Product Inventory Management System

1. Requirements

Project Overview:

The **Product Inventory Management System** is a web application designed to help users efficiently manage products by performing CRUD (Add, View, Update, Delete) operations, searching/filtering products, and ensuring proper authentication and validation. The system is built using **Angular (Frontend)** and **JSON Server (Backend)**.

Business Needs & Objectives:

- **User Authentication:** Secure login and registration.
- **Product Management:** CRUD operations on products.
- **Filtering & Searching:** Allow users to search and filter products dynamically.
- **Data Validation & Exception Handling:** Prevent invalid data entries.

---

2. Project Flow

DFD Explanation:

1. **User Interaction:** The user accesses the system via the UI.
2. **Authentication Process:**
   - New users register, and returning users log in.
   - Authentication is handled by auth.service.ts.
3. **Product Management Workflow:**
   - Users can add, view, edit, and delete products.
   - Data is sent to the JSON Server via API calls (data.service.ts).
   - On success, updated data is displayed in the UI.
4. **Search & Filter:** Users can dynamically search for products.
5. **Validation & Exception Handling:**
   - Prevents incorrect data input.
   - Catches API errors to ensure smooth UI interactions.

Project Flow:

1. **User Login/Register** → Authentication using JSON Server.
2. **Dashboard** → Displays product list.
3. **Product Management** → Add, Update, Delete, View product details.
4. **Search & Filter** → Users can refine product lists dynamically.
5. **Validation & Exception Handling** → Ensures only valid data is stored.

---

3. Project Code Structure

Definitions of Key Modules:

- **Auth Module:** Responsible for user authentication, including login, registration, and route protection using auth.guard.ts.
- **Core Module:** Contains essential services like auth.service.ts for authentication and data.service.ts for API interactions.
- **Inventory Module:** Handles product management operations such as adding, updating, deleting, and viewing products.

Routes Configuration:

The project uses Angular Routing to navigate between components. The Angular Router enables navigation between different components in the application by mapping URL paths to components.

RxJS (library)

RxJS (Reactive Extensions for JavaScript) is a library that helps us handle asynchronous operations efficiently using Observables. It is mainly used in Angular for handling API calls, user events, and data streams.

The project uses Angular Routing to navigate between components:

- **Auth Routes:**
  - /login → Navigates to the login page.
  - /register → Navigates to the registration page.
- **Inventory Routes:**
  - /products → Displays the product list.
  - /product/add → Allows adding a new product.
  - /product/edit/:id → Edits an existing product.
  - /product/view/:id → Displays product details.

- **Common Routes:**
    - /about → Navigates to the About page.
    - ** (Wildcard) → Redirects to a 404 page if the route does not exist.

Definitions of Key Modules:

- **Auth Module:** Responsible for user authentication, including login, registration, and route protection using auth.guard.ts.
- **Core Module:** Contains essential services like auth.service.ts for authentication and data.service.ts for API interactions.
- **Inventory Module:** Handles product management operations such as adding, updating, deleting, and viewing products.

Folder Breakdown:

```
ANGULAR-PRODUCT-INVENTORY/
|— src/
│   ├— app/
│ │   ├— core/
│ │ │   ├— auth.guard.ts
│ │ │   ├— auth.service.ts
│ │ │   ├— data.service.ts
│ │   ├— features/
│ │ │   ├— about/
│ │ │ │   ├— about.component.ts
│ │ │   ├— auth/
│ │ │ │   ├— register/
│ │ │ │ │   ├— register.component.ts
│ │ │ │   ├— sign-in/
│ │ │   ├— inventory/
│ │ │ │   ├— add-product/
│ │ │ │ │   ├— add-product.component.ts
│ │ │ │   ├— product-list/
│ │ │ │   ├— update-product/
│ │ │   ├— shared/
│   ├── assets/
│   │   ├── db.json
│   ├── environments/
│   ├── styles.css
│── package.json
│── README.md
│── tsconfig.json
```

---

4. Functionalities & Concepts Used

Functionalities:

1. **User Authentication** → Register/Login using auth.service.ts.
2. **Product Management** → Add, Update, Delete, and View products.
3. **Search & Filter** → Dynamically refine product lists.

Concepts Used:

- **Angular Routing & Route Guards** (Protecting Routes).
- **Dependency Injection** (Services for API calls).
- **HTTP Client for JSON Server API communication**.
- **Form Validation using Angular Validators**.

