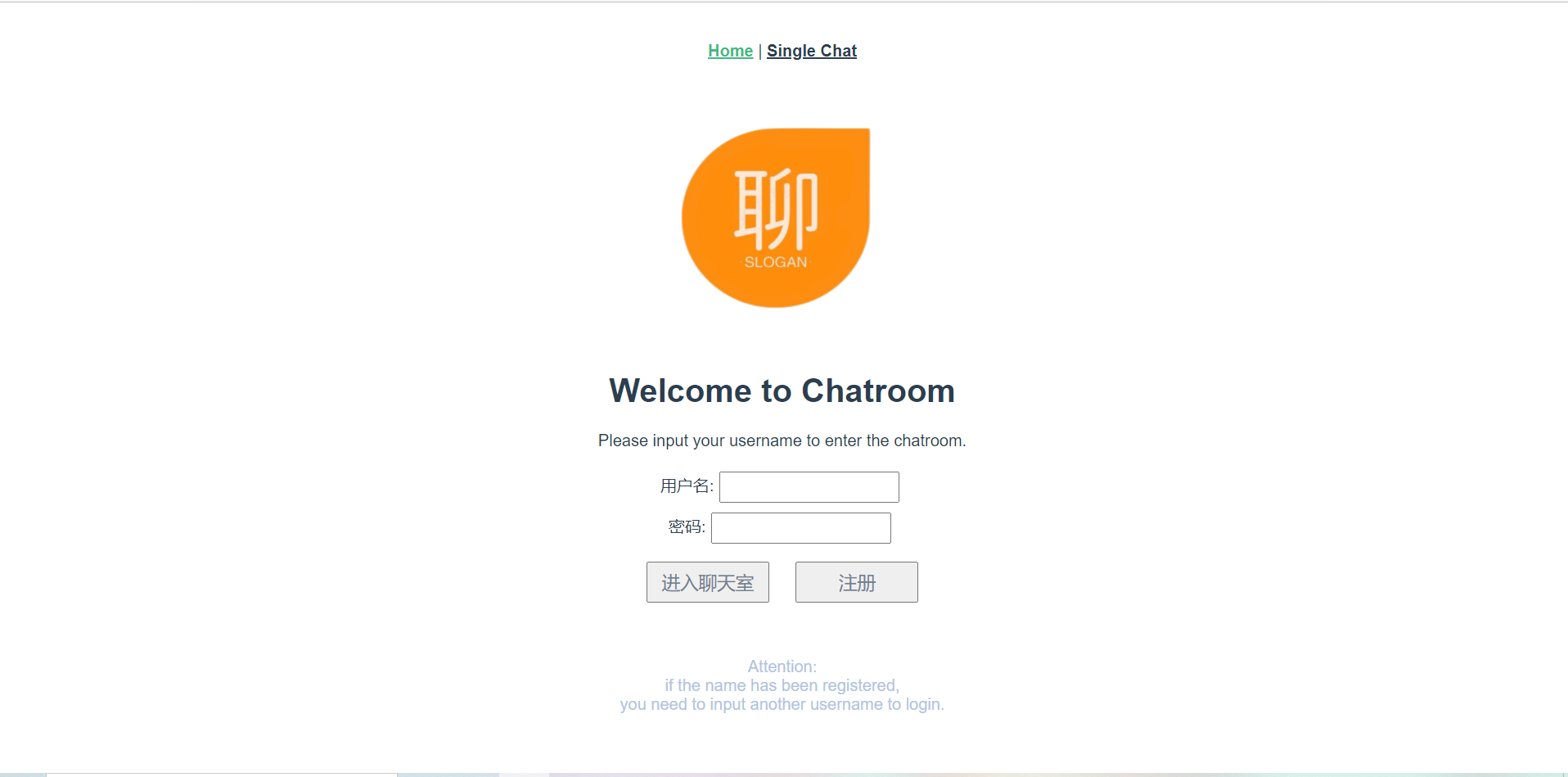
目前已实现功能：

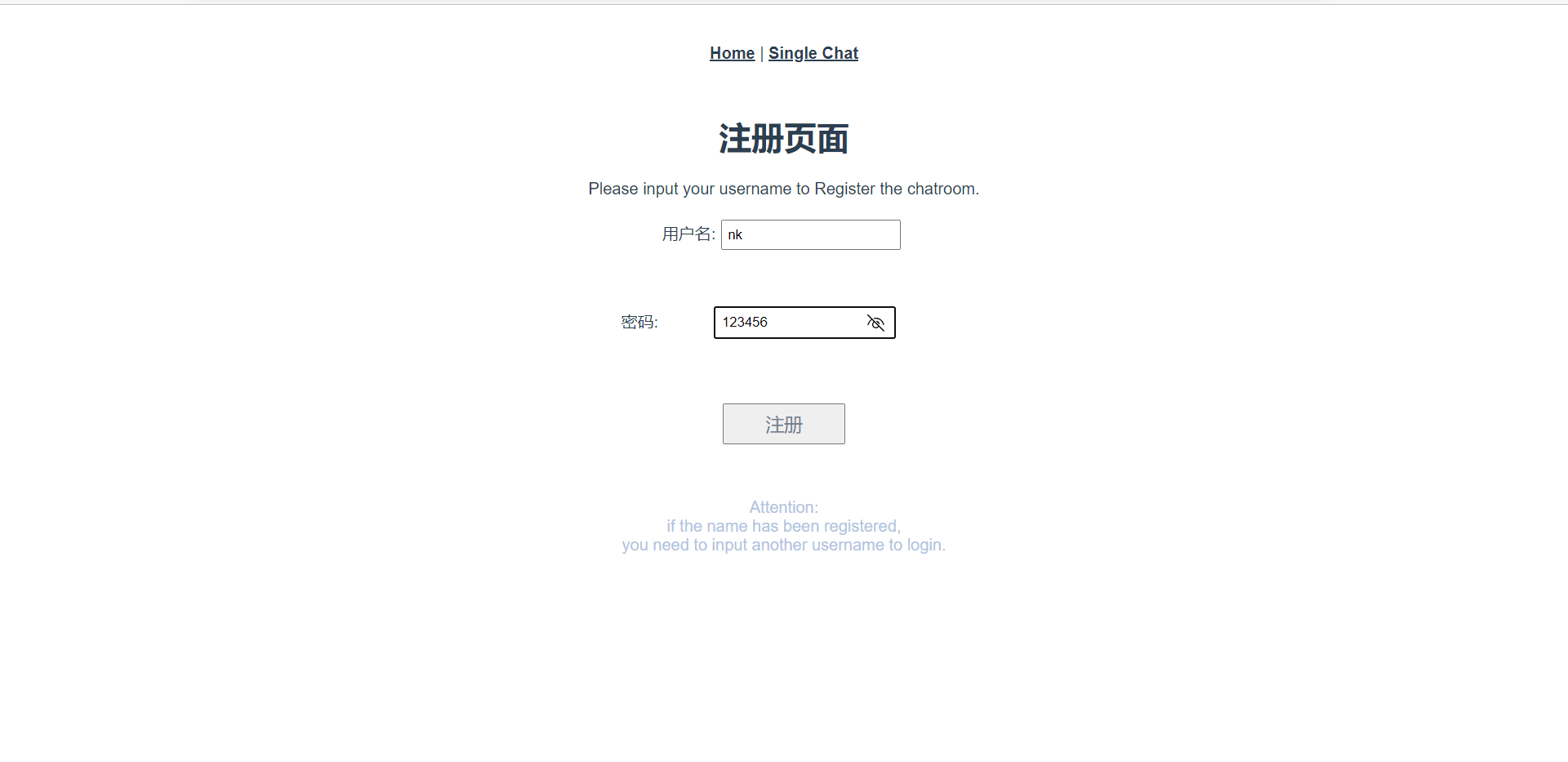
1. 注册账号
2. 登录账号
3. 单人对单人聊天
4. 聊天室聊天
5. 文件传输（实验一已经完成，正在优化）

