1. **这个系统是如何实现的？（搭建过程）**

**前端：**

1. 通过vite初始化vue项目，分别初始化用户端和管理员端，安装所需库，创建目录
2. 配置env环境变量，并在vite.config.js中配置根据不同环境载入不同环境变量
3. Axios二次封装(http.js文件)，配置请求拦截器和响应拦截器
4. 配置store（所有组件和页面都能访问的数据和方法）
5. 组件封装、页面开发
6. 对接口，调试

**后端：**

1. 初始化Express项目，安装所需库，创建目录
2. 配置数据库连接文件
3. 配置日志中间件、文件上传中间件
4. 跨域配置、JWT配置
5. 接口开发
6. **后端为什么选择express框架？**

1.轻量级、开发速度快，相较于Spring框架

2.前后端语言一致，减小切换开发语言的心智负担，适合单人开发

3.国外十分流行，生态繁荣，相关的中间件丰富

……

1. **说一下项目结构？**
2. **前后端交互是如何进行的？**

前后端之间通过 HTTP 请求进行交互。前端发送 HTTP 请求到后端服务器，后端接收请求并返回相应的数据。

1. **视频的上传功能是如何实现的？（文件上传）**

图片、视频上传，或者其他文件上传，都是使用Multer库，封装成一个文件上传中间件（middleware中的imageMulter跟videoMulter），该中间件可以配置文件存放的目录，对上传的文件重命名。

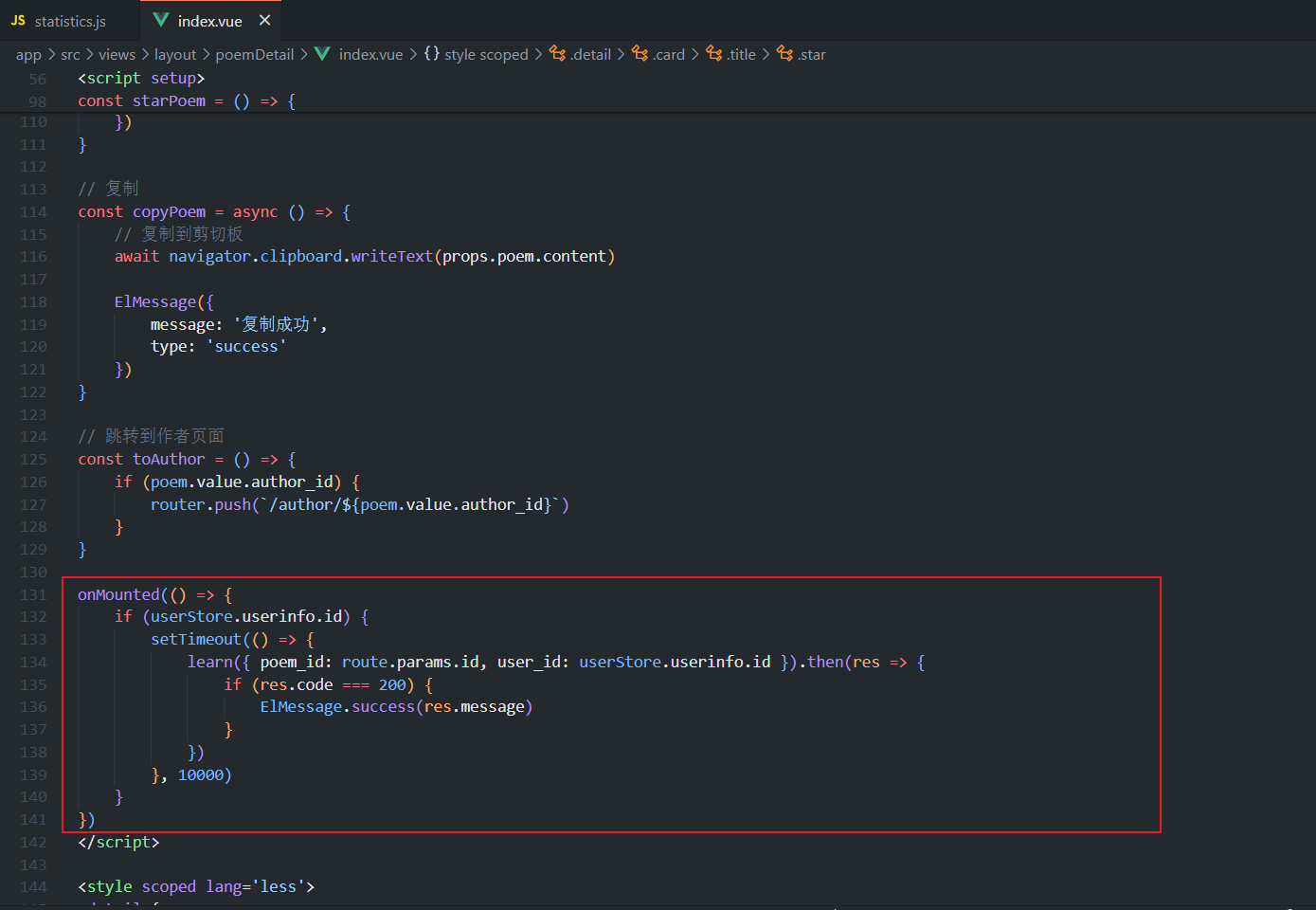
前端采用Element Plus的上传文件组件，将action设置成上传的接口地址，用户在选择文件之后自动触发上传。

