**Sourcecode for SportyShoes**

**package** com.controller;

**import** java.util.List;

**import** javax.servlet.http.HttpServletRequest;

**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.PostMapping;

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

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

**import** com.dao.OrderDAO;

**import** com.dao.ProductDAO;

**import** com.dao.UserDAO;

**import** com.entity.Order;

**import** com.entity.Product;

**import** com.entity.User;

@Controller

@RequestMapping("/admin")

**public** **class** AdminController {

@Autowired

**private** ProductDAO productDAO;

// inject

@Autowired

**private** UserDAO userDAO;

// inject

@Autowired

**private** OrderDAO orderDAO;

@GetMapping("/products")

**public** String products(Model theModel, HttpServletRequest request) {

HttpSession session = request.getSession(**true**);

User currentUser = (User) session.getAttribute("currentUser");

List<Product> products = productDAO.getProducts();

theModel.addAttribute("currentUser", currentUser);

theModel.addAttribute("products", products);

**return** "mange-products";

}

@GetMapping("/deleteProduct")

**public** String deleteProduct(@RequestParam("productId") **int** productId, Model theModel) {

productDAO.deleteProduct(productId);

**return** "redirect:/admin/products";

}

@GetMapping("/deleteUser")

**public** String deleteUser(@RequestParam("userId") **int** userId, Model theModel) {

userDAO.deleteUser(userId);

**return** "redirect:/admin/manageUsers";

}

@GetMapping("/deleteOrder")

**public** String deleteOrder(@RequestParam("orderId") **int** ordereId, Model theModel) {

orderDAO.deleteOrder(ordereId);

**return** "redirect:/admin/manageOrders";

}

@GetMapping("/manageUsers")

**public** String manageUsers(Model theModel) {

List<User> users = userDAO.getUsers();

theModel.addAttribute("users", users);

**return** "manageUsers";

}

@GetMapping("/manageOrders")

**public** String manageOrders(Model theModel) {

List<Order> orders = orderDAO.getOrder();

**int** totalPrice = 0;

**for** (Order order : orders) {

totalPrice = totalPrice + order.getProduct().getPrice();

}

theModel.addAttribute("orders", orders);

theModel.addAttribute("totalPrice", totalPrice);

**return** "manageOrders";

}

@GetMapping("/addProdcut")

**public** String addProdcut() {

**return** "addproduct";

}

@PostMapping("/addProductProcess")

**public** String addProductProcess(HttpServletRequest request,

@RequestParam("name") String name,

@RequestParam("company") String company,

@RequestParam("size") **int** size,

@RequestParam("price") **int** price,

@RequestParam("image") String image

) {

Product product = **new** Product(name, company, size, price, image);

productDAO.saveProduct(product);

**return** "redirect:/admin/products";

}

@GetMapping("/updateProduct")

**public** String updateProduct(@RequestParam("productId") **int** productId , Model theModel) {

Product product = productDAO.getProduct(productId);

theModel.addAttribute("product", product);

**return** "update-product";

}

@PostMapping("/updateProductProcess")

**public** String updateProductProcess(HttpServletRequest request,

@RequestParam("productId") **int** productId,

@RequestParam("name") String name,

@RequestParam("company") String company,

@RequestParam("size") **int** size,

@RequestParam("price") **int** price,

@RequestParam("image") String image

) {

Product product = productDAO.getProduct(productId);

product.setCompany(company);

product.setSize(size);

product.setName(name);

product.setPrice(price);

product.setImage\_link(image);

productDAO.saveProduct(product);

**return** "redirect:/admin/products";

}

}

**package** com.controller;

**import** java.util.ArrayList;

**import** java.util.List;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**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.PostMapping;

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

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

**import** com.dao.OrderDAO;

**import** com.dao.ProductDAO;

**import** com.entity.Order;

**import** com.entity.Product;

**import** com.entity.User;

@Controller

@RequestMapping("/homepage")

**public** **class** HomePageController {

// need to inject

@Autowired

**private** ProductDAO productDAO;

@Autowired

**private** OrderDAO orderDAO;

@GetMapping("/products")

**public** String products(Model theModel, HttpServletRequest request) {

HttpSession session = request.getSession(**true**);

