1) package com.simplilearn.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.simplilearn.dao.OrderDAO;

import com.simplilearn.dao.ProductDAO;

import com.simplilearn.dao.UserDAO;

import com.simplilearn.entity.Order;

import com.simplilearn.entity.Product;

import com.simplilearn.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");

// get all products from DAO

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

// add the products to the model

theModel.addAttribute("currentUser", currentUser);

// add the products to the model

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) {

// get all users from DAO

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

// add the users to the model

theModel.addAttribute("users", users);

return "manageUsers";

}

@GetMapping("/manageOrders")

public String manageOrders(Model theModel) {

// get all users from DAO

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

int totalPrice = 0;

for (Order order : orders) {

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

}

// add the users to the model

theModel.addAttribute("orders", orders);

// add the users to the model

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) {

// get all products from DAO

Product product = productDAO.getProduct(productId);

// add the users to the model

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.simplilearn.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.simplilearn.dao.OrderDAO;

import com.simplilearn.dao.ProductDAO;

import com.simplilearn.entity.Order;

import com.simplilearn.entity.Product;

import com.simplilearn.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");

// get all products from DAO

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());

}

// add the products to the model

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");

// get all products from DAO

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);

// get all products from DAO

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

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

// get all products from DAO

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);

}

// add the products to the model

theModel.addAttribute("products", products);

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

return "user-home";

}

else {

return "mange-products";

}

}

}

package com.simplilearn.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.simplilearn.dao.ProductDAO;

import com.simplilearn.dao.UserDAO;

import com.simplilearn.entity.Product;

import com.simplilearn.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);

// get all users from DAO

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.simplilearn.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.simplilearn.dao.ProductDAO;

import com.simplilearn.dao.UserDAO;

import com.simplilearn.entity.Product;

import com.simplilearn.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");

// add the products to the model

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.simplilearn.dao;

import java.util.List;

import com.simplilearn.entity.Order;

import com.simplilearn.entity.Product;

import com.simplilearn.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.simplilearn.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.simplilearn.entity.Order;

import com.simplilearn.entity.User;

@Repository

public class OrderDAOImpl implements OrderDAO {

// need to inject the session factory

@Autowired

private SessionFactory sessionFactory;

@Override

@Transactional

public List<Order> getOrder() {

// get the current hibernate session

Session currentSession = sessionFactory.getCurrentSession();

// create a query ... sort by last name

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

// execute query and get result list

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

// return the results

return orders;

}

@Override

@Transactional

public void saveOrder(Order order) {

// get current hibernate session

Session currentSession = sessionFactory.getCurrentSession();

// save/update the customer

currentSession.saveOrUpdate(order);

}

@Override

@Transactional

public Order getOrder(int orderId) {

// get the current hibernate session

Session currentSession = sessionFactory.getCurrentSession();

// now retrieve/read from database using the primary key

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

return order;

}

@Override

@Transactional

public void deleteOrder(int orderId) {

// get the current hibernate session

Session currentSession = sessionFactory.getCurrentSession();

// delete object with primary key

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

theQuery.setParameter("orderId", orderId);

theQuery.executeUpdate();

}

@Override

@Transactional

public List<Order> getUserOrders(int userId) {

// get the current hibernate session

Session currentSession = sessionFactory.getCurrentSession();

// create a query ... sort by last name

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

theQuery.setParameter("userId", userId);

// execute query and get result list

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

// return the results

return orders;

}

}

package com.simplilearn.dao;

import java.util.List;

import com.simplilearn.entity.Order;

import com.simplilearn.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.simplilearn.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.simplilearn.entity.Order;

import com.simplilearn.entity.Product;

@Repository

public class ProductDAOImpl implements ProductDAO {

// need to inject the session factory

@Autowired

private SessionFactory sessionFactory;

@Override

@Transactional

public List<Product> getProducts() {

// get the current hibernate session

Session currentSession = sessionFactory.getCurrentSession();

// create a query

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

// execute query and get result list

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

// return the results

return products;

}

@Override

@Transactional

public Product getProduct(int productId) {

// get the current hibernate session

Session currentSession = sessionFactory.getCurrentSession();

// now retrieve/read from database using the primary key

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

return prod;

}

@Override

@Transactional

public void saveProduct(Product product) {

// get current hibernate session

Session currentSession = sessionFactory.getCurrentSession();

// save/update

currentSession.saveOrUpdate(product);

}

@Override

@Transactional

public void deleteProduct(int productId) {

// get the current hibernate session

Session currentSession = sessionFactory.getCurrentSession();

// delete object with primary key

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

theQuery.setParameter("productId", productId);

theQuery.executeUpdate();

}

@Override

@Transactional

public List<Product> searchProducts(String key) {

// get the current hibernate session

Session currentSession = sessionFactory.getCurrentSession();

// create a query

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

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

// execute query and get result list

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

System.out.println();

// return the results

return products;

}

}

package com.simplilearn.dao;

import java.util.List;

import com.simplilearn.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.simplilearn.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.simplilearn.entity.User;

@Repository

public class UserDAOImpl implements UserDAO {

// need to inject the session factory

@Autowired

private SessionFactory sessionFactory;

@Override

@Transactional

public List<User> getUsers() {

// get the current hibernate session

Session currentSession = sessionFactory.getCurrentSession();

// create a query ... sort by last name

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

// execute query and get result list

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

// return the results

return users;

}

@Override

@Transactional

public User getUser(int userId) {

// get the current hibernate session

Session currentSession = sessionFactory.getCurrentSession();

// now retrieve/read from database using the primary key

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

return user;

}

@Override

@Transactional

public void saveUser(User user) {

// get current hibernate session

Session currentSession = sessionFactory.getCurrentSession();

// save/update the customer

currentSession.saveOrUpdate(user);

}

@Override

@Transactional

public void deleteUser(int userId) {

// get the current hibernate session

Session currentSession = sessionFactory.getCurrentSession();

// delete object with primary key

Query theQuery =

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

theQuery.setParameter("userId", userId);

theQuery.executeUpdate();

}

}

package com.simplilearn.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.simplilearn.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.simplilearn.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 + "]";

}

}