Capstone Project

Shop For Home

Group Name- Java C3- Group 3

Group Members

J.S.KIREETI DEV
M.MANMADHA KUMAR
K.YASWANTH
B.S.VIJAY SANKAR
R.SANDESH

Contents:-

1. Introduction

Electronic commerce refers to the buying and selling of products or services over electronic systems such as the Internet and other computer networks. The Shopping Cart is very important feature used in e-commerce to assist people making purchases products online. It also includes the entire online process of developing, marketing, selling, delivering, servicing and paying for products and services. In order to purchase a shopping cart is provided to the user. The amount of trade conducted electronically has grown extraordinarily with widespread Internet usage. The use of commerce is conducted in this way, spurring and drawing on innovations in electronic funds transfer, supply chain management, Internet marketing, online transaction processing, electronic data interchange (EDI), inventory management systems, and automated data collection systems. The e-commerce site will let customers to view and order products online from any part of the world.

The main advantage of e-commerce over traditional commerce is the user can browse online shops, compare prices and order merchandise sitting at home on their PC. Secure registration and profile management facilities for Customers. Shopping Cart feature allows online shopping customers to “place” items in the cart. It Decreases the cost of creating, processing, distributing, storing and retrieving paper-based information. Expands the marketplace to national and international markets. Upon “checkout” the software calculates as total for the order including shipping and handling postage, packing and taxes, if applicable. Reduces the time between the outlay of capital and the receipt of products and services. Customers should be able to mail the Shop about the items they would like to see in the Shop

The proposed system helps in building a website to buy, sell products or goods online using internet connection. Enables consumers to shop or do other transactions 24 hours a day, all year round from almost any location. It can be accessed over the Internet.

Purchasing of goods online, user can choose different products based on categories , online payments , delivery services and hence covering the disadvantages of the existing system and making the buying easier and helping the vendors to reach wider market. It Provides consumers with more choices. Customer can purchase Products Online.

2. Scope

E-commerce has bloomed over the years and is one of the fastest-growing domains in the online world. Though it took some time for this to be accepted by the end-users, today we are at a point where the majority of the people love to shop online. There were numerous concerns revolving around online shopping at its launch, but over years people tend to have started trusting E-commerce for all their shopping needs.

In India, people prefer shopping online these days rather than having to visit the physical store. The payment features that are smart and secure as well as the cash on delivery (COD), which makes the payment, even more, safer with hassle-free shipping, easy returns and reach out.

Let us check out the development or growth of this e-commerce sector in India. We have specifically collected substantial data from across the web after analysis and inferences of information acquired from authentic sources. In the year 2013, around 8 million people have been shopping online. And, the most interesting factor is that they have done shopping from some of the major online shopping sites. And, the number 8 million had risen to around 100 million by the year 2016. The new shoppers (customer base) accounting to around 50% came from the tier one and tier two towns of India. Today, we can proudly say that India is one of the places where online shopping has been booming and will continue to do so. This means that online shopping has a lot of prospects in the future.

3. Purpose

The primary goal of e-commerce is **to reach maximum customers at the right time to increase sales and profitability of the business**. Functions of e-commerce include buying and selling goods, transmitting funds or data over the internet.

A typical online store **enables the customer to browse the firm's range of products and services, view photos or images of the products, along with information about the product specifications, features and prices**. Online stores usually enable shoppers to use "search" features to find specific models, brands or items.

4. Technology Stack

1. **Angular**

2. **Java(Spring Boot)**

3. **Tomcat**

4. **MySql**

5. Software Requirements

Front End :-

Front end is build using Angular.

Back End :-

Java(Spring Boot Microservice) is used for Back-end.

Webserver :-

Tomcat is used for web server.

Database :-

MySql is used for database.

6. Design Goals

- The design of the web application involves the design of the forms for listing the products, search for products, display the complete specification for the product, and design a shopping cart that is easy to use.

- Design of an interactive application that enables the user to filter the products based on different parameters.

- Design of an application that has features like drag and drop etc.

- Design of application that decreases data transfers between the client and the server.

7. Shop Products Module

This module starts when the user visits the home page or when a user searches for a product by entering a search term. This part of the application includes displaying all the products that are available or the products that match the search term entered by the user. The user can then filter these products based on various parameters like manufacturer, product type, operating system supported or a price range. The user browse through the products and each product would be displayed with an image and its features like operating system supported, number of user licenses and if it is a full version or an upgrade version. A user can add a product to the cart either by dragging the product and dropping it in the cart or by clicking a button. The user would be able to see the shopping cart summary.

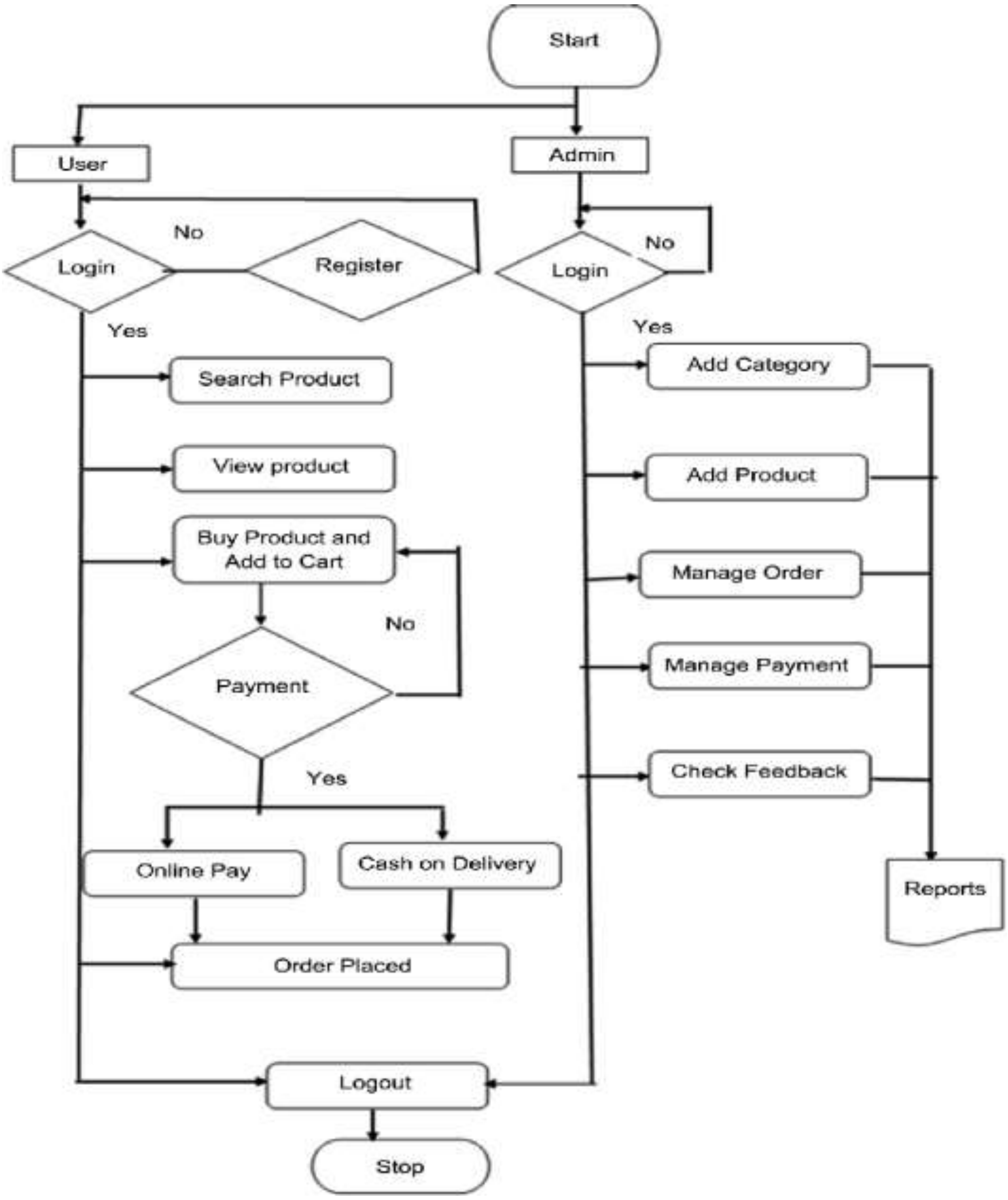
8. Product Description Module

This module starts when a user visits the product description page. A user can view various images of the product of different sizes. The use can see an enlarged image in a popup window. The user can view the complete specification of the product like its features, operating system supported, system requirements etc. A user can also view the manufacturer information and also information about rebates, exchange policies etc. A user can also view the reviews of the product. A user can also write a review for the product.

9. Shopping Cart Module

This module starts when the user views the shopping cart. All the products that have been added to the shopping cart by the user are listed along with their price and the quantity. The total price of all the products added to cart is displayed. A user can edit the quantity of each product or remove the product from the shopping cart. A user can remove the product from the cart by clicking a button or by dragging the product and dropping it outside the cart. The total price changes accordingly when a user edits the quantity of a product or when a product is removed from the cart.

