

# 中国大学生计算机设计大赛



## 软件开发类作品文档简要要求

作品编号： 2023021577

作品名称： 拾柒爱阅读

作 者： 佐佳豪、谢胜尧、雷万强

版本编号： V1.2.3.20230427-rc

填写日期： 2023 年 4 月 27 日

### 填写说明：

- 1、本文档适用于**所有**涉及软件开发的作品，包括：软件应用与开发、大数据应用、人工智能应用、物联网应用；
- 2、正文一律用五号宋体，一级标题为二号黑体，其他级别标题如有需要，可根据需要设置；
- 3、本文档为简要文档，不宜长篇大论，简明扼要为上；
- 4、提交文档时，以 PDF 格式提交本文档；
- 5、本文档内容是正式参赛内容组成部分，务必真实填写。如不属实，将导致奖项等级降低甚至终止本作品参加比赛。

# 目 录

<b>第一章 需求分析 .....</b>	<b>3</b>
<b>第二章 概要设计 .....</b>	<b>4</b>
<b>第三章 详细设计 .....</b>	<b>4</b>
界面设计: .....	4
数据库设计: .....	7
关键算法: .....	8
技术创新: .....	10
技术重点: .....	10
<b>第四章 主要测试 .....</b>	<b>11</b>
<b>第五章 安装及使用 .....</b>	<b>13</b>
运行、安装环境要求及过程.....	13
运行环境 .....	13
运行过程 .....	13
安装环境 .....	13
安装过程 .....	13
Nginx 配置文件.....	14
<b>第六章 项目总结 .....</b>	<b>15</b>
<b>参考文献 .....</b>	<b>16</b>

