Job Simulation: Expense Tracker with Chart Visualization

Project Title: Implementation of a Visual Expense Tracker

Role: Front-End Dashboard Intern – Charting & State Management

Technology Stack:

- React
- Material UI (MUI)
- Recharts
- React Hooks: useState, useContext

Objective:

To design and implement a responsive **expense tracker application** that enables users to:

- Add new expenses
- View total expenses in real time
- Visualize category-wise expense breakdown using a **pie chart**

This project teaches skills like:

- Real-time state management
- Chart integration with dynamic data
- Modular UI development using React and MUI

Task Overview:

You will build an interactive **expense dashboard** that allows users to:

Add Expenses

- Input **amount**, **category**, and **description**

View Expenses

- Display **total expense** value
- Show a **live pie chart** for category-wise expense distribution
- List all expenses with a **delete** option

Task Requirements

1. **Functionality**
   - Use useState or useContext for state management
   - Update pie chart automatically when new expenses are added
   - Group expenses by category and pass aggregated data to <PieChart>
2. **User Interface (UI)**
   - Use **Material UI (MUI)** and **Recharts** only
   - Layout form and chart side-by-side using <Grid>
   - Use components like:
     - <TextField>
     - <Select>, <MenuItem>
     - <Card>
     - <PieChart>
   - Display **Total Expense** at the top
   - List all expenses below with delete buttons

3. Code Structure:

File	Responsibility
App.jsx	Main layout & global state management
ExpenseForm.jsx	Handles user input for new expenses
ExpenseChart.jsx	Renders category-wise pie chart
ExpenseList.jsx	Displays list of expenses with delete

Inline Code Comments

- **State Updates:** Explain how adding or deleting an expense triggers a re-render of the chart
- **Data Grouping:** Show how expenses are grouped by category before passing to chart

Deliverables

Submit a working project folder containing:

File	Purpose
App.jsx	Main wrapper + state management
ExpenseForm.jsx	Form to input expenses
ExpenseChart.jsx	Pie chart visualization (Recharts)
ExpenseList.jsx	Table/List view of expenses

Note: Use only Material UI and Recharts – **No raw HTML or custom CSS** allowed.

Learning Outcomes

By completing this project, you will gain hands-on experience in:

- Managing application state with useState and useContext
- Visualizing data using **Recharts**
- Building responsive interfaces using **Material UI**
- Designing modular React components

Project: