

```
package com.sportyshoe.controller;
import java.util.ArrayList;
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.ModelMap;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
```

```
import com.sportyshoe.bean.CartItem;
import com.sportyshoe.bean.Product;
import com.sportyshoe.service.ProductService;
```

```
@Controller
public class CartController {
```

```
    @Autowired
    private ProductService productService;
```

```
//    @Autowired
//    private PurchaseService purchaseService;
//
//    @Autowired
//    private PurchaseItemService purchaseItemService;
```

```
    @SuppressWarnings("unchecked")
    @RequestMapping(value = "/showCart", method = RequestMethod.GET)
    public String showCart(ModelMap map, javax.servlet.http.HttpServletRequest request)
    {
        // check if user is logged in
        HttpSession session = request.getSession();
        if (session.getAttribute("user_id") == null) {
            map.addAttribute("error", "Error, You need to login before adding items to cart");
        } else {
            // if cart is already in session then retrieve it else create a new cart list and
            // save it to session
            List<CartItem> cartItems = new ArrayList<CartItem>();
            if (session.getAttribute("cartItems") != null)
                cartItems = (List<CartItem>) session.getAttribute("cartItems");

            // get total of all cart items
            double totalValue = getCartValue(cartItems);
            map.addAttribute("cartTotal", totalValue);
            map.addAttribute("cartItems", cartItems);
        }

        map.addAttribute("pageTitle", "SPORTY SHOES - YOUR CART");
    }
}
```

```

    return "cartDetails";
}

@SuppressWarnings("unchecked")
@RequestMapping(value = "/addToCart", method = RequestMethod.GET)
public String cartAddItem(@RequestParam String id, ModelMap map, HttpServletRequest request)
{
    HttpSession session = request.getSession();
    int idValue = Integer.parseInt(id);
    System.out.println("I am in addToCart");

    // if cart is already in session then retrieve it else create a new cart list and
    // save it to session
    List<CartItem> cartItems = new ArrayList<CartItem>();
    if (session.getAttribute("cartItems") != null)
        cartItems = (List<CartItem>) session.getAttribute("cartItems");

    if (isItemInCart(cartItems, idValue)) {
        map.addAttribute("error", "This item is already in your cart");
    } else {
        Product product = productService.findProduct(idValue);
        CartItem item = new CartItem();

        item.setProductId(idValue);
        item.setQty(1);
        item.setRate(product.getListPrice());
        double dprice = item.getRate() * item.getQty();
        item.setPrice(dprice);
        item.setProductName(product.getProductName());
        cartItems.add(item);

        session.setAttribute("cartItems", cartItems);
        session.setAttribute("cartTotal", getCartValue(cartItems));
    }
    return "redirect:home";
}

```

```

@SuppressWarnings("unchecked")
@RequestMapping(value = "/cartDeleteItem", method = RequestMethod.GET)
public String cartDeleteItem(ModelMap map, javax.servlet.http.HttpServletRequest request,
    @RequestParam(value="id", required=true) String id)
{
    // check if user is logged in
    HttpSession session = request.getSession();
    int idValue = Integer.parseInt(id);
    List<CartItem> cartItems = new ArrayList<CartItem>();

    if (session.getAttribute("cartItems") != null)
        cartItems = (List<CartItem>) session.getAttribute("cartItems");

    for(CartItem item: cartItems) {
        if (item.getProductId() == idValue) {
            cartItems.remove(item);
            session.setAttribute("cartItems", cartItems);
        }
    }
}

