海 南 大 学

毕 业 论 文（设计）

题 目：基于PHP的企业网站的开发与实现

学 号： 20130902320105

姓 名： 裴清溪

年 级： 2013级

学 院： 海南大学信息科学技术学院

系 别： 计算机系

专 业： 网络工程专业

指导教师：

完成日期： 年 月 日

**摘 要**

21世纪，已经进入了互联网飞速发展的时代，不管是事业单位，政府部门，学校，企业都不可避免的需要一个属于自己的门户网站，网站可以让人们更加方便的了解一个公司的企业文化，最新的新闻资讯、优惠信息、产品信息以及获取顾客对公司的反馈信息等，由于PHP开发的周期短，成本低，后期便于维护，并且开源项目丰富，目前大部分的WEB项目都是用PHP作为开发语言进行开发。

本文通过PHP技术开发一个小型的企业网站，阐述PHP开发WEB应用的相关技术。本文中开发的网站搭载在LNMP环境，服务器软件为Nginx，安装在Linux操作系统，并运行PHP7.0和MySql5.7数据库。本论文的主要研究方向是企业网站的需求分析，并在此基础上进行网站的系统详细设计，并使用PHP实现系统以及测试。

**关键字：**php、企业网站、mysql

**Abstract**

窗体顶端

窗体顶端

The 21st century has entered the era of rapid development of the Internet, whether institutions, government departments, schools, enterprises are inevitable need to own a portal, the site can make people more convenient to understand a company's corporate culture, the latest News, information, product information and access to customer feedback on the company, due to php fast, short cycle, low cost, low maintenance costs, open source products such as rich features, more and more companies choose php web Development.

This paper expatiates and discusses the web development technology based on php development by realizing the development of a small enterprise website. The site is designed with PHP + MySQl + Nginx framework, web server for Nginx for user requests, web page design based on PHP scripting language, MySQl as database storage network data. The main research direction of the design is the design of the background database of the enterprise website and the design of the whole framework. First, the basic front page display is realized. Secondly, the database and the background frame are designed and produced. Finally, the test is carried out.

The enterprise website is divided into two sections, the front page of the main display, including the home page, the company profile, product display, industry information, customer message, contact us six modules, background design,including administrator login, the latest announcement management, product management , Message management, contact us management, use phalcon framework to achieve the code, the database design using Medoo engine support.

窗体底端

Keywords: php, corporate website, mysqlKeywords: php, corporate website, mysqlKeywords: php, corporate website, mysqlKeywords: php, corporate website, mysql窗体底端

**Keywords**: php, corporate website, mysql

窗体底端

**目录**

**[摘 要](#_Toc26709)** [1](#_Toc26709)

**[Abstract](#_Toc32411)** [2](#_Toc32411)

**[1，引言](#_Toc12247)** [4](#_Toc12247)

**[1.1，背景](#_Toc7278)** [4](#_Toc7278)

**[1.2，研究的目的和意义](#_Toc119)** [4](#_Toc119)

**[2，相关技术](#_Toc5854)** [5](#_Toc5854)

**[2.1，B/S模式及其优势](#_Toc3155)** [5](#_Toc3155)

**[2.2，PHP后台语言](#_Toc16743)** [6](#_Toc16743)

**[2.3，MVC模式简介](#_Toc25470)** [6](#_Toc25470)

**[2.4，Phalcon框架介绍](#_Toc29748)** [7](#_Toc29748)

**[2.5，nginx服务器](#_Toc27730)** [8](#_Toc27730)

**[3，系统需求分析](#_Toc6050)** [8](#_Toc6050)

**[3.1开发环境介绍](#_Toc25443)** [8](#_Toc25443)

**[3.2可行性分析](#_Toc15614)** [8](#_Toc15614)

**[3.3 企业网站总体需求](#_Toc18373)** [9](#_Toc18373)

**[3.3.1 企业网站前台功能](#_Toc13851)** [9](#_Toc13851)

**[3.3.2 企业网站后台功能](#_Toc32454)** [10](#_Toc32454)

**[3.4 系统功能模块分析](#_Toc27315)** [11](#_Toc27315)

**[3.4.1管理员管理模块](#_Toc4690)** [12](#_Toc4690)

**[3.4.2 公告管理模块](#_Toc25008)** [13](#_Toc25008)

**[3.4.3 资讯管理模块](#_Toc18327)** [14](#_Toc18327)

**[3.4.4 留言管理模块](#_Toc26058)** [15](#_Toc26058)

**[3.4.5产品管理模块](#_Toc30630)** [15](#_Toc30630)

**[3.4.6公司简介模块和联系我们模块](#_Toc10280)** [16](#_Toc10280)

**[3.4.7首页模块](#_Toc8727)** [16](#_Toc8727)

**[3.5 企业网站的非功能性需求](#_Toc6363)** [17](#_Toc6363)

**[4，企业网站的系统设计](#_Toc12811)** [17](#_Toc12811)

**[4.1，系统总体架构](#_Toc11379)** [17](#_Toc11379)

**[4.2，系统各功能模块的设计](#_Toc28671)** [17](#_Toc28671)

**[4.2.1，后台管理功能模块](#_Toc16129)** [17](#_Toc16129)

**[4.2.1，管理员管理模块](#_Toc12093)** [18](#_Toc12093)

**[4.3，系统流程图设计](#_Toc2099)** [18](#_Toc2099)

**[4.3.1，管理员流程](#_Toc31742)** [18](#_Toc31742)

**[4.3.2，用户流程](#_Toc29058)** [19](#_Toc29058)

**[4.4系统数据库设计](#_Toc30585)** [20](#_Toc30585)

**[4.5系统详细数据表](#_Toc26236)** [24](#_Toc26236)

**[5,企业网站系统功能模块的实现。](#_Toc14697)** [27](#_Toc14697)

**[6，总结](#_Toc14514)** [33](#_Toc14514)

**[致谢](#_Toc2776)** [35](#_Toc2776)

**1，引言**

**1.1，背景**

当今社会是互联网蓬勃发展的信息化社会，各个企业为了展示企业的风采，提升企业的知名度，介绍企业的主要业务和产品，开发一款企业的官方网站必不可少。企业网站的建设在互联网应用上的地位十分重要，它已经成为政府部门，企业以及事业单位在信息化建设中不可缺少的一部分。因此，企业官方网站的建设备受企业管理层的重视。在这个信息化的社会，企业的官方网站可以说是企业在互联网上的一面窗户，人们想要了解一个企业的基本信息，最重要的渠道就是去该企业的官方网站上浏览和查阅相关信息，这些信息是最有参考价值的；相应的，一个企业向大众展现企业的风采、技术成果、研究领域、主要产品等，最好的途径就是在该企业的官方网站上发布消息。

PHP作为一门面向WEB开发的脚本语言，其应用在网站、系统等开发领域相比于Java、C++等其他的语言有着得天独厚的优势：

其一，PHP的设计初衷就是面向WEB端开发的，它集成了C/C++、Java、Perl等语言的优点，并且是基于CGI的脚本语言，并且提供了相当丰富的系统API，开发者可以很轻松地使用PHP开发WEB程序。

其二，PHP性能稳定而且消耗系统资源少，运行速度快。目前PHP7.0已经发布，本文所开发的网站是基于PHP7.0开发，其运行效率更是比PHP5.0快了一倍左右。使用PHP开发的网站或者系统有很强的稳定性和可维护性，能大大减少后期系统维护所需要的成本。

**1.2，研究的目的和意义**

本文的意义所在就是企业建立了一个信息网站，让用户和合作伙伴知道企业的重要信息以及合作价值。在电子商务和网络经济发展迅猛的今天，越来越多的企业认识到建立一个信息网站的重要性，通过建立企业信息网站，可以扩大知名度。企业网站就好像是窗口，通过这个窗口，你可以了解到这个企业的最新情况以及过去的情况，同时通过这个窗口也可以与外界进行商务方面的谈判和交流，作为一个企业的重要做成部分，一个企业信息网站的成功与否，关系到一个企业的发展和效率。

企业网站的具体作用：

(1)，企业通过企业官方网站可以展示企业的风采与文化，树立企业良好的形象，并提高企业的知名度；

(2), 通过对企业的产品介绍，让经销商更加了解企业，增加销量；

(3), 通过宣传公司的产品，企业文化，像经销商展示公司的风采，让经销商直接走进你，选择你；

(4)， 可以通过留言板，获取更多的建议，不断的改进自己。

开发一款针对建材公司的基于php的企业网站，囊括公司简介，最新公告，产品管理，联系我们，管理员登陆等功能，能够使管理员在后台系统中对登陆用户，产品，公告等进行修改的很方便的企业网站。

**2，相关技术**

**2.1，B/S模式及其优势**

B/S结构是浏览器/服务器模式，是结合多种脚本语言和ActiveX技术在不断成熟的Web浏览器技术的基础上用浏览器实现原本需要复杂的专用软件才能实现的功能，不仅节约成本，而且可以在任何地方进行操作，不需要安装任何专门的软件，仅仅需要一台可以上网的电脑就可以使用，客户端零安装、零维护，极大的减少了开发成本和使用的难度，系统的扩展特别容易。随着该模式的使用越来越多，也极大的推动了AJAX技术的发展，能够在客户端电脑上进行部分处理这一优势极大的减轻了服务器的负担，使得可以局部刷新，增加了交互性，B/S的结构图如下：

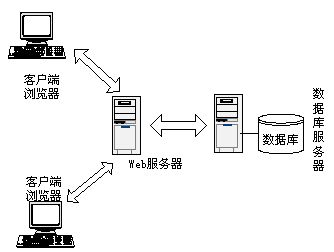


图2.3B/S结构图

**2.2，PHP后台语言**

**2.3，MVC模式简介**

MVC英文全称Model View Controller，即模型、视图和控制器模式。M为业务模型，V为视图，C为控制器。模型主要是对系统数据的封装，视图主要是系统的显示层，通常是用户界面。在传统的使用MVC的目的是将M和V实现代码分离，使得一个程序可以有不同的表现形式，C的目的是使M和V同步，当M改变时，V也跟着同步更新。MVC耦合性低，因为模型和业务模型、视图分离，所以很容易改变数据层和业务规则。由于运用MVC的三个部件是相互独立的，所以改变其中的任何一个，都不会影响其他两个，利用这种思想能构造良好的低耦合性构建，该模型生命周期成本低，可重用性高，部署快，可维护性高，通过以下的图对MVC的框架进行解释：

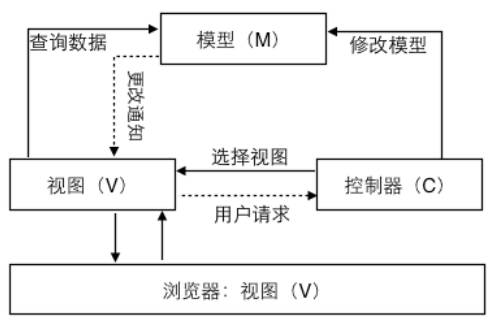


图2.1MVC在web中的应用中的结构

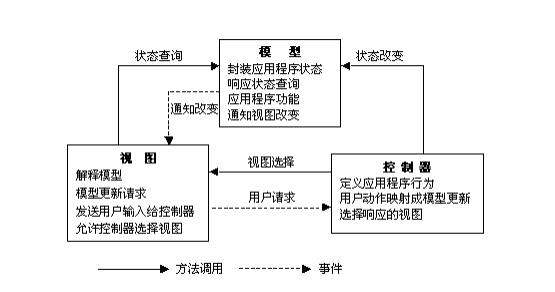


图2.2MVC组件类型的关系和功能

**2.4，Phalcon框架介绍**

Phalcon是一种高性能的PHP框架，其底层是用C语言实现的，所以性能十分良好。Phalcon是针对高性能优化的PHP框架，其设计上基于MVC模式，能够实现应用数据模型和页面视图的完美解耦。其次，Phalcon也实现了依赖注入，能够减少各个模块之间的耦合性。由于Phalon的底层实现语言是C语言，其他的PHP框架，如ThinkPHP、CodeIgniter等的运行速度远远不及Phalcon。下图是常用PHP框架的性能对比柱状图：

