2018-2019学年第2学期

**实 验 报 告**



课程名称:跨平台脚本开发技术

实验项目: 期末大作业

专业班级\_软工1602\_\_

学生学号\_31601381\_\_

学生姓名\_刘晓倩\_\_

实验指导教师:郭铭

目录

[1. 项目主题 4](#_Toc11878674)

[2. 项目地址 4](#_Toc11878675)

[3. 前端功能 5](#_Toc11878676)

[3.1查看主页 5](#_Toc11878677)

[3.2查看电影页 6](#_Toc11878678)

[3.3登录注册页面 6](#_Toc11878679)

[3.4查看电影详细信息 7](#_Toc11878680)

[3.5短评评论页面 8](#_Toc11878681)

[3.6影评评论页面 8](#_Toc11878682)

[3.7查看影评页面 9](#_Toc11878683)

[3.7.1大众影评 9](#_Toc11878684)

[3.7.2专家影评 10](#_Toc11878685)

[3.8个人中心 10](#_Toc11878686)

[3.8.1设置 10](#_Toc11878687)

[3.8.2个人基本信息 11](#_Toc11878688)

[3.9查看收藏 12](#_Toc11878689)

[4. 后台功能 12](#_Toc11878690)

[4.1用户管理 12](#_Toc11878691)

[4.1.1查看网站用户 12](#_Toc11878692)

[4.1.2编辑用户信息 13](#_Toc11878693)

[4.1.3封号、解封用户 13](#_Toc11878694)

[4.1.4后台管理员 14](#_Toc11878695)

[4.1.5查询管理员 14](#_Toc11878696)

[4.1.6专家申请审核 14](#_Toc11878697)

[4.1.7按需查找专家 15](#_Toc11878698)

[4.1.8查看申请详细信息 16](#_Toc11878699)

[4.2电影管理 17](#_Toc11878700)

[4.2.1电影上传 17](#_Toc11878701)

[4.2.2电影信息管理 18](#_Toc11878702)

[4.2.3编辑电影信息 19](#_Toc11878703)

[4.2.4按需查询电影 19](#_Toc11878704)

[4.2.5删除电影 21](#_Toc11878705)

[4.2.6幻灯片管理 22](#_Toc11878706)

[4.2.7添加幻灯片 22](#_Toc11878707)

[4.2.8搜索查找幻灯片 23](#_Toc11878708)

[4.2.9编辑修改幻灯片 23](#_Toc11878709)

[4.2.10替换幻灯片 24](#_Toc11878710)

[4.2.11删除幻灯片 25](#_Toc11878711)

[4.3影评管理 25](#_Toc11878712)

[4.3.1普通影评 25](#_Toc11878713)

[4.3.2查看普通影评 26](#_Toc11878714)

[4.3.3删除普通影评 27](#_Toc11878715)

[4.3.4按需查找普通影评 28](#_Toc11878716)

[4.3.5专家影评 29](#_Toc11878717)

[4.4管理员管理 30](#_Toc11878718)

[4.4.1管理员登录首页 30](#_Toc11878719)

[4.4.2管理员注册 30](#_Toc11878720)

[5. 项目管理实现 31](#_Toc11878721)

[5.1项目目录结构 31](#_Toc11878722)

[5.2主要功能 31](#_Toc11878723)

[5.3解决技术要点问题 31](#_Toc11878724)

[5.3.1前端 31](#_Toc11878725)

[5.3.2后台 40](#_Toc11878726)

[6. 项目评估 50](#_Toc11878727)

[6.1项目分工 50](#_Toc11878728)

[6.2自我评估表 50](#_Toc11878729)

[6.3项目自评等级（1-5） 50](#_Toc11878730)

[7. 心得体会 51](#_Toc11878731)

[7.1 课程感言 51](#_Toc11878732)

[7.3个人小结 51](#_Toc11878733)

[8. 参考资料 51](#_Toc11878734)

