# 基于文心大模型的智能编辑器

**目 录**

[一、产品功能说明 1](#_Toc18013)

[1.1 AI辅助编辑 1](#_Toc25933)

[1.2多媒体信息提取 1](#_Toc17235)

[1..3格式排版 2](#_Toc28018)

[二、功能架构 3](#_Toc1141)

[三、 创新点 6](#_Toc6522)

[四、技术流程 8](#_Toc15792)

[4.1环境准备 8](#_Toc4842)

[4.2 数据库和OCR识别 10](#_Toc19100)

[4.3 编辑器页面与大模型 11](#_Toc18070)

[五、模型ID号 12](#_Toc19948)

## 一、产品功能说明

本产品已经实现三大基础功能且机融合大模型与小模型技术，支持高交互性、美观性、易用性、智能化和多协同：

1.1 AI辅助编辑

利用文心大语言模型，根据上下文，实现了润色、续写、翻译、关键词自动提取、自动生成摘要等常见写作辅助功能。用户可以根据需要对选中的文字进行操作。

· **高智能性**：通过大语言模型理解用户输入的上下文，并给出合适的输出结果，满足用户的需要。

· **高交互性**：用户可以在编辑器中选择文本并调用智能辅助功能，实时看到修改建议。

· **应用场景**：适用于学术写作、商业报告、文案编辑等多个场景，提升写作效率和质量。

