2017-2018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\_计算1502\_\_
* 学生学号\_31501309\_\_
* 学生姓名\_杨嘉诚\_\_
* 实验指导教师:郭鸣

1. 项目分工表格

* 1人项目权重为1.0
* 2人项目权重和为 1.9

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 姓名 | 学号 | 班级 | 任务 | 权重 |
| 杨嘉诚 | 31501309 | 计算1502 | 后台与服务器交互的JS代码，编写路由。 | 0.95 |
| 马之骋 | 31501164 | 计算1502 | 大部分前端代码，负责前后端的串联与交互。 | 0.95 |

2.自我评估表

|  |  |
| --- | --- |
| 函数式编程 | 3 |
| fetch api | 3 |
| FRP | 2 |
| session Storage | 4 |
| 响应式设计(Desktop/Tablet/Mobile) | 5 |
| REST API | 4 |
| 前端BOOTSTRAP框架 | 5 |
| 后台服务器express框架 | 4 |
| 本地服务器部署 | 5 |

3.项目说明

* 1. 项目 是独立开发
     + 实现前端查看不同电影信息功能。
     + 实现不同电影不同场次的调用后台服务器查看座位情况。
     + 实验不同场次的订座，更新后台服务器数据库的功能。
     + 实现管理员权限可以清空某场次座位表的功能。
     + 实现管理员可以添加删除电影的功能，同时添加删除该电影衍生的场次信息
     + 添加了搜索电影功能。
     + 添加收藏电影的功能。

4.解决技术要点说明

* 1. 解决座位表自动生成的问题。关键代码与步骤如下

1. 编写每个座位的CSS代码

.seat-cell {

margin: 0 auto;

width: 10%;

height: 33px;

position: absolute;

/\*background: white;\*/ }

.empty-seat {

background: url(img/emptyseat.png) no-repeat center center;

}

.cant-seat {

background: url(img/cantseat.png) no-repeat center center;

}

.choose-seat {

background: url(img/chooseseat.png) no-repeat center center;

}

1. 编写JS代码来确定每个座位的坐标和样式

for(var i = 0; i < 10; i++)

for(var j = 0; j < 10; j++) {

var nseat = document.createElement("div");

nseat.classList.add("seat-cell");

nseat.id = i \* 10 + j + 1;

switch(status[nseat.id]) {

case '0':

nseat.classList.add("empty-seat");

break;

case '1':

nseat.classList.add("cant-seat");

break;

case '2':

nseat.classList.add("choose-seat");

console.log("impossible");

}

nseat.style.marginTop = (seath \* i+2\*i) + "px";

nseat.style.marginLeft = (seatw \* j+2\*j) + "px";

nseat.addEventListener("click", changestyle);

seats.appendChild(nseat);

}

1. 加上座位的行号

for(var i = 0; i < 10; i++) {

var numnode = document.createElement("div");

numnode.classList.add("numnode");

numnode.innerText = (i + 1);

numnode.style.marginTop = ((seath+2)\* i + 5) + "px";

numnode.style.marginLeft = ((seatw+2) \* -1) + "px";

seats.appendChild(numnode);

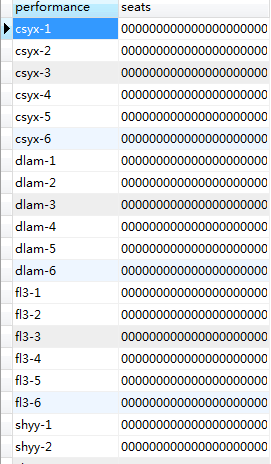
}

最终效果



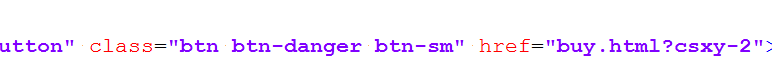
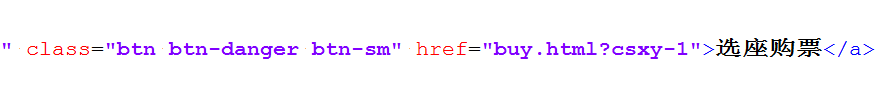
* 1. 解决 从服务器得到数据和修改数据， 关键代码与步骤如下

