування чат-ботами, за допомогою яких ми можемо бронювати / купувати квитки на транспорт, замовляти страви із закладів, читати / дивитися новини, тестувати свої знання і багато іншого саме онлайн.

Так як на сьогоднішній день більшість сфер життя перейшли в дистанційний режим, в зв'язку зі сформованою ситуацією, то і проведення співбесід є актуальними в такому форматі. Також Telegram широко застосовується різними агенціями, компаніями, організаціями для пошуку співробітників, кандидатів, виконавців роботи і незалежно від того, в якому місті чи навіть країні знаходиться людина. Це ще більше підвищує попит на проведення ділових зустрічей і онлайн співбесід за допомогою месенджера. Одним з установ, яка була затребувана в організації такого завдання, стало маркетингове агентство «Restoria».

Мета дипломного проекту - створити чат-бот в месенджері Telegram для маркетингового агентства «Restoria», за допомогою якого будуть здійснюватися онлайн співбесіди.

Переваги такого чат-бота в тому, що онлайн співбесіда допоможе «відсіяти» невідповідних кандидатів при цьому не витрачаючи час на них. Разом з тим співробітники агентства зможуть сфокусувати основну увагу і приділити більше часу тієї кандидатурі, яка пройшла перший етап на онлайн співбесіді. Це підвищує

продуктивність пошуку та відбору відповідних вимогам виконавців. В наслідок чого, обрана тема для реалізації чат-бота - досить актуальна.

На підставі поставленої задачі визначені наступні завдання :

- зробити огляд обраного предметного середовища;
- проаналізувати діючих чат-ботів;
- підібрати необхідні технології для розробки;
- створити та перевірити роботу чат-бота в Telegram.

РОЗДІЛ 1

ОГЛЯД ПРЕДМЕТНОГО СЕРЕДОВИЩА

1.1. Месенджер Telegram

Telegram – це безкоштовний багатоплатформовий додаток, що написаний на мовах програмування C++, Java. Месенджер надає можливість користувачам спілкуватися за допомогою повідомлень або дзвінків, відправляти файли різних форматів, створювати ботів, вести власні канали та інше. Крім того, в хмарному сховищі Telegram Passport можна зберігати різні документи, що містять персональні дані клієнта. Розробники гарантують високу ступінь безпеки, тому як навіть сам сервіс не має до них доступу.

Телеграм випущений 14 серпня 2013 року. За зразковими розрахунками кількість користувачів на 2021 рік сягає більш ніж 500 млн. Додаток продовжує розвиватися і вдосконалюватися, впроваджувати нові опції.

Месенджер лідирує серед популярних суперників завдяки таким перевагам:

- Велика швидкість роботи та миттєва синхронізація.
- Відмінна робота технічної підтримки месенджера.
- Зручність користування та відсутність реклами.

Розглянемо більш детально ще одну велику перевагу та особливість Telegram – високий рівень безпеки. Унікальність цього додатку це секретні чати і протокол шифрування MTProto. Якщо вам необхідно захиститися від витoku інформації або перехоплення повідомлень з певного листування, то саме секретні чати будуть дуже корисними. Можливості секретного чату:

Кафедра КІТ (47)				НАУ 21 33 93 000 ПЗ			
Виконав	Козіна Ю.В.			Огляд предметного середовища	Літера	Аркуш	Аркушів
Керівник	Савченко А.С.					9	12
Консульт.	Колісник О.В.				412 122		
Н-котрол.	Шевченко О.П.						
Зав. каф.	Савченко А.С.						

- історія повідомлень зберігається не в хмарі, а на пристроях учасників листування;
- таймер використовують для видалення повідомлень або файлів через встановлений проміжок часу. Видалення відбувається назавжди на стороні відправника та одержувача.

Telegram використовує криптографічний протокол MTProto для шифрування листування користувачів (рис. 1.1). Незважаючи на це, всі повідомлення шифруються за допомогою ключів з використанням принципу «від пристрою до пристрою» - в цьому випадку унеможлиблює перехоплення повідомлень, але ймовірність витoku конфіденційної інформації можливо шляхом крадіжки пристрою.

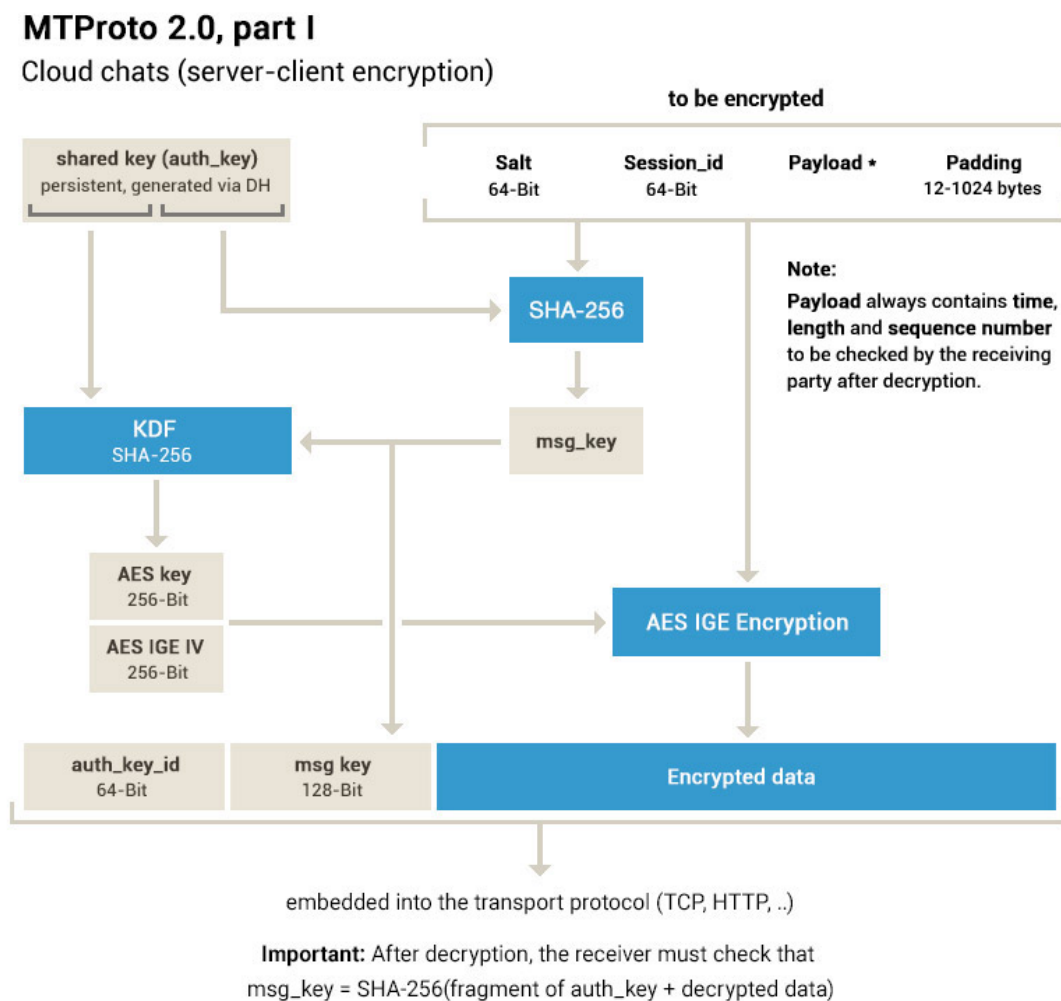


Рис. 1.1. Схема роботи алгоритму шифрування

Месенджер доступний для таких платформ, як Android, iOS, Windows Phone, Windows, macOS и GNU/Linux. Функціонал у всіх версіях аналогічний і не має обмежень. Інтерфейс додатку підтримує 19 мов, найпоширеніші з яких це англійська, німецька, російська та українська (рис. 1.2. та рис. 1.3.).

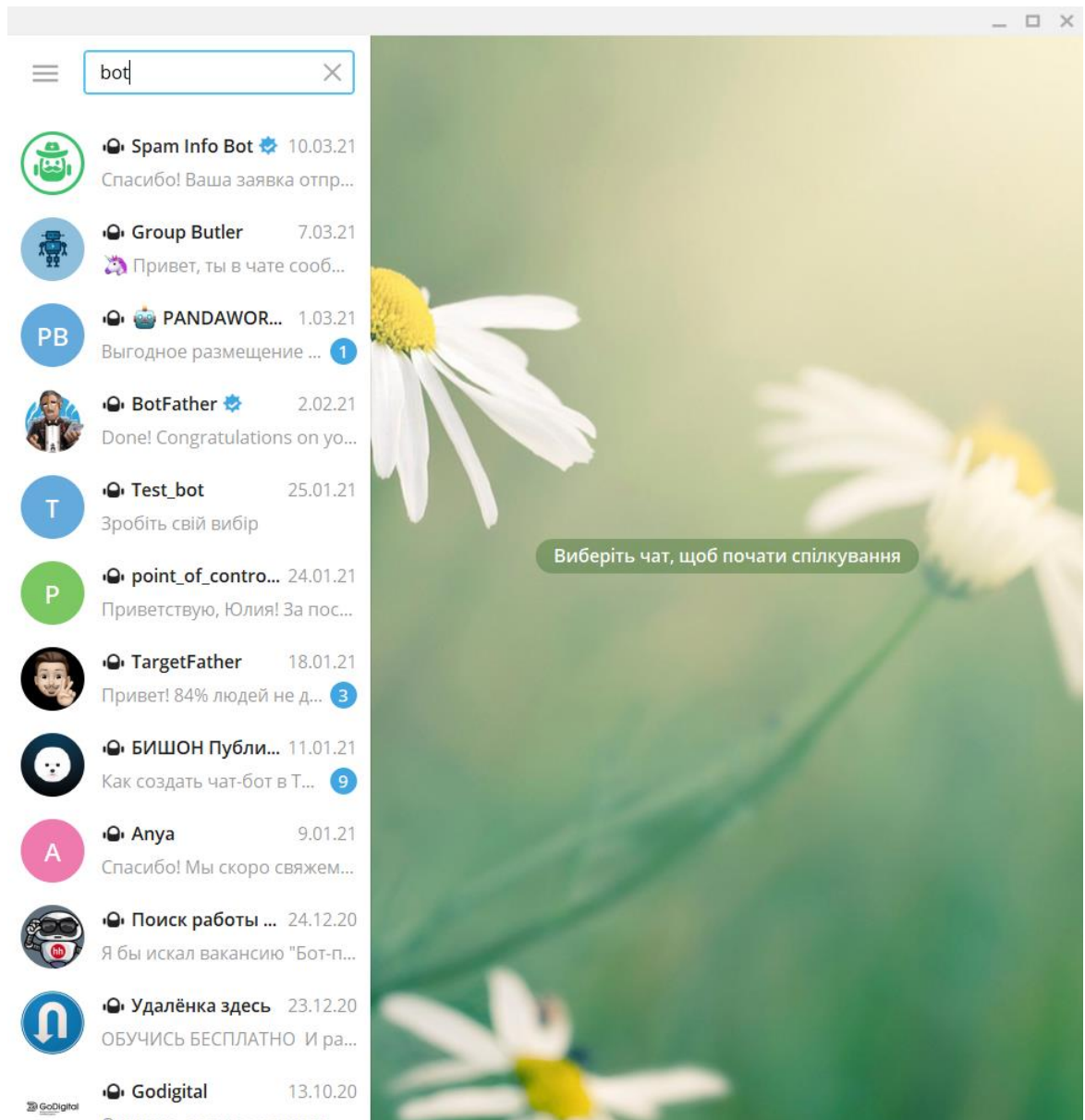
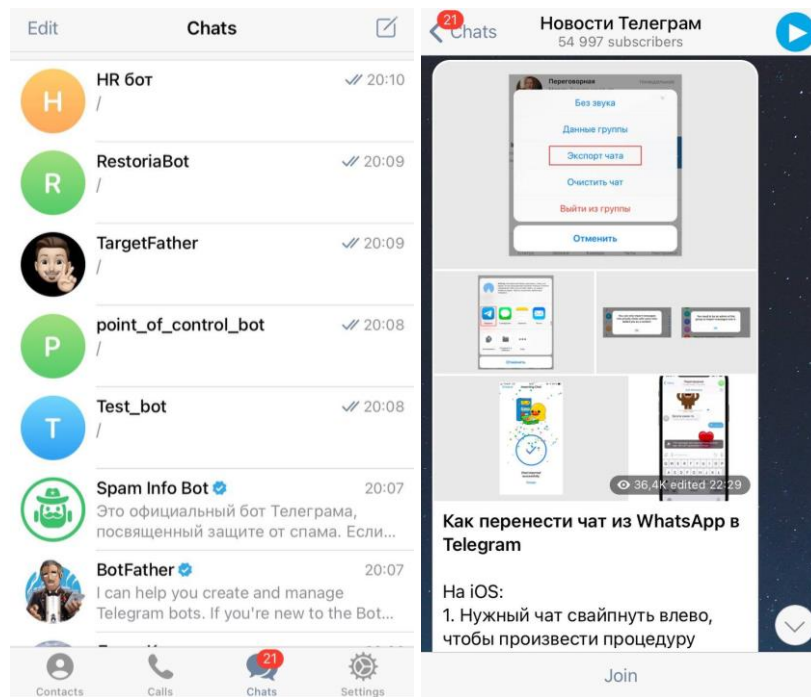


Рис.1.2. Telegram українською мовою Windows 10



а)

б)

Рис.1.3. Telegram iOS: а) чати; б) канал

1.2. Основна інформація про чат-боти

З кожним роком чат-боти стають дедалі популярнішими і користуються великим попитом. Вони є невід'ємною частиною повсякденності і дуже затребувані в багатьох і абсолютно різних сферах – від банківських структур до інтернет магазинів. Ботів можна зустріти або налаштувати в месенджерах і соцмережах (Telegram, Viber, Facebook), на сайтах, в мобільних додатках, системах управління (Siri), голосових асистентів (Аліса) і так далі. Давайте розберемо, що ж таке чат-боти і як вони функціонують.

Чат-бот - це спеціальна програма, в корені якої прописаний алгоритм, що дозволяє автоматично обробляти повідомлення від користувачів та надавати коректні й логічні відповіді. В основі механізму дії розташовується завчасно прописаний сценарій спілкування, що і дозволяє швидко реагувати на дії співрозмовника. Тобто, в чат-ботах користувач спілкується з роботом, який замінює реальну людину.

При створенні чат-ботів вибір типу має ключове значення. Існує два типи ботів: ускладнені та прості. Ускладнені чат-боти працюють на основі штучного інтелекту,

завдяки чому вони можуть розпізнавати питання будь-якої складності формулювання і на різних мовах. Прості - функціонують, дотримуючись своїх інструкцій: вони можуть надавати відповідь лише на ті питання, які вже прописані в системі. Оскільки в месенджерах, зокрема Telegram, використовуються прості боти, тому саме цей тип ми будемо розбирати.

Щоб зробити вмотивований вибір методології розробки програмного забезпечення, потрібно розглянути переваги та недоліки провідних чат-ботів.

Перевагами є:

- цілодобова можливість та доступність для користування;
- миттєве отримання відповіді;
- інтерфейс легкодоступний для розуміння та використання;
- висока безпека персональних даних;
- підтримка роботи з декількома користувачами одночасно;
- оптимізація процесів;
- зниження операційних витрат;
- зручні для клієнтів (користувачів);
- автоматизація простих комунікацій.

Як було зазначено раніше, чат-боти використовуються в різноманітних галузях, тому що вони підвищують продуктивність та якість надання послуг. Найчастіше використання ботів можна зустріти в наступних сферах:

- Бізнес. Можливості чат-боту в цій галузі залежать від виду діяльності підприємства. Наприклад, якщо розглядати бот для служби доставки їжі, то доступна функція замовлення страв; якщо це інтернет-магазин – користувачі можуть отримувати інформацію щодо знижок і так далі.

- Навчання. В даній сфері боти можуть проводити покрокові навчання або різноманітні тестування для визначення рівня знань, наприклад, англійської мови та інше.

