**Sporty Shoes (**e-commerce portal**)**

**Developer: Jyothi Kothapally**

Version History

|  |  |
| --- | --- |
| Author | Jyothi kothapally |
| Date | 12-nov-2021 |
| Version | 1.0 |
| Purpose | Specific documentation |

**Contents**

[1.GitHub Link 1](#_Toc87571610)

[2.Modules of the project 1](#_Toc87571611)

[3.Sprint Planning and Task Completion 1](#_Toc87571612)

[4.Pushing the code to GitHub repository 1](#_Toc87571613)

[5.technologies used 1](#_Toc87571614)

Sportyshoes

Sportyshoes.com is an e-commerce portal that lets people purchase shoes on their website.

# **GitHub Link**

This code for this project is hosted at [kothapallyjyothi/FlyAway (github.com)](https://github.com/kothapallyjyothi/FlyAway).

# **Modules of the project**

    1.      MVC model.

2.      Eclipse.

3.      Apache tomcat 9.0

4.      Hibernate.

5.      MySql-Database.

6.      HTML

7.      Spring boot.

8.      Java configuration model.

9.      Using spring framework.

10.   Thymeleaf.

# **3.Sprint Planning and Task Completion**

|  |  |
| --- | --- |
| 1 | Controller Files: AdminHome.java, AdminLogin.java, AdminProduct.java, UserHome.java, UserLogin.java, UserSignup.java |
| 2 | Entity Files: Admin.java, Product.java, PurchaseHistory.java, User.java, UserOrder.java. |
| 3 | Repository Files: AdminRepo.java, ProducRepo.java, PurchaseHistoryRepo.java, UserOrderRepo.java, UserRepo.java. |
| 4 | Configuration Files: MvcConfiguration.java, ViewConfiguration.java and DataModelApplication.java. |
| 5 | css Files: Home.css, Login.css, Signup.css.  HTML Files: adminHome.html, adminLogin.html, adminNotFound.html, ChangeAdminPassword.html, userHome.html, userLogin.html, userNotFound.html, userOrder.html, userSignup.html. |
| 6 | Application.properties, data.sql, AdminepoTest.java, DataModelApplicationTests.java, PurchaseReports.java, StudentRepoTest.java, UserOrderRepoTest.java, UserRepoTest.java. |

Sprint:1:

* Created pom.xml.
* Created repository.
* Created Entity.

Sprint:2:

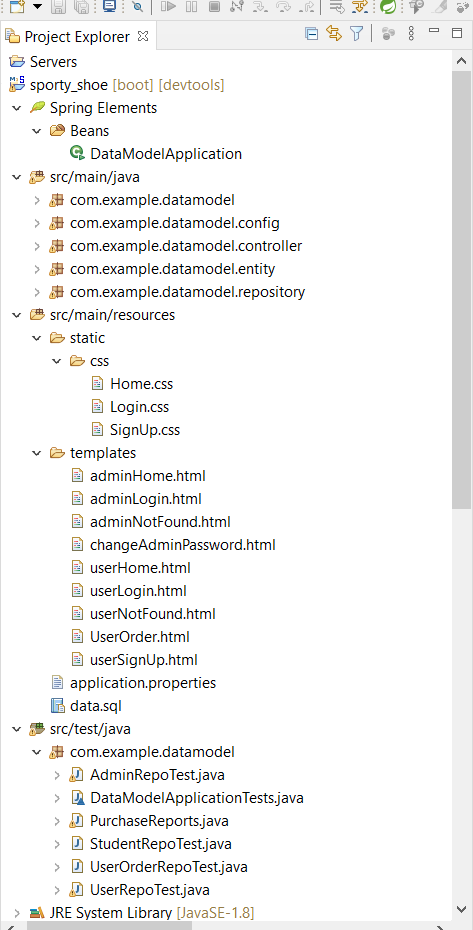
* Created Controller.
* Created main file.
* Created HTML files.

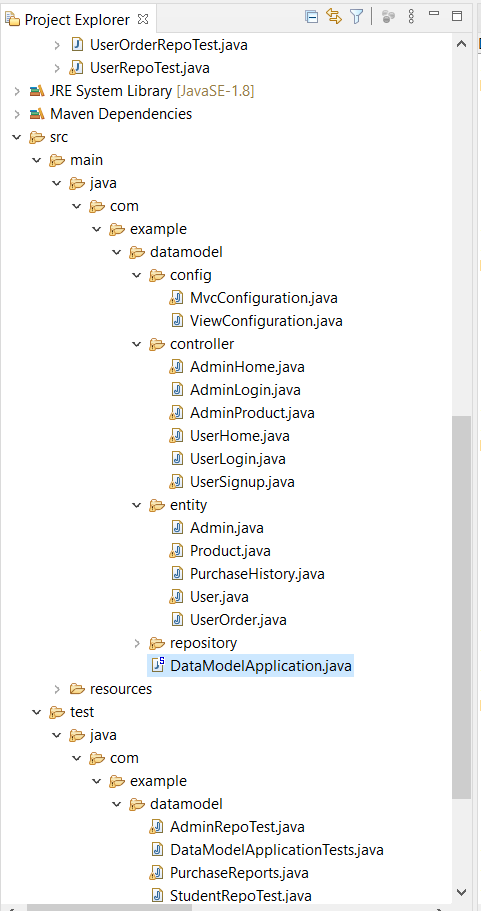
Sprint:3:

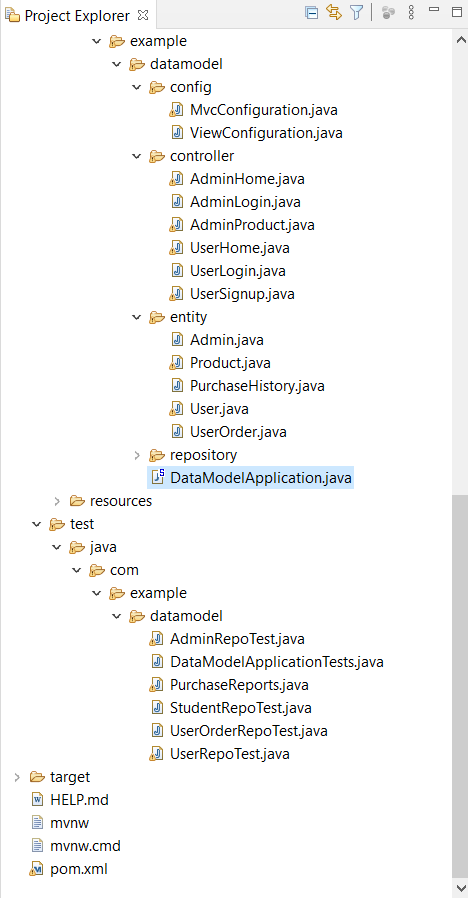
* Created property files.
* Created sportyshoesdbs database.
* Pushed the code to the repository.

Project code:

Folder Structure:







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>2.5.6</version>

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

</parent>

<groupId>com.sportsshoes</groupId>

<artifactId>sporty\_shoe</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>sporty\_shoe</name>

<description>Demo project for Spring Boot</description>

<properties>

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

</properties>

<dependencies>

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

<dependency>

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

<artifactId>spring-boot-starter-jdbc</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>mysql</groupId>

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

<scope>runtime</scope>

</dependency>

<dependency>

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

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

<scope>test</scope>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-thymeleaf -->

<dependency>

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

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

<version>2.5.6</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

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

</plugin>

</plugins>

</build>

</project>

DataModelApplication.java

package com.example.datamodel;

import com.example.datamodel.repository.\*;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

import org.springframework.boot.CommandLineRunner;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import java.util.Date;

@SpringBootApplication

public class DataModelApplication implements CommandLineRunner {

@Autowired

private UserOrderRepo userOrderRepo;

@Autowired

private ProductRepo productRepo;

@Autowired

private AdminRepo adminRepo;

@Autowired

private UserRepo userRepo;

@Autowired

private PurchaseHistoryRepo purchaseHistoryRepo;

private Logger logger = LoggerFactory.getLogger(this.getClass());

public static void main(String[] args) {

SpringApplication.run(DataModelApplication.class, args);

}

@Override

public void run(String... args) throws Exception {

//purchaseHistoryRepo.

