**E-Commerce Business Management Application**

Mr.Pravin Panditrao Chilme , Government Residential Womens Polytechinc Latur

ISTE: 42

LMS id: Ste75

**Technologies used:**

1. **Spring Boot->**Backend framework for building Java-based web applications.
2. **Thymeleaf->**Server-side Java template engine for dynamic HTML generation.
3. **MySQL(Spring JPA**) -> Relational database management system for data storage.
4. **IDE/Tool** -> Spring Tool Suite (Eclipse)
5. **Project Management** --> Maven
6. **Server**  --> Embedded Tomcat Server

**Features:**

1. **Customer Management**: Easily add, update, and delete customer information.
2. **Inventory Management**: Keep track of your inventory items, including stock levels and pricing.
3. **User Authentication**: Secure login and authentication for admin and staff members.
4. **Role-Based Access Control**: Define roles and permissions for different user types.
5. **Thymeleaf Templates**: Utilizes Thymeleaf for dynamic HTML templates.
6. **Database Integration**: Integrated with MySQL for data storage.

**User Modules:**

1. Administrator
2. Ordinary user

**Functional Module:**

1. User Authentication 🡪 Login, Register
2. Product category 🡪 Add category, view categories
3. Product 🡪 Add product, view products, search products by Category

**Installation:**

1. Clone the repository : $ git clone

https://github.com/senthilkumarIRTT/JFSD-Mini-Project/tree/mainor Download the zip

1. Import the project inside Eclipse :
   * Open Eclipse > file > import > maven > existing project > browse >finish .
2. Make sure you are in the Spring\_Boot\_Project-master directory.
3. Configure the database connection in application.properties (check the Database section for more information).
4. Run the project (by running main method is SsBeautyProductsApplication.java) OR right clink on the project > Run As > Spring Boot App.
5. Open <http://localhost:8080/home> in any browser.
6. Now your tables will be created in the databse.

* You have to add one admin data manually to login as admin, So add one admin data.

**Database:**

MySQL can be used as the database for this project. The database connection can be configured in the **application.properties** file, with the appropriate values for the following properties:

spring.datasource.name=beautyproject

spring.datasource.url=jdbc:mysql://localhost:3306/beautyproject

spring.datasource.password=Pchilme\_1984

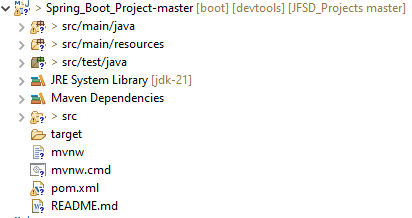
spring.datasource.username=root

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

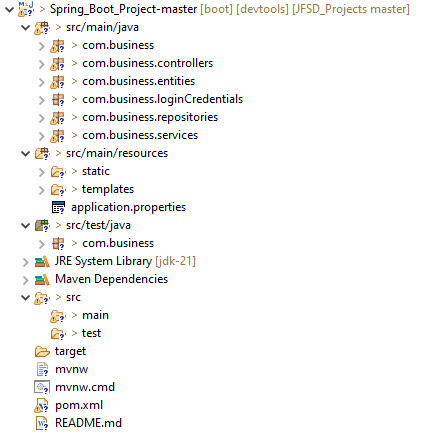
spring.jpa.hibernate.ddl-auto=update

spring.jpa.open-in-view=false

**Project Structure (Folders)**

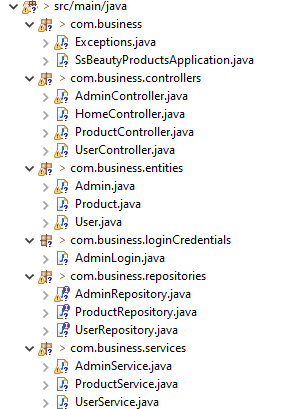


**Project Structure (Sub-Folders)**

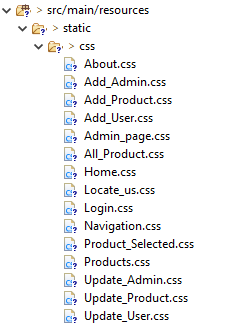


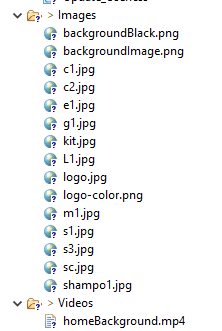
**Project Structure (Files in Sub-Folders)**

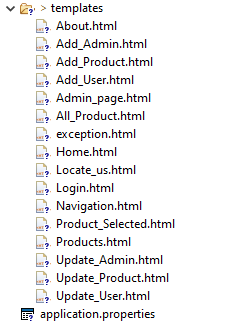
**1. src/main/java**



**2. src/main/resources**







**3. src/test/java**



**Project Source Code:**

**1. pom.xml**

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

<projectxmlns="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>3.1.3</version>

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

</parent>

<groupId>com.business</groupId>

<artifactId>SS\_Beauty\_Products</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>SS\_Beauty\_Products</name>

<description>Building Business related project using spring boot</description>

<properties>

<java.version>17</java.version>

</properties>

<dependencies>

<dependency>

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

<artifactId>spring-boot-starter-thymeleaf</artifactId>

</dependency>

<dependency>

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

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

</dependency>

<dependency>

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

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

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<dependency>

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

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

<scope>test</scope>

</dependency>

<dependency>

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

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

</dependency>

<dependency>

<groupId>com.mysql</groupId>

<artifactId>mysql-connector-j</artifactId>

<scope>runtime</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

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

</plugin>

</plugins>

</build>

</project>

**2. SsBeautyProductsApplication.java [package: com.business]**

package com.business;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SsBeautyProductsApplication {

public static void main(String[] args) {

SpringApplication.run(SsBeautyProductsApplication.class, args);

}

}

**3. AdminController.java**[**package com.business.controllers**]

package com.business.controllers;

import java.util.List;

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

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

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

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

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

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

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

import com.business.entities.Admin;

import com.business.entities.Product;

import com.business.entities.User;

import com.business.loginCredentials.AdminLogin;

import com.business.services.AdminService;

import com.business.services.ProductService;

import com.business.services.UserService;

@Controller

public class AdminController

{

@Autowired

private AdminServiceadminService;

@Autowired

private UserServiceuserService;

@Autowired

private ProductServiceproductService;

private String email;

private User user;

//validating login

@PostMapping("/adminLogin")

public String getallData(@ModelAttribute("adminLogin")AdminLoginlogin,Model model)

{

String email=login.getEmail();

String password=login.getPassword();

if(adminService.validateAdminCredentials(email, password))

{

return "redirect:/services";

}

else{

//return "redirect:/services";

model.addAttribute("error", "Invalid email or password");

return "Login";

}

}

@GetMapping("/services")

public String returnBack(Model model)

{

List<Admin>admins=this.adminService.getAll();

List<User>users=this.userService.getAllUser();

List<Product>products=this.productService.getAllProducts();

model.addAttribute("admins", admins);

model.addAttribute("users", users);

model.addAttribute("products", products);

return "Admin\_Page";

}

//Invoking addAdmin Page

@GetMapping("/addAdmin")

public String addAdminPage()

{

return "Add\_Admin";

}

//Handling AddAdmin

@PostMapping("addingAdmin")

public String addAdmin(@ModelAttribute Admin admin)

{

this.adminService.addAdmin(admin);

return "redirect:/services";

}

//invoking updateAdmin Page

@GetMapping("/updateAdmin/{adminId}")

public String update(@PathVariable("adminId") int id,Model model)

{

Admin admin = this.adminService.getAdmin(id);

model.addAttribute("admin", admin);

return "Update\_Admin";

}

//Handling Update Page

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

public String updateAdmin(@ModelAttribute Admin admin,@PathVariable("id") int id)

{

this.adminService.updateAdmin(admin, id);

return "redirect:/services";

}

//IHandling delete operation

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

public String deleteAdmin(@PathVariable("id") int id)

{

this.adminService.delete(id);

return "redirect:/services";

}

//invoking user page

@GetMapping("/addUser")

public String addUser()

{

return "Add\_User";

}

//adding user

@PostMapping("addingUser")

public String adduser(@ModelAttribute User user)

{

this.userService.addUser(user);

return "redirect:/services";

