

# E-Commerce Business Management Application

Dr.R.Senthilkumar, Government College of Engineering Erode

ISTE: 40

LMS id: Std71

## 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:

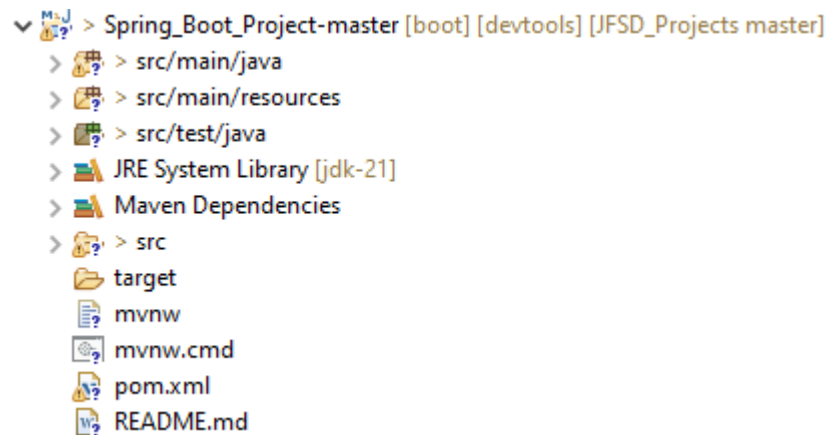
1. Clone the repository : \$ git clone <https://github.com/senthilkumarIRTT/JFSD-Mini-Project/tree/main> or Download the zip
2. Import the project inside Eclipse :
  - Open Eclipse > file > import > maven > existing project > browse > finish .
3. Make sure you are in the Spring\_Boot\_Project-master directory.
4. Configure the database connection in application.properties (check the Database section for more information).
5. Run the project (by running main method is SsBeautyProductsApplication.java) OR right click on the project > Run As > Spring Boot App.
6. Open <http://localhost:8080/home> in any browser.
7. Now your tables will be created in the database.
  - You have to add one admin data manually to login as admin, So add one admin data.

## Database:

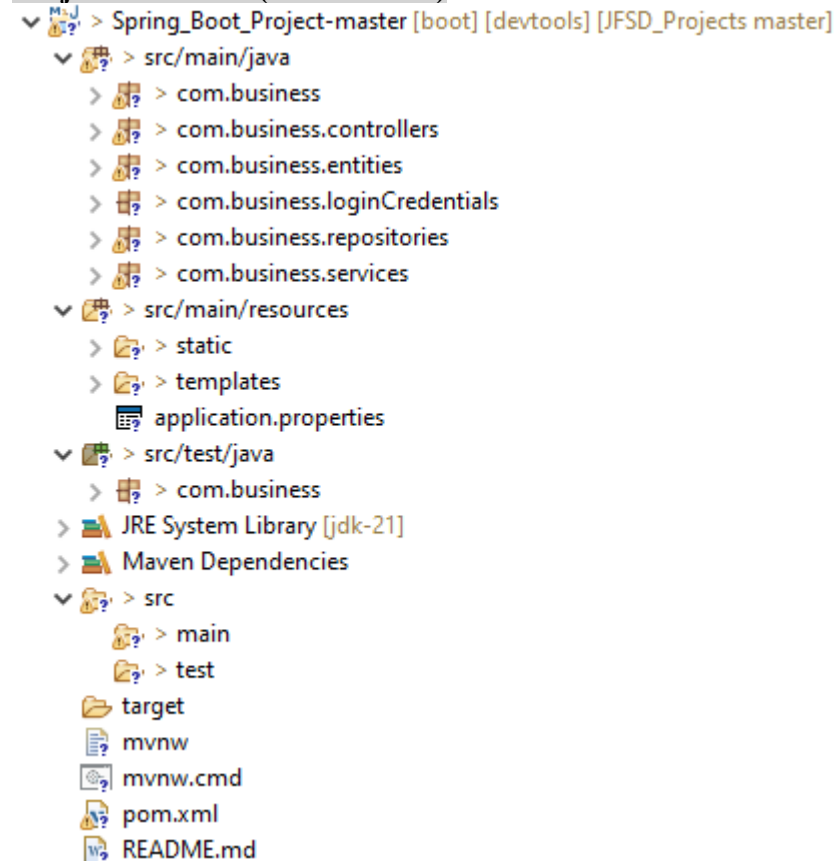
MySQL can be used as the database for this project. The database connection can be configured in the **application.properties** file, with the appropriate values for the following properties:

```
spring.datasource.name=beautyproject
spring.datasource.url=jdbc:mysql://localhost:3306/beautyproject
spring.datasource.password=Rsenthil_1976
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)





















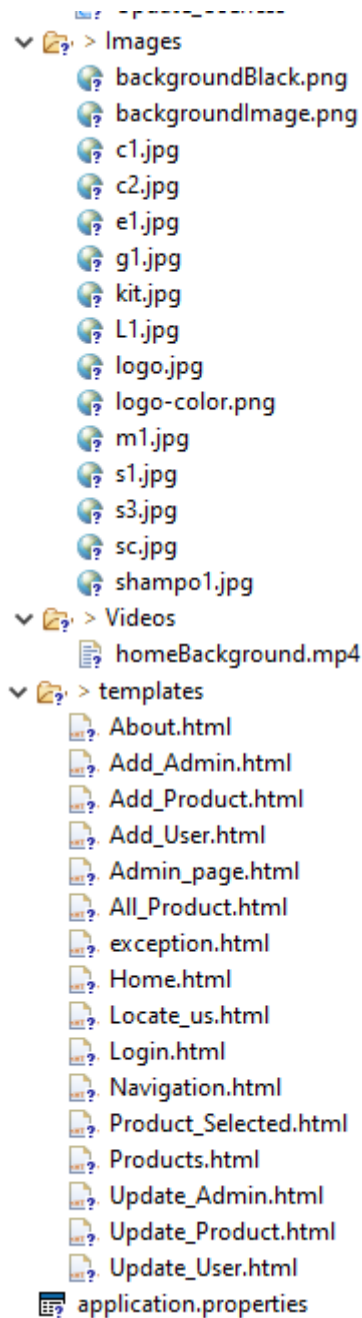
## Project Structure (Files in Sub-Folders)

### 1. src/main/java

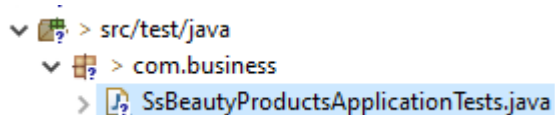
- ▼  > src/main/java
  - ▼  > com.business
    - >  Exceptions.java
    - >  SsBeautyProductsApplication.java
  - ▼  > com.business.controllers
    - >  AdminController.java
    - >  HomeController.java
    - >  ProductController.java
    - >  UserController.java
  - ▼  > com.business.entities
    - >  Admin.java
    - >  Product.java
    - >  User.java
  - ▼  > com.business.loginCredentials
    - >  AdminLogin.java
  - ▼  > com.business.repositories
    - >  AdminRepository.java
    - >  ProductRepository.java
    - >  UserRepository.java
  - ▼  > com.business.services
    - >  AdminService.java
    - >  ProductService.java
    - >  UserService.java

### 2. src/main/resources

- ▼  > src/main/resources
  - ▼  > static
    - ▼  > css
      -  About.css
      -  Add\_Admin.css
      -  Add\_Product.css
      -  Add\_User.css
      -  Admin\_page.css
      -  All\_Product.css
      -  Home.css
      -  Locate\_us.css
      -  Login.css
      -  Navigation.css
      -  Product\_Selected.css
      -  Products.css
      -  Update\_Admin.css
      -  Update\_Product.css
      -  Update\_User.css



### 3. src/test/java



## Project Source Code:

### 1. pom.xml

