**OBJECTIVE**

To build an UI/UX for online shopping

**Users of system**

1.Private companies

2. Public

3.Entrepreneur

**Functional requirement**

**\***built an application that public can access view the online shopping status .

\*the application should have sign up , log in , dashboard pages and individual records

1.state name

2.confirmation

3.price

4.address personal information

5. last update

\*this application should have provision to maintain a data base for an individual information.

\*filter like low to high price comparison .

\*app that provide less cost and high cost comparison facility .

\*app that support service used to respond specifically for users .

**NON-FUNCTIONAL REQUIREMENTS:**

Security

 The system shall enforce secure communication using encryption techniques.

 The system shall protect user data, including personal information and payment details, using secure storage methods.

STANDARD FEATURES:

1. Detailed Product Description

2. Product Customisation

3.Cross-selling

4.Promotions

5.Social marketing

6.Payment system

7.Integrated shipping solution

8.Speed and storage

9.Third-party Integration

10.Ready to use templates

11.Digital products

12.Search Engine Friendly

13.Mobile friendly

14.Reports and Analytics

15.Rating and Reviews

LOGGING:

 The system should support logging in web

Browser Compatible:

 All latest browsers

**OUTPUT /POST CONDITION**

\*Records that what you have purchased .

\*standard application / deployed in app container .

**NON FUNCTIONAL REQUIRMENT**

\*app must have publicly accesscible privacy policy .

**TECHNOLOGY STACK**

1.HTML, CSS

2.JAVASCRIPT

**KEYPOINT**

1.   The id (for frontend) and attributes(backend) mentioned in the SRS should not be

modified at any cost. Failing to do may fail test cases.

2.   Remember to check the screenshots provided with the SRS. Strictly adhere to id

mapping and attribute mapping. Failing to do may fail test cases.

3.   Strictly adhere to the proper project scaffolding (Folder structure), coding

conventions, method definitions and return types.

Adhere strictly to the endpoints given below.

**APPLICATION ASSUMPTIONS**

1.   The login page should be the first page rendered when the application loads.

2.   Manual routing should be restricted by using AuthGuard by implementing

the canActivate interface. For example, if the user enters

as http://localhost:8000/signup or http://localhost:8000/home the page should not

navigate to the corresponding page instead it should redirect to the login page.

3.   Unless logged into the system, the user cannot navigate to any other pages.

4.   Logging out must again redirect to the login page.

5.   To navigate to the admin side, you can store a user type as admin in the

database with a username and password as admin.

6.   Use admin/admin as the username and password to navigate to the admin

Dashboard

**PROJECT TASK**

\*privacy policy

\*for app that collect information in foreground services

\*for app that collect information running as background services

\*API requirement

\*editorial requirement

\*app review visibility

**USERS**

1. ***User Registration***
2. •The system shall allow users to create new accounts by providing their personal information.
3. •The system shall validate the user's email address and ensure uniqueness.
4. •The system shall generate a unique user ID for each registered user.
5. ***User Login***
6. •The system shall allow registered users to securely login using their email and password.
7. •The system shall validate the user's credentials before granting access.
8. ***Product Browsing***
9. •The system shall provide users with a search functionality to find products based on keywords.

•The system shall allow users to browse products by category and apply filters.

•The system shall display product details such as name, description, price, and availability.

1. ***Cart Management***
2. •The system shall allow users to add products to their cart.
3. •The system shall display the contents of the user's cart, including the total cost.
4. •The system shall allow users to remove items from their cart or update quantities.

***Checkout and Payment***

1. •The system shall provide a secure checkout process for users to review their order details.
2. •The system shall allow users to select a payment method and enter payment information.
3. •The system shall process payments and generate order confirmations
4. •.
5. ***Order Tracking***
6. •The system shall provide users with the ability to track the status of their orders.

***Customer Account Management***

•The system shall allow users to update their personal information, such as shipping address and contact details.

•The system shall allow users to view their order history and track the status of previous orders.

***Inventory Management***

•The system shall maintain an accurate inventory of products.

•The system shall update the product availability status based on purchases and returns. 4. Non-Functional Requirements

1. ***Usability***
2. •The system shall provide a user-friendly and intuitive interface.
3. •The system shall display clear and descriptive error messages
4. ***Performance***
5. •The system shall provide fast response times for product searches and page loads.
6. •The system shall be able to handle a large number of concurrent users without significant degradation in performance.
7. The system shall enforce secure communication using encryption techniques.
8. The system shall protect user data, including personal information and payment details, using secure storage methods.