# 项目主题

本次大作业是一个电影影评网站，网站包括首页、电影、影评和收藏，还可以查看电影详情，通过注册登录后可以进行评价。

后台进行用户管理，电影信息管理，影评信息管理。

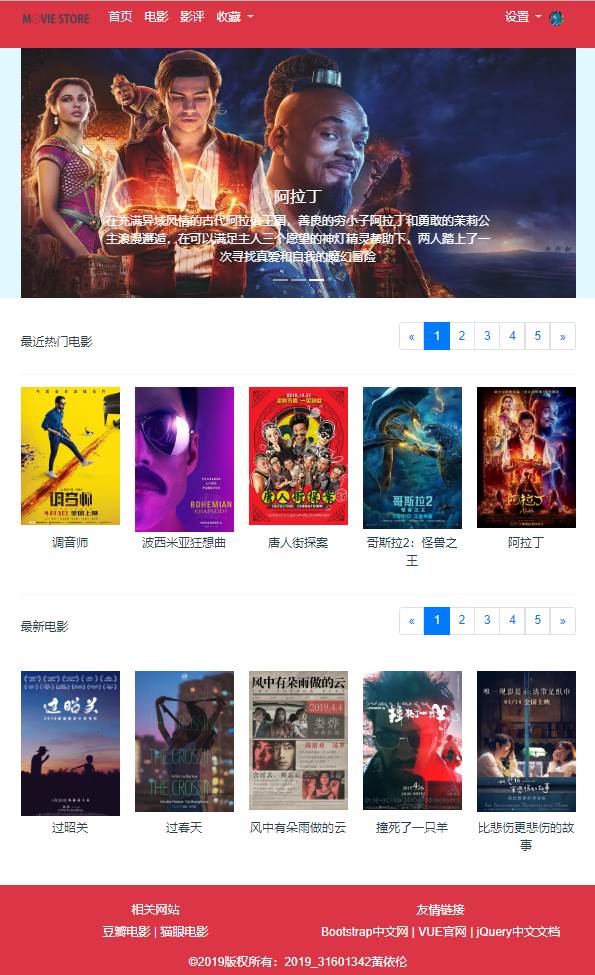
# 项目地址

项目已经公布在git上：<https://github.com/licecream/js>

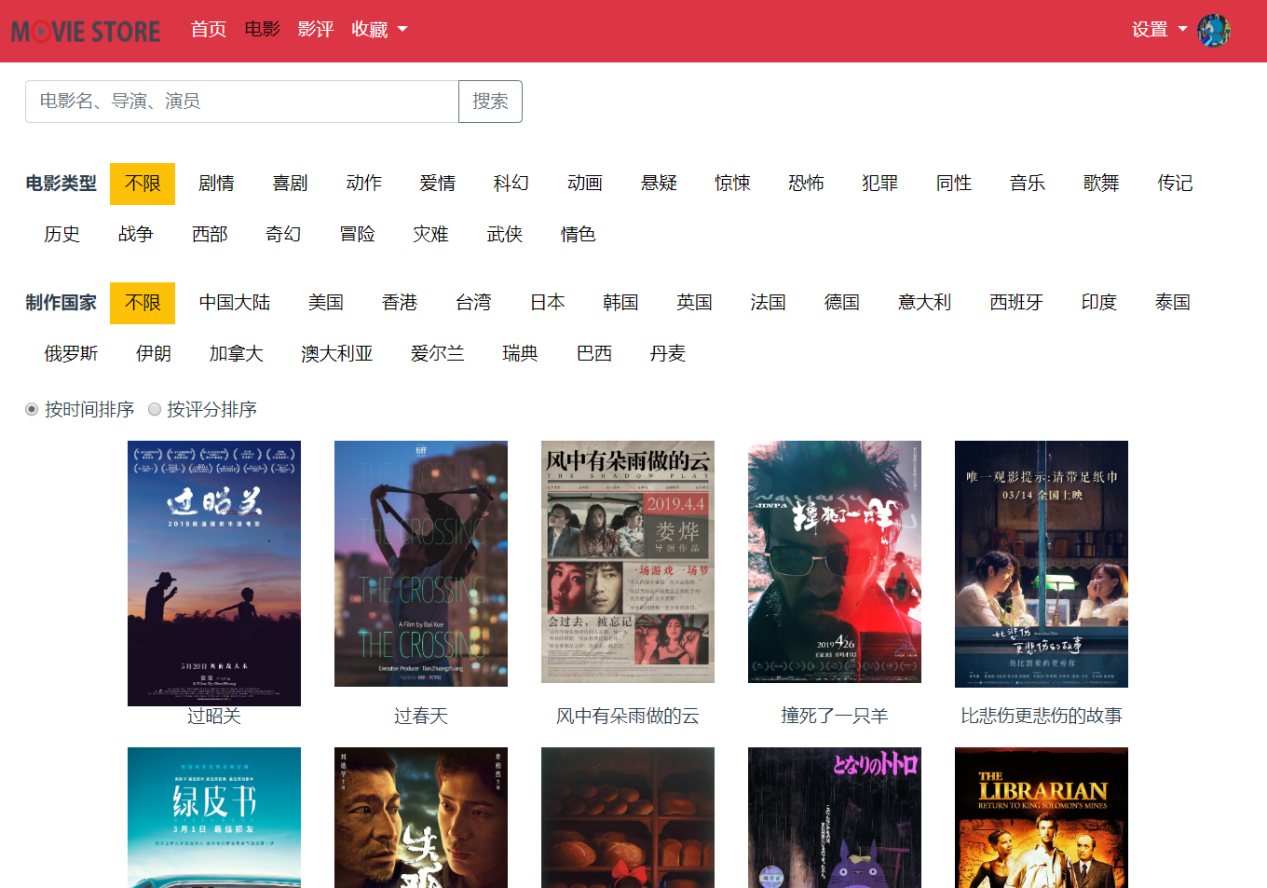
<https://github.com/huizhai/js>

# 前端功能

## 3.1查看主页



## 3.2查看电影页



## 3.3登录注册页面





## 3.4查看电影详细信息

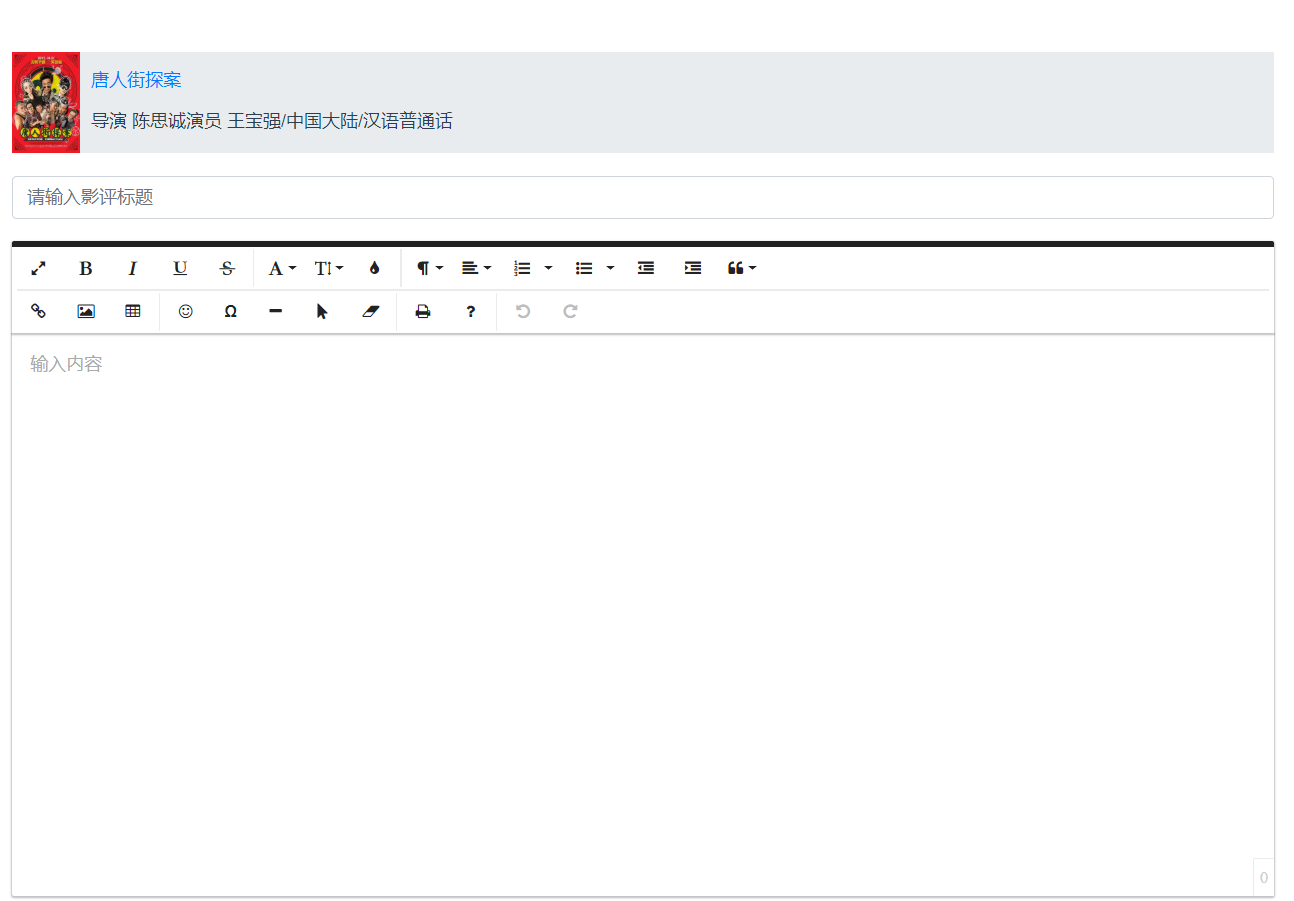
可以通过首页的电影点击或者电影页的电影，点击电影图片即可进入电影详细信息页



## 3.5短评评论页面



## 3.6影评评论页面



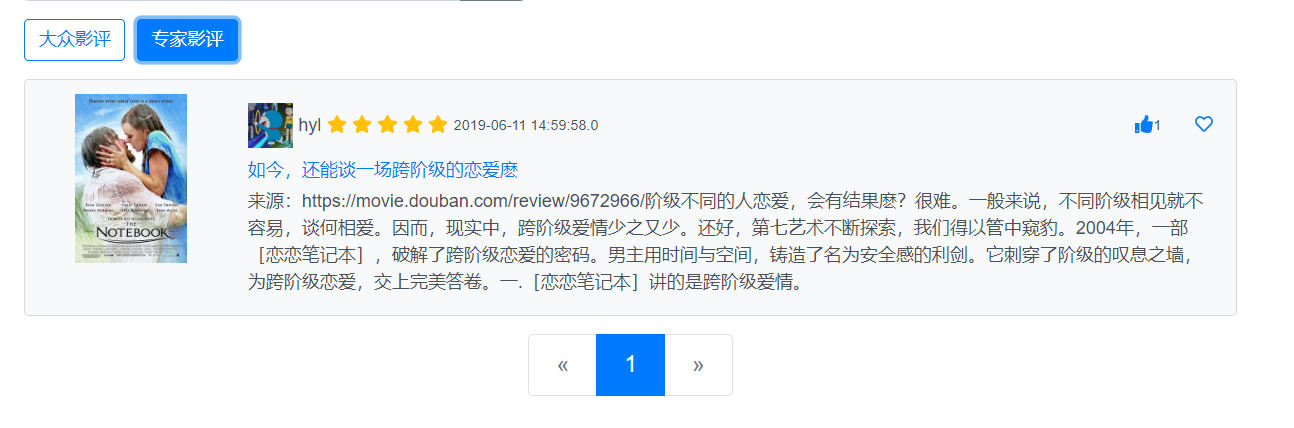


## 3.7查看影评页面

### 3.7.1大众影评



### 3.7.2专家影评



## 3.8个人中心

### 3.8.1设置



### 3.8.2个人基本信息



## 3.9查看收藏



# 后台功能

## 4.1用户管理

### 4.1.1查看网站用户



### 4.1.2编辑用户信息



### 4.1.3封号、解封用户



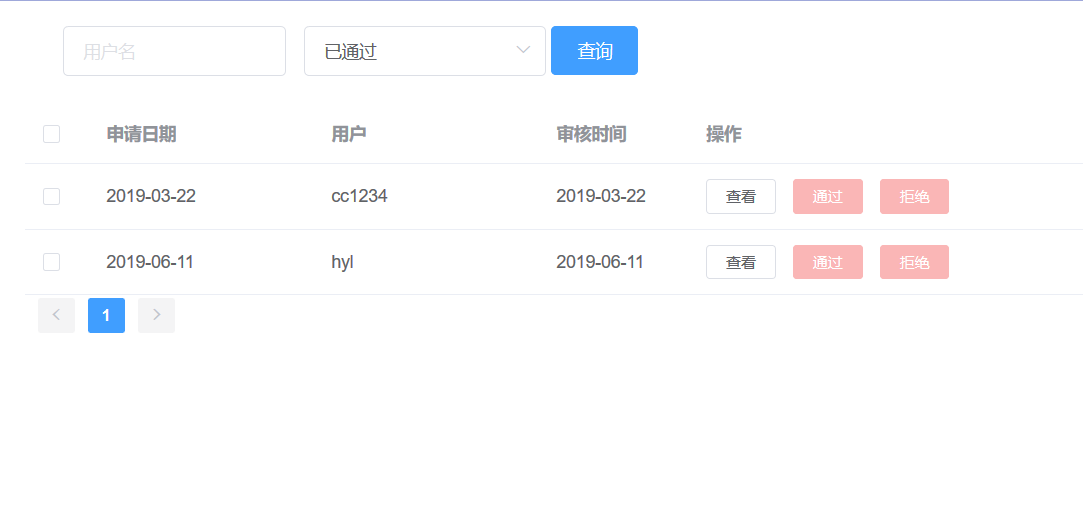
### 4.1.4后台管理员



### 4.1.5查询管理员



### 4.1.6专家申请审核



### 4.1.7按需查找专家

#### 4.1.7.1按用户名查找



#### 4.1.7.2按类型查找



### 4.1.8查看申请详细信息



## 4.2电影管理

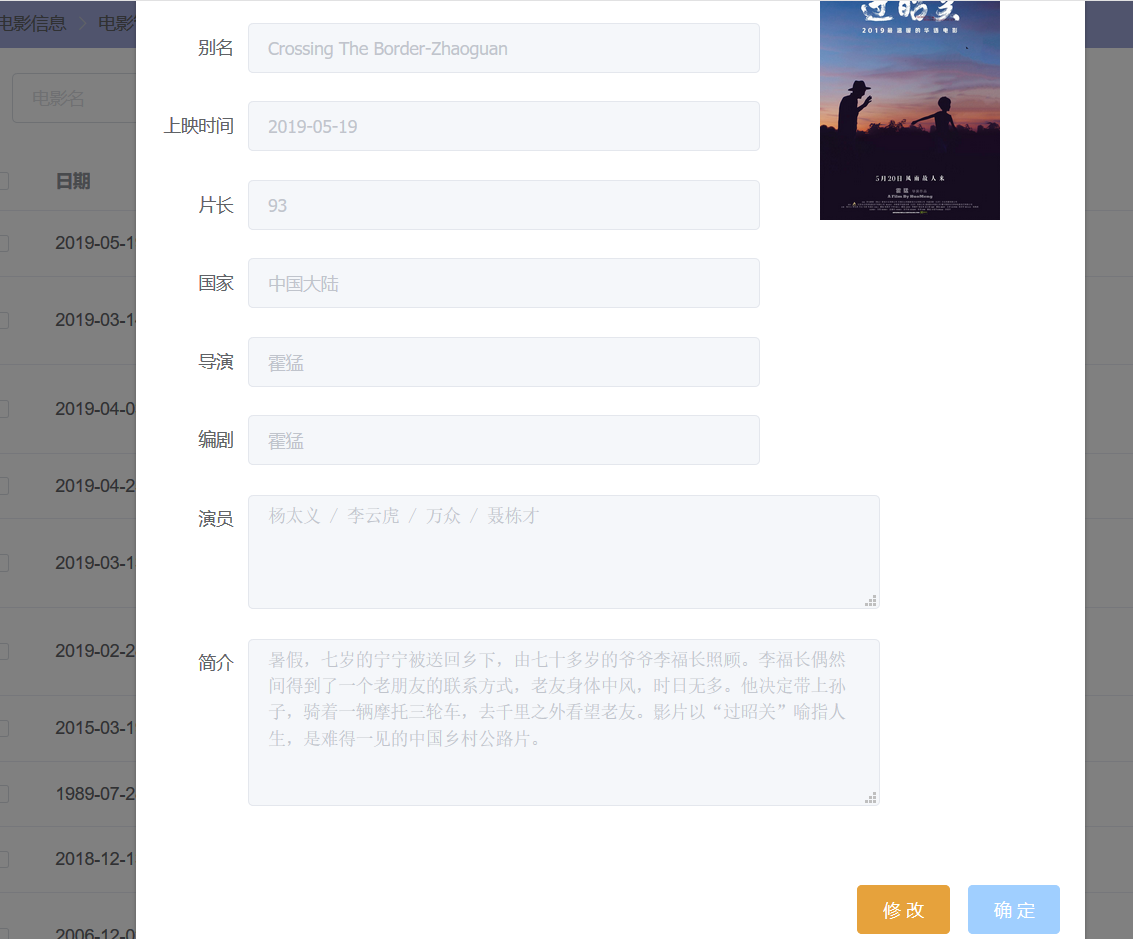
### 4.2.1电影上传



### 4.2.2电影信息管理



### 4.2.3编辑电影信息



### 4.2.4按需查询电影

#### 4.2.4.1按电影名查找



#### 4.2.4.2按电影类型查找



#### 4.2.4.3按国家查找



#### 4.2.4.4按语言查找



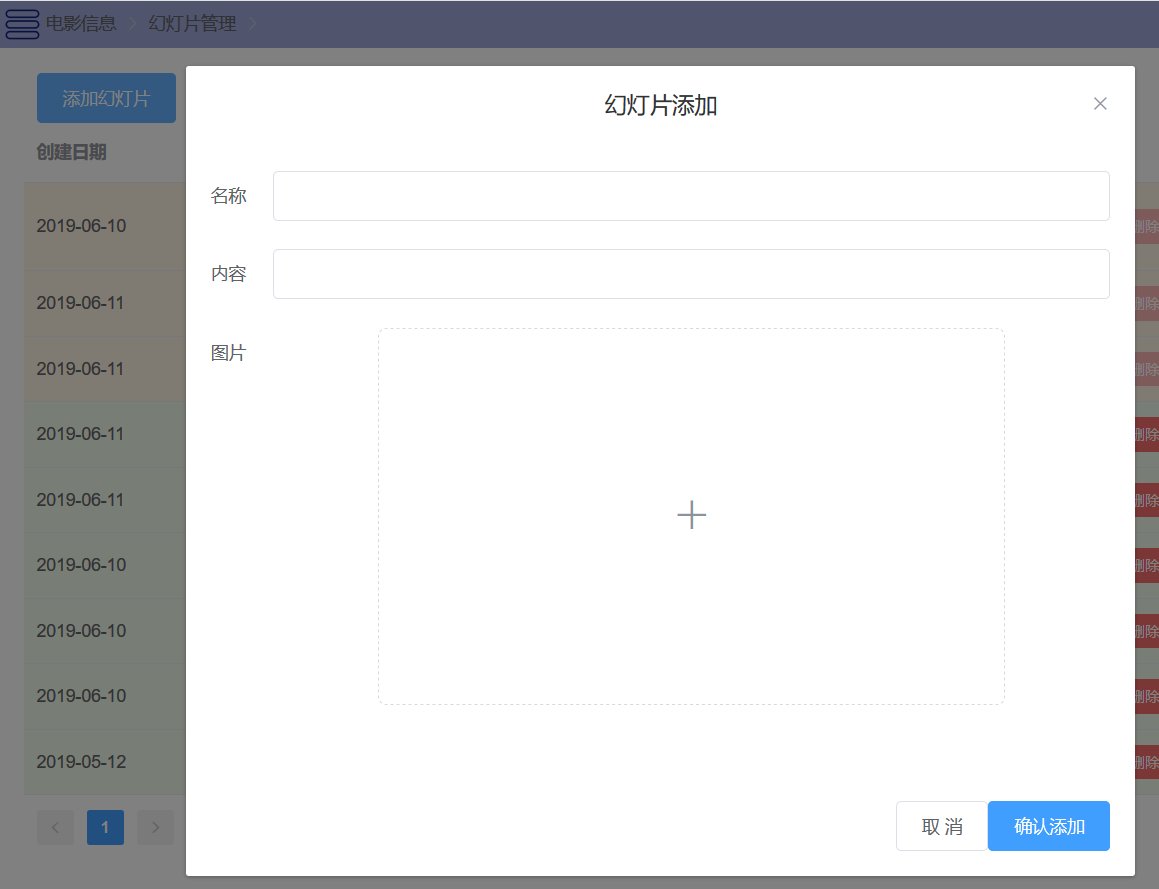
### 4.2.5删除电影



### 4.2.6幻灯片管理



### 4.2.7添加幻灯片

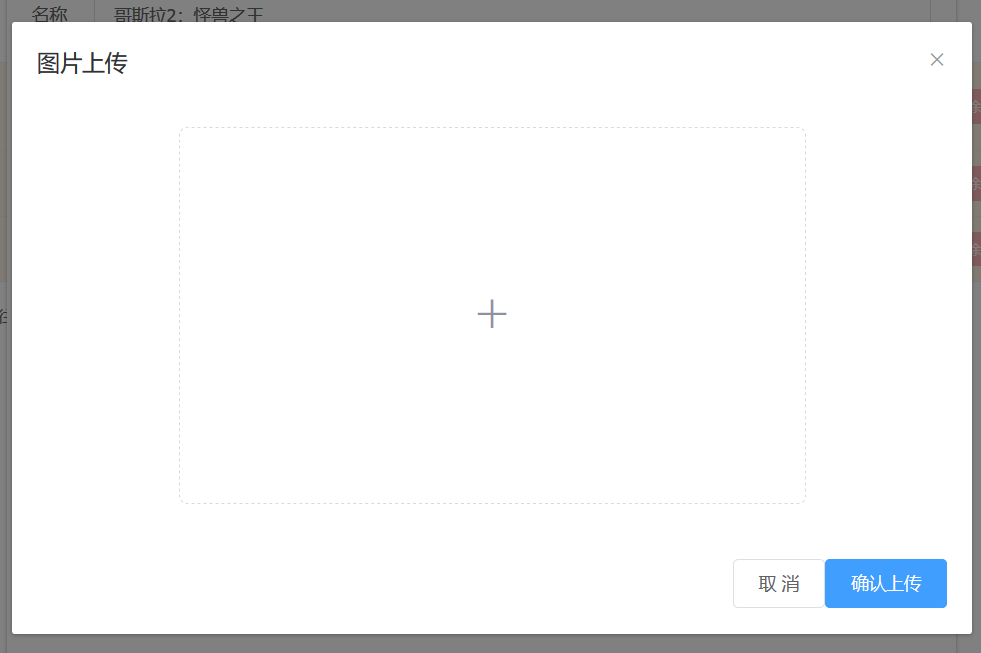


### 4.2.8搜索查找幻灯片



### 4.2.9编辑修改幻灯片





### 4.2.10替换幻灯片



### 4.2.11删除幻灯片



## 4.3影评管理

### 4.3.1普通影评



### 4.3.2查看普通影评

#### 4.3.2.1修改影评



#### 4.2.2.2封号



### 4.3.3删除普通影评



### 4.3.4按需查找普通影评

#### 4.3.4.1按电影名查找



#### 4.3.4.2按用户名查找



#### 4.3.4.3按影评名查找



### 4.3.5专家影评



专家影评其他功能与普通影评相同，不进行重复论述