```
<?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>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 AdminService adminService;
    @Autowired
    private UserService userService;
}

```

```

@Autowired
private ProductService productService;

private String email;
private User user;

//validating login

@PostMapping("/adminLogin")
public String getAllData(@ModelAttribute("adminLogin")AdminLogin login,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";
    }
    //Handling 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";
    }
}
```

#### 5. 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 ProductService productService;

    // 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 ProductRepository productRepository;

    @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 + "]\n";
}
}

```

## 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;
public class AdminLogin
{
    private String email;
    private String password;
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
    public String getPassword() {
        return password;
    }
    public void 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 AdminRepository adminRepository;
    //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 boolean validateAdminCredentials(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 ProductRepository productRepository;

    //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 UserRepository userRepository;
    //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=Rsenthil_1976
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>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8" />
    <title>SS Beauty : About</title>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.2/css/all.min.css">
    <link rel="stylesheet" href="" th:href="@{/css/Navigation.css}" />
    <link rel="stylesheet" href="" th:href="@{/css/About.css}" />
</head>
<body>
    <nav th:replace="~{Navigation::navbar()}">
    </nav>

    <div class="information">

        <h1>SS Beauty Product Shop</h1>
        <h4>Dr.R.Senthilkumar,</h4>
        <h4>Government College of Engineering Erode,</h4>
        <h4>Tamilnadu.</h4>

    </div>
    <div class="icons">

        <h3><i class="fa-solid fa-phone" style="color: black;"></i> : +91-
9940882605</h3>
        <h3><i class="fa-solid fa-envelope" style="color: black;"></i> :
rsenthil.optical@gmail.com</h3>

    </div>
</body>
</html>

```

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

```

<!doctype html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8" />
    <title>Add : Admin</title>
    <link rel="stylesheet" href="" th:href="@{/css/Add_Admin.css}" />

```

```

</head>
<body>
    <a href="/services">Back to Services</a>
    <div class="addAdmin">
        <h3 style="color: red;">ADD NEW ADMIN/PARTNER</h3>
        <form action="" th:action="@{/addingAdmin}" method="post">
            Admin_Name :
            <input type="text" name="adminName" placeholder="Enter
name..." /><br />
            Admin_Email :
            <input type="email" name="adminEmail" placeholder="Enter
email..." /><br />
            Admin_Number :
            <input type="text" name="adminNumber" placeholder="Enter
number..." /><br />

            <button type="submit">Add_Admin</button>
        </form>
    </div>
</body>
</html>

```

### (c) Add\_Product.html

```

<!doctype html>
<html lang="en" xmlns:th="http://www.thymeleaf.com">
<head>
    <meta charset="UTF-8" />
    <title>Document</title>
    <link rel="stylesheet" href="" th:href="@{/css/Add_Product.css}" />
</head>
<body>
    <a href="/services">Back to Services</a>
    <div class="addProduct">
        <h3 style="font-weight: bolder;color: seagreen;">ADD NEW PRODUCT :</h3>
        <form action="" th:action="@{/addingProduct}" method="post">
            Product_Name :
            <input type="text" name="pname" placeholder="Enter Product Name....." />
<br />
            Product_Price :
            <input type="text" name="pprice" placeholder="Enter Product Price...." /><br />
            Product_Desc : <br />
            <textarea name="pdescription" id="" cols="15" rows="1" placeholder="Enter Product
Description..">
</textarea><br />

```

```

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

    </form>
</div>
</body>
</html>

```

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

```

<!doctype html>
<html lang="en" xmlns:th="http://www.thymeleaf.com">

<head>
    <meta charset="UTF-8" />
    <title>Document</title>
    <link rel="stylesheet" href="" th:href="@{/css/Add_User.css}" />
</head>

<body>
    <a href="/services">Back to Services</a>
    <div class="addUser">
        <h3 style="color: red;">ADD NEW USER/CUSTOMER</h3>
        <form action="" th:action="@{/addingUser}" method="post">
            <label for="name">User_Name :</label>
            <input type="text" name="userName" />
            <label for="email">User_Email :
            <input type="email" name="userEmail" />
            <label for="telephone">User_Number :</label>
            <input type="tel" name="userNumber" />
            <input type="hidden" name="userPassword" value="User@123" />
        </form>
        <button type="submit">Add_User</button>
    </div>
</body>

</html>

```

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

```

<!doctype html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8" />
    <title>Document</title>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.2/css/all.min.css">
    <link rel="stylesheet" href="" th:href="@{/css/Admin_page.css}" />

```

```

</head>
<body>
  <a class="back" href="/home">Back to Home</a>
  <center>
    <section class="tables">
      <div class="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>
          <tr th:each="admin :${ admins}">
            <td th:text="${ admin.adminId}"></td>
            <td th:text="${ admin.adminName}"></td>
            <td th:text="${ admin.adminEmail}"></td>
            <td th:text="${ admin.adminNumber}"></td>
            <td><a style="color: blue;" href="
th:href="@ {'/updateAdmin/'+${ admin.adminId} }"><i
class="fa-solid fa-
pen"></i></a><span>&nbsp;</span>
<a style="color: red;" href="
th:href="@ {'/deleteAdmin/'+${ admin.adminId} }"><i
class="fa-solid fa-
trash"></i></a>
            </td>
          </tr>
        </table>
        <a href="" class="btns"
th:href="@{/addAdmin}">Add_Admin</a>
      </div>
      <div class="users" th:object="${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>
          <tr th:each="user :${ users}">
            <td th:text="${ user.userId}"></td>
            <td th:text="${ user.userName}"></td>
            <td th:text="${ user.userEmail}"></td>
            <td th:text="${ user.userNumber}"></td>

```



```

                                <td><a      style="color:    blue;"      href=""
th:href="@ {'/updateUser/'+${user.userId} }"><i
                                class="fa-solid          fa-
pen"></i></a><span>&nbsp;</span>
                                <a      style="color:    red;"      href=""
th:href="@ {'/deleteUser/'+${user.userId} }"> <i
                                class="fa-solid          fa-
trash"></i></a>
                                </td>
                                </tr>
                                </table>
                                <a      href=""
th:href="@ {'/addUser'}">Add_User</a>
                                class="btns"
                                </div>