**1.2多媒体信息提取及信息真实性判断**

利用OCR多模态感知技术，实现对**图像文字信息**的在线信息提取，弹窗显示图片文字内容和信息真假性判断，并支持复制，允许用户将内容有机整合到编辑器中。



图 1.1 OCR测试结果

· **高效识别**：将OCR功能封装为API，使用PaddleOCR技术对上传的图像进行文字识别，并实时返回识别结果。

· **无缝集成**：识别出的文字可以直接插入到编辑器中的任意位置，支持多格式文本处理。

· **应用场景**：适用于扫描文档的数字化、图片文字的提取和编辑、数据录入等场景。

**1..3格式排版**

编辑器具备对文本框中输入的文字进行加粗、倾斜、高亮、下划线、删除、返回上一步、清除格式以及创建各种序号列表、生成大纲、添加代码块等基础排版功能。

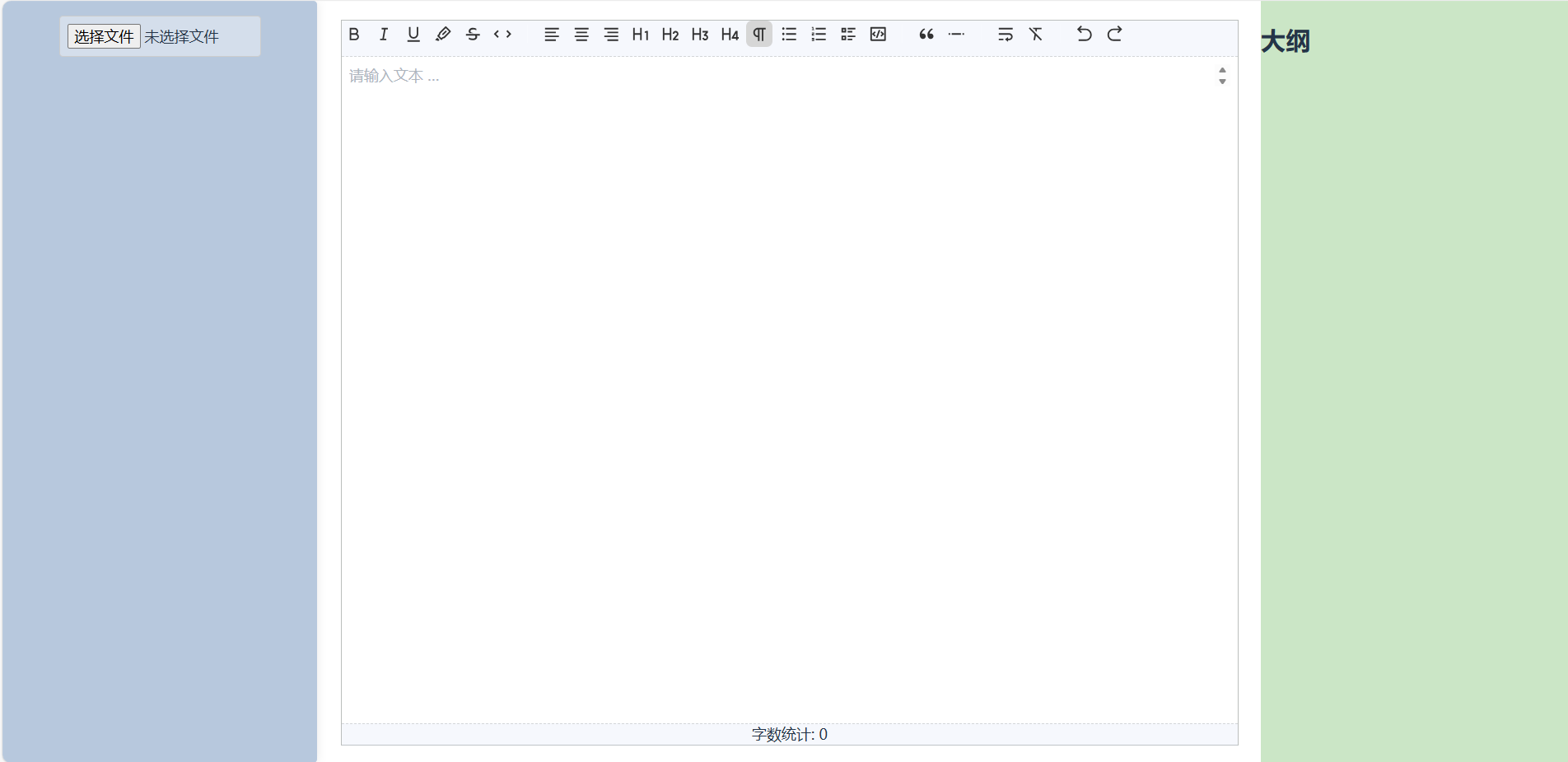
· **基本格式化功能**：加粗、倾斜、高亮、下划线、删除线、清除格式等。

· **高级排版功能**：创建各种序号列表、生成大纲、添加代码块、引用块等。

· **撤销与恢复**：支持撤销和返回上一步操作，确保编辑的灵活性和安全性。



图 1.1 首页



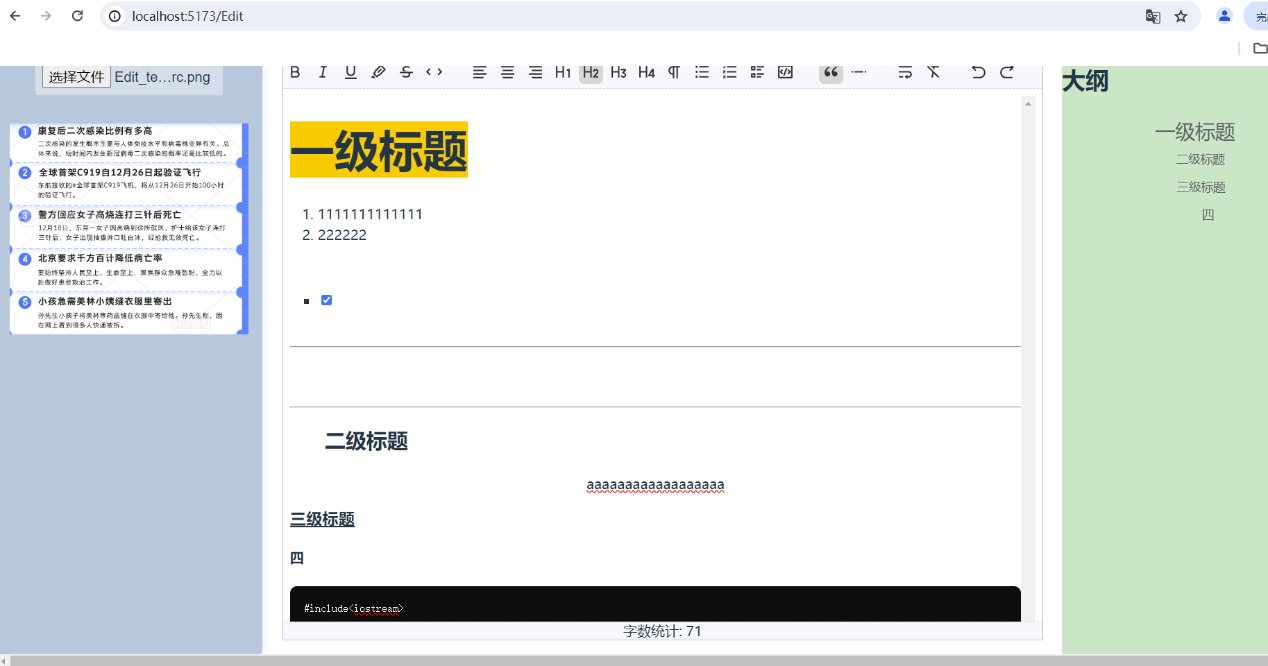


图 1.2 编辑器页面

## 二、功能架构

架构整合了前端和后端技术，结合了现代开发工具和库，同时集成了OCR和AI技术，提供了一个高效、智能且用户友好的编辑器平台。总体编辑器框架的功能架构可以概括为以下几个关键方面：