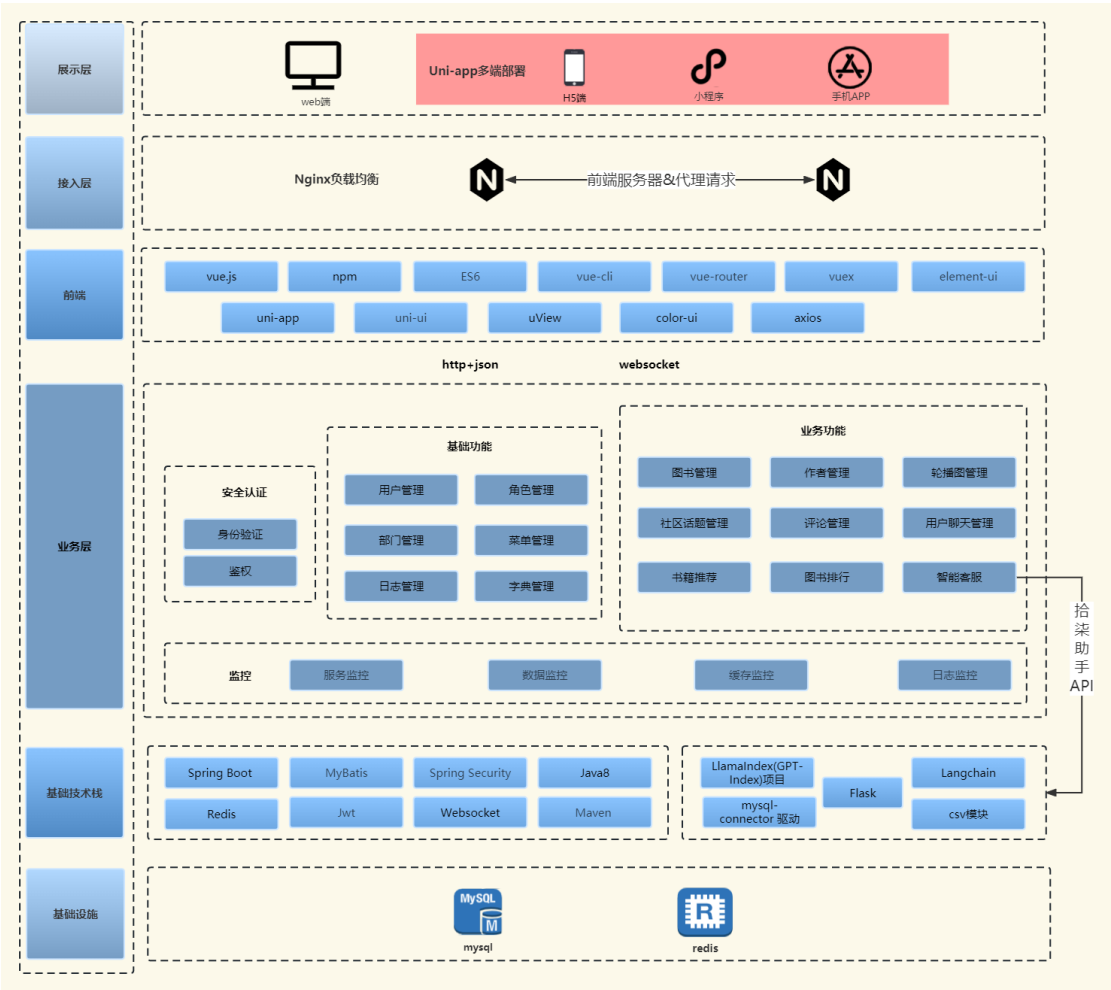
# 第一章 需求分析

为响应“学习强国”理念，增强“全民阅读”意识。为此，我们推出了这个长篇书籍阅读分享平台。现代人生活忙碌，阅读时间少。碎片化阅读虽快捷，却有诸多弊端，如浅表、误导、失落等。你可以在这里找到各类书籍和书评，按照兴趣选择阅读，并记录分享心得感受。你也可以和其他读者交流讨论，拓展视野知识。我们希望通过这个项目，让更多人享受阅读的乐趣和收获，培养阅读习惯和能力，提升文化素养和人文关怀。本作品对标了豆瓣和网易云阅读两个平台，将豆瓣的书评特色功能与网易云阅读的在线阅读功能相结合，使得用户拥有更好的功能体验。

下表为竞品分析表格：

维度	本作品	豆瓣	网易云阅读
在线阅读	支持 PDF 格式，目录、笔记功能	不支持在线阅读，只能查看图书信息、评论、购书单等内容	支持多种格式，目录、书签、笔记功能
书评功能	支持写书评、点赞、评论，查看他人书评和推荐，形成社交氛围	有豆瓣最受欢迎的书评资源	只支持查看他人评论
其他功能	支持模糊搜索、基于 ChatGPT 为平台的智能客服、加入书架、喜欢，个性化推荐	支持搜索、分类浏览、收藏、推荐，个性化推荐	同豆瓣

## 第二章 概要设计



## 第三章 详细设计

### 界面设计：

在前台图书详情界面中，鼠标悬浮至“移动端阅读”，将出现该图书的二维码，扫描二维码即可跳转至对应的 H5 端图书详情页。此二维码的生成并不是传统技术中的后端生成前端接收，而是直接在前端中根据图书的 ID 进行生成二维码，从而减少网络层压力。



## 中国历史中的生存游...

宗承源 著 四川出版社 出版

分类: 历史

发布于: 2015-03-03

最近更新: 2023-04-28

立即阅读

取消加入书架

扫码分享阅读



欢迎知名up "泡面拌饭" 加入拾柒爱阅读



宗承源

评论

发表评论

发表

下图展示了前台页面的评论模块，利用 Vue 的组件化特性和数据结构的原理，采用了父子组件的递归生成方式，并且父子组件可以通过方法和参数的传递实现互动，从而构建了高效可解耦的树形结构评论。



泡面拌饭 2023-04-27 13:04:50

一种生存的竞争和生命规则。

IP属地: 江西省 南昌市

回复 0



liefox 2023-04-27 21:23:56

有一种纯真的美

IP属地: 江西省 南昌市

回复 1

回复 liefox

发布



liefox 2023-04-28 21:20:39

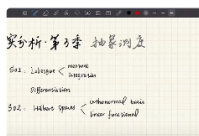
高效的评论结构树

IP属地: 江西省 南昌市

回复 0

下图展示了社区话题页面，其亮点在于每篇话题的“阅读全文”和“展开评论”的状态控制，由于每篇话题是通过“v-for”遍历生成的，如果使用同一个状态变量来管理，就会导致点击某篇话题的“阅读全文”后，其他文章也会跟着展开，影响用户体验。为了优化这个功能，我们给每篇话题添加了两个布尔值的属性，分别用来控制其“阅读全文”和“展开评论”的状态，并且在前端使用“map”函数对后端返回的数据进行处理，给每条数据插入这两个属性，实现了状态的独立控制。

