SOURCE CODE FOR DRIVE YOUR WAY

BACK-END APP:

```
package com;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.boot.autoconfigure.domain.EntityScan;
import
org.springframework.data.jpa.repository.config.EnableJpaRepositories;

@SpringBootApplication(scanBasePackages = "com")
@EntityScan(basePackages = "com.onlineshop.bean")
@EnableJpaRepositories(basePackages = "com.onlineshop.repository")
public class MyAppApplication {

public static void main(String[] args) {
SpringApplication.run(MyAppApplication.class, args);
System.out.println("Server running on port number 9090");
}

}
```

LOGIN.JAVA

```
package com.onlineshop.bean;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;

@Entity
public class Login {
@Id
private String emailid;
private String password;
@Column(name = "typeofuser")
private String typeOfUser;
public String getEmailid() {
return emailid;
}
public void setEmailid(String emailid) {
this.emailid = emailid;
}
public String getPassword() {
return password;
}
public void setPassword(String password) {
this.password = password;
}
public String getTypeOfUser() {
return typeOfUser;
}
public void setTypeOfUser(String typeOfUser) {
this.typeOfUser = typeOfUser;
}
@Override
public String toString() {
return "Login [emailid=" + emailid + ", password=" + password + ",
typeOfUser=" + typeOfUser + "]";
}

}
```

PRODUCT.JAVA

```
package com.onlineshop.bean;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class Product {
@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)            // auto_generate
private int pid;
private String pname;
private float price;
private String url;
public int getPid() {
        return pid;
}
public void setPid(int pid) {
        this.pid = pid;
}
public String getPname() {
        return pname;
}
public void setPname(String pname) {
        this.pname = pname;
}
public float getPrice() {
        return price;
}
public void setPrice(float price) {
        this.price = price;
}
public String getUrl() {
        return url;
}
public void setUrl(String url) {
        this.url = url;
}
@Override
public String toString() {
	return "Product [pid=" + pid + ", pname=" + pname + ", price=" + price + ", 
url=" + url + "]";
}

}
```

LOGINCONTROLLER.JAVA

```
package com.onlineshop.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.MediaType;
import org.springframework.web.bind.annotation.CrossOrigin;
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.RestController;

import com.onlineshop.bean.Login;
import com.onlineshop.service.LoginService;

@RestController
@RequestMapping("login")
@CrossOrigin
public class LoginController {


	@Autowired
	LoginService loginService;


	@PostMapping(value = "signIn",consumes = 
MediaType.APPLICATION_JSON_VALUE)
	public String signIn(@RequestBody Login login) {
		System.out.println("I cam here");
		return loginService.signIn(login);

	}

	@PostMapping(value = "signUp",consumes = 
MediaType.APPLICATION_JSON_VALUE)
	public String signUp(@RequestBody Login login) {
		System.out.println(login);
		return loginService.signUp(login);
	}
}
```

PRODUCTCONTROLLER.JAVA

```
package com.onlineshop.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.MediaType;
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.PatchMapping;
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.RestController;

import com.onlineshop.bean.Product;
import com.onlineshop.service.ProductService;

@RestController
@RequestMapping("product")
@CrossOrigin
public class ProductController {

@Autowired
ProductService productService;
@PostMapping(value = "storeProduct",consumes =
MediaType.APPLICATION_JSON_VALUE)
public String storeProduct(@RequestBody Product product) {
return productService.storeProduct(product);
}
@PatchMapping(value = "updateProduct",consumes =
MediaType.APPLICATION_JSON_VALUE)
public String updateProduct(@RequestBody Product product) {
return productService.updateProduct(product);
}
@GetMapping(value="findAllProduct",produces =
MediaType.APPLICATION_JSON_VALUE)
public List<Product> getAllProduct() {
return productService.getAllProducts();
}
@GetMapping(value="findProductByPrice/{price}",produces =
MediaType.APPLICATION_JSON_VALUE)
public List<Product> findProductByPrice(@PathVariable("price") float
price) {
return productService.findProductByPrice(price);
}
@GetMapping(value="findAllProduct/{pid}")
public String findProductById(@PathVariable("pid") int pid) {
return productService.findProductById(pid);
}
@DeleteMapping(value="deleteProduct/{pid}")
public String deleteProductUsingId(@PathVariable("pid") int pid) {
return productService.deleteProduct(pid);
}
}
```