---

5. Data Validation & Exception Handling

Form Validation in ********register.component.ts:

- Required fields, Email validation, Min/Max length checks.
- Ensuring correct password complexity.

Exception Handling in ********data.service.ts:

- Using catchError to handle API errors and prevent crashes.
- Displaying user-friendly error messages.

---

6. Demo Flow (Live Walkthrough)

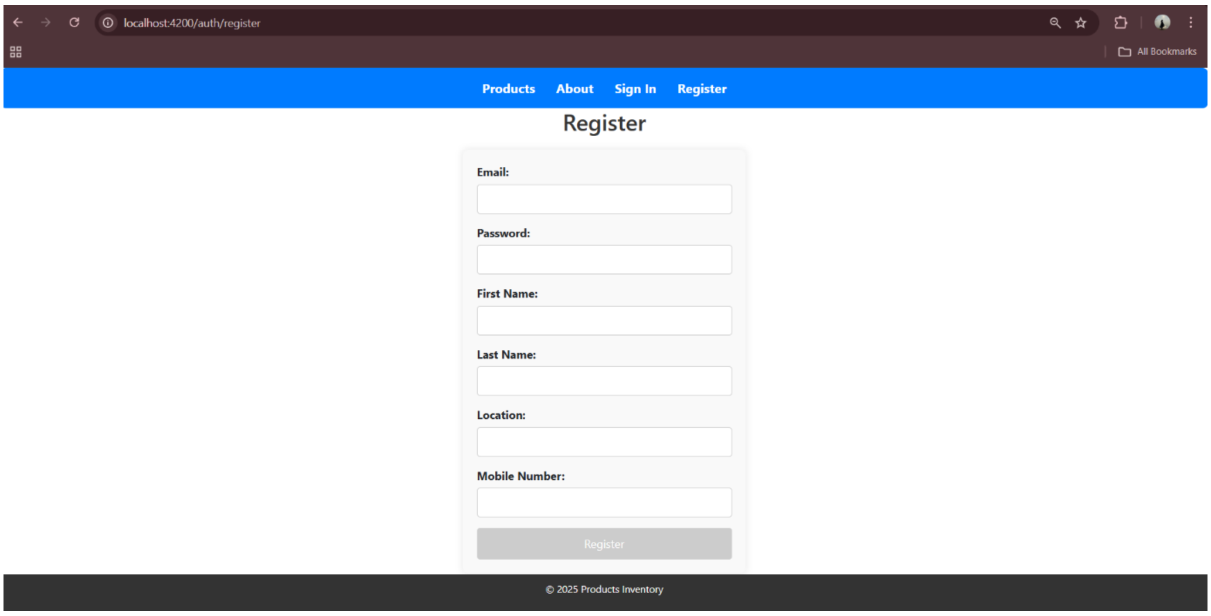
1. **Login/Register**
2. **Add a New Product**

3. **View Product List (Apply Filters)**
4. **Edit/Delete Product**
5. **Logout**

---

8. Output

Registering the user:



Register

Email:

Password:

First Name:

Last Name:

