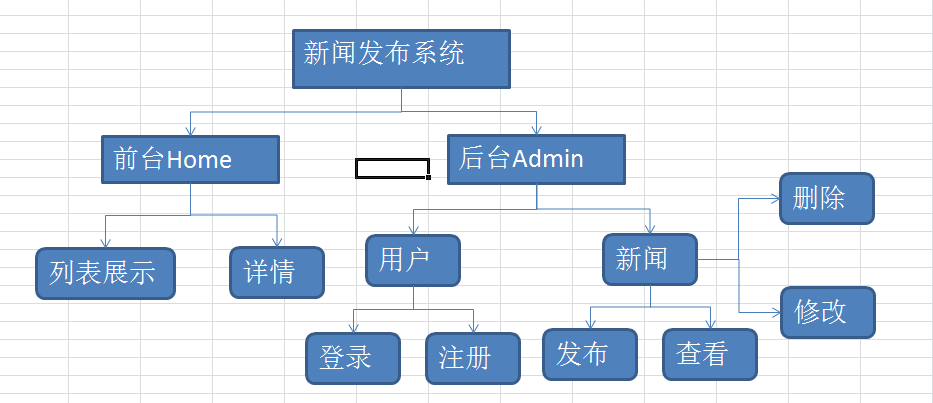
## 项目介绍

### 项目名称：新闻（博客）发布系统

### 项目周期：4天

### 功能图解：



### 功能介绍：

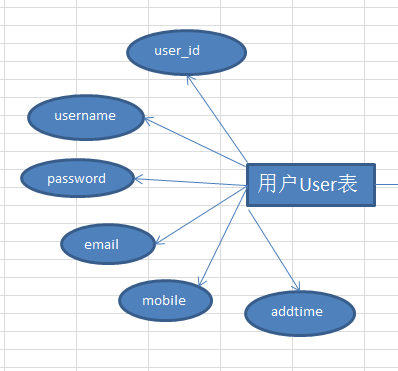
1. 后台：  
   用户的注册与登录，保证用户的唯一性。  
   新闻发布、查看、修改、删除，只能操作自己发布的新闻。  
   新闻列表，保证每个用户只能看到自己的新闻。
2. 前台：

新闻列表，当前用户发布的文章后面带有修改和删除功能

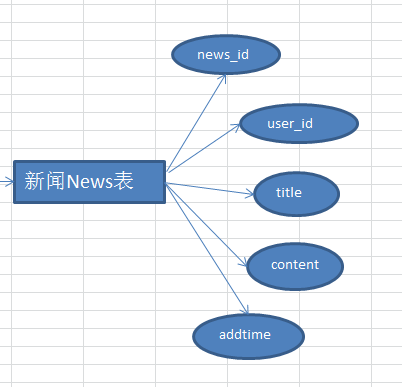
新闻详情

### ER图：

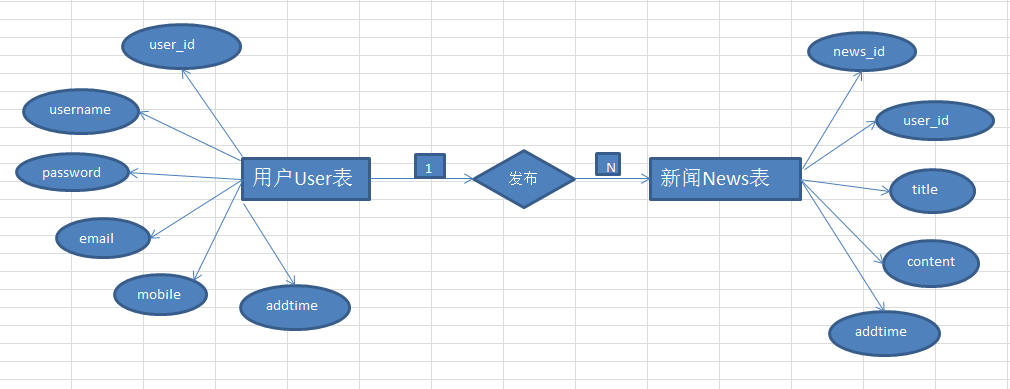
用户表（user）：



新闻表（news）：



用户表和新闻表是1对多的关系（1:N）, 用户发布新闻，一个用户可以发布多条新闻。



## 数据库设计

1. 创建数据库（blog）

|  |
| --- |
| **CREATE** DATABASE `blog`; |

1. 用户表（user）

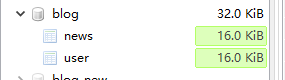
|  |
| --- |
| **CREATE** **TABLE** `user` (  `user\_id` **INT**(10) **UNSIGNED** **NOT** **NULL** **AUTO\_INCREMENT** **COMMENT** '会员主键ID',  `username` **VARCHAR**(32) **NOT** **NULL** **COMMENT** '用户名',  `password` **CHAR**(32) **NOT** **NULL** **COMMENT** '密码',  `email` **VARCHAR**(32) **NOT** **NULL** **COMMENT** '邮件',  `mobile` **CHAR**(11) **NOT** **NULL** **COMMENT** '手机号',  `addtime` **INT**(11) **UNSIGNED** **NULL** **DEFAULT** '0' **COMMENT** '注册时间',  **PRIMARY** **KEY** (`user\_id`)  )  **COMMENT**='用户表'  **COLLATE**='utf8\_general\_ci'  **ENGINE**=**InnoDB**  **AUTO\_INCREMENT**=1  ; |

