

西安工程大学

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。

本篇论文主要讲解了对西安工程大学科技处官网开发从开始到结束的开发过程，以及从前端到服务器，数据库的选择，和作用的讲解。还分析了开发模式的优势。

又介绍了现在主流的前端框架在西安工程大学科技处时带来的便利，以及各个框架的介绍等。

还讲了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。

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.

This thesis mainly explained on the official website of the Xi'an Engineering University of science and technology development from the beginning to the end of the development process, and from the front end to the server, database, and explain. Also analyze the advantage of development mode.

Also introduced the mainstream of the front-end framework in the Department of science and technology at the Xi'an Polytechnic University at the convenience of the Department, as well as the introduction of the framework.

Also spoke for the Xi'an Polytechnic University science and technology development process of debugging skills and debugging methods.

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

目录

第 1 章 绪 论.....	1
1.1 课题背景.....	1
1.1.1 科技处官网介绍.....	1
1.1.2 科技处官网也应该跟上时代.....	2
1.2 课题研究的目的和意义.....	3
1.2.1 浏览器市场份额报告.....	3
1.3 西安工程大学科技处网站全栈开发架构设计概述.....	4
1.3.1 西安工程大学科技处官方网站前端技术架构.....	4
1.3.2 西安工程大学科技处官方网站后端技术选择.....	5
1.3.3 数据库选择.....	5
第 2 章 前端开发.....	7
2.1 科技处官网前端开发概述.....	7
2.1.1 页面标记（HTML）.....	7
2.1.2 页面样式.....	7
2.1.3 前端编程.....	8
2.1.4 跨平台，跨浏览器.....	8
2.1.5 前端框架.....	8
2.1.6 调试工具.....	9
2.2 前端开发总体设计.....	9
2.3 前端开发详细设计.....	10
2.3.1 jquery 框架.....	10
2.3.2 bootstrap 类库.....	10
第 3 章 服务器开发.....	11
3.1 科技处官网服务器开发总体设计.....	11
3.1.1 基于 V8 引擎的服务器语言 nodeJs 的介绍.....	12
3.1.2 服务器的 npm 包管理器.....	12

3.2 科技处官网服务器搭建详细过程.....	13
3.2.1 下载.....	13
3.2.2 安装.....	13
3.2.3 配置.....	15
3.2.4 配置科技处官网的 Npm 包.....	17
3.2.5 科技处官网服务器的模板选择.....	18
第 4 章 系统测试与分析.....	19
4.1 科技处官网路由测试.....	19
4.2 科技处官网性能测试.....	19
4.3 科技处官网开发的调试方法.....	19
4.3.1 骨灰级调试大师 Alert	20
4.3.2 新一代调试王者 Console	20
4.3.3 JS 断点调试.....	21
第 5 章 结论.....	23
参考文献.....	25
附 录.....	26
致 谢.....	38

第1章 绪 论

1.1 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，官方网站却是很多年前的版面。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，我就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。

因为一个网站的开发涉及到的东西很多，所以就产生了一个全栈工程师。专门做全栈开发。

全栈开发，又名全端开发，英文是 Full Stack develop。指由掌握多种技能，并能利用多种技能独立完成产品的人开发出来的 web 网站。

1.1.1 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，但是后期维护不到位，到现在部分页面有错版，平台还有很多 BUG(错误)。如图 1-1 所示，首页的通知栏布局错误。



图 1-1 科技处官方网站首页截图

部分页面的导航布局错误，如图 1-2 所示。



图 1-2 科技处官方网站导航排版错误截图

1.1.2 科技处官网也应该跟上时代

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^[12]。

到目前为止，绝大多数的网站都是采用了 DIV+CSS 布局来排版的。

1.2 课题研究的目的和意义

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；
- (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；

1.2.1 浏览器市场份额报告

一个网站的入口，有 80% 都是由浏览器决定的，而西安工程大学科技处官方网站也不例外，他主要面相的对象是 PC 端客户。而西安工程大学科技处官方网站所选择的架构的兼容性大多集中到了 ie6 上。

我们从百度统计的数据可以看出，从 2016 年 1 月到 2016 年 5 月，ie6.0 浏览器在市场上的占比已经越来越低了，浏览器份额报告如下图 1-3 所示：