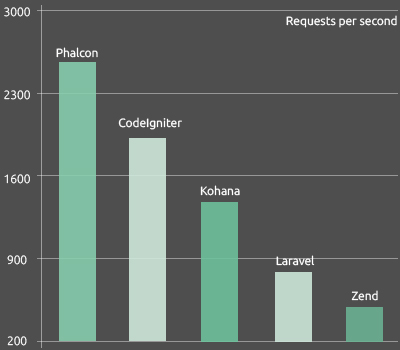
****

图1-1 PHP框架性能对比

**2.5，nginx服务器**

**3，系统需求分析**

**3.1开发环境介绍**

在基于php开发的企业网站时，该网站开发时使用的软件开发环境如下：

(1)，服务端

操作系统：Ubuntu 16.0

应用服务器：nginx 3.0

PHP版本：php7.0

数据库：MySQl5.7

mysql图形化管理软件：Navicat  premium 11.0

作图工具：Visio 2010

浏览器：IIE浏览器6.0以上版本

(2)，客户端

浏览器：IIE浏览器6.0以上版本

**3.2可行性分析**

(1)，经济可行性

利用PHP开发的企业网站。可以大大减少人力资源的成本，降低宣传本公司的产品及公司文化带来的宣传费用，而且通过网络可以扩大宣传力度，网络已经成为人们生活不可或缺的一部分，通过企业网站可以更加便捷的获取用户对本公司的建议，便于统计客户的需求，吸引更多的合作伙伴，同时使用php开发中小型的企业网站，便于管理和维护，极大的降低了这方面的成本。

(2)，运行可行性

这款基于PHP开发的企业网站，对于客户端只需要有一个超过6.0版本的IE浏览器，对于服务端来说这是一个相对独立的系统，只需要完成与操作系统的信息交互，对其他的系统并没有什么影响，所以在使用时出现运行冲突的概率几乎为零。

(3)，技术可行性

开发一个基于php的企业网站最重要的是，实现对网站上发布信息的及时更新，以及网站留言的回复等，这些使用ajax技术可以实现。对于企业网站的稳定性而言，由于使用mvc框架和php语言的特点，相对比较稳定。

(4)，法律可行性

该系统是一个企业网站，主要的功能在于发布信息，以及收集信息，并不需要用户登录，不会出现用户注册个人信息，发生信息泄露的可能性，而且企业后台系统的登陆只有后台管理员可以登录。

**3.3 企业网站总体需求**

Internet以人们始料不及的方式飞速发展，已经成为人们日常获取信息，沟通交流必不可少的渠道之一。在这样的大环境下，建立企业私有网站，可以极大的方便客户获取该企业的企业文化，产品信息等，同时简单的操作，简洁的服务，能够方便企业的管理。

经过某企业的需求分析和资料整理，该企业网站主要分为两个部分：前台部分由首页，公司简介，产品展示，行业资讯，客户留言，联系我们六个部分组成，后台部分由首页，公司简介，产品管理，联系我们，留言管理，管理员六个部分组成。其功能结构图如下：

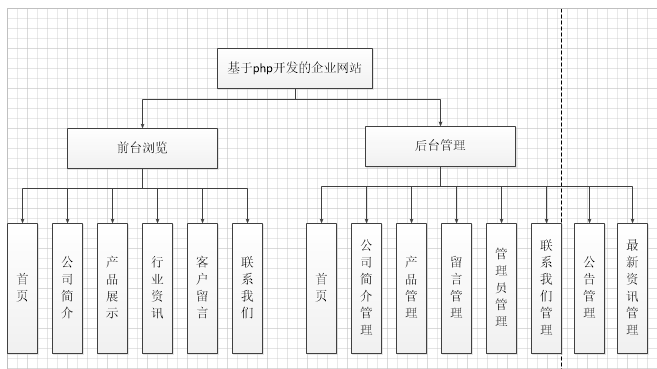


图3-1系统结构图

**3.3.1 企业网站前台功能**

1. , 首页：该页面包含公司的图片，部分公司简介，产品展示，最新公告，友情链接和行业咨询，点击更多是向下面的页面进行跳转，点击单项链接时会跳转到详情页。
2. ，公司简介：公司介绍的显示，侧边栏的我要留言的显示与跳转，最新公告和友情链接的显示，以及联系我们的底边栏的显示。
3. ，产品展示：产品名称以及图片的显示，以及分类和分页功能的实现，联系我们的底边栏的显示。
4. ，行业资讯：行业资讯的题目和更新时间的显示，以及分页功能的实现，点击会有详情页，最新公告和友情链接的显示，以及联系我们的底边栏的显示。
5. ，客户留言：客户的昵称和留言信息框的显示，提交后，成功提示框的跳出，最新公告和友情链接的显示，以及联系我们的底边栏的显示。
6. ，联系我们：公司的信息显示，联系人，地址，联系电话，以及公司位置的定位的显示，最新公告和友情链接的显示，以及联系我们的底边栏的显示。

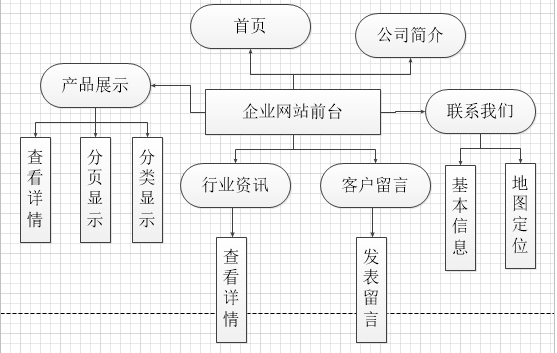


图3-2前台页面系统结构图

**3.3.2 企业网站后台功能**

1. ，首页：首页管理主要包括统计信息、系统信息、版权信息三个部分，其中统计信息对该系统的产品总数、文章总数、留言总数进行统计，系统信息主要包括该系统开发的主机名、操作系统、服务器软件、数据库平台的开发信息，版权信息顾名思义就是指该系统的版权所有、程序版本，技术知识。
2. ，公司简介：可以重新填写公司简介，并进行相应的修改。
3. ，产品管理：从后台可以添加产品的名称、产品的图片，并且可以对已经添加的产品进行编辑，修改，删除等操作。
4. ，公告管理；可以添加新的文章，也可以都已有的文章进行修改删除等操作，同时可以进行批量操作，及时删除一些很陈旧的信息。
5. ，最新资讯管理：可以添加新的文章，及时发布最新的行业资讯，公司的最新动态，也可以进行批量操作，删除一些过时的新闻资讯。
6. ，留言管理：可以获取客户的留言进行查看，对公司的一些制度进行修改，同时通过客户的留言，对本公司的一些产品的漏洞就行改造提升。
7. ，联系我们信息管理：对于存在于页面底部的联系我们，以及联系我们这一个页面的信息进行修改。
8. ，管理员管理：可以添加管理员的账号和密码，只有公司内部人员可以进入后台系统，同时也可以对于已经存在的账户和密码进行修改，同时可以启用和停用该账户，该模块还包括管理员登陆页面，输入用户名，密码，以及验证码与数据库中的信息进行核对，进行登陆。

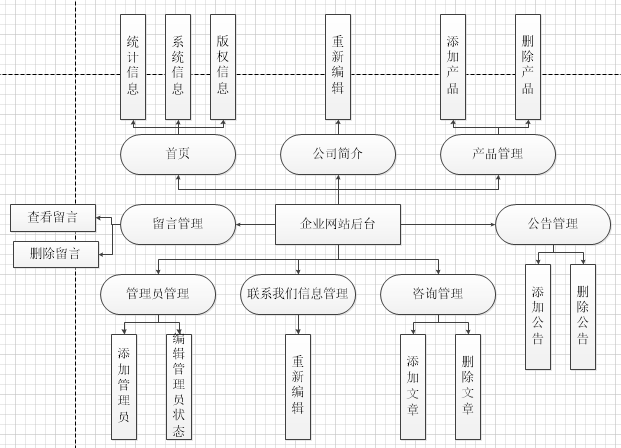


图3-3企业网站后台系统结构图

**3.4 系统功能模块分析**

企业网站包括前台模块和后台模块两个模块，其中后台的功能模块是本次研究的重点模块。该模块主要包括留言管理，公告管理，资讯管理，联系我们信息管理，管理员管理，产品管理，首页，公司简介八个模块。

**3.4.1管理员管理模块**

管理员管理模块是这个系统的核心模块之一，本系统不存在用户登录，由于该系统为建材公司的网站，主要目的是让用户了解公司的基本信息，产品内容，以及最新动态等内容，并不是B2C的商务网站，目的并不是网上销售，所以不需要用户登录，只需要管理员对网站上的信息进行及时的维护和更新就可以了。所以只有管理员登陆这个功能模块。

管理员可以添加管理员，修改管理员的密码，同时可以编辑管理员的账号的状态是启用还是停用。

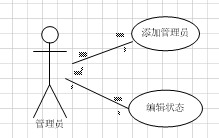


图3-4 管理员管理用例图

表3-1管理员管理用例分析

|  |  |
| --- | --- |
| 用例名称 | 管理员管理 |
| 用例Id | 001 |
| 角色 | 管理员 |
| 用例描述 | 管理员必须先登录后太管理系统，然后单击管理员登陆，可以添加管理员账号，同时可以对其他的管理账户状态进行编辑。 |
| 前置条件 | 管理员必须先登陆后台管理系统 |
| 后置条件 | 管理员修改信息 |

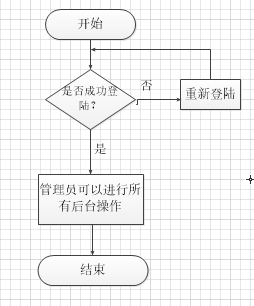


图3-5 管理员模块活动图

**3.4.2 公告管理模块**

公告管理模块的针对的主要是管理员，管理员可以在后台对公告进行新增发布，删除，修改等操作，同时该模块可以进行批量操作，极大的提高了管理的效率。

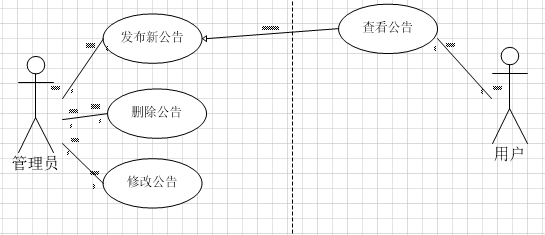


图3-6公告管理用例图

表3-2公告管理用分析

|  |  |
| --- | --- |
| 用例名称 | 管理员管理 |
| 用例Id | 002 |
| 角色 | 管理员 |
| 用例描述 | 管理员必须先登录后台管理系统，然后单击公告管理，可以对已有的公告进行修改跟删除操作，也可以添加新的公告，这个模块可以批量操作 |
| 前置条件 | 管理员必须先登陆后台管理系统 |
| 后置条件 | 跳转到前台的最新公告界面 |