## liefox: 《线性代数应该这样学》（《Linear Algebra Done Right》）这本书到底好在哪儿？

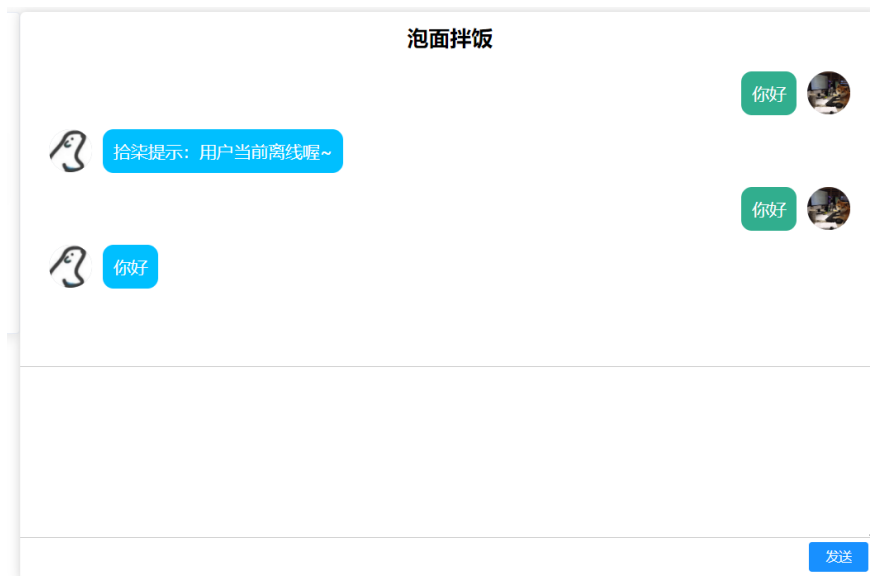


因为大家的评论反响还不错, 所以我在这放心地自荐一下, 相信题主会对这本书有一个全新的认识. 而且题主想要的信息在这个视频的评论区里基

0 条评论 2023-04-25 18:56:43

阅读全文

下图展示了前台的实时聊天功能界面，其中的聊天气泡是由“v-html”指令动态生成的，根据用户发送接收信息时的内容进行字符串拼接。同时，输入框绑定了键盘的回车事件，可以实现回车发送信息的便捷操作。



下图为本作品实现了实时的消息提醒功能，当用户不在聊天界面时，如果有人给他发私信，就会弹出一个通知框。该功能的难点在于利用全局的 WebSocket 监听私信事件，并弹出相应的提醒，同时支持用户一键点击跳转到相应的聊天界面。



下图为后台的数据大屏页面，使用 Echarts 展示了本平台的数据可视化，使得管理者更为直观的分析数据。该功能展示了本平台的用户量、在线用户数、图书参数等关键指标，以及图书的阅读量、喜欢数、加入书架数、评论数来生成图表。



下图为 H5 端的社区广场页面，该功能的特点是通过下滑加载更多数据。在用户下滑时，会先判断后端是否有下页数据，如果有，就将下页数据追加到列表中，如果没有，就提示“没有更多了”。











## 数据库设计：

本作品的数据库设计没有采用外键，主要基于以下几点考虑： 1、本作品面向的用户量大，需要支持高并发访问，而数据库服务器往往是性能的瓶颈，受限于 IO 能力，且难以实现水平扩展； 2、本作品将数据一致性的控制交由 Mybatis 事务处理，即由应用服务器承担这部分的压力； 3、本作品的应用服务器可以轻松地进行水平伸缩，提高系统的可扩展性。 因此，本作品放弃了使用外键的设计，同时也加速了开发进度。

