**实验八：身份认证及权限管理实验报告进阶**

姓名：江继军 学号：21373222 班级：213921

实验日期：2024年6月12日 实验地点：一号馆406 成绩：

1. **实验目的**

1. 进一步理解身份认证中的用户—角色映射关系，学会多层级角色的编程。

2. 理解多因素身份认证方法对平台认证的保障，学会除密码验证外至少一种验证方式。

**二、实验内容**

**进阶内容，以下三项自选一项完成即可**

1. 多层级角色设计与实现

基于实验7，为系统的某一类用户细化二级角色，例如教师角色划分为主讲教师和助教，二者权限有所区别。（此处如有创意的设计或者优秀代码，酌情+1~5分）

2. 设置登录验证码

1. 在实验7登录页面，添加“验证码”功能，用于区分登录者是人或是机器。
2. 实现在登录过程中输入、检查验证码字段功能。
3. 验证可点击“看不清楚，换一张”刷新。（优秀代码酌情+2~6分）

3. 其他登录验证方式

除课上讲过的密码和邮箱验证方式外，加入一种因素的身份验证方法，例如滑动验证条，图片转向、图片识别等。（优秀代码酌情+4~10分）

**三、实验环境**

分项列出主要实验环境（主机/操作系统、Python开发环境、Python版本、第三方库版本等）：

1. 主机/操作系统：Win11
2. Python开发环境：PyCharm Community Edition 2023.3.4
3. Python版本：3.9
4. flask版本：3.0.3
5. flask-login版本：0.6.3
6. flask-sqlalchemy版本：3.1.1
7. flask-wtf版本：1.2.1
8. captcha版本：0.5.0

**四、实验步骤**

如自选登录验证码，实验步骤举例如下：写出实现用户登录、检查验证码等步骤（采用1.2.3.4的序号方式列出详细步骤；写明所编写的代码；贴出效果示意图，例如用户登录成功/失败、检查验证码成功/失败结果截图等）。