User currentUser = (User) session.getAttribute("currentUser");

List<Product> products = productDAO.getProducts();

List<Order> userOrders = orderDAO.getUserOrders(currentUser.getId());

List<Integer> userProducts = **new** ArrayList<Integer>();

**for** (Order order : userOrders) {

userProducts.add(order.getProduct().getId());

}

theModel.addAttribute("products", products);

theModel.addAttribute("userProducts", userProducts);

theModel.addAttribute("currentUser", currentUser);

**return** "user-home";

}

@GetMapping("/orderProcess")

**public** String orderProcess(Model theModel,@RequestParam("productId") **int** productId, HttpServletRequest request) {

HttpSession session = request.getSession(**true**);

User currentUser = (User) session.getAttribute("currentUser");

Product product = productDAO.getProduct(productId);

Order order = **new** Order(currentUser, product);

orderDAO.saveOrder(order);

theModel.addAttribute("currentUser", currentUser);

**return** "redirect:/homepage/products";

}

@GetMapping("/mycart")

**public** String showMyCart(Model theModel, HttpServletRequest request) {

HttpSession session = request.getSession(**true**);

User currentUser = (User) session.getAttribute("currentUser");

List<Order> userOrders = orderDAO.getUserOrders(currentUser.getId());

List<Product> userProducts = **new** ArrayList<Product>();

**for** (Order order : userOrders) {

userProducts.add(order.getProduct());

}

**int** total\_price = 0;

**for** (Product product : userProducts) {

total\_price = total\_price + product.getPrice();

}

theModel.addAttribute("userProducts", userProducts);

theModel.addAttribute("total\_price", total\_price);

theModel.addAttribute("currentUser", currentUser);

**return** "mycart";

}

@PostMapping("/searchProducts")

**public** String searchProducts(HttpServletRequest request, Model theModel, @RequestParam("keySearch") String key) {

HttpSession session = request.getSession(**true**);

User currentUser = (User) session.getAttribute("currentUser");

System.***out***.println(key);

List<Product> products = productDAO.searchProducts(key);

**if** (currentUser.getType() == 0) {

List<Order> userOrders = orderDAO.getUserOrders(currentUser.getId());

List<Integer> userProducts = **new** ArrayList<Integer>();

**for** (Order order : userOrders) {

userProducts.add(order.getProduct().getId());

}

theModel.addAttribute("userProducts", userProducts);

theModel.addAttribute("currentUser", currentUser);

}

theModel.addAttribute("products", products);

**if** (currentUser.getType() == 0) {

**return** "user-home";

}

**else** {

**return** "mange-products";

}

}

}

**package** com.controller;

**import** java.util.List;

**import** javax.servlet.http.HttpServletRequest;

**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.ModelAttribute;

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

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

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

**import** com.dao.ProductDAO;

**import** com.dao.UserDAO;

**import** com.entity.Product;

**import** com.entity.User;

@Controller

**public** **class** LoginController {

// inject

@Autowired

**private** UserDAO userDAO;

@RequestMapping("/login")

**public** String login(Model theModel) {

**return** "login";

}

@RequestMapping("/register")

**public** String register(Model theModel) {

**return** "register";

}

@PostMapping("/loginProcess")

**public** String loginProcess(HttpServletRequest request, @RequestParam("username") String username,

@RequestParam("password") String password) {

HttpSession session = request.getSession(**true**);

List<User> users = userDAO.getUsers();

**boolean** isusesr = **false**;

User tempUser = **null**;

**for** (User user : users) {

**if** (user.getUsername().equals(username) && user.getPassword().equals(password)) {

isusesr = **true**;

tempUser = user;

**break**;

}

}

**if** (isusesr && tempUser.getType() == 0) {

session.setAttribute("currentUser", tempUser);

**return** "redirect:/homepage/products";

}

**else** **if** (isusesr && tempUser.getType() == 1) {

session.setAttribute("currentUser", tempUser);

**return** "redirect:/admin/products";

}

**else**

**return** "login";

}

@PostMapping("/registerProcess")