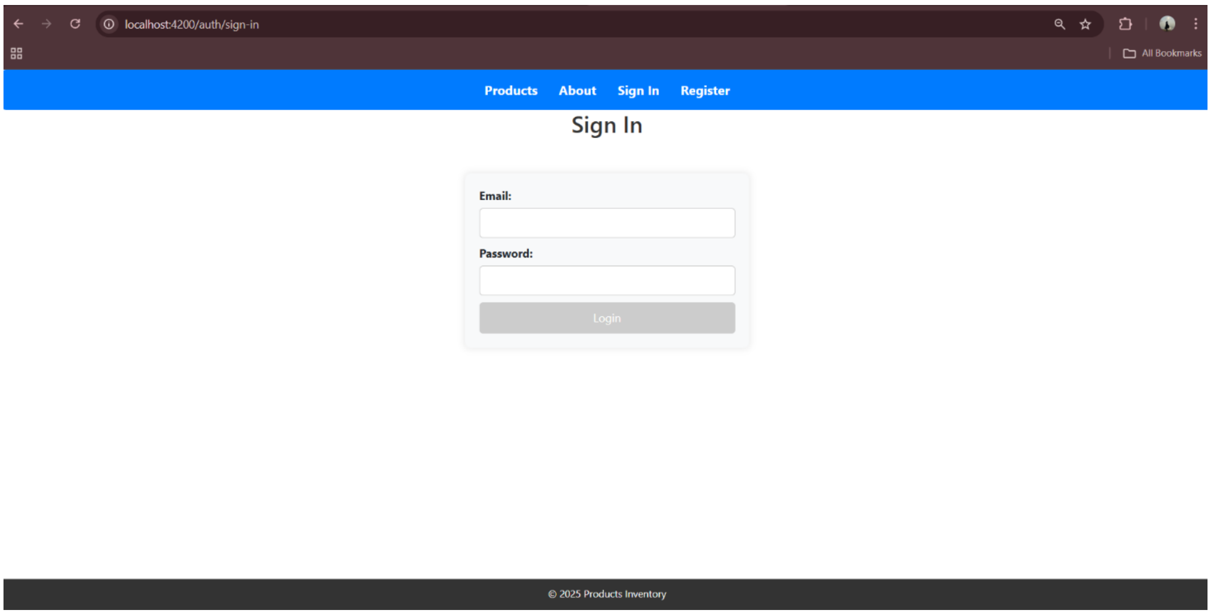
Location:

Mobile Number:

Register

© 2025 Products Inventory

User Sign-In :



Sign In

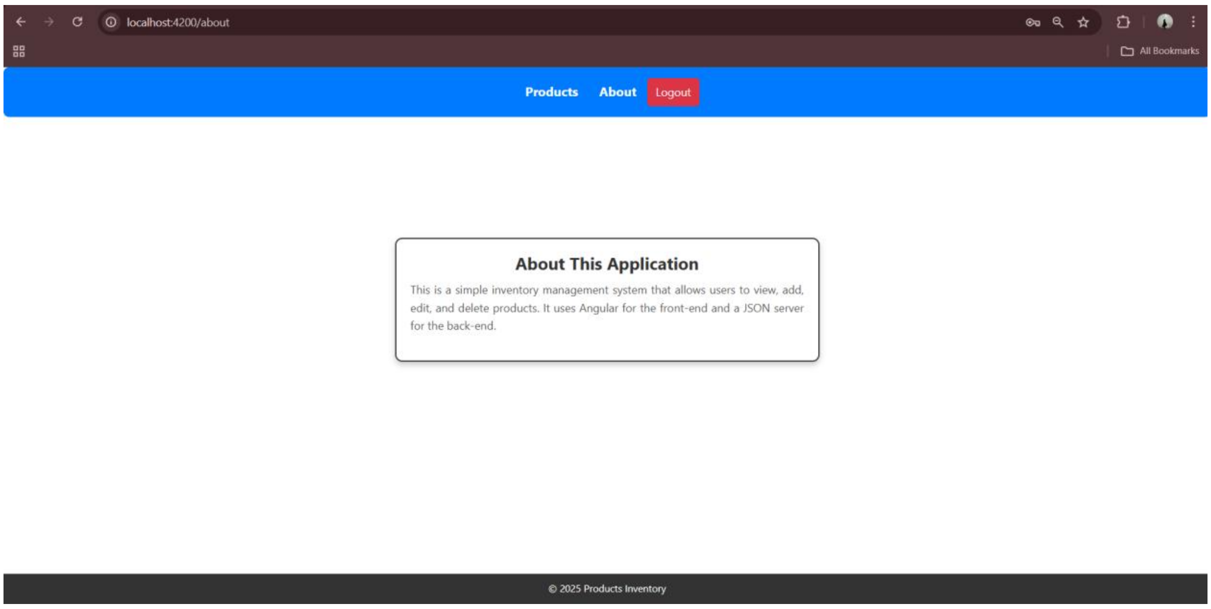
Email:

Password:

Login

© 2025 Products Inventory

About Page :