1. 用户密码登录功能：（上次实验内容，此处写出即可）

如图所示，在用户名栏和密码栏输入正确匹配的用户名和密码，点击登录，即可登录到个人主页

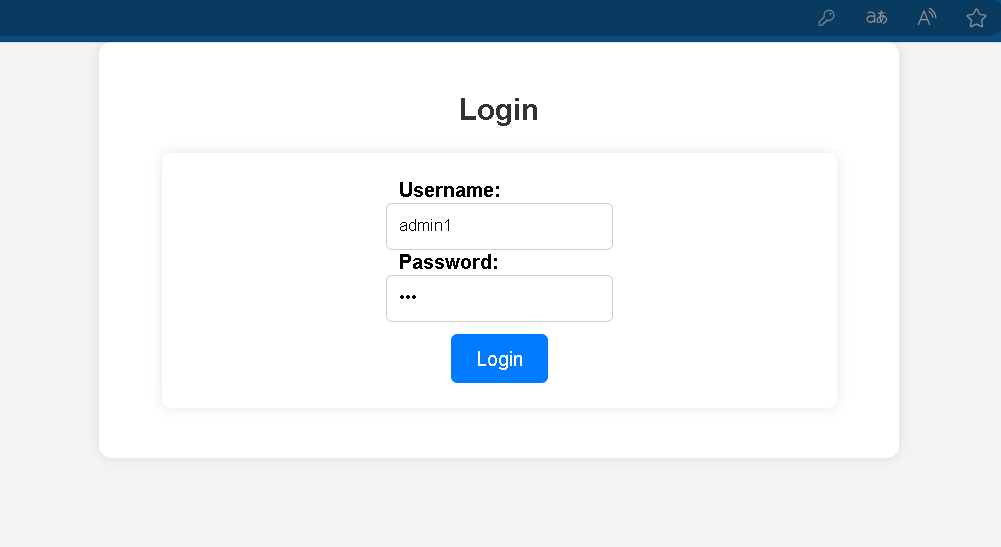


图 1 登录页面

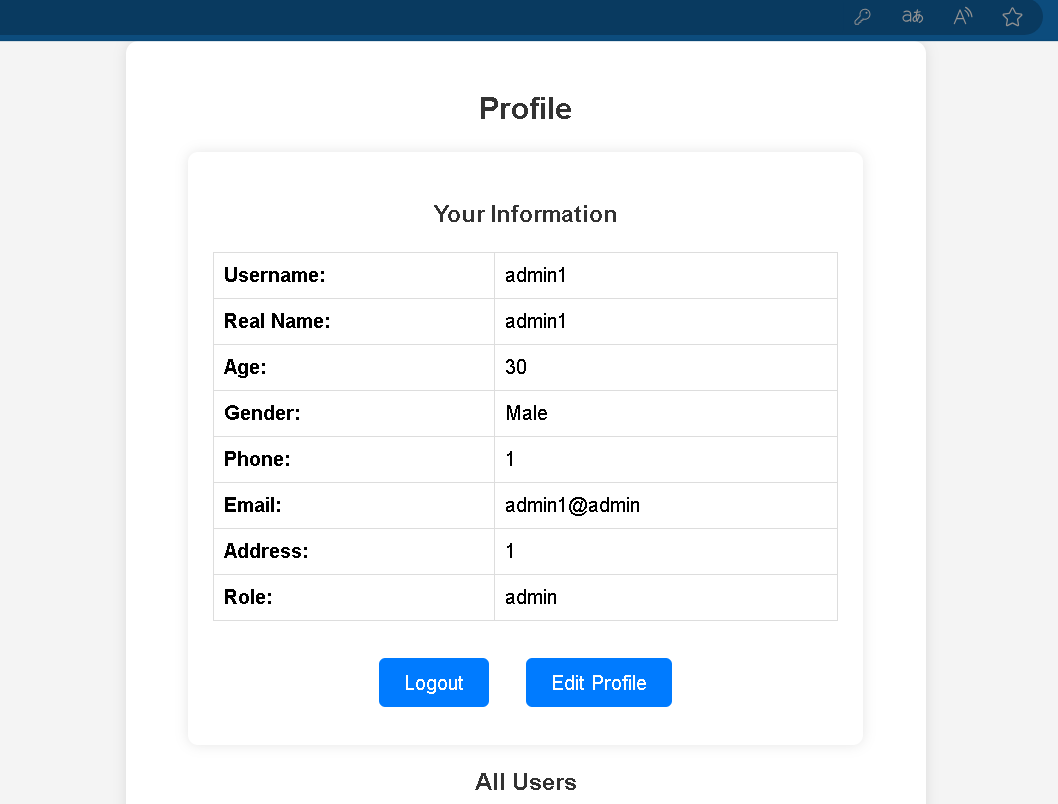


图 2 个人主页

1. 设置验证码：

加入验证码图片库，实现在用户登录时使用验证码校验身份的功能。（写明所编写的代码，给出验证码填写正确和错误的代码）。

