

**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

技术在过去的几十年里进步很快，也将在未来的几十年里发展得更快。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。

同时，由于公司组织结构的变迁，以及到变化的适应度，也决定了赋予每个人的职责将会越来越多。尽管我们看到工厂化生产带来的优势，但是我们也看到了精益思想带来的变革。正是这种变革让越来越多的专家走向全栈，让组织内部有更好的交流。

你还将看到专家和全栈的两种不同的学习模式，以及全栈工程师的未来。

从开始的 CGI 到 MVC 模式，再到前后端分离的架构模式，都在不断地降低技术的门槛。而这些门槛的降低，已经足以让一两个人来完成大部分的工作了

软件开发在过去的几十年里都是大公司的专利，小公司根本没有足够的能力去做这样的事。在计算机发明后的几十年里，开发软件是大公司才能做得起的。一般的非技术公司无法定制自己的软件系统，只能去购买现有的软件。而随着技术成本的下降，到了今天一般的小公司也可以雇佣一两个人来做同样的事

关键词：全栈工程师，软件开发，组织结构

**ABSTRACT**

Application of web technology in the past few decades progress soon, will in future decades developed faster. The threshold of technology today fell more and more quickly, the original need a team to do it, now only need to one or two people can be.

At the same time, due to the changes of the organizational structure of the company, and to change the fitness, also decided to give the responsibility of each person will be more and more. Although we see factory production advantages, but we also see the lean thinking brought by the transformation. Is this change makes more and more experts to full stack, within the organization have better exchanges.

You will also see the experts and the whole stack of two different learning models, as well as the future of the entire stack of engineers.

From the beginning of the CGI to MVC mode, and then to the front and back end of the separation of the architecture model, are constantly lowering the threshold of the technology, and the lower the threshold, which is enough to allow one or two people to complete most of the work.

Software development in the past decades are patent of large companies, small companies there is no enough ability to do such a thing. In the decades after the invention of the computer, the software development is big companies can afford to do. The general non technology companies to customize their own software system, only to buy existing software. And with the decline in the cost of technology to today generally small companies can also employ one or two people to do the same thing.

**KEYWORDS:** Full Stack Developer, software engineering, organization structure

目录

[摘 要 I](#_Toc452360889)

[**ABSTRACT** II](#_Toc452360890)

[第1章 绪 论 1](#_Toc452360891)

[1.1 课题背景 1](#_Toc452360892)

[1.1.1 科技处官网介绍 1](#_Toc452360893)

[1.1.2 前端风云变幻 2](#_Toc452360894)

[1.2 课题研究的目的和意义 4](#_Toc452360895)

[1.2.1 浏览器市场份额报告 4](#_Toc452360896)

[1.3 全栈开发架构设计概述 4](#_Toc452360897)

[1.3.1 前端技术架构 5](#_Toc452360898)

[1.3.2 后端技术选择 6](#_Toc452360899)

[1.3.3 数据库技术 6](#_Toc452360900)

[第2章 前端开发 7](#_Toc452360901)

[2.1 前端开发概述 7](#_Toc452360902)

[2.1.1 页面标记（HTML） 7](#_Toc452360903)

[2.1.2 页面样式 7](#_Toc452360904)

[2.1.3 前端编程 8](#_Toc452360905)

[2.1.4 跨平台，跨浏览器 8](#_Toc452360906)

[2.1.5 前端框架 8](#_Toc452360907)

[2.1.6 调试工具 9](#_Toc452360908)

[2.2 前端开发总体设计 9](#_Toc452360909)

[2.2.1 MVC模式 9](#_Toc452360910)

[2.2.2 ejs模板 10](#_Toc452360911)

[2.3 前端开发详细设计 10](#_Toc452360912)

[2.3.1 jquery框架 10](#_Toc452360913)

[2.3.2 bootstrap类库 10](#_Toc452360914)

[2.3.3 模板选择 11](#_Toc452360915)

[第3章 服务器开发 12](#_Toc452360916)

[3.1 服务器概述 12](#_Toc452360917)

[3.2 服务器开发总体设计 12](#_Toc452360918)

[3.2.1 基于V8引擎的服务器语言nodeJs介绍 12](#_Toc452360919)

[3.2.2 npm包管理器 13](#_Toc452360920)