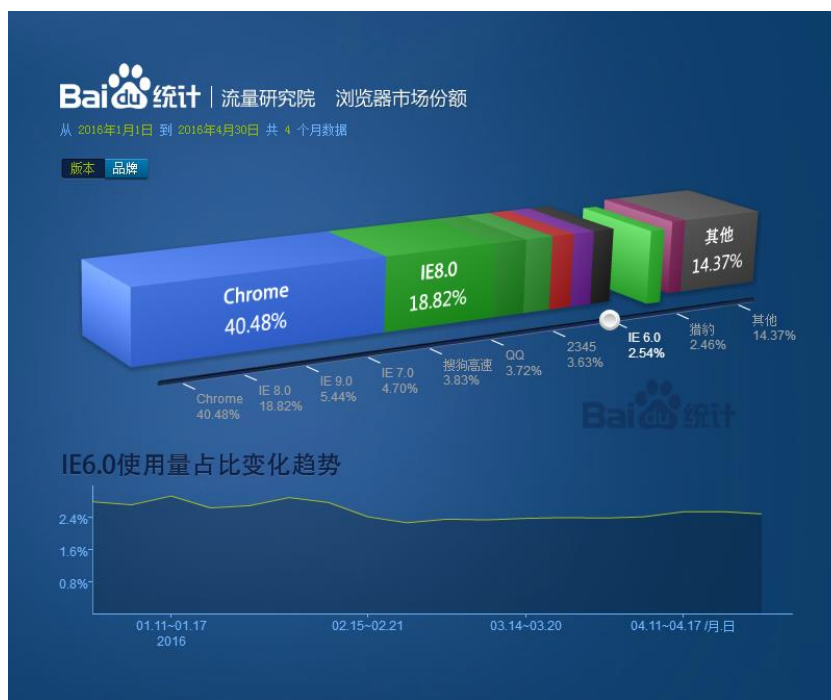


图 1-3 浏览器市场份额报告 2016.01-2016.04

然而西安工程大学科技处官方网站还是有许多代码是为了兼容 ie6.0 而设计的，这就造成了大量冗余代码，浪费了电脑性能，也让访问者支出了不必要的内存损耗。

1.3 西安工程大学科技处网站全栈开发架构设计概述

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^[9]。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^[8]。西安工程大学科技处官方网站的全栈开发也不例外，它必须有自己的服务器，有自己的前端页面，有自己的数据库。

1.3.1 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^[11]。

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，有这么一些事情可以做：

- 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東西，因为这是不同的协作环节。

1.3.2 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。

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^[1]。

1.3.3 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^[6]，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]。

第2章 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 前端开发技术也会有所不同

2.1 科技处官网前端开发概述

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 工具的使用等方法，以便于设计出更好的 UI 界面。尽管 Web 开发的范畴广泛，并且界限模糊，但是以下 7 点则是科技处官网前端开发必备的内容。

2.1.1 页面标记（HTML）

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，所以从学习的难易程度来说，HTML 应该是科技处官网前端开发中非常容易上手的技术。即使是一个新手也能在较短的时间里学会并编写出科技处的页面来。虽说入门容易，但是要编写语义良好、简洁整齐的 HTML 代码则需要大量的实践学习才能掌握。HTML 是科技处官网页面的基本结构组成，是网站的基础，臃肿混乱的 HTML 代码不但会影响科技处官网本身的展现，而且与它对应的 CSS 和 JavaScript 代码也会变得难以编写和维护。

2.1.2 页面样式

CSS 是 Cascading Style Sheets(层叠样式表)的简称。在标准页面设计中，CSS 负责网页内容的表现，所以 CSS 也是科技处官网前端开发需要掌握的核心内容^[13]。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。以上这两项基本技能是科技处

官网 UI 开发的必备技能。

2.1.3 前端编程

科技处官网的前端编程技能主要指的是对其 JavaScript 的编程。JavaScript 是一种居于对象和事件驱动的客户端脚本语言，是页面实时动态交互的技术基础。相较于 HTML 和 CSS，编写科技处官网的 JavaScript 代码更能找到全栈开发程序员的感觉。JavaScript 是非常灵活的脚本语言，它包含了高阶函数、动态类型和灵活的对象模型这些强大的语言特性，当然，JavaScript 的灵活性也导致代码不容易维护。除此外，浏览器的兼容性也增加了 JavaScript 编码的难度。科技处官网中的某一个功能，可能在不同的浏览器中就有不同的实现，例如，在科技处官网开发中，需要给其绑定一个事件，IE 浏览器中绑定事件使用的是 `attachEvent()` 方法，但在其它浏览器则使用的是 `addEventListener()` 这个方法。在熟悉 JavaScript 基本语法和基本的编码规范之外，开发科技处官网还应该掌握在不同浏览器中 JavaScript 的兼容性问题。