实现界面如下：

1. 注册账号
2. 启动vue和后台Application，同时启动RabbitMQ中间件（通过Docker Desktop），在浏览器地址框种输入<http://localhost:8080/>，进入主页面



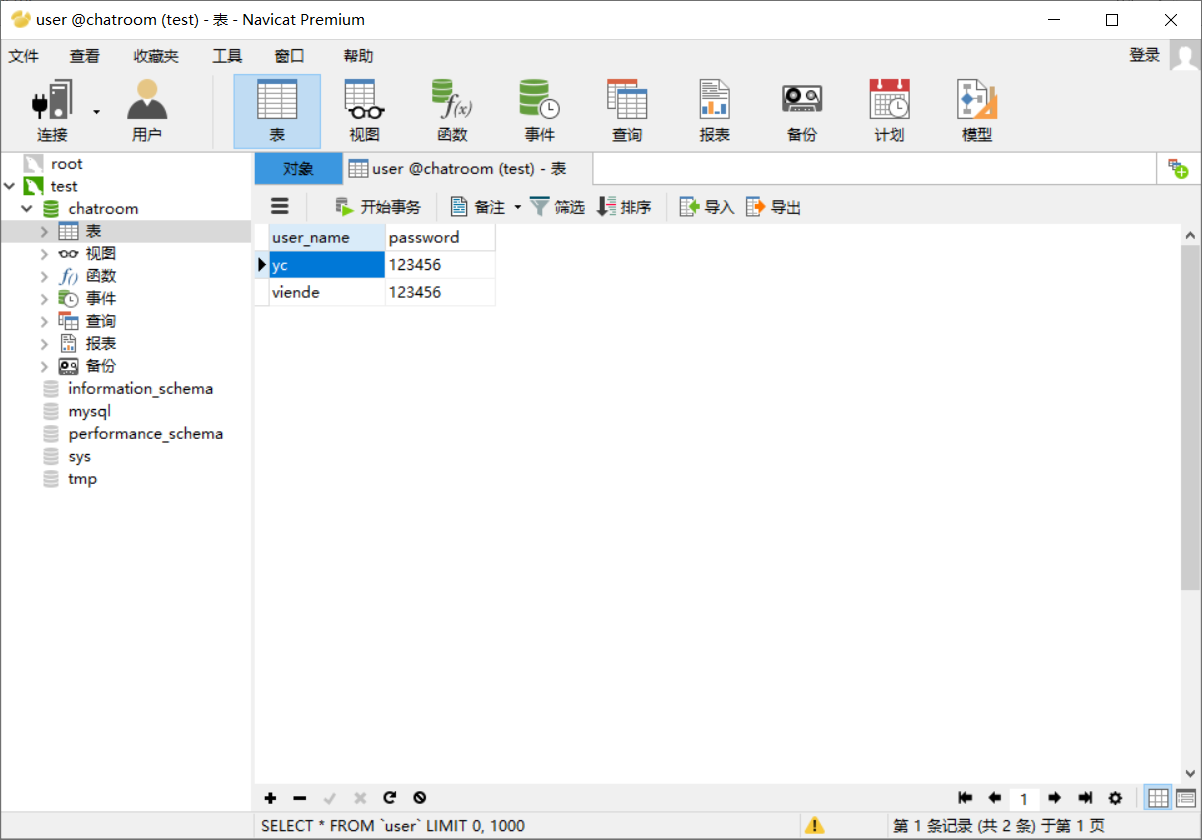
1. 点击注册按钮，输入用户名和密码



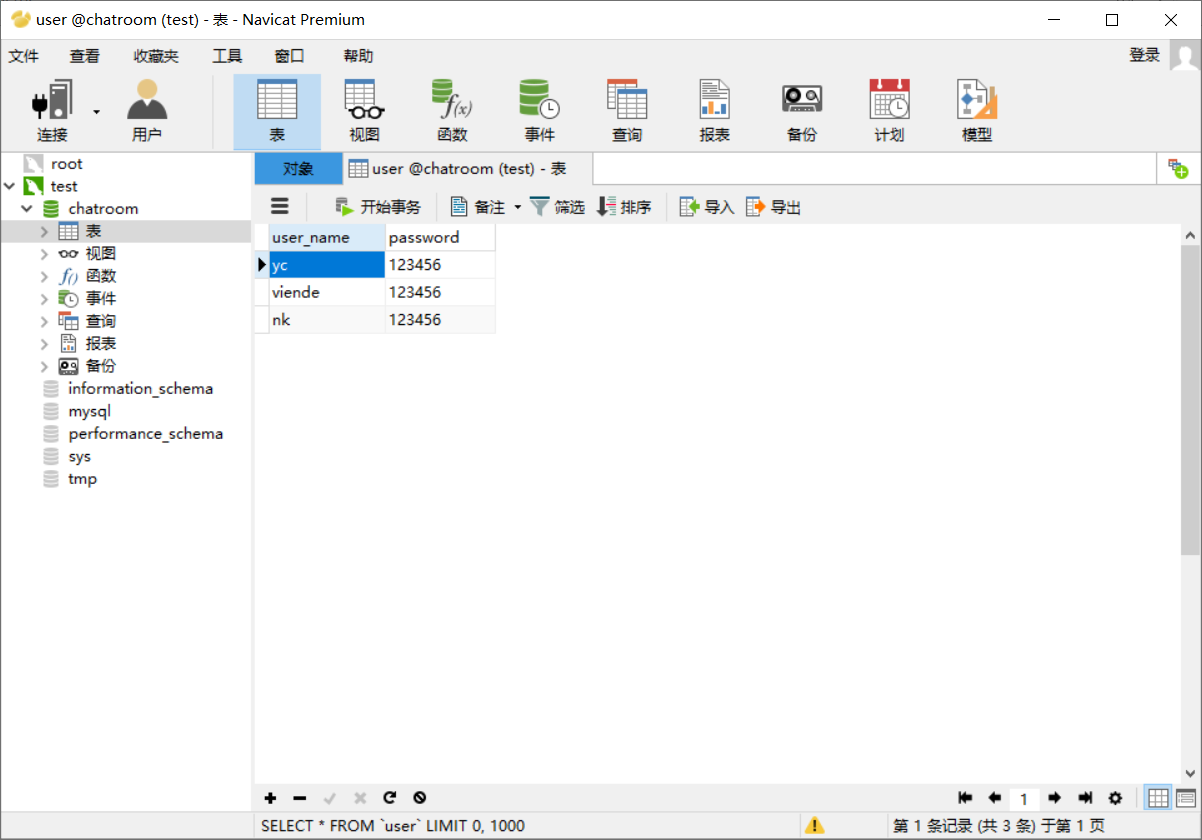


通过前端js代码传给后端。

