**Web Development tools and Methods Project**

**Domain:** Retail and Manufacturing

**Scope:**

To design Web application to handle order request for a computer appliance manufacturing company. The system should be able to serve as a common platform for admin/support team and customer to login and use the system to handle the information about the products sold by the company.

The Support Team/Admin can create Category and Products they are willing to sell.

The Support Team/Admin can update Product Availability, Products quantity and In Stock Details.

The Customer can browse through the products and Categories.

The Customer can add the products to his cart.

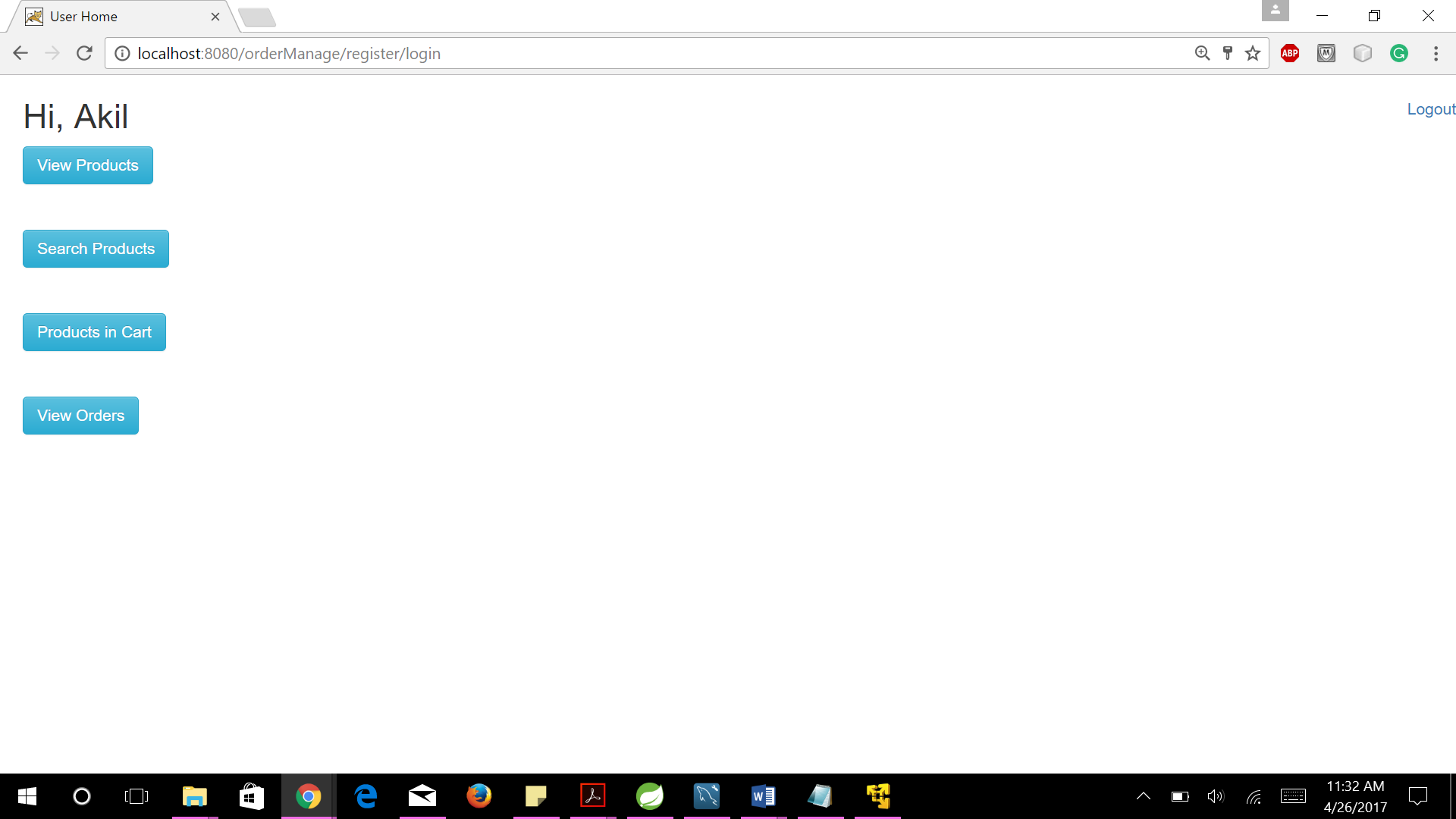
The Customer can place Order.

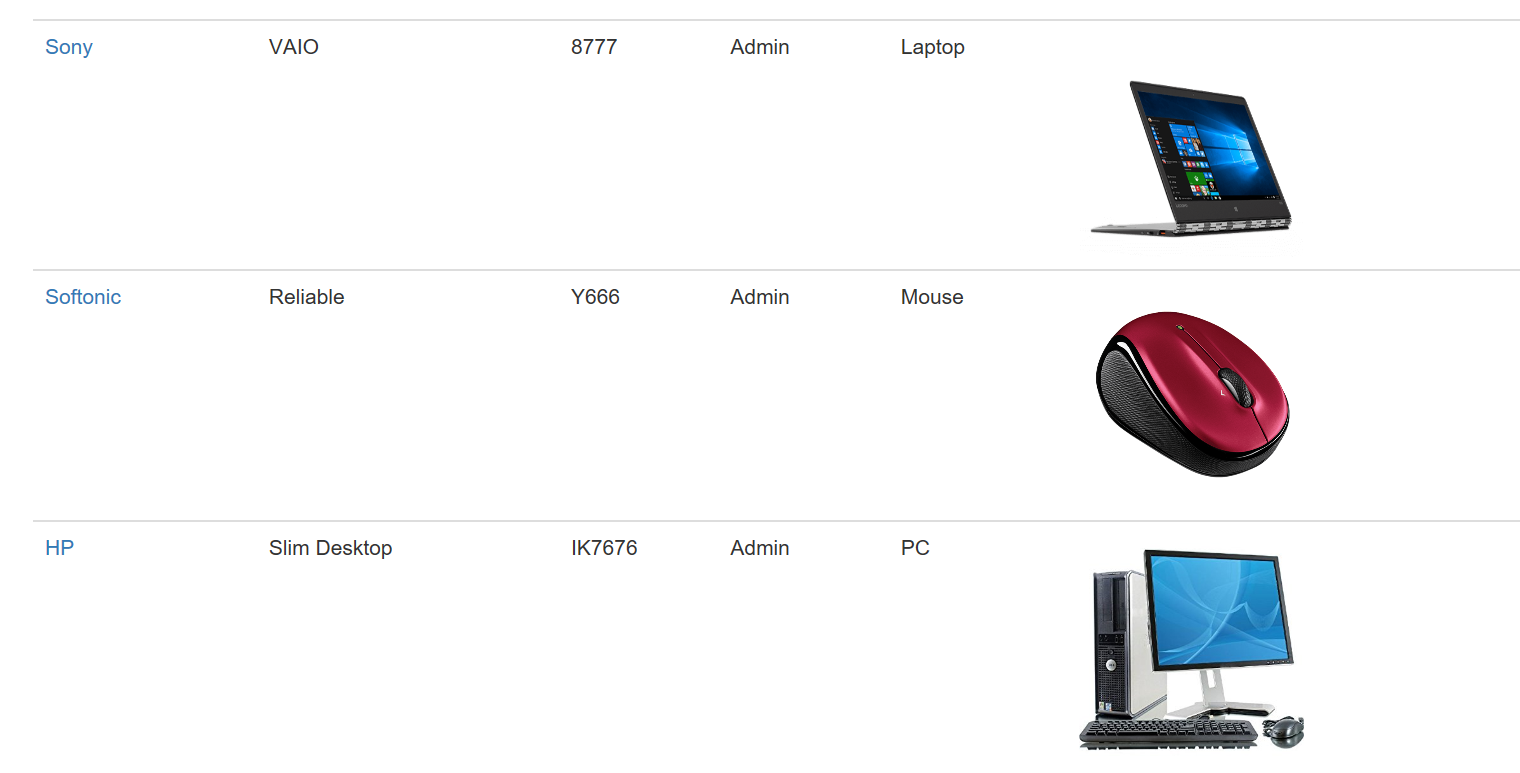
The Support team /Admin can see the orders placed by all users.

