

**留言板系统开发设计说明文档**

( 2019--2020年度第二学期)

名 称： Python期末大作业

题 目： 留言板系统

院 系： 控制与计算机工程学院

班 级： 软件1801

学 号： 120181080416

学生姓名： 王子轩

指导教师： 熊建国

成 绩：

日期： 2020 年 6 月 25 日

1. **题目描述**

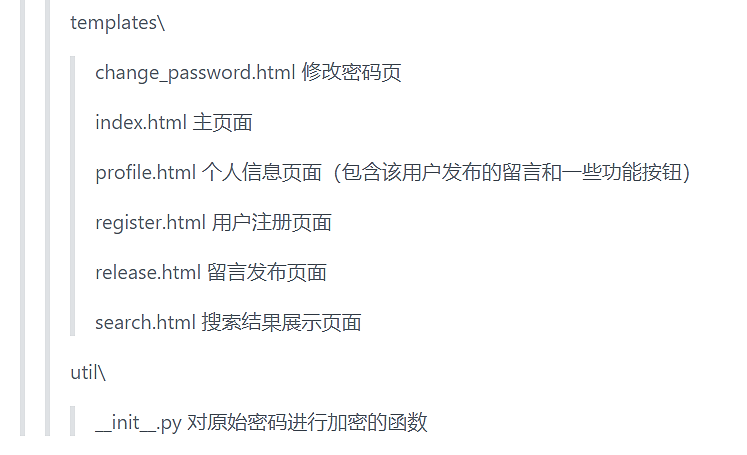
题目2：开发设计一个留言板系统；用户登陆后，能发布留言，能对自己发布的留言进行管理；能查看留言列表，分页显示，能根据留言的发布者，和留言内容，做模糊查询；基于flask框架。

1. **正文**
2. **需求分析**
   1. **功能需求**

* 用户注册（用户名用于登录和唯一标识发布留言的用户，对用户名和密码数据类型和长度进行限制）
* 用户登录
* 修改密码
* 留言发布（对留言长度和分行上限做了限制）
* 个人留言管理（修改留言、删除留言）
* 留言展示与分页（展示页内容按时间降序排列）
* 分页跳转
* 留言作者与内容的模糊查询
  1. **数据要求**
* 数据使用MySQL数据库存储
* 数据库存储密码时存储加密后的密码
* 用户名限制在1到15字，可以包含中文、英文、数字
* 密码限制在5到20字，可以包含英文、数字

1. **系统总体设计**

**2.1 系统结构与模块设计概述**



**2.2 数据库设计**

说明：

1. 数据库名：message\_board\_db
2. 表名：留言表message\_board，用户表users
3. 数据库配置文件：app\config.py
4. 数据库转储文件：message\_board\_db.sql

数据库表结构说明：

表1 留言表结构

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 字段约束 | 备注与说明 |
| id | int | 主键、非空、自动递增 | 留言序号 |
| content | varchar | 长度小于511、utf8编码 | 留言内容 |
| username | varchar | 长度小于511、utf8编码 | 留言者用户名 |
| time | datetime |  | 留言的创建或更新时间 |

表2 用户表结构

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 字段约束 | 备注 |
| user\_id | int | 主键、非空、自动递增 | 用户序号 |
| username | varchar | 长度小于255 | 用户名 |
| password | varchar | 长度小于255 | 密码，md5加密字符串 |
| register\_date | datetime |  | 用户注册日期 |

**2.3 界面设计**

界面为HTML页面，均放在bbs-demo\templates\下。界面包含：

* 主页：index.html，功能包括：分页留言展示、登录、搜索；
* 注册页：register.html，实现用户注册；
* 个人信息页：profile.html，包含该用户发布的留言和功能按钮；
* 留言发布页：release.html，填写留言内容并发布；
* 搜索结果展示页：search.html，展示搜索结果；
* 密码修改页：change\_password.html，实现密码修改功能；