2.1.4 跨平台，跨浏览器

科技处官网的前端代码本来是不存在跨平台方面的问题，但是随着移动 Web 平台的兴起，跨平台的问题就逐渐显现出来了。移动设备如智能手机和平板电脑在近几年发展迅猛，用户通过移动设备访问 Web 站点的比率也是逐年增高。如何将科技处官网也展现在众多移动平台、众多屏幕尺寸上，并且还能展现友好，这也是一项前端技能。这就要提到下一点要说的，科技处官网所选择的前端框架。目前流行的前端框架已经很好地处理了浏览器的兼容问题。

2.1.5 前端框架

各种前端框架的出现，很大程度上降低了前端开发的难度。框架统一了编码的方式、封装了浏览器兼容问题并添加大量的扩展功能。如今的 Web 项目中前端框架应用非常广泛，在开源社区 Github 上排名靠前的开源框架也是以前端框架居多^[3]。科技处官网选择一款优秀的前端框架可以很大程度上减少对它开发的周期，我这里选择了 JQuery 作为科技处官网的前端框架，因为 JQuery 几乎成为了 Web 项目默认的前端框架。

2.1.6 调试工具

对于科技处官网前端代码，在调试过程中需要查看页面的它的 HTML 结构变化，CSS 渲染效果，JavaScript 代码的执行情况及 HTTP 请求和返回的数据，并且还要查看整个科技处官方网站各个部分的性能等，甚至需要动态更改 HTML，CSS 代码来查看预期的效果，模拟发起 HTTP 请求来查看后端返回的数据。在调试科技处官网时，各主流浏览器都会有对应的浏览器插件来辅助完成这些工作，如：IE 中的 IE Dev Toolbar、Chrome 中的 Developer Tools、Firefox 中的 Firebug 等，此外还有 HTTP 请求监控和模拟工具，如：Fiddler 等。

2.2 前端开发总体设计

前端开发还是使用了标准的 MVC 模式。

MVC 模式（Model - view - controller）是软件工程中的一种软件架构模式，把软件系统分为三个基本部分：模型（Model）、视图（View）和控制器（Controller）^[13]。

科技处官网也采用了这种模式，实现一种动态的程序设计，使对程序的后续修改和扩展简化，并且使程序某一部分的重复利用成为了可能。除此之外，此模式通过对复杂度的简化，使程序结构更加直观。

MVC 分工如下：

- （控制器 Controller）- 负责转发请求，对请求进行处理^[8]。
- （视图 View）- 界面设计人员进行图形界面设计^[8]。
- （模型 Model）- 程序员编写程序应有的功能（实现算法等等）、数据库专家进行数据管理和数据库设计(可以实现具体的功能)^[8]。

MVC 框图如图 2-1：

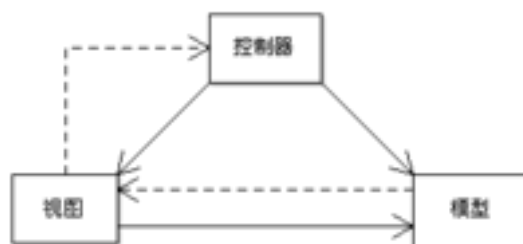


图 2-1 MVC 框图

2.3 前端开发详细设计

下面我们详细介绍一下科技处官网中使用到的各个框架和类库。

2.3.1 jquery 框架

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。那什么是 jQuery 呢？它是一套跨浏览器的 JavaScript 库，简化 HTML 与 JavaScript 之间的操作^[4]。由约翰·雷西格（John Resig）在 2006 年 1 月的 BarCamp NYC 上发布第一个版本^[4]。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^[4]。

2.3.2 bootstrap 类库

科技处官网的登陆注册页面使用到了 bootstrap 类库。整个后台都是基于 bootstrap 开发的。那什么是 Bootstrap 呢？它是一组用于网站和网络应用程序开发的开源的前端优秀框架，包含了 HTML、CSS 及 JavaScript 的框架，它提供字体排印、字体图标、窗体、按钮、导航及其他类的各种组件还包涵 JavaScript 扩展。

截止到 2016 年 5 月时，bootstrap 是 GitHub 上面被标记为“Starred”次数排名第二的项目。Starred 次数超过 96,500，而分支次数超过了 42,200 次，被标记数如下图 2-1

