目录

[一、工作需求分析 2](#_Toc154774713)

[1. 用户管理模块： 2](#_Toc154774714)

[1.1 注册与登录 2](#_Toc154774715)

[1.2 用户权限 2](#_Toc154774716)

[2. 题目录入模块： 2](#_Toc154774717)

[2.1 直接输入题目 2](#_Toc154774718)

[2.2 模板导入 2](#_Toc154774719)

[3. 题目浏览与管理模块： 3](#_Toc154774720)

[3.1 题目分类与标签 3](#_Toc154774721)

[3.2 浏览题目 3](#_Toc154774722)

[3.3 题目详细信息 3](#_Toc154774723)

[4. 输出模块： 3](#_Toc154774724)

[4.1 文件导出 3](#_Toc154774725)

[5. 多媒体题目支持模块： 3](#_Toc154774726)

[5.1 语音题目 3](#_Toc154774727)

[5.2 图形与图像题目 3](#_Toc154774728)

[6. 系统管理模块： 4](#_Toc154774729)

[6.1 数据库管理 4](#_Toc154774730)

[二、系统架构 4](#_Toc154774731)

[1.前端架构 4](#_Toc154774732)

[2. 后端架构 4](#_Toc154774733)

[3. 数据库 5](#_Toc154774734)

[三、功能实现解析 5](#_Toc154774735)

[3.1 图形界面 5](#_Toc154774736)

[3.2 题目输入 7](#_Toc154774737)

[3.2.1 直接输入 7](#_Toc154774738)

[3.2.2 模板导入 8](#_Toc154774739)

[3.3 题目存储 9](#_Toc154774740)

[3.4 题目浏览 9](#_Toc154774741)

[3.5 选择题目并输出 11](#_Toc154774742)

[3.6删除题目 14](#_Toc154774743)

[3.7 多种题目形式 14](#_Toc154774744)

[3.8 安全性 15](#_Toc154774745)

[3.9 性能优化 16](#_Toc154774746)

[四、小结及回顾 16](#_Toc154774747)

