

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
МАРІУПОЛЬСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ЕКОНОМІКО-ПРАВОВИЙ ФАКУЛЬТЕТ
КАФЕДРА СИСТЕМНОГО АНАЛІЗУ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ**

**До захисту допустити:
В.о. зав. кафедри**




Ганна МАРТИНЮК

«23» квітня 2024 р.

«СТВОРЕННЯ TELEGRAM БОТУ ДЛЯ ОСВІТНЬОГО КУРСУ»

Кваліфікаційна робота
здобувача вищої освіти першого
(бакалаврського) рівня
освітньо-професійної програми
«Комп'ютерні науки»
Писаткова Олександра Олександровича
Науковий керівник:
Дрейс Юрій Олександрович,
кандидат технічних наук, доцент,
доцент кафедри системного аналізу та
інформаційних технологій
Рецензент:
Лукашенко Вікторія Вікторівна,
кандидат технічних наук, доцент,
заступник декана факультету
комп'ютерних наук та технологій
Національного авіаційного університету

Кваліфікаційна робота захищена
з оцінкою відмінно 100 (А)
Секретар ЕК 

«05» червня 2024 р.

Київ - 2024

МАРІУПОЛЬСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ЕКОНОМІКО – ПРАВОВИЙ ФАКУЛЬТЕТ
КАФЕДРА СИСТЕМНОГО АНАЛІЗУ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

До захисту допустити:
В.о. зав. кафедри



Ганна МАРТИНЮК

«23» квітня 2024 р.

ПЛАН ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Писатков Олександр Олександрович

1. Тема роботи «Створення Telegram боту для освітнього курсу»
керівник роботи _____

затверджені наказом Маріупольського державного університету
від « »_ 2024 року №

2. Строк подання студентом роботи

3. Вихідні дані до роботи (мета, об'єкт, предмет)

4. Зміст роботи (перелік питань, які потрібно розробити)

5. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

6. Дата видачі завдання

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Ознайомлення з темою кваліфікаційної роботи	17 – 20 Листопада	Дізнався про явище телеграм робота в цілому
2	Створення боту	21 – 22 Листопада	Створив бота в BotFather та розробив дизайн для нього
3	Отримання довідкової інформації	23 Листопада – 13 Грудня	Пошук інформації. Дізнався як само пишеться бот, які можливості він має та як їх реалізувати
4	Вибір мови програмування	12 – 14 Грудня	
5	Вибір бібліотеки	4 - 6 Січня	

6	Написання коду	6 – 27 Січня, 3 – 20 Березня	
7	Написання кваліфікаційної роботи	22 Березня – 22 Квітня	

Студент:

Писатков Олександр Олександрович

Науковий керівник роботи:

Дрейс Юрій Олександрович,
кандидат технічних наук, доцент,
доцент кафедри системного аналізу
та інформаційних технологій

Зміст

Вступ	1
РОЗДІЛ 1	7
ОГЛЯД ДОВІДКОВОЇ ІНФОРМАЦІЇ ТА ІСНУЮЧИХ РІШЕНЬ	7
1.1 Технологія створення телеграм бота	7
1.2 Які боти для вивчення Python вже існують	9
1.3 Функціонал бота	10
1.4 Вибір технології для реалізації	11
Висновки до розділу 1	13
РОЗДІЛ 2	14
ЕТАПИ РОЗРОБКИ	14
2.1 BotFather. Налаштування бота	14
2.2. Написання коду для бота	15
2.2.1 Створення та перша команда	15
2.2.2 Надсилання відеоуроків	16
2.2.3 Додатковий функціонал	17
Висновки до розділу 2	19
РОЗДІЛ 3	20
ІНТЕРФЕЙС ТА ТЕСТУВАННЯ ТЕЛЕГРАМ БОТА	20
3.1 Пошук та початок взаємодії	20
3.2 Тестування всіх команд	22
Висновки розділу 3	24
Висновок	25
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	27
ДОДАТКИ	29

ВСТУП

В сучасному світі освіти та інформаційних технологій телеграм боти стають все більш популярним засобом навчання та надання корисної інформації. З метою розвитку сучасних форматів освіти та надання зручних інструментів для навчання була розроблена кваліфікаційна робота "Створення TELEGRAM бота для освітнього курсу". Але що таке TELEGRAM бот?

Телеграм бот - це програма, яка автоматизує взаємодію з користувачами у месенджері Telegram. Ці боти розроблені з використанням API Telegram і можуть виконувати різноманітні завдання, від надсилання повідомлень до виконання складних команд та обробки даних. Ось деякі основні характеристики телеграм ботів:

1. Відповіді на повідомлення: Боти можуть реагувати на повідомлення користувачів у Telegram, надсилаючи текстові відповіді або виконуючи певні дії.

2. Команди: Користувачі можуть відправляти команди боту для виконання певних дій. Наприклад, команда `"/start"` може ініціювати роботу бота або відкривати меню взаємодії з ним.

3. Обробка даних: Боти можуть обробляти вхідні дані від користувачів, виконувати розрахунки, проводити пошук інформації та надавати результати у вигляді повідомлень або файлів.

4. Інтерактивність: Боти можуть надавати інтерактивні меню та взаємодіяти з користувачем за допомогою кнопок, вікон та інших елементів інтерфейсу.

5. Повідомлення сповіщення: Боти можуть відправляти сповіщення користувачам про події, новини або іншу важливу інформацію.

6. Інтеграція з іншими сервісами: Деякі боти можуть інтегруватися з іншими сервісами і API для отримання або обробки додаткової інформації.

Телеграм боти можуть бути використані для різних цілей, від автоматизації бізнес-процесів до розважальних інтерактивних додатків. Вони є потужним інструментом для спілкування та автоматизації у Telegram.

Як освітній курс я вибрав саме вивчення мови програмування Python, але чому?

1. Популярність Python:

На сьогоднішній день Python є однією з найпопулярніших мов програмування у світі. А згідно ТЮВЕ Python зараз є найпопулярнішою мовою програмування. Він використовується для різноманітних завдань, від веб-розробки і аналізу даних до штучного інтелекту та наукових досліджень. Популярність Python пояснюється кількома факторами:

1. Простота вивчення: Синтаксис Python легкий для розуміння, що дозволяє швидко вивчити основи програмування.

2. Велика спільнота: Python має велику та активну спільноту розробників, яка постійно розширює функціональні можливості мови і розвиває її екосистему.

3. Універсальність: Python може використовуватися для різноманітних завдань, починаючи від скриптів для автоматизації до великих веб-проектів та наукових досліджень.

4. Широкі бібліотеки: Python має велику кількість сторонніх бібліотек і фреймворків, які полегшують роботу з різними завданнями, від обробки даних до розробки і штучного інтелекту.

5. Підтримка великих компаній: Багато великих компаній, таких як Google, Facebook, Netflix та інші, використовують Python у своїх проектах, що підвищує зацікавленість розробників у вивченні та використанні мови.

Загалом, Python здобув широку популярність завдяки своїм зручностям, універсальності та активній спільноті розробників.

2. Потреба в освіті з програмування:

Потреба в освіті з програмування постійно зростає у зв'язку з розвитком технологій та цифровою трансформацією у всіх сферах життя. Ось деякі ключові моменти, які пояснюють цю потребу:

1. Ринковий попит: Компанії у всіх галузях, від технологій до медицини, шукають спеціалістів у сфері програмування для розробки програмного забезпечення, веб-сайтів, мобільних додатків тощо.

2. Цифрова грамотність: У світі, де технології відіграють все більш важливу роль, розуміння основ програмування стає ключовим елементом цифрової грамотності.

3. Швидке розвиток технологій: Штучний інтелект, машинне навчання, інтернет речей (IoT), блокчейн - всі ці технології базуються на програмуванні. Отже, розуміння програмування стає важливим для роботи з такими інноваціями.

4. Креативність та розв'язання проблем: Програмування сприяє розвитку креативності та навичок аналізу, оскільки вимагає здатності розв'язувати складні завдання та створювати нові продукти і рішення.

5. Саморозвиток і самовираження: Для багатьох людей програмування є цікавою та захоплюючою активністю, яка дає можливість самовираження та постійного саморозвитку.

Загалом, освіта з програмування стає все більш важливою для людей у всіх сферах життя, незалежно від їхніх професійних цілей, оскільки це навичка, яка може стати корисною в багатьох аспектах сучасного світу.

3. Інтерактивність:

Через телеграм бота можна надати користувачам можливість вивчати Python в інтерактивному режимі. Вони можуть отримувати завдання, виконувати код, отримувати негайну зворотню зв'язок і порівнювати свої результати

4. Доступність та зручність:

Користувачі можуть вивчати Python безпосередньо зі свого мобільного телефону або комп'ютера, використовуючи телеграм бота. Це робить процес навчання більш доступним і зручним для широкого кола людей.

5. Підвищення навичок програмування:

Навчання через практичне виконання завдань і вивчення Python коду дозволить користувачам розвивати свої навички програмування та вирішувати реальні завдання.

6. Сприяння самоосвіті:

Такий бот може стати чудовим інструментом для самоосвіти в області програмування, що дозволить користувачам вчити нове у зручний для них час і темп.

Отже, створення телеграм бота для вивчення мови Python має багато переваг і може стати корисним інструментом для навчання програмування для широкого кола користувачів.

Об'єктом дослідження даної роботи є створення та аналіз ефективності TELEGRAM бота, який допомагатиме користувачам засвоювати основи програмування мовою Python.

Предметом дослідження є створення боту на основі соціальної мережі TELEGRAM (далі – телеграм)

Цілі та завдання роботи "Телеграм бот для вивчення мови Python" можуть бути сформульовані наступним чином:

Цілі:

1. Посилення інтересу до вивчення Python: За допомогою бота ми хочемо залучити більше людей до вивчення мови програмування Python та підвищити їхній інтерес до цього процесу.