## 4.4管理员管理

### 4.4.1管理员登录首页



### 4.4.2管理员注册



# 项目管理实现

## 5.1项目目录结构

build：打包发布需要的文件

config：存放项目的配置信息，服务器端口，连接的数据库端口等

src：开发目录

|---assets：资源目录

|---components：存放组件，都是.vue文件

|---router：存放路由文件

|---views：用于渲染的视图文件

|---App.vue:入口组件

|---main.js:启动文件

static：静态资源目录，可以放一些图片，字体之类

index.html:首页入口文件

packge.json: 项目名、描述、作者、依赖等等信息

## 5.2主要功能

用户端：

1. 实现了登录注册退出登录功能。
2. 实现了提交影评和收藏功能。
3. 可以查看所有电影和电影信息。
4. 个人中心可以修改信息、修改头像

后台：

1、登录注册功能

2、进行管理网站用户、后台管理员、专家申请操作：按用户名查询，按类型查询，编辑用户信息，删除管理员，封锁用户账号，解封用户等。

3、上传电影，输入电影详细信息：电影名称、外文名、片长、语言、电影类型、制片国家、上映时间、导演、编剧、主演、简介和电影海报，语言，电影类型。

4、搜索查询电影，管理幻灯片：查询幻灯片状态，幻灯片添加：输入名称、内容和图片

5、影评管理：查询搜索影评（可通过电影名、用户名、影评名）；查看和修改影评内容；可以对用户进行封号；可以对影评进行删除操作

## 5.3解决技术要点问题

### 5.3.1前端

1、解决登录注册退出登录问题

登录：将用户信息用sessionStorage存储token，登录输入的信息不在token里则报错

登出：删除所有数据：sessionStorage.clear();

注册：判断两次输入的密码是否相同，不同则会提示“请确认密码”，如果没有问题则执行注册操作。

login() {  
     this.$refs.login\_button.setAttribute("disabled", "true");  
     this.$axios  
      .post("/login", qs.stringify(this.login\_form))  
      .then(res => {  
         return Promise.resolve(res.data);  
      })  
      .then(json => {  
         this.$refs.login\_button.removeAttribute("disabled");  
         if (json.code === "ACK") {  
           this.user\_name = this.login\_form.username;  
           this.user\_role = json.data.role;  
           if (this.user\_role != "user"){  
             this.$parent.alert("danger", '此账号不是网站用户账号');  
             return;  
          }  
           this.user\_power = json.data.power;  
           this.$parent.alert("success", json.message);  