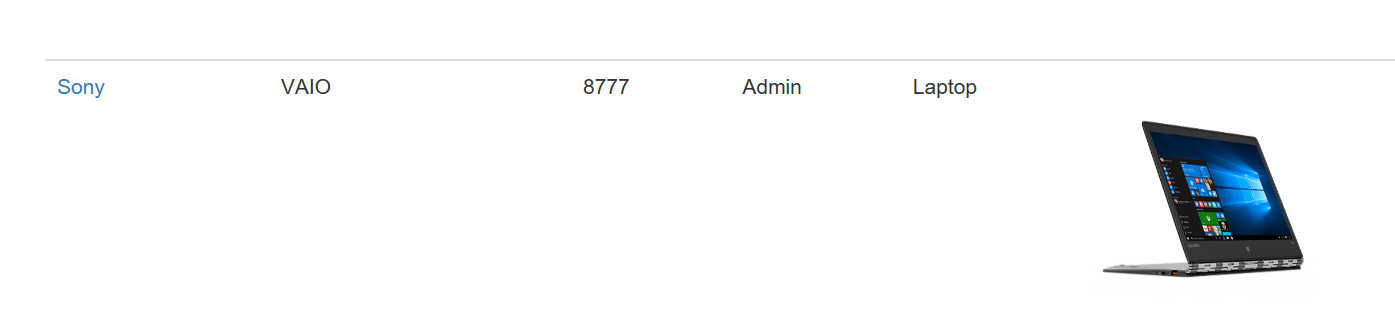
The Support team can view the In stock for each Product they have and update the quantity available for the order and amount for the product.

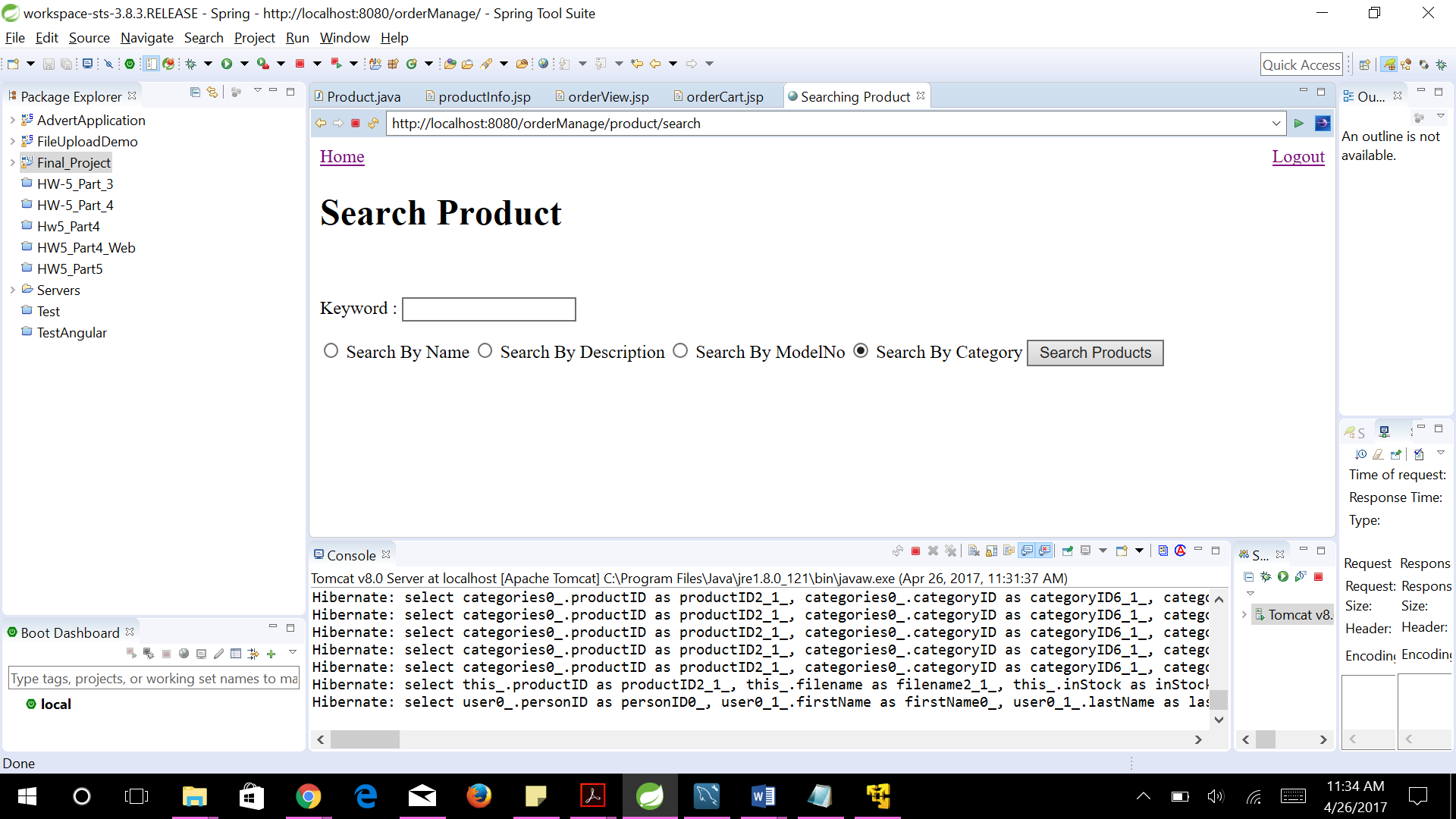
This updated information of the Order can be viewed by the Customer.