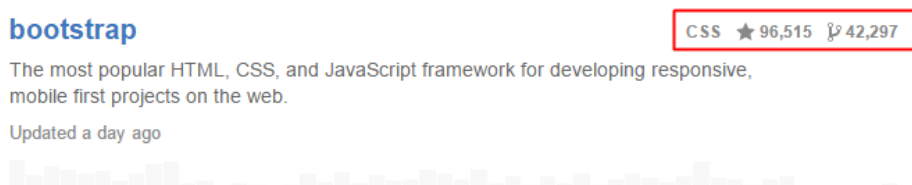


图 2-2 GitHub 上面被标记为“Starred”次数

第3章 服务器开发

网页服务器（Web server）这个词它有两个意思：

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^[10]

3.1 科技处官网服务器开发总体设计

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，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，所以就没使用这种方式。

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。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。

3.1.1 基于 V8 引擎的服务器语言 nodeJs 的介绍

看到 Node.js 这个名字，很多初学者都会误以为这是一个 Javascript 的应用，而事实上，Node.js 采用了 C++ 语言编写而成，是一个 Javascript 运行环境。为什么采用 C++ 语言呢？据 Node.js 创始人 Ryan Dahl 回忆，他最初希望采用 Ruby 来写 Node.js，但是后来发现 Ruby 虚拟机的性能不能满足他的要求，后来他尝试采用 V8 引擎，所以选择了 C++ 语言。既然不是 Javascript 应用，为何叫 .js 呢？因为 Node.js 是一个 Javascript 的运行环境。提到 Javascript，大家首先想到的是日常使用的浏览器，现代浏览器包含了各种组件，包括渲染引擎、Javascript 引擎等，其中 Javascript 引擎负责解释执行网页中的 Javascript 代码。作为 Web 前端最重要的语言之一，Javascript 一直是前端工程师的专利。不过，Node.js 是一个后端的 Javascript 运行环境（支持的系统包括 *nix、Windows），这意味着你可以编写系统级或者服务器端的 Javascript 代码，交给 Node.js 来解释执行，简单的命令类似于：

```
node helloworld.js
```

Node.js 采用了 Google Chrome 浏览器的 V8 引擎，性能很好，同时还提供了很多系统级的 API，如文件操作、网络编程等。浏览器端的 Javascript 代码在运行时会受到各种安全性的限制，对客户系统的操作有限。相比之下，Node.js 则是一个全面的后台运行时，为 Javascript 提供了其他语言能够实现的许多功能。

3.1.2 服务器的 npm 包管理器

NPM（node package manager），通常称为 node 包管理器^[5]。顾名思义，它的主要功能就是管理 node 包，包括：安装、卸载、更新、查看、搜索、发布等^[5]。

npm 的背后，是基于 couchdb 的一个数据库，详细记录了每个包的信息，包括作者、版本、依赖、授权信息等^[1]。它的一个很重要的作用就是：将开发者从繁琐的包管理工作（版本、依赖等）中解放出来，更加专注于功能的开发^[5]。