**1. 前端**

使用Vue.js和Vite构建现代前端项目，确保开发效率和应用性能。

利用Vue Router进行前端路由管理，Pinia进行状态管理，实现应用的模块化和状态控制。

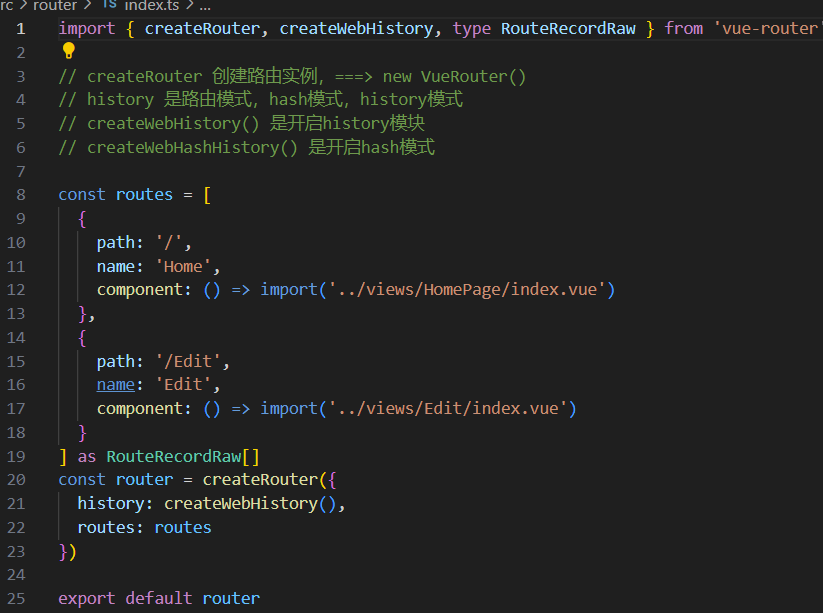


图 2.1 router相关代码

**2. UI和样式处理**

采用Element Plus作为UI库，提供丰富的组件和界面元素。

使用Sass作为CSS预处理器，提高样式的编写效率和可维护性。

利用Remix Icon作为图标库，增强界面的视觉效果。

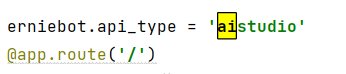
**3. 网络请求和数据交互**

使用axios进行网络请求，确保前后端数据交互的高效和稳定。

**4. 后端技术栈稳定性和扩展性**

使用Python的Flask框架开发后端，提供稳定可靠的API服务。





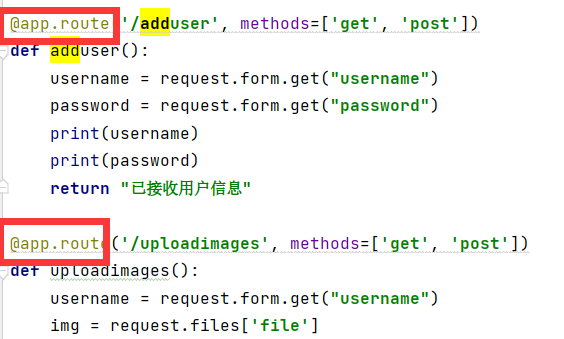


图 2.2 后端部分接口

配置PyMySQL进行MySQL数据库操作，确保数据存储的稳定性和安全性。

配置CORS以解决跨域问题，确保前后端通信顺畅。

**5. OCR功能智能化**

封装OCR功能为API，结合OpenCV进行图像处理，使用PaddleOCR进行文字识别。