LOGINREPOSITORY.JAVA

```java
package com.onlineshop.repository;

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

import com.onlineshop.bean.Login;

@Repository
public interface LoginRepository extends JpaRepository<Login,
String>{

}
```

PRODUCTREPOSITORY.JAVA

```java
package com.onlineshop.repository;

import java.util.List;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.query.Param;
import org.springframework.stereotype.Repository;

import com.onlineshop.bean.Product;

@Repository
public interface ProductRepository extends JpaRepository<Product,
Integer>{

//JPQL
@Query("select p from Product p where p.price > :price")
public List<Product> findProductByPrice(@Param("price") float price);
}
```

LOGINSERVICE.JAVA

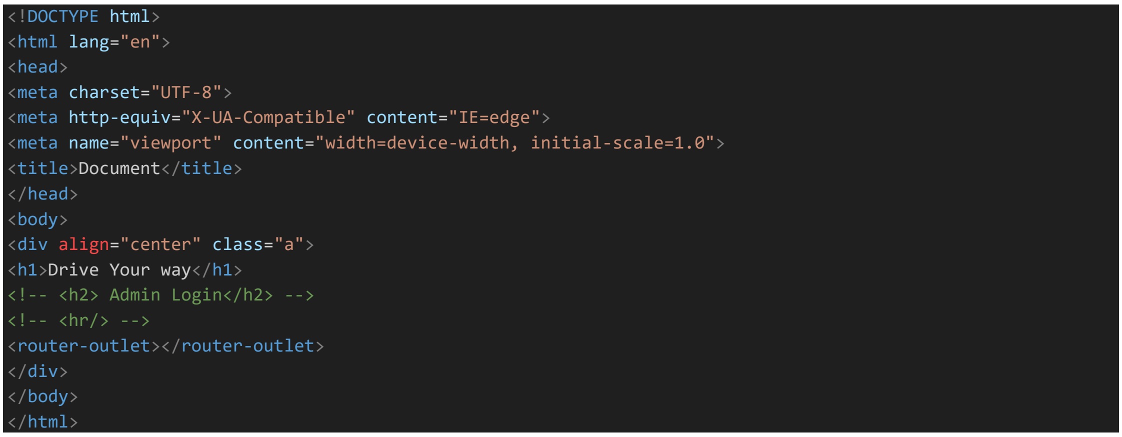
```java
package com.onlineshop.service;

import java.util.Optional;

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

import com.onlineshop.bean.Login;
import com.onlineshop.repository.LoginRepository;

@Service
public class LoginService {
@Autowired
LoginRepository loginRepository;
public String signIn(Login login) {
Optional<Login> result = loginRepository.findById(login.getEmailid());
if(result.isPresent()) {
Login ll = result.get();
if(ll.getPassword().equals(login.getPassword())) {
if(login.getTypeOfUser().equals(ll.getTypeOfUser()) &&
login.getTypeOfUser().equals("admin")) {
return "Admin sucessfully login";
}else if(login.getTypeOfUser().equals(ll.getTypeOfUser()) &&
login.getTypeOfUser().equals("user")){
return "User successfully login";
}else {
return "Invalid details";
}
}else {
return "InValid password";
}
}else {
return "InValid emailId";
}
}
public String signUp(Login login) {
Optional<Login> result = loginRepository.findById(login.getEmailid());
if(result.isPresent()) {
return "Email Id alreay exists";
}else {
if(login.getTypeOfUser().equals("admin")) {
return "You can't create admin account";
}else {
loginRepository.save(login);
return "Account created successfully";
}
}
}
}
```

