Milestone 2 – Product Dashboard Report

Student Name: ***Amirtha Varshini S***

Role: Full Stack MERN Developer Trainee

Assessment: Milestone 2 – React Product Dashboard with Express Backend

Date: 25/11/2025

Introduction

This milestone focuses on building a React-based Product Dashboard with routing, backend mock API integration, product detail pages, context-based global state sharing, and form handling using Formik with Yup validation. The project uses a manual folder structure with a backend server built using Express.js and an interactive UI built using React and Bootstrap.

The goal of this project is to demonstrate core Frontend and Backend development skills such as:

- Component architecture (class & functional components)
- Routing & navigation
- REST API integration
- Form validation & submission
- State management using Context API
- Error handling & loading states
- Lazy loading (Bonus marks)

Folder Structure

```
milestone-2-product-dashboard/
  frontend/
    backend/
      server.js
    src/
      components/
      context/
      App.js
      index.js
    screenshots/
    README.md
```

Technologies Used

Frontend	Backend	Other Tools
React.js	Node.js & Express.js	Axios
React Router	In-memory dataset API	Bootstrap
Formik & Yup	CORS	Google Docs
Context API	REST handlers	VS Code, Postman

User Story Implementations

User Story 1 – Display Product Catalog with Components

Requirement

- Display list of products using React components & Bootstrap styling
- Use **class and functional components**, props, state & event handling
- Add **favorite/unfavorite** functionality

Implementation Summary

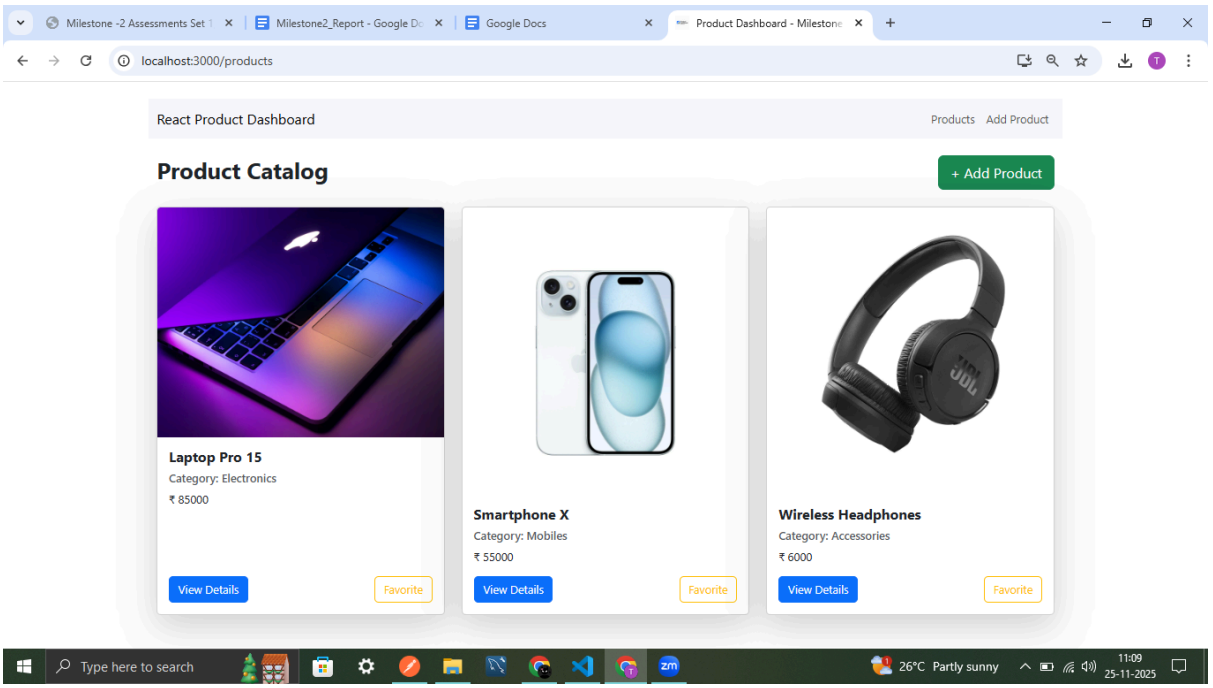
- `ProductList.jsx` created as a **Class Component**
- `ProductCard.jsx` created as a **Functional Component**
- Favorite toggle handled using state
- Bootstrap grid UI design

Relevant Files

- `src/components/ProductList.jsx`
- `src/components/ProductCard.jsx`

Output Screenshot

(Q1-product-list.png)



