**一个简单的**

**论坛管理系统**

**第六组**

**夏搏远**

**柯世纪**

**江星瑶**

**邓星雨**

**魏千懿**

目录

[1、目标与应用场景 3](#_Toc31308)

[2、设计思路 3](#_Toc27287)

[3、关键问题与实现代码 4](#_Toc11608)

[4、数据库结构 9](#_Toc10845)

[5、程序主要代码及其说明 10](#_Toc8929)

[6、运行效果图 21](#_Toc27720)

# 1、目标与应用场景

我们小组的任务是建立一个论坛，用户注册并登录后可以随时随地在论坛上发布自己的心情。这些帖子会显示在我们论坛的主页上，同时主页上还有实时的热门帖子，点击主页的图片或帖子标题，任何用户都可以看到这篇帖子，并且标明在题目下方的用户名、发布时间等信息也可以保证用户对帖子的所有权不被侵害。其他用户可以在帖子下方发表自己的回复来与原帖作者互动，并点击作者名字进入其主页。每位用户都有自己的个人主页，能够按时间顺序查看自己发表的评论，也可以通过点击评论回看原帖，回忆当时自己评论的心情。

我们的论坛让用户可以抒发自己当下的心情和感受，发表自己对某件事情的看法，并与其他人产生互动，进行思想的交流。我们希望人们能够拥有一个可以自由发表看法的封闭式平台，可以与人交互，但不至于受他人的言论影响太多。

# 2、设计思路

论坛，作为一个充满了互动、社交意味的名词，我们认为最重要的部分当属它的主页，其通前端登录内含热门帖最新帖标题后接具体的帖子和评论，是整个论坛的核心，因而我们将其作为全组论坛搭建工作的中心点，全组人一起规划设计主页与其他页面及其功能的连接点，并由此衍生出各自的任务。

首先，我们设计了注册与登录页面，注册登录、发帖功能、回到主页三个按钮一起构成了所有页面右上角的跳转口。点击登录/注册按钮后，首先弹出登录页面，页面下方与大多数网页一样，标识有“还未注册？注册账户”一行字，以供初次使用者注册成为我们的用户。根据密码的正确与否、登录的发生与否、网络连接是否顺利，登录结果也存在四种可能。当密码正确、未重复登录且网络通畅时，登录成功，用户进入主页，主页上方的登录与注册按钮也自动变为用户id。

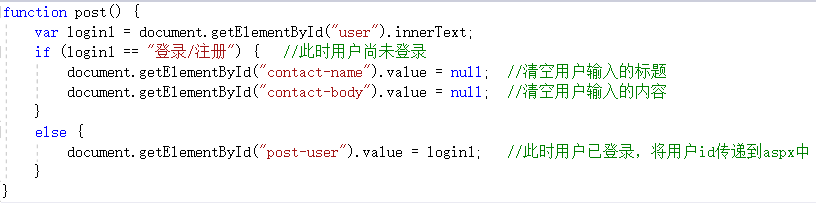
其次，我们设计了发帖页面。发帖中包含了自动审核功能，可以识别文字中的敏感内容并弹出‘含有敏感词’弹窗并返回发帖页面，让用户可以继续编辑含有敏感词的文字，通过审核后，系统弹出‘发布成功’弹窗并跳转回主页。

之后是点击主页中的帖子标题后，正文下方的评论界面。评论同样具有审核功能，发布成功后会加载到原贴的基础上，页面自动刷新为新的帖子页面，这样用户就能够直接欣赏到自己评论后的帖子内容。

最后我们打造了主页右上角的搜索框。在搜索框中输入用户id即可搜索其评论，这些按时间顺序排列眼前，评论后还附了评论过的帖子，点击评论后还能回到这条评论所发布的帖子。