1. **开发与运行环境**
   1. **开发环境**

Pycharm Professional 2020.1.2，项目根目录为bbs-demo.依赖第三方库及版本号已导出至requirements.txt。

* 1. **运行环境**

浏览器为Chrome浏览器，版本号为 83.0.4103.97。

主机为Windows 64位操作系统，基于x64处理器。8GB内存，Intel® Core™ i7-8750H CPU @ 2.20GHz

1. **系统部分具体实现**
   1. **开发技术**

服务端基于flask框架，使用flask\_sqlalchemy进行数据库操作，使用flask\_login进行用户登录；

* 1. **系统界面示意**
     1. 主页如图1所示



图1 系统主页

功能描述：

* 左下角输入页码可以进行跳转，已经做了数据验证，不合法数据会进行提示；
* 右下角是翻页按钮，当到达边界时自动设置为不可点击；
* 每页展示12条留言，在app\config.py中的POSTS\_PER\_PAGE进行配置；
* 右上角可以登录，未登录时只有一个登录按钮，点击后可以在弹出的模态框中登录。登录后可以发布留言或进入个人主页；
* 搜索框可以直接搜索无需登录。搜索后弹出新窗口展示搜索结果。对空字符串进行了判断；
  + 1. 个人主页如图2所示

图2 个人信息页

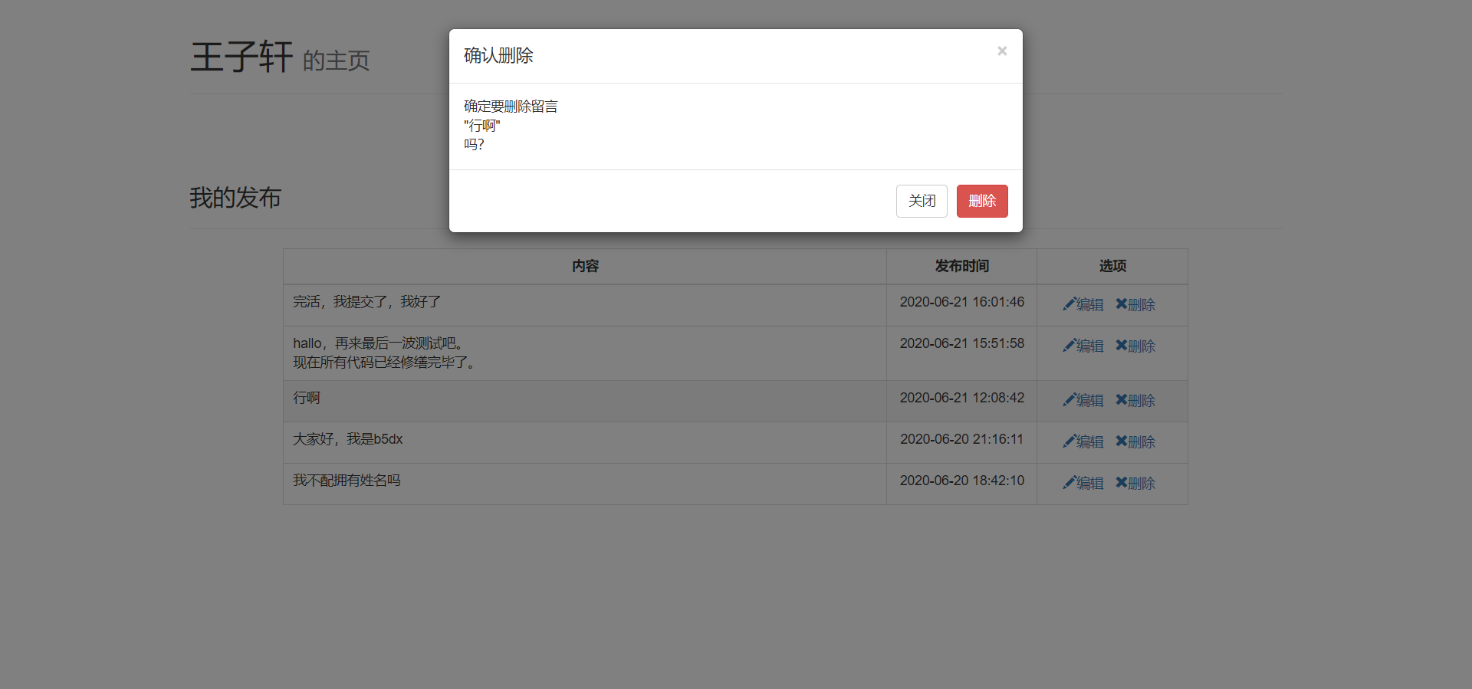
包含所有该用户发布的留言，并能对留言进行编辑修改和删除。上方是功能按键。点击删除后弹出确认删除的模态框，如图2.1所示