10. FLOWCHART

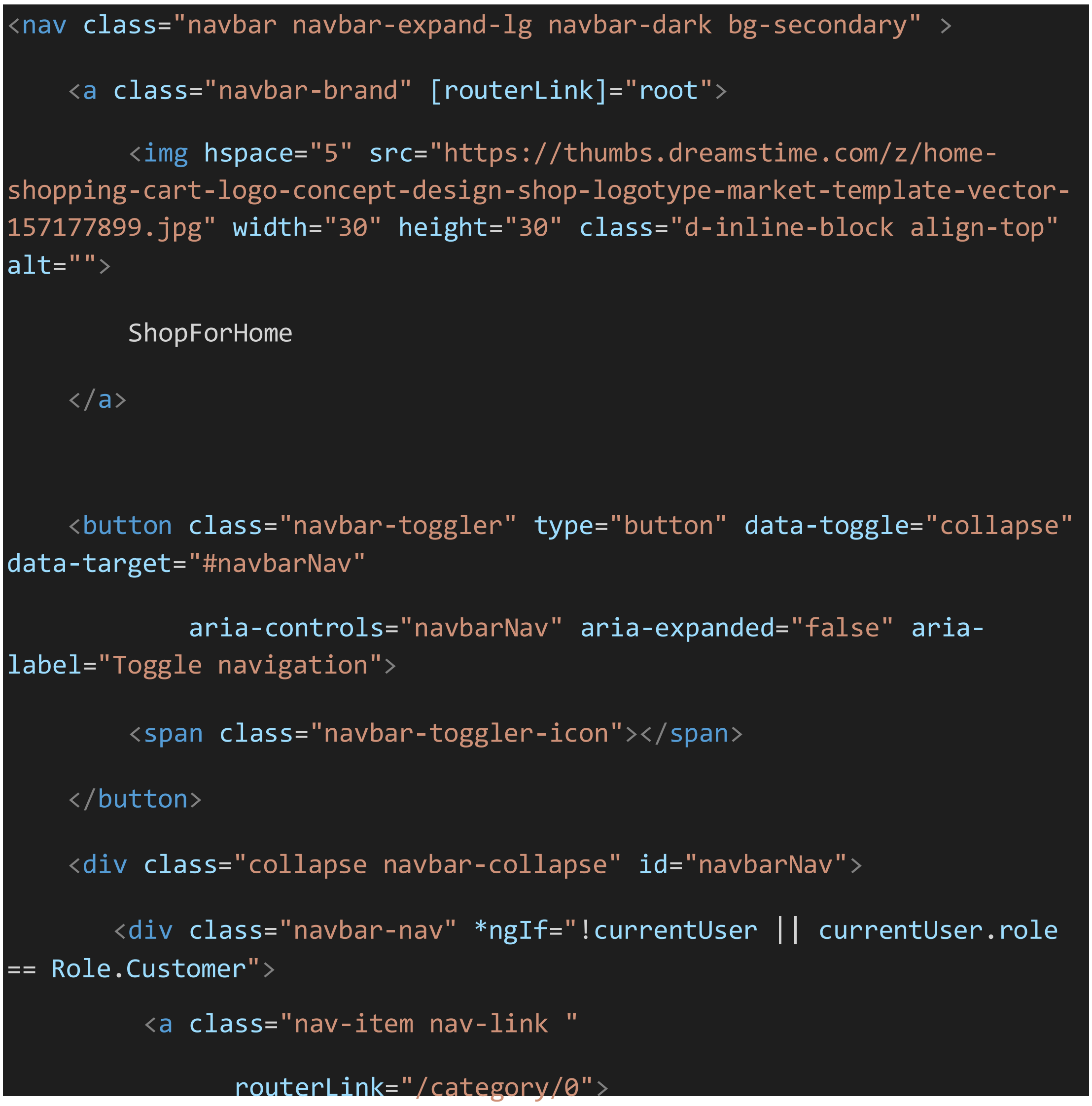


11. Results

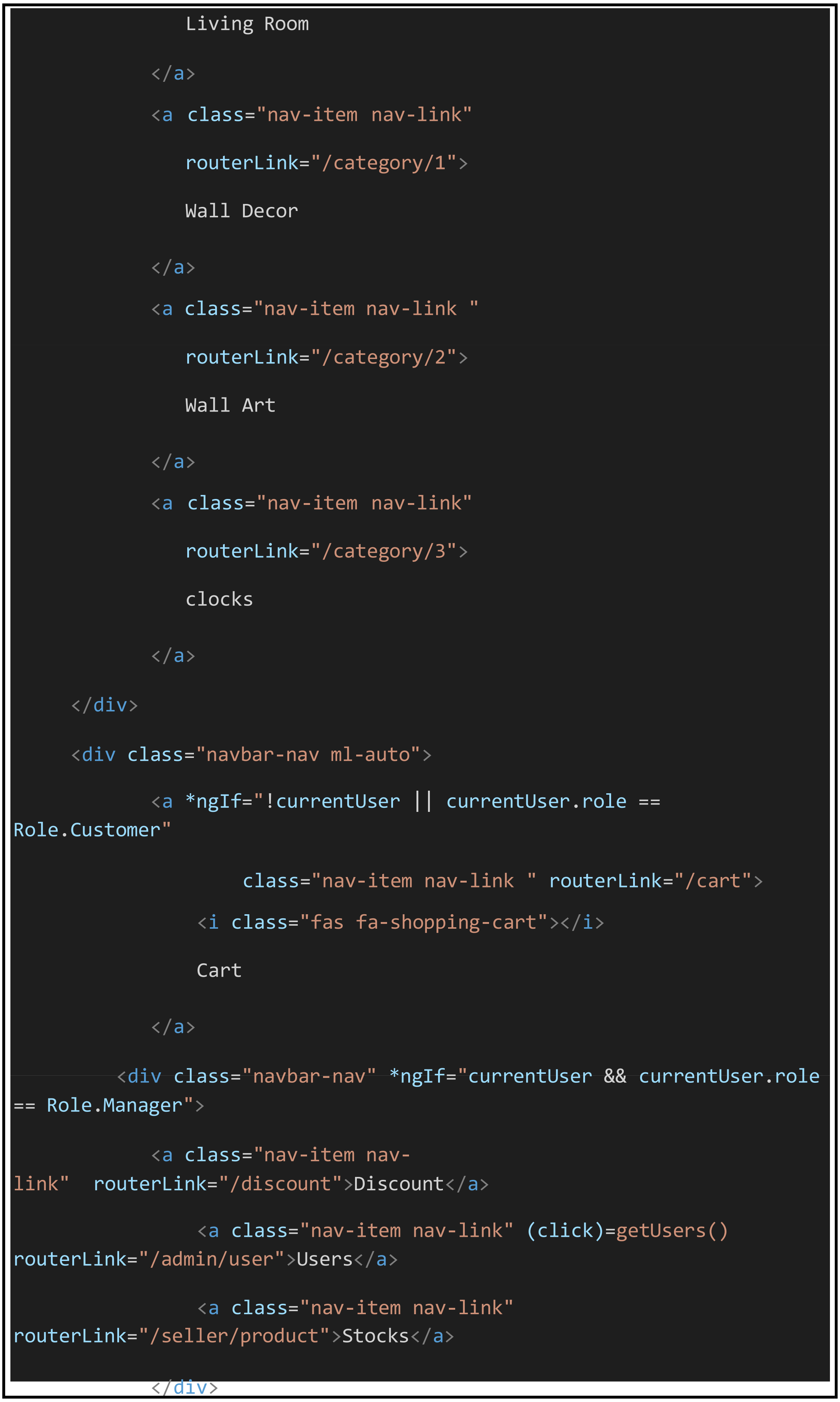
The application can be used for any Ecommerce application. It is easy to use. User friendly screens are provided. The application is easy to use and interactive making online shopping a recreational activity for users. It has been thoroughly tested and implemented.