- Розваги. З чат-ботом можна грати, наприклад, в інтелектуальні ігри, проходити вікторини, читати анекдоти та багато інших розваг.

- Новини. В цій галузі боти можуть надавати будь-яку інформацію в залежності від специфіки. Наприклад, ми можемо щодня отримувати інформацію про курс валюти або дізнаватися останні новини спорту в світі тощо.

- Банківська сфера. В Україні понад 38% банків використовують чат-боти. Можливості: керування картою, зв'язок з банком, фінансові операції та інше.

- Страхові послуги. Бот цієї сфери може авторизувати користувачів, розсилати повідомлення, організовувати документообіг та інше.

- Туризм. У користувача є можливість отримувати розсилку турів, бронювати квитки тощо.

Із перерахованого вище можемо зробити висновок, що використання ботів в будь-яких сферах життя не обмежено і з кожним роком їх функціонал розширюється, а потреба зростає.

Отже, боти в Telegram – це програми, що представляють собою стандартні акаунти та можуть виконувати різні функції, імітуючи дії людини. Головною задачею виступає автоматична та швидка відповідь після введення і відправки команди користувачем. Боти допомагають виконувати елементарні рутинні задачі, які витрачають багато часу. Також з їх допомогою існує можливість замовляти квитки, шукати корисну інформацію, слухати музику і багато іншого.

Чат-бот являє собою простий чат, який від звичайних користувачів можливо відрізнити за такими ознаками:

- в імені бота обов'язково міститься приставка "bot";
- при відкритті чату замість поля для повідомлення знаходиться напис «Запустити»;
- у бота не може бути статусу, коли він був в мережі, ця функція доступна реальному користувачеві.

Для того, щоб почати користуватись віртуальним товаришем необхідно відкрити чат та натиснути кнопку «Почати», яка знаходиться замість поля для введення повідомлення. Існує ще один спосіб запуску чат-боту - прописати команду «/start», але тільки в тому випадку, якщо Telegram бот попередньо був включений, або були проблеми з сервером, або раніше ним вже користувалися. Далі програма

запропонує варіанти можливих команд, які користувач повинен вибрати і натиснути на кнопку. Зазвичай використання чат-бота не доставляє труднощів, так як функціонал інтуїтивно зрозумілий навіть початківцю. У будь-якому випадку бот завжди видає підказки і пропонує наступні кроки.

Спілкуватися з ботом можна шляхом ручного введення даних або за допомогою існуючих кнопок, при натисканні яких користувач відправляє відповідну команду боту (рис.1.4).

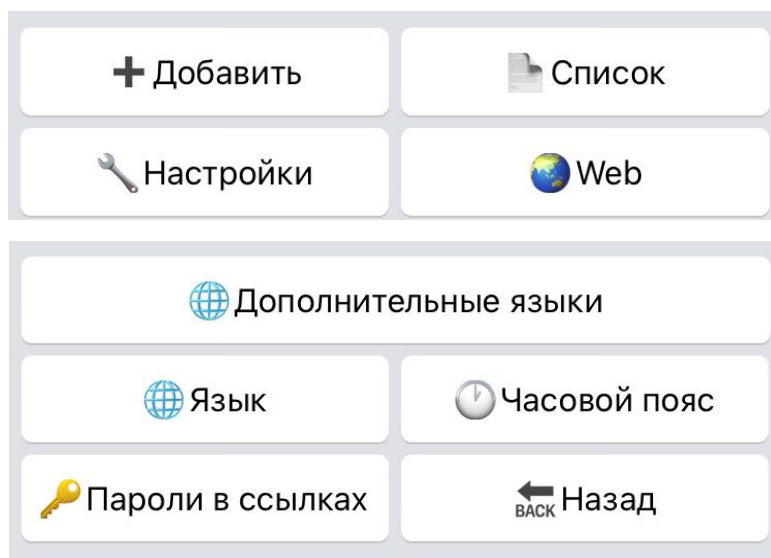


Рис.1.4. Приклад меню кнопок чат-бота

Для пошуку *Chatbot* існує два способи:

1. Пошук за назвою. Даний спосіб використовується, в тому випадку, якщо визначена мета, тобто який бот необхідний.
2. Пошук в списку або каталозі з повним описом функціоналу і вибрати найбільш відповідного.

1.3. Аналіз діючих чат-ботів

Запорука успішного створення продукту - це якісне дослідження ринку і аналіз конкурентів. Тому, перш ніж приступити до проектування та розробки бота для поточного дипломного проекту, розглянемо вже існуючі програми. Це допоможе

взяти до уваги всі недоліки і переваги діючих чат-ботів, не допустити їх помилок і створити досить оригінального бота з широким функціоналом. Для того, щоб визначити унікальність ідеї чат-бота для проходження онлайн співбесіди, був виконаний запит в мережі інтернет з метою знаходження схожих ботів. В результаті точні аналоги не знайдені, так як дана тематика чат-ботів продовжує розвиватися і на даний момент важко повністю довірити проведення онлайн співбесіди саме роботу. Розглянемо деякі популярні чат-боти месенджера Telegram.

1.3.1. OpenDataUa

OpenDataUa – це український чат-бот, який шукає інформацію в державних базах даних про людей, компанії, авто, нерухомості та видає абсолютно законний доступ до відкритої інформації (рис.1.5, рис.1.6). Цей бот підійде тим, кому для роботи необхідно перевіряти певні дані про різні компанії, фахівців тощо. Також є можливість достатньо швидко перевірити судимість або дію ліцензії, наприклад, адвоката.



Рис.1.5. Відомості бота OpenDataUa

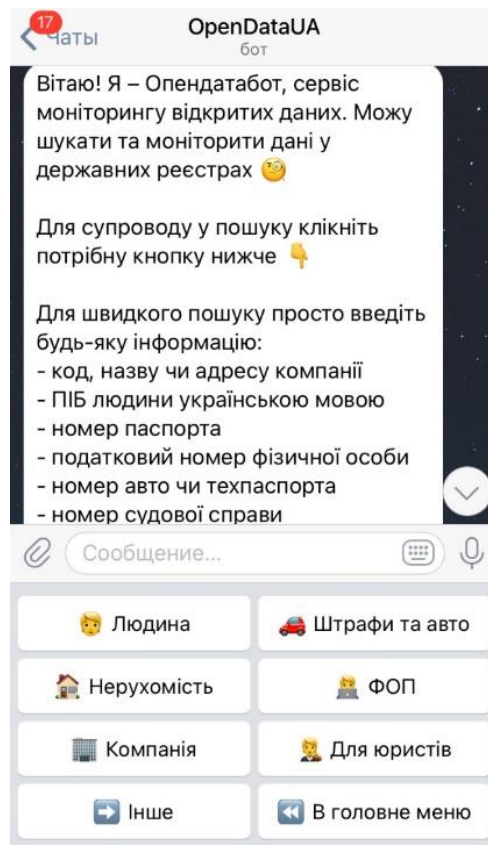


Рис.1.6. Вигляд бота OpenDataUa

Чат-бот почав свою роботу в квітні 2016 року і функціонує не тільки в телеграм, але і в багатьох інших популярних соціальних мережах (Viber, Skype та ін.). За останньою інформацією кількість користувачів сягає більш ніж 800 000. Основними користувачами OpenDataUa є такі фахівці, як юристи, журналісти та підприємці.

1.3.2. SaveEcoBot

SaveEcoBot – це чат-бот, який відправляє інформацію щодо чистоти повітря в багатьох містах України (рис.1.7, рис.1.8). Крім того він володіє відомостями відносно підприємств, які забруднюють екологію. Метою даного бота є звернути увагу жителів країни на руйнування навколишнього середовища і почати вживати заходи для поліпшення екологічних складових.



Привіт! Я – екологічний чат-бот. І я маю інформацію задля твого захисту! 🍀

Потрібна допомога? – пиши на [@pavlentij](#).

Рис.1.7. Відомості бота SaveEcoBot

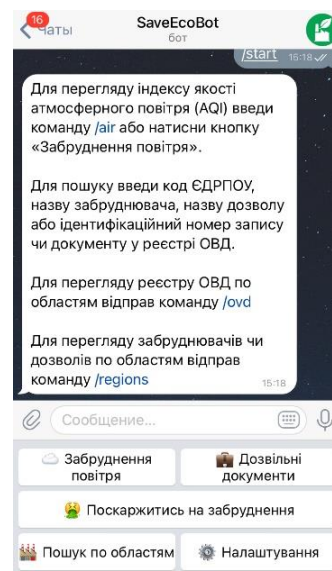


Рис.1.8. Вигляд бота SaveEcoBot

Цей чат-бот почав функціонувати в Telegram 4 липня 2018 року. Також створений для месенджерів: Facebook Messenger, Skype та Viber. На сьогоднішній день кількість користувачів бота становить приблизно 1500.

1.3.3. Andy English Bot

Andy English Bot – чат-бот призначений для вивчення і практики англійської мови (рис.1.9, рис.1.10). Він дозволяє спілкуватися, виконувати вправи, вчити граматику і грати в різні ігри.



Andy English Bot

@andyrobot

Practice English with a friendly robot Andy.
Talk, learn new words, study grammar and
play games!

Рис.1.9. Відомості бота Andy English Bot



Рис.1.10. Вигляд бота Andy English Bot

1.3.4. Skeddy

Skeddy – чат-бот, який обов’язково нагадає про справу, яку необхідно виконати, в заданий час (рис.1.11, рис.1.12). Достатньо прописати повідомлення та обрати час для інформування.



Skeddy

@skeddybot

Skeddy is a simple yet powerful reminder
tool that can help you create and manage
your reminders. <https://skeddy.me/>

Рис.1.11. Відомості бота Skeddy

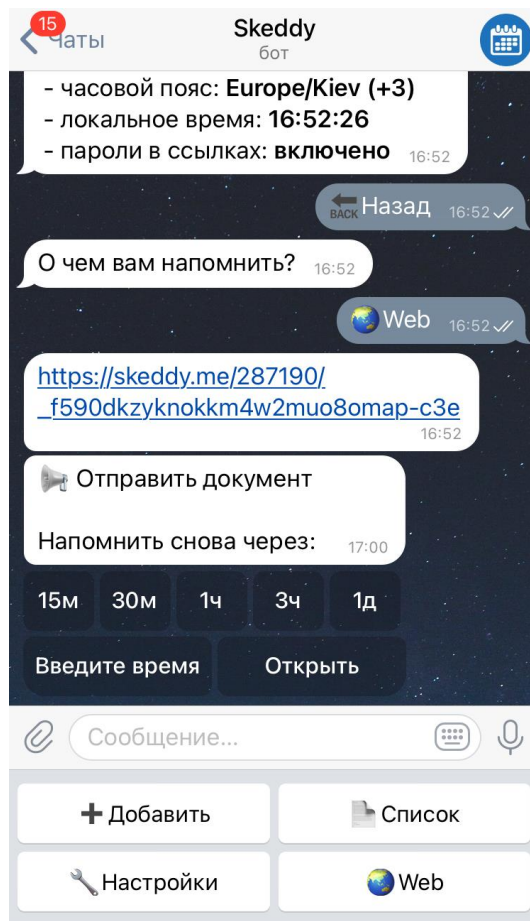


Рис.1.12. Вигляд бота Skeddy

Бот має такі можливості:

- створювати прості одноразові нагадування;
- створювати нагадування, які повторюються через заданий проміжок часу;
- переглядати список нагадувань;
- змінювати нагадування (час, інформація та ін.);
- деактивувати і активувати одні і ті ж нагадування;
- створювати звичайні нотатки без установки часу.

РОЗДІЛ 2

ПЛАНУВАННЯ ЧАТ-БОТА

2.1. Задача чат-бота

В першу чергу необхідно позначити, з якою метою буде проектуватися і розроблятися бот. Припустимо, що замовником чат-бота для проходження онлайн співбесіди виступає реальне маркетингове агентство «Restoria», яке надає послуги в онлайн режимі з просування різних компаній, закладів тощо. Дане агентство користується різними соціальними мережами, в тому числі і месенджером Telegram, для підбору фахівців, підтримки зв'язку з клієнтами та інше. Запит замовника полягав в тому, щоб заощадити час на пошуку відповідних кадрів з будь-якої точки світу у віддаленому форматі. У зв'язку з цим було вирішено створити чат-бот в Telegram.

Задачею чат-бота RestoriaBot є виконання таких функцій:

- надання інформації про агентство «Restoria»;
- інформування потенційних кандидатів про доступні вакансії;
- вибір вакансії відповідно до особистих побажань кандидатів;
- проведення тестування фахівців для визначення рівня кваліфікації;
- запис на співбесіду;
- забезпечення контактними даними кандидатів для зворотнього зв'язку.

Кафедра КІТ (47)				НАУ 21 33 93 000 ПЗ			
Виконав	Козіна Ю.В.			Планування чат-бота	Літера	Аркуш	Аркушів
Керівник	Савченко А.С.					21	13
Консульт.	Колісник О.В.				412 122		
Н-котрол.	Шевченко О.П.						
Зав. каф.	Савченко А.С.						

2.2. Вимоги до чат-бота

Перед тим, як починати розробку і реалізацію будь-якого продукту, в даному випадку чат-бота, необхідно запросити у замовника технічне завдання, яке повинно включати в себе побажання, умови, вимоги до функціональних можливостей майбутнього Telegram боту. У зв'язку з цим розглянемо основні умови функціонування бота, який розроблявся.

Перш за все, RestoriaBot має дотримуватися таких вимог:

- стабільне і якісне функціонування на будь-яких пристроях і версіях додатка Telegram;
- виконання всіх зазначених вище функцій;
- наповнення чат-бота тільки дійсною і достовірною інформацією;
- наявність зручного та інтуїтивно зрозумілого інтерфейсу;
- забезпечення цілісності, надійності та захищеності даних.

Також важливо відзначити, що при появі помилок або збоїв в роботі чат-бота, повинна бути можливість звернутися до адміністратора віртуального товариша та вирішити проблему.

2.3. Створення функціоналу чат-бота

Приступивши до розробки RestoriaBot, враховувалися всі вимоги до чат-боту. В результаті чого, були отримані деякі функціональні властивості. Розглянемо їх:

- простий для розуміння і зручний в експлуатації інтерфейс;
- меню кнопок, яке служить в якості навігації по чат-боту;
- швидкі і логічні відповіді користувачу;
- необмежена кількість запитів.

Не менш важливим є той факт, що чат-бот підтримує свою роботу на будь-якому пристрої з встановленим месенджером Telegram і з підключенням до мережі інтернет.

Для того, щоб запустити чат-бот необхідно прописати команду «/ start». Далі користувач отримає вітальне повідомлення і доступ до головного меню, на якому знаходяться перші кнопки-команди. Розглянемо більш детально функціонал бота, його можливості по відношенню до розділів і підрозділів основного меню.

Перше і головне меню, яке отримає користувач має наступні функції:

1. «Ознайомитися з агентством». Користувач може більш детально дізнатися про агентство (послуги, проекти, відгуки), в яке буде проходити співбесіду.
2. «Доступні вакансії». Бот повідомляє про доступні вакансії та надає інформацію (вимоги, обов'язки, пропозиції) про фахівців, яких шукає компанія. Після кандидат повинен вибрати наступну дію: повернутися назад до вибору вакансій, пройти тестування або повернутися в головне меню.
3. «Тестування на вакансії». Користувач може відразу перейти до тестування на одну із запропонованих вакансій, щоб відповісти на запропоновані питання та підтвердити свою кваліфікацію.
4. «Наші контакти». Функція, яка надає список контактних даних у вигляді посилань на соціальні мережі і особистий сайт.

Для зупинки роботи чат-бота досить вийти з чату або припинити натискати на кнопки-команди. Так само є можливість очистити історію листування з ботом або видалити чат, але ці функції належать вже до самих можливостей месенджера.

Розглянемо загальну структуру RestoriaBot на рис. 2.1. Взаємодія відбувається між користувачем, базою даних і чат-ботом. У кожного актора є свої функціональні можливості:

- користувач, в кінцевому підсумку, може проходити тестування і записуватися на співбесіду;
- в базу даних приходить інформація про кандидата, який записався на співбесіду;
- чат-бот, по суті, має відношення до всіх функцій, але основними є тестування і запис на співбесіду.

