Expense Tracker

-Showrish Reddy Sampath

Code Structure

- **HTML (`index.html`):**
  - Provides the structure of the Expense Tracker application.
  - Contains elements like the title, forms for adding transactions, summary sections, a filter dropdown, and the canvas for the chart.
- **CSS (`expense.css`):**
  - Styles the entire web application, making it visually appealing.
  - Includes styling for the body, container, summary section, forms, buttons, and the chart.
- **JavaScript (`expense.js`):**
  - Handles the core functionality of the application.
  - **Key Variables:**
    - `transactions`: An array to store transaction objects.
    - `transactionsList`, `totalIncomeEl`, `totalExpensesEl`, `netIncomeEl`, `filterCategoryEl`, `expenseChartEl`: DOM elements used to update the UI.
    - `chart`: Holds the Chart.js instance for the pie chart.
  - **Main Functions:**
    - `updateUI`: Updates the transaction list, summary, and chart.
    - `addTransaction`: Adds a new transaction to the list.
    - `removeTransaction`: Removes a transaction by its ID.
    - `editTransaction`: Edits an existing transaction.
    - `filterTransactions`: Filters transactions based on the selected category.
    - `populateCategoryFilter`: Updates the category filter dropdown.
    - `updateChart`: Updates the pie chart with current expense data.
    - `exportData`: Exports the transactions as a JSON file.
    - `importData`: Imports transactions from a JSON file.
  - **Event Listeners:**
    - Listeners for adding transactions, filtering by category, exporting data, and importing data.

How It Works

- **Adding Transactions:** When a user adds a transaction, it is stored in the `transactions` array. The UI is updated to reflect the new transaction, and the data is saved to `localStorage`.
- **Filtering:** Users can filter transactions by category using a dropdown menu. The `filterTransactions` function returns transactions that match the selected category.
- **Chart Visualization:** The `updateChart` function uses Chart.js to create a pie chart that displays the distribution of expenses by category.
- **Data Persistence:** The application uses `localStorage` to save transactions, ensuring data is retained even if the page is refreshed. Users can also export and import transactions as JSON files.

This documentation provides a high-level overview of how the Expense Tracker application is structured and functions, offering guidance for anyone looking to understand or modify the code.

Thank You