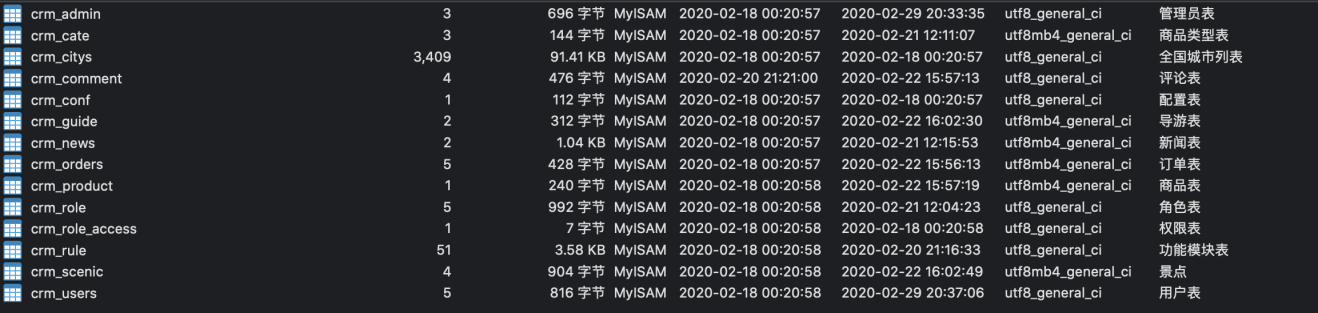
后台访问地址：<http://ccc.flv.pet/admin/index/index.html> 账号：admin 密码：123456

