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, useState, useContext

Objective

To design and implement a responsive expense tracker application that allows users to add expenses and view them visually using a pie chart categorized by type. This project teaches real-time state updates, chart integration, and dynamic data rendering using React and charting libraries.

Task Overview

As part of your hands-on learning, you are required to build an interactive dashboard that allows users to:

- Add new expenses with amount, category, and description
- View total expenses displayed prominently
- Display a live pie chart showing category-wise expense breakdown
- List each expense with an option to delete

Task Requirements

1. **Functionality:**

- Use useState or useContext to manage the list of expenses
- Add new expenses via a form (fields: amount, category, description)
- Update the chart automatically as expenses are added
- Group expenses by category and pass the totals to <PieChart>

2. **User Interface (UI):**

- Use only MUI and Recharts components
- Layout the form and chart side-by-side using <Grid>
- Use components like <TextField>, <Select>, <MenuItem>, <Card>, <PieChart>
- Display the total expense at the top of the layout
- List all expenses below the chart with a delete button

3. **Code Structure:**

- App.jsx – Main layout and global state holder
- ExpenseForm.jsx – Handles new expense inputs
- ExpenseChart.jsx – Displays a live pie chart using Recharts

- ExpenseList.jsx – Lists all expenses with delete functionality

Note: Include inline comments to explain:

- How state updates trigger the chart re-render
- How data is grouped by category for visualization

Bonus (Optional):

- Add filter options (e.g., by month or category)
- Animate the chart when new data is added
- Include an additional bar chart for time-based analysis

Deliverables

Submit a project folder that includes:

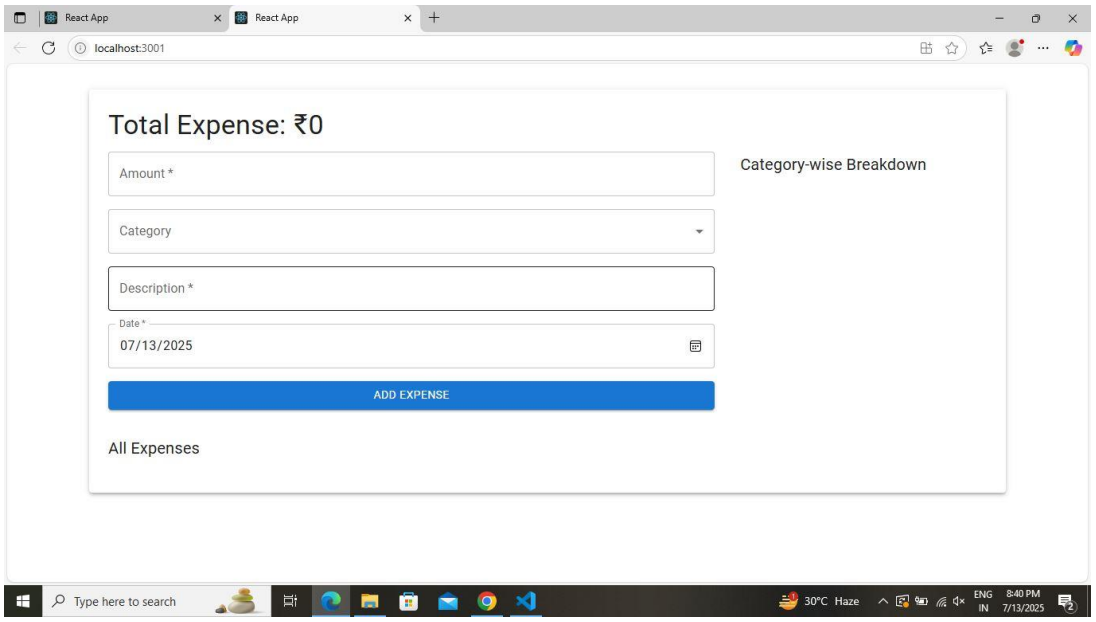
- App.jsx – Global wrapper and state management
- ExpenseForm.jsx – Form to input new expenses
- ExpenseChart.jsx – Pie chart visualization
- ExpenseList.jsx – Table/list of expense items

Note: Use only Material UI and Recharts – no custom CSS or raw HTML.