# 3、关键问题与实现代码

**发帖**：如何从网页上得到用户id，即将用户id与帖子连接起来；发帖成功后如何跳转回主页。





IMG_256

**评论**：如何把评论和用户连接到一起（评论和帖子本来就是连在一起的）；如何将评论发回原贴并更新网页。

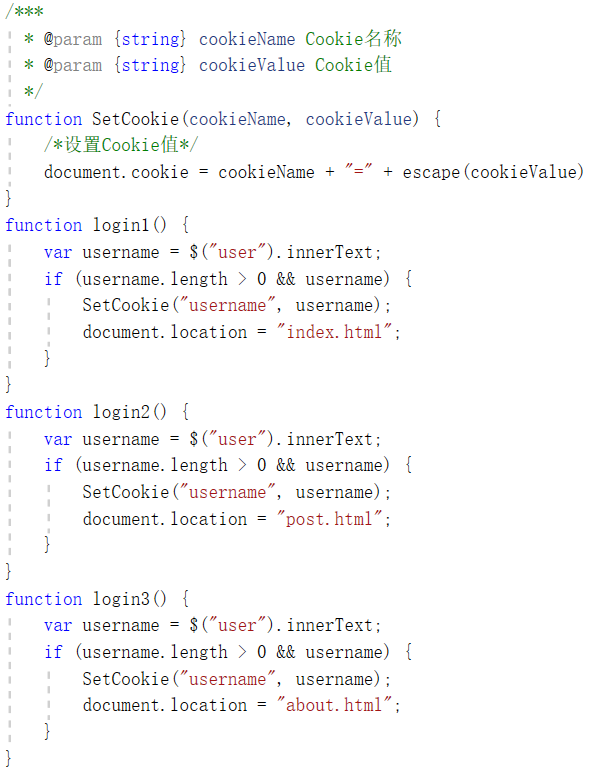


将评论的内容发回到评论页面上：





**登录**：如何用cookie功能记住用户的id。

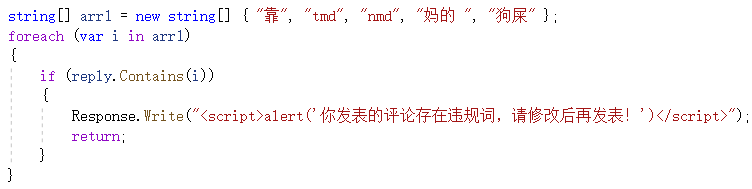




用cookie记住username并传输给init函数，init函数在每一个页面里onload.

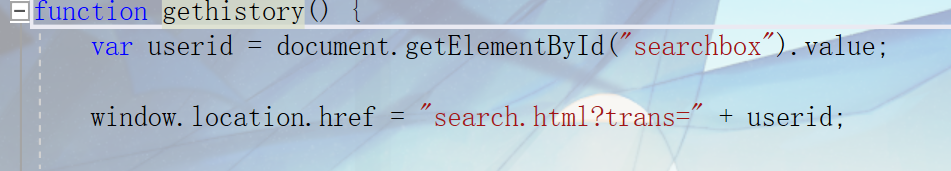
<body onload="init()">

**审核**：本身就是发帖和评论的关键。



**查看评论过的帖子**：匹配id和其评论。

展现本id所发帖子和评论的界面，同时传递参数id：



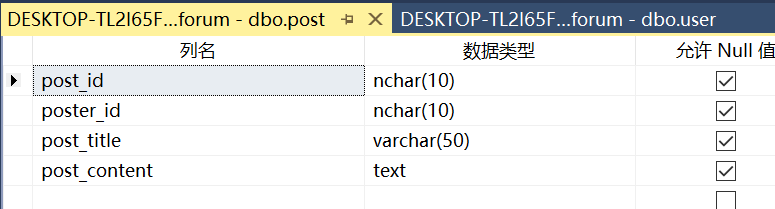
跳转至对应界面，接收id。根据接收的id，与json对象内的id进行匹配，如果匹配成功则展示对应的标题。如果没有查找的，则返回“尚未发帖”：



