# 1. 需求分析

### 1.1 愿景陈述

电影作为极具艺术审美性的服务型产品，生命周期短暂，受众偏好较难把握，属于典型的体验型商品，这些特性决定了电影票房的影响因素与其他商品有所不同，对电影票房预测的不确定性程度更高。对电影票房的分析，对于提升国产电影软实力，推动中国文化走出去具有重要意义。

如果我们有一个网站，能够实时爬取电影票房数据、将数据存储在数据库中，并且在网站页面对票房相关的数据进行可视化，显示出电影的排片、票房、购票指数、票房大盘等，这对于电影行业相关人员有着重要的参考作用。

### 1.2 用户与业务范围

1. 用户类型范围

本网站主要面向于电影行业相关人员和数据分析师，和一些电影爱好者。

1. 业务目标

BO-1：能够实现用户的登陆、注册以及个人信息的修改。

BO-2：实现不同等级的用户拥有不同的访问权限。

BO-3：正确显示电影的相关数据，如票房、购票指数、排片等。

1. 业务风险

RI-1：电影相关数据不完整、不正确。

RI-2：数据库遭到攻击，用户访问权限被修改。

RI-3：服务器负载不足。

### 1.3 需求分析结果

##### 1.3.1 页面设计

1. 登录

实现用户的登陆页面，如果没有注册，则跳转至注册页面；与数据库进行交互后，返回校验码确定用户是否能够登陆。

1. 注册

实现用户的注册页面，注册成功后即可登陆；注册后将响应数据存入数据库中。

1. 个人中心

实现用户的个人中心页面，用户可以修改自己的个人信息，以及修改密码；修改完成后对数据库相关的数据进行修改。

1. 反馈

实现反馈页面，用户可以通过该页面提交对网站的建议，或者提出相关的咨询，如订单问题等。对于与用户个人信息相关的反馈，将存入数据库中；对于网站方面的建议等，存入数据库后也会显示在反馈页面供其他用户浏览。

1. 票房榜

从数据库中获取电影票房的数据，并以金额大小从高到低排序进行显示，显示电影名称、上映日期、票房总额等。

1. 购票指数

从数据库中获取电影票房的购票指数、影片名称、上映日期、排名变化等信息。

1. 票房大盘

对于票房，以日期、国别、影片制式等为分界，对这些数据进行可视化供用户浏览。

1. 影片排片

从数据库中获得电影的排片数据，并以场次、座位数等指标进行可视化。

1. 充值

实现充值页面，用户选择对应的级别进行充值，每个级别对应不同的访问权限，充值成功后更新数据库中的用户访问权限。

##### 1.3.2 业务规则、主要操作者与用例

###### 1.3.2.1 业务规则

BR-1：网站的一切数据由数据库提供。

BR-2：用户登陆时，如果连续输入密码错误三次，则需要输入验证码；若错误五次，则锁定账户一天。

BR-3：对于用户充值，充值金额到账后方可授予用户访问权限。

BR-4：一个用户一天内最多提交三次反馈。

###### 1.3.2.2 主要操作者和用例

|  |  |
| --- | --- |
| 主要操作者 | 用例 |
| 用户 | 1. 注册 2. 登陆 3. 修改个人信息 4. 充值 5. 反馈 6. 查看票房指数 7. 查看购票指数 8. 查看票房大盘 9. 查看影片排片 |

###### 1.3.2.3 用例图



图1

# 2. 可行性分析：

### 2.1 技术可行性

##### 2.1.1风险分析

本实作有固定人员5人，不会造成人员流失。本实作成品采用node.js技术开发，此为成熟的技术所以并不会有技术性风险。

##### 2.1.2 资源分析

本实作的开发环境使用vscode进行开发，搭建一个服务性网站，现有的软件和硬件都能支撑项目开发。开发人员已经接受了前后端开发的培训，比较熟悉开发的技术。

##### 2.1.3 技术分析

（1）易用性：基于express框架开发的前后端服务性网站，能保证服务器正常运作下99%能成功进入系统。

（2）性能：基于express框架开发的网站能承担大约30条并发访问。系统能够存储大约1000条用户信息。

（3）防护性：采用post请求模式保证用户的数据在网站不被公开。用户注册登录信息、反馈信息、浏览信息存入本地数据库后保证不被泄露。

（4）健壮性：网站设计时，开发人员会考虑各种报错情况并及时给予错误反馈，所以网站遇到非法输入时，能及时做出正确反映，保证系统正常运行而不是崩溃。

### 2.2 经济可行性

##### 2.2.1 成本

项目开发中所使用的软件、硬件、设备无成本。开发人员为学生故无开发人力成本，也无维护成本。开发人员已接受培训，无成本。

##### 2.2.2 效益

此项目为学院实作项目作业，不会产生经济效益。

##### 2.2.3 投资回收率

因为没有经济效益，固然没有投资回收率。

##### 2.2.4 纯收入

不会产生收入。

### 2.3 法律可行性：

该实作所使用的数据为网络公开数据，故不会有侵权行为。且内容积极、向上、健康、非盈利，在法律的运行范围中。

# 3. 项目设计

### 3.1 概要设计：

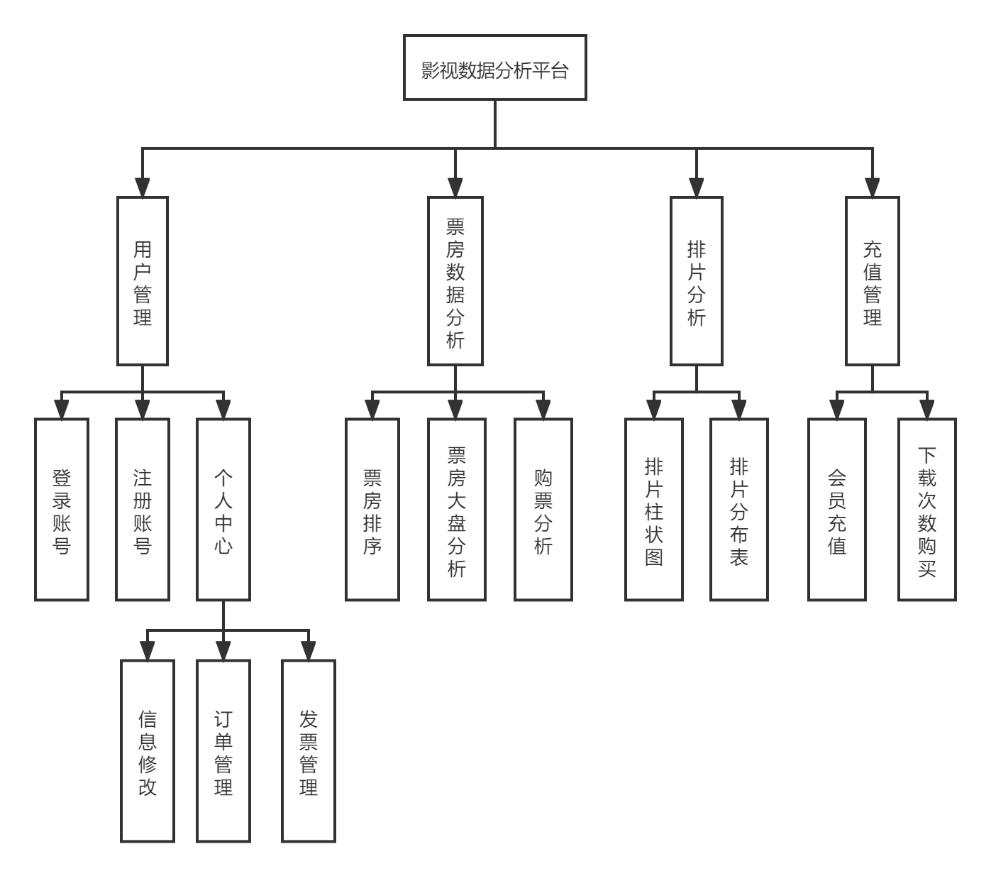
##### 3.1.1 总体设计原则

平台要有稳定可靠的性能，数据存储，查询，分析应该准确无误，在数据导入导出时不能造成数据的错误或丢失。总体设计要采用模块化、抽象、逐步求精、信息隐藏、模块独立的设计原则。界面设计时要遵守以用户为中心、一致性、简洁与明确、体现特色、兼顾不同浏览器、明确的导航设计等基本web界面设计原则。界面与数据处理分离，从而能够更灵活地修改系统。

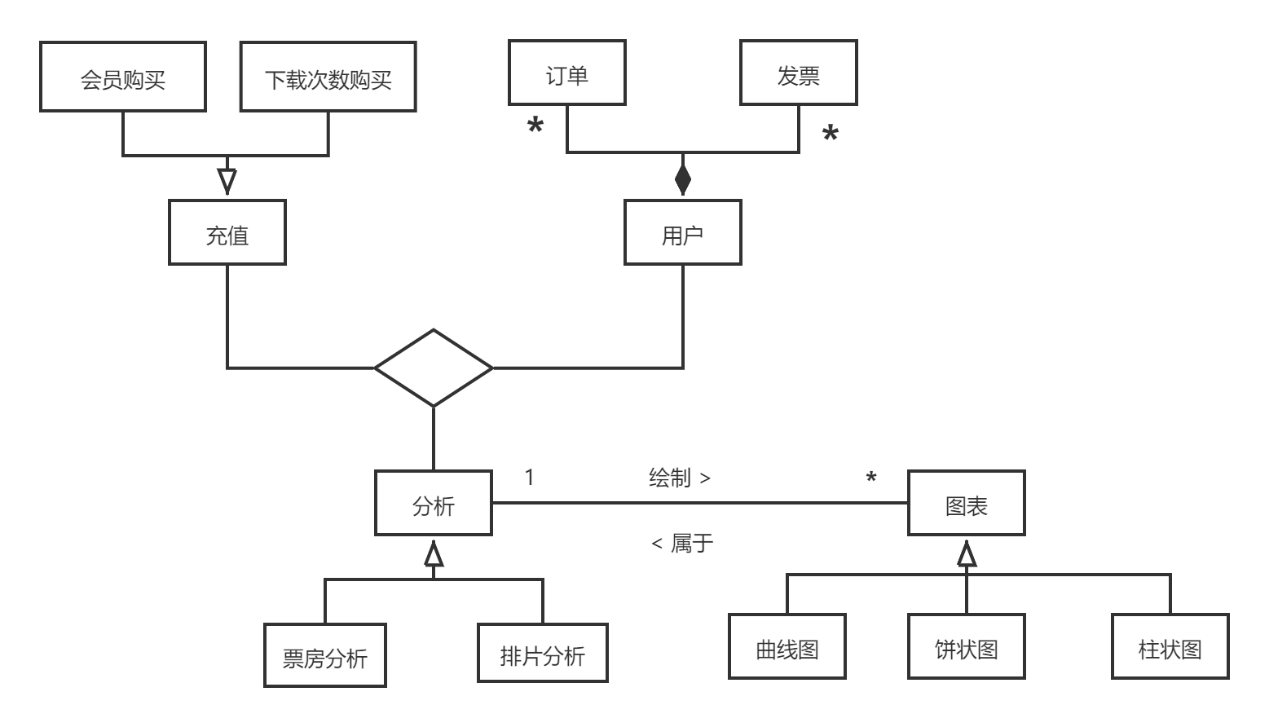
##### 3.1.2 运行环境

1. 允许在Chrome、Edge、Firefox、IE、Safari等浏览器运行。
2. Mysql 5.6 数据库

##### 3.1.3 功能结构图



##### ****3.1.4**** 类之间的关系



### 3.2 详细设计

##### 3.2.1 活动图

详细设计活动图：



登录活动图



反馈活动图



个人中心修改密码活动图



购买续费活动图



注册活动图

##### 3.2.2 类图



# 4. 项目实现

### 4.1 页面基本框架

主页面基本框架由顶栏（红）、侧栏（黄）、内容栏（蓝）三部分构成，采用响应式布局。顶栏设为一行，侧栏和内容栏设为一行，侧栏和内容栏为该行的两个不同列，如图2、图3。顶栏实现：采用一行三列的响应式布局，三列存放logo、搜索、登录按钮组件，如图4。

