мІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

нАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «лЬВІВСЬКА ПОЛІТЕХНІКА»

ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНОРМАЦІЙНИХ ТЕХНОЛОГІЙ

|  |
| --- |
|  |



ЗВІТ

Про виконання лабораторної роботи № 7

З дисципліни

«Спеціалізовані мови програмування»

на тему «Робота з API та веб-сервісами»

Виконала:

студентка гр. ІТ-31

Проців Р.В.

Прийняв:

доц. каф. ІСМ

Щербак С.С.

Львів – 2023

**Мета роботи**: Створення консольного об’єктно - орієнтованого додатка з використанням API

**Хід роботи**:

**Завдання 1:** Вибір провайдера API

Виберіть надійний API, який надає через HTTP необхідні дані для віддаленого зберігання, вивантаження або реалізуйте свій. Для прикладу це може бути jsonplaceholder.org

**Завдання 2:** Інтеграція API

Виберіть бібліотеку для роботи з API та обробки HTTP запитів (для прикладу це може бути бібліотека Requests). Інтегруйте обраний API в ваш консольний додаток на Python. Ознайомтеся з документацією API та налаштуйте необхідний API-ключ чи облікові дані.

**Завдання 3:** Введення користувача

Розробіть користувальницький інтерфейс, який дозволяє користувачам візуалізувати всі доступні дані в табличному вигляді та у вигляді списку. Реалізуйте механізм для збору та перевірки введеного даних користувачем.

**Завдання 4:** Розбір введення користувача

Створіть розбірник для видобування та інтерпретації виразів користувача на основі регулярних виразів, наприклад, для візуалізації дат, телефонів, тощо. Переконайтеся, що розбірник обробляє різні формати введення та надає зворотний зв'язок про помилки.

**Завдання 5:** Відображення результатів

Реалізуйте логіку для візуалізації даних через API в консолі. Обробляйте відповіді API для отримання даних у вигляді таблиць, списків. Заголовки таблиць, списків мають виділяться кольором та шрифтом, які задається користувачем

**Завдання 6:** Збереження даних

Реалізуйте можливості збереження даних у чіткому та читабельному форматі JSON, CSV та TXT

**Завдання 7:** Обробка помилок

Розробіть надійний механізм обробки помилок для керування помилками API, некоректним введенням користувача та іншими можливими проблемами. Надавайте інформативні повідомлення про помилки.

**Завдання 8:** Ведення історії обчислень

Включіть функцію, яка реєструє запити користувача, включаючи введені запити та відповідні результати. Дозвольте користувачам переглядати та рецензувати історію своїх запитів.

**Завдання 9:** Юніт-тести

Напишіть юніт-тести для перевірки функціональності вашого додатку. Тестуйте різні операції, граничні випадки та сценарії помилок.

**Програмний код**:

import requests

import sys

import os

sys.path.append(os.getcwd())

from services import UserService as user\_service, DisplayInTableService

import utils.json\_handling as json\_processor

import utils.colors\_handling as color\_processor

import json

import utils.files\_handling as file\_processor

def main():

current\_folder = os.path.abspath(os.path.dirname(\_\_file\_\_))

json\_file\_path = os.path.join(current\_folder, "results", "output.json")

history: list = []

successful\_result: bool = False

jsons: list = []

while True:

print("Choose an option:")

print("1. Display data of a personal profile")

print("2. Display data of profiles posts")

print("3. Save data in JSON format")

print("4 - Show history")

print("0 - Exit")

option = input("Your choice: ")

match option:

case "1":

jsons = []

linkedin\_url = input("Enter LinkedIn URL: ")

try:

jsons = user\_service.get\_personal\_profile(linkedin\_url)

print("Choose an option:")

print("1. Display data in a flattened way")

print("2. Display data in JSON format")

print("3. Display data in a table")

while True:

option = input("Your choice: ")

match option:

case "1":

color\_processor.display\_colors()

color\_position = int(input("Enter a color position: "))

json\_processor.display\_flattened\_json(

jsons, color\_position

)

break

case "2":

print(json.dumps(jsons, indent=4))

break

case "3":

print(

DisplayInTableService.display\_personal\_profile(

json.dumps(jsons, indent=4)

)

)

break

case \_:

print("Invalid option. Enter again!")

history.append(

f"Data of a personal profile where URL is {linkedin\_url}:\n{json.dumps(jsons, indent=4)}"

)

successful\_result = True

except ValueError as e:

print(e)

successful\_result = False

case "2":

jsons = []

linkedin\_url = input("Enter LinkedIn URL: ")

try:

jsons = user\_service.get\_profiles\_posts(linkedin\_url)

print("Choose an option:")

print("1. Display data in a flattened way")

print("2. Display data in JSON format")

print("3. Display data in a table")

while True:

option = input("Your choice: ")

match option:

case "1":

color\_processor.display\_colors()

color\_position = int(input("Enter a color position: "))

json\_processor.display\_flattened\_json(

jsons, color\_position

)

break

case "2":

print(json.dumps(jsons, indent=4))

break

case "3":

print(

DisplayInTableService.display\_profiles\_posts(

json.dumps(jsons, indent=4)

)

)

break

case \_:

print("Invalid option. Enter again!")

history.append(

f"Data of a personal profile where URL is {linkedin\_url}:\n{json.dumps(jsons, indent=4)}"

)

successful\_result = True

except ValueError as e:

print(e)

successful\_result = False

case "3":

if successful\_result:

try:

file\_processor.write\_into\_json(json\_file\_path, jsons)

except Exception as e:

print(e)

else:

print("No data to save!")

case "4":

if len(history) == 0:

print("No history!")

else:

for counter, item in enumerate(history):

print(f"{counter + 1}: {item}")

case "0":

exit(0)

case \_:

print("Invalid option. Enter again!")

if \_\_name\_\_ == "\_\_main\_\_":

main()

**GitHub Repository:** <https://github.com/fxxwol/python>

**Висновок:**  в результаті даної лабораторної роботи було створено проект, який надав цінний досвід роботи з API, дизайну користувацького інтерфейсу, валідації введення, обробки помилок та тестування.