                                <div class="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>
                                <tr th:each="product :${products}">
                                <td th:text="${product.pid} "></td>
                                <td th:text="${product.pname} "></td>
                                <td th:text="${product.pprice} "></td>
                                <td th:text="${product.pdescription} "></td>
                                <td><a      style="color:    blue;"href=""
th:href="@ {'/updateProduct/'+${product.pid} }">
                                <i      class="fa-solid          fa-
pen"></i></a><span>&nbsp;</span>
                                <a      style="color:    red;"href=""
th:href="@ {'/deleteProduct/'+${product.pid} }">
                                <i      class="fa-solid          fa-
trash"></i></a>
                                </td>
                                </tr>
                                </table>
                                <a      href=""
th:href="@ {'/addProduct'}">Add_Product</a>
                                class="btns"
                                </div>
                                </section>

```

```

</center>

</body>
</html>

(f). All_Product.html

<!doctype html>
<html lang="en" xmlns:th="http://www.thymeleaf.com">
<head>
    <meta charset="UTF-8" />
    <title>Document</title>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.2/css/all.min.css">
    <link rel="stylesheet" href="" th:href="@{/css/Admin_page.css}" />
    <link rel="stylesheet" href="" th:href="@{/css/All_Product.css}"/>
</head>
<body>

    <a class="back" href="/home">Back to Home</a>
    <center>
        <!--<section class="tables">-->

<div class="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>
        <tr th:each="product :${products}">
            <td th:text="${product.pid}"></td>
            <td th:text="${product.pname}"></td>
            <td th:text="${product.pprice}"></td>
            <td th:text="${product.pdescription}"></td>
            <td><input type="checkbox">choice</td>
        </tr>
    </table>
    <a href="" class="btns"
th:href="@{/selecting}">Product_Selected</a>

</div>

    <!--</section>-->
    </center>

```

```
</body>
</html>
```

### (g). Home.html

```
<!doctype html>
<html lang="en" xmlns:th="http://www.thymeleaf.com">

<head>
    <meta charset="UTF-8" />
    <title>SS Beauty : Home</title>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.2/css/all.min.css">
    <link rel="stylesheet" href="" th:href="@{/css/Navigation.css}" />
    <link rel="stylesheet" href="" th:href="@{/css/Home.css}" />
</head>

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

    <div class="back">
        <center><div class="text">
            <h1>WELCOME TO SS BEAUTY PRODUCTS</h1>
            <h3 style="color: darkgreen;font-family:cursive">Where Beauty Begins! <br
/> Explore our exquisite range of beauty products and discover your true radiance.</h3>
        </div></center>
        <video src="/Videos/homeBackground.mp4" loop autoplay muted></video>
    </div>
</body>

</html>
```

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

```
<!doctype html>
<html lang="en">

<head>
    <meta charset="UTF-8" />
    <title>SS Beauty : Location</title>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.2/css/all.min.css">
    <link rel="stylesheet" href="" th:href="@{/css/Navigation.css}" />
    <link rel="stylesheet" href="" th:href="@{/css/Locate_us.css}" />
</head>

<body>
    <nav th:replace="~{Navigation::navbar()}"></nav>
    <center>
```

```

        <h1 style="color: green; ">Our Location</h1>
        <small style="color: red;font-weight: bold;">Internet required..</small>
    </center>
    <div class="location">
        <iframe
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d30773977.6369028!2d
60.999633834631844!3d19.687529568953973!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3
m3!1m2!1s0x30635ff06b92b791%3A0xd78c4fa1854213a6!2sIndia!5e0!3m2!1sen!2sin!4v1
694664154511!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>
<html lang="en" xmlns:th="http://www.thymeleaf.org">

<head>
    <meta charset="UTF-8" />
    <title>SS Beauty : Login</title>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.2/css/all.min.css">
    <link rel="stylesheet" href="" th:href="@{/css/Navigation.css}" />
    <link rel="stylesheet" href="" th:href="@{/css/Login.css}" />
</head>

<body>
    <nav th:replace="~{Navigation::navbar()}">
    </nav>
    <div class="LoginContainer">
        <div class="admin">
            <div class="content">
                <strong style="font-size: 16px;margin-bottom:
0px;">BUSINESS</strong>
                <h2>For <span style="color:green;">Admins</span></h2>
                <form action="" th:action="@{/adminLogin}" method="post">
                    <b>Email :</b>
                    <input type="email" name="email" placeholder="Enter email..." class="email">
                <br />

```

```

        <b> Password:</b>
        <input type="password" name="password"
placeholder="Enter password" class="password">

        <small style="color: red;"
th:text="{error}"></small><br />
        <button type="submit" class="loginButton">Log
in</button>
    </form>
</div>
<div>
<div class="user">
    <div class="content">
        <h2> For <span style="color:green;">Users</span></h2>
        <form action="" th:action="@{/userlogin}" class="form">

            <b>Email :</b>
            <input type="email" name="userEmail"
placeholder="Enter email"><br />

            <b> Password:</b>
            <input type="password" name="userPassword"
placeholder="Enter password" required maxlength="8"
minlength="4" class="userPassword">

            <small style="color: red;"
th:text="{error2}"></small><br />
            <button type="submit" class="loginButton">Log
in</button><br>

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

```

## (j). Navigation.html

```

<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home :beauty product</title>

```



```

<head>
    <meta charset="UTF-8" />
    <title>SS Beauty : Products</title>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.2/css/all.min.css">
    <link rel="stylesheet" href="" th:href="@{/css/Navigation.css}" />
    <link rel="stylesheet" href="" th:href="@{/css/Products.css}" />

</head>

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

    <section class="p2">
<section>

        <div class="a1"><h3>Compact
Powder</h3><h4>450</h4></div>

        <div class="a3"><h3>Face
Mask</h3><h4>230</h4></div>
        <div class="a2"><h3>Kajal</h3><h4>150</h4></div>

</section>
<section>
        <div class="a4"><h3>Shampoo</h3><h4>180</h4></div>
        <div class="a7"><h3>SunScreen</h3><h4>250</h4></div>
        <div class="a6"><h3>Lipstick</h3><h4>300</h4></div>
</section>
<section>
        <div class="a8"><h3>FaceWash</h3><h4>200</h4></div>
        <div class="a9"><h3>Oil</h3><h4>225</h4></div>
        <div class="a9"><h3>Foundation</h3><h4>350</h4></div>
</section>
<section>
        <div class="a10"><h3>Conditionar</h3><h4>330</h4></div>
        <div class="a11"><h3>MakeUp
kit</h3><h4>500</h4></div>
        <div class="a12"><h3>Gulab
Jal</h3><h4>120</h4></div>

```

```
</section>
<section>
```

```
</section>
</section>
```

```
</body>
```

```
</html>
```

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

```
<!doctype html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8" />
    <title>Document</title>
    <link rel="stylesheet" href="" th:href="@{/css/Update_Admin.css}" />
</head>
<body>
    <a href="/services">Back to Services</a>
    <div class="updateAdmin">
        <h3 style="color: red;">UPDATE EXISTING ADMIN /
PARTNER DETAILS</h3>
        <form action="" th:action="@{/updatingAdmin/'${admin.adminId}'}"
th:object="${admin}">
            Admin_Id :
            <input type="hidden" name="adminId" th:value="${admin.adminId}"
th:text="${admin.adminId}"/><br />
            Admin_Name :
            <input type="text" name="adminName"
th:value="${admin.adminName}"/><br />
            Admin_Number :
            <input type="tel" name="adminNumber"
th:value="${admin.adminNumber}"/> <br />
            Admin_Email :
            <input type="email" name="adminEmail"
th:value="${admin.adminEmail}"/> <br />

