

目 录

目 录.....	I
第一章 简介.....	2
1.1 项目内容.....	2
1.2 项目目标.....	2
第二章 功能需求分析.....	2
2.1 登录功能.....	2
2.2 界面拦截功能.....	2
2.3 用户管理功能.....	2
2.4 书籍管理展示功能.....	2
2.5 主界面书籍展示功能.....	3
2.6 购物车功能.....	3
第三章 系统结构设计.....	4
3.1 系统模块结构图.....	4
3.2 登录模块.....	4
3.3 界面拦截模块.....	4
3.4 用户管理模块.....	5
3.5 书籍管理展示模块.....	5
3.6 主界面书籍展示功模块.....	5
3.7 购物车模块.....	5
第四章 系统实现.....	5
4.1 系统实现介绍.....	5
4.1.1 登录模块实现.....	5
4.1.2 界面拦截模块实现.....	8
4.1.3 用户管理模块实现.....	9
4.1.4 书籍管理展示模块实现.....	13
4.1.5 主界面书籍展示功模块实现.....	16
4.1.6 购物车模块实现.....	18
4.2 系统实现的不足.....	20
第五章 总结与展望.....	21

第一章 简介

1.1 项目内容

为了方便书店的书籍访问与管理，实现书店的网络化服务，于是尝试构建一个网络书城项目，对书店用户的需求进行网上的实现，同时实现对书籍的管理

1.2 项目目标

项目目标为实现一个功能较为完备的网上书城项目，需要实现用户的注册登录与删除功能，书籍的管理功能，登录界面的请求拦截功能，购物车的添加与删除功能等主要的常用功能，组合以实现一个简单的网上书城项目

