

**TITLE:**

**SHOPPING LIST APPLICATION – FRONTEND**

**SUBTITLE:**

**REACT APPLICATION WITH VERCEL DEPLOYMENT**

Your paragraph text

**III-BCA  
23SUCA50**



**This project is a full-stack web application developed to help users manage their shopping list within a fixed budget.**

**The application allows users to:**

- **Set a total budget**
- **Add shopping items with quantity and price**
- **Automatically calculate item totals**
- **Track remaining budget visually**
- **Delete items and update the budget in real time**

**The project follows modern DevOps practices, including:**

- **Frontend and backend separation**
- **REST API communication**
- **Dockerization**
- **CI/CD using GitHub Actions**
- **Code quality analysis using Sonar**
- **Cloud deployment using Vercel (frontend) and Render (backend)**



# How the Project Works (Flow):

- **Frontend (React)**
- **User enters a budget**
- **Adds items (name, quantity, price)**
- **UI calculates total cost and remaining budget**
- **Sends requests to backend using Axios**





# Frontend Description (React)

- Built using React JS
- Component-based architecture
- Uses Axios for API calls
- Real-time budget progress bar
- Clean and responsive UI
- Frontend Features
- Budget input and lock
- Add item form
- Dynamic progress bar
- Item list table
- Delete functionality





# Development Frontend:

The frontend of the Shopping List Application is developed using React JS. It provides an interactive user interface that allows users to add, view, and manage shopping items. The frontend communicates with the backend using REST APIs

localhost:3000

Chat

## Manage Your Budget and Shopping List

"Every rupee counts 🌟"

Enter Budget:  Refresh

Remaining Budget: 50 / 100 (50.00%)

Enter item name

Select category ▼

Enter price

Add Item

Item	Category	Quantity	Price	Total	Actions
apple		1	50	50	<button>Delete</button>



# Frontend Code Architecture:

The frontend follows a component-based architecture using React. Reusable components are used to manage UI logic and API interactions.

```
shoppinglist-frontend/  
├── .github/  
│   └── workflows/  
│       └── build.yml  
├── public/  
│   └── index.html  
├── src/  
│   ├── api/  
│   │   └── api.js  
│   ├── components/  
│   │   ├── AddItemForm.js  
│   │   ├── ProgressBar.js  
│   │   └── ShoppingList.js  
│   ├── App.js  
│   ├── App.test.js  
│   ├── index.js  
│   └── index.css  
├── Dockerfile  
├── package.json  
├── package-lock.json  
├── sonar-project.properties  
├── vercel.json  
├── README.md  
└── .gitignore
```

