**Source Code**

Sporty Shoes E-Comerce Website

Programmer: **Luis A. Silva Soto**

Spring Boot, Spring Data

com.Bean Package

1) Category Entity

package com.bean;

import java.util.List;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.OneToMany;

@Entity

public class Category {

@Id

@GeneratedValue(strategy =GenerationType.IDENTITY)

private int cid;

private String categoryName;

@OneToMany

@JoinColumn(name="categoryID")

private List<Product> listOfProducts;

public int getCid() {

return cid;

}

public void setCid(int cid) {

this.cid = cid;

}

public String getCategoryName() {

return categoryName;

}

public void setCategoryName(String categoryName) {

this.categoryName = categoryName;

}

public List<Product> getListOfProducts() {

return listOfProducts;

}

public void setListOfProducts(List<Product> listOfProducts) {

this.listOfProducts = listOfProducts;

}

}

2)Login Users

package com.bean;

import java.util.List;

import javax.persistence.Entity;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.OneToMany;

@Entity

public class Login {

@Id

private String emailid;

private String password;

private String typeOfUser;

@OneToMany

@JoinColumn(name="emailID")

private List<Orders> listOfOrders;

public String getEmailid() {

return emailid;

}

public void setEmailid(String emailid) {

this.emailid = emailid;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public String getTypeOfUser() {

return typeOfUser;

}

public void setTypeOfUser(String typeOfUser) {

this.typeOfUser = typeOfUser;

}

@Override

public String toString() {

return "Login [emailid=" + emailid + ", password=" + password + ", typeOfUser=" + typeOfUser + "]";

}

}

3) Orders

package com.bean;

import java.time.LocalDate;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import org.springframework.format.annotation.DateTimeFormat;

@Entity

public class Orders {

@Id

@GeneratedValue(strategy =GenerationType.IDENTITY)

private int ordersID;

@DateTimeFormat(pattern="yyyy-MM-dd")

private LocalDate orderPlaced;

private Integer productID;

private String emailID;

private int quantityOfProducts;

public int getQuantityOfProducts() {

return quantityOfProducts;

}

public void setQuantityOfProducts(int quantityOfProducts) {

this.quantityOfProducts = quantityOfProducts;

}

public int getOrdersID() {

return ordersID;

}

public void setOrdersID(int ordersID) {

this.ordersID = ordersID;

}

public LocalDate getOrderPlaced() {

return orderPlaced;

}

public void setOrderPlaced(LocalDate orderPlaced) {

this.orderPlaced = orderPlaced;

}

public Integer getProductID() {

return productID;

}

public void setProductID(Integer productID) {

this.productID = productID;

}

public String getEmailID() {

return emailID;

}

public void setEmailID(String emailID) {

this.emailID = emailID;

}

@Override

public String toString() {

return "Orders [ordersID=" + ordersID + ", orderPlaced=" + orderPlaced + ", productID=" + productID

+ ", emailID=" + emailID + ", quantityOfProducts=" + quantityOfProducts + "]";

}

4)Products

package com.bean;

import java.util.List;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.Lob;

import javax.persistence.OneToMany;

@Entity

public class Product {

@Id

@GeneratedValue(strategy =GenerationType.IDENTITY)

private int pid;

private String pName;

private float price;

private int quantity;

private Integer categoryID;

@Lob

private String pImage;

@OneToMany

@JoinColumn(name="productID")

private List<Orders> listOfOrders;

public int getPid() {

return pid;

}

public List<Orders> getListOfOrders() {

return listOfOrders;

}

public void setListOfOrders(List<Orders> listOfOrders) {

this.listOfOrders = listOfOrders;

}

public void setPid(int pid) {

this.pid = pid;

}

public String getpName() {

return pName;

}

public void setpName(String pName) {

this.pName = pName;

}

public float getPrice() {

return price;

}

Com Controller

5) Category Controller

package com.controller;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

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

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

import com.bean.Category;

import com.service.CategoryService;

import java.util.List;

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

@Controller

public class CategoryController {

@Autowired

CategoryService categoryService;

@RequestMapping(value="/adminHomepages",method =RequestMethod.GET)

public String back(Model mm,Category cc) {

mm.addAttribute("category", cc);

return "adminHomepage";

