



**QG工作室项目报告**

学院 物理与光电工程学院

专业 光电信息科学与工程

班级 22级光电信息科学与工程(3)班

组别 前端组

姓名 蒋卓然

学号 3122006814

2023年4月30日

广东工业大学计算机学院制

目录

[1. 项目简介 1](#_Toc10329)

[2. 技术选型 1](#_Toc30941)

[3. 项目要求 1](#_Toc8923)

[4.设计思路 2](#_Toc11888)

[5.详细功能 5](#_Toc29921)

[6.程序测试 13](#_Toc559)

[7.项目亮点 14](#_Toc26172)

项目代码亮点...........................................................................................................14

项目功能亮点............................................................................................................22

[8.项目难点 25](#_Toc30919)

[9.心得体会 26](#_Toc32656)

# 项目简介

为了提高图书管理的效率，本项目针对图书的管理，基于JavaScript设计了一个面向图书的管理系统。 本系统主要有四个大的功能：对图书的增删改查，对图书的留言评论与收藏，对图书的分类浏览与排名，以及对管理员和普通用户权限的管理。  
 关键词：管理系统 图书管理 权限分离 前后端分离

# 技术选型

1. 前端技术：HTML、CSS、JavaScript、js-xlsx
2. 后端技术：Node.js、MySQL、websocket

# 项目要求

1. 登录注册+用户信息展示和编辑（使用token或cookie验证身份）
2. 展示所有书籍简要信息列表，每一本书包括一张缩略图以及标题等必要信息，文字过长的处理方式 （自由发挥，点击进入详情）
3. 模糊查询：根据书名或作家名或书籍号查询书籍，并显示查询结果
4. 筛选：根据标签或类型筛选书籍
5. 实现权限分离，分为管理员和普通用户，各自特有功能如下：

**普通用户**

作者上传书籍，进行表单验证。需要填写的信息有：

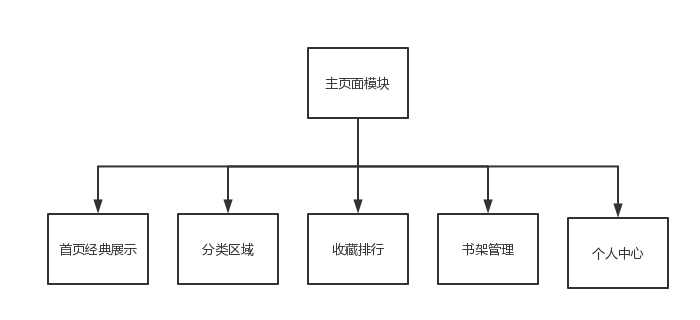
1. 申请人个人信息：包括姓名、电话、地址等基本信息（可自行确定）。数据格式需要检测，排版需要注意。
2. 书籍信息：书名、书籍介绍、书的封面（最好可预览和取消上传)、书籍内容(txt文件，内容无需太长，可自己定义)
3. 选择书籍类型（3-4个供选择）
4. 添加标签（自定义）
5. 查看平台所有书籍的详细信息（某些隐私信息不展示），选择部分书籍以轮播图形式展示信息(书籍推荐)
6. 个人中心，可修改自己的信息（例如密码），查看自己上传的书籍（待审核、已通过和被退回三类），被退回需要显示管理员的退回理由
7. 向管理员提出举报或反馈，可以查看结果

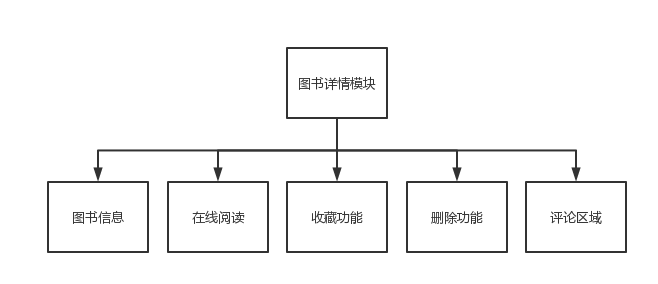
**管理员**

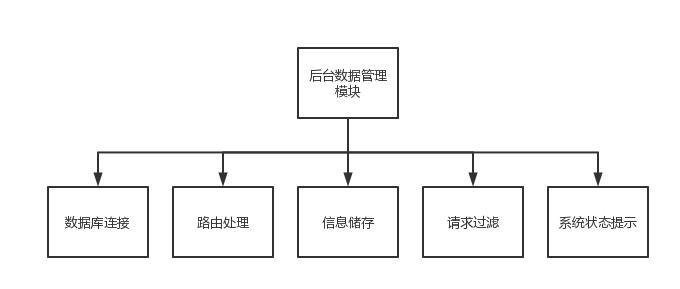
1. 可管理所有书籍，包括增删改
2. 审核书籍申请，给出通过或不通过的反馈
3. 处理并展示所有举报/反馈信息

# 4.设计思路

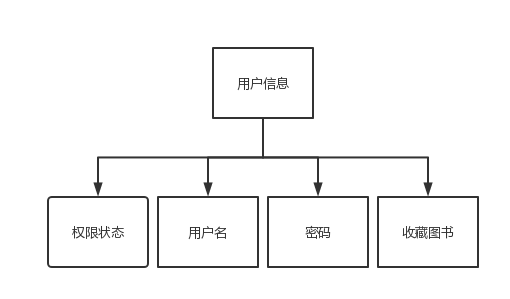
该系统分为三个模块，分别是主页面模块、图书详情模块以及后台数据管理模块，这三个模块分别实现了对图书信息的系统功能展示、前端动态渲染、后台增删改查等操作，其系统功能模块如下：