第二章 功能需求分析

2.1 登录功能

用户从主界面进入管理界面需要进行登录验证。需要使用读取用户在表单中填写的数据并与数据库中存储的用户信息进行验证，并且要有验证码功能。

2.2 界面拦截功能

在构建好主界面与登录界面的 HTML 文件并部署后发现用户可以直接更改访问地址在未登录的情况下进入书籍的管理和购买界面，这是不符合规范的，所以需要请求进行拦截

2.3 用户的管理功能

用户需要进行注册以及注销功能

2.4 书籍管理展示功能

书籍的展示：

在登入成功后进入管理主界面，从数据库中读取已经存入的书籍信息并生成

表单进行展示，并设置修改删除等按钮便于书籍的管理功能调用。

书籍的管理：

书籍的管理主要包括增添（包括基本信息和封面图片），删除，修改等功能，

2.5 主界面书籍展示功能

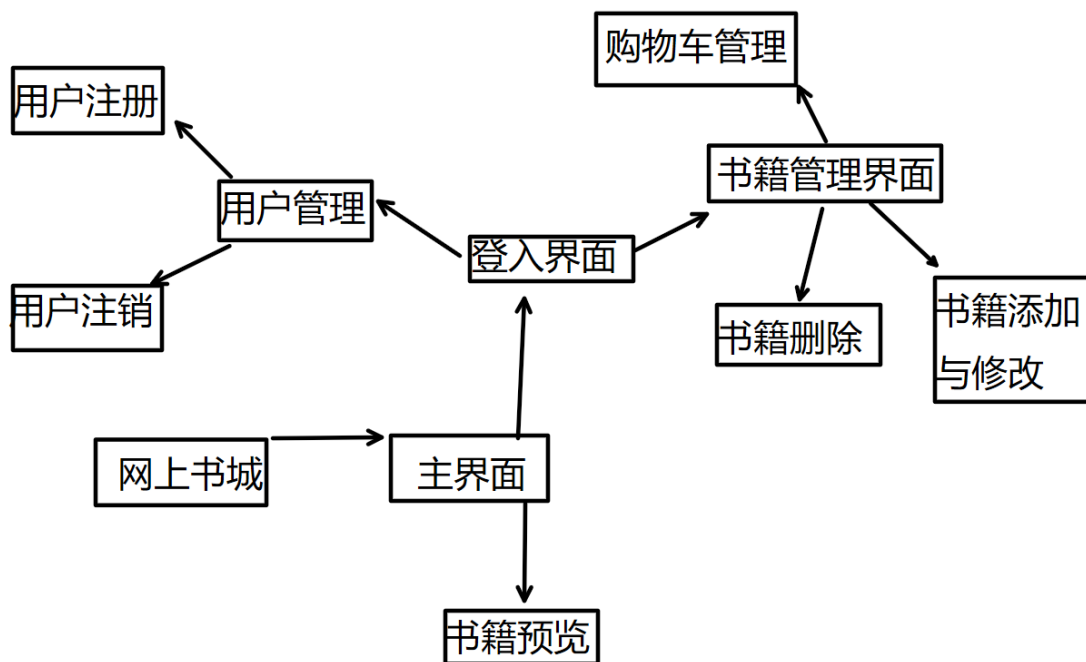
在后台数据库存储的书籍可以在未登入时的主界面进行展示

2.6 购物车功能

购物车功能需要一张新表来存放添加的书籍，并可以从购物车中移除。

第三章 系统结构设计

3.1 系统模块结构图



3.2 登录模块

登录书店需要用户输入用户名，密码并进行验证码验证，所以需要构造一个与数据库的连接以对用户数据进行查找请求，接收用户在登录界面的 form 表单中的数据并且验证验证码，如果验证通过则加上登入成功的 session 标记（使用枚举常量进行设置）

3.3 界面拦截模块

由于直接部署会使用户可以跳过登录验证直接访问书籍管理或购物车等功能界面，所以需要对未登录的用户进行拦截限制，设置过滤器将允许访问的界面以及功能放出，实现对管理系统的保护

3.4 用户管理模块

用户需要进行注册与注销功能，需要构建并调用基础数据库连接类，注册时将用户传递的用户名与密码按分别写入数据库的 `user` 与 `pass` 列内，用户注销时使用 `user` 列（用户名）进行检索并执行删除功能。

3.5 书籍管理展示模块

本模块分为展示模块与管理模块

展示模块为从数据库中按 `id` 循环读取数据并以 `table` 形式表格输出

管理模块为书籍的管理功能

添加书籍通过接收用户通过表单上传的书籍数据以及图片地址，通过基本链接模块完成传入数据库的操作。

3.6 主页书籍管理展示模块

本模块为在主页面将书籍数据打印出，实现预览效果，使用前后端分离的做法，将预览数据通过 `json` 格式传向前端，再通过 `js` 文件进行操作并生成预览

3.7 购物车模块

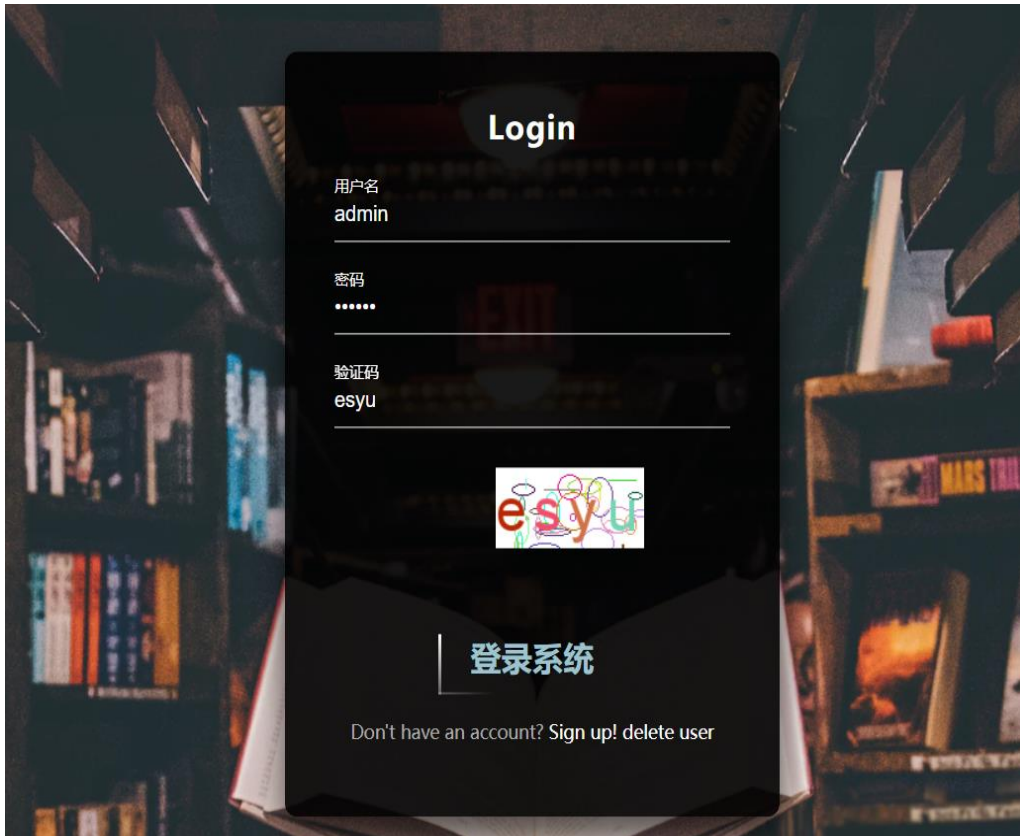
购物车要实现添加与商品展示，需要在数据库中设置新表专门存放购物车数据，并在购物车界面设置清除选项，类似于删除书籍的操作。