**3.4.3 资讯管理模块**

资讯管理的操作对象也主要是管理员，管理员登陆后台管理系统后，可以发布最新的新闻公告，展示行业内的最新消息，同时可以对已经过时的新闻资讯进行删除，可以批量操作，也可以对已经发布的咨询进行修改。当修改结束之后会直接跳转到前台的最新资讯模块。

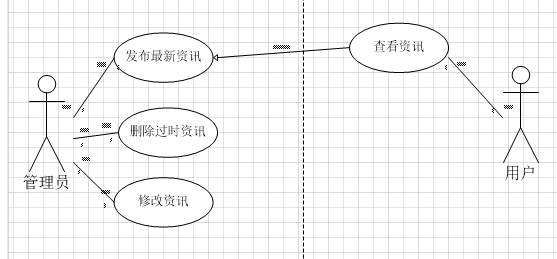


图3-7资讯管理用例图

表3-3资讯管理用分析

|  |  |
| --- | --- |
| 用例名称 | 管理员管理 |
| 用例Id | 003 |
| 角色 | 管理员 |
| 用例描述 | 管理员必须先登录后台管理系统，然后单击资讯管理，可以发布最新的新闻公告，同时可以对已经过时的新闻资讯进行删除，可以批量操作，也可以对已经发布的咨询进行修改。 |
| 前置条件 | 管理员必须先登陆后台管理系统 |
| 后置条件 | 跳转到前台的新闻资讯界面 |

**3.4.4 留言管理模块**

留言管理的操作人也是管理员，管理员对用户发布的留言进行操作。在这个模块我们只能看到用户的昵称和留言，本模块主要用于对公司的管理模式，服务态度等方面收集用户的意见的作用，所以并没有回复这个功能，只对留言进行删除和查看的操作，对于没有意义的意见进行删除，对有意义的意见进行整理。

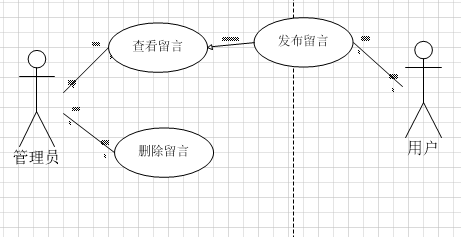


图3-8，留言管理用例图

表3-4留言管理用分析

|  |  |
| --- | --- |
| 用例名称 | 管理员管理 |
| 用例Id | 004 |
| 角色 | 管理员 |
| 用例描述 | 管理员必须先登录后台管理系统，然后单击留言管理，可以查看用户对本公司的反馈留言，同时可以对没有意义的留言进行删除。 |
| 前置条件 | 管理员必须先登陆后台管理系统 |
| 后置条件 | 后台查看用户留言 |

**3.4.5产品管理模块**

产品管理模块是由管理员上传最新的产品信息，包括图片，名称，详细信息。也可以对已经存在的产品进行信息维护，对于升级的产品或销量很少的产品进行删除操作，可批量操作。

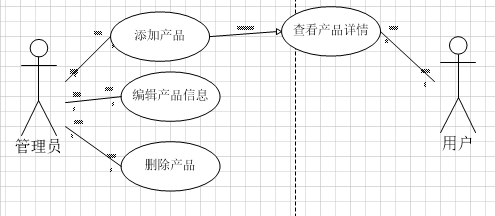


图3-9产品管理用例图

表3-5留言管理用分析

|  |  |
| --- | --- |
| 用例名称 | 管理员管理 |
| 用例Id | 005 |
| 角色 | 管理员 |
| 用例描述 | 管理员先登录后台管理系统，可以发布产品的图片，名称，以及详情信息，对已经发布的产品信息进行修改，对于已经使用量很少的产品可以进行删除。 |
| 前置条件 | 管理员必须先登陆后台管理系统 |
| 后置条件 | 跳转到前台产品信息页 |

**3.4.6公司简介模块和联系我们模块**

公司简介模块和联系我们模块的原理是一样的，都是管理员在登陆后台系统之后，可以点击公司简介模块（联系我们）后，可以对公司的简介信息（联系我们中的公司信息）进行重新发布。

**3.4.7首页模块**

首页模块主要包括三个部分，即统计信息、系统信息以及版权信息，在用户登录后台管理系统之后，可以很直观的看到该系统的发布文章总数、产品总数、留言总数、系统信息以及版权信息，是对整个系统的信息的一个汇总。

**3.5 企业网站的非功能性需求**

在这个企业网站的系统功能模块中，非功能性的需求也十分重要。什么叫做非功能性需求呢？就是说并不是系统在执行某个特定操作的时候的需求，而是只在某些特定条件下判断系统的运作情形或者特性。现在我们就本网站的非功能性需求进行分析：

1. ，实用性。由于社会发展的需求，企业网站已经是一个公司发展壮大比不可少的东西。
2. ，可操作性。该系统简单，易操作，功能明确。
3. ，扩展性。本网站基于php+MySQl+nginx进行web开发，本身niginx就是为php开发的低耦合性的框架，又是有mvc模式进行开发，易于维护和扩展。
4. ，安全性。本系统没有用户登录的操作，极大的保证了用户的个人信息不会被泄露，而且用户只有查看和发布留言的权限，后台的管理操作只有管理员才可以操作，所以该系统安全性可以保证。
5. ，可行性和安全性。该系统运行稳定，拥有一定的防护措施。

**4，企业网站的系统设计**

**4.1，系统总体架构**

该系统主要包含访问层、控制层、应用层、数据层四层，浏览器的终端在访问层负责和用户进行交互，获取到用户的输入信息提交给控制层；控制层负责对用户权限的判定，然后向应用层发送请求；应用层接收到来自控制层的y用户的请求参数后，向数据层发送该请求的数据调用的申请，然后接收数据层返回的数据，然后将数据返回到前台界面上显示；数据层是通过接收到用户层的请求之后，调用和操作数据库来完成数据的控制，并向应用层返回其申请的相应的数据。

**4.2，系统各功能模块的设计**

**4.2.1，后台管理功能模块**

后台管理模块包括产品管理模块，公告管理模块，资讯管理模块，留言管理模块，公司简介管理模块，联系我们信息管理模块，其在时序操作上具有相同内在性，所以时序图如下图所示：

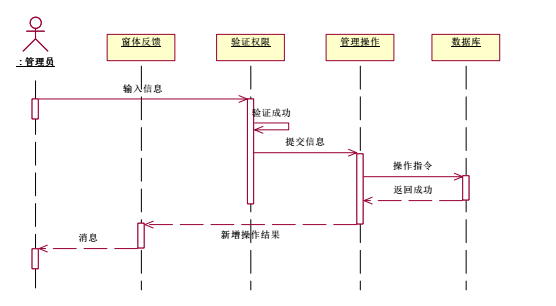


图4-1后台管理时序图

**4.2.1，管理员管理模块**

管理员管理是在管理员登陆后台系统之后进行的操作。协作图如下图所示：

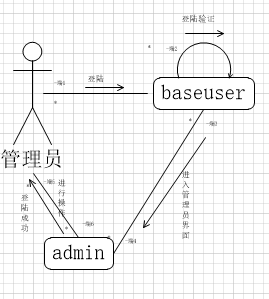


图4-2管理员管理协作图

**4.3，系统流程图设计**

**4.3.1，管理员流程**

管理员首先要进行登陆，登陆成功之后进入主界面即首页，可以查看系统文章数量、公告数量、留言数量，然后可以点击侧边栏，公司简介模块可以对公司的简介信息进行修改，重新提交；产品管理模块可以添加新产品，也可以对已有的产品进行编辑，同时可以对过时的产品进行批量的删除操作；资讯管理和公告管理可以对过时的公告或者资讯进行删除，也可以发布新的公告和资讯。留言管理可以对用户的留言进行查看和删除操作；联系我们模块可以对联系我们这一栏目里面的信息进行编辑，重新发布。

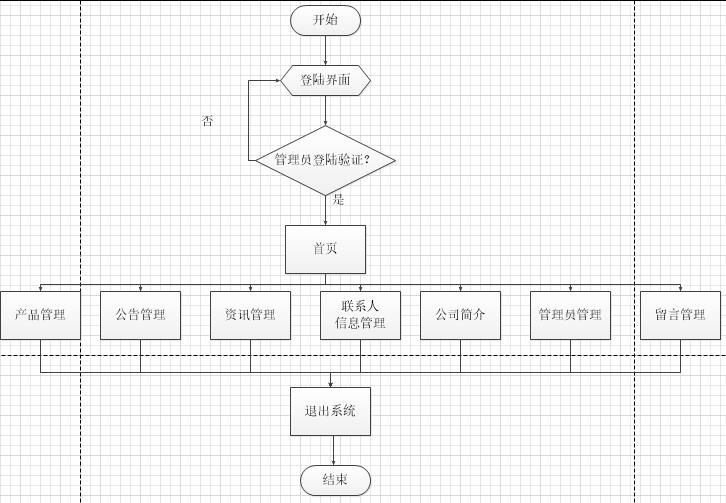


图4-3管理员流程图

**4.3.2，用户流程**

用户首先进入首页页面包含公司的图片，部分公司简介，产品展示，最新公告，友情链接和行业咨询，点击更多是向下面的页面进行跳转，点击单项链接时会跳转到详情页。其中公司简介，产品展示，行业资讯，友情链接，联系我们等页面用户可以查看，点击可查看详情页，客户留言页面，用户可以在页面进行留言，提交到数据库。

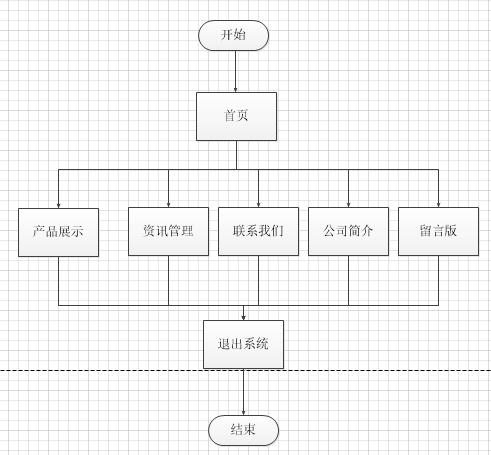


图4-4用户流程图

**4.4系统数据库设计**

数据库设计是一个系统设计中至关重要的一个部分，它关系到一个系统的成败。一个好的数据库设计是一个系统后续开发的基础，也可以提高整个系统的性能。如果基础没有打好，很有可能会导致整个系统的重建。