配置PaddlePaddle GPU环境，提升OCR识别的速度和准确性。

利用Aistudio平台进行OCR模型的部署，提高模型的实用性和易用性。

1. **数据库与数据管理**

使用MySQL数据库存储关键数据，如用户信息和图片路径，保证数据的持久化。

使用phpMyAdmin作为数据库管理工具，提供数据的可视化操作和管理。

**7. AI功能集成**

引入如erniebot等AI服务，提供文本润色、续写、翻译、关键词提取等等接口。

配置后端接口实现AI服务调用，前端提供用户交互界面，提升用户体验。

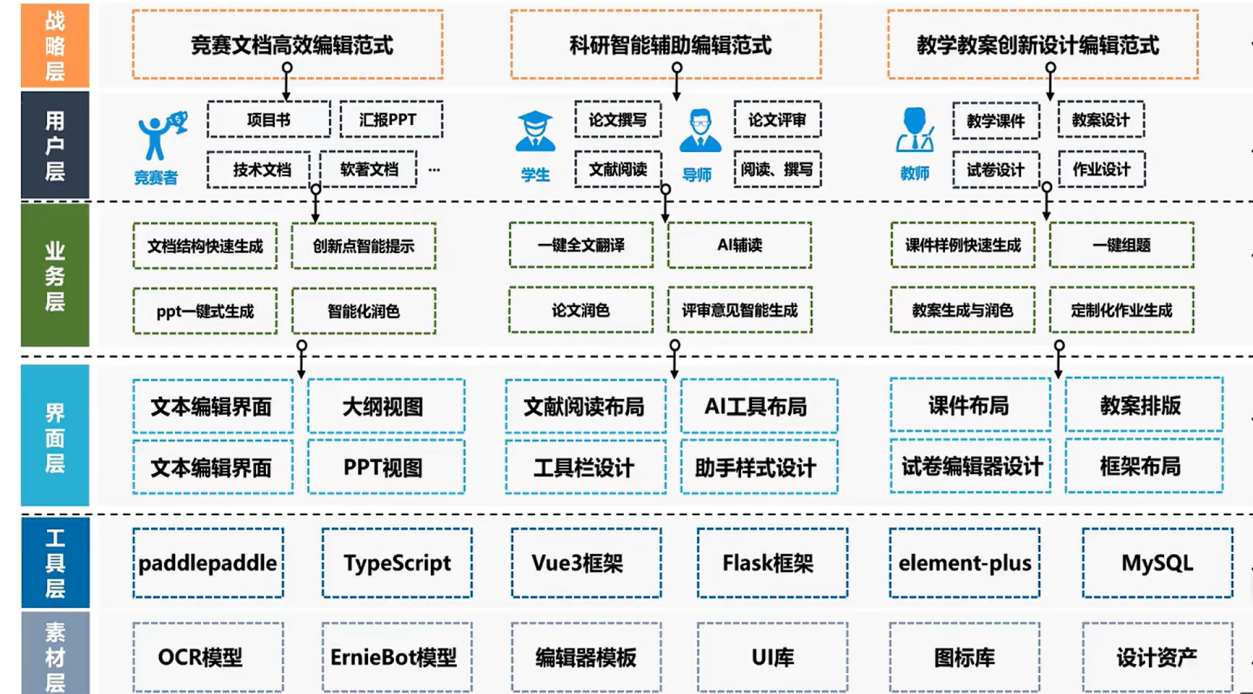


图 2.3 系统架构图

编辑器的产品框架是一个综合性的开发框架，涉及前端界面构建、后端服务开发、数据库管理、AI功能集成、环境配置、项目部署等关键环节，形成了从概念设计到实际应用的完整开发和运行体系。

界面层依赖工具层提供的技术和框架来实现其功能：

1. 文本编辑、大纲视图、PPT视图等功能用到TypeScript和Vue3框架来实现动态交互。
2. 后端功能如文本处理、文档生成等用到Flask框架和MySQL数据库。
3. element plus为界面提供UI组件，提升用户体验。
4. PaddlePaddle和ErnieBot模型提供AI能力支持，例如润色、续写、文本翻译、智能提示、AI辅助阅读（部分暂未开发）等。