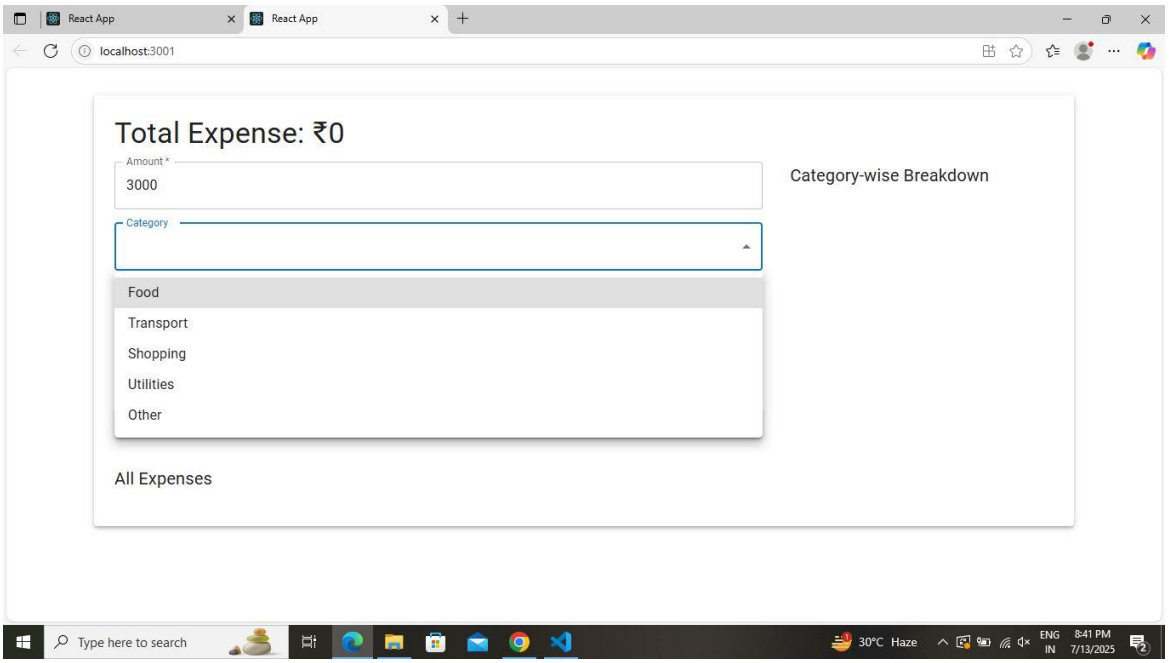
Learning Outcomes

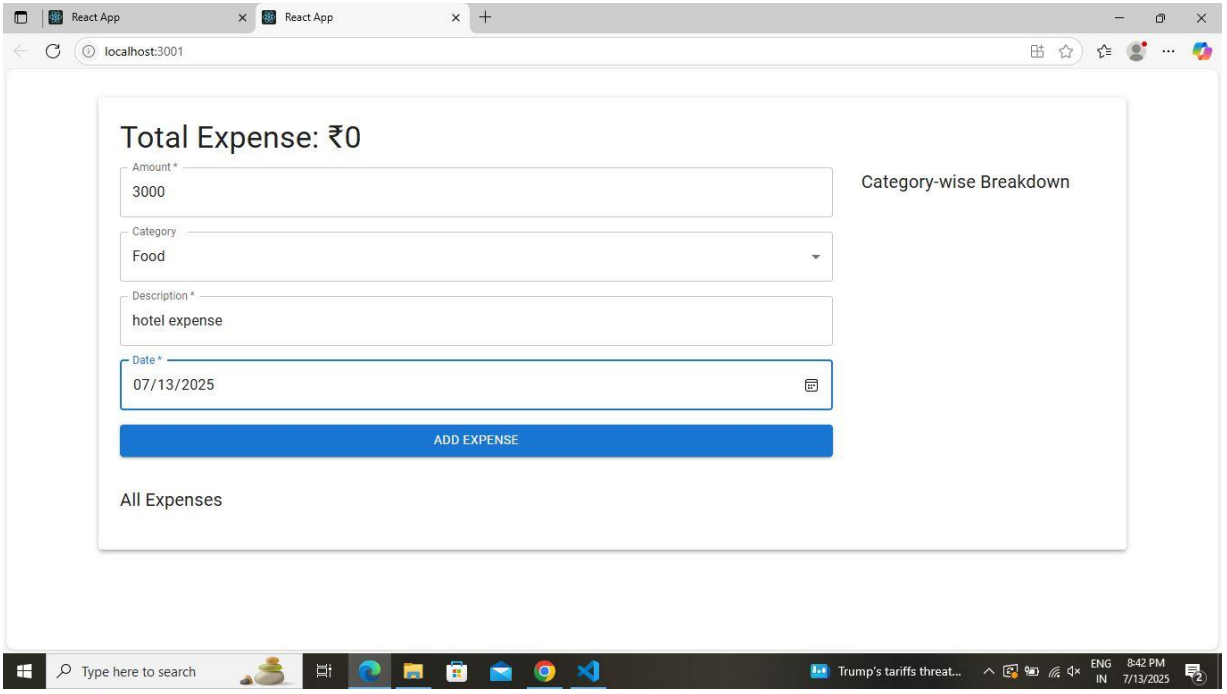
By completing this task, you will gain practical experience in:

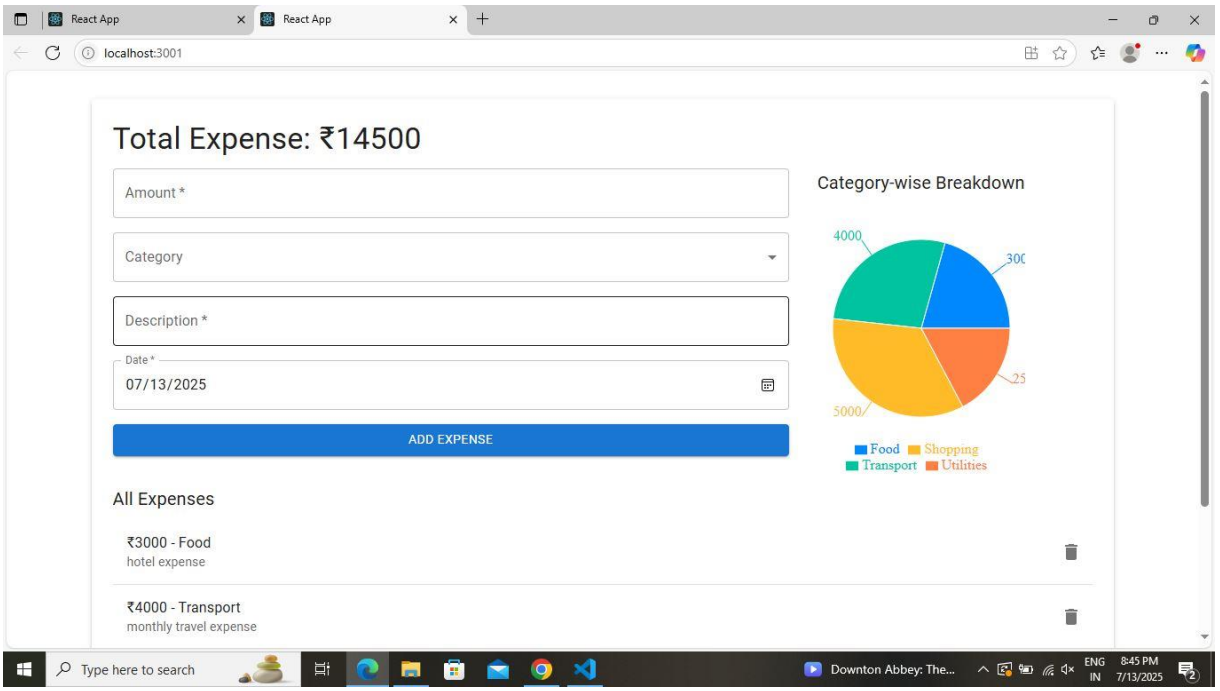
- Managing application state using useState and useContext
- Visualizing grouped data using Recharts
- Designing responsive UI with Material UI
- Building modular React components for dashboard features.