前台访问地址：<http://ccc.flv.pet/> 账号：qiaoshuai 密码：123456

第一步：数据库架构（创建数据库），使用的是mysql数据库，表引擎使用的MyISAM



第二步：开发后台系统：后台页面选取的是layui框架，业务框架采用thinkphp5.0

第三步：开发功能，功能依次顺序开发（登录，权限模块，基础配置，订单模块，信息模块，用户模块，新闻模块，评论模块）

1. 登录：使用的是aiax请求登录，登录成功跳转首页，进行的权限控制，不登录访问首页会自动跳转到登录页面

2. 权限模块：数据为myql存储，通过权限控制分配的方式，进行对应后台账户的功能权限管理，管理员可自行设置权限规则账号

3. 基础配置：配置了后台系统的系统名称，文件上传的大小限制，版权设置信息，还有用户登录的初始密码等，数据也是进行的后台mysql存储

4. 订单模块：商品的上下架管理，商品类型管理，订单的信息展示

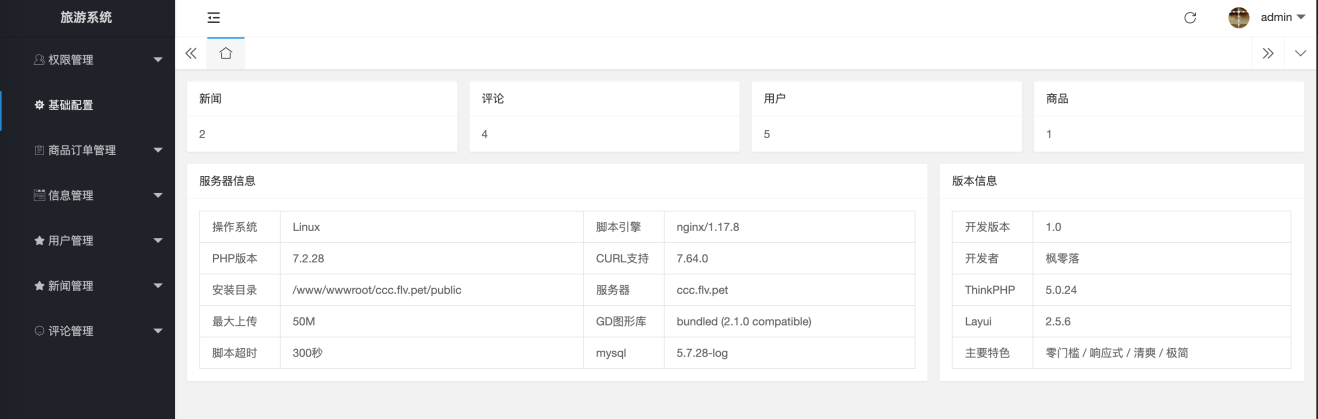
5. 信息模块：导游的增删改查，景点的增删改查

6. 用户模块：前台登录用户的增删改查和账号的管理

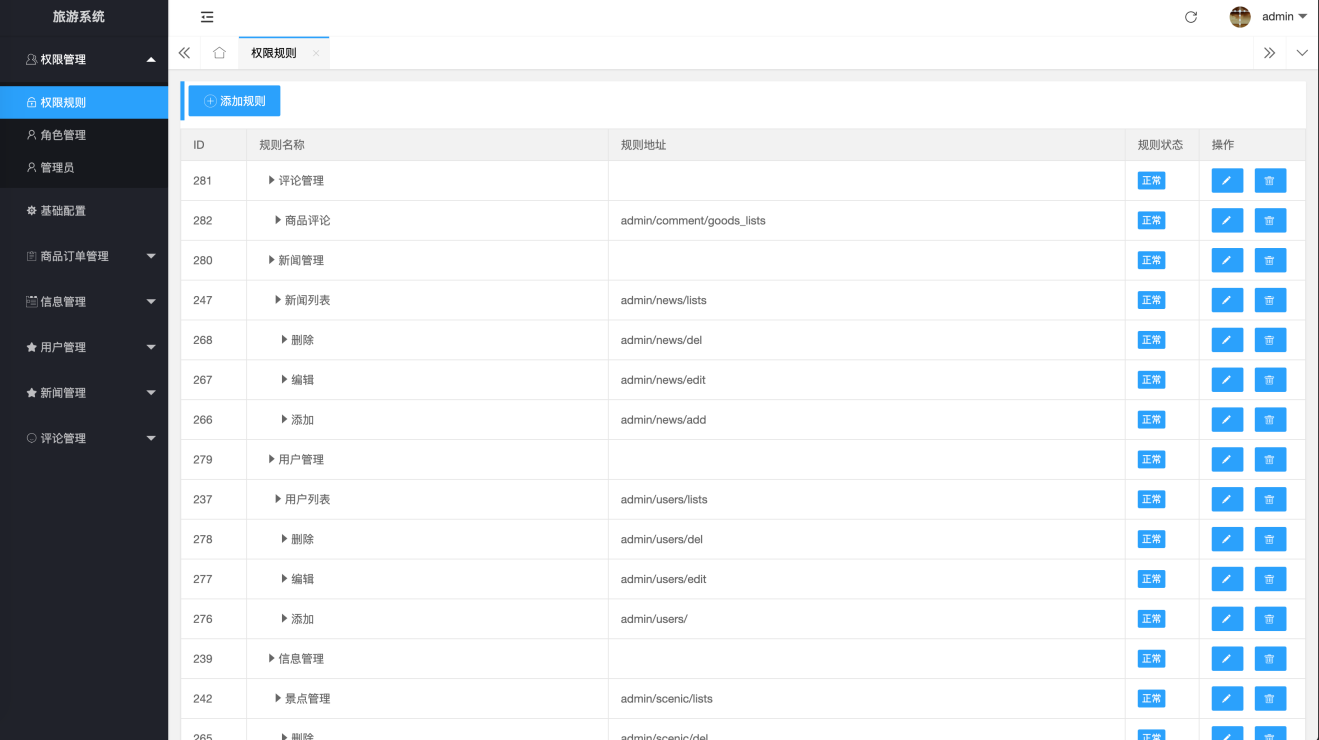
7. 新闻模块：新闻的增删改查

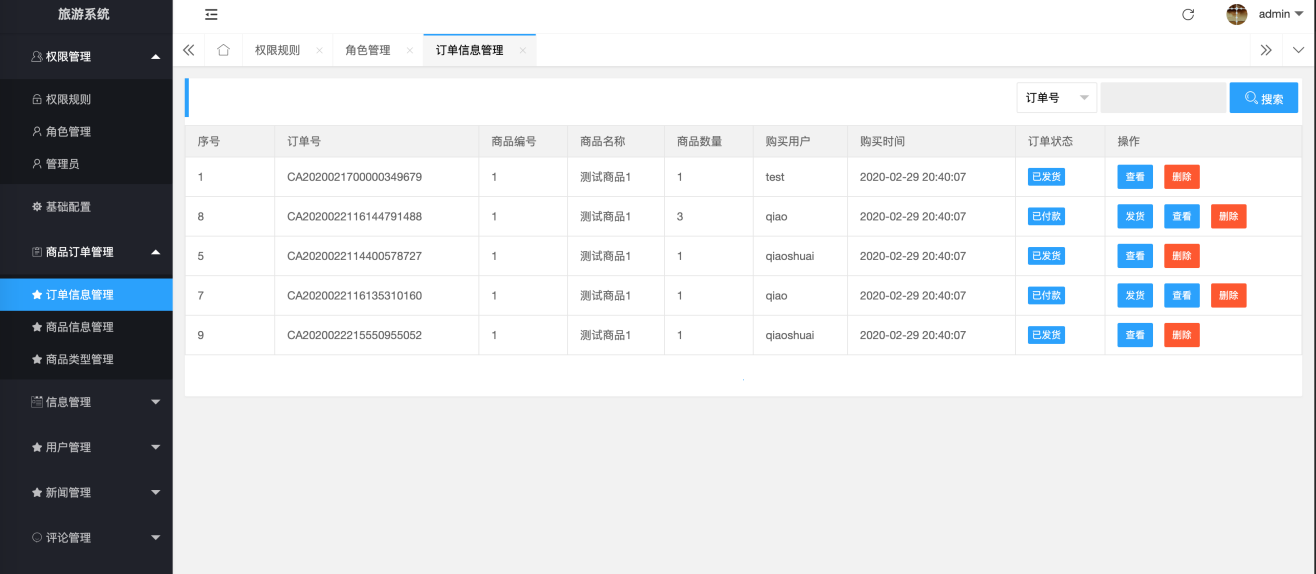
8. 评论模块：商品评论的管理

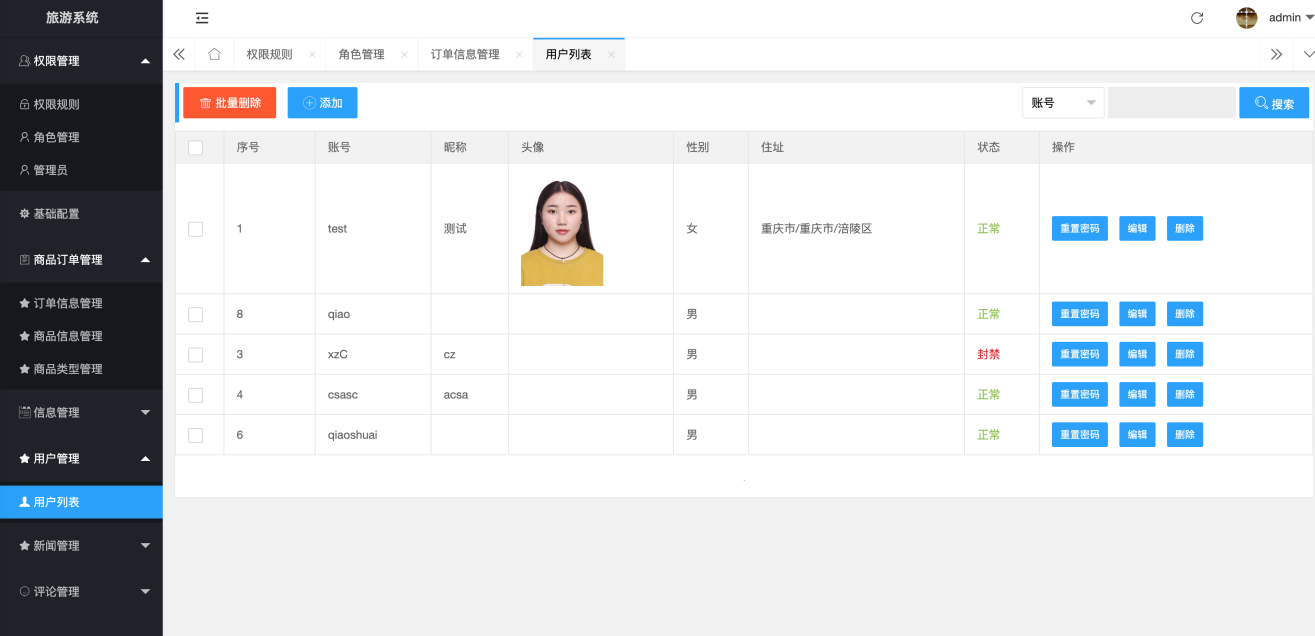
所有模块的数据列表展示全部都带有分页



第四步：开发第三步的相应模块功能









第五步：开发前台功能（模板网上扒取，根据后台数据库进行数据嵌套）

第六步：开发前台个人中心功能：登录，个人资料，我的订单，修改密码，退出等功能

技术知识点：

1. 后台权限模块：
2. 使用了权限二叉树算法进行数据重组，数据库表结构为无限级分类结构
3. 使用了layui的二叉树扩展包进行页面展示
4. 存储时进行了二叉树算法的解析储存
5. 全模块的数据列表：
6. 使用了thinkphp5.0的分页技术
7. 前后台登录功能的验证码：
8. 使用了thinkphp5.0的验证码扩展，该扩展需要手动安装：composer
9. js的ajax技术请求，thinkphp5.0的session存储机制
10. 用户模块：
11. 重置密码：使用了php7以上的内置密码加密解密函数（加盐方式）
12. 修改密码：同理

核心代码：

登录：

public function login\_action(){

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

$post = request()->post();

if(!captcha\_check($post['vercode'])){

$data = [

'msg' => '验证码输入不正确',

'icon' => 5

];