图2.1 确认删除

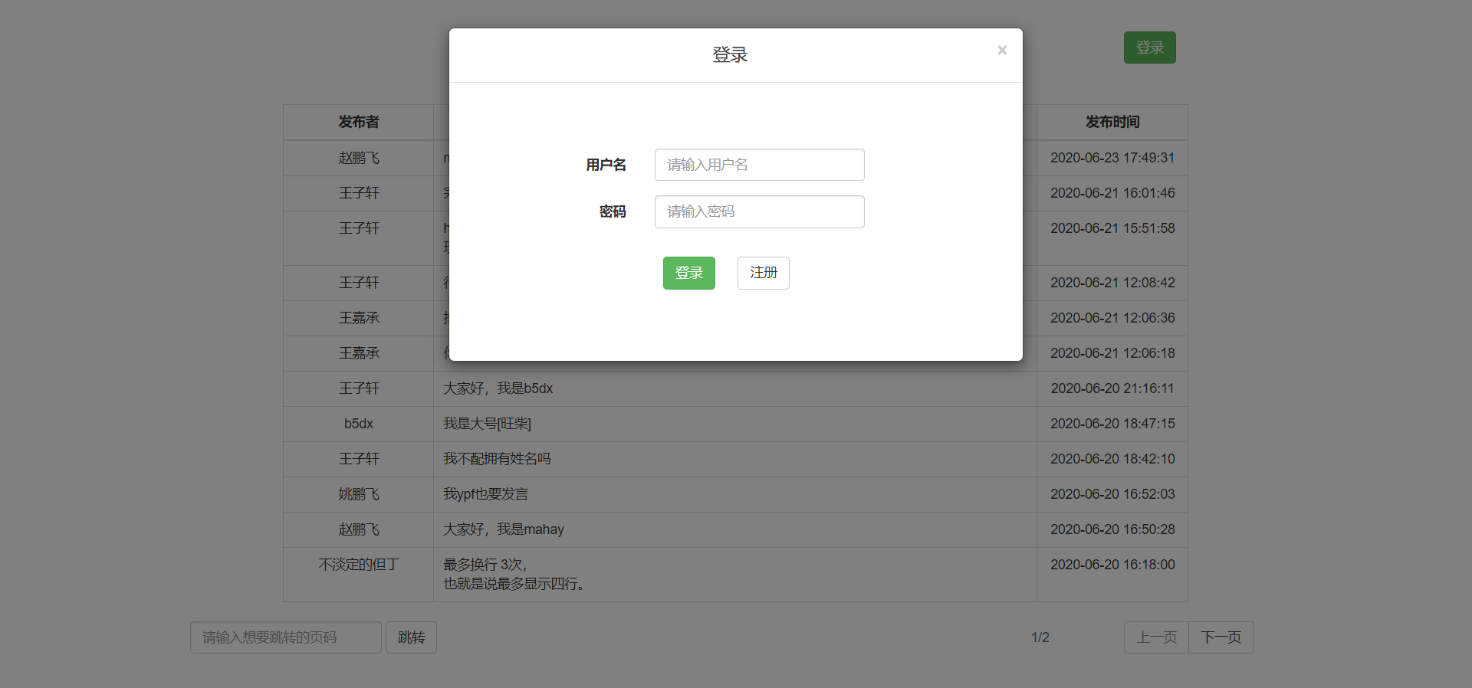
* + 1. 登录模态框如图3所示

图3 登录模态框

* + 1. 注册页如图4所示



图4 注册页

输入框中有占位符进行输入内容提示，用js进行数据类型和长度的初步验证，通过后传送至服务端和数据库交互，注册成功后回到主页，否则提示错误信息，仍然留在注册页等待再次注册。