1. 新闻表（news）

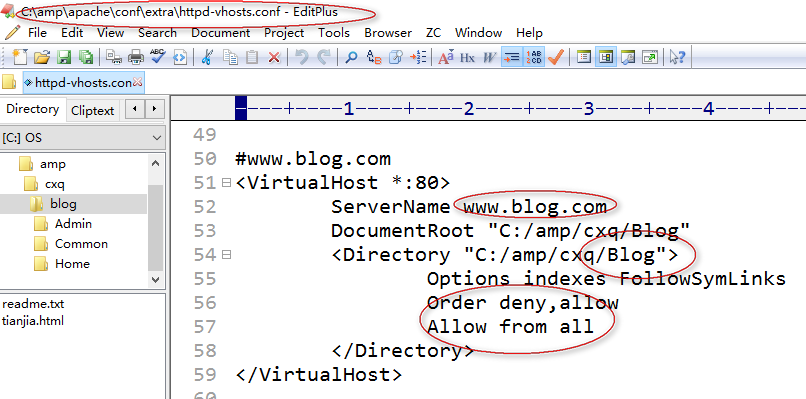
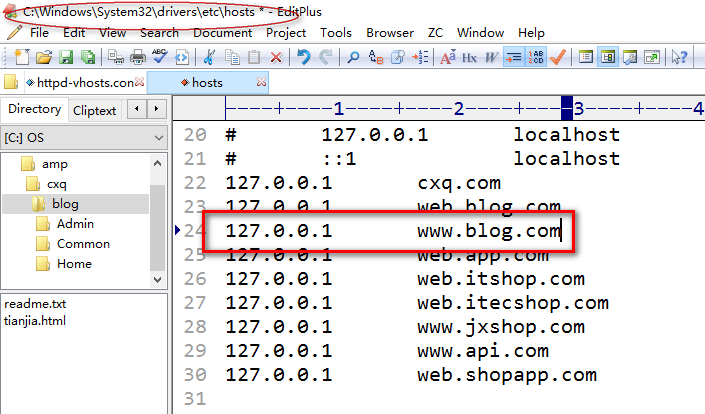
|  |
| --- |
| **CREATE** **TABLE** `news` (  `news\_id` **INT**(10) **UNSIGNED** **NOT** **NULL** **AUTO\_INCREMENT** **COMMENT** '自增主键ID',  `user\_id` **INT**(10) **UNSIGNED** **NOT** **NULL** **COMMENT** '用户ID',  `title` **VARCHAR**(50) **NOT** **NULL** **COMMENT** '新闻标题',  `content` **TEXT** **NOT** **NULL** **COMMENT** '新闻内容',  `addtime` **INT**(10) **UNSIGNED** **NOT** **NULL** **DEFAULT** '0' **COMMENT** '发布时间',  **PRIMARY** **KEY** (`news\_id`)  )  **COMMENT**='新闻表'  **COLLATE**='utf8\_general\_ci'  **ENGINE**=**InnoDB**  **AUTO\_INCREMENT**=1  ; |