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

        </form>
    </div>
</body>
</html>
```

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



```

<!doctype html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">

<head>
    <meta charset="UTF-8" />
    <title>Document</title>
    <link rel="stylesheet" href="" th:href="@{/css/Update_Product.css}" />
</head>

<body>
    <a href="/services">Back_To_Services</a>
    <div class="updateProduct">
        <h3 style="font-weight: bolder;color: seagreen;">UPDATE EXISTING
PRODUCT :</h3>
        <form action="" th:action="@{/updatingProduct/'+${product.pid}"}"
th:object="${product}">
            Product_Id:
            <input type="hidden" th:text="${product.pid}"
th:value="${product.pid}" /><br />
            Product_Name :
            <input type="text" name="pname" th:value="${product.pname}" />
<br />
            Product_Price :
            <input type="text" name="pprice" th:value="${product.pprice}" /><br
/>
            Product_Desc : <br />
            <textarea name="pdescription" th:value="${product.pdescription}"
id="" cols="10" rows="1">

            </textarea><br />
            <button type="submit">Update_Product</button>

        </form>
    </div>
</body>

</html>

```

#### (o). Update\_User.html

```

<!doctype html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8" />
    <title>Admin : Update</title>
    <link rel="stylesheet" href="" th:href="@{/css/Update_User.css}" />
</head>
<body>
    <a href="/services"> Back to Services</a>

```

```

<div class="updateUser">
    <h3 style="color: red;"> UPDATE EXISTING USER/CUSTOMER </h3>
    <form          action=""          th:action="@{/updatingUser/'+${user.userId}}"
th:object="${user}" ">
        User_Id :
        <input      type="hidden"    name="userId"    th:value="${user.userId}"
th:text="${user.userId}" /><br />
        User_Name :
        <input type="text" name="userName" th:value="${user.userName}" />
        User_Emai :
        <input type="email" name="userEmail" th:value="${user.userEmail}"
/>

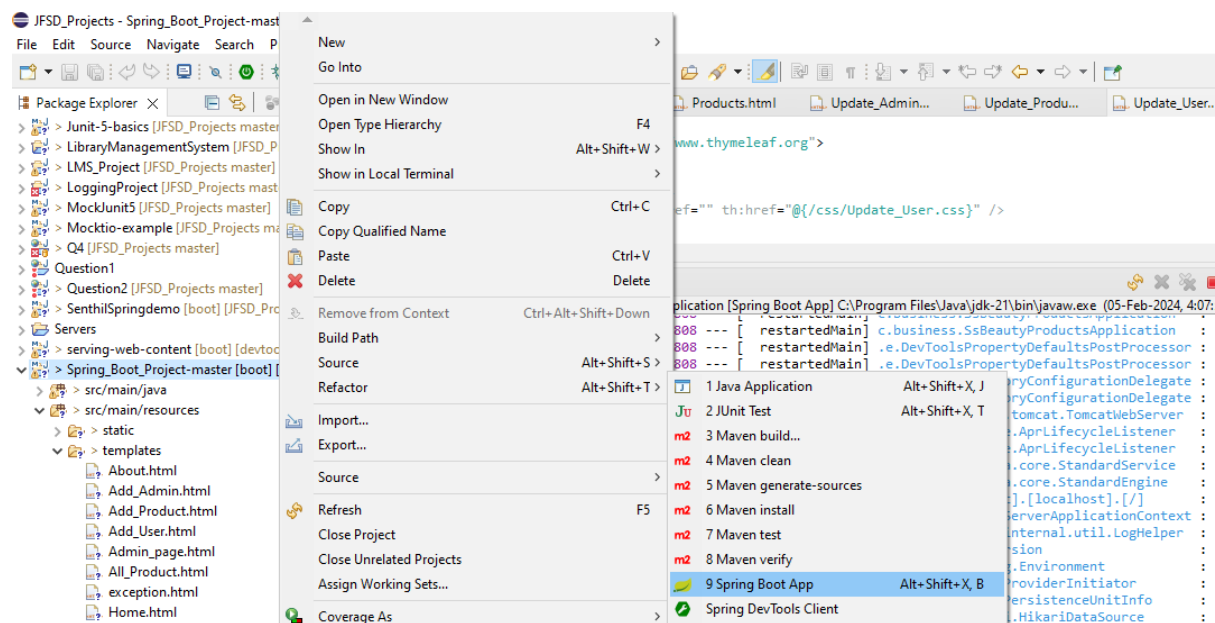
        User_Number :
        <input                                type="tel"                                name="userNumber"
th:value="${user.userNumber}" />

