**姓名：李志远 学号：20144864**

**任务重述：**

**1. 搭建一个本地web服务器, 访问指定页面能够显示当前服务器的资源信息(CPU/内存/磁盘占用率/网口的实时速率)2. 搭建一个本地留言系统, 访问页面能够填写发布人的名称-标题-内容, 也能够看到之前发布的历史留言3. 在2的基础上增加注册系统, 需要注册-登陆-验证后才能发布留言4. 在3的基础上增加管理员系统, 管理员能够管理注册账号和留言(一些常见的python web框架都会有后台管理功能, 要注意理解其原理和流程)5. 使用爬虫框架, 在留言系统的首页显示最新爬取的信息(选择自己感兴趣的网站爬吧, 最好不要都选同一个)**

我顺利完成任务1，2，3，4.

任务5我查阅资料推荐使用scrapy,python3安装scrapy爬虫框架费点劲，7.7日周五这天开始安装爬虫框架，开始学习scrapy，因此在攥写报告时任务5未完成，我会利用周末时间完成任务5并补全工作文档再发送一份

**学习收获：**

1. 使用django创建web工程以及app的命令

创建工程：dango-admin startproject projectname

创建app，在工程目录下: python manage.py startapp appname

1. 使用django框架开发简单的web应用：获取服务器资源信息，留言系统，用户登录认证系统
2. 前端开发的初步学习，对HTML，JavaScript，css，ajax的基础知识的学习与掌握，我最直观的感受:利用django开发，后端难度不大，学习他人的app或者查阅django官方文档，掌握了套路之后就比较容易。反而是前端，之前没有太多的前端开发经验，因此前端开发也费了不少时间去学习。
3. 了解了django的部分工作原理：

**①**当用户访问网页时,其用户浏览器会对访问的网页发出一个Request，网页接受请求后,会反馈用户浏览器一个Response对象

**②**一个django视图需要拥有视图函数,如果想要视图函数工作则需要把 url 映射l到视图. 一个django视图需要拥有视图函数,如果想要视图函数工作则需要把 url 映射l到视图。

**③**当请求一个页面时，Django会建立一个包含请求元数据的 [HttpRequest](http://www.cnblogs.com/#django.http.HttpRequest) 对象。 当Django 加载对应的视图时，[HttpRequest](http://www.cnblogs.com/" \l "django.http.HttpRequest" \o "django.http.HttpRequest) 对象将作为视图函数的第一个参数。每个视图会返回一个[HttpResponse](http://www.cnblogs.com/#django.http.HttpResponse) 对象。

**④**数据库表模型存储在models.py中，可在其中自定义一些数据库表结构。

**⑤**django相关参数配置可在setting.py中设定。比如加载的app在INSTALLED\_APPS中设定，数据库在DATABASES中设定，静态文件目录通过STATIC\_URL设定

1. Sqlite3数据库的常用命令的掌握，包括导入数据库，查看，创建数据表，数据库的增删改查操作
2. Django相关模块的使用，例如:使用jango表单，自定义forms.py设定表单结构
3. 对django自带认证后台的原理的理解.

**Web服务展示：**

**Web服务主页：**

****

**包括五个模块：**

**获取服务器相关信息**

**去留言**

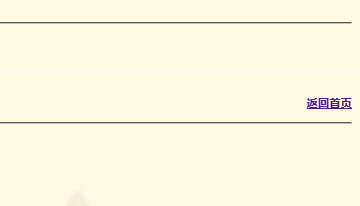
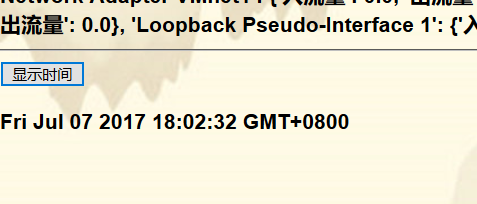
**登陆/注册后再留言**