如果不使用组件，也可以调用上传接口，手动将文件以form-data格式发送给后端，后端接收处理，同样的道理，只是组件帮我们封装好了。

1. **可视化图表的数据是如何处理的？涉及到的代码在哪里？判定诗词已学习的代码在哪里？**

可视化图表处理数据代码基本上在statistics.js中，还有些分布在poem.js、note.js中，基本上都是使用ES6的map方法处理数据。

判定诗词已学习是在用户进入一篇诗词详情页之后，开启一个定时器，10秒钟之后用户在仍然停留在此页面就调用已学习接口，学习记录会保存在数据库中，以保证不会重复计数。代码在poemDetail.vue



1. **模糊搜索是如何实现的？后台模糊搜索处理的代码在哪里？**

用户在输入框输入相关关键词后，点击查询发送ajax请求，后台执行相应的like模糊查询语句，执行select语句完成查询。

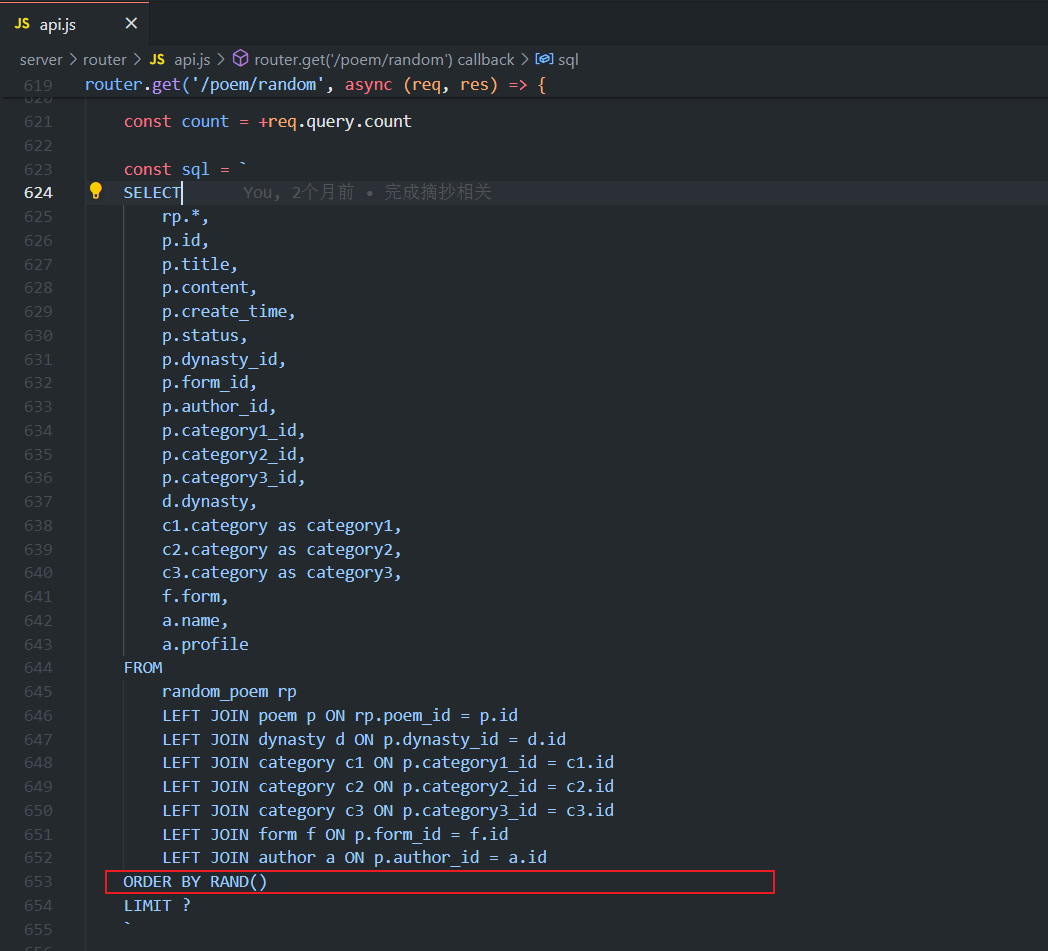
1. **分页功能是如何实现的？**

前端使用Element Plus的分页组件，将page、pageSize、换页方法等绑定对应变量和方法。