2. Навчання основ Python: Надати користувачам базові знання і навички у сфері програмування на Python.

3. Створення інтерактивного середовища: Забезпечити можливість виконання коду, отримання зворотнього зв'язку та вирішення завдань прямо в Telegram.

4. Сприяння самоосвіті: Надати користувачам можливість вчити нове у зручний для них час і темп, сприяючи самоосвіті в області програмування.

Завдання:

1. Створення освітньої програми: Розробити структуровану освітню програму, яка охоплює основні концепції та функціонал Python.

2. Розробка бота: Створити телеграм бота, який буде взаємодіяти з користувачем, надавати інформацію, виконувати завдання та аналізувати введений користувачем код.

3. Навчальні матеріали: Підготувати текстові, аудіо- та відеоматеріали для навчання, які будуть доступні користувачам через бота.

4. Практичні завдання: Створити набір практичних завдань, які допоможуть закріпити отримані знання і вміння.

5. Тестування та вдосконалення: Провести тестування бота та внести необхідні виправлення та покращення з урахуванням отриманих результатів.

6. Популяризація: Провести маркетингову роботу для просування бота серед цільової аудиторії та залучення більшої кількості користувачів.

Огляд актуальності теми "Телеграм бот для вивчення мови Python" відображається через кілька ключових пунктів:

1. Зростання популярності Python: Python продовжує залишатися однією з найбільш популярних мов програмування в світі. Його простий синтаксис, широкі можливості і велика спільнота роблять його привабливим для початківців та досвідчених програмістів.

2. Популярність месенджера Telegram: Telegram - одна з найбільш популярних месенджерів з широким функціоналом, включаючи можливість створення ботів. Його велика активна аудиторія створює потенціал для розширення освітніх можливостей через цю платформу.

3. Попит на онлайн-освіту: За останні кілька років спостерігається зростання попиту на онлайн-освіту. Люди шукають зручні та доступні способи вивчення нових навичок, включаючи програмування.

4. Потреба в інтерактивному навчанні: Інтерактивне навчання набуває все більшої популярності, оскільки дозволяє студентам отримувати негайний зворотній зв'язок та вчитися у своєму темпі.

5. Росте популярність навчальних ботів: Боти стають все більш популярними у навчальних цілях. Вони надають можливість створювати інтерактивні курси, виконувати завдання та отримувати навчальний матеріал у зручний час і форматі.

Враховуючи ці фактори, тема "Телеграм бот для вивчення мови Python" залишається актуальною і може бути досить успішною у сучасному освітньому середовищі.

РОЗДІЛ 1

ОГЛЯД ДОВІДКОВОЇ ІНФОРМАЦІЇ ТА ІСНУЮЧИХ РІШЕНЬ

1.1 Технологія створення телеграм бота

Існує кілька засобів та методів для створення телеграм ботів, які можна використовувати для розробки різноманітних функціональних можливостей. Ось декілька з них:

1. Telegram Bot API: Офіційний API Telegram надає можливість створювати ботів для Телеграму за допомогою різних мов програмування, таких як Python, JavaScript, Java, C#, і багато інших. Використання Telegram Bot API дозволяє реалізувати різноманітний функціонал, включаючи відправку повідомлень, обробку команд, реакцію на події та взаємодію з користувачами.

2. Бібліотеки для мов програмування: Існують багато бібліотек для різних мов програмування, які спрощують розробку телеграм ботів. Наприклад, для Python є бібліотеки, такі як `python-telegram-bot`, `Telepot` та `aiogram`, які надають високорівневий інтерфейс для роботи з Telegram Bot API.

3. Бот-будівники: Існують онлайн-платформи, такі як `Chatfuel`, `ManyChat`, `Votsify` та інші, які дозволяють створювати ботів без програмування. Ці платформи зазвичай надають візуальний інтерфейс для створення ботів, використовуючи блок-схеми або інші інструменти.

4. Інтегровані середовища розробки (IDE): Деякі IDE, такі як `PyCharm` для Python, мають плагіни або вбудовані інструменти для розробки телеграм ботів. Це дозволяє програмістам розробляти та тестувати ботів безпосередньо зі свого середовища розробки.

5. Хмарні платформи: Певні хмарні платформи, такі як `Amazon Web Services (AWS)`, `Google Cloud Platform (GCP)` та `Microsoft Azure`, надають інструменти для створення та розгортання телеграм ботів у хмарних середовищах.

Ці засоби та методи надають широкі можливості для розробки та розгортання телеграм ботів з різноманітним функціоналом. Вибір конкретного засобу залежить від потреб проєкту, рівня складності та власних вподобань розробника. Отримані засоби для створення телеграм ботів мають свої переваги і недоліки, які варто розглянути перед вибором конкретного рішення:

1. Telegram Bot API:

- Переваги:

- Повна контроль над функціоналом бота.
- Гнучкість у виборі мови програмування.
- Можливість взаємодії з усіма функціями Телеграму.

- Недоліки:

- Потребує програмування, що може бути складним для новачків.
- Необхідність забезпечення хостингу для розгортання бота.

2. Бібліотеки для мов програмування:

- Переваги:

- Спрощують розробку бота за рахунок високорівневого інтерфейсу.
- Мають різні функціональні можливості та підтримуються спільнотою.

- Недоліки:

- Залежність від сторонніх бібліотек, які можуть бути менш стабільними або не підтримуваними.

- Можливість обмеженого функціоналу порівняно з Telegram Bot API.

3. Бот-будівники:

- Переваги:

- Не потребують програмування, що робить їх доступними для широкого кола користувачів.

- Мають візуальний інтерфейс для створення бота.

- Недоліки:

- Обмежена гнучкість у реалізації складних функцій.
- Можливість обмеженого функціоналу порівняно з іншими рішеннями.

4. Інтегровані середовища розробки (IDE):

- Переваги:
 - Зручний інтерфейс для програмування та тестування бота.
 - Підтримка відлагодження коду, що полегшує розробку.
- Недоліки:
 - Може бути складним для новачків.
 - Необхідність налаштування середовища розробки.

5. Хмарні платформи:

- Переваги:
 - Забезпечують готове середовище для розгортання та масштабування бота.
 - Мають різні інструменти для моніторингу та управління.
- Недоліки:
 - Вимагають плати за користування хмарними ресурсами.
 - Можуть бути складними для налаштування та розгортання для новачків.

Вибір конкретного підходу до створення телеграм бота залежить від конкретних потреб проєкту, рівня складності функціоналу та ресурсів, які доступні для розробки та розгортання. Але я буду використовувати одразу декілька методів щоби збільшити функціонал бота.

1.2 Які боти для вивчення Python вже існують

На сьогоднішній день існує багато телеграм ботів для навчання програмування, які пропонують різноманітні курси, завдання, практичні вправи та можливості взаємодії з користувачами. Ось кілька прикладів таких ботів:

1. @PythonistBot: Цей бот пропонує курси з Python для початківців і досвідчених програмістів. Він містить відеоуроки, завдання та можливість вирішувати завдання прямо в чаті.

2. @CodeDogsBot: Цей бот допомагає вивчити Python та інші мови програмування через виконання завдань. Він має різні рівні складності та теми для навчання.

3. @SoloLearnBot: SoloLearn - це відома платформа для навчання програмування, яка також має свого телеграм бота. Він пропонує курси з Python, Java, JavaScript та інших мов, а також практичні завдання і відстеження прогресу.

4. @ProgForBot: Цей бот пропонує курси з програмування на різних мовах, включаючи Python, C++, Java та інші. Він містить відеоуроки, теоретичний матеріал і практичні завдання.

5. @CodewarsBot: Цей бот дозволяє розв'язувати програмістські завдання на різні теми і мови програмування, включаючи Python. Він пропонує завдання на різних рівнях складності і допомагає розвивати навички програмування.

Ці телеграм боти і багато інших надають користувачам можливість навчатися програмуванню у зручній для них час і форматі, виконуючи завдання, отримуючи навчальний матеріал і спілкуючись з іншими учасниками.

1.3 Функціонал бота

Виходячи зі списку ботів вище, ми вже можемо представити приблизний функціонал нашого майбутнього:

1. Реєстрація та авторизація:

- Користувач може створити обліковий запис та авторизуватися в системі.

- Можливість відновлення пароля у разі його втрати.

2. Навчальний контент:

- Надання навчальних матеріалів з основ Python: відеоуроки, текстові матеріали, практичні завдання тощо.

- Курси різного рівня складності: початковий, середній, продвинутий.

- Можливість відстеження прогресу в навчанні (наприклад, завдання, які було виконано, тести тощо).

3. Виконання практичних завдань:

- Можливість виконання практичних завдань прямо в чаті.

- Автоматична перевірка виконаних завдань і надання зворотного зв'язку користувачеві.

4. Взаємодія з користувачем:

- Запитання та відповіді на запитання користувачів.
- Інтерактивні уроки з можливістю вибору відповідей або виконання завдань.

5. Тестування знань:

- Можливість пройти тести з різних тем та отримати результат.
- Відстеження результатів тестування та надання рекомендацій для покращення знань.

6. Додаткові функції:

- Можливість обговорення тем з іншими користувачами у спеціальних чатах або форумах.
- Надання посилань на корисні ресурси та матеріали для поглиблення знань.

Ці функціональні вимоги допоможуть створити повноцінний телеграм бот для навчання мови програмування Python, який буде корисним для початківців і досвідчених програмістів.

1.4 Вибір технології для реалізації

Для вирішення нашого завдання є безліч способів.

JavaScript. JavaScript - мультипарадигменна мова програмування, яка підтримує об'єктно-орієнтований, імперативний та функціональні стилі. JavaScript – реалізація мови ECMAScript. JavaScript можна використовувати як мову, що вбудовується, для програмного доступу до об'єктів додатків, у нашому випадку - Telegram Bot. Найбільш широко застосовується у браузерях як мову сценаріїв для надання інтерактивності веб-сторінок. Основні архітектурні риси: слабка типізація, динамічна типізація, автоматичне керування пам'яттю, прототипне програмування, функції як об'єкти першого класу. На JavaScript вплинули багато мов, при розробці була мета зробити мову схожою на Java, але при цьому легкою для використання для тих, хто не

вміє програмувати. Мовою JavaScript не володіє певна компанія або організація, що відрізняє його від низки мов програмування, що використовуються у веб-розробці.