工具层中的技术和框架需要依赖素材层提供的基础素材和模型：

1. OCR模型用于文本识别，支持文档处理功能。
2. 编辑器模板、UI库、图标库和设计资产为界面层的开发提供基础资源，确保界面美观且功能丰富。

## 创新点

**1. 全栈开发与集成**

（1）前端技术栈的现代化

（2）使用Vue.js、Vite构建现代前端项目。

（3） 配置Vue Router进行路由管理，使用Pinia进行状态管理。

（4） 利用axios进行网络请求，使用Element Plus作为UI库，Sass作为CSS预处理器，以及Remix Icon作为图标库。

**2.后端技术栈的稳定性和扩展性**

（1）使用Python的Flask框架进行后端开发，提供可靠的API服务。

（2）配置PyMySQL进行MySQL数据库操作，确保数据存储的稳定性。

（3）配置CORS解决跨域问题，确保前后端的顺畅通信。

**3. OCR功能的智能化**

（1）封装成API，进行高效的图像处理和识别

（2）前端使用Element Plus的Upload组件实现用户友好的图片上传功能；后端结合OpenCV进行图像处理，使用PaddleOCR进行高效的文字识别。

网络谣言很多，于是作品同时具有谣言检测功能，基于设定的敏感词分析和cnsenti库的情感分析，能够为用户上传的图片文字信息作出初步的真假性判断。



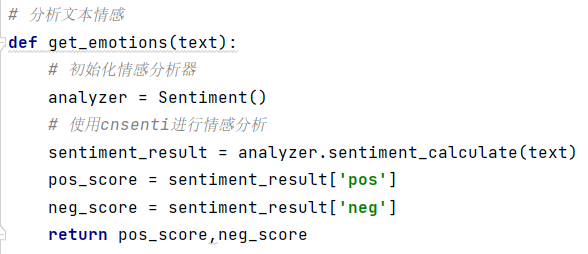


图 3.1 后端谣言检测相关代码

（3）配置PaddlePaddle GPU环境，充分利用显卡加速，提升OCR识别的速度和准确性。

（4）使用Aistudio平台的模型产线功能进行OCR模型的部署，提高模型的实用性和易用性。

4.数据库与数据管理

（1）完善的数据库集成

使用MySQL数据库存储用户信息、图片路径等数据，确保数据的持久化和安全性；使用phpMyAdmin作为数据库管理工具，提供便捷的数据可视化操作和管理。

5.编辑器的高效与智能

（1）采用Tiptap富文本编辑器，基于ProseMirror构建，提供灵活可扩展的编辑功能；使用插件系统定制编辑器功能和样式，满足不同用户的需求。

（2）设计网格布局，提供直观的视觉区分和便捷的操作体验。

6.AI功能集成

（1）引入erniebot等AI服务，提供智能编辑功能的接口。

（2）配置后端接口，实现AI服务的调用，并在前端提供用户交互界面，提升用户体验。

## 四、技术流程

### 4.1环境准备

1.Node.js的安装配置

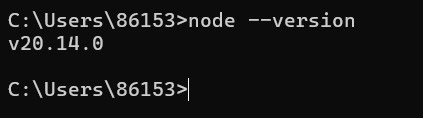


图 4.1 Node版本

2. Visual Studio Code (VSCode) 的安装和设置，并取消Typescript语法校验

3.前端项目构建

使用Vue.js作为前端框架，通过Vite创建新项目。

配置Vue项目，包括Vue Router进行路由管理，Pinia作为状态管理库，axios进行网络请求，element plus作为UI库，Sass作为CSS预处理器，以及Remix Icon作为图标库。

