Python Assignment: Practical Automation and Data Handling

Objective

This assignment tests your Python skills in **web scraping, automation, and data handling**. You are required to complete two tasks: building a web scraping tool and automating a KPI dashboard.

Problem 1: Web Scraping Tool

Task

Create a Python script to scrape data from a publicly accessible website and store it in a structured format (CSV or JSON).

Requirements

1. **Website Selection**
    - Choose a website that lists structured data (e.g., products, news articles, or job listings). Ensure it does not require authentication or violate the website's terms of service.
2. **Data Points to Scrape**
    - Select at least five data points for each item (e.g., title, price, date, URL, description).
3. **Pagination Handling**
    - Ensure your script can scrape data across multiple pages if applicable.
4. **Error Handling**
    - Implement error handling for scenarios like connection issues, timeouts, or missing data.
5. **Data Storage**
    - Save the scraped data into a **CSV** or **JSON** file.

Deliverables

- Python script (`web_scraper.py`) with comments explaining the code.
- A sample output file (`scraped_data.csv` or `scraped_data.json`).

Problem 2: Automating a KPI Dashboard

Task

Develop a Python script that automates the creation of a **Key Performance Indicator (KPI) dashboard** from a dataset.

Requirements

1. **Dataset**
    - Use the provided **sales dataset** (`sales_data.csv`).
2. **KPIs to Calculate**
    - Calculate yearly KPIs, such as:
        - **Total sales/revenue per category.**
        - **Return on Marketing Spend (ROMS) per category.**
        - **Average Order Value (AOV) per category.**
3. **Data Visualization**
    - Use a library like `matplotlib` or `plotly` to create visualizations (e.g., bar charts, line graphs).
4. **Dashboard Output**
    - Generate a **PDF** or **HTML file** containing:
        - Visualized KPIs.
        - The raw data used for calculations.
5. **Automation**
    - Automate the script to run at regular intervals (e.g., daily, weekly) using `cron` (Linux/Mac) or Task Scheduler (Windows).

Deliverables

- Python script (`kpi_dashboard.py`) with comments explaining the code.
- Sample output file (`kpi_dashboard.pdf` or `kpi_dashboard.html`).
- Description of how to schedule the script using `cron` or Task Scheduler.

Evaluation Criteria

1. **Correctness and Efficiency**
    - Scripts should perform tasks as expected with efficient use of resources.
2. **Code Quality**
    - Clean, readable code with proper documentation and comments.
3. **Error Handling**
    - Adequate handling of exceptions (e.g., network errors, missing data).
4. **Automation Setup**
    - Proper explanation and setup of automation scripts.
5. **Data Handling and Visualization**
    - Effective data processing and visually clear representation of outputs.

Submission Guidelines

- Submit all scripts and output files in a **ZIP folder**.
- Include a **README** file with:
    - Instructions for running the scripts.
    - Description of external dependencies or libraries used.