在原登录函数基础上增加验证码校验身份功能，增加代码已用红框圈出，忽视验证码的大小写，如果验证码正确则检查用户名密码是否正确匹配，否则提示验证码错误



图 3 增加验证码代码

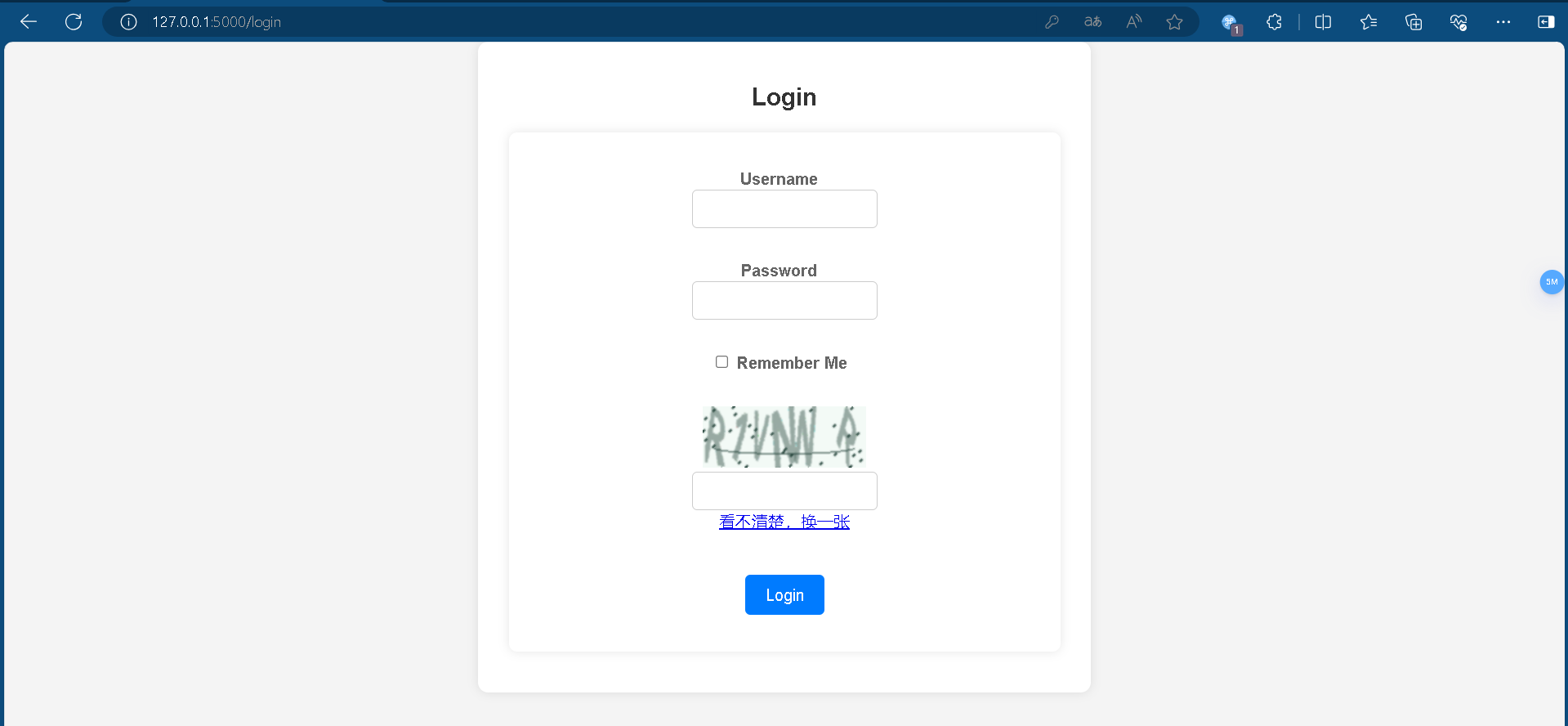


图 4 增加验证码登录页面

1. 采用验证码校验

当用户验证码，系统所增加的校验功能，填写错误时重新生成验证码并提示“验证码错误”（写明多因素登录校验的代码，以及校验成功或失败的功能截图）

规定验证码长度为6个字符，在未输入到6字符时提示，且限制输入最大长度为6字符

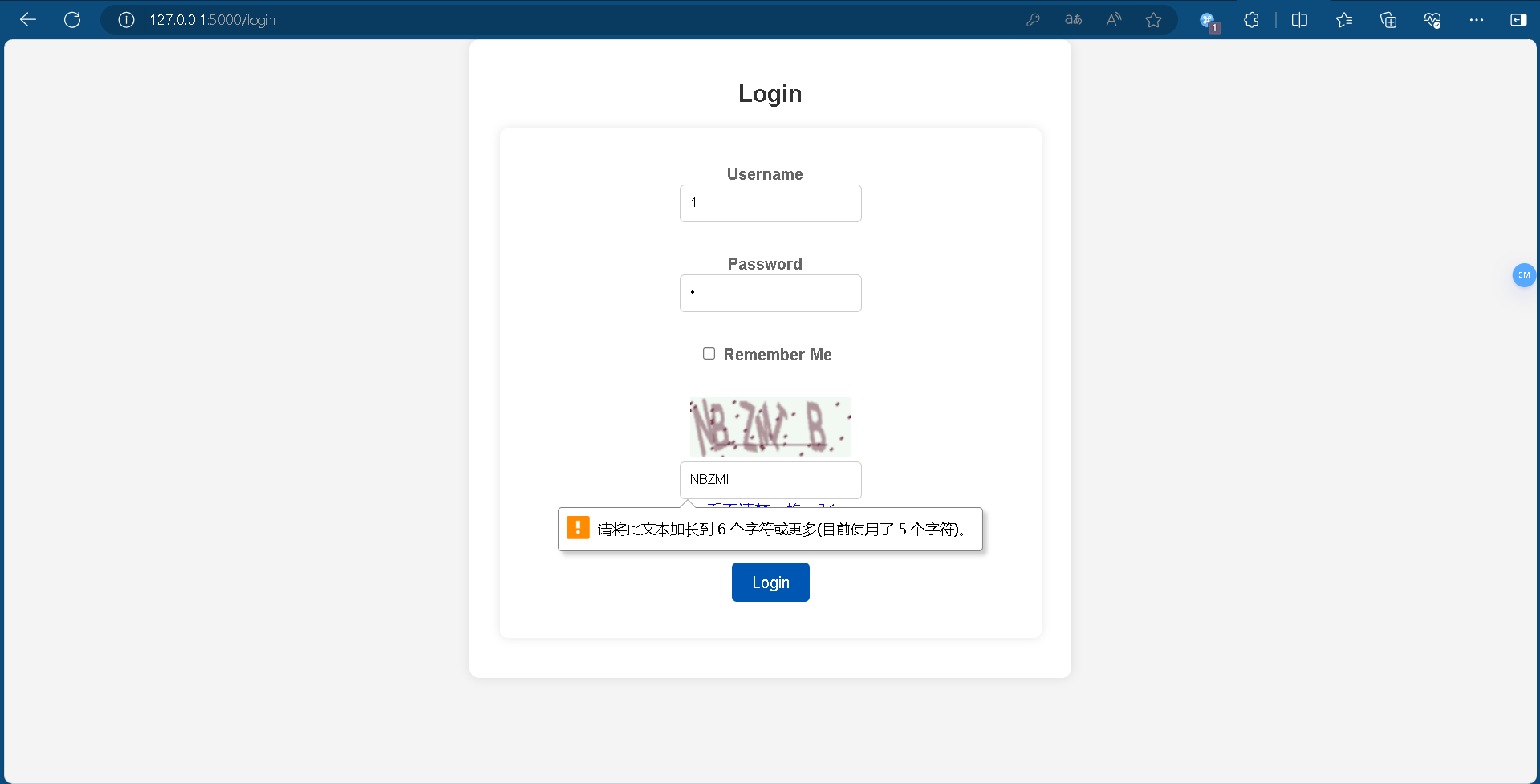