PRODUCTSERVICE.JAVA

```
package com.onlineshop.service;

import java.util.List;
import java.util.Optional;

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

import com.onlineshop.bean.Product;
import com.onlineshop.repository.ProductRepository;

@Service
public class ProductService {

@Autowired
ProductRepository productRepository;
public String storeProduct(Product product) {
productRepository.save(product);
return "Product details stored";
}
public List<Product> getAllProducts() {
return productRepository.findAll();
}
public String findProductById(int pid) {
Optional<Product> result = productRepository.findById(pid);
if(result.isPresent()) {
Product p = result.get();
return p.toString();
}else {
return "Product not present";
}
}
public List<Product> findProductByPrice(float price){
return productRepository.findProductByPrice(price);
}
public String deleteProduct(int pid) {
Optional<Product> result = productRepository.findById(pid);
if(result.isPresent()) {
Product p = result.get();
productRepository.delete(p);
return "Product deleted successfully";
}else {
return "Product not present";
}
}
public String updateProduct(Product product) {
Optional<Product> result =
productRepository.findById(product.getPid());
if(result.isPresent()) {
Product p = result.get();
p.setPrice(product.getPrice());
p.setUrl(product.getUrl());
productRepository.saveAndFlush(p);
return "Product updated successfully";
}else {
return "Product not present";
}
}
}
```

FRONTEND-APP:

APP.COMPONENET.HTML



```html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document</title>
</head>
<body>
<div align="center" class="a">
<h1>Drive Your way</h1>
<!-- <h2> Admin Login</h2> -->
<!-- <hr/> -->
<router-outlet></router-outlet>
</div>
</body>
</html>
```

APP.COMPONENET.TS

```
import { Component } from '@angular/core';
@Component({
selector: 'app-root',
templateUrl: './app.component.html',
styleUrls: ['./app.component.css']
})
export class AppComponent {
title = 'frontend-app';
}
```

APP.COMPONENET.CSS

```
body{
    background: rgb(238,174,202);
    background: radial-gradient(circle, rgba(238,174,202,1) 0%, rgba(148,187,233,1) 100%);
    height: 100vh;
    padding: 0px;
    margin: 0px;
    }
    .a{
    padding-top: 10px;
    color: crimson;
}
```

LOGIN.SERVICE.TS

```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { Observable } from 'rxjs';
@Injectable({
providedIn: 'root'
})
export class LoginService {
baseURL:string ="http://localhost:9090/login";
constructor(public http:HttpClient) { }
signIn(login:any):Observable<string> {
return this.http.post(this.baseURL+"/signIn",login,{responseType:"text"});
}
signUp(login:any):Observable<string> {
    return this.http.post(this.baseURL+"/signUp",login,{responseType:"text"});
    }
    }
```

LOGIN.TS

```
// map to entity class or json data.
export class Login {
constructor(public emailid:string,
public password:string,
public typeOfUser:string){}
}
```

PRODUCT.SERVICE.TS

```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { Observable } from 'rxjs';
import { Product } from './product';
@Injectable({
providedIn: 'root'
})
export class ProductService {
baseUrl:string ="http://localhost:9090/product"
constructor(public http:HttpClient) { }
storeProduct(product:any):Observable<string> {
return this.http.post(this.baseUrl+"/storeProduct",product,{responseType:"text"});
}
updateProduct(product:any):Observable<string> {
    return this.http.patch(this.baseUrl+"/updateProduct",product,{responseType:"text"});
    }
    findAllProduct():Observable<Product[]> {
    return this.http.get<Product[]>(this.baseUrl+"/findAllProduct");
    }
    findAllProductByPrice(price:number):Observable<Product[]> {
    return this.http.get<Product[]>(this.baseUrl+"/findProductByPrice/"+price);
    }
    findAllProductById(pid:number):Observable<string> {
    return this.http.get(this.baseUrl+"/findAllProduct/"+pid,{responseType:"text"});
    }
    deleteProductById(pid:number):Observable<string> {
    return this.http.delete(this.baseUrl+"/deleteProduct/"+pid,{responseType:"text"});
    }
    }
```

