河北师范大学

毕 业 论 文

|  |  |
| --- | --- |
| **题 目：** | 基于web前端的书籍推荐平台的实现 |
| **学 院：** | 软件学院 |
| **专 业：** | 软件工程 |
| **班 级：** | 2016级4班 |
| **姓 名：** | 李建辉 |
| **学 号：** | 2016011494 |
| **指导教师：** | 王勇 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 年 |  | 月 |  | 日 |

毕业论文诚信承诺书

1. 本人郑重承诺所提交的毕业论文《基于web前端的书籍推荐网站的实现》，是在指导教师的指导下，严格按照学校和学院的规定完成的。
2. 本人在毕业论文中引用他人的观点和参考资料均加以注释和说明。
3. 本人承诺在毕业论文选题和研究过程中没有抄袭他人研究成果和伪造相关数据等行为。
4. 在毕业论文中对侵犯任何方面知识产权的行为，由本人承担相应的法律责任。

学生签名：

日期：

**基于web前端的书籍推荐网站的实现  
【摘要】**随着web开发技术的不断更新与发展，我们如今的生活越来越方便。如何将web技术应用结合并应用于网站的开发与我们的日常生活中常见问题，是我们研究的问题。

本网站采用前后端分离的原则，使用MySQL数据库，将数据通过数据库和后台结合，再呈现到前端页面，通过Ajax请求在前后端传送数据；具体到网站的内容，就是结合自身兴趣搜索关键词网站会推荐出书籍，并将书籍的内容介绍与影视资料视频结合在一起，查询书籍的同时，如果有影视资料的匹配，那么也会有相关影视资料的展示；若书籍中的经典语句或者对白都会通过这个网站展示出来。

这个网站的意义在于推荐高质量的书籍或者小说，避免打开书城的时候有很多没有营养的快速文学，不知道该选择哪本书。有这个网站之后，我们就能够直接到书城搜索自己想看的书，不用被其他书籍干扰。

**【关键词】**前后端分离 MySql 书籍推荐与影视资料

**The implementation of book recommendation platform based on Web-Front-end**

**【Abstract】**As the web technology advanced,we live in a more convenient world. The question that how to combine the technology and our life and then finish a website is the purpose of my research.

My website use the model of the front and bank department, and use MySql database. Combined the back and database to get the data , then use Ajax to show the data in the website,which is my work. When it comes to the content, I will recommend some book you will be fond of according your keywords. We also will show you the video that connected with the book you search for .That’s mean when you find a book,we will show you some video if the book have films version. Besides, if the book has classical words, it will show you.

The meaning of my website is that we can filter the useless information when we want to search our desired book. When we open our Book City, it often has many things we don’t want to see and we don’t how to choose . We can use the website find out which book we are preferred and avoid to be bothered by other books.

**【Key Words】** department of the front and back MySql book recommendation and video

**目 录**

[摘要](#_Toc8754805)

[一、网站建立的目的及意义](#_Toc8754806)

[二、网站的设计](#_Toc8754807)

[（一）网站的UI设计](#_Toc8754808)

[（二）网站数据库配置及表格设计](#_Toc8754809)

[（三）网站语言，框架的安排](#_Toc8754810)

[（四）网站功能模块的设计](#_Toc8754810)

[三、数据的传送](#_Toc8754811)

[四、基于web前端书籍推荐网站的实现](#_Toc8754816)

[（一）登录模块的实现](#_Toc8754817)

[（二）注册模块的实现](#_Toc8754818)

[（二）用户收藏模块的实现](#_Toc8754818)

[（二）书籍推荐模块的实现](#_Toc8754818)

[结论](#_Toc8754819)

[参考文献](#_Toc8754820)

[致谢](#_Toc8754821)

# 一、网站建立的目的及意义

1.1书籍推荐网站的建立背景

在互联网技术十分发达的今天，一切行业和店铺都趋向于线上模式，我们很少去书店买纸质书籍，因此“快餐式”的线上阅读迎合了大部分人的需求。在这种大环境下，各种电子书籍网站和app层出不穷，但是这些网站或者app首页往往是一些内容质量并不高的书籍，这也导致我们的选择阅读书籍的时候往往受到误导，并非自己本意地去读一些书籍，浪费时间却没有收获。