Screenshots

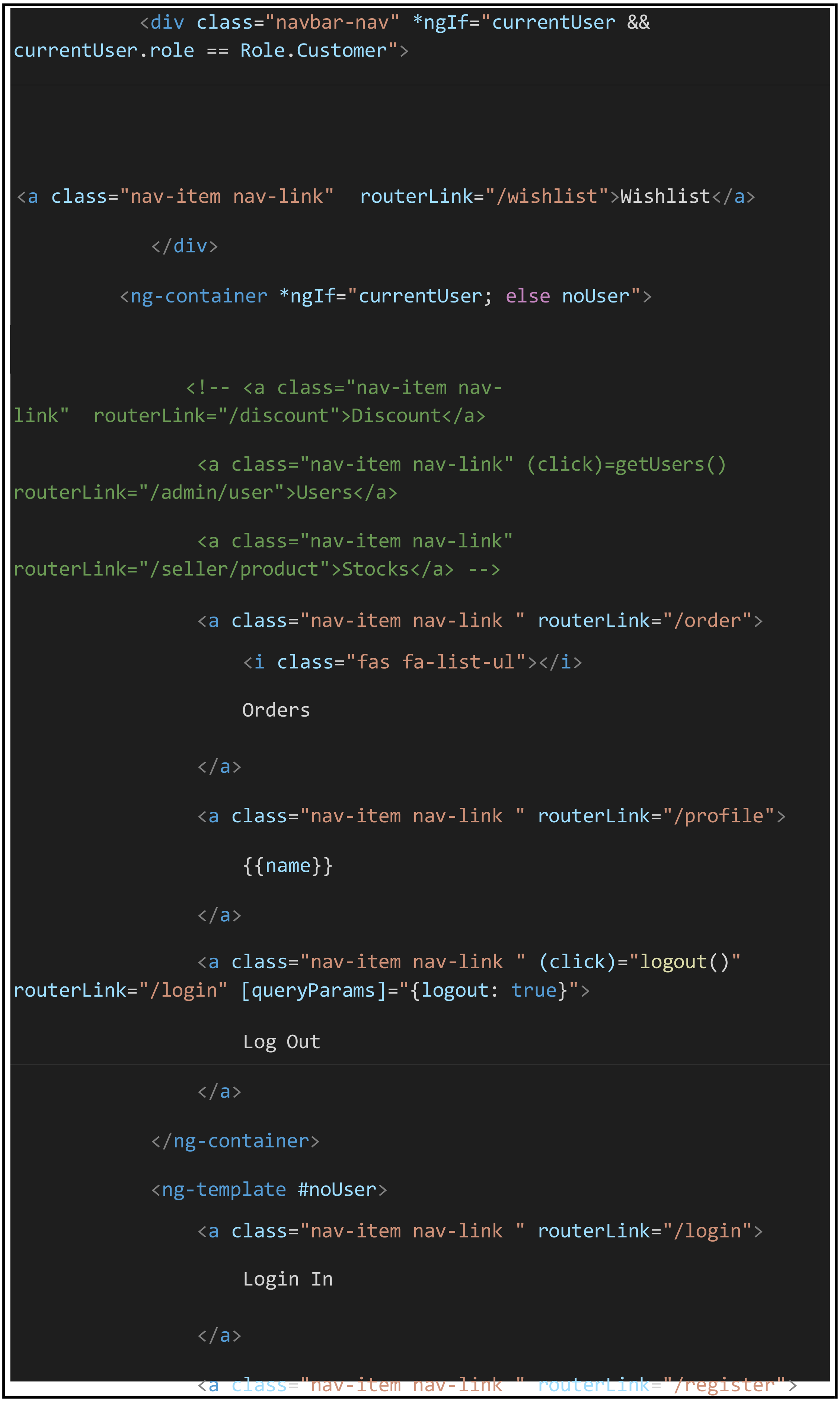
1. Home Source Code :-



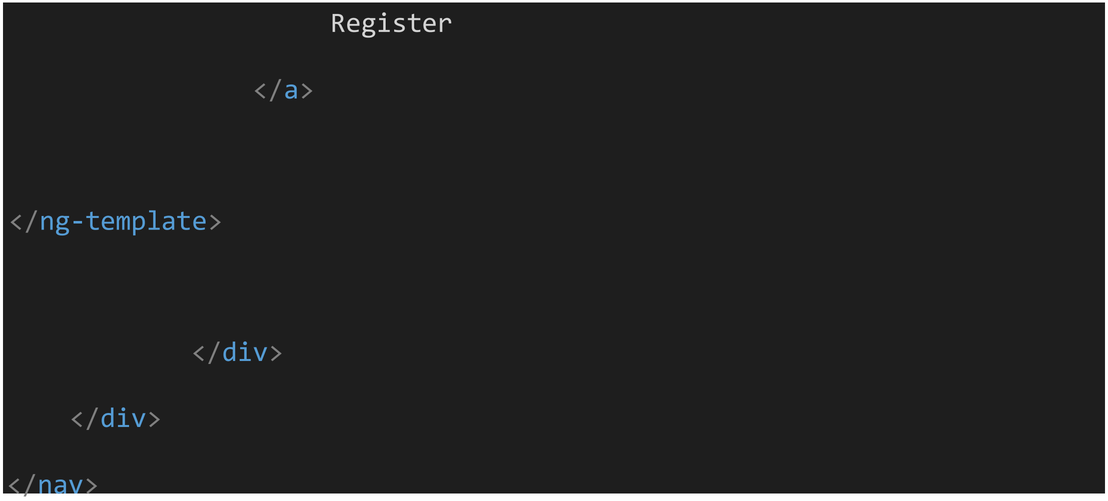
```
<nav class="navbar navbar-expand-lg navbar-dark bg-secondary" >

    <a class="navbar-brand" [routerLink]="root">

        <img hspace="5" src="https://thumbs.dreamstime.com/z/home-
shopping-cart-logo-concept-design-shop-logotype-market-template-vector-
157177899.jpg" width="30" height="30" class="d-inline-block align-top"
alt="">

        ShopForHome

    </a>


    <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarNav"

            aria-controls="navbarNav" aria-expanded="false" aria-
label="Toggle navigation">

        <span class="navbar-toggler-icon"></span>

    </button>

    <div class="collapse navbar-collapse" id="navbarNav">

       <div class="navbar-nav" *ngIf="!currentUser || currentUser.role
== Role.Customer">

         <a class="nav-item nav-link "

               routerLink="/category/0">
```



```
                Living Room

            </a>
            <a class="nav-item nav-link"

               routerLink="/category/1">

               Wall Decor

            </a>
            <a class="nav-item nav-link "

               routerLink="/category/2">

               Wall Art

            </a>
            <a class="nav-item nav-link"

               routerLink="/category/3">

               clocks

            </a>

     </div>

     <div class="navbar-nav ml-auto">

            <a *ngIf="!currentUser || currentUser.role ==
Role.Customer"

                    class="nav-item nav-link " routerLink="/cart">
                <i class="fas fa-shopping-cart"></i>

                Cart

            </a>

         <div class="navbar-nav" *ngIf="currentUser && currentUser.role
== Role.Manager">

            <a class="nav-item nav-
link"  routerLink="/discount">Discount</a>

                <a class="nav-item nav-link" (click)=getUsers()
routerLink="/admin/user">Users</a>

                <a class="nav-item nav-link"
routerLink="/seller/product">Stocks</a>

            </div>
```

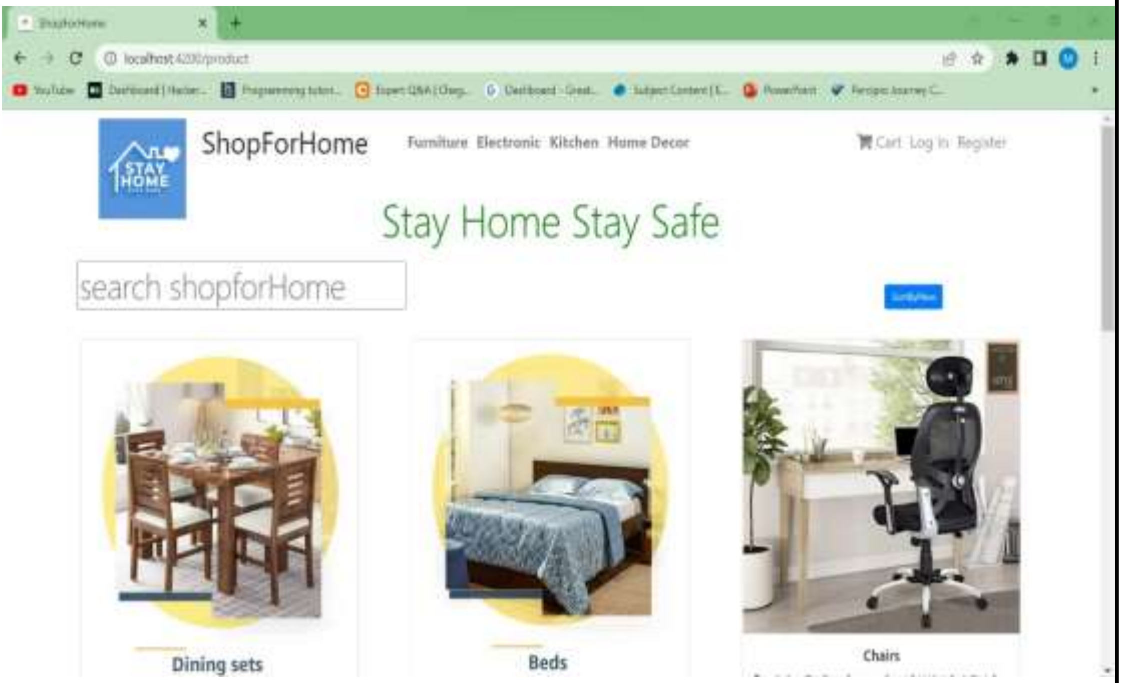


```
           <div class="navbar-nav" *ngIf="currentUser &&
currentUser.role == Role.Customer">



<a class="nav-item nav-link"  routerLink="/wishlist">Wishlist</a>

            </div>

         <ng-container *ngIf="currentUser; else noUser">


               <!-- <a class="nav-item nav-
link"  routerLink="/discount">Discount</a>

                <a class="nav-item nav-link" (click)=getUsers()
routerLink="/admin/user">Users</a>

                <a class="nav-item nav-link"
routerLink="/seller/product">Stocks</a> -->

                <a class="nav-item nav-link " routerLink="/order">

                    <i class="fas fa-list-ul"></i>

                    Orders

                </a>

                <a class="nav-item nav-link " routerLink="/profile">

                    {{name}}

                </a>

                <a class="nav-item nav-link " (click)="logout()"
routerLink="/login" [queryParams]="{logout: true}">

                    Log Out

                </a>

            </ng-container>

            <ng-template #noUser>

                <a class="nav-item nav-link " routerLink="/login">

                    Login In

                </a>

                <a class="nav-item nav-link " routerLink="/register">
```

