****

**毕业设计（论文）**

企业网站的设计与实现

|  |  |
| --- | --- |
| 学生姓名: | 戴志奇 |
| 学 号: | 2015100418 |
| 专 业: | 电子商务 |
| 指导教师: | 张小红 教授 |
| 学 院: | 电子商务学院 |

江西·新余

**独创性声明**

本人郑重声明：

所呈交的毕业论文是本人在指导教师指导下进行的研究工作及取得的研究成果。其中除加以标注和致谢的地方，以及法律规定允许的之外，不包含其他人已经发表或撰写完成并以某种方式公开过的研究成果，也不包含为获得其他教育机构的学位或证书而作的材料。其他同志对本研究所做的任何贡献均已在文中作了明确的说明并表示谢意。

本毕业论文成果是本人在江西工程学院期间在指导教师指导下取得的，成果归江西工程学院所有。

特此声明。

作者签名（手写）： 签名日期： 年 月 日

**版权使用授权书**

本毕业论文作者及指导教师完全了解江西工程学院有关保留、使用毕业论文的规定，有权保留并向国家有关部门或机构送交毕业论文的复印件和磁盘，允许毕业论文被查阅和借阅。

作者签名（手写）： 指导教师签名（手写）：

日期： 年 月 日 日期： 年 月 日

**论文题目：企业网站的设计与实现**

**专 业：电子商务学院**

**学生姓名：戴志奇**

**指导教师：张小红教授**

摘要

随着我国对互联网技术的重视，计算机技术的飞速发展，互联网在我国有着举足轻重的地位。互联网已经渗透人们生活的方方面面，包括经济生活，和社会生活，已经是不可或缺的一部分对于人们来说。互联网对于人们已经是生活的一部分，所以对于企业来说可以借助互联网发展自己的特点，自己的文化，让企业在互联网中出传播。网络平台现在已经是各行各业的竞争平台。对企业而言，现在不单单是在实体经济上面发展，还得在网络上提高自身的知名度。所以企业网站孕育而生。

一个优秀的网站会给企业带来不一样的竞争力，也在知名度等其他当面发挥着作用。企业商业模式也会随着当下的互联网模式发生改变。网站的运营也会是一个企业提升知名度的一种方式。由此可见，一个好的网站会给企业带来知名度的提升，而且也具有象征意义。

本文主要通过代码制做对企业网站的建设和研究，进行了设计。对于当前一些企业的不完善的需求做出了一些改进，比如：对数据的分页，前端的分页，和一些js代码事件的完善。采用laravel作为其基础技术框架;采用MySQL作为数据库管理工具。JavaScript，css作为前端设计工具。建立了4张基本数据库表，对4张表进行详细介绍;系统服务端采用基于php语言编写，采用开源软件Apache作为网站服务器。根据系统的功能需求，设计了登录模块、管理员模块、权限模块、角色模块、会员模块。全站共计5个功能模块，并对功能模块的关键算法进行了说明。

**关键词**：php；网站设计；数据交互；JavaScript； css ajax

**Title: Design and Construction of Enterprise Website**

**Specialty: Electronic information engineering**

**Applicant: Zhiqi Dai**

**Supervisor: Prof. Xiaohong Zhang**

Abstract

With the emphasis on Internet technology in China and the rapid development of computer technology, the Internet has a pivotal position in China. The Internet has penetrated every aspect of people's lives, including economic life, and social life, and is already an indispensable part for people. The Internet is already a part of people's lives, so companies can use the Internet to develop their own characteristics, their own culture, and let enterprises spread on the Internet. The network platform is now a competitive platform for all walks of life. For enterprises, it is not only in the real economy, but also to increase their visibility on the Internet. So the corporate website was born.  
A good website will bring different competitiveness to the company, and it will also play a role in other aspects such as popularity. The business model of the enterprise will also change with the current Internet model. The operation of the website will also be a way for companies to increase their visibility. It can be seen that a good website will bring awareness to the company and it is also symbolic.  
This paper mainly designs and researches the website of the enterprise through code system. Some improvements have been made to the current imperfect needs of some companies, such as paging of data, paging of front ends, and improvements to some js code events. Use laravel as its basic technical framework; use MySQL as a database management tool. JavaScript, css as a front-end design tool. Four basic database tables were created, and four tables were introduced in detail. The system server was written based on the php language and the open source software Apache was used as the web server. According to the functional requirements of the system, the login module, administrator module, permission module, role module and member module are designed. There are a total of five functional modules in the station, and the key algorithms of the functional modules are described.

**KEY WORDS:**php; Website design; Data interaction;JavaScript;css ajax

**目录**

[第一章 绪论 1](#_Toc27697)

[1.1选题的依据和意义 1](#_Toc19347)

[1.2国内外有关本选题研究的动态 1](#_Toc28478)

[1.2.1 国外研究 1](#_Toc7038)

[1.2.2 国内研究 2](#_Toc31463)

[第二章 网站开发关键技术 3](#_Toc29276)

[2.1页面开发技术 3](#_Toc17504)

[2.2 HTML技术 3](#_Toc30944)

[2.1.2 CSS技术 4](#_Toc8360)

[2.1.3 javaScript技术 5](#_Toc19105)

[2.2 AJAX技术 5](#_Toc21815)

[2.3 服务器开发技术 6](#_Toc5800)

[2.3.1 php技术 6](#_Toc32426)

[2.3.2Laravel框架技术 6](#_Toc4563)

[2.4 mysql数据库技术 6](#_Toc17984)

[第三章 企业网站的项目介绍 7](#_Toc12526)

[3.1 项目背景 7](#_Toc9201)

[3.2 项目功能 7](#_Toc4335)

[3.3项目的技术点 8](#_Toc6740)

[3.4 开发环境 9](#_Toc26042)

[第四章 企业网站的初始化 11](#_Toc698)

[4.1.环境的搭建 11](#_Toc19539)

[4.1.1创建laravel项目 11](#_Toc29482)

[4.2项目本地化操作 11](#_Toc883)

[4.3企业网站需求分析 12](#_Toc21786)

[4.4后台网站的设计 12](#_Toc26064)

[4.4.1用户模块设计 12](#_Toc15134)

[4.4.2 RBAC权限功能模块功能设计 14](#_Toc7066)

[4.4.3会员功能的设计 15](#_Toc13539)

[4.5配置数据库连接 15](#_Toc25856)

[4.6 数据库设计 16](#_Toc7216)

[第五章 后台的基础功能开发 19](#_Toc11704)

[5.1创建管理员数据表 19](#_Toc27335)

[5.2模拟管理员表的数据 19](#_Toc23001)

[5.2.1创建填充器文件 19](#_Toc26704)

[5.2.2执行填充器代码： 20](#_Toc32404)

[5.3后台登录页面展示 21](#_Toc14528)

[5.3.1 确定路由 21](#_Toc3459)

[5.4管理员登录的自动验证 22](#_Toc22233)

[5.4.1提交的路由地址 22](#_Toc28586)

[5.4.2编写check方法 22](#_Toc9135)

[5.5管理登录身份验证 23](#_Toc13671)

[5.5.1 guard实例 23](#_Toc15195)

[5.5.2创建需要使用的模型（guard需要使用） 24](#_Toc27353)

[5.6后台首页的展示 25](#_Toc67)

[5.6.1定义后台首页的路由 25](#_Toc23689)

[5.6.2编写index 方法、welcome方法 25](#_Toc6557)

[5.6.3创建视图文件 25](#_Toc13916)

[5.6.4 动态获取用户登录的用户名 26](#_Toc18204)

[5.7路由保护 27](#_Toc30487)

[第六章 后台内容的完善 29](#_Toc15074)

[6.1实现管理员列表的展示 29](#_Toc17659)

[6.1.1先定义需要的路径 29](#_Toc29744)

[6.1.2创建需要的控制器文件 29](#_Toc21570)

[6.2 RBAC权限管理 30](#_Toc10348)

[6.2.1介绍 30](#_Toc22146)

[6.2.2数据表的设计 31](#_Toc30062)

[6.2.3理清三表之间的关系 32](#_Toc25164)

[6.3权限管理操作 32](#_Toc12095)

[6.3.1展示相关的视图，并添加权限页面，列表页面 32](#_Toc31859)

[6.3.2实现权限列表功能 35](#_Toc27810)

[6.4角色管理操作 35](#_Toc30946)

[6.4.1前置工作处理（数据模拟、模型+控制器创建） 35](#_Toc20775)

[6.4.2实现角色的列表功能 36](#_Toc22264)

[6.4.3实现数据保存 38](#_Toc18585)

[6.5使用中间件进行RBAC的鉴权操作 38](#_Toc7597)

[6.6会员管理 39](#_Toc18172)

[6.6.1创建数据表 39](#_Toc7707)

[6.6.2建立测试数据 40](#_Toc31109)

[6.6.3路由与控制器、模型的创建 40](#_Toc30500)

[6.6.4实现列表展示 41](#_Toc30757)

[6.6.5实现会员的添加功能 42](#_Toc19257)

[第七章 登录页面的设计 45](#_Toc17570)

[7.1登录页面设计 45](#_Toc24996)

[7.1.1登录页面前端效果图 45](#_Toc2907)

[7.1.2登录页面勾选的设计 46](#_Toc30388)

[7.1.3在登录页面上显示错误信息 46](#_Toc31514)

[7.1.4后台首页的设计 47](#_Toc23234)

[7.1.5实现datatables列表无刷新分页 48](#_Toc29871)

[7.2 ajax异步请求 49](#_Toc15849)

[7.2.1ajax的数据提交 49](#_Toc6183)

[7.2.2实现头像异步上传 51](#_Toc14048)

[第八章 本文总结与展望 55](#_Toc9423)

[8.1总结 55](#_Toc17762)

[8.2展望 55](#_Toc2679)

[参考文献： 57](#_Toc21758)

[致谢 59](#_Toc15749)

# 绪论

## 1.1选题的依据和意义

当前互联网飞速发展，企业网站相当于一个人的名片，名片的精美程度，反应出网站的特点，优点。但是企业的网站的功能不止是名片那么简单，它的功能强于名片，多余名片。网站的特点不受于时间，空间的影响。客户可以随时随地的访问。而网站内容的展示就是企业所要思考的问题，怎么能一下子抓住客户的眼球，形象生动的展示自己的产品和特点。这就需要网站的设计，一个网站设计得好可以让企业大受客户的青睐，企业的产品、理念、历史、文化可以被客户熟知。企业的网站也相当与在网上开了一家公司，这是企业发展潜在的资源和环境。

网站以各种形式制作，大小不一，设计精美，信息丰富的网站给人以企业形象的印象，因此企业网站建设无疑是推广企业产品和服务的契机。

电子商务是一种新兴的商业模式，从不同的角度来看有不同的定义。可以理解，通过互联网，借助某些信用支付方式，产品的购买和销售，网络中的费用以及其他商业活动的支付。电子商务是未来经济形态发展的重要趋势。目标是建立有关网络连接和交易的电子信息。公司经常通过互联网访问行业协会或商业网站的网站，成为网络的成员或成员。在行业协会或商业网站的网站上发布供求信息，获取有关政策和市场的相关信息，享受其他服务。从销售角度来看，创建网站的公司可以减少交易中间并降低成本。企业网站也可以扩展到在线售后和销售咨询服务中心。