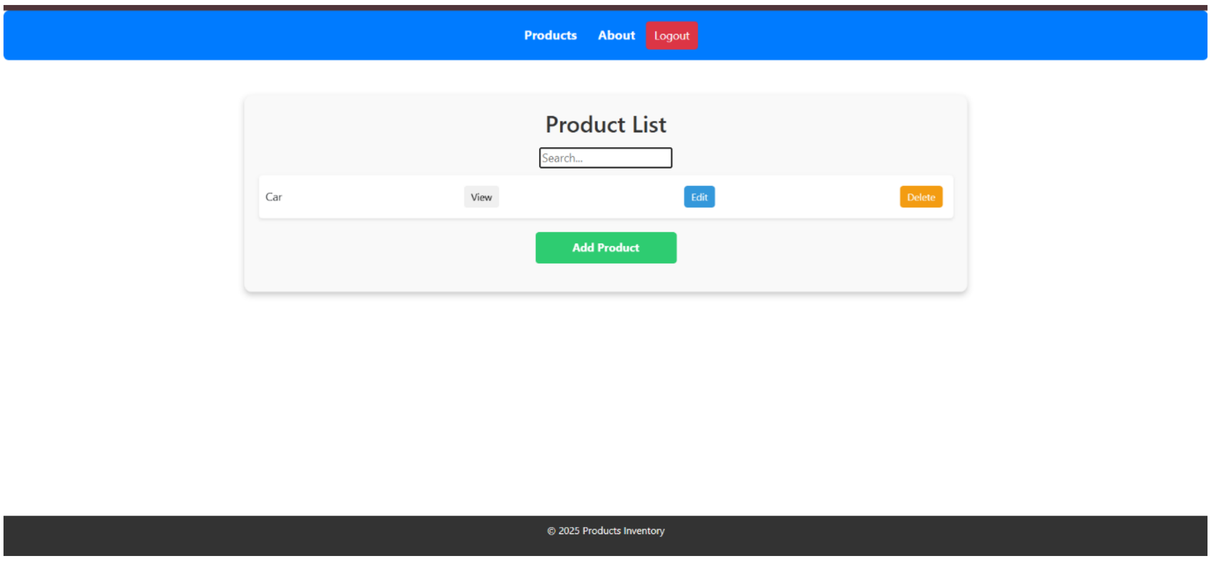


About This Application

This is a simple inventory management system that allows users to view, add, edit, and delete products. It uses Angular for the front-end and a JSON server for the back-end.

© 2025 Products Inventory

Product List page :



Product List

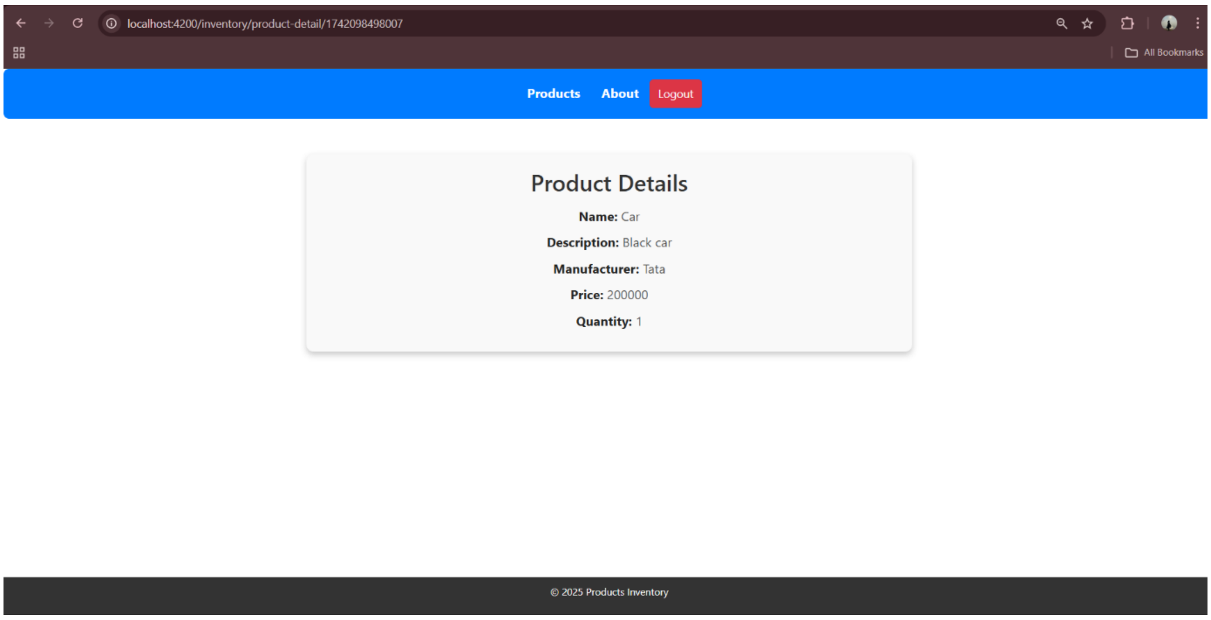
Search...

