|  |  |
| --- | --- |
| **Full Name** | **Shahnawaz Hashmi** |
| **Batch** | **Apr 2022** |
| **Project Title** | **E-commerce Website for Sporty Shoes** |
| **Project Submission Date** | **15-Aug – 2022** |
| **Git Hub Link** |  |

|  |
| --- |
| Source Code :- |
| **package** com.ss;  **import** org.springframework.boot.SpringApplication;  **import** org.springframework.boot.autoconfigure.SpringBootApplication;  @SpringBootApplication  **public** **class** Phase3ProjectApplication {  **public** **static** **void** main(String[] args) {  SpringApplication.*run*(Phase3ProjectApplication.**class**, args);  }  }  package com.ss.controller;  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.GetMapping;  import org.springframework.web.bind.annotation.PostMapping;  import org.springframework.web.bind.annotation.RequestParam;  import com.ss.model.Admin;  import com.ss.service.AdminService;  @Controller  public class AdminController {    @Autowired  private AdminService adminService;    @PostMapping("/verifyLogin")  public String verifyLogin(@RequestParam(name="username") String username,@RequestParam(name="password") String password,HttpSession session,Model model) {  if(!username.isEmpty() || !password.isEmpty()) {  if(adminService.loginVerify(username,password)) {  session.setAttribute("uname", username);  return "adminDashboard";  }  else {  model.addAttribute("action","Username or password wrong");  return "admin\_login";  }  }else {  model.addAttribute("action", "Fields must not be empty");  return "admin\_login";  }    }    @GetMapping("/getDashboard")  public String getDashboard() {  return "adminDashboard";  }    @GetMapping("/changePassword")  public String changeAdminPassword(HttpSession session, Model model) {  String username=(String) session.getAttribute("uname");  Admin admin = adminService.getAdmin(username);  model.addAttribute("admin", admin);  return "change\_password";  }    @PostMapping("/updatePassword")  public String updatePassword(@RequestParam(name="oldPassword") String oldPassword,@RequestParam(name="newPassword") String newPassword,HttpSession session,Model model) {  String username=(String) session.getAttribute("uname");  Admin admin = adminService.getAdmin(username);  if(oldPassword.equals(admin.getPassword())) {  admin.setPassword(newPassword);  adminService.updatePassword(admin);  model.addAttribute("action", "Password changed Successfully");  return "adminDashboard";  }else {  model.addAttribute("action", "Old Password not matching");  return "change\_password";  }    }    @GetMapping("/logout")  public String adminLogout(HttpSession session) {  session.invalidate();  return "redirect:/";  }  }  package com.ss.controller;  //import java.sql.Date;  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.ExceptionHandler;  import org.springframework.web.bind.annotation.GetMapping;  import org.springframework.web.bind.annotation.PostMapping;  import org.springframework.web.bind.annotation.RequestParam;  import com.ss.model.Cart;  import com.ss.model.Customer;  import com.ss.model.Product;  import com.ss.model.Purchase;  import com.ss.service.CartService;  import com.ss.service.CustomerService;  import com.ss.service.PurchaseService;  @Controller  public class CartController {    @Autowired  private CartService cartService;    @Autowired  private CustomerService customerService;    @Autowired  private PurchaseService purchaseService;    @ExceptionHandler(Exception.class)  public String handleSqlException(Exception e, HttpSession session) {  session.setAttribute("action", "Choose Payment before Buying");  return "redirect:/viewCart";  }  @PostMapping("/confirmCart")  public String addToCart(@RequestParam("size") float size,@RequestParam("quantity") int quantity,HttpSession session) {  Cart cart = new Cart();  Product product = (Product) session.getAttribute("product");  int min=100;int max=999;int b = (int)(Math.random()\*(max-min+1)+min);  cart.setId(b);  cart.setProductId(product.getId());  cart.setQuantity(quantity);  cart.setPrice(product.getPrice()\*quantity);  cart.setSize(size);  cartService.saveCart(cart);  session.setAttribute("action", "Product added to cart");  float temp=0;  if(session.getAttribute("sessionCost")==null) {  temp=0;  }else {  temp=(float) session.getAttribute("sessionCost");  }  float sessionCost=(cart.getPrice()+temp);  session.setAttribute("sessionCost", sessionCost);  return "redirect:/";  }    @GetMapping("/viewCart")  public String viewCart(Model model,HttpSession session) {  List<Cart> cartList = cartService.getAllCart();  if(!cartList.isEmpty()) {  model.addAttribute("cartList", cartList);  model.addAttribute("action", session.getAttribute("action"));  session.setAttribute("action", null);  return "viewCart";  }else {  session.setAttribute("action", "No products currently in Cart");  return "redirect:/";  }  }    @PostMapping("/buyNow")  public String buyProducts(@RequestParam("pm") String pm, HttpSession session) {  System.out.println(pm);  if(pm.equals("yes")) {  List<Cart> cartList = cartService.getAllCart();  Purchase purchase = new Purchase();  String email = (String) session.getAttribute("customerLogin");  Customer customer = customerService.getCustomer(email);  for(Cart cl:cartList) {  java.sql.Date date = new java.sql.Date(new java.util.Date().getTime());  int min=100000;int max=999999;int b = (int)(Math.random()\*(max-min+1)+min);  purchase.setId(b);  purchase.setDop(date);  System.out.println(date);  purchase.setCustomer(customer);  purchase.setProductid(cl.getProductId());  purchase.setQuantity(cl.getQuantity());  purchase.setTotalcost(cl.getPrice());  purchaseService.addPurchase(purchase);  }  session.setAttribute("action", "Products added to Customer Order List Sucessfully");  return "redirect:/";  }else {  session.setAttribute("action", "Make Payment before to finilize orders");  return "redirect:/viewCart";  }  }    }  package com.ss.controller;  import java.sql.SQLException;  import java.util.List;  import java.util.regex.Matcher;  import java.util.regex.Pattern;  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.ExceptionHandler;  import org.springframework.web.bind.annotation.GetMapping;  import org.springframework.web.bind.annotation.PathVariable;  import org.springframework.web.bind.annotation.PostMapping;  import org.springframework.web.bind.annotation.RequestParam;  import com.ss.model.Customer;  import com.ss.model.Purchase;  import com.ss.service.CartService;  import com.ss.service.CustomerService;  import com.ss.service.PurchaseService;  @Controller  public class CustomerController {  @Autowired  private CustomerService customerService;  @Autowired  private PurchaseService purchaseService;    @Autowired  private CartService cartService;  @ExceptionHandler(SQLException.class)  public String handleSqlException(SQLException e, HttpSession session) {  session.setAttribute("action", "User can't be deleted until their orders are deleted");  return "redirect:/manageCustomer";  }  @PostMapping("/saveCustomer")  