## 系统表

<b>gen_table_column</b> column_id bigint table_id varchar(64) column_name varchar(200) column_comment varchar(500) column_type varchar(100) java_type varchar(500) java_field varchar(200) is_pk char(1) is_increment char(1) is_required char(1) is_insert char(1) is_edit char(1) is_list char(1) is_query char(1) query_type varchar(200) html_type varchar(200) dict_type varchar(200) sort int create_by varchar(64) create_time datetime update_by varchar(64) update_time datetime	<b>gen_table</b> table_id bigint table_name varchar(200) table_comment varchar(500) sub_table_name varchar(64) sub_table_fk_name varchar(64) class_name varchar(100) tpl_category varchar(200) package_name varchar(100) module_name varchar(30) business_name varchar(30) function_name varchar(50) function_author varchar(50) gen_type char(1) gen_path varchar(200) options varchar(1000) create_by varchar(64) create_time datetime update_by varchar(64) update_time datetime remark varchar(500)	<b>sys_menu</b> menu_id bigint menu_name varchar(50) parent_id bigint order_num int path varchar(200) component varchar(255) query varchar(255) is_frame int is_cache int menu_type char(1) visible char(1) status char(1) perms varchar(100) icon varchar(100) create_by varchar(64) create_time datetime update_by varchar(64) update_time datetime remark varchar(500)	<b>sys_user</b> user_id bigint dept_id bigint user_name varchar(30) nick_name varchar(30) user_type varchar(2) email varchar(50) phonenumber varchar(11) sex char(1) avatar varchar(100) password varchar(100) status char(1) del_flag char(1) login_ip varchar(128) login_date datetime create_by varchar(64) create_time datetime update_by varchar(64) update_time datetime remark varchar(500)	<b>sys_oper_log</b> oper_id bigint title varchar(50) business_type int method varchar(100) request_method varchar(10) operator_type int oper_name varchar(50) dept_name varchar(50) oper_url varchar(255) oper_ip varchar(128) oper_location varchar(255) oper_param varchar(2000) json_result varchar(2000) status int error_msg varchar(2000) oper_time datetime
<b>sys_dict_data</b> dict_code bigint dict_sort int dict_label varchar(100) dict_value varchar(100) dict_type varchar(100) css_class varchar(100) list_class varchar(100) is_default char(1) status char(1) create_by varchar(64) create_time datetime update_by varchar(64) update_time datetime remark varchar(500)	<b>sys_dept</b> dept_id bigint parent_id bigint ancestors varchar(50) dept_name varchar(30) order_num int leader varchar(20) phone varchar(11) email varchar(50) status char(1) del_flag char(1) create_by varchar(64) create_time datetime update_by varchar(64) update_time datetime	<b>sys_role</b> role_id bigint role_name varchar(30) role_key varchar(100) role_sort int data_scope char(1) menu_check_strictly tinyint dept_check_strictly tinyint status char(1) del_flag char(1) create_by varchar(64) create_time datetime update_by varchar(64) update_time datetime remark varchar(500)	<b>sys_job</b> job_id bigint job_name varchar(64) job_group varchar(64) invoke_target varchar(500) cron_expression varchar(255) misfire_policy varchar(20) concurrent char(1) status char(1) create_by varchar(64) create_time datetime update_by varchar(64) update_time datetime remark varchar(500)	<b>asq_books</b> book_id bigint book_writer_name varchar(20) book_name varchar(20) book_type varchar(20) book_pub_date datetime book_up_date datetime book_press varchar(20) book_cover varchar(255) book_pdf_url varchar(255) book_audit int
<b>sys_notice</b> notice_id int notice_title varchar(50) notice_type char(1) notice_content longblob status char(1) create_by varchar(64) create_time datetime update_by varchar(64) update_time datetime remark varchar(255)	<b>sys_post</b> post_id bigint post_code varchar(64) post_name varchar(50) post_sort int status char(1) create_by varchar(64) create_time datetime update_by varchar(64) update_time datetime remark varchar(500)	<b>sys_logininfor</b> info_id bigint user_name varchar(50) ipaddr varchar(128) login_location varchar(255) browser varchar(50) os varchar(50) status char(1) msg varchar(255) login_time datetime	<b>sys_config</b> config_id int config_name varchar(100) config_key varchar(100) config_value varchar(500) config_type char(1) create_by varchar(64) create_time datetime update_by varchar(64) update_time datetime remark varchar(500)	<b>sys_job_log</b> job_log_id bigint job_name varchar(64) job_group varchar(64) invoke_target varchar(500) job_message varchar(500) status char(1) exception_info varchar(2000) create_time datetime
<b>sys_dict_type</b> dict_id bigint dict_name varchar(100) dict_type varchar(100) status char(1) create_by varchar(64) create_time datetime update_by varchar(64) update_time datetime remark varchar(500)	<b>sys_role_menu</b> role_id bigint menu_id bigint	<b>sys_user_post</b> user_id bigint post_id bigint	<b>sys_user_role</b> user_id bigint role_id bigint	<b>sys_role_dept</b> role_id bigint dept_id bigint

## 业务表

<div><div>asq_comment</div><div><div></div><div>comment_id int</div><div>comment_parent_id int</div><div>user_id int</div><div>com_body_id int</div><div>comment_text varchar(255)</div><div>com_body_type int</div><div>create_date datetime</div><div>like_num int</div><div>is_vio int</div></div></div>	<div><div>asq_topic</div><div><div></div><div>topic_id int</div><div>user_id int</div><div>topic_name varchar(255)</div><div>topic_content text</div><div>topic_cover varchar(255)</div><div>topic_createdate datetime</div></div></div>	<div><div>asq_rank_parm</div><div><div></div><div>rank_parm_id int</div><div>book_id int</div><div>read_num int</div><div>applaud_num int</div><div>comment_num int</div><div>collect_num int</div></div></div>	<div><div>asq_applaud</div><div><div></div><div>applaud_id int</div><div>user_id int</div><div>book_id int</div><div>cread_data datetime</div></div></div>	<div><div>asq_chat</div><div><div></div><div>id int</div><div>user_id int</div><div>from_id int</div><div>content text</div><div>time datetime</div></div></div>	<div><div>asq_collect</div><div><div></div><div>collect_id int</div><div>book_id int</div><div>user_id int</div><div>create_date datetime</div></div></div>
<div><div>asq_ppt</div><div><div></div><div>ppt_id int</div><div>ppt_name varchar(50)</div><div>ppt_pic varchar(255)</div><div>ppt_url varchar(255)</div><div>ppt_desc int</div><div>ppt_type int</div></div></div>	<div><div>asq_writer</div><div><div></div><div>writer_id bigint</div><div>writer_name varchar(20)</div><div>writer_info text</div><div>writer_intro varchar(255)</div><div>writer_pic varchar(255)</div></div></div>				