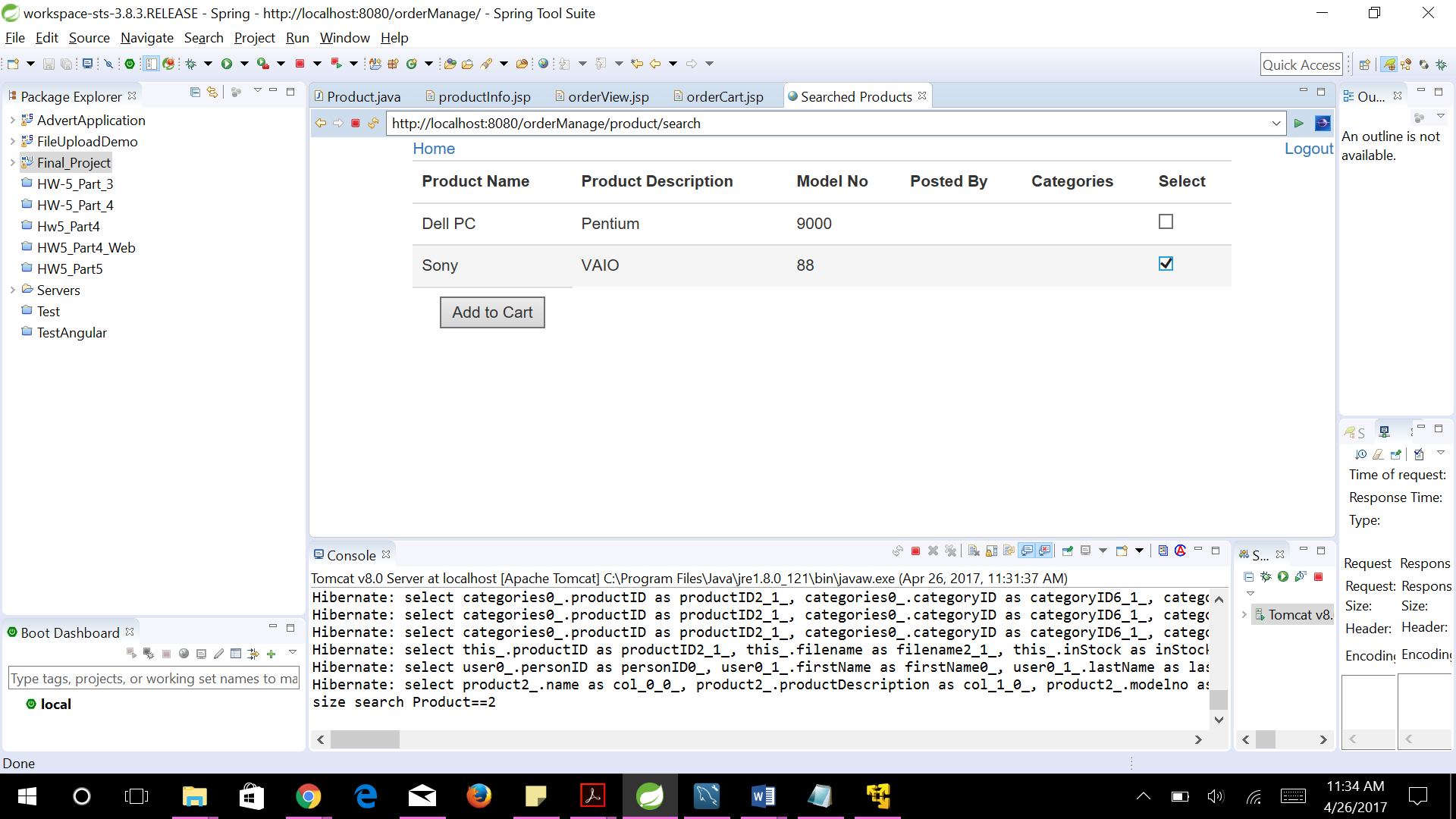
**Screen Shots**

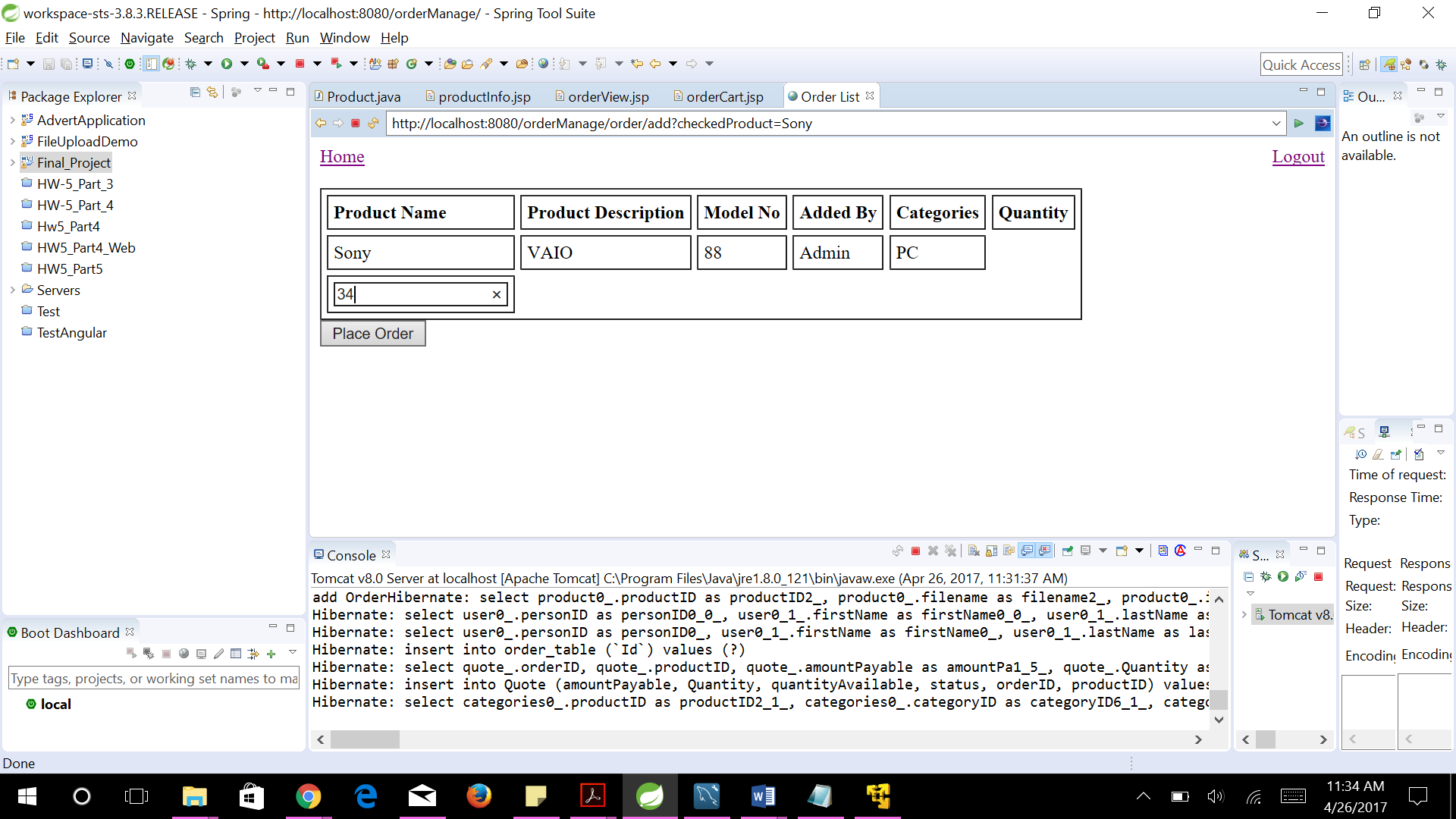


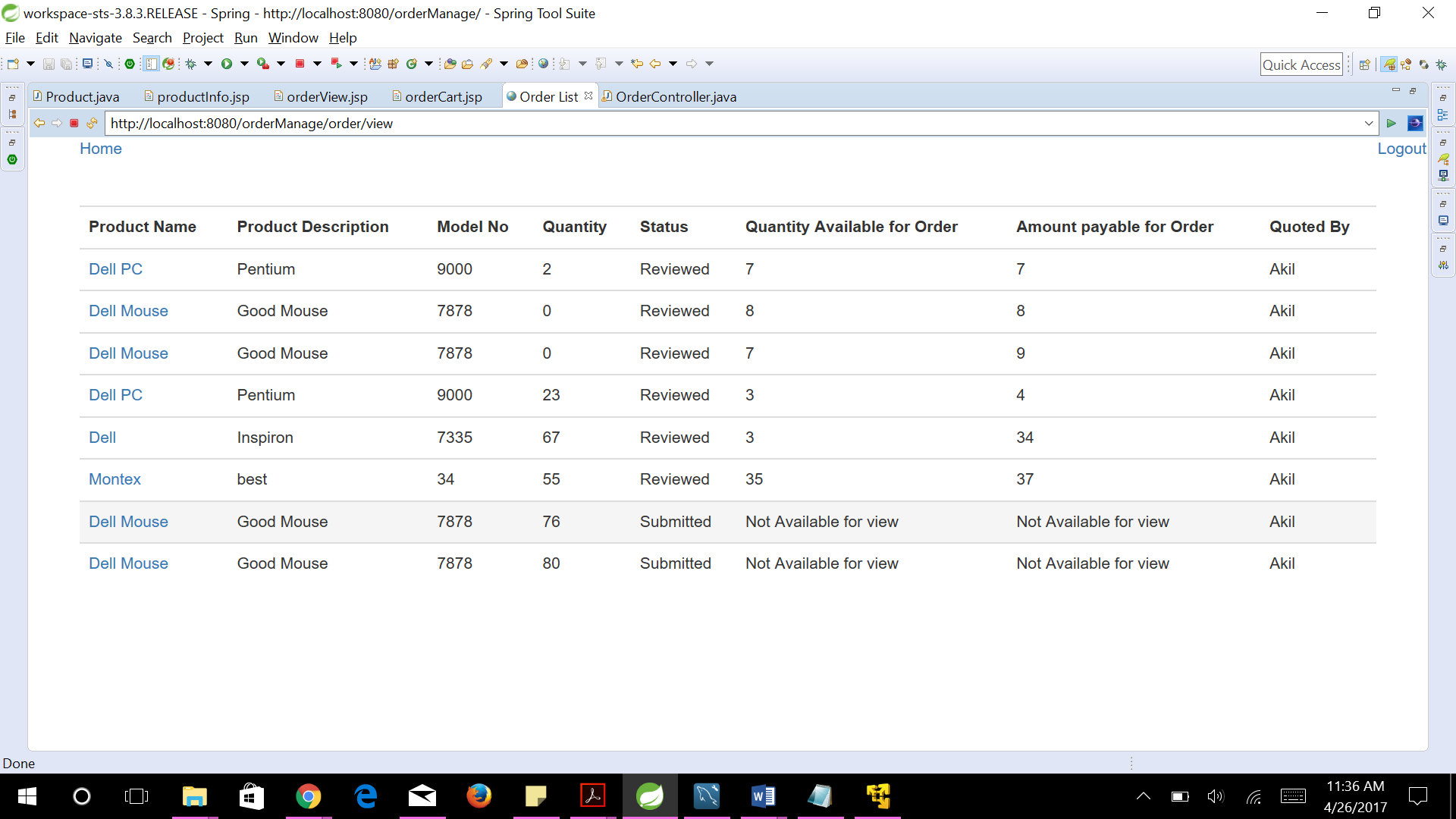




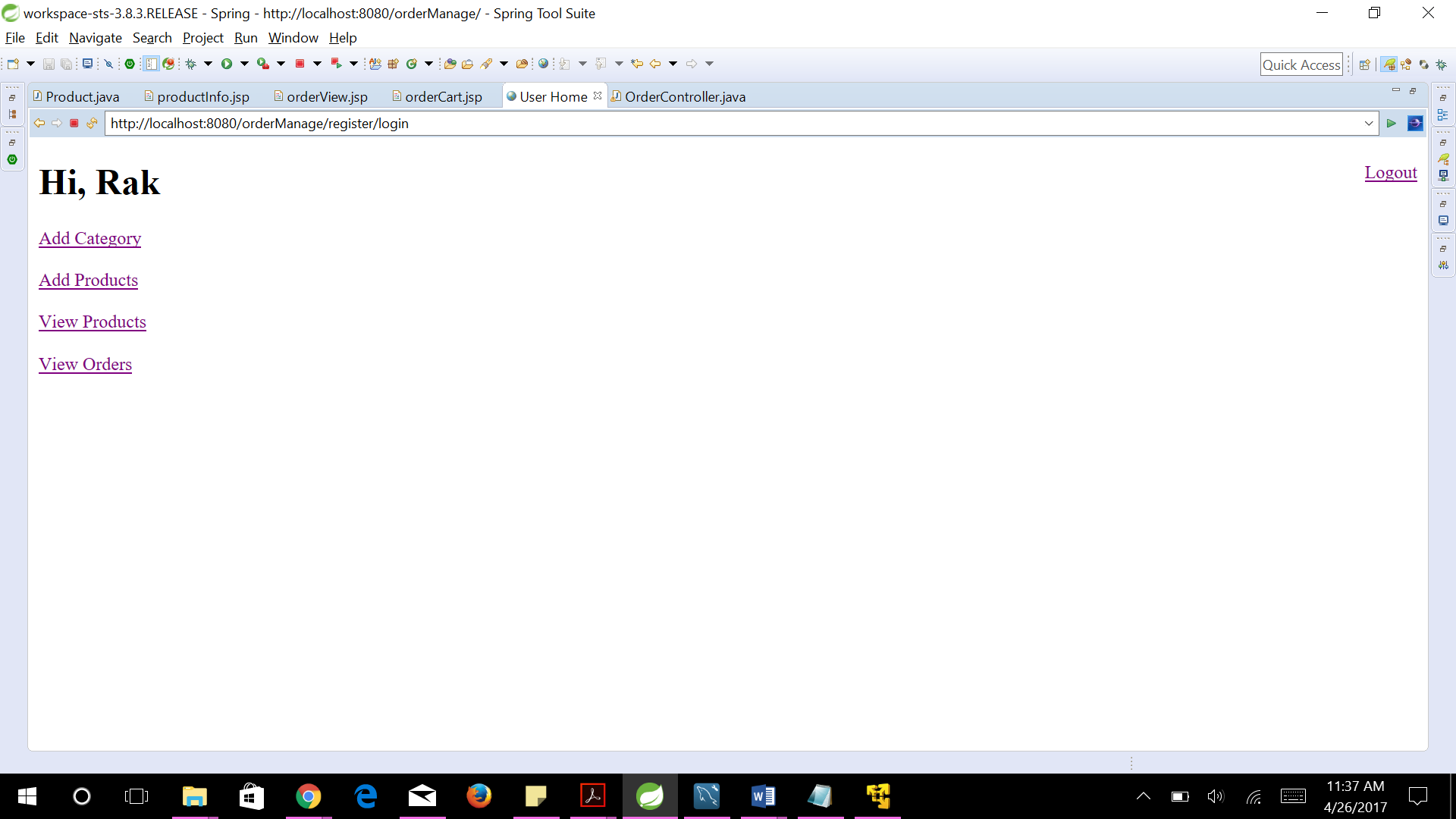


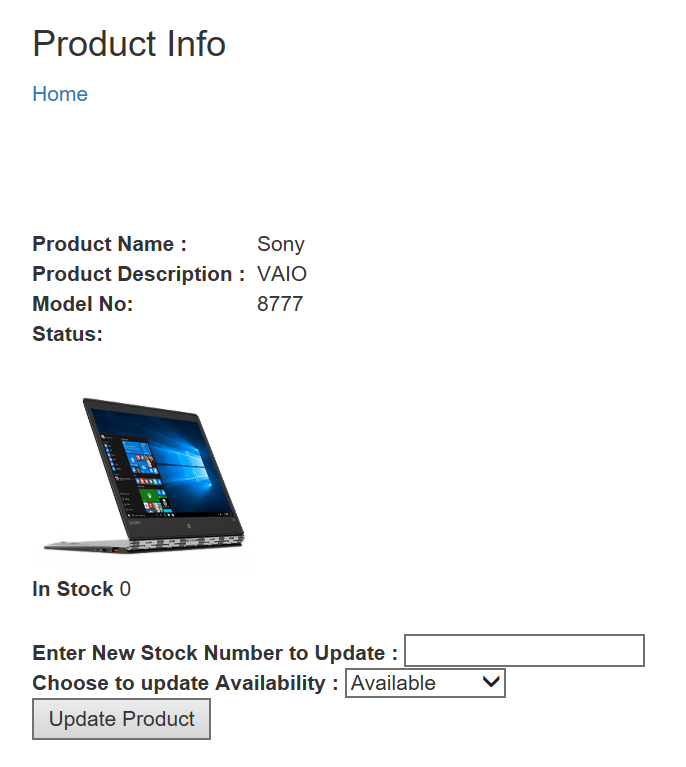


