中南大学

Web应用开发技术

实验报告

|  |  |
| --- | --- |
| 姓 名： | 张鹏程 |
| 学 号： | 8209210208 |
| 班 级： | 软件工程2102/2105 |
| 指导老师： | 宋 铁 |
| 完成时间： | 2022.12.29 |
| 分 数： |  |
| 批阅日期： |  |

**实验一 基于MVC用JSP/Servlet实现JPetStore**

1. **遇到的问题、错误及解决方法**
   1. 在实现商品展示功能时，发现网页总是报500错，经CSDN搜索后确认是服务器内部运行出错。发现报错信息为MethodNotFoundException后，回到IDEA发现并无报错，故其错误应该为运行时错误。断点调试发现进入许多源代码，理解难度较高。再进行搜索发现报错原因为程序运行时找不到目标方法，但是经过检查，发现类中方法齐备，并无缺少。向老师询问，老师也没有回应。经过半天艰苦卓绝的断点跟踪调试与源代码理解，发现尽管import进的类是自己写的product类，但每次调用的product类并非是自己写的，而是引入jar包中的product类。发现问题为类名重复造成后，尝试解除引入jar包的product类，难度较大，放弃，后将自己写的product类重构为productt类以此与product相区别。后续的Itemm类与Cartt类均遇到此问题，且均用此问题解决。
   2. 在实现验证码功能时，搜索拿到了由给定字符串生成验证码图片的代码，根据该代码实现了验证码的生成。而对于接下来的验证码刷新功能，原定设想如下：用一个a标签封起来一个change字符串，将href属性设置为当前界面URL，但出现输入信息丢失的问题，即：需要重新输入当前页面上的信息。经过讨论，想要改变href指向URL，先经过一个servlet，并将页面上的信息输入进session中。此时老师经过点出不足：这种设想太麻烦了，可将图片标签直接用a标签括起来，href直接填入图片的URL，此处的新问题在于会直接打开图片。经过小组成员及老师的激烈讨论，最终敲定将二维码图片的URL改为原URL加一个时间戳随机出来的整形值作为参数。将更新图片的URL绑定上onclick函数，使用js设定为每点击一次都改变图片URL后的整型值，以区分不同的请求，而每次请求均条用一次生成二维码的servlet，完成二维码的刷新。
   3. 在实现日志功能时，发现使用过滤器与直接在各servlet写调用service的语句差距不大，且多一层跳转会加长服务器反应时间。同时，负责日志功能的同学对于项目并不十分了解，因此选择直接在各servlet写调用service的语句，完成功能的同时加强该同学对项目的了解。
2. **源代码**

详细代码见github公开仓库，URL: <https://github.com/MrBhite/project.git>

下附代码压缩包:



此处只给出部分核心代码:

//AccountService

package CSUSoftWare21.web.projectJPetStore.service;

import CSUSoftWare21.web.projectJPetStore.domain.Account;

import CSUSoftWare21.web.projectJPetStore.persistence.AccountMapper;

import CSUSoftWare21.web.projectJPetStore.persistence.Dao.AccountDao;

import CSUSoftWare21.web.projectJPetStore.persistence.impl.AccountDaoImpl;

public class AccountService {

private AccountDao accountDao;

public AccountService(){

this.accountDao = new AccountDaoImpl() ;

}

public Account getAccount(String username) {

return accountDao.getAccountByUsername(username);

}

public Account getAccount(String username, String password) {

Account account = new Account();

account.setUsername(username);

account.setPassword(password);

return accountDao.getAccountByUsernameAndPassword(account);

}

public void insertAccount(Account account) {

accountDao.insertAccount(account);

accountDao.insertProfile(account);

accountDao.insertSignon(account);

}

public void updateAccount(Account account) {

accountDao.updateAccount(account);

accountDao.updateProfile(account);

if (account.getPassword() != null && account.getPassword().length() > 0) {

accountDao.updateSignon(account);

}

}

}

//CatalogService

package CSUSoftWare21.web.projectJPetStore.service;

import CSUSoftWare21.web.projectJPetStore.domain.Category;

import CSUSoftWare21.web.projectJPetStore.domain.Itemm;

import CSUSoftWare21.web.projectJPetStore.domain.Productt;

import CSUSoftWare21.web.projectJPetStore.persistence.Dao.CategoryDao;

import CSUSoftWare21.web.projectJPetStore.persistence.Dao.ItemDao;

import CSUSoftWare21.web.projectJPetStore.persistence.Dao.ProductDao;

import CSUSoftWare21.web.projectJPetStore.persistence.impl.CategoryDaoImpl;

import CSUSoftWare21.web.projectJPetStore.persistence.impl.ItemDaoImpl;

import CSUSoftWare21.web.projectJPetStore.persistence.impl.ProductDaoImpl;

import java.util.List;

public class CatalogService {

/\*实现功能\*/

private CategoryDao categoryDao;

private ItemDao itemDao;

private ProductDao productDao;

public CatalogService(){

this.categoryDao = new CategoryDaoImpl();

this.itemDao = new ItemDaoImpl();

this.productDao = new ProductDaoImpl();

}

public List<Category> getCategoryList() {

return categoryDao.getCategoryList();

}

public Category getCategory(String categoryId) {

return categoryDao.getCategory(categoryId);

}

public Productt getProduct(String productId) {

return productDao.getProduct(productId);

}

public List<Productt> getProductListByCategory(String categoryId) {

return productDao.getProductListByCategory(categoryId);

}

// TODO enable using more than one keyword

public List<Productt> searchProductList(String keyword) {

return productDao.searchProductList("%" + keyword.toLowerCase() + "%");

}

public List<Itemm> getItemListByProduct(String productId) {

return itemDao.getItemListByProduct(productId);

}

public Itemm getItem(String itemId) {

return itemDao.getItem(itemId);