图 5 未达到要求输入长度

如果验证码输入错误，会提示验证码错误，并且重新生成验证码

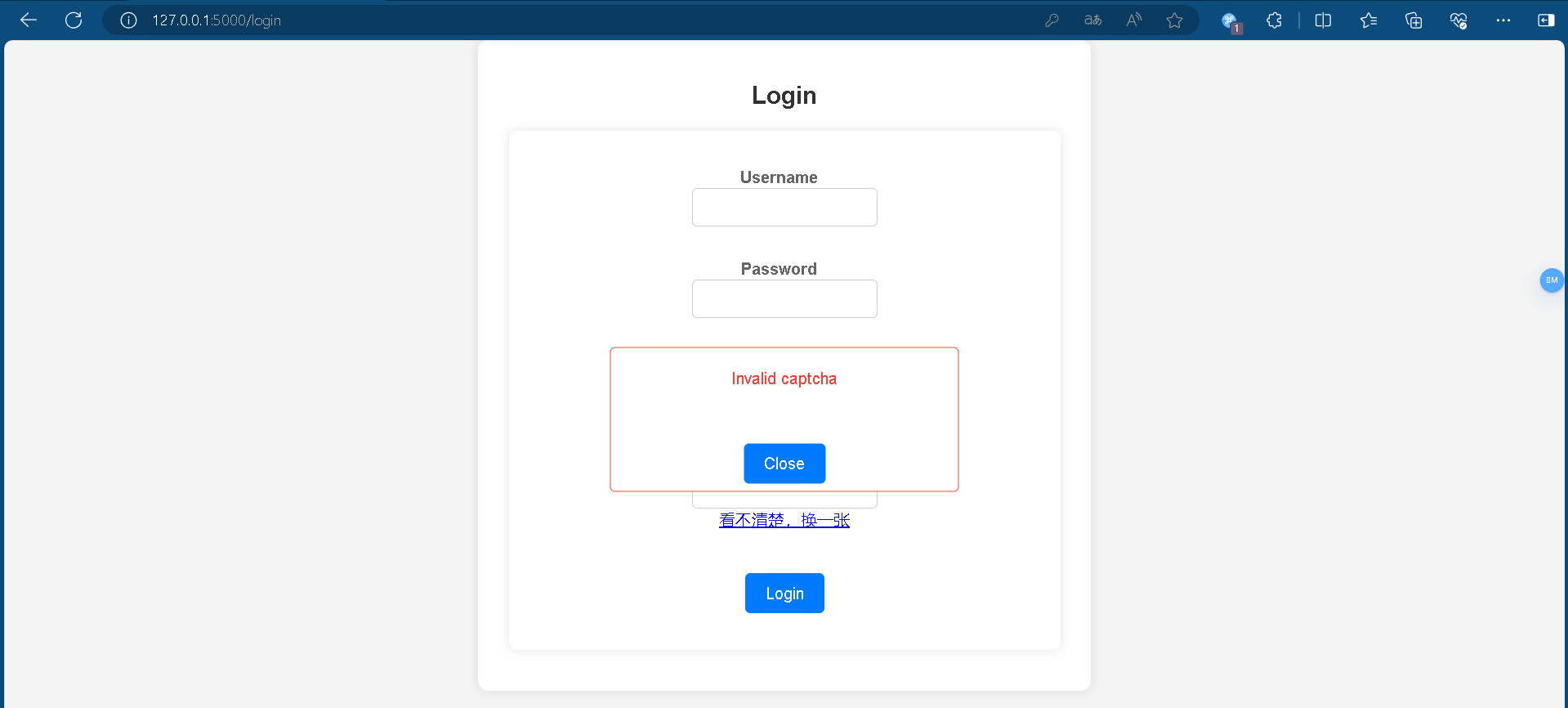


图 6 提示验证码错误

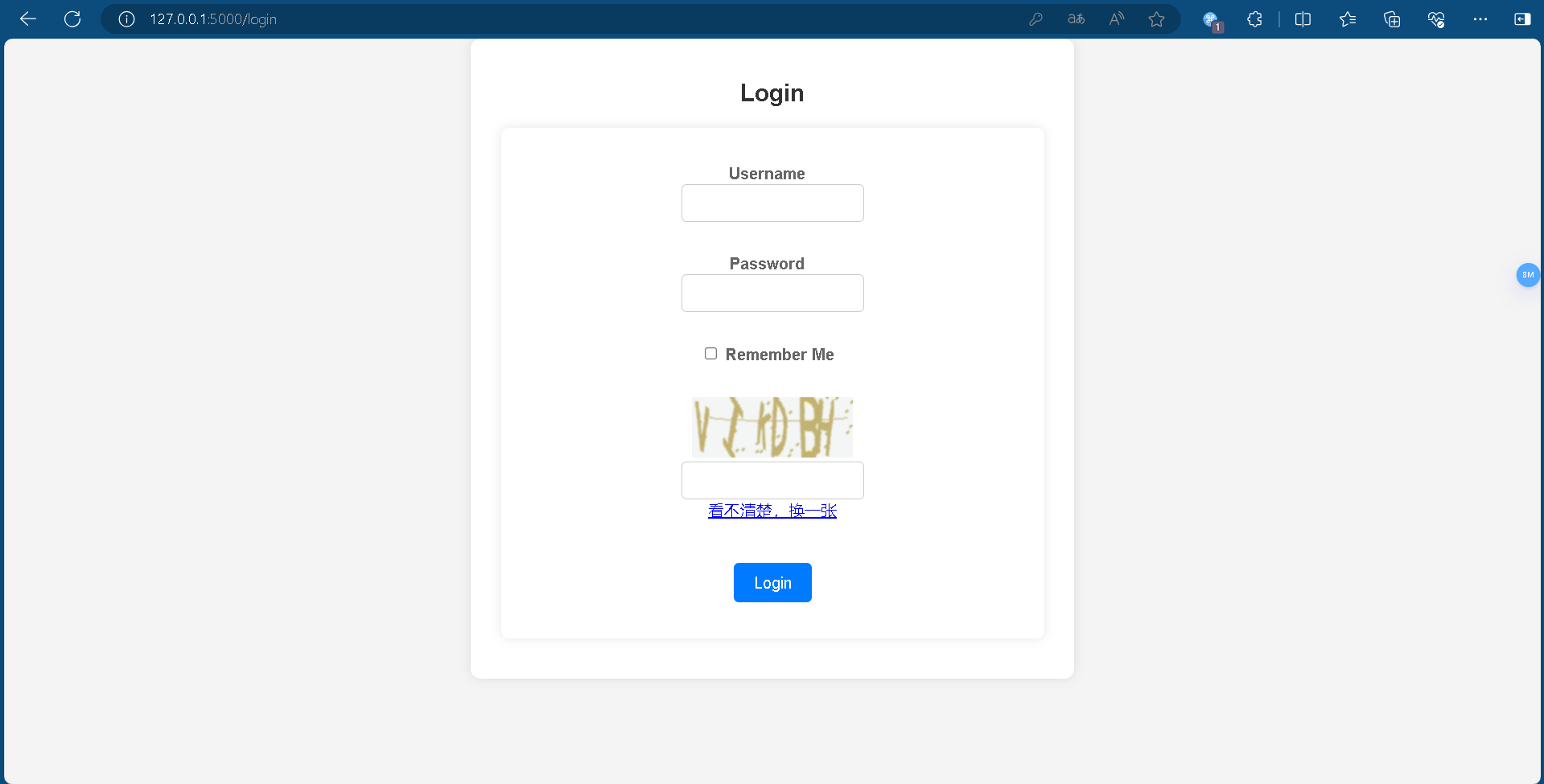


图 7 重新生成验证码

1. 验证码刷新

当用户点击验证码附近的“看不清楚，换一张”时，系统增加的校验码更新功能（写明验证码刷新的代码，以及功能截图）

当用户点击“看不清楚，换一张”时，验证码会自动刷新成新的验证码，报告中功能无法动态展示，已将GIF附在附件中

