

**毕业设计（论文）**

**题 目：西安工程大学科技处官网全栈开发**

**学 院： 电子信息学院**

**专业班级： 通信工程12级3班**

**指导教师： 朱耀麟 职称： 副教授**

**学生姓名： 张泽轩**

**学 号： 41203030317**

摘 要

西安工程大学科技处是西安工程大学的行政机构及直属单位，主要负责学校的学术研究，科技期刊等。它的官方网站依然用的是很久以前的技术。然而今天技术的门槛下降得越来越快，原本需要一个团队做出来的 Web 应用，现在只需要一两个人就可以了，这就是所谓的全栈开发。今天我就要用所学的知识对西安工程大学科技处的官方网站进行全栈重构。

科技处系统开发主要包括后台数据库的建立，服务器的搭建和前端应用程序三个方面。采用mongodb 数据库作为后台数据库，NodeJs作为系统的开发语言，MVC构架模式，webstorm、mongochef作为开发平台，主要完成官网的首页、机构设置、项目申报与管理、科技成果、规章制度等功能，各个功能是结合显示层与控制层以及业务逻辑层之间的交互，完成了科技处官网后台功能，而前端则是由静态网页构成，前端与后台之间的联系也是由模型层，即与数据库交互获取其中的字段，前台只需加入请求，通过对应后台逻辑，改变路由地址，就可以实现不同页面的跳转。针对西安工程大学科技处开发过程的调试技巧和调试方法做了分析和总结。

关键词：西安工程大学科技处，前端框架，调试，开发模式

**ABSTRACT**

Xi'an Engineering University of science and technology is Xi'an Engineering University Administration and directly affiliated institutions, mainly responsible for the school's academic research and scientific journals. Its official website is still using is a long time ago. However the threshold of technology today fell more and more quickly, the original need a team out of the web application, and now only need one or two people can be, this is called full stack development. Today, I would like to with the knowledge of Xi'an University of Engineering Science and technology at the official website to reconstruct a full stack.

The development of the system of science and technology mainly includes the establishment of the background database, the three aspects of server and front-end applications. Using mongodb database as the background database, NodeJs as the system development language, MVC framework, webstorm, mongochef as the development platform, mainly to complete the official website home page, setup, application and management of projects, scientific and technological achievements the function of rules and regulations, each function is combined with the interaction between the display layer and control layer and business logic layer, the Department of science and technology, website background function, while the front is composed of static web pages, and is also the link between the front end of the background by the model layer, which interacts with the database for the field, the only need to add request, through the corresponding logic, change the routing address, you can achieve different page Jump. According to the process of development of Xi'an Polytechnic University science and Technology Department of the debugging technology Coincidence and debugging methods are analyzed and summarized.

**KEYWORDS:** Science and Technology Department, Xi'an Polytechnic University, front frame, debugging, development mode

# 目录

[第1章 绪 论 1](#_Toc453516126)

[1.1 课题背景 1](#_Toc453516127)

[1.1.1 技处官网介绍 1](#_Toc453516128)

[1.1.2 科技处官网与时俱进 2](#_Toc453516129)

[1.2 课题研究的要求和意义 2](#_Toc453516130)

[1.3 西安工程大学科技处网站全栈开发架构设计研究内容 3](#_Toc453516131)

[1.3.1 西安工程大学科技处官方网站前端技术架构 3](#_Toc453516132)

[1.3.2 西安工程大学科技处官方网站后端技术选择 4](#_Toc453516133)

[1.3.3 数据库选择 4](#_Toc453516134)

[第2章 官网服务器开发 5](#_Toc453516135)

[2.1 科技处官网服务器选择 5](#_Toc453516136)

[2.2 官网服务器开发过程 7](#_Toc453516137)

[第3章 前端开发 11](#_Toc453516138)

[3.1 官网结构 11](#_Toc453516139)

[3.2 官网模块开发过程 11](#_Toc453516140)

[3.3 页面样式 14](#_Toc453516141)

[3.4 首页新闻通知栏开发过程 15](#_Toc453516142)

[3.5 每个页面的左侧导航栏的实现 17](#_Toc453516143)

[3.6 登陆界面开发 19](#_Toc453516144)

[第4章 系统测试与分析 23](#_Toc453516145)

[4.1 科技处官网路由测试 23](#_Toc453516146)

[4.2 科技处官网性能测试 23](#_Toc453516147)

[4.3 科技处官网开发过程所用到的调试方法 24](#_Toc453516148)

[4.3.1 Alert：测试之前的代码是否被执行了 24](#_Toc453516149)

[4.3.2 Console：执行语句的结果是什么 25](#_Toc453516150)

[4.3.3 JS断点调试：错误处理 26](#_Toc453516151)

[第5章 互联网访问科技处官网 27](#_Toc453516152)

[5.1 公网映射 27](#_Toc453516153)

[5.2 映射工具映射 27](#_Toc453516154)

[5.3 互联网访问科技处官网 28](#_Toc453516155)

[第6章 结论 29](#_Toc453516156)

[附 录 31](#_Toc453516157)

[参考文献 43](#_Toc453516158)

[致 谢 44](#_Toc453516159)

# 绪 论

## 课题背景

西安工程大学科技处是西安工程大学的行政机构及直属单位，主要负责学校的学术研究，科技期刊等。截至2016年4月底，西安工程大学拥有教育部重点实验室1个，还有省级重点实验室1个，还有省级2011协同创新中心1个，还有省级工程技术研究中心5个，还有省级哲学社会科学研究基地1个，还有校级研究发展中心1个。这么重要的一个行政机构及直属单位，官方网站却是几年前的版面。在对外宣传时严重影响到了学校的整体形象。为了提高西安工程大学科技处官方网站的网站性能，我使用现如今市面上比较流行的web技术，对西安工程大学科技处官方网站进行了重构。为的是让西安工程大学科技处官方网站也能与时俱进。

