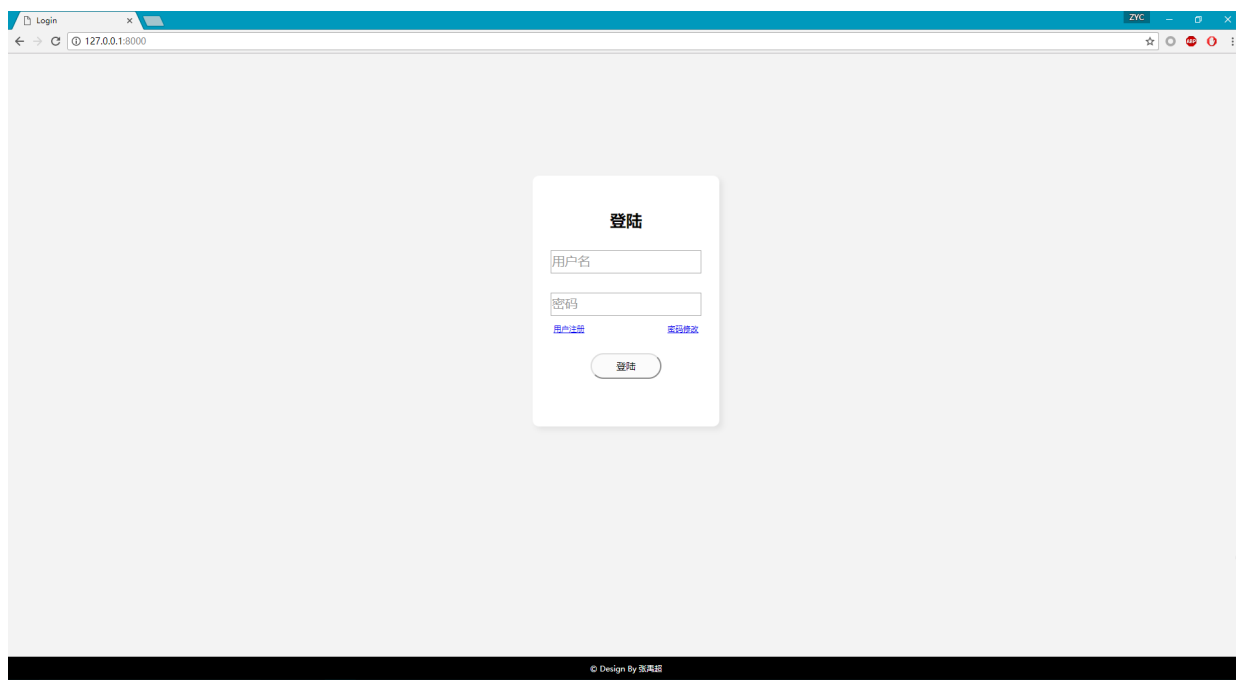


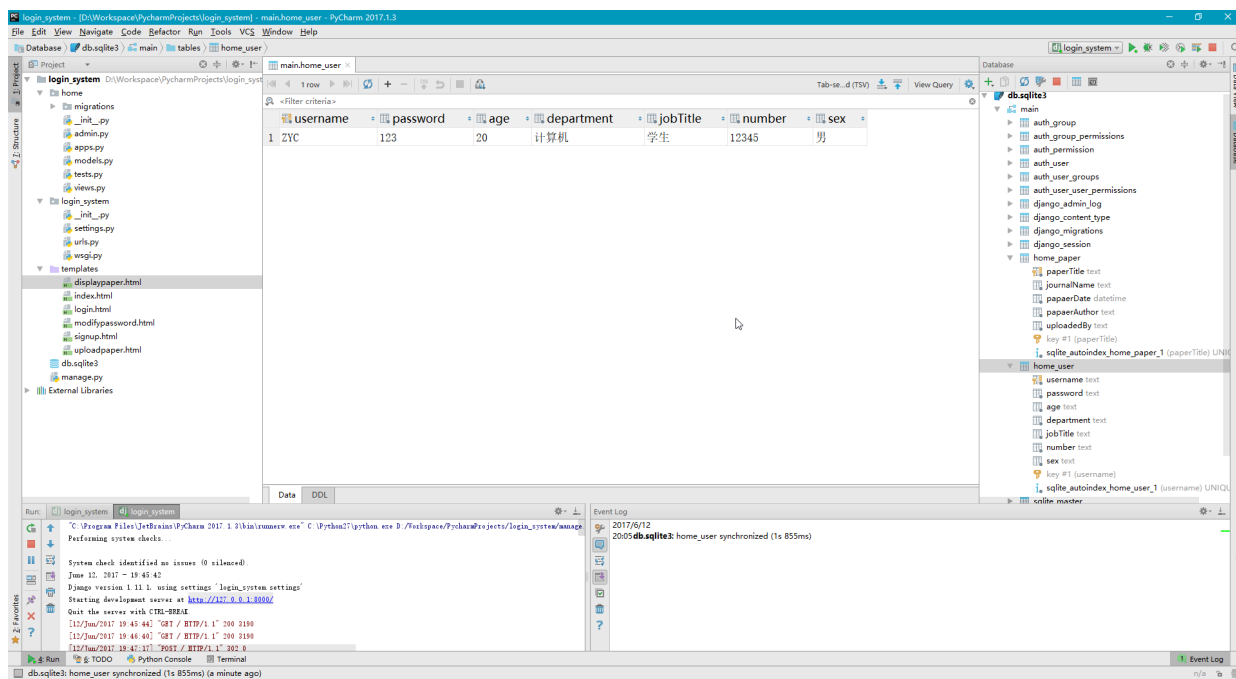
# 论文信息录入系统

## 一、功能演示

### 1. 主界面



此时数据库用户表中只有一条之前测试用的数据



### 2. 注册界面

点击首页的 **用户注册** ，进入用户注册界面

The screenshot shows a web browser window with the address bar displaying "127.0.0.1:8000/signup". The page content is a registration form titled "注册". The form contains the following fields and values:

- 工号: 12345
- 用户名: 李老師
- 性别: 男
- 年龄: 26
- 所在部门: 计算机
- 职称: 教師
- 密码: ...
- 再次输入密码: ...