```

```

        session.setAttribute("cartTotal", getCartValue(cartItems));
        break;
    }
}

return "redirect:showCart";
}

@SuppressWarnings("unchecked")
@RequestMapping(value = "/checkout", method = RequestMethod.GET)
public String checkout(ModelMap map, javax.servlet.http.HttpServletRequest request)
{
    // check if user is logged in
    HttpSession session = request.getSession();
    if (session.getAttribute("user_id") == null) {
        request.setAttribute("message", "You must login to check out");
        return "login";
    } else {
        List<CartItem> cartItems = new ArrayList<CartItem>();
        if (session.getAttribute("cart_items") != null)
            cartItems = (List<CartItem>) session.getAttribute("cart_items");
        double totalValue = getCartValue(cartItems);
        map.addAttribute("cartValue", totalValue);
        map.addAttribute("cartItems", cartItems);
    }
    map.addAttribute("pageTitle", "SPORTY SHOES - CHECKOUT");
    return "checkOutDetails";
}

// @RequestMapping(value = "/completepurchase", method = RequestMethod.GET)
// public String completePurchase(ModelMap map, javax.servlet.http.HttpServletRequest request)
// {
//     // check if user is logged in
//     HttpSession session = request.getSession();
//     if (session.getAttribute("user_id") == null) {
//         map.addAttribute("error", "Error, You need to login before completing purchase");
//     } else {
//         // take items from cart and update the databae
//         List<CartItem> cartItems = new ArrayList<CartItem>();
//         if (session.getAttribute("cart_items") != null)
//             cartItems = (List<CartItem>) session.getAttribute("cart_items");
//         BigDecimal totalValue = getCartValue(cartItems);
//
//         long userId = (Long) session.getAttribute("user_id") ;
//
//         Purchase purchase = new Purchase();
//         purchase.setUserId(userId);
//         purchase.setDate(Calendar.getInstance().getTime());
//         purchase.setTotal(totalValue);
//         long purchaseId = purchaseService.updatePurchase(purchase);
//
//         for(CartItem item: cartItems) {
//             PurchaseItem pItem = new PurchaseItem();
//             pItem.setPurchaseId(purchaseId);

```

```

//            pItem.setProductId(item.getProductId());
//            pItem.setUserId(userId);
//            pItem.setRate(item.getRate());
//            pItem.setQty(item.getQty());
//            pItem.setPrice(item.getPrice());
//
//            purchaseItemService.updateItem(pItem);
//        }
//        map.addAttribute("cartValue", totalValue);
//        map.addAttribute("cartItems", cartItems);
//    }
//
//    return "redirect:confirm";
// }

@SuppressWarnings("unchecked")
@RequestMapping(value = "/gateway", method = RequestMethod.GET)
public String gateway(ModelMap map, javax.servlet.http.HttpServletRequest request)
{
    // check if user is logged in
    HttpSession session = request.getSession();
    if (session.getAttribute("user_id") == null) {
        map.addAttribute("error", "Error, You need to login before making payment");
    } else {
        List<CartItem> cartItems = new ArrayList<CartItem>();
        if (session.getAttribute("cartItems") != null)
            cartItems = (List<CartItem>) session.getAttribute("cartItems");
        double totalValue = getCartValue(cartItems);
        map.addAttribute("cartValue", totalValue);
        map.addAttribute("cartItems", cartItems);
    }

    map.addAttribute("pageTitle", "SPORTY SHOES - PAYMENT GATEWAY");
    return "gateway";
}

```

```

@SuppressWarnings("unchecked")
@RequestMapping(value = "/confirm", method = RequestMethod.GET)
public String confirm(ModelMap map, javax.servlet.http.HttpServletRequest request)
{
    // check if user is logged in
    HttpSession session = request.getSession();
    if (session.getAttribute("user_id") == null) {
        map.addAttribute("error", "Error, You need to login before completing the purchase");
    } else {
        // clear items from cart as order has completed
        List<CartItem> cartItems = new ArrayList<CartItem>();
        if (session.getAttribute("cart_items") != null)
            cartItems = (List<CartItem>) session.getAttribute("cartItems");
        double totalValue = getCartValue(cartItems);
        map.addAttribute("cartValue", totalValue);
    }
}