### 科技处官网介绍

一个官方网站是一个部门的门面，现有的西安工程大学科技处官方网站（http://keyanc.xpu.edu.cn/）是由陕西新势力网络科技有限公司制作，从2012年开始到现在已经有四年时间了。这期间一直由科技处自己维护，但是由于后期维护困难，到现在部分页面有错版，平台还有一些BUG(错误)。如图1-1所示，首页的通知栏布局错乱。



图1-1 科技处官方网站首页截图

部分页面的导航布局错乱，如图1-2所示。



图1-2 科技处官方网站导航排版错误截图

### 科技处官网与时俱进

互联网十多年以来，风云变幻、日新月异，新的东西层出不穷。然而西安工程大学科技处官方网站还是原始的TABLE布局。

TABLE布局是在WEB早期CSS不存在的时候才兴起的，它是对TABLE标签的一种不正规的使用，Table标签就是表格布局，是用来显示网页数据的，而不是用来布局网页的，尽管它有时候布局网页确实很简单[1]。

到目前为止，绝大多数的网站都已经采用了DIV+CSS布局来排版了。

## 课题研究的要求和意义

课题研究的要求：

1. 仿照现在的官网UI进行开发
2. 功能和现有的官网功能不能出现大的变化。

课题研究的意义：

1. 给出解决西安工程大学科技处官方网站代码老，排版错乱问题的方法，使西安工程大学科技处官方网站成功平滑过渡到现在主流的平台上；
2. 对前端新技术进行分析，探讨西安工程大学科技处官方网站所属类型，以及用所学的基础理论对采取的技术做一个选型；

## 西安工程大学科技处网站全栈开发架构设计研究内容

直到今天，也没有一个全栈工程师的完整标准，有一千个开发者，心中就有一千个全栈的定义。有的把全栈工程师视为什么都懂，什么都不精通的勤杂人员，有的开发者以自己为全栈工程师为荣。从创业公司到大公司，都有各自的解读[2]。我从我对全栈开发的角度谈一下。

全栈开发主要是由前端开发，服务器端开发，数据库开发三部分组成，看似独立的三个部分，却有着千丝万缕的联系，一个模块儿的架构设计不理想，就会导致整个平台运作的失败[3]。西安工程大学科技处官方网站的全栈开发也不例外，它也有自己的服务器，也有自己的前端页面，也有自己的数据库。

### 西安工程大学科技处官方网站前端技术架构

环顾四周的大型互联网公司，基本上都有自己的前端框架，比如阿里的Kissy和Arale，腾讯的JX，百度的Tangram，360的QWrap等，为什么？因为要整合别的框架，并且在此基础上发展适合自己的组件库，代价非常大，初期没办法的时候只能凑合，长期来说，所有代码都可控的意义非常重要。反观西安工程大学科技处官方网站，大的版面设计基本相同，每个页面不同的是每次的导航和内容。那么我们就可以自己开发自己的一套前端库。

那么，是不是一套框架可以包打天下呢，这个真的很难。对于不同的产品形态，如果想要用一套框架去适应，有的会偏轻，有的又偏重，有的要兼容低端浏览器，有的又不要，很难取舍。而我们自己开发的专门为科技处使用的框架就没有了这些问题。

对西安工程大学科技处官方网站全栈开发，除了要做在浏览器中运行的代码，还写一些它的服务端的代码，这个需求又是从哪里来的呢？

这个需求其实来自对官网的优化。我们要优化西安工程大学科技处官方网站的前端部分，从以下这些方面开始做：

* 它的HTML结构的优化，还可以减少DOM树的层次
* CSS渲染性能的优化，批量写入DOM变更之类
* 资源文件的优化，比如整个网站一级导航栏小图片的合并，网站里图像格式的处理，最新的技术图标字体的使用等
* JavaScript逻辑的优化，模块化，异步加载，性能优化
* HTTP请求的优化，比如合并css文件；合并js文件等

这里面，除了前三条，其他都可能跟后端有些关系，尤其是最后一条。如果不懂全栈开发，前端的人就没法去优化后端的东西，因为这是不同的协作环节。

### 西安工程大学科技处官方网站后端技术选择

西安工程大学科技处官方网站是一个主数据交互的网站，从数据库拿到数据，再展现到前台的过程比较多，所以选择了现在市面上的主流的服务器语言，NodeJS。Node.js是一个开放源代码、跨平台的、可用于服务器端和网络应用的运行环境，它支持OS X、Microsoft Windows、Linux、FreeBSD、NonStop、IBM AIX、IBM System z和IBM i，Node.js由Node.js基金会拥有和维护，该基金会与Linux基金会有合作关系[4]。

西安工程大学科技处官网使用Node.js，因为nodejs可以提供事件驱动和非阻塞I/O API，还能提高应用程序的吞吐量和规模。其次Node.js采用Google的V8引擎来执行代码，如果科技处官网使用它，可以很大的提高代码运行速度。其次，Node.js的大部分基本模块都是用JavaScript写成的。Node.js含有一系列内置模块，使得程序可以作为独立服务器运行，从而脱离Apache HTTP Server或IIS运行，这样可以为科技处官网提供更稳定的服务[5]。

### 数据库选择

因为科技处官网的服务器选择的是nodeJS，为了能和nodeJS更好的配合使用，就选择了MongoDB数据库，它使用的是json的数据格式，而json又是从javascript的对象借鉴过来的，所以数据库最后就选择了MongoDB。MongoDB是一种文档导向数据库管理系统，它是由C++撰写而成的，这样一来可以解决应用程序开发过程中的大量现实问题[7]。

# 官网服务器开发

网页服务器（Web server）这一个词它有两个意思：

第一个：一台负责提供网页的电脑，主要由各种编程语言构建而成，通过HTTP或者HTTPS协议传给客户端（一般是指网页浏览器）。

第二个：一个提供网页的服务器程序。

每一台网页服务器（第1个意思）会运行最少一个网页服务器程序（第2个意思）。作为第1个意思，大陆地区称为网站服务器。此外，也通称作Web服务器。

## 科技处官网服务器选择

在做毕设，有时会发现自己在不断构建相同的页面；这是我就考虑到要采用一种服务器端技术。一开始觉得科技处官网属于小型站点并且更改不太频繁，就使用了Adobe Dreamweaver 的模板，它是这个样子的,如图：2-1