1.2书籍推荐网站的现实意义

时代和技术进步的同时，确实引发了其他的问题，让我们在享受生活的同时无法忽视其负面影响。具体问题有：第一，信息良莠不齐，是互联网时代发展的副作用产物，而我们怎样避免或者减轻这个问题，是我们需要进一步探索的意义；第二，我们总是在手机冲浪的时候就完全忘记了时间甚至忘记了打开智能设备的目的，等回头发现已经浪费了很多时间在无关的事上面；第三，在这个追求效率的时代里，花时间在无用事物上消耗人力物力，得不偿失。

正是由于以上原因，我结合自己的兴趣，想要做一个书籍推荐网站，通过这个网站，用户可以直接查到自己想要读的或者趋向去读的书籍，不用浪费时间在那些首页的书籍，提升效率。基于现在已经很成熟的web技术，完成这样一个网站需要经过需求分析、网站设计、代码实现以及网站测试这些环节。

# 网站设计

## （一）网站的UI设计

网页的整体页面设计，分为一级页面、二级页面和三级页面：其中一级页面有首页；二级页面包括登录页面、注册页面、青春读物页面、经典句子页面、句子收藏页面；三级页面有书籍详情页面；其中每个页面中有不同的图标/文字作为引导，js点击事件切换并进入不同页面。

在网站的UI设计层面，我首先选了主题色。出于希望自己的网站比较轻快明亮一些，我选择了淡紫色作为网站的主题颜色，RGB表示为rgb(236, 173, 205)；其次，因为网站是一个推荐网站，我选择了圆圈和爱心作为网站的logo，寓意是找自己比较喜欢的书籍。在整体视觉上，由于占满整个屏幕的网页不太好看，我选择了整个网页宽度占屏幕宽度的90%，并且水平居中，css的margin：0 auto；可以解决这个问题，这样的布局使整个页面看起来比较舒服。

具体页面的介绍如下：

首页是整个网站的门面，含有导航栏、轮播图以及热门书籍的呈现。导航栏中搜索框可以搜索关键字来找出自己可能喜欢的书籍，关键字可以是作者、书籍类型或者书名；其次，登录和注册按钮也设置在其中；轮播图采用js完成，每隔三秒触发一次切换图片的事件；轮播图右边有一个每日书单，用于推荐当日书籍，这个一般是根据个人搜索历史和类似书籍的品味设置的；热门书籍的呈现则是直接从数据库拿到的数据，从中选取搜索频率较高的书籍，在首页推荐给网站用户。如下图：

注册页面简洁明了，页面中央是需要提交的表单项，包括用户名、密码、手机号和邮箱的填写项，最下方会有一个提交按钮，负责将信息传送到后台，实际上注册完成之后也能够完善个人信息，比如性别；

如图：

登录页面采用模态窗口的形式，弹出含有登录选项的填写页面，可以通过填写正确的用户名/手机号和密码验证成功登录；



推荐书籍详情页是根据用户已经输入的关键词筛选符合词语的书籍，并且如果有相关影视资料，也会呈现在页面上；这个页面主要通过前后端交互实现，页面基本延续首页中热门书籍的样式，补充了内容简介和相关视频。

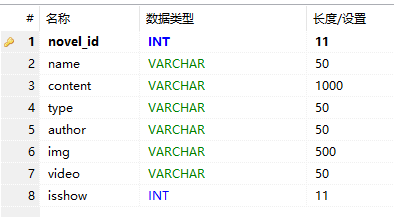
经典句子收藏页面是我们将在页面上呈现优质书籍中广为流传的经典语录或者段落，而用户可以根据自己的喜好来进行收藏；用户在登录状态下就可以看到自己收藏的句子。

## （二）网站数据库的配置及表格设计

网站使用mysql数据库，基于Ubuntu的服务器环境，我在安装完MySQL并且设置好密码之后，进入数据库并授权，这样在客户端可以使数据库图形化；接下来完成本网站需要数据库的配置，下面所有操作皆由SQL语句完成，首先创建自己这次要使用的book数据库，在book数据库下建立要用的表格，对应本次网站的页面设计，我首先建立了用户表，书籍表，书籍类型表，收藏表，具体如下：  
用户表，包含用户名、密码、电话号码、邮箱、性别字段，其中username是主键；