# 4、数据库结构

数据库三大核心表：用户信息数据库；发帖信息数据库和评论信息数据库。

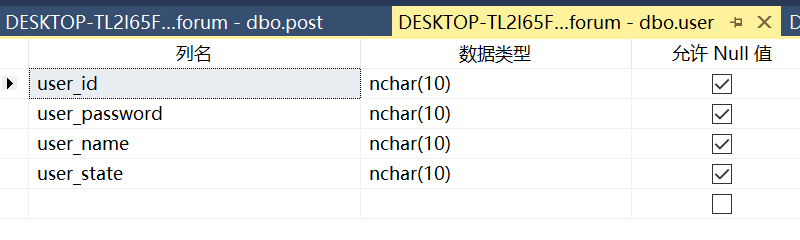
用户信息：



其中User\_id为表的主码，用户注册时输入呢称与密码，系统自动分配id（类似于QQ等常用社交软件）给用户。

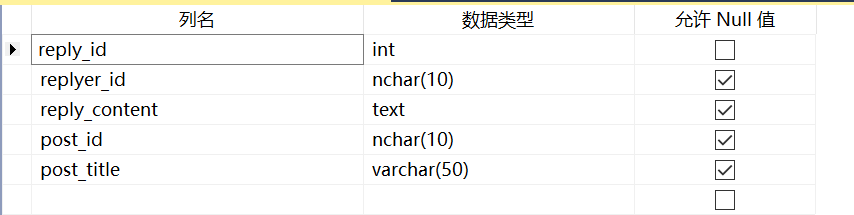
用户使用id和密码登录论坛，用户状态显示用户是否已经登录。

发帖信息：



其中post\_id是表的主码，poster\_id记录帖作者，title和content是帖子的具体内容。

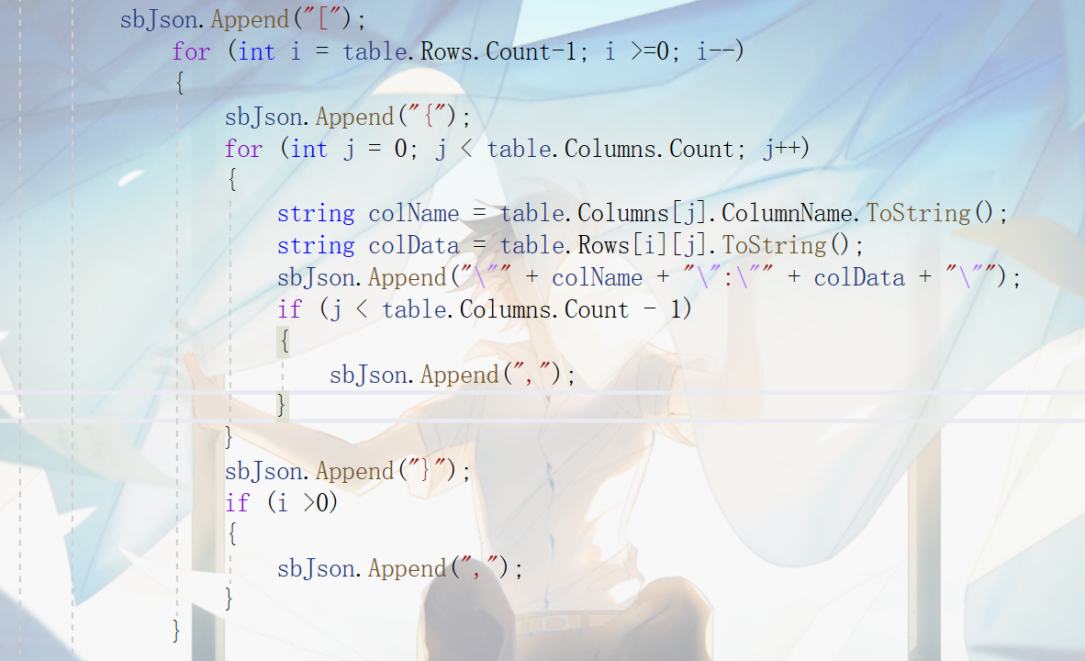
评论信息：

其中reply\_id是表的主码，replyer\_id记录评论者，post\_id记录评论的帖子，content是评论的具体内容。

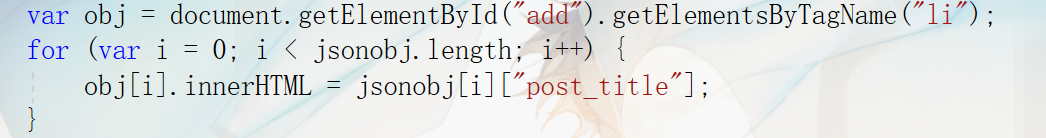