PHP. PHP — скриптова мова загального призначення, що застосовується для розробки веб-застосунків. В даний час підтримується переважна більшість хостинг-провайдерів і є одним з лідерів серед мов, що застосовуються для створення динамічних веб-сайтів. Мова та її інтерпретатор (Zend Engine) розробляються групою ентузіастів у рамках проекту з відкритим кодом. Проект розповсюджується під власною ліцензією, несумісною з GNU GPL.

Python

1. Установка бібліотеки дуже проста.
2. Бібліотека `pyTelegramBotAPI` сам відповідає за всі аспекти відправки та отримання запитів, дозволяє зосередитися безпосередньо на логіці.

Після налаштування задаємо пару стартових команд для нашого бота, щоб перевірити його.

Телеграм надсилає повністю повідомлення. Необхідність ставити перед командами “/” створено лише для того, щоб користувачам було зручно. Це дозволяє навчити програму відповідати по-людськи. Для того, щоб дати їй можливість отримувати у групах всі повідомлення, потрібно написати `/setprivacy` помічнику `@BotFather`, відключаючи таким чином приватність.

Далі в `Handler` потрібно додати обробник, до списку команд включити промову. Також надіслані повідомлення можуть бути не лише текстовими, до них можна прикріплювати медіафайли.

Після реалізації найпростіших команд ми отримуємо повноцінний бот, який може відповідати запити готовими командами.

У порівнянні з іншими мовами Python здався для вирішення цього завдання найзручнішим для мене. Так як у мові Python є великий вибір бібліотек, які більш мені знайомі, і є плюс у швидкості написання. Якщо навіть порівняти, як виглядає реалізація іншими мовами, на Python більш

зрозумілий синтаксис. Для Telegram бота я знайшов усі відповідні модулі та бібліотеки, які знадобляться мені у створенні цього продукту.

Висновки до розділу 1

Ми дізналися за допомогою яких інструментів робляться телеграм боти та функціонал для них, плюси та мінуси цих методів, познайомилися з подібними ботами, щоб знати чого варто прагнути, а також вибрали головний інструмент для цього – потрібну мову програмування, а саме Python. Ця інформація нам дуже допоможе і значно полегшить наше завдання

РОЗДІЛ 2

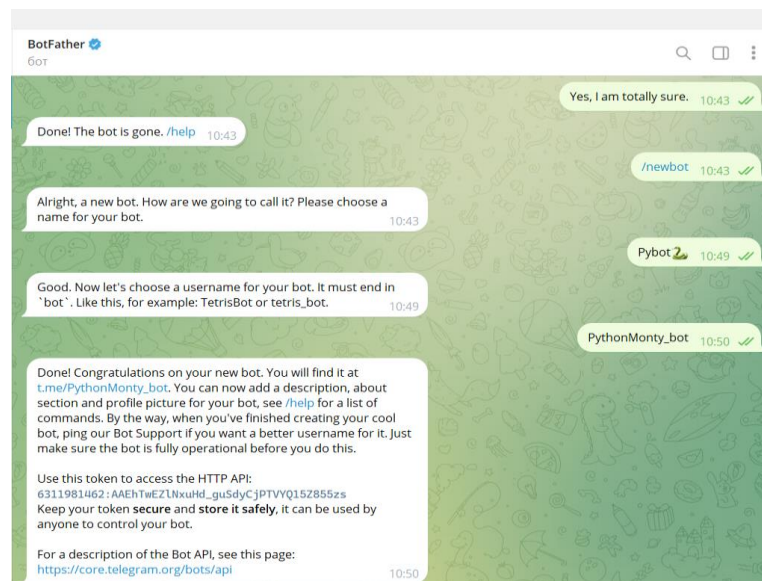
ЕТАПИ РОЗРОБКИ

2.1 BotFather. Налаштування бота

Для початку необхідно зареєструвати саму оболонку бота в самому месенджері, отримати присвоєний їй ID, який є також токеном. У телеграмі допомогу при такій реєстрації надає віртуальний помічник @BotFather.

Як це зробити:

1. Введіть команду /start для запиту інструкції.
2. Надсилання установки /newbot.
3. @BotFather пропонує вигадати ім'я для створюваного бота. Воно повинне закінчуватися словом "bot".
4. Далі помічник надсилає токен нового віртуального користувача та url для додавання його до списку контактів. Готово, у нас є токен, для подальшої роботи. (рис. 2.1.1)

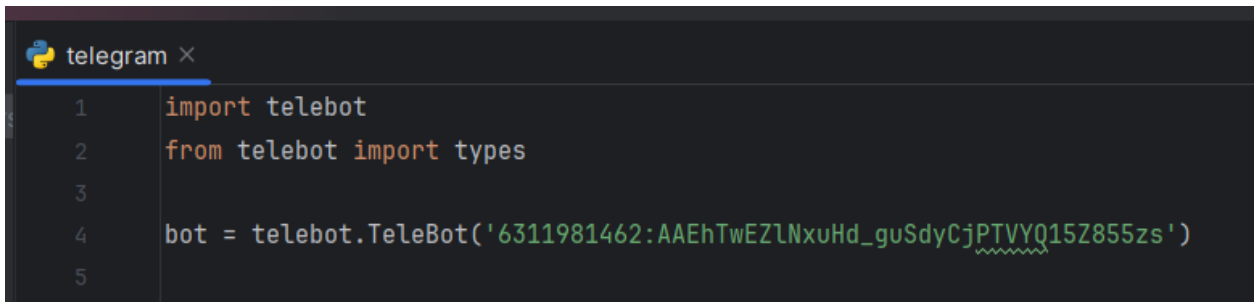


(рис. 2.1.1)

2.2. Написання коду для бота

2.2.1 Створення та перша команда

Для початку встановимо бібліотеку `pyTelegramBotAPI`. Для цього пишемо в терміналі – `pip install pyTelegramBotAPI`. Тепер підключимо її і відразу створимо змінну, до якої будемо звертатися, а в значення поставимо токен, який ми отримали від BotFather, так ми зможемо за допомогою коду звертатися безпосередньо до нашого бота (рис. 2.2.1)



```

telegram x
1  import telebot
2  from telebot import types
3
4  bot = telebot.TeleBot('6311981462:AAEhTwEZLNxuHd_guSdyCjPTVYQ15Z855zs')
5
  
```

(рис. 2.1.1)

Далі користуємося командою `@bot.message_handler(commands=['start'])`. Завдяки їй ми звертаємося до бота і говоримо йому що він повинен буде зараз зробити, відповісти, відправити якийсь файл і т.д. При цьому вказуючи, що він відповідатиме на певне слово чи команду. А саме в цьому випадку відповідати на команду `/start`. Коли людина тільки відкриє листування, у нього напис "запустити", при натисканні на яку від імені людини боту автоматично відправиться команда. Після чого бот видасть повідомлення у відповідь за допомогою команди `bot.send_message(message.chat.id,` і відправить зображення, до якого вказано шлях. Для надсилання цих повідомлень ми створили функцію, названу `start` (рис. 2.2.1)

```
@bot.message_handler(commands=['start'])
def start(message):
    markup = types.ReplyKeyboardMarkup()
    button_start = types.KeyboardButton('Почати')
    markup.row(button_start)
    bot.send_message(message.chat.id, text="Привіт 🌟 \nЯ бот, який допоможе тобі вивчити мову програмування Python."
    "\nОрієнтуватися ми будемо за планом, який зображений нижче"
    "\nПочнемо ми вже з сьогоднішнього дня 😊 \nНа початку хотів би"
    "\nпознайомити тебе з командами, що допомагають, ті, що знаходяться в лівому "
    "\nнижньому кутку нашого чату).\n/site - покаже тобі кілька сайтів, які тобі "
    "\стануть у нагоді і будуть корисні \n/plan - нагадає тобі наш шлях, через який "
    "\нам потрібно буде пройти.\n/more_info - джерела різної додаткової інформації"
    "\n/0step - /19step - команда для повернення до певного кроку в минулому, або ж, "
    "\якщо ти вже трохи знайомиш з мовою, щоб перескочити відразу до потрібного кроку."
    "\nЩоб я знав, що ти готовий до початку навчання просто напиши команду /ready 😊")
    file = open(r'C:\Users\aleks\PycharmProjects\pythonProject1\plan.JPG', 'rb')
    bot.send_photo(message.chat.id, file)
```

(рис. 2.2.1)

У повідомленні, яке відправить бот, ми відразу вказуємо команду для наступного кроку, щоб людина знала, що потрібно зробити для переходу далі, але при цьому зробила це в будь-який зручний час.

Наприкінці коду обов'язково варто вказати `bot.polling(none_stop=True)`, щоб бот не припиняв свою роботу відразу після запуску програми

2.2.2 Надсилання відеоуроків

Навчання програмування мовою Python відбуватиметься за допомогою відеоуроків. Відеоуроки можуть мати декілька переваг у порівнянні з іншими формами навчання. Ось кілька причин, чому відеоуроки можуть бути корисними:

1. Візуальність та аудіо-візуальна привабливість: Відеоуроки можуть надати візуальність та аудіо-візуальну привабливість, що допомагає залучити увагу студентів і полегшує засвоєння матеріалу.

2. Демонстрація процесу: Відео може показати реальний процес виконання завдань або використання конкретних інструментів, що дозволяє студентам краще розуміти матеріал.

3. Гнучкість у способі сприйняття: Люди мають різні стилі вивчення - одні краще засвоюють інформацію за допомогою читання, інші - за

допомогою аудіо або відео. Відеоуроки дають можливість навчатися тим, кому більше підходить аудіо-візуальна форма навчання.