书籍表，包含书籍id、名字、内容简介、书籍类型、作者、封面图片、视频以及是否展示字段，其中novel\_id作为主键；



句子表，包含句子id，句子内容，书的名称，其中juzi\_id作为主键；  
  
句子收藏表，包含用户名、句子的id;



创建完了数据库应该具有的基础表格和字段之后，基于nodejs平台，我在数据库与express框架之间运用mysql的createPool方法创建连接池，完成了后台与数据库的连接。这样通过express起后台服务，能够直接访问数据库，通过SQL语句对book数据库表格中的数据进行增删改查操作，也就可以更好的处理http请求。

## （三）网站语言、框架的安排

总体来说，在使用的语言层面，网站采用前后端分离的原则，使用HTML5，CSS3，和ionic前端框架，nodejs中express框架作为后台服务，以及MySQL数据库。

其中，我选择ionic是因为框架中有许多天然的优势，比如有好看又成熟的UI组件，封装了很多服务模块，包括http模块，通过它们可以很方便地请求后台数据和发送数据到后台。选择nodejs的express框架作为后台因为其routes的应用，不同模块对应不同功能，很适合作为网站的后台。在我看来，如果我单独把每个功能都放在一个独立的js模块中，那么如果在开发过程中出现问题或者代码未完成，我很容易根据自己设定的名称或代号找到相关代码，只要对指定的代码进行优化处理就可以了。

从软件过程层面来讲，我采用的是增量过程模型，根据优先级的不同，依次完成不同阶段的增量：第一个增量中完成了关键字查询书籍以及相关影视资料的功能，这也就整个网站的核心功能，也作为第一个版本；第二个增量是在第一版的基础上完成首页的页面布局以及各个页面跳转，自然也包括各个页面的设计实现；第三个增量版本完成了网站的登录、注册功能；第四个增量中完成了经典句子在前端页面的呈现以及收藏句子的功能；整体的增量过程之上，采用迭代的过程，在实现的同时将各个功能更加细化，去除不合理的环节，使用户使用环境更友好，功能更加完善。在整个过程中，我划分任务集的优先级方法就是对我想要实现的功能进行排序，明确在需求分析基础上各个功能的重要程度，功能越重要，优先级越高。

## （四）功能模块设计

本着“高内聚、低耦合”的原则，整个网站软件系统分为注册、登录、关键字查询推荐书籍、查看书籍介绍、收藏句子的功能模块，模块内部紧密联系，而模块与模块之间依赖性较低。如果有某个功能模块出现问题，可以最大程度地减少其他模块和整个项目的损失，能够有目的地解决问题。  
**三、数据传送方式**  
数据传送方式，如何将数据在前台和后台之间传送  
以首页为例，我是将数据库的内容渲染到了前端页面：  
后台代码如下：

var express = require('express');

var router = express.Router();

const db = require('../model/database');

router.get('/', function(req, res) {

res.header('Access-Control-Allow-Origin','\*');

res.header('Content-Type','text/plain; charset="utf-8"')

const sql = 'select \* from novel';

db.query(sql,(err,result)=>{

res.send(result);

});

});  
module.exports = router;

第一步，采用express的路由模块，运用sql语句直接获取到了表格中所有条项，然后用回调函数将获取得到的结果发送到前台，result整体是一个数组，数组中包含若干对象。实际上在浏览器访问服务器地址也能达到这个效果。第二步，在完成了跨域请求代理配置之后，按照后台逻辑，在前端设置同样的接口，请求到result中的数据并将数据赋值给已经设置好的变量，如下部分代码：

this.http.get('/').subscribe(data=>{ this.novel=data; });