npm 官网：<https://npmjs.org/>

npm 官方文档：<https://npmjs.org/doc/README.html>

3.2 科技处官网服务器搭建详细过程

理论知识再充沛，也是为实际操作服务的。接下来我们详细说下科技处官网服务器开发的那些事儿。

3.2.1 下载

到官网里下载镜像安装包

下载地址：<https://nodejs.org/en/>

3.2.2 安装

安装步骤如下：

图 3-1:打开镜像安装包,进入 nodeJs 安装欢迎界面。

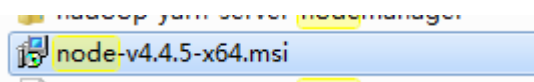


图 3-1 打开镜像安装包

图 3-2:点击 next，进入 nodeJS 安装条款界面

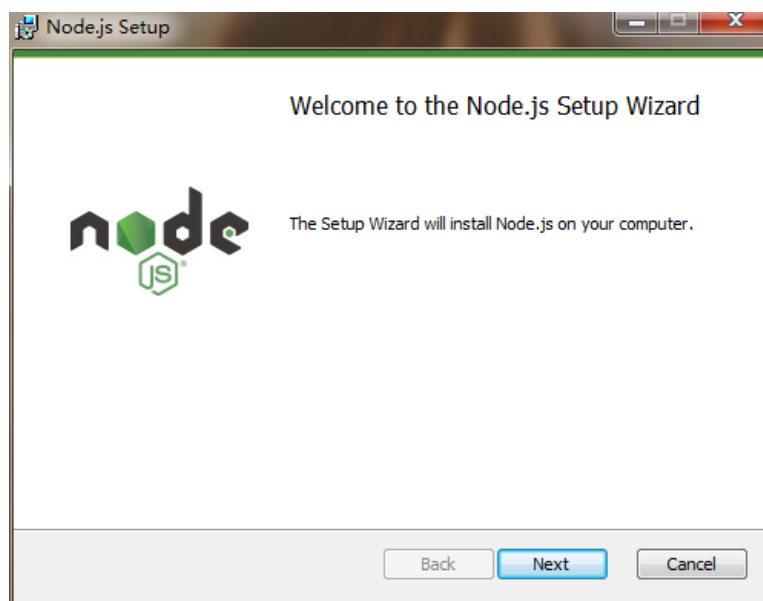


图 3-2 Nodejs 安装欢迎界面

图 3-3:同意 nodeJS 条款条例。点击 next，进入选择安装目录