           sessionStorage.setItem("TOKEN", json.data.token);  
           sessionStorage.setItem("INFO", JSON.stringify(json.data));  
           this.$refs.login\_button.classList.remove("disabled");  
           $("#loginModal").modal("hide");  
           this.$refs.login.setAttribute("disabled", "true");  
           //设置请求头  
           this.$axios.defaults.headers.common[  
             "Authorization"  
          ] = sessionStorage.getItem("TOKEN");  
           this.$options.methods.getUserInfo.bind(this)();  
           this.login\_show = false;  
        } else {  
           this.$parent.alert("danger", json.message);  
        }  
      })  
      .catch(error => {  
         this.$refs.login\_button.removeAttribute("disabled");  
         console.log(error);  
      });  
  },  
   logout() {  
     this.login\_show = true;  
     sessionStorage.clear();  
     window.location.reload();   
  },  
   register() {  
     if (this.register\_form.password != this.register\_form.repeat) {  
       this.$parent.alert("danger", "请确认密码");  
    } else {  
       this.$axios  
        .post("user/register", {  
           username: this.register\_form.username,  
           password: this.register\_form.password  
        })  
        .then(res => {  
           return Promise.resolve(res.data);  
        })  
        .then(json => {  
           if (json.code === "ACK") {  
             $("#registerModal").modal("hide");  
             this.login\_form.username = this.register\_form.username;  