注册之前的user表：



注册之后的user表：

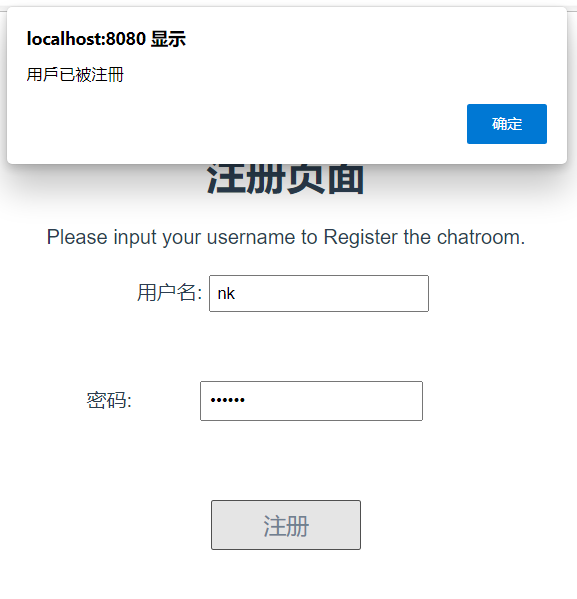


特殊情况处理：

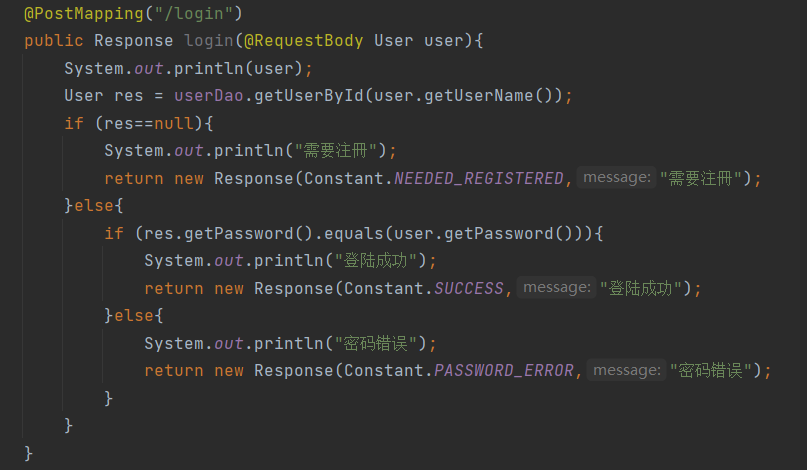
在Controller层中进行判断：



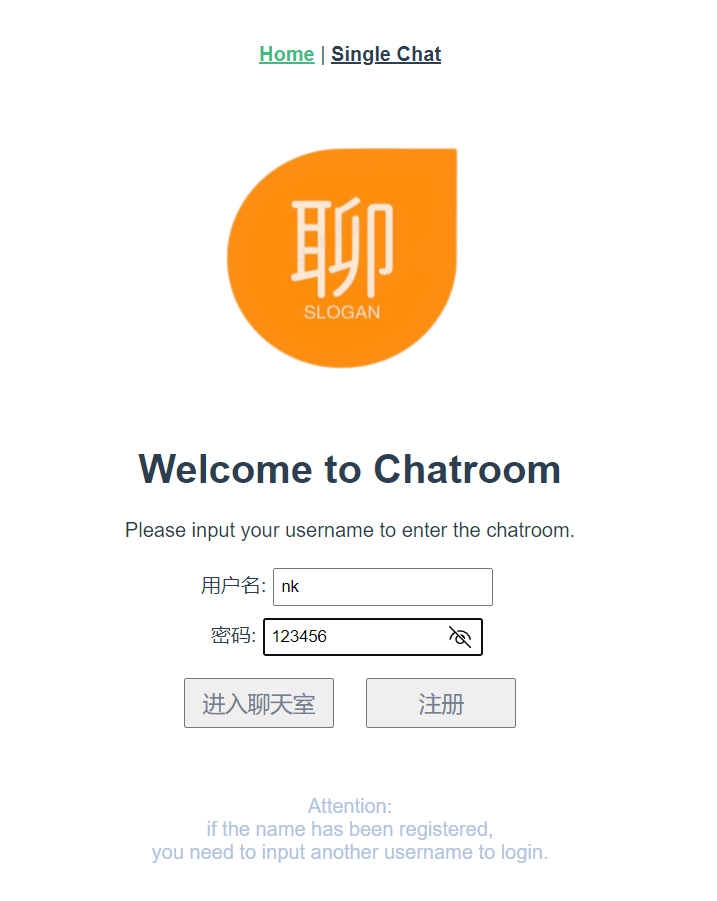
如果尝试注册一个已经被注册的用户名，会出现弹窗错误：



1. 登陆账号
2. 判断多种情况



