**Kitchen Story**

Source Code:

* BACKEND

Main

1. KitchenStoryApplication

package com.example;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class KitchenStoryApplication {

public static void main(String[] args) {

SpringApplication.run(KitchenStoryApplication.class, args);

}

}

Controller

1. AdminController

package com.example.controller;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.CrossOrigin;

import org.springframework.web.bind.annotation.DeleteMapping;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.PutMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.RestController;

import com.example.model.AdminLogin;

import com.example.repository.AdminRepository;

import com.example.service.AdminService;

@CrossOrigin("\*")

@RestController

@RequestMapping(path="/admin")

public class AdminController {

// AdminLogin adminLogin;

AdminLogin adminLoggedIn;

private String adminEmail;

@Autowired

AdminService adminService;

@Autowired

AdminRepository adminRepository;

@PostMapping("/login")

public Boolean validateAdmin(@RequestBody AdminLogin adminLogin) {

adminLoggedIn=adminLogin;

adminEmail=adminLogin.getAdminEmail();

adminLoggedIn.setAdminEmail(adminEmail);

Boolean adminLoginStatus = adminService.findByAdminMail(adminLogin);

return adminLoginStatus;

}

@PutMapping("/changePassword")

public void updatePassword(@RequestBody String newPassword) {

AdminLogin existingAdmin = adminRepository.findById(adminLoggedIn.getAdminEmail()).orElse(null);

existingAdmin.setAdminPassword(newPassword);

System.out.println(newPassword+"---"+ adminLoggedIn.getAdminEmail());

adminRepository.save(existingAdmin);

}

}

1. ProductController

package com.example.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.CrossOrigin;

import org.springframework.web.bind.annotation.DeleteMapping;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.ResponseBody;

import org.springframework.web.bind.annotation.RestController;

import com.example.repository.ProductRepository;

import com.example.service.ProductService;

import com.example.model.Products;

@CrossOrigin("\*")

@RestController

@RequestMapping(path="/products")

public class ProductController {

@Autowired

ProductService productService;

@Autowired

ProductRepository productRepository;

@GetMapping("/getAllItems")

public List<Products> getAllItems() {

List<Products> product = (List<Products>) productRepository.findAll();

return product;

}

@GetMapping("/getProduct/{id}")

public Products getProduct(@PathVariable("id")int id) {

Products product =productRepository.findById(id).orElse(null);

return product;

}

@PostMapping(path = "/addItem")

public @ResponseBody void addProduct(@RequestBody Products product) {

System.out.println("inside add product ----"+product.getCategory());

productRepository.save(product);

}

@DeleteMapping("/deleteProduct/{id}")

public void deleteProduct(@PathVariable("id")int id) {

Products product =productRepository.findById(id).orElse(null);

productRepository.delete(product);

}

}

Model

1. AdminLogin

package com.example.model;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

import lombok.Getter;

import lombok.Setter;

@Entity

@Table(name="tbl\_adminLogin")

@Setter

@Getter

public class AdminLogin {

@Id

private String adminEmail;

private String adminPassword;

}

1. Products

package com.example.model;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

import lombok.Getter;

import lombok.Setter;

@Entity

@Table(name="tbl\_products")

@Setter

@Getter

public class Products {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String item;

private String category;

private String description;

private int price;

}

Repository

1. AdminRepository

package com.example.repository;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.stereotype.Repository;

import com.example.model.AdminLogin;

@Repository

public interface AdminRepository extends JpaRepository<AdminLogin, String>{

}

1. ProductRepository

package com.example.repository;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.stereotype.Repository;

import com.example.model.Products;

@Repository

public interface ProductRepository extends JpaRepository<Products, Integer>{

}

1. AdminService

package com.example.service;

import com.example.model.AdminLogin;

public interface AdminService {

Boolean findByAdminMail(AdminLogin adminLogin);

}

1. AdminServiceImpl

package com.example.service;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import com.example.model.AdminLogin;

import com.example.repository.AdminRepository;

@Service

public class AdminServiceImpl implements AdminService {

@Autowired

AdminRepository adminRepository;

@Override

public Boolean findByAdminMail(AdminLogin adminLogin) {

if (adminRepository.findById(adminLogin.getAdminEmail()).isPresent()) {

AdminLogin admin = adminRepository.findById(adminLogin.getAdminEmail()).get();

System.out.println(admin.getAdminPassword()+"--------"+adminLogin.getAdminPassword());

String dbPassword = admin.getAdminPassword().toString();