侧栏标签的伸展效果CSS实现，效果如图5、图6、图7。内容栏用来展示每个页面的数据内容，上下滑动滚动轴只在内容栏实现滑动，而侧栏顶栏不会受滚动影响，如图8。内容栏内容是响应式布局效果，如图9。内容栏、顶栏、侧栏采用分离式结构，即在不同HTML文件上实现，在多变的内容栏html上导入顶栏侧栏的html代码，减少代码冗余，如图10、11。

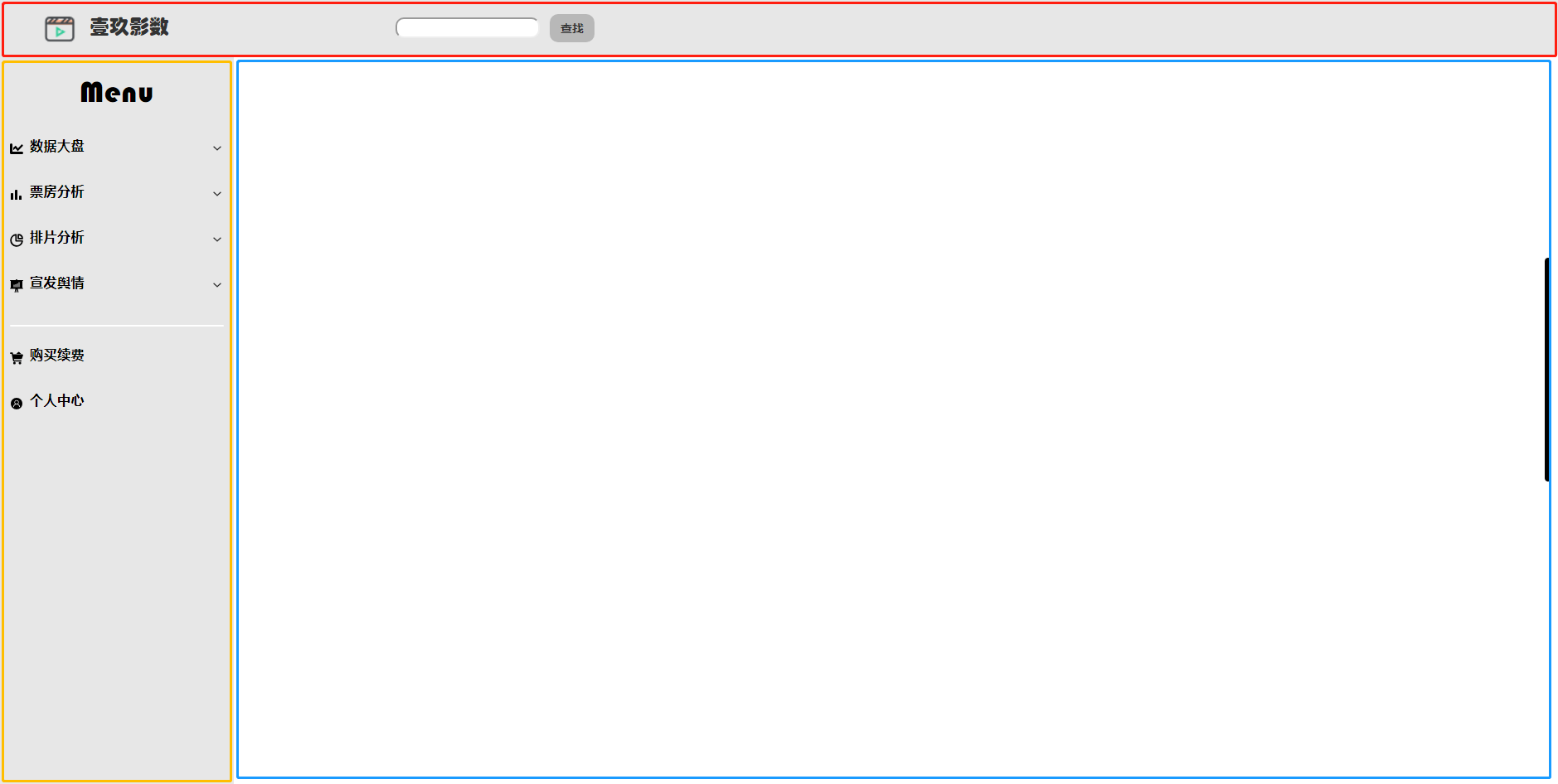


图2

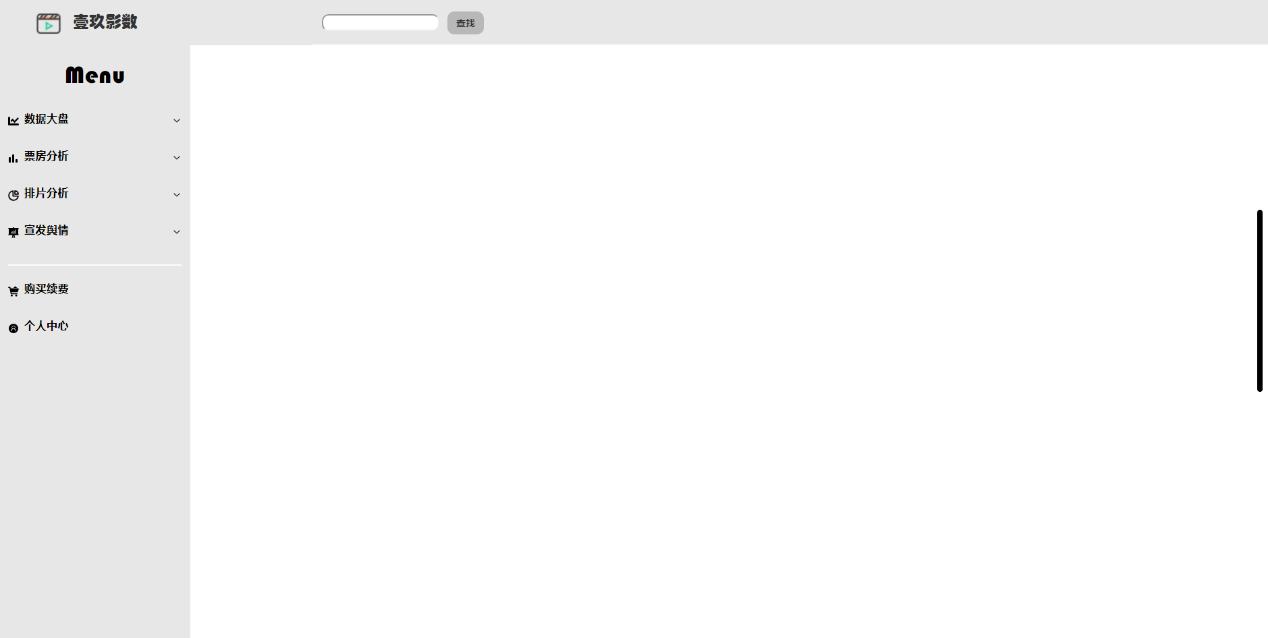


图3



图4

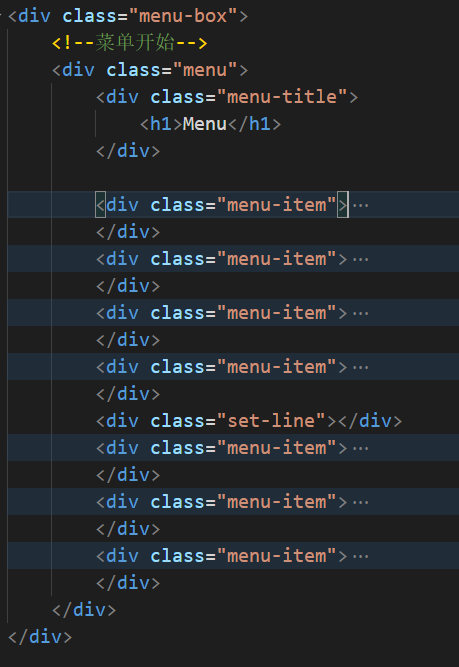


图5

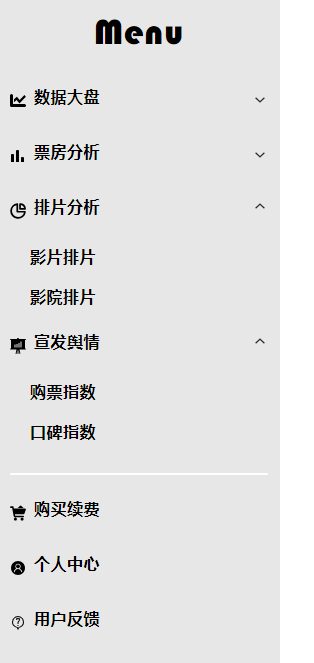


图6

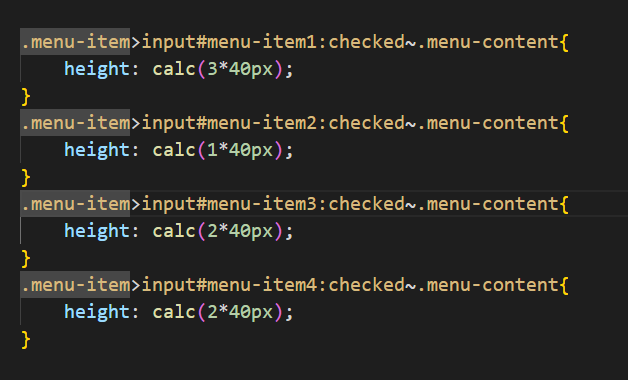


图7

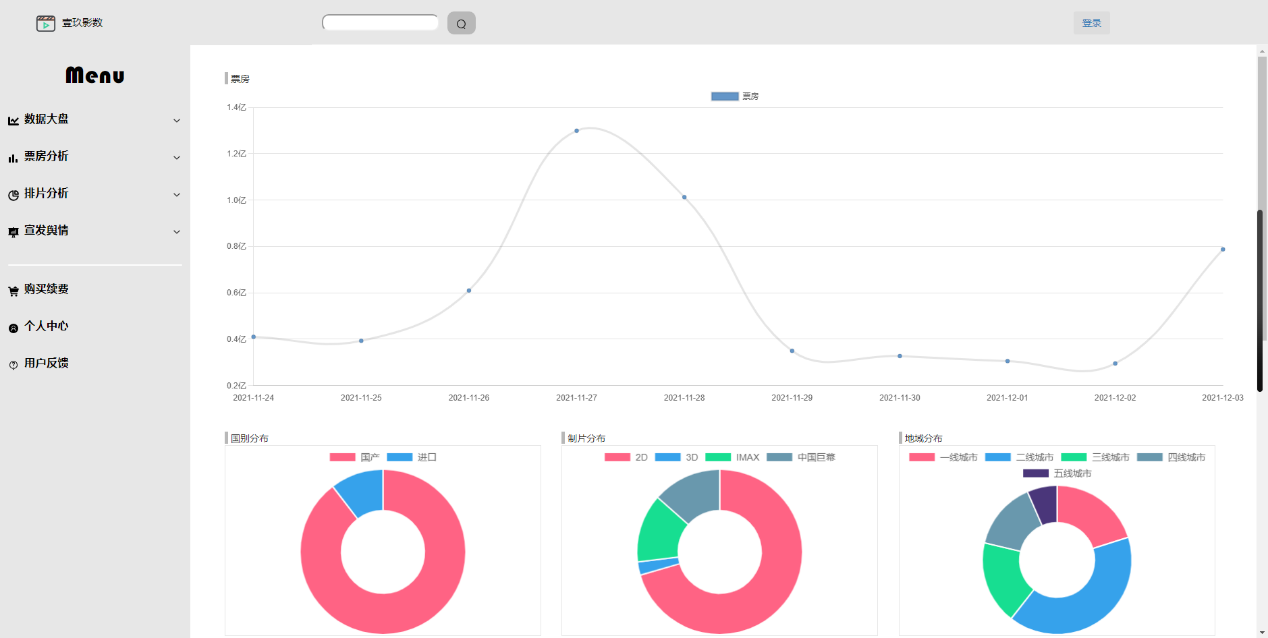


图8

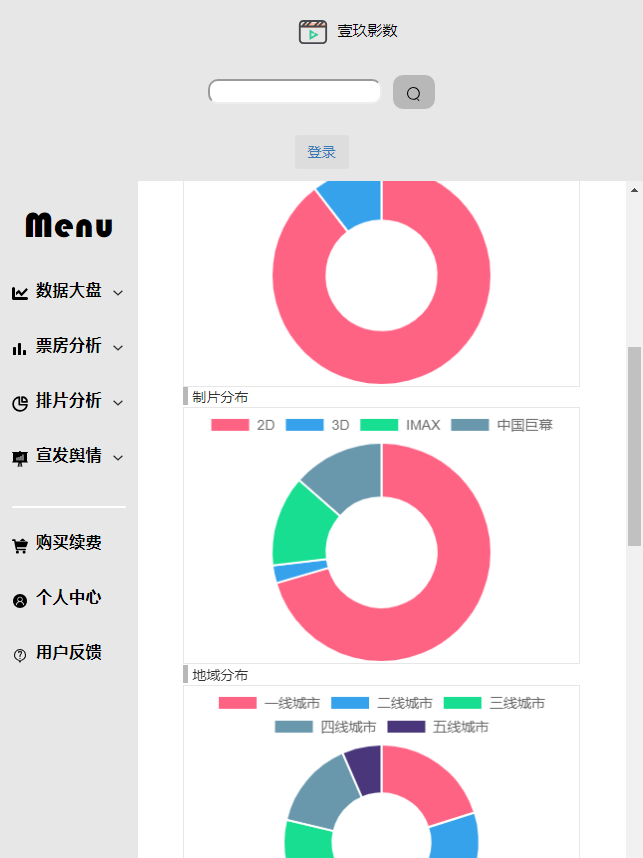


图9

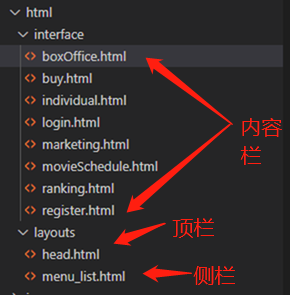


图10



图11

### 4.2 图表的实现



图12 折线图的实现

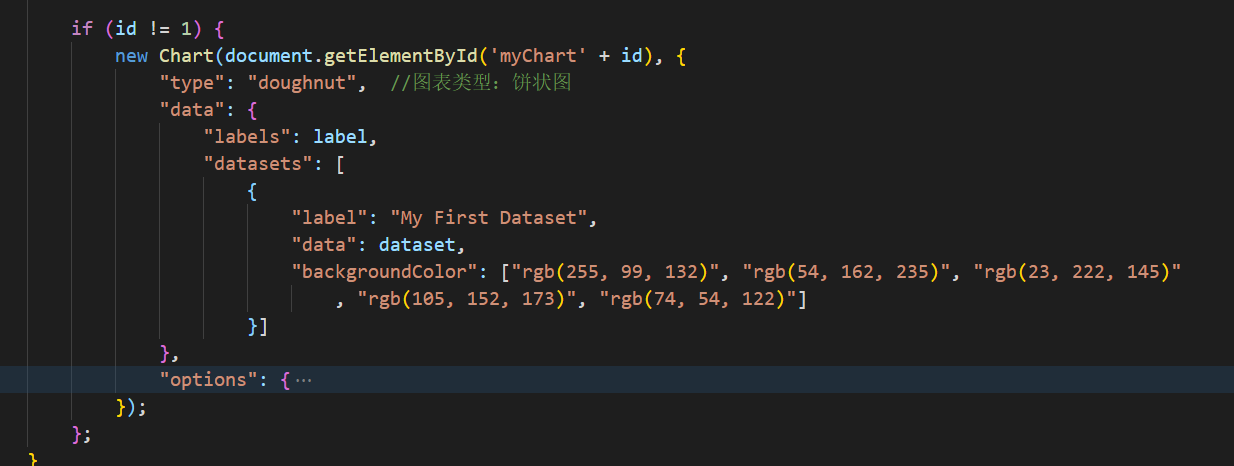


图13 饼状图的实现

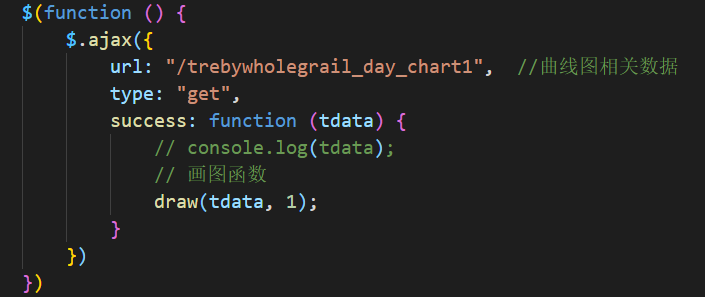
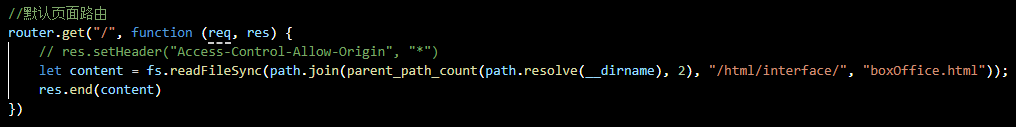
