A black background with white text

AI-generated content may be incorrect.

|  |  |
| --- | --- |
| **Name:** | **Arisha Khan** |
| **Intern ID:** | **TN/IN02/PY/006** |
| **Internship Domain:** | **Python Development** |
| **Email ID:** | **Akhanjadoon2005@gmail.com** |
| **Instructor Name:** | **Mr. Hassan Ali** |

# TASK: WEEK FOUR

**TASK:**

**Integration Mini Project – Theory.**

Mini Project Using these libraries

Use request to get some data from website

Clean data using pandas

Make Graphical representation using matplotlib

Use gradio to show in your Web page

Build a small project by using **four libraries**:

1. **Requests** → fetch data from a website (example: GitHub API).
2. **Pandas** → clean and process the data (e.g., select important columns, handle missing values).
3. **Matplotlib** → visualize the processed data (e.g., bar chart of stars per repo).
4. **Gradio** → build a simple web app to show the results interactively.

* **Tasks – requests**

**Task : Fetch GitHub info of a user and print repos of your account**

import requests

username = "octocat"   # replace with your GitHub username

url = f"https://api.github.com/users/{username}/repos"

r = requests.get(url)

if r.ok:

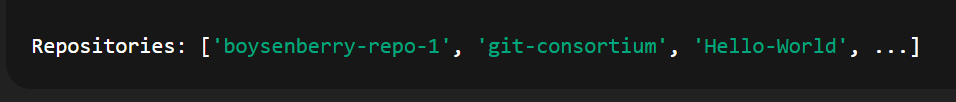
    repos = [repo['name'] for repo in r.json()]

    print("Repositories:", repos)

else:

    print("Error:", r.status\_code)

**OUTPUT:**



* **Tasks – requests:**

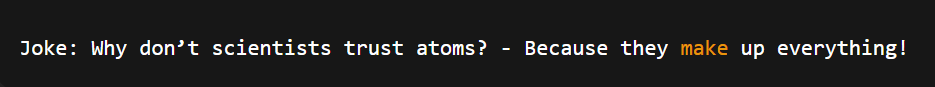
**Task: Use joke API to fetch and print a joke.**

joke\_url = "https://official-joke-api.appspot.com/random\_joke"

j = requests.get(joke\_url).json()

print(f"Joke: {j['setup']} - {j['punchline']}")

**OUTPUT:**



* **Tasks – pandas**

**Task: Clean students.csv and print average marks of each subject**

import pandas as pd

df = pd.read\_csv("students.csv")

# Fill missing values with column mean

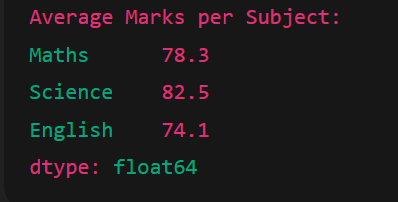
df.fillna(df.mean(numeric\_only=True), inplace=True)

# Print average marks per subject

print("Average Marks per Subject:")

print(df.mean(numeric\_only=True))

**OUTPUT:**



* **Tasks – matplotlib**

**Task: Bar chart of marks per student from students.csv.**

import matplotlib.pyplot as plt

plt.bar(df['Name'], df['Maths'])

plt.xlabel("Students")

plt.ylabel("Maths Marks")

plt.title("Maths Marks per Student")

plt.xticks(rotation=45)

plt.show()

* **Tasks – Gradio:**

**Task: Text analyzer app (character count).**

import gradio as gr

def text\_analyzer(text):

    return f"Character Count: {len(text)}"

demo = gr.Interface(fn=text\_analyzer, inputs="text", outputs="text")

demo.launch()