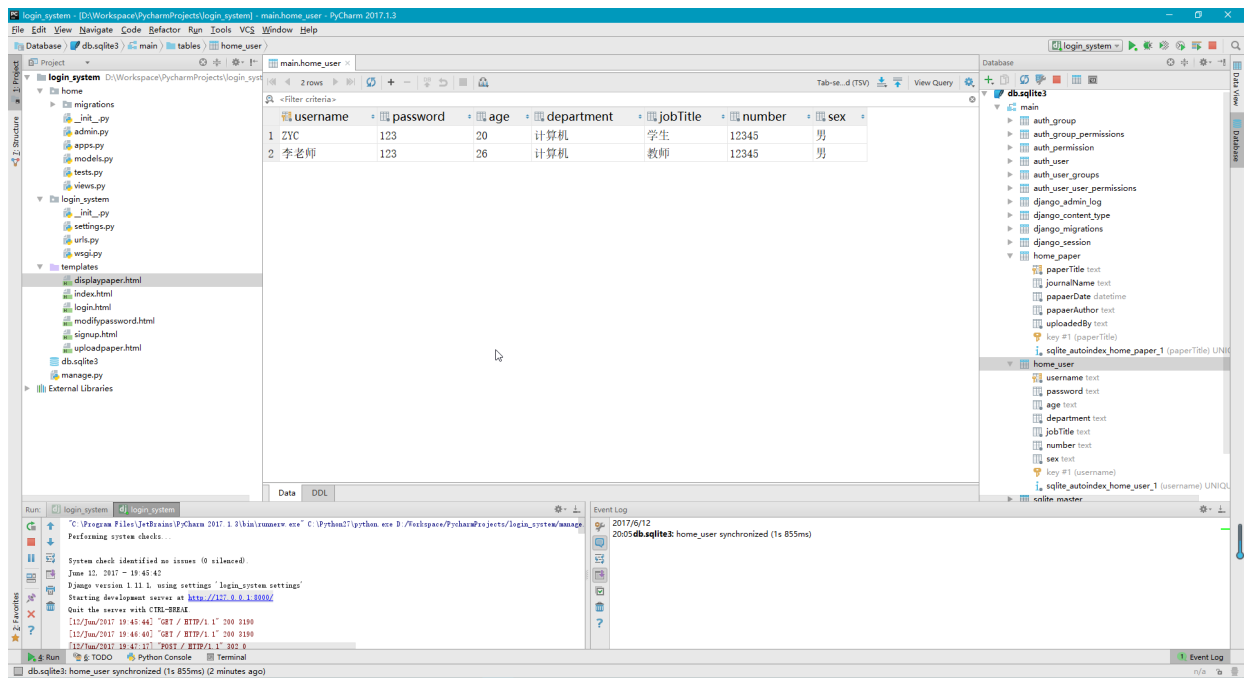
A "注册" button is located at the bottom of the form.

The screenshot shows a web browser window with the address bar displaying "127.0.0.1:8000/signup". The page content is a registration form titled "注册". The form contains the following fields and values:

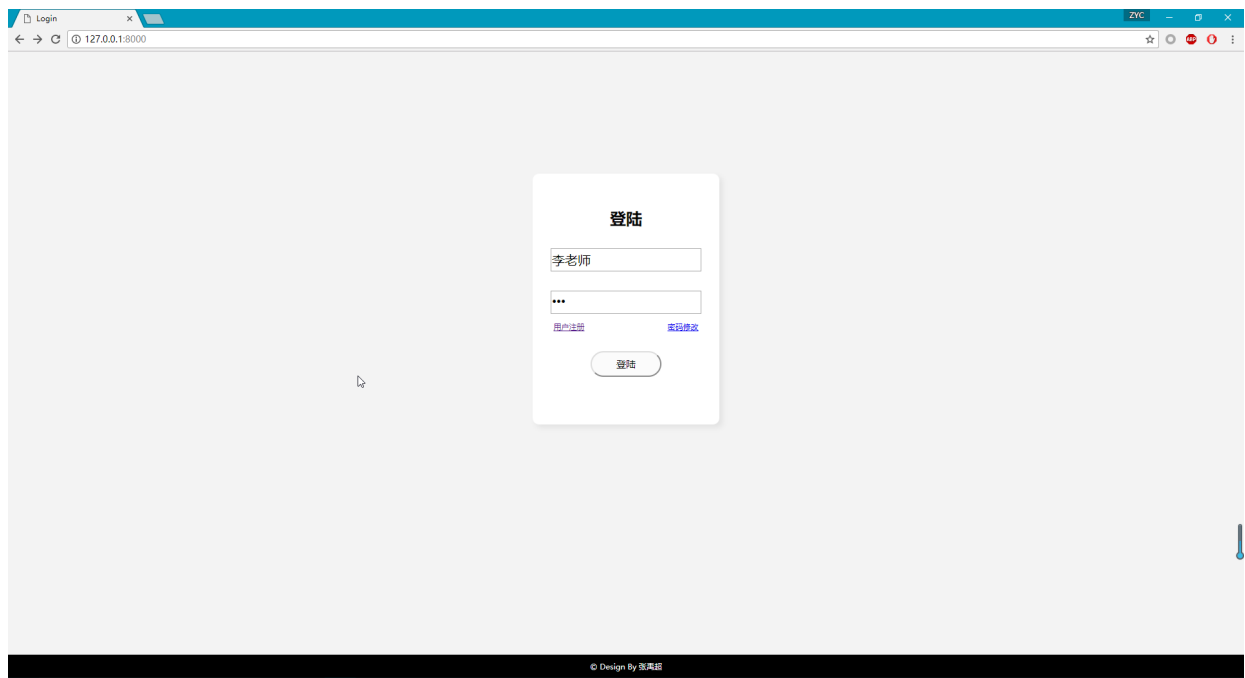
- 工号: 12345
- 用户名: 李老師
- 性别: 男
- 年龄: 26
- 所在部门: 计算机
- 职称: 教師
- 密码: ...
- 再次输入密码: ...

A "注册" button is located at the bottom of the form.