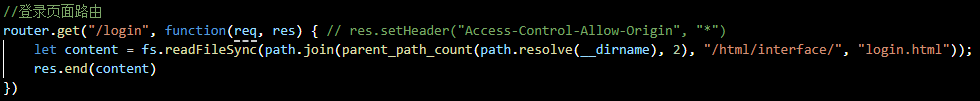


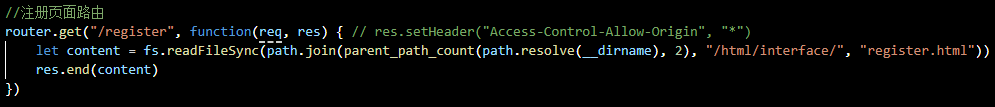
图14 Ajax从后端获取数据传参给画图函数进行构图

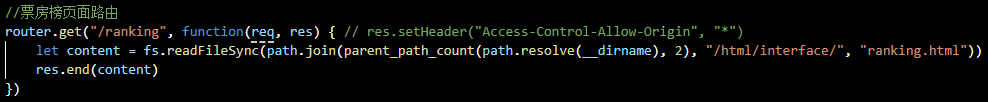
### 4.3 页面跳转路由

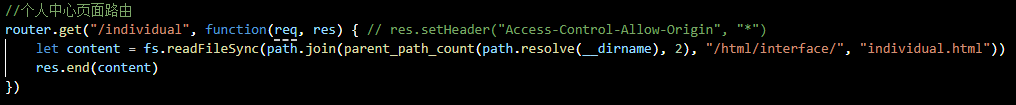
所有页面跳转路由，包括默认页面路由，登录页面路由，注册页面路由，票房榜页面路由，个人中心页面路由，购票指数页面路由，票房大盘页面路由，影片拍片页面路由，购买续费页面路由，反馈页面路由共10个页面路由，如图15：

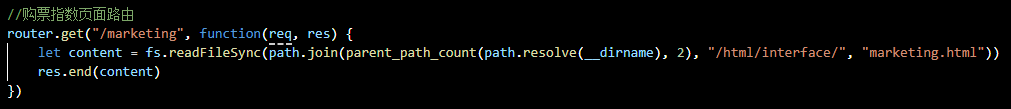


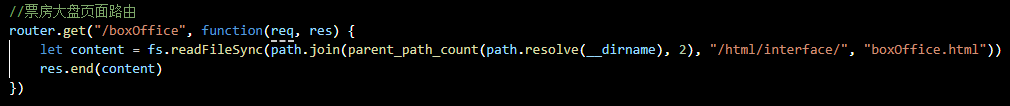


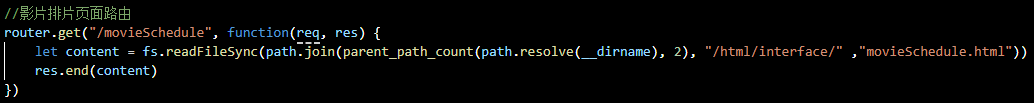


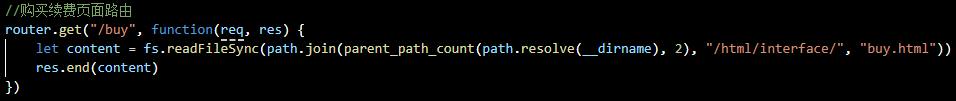












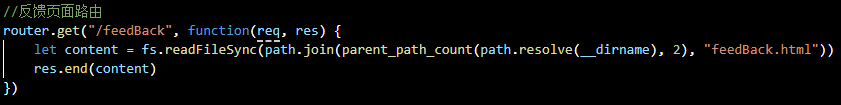


图15 所有页面路由

### 4.4 注册功能

设置输入框与注册按钮表单，如图16。使用ajax设置按钮点击事件：包含正则表达式判断以及将数据传送到数据库。数据传送完成后，弹框提示注册成功，并跳转到登陆页面，如图17、图18、图19。路由获取前端发送的data并解析，生成当前时间并进行格式变换，设置sql语句将数据插入数据库，如图20；