User Story 2 – Product Detail Page with React Router

Requirement

- Dynamic product detail page using URL parameters
- Load details using backend API with loading & error handling
- Implement API call using Axios

Implementation Summary

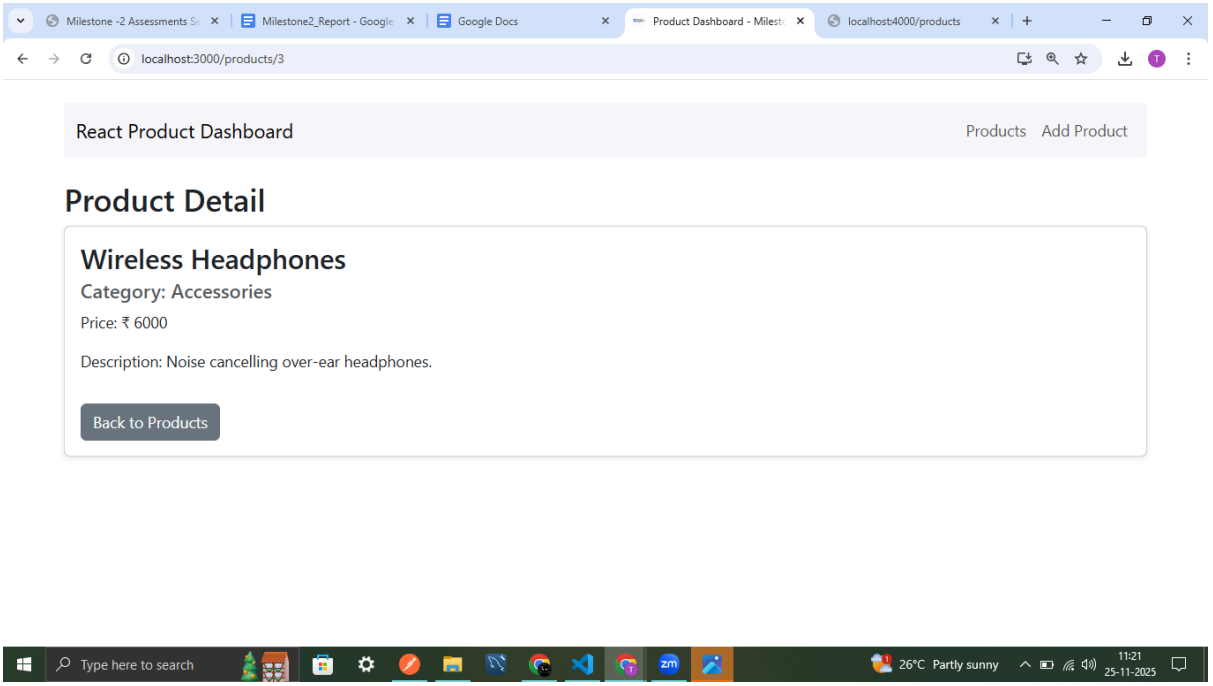
- Used React Router: `Route path="/products/:id"`
- Detail page built with Handling loading / error UI
- API call: `GET /products/:id`

Relevant Files

- `App.js`
- `ProductDetail.jsx`
- `backend/server.js`

Output Screenshot

(Q2-product-detail.png)



User Story 3 – Form for Adding New Product

Requirement

- Add product form built using Formik + Yup for validation
- Submit using **POST /products** API
- Update global state through **Context API**