第三步，运用变量和angular中的数组ngFor，将所有数据呈现到页面上，配合使用css使样式美观。部分代码：

<ul \*ngFor='let item of novel'>

<li>

<div class="book-img">

<a href="#" ><img src="{{item.img}}" /></a>

</div>

<div class="book-info">

<a><h4>{{item.name}}</h4> </a>

<p>{{item.content}}</p>

</div>

</li>

</ul>

类似于首页，其他页面也是采用相同的方法来实现数据交互。涉及到表单填写并提交到后台时，具体问题具体分析，应当采用post请求。

# 四、基于web前端书籍推荐网站的实现 （一）注册模块的实现

1.1注册功能流程  
 输入用户名、手机号、密码、以及密码确认；是否符合填入规则，如果符合规则，那么将各个项写入数据库中；否则提示用户应该输入满足条件的数据。

如下图：

输入必填信息

N

验证规则

Y

信息写入数据库

1.2用户注册  
 用户提交数据校验的规则如下：手机号必须为11位数字，且以1开头，利用正则表达式^[1][3,4,5,7,8][0-9]{9}$ 验证；确认密码和密码必须填写一致；邮箱必须符合格式，验证代码如下：

myreg=/^[1][3,4,5,7,8][0-9]{9}$/;

emailreg=/\w+[@][a-zA-Z0-9\_]+(\.[a-zA-Z0-9\_]+)/;

go(username: HTMLInputElement, password: HTMLInputElement,psw:HTMLInputElement,email:HTMLInputElement,tel:HTMLInputElement){

console.log(username.value);

if (username.value == '') {

alert("请输入账号");

} else if (password.value == '') {

alert("请输入密码");

}else if(password.value!=psw.value){

alert("请保证两次密码一致");

}else if(!this.myreg.test(tel.value)){

alert("清输入合法的手机号");

}else if (email.value == '') {

alert("请输入邮箱");

}else if (!this.emailreg.test(email.value)) {

alert("请输入正确邮箱");

}else{

let userinfo: string = '用户名：' + username.value + '密码：' + password.value + '邮箱：' + email.value ;  
let a={

username:username.value,

password:password.value,

email:email.value,

tel:tel.value

}

this.http.post('/zhuce',a,{

headers : this.headers,

observe : 'body',

responseType : 'json'

}

).subscribe(data=>{

console.log(data);

});

this.navCtrl.push(LoginPage);

}

};

注册用户代码zhuce.js：

router.post('/', function(req, res) {

res.header('Access-Control-Allow-Origin','\*');

res.header('Content-Type','text/plain; charset="utf-8"');

var username=req.body.username;

var password=req.body.password;

var email = req.body.email;

var tel=req.body.tel;

console.log(username,password,email,tel);

const sql='insert into user(username,password,tel,email) values(?,?,?,?)';

if(username && password && email && tel){

db.query(sql,[username,password,email,tel],(err,result)=>{

if(err){

console.error("Error:",err);

process.exit();

}

console.log(result);

res.send(result);

});

}

## （二）登录模块的实现

2.2用户登录  
用户输入用户名/手机号、密码；后台代码根据查询数据库是否有该用户，如果有该用户且密码正确，那么成功登录，并且按照登录用户的数据重新渲染页面；如果密码不正确，则提示用户。

流程图如下：

输入信息

N

验证输入信息是否正确

Y

显示含有用户信息的界面

用户登录的另一个功能是之前收藏的内容可以在登录时一下呈现出来，其中涉及到本地存储的localstorage，在登录时将用户名存储，之后其他页面使用此用户名，也就能使用户获取自身的相关信息。

登录代码login.js：

router.post('/', function(req, res) {

res.header('Access-Control-Allow-Origin','\*');

res.header('Content-Type','text/plain; charset="utf-8"');

var username=req.body.username;

var password=req.body.password;

const sql = "select \* from user where username = '"+ username +"'and password = '"+password+"' ";

db.query(sql,(err,result)=>{

if(err){

res.send('查询失败：'+err);

}else{

res.send(result);

}

})

**（三）用户收藏模块的实现**  
经典句子收藏功能

描述：用户观看界面的经典句子/语录时，如果喜欢可以点击收藏，下次登录时会自动显示在用户页面中。点击收藏时，系统会随着用户本身的状态来调整页面图标的颜色。

关键部分代码及说明：

在点击到经典句子页面时，为了显示正确的颜色，我为从后台获取的数组增加了flag属性，以便于标记句子的状态（用户此时是否收藏）。随着点击事件的触发，网站要立即做出颜色变化反应，切换句子是否收藏的状态。整体的解决方案为嵌套两层请求，第一次get请求获取到数据库中所有句子信息，内层请求发送post请求获取到数据库中用户收藏的句子信息，最后将两层获取的数据进行循环判断，将重合部分添加flag属性为true。