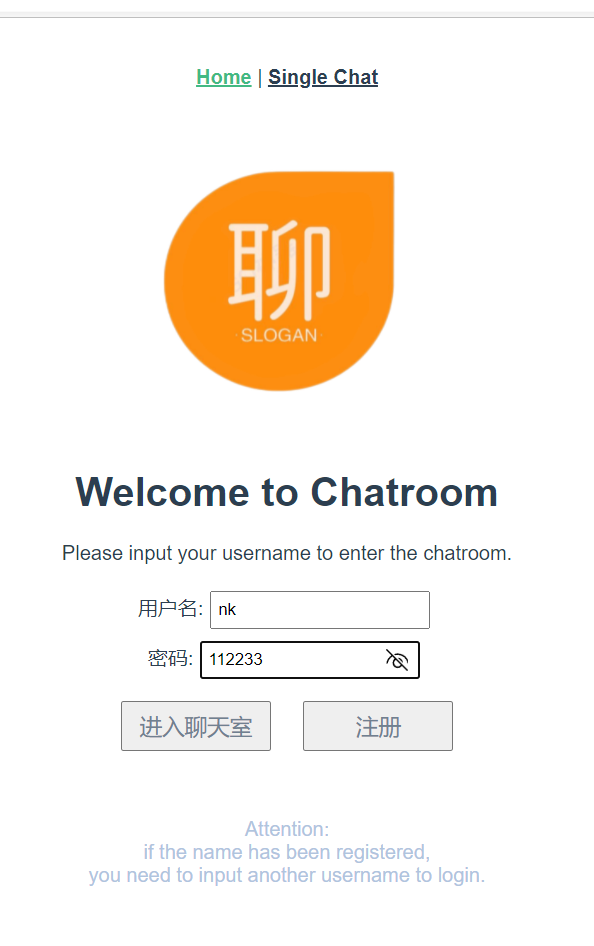
输入正确的账号密码，登陆成功，跳转到群聊聊天室界面



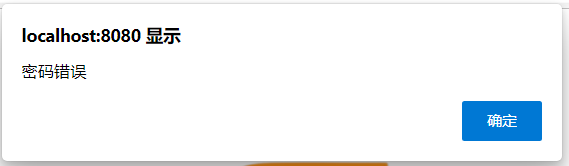
此时，使用者作为nk在聊天室中进行聊天。



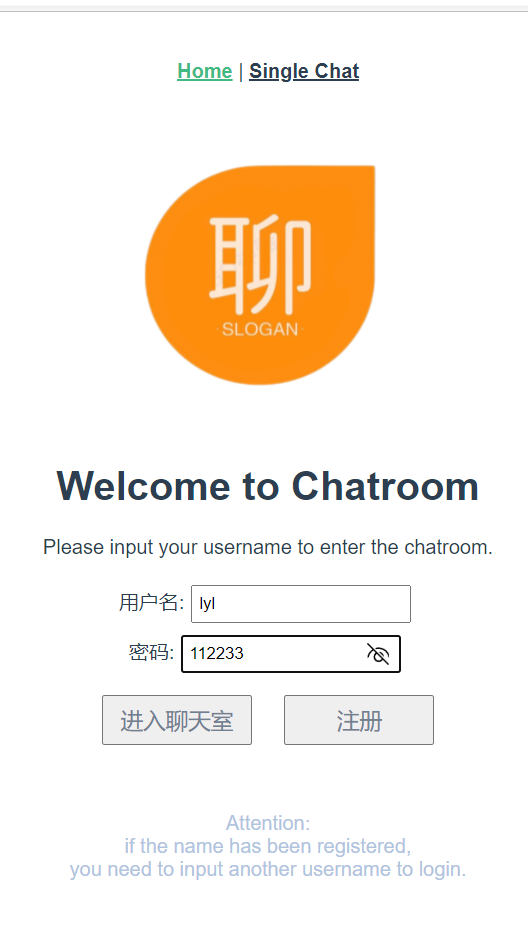
1. 输入错误的账号密码



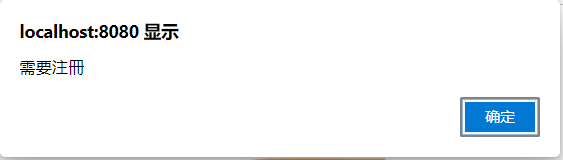
