**上机3留言板制作**

**1.上机目的**

了解并掌握Node.js的基本开发方法。使用 HTML、CSS、JavaScript 作为前端，Node.js 和 Express 作为后端，使用MySQL等数据库管理系统，构建简单留言板Web应用。

**2.上机软件环境**

OS：Ubuntu22.04；

编辑软件、调试工具：VSCode，常用的浏览器等；

**3.上机内容**

（1）掌握Node.js的基本开发方法；

（2）连接MySQL数据库，实现用户注册与登录功能；

（3）根据留言板应用的需求，能够合理对前端表现层进行规划，即通过HTML中的元素、CSS等工具能够创建符合需求的页面；

（4）创建一个简易的留言板，至少包含留言、回复、展示留言的功能。

**4.上机要求**

实验独立完成，按时完成报告撰写，并排版规范。软件功能执行流畅，界面简洁、高效、大方。报告内容包括：上机实验目的、软件环境、上机要求、关键代码与步骤、上机小结。

**5.程序关键代码与界面**

技术栈使用：

前端：React18 + Antd Design（样式库）

后端：Express +MongoDB Atlas +cors（中间件，处理跨域）+express-session（中间件，保存会话信息）

源代码地址：<https://github.com/mxxxOrz12/hw1>

（1）登录注册页面

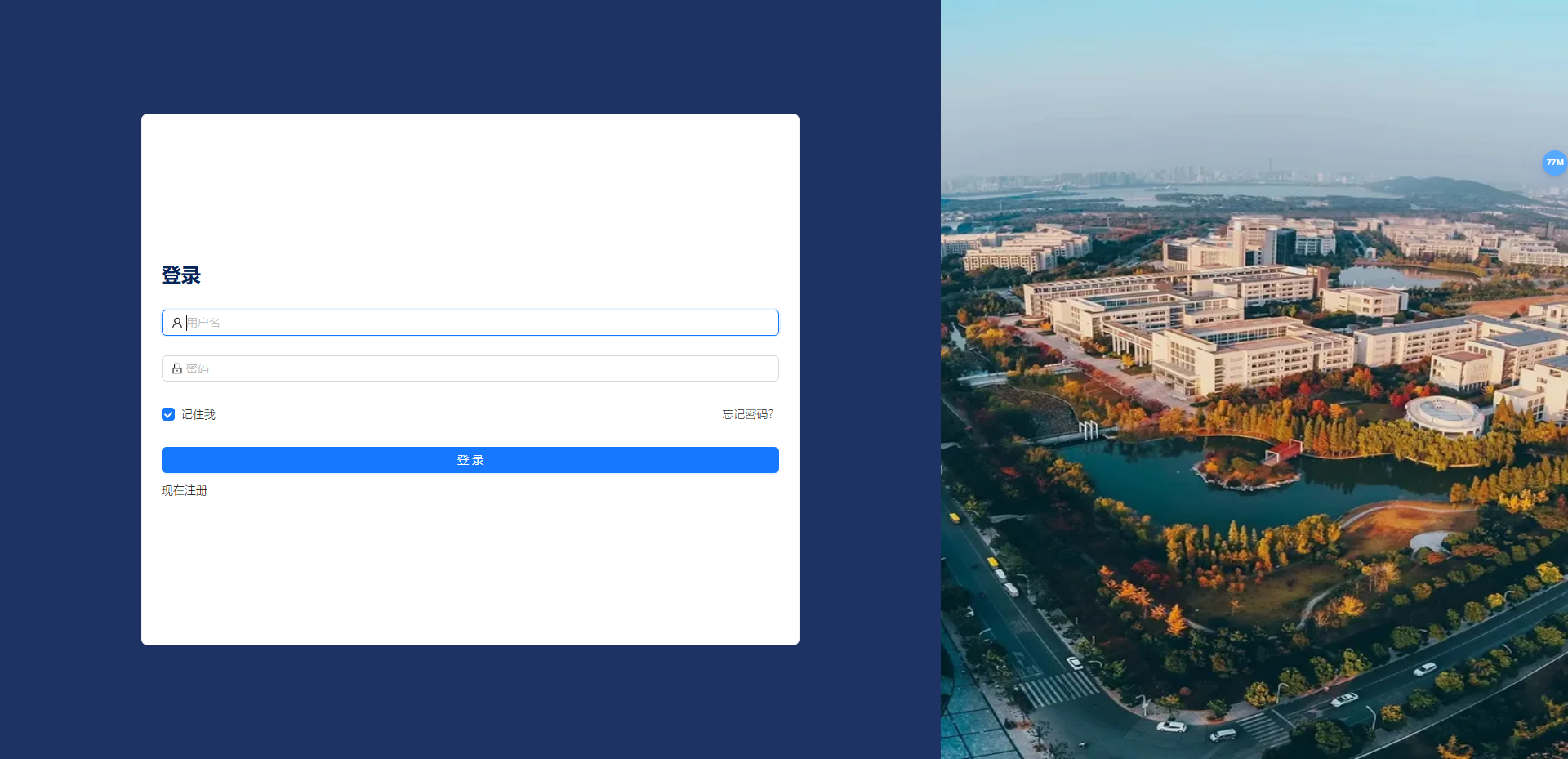


图1 登录页

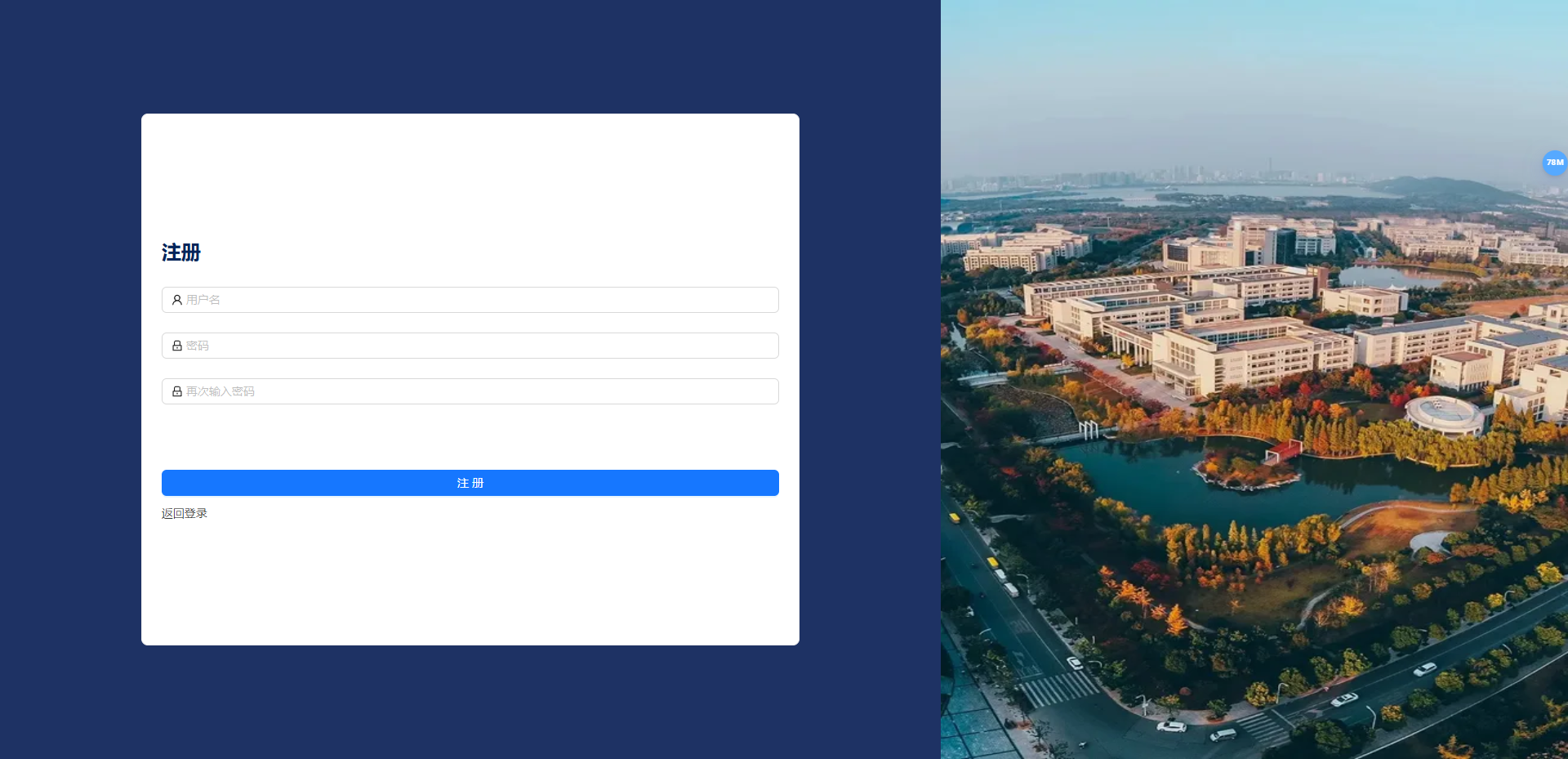


图2 注册页

我使用了React中的useState hooks定义了一个isSignup变量去判断是注册还是登录，并通过该变量动态渲染页面，意思是，我可以将登录和注册写为一个页面，通过左下方的现在注册/返回登录来切换，无需写两个页面。代码如下：



图3 登录注册页部分代码

为了实时的能够传送表单数据，我使用React的useState hooks定义一个form变量，并初始化为initialState，通过form表单中各个input标签中定义的onChange函数去实时改变，代码示例如下：