2017年7月13日11:09:08

展示效果：



## 配置虚拟主机

1. 修改httpd-vhosts.conf文件  
   
2. 做域名解析  
   
3. 重启Apache服务器

测试效果：创建站点根目录blog



此时，说明我们的虚拟主机已经配置OK，接下来就可以开发项目了。

## 初始化目录结构：

在站点根目录blog下面创建三个目录，Admin、Home、Common，分别作用于后台、前台、公共。

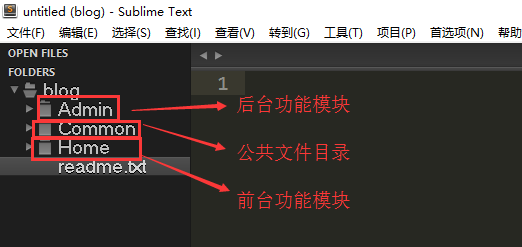
Admin：主要是后台的功能文件。

Home：主要是前台的功能文件。

Common：公共目录，存放一些公用的函数、静态资源等

Public：存放前端资源：CSS、JS、Img，Upload

### 目录结构如下图所示：

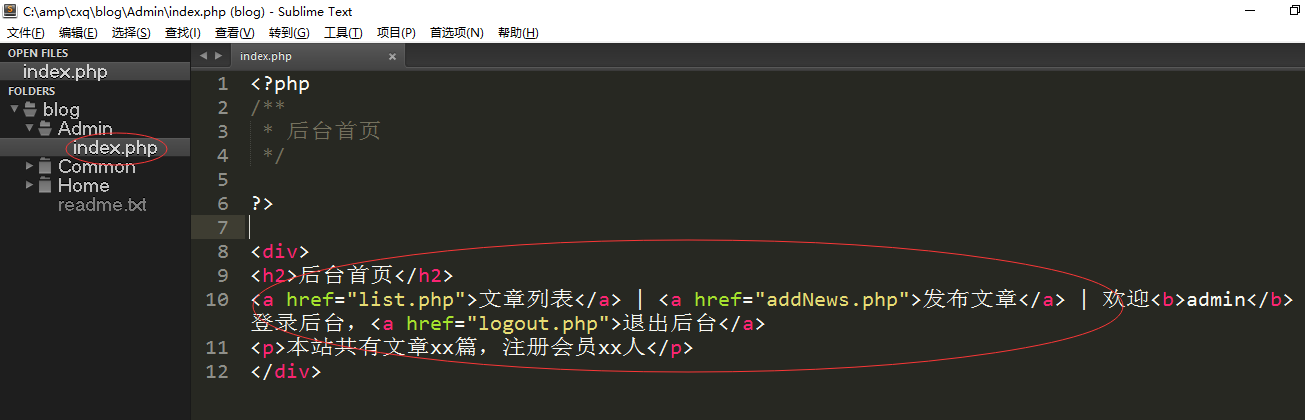


## 后台功能开发

### 1.1后台首页模板

创建/admin/index.php，放入预先制作的模板。

代码如下

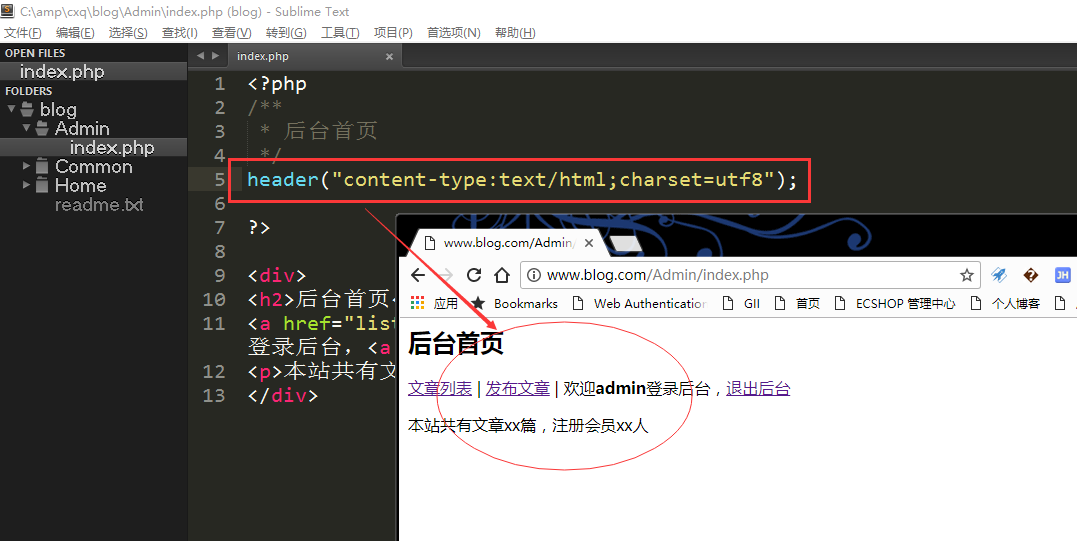