刷新数据库后可以看见刚才注册的用户数据



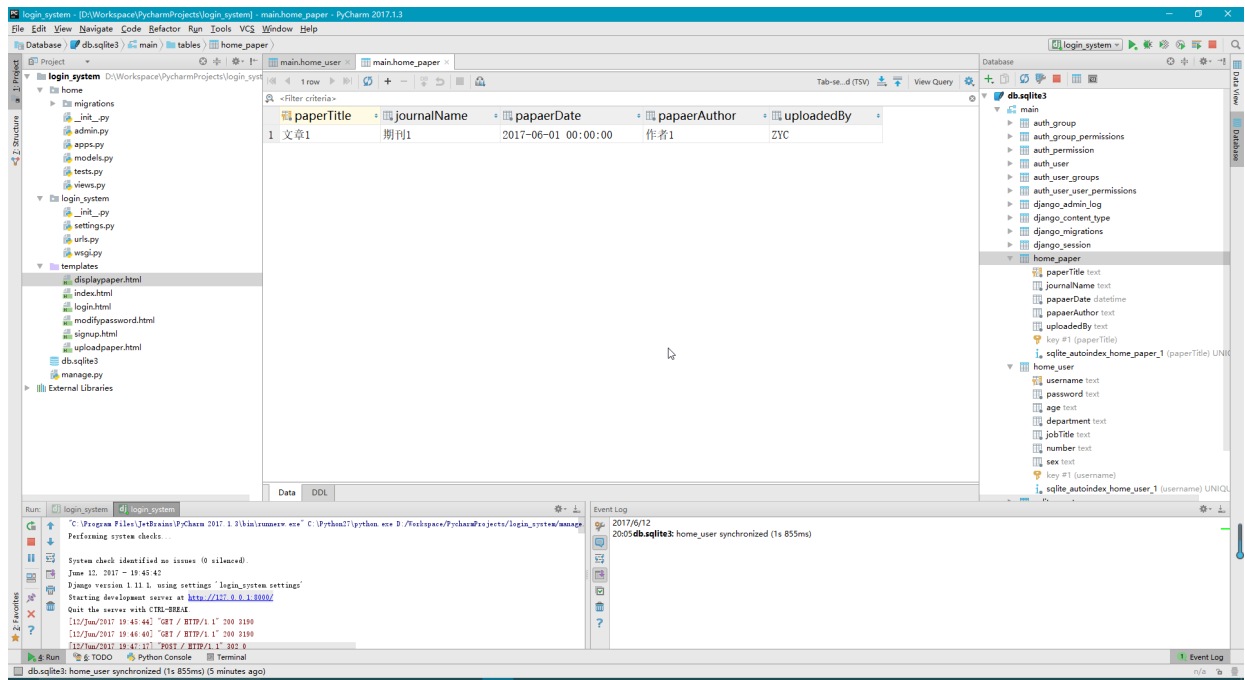
### 3. 登陆



若该用户未录入过论文则登陆成功后会跳转到科研论文录入界面



同样，此时论文表中只有一条测试数据



## 4. 论文信息录入

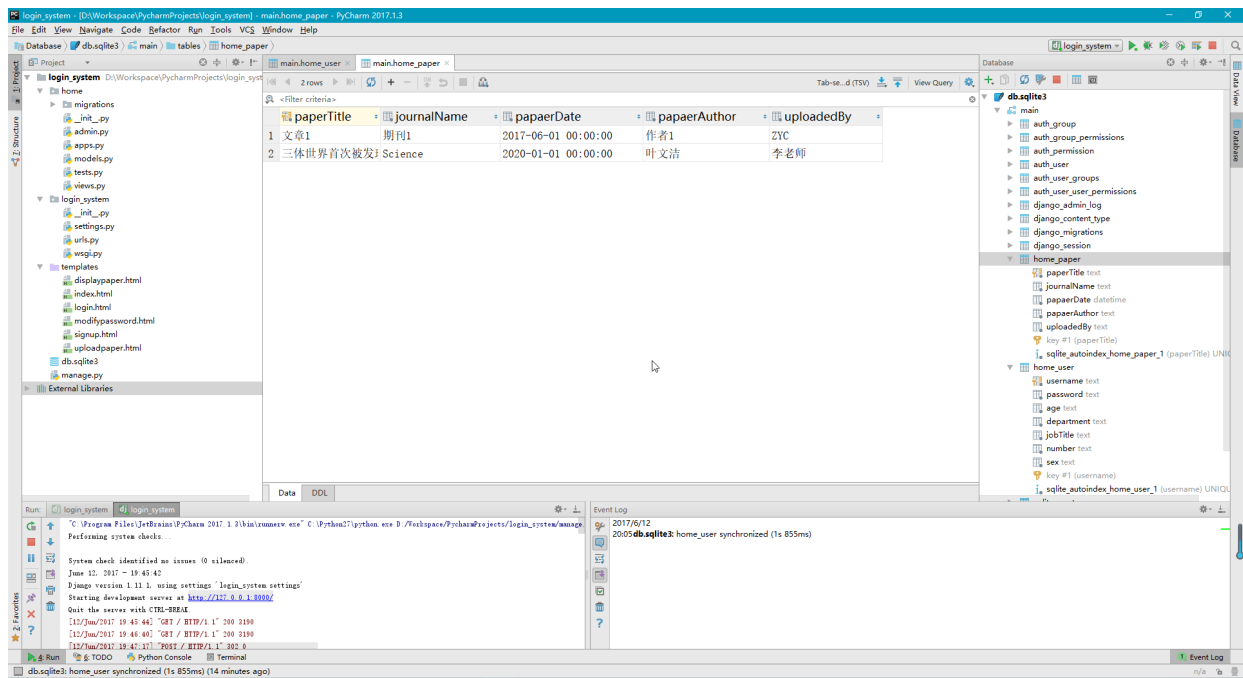
输入论文信息



点击提交后，跳转到 **论文信息展示** 界面。（此界面显示登陆用户上传的所有论文信息）