[3.3 服务器搭建详细介绍 13](#_Toc452360921)

[3.3.1 下载 14](#_Toc452360922)

[3.3.2 安装 14](#_Toc452360923)

[3.3.3 配置 16](#_Toc452360924)

[3.3.4 Npm包 18](#_Toc452360925)

[第4章 系统测试与分析 20](#_Toc452360926)

[4.1 测试目的 20](#_Toc452360927)

[4.2 路由测试 20](#_Toc452360928)

[4.3 性能测试 20](#_Toc452360929)

[4.4 调试与分析 20](#_Toc452360930)

[4.4.1 骨灰级调试大师Alert 21](#_Toc452360931)

[4.4.2 新一代调试王者Console 21](#_Toc452360932)

[4.4.3 JS断点调试 21](#_Toc452360933)

[4.4.4 DOM断点调试 22](#_Toc452360934)

[4.4.5 XHR Breakpoints 22](#_Toc452360935)

[4.4.6 Event Listener Breakpoints 22](#_Toc452360936)

[第5章 结论 23](#_Toc452360937)

[参考文献 25](#_Toc452360938)

[附 录 26](#_Toc452360939)

[致 谢 30](#_Toc452360940)

# 第1章 绪 论

## 课题背景

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

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，西安工程大学有教育部重点实验室1个，省级重点实验室1个，省级2011协同创新中心1个，省级工程技术研究中心5个，省级哲学社会科学研究基地1个，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。

### 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

### 前端风云变幻

互联网十多年风云变幻、日新月异，新的东西层出不穷。

但看到现在仍有新人还要为IE低版本而费尽苦心去学习、钻研，让人感到心寒，同时也显示出一些前端教学人和一些前端先行者的可耻，这个五彩缤纷的年代，低版本兼容方法已经没有那么的重要了，没必要为了显得自己是多么的“博学”与“高深”而去煞费苦心的学习一下回报很低的东西。

感慨之余，回想起十多年前的互联网。

那时，还不懂IE还有版本之分的说法，只知道这个e图标(IE快捷方式)是用来上网冲浪的。而所谓的上网冲浪，就是打开雅虎中国，百度，而当不小心打开雅虎时（英文），还傻傻分不清为什么有时候打开的是汉语的，而有时候打开的是英文的。（在多年后才知道yahoo.com.cn与yahoo.com的区别）

那时，网易的域名还不是163.com，而是netease.com，那时英文不好，老记不住这域名，老是打成neteasy.com而打不开网页，一直在苦恼，为什么不是easy，而是ease呢？！那时所谓的注册电子邮箱，就是注册网易的邮箱，傻傻地认为，每一个步骤都得按教程操作。点击哪，出现什么效果，然后再到哪一步。不敢乱动，以免“走丢”了找不回页面，或操作不当造成注册失败。

然后，网页慢慢地浏览多了，就在想，这是怎么做到的呢？有了疑惑就跑到书店查书，那时关于web的书就没发现过。直到某一天，在学校机房上网（那时候没啥人上，电话拔号，网速慢，网费贵），也不知道哪个班来上网络课，也不清场（现在也不明白那时候怎么有网络课了呢），其中那位老师就用记事本展示了一个最基础的网页，当时豁然开朗，感觉是不是老天特意安排的！再后来，想要一个记事本来练习一下，而当时的网吧全都是美萍啊、万象啊等等的管理软件，电脑的东西全被隐藏住了，也就桌面上有些可用的图标。又到书店查书，去破解网吧软件的限制，后来发现查看源码可以调出记事本，从中也知道了怎么查看网页的源码，从而开始了挖别人源码的生涯。

那时候可菜了，菜得都不好意思说了，看到if语句，也不知道是用来干嘛的，为什么要判断，有什么意义嘛？其他的语法更不明白了，也没本书来参考。看到人家页面上那些漂亮的效果怎么办？那时候就是背，今天没背出来，明天接着背，也不理解，就是背、背、背，默写、默写、默写。。。背多了、写多了，慢慢地悟出点门道了。就开始在网上申请免费域名和免费空间，经常跑网吧去做自己的静态网站，说是网站，其实也就只有一两页。后来，在地摊上发现一本叫《网页动态效果大全》？（类似的名字），书有点厚，好像是20元（当时还是有点小贵的），就买了下来，里面都是些常用的效果，javascript中还夹带着asp代码，各种大杂烩，也没说asp怎么配置，就那样凑合着看呗。