**public** String registerProcess(@RequestParam("username") String username,

@RequestParam("age") String age,

@RequestParam("password") String password) {

User user = **new** User(username, password, Integer.*parseInt*(age));

userDAO.saveUser(user);

**return** "login";

}

}

**package** com.controller;

**import** javax.servlet.http.HttpServletRequest;

**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.PostMapping;

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

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

**import** com.dao.ProductDAO;

**import** com.dao.UserDAO;

**import** com.entity.Product;

**import** com.entity.User;

@Controller

@RequestMapping("/user")

**public** **class** UserController {

@Autowired

**private** ProductDAO productDAO;

// inject

@Autowired

**private** UserDAO userDAO;

@GetMapping("/myaccount")

**public** String myaccount(Model theModel, HttpServletRequest request) {

HttpSession session = request.getSession(**true**);

User currentUser = (User) session.getAttribute("currentUser");

theModel.addAttribute("currentUser", currentUser);

**return** "myaccount";

}

@PostMapping("/updateAcount")

**public** String updateAcount(HttpServletRequest request,

@RequestParam("username") String username,

@RequestParam("age") String age,

@RequestParam("password") String password , Model theModel

) {

HttpSession session = request.getSession(**true**);

User currentUser = (User) session.getAttribute("currentUser");

System.***out***.println(currentUser.toString());

currentUser.setUsername(username);

currentUser.setAge(Integer.*parseInt*(age));

currentUser.setPassword(password);

theModel.addAttribute("currentUser", currentUser);

userDAO.saveUser(currentUser);

**return** "myaccount";

}

}

**package** com.dao;

**import** java.util.List;

**import** com.entity.Order;

**import** com.entity.Product;

**import** com.entity.User;

**public** **interface** OrderDAO {

**public** List<Order> getOrder();

**public** Order getOrder(**int** orderId);

**public** **void** saveOrder(Order order);

**public** **void** deleteOrder(**int** orderId);

**public** List<Order> getUserOrders(**int** userId);

}

**package** com.dao;

**import** java.util.List;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.query.Query;

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

**import** org.springframework.stereotype.Repository;

**import** org.springframework.transaction.annotation.Transactional;

**import** com.entity.Order;

**import** com.entity.User;

@Repository

**public** **class** OrderDAOImpl **implements** OrderDAO {

// need to inject the session factory

@Autowired

**private** SessionFactory sessionFactory;

@Override

@Transactional

**public** List<Order> getOrder() {

Session currentSession = sessionFactory.getCurrentSession();

Query<Order> theQuery = currentSession.createQuery("from Order", Order.**class**);

List<Order> orders = theQuery.getResultList();

**return** orders;

}

@Override

@Transactional

**public** **void** saveOrder(Order order) {

Session currentSession = sessionFactory.getCurrentSession();

currentSession.saveOrUpdate(order);

}

@Override

@Transactional

**public** Order getOrder(**int** orderId) {

Session currentSession = sessionFactory.getCurrentSession();

Order order = currentSession.get(Order.**class**, orderId);

**return** order;

}

@Override

@Transactional

**public** **void** deleteOrder(**int** orderId) {

Session currentSession = sessionFactory.getCurrentSession();

Query theQuery = currentSession.createQuery("delete from Order where id=:orderId");

theQuery.setParameter("orderId", orderId);

theQuery.executeUpdate();

}

@Override

@Transactional

**public** List<Order> getUserOrders(**int** userId) {

Session currentSession = sessionFactory.getCurrentSession();

Query<Order> theQuery = currentSession.createQuery("from Order where user\_id =: userId", Order.**class**);

theQuery.setParameter("userId", userId);

List<Order> orders = theQuery.getResultList();

**return** orders;

}

}

**package** com.dao;

**import** java.util.List;

**import** com.entity.Order;

**import** com.entity.Product;

**public** **interface** ProductDAO {

**public** List<Product> getProducts();

**public** Product getProduct(**int** productId);

**public** **void** saveProduct(Product product);

**public** **void** deleteProduct(**int** productId);

**public** List<Product> searchProducts(String key);

}

**package** com.dao;

**import** java.util.List;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.query.Query;

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

**import** org.springframework.stereotype.Repository;

**import** org.springframework.transaction.annotation.Transactional;

**import** com.entity.Order;

**import** com.entity.Product;

@Repository

**public** **class** ProductDAOImpl **implements** ProductDAO {

@Autowired

**private** SessionFactory sessionFactory;

@Override

@Transactional

**public** List<Product> getProducts() {

Session currentSession = sessionFactory.getCurrentSession();