此时弹窗错误：



1. 输入没有注册过的账号

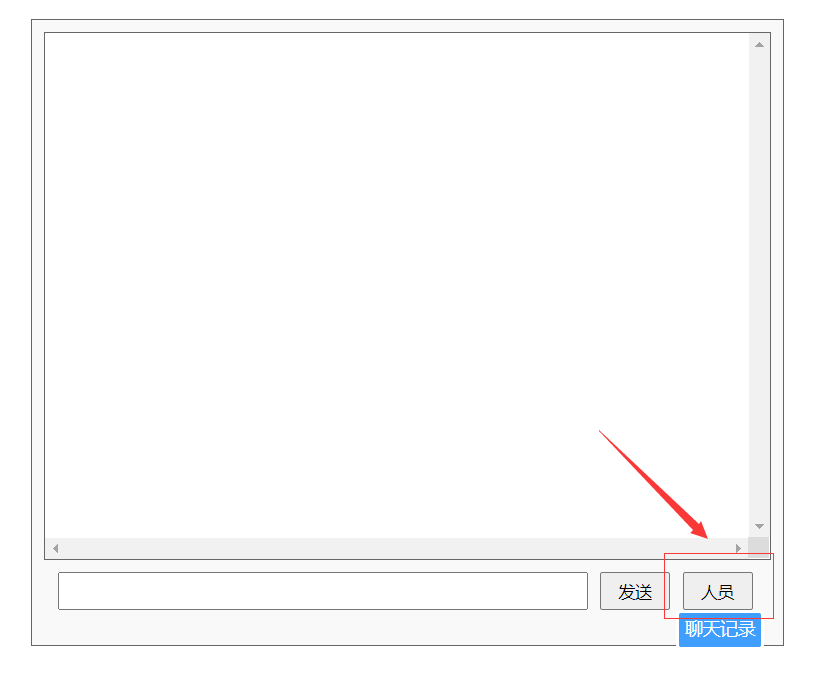


跳出弹窗错误：



1. 单人对单人聊天

作为nk进入聊天列表。



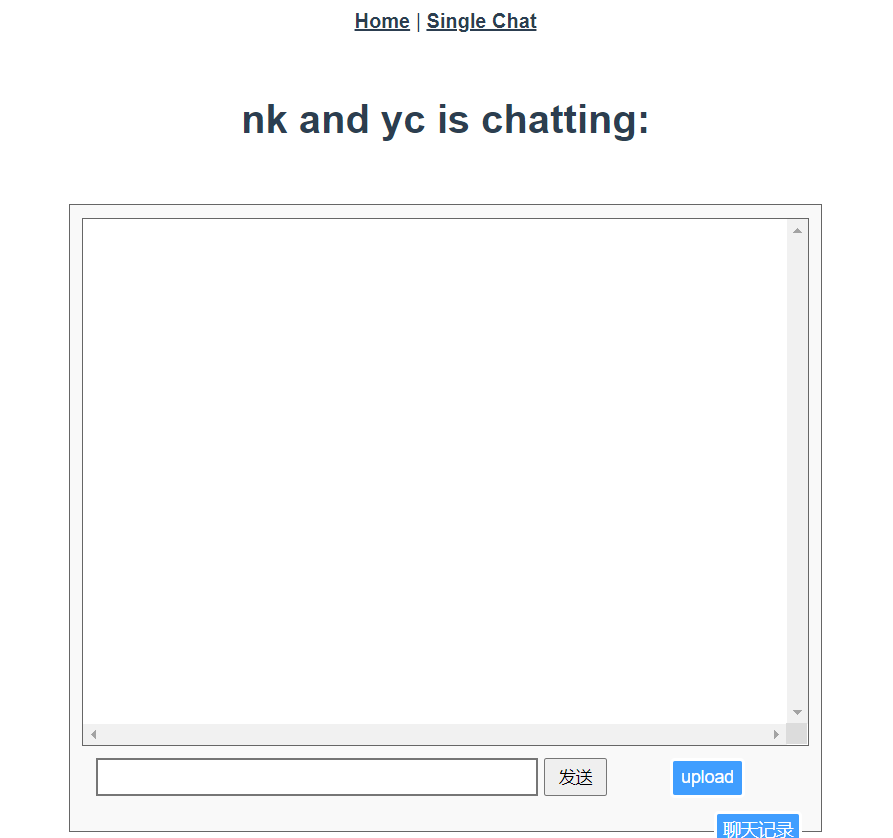
点击人员，查看群聊当中的人员。

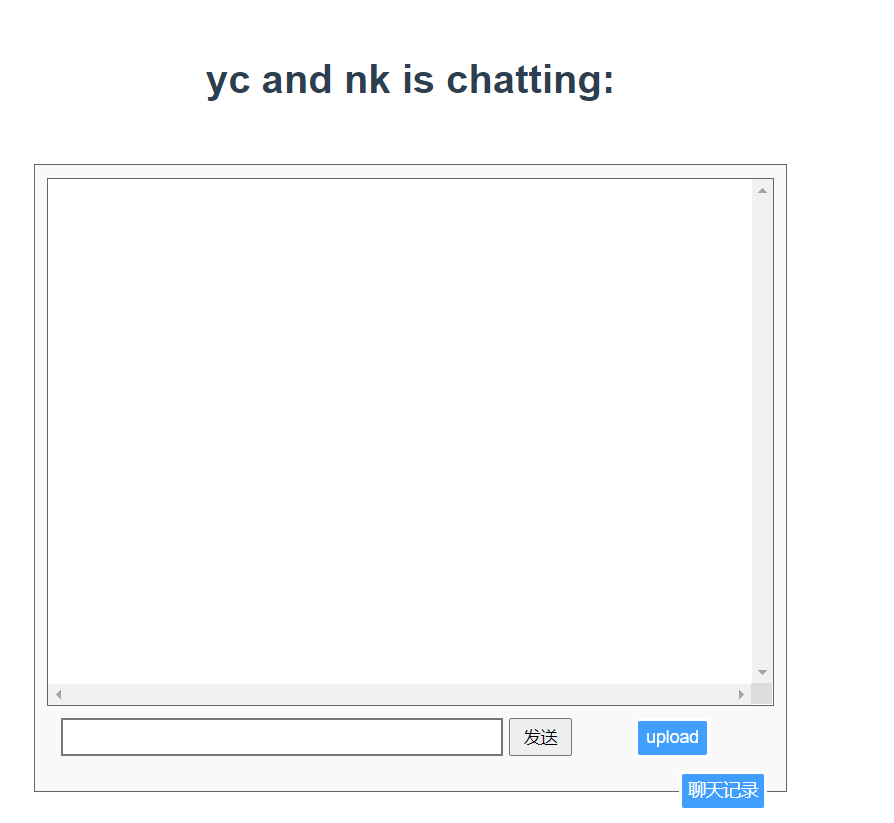
选择yc作为聊天对象。