第四章 系统实现

4.1 系统实现介绍

4.1.1 登录模块实现

登录的过程如下，首先先在登入界面输入用户名，密码与验证码，此部分通过前端的 `html` 即可实现登入框，效果如下



登入框部分的 html 代码实现:

```
<form action="./login" method="post">
  <div class="user-box">
    <input required="" name="user" type="text" value="admin">
    <label>用户名</label>
  </div>
  <div class="user-box">
    <input required="" name="pass" type="password" value="123456">
    <label>密码</label>
  </div>
  <div class="user-box">
    <input required="" name="code" type="text">
    <label>验证码</label>
  </div>

  <a >
    <span></span>
    <span></span>
    <span></span>
    <span></span>
    <input class = "submit" type="submit" value="登录系统">
  </a>
</form>
```

其中验证码部分通过./code 进行自动生成，使用了 hutoll 实现图片验证码的生成：

```
@WebServlet("/code")
public class AuthCodeServlet extends HttpServlet {

    4 usages
    public final static String AUTH_CODE = "AUTH_CODE";

    lastcavespider
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {
        CircleCaptcha captcha = CaptchaUtil.createCircleCaptcha( width: 120, height: 60, codeCount: 4, cir
        String code = captcha.getCode();
        request.getSession( b: true).setAttribute(AUTH_CODE, code);

        try(OutputStream outputStream = response.getOutputStream()) {
            captcha.write(outputStream);
        }
    }
}
```

在后端通过 loginservlet（）类对用户提交的用户名密码以及验证码进行确认登入成功即可跳转主界面

```
if (code == null || !code.equalsIgnoreCase((String) session.getAttribute(AuthCodeServlet.AUTH
    response.sendRedirect( s: "./admin.html");
    return;
}
boolean star = LoginDao.loginStar(user, pass); //将获取的账号密码传过去，如果账号密码正确就会返回true
if ( star ) {
    // 登陆成功，在session中添加登陆成功的标记
    session.setAttribute(AuthFilter.AUTH_STATUS, AuthStatus.LOGIN_SUCCESS);
    // 返回一个重定向
    response.sendRedirect( s: "./main");

    System.out.println(user+pass);

    return;
}
```

首先判断验证码是否正确，正确则调用 logindao（这个类包含了用户的登入，注册，删除等一系列操作）中的用户登入部分

```

public static boolean loginStar(String account, String password) { //提供两个参数，第一
    PreparedStatement preSql; //预处理语句
    ResultSet rs; //定义一个结果集（存放结果）
    String sqlStr = "select * from users where account = ? and password = ?"; //?表
    try {
        preSql = con.prepareStatement(sqlStr);
        preSql.setString( parameterIndex: 1, account); //相当于把上面第一个问号设置成账号
        preSql.setString( parameterIndex: 2, password); //把上面第二个问号设置成密码

        rs = preSql.executeQuery(); //查询，返回值放到rs里面
        if ( rs.next() ) { //rs是表头，看他下一个是否有东西，如果里面有东西执行了，就返回true
            return true;
        } else { //如果查询失败了
            return false;
        }
    } catch (SQLException e) {
        return false;
    }
}
}

