

# 内容管理系统

内容管理系统(Content Management System, CMS)是互联网上对信息进行分类管理, 并快捷更新的 Web 应用系统。它能够将杂乱无章的信息,及时、准确、有序地呈现在网络用 户面前,使信息的共享更加快捷和方便。

本章通过一个简单的内容管理系统的开发,向读者介绍 PHP Web 应用程序开发的基本方法、技术规范,以及相关的开发技术。

### 5.1 项目简介



视频讲解

CMS应用非常广泛,例如企业网站、政府信息平台等。在网络中,常见的门户、新闻、博客、文章等类型的网站,以及各类信息、数据查询系统,都可以利用 CMS 进行开发。

### 5.1.1 功能描述

作为第一个项目,通过该项目的开发,熟悉 PHP Web 应用项目开发的各项技术,进一步巩固 PHP 程序设计语言的语法及编程规范并能综合应用基础理论知识,解决实际的工程、技术或应用问题。鉴于此目的,本项目采用面向过程的程序设计方法,实现内容管理系统的基本功能,包括前台展示功能与后台管理功能。

前台功能包括内容的列表展示、内容的分类展示、内容的详情展示、内容的搜索,以及用 户的注册与登录等。

后台功能包括用户管理、内容管理、分类管理等。

### 5.1.2 运行预览

本章项目的完整运行效果,请读者自行运行本书提供的源码,项目目录名称为 chap05。 这里只展示前台首页、内容详情、分类查询、后台主页、后台用户管理页面的运行效果,如 图 5.1~图 5.5 所示。

微梦内容管理系统		× +															-	a	×	
← → C' @ ♦ ⊕ ####	<b>回</b> (社)	<b>ể</b> bo	ok.php.chpl	05					- 10 -	·· ☆	Q. 搜索		μı.	ŧ,	• 6	• •	<b>ා ස</b> ් 	》 @新上的	= <sup>6</sup> 988	
	首页	PHP	AVAL	HTML	CSS	JQUERY	用户中心 🔻	关于我们		涛	输入标题	搜索		₫	<b>剥</b> 注	Ð			Î	
全种	部文章									热	门浏览									
<b>總速入门 🦻 🌧</b>		PHP특 (158):	iAjax操作	(赤島1回: 107)	0.01.01			Ż	章标题				33	ā						
把刺	PHP与Ajax 视频数学	n	PHP与Ajax操作PHP与Ajax操作PHP与Ajax操作PHP与Ajax操作P与 与Ajax操作PHP与Ajax操作PHP与Ajax						PHP与Ajaz提作 慢作PHP ThinkPHP5企业站点开发指南					E	2					
		ittimi (	24.5	年5自	自己动手写框架枚程					ThinkPHP5经典视频数程										
1	<			作者:	wp 发?	師前间: 2018-	01-26			自己动手写框架教程					E	2				
2				自己动程自己	手写框架敷 动手	程自己动手写	框架数程自己动手	写框架数程自己动手	写框架教	87	闻发布系统数程					1				
PH	in <b>R</b>	Alt Alt		PHP核 作者: PHP模 種查询	は期直1句技 admin 3 観直御技术 技术PHP機	术 (布时间:197 PHP機械直向) 構直向	0-01-01 支术PHP模模查询	技术PHP機能查询技术	RPHP											
Th	inkPHF	• •	=i	Thinki 作者:	PHP5企业 admin 3	<b>站点开发指</b> (布时间: 197	<b>南</b> 0-01-01												÷	



微梦内容管理系统		× +												-	٥	×
€ → ℃ ŵ	Ø	🔏 boo	k.php.chp	05/index.ph	p/home/d	etail?id=17			巅 …	● Q 搜索		١I/	<b>4 9 0</b>	•	\$° >>	≡A
♀最常访问 □ 常用网	址			_		_	_	_						0.8	动设备上	2018年
	顷	PHP	AVAL	HTML	CSS	JQUERY	用户中心 🔻	关于我们		请输关键字	搜索		登录 注册	ŧ		
文章	詳情									热门浏览						
PHP	与Aja	x操作								文章标题			透底			
作者: a	in the second	发布时间	: 2018-00	2-01					412.00-	PHP与Ajax操作			5			
PHP5/ PHP5/ PHP5/	yaomente Ajaomente Ajaomente	PHP5Aja PHP5Aja PHP5Aja	X操作PHP4 X操作PHP4 X操作PHP4	Ajaxax無件EP SAjax操作Ph SAjax操作Ph	HP与Ajax提 IP与Ajax提	作PHP与Ajaxim 作PHP与Ajaxim 作PHP与Ajaxim	HPHP与Ajax課作 HPHP与Ajax操作 HPHP与Ajax操作	HP与AjaX展作PHP与A PHP与AjaX操作PHP与A PHP与AjaX操作PHP与A	Vax最佳 Vax最佳	ThinkPHP5企业总统	开发描画		2			
PHP5/	<b>Vadent</b>	PHP与Aja	x WIFFPHPS	5Ajax #/FP	IP与Ajad操	f⊧ f⊧	ant in straighter	and a growth of the	Jocusti	ThinkPHP5径典视频	動程		1			
										自己动手写框架教育	Ł		2			
										新闻发布系统教程			1			

图 5.2 内容详情

微梦内容管理系统	× +													- 6	a ×
(←) → C* @	0 🔏 ba	ook.php.chpl	05/index.php	/home/cat	tegory?cate_	name=php		× .	· tr	Q. 搜索		III\	4 9 0	• ®	» ≡
◆ 最早访问 □ 常用网站	t									7				口移动设	静上的书签
m	页 PHP	AVAL	HTML	CSS	JQUERY	甩户中心 🔻	关于我们		请抢步	关键字	搜索		登录 注册		ĺ
php									热门	浏览					
相通	AD 💋		PHP与	Ajax操作	1940 - 1940 - 1940 - 1940 - 1940 - 1940 - 1940 - 1940 - 1940 - 1940 - 1940 - 1940 - 1940 - 1940 - 1940 - 1940 -				文章	标题			浏范		
PHP	与Ajax	Ä	TER: a	amin 22	April 201	ND AisyavilleD		will/moup	PHP	与Ajax操作			5		
1369949	X <del></del>	11	与Ajax搬	HEPHP-15A	ax操作PHP与	iAjax	nr-Jnjavan Fr-Jn	Agy 280 Fear 16	Thin	kPHP5企业站总	开发指南		2		
	and the second second		PDO#8	神入门約	10				Thin	kPHP5经典视绿	被程		1		
			作者: a	dmin 发	布时间: 201	8-02-01			自己	动手写框架数程			2		
à	A	Ā	PDO极近 极速入门	■入门数程F 1数程PDO	PDO极速入门 极速入门	教程PDO极速入	门教程PDO极速入	门敷程PDO	新闻	发布系统教程					
			自己动	手写框架	教程										- 1
	<"	Salar F	作者: a	dmin 22	布时间: 201	8-02-01									
1			自己动手 程自己动	「写框架教授 り手	星自己动手写	框架教程自己动行	戶写框架数程自己設	的手写框架教							
	1000		PDO极	速入门教	程										
and the second second			作者: a	dmin 发	布时间: 201	8-02-01									

图 5.3 分类查询

后台管理系统 🗙 🕂				- 0
→ C 金 ② & book.php.chj 副第56月 □ 第月月11日	p05/index.php/a	dmirv/index	麗 … 合 包 複素	II/ は 争 口 ゥ & > 1#9#9#H
☆ 微梦后台管理系统		<b>当前位置:</b> 后台 / 主页	現在是: 2020年01月12日 12时30分	当前在线: 微梦 退出系统
> 用户管理	文章列	8		16.30
> 分类管理	ID	标题	藻加时间	操作
> 文章管理	1	PHPJ6Ajaxi變作	2018-02-01 15:35:13	第版 200
> 系統管理	2	自己动手写框架数程	2018-02-01 15:36:08	编辑 1959
> 管理员管理	3	PHP模糊查询技术	2018-02-01 15:37:28	编辑 数19
> 菜单管理	4	ThinkPHP5企业站点开发指南	2018-02-01 15:39:29	SEG 809
	5	ThinkPHP5经典视频数程	2018-02-01 15:41:08	術語 動分
	6	新闻发布系统教程	2018-02-01 15:42:23	编辑 图9
	7	PDO极速入门数程	2018-02-01 15:34:04	論版 對於
	8	PHP与Ajax操作	2018-02-01 15:35:13	新羅 教師
	9	自己动手写框架数程	2018-02-01 15:36:08	编辑 整种
	10	PHP模糊查询技术	2018-02-01 15:37 28	编辑 動物

图 5.4 后台主页

其他页面效果请运行源码,本章项目源码目录为 chap05。运行时请先阅读案例项目运行说明。

→ C <sup>*</sup> 企 0 <u>&amp;</u> book.php.chp05/i 転送月 世 常用発生	ndex.php/	'admin/user_list			i ··· 公	Q. 搜索	li\ t, ⊅	① ち む ※ 〇年初役会上
↓ 微梦后台管理系统 - 管理员信	息列表	当前位置:	后台 / 主页 / 用	的	現在是: 2020年	01月12日 12时32分	当前在线:	微梦 退出系统
· 甩户管理	管理	员列表						38.ac
- 秦加管理员	ID	用户名	真实姓名	角色	状态	爆加时间	緩作	
管理员列表 用户列表	1	admin	做梦	系統管理员	正常	1970-01-03 04:53:41	<b>66</b>	<b>B</b> 30
Owne	2	admin2	微梦2	系統管理员	禁用	1970-09-29 19:46:29	<b>\$6</b>	B39
	3	<b>王</b> —	ΞΞ	系统开发员	正常	2019-06-24 19:08:02	编辑	B(19
x/// 1872	4	李四	李四	系统开发员	意用	2019-06-24 19:08:45	编辑	<b>股份</b>
	5	微梦	孝木子	文字编辑	類用	2019-07-05 11:18:06	编辑	#599
- Electric								
NTER.								