此时刷新数据库论文表，新的一条信息已经写入



用户此时可以选择继续添加新的论文信息，或者退出登陆

## 5. 退出登陆

点击推出登陆按钮，对用户进行登出操作，提示 退出登录



【验证】此时访问 论文信息展示 界面以及 论文信息录入 界面均会显示 未登录



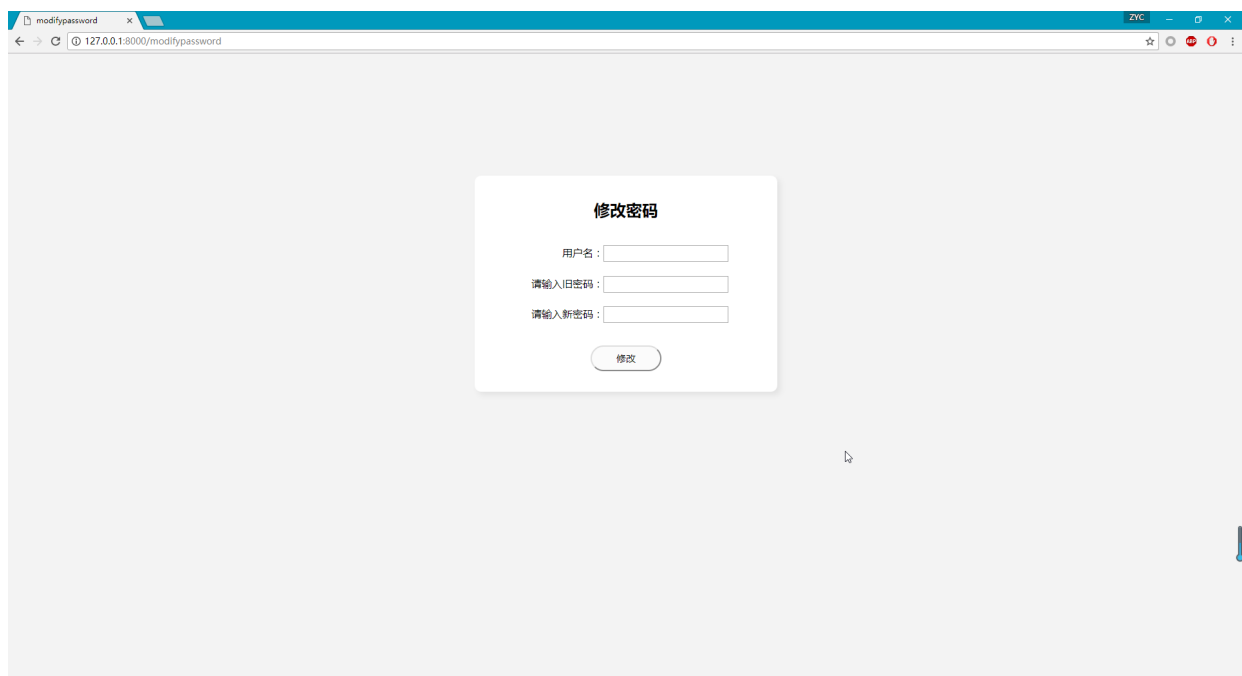
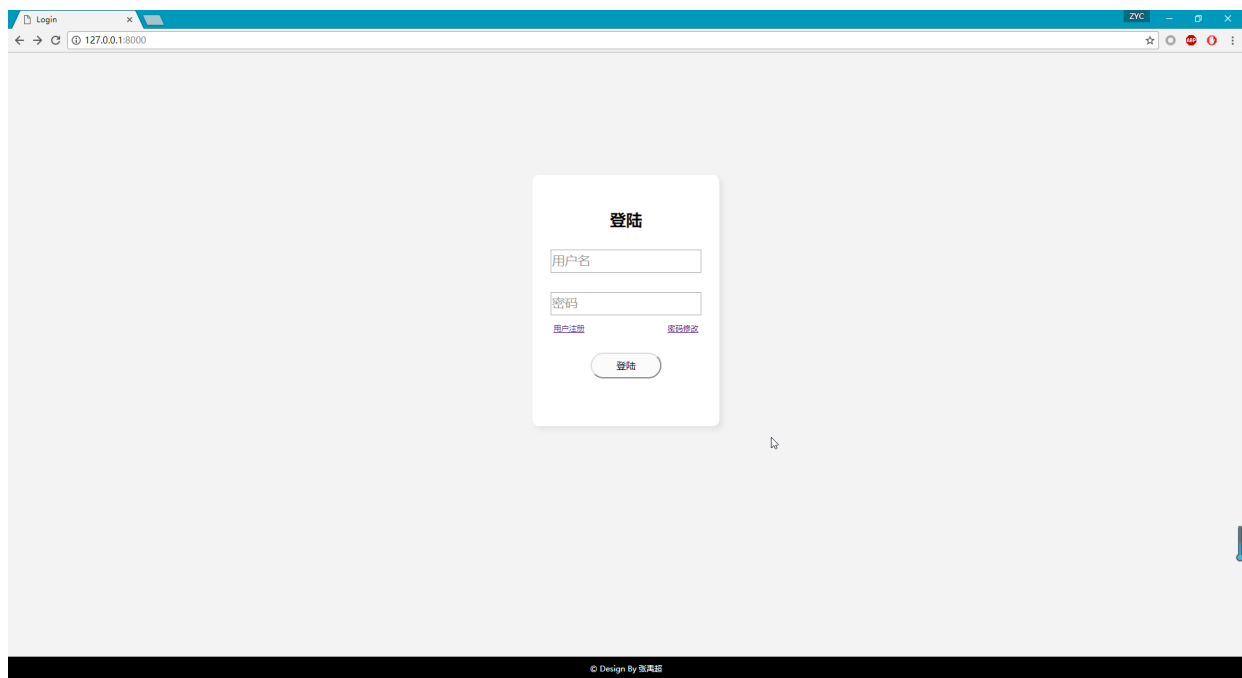
未登录