### 一、工作需求分析

本项目需要完成一个简易的题库系统，有可直接与用户交互的图形界面，用户可以导入题目，并且可以浏览题目，删除题目和导出题目，题目可以是文字、音频、图像和图形。

#### 1. 用户管理模块：

##### 1.1 注册与登录

功能：

用户注册：提供注册功能，新用户需填写必要信息进行注册。

用户登录：验证用户身份，确保合法用户能够登录系统。

##### 1.2 用户权限

功能：

角色分配：系统管理员能够分配不同角色给用户，如普通用户和管理员。

权限控制：不同角色有不同的操作权限，确保安全性。

#### 2. 题目录入模块：

##### 2.1 直接输入题目

功能：

输入题目：用户能够在系统中直接输入题目、选项和答案。

富文本编辑：支持富文本编辑，使用户能够格式化题目内容。

##### 2.2 模板导入

功能：

模板选择：允许用户选择合适的导入模板。

模板格式检查：系统能够检查导入的模板是否符合系统规范。

#### 3. 题目浏览与管理模块：

##### 3.1 题目分类与标签

功能：

分类管理：系统管理员能够创建、删除题目分类。

标签管理：用户能够为题目添加标签，方便检索。

##### 3.2 浏览题目

功能：

全局浏览：用户能够通过页面浏览题目。

##### 3.3 题目详细信息

功能：

详细查看：用户能够查看题目的详细信息，包括题目内容、选项。

#### 4. 输出模块：

##### 4.1 文件导出

功能：

文件格式导出：支持将选中的题目以Word格式导出。

#### 5. 多媒体题目支持模块：

##### 5.1 语音题目

功能：

录入语音：用户能够录入语音形式的题目。

语音播放：系统能够播放语音题目。

##### 5.2 图形与图像题目

功能：

上传图形与图像：用户能够上传图形和图像形式的题目。

图形查看：系统能够展示图形和图像题目。

#### 6. 系统管理模块：

##### 6.1 数据库管理

功能：

数据备份和还原：系统管理员能够执行数据库备份和还原操作。

数据清理：定期清理无用数据，确保数据库性能。