### 5.2 项目准备

Web项目的开发是一项复杂的系统工程,一般包括系统分析、系统设计、系统实现,以 及系统测试等诸多环节,其中,系统分析与系统设计需要在项目的准备阶段完成。

由于篇幅的限制,这里只简单地介绍一下系统设计,项目的系统分析,例如需求分析、可行性分析、项目计划书的编写等工作,读者可自行参考其他技术文档学习完成。

### 5.2.1 系统初步设计

系统初始设计包括确定系统目标、系统功能、数据库类型,以 及系统开发框架等内容。下面简单介绍项目的创建、静态资源的 准备以及初始化文件设计等内容。

1. 创建 PHP 项目

创建一个 PHP 本地项目,项目名称为"微梦内容管理系统",项目目录为 chap05,对应的虚拟主机为 book.php.chp05。项目目录结构如图 5.6 所示。

该项目分为前台与后台两个模块,其中,前台文件存放在项目 根目录下的 home 子目录中,后台模块文件存放在 admin 目录下。 项目根目录下的 common 与 public 子目录中存放前后台模块的公 ✓ ∰ admin > ( view > P admin\_add.php > P captcha.php > P deleteExc.php > P index.php > P insert\_article.php > P insertExc.php > P login.php > P loginExc.php > P logout.php > P user\_list.php > P welcome.php · ( common > 进 captcha > 🥵 lib > ( error 🗸 🖪 frame P config.php > > P init.php > 3 home 😕 public > I static > 进 upload .buildpath X .project > P index.php

图 5.6 项目目录结构

用文件。项目结构中的目录及文件说明如表 5.1 所示。

属性	说 明
admin	后台模块目录
admin/	后台模块中的页面文件目录
admin/view	后台模块视图文件目录
common	前后台公共文件目录
common/captcha/	图形验证码文件目录
common/lib/	自定义库文件目录
error	错误页面文件目录
frame	系统配置与初始化目录
home	前台模块目录
public	系统公共静态资源目录
Index. php	系统入口文件

表 5.1 项目目录及文件说明

这里只介绍了主要目录与文件,详情请参见源码。

#### 2. 准备公共资源

项目的公共资源,主要包括项目前后台模块使用的公共 CSS 样式文件、JavaScript 脚本 文件、图像文件,以及共同使用的自定义和第三方开发的库文件等。

1) 引入 CSS 和 JS 文件

本项目前端使用 Layui 及 jQuery 前端框架。下载 Layui 和 jQuery 资源,并将其复制到 项目根目录下的 public\static 子目录中。

2) 加载自定义函数库

在项目根目录下的 common 目录中创建子目录 lib,并在其中新建一个名为 function. php 的 PHP 文件。该文件是自定义库文件,存放项目模块的公共函数。

函数库文件 function. php 的加载,一般在项目的初始化文件中完成。下面创建项目的 初始化文件 init. php。

#### 3. 创建初始化文件

项目的初始化文件主要用于完成定义常量、加载资源,以及启动 SESSION 等工作。在项目根目录下的 frame 子目录中,新建 init. php 文件并编写如下代码。

<?php /\*\* \* 项目初始化文件 \* 定义常量、加载资源、启动 SESSION \* @author weiwenping \*/ define('APP\_NAME', '微梦内容管理系统'); define('DOC\_ROOT', \$\_SERVER['DOCUMENT\_ROOT']);

```
define('DOC COMMON', DOC ROOT.'/common/');
define('DOC LIB', DOC COMMON.'lib/');
define('CAPTCHA FONT PATH', DOC COMMON. 'captcha/');
define('CONFIG PATH', DOC ROOT.'/frame/');
//前台文件目录常量
define('HOME ROOT', DOC ROOT.'/home/');
define('HOME VIEW PATH', HOME ROOT. '/view/');
//后台文件目录常量
define('ADMIN ROOT', DOC ROOT.'/admin/');
define('ADMIN VIEW PATH', ADMIN ROOT.'/view/');
//公共资源引用目录常量
define('WEB_ROOT', 'http://book.php.chp05/');
define('WEB_STATIC', WEB_ROOT.'public/static/');
define('LAYUI_PATH', WEB_STATIC.'layui/');
define('LAYUI_CSS_PATH', LAYUI_PATH.'css/');
define('LAYUI_JS_PATH', LAYUI_PATH);
define('JQUERY_PATH', WEB_STATIC.'jquery/');
define('CSS_PATH', WEB_STATIC.'css/');
define('JS PATH', WEB STATIC.'js/');
define('UPLOAD PATH', WEB ROOT.'public/upload/');
//后台模块目录常量
define('WEB_ADMIN', WEB_ROOT.'admin/');
//加载公共资源函数库
require DOC LIB. 'function.php';
//开启 SESSION
session_start();
/ * *
 * 项目启动函数
 * @return string[] unknown[]
 * /
function run(){
    //获取请求的 URI
    $ request uri = $ SERVER['REQUEST URI'];
    //获取请求的入口文件
    $ script_name = $ _SERVER['SCRIPT_NAME'];
    //获取 URL 中的模块、方法和查询字符串
    $ request = str_replace( $ script_name, '', $ request_uri);
    $ request = ltrim( $ request, '/');
    //将查询字符串分离
    $ request_array = explode('?', $ request);
    //将模块和方法存放在数组中
    $ module action = $ request array[0];
    if (empty($ module action)) {
        $ module = 'home';
        $ action = 'index';
    }else{
        $ module_action = explode('/', $ module_action);
        //获取模块
        if (isset( $ module_action[0]) && !empty( $ module_action[0])) {
            $ module = $ module_action[0];
```

```
}else{
             $ module = 'home';
        }
        //获取方法
        if (isset( $ module action[1]) && !empty( $ module action[1])) {
             $ action = $ module action[1];
        }else{
             $ action = 'index';
        }
    }
    //请求的文件
    $ require = DOC_ROOT. '/'. $ module. '/'. $ action. '.php';
    if (file_exists( $ require)) {
        require_once $ require;
    }else{
        require once DOC ROOT. '/error/error.html';
    }
}
```

在系统初始化文件中,除定义一些必要的常量外,还定义了一个名为 run()的函数,该 函数是系统的启动函数,起着一个简单的路由作用。

### 5.2.2 系统流程设计

系统功能确定以后,需要设计系统架构,也就是系统的整体框架。本系统是一个简单的 PHP设计项目,所以项目功能不复杂。本项目系统架构模拟 MVC 设计模式。

下面通过测试页面,对项目前台模块执行流程进行设计。首先设计系统入口文件。在项目根目录下创建 index. php 文件,代码如下。

```
<?php
/**
* 系统人口文件
* @author weiwenping
*/
//加载项目初始化文件
require './frame/init.php';
//启动项目
run();
```

接着,在前台模块 home 目录中新建测试文件,分别为 test. php 和 view\test. php,代码如下。

```
<?php
defined('APP_NAME') or exit('非法访问');
/**
* 前台测试文件 test.php
*/
$ data['test'] = array(
[
```

```
'id' => '1'.
      'name' => '面向对象程序设计(C++语言描述)',
      'author' => '马石安 魏文平',
      'press' => '清华大学出版社'
   ],
   ...
);
//加载视图
require HOME VIEW_PATH. 'v_test.php';
<!-- 前台测试视图文件 home\view\test.php -->
<! DOCTYPE html >
< html lang = "en">
< head >
   <meta charset = "UTF - 8">
   <title>前台测试页面</title>
   <link rel = "stylesheet" href = "<?php echo LAYUI CSS PATH; ?> layui.css" />
</head>
< body class = "layui - body">
   <h2>前台测试页面</h2>
   < hr >
   <div class = "layui - table">
   < thead >
         序号教材名称作者出版社
      </thead>
      <?php foreach( $ data['test'] as $ v): ?>
         <?php echo $v['author'];?><?php echo $v['press'];?>
         \langle /tr \rangle
         <?php endforeach; ?>
      </div>
</body>
</html>
```

打开浏览器,在地址栏中输入"http://book.php.chp05/index.php/home/test",运行 效果如图 5.7 所示。

在上述 URL 中,"index. php"表示系统入口文件;"home"表示系统前台模块;"test"表示需要访问的前台文件。如果与 MVC 模式项目进行类比,可以看出,前台模块中的 PHP 文件相当于 MVC 的控制器;视图目录 view 中的 PHP 文件相当于 MVC 的视图;而控制器中的数据准备则相当于 MVC 的模型。

所以,本系统的访问流程确定为:系统域名→入口文件→模块→页面文件→页面视图。

在上述测试页面中,测试数据是通过数组的方式直接准备的,在后续的开发过程中,页 面中的数据要从数据库中通过"模型"来获取。

前台灣這页面	× +												σ	
(←) → ℃ @	0 🔏 boo	k.php.chp05/index.php/home/test			飘			Q. 搜索	hi/\	4	۵	•	1° ;	» =
◆ 最常访问 □ 常用用址												08	动设备	上的书册
	前台测	试页面												
	序号	数材名称	作者	出版社										
	1	面向对象程序设计(C++语言描述)	马石安 魏文平	清华大学出版社										
	2	Visual C++程序设计与应用数程	马石安魏文平	清华大学出版社										
	3	PHP Web程序设计与项目案例开发	马石安魏文平	清华大学出版社										
	4	PHP项目实践与课程设计指导	马石安 戰文平	清华大学出版社										

图 5.7 前台测试页面运行效果

### 5.2.3 数据库设计

数据库设计是 PHP Web 项目开发的重要环节,数据库设计的质量直接影响着系统运行的效率与速度。

为项目创建一个名为 wmchap05db 的 MySQL 数据库,并在其中创建数据表,数据表名 均使用"wm\_"前缀。由于篇幅的限制,这里只展示系统管理员数据表、文章数据表和菜单 数据表,其他数据表请参见源码包中的相关资源。

项目管理员数据表 wm\_admin 的结构及字段含义分别如图 5.8 和表 5.2 所示。

Field	Туре	Comment
id	int(10) NOT NULL	
username	varchar(20) NOT NULL	
password	varchar(32) NOT NULL	
truename	varchar(20) NOT NULL	
gid	int(10) NOT NULL	
status	tinyint(1) NOT NULL	
add_time	int(10) NOT NULL	

图 5.8 数据表 wm\_admin 的结构

表 5.2 数据表 wm\_admin 字段说明

字段	说 明
id	序号。主键
username	用户名
password	密码。MD5 加密
truename	用户真实姓名
gid	用户类型分组 ID。对应数据表 tb_admin_groups 中的 GID
status	管理员账号状态。1表示禁用,0表示正常
add_time	注册时间

项目文章数据表 wm\_article 的结构及字段含义分别如图 5.9 和表 5.3 所示。

	Field	Туре	Comment
4	id	int(4) unsigned NOT NULL	主键
	title	varchar(255) NOT NULL	文档标题
	title_img	varchar(200) NOT NULL	标题图片
	is_hot	tinyint(4) unsigned NOT NULL	是否热门1是0否
	is_top	tinyint(4) unsigned NOT NULL	是否置顶1是0否
	<pre>is_hot is_top cate_id user_id content pv</pre>	int(10) NOT NULL	栏目主键
		int(10) NOT NULL	用户主键
	content	text NOT NULL	文档内容
	pv	int(10) NOT NULL	阅读量
	status	tinyint(4) NOT NULL	状态1显示0隐藏
	create_time	int(10) NOT NULL	创建时间
	update_time	int(10) NOT NULL	更新时间