访问地址：<http://www.blog.com/Admin/index.php>

展示访问效果：



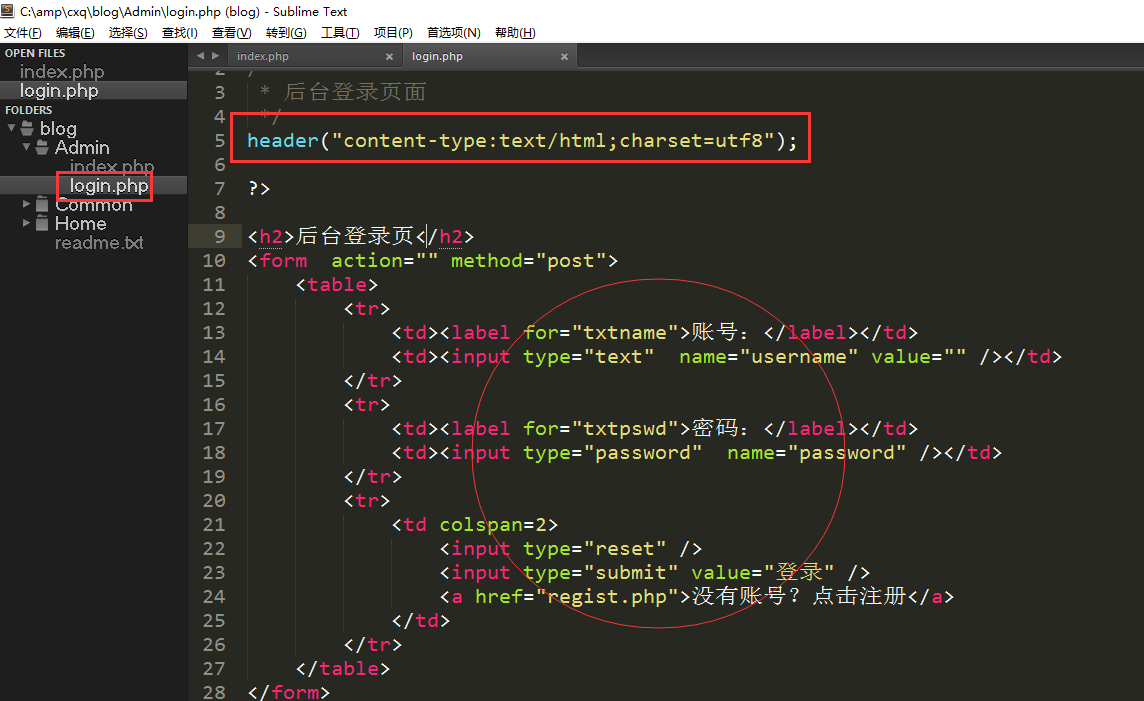
解决乱码问题，设置header头即可：



### 1.2后台登录模板

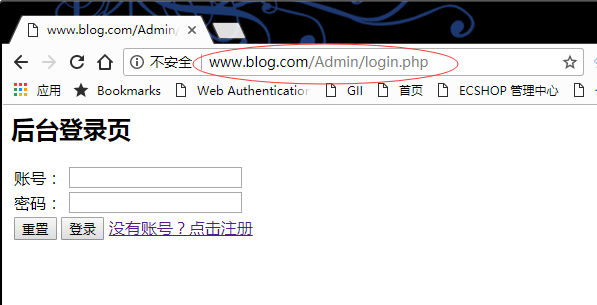
创建/Admin/login.php，放入预先制作的模板。

代码如下：



访问地址：<http://www.blog.com/Admin/login.php>

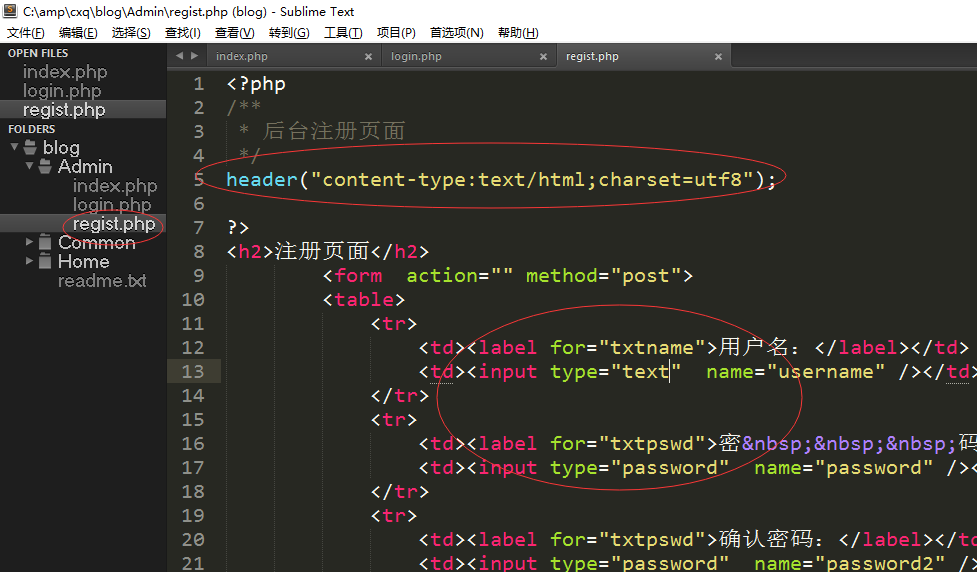
展示效果：



### 1.3后台注册模板

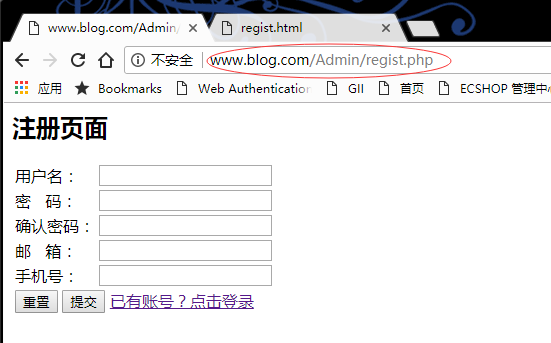
创建/Admin/regist.php，放入预先制作的模板。

代码展示如下：

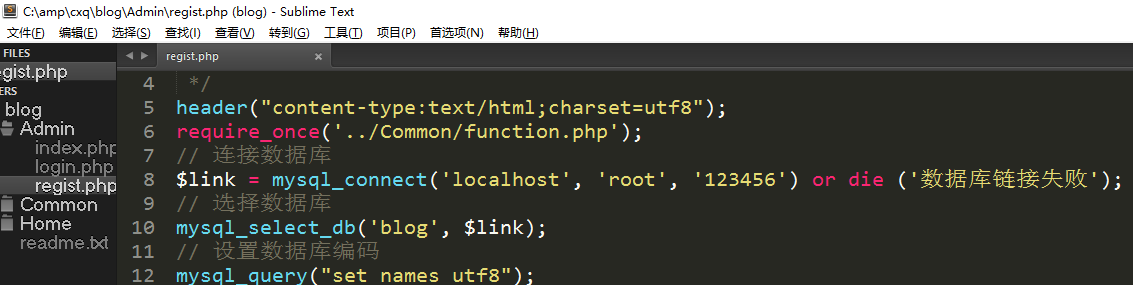
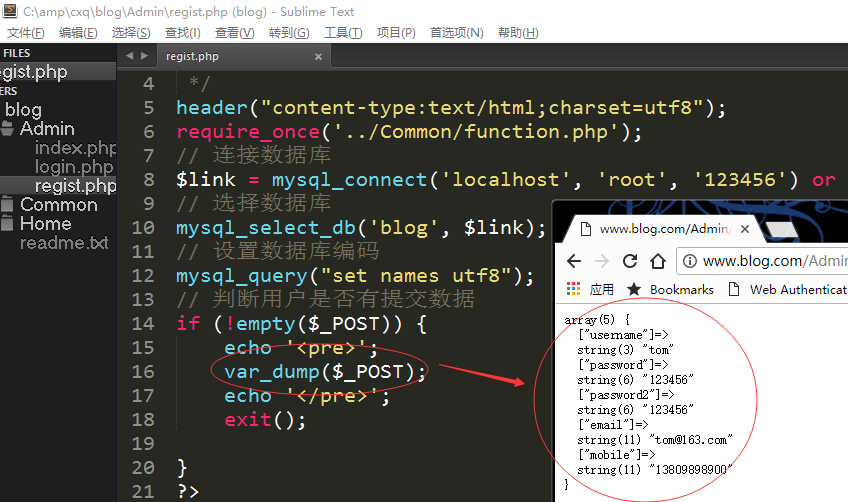