图3-1 Dreamweaver构建

但是Adobe Dreamweaver模板有一个缺点，就是需要单独构建每个页面并将它上传到远程服务器。如果需要更改模板的时候，则需要重新上传每个文件，所以就放弃了使用这种方式。

再后来开发科技处官网时使用了服务器端技术，服务器端技术可以从最简单的层面允许我使用一个外部文件，将相同的代码包含在多个页面里。例如，科技处官网里，将路由文件放入到了一个外部文件。在就是导航栏放到一个文件里，每当更新导航时，我只需将这个文件上传到 Web 服务器，它就会自动更新每个页面[8]。这个与使用模板十分相似，但是无需在每次更改时逐一上传每个文件。因此，它更快、而且更容易维护、修改。

服务器开发有服务器的开发语言，现在市面上比较多的是java、PHP、.net、nodeJs。这些语言都可以作为科技处官网的开发语言，但是去选择使用哪种服务器端技术是一个重要的决策。我无法说哪一种技术更好，但每种技术的这些方面可以帮助我做出决定，我选择nodejs作为科技处官网的服务端语言是因为它的易学性、可用性、成本以及软件支持。NodeJS服务器的项目结构如图2-2所示：

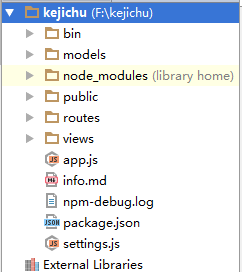


图2-2 官网服务器结构

官网服务器构建很简单，首先我们得安装nodejs。

nodeJS下载地址：<https://nodejs.org/en/>

安装还后我们启动node应用程序，只需简单的几行代码，就可以构建一个我们需要的服务器了，代码如下

express name –ejs cd name && npm installnpm start // 启动服务器

这时我们的服务器就构建完成了。

## 官网服务器开发过程

整个官网服务器的结构如下：

服务器的开发使用的工具是webstorm，此IDE可以更好的对官网服务器做开发，它可以对项目结构有更好的展现。

路由文件是在app.js文件里调用的，它的代码是这样的：

var routes = require('./routes/index');

var users = require('./routes/users');

var guizhangzhidu = require('./routes/guizhangzhidu');

var indexInfo = require('./routes/indexInfo');

var jigoushezhi = require('./routes/jigoushezhi');

var kejichengguo = require('./routes/kejichengguo');

var keyanjidi = require('./routes/keyanjidi');

var xiangmushenbao = require('./routes/xiangmushenbao');

var xuebaobianji = require('./routes/xuebaobianji');

var xueshuhuiyi = require('./routes/xueshuhuiyi');

var yijianfankui = require('./routes/yijianfankui');

这些代码控制着官网里的路由模块的加载。在主文件里调用到路由文件夹里的各个路由模块。整个页面的路由又是由自己的独立的路由代码控制。各自的路由被整合到了一个被命名为router的文件夹下，路由文件如图2-3

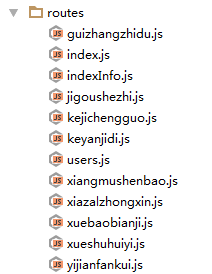


图2-3 路由文件

我们以首页路由文件为例，首页路由文件代码如下：

var yijianfankui = require('./routes/yijianfankui');

var express = require('express');

var router = express.Router();

var mongoose = require("mongoose");

var db = mongoose.connect("mongodb://127.0.0.1:27017/kejichu");//链接到数据库test集合上；

var TestSchema = new mongoose.Schema({

div:{type:String},

name: {type: String}

});

/\* GET home page. \*/

/\*机构设置前台路由规则\*/

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

//数据库操作

var Model = db.model("jigoushezhis", TestSchema);

Model.find({div:"left"},{\_id:0,div:0}, function (error, docs) {

if(error){

console.log("error :" + error);

}else{

res.render('jigoushezhi',{title:docs[0].name,title2:docs[1].name,title3:docs[2].name});

}

});

});

module.exports = router;

这些代码主要实现的功能是，在首页时点击机构设置页面的跳转链接时，加载机构设置页面的过程。第5行代码是链接到我们启动的mongodb数据库27017端口上。第12行开始是路由跳转路径的设置，我们在获得get请求时，返回的路由是'/page/jigoushezhi'，返回的数据是jigoushezhi这个模板里的文件。

说到模板，官网的模板全部被整理到了项目目录的一个叫views的文件夹下了。模板文件结构如图2-4：

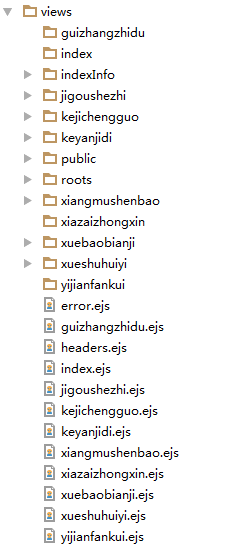


图2-4 模板文件结构

模板里会用到一些数据，项目也会用到一些图片、框架、类库等，这些文件都被整理到了public文件夹下，public文件夹结构如图2-5：

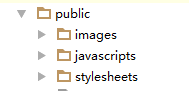


图2-5 public文件夹结构

# 前端开发

因为Web前端技术兴起的时间不长，所以还没有明确的界限定义，在不同的Web项目中，要求的Web前端开发技术会也会有所不同。

针对西安工程大学科技处官网这个项目而言，需要既了解前端开发，又了解一些后端技术，这样前端开发才可以更好地与后端开发配合，比如在页面上留下一些后端需要调用的钩子等，其次，此项目还需要懂一些UI设计、Photoshop工具的使用方法等，以便于设计出更好的UI界面。