4. Взаємодія та практика: Деякі відеоуроки можуть містити завдання або вправи для студентів, що сприяє взаємодії та практиці набутих знань.

Відео відправляється також як і зображення, спочатку вказується змінна з місцем розташування, після чого йде сама відправка цієї змінної. Важливо вказати час завантаження відео в телеграм командою `timeout`(рис. 2.2.2)

```
@bot.message_handler(commands=['step0'])
def what_is_python(message):
    bot.send_message(message.chat.id, text='У цьому кроці ми встановимо саму мову Python та IDE (середовище розробки), а '
                                'також познайомимося з командою виведення інформації \n'
                                'Посилання для скачування Python - https://www.python.org/ \nПосилання для '
                                'скачування PyCharm - https://www.jetbrains.com/ru-ru/pycharm/')
    video = open(r'C:\Users\aleks\PycharmProjects\pythonProject1\video1.mp4', 'rb')
    bot.send_video(message.chat.id, video, timeout=25)
    bot.send_message(message.chat.id, text='Якщо ти вже все зробив, то давай приступимо до наступного кроку, для цього '
                                'введи команду /step1 😊')
```

(рис. 2.2.2)

2.2.3 Додатковий функціонал

Не обійшлося і без додаткових функцій, а саме:

1. За допомогою команди `/plan` повторно відправитися зображення з планом вивчення, щоб щоразу не гортати на початок чату. (рис. 2.2.3)

```
@bot.message_handler(commands=['plan'])
def send_photo(message):
    file = open(r'C:\Users\aleks\PycharmProjects\pythonProject1\plan.JPEG', 'rb')
    bot.send_photo(message.chat.id, file, caption='Ти впораєшся з цим 😊')
```

(рис. 2.2.3)

2. За допомогою команди `/site` прийде повідомлення з посиланнями на різні корисні сайти, наприклад StackOverflow. (рис. 2.2.4)

```
@bot.message_handler(commands=['site'])
def website(message):
    markup = types.InlineKeyboardMarkup()
    button_1 = types.InlineKeyboardButton(text: 'Документація Python', url='https://docs.python.org/3/')
    button_2 = types.InlineKeyboardButton(text: 'Stack overflow', url='https://stackoverflow.com')
    markup.row(button_1)
    markup.row(button_2)
    bot.send_message(message.chat.id, text: '📄 Документація Python – сайт-біблія для розробників Python. Місце, де '
    'вказані всі правила написання коду та весь синтаксис, який є в мові від творців '
    'самого Python\n📄 Stack overflow - Порятунком всієї '
    'IT, саме тут знаходяться відповіді на всі твої запитання', reply_markup=markup)
```

(рис. 2.2.4)

3. За допомогою команди /more_info буде отримано повідомлення з посиланнями на додаткову інформацію, наприклад, про PEP 8 (рис. 2.2.5)

```
@bot.message_handler(commands=['more_info'])
def add_info(message):
    markup = types.InlineKeyboardMarkup()
    button_01 = types.InlineKeyboardButton(text: 'Високорівневі та низькорівневі МП 🤖',
    url='https://foxminded.ua/nyzkorivneve-programuvannia/')
    button_02 = types.InlineKeyboardButton(text: 'Стандарти написання мови і такий важливий PEP 8 🤖',
    url='https://www.vitaliyopodoba.com/2014/07/python-style-guide/')
    button_03 = types.InlineKeyboardButton(text: 'Компільовані та інтерпретовані МП 🤖',
    url='https://foxminded.ua/kompiliator-ta-interpretator/')
    markup.row(button_01)
    markup.row(button_02)
    markup.row(button_03)
    bot.send_message(message.chat.id, text: 'Тут буде знаходитись інформація з різних додаткових тем, які не важливі для '
    'самого кодингу, але важливі для влаштування на роботу або для розуміння роботи '
    'програм та мов програмування', reply_markup=markup)
```

(рис. 2.2.5)

4. При надсиланні відео, аудіо або зображення бот відповідь в залежності від типу файлу (рис. 2.2.6)

```
@bot.message_handler(content_types=['photo'])
def get_photo(message):
    bot.reply_to(message, text: 'Дякую, додам до своєї галереї')

@bot.message_handler(content_types=['video'])
def get_video(message):
    bot.reply_to(message, text: 'Дякую, але не треба надсилати мені відео')

@bot.message_handler(content_types=['audio'])
def get_audio(message):
    bot.reply_to(message, text: 'Вибач, але я не можу це прослухати')
```

(рис. 2.2.6)

Висновки до розділу 2

Було створено бот і написано код з таким функціоналом як:

Надсилання повідомлень. Бот може надсилати повідомлення з потрібною інформацією або додатковими відомостями

Надсилання відео файлів та зображень. Бот надсилає відео уроки та різні зображення

Надсилання посилань. Бот може надіслати посилання на документацію Python, StackOverflow

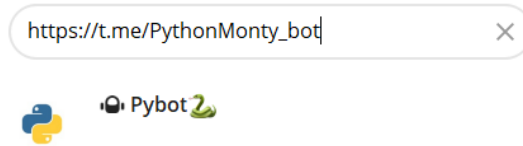
Реакція на надсилання йому відео, аудіо або зображень людиною. При надсиланні зображення бот відповість, що обов'язково збереже це у себе в галереї

РОЗДІЛ 3

ІНТЕРФЕЙС ТА ТЕСТУВАННЯ ТЕЛЕГРАМ БОТА

3.1 Пошук та початок взаємодії

1. Знаходимо бот у пошуку за адресою @PythonMonty_bot (рис.3.1.1)



(рис.3.1.1)

2. Натискаємо кнопку запустити, або ж вводимо команду /start (рис.3.1.2)



(рис.3.1.2)

Після чого отримаємо повідомлення(рис.3.1.3)

Pybot 🐍

бот

Привіт 🙌

Я бот, який допоможе тобі вивчити мову програмування Python. Орієнтуватися ми будемо за планом, який зображений нижче. Почнемо ми вже з сьогоднішнього дня 🤗

На початку хотів би познайомити тебе з командами, що допомагають, ті, що знаходяться в лівому нижньому кутку нашого чату).

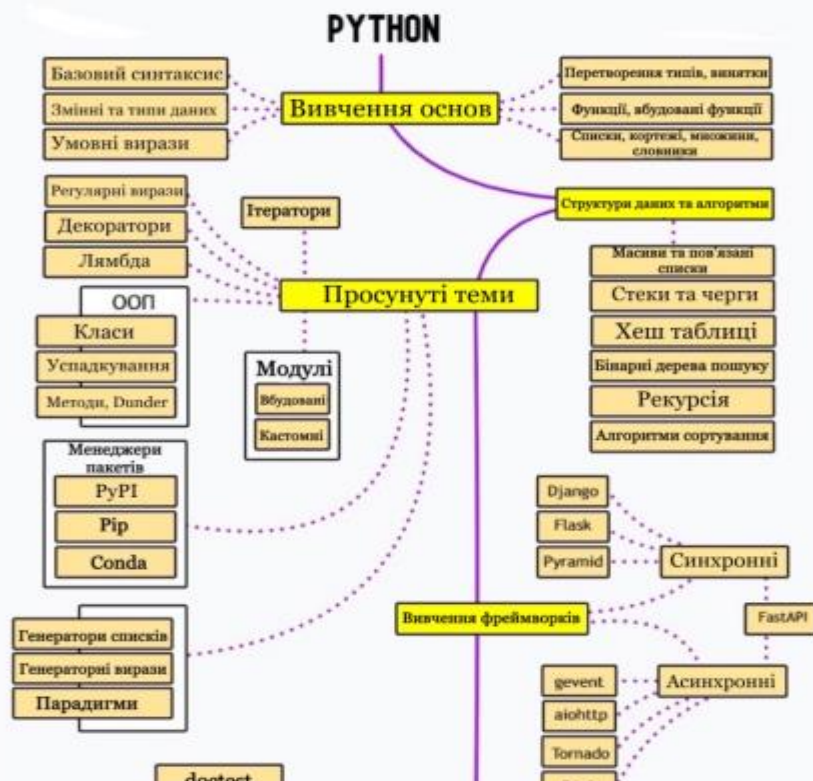
`/site` - покаже тобі кілька сайтів, які тобі стануть у нагоді і будуть корисні

`/plan` - нагадає тобі наш шлях, через який нам потрібно буде пройти.