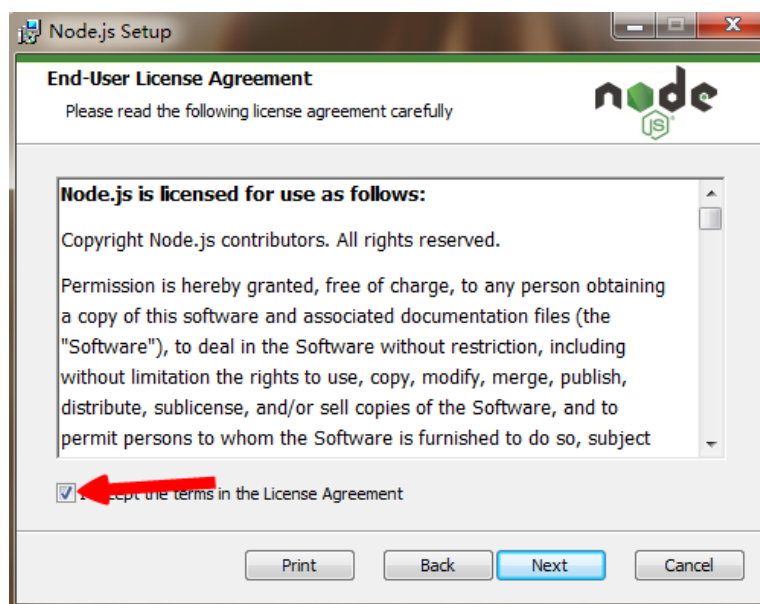


图 3-3 nodeJS 条款条例图

图 3-4:选择安装目录，点击 next，进入 node 选择安装功能页

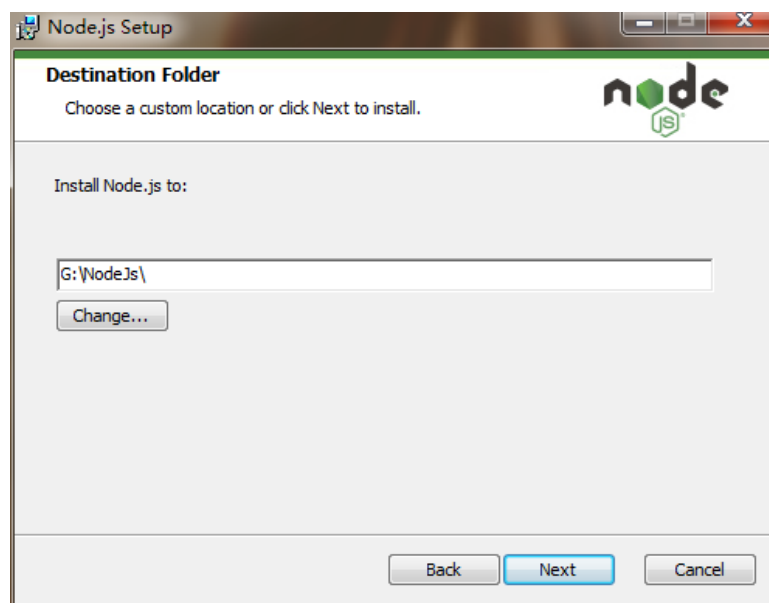


图 3-4node 择安装目录

图 3-5: 选择安装功能页，点击 next 进入安装页

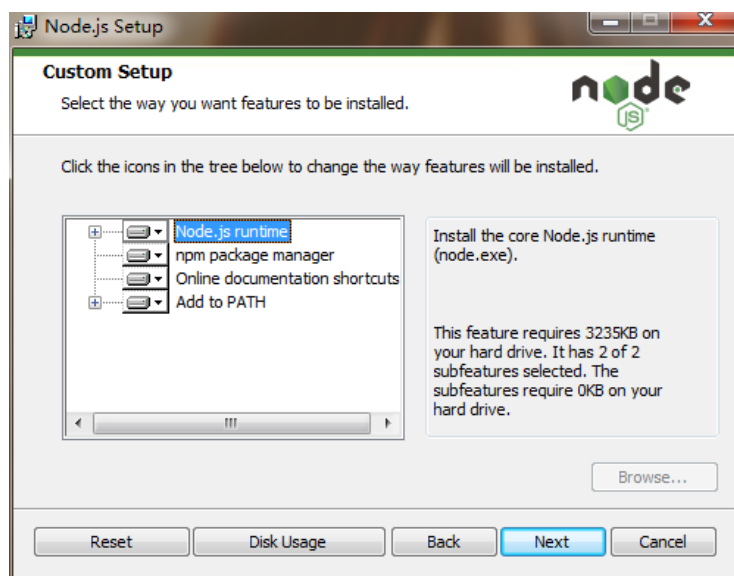


图 3-5 node 选择安装功能

图 3-6: 安装页，点击 instal 进入安装

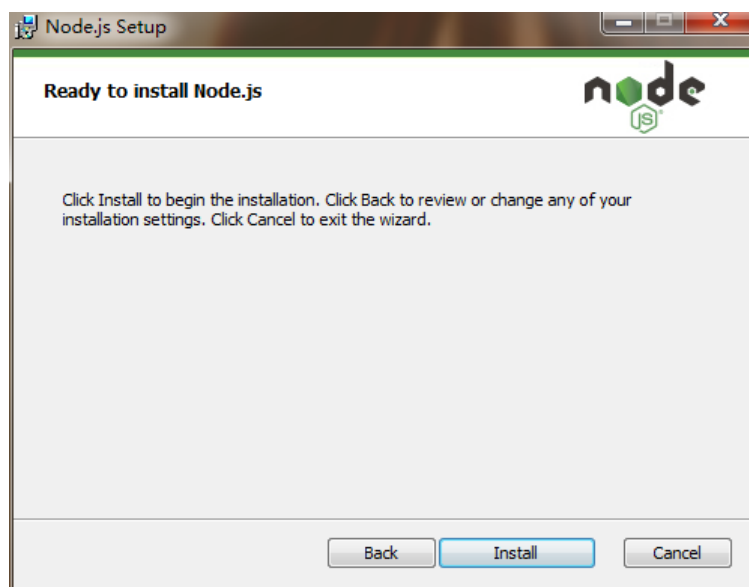


图 3-6 安装页

等待安装完成！

到这里，nodejs 就已经安装成功了。

3.2.3 配置

添加 nodejs 的环境变量，我把 node 安装在了 G 盘，我就以 G 盘为例。

在桌面上的 我的电脑 点击右键 -> 属性 -> 高级系统设置(如图:3-7)