}

public boolean isItemInStock(String itemId) {

return itemDao.getInventoryQuantity(itemId) > 0;

}

}

//LogService

package CSUSoftWare21.web.projectJPetStore.service;

import CSUSoftWare21.web.projectJPetStore.domain.Log;

import CSUSoftWare21.web.projectJPetStore.persistence.Dao.LogDAO;

import CSUSoftWare21.web.projectJPetStore.persistence.impl.LogDAOImpl;

public class LogService {

Log log;

private LogDAO logDAO;

public LogService(){

log = new Log();

logDAO = new LogDAOImpl();

}

public String logInfo(Object ...s){

return log.logInfomation(s);

}

public void insertLogInfo(String username, String logInfo){

logDAO.insertLog(username, logInfo);

}

}

//OrderService

package CSUSoftWare21.web.projectJPetStore.service;

import CSUSoftWare21.web.projectJPetStore.domain.Itemm;

import CSUSoftWare21.web.projectJPetStore.domain.LineItem;

import CSUSoftWare21.web.projectJPetStore.domain.Order;

import CSUSoftWare21.web.projectJPetStore.domain.Sequence;

import CSUSoftWare21.web.projectJPetStore.persistence.Dao.ItemDao;

import CSUSoftWare21.web.projectJPetStore.persistence.Dao.LineItemDao;

import CSUSoftWare21.web.projectJPetStore.persistence.Dao.OrderDao;

import CSUSoftWare21.web.projectJPetStore.persistence.Dao.SequenceDao;

import CSUSoftWare21.web.projectJPetStore.persistence.impl.ItemDaoImpl;

import CSUSoftWare21.web.projectJPetStore.persistence.impl.LineItemDaoImpl;

import CSUSoftWare21.web.projectJPetStore.persistence.impl.OrderDaoImpl;

import CSUSoftWare21.web.projectJPetStore.persistence.impl.SequenceDaoImpl;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

public class OrderService {

private ItemDao itemDao;

private OrderDao orderDao;

private SequenceDao sequenceDao;

private LineItemDao lineItemDao;

public OrderService(){

this.itemDao = new ItemDaoImpl();

this.orderDao = new OrderDaoImpl();

this.sequenceDao = new SequenceDaoImpl();

this.lineItemDao = new LineItemDaoImpl();

}

public void insertOrder(Order order) {

order.setOrderId(getNextId("ordernum"));

for (int i = 0; i < order.getLineItems().size(); i++) {

LineItem lineItem = (LineItem) order.getLineItems().get(i);

String itemId = lineItem.getItemId();

Integer increment = lineItem.getQuantity();

Map<String, Object> param = new HashMap<String, Object>(2);

param.put("itemId", itemId);

param.put("increment", increment);

itemDao.updateInventoryQuantity(param);

}

orderDao.insertOrder(order);

orderDao.insertOrderStatus(order);

for (int i = 0; i < order.getLineItems().size(); i++) {

LineItem lineItem = (LineItem) order.getLineItems().get(i);

lineItem.setOrderId(order.getOrderId());

lineItemDao.insertLineItem(lineItem);

}

}

public Order getOrder(int orderId) {

Order order = orderDao.getOrder(orderId);

order.setLineItems(lineItemDao.getLineItemsByOrderId(orderId));

for (int i = 0; i < order.getLineItems().size(); i++) {

LineItem lineItem = (LineItem) order.getLineItems().get(i);

Itemm item = itemDao.getItem(lineItem.getItemId());

item.setQuantity(itemDao.getInventoryQuantity(lineItem.getItemId()));

lineItem.setItem(item);

}

return order;

}

public List<Order> getOrdersByUsername(String username) {

return orderDao.getOrdersByUsername(username);

}

public int getNextId(String name) {

Sequence sequence = new Sequence(name, -1);

sequence = (Sequence) sequenceDao.getSequence(sequence);

if (sequence == null) {

throw new RuntimeException("Error: A null sequence was returned from the database (could not get next " + name

+ " sequence).");

}

Sequence parameterObject = new Sequence(name, sequence.getNextId() + 1);

sequenceDao.updateSequence(parameterObject);

return sequence.getNextId();

}

}

//CheckCodeSerclet

package CSUSoftWare21.web.projectJPetStore.web.servlet.account;

import CSUSoftWare21.web.projectJPetStore.persistence.impl.CheckCodeUtil;

import javax.servlet.ServletException;

import javax.servlet.ServletOutputStream;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import java.io.IOException;

public class CheckCodeServlet extends HttpServlet {

@Override

protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

// 生成验证码

//resp.setContentType("image/jpeg");

ServletOutputStream os = resp.getOutputStream();

String checkCode = CheckCodeUtil.outputVerifyImage(100, 50, os, 4);

// 存入Session

HttpSession session = req.getSession();

session.setAttribute("checkCode",checkCode);

}

@Override

protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

this.doGet(req,resp);

}

}

//EditAccountServlet

package CSUSoftWare21.web.projectJPetStore.web.servlet.account;

import CSUSoftWare21.web.projectJPetStore.domain.Account;

import CSUSoftWare21.web.projectJPetStore.service.AccountService;

import CSUSoftWare21.web.projectJPetStore.service.LogService;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import java.io.IOException;

public class EditAccountServlet extends HttpServlet {

private static final String MY\_ACCOUNT\_FORM = "/WEB-INF/jsp/account/editAccount.jsp";

AccountService accountService;

public EditAccountServlet(){

this.accountService= new AccountService();

}

@Override

protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

if(req.getParameter("password")==null||req.getParameter("password").equals("")){

req.setAttribute("errorMsg","Blank Password Detected");

req.getRequestDispatcher(MY\_ACCOUNT\_FORM).forward(req,resp);