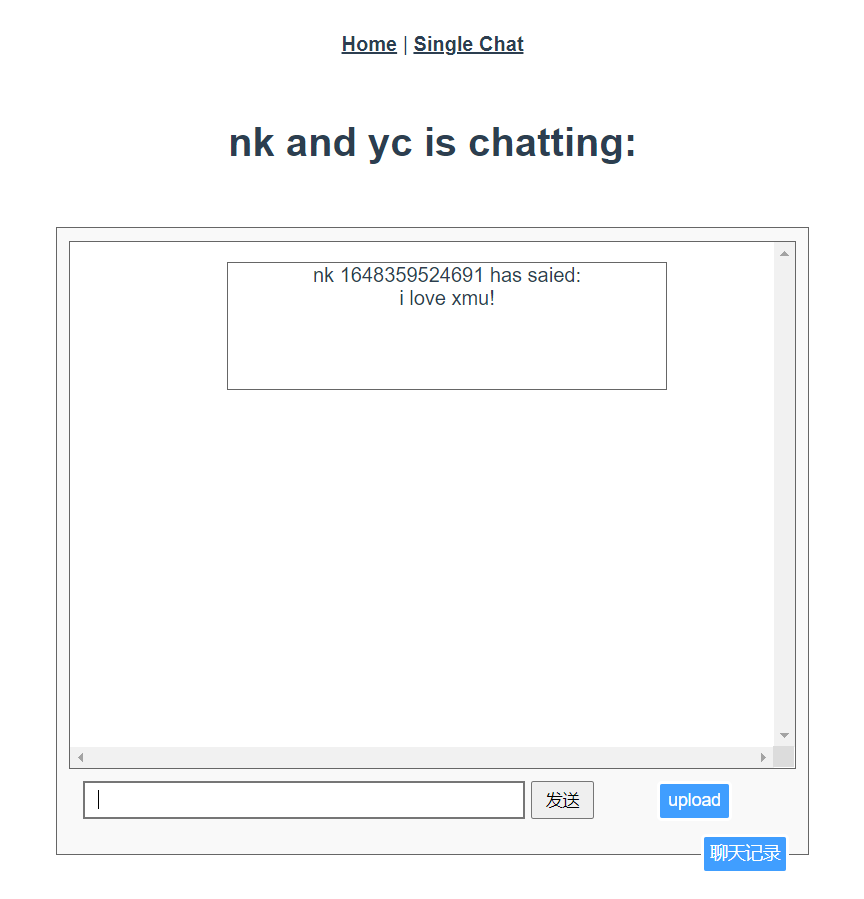
此时跳转到单人聊天界面。

此时打开另外一个localhost界面，登录yc账号，对应另外一个聊天界面。

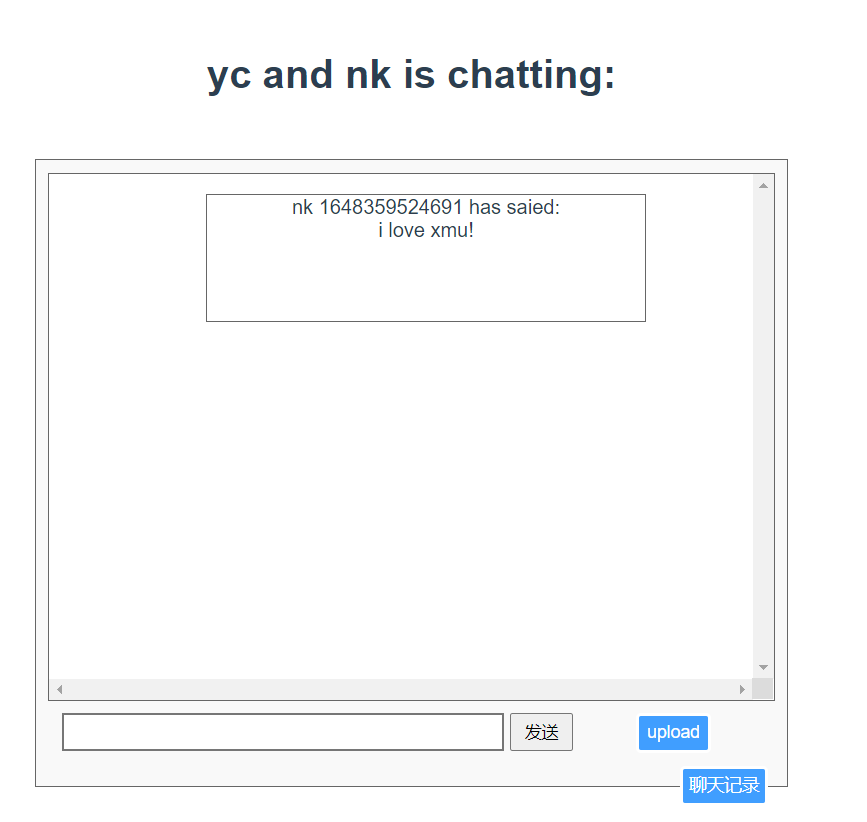




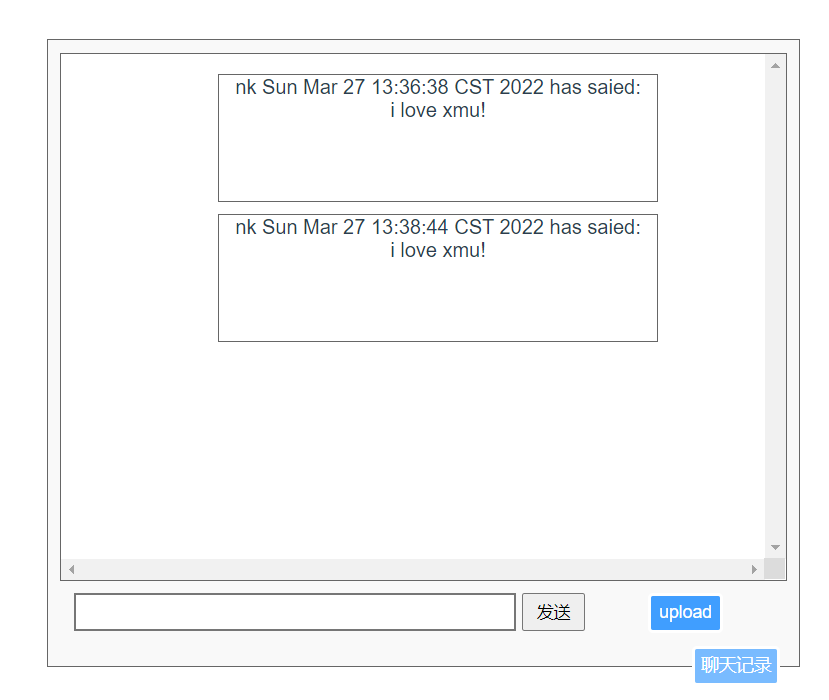
yc输入文字并发送：



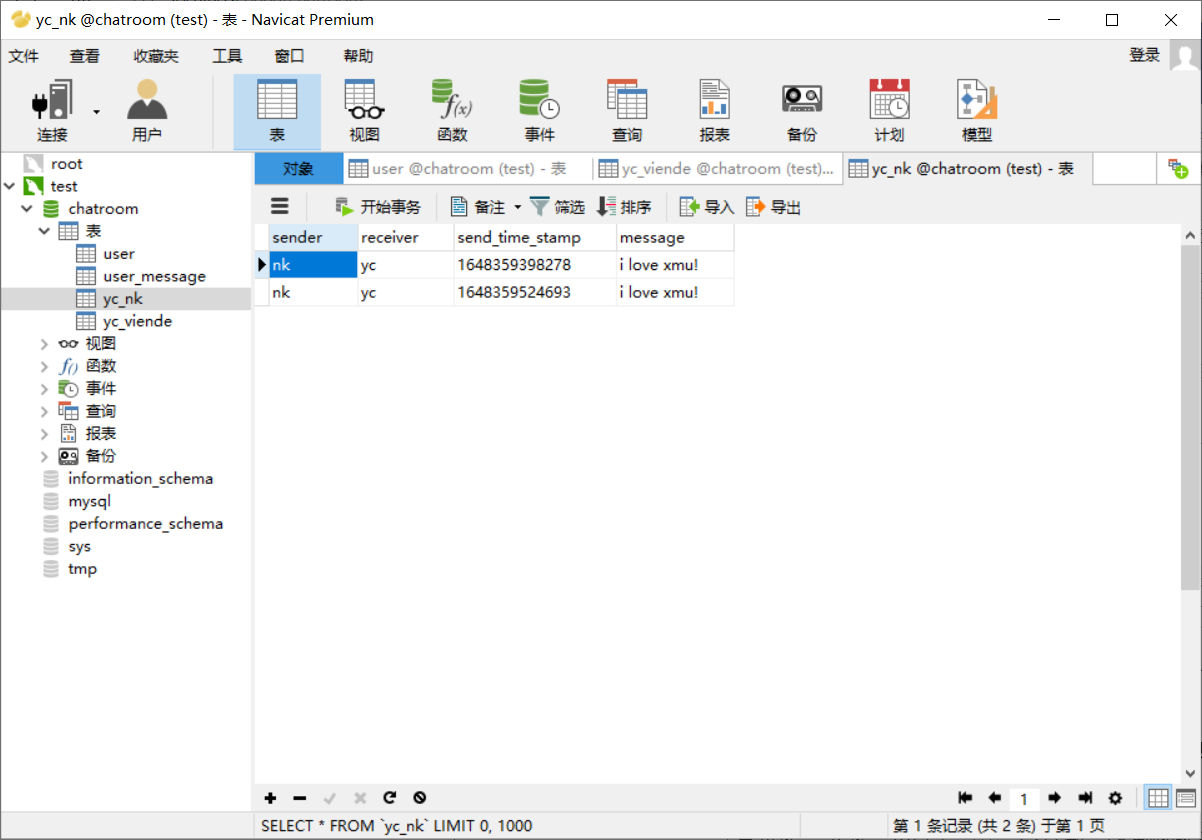
