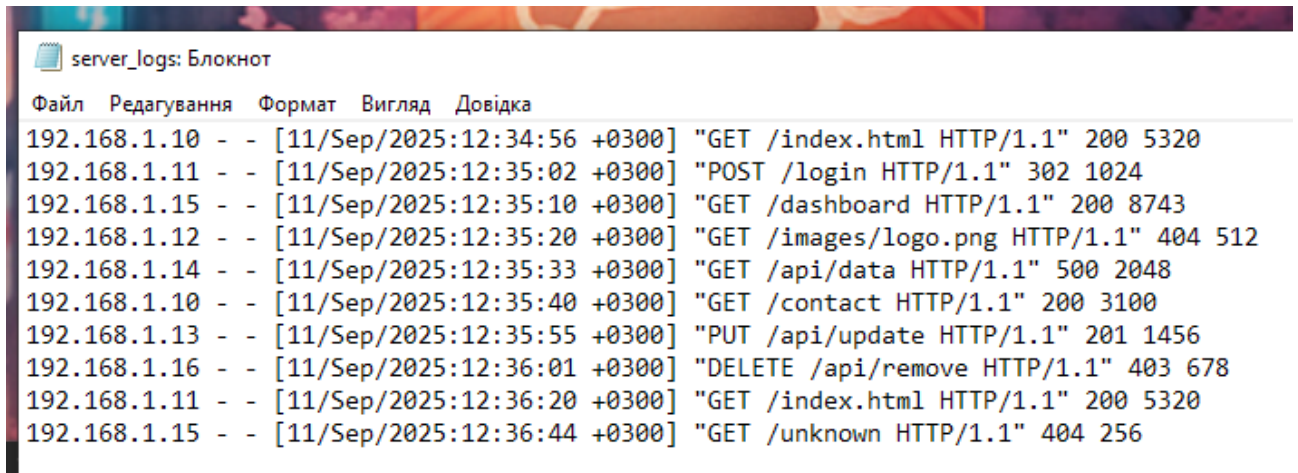


Лабораторна робота 2

Мета: ознайомитись з методами для роботи з файлами в середовищі Google Colab. Набуття базових навичок читання, створення, записування та обробки файлів різних форматів. Закріплення навичок виведення статистичних даних

Виконання роботи

1)Створити файл server_logs.txt та записати логи сервера в додатку 1;



```
server_logs: Блокнот
Файл  Редагування  Формат  Вигляд  Довідка
192.168.1.10 - - [11/Sep/2025:12:34:56 +0300] "GET /index.html HTTP/1.1" 200 5320
192.168.1.11 - - [11/Sep/2025:12:35:02 +0300] "POST /login HTTP/1.1" 302 1024
192.168.1.15 - - [11/Sep/2025:12:35:10 +0300] "GET /dashboard HTTP/1.1" 200 8743
192.168.1.12 - - [11/Sep/2025:12:35:20 +0300] "GET /images/logo.png HTTP/1.1" 404 512
192.168.1.14 - - [11/Sep/2025:12:35:33 +0300] "GET /api/data HTTP/1.1" 500 2048
192.168.1.10 - - [11/Sep/2025:12:35:40 +0300] "GET /contact HTTP/1.1" 200 3100
192.168.1.13 - - [11/Sep/2025:12:35:55 +0300] "PUT /api/update HTTP/1.1" 201 1456
192.168.1.16 - - [11/Sep/2025:12:36:01 +0300] "DELETE /api/remove HTTP/1.1" 403 678
192.168.1.11 - - [11/Sep/2025:12:36:20 +0300] "GET /index.html HTTP/1.1" 200 5320
192.168.1.15 - - [11/Sep/2025:12:36:44 +0300] "GET /unknown HTTP/1.1" 404 256
```

Рис.2.1 Створений файл server_logs.txt

2)На основі файлу server_logs.txt:

- Підрахувати кількість унікальних IP;
- Визначити найчастіші HTTP-методи;
- Знайти кількість помилок;
- Порахувати загальний обсяг переданих даних;
- Відфільтрувати всі запити “api”;
- Записати дані у новий файл results.txt та побудувати графіки найчастіших HTTP методів.