}else if(!req.getParameter("password").equals(req.getParameter("repeatedPassword"))){

req.setAttribute("errorMsg","The Two Inputs Are Inconsistent");

req.getRequestDispatcher(MY\_ACCOUNT\_FORM).forward(req,resp);

}else {

Account account = (Account) req.getSession().getAttribute("account");

account.setPassword(req.getParameter("password"));

account.setFirstName(req.getParameter("account.firstName"));

account.setLastName(req.getParameter("account.lastName"));

account.setEmail(req.getParameter("account.email"));

account.setPhone(req.getParameter("account.phone"));

account.setAddress1(req.getParameter("account.address1"));

account.setAddress2(req.getParameter("account.address2"));

account.setCity(req.getParameter("account.city"));

account.setState(req.getParameter("account.state"));

account.setZip(req.getParameter("account.zip"));

account.setCountry(req.getParameter("account.country"));

account.setLanguagePreference(req.getParameter("account.languagePreference"));

account.setFavouriteCategoryId(req.getParameter("account.favouriteCategoryId"));

if(req.getParameter("account.listOption")!=null){

account.setListOption(true);

}

if(req.getParameter("account.bannerOption")!=null){

account.setBannerOption(true);

}

accountService.updateAccount(account);

if(account != null){

HttpServletRequest httpRequest= req;

String strBackUrl = "http://" + req.getServerName() + ":" + req.getServerPort()

+ httpRequest.getContextPath() + httpRequest.getServletPath() + "?" + (httpRequest.getQueryString());

LogService logService = new LogService();

String logInfo = logService.logInfo(" ") + strBackUrl + " Jump to the Edit Account Information page";

logService.insertLogInfo(account.getUsername(), logInfo);

}

req.getRequestDispatcher(MY\_ACCOUNT\_FORM).forward(req,resp);

}

}

}

//MyAccountFormServlet

package CSUSoftWare21.web.projectJPetStore.web.servlet.account;

import CSUSoftWare21.web.projectJPetStore.domain.Account;

import CSUSoftWare21.web.projectJPetStore.service.LogService;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import java.io.IOException;

public class MyAccountFormServlet extends HttpServlet {

private static final String MY\_ACCOUNT\_FORM = "/WEB-INF/jsp/account/editAccount.jsp";

@Override

protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

Account account = (Account) req.getSession().getAttribute("account");

if(account != null){

HttpServletRequest httpRequest= req;

String strBackUrl = "http://" + req.getServerName() + ":" + req.getServerPort()

+ httpRequest.getContextPath() + httpRequest.getServletPath() + "?" + (httpRequest.getQueryString());

LogService logService = new LogService();

String logInfo = logService.logInfo(" ") + strBackUrl + " Jump to the Edit Account Information page";

logService.insertLogInfo(account.getUsername(), logInfo);

}

req.getRequestDispatcher(MY\_ACCOUNT\_FORM).forward(req,resp);

}

}

//NewAccountFormServlet

package CSUSoftWare21.web.projectJPetStore.web.servlet.account;

import CSUSoftWare21.web.projectJPetStore.domain.Account;

import CSUSoftWare21.web.projectJPetStore.service.LogService;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import java.io.IOException;

public class NewAccountFormServlet extends HttpServlet {

private static final String NEW\_ACCOUNT\_FORM = "/WEB-INF/jsp/account/newAccount.jsp";

@Override

protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

String num = req.getParameter("num");

Account account = (Account) req.getSession().getAttribute("account");

if(account != null){

HttpServletRequest httpRequest= req;

String strBackUrl = "http://" + req.getServerName() + ":" + req.getServerPort()

+ httpRequest.getContextPath() + httpRequest.getServletPath() + "?" + (httpRequest.getQueryString());

LogService logService = new LogService();

String logInfo = logService.logInfo(" ") + strBackUrl + " Jump to the Register New Account screen";

logService.insertLogInfo(account.getUsername(), logInfo);

}

req.getRequestDispatcher(NEW\_ACCOUNT\_FORM).forward(req,resp);

}

}

//NewAccountServlet

package CSUSoftWare21.web.projectJPetStore.web.servlet.account;

import CSUSoftWare21.web.projectJPetStore.domain.Account;

import CSUSoftWare21.web.projectJPetStore.service.AccountService;

import CSUSoftWare21.web.projectJPetStore.service.LogService;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import java.io.IOException;

public class NewAccountServlet extends HttpServlet {

private static final String MAIN\_FORM = "/WEB-INF/jsp/catalog/main.jsp";

private static final String NEW\_ACCOUNT\_FORM = "/WEB-INF/jsp/account/newAccount.jsp";

AccountService accountService;

public NewAccountServlet(){

accountService = new AccountService();

}

@Override

protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

HttpSession session = req.getSession();

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

System.out.println(checkCode);

String userName = req.getParameter("username");

String password = req.getParameter("password");

String checkcode= req.getParameter("checkCode");

if(userName==null||userName.equals("")){

req.setAttribute("errorMsg","Blank Username Detected");

req.getRequestDispatcher(NEW\_ACCOUNT\_FORM).forward(req,resp);

}else if(password==null||password.equals("")){

req.setAttribute("errorMsg","Blank Password Detected");

req.getRequestDispatcher(NEW\_ACCOUNT\_FORM).forward(req,resp);

}else if(checkcode.equals(checkCode)){

Account account = new Account();

account.setUsername(req.getParameter("username"));

account.setPassword(req.getParameter("password"));

account.setFirstName(req.getParameter("account.firstName"));

account.setLastName(req.getParameter("account.lastName"));

account.setEmail(req.getParameter("account.email"));

account.setPhone(req.getParameter("account.phone"));

account.setAddress1(req.getParameter("account.address1"));

account.setAddress2(req.getParameter("account.address2"));

account.setCity(req.getParameter("account.city"));

account.setState(req.getParameter("account.state"));

account.setZip(req.getParameter("account.zip"));

account.setCountry(req.getParameter("account.country"));

account.setLanguagePreference(req.getParameter("account.languagePreference"));

account.setFavouriteCategoryId(req.getParameter("account.favouriteCategoryId"));

if(req.getParameter("account.listOption")!=null){

account.setListOption(true);