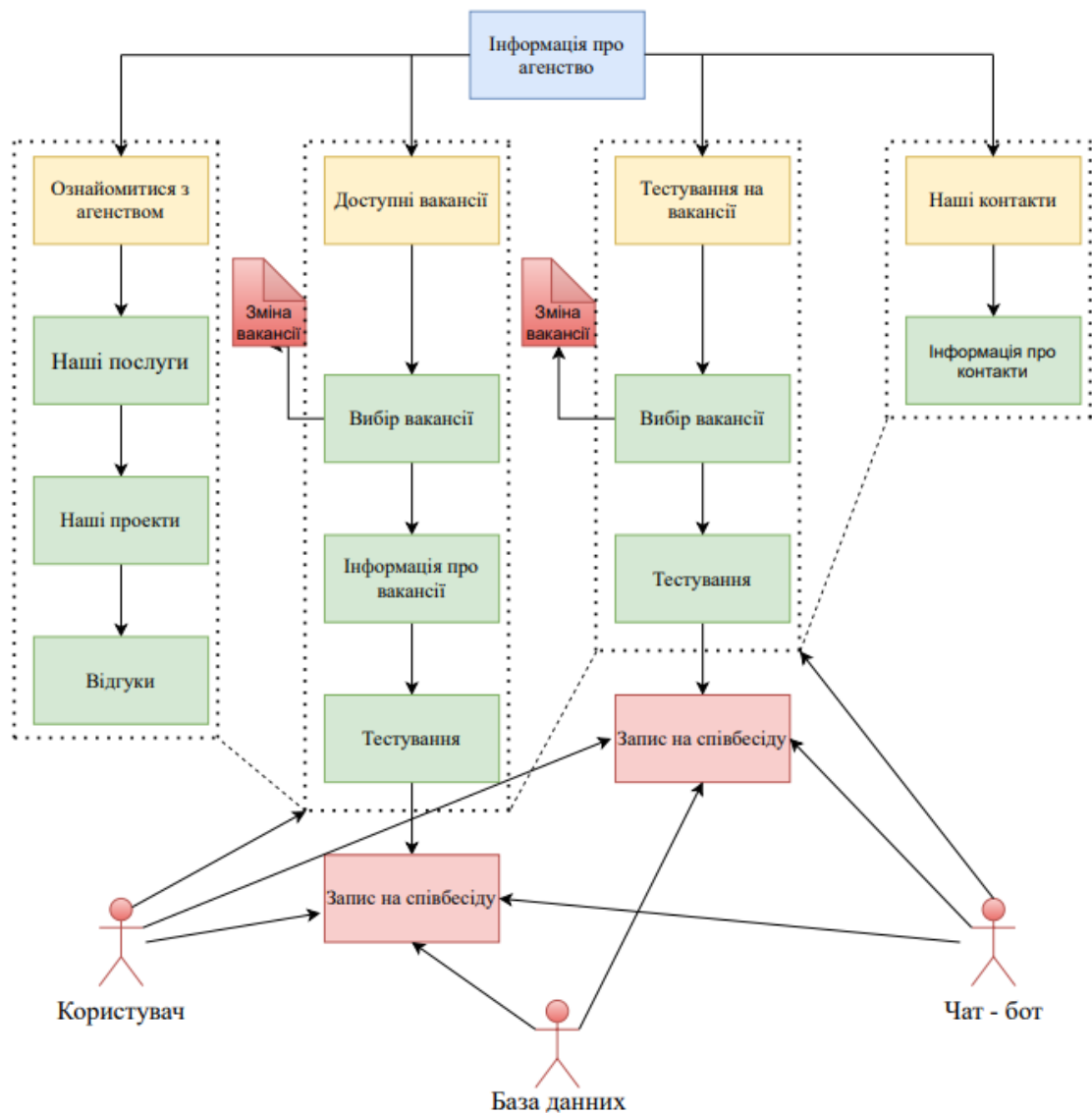


Рис.2.1. Логіка роботи Telegram боту

2.4. Характеристика обраних засобів розробки

2.4.1. Мова програмування Python

При створенні дипломного проекту важливою складовою є якісно підібрані засоби розробки. Внаслідок чого, перед тим, як починати розробку чат-бота, для початку необхідно вибрати і проаналізувати програмну мову.

Telegram ботів можна писати на PHP, Python, Node is, Go, Java. В результаті огляду та порівнянь зазначених мов програмування було визначено, що саме Python найбільш придатний для використання в даному проекті.

Python - це високорівнева, об'єктно-орієнтована, інтерпретована скриптова мова програмування, яка активно розвивається. Python вважається одним з найпопулярніших мов програмування та найчастіше використовується в створенні різних додатків для комп'ютерів і телефонів, при розробці сайтів, в машинному навчанні та багато іншого. Дана мова програмування має логічний і простий синтаксис, відповідно написаний код досить читабельний і легко сприймається користувачами.

Як було зазначено, Python є інтерпретованою мовою. Це означає, що вихідний код компілюється не в машинний, а у внутрішній байт-код та надалі виконується вже інтерпретатором Python (рис. 2.2). Якщо порівнювати з мовами C, то найчастіше на Python програми виконуються повільніше, але швидкість реалізації того чи іншого коду швидше завдяки легкому синтаксису. Формат роботи інтерпретатора - інтерактивний, що дозволяє користувачам вводити команди, які построчно виконуються і після кожної команди інтерпретатор в той же час видає результат. Для того, щоб в консолі отримати результат прописаної команди, необхідно після введення натиснути на клавішу Enter.

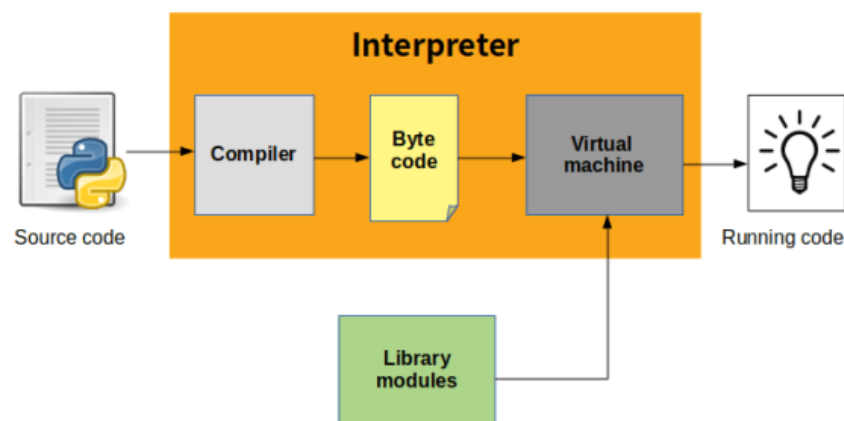


Рис.2.2. Поетапне виконання Python програми інтерпретатором

Вперше мова програмування Python з'явилась в 1991 році, але саму реалізацію Гвідо ван Россум, співробітник голландського Національного дослідницького інституту математики та інформатики, почав в 1989 році.

На сьогоднішній день існують 3 версії Python. Версія 1.0 була випущена в січні 1994 року, версія 2.0 - 16 жовтня 2000 року, версія 3.0 - 3 грудня 2008 року. Крім основних версій існують ще проміжні, які випускали в періоди між основними. Самою останньою версією є Python 3.9.5 (3 травня 2021 роки).

Основні переваги та особливості:

- Будь-який бажаючий може навчитися програмувати на Python всього за кілька годин або кілька днів. Освоєння Python і всіх його складних концепцій, пакетів і модулів може зайняти більше часу. Однак вивчити базовий синтаксис дуже легко в порівнянні з іншими популярними мовами, такими як C, C++ і Java.
- З синтаксису було прибрано все зайве, код чистий і зрозумілий без зайвих дужок і виразів.
- PEP - єдиний стандарт для написання коду, що робить код підтримуваним і читабельним навіть при переході від одного програміста до іншого.
- Python розроблений під ліцензією з відкритим вихідним кодом, схваленої OSI. Отже, його можна використовувати абсолютно безкоштовно навіть в комерційних цілях.
- Крім стандартних бібліотек Python, до нього можна підключити просто незліченну кількість додаткових модулів і бібліотек, доступних для всіх. Також величезним плюсом є те, що ми можемо використовувати C код з Python.
- Немає необхідності компілювати Python, тому що він обробляється інтерпретатором під час виконання.
- Open Source - код інтерпретатора Python є відкритим, що дозволяє будь-кому, хто зацікавлений у розвитку мови взяти участь в його розробці і поліпшити його. Якщо дивитися деталі релізу однією з версій мови, то можна помітити, що величезні частини нового функціоналу реалізовані сторонніми розробниками.
- Python пропонує різні набори інструментів, такі як Tkinter, wxPython і JPython, які дозволяють легко і швидко розробляти графічний інтерфейс. Python - це мова програмування високого рівня, оскільки програмістам не потрібно пам'ятати архітектуру системи і керувати пам'яттю.

- Python реалізований під усіма поширеними операційними системами і на безлічі архітектур - Windows, Linux, MacOS, навіть на міні-комп'ютерах Arduino. Система залежностей добре продумана, і розгортання додатків на іншій машині відбувається легко.

2.4.2. Документація Telegram Api

На наступному етапі вибору засобів розробки необхідно ознайомитися з принципом роботи Telegram ботів. Тому була вивчена документація Telegram Bot Api з описом методів і параметрів, які передаються.

Bot Api - це інтерфейс на базі HTTP, який створений з метою розробки ботів для месенджера Telegram. Як вже було розглянуто, бот представляє собою спеціальний акаунт, який автоматично обробляє повідомлення та відправляє відповіді.

Розглянемо принцип отримання оновлень від ботів. Під оновленням в даному випадку мається на увазі яка-небудь діяльність з ботом (наприклад, отримання повідомлення від користувача). Telegram має два способи приймати оновлення: періодичні запити (метод `getUpdates`) і установка веб-хуків (метод `setWebhook`). Вони абсолютно протилежні один одному за логікою. Всі оновлення, які приходять, зберігаються до моменту обробки сервером протягом 24 години. А у відповідь приходить об'єкт `Update`, що серіалізований в JSON.

Періодичні запити або метод `getUpdates`. Суть методу в тому, щоб періодично опитувати сервери месенджера Telegram на присутність нової інформації. Існує два зв'язки: `long polling` та `polling`. При `long polling` бот працює з затримкою, яку було задано в програмному коді, і сервер відповідає тільки в двох ситуаціях: якщо прийшло нове повідомлення або необхідно розривати з'єднання. В даному дипломному проектуванні був використаний `polling` – кожену секунду відправляються запити на сервер для перевірки оновлень. У таблиці 2.1 розглянемо параметри методу і відомості про них.

Відомості про параметри методу getUpdates

Параметри	Тип	Статус	Опис
offset	Integer	Необов'язковий	Перегляд нових повідомлень з ігноруванням старіших
limit	Integer	Необов'язковий	Обмеження в отриманні нових оновлень. Доступна кількість: 1 - 100
timeout	Integer	Необов'язковий	Період часу, протягом якого необхідно дати відповідь

Важливо:

1. Метод не буде функціонувати, якщо заздалегідь був підключений webhook.
2. Щоб виключити повторення оновлень, необхідно кожен раз по новій вираховувати offset.

Установка веб-хуків або метод setWebhook. Логіка роботи вебхука полягає в тому, що при надходженні нових повідомлень в чат безпосередньо Telegram сповіщає про це. Відповідно немає необхідності виконувати періодичні опитування сервера. Метод дозволяє задавати URL адресу веб-хука, який буде приймати оновлення від бота. Коли на цю адресу надходитимуть оновлення, кожен раз буде відправлен HTTPS POST. Якщо запит до сервера невдалий - спроба буде повторюватися невелику кількість разів. Заради більш кращої безпеки і впевненості, що запити до веб-хуку відправляє не інакше як Telegram, можна включити токен (особистий ключ авторизації) в URL адресу веб-хука, наприклад: [https://webhookserver.com/ <token>](https://webhookserver.com/<token>). Токен можна отримати, написавши спеціалізованому боту @BotFather. У таблиці 2.2 розглянемо параметри методу setWebhook і відомості про них.

Відомості про параметри методу setWebhook

Параметри	Тип	Статус	Опис
url	String	Необов'язковий	URL-адресу HTTPS - використовується для відправки запитів. Для видалення вебхука досить відправити незаповнену рядок
certificate	InputFile	Необов'язковий	Перевірка кореневого сертифіката за допомогою завантаження публічного ключа

Важливо:

1. Якщо використовується сертифікат, який підписаний самостійно - важливо зробити завантаження публічного ключа, застосовуючи параметр certificate.
2. Для відправки оновлень через webhook на даний момент доступні тільки такі порти, як: 443, 80, 88, 8443.

Існує велика кількість інших методів для Telegram Bot Api. Найпоширеніші це:

- sendMessage: метод дозволяє відправляти текстові повідомлення;
- forwardMessage: метод дозволяє пересилати повідомлення;
- sendPhoto: метод дозволяє відправляти фотографії;
- sendAudio: метод дозволяє відправляти аудіофайли у форматі .mp3;
- sendSticker: метод дозволяє відправляти стікери в форматі .webp;
- sendVideo: метод дозволяє відправляти відеофайли у форматі .mp4;
- sendContact: метод дозволяє відправляти телефонні контакти.

Найпоширеніші об'єкти:

- Reply Keyboard Markup (об'єкт являє собою клавіатуру з опціями відповіді);
- Keyboard Button (об'єкт являє собою одну кнопку в клавіатурі відповіді);

- Reply Keyboard Hide (після отримання повідомлення з цим об'єктом, додаток Telegram згорне клавіатуру бота і відобразить стандартну клавіатуру пристрою - з буквами);
- Inline Keyboard Markup (об'єкт являє собою вбудовану клавіатуру, яка з'являється під відповідним повідомленням);
- Inline Keyboard Button (об'єкт являє собою одну кнопку вбудованої клавіатури);
- Callback Query (об'єкт являє собою вхідний запит зворотнього зв'язку від інлайн-кнопки з заданим callback_data).

Запити в Telegram Bot Api допускаються через HTTPS в такому вигляді:
<https://api.telegram.org/bot<токен>/метод>.

На рис. 2.3 розглянемо принцип роботи чат-бота на платформі Telegram.

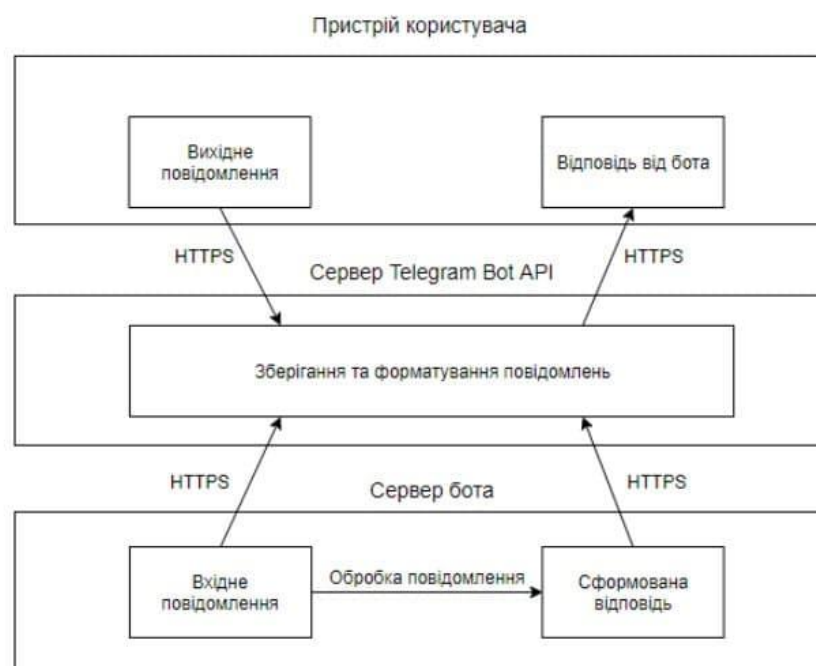


Рис.2.3. Представлення роботи чат-бота в Telegram

2.5. Характеристика середовища розробки Pycharm

Підходящою програмою для обраної мови програмування Python є PyCharm.