String userInputPassword =adminLogin.getAdminPassword().toString();

if (dbPassword.equals(userInputPassword)) {

return true;

}

}

return false;

}

}

1. ProductService

package com.example.service;

public interface ProductService {

}

1. ProductServiceImpl

package com.example.service;

import org.springframework.stereotype.Service;

*@Service*

public class ProductSeriveImpl implements ProductService {

}

1. POM

<?**xml** version=*"1.0"* encoding=*"UTF-8"*?>

<**project** xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<**modelVersion**>4.0.0</**modelVersion**>

<**parent**>

<**groupId**>org.springframework.boot</**groupId**>

<**artifactId**>spring-boot-starter-parent</**artifactId**>

<**version**>2.6.3</**version**>

<**relativePath**/> <!-- lookup parent from repository -->

</**parent**>

<**groupId**>com.example</**groupId**>

<**artifactId**>KitchenStory</**artifactId**>

<**version**>0.0.1-SNAPSHOT</**version**>

<**name**>KitchenStory</**name**>

<**description**>Kitchen Story project for Simplilearn</**description**>

<**properties**>

<**java.version**>1.8</**java.version**>

<**maven-jar-plugin.version**>2.6</**maven-jar-plugin.version**>

</**properties**>

<**dependencies**>

<**dependency**>

<**groupId**>org.springframework.boot</**groupId**>

<**artifactId**>spring-boot-starter-data-jpa</**artifactId**>

</**dependency**>

<**dependency**>

<**groupId**>org.springframework.boot</**groupId**>

<**artifactId**>spring-boot-starter-web</**artifactId**>

</**dependency**>

<**dependency**>

<**groupId**>org.projectlombok</**groupId**>

<**artifactId**>lombok</**artifactId**>

<**version**>1.18.20</**version**>

<**scope**>provided</**scope**>

</**dependency**>

<**dependency**>

<**groupId**>org.springframework.boot</**groupId**>

<**artifactId**>spring-boot-devtools</**artifactId**>

<**scope**>runtime</**scope**>

<**optional**>true</**optional**>

</**dependency**>

<**dependency**>

<**groupId**>mysql</**groupId**>

<**artifactId**>mysql-connector-java</**artifactId**>

<**scope**>runtime</**scope**>

</**dependency**>

<**dependency**>

<**groupId**>org.springframework.boot</**groupId**>

<**artifactId**>spring-boot-starter-test</**artifactId**>

<**scope**>test</**scope**>

</**dependency**>

<**dependency**>

<**groupId**>mysql</**groupId**>

<**artifactId**>mysql-connector-java</**artifactId**>

<**version**>5.1.6</**version**>

</**dependency**>

</**dependencies**>

<**build**>

<**plugins**>

<**plugin**>

<**groupId**>org.springframework.boot</**groupId**>

<**artifactId**>spring-boot-maven-plugin</**artifactId**>

</**plugin**>

</**plugins**>

</**build**>

</**project**>

1. Application.properties

spring.datasource.url=jdbc:mysql://localhost:3306/kitchenstory?useSSL=false

spring.datasource.username=root

spring.datasource.password=root

spring.datasource.driver-class-name=com.mysql.jdbc.Driver

spring.jpa.database=mysql

spring.jpa.hibernate.ddl-auto=update

* FRONTEND

**Add Product Component**

1. App.modul.ts
2. import { RouterModule, Routes } from '@angular/router';
3. import { NgModule } from '@angular/core';
4. import { BrowserModule } from '@angular/platform-browser';
5. import { HttpClientModule } from '@angular/common/http';
6. import { AppRoutingModule } from './app-routing.module';
7. import { AppComponent } from './app.component';
8. import { FormsModule } from '@angular/forms';
9. import { ListProductsComponent } from './components/list-products/list-products.component';
10. import { AdminLoginComponent } from './components/admin-login/admin-login.component';
11. import { ChangePasswordComponent } from './components/change-password/change-password.component';
12. import { EditProductsComponent } from './components/edit-products/edit-products.component';
13. import { AddProductsComponent } from './components/add-products/add-products.component';
14. import { BuyProductsComponent } from './components/buy-products/buy-products.component';
15. import { PaymentComponent } from './components/payment/payment.component';
16. const routers: Routes = [
17. {
18. path: 'listProducts',
19. component: ListProductsComponent,
20. },
21. {
22. path: 'editProducts',
23. component: EditProductsComponent,
24. },
25. {
26. path: 'editProducts/:id',
27. component: AddProductsComponent,
28. },
29. {
30. path: 'buyProduct/:id',
31. component: BuyProductsComponent,
32. },
33. {
34. path: 'addProducts',
35. component: AddProductsComponent,
36. },
37. {
38. path: 'admin',
39. component: AdminLoginComponent,
40. },
41. {
42. path: 'payment/:id',
43. component: PaymentComponent,
44. },
45. {
46. path: 'admin/changePassword',
47. component: ChangePasswordComponent,
48. },
49. ];
50. @NgModule({
51. declarations: [
52. AppComponent,
53. ListProductsComponent,
54. AdminLoginComponent,
55. ChangePasswordComponent,
56. EditProductsComponent,
57. AddProductsComponent,
58. BuyProductsComponent,
59. PaymentComponent,
60. ],
61. imports: [
62. RouterModule.forRoot(routers),
63. BrowserModule,
64. AppRoutingModule,
65. FormsModule,
66. HttpClientModule,
67. ],
68. providers: [],
69. bootstrap: [AppComponent],
70. })
71. export class AppModule {}

