

# 同濟大學

TONGJI UNIVERSITY

## 《WEB 技术》

### 实验报告（大作业）

实验名称	文本博客系统
小组成员	曾崇然-2152809
学院（系）	电子与信息工程学院
专 业	计算机科学与技术
任课教师	郭玉臣
日 期	2023 年 6 月 28 日

## 目录

一．项目背景.....	1
二．需求分析.....	1
三．整体设计.....	1
四．详细设计.....	1
1. 数据库设计 .....	1
2. 功能模块设计 .....	6
五．测试数据.....	7
六．小组分工.....	13
七．心得体会.....	13
八．web 技术的认识.....	14

装  
订  
线

## 一. 项目背景

随着互联网的发展，我们和短视频等媒体的接触越来越多，而和文字的接触时间相对减少了，因此我想完成一个文本的博客系统，既能增加一些文字的交流，也作为一个记录想法的工具。

## 二. 需求分析

在日常生活中，大部分人偶尔都有过一些想记录什么的想法，作为大学生，很多人的内心都会有很多想法，想将其表达出来，此外除了这种表达的想法，还有一些交流的欲望，想别人能够看到自己的想法并对此做出评价，所以我认为这样的一个文本博客系统是能够被使用的。

## 三. 整体设计

考虑到随手利用空闲时间刷一刷博客或者临时有想法想写一点什么的情况会更多一点，我认为使用手机来使用这个系统的可能会更高，因此主要设计了手机端的样式，使用了 bootstrap5 作为样式设计的框架；

由于博客系统的功能比较多，选择一个能够进行系统性的设计的框架是有必要的，我学习并使用了 django 框架来实现后端的部分，这使得我本次实验的完成更有条理，不混乱。

另外，在本次实验中，我尝试使用了一些简单的 vue 的功能，来提升网站的动态性和交互性，但是由于对 vue 的学习不够深入，并没有能使用一些进一步的功能。

## 四. 详细设计

### 1. 数据库设计

我使用了 ORM 来进行数据库的操作，设计了用户基本信息、用户交互信息、点赞点踩评论关注收藏等交互的信息、博客信息、信箱信息等数据表，具体如下：

#### ① 用户基本信息：

```
class Users(models.Model):
    user_name = models.CharField(max_length = 20)
    user_password = models.CharField(max_length = 100)
    user_email = models.EmailField()
    favor_movie = models.CharField(max_length = 100)
    favor_novel = models.CharField(max_length=100)
    favor_song = models.CharField(max_length=100)
    admin_right = models.CharField(max_length = 4)
```

#### ② 用户交互信息：

```
class User_info(models.Model):
    user_name = models.CharField(max_length=20)
    user_pp = models.CharField(max_length=25)
    user_ideas = models.IntegerField()
    user_echo = models.IntegerField()
    user_fan = models.IntegerField()
    user_like = models.IntegerField()
    user_reply = models.IntegerField(default=0)
    user_concen = models.IntegerField()
    user_coll = models.IntegerField()
    user_generate = models.IntegerField()
    user_gender = models.CharField(max_length=20)
    user_age = models.IntegerField()
    user_nature = models.CharField(max_length=20)
    user_prefer = models.CharField(max_length=20)
```

③ 点赞点踩收藏关注等信息：

装

订

线

```

class Like(models.Model):
    user_name = models.CharField(max_length=20)
    target = models.CharField(max_length=20)
    type = models.CharField(max_length=20, default="创作")

7 usages
class Clike(models.Model):
    user_name = models.CharField(max_length=20)
    target = models.CharField(max_length=20)

10 usages
class Dislike(models.Model):
    user_name = models.CharField(max_length=20)
    target = models.CharField(max_length=20)
    type = models.CharField(max_length=20, default="创作")

6 usages
class Cdislike(models.Model):
    user_name = models.CharField(max_length=20)
    target = models.CharField(max_length=20)

9 usages
class Comment(models.Model):
    user_name = models.CharField(max_length=20)
    target = models.CharField(max_length=20)
    content = models.TextField()
    type = models.CharField(max_length=20, default="创作")
    like = models.IntegerField(default=0)
    dislike = models.IntegerField(default=0)

8 usages
class Concen(models.Model):
    user_name = models.CharField(max_length=20)
    target = models.CharField(max_length=20)

class Concen(models.Model):
    user_name = models.CharField(max_length=20)
    target = models.CharField(max_length=20)
    
```