访问地址：<http://www.blog.com/Admin/regist.php>

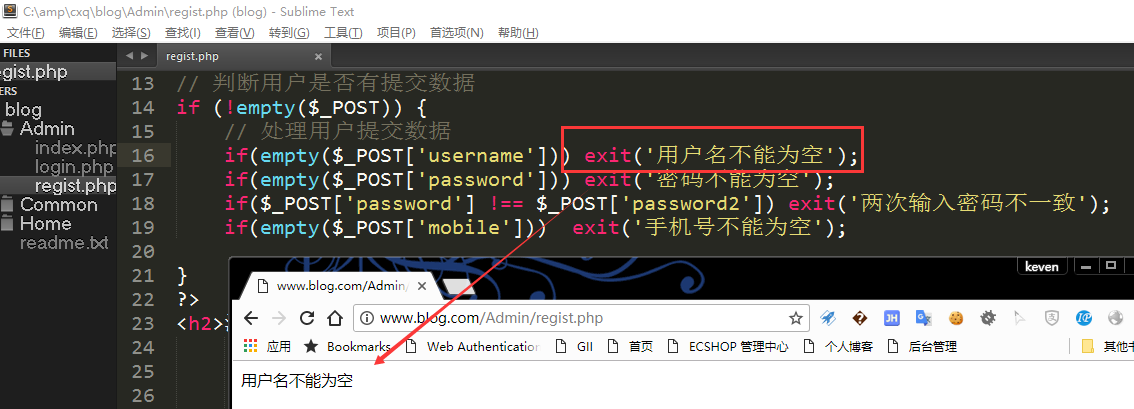
访问效果如下：



### 2.1注册功能实现

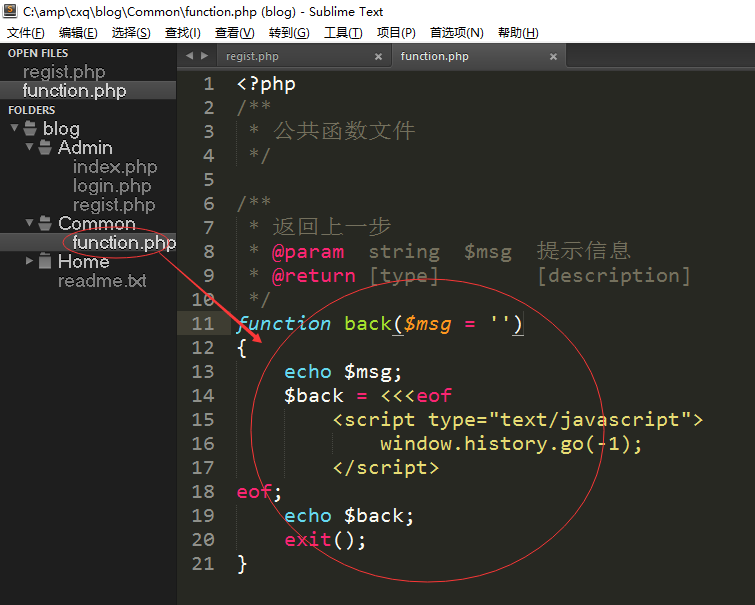
1. 连接数据库  
   
2. 判断用户提交的数据合法性  
   先去判断用户是否提交数据以及提交的数据信息，代码及展示效果如下图：  
   

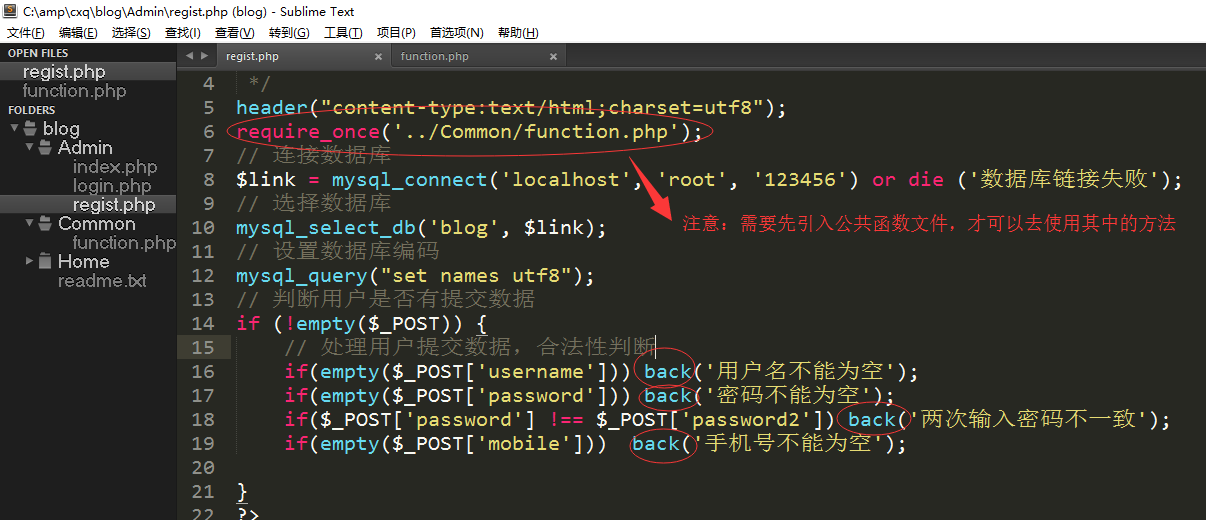
接下来判断用户的合法性：

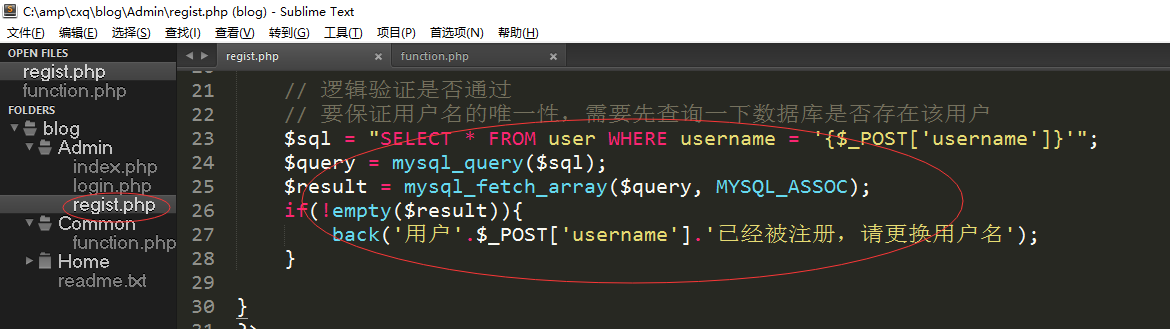
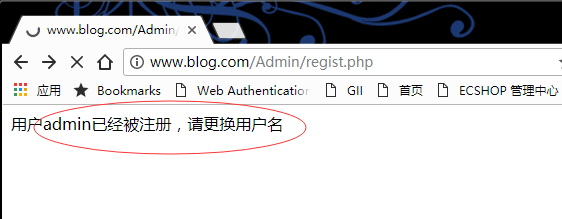
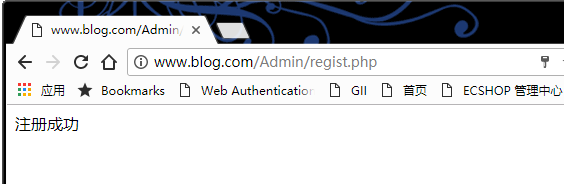
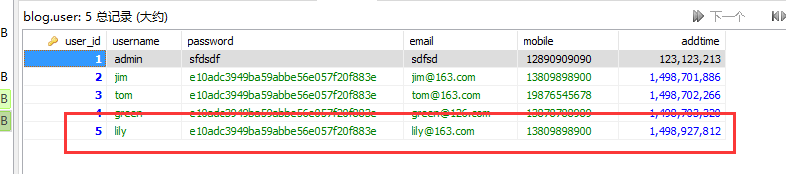


