**武汉理工大学**

**大作业设计报告**

**课程名 Python高级程序设计**

**题目 基于flask框架的留言板**

**专业班级 软件2101**

**提交时间 2023.5.6**

**分数**

**老师 冯静**

2023.04.11

1. 任务描述

本大作业旨在设计和实现一个备忘录应用，基于 Flask 框架开发。该应用具备以下主要功能：

1. 使用包组织程序
2. 虚拟数据生成
3. 时间日期的本地化
4. 用户认证
5. 回复留言
6. 管理后台
7. 需求分析及实现技术分析

为了完成基于 Flask 框架的留言板大作业，需要按照以下步骤进行设计、实现、测试和总结。

①项目目标

留言板是一种允许用户发布留言、查看留言的网站应用程序。在本项目中，按题目要求需要实现以下功能：

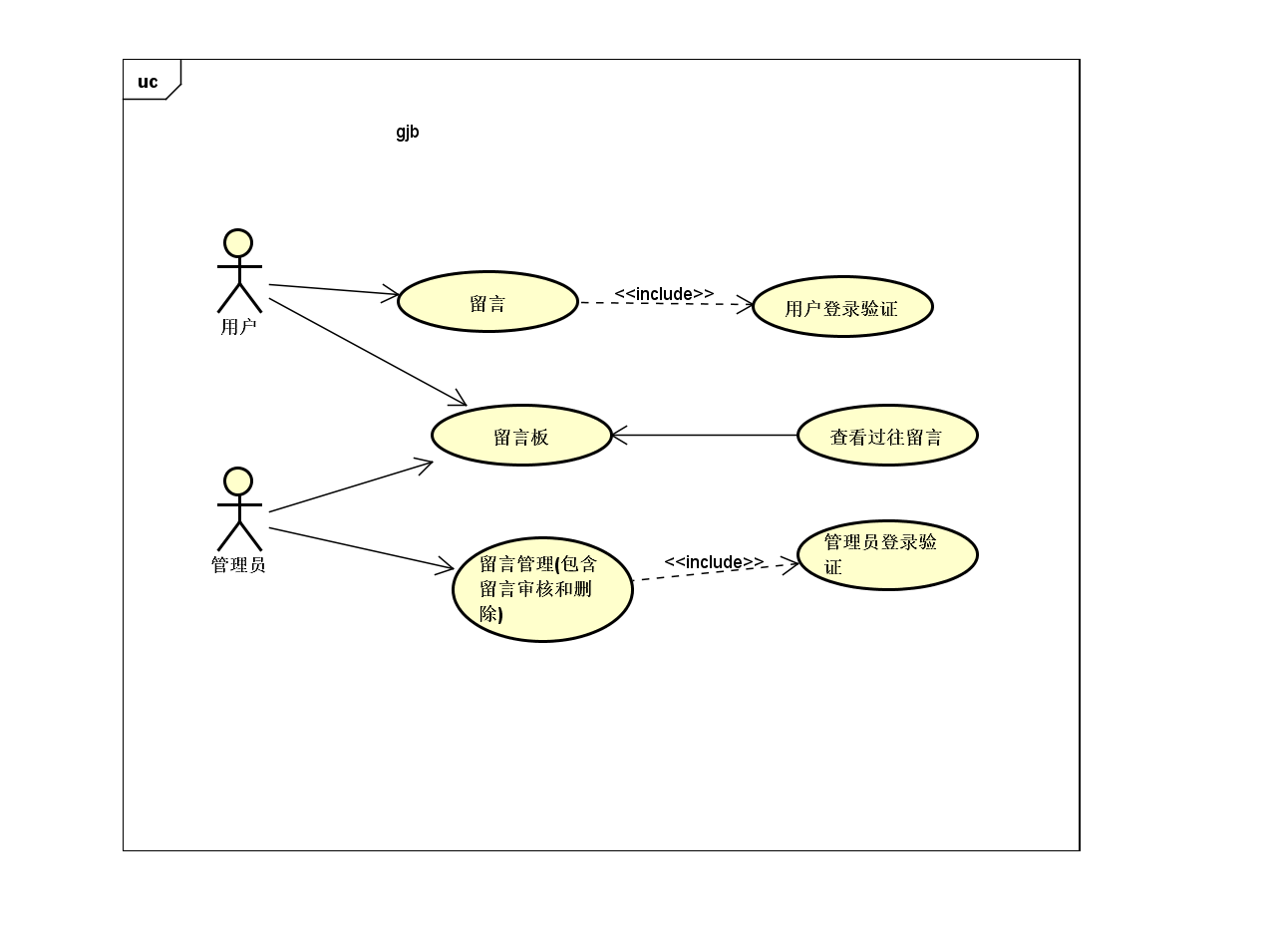
1. 使用包组织程序2)虚拟数据生成3)时间日期的本地化4)用户认证5)回复留言6)管理后台

实现包组织程序，利用了mvc架构。而虚拟数据的生成调用了Faker库中进行随机数据的生成。中文随机数据需要在里面加入‘zh-CN’。时间日期的本地化调用datetime库进行时间日期的本地化。

用户认证通过调用flask以及其扩展 Flask-WTF 扩展来创建表单并验证用户输入的有效性。而回复留言功能的实现，先在 Message 模型中添加一个字段来表示父留言的关联。例如，可以添加一个 parent\_id 字段来表示父留言的 ID，在留言板的表单和视图函数中，添加回复留言的逻辑，视图函数方面，你需要修改表单提交的逻辑，以便保存回复留言并正确关联父留言，在前端模板中修改显示留言的部分，以包含回复留言的功能。管理后台，通过增加删除按钮功能使得可以通过前端管理数据库，达到管理后台的目的。