             this.login\_form.password = this.register\_form.password;  
             this.$options.methods.login.bind(this)();  
          } else {  
             this.$parent.alert("danger", json.message);  
          }  
        })  
        .catch(error => {  
           console.log(error);  
        });  
    }  
  },

2、解决提交影评功能

短评:提交短评内容 评分，内容太短少于10 将无法评论成功

submitShortCommentary() {  
     this.$axios  
      .post("short/add/", this.userShortCommentary, {  
         headers: {  
           Authorization: sessionStorage.getItem("TOKEN")  
        }  
      })  
      .then(res => {  
         return Promise.resolve(res.data);  
      })  
      .then(json => {  
         if (json.code === "ACK") {  
           this.$parent.$parent.alert("success", json.message);  
        } else {  
           this.$parent.$parent.alert("warning", json.message);  
        }  
         this.$refs.mark.$refs.modalCancel.click();  
         this.reload();  
      })  
      .catch(error => {  
         console.log(error);  
      });  
  },

之后会在电影页下面实时获得短评记录

getMovieShortCommentarys() {  
     this.$axios  
      .post("short/search/", this.pageHelper)  
      .then(res => {  
         return Promise.resolve(res.data);  
      })  
      .then(json => {  
         if (json.code === "ACK") {  
           this.shortCommentarys = json.data;  
        } else {  
           this.$parent.alert("warning", json.message);  
        }  
      })  
      .catch(error => {  
         console.log(error);  
      });  
  },