```

在 loginstart 部分通过上去了语句根据用户名与密码在 user 表中查询，查询成功则返回 true

然后回到 loginservlet 中添加登入成功的 session 标记，即可转跳到主界面

4.1.2 界面拦截模块实现

界面拦截模块使用过滤器进行处理

```

public void doFilter(HttpServletRequest request, HttpServletResponse response,
    // 放行不需要拦截的资源路径
    String uri = request.getRequestURI();

    for (String ignoreUrl : this.ignoreUrls) {
        if (uri.endsWith(ignoreUrl)) {
            chain.doFilter(request, response);
            return;
        }
    }

    // 对其它资源进行登录验证
    HttpSession session = request.getSession( b: true);
    AuthStatus authStatus = (AuthStatus) session.getAttribute(AUTH_STATUS);
    if ( authStatus != null && authStatus.equals(AuthStatus.LOGIN_SUCCESS) ) {
        chain.doFilter(request, response);
        return;
    } else {
        response.sendRedirect( s: "./index.html");
    }
}
}

```


将不需要拦截的网页，Java 类，图片等资源放行（在 xml 文件进行过滤器部署）

```
<filter>
  <filter-name>AuthFilter</filter-name>
  <filter-class>cn.edu.swu.gyh.security.AuthFilter</filter-class>
  <init-param>
    <param-name>ignoreUrls</param-name>
    <param-value>index.html;admin.html;register.html;register_successful.html;delete.html;delete_su
  </init-param>
</filter>
```

这样即可防止用户跳过登入界面直接进入管理系统。

4.1.3 用户管理模块实现

用户管理模块包括用户的注册与注销功能

首先实现用户的注册功能

大致思路类似于用户验证，首先由用户在 form 表单中填写数据并传输，并两次确认密码相同，在后端接受后使用 SQL 语句插入数据库的 user 表中即完成了用户的添加

```
if ( !pass1.equals(pass2) ) {
    response.setContentType("text/html;charset=utf-8");
    response.getWriter().print("<script language='javascript'>" +
        "alert('两次密码不一致，请重新输入! ');" +
        "window.location.href='register.html';</script>");
}

if ( pass1.equals(pass2) ) {
    int a = LoginDao.loginRegister(user, pass1);
    if ( a == 1 ) {
        //注册成功，在session中添加登陆成功的标记
        session.setAttribute(AuthFilterRegister.Register_SUCCESS, AuthStatus.Register_SUCCE
        //返回重一个定向
        response.sendRedirect( s: "./register_successful.html");
    } else if ( a == 3 ){
        response.setHeader( s: "Refresh", s1: "3;url=./register.html"); //隔3秒钟刷新到register
        Writer writer = response.getWriter();
        writer.write( str: "<center><h1 style='color:red'>用户名密码不正确，需要重新注册，3秒后自动跳
    } else if ( a == 0 ) {
        JOptionPane.showMessageDialog( parentComponent: null, message: "账号重复，请重新注册!", tit
        response.sendRedirect( s: "./register.html");
    }
}
```

Logindao 类中向数据库中插入用户的部分

```

public static int loginRegister(String account, String password) { //提供两个参数
    PreparedStatement preSql; //预处理语句
    String sqlStr = "insert into users values(?, ?)"; //?表示从外部输入的
    int num;

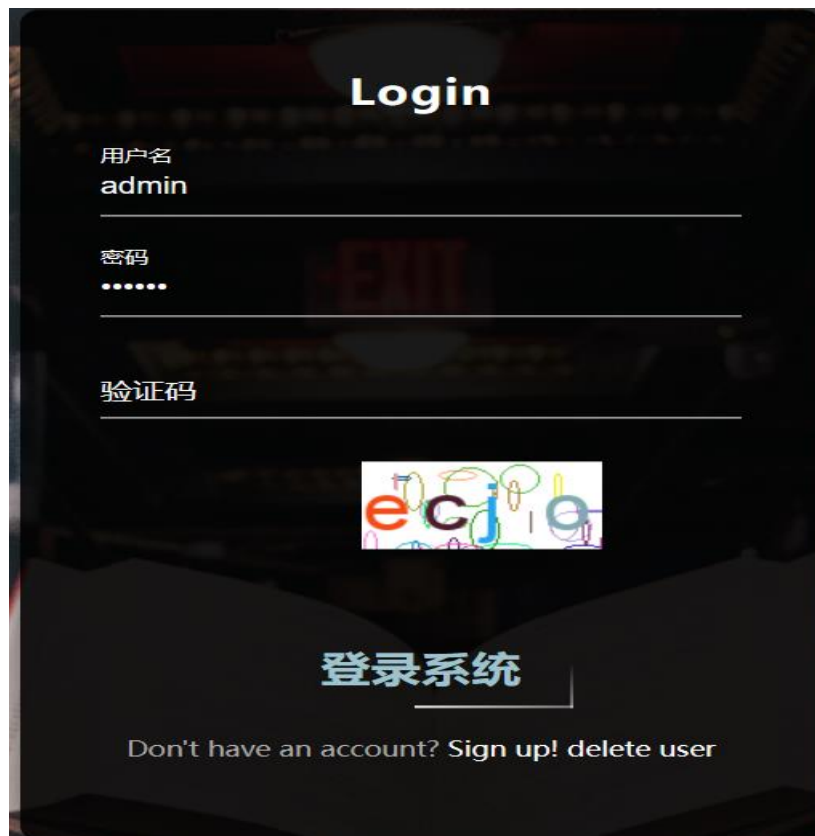
    try {
        preSql = con.prepareStatement(sqlStr);
        preSql.setString( parameterIndex: 1, account); //相当于把上面第一个问号设置成账号
        preSql.setString( parameterIndex: 2, password); //把上面第二个问号设置成密码

        num = preSql.executeUpdate(); //更新数据，返回值放到num里面
        return num;
    } catch (SQLException e) { //账号重复
        return 3;
    }
}