发布企业广告，网站作为媒体形式，企业网站本身就是一个广告，一些公司在互联网上建立自己的网页，或者打开自己的网站，集中企业信息，排序和列，便于浏览。 现代社会的所有知名公司都在网上建立了自己的网页或网站。

综上所述，企业建设网站是在互联网上开设分支机构，让更多人了解公司的产品和服务，为企业带来更多潜在客户和咨询，是一种具有高性价比的新型宣传模式，是当前的趋势，所以如果企业想在竞争中立于不败之地，得紧跟时代步伐，并使网站制作，那就是首选。

## 1.2国内外有关本选题研究的动态

### 1.2.1 国外研究

目前，国外在网站界面设计方面取得了很多成果，形成了一些比较成熟的理论。 他们一直处于网站界面设计的最前沿。他们的作品始终关注用户的体验和创新意识。引领网站界面设计的发展趋势。还有很多书籍对网站界面设计提出了自己的看

法，如比尔.斯科特《Web界面设计》[1]，着重总结提炼了WEB界面设计的六大原则-直截了当、简化交互、足不出户、提供邀请、使用变换和即时反应，并阐明了数以百计行之有效的最佳实践，史蒂夫克鲁克写的《Don't make me think》[2]，从WEB界面设计的可用性方面做了详细的阐述，彼得.莫维尔写的《WEB信息架构》[3]分别介绍了信息构架的组织系统、标签系统、导航系统、搜索系统等，并分别介绍了两个案例研究，为网站界面设计中使用信息架构提供了良好的参考价值。

与往年相比，根据书籍和在线学术论文的回顾，目前对大学门户网站的学术研究逐年增多，但对地方工科院校界面设计的研究还不够理想。主要原因在于以下几个方面，大多数本地工程门户设计师都是计算机专业诞生的，网站界面的设计稍微不那么美观。

WangXH[4]分析网络客户的需求特点和消费心理，提出在企业网站建设过程中，要以满足Web客户的个性心理状态为前提，以尊重网络客户的心理认知为指令，以网络顾客的情感体验为核心，以Web客户意愿形成传递。ZhaoHong认为企业网站建设包括四个基本元素:结构、内容、服务和功能。网站建设面向用户，网站设计面向搜索引擎，网站建设基于网站可信度。

电子商务也需要新生事物来刺激企业发展的感官。传统模式的在原有的基础上应做不变的保留，因为这是历经检测得出来的有效展示方式[5]。

### 1.2.2 国内研究

我国国内最开始的是B2C类网站平台，比如当当网。C2C类平台也紧随其后，其中名声最大的即是淘宝网。在通过长时间的比拼，当前B2C类平台有京东、苏宁，这种大企业的网站。但是就企业而言，第三方平台不能够满足企业对企业介绍、产品展示、产品销售、产品服务的需求回[6]。

2017年11月27日国务院发布了《关于深化“互联网+先进制造业”发展IT业互联网的指导意见》，为我国工业互联网的发展指明了方向，也是工业互联网发展的纲领性文件。自2018年2月国家宣布成立工业互联网专项工作组以来，工业互联网在中国制造2025战略中的地位得到提升，未来国家层面的推广将进一步加快。

2018年6月7日，工信部印发《工业互联网发展行动计划(2018-2020年)》和《工业互联网专项工作组2018年工作计划》；2018年7月9日，印发《工业互联网平台建设及推广指南》和《工业互联网平台评价方法》，《指南》提出，到2020年，培育10家左右的跨行业跨领域工业互联网平台和一批面向特定行业、特定区域的企业级工业互联网平台，工业互联网平台生态初步形成。至2018年8月，我国主要制造业省市又密集出台各自的工业互联网产业发展政策，以此推动制造业转型升级。巨大的市场规模和政策的扶持，使得工业互联网发展迅速，前景可期。

# 网站开发关键技术

## 2.1页面开发技术

《设计动态网站的最佳方案:Apache+PHP+MySQL》[8]：Apache 是目前应用最广的Web 服务器；PHP 是一种服务器端脚本语言，而MySQL 是一个小巧的数据库系统软件，它们特别适用于网站建设。Apache+PHP+MySQL 不仅仅是开源项目，可免费获取，而且它支持Linux、UNIX、OS/2 和Windows 多个操作系统，可移植性好。因此，这种组合是设计动态网站的最佳解决方案。在Windows 操作系统平台下选择Apache+PHP+MySQL 组合的网站的开发原理、系统功能的具体设计与实现。

动态网页工作方式具有信息发布和交流简便快捷、更新速度快的特点。但是, 由于动态网页设计技术是将众多的脚本编程语言、数据库、电脑美术设计等技术相整合而形成的一门交叉技术, 设计和管理动态网站都有一定的难度[7]。

2.2 HTML技术

HTML是英文Hype Text Markup Language的缩写，中文意思是超文本标记语言。HTML页面的内容提取是web页面聚类和信息检索的基础，因此有必要消除杂乱的信息，准确提取页面的内容非常重要。网页是用HTML编写的纯文本文件。用户通过浏览器看到的每个网页都包含多媒体信息，如文本，图像，动画等，实质上，浏览器解释纯文本文件并引用相应的图像，动画和其他资源文件来生成丰富多彩的文件网页。

根据Internet的特点, 网站应该开始从客户的角度来考虑问题, 因为决定您的站点是否成功是来此访问的那些客户, 只有您的客户经常性的来访问您的站点, 花时间来阅读您的内容, 您才能达到制作站点的目的[9]。

h5也解决跨浏览器的问题，如《基于HTML5技术企业网站设计》：

在h5之前，浏览器之间兼容性很差，JavaScript语言编写还得考虑刘篮球的兼容性，这给技术人员带来很大的困恼。因此市场需要一个规范标准，所以h5可以满足这个标准。现在各个浏览器对h5的支持非常好，国外的Chrome(谷歌浏览器)、Safari等，国内的360浏览器,搜狗浏览器等。

HTML是一种标记语言，可以认为，HTML文档就是普通文本+HTML标记，而不同的HTML标记能表示不同的效果，如表格、图像、表单、文字等。HTML文档可以运行在不同的操作系统平台和浏览器上，是所有网页制作技术的基础。无论是在WEB上发布信息还是编写可供交互的程序，都离不开HTML语言。HTML的结构包括头部(Head)、主体(Body)两大部分，其中头部描述浏览器所需的信息，而主体则包含所要说明的具体内容。

使用HTML编写网页不需要专门的工具，只需能够进行文字编辑的软件即可。HTML通过标记来描述网页的元素。标记是HTML的标记符号和用标记符号构成的各种元素的统称。标记用一对“<>”括起来，通常成对出现，起始标记和终止标记之间的内容称为元素。一个HTML文档以“.html”或者“.htm” 为扩展名，基本格式如下：

<!doctype html>

--文档申明

<html>

<head>

--头部文件

<meta charset="utf-8">

<title></title>

--标题

<body>

--内容

</body>

</html>

一个HTML页面从<html>标记开始到</html>标记结束,包括写在<head></head>之间的文档头部内容和文档的主体<body></body>之间的内容。头部-般包括文档标题、页面属性等，这些内容不显示在页面上。主体部分可以描述页面的各种内容，一般而言所有内容都可以在页面上显示。

2.1.2 CSS技术 CSS (Cascading Styles Sheets, 层叠样式表)是用于控制网页样式并允许将样式信息与网页内容分离的一种标记性语言。CSS是为了弥补HTML在显示方面的不足而制定的一套扩展样式标准。CSS可以重新定义html的标签样式，并引入类、层的概念。这样可以统一管理html标签相同的样式。CSS犹如一个化妆师，给html静态网页修饰作用，单独Html网页页面样式比较单调，所以需要css来修饰html的内容。这样才能让网页结构和内容看起来简约简单。也可以方便修改网页的风格特点，因为CSS和html页面区分开来，且互不干扰，只需要加载CSS样式即可。Css丰富的样式也它的特点之一。

CSS样式表可以嵌入HTML网页文件中，也可以单独作为一个文件被引入html文件中，样式表的定义格式为:选择符{规则表}。

### 2.1.3 javaScript技术

JavaScript是-种脚本语言。脚本(Script) 是-段可以嵌入到其他文档中的程序，用来完成某些特殊的功能。脚本既可以运行在浏览器(称为客户端脚本)，也可以运行在服务器(称为服务器脚本)。

客户端脚本通常用于验证用户提交的数据，比如登录页面的用户名，和密码是否符合验证规则，这样可以减少对服务器的请求。同时也可以动态地修改html的属性和CSS的属性，js代码也通常用在响应用户操作的事件。因为脚本代码就在客户端，可以减少对服务器的响应，减少服务器的负载。

JavaScript可以完成以下任务:

1）JavaScript是一种脚本语言，所以无需运行环境，它在任何环境下都可以运行，包括在html页面中。同时也完善了html页面的不足。  
2）JavaScript可以动态地添加内容。  
3）JavaScript可以响应事件，只要浏览器有事件驱动，js就可以响应事件，对该事件作出相应的处理。  
4）JavaScript可以动态找出html的属性和csss属性，可以动态地修改属性内容。

5）JavaScript可以验证表单数据，对于表单数据是否符合规范。  
6）JavaScript可以创建和读取cookie，为客户提供个性化服务。

## 2.2 AJAX技术

Ajax 是一种交互技术，可以由客户端远程发送异步请求到服务器端，服务器端接收到请求数据，并返回给客户端作http响应。该技术一般用于表单验证、搜索请求等页面请求。因为客户在填写表单验证的时候，不必等客户点提交的是时候才去验证或对比数据库中的数据，这个时候可以利用ajax技术提交一部分异步请求。这样分布式访问可以减少web服务器端的请求压力，也可以增加访问页面的流畅性。

AJAX交互模型：  
同步：脚本会停留并等待服务器发送回复然后再继续。  
异步：脚本允许页面继续其进程并处理可能的回复。  
跨域问题简单的理解就是因为JS同源策略的限制，a.com域名下的JS无法操作b.com或c.a.com下的对象，具体场景如下：   
（1）如果是端口或者协议造成的跨域问题前端是无能为力的 。  
(2) 在跨域问题上，域仅仅通过URL的首部来识别而不会尝试判断相同的IP地址对应的域或者两个域是否对应一个IP 。  
前端对于跨域的解决办法：   
(1) document.domain+iframe 。  
(2) 动态创建script标签 。

## 2.3 服务器开发技术

### 2.3.1 php技术

Php：[超文本](https://baike.baidu.com/item/%E8%B6%85%E6%96%87%E6%9C%AC" \t "https://baike.baidu.com/item/php%E7%BC%96%E7%A8%8B%E6%8A%80%E6%9C%AF/_blank)预[处理器](https://baike.baidu.com/item/%E5%A4%84%E7%90%86%E5%99%A8" \t "https://baike.baidu.com/item/php%E7%BC%96%E7%A8%8B%E6%8A%80%E6%9C%AF/_blank)（PHP: Hypertext Preprocessor）。Php是开源的，基于C语言开发的一种服务器端语言，可以直接从官方网站下载。php语法融合了c语言和java语法，也可以面向对象编程（pdo）。Php语言也具有一般语言的特性：抽象、封装、继承和多态。Php因为是服务器端的语言，所以一般与Apache一起搭配，还有数据库mysql。Php也可嵌入前端代码中，比如嵌入html页面中，实现数据的交互。Php的灵活运用，也广受网页制作的程序员青睐，所以一般在制作网页中都会选择php作为后台轻量级网站的开发。Php一般也与数据库mysql搭配，PHP是目前的Web开发语言[10]。