图16



图17

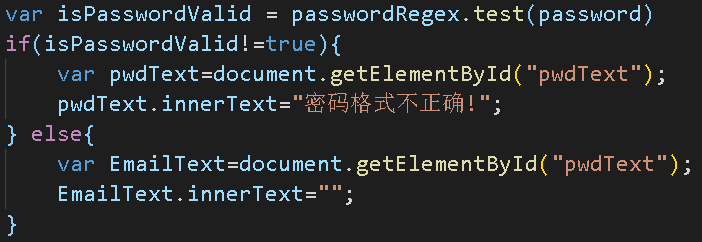


图18



图19

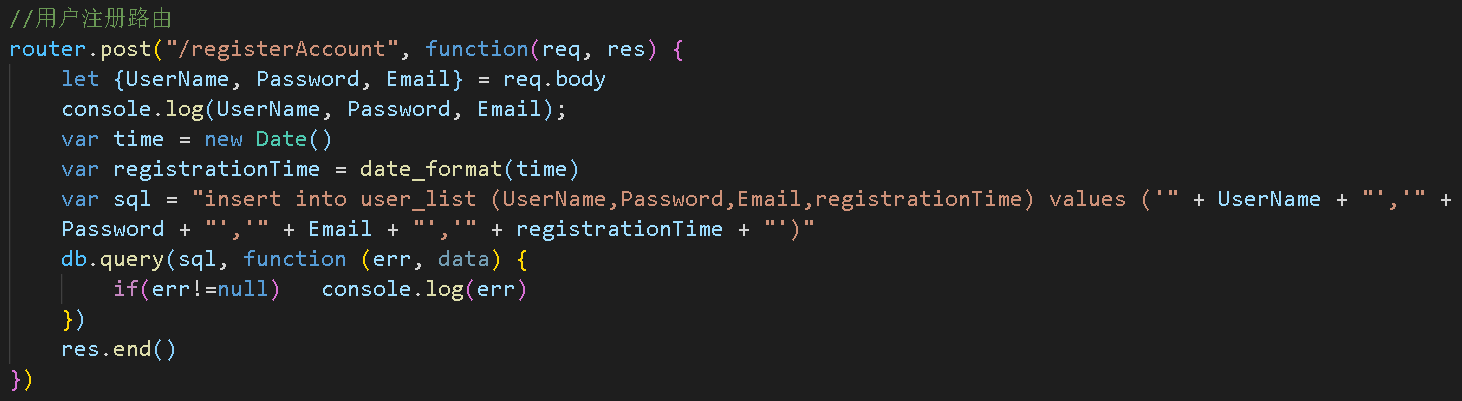


图20

### 4.5 登录功能

设置输入框与注册按钮表单，如图21。使用ajax设置按钮点击事件：将用户名传送到数据库，并获取该用户对应的正确密码与用户的输入密码进行判断。如果密码正确，自动跳转到票房榜页面，如图22。获取前端传送的用户名，查询该用户对应的正确密码并返回前端，如图23；



图21

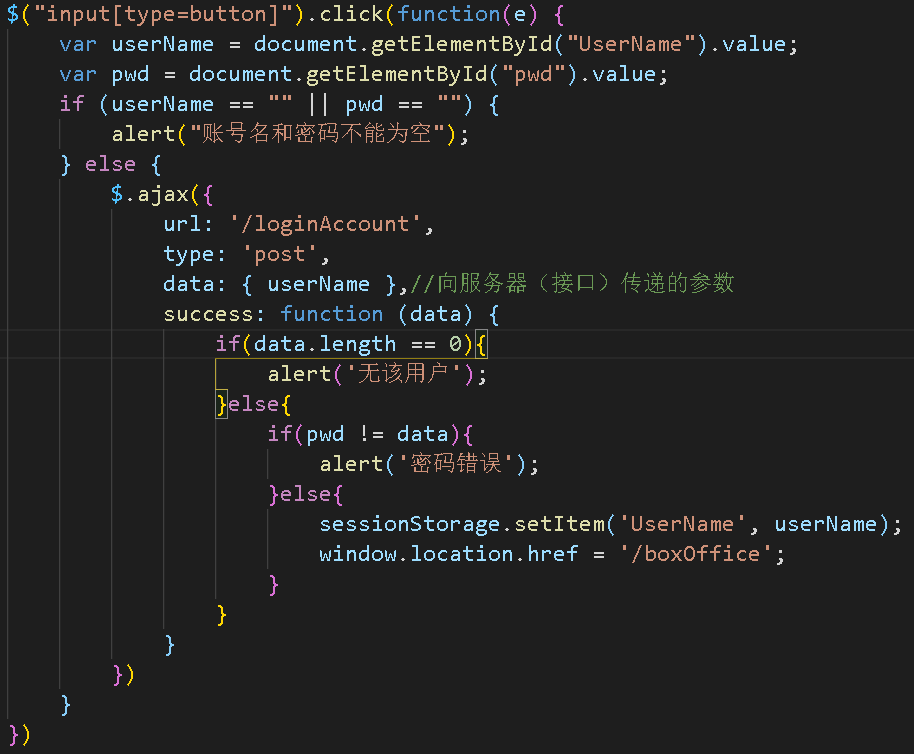


图22

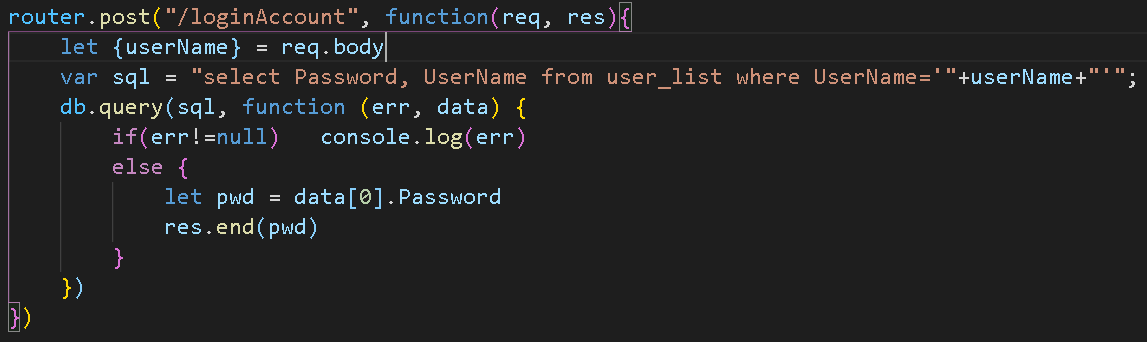


图23

### 4.6 判断是否登录功能

设置显示用户名的标签<span>，如图24。登录成功后，利用sessionStorage. setItem(‘UserName’, userName)在本地设置一个变量UserName，如图25。在head.html判断sessionStorage的长度，如果为0则没有用户登陆过。如果为1，则用户登录并注销过，要判断UserName的属性名。如果为字符串’null’，则说明用户注销，不需要显示用户名；若字符串不为’null’，说明用户已登录，修改<span>标签内容，显示用户名，如图26。

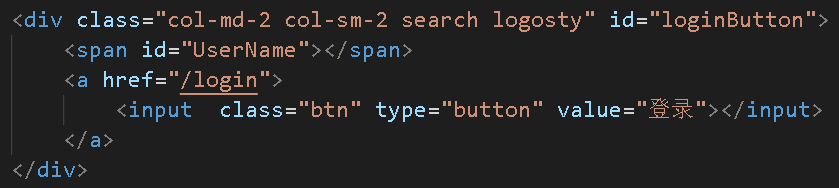


图24

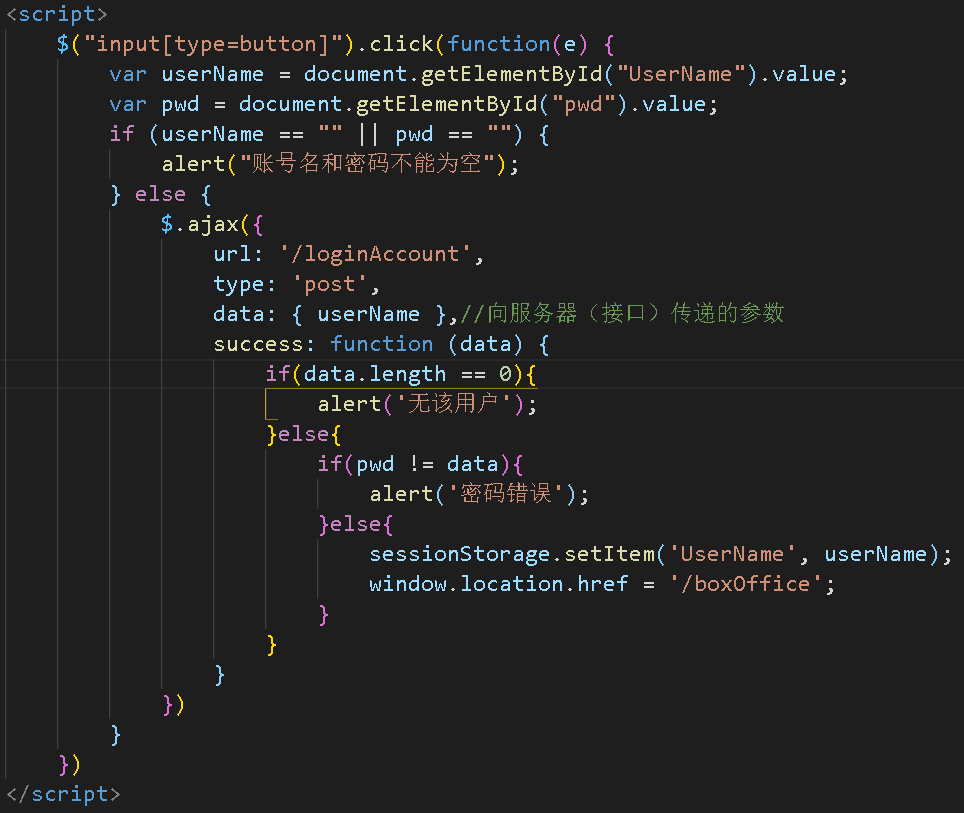


图25

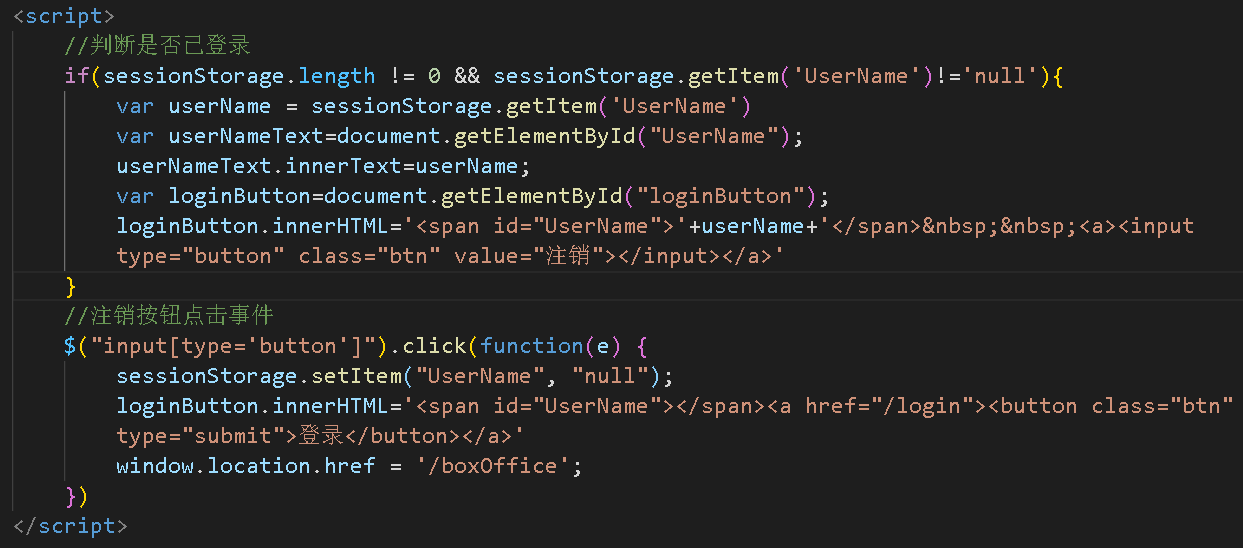


图26

### 4.7 购买续费

在右上角设置一个触发遮罩窗体，中间有关会员定价的3幅图片也一并设置成触发遮罩窗体，如图27。右上角的“购买下载次数”的内容如图28。“购买下载次数”实现方法如图29。中间“会员定价”中“VIP会员的内容”如图30。“VIP会员”的实现方法如图31和图32。“SVIP会员”同理。



