Build an Personal Finance Tracker Application

Assignment -7

Objective

The Personal Finance Tracker application is designed to help users efficiently manage their finances by providing a comprehensive platform to track income, expenses, and budgets. This application consists of several key pages, each serving a specific purpose to enhance the user's financial awareness and control.

Project Requirement:

Dashboard Page:

- Global **Header** and **Footer** across the app.
- Four summary cards: **Total Income, Total Expenses, Remaining Budget, Savings**.
- Interactive charts:
  - Monthly spending trend (bar or line).
  - Category-wise expense split (pie).
- "Today's Expenses" table with amount, category, and note.
- **Date filter** to narrow all dashboard data to a specific day or range.

Transaction Page:

- Full transaction list in a sortable table with columns:
  - Type (Income/Expense)
  - Amount
  - Category
  - Date
  - Description (optional)
- **Add a Transaction** form/dialog that captures the same fields.
- **Edit** and **Delete** actions for every row with complete functionality.

Budgets Page

- Let users set a monthly budget per category (e.g., ₹ 10,000 Groceries).

- Visual progress (bars, rings, or meters) showing current spend vs. budget.
- Real-time **overspend alert** when any category exceeds its limit.

Profile Page:

- Display user details (name, email, etc.).
- **Change the Default Currency** dropdown (₹, $, €, £ …).
- Show lifetime **Total Expenses** and **Total Savings**.
- **Edit Profile** button to update user info.

Add React Toastify

- Use Toastify for success, error, and overspend notifications (e.g., "Transaction added", "Budget exceeded").
- Trigger toasts on add/edit/delete actions and overspend events.

General Front-End Rules

- Pure React JS (functional components, hooks, Context, or Redux for state).
- React Router for page navigation.
- Data persistence in localStorage or any mock API.
- Responsive design (mobile ↔ desktop).
- Clean the codebase and format all component files by using the Prettier VS Code extension.
- Add some default data

Submission Guidelines

1. Deploy the React code on Netlify and Vercel.
2. If you don't know how to deploy React code on a free hosting platform, watch this YouTube video https://youtu.be/AP0fKMrmyKA?si=wKbklcCA3rGCFDb7
3. **Deploy Link:** Submit Deploy Link in Readme.md file of task-related GitHub repo.
4. **Readme.md:** Create the readme.md file like Professional Level, which includes only project-related content.
5. **Submission Link:** Submit the GitHub repo Link in the Submission Box.