## **Budget Planner with Data Visualizations**

**Submitted by:** Shalini D  
**College:** Agni College of Technology

### **1. Introduction**

The **Budget Planner with Data Visualizations** is a responsive, modern web application that helps users manage their personal finances. It allows users to record income and expenses, categorize their spending, and visualize their data through interactive charts.  
The project is designed using React JS with Tailwind CSS for a professional, gradient-themed interface and Chart.js for real-time data visualization. All user data is stored locally in the browser, ensuring privacy and offline functionality.

**2. Abstract**

Budgeting is an essential part of financial discipline. This project offers an intuitive and attractive interface that allows users to easily track their income, expenses, and balance.  
Each transaction added by the user automatically updates the summary section and two visual charts a **Pie Chart** displaying category-wise expense distribution and a **Bar Chart** showing monthly income-expense trends.  
Because it is entirely frontend-based and uses **local storage**, it does not depend on any backend or third-party service, making it lightweight, secure, and ideal for quick deployment or GitHub hosting.

### **3. Tools Used**

| **Layer** | **Technology** | **Purpose** |
| --- | --- | --- |
| Frontend | **React JS** | Builds dynamic UI components |
| Styling | **Tailwind CSS** | Provides responsive, gradient-based design |
| Charts | **Chart.js / react-chartjs-2** | Renders interactive charts |
| Data Storage | **Browser local Storage** | Stores user data offline |
| IDE | **Visual Studio Code** | Code editing and debugging |
| Hosting | **GitHub Pages (optional)** | To publish the project online |

### **4. Steps Involved in Building the Project**

1. **Project Setup:**  
   Initialized a React JS project using create-react-app and integrated Tailwind CSS and Chart.js libraries.
2. **UI Design:**  
   Implemented a clean, modern gradient layout with a responsive grid that adapts to all screen sizes.
3. **Form and State Management:**  
   Created forms to add income and expense entries with fields for amount, category, and date. Used React state hooks (use State) to manage user input dynamically.
4. **Local Storage Integration:**  
   Used the browser’s local Storage API to persist user transactions even after page refresh or browser restart.
5. **Data Visualization:**  
   Implemented **Pie Chart** for expense categories and **Bar Chart** for monthly statistics using Chart.js for better clarity.
6. **Summary and Filtering:**  
   Calculated total income, total expenses, and overall balance; displayed them in real time as users add new transactions.
7. **Testing & Responsiveness:**  
   Verified layout on mobile, tablet, and desktop; ensured charts re-render smoothly across different devices.

### **5. Conclusion**

The **Budget Planner with Data Visualizations** demonstrates the power of frontend frameworks in building complete, data-driven applications without relying on any backend.  
It combines functional simplicity with an elegant modern interface, offering a visually rich experience for managing finances.  
The project enhances understanding of **React JS, Tailwind CSS, and Chart.js**, and serves as a valuable portfolio piece showing skills in responsive design, component architecture, and local data handling.

### **6. Future Enhancements**

* Add user authentication for multi-user access.
* Integrate cloud storage or database for remote data backup.
* Include PDF export of monthly reports and goal tracking.