Personal Expense Tracker - Revised Project Specification

Project Overview:

The **Personal Expense Tracker** is a web-based application designed to help users track their daily expenses in a simple and user-friendly way. The application allows users to input their daily expenditures, categorize them, and view their spending in a summarized format. Future iterations will include budget management and spending visualizations to give users better insights into their financial habits.

Key Features:

1. **Expense Entry Form**: Users can enter expenses, specifying the date, category (e.g., food, transport, entertainment), and the amount.
2. **Dynamic Expense Summary Table**: A table that dynamically updates with each new entry, showing all expenses in a clean, tabular format.
3. **UI Layout Adjustments**: A user-friendly interface with the expense form on the left and the expense summary table on the right.
4. **Budget Setting Feature (Revised)**: A placeholder form exists for users to set budgets by category, but the feature is not yet functional. It will be implemented in future iterations to allow users to track whether they are exceeding their budget limits.
5. **Data Persistence (Revised)**: Currently, expenses are stored in memory and are lost upon restarting the app. In the next iteration, expenses and budgets will be stored using a database or file-based system.
6. **Spending Visualizations (Proposed)**: Future versions will include graphical representations of spending patterns using libraries like Matplotlib or Plotly to give users insights into their spending trends.
7. **User Authentication (Optional)**: If time permits, this feature will allow multiple users to track their expenses and budgets individually by logging in.

Revised Feature Breakdown and Feasibility:

1. Core Features (Already Implemented)

- **Expense Entry Form**:
    - Status: Fully implemented. Users can enter their expenses with the date, category, and amount.
    - Feasibility: Complete and functional.
- **Dynamic Expense Table**:
    - Status: Fully implemented. The table updates automatically with each new expense entry.

    - Feasibility: Complete and functional.

- **UI Layout Adjustments**:
    - Status: Implemented. The form is positioned on the left and the table on the right to improve usability.
    - Feasibility: Complete and functional.

2. Budget Setting (Revised)

- **Previous Plan**: Initially planned to have the full budget setting feature working in Version 1.
- **Current State**: A placeholder exists, but no functionality has been implemented yet.
- **Revision**: The budget setting feature will now be fully implemented in the next phase, including the ability for users to set limits for each category and track their spending against these limits.
- **Feasibility**: This feature is planned for the next version and is feasible within the remaining time. However, more complex features like alerts when the budget is exceeded may need to be deprioritized until later.

3. Data Persistence (Revised)

- **Previous Plan**: Initially, the project did not account for data persistence, and expenses were stored in memory only.
- **Current State**: Expenses are lost when the app is restarted.
- **Revision**: In the next phase, data persistence will be added using either a lightweight database solution like SQLite or a file-based storage system (e.g., JSON or CSV). This will ensure that expenses and budget data are saved across sessions.
- **Feasibility**: Feasible within the remaining time, as implementing SQLite is relatively straightforward and can be done alongside the budget feature.

4. Spending Visualizations (Proposed)

- **Previous Plan**: Visualizations were not part of the original spec.
- **Proposed Addition**: To give users insights into their spending, I propose adding spending visualizations, such as pie charts or bar graphs, that will display category-wise breakdowns of expenses.
- **Feasibility**: This feature is proposed as a "nice to have" and will be implemented after the core functionality (expense tracking, budget setting, data persistence) is complete. It will use libraries like Matplotlib or Plotly, which integrate well with Flask.

5. User Authentication (Proposed, Optional)

- **Previous Plan**: Authentication was not included in the initial spec.
- **Proposed Addition**: If time permits, I propose adding user authentication so that multiple users can track their expenses separately. This feature is not critical for Version 2 but could be a useful addition in the future.
- **Feasibility**: This is a stretch goal and will only be implemented if there is time after completing the core features.

Revised Timeline and Milestones:

Milestone 1: Budget Setting Implementation

- **Description**: Add functionality to allow users to set budget limits for each category and track their expenses against these limits.
- **Planned Completion**: Next two weeks.
- **Dependencies**: Requires budget storage to be integrated with expense tracking.

Milestone 2: Data Persistence

- **Description**: Implement data persistence using SQLite or file-based storage to store expenses and budget data across sessions.
- **Planned Completion**: After the budget setting functionality.
- **Dependencies**: Data storage should be set up before adding any further features like visualizations.

Milestone 3: Spending Visualizations

- **Description**: Add graphical representations (e.g., pie charts, bar graphs) to give users insights into their spending patterns.
- **Planned Completion**: Once data persistence is in place.
- **Dependencies**: Requires data storage to pull the necessary information for visualizations.

Milestone 4: User Authentication

- **Description**: Implement a basic authentication system to allow multiple users to track their expenses individually.
- **Planned Completion**: If time allows, after all other core functionality is complete.
- **Dependence**: Requires data storage to keep track of individual user data.

Summary of Revisions:

1. **Budget Setting Feature**: Revised to be implemented in Version 2, with core functionality to track whether users exceed their budgets.
2. **Data Persistence**: Revised to be added in the next version, using a database or file storage to ensure that expenses and budget data are saved across sessions.
3. **Proposed Features**: Added spending visualizations and user authentication as "nice to have" features, to be implemented after core functionality, if time permits.