LOGIN.COMPONENET.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document</title>
</head>
<body>
<div class="admin">
<div class="signIn">
<h2>Login Page</h2>
<form [formGroup]="loginRef" (ngSubmit)="signIn()">
<div class="form-container-t3">
<label>EmailId</label>
<input type="email" formControlName="emailid" placeholder="enter email"/><br/>
<label>Password</label>
<input type="password" formControlName="password" placeholder="enter password"/><br/>
<label>TypeOfUser</label>
<input type="radio" id="admin1" name="typeOfUser" value="admin"
formControlName="typeOfUser"/>admin
<input type="radio" id="user1" name="typeOfUser" value="user"
formControlName="typeOfUser"/>user<br/>
<input type="submit" id="signIn12" class="btn" value="signIn"/>
<input type="reset" class="btn" value="reset"/>
</div>
</form>
<br/>
<span style="color:red">{{msg}}</span><br/>
<a routerLink="/signUp">SignUp</a>
</div>
</div>
</body>
</html>
```

LOGIN.COMPONENT.TS

```
import { Component, OnInit } from '@angular/core';
import {FormGroup,FormControl} from '@angular/forms';
import { Router } from '@angular/router';
import { LoginService } from '../login.service';
@Component({
selector: 'app-login',
templateUrl: './login.component.html',
styleUrls: ['./login.component.css']
})
export class LoginComponent implements OnInit {
loginRef = new FormGroup({
emailid:new FormControl(),
password:new FormControl(),
typeOfUser:new FormControl()
});
msg:string=""
constructor(public ls:LoginService,public router:Router) { }
ngOnInit(): void {
}
signIn(){
let login = this.loginRef.value;
console.log(login);
this.ls.signIn(login).subscribe({
next:(result:any)=>{
console.log(result);
if(result=="Admin sucessfully login"){
sessionStorage.setItem("userDetails",login.emailid);
this.router.navigate(["adminHome"])
}else if(result=="User successfully login"){
sessionStorage.setItem("userDetails",login.emailid);
this.router.navigate(["userHome"])
}else {
this.msg=result;
}
},
error:(error:any)=>console.log(error),
complete:()=>console.log("completed")
})
}
}
```

SIGNUP.COMPONENET.HTML

```
<div class="user">
    <div class="signUp">
    <h2>Account Create</h2>
    <form [formGroup]="loginRef" (ngSubmit)="signUp()">
    <label>EmailId</label>
    <input type="email" formControlName="emailid"/><br/>
<label>Password</label>
<input type="password" formControlName="password"/><br/>
<label>TypeOfUser</label>
<input type="radio" name="typeOfUser" value="admin" formControlName="typeOfUser"/>admin
<input type="radio" name="typeOfUser" value="user" formControlName="typeOfUser"/>user<br/>
<input type="submit" class="btn"value="signUp"/>
<input type="reset" class="btn" value="reset"/>
</form>
<br/>
<span style="color:red">{{msg}}</span><br/>
<a routerLink="/login">login</a>
</div>
</div>
```

SIGNUP.COMPONENT.TS

```
import { Component, OnInit } from '@angular/core';
import {FormGroup,FormControl} from '@angular/forms';
import { LoginService } from '../login.service';
@Component({
selector: 'app-signup',
templateUrl: './signup.component.html',
styleUrls: ['./signup.component.css']
})
export class SignupComponent implements OnInit {
loginRef = new FormGroup({
emailid:new FormControl(),
password:new FormControl(),
typeOfUser:new FormControl()
});
msg:string=""
constructor(public ls:LoginService) { }
ngOnInit(): void {
}
signUp() {
let login = this.loginRef.value;
this.ls.signUp(login).subscribe({
next:(result:any)=>this.msg=result,
error:(error:any)=>console.log(error),
complete:()=>console.log("completed")
})
}
}
```

ADMINDASHBOARD.COMPONENT.HTML

```
<div>
    <h2>Welcome to home page admin {{user}}</h2>
    <a id="add1" routerLink="addProduct">Add Product</a> |
    <a routerLink="findAllProduct">View Product</a>
    <br/>
    <hr/>
    <router-outlet></router-outlet>
    <hr/>
<br/>
<input type="button" value="logout" (click)="logout()"/>
</div>
```

ADMINDASHBOARD.COMPONENT.TS