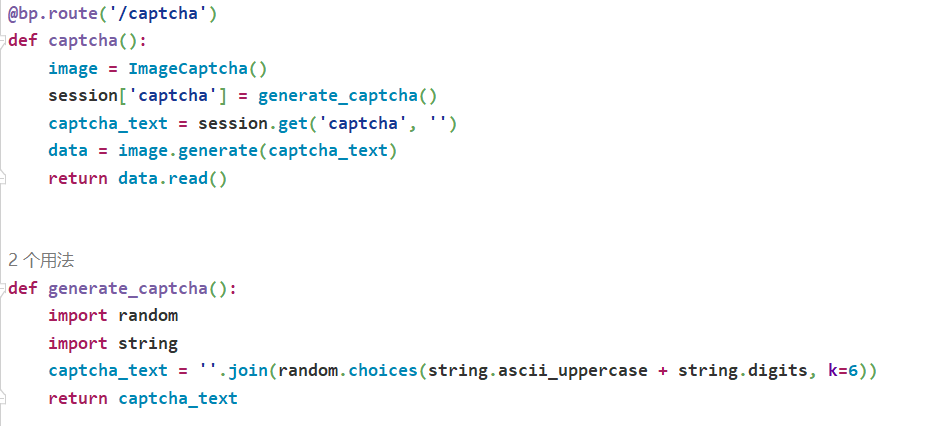


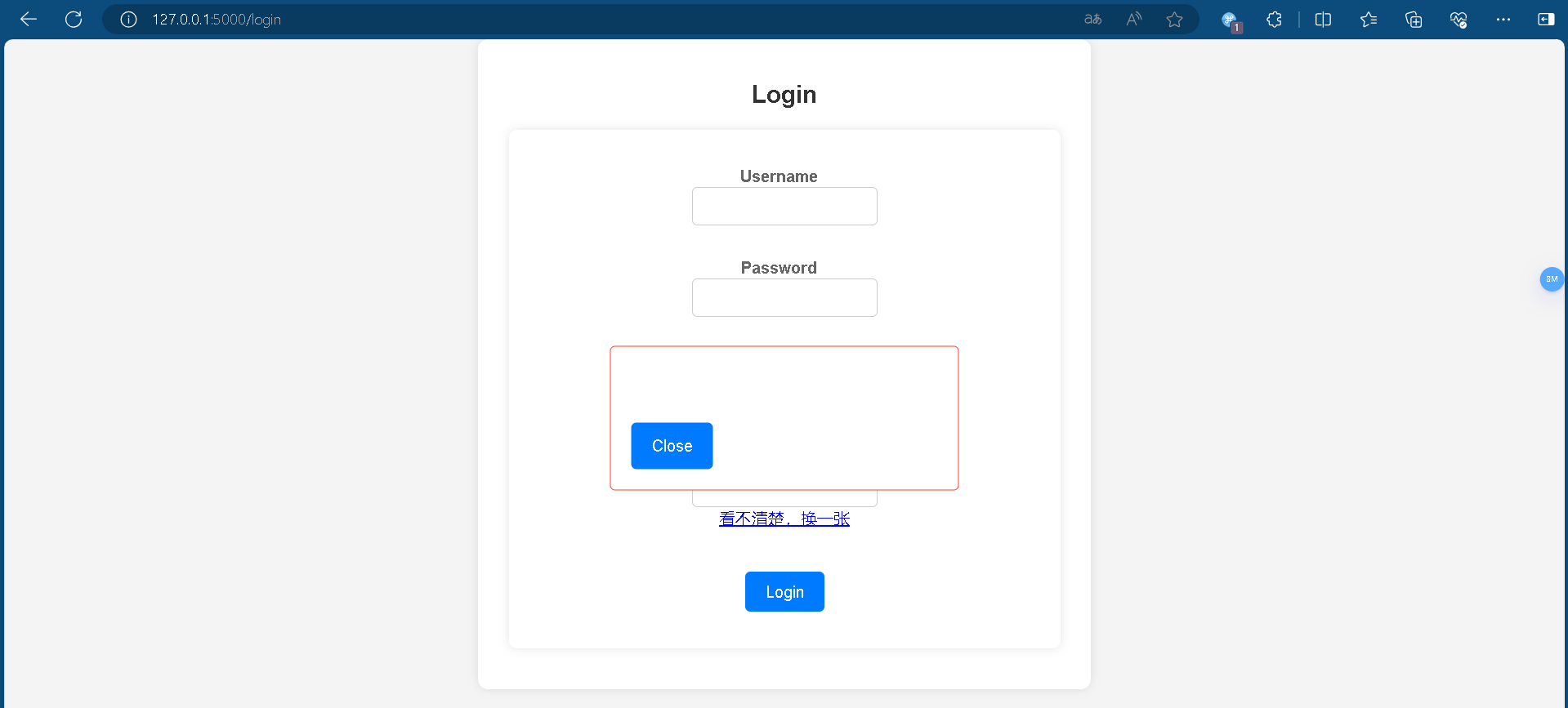
图 8 验证码刷新代码



图 9 前端代码

**五、问题和解决方法**

1. 在进入登录页面时，会自动弹出填写错误警告信息，但没有具体内容，如下图所示



解决方法：在styles.css文件中，errorPopup样式中display设置成了flex，导致错误信息弹窗会一直显示在页面上，将display设置为none即可解决问题。