public String saveCustomer(Customer customer, Model model, HttpSession session) {  List<String> cEmails = customerService.customerEmails();  boolean notExist = true;  for(String e : cEmails) {  if(customer.getEmail().equals(e))  notExist=false;  }  if(notExist) {  if (validate(customer.getEmail())) {  customerService.saveCustomer(customer);  model.addAttribute("action", "Added successfully, login to shop");  session.setAttribute("customerLogin", customer.getEmail());  session.setAttribute("custName", customer.getName());  cartService.cartDeleteAll();  return "redirect:/";  } else {  model.addAttribute("action", "Email pattern doesn't match");  return "new\_customer";  }  }else {  session.setAttribute("action", "Entered Email Already Exist please Login");  return "redirect:/";  }    }  @PostMapping("/verifyCustLogin")  public String verifyLogin(@RequestParam(name = "email") String email,  @RequestParam(name = "password") String password, HttpSession session, Model model) {  if (!email.isEmpty() || !password.isEmpty()) {  if (customerService.loginVerify(email, password)) {  session.setAttribute("customerLogin", email);  Customer customer = customerService.getCustomer(email);  session.setAttribute("custName", customer.getName());  cartService.cartDeleteAll();  return "redirect:/";  } else {  model.addAttribute("action", "email or password wrong");  return "customer\_login";  }  } else {  model.addAttribute("action", "Fields must not be empty");  return "customer\_login";  }  }  @GetMapping("/customerLogout")  public String customerLogout(HttpSession session) {  cartService.cartDeleteAll();  session.invalidate();  return "redirect:/";  }  @GetMapping("/manageCustomer")  public String manageCustomer(Model model,HttpSession session) {  model.addAttribute("action", session.getAttribute("action"));  session.setAttribute("action", null);  model.addAttribute("customers", customerService.getAllCustomers());  return "manageCustomer";  }  @GetMapping("/deleteCustomer/{email}")  public String deleteCustomer(@PathVariable(name = "email") String email, Model model) {  customerService.deleteCustomer(email);  model.addAttribute("action", "Customer Deleted Sucessfully");  return "redirect:/manageCustomer";  }  @GetMapping("/customerOrders/{email}")  public String customerOrders(@PathVariable(name = "email") String email, Model model,HttpSession session) {  List<Purchase> sPurchase = purchaseService.getByEmail(email);  if(!sPurchase.isEmpty()) {  model.addAttribute("sPurchase", sPurchase);  return "customerPurchase";  }else {  session.setAttribute("action", "No Active Orders/Purchases by Customer");  return "redirect:/manageCustomer";  }  }    @PostMapping("/searchCustomer")  public String searchCustomer(@RequestParam("keyword") String keyword,Model model) {  List<Customer> sCustomer = customerService.searchCustomer(keyword);  if(sCustomer.isEmpty()) {  model.addAttribute("action", "No Customer found");  model.addAttribute("customers", customerService.getAllCustomers());  return "manageCustomer";  }else {  model.addAttribute("searchHeading","Entered Catogery");  model.addAttribute("sCustomer", sCustomer);  return "searchCustomer";  }    }  public static final Pattern VALID\_EMAIL\_ADDRESS\_REGEX = Pattern.compile("^[A-Z0-9.\_%+-]+@[A-Z0-9.-]+\\.[A-Z]{2,6}$",  Pattern.CASE\_INSENSITIVE);  public static boolean validate(String emailStr) {  Matcher matcher = VALID\_EMAIL\_ADDRESS\_REGEX.matcher(emailStr);  return matcher.find();  }  }  package com.ss.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.GetMapping;  import org.springframework.web.bind.annotation.PathVariable;  import org.springframework.web.bind.annotation.PostMapping;  import org.springframework.web.bind.annotation.RequestParam;  import com.ss.model.Admin;  import com.ss.model.Cart;  import com.ss.model.Customer;  import com.ss.model.Product;  import com.ss.model.Purchase;  import com.ss.service.ProductService;  import com.ss.service.PurchaseService;  //import com.ss.service.PurchaseService;  @Controller  public class MainApp {    @Autowired  private ProductService productService;    @Autowired  private PurchaseService purchaseService;    @GetMapping("/")  public String viewHomePage(Model model,HttpSession session) {  model.addAttribute("action", session.getAttribute("action"));  session.setAttribute("action", null);  if(session.getAttribute("productList")==null) {  session.setAttribute("productList", productService.getAllProducts());  session.setAttribute("searchH", null);  }  return "home";  }    @GetMapping("/goHome")  public String goHome(Model model,HttpSession session) {  model.addAttribute("action", session.getAttribute("action"));  session.setAttribute("action", null);  session.setAttribute("productList", productService.getAllProducts());  session.setAttribute("searchH", null);  return "home";  }    @PostMapping("/searchHome")  public String searchHome(@RequestParam("keyword") String keyword,Model model,HttpSession session) {  model.addAttribute("action", session.getAttribute("action"));  session.setAttribute("action", null);  List<Product> productList = productService.homeSearch(keyword);  if(productList.isEmpty()) {  session.setAttribute("action", "Currently no products for searched");  session.setAttribute("productList", null);  return "redirect:/";  }  session.setAttribute("productList", productList);  session.setAttribute("searchH", "yes");  return "home";  }      @GetMapping("/register")  public String register(Model model) {  Customer customer = new Customer();  model.addAttribute("customer", customer);  return "new\_customer";  }    @GetMapping("/login")  public String customerLogin(Model model) {  Customer customer = new Customer();  model.addAttribute("customer", customer);  return "customer\_login";  }    @GetMapping("/adminLogin")  public String adminLogin(Model model) {  Admin admin = new Admin();  model.addAttribute("admin",admin);  return "admin\_login";  }    @GetMapping("/addCart/{id}")  public String selectProduct(@PathVariable("id") int id,HttpSession session,Model model) {  if(session.getAttribute("customerLogin")==null) {  session.setAttribute("action", "Login or Register to start shopping");  return "redirect:/";  }else {  session.setAttribute("product", productService.getProductById(id));  Cart cart = new Cart();  model.addAttribute("cart", cart);  return "addCart";  }  }    @GetMapping("/viewOrders/{email}")  public String customerOrders(@PathVariable(name = "email") String email, Model model,HttpSession session) {  List<Purchase> sPurchase = purchaseService.getByEmail(email);  if(!sPurchase.isEmpty()) {  model.addAttribute("sPurchase", sPurchase);  return "ViewOrders";  }else {  session.setAttribute("action", "No Active Orders/Purchases by Customer");  return "redirect:/";  }  }  }  package com.ss.controller;  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.GetMapping;  import org.springframework.web.bind.annotation.ModelAttribute;  import org.springframework.web.bind.annotation.PathVariable;  