}

if(req.getParameter("account.bannerOption")!=null){

account.setBannerOption(true);

}

accountService.insertAccount(account);

req.getSession().setAttribute("account",account);

if(account != null){

HttpServletRequest httpRequest= req;

String strBackUrl = "http://" + req.getServerName() + ":" + req.getServerPort()

+ httpRequest.getContextPath() + httpRequest.getServletPath() + "?" + (httpRequest.getQueryString());

LogService logService = new LogService();

String logInfo = logService.logInfo(" ") + strBackUrl + " Register a new account";

logService.insertLogInfo(account.getUsername(), logInfo);

}

req.getRequestDispatcher(MAIN\_FORM).forward(req,resp);

}

else{

req.setAttribute("errorMsg","Wrong CheckCode");

req.getRequestDispatcher(NEW\_ACCOUNT\_FORM).forward(req,resp);

}

}

}

//SignOnFormServlet

package CSUSoftWare21.web.projectJPetStore.web.servlet.account;

import CSUSoftWare21.web.projectJPetStore.domain.Account;

import CSUSoftWare21.web.projectJPetStore.service.LogService;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import java.io.IOException;

public class SignOnFormServlet extends HttpServlet {

private static final String SIGN\_ON\_FORM = "/WEB-INF/jsp/account/signOn.jsp";

@Override

protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

HttpSession session = req.getSession();

req.getRequestDispatcher(SIGN\_ON\_FORM).forward(req,resp);

}

}

//SignOnServlet

package CSUSoftWare21.web.projectJPetStore.web.servlet.account;

import CSUSoftWare21.web.projectJPetStore.domain.Account;

import CSUSoftWare21.web.projectJPetStore.service.AccountService;

import CSUSoftWare21.web.projectJPetStore.service.LogService;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import java.io.IOException;

public class SignOnServlet extends HttpServlet {

private static final String MAIN\_FORM = "/WEB-INF/jsp/catalog/main.jsp";

private static final String SIGN\_ON\_FORM = "/WEB-INF/jsp/account/signOn.jsp";

private AccountService accountService;

public SignOnServlet(){

this.accountService = new AccountService();

}

@Override

protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

HttpSession session = req.getSession();

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

String userName = req.getParameter("username");

String password = req.getParameter("password");

String checkcode= req.getParameter("checkCode");

Account account = accountService.getAccount(userName,password);

if(account != null){

HttpServletRequest httpRequest= req;

String strBackUrl = "http://" + req.getServerName() + ":" + req.getServerPort()

+ httpRequest.getContextPath() + httpRequest.getServletPath() + "?" + (httpRequest.getQueryString());

LogService logService = new LogService();

String logInfo = logService.logInfo(" ") + strBackUrl + " User login";

logService.insertLogInfo(account.getUsername(), logInfo);

}

if(userName==null||userName.equals("")||password==null||password.equals("")){

req.setAttribute("errorMsg","Blank Input Detected");

req.getRequestDispatcher(SIGN\_ON\_FORM).forward(req,resp);

}else if(account == null){

req.setAttribute("errorMsg","Wrong Username Or Password");

req.getRequestDispatcher(SIGN\_ON\_FORM).forward(req,resp);

}else if(!checkcode.equals(checkCode)){

req.setAttribute("errorMsg","Wrong CheckCode");

req.getRequestDispatcher(SIGN\_ON\_FORM).forward(req,resp);

}else {

req.getSession().setAttribute("account",account);

req.getRequestDispatcher(MAIN\_FORM).forward(req,resp);

}

}

}

//SignOutServlet

package CSUSoftWare21.web.projectJPetStore.web.servlet.account;

import CSUSoftWare21.web.projectJPetStore.domain.Account;

import CSUSoftWare21.web.projectJPetStore.service.LogService;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import java.io.IOException;

public class SignOutServlet extends HttpServlet {

private static final String MAIN\_FORM = "/WEB-INF/jsp/catalog/main.jsp";

private Account account;

@Override

protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

HttpSession session = req.getSession();

account = (Account)session.getAttribute("account");

if(account != null){

HttpServletRequest httpRequest= req;

String strBackUrl = "http://" + req.getServerName() + ":" + req.getServerPort()

+ httpRequest.getContextPath() + httpRequest.getServletPath() + "?" + (httpRequest.getQueryString());

LogService logService = new LogService();

String logInfo = logService.logInfo(" ") + strBackUrl + " Log out to return to the main interface";

logService.insertLogInfo(account.getUsername(), logInfo);

}

req.getSession().removeAttribute("account");

req.getRequestDispatcher(MAIN\_FORM).forward(req,resp);

}

}

//web.xml

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

<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"

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

xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd"

version="4.0">

<servlet>

<servlet-name>MainForm</servlet-name>

<servlet-class>CSUSoftWare21.web.projectJPetStore.web.servlet.catalog.MainFormServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>MainForm</servlet-name>

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

</servlet-mapping>

<servlet>

<servlet-name>CheckCode</servlet-name>

<servlet-class>CSUSoftWare21.web.projectJPetStore.web.servlet.account.CheckCodeServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>CheckCode</servlet-name>

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

</servlet-mapping>

<servlet>

<servlet-name>CategoryForm</servlet-name>

<servlet-class>CSUSoftWare21.web.projectJPetStore.web.servlet.catalog.CategoryFormServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>CategoryForm</servlet-name>