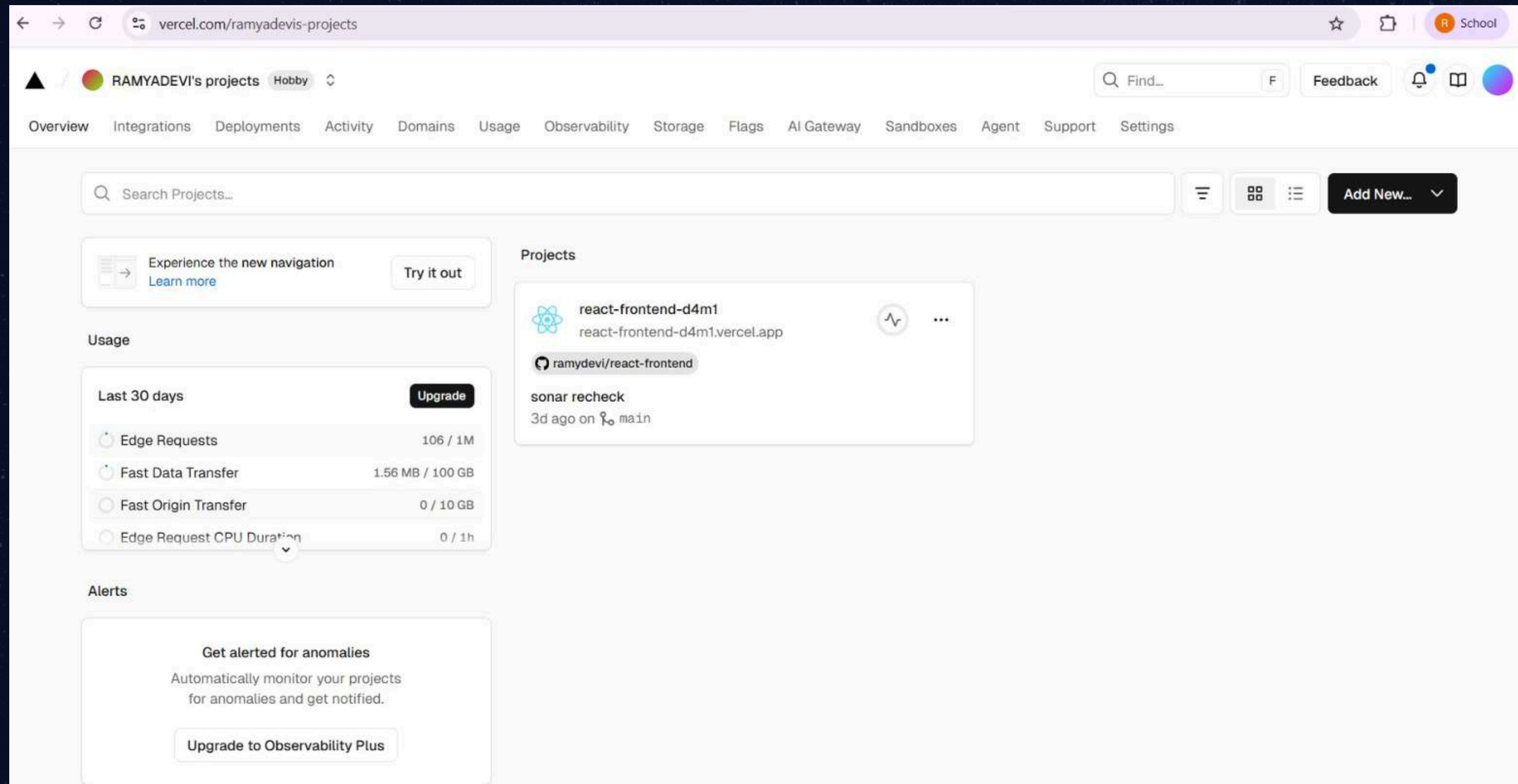
7:17 am





# Frontend Deployment – Vercel (Vercel Deployment):

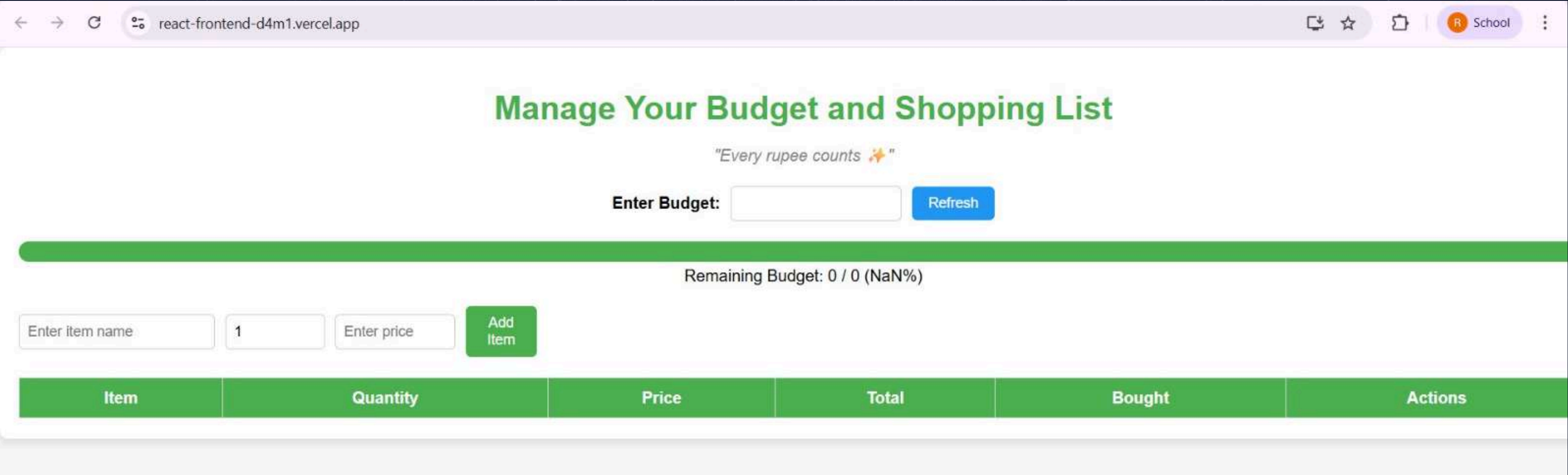
The frontend was deployed on Vercel. Vercel automatically builds and deploys the React application from GitHub.





# Vercel Deployment with Domain:

Vercel provides a public domain URL which allows users to access the application globally without local setup.



The screenshot shows a web browser window with the address bar displaying `react-frontend-d4m1.vercel.app`. The page title is "Manage Your Budget and Shopping List" in green. Below the title is a quote: "Every rupee counts 🌟". There is a form to "Enter Budget:" with a text input field and a blue "Refresh" button. A green progress bar is shown below the budget input, with the text "Remaining Budget: 0 / 0 (NaN%)" centered below it. At the bottom, there is a form to "Add Item" with three input fields: "Enter item name", "1" (quantity), and "Enter price", followed by a green "Add Item" button. Below the form is a table with a green header row and one empty data row.

Item	Quantity	Price	Total	Bought	Actions



# Frontend Challenges Faced:

- API integration issues
- CORS handling
- CI build warnings