④ 博客信息

```
class Creation(models.Model):
    user_name = models.CharField(max_length=20)
    creation_id = models.IntegerField(default=0)
    title = models.TextField(default="")
    introd_cont = models.TextField()
    content = models.TextField()
    is_punish = models.CharField(max_length=10)
    look = models.IntegerField(default=0)
    like = models.IntegerField(default=0)
    dislike = models.IntegerField(default=0)
    comment = models.IntegerField(default=0)
```

## ⑤ 信箱信息

```
class Message(models.Model):
    target = models.CharField(max_length=20, default="")
    content = models.TextField()
    is_read = models.CharField(max_length=10, default="未读")
    type = models.CharField(max_length=10, default="")
```

## 2. 功能模块设计

在本次实验中，我将整个系统分为了四个 app，一个是实现主页相关功能的 HOME，一个是实现交互功能的 INTERACT，一个是实现个人信息编辑的 PER\_INFO，最后是实现登录注册功能的 REG\_LOG，具体如下。

### ① HOME 的视图函数：

home-user：将用户的一些基本信息传递到模板用于主页的显示。

home-articles：根据不同的请求来在主页显示不同的博客在主页，如果不加特殊选择，则根据博客的点赞数和点踩数和随机因素选择一定量的博客加载到主页；如果是使用了搜索，则筛选出与搜索关键词差异在一定范围内的结果并加载到主页；如果是点击了收藏按钮，则筛选出用户收藏的博客加载到主页。

home-exit：实现退出登录的功能。

my-blocks-user：将用户的一些基本信息加载到博客编辑页面。

user-articles：将用户的博客加载到博客管理页面进行编辑或删除操作。

### ② INTERACT 的视图函数：

add-idea：实现添加博客功能的视图函数。

edit-idea：实现编辑博客功能的视图函数。

delete-idea：实现删除博客功能的视图函数。

sava-idea：实现保存博客功能的视图函数。

publish\_idea：实现发布博客功能的视图函数。

look-creation：实现浏览某篇博客功能的视图函数。

add-like：给某篇博客点赞。

add-dislike：给某篇博客点踩。

add-coll：收藏某篇博客。

add-concen：关注某个博主。

send-comment：发送评论。

get-comments：将评论送到模板用于显示。

add-likec：给某个评论点赞。

add-dislikec：给某个评论点踩。

mess-dis1: 将用户信息加载到空间信息显示页面。

mess-dis2: 将对应的信息加载到空间显示页面。

message-show1: 将用户信息加载到信箱页面。

message-show2: 显示邮箱内容。

message-delete: 删除邮箱信件。

message-check: 确认邮箱信件。

report: 举报某篇博客。

### ③ PER\_INFO 的视图函数:

info-ini: 将用户当前的基本信息加载到用户信息管理页面。

edit-name: 编辑用户姓名。

edit-gender: 编辑用户的性别。

edit-age: 编辑用户的年龄。

edit-nature: 编辑用户的性格。

edit-pwd: 编辑用户的密码。

edit-email: 编辑用户的邮箱。

edit-pp: 更改用户的头像。

### ④ REG\_LOG 的视图函数:

reg: 加载注册页面。

reg-in: 实现注册功能。

log: 加载登录页面。

log-in: 实现登录功能。

## 五. 测试数据

### 1. 登录注册:

注册账户

昵称

密码(8-16数字加字母)

重复密码

电子邮件地址

一部喜爱的电影

一本喜爱的书

一首喜爱的歌

注册

已有账户, 去登录

登录账户

昵称

密码(8-16数字加字母)

登录

没有账户, 去注册

忘记密码? 点我找回

### 2. 主页:

① 显示:



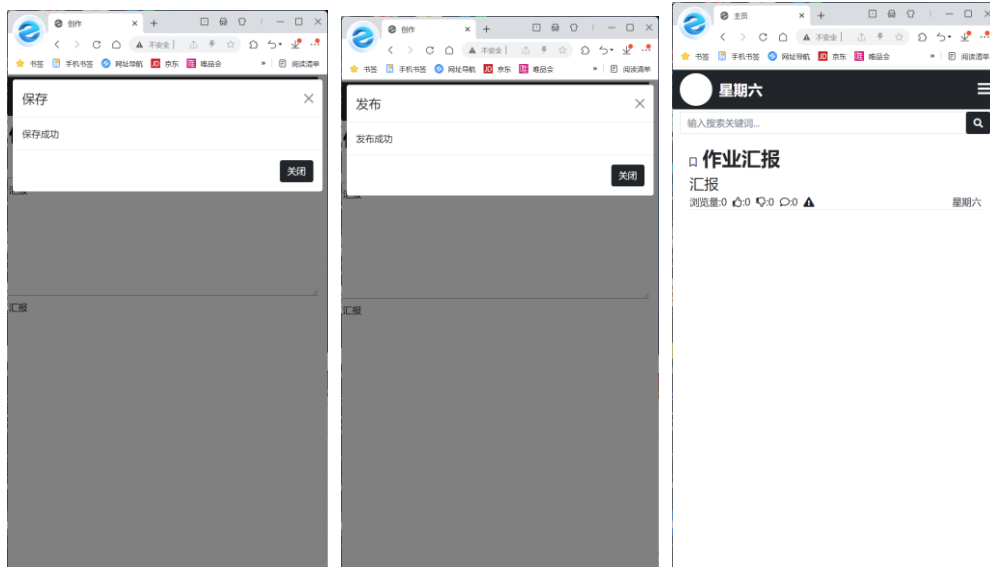
② 搜索:



③ 添加博客:

保存、发布、搜索的显示结果如下:





- ④ 进行交互：  
主页面交互，点赞、点踩、收藏：

## 计算机组成原理

这是一门由同济大学好老师开展的一门计算机专业基础课程

浏览量:15 4 1 3 1

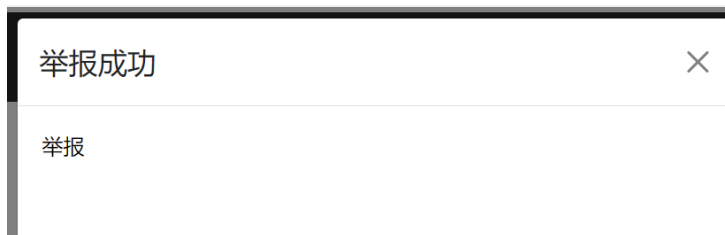
博客内交互，点赞、点踩、收藏、关注：



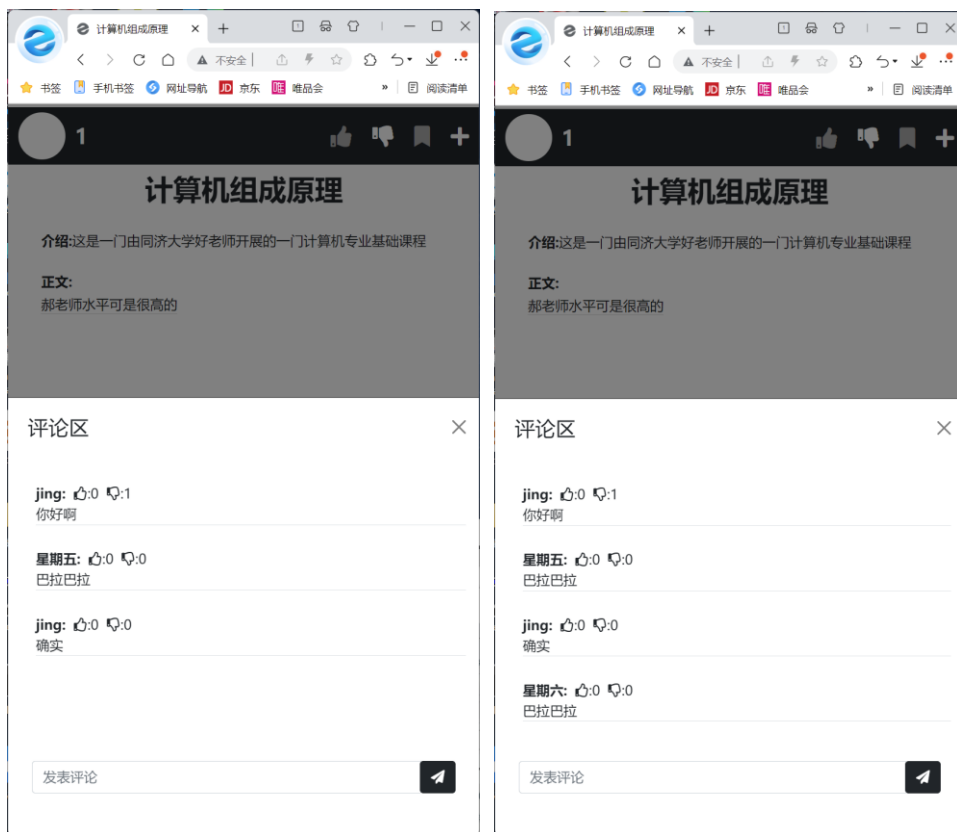
## 计算机组成原理

介绍：这是一门由同济大学好老师开展的一门计算机专业基础课程

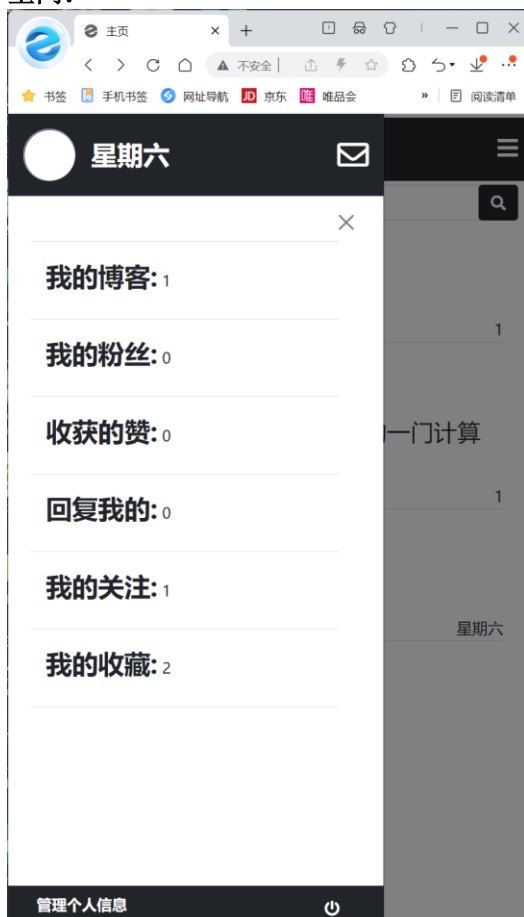
举报：



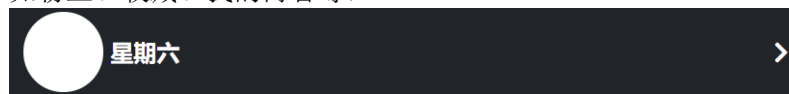
评论区：



⑤ 空间:



如粉丝、收藏、我的博客等：



Lu



输入搜索关键词...