}else{

$res = (new Users())->where('user\_name',$post['user\_name'])->find();

if(empty($res)){

$data = [

'msg' => '账号错误，请重新填写！！！',

'icon' => 5

];

}else{

if (cms\_pwd\_verify($post['user\_pwd'],$res['user\_pwd'])) {

if ($res['user\_status'] == 0) {

$data = [

'msg' => '用户已被冻结！',

'icon' => 5

];

}else{

$update = [

'last\_login' => time(),

'user\_host' => request()->ip()

];

(new Users())->where('id',$res['id'])->update($update);

$session\_user = array(

'uid' => $res['id'],

'username' => $res['user\_name'],

);

session('users', $session\_user);

$data = [

'msg' => '登录成功，正在跳转中.....',

'icon' => 6,

];

}

}else {

$data = [

'msg' => '密码错误，请重新填写！！！',

'icon' => 5

];

}

}

}

return json($data);

}

}

注册的：

public function zhuce\_action(){

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

$post = request()->post();

if(!captcha\_check($post['vercode'])){

return json(['msg'=>'验证码输入不正确','icon'=>5]);

}else{

if ($post['user\_pwd'] != $post['user\_repwd']) {

return json(['msg'=>'两次输入的密码不正确','icon'=>5]);

}

$post['user\_pwd'] = cms\_pwd\_encode($post['user\_pwd']);

$res = (new Users())->where('user\_name',$post['user\_name'])->find();

if ($res) {

return json(['msg'=>'该用户已注册,请登录','icon'=>5]);

}else{

if ((new Users())->save($post)) {

return json(['msg'=>'注册成功,请登录','icon'=>6]);

