Cart addition

================

1.Principle usage for getting user axccess

---------------------------------------------------------

a.In the User class , add @Component beneath @Entity.

b.Open IUserDAO class , and add:

public User getUserByUserName(String userName);

c.Open the UserDAOImpl class , add definition for the new method :

@Override

public User getUserByUserName(String userName) {

String selectUser="from User where userName=:userName";

**return** sessionFactory.getCurrentSession().createQuery(selectUser,User.**class**).setParameter("userName",userName).getSingleResult();

}

2.Cart functionality

---------------------------

a.In the ICartDAO update:

package net.sports.ZenSportsBackEnd.dao;

import java.util.List;

import net.sports.ZenSportsBackEnd.model.Cart;

import net.sports.ZenSportsBackEnd.model.CartItem;

public interface ICartItemDAO

{

public List<CartItem> getAllCartItem(Cart item);

public CartItem getCartItem(int id);

//CRUD for cart items

public boolean addCartItem(CartItem ci);

public boolean updateCartItem(CartItem ci);

public boolean deleteCartItem(int cartItemId);

public boolean deleteAllCartItem(Cart c);

//CRUD for cart

public boolean updateCart(Cart c);

public boolean deleteCart(int cid);

}

b.In the CartItemDAOImpl update:

package net.sports.ZenSportsBackEnd.daoimpl;

import java.util.List;

import org.hibernate.SessionFactory;

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

import org.springframework.stereotype.Repository;

import org.springframework.transaction.annotation.Transactional;

import net.sports.ZenSportsBackEnd.dao.ICartItemDAO;

import net.sports.ZenSportsBackEnd.dao.IUserDAO;

import net.sports.ZenSportsBackEnd.model.Cart;

import net.sports.ZenSportsBackEnd.model.CartItem;

@Repository("cartItemDAO")

@Transactional

public class CartItemDAOImpl implements ICartItemDAO {

@Autowired

SessionFactory sessionFactory;

IUserDAO userDAO;

public List<CartItem> getAllCartItem(Cart cart) {

return sessionFactory.getCurrentSession().createQuery("from CartItem where cart=:cart").setParameter("cart",cart).getResultList(); }

public CartItem getCartItem(int id) {

return sessionFactory.getCurrentSession().get(CartItem.class, Integer.valueOf(id));

}

public boolean addCartItem(CartItem ci) {

try {

sessionFactory.getCurrentSession().save(ci);

return true;

} catch (Exception e) {

e.printStackTrace();

return false;

}

}

public boolean updateCartItem(CartItem ci) {

try {

sessionFactory.getCurrentSession().update(ci);

return true;

} catch (Exception e) {

e.printStackTrace();

return false;

}

}

public boolean deleteCartItem(int cartItemId) {

try {

sessionFactory.getCurrentSession().delete(getCartItem(cartItemId));

return true;

} catch (Exception e) {

e.printStackTrace();

return false;

}

}

@Override

public boolean deleteAllCartItem(Cart c) {

try {

sessionFactory.getCurrentSession().delete(c.getCartId());

return true;

} catch (Exception e) {

e.printStackTrace();

return false;

}

}

@Override

public boolean updateCart(Cart c) {

try {

sessionFactory.getCurrentSession().update(c);

return true;

} catch (Exception e) {

e.printStackTrace();

return false;

}

}

@Override

public boolean deleteCart(int cid) {

try {

sessionFactory.getCurrentSession().delete(cid);

return true;

} catch (Exception e) {

e.printStackTrace();

return false;

}

}

}

c.Make a FrontEndCartController , inside it , add the following :

package net.sports.ZenSportsFrontEnd.controller;

import java.security.Principal;

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

import org.springframework.stereotype.Controller;

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

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

import org.springframework.web.servlet.ModelAndView;

import net.sports.ZenSportsBackEnd.dao.ICartItemDAO;

import net.sports.ZenSportsBackEnd.dao.IProductDAO;

import net.sports.ZenSportsBackEnd.dao.IUserDAO;

import net.sports.ZenSportsBackEnd.model.Cart;

import net.sports.ZenSportsBackEnd.model.CartItem;

import net.sports.ZenSportsBackEnd.model.Product;

import net.sports.ZenSportsBackEnd.model.User;

@Controller

public class FrontEndCartController {

@Autowired

IProductDAO productDAO;

@Autowired

ICartItemDAO cartItemDAO;

@Autowired

IUserDAO userDAO;

@RequestMapping(value = { "/user/{productId}/{quantity}/cart" })

public ModelAndView addToCart(@PathVariable("productId") int id, @PathVariable("quantity") int quantity,