未登录

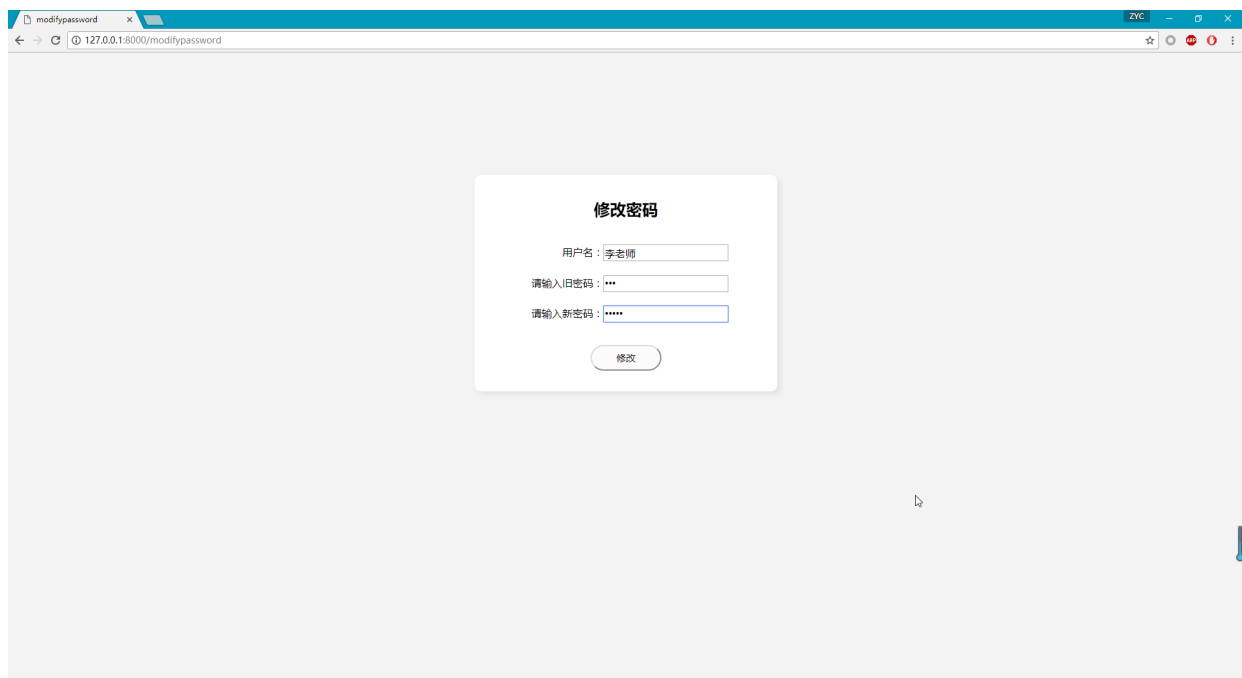
## 6. 密码修改

点击主界面的 密码修改

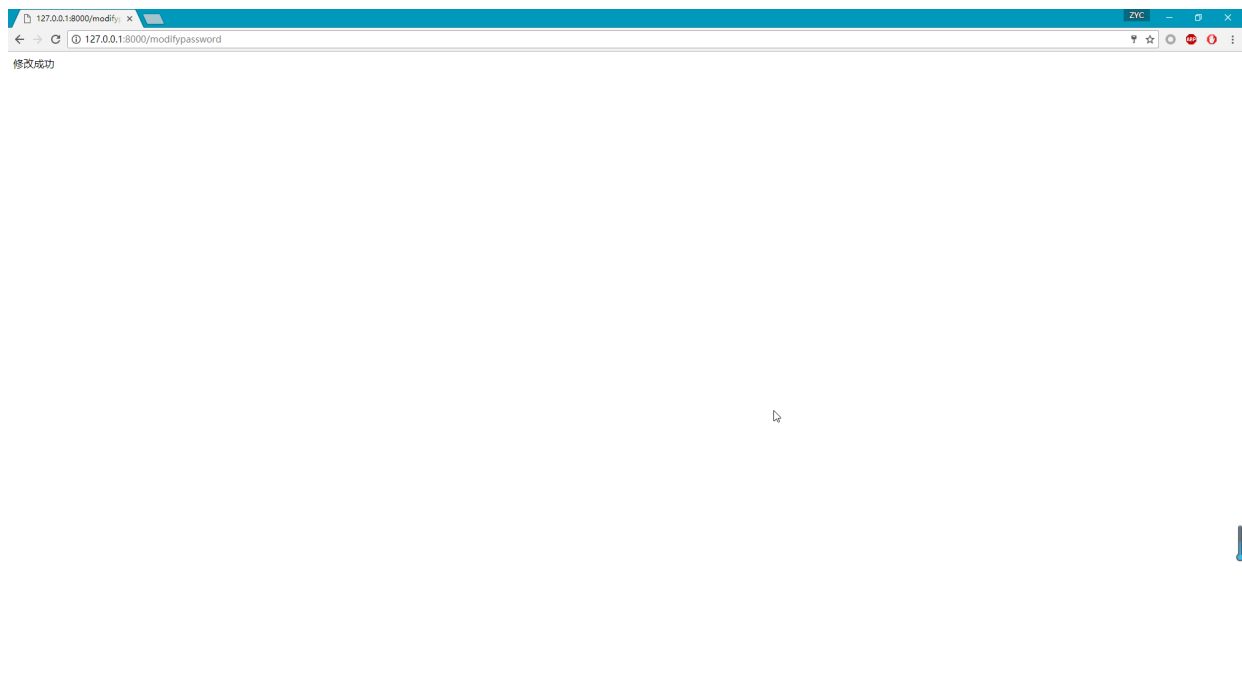


输入信息

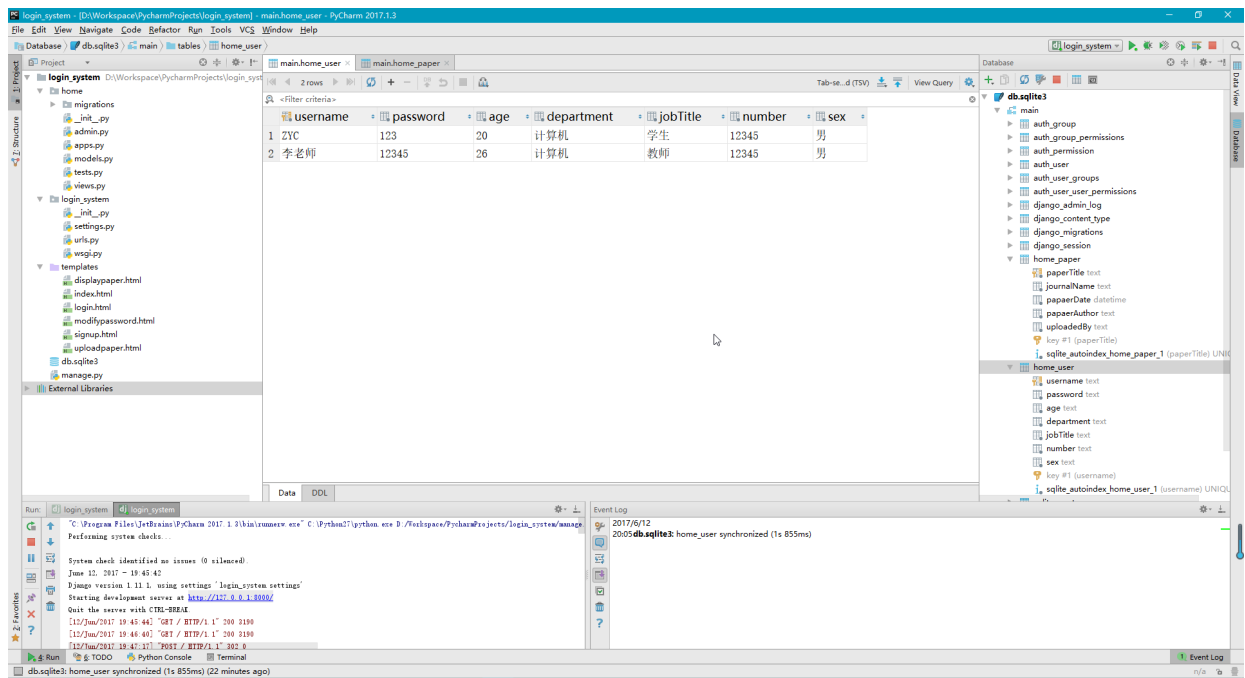




