Advance Python

Internal Assessment - Assignment 1

Quest 1-

Your task is to write a Python program that reads this CSV file, calculates the average score for each student, and then creates a new CSV file named "student\_average\_grades.csv”

* Steps to Solve

1. Read the data from "student\_grades.csv" using CSV file handling in Python.
2. For each student, calculate their average score across all subjects (Maths, Science, and English).
3. Create average functions to calculate the average for each student.
4. Store the student's name and their corresponding average score in a new dictionary.
5. Write the data from the dictionary into a new CSV file named "student\_average\_grades.csv" with two columns: "Name" and "Average."

Ques 2-

You are working as a data engineer for a large retail company. Your team is responsible for processing and analyzing sales data from multiple stores across the country. The sales data is stored in CSV files, and each file represents sales data for a specific month and year. Each CSV file has the following columns:

* Date (in the format "YYYY-MM-DD")
* Store ID (a unique alphanumeric code)
* Product ID (a unique alphanumeric code)
* Quantity sold (an integer representing the number of products sold on that date)

The "product\_names.csv" file has two columns: "Product ID" and "Product Name," and it contains the mapping for all products in the sales data.

Your task is to write a Python program that performs the following operations:

* Read the sales data from all the CSV files in a given directory and its subdirectories.
* Calculate the total sales (quantity sold) for each product across all stores and all months.
* Determine the top 5 best-selling products in terms of the total quantity sold.

Create a new CSV file named "sales\_summary.csv" and write the following information into it:

* Product ID
* Product Name
* Total Quantity Sold
* Average Quantity Sold per month (considering all months available in the data)

Ques 3-

You are working as a data scientist for a healthcare organization, and your team has been tasked with analyzing COVID-19 data from multiple countries. The data is stored in JSON files, with each file representing the daily COVID-19 statistics for a specific country. Each JSON file has the following structure:

{ "country": "Country Name",

"date": "YYYY-MM-DD",

"confirmed\_cases": { "total": 1000, "new": 50 },

"deaths": { "total": 20, "new": 2 },

"recovered": { "total": 800, "new": 30 }

}

Your task is to write a Python program that performs the following operations:

1. Read COVID-19 data from all JSON files in a given directory and its subdirectories.
2. Calculate and display the following statistics for each country:
   1. Total confirmed cases.
   2. Total deaths.
   3. Total recovered cases.
   4. Total active cases (total confirmed cases minus total deaths and total recovered).
3. Determine the top 5 countries with the highest number of confirmed cases and the lowest number of confirmed cases.
4. Generate a summary report in JSON format that includes the statistics for all countries and save it to a file named "covid19\_summary.json".

Ques 4-

You are working for a company that sells products online. Your task is to develop a Python program that reads order data from a CSV file, generates individual PDF invoices for each order, and then merges all the PDF invoices into a single PDF file.

1. Load Order Data: The program should read order data from a CSV file named "orders.csv." Each row in the CSV file represents an order with the following information:
   1. Order ID (a unique alphanumeric code)
   2. Customer Name
   3. Product Name
   4. Quantity
   5. Unit Price
2. Calculate Total Amount: For each order, calculate the total amount by multiplying the quantity with the unit price.

Generate PDF Invoices: Create individual PDF invoices for each order. Each invoice should contain the following details:

* 1. Invoice Number (same as the Order ID)
  2. Date of Purchase (current date)
  3. Customer Name
  4. Product Name
  5. Quantity
  6. Unit Price
  7. Total Amount

Ques 5-

* You are working on a project to build a custom text processing tool that reads input from various sources, processes the text data, and stores the results in an output file. As part of this project, you need to implement a robust exception handling mechanism to handle potential errors that may arise during the text processing.
* The tool needs to perform the following steps:

1. Read the input data from a file specified by the user.
2. Process the text data by performing various operations, such as counting words, calculating character frequencies, and generating word clouds.
3. Store the processed results in an output file.

Your task is to design a Python program that incorporates appropriate exception handling to handle the following situations:

1. File Not Found Error: If the user provides an invalid file path or the input file is not found, your program should raise a custom exception FileNotFoundError with a suitable error message.
2. Invalid Input Data: During text processing, if any unexpected input data is encountered (e.g., non-string values or missing data), your program should raise a custom exception InvalidInputDataError with relevant details.
3. Disk Space Full: If the output file cannot be written due to insufficient disk space, your program should raise a custom exception DiskSpaceFullError

Ques 6-

You are developing a command-line task management system for a small team of users.

User Management:

* Implement a user registration system where users can sign up and log in. Store user data in a file, including usernames and hashed passwords.

Ques 7-

**Task: Household Expenses Tracker**

You have been tasked with creating a Python program to help manage household expenses. The program should allow family members to input their daily expenses, store them in a CSV file, and provide functionalities for analysis and reporting.

1. **Expense Logging:** Create a Python program that allows users to input their daily expenses. The program should prompt the user for their name, date of the expense, description, and amount spent. The data should be stored in a CSV file named **expenses.csv** with columns 'Name', 'Date', 'Description', and 'Amount'.
2. **Expense Analysis:** Develop a function that reads the **expenses.csv** file and calculates the total expenses for each family member. Display the total expenses for each member along with the average daily expense for the household.
3. **Expense Trends:** Implement a feature that generates a line chart using a plotting library (e.g., Matplotlib) to visualize the expense trends over the last month. The x-axis should represent the dates, and the y-axis should show the cumulative expenses for each day.
4. **Expense Categorization:** Enhance the program to allow users to categorize their expenses. Prompt the user to assign a category (e.g., groceries, utilities, entertainment) to each expense entry. Update the CSV file to include a 'Category' column.
5. **Expense Reporting:** Create a monthly expense report by reading the data from **expenses.csv** and generating a report that includes the following:
   * Total expenses for each family member for the month.
   * A breakdown of expenses by category.
   * A comparison of monthly expenses over different months using bar charts.
6. **Expense Budgeting:** Add an option for users to set a monthly budget for each category. After entering expenses, the program should calculate the remaining budget for each category and provide a warning if the budget is exceeded.
7. **Data Backup and Restore:** Implement a backup and restore feature that allows users to save a copy of the **expenses.csv** file to a backup location and restore it if needed. Handle cases where the file might be missing or corrupted.