`/more_info` - джерела різної додаткової інформації

`/0step` - `/19step` - команда для повернення до певного кроку в минулому, або ж, якщо ти вже трохи знайомий з мовою, щоб перескочити відразу до потрібного кроку. Щоб я знав, що ти готовий до початку навчання просто напиши команду `/ready` 😊

11:18

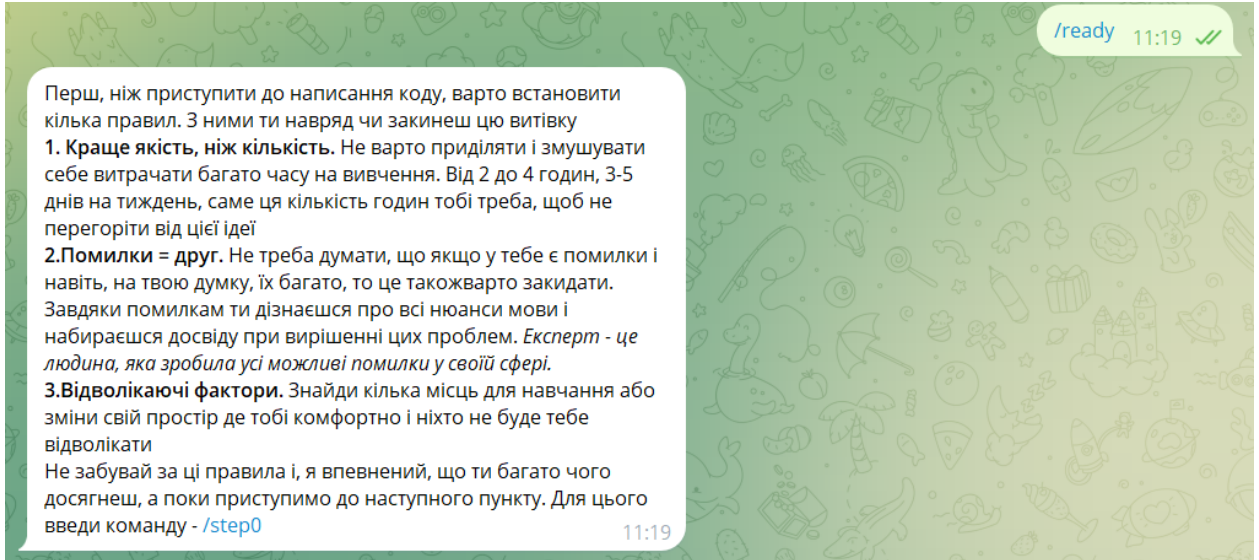


(рис.3.1.1)

3.2 Тестування всіх команд

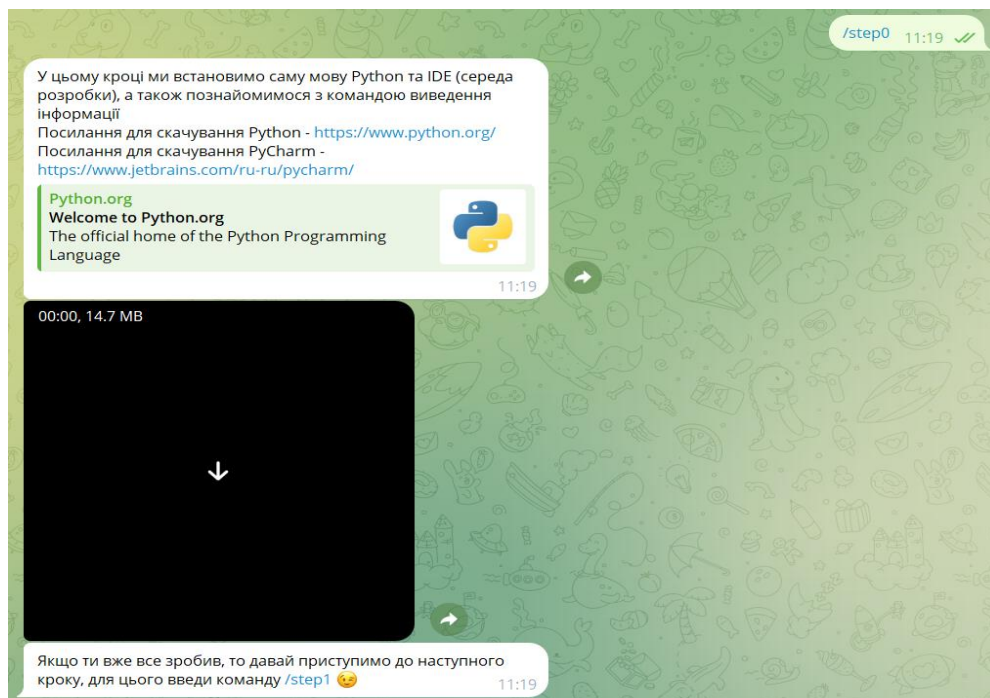
Спочатку йде перевірка основного курсу, а саме:

1. Команда /ready (рис.3.2.1)

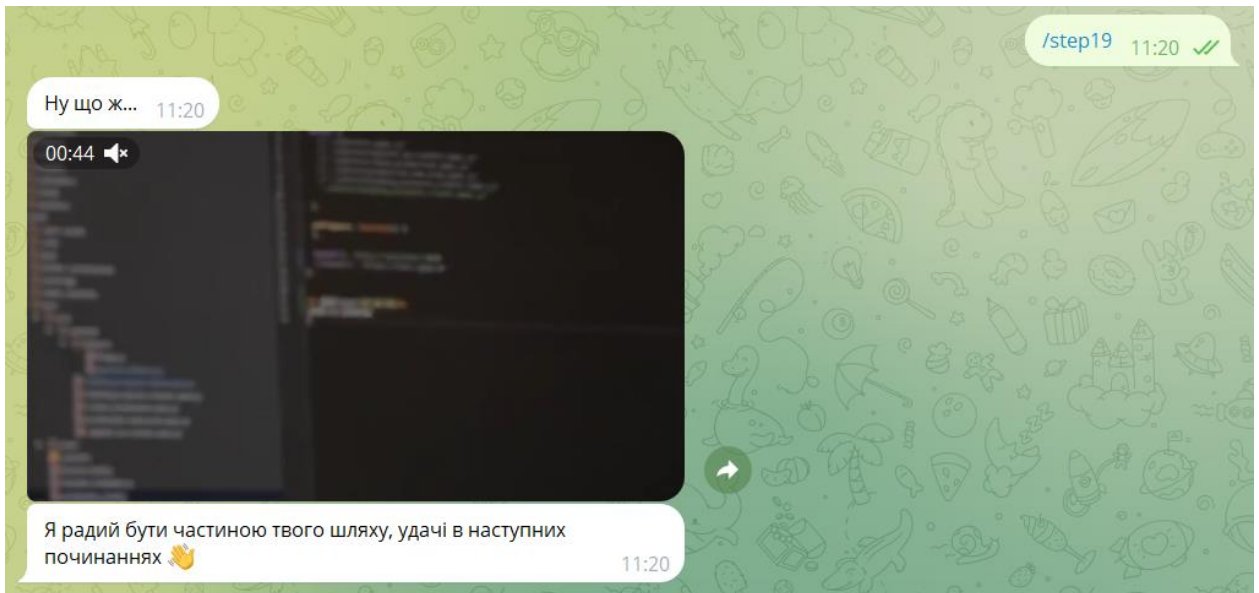


(рис.3.2.1)

2. Команди /step0 – /step19 (рис.3.2.2 – 3.2.3)



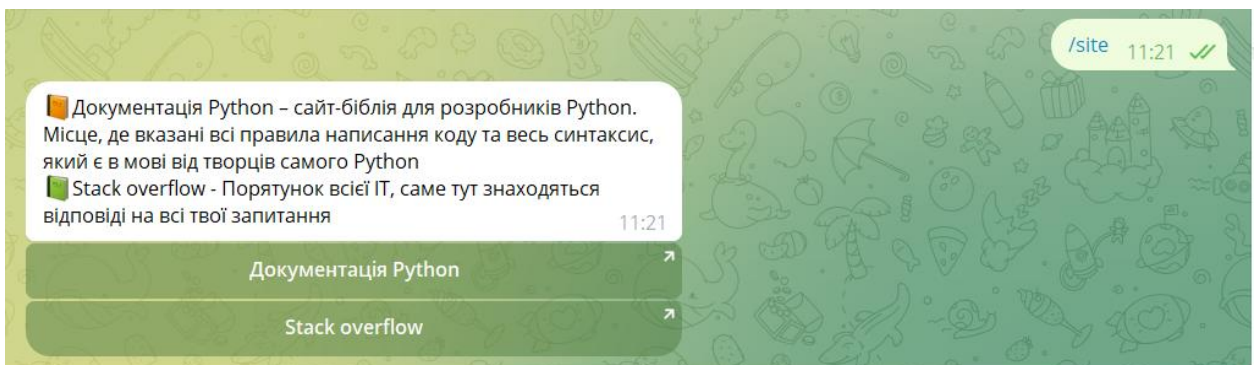
(рис.3.2.2)



(рис. 3.2.3)

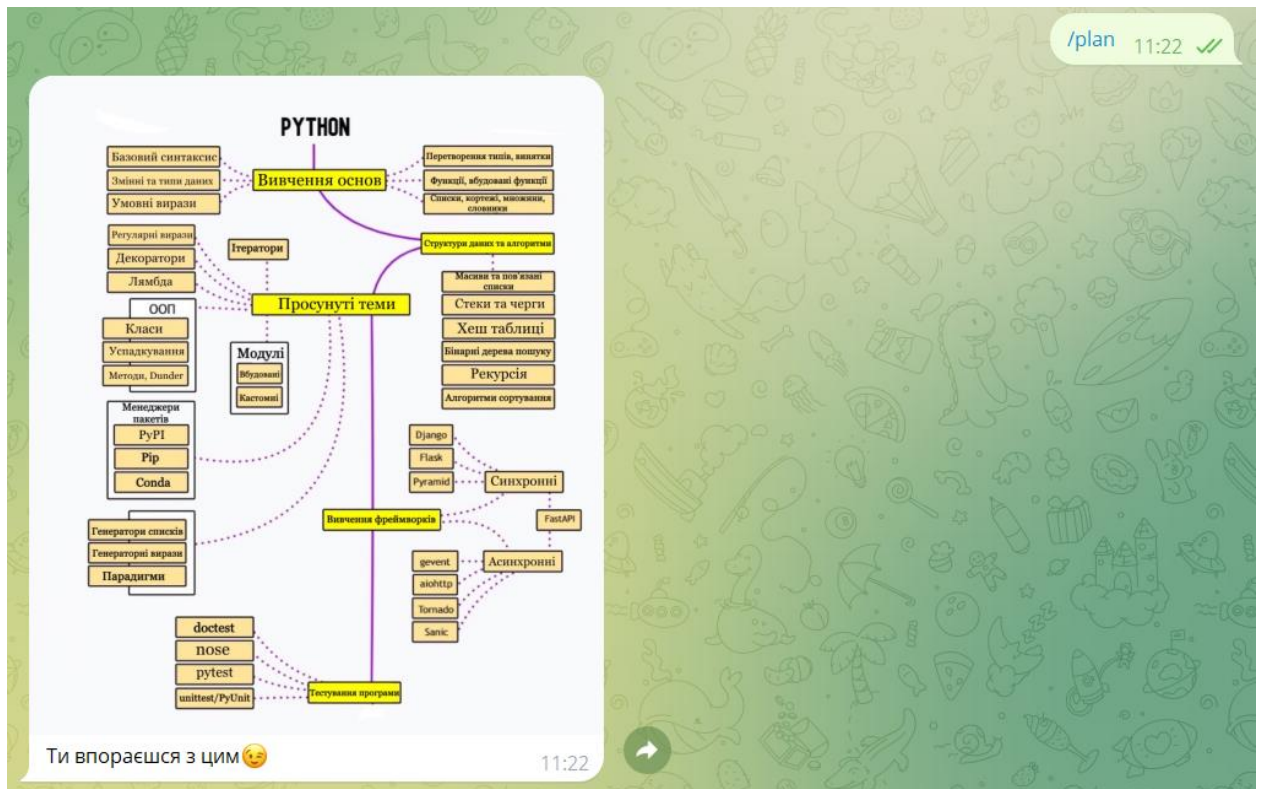
Всі повідомлення, відео та посилання працюють, як щодо додаткового функціоналу?