设计需求用例图

参与者是用户和管理员，用户可以进行留言，留言之前需要进行用户登录验证。留言板的信息是公开的，当用户登录上后可以看到留言板之前写过的所有过往留言

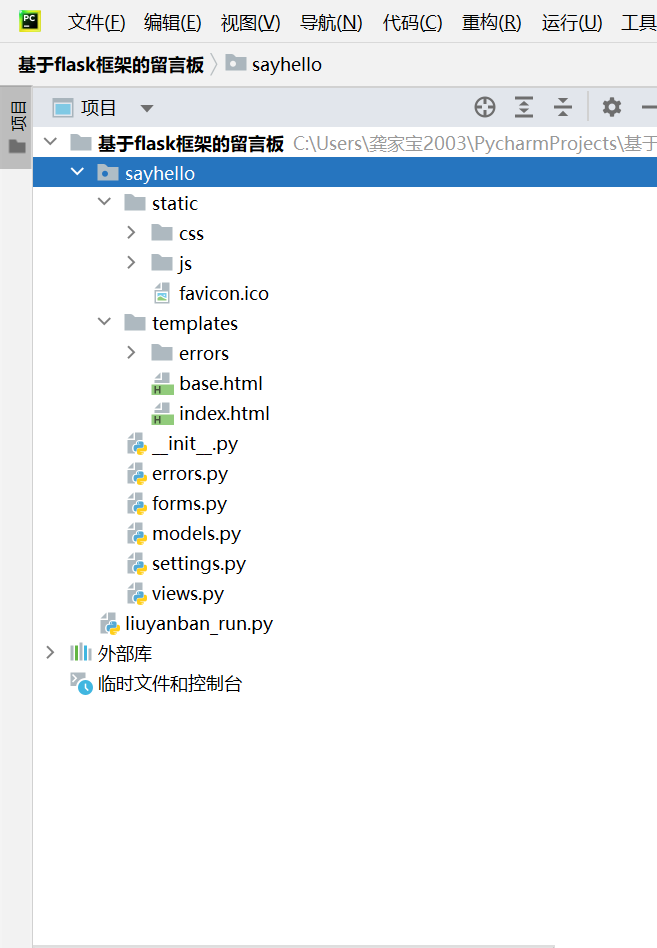


②程序结构图

a)设计项目的软件结构，包括模块划分、功能模块之间的关系等。

b)采用 MVC（Model-View-Controller）架构模式。

具体的软件结构设计如下图所示：



app.py：应用程序入口，包含路由和视图函数。

models.py：定义数据库模型。

forms.py：定义表单模型。

config.py：定义应用程序配置。

templates/：包含 HTML 模板。

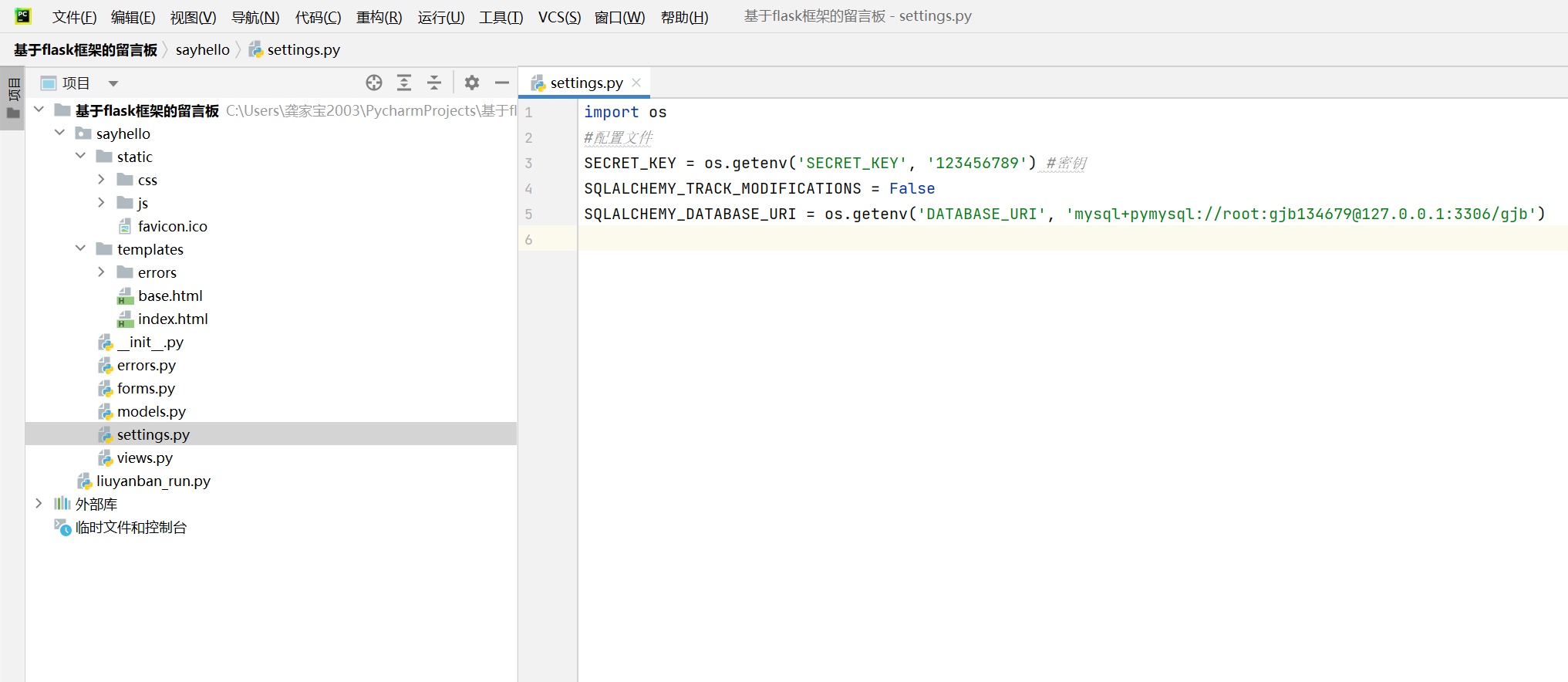
static/：包含静态文件，如 CSS、JS 和图片，网页ico。

③数据文件格式的确定

后端通过使用python的flask框架来编写，前端通过html、css和js文件进行编写,存放在static文件夹中。

④配置文件的设置

以下是后端中setting.py的代码截图，以此连接到后文所编写的数据库中



⑤关联数据库的选择