### 2.3.2Laravel框架技术

Laravel是一个由Taylor Otwell所建立，免费的开源 PHP Web框架，旨在实现Web软件的MVC架构，并作为Codelgniter的替代方案。其源代码托管于GitHub,许可条款为MIT许可证。Laravel的特色包含:支持用户身份验证和许可，具有模块化的包系统且有专属的包库，提供连接许多种类的关联式数据库的方式，提供工具来协助应用程序的部署和维持，并且发展出许多语法糖。

## 2.4 mysql数据库技术

MySQL数据库是基于Linux操作系统开发的数据库。 Linux是一个开源操作系统。因此，MySQL数据库也是一个开源免费数据库。

Mysql也为多种语言提供了链接支持，包括java、php和c等。特别对php有良好的支持，php提供专门的函数可以直接对接数据库。Mysql也支持大型的数据库，也可编写存储过程处理上千万条的数据记录。mysql是非常灵活的，对视图也支持的非常好，如果有些数据不知道怎么处理，可以利用视图功能对数据进行增删改查。数据库对事物也支持的非常好，可以保持数据的安全性。Mysql的查询语句可以优化，可以子查询，也可以连接查询，为了优化sql语句，一般会用连接查询。

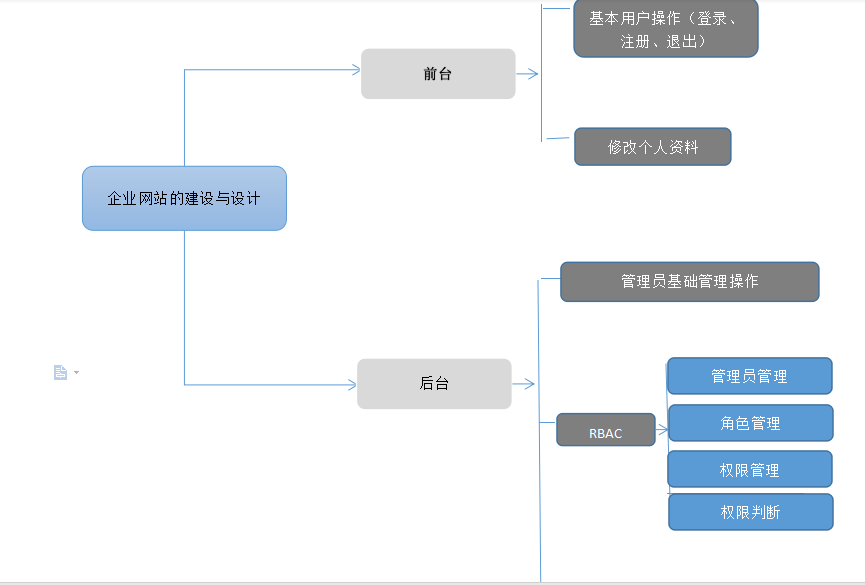
# 业网站的项目介绍

## 3.1 项目背景

企业网站的形式丰富多彩，规模大小不一，一个好的网站可以间洁，信息饱满，不会是一个空洞的网站。这样的网站可以直接展示企业或者公司的优点。产品，也可以给客户一个直观的感受。企业网站就是企业的名片，企业的形象。网站也可以说是企业的一种广告形式，这种形式可以可以推广自身的信息，自身的产品和特点，同样也是一种形象的建设，网站就是企业的形象。

## 3.2 项目功能

给企业网站建立前台功能和后台功能如图3-1，前台是客户和用户操作的界面，后台是企业内部管理人员操作的界面。分别对应不同的功能。后台有权限操作，前台有修改操作。



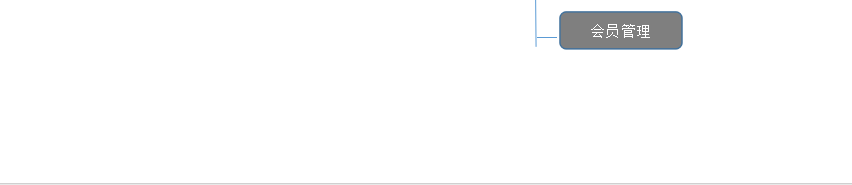


图3-1项目功能介绍

## 3.3项目的技术点

将涉及的技术点按类型进行归类，大致如下:

前端技术：为了提高前端的用户体验使用了如datatables (数据表格插件)、layer (弹窗插件)、webuploader (上传插件)、zyupload (多文件上传插件)、ckplayer (在线播放器)等稳定成型的插件;

后端技术：为了满足项目需求并且提高开发效率以及后期操作的便捷程度，后端使用了如LaravelExcel (excel文件操作类)、支付接口(用于支付，支付宝、微信均可)等技术;  
 安全技术：在项目中，对应的敏感字段，包括但不限于用户密码，都使用了不可逆加密方法对数据进行了严格的加密;其次，为了防止数据在传输过程中数据被第三方窃取篡改，项目在最终上线部署的时候使用了安全的https协议进行全程加密处理;

## 3.4 开发环境

项目开发环境：Windows+apache+php+mysql。

版本要求：Windows>winxp。

Apache>2.2。

Mysql>5.5。

Php>7.0。

开发框架：laravel(使用composer默认创建版本)。

# 企业网站的初始化

## 4.1.环境的搭建

### 4.1.1创建laravel项目

时区、本地化、数据库配置、禁用严格模式、刪除自带的测用户基础文件。

配置好php的路径下载composer。

如图：

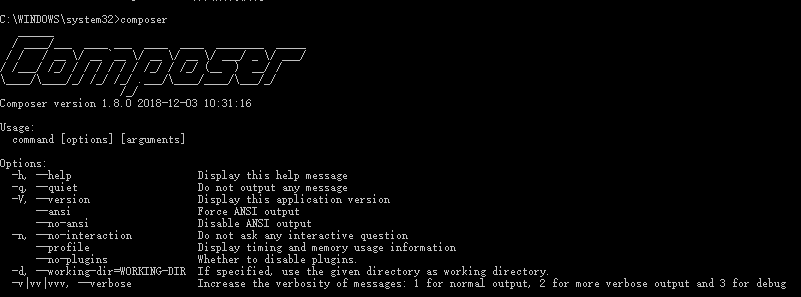
产生如下结果如图4-1：

图4-1composer 下载的结果

说明下载成功，再输入composer -v，产生如图4-2结果：

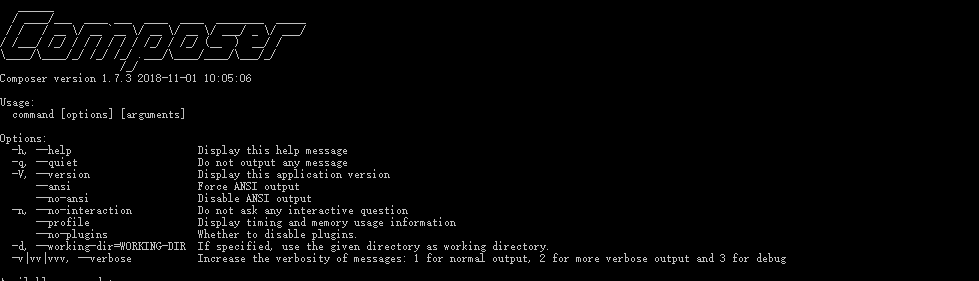


图4-2composer 安装的结果

打开电脑cmd 指令 输入：composer create-project laravel/larvel=5.4.\* --prefer/-dist \*/。

Laravel 安装成功。Php laravel框架安装成功,可以进行项目初创建。设置laravel文件的时区config/app.php，找到timezone 这个键值对。

把UTC 改为中国时区PRC。

## 4.2项目本地化操作

下载语言包：laravel-lang。

下载地址http://packagist.org/?q=laravel-lang&p=0。

也可以直接用composer命令直接下载。

指令：composer require caouecs/laravel-lang:~30。

将语言包vendor/caouecs/laravel-lang/src 复制到resource/lang目录下。

修改config/app.php中关于本地化得配置项：locale=>’zh-CN’。

## 4.3企业网站需求分析

企业网站建立的开始，需要全面了解该企业的文化，历史，背景，产品和企业的运转流程。做到对企业的细致了解和总体运行情况，那么在设计和建设网站的过程中，才能根据企业的实际情况进行设计页面布局和网络后台的数据更新。这样才能根据客户的需求设计网站，如果脱离客户的实际需求，那么网站建设无疑是空中楼阁，无疑是失败的。只有在顾客体验网站的舒适程度，我们才能设计更好的网站。而且还得让顾客体验出企业的文化和特点。

网站的动态交互式网站见识的难点和重点。设计静态页面也是可以满足需求的。如果碰上高并发请求，静态页面就可以缓解企业服务的数据交互压力，从而提升企业网站的流畅度。交互式网站会增加服务器的压力，所以交互式得考虑服务器的压力，多设计静态页面，和增加网站的负载均衡可以解决，提高客户的体验度。

## 4.4后台网站的设计

通过对后台企业网站的分析，已经明确要实现的功能，了解企业网站的业务流程和功能。对功能模块进行设计，如图4-3。

图4-3 企业网站业务流程图

### 4.4.1用户模块设计

用户注册功能与实现

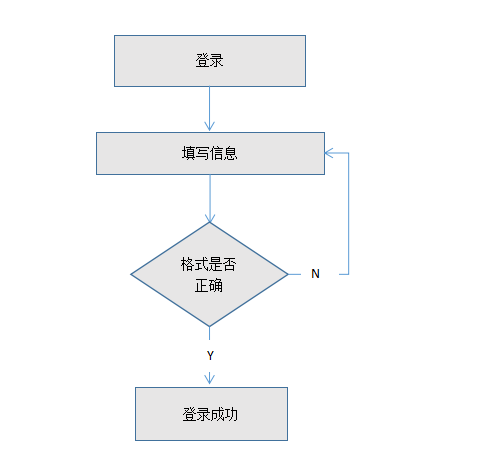


图4-4网站登录流程图

如图4-4，用户登录网站，如果登录失败，会提示错误信息，如果登录成功，跳转到主页，可以进行编辑个人信息后台等操作。

### 4.4.2 RBAC权限功能模块功能设计

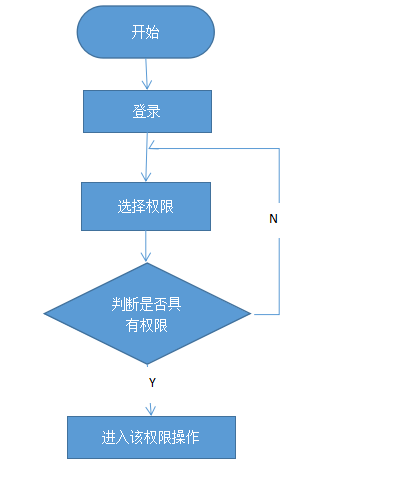


图4-5 权限管理流程图

如图4-5，当用户会员登录网站后，才可以权限操作；若管理员权限不通过，则会弹出提示你没有该权限，请重新选择权限。

### 4.4.3会员功能的设计

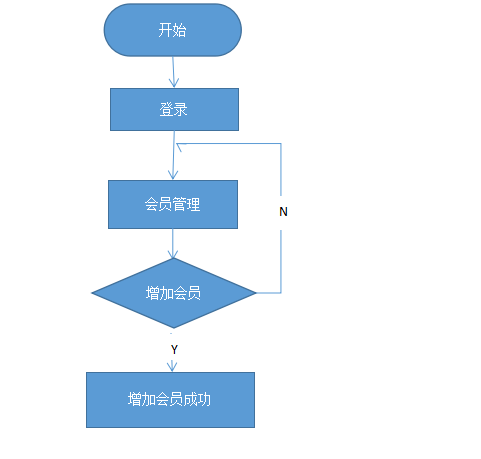


图4-6 会员增添流程图

如图4-6流程，管理员后台登录，若管理员登录成功，则会跳转管理平台主页，进行会员信息管理。若操作失败，会提示错误信息。

## 4.5配置数据库连接

先创建一个数据库数据库名称是：test\_data。数据库的创建语句是：

create database test\_data CHARACTER SET utf8 COLLATE utf8\_general\_ci。

配置数据连接需要修改.env配置文件，在laravel目录下代开.env配置文件，配置相关的参数。

推荐使用禁用Mysql的严格模式，需要修改config/database.php;