* + 1. 留言发布页面如图5所示

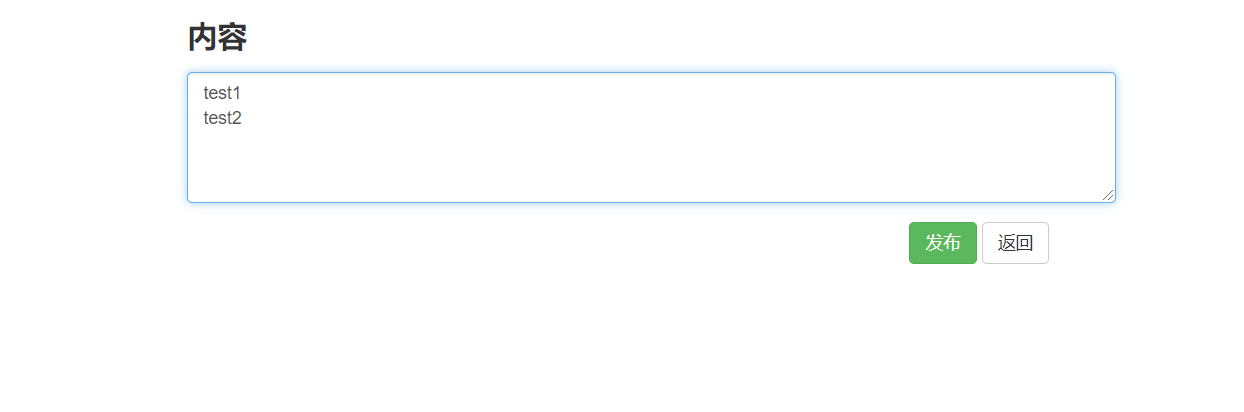


图5 留言发布页

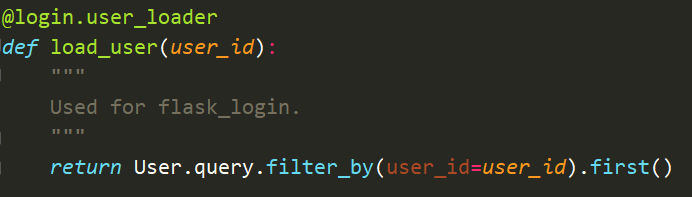
内容限制255字以内，最多手动回车分行3次。

* 1. **分页留言展示功能的详细设计与实现**

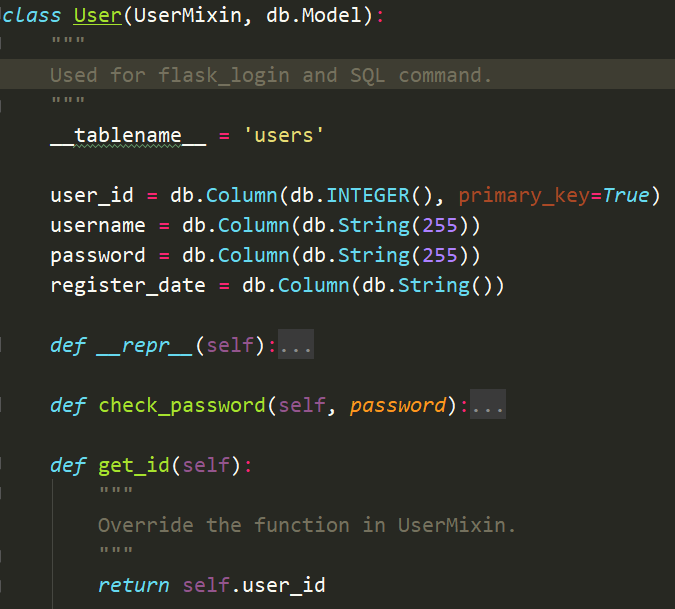


首先用request.args.get获得url中的所请求的页面的信息，然后调用sql对象（从module/\_\_init\_\_.py中import进来的SQL类的对象）的get\_message\_num()获取当前留言的总数，并计算总页面数（用于判断分页边界，控制按钮disabled属性和跳转页面的合法范围）。利用BaseQuery对象的paginate方法获取Paginate对象，将next\_url（下一页的请求链接）和pre\_url（上一页的请求链接）生成。最后使用index.html模板。传入参数message\_list消息列表（由Message对象组成的list，Message对象在module\model.py中），也即本页要展示的消息，然后传入两个url，用于点击button时触发翻页事件切换到对应页面。cur\_page和total\_page用于主页的页码展示。