**4.4.1系统的ER图**

企业网站系统中的实体主要包括：管理员、产品表、留言表、资讯信息表、公告信息表、公司信息表、友情链接表、文章表等。

管理员表主要包括管理员ID，管理员用户名，管理员登陆密码，管理员注册IP，管理员账号状态，账号注册时间，其ER图如图所示：

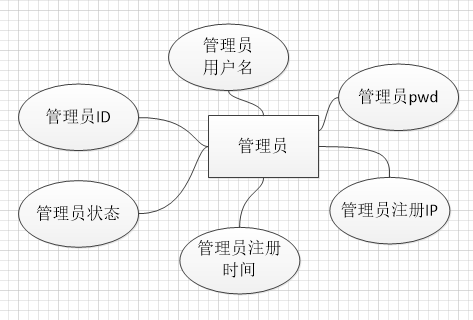


图4-5管理员ER图

产品表主要包括产品ID，产品类别，添加人ID，产品描述，产品名称，图片路径，添加时间，其ER图如下图所示。

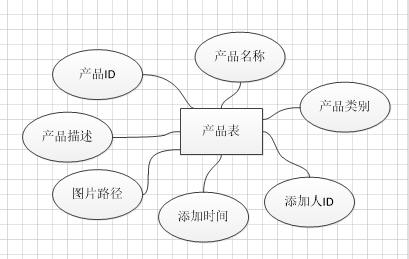


图4-6产品表ER图

留言表主要包括留言ID，留言内容，留言昵称，发布时间，其ER图如下图所示。

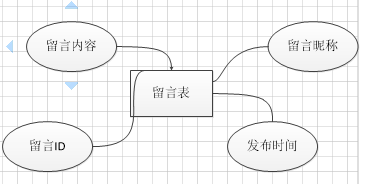


图4-7留言表ER图

资讯表包括资讯ID、发布人ID，资讯标题，资讯内容，发布时间，其ER图如下图所示。

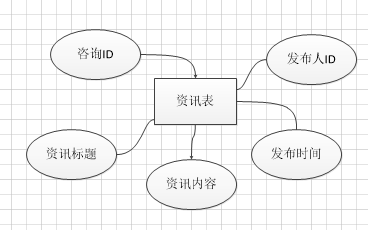


图4-8资讯表ER图

版权表主要包括版权信息ID、版权所有人ID、版权号、程序版本、技术支持，其ER图如下图所示。

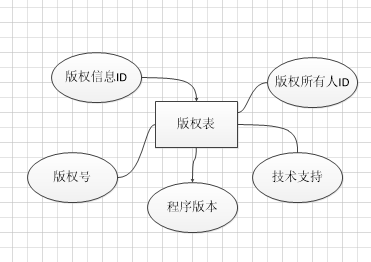


图4-9版权表ER图

联系我们表主要包括记录ID，用户ID，内容，添加时间，其ER图如下。

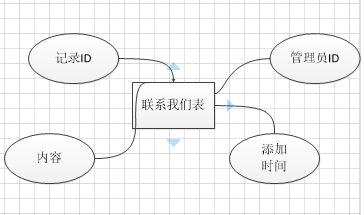


图4-10联系我们表ER图

公司信息表主要包括公司信息ID，公司名称，公司简介，公司地址，联系人姓名，移动电话，固定电话，传真，版权ID，TCP备案号，更新用户ID，添加时间，其ER图如下图所示。

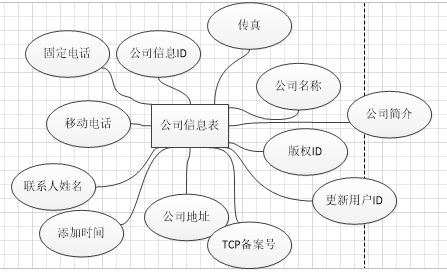


图4-11公司信息表ER图

友情链接表主要包括友情链接ID，页面ID，友情链接名称，友情链接内容，其ER图如下图所示：

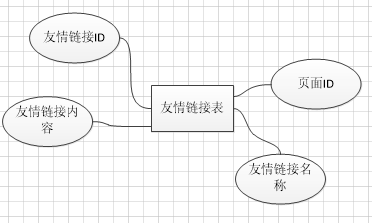


图4-12友情链接表ER图

文章表主要包括文章ID，文章内容，文章类型，文章标题，发布人ID，发布时间，其ER图如下图所示。

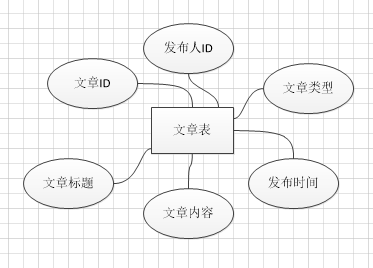


图4-13文章表ER图

通过对各实体的ER图进行分析，得出企业网站的总ER图，如下图所示。

**总系统ER图没有画出来**

**4.5，系统详细数据表**

基于PHP开发企业网站的数据库设计使用的是MySQl5.7，该系统的数据库中有如下几个表：管理员表，留言表，文章表，资讯表，友情链接表，公司信息表，联系我们表，产品表。具体的表设计如下所示。

1. ，管理员表(admin)

表4-1管理员表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段大小 | 允许null值 | 说明 |
| Id | int | 11 | 否 | 管理员ID |
| name | varchar | 30 | 否 | 管理员用户名 |
| password | varchar | 80 | 否 | 管理员密码 |
| register\_time | int | 11 | 否 | 注册时间 |
| Register\_ip | varchar | 20 | 否 | 注册的IP |
| User\_state | tinyint | 1 | 否 | 用户状态 |

(2)，产品表

表4-2产品表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段大小 | 允许null值 | 说明 |
| id | int | 11 | 否 | 产品ID |
| Product\_type | tinyint | 2 | 否 | 产品类型 |
| User\_id | int | 11 | 否 | 添加人ID |
| name | varchar | 30 | 否 | 产品名称 |
| desn | text |  | 否 | 产品描述 |
| Pic\_path | varchar | 255 | 否 | 图片路径 |
| Add\_time | int | 11 | 否 | 添加时间 |

(3),留言表

表4-3留言表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段大小 | 允许null值 | 说明 |
| Id | int | 11 | 否 | 留言ID |
| nickname | varchar | 30 | 否 | 昵称 |
| content | text |  | 否 | 留言内容 |
| Pub\_time | int | 11 | 否 | 发布时间 |

(4),资讯表

表4-4咨询表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段大小 | 允许null值 | 说明 |
| Id | int | 11 | 否 | 资讯ID |
| Pub\_uid | int | 11 | 否 | 发布人ID |
| title | varchar | 80 | 否 | 资讯标题 |
| content | text |  | 否 | 资讯内容 |
| Pub\_time | int | 11 | 否 | 发布时间 |

(5),版权表

表4-5版权表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段大小 | 允许null值 | 说明 |
| Id | int | 11 | 否 | 版权信息ID |
| cpn\_id | int | 11 | 否 | 版权所有人ID |
| Copyright\_no | varchar | 30 | 否 | 版权号 |
| Program\_version | varchar | 80 | 否 | 程序版本 |
| Tech\_spt | varchar | 80 | 否 | 技术支持 |

(6),联系我们表

表4-6联系我们表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段大小 | 允许null值 | 说明 |
| Id | int | 11 | 否 | 联系我们记录ID |
| User\_id | int | 11 | 否 | 用户ID |
| content | text |  | 否 | 内容 |
| Add\_time | int | 11 | 否 | 添加时间 |

(7)，公司信息表

表4-7公司信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段大小 | 允许null值 | 说明 |
| Id | int | 11 | 否 | 公司信息id |
| cpn\_name | varchar | 255 | 否 | 公司名称 |
| Cpn\_intro | text |  | 否 | 公司简介 |
| addr | varchar | 30 | 否 | 地址 |
| contacts | varchar | 30 | 否 | 联系人姓名 |
| mobile | varchar | 11 | 否 | 移动电话 |
| fixed\_tel | varchar | 12 | 否 | 固定电话 |
| fax | varchar | 12 | 否 | 传真 |
| Copyright\_id | int | 11 | 否 | 版权id |
| Icp\_no | varchar | 30 | 否 | ICP备案号 |
| User\_id | int | 11 | 否 | 更新用户ID |
| Add\_time | int | 11 | 否 | 添加时间 |

(8),友情链接表

表4-8友情链接表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段大小 | 允许null值 | 说明 |
| Id | int | 11 | 否 | 友情链接id |
| page\_id | int | 11 | 否 | 页面id |
| name | varchar | 30 | 否 | 友情链接名称 |
| content | text |  | 否 | 友情链接内容 |

(9),文章表

表4-9文章表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段大小 | 允许null值 | 说明 |
| Id | int | 10 | 否 | 文章id |
| Article\_type | tinyint | 1 | 否 | 文章类型 |
| Pub\_uid | int | 10 | 否 | 发布人id |
| title | varchar | 10 | 否 | 文章标题 |
| content | text |  | 否 | 文章内容 |
| Pub\_time | int | 10 | 否 | 发布时间 |

**5系统文件目录结构与MVC架构**

## 5.1系统代码组织结构设计

由于本文开发的企业网站分为前台网页和后台管理系统两部分组成，并且该系统基于Phalcon框架开发的，在开始实现系统之前，确定了系统的工程名为jinlingPhalcon。由于Phalcon是一个基于MVC架构的框架，故网站系统在架构上也是依据M(Model)V(View)C(Controller)3层架构体系来开发的。MVC在设计上将系统分为数据模型（Model）、视图（View）和控制器（Controller）3层，视图是与用户进行交互的，在系统中一个视图对应一个页面。视图需要的数据都是通过数据模型（Model）来获取的，但是为了提高程序的可扩展性，框架将视图与数据模型解耦，视图层通过控制器来与数据模型进行交互。系统的包结构如表5-1所示：

**表5-1 企业网站项目包目录结构表**

|  |  |
| --- | --- |
| 包名 | 说明 |
| app/config | 系统的配置文件都在该目录下。这些配置文件主要定义了系统主要目录的加载器，系统的路由规则、数据库等等配置 |
| app/controllers | 系统所有的控制器都放在这个目录下面 |
| app/library | 该目录下主要是一些工具类，比如Cookie类和Session类、Medoo数据库引擎、字符串处理类等等。 |
| app/models | **系统的所有数据模型，一张表对应一个Model，每个model都有相应的增删改查的函数** |
| app/plugins | **系统需要的两个插件，一个用于处理系统找不到指定文件的异常，一个用于安全验证** |
| app/runtime | **系统运行时，主要是一些日志文件，系统运行的日志都会输出到这个目录，方便查看** |
| app/services | 系统的服务类存放在这个目录 |
| app/tasks | **系统需要执行的自动化脚本存放在这个目录** |
| app/views | 系统的视图，app/views目录下面的子目录对应控制器名称，子目录下面的视图文件对应控制器中的Action函数。 |
| public | 工程挂载的目录，所有的请求会先进入public目录，当public目录下找不到资源时会将请求分发给app目录下面的控制器，项目的js、css代码也存放在public下面的子目录中 |

