# 

# **МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»**



##### **Факультет управления и информатики в технологических системах**

###### **Кафедра корпоративных информационных систем и программирования**

**Направление подготовки (специальность) 09.03.02 Информационные системы и технологии**

**Отчет**

**по учебной (ознакомительной) практике**

наименование (вид) практики

Выполнил студент гр. У-234

Матросов Артём Сергеевич

###### (Ф.И.О.)

###### \_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись)*

###### Проверили:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(должность руководителя от организации) (ф.и.о.)*

*\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(оценка) (подпись)*

*\_\_\_\_\_\_\_\_\_\_\_\_\_*

**МП** *(дата)*

\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(должность руководителя от университета) (ф.и.о)*

*\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(оценка) (подпись)*

*\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(дата)*

Воронеж – 2024

Содержание

|  |  |
| --- | --- |
| 1. Введение | 2 |
| 1. Индивидуальное задание | 3 |
| 1. Реализация | 4 |
| 1. Приложение с кодом | 7 |
| 1. Вывод | 12 |
| 1. Список источников | 14 |

Введение

1) Цель учебной практики направлена на закрепление и углубление теоретической подготовки специалиста, и приобретение им практических навыков и компетенций в сфере профессиональной деятельности. Научиться осуществлять поиск необходимой информации, приобрести умение анализировать чужой код, обучиться применять готовые решения для своих проектов.

2) Задачи практики:

- получение и закрепление теоретических знаний в сфере программирования;

- повышение своей квалификации и мастерства;

- освоение и понимание основных проблем дисциплин, которые определяют область профессиональной деятельности.

Индивидуальное задание

Индивидуальное задание № 11

Чат-бот рецептов для Telegram. Реализовать чат-бота который будет производить поиск по ключевым словам-ингредиентам и выдавать список самых распространенных актуальных рецептов. Реализовать функционал уже просмотренных рецептов и их исключение из последующего поиска.

Реализация

Среда разработки: PyCharm Community edition

Использовав имеющиеся знания в области Python, а также изучив разную литературу, я создал чат-бота, с которым можно взаимодействовать с помощью простых команд и в ответ получать ссылки, ведущие на веб-сайты, на которых описаны рецепты, содержащие ключевые слова, ранее переданные чат-боту.

1. Начало работы

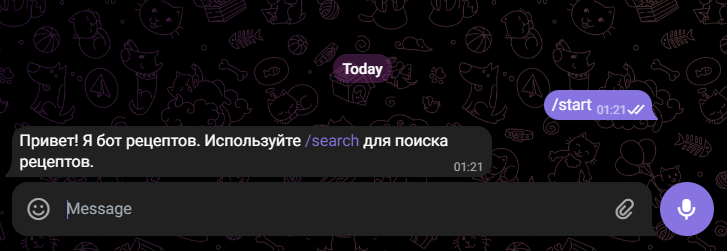


Рисунок 1 – Начало работы с чат-ботом, с помощью слова “/start”

2. Поиск рецептов по ключевым словам, с помощью команды “/search…”

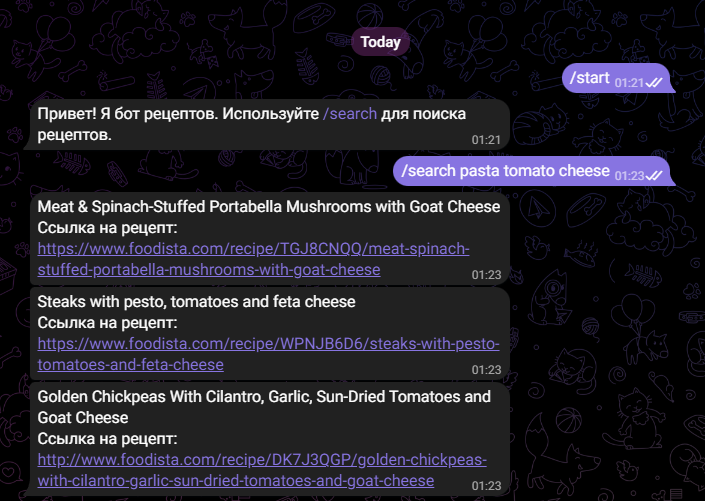


Рисунок 2 – Команда “/search…” и ссылки на рецепты, полученные в ответ.

3. Реализация функционала уже просмотренных рецептов и их исключение из последующего поиска.

