Expense Tracker Application

Project Description:

The goal of this project is to create a Python application that helps users track and analyze their daily expenses. The application will allow users to add, edit, and delete expense records, categorize expenses, and generate summary reports. Additionally, the project integrates NumPy, Pandas, and Matplotlib for efficient data processing and visualization.

Features:

1. **Add Expense**
   Users can input details for an expense, including:
   - Amount
   - Date
   - Category (e.g., Food, Transport, Entertainment)
   - Description
   - Expense Management,
   - Payment Method
   - Tags, etc.

2. **Edit/Delete Expense**
   - Users can select an existing expense and either update its details or delete it.

3. **Budget Tracking & Alerts**

   - Users can set monthly or category-specific budgets.
   - Highlight overspending in categories or overall monthly expenses.
   - Notifications or summary messages for exceeding budgets.

4. **View Expenses: Dynamic Filtering & Sorting**

   - Display a list of all expenses.
   - Filter by date range, category, payment method, amount range, or tags.
   - Sort expenses by any field (date, amount, category, etc.).

5. **Generate Advanced Summary Reports**

   - Total, average, median, min, max, and standard deviation of expenses.
   - Category-wise summaries with totals, averages, and percentages.
   - Monthly, quarterly, and yearly trends for all expenses or per category.
   - Top N expenses to quickly identify large spending items.
   - Custom date range reports for flexible analysis.
   - Highest and lowest expenses.

6. **Visual Analytics**

   - Charts for category totals.
   - Charts for percentage contribution per category.
   - Charts for monthly/yearly trends.

7. **Data Persistence**

   - Save all expense records in CSV files for long-term storage.
   - Load data into Pandas DataFrames at startup for analysis.
   - Save the changes in the data made before closing the program.

8. **Export Reports**

   Export summaries and charts as PDF, Excel, or image files for sharing or record-keeping.

Technical Requirements:

1. **Programming Concepts:**

   - Use OOP principles to design classes (such as: Expense, ExpenseManager, ReportGenerator, Visualizer).
   - Modular code with clear separation of concerns.

2. **Data Structures:**

   - Use lists, dictionaries, and Pandas DataFrames for in-memory storage and manipulation.

3. **Libraries & Tools**

   - NumPy: Efficient numerical calculations (totals, averages, min/max, statistics).
   - Pandas: Data storage, filtering, grouping, aggregation, and file I/O.
   - Matplotlib / Seaborn: Advanced visualization of spending trends and patterns.

4. **Error Handling**

   - Validate inputs (amount, date format, category).
   - Handle missing or malformed files gracefully.

Evaluation Criteria:

1. **Functionality:** Full-featured expense tracking with advanced reports and visualizations.

2. **Data Analysis:** Correct statistical calculations, trends, and summaries.

3. **Visualization:** Clear, informative, and interactive charts.

4. **Code Quality:** Proper OOP use, modularity, readability, and maintainability.

5. **Persistence & Reporting:** Reliable saving/loading and exportable reports.

6. **Error Handling:** Handle scenarios like missing input fields or invalid data gracefully.

7. **Data Persistence:** Ensure all data is saved and loaded correctly from the file.

8. **Presentation:** The answer to the questions asked during the presentation. Ability to explain analytics and visualization choices.