**数据库pj-笔记管理系统**

# 需求分析

1.1**系统目的**

设计并实现一个笔记管理系统，该系统允许用户创建、编辑、保存和删除支持Markdown语法的笔记。系统需要提供一个简单易用的用户界面，并通过关系型数据库进行数据管理。

1.2**功能需求**

1.2.1**注册**：用户可以创建一个新账户。注册过程需要用户提供完整信息，包括用户名、密码、姓名、电子邮件地址。

1.2.2**登录**：用户需要使用用户名和密码登录系统。

1.2.3**笔记本管理**：

（1）查看笔记本：用户登录后，应能看到一个清单，列出所有他们创建的笔记本。

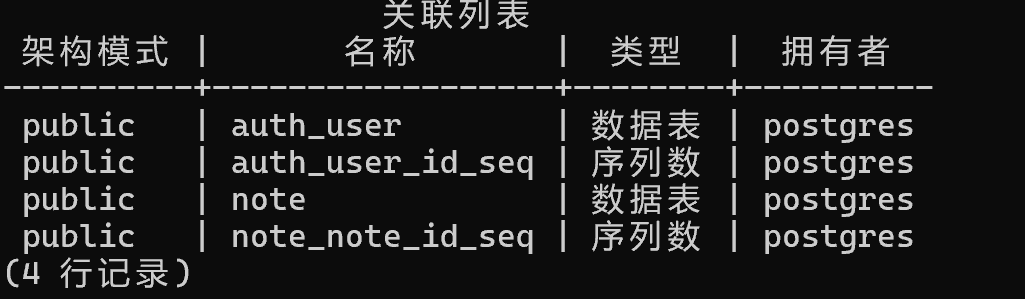
（2）新建笔记本：用户可以创建一个新的笔记本，为其命名，并开始添加内容。

（3）编辑笔记本：用户可以选择任何现有笔记本进行编辑。编辑器应支持Markdown语法，并能实时预览Markdown编译结果。编辑功能应支持快捷键Ctrl + S来保存笔记。