					МІВТ.420.002.002-ЛР2			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Величко В.С.			ПЗІВС		Літ.	Арк.
Перевір.		Рижук А.В						Акрушів
							1	2
Н. Контр.					Звіт з лабораторних робіт		ДУ “Житомирська політехніка”, ІВТ-6	
Затверд.								

```

import re
import pandas as pd
import matplotlib.pyplot as plt
with open("server_logs.txt", "r") as f:
    logs = f.readlines()
pattern = re.compile(
    r'(?P<ip>\d+\.\d+\.\d+\.\d+).+(?P<method>\w+) (?P<url>\S+) HTTP/\d\.\d" (?P<status>\d+) (?P<size>\d+)'
)

data = []
for line in logs:
    match = pattern.search(line)
    if match:
        data.append(match.groupdict())

df = pd.DataFrame(data)
df["status"] = df["status"].astype(int)
df["size"] = df["size"].astype(int)

unique_ips = df["ip"].nunique()

method_counts = df["method"].value_counts()

errors_count = df[(df["status"] >= 400) & (df["status"] < 600)].shape[0]

total_data = df["size"].sum()

api_requests = df[df["url"].str.contains("/api")]

with open("results.txt", "w") as f:
    f.write("Результати аналізу логів сервера:\n")
    f.write(f"Кількість унікальних IP: {unique_ips}\n")
    f.write("Найчастіші HTTP-методи:\n")
    f.write(method_counts.to_string())
    f.write(f"\nКількість помилок (4xx, 5xx): {errors_count}\n")
    f.write(f"Загальний обсяг переданих даних: {total_data} байт\n")
    f.write(f"Кількість запитів до /api: {len(api_requests)}\n")
    f.write("--- Запити до API ---\n")
    f.write(api_requests.to_string(index=False))

print(" Аналіз завершено. Результати записано у файл results.txt")

plt.figure(figsize=(6, 4))
method_counts.plot(kind="bar", color="skyblue", edgecolor="black")
plt.title("Найчастіші HTTP-методи")
plt.xlabel("HTTP-метод")
plt.ylabel("Кількість запитів")
plt.tight_layout()
plt.savefig("http_methods_chart.png")
plt.show()