## 关键算法：

协同过滤算法（根据用户的行为：对图书的评论，喜欢，阅读，加入书架等，来进行推荐相关的图书）：

```
/**
 * 协同过滤算法
```



```

* 根据用户的行为：对图书的评论，喜欢，阅读，加入书架等，来进行推荐相关的图书
*
* @return 用户可能喜欢的书籍
*/
@Override
public List<AsqBooks> GYWLike() {
    ArrayList<AsqBooks> asqBooks1 = new ArrayList<>();
    Random random = new Random();
    AsqApplaudAndBookInfoVo vo = new AsqApplaudAndBookInfoVo();
    //判断用户是否登录
    if (SecurityUtils.getUnUserId() != -1) {
        vo.setUserId(SecurityUtils.getUnUserId());
    }
    //查询用户加入书架的图书列表
    List<AsqApplaudAndBookInfoVo> bookInfoList =
asqApplaudMapper.selectAsqApplaudAndBookInfoList(vo);
    //如果用户有加入书架的图书
    if (bookInfoList.size() != 0) {
        //书籍类型数组
        String[] key = new String[3];
        for (int i = 0; i < 4; i++) {
            //书籍类型随机添加到书籍类型数组
            AsqApplaudAndBookInfoVo vo1 =
bookInfoList.get(random.nextInt(bookInfoList.size()));
            key[0] = vo1.getBookType();
        }
        AsqBooks asqBooks = new AsqBooks();
        for (int i = 0; i < key.length; i++) {
            //根据书籍类型数组查询图书，并将查询的书籍随机返回
            asqBooks.setBookType(key[i]);
            List<AsqBooks> asqBooks2 =
asqBooksMapper.selectAsqBooksList(asqBooks);
            asqBooks1.add(asqBooks2.get(random.nextInt(asqBooks2.size())));
        }
        return asqBooks1;
    } else {
        //如果书架没有一本书的话，就随机查询
        List<AsqBooks> asqBooks = asqBooksMapper.selectAsqBooksList(new
AsqBooks());
        int[] ints = randomKey(asqBooks.size());
        for (int i = 0; i < ints.length; i++) {
            asqBooks1.add(asqBooks.get(ints[i]));
        }
        return asqBooks1;
    }
}
}

```

```

/**
 * @param size 随机数组的长度
 * @return 返回随机数组
 */
private static int[] randomKey(int size) {
    Set<Integer> ints = new HashSet<>();
    while (true) {
        ints.add(new Random().nextInt(size));
        if (ints.size() == 3) {break;}
    }
    int[] ints1 = new int[3];
    Object[] objects = ints.toArray();
    for (int i1 = 0; i1 < ints.size(); i1++) {
        ints1[i1] = (int) objects[i1];
    }
    return ints1;
}

```

下面 SQL 是进行书籍的排名参数权重来计算排名，从而加入推荐列表：

```

SELECT
    ab.*
FROM
    asq_rank_parm arp
    LEFT JOIN asq_books ab ON arp.book_id = ab.book_id
ORDER BY
    (
        # read_num 阅读数*0.2+applaud_num 喜欢数*0.3+comment_num 评论数
        *0.2+collect_num 加入书架数*0.3
        arp.read_num * 0.2 + arp.applaud_num * 0.3 + arp.comment_num * 0.2
        + arp.collect_num * 0.3
    ) DESC
LIMIT 3

```

## 技术创新：

平台利用 Redis 的特性，实现了 24 小时阅读记录、未读私信、评论通知、首页数据缓存、数据字典等功能，提升了系统速度和用户体验。平台还使用 ChatGPT 作为智能客服平台，结合业务需求，打造了专属的智能客服。但由于国内对 ChatGPT 的限制，响应速度有待提高，我们将持续优化。