PyCharm - це інтегроване середовище розробки, що призначене для Python. Дана програма підтримує HTML, CSS, JavaScript і крім того можна додати інші мови за допомогою плагінів. Так само як і Python, Pycharm має 3 версії: 1.0 випущена 2010 року, 2.0 - 2011 року, 3.0 - 2013 року. На рис. 2.4 зображений інтерфейс PyCharm версії 3.3.

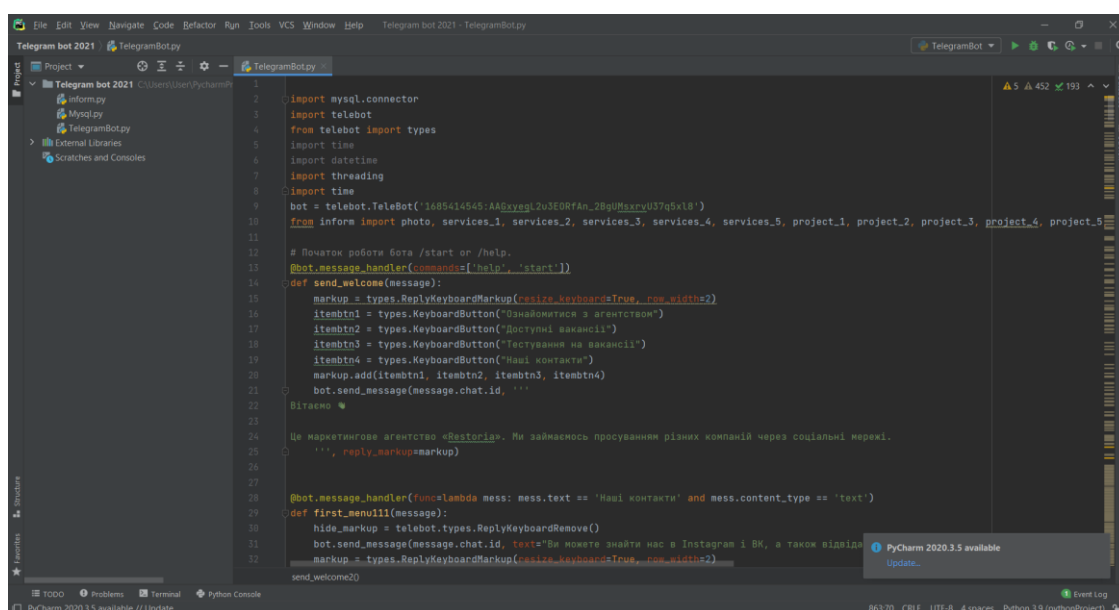


Рис.2.4. Інтерфейс Pycharm

Можливості та переваги програми:

- Доступне редагування коду. Наявність таких функцій, як автодоповнення, форматування, відступи, виправлення помилок з підсвічуванням коду сприяє більш зручній та швидкій, ефективній і продуктивній роботі користувачів.
- Наявність рефакторинга коду. Користувачі мають можливість розділяти розширені функції або класи методом вилучення, швидко змінювати локальні та глобальні змінні, покращувати структуру всередині та при цьому не змінювати продуктивність коду зовні.

- Наявність навігації коду. Розробнику легше та набагато швидше редагувати, вдосконалювати код, переходити до будь-яких файлів або функцій з класами, знаходити певні елементи та перевіряти код.
- Підтримка бібліотек. Pycharm підтримує Matplotlib, Anaconda, NumPy (бібліотеки Python), що допомагає при побудові проектів Machine Learning та Data Science.
- Підтримка поширених фреймворків мови Python. Підтримує Django, Pyramid, web2py та багато інших фреймворків. Допомагає налагодити код, пропонує автозаповнення щодо параметрів Django.
- Крос-платформеність. Pycharm працює на Linux, Windows і Mac OS.
- Доступ до баз даних (MySQL, Oracle, Server і ін).

2.6. Характеристика бази даних MySQL

У дипломному проектуванні, яке розроблялося, одним з найважливіших етапів є запис і зберігання контактних даних потенційних кандидатів для подальшого зв'язку з ними. В наслідок чого, наступний крок при проектуванні Telegram бота - вибір бази даних.

База даних - організований структурований набір даних для зберігання і використання пов'язаної між собою інформації. Якісна база даних повинна надавати вільний доступ до інформації та незалежність даних, забезпечувати синхронну роботу з декількома користувачами одночасно і гарантувати безперебійну роботу тощо.

Для чат-ботів можна підключати будь-які БД, але більш практичними є саме реляційні - відображають інформацію у вигляді таблиці. За результатами досліджень і пошуку найбільш придатних, сучасних засобів зберігання персональних даних було вирішено використовувати СУБД MySQL.

MySQL - це реляційна система управління БД, яка була розроблена і продовжує підтримку корпорацією Oracle. На відміну від інших програм MySQL з легкістю працює з інтерфейсом API. Тобто до поточної бази даних можна підключати додатки, які створені на таких мовах програмування: C, C++, Java, Python, Visual Studio та ін.

Розглянемо провідні переваги MySQL:

- велика швидкість при обробці інформації та виконання команд;
- наявність відкритого коду, тобто кожен користувач має безкоштовний доступ до скачування і навіть доопрацювання;
- легкість у використанні: має велику кількість плагінів і допоміжних додатків, які полегшують роботу з базами даних;
- підтримка кросплатформенних: MySQL працює на macOS, Linux, Microsoft Windows та на багатьох інших;
- підвищена безпека: має широку систему паролів і привілеїв;
- гнучкість: підтримка безлічі таблиць, завдяки чому користувачі можуть вибирати і таблиці, які підтримують повнотекстовий пошук (MyISAM), і таблиці, які підтримують транзакції на рівні індивідуальних записів (InnoDB);
- підтримка роботи відразу з великою кількістю користувачів.

РОЗДІЛ 3

СТВОРЕННЯ ЧАТ-БОТА

Переходимо до створення самого чат-боту за допомогою розглянутих та обраних у другому розділі засобів розробки.

3.1. Налаштування чат-бота

Для початку необхідно зареєструвати бота в месенджері Telegram. У цьому допоможе основний бот - BotFather, який призначений для реєстрації, настройки та управління Telegram ботами. Після того, як в пошуку додатка знайшли BotFather і почали з ним роботу за допомогою команди «/start», необхідно відправити наступну команду в чат: «/newbot». Далі слід придумати і написати назву бота, в якому обов'язково повинна бути приставка «bot». В кінцевому підсумку, якщо все зроблено правильно, то BotFather відправляє повідомленням з токеном (набір символів, який служить для отримання і відправки повідомлення за допомогою Bot API) і URL-адресою, який надає доступ до чат-боту. На рис. 3.1 зображена реєстрація RestoriaBot.

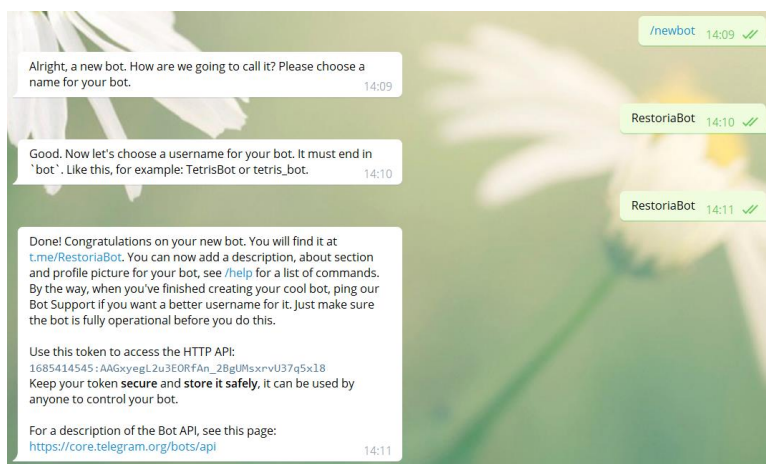


Рис.3.1. Реєстрація RestoriaBot в додатку Telegram

Кафедра КІТ (47)				НАУ 21 33 93 000 ПЗ			
Виконав	Козіна Ю.В.			Створення чат-бота	Літера	Аркуш	Аркушів
Керівник	Савченко А.С.					34	22
Консульт.	Колісник О.В.				412 122		
Н-котрол.	Шевченко О.П.						
Зав. каф.	Савченко А.С.						

Після реєстрації є можливість оформити бота за своїм бажанням: додати фото, опис і багато іншого. Для редагування BotFather пропонує перелік команд, що продемонстровані в таблиці 3.1.

Таблиця 3.1

Команди для редагування ботів

Назва команди	Опис команди
/setname	Змінити назву бота
/setdescription	Змінити або додати опис бота
/setabouttext	Змінити або додати інформацію про бота
/setuserpic	Змінити або додати фото бота
/setcommands	Змінити або додати список команд
/deletebot	Видалити бота

При налаштуванні та редагуванні RestoriaBot були використані деякі перераховані вище команди. Інтерфейс бота, який розроблявся, можна знайти в додатку Б.

3.2. Розробка програмної частини

Програмна частина чат-боту написана на мові програмування Python та складається з файлів, які відображені на рис. 3.2. Як було зазначено у другому розділі, використано зв'язок Polling для отримання оновлень бота. Це пояснюється тим, що даний спосіб найбільш простий, зручний у використанні і не вимагає додаткових процедур для коректної роботи.

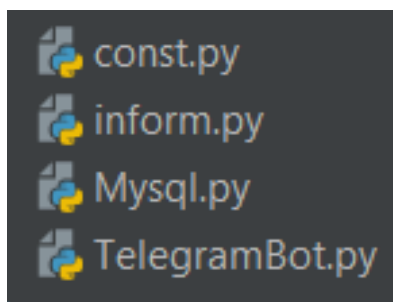


Рис.3.2. Система файлів для чат-боту в PyCharm

Детальніше про зміст кожного файлу структури:

1. const.py – знаходяться константи, які використовуються у проекті;
2. inform.py – знаходиться інформація про проекти та послуги агентства Restoria, яка буде використовуватися TelegramBot.py;
3. Mysql.py – підключення до сервера MySQL, створення бази даних, створення таблиці для запису інформації;
4. TelegramBot.py - реалізація самого боту, запис інформації в БД.

Для того, щоб почати створювати сценарій чат-бота, слід завантажити і підключити всі необхідні бібліотеки. На рис. 3.3 зображений код з підключенням до проекту всіх бібліотек та модулів, які знадобляться для розробки.

```
import mysql.connector
import telebot
from telebot import types
import datetime
import threading

from inform import photo, services_1, \
    services_2, services_3, services_4, \
    services_5, project_1, project_2, \
    project_3, project_4, project_5, project_6
from const import count_web, count_target, \
    p_i_p, black_list_target, black_list_web
```

Рис.3.3. Процес підключення бібліотек та відповідних модулів

3.2.1. Створення вітального повідомлення та головного меню

Як і будь-який інший чат-бот, RestoriaBot запускається і починає свою роботу за допомогою команди «/start», прописаної користувачем в поле для введення тексту. Але для того, щоб чат-бот сприйняв цю команду і продовжив своє функціонування, необхідно створити декоратор.

Декоратор представляє функцію, яка дозволяє обернути іншу функцію для розширення її функціональних можливостей без безпосередньої зміни коду. На рис. 3.4 зображена частина коду для створення декоратора, який відіграє роль «слухача» з подальшою обробкою вхідних повідомлень.



```
# Початок роботи бота /start or /help.  
@bot.message_handler(commands=['help', 'start'])
```

Рис.3.4. Створення декоратора для чат-бота

Тепер декоратор дозволяє боту обробити першу команду, в результаті чого користувач у відповідь отримає вітальне повідомлення та доступ до головного меню. Розберемо процес реалізації цих функцій:

- метод *send_message* слугує для відправлення першого повідомлення в чат, а об'єкт *reply_markup* використовується для того, щоб після вітального повідомлення зразу з'явилися перші кнопки;
- метод *ReplyKeyboardMarkup* призначений для створення клавіатури, а параметри *resize_keyboard* та *row_width* задають зовнішній вигляд. Для зміни розміру клавіатури з урахуванням кількості кнопок використовується параметр *resize_keyboard*, а для встановлення ширини рядків – параметр *row_width*;
- метод *KeyboardButton* призначений для створення кнопки, натискання якої дозволить користувачеві дати відповідь і відповідно відправити команду боту;

На рис. 3.5 зображено код програми в Pycharm, завдяки якому реалізується процес відправки вітального повідомлення та створення головного меню.

```
def send_welcome(message):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True, row_width=2)
    itembtn1 = types.KeyboardButton("Ознайомитися з агентством")
    itembtn2 = types.KeyboardButton("Доступні вакансії")
    itembtn3 = types.KeyboardButton("Тестування на вакансії")
    itembtn4 = types.KeyboardButton("Наші контакти")
    markup.add(itembtn1, itembtn2, itembtn3, itembtn4)
    bot.send_message(message.chat.id, '''
Вітаємо 🖐️

Це маркетингове агентство «Restoria». Ми займаємось просуванням різних компаній через соціальні мережі.
''', reply_markup=markup)
```

Рис.3.5. Створення вітального повідомлення та головного меню

Розглянемо розділи основного меню:

- «Ознайомитися з агентством»;
- «Доступні вакансії»;
- «Тестування на вакансії»;
- «Наші контакти».

Результат роботи розглянутого фрагмента коду можна побачити на рис. 3.6.

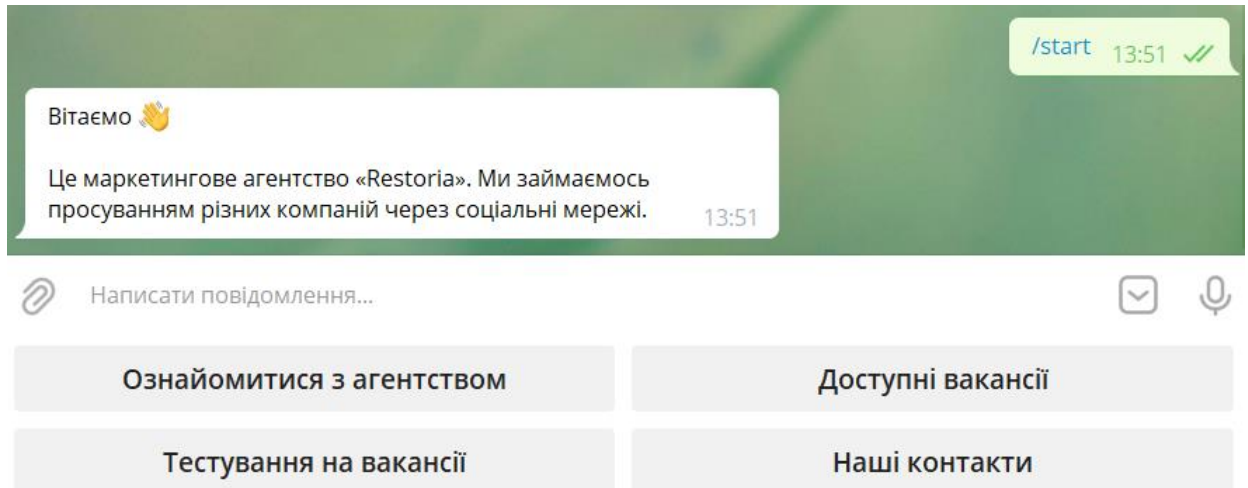


Рис.3.6. Головне меню RestoriaBot

Розберемо кожний розділ головного меню детальніше:

1. «Ознайомитися з агентством».

Цей розділ дозволяє потенційному кандидату дізнатися більше інформації про агентство, в яке він планує влаштовуватися. Складові розділу:

- «Наші послуги»;

- «Наші проекти»;
- «Відгуки»;
- «На головну».

Складові розділу «Ознайомитися з агентством» також формують меню (рис. 3.7), сценарій створення якого аналогічний головному меню.