图27



图28

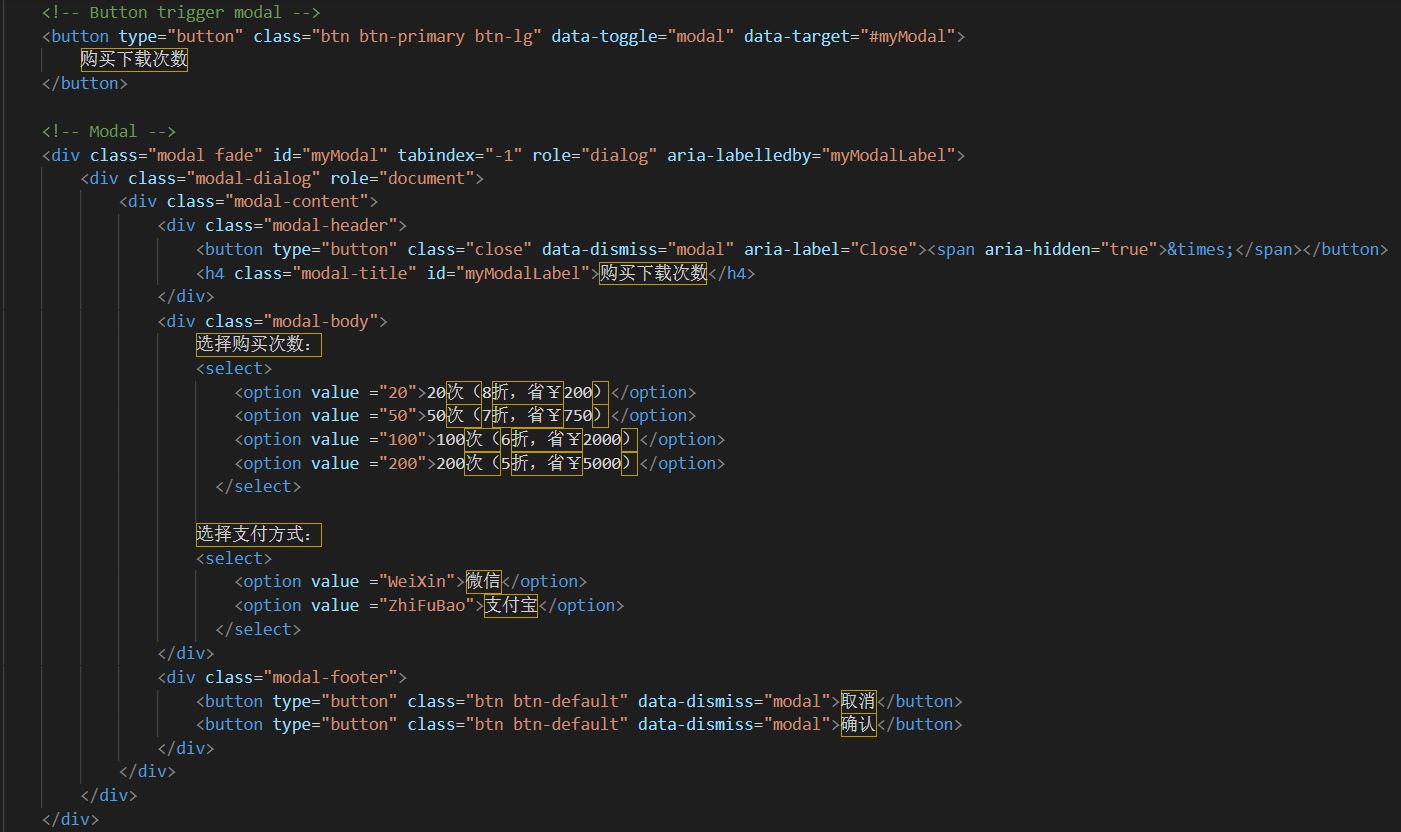


图29



图30

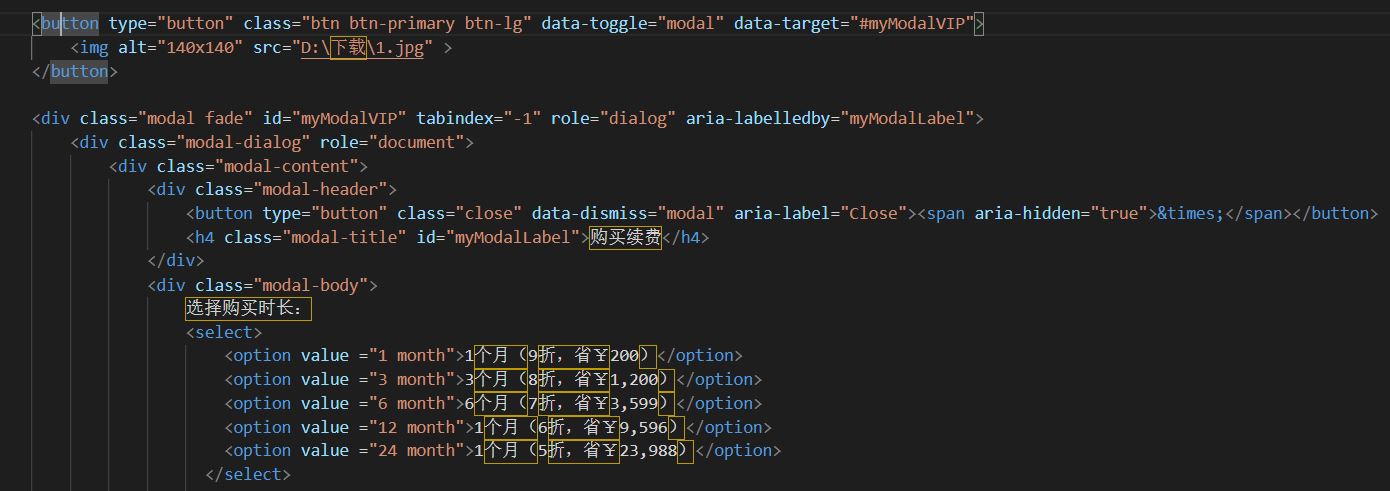


图31



图32

### 4.8 个人中心与用户反馈

通过切换卡实现页面的切换，如图33。“我的资料”内容如图34，实现方法如图35、36、37。“子账号管理”内容如图38，实现方法如图39。“我的订单”内容如图40，实现方法如图41、42。最后将“用户反馈”按钮设置成超链接，地址为问卷地址，问卷采用问卷星编辑。