2. app.component.ts

import { Component } from '@angular/core';

@Component({

  selector: 'app-root',

  templateUrl: './app.component.html',

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  title = 'KitchenStory\_frontend';

}

3. app.coponent.html

<div class="row text-center text-black mb-5">

    <div class="col-lg-7 mx-auto">

        <h1 class="display-4">Kitchen Story</h1>

    </div>

</div>

<div class="row" style="justify-content: center;">

    <div style="padding: 2rem;">

        <a routerLink="/admin">Admin Login</a><br>

    </div>

    <div style="padding: 2rem;">

        <a routerLink="/listProducts">Buy Products</a>

    </div>

</div>

<router-outlet></router-outlet>

<!-- <img src="https://w0.peakpx.com/wallpaper/760/93/HD-wallpaper-food-still-life-meat-mushroom-spices.jpg"

    style="position: sticky; width: 100%;" alt=""> -->

**ADD PRODUCT COMPONENT**

Add-product.component.ts

import { Products } from './../../models/products';

import { Component, OnInit } from '@angular/core';

import { ActivatedRoute, Router } from '@angular/router';

import { ProductsService } from './../../services/products.service';

@Component({

  selector: 'app-add-products',

  templateUrl: './add-products.component.html',

  styleUrls: ['./add-products.component.css'],

})

export class AddProductsComponent implements OnInit {

  product: Products = new Products();

  constructor(

    private \_productService: ProductsService,

    private \_router: Router,

    private \_activatedRoute: ActivatedRoute

  ) {}

  ngOnInit(): void {

    const isIdPresent = this.\_activatedRoute.snapshot.paramMap.has('id');

    if (isIdPresent) {

      const id = +this.\_activatedRoute.snapshot.paramMap.get('id')!;

      this.\_productService

        .getProduct(id)

        .subscribe((data) => (this.product = data));

    }

  }

  saveProduct() {

    console.log('---- inside save product');

    this.\_productService.saveProduct(this.product).subscribe((data) => {

      alert('product added successfully');

      this.\_router.navigateByUrl('/editProducts');

    });

  }

  deleteProduct(id: number) {

    this.\_productService.deleteProduct(id).subscribe((data) => {

      alert('product deleted successfully');

      this.\_router.navigateByUrl('/editProducts');

    });

  }

}

Add-product.component.html

<div class="container" style="width: 50%;">

    <form (ngSubmit)="saveProduct()">

        <div class="form-group">

            <input type="text" name="product" class="form-control" [(ngModel)]="product.category"

                placeholder="Enter the Product category">

        </div>

        <div class="form-group">

            <input type="text" name="item" class="form-control" [(ngModel)]="product.item"

                placeholder="Enter the item ">

        </div>

        <div class="form-group">

            <textarea type="text" name="description" [(ngModel)]="product.description"

                placeholder="Enter the Description" class="form-control"></textarea>

        </div>

        <div class="form-group">

            <input type="number" name="amount" class="form-control" [(ngModel)]="product.price"

                placeholder="Enter the price ">

        </div>

        <button type="submit" class="btn btn-primary">Save / Update</button>

    </form>

    <br>

    <button \*ngIf="product.id" (click)="deleteProduct(product.id)" class="btn btn-danger">Delete</button>