3、影评功能

在影评评论页使用了富文本编辑器，首先获取你要评论的电影头像和名称信息，填写影评内容，用户都可以进行普通影评，选择专家会判断你是否为专家，如果都没有问题影评则提交成功被push到影评页。

getMovieSimpleInfo(movieId) {  
     var url = this.GLOBAL.BASE\_URL + "movie/info/simple?movieId=" + movieId;  
     this.$axios  
      .get(url)  
      .then(res => {  
         return Promise.resolve(res.data);  
      })  
      .then(json => {  
         if (json.code === "ACK") {  
           this.movie\_info = json.data;  
           this.movieId = movieId;  
        } else {  
           console.log(json.message);  
        }  
      })  
      .catch(error => {  
         console.log(error);  
      });  
  },  
   submitCommentary() {  
     let url = "long/add/";  
     if(this.commentary.type === 0){  
       url+="normal";  
    }else{  
       if(JSON.parse(sessionStorage.getItem("INFO")).power != "pro"){  
         alert("你不是专家，不能发布专家影评")  
         return;  
      }  
       url+="pro";  
    }  
     this.$axios  
      .post(url, this.commentary)  
      .then(res => {  
         this.$refs.commentaryDialog.$refs.modalCancel.click();  
         return Promise.resolve(res.data);  
      })  
      .then(json => {  
         if (json.code === "ACK") {  
           this.$parent.$parent.alert("success", json.message);  
           this.$options.methods.turnToCommentary.bind(this)();  
           this.$router.push("/commentary")  
        } else {  
           this.$parent.$parent.alert("warning", json.message);  
        }  
      })  
      .catch(error => {  
         console.log(error);  
      });  
  },  
  4、影评页get影评

getCommentary(id) {  
     let url = this.GLOBAL.BASE\_URL + "long/get/id?id=" + id;  
     this.$axios  
      .get(url)  
      .then(res => {  
         return Promise.resolve(res.data);  
      })  
      .then(json => {  
         if (json.code === "ACK") {  
           this.commentary = json.data;  
        } else {  
           this.$parent.$parent.alert("danger", json.message);  
        }  
      })  
      .catch(error => {  
         console.log(error);  
      });  
  }

5、点赞影评

clickLike(commentary) {  
     let flag = commentary.like;  
     commentary.like = !commentary.like;  
     if (sessionStorage.getItem("TOKEN") == null) {  
       if(flag === true)  
         commentary.likeNumber--;  
       else  
         commentary.likeNumber++;  
       return;  
    }  
     var url = this.GLOBAL.BASE\_URL;  
     if (flag === true) {  
       url = url + "long/like/cancel?id=" + commentary.id;  
       commentary.likeNumber--;  
    } else {  
       url = url + "long/like?id=" + commentary.id;  
       commentary.likeNumber++;  
    }  
     this.$options.methods.getRequest.bind(this)(url);  
  },

6、收藏影评

clickCollect(commentary) {  
     let flag = commentary.collection;  
     commentary.collection = !commentary.collection;  
     if (sessionStorage.getItem("TOKEN") == null) return;  
     var url = this.GLOBAL.BASE\_URL;  
     if (flag === true) {  
       url = url + "long/collection/cancel?id=" + commentary.id;  
    } else {  
       url = url + "long/collection/add?id=" + commentary.id;  
    }  
     this.$options.methods.getRequest.bind(this)(url);  
  },

7、查看收藏，点击表示收藏的心型图标，就将该条评论push到我的收藏影评里

search(page){  
     if(sessionStorage.getItem("TOKEN") == null)  
       return;  
     //设置请求头  
     this.$axios.defaults.headers.common[  
       "Authorization"  
    ] = sessionStorage.getItem("TOKEN");  
     let url = this.GLOBAL.BASE\_URL+"long/get/collection?page="+page  
     this.$axios.get(url)  
    .then(res=>{return Promise.resolve(res.data)})  
    .then(json=>{  
       if(json.code === "ACK"){  
         this.commentaryList = json.data.list;  
         this.pagination.totalPage = Math.ceil(json.data.total/this.pagination.pageRange);  
      }else{  
         console.log(json.message)  
      }  
    })  
    .catch(error=>{console.log(error)})  
  },  
         
     getRequest(url) {  
     //设置请求头  
     this.$axios.defaults.headers.common[  
       "Authorization"  
    ] = sessionStorage.getItem("TOKEN");  
     this.$axios  
      .get(url)  
      .then(res => {  
         return Promise.resolve(res.data);  
      })  
      .then(data => {  
         if (data.code === "ACK") {  
           return data.data;  
        } else {  
           console.log(data);  
        }  
      })  
      .catch(error => {  
         console.log(error);  
      });  
  }

8、个人信息修改

uploadImg() {  
     var form = new FormData();  
     var img = this.cropImg;  
     var imgName = this.cropImgName;  
     var file = this.$options.methods.dataURLtoFile(img, imgName);  
     var fileSize = file.size;  
     //检查头像图片的大小是否小于3MB  
     if (fileSize > 3 \* 1024 \* 1024) {  
       console.log("图像超过3MB，无法上传");  
       return;  
    }  
     // console.log("file size:" + file.size);  
     form.append("file", file);  
     this.$axios  
      .post("/upload/img/movie", form)  
      .then(res => {  
         return Promise.resolve(res);  
      })  
      .then(json => {  
         json = json.data;  
         this.$refs.closeModal.click();  
         this.$parent.user.headerImgUrl = json.data;  
         this.user.headerImgUrl = json.data;  
         this.$parent.$parent.alert("success", json.message);  
      })  
      .catch(error => {  
         console.log(error);  
         this.$parent.$parent.alert("danger", error);  
      });  
  },  
   dataURLtoFile(dataurl, filename) {  
     var arr = dataurl.split(","),  
       mime = arr[0].match(/:(.\*?);/)[1],  
       bstr = atob(arr[1]),  
       n = bstr.length,  
       u8arr = new Uint8Array(n);  
     while (n--) {  
       u8arr[n] = bstr.charCodeAt(n);  
    }  
     return new File([u8arr], filename, { type: mime });  
  },  
   getUserInfo() {  
     var url = "/user/info?username=" + this.$parent.user.username;  
     this.$axios  
      .get(url)  
      .then(res => {  
         return Promise.resolve(res);  
      })  
      .then(json => {  
         json = json.data;  
         this.user = json.data;  
         this.$options.methods.regionExchange.bind(this)(this.user.location);  
      });  
  },  
   addressSelect(object) {  
     this.user.location = object.area.code;  
  },  
   regionExchange(id) {  
     let regionId = id;  
     let splitRegion = /(\d{2})(\d{2})/.exec(regionId);  
     let provinceId = splitRegion[1] \* 10000;  
     let cityId = provinceId + splitRegion[2] \* 100;  
     var location = this.GLOBAL.BASE\_REGIONS;  
     for (let region of location) {  
       if (region.id === provinceId) {  
         location = region;  
         break;  
      }  
    }  
     this.location.province = location.name;  
     if(provinceId === cityId)  
       return  
     for (let region of location.children) {  
       if (region.id === cityId) {  
         location = region;  
         break;  
      }  
    }  
     this.location.city = location.name;  
     if(regionId === cityId)  
       return  
     for (let region of location.children) {  
       if (region.id === regionId) {  
         location = region;  
         break;  
      }  
    }  
     this.location.area = location.name;  
  },  
   changeLocation(e){  
     if(e.code!=null)  
       this.user.location = e.code  
  },  
   modifySubmit() {  
     this.$axios  
      .post("/user/update", this.user)  
      .then(res => {  
         return Promise.resolve(res);  
      })  
      .then(json => {  
         json = json.data;  
         this.$parent.$parent.alert("success", json.message);  
         this.reload()  
      });  
  }