系统的包结构如图5-1所示：

图5-1网站项目的包结构

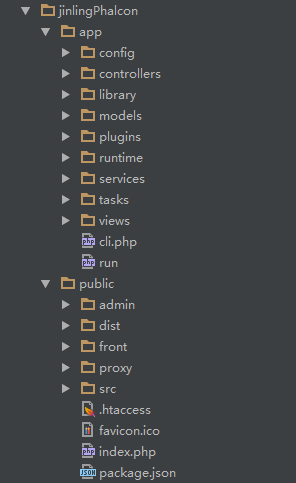


图5-1系统的包结构

# 6企业网站系统实现

## 6.1企业网站后台管理系统实现

### 6.1.1管理员登陆

在进入后台管理系统时，程序会判断用户是否已经登录，当用户未登录时，会自动跳转到登录页面进行登录。登录页面的视图是/admin/public/login.phtml。在登录页启动和点击验证码图片时，都会请求接口获取验证码图片，用户输入用户名、密码和验证码后，点击“登录”按钮，程序先会判断用户名、密码是否合法，当合法时，程序通过POST请求的方式将数据发送到controllers/admin/PublicController.php的loginAction函数，并通过调用模型JinLingAdmin.php中的相关函数验证用户名与密码的正确性。企业网站的后台管理员登录页如下图所示：



图5-1后台管理员登录

主要的代码如下：

1. 页面视图login.phtml

<form id="loginform" method="post" action="">

<table class="mainbox">

<tbody>

<tr>

<td class="loginbox">

<img src="/admin/images/login.gif">

</td>

<td class="login">

<div class="errormsg loginmsg"><p>用户名不能为空！</p></div>

<table class="login\_tab">

<tr>

<td>用户名:</td>

<td><input type="text" id="username" tabindex="1" class="txt" name="username"/></td>

</tr>

<tr>

<td>密　码:</td>

<td><input type="password" value="" id="password" tabindex="2" class="txt" name="password"/></td>

</tr>

<tr>

<td>验证码:</td>

<td><input type="text" value="" tabindex="3" class="txt" name="captcha" style="width:70px;"/> <img id="imgcode" width="85px" height="41px" src="/admin/public/captcha?v=564e9b9d57282" onclick="this.src='/admin/public/captcha?&v='+Math.random()" title="看不清楚，点击换一张"/></td>

</tr>

<tr>

<td></td>

<td><input type="submit" tabindex="4" class="btn" value="登 录" name="submit"/></td>

</tr>

</table>

</td>

</tr>

</tbody>

</table>

</form>

1. 控制器PublicController.php

public function loginAction()

{

if ($this->request->isPost()) {

$userName = $this->request->getPost('userName', 'string');

$userPassword = $this->request->getPost('userPassword', 'string');

$captcha = $this->request->getPost('captcha', 'string');

$acaptcha = new \MyApp\Library\XCaptcha;

$res = $acaptcha->checkCode($captcha);

if (empty($userName)) {

$this->view->accountMsg = '请输入用户名！';

} elseif (empty($userPassword)) {

$this->view->passwordMsg = '请填写登录密码！';

} elseif (empty($captcha)) {

$this->view->captchaMsg = '请填写验证码！';

} elseif (!$res) {

$this->view->captchaMsg = '验证码不正确！';

} else {

$adminM = new JinLingAdmin();

$adminService = new AdminService();

$utilities = new Utilities();

$ip = $utilities->getClientIP();

$data = $adminService->isAdminUser($userName);

$check = $adminService->isAdminValid($userName, $userPassword);

if($data === null){

$this->view->errorMsg = '用户不存在！';

$adminService->traceLog(['catalog' => 'login', 'intro' => '登录失败，用户不存在:' . json\_encode($userName), 'userId' => 0, 'createTime' => time(), 'url' => '/admin/public/login', 'ip' => $ip]);

}elseif(isset($data['user\_state']) && $data['user\_state'] == 0){

$this->view->errorMsg = '用户未启用！';

}elseif(isset($check['isAdmin']) && !$check['isAdmin'] == true){

$this->view->errorMsg = '用户名或密码错误！';

$adminService->traceLog(['catalog' => 'login\_pass\_error', 'intro' => '登录失败，密码不正确:' . json\_encode($userName), 'userId' => 0, 'createTime' => time(), 'url' => '/admin/public/login', 'ip' => $ip]);

}else{

$this->session->set('\_adminiData', $check['info']);

$this->session->set('\_adminiUserId', $check['info']['id']);

$this->session->set('\_adminiUserName', $check['info']['userName']);

$id = $check['info']['id'];

$where = ['id' => $id];

$updateData = ['lastLoginIp' => $utilities->getClientIP(), 'lastLoginTime' => time(), 'loginCount' => $check['info']['loginCount'] + 1];

$adminM->update($updateData, $where);

//$this->view->disable();

$adminService->traceLog(['catalog' => 'login', 'userId' => $id, 'intro' => '用户登录成功' . json\_encode($userName), 'createTime' => time(), 'url' => '/admin/public/login', 'ip' => $ip]);

$this->message('success', '登录成功', '/admin/public/index');

}

}

}

}

### 6.1.2后台首页

管理员登录成功之后，进入后台首页，后台首页主要是展示系统的一些基本信息。如网站发布的新闻与资讯总数，公司产品的总数，游客留言总数，以及系统和版权的信息等。网站首页的页面如图5-2所示：



图5-2后台首页信息

后台首页（admin/index/index.phtml）的页面布局较为简单，主要由3个<iframe>标签定义头部，菜单和右侧的内容视图，控制器代码如下：

public function sysinfoAction()

{

$jinlingArticleM = new JinLingArticle();

$jinlingLeaveMsgM = new JinLingLeaveMsg();

$jinlingProductM = new JinLingProduct();

$articleNum = $jinlingArticleM->count('');

$leaveMsgNum = $jinlingLeaveMsgM->count('');

$productNum = $jinlingProductM->count('');

$this->view->setVars(

[

'articleNum'=>$articleNum,

'leaveMsgNum'=>$leaveMsgNum,

'productNum'=>$productNum,

'serverName'=>$\_SERVER['SERVER\_NAME'],

'os'=>PHP\_OS,

'serverSofeware'=>$\_SERVER ['SERVER\_SOFTWARE'],

'database'=>'mysql 5.7.1'

]

);

}

页面视图的代码如下：

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>金陵贸易 后台管理系统</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<?php $this->partial("shared/common/commonStyle"); ?>

</head>

<body scroll="no">

<table height="100%" cellspacing="0" cellpadding="0" width="100%">

<tbody>

<tr>

<td colspan="2" height="69">

<iframe name="header" src="/admin/index/top" frameBorder="0" width="100%" scrolling="no" height="69"></iframe>

</td>

</tr>

<tr>

<td valign="top" width="160">

<iframe name="menu" src="/admin/index/menu" frameBorder="0" width="160" scrolling="no" height="100%" target="main"></iframe>

</td>

<td valign="top" width="100%">

<iframe style="overflow:visible" name="main" src="/admin/index/sysinfo" frameBorder="0" width="100%" scrolling="yes" height="100%"></iframe>

</td>

</tr>

</tbody>

</table>

</body>

</html>

左侧菜单对应的视图为/admin/index/menu.phtml

<body>

<div class="menu">

<ul id="leftmenu">

<li><a class="tabon" href="/admin/index/sysinfo" target="main">首页</a> </li>

<li><a href="/admin/index/companyIntro" target="main">公司简介</a> </li>

<li><a href="/admin/index/articleList" target="main">最新公告</a> </li>

<li><a href="/admin/index/productList" target="main">产品管理</a> </li>

<li><a href="/admin/index/contactUs" target="main">联系我们</a> </li>

<li><a href="/admin/index/admin" target="main">管理员</a> </li>

</ul>

</div>

</body>

右侧内容视图为admin/index/sysinfo.phtml，其主要代码如下：

<div class="container">

<h3>统计信息</h3>

<ul class="memlist fixwidth">

<li><em><a href="/admin/index/productList">产品总数</a>:</em><?=$productNum?></li>

<li><em><a href="/admin/index/articleList">文章总数</a>:</em><?=$articleNum?></li>

<li><em><a href="/admin/index/guestBook">留言总数</a>:</em><?=$leaveMsgNum?></li>

</ul>

<h3>系统信息</h3>

<ul class="memlist fixwidth">

<li><em>主机名:</em><?=$serverName?></li>

<li><em>操作系统</em><?=$os?></li>

<li><em>服务器软件:</em><?=$serverSofeware?></li>

<li><em>数据库平台:</em><?=$database?></li>

</ul>

<h3>版权信息</h3>

<ul class="memlist fixwidth">

<li>

<em>版权所有:</em>

<em class="memcont"><a href="http://www.jinling.com/" target="\_blank">金陵贸易有限公司</a></em>

</li>

<li>

<em>程序版本:</em>

<em class="memcont">Jinling 1.0 Release</em>

</li>

<li>

<em>技术支持:</em>

<em class="memcont">admin@gmail.com</em>

</li>

</ul>

</div>

### 6.1.3产品管理

产品管理模块主要功能是企业网站后台管理员登录之后查看、添加、修改、删除需要在企业网站前台系统向访客展示的产品。主要控制器是ProductController，ProductController继承自BaseController，主要包含2个“动作”（action）,分别是：indexAction和editAction。indexAction主要是通过访问JinLingProduct模型中的函数，查询数据库，为产品列表视图提供所需要的数据，同时可以处理Post请求来批量删除产品。editAction主要是处理产品的编辑，更新和插入新产品相关的逻辑。在JinLingProduct模型中定义了getSource()函数指定数据库表，同时继承自Model，包含产品的增删改查的函数。后台产品管理模块的视图层主要有2个页面，分别是产品列表和产品编辑页面。

产品管理相关的控制器代码如下：

<?php

namespace MyApp\Controllers\Admin;

use MyApp\Controllers\Admin\BaseController;

use MyApp\Library\Paginator;

use MyApp\Models\JinLingArticle;

use MyApp\Models\JinLingCompanyInfo;

use MyApp\Models\JinLingLeaveMsg;

use MyApp\Models\JinLingProduct;

class ProductController extends BaseController

{

/\*\*

\* 产品列表

\*/

public function indexAction()

{

//产品数据模型

$productM = new JinLingProduct();

$currentPage = $this->request->get('page', 'int', 1); //当前页

if($this->request->isPost()){//删除产品

$delProductIds = $this->request->getPost('delProductIds','array',[]);

if(!count($delProductIds) > 0)

{

$this->message('error','请至少选择一个产品！','/admin/product/index');

}else{

$deleteResult = $productM->delete(

[

'id'=>$delProductIds

]

);

if($deleteResult){

$this->message('success','删除产品成功！','/admin/product/index');

}else{

$this->message('error','删除产品失败，请重试！','/admin/product/index');

}

}

}

//左连查询

$join = ["[>]user" => ["user\_id" => "id"]];

$pageSize = 10;//每页显示10条数据

$offset = $pageSize \* ($currentPage - 1);

$where['ORDER'] = ['id' => 'DESC'];

$count = $productM->count($where);

$page = new Paginator($count, $pageSize);//分页类

$where['LIMIT'] = [$offset, $pageSize];

$columns = ['products.name', 'products.pic\_path', 'users.name', 'products.register\_time'];

//连表查询所需要的数据

$products = $productM->db->select('products', $join, $columns, $where);

$products = $products ? $products : [];

//给前台页面传递数据

$this->view->setVar('products', $products);

$this->view->setVar('page', $page->showpage());

}

}

其主要的页面视图代码如下：

<?php

foreach ($products as $product) {

?>

<tr>

<td width="80"><input name="<?=$product['id'];?>" value="" class="checkbox" type="checkbox"></td>