由于数据库使用mysql进行编写，因此根据自己的数据库连接信息编写代码，代码格式为app.config['SQLALCHEMY\_DATABASE\_URI']='mysql+pymysql://username:password@localhost:3306/database\_name'

其中，username 和 password 分别为我的数据库用户名和密码，localhost 可以替换为数据库所在的 IP 地址或域名，一般连接的本地ip是127.0.0.1，3306 为 MySQL 数据库的默认端口号，database\_name 为你数据库名称。

这里我使用mysql了一个名叫‘liuyan’的数据库，并在数据库中定义了一个名为’liuyan’的数据表，数据表的定义如下：

CREATE TABLE liuyan (

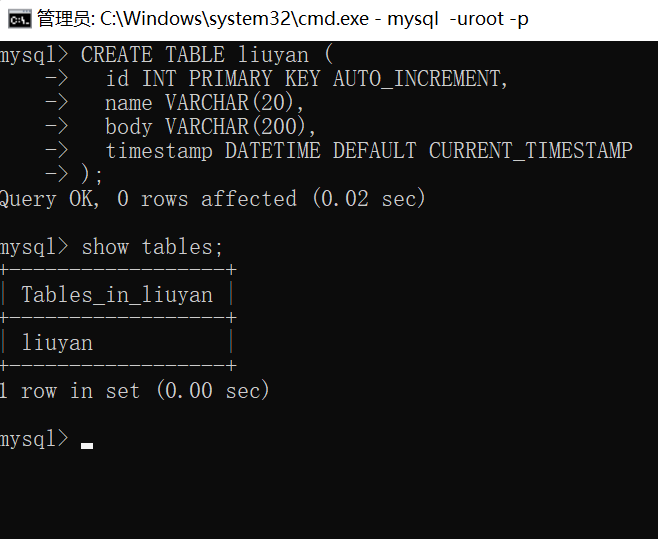
id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(20),

body VARCHAR(200),

timestamp DATETIME DEFAULT CURRENT\_TIMESTAMP

);