■ 发电1号

1号



■ 作业汇报

汇报

浏览量:0 点赞:0 评论:0 警告



## ⑥ 个人信息修改：

## ⑦ 信箱:



星期六 赞了你的博客 《作业汇报》

未读

删除

## ⑧ 管理:

在管理员账号的信箱可以看到举报信息，管理员账号为：昵称：jing，密码：cr20032698，收到举报信息后，管理员可登录 django 自带的后台管理，我将用户和博客信息注册到了管理界面，管理员可以在后台进行修改，django 的账号和密码为：admin，cr2152809。



Action: -----

0 of 9 selected

☐ ID

☐ 76

☐ 75

☐ 71

Action: -----

0 of 6 selected

☐ USER NAME

☐ 星期六

☐ 星期五

☐ 没有问题

☐ Lu

☐ 1

☐ jing

## 六. 小组分工

本小组仅 2152809-曾崇然一人。

## 七. 心得体会

### 1. 收获:

在这次作业完成过程中,我主要完成了网页样式设计,用户交互设计,后台逻辑处理等工作,完成的重点放在了博客相关的流程的用户之间的交流上。

在完成本次作业的过程中,我学习了并使用了 python 的 django 框架,并使用其作为后端语言实现动态部分,前端使用了 bootstrap5 框架来设计页面的样式,并尝试使用了 vue3 来提升页面的交互性和动态效果。从整体上讲,我这次依照了 MVT 的开发模式,依照模板,视图,模型三个部分来进行完成,这使得我的开发相对比较有条理,在较多功能要实现的情况下依然不会太过混乱。在调试的过程中,我是打开了电脑 8000 端口的防火墙,使局域网能够访问,在完成之后使用了 ngrok 进行了内网穿透,使其在我服务器打开时公网也能访问,这个过程让我了解了一些有关安全问题的一些知识,比如跨域,csrftoken、防注入等。

### 2. 反思:

在本次实验的完成中存在很多的不足之处,很多地方都有很大的进步空间,第一个学习过程,这次作业用到的 bootstrap5, vue3, django 等工具的学习是不太到位的,只是匆忙的学习了一下就开始上手写作业,很多地方一知半解,这给调试带来了很大的困难。

其次样式设计,尽管花了很多的时间在设计样式上,但是效果仍然不突出,在 css 的熟练使用上还有很大问题。

在用户友好方面,该系统也存在很大的问题,有些地方需要手动刷新,一些提示不够显眼,一些排序也不够合理等等等等。另外对于网站的健壮性我是不敢确信的,由于测试的太少,我不太确定什么时候就会出现 bug,这也是之后需要改进的。

在数据库设计和操作方面也存在一些不合理的地方,有些操作过于繁琐和低效,在应对

较大的数据量时可能出现问题，这也是后续需要改进的部分。

## 八. web 技术的国内外认识

**前端开发：**前端开发是构建用户界面的关键领域。国内外都广泛使用 HTML、CSS 和 JavaScript 来开发交互式网页。国外的前端技术社区比较活跃，有很多流行的框架和库，如 React、Angular 和 Vue.js。国内也有类似的框架和库，如 Vue.js、React 和 Ant Design 等，同时还有一些国内独有的前端技术生态。

**后端开发：**后端开发涉及服务器端的逻辑和数据处理。在国外，常见的后端开发语言包括 Java、Python、Ruby 和 Node.js 等。主流的后端框架有 Spring（Java）、Django（Python）和 Ruby on Rails（Ruby）等。国内也有自己的后端开发技术栈，如 Java、Python、Node.js 等，同时还有一些国内特色的后端框架和工具。

**数据库：**数据库是存储和管理数据的关键组件。在国外，广泛使用的关系型数据库包括 MySQL、PostgreSQL 和 Oracle 等，而 NoSQL 数据库如 MongoDB 和 Redis 也越来越受欢迎。在国内，MySQL 也是常用的关系型数据库，另外还有一些国内自主研发的数据库产品，如华为的 OceanBase 和阿里巴巴的 MaxCompute 等。

**安全和性能优化：**在 Web 开发中，安全性和性能优化是重要的关注点。国内外都有一些通用的最佳实践，如使用 HTTPS 加密通信、防止跨站脚本攻击（XSS）、防止跨站请求伪造（CSRF）等。此外，性能优化方面，国内外都有一些工具和技术，如使用 CDN 加速、优化图片和资源加载、使用缓存等。

装

订

线