图33

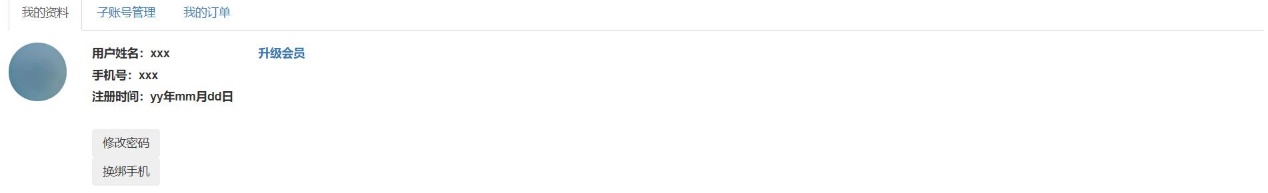


图34

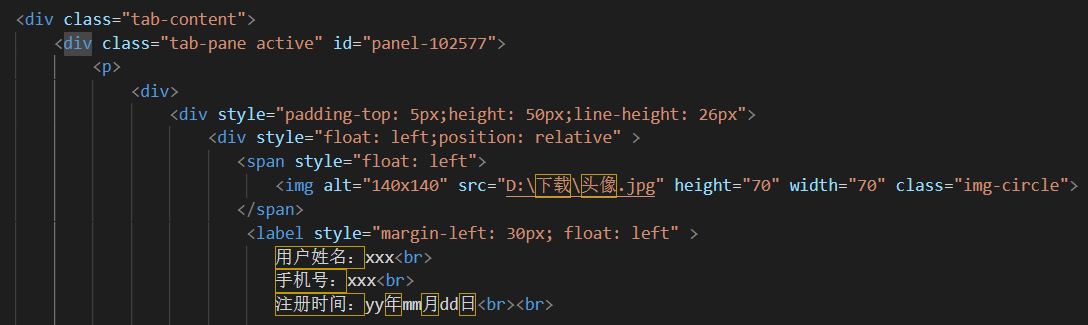


图35

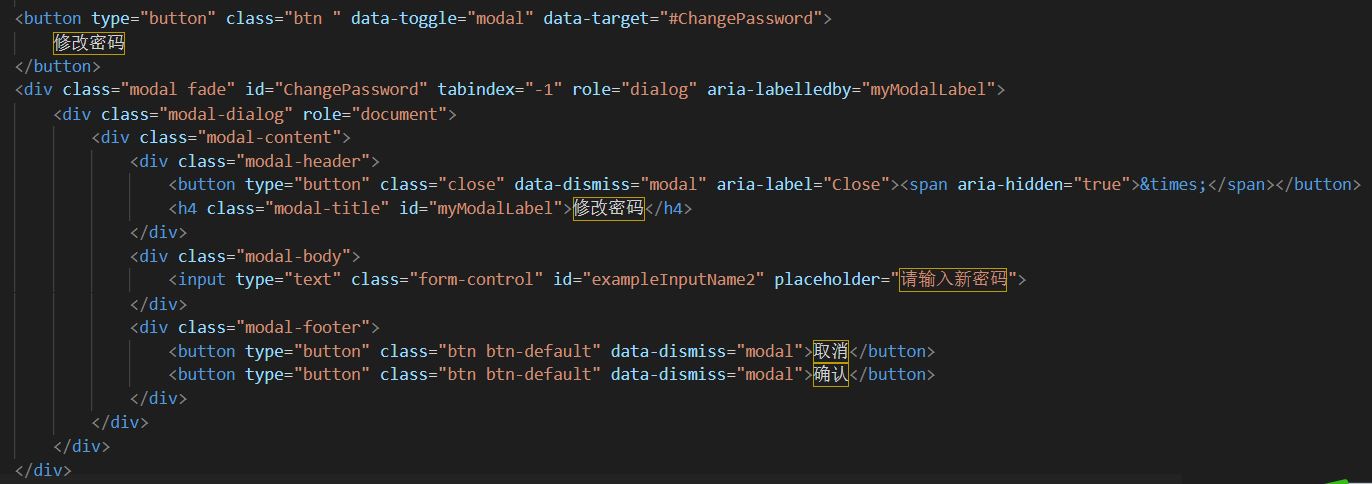


图36

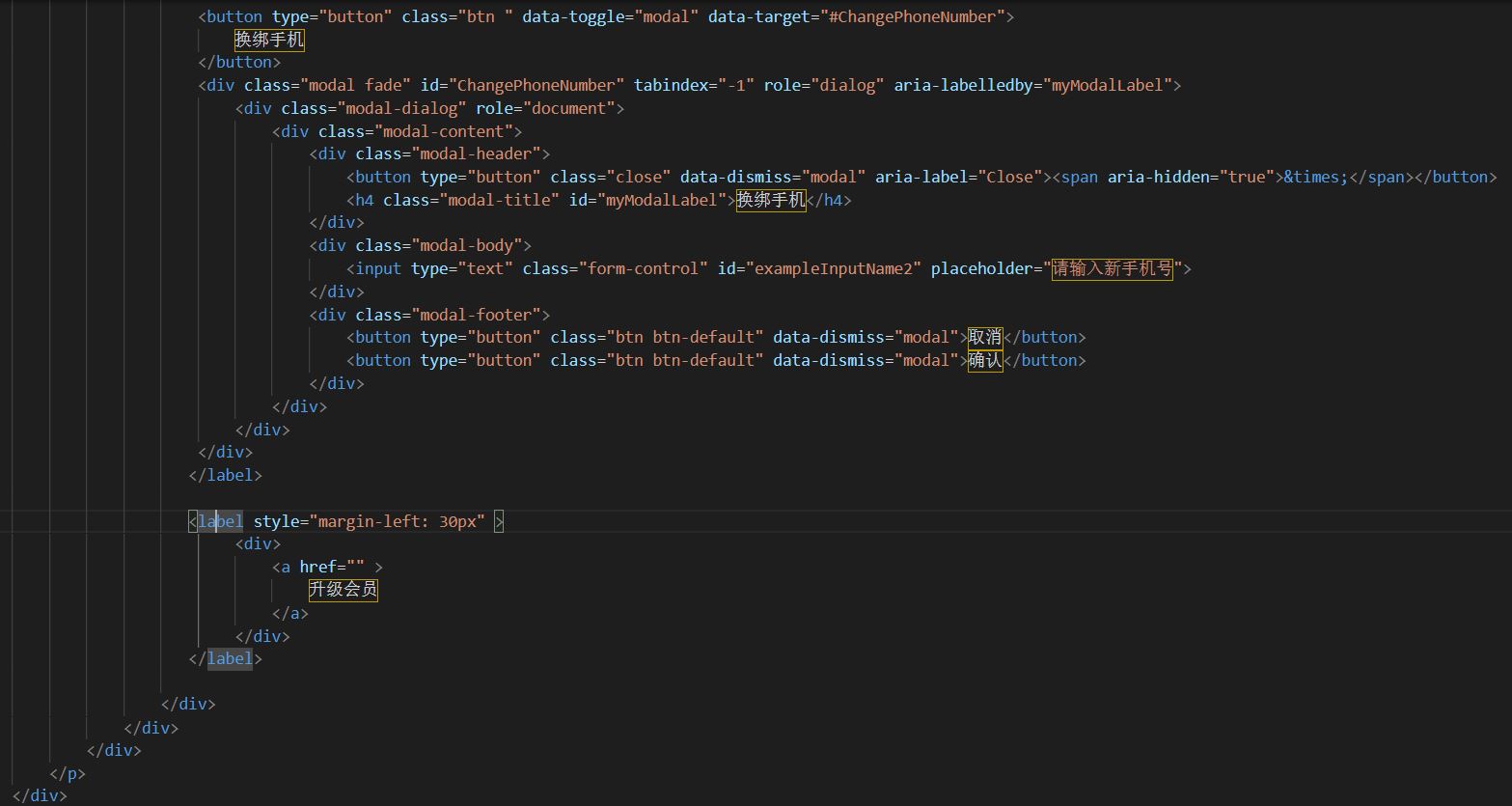


图37



图38



图39



图40

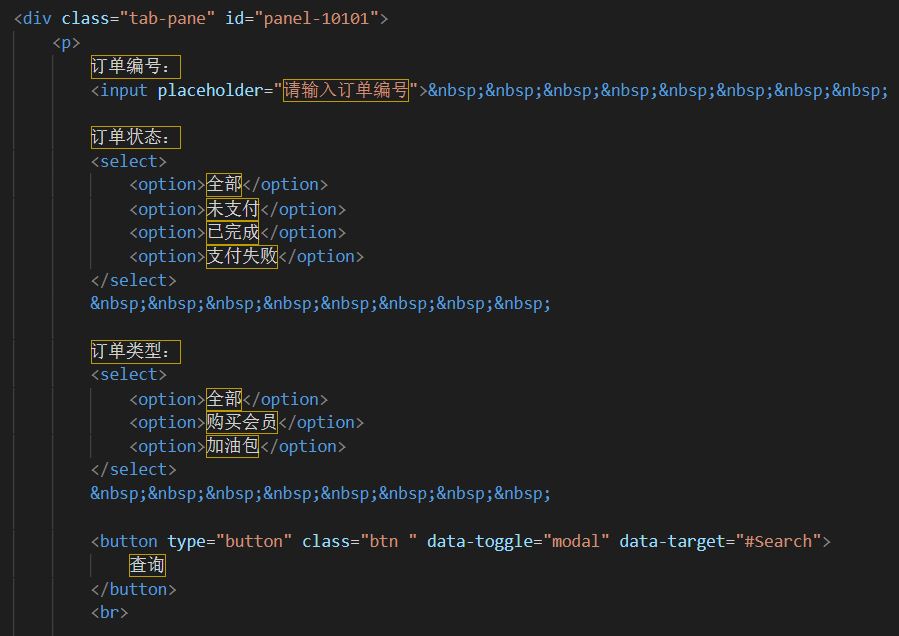


图41

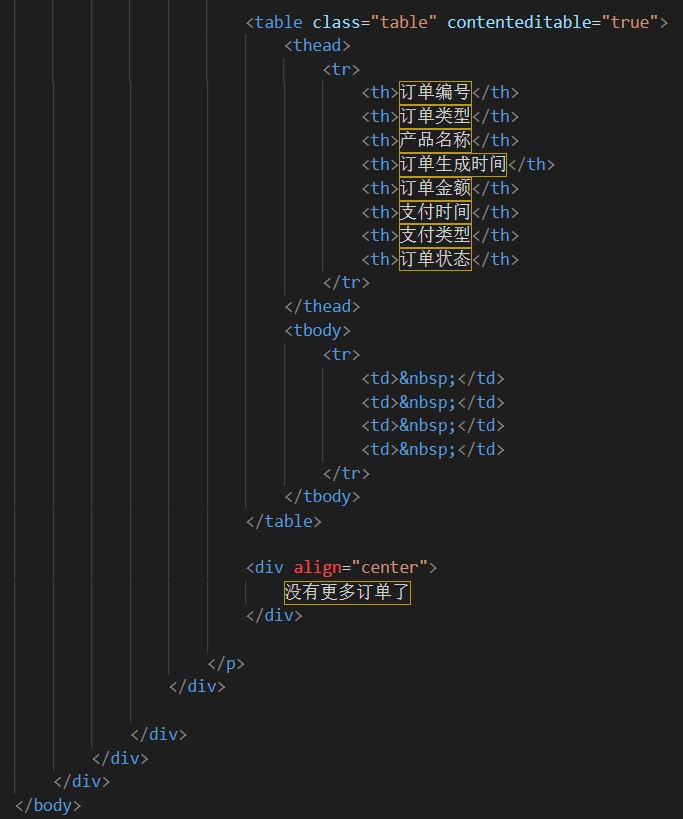
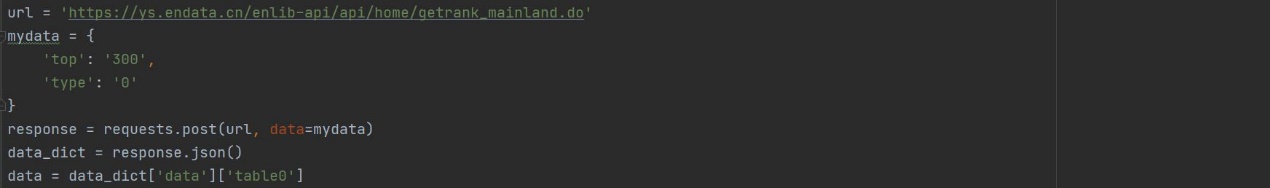


图42

### 4.9 电影数据获取

数据的获取是利用python爬虫技术从电影网站爬取所需的相关电影数据，这里用到了request包来请求数据，如图43。将爬取到的数据解析并使用pymysql包存入到mysql数据库中，实现方法如图44、45。数据库部分数据截图如图46。