图 5.9 数据表 wm\_article 的结构

表 5.3	数据表	wm	article	字段说明
	20 NH - 10			1 12 00.00

字段	说  明
id	序号,主键
title	文章标题
title_img	文章标题图片,在前台主页列表显示时使用,如图 5.1 所示
is_hot	是否是热门文章
is_top	是否将文章置顶
cate_id	文章类型 ID
user_id	作者 ID
content	文章内容
pv	文章浏览次数
status	文章显示状态
create_time	文章提交时间
update_time	文章修改时间

项目菜单数据表 wm\_menu 的结构及字段含义分别如图 5.10 和表 5.4 所示。

Field	Туре	Comment
id	int(10) unsigned NOT NULL	
name	varchar(50) NOT NULL	
href	varchar(100) NOT NULL	
position	varchar(30) NOT NULL	
pid	tinyint(1) NOT NULL	父菜单ID
status	tinyint(1) NOT NULL	1正常2禁用
module	varchar(20) NOT NULL	
order	tinyint(1) NOT NULL	

图 5.10 数据表 wm\_menu 的结构

字 段	说 明
 id	序号。主键
name	菜单名称
href	菜单 URL
position	菜单位置
pid	菜单的父菜单 ID
status	菜单状态
order	菜单排列顺序

表 5.4 数据表 wm\_menu 字段说明

上述对数据库的操作与管理,使用的是 SQLyog 工具软件。请读者选择自己熟悉的数据库管理工具进行操作。

### 5.3 后台功能实现

Web项目的开发往往都是从后台开发开始的。下面实现项目后台管理系统的部分功能,包括管理员信息显示,管理员登录验证,管理员添加、编辑与删除;内容分类管理、内容详情管理等。

### 5.3.1 主页设计

系统后台主页运行效果如图 5.4 所示。

#### 1. 后台布局

系统后台模块主要负责系统数据的管理,例如,管理员的添加、删除、信息修改,以及信息显示;管理员权限设置;内容的添加、编辑、删除、审核等。

把后台模块的所有操作集中到页面左侧的菜单,操作内容则在页面的主显示区域中进行显示,如图 5.5 所示。

单击页面中侧边栏"用户管理"菜单下的"用户列表"子菜单,则会在页面的主显示区内, 显示所有的系统用户信息,同时显示各种操作按钮。

单击页面左上角的主页图标及文本,可以重新加载后台模块封面页面;单击页面右上 角的"退出系统",则可以退出系统的后台管理,如图 5.11 所示。

#### 2. 侧边栏设计

侧边栏中的导航菜单信息不是直接固定在页面代码中的,它们来自数据库中的 wm\_ menus 数据表。

1) 数据库操作

在系统自定义库文件目录 common\lib 中新建 db. php 文件,在该文件中编写数据库连接、数据表查询等函数,代码如下。



后台管理系统	× +									
↔ ở ở ở ở ##68 🗁 #894±	Ø & book.php.chp05/index	.php/admin/welcome	22	&	Q, 推索	III.	t,	• •	ా లో [8ల	>> ∃ <sup>d</sup> Q∰上的书题
<ul> <li></li></ul>	馏							后台主	<sub>ॼ</sub> ɗ	<u>ב</u>
() 使用说	朝	微梦后台 <sub>开发者:马</sub>	管理系统	5				忘记密	弱く	Ð
() 技术支	2持	2020 4	¥1月				ŝ	申请注	<del>т</del> (	Ð
(;;) 关于那	转门							系统前	6 (	D

图 5.11 系统后台主页

```
<?php
/ * *
 * 数据库连接
 * @return NULL|mixed
 * /
function db_connect() {
    static $ link = null;
    if (! $ link) {
        $ link = call_user_func_array('mysqli_connect', getConfigItem('DB_CONNECT'));
        if (! $ link) {
            exit('数据库连接错误');
        }
    }
   return $ link;
}
/ * *
 * 数据查询
 * @param string $ sql
 * @param string $ type
 * @param array $ data
 * @return object
 * /
function db_query( $ sql, $ type = '', $ data = []) {
    $ link = db connect();
    $ stmt = mysqli_prepare($link, $sql);
    if (! $ stmt) {
        $ error = mysqli_error($link).'<br>SQL语句:'.$sql;
        exit('数据库操作错误!'.$error);
    }
    if ($ data == []) {
```

```
mysqli_stmt_execute( $ stmt);
    }else{
        $ data = (array) $ data;
        db bind param($ stmt, $ type, $ data);
        mysqli stmt execute($ stmt);
    }
    return $ stmt;
}
/ * *
 * 查询数据绑定
 * @param object $ stmt
 * @param string $ tyle
 * @param array $ data
 * /
function db_bind_param($ stmt, $ tyle, &$ data) {
    $ params = [$ stmt, $ tyle];
    foreach ($ data as & $ params[]) { }
    call_user_func_array('mysqli_stmt_bind_param', $ params);
}
define('DB_ALL', 0);
define('DB_ROW', 1);
define('DB_COLUMN', 2);
define('DB AFFECTED', 3);
define('DB_LASTED', 4);
/ * *
 * 查询结果处理
 * @param int $ mode
 * @param string $ sql
 * @param string $ type
 * @param array $ data
 * @return array
 * /
function db_fetch($mode, $sql, $type='', $data=[]) {
    $ stmt = db_query( $ sql, $ type, $ data);
    $ result = mysqli_stmt_get_result( $ stmt);
    switch ($ mode) {
        case DB_ROW:
            return mysql;i_fetch_assoc( $ result);
        break;
        case DB COLUMN:
            return current((array)mysqli fetch row($ result));
        break;
        default:
            return mysqli_fetch_all( $ result, MYSQLI_ASSOC);
        break;
    }
}
/ * *
 * 数据查询,没有结果集的查询
```

```
* @param int $ mode
 * @param string $ sql
 * @param string $ type
 * @param array $ data
 * @return int
 * /
function db_exec($ mode, $ sql, $ type = '', $ data = []) {
    $ stmt = db guery( $ sql, $ type, $ data);
    switch ($ mode) {
        case DB LASTED:
            return mysqli_stmt_insert_id( $ stmt);
            break;
        default:
            return mysgli stmt affected rows($ stmt);
            break;
    }
}
function db_connect() {
    static $ link = null;
    if (! $ link) {
         $ link = call_user_func_array('mysqli_connect', getConfigItem('DB_CONNECT'));
        if (! $ link) {
            exit('数据库连接错误');
        }
    }
    return $ link;
}
```

上述代码中的 getConfigItem()是自定义函数,用于获取配置文件中的数据库连接信息。数据库连接数据存放在系统根目录下的 frame 子目录中,文件名为 config. php。

数据库操作文件 db. php 中的功能函数,由于代码都比较简单,这里不再详细说明,请使用源码自行学习。

2) 获取菜单数据

在系统 common\lib\function.php 文件中,编写菜单处理函数。其中,getMenus()函数 返回某个模块、某个位置中的全部菜单;getSubMenus()函数返回某个菜单项的二级菜单。 为了简单一些,本项目菜单只分为二级,代码如下。

```
function getMenus($ module, $ position) {
    //从数据库中获取菜单信息
    $ sql = "select * from wm_menus where module = '".
        $ module."'and position = '". $ position."'"
            .'order by order';
    $ menus = db_fetch(DB_ALL, $ sql);
    return $ menus ? $ menus : null;
}
function getSubMenus($ menus, $ menus_id) {
    $ subMenus = array();
}
```

```
foreach ($menus as $v) {
        if ( $ v[ 'pid'] == $ menus id) {
            array_unshift($ subMenus, $ v);
        }
    }
    return $ subMenus ? $ subMenus : null;
}
在后台页面的 PHP 文件中,调用上述方法,获取菜单数据,代码如下。
//获取侧边栏导航菜单
$ side navs = getMenus('admin', 'side');
foreach (\$ side navs as \$ k => \$ v) {
    $ data['side_navs'][ $ k] = $ v;
    $ data['side_navs'][ $ k]['sub'] = getSubMenus( $ side_navs, $ v['id']);
}
最后,设计侧边栏视图。文件为 admin\view\side. php,代码如下。
<?php defined('APP NAME') or exit('非法访问');?>
< div class = "layui - collapse" lay - accordion >
    <?php foreach ($ data['side navs'] as $ v):?>
    <?php if( $ v[ 'pid'] == 0):?>
    < div class = "layui - colla - item" style = "padding - left:10px;">
        <h2 class = "layui - colla - title"><? = $ v[ 'name' ]?></h2>
        < div class = "layui - colla - content">
            <?php if( $ v['sub'] === null): ?>
                < span style = "padding:5px;">< a href = "<? = $ v[ 'href']?>">管理主页</a><///a><///><///>
span >
            <?php else: ?>
            <?php foreach( $ v['sub'] as $ sub): ?>
                style = "padding:5px; border - bottom: 1px dashed #FF5722;">
                    <a href = "<? = $ sub[ 'href']?>"><? = $ sub[ 'name']?></a>
                <?php endforeach; ?>
            <?php endif;?>
        </div >
    </div>
    <?php endif;?>
    <?php endforeach;?>
</div>
```

侧边栏运行效果如图 5.12 所示。

这里使用了 Layui 的面板组件,单击一级菜单名称,展开二级菜单列表;单击二级菜单,在页面右侧的区域显示主内容。

#### 3. 后台主页设计

用户访问系统后台主页,就是加载系统的 admin\index. php 文件,该文件代码如下。

~	用户管理
汤	动用户
月	沪列表
>	分类管理
>	文章管理
>	系统管理
>	管理员管理
>	菜单管理

图 5.12 后台页面中的侧边栏

```
<?php
defined('APP_NAME') or exit('非法访问');
/ * *
 * 系统后台首页文件
 * admin\index.php
 * /
//视图数据
$ data = array(
    'title'=>'后台首页',
    'css_files' = > array(
        LAYUI_CSS_PATH. 'layui.css',
        CSS_PATH. 'admin_index.css',
    ),
    'js_files' => array(
        JQUERY PATH. 'jquery - 3.3.1.min.js',
        LAYUI_JS_PATH. 'layui.js',
),
...
);
//加载视图
require ADMIN_VIEW_PATH. 'v_index.php';
```

接着,编写视图文件 admin\view\v\_index. php,代码如下。

```
<?php include ADMIN VIEW PATH. 'side.php';?>
        </div>
        < div class = "layui - col - md9 content">
             <h3>系统后台首页</h3>
        </div>
    </div>
    < div class = "lavui - row lavui - bg - cvan footer">
        <?php include ADMIN VIEW PATH. 'bottom.php';?>
    </div>
</div>
<?php include ADMIN_VIEW_PATH. 'footer.php';?>
< script >
    $ (document).ready(function() {
        layui.use('element', function(){
            var element = layui.element;
        });
    });
</script>
```