```

注销用户与注册用户类似，仍是通过 form 表单传输注销用户的数据，并在数据库的用户表中删除数据及完成用户的注销

用户管理的效果展示：



点击 sign up 进入用户注册页面

欢迎注册成为本书城用户

Register

Username
高裕涵

Password
.....

input Password again
.....

code
wlo2



Register clear

返回登录界面

在表单中填写好数据后点击注册按钮



提示注册成功

检查数据库的用户表格

admin	123456
spider	20040507
高裕涵	197896

发现注册的用户已被写入数据库，证明注册成功

进入用户注销界面

账户注销，请谨慎考虑！

Delete

Username

Password

code



Delete

[返回登录界面](#)

填写好用户名与密码等内容后尝试注销

users 输入一个 SQL 表达式来过滤结果 (假

	account	password
1	admin	123456
2	spider	20040507

刷新数据库发现用户已从数据库中删除，证明注销成功

4.1.4 书籍管理展示模块

书籍管理模块与用户模块操作类似，实现书籍的添加，修改，查找等主要功能

添加修改仍是通过 form 表单向后端传输数据进行数据库的修改实现，书籍的展示通过 pagetool 构造 table 对数据库内的元素进行展示

在 pagetool 类中使用 for 循环生成数据的表格进行展示

```
public static String wrapBookTable(List<Book> books) {
    StringBuilder table = new StringBuilder();
    table.append("<table class='tb-book'>");
    table.append("<tr><th>ID</th><th>书名</th><th>作者</th><th>价格</th><th>内容</th><th>图片</th><th>操作</th></tr>");
    for (Book book : books) {
        table.append(String.format(
            "<tr><td>%s</td><td>%s</td><td>%s</td><td>%s</td><td>%s</td><td>%s</td><td><a href='./image/uplo: "
            + "<td><a href='./deleteBook?bookId=%s'>删&nbsp;除</a></td>" +
            "<td><a href='./updateBook.html?bookId=%s'>修&nbsp;改</a></td>" +
            "<td><a href='./Addshopping.html?bookId=%s'>加入<br>购物车</a></td></tr>",
            book.getId(), book.getName(), book.getAuthor(),
            book.getPrice(), book.getContent(), book.getPicture(), book.getPicture(),
            book.getId(), book.getId(), book.getId()
        ));
    }
    table.append("</table>");
    return table.toString();
}
```

其中书籍的添加与用户注册略有不同的地方在于书籍图片的添加为多媒体上传

```
public void doPost(HttpServletRequest request, HttpServletResponse response)
// 检测是否为多媒体上传
if (!JakartaServletFileUpload.isMultipartContent(request)) {
    // 如果不是则停止
    PrintWriter writer = response.getWriter();
    writer.println("Error: 表单必须包含 enctype=multipart/form-data");
    writer.flush();
    return;
}

// 配置上传参数
DiskFileItemFactory.Builder builder = DiskFileItemFactory.builder();
builder.setCharset("UTF-8");
DiskFileItemFactory factory = builder.get();

JakartaServletFileUpload upload = new JakartaServletFileUpload(factory);
upload.setFileSizeMax(MAX_FILE_SIZE);
upload.setSizeMax(MAX_REQUEST_SIZE);
upload.setHeaderCharset(Charset.forName("UTF-8"));
String uploadPath = request.getServletContext().getRealPath(s: ".")
    + File.separator + "image" + File.separator + "upload";
```