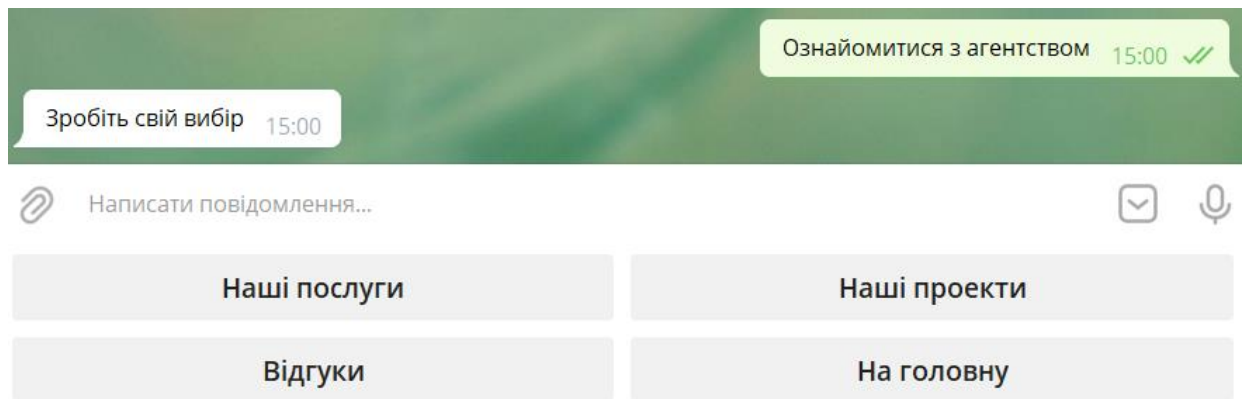


Рис.3.7. Меню розділу «Ознайомитися з агентством»

Розглянемо роботу чат-бота при виборі однієї з кнопок. Наприклад, користувач обрав «Наші послуги», а у відповідь бот миттєво відправляє відповідну інформацію (рис. 3.8). При цьому клавіатура змінюється на кнопки:

- «Назад»: повертає кандидата на крок назад;
- «На головну»: є в кожному розділі для того, щоб користувач в будь-який момент міг повернутися до головного меню.

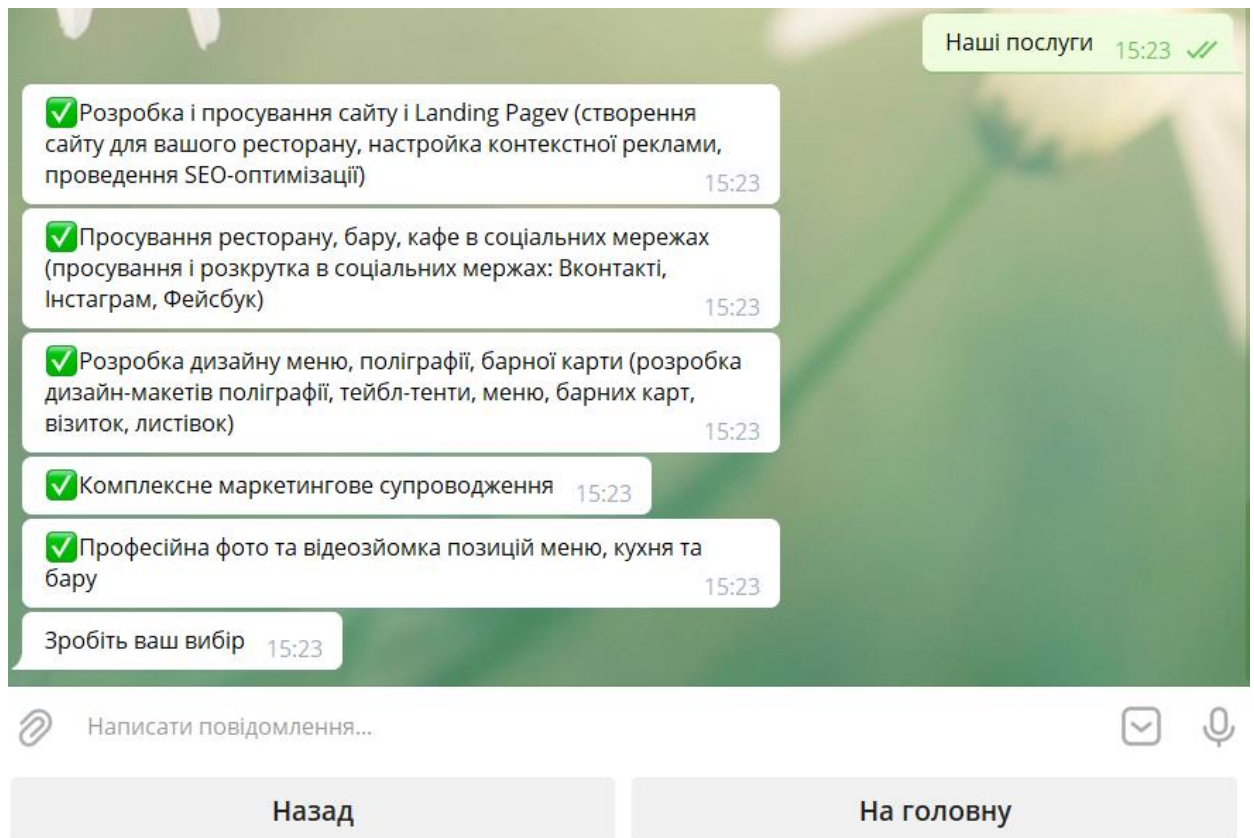


Рис.3.8. Результат натиснення кнопки «Наші послуги»

Дані, пов'язані з інформацією про агентство («Наші послуги», «Наші проекти», «Відгуки»), знаходяться у файлі *inform.py*. Вони прив'язані до конкретних змінних, тому що так можна виводити будь-який обсяг інформації, не навантажуючи код звичайним текстом.

2. Розділ «Тестування на вакансії».

Поточний розділ є основним, так як в ньому відбувається основна функція - тестування для перевірки кваліфікації і зберігання контактних даних підходящих кандидатів, тобто здійснюється онлайн співбесіда. Складові:

- «Тест на таргетолога»;
- «Тест на веб-дизайнера»;
- «На головну».

RestoriaBot пропонує користувачу обрати одну з двох доступних вакансій для подальшого проходження тестування (рис. 3.9).

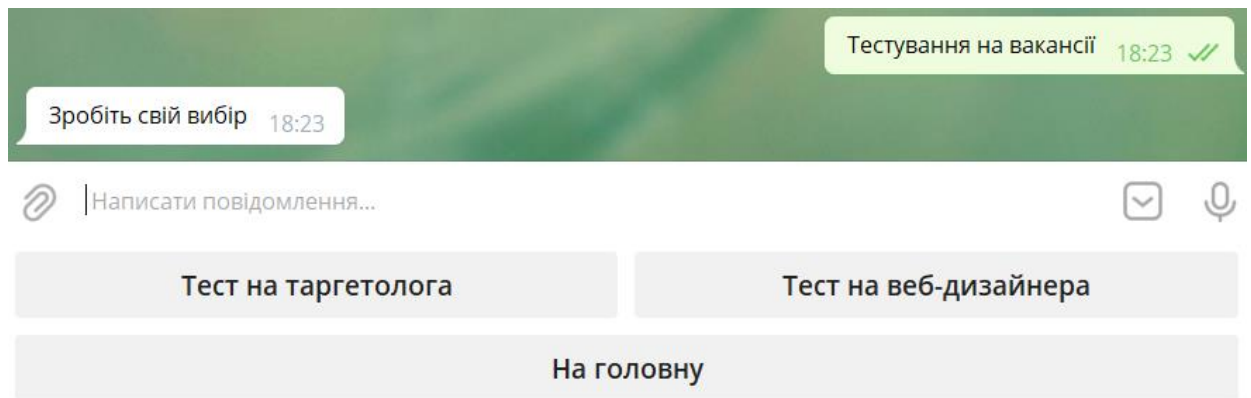


Рис.3.9. Розділ «Тестування на вакансії»

Принцип та логіка роботи наступна. Наприклад, кандидат обирає кнопку «Тест на веб-дизайнера» та отримує миттєву відповідь від бота з підтвердженням свого вибору. При цьому з'являється інша клавіатура - вбудована, яка створюється за допомогою методу *InlineKeyboardMarkup*. Ця клавіатура розташовується в самому чаті під відповідним повідомленням бота, на відміну від звичайної, яка знаходиться в нижній частині коду у вигляді меню. А з використанням методу *InlineKeyboardButton* створюються кнопки вбудованої клавіатури. Програмний код для реалізації описаного зображений на рис. 3.10. А результат та вигляд вбудованої клавіатури в Telegram – на рис. 3.11.

```
@bot.message_handler(func=lambda mess: mess.text == 'Тест на веб-дизайнера' and mess.content_type == 'text')
def first_menu4(message):
    hide_markup = telebot.types.ReplyKeyboardRemove()
    bot.send_message(message.chat.id, text="Тест на веб-дизайнера", reply_markup=hide_markup)
    keyboard = types.InlineKeyboardMarkup()
    button_1 = types.InlineKeyboardButton(text="Так", callback_data='yeq')
    button_2 = types.InlineKeyboardButton(text="Ні", callback_data='noq')
    keyboard.add(button_1)
    keyboard.add(button_2)
    bot.send_message(message.chat.id, text="Підтвердити вибір", reply_markup=keyboard)
```

Рис.3.10. Розробка вбудованої клавіатури



Рис.3.11. Приклад вбудованої клавіатура в Telegram

Припустимо, що кандидат підтверджує свій вибір та натискає на кнопку «Так». Для того, щоб бот продовжив свою роботу та відповідно обробив команду, в сценарії було створено обробник подій `@bot.callback_query_handler` – несе відповідальність за коректну роботу з inline-кнопками (вбудованими). В результаті чого, після натиснення кнопки (на вбудованій клавіатурі) віртуальний помічник не зупинить свою роботу. І в даному випадку відправить повідомлення з проханням ввести своє ППІ кандидату (при виборі кнопки «Ні» бот повертає на крок назад). На рис. 3.12 зображено програмний код для запиту та ведення користувачем ППІ перед початком тестування, а на рис. 3.13 – результат в Telegram.

```
@bot.callback_query_handler(func=lambda c: c.data and c.data.startswith('yeq'))
def web_yes(c):
    global count_web, msgg, web_job
    web_job = "Веб-дизайнер"
    msgg = bot.send_message(c.message.chat.id, "Введіть прізвище, ім'я, по-батькові")
    bot.register_next_step_handler(msgg, p_i_p_web)
```

Рис.3.12. Запит ППІ перед початком тестуванням в Pycharm



Рис.3.13. Підтвердження вибору

Введення ППП необхідно не тільки для того, щоб записати дані кандидата в БД, але і для того, щоб при невдалому проходженні тестування один і той же користувач не зміг пройти повторно. Це пов'язано з тим, що якщо кандидат в перший раз завалив тест - він не пройшов співбесіду і відповідно не підходить для роботи в агентстві. Дана функція була реалізована в програмному коді за допомогою умов *if* та *else*:

- В умові *if* йде порівняння тільки що записаної інформації зі списком тих кандидатів, які вже проходили тестування (запис нового імені в список відбувається, коли кандидат отримує перше питання під час тестування). Якщо спостерігається збіг між ППП, то бот відправляє повідомлення «Ви вже проходили тестування» та з'явиться кнопка «На головну».

- Умова *else* активізується, якщо при перевірці не виявили збігів. Далі користувач отримує повідомлення про правила проходження теста та дві кнопки: «Вперед!» (кандидат переходить до тесту) та «Повернутися на головну» (у випадку натиснення цієї кнопки нове ППП видалиться зі списку).

Фрагмент коду с перевіркою на повторне проведення тесту зображено на рис. 3.14.

```
def p_i_p_web(message):
    global web_pip
    web_pip = str(message.text)
    global black_list_web
    global pip_in_black_list_web
    pip_in_black_list_web = str(message.text)
    pip_in_black_list_web = pip_in_black_list_web.lower()
    if pip_in_black_list_web in black_list_web:
        markup = types.ReplyKeyboardMarkup(resize_keyboard=True, row_width=2)
        itembtn1 = types.KeyboardButton("На головну")
        markup.add(itembtn1)
        bot.send_message(message.chat.id, "Ви вже проходили тест", reply_markup=markup)
    else:
        keyboard = types.InlineKeyboardMarkup()
        button_1 = types.InlineKeyboardButton(text="Вперед!", callback_data='web_test')
        button_2 = types.InlineKeyboardButton(text="Повернутися на головну", callback_data='go_main')
        keyboard.add(button_1)
        keyboard.add(button_2)
        bot.send_message(message.chat.id, text='''
Будь ласка, пройдіть невеличке тестування. Це допоможе заощадити наш час при виборі кандидата.

Але не все так просто 🙄

На відповідь кожного питання вам дається 15 сек, якщо ви не встигаєте відповісти - відповідь не зараховується.

Готові?
''', reply_markup=keyboard)
```

Рис.3.14. Перевірка повторного тестування

Повідомлення, яке отримує користувач після введення інформації, вказує на те, що тестування обмежено в часі. А саме: на відповідь кожного питання дається 15 секунд – в діалозі користувачеві відображається таймер. Сценарій написання таймеру зображено на рис. 3.15, а вигляд в Telegram – на рис. 3.16.

```
def web_timerr_0():
    global chat_id_message_web, web_znak, chat_message_1, timer_1
    timer_1.cancel()
    i = 15
    i_1 = 0
    asx = bot.send_message(chat_id_message_web, text = "Залишилось 15 сек.")
    while web_znak:
        i-=1
        i_1+=1
        if i_1<=15 and i_1>0:
            bot.edit_message_text(chat_id=chat_id_message_web, message_id=asx.message_id, text = "Залишилось " + str(i)+" сек.")
        else:
            pass
    time.sleep(1)
```

Рис.3.15. Створення таймеру



Рис.3.16. Вигляд роботи таймера в Telegram

Після натиснення кнопки «Вперед!» починається саме тестування, як можна бачити на рис. 3.16. Тест містить 5 питань, а мінімальна кількість правильних відповідей – 3. Якщо кандидат не встигає за 15 секунд дати відповідь, то вона зараховується як неправильна. А бот в свою чергу відправляє повідомлення з нагадуванням про обмежений час та кнопку «Продовжити тест», натиснення якої призведе до відправки наступного питання (рис. 3.17). Відповідно, коли кандидат обирає відповідь, незалежно від того чи правильна вона, бот відправляє також чергове питання (рис. 3.18). Фрагмент коду з реалізацією тестування зображено на рис. 3.18.