}

@RequestMapping(value="/addCategoryPage",method =RequestMethod.GET)

public String openAddCategoryPage(Model mm,Category cc) {

mm.addAttribute("category", cc);

return "addCategory";

}

@RequestMapping(value="/storeCategoryInfo",method =RequestMethod.POST)

public String storeAddCategoryPage(Model mm,Category cc) {

String temp = categoryService.storeCategory(cc);

mm.addAttribute("category", cc);

mm.addAttribute("msg", temp);

return "addCategory";

}

@RequestMapping(value="/viewCategoryPage",method =RequestMethod.GET)

public String viewCategories (Model mm,Category cc) {

List<Category> listOfCategories = categoryService.findAllCategory();

mm.addAttribute("category", listOfCategories);

return "viewCategory";

}

6) Login Controller

package com.controller;

import java.util.List;

import javax.servlet.http.HttpSession;

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

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

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

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

import com.bean.Login;

import com.service.LoginService;

@Controller

public class LoginController {

@Autowired

LoginService loginService;

@RequestMapping(value = "/", method = RequestMethod.GET)

public String open(Model rr, Login tt) {

rr.addAttribute("login", tt);

return "index";

}

@RequestMapping(value = "/signIn", method = RequestMethod.POST)

public String signin(Model rr, Login tt, HttpSession x) {

String h = loginService.signIn(tt);

System.out.println(h);

rr.addAttribute("login", tt);

if(h.equals("User logged in Successfully")) {

x.setAttribute("emailid", tt.getEmailid()); // to store session.

return "customerHomepage";

}else if(h.equals("Administrator logged in successfully")) {

return "adminHomepage";

}else {

return "index";

}

}

@RequestMapping(value = "/registerCustomer", method = RequestMethod.GET)

public String openRegisterCustomer(Model rr, Login tt) {

String h = loginService.signIn(tt);

System.out.println(h);

rr.addAttribute("login", tt);

return "registerCustomer";

}

@RequestMapping(value = "/registerCustomer",method = RequestMethod.POST)

public String registerCustomer(Model rr, Login tt) {

String h = loginService.register(tt);

rr.addAttribute("msg", h);

rr.addAttribute("login", tt);

System.out.println(h);

return "registerCustomer";

}

@RequestMapping(value = "/viewUsers",method = RequestMethod.GET)

public String viewUsers(Model rr, Login tt) {

List<Login> listOfUsers = loginService.viewAllLoginDetails();

rr.addAttribute("login", listOfUsers);

return "viewUsers";

}

@RequestMapping(value = "/adminHomepage",method = RequestMethod.GET)

public String back(Model rr, Login tt) {

rr.addAttribute("login", tt);

return "adminHomepage";

}

@RequestMapping(value = "/searchUsers",method = RequestMethod.POST)

public String searchUsers(Model rr, Login tt) {

String h = loginService.searchUser(tt);

rr.addAttribute("msg", h);

rr.addAttribute("login",tt);

return "searchUsers";

}

@RequestMapping(value = "/searchUsers",method = RequestMethod.GET)

public String openSearchUsers(Model rr, Login tt) {

rr.addAttribute("login",tt);

return "searchUsers";

}

}

7) Orders Controller

package com.controller;

import java.time.LocalDate;

import java.util.ArrayList;

import java.util.Iterator;

import java.util.List;

import java.util.Optional;

import javax.servlet.http.HttpSession;

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

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

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

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

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

import com.bean.Category;

import com.bean.Login;

import com.bean.Orders;

import com.bean.Product;

import com.service.OrdersService;

import com.service.ProductService;

@Controller

public class OrdersController {

@Autowired

ProductService productService;

@Autowired

OrdersService ordersService;

@RequestMapping(value = "/ordersReports",method = RequestMethod.GET)

public String viewOrders(Model rr, Orders oo) {

List<Orders> listOfOrders = ordersService.viewAllOrderDetails();

System.out.println(listOfOrders.toString());

rr.addAttribute("orders", listOfOrders);

//rr.addAttribute("orders",oo);

return "ordersReports";