这样磨蹭了一两年，直接2008年大年初一，大过年的，大伙都没啥事，都在逛街，我就跑到书店里去转转，发现了一个《动态网页制作》，32元，这里的动态，其实是javascript展示的动态，而不是有后台的动态网站，到此才刚刚踏入web制作的大门。

转眼间，风云际会，互联网迅速发展，各种新颖的东西层出不穷。取舍取舍，要取就要舍，当为年轻的一代，为什么还抱着过时的、低版本的IE，这样不舍得，怎么迎取新技术的入场？！可知道，netease已经变为了163，tencent已经成了QQ，yahoo.com.cn已经不再是当初的那个雅虎中国！为什么不与时俱进，科技处的网站也应当如此。用现如今的jQuery，bootstrap开发，速度也会很快，再加上nodeJS的出现，

## 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，已及技术选型所必要的基础理论；

### 浏览器市场份额报告

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，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，浏览器份额报告如下图1-3所示：

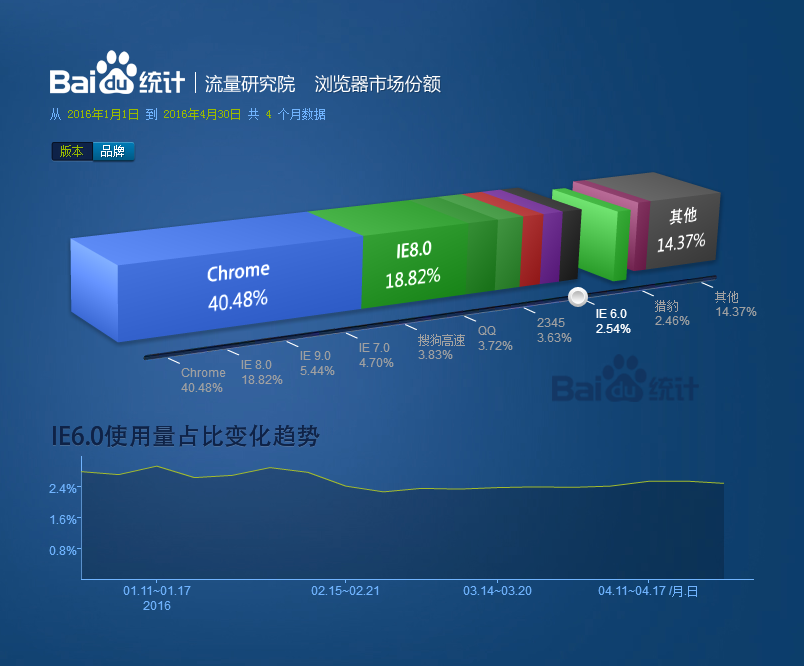


图1-3 浏览器市场份额报告 2016.01-2016.04

## 全栈开发架构设计概述

时至今日，也没有一个全栈工程师的标准，一千个开发者心中有一千个全栈的定义。有的视全栈工程师为什么都懂，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。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。

### 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。

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。

常见的前端产品形态包括：

* 内容型Web站点 侧重渲染方面的优化，前端逻辑比重小
* 操作型B/S系统 以数据和逻辑为中心，界面较规整
* 内嵌Web的本地应用 要处理缓存和一些本地接口，包括PC客户端和移动端

全栈的意思是，除了只做在浏览器中运行的代码，还写一些服务端的代码，这个需求又是从哪里来的呢？

这个需求其实来自优化。我们要优化一个系统的前端部分，有这么一些事情可以做：

* HTML结构的优化，减少DOM树的层次等等
* CSS渲染性能的优化，批量写入DOM变更之类
* 资源文件的优化，比如小图片的合并，图像格式的处理，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
* 加载字节量的优化，主要是分摊的策略
* HTTP请求的优化

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。但是前端的人没法去优化后端的东西，这是不同的协作环节，所以就很麻烦。

但是，如果有了全栈，这个问题就可以解决了。

### 后端技术选择

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。Node.js应用JavaScript语言写成，在Node.js运行时运行。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。