图 3-7 高级系统设置

→ 点击环境变量，如图 3-8

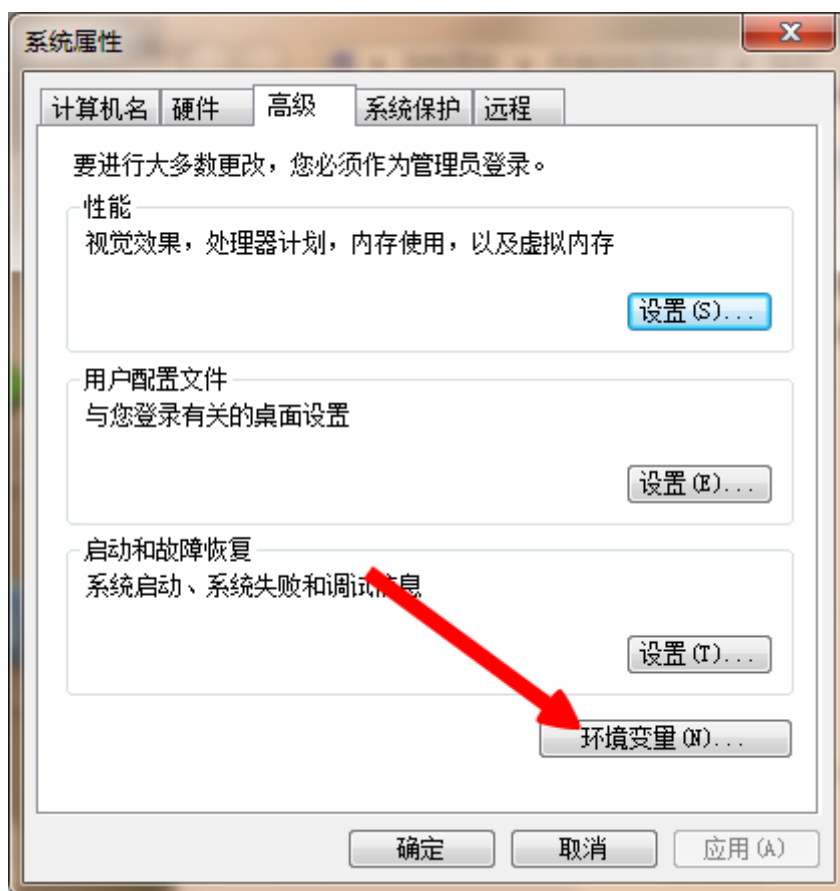


图 3-8 环境变量

找到 Path，点击编辑，如图 3-9

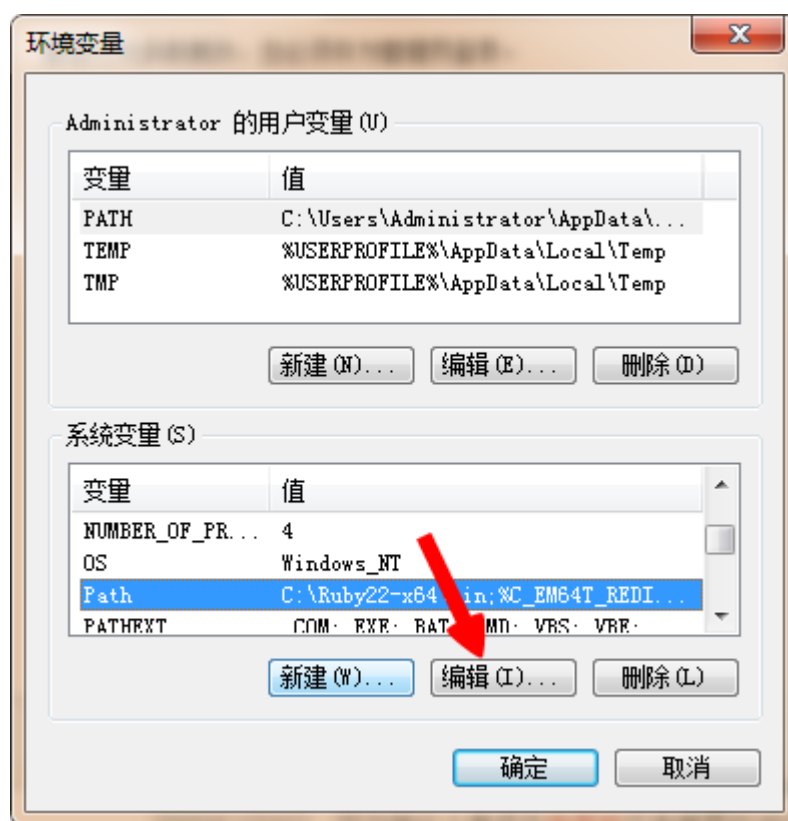


图 3-9 环境变量

在变量值一栏最后添加 ;G:\NodeJs\bin\ 如图 3-10



图 3-10 添加环境变量

添加完成后点击确定，到这里，我们的 nodeJS 就配置完了。

3.2.4 配置科技处官网的 Npm 包

Npm 包是 nodeJS 开发必不可少的一部分，科技处官网也是基于一个叫 Express 的 npm 包搭建的服务器。

先安装全局的 express 包

打开 nodejs，进入到项目目录下，输入：

```
install express -g
```

回车后，等待安装完成。

接下来创建 **node** 服务器，继续在 **nodeJS** 里输入

```
express name -ejs  
cd name && npm install  
npm start // 启动服务器
```

到这里我们的科技处官网的服务器就启动起来了

3.2.5 科技处官网服务器的模板选择

传统模板技术其实是通用模板，即模板引擎并不 care 你输出的是 html 还是其他格式的文本。科技处官网的构架是相同的模板计较多，所以我们选择了 **ejs** 模板，因为 **EJS** 专为 **HTML** 设计的，并且它可以做许多传统模板做不到的专门针对 **html** 的优化。其次，**EJS** 是用 **JSON** 数据来生成 **HTML** 的字符串的。