```

```

        cartItems.clear();
        session.setAttribute("cart_items", null);
    }
    map.addAttribute("pageTitle", "SPORTY SHOES - PURCHASE CONFIRMATION");
    return "confirm";
}

/**
 * Check if an item is already in the cart
 * @param list
 * @param item
 * @return
 */
private boolean isItemInCart(List<CartItem> list, long item) {
    boolean retVal = false;

    for(CartItem thisItem: list) {
        if (item == thisItem.getProductId()) {
            retVal = true;
            break;
        }
    }
    return retVal;
}

/**
 * Get total value of items in cart
 * @param list
 * @return
 */
private double getCartValue(List<CartItem> list) {
    double total = 0.0;

    for(CartItem item: list) {
        double dprice = item.getRate() * (item.getQty());
        total += dprice;
    }
    return total;
}
}

```

```
package com.sportyshoe.bean;
```

```
public class CartItem {
```

```
    private int productId;  
    private String productName;  
    private double rate;  
    private int qty;  
    private double price;
```

```
    public int getProductId() {  
        return productId;  
    }
```

```
    public void setProductId(int productId) {  
        this.productId = productId;  
    }
```

```
    public String getProductName() {  
        return productName;  
    }
```

```
    public void setProductName(String productName) {  
        this.productName = productName;  
    }
```

```
    public double getRate() {  
        return rate;  
    }
```

```
    public void setRate(double rate) {  
        this.rate = rate;  
    }
```

```
    public int getQty() {  
        return qty;  
    }
```

```
    public void setQty(int qty) {  
        this.qty = qty;  
    }
```

```
    public double getPrice() {  
        return price;  
    }
```

```
    public void setPrice(double price) {  
        this.price = price;  
    }
```

```
    public CartItem(int productId, String productName, double rate, int qty, double price) {  
        super();  
        this.productId = productId;  
        this.productName = productName;  
        this.rate = rate;  
        this.qty = qty;  
        this.price = price;  
    }
```

```
    public CartItem() {  
        super();  
        // TODO Auto-generated constructor stub
```

```
    }  
    @Override  
    public String toString() {  
        return "CartItem [productId=" + productId + ", productName=" + productName + ", rate=" + rate + ", qty=" +  
qty        + ", price=" + price + "];"  
    }  
  
}
```

```
package com.sportyshoe.controller;
```

```
import java.util.Date;
```

```
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.web.bind.annotation.RequestMapping;
```