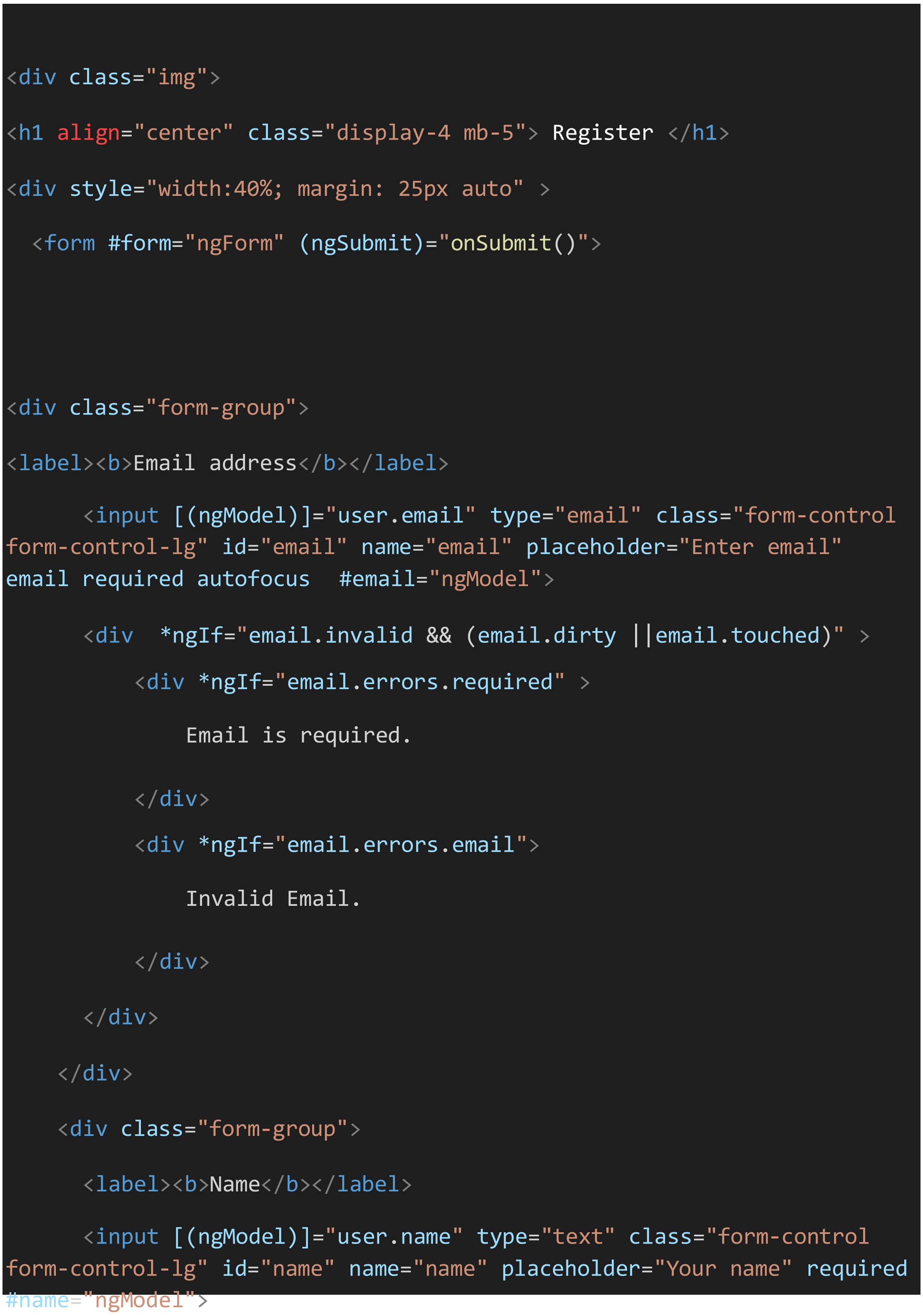


```
                     Register

                </a>


</ng-template>


            </div>

    </div>

</nav>
```