Node.js提供事件驱动和非阻塞I/O API，可优化应用程序的吞吐量和规模。这些技术通常被用于实时应用程序。

Node.js采用Google的V8引擎来执行代码。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。

Node.js正在向服务器端平台发展，并已被IBM、Microsoft、Yahoo!、Walmart、Groupon、SAP、LinkedIn、Rakuten、PayPal、Voxer和GoDaddy等采用

### 数据库技术

MongoDB是一种文档导向数据库管理系统，由C++撰写而成，以此来解决应用程序开发社区中的大量现实问题。2007年10月，MongoDB由10gen团队所发展。2009年2月首度推出

# 前端开发

## 前端开发概述

由于Web前端技术兴起的时间不长，因此还没有明确的界限定义，不同的Web项目中可能要求的Web前端开发技术会有所不同。比如，某些项目可能需要前端开发人员了解一些后端技术，这样前端开发人员才可以更好地与后端开发人员配合，比如在页面上留下一些后端需要调用的钩子等，而某些项目可能需要前端开发懂一些UI设计、Photoshop工具的使用方法等，以便于和UI设计师沟通和配合。尽管Web前端开发的范畴广泛，并且界限模糊，但是以下7点则是Web前端开发必备的技能。

### 页面标记（HTML）

由于页面HTML代码结构基本固定，HTML的标签数量也不多，所以从学习的难易程度来说，HTML应该是前端技术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一个结构良好的页面来。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代码不但会影响其本身的展现，而且与其对应的CSS和JavaScript代码也会变得难以编写和维护。

### 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也是前端开发需要掌握的核心内容。丰富的CSS样式能让平淡的HTML展现出绚丽的效果，使得页面更为友好。好的样式可以让用户在页面上停留的时间更久一些，也可以帮助用户更好地阅读网站内容，同时，还可以让用户在不同浏览器上有着相同的体验。CSS和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前端UI开发必备技能。

### 前端编程

前端编程技能主要指定是JavaScript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JavaScript代码更能让前端开发人员找到后端程序员的感觉。JavaScript是非常灵活的脚本语言，包含了高阶函数、动态类型以及灵活的对象模型等强大的语言特性，当然，JavaScript的灵活性也导致代码不易维护。此外，浏览器的兼容性也增加了JavaScript编码的难度。同一个功能，可能在不同的浏览器中有不同的实现， 例如，在IE浏览器中事件绑定使用的是attachEvent()方法，但其它浏览器则使用的是addEventListener()方法。开发人员在熟悉JavaScript基本语法和基本的编码规范之外，还应该掌握在不同浏览器中JavaScript的兼容性问题。另外，作为前端开发工程师，必定会涉及后端的编程，一些原因是目前流行的Web编程方式会有部分后端代码存在于前端页面中，和前端的HTML、JavaScript等混合在一起，比如PHP、JSP、ASP.net等，所以前端工程师也有必要了解一些后端编程技术。

### 跨平台，跨浏览器

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。如何在众多移动平台、众多屏幕尺寸上展现友好的Web站点成为了一项前端技能。不过，目前跨浏览器没有像几年前表现得那么突出了，这要归功于IE6、IE7浏览器的占有率快速下降和众多浏览器对标准的重视，另外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。尽管如此，但是还需要熟悉常见的浏览器兼容方法，主要包括：IE7、IE8的兼容，HTML5中新特性的兼容等。

### 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。优秀的前端框架可以很大程度上减少项目开发的周期，尤其是jQuery，几乎成为了Web项目默认的前端框架。但是前端框架的接口众多，各种框架的使用方式和编码方式也不尽相同。作为前端开发工程师，需要熟悉一些常用框架的使用方法，并且要了解如何编写常用框架的扩展插件，如：jQuery、YUI、ExtJs等。

### 调试工具

对于前端代码，在调试过程中需要查看页面的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。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。 开发工程师需要熟练使用这些工具来辅助完成前端代码的调试。

## 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

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）。

MVC模式最早由Trygve Reenskaug在1978年提出 ，是施乐帕罗奥多研究中心（Xerox PARC）在20世纪80年代为程序语言Smalltalk发明的一种软件架构。MVC模式的目的是实现一种动态的程序设计，使后续对程序的修改和扩展简化，并且使程序某一部分的重复利用成为可能。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。软件系统通过对自身基本部分分离的同时也赋予了各个基本部分应有的功能。专业人员可以通过自身的专长分组：