}

@RequestMapping(value = "placeOrder/{pid}")

public String placeOrder(Model mm,@PathVariable("pid") int pid,HttpSession x,Orders order) {

System.out.println("Pid is "+pid);

String emailid = (String)x.getAttribute("emailid");

mm.addAttribute("orders", order);

order.setEmailID(emailid);

order.setOrderPlaced(LocalDate.now());

order.setProductID(pid);

order.setQuantityOfProducts(1);

String result = ordersService.placeOrder(order);

productService.decrementQty(pid);

List<Product> listOfProducts = productService.findAllProducts();

mm.addAttribute("products", listOfProducts);

mm.addAttribute("msg", result);

mm.addAttribute("orders",order);

return "viewProductsByCustomer";

}

@RequestMapping(value = "/ordersByDate",method = RequestMethod.POST)

public String viewOrdersDate(Model rr, Orders oo) {

//System.out.println(l.toString());

//List<Orders> listOfOrders = ordersService.viewAllOrderDetails();

//List<Orders> temp = new ArrayList<Orders>();

List <Orders> result = ordersService.findAllByID(oo.getOrderPlaced());

//System.out.println("Test: " + l.toString());

//String r = l.toString();

//List<Orders> listOfOrders = ordersService.filterByDate(l);

//List<Orders> listOfOrders = ordersService.viewAllOrderDetails();

rr.addAttribute("orders", oo);

rr.addAttribute("orders",result);

return "viewOrdersDate";

}

@RequestMapping(value = "/ordersByDate",method = RequestMethod.GET)

public String openViewOrdersDate(Model rr, Orders oo) {

//LocalDate l=oo.getOrderPlaced();

//List <Orders> result = ordersService.findAllByID(l);

//LocalDate l = oo.getOrderPlaced();

//String r = l.toString();

//System.out.println("Test: " + l.toString());

//List<Orders> listOfOrders = ordersService.filterByDate(l);

//List<Orders> listOfOrders = ordersService.viewAllOrderDetails();

rr.addAttribute("orders", oo);

//rr.addAttribute("orders",result);

return "ordersByDate";

}

@RequestMapping(value = "/ordersByCategory",method = RequestMethod.GET)

public String openViewOrdersProduct(Model rr, Product product) {

//Product product = productService.findProductById(pid);

//List <Orders> result = ordersService.viewProductCategories(pp);

rr.addAttribute("product", product);

//rr.addAttribute("orders",result);

return "ordersByCategory";

}

@RequestMapping(value = "/ordersByCategory",method = RequestMethod.POST)

public String ViewOrdersProduct(Model rr, Product product) {

//List <Orders> result = ordersService.viewPr;

rr.addAttribute("product", product);

//rr.addAttribute("orders", result);

return "openOrdersCategory";

}

//@RequestMapping(value = "/viewOrdersDate",method = RequestMethod.GET)

// public String ViewOrdersProduct1(@PathVariable("pid") int pid,Model rr) {

// Product product = productService.findProductById(pid);

//rr.addAttribute("product", product);

//rr.addAttribute("orders", result);

//return "viewOrdersDate";

//}

@RequestMapping(value = "/openOrdersCategory/{pid}",method = RequestMethod.GET)

public String openUpdateForm(@PathVariable("pid") int pid, Model mm) {

Product product = productService.findProductById(pid);

List <Orders> result = ordersService.viewProductCategories(product);

mm.addAttribute("product", product); // store product object with name product

mm.addAttribute("orders",result);

return "viewOrdersDate";

}

}

//@RequestMapping(value = "/viewOrdersDate",method = RequestMethod.GET)

//public String viewOrdersDateD(Model rr, Orders oo) {

// rr.addAttribute("orders", oo);

//LocalDate l = oo.getOrderPlaced();

//System.out.println("Test: " + l.toString());

//String r = l.toString();

// List<Orders> listOfOrders = ordersService.viewAllOrderDetails();

//rr.addAttribute("orders", oo);

//LocalDate l = oo.getOrderPlaced();

//System.out.println("Test: " + l.toString());

//rr.addAttribute("orders", listOfOrders);

//return "viewOrdersDate";

//}

//@RequestMapping(value = "/viewOrdersDate",method = RequestMethod.GET)