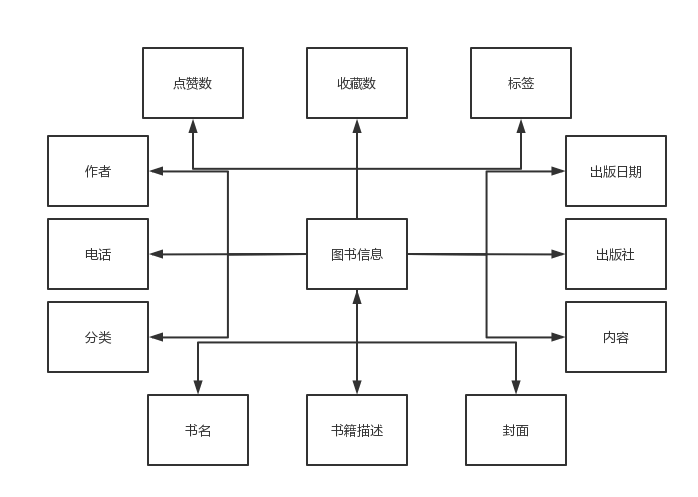


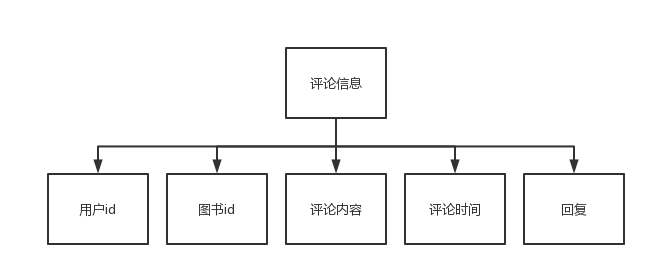
****

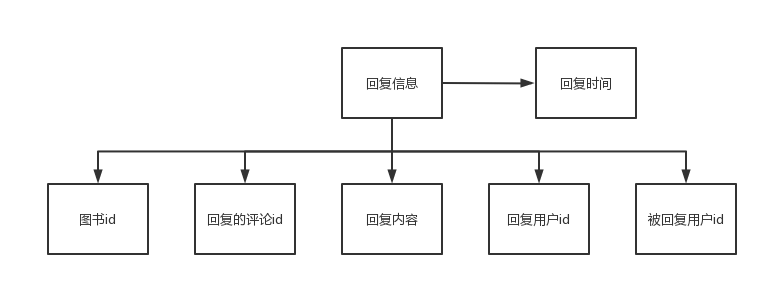
****

除此之外，该系统对应的数据存储结构分为三种，分别是用户信息存储结构、图书信息存储结构以及评论信息存储结构，如下所示：



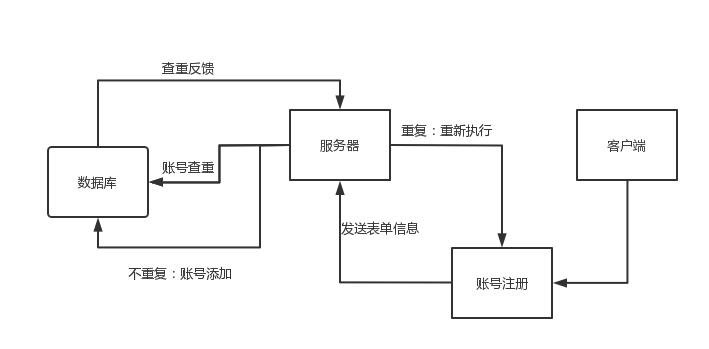


****

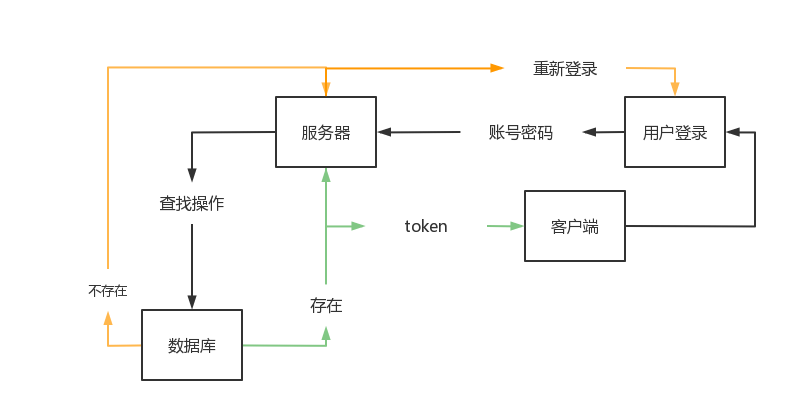
****

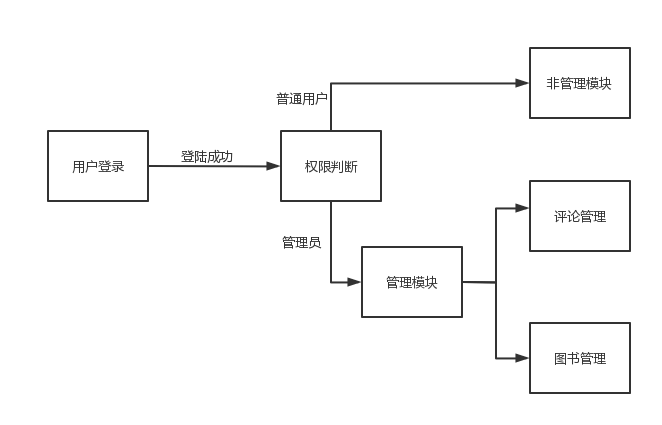
# **5.**详细功能

1. 用户注册。用户可以进行新帐号的创建，通过服务器对数据库中的用户信息进行添加操作。

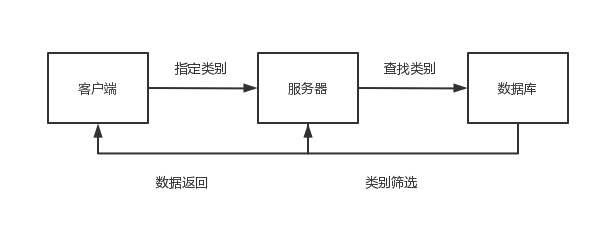


1. 用户登录。管理员或普通用户根据用户名和密码进行[身份验证](https://cloud.tencent.com/product/mfas?from=20065&from_column=20065" \t "https://cloud.tencent.com/developer/article/_blank)（token）登录系统。系统会根据登录信息及进行身份判断，并进行权限设置。权限状态若判断为用户，则无法点击管理模块的按钮与进行管理相关的操作(例如：权限状态判断之后若为普通用户，删除按钮会被隐藏)。