}

//Invoking UpdateUser Page

@GetMapping("/updateUser/{userId}")

public String updateUserPage(@PathVariable("userId") int id,Model model)

{

User user = this.userService.getUser(id);

model.addAttribute("user", user);

return "Update\_User";

}

//Handling Update Page

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

public String updateUser(@ModelAttribute User user,@PathVariable("id") int id)

{

this.userService.updateUser(user, id);

return "redirect:/services";

}

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

public String deleteUser(@PathVariable("id")int id)

{

this.userService.deleteUser(id);

return "redirect:/services";

}

//handling Product page

@GetMapping("/addProduct")

public String addProduct()

{

return "Add\_Product";

}

@PostMapping("addingProduct")

public String addproduct(@ModelAttribute Product product)

{

this.productService.addProduct(product);

return "redirect:/services";

}

@GetMapping("/updateProduct/{productId}")

public String updateProduct(@PathVariable("productId") int id,Model model)

{

Product product=this.productService.getProduct(id);

System.out.println(product);

model.addAttribute("product", product);

return "Update\_Product";

}

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

public String updateProduct(@ModelAttribute Product product,@PathVariable("id") int id)

{

this.productService.updateproduct(product, id);

return "redirect:/services";

}

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

public String deleteProduct(@PathVariable("id")int id)

{

this.productService.deleteProduct(id);

return "redirect:/services";

}

}

**4. HomeController.java**

package com.business.controllers;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

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

@Controller

public class HomeController

{

@GetMapping("/home")

public String home()

{

return "Home";

}

@GetMapping("/products")

public String products( Model model)

{

return "Products";

}

@GetMapping("/location")

public String location()

{

return "Locate\_us";

}

@GetMapping("/about")

public String about()

{

return "About";

}

@GetMapping("/login")

public String login(Model model)

{

return "Login";

}

}

* 1. **ProductController.java**

package com.business.controllers;

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

import org.springframework.stereotype.Controller;

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

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

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

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

import com.business.entities.Product;

import com.business.services.ProductService;

@Controller

public class ProductController

{

@Autowired

private ProductServiceproductService;

// AddProduct

@PostMapping("/addingProduct")

public String addProduct(@ModelAttribute Product product)

{

this.productService.addProduct(product);

return "redirect:/services";

}

// UpdateProduct

@GetMapping("/updatingProduct/{productId}")

public String updateProduct(@ModelAttribute Product product,@PathVariable("productId") int id)

{

this.productService.updateproduct(product, id);

return "redirect:/services";

}

//DeleteProduct

@GetMapping("/deleteProduct/{productId}")

public String delete(@PathVariable("productId") int id)

{

this.productService.deleteProduct(id);

return "redirect:/services";

}

}

**6. UserController.java**

package com.business.controllers;

import java.util.List;

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

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

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

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

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

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

import com.business.entities.Product;

import com.business.entities.User;

import com.business.repositories.ProductRepository;

import com.business.repositories.UserRepository;

import com.business.services.ProductService;

import com.business.services.UserService;

@Controller

public class UserController

{

@Autowired

private ProductRepositoryproductRepository;

@GetMapping("/userlogin")

public String allProduct(Model model)

{

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

model.addAttribute("products", product);

return "All\_Product";

}

@GetMapping("/selecting")

public String selectedProduct()

{

return "Product\_Selected";

}

}

**7. Admin.java[package com.business.entities]**

package com.business.entities;

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

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.Table;

@Entity

@Table(name="Admin")

public class Admin

{

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int adminId;

private String adminName;

private String adminEmail;

@Value("Admin@123")

private String adminPassword;

private String adminNumber;

public int getAdminId() {

return adminId;

}

public void setAdminId(int adminId) {

this.adminId = adminId;

}

public String getAdminName() {

return adminName;

}

public void setAdminName(String adminName) {

this.adminName = adminName;

}

public String getAdminEmail() {

return adminEmail;

}

public void setAdminEmail(String adminEmail) {

this.adminEmail = adminEmail;

}

public String getAdminPassword() {

return adminPassword;

}

public void setAdminPassword(String adminPassword) {

this.adminPassword = adminPassword;

}

public String getAdminNumber() {

return adminNumber;

}

public void setAdminNumber(String adminNumber) {

this.adminNumber = adminNumber;

}

@Override

public String toString() {

return "Admin [adminId=" + adminId + ", adminName=" + adminName + ", adminEmail=" + adminEmail + ", adminPassword="+ adminPassword + ", adminNumber=" + adminNumber + "]";

}

}

**8. Product.java [package com.business.entities]**

package com.business.entities;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

@Entity

public class Product

{

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int pid;

private String pname;

private double pprice;

private String pdescription;

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 double getPprice() {

return pprice;

}

public void setPprice(double pprice) {

this.pprice = pprice;

}

public String getPdescription() {

return pdescription;

}

public void setPdescription(String pdescription) {

this.pdescription = pdescription;

}

@Override

public String toString() {

return "Product [pid=" + pid + ", pname=" + pname + ", pprice=" + pprice + ", pdescription=" + pdescription

+ "]";

}

}

**9. User.java [package com.business.entities**]

package com.business.entities;

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

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

@Entity

public class User

{

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int UserId;

private String userName;

private String userEmail;

private String userPassword;

private Long userNumber;

public int getUserId() {

return UserId;

}

public void setUserId(int userId) {

UserId = userId;

}

public String getUserName() {

return userName;

}

public void setUserName(String userName) {

this.userName = userName;

}

public String getUserEmail() {

return userEmail;

}

public void setUserEmail(String userEmail) {

this.userEmail = userEmail;

}

public String getUserPassword() {

return userPassword;

}

public void setUserPassword(String userPassword) {

this.userPassword = userPassword;

}

public Long getUserNumber() {

return userNumber;

}

public void setUserNumber(Long userNumber) {

this.userNumber = userNumber;

}

@Override

public String toString() {

return "User [UserId=" + UserId + ", userName=" + userName + ", userEmail=" + userEmail + ", userPassword="

+ userPassword + ", userNumber=" + userNumber + "]";

}

}

**10. AdminLogin.java[package com.business.loginCredentials]**

**package**com.business.loginCredentials;

**publicclass**AdminLogin

{

**private** String email;

**private** String password;

**public** String getEmail() {

**return**email;

}

**publicvoid**setEmail(String email) {

**this**.email = email;

}

**public** String getPassword() {

**return**password;

}

**publicvoid**setPassword(String password) {

**this**.password = password;

}

@Override

**public** String toString() {

**return**"AdminLogin [email=" + email + ", password=" + password + "]";

}

}

**11. AdminRepository.java[package com.business.repositories]**

package com.business.repositories;

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

import org.springframework.data.repository.CrudRepository;

import com.business.entities.Admin;

import com.business.loginCredentials.AdminLogin;

public interface AdminRepository extends CrudRepository<Admin, Integer>

{

public Admin findByadminEmail(String email);

}

**12. ProductRepository.java[package com.business.repositories]**

package com.business.repositories;

import org.springframework.data.repository.CrudRepository;

import com.business.entities.Product;

public interface ProductRepository extends CrudRepository<Product, Integer>

{

public Product findByPname(String name);

}

**13. UserRepository.java[package com.business.repositories]**

package com.business.repositories;

import org.springframework.data.repository.CrudRepository;

import com.business.entities.User;

public interface UserRepository extends CrudRepository<User, Integer>

{

}

**14. AdminService.java[package com.business.services]**

package com.business.services;

import java.util.List;

import java.util.Optional;

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

import org.springframework.stereotype.Component;

import com.business.entities.Admin;

import com.business.loginCredentials.AdminLogin;

import com.business.repositories.AdminRepository;

@Component

public class AdminService

{

@Autowired

private AdminRepositoryadminRepository;

//Get all Admins

public List<Admin>getAll()

{

List<Admin> admins =(List<Admin>) this. adminRepository.findAll();

return admins;

}

//Get Single Admin

public Admin getAdmin(int id)

{

Optional<Admin> optional = this.adminRepository.findById(id);

Admin admin=optional.get();

return admin;

}

// //Update Admin

public void updateAdmin(Admin admin,int id)

{

for(Admin ad:getAll())

{

if(ad.getAdminId()==id)

{

this.adminRepository.save(admin);

}

}

}

//delete Admin

public void delete(int id)