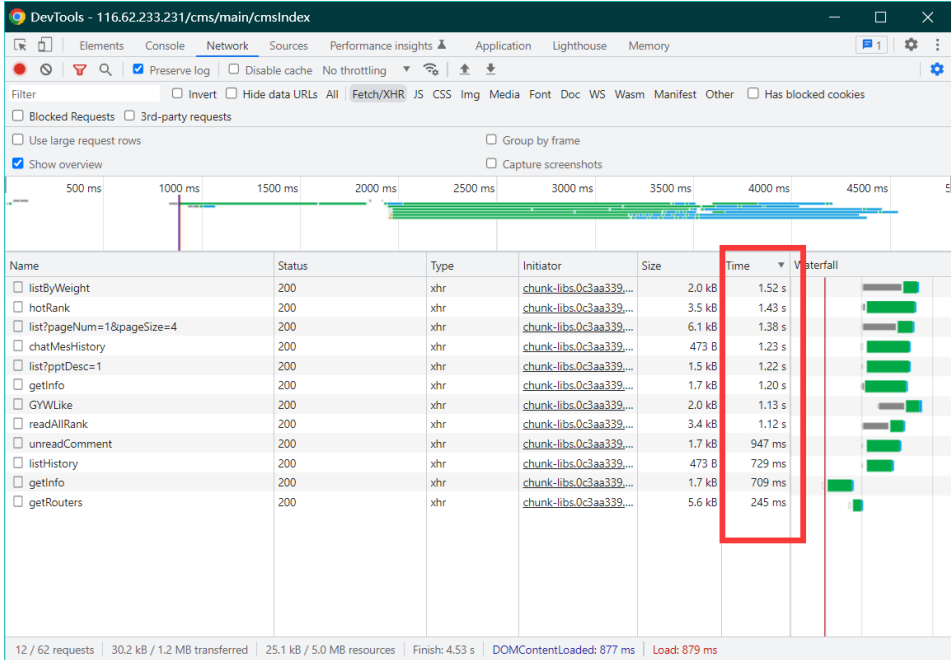
## 技术重点：

我们的平台提供了基于 WebSocket 的实时聊天功能，让用户能够方便地沟通和交流。同时，我们采用了前后端分离的开发模式，提高了开发效率和质量。此外，我们还使用了 uni-app 作为我们 H5 端的框架，支持了多种平台和设备的兼容性。