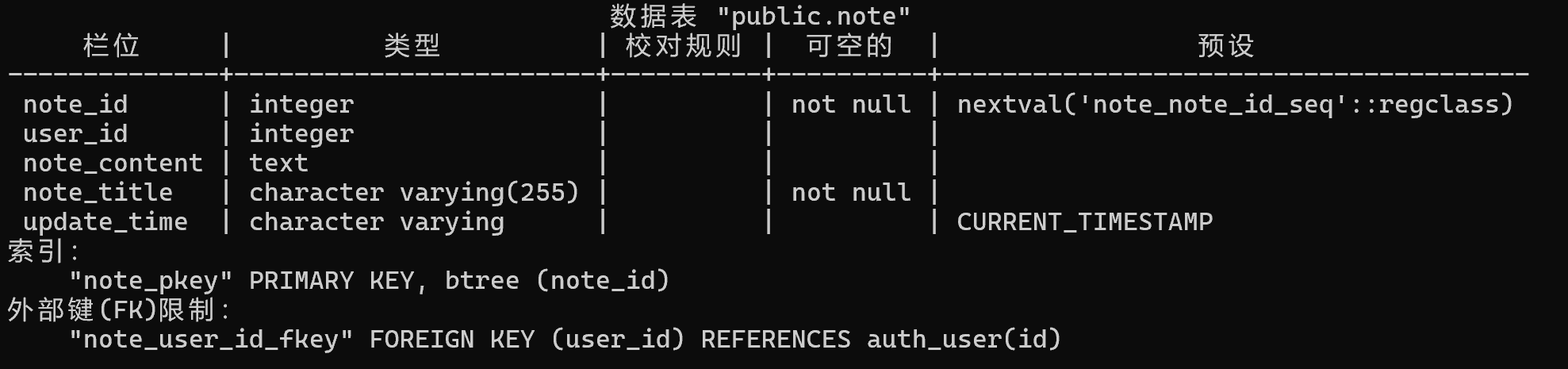
（4）删除笔记本：用户可以删除自己的任何笔记本。

# 数据库设计

2.1**数据库表**







具体说明如下：

（1）**auth\_user**表（用户信息表）

该表负责存储所有用户账户相关信息，支持用户的注册、登录。具体字段如下：

**id**：用户的唯一标识符，自增字段，用于唯一确定每一个用户。

**username**：用户的登录名，必须唯一。

**password**：用户的登录密码，存储加密后的密码。

**is\_superuser**：标识用户是否为超级用户。

**first\_name**：用户的名字。

**last\_name**：用户的姓氏。

**email**：用户的电子邮件地址。

**is\_active**：表示用户的账户是否处于激活状态。

索引和约束如下：

**主键约束**：**auth\_user\_pkey**，使用**id**字段。

**唯一性约束**：**auth\_user\_username\_key**，保证**username**字段的唯一性。

（2）**note**表（笔记信息表）

该表用于存储用户创建的笔记内容，每条笔记都关联到一个用户。具体字段如下：

**note\_id**：笔记的唯一标识符，自动生成。

**user\_id**：与**auth\_user**表的**id**字段相关联的外键，标识创建该笔记的用户。

**note\_content**：笔记的详细内容，存储用户的笔记正文。

**note\_title**：笔记的标题。

**update\_time**：记录笔记的最后更新时间。

索引和约束如下：

**主键约束**：**note\_pkey**，使用**note\_id**字段。

**外键约束**：**note\_user\_id\_fkey**，确保**note**表中的**user\_id**字段与**auth\_user**表的**id**字段正确关联

# 详细设计

3.1**前端**