}else{

return json(['msg'=>'注册失败,请重新注册','icon'=>5]);

}

}

}

}else{

return json(['msg'=>'请求错误','icon'=>5]);

}

}

个人中心：

个人资料

public function users(){

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

$post = request()->post();

$post['user\_city'] = $post['user\_city'].'&nbsp;'.$post['user\_city1'];

if ((new UsersModel())->update($post)) {

$data = [

'msg' => '保存成功',

'icon' => 6

];

}else{

$data = [

'msg' => '保存失败',

'icon' => 5

];

}

return json($data);

}else{

$user = (new UsersModel())->where(['id'=>$this->user['uid']])->find();

if (!empty($user['user\_city'])) {

$citys = explode('&nbsp;',$user['user\_city']);

$user['user\_city'] = $citys[0];

$user['user\_city1'] = $citys[1];

}else{

$user['user\_city'] = '';

$user['user\_city1'] = '';

}

$this->assign([

'user' => $user

]);

return $this->fetch();

}

}

个人中心的修改密码

public function edit\_pwd(){

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

$post = request()->post();

$encode\_pwd = cms\_pwd\_encode($post['user\_pwd']);

if (cms\_pwd\_verify($post['user\_pwd'],$encode\_pwd)) {

$post['user\_pwd'] = cms\_pwd\_encode($post['user\_pwd\_news']);

if ((new UsersModel())->update($post)) {

return json(['msg'=>'保存成功','code'=>200]);

}else{

return json(['msg'=>'保存失败','code'=>400]);