{

this.adminRepository.deleteById(id);

}

//add Admin

public void addAdmin(Admin admin)

{

this.adminRepository.save(admin);

}

public booleanvalidateAdminCredentials(String email,String password)

{

Admin admin=adminRepository.findByadminEmail(email);

if(admin!=null &&admin.getAdminPassword().equals(password))

{

return true;

}

return false;

}

}

**15. ProductService.java[package com.business.services]**

package com.business.services;

import java.util.List;

import java.util.Optional;

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

import org.springframework.stereotype.Component;

import com.business.entities.Product;

import com.business.repositories.ProductRepository;

@Component

public class ProductService

{

@Autowired

private ProductRepositoryproductRepository;

//add product

public void addProduct(Product p)

{

this.productRepository.save(p);

}

//getAll product

public List<Product>getAllProducts()

{

List<Product> products = (List<Product>) this.productRepository.findAll();

return products;

}

//get singal product

public Product getProduct(int id)

{

Optional<Product> optional = this.productRepository.findById(id);

Product product=optional.get();

return product;

}

//update product

public void updateproduct(Product p,int id)

{

p.setPid(id);

Optional<Product> optional = this.productRepository.findById(id);

Product prod=optional.get();

if(prod.getPid()==id)

{

this.productRepository.save(p);

}

}

//delete product

public void deleteProduct(int id)

{

this.productRepository.deleteById(id);

}

}

**16. UserService.java[package com.business.services]**

package com.business.services;

import java.util.List;

import java.util.Optional;

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

import org.springframework.stereotype.Component;

import com.business.entities.User;

import com.business.repositories.UserRepository;

@Component

public class UserService

{

@Autowired

private UserRepositoryuserRepository;

//Get All Users

public List<User>getAllUser()

{

List<User> users = (List<User>) this.userRepository.findAll();

return users;

}

//Get Single User

public User getUser(int id)

{

Optional<User> optional = this.userRepository.findById(id);

User user = optional.get();

return user;

}

//Update

public void updateUser(User user,int id)

{

user.setUserId(id);

this.userRepository.save(user);

}

//delete single User

public void deleteUser(int id)

{

this.userRepository.deleteById(id);

}

//Add User

public void addUser(User user)

{

this.userRepository.save(user);

}

}

**17. application.properties**

spring.datasource.name=beautyproject

spring.datasource.url=jdbc:mysql://localhost:3306/beautyproject

spring.datasource.password=Pchilme\_1984

spring.datasource.username=root

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

spring.jpa.hibernate.ddl-auto=update

spring.jpa.open-in-view=false

**18. SsBeautyProductsApplicationTests.java[Junit 5 test]**

package com.business;

import org.junit.jupiter.api.Test;

import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest

class SsBeautyProductsApplicationTests {

@Test

void contextLoads() {

}

}

**19. Templates**

**(a). About.html**

<!doctype html>

<htmllang="en"xmlns:th="http://www.thymeleaf.org">

<head>

<metacharset="UTF-8" />

<title>SS Beauty : About</title>

<linkrel="stylesheet"href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.2/css/all.min.css">

<linkrel="stylesheet"href=""th:href="@{/css/Navigation.css}" />

<linkrel="stylesheet"href=""th:href="@{/css/About.css}" />

</head>

<body>

<navth:replace="~{Navigation::navbar()}">

</nav>

<divclass="information">

<h1>SS Beauty Product Shop</h1>

<h4>Dr.R.Senthilkumar,</h4>

<h4>Government College of Engineering Erode,</h4>

<h4>Tamilnadu.</h4>

</div>

<divclass="icons">

<h3><iclass="fa-solid fa-phone"style="color: black;"></i> : +91-9940882605</h3>

<h3><iclass="fa-solid fa-envelope"style="color: black;"></i> : Pchilme.optical@gmail.com</h3>

</div>

</body>

</html>

**(b). Add\_Admin.html**

<!doctype html>

<htmllang="en"xmlns:th="http://www.thymeleaf.org">

<head>

<metacharset="UTF-8" />

<title>Add : Admin</title>

<linkrel="stylesheet"href=""th:href="@{/css/Add\_Admin.css}"/>

</head>

<body>

<ahref="/services">Back to Services</a>

<divclass="addAdmin">

<h3style="color: red;">ADD NEW ADMIN/PARTNER</h3>

<formaction=""th:action="@{/addingAdmin}"method="post">

Admin\_Name :

<inputtype="text"name="adminName"placeholder="Enter name..."/><br />

Admin\_Email :

<inputtype="email"name="adminEmail"placeholder="Enter email..."/><br />

Admin\_Number :

<inputtype="text"name="adminNumber"placeholder="Enter number..."/><br />

<buttontype="submit">Add\_Admin</button>

</form>

</div>

</body>

</html>

**(c) Add\_Product.html**

<!doctype html>

<htmllang="en"xmlns:th="http://www.thymeleaf.com">

<head>

<metacharset="UTF-8" />

<title>Document</title>

<linkrel="stylesheet"href=""th:href="@{/css/Add\_Product.css}" />

</head>

<body>

<ahref="/services">Back to Services</a>

<divclass="addProduct">

<h3style="font-weight: bolder;color: seagreen;">ADD NEW PRODUCT :</h3>

<formaction=""th:action="@{/addingProduct}"method="post">

Product\_Name :

<inputtype="text"name="pname"placeholder="Enter Product Name....."/><br />

Product\_Price :

<inputtype="text"name="pprice"placeholder="Enter Product Price...."/><br />

Product\_Desc :<br />

<textareaname="pdescription"id=""cols="15"rows="1"placeholder="Enter Product Description..">

</textarea><br />

<buttontype="submit">Add\_Product</button>

</form>

</div>

</body>

</html>

**(d). Add\_User.html**

<!doctype html>

<htmllang="en"xmlns:th="http://www.thymeleaf.com">

<head>

<metacharset="UTF-8" />

<title>Document</title>

<linkrel="stylesheet"href=""th:href="@{/css/Add\_User.css}" />

</head>

<body>

<ahref="/services">Back to Services</a>

<divclass="addUser">

<h3style="color: red;">ADD NEW USER/CUSTOMER</h3>

<formaction=""th:action="@{/addingUser}"method="post">

<labelfor="name">User\_Name :</label>

<inputtype="text"name="userName" />

<labelfor="email" />User\_Email :

<inputtype="email"name="userEmail" />

<labelfor="telephone">User\_Number :</label>

<inputtype="tel"name="userNumber" />

<inputtype="hidden"name="userPassword"value="User@123" />

</form>

<buttontype="submit">Add\_User</button>

</div>

</body>

</html>

**(e). Admin\_page.html**

<!doctype html>

<htmllang="en"xmlns:th="http://www.thymeleaf.org">

<head>

<metacharset="UTF-8" />

<title>Document</title>

<linkrel="stylesheet"href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.2/css/all.min.css">

<linkrel="stylesheet"href=""th:href="@{/css/Admin\_page.css}" />

</head>

<body>

<aclass="back"href="/home">Back to Home</a>

<center>

<sectionclass="tables">

<divclass="admins">

<h2>Admins :</h2>

<table>

<tr>

<th>Admin\_Id</th>

<th>Admin\_Name</th>

<th>Admin\_Email</th>

<th>Admin\_Number</th>

<th>Actions</th>

</tr>

<trth:each="admin :${admins}">

<tdth:text="${admin.adminId}"></td>

<tdth:text="${admin.adminName}"></td>

<tdth:text="${admin.adminEmail}"></td>

<tdth:text="${admin.adminNumber}"></td>

<td><astyle="color: blue;"href=""th:href="@{'/updateAdmin/'+${admin.adminId}}"><i