```

Рис.2.2 Код для графіка

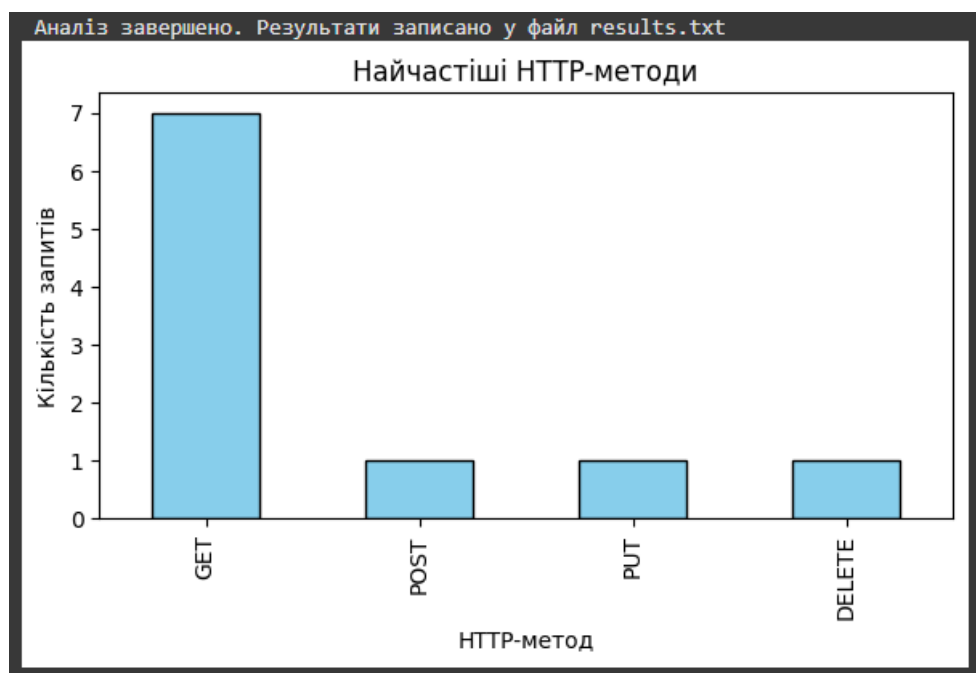


Рис 2.3 Результат

					МІВТ.420.002.002-ЛР2	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

3) Створити файл data.csv записати в нього дані з додатку 2;

```
import csv
import pandas as pd
import matplotlib.pyplot as plt

data = [
    ['Date', 'SensorID', 'Temperature', 'Humidity', 'Pressure', 'Status', 'Active'],
    ['2025-09-01', 'S001', 22.5, 45, 1012, 'OK', True],
    ['2025-09-01', 'S002', 23.0, 47, 1011, 'OK', True],
    ['2025-09-01', 'S003', 21.8, 44, 1013, 'FAULT', False],
    ['2025-09-02', 'S001', 22.2, 46, 1012, 'OK', True],
    ['2025-09-02', 'S002', 23.5, 48, 1010, 'OK', True],
    ['2025-09-02', 'S003', 24.0, 49, 1011, 'OK', True],
    ['2025-09-03', 'S001', 22.8, 45, 1012, 'OK', True],
    ['2025-09-03', 'S002', 23.2, 46, 1013, 'OK', True],
    ['2025-09-03', 'S003', 22.0, 44, 1012, 'FAULT', False],
    ['2025-09-04', 'S001', 23.5, 47, 1010, 'OK', True],
    ['2025-09-04', 'S002', 24.0, 49, 1011, 'OK', True],
    ['2025-09-04', 'S003', 22.5, 45, 1012, 'OK', True],
    ['2025-09-05', 'S001', 23.8, 48, 1013, 'OK', True],
    ['2025-09-05', 'S002', 22.9, 46, 1010, 'OK', True],
    ['2025-09-05', 'S003', 23.1, 47, 1011, 'FAULT', False],
    ['2025-09-06', 'S001', 22.7, 45, 1012, 'OK', True],
    ['2025-09-06', 'S002', 23.4, 48, 1013, 'OK', True],
    ['2025-09-06', 'S003', 24.0, 49, 1011, 'OK', True],
    ['2025-09-07', 'S001', 22.9, 46, 1010, 'OK', True],
    ['2025-09-07', 'S002', 23.6, 47, 1012, 'OK', True],
    ['2025-09-07', 'S003', 22.8, 45, 1013, 'FAULT', False],
]

with open('data.csv', 'w', newline='') as f:
    writer = csv.writer(f, quoting=csv.QUOTE_NONNUMERIC)
    for row in data:
        writer.writerow(row)

print(" Файл data.csv створено з лапками навколо всіх значень.\n")
```

Рис.2.4 Код для графіка

4) На основі файлу data.csv виконати:

- Читання даних з файлу;
- Підрахувати кількість рядків та вивести кількість активних сенсорів;
- Відфільтрувати данні в яких Status=FAULT, в новий файл записати активні датчики Active=True;
- Створити список даних для S001 датчика;
- Обчислити середню температуру для кожного сенсора окремо;
- Знайти мінімальне та максимальне значення вологості по всім даним;
- Побудувати графік вологості з вказанням найбільшого та найменшого значення.

					МІВТ.420.002.002-ЛР2	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

print(" Файл data.csv створено з лапками навколо всіх значень.\n")

df = pd.read_csv("data.csv")

df.columns = df.columns.str.strip()

print("=== Перші рядки таблиці ===")
print(df.head(), "\n")

num_rows = len(df)
active_sensors = df[df["Active"] == True]["SensorID"].nunique()

print(f"Кількість рядків: {num_rows}")
print(f"Кількість активних сенсорів: {active_sensors}\n")

fault_data = df[df["Status"] == "FAULT"]
active_data = df[df["Active"] == True]
active_data.to_csv("active_sensors.csv", index=False)
print("Файл active_sensors.csv створено з активними сенсорами.\n")

s001_data = df[df["SensorID"] == "S001"].to_dict(orient="records")
print("Дані для датчика S001:")
for row in s001_data:
    print(row)
print()

avg_temp = df.groupby("SensorID")["Temperature"].mean()
print("Середня температура для кожного сенсора:")
print(avg_temp, "\n")

min_humidity = df["Humidity"].min()
max_humidity = df["Humidity"].max()
print(f"Мінімальна вологість: {min_humidity}")
print(f"Максимальна вологість: {max_humidity}\n")

plt.figure(figsize=(8, 5))
plt.plot(df["Date"], df["Humidity"], marker="o", label="Вологість")
plt.title("Графік вологості за всіма даними")
plt.xlabel("Дата")
plt.ylabel("Вологість (%)")
plt.grid(True)

plt.scatter(df.loc[df["Humidity"].idxmin()], "Date", min_humidity, color="blue", s=100, label=f"Min = {min_humidity}")
plt.scatter(df.loc[df["Humidity"].idxmax()], "Date", max_humidity, color="red", s=100, label=f"Max = {max_humidity}")
plt.legend()
plt.tight_layout()
plt.savefig("humidity_chart.png")
plt.show()

```

Рис.2.5 Код для графіка

					МІВТ.420.002.002-ЛР2	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

```
Файл data.csv створено з лапками навколо всіх значень.