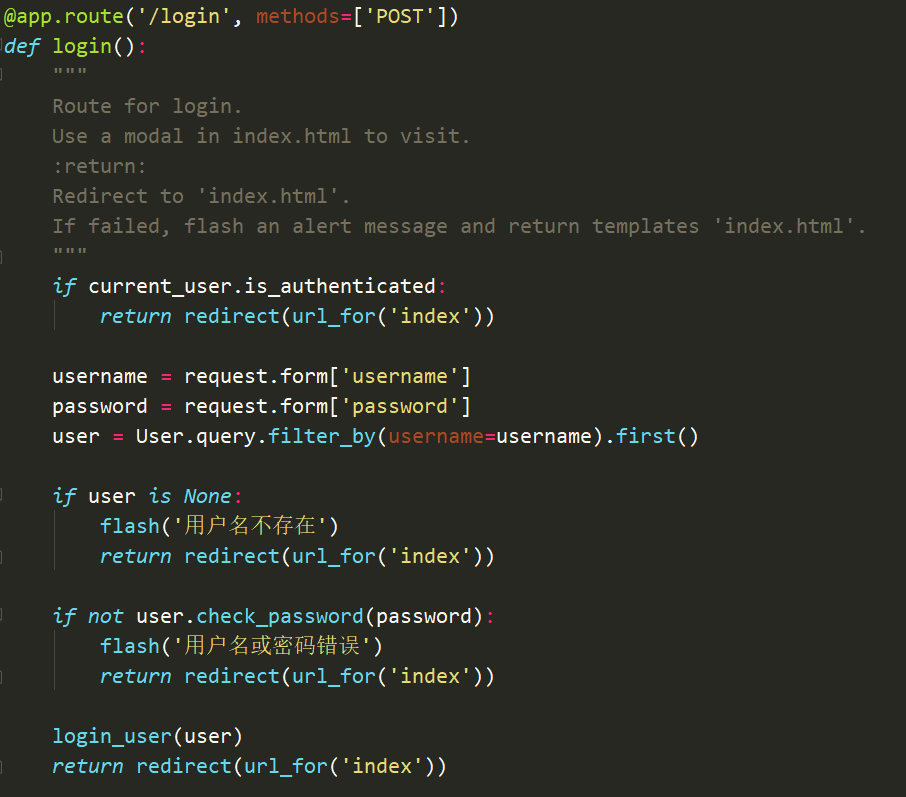
* 1. **用户登录功能的详细设计与实现**



在module\model.py中使用login.user\_loader装饰器装饰load\_user函数来让flask\_login中的login\_user(User)正确找到user。为了使用user\_id获取user，需要重载继承了UserMixin类的User类中的get\_id方法，如下图所示