class="fa-solid fa-pen"></i></a><span>&nbsp;</span>

<astyle="color: red;"href=""th:href="@{'/deleteAdmin/'+${admin.adminId}}"><i

class="fa-solid fa-trash"></i></a>

</td>

</tr>

</table>

<ahref=""class="btns"th:href="@{/addAdmin}">Add\_Admin</a>

</div>

<divclass="users"th:object="${userObject}">

<h2>Users :</h2>

<table>

<tr>

<th>User\_Id</th>

<th>User\_Name</th>

<th>User\_Email</th>

<th>User\_Number</th>

<th>Actions</th>

</tr>

<trth:each="user :${users}">

<tdth:text="${user.userId}"></td>

<tdth:text="${user.userName}"></td>

<tdth:text="${user.userEmail}"></td>

<tdth:text="${user.userNumber}"></td>

<td><astyle="color: blue;"href=""th:href="@{'/updateUser/'+${user.userId}}"><i

class="fa-solid fa-pen"></i></a><span>&nbsp;</span>

<astyle="color: red;"href=""th:href="@{'/deleteUser/'+${user.userId}}"><i

class="fa-solid fa-trash"></i></a>

</td>

</tr>

</table>

<ahref=""class="btns"th:href="@{'/addUser'}">Add\_User</a>

</div>

<divclass="products">

<h2>Products :</h2>

<table>

<tr>

<th>Product\_Id</th>

<th>Product\_Name</th>

<th>Product\_Price</th>

<th>Product\_Description</th>

<th>Action</th>

</tr>

<trth:each="product :${products}">

<tdth:text="${product.pid}"></td>

<tdth:text="${product.pname}"></td>

<tdth:text="${product.pprice}"></td>

<tdth:text="${product.pdescription}"></td>

<td><astyle="color: blue;"href=""th:href="@{'/updateProduct/'+${product.pid}}">

<iclass="fa-solid fa-pen"></i></a><span>&nbsp;</span>

<astyle="color: red;"href=""th:href="@{'/deleteProduct/'+${product.pid}}">

<iclass="fa-solid fa-trash"></i></a>

</td>

</tr>

</tr>

</table>

<ahref=""class="btns"th:href="@{'/addProduct'}">Add\_Product</a>

</div>

</section>

</center>

</body>

</html>

**(f). All\_Product.html**

<!doctype html>

<htmllang="en"xmlns:th="http://www.thymeleaf.com">

<head>

<metacharset="UTF-8" />

<title>Document</title>

<linkrel="stylesheet"href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.2/css/all.min.css">

<linkrel="stylesheet"href=""th:href="@{/css/Admin\_page.css}" />

<linkrel="stylesheet"href=""th:href="@{/css/All\_Product.css}"/>

</head>

<body>

<aclass="back"href="/home">Back to Home</a>

<center>

<!--<section class="tables">-->

<divclass="products">

<h2> Available Products :</h2>

<table>

<tr>

<th>Product\_Id</th>

<th>Product\_Name</th>

<th>Product\_Price</th>

<th>Product\_Description</th>

<th>Select\_Product</th>

</tr>

<trth:each="product :${products}">

<tdth:text="${product.pid}"></td>

<tdth:text="${product.pname}"></td>

<tdth:text="${product.pprice}"></td>

<tdth:text="${product.pdescription}"></td>

<td><inputtype="checkbox">choice</td>

</tr>

</table>

<ahref=""class="btns"th:href="@{/selecting}">Product\_Selected</a>

</div>

<!--</section>-->

</center>

</body>

</html>

**(g). Home.html**

<!doctype html>

<htmllang="en"xmlns:th="http://www.thymeleaf.com">

<head>

<metacharset="UTF-8" />

<title>SS Beauty : Home</title>

<linkrel="stylesheet"href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.2/css/all.min.css">

<linkrel="stylesheet"href=""th:href="@{/css/Navigation.css}" />

<linkrel="stylesheet"href=""th:href="@{css/Home.css}" />

</head>

<body>

<navth:replace="~{Navigation::navbar()}"></nav>

<divclass="back">

<center><divclass="text">

<h1>WELCOME TO SS BEAUTY PRODUCTS</h1>

<h3style="color: darkgreen;font-family:cursive">Where Beauty Begins! <br /> Explore our exquisite range of beauty products and discover your true radiance.</h3>

</div></center>

<videosrc="/Videos/homeBackground.mp4"loopautoplaymuted></video>

</div>

</body>

</html>

**(h). Locate\_us.html**

<!doctype html>

<htmllang="en">

<head>

<metacharset="UTF-8" />

<title>SS Beauty : Location</title>

<linkrel="stylesheet"href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.2/css/all.min.css">

<linkrel="stylesheet"href=""th:href="@{/css/Navigation.css}" />

<linkrel="stylesheet"href=""th:href="@{/css/Locate\_us.css}" />

</head>

<body>

<navth:replace="~{Navigation::navbar()}"></nav>

<center>

<h1style="color: green; ">Our Location</h1>

<smallstyle="color: red;font-weight: bold;">Internet required..</small>

</center>

<divclass="location">

<iframesrc="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d30773977.6369028!2d60.999633834631844!3d19.687529568953973!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x30635ff06b92b791%3A0xd78c4fa1854213a6!2sIndia!5e0!3m2!1sen!2sin!4v1694664154511!5m2!1sen!2sin"

width="1200"

height="500"style="border:0;"

allowfullscreen="true"

loading="lazy"

referrerpolicy="no-referrer-when-downgrade"></iframe>

</div>

</body>

</html>

**(i). Login.html**

<!doctype html>

<htmllang="en"xmlns:th="http://www.thymeleaf.org">

<head>

<metacharset="UTF-8" />

<title>SS Beauty : Login</title>

<linkrel="stylesheet"href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.2/css/all.min.css">

<linkrel="stylesheet"href=""th:href="@{/css/Navigation.css}" />

<linkrel="stylesheet"href=""th:href="@{/css/Login.css}" />

</head>

<body>

<navth:replace="~{Navigation::navbar()}">

</nav>

<divclass="LoginContainer">

<divclass="admin">

<divclass="content">

<strongstyle="font-size: 16px;margin-bottom: 0px;">BUSINESS</strong>

<h2>For <spanstyle="color:green;">Admins</span></h2>

<formaction=""th:action="@{/adminLogin}"method="post">

<b>Email :</b>

<inputtype="email"name="email"placeholder="Enter email..."class="email"><br />

<b> Password:</b>

<inputtype="password"name="password"placeholder="Enter password"class="password">

<smallstyle="color: red;"th:text="${error}"></small><br />

<buttontype="submit"class="loginButton">Log in</button>

</form>

</div>

</div>

<divclass="user">

<divclass="content">

<h2> For <spanstyle="color:green;">Users</span></h2>

<formaction=""th:action="@{/userlogin}"class="form">

<b>Email :</b>

<inputtype="email"name="userEmail"placeholder="Enter email"><br />

<b> Password:</b>

<inputtype="password"name="userPassword"placeholder="Enter password"requiredmaxlength="8"

minlength="4"class="userPassword">

<smallstyle="color: red;"th:text="${error2}"></small><br />

<buttontype="submit"class="loginButton">Log in</button><br>

<p>I don't have account ?<ahref="register.html"style="color: green;">Register Here</a></p>

</form>

</div>

</div>

</div>

</body>

</html>

**(j). Navigation.html**

<!DOCTYPE html>

<htmllang="en"xmlns:th="http://www.thymeleaf.org">

<head>

<metacharset="UTF-8">

<metaname="viewport"content="width=device-width, initial-scale=1.0">

<title>Home :beauty product</title>

<linkrel="stylesheet"href=""th:href="@{/css/Navigation.css}">

<linkrel="stylesheet"href=""th:href="@{/css/Home.css}" />

<linkrel="stylesheet"href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.2/css/all.min.css">

</head>

<body>

<navth:fragment="navbar()">

<ul>

<imgsrc="/Images/logo.jpg"alt="Owner\_Logo"class="Owner\_Logo">

<ahref="/home">Home</a>

<ahref="/products">Products</a>

<ahref="/location">Locate\_Us<iclass="fa-solid fa-location-crosshairs"></i></a>

<ahref="/about">About\_Us</a>

<ahref="/login">Log\_In<iclass="fa-solid fa-right-to-bracket"></i>&nbsp;&nbsp;&nbsp;&nbsp;</a>

</ul>

</nav>

>

</body>

</html>

**(k). Product\_Selected.html**

<!doctype html>

<htmllang="en"xmlns:th="http://www.thymeleaf.org">

<head>

<metacharset="UTF-8" />

<title>Document</title>

<linkrel="stylesheet"href=""th:href="@{/css/Product\_Selected.css}" />

</head>

<body>

<divclass="success">

<h1>PRODUCT SUCCESSFULLY SELECTED</h1>

<h3>We Will Send Your Product Immediately</h3>

<h3>Thank You For Visiting Our Shop</h3>

<h4>\*\*\*\*\*\*\*\*</h4>

</div>

</body>

</html>

**(l). Products.html**

<!doctype html>

<htmllang="en">

<head>

<metacharset="UTF-8" />

<title>SS Beauty : Products</title>

<linkrel="stylesheet"href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.2/css/all.min.css">