## 官网结构

## 官网模块开发过程

官网的主要由html代码构成。但是由于科技处官网的页面HTML代码结构基本固定，它的HTML的标签数量也不多。从布局上来说主要分为了四大块。他慢分别是banner+nav导航部分，左侧导航部分，主要内容显示区部分，底部footer部分；

结构实际开发如图3-1所示:



图3-1 html结构图

这些样式代码都写到了ejs模板里了，在这四大部分里，有三部分是每个页面都会用到的，所以我把他们单独的提出来，写成了公共的模板。公共模板如下图3-2:

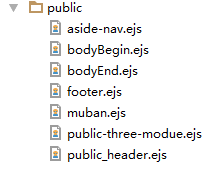


图3-2 公共模板文件

以导航条为例：它的代码是这样的：

<div id="div1" class="div1">

<img usemap="#pap2" src="/images/index/index\_01.jpg" width="1002" height="177" style="display: block">

<map name="pap2" id="pap2">

<area shape="rect" coords="917,8,1004,32" href="http://www.xpu.edu.cn/" /><!--地址是西安工程大学网址-->

</map>

</div>

<div id="div2" class="div2">

<img src="/images/index/index\_02.jpg" width="1002" height="39" border=0 usemap="#pap" style="display: block"/>

<map name="pap" id="pap">

<!--onFocus是点击后的显示效果-->

<area shape="rect" coords="33,0,78,39" href="/index" id="area1" onFocus="styleB(this.id)"/>

<area shape="rect" coords="98,0,162,39" href="/page/jigoushezhi" onFocus="this.blur()"/>

<area shape="rect" coords="188,1,299,40" href="/page/xiangmushenbao" onFocus="this.blur()"/>

<area shape="rect" coords="320,0,384,39" href="/page/kejichengguo" onFocus="this.blur()"/>

<area shape="rect" coords="414,0,478,39" href="/page/keyanjidi" onFocus="this.blur()"/>

<area shape="rect" coords="500,0,564,39" href="/page/guizhangzhidu" onFocus="this.blur()"/>

<area shape="rect" coords="589,2,697,37" href="/page/xueshuhuiyi" />

<area shape="rect" coords="718,2,798,37" href="/page/xuebaobianji" />

<area shape="rect" coords="823,2,891,37" href="/page/xiazaizhongxin" />

<area shape="rect" coords="911,2,978,37" href="/page/yijianfankui" />

</map>

</div>

这些代码实现的布局如下图3-3代码视图：



图3-3 代码视图

这段代码使用的方式是：在需要添加次样式图的地方写下如下代码即可被调用实现视图：

<% include public\_header %>

## 页面样式

页面样式即CSS， 是 Cascading Style Sheets(层叠样式表)的简称。

官网里的css样式都在整理到里公共文件夹下，文件目录如图3-4：

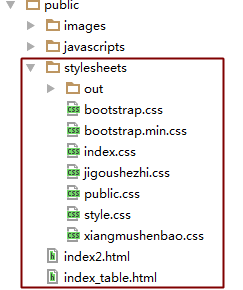


图3-4 css样式文件目录

在标准页面设计中，CSS负责网页内容的表现，所以CSS也是前端开发需要掌握的核心内容。

Css的代码样式样例：

.lump{

display: inline-block;

float: left;

width: 10px;

height:20px;

margin:10px;

background: blue;

}

.nav-name{

display: inline-block;

float: left;

font-size: 13px;

font-weight: bold;

}

.nav-address{

display: inline-block;

float:right;

font-size: 13px;

margin-right:10px;

}

.info{

margin: 20px;

}

丰富的CSS样式能让平淡的科技处官网的HTML展现出绚丽的效果，使得页面更加友好。

一个好的样式可以让用户在科技处官网页面上停留的时间更久一些，也可以帮助用户更好地阅读科技处官网网站内容，同时，还可以让用户在不同浏览器上有着相同的体验。科技处官网的CSS和它的HTML代码一样，没有复杂的逻辑，上手也比较容易。其主要的难点在于如何合理利用CSS的组合和继承特性来编写简洁、可维护性好的CSS代码。

## 首页新闻通知栏开发过程

整个平台每个页面都用到了javascript，以官网首页的通知栏为例。通知界面如图3-5:



图3-5 首页通知栏

此模块主要是通过javascript控制前端页面，每隔一定时间进行一次跳变，而且在跳变的过程中会有运动过度的效果，右侧的按钮会和内容对应变化。此模块是基于jquery框架开发。jQuery在科技处官网中地位很高，整个网站的javascript基本是由jQuery构建的。jQuery它是一套跨浏览器的JavaScript库，简化HTML与JavaScript之间的操作[9]。由约翰•雷西格（John Resig）在2006年1月的BarCamp NYC上发布第一个版本。目前是由Dave Methvin领导的开发团队进行开发。在全球的前10,000个访问量最高的网站中，有65%的网站使用了jQuery，是目前最受欢迎的JavaScript库。

此模块代码如下：

//新闻滚动条

function scrollNews(selector, Entry, time, StartIndex) {

var \_self = this;

this.selector = selector;

this.Entry = Entry;

this.time = time;

this.i = StartIndex || 0;

this.Count = $(this.selector + " ul li").length;

$(this.selector + " ul li").hide();//全部隐藏

$(this.selector + " ul li").eq(this.i).show();//第i个显示

$(this.selector).bind("mouseenter", function () {

if (\_self.sI) {

clearInterval(\_self.sI);

}

}).bind("mouseleave", function () {

\_self.showIndex(\_self.i++);

})

/\*生成激活OL项目\*/

for (var j = 0; j < this.Count; j++)

$(this.selector + " .activeOL").append('<li><a onclick="' + this.Entry + '.showIndex(' + j + ');" href="#"><img src="images/out/crystal.gif"></a></li>');