```
import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
@Component({
selector: 'app-admindashboard',
templateUrl: './admindashboard.component.html',
styleUrls: ['./admindashboard.component.css']
})
export class AdmindashboardComponent implements OnInit {
user:string ="";
constructor(private router:Router) { }
ngOnInit(): void {
let obj = sessionStorage.getItem("userDetails");
if(obj!=null){
this.user=obj;
}
}
logout() {
    sessionStorage.removeItem("userDetails");
    this.router.navigate(["login"]);
    }
    }
```

ADD-PRODUCT.COMPONENET.HTML

```
<div>
    <h2>Add Product</h2>
    <form [formGroup]="productRef" (ngSubmit)="storeProduct()">
    <label>PName</label>
    <input id ="pname11" type="text" formControlName="pname"><br/>
    <label>Price</label>
    <input id="price11" type="number" formControlName="price"><br/>
    <label>URL</label>
    <input id="url11" type="url" formControlName="url"><br/>
    <input id="submit11" type="submit" value="store Product"/><br/>
    <input type="reset" value="reset"/><br/>
    </form><br/>
    <span style="color:red">{{storeMsg}}</span>
    </div>
```

ADD-PRODUCT.COMPONENT.TS

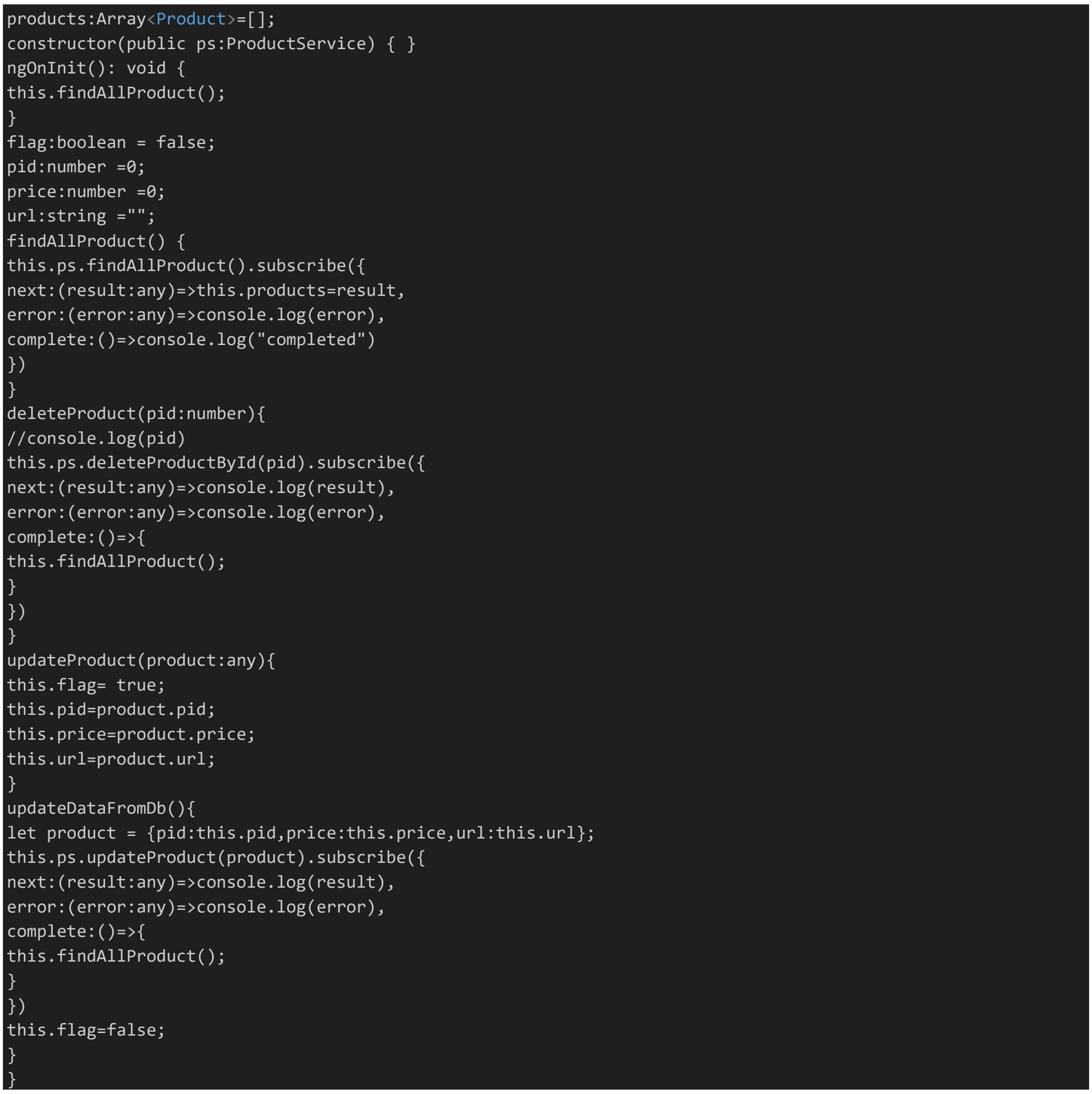
```
import { Component, OnInit } from '@angular/core';
import {FormGroup,FormControl} from '@angular/forms'
import { ProductService } from '../product.service';
@Component({
selector: 'app-add-product',
templateUrl: './add-product.component.html',
styleUrls: ['./add-product.component.css']
})
export class AddProductComponent implements OnInit {
productRef = new FormGroup({
pname:new FormControl(),
price:new FormControl(),
url:new FormControl()
})
storeMsg :string =""
constructor(public ps:ProductService) { }
ngOnInit(): void {
}