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

</servlet-mapping>

<servlet>

<servlet-name>ProductForm</servlet-name>

<servlet-class>CSUSoftWare21.web.projectJPetStore.web.servlet.catalog.ProductFormServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>ProductForm</servlet-name>

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

</servlet-mapping>

<servlet>

<servlet-name>ItemForm</servlet-name>

<servlet-class>CSUSoftWare21.web.projectJPetStore.web.servlet.catalog.ItemFormServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>ItemForm</servlet-name>

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

</servlet-mapping>

<servlet>

<servlet-name>SearchFormServlet</servlet-name>

<servlet-class>CSUSoftWare21.web.projectJPetStore.web.servlet.catalog.SearchFormServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>SearchFormServlet</servlet-name>

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

</servlet-mapping>

<servlet>

<servlet-name>CartFormServlet</servlet-name>

<servlet-class>CSUSoftWare21.web.projectJPetStore.web.servlet.cart.CartFormServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>CartFormServlet</servlet-name>

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

</servlet-mapping>

<servlet>

<servlet-name>AddItemToCartForm</servlet-name>

<servlet-class>CSUSoftWare21.web.projectJPetStore.web.servlet.cart.AddItemToCartServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>AddItemToCartForm</servlet-name>

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

</servlet-mapping>

<servlet>

<servlet-name>UpdateCartServlet</servlet-name>

<servlet-class>CSUSoftWare21.web.projectJPetStore.web.servlet.cart.UpdateCartServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>UpdateCartServlet</servlet-name>

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

</servlet-mapping>

<servlet>

<servlet-name>RemoveCartItemServlet</servlet-name>

<servlet-class>CSUSoftWare21.web.projectJPetStore.web.servlet.cart.RemoveCartItemServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>RemoveCartItemServlet</servlet-name>

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

</servlet-mapping>

<servlet>

<servlet-name>CheckOutServlet</servlet-name>

<servlet-class>CSUSoftWare21.web.projectJPetStore.web.servlet.cart.CheckOutServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>CheckOutServlet</servlet-name>

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

</servlet-mapping>

<servlet>

<servlet-name>SignOnFormServlet</servlet-name>

<servlet-class>CSUSoftWare21.web.projectJPetStore.web.servlet.account.SignOnFormServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>SignOnFormServlet</servlet-name>

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

</servlet-mapping>

<servlet>

<servlet-name>SignOnServlet</servlet-name>

<servlet-class>CSUSoftWare21.web.projectJPetStore.web.servlet.account.SignOnServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>SignOnServlet</servlet-name>

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

</servlet-mapping>

<servlet>

<servlet-name>MyAccountFormServlet</servlet-name>

<servlet-class>CSUSoftWare21.web.projectJPetStore.web.servlet.account.MyAccountFormServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>MyAccountFormServlet</servlet-name>

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

</servlet-mapping>

<servlet>

<servlet-name>EditAccountServlet</servlet-name>

<servlet-class>CSUSoftWare21.web.projectJPetStore.web.servlet.account.EditAccountServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>EditAccountServlet</servlet-name>

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

</servlet-mapping>

<servlet>

<servlet-name>NewAccountFormServlet</servlet-name>

<servlet-class>CSUSoftWare21.web.projectJPetStore.web.servlet.account.NewAccountFormServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>NewAccountFormServlet</servlet-name>

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

</servlet-mapping>

<servlet>

<servlet-name>NewAccountServlet</servlet-name>

<servlet-class>CSUSoftWare21.web.projectJPetStore.web.servlet.account.NewAccountServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>NewAccountServlet</servlet-name>

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

</servlet-mapping>

<servlet>

<servlet-name>SignOutServlet</servlet-name>

<servlet-class>CSUSoftWare21.web.projectJPetStore.web.servlet.account.SignOutServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>SignOutServlet</servlet-name>

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

</servlet-mapping>

<servlet>

<servlet-name>NewOrderFormServlet</servlet-name>

<servlet-class>CSUSoftWare21.web.projectJPetStore.web.servlet.order.NewOrderFormServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>NewOrderFormServlet</servlet-name>

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

</servlet-mapping>

<servlet>

<servlet-name>NewOrderServlet</servlet-name>

<servlet-class>CSUSoftWare21.web.projectJPetStore.web.servlet.order.NewOrderServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>NewOrderServlet</servlet-name>

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

</servlet-mapping>

<servlet>

<servlet-name>ConfirmOrderFormServlet</servlet-name>

<servlet-class>CSUSoftWare21.web.projectJPetStore.web.servlet.order.ConfirmOrderFormServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>ConfirmOrderFormServlet</servlet-name>

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

</servlet-mapping>

<servlet>

<servlet-name>ViewOrderFormServlet</servlet-name>

<servlet-class>CSUSoftWare21.web.projectJPetStore.web.servlet.order.ViewOrderFormServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>ViewOrderFormServlet</servlet-name>

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

</servlet-mapping>

<servlet>

<servlet-name>ListOrdersFormServlet</servlet-name>

<servlet-class>CSUSoftWare21.web.projectJPetStore.web.servlet.order.ListOrdersFormServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>ListOrdersFormServlet</servlet-name>