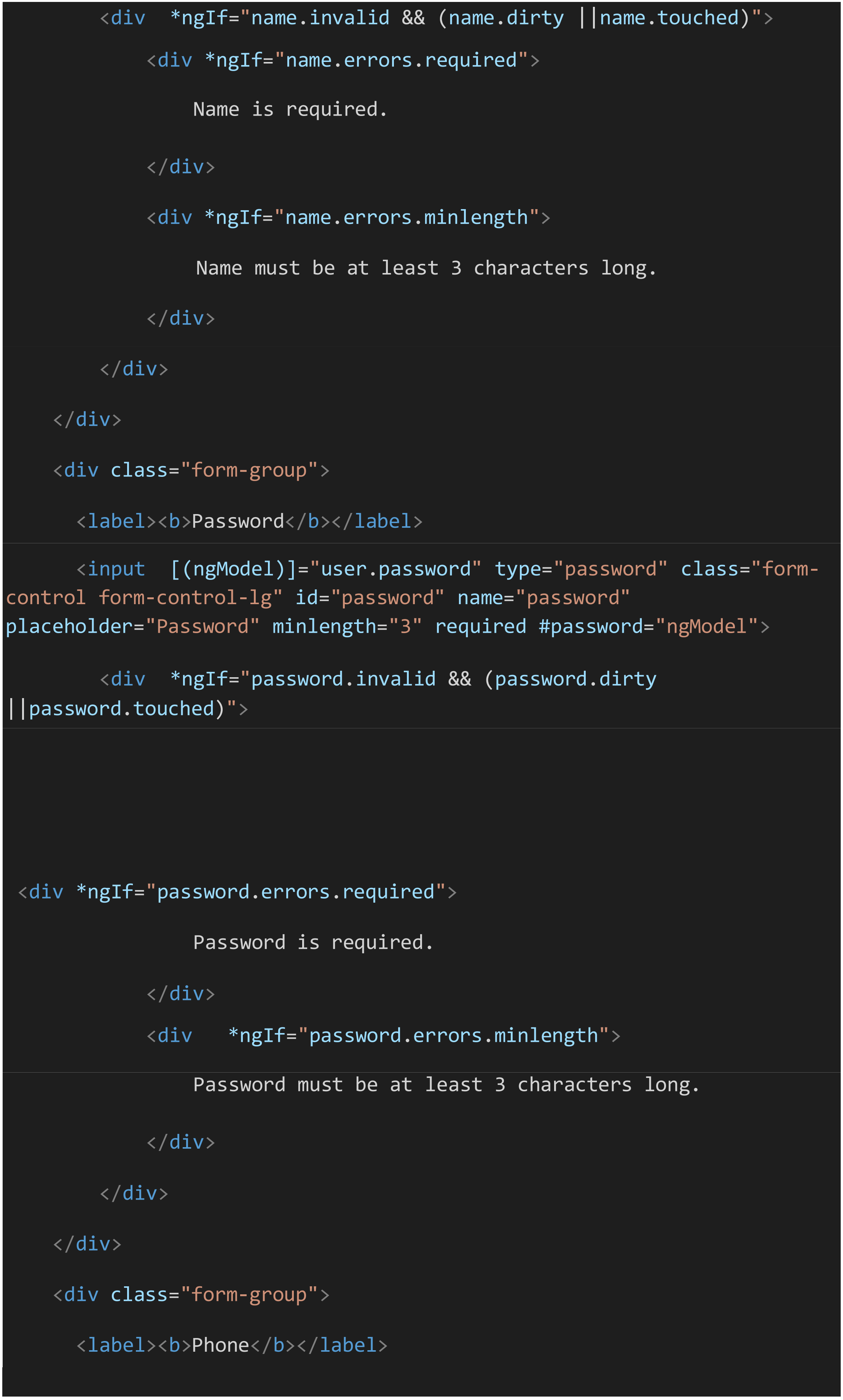
Output :-



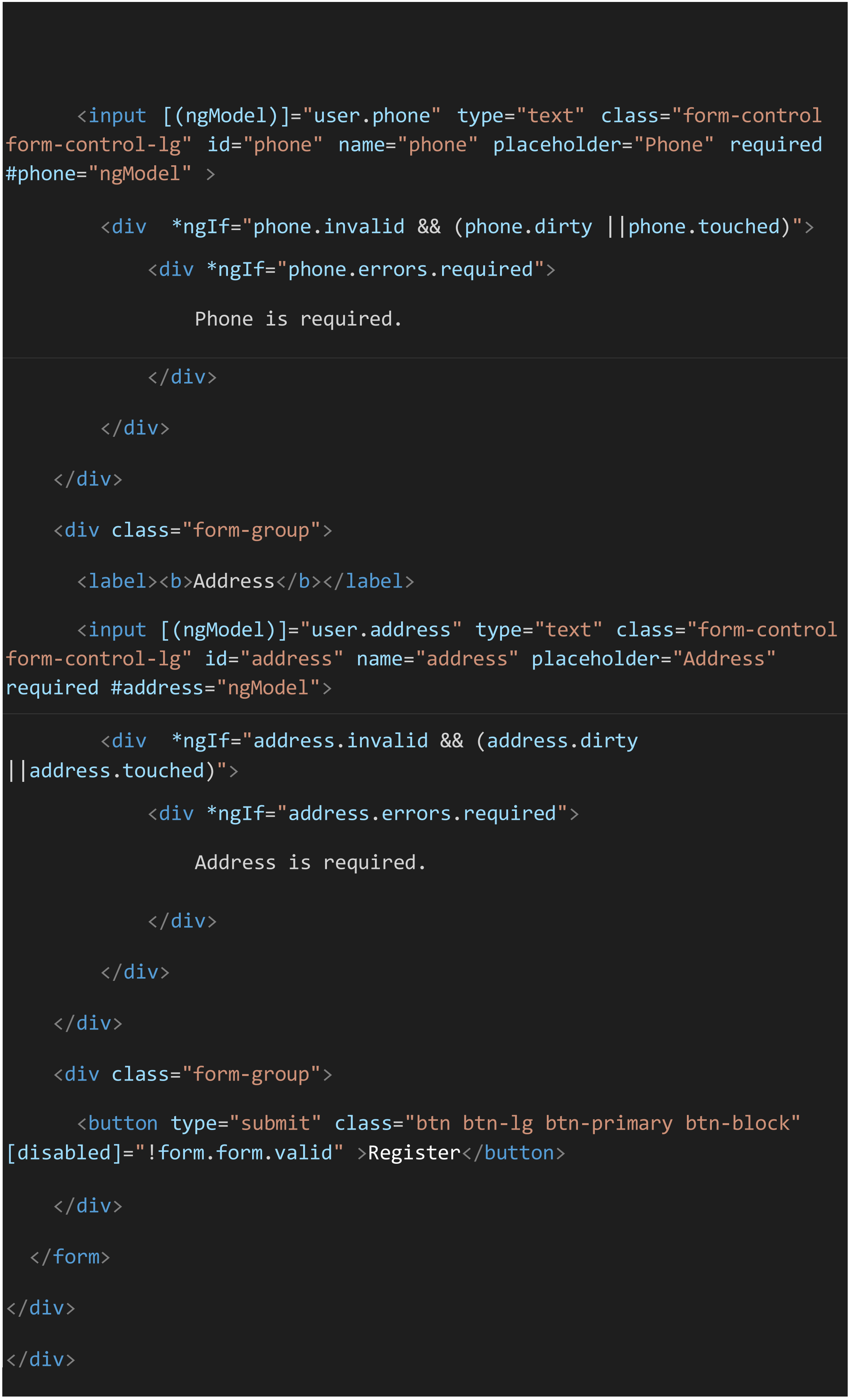
2. <u>REGISTER SOURCE CODE</u> :-



```
<div class="img">

<h1 align="center" class="display-4 mb-5"> Register </h1>

<div style="width:40%; margin: 25px auto" >

  <form #form="ngForm" (ngSubmit)="onSubmit()">




<div class="form-group">

<label><b>Email address</b></label>

      <input [(ngModel)]="user.email" type="email" class="form-control
form-control-lg" id="email" name="email" placeholder="Enter email"
email required autofocus  #email="ngModel">

      <div  *ngIf="email.invalid && (email.dirty ||email.touched)" >

          <div *ngIf="email.errors.required" >

              Email is required.

          </div>

          <div *ngIf="email.errors.email">

              Invalid Email.

          </div>

      </div>

    </div>

    <div class="form-group">

      <label><b>Name</b></label>

      <input [(ngModel)]="user.name" type="text" class="form-control
form-control-lg" id="name" name="name" placeholder="Your name" required
#name="ngModel">
```

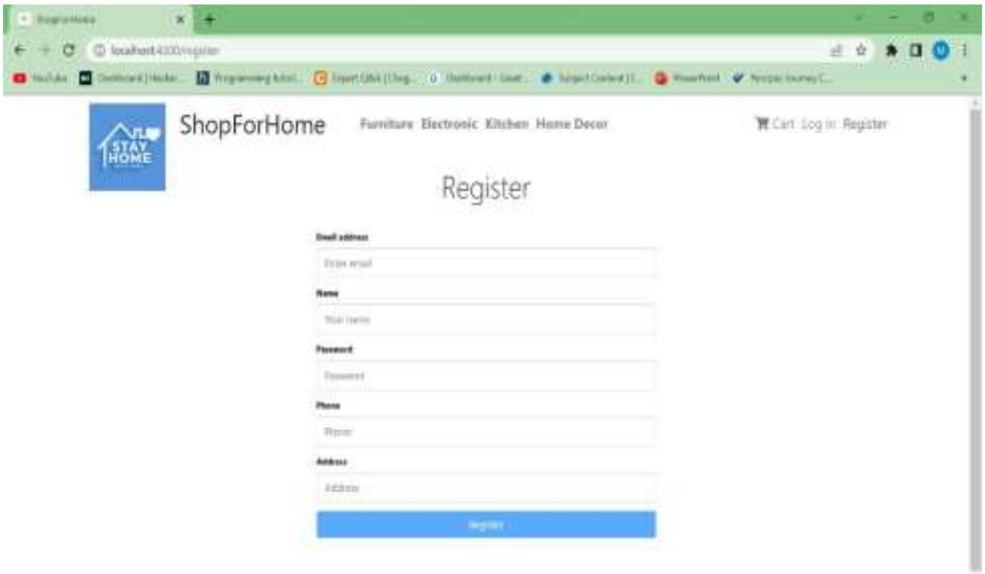


```
        <div  *ngIf="name.invalid && (name.dirty ||name.touched)">
            <div *ngIf="name.errors.required">
                Name is required.
            </div>
            <div *ngIf="name.errors.minlength">
                Name must be at least 3 characters long.
            </div>
        </div>
    </div>
    <div class="form-group">
      <label><b>Password</b></label>
      <input  [(ngModel)]="user.password" type="password" class="form-
control form-control-lg" id="password" name="password"
placeholder="Password" minlength="3" required #password="ngModel">
        <div  *ngIf="password.invalid && (password.dirty
||password.touched)">




 <div *ngIf="password.errors.required">
                Password is required.
            </div>
            <div   *ngIf="password.errors.minlength">
                Password must be at least 3 characters long.
            </div>
        </div>
    </div>
    <div class="form-group">
      <label><b>Phone</b></label>
```



```
      <input [(ngModel)]="user.phone" type="text" class="form-control
form-control-lg" id="phone" name="phone" placeholder="Phone" required
#phone="ngModel" >
        <div  *ngIf="phone.invalid && (phone.dirty ||phone.touched)">
            <div *ngIf="phone.errors.required">
                Phone is required.
            </div>
        </div>
    </div>
    <div class="form-group">
      <label><b>Address</b></label>
      <input [(ngModel)]="user.address" type="text" class="form-control
form-control-lg" id="address" name="address" placeholder="Address"
required #address="ngModel">
        <div  *ngIf="address.invalid && (address.dirty
||address.touched)">
            <div *ngIf="address.errors.required">
                Address is required.
            </div>
        </div>
    </div>
    <div class="form-group">
      <button type="submit" class="btn btn-lg btn-primary btn-block"
[disabled]="!form.form.valid" >Register</button>
    </div>
  </form>
</div>
</div>
```