当用户输入的数据有误时需要告知用户并返回上一页，所以我们写一个自定义函数，让操作失败的时候返回上一页。

在Common下面创建function.php用于存放自定义公共函数，并创建back方法，如下图所示：



将注册页面的数据合法性验证使用的exit()方法改为back()方法，这样当用户合法性验证未通过之后就会在1秒之后返回上一步，具体代码修改如下：  


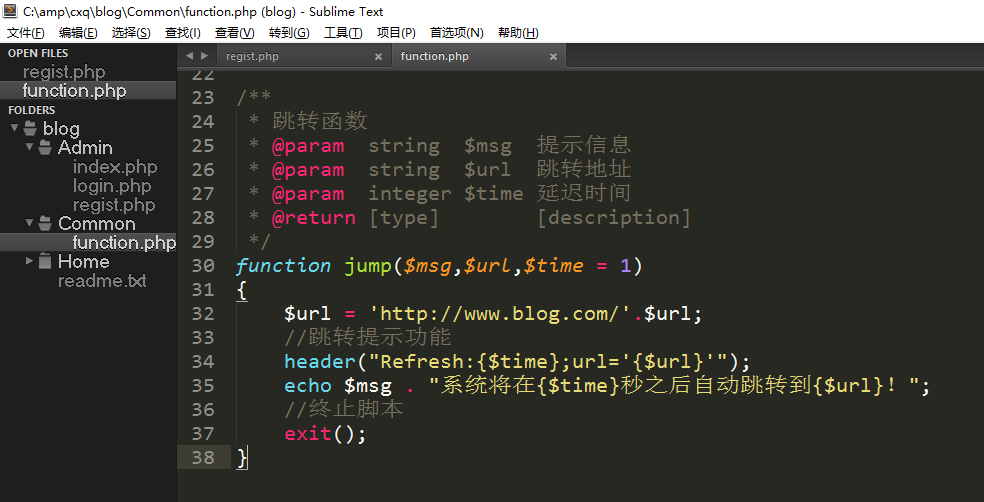
1. 判断数据的逻辑性是否通过  
   本次只需要验证用户名是否重复即可，如果用户名重复则不允许注册，返回上一步。**实现思路：根据用户名去user数据库查询是否存在该用户信息，如果存在说明该用户名已被占用，如果查不到说明该用户名可用。**  
     
   展示效果：  
   
2. **将用户信息入库**需要把需要处理的数据进行处理，如密码加密，时间戳等等。  
     
   页面展示效果：  
     
   数据库效果体现：  
   

### 2.2注册成功跳转

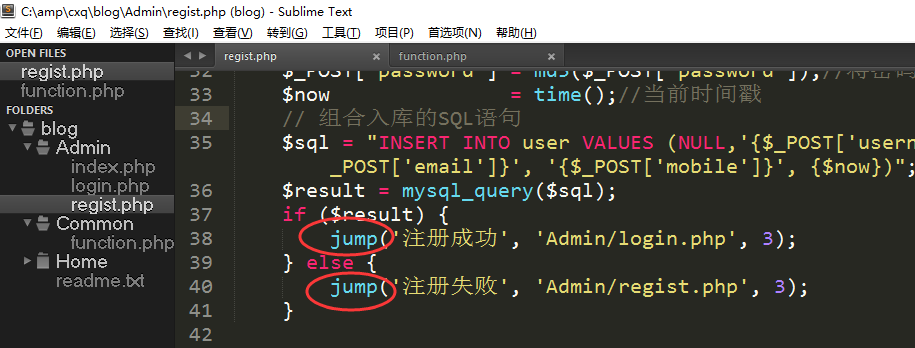
当用户注册成功之后往往需要跳转到登录页，这里我们用exit()或者back()方法都不太合适了，需要创建一个自定义跳转方法，使用header方法跳转。