</div>

**ADMIN LOGIN COMPONENT**

Admin-login.component.ts

import { AdminService } from './../../services/admin.service';

import { Admin } from './../../models/admin';

import { Component, OnInit } from '@angular/core';

import { ActivatedRoute, Router } from '@angular/router';

@Component({

  selector: 'app-admin-login',

  templateUrl: './admin-login.component.html',

  styleUrls: ['./admin-login.component.css'],

})

export class AdminLoginComponent implements OnInit {

  admin: Admin = new Admin();

  adminValidated: boolean;

  constructor(

    private \_adminService: AdminService,

    private \_router: Router,

    private \_activatedRoute: ActivatedRoute

  ) {}

  ngOnInit(): void {}

  validateAdmin(): any {

    this.\_adminService.validateAdmin(this.admin).subscribe((data) => {

      if (data) {

        this.\_router.navigateByUrl('/admin/changePassword');

      } else {

        alert('Wrong credentials');

        this.\_router.navigateByUrl('/admin');

      }

    });

    //

  }

}

Admin-login.component.html

<!-- <form (ngSubmit)="validateAdmin()">

    <input type="text" name="adminEmail" [(ngModel)]="admin.adminEmail">

    <input type="password" name="adminPassword" [(ngModel)]="admin.adminPassword">

    <button type="submit">Login</button>

</form> -->

<div class="container" style="width: 50%;">

    <form (ngSubmit)="validateAdmin()">

        <div class="form-group">

            <input type="text" class="form-control" name="adminEmail" [(ngModel)]="admin.adminEmail"

                placeholder="Enter email">

            <small id="emailHelp" class="form-text text-muted">We'll never share your email with anyone else.</small>

        </div>

        <div class="form-group">

            <input type="password" class="form-control" name="adminPassword" [(ngModel)]="admin.adminPassword"

                placeholder="Password">

        </div>

        <button type="submit" class="btn btn-primary">Submit</button>

    </form>

</div>

BUY PRODUCT COMPONENT

Buy-product.component.ts

import { Component, OnInit } from '@angular/core';

import { ActivatedRoute, Router } from '@angular/router';

import { Products } from 'src/app/models/products';

import { ProductsService } from 'src/app/services/products.service';

@Component({

  selector: 'app-buy-products',

  templateUrl: './buy-products.component.html',

  styleUrls: ['./buy-products.component.css'],

})

export class BuyProductsComponent implements OnInit {

  product: Products = new Products();

  constructor(

    private \_productService: ProductsService,

    private \_router: Router,

    private \_activatedRoute: ActivatedRoute

  ) {}

  ngOnInit(): void {

    const isIdPresent = this.\_activatedRoute.snapshot.paramMap.has('id');

    if (isIdPresent) {

      const id = +this.\_activatedRoute.snapshot.paramMap.get('id')!;

      this.\_productService

        .getProduct(id)

        .subscribe((data) => (this.product = data));

    }

  }

}

Buy-product.component.html

<div class="productBody" style="padding: 2rem;">

    <div class="container" style="margin: 0px auto;">

        <div class="card" style="background-color: #2e2e30;">

            <div class="card-header">

                <h4>{{product.category}}</h4>

            </div>

            <div class="card-body">

                <h3 class="card-title">{{product.item}}</h3>

                <p class="card-text">{{product.description}}</p>

                <h5>{{product.price}}</h5>

                <a routerLink="/payment//{{product.id}}" class="btn btn-primary">Proceed to payment</a>

            </div>

        </div>

    </div>

</div>

Change-product.component.ts

import { Component, OnInit } from '@angular/core';

import { ActivatedRoute, Router } from '@angular/router';

import { AdminService } from 'src/app/services/admin.service';

import { HttpClient } from '@angular/common/http';

@Component({

  selector: 'app-change-password',

  templateUrl: './change-password.component.html',

  styleUrls: ['./change-password.component.css'],

})

export class ChangePasswordComponent implements OnInit {

  public newPassword: string;

  public confirmPassword: string;

  private getUrl: string = 'http://localhost:8080/admin/changePassword';

  constructor(

    private \_router: Router,

    private \_activatedRoute: ActivatedRoute,

    private \_adminService: AdminService,

    private \_httpClient: HttpClient

  ) {}

  ngOnInit(): void {}

  changePassword(newPassword: string, confirmPassword: string): any {

    console.log(newPassword, '----', confirmPassword);

    this.\_httpClient.put(this.getUrl, newPassword).subscribe((data) => {

      alert('Successfully updated the password.');

      this.\_router.navigateByUrl('/editProducts');