//public String openSearchUsers(Model rr, Orders oo) {

//rr.addAttribute("orders",oo);

//return "ordersByDate";

//}

//@RequestMapping(value = "/ordersByDate",method = RequestMethod.POST)

//public String ordersByDate(Model rr, Orders oo) {

//List<Orders> listOfOrders = ordersService.viewAllOrderDetails();

//rr.addAttribute("orders", listOfOrders);

//return "viewOrdersDate";

//}

8) Product Controller

package com.controller;

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.RequestMapping;

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

import com.bean.Category;

import com.bean.Product;

import com.service.CategoryService;

import com.service.ProductService;

@Controller

public class ProductController {

@Autowired

ProductService productService;

@Autowired

CategoryService categoryService;

@RequestMapping(value="/adminHomepagesP",method =RequestMethod.GET)

public String back1(Model mm,Product pp) {

mm.addAttribute("product", pp);

return "adminHomepage";

}

@RequestMapping(value="/addProductPage",method =RequestMethod.GET)

public String openAddProductPage(Model mm,Product pp) {

mm.addAttribute("product", pp);

List<Category> listOfCategory = categoryService.findAllCategory();

mm.addAttribute("category", listOfCategory);

return "addProduct";

}

@RequestMapping(value="/storeProductInfo",method =RequestMethod.POST)

public String storeProductPage(Model mm,Product pp) {

String temp = productService.storeProduct(pp);

mm.addAttribute("product", pp);

mm.addAttribute("msg", temp);

List<Category> listOfCategory = categoryService.findAllCategory();

mm.addAttribute("category", listOfCategory);

return "addProduct";

}

@RequestMapping(value = "/viewProductDetailsByCustomer",method = RequestMethod.GET)

public String viewProduct(Model mm, Product pp) {

List<Product> listOfProducts = productService.findAllProducts();

mm.addAttribute("products", listOfProducts);

return "viewProductsByCustomer";

}

}

com.Repository

9) Category Repository

package com.repository;

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

import org.springframework.stereotype.Repository;

import com.bean.Category;

@Repository

public interface CategoryRepository extends JpaRepository<Category, Integer>{

}

10) Login Repository

package com.repository;

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

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

import org.springframework.stereotype.Repository;

import org.springframework.data.repository.query.Param;

import com.bean.Login;

@Repository

public interface LoginRepository extends JpaRepository<Login,String>{

@Query("select l from Login l where l.emailid = :emailid and l.password = :password and l.typeOfUser = :typeOfUser")

public Login signin(@Param("emailid") String emailid,

@Param("password") String password,

@Param ("typeOfUser") String typeofuser);

}

11) Orders Repository

package com.repository;

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

import org.springframework.stereotype.Repository;

import com.bean.Orders;

@Repository

public interface OrdersRepository extends JpaRepository<Orders, Integer>{

//@Query(value="select \* from orders l where l.ordersid = :ordersid and l.emailid = :emailid and l.order\_placed=:order\_placed and l.productid=:productid and l.quantity\_of\_products=:quantity\_of\_products",nativeQuery = true)

//public Orders report(@Param("ordersid") int ordersid,

// @Param("emailid") String emailid,

//@Param("order\_placed") LocalDate order\_placed, @Param("productid") int productid, @Param("quantity\_of\_products") int quantity\_of\_products);

//@Query(value="select \* from orders l where l.order\_placed = :order\_placed ",nativeQuery = true)

//public Orders dateReport(@Param("ordersid") int ordersid,

// @Param("emailid") String emailid,

//@Param("order\_placed") LocalDate order\_placed, @Param("productid") int productid, @Param("quantity\_of\_products") int quantity\_of\_products);

//@Query(value="select \* from orders l where l.order\_placed = :orderPlaced ",nativeQuery = true)

//public List<Orders> findByOrderDate(@Param("ordersID") int ordersid,@Param("orderPlaced") LocalDate orderPlaced,@Param("productID") Integer productid,@Param("emailID") String emailid,@Param("quantityOfProducts") int quantityOfProducts );

//List<Orders> findByOrderPlacedIn(LocalDate localDate );