# 5、程序主要代码及其说明

1. **帖子展示部分**

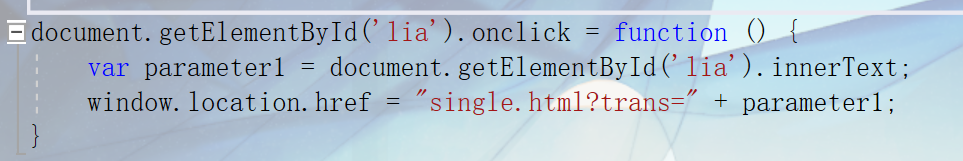
①连接数据库，并将其写成json对象的形式



②展示帖子，将字符串转换为JSON对象，然后对其标题进行获取和展示。

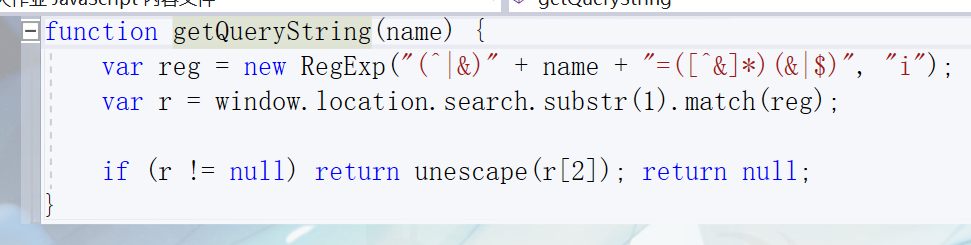


③为每一个展示的对象设立动作onclick(),在点击的同时跳转到其他页面(post.html)，并传递帖子title（post\_title）参数。



④在对应的post.html界面设置对应的接受代码

（接受用函数getQueryString()）

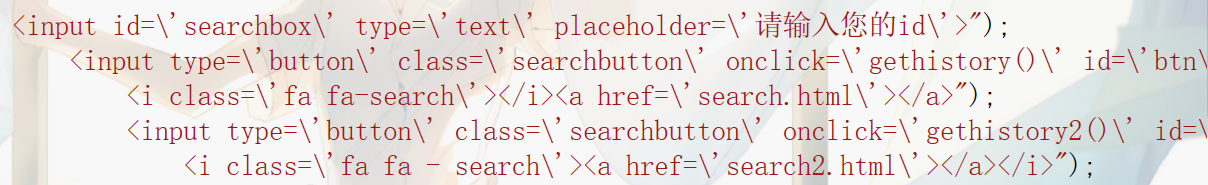


（接收对应的帖子标题，并与json对象进行循环匹配，将匹配成功的内容（post\_content）展现出来）

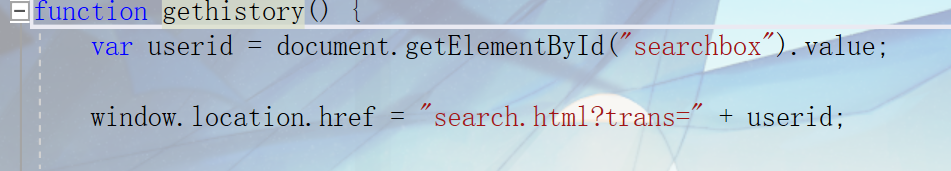