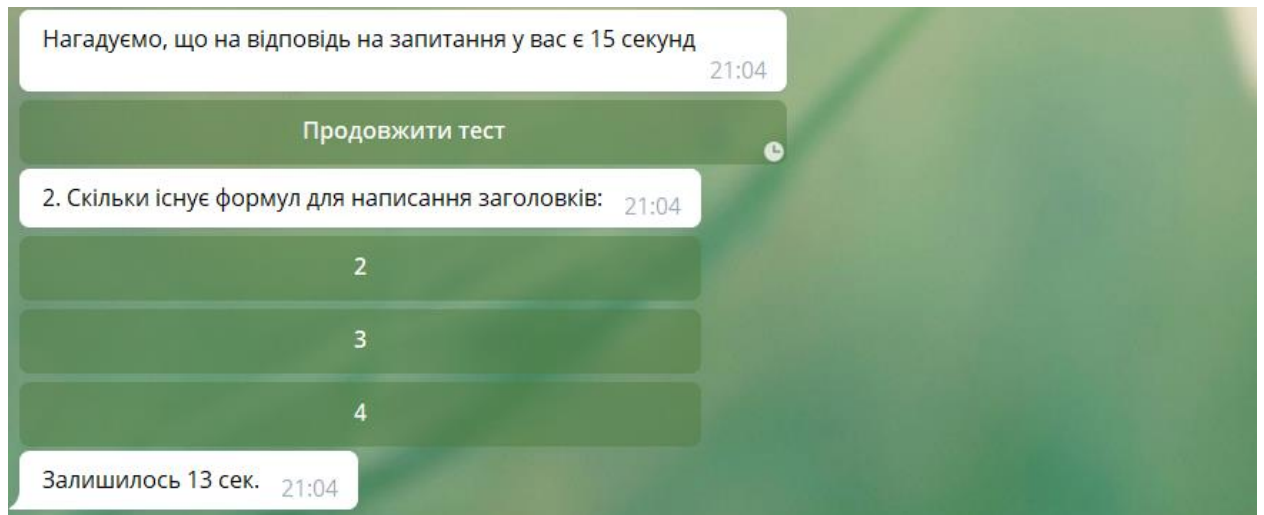


Рис.3.17. Нагадування про обмежений час

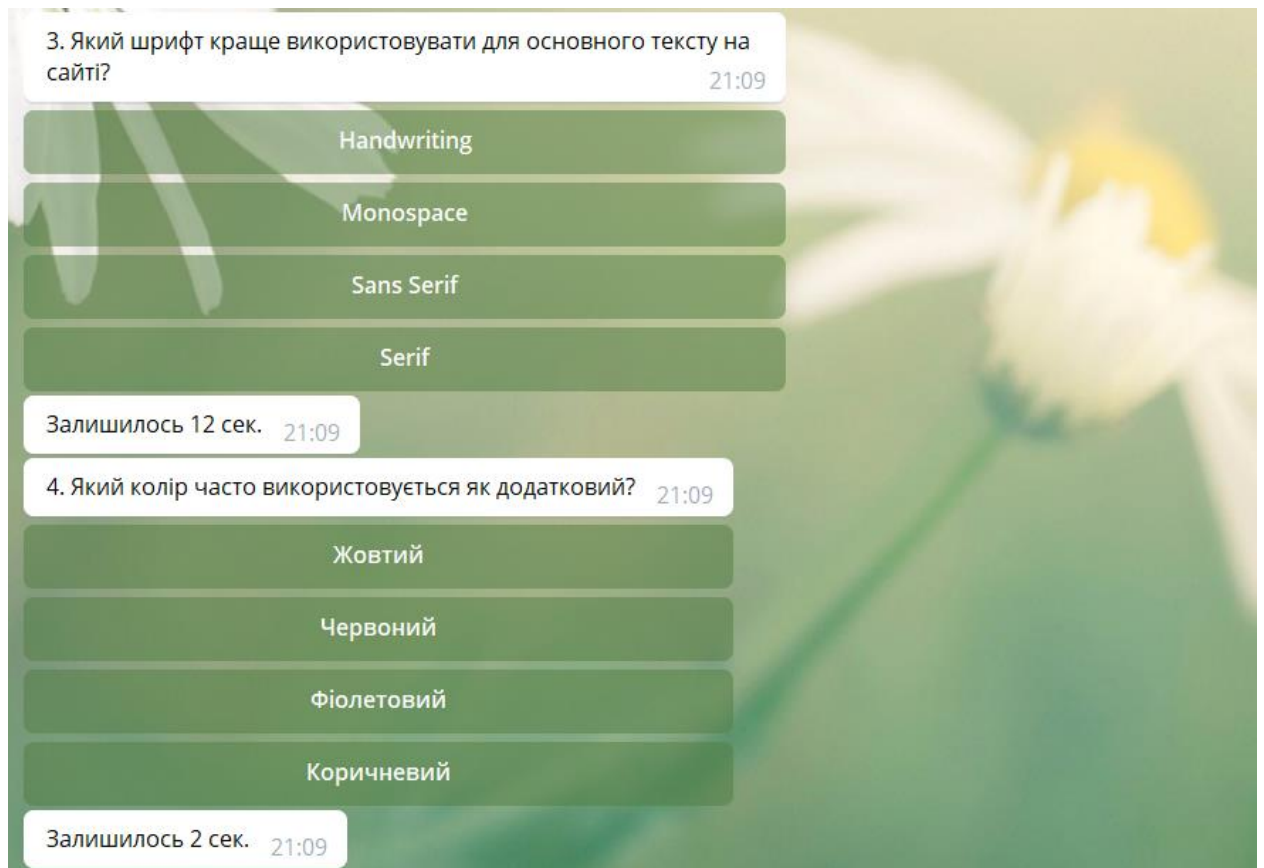


Рис.3.18. Відправка наступного питання після відповіді

```
@bot.callback_query_handler(func=lambda c: c.data and c.data.startswith('web_test'))
def web_yes_test(c):
    global black_list_web, chat_id_message_web, web_znak, pip_in_black_list, timer, timer_1
    global pip_in_black_list_web
    black_list_web.append(pip_in_black_list_web)
    chat_id_message_web = c.message.chat.id
    web_znak = True
    timer = threading.Timer(15.0, web_job_1)
    timer_1 = threading.Timer(1.0, web_timerr_0)
    keyboard = types.InlineKeyboardMarkup()
    button_1 = types.InlineKeyboardButton(text="20 %", callback_data='first_web')
    button_2 = types.InlineKeyboardButton(text="50 %", callback_data='second_web')
    button_3 = types.InlineKeyboardButton(text="80 %", callback_data='third_web')
    keyboard.add(button_1)
    keyboard.add(button_2)
    keyboard.add(button_3)
    bot.send_message(c.message.chat.id, text="1. Скільки відсотків займає дизайн на сайті (то")
    timer_1.start()
    time.sleep(1)
    timer.start()
```

Рис.3.18. Реалізація тесту за допомогою мови Python

Після успішного проходження тестування RestoriaBot відправляє повідомлення з вітаннями та проханням відповісти ще на декілька питань, але стосовно вже досвіда роботи потенційного кандидата та контактних даних, яких не вистачає (рис. 3.19). Для відправки необхідно інформації користувач повинен натиснути на кнопку «Почати», яка знаходиться під останнім повідомленням. Після того, як дані були отправлені, з'являється наступне повідомлення «Ви не помилилися у відповідях?» та дві кнопки:

- «Все добре!»: при виборі даної кнопки приходить останнє повідомлення, в якому інформують про подальший зв'язок з потенційним кандидатом та пропонують перелік посилань (у вигляді вбудованих кнопок) на соціальні мережі агентства (рис. 3.20, рис. 3.21).

- «Записати нові відповіді»: якщо користувач помилився у своїх відповідях, в нього є можливість відправити нові.

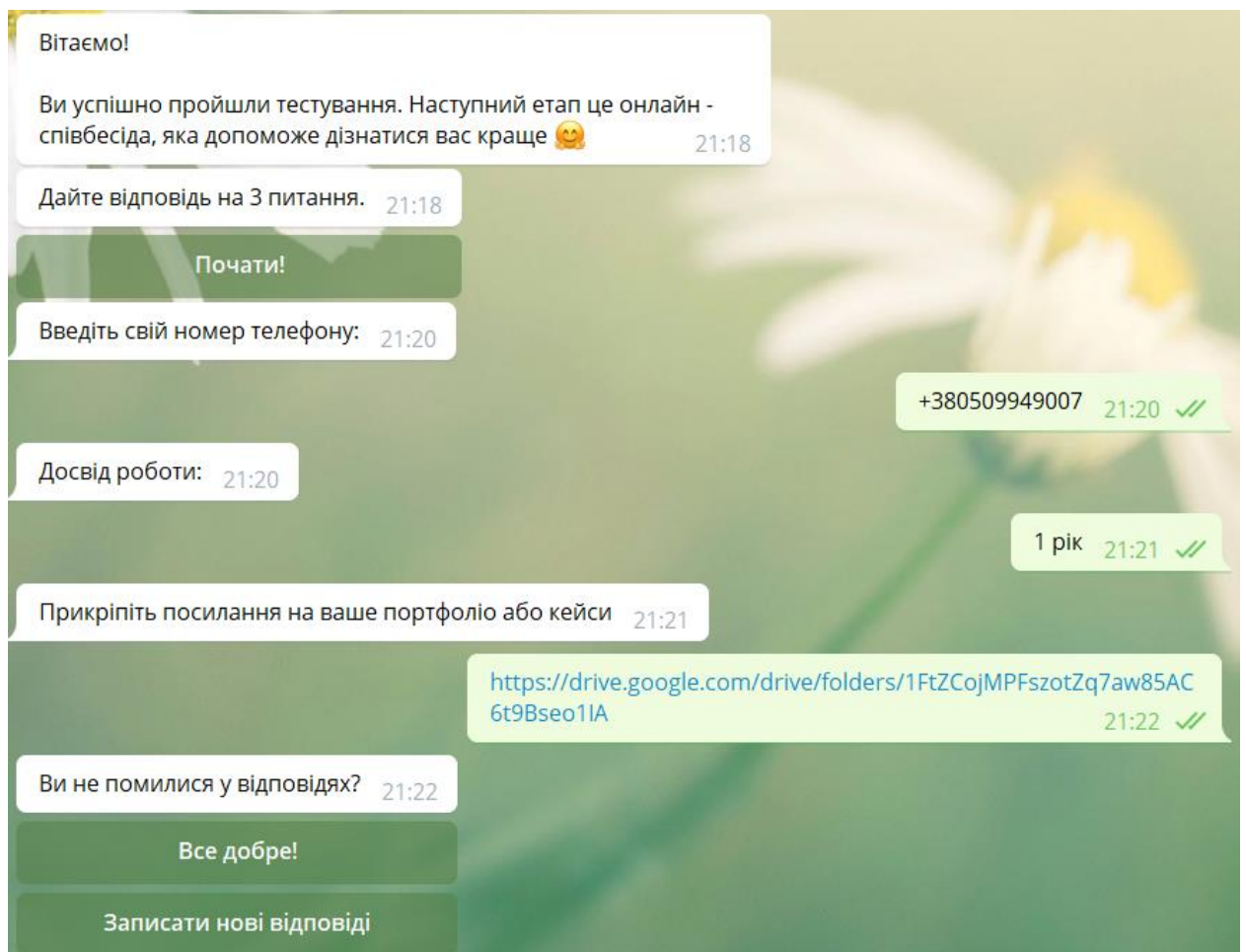


Рис.3.19. Результат успішного проходження тестування

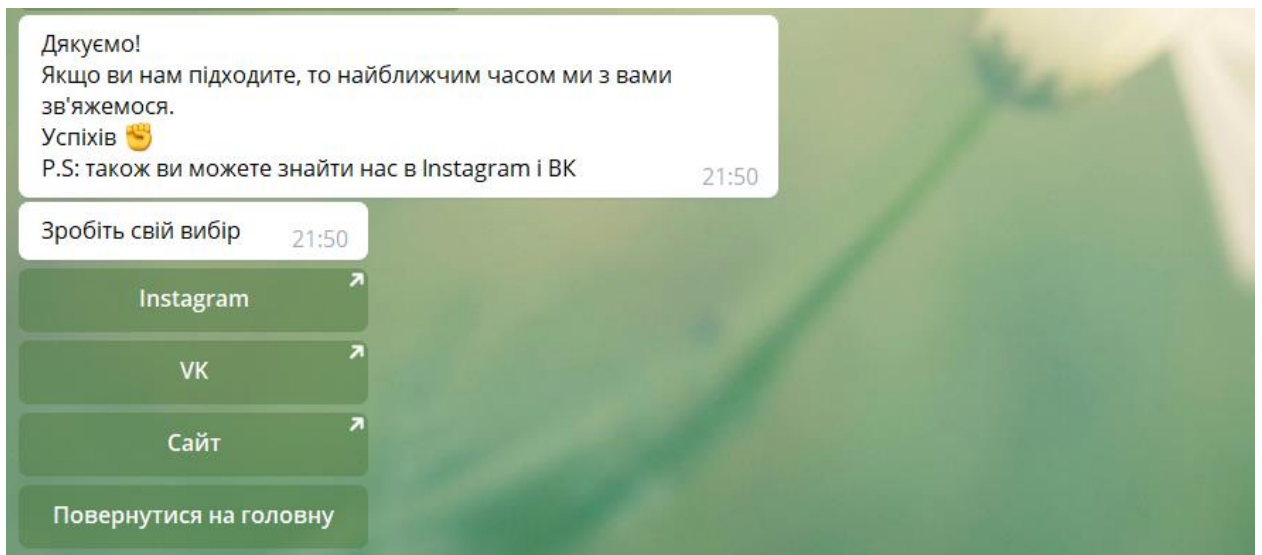


Рис.3.20. Результат натиснення кнопки «Все добре!»

```
@bot.callback_query_handler(func=lambda c: c.data and c.data.startswith('end_message_web'))
def end_message_allweb(c):
    global web_pip, phone_web_1, dosvid_web_1, port_web_1, web_job
    mydb = mysql.connector.connect(
        host="localhost",
        user="root",
        password="08032015",
        auth_plugin='mysql_native_password',
        database="Restoria_db")
    my_cursor = mydb.cursor()
    registration_request_call = "INSERT INTO target_web (PIP, Phone_number, Vacancy, Experience, Portfolio) VALUES (%s, %s, %s, %s, %s)"
    record3 = (web_pip, phone_web_1, web_job, dosvid_web_1, port_web_1)
    my_cursor.execute(registration_request_call, record3)
    mydb.commit()
    mydb.close()
    bot.send_message(c.message.chat.id, text='')
    Дякуємо!
    Якщо ви нам підходите, то найближчим часом ми з вами зв'яжемося.
    Успіхів 🍀
    P.S: також ви можете знайти нас в Instagram і ВК'''
    keyboard = types.InlineKeyboardMarkup()
    button_1 = types.InlineKeyboardButton(text="Instagram", url="www.instagram.com/restoria.agency")
    button_2 = types.InlineKeyboardButton(text="VK", url="vk.com/restoria.agency")
    button_3 = types.InlineKeyboardButton(text="Сайт", url="restoria.agency")
    button_4 = types.InlineKeyboardButton(text="Повернутися на головну", callback_data="go_main")
    keyboard.add(button_1)
    keyboard.add(button_2)
    keyboard.add(button_3)
    keyboard.add(button_4)
    bot.send_message(c.message.chat.id, "Зробіть свій вибір", reply_markup=keyboard)
```

Рис.3.21. Реалізація останнього повідомлення та посилань в Pycharm

Якщо тестування провалено, бот відправляє повідомлення про це та дві кнопки:

- «Подивитися правильні відповіді»: користувач має можливість передивитись всі відповіді – свої та правильні. Це дозволить зрозуміти, в яких питаннях були допущені помилки (рис. 3.22).
- «На головну»: бот повертає до головного меню.