//@Repository

//public interface OrdersRepository1 extends JpaRepository<Orders,OrdersKey>{

//@Query(value="select \* from orders l where l.ordersid=:ordersID or l.order\_placed = :orderPlaced ",nativeQuery = true)

//public Orders findAll(@Param("ordersID") Integer ordersID, @Param("orderPlaced") LocalDate orderPlaced);

//@Param("ordersID") Integer ordersID,

//}

}

12) Product Repository

package com.repository;

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

import org.springframework.stereotype.Repository;

import com.bean.Product;

@Repository

public interface ProductRepository extends JpaRepository<Product, Integer>{

}

com.Service

13) Category Service

package com.service;

import java.util.List;

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

import org.springframework.stereotype.Service;

import com.bean.Category;

import com.repository.CategoryRepository;

@Service

public class CategoryService {

@Autowired

CategoryRepository categoryRepository;

public String storeCategory(Category category) {

try {

categoryRepository.save(category);

}catch (Exception e) {

System.out.println("There was an error please try again");

}

return "Category Created succesfully";

}

public List<Category> findAllCategory(){

return categoryRepository.findAll();

}

}

14) Login Service

package com.service;

import java.util.Optional;

import java.util.List;

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

import org.springframework.stereotype.Service;

import com.bean.Login;

import com.repository.LoginRepository;

@Service

public class LoginService {

@Autowired

LoginRepository loginRepository;

public String register(Login login) {

Optional<Login> temp = loginRepository.findById(login.getEmailid());

if (temp.isPresent()) {

return "Email is already in use by a user account, please Login instead";

}else {

try {

loginRepository.save(login);

}catch(Exception e) {

System.out.println("There was an issue with the registration, please try again later");

}

return " Account created successfully";

}

}

public String signIn(Login login) {

Login temp2 = loginRepository.signin(login.getEmailid(), login.getPassword(), login.getTypeOfUser());

if (temp2 == null) {

return "Account does not exist, please create one";

}else {

if (temp2.getTypeOfUser().equalsIgnoreCase("Administrator")) {

return "Administrator logged in successfully";

}else {

return "User logged in Successfully";

}

}

}

public List<Login> viewAllLoginDetails() {

return loginRepository.findAll();

}

public String searchUser(Login login) {

Optional<Login> temp1 = loginRepository.findById(login.getEmailid());

if (temp1.isPresent()) {

return "Account found ! User with email ID " + login.getEmailid() + " is Registered in the website";

}else {

return "Account not found";

}

}

}

15) Orders Service

package com.service;

import java.time.LocalDate;

import java.util.ArrayList;

import java.util.List;

import java.util.Optional;

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

import org.springframework.stereotype.Service;

import com.bean.Orders;

import com.bean.Product;

import com.repository.OrdersRepository;

//import com.repository.OrdersRepository1;

import com.repository.ProductRepository;

@Service

public class OrdersService {

@Autowired

OrdersRepository ordersRepository;

@Autowired

ProductRepository productRepository;

public String placeOrder(Orders order) {

ordersRepository.save(order);

return "Order Placed successfully";

}

public List<Orders> viewAllOrderDetails() {

return ordersRepository.findAll();

}

public List<Orders> findAllByID(LocalDate l) {

//int x=1;

List<Orders> oo = ordersRepository.findAll();

Optional<Orders> temp ;

//ordersRepository.findById(x);

List<Orders> result = new ArrayList<> ();

Orders o ;

//String b;

//temp.get();

//result.add(o);

//System.out.println(temp.toString());

//System.out.println(result.toString());

for (int i=1;i< oo.size();i++) {

temp = ordersRepository.findById(i);

o=temp.get();

LocalDate d = o.getOrderPlaced();

System.out.println(d.toString());

//String a = d.toString();

//String b= l.toString();

//System.out.println(a);

//System.out.println(b);

Integer a = d.compareTo(l);

if (a.equals(0)) {

result.add(o);

}

//System.out.println(l.toString());

//result.add(temp1);

//result.add(kk);

}

//if (result.isEmpty()) {

//System.out.println(" Empty !!!");

//}//else {

//System.out.println(result.toString());

//temp1.addAll(result);