调用接口时，将page（页码）跟pageSize（条数）一起传递给后端，后端通过sql的limit语句获取指定数据，同时还需要获取总条数，一起发送给前端。

1. **摘抄的随机展示是如何实现的？代码在哪里？**

主要是通过sql语句实现的，通过RAND()函数和Limit获取指定条数的随机摘抄



1. **网站怎么实现安全性？**（权限登录时如何实现的）
2. jwt token：用户登录之后后端会返回一个token字符串（存储了用户信息加密而成），前端将token存在浏览器本地存储中，每次发请求都会将token携带在请求头中，后端以此识别用户身份。

2. 路由守卫：前端通过判断该用户是否登录，限制其访问某些页面，比如未登录的情况下，用户无法访问个人中心等页面。

11、**登录逻辑是如何实现的？**

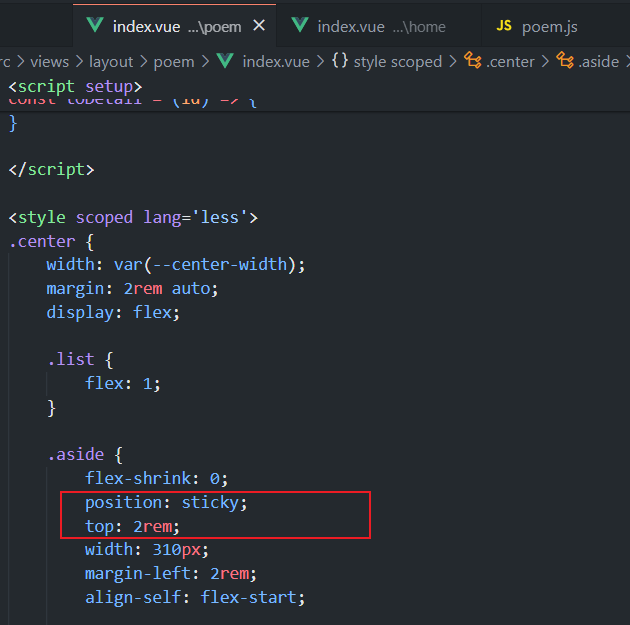
检查用户输入的账号和密码是否为空，如果为空，则显示错误提示消息。如果不为空则调用 login 函数，向后端发送登录请求，请求中包含用户输入的账号和密码。使用 await 等待登录请求的响应，并将响应结果保存在 res 中。根据响应的状态码进行相应的处理：