1. 在点击“看不清楚，换一张”后，验证码只是样式发生改变，而里面的具体内容并未改变。

解决方法：在原captcha()函数中，并未有重新生成验证码内容函数，调用此函数只会改变验证码的图样，在此函数中加入调用验证码生成函数generate\_captcha()函数即可做到验证码内容和样式一同更新。

**六、实验总结**

通过本次实验，我对Flask框架的基本使用有了更深入的了解，实现了“验证码”功能，用于区分登录者是人或是机器。实验中，我遇到了一些问题和挑战，例如前端页面的编写、表单验证、前端与后端数据交互等。通过查阅文档和调试代码，成功解决了这些问题，并实现了预期的功能。

主要收获：

* Flask 框架的基本使用：熟悉了Flask的项目结构、路由定义、模板渲染等基本用法。
* 用户认证与权限管理：掌握了如何使用Flask-Login进行用户登录管理，以及如何基于用户身份实现权限控制。
* 数据库操作：通过SQLAlchemy进行数据库建模和操作，掌握了基本的增删改查操作。
* 前后端交互：了解了如何使用Flask-WTF进行表单处理，以及如何通过JavaScript实现前端交互功能。

**七、附件**

提交可执行的原程序压缩包，随电子版实验报告提交，文件名称为学号+姓名+实验八。