**给站长留言**

**任务一：获取服务器相关信息：**

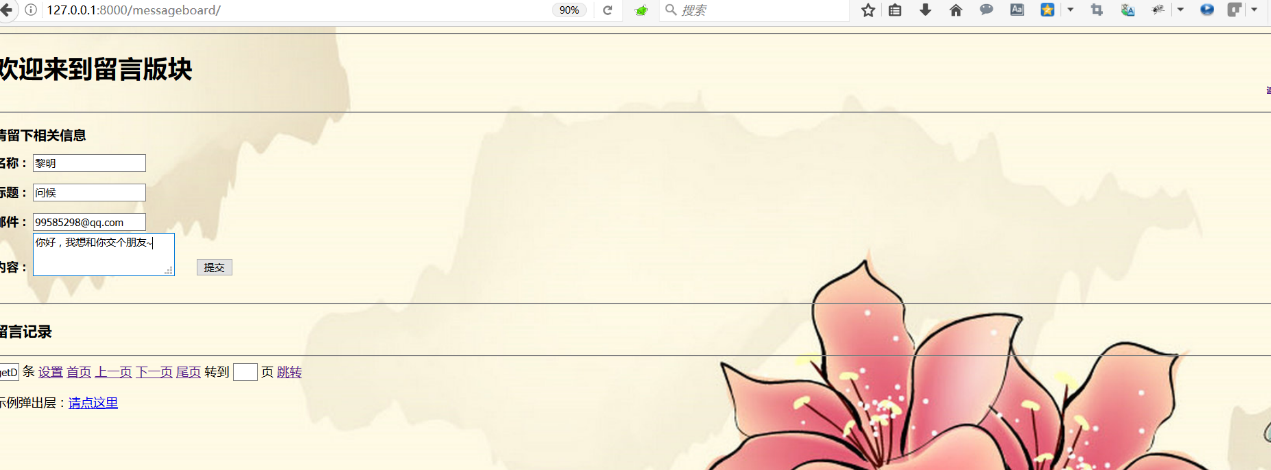
****

**点击显示时间，可实时获取系统当前时间，右上角连接可以返回首页**

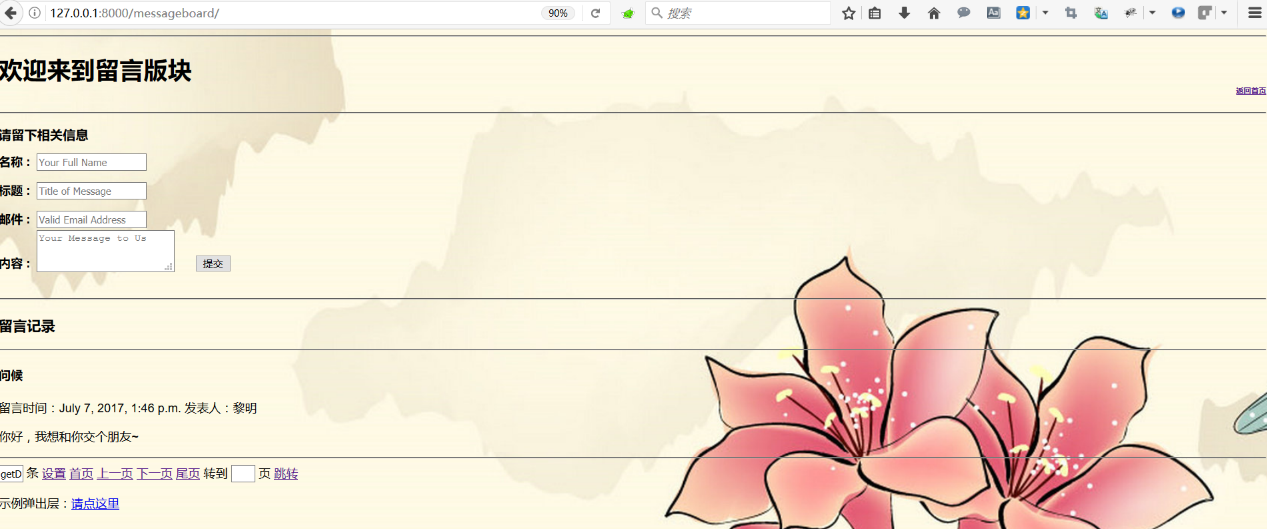
****

**任务二：留言(无须登陆或注册)**

**在留言板填入相关信息：**

****

**实时刷新以往的留言记录**：

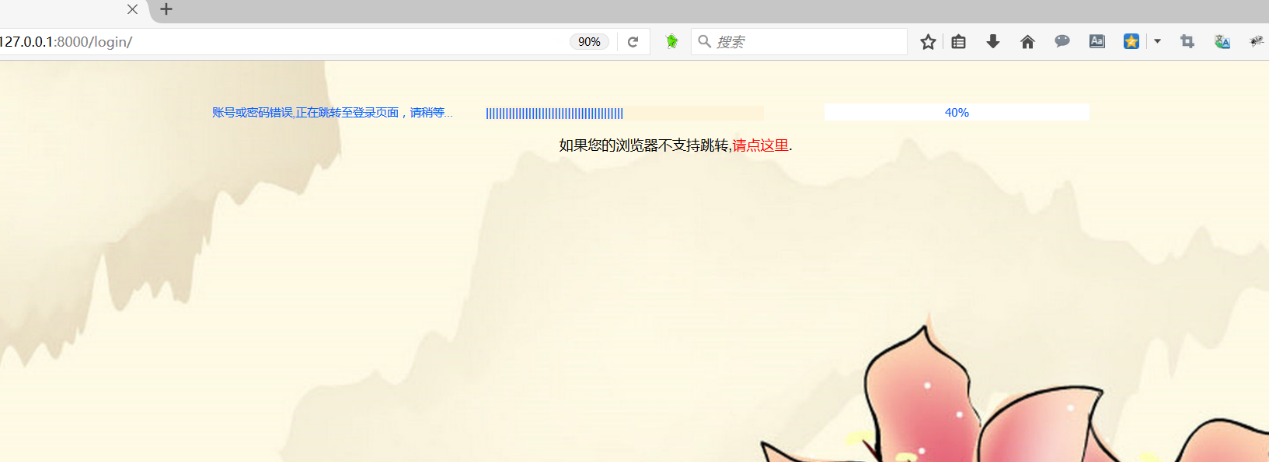
****

**任务三：先注册/登陆 再留言：**

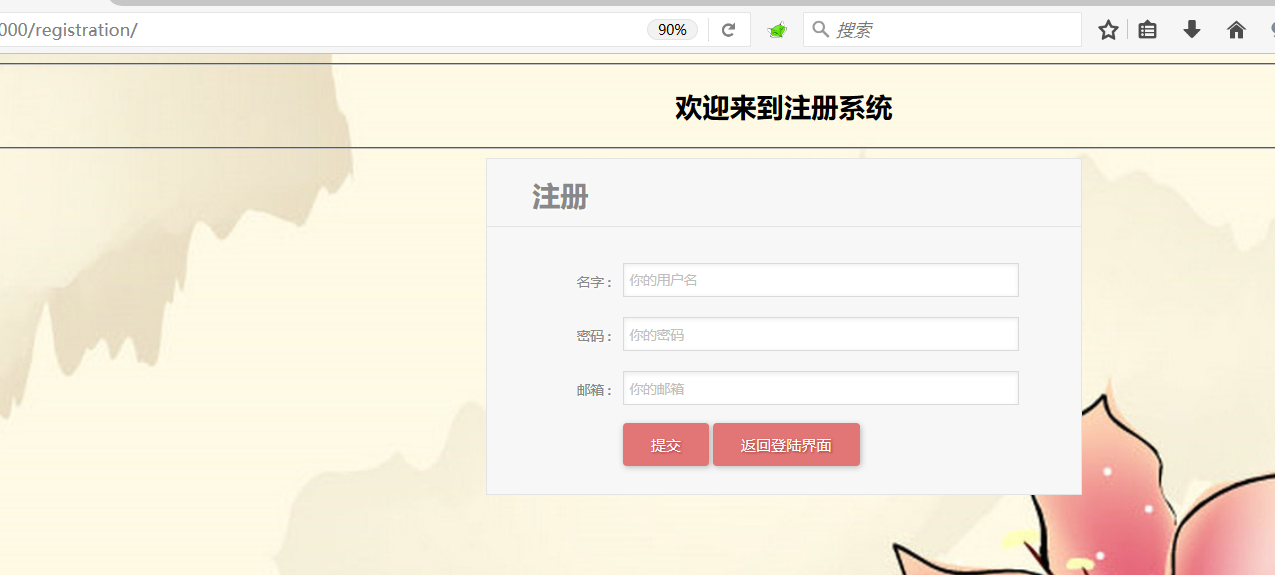