需要进行处理，将图片的路径进行拼接

删除书籍与用户的注销功能十分类似

书籍管理效果演示：

欢迎访问SWU网上书城

购物车 首页 添加

书名 查询

ID	书名	作者	价格	内容	图片			
12	CS:APP	Bryant	128.0	《深入理解计算机系统》——简称CSAPP，被称为计算机领域的圣经		删除	修改	加入购物车
7	计算机网络	James	127.0	计算机网络经典黑皮书		删除	修改	加入购物车
5	算法导论	托马斯·科尔曼	80.0	MIT四大名师联手铸就的计算机算法圣经		删除	修改	加入购物车

第1页 / 共1页 1

进行书籍的添加：

欢迎访问SWU网上书城

购物车 首页 添加

书名 查询

ID	书名	作者	价格	内容	图片			
13	鸟哥的Linux私房菜	鸟哥	199.0	用通俗易懂的语言细致讲解Linux操作系统		删除	修改	加入购物车
12	CS:APP	Bryant	128.0	《深入理解计算机系统》——简称CSAPP，被称为计算机领域的圣经		删除	修改	加入购物车
								

书籍被成功添加并显示

进行修改（修改价格作为演示）

购物车 首页 添加

书名▼ 查询

书名:


作者:

价格:

内容:


图片: 未选择任何文件

修改价格为 99 并提交

ID	书名	作者	价格	内容	图片			
13	鸟哥的Linux私房菜	鸟哥	99.0	用通俗易懂的语言细致讲解Linux操作系统		删除	修改	加入购物车

价格修改成功

删除书籍测试

ID	书名	作者	价格	内容	图片			
13	鸟哥的Linux私房菜	鸟哥	99.0	用通俗易懂的语言细致讲解Linux操作系统		删除	修改	加入购物车

点击删除启动删除类

ID	书名	作者	价格	内容	图片			
12	CS:APP	Bryant	128.0	《深入理解计算机系统》——简称CSAPP，被称为计算机领域的圣经		删除	修改	加入购物车
7	计算机网络	James	127.0	计算机网络经典黑皮书		删除	修改	加入购物车
5	算法导论	托马斯·科尔曼	80.0	MIT四大名师联手铸就的计算机算法圣经		删除	修改	加入购物车

发现书籍别删除成功

搜索测试

购物车 首页 添加

书名▼算查询

ID	书名	作者	价格	内容	图片			
12	CS:APP	Bryant	128.0	《深入理解计算机系统》——简称CSAPP，被称为计算机领域的圣经		删除	修改	加入购物车
7	计算机网络	James	127.0	计算机网络经典黑皮书		删除	修改	加入购物车
5	算法导论	托马斯·科尔曼	80.0	MIT四大名师联手铸就的计算机算法圣经		删除	修改	加入购物车

尝试查询带有算字的书籍

ID	书名	作者	价格	内容	图片			
5	算法导论	托马斯·科尔曼	80.0	MIT四大名师联手铸就的计算机算法圣经		删除	修改	加入购物车
7	计算机网络	James	127.0	计算机网络经典黑皮书		删除	修改	加入购物车

查询结果成功

至此书籍管理的基本功能实现完成

4.1.5 主界面书籍展示模块实现

此部分采用 JSON 格式从后端获取数据尝试实现前后端分离


```

public void doGet(HttpServletRequest request, HttpServletResponse response)thr

    try {
        String sql = "select * from book order by id desc";
        List<Book> books = BookRepo.getInstance().queryBook(sql);
        ObjectMapper mapper = new ObjectMapper();
        String json = mapper.writeValueAsString(books);

        response.setContentType("application/json");
        response.setCharacterEncoding("UTF-8");
        try(Writer writer = response.getWriter()){
            writer.write(json);
        }

    } catch (SQLException | IOException e) {
        throw new ServletException(e);
    }
}