<td width="200"><strong><?=$product['name'];?></strong></td>

<td width="100"><img src=<?=$product['pic\_path'];?>></td>

<td width="100"><?=$product['userName'];?>?></td>

<td width="150"><?php echo date('Y-m-d H:i',$product['add\_time']); ?></td>

<td width="100"><a href="/admin/product/edit?productId=<?=$product['id']?>" target="main">编辑</a></td>

</tr>

<?php

}

?>

产品列表页面显示效果如下图所示：

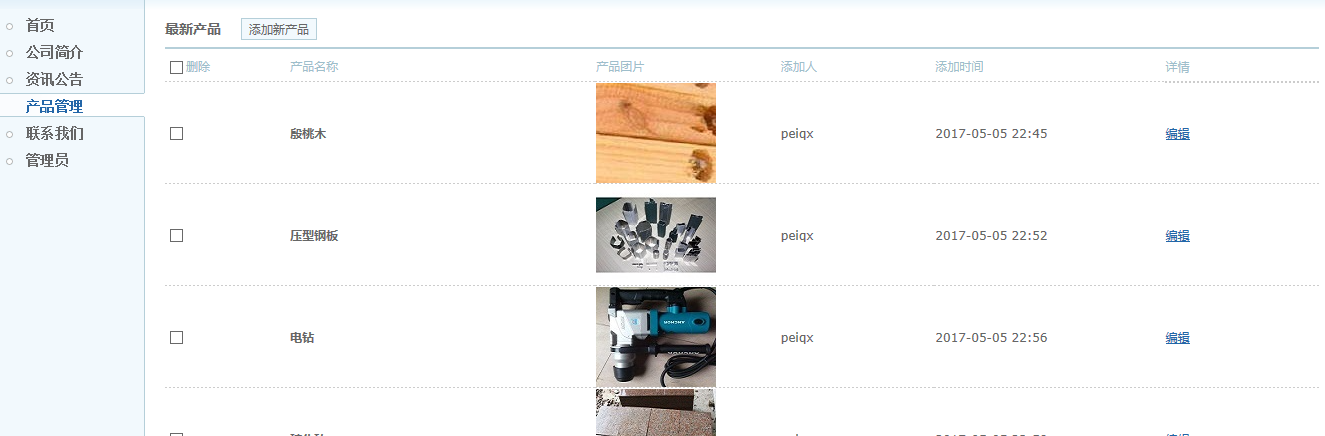


图5-3产品管理页

### 6.1.4 资讯公告管理

（1）资讯公告列表

公告和资讯是企业网站上发布的文章，后台管理员可以在公告资讯管理里面添加文章和修改文章，公告资讯首页主要是文章的列表，管理员可以点击编辑，也可以多选删除等。页面数据通过调用/admin/IndexController的articleListAction函数获取数据，在函数中通过Medoo连表查询获取数据，并通过调用分页类来实现数据的分页加载，articleListAction函数的代码如下：

/\*\*

\* 文章列表

\*/

public function articleListAction()

{

//文章模型

$articleM = new JinLingArticle();

$join = ["[>]user" => ["pub\_uid" => "id"]];

$currentPage = $this->request->get('page', 'int', 1); //当前页

$pageSize = 10;//每页显示10条数据

$offset = $pageSize \* ($currentPage - 1);

$where['ORDER'] = ['userId' => 'DESC'];

$count = $articleM->count($where);

$page = new Paginator($count, $pageSize);//分页类

$where['LIMIT'] = [$offset, $pageSize];

$columns = ['user.name','articles.title','articles.id','articles.pub\_time'];

$articles = $articleM->db->select('articles',$join,$columns,[]);

$articles = $articles ? $articles : [];

//给前台页面传递数据

$this->view->setVar('article',$articles);

$this->view->setVar('page',$page->showpage());

}

资讯公告列表页面的视图层代码如下：

<div class="container">

<h3 class="marginbot">资讯公告<a href="news\_edit.html" class="sgbtn">添加新文章</a></h3>

<div class="mainbox">

<form action="" method="post">

<table class="datalist fixwidth">

<tbody>

<tr>

<th nowrap="nowrap"><input name="chkall" id="chkall" class="checkbox" type="checkbox"><label for="chkall">删除</label></th>

<th nowrap="nowrap">文章名称</th>

<th nowrap="nowrap">添加人</th>

<th nowrap="nowrap">添加时间</th>

<th nowrap="nowrap">详情</th>

</tr>

<?php foreach ($articles as $article):?>

<tr>

<td width="80"><input name="<?=$article['id']?>" value="<?=$article['id']?>" class="checkbox" type="checkbox"></td>

<td><strong><?=$article['title']?></strong></td>

<td width="100"><?=$article['name']?></td>

<td width="150"><?=date("Y-m-d H:i",$article['pub\_time'])?></td>

<td width="100"><a href="/admin/index/articleEdit">编辑</a></td>

</tr>

<? endforeach;?>

<tr class="nobg">

<td><input value="提 交" class="btn" type="submit"></td>

<td class="tdpage" colspan="4">

<div class="pages">

<?php echo $page?><!-- 分页的加载 -->

</div>

</td>

</tr>

</tbody>

</table>

<div class="margintop"></div>

</form>

</div>

</div>

从代码中可以知道，系统在控制器中通过$this->view->setVar()函数来向视图层传递数据，视图中直接通过setVar函数定义的变量名来接收数据，并通过一个foreach循环来遍历输出每一个文章，资讯公告里列表页面的显示效果如图5-4所示：

图5-4资讯公告列表

（2）编辑与新增公告或资讯

在资讯公告列表页点击“编辑”或者“添加新文章”都可以进入公告资讯编辑页面。其控制器代码主要是admin/IndexController的articleEditAction函数，在函数中获取一个GET请求的参数，名为“articleId”，并通过该参数是否存在来判断当前操作是“更新”还是新建，相应的控制器代码如下：

/\*

\* 编辑或者新增文章

\*/

public function articleEditAction()

{

$articleId = $this->request->get('articleId', 'int', 0);

$articleM = new JinLingArticle();

$articleType = $this->request->getPost('articleType', 'string', '');

$articleTitle = $this->request->getPost('articleTitle', 'string', '');

$articleContent = $this->request->getPost('articleContent', 'string', '');

$publishUserId = $this->session->get('\_adminId');//获取当前登录的管理员

if ($articleId) {//编辑文章

$article = $articleM->get('\*', ['id' => $articleId]);

$this->view->setVar('article', $article);

if ($this->request->isPost()) {//提交更新

$updateResult = $articleM->update(

[

'article\_type' => $articleType,

'pub\_uid' => $publishUserId,

'title' => $articleTitle,

'content' => $articleContent

],

[

'id' => $articleId

]

);

if ($updateResult !== false) {

$this->message('success', '修改成功！', '/admin/index/articleList');

} else {

$this->message('error', '修改失败！', '/admin/index/articleList');

}

} else {

if ($this->request->isPost()) {

$insertResult = $articleM->insert(

[

'article\_type' => $articleType,

'pub\_uid' => $publishUserId,

'title' => $articleTitle,

'content' => $articleContent

]

);

if ($insertResult !== false) {

$this->message('success', '修改成功！', '/admin/index/articleEdit');

} else {

$this->message('error', '添加失败！', '/admin/index/articleEdit');

}

}

}

}

}

其视图页面相关的代码如下：

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>编辑文章</title>

<link rel="stylesheet" href="styles/style.css" type="text/css" media="all">

<style type="text/css">

.err{

text-align: left;

border: 1px solid red;

background-color: #FFEBEB;

}

</style>

<script type="text/javascript">

function check(fm) {

var eleTitle = document.getElementById("article\_title");

var eleContent = document.getElementById("article\_content");

var errTitle = document.getElementById("err\_article\_title");

var errContent = document.getElementById("err\_article\_content");

if(eleTitle.value == ""){

errTitle.style.display = "inline";

eleTitle.focus();

return false;

}

errTitle.style.display = "none";

if(eleContent.value == ""){

errContent.style.display = "inline";

eleContent.focus();

return false;

}

errContent.style.display = "none";

return true;

}

</script>

</head>

<body>

<div class="container">

<h3 class="marginbot">编辑文章<a href="/admin/index/articleList" class="sgbtn">返回文章列表</a></h3>

<div class="mainbox">

<form action="/admin/index/articleEdit" method="post" enctype="multipart/form-data" target="main" onsubmit="return check(this)">

<input type="hidden" name = "articleId" value="<?php if($article): echo $article['id']; endif;?>"/>

<table class="opt" style="width:600px;">

<tbody>

<tr>

<th>文章标题：</th>

</tr>

<tr>

<td>

<input name="article\_title" id="article\_title" class="txt" value="<?php if($isArticleUpdate && $articleObj->id != 0){echo $articleObj->title;} ?>" style="width:400px;" type="text">

</td>

<td><p class="err" style="display:none" id="err\_article\_title">文章标题不能为空！</p></td>

</tr>

<tr>

<th>文章类别：

<select name="article\_type">

<option <?php if($article && $article['article\_type'] == 1):?> selected = "selected" <?php endif;?>value="1">公 告</option>

<option <?php if($article && $article['article\_type'] == 1):?> selected = "selected" <?php endif;?>value="2">资 讯</option>

</select>

</th>

</tr>

<tr>

<th>文章内容：</th>

</tr>

<tr>

<td><textarea style="width:400px; height:150px" name="article\_content" id = "article\_content"><?php if($article):?><?= $article['content']; endif;?></textarea>

</td>

<td><p class="err" id = "err\_article\_content" style="display: none">文章内容不能为空</p></td>

</tr>

</tbody>

</table>

<div class="opt"><input name="submit" value=" 提 交 " class="btn" tabindex="3" type="submit"></div>

</form>

</div>

</div>

</body>

</html>

文章编辑的页面显示效果如图5-所示：



图5-5新增或者编辑新闻或资讯

### 6.1.5管理员管理

AdminController是管理员管理模块的控制器，主要包含3个Action函数，分别是indexAction、adminAddAction和adminEditAction，其中indexAction是管理员管理首页，主要查询当前系统的管理员列表并返回，并处理管理员删除的功能逻辑。adminAddAction是管理员添加的相关代码，该Action没有对应的视图，由/admin/admin/index.phtml视图通过一个ajax请求来实现。adminEditAction主要是实现管理员信息的更新。控制器相关代码如下：

<?php

namespace MyApp\Controllers\Admin;

use MyApp\Controllers\Admin\BaseController;

use MyApp\Library\Paginator;

use MyApp\Library\Utilities;

use MyApp\Models\JinLingAdmin;

use MyApp\Services\AdminService;

class AdminController extends BaseController

{

/\*\*

\* 管理员管理首页

\*/

public function indexAction()

{

$adminM = new JinLingAdmin();

//批量删除管理员

if($this->request->isPost())

{

$delAdminIds = $this->request->getPost('delAdminIds','array',[]);

if(!count($delAdminIds) > 0){//为空

$this->message('notice', '请至少选择一项！', '/admin/admin/index');

}else{

$deleteResult = $adminM->delete(

[

'id'=>$delAdminIds

]

);

if($deleteResult){

$this->message('success', "删除成功！", '/admin/admin/index');

}

}

}

//查询所有的管理员信息

$currentPage = $this->request->get('page', 'int', 1); //当前页

$pageSize = 10;//每页显示10条数据

$offset = $pageSize \* ($currentPage - 1);

$where['ORDER'] = ['id' => 'DESC'];//按照注册时间降序排序

$count = $adminM->count($where);