System.out.println(result.toString());

return result;

}

//List<Orders> result = ordersRepository1.findAll(x, l);

public List<Orders> viewProductCategories(Product product) {

Optional<Product> result = productRepository.findById(product.getPid());

Product u;

Product p = result.get();

List<Orders> oo = ordersRepository.findAll();

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

List<Orders> temp2 = new ArrayList<Orders> ();

for(int i =1;i <temp.size()+1;i++) {

Optional<Product> temp3 = productRepository.findById(i);

u = temp3.get();

Integer r = u.getPid();

if (u.getCategoryID().equals(p.getCategoryID())) {

for (int j=1;j< oo.size();j++) {

Optional<Orders> temp4 = ordersRepository.findById(j);

Orders o = temp4.get();

Integer t = o.getProductID();

if (t.equals(r)) {

temp2.add(o);

}

}

}

}

return temp2;

}

//public Orders findOrdersById(int orderID) {

//Optional<Orders> result = ordersRepository.findById(orderID);

//if(result.isPresent()) {

//Orders order = result.get();

//return order;

//}else {

//return null;

//}

//}

//public List<Orders> filterByDate() {

//List<Orders> ordersDate = ordersRepository.findByOrderDate(order.getOrdersID(), order.getOrderPlaced(), order.getProductID(), order.getEmailID(), order.getQuantityOfProducts());

//Set<Orders> orderDate = (Set<Orders>) ordersRepository.findByOrderPlacedIn(order.getOrderPlaced());r

//return ordersRepositoryDate.findAllByOrderPlaced();

//}

//ArrayList<Orders> temp = (ArrayList<Orders>) ordersRepository.findAll();

//List<Orders> temp1 ;

//for (int i =0 ;i < temp.size();i++) {

//temp[i]

//if (temp[i].

//}

//return ordersRepository.findAll();

//}

//public Orders ordersByCategory(Orders order) {

//Optional<Orders> temp = ordersRepository.findById(order.getProductID());

//if (temp.isPresent()) {

//Orders o = temp.get();

//return o;

//}else {

//return null;

//}

}

16) Product Service

package com.service;

import java.util.List;

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

import org.springframework.stereotype.Service;

import java.util.Optional;

import com.bean.Product;

import com.repository.ProductRepository;

@Service

public class ProductService {

@Autowired

ProductRepository productRepository;

public String storeProduct(Product product) {

try {

productRepository.save(product);

}catch(Exception e) {

System.out.println("There was an issue storing the product, please try again later");

}

return "Product was added to the Inventory Succesfully";

}

public List<Product> findAllProducts(){

return productRepository.findAll();

}

public void decrementQty(int pid) {

Optional<Product> result = productRepository.findById(pid);

if(result.isPresent()) {

Product p = result.get();

p.setQuantity(p.getQuantity()-1);

productRepository.saveAndFlush(p); // like update

}

}

public Product findProductById(int pid) {

Optional<Product> result = productRepository.findById(pid);

if(result.isPresent()) {

Product p = result.get();

return p;

}else {

return null;

}

}

}

com

17) Sporty shoes App Aplication

package com;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.boot.autoconfigure.domain.EntityScan;

import org.springframework.data.jpa.repository.config.EnableJpaRepositories;

@SpringBootApplication(scanBasePackages = "com")

@EnableJpaRepositories(basePackages = "com.repository")

@EntityScan(basePackages = "com.bean")

public class SportyShoesPhase3AppApplication {

public static void main(String[] args) {

SpringApplication.run(SportyShoesPhase3AppApplication.class, args);

System.out.println("spring boot up");

}

}

Templates

18) Add Category

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

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "<http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>">

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

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<title>Insert title here</title>

</head>

<body>

<span th:text="${msg}"></span>

<h2> Add Category </h2>

<form action="#" th:action="@{/storeCategoryInfo}" th:object="${category}" method="post">

<label>Category Name</label>

<input type="text" th:field="\*{categoryName}"/><br/>

<input type="submit" value="Store Category"/>

</form>

<br/>

<a th:href = "@{adminHomepage}">Back</a>

</body>

</html>

19) Add Product

<?xml version="1.0" encoding="ISO-8859-1" ?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "<http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>">