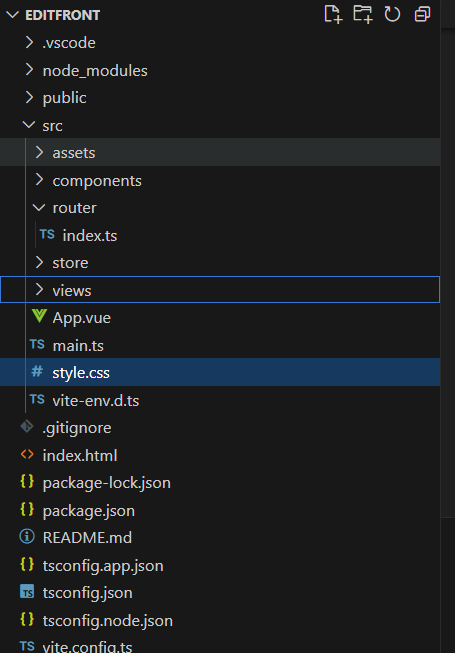


图 4.2 前端框架

4.后端项目搭建

使用Python的Flask框架进行后端开发。

配置数据库连接，使用PyMySQL进行MySQL数据库操作。

5.前后端交互

通过axios在前端发送请求到后端，使用Flask框架接收并处理数据。

6. 数据库环境准备

使用phpStudy进行数据库环境的搭建，包括MySQL服务器和phpMyAdmin的安装配置。

1. PaddlePaddle GPU环境搭建

为了支持AI功能，进行PaddlePaddle GPU版本的安装配置，包括显卡驱动、CUDA和cuDNN的安装，以及PaddlePaddle GPU的安装和测试。