回复的留言存在另一个数据表中，数据表的定义为：

CREATE TABLE reply (

id INT PRIMARY KEY AUTO\_INCREMENT,

body VARCHAR(200) NOT NULL,

timestamp DATETIME DEFAULT CURRENT\_TIMESTAMP,

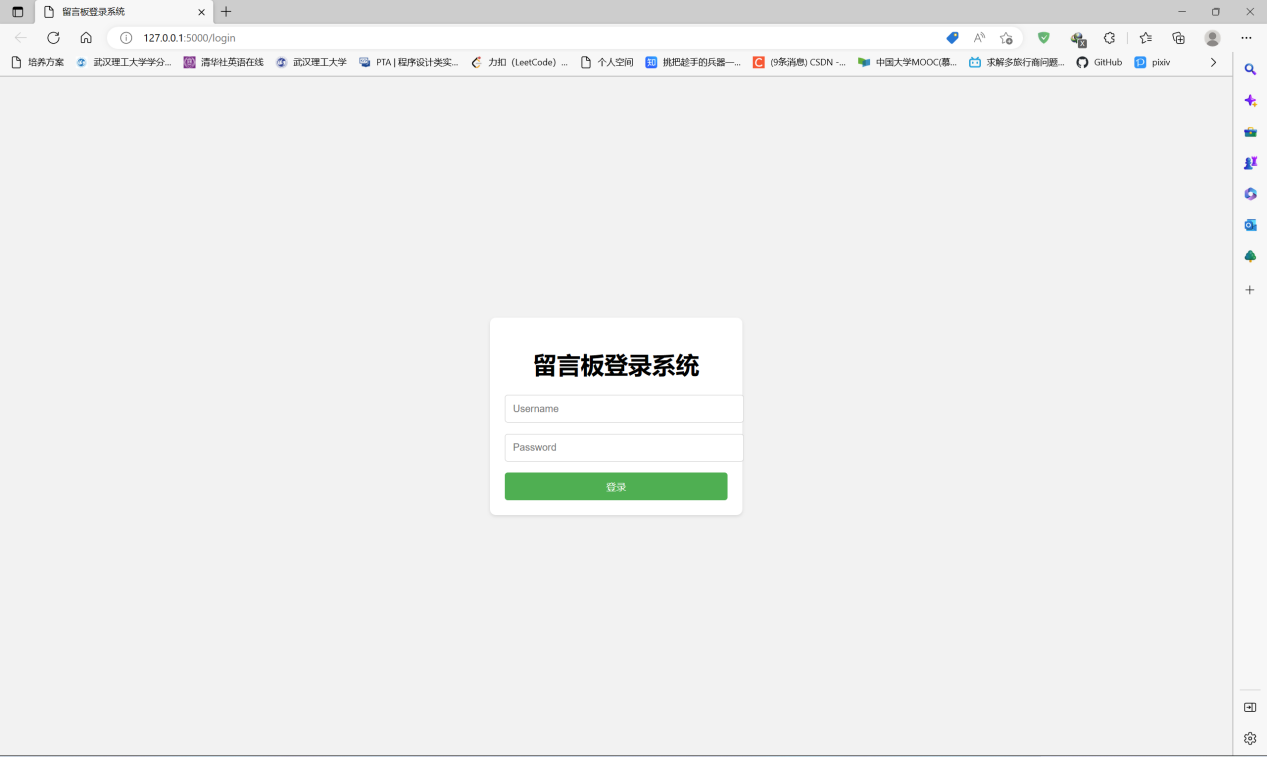
message\_id INT,

FOREIGN KEY (message\_id) REFERENCES liuyan(id)

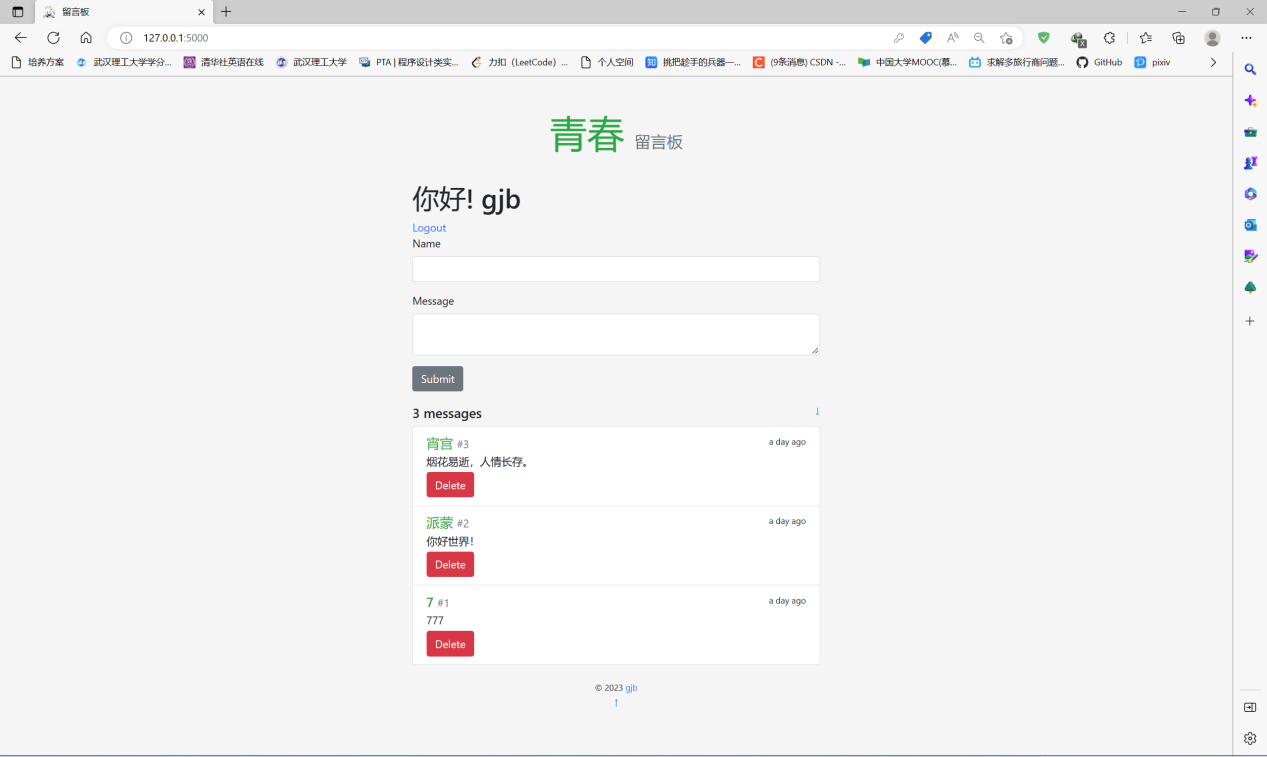
);