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

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<title>Insert title here</title>

</head>

<body>

<span th:text="${msg}"></span>

<h2>Add Product</h2>

<form action="#" th:action="@{/storeProductInfo}" th:object="${product}" method="post">

<label>Product Name</label>

<input type="text" th:field="\*{pName}"/><br/>

<label>Price</label>

<input type="number" th:field="\*{price}"/><br/>

<label> Quantity </label>

<input type="text" th:field="\*{quantity}"/><br/>

<label>Category</label>

<select th:field="\*{categoryID}">

<option value="">--Select Category--</option>

<option th:each="cc:${category}" th:value="${cc.cid}">

<span th:text="${cc.categoryName}"></span>

</option>

</select><br/>

<label>Product Image</label>

<input type="text" th:field="\*{pImage}"/><br/>

<input type="submit" value="Store Product"/>

</form>

<br/>

<a th:href="@{/adminHomepage}">Back</a>

</body>

</html>

20) Admin Homepage

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

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "<http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>">

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

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<title>Insert title here</title>

</head>

<body>

<h2> Admin Homepage</h2>

<a th:href="@{/viewUsers}">View All Users</a> |

<a th:href="@{/searchUsers}">Search for a specific User</a> |

<a th:href="@{/addCategoryPage}">Add a Category</a> |

<a th:href="@{/viewCategoryPage}">View all Categories</a> |

<a th:href="@{/addProductPage}">Add a Product</a> |

<a th:href="@{/ordersReports}">View all Orders placed</a> |

<a th:href="@{/ordersByDate}">View report of orders by Date</a> |

<a th:href="@{/ordersByCategory}">View report of orders by Category of the Products</a> |

<a th:href="@{/}">Logout</a> |

</body>

</html>

21)Customer Homepage

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

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "<http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>">

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

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<title>Insert title here</title>

</head>

<body>

<h2>Customer Homepage!</h2>

<a th:href="@{/viewProductDetailsByCustomer}">View available Products</a> |

</body>

</html>

22) Index

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

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "<http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>">

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

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<title>Insert title here</title>

</head>

<body>

<h2> Sporty Shoes ecomerce website (Phase3 project)</h2>

<form action="#" th:action="@{/signIn}" th:object="${login}" method="post">

<label>EmailId</label>

<input type="email" th:field="\*{emailid}"/><br/>

<label>Password</label>

<input type="password" th:field="\*{password}"/><br/>

<label>Type of user</label><br/>

<input type="radio" value="administrator" th:field="\*{typeOfUser}"/>Administrator

<input type="radio" value="customer" th:field="\*{typeOfUser}"/>Customer

<br/>

<input type="submit" value="Sign In"/>

</form>

<br/>

<a th:href="@{/registerCustomer}">Register Customer</a>

</body>

</html>

23) Orders by Categories

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

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "<http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>">

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

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<title>Insert title here</title>

</head>

<body>

<h2>Filter orders by Category ID of Products</h2>

<form action="#" th:action="@{/ordersByCategory/}" th:object="${product}" method="post">

<label>Product ID</label>

<input type="pid" th:field="\*{pid}"/><br/>

<td><a th:href="@{/openOrdersCategory/{pid} (pid=${product.pid}) }">Submit</a></td>

</form>

<br/>

<a th:href="@{/adminHomepage}">Back</a>

</body>

</html>

24) Orders by Date

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

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "<http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>">

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

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<title>Insert title here</title>

</head>

<body>

<h2>Filter orders by Date</h2>

<form action="#" th:action="@{/ordersByDate}" th:object="${orders}" method="post">

<label>Date in yyyy-mm-dd format</label>

<input type="orderPlaced" th:field="\*{orderPlaced}"/><br/>

<input type="submit" value="Search"/>

</form>

<br/>

<a th:href="@{/adminHomepage}">Back</a>

</body>

</html>

25) Orders Report

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

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "<http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>">

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

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<title>Insert title here</title>