        <button type="submit">Update_User</button>
    </form>
</div>
</body>
</html>

```

## Run the Project

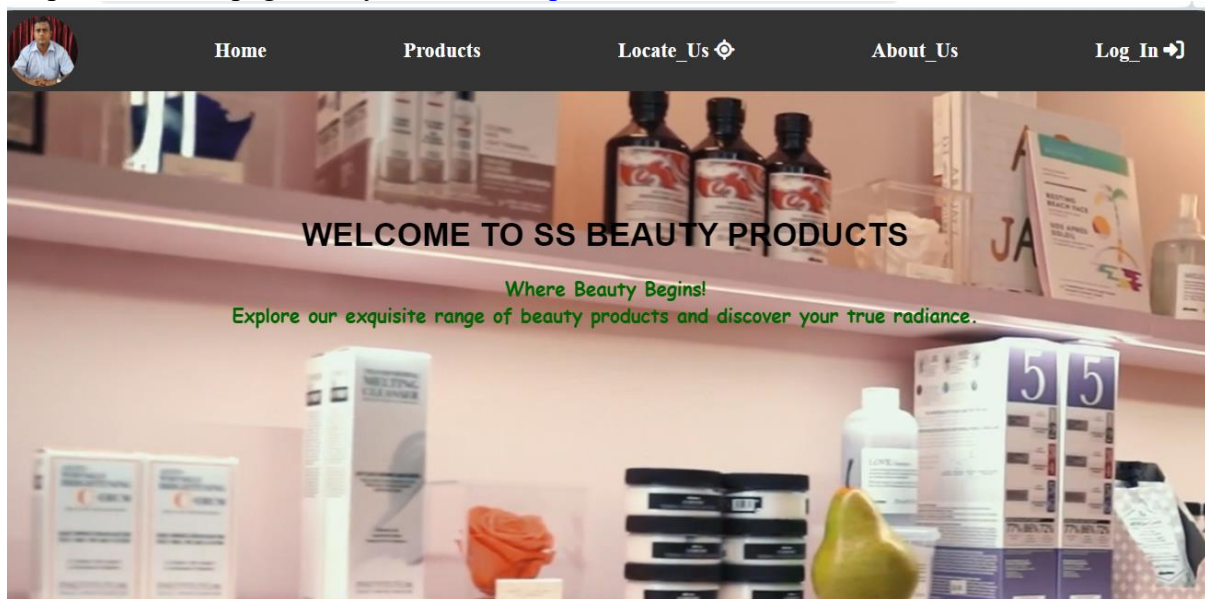
### Step 1:



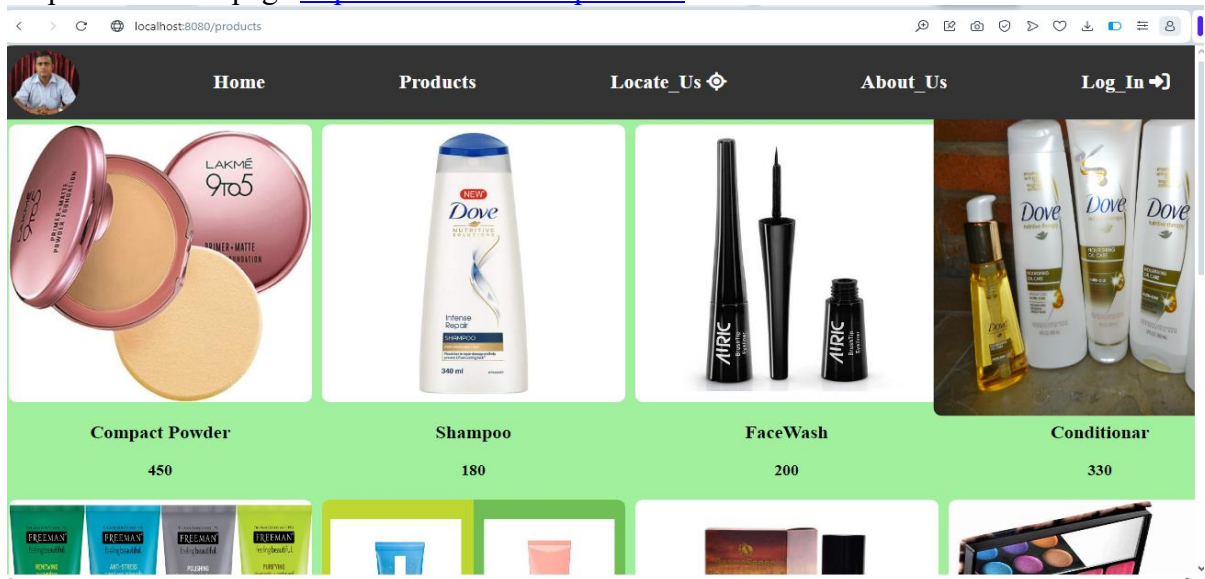
## Step 2: No Error