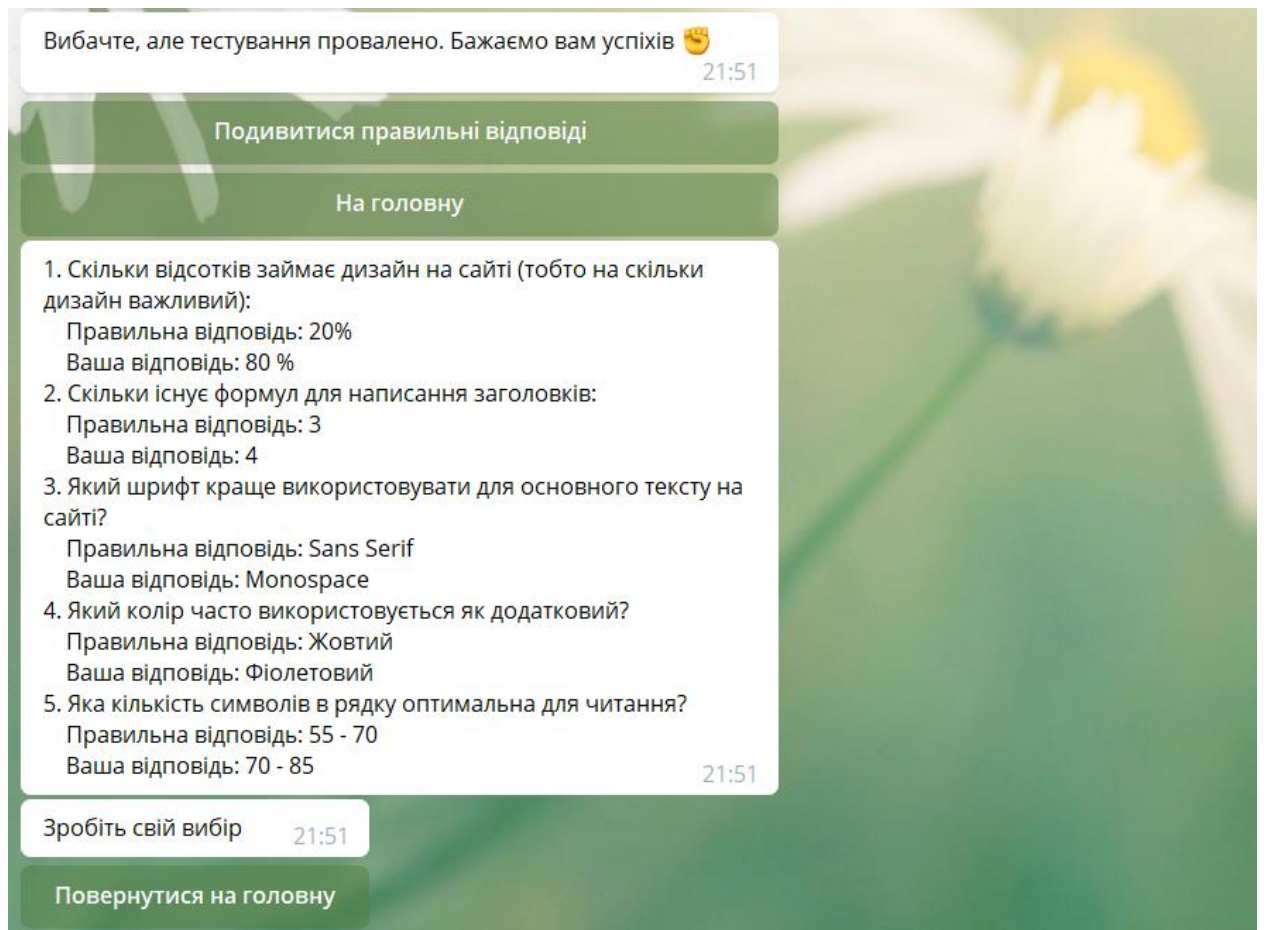


Рис.3.22. Результат проваленого тестування

Фрагмент коду з реалізацією відправки повідомлення при успішному тестуванні та проваленому зображено на рис. 3.23.

```

bot.send_message(c.message.chat.id, text='')
Вітаємо!

Ви успішно пройшли тестування. Наступний етап це онлайн - співбесіда, яка допоможе дізнатися вас краще 🤖'''
keyboard = types.InlineKeyboardMarkup()
button_1 = types.InlineKeyboardButton(text="Почати!", callback_data='Start_web')
keyboard.add(button_1)
bot.send_message(c.message.chat.id, text="Дайте відповідь на 3 питання.", reply_markup=keyboard)
else:
    keyboard = types.InlineKeyboardMarkup()
    button_1 = types.InlineKeyboardButton(text="Подивитися правильні відповіді", callback_data='web_answer')
    button_2 = types.InlineKeyboardButton(text="На головну", callback_data='go_main')
    keyboard.add(button_1)
    keyboard.add(button_2)
    bot.send_message(c.message.chat.id, text="Вибачте, але тестування провалено. Бажаємо вам успіхів 🍀", reply_markup=keyboard)

```

Рис.3.23. Код для відправки повідомлень після тестування

3. Розділ «Доступні вакансії».

В цьому розділі міститься інформація про наявні вакансії для потенційних кандидатів (рис. 3.24). Складові:

- «Таргетолог»;
- «Веб-дизайнер»;
- «На головну».

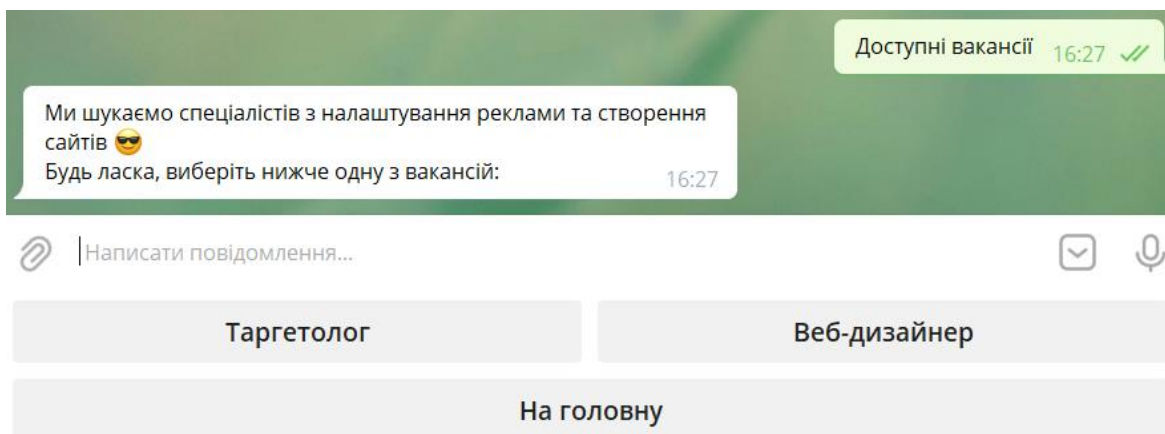


Рис.3.24. Розділ «Доступні вакансії»

При переході на одну з вакансій бот відправляє повідомлення з описом вимог до кандидата, його майбутніх обов'язків і пропозицій від агентства (рис. 3.25). Та разом з повідомленням кандидат отримує вбудовану клавіатуру з наступними кнопками:

- «Повернутися до вибору вакансій»: бот відповідно повертає користувача на крок назад – до вибору однієї з двох вакансій;
- «Пройти тестування»: потенційний кандидат переходить до тестування для підтвердження своєї кваліфікації (повторюється блок як в розділі «Тестування на вакансії»);
- «На головну»: бот повертає до головного меню.

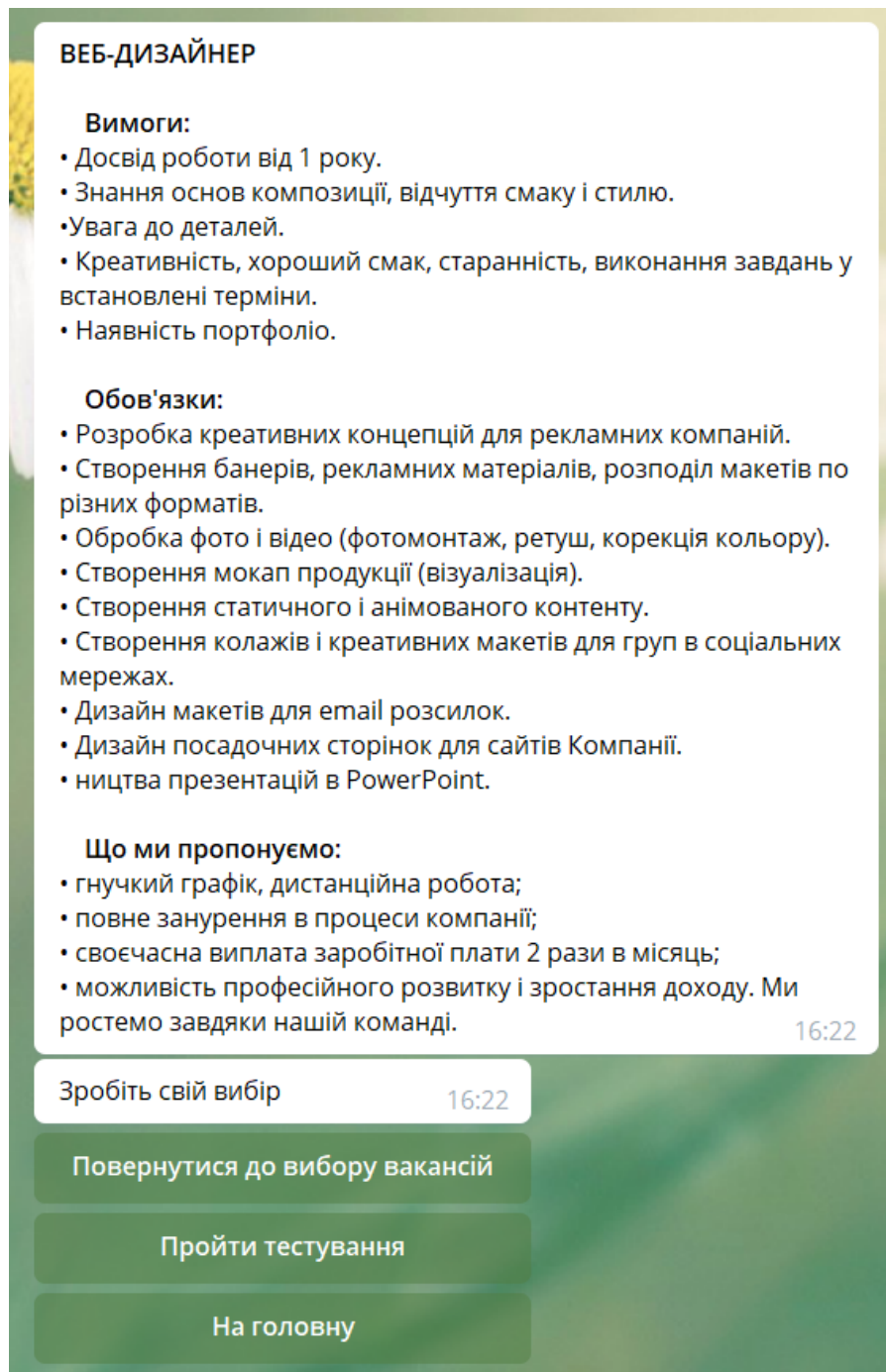


Рис.3.25. Вакансія «Веб-дизайнер»

Мова програмування Python дозволяє оформляти текст (застосовувати курсив, обирати жирність шрифту) та додавати різні смайли. Реалізація в програмному коді відображена на рис. 3.26.

```

bot.send_message(message.chat.id, text='''
    *ВЕБ-ДИЗАЙНЕР*

    *Вимоги:*
    * Досвід роботи від 1 року.
    * Знання основ композиції, відчуття смаку і стилю.
    * Увага до деталей.
    * Креативність, хороший смак, старанність, виконання завдань у встановлені терміни.
    * Наявність портфоліо.

    *Обов'язки:*
    * Розробка креативних концепцій для рекламних компаній.
    * Створення банерів, рекламних матеріалів, розподіл макетів по різних форматів.
    * Обробка фото і відео (фотомонтаж, ретуш, корекція кольору).
    * Створення мокап продукції (візуалізація).
    * Створення статичного і анімованого контенту.
    * Створення колажів і креативних макетів для груп в соціальних мережах.
    * Дизайн макетів для email розсилок.
    * Дизайн посадочних сторінок для сайтів Компанії.
    * ництва презентацій в PowerPoint.

    *Що ми пропонуємо:*
    * гнучкий графік, дистанційна робота;
    * повне занурення в процеси компанії;
    * своєчасна виплата заробітної плати 2 рази в місяць;
    * можливість професійного розвитку і зростання доходу. Ми ростемо завдяки нашій команді.
''', reply_markup=hide_markup, parse_mode='Markdown')

```

Рис.3.26. Вигляд відформатованого тексту в коді

4. Розділ «Наші контакти».

Розділ надає всі свої контакти у вигляді посилань на соціальні мережі (Instagram, VK) і офіційний сайт (рис. 3.28). Всі кнопки є вбудованими (inline). Реалізація в програмному коді зображена на рис. 3.27.

```

@bot.message_handler(func=lambda mess: mess.text == 'Наші контакти' and mess.content_type == 'text')
def first_menu111(message):
    hide_markup = telebot.types.ReplyKeyboardRemove()
    bot.send_message(message.chat.id, text="Ви можете знайти нас в Instagram і ВК, а також відвідати
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True, row_width=2)
    itembtn1 = types.KeyboardButton("На головну")
    markup.add(itembtn1)
    keyboard = types.InlineKeyboardMarkup()
    button_1 = types.InlineKeyboardButton(text="Instagram", url="www.instagram.com/restoria.agency")
    button_2 = types.InlineKeyboardButton(text="VK", url="vk.com/restoria.agency")
    button_3 = types.InlineKeyboardButton(text="Сайт", url="restoria.agency")
    keyboard.add(button_1)
    keyboard.add(button_2)
    keyboard.add(button_3)
    bot.send_message(message.chat.id, "Зробіть свій вибір", reply_markup=keyboard)
    bot.send_message(message.chat.id, text="Щоб повернутися до головного меню, натисніть 'На головну'

```

Рис.3.27. Фрагмент коду для розділу «Наші контакти»

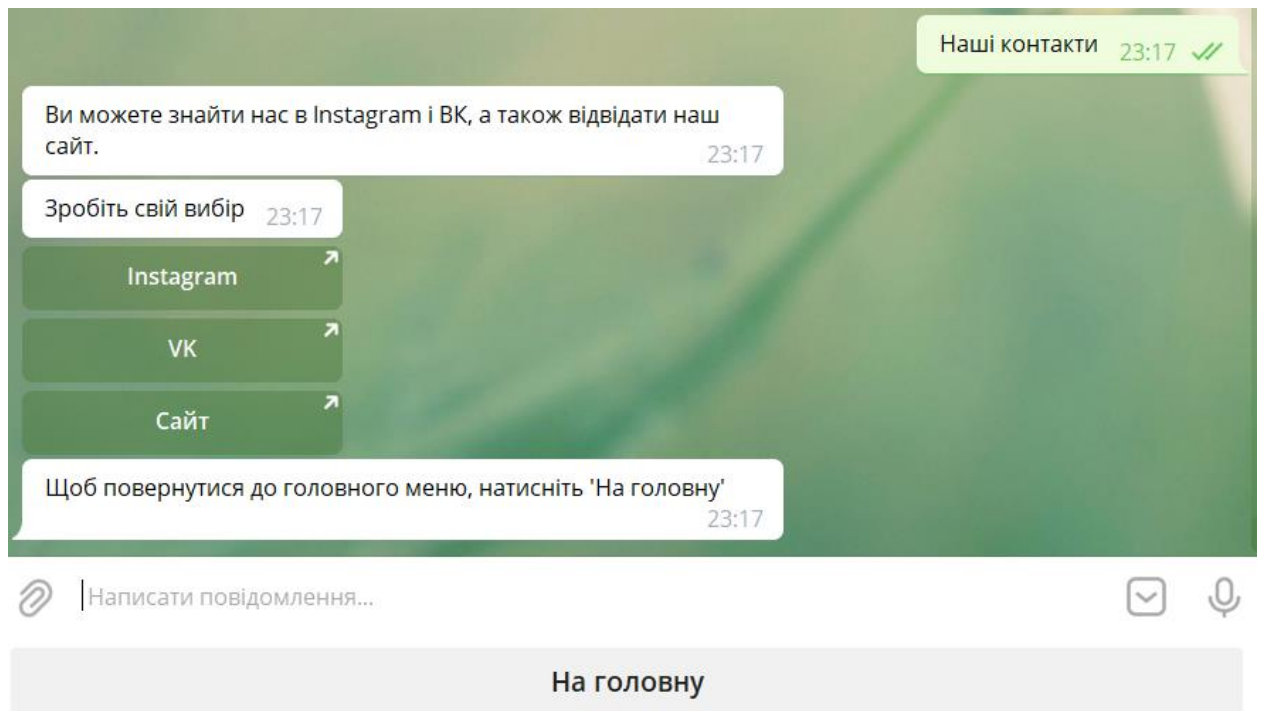


Рис.3.28. Розділ «Наші контакти»

3.3. Приєднання бази даних

Для можливості збереження контактних даних потенційних кандидатів та подальшого зв'язку з ними необхідно підключити БД MySQL (рис. 3.29). Вона використовується, коли потенційний кандидат залишає своє ППІ перед тестуванням та свої контактні дані після успішного проходження тесту.