import org.springframework.web.bind.annotation.PostMapping;  import com.ss.model.Product;  import com.ss.service.ProductService;  @Controller  public class ProductController {    @Autowired  private ProductService productService;    @GetMapping("/manageProduct")  public String manageProduct(Model model) {  model.addAttribute("products",productService.getAllProducts());  Product product = new Product();  model.addAttribute("product", product);  return "manageProduct";  }    @PostMapping("/addProduct")  public String addProduct(@ModelAttribute("product") Product product, Model model, HttpSession session) {  int min=10000;int max=99999;int b = (int)(Math.random()\*(max-min+1)+min);  product.setId(b);  productService.addProduct(product);  session.setAttribute("action","Product Added succesfully");  model.addAttribute("product", product);  return "redirect:/manageProduct";  }    @GetMapping("/showProductUpdate/{id}")  public String showProductUpdate(@PathVariable(value="id") int id, Model model) {  Product product = productService.getProductById(id);  model.addAttribute("product", product);  return "update\_product";  }    @PostMapping("/updateProduct")  public String updateProduct(@ModelAttribute("product") Product product, Model model,HttpSession session) {  productService.addProduct(product);  session.setAttribute("action","Product Updated succesfully");  model.addAttribute("product", product);  return "redirect:/manageProduct";  }    @GetMapping("/deleteProduct/{id}")  public String deleteProduct(@PathVariable(value="id") int id,Model model,HttpSession session) {  productService.deleteProduct(id);  session.setAttribute("action", "Product Deleted Succesfully");  Product product = new Product();  model.addAttribute("product", product);  return "redirect:/manageProduct";  }  }  package com.ss.controller;  import java.text.SimpleDateFormat;  import java.sql.Date;  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.PathVariable;  import org.springframework.web.bind.annotation.PostMapping;  import org.springframework.web.bind.annotation.RequestParam;  import com.ss.model.Purchase;  import com.ss.service.PurchaseService;  @Controller  public class PurchaseController {    @Autowired  private PurchaseService purchaseService;    @GetMapping("/managePurchase")  public String managePurchase(Model model) {  model.addAttribute("purchases", purchaseService.getAllPurchases());  return "managePurchase";  }    @PostMapping("/searchPurchaseDate")  public String searchPurchaseDate(@RequestParam("keyword") String keyword,Model model) {  Date date=null;  try {  //DateFormat parser = new SimpleDateFormat("yyyy-MM-dd");  date = new Date(new SimpleDateFormat("yyyy-MM-dd").parse(keyword).getTime());  }catch(Exception e) { System.out.println(e); }  List<Purchase> sPurchase = purchaseService.getPurchaseByDate(date);  if(sPurchase.isEmpty()) {  model.addAttribute("action", "No purchases on the selected date");  model.addAttribute("purchases", purchaseService.getAllPurchases());  return "managePurchase";  }else {  model.addAttribute("searchHeading","selected Date");  model.addAttribute("sPurchase", sPurchase);  return "searchPurchase";  }    }    @PostMapping("/searchPurchaseCategory")  public String searchPurchaseCategory(@RequestParam("keyword") String keyword,Model model) {  List<Purchase> sPurchase = purchaseService.getPurchaseByCategory(keyword);  if(sPurchase.isEmpty()) {  model.addAttribute("action", "No purchases on the Entered Category");  model.addAttribute("purchases", purchaseService.getAllPurchases());  return "managePurchase";  }else {  model.addAttribute("searchHeading","Entered Catogery");  model.addAttribute("sPurchase", sPurchase);  return "searchPurchase";  }    }    @GetMapping("/deletePurchase/{id}")  public String deletePurchase(@PathVariable("id") int id,Model model) {  purchaseService.deletePurchase(id);  model.addAttribute("action", "Purchase Deleted Succesfully");  return "redirect:/managePurchase";  }  }  **package** com.ss.model;  **import** javax.persistence.Entity;  **import** javax.persistence.Id;  @Entity  **public** **class** Admin {  @Id  **private** String username;  **private** String password;    **public** Admin() {  **super**();  }  **public** Admin(String username, String password) {  **super**();  **this**.username = username;  **this**.password = password;  }  **public** String getUsername() {  **return** username;  }  **public** **void** setUsername(String username) {  **this**.username = username;  }  **public** String getPassword() {  **return** password;  }  **public** **void** setPassword(String password) {  **this**.password = password;  }  }  package com.ss.model;  import javax.persistence.Entity;  import javax.persistence.Id;  @Entity  public class Cart {    @Id  private int id;  private int productId;  private int quantity;  private float size;  private float price;    public Cart() {  super();  }    public Cart(int id, int productId, int quantity, float size, float price) {  super();  this.id = id;  this.productId = productId;  this.quantity = quantity;  this.size = size;  this.price = price;  }  public int getId() {  return id;  }  public void setId(int id) {  this.id = id;  }  public int getProductId() {  return productId;  }  public void setProductId(int productId) {  this.productId = productId;  }  public int getQuantity() {  return quantity;  }  public void setQuantity(int quantity) {  this.quantity = quantity;  }  public float getSize() {  return size;  }  public void setSize(float size) {  this.size = size;  }  public float getPrice() {  return price;  }  public void setPrice(float price) {  this.price = price;  }      }  package com.ss.model;  import javax.persistence.Entity;  import javax.persistence.Id;  @Entity  public class Customer {  @Id  private String email;  private String name;  private String password;  private long contact;    public Customer() {  super();  }    public Customer(String email, String name, String password, long contact) {  super();  this.email = email;  this.name = name;  this.password = password;  this.contact = contact;  }    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;  }  public long getContact() {  return contact;  }  public void setContact(long contact) {  this.contact = contact;  }      }  package com.ss.model;  import javax.persistence.Entity;  import javax.persistence.Id;  @Entity  public class Product {  @Id  private int id;  private String company;  private String name;  private float price;  private String category;    public Product(int id, String company, String name, float price, String category) {  super();  this.id = id;  this.company = company;  this.name = name;  this.price = price;  this.category = category;  }  public Product() {  super();  }  public int getId() {  return id;  }  public void setId(int id) {  this.id = id;  }  public String getCompany() {  return company;  }  public void setCompany(String company) {  this.company = company;  }  public String getName() {  return name;  }  