视图由公共部分和特别部分组成,公共部分文件也是存放在 admin\view 目录中,各部分代码详见源码。

#### 4. 后台封面设计

为系统后台设计一个封面页面,用于承载系统介绍、使用说明、用户登录,以及广告信息 等内容,如图 5.11 所示。

在系统 admin 目录下新建 welcome. php 文件,代码如下。

```
<?php
defined('APP NAME') or exit('非法访问');
/ * *
 * 系统后台欢迎页面
 * admin/welcome.php
 * /
//视图数据
$ data = array(
    'title' =>'后台管理系统',
    'css files' => array(
       LAYUI CSS PATH. 'layui.css',
       CSS PATH. 'admin welcome.css',
    ),
    'js_files' = > array(
       JQUERY_PATH. 'jquery - 3.3.1.min.js',
       LAYUI_JS_PATH. 'layui. js',
    ),
    'content' => array(
        'title' => '微梦后台管理系统',
        'author' => '开发者: 马石安 魏文平',
        'time' => '2020 年 1 月 '
    ),
```

);

「粝讲飯

```
//加载视图
```

require ADMIN\_VIEW\_PATH. 'v\_welcome.php';

在 admin/view 目录中,新建视图文件 v\_welcome. php。该视图文件非常简单,请参考项目源码。

### 5.3.2 登录与登出

用户的登录与登出,是任何 Web 项目都必须具有的功能。用户登录功能实现时,重点 注意用户输入数据的验证。

#### 1. 登录表单设计

1) 创建页面

在系统后台模块中,添加管理员登录视图文件 admin\view\v\_login.php,页面效果如图 5.13 所示。

微梦后台管理系统·管理员登录 ×							-	٥	×
← → C* @ 0 & book.php.chp	05/index.php/admin/login	※ … ☆	Q, 搜索	III V	4	۰ ۵	•	(8) ≫ (200040 ⊨)	≡́
								⊕-W/00MIL	171022
	管理员登录								
	用户名								
	10 B)								
	验证码	9C <sup>3L</sup> Q							
	\$2.02 \$2.02								
	_								

图 5.13 后台用户登录表单

系统后台管理员登录页面,通过单击图 5.11 页面中的"用户登录"超级链接来实现。加载该页面的 PHP 文件为 admin\login.php,其代码与系统后台模块中的其他 PHP 文件相 似,请参见源码。

在如图 5.13 所示的页面中有一个图形验证码,用于加强用户登录验证。下面实现验证 码图片的显示,以及单击验证码图片更新验证码的功能。

2) 生成验证码

在项目根目录下的 common\lib 子目录中添加 captcha. php 文件,用于存放生成图形验

证码、输出图形验证码、检验验证码是否正确的函数,代码如下。

```
<?php
/ * *
 * 生成验证码
 * @param int $ count 验证码长度
 * @return string
 * /
function create captcha( $ count = 5) {
    $ code = '';
    $ charset = 'ABCDEFGHJKLMNPORSTUVWXY23456789';
    \ len = strlen(\ charset) - 1;
   for (\$i = 0; \$i < \$count; \$i++) {
        $ code . = $ charset[mt rand(0, $ len)];
   }
   return $ code;
}
/ * *
 * 显示图形验证码
 * @param string $ code
 * /
function captcha img($ code) {
    \$ width = 120;
                                                       //验证码图片宽度
    $ height = 35;
                                                       //验证码图片高度
    $ img = imagecreate($width, $height);
                                                       //创建验证码图像
   //设置验证码图像的背景颜色
   imagecolorallocate($ img, mt rand(50,255), mt rand(0,155), mt rand(0,155));
    $ fontSize = 18;
                                                       //验证码文字大小
    $ fontColor = imagecolorallocate($ img, 255, 255, 255); //验证码文字颜色
    $ fontStyle = CAPTCHA FONT PATH. 'font. TTF';
                                                       //验证码文字样式
    $ len = strlen($ code);
   //生成指定长度的验证码
   for($i=0; $i<$len; ++ $i){</pre>
       imagettftext(
           $ img,
           $ fontSize,
           mt_rand(0,20) - mt_rand(0,25),
                                                       //设置验证码文字倾斜角度
           //随机设置验证码文字显示坐标
           $ fontSize * $ i + 12,
           mt rand($ height/2, $ height),
           $ fontColor,
           $ fontStyle,
           $code[$i]
        );
   }
   //为验证码图片生成彩色噪点
   for($i=0; $i<200; ++ $i){</pre>
       //随机生成颜色
        $ color = imagecolorallocate($ img, mt rand(0, 255), mt rand(0, 255), mt rand(0,
255));
       //随机绘制干扰点
```

```
imagesetpixel($ img, mt rand(0, $ width), mt rand(0, $ height), $ color);
    }
   //绘制 10 条干扰线
   for($i=0; $i<5; ++$i){
       //随机生成干扰线颜色
        $ color = imagecolorallocate($ img, mt rand(0, 255), mt rand(0, 255), mt rand(0,
255));
        //随机绘制干扰线 imageline($ img, mt rand(0, $ width), 0, mt rand(0, $ width),
$ height, $ color);
    }
    header('Content - Type: image/png');
                                                          //输出图像
    imagepng($ img);
    imagedestroy($ img);
                                                          //释放内存
}
/ * *
 * 检查验证码是否正确
 * @param string $ captcha
 * @return boolean
 * /
function captcha check($ captcha) {
    if ($ captcha == $ SESSION['wmcms']['captcha']) {
        return true;
    }
   return false;
}
```

验证码需要以某种样式显示,所以需要扩展名为".ttf"的字体样式文件。该文件可以 在操作系统的字体库中复制。这里将其存放在项目 common\captcha 子目录中。

上述代码生成了图形验证码,下面将其在用户登录页面中显示出来。在项目的 admin 目录下,新建一个名为 captcha. php 文件,代码如下。

```
<?php
/**
* 生成图形验证码并输出
* @author weiwenping
*/
//加载验证码函数文件
require DOC_LIB. 'captcha.php';
//生成验证码
$ code = create_captcha();
//输出验证码图像
captcha_img($ code);
//将验证码保存到 SESSION 中
$_SESSION['wmcms']['captcha'] = $ code;
最后,修改登录表单中的验证码图片的 HTML 元素属性,代码如下。
```

```
< div class = "layui - form - item">
    < label for = "captcha" class = "layui - form - label">验证码</label >

< div class = "layui - input - inline">

< input type = "text" name = "captcha" id = "captcha" class = "layui - input">
```

```
</div>
<img src = "captcha " id = "captchaImg" />
</div>
```

3) 验证码更新

在用户输入登录数据时,页面上的验证码有时会看不清楚,这时要让用户单击验证码图 片来进行重置。这个功能可以用 jQuery 轻松实现,代码如下。

```
//单击图形验证码进行重置
$ ('#captchaImg').click(function(event) {
    $ (this).attr('src', 'captcha?rand = ' + Math.random());
});
```

代码中的"captchaImg"是显示验证码图片的 img 标签 ID。

#### 2. 测试数据准备

为了测试用户登录功能,需要在数据库中添加一些测试数据。登录 MySQL 数据库服务器,在项目数据库 wmchapt05db 的 wm\_admin 数据表中插入测试数据。例如,用户名 admin,用户密码 md5('123456')等。

#### 3. 数据验证

表单数据的验证分为前端验证与后端验证。前端验证就是在页面上通过 JavaScript 代码进行验证,而后端验证是通过 PHP 与数据库的交互来对用户输入数据进行检验。

1) 前端验证

新建 public\static\js\admin\_login.js 文件,并在文件中添加 onlogin()函数,用于处理 表单提交,代码如下。

```
//处理登录表单提交
function onlogin() {
   //获取表单数据
   var username = $.trim($('#username').val());
   var password = $.trim($('#password').val());
   var captcha = $.trim($('#captcha').val());
   //检查输入框是否为空
   if (username == '') {
       layer.alert('请输入用户名!', {'icon': 2 });
       return;
   }
   if (password == '') {
       layer.alert('请输入密码!', { 'icon': 2 });
       return;
   }
   if (captcha == '') {
       layer.alert('请输入验证码!', { 'icon': 2 });
       return;
   }
   //提交表单数据
```

```
$.post(
        $ ('form').first().attr('action'),
        { 'username': username, 'password': password, 'captcha': captcha },
       function(res) {
           if (res.code > 0) {
               //重置验证码
                $ ('#captcha').val('');
               reloadImg();
               //错误信息提示
                layer.alert(res.msg, { icon: 2 });
            } else {
               //登录成功信息提示
                layer.msg(res.msg);
               //1 秒后跳转到后台主页
                setTimeout(function() {
                       window.location.href = 'index'
                },
               1000);
            }
        }, 'json');
}
```

这里只简单地验证表单数据是否为空。其他验证,如用户名长度不能少于3位,密码长 度不能少于6位等,请自行实现。页面效果如图5.14所示。

微梦后台管理系统 · 管理员登录 × +					- ø ×
← → で 金 ● 凝 book.php.ch ◆ 最新時間 第期時址	p05/index.php/admin/login		調 … ☆ Q 推測	W # 9 0	। ় ৩° ≫ ⊟ <sup>6</sup> ⊡≋জর#⊥চার≣
	管理员登录	_			
	用户名		1. C. 1.		
	an 44 a	an ×			
	1998 () 1997 ()	X) 清输入用户名!	RH		
		of the			

#### 图 5.14 表单前端验证

图 5.14 是用户直接单击"登录"按钮后的效果。用户单击"登录"按钮后,页面中的 jQuery 代码会调用上述的 onlogin()函数,代码如下。

//用户单击"登录"按钮,提交数据登录

```
$ ('# submit').click(function(event) {
    //处理表单提交
    onlogin();
});
```