### 5.3.2后台

1、解决登录注册退出登录问题

登录：将用户信息用sessionStorage存储token，登录输入的信息不在token里则报错

登出：删除所有数据：sessionStorage.clear();

注册：判断两次输入的密码是否相同，不同则会提示“请确认密码”，如果没有问题则执行注册操作。

login() {

this.loading = true;

var data = qs.stringify({

username: this.form.name,

password: this.form.password

});

this.$axios

.post("/login", data)

.then(res => {

this.loading = false;

return Promise.resolve(res.data);

})

.then(json => {

if (json.code == "ACK") {

if (json.data.role === "admin") {

sessionStorage.setItem("JWT", json.data.token);

sessionStorage.setItem("username", this.form.name);

this.$router.push("/admin/page/manage/user");

} else this.$message("你不是管理员，登录失败");

} else {

this.$message.error(json.message);

}

})

.catch(function(error) {

console.log(error);

});

}

},

components: {

Icon

}

};

2、解决用户管理问题

编辑、查询、封号、解封用户

onSubmit() {

var data = {

username: this.formInline.user,

pageSize: 10,

pageNum: 1

};

this.$axios

.post("/user/search", data)

.then(function(res) {

return Promise.resolve(res.data);

})

.then(json => {

this.tableData = json.data.list;

this.pageNumber = json.data.pageNumber;

})

.catch(function(error) {

console.log(error);

});

},

toggleSelection(rows) {

if (rows) {

rows.forEach(row => {

this.$refs.multipleTable.toggleRowSelection(row);

});

} else {

this.$refs.multipleTable.clearSelection();

}

},

handleSelectionChange(val) {

this.multipleSelection = val;

},

handleEdit(index, row) {

this.dialogFormVisible = true;

this.form = row;

this.selectedOptions = [];

if (this.form.location != null)

this.selectedOptions = linkFatherRegions(this.form.location);

},

handleBanAccount(index, row) {

this.$axios

.get("/user/ban", { params: { username: row.username } })

.then(res => {

let data = Promise.resolve(res.data);

return data;

})

.then(data => {

if (data.code == "ACK") {

this.$message({

message: data.message,

type: "success"

});

this.reload();

} else {

this.$message.error(data.message);

}

})

.catch(function(error) {

console.log(error);

});

},

handleUnBanAccount(index, row) {

this.$axios

.get("/user/unban", { params: { username: row.username } })

.then(res => {

let data = Promise.resolve(res.data);

return data;

})

.then(data => {

if (data.code == "ACK") {

this.$message({

message: data.message,

type: "success"

});

this.reload();

} else {

this.$message.error(data.message);

}

})

.catch(function(error) {

console.log(error);

});

},

submitUserForm() {

this.$axios

.post("/user/update", this.form)

.then(function(res) {

let data = Promise.resolve(res.data);

return data;

})

.then(data => {

if (data.code == "ACK") {

this.$message({

message: data.message,

type: "success"

});

this.reload();

} else {

this.$message.error(data.message);

}

this.dialogFormVisible = false;

this.reload();

})

.catch(function(error) {

console.log(error);

});

},

getSelectPage(page) {

this.$options.methods.getUserList.bind(this)(page);

},

getUserList(page) {

var data = {

username: this.formInline.user,

pageNum: page,

pageSize: 10

};

this.$axios

.post("/user/search", data)

.then(function(res) {

return Promise.resolve(res.data);

})

.then(json => {

this.pageNumber = json.data.pageNumber;

this.tableData = json.data.list;

for (var data of json.data.list) {

convertToString(data);

}

})

.catch(function(error) {

console.log(error);

});

},

regionChange(value) {

this.form.location = value[value.length - 1];

}

}

};

function convertToString(form) {

if (form.sex === 0) form.sex = "0";

else if (form.sex === 1) form.sex = "1";

}

function linkFatherRegions(region) {

var locations = [];

locations[0] = region - (region % 1000);

if (region != locations[0]) {

locations[1] = region - (region % 100);

if (region != locations[1]) locations[2] = region;

}

return locations;

}

3、解决电影管理：电影上传

onSubmit() {

this.$refs["form"].validate(valid => {

if (valid) {

if (this.imgFileSize > 0) {

this.$refs.upload\_img.submit();

} else {

this.$options.methods.submitFilmFormDate.bind(this)();

}

} else {

this.$message({

message: "请填写必填项",

type: "warning"

});

}

});

},

overLimit(files, fileList) {

this.$message("只能上传一张图片");

},

imgUpload(data) {

let config = {

headers: { "Content-Type": "multipart/form-data" }

};

let formDate = new FormData();

formDate.append("file", data.file);

this.$axios

.post("/movie/img/upload", formDate, config)

.then(function(res) {

return Promise.resolve(res.data);

})

.then(json => {

this.form.imgUrl = json.data;

this.$options.methods.submitFilmFormDate.bind(this)();

})

.catch(function(error) {

console.log(error);

});

},

submitUpload() {

this.$refs.upload\_img.submit();

},

handleRemove(file, fileList) {

this.imgFileSize = fileList.length;

},

changeImgFileSize(file, fileList) {

this.imgFileSize = fileList.length;

},

submitFilmFormDate() {

this.$axios

.post("/movie/add", this.form)

.then(res => {

let data = Promise.resolve(res.data);

return data;

})

.then(data => {

if (data.code == "ACK") {

this.$message({

message: data.message,

type: "success"

});

this.reload()

} else {

this.$message.error(data.message);

}

})

.catch(function(error) {

console.log(error);

});

},

resetAddForm(){

for(let index in this.form){

this.form[index] = "";

}

this.filmCountry = [];

this.filmLanguage = [];

this.filmType = [];

this.fileList = [];

}

}

};