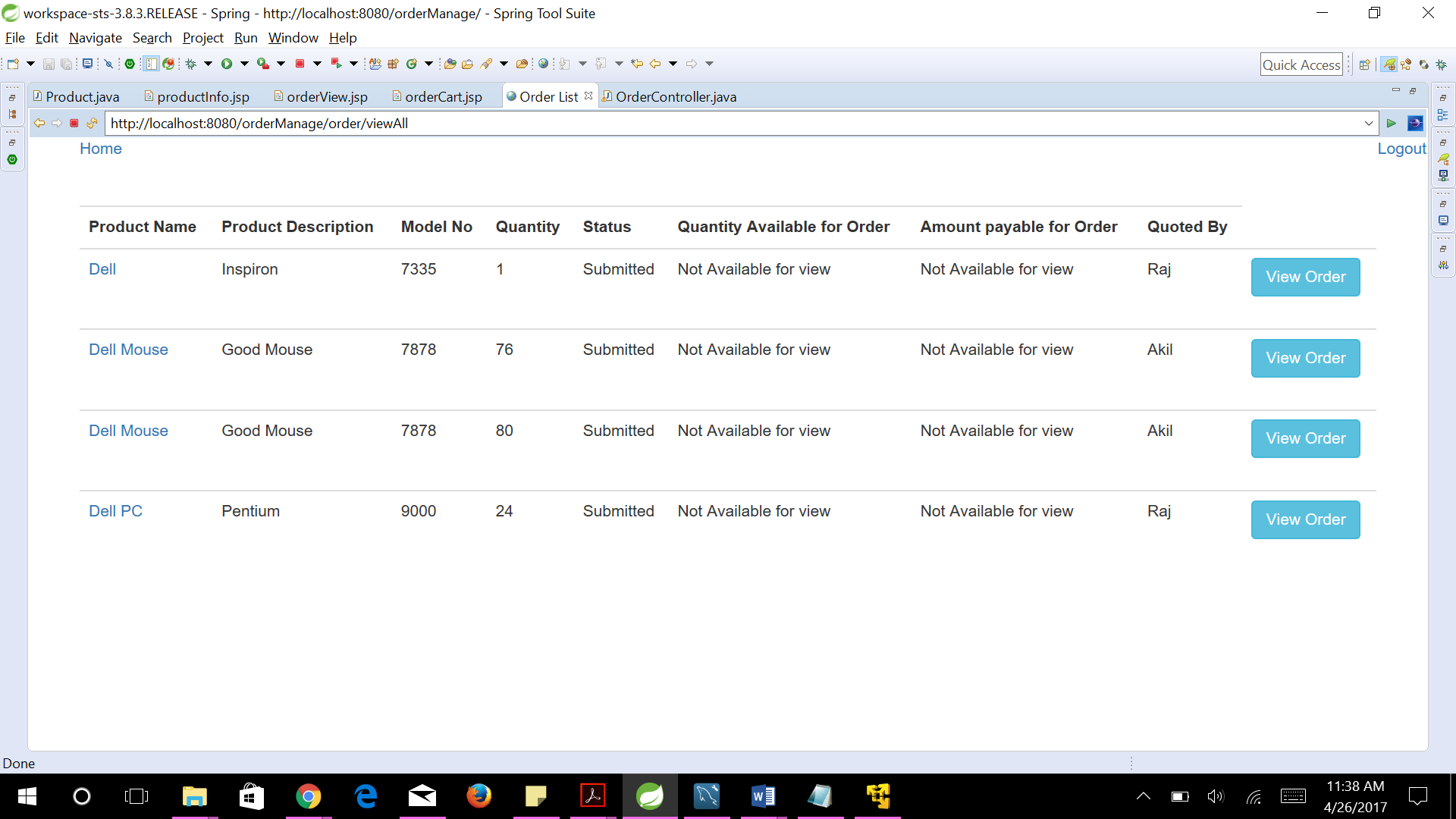


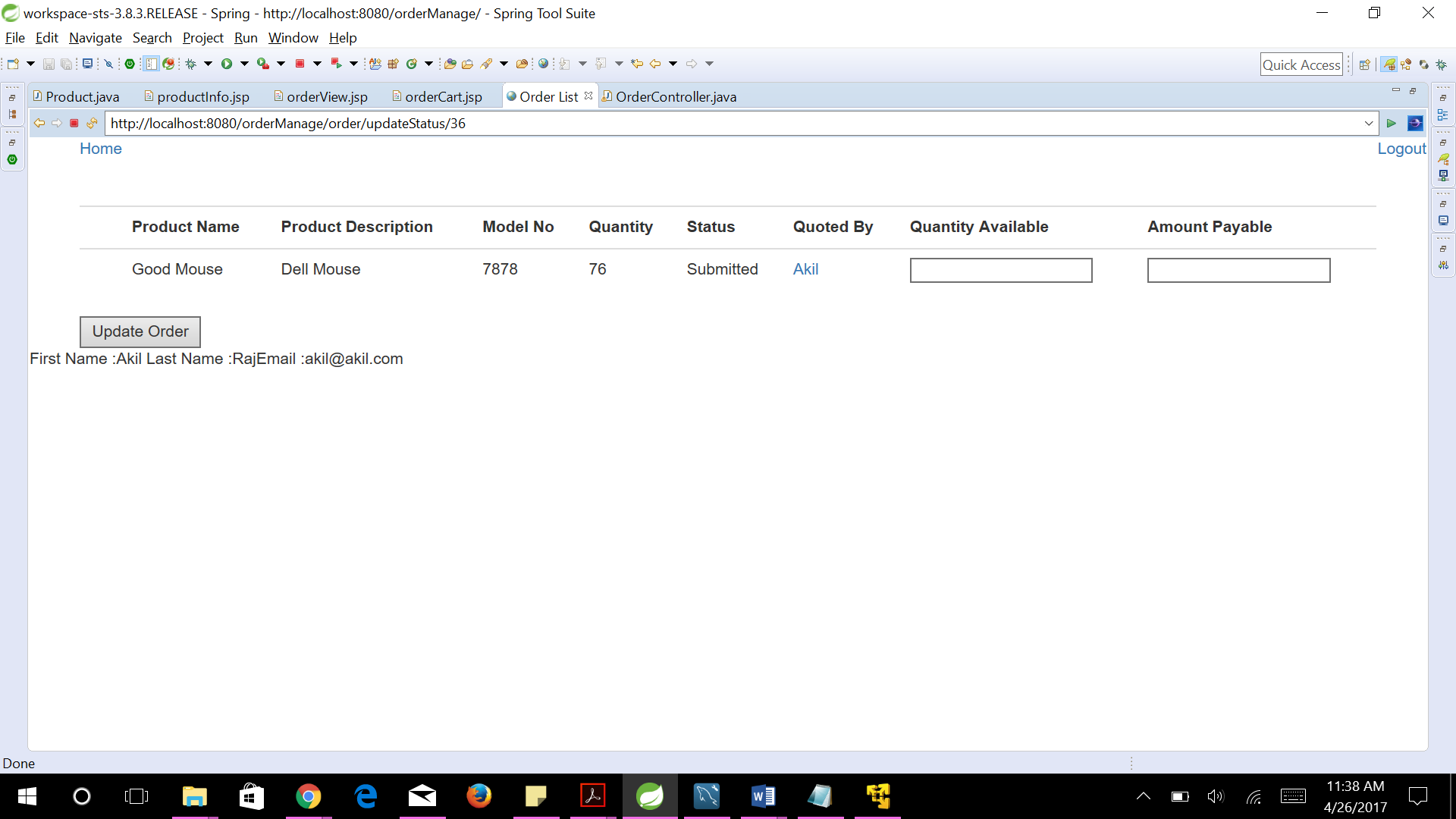


Support Login









**Controller Codes:**

package com.ordermanage.controller;

import java.io.IOException;

import java.util.ArrayList;

import java.util.HashSet;

import java.util.Iterator;

import java.util.List;

import java.util.Set;

import javax.servlet.http.HttpSession;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.hibernate.Session;

import org.json.simple.JSONObject;