3. Команда /site (рис.3.2.4)



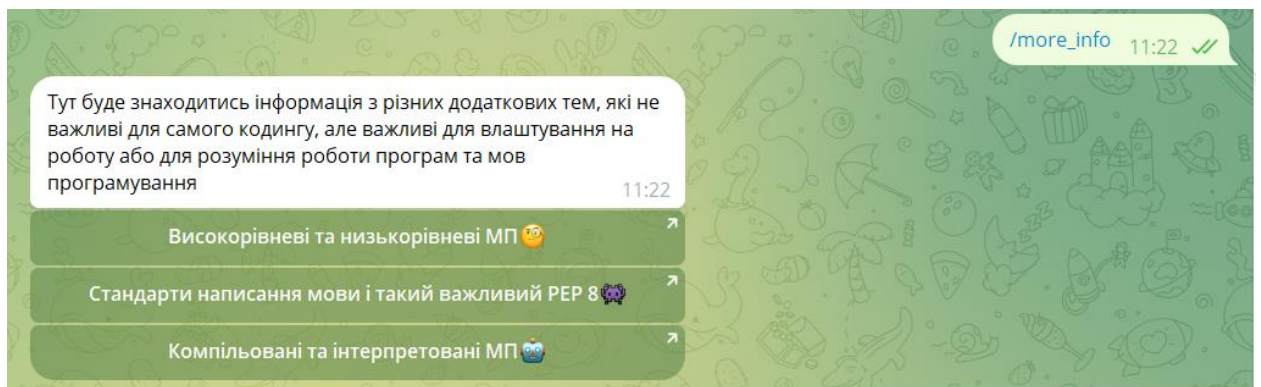
(рис.3.2.4)

4. Команда /plan (рис.3.2.5)



(рис.3.2.5)

5. Команда /more_info (рис.3.2.6)



(рис.3.2.6)

Висновки розділу 3

Telegram бот працює без помилок, відповідає на кожну команду, відсилає всі файли, які повинен, а також відправляє посилання, все це робиться без проблем і будь-яких затримок

ВИСНОВОК

У результаті проведеного дослідження та розробки телеграм бота для вивчення мови Python можна зробити наступні висновки:

Телеграм бот виявився ефективним та зручним засобом для навчання мови Python. Він забезпечує користувачам доступ до навчального контенту у форматі чату, що дозволяє навчатися у будь-який зручний час та місце. Результати дослідження свідчать про потенціал використання телеграм ботів для навчання програмування та підтримки студентів у процесі вивчення мови Python.

Подальший розвиток та покращення функціональності та можливостей телеграм бота для вивчення мови Python може бути спрямований на наступні напрями:

1. Розширення вмісту: Додавання нових уроків, завдань та практичних прикладів для більш повного охоплення різних аспектів програмування на Python.

2. Розробка функціоналу для індивідуалізованого навчання, такого як адаптивні уроки, які враховують рівень знань користувача та пропонують матеріал відповідно до його потреб.

3. Завдання зі спільнотою: Створення можливості для користувачів ділитися своїми власними проектами та кодом, обговорювати їх із спільнотою, а також отримувати фідбек від інших користувачів.

4. Інтерактивність та гейміфікація: Впровадження елементів гейміфікації, таких як нагороди, досягнення та лідерські таблиці, для стимулювання користувачів до активного навчання та участі.

5. Використання штучного інтелекту: Використання штучного інтелекту для створення персоналізованих рекомендацій та аналізу прогресу користувачів, а також для автоматизації підтримки та відповідей на питання користувачів.

6. Інтеграція з іншими ресурсами: Підключення до інших онлайн-ресурсів для навчання, таких як курси на платформах відомих університетів або веб-сайти з практичними завданнями та прикладами.

7. Мультимовний підхід: Розробка можливості вибору мови для навчання, що дозволить розширити аудиторію та зробити бота доступним для користувачів з різних країн.

Ці напрями можна поєднати для створення більш ефективного та захоплюючого середовища для навчання програмування на Python через телеграм бота.

Отже, розроблений телеграм бот є ефективним та перспективним інструментом для вивчення мови програмування Python та заслуговує на увагу в контексті сучасних методів навчання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Уроки Python для початківців. Програмування мовою Python з нуля URL: <https://www.youtube.com/playlist?list=PLDyJYA6aTY11PWXBPk0gw6gR8fEtPDGKa>.
2. Розробка Telegram Ботів на Python з нуля / Телеграм Bot для початківців URL: https://www.youtube.com/playlist?list=PL0lO_mIqDDFUev1gp9yEwmwcy8SicqKbt
3. Індекс TIOBE є показником популярності мов програмування URL: <https://www.tiobe.com/tiobe-index/>
4. Бібліотека для створення телеграм боту URL: <https://pypi.org/project/pyTelegramBotAPI/>
5. Що таке чат-бот URL: <https://creativesmm.com.ua/shho-take-chatbot-ta-komu-vonu-potribni/>
6. Все, про що повинен знати розробник телеграм бот URL: <https://habr.com/>
7. 7.Requests в Python – приклади виконання HTTP запитів. URL: <https://python-scripts.com/>
8. Довідник по BotAPI. URL: <https://tlgrm.ru/docs/bots/api>
9. Jet Brains. IDE для професійної розробки на Python. URL: <https://www.jetbrains.com/pycharm/>
10. Credence Research. Прогнозування ринку чат-ботів. URL: <https://www.credenceresearch.com/report/chatbots-market>
11. Ріст популярності чат-ботів URL: <https://forklog.com/sp/chatbots/>
12. Спілкування з чат-ботом. URL: <https://evergreens.com.ua/ru/articles/talking-bots.html>
13. 13 Статистика людей, які надають перевагу спілкуванню з чат-ботом. URL: <https://www.ubisend.com/insights/chatbot-statistics-cheatsheet>
14. Переваги чат-ботів. URL: <https://ideadigital.agency/blog/top-10->

[prichin-chomu-chat-boti-u-sotsmerezahpotribni-kozhnomu/](#)

15. Функції чат-ботів. URL: <https://www.carrotquest.io/chatbot/chatbot-types/>

16. Способи розробки чат-ботів. URL: <https://evergreens.com.ua/ru/articles/create-bot.html>

17. Переваги Python. URL: <https://xn----7sbbaqh1km9ah9aiq.net/ua/news-new/preimushchestva-i-nedostatkisayta-na-python.html>

18. Створення сценарію для чат-боту. URL: <https://bot.konveier.com/poleznoye/kak-sostavit-stsenarijdialoga-dlya-chat-bota/>.

19. Створення чат-бота за допомогою @BotFather. URL: <https://lessondelivery.com/telegram/botfather.html>.

20. Види кнопок у Telegram. URL: <https://mastergroosha.github.io/telegram-tutorial-2/buttons/>.

ДОДАТКИ

```

import telebot
from telebot import types

bot =
telebot.TeleBot('6311981462:AAEhTwEZlNxuHd_guSdyCjPTVYQ
15Z855zs')

@bot.message_handler(commands=['start'])
def start(message):
    markup = types.ReplyKeyboardMarkup()
    button_start = types.KeyboardButton('Почати')
    markup.row(button_start)
    bot.send_message(message.chat.id, "Привіт 🤖 \nЯ
бот, який допоможе тобі вивчити мову програмування
Python."
                                "\nОрієнтуватися
ми будемо за планом, який зображений нижче"
                                "\nПочнемо ми вже
з сьогоднішнього дня 🕒 \nНа початку хотів би"
                                "познайомити тебе
з командами, що допомагають, ті, що знаходяться в
лівому "
                                "нижньому кутку
нашого чату).\n/site - покаже тобі кілька сайтів, які
тобі "
                                "стануть у нагоді
і будуть корисні \n/plan - нагадає тобі наш шлях, через
який "
                                "нам потрібно
буде пройти.\n/more_info - джерела різної додаткової
інформації"
                                "\n/0step -
/19step - команда для повернення до певного кроку в
минулому, або ж, "
                                "якщо ти вже
трохи знайомий з мовою, щоб перескочити відразу до
потрібного кроку."
                                "Щоб я знав, що
ти готовий до початку навчання просто напиши команду
/ready 😊")
    file =
open(r'C:\Users\aleks\PycharmProjects\pythonProject1\pl