```

将数据库中的数据以 JSON 格式取出

在 JavaScript 文件中循环读取 books 中的每一本书籍并将内容展示

```

for (var book of books){

    var div =$(`
        <div class="col-md-4">
            <div class="card">
                <div class="card-body pb-5">
                    <div class="pt-4 pb-5">
                        <img src="" class="img-fluid img-center" style="height: 150px;" a
                    </div>
                    <h5 class="h4 lh-130 mb-3"></h5>
                    <p class="text-muted mb-0 author-price"></p>
                    <p class="text-muted mb-0 content"></p>
                </div>
            </div>
        </div>
    `);

    div.find("img").attr("src", "./image/upload/" + book.picture);
    div.find("h5").append(book.name);
    div.find("p.author-price").append("作者: "+book.author+", 价格: ¥"+ book.price);
    div.find("p.content").append(book.content);
    $("#booksView").append(div);
}

```

效果展示

书籍信息展示



CS:APP

作者: Bryant, 价格: ¥128
《深入理解计算机系统》——简称CSAPP, 被称为计算机领域的圣经



计算机网络

作者: James, 价格: ¥127
计算机网络经典黑皮书



算法导论

作者: 托马斯·科尔曼, 价格: ¥80
MIT四大名师联手铸就的计算机算法圣经

4.1.6 购物车模块实现

购物车模块的实现部分, 我采用的思路较为简单, 当点击添加购物车是通过一个中间的购物车确认界面, 将数据通过 form 表单形式传入数据库中是 shoppingcart 表中, 并进行显示, 数据的删除与书籍管理 删除功能类似

购物车功能实现所需要的 Java 类

```
shopping
├── AddBookCart
├── Addshoppingcart
├── CartRepo
├── DeleteBookCart
├── MainShoppingCart
├── SearchBookCart
├── ShoppingCart
└── ShoppingPageTool
```

效果演示:

ID	书名	作者	价格	内容	图片	删除	修改	
12	CS:APP	Bryant	128.0	《深入理解计算机系统》——简称CSAPP, 被称为计算机领域的圣经		删除	修改	加入购物车
7	计算机网络	James	127.0	计算机网络经典黑皮书		删除	修改	加入购物车
5	算法导论	托马斯·科尔曼	80.0	MIT四大名师联手铸就的计算机算法圣经		删除	修改	加入购物车

第1页 / 共1页

点击加入购物车链接

购物车 首页 添加

书名 查询

书名:

作者:

价格:

内容:

《深入理解计算机系统》——简称CSAPP，被称为计算机领域的圣经

图片:

选择文件

 11549886167...96143174.jpg

加入购物车

进入确认加入界面
确认加入后

购物车

返回 首页 书名 查询

ID	书名	作者	价格	内容	图片	
6	CS:APP	Bryant	128.0	《深入理解计算机系统》——简称CSAPP，被称为计算机领域的圣经		删除

第1页 / 共1页, 1

即可在购物车见到该商品

购物车表格的数据

shoppingcart 输入一个 SQL 表达式来过滤结果 (使用 Ctrl+Space)

	123 id	abc name	abc author	123 price	abc content	abc picture
1	6	CS:APP	Bryant	128	《深入理解计算机系统》——简称	902e56d3-2e23-4a84-ba83

点击删除

购物车

返回 首页 书名 查询

ID	书名	作者	价格	内容	图片	
----	----	----	----	----	----	--

第1页 / 共0页,

购物车被清空

shoppingcart 输入一个 SQL 表达式来过滤结果 (使用 Ctrl+Space)					
	123 id	ABC name	ABC author	123 price	ABC content

购物车数据表刷新后也被清空

购物车功能即完成实现

4.2 系统实现的不足

在此次网上书店项目设计中，我的系统仍存在不少不足，首先是界面美化部分做的不到位，其次是购物车的实现较为简陋，并没有将用户与购物车数据绑定，还有在进行查找后分页并未重置的问题，由于时间有限，目前并不能很好的进行解决

第五章 总结与展望

此次网上书城项目给我的感受颇深，Java 的面向对象编程与以前我在 C 语言学习时常用的面向过程思维有着较大的不同，面向对象的网络编程更像是对黑盒功能的直接调用组合而可以对底层接口实现的过程进行忽略，通过一个个接口的交错调用最终形成了一份较为完整的项目，这次设计无论是对我的编程能力还是学习能力都是一次较大的考验，遇到了不少问题也在老师同学的帮助之下得以解决，总体来说还是将网上书城的功能进行了基本的操作实现。希望在以后的空闲时间能继续自行学习技术将一些遗留的问题进行解决，将这个项目完善成一个功能齐全且好用的较为完整的项目！