使用vue3+ts+js的框架，使用MVVM的架构，信息用pinia进行管理，与后端的交互函数使用axios进行封装。

3.2**核心功能**

（1）用户注册与登录

注册：前端会验证提供的信息，后端将用户信息存储在auth\_user表中。

登录：后端通过验证auth\_user表中的用户名和密码实现用户认证。

（2）笔记管理

创建笔记：用户可以创建新的笔记，系统会将笔记信息存储到note表中。

编辑笔记：用户选择特定笔记进行编辑，系统通过note\_id检索并更新note表的note\_content中。

查看笔记：用户登录后，系统通过user\_name检索note表（note表中没有user\_name，需要连接auth\_user表），显示所有相关的笔记。

删除笔记：用户可以删除任何他们创建的笔记，系统将从note表中移除相关记录。

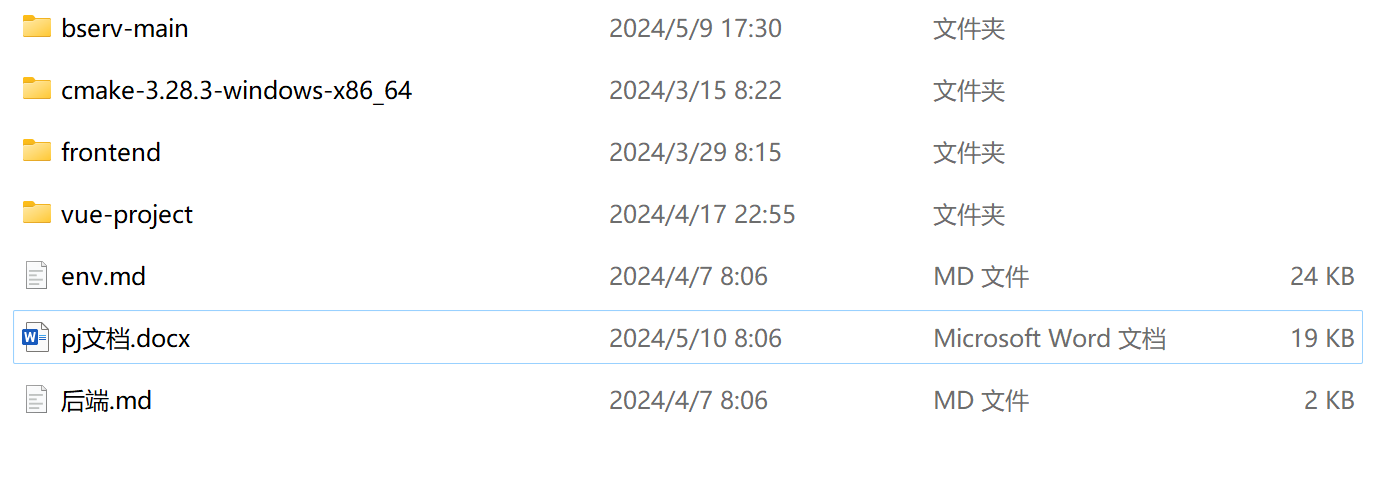
3.3**具体流程**

前端：写好前端页面与前端组件，封装好与后端交互的函数，大部分功能都会由模块中的@click函数进行触发。在该函数中调用异步的后端访问请求，并同步等待。等到收到后端的成功的返回之后，再更新pinia中的store，并触发其他效果。

后端：在webapp.cpp中注册好函数，在handlers.h和handler.cpp中写好函数，模仿模板函数。

debug：前端可以通过F12检查（console.log）的方法查看相关变量。后端可以通过lginfo、lgdebug的方法在控制台看到相关变量

# 部署方法

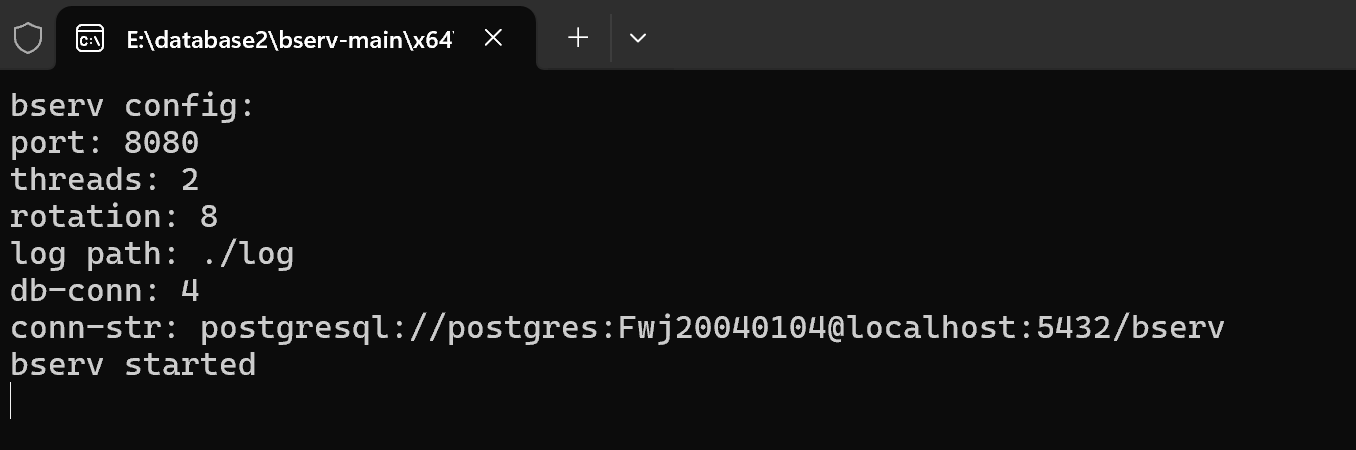
后端：

在该目录下点击进入bserv-main，找到WebApp.sln，

进入后点击左上方windows本地调试或者执行。



看到控制台有如下出现即成功。



前端：

点击进入vue-project，里面还有一个vue-project，用vscode打开里面这个vue-project。新建终端。先npm install 下载依赖。然后npm run dev。在浏览器输入<http://localhost:5173/>即可看到。