}

}else{

return json(['msg'=>'旧密码错误','code'=>400]);

}

}else{

return json(['msg'=>'请求错误','code'=>400]);

}

}

后台：

权限核心代码：

public function getChildMenu(){

$parent\_id = input('id');

$user = session('user');

$childMenus = (new Rule)->getChildMenu($parent\_id,$user['rule']);

$str = "";

foreach ($childMenus as $cm){

$str .= '<li class="layui-nav-item">';

if (!isset($cm['url'])) {

$str .= '<a href="javascript:;" lay-tips="'.$cm['name'].'" lay-direction="2">

<i class="layui-icon '.$cm['icon'].'"></i>

<cite>'.$cm['name'].'</cite>

</a>';

}else{

$str .= '<a lay-href="'.$cm['url'].'" lay-tips="'.$cm['name'].'" lay-direction="2">

<i class="layui-icon '.$cm['icon'].'"></i>

<cite>'.$cm['name'].'</cite>

</a>';

}

if(!empty($cm['child'])){

$str .= '<dl class="layui-nav-child">';

foreach($cm['child'] as $cd){

$str .= '<dd><a lay-href="'.$cd['url'].'" >

<i class="layui-icon '.$cd['icon'].'"></i>

<cite>'.$cd['name'].'</cite>

</a>

</dd>';

}

$str .= '</dl>';

}

}

if(empty($str)){

$str="<script>alert('您没有操作权限')</script>";

}

return $str;

}

上传图片：

public function upload(){

$file = request()->file('file');//获取文件信息

$path = 'uploads';//文件目录

//创建文件夹

if(!is\_dir($path)){

mkdir($path, 0755, true);

}

$info = $file->move($path);//保存在目录文件下

if ($info && $info->getPathname()) {

$data = [

'status' => 1,

'data' => '/'.$info->getPathname(),

];

return json\_encode($data);

} else {

return json\_encode($file->getError());

}

}

清除缓存

// 清除缓存

public function delruntime(){

$this->clear\_sys\_cache();

$this->clear\_temp\_ahce();

$this->clear\_log\_chache();

return 1;

}

public function clear\_sys\_cache() {

Cache::clear();

return true;

}

// 清除模版缓存 不删除 temp目录

public function clear\_temp\_ahce() {

array\_map( 'unlink', glob( TEMP\_PATH.DS.'.php' ) );

return true;

}

// 清除日志缓存 不删出log目录

public function clear\_log\_chache() {

$path = glob( LOG\_PATH.'/' );

foreach ($path as $item) {

array\_map( 'unlink', glob( $item.DS.'.' ) );

rmdir( $item );

}

return true;

}

分配权限：

public function RuleTree($id=0){

$Rules = $this->select();

$rules = (new Role)->field('rules')->where('id',$id)->find();

if(!empty($rules)){

$rule = explode(',', $rules['rules']);

$arr = $this->sort($Rules,$rule);

}else{

$arr = $this->sort($Rules,$rules);

}

return $arr;

}

// xtree数据格式递归

public function sort($data,$rules,$pid=0){

$arr = $one = array();

foreach ($data as $k => $v) {

$one['title'] = $v['name'];

if($v['pid']==$pid){

$one['value']=$this->getparentid($v['id']);

if (!empty($rules) && in\_array($v['id'], $rules)){

$one['checked'] = true;

}else{

$one['checked'] = false;

}

$one['data'] = $this->sort($data,$rules,$v['id']);

$arr[]=$one;

}

}

return $arr;

}

public function GetRulesAll(){

$RulesRes = $this->field('id,url,name,status,pid,level\_id')->order('id','desc')->select();

return $this->GetRules($RulesRes);

}

public function GetRules($RulesRes,$pid=0){

static $RulesAll = array();

foreach ($RulesRes as $Rules){

if ($Rules['pid'] == $pid){

$str = ($Rules['level\_id'] != 0)?'<i class="layui-icon layui-icon-triangle-r"></i>':'';

if ($Rules['url'] == '#') {

$Rules['url'] = '';

}

$Rules['\_name'] =str\_repeat("&nbsp;&nbsp;",$Rules['level\_id']\*1).$str.$Rules['name'];

$RulesAll[] = $Rules;

$this->GetRules($RulesRes,$Rules['id']);

}

}

return $RulesAll;

}