```

```

an.JPG', 'rb')
    bot.send_photo(message.chat.id, file)

@bot.message_handler(commands=['ready'])
def before_start_step(message):
    bot.send_message(message.chat.id, 'Перш, ніж
приступити до написання коду, варто встановити кілька
правил. З '
                                'ними ти навряд
чи закинеш цю витівку <b>\n1. Краще якість, ніж '
                                'кількість.</b>
Не варто приділяти і змушувати себе витратити багато
часу на '
                                'вивчення. Від 2
до 4 годин, 3-5 днів на тиждень, саме ця кількість
годин тобі '
                                'треба, щоб не
перегоріти від цієї ідеї\n<b>2.Помилки = друг.</b> Не
треба '
                                'думати, що якщо
у тебе є помилки і навіть, на твою думку, їх багато, то
це також'
                                'варто закидати.
Завдяки помилкам ти дізнаєшся про всі нюанси мови і
набираєшся '
                                'досвіду при
вирішенні цих проблем. <em>Експерт – це людина, яка '
                                'зробила усі
можливі помилки у своїй сфері.</em> \n<b>3.Відволікаючі
фактори.'
                                '</b> Знайди
кілька місць для навчання або зміни свій простір де
тобі '
                                'комфортно і
ніхто не буде тебе відволікати\nНе забувай за ці
правила і, я '
                                'впевнений, що ти
багато чого досягнеш, а поки приступимо до наступного
пункту. '
                                'Для цього введи
команду - /step0', parse_mode='html')

@bot.message_handler(commands=['step0'])

```

```

def what_is_python(message):
    bot.send_message(message.chat.id, 'У цьому кроці ми
встановимо саму мову Python та IDE (середя розробки), а
'
                                'також
познайомимося з командою виведення інформації \n'
                                'Посилання для
скачування Python - https://www.python.org/ \nПосилання
для '
                                'скачування
PyCharm - https://www.jetbrains.com/ru-ru/pycharm/')
    video =
open(r'C:\Users\aleks\PycharmProjects\pythonProject1\vi
deo1.mp4', 'rb')
    bot.send_video(message.chat.id, video, timeout=25)
    bot.send_message(message.chat.id, 'Якщо ти вже все
зробив, то давай приступимо до наступного кроку, для
цього '
                                'введи команду
/step1 😊')

@bot.message_handler(commands=['step1'])
def step_1(message):
    bot.send_message(message.chat.id, 'У цьому уроці ми
розберемо деякі базові операції у мові Python')
    video =
open(r"C:\Users\aleks\PycharmProjects\pythonProject1\vi
deo2.mp4", 'rb')
    bot.send_video(message.chat.id, video, timeout=25)
    bot.send_message(message.chat.id, 'Для наступного
кроку введи команду /step2, до речі, а ти знав, що
Python '
                                "'з'явився із
захоплення? Якось голландський розробник Гвідо ван
Россум думав, "
                                "'чим зайнятися в
передріздвяний тиждень, і зупинився на нетривіальному'
                                'хобі - створення
нової ІТ-мови. Було це у грудні 1989 року. Але не
тільки '
                                "'бажання
розважитися рухало програмістом. Гвідо працював у
компанії, яка '
                                "'потребувала мови

```

```

сценаріїв – так розробник вирішив зробити внесок у
компанію. '
'Незабаром кодер
показав напрацювання колегам. Вже в 1991 році проект
отримав '
'успіх. А ти теж
вигадуєш мову програмування, коли нічим зайнятися?')

@bot.message_handler(commands=['step2'])
def step_2(message):
    bot.send_message(message.chat.id, 'У цьому уроці ми
будемо працювати зі змінними та вивчимо типи даних, які
'
'є в мові
Python')
    video =
open(r"C:\Users\aleks\PycharmProjects\pythonProject1\vi
deo3.mp4", 'rb')
    bot.send_video(message.chat.id, video, timeout=25)
    bot.send_message(message.chat.id, 'Для наступного
кроку введи команду /step3, до речі, а ти знав, що
Python - '
'це не про зміїв?
Може здатися, що найменування мови пов'язане з
рептиліями, "
'але це не так.
Гвідо ван Россум вибрав назву на честь англійської
комік-трупі '
'«Монті Пайтон»,
шанувальником якої був. Якщо ти не дивився скетч-
серіал, '
'не біда: як
написано в документації, захопленість «Літаючим цирком
Монті'
'Пайтон' – не
обов'язкова умова використання Python. Проте документ "
'стверджує, що
любов до тв-шоу допомагає. Як конкретно, на жаль, не '
'вказано.
Більшість людей, як і раніше, асоціює мову зі зміями. З
цим, '
'мабуть, уже
нічого не вдієш.')

```

```

@bot.message_handler(commands=['step3'])
def step_3(message):
    bot.send_message(message.chat.id, 'У ході цього уроку ми розглянемо різні умовні конструкції, які '
                                     'присутні в мові Python')
    video =
open(r'C:\Users\aleks\PycharmProjects\pythonProject1\video4.mp4', 'rb')
    bot.send_video(message.chat.id, video, timeout=25)
    bot.send_message(message.chat.id, 'Для наступного кроку введи команду /step4, до речі, а ти знав, що на емблемі '
                                     'Python зображені саме змії. Плазуни утворюють квадрат. Склав логотип брат '
                                     'Гвідо, дизайнер Юст ван Россум. Шрифт тексту також винайшов він. До 2006 '
                                     'року логотип Пітона був текстовим. Але кобри різних видів у книгах, журналах і '
                                     'на сайтах про Python підштовхнули до зміни лого щоб уникнути плутанини')

@bot.message_handler(commands=['step4'])
def step_4(message):
    bot.send_message(message.chat.id, 'Під час цього уроку ми навчимося працювати з циклами, а також з операторами '
                                     'циклів у мові Python')
    video =
open(r"C:\Users\aleks\PycharmProjects\pythonProject1\video5.mp4", 'rb')
    bot.send_video(message.chat.id, video, timeout=25)
    bot.send_message(message.chat.id, 'Для наступного кроку введи команду /step5, до речі, а ти знав, що Python '
                                     'славиться мультиплатформенністю. Він експлуатується всюди: від смартфонів та ПК')

```

```

        ' до серверних
мереж. Мова підтримує роботу з операційними системами
Windows, '
        'Linux, MacOS,
Symbian, Android. Результативність запуску на
платформах '
        'одноманітна.
Розробка додатків, машинне навчання, ІІ, робота з
великими '
        'даними,
мікропроцесори - о так, Пітон всюдисущий.')

@bot.message_handler(commands=['step5'])
def step_5(message):
    bot.send_message(message.chat.id, 'Під час цього
уроку ми навчимося працювати зі списками даних в мові
Python, '
        'а також вивчимо
методи для роботи з ними')
    video =
open(r"C:\Users\aleks\PycharmProjects\pythonProject1\vi
deo6.mp4", 'rb')
    bot.send_video(message.chat.id, video, timeout=25)
    bot.send_message(message.chat.id, 'Для наступного
кроку введи команду /step6, до речі, а ти знав, що
Python '
        'має Python має
різновиди ІТ-мов С і Java. Варіант С відомий як
CPython. Він дав '
        'Python гідності
С - наприклад, продуктивність, комфорт та шанс
компілювати '
        'програми мовою
С. Продуктивна Java-версія Python називається Jython.
Мова '
        'дозволяє
імпортувати класи Java в Python. Існують інші
реалізації Пайтона: '
        'IronPython,
Brython, RubyPython, PyPy, MicroPython.')

@bot.message_handler(commands=['step6'])
def step_6(message):

```

```

    bot.send_message(message.chat.id, 'У ході цього
уроку ми поговоримо про рядки, які функції існують для
'
                                'роботи з
рядками, а також вивчимо тему індексів та зрізів')
    video =
open(r"C:\Users\aleks\PycharmProjects\pythonProject1\vi
deo7.mp4", 'rb')
    bot.send_video(message.chat.id, video, timeout=25)
    bot.send_message(message.chat.id, 'Для наступного
кроку введи команду /step7, до речі, а ти знав, що
найбільші '
                                'підприємства
світу застосовують Пітон для розробок. Netflix та NASA,
Spotify та '
                                'Amazon, Mozilla
і YouTube, Microsoft і Pinterest - ці та інші
корпорації '
                                'експлуатують
мову. В Yahoo Python використовується в багатьох
програмних '
                                'продуктів:
TurboGears, Django, Zope. А ще Пітон - одна з офіційних
мов '
                                'веб-розробників
Google')

@bot.message_handler(commands=['step7'])
def step_7(message):
    bot.send_message(message.chat.id, 'Під час цього
уроку ми поговоримо про кортежі у мові Python')
    video =
open(r"C:\Users\aleks\PycharmProjects\pythonProject1\vi
deo8.mp4", 'rb')
    bot.send_video(message.chat.id, video, timeout=25)
    bot.send_message(message.chat.id, 'Для наступного
кроку введи команду /step8, до речі, а ти знав, що
Python '
                                'обігнав
французьку. Несподівано, але це так. В результаті
проведеного в Англії '
                                'опитування
Python виявився привабливим для вивчення в початкових
британських '

```

```

        'школах мовою.
60% батьків бажать, щоб школярі освоювали кодинг на
Пітоні, '
        'а не іноземну
мову.')

@bot.message_handler(commands=['step8'])
def step_8(message):
    bot.send_message(message.chat.id, 'Під час цього
уроку ми навчимося працювати зі словниками Python, а
також '
        'вивчимо функції
для роботи з ними')
    video =
open(r"C:\Users\aleks\PycharmProjects\pythonProject1\vi
deo9.mp4", 'rb')
    bot.send_video(message.chat.id, video, timeout=25)
    bot.send_message(message.chat.id, 'Для наступного
кроку введи команду /step9, до речі, а ти знав, що таке
'
        'інтерпретатор?
Інтерпретатор є програмою, яка виконує код, написаний '
        'мовою
програмування. Вона не переводить його в машинні коди
повністю, а '
        'рядково приймає
команди і одразу виконує їх. Можна віддати
інтерпретатору '
        'команду й одразу
зрозуміти, чи спрацювала вона.')

@bot.message_handler(commands=['step9'])
def step_9(message):
    bot.send_message(message.chat.id, 'Під час цього
уроку ми вивчимо множини Python, set, а також
frozenset')
    video =
open(r"C:\Users\aleks\PycharmProjects\pythonProject1\vi
deo10.mp4", 'rb')
    bot.send_video(message.chat.id, video, timeout=25)
    bot.send_message(message.chat.id, 'Для наступного
кроку введи команду /step10, до речі, інтерпретатор це
один з')