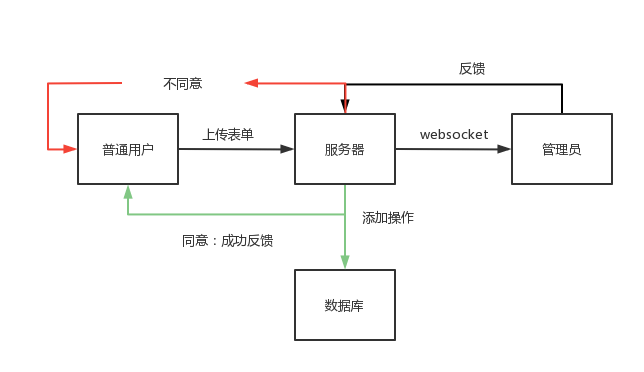


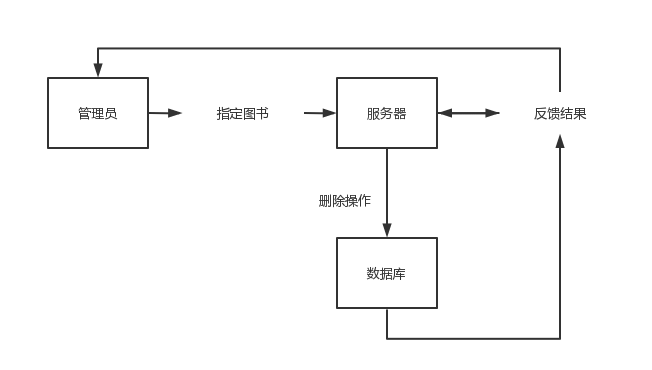


1. 轮播图和经典展示。选取具有年度热度的图书海报和图书简介在首页进行展示。
2. 图书分类。根据图书上传的类别进行筛选，在用户进行选择时把指定类图书从数据库中筛选出来并返回给服务器，再由服务器发送至客户端进行页面渲染。

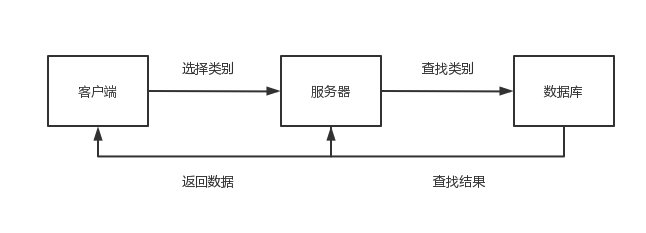


1. 图书管理。用户可以上传图书，并将表单信息转发至管理员的对应页面，并显示处理状态。管理员对用户上传的图书表单信息进行审核，自动删除该条申请，并返回处理结果给用户。通过后发送至服务器上传数据库，同时可以根据图书编号、图书名称查询图书基本信息并进行对应图书信息的修改或删除操作。

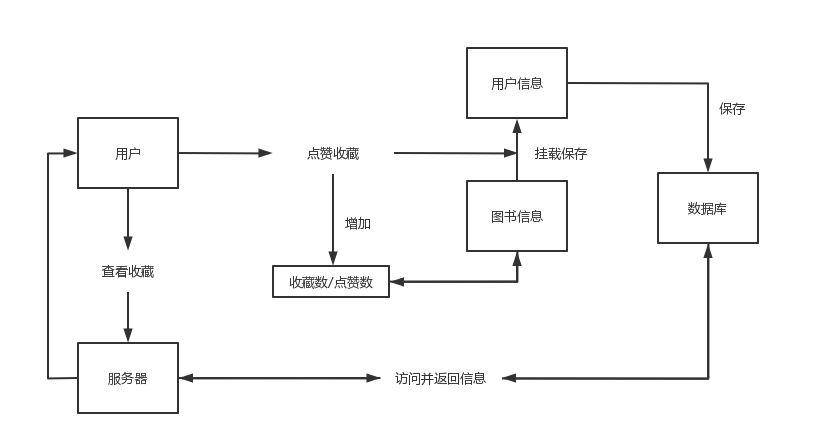




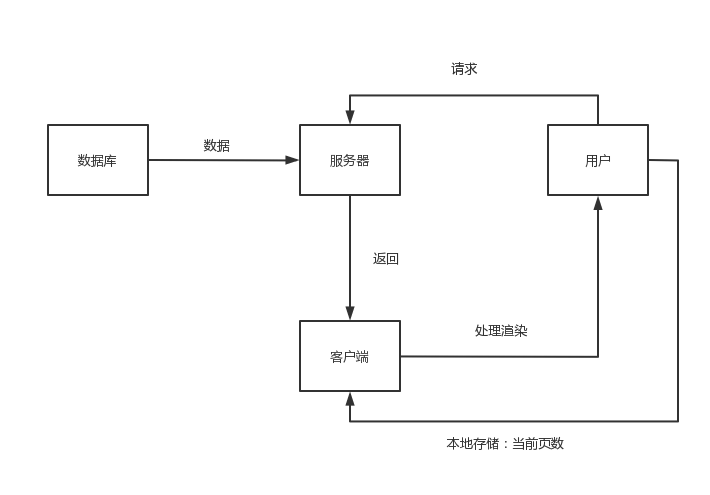
1. 图书排行。服务器根据书籍收藏量或点赞数的大小对大量图书数据进行单变量从高到低的排队，并把处理好的数据返回给客户端进行渲染。