OUTPUT :-



Register

Email address

Enter email

Name

Your name

Password

Password

Phone

Phone

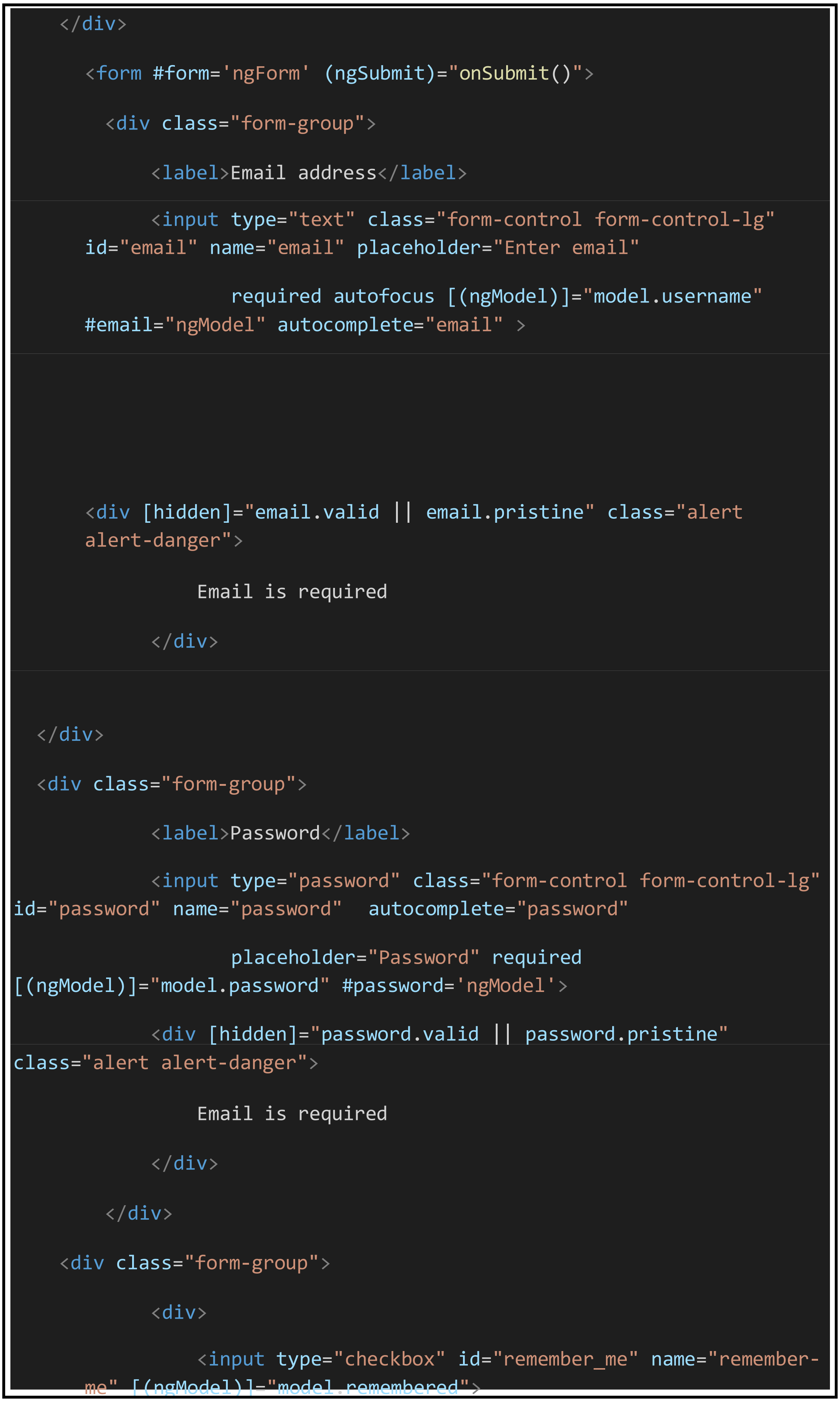
Address

Address

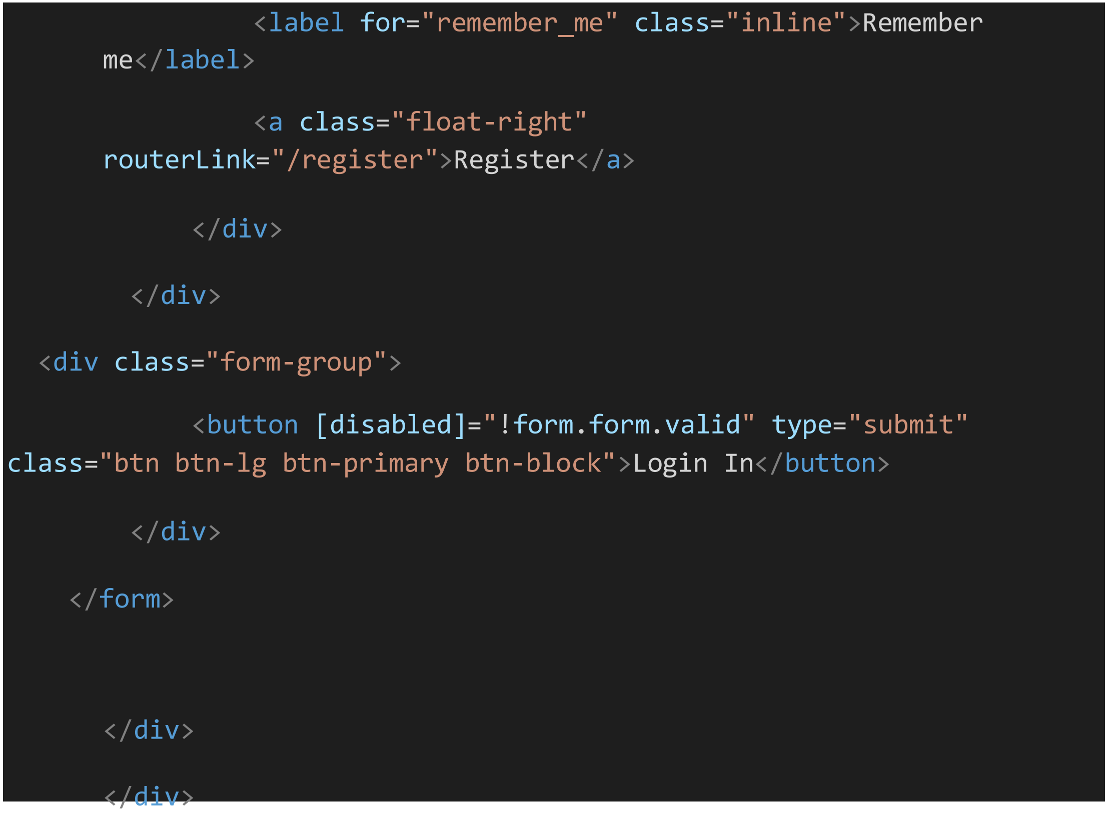
Register

LOGIN SOURCE CODE :-

```
<div class="img">

<h1 align="center" class="display-4 mb-5">Login</h1>

<div style="width:40%; margin: 25px auto">


    <div class="alert alert-danger" *ngIf="isInvalid">

        Invalid username and password.

    </div>

    <div class="alert alert-info" *ngIf="isLogout">

        You have been logged out.
```

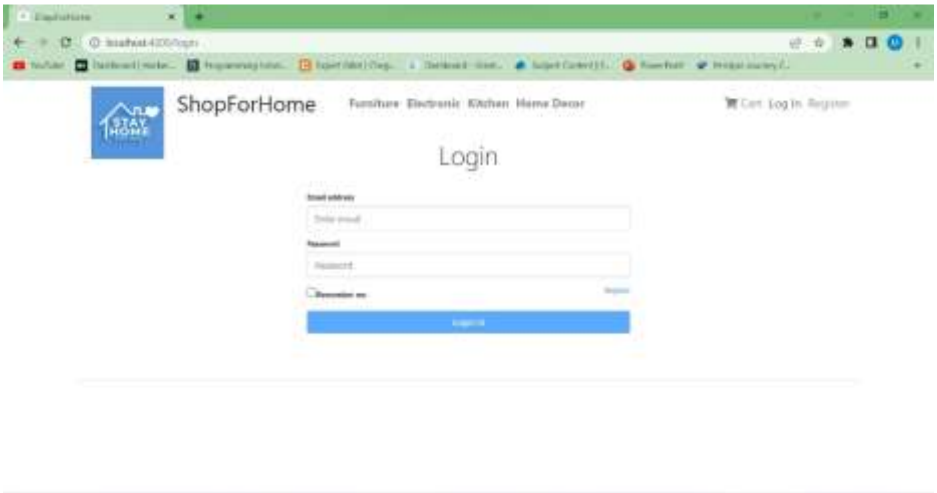


```
    </div>

      <form #form='ngForm' (ngSubmit)="onSubmit()">

        <div class="form-group">

            <label>Email address</label>

            <input type="text" class="form-control form-control-lg"
      id="email" name="email" placeholder="Enter email"

                   required autofocus [(ngModel)]="model.username"
      #email="ngModel" autocomplete="email" >




      <div [hidden]="email.valid || email.pristine" class="alert
      alert-danger">

                Email is required

            </div>


  </div>

  <div class="form-group">

            <label>Password</label>

            <input type="password" class="form-control form-control-lg"
id="password" name="password"  autocomplete="password"

                   placeholder="Password" required
[(ngModel)]="model.password" #password='ngModel'>

            <div [hidden]="password.valid || password.pristine"
class="alert alert-danger">

                Email is required

            </div>

        </div>

    <div class="form-group">

            <div>

                <input type="checkbox" id="remember_me" name="remember-
      me" [(ngModel)]="model.remembered">
```



```
                <label for="remember_me" class="inline">Remember
      me</label>

                <a class="float-right"
      routerLink="/register">Register</a>

            </div>

        </div>

  <div class="form-group">

            <button [disabled]="!form.form.valid" type="submit"
class="btn btn-lg btn-primary btn-block">Login In</button>

        </div>

    </form>



      </div>

      </div>
```