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

```
import org.springframework.web.servlet.ModelAndView;
```

```
import com.sportyshoe.bean.User;
```

```
import com.sportyshoe.service.LoginService;
```

```
@Controller
```

```
public class LoginController {
```

```
    @Autowired
```

```
    LoginService loginService;
```

```
    @RequestMapping(value = "/loginForm")
```

```
    public ModelAndView loginForm(HttpServletRequest req) {
```

```
        ModelAndView mav = new ModelAndView();
```

```
        mav.setViewName("login");
```

```
        return mav;
```

```
    }
```

```
    @RequestMapping(value="/signIn", method=RequestMethod.POST)
```

```
    public ModelAndView signIn(HttpServletRequest req) {
```

```
        User ll = new User();
```

```
        ll.setEmailId(req.getParameter("email"));
```

```
        ll.setPwd(req.getParameter("password"));
```

```
        String result = loginService.checkUser(ll, req);
```

```
        ModelAndView mav = new ModelAndView();
```

```
        System.out.println("Value Returned: " + result);
```

```
        if (result.matches("failure")) {
```

```
            mav.setViewName("login");
```

```
        }else if (result.matches("user")){
```

```
            mav.setViewName("cartDetails");
```

```
        }else if (result.matches("admin")) {
```

```
            mav.setViewName("/adminHome");
```

```
        }else {
```

```
            mav.setViewName("index");
```

```
        }
```

```
        return mav;
```

```
    }
```

```
    @RequestMapping(value="signUp", method=RequestMethod.POST)
```

```
    public ModelAndView signUp(HttpServletRequest req) {
```



```
Date date = new Date();
```

```
User ll = new User();  
ll.setEmailId(req.getParameter("email"));  
ll.setPwd(req.getParameter("password"));  
ll.setName(req.getParameter("name"));  
ll.setAddress(req.getParameter("address"));  
ll.setUserRole("user");  
ll.setDateAdded(date);
```

```
String result = loginService.signUp(ll);  
ModelAndView mav = new ModelAndView();
```

```
System.out.println(result);  
mav.setViewName("login");  
mav.addObject("msg", result);
```

```
return mav;
```

```
}
```

```
@RequestMapping(value = "/signupForm")  
public ModelAndView signupForm(HttpServletRequest req) {  
    ModelAndView mav = new ModelAndView();  
    mav.setViewName("signupForm");  
    return mav;  
}
```

```
@RequestMapping(value = "/logout")  
public ModelAndView logout(HttpServletRequest req) {  
    HttpSession session = req.getSession();  
    session.invalidate();
```

```
    ModelAndView mav = new ModelAndView();  
    mav.setViewName("index");  
    return mav;
```

```
}
```

```
}
```

```

package com.sportyshoe.service;

import java.util.Optional;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;

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

import com.sportyshoe.bean.User;
import com.sportyshoe.repository.UserRepository;

@Service
public class LoginService {

    @Autowired
    UserRepository ur;

    public String checkUser(User user, HttpServletRequest request) {
        HttpSession session = request.getSession();
        String email = user.getEmailId();
        String pwd = user.getPwd();

        Optional<User> dbUser = ur.findById(email);
        if (dbUser.isPresent()) {
            String dbEmail = dbUser.get().getEmailId();
            String dbPassword = dbUser.get().getPwd();
            String dbRole = dbUser.get().getUserRole();

            if (dbEmail.matches(email) && dbPassword.matches(pwd)) {
                session.setAttribute("user_id", dbEmail);
                session.setAttribute("user_role", dbRole);
                return dbRole;
            }
            else {
                return "failure";
            }
        }
        else {
            return "failure";
        }
    }

    public String signUp(User ll) {
        Optional<User> user = ur.findById(ll.getEmailId());

        if (user.isPresent()) {
            return "User already exists";
        }
        else {
            ur.save(ll);
            return "User account created.";
        }
    }
}

```



```
package com.sportyshoe.bean;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.Id;
```

```
@Entity
```

```
public class Product {
```

```
    @Id
```

```
    private int id;
```

```
    private String productName;
```

```
    private String productBrand;
```

```
    private String productCategory;
```

```
    private double productSize;
```

```
    private boolean productGender;
```

```
    private String productDescription;
```

```
    private double listPrice;
```

```
    public Product() {
```

```
        super();
```

```
        // TODO Auto-generated constructor stub
```

```
    }
```

```
    public String getProductManufacturer() {
```

```
        return productBrand;
```

```
    }
```

```
    public void setProductManufacturer(String productManufacturer) {
```

```
        this.productBrand = productManufacturer;
```

```
    }
```

```
    public String getProductCategory() {
```

```
        return productCategory;
```

```
    }
```

```
    public void setProductCategory(String productCategory) {
```

```
        this.productCategory = productCategory;
```

```
    }
```

```
    public double getProductSize() {
```

```
        return productSize;
```

```
    }
```

```
    public void setProductSize(double productSize) {
```

```
        this.productSize = productSize;
```

```
    }
```

```
    public boolean isProductGender() {
```

```
        return productGender;
```

```
    }
```

```
    public void setProductGender(boolean productGender) {
```

```
        this.productGender = productGender;
```

```
    }
```

```
public int getId() {  
    return id;  
}
```

```
public void setId(int id) {  
    this.id = id;  
}
```

```
public String getProductName() {  
    return productName;  
}
```

```
public void setProductName(String productName) {  
    this.productName = productName;  
}
```

```
public String getProductDescription() {  
    return productDescription;  
}
```

```
public void setProductDescription(String productDescription) {  
    this.productDescription = productDescription;  
}
```

```
public double getListPrice() {  
    return listPrice;  
}
```

```
public void setListPrice(double listPrice) {  
    this.listPrice = listPrice;  
}
```

```
@Override
```

```
public String toString() {  
    return "Product [id=" + id + ", productName=" + productName + ", productManufacturer=" + productBrand  
        + ", productDescription=" + productDescription + ", productCategory=" + productCategory  
        + ", productSize=" + productSize + ", productGender=" + productGender + ", listPrice=" + listPrice  
        + "]";  
}
```

```
}
```

```
package com.sportyshoe.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.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.RequestParam;
```

```
import com.sportyshoe.bean.Product;  
import com.sportyshoe.service.ProductService;
```

```
@Controller
```

```
public class ProductController {
```

```
    @Autowired  
    ProductService ps;
```