## 4.6 数据库设计

通常数据库设计是在一个已知的运行环境下，创建最佳的数据模式，用于实现数据的存储与互通，从而满足系统功能的各类具体的需要。数据库是系统设计的最核心部分，也是企业后台数据的主要部分，除了企业网站的数据展示页面，还有后台数据库的核心部分。企业通过后台的数据，可以分析出客户的特点以及企业所要针对的人群。数据库设计也是不可或缺的，数据库设计遵守三范式。三大范式只是一般设计数据库的基本理念，可以建立冗余较比小、结构合理的数据库。不过设计数据表也不一昧追求三大范式，得根据实际情况去设计数据表。

只有这样才可以让网站数据交互流畅，客户体验程度高。网站更新升级，才可以不必太过于复杂。

4-1manager后台用户表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 | manager | | | | | |
| 序号 | 字段 | 类型 | 说明 | 空 | 默认值 | 主键 |
| 1 | id | int(10) unsigned | 自增主键 | N | N/A | 是 |
| 2 | username | varchar(20) | 用户名 | N | N/A |  |
| 3 | password | varchar(255) | 密码 | N | N/A |  |
| 4 | gender | enum('1','2','3') | 性别 | N | 1 |  |
| 5 | mobile | varchar(11) | 电话 | N | N/A |  |
| 6 | email | varchar(40) | 邮件 | N | N/A |  |
| 7 | role\_id | tinyint(4) | 角色id | N | N/A |  |
| 8 | created\_at | timestamp | 创建时间 | Y | N/A |  |
| 9 | Update\_at | timestamp | 更新时间 | Y | N/A |  |
| 10 | Remember\_token | varchar(100) | Token值 | Y | N/A |  |
| 11 | status | enum('1','2') | 状态 | N | 2 |  |

4-2 role后台角色表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 | role | | | | | |
| 序号 | 字段 | 类型 | 说明 | 空 | 默认值 | 主键 |
| 1 | id | int(10) unsigned | 自增主键 | N |  | 是 |
| 2 | role\_name | varchar(20) | 角色名称 | N |  |  |
| 3 | auth\_ids | text | 权限表主键 | Y |  |  |
| 4 | auth\_ac | text | 权限对应的控制器方法集合 | Y |  |  |

4-3 auth后台权限表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 | auth | | | | | |
| 序号 | 字段 | 类型 | 说明 | 空 | 默认值 | 主键 |
| 1 | id | int(10) unsigned | 自增主键 | N | N/A | 是 |
| 2 | auth\_name | varchar(20) | 权限名称 | N | N/A |  |
| 3 | controller | varchar(40) | 权限对应的控制器 | N | N/A |  |
| 4 | action | varchar(20) | 权限对应的方法 | N | N/A |  |
| 5 | pid | tinyint(4) | 权限上一级的id | N | N/A |  |
| 6 | is\_nav | enum('1','2') | 是否具有导航的权限 | N | 1 |  |

4-4 member会员表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 | member | | | | | |
| 序号 | 字段 | 类型 | 说明 | 空 | 默认值 | 主键 |
| 1 | id | int(10) unsigned | 自增主键 | N | N/A | 是 |
| 2 | username | varchar(20) | 用户名 | N | N/A |  |
| 3 | password | varchar(255) | 密码 | N | N/A |  |
| 4 | gender | enum('1','2','3') | 性别 | N | 1 |  |
| 5 | mobile | varchar(11) | 电话 | N | N/A |  |
| 6 | email | varchar(40) | 邮件 | N | N/A |  |
| 7 | avatar | tinyint(4) | 图片地址 | N | N/A |  |
| 8 | created\_at | timestamp | 创建时间 | Y | N/A |  |
| 9 | Update\_at | timestamp | 更新时间 | Y | N/A |  |
| 10 | Remember\_token | varchar(100) | Token值 | Y | N/A |  |
| 11 | type | enum('1','2') | 是否为会员 | N | 1 |  |
| 12 | status | enum('1','2') | 状态 | N | 2 |  |

# 第五章 后台的基础功能开发

基础功能:后台登录，退出，后台首页功能展示。

## 5.1创建管理员数据表

表名：manager。依据数据字典创建数据表；创建迁移文件,利用迁移文件创建数据库，这样更方便。

代码如下：php artisan make:migration create\_manager \_table

执行代码迁移文件的代码生成迁移文件。

创建完迁移文件，打开迁移文件，修改其中的代码，把要生成的数据表字段的代码添加到迁移文件中。

添加的关键代码如下;

$table->increments('id');

$table->string('username',20)->notNull();

$table->string('password')->notNull();

$table->enum('gender',[1,2,3])->notNull()->default('1');

$table->string('mobie',11)->notNull();

$table->string('email',40)->notNull();

$table->tinyInteger('role\_id');

$table->timestamps();

$table->rememberToken();

$table->enum('status',[1,2])->notNUll()->default('2');

代码添加完毕，随后执行迁移文件，创建了manager 表。

代码如下:php artisan migrate: install;php artisan migrate;

## 5.2模拟管理员表的数据

创建完数据表后，由于表中数据是空，所以创建模拟数据来测试网站后台。这样方便后期的测试，也可以模拟用户的数据。要生产模拟数据，得有以下步骤。

### 5.2.1创建填充器文件

代码：php artisan make:seeder Manager TableSeede。

编写填充器的代码文件，因为测试数据需要，所以至少模拟出100条不一样的数据。生产的思路借助循环，faker 代码依赖。faker代码库不需要安装，laravel在创建自身项目的时候已经自带了。使用方法参考：[https://packagist.org/packages/fzniotto/faker](https://packagist.org/packages/fzniotto/faker。)[[11]](https://packagist.org/packages/fzniotto/faker。)[。](https://packagist.org/packages/fzniotto/faker。)

使用方法：1.使用Faker\Factory::create();创建出生成器；

通过实例的对象变量去访问属性来生成测试数据；在laravel 系统中提供了一个用户密码加密的函数，bcrypt,语法bcrypt(‘密码铭文’)。编写生成测试数据代码。

代码如下：

//产生faker实例

$faker=\Faker\Factory::create('zh\_CN');

for($i=0;$i<100;$i++){

//访问具体的属性来获取数据

$date[]=[

'username' =>$faker->userName,

'password' =>bcrypt('123456'),

'gender' =>rand(1,3),

'mobie' =>$faker->phoneNumber,

'email' =>$faker->email,

'role\_id' =>rand(1,6),

'created\_at' =>date('yY-m-d H:i:s',time()),

'status' =>rand(1,2),

];

}

DB::table('manager')->insert($date);

}

### 5.2.2执行填充器代码：

编写完填充器代码之后，得执行填充器文件，因为Laravel文档的特殊性所以执行artisan命令即可：

Artisan的执行代码如下：

Php artisan db:seed --class=ManagerTableSeeder;

执行完命令之后，查看是否生成100条数据。由于100条数据很多。所以执行简化代码。执行代码 select count（\*）from manager。如果显示100则创建成功。

## 5.3后台登录页面展示

一个企业的网站的登录页面非常重要，因为可以筛选掉一些虚假用户，注水用户，登录页面获取用户的信息也可以直接在数据库后台显示，也可以知道用户真实情况，让企业更有针对性发展。如5-1图：

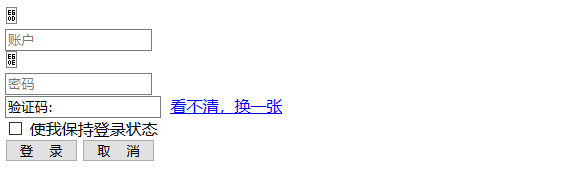


图5-1登录页面

### 5.3.1 确定路由

因为larave框架，也是MVC设计结构，模型，视图，控制器，所以先确定路由，由路由调用控制器。控制器在调用后台数据。首先确定登录的路由，路由路径为：admin/public/login。

1.创建控制器文件

代码：php artian make:controller Admin\Public\Controller。

2.创建需要的login方法

代码：

public function index(){

return view('admin.index.index');

}

3.创建视图文件

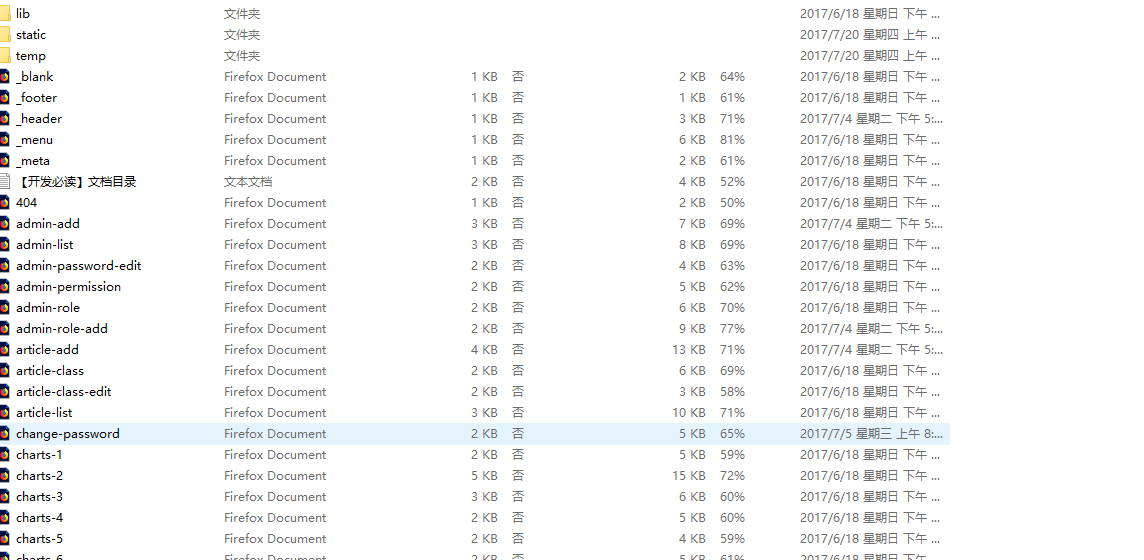
视图可以选择开源的模板，在网站搜索www.h-ui.net/H-ui.admin.shtml网址，打开网址，点击按钮下载H-ui.Admin v3.1.3 版本。下载的得到压缩包，解压压缩包如图5-3。

图5-2 解压的H-ui.Admin v3.1.文件

Lib目录，使用的第三方扩展库。

Static 目录，框架需要使用一些静态文件，如js、css、images等。

.html文件，视图。

将login.html模板文件复制出来，保存到新建文件中，在新建文件中删除一些冗余的代码，保存在相对路径resources/view/admin/login.blade.php 文件中需要的视图创建好了。

## 5.4管理员登录的自动验证

### 5.4.1提交的路由地址

1.路由：admin/public/check,用post方式提交。

2.修改登录表单页面。

修改action值。

修改name值让数据可以提交。

添加csrf 验证。

给“使我保持登录状态”复选框添加一个默认值。

### 5.4.2编写check方法

编写public控制器下的check方法，实现自动验证（验证数据的有效性）。在登录页面显示自动验证没有通过的错误信息。

语法：$this->validate->($require,[规则数组]);

代码如下：

$this->validate($request,[

//验证规则 需要验证的字段名 => '验证规则1'|'验证规则2'|'验证规则3'......