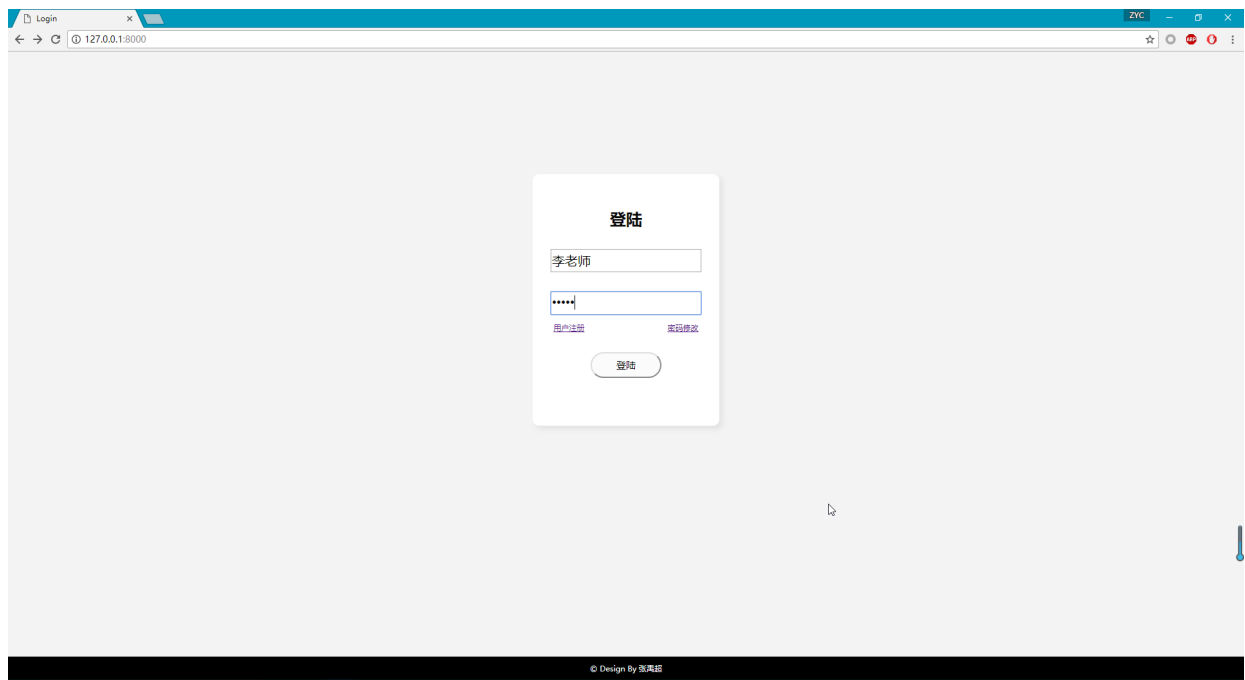
点击修改按钮后，提示 修改成功



刷新数据库，此时密码已经从 123 改为 12345



返回主界面，尝试用新密码登陆



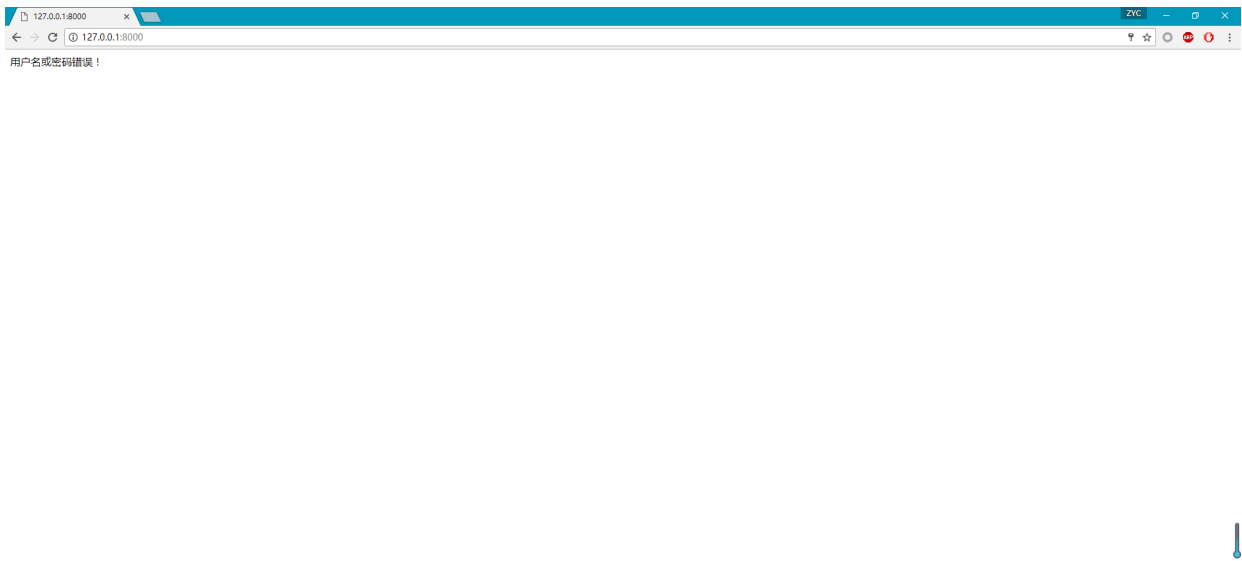
登陆成功！由于 李老师 这个用户已经录入过论文信息，所以登陆成功后跳转到 论文信息展示 界面