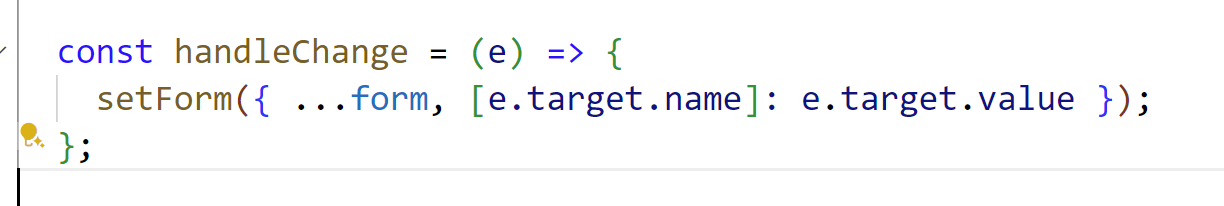


图4 handleChange函数

我通过这个句柄函数来改变form的值，其中用到了ES6的扩展运算符，event事件获取输入信息，并通过useState的setForm更新form的值。



图4 onSubmit函数

在Form表单中写一个onSubmit函数，并由一个按钮触发，完成前后端校验登录，我使用了axios，并通过isSignup变量来动态实现url的改变。



图5 express后端连接数据库

后端我使用了express和MongoDB，与mysql相比，MongoDB是文档类型数据库，更好存储Json格式数据。通过mongoose进行数据库连接。



图6 express中间件配置

我使用了express-session和cors跨域，用来接受前端发来的请求，session用来保存用户登录信息，还使用express.Router实现后端模块化。

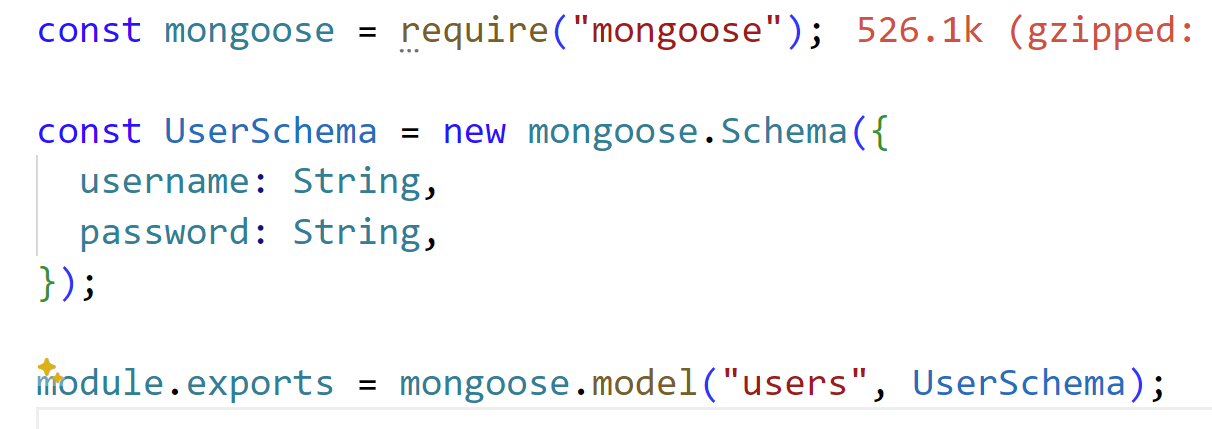


图7 创建用户Schema

通过mongoose创建用户Schema，类似于mysql创建表，而MongoDB更灵活。



图8 express配置登录注册接口

在注册接口中对输入的username进行判断，如果已经存在用户，返回状态码400，不存在则通过create创建用户。

在登录接口中对request.body进行解析，首先查找username对应的用户，如果没有则返回404，没有用户，找到用户进行密码比对，设置req.session.username，否则返回密码错误。