1. 解决幻灯片上映问题

tableRowClassName({ row, rowIndex }) {

if (rowIndex < 3) {

return "warning-row";

} else {

return "success-row";

}

},

handleEdit(index, row) {

this.EditFormVisible = true;

this.slide\_item = row;

},

handleDelete(index, row) {

this.deleteSlideId = row.id;

this.warningVisible = true;

},

exchangeSlideSelect(val) {

this.slideExchange.oldSlideId = val.id;

},

handleExchange(index, row) {

this.slideExchangeVisible = true;

this.slideExchange.newSlideId = row.id;

},

//上传图片之前先检查图片信息

beforeAvatarUpload(file) {

const isJPG = file.type === "image/jpeg";

const isLt2M = file.size / 1024 / 1024 < 10;

if (!isJPG) {

this.$message.error("上传头像图片只能是 JPG 格式!");

}

if (!isLt2M) {

this.$message.error("上传头像图片大小不能超过 10MB!");

}

return isJPG && isLt2M;

},

changeUploadImgURL(file) {

this.upload\_img = URL.createObjectURL(file.raw);

},

submitUpload() {

this.$refs.upload\_slide.submit();

},

//幻灯片图片更换成功调用的函数

handleAvatarSuccess(res, file) {

},

//幻灯片添加成功调用的函数

handleAddSlideSuccess(res, file) {

if (res.code === "ACK") {

this.slide\_add\_item.img = res.data;

this.$options.methods.submitSlideAddForm.bind(this)();

} else {

this.$message({

message: res.message,

type: "warning"

});

}

this.reload();

},

1. 解决影评管理信息：查询、查看、删除影评

onSubmitSearch(page) {

this.formInline.pageNum = page;

this.$axios.post("long/search",this.formInline)

.then(res=>{return Promise.resolve(res.data)})

.then(json=>{

if(json.code==="ACK"){

this.tableData = json.data.list;

this.total = json.data.total;

}else{

this.$message.error(json.message);

}

})

.catch(error=>{this.$message.error(error);})

},

getSelectPage(page){

this.$options.methods.onSubmitSearch.bind(this)(page);

},

handleWatch(index,row){

this.commentary = row;

this.dialogFormVisible = true;

},

handleDelete(index,row){

this.$options.methods.openDelete.bind(this)(row.id)

},

delete(id){

let url = this.GLOBAL.BASE\_URL+"long/delete?id="+id;

this.$axios.get(url)

.then(res=>{return Promise.resolve(res.data)})

.then(json=>{

if(json.code==="ACK"){

this.$message({

type: "success",

message: json.message

});

}else{

this.$message.error(json.message)

}

})

.catch(error=>{this.$message.error(error)})

},

openDelete(id) {

this.$confirm("此操作将删除该影评, 是否继续?", "提示", {

confirmButtonText: "确定",

cancelButtonText: "取消",

type: "warning"

})

.then(() => {

this.$options.methods.delete.bind(this)(id);

})

.catch(() => {

this.$message({

type: "info",

message: "已取消删除"

});

});

},

handleBanAccount(username) {

this.$confirm("确认对该用户给予封号处理, 是否继续?", "提示", {

confirmButtonText: "确定",

cancelButtonText: "取消",

type: "warning"

})

.then(() => {

this.$axios

.get("/user/ban", { params: { username: username } })

.then(res => {

let data = Promise.resolve(res.data);

return data;

})

.then(data => {

if (data.code == "ACK") {

this.$message({

message: data.message,

type: "success"

});

dialogFormVisible = false;

} else {

this.$message.error(data.message);

}

})

.catch(function(error) {

console.log(error);

});

})

.catch(() => {

this.$message({

type: "info",

message: "已取消删除"

});

});

}

}

};

# 项目评估

## 6.1项目分工

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 姓名 | 学号 | 班级 | 任务 | 权重 |
| 黄依伦 | 31601342 | 软工1601 | 前端实现 | 1.0 |
| 刘晓倩 | 31601381 | 软工1602 | 后台实现 | 0.9 |

## 6.2自我评估表

黄依伦

|  |  |
| --- | --- |
| 函数式编程 | 5 |
| fetch api | 3 |
| FRP | 3 |
| session Storage | 5 |
| 响应式设计(Desktop/Tablet/Mobile) | 3 |
| REST API | 4 |
| 其他 。。。。 | 5 |

刘晓倩

|  |  |
| --- | --- |
| 函数式编程 | 4 |
| fetch api | 3 |
| FRP | 2 |
| session Storage | 5 |
| 响应式设计(Desktop/Tablet/Mobile) | 4 |
| REST API | 3 |
| 其他 。。。。 | 4 |

## 6.3项目自评等级（1-5）

完成自定义项目，前端使用Bootstrap+Vue实现，后台使用element+Vue实现，采用跨平台技术，实现服务端与浏览器端交互。

黄依伦：4

刘晓倩：4

# 心得体会

## 7.1 课程感言

通过这一个学期的跨平台脚本开发课程的学习，我认为收获还是颇多，郭老师上课很耐心的带着我们学习，在老师的引导下，我学会了很多有关于跨平台脚本开发技术的内容，也进行了很多项目实战，希望在后面的学习中能更好的掌握。

课程难度：感觉偏难，初期还可以接受，中后期作业就有点吃力了

进度：适中

课程内容：前期学习基本语法操作，中后期的项目实战内容比较丰富，有专门的网站进行学习（10.66.27.234:3001）

授课方式：感觉还是一边讲一边做比较好，就是老师在上边演示的同时学生能跟着操作，不然有些过程很容易忘记，或者下面听的学生比较困倦，就无法理解对应的过程；我希望老师以后就这门课上课时可以在Doctorz让我们进行课堂练习，做一些习题，我觉得这样更容易掌握上节课的知识点，能够帮助更好地理解和学习。

## 7.3个人小结

我觉得自己在这次的大作业中页面实现的效果还不错，功能也挺多，我也收获了很多，刚开始只刚入门JavaScript的基本知识，从webpack配置开始才慢慢觉得在做一个项目的开始，整体实现工程化，通过各种网络途径学习了相关知识，还跟着视频学习了喵喵电影的制作，从一门课引申学了很多知识，这次课程还是学到了很多，像git、mysql都是很好的巩固，学会了很多前端框架，和后台存储管理的交互知识，很好的了解了函数式编程，体验了JavaScript的魅力和便捷性，希望在以后能更深入地学习js，我感觉收获颇丰，整个课程的体验感也很好。

# 参考资料

<https://element.eleme.cn/#/zh-CN>

<https://cn.vuejs.org/>

<https://www.cnblogs.com/lhp2012/p/6369489.html>

<http://diannaobos.com/frp/>

<https://www.leancloud.cn/docs/rest_api.html>