如果响应状态码为 200，表示登录成功，将返回的 token 存储在本地存储中，并更新用户信息和 token，显示成功提示消息，并返回上一个页面。

如果响应状态码不为 200，表示登录失败，显示错误提示消息。

1. **诗词页面的右侧侧边栏是如何实现整体滚动一会儿才滚动右侧侧边栏的？**

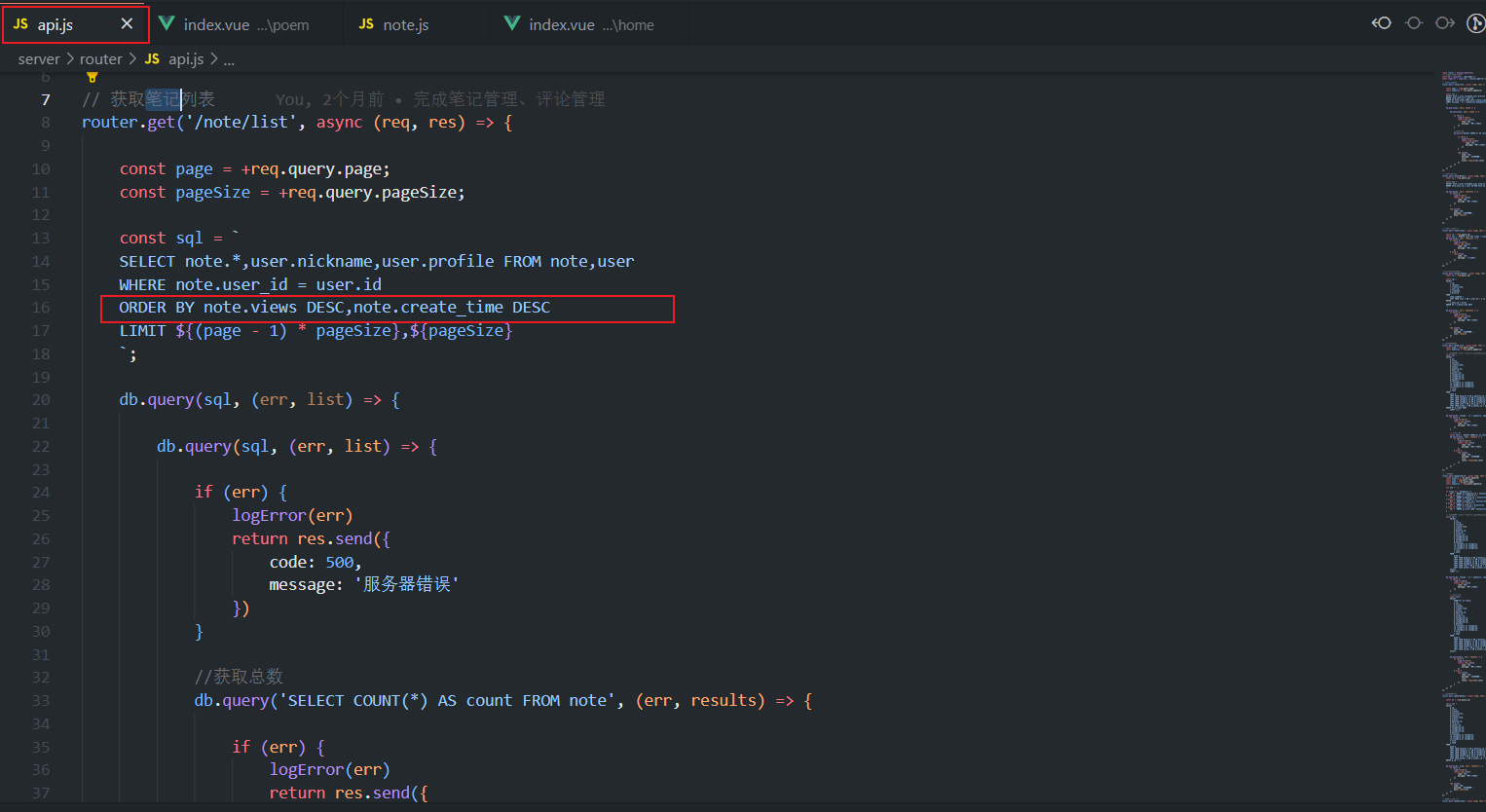
将该div的CSS设置成，position: sticky;



1. **社区页面的瀑布流布局是如何实现的？代码在哪里？根据笔记浏览量进行排序的代码在哪？**
2. 瀑布流使用组件库vue-masonry，具体可以参考：

<https://github.com/shershen08/vue-masonry>

1. SQL语句实现：通过·ORDER BY 语句，将 views、create\_time降序排序



**14、点赞功能是如何实现的？**

先判断用户是否已登录，如果用户已登录，则调用 like 函数，向后端发送点赞请求，请求中包含用户的ID和笔记的ID。如果响应状态码为200，则点赞成功，并将 isLiked 的值设为 true。如果不为200，则点赞失败。

**15、系统数据如何跟数据库进行交互？数据库之间的表是如何连接的？**

通过mysql库连接数据库，使用sql语句操作数据库。