## **1. 📌 Purpose**

This application is built using Angular 19 and NgRx for state management. It provides a basic authentication system (login/signup), displays a list of products from a fake API upon successful login, and protects routes using guards. The UI is styled using Bootstrap.

## **2. 🏗️ Architecture Overview**

### **2.1 Application Layers**

* **Presentation Layer** Angular components (login, signup, product list) for user interaction, built with Bootstrap for responsiveness.
* **Business Logic Layer** NgRx is used for managing authentication and product states using actions, reducers, selectors, and effects.
* **Data Layer** Services interact with APIs (e.g., fakestoreapi.com) and localStorage.

## **3. 📍 Key Features**

| **Feature** | **Description** |
| --- | --- |
| User Authentication | Login and Signup with local storage for session tracking. |
| Protected Routes | Guard prevents unauthenticated access to product list (home page). |
| State Management | NgRx used to handle authentication state and product data. |
| Product Listing | Displays products from https://fakestoreapi.com/products. |
| Bootstrap Styling | Bootstrap 5. |

## **4. 🔐 Authentication Flow**

* User lands on the **Login** page by default.
* If new, the user can navigate to the **Signup** page.
* On login:  
  + Credentials are checked via service/fake backend.
  + On success, user data/token stored in localStorage.
  + User is redirected to **Home** page (products).
* On logout:  
  + Session is cleared and redirected back to login.

## **5. 🔄 State Management (NgRx)**

* **Auth State**
  + Holds user info, token, and authentication status.
  + Actions: login, loginSuccess, loginFailure, logout, signup.
* **Product State**
  + Holds a list of products.
  + Actions: loadProducts, loadProductsSuccess, loadProductsFailure.
* **Effects**
  + Side effects like API calls and navigation handled in AuthEffects and ProductEffects.

## **6. 🔀 Routing Overview**

| **Path** | **Component** | **Guarded?** | **Description** |
| --- | --- | --- | --- |
| / | Redirect to login | No | Default route |
| /login | LoginComponent | No | Login form |
| /signup | SignupComponent | No | Signup form |
| /home | ProductListComponent | ✅ Yes | Protected home/product page |

## **7. 🧱 UI/UX Design**

* Clean layout using Bootstrap containers and cards.
* Form validation feedback using Bootstrap alerts.
* Product listing in bootstrap layout.

## **8. 📦 External Dependencies**

| **Package** | **Usage** |
| --- | --- |
| @ngrx/store | Centralized state |
| @ngrx/effects | Handle side effects |
| @ngrx/store-devtools | Debugging via Redux DevTools |
| Bootstrap | Styling UI components |
| RxJS | Reactive programming |

## **9. 🔧 Environment & Tools**

* **Language**: TypeScript
* **Framework**: Angular 19
* **State Management**: NgRx 16
* **UI Framework**: Bootstrap 5
* **API**: [Fake Store API](https://fakestoreapi.com/)
* **Dev Tools**: Angular CLI, Store DevTools

## **10. 🧪 Testing & Debugging**

* Use StoreDevToolsModule to monitor NgRx actions/state.
* Manual testing for login/signup and route protection.
* Validate API failure handling for product loading.

## **11. 🚀 Future Enhancements**

* Add products
* Unit tests for effects and reducers,components