    });

  }

}

Change-password.component.html

<div class="container" style="width: 50%;" ng-controller="MyFormController">

    <form (ngSubmit)="changePassword(newPassword,confirmPassword)">

        <div class="form-group">

            <input type="password" class="form-control" name="newPassword" [(ngModel)]="newPassword"

                placeholder="Enter new password">

        </div>

        <div class="form-group">

            <input type="password" class="form-control" name="confirmPassword" [(ngModel)]="confirmPassword"

                placeholder="Confirm password">

        </div>

        <button type="submit" class="btn btn-primary">Change</button>

    </form>

    <br>

    <a routerLink="/editProducts">skip</a>

</div>

Edit-product.component.ts

import { Component, OnInit } from '@angular/core';

import { Products } from './../../models/products';

import { ProductsService } from './../../services/products.service';

@Component({

  selector: 'app-edit-products',

  templateUrl: './edit-products.component.html',

  styleUrls: ['./edit-products.component.css'],

})

export class EditProductsComponent implements OnInit {

  products: Products[] = [];

  constructor(private \_productService: ProductsService) {}

  ngOnInit(): void {

    this.listProducts();

  }

  listProducts() {

    this.\_productService

      .getProducts()

      .subscribe((data) => (this.products = data));

  }

}

Edit-product.component.html

<div class="container">

    <br>

    <a routerLink="/addProducts">Add Expense</a>

    <br>

    <br>

    <br>

    <h3 style="text-align: center;">Master List of Products</h3>

    <table class="table table-striped">

        <thead>

            <tr>

                <th>Category</th>

                <th>Item</th>

                <th>Description</th>

                <th>Price</th>

                <th>#</th>

            </tr>

        </thead>

        <br>

        <tbody ng-repeat="product in products">

            <tr \*ngFor="let product of products">

                <td scope="row">{{product.category}}</td>

                <td>{{product.item}}</td>

                <td>{{product.description}}</td>

                <td>{{product.price}}</td>

                <td>

                    <a routerLink="/editProducts/{{product.id}}">Edit</a>

                </td>

            </tr>

            <br>

        </tbody>

    </table>

</div>

**LIST PRODUCT COMPONENT**

List-product.component.ts

import { Products } from './../../models/products';

import { ProductsService } from './../../services/products.service';

import { Component, OnInit } from '@angular/core';

@Component({

  selector: 'app-list-products',

  templateUrl: './list-products.component.html',

  styleUrls: ['./list-products.component.css'],

})

export class ListProductsComponent implements OnInit {

  products: Products[] = [];

  filters = {

    keyword: '',

    sortBy: 'Name',

  };

  constructor(private \_productService: ProductsService) {}

  ngOnInit(): void {

    this.listProducts();

  }

  listProducts() {

    this.\_productService

      .getProducts()

      .subscribe((data) => (this.products = this.filterProducts(data)));

  }

  filterProducts(products: Products[]) {

    return products.filter((e) => {

      return e.category

        .toLowerCase()

        .includes(this.filters.keyword.toLowerCase());

    });

  }

}

List-product.component.html

<!-- --------------------------------------------------------------- -->

<div class="productsBody">

    <div class="container py-5">

        <div class="row">

            <div class="col-lg-8 mx-auto">

                <div class="form-group">

                    <input class="form-control" type="text" [(ngModel)]="filters.keyword" name="keyword"

                        (input)="listProducts()" placeholder="Search by category">

                </div>

                <!-- List group-->

                <ul class="list-group shadow">

                    <!-- list group item-->

                    <div \*ngFor="let product of products">

                        <li class="list-group-item" style="background-color: #2e2e30;">

                            <!-- Custom content-->

                            <div class="media align-items-lg-center flex-column flex-lg-row p-3">

                                <div class="media-body order-2 order-lg-1">

                                    <h4 class="mt-0 font-weight-bold mb-2">{{product.category}}</h4>

                                    <div class="row  text-white mb-5">

                                        <div class="col-lg-7 ">

                                            <h3 class="display-4">{{product.item}}</h3>

                                        </div>

                                    </div>

                                    <h5 class="font-italic text-muted mb-0 ">{{product.description}}</h5>

                                    <!-- <div class="d-flex align-items-center justify-content-between mt-1">

                                    <h6 class="font-weight-bold my-2">₹64,999</h6>

                                </div> -->

                                </div>

                            </div> <!-- End -->

                            <div style="padding: 1rem;">

                                <a routerLink="/buyProduct/{{product.id}}">Buy</a>

                            </div>

                        </li> <!-- End -->

                    </div>

                </ul> <!-- End -->

            </div>

        </div>

    </div>

</div>

PAYMENT COMPONENT

Payment component.ts

import { Component, OnInit } from '@angular/core';

import { ActivatedRoute, Router } from '@angular/router';

import { Products } from 'src/app/models/products';

import { ProductsService } from 'src/app/services/products.service';