$(this.selector + " ol li a").eq(this.i).addClass("active");

this.sI = setInterval(this.Entry + ".showIndex(null)", this.time);

this.Getselector = function () {

return this.selector;

}

this.showIndex = function (index) {

this.i++;//显示下一个

if (this.sI) {

clearInterval(this.sI);

}

this.sI = setInterval(this.Entry + ".showIndex()", this.time);

if (index != null) {

this.i = index;

}

if (this.i == this.Count)

this.i = 0;

$(this.selector + " ul li").hide();

$(this.selector + " ul li").eq(this.i).slideDown();

$(this.selector + " ol li a").removeClass("active");

$(this.selector + " ol li a").eq(this.i).addClass("active");

}

}

要显示的新闻内容都会存储在一个数组中，当scrollNews函数被调用的时候，此数组会以参数的形式传入被调用的函数里，函数会判断参数的length属性，以此确定for循环的循环长度。

## 每个页面的左侧导航栏的实现

机构设置页面的左侧导航栏如图3-6：

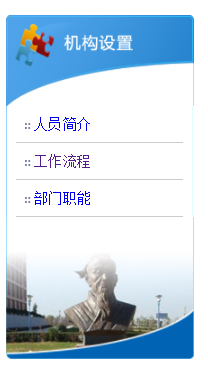


图3-6 机构设置页面的左侧导航栏

首先分析此布局，由三部分组成，分别是上部图片、中间导航内容和下部图片组成，我先用html搭建公共模板：代码如下：

<div id="aside-nav-warp" class="aside-nav-warp">

<img class="aside-nav-top" src="" style="display: block">

<ul class="aside-nav" id="aside-nav">

</ul>

<img class="aside-nav-bottom" src="">

</div>

我们可以发现，我在添加图片的时候都是给img标签的src属性都是一个空字符串，这里不同的页面所调用的图片的地址是不一样的，所以我就把这里给了一个接口，而每次的图片地址都是动态获取。这里就用javascript配合，代码如下：

<script>

function greatAsideNav (num,arr,href,imgSrc){

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

var $a = $('<a>').attr('href',href[i]).attr('class','aside-nav-a');

var $li = $('<li>'+arr[i]+'</li>');

$li.attr('class','aside-nav-li');

var $img = $ ('<img>');

$img.attr('class','aside-nav-img').attr('src','/images/xiaojiantou-left.jpg');

$li.append($img);

$a.append($li);

$('#aside-nav').append($a);

}

$('.aside-nav-top').attr('src',imgSrc[0]);

$('.aside-nav-bottom').attr('src',imgSrc[1]);

}

</script>

这个函数的功能是当函数被调用的时候，就做这些事情：

1. 根据传入此函数的第一个参数判断，本页面会有几条导航，就动态创建几条导航

2. 根据传入此函数的第二个参数给每一条导航赋值

3. 根据传入此函数的第三个参数给每一条导航添加跳转的链接

4. 根据传入此函数的第三个参数给上下图片的src属性赋值，以便于此页面显示自己页面的图片

5. 最后将这些内容返回给调用此函数的对象。

## 登陆界面开发

登陆界面是运营平台的登陆入口，这个页面的的作用就是对平台管理权限的一个限制[10]。登陆界面的样式如图3-7



图3-7 登陆界面

登陆页面的开发由一张背景图片加一个form表单标签组成，代码如下：

</script><!DOCTYPE html>

<html>

<head lang="en">

<style>

\* {margin:0; padding: 0;}

body {background: url("/images/root/reg\_bg.jpg") repeat-x;}

.form {width: 579px; height: 415px;margin: 100px auto;/\* background: url("images/8\_1\_04.jpg");\*/ position: relative;}

.form input { font-size: 18px; border: 0;padding-left: 8px;color: #333333;background: transparent;}

.txt1 { position: absolute; top: 164px;left: 191px;width: 265px;height: 39px;-moz-border-radius: 6px;-webkit-border-radius: 6px;border-radius: 6px;}

.txt2 { position: absolute; top: 218px;left: 191px;width: 265px;height: 39px;-moz-border-radius: 6px;-webkit-border-radius: 6px;border-radius: 6px;}

</style>

<meta charset="UTF-8">

<title></title>

<script type="text/javascript">

</script>

</head>

<body>

<form id="form" class="form">

<img src="/images/root/reg\_bg2.jpg" usemap="#img">

<map id="img" name="img">

<area shape="rect" coords="196,290,295,328" href="/roots/index" alt="denglu">

<area shape="rect" coords="359,290,457,328" href="javascript:" alt="denglu">

</map>

<input type="text" name="name" class="txt1">

<input type="password" name="password" class="txt2">

</form>

</body>

</html>

# 系统测试与分析

最近正好在看一本书《软件测试之魂》，书中提及软件测试的目的就是发现软件的缺陷，而从测试工程师的角度分析，测试目的是：发现程序中的错误、降低代码出错风险、提高代码质量[11]。从公司的角度分析：降低公司成本调试方案调试结果与分析

## 科技处官网路由测试

路由测试其实就是测试平台上的每一个链接都保证能到它该链接的地方，而测试的方法有黑盒测试，和白盒测试，我主要用的是黑盒测试的方式。

测试工具：chrome浏览器

测试方法：项目建立起来后，访问项目链接

访问地址：**localhost:3000/index.html**

再进行每一个路由测试的时候，项目都会完成一次上图的过程。如果返回的数据与设定的数据是一样的，则说明路由是正确的。

## 科技处官网性能测试

科技处官网虽说项目不大，但是也是需要对其进行性能测试的，为的是精益求精。我对科技处官网测试的时候使用了chrome浏览器的插件PageSpeed Insights。

PageSpeed Insights的是一个开源的C ++ SDK，即适用于Chrome和Firefox浏览器的权力扩展，以及PageSpeed Insights的在线版本[12]。网站管理员和开发人员可以使用PageSpeed Insights的评估他们网页的性能，并且能获得关于网站如何更好的优化的建议。