同时，yc账号的聊天框也出现了对应信息。



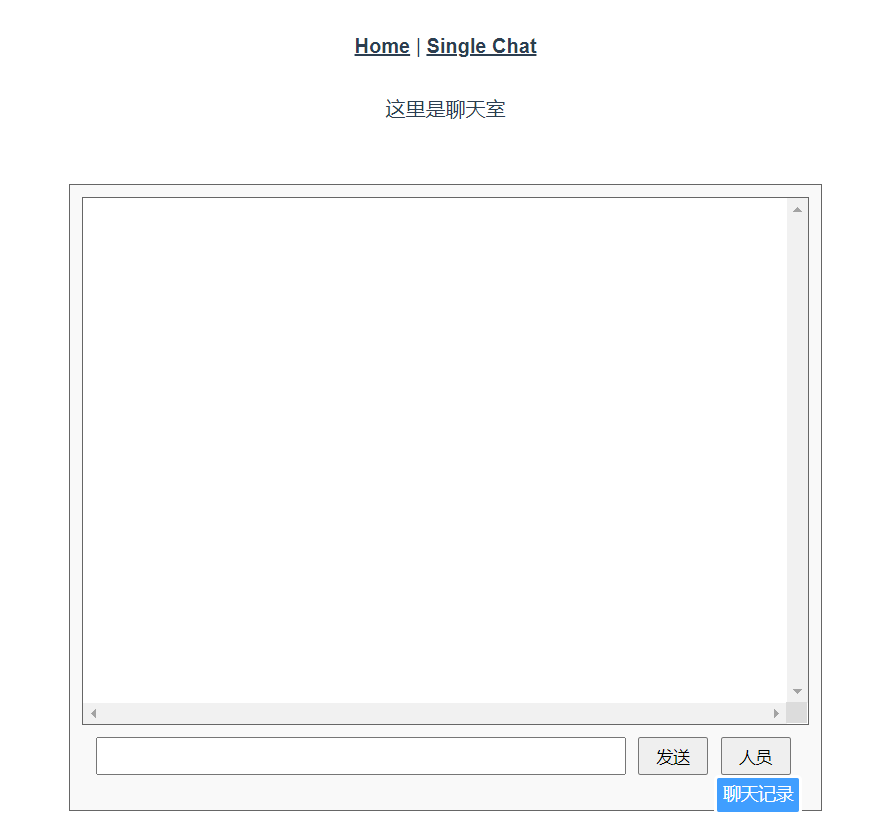
点击聊天记录按钮，查看聊天历史记录：



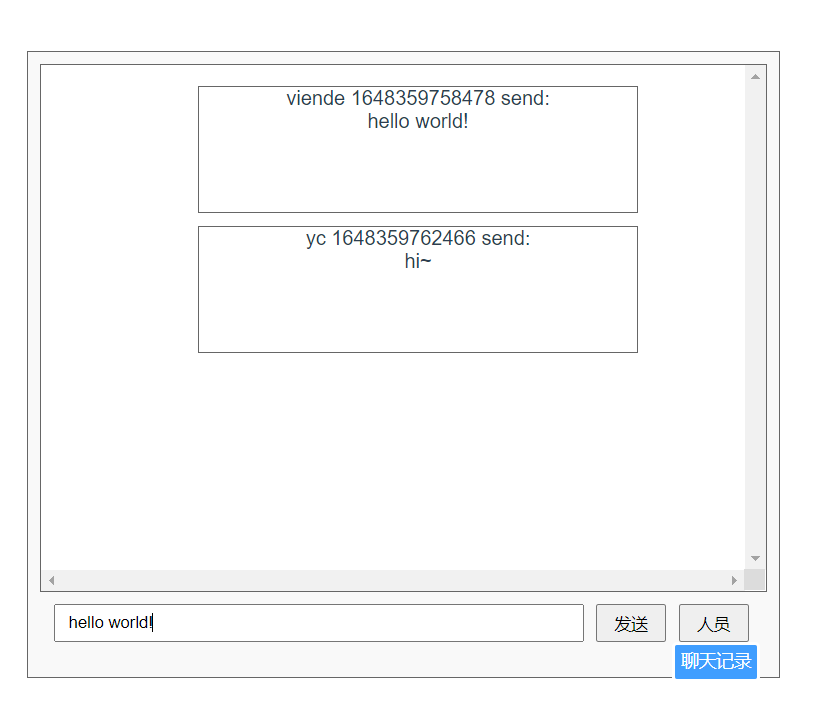
查看数据库，聊天信息已被存入yc\_nk表中：



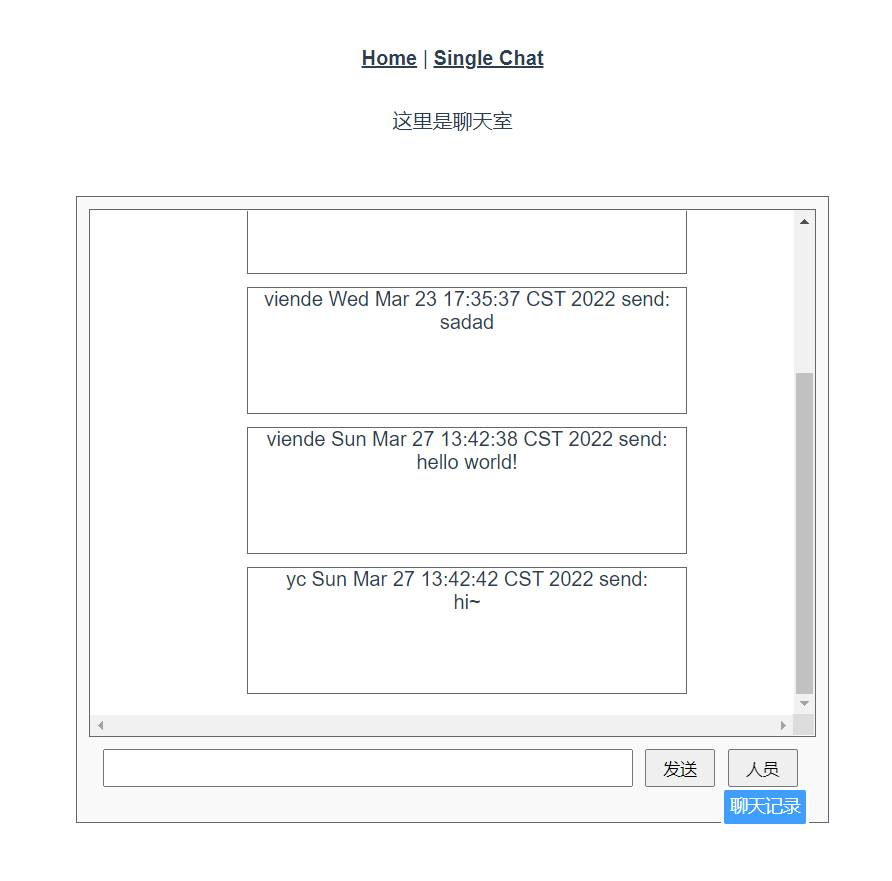
1. 聊天室聊天
2. 分别登录yc和viende账号。
3. 进入聊天室界面。