//必填,长度介于2-20

'username' => 'required|min:2|max:20',

//密码,必填,长度6位

'password' => 'required|min:6',

]);

## 5.5管理登录身份验证

在Laravel系统中给用户提供了一个完备认证机制，其使用通过Auth （门面）去实现的。Auth还提供“记住我”的功能。如果要使用“记住我”的功能，可以传递一个布尔值作为第二个参数到attempt 方法，这样用户登录认证状态就一直会保持直到用户手动退出，当然，你的User 表含remember\_token 字段，该字段用于存储“记住我”令牌。

### 5.5.1 guard实例

访问指定 Guard实例，可以使用Auh方法指定想要使用的Guard实例，这种机制允许你在同一个应用中对不同的认证模型或用户表实现完全独立的用户认证。

语法：Auth::guard(指定guard 实例)->attempt(需要验证的用户信息关联数组,[bool 值])；

Guard 实例主要区分前后台的。

在app.php中门面本身已经引入别名，因此在控制器中可以直接使用Auth;先去设定guard(守门员)，在config/auth.php 。

'guards' => [

'web' => [

'driver' => 'session',

'provider' => 'users',

],

'api' => [

'driver' => 'token',

'provider' => 'users',

],

//自己定义后端的guard

'admin' => [

'driver' => 'session',

'provider' => 'admin',

],

],

'admin'=>[

'driver' => 'eloquent',

'model' => App\Admin\Manager::class,

],

],

在App\Http\Controllers\Admin控制器使用Auth类。

### 5.5.2创建需要使用的模型（guard需要使用）

因为laravel模型要实现系统的自动验证，必须创建guard满足模型代码。

创建模型代码如下：

Php artisan make:model Admin/Manager

执行代码成功。生成模型文件成功，文件生成在相对路径app/admin/Manager下。

模型文件的核心代码如下：

protected $table='manager';

修改完模型文件后，可以验证数据登录，输入与数据库数据一致的用户名和密码。但是在验证对的情况下会产生报错。

因为使用laravel内部的验证，所以还得实现一个内部的类，这个类是\Illuminate\Contracts\Auth\Authenticatable。因为这个类一个接口，所以还得实现内部方法，因为模型的自身机制，所以模型自身提供了这个内部方法代码块，在文件\Illuminate\Auth\Authenticatable中。这是一段代码块trait，适用于php 5.4之后的版本，代码块的用法和类相似。所以可以直接引用。

引用方式如下：

namespace App\Admin;

use Illuminate\Database\Eloquent\Model;

//引入trait;

use \Illuminate\Auth\Authenticatable;

class Manager extends Model implements \Illuminate\Contracts\Auth\Authenticatable

{

//定义当前模型需要关联的数据表

protected $table='manager';

//使用trait,就相当于将整个trait代码复制到这个位置

use Authenticatable;

//定义模型与角色模型关联的操作

}

}

## 5.6后台首页的展示

### 5.6.1定义后台首页的路由

由于后台有两个页面组成，所以得约定两个路由。

约定后台路由的路径为：/admin/index/inde、/admin/index/welcome。

约定完路径之后，需要构建相关的控制器。

创建控制器的代码：

Php artisan make:controller Admin/indexController;

### 5.6.2编写index 方法、welcome方法

由于后台首页控制器已经创建，所以编写各自的方法。不同的参数跳转不同的页面，展示不同的视图。

代码如下：

public function index(){

return view('admin.index.index');

}

public function welcome(){

return view('admin.index.welcome');

}

### 5.6.3创建视图文件

由于首页有两个页面，所以得创建两个视图文件（index.blade.php、welcome.blade.php）。在H-ui.Admin v3.1.3 模板文件中找到index、welcome模板文档，保存到相对路径resource/view/admin/index/下,修改掉其中跳转的路径。

后台首页页面如下:

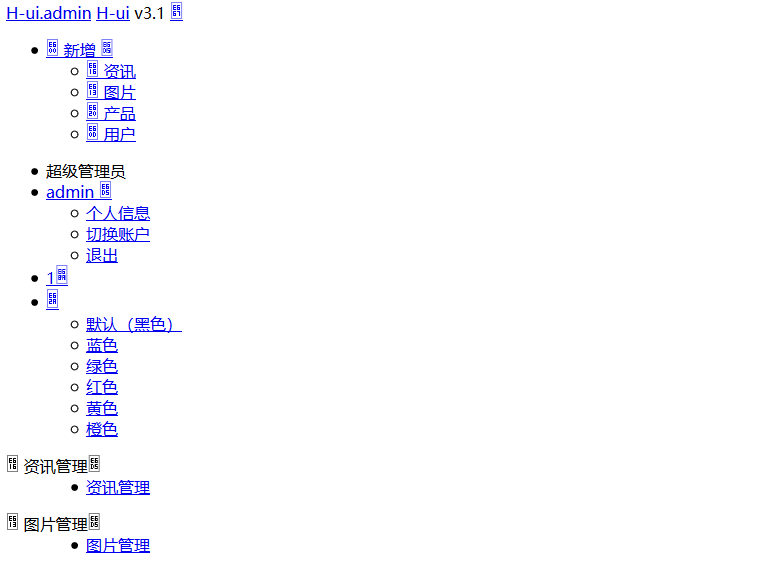


图5-3视图文件

### 5.6.4 动态获取用户登录的用户名

修改视图文件index.blade.php文件。获取认证用户。

语法：Auth：：guard（guard名称）->user()。

<ul class="cl">

<li>{{Auth::guard('admin')->user()->role->role\_name}}</li>

<li class="dropDown dropDown\_hover">

<a href="#" class="dropDown\_A">{{Auth::guard('admin')->user()->username}} <i class="Hui-iconfont">&#xe6d5;</i></a>

<ul class="dropDown-menu menu radius box-shadow">

<li><a href="javascript:;" onClick="myselfinfo()">个人信息</a></li>

<li><a href="#">切换账户</a></li>

<li><a href="../public/logout">退出</a></li></ul>

</ul>

实现退出登录并编写退出方法logout。跳转到首页登录页面。

public function logout(){

Auth::guard('admin')->logout();

return redirect('/admin/public/login');

}

## 5.7路由保护

含义：只有通过认证（符合某个条件）的用户才能访问指定的路由，该方式称之为路由保护。

例如：后台所有页面（除登录页面外）只有通过用户名密码认证才能进行访问，因此需要给对应的页面去做上路由保护。

在laravel框架中解决该类的问题需要使用中间件：

路由 → 中间件 → 目标访问控制器。

中间件不是控制器文件，但是也是一个类。中间件在laravel框架中有2个大类：系统提供的，用户自己定义的。

中间件使用方法：路由中间件可用于只允许通过认证的用户访问给定的路由。Laravel通过定义在app/Http中间件已经在HTTP kernel中定义，需要做的仅仅是将中间件添加到相应的路由定义中。

路由 → middleware(中间件名称)。

需要判断是否登录页面全部加上中间件的保护；在跳转的时候，系统默认使用login路由别名，如果没有则报“路由找不到”的错误，因此需要给登录路由添加“login”别名。

Route::group(['prefix'=>'admin'],function(){

Route::get('public/login','Admin\PublicController@login')->name('login');

});

Route::group(['prefix'=>'admin','middleware'=>['auth:admin']],function(){});

# 第六章 后台内容的完善

## 6.1实现管理员列表的展示

### 6.1.1先定义需要的路径

约定好后台路由路径：admin/manager/。

修改管理员列表的路径。

### 6.1.2创建需要的控制器文件

创建控制器文件的语句：php artisan make:controller Admin/ManagerController。执行此语句，创建控制文件。文件创建好后，需要在控制器中创建需要的index方法（查询列表的数据、展示视图）。

public function index(){

//查询数据

$data=Manager::get();

//展示视图

return view('admin.manager.index',compact('data'));

}

创建完方法后，需要创建展示视图的文件。模板使用的是：admin-list.html 。把模板文件代码创建好，循环数据库中的数据。把数据库中数据展示到列表中。代码如下：

@foreach($data as $val)

<tr class="text-c">

<td><input type="checkbox" value="{{$val->id}}" name=""></td>

<td>{{$val->id}}</td>

<td>{{$val->username}}</td>

<td>{{$val->mobie}}</td>

<td>{{$val->email}}</td>

<td>{{$val->role->role\_name}}</td>

<td>{{$val->created\_at}}</td>

<td class="td-status">

如图6-1管理员列表

核心代码如下：

@if($val->status=='2')

<span class="label label-success radius">已启用</span>

@else

<span class="label radius">已停用</span>

@endif

</td>

<td class="td-manage">

@if($val->status=='2')

<a style="text-decoration:none" onClick="admin\_stop(this,'10001')" href="javascript:;" title="停用"><i class="Hui-iconfont">&#xe631;</i></a>

@else

<a style="text-decoration:none" onClick="admin\_start(this,'10001')" href="javascript:;" title="启用"><i class="Hui-iconfont">&#xe615;</i></a>

@endif

<a title="编辑" href="javascript:;" onclick="admin\_edit('管理员编辑','admin-add.html','1','800','500')" class="ml-5" style="text-decoration:none"><i class="Hui-iconfont">&#xe6df;</i></a>

<a title="删除" href="javascript:;" onclick="admin\_del(this,'1')" class="ml-5" style="text-decoration:none"><i class="Hui-iconfont">&#xe6e2;</i></a></td>

</tr>

@endforeach

## 6.2 RBAC权限管理

### 6.2.1介绍

RBAC介绍：Role-Based 基于角色的访问控制，是一个比较完善的权限访问控制机制，目前有2套权限控制方案，基于用户的权限控制，基于角色的权限控制。

RBAC在后期权限维护上是很方便的，权限的方式在项目开发的初始阶段已经有一个成型的标准。所以在目前很多大的项目后台使用的RBAC权限控制方式。

RBAC方式体现，目前2种方式：三表rbac形式，五表rbac形式，不管是三表还是五表，其原理是一致的。

出现三表（用户表、角色、权限）和五表（用户表、用户与角色的关系表、角色表、角色与权限的关系表）的原因是设计规范的要求（三范式）。

三表：优点是表少，维护方便，理解起来也方便；缺点就是不遵循范式要求；

五表：优点是遵循范式要求；

### 6.2.2数据表的设计

三表：用户表，角色表，权限表。其中用户表已经存在，只需要去创建权限表和角色表即可。

角色表：role。后台权限表：auth。

创建表所需要的迁移文件。

PHP artisan make:migration create\_role\_table.