获取正确的信息：

this.http.get('/juzi').subscribe(data=>{

this.juzi=data;

this.username = localStorage.getItem('username');

let a={username:this.username};

this.http.post('/shoucang/content',a,{

headers : this.headers,

observe : 'body',

responseType : 'json'

}).subscribe(data1=>{

this.scjz=data1;

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

for(var j=0;j<this.scjz.length;j++){

if(this.juzi[i].juzi\_id==this.scjz[j].juzi\_id){

this.juzi[i].flag=true;}

}

}

});

});

点击收藏模块js部分代码：

shoucang(index){

this.flag=this.juzi[index].flag;

if(!this.flag){ this.flag=false; }

let a={username:this.username,juzi\_id:index};

//每点击一次，应该重新获取状态，使flag正确标识图标

if(!this.flag){

this.http.post('/shoucang/insert',a,{

headers : this.headers,

observe : 'body',

responseType : 'json'

}

).subscribe(data=>{

console.log(data);

});

this.juzi[index].flag=!this.flag;

}else{

this.http.post('/shoucang/delete',a,{

headers : this.headers,

observe : 'body',

responseType : 'json'

}).subscribe(data=>{

console.log(data);

});

this.juzi[index].flag=!this.flag;

}

};

Express后台代码：

router.post('/insert', function(req, res) {

res.header('Access-Control-Allow-Origin','\*');

res.header('Content-Type','text/plain; charset="utf-8"');

var username=req.body.username;

console.log(ifsc);

const sql = 'insert into shoucang(username,juzi\_id) values(?,?) ';

db.query(sql,[username,juzi\_id],(err,result)=>{

if(err){ console.log(err); }

res.send(result);

});

});

router.post('/delete',function(req,res){

res.header('Access-Control-Allow-Origin','\*');

res.header('Content-Type','text/plain; charset="utf-8"');

var username=req.body.username;

var juzi\_id=req.body.juzi\_id;

const sql = 'delete from shoucang where username=? and juzi\_id=?';

db.query(sql,[username,juzi\_id],(err,result)=>{

if(err){ console.log(err); }

res.send(result);

});

});

## （四）关键字查询书籍模块的实现

流程：用户输入作者/书籍类型，网站会推荐出最匹配的书籍，并且显示在他的界面上；其详细步骤是，用户点击查询图标时，触发了点击事件，js部分点击事件的作用是将输入信息传送到后台nodejs服务；后台中使用post方法得到前端传送的数据A，根据A的内容，对数据库进行select筛选查询，选出符合A的书籍，并将结果（是一个数组，数组里面包含对象）发送至前端；前端对数组中对象进行解析，同时运用框架的双向数据绑定完成前端页面的渲染。

将查找内容作为参数传递到novel页面：  
search(type:HTMLInputElement){

this.navCtrl.push(NovelPage,{type:type.value});

}

NovelPage获取参数并查询：

this.type=this.navParams.get('type');  
let a={type:this.type};

this.http.post('/novel/type',a,{

headers : this.headers,

observe : 'body',

responseType : 'json'

}

).subscribe(data=>{

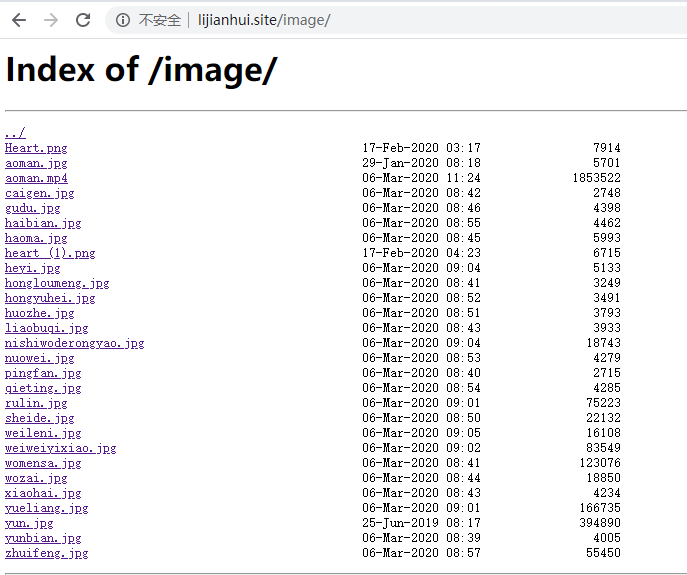
console.log(data);