9. 项目配置和优化

包括Vite和webpack的配置，路径别名的设置，以及tsconfig.json的配置。

### 4.2 数据库和OCR识别

1. 数据库集成

集成MySQL数据库，用于存储用户信息、图片路径等数据。

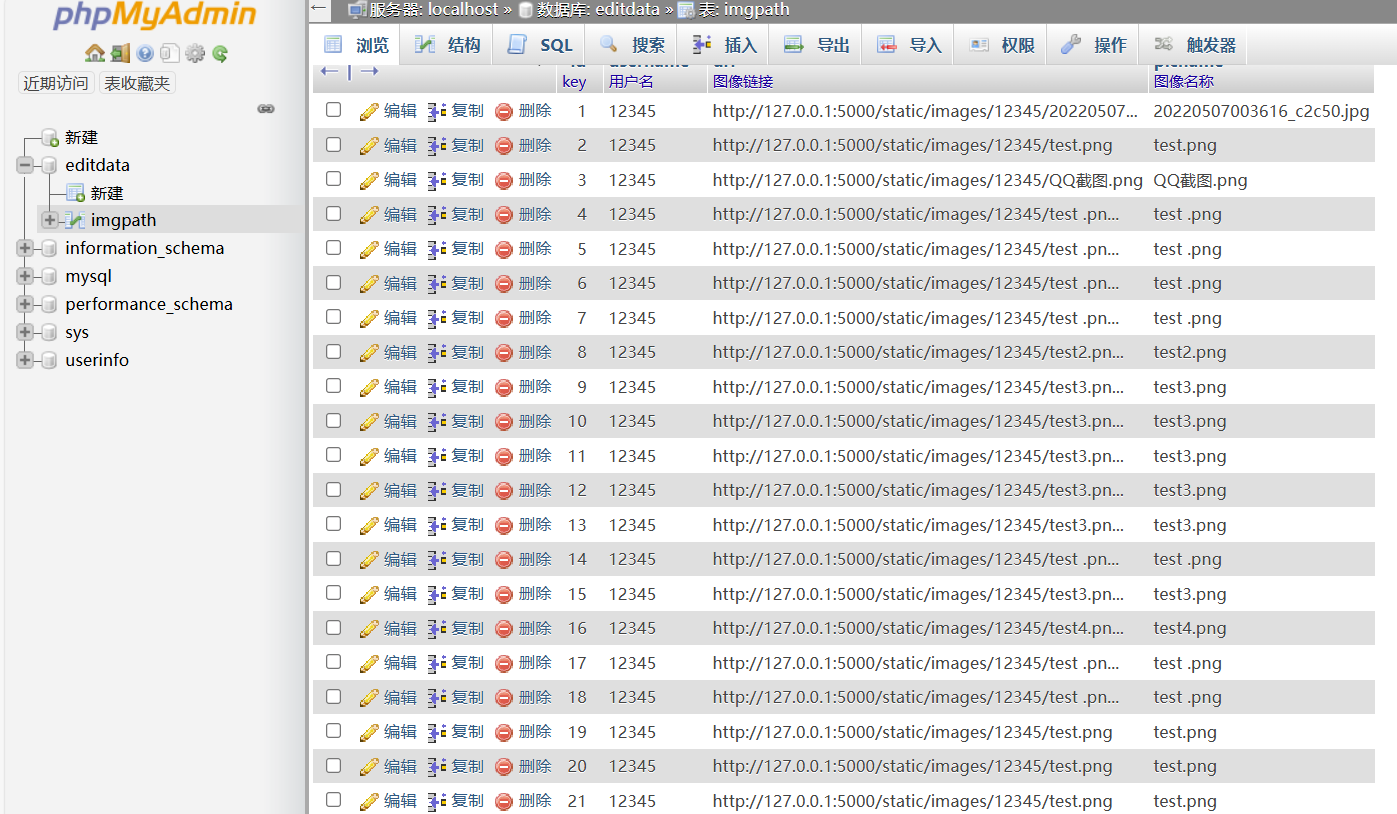


图 4.3数据库信息

使用phpMyAdmin作为数据库管理工具，方便数据的可视化操作和管理。

2.后端与数据库的连接

使用PyMySQL作为Python操作MySQL数据库的驱动程序。

3.OCR识别功能 （引入小模型）

**具体步骤**

（1）图片上传和处理

在前端使用Element Plus的Upload组件实现图片上传功能，uploadimg函数用于将选中的图片通过FormData发送到后端服务（/uploadimages端点），并处理响应结果。

在后端使用Flask框架接收上传的图片，并利用OpenCV库进行图像处理和保存。

（2）PaddlePaddle与OCR项目

使用PaddlePaddle框架结合OCR技术实现文字识别。

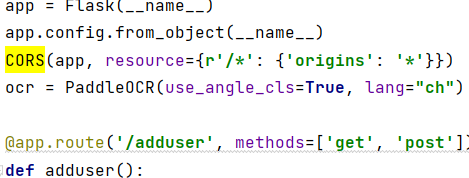
进行PaddleOCR环境的安装、OCR模型的本地部署和使用，以及将OCR模型封装为API接口，上传成功后，将后端返回的识别结果和真假性判断展示给用户，并显示结果弹窗。

（3）产线环境配置部署

使用Aistudio平台的模型产线功能进行OCR模型的部署。

（4）接口安全和性能

使用Flask Cors解决跨域问题，允许所有来源的跨域请求。



使用Axios进行前端与后端的通信，处理请求和响应。

### 4.3 编辑器页面与大模型

**1.编辑器核心组件**

采用Tiptap富文本编辑器，基于ProseMirror构建，提供灵活可扩展的编辑功能。包含Core、Extensions、Commands、Schema、Vue/React等组件，支持通过插件系统定制编辑器功能和样式。

**2.编辑器安装与配置**

使用npm或pnpm安装Tiptap及其相关插件，如StarterKit、扩展模块等。

配置Vue组件，初始化Tiptap编辑器实例，并在组件挂载后创建以及在组件卸载前销毁。

**3.用户界面布局**

设计采用网格布局，划分为左侧工具栏、编辑区、右侧工具栏，以及底部字数统计区域。使用CSS进行样式设计，为不同模块提供直观的视觉区分。

**4.基本工具栏**

顶部设置功能菜单，包含基本样式和图标，使用MenuItem组件构建工具栏按钮。工具栏提供编辑功能，如加粗、斜体、列表、链接等，并与编辑器实例交互。

编辑器左侧可根据用户喜好上传图片，如日志清单、编辑任务、写作思路等等，可为编辑提供参考，也可上传装饰性的图片增加页面的美观性。

**5.字数统计与大纲**

实现编辑器底部的字数统计功能，动态显示当前文本的字符数。

开发大纲功能，识别标题并生成文档结构视图。

6. AI集成

引入erniebot等AI服务，提供文本润色、续写、翻译等等接口。配置后端接口，实现AI服务的调用，并在前端提供用户交互界面。

@app.route('/getpolish', methods=["GET", "POST"])

@app.route('/getcontinuation', methods=["GET", "POST"])

@app.route('/gettranslation', methods=["GET", "POST"])

@app.route('/getKeyword\_extraction', methods=["GET", "POST"])

@app.route('/getsummary', methods=["GET", "POST"])

## 五、模型ID号

使用飞桨的模型产线创建OCR项目并进行本地部署

模型ID号为：p -3ef86b453271