$page = new Paginator($count, $pageSize);//分页类

$where['LIMIT'] = [$offset, $pageSize];

$admins = $adminM->select('\*',$where);

$this->view->setVars(

[

'admins'=>$admins,

'page'=>$page

]

);

}

/\*\*

\* 管理员新增用户

\*/

public function adminAddAction(){

$this->view->disable();//不显示视图

$utilities = new Utilities();

if($this->request->isPost()){

$userName = $this->request->getPost('userName','string','');

$password = $this->request->getPost('password','string','');

$adminM = new JinLingAdmin();

$adminService = new AdminService();

if(!$userName){

$utilities->apiResultStandard('1001','用户名不能为空！');

}elseif(!$password){

$utilities->apiResultStandard('1002','密码不能为空！');

}elseif(strlen($password) < 6 || strlen($password) > 18){

$utilities->apiResultStandard('1003','密码不能小于6位，且不能大于18位！');

}else{

$password = $adminService->hashPassword($userName,$password);

$insertResult = $adminM -> insert(

[

'name'=>$userName,

'password'=>$password,

'register\_time'=>time(),

'register\_ip'=>$utilities->getClientIP(),

'user\_state'=>1

]

);

if($insertResult){

$utilities->apiResultStandard('1000','新增用户成功！',$insertResult);

}else{

$utilities->apiResultStandard('1004','新增用户失败，请稍后再试！');

}

}

}

}

/\*\*

\* 管理员编辑页面

\*/

public function adminEditAction()

{

$adminId = $this->request->get('adminId', 'int', 0);

$adminM = new JinLingAdmin();

$userName = $this->request->getPost('userName', 'string', '');

$password = $this->request->getPost('password', 'string', '');

$adminState = $this->request->getPost('adminState', 'int', '');

if ($adminId) {//编辑管理员信息

$admin = $adminM->get('\*', ['id' => $adminId]);

$this->view->setVar('admin', $admin);

if ($this->request->isPost()) {//提交更新

$updateResult = $adminM->update(

[

'name' => $userName,

'password' => $password,

'user\_state' => $adminState

],

[

'id' => $adminId

]

);

if ($updateResult !== false) {

$this->message('success', '修改成功！', '/admin/admin/index');

} else {

$this->message('error', '修改失败！', '/admin/admin/index');

}

} else {

if ($this->request->isPost()) {

$insertResult = $adminM->insert(

[

'name' => $userName,

'password' => $password,

'user\_state' => $adminState

]

);

if ($insertResult !== false) {

$this->message('success', '修改成功！', '/admin/admin/index');

} else {

$this->message('error', '添加失败！', '/admin/admin/index');

}

}

}

}

}

}

管理员管理首页（app/views/admin/admin/index.phtml）的页面代码如下：

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>管理员列表</title>

<?php $this->partial("shared/common/commonStyle"); ?>

</head>

<body>

<div class="container">

<div class="hastabmenu">

<form action="/admin/admin/adminAdd" target="main" method="post">

<ul class="tabmenu">

<li id="adduserbtn" class="tabcurrent"><a href="#" >添加管理员</a></li>

</ul>

<div id="adduserdiv" class="tabcontentcur">

<table width="100%">

<tbody>

<tr>

<td>用户名:</td>

<td><input name="username" class="txt" type="text"></td>

<td>密码:</td>

<td><input name="password" class="txt" type="password"></td>

<td><input value="提 交" name = "subAddUser" class="btn" type="submit"></td>

</tr>

</tbody>

</table>

</form>

</div>

</div>

<br>

<h3>管理员列表</h3>

<div class="mainbox">

<form action="/admin/admin/index" target="main" method="post">

<table class="datalist fixwidth">

<tbody>

<tr>

<th>

<input name="chkall" id="chkall" class="checkbox" type="checkbox"><label for="chkall">删除</label>

</th>

<th>用户名</th>

<th>注册日期</th>

<th>注册IP</th>

<th>用户状态</th>

<th>编辑</th>

</tr>

<?php foreach ($admins as $admin):?>

<tr>

<td class="option">

<input name="<?=$admin->id;?>" value="<?$admin->id;?>" class="checkbox" type="checkbox">

</td>

<td><strong><?=$admin->name;?></strong></td>

<td><?=date('Y-m-d H:i',$admin->register\_time);?></td>

<td><?=$admin->register\_ip;?></td>

<td><?=$admin->user\_state == 0 ? '未启用' : '启用';?></td>

<td><a href="/admin/admin/adminEdit?adminId=<?=$admin->id;?>">编辑</a></td>

</tr>

<?php endforeach;?>

<tr class="nobg">

<td><input value="提 交" name="subDelUser" class="btn" type="submit"></td>

<td class="tdpage" colspan="6">

<div class="pages">

<?= $page?><!-- 分页 -->

</div>

</td>

</tr>

</tbody>

</table>

</form>

</div>

</div>

</body>

</html>

管理员编辑页面（views/admin/admin/adminEdit.phtml）的相关代码如下所示：

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>编辑用户资料</title>

<?php $this->partial("shared/common/commonStyle"); ?>

</head>

<body>

<div class="container">

<h3 class="marginbot">编辑用户资料<a href="/admin/admin/index" class="sgbtn">返回用户列表</a></h3>

<div class="mainbox">

<form action="/admin/admin/adminEdit" target="main" method="post">

<table class="opt">

<tbody>

<tr>

<th>用户名:</th>

</tr>

<tr>

<td>

<input name="username" class="txt" value="<?php if(count($admin)) echo $admin['name']?>" type="text">

</td>

</tr>

<tr>

<th>密　码:<span style="font-weight:normal"> [密码留空，保持不变]</span></th>

</tr>

<tr>

<td>

<input name="password" value="" class="txt" type="password">

</td>

</tr>

<tr>

<th>用户状态：</th>

<td>

<select name="userState">

<option <?php if($admin['user\_state'] == '1'):?> selected = "selected"<?php endif;?>value="1">启用</option>

<option <?php if($admin['user\_state'] == '1'):?> selected = "selected"<?php endif;?>value="2">未启用</option>

</select>

</td>

</tr>

</tbody>

</table>

<div class="opt"><input name="submit" value=" 提 交 " class="btn" tabindex="3" type="submit"></div>

</form>

</div>

</div>

</body>

</html>

管理员编辑页面的显示效果如图5-6所示：



图5-6编辑管理员用户

### 6.1.6公司简介

CompanyController主要是处理公司相关信息的控制器，在CompanyController里面有一个companyIntroAction函数来处理公司简介的查询和更新操作，其代码如下：

<?php

namespace MyApp\Controllers\Admin;

use MyApp\Controllers\Admin\BaseController;

use MyApp\Models\JinLingCompanyInfo;

class IndexController extends BaseController

{

/\*\*

\* 公司简介

\*/

public function companyIntroAction()

{

$companyInfoM = new JinLingCompanyInfo();

$companyIntro = $companyInfoM->get('cpn\_intro', []);

$companyIntro = $companyIntro ? $companyIntro : '';

if ($this->request->isPost()) {

$companyIntro = $this->request->getPost('companyIntro', 'string', '');

$updateResult = $companyInfoM->update([

'cpn\_intro' => $companyIntro,

'add\_time' => time()

], ['id' => 1]);

if ($updateResult !== false) {

$this->message('success', '修改成功！', '/admin/index/companyIntro');

} else {

$this->message('error', '修改失败！', '/admin/index/companyIntro');

}

}

$this->view->setVar('companyIntro', $companyIntro);

}

}

对应的页面视图（views/admin/company/companyIntro.phtml）代码如下：

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>联系我们</title>

<?php $this->partial("shared/common/commonStyle"); ?>

</head>

<body>

<div id="append"></div>

<div class="container">

<h3>联系我们</h3>

<div class="mainbox">

<form action="/admin/company/companyIntro" method="post">

<table style="width:700px;">

<tbody>

<tr>

<td><textarea rows="15" cols="80" name="companyIntro"><?=$companyIntro?></textarea></td>

</tr>

<tr>

<td>添加人：<?php var\_dump($JinLingUser)?> &nbsp;&nbsp;&nbsp;添加时间：2009-07-01 12:05</td>

</tr>

<tr>

<td><input value="提 交" class="btn" type="submit"></td>

</tr>

</tbody>

</table>

</form>

</div>

</div>

</body>

</html>

页面显示的效果如图5-6所示：

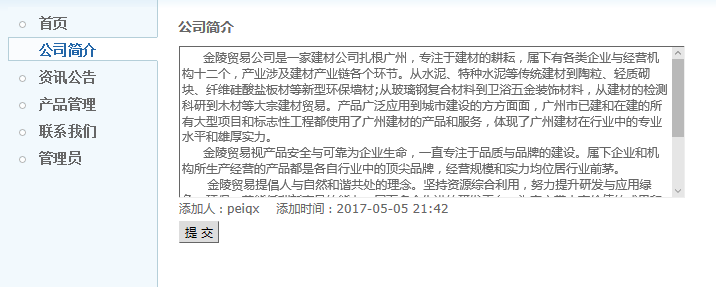


图5-6公司简介管理页

## 6.2企业网站前台实现

### 6.2.1企业网站前台首页

企业网站前台首页展示了公司的整体情况，所以需要查询的数据量最大。其控制器在controller的front目录下面,取名为IndexController，企业网站首页的页面视图对应从其对应的indexAction()函数获取数据，相关代码如下：

<?php

namespace MyApp\Controllers;

use MyApp\Controllers\BaseController;

use MyApp\Models\JinLingArticle;

use MyApp\Models\JinLingCompanyInfo;

use MyApp\Models\JinLingProduct;

use MyApp\Services\CountersService;

class IndexController extends BaseController

{

public function indexAction()

{

$productM = new JinLingProduct();

$articleM = new JinLingArticle();

$companyInfoM = new JinLingCompanyInfo();

$where = [];

$where['ORDER']['add\_time'] = 'DESC';

$where['LIMIT'] = 6;

$products = $productM->select('\*',$where);

$where = [];

$where['ORDER']['pub\_time'] = 'DESC';

$where['LIMIT'] = 6;

$where['AND']['article\_type']=2;

$infomations = $articleM->select('\*',$where);//行业资讯

$where = [];

$where['ORDER']['pub\_time'] = 'DESC';

$where['LIMIT'] = 6;

$where['AND']['article\_type']=1;

$news = $articleM->select('\*',$where);//行业资讯

$companyInfo = $companyInfoM->get('\*',['id'=>1]);

$this->view->setVars(

[

'products'=>$products,

'infomations'=>$infomations,

'news'=>$news,

'companyInfo'=>$companyInfo

]

);

}

}

页面显示效果如图x-x所示：



图5-7前台首页

### 6.2.2企业网站公司简介

企业网站公司简介主要由公司简介的内容和最新公告列表组成，其控制器代码为IndexController中的companyIntroAction()函数，具体代码如下：

public function companyIntroAction(){

$articleM = new JinLingArticle();

$companyInfoM = new JinLingCompanyInfo();

$blogrollM = new JinLingBlogrolls();

$blogrolls = $blogrollM->select('\*','');//友情链接

$where = [];

$where['ORDER']['pub\_time'] = 'DESC';

$where['LIMIT'] = 6;

$where['AND']['article\_type']=1;

