浙江工艺大学

Web 开发:

期末大作业——网上购物系统

2019/2020(1)



学生姓名 _	赵水
学生学号 -	201806150329
学生班级	软工 1803
任课教师	吕明琪

一、系统主要功能

1.1 普通用户注册



- ①通过 AJAX 即时验证新用户名是否含有与数据库中相同的用户。
- ②通过 JS 即时验证输入的密码是否符合要求切确认密码是否和密码一致。
- ③若有不满足条件则,则阻止表单提交;若全满足要求,这注册成功转向主页。



1.2 主页:商品概览,查询与详细信息

从数据库中读取所以商品信息,并按照每页10条数据进行分页。



①若用户尚未登录,用户仍可以进行查看、搜索商品功能。



添加到购物车

②若点击添加至购物车,则自动跳转至登录界面。

● 用户信息表单× +	-	Ć	3	×		
← → C ① localhost:8080/Project/cart.do?source=addToCart&id=11						
返回首页						
请輸入用户名和密码						
用户名:						
密码:						
登录						

若已登录,则成功添加



1.3 管理员和普通用户权限

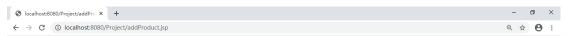


①管理员:可以添加商品、下架商品、修改商品信息、搜索与管理用户信息等。



新增商品信息

I 添加商品:



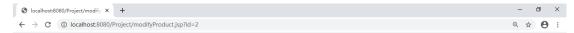
欢迎您,111111 个人空间 管理用户 查看购物车 退出登录

您的位置: 首页>>新增商品信息

商品信息输入

商品名	西瓜
产地	新疆
价格	20
库存	30
	提交

Ⅱ 修改商品:



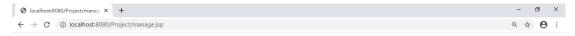
欢迎您,111111 个人空间 管理用户 查看购物车 我的仓库 退出登录

您的位置: 首页>>商品信息编辑

商品信息修改

商品名	香蕉
产地	贵州22
价格	15.0
库存	59
	提交

Ⅲ 管理用户:



欢迎您,111111个人空间 管理用户 查看购物车 我的仓库 退出登录

您的位置: 首页>>管理用户

用户管理 講輸入想搜索的用户 搜索

id	权限	用户名	操作
1	管理员	111111	无法管理管理员用户
4	普通用户	11	删除用户数据

一共有2条数据,共1页。

1

②普通用户(与管理员用户):

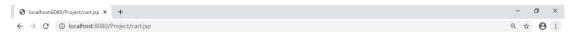


I 个人信息管理:



与注册时功能相似,不同点是当用户名与原名相同时也能够通过修改验证。

Ⅱ 添加至购物车以及管理(改变数量、移除、购买等):



欢迎您,11个人空间 查看购物车 我的仓库 退出登录

您的位置: 首页>>我的购物车

购物车列表

序号	商品名	查看	操作	购物车数量	价格
1	柠檬	查看	<u> 移除出购物车 +1 -1 购买</u>	1	19.0
2	香蕉	查看	<u> 移除出购物车 +1 -1 购买</u>	1	15.0



购买确认

商品名:	柠檬
产地:	新疆
单价:	19.0
库存:	99
数量:	2

总价格: 38.0

确认支付

确认购买后,清除购物车相关数据,且商品库存相应减少,后跳转至仓库界面(即购买记录) 用户也可管理删除记录:



您的位置: 首页>>我的仓库 我的仓库

序号	商品名	查看	单价	购买数量	购买时间	操作
1	柠檬	査看	19.0	2	2020-01-09 13:28:47.0	删除记录

二、系统使用框架

2.1 MVC 框架模式的使用

MVC 全名是 Model View Controller,是模型(model)一视图(view)一控制器 (controller)的缩写,一种软件设计典范,用一种业务逻辑、数据、界面显示分离的方法组织代码,将业务逻辑聚集到一个部件里面,在改进和个性化定制界面及用户交互的同时,不需要重新编写业务逻辑。MVC 被独特的发展起来用于映射传统的输入、处理和输出功能在一个逻辑的图形化用户界面的结构中。

在本系统中,MVC 中的 C-控制器 在其中的体现表现为 java 类中 Servlet 类的使用,UserServlet(用于实现与用户管理有关的功能)、ProductServlet(用于实现商品信息管理的功能)和 CartServlet(用于实现购物车信息管理的功能)。近乎网页中所有的超链接所指的对象都为这些 Servlet 类,由他们将数据写入或读出数据库后再携带特定参数跳转回指定页面,完成业务层面的功能;而 V-显示 则是体现为本系统中的所有 jsp 文件中包括所有的 HTML 元素、JavaScript 脚本和对 Servlet 类传出参数的处理显示,形成友好美观的用户交互界面及数据显示;M-数据模型 则是 JavaBean 类的使用,如商品信息、用户信息等具有一些相同特征的但数值不同的数据组合封装成复杂数据对象,再以这些整体为单位将其交由控制器处理。这样组织整个系统的框架使得整个系统的模块化,不同模块只需负责自己的功能,这样分块条理清晰,且更加方便于后期整体大规模的维护。