storeProduct() {
    let product = this.productRef.value;
    this.ps.storeProduct(product).subscribe({
    next:(result:any)=>this.storeMsg=result,
    error:(error:any)=>console.log(error),
    complete:()=>console.log("completed")
    })
    this.productRef.reset();
    }
    }
```

ADMIN-PRODUCT-RETRIEVE.COMPONENT.HTML

```
<div>
    <h2>Add Product Details</h2>
    <div *ngIf="flag">
    <h2>Update Product</h2>
    <form (ngSubmit)="updateDataFromDb()">
    <label>PId</label>
    <input type="number" name="pid" [(ngModel)]="pid" readonly/><br/>
    <label>Price</label>
    <input type="number" name="price" [(ngModel)]="price"/><br/>
    <label>URL</label>
    <input type="URL" name="url" [(ngModel)]="url"/><br/>
    <input type="submit" value="update data"/>
    <input type="reset" value="reset"/>
    </form>
    </div>
    <span *ngFor="let p of products">
    <img src={{p.url}} width="300px" height="300px"/>
    <span>{{p.pname}} {{p.price}}</span>
    <input type="button" value="delete" (click)="deleteProduct(p.pid)"/>
    <input type="button" value="update" (click)="updateProduct(p)"/>
    </span>
    </div>
```

ADMIN-PRODUCT-RETRIEVE.COMPONENT.TS

```
import { Component, OnInit } from '@angular/core';
import { Product } from '../product';
import { ProductService } from '../product.service';
@Component({
selector: 'app-admin-product-retrieve',
templateUrl: './admin-product-retrieve.component.html',
styleUrls: ['./admin-product-retrieve.component.css']
})
export class AdminProductRetrieveComponent implements OnInit {
```



```
products:Array<Product>=[];
constructor(public ps:ProductService) { }
ngOnInit(): void {
this.findAllProduct();
}
flag:boolean = false;
pid:number =0;
price:number =0;
url:string ="";
findAllProduct() {
this.ps.findAllProduct().subscribe({
next:(result:any)=>this.products=result,
error:(error:any)=>console.log(error),
complete:()=>console.log("completed")
})
}
deleteProduct(pid:number){
//console.log(pid)
this.ps.deleteProductById(pid).subscribe({
next:(result:any)=>console.log(result),
error:(error:any)=>console.log(error),
complete:()=>{
this.findAllProduct();
}
})
}
updateProduct(product:any){
this.flag= true;
this.pid=product.pid;
this.price=product.price;
this.url=product.url;
}
updateDataFromDb(){
let product = {pid:this.pid,price:this.price,url:this.url};
this.ps.updateProduct(product).subscribe({
next:(result:any)=>console.log(result),
error:(error:any)=>console.log(error),
complete:()=>{
this.findAllProduct();
}
})
this.flag=false;
}
}
```

SELENIUM WITH TESTNG