PHP artisan make:migration create\_auth\_table.

执行完迁移文件代码，然后编写代码：

Role表代码如下：

Schema::create('role', function (Blueprint $table) {

$table->increments('id');//主键

$table->string('role\_name',20)->notNull();//角色名称

$table->text('auth\_ids');//权限id集合

$table->text('auth\_ac');//权限控制器和方法组合的字符串

});

Auth表代码如下：

class CreateAuthTable extends Migration

{

/\*\*

\* Run the migrations.

\*

\* @return void

\*/

public function up()

{

Schema::create('auth', function (Blueprint $table) {

$table->increments('id');

$table->string('auth\_name',20)->notNUll();

$table->string('controller',40);

$table->string('action',20);

$table->tinyInteger('pid');

$table->enum('is\_nav',[1,2])->notNull()->default('1');

});

}

/\*\*

\* Reverse the migrations.

\*

\* @return void

\*/

public function down()

{

Schema::dropIfExists('auth');

}

}

执行迁移文件生成的数据表：

Php artisan migrate。

### 6.2.3理清三表之间的关系

三表：用户表，角色表，权限表。用户表--角色表：manager.role\_id=role.id。角色表--权限表：role.auth\_ids=auth.id。注意：用户表与权限表没有直接的关系。

## 6.3权限管理操作

### 6.3.1展示相关的视图，并添加权限页面，列表页面

A.确定需要的路由。

添加页面：/admin/auth/add any类型。

列表页面：/admin/auth/index get类型。

Route::get('auth/index','Admin\AuthController@index');

Route::any('auth/add','Admin\AuthController@add');

同时需要修改左侧菜单的链接地址，将对应的文字超链接到列表页面。

B.创建需要的控制器文件。

Php artisan make:controller Admin/AuthController。

C.创建2个方法：Auth控制器下的index方法、add方法，展示各自的视图即可。

代码如下：

public function index(){

return view(‘admin.auth.index’);

}

D.创建需要的视图文件

需要：admin.auth.index 和admin.auth.add(默认自己创建)

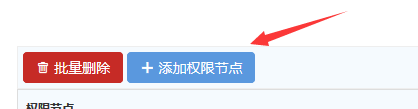
修改其他的添加模板得到添加权限的模板，同时修改其引入文件的路径。最

public function(){

Return view(‘admin.auth.add’);

}

后修改在index.blade.php 文件中对add.blade.php如图6-1文件引入的路径。添加权限也是企业中重要的部分。因为企业各个部门层级不同所以各个权限不同，权限添加就显得尤为重要，最高管理者拥有最多权限。其他的权限随着层级降低，权限也要随之减少。



如图 6-1 添加权限节点

Add.blade.php 页面代码如下：并在页面中添加csrf隐藏域。

编写后台add方法，处理提交过来的数据。先在AuthController控制器下引入Input类。在引入之前得先修改一下app.php配置文件定义input别名，然后修改add方法，将数据写入数据表。在写入之前定义Auth模型文件。

模型文件代码是：php artisan make:model Admin/Auth;

模型文件代码：

namespace App\Admin;

use Illuminate\Database\Eloquent\Model;

class Auth extends Model

{

//

//定义关联数据表

protected $table='auth';

//禁用时间

public $timestamp=false;

}

在控制器引入模型代码。

修改添加页面效果，展示出父级权限选择列表。在页面中展示输出父级权限数据。是权限动态添加。

修改代码如下：

<select class="select" name="pid" size="1">

<option value="0">作为顶级父级权限</option>

@foreach($parents as $val)

<option value="{{$val->id}}">{{$val->auth\_name}}</option>>

@endforeach

</select>

效果如图6-2：

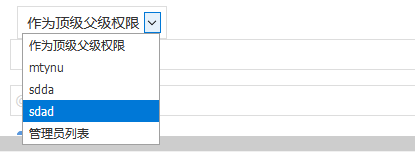


图6-2 权限层级

### 6.3.2实现权限列表功能

查询相关的数据，并且把数据携带到模板中。查询思路：表1：t1;表2: t2。演示sql语句：select t1.\*,t2.auth\_name as parent\_name from auth as t1 left join auth as t2 on t1.pid=t2.id;

将sql语句的代码查询语句，放入php代码中，得到的结果，在页面中展示。

展示在网页的数据页面如图6-3：



图6-3权限列表图

## 6.4角色管理操作

### 6.4.1前置工作处理（数据模拟、模型+控制器创建）

A.先模拟数据。

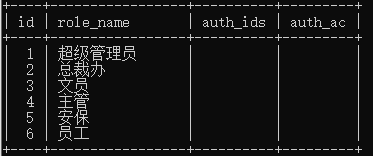


图6-4 role表的数据模拟图

B.创建需要控制器的文件。

Php artisan make:controller Admin/RoleController;

Php artisan make:model Admin/Role;

分别执行上述代码。然后定义模型属性。

C.需要创建后续需要的路由。

角色展示：admin/role/index get类型。

角色权限分派：/admin/role/assign any类型。

核心代码：

Route::get('role/index','Admin\RoleController@index');

Route::any('role/assign','Admin\RoleController@assign');

D.完善前面的管理员列表页面“角色”列的显示补全。

完善方式：关联模型（用户模型关联角色模型）,在视图中（admin/manager/index）使用刚才的关联关系。

同样需要处理类似的还有右上角（admin.index.index）效果图如6-5;



图6-5 角色名称

修改的代码为：

<li>{{Auth::guard('admin')->user()->role->role\_name}}</li>

### 6.4.2实现角色的列表功能

企业中用户表由数据库展示到页面中是及其重要的，这样可以一目了然查看各个层级的权限以及各个用户的层级。所以就得在前端实现用户列表功能以及操作的权限功能。不但要实现表的功能。还的给表排序。

修改后台首页左侧菜单地址，然后创建RoleController下面的index方法，查询数据并展示需要的视图。查询完后创建视图文件，视图文件的模板文件：admin-role.html。引入视图文件后，设计一下表头。然后展示输出数据。

表头设计如图6-6：



图6-6角色管理表图

设计代码如下：

<tr>

<th scope="col" colspan="6">角色管理</th>

</tr>

<tr class="text-c">

<th width="25"><input type="checkbox" value="" name=""></th>

<th width="40">ID</th>

<th width="100">角色名</th>

<th width="150">权限id集合</th>

<th width="150">权限ac集合</th>

<th width="70">操作</th>

</tr>

</thead>

<tbody>

@foreach($data as $val)

<tr class="text-c">

<td><input type="checkbox" value="{{$val->id}}" name=""></td>

<td>{{$val->id}}</td>

<td>{{$val->role\_name}}</td>

<td><a href="#">{{$val->auth\_ids}}</a></td>

<td>{{$val->auth\_ac}}</td>

<td class="f-14"><a title="分派" href="javascript:;" onclick="admin\_role\_assign('分派权限','../../admin/role/assign','{{$val->id}}')" style="text-decoration:none"><i class="Hui-iconfont">&#xe603;</i></a>

<a title="编辑" href="javascript:;" onclick="admin\_role\_edit('角色编辑','admin-role-add.html','1')" style="text-decoration:none"><i class="Hui-iconfont">&#xe6df;</i></a>

<a title="删除" href="javascript:;" onclick="admin\_role\_del(this,'1')" class="ml-5" style="text-decoration:none"><i class="Hui-iconfont">&#xe6e2;</i></a>

</td>

</tr>

@endforeach

</tbody>

### 6.4.3实现数据保存

在开发MVC模式的项目的时候，建议数据的处理和业务逻辑放到M层，当前功能需要对数据进行处理才能入表，所以在控制器中只写流程，具体的处理交给模型处理。

A.创建控制中的流程。

B.创建模型中自定义方法实现数据的处理并且修改到数据表。

C.分派完权限之后页面（列表）并没有刷新最新数据，实现刷新效果。

D.在再次给已经分派过权限的角色进行分派权限的时候，应该需要勾选住已经存在的权限判断依据，在循环的时候，将当前循环的权限id与该角色已经具备的权限id集合进行比较，如果当前权限id在集合中，则认为是有权限的（勾选效果），否则没有权限（不勾选效果）。

先查询当前角色具备的权限id集合。

## 6.5使用中间件进行RBAC的鉴权操作

中间件：中间件介于路由和具体访问操作之间的一个“件”，称之为“中间件”。

创建中间件需要自定义中间件（系统没有自带提供）

创建中间件的代码：php artisan make:middleware 中间件的名称 【中间件不需要区分目录】。

自定义中间件（路由中间件）在使用的时候需要注册，注册在app/Http/Kernel.php。

注册好之后将路由中间件应用到路由中去，之前中间件只有一个所以可以是字符串，但现在之后可能有多个，因此需要转变形式为数组。

编写checkrbac中间件的鉴权代码。

思路：路由 = action =权限。

只需要获取当前其访问的路由，去其角色表中找到拥有的权限，然后比对，如果路由在权限ac中，则其被认为具有权限，允许继续访问；否则没有权限，不让访问。注意：注意特别用户/角色【超级管理员】、注意特殊权限【首页，欢迎页面】。

获取当前访问路由。获取当前用户对应的角色权限ac的信息，通过进行比对权限判断。

## 6.6会员管理

### 6.6.1创建数据表

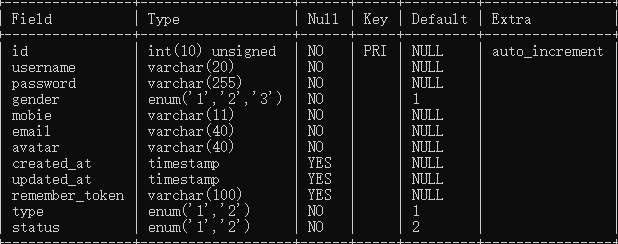


图6-7 会员列表的格式

创建迁移文件：php artisan make:migration create\_member\_table。

编写迁移文件的代码生成数据表，核心代码：

public function up()

{

Schema::create('member', function (Blueprint $table) {

$table->increments('id');

$table->string('username',20)->notNull();

$table->string('password')->notNull();

$table->enum('gender',[1,2,3])->notNull()->default('1');

$table->string('mobie',11);

$table->string('email',40);

$table->string('avatar',40);

$table->timestamps();

$table->rememberToken();

$table->enum('type',[1,2])->notNUll()->default('1');

$table->enum('status',[1,2])->notNUll()->default('2');

});