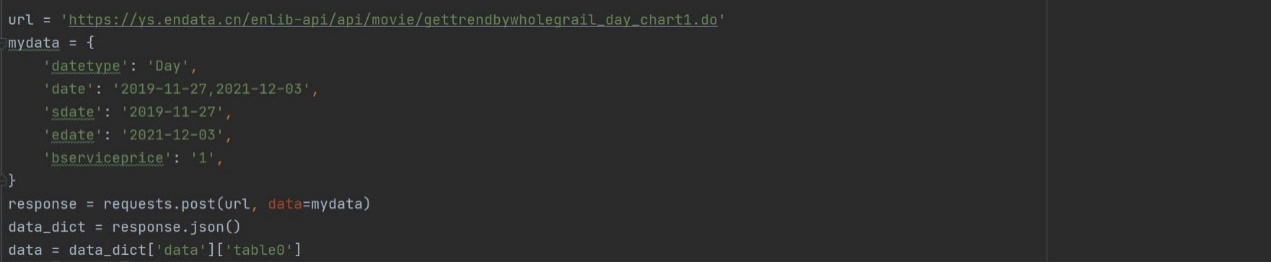


图43

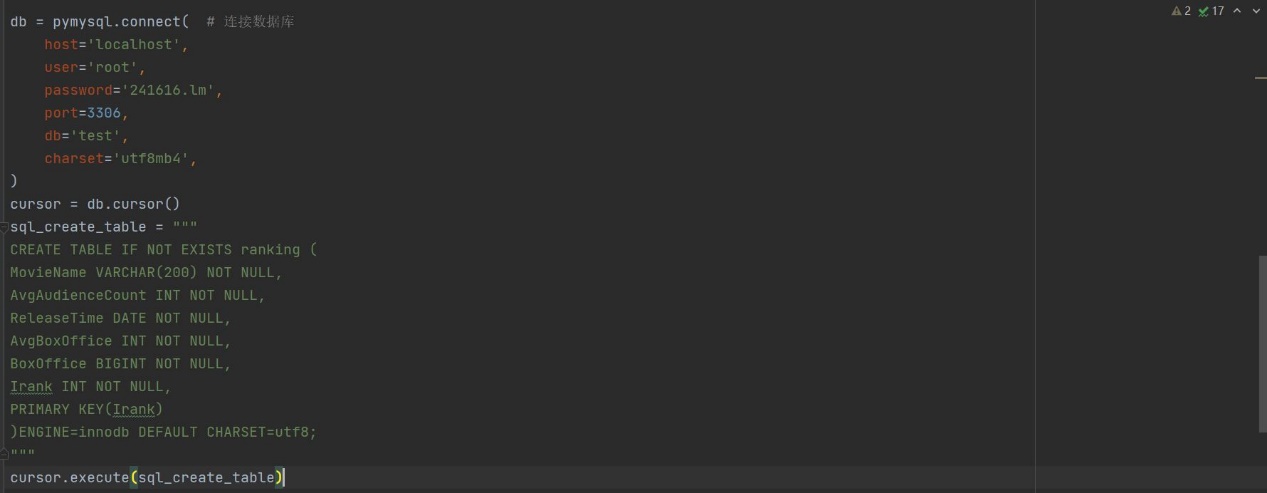


图44

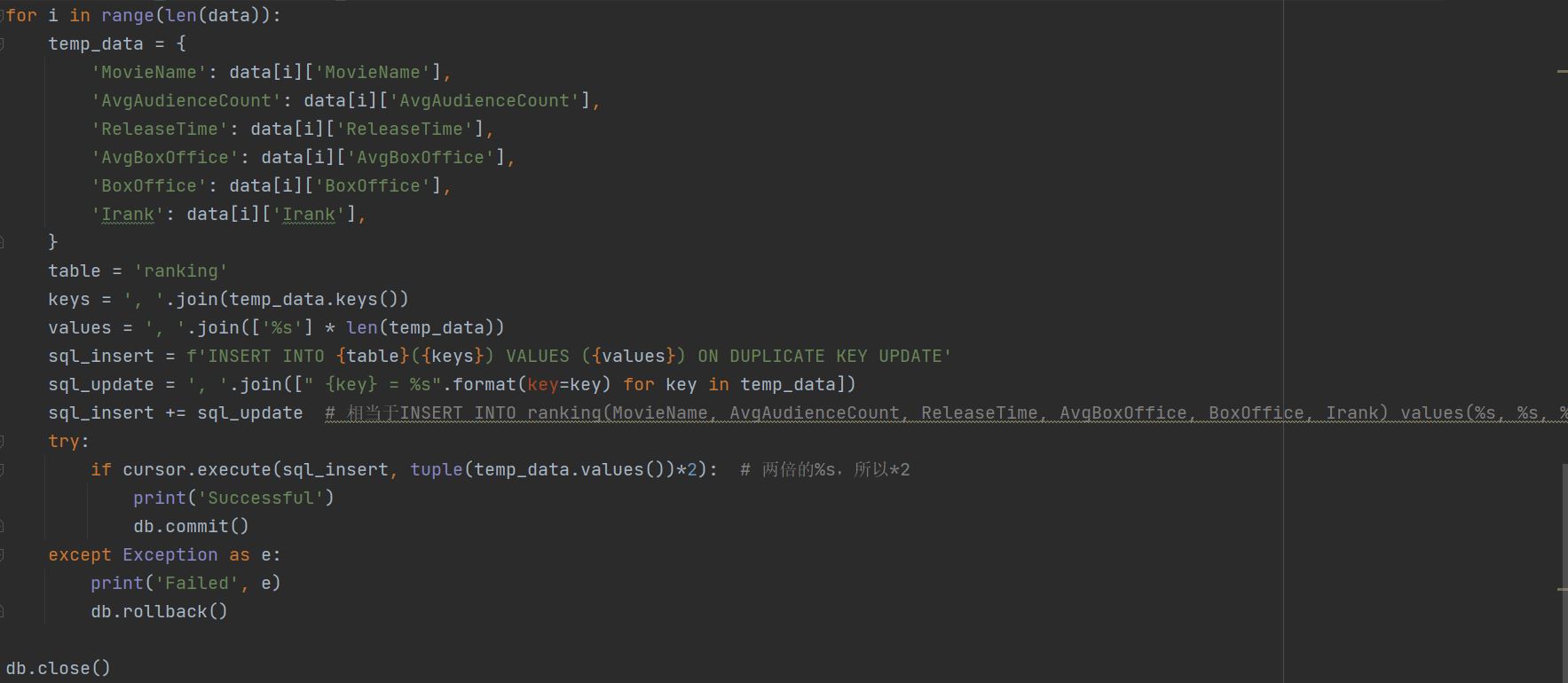


图45

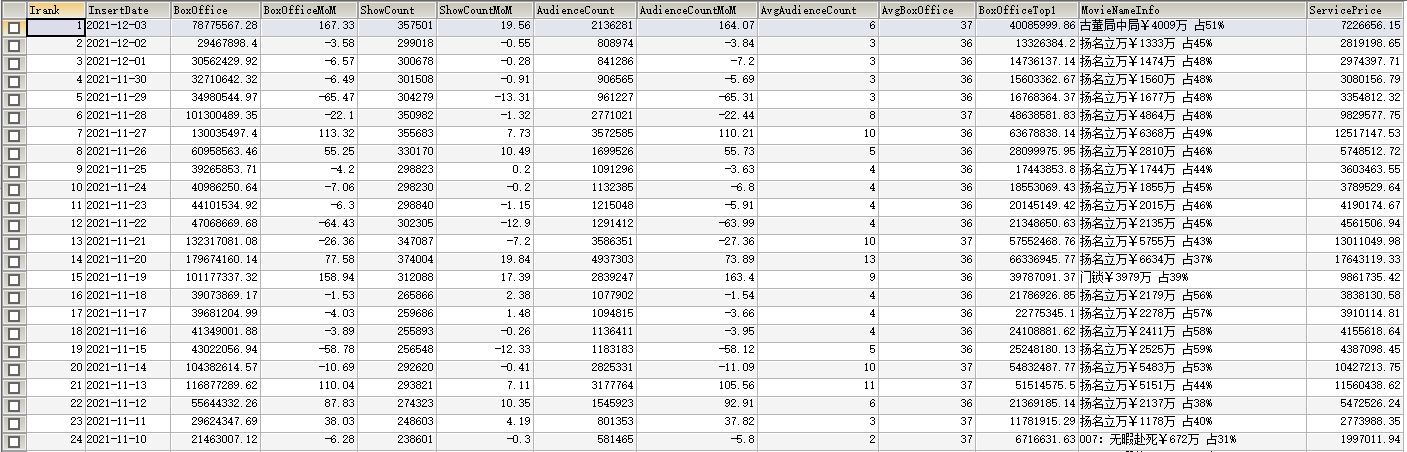


图46

### 4.10 数据库数据的请求

图表数据的获取利用ajax向服务器端发送数据请求，如图47、48。表格所需数据的请求使用了boostrap-table内置的数据请求，如图49。

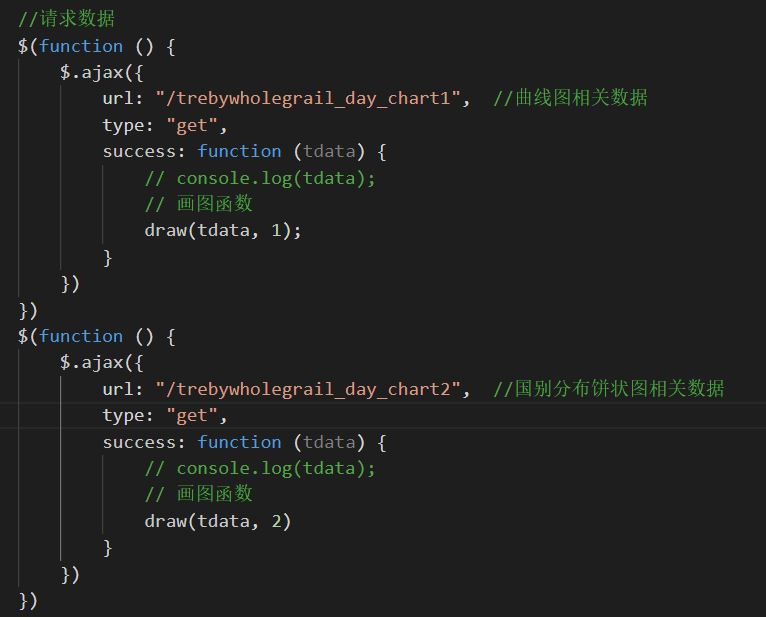


图47



图48

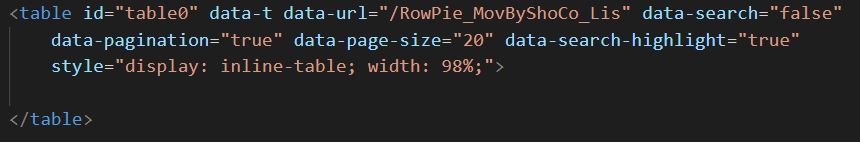
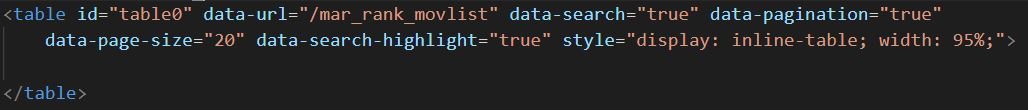