Car	View	Edit	Delete

Add Product

© 2025 Products Inventory

Product View page :


localhost:4200/inventory/product-detail/1742098498007

Product Details

Name: Car

Description: Black car

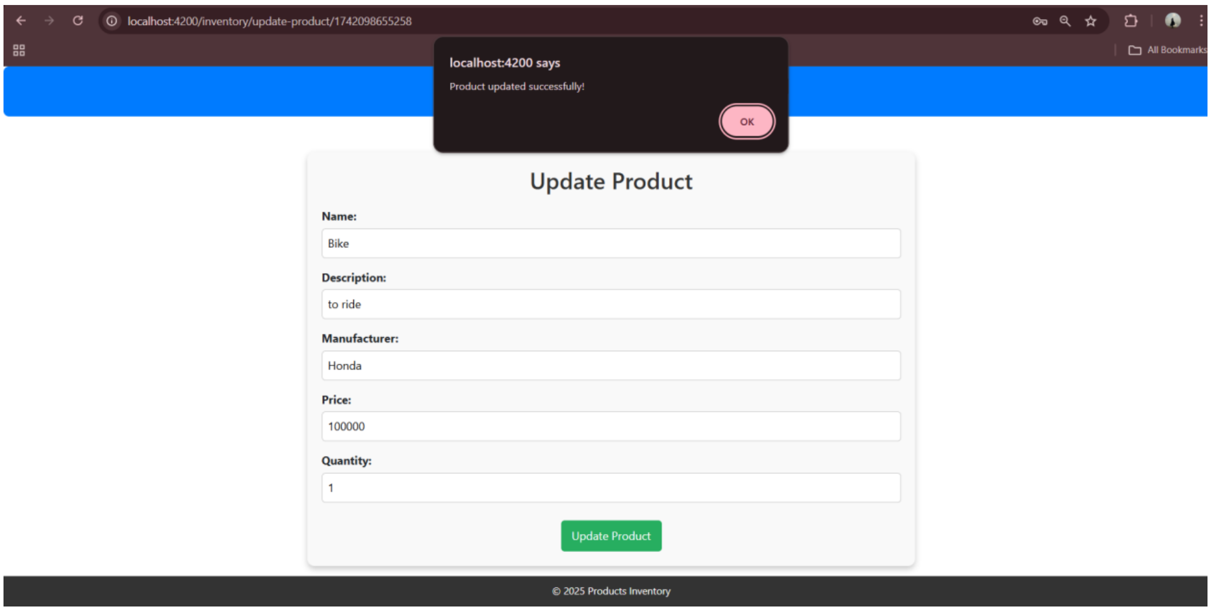
Manufacturer: Tata

Price: 200000

Quantity: 1

© 2025 Products Inventory

Product Update page :