$news = $articleM->select('\*',$where);//行业资讯

$companyInfo = $companyInfoM->get('\*',['id'=>1]);

$this->view->setVars(

[

'blogrolls'=>$blogrolls,

'news'=>$news,

'companyInfo'=>$companyInfo

]

);

}

页面视图层的主要代码如下：

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title><?php echo $companyInfo['cpn\_name']?></title>

<?php $this->partial("shared/common/commonStyle"); ?>

</head>

<body onload="startTime()">

<div class="header">

<h1 class="logo" title="<?=$companyInfo['cpn\_name']; ?>"><a href="/front/index/index"><img src="/front/images/logo.gif" alt="<?= $companyInfo['cpn\_name']; ?>" /></a></h1>

<p class="top\_r"><a href="javascript:void(0);" onclick="SetHome(this,'http://112.74.109.131');" class="btn\_i">设为主页</a><a href="javascript:void(0);" onclick="AddFavorite('我的网站',location.href)" class="btn\_f">收藏本站</a></p>

</div>

<div class="nav">

<div class="nav\_left"></div>

<ul>

<li><a href="/front/index/index" >首 页</a></li>

<li class="sel"><a href="/front/index/companyIntro">公司简介</a></li>

<li><a href="/front/index/productList">产品展示</a></li>

<li><a href="/front/index/info">行业资讯</a></li>

<li><a href="/front/index/guestbook">客户留言</a></li>

<li><a href="/front/index/contactUs" class="nobg">联系我们</a></li>

</ul>

<div class="time"></div>

<div class="nav\_right"></div>

</div>

<div class="banner">

<a href="#"><img src="/front/images/banner.jpg" align="banner" /></a>

</div>

<div class="content">

<div class="lefter">

<div class="title">

<h2 class="cBlue fB">公司简介<b class="cGrey fn">About us</b></h2>

</div>

<div class="intro">

<?=$companyInfo['cpn\_intro'];?>

</div>

</div>

<div class="righter">

<div class="rightBox">

<a href="/front/index/guestbook" class="btn\_s">我要留言</a>

</div>

<div class="blank10"></div>

<div class="rightBox">

<h3>最新公告<b class="cGrey fn">News</b></h3>

<ul class="list\_r">

<?php if(count($news) > 0){

foreach ($news as $newsitem) {

?>

<li>

<a href="article.php?type=1&id=<?php echo $newsitem['id'];?>"><?php echo mb\_substr($news['title'],0,13).'...';?></a>

</li>

<?php

}

} ?>

</ul>

</div>

<div class="blank10"></div>

<div class="rightBox">

<h3>友情链接<b class="cGrey fn">Links</b></h3>

<ul class="list\_r">

<?php if(count($arryBlogrolls) > 0){

foreach ($arryBlogrolls as $blogrollObj){

?>

<li><a href="<?php echo $blogrollObj->content ?>" target="\_blank"><?php echo $blogrollObj->name ?></a></li>

<?php

}

}?>

</ul>

</div>

</div>

<div class="hackbox"></div>

</div>

<div class="footer">

<p>地址：<?=$companyInfo['addr']?> 联系人：<?=$companyInfo['contacts']?> 移动电话：<?=$companyInfo['mobile']?> 固定电话：<?=$companyInfo['fixed\_tel']?>传 真：<?=$companyInfo['fax']?></p>

<p><?=$companyInfo['copyright\_no']?> <?=$companyInfo['cpn\_name']?> 版权所有</p>

<p><a href="#"><?=$companyInfo['icp\_no']?></a></p>

</div>

</body>

</html>

公司简介页面效果如图 x-x所示



图5-8公司简介前台页面

### 6.2.3企业网站产品详情

产品详情页需要查询产品详情，在请求时通过GET的方式将产品id发送到Controller中，Controller获取到产品id，从而调用JinLingProducts模型里面的函数获取数据，具体代码如下：

/\*\*

\* 产品详情页需要查询的数据

\*/

public function productDetailAction(){

$jinlingProduct = new JinLingProduct();

$articleM = new JinLingArticle();

$companyInfoM = new JinLingCompanyInfo();

$productId = $this->request->get('productId','int',0);

$productMsg = $jinlingProduct->get('\*',['id'=>$productId]);

$blogrollM = new JinLingBlogrolls();

$blogrolls = $blogrollM->select('\*','');//友情链接

$where = [];

$where['ORDER']['pub\_time'] = 'DESC';

$where['AND']['article\_type']=1;

$news = $articleM->select('\*',$where);//公告

$companyInfo = $companyInfoM->get('\*',['id'=>1]);

$this->view->setVars(

[

'productMsg'=>$productMsg,

'blogrolls'=>$blogrolls,

'news'=>$news,

'companyInfo'=>$companyInfo

]

);

}



图5-9产品详情页

### 6.2.4企业网站公告与资讯展示

企业网站的公告与资讯页面的可以查看所有的行业资讯和最新的公告。其控制器端代码如下：

/\*\*

\* 公告或生活

\*/

public function articleListAction()

{

$articleM = new JinLingArticle();

$companyInfoM = new JinLingCompanyInfo();

$blogrollM = new JinLingBlogrolls();

$blogrolls = $blogrollM->select('\*','');//友情链接

$where = [];

$where['ORDER']['pub\_time'] = 'DESC';

$where['AND']['article\_type']=1;

$currentPage = $this->request->get('page', 'int', 1); //当前页

$pageSize = 10;//每页显示10条数据

$offset = $pageSize \* ($currentPage - 1);

$count = $articleM->count($where);

$page = new Paginator($count, $pageSize);//分页类

$where['LIMIT'] = [$offset, $pageSize];

$information = $articleM->select('\*',$where);//行业资讯

$companyInfo = $companyInfoM->get('\*',['id'=>2]);

//公告

$where = [];

$where['ORDER']['pub\_time'] = 'DESC';

$where['AND']['article\_type']=1;

$news = $articleM->select('\*',$where);//公告

$this->view->setVars(

[

'blogrolls'=>$blogrolls,

'news'=>$news,

'companyInfo'=>$companyInfo,

'page'=>$page->showpage(),

'information'=>$information

]

);

}

其显示效果图如下面：



图5-11公告详情页



图5-12行业咨询页

### 6.2.5企业网站客户留言

客户留言页面主要有留言输入框和最新公告、友情链接等。游客可以输入昵称和内容，通过一个Ajax的POST请求，并存储在数据库中。其控制器主要业务代码如下：

<?php

namespace MyApp\Controllers\Front;

use MyApp\Controllers\Front\BaseController;

use MyApp\Library\Paginator;

use MyApp\Library\Utilities;

use MyApp\Models\JinLingArticle;

use MyApp\Models\JinLingBlogrolls;

use MyApp\Models\JinLingCompanyInfo;

use MyApp\Models\JinLingLeaveMsg;

use MyApp\Models\JinLingProduct;

/\*\*

\* Class IndexController

\* @package MyApp\Controllers\Front

\*/

class LeaveMsgController extends BaseController

{

public function indexAction()

{

$productM = new JinLingProduct();

$articleM = new JinLingArticle();

$companyInfoM = new JinLingCompanyInfo();

$blogrollM = new JinLingBlogrolls();

$blogrolls = $blogrollM->select('\*','');//友情链接

$where = [];

$where['ORDER']['pub\_time'] = 'DESC';

$where['LIMIT'] = 6;

$where['AND']['article\_type']=1;

$news = $articleM->select('\*',$where);//公告

$companyInfo = $companyInfoM->get('\*',['id'=>1]);

$this->view->setVars(

[

'blogrolls'=>$blogrolls,

'news'=>$news,

'companyInfo'=>$companyInfo,

]

);

}

public function saveAction()

{

$leaveMsg = new JinLingLeaveMsg();

$adminId = $this->session->get('\_admin');

$nickName = $this->request->getPost('nickName');

$content = $this->request->getPost('content');

$utilies = new Utilities();

//插入一条客户留言

$insertResult = $leaveMsg->insert(

[

'nickName'=>$nickName,

'content'=>$content,

'pub\_time'=>time()

]

);

if($insertResult){

$utilies->apiResultStandard('1000',"插入留言成功!",$insertResult);

}else{

$utilies->apiResultStandard('1002','插入留言失败!');

}

$this->view->disable();

}

}

其核心页面的显示效果如下：



图5-13客户留言页面

**6，总结**

本文通过对一个企业网站的需求，使用用例图对本系统进行了分析设计了该金陵贸易有限公司的企业网站，该网站前台设计了首页，公司简介，产品展示，最新资讯，联系我们，客户留言六个模块，并对这六个板块进行了页面排版，通过时序图，流程图和ER图进行了后台的设计，包括首页，产品管理，公告资讯管理，留言管理，联系我们信息管理，公司简介管理，管理员管理等模块，使用php语言以及Phalcon框架对后台进行了设计实现。

该系统主要的优点是：

1. ，本系统前台界面简洁大方，很方便用户的使用
2. ，该系统没有用户登录的操作，方便用户浏览企业信息
3. ，该系统后台操作简单，便于管理
4. ，该系统使用php开发，运行速度开，便于升级和维护。

由于时间少，任务重，系统在安全方面还存在一定的问题，希望在以后的工作中可以完善本系统。

**致谢**

我对PHP网站建设的兴趣主要来源于近几年PHP的发展态势，以及我的实习单位是使用php进行后台开发的，在毕业论文完成之际，首先我要感谢我的导师钟杰卓老师，她在我选题以及论文的指导上给了我很多的建议，同时督促我按时完成论文。我相信这段时间的php语言的学习会成为我以后工作中的一大助力，同时在此感谢在我论文创作过程中给我带来很大帮助的老师和同学，从开题到资料的收集，项目的实现等都得到了各位的大力支持。在即将毕业之际，非常感谢各位不论是在生活上还是在学习上对我的关心与支持。

**参考文献**

[1]王晓滨. 基于PHP的物业管理网站设计与实现[D].中国地质大学(北京),2016.

[2]吉小霞. 基于PHP的企业网站开发及安全实现[D].长江大学,2016.

[3]李良. 基于PHP的商业站点设计与实现[D].南昌大学,2010

[4]张子丰. 基于PHP的商务网站设计与实现[D].电子科技大学,2015.

[5]吕忠文. 基于PHP的医院管理信息系统的设计与实现[D].吉林大学,2014.

[6]万颖. 基于PHP+MySQL的图书在线销售系统的设计与实现[D].吉林大学,2014.

[7]王刚. 基于PHP+MySQL的电商平台系统的设计与实现[D].电子科技大学,2014.

[8]李谋平. 基于PHP的图书馆设备管理系统研究与开发[D].安徽大学,2011.

[9]李良. 基于PHP的商业站点设计与实现[D].南昌大学,2010.

[10]赵建华. 基于PHP的大学生在线影视系统的设计与实现[D].吉林大学,2010.

[11]肖维明. 基于PHP+MySQL的网站开发[J]. 物流工程与管理,2009,(06):90-92.

[12]刘健. 基于PHP的教师博客系统的设计与实现[D].中国海洋大学,2008.

[13]康贤昆. PHP开发框架的研究与实现[D].天津大学,2007.

[14]邱怀姗,朱群雄,史晟辉. 基于PHP的网络化车辆管理系统的设计与实现[J]. 北京化工大学学报(自然科学版),2003,(06):72-76.