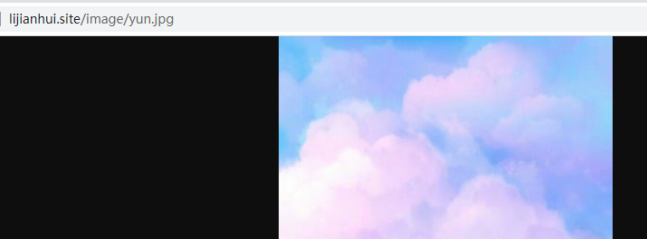
this.novel=data;

});

将推荐书籍与相关影视资料的联系  
描述：用户查出推荐书籍的同时，如果书籍有对应的影视资料，将会自动显示在页面里。

功能实现具体步骤：在后台通过关联书籍表格和影视表格，将对应书籍的信息以及影视资料筛选出来，这样直接在页面显现的时候发送到前端页面。

这个功能实际上是我自己结合自身兴趣，用户中也有一部分人群可能有着类似需求。现如今，越来越多的文学作品被当作素材拍成了影视作品，我们也会希望在看到推荐书籍简介的同时也有关联影视作品的呈现。可在实现过程中，如何获取这些视频也成了我的难题，解决这个问题我花费了一些时间，按照原来计划我直接利用网络资源，使用直接从网络上请求图片/视频的链接地址。但是在写项目过程中，我发现随着时间推移这些链接部分会失效，这也就导致了我的网站软件系统的可用性和生存性并不高。在思考过后，最终的解决方案是先将图片和视频放在我自己的云服务器的指定文件夹里，利用nginx配置中的location进行网站路由的划分，这样通过自己域名对应的网址就可以直接获取到文件资源（包括图片和视频），在数据库对应video/img字段直接输入对应网址就可以完成任务。如下图：  




//实现搜索关键字查找出书籍模块代码novel.js：

router.post('/type',function(req,res){

res.header('Access-Control-Allow-Origin','\*');

res.header('Content-Type','text/plain;charset="utf-8"');

var type=req.body.type;

const sql='select \* from novel where type=?';

db.query(sql,[type],(err,result)=>{

if(err){

console.log(err);

process.exit();

}

res.send(result);

})

})

在这次实践中我更加遵循敏捷开发过程，遵循可运行的软件胜过文档的原则。在实现功能时其实我并没有提前完成许多文档，但是当最后完成整个项目时，所有的东西都在过程中有了体现，比如系统需求分析、基于场景的需求模型即用例以及软件质量管理和整体项目管理等。在整个过程中，我不断调整，由于客观原因，我只能逐渐地对原有计划进行变更，在变更中尽力靠近目标愿景。

总体来说，这次毕业设计对于我而言是一次很好的锻炼，既增加了我对各个框架内容的理解，又增加了我的实战经验，同时我发现了自己的不足。在宏观层面上，我最大的收获就是加深了总体对于整个开发流程的了解，具体到项目内部细节，发现自己对于知识的探索之前并没有做得很好，但是这次的毕业设计让我从之前的一知半解到开始抠各种细节，不论是简单的UI设计，还是复杂的后端代码完善，如何达到预想实现的功能，是我一直在追求的。

# 参考文献

[1]软件工程实践者的研究方法

[2]深入浅出nodejs

[3]ES6

# 致谢

一眨眼，我们已经在河北师范大学转眼间度过了四个春夏秋冬。曾经我们在去学院的路上不断抱怨路程太长，可现在终于明白，从宿舍去学院的路仅仅是四年那么长。在这四年里，我收获了经久不变的知识，也学习了随着时代前行而不断进步的互联网技术，最重要的是，我感谢在这四年里遇到的同学和老师。当我步入人生的另一阶段，我希望带着这四年的积累不断前行，找好自己的方向，不断完成一个个目标。