```
Spring Boot Project-master - SsBeautyProductsApplication [Spring Boot App] C:\Program Files\Java\jdk-21\bin\javaw.exe (05-Feb-2024, 4:07:56 pm) [pid: 11808]
2024-02-05T16:08:03.252+05:30 INFO 11808 --- [restartedMain] c.business.SsBeautyProductsApplication : No active profile set, falling back to 1 default profile: "default"
2024-02-05T16:08:03.446+05:30 INFO 11808 --- [restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : DevTools property defaults active! Set 'spring.devtools.add-prope
2024-02-05T16:08:03.446+05:30 INFO 11808 --- [restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : For additional web related logging consider setting the 'logging.
2024-02-05T16:08:05.270+05:30 INFO 11808 --- [restartedMain] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2024-02-05T16:08:05.415+05:30 INFO 11808 --- [restartedMain] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 125 ms. Found 3 JPA r
2024-02-05T16:08:06.989+05:30 INFO 11808 --- [restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2024-02-05T16:08:06.996+05:30 INFO 11808 --- [restartedMain] o.a.catalina.core.AprLifecycleListener : Loaded Apache Tomcat Native library [2.0.6] using APR version [1.
2024-02-05T16:08:07.062+05:30 INFO 11808 --- [restartedMain] o.a.catalina.core.AprLifecycleListener : OpenSSL successfully initialized [OpenSSL 3.0.11 19 Sep 2023]
2024-02-05T16:08:07.085+05:30 INFO 11808 --- [restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2024-02-05T16:08:07.085+05:30 INFO 11808 --- [restartedMain] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.12]
2024-02-05T16:08:07.269+05:30 INFO 11808 --- [restartedMain] o.s.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2024-02-05T16:08:07.270+05:30 INFO 11808 --- [restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 3822 ms
2024-02-05T16:08:07.672+05:30 INFO 11808 --- [restartedMain] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [name: default]