=== Перші рядки таблиці ===
Date SensorID Temperature Humidity Pressure Status Active
0 2025-09-01 S001 22.5 45 1012 OK True
1 2025-09-01 S002 23.0 47 1011 OK True
2 2025-09-01 S003 21.8 44 1013 FAULT False
3 2025-09-02 S001 22.2 46 1012 OK True
4 2025-09-02 S002 23.5 48 1010 OK True

Кількість рядків: 21
Кількість активних сенсорів: 3

Файл active_sensors.csv створено з активними сенсорами.

Дані для датчика S001:
{'Date': '2025-09-01', 'SensorID': 'S001', 'Temperature': 22.5, 'Humidity': 45, 'Pressure': 1012, 'Status': 'OK', 'Active': True}
{'Date': '2025-09-02', 'SensorID': 'S001', 'Temperature': 22.2, 'Humidity': 46, 'Pressure': 1012, 'Status': 'OK', 'Active': True}
{'Date': '2025-09-03', 'SensorID': 'S001', 'Temperature': 22.8, 'Humidity': 45, 'Pressure': 1012, 'Status': 'OK', 'Active': True}
{'Date': '2025-09-04', 'SensorID': 'S001', 'Temperature': 23.5, 'Humidity': 47, 'Pressure': 1010, 'Status': 'OK', 'Active': True}
{'Date': '2025-09-05', 'SensorID': 'S001', 'Temperature': 23.8, 'Humidity': 48, 'Pressure': 1013, 'Status': 'OK', 'Active': True}
{'Date': '2025-09-06', 'SensorID': 'S001', 'Temperature': 22.7, 'Humidity': 45, 'Pressure': 1012, 'Status': 'OK', 'Active': True}
{'Date': '2025-09-07', 'SensorID': 'S001', 'Temperature': 22.9, 'Humidity': 46, 'Pressure': 1010, 'Status': 'OK', 'Active': True}

Середня температура для кожного сенсора:
SensorID
S001 22.914286
S002 23.371429
S003 22.885714
Name: Temperature, dtype: float64