localhost:4200/inventory/update-product/1742098655258

localhost:4200 says

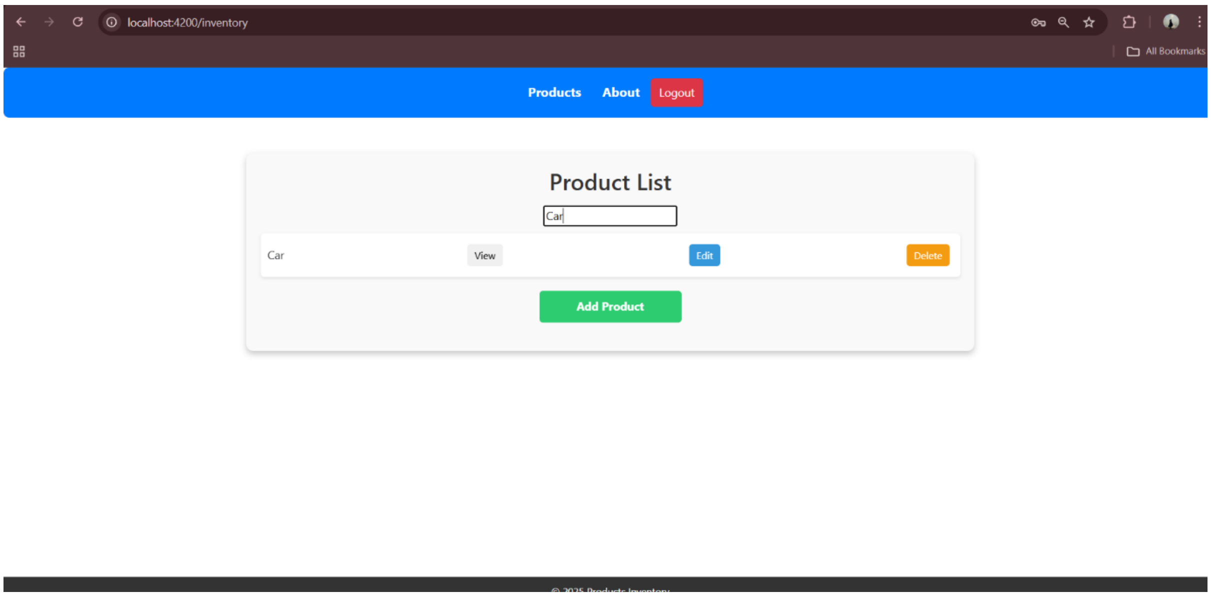
Product updated successfully!

Update Product

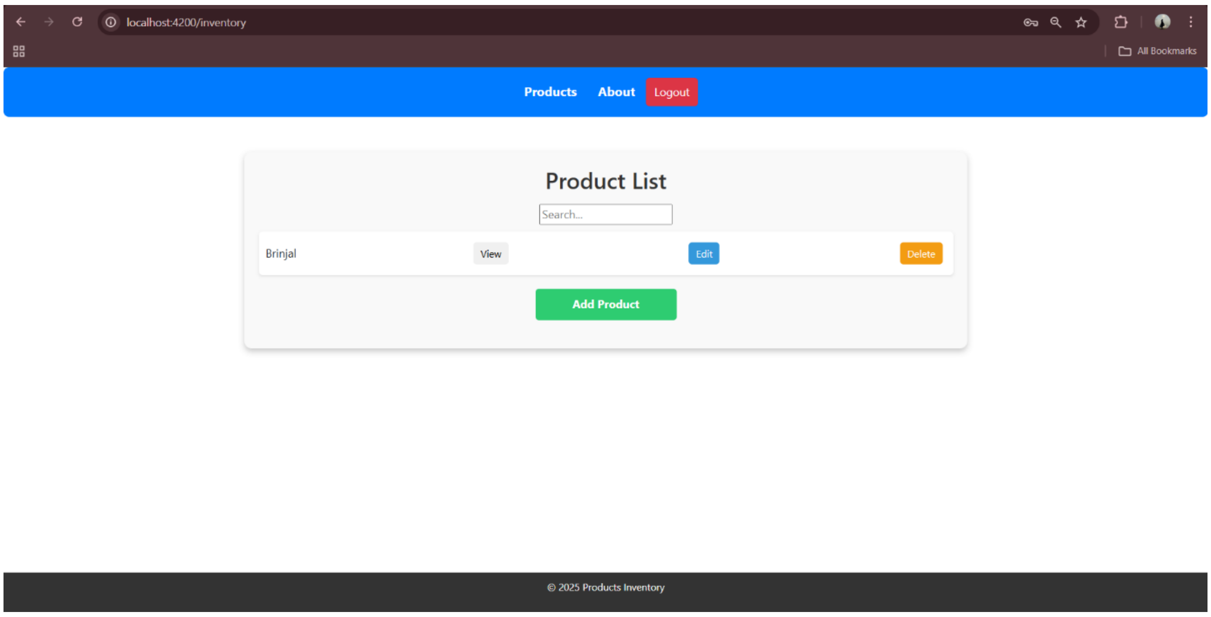
Name:	Bike
Description:	to ride
Manufacturer:	Honda
Price:	100000
Quantity:	1

© 2025 Products Inventory

Product Search :



After Product Deletion :



7. Conclusion

This project effectively implements:

- **User authentication & product management**.
- **Efficient data handling using Angular & JSON Server**.
- **Modular & reusable component structure**.
- **Robust error handling and validation mechanisms**.