2024-02-05T16:08:07.846+05:30 INFO 11808 --- [restartedMain] org.hibernate.Version : HHH000412: Hibernate ORM core version 6.2.7.Final
2024-02-05T16:08:07.854+05:30 INFO 11808 --- [restartedMain] org.hibernate.cfg.Environment : HHH000406: Using bytecode reflection optimizer
2024-02-05T16:08:08.265+05:30 INFO 11808 --- [restartedMain] o.h.b.i.BytecodeProviderInitiator : HHH000021: Bytecode provider name : bytebuddy
2024-02-05T16:08:08.625+05:30 INFO 11808 --- [restartedMain] o.s.o.j.p.SpringPersistenceUnitInfo : No LoadTimeWeaver setup: ignoring JPA class transformer
2024-02-05T16:08:08.674+05:30 INFO 11808 --- [restartedMain] com.zaxxer.hikari.HikariDataSource : beautyproject - Starting...
2024-02-05T16:08:10.446+05:30 INFO 11808 --- [restartedMain] com.zaxxer.hikari.pool.HikariPool : beautyproject - Added connection com.mysql.cj.jdbc.ConnectionImpl
2024-02-05T16:08:10.456+05:30 INFO 11808 --- [restartedMain] com.zaxxer.hikari.HikariDataSource : beautyproject - Start completed.
2024-02-05T16:08:11.406+05:30 INFO 11808 --- [restartedMain] o.h.b.i.BytecodeProviderInitiator : HHH000021: Bytecode provider name : bytebuddy
2024-02-05T16:08:13.171+05:30 INFO 11808 --- [restartedMain] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000490: Using JtaPlatform implementation: [org.hibernate.engin
2024-02-05T16:08:13.603+05:30 INFO 11808 --- [restartedMain] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2024-02-05T16:08:15.643+05:30 INFO 11808 --- [restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 35729
2024-02-05T16:08:15.762+05:30 INFO 11808 --- [restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2024-02-05T16:08:15.787+05:30 INFO 11808 --- [restartedMain] c.business.SsBeautyProductsApplication : Started SsBeautyProductsApplication in 13.658 seconds (process ru
```

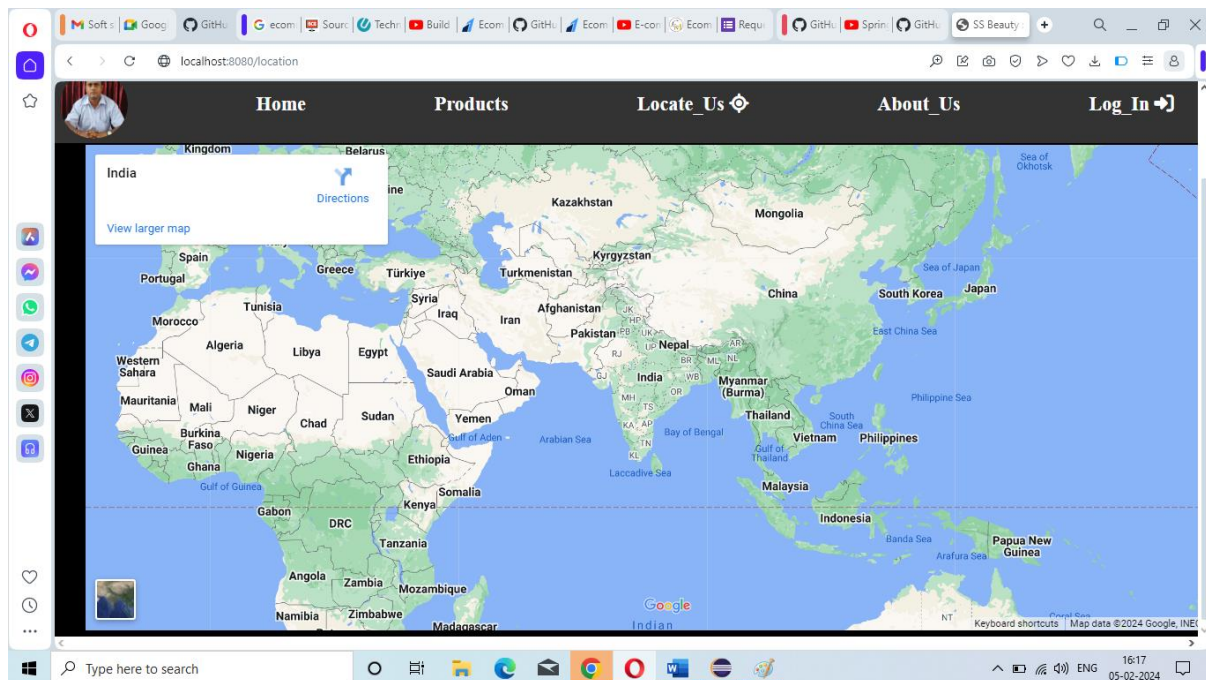
## Step 3: Go to the page in any browser : <http://localhost:8080/home>



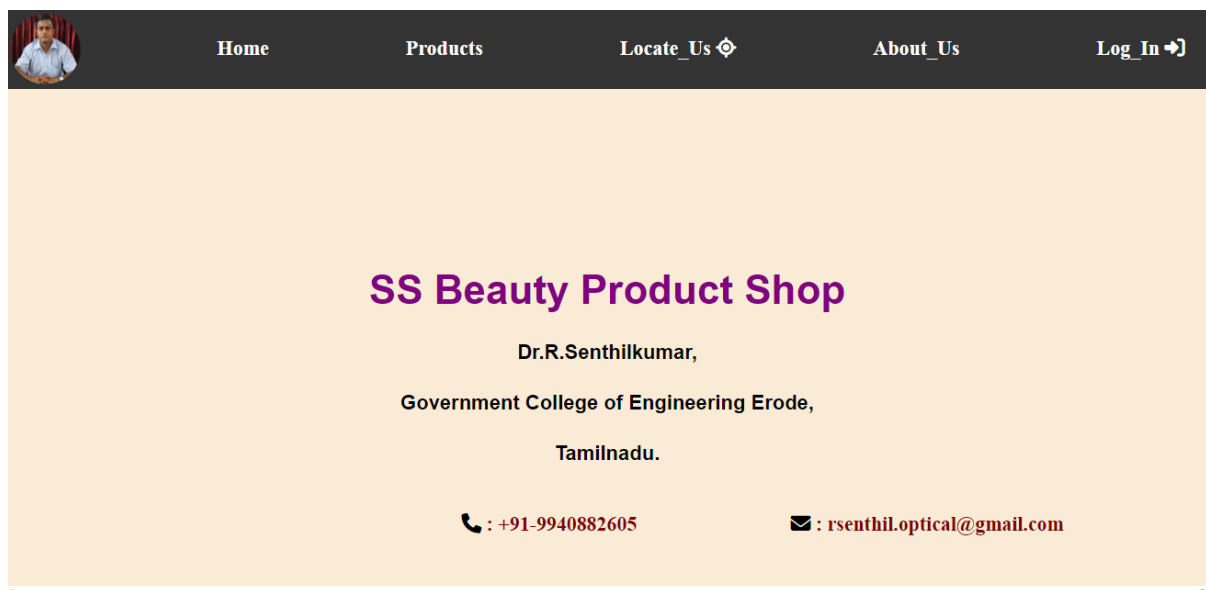