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

</servlet-mapping>

</web-app>

1. **小组成员分工**

组长：8209210208张鹏程 基础功能的实现

组员：8202210228吴昭棻 二维码相关功能的实现

组员：8204210706董泽 系统日志相关功能的实现

**实验二 使用AJAX和jQuery改善用户体验**

1. **遇到的问题、错误及解决方法**
   1. 在学习过程中，js由于其不易报错而只是没有效果的特性十分迷惑人，调试时设置断点也只有“进得去”与“进不去”两种情况，要搞清楚是哪里有问题，问题是什么类型十分困难。比如说在练习添加监听函数时，试图对text框增加onblur函数，可是调试时函数添加语句能运行，但onblur里面的函数体无论如何进不去。报错也不报错，断点调试也进不去，很恼人。最后在CSDN上搜寻并间断性讨论了一天，才发现onblur函数是表单绑定函数，若控件不在表单内则无法绑定onblur。诸如此类的问题极多，且解决起来都不比这个问题简单。只能通过不断的搜索、尝试、讨论去寻找答案。
   2. 在对网页的代码进行调试时，经常发现有时在代码上给出的改动并不能在网页上体现出来，而往往在重构了文件名后就可以在网页上显示出来，起初一直是改一段功能就重构一次名字。最后忍无可忍了花了一下午时间百度，最后发现是网络缓冲没有禁用。禁用了就解决了问题
   3. 在实现ajax时有两种方法，一是使用XMLHttpRequest并借由较为底层底层的send等函数实现，二是使用jQuery实现。为了弄明白二者的具体区别，本次实验两种方式均采用了：实现悬浮窗的时候使用了XMLHttpRequest实现，其他的ajax功能则由jQuery实现。在实现过程中发现：使用XMLHttpRequest虽然代码量较高，但函数耦合性低，高度模块化，程序阅读易懂；使用jQuery虽较为方便，但代码耦合性高，不易读懂。而在实际效果方面，二者虽都能达到目标，但一定会在性能上有所差异，可惜由于实现的功能并不复杂，且当今电脑的性能均较好，并未能够在可以测获的范围内得到二者在性能上的差距。现已准备在后续的学习中继续关注此点。
   4. 显然，UI设计和UI实现是两码事，程序员负责把前端界面的相关功能实现，而至于界面是怎么布局，好不好看，如何设计出来的则应该是美工的活计。显而易见的是，虽然我们小组内有着整个计算机学院最为稀少的资源——活生生的女同学，但总体审美依然捉襟见肘。各自设计了几版布局，仔细比较还没有原版的布局好看，最后迫不得已选择了原版做模板，修改配色和按钮样式。
   5. git的使用无疑极大的便利了合作开发，但对git的不熟练使用则无疑是灾难性的，尤其是误打误撞各种合并、变基、提交、拉取、回滚、撤销把版本彻底打乱之后，最后迫不得已又从头开始(即从上次实验一的版本开始)又重新进行开发，浪费了不少时间。下次在使用git前一定要注意对组员的相关方面培训。
2. **源代码**

详细代码见github公开仓库，URL: <https://github.com/MrBhite/project.git>

下附代码压缩包:



此处只给出部分核心代码(js/css):

JPetStore.css

body {

margin: 0ex 5ex 0ex 6ex;

padding: 0ex;

font-family: helvetica, tahoma, arial, verdana, sans-serif;

font-size: 2ex;

color: #333;

background-color: #ffffff;

}

pre {

font-family: "Courier New", Courier, mono;

font-style: normal;

/\*中南蓝#02468e;\*/

background-color: #FFFFFF;

white-space: pre

}

h1 {

margin: 1ex 0ex 1ex 0ex;

padding: 0ex;

line-height: 3ex;

font-weight: 900;

color: #666;

}

h2 {

margin: 2ex 0ex 1ex 0ex;

padding: 0ex;

line-height: 2ex;

font-weight: 700;

color: #444;

}

h3 {

margin: 1ex 0ex 1ex 0ex;

padding: 0ex;

line-height: 1.6ex;

font-weight: 700;

color: #222;

}

p {

font-family: helvetica, tahoma, arial, verdana, sans-serif;

margin: 0ex 0ex 0ex 0ex;

padding: 2ex;

}

img {

border: 0;

}

li {

font-family: helvetica, tahoma, arial, verdana, sans-serif;

margin: 0ex 0ex 0ex 0ex;

padding: 0ex;

}

table {

border-width: 0;

empty-cells: show;

}

td, th {

empty-cells: show;

padding: .3ex .3ex;

vertical-align: top;

text-align: left;

border-width: 0;

border-spacing: 0;

background-color: #ececec

}

th {

font-weight: bold;

background-color: #e2e2e2;

}

a, a:visited, a:link {

color: #039;

text-decoration: none;

font-family: helvetica, tahoma, arial, verdana, sans-serif;

}

a:hover {

color: #69f;

}

a.Button, a.Button:link, a.Button:visited {

padding: .3ex;

color: #ffffff;

background-color: #02468e;

text-decoration: none;

font-family: helvetica, tahoma, arial, verdana, sans-serif;

font-size: 1.5ex;

}

a.Button:hover {

color: #000;

background-color: #54c07a;

}

#Logo {

width: 33%;

height: 9ex;

margin: 0ex 0ex 0ex 0ex;

padding: 0ex 0ex 0ex 0ex;

border-width: 0ex 0ex .3ex 0px;

border-style: solid;

border-color: #ccc;

float: left;

background-color: #02468e;

color: #fff;

line-height: 9ex;

voice-family: "\"}\"";

voice-family: inherit;

height: 9ex;

}

#mainPic{

height:9ex;

}

body>#Logo {

height: 9ex;

}

#Menu {

width: 33%;

height: 9ex;

margin: 0ex 0ex 0ex 0ex;

padding: 0ex 0ex 0ex 0ex;

border-width: 0ex 0ex .3ex 0px;

border-style: solid;

border-color: #ccc;

float: left;

background-color: #02468e;

color: #fdfdfd;

text-decoration: none;

font-family: helvetica, tahoma, arial, verdana, sans-serif;

text-align: center;

line-height: 9ex;

voice-family: "\"}\"";

voice-family: inherit;

height: 9ex;

}

#Menu, #Menu a, #Menu a:link, #Menu a:visited, #Menu a:hover {

color: #ffffff;

text-decoration: none;

font-family: helvetica, tahoma, arial, verdana, sans-serif;

}

body>#Menu {

height: 9ex;