1. **根据id搜索对应帖子和评论**

①放入一个搜索栏searchbox，并放置两个按钮btn1,btn2



②点击两个按钮会跳转至展现本id所发帖子和评论的界面。

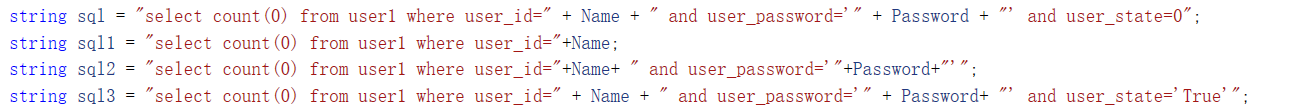


（同时传递参数id）

③跳转至对应界面，接收id。根据接收的id，与json对象内的id进行匹配，如果匹配成功则展示对应的标题。如果没有查找的，则返回“尚未发帖”。



**三、登录、注册与页面头部：**

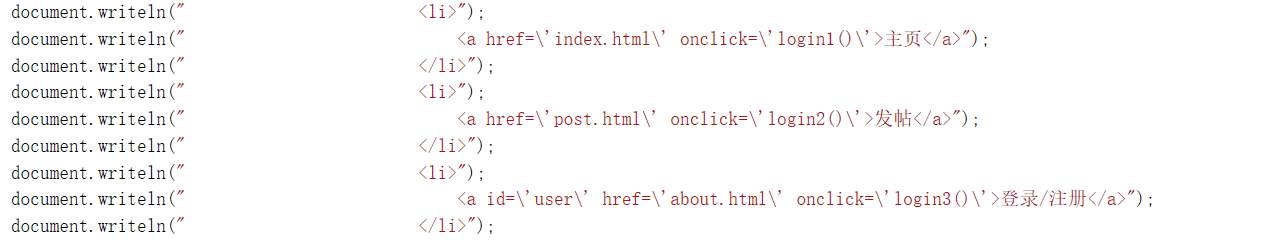
about.aspx.cs 

about.html

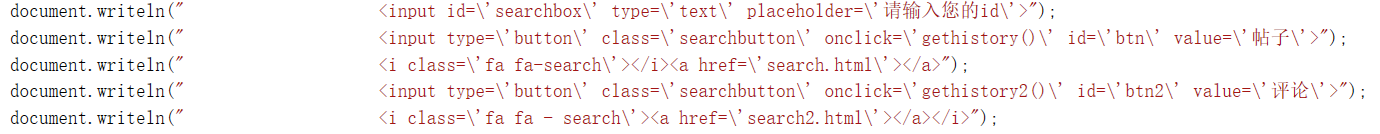


Header.js

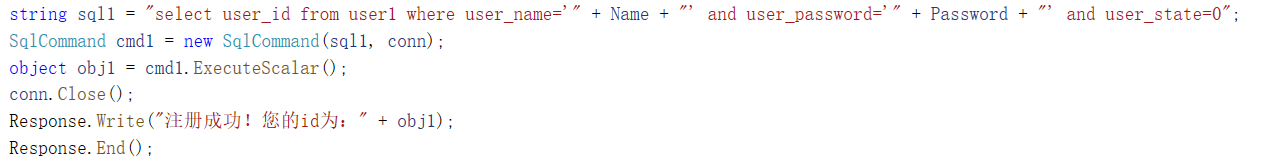
用js存储公共header，有三个按钮：主页，发帖和登录，以及一个搜索框，可以搜索输入的id的帖子和评论。