Principal principal) {

ModelAndView model;

if (principal != null) {

model = new ModelAndView("page");

User u = userDAO.getUserByUserName(principal.getName());

Product p = productDAO.getProduct(id);

// User u=userDAO.getUser(uid);

CartItem ci = new CartItem();

Cart c = u.getCart();

// calculating subTotal for cartItems

int unitPrice = p.getProductPrice();

int subTotal = unitPrice \* quantity;

// updating cart instance

int totalPrice = c.getGrandTotal();

totalPrice += subTotal;

c.setGrandTotal(totalPrice);

int items = c.getItemCounts();

items+=1;

c.setItemCounts(items);

// update cartitem instance

ci.setCart(c);

ci.setQuantity(quantity);

ci.setTotalPrice(new Double(subTotal));

ci.setProduct(productDAO.getProduct(id));

// updating product instance

// int i\_q = p.getProductQuantity();

// int f\_q = i\_q - quantity;

// p.setProductQuantity(f\_q);

// addding cart item

cartItemDAO.addCartItem(ci);

// updating cart

cartItemDAO.updateCart(c);

// updating product

// productDAO.updateProduct(p);

model.addObject("title", "Products");

model.addObject("userClickProducts", true);

model.addObject("product", new Product());

model.addObject("productList", productDAO.getProducts());

}

else {

model = new ModelAndView("login");

}

return model;

}

}

d.cart.jsp-update the page

<div class=*"content"*>

<form:form modelAttribute=*"prod"*>

<div class=*"row"*>

<div class=*"col-md-12"* style="height:*5%*: *border-bottom*:*1px solid black*">

<h2>Shopping Cart <span>(<!-- ${cart.itemCount} --> Item(s))</span></h2>

</div>

</div>

<div class=*"row"*>

<div class=*"col-md-12"*>

<table class=*"table table-responsive table-sripped"*>

<tr>

<th colspan=*"2"* style="color:*darkgrey*">Item</th>

<th style="color:*darkgrey*">Quantity</th>

<th style="color:*darkgrey*">Price</th>

<th style="color:*darkgrey*">Sub Total</th>

<th></th>

</tr>

<c:set var=*"price"* value="${0}"/>

<c:forEach items="${prod}" var=*"p"*>

<tr>

<td><img src=*"*${images}*/*${p.product.productImage}*"* width=*"70px"* height=*"70px"*/></td>

<td>${p.product.productName}</td>

<td>

${p.quantity}

</td>

<td>&#8377; ${p.totalPrice} </td>

<c:set var=*"price"* value="${price+p.totalPrice }"/>

<td></td>

<td><span class=*"glyphicon glyphicon-remove-sign"* style="text-size:*30px*; color:*crimson*"></span></div></td>

</tr>

</c:forEach>

</table>

</div>

</div>

<div class=*"row"*>

<div class=*"col-md-12"*>

<table class=*"table table-responsive"*>

<tr>

<td></td>

<td></td>

<td>Grand Total</td>

<td>${price}</td>

</tr>

<tr>

<td></td>

<td></td>

<td><a href=*"*${contextRoot}*/user/products"* class=*"btn btn-danger btn-md"*>Continue Shopping</button> </td>

<td><a href=*"*${contextRoot}*/payment"* class=*"btn btn-danger btn-lg"*>Place Order</button></td>

</tr>

</table>

</div>

</div>

</form:form>

</div>

</div>

3.Add CartItemTestCase , and add the following code:

**package** net.sports.ZenSportsBackEnd.test;

**import** org.junit.BeforeClass;

**import** org.junit.Test;

**import** **static** org.junit.Assert.\*;

**import** org.springframework.context.annotation.AnnotationConfigApplicationContext;

**import** net.sports.ZenSportsBackEnd.dao.ICartItemDAO;

**import** net.sports.ZenSportsBackEnd.dao.IProductDAO;

**import** net.sports.ZenSportsBackEnd.dao.IUserDAO;

**import** net.sports.ZenSportsBackEnd.model.Cart;

**import** net.sports.ZenSportsBackEnd.model.CartItem;

**import** net.sports.ZenSportsBackEnd.model.Product;

**import** net.sports.ZenSportsBackEnd.model.User;

**public** **class** CartItemTestCase

{

**private** **static** AnnotationConfigApplicationContext *context*;

**private** **static** IProductDAO *productDAO*;

**private** **static** IUserDAO *userDAO*;

**private** **static** ICartItemDAO *cartItemDAO*;

@BeforeClass

**public** **static** **void** init() {

*context* = **new** AnnotationConfigApplicationContext();