注意:由于表单的提交按钮"登录"位于 form 标签内,需要阻止该按钮的默认表单提交事件发生。

2) 后端验证

在前端的验证中,只是验证了用户输入的验证码是否为空,并没有判断它的正确性。下 面在后端编写代码,检查用户输入的验证码是否正确。

在系统后台模块中添加 admin \login Exc. php 文件,并编写代码。

```
<?php
/ * *
 * 微梦后台管理系统
 * 管理员登录处理
 * @author weiwenping
 * /
//加载验证码函数文件
require once DOC LIB. 'captcha.php';
//接收表单数据
$ username = trim( $ _POST[ 'username']);
$ password = trim( $ _POST[ 'password']);
$ captcha = trim( $ POST['captcha']);
//验证输入是否为空
if ($username == '') {
   exit(json_encode(array('code' = > 1, 'msg' = >'用户名不能为空!')));
}
if ($ password == '') {
   exit(json encode(array('code'=>1,'msg'=>'密码不能为空!')));
}
if ($ captcha == '') {
   exit(json_encode(array('code'=>1,'msg'=>'验证码不能为空!')));
}
//验证验证码
if (!captcha check(strtoupper($captcha))) {
   exit(json encode(array('code'=>1,'msg'=>'验证码错误!')));
}
页面效果如图 5.15 所示。
```

后端验证码检查完毕后,接着进行用户输入数据的后端验证,也就是与数据库中的数据 进行比对。

在 admin\loginExc. php 文件中添加代码,完成数据库验证,代码如下。

```
//数据库验证,用户验证
$ sql = 'select * from wm_admin where username = ?';
$ admin = db_fetch(DB_ROW, $ sql, 's', array('username' => htmlentities($ username)));
//用户验证
if (!$ admin) {
```



微梦后台管理系统 · 管理员叠梁 🛛 🗙 🕂			- ø ×
← → ご 企 ② 差 book.php.chp05/m ◇ 最新活用 世界用料止	idex.php/admin/login	読 … ☆ ① 読素	■ 11、12、19 00 今 30° > 三 <sup>4</sup> ① \$#\$\$\$\$\$#_b\$#35
			C wastern Thy DE
	管理员登录		
	用户名 888888		
	me ×		
		Koo	
		12×117	
	42		
		_	

图 5.15 验证码后端验证

```
exit(json_encode(array('code' = > 1, 'msg' = >'该用户不存在!')));
}
//验证密码
if (md5($ admin['username'].$ password) != $ admin['password']) {
    exit(json_encode(array('code' = > 1, 'msg' = >'密码错误!')));
}
//账号是否被禁用
if ($ admin['status'] == 1) {
    exit(json_encode(array('code' = > 1, 'msg' = >'该用户已被禁用!')));
}
//设置 SESSION
$_SESSION
$_SESSION['wmcms']['admin'] = $ admin;
//登录成功返回信息
exit(json_encode(array('code' = > 0, 'msg' = >'恭喜,登录成功!')));
```

测试效果如图 5.16 所示。

图中展示的是"用户名"输入错误时的返回信息。注意,这里表单提交采用的是 AJAX 请求方式,详细代码请参见源码。

#### 4. 管理员退出系统

管理员退出系统功能的实现非常简单,只需要删除 SESSION 中的管理员数据即可。 首先,在页面中添加代码,监听用户是否单击了"退出系统"按钮,代码如下。

```
//退出系统处理
$ ('#logout').click(function(event) {
    var url = $(this).attr('href');
    layer.confirm('您确定要退出吗?', {
        btn: ['确定', '取消'],
        icon: 3,
```



图 5.16 用户登录数据库验证

});

上述代码中的"logout"是"退出系统"按钮标签的 ID,当用户单击提示框中的"确定"按钮时,调用 onlogout()函数完成用户退出功能。onlogout()函数代码如下。

```
function onlogout(url) {
    $ .post(url, {},
    function(res) {
        if (res.code > 0) {
            layer.alert(res.msg, { icon: 2 });
        } else {
            layer.msg(res.msg);
            setTimeout(function() {
                window.location.href = 'index';
            },
            1000);
        }
    }, 'json');
}
```

从上述代码可以看出,用户退出系统的处理逻辑是在 admin\logout.php 文件中实现 的。该文件代码如下。

<?php /\*\* \* 管理员退出系统处理

```
* @author weiwenping
*/
//退出系统
if (isset($_SESSION['wmcms']['admin'])) {
    unset($_SESSION['wmcms']['admin']);
}
//返回提示信息
exit(json encode(array('code'=>0,'msg'=>'您已退出后台管理系统!')));
```

从上述代码可以看出,用户退出系统时,直接删除 SESSION 中的用户数据即可。用户 退出系统后,不能再访问系统后台内页,需要对用户权限进行控制。

在自定义函数库 common/lib/function. php 中,新建函数 privilege(),代码如下。

```
//用户权限控制
function privilege() {
    $ res = false;
    //用户是否登录
    $ res = isset($_SESSION['wmcms']['user'])? true : false;
    return $ res;
}
```

在系统后台主页 admin\index. php 中添加代码,进行用户访问控制,代码如下。

```
//判断用户是否有权限访问
if (!privilege()) {
    header('Location: admin/welcome');
    exit;
}
```

如果用户没有访问权限,跳转到系统后台欢迎页面。

#### ■ 約1回 ● 約1回 ● 約5回 初5回

5.3.3 用户信息管理

系统用户分为系统管理员和普通用户,系统管理员可以登录到系统后台,而普通用户只 能使用系统前台功能。

用户信息管理,包括用户信息列表显示、用户的添加/编辑和删除、用户权限管理等。下 面以管理员用户信息管理为例来进行介绍。

#### 1. 信息显示

要对系统用户进行管理,首先必须将所有的用户信息在页面中显示出来,然后再针对不同的用户账号进行相应的操作。用户信息列表如图 5.5 所示。

页面中显示的管理员信息来自数据表 wm\_admin 和 wm\_admin\_groups。数据表 wm\_admin\_groups 中存储系统管理员的分组,不同组管理员具有不同的操作权限。

在系统后台模块中添加 admin\user\_list. php 文件,在文件中编写代码从数据库获取数据,代码如下。

<?php

```
...
//从数据库中读取全部管理员信息
$ sql = 'select * from wm_admin';
$ user_lists = db_fetch(DB_ALL, $ sql);
//将管理员的 GID 转换成名称
$ sql = 'select * from wm_admin_groups';
$ roles = db_fetch(DB_ALL, $ sql);
foreach ($ user_lists as $ key => $ v) {
    foreach ($ user_lists as $ key => $ v) {
        if ($ v['gid'] == $ vr['gid']) {
            $ user_lists[$ key]['role'] = $ vr['title'];
        }
    }
}...
```

管理员数据表 wm\_admin 中的分组 gid 记录的是数据表 wm\_admin\_group 中的序号 gid,它是一个整型数据,所以必须将其替换成相应的名称,如系统管理员、开发人员、文章编辑等。

最后,编写视图文件 admin\view\v\_user\_list. php 代码。其中数据循环输出代码如下。

```
<?php foreach ( $ data['user_lists'] as $ v): ?>
<?php echo $v['id']?>
   <?php echo $v['username']?>
   <?php echo $v['truename']?>
   <?php echo $v['role']?>
   <?php echo $v['status']? '< span style = "color:red">禁用< span >': '正常'; ?>
   <?php echo date('Y - m - d H:i:s', $v['add_time'])?>
   <br/>
<button class = "layui - btn layui - btn - xs" onclick = "edit()">编辑</button >
       < button class = "layui - btn layui - btn - danger layui - btn - xs" onclick = "del()">删
除</button>
   <?php endforeach;?>
完整代码请参见源码。
```

2. 添加管理员

管理员的添加功能,通过单击如图 5.5 所示页面中的"添加"按钮来实现。表单页面如 图 5.17 所示。

该表单页面文件为 admin\view\v\_admin\_add. php,视图通过 Layui 的弹窗来渲染。在文件 admin\view\v\_user\_list. php 中添加 JS 代码。

```
//处理添加事件
$ ('#add').click(function(event) {
        add();
});
```

→ C 企 0 ≦ bookphp.d 最繁活明 田東用時社	hp05/index.php/adm	iin/user_list	题 … 公	Q, 搜索	08089251 N 5 9 0 5 5 %
☆ 微梦后台管理系统 - 管理	眼信息列表	<b>当前位置:</b> 后台 / 主页 / 用户列表	現在是: 2020年01,	月11日 11时03分	当前在线: 健梦 退出系统
> 用户管理	管理部署	添加管理员	×		##
> 分类管理	ю	用户名		statuat inij	10/1
> 文章管理	1	角色 请选择角色		1970-01-03 04 53 41	
> 系统管理	2	1017147		1970-09-29 19 46 29	
> 管理员管理	3	224948		2019-06-24 19:08:02	955 805
> 菜单管理	4	真实姓名		2019-06-24 19:08:45	900 BDD
	5	状态		2019-07-05 11:18:06	
		添加管理员			
				100	

图 5.17 添加管理员表单

```
/**
 * 加载管理员添加表单页面
 */
function add() {
    layer.open({
        type: 2,
        title: '添加管理员',
        shade: 0.3,
        area: ['480px', '420px'],
        content: 'admin_add?rand = ' + Math.random(),
    });
}
编写上述表单页面的 PHP 文件 admin\admin_add.php 代码。
```

```
<?php
…
//从数据库中读取管理员分组信息
$sql = 'select * from tb_admin_groups';
$data['role'] = db_fetch(DB_ALL, $sql);
…
//加载视图
require ADMIN_VIEW_PATH.'v_admin_add.php';
```

处理表单提交,在 admin/view/v\_admin\_add. php 文件中添加 JS 代码,如下所示。

```
//使用户名输入框获得输入焦点
$('input[name = "username"]').focus();
//处理表单提交
$('#submit').click(function(event) {
    add_submit($('form').first().attr('action'));
});
```

```
function add submit(action) {
    var username = $.trim($('input[name = "username"]').val());
    var password = $.trim($('input[name = "password"]').val());
    var gid = $.trim($('select[name = "gid"]').val());
    var truename = $.trim($('input[name = "truename"]').val());
    if (username == '') {
        layer.alert('请输入用户名', { icon: 2 });
        return;
    }
    if (gid == '0') {
        layer.alert('请选择角色', { icon: 2 });
        return;
    }
    if (password == '') {
        layer.alert('请输入密码', { icon: 2 });
        return;
    }
    if (truename == '') {
        layer.alert('请输入真实姓名', { icon: 2 });
        return;
    }
    $.post(action, $('form').serialize(), function(res) {
        if (res.code > 0) {
            layer.alert(res.msg, { icon: 2 });
        } else {
            layer.msg(res.msg);
            setTimeout(function() {
                parent.window.location.reload();
            }, 1000);
        }
    }, 'json');
```

```
}
```

```
编写表单提交 PHP 处理文件 admin\insertExc. php,代码如下。
```