Footer.js



zhuce.aspx.cs

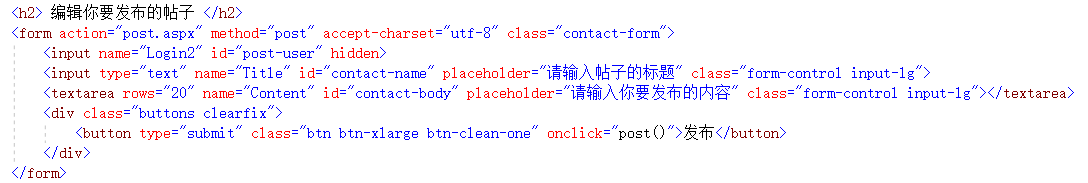


zhuce.html

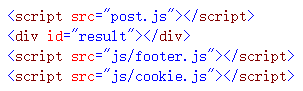


**四、发帖与评论：**

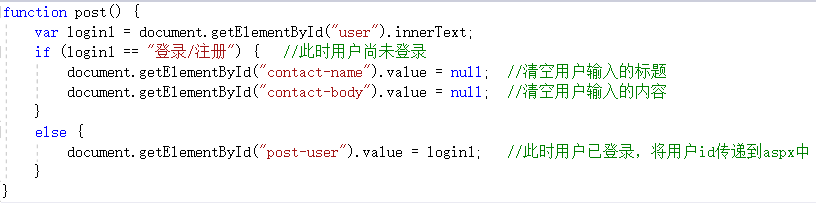
post.html，HTML中的发帖界面：



引入外部js文件：

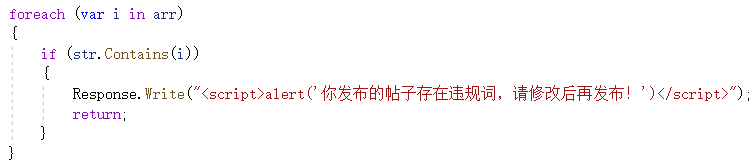


post.js

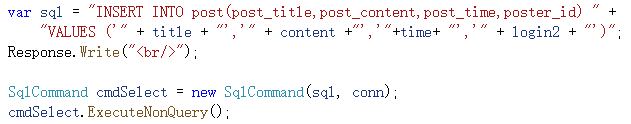


post.aspx.cs



**五、审核功能：**

将内容存入数据库：



用户已登录、审核通过，发布成功自动跳转回主页

IMG_256

single.html

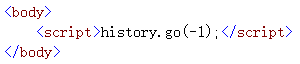
**六、发表评论界面：**



向后台发送请求：



comment.aspx ：返回评论页面



comment.aspx.cs：各行代码功能与发帖部分基本一致