*context*.scan("net.sports.ZenSportsBackEnd");

*context*.refresh();

*productDAO*=(IProductDAO)*context*.getBean("productDAO");

*userDAO*=(IUserDAO)*context*.getBean("userDAO");

*cartItemDAO*=(ICartItemDAO)*context*.getBean("cartItemDAO");

}

@Test

**public** **void** testAddCartItem()

{

Product p=*productDAO*.getProduct(4);

User u=*userDAO*.getUser(1);

CartItem ci = **new** CartItem();

Cart c = u.getCart();

// calculating subTotal for cartItems

**int** unitPrice = p.getProductPrice();

**int** subTotal = unitPrice \* 2;

// updating cart instance

**int** totalPrice = c.getGrandTotal();

totalPrice += subTotal;

c.setGrandTotal(totalPrice);

**int** items = c.getItemCounts();

items+=1;

c.setItemCounts(items);

// update cartitem instance

ci.setCart(c);

ci.setQuantity(2);

ci.setTotalPrice(**new** Double(subTotal));

ci.setProduct(*productDAO*.getProduct(1));

// updating product instance

// int i\_q = p.getProductQuantity();

// int f\_q = i\_q - 2;

// p.setProductQuantity(f\_q);

// addding cart item

*cartItemDAO*.addCartItem(ci);

// updating cart

//cartItemDAO.updateCart(c);

// updating product

//productDAO.updateProduct(p);

*assertEquals*("Successfully added a product inside the table .",**true**,*cartItemDAO*.updateCart(c));

}

}

4.Update the productDetails page

<!-- Product Details -->

<div class=*"container content"*>

<form:form modelAttribute=*"prod"* action="${contextRoot}*/user/*${prod.productId}">

<div style="height: *20%*"></div>

<div class=*"row well"*>

<div class=*"col-md-3"*>

<div id=*"*${prod.productId}*"*></div>

<div>

<img src=*"*${images}*/*${prod.productImage}*"* height=*"300"* width=*"200"*>

</div>

</div>

<div class=*"col-md-6"*>

<div class=*"row"*>

<p>

<h1 style="color: *blue*; font-weight: *bold*">${prod.productName}</h1>

</p>

</div>

<div class=*"row"*>

<p>

<h4>

<b>Category : </b>${prod.productCategory}</h2>

</p>

</div>

<div class=*"row"*>

<p>

<h4>Keywords : ${prod.productKeyword}</h4>

</p>

</div>

<div class=*"row"*>

<br />

<p>

<h4>

<b>DESCRIPTION</b><br />${prod.productDescription}</h4>

</p>

</div>

<div class=*"row"*>

<p>

<b></b>

<h4 style="color: *red*">&#8377 ${prod.productPrice}</h4>

</b>

</p>

</div>

<div class=*"row"*>

<p>

<b></b>

<h4 style="color: *orange*">${prod.productQuantity} pieces left

.</h4>

</b>

</p>

</div>

<div class=*"row"*>

<select name=*"quantity"* id=*"quantity"*>

<c:forEach var=*"i"* begin=*"1"* end="${prod.productQuantity}">

<option value=*"*${i}*"*>${i}</option>

</c:forEach>

</select>

</div>

<div class=*"row"*>

<p>

<button style="text-size: *30px*" class=*"btn btn-danger btn-md"*>

<span class=*"glyphicon glyphicon-shopping-cart"*

style="text-size: *30px*"></span> Add to Cart

</button>

</p>

</div>

</div>

</div>

</form:form>

</div>

5.Remove the last column from the datatablescript.js , first script.Also remove the last column(add to cart )from products.jsp page.

6.In the frontEndProductController , add:

@RequestMapping(value={"/user/cart"})

**public** ModelAndView cart(Principal p){

ModelAndView model =**new** ModelAndView("page");

model.addObject("userClickCart","true");

model.addObject("user","true");

User u=userDAO.getUserByUserName(p.getName());

Cart c=u.getCart();

model.addObject("prod",cartItemDAO.getAllCartItem(c));

//model.addObject("prod", productDAO.getProduct(id));

**return** model;

}

6.Add rhe following line in the dispatcher servlet after register workflow line;

<webflow:flow-location id="user/shipping" path="/cartFlow.xml" />

7.In the IAddressDAO file , add a method:

**public** Address getAddressByUser(User user);

8.In the AddressDAOImpl file , add the definition:

@Override

**public** Address getAddressByUser(User user) {

**return** (Address)sessionFactory.getCurrentSession().createQuery("from Address where user=:user").setParameter("user",user).getSingleResult();

}

9.