}

{

}

#Search {

width: 33%;

height: 9ex;

margin: 0ex 0ex 0ex 0ex;

padding: 0ex 0ex 0ex 0ex;

border-width: 0ex 0ex .3ex 0px;

border-style: solid;

border-color: #ccc;

float: left;

text-align: center;

background-color: #02468e;

color: #eaac00;

line-height: 9ex;

voice-family: "\"}\"";

voice-family: inherit;

height: 9ex;

}

body>#Search {

height: 9ex;

}

#Search input {

border-width: .1ex .1ex .1ex .1ex;

border-style: solid;

border-color: #aaa;

background-color: #ffffff;

color: #447cff;

}

#QuickLinks {

text-align: center;

background-color: #FFF;

width: 99%;

}

#PoweredBy {

width: 30%;

height: 9ex;

margin: 0ex 0ex 0ex 0ex;

padding: 0ex 0ex 0ex 0ex;

border-width: .3ex 0ex .3ex 0px;

border-style: solid;

border-color: #ccc;

float: left;

background-color: #02468e;

color: #fff;

line-height: 9ex;

voice-family: "\"}\"";

voice-family: inherit;

height: 9ex;

}

#PoweredBy a{

color: #ffffff;

}

#PoweredBy a:hover{

color: #b3deff;

}

body>#PoweredBy {

height: 9ex;

}

#Banner {

width: 69%;

height: 9ex;

margin: 0ex 0ex 0ex 0ex;

padding: 0ex 0ex 0ex 0ex;

border-width: .3ex 0ex .3ex 0px;

border-style: solid;

border-color: #ccc;

float: left;

background-color: #02468e;

color: #fff;

line-height: 9ex;

voice-family: "\"}\"";

voice-family: inherit;

height: 9ex;

}

body>#Banner {

height: 9ex;

}

#Content {

margin: 0;

padding: 0ex 0ex 0ex 0ex;

width: 99%;

min-height:550px;

color: #333;

background-color: #FFF;

border-width: 0 2px 0 2px;

border-style: none solid none solid;

border-color: rgba(70,125,255,0.47);

}

#Separator {

clear:both;

margin: 0;

height:0;

}

#Main {

margin: 0;

padding: 1ex;

color: #333;

background-color: #FFF;

border-width: 1ex 0ex 2ex 0px;

border-style: solid;

border-color: #fff;

}

#Sidebar {

float: left;

background:inherit;

width: 30%;

}

#MainImage {

float: left;

background:inherit;

text-align:center;

width: 50%;

}

#Catalog {

padding: 1ex;

background:inherit;

text-align:center;

}

#Catalog input[type="submit"]{

padding: .3ex;

color: #fff;

background-color: #02468e;

text-decoration: none;

font-family: helvetica, tahoma, arial, verdana, sans-serif;

font-size: 1.5ex;

border-width:0;

}

#Catalog input[type="submit"]:hover {

color: #ffffff;

background-color: #3189ff;

cursor:pointer;

}

#Catalog table{

margin-left:auto;

margin-right:auto;

border-spacing: 0 5px;

}

#Catalog .orderTable td{

}

#Catalog #secondTable td{

display: none;

}

#Catalog #thirdTable td{

display: none;

}

#Catalog .second {

}

#Catalog .press{

padding: 2px 0 0 310px;

}

.open{

}

#BackLink{

padding: 1ex;

float: right;

border-width: .1ex 0ex .1ex 0px;

border-style: solid;

border-color: #000;

}

#Cart{

width: 69.99%;

float: left;

background-color:#fff;

position: relative;

left: 195px;

}

#MyList{

width: 30%;

float: left;

background-color:#ccc;

text-align:left;

}

#productAutoComplete {

background: #ffffff;

position: relative;

top: -46px;

left: 126px;

width: 250px;

border:solid;

border-color: #02468e;

border-radius: 5px;

color: #447cff;

z-index: 999;

display: none;

}

#inform hr{

border:1px;

width: 325px;

margin-bottom: 0px;

}

.productAutoComplete{

margin: 0;

padding: 10px;

font-size: 10px;

list-style-type: none;

text-align: left;

}

.productAutoComplete li{

line-height: 30px;

border-bottom: solid;

border-bottom-color: #447cff;

border-bottom-width: 1px;

}

//productAuto.js

$(function (){

$('#keyword').on('keyup',function (){

var keyword=$(this).val();

if(keyword!==''&&keyword!==null&&keyword.length!==0){

$.ajax({

type:'GET',

url :'http://localhost:8088/project/productAuto?keyword='+keyword,

success:function (data){

console.log(data);

var productListHTML='';

for(var i=0;i<data.length;i++){

productListHTML+='<li class=\"productAutoItem\" data-productId="';

productListHTML+=data[i].productId;

productListHTML+='">';

productListHTML+=data[i].categoryId;

productListHTML+=':';

productListHTML+=data[i].name;

productListHTML+='</li>';

}

$('#productAutoList').html(productListHTML);

$('#productAutoComplete').show();

},

error:function (errorMsg) {

console.log(errorMsg);

}

});

}else{

$('#productAutoComplete').hide();

}

});

// $('.productAutoItem').on('click',function (){

//

// })

$(document).on('click','.productAutoItem',function (){ //动态绑定

var productId=$(this).data('productid');

$('#productAutoComplete').hide();

$('#keyword').val('');

window.location.href='http://localhost:8088/project/product?productId='+productId;

});

//注意区分mouseleave和mouseout的区别

$('#productAutoComplete').on('mouseleave',function (){

$(this).hide();

$('#keyword').val('');

})

});

//cart.js

var xhr = ''

var cart = document.getElementById('cartTable');

cart.addEventListener('blur', function (e) {

var cartItem = e.target.parent,

itemID = cartItem[0],

quantity = cartItem[4];

console.log(itemID);

console.log(quantity);