// logger.info("list result -> {}",adminRepo.filterByDateAndCategory());

// logger.info("gotcha -> {}", userOrderRepo.insert());

}

}

**MvcConfiguration.java**.

package com.example.datamodel.config;

import org.springframework.context.annotation.Configuration;

import org.springframework.web.servlet.config.annotation.EnableWebMvc;

@Configuration

public class MvcConfiguration {

}

**ViewConfiguration.java**

package com.example.datamodel.config;

import org.springframework.context.ApplicationContext;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.web.servlet.ViewResolver;

import org.thymeleaf.spring5.SpringTemplateEngine;

import org.thymeleaf.spring5.templateresolver.SpringResourceTemplateResolver;

import org.thymeleaf.spring5.view.ThymeleafViewResolver;

import org.thymeleaf.templateresolver.ITemplateResolver;

@Configuration

public class ViewConfiguration {

private final ApplicationContext applicationContext;

public ViewConfiguration(ApplicationContext applicationContext) {

this.applicationContext = applicationContext;

}

@Bean

public ITemplateResolver templateResolver() {

SpringResourceTemplateResolver templateResolver = new SpringResourceTemplateResolver();

templateResolver.setApplicationContext(applicationContext);

templateResolver.setPrefix("classpath:/templates/");

templateResolver.setSuffix(".html");

templateResolver.setTemplateMode("HTML");

templateResolver.setCharacterEncoding("UTF-8");

return templateResolver;

}

@Bean

public SpringTemplateEngine templateEngine() {

SpringTemplateEngine templateEngine = new SpringTemplateEngine();

templateEngine.setEnableSpringELCompiler(true);

templateEngine.setTemplateResolver(templateResolver());

return templateEngine;

}

@Bean

public ViewResolver viewResolver() {

ThymeleafViewResolver viewResolver = new ThymeleafViewResolver();

viewResolver.setTemplateEngine(templateEngine());

return viewResolver;

}

}

**AdminHome.java**

package com.example.datamodel.controller;

import com.example.datamodel.entity.\*;

import com.example.datamodel.repository.AdminRepo;

import com.example.datamodel.repository.ProductRepo;

import com.example.datamodel.repository.PurchaseHistoryRepo;

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

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

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

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

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

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

@Controller

public class AdminHome {

@Autowired

private AdminRepo adminRepo;

@Autowired

private ProductRepo productRepo;

@Autowired

private PurchaseHistoryRepo purchaseHistoryRepo;

@RequestMapping("/change")

public String change() {

return "changeAdminPassword";

}

@PostMapping("/changePassword")

public String changePassword(@RequestParam Map<String,String>maps) {

String email = maps.get("email");

String oldPassword = maps.get("oldPassword");

String newPassword = maps.get("newPassword");

Admin admin = adminRepo.changePasswordByEmail(email,oldPassword,newPassword);

if(admin == null) {

return "adminNotFound";

} else {

return "adminLogin";

}

}

@RequestMapping("/getUsers")

public String getUsers(Model model) {

List<User> users = adminRepo.getAllUsersForAdmin();

model.addAttribute("users",users);

return "adminHome";

}

@RequestMapping("/searchUser")

public String searchUserByName(Model model, @RequestParam Map<String, String> maps) {

String name = maps.get("name");

List<User> users = adminRepo.searchUser(name);

model.addAttribute("users",users);

return "adminHome";

}

@RequestMapping("/purchaseHistory")

public String purchaseReports(Model model) {

List<PurchaseHistory> purchaseHistories = purchaseHistoryRepo.getPurchaseHistory();

model.addAttribute("purchaseHistory",purchaseHistories);

return "adminHome";

}

}

**AdminLogin.java**

package com.example.datamodel.controller;

import com.example.datamodel.repository.AdminRepo;

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

import org.springframework.stereotype.Controller;

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

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

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

import java.util.Map;

@Controller

public class AdminLogin {

@Autowired

private AdminRepo adminRepo;

@RequestMapping("/admin")

public String admin() {

return "adminLogin";

}

@PostMapping("/adminLogin")

public String adminLogin(@RequestParam Map<String, String> maps) {

String email = maps.get("email");

String password = maps.get("password");

if(adminRepo.verifyAdmin(email,password) ==null) {

return "adminNotFound";

}

else {

return "adminHome";

}

}

}

**AdminProduct.java**

package com.example.datamodel.controller;

import com.example.datamodel.entity.Product;

import com.example.datamodel.repository.ProductRepo;

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

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

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

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

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

import java.util.List;

@Controller

public class AdminProduct {

@Autowired

private ProductRepo productRepo;

@RequestMapping("/getProducts")

public String getProducts(Model model) {

List<Product> products = productRepo.getAllProducts();

model.addAttribute("products", products);

return "adminHome";

}

@RequestMapping("/addProduct")

public String addProduct(@ModelAttribute Product product) {

Product product1 = productRepo.addProduct(product);

return "adminHome";

}

@RequestMapping("/delProduct")

public String delProduct(@RequestParam("id") Long id) {

productRepo.delProduct(id);

return "adminHome";

}

@RequestMapping("/updateProduct")

public String updateProduct(@RequestParam("id") Long id, @ModelAttribute Product product) {

productRepo.updateProduct(id,product);

return "adminHome";

}

}

**UserHome.java**

package com.example.datamodel.controller;

import com.example.datamodel.entity.Product;

import com.example.datamodel.entity.User;

import com.example.datamodel.entity.UserOrder;

import com.example.datamodel.repository.ProductRepo;

import com.example.datamodel.repository.UserOrderRepo;

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

import org.springframework.boot.web.servlet.server.Session;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

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

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

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import java.util.List;

import java.util.Map;

@Controller

public class UserHome {

@Autowired

private ProductRepo productRepo;

@Autowired

private UserOrderRepo userOrderRepo;

@RequestMapping("/add")

public String addProduct(Model model, HttpServletRequest request, HttpServletResponse response) {

List<Product> products = productRepo.getAllProducts();

model.addAttribute("products",products);

return "userHome";

}

@RequestMapping("/addProducts")

public String addProductToOrder(Model model, @RequestParam("id") Long id, HttpServletRequest request, HttpServletResponse response) {

User user = (User)request.getSession().getAttribute("user");

Long userId = user.getId();

UserOrder userOrder = productRepo.addProductToUserOrder(id,userId);

Long orderId = userOrder.getId();

Long price = userOrderRepo.addTotalCostToOrder(orderId);

model.addAttribute("order",userOrder);

model.addAttribute("price",price);

return "UserOrder";

}

}

**UserLogin.java**

package com.example.datamodel.controller;

import com.example.datamodel.entity.User;

import com.example.datamodel.repository.UserRepo;

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

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.web.bind.annotation.\*;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import java.util.Map;

@Controller

@SessionAttributes({"user"})

public class UserLogin {

@Autowired

private UserRepo userRepo;

@Autowired

HttpSession session;

@RequestMapping("/")

public String loginPage() {

return "userLogin";

}

@PostMapping("/userLogin")

public String loginUser(Model model, @RequestParam Map<String, String> maps, HttpServletRequest request, HttpServletResponse response) throws Exception {

String email = maps.get("email");

String password = maps.get("password");

User user = userRepo.verifyUser(email,password);

request.getSession().setAttribute("user",user);

if(userRepo.verifyUser(email,password) ==null) {

return "userNotFound";

}

else {

return "userHome";

}

}

}

**UserSignup.java**

package com.example.datamodel.controller;

import com.example.datamodel.entity.User;

import com.example.datamodel.repository.UserRepo;

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

import org.springframework.stereotype.Controller;

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

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

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

@Controller

public class UserSignup {

@Autowired

private UserRepo userRepo;

@RequestMapping("/signup")

public String signUp() {

return "userSignUp";

}

@RequestMapping("/UserSignUp")

public String createUser(@ModelAttribute User user) {

User u = userRepo.addNewUser(user);

return "userLogin";

}

}

**Admin.java**

**package** com.example.datamodel.entity;

**import** javax.persistence.\*;

/\*

\* STORES ADMIN DETAILS TO DATABASE

\* \*/

@Entity

@NamedQuery(name = "get\_all\_admins", query = "select a from Admin a")

**public** **class** Admin {

@Id

@GeneratedValue(strategy = GenerationType.***IDENTITY***)