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

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

import org.springframework.remoting.httpinvoker.HttpInvokerRequestExecutor;

import org.springframework.stereotype.Controller;

import org.springframework.validation.BindingResult;

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

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

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

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

import org.springframework.web.servlet.ModelAndView;

import com.ordermanage.dao.OrderDAO;

import com.ordermanage.dao.UserDAO;

import com.ordermanage.dao.ProductDAO;

import com.ordermanage.dao.QuoteDAO;

import com.ordermanage.exception.OrderException;

import com.ordermanage.exception.ProductException;

import com.ordermanage.exception.QuoteException;

import com.ordermanage.exception.UserException;

import com.ordermanage.pojo.Category;

import com.ordermanage.pojo.Order;

import com.ordermanage.pojo.OrderQuantity;

import com.ordermanage.pojo.Product;

import com.ordermanage.pojo.Quote;

import com.ordermanage.pojo.User;

@Controller

//@RequestMapping("/order/\*")

public class OrderController {

@Autowired

@Qualifier("userDao")

UserDAO userDao;

@Autowired

@Qualifier("orderDao")

OrderDAO orderDao;

@Autowired

@Qualifier("productDao")

ProductDAO productDao;

@Autowired

@Qualifier("quoteDao")

QuoteDAO quoteDao;

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

protected ModelAndView addOrder(HttpServletRequest request,HttpServletResponse response) throws Exception {

System.out.print("add Order");

HttpSession session = request.getSession();

// return new ModelAndView("productAdd", "product", product);

try {

Order order = new Order();

String names[] = request.getParameterValues("checkedProduct");

String quantity[] = request.getParameterValues("quantity");

Set<Product> orderedProducts = new HashSet<Product>();

if (names==null){

// response.sendRedirect("productSearchResults.jsp");

session.setAttribute("errormissing", "Select to add to Cart");

String searchBy = (String) session.getAttribute("searchBy");

if (searchBy.equals("category")) {

List<Product> searchedProductsCat = (List<Product>) session.getAttribute("searchedProductsCat");

return new ModelAndView("productSearchResults", "searchedProductsCat", searchedProductsCat);

}

else {

List<Product> searchedProducts = (List<Product>) session.getAttribute("searchedProducts");

return new ModelAndView("productSearchResults", "searchedProducts", searchedProducts);

}

}

for (String productName : names) {

Product p = productDao.get(productName);

Quote quote = new Quote();

quote.setOrder(order);

quote.setProduct(p);

quote.setQuantity(0);

quote.setStatus("InCart");

orderedProducts.add(p);

// p.getQuote().add(quote);

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

order.setPostedBy((int) user.getPersonID());

order.setUser(user);

User u = userDao.get(order.getPostedBy());

order.getQuote().add(quote);

session.setAttribute("products", orderedProducts);

session.setAttribute("order", order);

order = orderDao.create(order);

quote = quoteDao.create(quote);

}

/\*

\* for(Product p: order.get){ p = productDao.get(p.getName());

\* p.getOrders().add(order); productDao.update(p); }

\*/

return new ModelAndView("orderAdd", "order", order);

}

catch (QuoteException e) {

System.out.println(e.getMessage());

return new ModelAndView("error", "errorMessage", "error while login");

}

}

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

protected ModelAndView cartOrder(HttpServletRequest request) throws Exception {

HttpSession session = request.getSession();

long userid = (Long) session.getAttribute("userid");

System.out.println("userid = " + userid);

// Set<Product> order edProducts=

// (Set<Product>)session.getAttribute("products");

List<Object> quotes = (List<Object>) orderDao.getOrdersinCart(userid);

int count = quotes.size();

session.setAttribute("size", count);

System.out.println("size==" + count);

List<OrderQuantity> oqListCart = new ArrayList<OrderQuantity>();

Iterator itr = quotes.iterator();

while (itr.hasNext()) {

Object[] obj = (Object[]) itr.next();

OrderQuantity oq = new OrderQuantity();

oq.setOrderId((Long) obj[0]);

oq.setProductId((Long) obj[1]);

oq.setProductDescription((String) obj[3]);

oq.setName((String) obj[2]);

oq.setModelNo((String) obj[4]);

oq.setQuantity((Integer) obj[5]);

oq.setStatus((String) obj[6]);

oq.setUsername((String) obj[7]);

oq.setPersonID((Long)obj[8]);

oq.setQuantityAvailable((Integer)obj[9]);

oq.setAmountPayable((Integer)obj[10]);

oqListCart.add(oq);

}

session.setAttribute("oqListCart", oqListCart);

return new ModelAndView("orderCart", "oqListCart", oqListCart);

}

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

protected ModelAndView placeOrder(HttpServletRequest request) throws Exception {

HttpSession session = request.getSession();

String quantity[] = request.getParameterValues("quantity");

Order order = (Order) session.getAttribute("order");

List<OrderQuantity> oqListCart = (List<OrderQuantity>) session.getAttribute("oqListCart");

try {

int count = 0;

int no = 0;

for (OrderQuantity p : oqListCart) {

quoteDao.updateQuantity(p.getOrderId(), p.getProductId(), Integer.parseInt(quantity[count]));

count++;

no++;

}

return new ModelAndView("orderCreated", "no", no);

} catch (QuoteException e) {

System.out.println(e.getMessage());

return new ModelAndView("error", "errorMessage", "error while login");

}

}

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

protected ModelAndView viewOrder(HttpServletRequest request) throws Exception {

HttpSession session = request.getSession();

long userid = (Long) session.getAttribute("userid");

System.out.println("userid = " + userid);

// Set<Product> order edProducts=

// (Set<Product>)session.getAttribute("products");

List<Object> quotes = (List<Object>) orderDao.getOrders(userid);

int count = quotes.size();

session.setAttribute("size", count);

System.out.println("size==" + count);

List<OrderQuantity> oqList = new ArrayList<OrderQuantity>();

Iterator itr = quotes.iterator();

while (itr.hasNext()) {

Object[] obj = (Object[]) itr.next();

OrderQuantity oq = new OrderQuantity();

oq.setOrderId((Long) obj[0]);

oq.setProductId((Long) obj[1]);

oq.setProductDescription((String) obj[3]);

oq.setName((String) obj[2]);

oq.setModelNo((String) obj[4]);

oq.setQuantity((Integer) obj[5]);

oq.setStatus((String) obj[6]);

oq.setUsername((String) obj[7]);

oq.setPersonID((Long)obj[8]);

oq.setQuantityAvailable((Integer)obj[9]);

oq.setAmountPayable((Integer)obj[10]);

oqList.add(oq);

}

return new ModelAndView("orderView", "oqList", oqList);

}

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

protected ModelAndView viewAllOrder(HttpServletRequest request) throws Exception {

HttpSession session = request.getSession();

long userid = (Long) session.getAttribute("userid");

System.out.println("userid = " + userid);

// Set<Product> order edProducts=

// (Set<Product>)session.getAttribute("products");

List<Object> quotes = (List<Object>) orderDao.getAllOrders();

int count = quotes.size();

session.setAttribute("size", count);

System.out.println("size==" + count);

List<OrderQuantity> oqList = new ArrayList<OrderQuantity>();

Iterator itr = quotes.iterator();

while (itr.hasNext()) {

Object[] obj = (Object[]) itr.next();

