

**TITLE:**

SHOPPING LIST APPLICATION – FRONTEND

**SUBTITLE:**

REACT APPLICATION WITH VERCCEL DEPLOYMENT

Your paragraph text

III-BCA  
23SUCAS0

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.

The screenshot shows a web browser window for 'localhost:3000'. The title bar displays the URL. The main content area has a green header with the text 'Manage Your Budget and Shopping List' and a subtitle 'Every rupee counts!'. Below this is a budget input field with '100' and a 'Refresh' button. A progress bar indicates a remaining budget of '50 / 100 (50.00%)'. At the bottom, there's a table for managing items with columns: Item, Category, Quantity, Price, Total, and Actions. A single item 'apple' is listed with a quantity of 1, price of 50, and total of 50. There are 'Delete' and 'Edit' buttons in the Actions column.

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

The screenshot shows the Vercel dashboard interface. At the top, there's a navigation bar with links for Overview, Integrations, Deployments, Activity, Domains, Usage, Observability, Storage, Flags, AI Gateway, Sandboxes, Agent, Support, and Settings. A search bar labeled "Search Projects..." is located at the top left, along with a "Find..." button and other icons. On the right side of the header, there are buttons for "Feedback", a bell icon, and a profile picture. Below the header, a banner says "Experience the new navigation" with a "Try it out" button. The main area is titled "Projects" and lists three projects:

- react-frontend-d4m1** (by ramydevi/react-frontend) - Last updated 3d ago on main. It has a green "Up to date" status icon.
- sonar recheck** (by ramydevi/react-frontend) - Last updated 3d ago on main. It has a yellow "Warning" status icon.
- react-frontend-d4m1.vercel.app** (by ramydevi/react-frontend) - Last updated 3d ago on main. It has a green "Up to date" status icon.

On the left side of the dashboard, there are sections for "Usage" and "Alerts". The "Usage" section shows metrics for the last 30 days, including Edge Requests (106 / 1M), Fast Data Transfer (1.56 MB / 100 GB), Fast Origin Transfer (0 / 10 GB), and Edge Request CPU Duration (0 / 1h). The "Alerts" section encourages users to "Get alerted for anomalies" by automatically monitoring projects for anomalies and upgrading to Observability Plus.

# 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 URL `react-frontend-d4m1.vercel.app` in the address bar. The page title is "Manage Your Budget and Shopping List" with the subtitle "*Every rupee counts* 💰". A budget input field contains "0" and a "Refresh" button is nearby. Below this is a green bar displaying "Remaining Budget: 0 / 0 (NaN%)". At the bottom, there's a form for adding items with fields "Enter item name", "1", "Enter price", and a green "Add Item" button. A table header with columns "Item", "Quantity", "Price", "Total", "Bought", and "Actions" is visible.

Item	Quantity	Price	Total	Bought	Actions
------	----------	-------	-------	--------	---------

# Frontend Challenges Faced:

- API integration issues
- CORS handling
- CI build warnings