测试结果样例如图4-3：

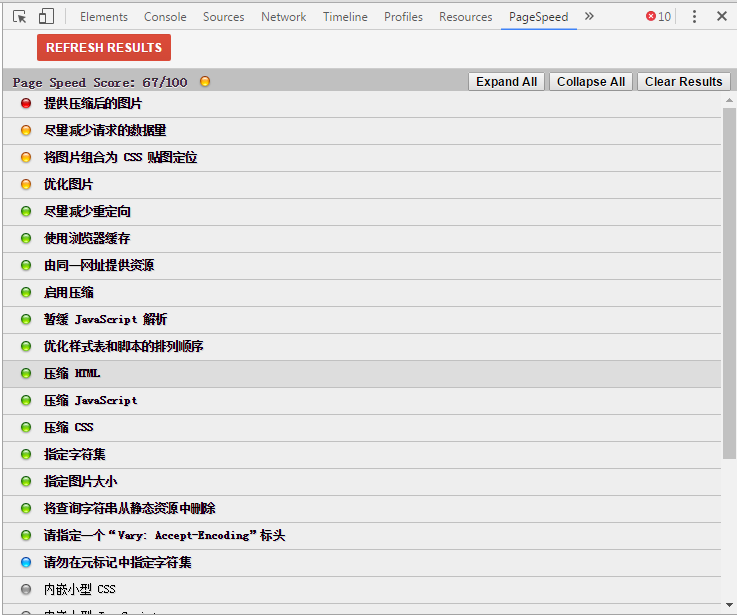


图4-3性能测试样图

在测试时我优化了这些项目：图片压缩，设定编码方式，合并多个css文件，提取公共文件，压缩HTML文件等选项

PageSpeed官网：https://developers.google.com/speed/pagespeed/

## 科技处官网开发过程所用到的调试方法

调试，在任何一项技术研发中都可谓是必不可少的技能。掌握各种调试技巧，必定能在工作中起到事半功倍的效果。譬如，快速定位问题、降低故障概率、帮助分析逻辑错误等等。讲解各种JS的调试技巧。

### Alert：测试之前的代码是否被执行了

Alert调试还是互联网刚刚起步，网页前端还主要以内容展示为主，浏览器脚本还只能为页面提供非常简单的辅助功能的时候。那个时候，网页主要运行在以IE6为主的浏览器中，JS的调试功能还非常弱，只能通过内置于Window对象中的alert方法来调试，那时候的代码是这个样子的：

alert（’ 科技处官网’）;

虽然，alert的调试方式非常原始，但是确实有它不可磨灭的价值。Alert调试的截图如图4-1



图4-2 alert调试效果图

### Console：执行语句的结果是什么

传统的alert调试方式在科技处官网开发过程中是很方便，但是比较JS已经越来越流行，所以它已经渐渐不能满足科技处官网开发的的种种场景。而且alert调试方式在调试科技处官网时弹出的调试信息，那个窗口很不美观，而且会遮挡官网的部分页面内容，感觉有些不太友好。

另一方面，alert的调试信息，必须在程序逻辑中添加类似”alert(xxxxx)”这样的语句，才能正常工作，并且alert会阻碍官网页面的继续渲染。这就意味着科技处官网调试完成后，必须手动清除这些调试代码，非常麻烦的。

所以，在调试科技处官网时，我还使用了另一种调试方式，代码是这个样子的：

console.log(‘科技处官网’);

它可以在控制台打印出调试信息，并且不直接影响页面显示。调试如图4-2所示

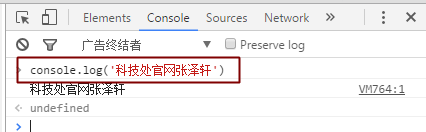


图4-3 console调试图

### JS断点调试：错误处理

在开发科技处官网服务器时，断点调试器是必不可少的，nodejs开发基本是以JS为主的。而开发中做调试是必不可少的环节。JS断点调试可以让程序在需要的地方中断，从而方便的分析逻辑，理清思路。也可以在某次调试中设置断点，到下一次时只需让程序自动运行到设置断点位置即可，这样便可在上次设置断点的位置中断下来，极大的方便了调试操作，也同时节省了时间。

JS断点调试，在整个科技处官网项目开发的过程中，有着异常的重要性。

# 互联网访问科技处官网

写的再好的程序如果不能应用到实际产品中，那说再多都是耍流氓。

## 公网映射

端口映射就是将外网主机的IP地址的一个端口映射到内网中一台机器，提供相应的服务。当用户访问该IP的这个端口时，服务器自动将请求映射到对应局域网内部的机器上。端口映射有动态和静态之分。

通俗来讲，端口映射是将一台主机的内网(LAN)IP地址映射成一个公网(WAN)IP地址，当用户访问提供映射端口主机的某个端口时，服务器将请求转移到本地局域网内部提供这种特定服务的主机；利用端口映射功能还可以将一台外网IP地址机器的多个端口映射到内网不同机器上的不同端口。 端口映射功能还可以完成一些特定代理功能，比如代理POP，SMTP，TELNET等协议。理论上可以提供65535(总端口数)-1024(保留端口数)=64511个端口的映射。

## 映射工具映射

本地服务器成功启动后，打开花生壳软件，配置相关参数，确认即可完成映射，我的配置参数如图5-1



图5-1 花生壳配置

## 互联网访问科技处官网

在任何有网的pc端都可以直接访问科技处官网的本地链接。

# 结论

这次毕业设计让我更加熟悉了从理论到实践的跨越。从当初的查阅图书，到现在的网站成功运行，这中间有很多值得回味的地方。

从科技处官方网站服务器的搭建开始，就要考虑需要理解哪些模块可能失效以及为什么，还得知道如何运用服务器的接口、数据提取、网络请求等知识，了解应用的科技处官方网站规模大小，访问量的预判定等。 接下来是建立适合科技处官网的数据库，如果开始的数据模型存在缺陷，那么就常常需要在业务逻辑和更高的层面上使用丑陋的代码来弥补那些数据模型没办法覆盖的死角。还得知道如何建立合理的范式关系模型，具备完善的外键、索引、视图、查询表等等。还要熟悉非关系型数据的存储，知道非关系型数据库在哪些方面超越关系型数据库。