**登陆页面：**

****

**当名字和密码为空或与数据库不匹配时，倒计时跳转回登陆页面**

****

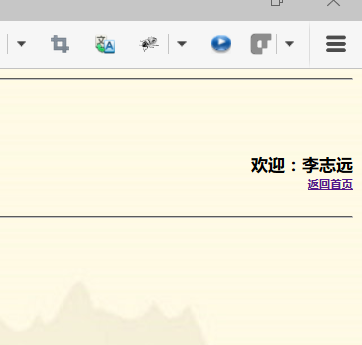
**注册模块：**

****

输入相关信息并提交：

****

**注册成功，跳转到留言板块，并自动填入名称，邮件部分**

****

**可以看到在注册成功后成功地提交了留言：**

****

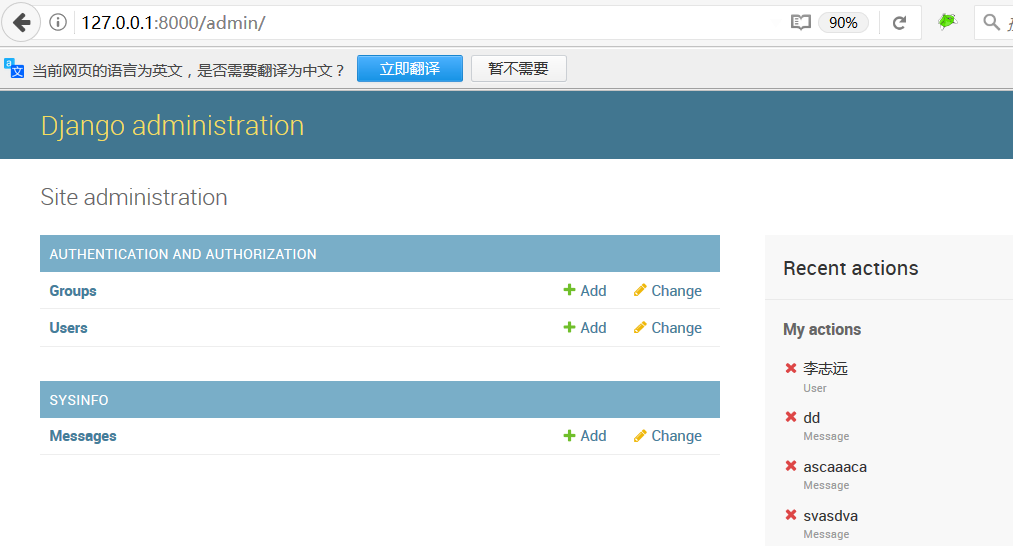
**同样的，这时候通过登录页面进入留言板块也是没问题的，这里不截图了**

**User数据表以username，也就是名称为主码，确定其唯一性，因此当在注册时输入重复名称，做了处理，提示用户名已存在，并跳转至注册页面**

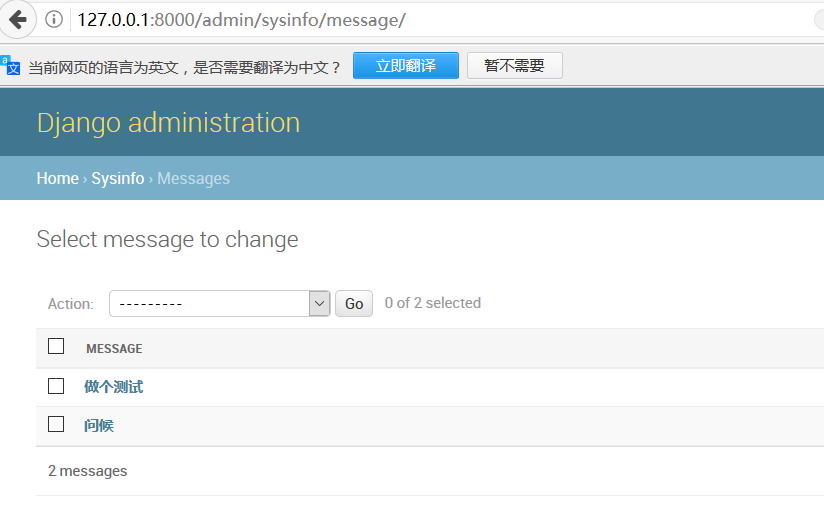