```



```

"двох способів
перекладу коду в зрозумілий комп'ютеру вигляд (а саме
переклад в "
        'двійковий код -
0 та 1). Другий варіант - компіляція, коли код
«переводять» в '
        'машинні команди
повністю. Мови програмування, які працюють з
інтерпретаторами, '
        'називаються
інтерпретованими. Більшість популярних зараз мов саме
такі. Для '
        'більш детальної
інформації на цю тему введи команду - /more_info')

@bot.message_handler(commands=['step10'])
def step_10(message):
    bot.send_message(message.chat.id, 'У ході цього
уроку ми навчимося створювати, а також використовувати
функції '
        'Python\n Але
спочатку... \nТи молодець, ти вже пройшов величезну
частину шляху і'
        ' я щиро радий
твоєму прогресу. Продовжуй у тому ж темпі і ти всього
досягнеш!')
    video =
open(r"C:\Users\aleks\PycharmProjects\pythonProject1\vi
deo11.mp4", 'rb')
    bot.send_video(message.chat.id, video, timeout=25)
    bot.send_message(message.chat.id, 'Для наступного
кроку введи команду /step11, до речі, ти знав, що
Python '
        'є високорівневою
мовою програмування? Що це означає? Насправді, '
        'це дуже велика
тема, коротко - такі мови спеціально розроблені для '
        'швидкості та
зручності використання. Тобі не треба знати будову
"заліза", щоб '
        'користуватися
мовами високого рівня програмування. Більш детально ти
можеш '
        'дізнатися про це

```



```

        video =
open(r"C:\Users\aleks\PycharmProjects\pythonProject1\vi
deo15.mp4", 'rb')
        bot.send_video(message.chat.id, video, timeout=25)
        bot.send_message(message.chat.id, 'Для наступного
кроку введи команду /step15, до речі, ще з '
                                'найпопулярніших
Бібліотек для Python є OpenCV-Python. Це бібліотека, '
                                'що
використовується для обробки зображень під час роботи з
машинним зором. Вона '
                                ' містить безліч
функцій обробки зображень, таких як одночасне читання '
                                ' і запис
зображень, перетворення двовимірного середовища в
тривимірне, а також '
                                'захоплення та
аналіз зображень з відео.')

@bot.message_handler(commands=['step15'])
def step_15(message):
    bot.send_message(message.chat.id, 'У ході цього
уроку ми поговоримо і розберемося в такому понятті як
ООП - '
                                "об'єктно-
орієнтоване програмування")
    video =
open(r"C:\Users\aleks\PycharmProjects\pythonProject1\vi
deo16.mp4", 'rb')
    bot.send_video(message.chat.id, video, timeout=25)
    bot.send_message(message.chat.id, 'Для наступного
кроку введи команду /step16, до речі, ще однією з '
                                'найпопулярніших
бібліотек для Python є Keras. Це бібліотека глибокої '
                                'нейронної мережі
Python з відмінними функціями обробки даних та
візуалізації. '
                                'Keras підтримує
безліч нейронних мереж. Бібліотека має модульну
структуру, що '
                                'забезпечує
гнучкість при написанні інноваційних додатків.')

```

```

@bot.message_handler(commands=['step16'])
def step_16(message):
    bot.send_message(message.chat.id, 'Під час цього уроку ми вивчимо конструктори у класах, а також вивчимо тему'
                                     'перевизначення методів ')
    video =
open(r"C:\Users\aleks\PycharmProjects\pythonProject1\video17.mp4", 'rb')
    bot.send_video(message.chat.id, video, timeout=25)
    bot.send_message(message.chat.id, 'Для наступного кроку введи команду /step17, до речі, ще з '
                                     'найпопулярніших бібліотек для Python є NumPy. Це популярна бібліотека, '
                                     'використовувана розробниками для простого створення масивів та управління ними, '
                                     'також управління логічними фігурами та виконання операцій лінійної алгебри. '
                                     'NumPy підтримує інтеграцію з багатьма мовами. Наприклад, C та C++.')

@bot.message_handler(commands=['step17'])
def step_17(message):
    bot.send_message(message.chat.id, 'Під час цього уроку ми вивчимо успадкування, поліморфізм та інкапсуляцію')
    video =
open(r"C:\Users\aleks\PycharmProjects\pythonProject1\video18.mp4", 'rb')
    bot.send_video(message.chat.id, video, timeout=25)
    bot.send_message(message.chat.id, 'Для наступного кроку введи команду /step18, до речі, ще з '
                                     'найпопулярніших бібліотек для Python є Pandas. Pandas містить '
                                     'оптимізовані та гнучкі структури даних, які можна використовувати для '
                                     'управління даними тимчасових рядів і структурованими даними, такими як '
                                     'таблиці та

```

```

масиви. Наприклад, Pandas можна використовувати для
читання, запису, '
                                "об'єднання,
фільтрації та угруповання даних. Також дана бібліотека
широко "
                                'застосовується в
науці про дані, аналіз даних та завдання машинного
навчання.')

@bot.message_handler(commands=['step18'])
def step_18(message):
    bot.send_message(message.chat.id, 'Під час цього
уроку ми навчимося взаємодіяти з ітераторами у мові '
                                'Python')

    video =
open(r"C:\Users\aleks\PycharmProjects\pythonProject1\vi
deo19.mp4", 'rb')
    bot.send_video(message.chat.id, video, timeout=25)
    bot.send_message(message.chat.id, 'Для наступного
кроку введи команду /step19, до речі, ще з '
                                'найпопулярніших
бібліотек для Python є Requests. Бібліотека Requests'
                                'містить корисні
функції, необхідні для веб-розробки. Їх можна
використовувати'
                                ' для надсилання
HTTP-запитів, додавання заголовків, додавання
параметрів URL, '
                                "додавання даних
та виконання багатьох інших завдань, пов'язаних з "
                                'інтернет-
додатками')

@bot.message_handler(commands=['step19'])
def step_19(message):
    bot.send_message(message.chat.id, 'Ну що ж...')
    video =
open(r"C:\Users\aleks\PycharmProjects\pythonProject1\vi
deo20.mp4", 'rb')
    bot.send_video(message.chat.id, video, timeout=25)
    bot.send_message(message.chat.id, 'Я радий бути
частиною твого шляху, удачі в наступних починаннях 🙌')

```

```

@bot.message_handler(commands=['plan'])
def send_photo(message):
    file =
open(r'C:\Users\aleks\PycharmProjects\pythonProject1\pl
an.JPG', 'rb')
    bot.send_photo(message.chat.id, file, 'Ти впорашся
з цим 😊')

@bot.message_handler(commands=['main', 'hello'])
def main(message):
    bot.send_message(message.chat.id, f'<b>Привіт,
{message.from_user.first_name}. Я буду твоїм '
                                f'помічником у
цьому нелегкому шляху вивчення мови програмування '
                                f'Python. </b>',
parse_mode='html')

@bot.message_handler(commands=['id'])
def id_person(message):
    bot.send_message(message.chat.id, message)

@bot.message_handler(commands=['more_info'])
def add_info(message):
    markup = types.InlineKeyboardMarkup()
    button_01 =
types.InlineKeyboardButton('Високорівневі та
низькорівневі МП 😊',

url='https://foxminded.ua/nyzkorivneve-
prohramuvannia/')
    button_02 = types.InlineKeyboardButton('Стандарти
написання мови і такий важливий РЕР 8 🤖',

url='https://www.vitaliyodoba.com/2014/07/python-
style-guide/')
    button_03 =
types.InlineKeyboardButton('Компільовані та
інтерпретовані МП 🤖',

url='https://foxminded.ua/kompiliator-ta-

```

```

interpretator/')
    markup.row(button_01)
    markup.row(button_02)
    markup.row(button_03)
    bot.send_message(message.chat.id, 'Тут буде
знаходиться інформація з різних додаткових тем, які не
важливі для '
                                                    ' самого кодингу,
але важливі для влаштування на роботу або для розуміння
роботи '
                                                    ' програм та мов
програмування', reply_markup=markup)

@bot.message_handler(commands=['site'])
def website(message):
    markup = types.InlineKeyboardMarkup()
    button_1 = types.InlineKeyboardButton('Документація
Python', url='https://docs.python.org/3/')
    button_2 = types.InlineKeyboardButton('Stack
overflow', url='https://stackoverflow.com')
    markup.row(button_1)
    markup.row(button_2)
    bot.send_message(message.chat.id, '📖Документація
Python - сайт-біблія для розробників Python. Місце, де
'
                                                    ' вказані всі
правила написання коду та весь синтаксис, який є в мові
від творців '
                                                    ' самого
Python\n📖Stack overflow - Порятунк всієї '
                                                    ' ІТ, саме тут
знаходяться відповіді на всі твої запитання',
reply_markup=markup)

@bot.message_handler()
def dop(message):
    if message.text.lower() == 'привіт':
        bot.send_message(message.chat.id, f'<b>Привіт,
{message.from_user.first_name}</b>', parse_mode='html')
    elif message.text.lower() == 'id':
        bot.reply_to(message, f'ID:
{message.from_user.id}')

```



```
@bot.message_handler(content_types=['photo'])
def get_photo(message):
    bot.reply_to(message, 'Дякую, додам до своєї галереї')

@bot.message_handler(content_types=['video'])
def get_video(message):
    bot.reply_to(message, 'Дякую, але не треба надсилати мені відео')

@bot.message_handler(content_types=['audio'])
def get_audio(message):
    bot.reply_to(message, 'Вибач, але я не можу це прослухати')

bot.polling(none_stop=True)
```