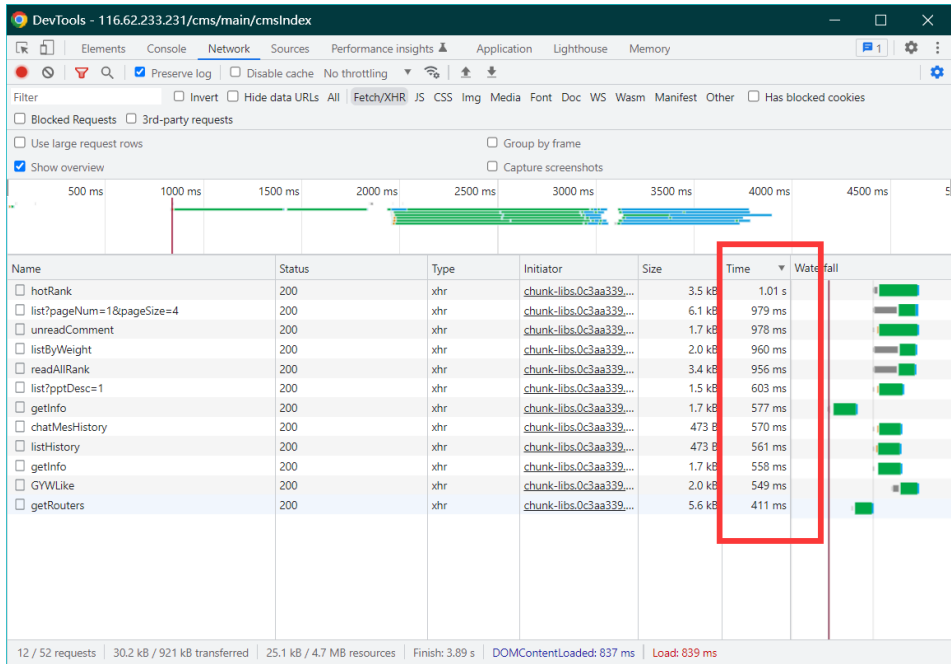
# 第四章 主要测试

## 1. 首页数据接口的访问速度测试：

下图为访问速度，发现平均每个接口耗时 1s 左右，为优化首页访问速度，特将接口数据存入 Redis 进行缓存处理。



下图为修正后的结果，明显提升了加载速度。

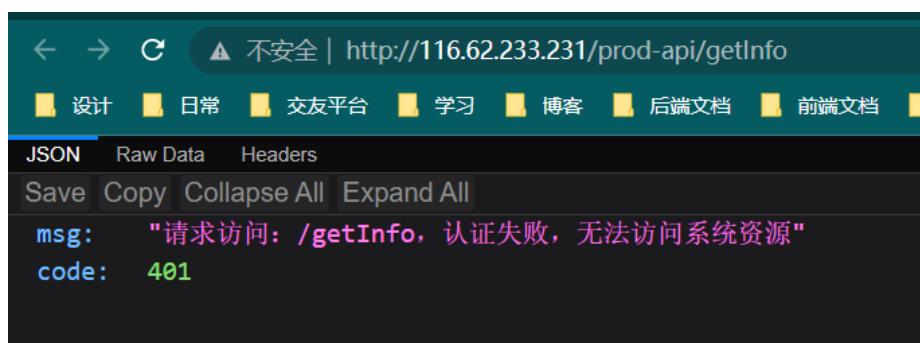


## 2. 全平台的数据鉴权测试：

下图为普通用户的后台的菜单，在页面杜绝越权操作。



下图为非法访问本平台接口，杜绝的接口对外暴露。



### 3. 平台的功能扩展性测试：

本平台采用前后端分离的开发模式，使用 Git 进行版本控制，具有良好的功能扩展性。同时，使用 Nginx 作为负载均衡和代理服务器，支持高并发和高负载的访问。

### 4. 本平台部署所需环境和步骤：

环境：JDK1.8、MySQL5.7、Python 3.9、Redis、Nginx。

部署步骤：

1. 使用 Maven 将后端代码打包成 Jar 包；
2. 使用 “npm run build:prod” 命令将前端代码打包成静态文件；
3. 使用 HBuilder X 的打包命令将 H5 端代码打包成多端项目；
4. 将前端和 H5 端的静态文件放到 Nginx 服务器的相应目录下；
5. Linux 服务器上执行 “java -jar 包名” 命令运行后端 Jar 包；
6. 将 chatbotapi1 文件夹上传至服务器，后运行 “pip install -r modules.txt” 命令，在运行 “sudo nohup python main.py &” 命令即可。

### 5. 平台可用性：

为了适应不同设备的访问需求，本平台在前端增加了移动端设备的判断和重定向功能。当用户使用移动端设备访问 web 端时，前端会自动重定向至 H5 端，以提供更好的用户体验。

# 第五章 安装及使用

## 运行、安装环境要求及过程

### 运行环境

- 1、JDK  $\geq$  1.8（推荐 1.8 版本）；
- 2、MySQL  $\geq$  5.7.0（推荐 5.7 版本）；
- 3、Redis  $\geq$  3.0；
- 4、Maven  $\geq$  3.0；
- 5、Node  $\geq$  12。

### 运行过程

- 1、将 sql 文件夹下的 sq\_books.sql 文件在 MySQL 中执行，创建 sq\_books 数据库；
- 2、将总项目文件夹导入 IDEA，打开 application.yml 和 application-druid.yml 配置文件，进行 mysql 和 redis 的相关配置，加载 Maven 依赖，后运行 springboot 启动类；待控制台出现“拾柒启动成功”即后端项目启动成功；
- 3、在进入 sq-ui 文件夹下，打开 cmd 命令行，输入“npm i”后等待加载依赖，待依赖加载完毕后再执行“npm run dev”，编译完成后便会自动打开前台项目（注意：配置 src 目录下的 config.js 文件）；
- 4、将 sq-app 项目导入至 HBuilderX 中，点击菜单中的运行至浏览器即可（注意：配置根目录下的 config.js 文件）。

### 安装环境

- 1、一台 Linux CentOS 7.9 64 位操作系统服务器；
- 2、JDK 1.8；
- 3、MySQL 5.7；
- 4、Nginx 1.17；
- 5、Redis 3.2；
- 6、一台海外 Linux CentOS 7.9 64 位操作系统服务器（国内 ChatGPT 限制）；
- 7、Python 3.9。

### 安装过程

- 1、创建好数据库，导入 SQL 文件，配置 Redis 和 Nginx 配置文件；
- 2、打包后端文件，使用 IDEA 自带的 Maven 工具，分别先后执行 Maven Clean、Maven Compile、Maven package 命令，然后在 ruoyi-admin 文件夹下的 target 文件夹下会生成一个 Jar 包，将 Jar 包上传至服务器指定目录下后，运行“nohup java -jar 包名.jar & tail -100f nohup.out”即可运行项目，待输出“拾柒启动成功”即可；

