ONLINE SHOPPING

Submitted by:

ADITHYA BABA B

DHARSHINI S

DIVYADHARSHNI G

Objective:

Shopaholic is an online application which is used to shop from online all over the country.

Users of the system:

1. Admin

2. Customer

Functional Requirements:

1.Build an application that people can access and shop through online

2. The application should have a signup, login, profile, dashboard

page, and search page

3. This page should have provision to maintain a database of:

→offers related to the products

→ delivery detailes

#delivery address

#payment details

→statistics

#number of products available

#discounts

variety of brands

price

☐ An integrated platform required for admin and people.

☐ Administration module to include options for adding /

modifying the people information(s). .

Non-Functional Requirements:

Security

☐ App Platform – Username / Password – Based Credentials.

☐ Sensitive data has to be categorized and stored in a secure

manner.

☐ Secure connections for transmissions of any data

Performance

☐ Peak Load performance

☐ eCommerce -< 3 sec

☐ Admin application < 2 sec

☐ Non-Peak Load Performance

☐ eCommerce <2 sec

☐ Admin Applications < 2 sec

Availability

☐ 99.99% Available

Standard Features

☐ Scalability

☐ Maintainability

☐ Usability

☐ Availability

☐ Failover

Logging and Auditing

☐ The system should support logging (app/web/DB) and auditing

at all levels

Monitoring

☐ Should be able to monitor via as-in enterprise monitoring tools

Cloud:

☐ The Solutions should be made Cloud-ready and should have a

minimum impact when moving away to Cloud infrastructure

Browser Compatible:

All Latest Browsers

Technology stack:

Front End Angular 10+/ React 16+

Material Design Bootstrap / Bulma

Server Side Spring Boot / .Net WebAPI/ Node Js

Database MySQL or Oracle or MSSQL

Applications 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

Validations:

1. Basic email validation should be performed.

2. Basic mobile validation should be performed.

Password validations should be performed.

Project Tasks:

API Endpoints:

USER:

Action URL Method Response:

Login :login POST true/false

Signup :signup POST true/false

Sort –by: Home /home GET Array of sort-by method

Placing order : Home /home GET Array of placing order model

Mode of pay model:POST statistics of confirmed order

Lay out of product: POST statistics of confirmed order

ADMIN:

Action URL Method Response:

Sort-by/admin GET Array of sort-by

Placing order/admin GET Array of placing order

Mode of pay /admin/ } POST mode of pay

Lay out of product /admin/POST lay out of the product

FRONTEND:

CUSTOMER:

1. Auth: The customer can authenticate login and signup credential.

2. Register: The new customer has options to sign up by providing

their basic details.

a. ids:

☐ Email

☐ Username

☐ Password

☐ Re-typepassword

☐ RegisterButton

☐ SigninLink

☐ RegisterBox

OUTPUT SCREENSHOT

SHOPAHOLIC

NEW ARRIVALS

Lorem ipsum dolor sit amet, consectetuer

₹ 2500

Lorem ipsum dolor sit amet, consectetuer

₹ 2500
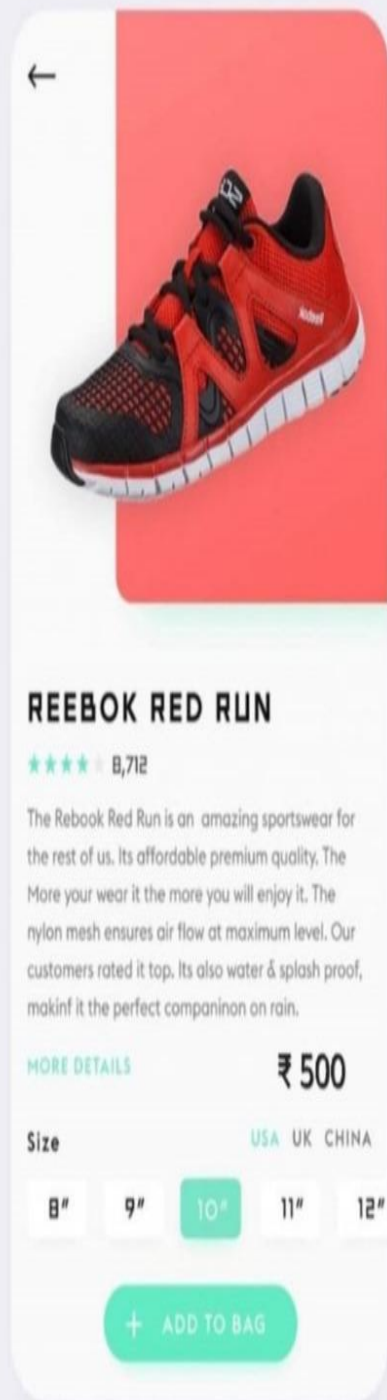
Cart

Product	Price	Quantity	Subtotal
✕ Animal Pillow	₹900	1	₹900
✕ Gold Headphones	₹1,200	1	₹2,300

Coupon code	APPLY COUPON		UPDATE CART

Cart totals

SUBTOTAL	₹3,200
TOTAL	₹3,200

PROCEED TO CHECKOUT

🔒

Your Information Is Safe

🛡️

Secure Checkout

📞

24/7 Customer Support

ADMIN:

ADMIN DASH BOARD:

This is the place where the multiple options for the users to access the application will be displayed

BACK END:

MODEL LAYER:

1.USER MODEL:

The place where all the information of the user or the customer are stored

a.Attributes:

*email:string

*username:string

*password:string

*location:string

*mobile number:string

2.**LOGIN MODEL**: This class contains the email and password of the user to access the application

A.attributes:

*email:string

*password:string

3.**SORT-BY MODEI**: This contains the sorting methods of the products that are been displayed In the sale page

A.attributes:

        *sort-by trend:character

        *sort-by cost:character

        *sort-by alphabet:character

4.PLACING ORDER MODEL :This contains the information of the product and the cost of the order placed

A.atttributes:

        *product name:character

        *product price:character

        *delivery location:string

        *delivery charge:character

        *total price:character

5.MODE OF PAY MODEL: this contains the information of the mode of pay for the product

A.attributes:

        *g-pay:character

        *cash on delivery:character

        *phone pay:character

        *bank-transfer:character

6.LAYOUT OF THE PRODUCT MODEL:this contains the complete detail of the product that has been placed order

A.attribues:

        *delivery date:character

*delivery location:character

*total cost:character

*order date:character

*dispatch date:character

*product track:character

Controller Layer:

1.Signup Controller: This class control the user signup

A. Methods:

i. (User Model user): This method helps to store

users in the database and return true or false based on the database

Transaction

2. Login Controller: This class controls the user login.

a. Methods:

i. Check User(Login Model data): This method helps the user

to sign up for the application and must return true or false.