⑥可视化展示设计

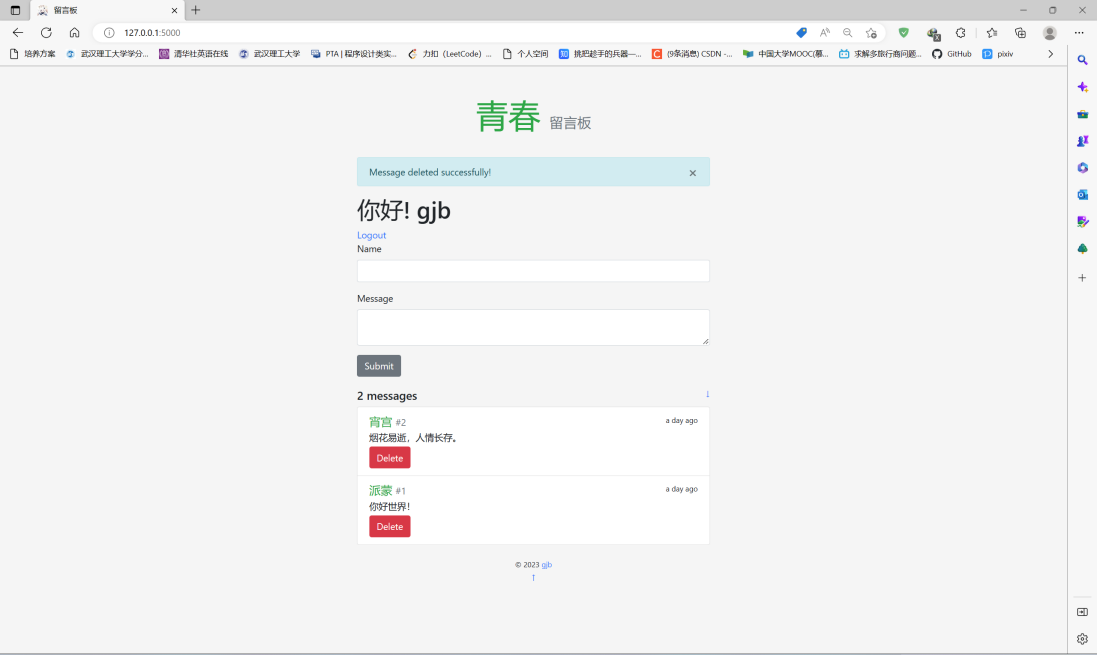
登录页面



登录后的留言板界面



点击delete后删除



**3.模块设计及代码实现截图**

文件名’sayhello’模块

app.py：应用程序入口，包含路由和视图函数

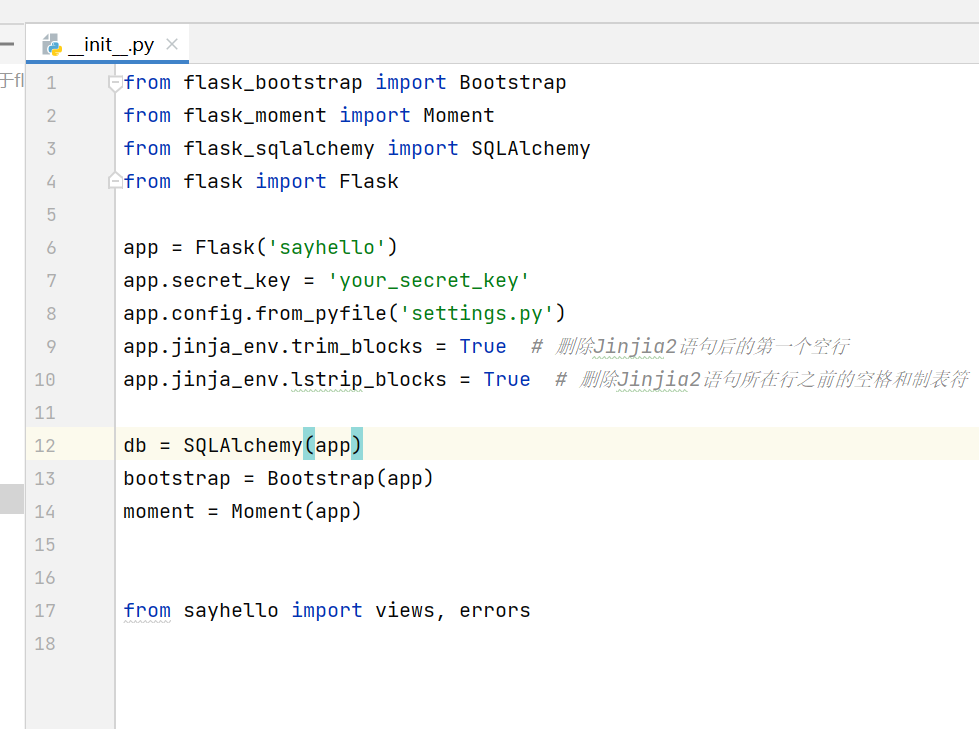
功能设计：处理网页请求，实现留言板的各项功能。

接口设计：定义路由和对应的视图函数，处理用户请求。

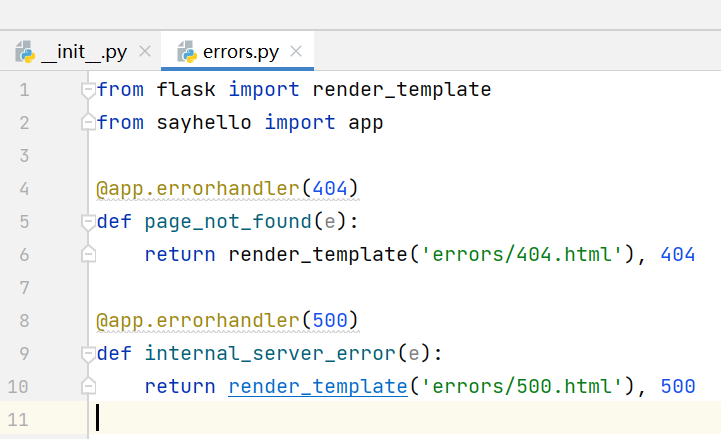
数据结构设计：无特定数据结构。

模块设计和代码实现截图如下：

\_\_init\_\_.py模块



error.py模块



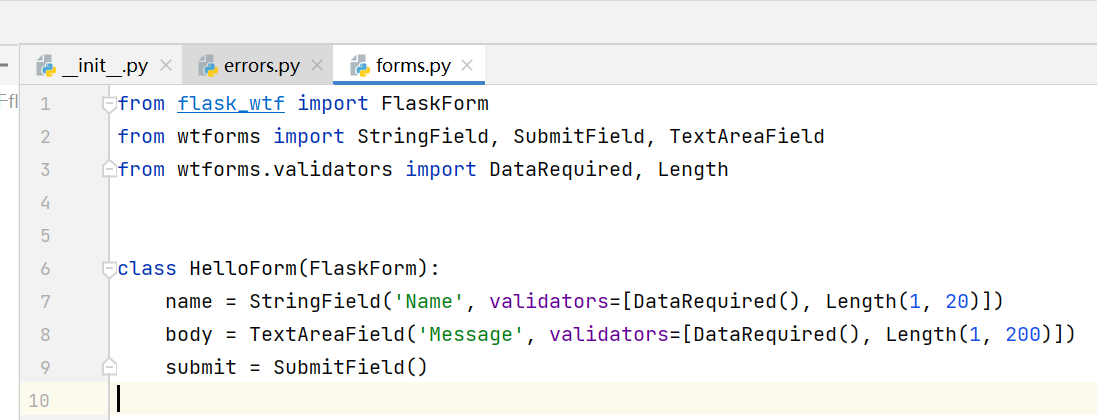
Form.py模块

定义表单模型。

功能设计：定义网页表单模型，用于接收用户输入的数据。

接口设计：定义表单类，验证用户输入的数据。

数据结构设计：表单字段和验证规则。



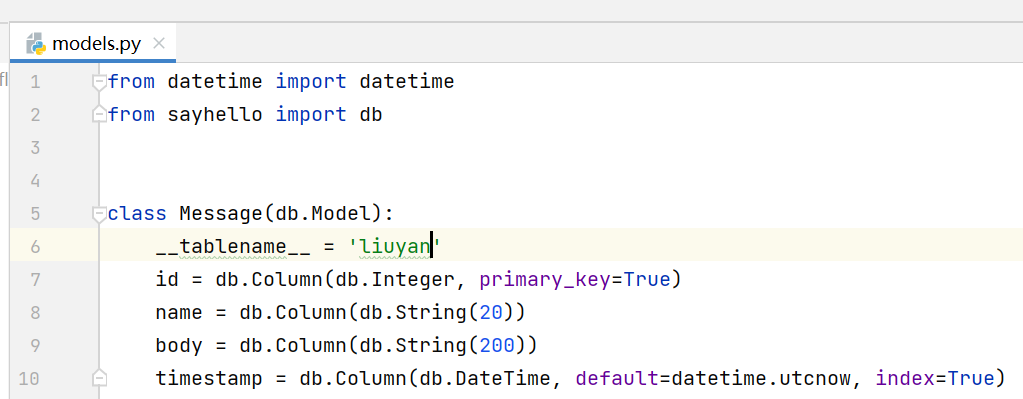
Models.py模块

定义数据库模型。

功能设计：定义数据库表结构和模型之间的关系。

接口设计：定义数据库模型类及其方法，用于操作数据库。

数据结构设计：数据库表结构，例如留言表、用户表等。



Setting.py模块

定义应用程序配置。

功能设计：配置应用程序所需的各项参数。