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

```
    public String homePage(HttpServletRequest request) {
```

```
        HttpSession session = request.getSession();  
        List<Product> pList = ps.findAllProducts();  
        System.out.println(pList);  
        request.setAttribute("pList", pList);
```

```
        if (session.getAttribute("cartItems") == null) {  
            session.setAttribute("cartTotal", 0);  
        }
```

```
        return "index";
```

```
    }
```

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

```
    public String addProduct(HttpServletRequest request) {
```

```
        HttpSession session = request.getSession();  
        String user_role = "";
```

```
        if (session.getAttribute("user_role") != null) {  
            user_role = (String)session.getAttribute("user_role");  
        } else {  
            return "login";  
        }
```

```
        if (user_role.matches("admin")) {  
            return "addProduct";  
        } else {  
            return "login";  
        }
```

```
    }
```

```

@RequestMapping(value = "/adminHome", method = RequestMethod.GET)
public String adminHome(HttpServletRequest request) {
    HttpSession session = request.getSession();
    String user_role = "";

    if (session.getAttribute("user_role") != null) {
        user_role = (String)session.getAttribute("user_role");
    } else {
        return "login";
    }

    if (user_role.matches("admin")) {
        List<Product> pList = ps.findAllProducts();
        request.setAttribute("pList", pList);
        return "adminHome";
    } else {
        return "login";
    }
}

@RequestMapping(value = "/getProductDetails", method = RequestMethod.GET)
public String getProductDetails(@RequestParam String id, HttpServletRequest request) {
    int pid = Integer.parseInt(id);
    Product item = ps.findProduct(pid);
    request.setAttribute("item", item);
    System.out.println(item);
    return "productDetails";
}

@RequestMapping(value = "/deleteProduct", method = RequestMethod.GET)
public String deleteProduct(@RequestParam String id, HttpServletRequest request) {
    HttpSession session = request.getSession();
    String user_role = "";

    if (session.getAttribute("user_role") != null) {
        user_role = (String)session.getAttribute("user_role");
    } else {
        return "login";
    }

    if (user_role.matches("admin")) {
        int pid = Integer.parseInt(id);
        String result = ps.deleteProduct(pid);
        request.setAttribute("message", result);
        return "adminHome";
    } else {
        return "login";
    }
}

```

```

@RequestMapping(value = "/storeProduct", method = RequestMethod.POST)
public String storeProduct(HttpServletRequest request) {
    HttpSession session = request.getSession();
    String user_role = "";

    if (session.getAttribute("user_role") != null) {
        user_role = (String)session.getAttribute("user_role");
    } else {
        return "login";
    }

    if (user_role.matches("admin")) {
        Product item = new Product();
        item.setId(Integer.parseInt(request.getParameter("id")));
        item.setProductName(request.getParameter("productName"));
        item.setProductManufacturer(request.getParameter("productBrand"));
        item.setProductCategory(request.getParameter("productCategory"));
        item.setProductDescription(request.getParameter("productDescription"));

        boolean isFemale = request.getParameter("gender").matches("female");
        System.out.println(request.getParameter("gender") + " : " + isFemale);
        item.setProductGender(isFemale);

        item.setProductSize(Double.parseDouble(request.getParameter("productSize")));
        item.setListPrice(Double.parseDouble(request.getParameter("listPrice")));

        String result = ps.updateProduct(item);
        request.setAttribute("message", result);

        return "adminHome";
    } else {
        return "login";
    }
}

