Principles of Programming

Coursework Specification | Villa College

Module Details

Module Code: UFCFHS-30-1	UFCFHS-30-1
Module Title	Principles of Programming
Module Leaders	Ahsan Ali, Hassan Maheedh Mohamed
Academic Year	2023
Component/Element	B
Total Assessments	One comprehensive assignment
Weighting	This coursework represents 50% of your total module grade.
Collaboration	Individual submission

Submission Details

Deadline	7th September 2023, 23:00
Platform	Moodle
Submission Format	Single ZIP file (FIRSTNAME_LASTNAME_ID.py) The file should contain: .py file .pdf file

Coursework Overview:

Monthly Expense Predictor and Savings Planner CLI Application

Objective

Design a command-line application that allows users to input monthly expenses in various categories, predict next month's expenses, and plan savings accordingly.

Tasks

1. Data Entry Interface:

- **Description:** Develop a command-line interface for users to enter the description, category, amount, and date of each expense.
- **Implementation Details:**
    - Allow the user to enter the description of an expense.
    - Provide a list of predefined categories for the user to choose from (e.g., Coffees, Movies, Dinners, etc.).
    - Accept the expense amount.
    - Allow the user to input the date of the expense.
    - Validate all input data, ensuring there are no empty fields and data formats are consistent.

2. Data Storage:

- **Description:** Store the user's entered data in a .txt or .csv file.
- **Implementation Details:**
    - Each time the user enters an expense, append it to the file.
    - Efficiently read and write data without errors.

3. Prediction Interface:

- **Description:** Predict the user's expenses for the upcoming month based on the data of the past three months.
- **Implementation Details:**
    - For the prediction formula, consider using a simple moving average or linear regression as a hint.

- Display the predicted amount for each category for the upcoming month.

4. Savings Planner Interface:

- **Description**: Prompt the user for a savings goal for the next month.
- **Implementation Details:**
  - Recommend adjustments to their spending in various categories to help achieve the savings goal.

BONUS: Visualization of Trends:

- **Description:** Represent the user's spending trend over the past three months for a chosen category.
- **Implementation Details:**
  - Using ASCII characters or simple text-based graphics, show a basic trend for a category over three months. For example, for every 100 MVR spent, represent it with a '#'. Or you can use any suitable Python library for this, such as “matplotlib”.