public void setName(String name) {  this.name = name;  }  public float getPrice() {  return price;  }  public void setPrice(float price) {  this.price = price;  }  public String getCategory() {  return category;  }  public void setCategory(String category) {  this.category = category;  }      }  package com.ss.model;  import java.sql.Date;  import javax.persistence.Entity;  import javax.persistence.Id;  import javax.persistence.OneToOne;  import com.ss.model.Customer;  @Entity  public class Purchase {  @Id  private int id;  private float size;  private Date dop;  private int quantity;  private float totalcost;  private int productid;  @OneToOne  private Customer customer;    public Purchase() {  super();  }  public Purchase(int id, float size, Date dop, int quantity, float totalcost, int productid, Customer customer) {  super();  this.id = id;  this.size = size;  this.dop = dop;  this.quantity = quantity;  this.totalcost = totalcost;  this.productid = productid;  this.customer = customer;  }    public int getId() {  return id;  }  public void setId(int id) {  this.id = id;  }  public float getSize() {  return size;  }  public void setSize(float size) {  this.size = size;  }  public Date getDop() {  return dop;  }  public void setDop(Date dop) {  this.dop = dop;  }  public int getQuantity() {  return quantity;  }  public void setQuantity(int quantity) {  this.quantity = quantity;  }  public float getTotalcost() {  return totalcost;  }  public void setTotalcost(float totalcost) {  this.totalcost = totalcost;  }  public int getProductid() {  return productid;  }  public void setProductid(int productid) {  this.productid = productid;  }  public Customer getCustomer() {  return customer;  }  public void setCustomer(Customer customer) {  this.customer = customer;  }      }  package com.ss.repository;  import java.util.List;  import org.springframework.data.jpa.repository.JpaRepository;  import org.springframework.data.jpa.repository.Query;  import com.ss.model.Admin;  public interface AdminRepository extends JpaRepository<Admin,String>{  Admin findByUsername(String username);    @Query(value="SELECT username FROM Admin",nativeQuery=true)  public List<String> findUsenames();  }  package com.ss.repository;  import org.springframework.data.jpa.repository.JpaRepository;  import com.ss.model.Cart;  public interface CartRepository extends JpaRepository<Cart, Integer>{  }  package com.ss.repository;  import java.util.List;  import org.springframework.data.jpa.repository.JpaRepository;  import org.springframework.data.jpa.repository.Query;  import com.ss.model.Customer;  public interface CustomerRepository extends JpaRepository<Customer, String> {  Customer findByEmail(String email);    @Query("SELECT c FROM Customer c WHERE c.email LIKE %?1%"  +" OR c.name LIKE %?1%"  +" OR c.contact LIKE %?1%")  public List<Customer> userSearch(String name);    @Query("SELECT c.email from Customer c")  public List<String> customerEmails();  }  package com.ss.repository;  import java.util.List;  import org.springframework.data.jpa.repository.JpaRepository;  import org.springframework.data.jpa.repository.Query;  import com.ss.model.Product;  public interface ProductRepository extends JpaRepository<Product, Integer> {  Product findById(int id);    @Query("Select id from Product where category Like %?1%")  List<Integer> getByCategory(String keyword);    @Query("SELECT p FROM Product p WHERE p.company LIKE %?1%"  +" OR p.name LIKE %?1%"  +" OR p.price LIKE %?1%"  +" OR p.category LIKE %?1%")  public List<Product> homeSearch(String keyword);  }  package com.ss.repository;  import java.sql.Date;  import java.util.List;  import org.springframework.data.jpa.repository.JpaRepository;  import org.springframework.data.jpa.repository.Query;  import com.ss.model.Purchase;  public interface PurchaseRepository extends JpaRepository<Purchase, Integer>{    @Query("SELECT p FROM Purchase p WHERE p.customer.email LIKE %?1%")  public List<Purchase> getByEmail(String email);    public List<Purchase> findByDop(Date dop);  public List<Purchase> findByproductid(int id);  }  **package** com.ss.model;  **import** javax.persistence.Entity;  **import** javax.persistence.Id;  @Entity  **public** **class** Admin {  @Id  **private** String username;  **private** String password;    **public** Admin() {  **super**();  }  **public** Admin(String username, String password) {  **super**();  **this**.username = username;  **this**.password = password;  }  **public** String getUsername() {  **return** username;  }  **public** **void** setUsername(String username) {  **this**.username = username;  }  **public** String getPassword() {  **return** password;  }  **public** **void** setPassword(String password) {  **this**.password = password;  }  }  package com.ss.service;  import java.util.List;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.stereotype.Service;  import com.ss.model.Cart;  import com.ss.repository.CartRepository;  @Service  public class CartService {    @Autowired  private CartRepository cartRepository;  public void saveCart(Cart cart) {  cartRepository.save(cart);  }    public List<Cart> getAllCart() {  return cartRepository.findAll();  }    public void cartDeleteAll() {  cartRepository.deleteAll();  }  }  package com.ss.service;  import java.util.List;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.stereotype.Service;  import com.ss.model.Customer;  import com.ss.repository.CustomerRepository;  @Service  public class CustomerService {    @Autowired  private CustomerRepository customerRepository;  public void saveCustomer(Customer customer) {  this.customerRepository.save(customer);    }  public boolean loginVerify(String email, String password) {  Customer customer = customerRepository.findByEmail(email);  if (customer!= null && customer.getEmail().equals(email) && customer.getPassword().equals(password)) {  return true;  }  return false;  }  public Customer getCustomer(String email) {  return customerRepository.findByEmail(email);  }  public List<Customer> getAllCustomers() {  return customerRepository.findAll();  }  public void deleteCustomer(String email) {  customerRepository.deleteById(email);  }  public List<Customer> searchCustomer(String keyword) {  return customerRepository.userSearch(keyword);  }    public List<String> customerEmails(){  return customerRepository.customerEmails();  }  }  **package** com.ss.service;  **public** **class** MainAppService {  }  package com.ss.service;  import java.util.List;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.stereotype.Service;  import com.ss.model.Product;  import com.ss.repository.ProductRepository;  @Service  public class ProductService {    @Autowired  private ProductRepository productRepository;    public List<Product> getAllProducts(){  return productRepository.findAll();  }  public void addProduct(Product product) {  productRepository.save(product);    }  public Product getProductById(int id) {  return productRepository.findById(id);  }  public void