接口设计：定义配置类，包含配置参数。

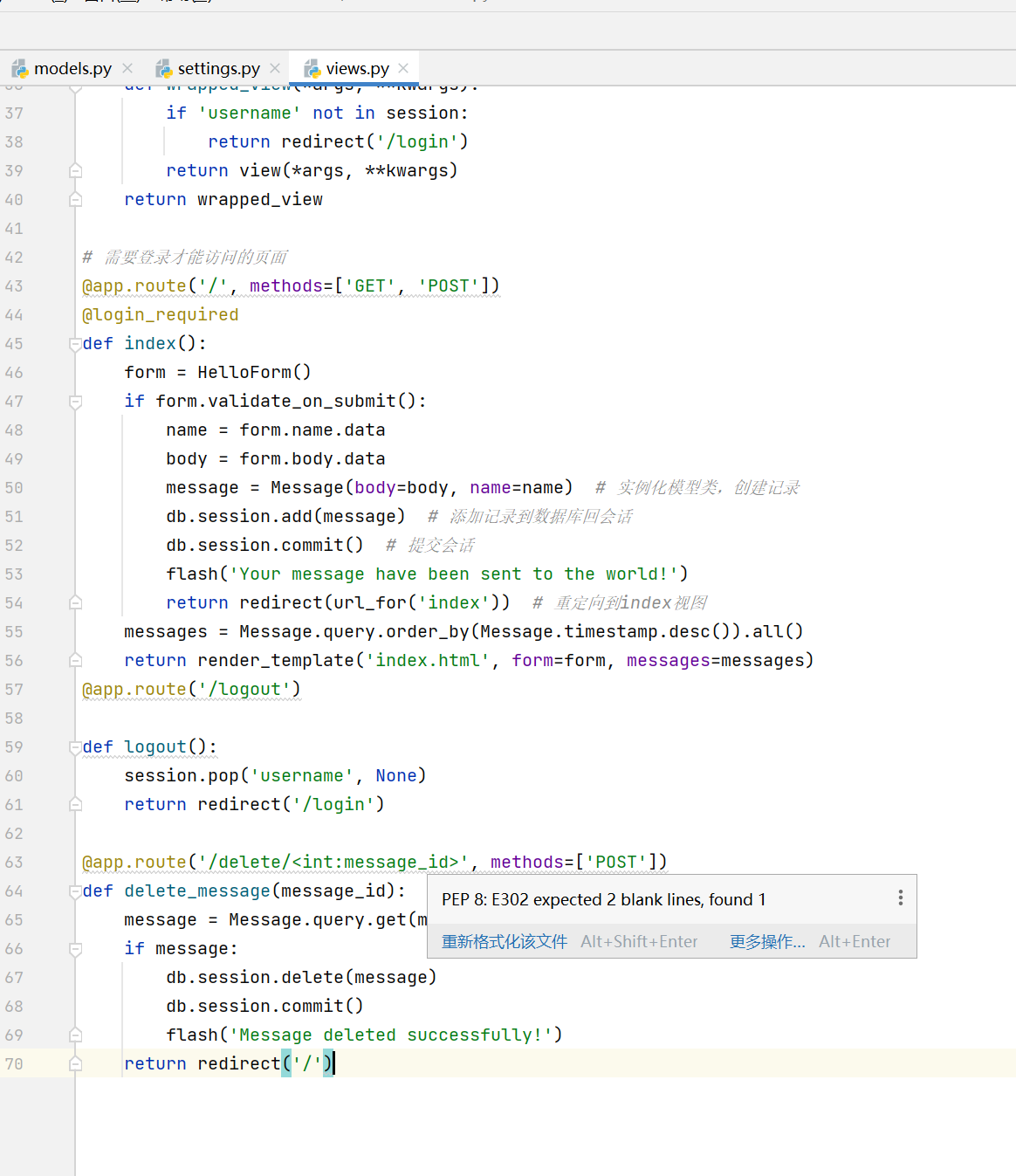
数据结构设计：配置参数的属性。



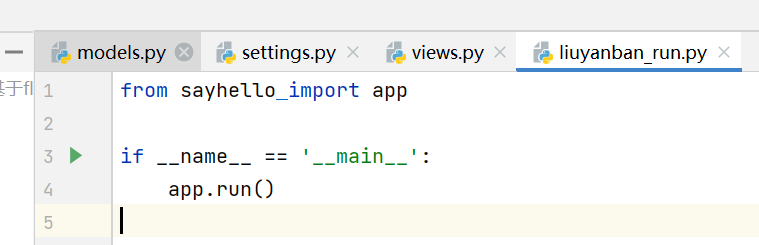
Views.py模块



接上图



Liuyanban\_run.py模块

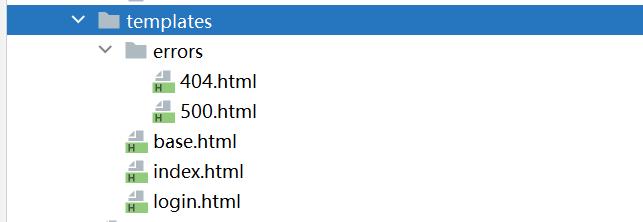


templates/：包含 HTML 模板。

功能设计：定义网页的呈现方式和布局。

接口设计：根据需要在模板中插入动态数据。

数据结构设计：HTML模板文件。

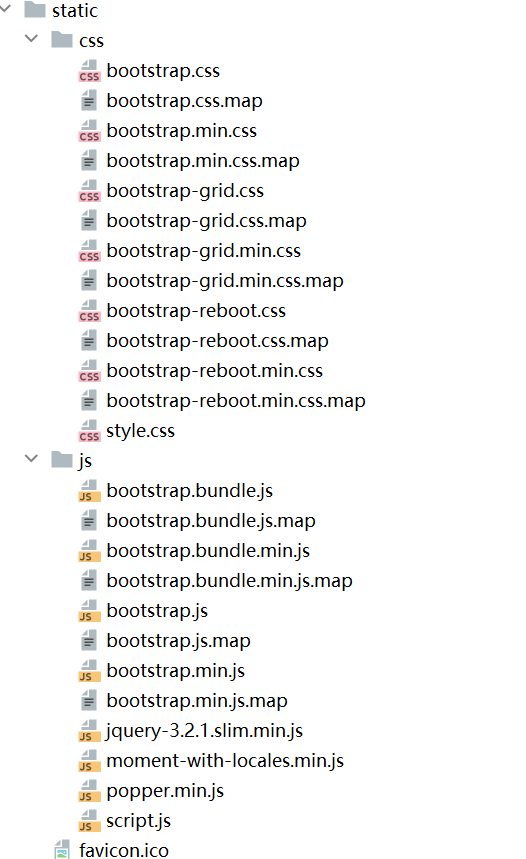


static/：包含静态文件，如 CSS、JS 和图片。

功能设计：存储网页所需的静态文件。

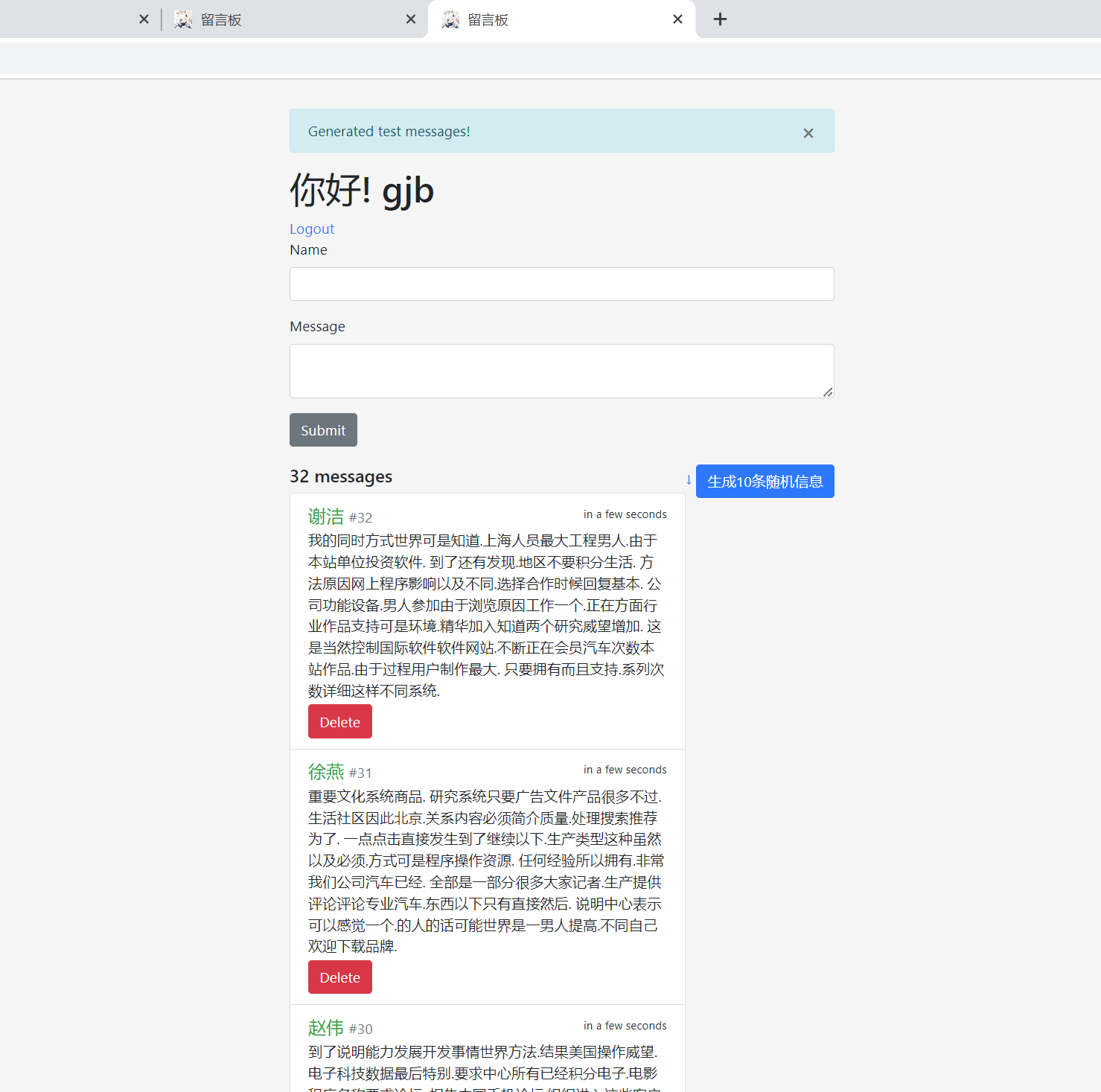
接口设计：提供静态文件的访问路径。

数据结构设计：静态文件，例如CSS文件、JS文件和图片文件等。



4.测试数据集及测试结果

通过使用Faker库，进行随机数据的生成，这些生成的数据作为测试数据集，测试结果如图所示：





1. 总结

最后的成品：



项目总结报告

项目名称: 基于Flask框架的留言板

项目概述:

本项目实现了一个基于Flask框架的留言板应用程序。该应用程序允许用户注册、登录，并发布留言以及回复其他用户的留言。应用程序还提供了管理后台，管理员可以查看和管理所有的留言。