```

```

@RequestMapping(value = "/updateProduct", method = RequestMethod.POST)
public String updateProduct(HttpServletRequest request) {
    HttpSession session = request.getSession();

    String user_role = "";

    if (session.getAttribute("user_role") != null) {
        user_role = (String)session.getAttribute("user_role");
    } else {
        return "login";
    }

    if (user_role.matches("admin")) {
        Product item = new Product();
        item.setId(Integer.parseInt(request.getParameter("id")));
    }
}

```



```
item.setProductName(request.getParameter("productName"));
item.setProductManufacturer(request.getParameter("productBrand"));
item.setProductCategory(request.getParameter("productCategory"));
item.setProductDescription(request.getParameter("productDescription"));
```

```
boolean isFemale = request.getParameter("gender").matches("female");
item.setProductGender(isFemale);
```

```
item.setProductSize(Double.parseDouble(request.getParameter("productSize")));
item.setListPrice(Double.parseDouble(request.getParameter("listPrice")));
```

```
String result = ps.updateProduct(item);
request.setAttribute("message", result);
```

```
return "adminHome";
```

```
} else {
    return "login";
}
```

```
}
```

```
}
```

```
package com.sportyshoe.repository;
```

```
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;
```

```
import com.sportyshoe.bean.Product;
```

```
@Repository
```

```
public interface ProductRepository extends JpaRepository<Product, Integer> {
```

```
}
```

```

package com.sportyshoe.service;

import java.util.List;
import java.util.Optional;

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

import com.sportyshoe.bean.Product;
import com.sportyshoe.repository.ProductRepository;

@Service
public class ProductService {
    @Autowired
    ProductRepository pr;

    public List<Product> findAllProducts(){
        return pr.findAll();
    }

    public Product findProduct(int id) {
        Optional<Product> product = pr.findById(id);
        if (product.isPresent() ) {
            return product.get();
        } else {
            return null;}
    }

    public String storeProduct(Product item) {
        boolean existing = (pr.findById(item.getId()) != null);

        if (existing) {
            return "Duplicate product id";
        } else if (pr.save(item) != null) {
            return "Record Stored";
        } else {
            return "Record could not be save";
        }
    }

    public String updateProduct(Product item) {
        boolean existing = (pr.findById(item.getId()) != null);

        if (existing) {
            if (pr.saveAndFlush(item) != null) {return "Record Updated";}
            else {return "Record could not be updated";}
        } else {return "Item does not exist";}
    }

    public String deleteProduct(int id) {
        Optional<Product> product = pr.findById(id);
        if (product.isEmpty()) {
            return "No product was found";
        }
    }
}

```

```
    } else {  
        pr.deleteById(id);  
        return "Product was deleted";  
    }  
}  
  
}
```

```
package com.sportyshoe.bean;

import java.util.Date;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name= "purchases")
public class Purchase {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;
    private String emailId;
    private Date date;
    private double total;

    public Purchase() {
        super();
        // TODO Auto-generated constructor stub
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getEmailId() {
        return emailId;
    }

    public void setEmailId(String emailId) {
        this.emailId = emailId;
    }

    public Date getDate() {
        return date;
    }

    public void setDate(Date date) {
        this.date = date;
    }

    public double getTotal() {
        return total;
    }
}
```

```
public void setTotal(double total) {  
    this.total = total;  
}
```

```
}
```

```
package com.sportyshoe.controller;
```

```
import java.util.ArrayList;
import java.util.Date;
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.ModelMap;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
```

```
import com.sportyshoe.bean.Purchase;
import com.sportyshoe.service.PurchaseService;
```

```
@Controller
```

```
public class PurchaseController {
```

```
    @Autowired
```

```
    private PurchaseService purchaseService;
```