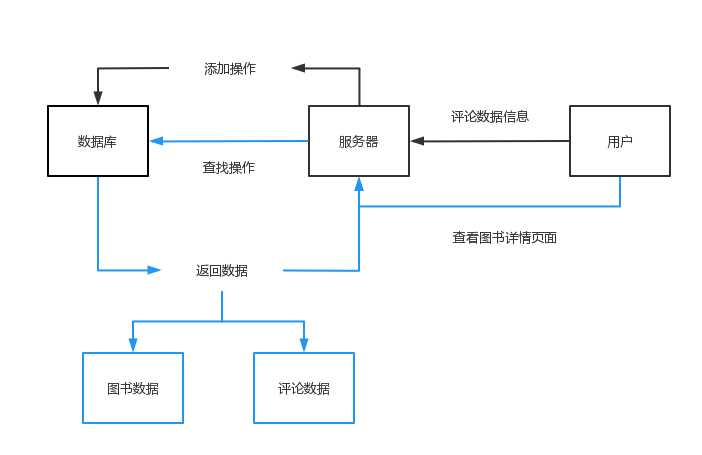
1. 图书点赞收藏。用户可以对图书进行点赞和收藏操作，点击图书详情页的收藏按钮可以将该图书信息挂载到用户数据结构的节点下，当查看收藏图书模块时会向服务器发送请求拿到数据，进行渲染。用户对点赞和收藏的点击事件会进行次数统计，保存到书籍信息中。



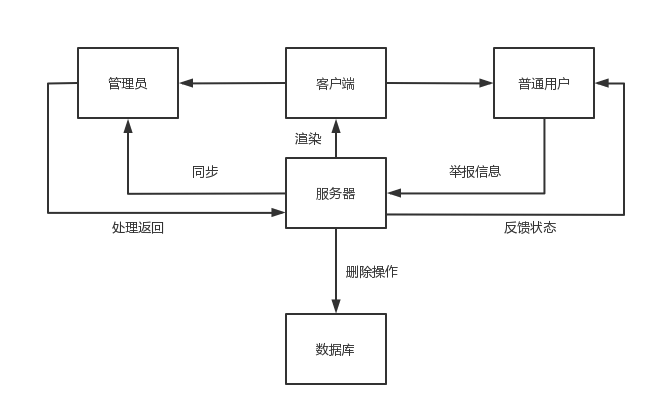
1. 图书阅读。用户对内容可进行查看，即将服务器返回的图书信息结构下的内容结点文件（txt）读取并渲染到指定的阅读区域，而且系统会根据阅读区域的视窗大小经过一系列简单算法实现自动分页功能，并对用户查看的页数进行记录并本地保存，在下次查看时即可通过该页数直接跳转，实现书签功能。除此之外，用户还可以将文件下载下来。



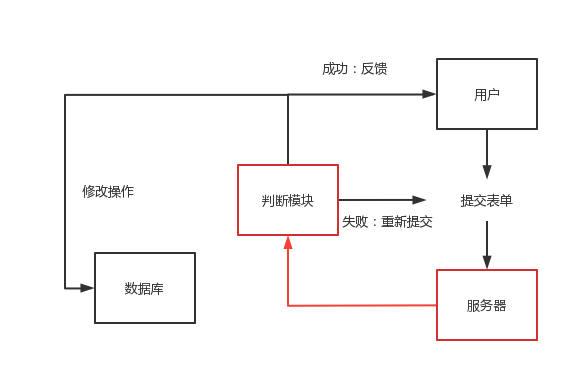
1. 图书信息导出。通过xlsx可以将服务器返回的图书作者、出版社等信息以EXCEL表格形式导出。
2. 评论管理。所有用户在评论区的所有评论与回复以及对应图书的编号会挂载到评论信息储存结构的结点上，当加载某图书的书籍详情页时会通过对应图书编号自动从该数据库中获取具体的评论信息并返回渲染。



1. 举报反馈。用户可以对图书或者其他用户评论进行举报申请，举报的理由会通过websocket同步到管理员的举报反馈页面，管理员通过查看申请判断是否执行有关内容删除操作，并同步返回处理结果至用户举报反馈界面，若提交后管理员没有及时反馈，会在用户界面显示处理中的状态。