图9 创建回复Schema

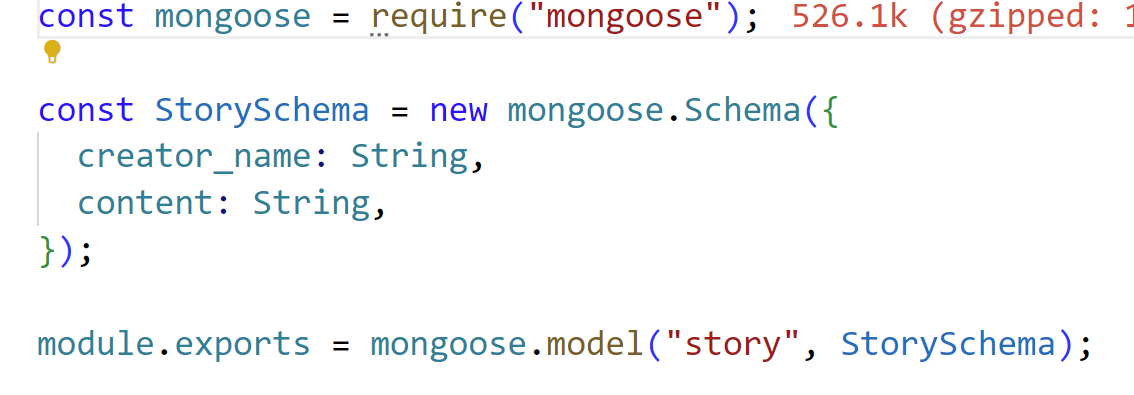


图10 创建评论Schema

创建回复Schema时，要保存它的Parent，即评论。



图11 express配置评论获取发布，回复获取发布接口

实现评论和回复的接口，这里creator\_name使用了之前登录时保存的req.session.username，保存了用户会话信息。

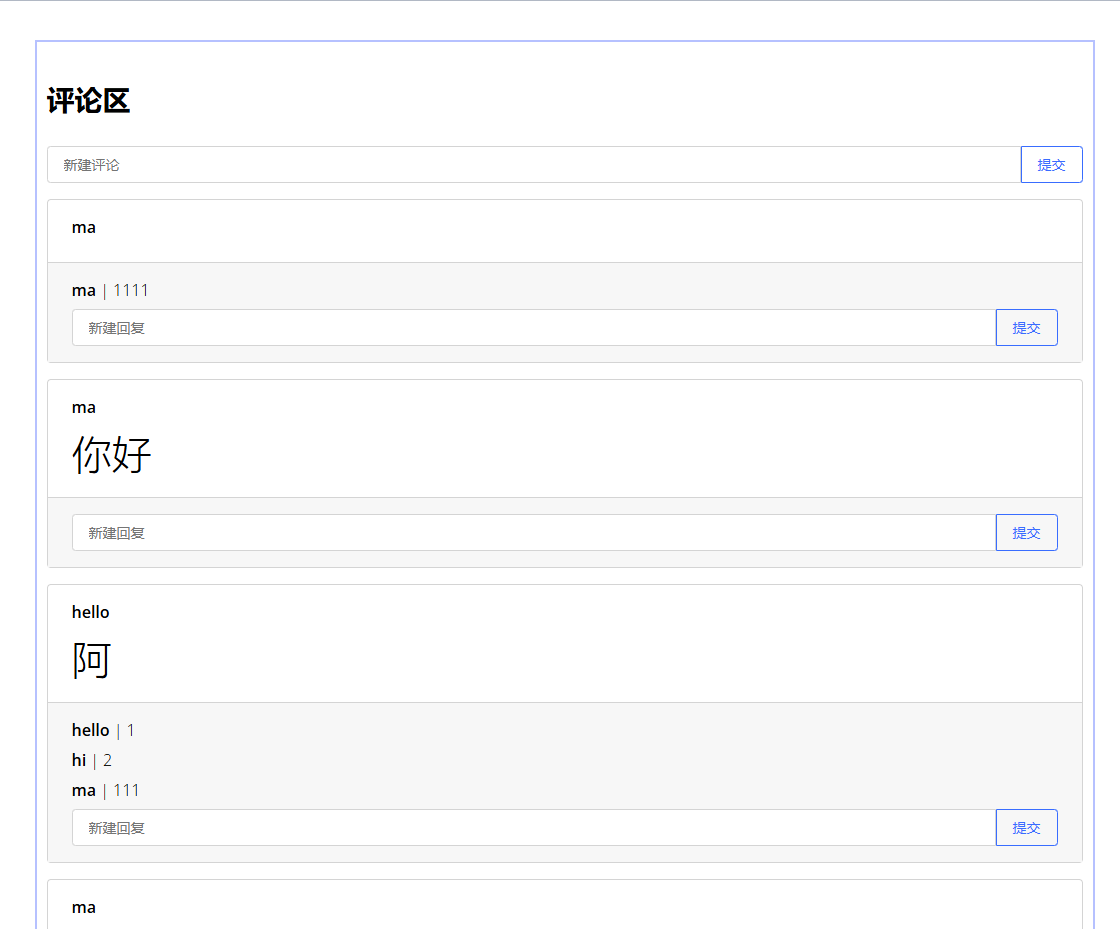


图12 评论回复界面

**6.小结**

通过使用React，MongoDB实现留言板，利用了多个Express中间件，通过此次上机对Web开发有了更深的认识，通过设置session来保存用户信息，开发基本的接口，前端使用Axios实现请求，加载后端、数据库中的资源，实现了完整的流程。其中，卡住我的是跨域session的设置，session一直配置不成功，原因是cors跨域中要指定端口，而不能使用默认所有端口，还需要加深自己的计算机网络的基础知识，如CSRF、XSS等。