}

执行迁移文件；

Php artisan migrate。

### 6.6.2建立测试数据

使用faker 代码依赖生成测试数据500条，在生成数据之前，得创建填充器文件。

创建填充器文件的代码：php artisan make:seeder MemberTableSeeder。

编写填充器代码使用faker生成假数据。核心代码：

public function run()

{

//

$faker=\Faker\Factory::create('zh\_CN');

for($i=0;$i<500;$i++){

//访问具体的属性来获取数据

$data[]=[

'username' =>$faker->userName,

'password' =>bcrypt('123456'),

'gender' =>rand(1,3),

'mobie' =>$faker->phoneNumber,

'email' =>$faker->email,

'avatar' =>'/statics/avatar.jpg',

'created\_at' =>date('yY-m-d H:i:s',time()),

'type' =>rand(1,2),

'status' =>rand(1,2),

];

}

DB::table('member')->insert($data);

}

执行填充器文件：PHP artisan db:seed --class=MemberTableSeeder。

### 6.6.3路由与控制器、模型的创建

路由：会员列表页面：admin/member/index get类型。

会员的添加页面：/admin/member/add any类型。

异步头像上传地址： /admin/uploader/webuploader post类型。

Route::get('member/index','Admin\MemberController@index');

Route::any('member/add','Admin\MemberController@add');

Route::post('uploader/webuploader','Admin\UploaderController@webuploader');//异步上传

Route::any('member/getareabyid','Admin\MemberController@getAreaById');//ajax联动

创建命令：php artisan make:controller Admin/MemberController。

创建命令：php artisan make:model Admin/Member。

定义关联的数据表“manager”。

修改当前地址的左侧的菜单链接如图6-8。

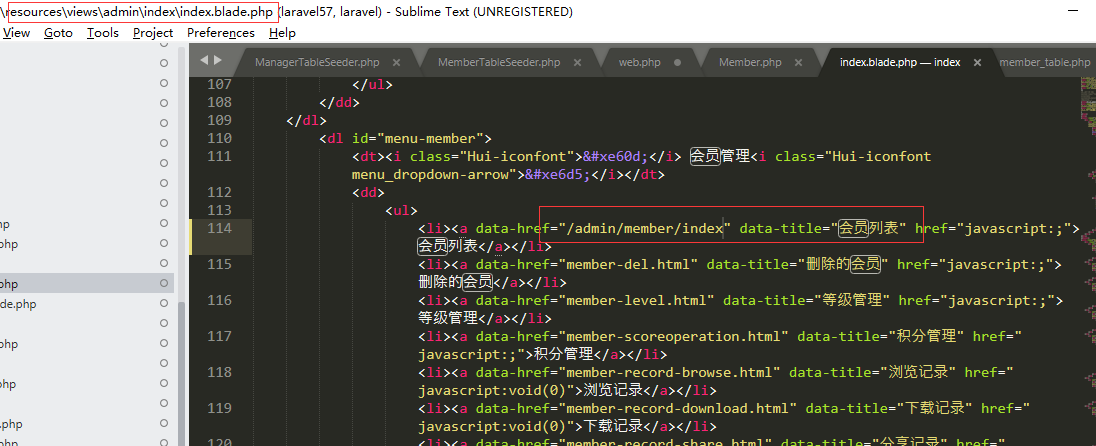


图6-8 地址修改

### 6.6.4实现列表展示

在MemberController创建index 方法，获取数据，展示视图。

创建需要的视图文件index.blade.php。

循环输出所需要的数字，展示效果如图6-9：



图6-9 用户列表图

核心代码：

@foreach($data as $val)

<tr class="text-c">

<td><input type="checkbox" value="{{$val->id}}" name=""></td>

<td>{{$val->id}}</td>

<td>{{$val->username}}</td>

<td><img src="../..{{$val->avatar}}" width="50" height="80"></td>

<td>@if($val->gender == '1') 男 @elseif($val->gender == '2')女 @else 保密 @endif</td>

<td>{{$val->mobie}}</td>

<td>{{$val->email}}</td>

<td>{{$val->created\_at}}</td>

<td class="td-status">

@if($val->type == '1')

管理

@else

客户

@endif

</td>

@if($val->status == '1')

<span class="label label-danger radius">已停用</span>

@else

<span class="label label-success radius">已启用</span>

@endif

</td>

<td class="td-status">

@if($val->type == '1')

管理

@else

客户

@endif

</td>

### 6.6.5实现会员的添加功能

修改“会员列表”页面左上角的“添加”按钮的链接，将其修改为之前的“admin/member/add”的路由。按钮的效果图如6-10。



图6-10 按钮

修改此处的代码：

<div class="cl pd-5 bg-1 bk-gray mt-20"> <span class="l"><a href="javascript:;" onclick="datadel()" class="btn btn-danger radius"><i class="Hui-iconfont">&#xe6e2;</i> 批量删除</a> <a href="javascript:;" onclick="member\_add('添加用户','../../admin/member/add','','510')" class="btn btn-primary radius"><i class="Hui-iconfont">&#xe600;</i> 添加用户</a></span> <span class="r">共有数据：<strong>88</strong> 条</span> </div>

编写add方法，实现表单页面的展示。代码如下：

public function add(){

//判段请求

if(Input::method() == "POST"){

$result=Member::insert([

'username' =>Input::get('username'),

'password' =>bcrypt('password'),

'gender' =>Input::get('gender'),

'mobie' =>Input::get('mobie'),

'email' =>Input::get('email'),

'avatar' =>Input::get('avatar'),

'type' =>Input::get('type'),

'status' =>Input::get('status'),

'created\_at'=>date('Y-m-d H:i:s')

]);

return $result ?1:0;

}else{

return view('admin.member.add');

}

}

创建需要的视图文件add.blade.php。

# 第七章 登录页面的设计

## 7.1登录页面设计

### 7.1.1登录页面前端效果图

因为直接由后端代码展示的界面非常不美观，所以不需借助js、css代码来美化网页界面。这样可以提升用户的体验，也不会显得网页的页面的单薄、贫乏。所以得借助前端的代码提升前端的用户界面。这样也能突显出企业网站的水准，和态度。

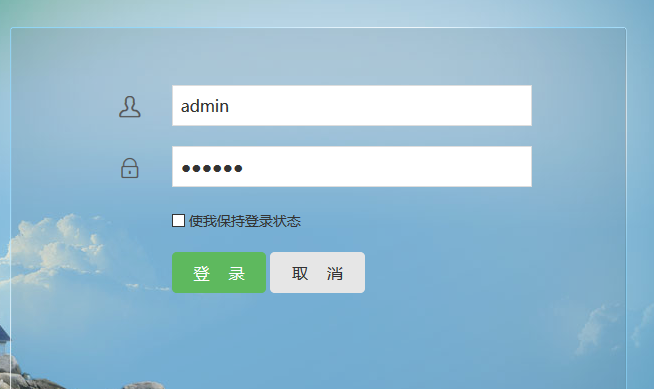


图7-1 登录界面

因为加载了H-ui.Admin v3.1.3 版本，所以设计前端登录页面，只需引入H-ui.Admin v3.1.3 版本自带的js和css 文件。在视图文件login中外部文件的引入路径，引入后效果如图7-1。

引入css 文件：

<link href="../../../public/admin/static/h-ui/css/H-ui.min.css" rel="stylesheet" type="text/css" />

<link href="../../../public/admin/static/h-ui.admin/css/H-ui.login.css" rel="stylesheet" type="text/css" />

<link href="../../../public/admin/static/h-ui.admin/css/style.css" rel="stylesheet" type="text/css" />

<link href="../../../public/admin/lib/Hui-iconfont/1.0.8/iconfont.css" rel="stylesheet" type="text/css" />

引入js:

<script type="text/javascript" src="../../../public/admin/lib/jquery/1.9.1/jquery.min.js"></script>

<script type="text/javascript" src="../../../public/admin/static/h-ui/js/H-ui.min.js"></script>

<script type="text/javascript" src="../../../public/admin/lib/layer/2.4/layer.js"></script>

后面所有的视图文件引入的js、css文件目录路径/static/h-ui/css/H-ui.min.css、lib/Hui-iconfont/1.0.8/iconfont.css、lib/jquery/1.9.1/jquery.min.jsstatic、/h-ui/js/H-ui.min.js等全部开头地址全部替加上../../../public/admin/。

### 7.1.2登录页面勾选的设计

效果7-2如图：



图7-2 登录状态效果图

### 7.1.3在登录页面上显示错误信息

由于害怕一些恶意的用户注册网站，所以我们必须设置一个错误的提示信息。这样也可以排除一些恶意的用户，也可以个用户一定得提示作用，错误信息显示效果图7-3 ：

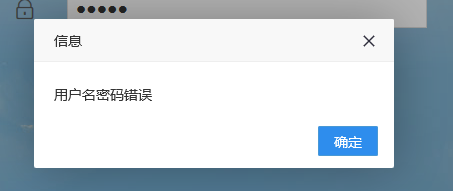


图7-3 错误信息显示图

在当前页面如果使用alert 会很丑，建议使用JavaScript插件layer进行提示。使用方式：先引入layer.js文件，并且在jQuery之后引入；再去引入later。

前端代码：

<script type="text/javascript">

$(function(){

@if(count($errors)>0)

var allError='';

@foreach($errors->all() as $error)

allError+="{{$error}}";

@endforeach

//输出错误信息

layer.alert(allError,{title:'错误提示',icon:2});

@endif

});

</script>

### 7.1.4后台首页的设计

效果如图7-4：



图7-4首页界面

引入的css代码：

<link rel="stylesheet" type="text/css" href="../../../public/admin/static/h-ui/css/H-ui.min.css" />

<link rel="stylesheet" type="text/css" href="../../../public/admin/static/h-ui.admin/css/H-ui.admin.css" />

<link rel="stylesheet" type="text/css" href="../../../public/admin/lib/Hui-iconfont/1.0.8/iconfont.css" />

<link rel="stylesheet" type="text/css" href="../../../public/admin/static/h-ui.admin/skin/default/skin.css" id="skin" />

<link rel="stylesheet" type="text/css" href="../../../public/admin/static/h-ui.admin/css/style.css" />

引入的js代码：

<script type="text/javascript" src="../../../public/admin/lib/jquery/1.9.1/jquery.min.js"></script>

<script type="text/javascript" src="../../../public/admin/lib/layer/2.4/layer.js"></script>

<script type="text/javascript" src="../../../public/admin/static/h-ui/js/H-ui.min.js"></script>

<script type="text/javascript" src="../../../public/admin/static/h-ui.admin/js/H-ui.admin.js"></script> <!--/\_footer 作为公共模版分离出去-->

<!--请在下方写此页面业务相关的脚本-->

<script type="text/javascript" src="../../../public/admin/lib/jquery.contextmenu/jquery.contextmenu.r2.js"></script>

### 7.1.5实现datatables列表无刷新分页

实现页面视图文件admin/manager/index.blade.php数据分页。

效果如图7-5：



图7-5 管理员数据分页图

Datatables插件是一款基于jQuery框架进行开发的无刷新分页插件，其除了分页还有排序、搜索等功能。

官网:http://www.datatables.net[12]。

使用客户端分页式方式实现分页。

使用步骤：先在确保引入jQuery之后，然后引入datatables的JavaScript文件；需要初始化datatables插件；

代码如下：

<script type="text/javascript" src="../../../public/admin/lib/laypage/1.2/laypage.js"></script>