- 3、在 sq-ui 文件夹下打开 cmd 命令行，运行 “npm build:prod” 命令，然后会在根目录下生成一个 dist 文件夹，将 dist 文件夹上传至 Nginx 服务器上（路径：/home/shiqi），即可访问（注意：配置 Nginx 监听端口）；
- 4、在 HBuilderX 中打开 sq-app，点击菜单中的发行 -> 网站后，构建打包成功之后，会在根目录生成 unpackaged/dist/build/h5 文件夹，将 h5 文件夹上传至 Nginx 服务器（路径：/home/shiqi），即可访问（注意：配置 Nginx 监听端口）；
- 5、打开 chatbotapi1 中的 dbreader.py 文件，修改对应的数据库连接配置，然后将 chatbotapi1 文件夹上传至海外服务器中，运行 “pip install -r modules.txt” 命令，在运行 “sudo nohup python main.py &” 命令即可。

## Nginx 配置文件

```
worker_processes 1;

events {
    worker_connections 1024;
}

http {
    include mime.types;
    default_type application/octet-stream;
    sendfile on;
    keepalive_timeout 65;
    client_max_body_size 20M;

    server {
        listen 80;
        server_name localhost;
        charset utf-8;

        location / {
            root /home/shiqi/dist;
            try_files $uri $uri/ /index.html;
            index index.html index.htm;
        }

        location /prod-api/ {
            proxy_set_header Host $http_host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header REMOTE-HOST $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_pass http://localhost:8080/;
        }

        error_page 500 502 503 504 /50x.html;
```

```

        location = /50x.html {
            root    html;
        }
    }
server {
    listen        81;
    server_name   localhost;
    charset       utf-8;

    location / {
        root      /home/shiqi/h5;
        try_files $uri $uri/ /index.html;
        index      index.html index.htm;
    }

    error_page    500 502 503 504 /50x.html;
    location = /50x.html {
        root      html;
    }
}
}
}

```

## 第六章 项目总结

### 1. 项目协调

协调是整个项目的灵魂环节。我们需要和不同的团队成员建立紧密的协调，保障项目的进度和质量。为了达成这一目标，我们使用了 Git 作为团队协作的神器，将代码托管在 Gitee 上，用 Swagger 来进行生成前后端分离的接口文档。通过这些工具，我们可以轻松地进行实时交流，并及时解决问题和更新进度。

### 2. 任务分解

任务分解是项目成功的秘诀。在这个项目中，我们采用了前后端分离+敏捷开发的组合拳，将整个项目拆分为多个迭代，每个迭代都有清晰的任务和目标。这样整个项目就可以在可控的范围内高效地进行。

### 3. 克服的困难

在这个项目中，最大的挑战就是对接 ChatGPT，由于国内的网络限制导致响应速度不稳定。为了解决这个问题，我们采用了拾柒助手后端项目在国外服务器上部署，可以有效提升服务响应的速度和质量。

### 4. 水平提升

这个项目让我们体验了一场技术的盛宴，我们尝试了很多新鲜和有趣的技术和方法，比如用 Python 编程语言来作为基础来实现用 ChatGPT 接口调用智能问答机器人，用 Websocket 通信协议来实现实时交互，用 Redis 缓存来提升数据的读写速度和实际业务场景的应用，用数据分析和可视化工具来展示数据的价值等。这些都是非常酷和有前途的技能，可以让我们在未来的工作中更加游刃有余和出类拔萃。我们也学习了更好的项目管理和协作方式，比如用 Git 来管理代码的版本，用 Swagger 来进行生成前后端分离的接口文档。这些都是非常方便和高效的工具，可以让我们在未来的项目中更加顺畅和默契。

### 5. 升级演进

为了让这个项目更加完善和实用，我们在设计时也考虑了升级和改进的空间。我们可以在未来添加更多的数据源，比如从其他网站或平台爬取数据，或者从用户反馈中获取数据，以增加数据的多样性和丰富性。我们也可以扩充 Redis 集群和 MySQL 集群的规模，以提高数据的存储和读取的效率和稳定性。另外，我们还可以提高数据处理的精度，比如使用更先进的算法或模型，或者优化数据清洗和预处理的流程，以提高数据的质量和可信度。我们还可以覆盖更多的问题类型，比如除了问答之外，还可以支持推荐、搜索、分类等功能，以满足用户的不同需求。

## 6. 商业推广

这个项目具有很强的商业推广潜力。它借鉴了豆瓣和网易云阅读等竞品的优势，结合了书评社区和在线阅读的功能，打造了一个完善的平台。同时，它还支持广告接入和创作者激励机制，能够吸引更多的广告商和优质创作者入驻。这样，就能形成一个良性循环：广告商的投放促进创作者产出高质量内容，高质量内容又吸引更多的用户，用户的增长又带动更多的广告商和创作者加入，从而实现商业闭环。

# 参考文献

暂无引用参考文献。