在Common/function.php里创建jump方法

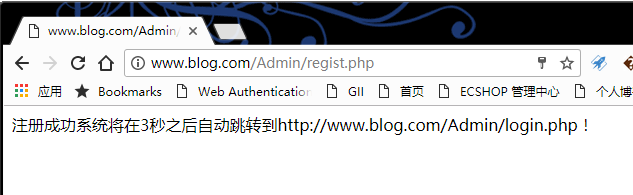
具体代码如下图。



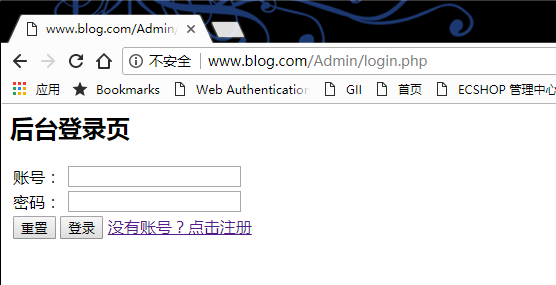
修改regist.php注册完成之后的代码：



展现的效果如下：



3秒之后跳转到登录页面：

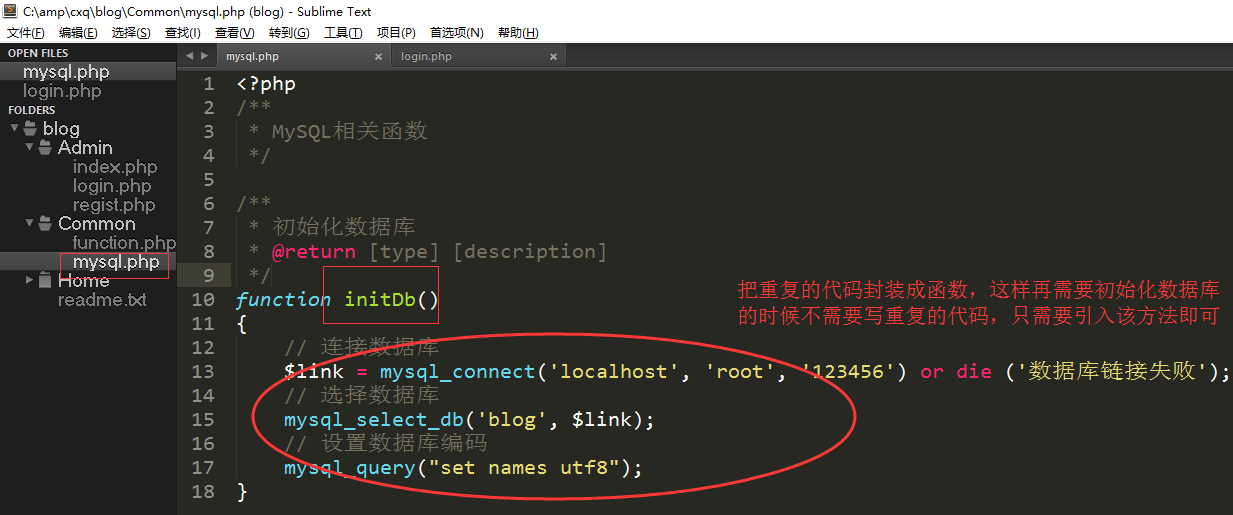


### 3.1自定义MySQL函数

登录页面同注册页面一样，都需要连接数据库，选择数据库等操作，为了避免写一些重复的代码，我们可以把初始化数据库的一些操作写到一个公共函数里面。

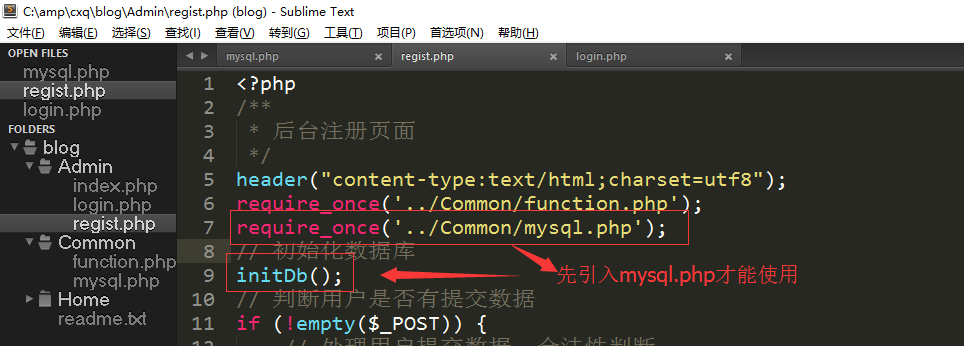
创建Common/mysql.php，此函数主要是用来操作数据库。

创建initDb函数，用来初始化数据库：



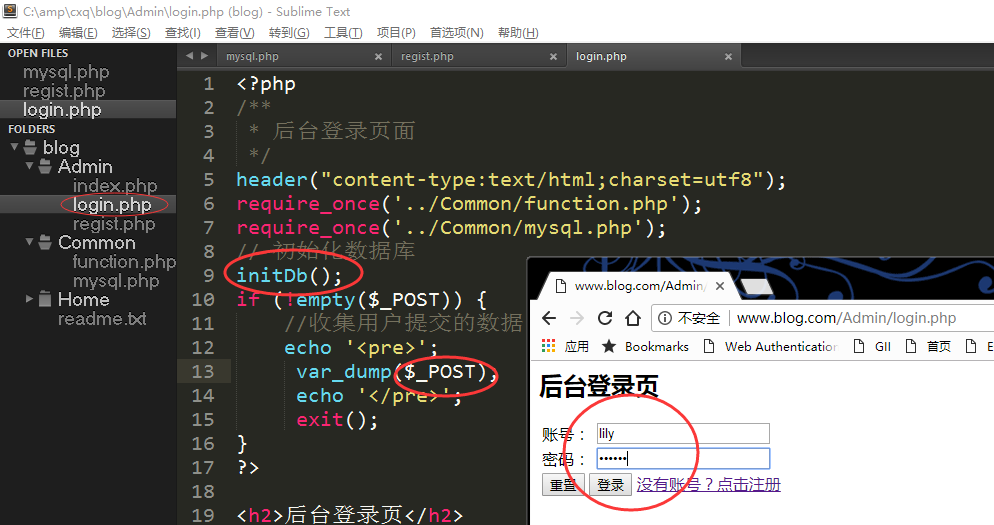
注意：在使用initDb方法之前不要忘了引入mysql.php

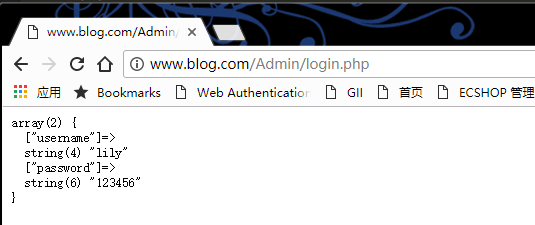
这样我们就可以把regist简化为如下代码：