<?php

```
...
//获取表单数据
$ data['username'] = trim(htmlspecialchars( $ POST['username']));
$ password = trim(htmlspecialchars( $ POST['password']));
$ data['truename'] = trim(htmlspecialchars( $ _POST['truename']));
$ data['gid'] = (int)trim(htmlspecialchars($_POST['gid']));
$ data['status'] = isset($ POST['status']) ? (int)trim(htmlspecialchars($ POST['status']))
: 0;
$ data['add time'] = time();
//数据非空验证
if (! $ data['username']) {
    exit(json_encode(array('code'=>1,'msg'=>'用户名不能为空!')));
}
if (! $ password) {
    exit(json encode(array('code'=>1,'msg'=>'密码不能为空!')));
}
if (! $ data['truename']) {
```

```
exit(json_encode(array('code' = > 1, 'msg' = >'真实姓名不能为空!')));
}
if (! $ data['gid']) {
   exit(json encode(array('code'=>1,'msg'=>'角色不能为空!')));
}
//检查用户名是否已被注册
$ sql = 'select username from wm_admin where username = ?';
$ itme = db_fetch(DB_ROW, $ sql, 's', $ data['username']);
if ($ itme) {
   exit(json encode(array('code'=>1,'msg'=>'用户名已经被注册!')));;
}
//密码加密
$ data['password'] = md5( $ data['username']. $ password);
//完成数据插入
$ sql = "INSERT INTO wm _ admin (username, truename, gid, status, add _ time, password)
VALUES (?,?,?,?,?)";
$ id = db_exec(DB_LASTED, $ sql, 'ssisis', array_values($ data));
if ($id) {
   exit(json encode(array('code'=>0,'msg'=>'管理员添加成功!')));
}else{
    exit(json encode(array('code'=>1,'msg'=>'管理员添加失败!')));
}
```

#### 管理员添加页面效果,如图 5.18 所示。

|                        |  |   | a tradition of                 |  |  | 添加   |
|------------------------|--|---|--------------------------------|--|--|--|
| ID                     | 用户名  | 真实姓名                                      | /形/山昌/建议                       |  | ^  | 操作   |
| 1                      | admin                                      | 微梦  | 用户名                            | 微梦   |  | 编辑 删法  |
| 2                      | admin2                                     | 微梦2                                       | 角色                             | 文字编辑   | ÷.   | 2010 BISS  |
| 3                      | Ξ-   | ΞΞ  | 密码框                            | 123456   |  | tite and the second sec |
| 4                      | 李四   | 李四  |                                |  |  | 编辑 删除  |
|                        |  |   | 具实姓名                           | 李木子  |  |  |
|                        |  |   | 状态                             | *****  |  |  |
|                        |  |   | -1                             |  |  |  |
|                        |  |   |                                | 添加管理员  |  |  |
|                        |  |   |                                |  |  |  |
|                        |  |   |                                |  |  |  |
|                        |  |   |                                |  |  |  |
| 管                      | 理员列表                                       |   |                                |  |  | intek.   |
| 1D                     | 型员列表<br>用户名                                | 真实姓名                                      | 角色                             | 状态   | 添加时间   | ista<br>操作   |
| (図<br>10<br>1          | 理员列表<br>用户名<br>admin                       | 真实姓名                                      | 角色系统管                          | -  | 添加时间<br>1970-01-03 04:53:41  | 法加<br>操作<br>個田 墨林  |
| 管<br>ID<br>1<br>2      | 理员列表<br>用户名<br>admin<br>admin2             | 直实姓名<br>微梦<br>微梦2                         | 角色系统管系统管                       | 状态       埋员     正常       埋员     募用   | 減加時加<br>1970-01-03 04:53:41<br>1970-09-29 19:46:29   | 冷加<br>授作<br>编辑 题称<br>编辑 题称   |
| 管<br>ID<br>1<br>2<br>3 | 理反列表<br>用户名<br>admin<br>admin2<br>王一       | 直实姓名<br>微梦<br>微梦2<br>王二                   | 角色系统管系统管                       | 状态           理员         正常           理员         5月           发员         正常 | 添加时间<br>1970-01-03 04:53:41<br>1970-09-29 19:46:29<br>2019-06-24 19:08:02                        | 法加<br>法付<br>保付:<br>総加 継統<br>総加 継統<br>総加 継統   |
| 1D<br>1<br>2<br>3<br>4 | 理長列表<br>用户名<br>admin<br>admin2<br>王一<br>字四 | 直实姓名       微梦       微梦2       王二       李四 | 角色<br>系统管<br>系统管<br>系统开<br>系统开 | 状态       埋员     正常       埋员     募用       发员     正常                         | 済加時期<br>1970-01-03 04:53:41<br>1970-09-29 19:46:29<br>2019-06-24 19:08:02<br>2019-06-24 19:08:45 |  |

图 5.18 管理员添加页面效果

从图中输出结果可以看出,新管理员被成功插入数据表 wm\_admin 中。这里采用了 AJAX 和 Layui 的弹窗来共同实现管理员的添加功能,以获得最佳的用户体验。

#### 3. 编辑管理员

管理员的编辑功能,通过单击如图 5.5 所示页面中的"编辑"按钮来实现。表单页面如 图 5.19 所示。

| 0 | 田户名    | 南亚纳索  |      |         | <br>揚作          |
|---|--------|-------|------|---------|-----------------|
| 1 | admin  | 御梦    | 用户名  | 编辑测试    | State and State |
| 2 | admin2 | 微梦2   | 角色   | 文字编辑    | Sinte and       |
| 3 | I-     | ΞΞ    | 密码框  |         | \$948 (W.83)    |
| 4 | 李四     | 季四    |      |         | Side Bist       |
| 5 | 微梦     | 李木子   | 具头姓名 | 虚拟管理员   | 1948 BESS       |
| 6 | 编辑测试   | 虚拟管理员 | 状态   | atari 🗸 | Sitte Bills     |

图 5.19 管理员编辑表单

该表单页面其实就是图 5.17 中管理员的添加页面,只是在窗口弹出时,在表单元素中显示了需要编辑的某个管理员的信息。

单击图 5.19 页面中的"编辑"按钮,触发按钮单击事件,调用 edit()函数实现编辑窗口的显示以及数据的初始化。在文件 admin\view\v\_user\_list.php 中,给"编辑"按钮绑定事件,代码如下。

```
<br/>< button class = "layui - btn layui - btn - xs" onclick = "edit(<?php echo $v['id']?>)">编辑</br><br/>button >
```

在 admin/js/admin\_lists. php 文件中,新建 edit()函数,代码如下。

```
function edit(id) {
    layer.open({
        type: 2,
        title: '编辑管理员信息',
        shade: 0.3,
        area: ['480px', '420px'],
        content: 'admin_add?id = ' + id,
    });
}
```

修改管理员添加功能模块的 PHP 文件 admin\admin\_add. php,在加载视图之前获取 需要编辑的管理员信息,代码如下。

```
//获取需要编辑的管理员数据
if (isset($_GET['id']) && (int)$_GET['id'] > 0) {
```

```
//获取需要编辑的管理员信息
$ sql = 'select * from wm_admin where id = ?';
$ admin = db_fetch(DB_ROW, $ sql, 'i', [(int) $ _GET['id']]);
}
```

修改管理员添加功能模块的视图文件 admin\view\v\_admin\_add. php,使其适合添加、 编辑两种情况。详细代码请参见源码。

修改表单提交文件 insertExc. php,使其适合添加、编辑两种情况。完整代码如下。

```
<?php
//获取表单数据
$ id = isset($ POST['id']) ? (int)trim(htmlspecialchars($ POST['id'])) : 0;
$ data['username'] = trim(htmlspecialchars( $ POST['username']));
$ password = trim(htmlspecialchars( $ POST['password']));
$ data['truename'] = trim(htmlspecialchars( $ POST['truename']));
$ data['gid'] = (int)trim(htmlspecialchars($ POST['gid']));
$ data['status'] = isset($ POST['status'])?(int)trim(htmlspecialchars($ POST['status']))
: 0;
$ data['add time'] = time();
//数据非空验证
if (! $ data['username']) {
    exit(json_encode(array('code' = > 1, 'msg' = >'用户名不能为空!')));
}
//添加管理员时判断
if ($ id == 0 && ! $ password) {
    exit(json_encode(array('code' =>1, 'msg' =>'密码不能为空!')));
ļ
//编辑时,若用户没有修改密码,则使用原密码
if ($ id > 0) {
    if (! $ password) {
        $ data['password'] = $ _POST['password1'];
    }else{
        //密码加密
        $ data['password'] = md5( $ data['username']. $ password);
    }
}else{
    //密码加密
    $ data['password'] = md5( $ data['username']. $ password);
}
if (! $ data['truename']) {
    exit(json encode(array('code'=>1, 'msg'=>'真实姓名不能为空!')));
if (! $ data['gid']) {
    exit(json encode(array('code'=>1,'msg'=>'角色不能为空!')));
}
//执行添加或编辑操作
if ($ id > 0) {
    //完成信息编辑
    $ sql = "update wm_admin set truename = ?, gid = ?, status = ?, password = ? where id =
```

```
?";
    $ eid = db_exec(DB_AFFECTED, $ sql, 'siisi', [$ data['truename'], $ data['gid'], $ data['
status'], $ data['password'], $ id]);
    if ($eid) {
        exit(json encode(array('code'=>0,'msg'=>'管理员编辑成功!')));
    }else{
        exit(json encode(array('code' = >1, 'msg' = >'管理员编辑失败!')));
    }
}else{
    //检查用户名是否已被注册
    $ sql = 'select username from wm admin where username = ?';
    $ itme = db fetch(DB ROW, $ sql, 's', $ data['username']);
    if ($ itme) {
        exit(json encode(array('code'=>1,'msg'=>'用户名已经被注册!')));;
    }
    //完成数据插入
    $ sql = "INSERT INTO wm admin (username, truename, gid, status, add time, password)
VALUES (?,?,?,?,?,?)";
    $ nid = db exec(DB LASTED, $ sql, 'ssisis', array values($ data));
    if ($nid) {
        exit(json_encode(array('code' = > 0, 'msg' = >'管理员添加成功!')));
    }else{
        exit(json_encode(array('code' =>1, 'msg' =>'管理员添加失败!')));
    }
}
```