1. 个人信息修改。通过表单验证（旧密码的验证），用户可以自行对自己的密码等进行更改，并将更改过后的信息保存到数据库中。



# **6.**程序测试

一个好的程序离不开成千上万次的测试，下列为测试的标准及期望：

1. **UI界面测试**

界面设计风格：符合项目整体风格，色彩搭配协调，整体美感良好。

布局排版：页面元素布局整齐有序，不会产生错位或遮挡等问题。

图片和图标：图片清晰且展现效果良好，图标大小适宜，不会出现模糊等情况。

字体和字号：字体大小适当、字体类型清晰易读，字号与内容相匹配。

界面交互：交互逻辑清晰，响应速度快，用户体验良好。

1. **功能测试**

功能覆盖率：测试用例需全面覆盖所有功能点。

功能正确性：功能点需按照需求文档实现，不能出现错误或异常情况。

数据有效性：对输入数据进行有效性验证，确保数据格式正确并能被接受。

数据完整性：保证数据在存储、处理过程中不会丢失或损坏。

安全性：检查用户输入是否存在安全隐患，防止系统被恶意攻击。

1. **兼容性测试**

系统环境：测试需覆盖主流操作系统和浏览器版本。

设备适配：测试需覆盖PC、手机、平板等设备的兼容性。

响应式布局：确保页面在不同分辨率下能够正常显示。

兼容性问题修复：及时解决出现的兼容性问题，确保系统能够稳定运行。

1. **易用性测试**

操作简便性：用户可以快速找到需要的功能，操作流程简单明了。

可理解性：用户能够正确理解各个功能的意义和作用。

易学性：新用户可以在短时间内掌握基本使用方法。

反馈及时性：系统反馈信息及时精准，用户能够获得良好的使用体验。

1. **性能测试**

页面加载速度：测量页面加载速度，确保用户不会因为等待时间过长而流失。

响应时间：检查系统响应时间是否符合要求，避免用户操作时出现卡顿现象。

并发性能：测试系统在高并发环境下的稳定性和承载能力。

内存泄漏：检查系统内存使用情况，防止出现内存泄漏问题。

1. **安全测试**

XSS攻击：防止恶意脚本攻击，避免用户输入数据引发安全隐患。

CSRF攻击：防止跨站点请求伪造攻击，保护系统数据不被非法篡改。

SQL注入攻击：防止用户输入的恶意SQL语句对数据库进行攻击。

权限控制：确保用户访问权限符合角色所规定的范围，避免用户越权操作。

# **7.**项目亮点

**项目代码亮点：**

1. **分包与代码风格**

**HTML规范**

基于 W3C、苹果开发者 等官方文档，并结合团队业务和开发过程中总结的规范约定，让页面HTML代码更具语义性。

**图片规范**

了解各种图片格式特性，根据特性制定图片规范，包括但不限于图片的质量约定、图片引入方式、图片合并处理等，旨在从图片层面优化页面性能。

**CSS规范**

统一规范团队 CSS 代码书写风格和使用 CSS 预编译语言语法风格，提供常用媒体查询语句和浏览器私有属性引用，并从业务层面统一规范常用模块的引用。

**命名规范**

从目录、图片、HTML/CSS文件、ClassName的命名等层面约定规范团队的命名习惯，增强团队代码的可读性。

**JavaScript规范**

* 与效率相关（开发、迭代和维护，重点提升维护及迭代效率）
* 与团队相关（减少团队之间的不一致性）
* 与面试相关（提高代码健壮性）
* 与习惯相关（保证最近实践）
* 与开源相关（开源项目均有严格的开发规范）

1. **字体图标及svg矢量图**

* 轻量级：一个图标字体要比一系列的图像要小。一旦字体加载了，图标就会马上渲染出来，不需要下载一个个图像。这样可以减少HTTP的请求数量，而且和HTML5的离线存储配合，可以对性能做出优化。
* 灵活性：不调字体可以像页面中的文字一样，通过font-size属性来对其进行大小的设置，而且还可以添加各种文字效果，如color、hover、filter、text-shadow、transform等效果。
* 兼容性：图标字体支持现代浏览器，甚至是低版本的IE浏览器，所以可以放心的使用它。
* 相比于位图放大图片会出现失真、缩小又会浪费掉像素点，图标字体不会出现这种情况。

1. **reset样式文件**

在HTML标签在浏览器里有默认的样式，例如 p 标签有上下边距，strong标签有字体加粗样式，em标签有字体倾斜样式，不同浏览器的默认样式之间也会有差别。

reset.css是重置浏览器标签的样式表，其作用就是重新定义标签样式，覆盖浏览器的CSS默认属性，也就是指把浏览器提供的默认样式覆盖掉。

1. **CSS3关键帧及状态过渡的动画效果**

* 浏览器可以对动画进行优化（元素不可见时不动画，减少对FPS的影响）
* 实现代码比较简单
* 可以利用硬件加速
* 不占用主线程
* 相较于JavaScript动画，避免触发不必要的回流与重绘，减少内存堆

1. CSS自定义属性及变量的应用

复杂的网站都会有大量的 CSS 代码，通常也会有许多重复的值。举个例子，同样一个颜色值可能在成千上百个地方被使用到，如果这个值发生了变化，需要全局搜索并且一个一个替换（十分麻烦与繁琐）。自定义属性在某个地方可以存储一个值，然后在其他许多地方都能引用它

另一个好处是语义化的标识。比如，–main-text-color 会比#00ff00更易理解，尤其是这个颜色值在其他上下文中也被使用到。

1. 代码注释(TODO)

* 有助于保持一致性
* 有助于理解
* 帮助修补程序或快速修复
* 有助于加快开发过程
* 有助于提高协作效率

1. **防抖与节流**

## 防抖可以有效的帮助前端优化性能，避免一些不必要的操作，如在发送ajax请求时，用户短时间内点击多次，就会发送多次请求，使用防抖函数就可以有效的避免这一问题（多次触发 只执行最后一次）

## 假设有个频繁触发的函数，在进行节流处理后，在规定的时间内，只让函数的第一次生效，第二次大于设定时间后再生效，以此类推。（规定时间内事件只触发一次）

1. **阻止默认行为和事件冒泡**

有些HTML元素存在默认的行为，比如说当点击a标签的时候，会执行跳转，点击form表单中的submit的时候，会提交表单并且跳转。

DOM标准事件流的触发的先后顺序为：先捕获再冒泡，即当触发DOM事件时，会先进行事件捕获，捕获到事件源之后通过事件传播进行事件冒泡。在绑定事件被触发时，往往会出现意料之外的额外事件触发。

1. **ES6及以上语法的广泛应用**

**箭头函数**

* 简洁的语法
* 隐式返回，如下面的代码可以去掉return,代码移到一行，减少代码量
* 解决了this的指向问题，原生的写法this指向的是调用者，箭头函数this绑定的是定义时的那个对象。如果有对象嵌套的情况，则this绑定到最近的一层对象上

**Promise对象**

有时我们要进行一些相互间有依赖关系的异步操作，比如有多个请求，后一个的请求需要上一次请求的返回结果。过去常规做法只能 callback 层层嵌套，但嵌套层数过多的话就会有回调地狱的问题。