## 7.错误提示

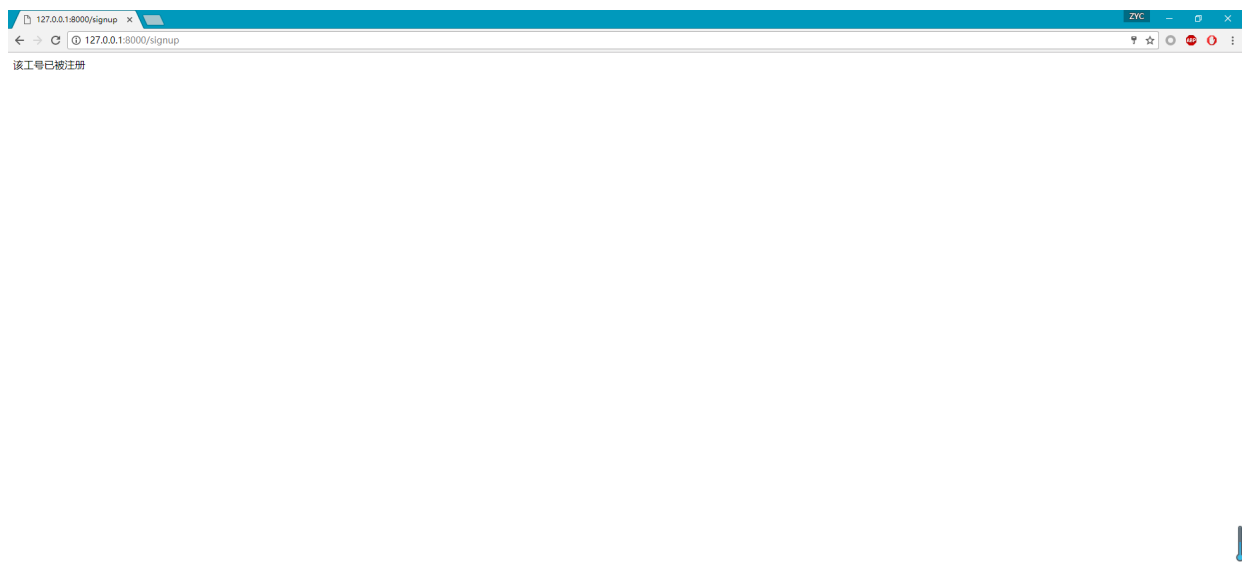
主界面

用户名或密码错误

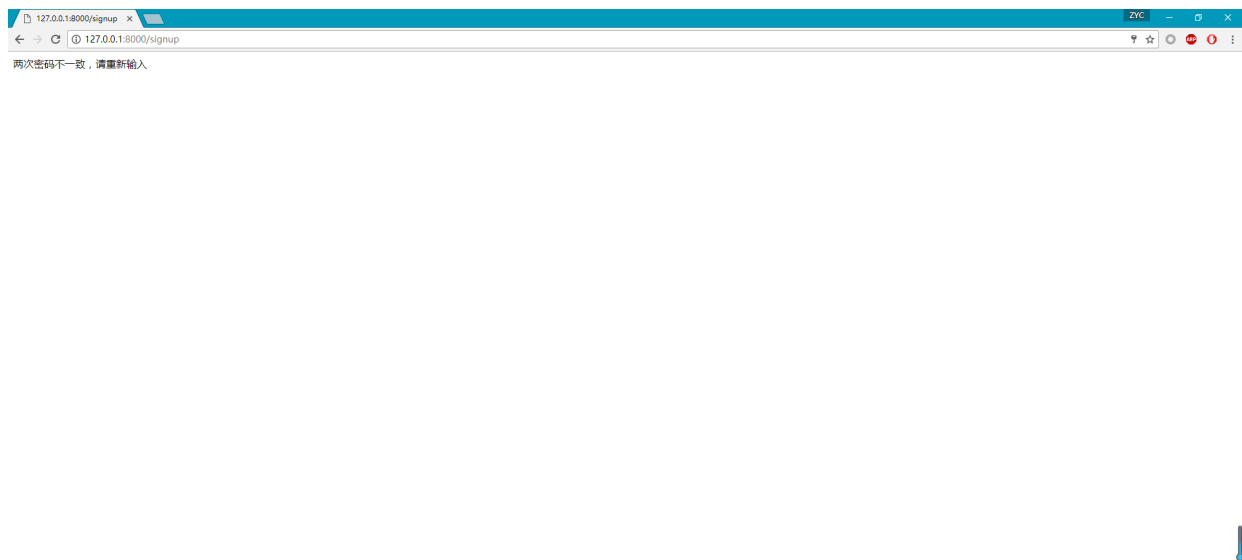


注册界面

工号重复



## 两次密码输入不一致



## 论文信息录入界面

### 文章名称重复



## 密码修改界面

原始密码或用户名错误



新密码与旧密码一致



演示结束

## 二、介绍

本系统以Python为主要编程语言，采用Django Web框架。

### 逻辑

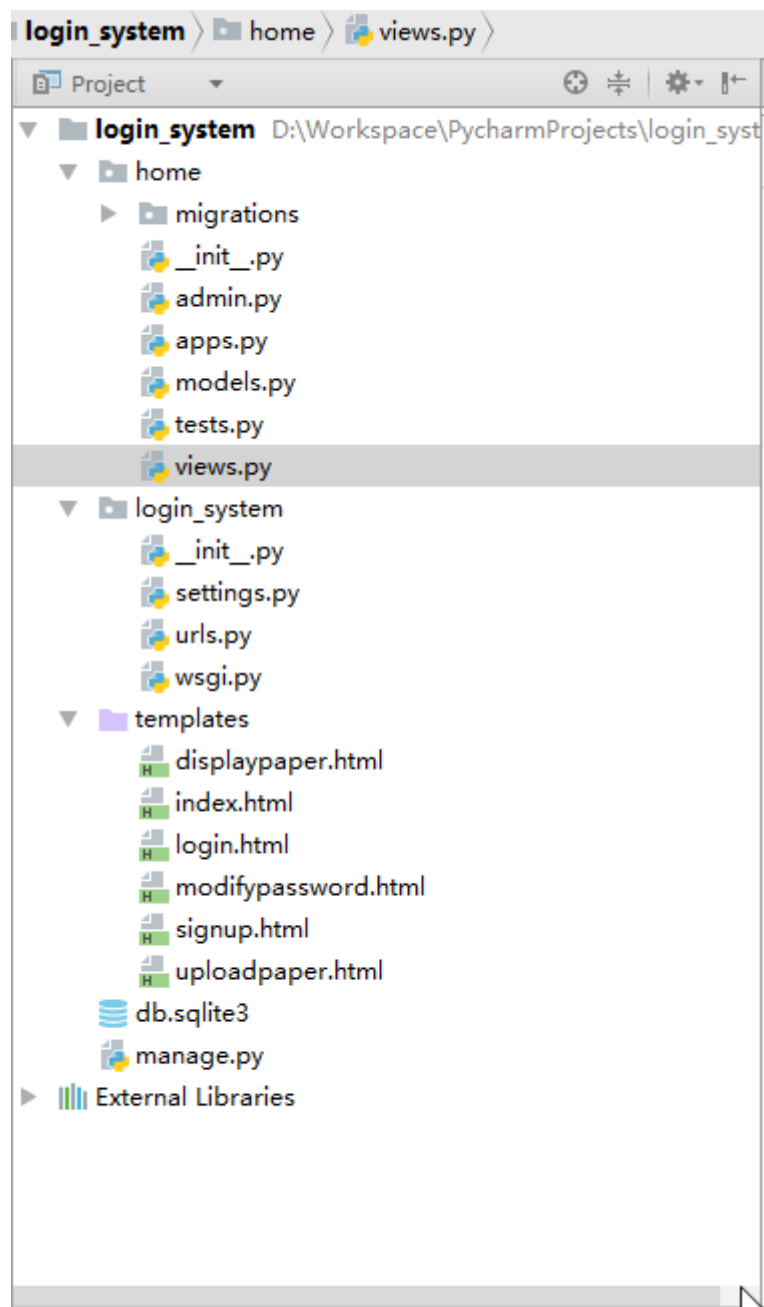
- 注册、登陆、修改密码采用POST方法，其余用到表单的部分为GET方法。
- 在判断逻辑部分将基本的几种错误情况（名称重复、密码不一致等）都考虑了进去，可能还有不足，欢迎指点。
- 登陆后以session[username]来检查用户是否登陆，同时进行跨网页之间的通讯。

### MVC

MVC全名是Model View Controller，是模型(model) - 视图(view) - 控制器(controller)的缩写，一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。

- Model（模型）表示应用程序核心（比如数据库记录列表）。
- View（视图）显示数据（数据库记录）。
- Controller（控制器）处理输入（写入数据库记录）。

从文件树种可以看出，本系统遵从MVC模型



`models` 为模型层，数据库操作相关

```
models.py ×
1  # -*- coding: utf-8 -*-
2  import ...
3
4
5
6
7  # Create your models here.
8  class User(models.Model):
9      number = models.TextField(primary_key=True)
10     username = models.TextField()
11     sex = models.TextField()
12     age = models.TextField()
13     department = models.TextField()
14     jobTitle = models.TextField()
15     password = models.TextField()
16
17
18 class Paper(models.Model):
19     paperTitle = models.TextField(primary_key=True)
20     journalName = models.TextField()
21     papaerDate = models.DateTimeField()
22     papaerAuthor = models.TextField()
23     unloadeBy = models.TextField()
```

views.py 为逻辑处理部分

```
views.py ×
uploadpaper() else
1  # -*- coding: utf-8 -*-
2  from __future__ import unicode_literals
3  from models import User, Paper
4  from django.http import HttpResponseRedirect
5  from django.shortcuts import render
6
7
8  # Create your views here.
9  def index(request):
10     return render(request, 'index.html')
11
12
13  def login(request):
14     username = request.POST.get('username') # username = request.POST['username']
15     password = request.POST.get('password')