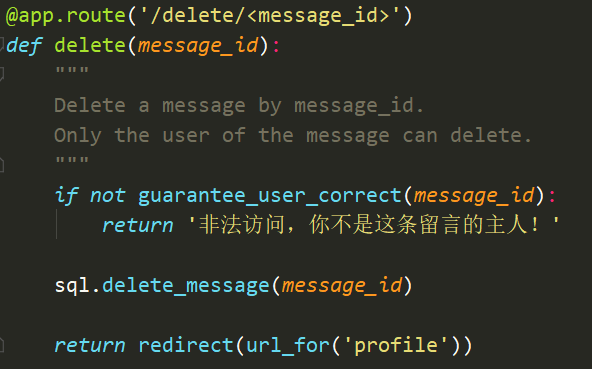


最后，在app\routes.py文件中进行服务端的登录操作，包含获取登录表单信息，验证信息，返回提示信息：



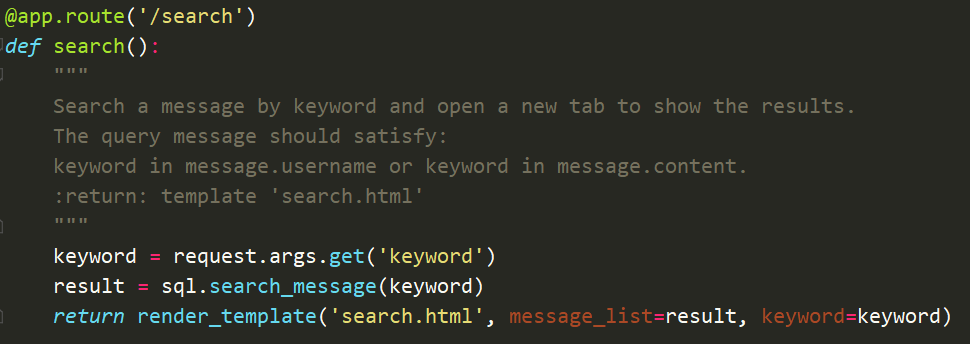
* 1. **删除指定留言功能的详细设计与实现**

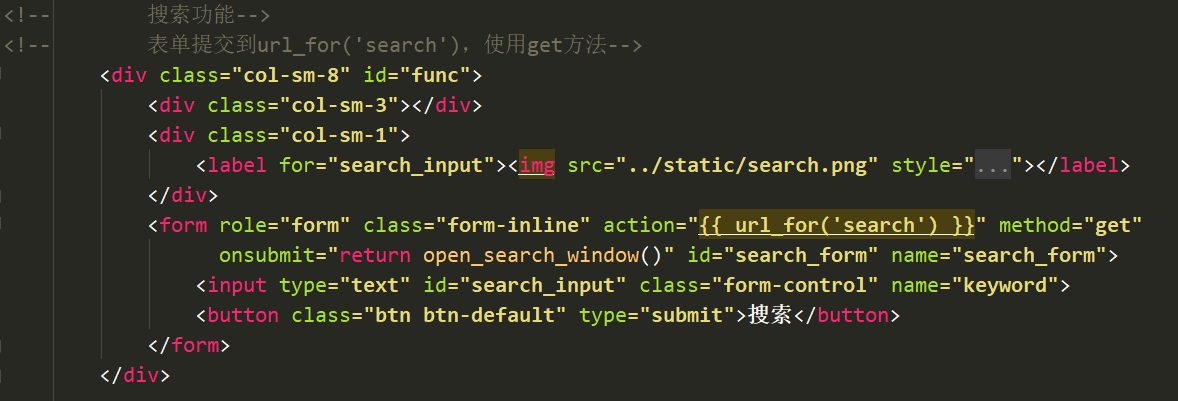
删除留言需要将message\_id提供给app\routes.py中的delete函数，通过访问delete/<message\_id>路径来将数据传送给flask。delete函数如图

访问链接前检验是否为留言的拥有者，否则为用户非法访问。确认后调用sql的delete\_message方法通过message\_id删除留言，然后跳转回个人主页。接下来的问题就是如何把message\_id从前端传入。在profile.html的删除确认模态框中，利用jinja来将对应的message传入js函数。其中，i是一个Message对象，i.id获取被点击的删除按钮对应的message的id。此处有红色下划线报错是因为Pycharm不能识别jinja，但是运行时并不会有问题，因为jinja会在用户访问到的HTML中动态的将此处替换成静态值。

当点击删除按钮确认删除后，confirm\_delete就是改变当前url为delete/<message\_id>，也就意味着调用了routes.py中的delete函数，如果正常操作删除成功后会立刻被重定向到profile页面，也就是说在用户眼中点击了确认删除按钮后仿佛刷新了一下个人主页一般，留言已经消失。

* 1. **搜索功能的详细设计与实现**

搜索结果页面如上图所示，输入内容并点击搜索后会打开新窗口展示内容，原窗口的搜索框输入内容被清空。若输入空串或者不输入会有提醒。

routes.py中的search函数进行搜索处理。直接从url中使用request.args.get方法获得搜索内容，用sql对象的search\_message来进行模糊搜索，只要发布者的用户名和发布内容中含有keyword就可以被检索到，检索结果也是按照时间降序排列的。接下来看前端index.html中对搜索的实现。

表单的action参数意味着想search函数对应的url发起GET请求，通过onsubmit属性中的open\_search\_window来判断输入内容是否合法，如果合法则提交表单然后打开用于展示搜索结果的新页面search.html。

1. **总结**
2. **总结及体会**

本次课程设计基本是从对web开发零了解的情况下进行，同时进行HTML、JavaScript、CSS的简单了解，也要了解bootstrap框架，同时扩充flask的知识。总的来说对不了解web开发的我来说难度和任务量不小。从一开始的一头雾水，到能够设计出简洁清爽的界面，对前后端的交互有一定的了解，实现简单的功能。对目前的成果比较满意，但如果有更多时间想要扩展更多功能例如留言回复，找回密码操作，个人信息展示等。

**2. 难题及解决方案**

遇到的有印象的困难和对应的解决方法如下：

* 刚开始不清楚表单的操作，不清楚如何让前后端交互。通过学习form的属性，看教学视频来理解表单提交操作。
* 学习bootstrap布局，学习JavaScript与HTML的交互。
* 不清楚如何让用户登录。看文档，看老师上课分享的教程，学习了flask\_login解决了登录问题。
* 不清楚如何实现分页。为此学习了paginate方法。
* 不清楚如何删除指定留言，不懂如何让用户点击删除后确认操作。学习bootstrap的模态框使用，想到用jinja直接动态生成HTML来传输留言id，对jinja有了更深的了解。