<script type="text/javascript">

$('table').dataTable({

//禁用掉第一列排序

"aoColumnDefs":[{"bSortable":false,"aTargets":[0]}],

//默认在初始化的时候按照那一列进行排序

"aaSorting":[[1,"asc"]],

//禁用搜索

"searching":false,

});

</script>

## 7.2 ajax异步请求

### 7.2.1ajax的数据提交

视图文件admin/auth/add.blade.php 数据提交到控制器，需要异步请求，所以需要ajax提交请求，效果如7-6。

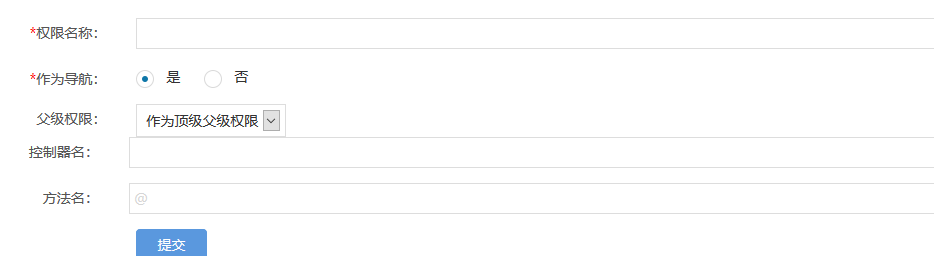


图7-6 数据请求图

Ajax的代码如下：

submitHandler:function(form){

$(form).ajaxSubmit({

type: 'post',

url: "" ,

success: function(data){

if(data!=0){

layer.msg('添加成功!',{icon:1,time:1500},function(){

var index = parent.layer.getFrameIndex(window.name);

parent.$('.btn-refresh').click();

parent.layer.close(index);

});

}else{

layer.msg('添加失败!',{icon:2,time:1500});

}

},

error: function(XmlHttpRequest, textis\_nav, errorThrown){

layer.msg('error!',{icon:1,time:1500});

}

});

}

### 7.2.2实现头像**异**步上传

异步： ajax 需要借助于ajax 来实现目标效果（无刷新）。

由于异步的上传代码篇幅较大，也不好去调试，纯自己手写是很大的挑战，一般都会使用一些开源成型的Jquery 插件。

插件：webuploader(基于Jquery开发)；

官网：<http://fex.baidu.com/webuploader/>[13]。

下载地址：<https://github.com/fex-team/webuploader/releases>[14]。

下载完后，需要引入资源。将整个主要使用的webuploader目录复制到当前的站点目录public/statics下。

在视图(view/admin/member/add.blade.php)中引入资源。

引入css

<link rel="stylesheet" type="text/css" href="../../statics/webuploader-0.1.5/webuploader.css">

引入js

<script type="text/javascript" src="../../statics/webuploader-0.1.5/webuploader.js"></script>

使用图片上传功能：

先引入html代码（创建选择文件的按钮，创建存放文件的信息的容器）。

<div id="uploader-demo">

<!--用来存放item-->

<div id="fileList" class="uploader-list">

<!-- 添加隐藏域 -->

<input type="hidden" name="avatar" value="">

</div>

<div id="filePicker">选择图片</div>

</div>

</div>

B.实例化工作：webuploader(初始化操作)。

在案例中官网提供比较完善的js源码文件，可以参考。<http://fex.baidu.com/webuploader/js/getting-started.js>[15]找到这个文件中针对“图片上传demo”的部分，将其复制出来，粘贴到新建立的一个javaScript文件中，在视图里面做一个引入即可。avatar.js就是复制出来的JavaScript文件。保存到相对地址public/statics/中。

C.将avatar.js文件在视图中引入。

<script type="text/javascript" src="../../statics/avatar.js"></script>

D.根据实际情况修改avatar.js文件。

swf: 'statics/webuplosder-0.1.5/js/Uploader.swf',

// 文件接收服务端。

server: '../../admin/uploader/webuploader',

E.根据路由信息，创建对应的控制器和方法去处理异步上传的请求。

异步文件上传的流程说明：

产生异步的请求。

请求有一个响应的方法（需要返回值，返回的内容是图片的路径；在此不需要进行数据交互操作）。

当异步请求获取到响应后，需要存储图片的地址（但是又没有必要给用户看，因此可以使用隐藏域的方式进行存储）。

当用户提交表单的时候顺带提交隐藏域的地址。

创建控制器

Php artisan make:controller Admin/UploaderController。

创建所需要的方法webuploader。

解决csrf的验证问题。

由于异步提交和表单不是同一次请求，所以csrftoken值不能被异步的ajax使用。需要单独给ajax异步添加csrf的token值。

针对webuploader 在上传的时候使用其他参数的说明，参考地址：

<http://github.com/fex-team/webuploader/issues/145>[16]。

在框架中如果引入的外部文件，需要注意在外部文件中不能使用模板引擎技术的内容，否则不解析。

G.实现上传处理方法。

在laravel 项目中除了在上传的时候可以使用move和store方法【都是request实例封装的】之外，还可以使用高级的上传文件保存方法（有storage门面提供的）。

Storage门面在使用的时候需要注意的实项：

I:在storage门面中，有一个“磁盘”的概念，“磁盘”一共有三种：本地磁盘、公共磁盘、第三方磁盘；“磁盘”实质上就是一个目录；

II:本地磁盘（local磁盘），存放在local磁盘中的文件是不能被浏览器访问到的，存储在其中的文件一般都是项目的php代码自身使用；【磁盘位置位于/storage/app】

III:公共磁盘（public磁盘），存放在该磁盘中的文件一般是允许浏览器进行访问的；【磁盘的位置位于/Stroage/app/public】

IV:第三方磁盘：驱动形式的一种，允许开发者将文件上传到第三方服务器上；

默认的情况下，浏览器是无法访问到public 磁盘的目录，但是其又想让服务器访问，所以需要做一“软链接”的操作。

软链接的代码：Php artisan storage :link。

Storage门面通用文件保存方法；storage::disk(磁盘名)->put(文件名，文件的内容)。

H.异步请求回调处理。

添加隐藏域，用于存放地址，编写回调代码。

uploader.on( 'uploadSuccess', function( file,response ) {

$( '#'+file.id ).addClass('upload-state-done');

//$('input[name=avatar]').val(response.path);

console.log(response.path);

});

数据入表，在MemberController控制器修改代码。

public function add(){

//判段请求

if(Input::method() == "POST"){

$result=Member::insert([

'username' =>Input::get('username'),

'password' =>bcrypt('password'),

'gender' =>Input::get('gender'),

'mobie' =>Input::get('mobie'),

'email' =>Input::get('email'),

'avatar' =>Input::get('avatar'),

'type' =>Input::get('type'),

'status' =>Input::get('status'),

'created\_at'=>date('Y-m-d H:i:s')

]);

return $result ?1:0;

}else{

return view('admin.member.add');

}

}

# 第八章 本文总结与展望

## 8.1总结

本文主要通过代码制做对企业网站的建设和研究，进行了设计。对于当前一些企业的不完善的需求做出了一些改进，比如：对数据的分页，前端的分页，和一些js代码事件的完善。采用laravel作为其基础技术框架;采用MySQL作为数据库管理工具。JavaScript，css作为前端设计工具。建立了4张基本数据库表，对4张表进行详细介绍;系统服务端采用基于php语言编写，采用开源软件Apache作为网站服务器。根据系统的功能需求，设计了登录模块、管理员模块、权限模块、角色模块、会员模块。全站共计5个功能模块，并对功能模块的关键算法进行了说明。

我们也结合当下客户的习惯对网站进行设计，也运用了一些设计模式，以便网站进行更新和维护。

## 8.2展望

(1)因为时间比较仓促，网站开发周期较短，网站的界面还不够美观，色彩搭配还并不理想，页面布局还不够合理。

(2)网站的安全性是一个比较严重的问题，因为没有验证码验证，所以以后会有客户恶意攻击网站。用脚本攻击网站，以致会影响后台数据。

(3) 考虑到互联网技术的突飞猛进，以后会有更新的技术实现网站，这套技术可能存在代码的冗余，也可能会影响后台的服务器运行。

(4) 随着用户数量的增多，数据交互增加，会加重数据库的负担从而增加企业服务器的负担，所以要增加缓存技术，缓存技术可以大大的减少数据库的负担，同时也能减少服务器的负担。增加企业网站的数据交互效率。

由于论文是本人独自完成，所以一定存在技术缺陷和思维的不完善。

参考**文献：**

[1]比尔.斯科特. WEB界面设计[M].电子工业出版社.2009

[2]史蒂夫.克鲁克. Don`t make me think [M].New Riders Press 2005

[3]彼得-莫维尔. WEB信息架构[M]，电子工业出版社，2008-8

[4]Wang,XH.The suggestions for the website construction based on The customers` consuming mentalstate[C].liaoning Tech Univ. ,Fuxin, PEOPLES

RCHINA:3rd International Symposium on ModernMining and Safety Technology ,2008

1. L. Crawford, J. Pollack,D. Engl and. Uncovering the trends in project management: Journal emphases over the last 10 years[J]. International Journal of Project Management,2006,24:175-184

[6]Ali, A.S.B., Money, W.H. A Study of Project Management System Acceptance[C]. System Sci ences, Hawaii,USA,2005:234-244

[7]王凤玲. 基于PHP+MYSQL的新闻发布系统的研究与实现[J]，计算机应用与软件，2012

[8] 赵鹤芹. 设计动态网站的最佳方案:Apache+PHP+MySQL[M]. 计算机工程与设计，2007

[9] GuptaS, K aiser G, NeistadtD, et al.  DOM-based content extraction of HTML documents Intemational Conference on World W ide Web. ACM, 2003: 207-214

[10]朱频，程治平. 基于PHP技术网站的建设[J]. 甘肃冶金，2008，30(2).

[11][https://packagist.org/packages/fzniotto/faker](https://packagist.org/packages/fzniotto/faker。)[EB/OL]

[12]http://www.datatables.net[EB/OL]

[13]<http://fex.baidu.com/webuploader/>[EB/OL]

[14]<https://github.com/fex-team/webuploader/releases>[EB/OL]

[15]<http://fex.baidu.com/webuploader/js/getting-started.js>[EB/OL]

[16]<http://github.com/fex-team/webuploader/issues/145>[EB/OL]

致谢

在大学求学生涯中，虽然过程很行苦，但是在老师和同学的鼓励支持下，在学业和知识技能等方面有了极大的提升。

我的导师，张小红教授，在我写论文期间，他的一丝不苟的精神让我的论文有了极大的进展。他不但在学业上给我帮助，生活中也会给予我关心。在此，我向我的导师致以敬意和谢意。他也引导着我其它的方面，在思想和待人接物方面，教导着我保持谦逊之心。

同时，我还要感谢我的母亲。在我人生成长的途中一直激励着我前行，在学习上遇到困难，也是鼓励我的态度。养育之恩难于回报，我也只有做好当下的事情就是对你们最大回报。在这里也感谢身边的朋友，是你们给予了我极大的帮助。