<linkrel="stylesheet"href=""th:href="@{/css/Navigation.css}" />

<linkrel="stylesheet"href=""th:href="@{css/Products.css}" />

</head>

<body>

<navth:replace="~{Navigation::navbar()}"></nav>

<sectionclass="p2">

<section>

<divclass="a1"><imgth:src="@{/images/c1.jpg}"/><h3>Compact Powder</h3><h4>450</h4></div>

<divclass="a3"><imgth:src="@{/images/m1.jpg}"/><h3>Face Mask</h3><h4>230</h4></div>

<divclass="a2"><imgth:src="@{/images/sc.jpg}"/><h3>Kajal</h3><h4>150</h4></div>

</section>

<section>

<divclass="a4"><imgth:src="@{/images/shampo1.jpg}"/><h3>Shampoo</h3><h4>180</h4></div>

<divclass="a7"><imgth:src="@{/images/s1.jpg}"/><h3>SunScreen</h3><h4>250</h4></div>

<divclass="a6"><imgth:src="@{/images/L1.jpg}"/><h3>Lipstick</h3><h4>300</h4></div>

</section>

<section>

<divclass="a8"><imgth:src="@{/images/e1.jpg}"/><h3>FaceWash</h3><h4>200</h4></div>

<divclass="a9"><imgth:src="@{/images/s3.jpg}"/><h3>Oil</h3><h4>225</h4></div>

<divclass="a9"><imgth:src="@{/images/m1.jpg}"/><h3>Foundation</h3><h4>350</h4></div>

</section>

<section>

<divclass="a10"><imgth:src="@{/images/c2.jpg}"/><h3>Conditionar</h3><h4>330</h4></div>

<divclass="a11"><imgth:src="@{/images/kit.jpg}"/><h3>MakeUp kit</h3><h4>500</h4></div>

<divclass="a12"><imgth:src="@{/images/g1.jpg}"/><h3>Gulab Jal</h3><h4>120</h4></div>

</section>

<section>

</section>

</section>

</body>

</html>

**(m). Update\_Admin.html**

<!doctype html>

<htmllang="en"xmlns:th="http://www.thymeleaf.org">

<head>

<metacharset="UTF-8" />

<title>Document</title>

<linkrel="stylesheet"href=""th:href="@{/css/Update\_Admin.css}" />

</head>

<body>

<ahref="/services">Back to Services</a>

<divclass="updateAdmin">

<h3style="color: red;">UPDATE EXISTING ADMIN / PARTNER DETAILS</h3>

<formaction=""th:action="@{'/updatingAdmin/'+${admin.adminId}}"th:object="${admin}">

Admin\_Id :

<inputtype="hidden"name="adminId"th:value="${admin.adminId}"th:text="${admin.adminId}"/><br />

Admin\_Name :

<inputtype="text"name="adminName"th:value="${admin.adminName}"/><br />

Admin\_Number :

<inputtype="tel"name="adminNumber"th:value="${admin.adminNumber}"/><br />

Admin\_Email :

<inputtype="email"name="adminEmail"th:value="${admin.adminEmail}" /><br />

<buttontype="submit">Update\_Admin</button>

</form>

</div>

</body>

</html>

**(n). Update\_Product.html**

<!doctype html>

<htmllang="en"xmlns:th="http://www.thymeleaf.org">

<head>

<metacharset="UTF-8" />

<title>Document</title>

<linkrel="stylesheet"href=""th:href="@{/css/Update\_Product.css}" />

</head>

<body>

<ahref="/services">Back\_To\_Services</a>

<divclass="updateProduct">

<h3style="font-weight: bolder;color: seagreen;">UPDATE EXISTING PRODUCT :</h3>

<formaction=""th:action="@{'/updatingProduct/'+${product.pid}}"th:object="${product}">

Product\_Id:

<inputtype="hidden"th:text="${product.pid}"th:value="${product.pid}" /><br />

Product\_Name :

<inputtype="text"name="pname"th:value="${product.pname}" /><br />

Product\_Price :

<inputtype="text"name="pprice"th:value="${product.pprice}" /><br />

Product\_Desc :<br />

<textareaname="pdescription"th:value="${product.pdescription}"id=""cols="10"rows="1">

</textarea><br />

<buttontype="submit">Update\_Product</button>

</form>

</div>

</body>

</html>

**(o). Update\_User.hml**

<!doctype html>

<htmllang="en"xmlns:th="http://www.thymeleaf.org">

<head>

<metacharset="UTF-8" />

<title>Admin : Update</title>

<linkrel="stylesheet"href=""th:href="@{/css/Update\_User.css}" />

</head>

<body>

<ahref="/services"> Back to Services</a>

<divclass="updateUser">

<h3style="color: red;"> UPDATE EXISTING USER/CUSTOMER </h3>

<formaction=""th:action="@{'/updatingUser/'+${user.userId}}"th:object="${user}">

User\_Id :

<inputtype="hidden"name="userId"th:value="${user.userId}"th:text="${user.userId}"/><br />

User\_Name :

<inputtype="text"name="userName"th:value="${user.userName}"/>

User\_Emai :

<inputtype="email"name="userEmail"th:value="${user.userEmail}" />

User\_Number :