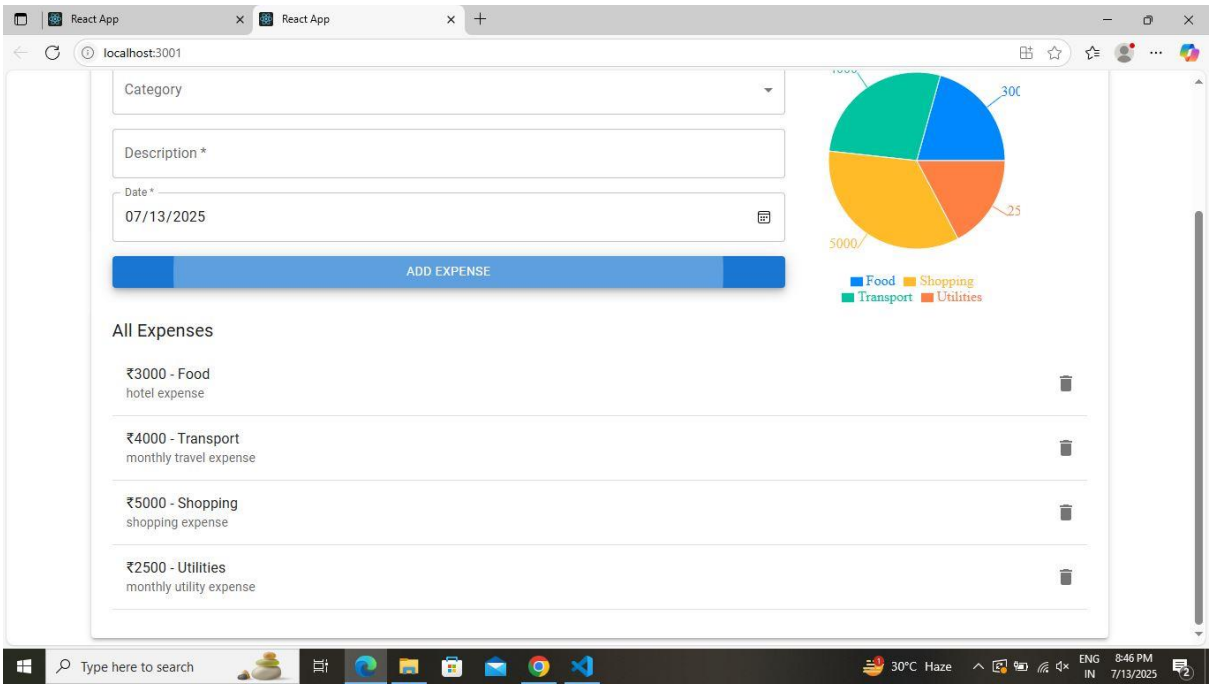


-

- 
React App
React App
localhost:3001
Total Expense: ₹0
Amount *
3000
Category
Food
Transport
Shopping
Utilities
Other
Category-wise Breakdown
All Expenses
Type here to search
30°C Haze
ENG
IN
8:41 PM
7/13/2025


React App
React App
localhost:3001
Total Expense: ₹0
Amount *
3000
Category
Food
Description *
hotel expense
Date *
07/13/2025
ADD EXPENSE
Category-wise Breakdown
All Expenses
Type here to search
Trump's tariffs threat...
ENG
IN
8:42 PM
7/13/2025


React App
React App
localhost:3001
Total Expense: ₹14500
Amount *
Category
Description *
Date *
07/13/2025
ADD EXPENSE
Category-wise Breakdown
4000
300
25
5000
Food
Shopping
Transport
Utilities
All Expenses
₹3000 - Food
hotel expense
₹4000 - Transport
monthly travel expense
Type here to search
Downton Abbey: The...
ENG
IN
8:45 PM
7/13/2025


React App
React App
localhost:3001
Category
Description *
Date *
07/13/2025
ADD EXPENSE
300
25
5000
Food
Shopping
Transport
Utilities
All Expenses
₹3000 - Food
hotel expense
₹4000 - Transport
monthly travel expense
₹5000 - Shopping
shopping expense
₹2500 - Utilities
monthly utility expense
Type here to search
30°C Haze
ENG
IN
8:46 PM
7/13/2025