/\*if(e.target.matches('text')){

var cartItem = e.target.parent,

itemID = cartItem[0],

quantity = cartItem[4];

console(itemID);

console(quantity);

}\*/

} )

//cartChange.js

function remove() {

var el = document.getElementById("remove");

var el2 = el.parentNode.parentNode;

el.parentNode.parentNode.remove();

//var el3 = document.getElementById("lastTR");

var el3 = document.getElementById("lastTR");

el3.before(el2);

}

//this.parentNode.parentNode.parentNode.removeChild(this.parentNode.parentNode);

$(document).ready(function(){

//jQuery调用

$('#remove').on('click',function () {

$(this.parentNode.parentNode).remove();

});

$("tr").css("background-color", "red");

});

//orderPress.js

var opens = document.querySelectorAll('.tbody');

for(var i = 0; i < opens.length; i++){

}

opens.forEach(this.addEventListener('click', function (e){

if(e.target.matches('button')){

var tbody = e.target.parentNode.parentNode.parentNode;

if(window.getComputedStyle(tbody.children[1].children[0]).display==='none'){

for(var i = 1; i<tbody.childElementCount; i++){

tbody.children[i].children[0].style.display='table-cell';

tbody.children[i].children[1].style.display='table-cell';

}

}else{

for(var i = 1; i<tbody.childElementCount; i++){

tbody.children[i].children[0].style.display='none';

tbody.children[i].children[1].style.display='none';

}

}

}

}));

//productDetail.js

//商品展示js

var xhr;

var name;

var inform = document.getElementById('inform');

var images = document.getElementById('mainimages');

images.addEventListener('mouseover', function(e){

showInform(e.target.alt);

})

images.addEventListener('mouseleave',function (){

inform.style.display='none';

})

function createXMLHttpRequest()

{

xhr = new XMLHttpRequest();

}

function showInform(categoryId)

{

name = categoryId;

var url = "categoryShowJs?categoryId="+categoryId;

sendRequest(url);

}

function sendRequest(url)

{

createXMLHttpRequest();

xhr.onreadystatechange = processInform;

xhr.open("GET",url,true);

xhr.send(null);

}

function processInform()

{

if (xhr.readyState ==4)

if(xhr.status ==200)

{

var rep = xhr.responseText;

inform.innerText = rep;

inform.className = name;

inform.style.display = 'block';

}

}

function hiddenInform(event)

{

var informDiv = document.getElementById('inform');

var x = event.clientX;

var y = event.clientY;

var divx1 = informDiv.offsetLeft;

var divy1 = informDiv.offsetTop;

var divx2 = informDiv.offsetLeft+informDiv.offsetWidth;

var divy2 = informDiv.offsetTop+informDiv.offsetHeight;

if(x < divx1 || x > divx2 || y < divy1 || y> divy2)

{

document.getElementById('inform').style.display = 'none';

}

}

//register.js

var xmlHttpRequest;

function createXMLHttpRequest()

{

if (window.XMLHttpRequest) //非IE浏览器

{

xmlHttpRequest = new XMLHttpRequest();

}

else if (window.ActiveObject)//IE6以上版本的IE浏览器

{

xmlHttpRequest = new ActiveObject("Msxml2.XMLHTTP");

}

else //IE6及以下版本IE浏览器

{

xmlHttpRequest = new ActiveObject("Microsoft.XMLHTTP");

}

}

function usernameIsExist() {

//var username = document.NewAccountForm.username.value;

var username = document.getElementById('username').value;

sendRequest("usernameIsExistServlet?username=" + username);

}

function sendRequest(url) {

createXMLHttpRequest();

xmlHttpRequest.open("GET", url, true);

xmlHttpRequest.onreadystatechange = processResponse;

xmlHttpRequest.send(null);

}

function processResponse() {

if (xmlHttpRequest.readyState == 4) {

if (xmlHttpRequest.status == 200) {

var responseInfo = xmlHttpRequest.responseXML.getElementsByTagName("msg")[0].firstChild.data;

var div1 = document.getElementById('usernameMsg');

if (responseInfo == "Exist") {

div1.innerHTML = "<font color='red'>用户名已存在</font>";

} else {

div1.innerHTML = "<font color='green'>用户名可用</font>";

}

}

}

}

//updateCart.js

function createXMLHttpRequest(){

if (window.XMLHttpRequest) //非IE浏览器

{

xmlHttpRequest = new XMLHttpRequest();

}

else if (window.ActiveObject)//IE6以上版本的IE浏览器

{

xmlHttpRequest = new ActiveObject("Msxml2.XMLHTTP");

}

else //IE6及以下版本IE浏览器

{

xmlHttpRequest = new ActiveObject("Microsoft.XMLHTTP");

}

}

function updateCart() {

var quantity = document.getElementById("quantity").value;

sendRequest("updateCartJSServlet?quantity="+ quantity);

}

function sendRequest(url) {

createXMLHttpRequest();

xmlHttpRequest.open("GET", url, true);

xmlHttpRequest.onreadystatechange = processResponse;

xmlHttpRequest.send(null);

}

function processResponse() {

if (xmlHttpRequest.readyState == 4) {

if (xmlHttpRequest.status == 200) {

var resp = xmlHttpRequest.responseText;

var array = resp.split(",");

var quantity = document.getElementById("quantity");

var total = document.getElementById("total");

var subtotal = document.getElementById("subtotal");

quantity.innerText = array[0];

total.innerText = array[1];

subtotal.innerText = array[2];

}

}

}

1. **小组分工**

组长：8209210208张鹏程 悬浮窗、订单多个信息部分功能实现

组员：8202210228吴昭棻 相关UI、样式、界面效果绘制

组员：8204210706董泽 账号管理、查询商品、购物车模块功能实现