* （控制器 Controller）- 负责转发请求，对请求进行处理。
* （视图 View） - 界面设计人员进行图形界面设计。
* （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)。

MVC框图如图2-1：



图2-1 MVC框图

### ejs模板

## 前端开发详细设计

下来我们详细介绍一下各个框架和类库。

### jquery框架

jQuery是一套跨浏览器的JavaScript库，简化HTML与JavaScript之间的操作。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。全球前10,000个访问最高的网站中，有65%使用了jQuery，是目前最受欢迎的JavaScript库

### bootstrap类库

Bootstrap是一组用于网站和网络应用程序开发的开源前端（所谓“前端”，指的是展现给最终用户的界面。与之对应的“后端”是在服务器上面运行的代码）框架，包括HTML、CSS及JavaScript的框架，提供字体排印、窗体、按钮、导航及其他各种组件及Javascript扩展，旨在使动态网页和Web应用的开发更加容易。

在2016年5月时，是GitHub上面被标记为“Starred”次数排名第二最多的项目。Starred次数超过96,000，而分支次数超过了42,000次，被标记数如下图2-1

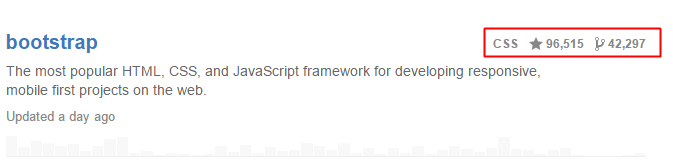


图2-2 GitHub上面被标记为“Starred”次数

### 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。而EJS专为HTML设计，因此可以做许多传统模板做不到的专门针对html的优化。

EJS是一个JavaScript模板库，用来从JSON数据中生成HTML字符串。

# 服务器开发

## 服务器概述

网页服务器（Web server）一词有两个意思：

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

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。

## 服务器开发总体设计

作为一名网络开发人员，有时会发现自己在不断构建相同的页面；此时应当好好考虑采用一种服务器端技术。如果唯一不同的是文本或是一些图片，Adobe Dreamweaver 模板十分适合小型站点以及更改不太频繁的内容。但模板有一个缺点，即需要单独构建每个页面并将它上传到远程服务器。如果更改模板，则需要重新上传每个文件。

服务器端技术从最简单的层面允许您使用一个外部文件将相同的代码包含在多个页面中。例如，您可以将站点菜单放入一个外部文件，将它包含在每个页面中。每当更新菜单时，您只需将这个文件上传到 Web 服务器，它会自动更新每个页面。它与使用模板十分相似，但您无需在每次更改时逐一上传每个文件。因此，它更快、更高效。

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。选择使用哪种服务器端技术是一个重要的决策。我们无法说哪一种技术更好，但每种技术有一些方面可以帮助您做出决定，如易学性、可用性、成本及软件支持

### 基于V8引擎的服务器语言nodeJs介绍

看到Node.js这个名字，初学者可能会误以为这是一个Javascript应用，事实上，Node.js采用C++语言编写而成，是一个Javascript的运行环境。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\*nux、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。

### npm包管理器

NPM（node package manager），通常称为node包管理器。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。

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。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。