1. 确定数据库的形式



其中preformance是场次名，seats是用二进制来表示座位的选取情况，0为未被选取，1为被选取。

1. 在前端跳转页面时，在URL中加入当前场次的场次名称。



统一跳转到buy页面后，用JS解析出当前场次名

var url=window.location.search;

url=url.slice(1);

1. 用JS发起对服务器的请求

var req = new XMLHttpRequest();

req.open("GET", "sql/?name="+url, false);

req.send(null);

status = req.responseText;

以上为得到当前场次的座位数据，保存在status中

var data = {name : url , val: status};

req.open("put", "sql", true);

req.setRequestHeader("Content-Type", "application/json");

req.send(JSON.stringify(data));

以上为订票后向服务器传输修改后的座位数据

1. 服务器端的路由解析

app.use('/sql',sqlRouter);

var express = require('express');

var router = express.Router();

/\* GET users listing. \*/

router.get(/\w\*/, function(req, res, next) {

var mysql = require('mysql');

var connection = mysql.createConnection({

host: 'localhost',

user: 'root',

password: '123456'

});

connection.connect();

console.log(req.query.val);

//var qq=["a","b"].join('');

var qq = "SELECT `seats` FROM movie.data where performance='" + req.query.name+"'";

console.log(qq);

var ans;

connection.query(qq, function(err, rows, fields) {

if(err) throw err;

ans = new String(rows[0].seats);

console.log('The solution is: ', ans);

res.send(ans);

});

connection.end();

});

PUT的路由监听

router.put(/\w\*/, function(req, res, next) {

var mysql = require('mysql');

var connection = mysql.createConnection({

host: 'localhost',

user: 'root',

password: '123456'

});

connection.connect();

var qq = "UPDATE movie.data SET seats = " + req.body.val + " WHERE performance ='" + req.body.name+"'";

console.log(qq);

connection.query(qq, function(err, rows, fields) {

if(err) throw err;

});

res.send("success")

connection.end();

});

module.exports = router;

1. 最终效果



* 1. 解决 响应式设计的各类问题， 关键代码与步骤如下

1. 将座位大小尺寸用变量来决定

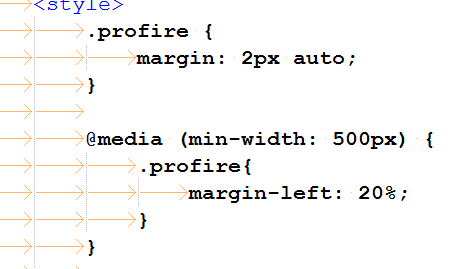
Var status = "";

var seath = 33;

var choosenumber=0;

var seatw = (document.getElementsByClassName("seats")[0].offsetWidth)\*0.097;

1. 响应式设计



实现前段生成服务器传回的电影信息

前端：

let req = new XMLHttpRequest();

req.open("get", "sql/detail", true);

req.onload = function(e) {

var detail = JSON.parse(req.response);

console.log(JSON.parse(req.response));

var panel = document.getElementsByClassName("movie-card")[0];

//console.log(panel);

for(var d of detail) {

var now = panel.cloneNode(true);

var name = d.name;

console.log(d.name);

let id = d.id;

var score = "评分" + d.score;

var director = "导演：" + d.director;

var actor = "演员：" + d.actor;

var comment = d.comment;

var image=d.image;

now.getElementsByClassName("moviename")[0].textContent = name;

now.getElementsByClassName("moviescore")[0].textContent = score;

now.getElementsByClassName("moviedirector")[0].textContent = director;

now.getElementsByClassName("movieactor")[0].textContent = actor;

now.getElementsByClassName("movieimage")[0].src = "img/"+image;

now.getElementsByClassName("moviecomment")[0].textContent = comment;

now.onclick = function() {

//window.location.href="movie\_"+id+".html";

window.location.href = "movie\_detail.html?" + id;

}

//console.log(now.onclick);

var bd = document.getElementById("now");

now.style.display = "block";

bd.appendChild(now);

}

}

req.setRequestHeader("Content-Type", "application/json");

req.send("");

后端：

router.get("/detail", function(req, res, next) {

var mysql = require('mysql');

var connection = mysql.createConnection({

host: 'localhost',

user: 'root',

password: '123456'

