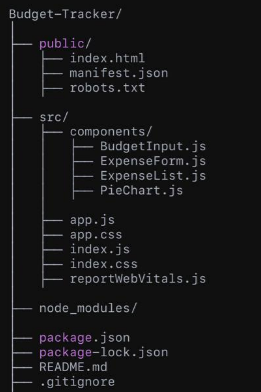
**Budget Tracker Application:**

**Project Structure:**

****

**Project Overview**

The Budget Tracker Application is a web-based tool developed using ReactJS to help users efficiently manage their income, expenses, and budgets. It features a responsive design, modular architecture, real-time updates, and data visualization through interactive charts, offering users a seamless and engaging experience.

**Features**

1. **Add and Manage Budgets**:
   * Input a monthly budget with an intuitive form.
   * Track real-time remaining balance as expenses are added.
2. **Expense Management**:
   * Log expenses by category (e.g., food, travel, entertainment).
   * View a detailed list of recorded expenses with descriptions.
3. **Data Visualization**:
   * Interactive pie charts to visualize spending distribution across categories.
4. **Persistent Data Storage**:
   * Data is stored in the browser's local storage, allowing users to retain their budgets and expenses across sessions.
5. **Responsive Design**:
   * The app is mobile-friendly and adapts to different screen sizes for an optimal user experience.

**Technologies Used**

* **ReactJS**: Core framework for building a dynamic and interactive user interface.
* **CSS and Bootstrap**: For creating a responsive and visually appealing design.
* **Local Storage**: Ensures data persistence across sessions.
* **Chart.js (or any library)**: For rendering interactive pie charts for data visualization.

**Project Architecture**

* **Public Folder**:
  + Contains index.html (entry point of the app) and supporting files like manifest.json.
* **src Folder**:
  + **Components**: Contains modular components used across the app.
    - BudgetInput.js: Handles user input for the monthly budget.
    - ExpenseForm.js: A form for logging expenses.
    - ExpenseList.js: Displays a detailed list of all expenses.
    - PieChart.js: Visualizes spending patterns using a pie chart.
  + **app.js**: The main component where all other components are integrated.
  + **app.css**: Styles specific to the application.
  + **index.js**: Renders the React app in the DOM.

**Implementation Highlights**

1. **State Management**:
   * Used React hooks like useState and useEffect for managing and updating the application state in real time.
2. **Responsive Design**:
   * Integrated Bootstrap’s grid system and utilities to ensure a mobile-friendly, responsive layout.
3. **Data Persistence**:
   * Leveraged local storage to save budgets and expenses so users can access their data even after closing the application.
4. **Interactive Visualization**:
   * Used pie charts to present spending data in an engaging, easy-to-understand format.

**How the App Works**

1. **User Workflow**:
   * The user enters their monthly budget using the **BudgetInput** component.
   * Expenses are added through the **ExpenseForm**, specifying details like amount, category, and description.
   * The **ExpenseList** displays all recorded expenses, and the **PieChart** updates dynamically to reflect spending distribution.
2. **Real-Time Updates**:
   * React hooks ensure that every user action (e.g., adding an expense) triggers an immediate update across all components.

**Future Enhancements**

1. Add user authentication for secure access to personal financial data.
2. Integrate a backend (e.g., Firebase or Node.js) for storing data online.
3. Export data as CSV or PDF for offline analysis.

**Screenshots:**