npm官网：https://npmjs.org/

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

## 服务器搭建详细介绍

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服务器开发的那些事儿。

### 下载

到官网里下载镜像安装包

下载地址：https://nodejs.org/en/

### 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。

C:\Users\Administrator\Desktop\BaiduShurufa_2016-5-28_20-30-43.png

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

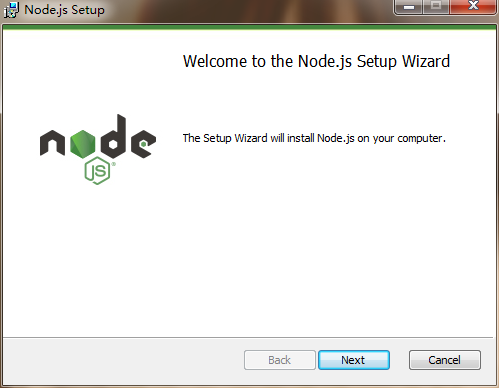


图3-2 Nodejs安装欢迎界面

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

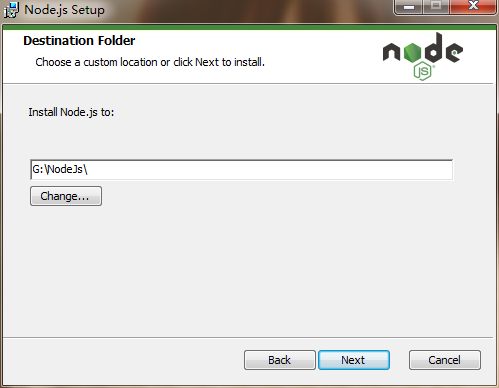


图3-4node择安装目录

图3-5: 选择安装功能页，点击next进入安装页

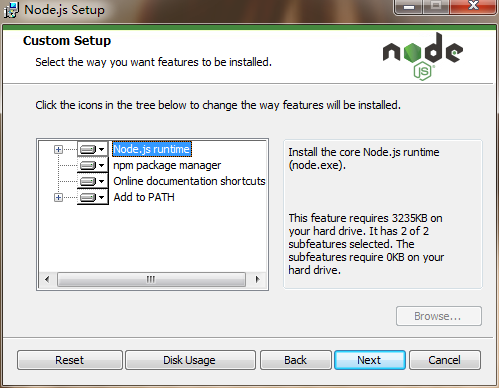


图3-5 node选择安装功能

图3-6: 安装页，点击instal进入安装

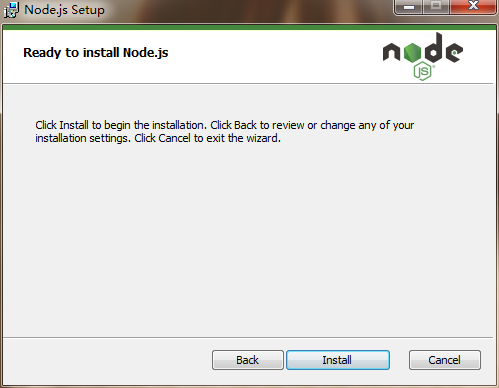


图3-6安装页

等待安装完成！

到这里，nodejs就已经安装成功了。

### 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

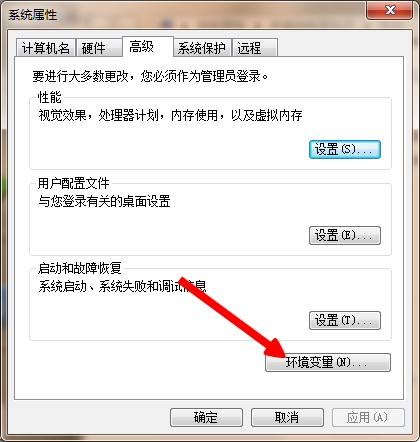


图3-8环境变量

找到Path，点击编辑，如图3-9

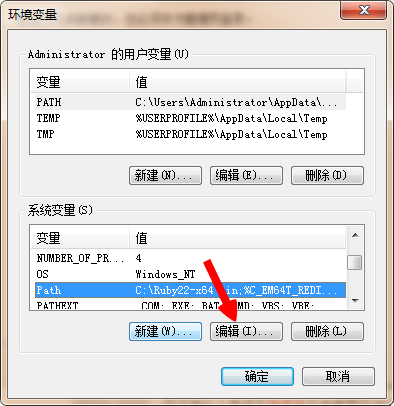


图3-9环境变量

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。

### Npm包

Npm包是nodeJS开发必不可少的一部分，本次课题也是基于一个叫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 installnpm start // 启动服务器

到这里我们的服务器就启动起来了

# 系统测试与分析

## 测试目的

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

## 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，用chrome浏览器打开项目平台，去点击页面上的每一个连接，来看平台上的所有链接是否都跳转到了它该跳转的地方，整个网站是不是都是互通的

## 性能测试

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。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。

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

## 调试与分析

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。而在互联网开发越来越重要的今天，如何在前端开发中降低开发成本，提升开发效率，掌握前端开发调试技巧尤为重要。讲解各种JS的调试技巧。

### 骨灰级调试大师Alert

那还是互联网刚刚起步的时代，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（’some word’）;