deleteProduct(int id) {  productRepository.deleteById(id);  }    public List<Product> homeSearch(String keyword) {  return productRepository.homeSearch(keyword);  }  }  package com.ss.service;  import java.sql.Date;  import java.util.ArrayList;  import java.util.List;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.stereotype.Service;  import com.ss.model.Purchase;  import com.ss.repository.ProductRepository;  import com.ss.repository.PurchaseRepository;  @Service  public class PurchaseService {    @Autowired  private PurchaseRepository purchaseRepository;    @Autowired  private ProductRepository productRepository;    public List<Purchase> getAllPurchases(){  return purchaseRepository.findAll();  }    public List<Purchase> getByEmail(String email){  return purchaseRepository.getByEmail(email);  }  public List<Purchase> getPurchaseByDate(Date keyword) {  return purchaseRepository.findByDop(keyword);  }    public List<Purchase> getPurchaseByCategory(String keyword) {  List<Purchase> sPurchase = new ArrayList<>();  List<Integer> productIds = productRepository.getByCategory(keyword);  if (!productIds.isEmpty()) {  for (int id : productIds) {  List<Purchase> tempList = purchaseRepository.findByproductid(id);  if (!tempList.isEmpty()) {  for (Purchase p : tempList) {  sPurchase.add(p);  }  }  }  }  return sPurchase;  }  public void deletePurchase(int id) {  purchaseRepository.deleteById(id);    }  public void addPurchase(Purchase purchase) {  purchaseRepository.save(purchase);  }  }  <!DOCTYPE html>  <html lang="en" xmlns:th="http://www.thymeleaf.org">  <head>  <meta charset="ISO-8859-1">  <title>Sporty Shoes</title>  <link rel="stylesheet"  href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"  integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"  crossorigin="anonymous">  </head>  <style>  img{  width: 200px;  height: 200px;  }  </style>  <body>  <br>  <div class="container" style="text-align: right">  <a th:href="@{/}" class="btn btn-warning btn-sm mb-3">Home</a>  </div>  <div class="container my-2">  <h1 style="color:DodgerBlue"> Cart </h1>  <form action="#" th:action="@{/confirmCart}", method="POST">  <table border="0">  <tbody>  <tr>  <td>&nbspProduct</td>  <td><img th:src="@{/Images/shoe.jpg}" alt="Image displayed here"></td>  </tr>  <tr>  <td>&nbspProduct Name</td>  <td><span th:text="${session.product.name}" style="text-align:center"></span></td>  </tr>  <tr>  <td>&nbspPrice</td>  <td><span th:text="${session.product.price}" style="text-align:center"></span></td>  </tr>  <tr>  <td>&nbspProduct Category</td>  <td><span th:text="${session.product.category}" style="text-align:center"></span></td>  </tr>  <tr>  <td>&nbspRequired Size</td>  <td><input type="text" name="size"  placeholder="size" class="form-control mb-3 col-15" required></td>  </tr>  <tr>  <td>&nbspRequired Quantity</td>  <td><input type="number" name="quantity"  th:placeholder="Quantity" class="form-control col-15" required></td>  </tr>  <tr>  <td colspan="2" style="text-align:center"><button type="submit"  class="btn btn-secondary col-10">Add to cart</button></td>  </tr>  </tbody>  </table>  </form>  </div>  <hr>  </body>  </html>  <!DOCTYPE html>  <html lang="en" xmlns:th="http://www.thymeleaf.org">  <head>  <meta charset="ISO-8859-1">  <title>Sporty Shoes</title>  <link rel="stylesheet"  href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"  integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"  crossorigin="anonymous">  </head>  <body>  <br>  <br>  <div style="text-align:center">  <h6 th:text="${action}"></h6>  </div>  <div class="container">  <h1 style="color:DodgerBlue"> Admin Login</h1>  <hr>  <h3 style="color:SlateBlue"> Enter Details</h3>  <form action="#" th:action="@{/verifyLogin}", method="POST">  <input name="username" type="text" placeholder="Username" class="form-control mb-4 col-4" required/>  <input name="password" type="password" placeholder="Password"class="form-control mb-4 col-4" required/>  <button class="btn btn-dark col-2" type="submit">Log In</button>  </form>  </div>  </body>  </html>  <!DOCTYPE html>  <html lang="en" xmlns:th="http://www.thymeleaf.org">  <head>  <meta charset="ISO-8859-1">  <title>Sporty Shoes</title>  <link rel="stylesheet"  href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"  integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"  crossorigin="anonymous">  </head>  <body>  <br>  <br>  <h6 style="color:red;text-align:center" th:text="${action}"></h6>  <br>  <div class="container" style="text-align:right">  <a th:href="@{/changePassword}" class="btn btn-warning btn-sm mb-3">Change Password</a>  <a th:href="@{/logout}" class="btn btn-warning btn-sm mb-3">Logout</a>  <!-- <a th:href="@{/}" class="btn btn-primary btn-sm mb-3">Main Home Page</a> -->  </div>  <br>  <br>  <hr>  <br>  <br>  <div class="container" style="text-align:center">  <a th:href="@{/manageProduct}" class="btn btn-secondary btn-sm mb-3">Administer Products</a>  <a th:href="@{/manageCustomer}" class="btn btn-secondary btn-sm mb-3">Administer Customers</a>  <a th:href="@{/managePurchase}" class="btn btn-secondary btn-sm mb-3">Administer Purchase</a>  </div>  </body>  </html>  <!DOCTYPE html>  <html lang="en" xmlns:th="http://www.thymeleaf.org">  <head>  <meta charset="ISO-8859-1">  <title>Sporty Shoes</title>  <link rel="stylesheet"  href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"  integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"  crossorigin="anonymous">  </head>  <body>  <br>  <br>  <div style="text-align:center">  <h6 th:text="${action}"></h6>  </div>  <div class="container">  <h1 style="color:Tomato">Change Admin Password</h1>  <hr>  <h6 style="color:Tomato" th:text="${action}"></h6>  <h3 style="background-color:powderblue;color:Gray" >Enter Details</h3>  <form action="#" th:action="@{/updatePassword}", method="POST">  <table border="0">  <tr>  <td><label>Username</label></td>  <td th:text="${session.uname}"></td>  </tr>  <tr>  <td>Enter Old Password</td>  <td><input type="password" name="oldPassword"  th:placeholder="..." class="form-control col-15" required></td>  </tr>  <tr>  <td>Enter New Password</td>  <td><input type="password" name="newPassword"  th:placeholder="..." class="form-control col-15" required></td>  </tr>  </table>  <input type="submit" value="Change Password">  </form>  </div>  </body>  </html>  <!DOCTYPE html>  <html lang="en" xmlns:th="http://www.thymeleaf.org">  <head>  <meta charset="ISO-8859-1">  <title>Sporty Shoes</title>  <link rel="stylesheet"  href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"  integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"  crossorigin="anonymous">  </head>  <body>  <br>  <br>  <div style="text-align:center">  <h6 th:text="${action}"></h6>  </div>  <div class="container">  <h1 style="color:DodgerBlue">Customer Login</h1>  <hr>  <form action="#" th:action="@{/verifyCustLogin}", method="POST">  <input name="email" type="text" placeholder="Email" class="form-control mb-4 col-4" required/>  <input name="password" type="password" placeholder="Password"class="form-control mb-4 col-4" required/>  <button class="btn btn-dark col-2" type="submit">Log In</button>  </form>  </div>  </body>  </html>  <!DOCTYPE html>  <html lang="en" xmlns:th="http://www.thymeleaf.org">  <head>  <meta charset="ISO-8859-1">  <title>Sporty Shoes</title>  <link rel="stylesheet"  href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"  integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"  crossorigin="anonymous">  </head>  <body>  <br>  <div class="container" style="text-align: right">  <a th:href="@{/manageCustomer}" class="btn btn-warning btn-sm mb-3">Back to Administer Customer</a>  </div>  <hr>  <div class="container my-2">  <h1 style="color:DodgerBlue"><span>Active Orders </span></h1>  <h6 th:text="${action}"></h6>  <span th:text="${session.action}"  style="text-align: center; color: red"></span>  <table border="0" class="table table-striped table-responsive-md">  <thead>  <tr>  <th>Order Id</th>  <th>Product Id</th>  <th>Purchase Date</th>  <th>Size</th>  <th>Quantity</th>  <th>Total Cost</th>  <th>Ordered By</th>  <th>Customer Name</th>  </tr>  </thead>  <tbody>  <tr th:each="purchase : ${sPurchase}">  <td th:text="${purchase.id}"></td>  <td th:text="${purchase.productid}"></td>  <td th:text="${purchase.dop}"></td>  <td th:text="${purchase.size}"></td>  <td th:text="${purchase.quantity}"></td>  <td th:text="${purchase.totalcost}"></td>  <td th:text="${purchase.customer.email}"></td>  <td th:text="${purchase.customer.name}"></td>  </tr>  </tbody>  </table>  </div>  </body>  </html>  <!DOCTYPE html>  <html lang="en" xmlns:th="http://www.thymeleaf.org">  <head>  <meta charset="ISO-8859-1">  <title>Sporty Shoes</title>  <link rel="stylesheet"  href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"  integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"  crossorigin="anonymous">  </head>  <style>  img{  width: 200px;  height: 150px;  }  .containerSC {  position: absolute;  left: 58px;  top: 20px;  display: inline-block;  /\* border: 2px solid red; \*/  }  </style>  <body>  <br>  <br>  <br>  <div class="container" style="text-align:right" th:if="${session.customerLogin == null}">  <a th:href="@{/register}" class="btn btn-warning btn-sm mb-3">Sign up</a>  <a th:href="@{/login}" class="btn btn-success btn-sm mb-3">Sign in</a>  <a th:href="@{/adminLogin}" class="btn btn-danger btn-sm mb-3">Admin Login</a>  <br>  <h4 style="color:Tomato">Sign up to start shopping get special news and offers</h4>  </div>  <div class="container" style="text-align:right" th:if="${session.customerLogin != null}">  <div class="containerSC">  <form th:action="@{/searchHome}" method="POST">  <input type="text" name="keyword" Placeholder="Search Products">  <button type="submit" class="btn btn-dark btn-sm mb-3" style="margin-top:12px">Search</button>  </form>  </div>  <div th:if="${session.searchH != null}">  <a th:href="@{/goHome}" class="btn btn-primary btn-sm mb-3">Home</a>  </div>  <a th:href="@{/customerLogout}" class="btn btn-warning btn-sm mb-3">Logout</a>  <a th:href="@{/viewCart}" class="btn btn-warning btn-sm mb-3">View Cart</a>  <a th:href="@{/viewOrders/{email}(email=${session.customerLogin})}" class="btn btn-warning btn-sm mb-3">View Active Orders</a>  </div>  <div style="text-align:center">  </div>  <hr>  <div class="container">  <span>Hi, </span><span th:text="${session.custName}"></span>  <h1 style="font-family:verdana;">Sporty Shoes</h1>  <h5 th:text="${action}" style="color: red"></h5>  <div class="row">  <div class="col" th:each="product : ${session.productList}" style="padding-bottom:15px">  <table border="0">  <tr>  <td colspan="2"><img th:src="@{/Images/shoe.jpg}" alt="Image displayed here"></td>  </tr>  <tr>  <td>Company</td>  <td th:text="${product.company}"></td>  </tr>  <tr>  <td>Name</td>  <td th:text="${product.name}"></td>  </tr>  <tr>  <td>Price</td>  <td th:text="${product.price}"></td>  </tr>  <tr>  <td>Category</td>  <td th:text="${product.category}"></td>  </tr>  <tr>  <td colspan="2" style="text-align:center"><a th:href="@{/addCart/{id}(id=${product.id})}" class="btn btn-secondary">Add to Cart</a></td>  </tr>  </table>  </div>  </div>  </div>  </body>  </html>  <!DOCTYPE html>  <html lang="en" xmlns:th="http://www.thymeleaf.org">  <head>  <meta charset="ISO-8859-1">  <title>Sporty Shoes</title>  <link rel="stylesheet"  href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"  integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"  crossorigin="anonymous">  </head>  <style>  .containerSC {  position: absolute;  left: 58px;  top: 20px;  display: inline-block;  /\* border: 2px solid red; \*/  }  </style>  <body>  <br>  <div class="containerSC">  <form th:action="@{/searchCustomer}" method="POST">  <input type="text" name="keyword" Placeholder="Search customer">  <button type="submit" class="btn btn-dark btn-sm mb-3" style="margin-top:12px">Search</button>  </form>  </div>  <div class="container" style="text-align: right">  <a th:href="@{/getDashboard}" class="btn btn-warning btn-sm mb-3">Dashboard</a>  </div>  <hr>  <div class="container my-2">  <h1 style="color:DodgerBlue">Customers</h1>  <h6 th:text="${action}"></h6>  <span th:text="${session.action}" style="text-align:center;color:red"></span>  <table border="0" class="table table-striped table-responsive-md">  <thead>  <tr>  <th>Email</th>  <th>Name</th>  <th>Contact</th>  <th>Actions</th>  </tr>  </thead>  <tbody>  <tr th:each="customer : ${customers}">  <td th:text="${customer.email}"></td>  <td th:text="${customer.name}"></td>  <td th:text="${customer.contact}"></td>  <td><a th:href="@{/deleteCustomer/{email}(email=${customer.email})}"  class="btn btn-danger">Delete</a>  <a th:href="@{/customerOrders/{email}(email=${customer.email})}"  class="btn btn-success">View Customer orders</a></td>  </tr>  </tbody>  </table>  </div>  </body>  </html>  <!DOCTYPE html>  <html lang="en" xmlns:th="http://www.thymeleaf.org">  <head>  <meta charset="ISO-8859-1">  <title>Sporty Shoes</title>  <link rel="stylesheet"  href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"  integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"  crossorigin="anonymous">  </head>  <body>  <br>  <div class="container" style="text-align: right">  <a th:href="@{/getDashboard}" class="btn btn-warning btn-sm mb-3">Dashboard</a>  </div>  <div class="container my-2">  <h1 style="color:DodgerBlue">Add a new Product</h1>  <form action="#" th:action="@{/addProduct}" th:object="${product}"  , method="POST">  <table border="0">  <tbody>  <tr>  <td>&nbspManufacture Company Name</td>  <td><input type="text" th:field="\*{company}"  placeholder="Company Name" class="form-control col-15" required></td>  </tr>  <tr>  <td>&nbspProduct Name</td>  <td><input type="text" th:field="\*{name}"  placeholder="Product name" class="form-control col-15" required></td>  </tr>  <tr>  <td>&nbspPrice</td>  <td><input type="text" th:field="\*{price}"  