@Component({

  selector: 'app-payment',

  templateUrl: './payment.component.html',

  styleUrls: ['./payment.component.css'],

})

export class PaymentComponent implements OnInit {

  product: Products = new Products();

  constructor(

    private \_productService: ProductsService,

    private \_router: Router,

    private \_activatedRoute: ActivatedRoute

  ) {}

  ngOnInit(): void {

    const isIdPresent = this.\_activatedRoute.snapshot.paramMap.has('id');

    if (isIdPresent) {

      const id = +this.\_activatedRoute.snapshot.paramMap.get('id')!;

      this.\_productService

        .getProduct(id)

        .subscribe((data) => (this.product = data));

    }

  }

}

Payment.component.html

<div class="paymentBody">

    <div class="container p-0">

        <div class="card px-4">

            <h2 style="text-align: center;">Payment Details</h2>

            <div class="row gx-3">

                <div class="col-12">

                    <div class="d-flex flex-column">

                        <input class="form-control mb-3" type="text" placeholder="Name" required>

                    </div>

                </div>

                <div class="col-12">

                    <div class="d-flex flex-column">

                        <input class="form-control mb-3" type="text" placeholder="Card Number" required>

                    </div>

                </div>

                <div class="col-6">

                    <div class="d-flex flex-column">

                        <input class="form-control mb-3" type="text" placeholder="Expiry - MM/YYYY" required>

                    </div>

                </div>

                <div class="col-6">

                    <div class="d-flex flex-column">

                        <input class="form-control mb-3 pt-2 " type="password" placeholder="CVV/CVC - \*\*\*" required>

                    </div>

                </div>

                <div class="col-12">

                    <input class="btn btn-primary mb-3" type="submit" onclick="alert('Product ordered successfully !!')"

                        value="Pay Rs. {{product.price}}" style="text-align: center; justify-content: center;">

                </div>

            </div>

        </div>

    </div>

</div>

Admin.ts

export class Admin {

  adminEmail: string;

  adminPassword: string;

}

Product.ts

export class Products {

  id: number;

  item: string;

  category: string;

  description: string;

  price: number;

}

Admin.service.ts

import { Injectable } from '@angular/core';

import { Observable, map } from 'rxjs';

import { HttpClient } from '@angular/common/http';

import { Admin } from '../models/admin';

@Injectable({

  providedIn: 'root',

})

export class AdminService {

  private getUrl: string = 'http://localhost:8080/admin/login';

  private adminValidated: boolean;

  constructor(private \_httpClient: HttpClient) {}

  validateAdmin(admin: Admin): Observable<Admin> {

    return this.\_httpClient.post<Admin>(this.getUrl, admin);

  }

}

Product.service.ts

import { Products } from './../models/products';

import { HttpClient } from '@angular/common/http';

import { Injectable } from '@angular/core';

import { Observable, map } from 'rxjs';

@Injectable({

  providedIn: 'root',

})

export class ProductsService {

  private getUrl1: string = 'http://localhost:8080/products/getAllItems';

  private getUrl2: string = 'http://localhost:8080/products/addItem';

  private getUrl3: string = 'http://localhost:8080/products/getProduct';

  private getUrl4: string = 'http://localhost:8080/products/deleteProduct';

  constructor(private \_httpClient: HttpClient) {}

  getProducts(): Observable<Products[]> {

    return this.\_httpClient

      .get<Products[]>(this.getUrl1)

      .pipe(map((response) => response));

  }

  getProduct(id: number): Observable<Products> {

    return this.\_httpClient

      .get<Products>(`${this.getUrl3}/${id}`)

      .pipe(map((response) => response));

  }

  saveProduct(product: Products): Observable<Products> {

    console.log('----inside service ---', product.category);

    return this.\_httpClient.post<Products>(this.getUrl2, product);

  }

  deleteProduct(id: number): Observable<any> {

    return this.\_httpClient.delete(`${this.getUrl4}/${id}`, {

      responseType: 'text',

    });

  }

}