16     if username == '' or username is None or password == '' or password is None:
17         return render(request, 'login.html')
18     else:
19         try:
20             user = User.objects.get(username=username, password=password)
21             except:
```

urls.py 由URL渲染指定模板视图

templates 为视图层，模板部分（可用简单模板语言）



```
displaypaper.html x
html body table.container tr
45 </style>
46 </head>
47 <body>
48 <h1>{{ username }}上传的论文信息</h1>
49 <table class="container" border="1" cellpadding="14">
50 <tr><th>文章名称</th><th>期刊名称</th><th>发表时间</th><th>第一作者</th><tr>
51 {% for i in paperList %}
52 <tr>
53 <td>{{ i.paperTitle }}</td>
54 <td>{{ i.journalName }}</td>
55 <td>{{ i.papaerDate }}</td>
56 <td>{{ i.papaerAuthor }}</td>
57 </tr>
58 {% endfor %}
59 </table>
60 <div class="container">
61 <a href="uploadpaper"><button id="button-addPaper" class="button">添加论文</button></a>
62 <a href="signout"><button id="button-signOut" class="button">退出登陆</button></a>
63 </div>
64 </body>
```

```
views.py x
signup() if User.objects...
1 # -*- coding: utf-8 -*-
2 from __future__ import unicode_literals
3 from models import User, Paper
4 from django.http import HttpResponse, HttpResponseRedirect
5 from django.shortcuts import render
6
7
8 # Create your views here.
9 def index(request):
10     return render(request, 'index.html')
11
12
13 def login(request):
14     username = request.POST.get('username') # username = request.POST['username']
15     password = request.POST.get('password')
16     if username == '' or username is None or password == '' or password is None:
17         return render(request, 'login.html')
18     else:
19         try:
20             user = User.objects.get(username=username, password=password)
21         except:
```

## 数据库

数据库为 **sqlite3**

Uer——用户表

number ( 工号 ) 为主键

Paper——论文表

paperTitle ( 文章名称 ) 为主键