注意:编辑时表单中的"用户名"元素是只读的,也就是不允许修改管理员的用户名。 编辑时,若密码框为空,则默认为用户不修改密码。

| 12 | 理员列表            |              |       |     |                     | 洒          |
|----|-----------------|--------------|-------|-----|---------------------|------------|
| ID | 用户名 admin       | 真实姓名         | 角色    | 状态  | 编辑管理员信息             |            |
| 1  | admin           | 微梦           | 系统管理员 | 正常  | 用户名 编辑测试            |            |
| 2  | admin2          | 微梦2          | 系统管理员 | 禁用  | 角色 系统开发员            | v          |
| 3  | Ξ-              | ΞΞ           | 系统开发员 | 正常  | ATTENDED.           |            |
| 4  | 李四              | 李四           | 系统开发员 | 禁用  | d2#PHR              |            |
| 5  | 微梦              | 李木子          | 文字编辑  | 禁用  | 真实姓名 虚拟管理员          |            |
| 6  | 编辑测试            | 虚拟管理员        | 文字编辑  | 禁用一 | 秋 态 👘               |            |
|    |                 |              |       |     | 編輯管理员               |            |
| 马石 | 安魏文平 - 【PHP项目实践 | 与课程设计指导】 请华大 | 学出版社  |     |                     |            |
| 5  | 微梦              | 李木子          | 又字编辑  | 禁用  | 2019-07-05 11:18:06 | 99411 2043 |
| 6  | 编辑测试            | 虚拟管理员        | 系统开发员 | 正常  | 2019-07-05 19:39:53 | Signi HAR  |

运行效果如图 5.20 所示。

图 5.20 管理员编辑页面效果

从图中可以看出,第6条记录中的"角色"与"状态"数据已被成功修改。

#### 4. 删除管理员

删除管理员功能的实现非常简单,只需要给管理员列表页面中的"删除"按钮绑定一个单击函数 del(),并在 admin\deleteExc. php 文件中将数据从数据表中删除即可。

页面中的 JS 函数 del()代码如下。

```
function del(id) {
    layer.confirm(
        '确定要删除吗?',
        { btn: ['确定', '取消'], icon: 3 },
        function() {
         $.post(
             './deleteExc',
            { 'id': id },
             function(res) {
                 if (res.code > 0) {
                     layer.alert(res.msg, { icon: 2 });
                 } else {
                     layer.msg(res.msg);
                     setTimeout(function() {
                         window.location.reload();
                     }, 1000);
            }, 'json');
    });
}
```

删除之前需要再次询问用户是否确定删除,这里用 Layui 的"确认"框来实现,如图 5.21 所示。

| 12 | 理员列表   |       |                   |       |                     |             |
|----|--------|-------|-------------------|-------|---------------------|-------------|
| ID | 用户名    | 真实姓名  | 角色                | 状态    | 添加时间                | 操作          |
| 1  | admin  | 微梦    | 系统管理员             | 正常    | 1970-01-03 04:53:41 | inte anti-  |
| 2  | admin2 | 微梦2   | 信息                | ×     | 1970-09-29 19:46:29 | SPHIL INSIS |
| 3  | Ξ      | ΞΞ    |                   | 58Q.2 | 2019-06-24 19:08:02 | \$9412 BUSA |
| 4  | 李四     | 李四    | 1 WHATE SELECTION | n=91  | 2019-06-24 19:08:45 | \$7460 BMAX |
| 5  | 微梦     | 李木子   |                   | 確定 取消 | 2019-07-05 11:18:06 | titt: ment: |
| 6  | 编档型测试  | 虚拟管理员 | 系统开发员             | 正常    | 2019-07-05 19:39:53 | STIT INSE   |

图 5.21 管理员删除确认

实现数据删除的文件为 admin\deleteExc. php,代码如下。

```
<?php
```

...

//获取表单数据

```
$ id = isset($_POST['id']) ? (int)trim(htmlspecialchars($_POST['id'])) : 0;
if ($ id > 0) {
    $ sql = 'delete from wm_admin where id = ?';
    $ did = db_exec(DB_AFFECTED, $ sql, 'i', [$ id]);
    if ($ did) {
        exit(json_encode(array('code' = > 0, 'msg' = >'删除成功!')));
    }
    else {
        exit(json_encode(array('code' = > 1, 'msg' = >'删除失败!')));
    }
}
exit(json_encode(array('code' = > 1, 'msg' = >'删除失败!')));
```

由于篇幅的限制,用户信息管理模块功能就介绍这些,请读者依照本节方法完善该模块 的其他功能。例如,信息显示时的翻页、用户头像的上传等。

### 5.3.4 内容管理



#### 1. 内容显示

文章内容的列表在系统后台的主页上显示,如图 5.4 所示。该功能的实现非常简单,只 需要在 admin\index.php 文件中从数据表 wm\_article 中获取到全部文章即可,代码如下。

```
//获取全部文章
$ sql = 'select id,title,create_time from wm_article';
$ articles = db_fetch(DB_ALL, $ sql);
```

在视图文件 admin\view\v\_index. php 中循环输出这样文章的 ID、标题和添加时间,代码如下。



#### 2. 内容添加

单击后台页面侧边栏中的"添加文章"菜单,或者单击页面右侧的"添加"按钮,即可打开 文章添加页面,如图 5.22 所示。

| 微学后台管理系统   | ×  | +            |                       |                     |           |    |      |       |      |                                   |   |            |            |             | -   | σ    | ×  |
|--|----|--------------|-----------------------|---------------------|-----------|----|------|-------|------|-----------------------------------|---|------------|------------|-------------|-----|------|----|
| ← → C @ ♦ #####                                  | •  | book.php.chp | 05/index.php/admin/ir | 8                   |           | 4  | Q 投友 | lin,  | 4.9  |                                   | ۵ | • 4<br>0#4 | ذ≫<br>NRR∐ | 988<br>2890 |     |      |    |
| ☆ 微梦后台   | 管理 | 系统           | <b>11</b>             | <b>前位置:</b> 后台 / 主部 | 5 / 16mcs | 2章 | 現在是  | 20204 | F01) | 12日 10 <b>81</b> 23 <del>5)</del> |   | 当前在        | 18         | QУ          | 退出来 | iife | Î  |
| > 用户管理   |    |              | 澤加文章                  |                     |           |    |      |       |      |                                   |   |            |            |             | 21  | 8510 | 11 |
| <ul> <li>&gt; 分类管理</li> <li>&gt; 文宣管理</li> </ul> |    |              | 文章标题                  | 请输入文章标题             |           |    |      |       |      |                                   |   |            |            |             |     |      |    |
| > 系統管理   |    |              | 作者姓名                  | QY.                 |           |    |      |       |      |                                   |   |            |            |             |     |      | Ш  |
| > 管理员管理  |    |              | 播加时间                  | 2020-01-12 10:23    | :03       |    |      |       |      |                                   |   |            |            |             |     |      | Ш  |
| > 单单管理   |    |              | 文章类型                  | 请选择文章类型             | Ŧ         |    |      |       |      |                                   |   |            |            |             |     |      | Ш  |
|  |    |              | 是否置质                  | •                   |           |    |      |       |      |                                   |   |            |            |             |     |      | Ш  |
|  |    |              | 文章内容                  | 清输入文章内容             |           |    |      |       |      |                                   |   |            |            |             |     |      |    |
|  |    |              |                       |                     |           |    |      |       |      |                                   |   |            |            |             |     |      |    |
|  |    |              |                       | 立即提交                | 重党        |    |      |       |      |                                   |   |            |            |             |     |      |    |

图 5.22 添加文章表单

页面中的文章类型数据来自数据库中的 wm\_article\_category 数据表。选择框展开后的效果如图 5.23 所示。

| → C  | ook. <b>php.chp05</b> /index.php/admin/li | nsert_article            |   | 81 | ☆ | Q複素 | 11/ | * | • 0 | • | د "ک<br>(#00%) | E10753 |
|--|---|--------------------------|---|----|---|-----|-----|---|-----|---|----------------|--------|
| > 用户管理   | 逐加文章                                      |                          |   |    |   |     |     |   |     | ļ | <b>建块</b> 适应   |        |
| <ul> <li>&gt; 分與管理</li> <li>&gt; 文章管理</li> </ul> | 文章乐题                                      | 请输入文章标题                  |   |    |   |     |     |   |     |   |                |        |
| > 5.68WB   | 作者姓名                                      | ay                       |   |    |   |     |     |   |     |   |                |        |
| > 管理员管理  | S&acart(#)                                | 2020-01-12 10:23:03      |   |    |   |     |     |   |     |   |                |        |
| > 菜串製道   | 文章美型                                      | 请选择文章类型                  | - |    |   |     |     |   |     |   |                |        |
|  | 是否置顶                                      | 请选择文章类型                  |   |    |   |     |     |   |     |   |                |        |
|  | 文章内容                                      | php<br>mysql<br>thinkphp |   |    |   |     |     |   |     |   |                |        |
|  |   | java<br>css              |   |    |   |     |     |   |     |   |                |        |
| 【PHP语目实践与课程设计能导】                                 | 数材定则 马石安 教文平 清泉大学                         | jquery                   |   |    |   |     |     |   |     |   |                |        |

图 5.23 文章类型选择

添加文章表单页面设计完成后,就可以编写代码完成数据的插入了。实现该功能的代码与上述添加管理员代码相同,只是将操作的数据表更改为 wm\_article 即可。这里不再赘述。

### 5.4 前台功能实现



PHP Web 应用项目的前台一般用于展示系统信息,因此,主要使用的是数据库的查询。本节实现案例项目前台的部分功能,包括首页、详情,以及分类查询等。

### 5.4.1 前台首页

系统前台首页用于概要性地展示系统发布的信息,页面效果如图 5.1 所示。

#### 1. 主菜单

系统前台首页顶部的导航菜单,与5.3.1节中的侧边栏导航菜单一样,其数据也是来自数据库中的 wm\_menus 表。因此,获取菜单数据的方式也与前述相同,只是这里获取的是 home 模块中位置为 top 的菜单项,代码如下。

| //获取菜单   |             |
|--|-------------|
| <pre>\$ top_navs = getMenus('home', 'top');</pre>                        |             |
| foreach ( $\$$ top_navs as $\$$ k => $\$$ v) {                           |             |
| \$ data['top_navs'][ \$ k] = \$ v;                                       |             |
| <pre>\$ data['top_navs'][ \$ k]['sub'] = getSubMenus( \$ top_navs,</pre> | \$v['id']); |
| }  |             |

菜单效果如图 5.24 所示。

| 微梦内容管理系统 | × +    |            |     |                |  |        |       |       |    |        |      |    |     | σ   |     |
|----------|--------|------------|-----|----------------|--|--------|-------|-------|----|--------|------|----|-----|-----|-----|
| € → ሮ @  | 0 🔏 bo | ok.php.chp | 05  |                |  |        | 麗 … 4 | く設ま   |    | II\ 14 |      | ۵  | • 6 | î » | Ę   |
|          | 1      |            |     |                |  |        |       |       |    |        |      |    | 088 |     | 3## |
| m3       |        | AVAL.      |     |                |  | 用户中心 🔺 | 关于我们  | 青轮关键字 | 搜索 | 3      | 医尿 : | 注册 |     |     | Î   |
|          |        | 个人信息       |     |                |  |        |       |       |    |        |      |    |     |     |     |
| 全部以      | 〈章     |            |     |                |  | 我的收藏   | R     | 別浏览   |    |        |      |    |     |     |     |
|          |        |            |     |                |  | 我的文章   |       |       |    |        |      |    |     |     |     |
| 623 TH A | a      | -          | PHP | <b>FAjax操作</b> |  | ·      |       | 文章标题  |    |        | 5    | 遗  |     |     |     |