placeholder="Price" class="form-control col-15" required></td>  </tr>  <tr>  <td>&nbspProduct Category</td>  <td><input type="text" th:field="\*{category}"  placeholder="Category" class="form-control col-15" required></td>  </tr>  <tr>  <td colspan="2" style="text-align:center"><button type="submit"  class="btn btn-dark col-10">Add Product</button></td>  </tr>  </tbody>  </table>  </form>  </div>  <hr>    <div class="container my-2">  <h1>Product List</h1>  <span th:text="${session.action}" style="text-align:center;color:red"></span>  <table border="0" class="table table-striped table-responsive-md">  <thead>  <tr>  <th>Id</th>  <th>Company Name</th>  <th>Product name</th>  <th>Price</th>  <th>Category</th>  <th>Actions</th>  </tr>  </thead>  <tbody>  <tr th:each="product : ${products}">  <td th:text="${product.id}"></td>  <td th:text="${product.company}"></td>  <td th:text="${product.name}"></td>  <td th:text="${product.price}"></td>  <td th:text="${product.category}"></td>  <td><a th:href="@{/showProductUpdate/{id}(id=${product.id})}"  class="btn btn-success">Update</a> <a  th:href="@{/deleteProduct/{id}(id=${product.id})}"  class="btn btn-danger">Delete</a></td>  </tr>  </tbody>  </table>  </div>  </body>  </html>  <!DOCTYPE html>  <html lang="en" xmlns:th="http://www.thymeleaf.org">  <head>  <meta charset="ISO-8859-1">  <title>Sporty Shoes</title>  <link rel="stylesheet"  href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"  integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"  crossorigin="anonymous">  </head>  <style>  .containerSD {  position: absolute;  left: 58px;  top: 20px;  display: inline-block;  /\* border: 2px solid red; \*/  }  .containerSC {  position: absolute;  left: 400px;  top: 20px;  display: inline-block;  /\* border: 2px solid r  /\* border: 2px solid green; \*/  }  </style>  <body>  <br>  <div class="container" style="text-align: right">  <a th:href="@{/getDashboard}" class="btn btn-warning btn-sm mb-3">Dashboard</a>  </div>  <div class="containerSD">  <form th:action="@{/searchPurchaseDate}" method="POST">  <input type="date" name="keyword" Placeholder="YYYY-MM-DD">  <button type="submit" class="btn btn-warning btn-sm mb-3" style="margin-top:12px">Search</button>  </form>  </div>  <div class="containerSC">  <form th:action="@{/searchPurchaseCategory}" method="POST">  <input type="text" name="keyword" Placeholder="Product Category">  <button type="submit" class="btn btn-warning btn-sm mb-3" style="margin-top:12px">Search</button>  </form>  </div>  <hr>  <div class="container my-2">  <h1 style="color:DodgerBlue"> Purchase List</h1>  <h6 th:text="${action}" style="color: red"></h6>  <span th:text="${session.action}" style="text-align: center; color: red"></span>  <table border="0" class="table table-striped table-responsive-md">  <thead>  <tr>  <th>Order Id</th>  <th>Product Id</th>  <th>Purchase Date</th>  <th>Size</th>  <th>Quantity</th>  <th>Total Cost</th>  <th>Ordered By</th>  <th>Customer Name</th>  <th>Actions</th>  </tr>  </thead>  <tbody>  <tr th:each="purchase : ${purchases}">  <td th:text="${purchase.id}"></td>  <td th:text="${purchase.productid}"></td>  <td th:text="${purchase.dop}"></td>  <td th:text="${purchase.size}"></td>  <td th:text="${purchase.quantity}"></td>  <td th:text="${purchase.totalcost}"></td>  <td th:text="${purchase.customer.email}"></td>  <td th:text="${purchase.customer.name}"></td>  <td><a th:href="@{/deletePurchase/{id}(id=${purchase.id})}"  class="btn btn-danger">Delete</a></td>  </tr>  </tbody>  </table>  </div>  </body>  </html>  <html lang="en" xmlns:th="http://www.thymeleaf.org">  <head>  <meta charset="ISO-8859-1">  <title>EmployeeManagement System</title>  <link rel="stylesheet"  href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"  integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"  crossorigin="anonymous">  </head>  <body>  <div style="text-align:center">  <h6 th:text="${action}"></h6>  </div>  <div class="container">  <h1 style="background-color:#d5f4e6;color:DodgerBlue">Sporty Shoes</h1>  <hr>  <h2 style="color:SlateBlue"> Welcome, Please fill out your contact details</h2>  <form action="#" th:action="@{/saveCustomer}" th:object="${customer}", method="POST">  <input type="text" th:field="\*{email}"  placeholder="Enter Email" class="form-control mb-4 col-4" required>  <input type="text" th:field="\*{name}"  placeholder="Enter Name" class="form-control mb-4 col-4" required>  <input type="text" th:field="\*{contact}"  placeholder="Enter Contact number" class="form-control mb-4 col-4" required>  <input type="text" th:field="\*{password}"  placeholder="Enter password" class="form-control mb-4 col-4" required>  <button type="submit" class="btn btn-dark col-2">Register</button>  </form>  </div>  </body>  </html>  <!DOCTYPE html>  <html lang="en" xmlns:th="http://www.thymeleaf.org">  <head>  <meta charset="ISO-8859-1">  <title>Sporty Shoes</title>  <link rel="stylesheet"  href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"  integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"  crossorigin="anonymous">  </head>  <body>  <br>  <div class="container" style="text-align: right">  <a th:href="@{/manageCustomer}" class="btn btn-warning btn-sm mb-3">Back to Administer Customer</a>  </div>  <hr>  <div class="container my-2">  <h1 style="color:DodgerBlue">Customers</h1>  <h6 th:text="${action}"></h6>  <span th:text="${session.action}" style="text-align:center;color:red"></span>  <table border="0" class="table table-striped table-responsive-md">  <thead>  <tr>  <th>Email</th>  <th>Name</th>  <th>Contact</th>  <th>Actions</th>  </tr>  </thead>  <tbody>  <tr th:each="customer : ${sCustomer}">  <td th:text="${customer.email}"></td>  <td th:text="${customer.name}"></td>  <td th:text="${customer.contact}"></td>  <td><a th:href="@{/deleteCustomer/{email}(email=${customer.email})}"  class="btn btn-danger">Delete</a>  <a th:href="@{/customerOrders/{email}(email=${customer.email})}"  class="btn btn-success">View Customer orders</a></td>  </tr>  </tbody>  </table>  </div>  </body>  </html>  <!DOCTYPE html>  <html lang="en" xmlns:th="http://www.thymeleaf.org">  <head>  <meta charset="ISO-8859-1">  <title>Sporty Shoes</title>  <link rel="stylesheet"  href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"  integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"  crossorigin="anonymous">  </head>  <style>  .containerSD {  position: absolute;  left: 58px;  top: 20px;  display: inline-block;  /\* border: 2px solid red; \*/  }  .containerSC {  position: absolute;  left: 500px;  top: 20px;  display: inline-block;  /\* border: 2px solid r  /\* border: 2px solid green; \*/  }  </style>  <body>  <br>  <div class="container" style="text-align: right">  <a th:href="@{/managePurchase}" class="btn btn-warning btn-sm mb-3">Back to Administer Purchase</a>  </div>  <hr>  <div class="container my-2">  <h1 style="color:DodgerBlue"><span>Purchase List based on </span><span th:text="${searchHeading}"></span></h1>  <h6 th:text="${action}"></h6>  <span th:text="${session.action}"  style="text-align: center; color: red"></span>  <table border="0" class="table table-striped table-responsive-md">  <thead>  <tr>  <th>Order Id</th>  <th>Product Id</th>  <th>Purchase Date</th>  <th>Size</th>  <th>Quantity</th>  <th>Total Cost</th>  <th>Ordered By</th>  <th>Customer Name</th>  </tr>  </thead>  <tbody>  <tr th:each="purchase : ${sPurchase}">  <td th:text="${purchase.id}"></td>  <td th:text="${purchase.productid}"></td>  <td th:text="${purchase.dop}"></td>  <td th:text="${purchase.size}"></td>  <td th:text="${purchase.quantity}"></td>  <td th:text="${purchase.totalcost}"></td>  <td th:text="${purchase.customer.email}"></td>  <td th:text="${purchase.customer.name}"></td>  </tr>  </tbody>  </table>  </div>  </body>  </html>  <!DOCTYPE html>  <html lang="en" xmlns:th="http://www.thymeleaf.org">  <head>  <meta charset="ISO-8859-1">  <title>Insert title here</title>  <link rel="stylesheet"  href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"  integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"  crossorigin="anonymous">  </head>  <body>  <div class="container my-2">  <h1 style="color:DodgerBlue">Update Product</h1>  <form action="#" th:action="@{/updateProduct}" th:object="${product}"  , method="POST">  <table border="0">  <tbody>  <tr>  <td>&nbspProduct Id</td>  <td><input type="hidden" th:field="\*{id}" placeholder="${product.id}"/></td>  </tr>  <tr>  <td>&nbspManufacture Company Name</td>  <td><input type="text" th:field="\*{company}"  placeholder="${product.company}" class="form-control col-15" required></td>  </tr>  <tr>  <td>&nbspProduct Name</td>  <td><input type="text" th:field="\*{name}"  placeholder="${product.name}" class="form-control col-15" required></td>  </tr>  <tr>  <td>&nbspPrice</td>  <td><input type="text" th:field="\*{price}"  placeholder="${product.price}" class="form-control col-15" required></td>  </tr>  <tr>  <td>&nbspProduct Category</td>  <td><input type="text" th:field="\*{category}"  placeholder="${product.category}" class="form-control col-15" required></td>  </tr>  <tr>  <td colspan="2" style="text-align:center"><button type="submit"  class="btn btn-dark col-10">Add Product</button></td>  </tr>  </tbody>  </table>  </form>  </div>  </body>  </html>  <!DOCTYPE html>  <html lang="en" xmlns:th="http://www.thymeleaf.org">  <head>  <meta charset="ISO-8859-1">  <title>Sporty Shoes</title>  <link rel="stylesheet"  href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"  integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"  crossorigin="anonymous">  </head>  <body>  <br>  <div class="container" style="text-align: right">  <a th:href="@{/}" class="btn btn-warning btn-sm mb-3">Home</a>  <form th:action="@{/buyNow}", method="POST">  <label class="radio-inline">Payment Made</label>  <input type="radio" name="pm" id="yes" value="yes"/><label for="yes">YES</label>  <input type="radio" name="pm" id="no" value="no"/><label for="no">NO</label>  <button class="btn btn-secondary col-2" type="submit">Check out</button>  </form>  <!-- <a th:href="@{/buyNow}" class="btn btn-primary btn-sm mb-3">Buy all Now</a> -->  </div>  <hr>  <div class="container my-2">  <h1 style="color:DodgerBlue"><span>Total Cost = </span><span style="color:Brown" th:text="${session.sessionCost}"></span></h1>  <h6 th:text="${action}" style="text-align: center; color: red"></h6>  <span th:text="${session.action}"  style="text-align: center; color: red"></span>  <table border="0" class="table table-striped table-responsive-md">  <thead>  <tr>  <th>Product Id</th>  <th>Quantity</th>  <th>Size</th>  <th>Price</th>  </tr>  </thead>  <tbody>  <tr th:each="cart : ${cartList}">  <td th:text="${cart.productId}"></td>  <td th:text="${cart.quantity}"></td>  <td th:text="${cart.size}"></td>  <td th:text="${cart.price}"></td>  </tr>  </tbody>  </table>  </div>  </body>  </html>  <!DOCTYPE html>  <html lang="en" xmlns:th="http://www.thymeleaf.org">  <head>  <meta charset="ISO-8859-1">  <title>Sporty Shoes</title>  <link rel="stylesheet"  href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"  integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"  crossorigin="anonymous">  </head>  <body>  <br>  <div class="container" style="text-align: right">  <a th:href="@{/}" class="btn btn-warning btn-sm mb-3">Home</a>  </div>  <hr>  <div class="container my-2">  <h1 style="color:DodgerBlue"><span>Active Orders</span><span th:text="${searchHeading}"></span></h1>  <h6 th:text="${action}"></h6>  <span th:text="${session.action}"  style="text-align: center; color: tomato"></span>  <table border="0" class="table table-striped table-responsive-md">  <thead>  <tr>  <th>Order Id</th>  <th>Product Id</th>  <th>Purchase Date</th>  <th>Size</th>  <th>Quantity</th>  <th>Total Cost</th>  <th>Ordered By</th>  <th>Customer Name</th>  </tr>  </thead>  <tbody>  <tr th:each="purchase : ${sPurchase}">  <td th:text="${purchase.id}"></td>  <td th:text="${purchase.productid}"></td>  <td th:text="${purchase.dop}"></td>  <td th:text="${purchase.size}"></td>  <td th:text="${purchase.quantity}"></td>  <td th:text="${purchase.totalcost}"></td>  <td th:text="${purchase.customer.email}"></td>  <td th:text="${purchase.customer.name}"></td>  </tr>  </tbody>  </table>  </div>  </body>  </html>  server.port=8080  # DATASOURCE (DataSourceAutoConfiguration & DataSourceProperties)  spring.datasource.url=jdbc:mysql://localhost:3306/phase\_3\_project?useSSL=false&serverTimezone=UTC&useLegacyDatetimeCode=false  spring.datasource.username=root  spring.datasource.password=0000  # Hibernate  # The SQL dialect makes Hibernate generate better SQL for the chosen database  spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5InnoDBDialect  # Hibernate ddl auto (create, create-drop, validate, update)  spring.jpa.hibernate.ddl-auto = update  logging.level.org.hibernate.SQL=DEBUG  #logging.level.org.hibernate.type=TRACE |

package com.ss;

import org.junit.jupiter.api.Test;

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

@SpringBootTest

class Phase3ProjectApplicationTests {

@Test

void contextLoads() {

}

}

<?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>2.4.8</version>

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

</parent>

<groupId>com.ss</groupId>

<artifactId>phase-3-project</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>phase-3-project</name>

<description>Phase 3 simplilearn project</description>

<properties>

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

</properties>

<dependencies>

<dependency>

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

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

</dependency>

<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>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>

</dependencies>

<build>

<plugins>

<plugin>

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

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

</plugin>

</plugins>

</build>

</project>