OrderQuantity oq = new OrderQuantity();

oq.setOrderId((Long) obj[0]);

oq.setProductId((Long) obj[1]);

oq.setProductDescription((String) obj[3]);

oq.setName((String) obj[2]);

oq.setModelNo((String) obj[4]);

oq.setQuantity((Integer) obj[5]);

oq.setStatus((String) obj[6]);

oq.setUsername((String) obj[7]);

oq.setPersonID((Long)obj[8]);

oq.setQuantityAvailable((Integer)obj[9]);

oq.setAmountPayable((Integer)obj[10]);

oqList.add(oq);

}

return new ModelAndView("orderView", "oqList", oqList);

}

@RequestMapping(value = "/order/updateStatus/{orderId}", method = RequestMethod.GET)

public String updateOrderStatus(@PathVariable("orderId") String orderId, HttpServletRequest request)

throws Exception {

// String orderidSelected= request.getParameter("orderId");

Long orderiduse = Long.parseLong(orderId);

List<Object> quotes = (List<Object>) orderDao.getOrdersbyOrderId(orderiduse);

List<OrderQuantity> updateOrderList = new ArrayList<OrderQuantity>();

Iterator itr = quotes.iterator();

while (itr.hasNext()) {

Object[] obj = (Object[]) itr.next();

OrderQuantity oq = new OrderQuantity();

oq.setOrderId((Long) obj[0]);

oq.setProductId((Long) obj[1]);

oq.setProductDescription((String) obj[2]);

oq.setName((String) obj[3]);

oq.setModelNo((String) obj[4]);

oq.setQuantity((Integer) obj[5]);

oq.setStatus((String) obj[6]);

oq.setUsername((String) obj[7]);

oq.setPersonID((Long)obj[8]);

System.out.println("oq.setPersonID"+oq.getPersonID());

updateOrderList.add(oq);

}

HttpSession sessionOrd = request.getSession();

sessionOrd.setAttribute("updateOrderList", updateOrderList);

return ("orderUpdateStatus");

}

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

public void userInfo(HttpServletRequest request, HttpServletResponse response) throws IOException{

response.setContentType("text/html");

// JSONObject obj = new JSONObject();

try {

String useridReq = request.getParameter("userid");

System.out.println("in ajax"+useridReq);

User user = userDao.get(Long.parseLong(useridReq));

String ret = "First Name :"+user.getFirstName() + "\n"+ "Last Name :"+user.getLastName() + "" + "Email :"+user.getEmail();

response.getWriter().print(ret);

//obj.put("First Name ", user.getFirstName());

// obj.put("Last Name ", user.getLastName());

//obj.put("Email ", user.getEmail());

} catch (UserException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

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

public String updateOrder(

HttpServletRequest request) throws Exception {

// String orderidSelected= request.getParameter("orderId");

Long orderiduse = Long.parseLong(request.getParameter("orderid"));

Long prouctiduse = Long.parseLong(request.getParameter("productid"));

HttpSession session = request.getSession();

String quantity =(String) request.getParameter("quantityAvailable");

String amount = (String) request.getParameter("amountPayable");

quoteDao.updateQuoteAmount(orderiduse, prouctiduse, Integer.parseInt(quantity), Integer.parseInt(amount));

return ("orderUpdated");

}

}

package com.ordermanage.controller;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

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

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

import org.springframework.stereotype.Controller;

import org.springframework.validation.BindingResult;

import org.springframework.web.bind.WebDataBinder;

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

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

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

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

import org.springframework.web.servlet.ModelAndView;

import com.ordermanage.pojo.Category;

import com.ordermanage.exception.CategoryException;

import com.ordermanage.dao.CategoryDAO;

import com.ordermanage.validator.CategoryValidator;

@Controller

@RequestMapping("/category/\*")

public class CategoryController {

@Autowired

@Qualifier("categoryDao")

CategoryDAO categoryDao;

@Autowired

@Qualifier("categoryValidator")

CategoryValidator categoryValidator;

@InitBinder

private void initBinder(WebDataBinder binder) {

binder.setValidator(categoryValidator);

}

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

protected ModelAndView addCategory (@ModelAttribute("category") Category category, BindingResult result,HttpServletRequest request) throws Exception {

System.out.print("add Category");

HttpSession session = request.getSession();

String userType= (String)session.getAttribute("userType");

if (userType=="customer"){

//System.out.println("customer-1");

return new ModelAndView ("error");

}

categoryValidator.validate(category, result);

if (result.hasErrors()) {

return new ModelAndView("categoryAdd", "category", category);

}

try {

category = categoryDao.create(category.getdescription());

} catch (CategoryException e) {

System.out.println(e.getMessage());

return new ModelAndView("error", "errorMessage", "error while login");

}

return new ModelAndView("categoryCreated", "category", category);

}

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

public ModelAndView addCategory(HttpServletRequest request) throws Exception {

HttpSession session = request.getSession();

String userType= (String)session.getAttribute("userType");

if (userType=="customer"){

//System.out.println("customer-1");

return new ModelAndView ("error");

}

return new ModelAndView("categoryAdd", "category", new Category());

}

}

package com.ordermanage.controller;

import java.io.File;

import java.util.ArrayList;

import java.util.Iterator;

import java.util.List;

import java.util.Set;

import javax.servlet.ServletContext;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

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

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

import org.springframework.stereotype.Controller;

import org.springframework.validation.BindingResult;

import org.springframework.web.bind.WebDataBinder;

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

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

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

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

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

import org.springframework.web.multipart.commons.CommonsMultipartFile;

import org.springframework.web.servlet.ModelAndView;

import com.ordermanage.pojo.Category;

import com.ordermanage.pojo.Order;

import com.ordermanage.pojo.OrderQuantity;

import com.ordermanage.pojo.User;

import com.ordermanage.dao.UserDAO;

import com.ordermanage.dao.CategoryDAO;

import com.ordermanage.dao.OrderDAO;

import com.ordermanage.dao.ProductDAO;

import com.ordermanage.dao.QuoteDAO;

import com.ordermanage.exception.ProductException;

import com.ordermanage.pojo.Product;

import com.ordermanage.validator.ProductValidator;

@Controller

// @RequestMapping("/product/\*")

public class ProductController {

@Autowired

@Qualifier("productDao")

ProductDAO productDao;

@Autowired

@Qualifier("categoryDao")

CategoryDAO categoryDao;

@Autowired

@Qualifier("userDao")

UserDAO userDao;

@Autowired

@Qualifier("orderDao")

OrderDAO orderDao;

@Autowired

@Qualifier("quoteDao")

QuoteDAO quoteDao;

@Autowired

@Qualifier("productValidator")

ProductValidator productValidator;

@InitBinder

private void initBinder(WebDataBinder binder) {

binder.setValidator(productValidator);

}

@Autowired

ServletContext servletContext;

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

protected ModelAndView addProduct(@ModelAttribute("product") Product product, BindingResult result,HttpServletRequest request)

throws Exception {

HttpSession session = request.getSession();

String userType= (String)session.getAttribute("userType");

if (userType=="customer"){

//System.out.println("customer-1");

return new ModelAndView ("error");

}

System.out.print("add Product");

productValidator.validate(product, result);

if (result.hasErrors()) {

return new ModelAndView("productAdd", "product", product);

}

try {

User u = userDao.get(product.getPostedBy());

product.setUser(u);

String filename=product.getName()+product.getProductId();

product.setFilename(filename);

if (product.getFilename().trim() != "" || product.getFilename() != null) {

File directory;

String check = File.separator; // Checking if system is linux

// based or windows based by

// checking seprator used.

String path = null;

if (check.equalsIgnoreCase("\\")) {

path = servletContext.getRealPath("").replace("build\\", ""); // gives real path as Lab9/build/web/

// so we need to replace build in the path

}

if (check.equalsIgnoreCase("/")) {

path = servletContext.getRealPath("").replace("build/", "");

path += "/"; // Adding trailing slash for Mac systems.