Implementation Summary

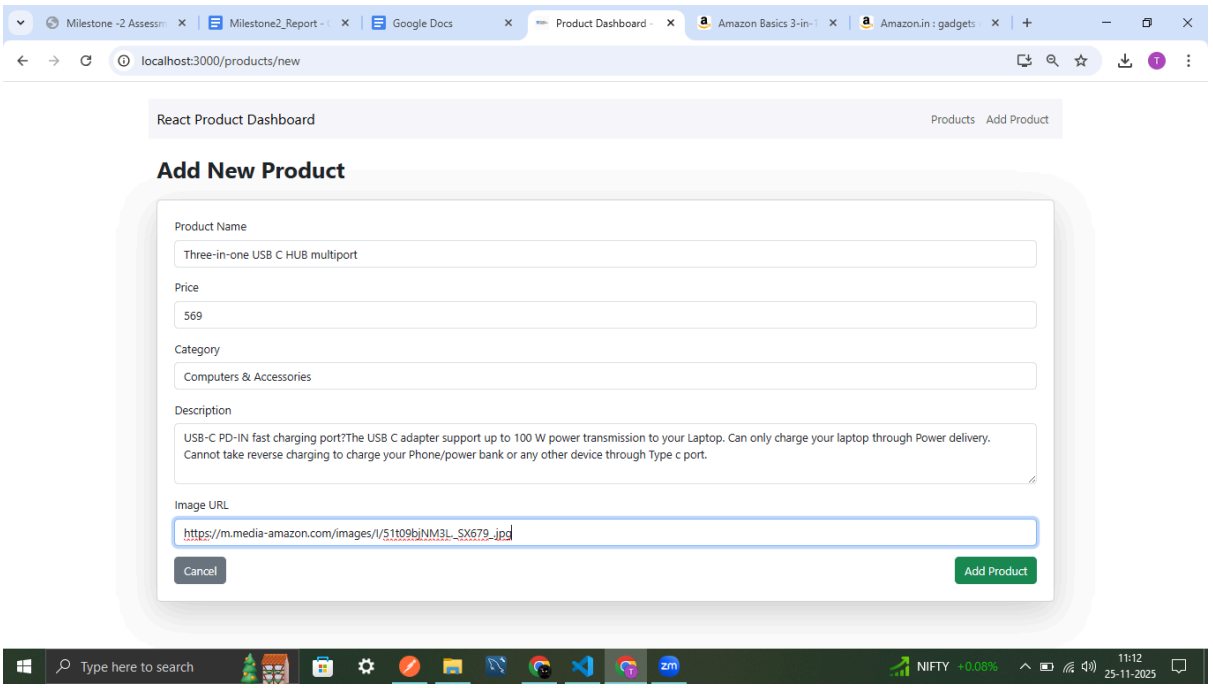
- Created `AddProductForm.jsx`
- Form validation using Yup schemas
- On success, product is added & list refreshed

Relevant Files

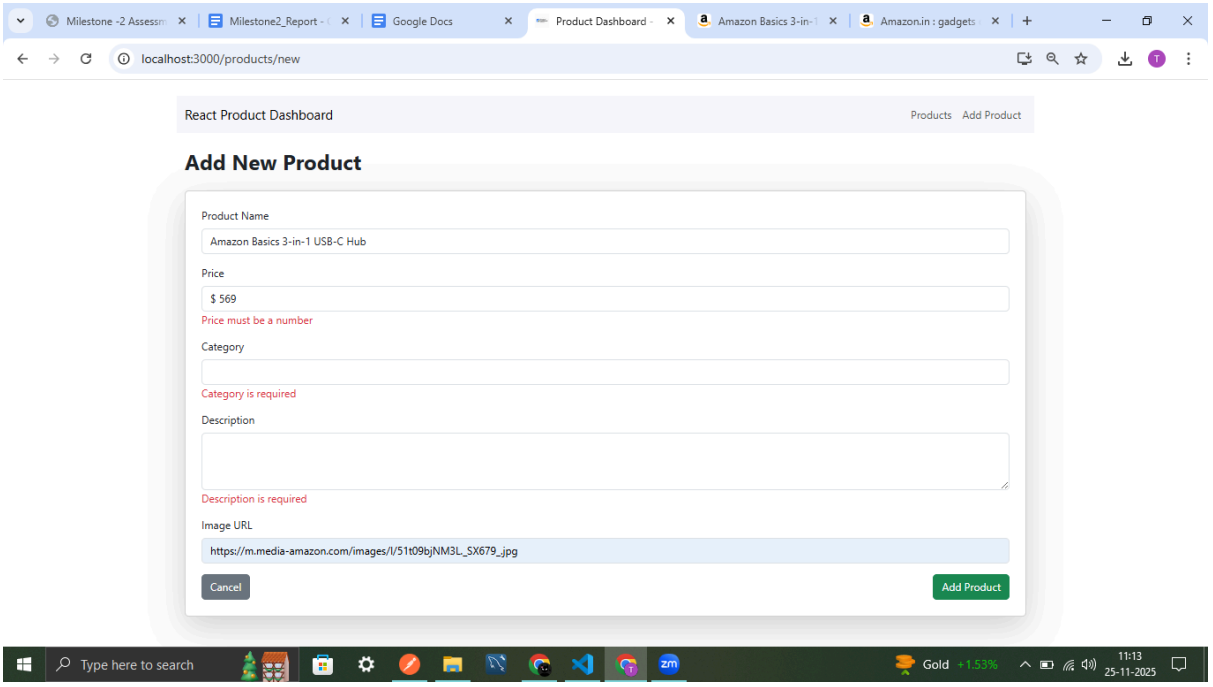
- `src/components/AddProductForm.jsx`
- `src/context/ProductContext.jsx`
- `backend/server.js`