<inputtype="tel"name="userNumber"th:value="${user.userNumber}"/>

<buttontype="submit">Update\_User</button>

</form>

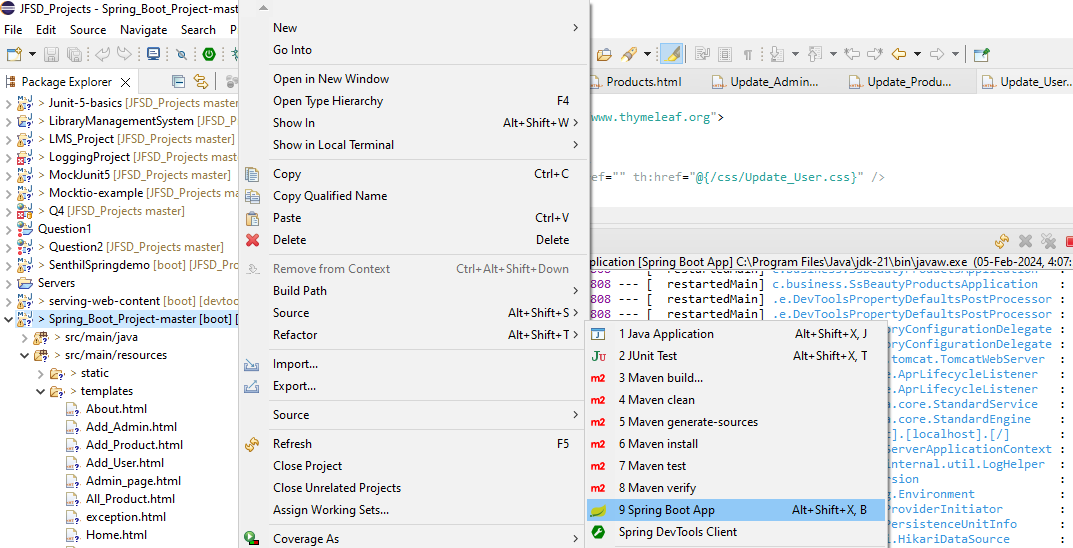
</div>

</body>

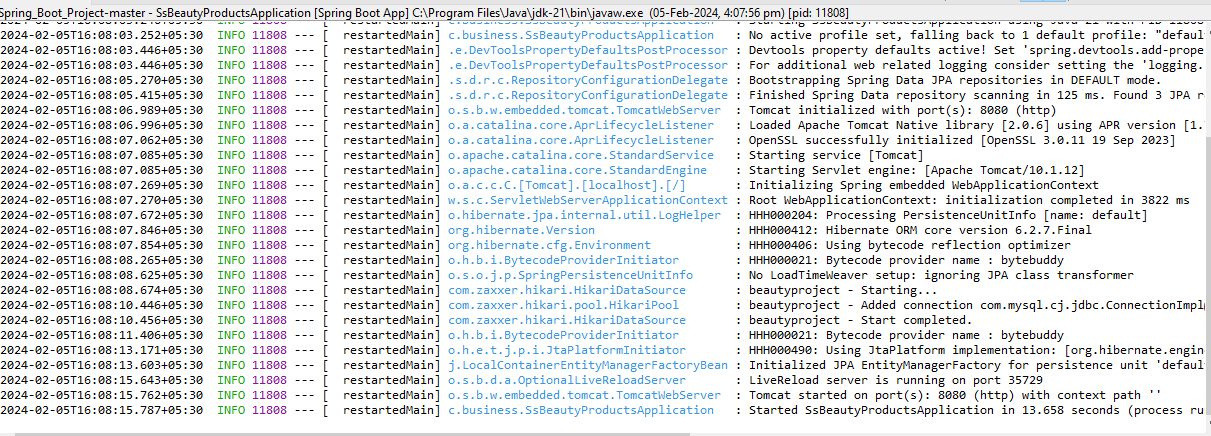
</html>

**Run the Project**

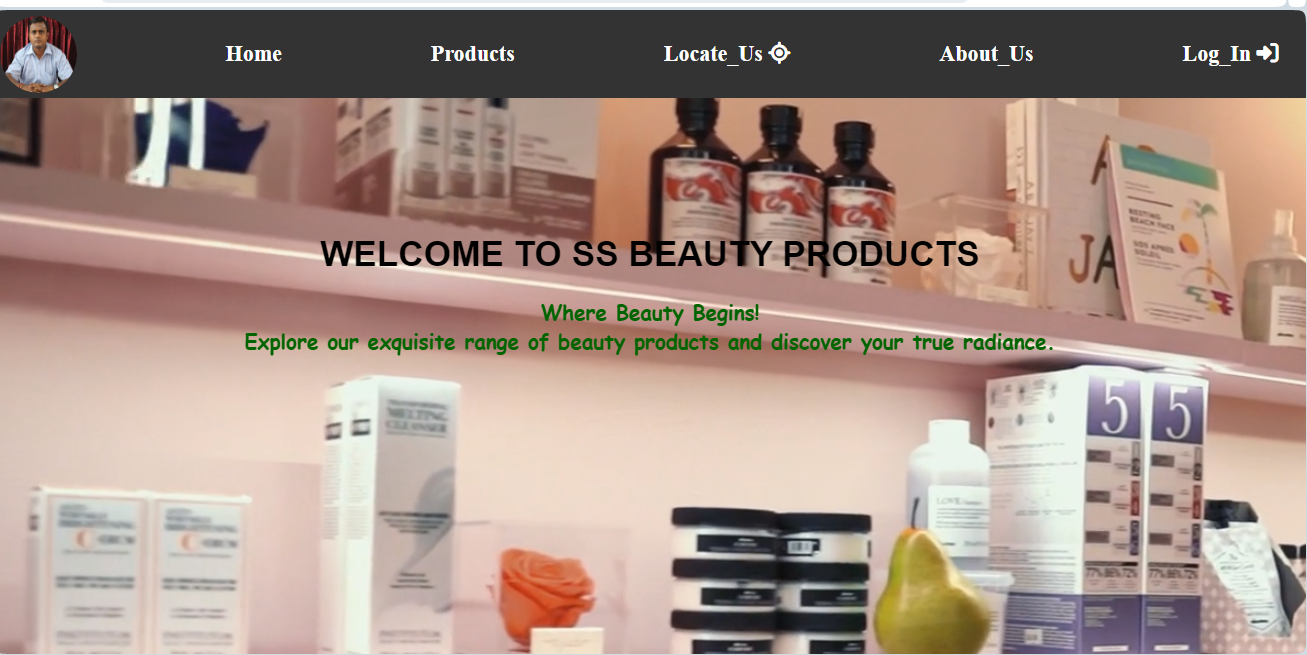
**Step 1:**



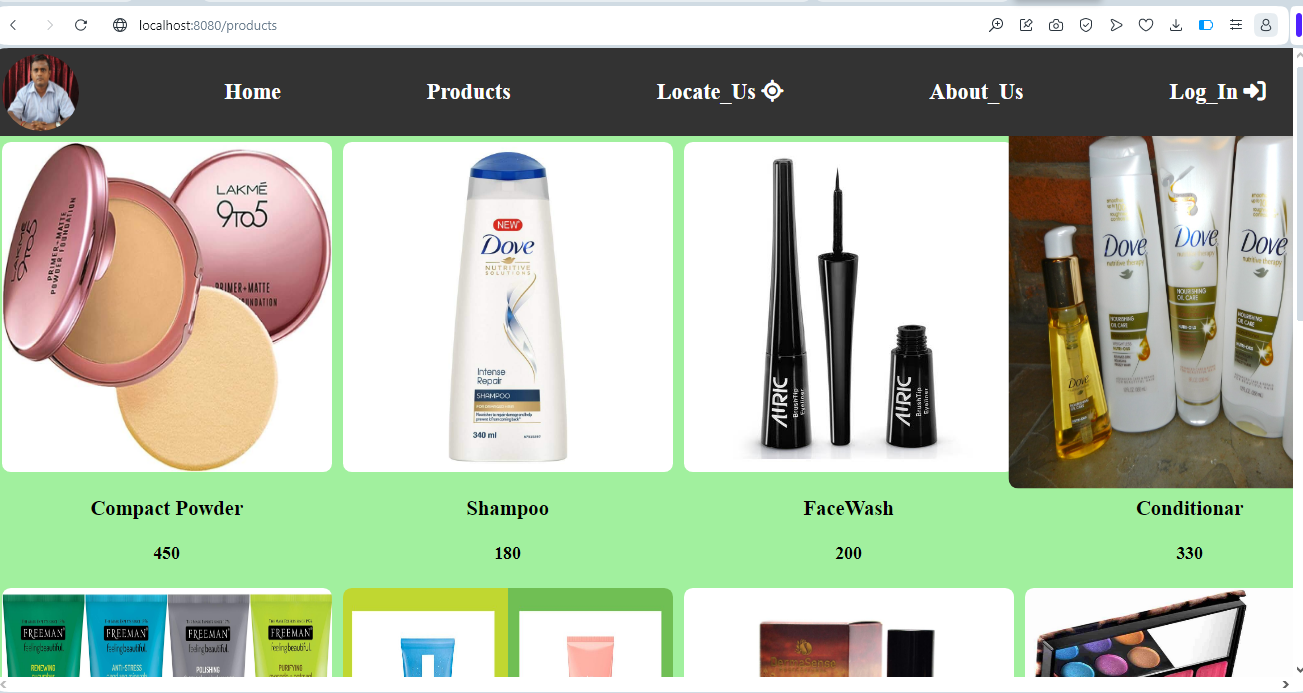