2.2 DAO 模式的使用

DAO (Data Access objects 数据存取对象)是指位于业务逻辑和持久化数据之间实现对持久化数据的访问。通俗来讲,就是将数据库操作都封装起来。

数据库操作在上述的 MVC 模式中也属于其中 C-控制器 的范畴,但因其是专门用于与数据库交互的,且数据库操作众多且操作复杂,将其并入 Servlet 类当中会使得其相当的臃肿,于是便将这部分专门用于操作数据库的代码封装成单独的类及其中的类方法,即 DAO 类。这样一来,每当需要进行数据库操作时,只需要通过 DAO 类对象调用数据库操作方法就能完成数据库的操作。DAO 模式的使用,在一定程度上也是对 MVC 模式的继承,将数据库操作的方法单独封装成类,能够使得维护管理效率更高。

在本系统中,一共有 UserDAO (用户信息访问数据库类),CartDAO (购物车信息访问数据库类),ProductDAO (商品信息访问数据库类)和 StorageDAO (仓库信息访问数据库类)四个 DAO 类。将管理不同数据库的类方法划分在不同的 DAO 类当中,使得各版块的分工更加明确,管理效率更高。

三、系统亮点

3.1 搜索功能、分页功能提高了管理效率。

搜索功能:商品信息、用户信息等都添加了搜索框,用户想要搜索某个商品时,在搜索框中输入关键字,就会返还含有这些关键字的商品,使得用户能够更加快捷的索引到想要的商品列表。



分页功能:按照每页 10 条数据的方式排列数据,使得在数据量过大时有限的页面大小的排布更加稳定更加清晰,若用户想浏览靠后的数据也能够快速到达指定位置。



用户信息管理也具有上述功能:



3.2 整齐的布局

该系统中所有的页面都大致分为上中下三个板块,且风格、色调相近却又不完全一致而显得简陋枯燥



欢迎您,11个人空间 查看购物车 我的仓库 退出登录

您的位置: 首页

商品信息一览

请输入想搜索的商品 搜索

序号	商品名	查看	操作	库存	单价
1	香蕉	查看	添加到购物车	59	15.0
2	柠檬	查看	添加到购物车	97	19.0
3	橙子	查看	添加到购物车	89	10.0
4	香蕉	查看	添加到购物车	90	15.0
5	c++从入门到入土	查看	添加到购物车	100	10.0
6	web入门	查看	添加到购物车	222	15.0
7	сссс	查看	添加到购物车	1	1.0
8	dddd	查看	添加到购物车	1	1.0
9	概率论+	查看	添加到购物车	2	15.0
10	概率论++	查看	添加到购物车	3	22.0

一共有11条数据,共2页。

12



您的位置: 首页>>商品信息 商品详细信息

商品名:	香蕉
产地:	贵州22
价格:	15.0
库存:	59

添加到购物车

3.3 友好的提示

注册信息以及修改信息时会有即使弹框提示:



首页、搜索结果页、购物车页、仓库页面数据为空时的提示



3.4 细节的判定

- ①用户注册后即为登录状态。
- ②若商品库存为零或已下架,则自动从商品列表移除(但不从数据库中移除数据,因为用户可以从购买记录查看自己以及购买的商的信息),届时,即使已存在于购物车和商品详情信息中的商品购买链接消失





③若用户购买数量大于库存数量,则交易不成功,库存不会减少,且购物车的数据不会被 清空。

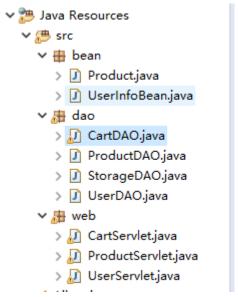
④安全保护机制: 若用户以 URL 以文件和参数的形式访问系统,会因没有通过设置 session 而被强制转回登录界面(原理与"添加至购物车"原理相似,但适用于所有登陆后才能访问的页面)

3.5 数据库资源的清理回收

- ①用户购买商品后购物车列表中的相应商品移除。
- ②管理员删除普通用户数据后普通用户数据库中用户表、购物车表、仓库信息表都全将删除,回收资源。

3.6 代码复用

所有的业务功能的主体全部凝聚为 2 个 JavaBean 类 3 个 Servlet 类 4 个 DAO 类 近乎所有的超链接都指向这三个 Servlet 类当中。所有的超链接都携带了一个参数 "source" 表明超链接的来源和想要实现的功能,通过 if 分支来选取想要调用的代码块执行任务。



```
🔝 index.jsp 🔑 CartServlet.java 🛭
$14 public class CartServlet extends HttpServlet {
        public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletExcept
△15⊝
            doPost(request, response);
16
17
 18
△19⊝
        public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletExcep
            String source = request.getParameter("source");
 21
            if (source.equals("addToCart")) {//添加至购物车
 22
 23
                UserInfoBean user = (UserInfoBean) request.getSession().getAttribute("user");
                if (user == null) {
 24
                    request.setAttribute("msg", "Please login first!");
                    request.getRequestDispatcher("/login.jsp").forward(request, response);
                } else {
 28
                    int proId = Integer.parseInt(request.getParameter("id"));
 29
                    CartDAO cartdao = new CartDAO();
 30
                    cartdao.addCart(proId, user.getId());
                    request.getRequestDispatcher("/cart.jsp").forward(request, response);
 32
                }
            }
 34
            if (source.equals("removeFromCart")) {//移除购物车
 36
                UserInfoBean user = (UserInfoBean) request.getSession().getAttribute("user");
                int proId = Integer.parseInt(request.getParameter("id"));
 38
                CartDAO cartdao = new CartDAO();
                cartdao.removeFromCart(proId, user.getId());
```