```
package com;

import org.testng.annotations.Test;
//	import org.testng.annotations.AfterClass;
//	import org.testng.annotations.Test;
//	import org.testng.AssertJUnit;
//	import org.testng.annotations.Test;
	import org.testng.asserts.SoftAssert;
//	import java.util.concurrent.TimeUnit;

	import org.openqa.selenium.By;
//import org.openqa.selenium.By.ById;
//import org.openqa.selenium.By.ByXPath;
////import org.openqa.selenium.JavascriptExecutor;
//	import org.openqa.selenium.NoSuchElementException;
	import org.openqa.selenium.WebDriver;
	import org.openqa.selenium.WebElement;
```

```
    import org.openqa.selenium.chrome.ChromeDriver;
//  import org.openqa.selenium.support.ui.FluentWait;
//  import org.openqa.selenium.support.ui.Wait;
//  import org.testng.annotations.AfterClass;
//  import org.testng.annotations.AfterMethod;

    public class driver_your_way_test {

        // Step 1: Initialize the webdriver
        WebDriver driver = null;
        SoftAssert soft = new SoftAssert();

        @Test
        public void initialization_T0() {
            // Step 2: Declare a path and set property for google chrome
driver
            String path =
"C:\\Users\\KIIT\\Downloads\\chromedriver_win32\\chromedriver.exe";
            System.setProperty("webdriver.chrome.driver", path);
            driver = new ChromeDriver();

        }

        @Test(groups = "Chrome", dependsOnMethods =
{ "initialization_T0" })
        public void cross_T1() {
            System.out.println("Testcases Starting...");
            System.out.println();

            // starting chrome
            driver.get("http://localhost:4200/login");

            try {
                Thread.sleep(5000);

            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            WebElement
email=driver.findElement(By.xpath("/html/body/app-root/html/body/div/app-
login/html/body/div/div/form/div/input[1]"));
            email.sendKeys("admin@gmail.com");
            WebElement
password=driver.findElement(By.xpath("/html/body/app-root/html/body/div/app-
login/html/body/div/div/form/div/input[2]"));
            password.sendKeys("admin@123");
            WebElement admin=driver.findElement(By.id("admin1"));
            admin.click();
            WebElement signIn=driver.findElement(By.id("signIn12"));
            signIn.submit();
```

```
                        try {
                                Thread.sleep(5000);

                        } catch (InterruptedException e) {
                                e.printStackTrace();
                        }

                }


                @Test(groups = "Chrome", dependsOnMethods = {"cross_T1"})
                public void cross_T2() {
                        try {
                                Thread.sleep(5000);

                        } catch (InterruptedException e) {
                                e.printStackTrace();
                        }
                        // Clicking Search Button

                        WebElement
addproduct11=driver.findElement(By.xpath("//*[@id=\"add1\"]"));
                        addproduct11.click();
                        WebElement pname=driver.findElement(By.id("pname11"));
                        pname.sendKeys("I10");
                        try {
                                Thread.sleep(3000);

                        } catch (InterruptedException e) {
                                e.printStackTrace();
                        }
                        WebElement price=driver.findElement(By.id("price11"));
                        price.sendKeys("500000");
                        try {
                                Thread.sleep(3000);

                        } catch (InterruptedException e) {
                                e.printStackTrace();
                        }
                        WebElement url=driver.findElement(By.id("url11"));

        url.sendKeys("https://upload.wikimedia.org/wikipedia/commons/thumb/4/44/
Hyundai_i10_1.0_Intro_%28III%29_%E2%80%93_f_03012021.jpg/640px-
Hyundai_i10_1.0_Intro_%28III%29_%E2%80%93_f_03012021.jpg");
                        WebElement store=driver.findElement(By.id("submit11"));
                        try {
                                Thread.sleep(5000);

                        } catch (InterruptedException e) {
                                e.printStackTrace();
                        }
                        store.submit();

                }

}
```

APPLICATION.PROPERTIES

```
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/capstone
spring.datasource.username=root
spring.datasource.password=soumyaranjan@261412
spring.jpa.hibernate.ddl-auto=update
server.port=9090
```