Output Screenshots

(*Q3-add-form-success.png*)



(*Q3-add-form-invalidation.png*)



Bonus Feature – Lazy Loading

Requirement

- Implement React.lazy & Suspense

Summary

- Product Detail page loaded dynamically to improve performance

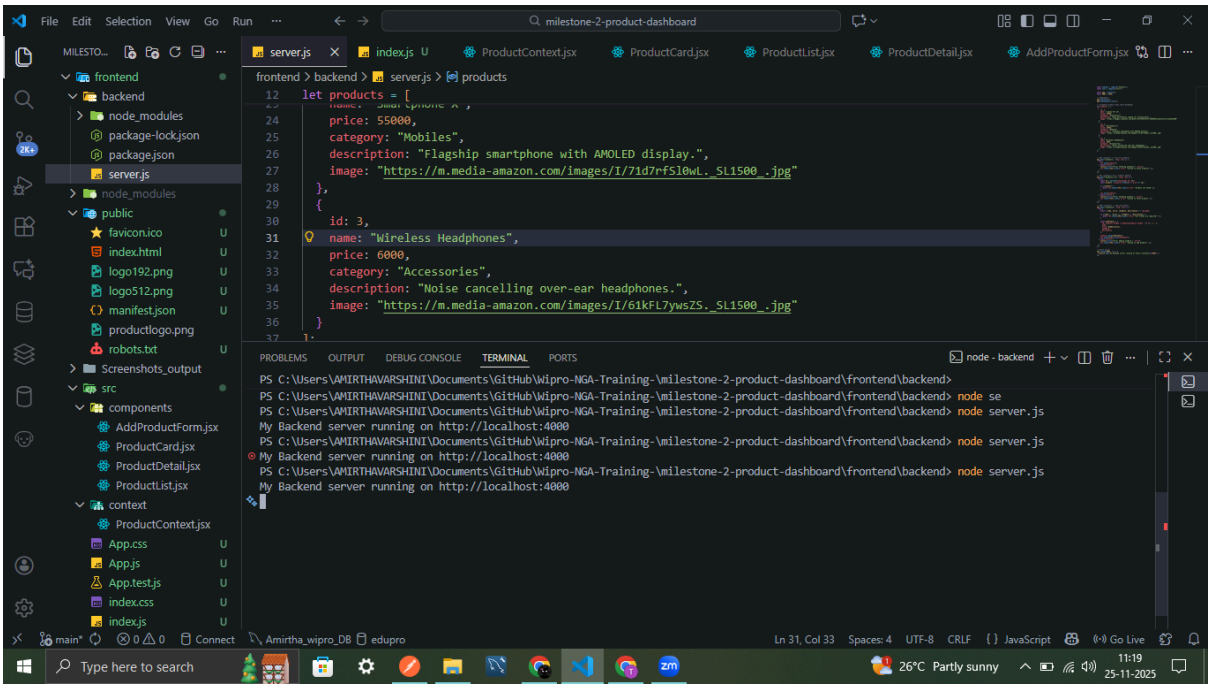
Relevant Code Snippet

```
const ProductDetail = lazy(() =>
import("./components/ProductDetail"));
```

Backend API Endpoints

Method	Endpoint	Description
GET	`/products`	Returns all products
GET	`/products/:id`	Returns single product
POST	`/products`	Adds new product

Backend Running Screenshot



Testing & Verification

- Manual testing performed in browser and console
- Verified routing navigation, error states & form validation
- Verified data persistence in state after POST request

Screenshots Folder

1. Product Catalog – `03-product-catalog.png`
2. Product Detail – `04-product-detail.png`
3. Add Product Form – `05-add-product-form-empty.png`
4. Validation Errors – `06-add-product-validation-errors.png`
5. After Adding Product – `07-product-list-after-add.png`
6. Backend Server – `01-backend-server-running.png`

Conclusion

This milestone project successfully demonstrates:
✔ Understanding of React component architecture
✔ Logical routing structure and UI navigation
✔ REST API communication using Axios
✔ Form implementation with validation
✔ Global state sharing using Context API
✔ Backend handling via Express.js
✔ Lazy loading optimization

I gained practical hands-on experience building production-level MERN application structure and strengthened essential skills required for professional full-stack development.

Submitted By

Name: ***Amirtha Varshini S***

Milestone: 2 – Completed Successfully