更好地进行错误捕获。多重嵌套 callback 除了会造成上面讲的代码缩进问题，更可怕的是可能会造成无法捕获异常或异常捕获不可控。

如果使用 promises 的话，代码就会变得扁平且更可读了。前面提到 then 返回了一个 promise，因此我们可以将 then 的调用不停地串连起来。其中 then 返回的 promise 装载了由调用返回的值。而且，通过 reject 方法把 Promise 的状态置为 rejected，这样我们在 then 中就能捕捉到，然后执行“失败”情况的回调。

**扩展运算符、剩余运算符以及解构赋值**

扩展运算符可以将一个数组转为用逗号分隔的参数序列。其与正常的函数参数可以结合使用，后面也可以放置表达式。

剩余运算符将逗号隔开的值序列组合成一个数组或对象。

解构赋值按照一定的模式，从数组或对象中提取值，并对变量进行赋值。本质上，这种写法属于 “模式匹配”，只要等号两边的模式相同，左边的变量就会被赋予对应的值。

利用上述三种新语法，可以从存储服务器返回数据的数组中快速获取到目标个体的值或目标群体的集合并进行一系列的前端渲染操作，大大减少了中间处理步骤，提高了代码的效率。

**模板字符串**

* 可以多行拼接，维持良好的代码风格  
  可以在拼接中插入变量，实现定制式渲染  
  可以进行简单的运算  
  可以互相[嵌套](https://so.csdn.net/so/search?q=%E5%B5%8C%E5%A5%97&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/weixin_42929948/article/details/_blank)，实现结构完整强大的组件创建  
  简单，方便，整洁

1. **异步处理**

(关键词await/async/fetch/then/catch等)

* 异步流程可以立即给调用方返回初步的结果。
* 异步流程可以延迟给调用方最终的结果数据，在此期间可以做更多额外的工作，例如结果记录等等。
* 异步流程在执行的过程中，可以释放占用的线程等资源，避免阻塞，等到结果产生再重新获取线程处理。
* 异步流程可以等多次调用的结果出来后，再统一返回一次结果集合，提高响应效率

1. **SEO优化**

**优化SEO能在搜索引擎获得不错的排名，为网站或者网店带来流量，以下是优化的做法：**

### 合理的title、description、keywords

### 语义化的HTML代码，符合W3C规范：语义化代码让搜索引擎容易理解网页

### 非装饰性图片必须加alt

### 重要内容HTML代码放在最前：搜索引擎抓取HTML顺序是从上到下，保证重要内容一定会被抓取

### 少用iframe：搜索引擎不会抓取iframe中的内容

### 提高网站速度：网站速度是搜索引擎排序的一个重要指标

1. **闭包**

**闭包即当一个函数的返回值是另外一个函数,而返回的那个函数如果调用了其父函数内部的变量,且返回的这个函数在外部被执行。**

* 保护函数内的变量安全，实现封装，防止变量流入其他环境发生命名冲突
* 在内存中维持一个变量，可以做缓存
* 匿名自执行函数可以减少内存消耗

**b.项目功能亮点：**

1. **websocket实现心跳更新**

websocket协议相比http优势很明显，首先在效率上有多方面的提升。

传统的http请求，其并发能力都是依赖同时发起多个TCP连接访问服务器实现的(因此并发数受限于浏览器允许的并发连接数)，而websocket则允许我们在一条ws连接上同时并发多个请求，即在A请求发出后A响应还未到达，就可以继续发出B请求。由于TCP的慢启动特性（新连接速度上来是需要时间的），以及连接本身的握手损耗，都使得websocket协议的这一特性有很大的效率提升。

http协议的头部太大，且每个请求携带的几百上千字节的头部大部分是重复的，很多时候可能响应都远没有请求中的header空间大。如此多无效的内容传递是因为无法利用上一条请求内容，websocket则因为复用长连接而没有这一问题。

websocket支持服务器推送消息，这带来了及时消息通知的更好体验，也是ajax请求无法达到的。

1. **实现书籍评论功能，普通用户可以在书籍下方评论、点赞，也可以回复别人的评论。**

用户可以通过访问网站页面参与主题讨论，并留言保持文章新鲜感。

搜索引擎会通过文章的评论对文章所在页面进行评分。评论量越多，说明这个页面对用户的价值就越高，搜索引擎也就会提高网站排名。

评论越多，页面内容的可读性就越高，并增强网站上其他用户的信任。

评论内容应该是与所在网页内容有一定关联的，这可以提高网站的浏览量和转化率。

1. **以excel表格形式导出作家信息和书籍信息(核心技术js-xlsx)**

[xlsx](https://link.juejin.cn?target=https://github.com/SheetJS/js-xlsx" \o "https://github.com/SheetJS/js-xlsx" \t "_blank) 用于解析和编写各种电子表格。比如excel、csv、html文件

[xlsx-style](https://link.juejin.cn?target=https://www.npmjs.com/package/xlsx-style" \o "https://www.npmjs.com/package/xlsx-style" \t "_blank) 为xlsx库添加样式，比如字体颜色，大小，行宽等。

[FileSaver.js](https://link.juejin.cn?target=https://github.com/eligrey/FileSaver.js" \o "https://github.com/eligrey/FileSaver.js" \t "_blank) 负责下载保存文件。对兼容性比较友好，对跨域文件也有处理。

1. **图片懒加载**

减少了加载时的线程数量，使可视区域内的图片也能够快速加载，优化了用户体验。  
 减少了同一时间发向服务器的请求数即高并发情况，大大减轻了服务器的压力。

1. **在线阅读时，书籍内容过长时实现翻页并添加书签**
2. **实现草稿箱，保存被退回或正在编辑的书籍申请**

早些时候,本地存储使用的是 cookie。但是Web 存储需要更加的安全与快速. 这些数据不会被保存在服务器上，只用于用户请求网站数据上.它也可以存储大量的数据，而不影响网站的性能.  
 localStorag 能长久保存整个网站的数据，保存的数据没有过期时间，直到手动去除。