第4章 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。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

4.1 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^[8]，我主要用的是黑盒测试，用 `chrome` 浏览器打开项目平台，去点击科技处前端页面上的每一个连接，来看科技处官网上的所有链接是否都跳转到了它该跳转的地方，整个网站是不是都可以互通。

4.2 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^[14]。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^[15]。

在测试是我优化了这些项目：图片压缩，设定编码方式，合并多个 `css` 文件，提取公共文件。

`PageSpeed` 官网：<https://developers.google.com/speed/pagespeed/>

4.3 科技处官网开发的调试方法

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。

4.3.1 骨灰级调试大师 Alert

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^[8]。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，我在开发科技处官网时调试，大概有 30%是使用了这种方式。所以直到今天，alert 依然有其它的武之地。Alert 调试的截图如图 4-1



图 4-1 alert 调试效果图

4.3.2 新一代调试王者 Console

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^[8]。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

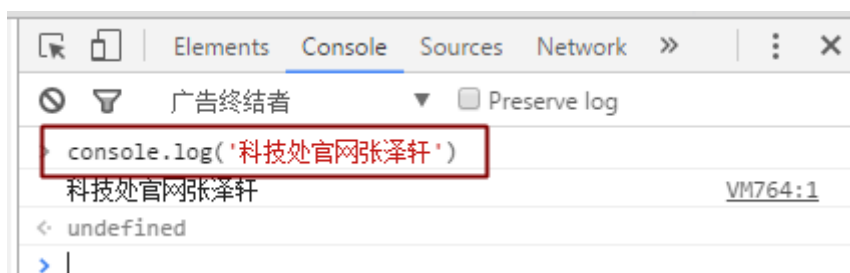


图 4-2 console 调试图

4.3.3 JS 断点调试

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^[8]。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。

第5章 结论

要一个人开发出科技处官方网站，至少要使用过多种编程语言，并且在各种领域的不同平台工作多年。开发出科技处官方网站全部内容，除了像“高级工程师”那样要掌握多门编程语言，还必须具有更高的视野将所有的模块联系起来。

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，必须得有能写出一致、清晰、使用简单的接口。

还有就是视图界面，科技处官方网站的开发需要知道：a) 懂得如何构建易读的布局；b) 能意识到自己设计的界面和图形的作用。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。不过从目前的趋势来看，JavaScript 将会是明日之星，最近这一领域硕果累累(NodeJs, backbone, knockout, Angular)^[2]。因此，次项目使用 JavaScript 作为主要语言开发也算是一种正确的选择。

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。因为有时一些无意的错误信息，会让用户感到莫名奇妙。

最后就是理解用户和市场需求，这层关系到系统架构，但也绝不是一个不能碰的角色。科技处官方网站开发时，还需要领悟客户使用浏览器访问的过程中会需要些什么。

参考文献

- [1] 朴灵. 深入浅出 nodeJS [M]. 人民邮电出版社. 2013 年;
- [2] Nicholas C.Zakas. JavaScript 高级程序设计[M]. 人民邮电出版社. 2012 年;
- [3] (美)弗拉纳根. JavaScript 权威指南[M]. 机械工业出版社. 2007 年;
- [4] 单东林, 张晓菲, 魏然. 锋利的 JQuery[M]. 人民邮电出版社. 2009 年;
- [5] BYVoid. Node.js 开发指南[M]. 人民邮电出版社. 2012 年;
- [6] Kristina Chodorow. 深入学习 MongoDB 人民邮电出版社 2012 年 3 月第一次印刷
- [7] 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
- [8] 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
- [9] 刘小东 陈艳玲: 《Dreamweaver MX 2004 与 ASP 动态网页设计实训教程》, 第一版: 上海科学普及出版社; 2005 年 7 月
- [10] 张伍荣 WindowsServer2003 服务器架设与管理 清华大学出版社; 2008 年 8 月
- [11] 吴涛: 《网站全程设计技术(修订本)》, 北京: 清华大学出版社; 北京交通大学出版社
- [12] 任学文、范严编: 《网页设计与制作》, 中国科学技术出版社
- [13] 杜巧玲等编: 《网页设计超级梦幻组合》, 清华大学出版社
- [14] 吴黎兵、罗云芳编: 《网页设计教程》, 武汉大学出版社
- [15] 庄王健编: 《网页设计三剑客白金教程》, 电子工业出版社

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;
```

}

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的。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。我的生活因为有了你们而更精彩！

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：