**Expense Tracker**

1. **Expense Management:**

- Create a secure database for storing expense information, including expense categories, dates, amounts, payment methods, and descriptions.

2. **Expense Entry:**

- Implement a module for users to enter and categorize expenses.

- Provide fields for the expense date, amount, category, and description.

3. **Expense Tracking:**

- Develop a system that allows users to track their expenses over time.

- Implement features like sorting and filtering by date or category.

4. **Budget Management:**

- Include a budgeting module where users can set monthly or yearly spending limits.

- Provide alerts or notifications when users approach or exceed their budget.

5. **Reports and Analytics:**

- Offer tools for generating reports and analytics on spending patterns, expense categories, and budget adherence.

- Allow users to visualize their financial data through charts and graphs.

6. **Income Tracking:**

- Add functionality for users to track their income sources alongside expenses.

- Calculate net income and savings.

7. **Access Control Levels:**

- Administrator: Administrators have full control over the system, including user management, system settings, and access to all expense records and functionalities.

- Standard Users: They can manage their own expenses, create budgets, and view their financial reports. They do not have access to other users' financial data.

8. **User Dashboard:**

- Provide a user dashboard where users can get an overview of their financial health, including income, expenses, and budget status.

**Technology Stack (Expense Tracker):**

**Back-end Application**:

- Use ASP.NET Core Web API for the back end.

- Implement MS SQL as the database for storing expense data.

- Create API documentation using Swagger for the back-end API.

**Front-end Application:**

- Utilize React.js as the web development framework for the front end.

- Choose a UI framework for styling (Material-UI, React-Bootstrap) for the user interface.