Query<Product> theQuery = currentSession.createQuery("from Product", Product.**class**);

List<Product> products = theQuery.getResultList();

**return** products;

}

@Override

@Transactional

**public** Product getProduct(**int** productId) {

Session currentSession = sessionFactory.getCurrentSession();

Product prod = currentSession.get(Product.**class**, productId);

**return** prod;

}

@Override

@Transactional

**public** **void** saveProduct(Product product) {

Session currentSession = sessionFactory.getCurrentSession();

currentSession.saveOrUpdate(product);

}

@Override

@Transactional

**public** **void** deleteProduct(**int** productId) {

Session currentSession = sessionFactory.getCurrentSession();

Query theQuery = currentSession.createQuery("delete from Product where id=:productId");

theQuery.setParameter("productId", productId);

theQuery.executeUpdate();

}

@Override

@Transactional

**public** List<Product> searchProducts(String key) {

Session currentSession = sessionFactory.getCurrentSession();

Query<Product> theQuery = currentSession.createQuery("from Product where name like: key", Product.**class**);

theQuery.setParameter("key", "%" + key + "%");

List<Product> products = theQuery.getResultList();

System.***out***.println();

**return** products;

}

}

**package** com.dao;

**import** java.util.List;

**import** com.entity.User;

**public** **interface** UserDAO {

**public** List<User> getUsers();

**public** User getUser(**int** user);

**public** **void** saveUser(User user);

**public** **void** deleteUser(**int** user);

}

**package** com.dao;

**import** java.util.List;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.query.Query;

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

**import** org.springframework.stereotype.Repository;

**import** org.springframework.transaction.annotation.Transactional;

**import** com.entity.User;

@Repository

**public** **class** UserDAOImpl **implements** UserDAO {

@Autowired

**private** SessionFactory sessionFactory;

@Override

@Transactional

**public** List<User> getUsers() {

Session currentSession = sessionFactory.getCurrentSession();

Query<User> theQuery = currentSession.createQuery("from User", User.**class**);

List<User> users = theQuery.getResultList();

**return** users;

}

@Override

@Transactional

**public** User getUser(**int** userId) {

Session currentSession = sessionFactory.getCurrentSession();

User user = currentSession.get(User.**class**, userId);

**return** user;

}

@Override

@Transactional

**public** **void** saveUser(User user) {

Session currentSession = sessionFactory.getCurrentSession();

currentSession.saveOrUpdate(user);

}

@Override

@Transactional

**public** **void** deleteUser(**int** userId) {

Session currentSession = sessionFactory.getCurrentSession();

Query theQuery = currentSession.createQuery("delete from User where id=:userId");

theQuery.setParameter("userId", userId);

theQuery.executeUpdate();

}

}

**package** com.entity;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.FetchType;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** javax.persistence.JoinColumn;

**import** javax.persistence.ManyToOne;

**import** javax.persistence.Table;

@Entity

@Table(name="orders")

**public** **class** Order {

@Id

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

@Column(name="id")

**private** **int** id;

@ManyToOne

@JoinColumn(name="user\_id")

**private** User user;

@ManyToOne

@JoinColumn(name="product\_id")

**private** Product product;

**public** Order() {

}

**public** Order(User user, Product product) {

**super**();

**this**.user = user;

**this**.product = product;

}

**public** **int** getId() {

**return** id;

}

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

**this**.id = id;

}

**public** User getUser() {

**return** user;

}

**public** **void** setUser(User user) {

**this**.user = user;

}

**public** Product getProduct() {

**return** product;

}

**public** **void** setProduct(Product product) {

**this**.product = product;

}

@Override

**public** String toString() {

**return** "Order [id=" + id + ", user=" + user + ", product=" + product + "]";

}

}

**package** com.entity;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** javax.persistence.Table;

@Entity

@Table(name="products")

**public** **class** Product {

@Id

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

@Column(name="id")