@Column(name = "id", updatable = **false**, nullable = **false**)

**private** Long id;

**private** String password;

**private** String adminName;

**private** String email;

**public** Admin(String password, String adminName, String email) {

**this**.password = password;

**this**.adminName = adminName;

**this**.email = email;

}

/\*

\* Empty constructor to make JPA happy

\* so any kind of data can be acceptable\*/

**public** Admin() {

}

/\*

\* getter & setter for fetching and updating the data\*/

**public** Long getId() {

**return** id;

}

**public** **void** setId(Long id) {

**this**.id = id;

}

**public** String getPassword() {

**return** password;

}

**public** **void** setPassword(String password) {

**this**.password = password;

}

**public** String getAdminName() {

**return** adminName;

}

**public** **void** setAdminName(String adminName) {

**this**.adminName = adminName;

}

**public** String getEmail() {

**return** email;

}

**public** **void** setEmail(String email) {

**this**.email = email;

}

@Override

**public** String toString() {

**return** "Admin{" +

"id=" + id +

", password='" + password + '\'' +

", adminName='" + adminName + '\'' +

", email='" + email + '\'' +

'}';

}

}

**Product.java**

package com.example.datamodel.entity;

import javax.persistence.\*;

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

@Entity

@NamedQueries(value = {

@NamedQuery(name = "get\_all\_products", query = "Select p FROM Product p")

})

public class Product {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name = "id", updatable = false, nullable = false)

private Long id;

private String category;

private String productName;

private Long cost;

@ManyToMany(fetch = FetchType.EAGER)

@JoinTable(name="order\_product",

joinColumns = @JoinColumn(name="product\_id"),

inverseJoinColumns = @JoinColumn(name = "order\_id")

)

private List<UserOrder> userOrders = new ArrayList<>();

public Product() {

}

public Product(String productName,String category, Long cost, List<UserOrder> userOrders) {

this.productName = productName;

this.category = category;

this.cost = cost;

this.userOrders = userOrders;

}

public Product(String productName,String category, Long cost) {

this.productName = productName;

this.category = category;

this.cost = cost;

}

public String getCategory() {

return category;

}

public void setCategory(String category) {

this.category = category;

}

public Long getCost() {

return cost;

}

public void setCost(Long cost) {

this.cost = cost;

}

public Product(String productName) {

this.productName = productName;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getProductName() {

return productName;

}

public void setProductName(String productName) {

this.productName = productName;

}

public List<UserOrder> getUserOrders() {

return userOrders;

}

public void addUserOrders(UserOrder userOrder) {

this.userOrders.add(userOrder);

}

public void delUserOrders(UserOrder userOrder) {

this.userOrders.remove(userOrder);

}

@Override

public String toString() {

return "Product{" +

"id=" + id +

", productName='" + productName + '\'' +

'}';

}

}

PurchaseHistory.java

package com.example.datamodel.entity;

import javax.persistence.\*;

import java.util.Date;

@Entity

public class PurchaseHistory {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name = "id", updatable = false, nullable = false)

private Long id;

private Long productId;

private Long orderId;

private Long cost;

private Long total;

private Long UserId;

private String category;

private String productName;

private Date date;

public PurchaseHistory() {

}

public PurchaseHistory(Long productId, Long orderId, Long cost, Long total, Long userId, String category, String productName, Date date) {

this.productId = productId;

this.orderId = orderId;

this.cost = cost;

this.total = total;

UserId = userId;

this.category = category;

this.productName = productName;

this.date = date;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public Long getProductId() {

return productId;

}

public void setProductId(Long productId) {

this.productId = productId;

}

public Long getOrderId() {

return orderId;

}

public void setOrderId(Long orderId) {

this.orderId = orderId;

}

public Long getCost() {

return cost;

}

public void setCost(Long cost) {

this.cost = cost;

}

public Long getTotal() {

return total;

}

public void setTotal(Long total) {

this.total = total;

}

public Long getUserId() {

return UserId;

}

public void setUserId(Long userId) {

UserId = userId;

}

public String getCategory() {

return category;

}

public void setCategory(String category) {

this.category = category;

}

public String getProductName() {

return productName;

}

public void setProductName(String productName) {

this.productName = productName;

}

public Date getDate() {

return date;

}

public void setDate(Date date) {

this.date = date;

}

@Override

public String toString() {

return "PurchaseHistory{" +

"id=" + id +

", productId=" + productId +

", orderId=" + orderId +

", cost=" + cost +

", total=" + total +

", UserId=" + UserId +

", category='" + category + '\'' +

", productName='" + productName + '\'' +

", date=" + date +

'}';

}

}

**User.java**

package com.example.datamodel.entity;

import org.springframework.web.context.annotation.SessionScope;

import javax.persistence.\*;

@Entity

@NamedQueries(

value = {

@NamedQuery(name="get\_all\_users", query = "select u from User u")

}

)

public class User {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name = "id", updatable = false, nullable = false)

private Long id;

private String userName;

private String email;

private String name;

private String password;

@OneToOne(cascade = {CascadeType.ALL})

private UserOrder userOrder;

public UserOrder getUserOrder() {

return userOrder;

}

public void setUserOrder(UserOrder userOrder) {

this.userOrder = userOrder;

}

public User() {

}

public User(String userName, String email, String name, String password) {

this.userName = userName;

this.email = email;

this.name = name;

this.password = password;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getUserName() {

return userName;

}

public void setUserName(String userName) {

this.userName = userName;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

@Override

public String toString() {

return "User{" +

"id=" + id +

", userName='" + userName + '\'' +

", email='" + email + '\'' +

", name='" + name + '\'' +

", password='" + password + '\'' +

'}';

}

}

**UserOrder.java**

package com.example.datamodel.entity;

import javax.persistence.\*;

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

@Entity

@NamedQueries(value = {

@NamedQuery(name = "get\_all\_orders", query = "Select u FROM UserOrder u")

})

public class UserOrder {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name = "id", updatable = false, nullable = false)

private Long id;

private Long total;

public User getUser() {

return user;

}

public void setUser(User user) {

this.user = user;

}

@OneToOne(cascade = {CascadeType.ALL})

private User user;

private Date date;

@ManyToMany(mappedBy = "userOrders",fetch = FetchType.EAGER)

private List<Product> products = new ArrayList<>();

public UserOrder() {

}

public UserOrder(Long total, Date date) {

this.total = total;

this.date = date;

}

public UserOrder(Date date) {

this.date = date;

this.id= user.getId();

}

public UserOrder(Date date,Long id) {

this.date = date;

}

public UserOrder(Long total, Date date, List<Product> products) {

this.total = total;

this.date = date;

this.products = products;

}

public UserOrder(Long total) {

this.total = total;

}

public Date getDate() {

return date;

}

public void setDate(Date date) {

this.date = date;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public Long getTotal() {

return total;

}

public void setTotal(Long total) {

this.total = total;

}

public List<Product> getProducts() {

return products;

}

public void addProducts(Product product) {

this.products.add(product);

}

public void delProducts(Product product) {

this.products.remove(product);