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

```
    public String memberpurchases(ModelMap map, HttpServletRequest request)
```

```
    {
```

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

```
        if (session.getAttribute("user_id") == null) {
```

```
            return "login";
```

```
        }
```

```
        String userId = (String) session.getAttribute("user_id");
```

```
        String result = purchaseService.completePurchase(userId, request);
```

```
        request.setAttribute("message", result);
```

```
        return "receipt";
```

```
    }
```

```
    @RequestMapping(value = "/purchaseToday")
```

```
    public String purchaseToday(HttpServletRequest request) {
```

```
        Date date = new Date();
```

```
        System.out.println("Inside purchase ");
```

```
        List<Purchase> purchaseList = new ArrayList<Purchase>();
```

```
        purchaseList = purchaseService.findPurchaseByDate(date, request);
```

```
        request.setAttribute("purchaseList", purchaseList);
```

```
        return "displayPurchase";
```

```
    }
```

```
}
```

```
package com.sportyshoe.bean;
```

```
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import javax.persistence.GenerationType;  
import javax.persistence.Id;  
import javax.persistence.Table;
```

```
@Entity  
@Table(name= "purchase_items")  
public class PurchaseItem {  
  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private int id;  
    private int purchaseId;  
    private String userEmail;  
    private int productId;  
    private String productName;  
    private double rate;  
    private int qty;  
    private double price;  
  
    public PurchaseItem() {  
        super();  
        // TODO Auto-generated constructor stub  
    }  
    public int getId() {  
        return id;  
    }  
    public void setId(int id) {  
        this.id = id;  
    }  
    public int getPurchaseId() {  
        return purchaseId;  
    }  
    public void setPurchaseId(int purchaseId) {  
        this.purchaseId = purchaseId;  
    }  
    public int getProductId() {  
        return productId;  
    }  
    public void setProductId(int productId) {  
        this.productId = productId;  
    }  
    public double getRate() {  
        return rate;  
    }  
    public void setRate(double rate) {  
        this.rate = rate;  
    }  
    public int getQty() {
```



```
        return qty;
    }
    public void setQty(int qty) {
        this.qty = qty;
    }
    public double getPrice() {
        return price;
    }
    public void setPrice(double price) {
        this.price = price;
    }
    public String getProductName() {
        return productName;
    }
    public void setProductName(String productName) {
        this.productName = productName;
    }
    public String getUserEmail() {
        return userEmail;
    }
    public void setUserEmail(String userEmail) {
        this.userEmail = userEmail;
    }
}

}
```

```
package com.sportyshoe.repository;
```

```
import org.springframework.data.jpa.repository.JpaRepository;
```

```
import org.springframework.stereotype.Repository;
```

```
import com.sportyshoe.bean.PurchaseItem;
```

```
@Repository
```

```
public interface PurchaseItemRepository extends JpaRepository<PurchaseItem, Integer>{
```

```
}
```

```
package com.sportyshoe.service;
```

```
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;
```

```
import com.sportyshoe.repository.ProductRepository;
```

```
@Service
```

```
public class PurchaseItemService {
```

```
    @Autowired
```

```
    ProductRepository pr;
```

```
}
```

```
package com.sportyshoe.repository;
```

```
import java.util.Date;  
import java.util.List;
```

```
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;
```

```
import com.sportyshoe.bean.Purchase;
```

```
@Repository
```

```
public interface PurchaseRepository extends JpaRepository<Purchase, Integer>{
```

```
    public List<Purchase> findByDate(Date date);
```

```
    public List<Purchase> findByEmailId(String emailId);
```

```
}
```

```

package com.sportyshoe.service;

import java.util.ArrayList;
import java.util.Date;
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.Service;

import com.sportyshoe.bean.CartItem;
import com.sportyshoe.bean.Purchase;
import com.sportyshoe.bean.PurchaseItem;
import com.sportyshoe.repository.PurchaseItemRepository;
import com.sportyshoe.repository.PurchaseRepository;

@Service
public class PurchaseService {
    @Autowired
    PurchaseRepository pr;

    @Autowired
    PurchaseItemRepository ps;

    @SuppressWarnings("unchecked")
    public String completePurchase(String userId, HttpServletRequest request) {
        HttpSession session = request.getSession();
        Date date = new Date();

        List<CartItem> cartItems = new ArrayList<CartItem>();
        if (session.getAttribute("cartItems") != null)
            cartItems = (List<CartItem>) session.getAttribute("cartItems");

        double total = (Double) session.getAttribute("cartTotal");

        Purchase purchase = new Purchase();

        purchase.setDate(date);
        purchase.setEmailId(userId);
        purchase.setTotal(total);
        Purchase completedPurchase = pr.save(purchase);
        int purchaseId = completedPurchase.getId();
        session.setAttribute("email", userId);
        session.setAttribute("purchaseId", purchaseId);
        session.setAttribute("date", date.toString());

        List<PurchaseItem> purchaseDetails = new ArrayList<PurchaseItem>();

        for (CartItem item: cartItems) {