菜单视图采用 Layui 的水平菜单,请参见源码。

#### 2. 全部文章

在系统前台主页的左侧,显示了系统中的全部文章概要,包括图标、标题、作者、发布日 期,以及内容缩略。

系统文章数据存放在数据库中的 wm\_article 数据表中,将其取出显示即可,代码如下。

//获取全部文章
\$ sql = 'select \* from wm\_article as a join wm\_user as u where a.user\_id = u.id';
\$ articles = db fetch(DB ALL, \$ sql);

注意:这里使用了数据表的联合查询,因为需要从 wm\_user 表中读取作者的姓名。在

## PHP

#### 项目案例开发从入门到实战-微课视频版

wm\_article 表中,只记录了作者的 ID,也就是 wm\_user 表中的 id 字段的值。

3. 热门浏览

在系统前台主页的右侧,显示了最近的热门文章概要,包括文章标题和浏览次数。热门 文章是从上面获取到的全部文章中筛选出来的,代码如下。

```
//获取全部文章
$ sql = 'select * from wm_article as a join wm_user as u where a. user_id = u.id';
$ articles = db_fetch(DB_ALL, $ sql);
//获取热门文章
$ hot_articles = array();
foreach ($ articles as $ a) {
    if ($ a['is_hot'] == 1) {
        array_push($ hot_articles, $ a);
    }
}
```

热门文章在数据表中存储时,其 is\_hot 字段的值为 1。

### 5.4.2 内容详情

单击主页中显示的文章标题,即可查看该篇文章的详情,如图 5.2 所示。

#### 1. 获取数据

根据文章标题链接传递过来的文章 ID,从数据表 wm\_article 中查询该篇文章的详细内容,代码如下。

```
//获取文章 ID
$ article_id = isset($_GET['id'])?(int)$_GET['id']:null;
if($ article_id === null) {
    header('Location: /');
    exit;
}
//获取该篇文章
$ sql = 'select * from wm_article where id = '.$ article_id;
$ article = db_fetch(DB_ALL, $ sql);
//获取作者姓名
$ sql = 'select name from wm_user where id = '.$ article[0]['user_id'];
$ user = db_fetch(DB_ALL, $ sql);
$ article[0]['uname'] = $ user[0]['name'];
//获取热门文章
$ sql = 'select * from wm_article where is_hot = 1';
$ hot_articles = db_fetch(DB_ALL, $ sql);
```

与主页一样,在文章详情页面的右侧也显示了最近的热门文章,所以这里也要再次获取 到这些文章的标题及浏览次数。

#### 2. 显示详情

```
在视图中显示文章详情,代码如下。
< div class = "layui - col - md8">
   < div class = "layui - card main - title" >
       <div class = "layui - card - header">文章详情</div>
   </div >
   < div class = "layui - row" style = "height:130px;border - bottom: 1px solid # eee;">
       <h2 style = "color: # 009688"><? = $ data['article']['title']?></h2>
          < span style = "padding - right: 20px;">作者: <? = $ data['article']['uname']?</pre>
></span>
             <span>发布时间: <? = date('Y-m-d', $ data['article']['create time'])?>
span > 
       <?php echo htmlspecialchars( $ data['article']['content']); ?>
   </div>
</div>
```

注意: 在文章详情页面,单击右侧的热门文章标题,也可以在页面左侧显示该文章详 情。实现代码参见源码。

### 5.4.3 分类查询

本项目中文章的分类查询通过主菜单来实现。也就是说,系统前台主菜单中的 PHP、 JAVA、CSS 等实际上是文章的分类名称。

例如,单击 PHP 菜单,则会查询出所有类型为 PHP 的文章,如图 5.25 所示。

```
微梦内容管理系统
                  × +
€ → C @

    Kook.php.chp05/index.php/home/category?cate_name=php

                                                                             誕 … ☆ Q 搜索
                                                                                                        M な 🗭 🖸 ゥ 📽 »
◆最早访问 □ 常用网址
                                                                                                                     日移动设备上的书签
                                                                                   请给关键字
                                                                                                   搜索
                                                                                                             登录 注册
          php
                                                                                   热门浏览
                                 PHP与Aiax操作
                                                                                    文章标题
                                                                                                                浏览
                                 作者: admin 岁布时间: 2018-02-01
           PHP与Ajax
                                                                                    PHP与Ajax操作
                                                                                                                 5
                                 PHP与Ajax操作PHP与Ajax操作PHP-Ajaxax操作PHP与Ajax操作P与Ajax操作F
与Ajax操作PHP与Ajax操作PHP与Ajax
           印颜韵等
                                                                                    ThinkPHP5企业站点开发指南
                                                                                                                 2
                                                                                    ThinkPHP5禄彝视颂教程
                                                                                                                 1
                                 PDO极速入门教程
                                  作者: admin 发布时间: 2018-02-01
                                                                                    自己动手写框架数程
                                                                                                                 2
                                 PDO极速入门数程PDO极速入门数程PDO极速入门数程PDO极速入门数程PDO
                                                                                    新闻发布系统教程
                                                                                                                 1.1
                                  极速入门数程PDO极速入门
                                 自己动手写框架教程
                                  作者: admin 发布时间: 2018-02-01
                                 自己动手写框架教程自己动手写框架教程自己动手写框架教程自己动手写框架教
                                   自己动手
                                 PDO极速入门教程
                                 作者: admin 发布时间: 2018-02-01
```

图 5.25 分类查询

**注意**:主页中原来的"全部文章"文本,更改为"php",说明查询到的是"php"分类的文章。 查询代码如下。

```
//获取分类名称
$ article category = isset($ GET['cate name']) ? htmlentities(trim($ GET['cate name'])):
null;
if ($ article category === null) {
    header('Location: /');
    exit;
}
$ sql = "select id from wm article category where name = '". $ article category."'';
$ category id = db fetch(DB ALL, $ sql);
//获取分类文章
$ sql = 'select a. id as aid, a. title, u. name, a. create time, a. content, a. title img, a. is hot, a.
pv from wm_article as a join wm_user as u where a.user_id = u.id and a.cate_id = '. $ category_
id[0]['id'];
$ articles = db_fetch(DB_ALL, $ sql);
//获取热门文章
$ sql = 'select * from wm article where is hot = 1';
$ hot_articles = db_fetch(DB_ALL, $ sql);
```

注意:分类名称由菜单的 href 属性传递,代码如下。

< a href = "<?php echo \$ m['href'].'?cate\_name = '.strtolower
(\$ m['name'])?>"><? = \$ m['name']?></a>

详细代码请参见源码。

### 5.4.4 文章搜索

文章的搜索通过主菜单右侧的搜索表单来实现,这里搜索的是文章标题中的关键字,采 用模糊查询方式。运行效果如图 5.26 所示。

散梦内容管理系统	51	× +														-	ø	×
← → C <sup>2</sup> ♦ ₩₩60 (2)	<b>छे छ</b> प्रमान्नध	) 🔏 book.php.chp05/index.php/home/search							82	··· ☆	Q, 搜索		hit.		•	5 C	°≫ NR∰⊥1	≣A Bate
	首页	PHP	AVAL	HTML	CSS	JQUERY	用尸中心▼	关于我们		aja	u	搜索	2	<u>D</u> .	≹ 注册	2		Î
	搜索到的	漆到的所有文章 热门浏览																
1	僻违人门	15		PHP	āAjax操作	i				文	章标题				MA			
1	PHP=TAja 視频教学	ax	n	PHP5	PHP、AUMAN 2005月9月、2010-02-01 PHP与A(ad程作PHP与A(ad程作PHP与A(ad程作PHP与A(ad程作P与A(ad程作 Faland任PHP)日になった日本					P	PHP与Ajao操作				5			
				-stadiovanist na analogovanist na alas dovenna							ThinkPHP5企业站而升发错阈 ThinkPHP5经典视频数程				2			
	信道人们 PHP与Ai	p.	•	PHP5 作者:	aAjax操作 wwp 发动	时间: 2018-0	02-01			6	1己动手写框架	敗程		2	2			
- 1	視頻較学		n	PHP与 PHP与	PHP与Ajac機作PHP与Ajac PHP与Ajac機作PHP与Ajac		α操作PHP与Ajax操作PHP与Ajax操作PHP与Aj α操作PHP与Aj		Ajax操作	85	闻发布系统数	12						
	健强人们 PHP与Aji 視频教学	ax	п	PHP의 作者: PHP与 PHP与	GAjax操作 mp 224 Ajao銀作PH Ajao銀作PH	。 68分词:2018- P与Ajax操作户 P与Ajax操作户	02-01 旧与Ajax銀竹PHI 旧与Aj	₽与Ајах操作₽Н₽与	Ajax疑作									

图 5.26 文章搜索

```
数据查询代码如下。
```

```
//获取文章标题中的关键字
$ keyword = isset($_POST['keyword']) ? htmlentities(trim($_POST['keyword'])): null;
if ($ keyword === null) {
    header('Location: /');
    exit;
}
//获取查询到的全部文章
$ sql = "select a. id as aid, a. title, u. name, a. create_time, a. content, a. title_img, a. is_hot, a.
pv from wm_article as a join wm_user as u where a. user_id = u. id and a. title like '%".
$ keyword." % '";
$ articles = db_fetch(DB_ALL, $ sql);
```

页面头部右侧的搜索框视图代码存放在 home\view\top\_search. php 文件中,代码如下。

```
<?php defined('APP_NAME') or exit('非法访问');?>
<form class = "layui - form" action = "/index.php/home/search" method = "post">
<form class = "layui - inline" style = "padding:10px;">
<form class = "layui - inline" style = "padding:10px;">
<form class = "layui - inline" style = "padding:10px;">
</div class = "layui - input - inline">
</div class = "layui - inline">
</div class = "layui - inline">
</div class = "layui - input - i
```

### 5.5 本章小结

本章详细介绍了 PHP 语言面向过程的编程方法,以及开发一个简单的内容管理系统 的过程。由于篇幅的限制,这里重点讲述了系统前/后台一些基本功能的实现。关于系统中 的一些其他功能,或者是读者觉得案例中不是很满意的功能的实现,留给读者作为本章学习 的课后作业。希望读者充分发挥自己的想象力与创造力,使本章案例项目变得更加完美。