****

**任务四：后台管理**

**这里我用的是django默认后台管理，/admin/**

****

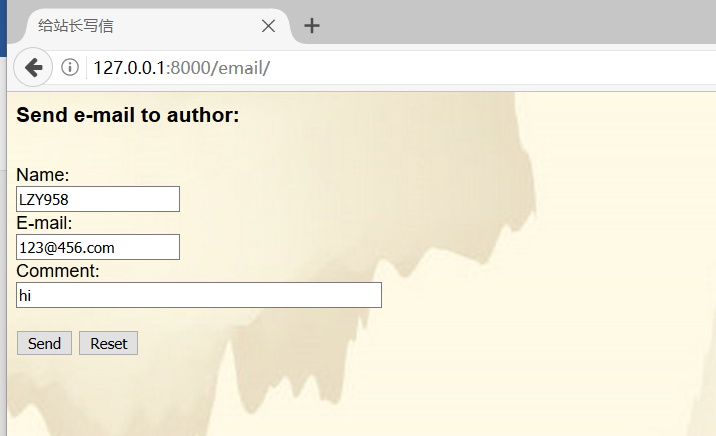
****

****

**之前的所有信息，都可以查到，这里超级用户是david 密码：lzy123456，可在后台对用户以及留言进行管理，以及密码更改等操作。**

**给站长留言：**

**我添加了这个模块为了让用户通过邮件向我反馈消息**

****

**在弹出的邮件中看见自己输入的内容**

****

**存在问题：**

**当留言过多时，在留言板块应该对留言进行分页展示（虽然页面上有分页连接，但实际功能并未完成），其实就是前端开发缺乏经验。**

**之前查阅资料，有个django-user2的第三方库对django User类进行扩展，本来是要用那个的，但调试许久也没成功，暂时搁置了。**