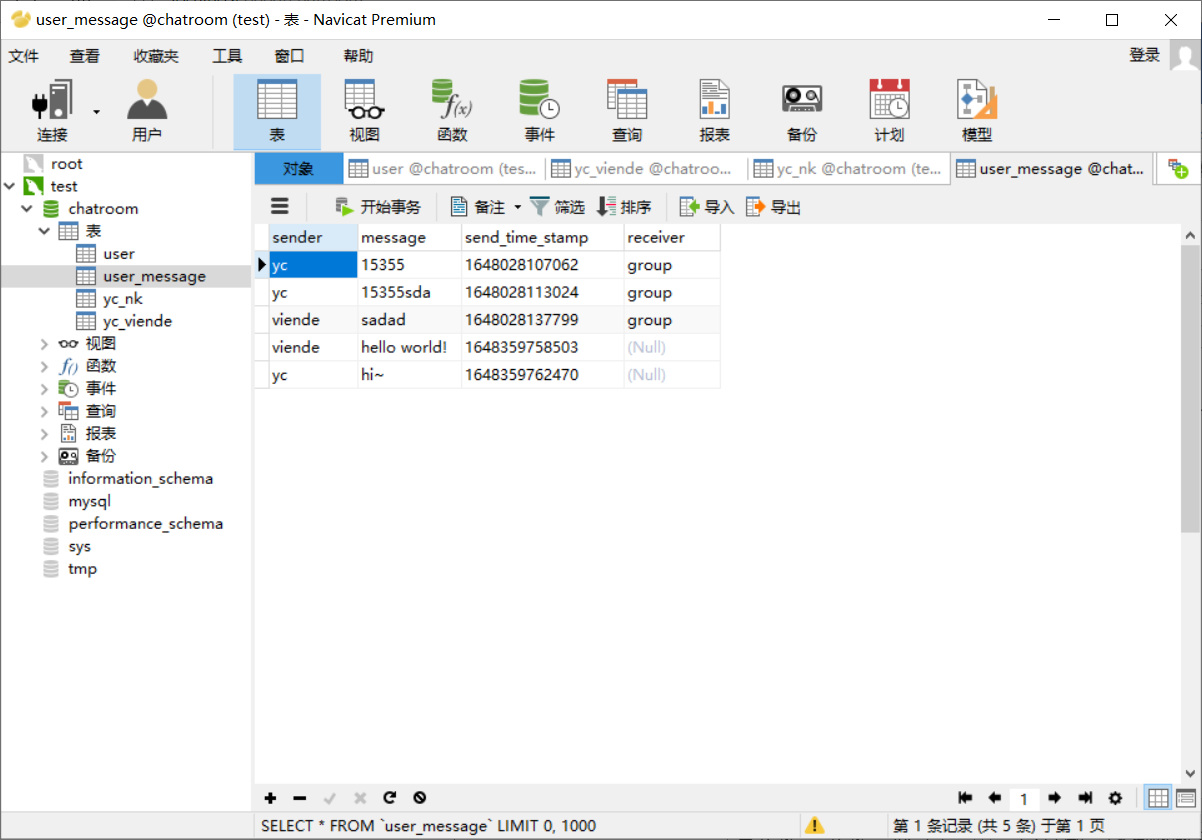
1. viende输入聊天内容，yc输入聊天内容。



1. 点击查看聊天纪录

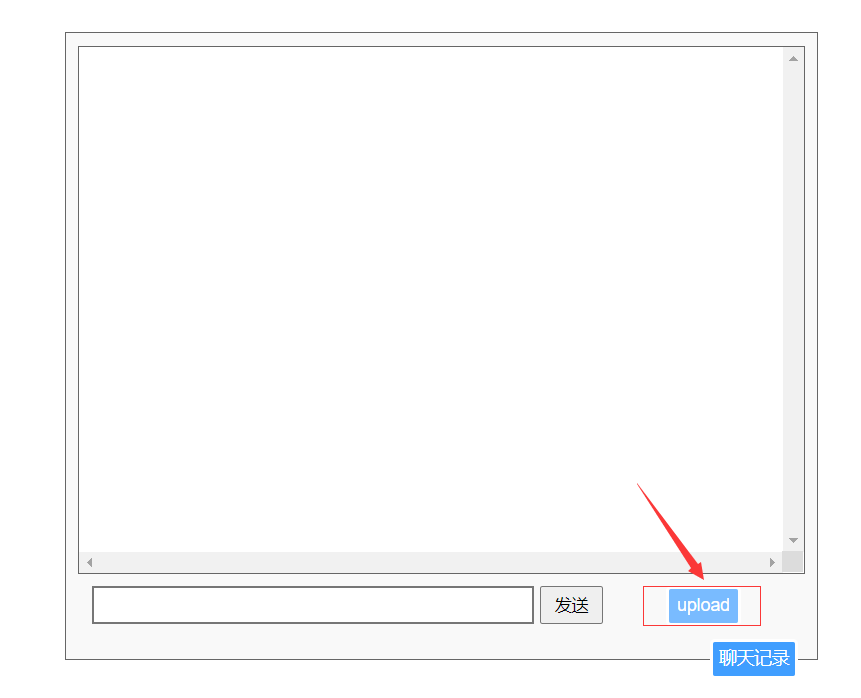


查看数据库，聊天信息已经被存储到user\_message表中：

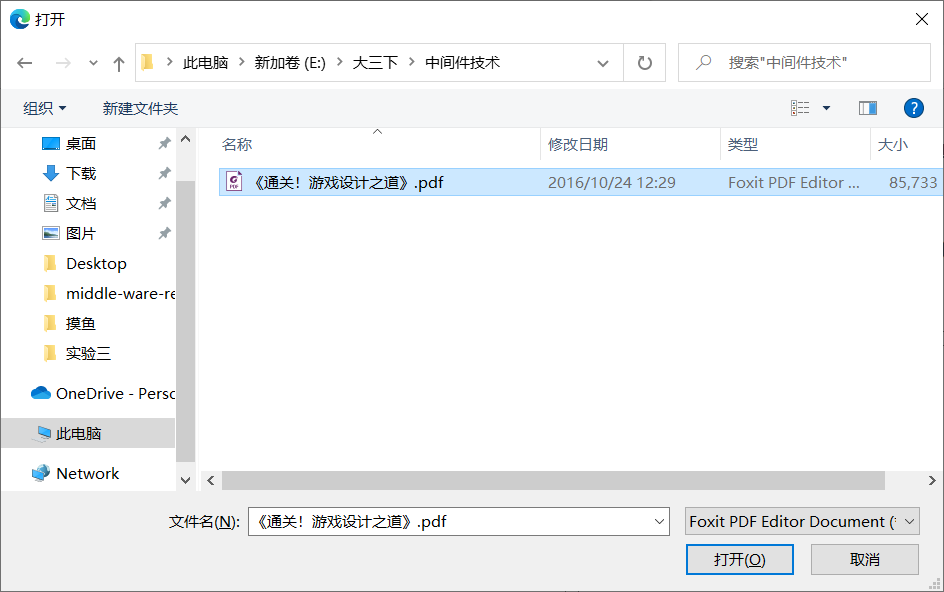


1. 文件传输

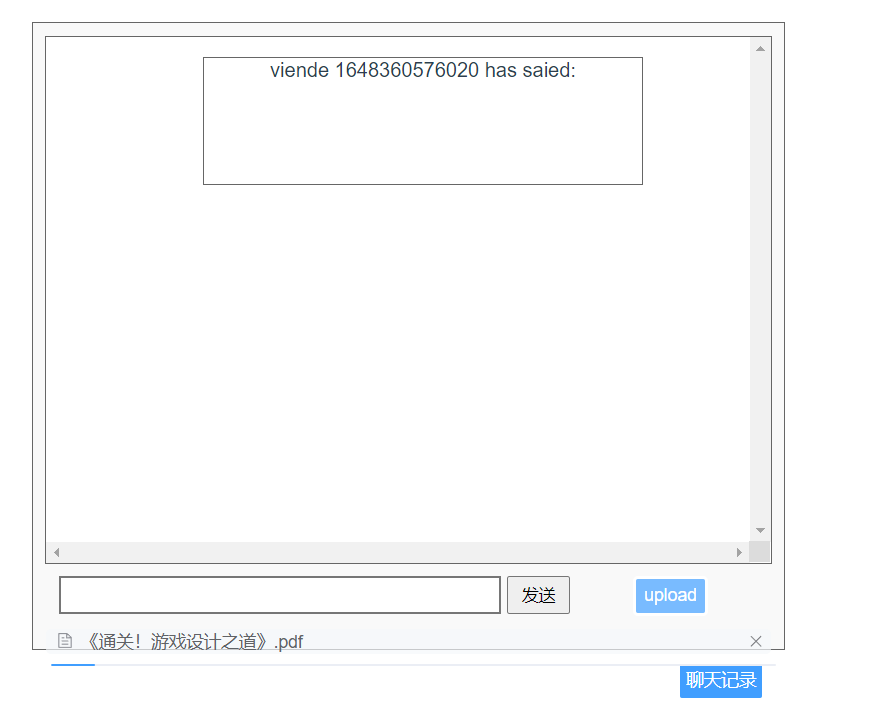
进入单人聊天框，点击upload按钮。



选择传输的文件：



文件开始传送：



RabbitMQ文件存储转发实现步骤：

* 1. 用户A上传文件，后端将文件内容转化为二进制编码。
  2. RabbitMQ将消息存储到queue中。
  3. 检测用户B是否上线，若用户B上线，则queue将转化为二进制编码的文件转发给用户B。

异步转发的好处：

1. 多个应用解耦，当数据库出现故障,消息队列也能保证消息的可靠投递,不会导致消息丢失。
2. 应用内的同步变异步，大量线程的阻塞会降低计算机的性能。
3. 消息驱动，系统分解为消息队列，和消息制造者和消息消费者，一个处理流程可以根据需要拆成多个阶段（Stage），阶段之间用队列连接起来，前一个阶段处理的结果放入队列，后一个阶段从队列中获取消息继续处理。