}

directory = new File(path + "\\" + product.getFilename());

boolean temp = directory.exists();

if (!temp) {

temp = directory.mkdir();

}

if (temp) {

// We need to transfer to a file

CommonsMultipartFile photoInMemory = product.getPhoto();

String fileName = photoInMemory.getOriginalFilename();

// could generate file names as well

File localFile = new File(directory.getPath(), fileName);

// move the file from memory to the file

photoInMemory.transferTo(localFile);

product.setFilename(localFile.getPath());

System.out.println("File is stored at" + localFile.getPath());

System.out.print("registerNewUser");

product = productDao.create(product);

for (Category c : product.getCategories()) {

c = categoryDao.get(c.getdescription());

c.getProducts().add(product);

categoryDao.update(c);

}

} else {

System.out.println("Failed to create directory!");

}

}

return new ModelAndView("productCreated", "product", product);

}

catch (ProductException e) {

System.out.println(e.getMessage());

return new ModelAndView("error", "errorMessage", "error while login");

}

}

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

public ModelAndView addProduct(HttpServletRequest request) throws Exception {

HttpSession session = request.getSession();

String userType= (String)session.getAttribute("userType");

if (userType=="customer"){

//System.out.println("customer-1");

return new ModelAndView ("error");

}

ModelAndView mv = new ModelAndView();

mv.addObject("categories", categoryDao.list());

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

mv.setViewName("productAdd");

return mv;

}

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

public ModelAndView viewProduct(HttpServletRequest request) throws Exception {

try {

List<Product> products = productDao.list();

return new ModelAndView("productList", "products", products);

} catch (ProductException e) {

System.out.println(e.getMessage());

return new ModelAndView("error", "errorMessage", "error while showing product list");

}

}

@RequestMapping(value = "/product/info/{productId}", method = RequestMethod.GET)

public ModelAndView viewProductInfo(@PathVariable("productId")String productId,HttpServletRequest request) throws Exception {

try {

List<Product> products = productDao.getProductInfo(Long.parseLong(productId));

Product inf = new Product();

for (Product p: products){

inf.setName(p.getName());

inf.setProductId(p.getProductId());

inf.setProductDescription(p.getProductDescription());

inf.setModelNo(p.getModelNo());

inf.setFilename(p.getFilename());

inf.setStatus(p.getStatus());

inf.setInStock(p.getInStock());

}

return new ModelAndView("productInfo", "products", inf);

} catch (ProductException e) {

System.out.println(e.getMessage());

return new ModelAndView("error", "errorMessage", "error while showing product list");

}

}

@RequestMapping(value = "/product/update/{productId}", method = RequestMethod.GET)

public ModelAndView updateProductInfo(@PathVariable("productId")String productId,HttpServletRequest request) throws Exception {

try {

Product product = productDao.get(Long.parseLong(productId));

//Product inf = new Product();

String inStock = request.getParameter("inStock");

String availability= request.getParameter("status");

product.setInStock(Integer.parseInt(inStock));

product.setStatus(availability);

productDao.update(product);

return new ModelAndView("productCreated","product",product);

} catch (ProductException e) {

System.out.println(e.getMessage());

return new ModelAndView("error", "errorMessage", "error while showing product list");

}

}

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

public ModelAndView searchProductPage(HttpServletRequest request) throws Exception {

return new ModelAndView("searchProduct");

}

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

public ModelAndView searchProduct(HttpServletRequest request) throws Exception {

String searchBy = request.getParameter("searchby");

String keyword = request.getParameter("keyword");

HttpSession session = request.getSession();

session.setAttribute("searchBy", searchBy);

try {

if (searchBy.equals("category")) {

List<Object> pcList = (List<Object>) productDao.searchByCategory(searchBy, keyword);

int count = pcList.size();

// session.setAttribute("size", count);

System.out.println("size search Product==" + count);

List<Product> searchedProductsCat = new ArrayList<Product>();

Iterator itr = pcList.iterator();

while (itr.hasNext()) {

Object[] obj = (Object[]) itr.next();

Product oq = new Product();

oq.setName(((String) obj[0]));

oq.setProductDescription((String) obj[1]);

oq.setModelNo((String) obj[2]);

// oq.setCategories((Integer)obj[4]);

searchedProductsCat.add(oq);

session.setAttribute("searchedProductsCat", searchedProductsCat);

}

return new ModelAndView("productSearchResults", "searchedProductsCat", searchedProductsCat);

} else {

List<Product> searchedProducts = (List<Product>) productDao.search(searchBy, keyword);

session.setAttribute("searchedProducts", searchedProducts);

return new ModelAndView("productSearchResults", "searchedProducts", searchedProducts);

}

} catch (ProductException e) {

System.out.println(e.getMessage());

return new ModelAndView("error", "errorMessage", "error while login");

}

}

}

package com.ordermanage.controller;

import java.io.IOException;

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.beans.factory.annotation.Qualifier;

import org.springframework.stereotype.Controller;

import org.springframework.validation.BindingResult;

import org.springframework.web.bind.WebDataBinder;

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

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

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

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

import org.springframework.web.servlet.ModelAndView;

import com.ordermanage.dao.UserDAO;

import com.ordermanage.exception.UserException;

import com.ordermanage.pojo.User;

import com.ordermanage.validator.UserValidator;

import org.json.simple.JSONObject;;

@Controller

//@RequestMapping("/user/\*")

public class UserController {

// annotations go here

@Autowired // if there is just one bean of this type, spring provides IoC

// automatically

UserDAO userDao;

@Autowired

@Qualifier("userValidator")

UserValidator validator;

@InitBinder

private void initBinder(WebDataBinder binder) {

binder.setValidator(validator);

}

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

protected String goToUserHome(HttpServletRequest request) throws Exception {

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

User uS = (User) session.getAttribute("user");

User userSupport = userDao.getSupport(uS.getUsername(), uS.getPassword());

if ((userSupport == null)) {

return "home";

} else {

return "supportHome";

}

}

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

protected ModelAndView registerUser() throws Exception {

System.out.print("registerUser");

return new ModelAndView("register", "user", new User());

}

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

protected String loginUser(HttpServletRequest request) throws Exception {

try {

// add

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

System.out.println("loginUser");

User u = userDao.get(request.getParameter("username"), request.getParameter("password"));

if (u == null) {

System.out.println("UserName/Password does not exist");

session.setAttribute("errorMessage", "UserName/Password does not exist");

return "error";

}

session.setAttribute("user", u);

User userSupport = userDao.getSupport(request.getParameter("username"), request.getParameter("password"));

if ((userSupport == null)) {

session.setAttribute("userType", "customer");

return "home";

} else {

session.setAttribute("userType", "support");

return "supportHome";

}

} catch (UserException e) {

System.out.println("Exception: " + e.getMessage());

// session.setAttribute("errorMessage", "error while login");

return "error";

}

}

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

protected ModelAndView registerNewUser(HttpServletRequest request, @ModelAttribute("user") User user,

BindingResult result) throws Exception {

// add

validator.validate(user, result);

if (result.hasErrors()) {

return new ModelAndView("register", "user", user);

}

try {

// add

System.out.println("register New User");

User u = userDao.register(user);

request.getSession().setAttribute("user", u);

return new ModelAndView("home", "user", u);

} catch (UserException e) {

System.out.println("Exception: " + e.getMessage());

return new ModelAndView("error", "errorMessage", "error while login");

}

}

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

protected void logoutUser(HttpServletRequest request, HttpServletResponse response) throws Exception {

HttpSession session = request.getSession();

session.invalidate();

response.sendRedirect("http://localhost:8080/orderManage/index.jsp");

// return ("index");

}

}