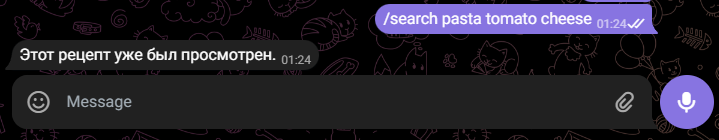
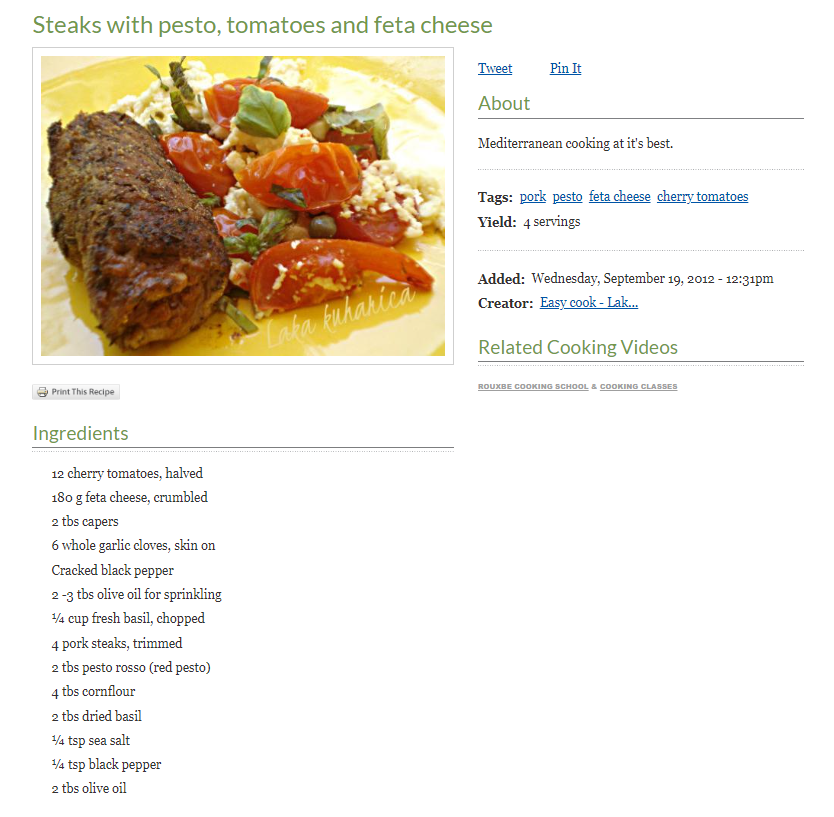


Рисунок 3 – При повторном запросе одних и тех же ингредиентов, бот ответит, что рецепт уже был просмотрен.

4. Рецепт, полученный в ответ от чат-бота.

 Рисунок 4 – Результат перехода по ссылке от чат-бота, содержащий рецепт с ключевыми словами “pasta tomato cheese”

Приложение с кодом

import logging  
import requests  
import executor  
from aiogram import Bot, Dispatcher, types  
  
TOKEN = '6345275285:AAGinXEN4HsYegsj-g6bEvlwUT0oDrERGqY'  
SPOONACULAR\_API\_KEY = '68065249552746bdbae5903805fc3c72'  
viewed\_recipes = {}  
  
logging.basicConfig(format='%(asctime)s - %(name)s - %(levelname)s - %(message)s', level=logging.INFO)  
logger = logging.getLogger(\_\_name\_\_)  
  
bot = Bot(token=TOKEN)  
dp = Dispatcher(bot=bot)   
  
@dp.message\_handler(commands=['start'])  
async def start(message: types.Message) -> None:  
 await message.answer('Привет! Я бот рецептов. Используйте /search для поиска рецептов.')  
  
@dp.message\_handler(commands=['search'])  
async def search(message: types.Message) -> None:  
 user\_id = message.from\_user.id  
 keywords = message.get\_args()  
  
 if not keywords:  
 await message.answer('Пожалуйста, укажите ключевые слова для поиска рецептов сразу после /search. Например /search pasta tomato cheese.')  
 return  
  
 query = ' '.join(keywords)  
  
 if query in viewed\_recipes.get(user\_id, []):  
 await message.answer('Этот рецепт уже был просмотрен.')  
 return  
  
 url = f'https://api.spoonacular.com/recipes/search?query={query}&apiKey={SPOONACULAR\_API\_KEY}&number=5'  
 response = requests.get(url)  
 data = response.json()  
  
 if response.status\_code == 200:  
 recipes = data.get('results', [])  
  
 if not recipes:  
 await message.answer('Я таких не знаю. Попробуйте другие ключевые слова.')  
 return

for recipe in recipes:  
 text = f"{recipe['title']}\nСсылка на рецепт: {recipe['sourceUrl']}"  
 await message.answer(text, parse\_mode=types.ParseMode.MARKDOWN)  
 else:  
 await message.answer('Ошибка при выполнении поиска. Пожалуйста, попробуйте позже.')

if \_\_name\_\_ == '\_\_main\_\_':  
 from aiogram import executor  
 executor.start\_polling(dp, skip\_updates=True)

Вывод

В ходе выполнения практической работы мы научились проводить анализ готовых решений, применять библиотеки, взаимодействовать с API-Telegram и Spoonacular.

Во время выполнения учебной практики я улучшил свои знания в области программирования и стал лучше знать язык Python.

Все поставленные задачи были выполнены. Цель была достигнута.

Список литературы

1. Официальная документация Telegram Bot API: https://core.telegram.org/bots/api

2. Python-Telegram-Bot документация:

https://python-telegram-bot.readthedocs.io/en/stable/

3. Udemy: Создание чат-ботов для Telegram на Python: https://www.udemy.com/course/chatbots-telegram-python/

4. Инструкция по получению Токена для Телеграмм бота

https://www.cossa.ru/instahero/321374/

5. YouTube: Создание Telegram ботов на Python: https://www.youtube.com/watch?v=PTz0Hj-vMVg

6. Telegram Bot API: Getting Started:

https://core.telegram.org/bots#3-how-do-i-create-a-bot