Project Title: Personal Expense Tracker

Project Overview:

In this project, you will build a command-line-based **Personal Expense Tracker** to help users record and manage their daily expenses. The program will allow users to:

1. **Add, View, and Categorize** Expenses.
2. **Generate Reports** on expenses (total spent, category-wise spending, etc.).
3. **Store Data** in a file (CSV or text) for future use.
4. **Visualize Expenses** using simple charts.
5. Apply error handling to make sure inputs and data are valid.

Key Components:

1. **Basic Operations**:

   - Add Expenses: Allow the user to input the expense amount, category (e.g., Food, Transport, Entertainment), and date.
   - View Expenses: Display a list of all expenses added by the user.
   - Categorize Expenses: Organize expenses by categories (Food, Transport, etc.) to make it easy to track spending habits.

2. **Data Structures**:

   - Use Lists and Dictionaries to store and organize the expenses.
   - For example, store each expense as a dictionary with keys like `amount`, `category`, and `date`.

3. **File Handling**:
   - Save the expenses to a CSV file using Python's `csv` module.
   - Load the expenses from the file when the program starts so users can keep a history of their spending.

4. **Functions**:
   - Create functions for each operation: `add_expense()`, `view_expenses()`, `generate_report()`, and so on.
   - Modularize the code for easier readability and maintainability.

5. **Reports and Calculations**:
   - Implement functions to **generate reports** like total monthly spending, spending by category, and highest expense.

- Use basic arithmetic and logical operators to calculate totals, averages, and comparisons.

6. **Data Visualization**:
   - Use Matplotlib to generate simple bar charts or pie charts to show the distribution of spending across categories.
   - Show total monthly or weekly spending in a graphical format.

7. **Control Flow**:
   - Use if-else statements to navigate the menu (add/view expenses, generate reports).
   - Implement loops (while loops and for loops) to keep the program running until the user decides to exit.

8. **Error and Exception Handling**:
   - Handle common errors like invalid input (e.g., entering text instead of numbers for amounts).
   - Use `try-except` blocks to catch and manage such errors.

9. **Mini Project (Before Full)**:
   - You can start with a mini project where you implement just the basic functionality (add expenses and view them) and then gradually build upon that.

Sample Menu:

Welcome to Personal Expense Tracker!

1. Add an Expense
2. View All Expenses
3. Generate Report
4. Save and Exit

Enter your choice:

Tools and Libraries:

- **Standard Python Libraries**: `csv`, `datetime`
- **Visualization**: `matplotlib` for basic charts
- **Data Handling**: Use lists, dictionaries, and file handling techniques

Sample Project Breakdown:
- **Phase 1**: Implement the basic menu and functions to add/view expenses.
- **Phase 2**: Add file handling to save expenses to a CSV file and load them when the program starts.
- **Phase 3**: Implement simple reports and calculations (total spent, spending by category).

- **Phase 4**: Add data visualization using `matplotlib` (optional for a simple pie chart or bar graph).

<u>**Example Usage:**</u>

- Add expense: Enter an amount (e.g., $50), category (e.g., Food), and date.
- View expenses: Displays a list like:

```
Date            Category    Amount
2024-10-13        Food        $50
2024-10-12      Transport     $20
```

- Generate report: Shows total spending and spending by category.

This project is straightforward but still covers many essential Python concepts such as control flow, data types, file handling, functions, and visualization. It is suitable for beginner-level Python learners but provides a practical application.

For any help feel free to take help from google or youtube also mail us to support@ediglobe.com. Make a pdf with all codes and necessary screenshot of output.