再就是框架的事情，科技处官方网站依然选择了传统的MVC框架。这些关乎外部世界如何与业务逻辑及数据模型交互。会使用到大量的框架。想开发出好的科技处官方网站，必须得有能力写出一致、清晰、使用简单的接口。

还有就是视图界面，如何更好的显示数据，如何更好的去布局这些数据的变现形式。总之，实现一套好的视觉设计方案也是异常的关键。科技处官方网站的项目里还使用了一些HTML5/CSS。

比较重要的还有用户体验。在开发科技处官方网站时应该意识到：访问者官网者仅仅是希望一切运转良好。一个好的官网是不会让用户得‘腕管综合征’或产生眼疲劳的。科技处官网全栈开发的过程中需要审视全局，我还将一个需要3次点击+2个步骤的过程，简化为仅需要一次点击就可以完成。在开发过程中，还注意到了如何向用户提示错误信息。例如如果官网出现故障时，会真诚道歉。

这次的设计，从选题到详细设计都是一个循循渐进的过程。编码的实现几乎都是自己独立完成的。从前台网页设计的实现，到后台代码的编辑，我用到的软件主要有Webstorm、sublime、mongoChef等，并首次运用nodeJS语言，开发了科技处官方网站。在系统的开发过程中，遇到了好多不懂的知识，数据库原理、面向对象程序设计等，强烈地感觉到理论课程在实践中的重要性。

整个过程中，从需求分析到设计、编码、测试，我都力求规范化和文档化，努力让自己以前学的知识运用到本网站的开发中，尽量保证整个系统的开发进度和质量，顺利完成这次的毕业设计，为自己的大学生涯画上一个完美的句号。

不过，在系统开发过程中，好多知识都是随学随用，就增加了很多不必要的麻烦。虽说这些都会消耗系统开发的时间，但在自己的不断努力下，不怕麻烦，不怕重复，当克服了这些问题之后，我会感觉到自己的知识在一点一滴地积累，不知不觉中增加。

虽然科技处官网已经初步完成了，但是系统中难免还有很多不足之处需要好好改进。

我相信，只要肯钻研，只要挤时间，一切自己想要的知识都可以掌握。

附 录

附录1：express包的app.js：

var express = require('express');

var path = require('path');

var favicon = require('serve-favicon');

var logger = require('morgan');

var cookieParser = require('cookie-parser');

var bodyParser = require('body-parser');

var routes = require('./routes/index');

var users = require('./routes/users');

var guizhangzhidu = require('./routes/guizhangzhidu');

var indexInfo = require('./routes/indexInfo');

var jigoushezhi = require('./routes/jigoushezhi');

var kejichengguo = require('./routes/kejichengguo');

var keyanjidi = require('./routes/keyanjidi');

var xiangmushenbao = require('./routes/xiangmushenbao');

var xuebaobianji = require('./routes/xuebaobianji');

var xueshuhuiyi = require('./routes/xueshuhuiyi');

var yijianfankui = require('./routes/yijianfankui');

var app = express();

// view engine setup

app.set('views', path.join(\_\_dirname, 'views'));

app.set('view engine', 'ejs');

// uncomment after placing your favicon in /public

//app.use(favicon(\_\_dirname + '/public/favicon.ico'));

app.use(logger('dev'));

app.use(bodyParser.json());

app.use(bodyParser.urlencoded({extended: false}));

app.use(cookieParser());

app.use(express.static(path.join(\_\_dirname, 'public')));

app.use('/', routes);

app.use('/users', users);

app.use('/',guizhangzhidu);

app.use('/',indexInfo);

app.use('/',jigoushezhi);

app.use('/',kejichengguo);

app.use('/',keyanjidi);

app.use('/',xiangmushenbao);

app.use('/',xuebaobianji);

app.use('/',xueshuhuiyi);

app.use('/',yijianfankui);

// catch 404 and forward to error handler

app.use(function (req, res, next) {

var err = new Error('Not Found');

err.status = 404;

next(err);

});

// error handlers

// development error handler

// will print stacktrace

if (app.get('env') === 'development') {

app.use(function (err, req, res, next) {

res.status(err.status || 500);

res.render('error', {

message: err.message,

error: err

});

});

}

// production error handler

// no stacktraces leaked to user

app.use(function (err, req, res, next) {

res.status(err.status || 500);

res.render('error', {

message: err.message,

error: {}

});

});

module.exports = app;

附录2：路由文件代码：

var express = require('express');

var router = express.Router();

var mongoose = require("mongoose");

var db = mongoose.connect("mongodb://127.0.0.1:27017/kejichu");//链接到数据库test集合上；

var TestSchema = new mongoose.Schema({

div:{type:String},

name: {type: String}

});

/\* GET home page. \*/

/\*前台路由规则\*/

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

res.render('index', { title: '这是首页' });

});

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

res.render('index',{title:'这是首页'});

});

// 机构设置

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

//数据库操作

var Model = db.model("jigoushezhis", TestSchema);

Model.find({div:"left"},{\_id:0,div:0}, function (error, docs) {

if(error){

console.log("error :" + error);

}else{

res.render('jigoushezhi',{title:docs[0].name,title2:docs[1].name,title3:docs[2].name});

}

});

});

// 项目申报

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

res.render('xiangmushenbao',{title:"dkljsadf"});

});

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

res.render('kejichengguo',{title:"dkljsadf"});

});

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

res.render('keyanjidi',{title:'这是科研基地页'});

});

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

res.render('guizhangzhidu',{title:'这是规章制度页'});

});

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

res.render('xueshuhuiyi',{title:'这是学术会议页'});