Step 2: No Error



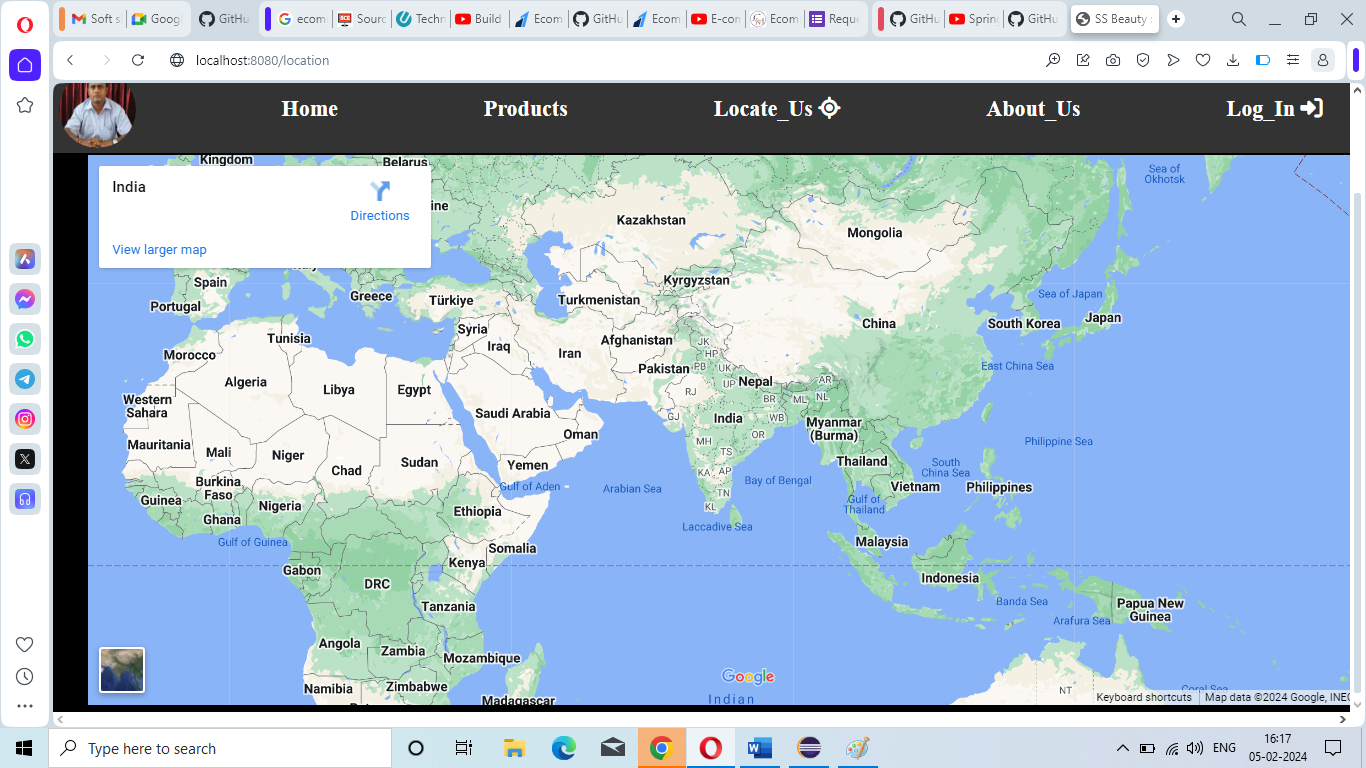
Step 3: Go to the page in any browser :<http://localhost:8080/home>



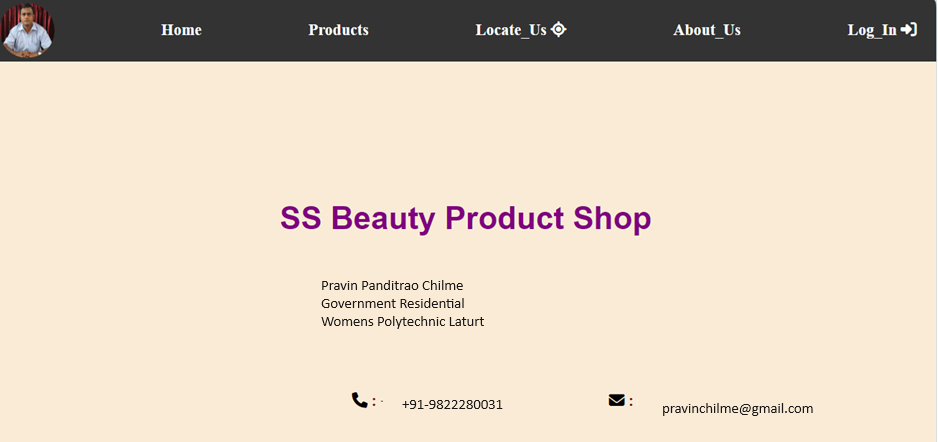
Step 4: Go to the page <http://localhost:8080/products> or Products



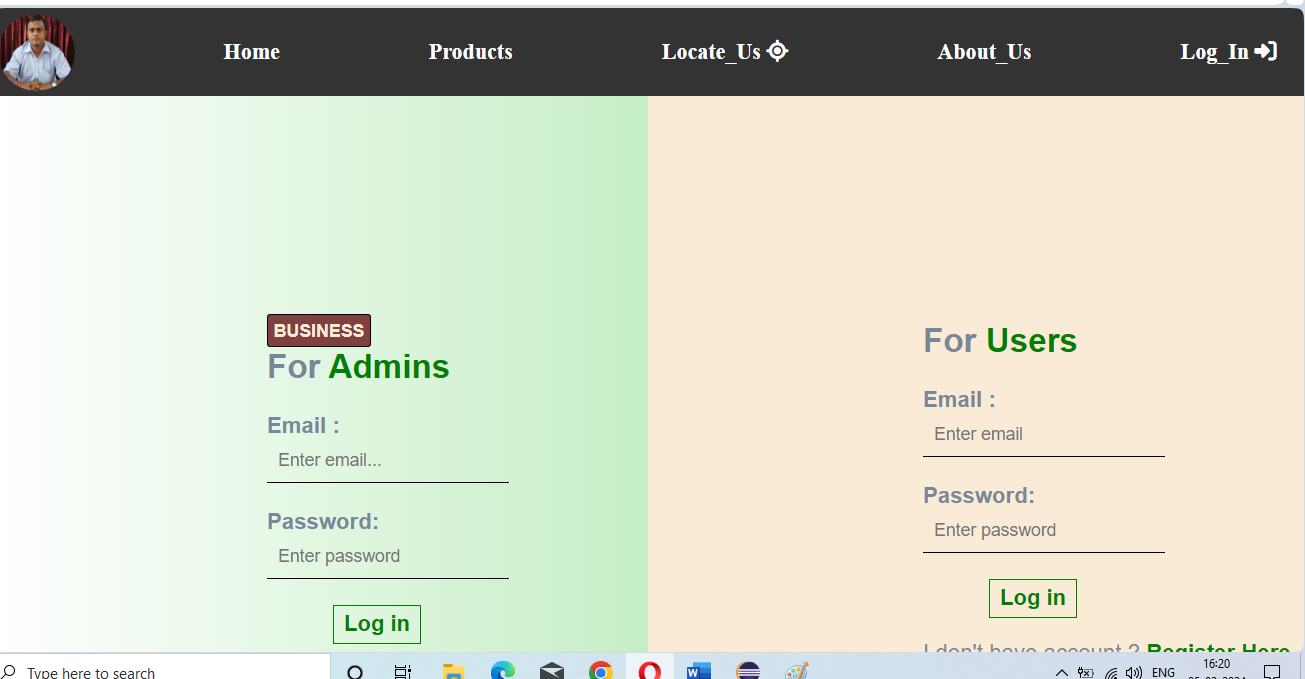
Step 5: Locate us: http://localhost:8080/location