**private** **int** id;

@Column(name="name")

**private** String name;

@Column(name="company")

**private** String company;

@Column(name="size")

**private** **int** size;

@Column(name="price")

**private** **int** price;

@Column(name="image\_link")

**private** String image\_link;

**public** Product() {

}

**public** Product(String name, String company, **int** size, **int** price, String image\_link) {

**super**();

**this**.name = name;

**this**.company = company;

**this**.size = size;

**this**.price = price;

**this**.image\_link = image\_link;

}

**public** **int** getId() {

**return** id;

}

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

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getCompany() {

**return** company;

}

**public** **void** setCompany(String company) {

**this**.company = company;

}

**public** **int** getSize() {

**return** size;

}

**public** **void** setSize(**int** size) {

**this**.size = size;

}

**public** **int** getPrice() {

**return** price;

}

**public** **void** setPrice(**int** price) {

**this**.price = price;

}

**public** String getImage\_link() {

**return** image\_link;

}

**public** **void** setImage\_link(String image\_link) {

**this**.image\_link = image\_link;

}

@Override

**public** String toString() {

**return** "Product [id=" + id + ", name=" + name + ", company=" + company + ", size=" + size + ", price=" + price

+ ", image\_link=" + image\_link + "]";

}

}

**package** com.entity;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** javax.persistence.Table;

**import** javax.validation.constraints.Max;

**import** javax.validation.constraints.NotNull;

**import** javax.validation.constraints.Pattern;

**import** org.hibernate.validator.constraints.Range;

@Entity

@Table(name="users")

**public** **class** User {

@Id

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

@Column(name="id")

**private** **int** id;

@Column(name="type")

**private** **int** type;

@Column(name="username")

**private** String username;

@Column(name="password")

**private** String password;

@Range(min=1, max=120,message = "Invalied Age")

@Column(name="age")

**private** **int** age;

**public** User() {

}

**public** User(String username, String password, **int** age) {

**super**();

**this**.username = username;

**this**.password = password;

**this**.age = age;

}

**public** **int** getId() {

**return** id;

}

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

**this**.id = id;

}

**public** **int** getType() {

**return** type;

}

**public** **void** setType(**int** type) {

**this**.type = type;

}

**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;

}

**public** **int** getAge() {

**return** age;

}

**public** **void** setAge(**int** age) {

**this**.age = age;

}

@Override

**public** String toString() {

**return** "User [id=" + id + ", type=" + type + ", username=" + username + ", password=" + password + ", age="

+ age + "]";

}

}

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

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:tx=*"http://www.springframework.org/schema/tx"*