</head>

<body>

<table border="1">

<tr>

<th>Order ID</th>

<th>Date of Order</th>

<th>Product Ordered ID</th>

<th>Email ID of the User</th>

<th>Quantity of products ordered</th>

</tr>

<tr th:each="oo:${orders}">

<td th:text="${oo.ordersID}"></td>

<td th:text="${oo.orderPlaced}"></td>

<td th:text="${oo.productID}"></td>

<td th:text="${oo.emailID}"></td>

<td th:text="${oo.quantityOfProducts}"></td>

</tr>

</table>

<br/>

<a th:href="@{/adminHomepage}">Back</a>

</body>

</html>

26) Register Customer

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

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "<http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>">

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

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<title>Insert title here</title>

</head>

<body>

<h2> Account Customer creation</h2>

<form action="#" th:action="@{/registerCustomer}" th:object="${login}" method="post">

<label>EmailId</label>

<input type="email" th:field="\*{emailid}"/><br/>

<label>Password</label>

<input type="password" th:field="\*{password}"/><br/>

<label>Type of User</label><br/>

<input type="radio" value="customer" th:field="\*{typeOfUser}"/>Customer

<br/>

<input type="submit" value="Register"/>

</form>

<span th:text="${msg}"></span>

<br/>

<a th:href="@{/}">Login Page</a> |

</body>

</html>

27) Search Users

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

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "<http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>">

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

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<title>Insert title here</title>

</head>

<body>

<h2>Search a specific user with Email ID</h2>

<form action="#" th:action="@{/searchUsers}" th:object="${login}" method="post">

<label>EmailId</label>

<input type="email" th:field="\*{emailid}"/><br/>

<input type="submit" value="Search"/>

</form>

<br/>

<span th:text="${msg}"></span>

<br/>

<a th:href = "@{adminHomepage}">Back</a>

</body>

</html>

28) View Category

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

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "<http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>">

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

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<title>Insert title here</title>

</head>

<body>

<table border="1">

<tr>

<th>CategoryId</th>

<th>Category Name</th>

</tr>

<tr th:each="cc:${category}">

<td th:text="${cc.cid}"></td>

<td th:text="${cc.categoryName}"></td>

</tr>

</table>

<br/>

<a th:href="@{/adminHomepage}">Back</a>

</body>

</html>

29) View Orders Date

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

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "<http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<title>Insert title here</title>

</head>

<body>

<table border="1">

<tr>

<th>Order ID</th>

<th>Date of Order</th>

<th>Product Ordered ID</th>

<th>Email ID of the User</th>

<th>Quantity of products ordered</th>

</tr>

<tr th:each="oo:${orders}">

<td th:text="${oo.ordersID}"></td>

<td th:text="${oo.orderPlaced}"></td>

<td th:text="${oo.productID}"></td>

<td th:text="${oo.emailID}"></td>

<td th:text="${oo.quantityOfProducts}"></td>

</tr>

</table>

<br/>

<a th:href="@{/adminHomepage}">Back</a>

</body>

</html>

30) View Products- Purchase by customers

<?xml version="1.0" encoding="ISO-8859-1" ?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "<http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>">

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

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<title>Insert title here</title>

</head>

<body>

<h2> All Products in the store</h2>

<table border="1">

<tr>

<th>PId</th>

<th>PName</th>

<th>Price</th>

<th>Total Quantity in stock</th>

<th>Image URL</th>

<th>Place Order</th>

</tr>

<tr th:each="product:${products}">

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

<td th:text="${product.pName}"></td>

<td th:text="${product.price}"></td>

<td th:text="${product.quantity}"></td>

<td><img th:src="${product.pImage}" width="50px" height="50px"/></td>

<td><a th:href="@{/placeOrder/{pid} (pid=${product.pid}) }">Place Order</a></td>

</tr>

</table>

<br/>

<a th:href="@{/customerHomepage}">Back</a>

</body>

</html>

31) View Users

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

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "<http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>">

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

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<title>Insert title here</title>

</head>

<body>

<table border="1">

<tr>

<th>Email ID</th>

<th>Type of User</th>

</tr>

<tr th:each="tt:${login}">

<td th:text="${tt.emailid}"></td>

<td th:text="${tt.typeOfUser}"></td>

</tr>

</table>

<br/>

<a th:href="@{/adminHomepage}">Back</a>

</body>

</html>