```

```
PurchaseItem pi = new PurchaseItem();
pi.setPurchaseId(purchaseId);
pi.setUserEmail(userId);
pi.setProductId(item.getProductId());
pi.setProductName(item.getProductName());
pi.setRate(item.getRate());
pi.setQty(item.getQty());
pi.setPrice(item.getPrice());
purchaseDetails.add(ps.save(pi));
}
```

```
session.setAttribute("purchaseDetails", purchaseDetails);
session.setAttribute("cartItems", null);
session.setAttribute("cartTotal", 0);
session.setAttribute("purchaseTotal", total);
```

```
return "Purchase Confirmed";
}
```

```
public List<Purchase> findPurchaseByDate(Date date, HttpServletRequest request){
    List<Purchase> purchaseList = new ArrayList<Purchase>();
    purchaseList = pr.findAll();

    return purchaseList;
}
```

```
}
```

```
package com.sportyshoe.bean;
```

```
import java.util.Date;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
```

```
@Entity
@Table(name= "users")
public class User {
    @Id
    private String emailId;
    private String pwd;
    private String name;
    private String address;
    private Date dateAdded;
    private String userRole;

    public String getUserRole() {
        return userRole;
    }
    public void setUserRole(String userRole) {
        this.userRole = userRole;
    }
    public User() {
        super();
        // TODO Auto-generated constructor stub
    }
    public String getEmailId() {
        return emailId;
    }
    public void setEmailId(String emailId) {
        this.emailId = emailId;
    }
    public String getPwd() {
        return pwd;
    }
    public void setPwd(String pwd) {
        this.pwd = pwd;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getAddress() {
        return address;
    }
    public void setAddress(String address) {
        this.address = address;
    }
}
```

```
public Date getDateAdded() {  
    return dateAdded;  
}  
public void setDateAdded(Date dateAdded) {  
    this.dateAdded = dateAdded;  
}
```

```
}
```



```
package com.sportyshoe.repository;
```

```
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;
```

```
import com.sportyshoe.bean.User;
```

```
@Repository
```

```
public interface UserRepository extends JpaRepository<User, String> {
```

```
}
```

```
package com.sportyshoe;

import java.util.Date;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.boot.autoconfigure.domain.EntityScan;
//import org.springframework.boot.autoconfigure.jdbc.DataSourceAutoConfiguration;

@SpringBootApplication(scanBasePackages = "com.sportyshoe")
@EntityScan(basePackages = "com.sportyshoe.bean")
public class WebstoreApplication {

    public static void main(String[] args) {
        SpringApplication.run(WebstoreApplication.class, args);
        Date currentDate = new Date();
        System.out.println("Spring Server Running at : " + currentDate.toString());
    }

}
```