Step 6: <http://localhost:8080/about>

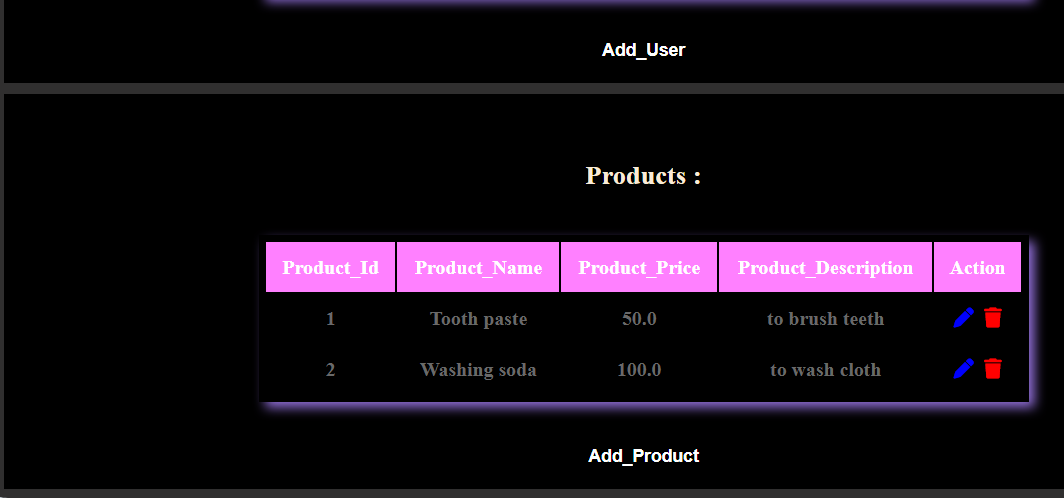


Step 7 : Login: http://localhost:8080/login

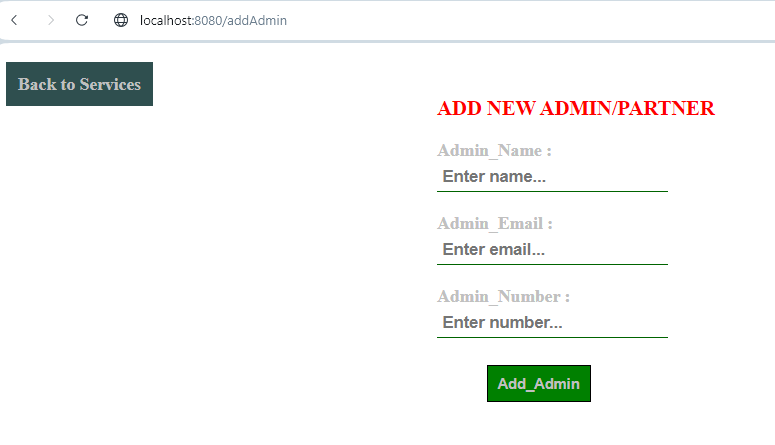


Step 8: Services: http://localhost:8080/services

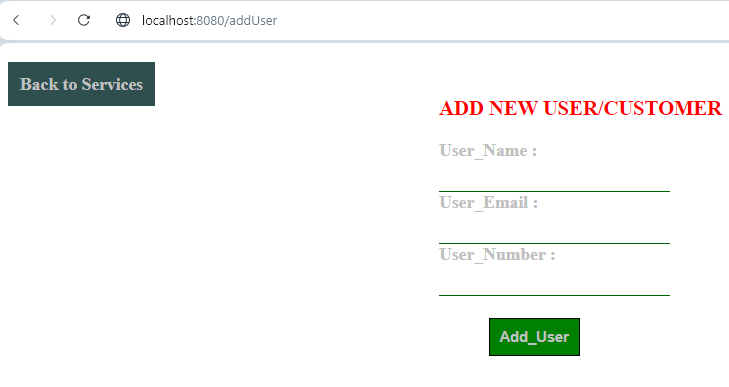




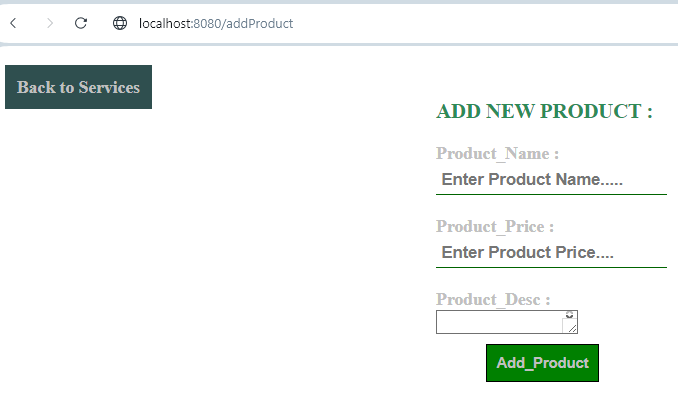
Step 9: http://localhost:8080/addAdmin



Step 10: http://localhost:8080/addUser

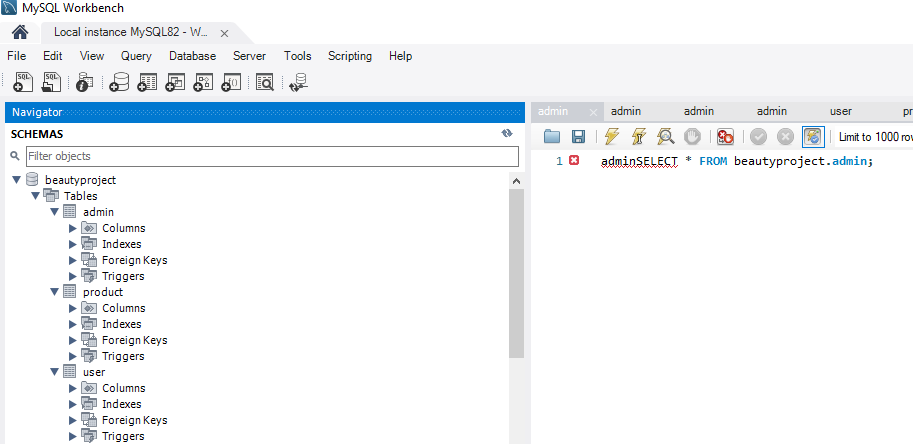


Step 11: http://localhost:8080/addProduct

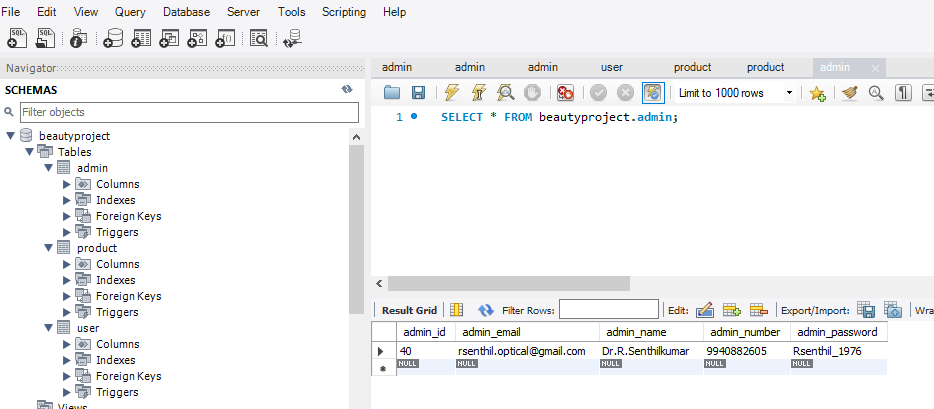


**MySQL Workbench**

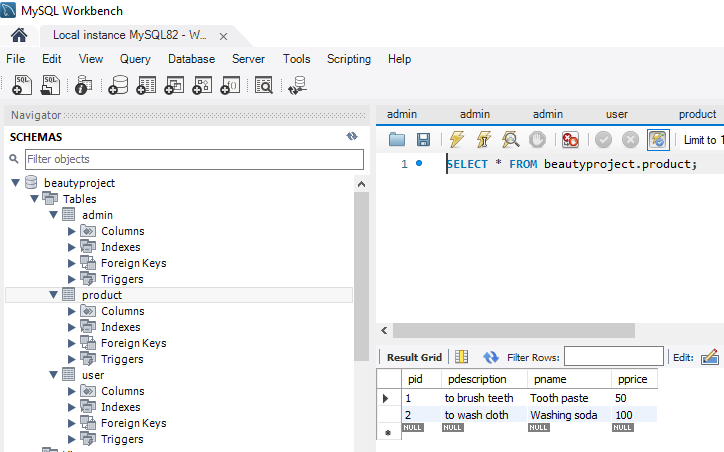
Beautyproject schema



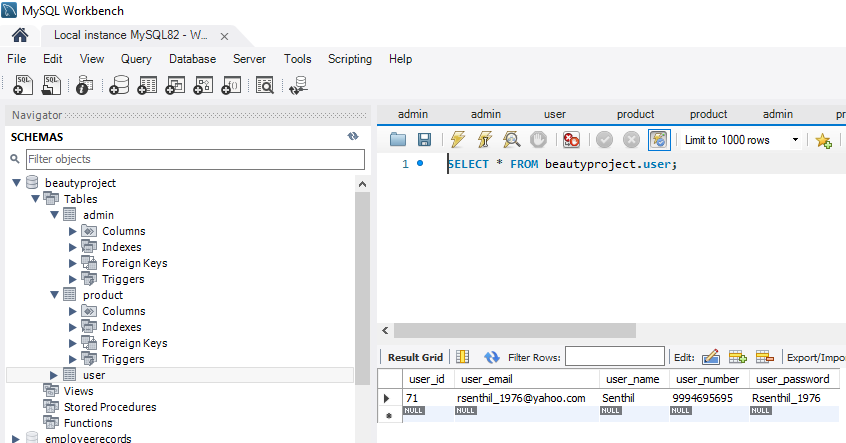
**Admin table**



**Product table**



**User table**



Report and project files are available in github:

https://github.com/pravinchilme/JFSD\_Project.git