虽然，alert的调试方式很原始，但当时确实有它不可磨灭的价值，甚至到今天，依然有其用武之地。

### 新一代调试王者Console

随着JS在Web前端中能做的事情越来越多，责任越来越大，而地位也越来越重要。传统的alert调试方式已经渐渐不能满足前端开发的种种场景。而且alert调试方式弹出的调试信息，那个窗口着实不太美观，而且会遮挡部分页面内容，着实有些不太友好。

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会阻碍页面的继续渲染。这就意味着开发人员调试完成后，必须手动清除这些调试代码，实在有些麻烦。

所以，新一代的浏览器Firefox、Chrome，包括IE，都相继推出了JS调试控制台，支持使用类似”console.log(xxxx)”的形式，在控制台打印调试信息，而不直接影响页面显示。代码是这个样子的：

console.log(‘some word’);

### JS断点调试

断点，调试器的功能之一，可以让程序中断在需要的地方，从而方便其分析。也可以在一次调试中设置断点，下一次只需让程序自动运行到设置断点位置，便可在上次设置断点的位置中断下来，极大的方便了操作，同时节省了时间。——百度百科

JS断点调试，即是在浏览器开发者工具中为JS代码添加断点，让JS执行到某一特定位置停住，方便开发者对该处代码段的分析与逻辑处理。Chrome浏览器中自带断点调试功能：Sources断点，它的使用教程可以在网上搜索到。

### DOM断点调试

DOM断点，顾名思义就是在DOM元素上添加断点，进而达到调试的目的。而在实际使用中断点的效果最终还是落地到JS逻辑之内。

当这些情况发生时都可以打断点：

* 当节点内部子节点变化时断点
* 当节点属性发生变化时断点
* 当节点被移除时断点

### XHR Breakpoints

这几年前端开发发生了翻天覆地的变化，从当初的名不见经传到如今的盛极一时，Ajax驱动Web富应用，移动WebApp单页应用风生水起。这一切都离不开XMLHttpRequest对象，而“XHR Breakpoints”正是专为异步而生的断点调试功能。

### Event Listener Breakpoints

事件监听器断点，即根据事件名称进行断点设置。当事件被触发时，断点到事件绑定的位置。事件监听器断点，列出了所有页面及脚本事件，包括：鼠标、键盘、动画、定时器、XHR等等。极大的降低了事件方面业务逻辑的调试难度

# 第5章 结论

要成为一名全栈工程师，至少要使用过多种编程语言在各种领域的不同平台工作多年。全栈工程师除了像“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，知道如何运用操作系统、云存储、网络资源等知识，了解应用的规模大小，访问量的预判定。

接下来是数据建模，如果数据模型存在缺陷，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。

再下来是业务逻辑，这是系统应用价值的关键所在。掌握SOLID设计能力（单一职责，开放且封闭Liskov替代性，接口分离，依赖注入）。熟悉一些常用的框架。

在就是框架的事情， MVC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。做为一名全栈开发工程师，必须得有能力写出一致、清晰、使用简单的接口。

还有就是用户界面，全栈工程师需要：a）懂得如何构建易读的布局；b）能意识到艺术家和图形设计师的作用。总之，实现一套好的视觉设计方案尤为关键。熟练掌握HTML5/CSS。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）。因此，掌握JavaScript必不可少。

比较重要的还有用户体验。全栈工程师应该意识到：用户仅仅是希望一切运转良好。一个好的系统是不会让用户得腕管综合征或产生眼疲劳的。全栈工程师能审视全局，可以将一个需要8次点击+3个步骤的过程，简化为仅需要一次点击。知道如何向用户提示错误信息。如果出现故障，真诚道歉。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。全栈工程师要领悟客户使用软件过程中会需要些什么以及整个市场需要些什么。

将代码与某个特定的实现（无论是软件库，操作系统，硬件，等等）紧绑在一起并不是好的实践。全栈工程师虽然学术渊博，但不意味着他们能抄捷径，懂得东西多了，精通的东西自然就少了，这是必然的。

学习的根本是为了适应市场的需求，技术型的创业企业在初期为了满足灵活性，对全栈工程师的需求比较大，但是，随着企业的成熟，会需要越来越多的专业型人才。

参考文献

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

[2] Nicholas C.Zakas．JavaScript高级程序设计[M]. 人民出邮电出版社. 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]. 人民出邮电出版社. 2009年;

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

[6] Kristina Chodorow. 深入学习MongoDB 人民出邮电出版社 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'});

});

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

});

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。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：