图49

### 4.11 服务器端的数据响应

服务器端获取请求后从数据库中读取数据，对数据进行时间格式化，价格数据小数点保留位数，价格单位转换等处理，如下列图。

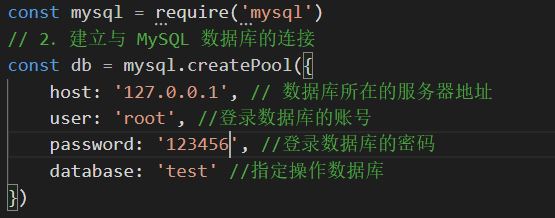


图50

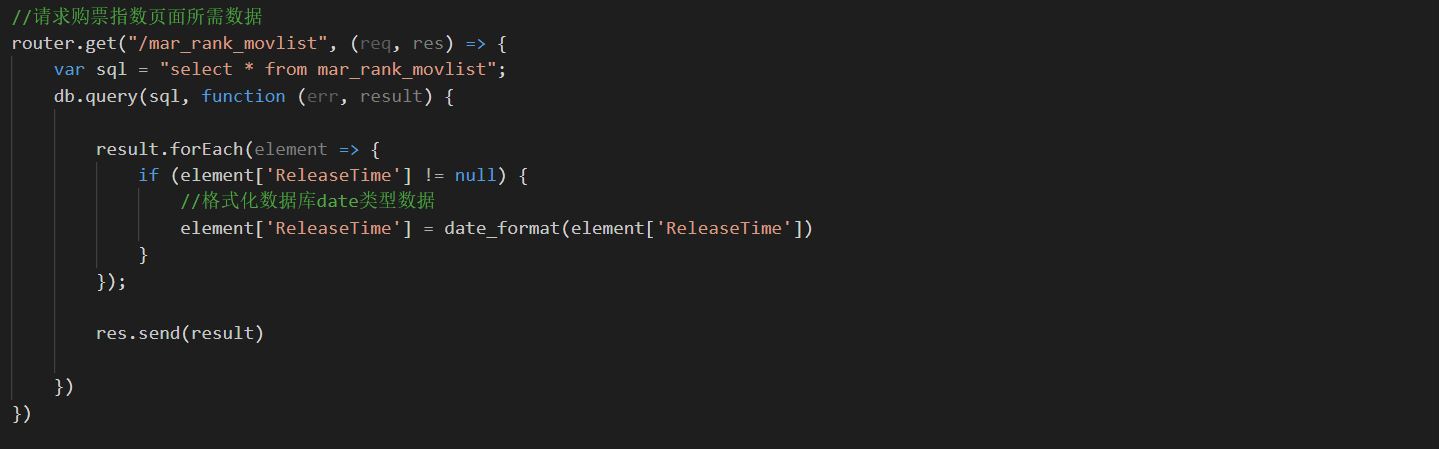


图51

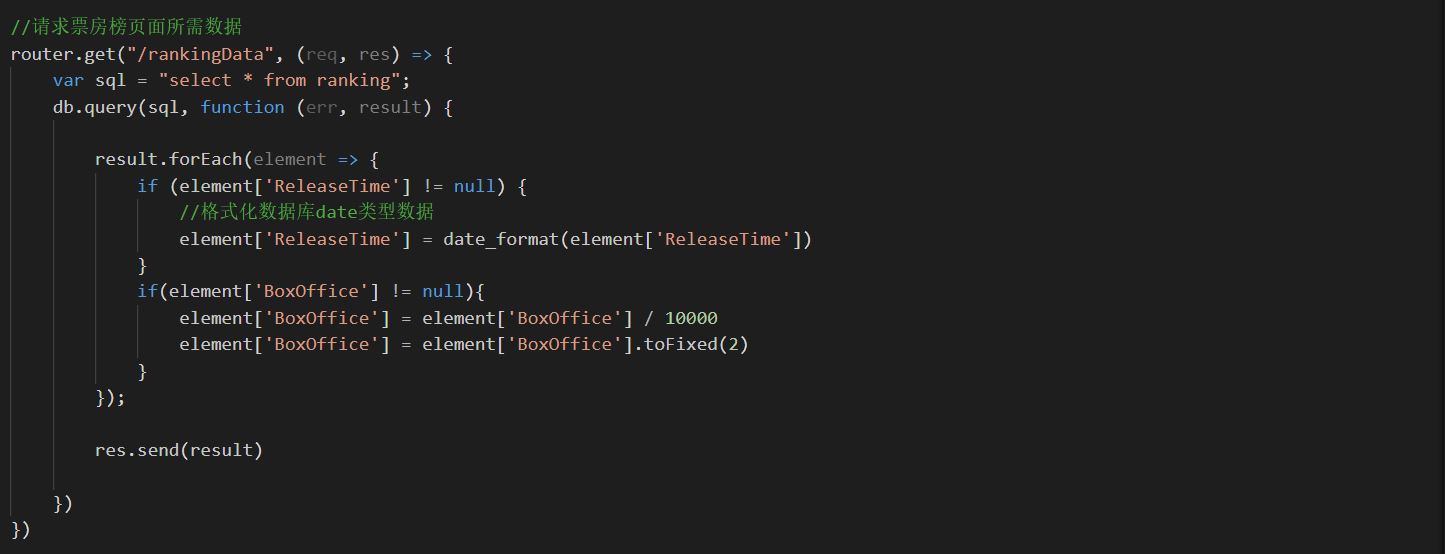


图52



图53

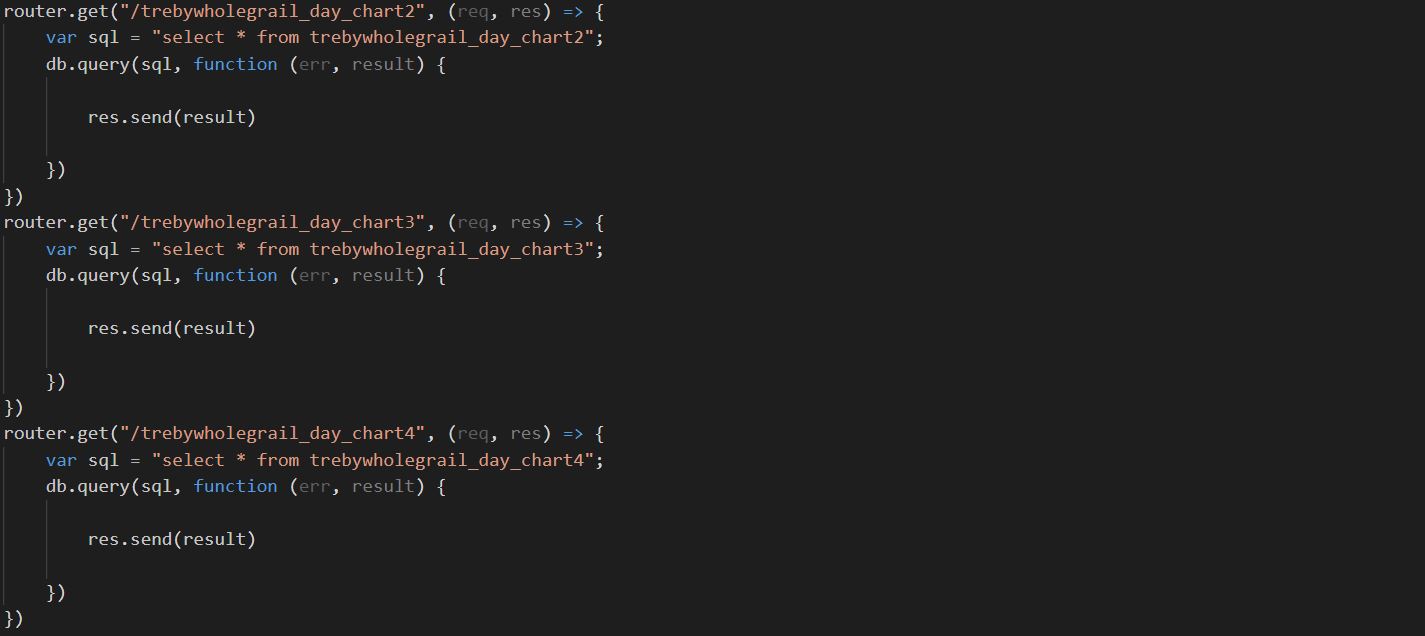


图54

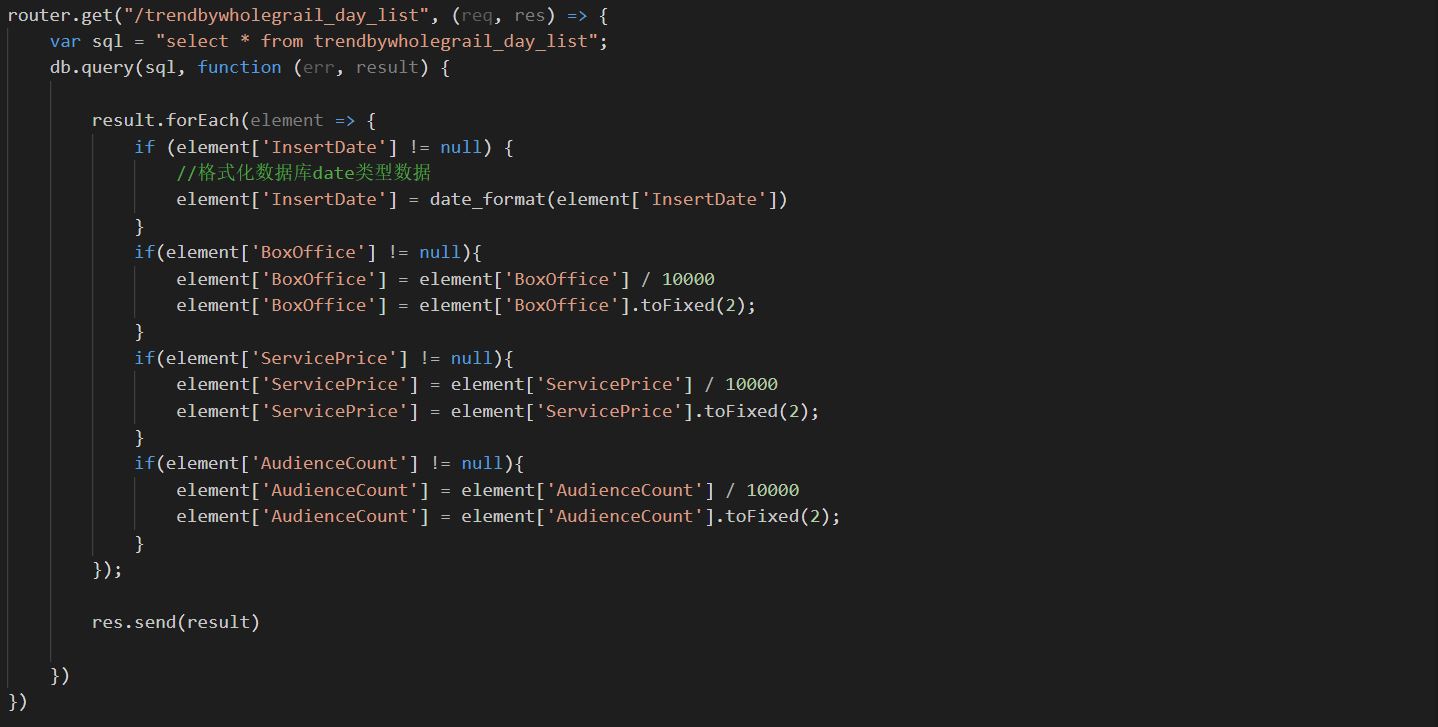


图55

### 4.12 表格的绘制

表格的制作使用了Bootstrap-Table插件，显示数据的方式采用了客户端模式，把数据在服务器端一次性加载出来，然后转换成JSON格式传到要显示的界面中，设置表格可分页。下图都是表格属性的设置，其它表格制作方式一样，如图56、57。数据库表格格式如图58-61。

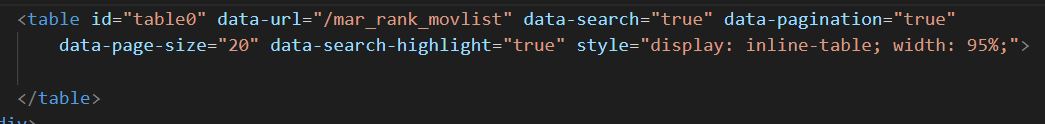


图56



图57

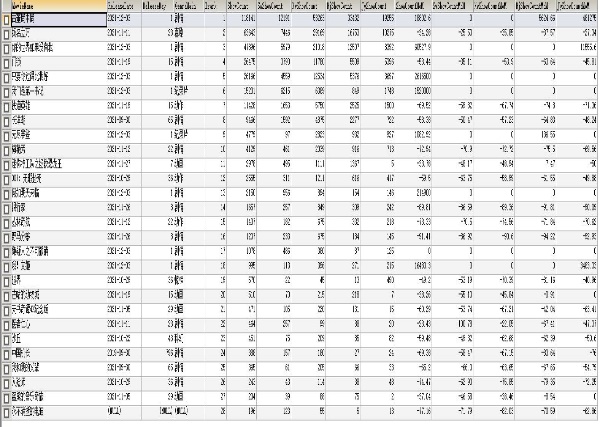


图58

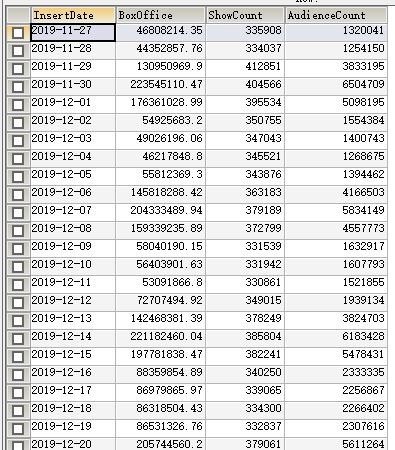


图59

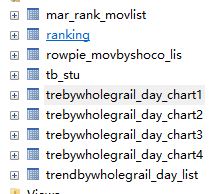


图60

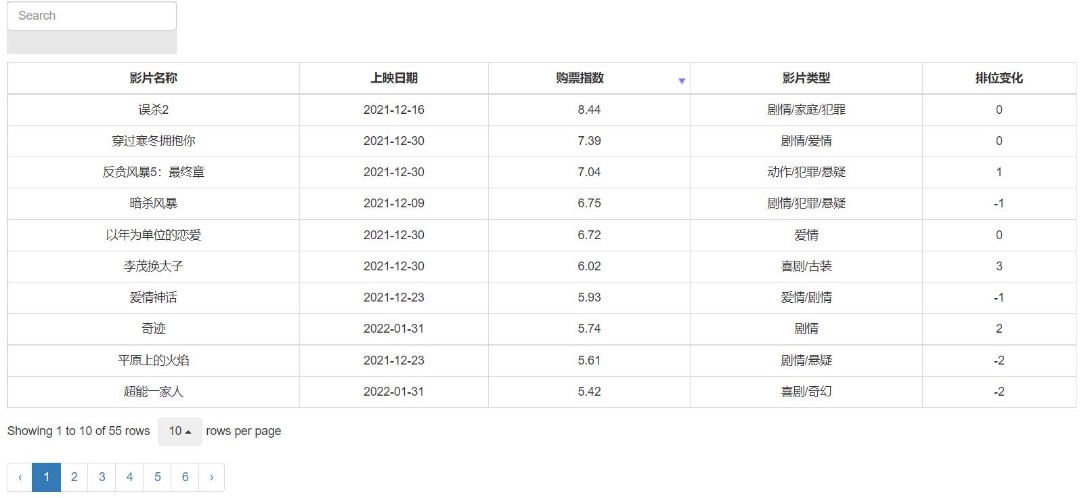


图61