xmlns:mvc=*"http://www.springframework.org/schema/mvc"*

xsi:schemaLocation=*"*

*http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context.xsd*

*http://www.springframework.org/schema/mvc*

*http://www.springframework.org/schema/mvc/spring-mvc.xsd*

*http://www.springframework.org/schema/tx*

*http://www.springframework.org/schema/tx/spring-tx.xsd"*>

<!-- Add support for component scanning -->

<context:component-scan base-package=*"com"* />

<!-- Add support for conversion, formatting and validation support -->

<mvc:annotation-driven/>

<!-- Define Spring MVC view resolver -->

<bean

class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>

<property name=*"prefix"* value=*"/WEB-INF/view/"* />

<property name=*"suffix"* value=*".jsp"* />

</bean>

<!-- Step 1: Define Database DataSource / connection pool -->

<bean id=*"myDataSource"* class=*"com.mchange.v2.c3p0.ComboPooledDataSource"*

destroy-method=*"close"*>

<property name=*"driverClass"* value=*"com.mysql.jdbc.Driver"* />

<property name=*"jdbcUrl"* value=*"jdbc:mysql://localhost:3306/sportyshoes"* />

<property name=*"user"* value=*"root"* />

<property name=*"password"* value=*"Prembattineni@123"* />

<!-- these are connection pool properties for C3P0 -->

<property name=*"minPoolSize"* value=*"5"* />

<property name=*"maxPoolSize"* value=*"20"* />

<property name=*"maxIdleTime"* value=*"30000"* />

</bean>

<!-- Step 2: Setup Hibernate session factory -->

<bean id=*"sessionFactory"*

class=*"org.springframework.orm.hibernate5.LocalSessionFactoryBean"*>

<property name=*"dataSource"* ref=*"myDataSource"* />

<property name=*"packagesToScan"* value=*"com.entity"* />

<property name=*"hibernateProperties"*>

<props>

<prop key=*"hibernate.dialect"*>org.hibernate.dialect.MySQL5Dialect</prop>

<prop key=*"hibernate.show\_sql"*>true</prop>

</props>

</property>

</bean>

<!-- Step 3: Setup Hibernate transaction manager -->

<bean id=*"myTransactionManager"*

class=*"org.springframework.orm.hibernate5.HibernateTransactionManager"*>

<property name=*"sessionFactory"* ref=*"sessionFactory"*/>

</bean>

<!-- Step 4: Enable configuration of transactional behavior based on annotations -->

<tx:annotation-driven transaction-manager=*"myTransactionManager"* />

<!-- Add support for reading web resources: css, images, js, etc ... -->

<mvc:resources location=*"/resources/"* mapping=*"/resources/\*\*"*></mvc:resources>

</beans>

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

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"* xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd"* id=*"WebApp\_ID"* version=*"3.1"*>

<display-name>Sporty-shoes-store</display-name>

<absolute-ordering/>

<welcome-file-list>

<welcome-file>index.jsp</welcome-file>

<welcome-file>index.html</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>dispatcher</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<init-param>

<param-name>contextConfigLocation</param-name>

<param-value>/WEB-INF/sporty-shoes-servlet.xml</param-value>

</init-param>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>dispatcher</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>

</web-app>

<% response.sendRedirect("login"); %>

<?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 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.simplilearn</groupId>

<artifactId>Sporty-Shoes-Store</artifactId>

<version>1.0</version>

<packaging>war</packaging>

<name>Sporty-Shoes-Store Maven Webapp</name>

<!-- FIXME change it to the project's website -->

<url>http://www.example.com</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<maven.compiler.source>1.8</maven.compiler.source>

<maven.compiler.target>1.8</maven.compiler.target>

</properties>

<dependencies>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-webmvc -->

<!-- https://mvnrepository.com/artifact/org.springframework/spring-web -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>5.2.15.RELEASE</version>

</dependency>

<!-- https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>javax.servlet-api</artifactId>

<version>4.0.1</version>

<scope>provided</scope>

</dependency>

<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>5.5.2.Final</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-tx -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-tx</artifactId>

<version>5.3.3</version>

</dependency>

<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->

<dependency>

<groupId>mysql</groupId>

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

<version>8.0.12</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-validator -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-validator</artifactId>

<version>6.1.0.Final</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-core -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>5.3.4</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>5.2.15.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>5.1.6.RELEASE</version>

</dependency>

<!-- https://mvnrepository.com/artifact/com.mchange/c3p0 -->

<dependency>

<groupId>com.mchange</groupId>

<artifactId>c3p0</artifactId>

<version>0.9.5.5</version>

</dependency>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.11</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<finalName>Sporty-Shoes-Store</finalName>

<pluginManagement><!-- lock down plugins versions to avoid using Maven

defaults (may be moved to parent pom) -->

<plugins>

<plugin>

<artifactId>maven-clean-plugin</artifactId>

<version>3.1.0</version>

</plugin>

<!-- see http://maven.apache.org/ref/current/maven-core/default-bindings.html#Plugin\_bindings\_for\_war\_packaging -->

<plugin>

<artifactId>maven-resources-plugin</artifactId>

<version>3.0.2</version>

</plugin>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.8.0</version>

</plugin>

<plugin>

<artifactId>maven-surefire-plugin</artifactId>

<version>2.22.1</version>

</plugin>

<plugin>

<artifactId>maven-war-plugin</artifactId>

<version>3.2.2</version>

</plugin>

<plugin>

<artifactId>maven-install-plugin</artifactId>

<version>2.5.2</version>

</plugin>

<plugin>

<artifactId>maven-deploy-plugin</artifactId>

<version>2.8.2</version>

</plugin>

</plugins>

</pluginManagement>

</build>

</project>