SqlHelper.cs：将评论的内容发回到评论页面上



addcomment.js:显示用户是否登陆，若未登录评论内容为空



addcomment.aspx.cs

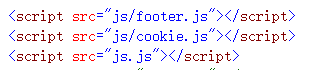
将最新发布的评论发布在原贴页面评论最下方，作为最新评论显示



**七、主页：**

index.html

外部的尾部文件、注册文件和登陆文件等





右侧显示栏：







# 6、运行效果图

* **展示新帖：**



* **搜帖子，并成功出现该用户发布过的帖子：**





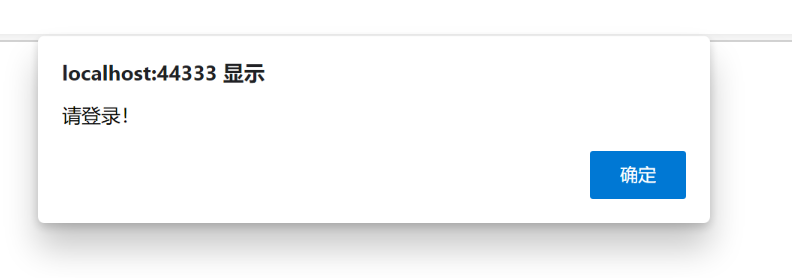
**搜评论并成功出现该用户发过的评论：**

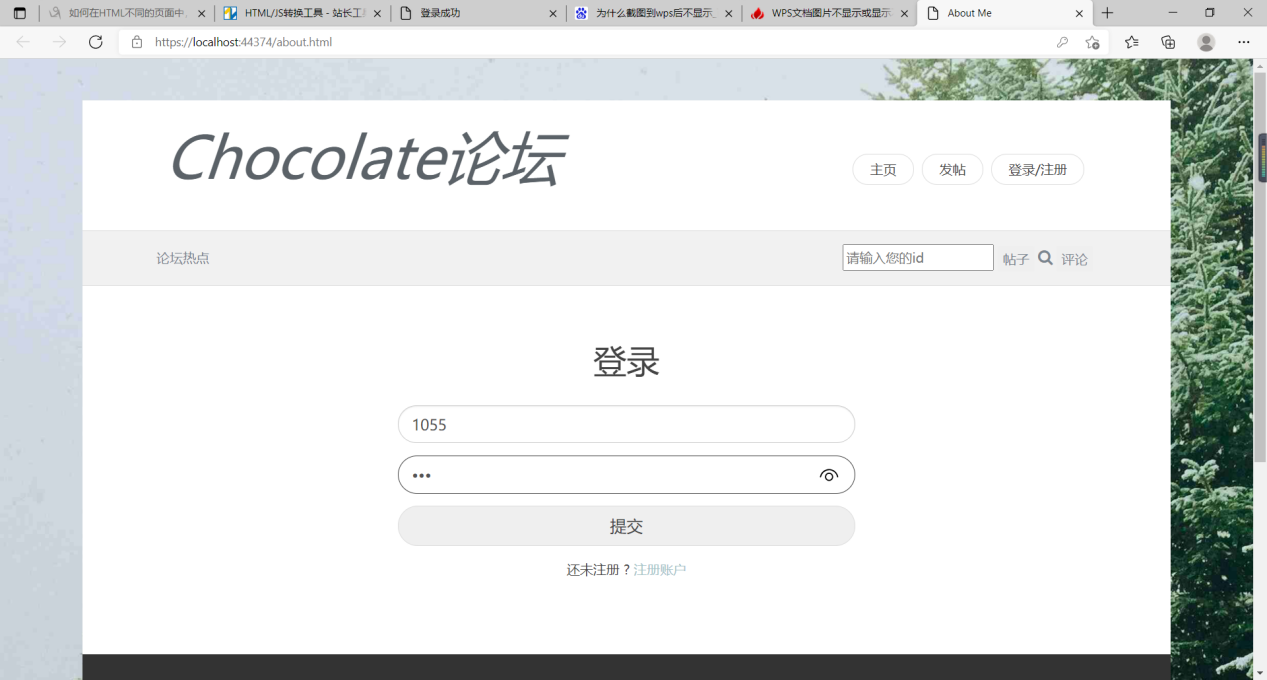

* **发帖：**



（如果未登录）



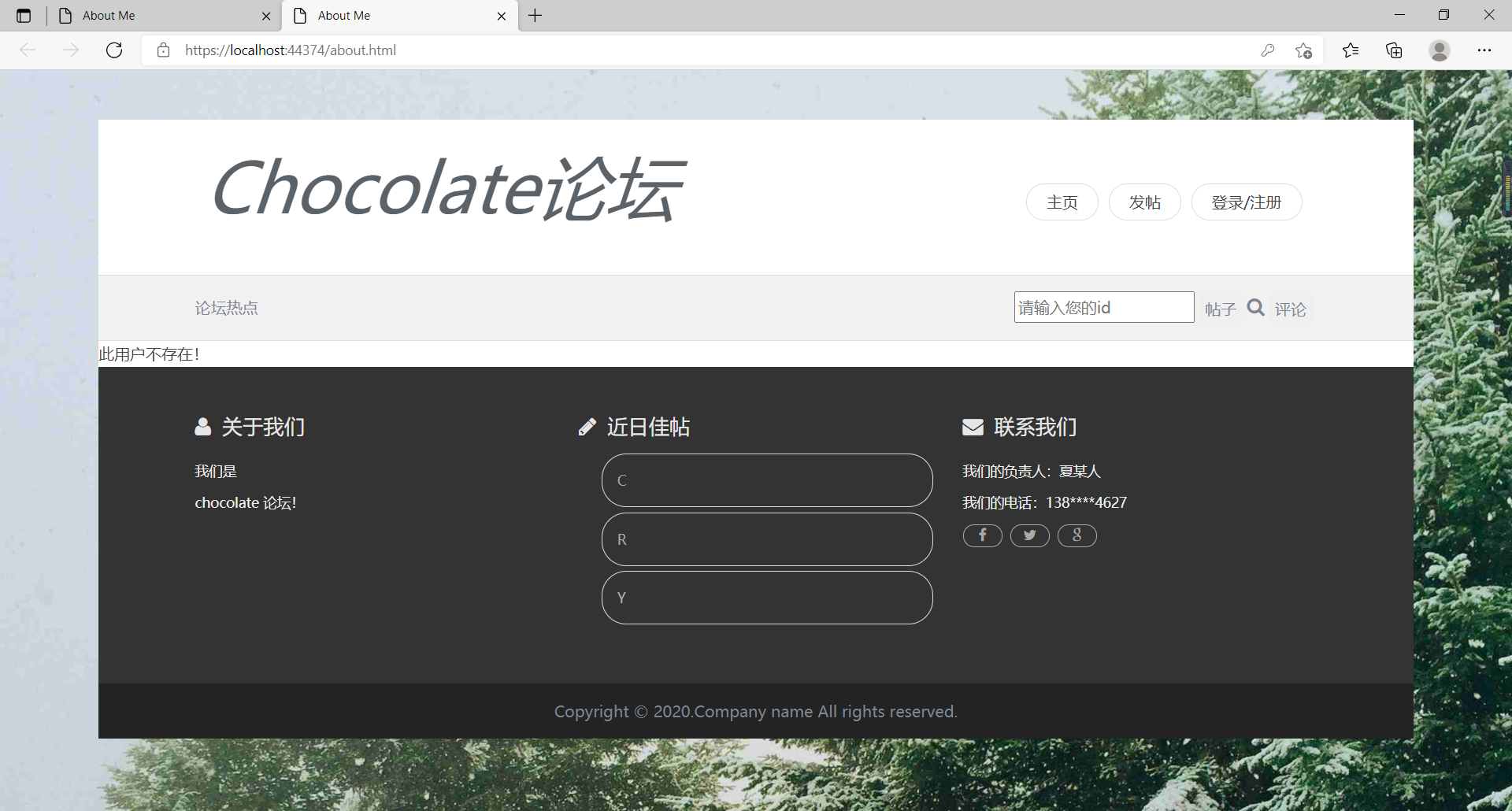
* **登录：**



成功的话



用户不存在的情况



密码错误



* **注册**





* **发表评论**



评论成功