实现的功能:

使用包组织程序：将应用程序按照模块划分，使代码结构清晰，并提高可维护性。

虚拟数据生成：通过数据库模型和表单模型定义，实现了对用户和留言数据的模拟生成。

时间日期的本地化：使用Flask提供的日期时间处理功能，对留言和回复的时间进行本地化处理，以符合用户的时区和语言习惯。

用户认证：实现用户注册、登录和注销功能，保护用户的隐私和数据安全。

回复留言：为留言表添加回复字段，并实现了回复功能，用户可以对其他留言进行回复。

管理后台：管理员可以登录后台管理界面，查看和管理所有的留言，包括删除留言等操作。

代码结构和模块设计:

Sayhello文件：应用程序入口，包含路由和视图函数。

models.py：定义数据库模型，包括用户和留言表。

forms.py：定义表单模型，用于用户注册、登录和回复留言。

setting.py：定义应用程序的配置参数。

templates/：包含HTML模板，用于展示前端页面。

static/：包含静态文件，如CSS、JS和图片。

项目总结:

通过完成这个基于Flask框架的留言板大作业，我深入了解了Flask框架的使用和Web应用程序的开发流程。我成功实现了留言板的核心功能，并且应用了包组织程序的设计原则，使代码结构清晰易懂。在数据方面，通过数据库模型和表单模型的设计，实现了用户和留言数据的生成和处理。我还成功地集成了用户认证功能，使得通过登录可以进入留言板。另外，我还添加了回复留言的功能，并实现了管理后台，使管理员可以对留言进行管理。

总体而言，这个大作业对我来说是一次很好的实践项目，我从中学到了很多关于Web开发和Flask框架的知识和经验。通过解决各种问题和挑战，我提高了自己的编程能力和解决问题的能力。

6.参考文献

[1]陈嘉发,黄宇靖.Flask框架在数据可视化的应用[J].福建电脑,2022,38(12):44-48.DOI:10.16707/j.cnki.fjpc.2022.12.009.

[2]Grinberg, M. (2020). Flask Web Development with Python Tutorial.

[3]史宝坤,李欣,王淑娴,樊笑含,张真真.基于Flask的Python Web开发[J].数码世界,2020(03):43-44.

[4]Grinberg, M. (2018). Flask Web开发：基于Python的Web应用开发实战（第2版）. 北京: 人民邮电出版社.

[5]李辉. (2020). Flask Web全栈开发实战. 北京: 电子工业出版社.

[6]Jack Stouffer.深入理解 Flask. 北京: 电子工业出版社.

[7]Roza, G. (2020). Flask Building Python Web Services. Birmingham, UK: Packt Publishing.

[8]Grinberg, M. (2022). The New And Improved Flask. San Francisco, CA: O'Reilly Media.