});

connection.connect();

if(!req.query.id) {

var qq = "SELECT \* FROM movie.movie\_detail";

console.log(qq);

var ans = [];

connection.query(qq, function(err, rows, fields) {

if(err) throw err;

for(var row of rows) {

var now = getrow(row);

ans.push(now);

}

//console.log(ans);

return res.json(ans);

});

} else {

var qq = "SELECT \* FROM movie.movie\_detail where id='" + req.query.id + "'";

console.log(qq);

var ans = [];

connection.query(qq, function(err, rows, fields) {

if(err) throw err;

for(var row of rows) {

var now = getrow(row);

ans.push(now);

}

//console.log(ans);

return res.json(ans);

});

}

});

实现查找功能

前端：

submit.addEventListener("click", function() {

var search = document.getElementById("search");

var req = new XMLHttpRequest();

req.open("GET", "sql/search?moviename=" + search.value, false);

req.onload = function(e) {

var detail = JSON.parse(req.response);

console.log(JSON.parse(req.response));

var panel = document.getElementsByClassName("movie-card")[0];

//console.log(detail);

var bd = document.getElementById("movies");

removeall(bd);

for(var d of detail) {

var now = panel.cloneNode(true);

var name = d.name;

console.log(d.name);

let id = d.id;

var score = "评分" + d.score;

var director = "导演：" + d.director;

var actor = "演员：" + d.actor;

var image = d.image;

//var comment="影评："+d.comment;

var comment = d.comment;

now.getElementsByClassName("moviename")[0].textContent = name;

now.getElementsByClassName("moviescore")[0].textContent = score;

now.getElementsByClassName("moviedirector")[0].textContent = director;

now.getElementsByClassName("movieactor")[0].textContent = actor;

now.getElementsByClassName("movieimage")[0].src = "img/" + image;

now.getElementsByClassName("moviecomment")[0].textContent = comment;

now.onclick = function() {

window.location.href = "movie\_detail.html?" + id;

}

//console.log(now.onclick);

now.style.display = "block";

bd.appendChild(now);

}

}

req.setRequestHeader("Content-Type", "application/json");

req.send("");

后端：

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

var mysql = require('mysql');

var connection = mysql.createConnection({

host: 'localhost',

user: 'root',

password: '123456'

});

connection.connect();

var qq = "SELECT \* FROM movie.movie\_detail where name like'%"+req.query.moviename+"%'";

console.log(qq);

var ans = [];

connection.query(qq, function(err, rows, fields) {

if(err) throw err;

for(var row of rows) {

var now = getrow(row);

ans.push(now);

}

//console.log(ans);

return res.json(ans);

});

});

实现了图片的上传

var multer = require('multer');

var upload = multer({

dest: 'upload\_tmp'

});

router.post('/insert', upload.single('image'), function(req, res, next) {

fs.rename(req.file.path, "mzc\_html/img/" + req.file.originalname, function(err) {

if(err) {

throw err;

}

console.log('上传成功!');

})

1. 心得体会（结合自己情况具体说明）
   1. 大项目开发过程心得
      * 遇到哪些困难，经历哪里过程，有哪些收获
      * 一开始对于项目的选择有过犹豫，没有想清楚到底如何讲本课程的知识运用到网页设计开发中，但最后在和队友的讨论中决定了题目。
      * 选择了express作为服务器端的框架，花了不少时间来理解其中路由和中间件的原理和使用方法，不过掌握之后这个框架的确加快了项目的开发。
      * 一张网页往往都要涉及前端和后端，在开发之前，我们认为这两者可以完全分离。但在过程中，我们渐渐发现这并不是一件简单的事情，往往还是有不少内容需要前后端进行同步与交互。
      * 经过整个项目周期，我不仅收获了大量的项目开发经验，更是在过程中了解到了更多的前后端网页开发技巧，收益颇丰。
   2. 本课程建议
      * 课程难度方面，进度方面，课程内容，授课方式等，给出你的意见
      * 本课程主要内容为JS语言的掌握和使用，老师上课也讲了许多JS的各类编程思想和方法，虽然难度不低，但的确有所收获。
      * 就我的建议而言，希望本课程多一些与网页制作相关的实践内容，毕竟学习JS语言的最主要目的还是能够运用在网页制作上。