## Step 4: Go to the page <http://localhost:8080/products> or Products



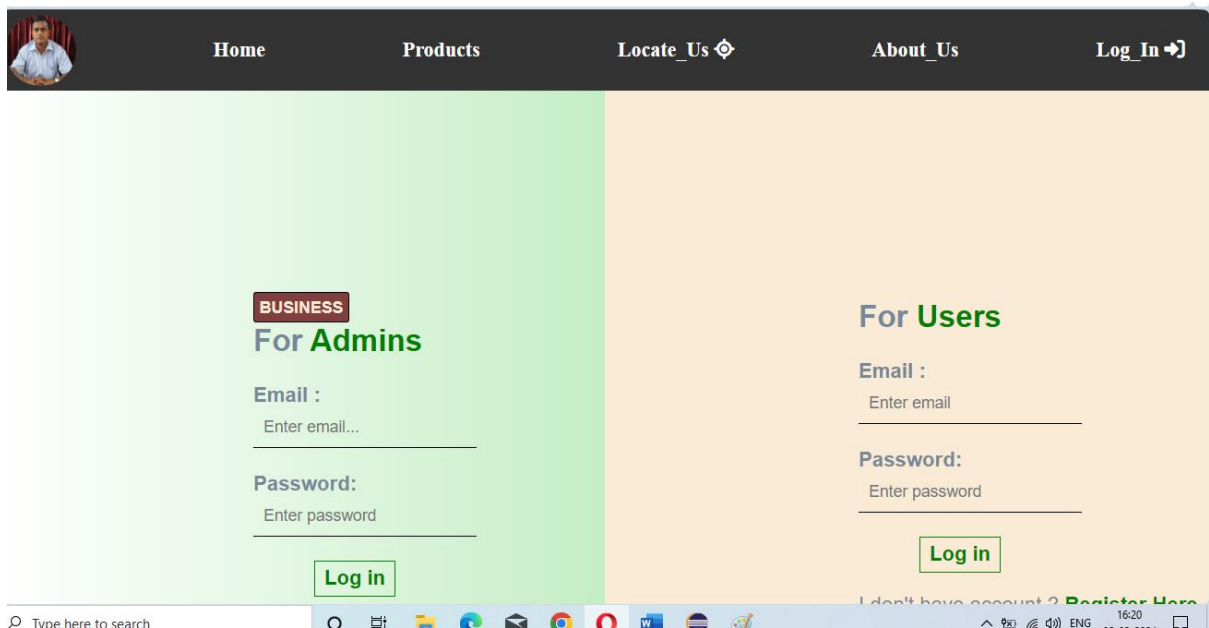
Step 5: Locate us: <http://localhost:8080/location>



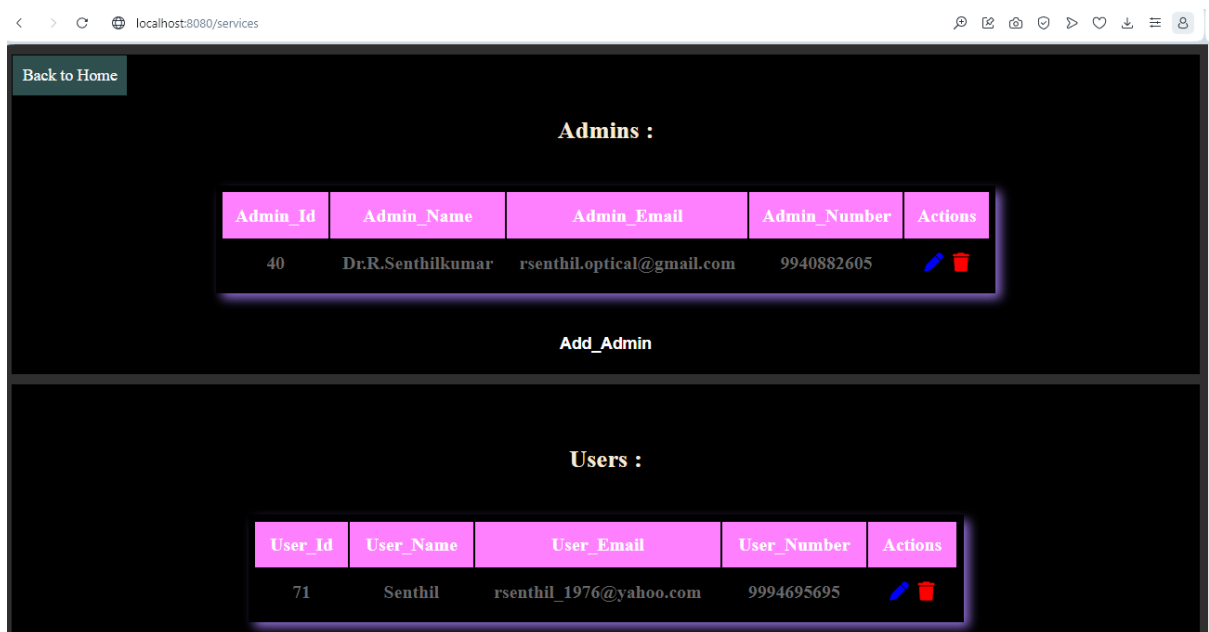
Step 6: <http://localhost:8080/about>



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



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





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

The screenshot shows the "addAdmin" page. It features a "Back to Services" button on the left. On the right, the title "ADD NEW ADMIN/PARTNER" is displayed in red. Below the title, there are three input fields for "Admin\_Name", "Admin\_Email", and "Admin\_Number", each with a placeholder text "Enter name...", "Enter email...", and "Enter number..." respectively. At the bottom right, there is a green "Add\_Admin" button.

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

The screenshot shows the "addUser" page. It features a "Back to Services" button on the left. On the right, the title "ADD NEW USER/CUSTOMER" is displayed in red. Below the title, there are three input fields for "User\_Name", "User\_Email", and "User\_Number", each with a placeholder text "Enter name...", "Enter email...", and "Enter number..." respectively. At the bottom right, there is a green "Add\_User" button.



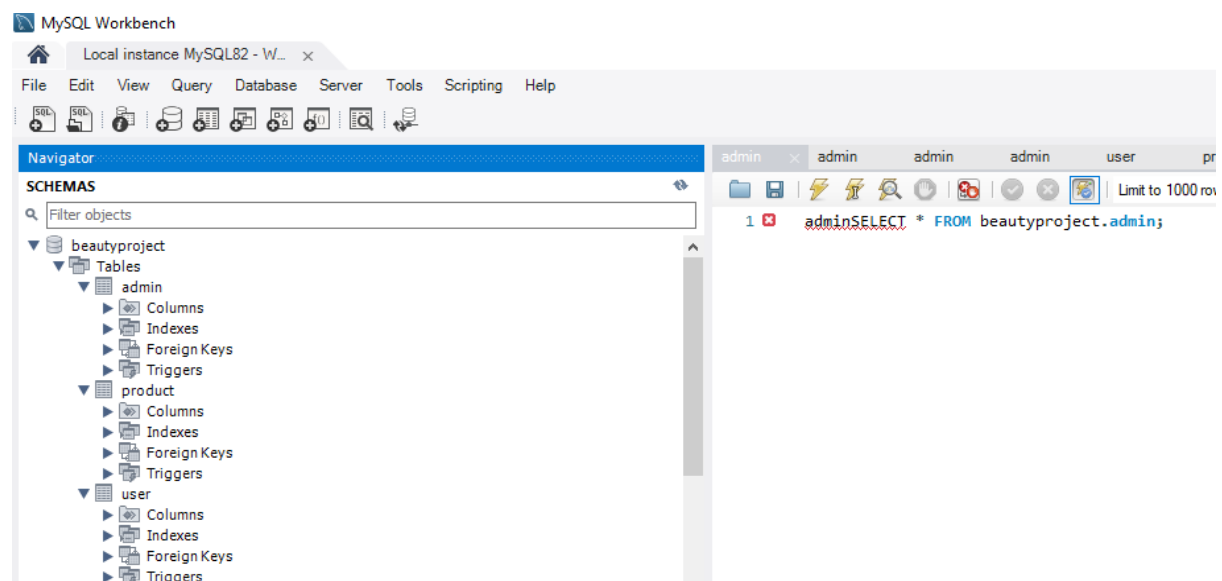
Step 11: <http://localhost:8080/addProduct>

[Back to Services](#)

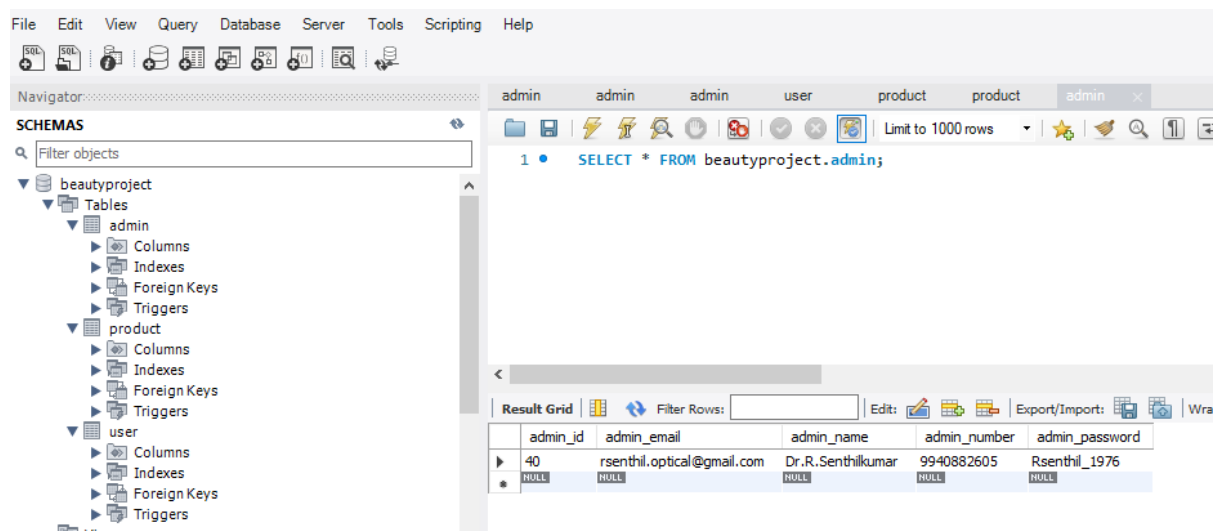
**ADD NEW PRODUCT :**  
**Product\_Name :**  
  
**Product\_Price :**  
  
**Product\_Desc :**

## MySQL Workbench

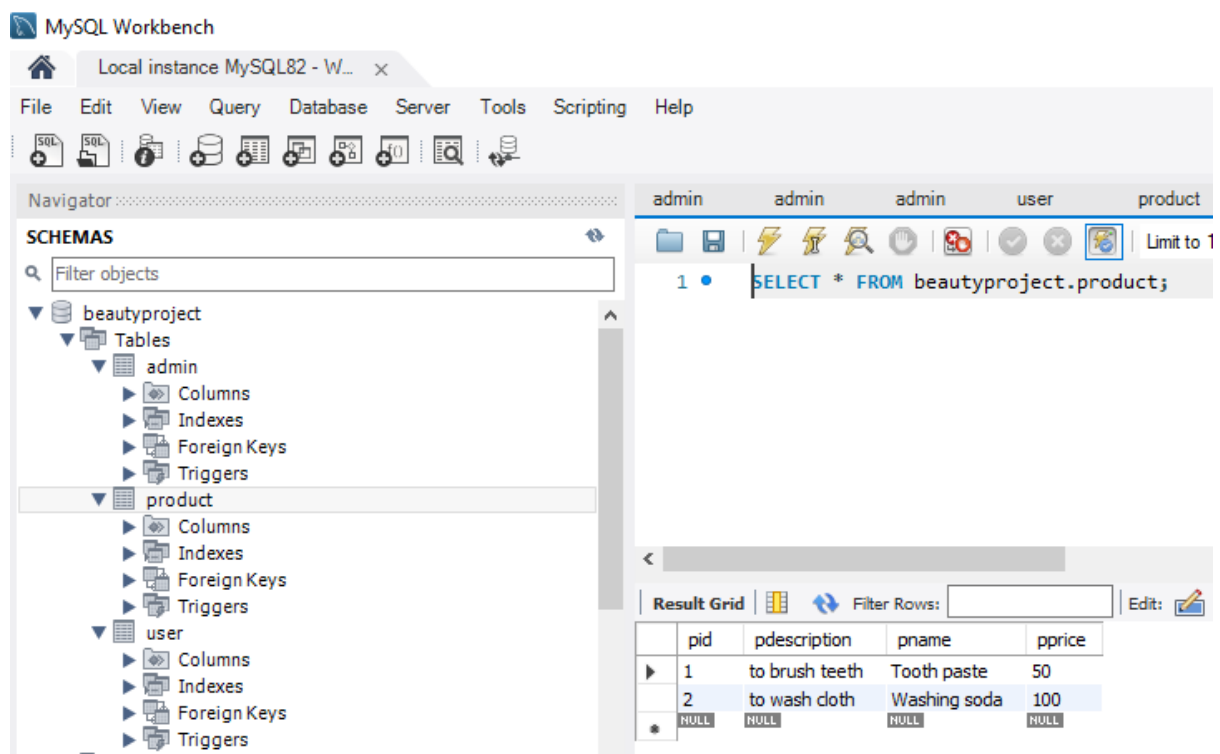
### Beautypoint schema



## Admin table

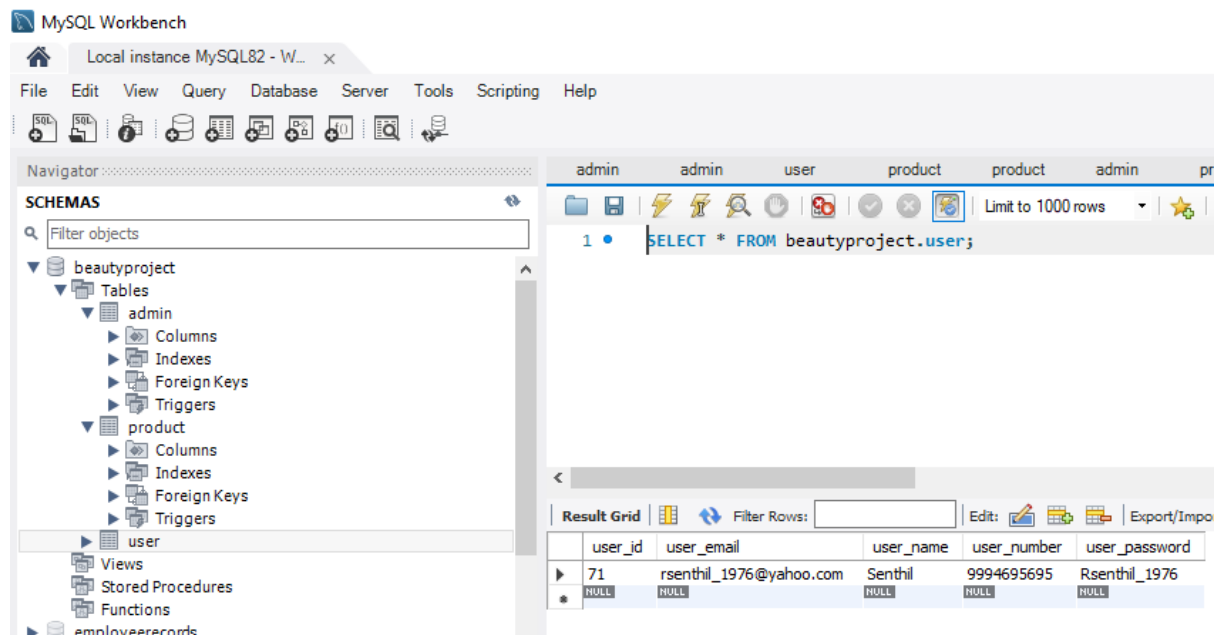


## Product table



## User table





Report and project files are available in github:

<https://github.com/senthilkumarIRTT/JFSD-Mini-Project/tree/main>