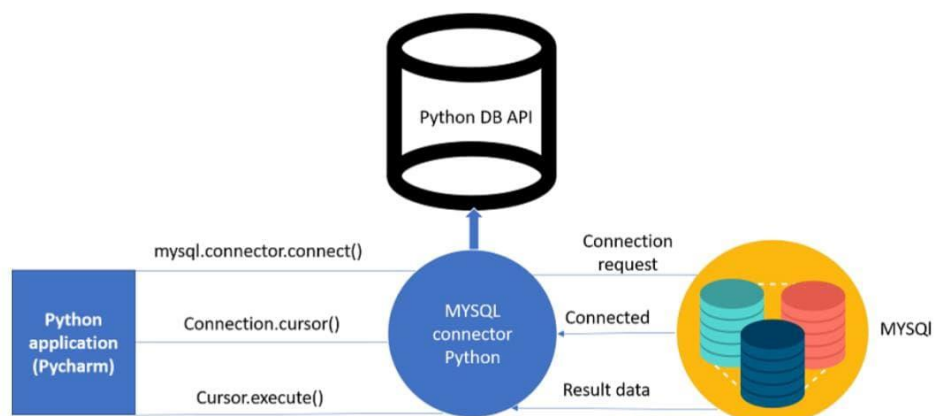


Рис.3.29. Принцип підключення бази даних

В першу чергу був встановлений драйвер *mysql.connector*, призначений для спілкування бази даних з проектом, тобто встановлення зв'язку з базою даних MySQL.

Далі був написаний програмний код для бази даних MySQL, який знаходиться в файлі *Mysql.py*. Під налаштуванням мається на увазі прописані всі необхідні методи і команди, а саме:

- метод *connect()* використаний для підключення до БД;
- метод *cursor()* використаний для створення курсора на початку списку (об'єкт, що здійснює запити та у відповідь отримує результати), за допомогою якого формується порожня БД та записуються дані;
- метод *execute()* використаний для створення: БД (за допомогою параметру '*CREATE DATABASE Restoria_db*') та таблиці *Target_web* в БД;
- метод *close()* використаний для розрива з'єднання з БД.

Розглянемо існуючі стовпчики таблиці *Target_web*: *id*, *PIP*, *Phone_number*, *Vacancy*, *Experience*, *Portfolio*.

Сценарій створення файлу *Mysql.py* можна побачити на рис. 3.30.

```
mydb = mysql.connector.connect(  
    host="localhost",  
    user="root",  
    password="08032015",  
    auth_plugin='mysql_native_password')  
  
my_cursor = mydb.cursor()  
my_cursor.execute('CREATE DATABASE Restoria_db')  
  
mydb = mysql.connector.connect(  
    host="localhost",  
    user="root",  
    password="08032015",  
    auth_plugin='mysql_native_password',  
    database = "Restoria_db")  
  
my_cursor = mydb.cursor()  
my_cursor.execute("CREATE TABLE Target_web(id INTEGER AUTO_INCREMENT PRIMARY KEY, "  
    "PIP VARCHAR(255), Phone_number VARCHAR(255), "  
    "Vacancy VARCHAR(255), Experience VARCHAR(255), "  
    "Portfolio VARCHAR(255))")  
  
mydb.close()
```

Рис.3.30. Файл *Mysql.py*

На наступному етапі була реалізована функція запису даних у створену базу даних. Для цього в основному файлі (TelegramBot.py) створено підключення RestoriaBot до сервера (рис. 3.31) і занесення даних за допомогою методу *execute()* та команди *"INSERT INTO"* (рис. 3.32).

```
mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="08032015",
    auth_plugin='mysql_native_password',
    database="Restoria_db")
my_cursor = mydb.cursor()
```

Рис.3.31. Підключення RestoriaBot до сервера

```
registration_request_call = "INSERT INTO target_web (PIP, " \
    "Phone_number, Vacancy, Experience, " \
    "Portfolio) VALUES (%s, %s, %s, %s, %s)"
record3 = (web_pip, phone_web_1, web_job, dosvid_web_1, port_web_1)
my_cursor.execute(registration_request_call, record3)
mydb.commit()
mydb.close()
```

Рис.3.32. Занесення даних до таблиці

За допомогою метода *commit()* відбувається збереження всіх записаних даних. На рис. 3.33 зображено результат проходження онлайн-співбесіди.

id	PIP	Phone_number	Vacancy	Experience	Portfolio
1	Козіна Юлія Володимирівна	+380509949007	Веб-дизайнер	1 рік	https://drive.google.com/drive/folders/1FtZCoj...
2	Івано Іван Іванович	0999999999	Таргетолог	6 місяців	https://drive/s/portfolio
NULL	NULL	NULL	NULL	NULL	

Рис.3.33. Записана інформація в БД після проходження онлайн співбесіди

ВИСНОВКИ

В результаті реалізації дипломного проектування був створений чат-бот в Telegram для маркетингового агентства «Restoria», який виконує функцію онлайн співбесіди.

Перший розділ: проаналізовано предметну область - месенджер Telegram, який є інтерфейсом для чат-ботів, основні відомості і логіку роботи ботів. Були розглянуті діючі популярні чат-боти, щоб визначити актуальну тему і функціонал для свого Telegram бота.

Другий розділ: визначено завдання і функції чат-бота, які він повинен виконувати. Також описані вимоги і умови функціонування, які були нібито обговорені з замовником. Далі для коректної роботи і створення широкого функціоналу були ретельно підібрані такі засоби і технології розробки, як мова програмування, середовище розробки, бази даних та було ознайомлено з API Telegram.

Третій розділ: представлення всіх етапів розробки бота. А саме: налагодження та реєстрація чат-бота перед створенням, розробка функціоналу та його складових, підключення до бази даних і надання результатів.

За рахунок якісного опрацювання кожної стадії при розробці і виборі відповідних технологій для отримання актуального і хорошого продукту, був створений чат-бот для проведення онлайн співбесід. Бот повністю відповідає поставленим завданням, вимогам і планованому функціоналу. RestoriaBot працює стабільно, досить швидко і підтримує одночасне функціонування з декількома користувачами.

Даний чат-бот однозначно полегшує роботу співробітникам агентства «Restoria» при пошуку кандидатури і прискорює підбір персоналу за рахунок того, що залишені контакти відносяться до тих кандидатів, які з більшою ймовірністю відповідають вимогам і підходять на ту чи іншу посаду.

СПИСОК БІБЛОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Telegram-store [Електронний ресурс] – Режим доступу: <https://ru.telegram-store.com/> (дата звернення 13.04.2021р.) – Назва з екрана.
2. Telegram [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/Telegram#Боти> (дата звернення 13.04.2021р.) – Назва з екрана.
3. Что такое чат-боты и какие они бывают [Електронний ресурс] – Режим доступу: <https://www.carrotquest.io/blog/chatbot-types/> (дата звернення 13.04.2021р.) – Назва з екрана.
4. Боты в Telegram что это такое и как они работают [Електронний ресурс] – Режим доступу: <https://sharkdevelop.com/boty-v-telegram/> (дата звернення 13.04.2021р.) – Назва з екрана.
5. ТОП-25 полезных чат-ботов для разных случаев [Електронний ресурс] – Режим доступу: <https://ilounge.ua/review/top-25-poleznyh-chat-botov-dlya-raznyh-sluchaev> (дата звернення 14.04.2021р.) – Назва з екрана.
6. ТОП-40 популярных телеграм-ботів в Україні: фінанси, шопінг і відпочинок [Електронний ресурс] – Режим доступу: <https://psm7.com/uk/news/top-40-boty-telegram-v-ukraine.html> (дата звернення 14.04.2021р.) – Назва з екрана.
7. Python – что это такое [Електронний ресурс] – Режим доступу: <https://mchost.ru/articles/chto-takoe-python/> (дата звернення 20.04.2021р.) – Назва з екрана.
8. Python [Електронний ресурс] – Режим доступу: <https://ru.wikipedia.org/wiki/Python> (дата звернення 20.04.2021р.) – Назва з екрана.
9. Pycharm [Електронний ресурс] – Режим доступу: <https://ru.wikipedia.org/wiki/PyCharm> (дата звернення 21.04.2021р.) – Назва з екрана.
10. Справочник по Bot API [Електронний ресурс] – Режим доступу: <https://tlgrm.ru/docs/bots/api> (дата звернення 22.04.2021р.) – Назва з екрана.

11. MySQL [Электронный ресурс] – Режим доступа:
<https://ru.wikipedia.org/wiki/MySQL> (дата звернения 22.04.2021р.) – Назва з екрана.

12. База данных сайта MySQL : что это такое, преимущества перед другими СУБД [Электронный ресурс] – Режим доступа:
<https://imajor.ru/razrabotka/hosting/baza-dannih-mysql> (дата звернения 22.04.2021р.) – Назва з екрана.

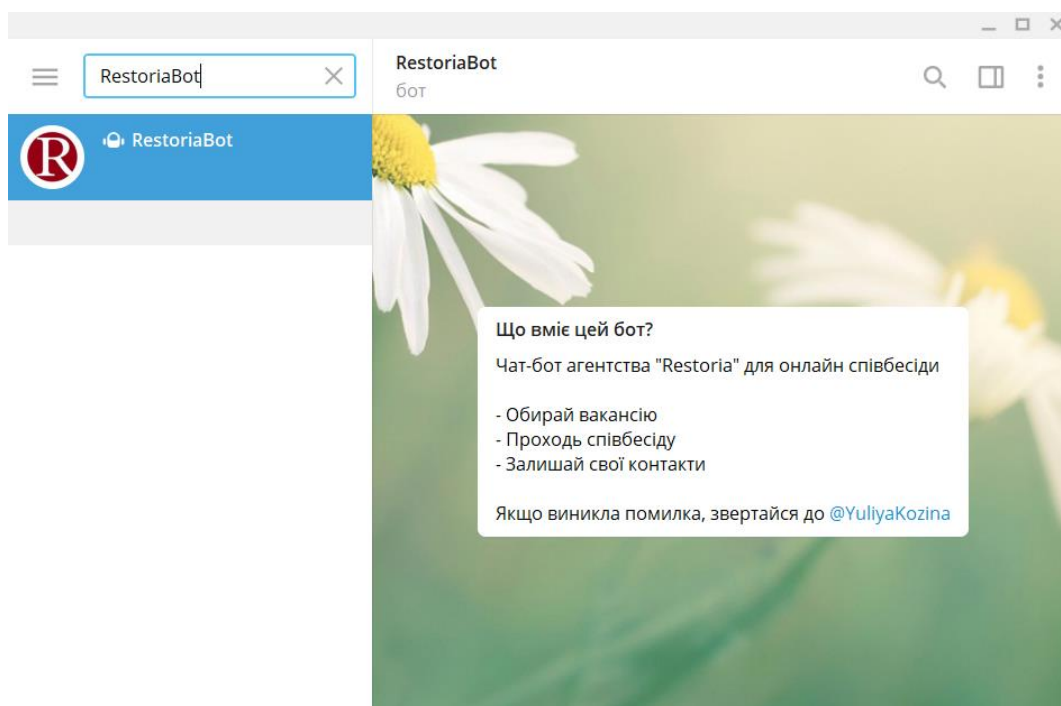


Рис. А.1. Інтерфейс RestoriaBot на Windows

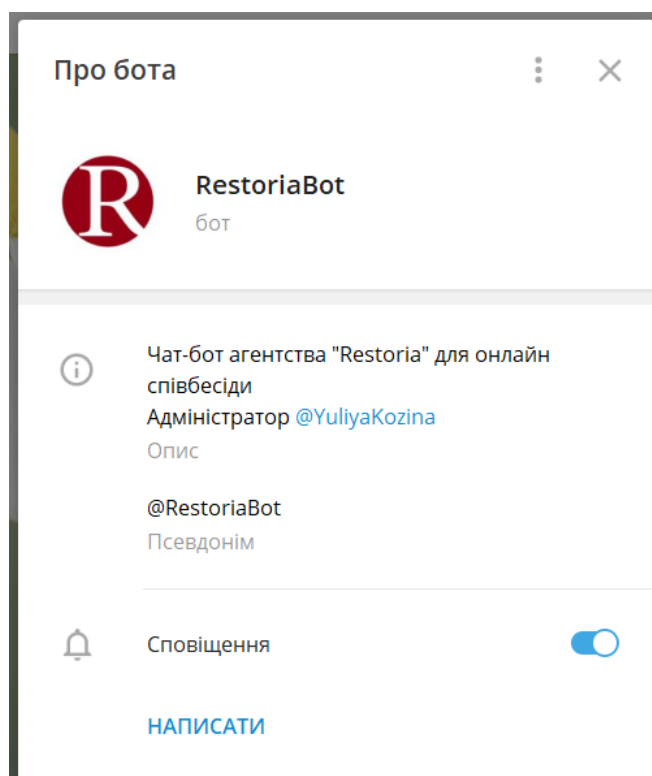
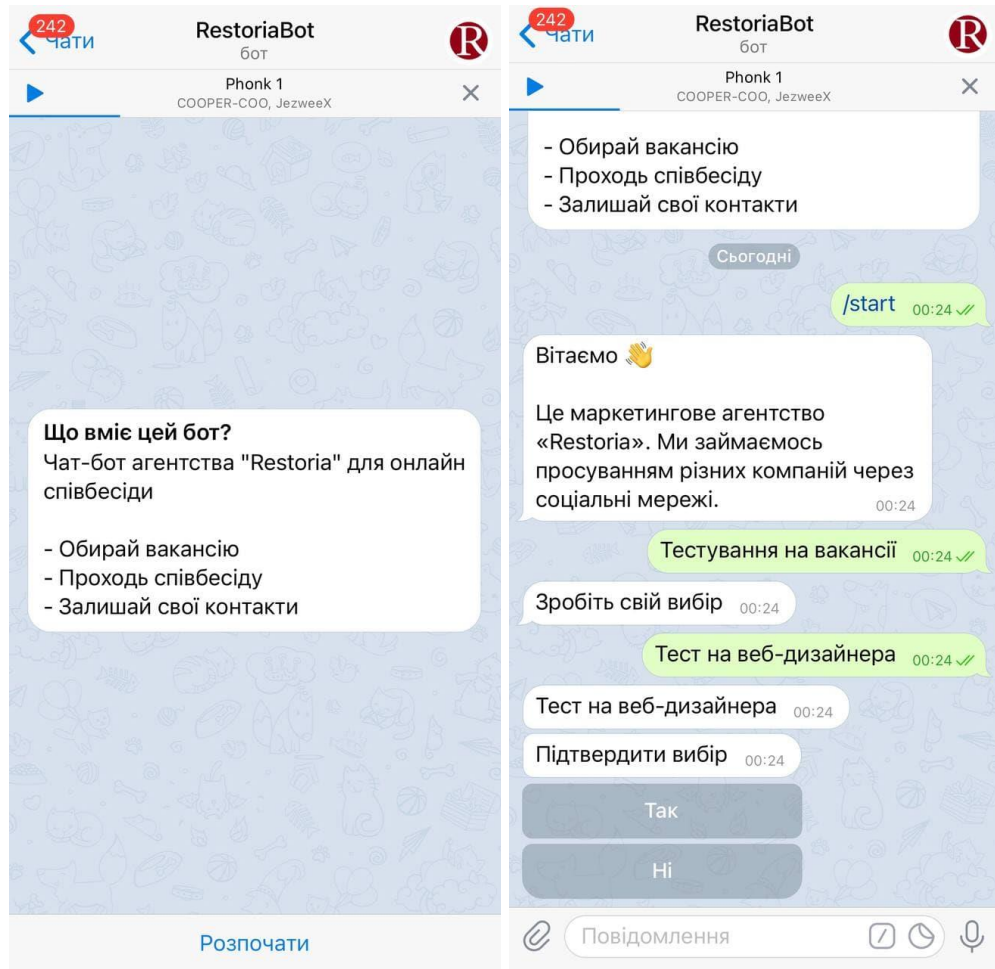


Рис. А.2. Опис бота



а)

б)

Рис. А.3. Інтерфейс RestoriaBot на Android:

а) функціонал перед початком роботи; б) процес роботи чат-бота