}

@Override

public String toString() {

return "UserOrder{" +

"id=" + id +

", total=" + total +

", date=" + date +

", products=" + products +

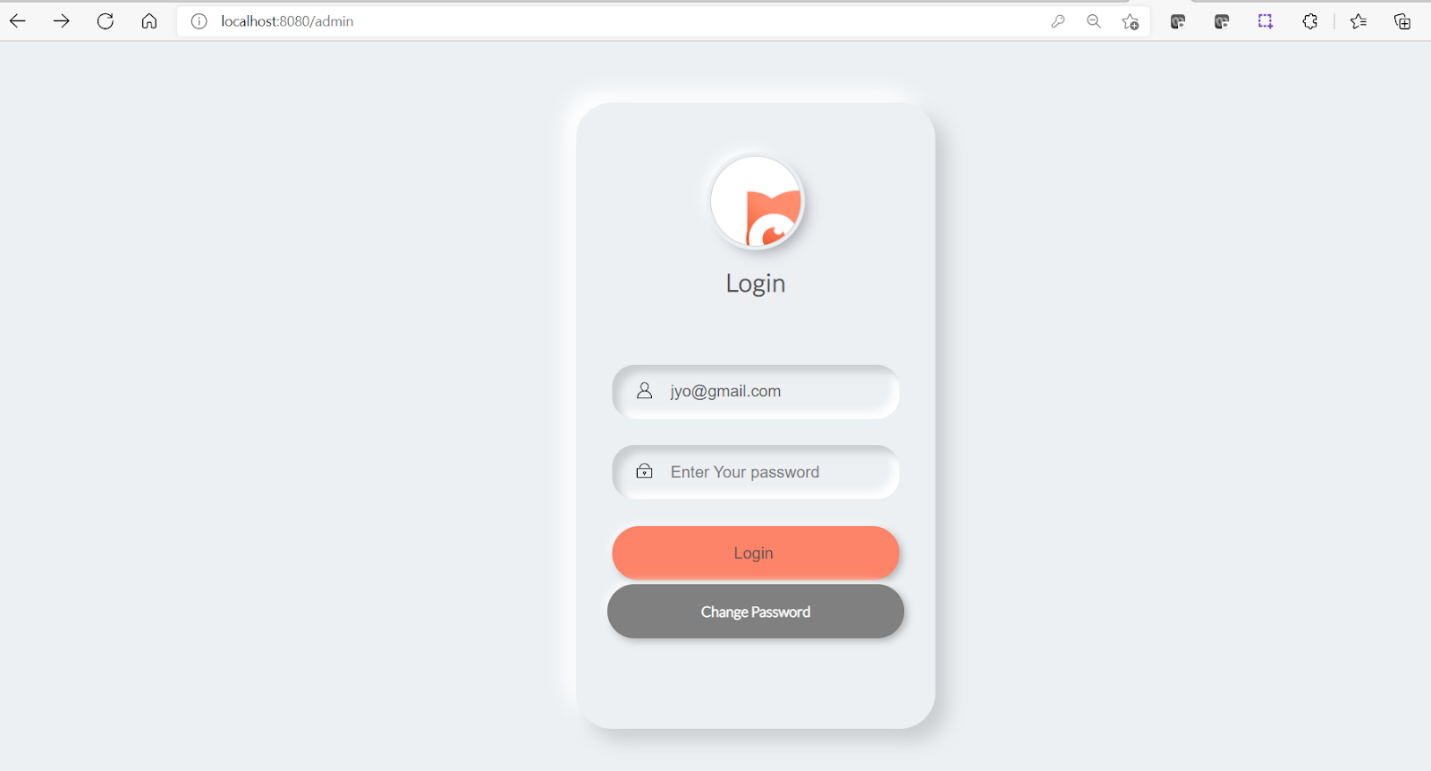
'}';

}

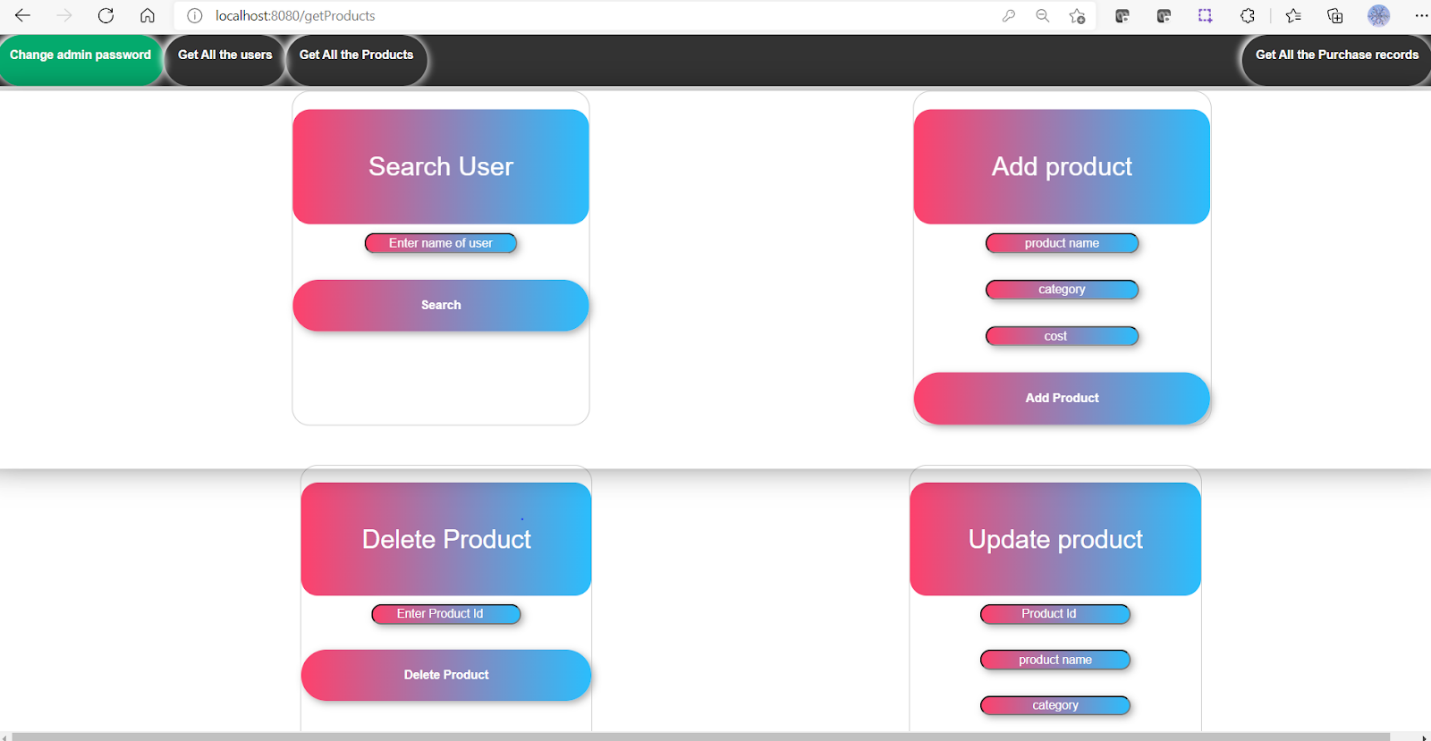
}

Screenshots:

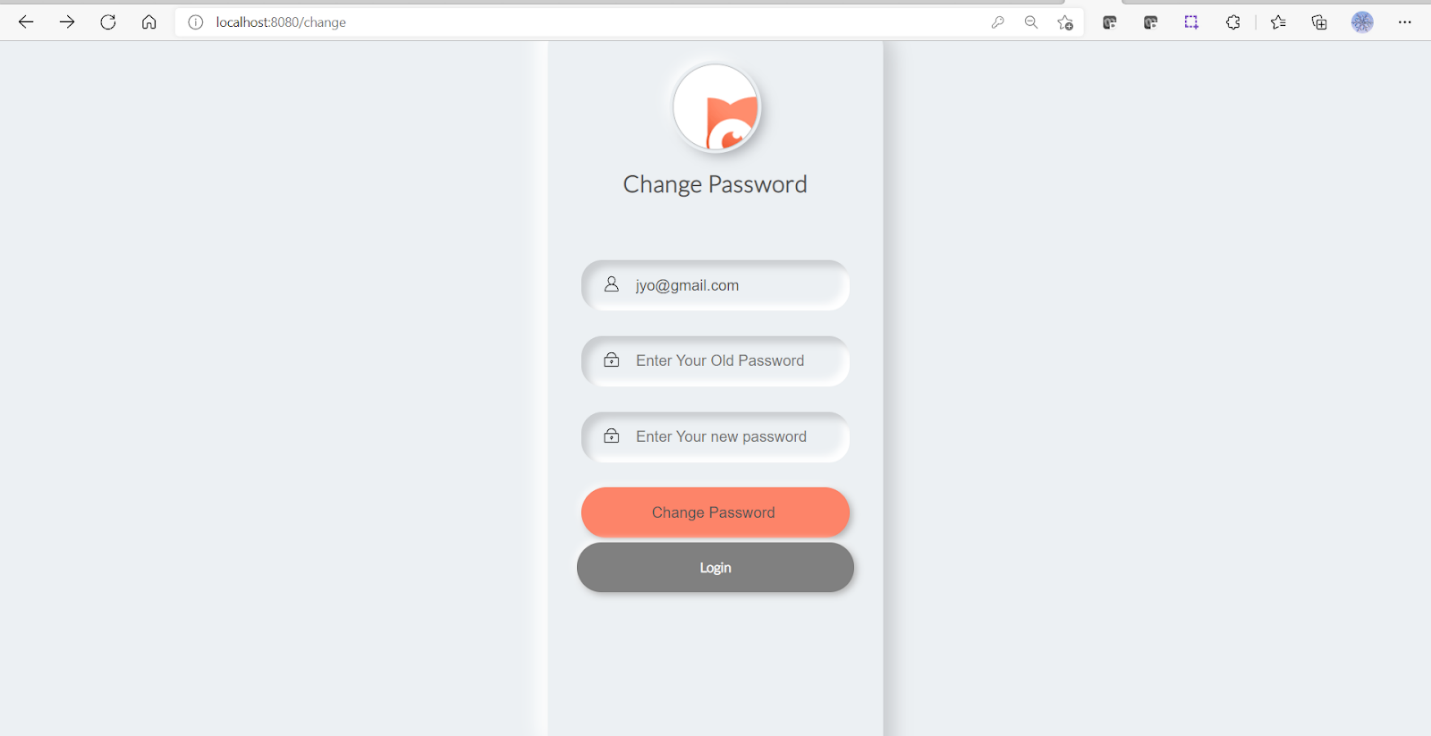
Admin Login Page:



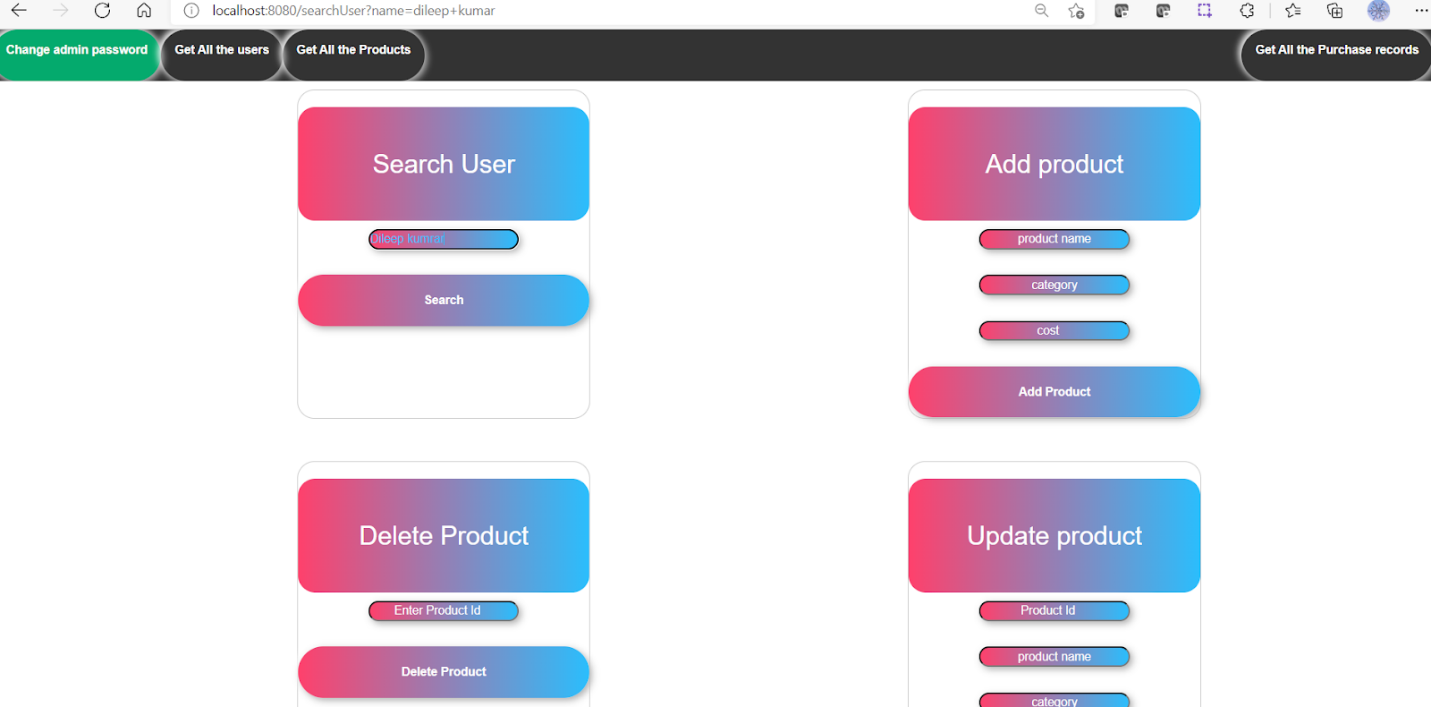
Admin Home page:

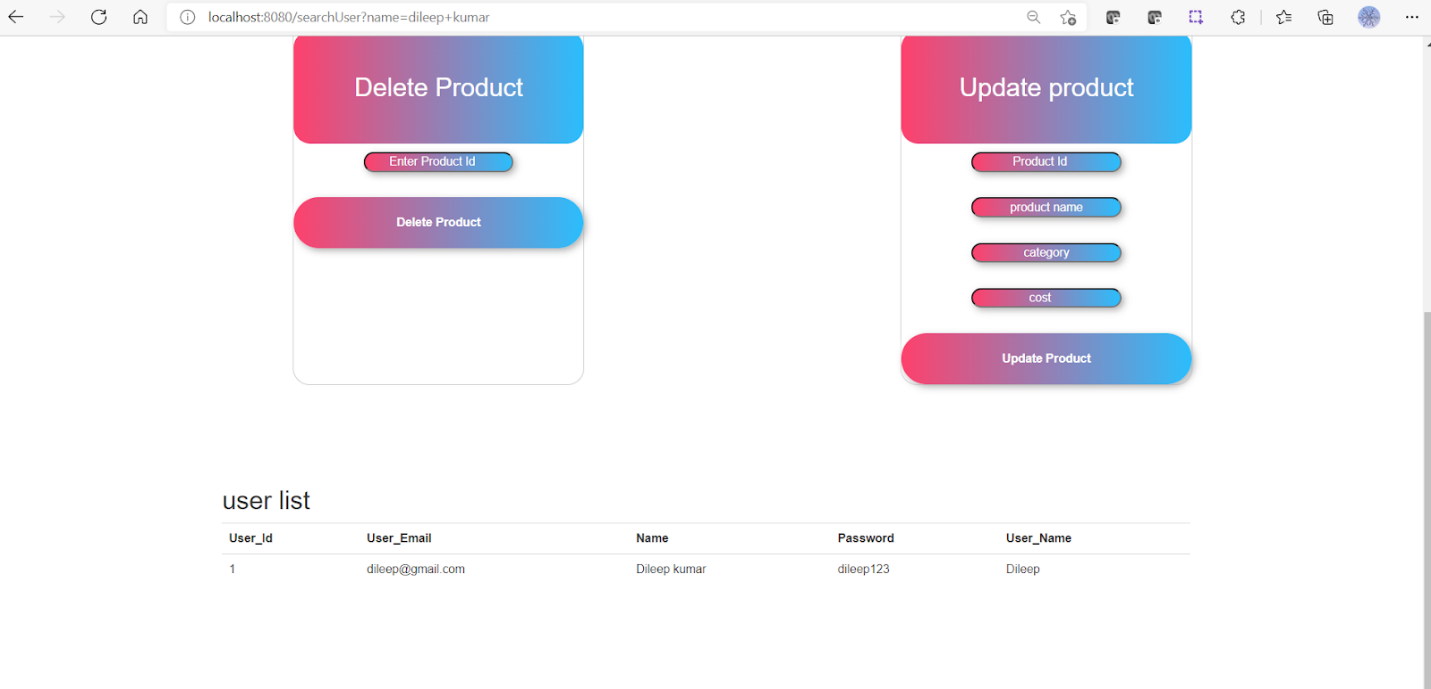


Admin Change Password:

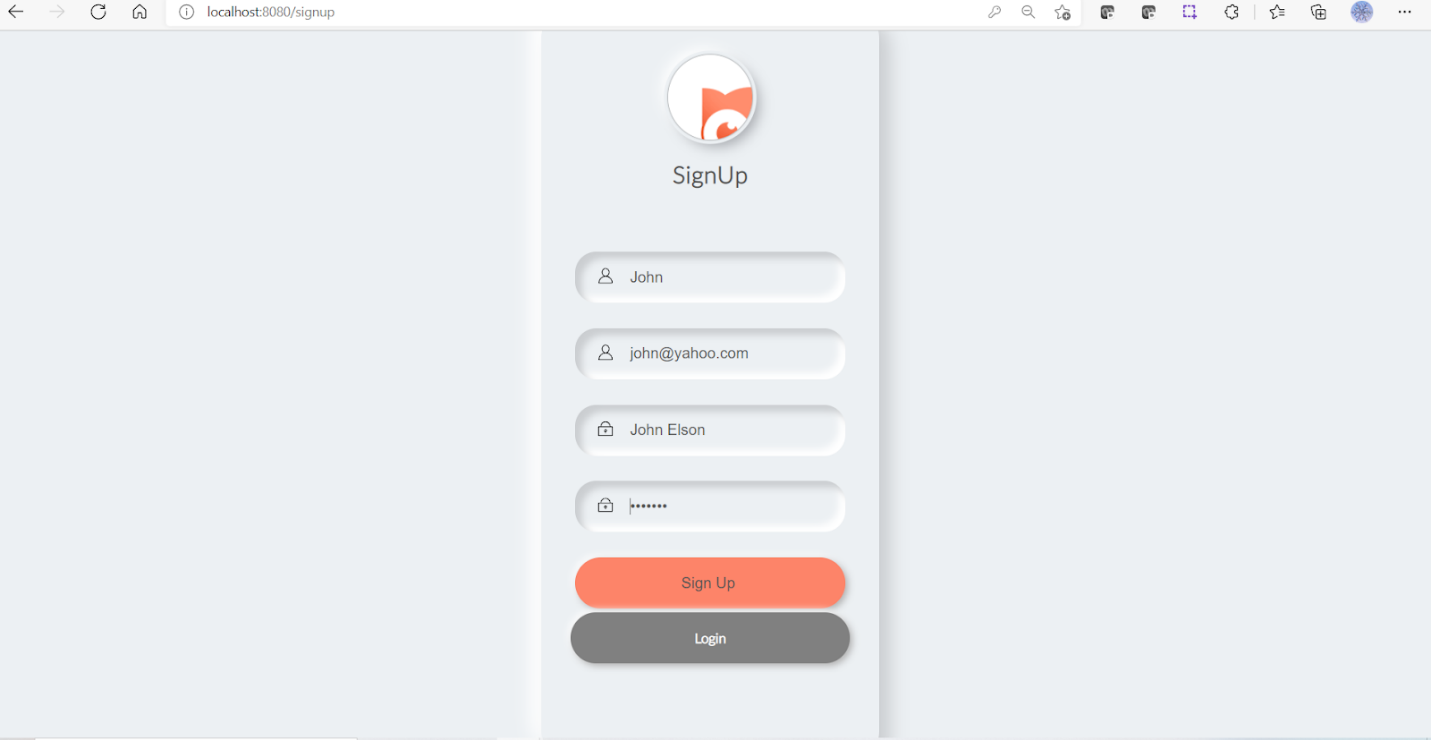


Search User:

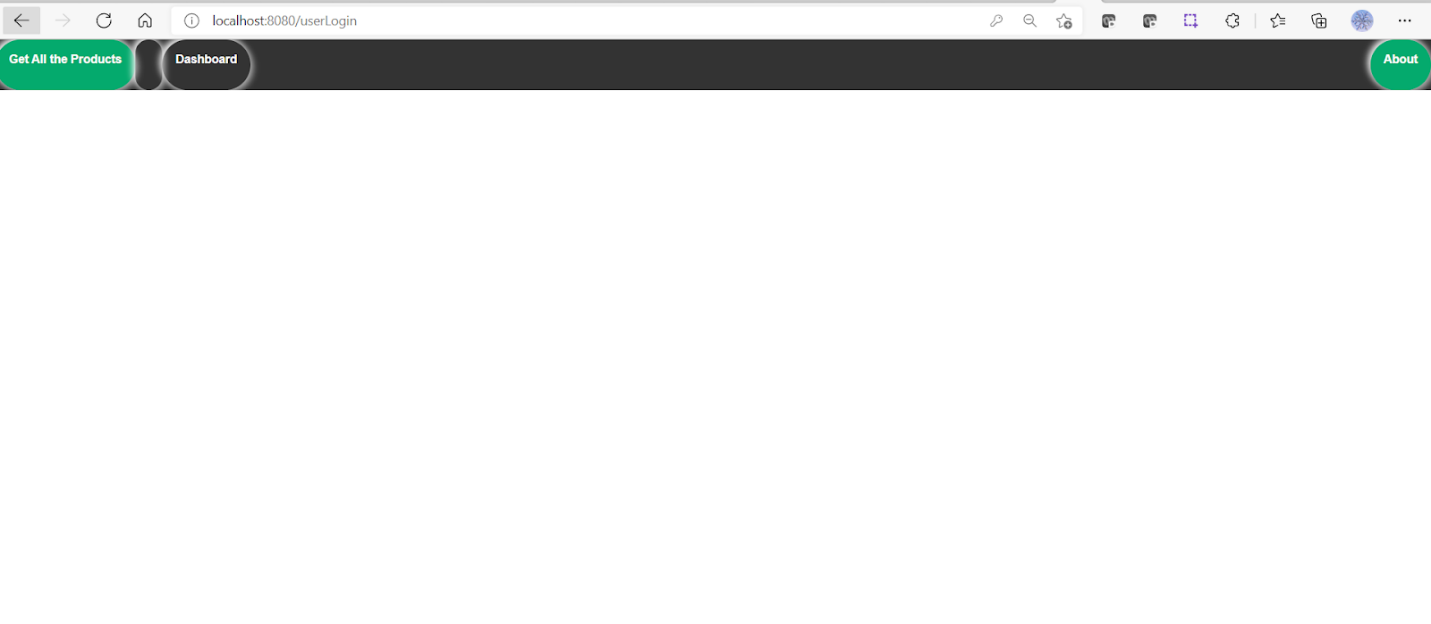
``



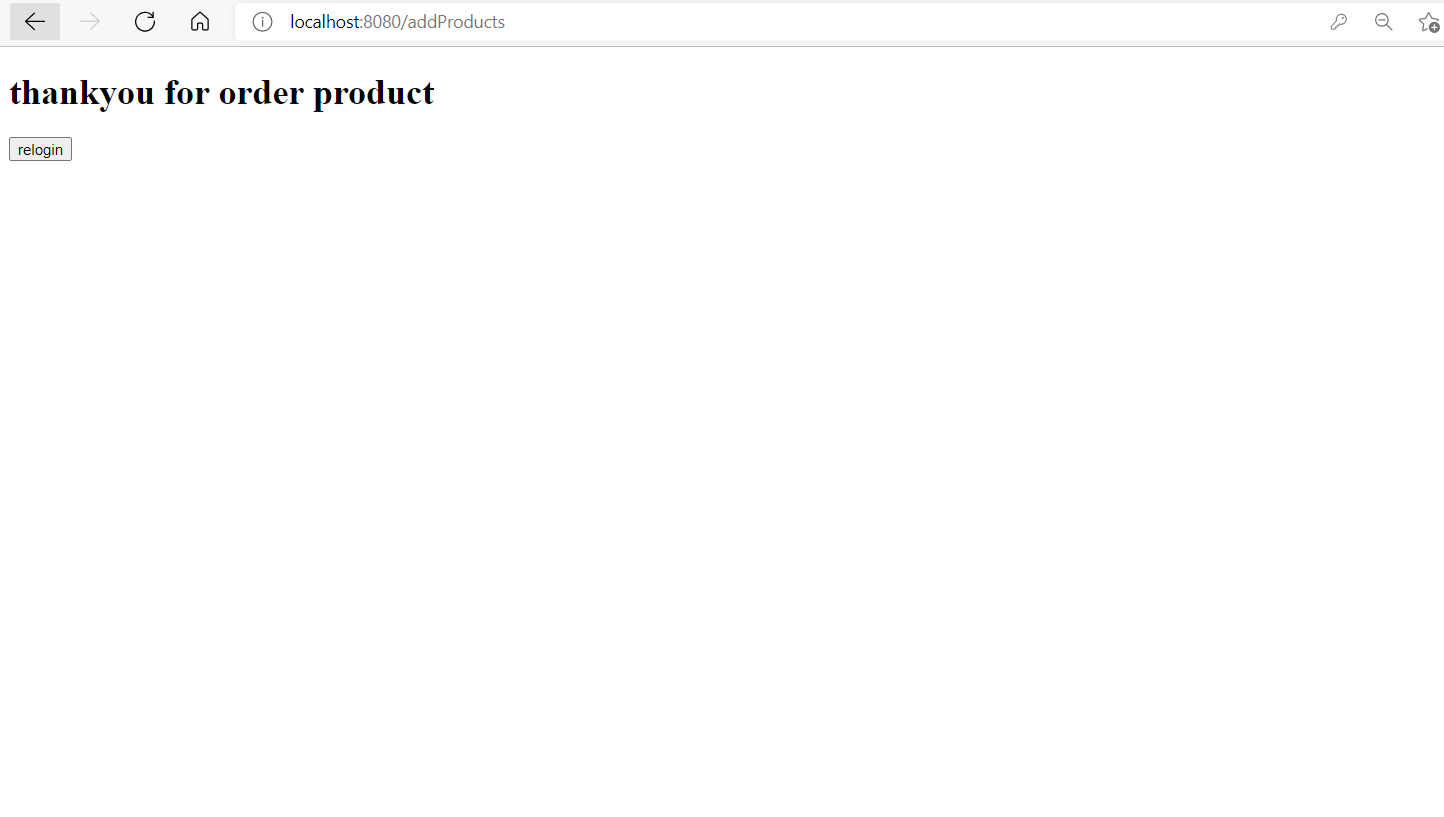
User registration:



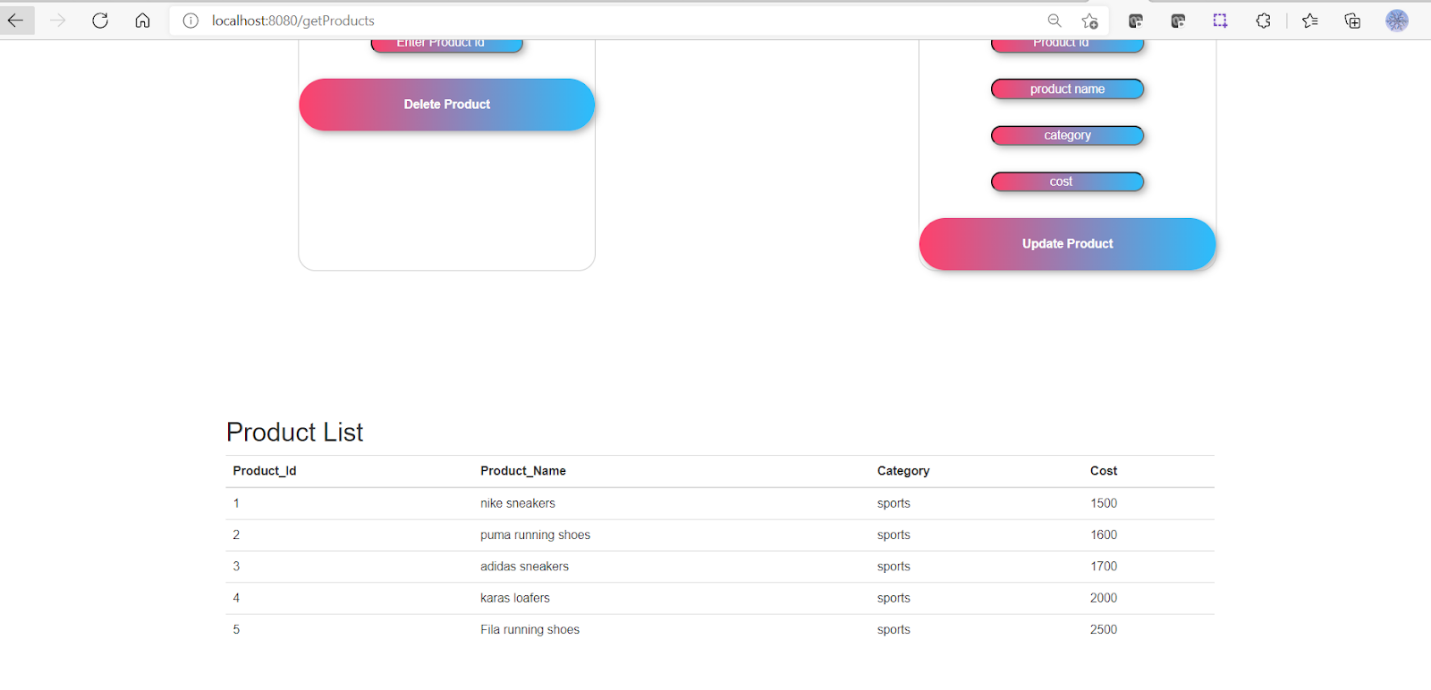
Dashboard Page:

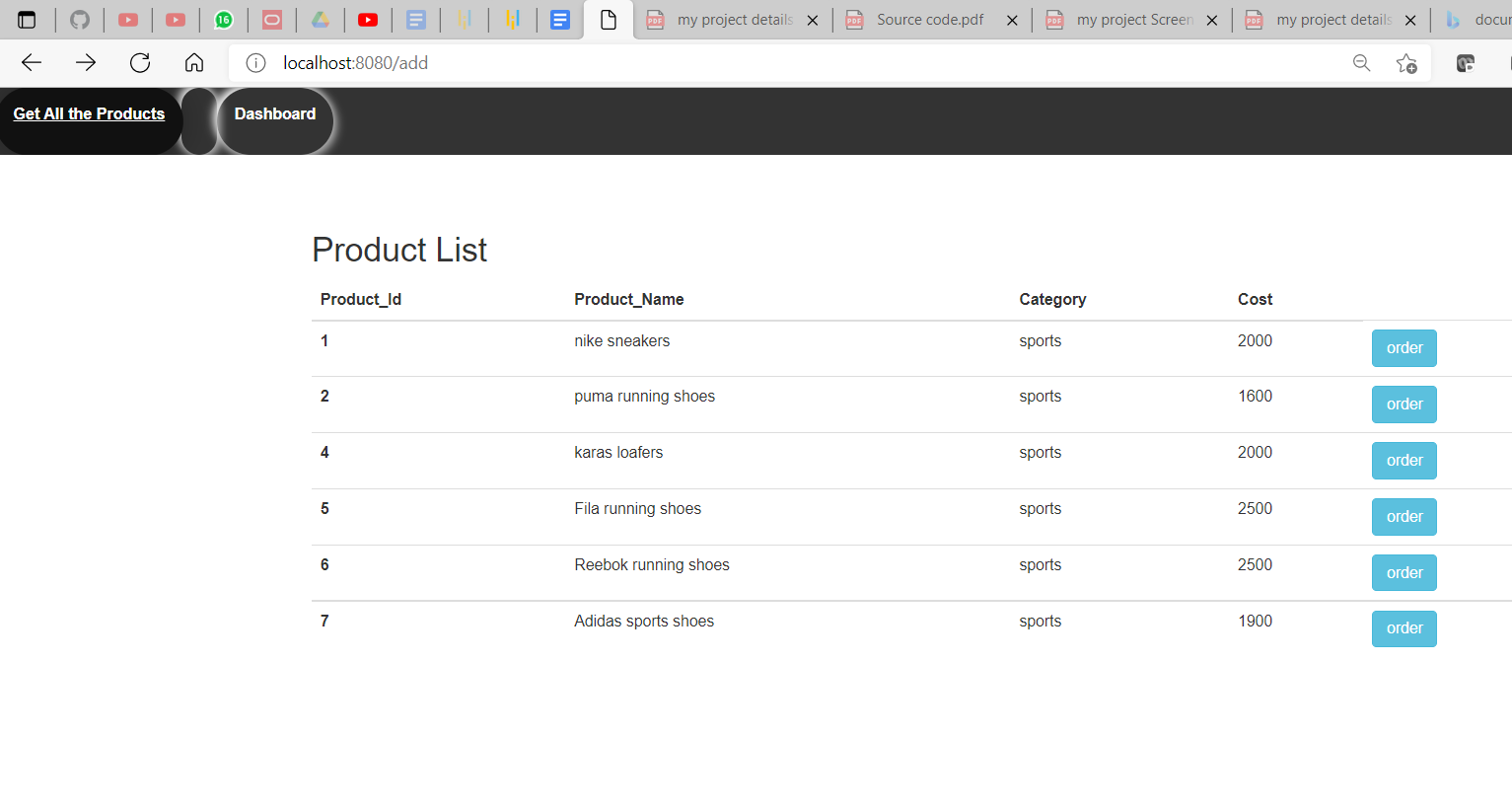


# Ordering Product:

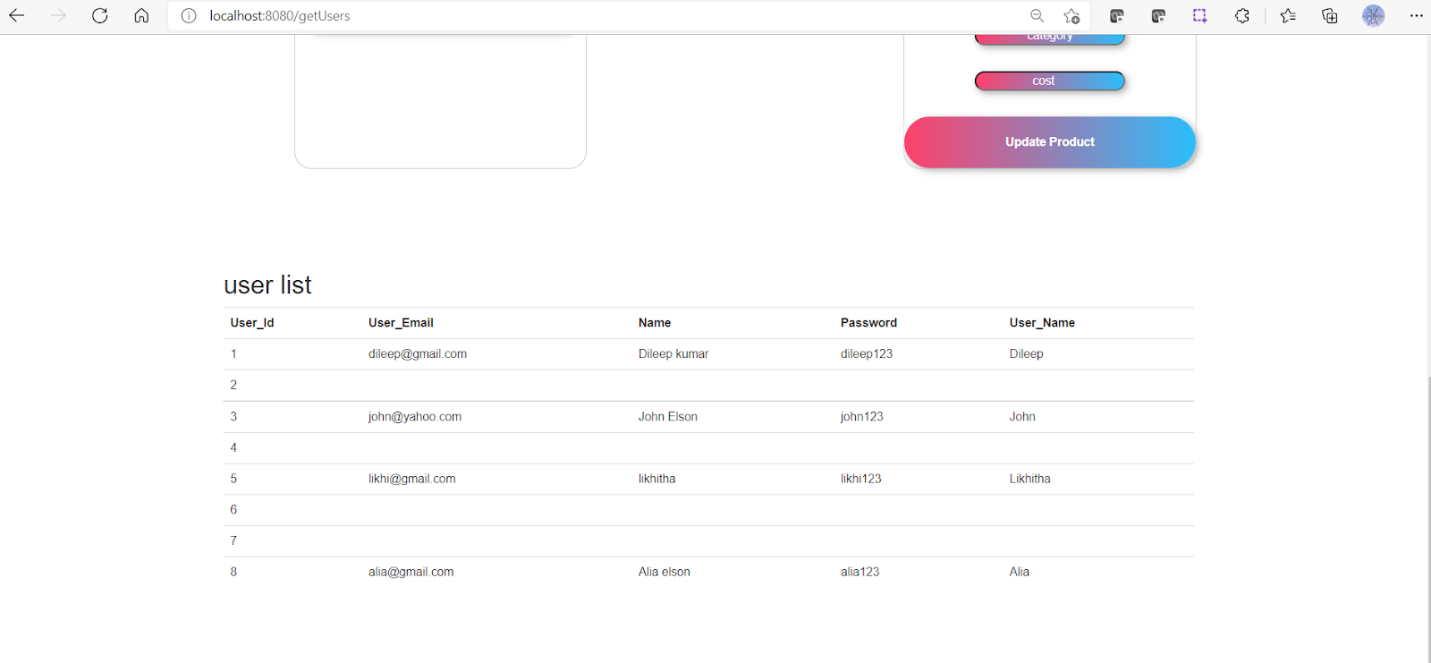


Product List

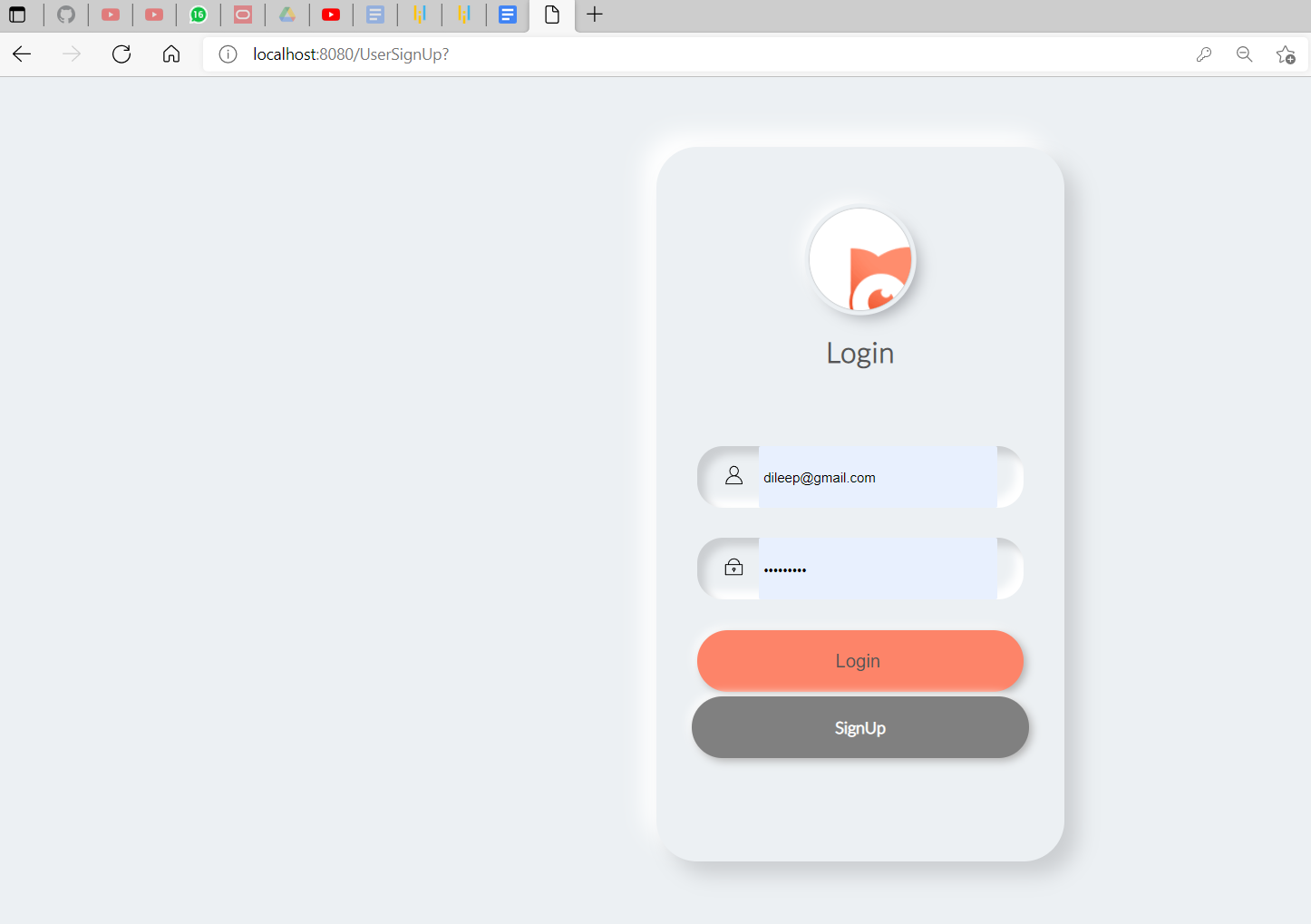




User list



ReloginPage



# **4.Pushing the code to GitHub repository**

Instructions are to be followed:

* First of all, we need account in Github to push the code in GitHub.
* Create account in GitHub.
* Create new repository.
* Open your command prompt and navigate to the folder where you have created your files.

cd

* Initialize repository using the following command:

git init

* Add all the files to your git repository using the following the command:

git add

* Commit the changes using the following command:

git commit . -m <commit message>

* Push the files to the folder you initially created using the following command:

git push -u origin main

# **5.technologies used**

Apache tomcat server 9.0.52

MySql 8.0.26

Java 1.8

Maven dependency

Eclipse/IntelliJ

Spring Boot

.