拓展了cookie的4k限制。

可以将第一次请求的5M大小数据直接存储到本地，相比于cookie可以节约带宽。

localStorage的使用也是遵循同源策略的，所以不同的网站直接是不能共用相同的localStorage，一定程度上加强了数据安全性。

# **8.**项目难点

1.良好的页面布局会不可避免地导致代码量过大，区域功能及分类不明显，结构较为复杂，维护难度及成本大大上升。需要良好的代码注释以及f12网页开发者工具的辅助，才能在调试阶段进行问题的定位与修改。

2.使用原生的代码进行开发会要求具备很强的逻辑思维能力以及基础知识应用能力。相较于使用框架，原生代码对于同一个功能的实现步骤可能会繁琐一些，这也一定程度上证明了维护的难度在升级。

3.实时同步显示书籍处理数据以及举报信息，需要实现心跳功能，实时向服务器发起请求，并更新最新消息状态。

4.加载只有一个HTML文件的单页面应用时，给DOM元素所起类名可能会出现重复现象，会造成CSS样式冲突，导致非意识性覆盖，影响页面效果，所以一个好的命名规范十分重要的。

5.管理系统中的数据量较大，需要高频次地向服务器申请大量资源，常常会导致一系列如因无法及时接受到对应的资源而导致的报错。并且高并发的请求模式会给服务器带来巨大的压力，从而导致反馈速度下降。因此，如何有效解决异步阻塞是本系统的一大难点。

6.svg在绘制静态图标上具有一定的优势，会需要大量的测试与比对才能实现一个图标的创作，如同在现实中拿着一支画笔，从绘制贝塞尔曲线到添加滤镜，制作过程需要耗费一定的时间，但同时也能在性能上带来一定的优化。

7.普通用户和管理员的权限分离，在多数情况下都是通过页面分流进行功能实现的，本系统为单页面应用，则需要通过有效的相关模块处理（如显示与隐藏管理员的工作模块、图书与评论的删除模块等），来表现管理员的权限不同，同时较少的页面跳转也大大减少了服务器的压力，提高了用户的体验质量。

8.亮点即是难点。所有亮点都是在克服所面临难点的基础上打磨出来的，因为克服难点这一过程的本身就是亮点。

# **9.**心得体会

宝剑锋从磨砺出，梅花香自苦寒来。经过训练营和最终考核的淬炼，我收获颇多，受益匪浅，不知不觉中，我已经成长为一个更好的自己:

**在编程能力方面**，我有了脱胎换骨的提升，从空有理论，纸上谈兵，到按部就班，脚踏实地；从惊慌失措，无从下手，到稳扎稳打，深思熟虑。实践才是检验真理的唯一标准，可以说，QG给予了我尝试与锻炼的机会，让我看到了自己的不足与局限。在训练营的学习中，我掌握并实现了C语言中的多种数据结构(栈，队列，链表，二叉树等）和若干排序算法（如十大排序），对指针的“万金油”作用有了比较全面的认识，同时丰富化的知识也对我最近的C语言课设考核有举足轻重的影响，相比于其他人的手足无措，我在项目搭建的过程中得心应手，运用所学的知识，在思考优化代码的同时，也能为课设小组的其他同学解答疑惑，推动小组工作稳步前进，最终拿出一份亮点十足的课设作业。其实，不积跬步，无以至千里;不积小流，无以成江海。一份好的课设正是对我在训练营中努力的最好回报。

此外，我还深刻认识到了HTML，CSS和原生JavaScript的底层逻辑，不再停留于JQuery，Bootstrapt，Vue等一系列框架的表层代码。而且，偏向于前端的我，在后台的处理业务能力也有了很好的发展，从使用Node.js搭建服务器，向其发送ajax请求，到cors实现跨域资源共享，解决同源策略阻碍问题。在最终考核的服务器搭建过程中，我更是学到了很多逻辑思维，很多后台问题的解决办法，积累了一定的经验。

**在意志磨炼方面**，谁无暴风劲雨时，守得云开见月明，在选择加入训练营，选择为自己的目标奋斗的那一刻，我就已经做好了打硬战，持久战的准备，每天在学科任务完成之余，我会抓紧一分一毫的时间完成训练营的作业，常常伴随着几声虫鸣持续到深夜，有人会问这样坚持的目的是什么，我觉得这不仅仅是对知识应用上的一种考验，更是对意志精神上的一种挑战，通过不懈的坚持，

**在兴趣培养方面**，训练营激发了我更强的兴趣欲望，我不断汲取陌生的知识与更新的技术，渴望将自己的项目做完美，做出色。在前端的学习中，我发掘了许多社区中新颖漂亮的组件设计，了解了许多大前端前沿的最新资讯，对高质量轻量化的页面布局与动画创作感到满腔热爱

**在人际交流方面**，很高兴通过QG认识了许多优秀的师兄们，他们为我们分享了很多经验，帮助我们解决了很多技术性的问题。也很高兴认识了许多志同道合的朋友伙伴，在训练营中常常需要自主地完成项目任务，并且不断地进行代码审查和反馈。通过这个过程，我学会了如何与他人合作和沟通，学会了倾听他人的意见和建议，并且尊重他们的想法。在难题上，我们需要经常互相询问讨论解决方案，我体会到了沟通过程中的重要性，只有互相理解、互相支持才能更好地完成任务。

其次，在代码审查和反馈环节中，我也学会了如何提供清晰、准确的反馈，同时也学会了如何接受他人的批评和指导。通过这个过程，我认识到了交流中的双向性，不能只关注自己的观点，而是应该站在对方的角度思考问题，积极寻求共同点。

总之，参加前端训练营对我来说是一个很好的人际交流机会。通过与导师合作、沟通和反馈，我逐渐学会了如何与人交流、如何与人合作。这些技能将在我的未来学习和工作中发挥重要的作用，让我能够更好地与他人合作、完成任务，并且提升自己的专业技能，为将来团队的合作打下夯实基础。