OUTPUT :-



ShopForHome

Furniture Electronic Kitchen Home Decor

Cart Log In Register

Login

Email address

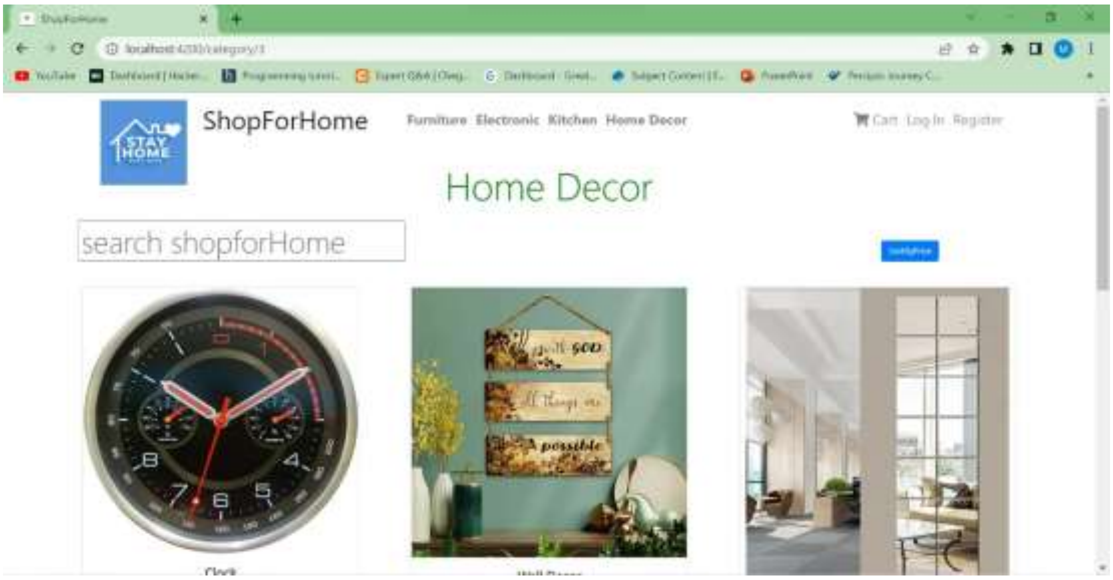
Enter email

Password

Password

□ Remember me

CATEGORIES :-



localhost:4200/category/3

ShopForHome

Furniture Electronic Kitchen Home Decor

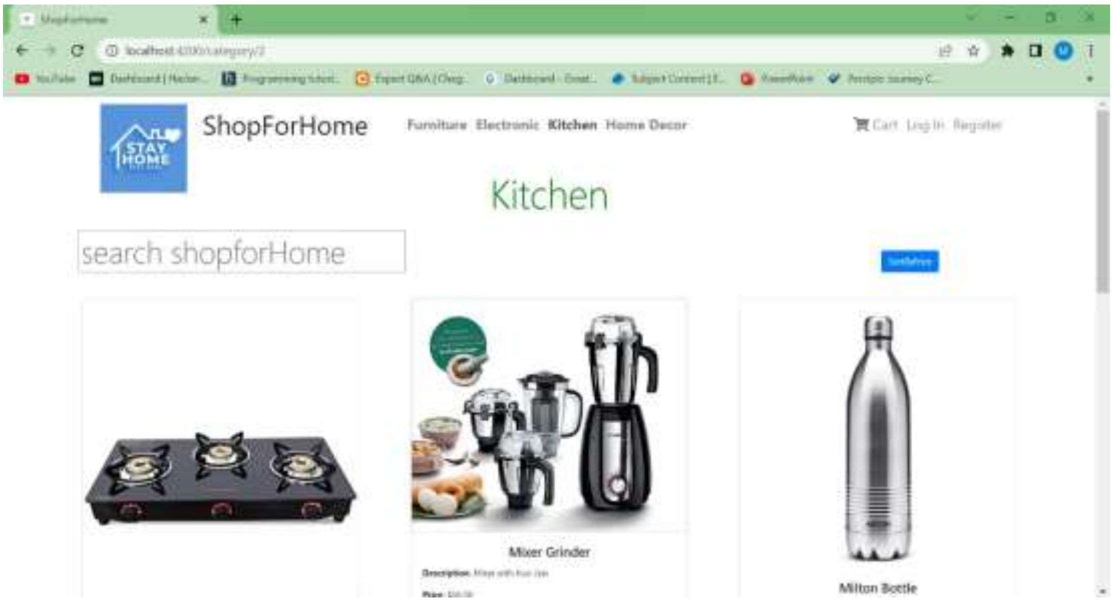
Cart Log In Register

Home Decor

search shopforHome

Clock

Wall Decor



localhost:4200/category/2

ShopForHome

Furniture Electronic Kitchen Home Decor

Cart Log In Register

Kitchen

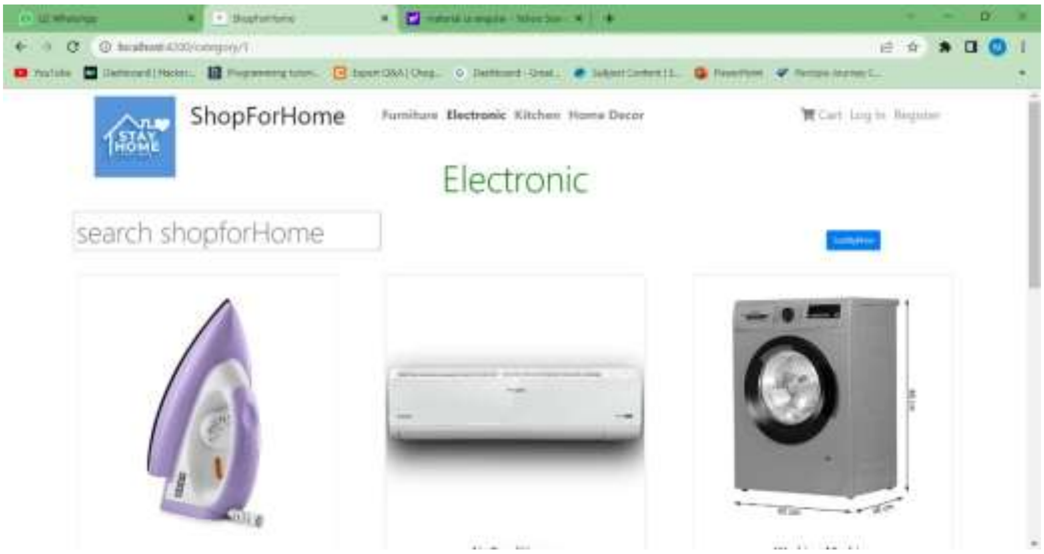
search shopforHome

Mixer Grinder

Description:

Price:

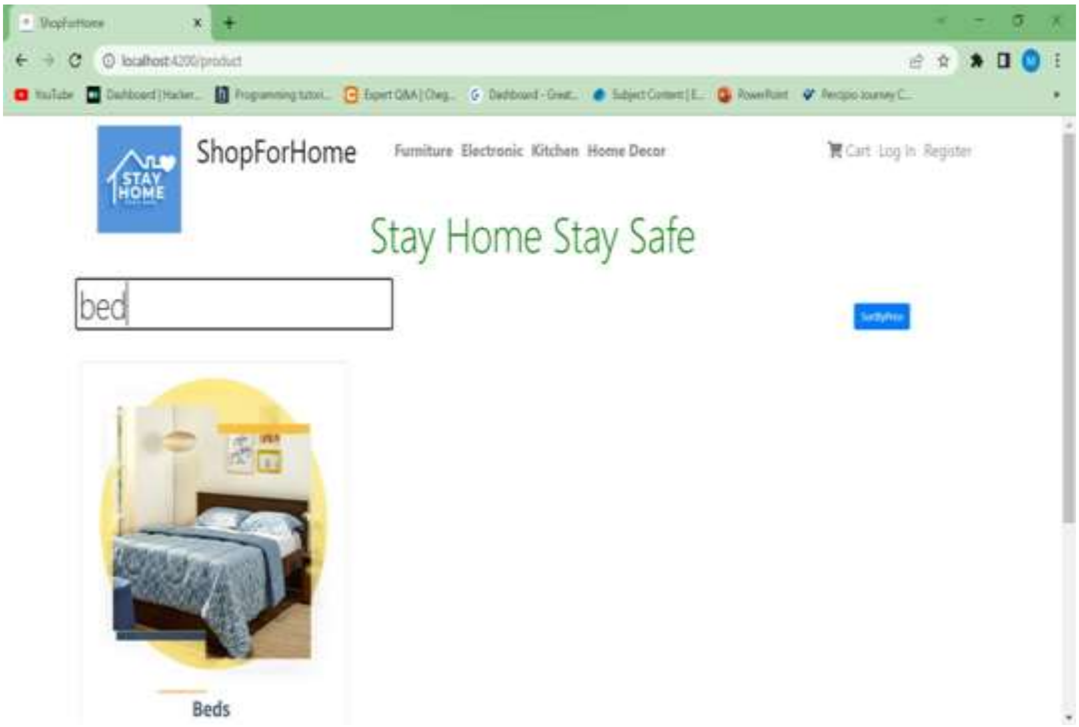
Milton Bottle



Electronic

search shopforHome

SEARCH :-

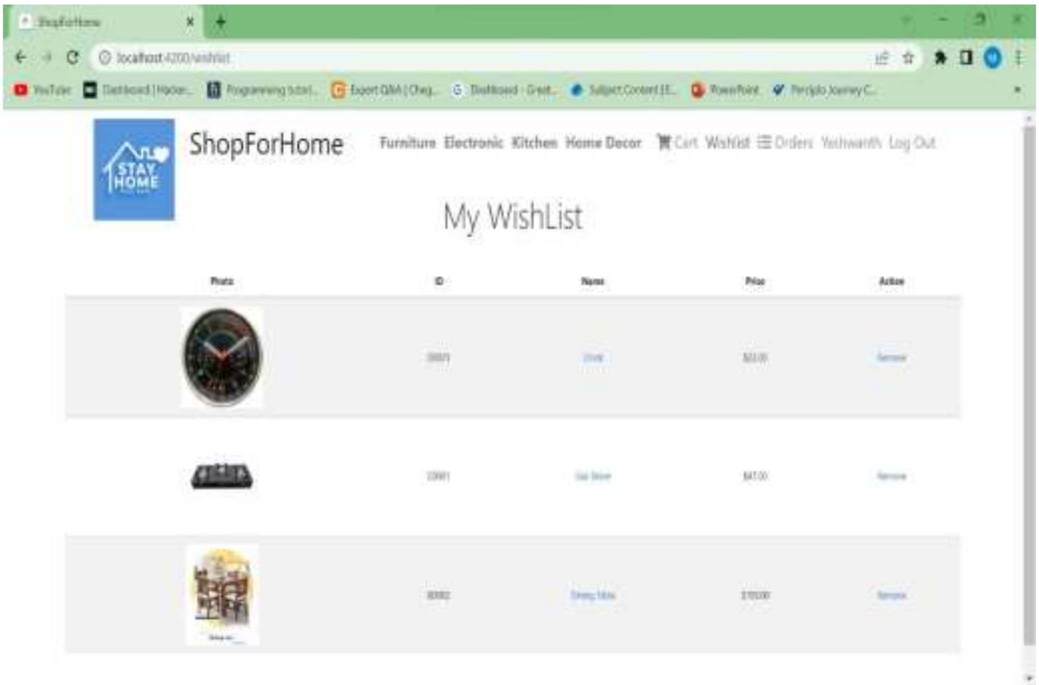


Stay Home Stay Safe

bed

Beds

WISHLIST'S :-



ShopForHome

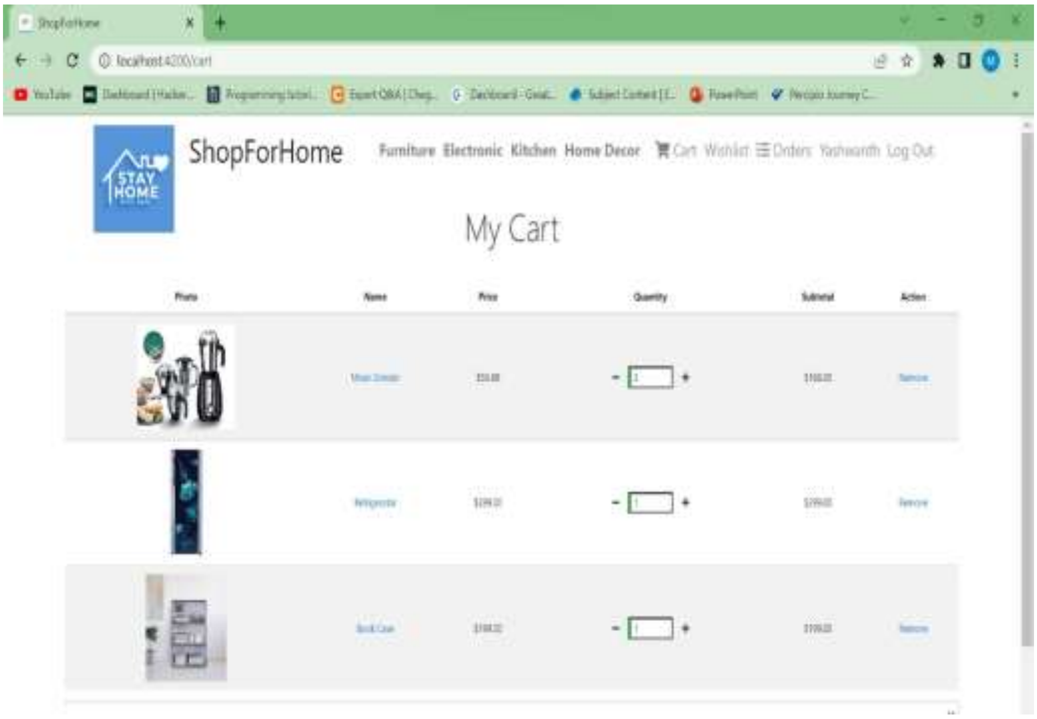
localhost:4200/wishlist

ShopForHome

My WishList

Photo	ID	Name	Price	Action
	[illegible]	Clock	$22.00	Remove
	[illegible]	Gas Stove	$47.00	Remove
	[illegible]	Dining Table	$100.00	Remove

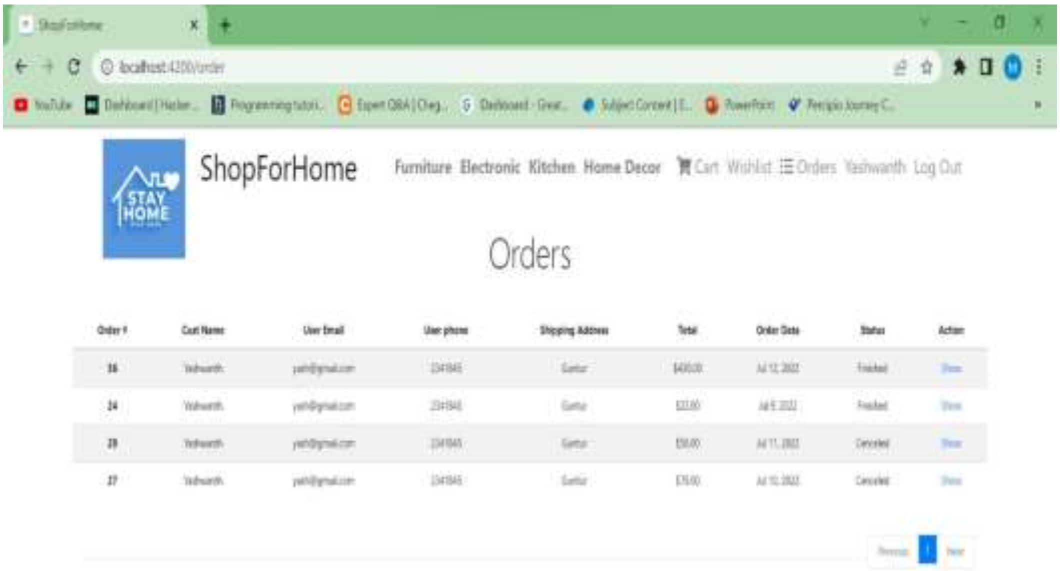
CART :-



My Cart

Photo	Name	Price	Quantity	Subtotal	Action
	Mixer Grinder	$55.00	- 1 +	[illegible]	Remove
	Refrigerator	$299.00	- 1 +	$299.00	Remove
	Book Case	$199.00	- 1 +	$199.00	Remove

ORDER DETAILS :-



Orders

Order #	Cart Name	User Email	User phone	Shipping Address	Total	Order Date	Status	Action
16	Yashwanth	yash@gmail.com	2341845	Guntur	$430.00	Jul 12, 2022	Finished	Show
24	Yashwanth	yash@gmail.com	2341845	Guntur	$22.00	Jul 9, 2022	Finished	Show
29	Yashwanth	yash@gmail.com	2341845	Guntur	$58.00	Jul 11, 2022	Canceled	Show
27	Yashwanth	yash@gmail.com	2341845	Guntur	$75.00	Jul 10, 2022	Canceled	Show

12. Challenges

- Compatibility with browsers like Mozilla Firefox, Internet explorer etc

- Using a layered approach in developing the application which would make the application maintainable.

- Learning new technologies like using Spring Boot behavior and Angular . The overall idea of doing this project is to get a real time experience. Learn new technologies.

13. Conclusion

The ‘Online Shopping’ is designed to provide a web based application that would make searching, viewing and selection of a product easier. The search engine provides an easy and convenient way to search for products where a user can Search for a product interactively and the search engine would refine the products available based on the user’s input. The user can then view the complete specification of each product. They can also view the product reviews and also write their own reviews.

14. Future Scope

The following things can be done in future.

- The current system can be extended to allow the users to create accounts and save products in to wish list.

- The users could subscribe for price alerts which would enable them to receive messages when price for products fall below a particular level.

- The current system is confined only to the shopping cart process. It can be extended to have an easy to use check out process.

- Users can have multiple shipping and billing information saved. During checkout they can use the drag and drop feature to select shipping and billing information.