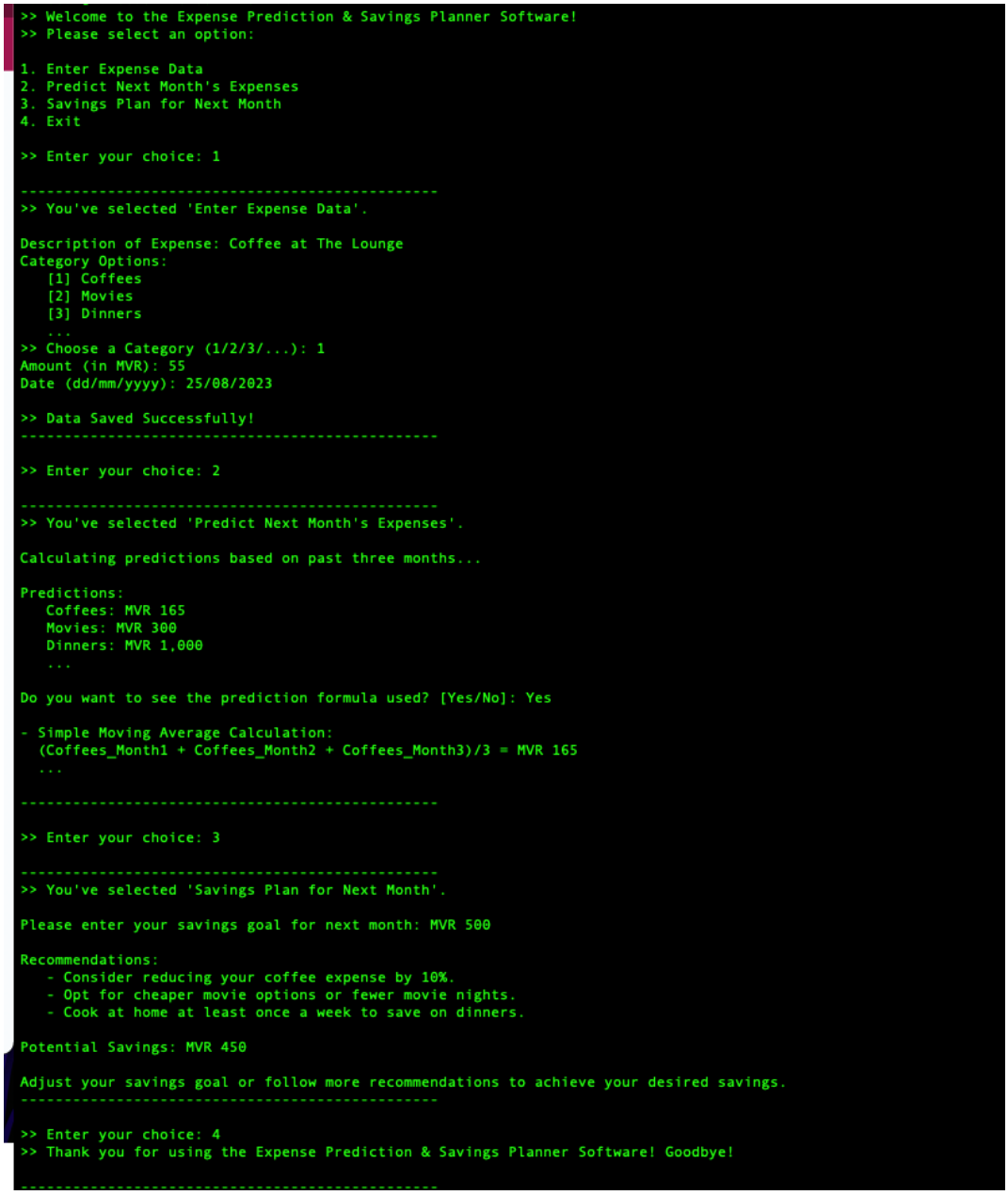
Example Use Case and Output

Use Case:
Aisha is a university student. She has been diligently keeping track of her expenses for the past six months. She's decided to use the provided software to predict her expenses for the next month and develop a savings plan.

Input:

```
>> Welcome to the Expense Prediction Software!
>> Please input your name: Aisha

>> Hello, Aisha! Let's predict your expenses for next month.
>> Please enter your expenses for the past six months for each category.

Category: Coffees
Please input expenses for the past 6 months in MVR: 100, 125, 122, 124, 123, 126

Category: Dates
Please input expenses for the past 6 months in MVR: 500, 600, 480, 530, 560, 520

Category: Dinner out
Please input expenses for the past 6 months in MVR: 1000, 980, 1020, 1030, 1050, 1010

Category: Entertainment
Please input expenses for the past 6 months in MVR: 300, 280, 290, 320, 310, 300

Category: Uni Fee
Please input expenses for the past 6 months in MVR: 5000, 5050, 5020, 5000, 5030, 5040

Category: Phone bill
Please input expenses for the past 6 months in MVR: 400, 420, 400, 410, 400, 420
```

Output:


>> Welcome to the Expense Prediction & Savings Planner Software!
>> Please select an option:
1. Enter Expense Data
2. Predict Next Month's Expenses
3. Savings Plan for Next Month
4. Exit
>> Enter your choice: 1
>> You've selected 'Enter Expense Data'.
Description of Expense: Coffee at The Lounge
Category Options:
[1] Coffees
[2] Movies
[3] Dinners
...
>> Choose a Category (1/2/3/...): 1
Amount (in MVR): 55
Date (dd/mm/yyyy): 25/08/2023
>> Data Saved Successfully!
>> Enter your choice: 2
>> You've selected 'Predict Next Month's Expenses'.
Calculating predictions based on past three months...
Predictions:
Coffees: MVR 165
Movies: MVR 300
Dinners: MVR 1,000
...
Do you want to see the prediction formula used? [Yes/No]: Yes
- Simple Moving Average Calculation:
(Coffees_Month1 + Coffees_Month2 + Coffees_Month3)/3 = MVR 165
...
>> Enter your choice: 3
>> You've selected 'Savings Plan for Next Month'.
Please enter your savings goal for next month: MVR 500
Recommendations:
- Consider reducing your coffee expense by 10%.
- Opt for cheaper movie options or fewer movie nights.
- Cook at home at least once a week to save on dinners.
Potential Savings: MVR 450
Adjust your savings goal or follow more recommendations to achieve your desired savings.
>> Enter your choice: 4
>> Thank you for using the Expense Prediction & Savings Planner Software! Goodbye!

Assignment Marking Criteria (Total: 100 Marks)

1. Code Functionality (50 Marks)

- Data Input and Validation (10 Marks)
    - Successful data input from the user for each category: 5 Marks
    - Proper validation (ensuring values are numerical, non-negative, etc.): 5 Marks
- Expense Prediction (20 Marks)
    - Implementation of the prediction algorithm (even if it's simple, like an average): 10 Marks
    - Accuracy in predicting next month's expenses based on user data: 10 Marks
- Saving Plan Recommendations (10 Marks)
    - Clear recommendations based on savings goals and predictions: 5 Marks
    - Sensible adjustments and savings suggestions: 5 Marks
- User Interface & Error Handling (10 Marks)
    - User-friendly command-line interface: 5 Marks
    - Proper error handling and feedback to the user (e.g., invalid inputs): 5 Marks

2. Documentation (35 Marks)

- Introduction (5 Marks)
    - Brief explanation of the software purpose: 3 Marks
    - Clear user instructions: 2 Marks
- Code Explanation (15 Marks)
    - Comments and documentation in code: 5 Marks
    - Separate written explanation of how the main functions/methods work: 10 Marks
- Use Cases & Testing (10 Marks)
    - Explanation of at least two use cases with expected inputs and outputs: 5 Marks
    - Demonstration of testing strategies and results: 5 Marks
- Conclusions & Recommendations (5 Marks)
    - Reflection on the software’s performance and reliability: 3 Marks
    - Suggestions for future improvements: 2 Marks

3. Online Demo Presentation (10 Marks)

- Understanding & Clarity (5 Marks)
    - Clear demonstration of how the software works: 3 Marks
    - Ability to explain code and design decisions: 2 Marks
- Q&A (5 Marks)
    - Responsiveness to questions and feedback during the presentation: 5 Marks

4. Bonus: Visualization Implementation (5 Marks)

- Visualization (5 Marks)
    - Successful implementation of a visualization feature showing the comparison between current expenses and predictions: 5 Marks

---