Мінімальна вологість: 44
Максимальна вологість: 49
```

Рис.2.6 Виконане завдання

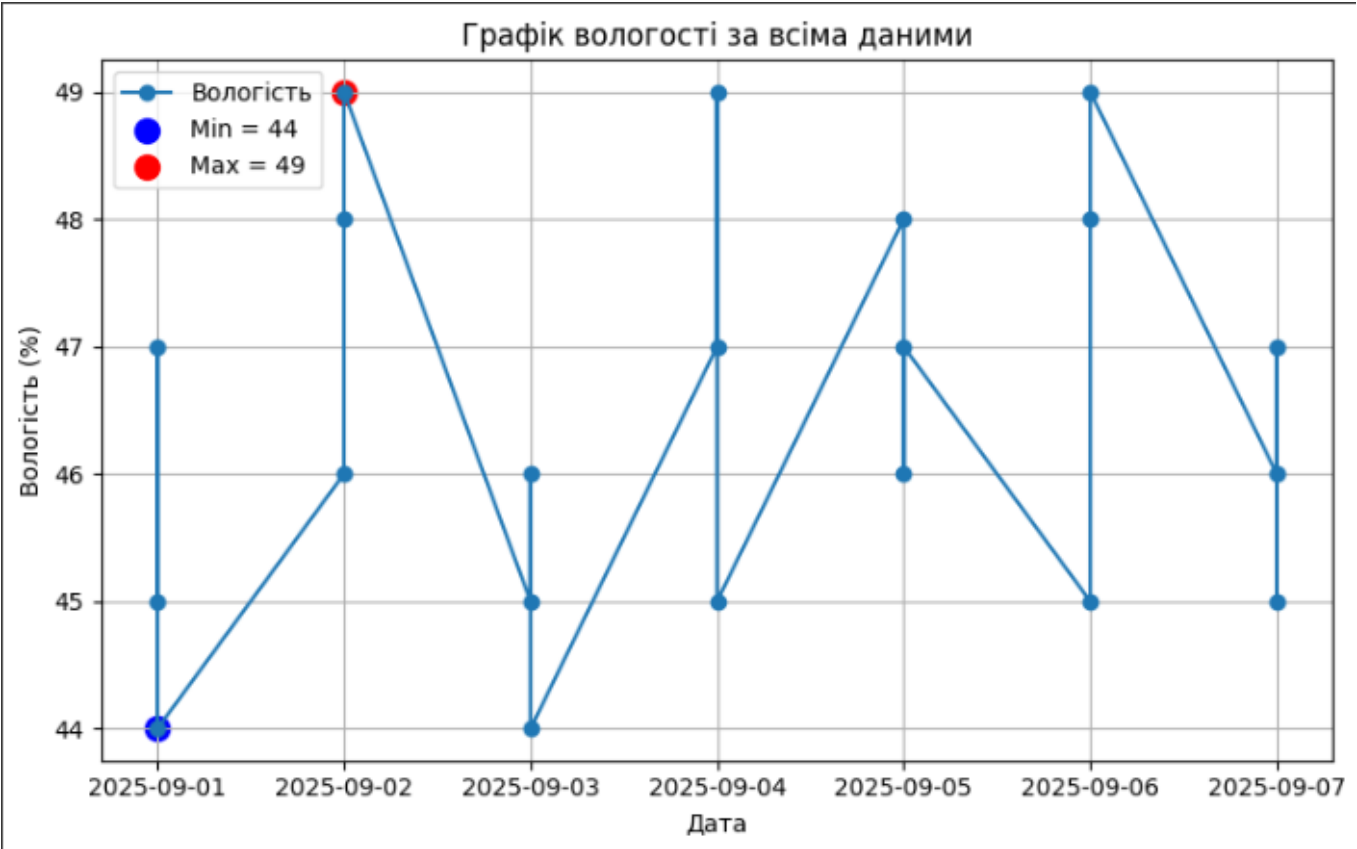


Рис 2.7 Графік вологості за всіма даними

Висновок: ознайомилась з методами для роботи з файлами в середовищі Google Colab. Набула базових навичок читання, створення, записування та обробки файлів різних форматів. Закріплення навичок виведення статистичних даних

Контрольні запитання

1. Які типи файлів ви можете відкривати у Python?

У Python можна відкривати будь-які типи файлів — текстові (.txt, .csv, .json, .xml), бінарні (.jpg, .png, .exe) та інші. Головне — правильно вибрати режим відкриття (текстовий або бінарний).

2. Що таке режим відкриття файлу (r, w, a, rb, wb)?

Режими відкриття файлу:

r – читання (файл повинен існувати),

w – запис (створює новий файл або перезаписує існуючий),

a – додавання (записує дані в кінець файлу),

rb – читання у бінарному режимі,

wb – запис у бінарному режимі.

3. Чим відрізняється текстовий файл від бінарного у Python?

Текстовий файл містить символи у вигляді тексту (UTF-8 тощо), а бінарний — байти. У текстовому режимі Python автоматично перетворює символи, а в бінарному — працює напряму з байтами.

4. Що відбувається, якщо забути закрити файл?

Якщо не закрити файл, дані можуть не зберегтися повністю або залишитись у буфері, а також це може займати ресурси системи.

Рекомендується використовувати конструкцію with open(...) as f: — тоді файл закриється автоматично.

5. Як прочитати весь текст із файлу example.txt за один раз?

```
with open("example.txt", "r", encoding="utf-8") as f: text = f.read() print(text)
```

6. Як прочитати файл рядок за рядком?

```
with open("example.txt", "r", encoding="utf-8") as f: for line in f: print(line.strip())
```

					МІВТ.420.002.002-ЛР2	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

7. Як додати новий рядок у кінець файлу?

```
with open("example.txt", "a", encoding="utf-8") as f: f.write("\nНовий  
рядок")
```

8. Як фільтрувати рядки CSV за значенням у конкретному стовпці?

```
import csv with open("data.csv", newline="", encoding="utf-8") as f: reader =  
csv.DictReader(f) for row in reader: if row["Статус"] == "Активний":  
print(row)
```

9. Як записати нові дані у CSV з додаванням нового стовпця?

```
import csv rows = [{"Ім'я": "Андрій", "Вік": 20, "Місто": "Київ"}, {"Ім'я":  
"Оля", "Вік": 22, "Місто": "Львів"}] with open("new_data.csv", "w",  
newline="", encoding="utf-8") as f: fieldnames = ["Ім'я", "Вік", "Місто"]  
writer = csv.DictWriter(f, fieldnames=fieldnames) writer.writeheader()  
writer.writerows(rows)
```

10. Як працювати з числами та текстом у CSV-файлі одночасно?

```
import csv with open("mixed.csv", newline="", encoding="utf-8") as f: reader  
= csv.reader(f) for row in reader: name = row[0] age = int(row[1])  
print(f' {name} має {age} років")
```

					МІВТ.420.002.002-ЛР2	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		