});

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

res.render('xuebaobianji',{title:'这是学报编辑处页'});

});

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

res.render('xiazaizhongxin',{title:'这是下载中心页'});

});

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

res.render('yijianfankui',{title:'这是意见反馈页'});

});

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

res.render('indexInfo',{title:'这是意见反馈页'});

});

/\*以下是后台路由规则\*/

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

res.render('roots/index',{title:'首页'});

});

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

res.render('roots/login',{title:'登陆'});

});

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

});

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

res.render('roots/reg',{title:'注册'});

});

// test public-three-mod

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

res.render('public/public-three-modue',{title:'测试页面'});

});

module.exports = router;

附录三：CSS样式代码

\*{padding: 0;margin: 0;}

body { min-width: 1002px;width: 1002px; height: 100%;margin: 0 auto;background: #8dd8ef; }

.public-header-img {width: 1002px; }

.public-header-nav {width: 1002px; border-top: 1px solid #00a6da; }

.public-header-nav ul li {float: left;}.div3\_right\_div img {margin: 5px 10px}

.footer {width: 1002px;height: 100%; background: #f0f0f0; font-size: 12px; text-align: center; overflow: hidden;}

.footer p {margin: 20px auto;}

.boder{

width:100%;

background: #fff;

}

.aside-nav{

background:#fff;

list-style-type:none;

width: 206px;

border-left:1px solid #9DD2FD;

border-right:1px solid #e3fdff;

}

.aside-nav-warp{

margin-top:20px;

width: 209px;

float: left;

margin-left: 10px;

}

.aside-nav-warp::after{

content: '';

display: block;

clear: both;

}

.aside-nav-a{

text-decoration: none;

}

.aside-nav-li{

position:relative;

line-height: 40px;

height:40px;

border-bottom:1px solid #ccc;

margin:0 10px;

padding-left: 20px ;

}

.aside-nav-img{

width: 7px;

height:7px;

position: absolute;

left: 10px;

top: 19px;

}

.public-body{

width: 720px;

float: right;

margin-right:20px;

margin-top:20px;

border:1px solid grey;

}

.boder:after{

content:'';

display:block;

clear:both;

}

/\* 公共的主要内容 \*/

.nav-warp{

margin: 10px;

border:1px solid #cccccc;

}

.nav-warp>p{

height: 40px;

line-height: 40px;

}

.lump{

display: inline-block;

float: left;

width: 10px;

height:20px;

margin:10px;

background: blue;

}

.nav-name{

display: inline-block;

float: left;

font-size: 13px;

font-weight: bold;

}

.nav-address{

display: inline-block;

float:right;

font-size: 13px;

margin-right:10px;

}

.info{

margin: 20px;

}

/\*全局public样式\*/

.float-right{

float:right;

}

.public-ul{

font-size: 14px;

}

.public-li{

list-style-position: inside ;

border-bottom:1px dotted #cccccc ;

}

.public-a{

text-decoration: none;

height: 30px;

line-height: 30px;

}

参考文献

[1] 李园园. 面向DIV+CSS布局的Web页面分块算法研究[D]. 广西师范大学, 2013.

[2] 余果. Web全栈工程师的自我修养[M]. 人民邮电出版社, 2015.

[3] 马尔达若夫[美]. JavaScript快速全栈开发[M]. 人民邮电出版社, 2015.

[4] 朴灵．深入浅出nodeJS [M]. 人民出邮电出版社. 2013年;

[5] BYVoid．Node.js开发指南[M]. 人民出邮电出版社. 2012年;

[6] 红丸. MongoDB管理与开发精要[M]. 机械工业出版社, 2012.

[7] Nicholas C.Zakas．JavaScript高级程序设计[M]. 人民出邮电出版社. 2012年;

[8] (美)弗拉纳根．JavaScript权威指南[M]. 机械工业出版社. 2007年;

[9] 单东林，张晓菲，魏然．锋利的Jquery[M]. 人民出邮电出版社. 2009年;

[10] Kristina Chodorow. 深入学习MongoDB 人民出邮电出版社 2012年3月第一次印刷

[11] 苗雪兰 刘瑞新 王怀峰：《数据库系统原理及应用教程》，第一版：机械工业出版社；2001年7月

[12] 邵丽萍 张后扬 郭春芳：《网站编程技术实用教程》，第一版：清华大学出版社；2005年1月

致 谢

转眼之间，大学四年的时光就要结束了，回想起自己在大学的这四年时光，感觉就是一眨眼的功夫而已。在现在回想起刚上大学时军训的情景，还是历历在目，那时的我还是一个懵懂的小孩，还有太多不懂的地方。是上了大学以后，才开始了自己真正的人生，而在大学期间，我也是获得了许多人的帮助和鼓励。

首先，我想对我的辅导员刘刘老师说一句，谢谢您，入学开始认识的第一位老师就是刘老师，在我眼里，刘老师就像我的一个大姐姐一样，她经常会到我们的公寓楼来慰问我们，关怀我们，让我在第一离家这么远的地方感受到了家的感觉。

其次要感谢的人是我的班主任，朱老师，虽然他只带我们到大二结束。但是他却在我的人生道路上帮助我度过了那段艰难的日子，当时我还是班长，在管理和服务方面，特别欠缺。是朱老师不厌其烦的教导我，让我在我这方面有了很大的提高。让我接下来的时间，管理班级轻松了很多。

最后我要感谢在我大学四年里认识的，不认识的校友和老师们。是你们陪伴我度过了我最幸福的四年。我们在这里，一起生活，共同成长。我的生活因为有你们而更精彩！

**西安工程大学**

**本科毕业设计（论文）**

**诚信声明**

禀承学校优良传统学风，保持我校学生一贯诚信风尚，本人郑重声明：所呈交毕业设计（论文）是在指导老师的指导下独立完成的，无抄袭和剽窃现象。

特此声明。

学生签名：

指导教师签名：

日　　期：