**在难题处理方面**，我认识到下列规范的重要性：

**注释与模块化的重要性**

注释可以帮助开发者更好地理解代码，提高代码可读性和可维护性。通过为代码添加注释，可以让其他人更容易地理解代码的意图、结构和逻辑。同时，注释还可以帮助开发者自己回顾代码，减少因为长时间不接触某段代码而导致的忘记或犯错的情况。

模块化是将一个大的项目拆分成较小的模块来实现的过程。这种方式可以使得程序结构更加清晰，减少耦合性，提高代码的可重用性和可维护性。在模块化编程中，一个模块只需关心自己的功能，而不需要考虑其它模块，这样可以更方便地进行单元测试和调试。

总的来说，注释和模块化都是开发高质量软件的必要手段。通过良好的注释和模块化，可以提高代码的可读性、可维护性和可扩展性，从而更好地满足用户的需求。

**程序阶段性测试的重要性**

程序阶段性测试是指在程序开发过程中，对每个重要的阶段进行测试和调试，以确保程序的质量和功能符合预期。这种测试可以帮助程序员及时发现和修复潜在的问题，避免在后期出现难以解决的 bug。

**F12网页开发者工具的重要性**

F12网页开发者工具是一种在浏览器中内置的开发工具，它提供了许多功能，如调试 JavaScript、查看 DOM 和 CSS 属性等。这种工具可以帮助开发人员轻松地分析和调试网站，并提高开发效率，减少时间和非必要输出成本。

**辅助开发插件的重要性**

辅助开发插件是一种可以帮助开发人员提高编码速度和准确性的工具。这些插件可以自动完成某些常见的代码片段，避免手动编写出错或耗时的代码，并提供其他实用的功能，如自动格式化代码、管理依赖项等。

**结构，逻辑再到样式的重要性**

在前端项目搭建中，从结构开始写、然后逻辑、最后样式的顺序是非常重要的，原因如下：

结构先行可以更好地规划整个项目的框架。在编写结构时，开发人员需要思考如何组织 HTML 元素以及各个元素之间的关系，这有助于形成清晰的项目结构，并确保代码能够良好地扩展和维护。

逻辑与结构相辅相成，编写逻辑代码可以更好地实现页面功能。在编写逻辑代码时，开发人员可以按照设计要求实现各种功能，例如表单验证、响应式设计等。

样式通常是最后一个步骤，因为它通常是最容易修改的部分。如果在设计过程中首先处理样式，则可能会导致结构或逻辑上的问题，从而导致调整工作量的增加。

总之，从结构开始写、然后逻辑、最后样式的顺序是一种合理的开发流程，它有助于提高项目的可维护性和可扩展性，同时也有助于提高开发效率。

**思维 具体 完整 可拓展的重要性**

在项目搭建过程中，还不是特别习惯面向对象的思考方式，总是用面向过程的思维去解决问题，写到一半才会瞬间发觉用面向对象的思维解决这个问题会更简单。

而且在某个具体模块的实现上，思维总是不能很好地处理一些细节末枝的东西，不能善始善终，如果有新的需求添加或者要对模块的产出进行加工时，往往就会出现思维短路的情况，不能清晰地做进一步的处理，导致久积成患。

**文档查阅，不断学习**

前端查阅 JavaScript 代码文档的重要性是非常大的，因为 JavaScript 是一种非常灵活和功能强大的编程语言，有很多不同的 API 和库可以使用。通过查阅文档，开发人员可以更好地了解各种 API 的用法、参数和返回值，并通过这些知识更好地构建复杂的应用程序。

另外，熟悉基本组件（如搜索框等）的搭建也非常重要。这些基本组件通常是网站或应用程序的核心组成部分，因此开发人员需要对它们的实现方式非常熟悉。通过熟悉基本组件的搭建方式，开发人员可以更快速地创建出相应的组件，并且对于调试和优化也可以更加得心应手。

在实践中，前端开发人员应该坚持不断学习和探索最新的技术和工具，例如 React、Angular 等流行的前端框架，以及其他的第三方库和工具。同时还应该着重学习 HTML、CSS 和 JavaScript 的基础知识，包括 DOM 操作、事件处理、数据类型和函数等内容。只有通过深入学习这些基础知识，才能够真正成为一名高效的前端开发人员。

**最后，想谈一谈什么是一个好的项目**

**首先，我认为一个好的图书管理系统需要具备以下几个方面的特点：**

功能齐全

一个好的图书管理系统必须要有完备的功能，包括图书信息管理、读者管理、评论与反馈、统计与报表等模块。这些模块相互之间应该能够协同工作，使得整个系统更加高效。

界面友好

一个好的图书管理系统需要有良好的用户界面设计，让用户能够在使用时轻松快捷地完成操作。对于普通用户来说，简单易懂的界面可以提高用户的满意度。

数据安全

由于图书管理系统存储了大量的个人信息和敏感数据，因此必须要保障数据的安全性。系统应该采用合适的加密算法和安全机制，确保数据不被非法获取、篡改或泄露。

**在实际开发中，应该遵循以下几个核心原则：**

定期迭代

作为一个长期运营的软件产品，图书管理系统必须要不断更新迭代。我们需要不断根据用户反馈和市场需求进行优化，加入新的功能模块，提高系统效率。

强调代码规范

一个好的图书管理系统需要有清晰、简洁、可读性强的代码结构。为此，我们在开发过程中始终遵循了一些代码规范，如命名规范、注释规范等。

不断学习

作为开发者，我认为不断学习是非常重要的。我们需要时刻关注新技术、新思路，不断学习并尝试应用到实践中。这样才能保持我们在开发中的竞争力。

总的来说，开发一个好的图书管理系统并不是一件容易的事情，它需要我们有良好的团队协作精神，需要我们不断努力和追求卓越。但只要我们能够在上述原则和特点的基础上进行开发，相信肯定可以开发出一个满足用户需求的优秀系统。