### 3.2登录功能实现

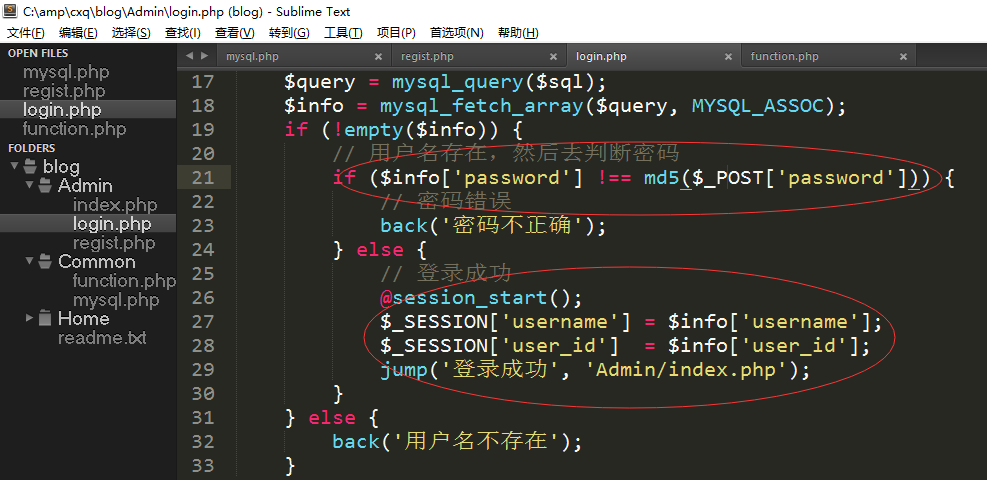
1) 打印收集到的数据

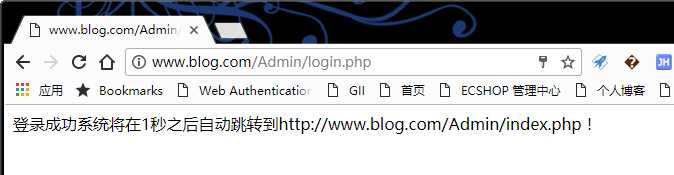




1. 判断用户提交数据的合法性  
   

3) 根据用户名查询用户信息  

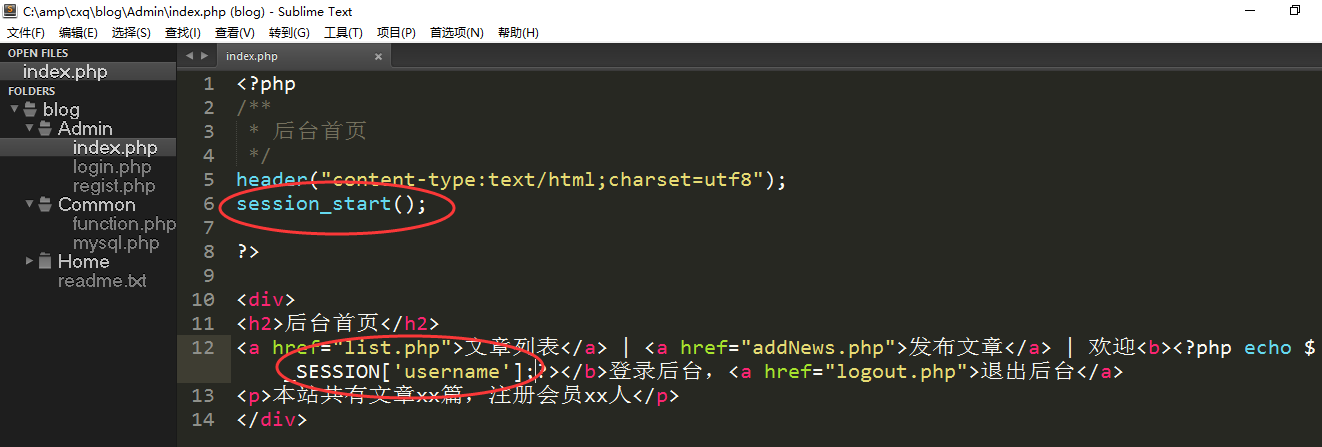

4) 用户名存在，则去判断密码是否正确，并持久化用户信息  
  
登录成功展示效果：



跳转到首页效果：  

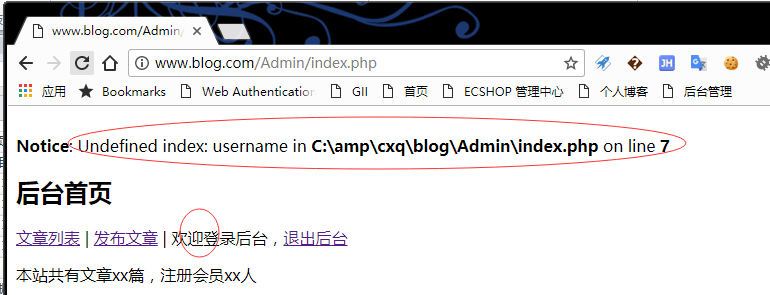

### 4.1后台首页实现

后台首页暂时只把当前登录后台的用户名给展示出来：



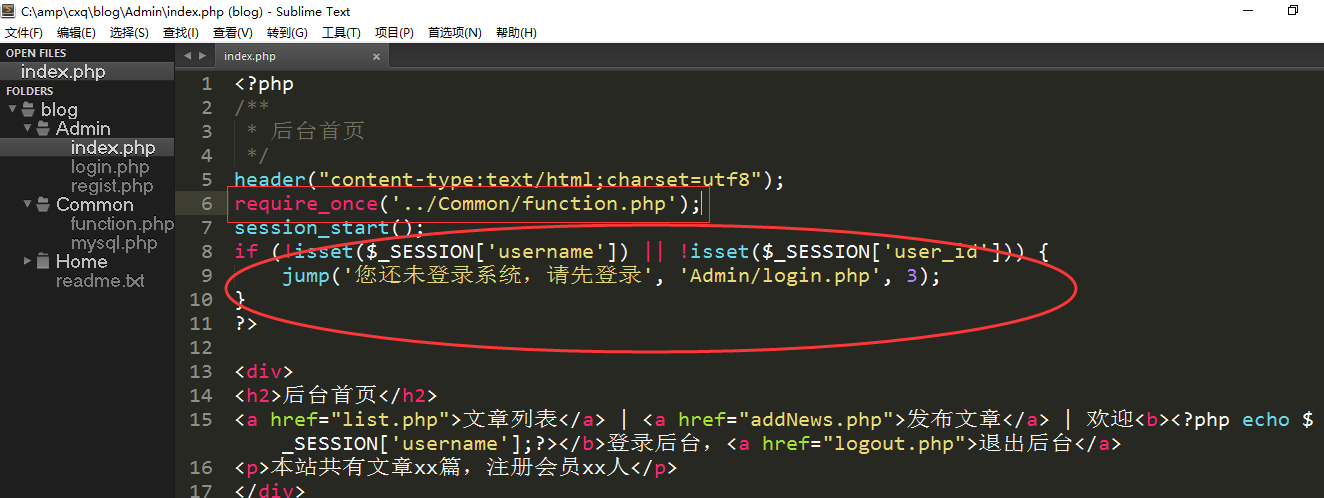
页面展示效果：



如果过一段时间Session过期了或者用户没有登录就会出现如下警告：  


这是因为Session中没有username的值了，会提示警告，需要做一个判断，让程序更严谨。当Session中不存在持久化的登录信息时提示用户跳转到登录页面。

代码如下：



页面效果展示：  