### 二、系统架构

#### 1.前端架构

1. Vue 3

使用 Vue 3 作为前端框架，利用其组件化和响应式的特性，实现页面的动态展示和交互。

1. Vuex

使用 Vuex 进行状态管理，方便组件间的状态共享和管理。

1. Vue Router

使用 Vue Router 实现前端路由，实现单页应用（SPA）的效果，提高用户体验。

1. Axios

使用 Axios 进行 HTTP 请求，与后端进行数据交互。

1. Element Plus 和[Ant Design Vue](https://antdv.com/index) 的UI 框架

使用 Element Plus 及Ant Design Vue UI 框架，加速界面开发，提供丰富的 UI 组件。

1. Webpack

使用 Webpack 或 Vite 进行前端项目的打包和构建。

#### 2. 后端架构

1. Spring Boot

使用 Spring Boot 作为后端框架，简化开发，提高效率。

1. Spring Security

使用 Spring Security 进行身份验证和授权管理，确保系统安全性。

1. Spring Data JPA

使用 Spring Data JPA 简化对 MySQL 数据库的操作，提高数据访问效率。

1. RESTful API

设计符合 RESTful 风格的 API，实现前后端之间的数据交互。

#### 3. 数据库

1. MySQL

使用 MySQL 作为关系型数据库存储系统核心数据，包括用户信息、题目数据等。

1. 数据库设计

设计规范的数据库表结构，包括用户表、题目表、分类表等。

安全性与性能：

1. Token 鉴权

使用 Token 鉴权机制，保障 API 请求的安全性。

### 三、功能实现解析

#### 3.1 图形界面

前端使用Vue3搭建web网页

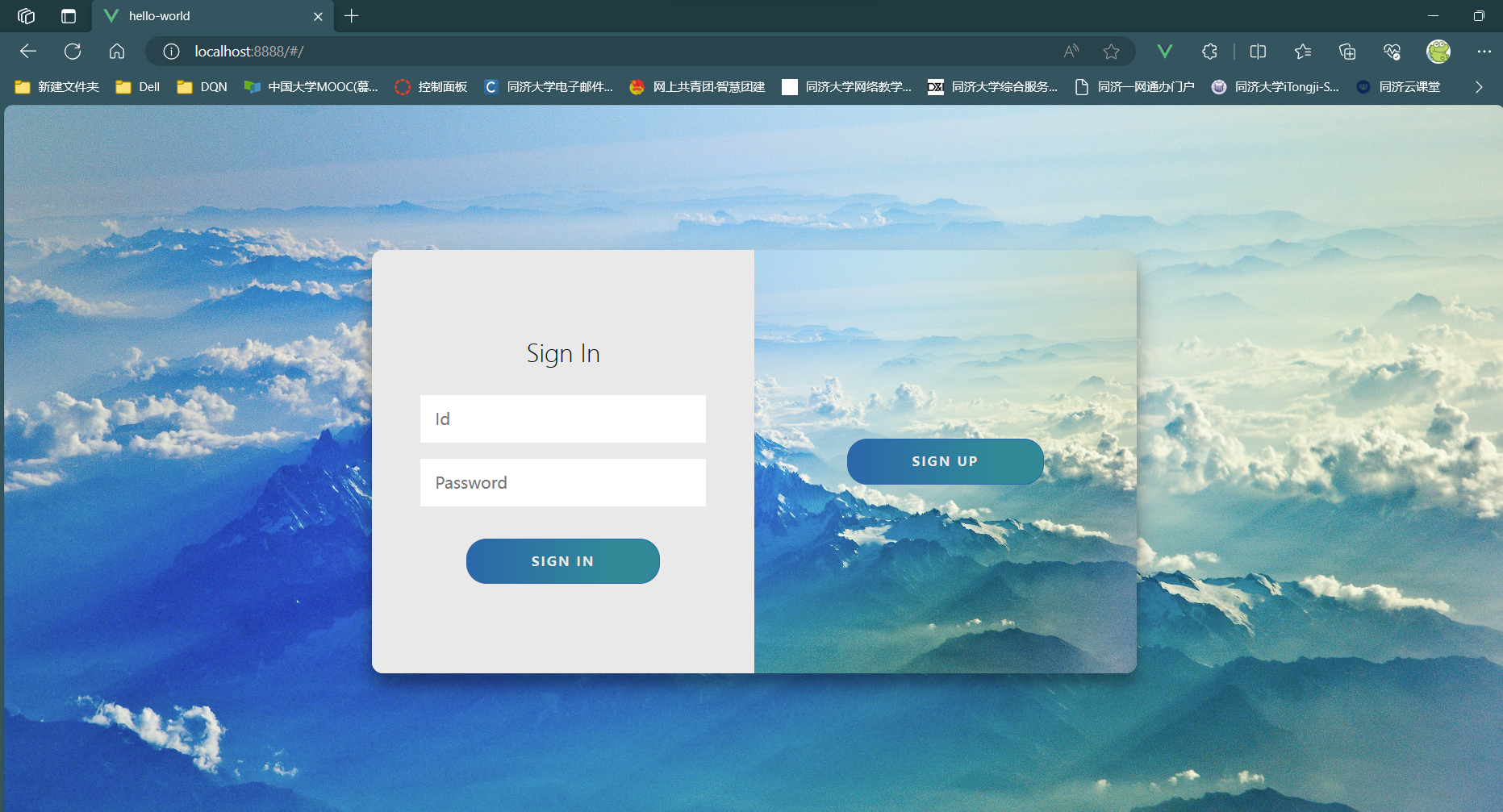
使用 Vue3 构建现代化的图形用户界面。

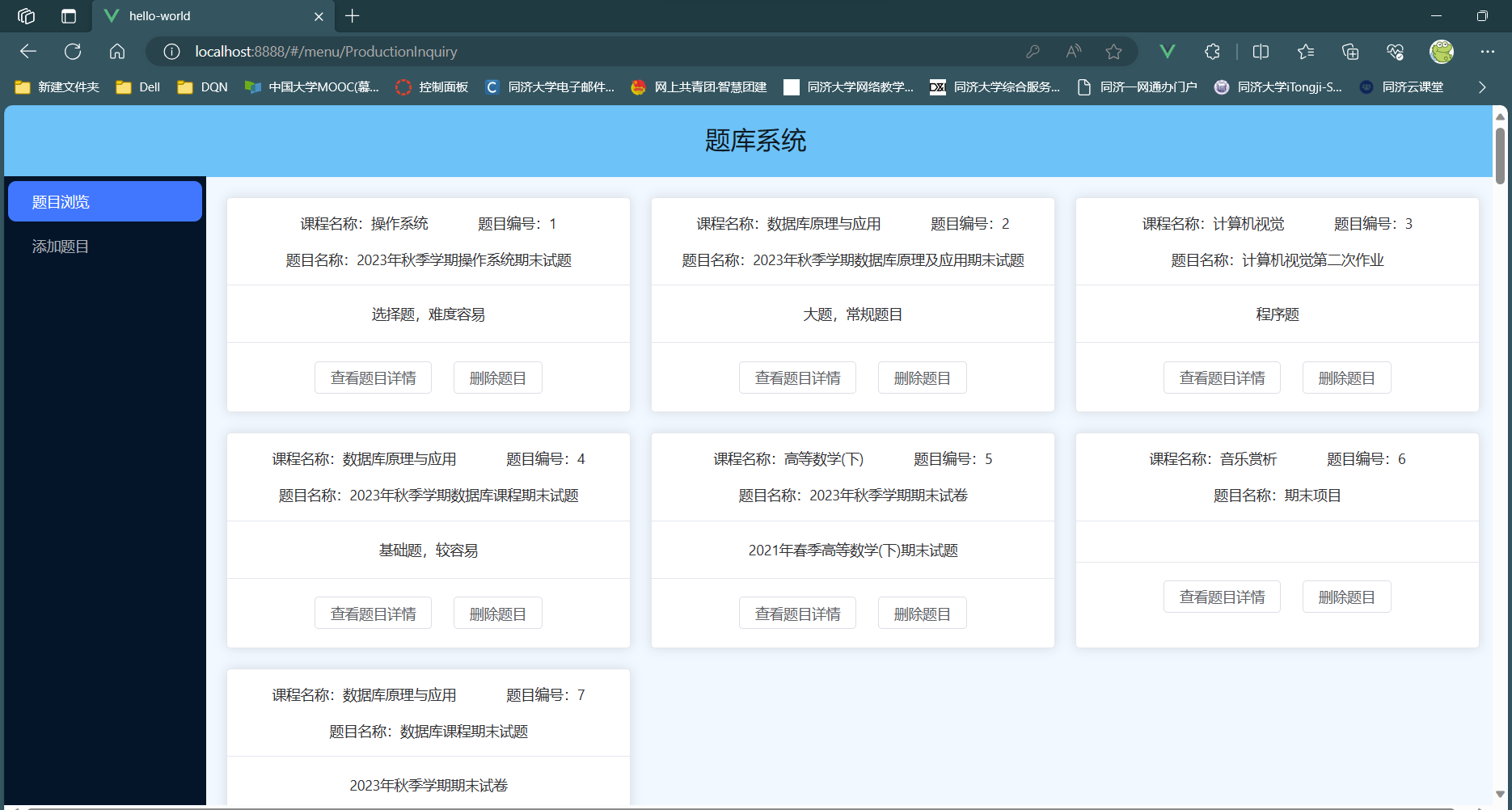
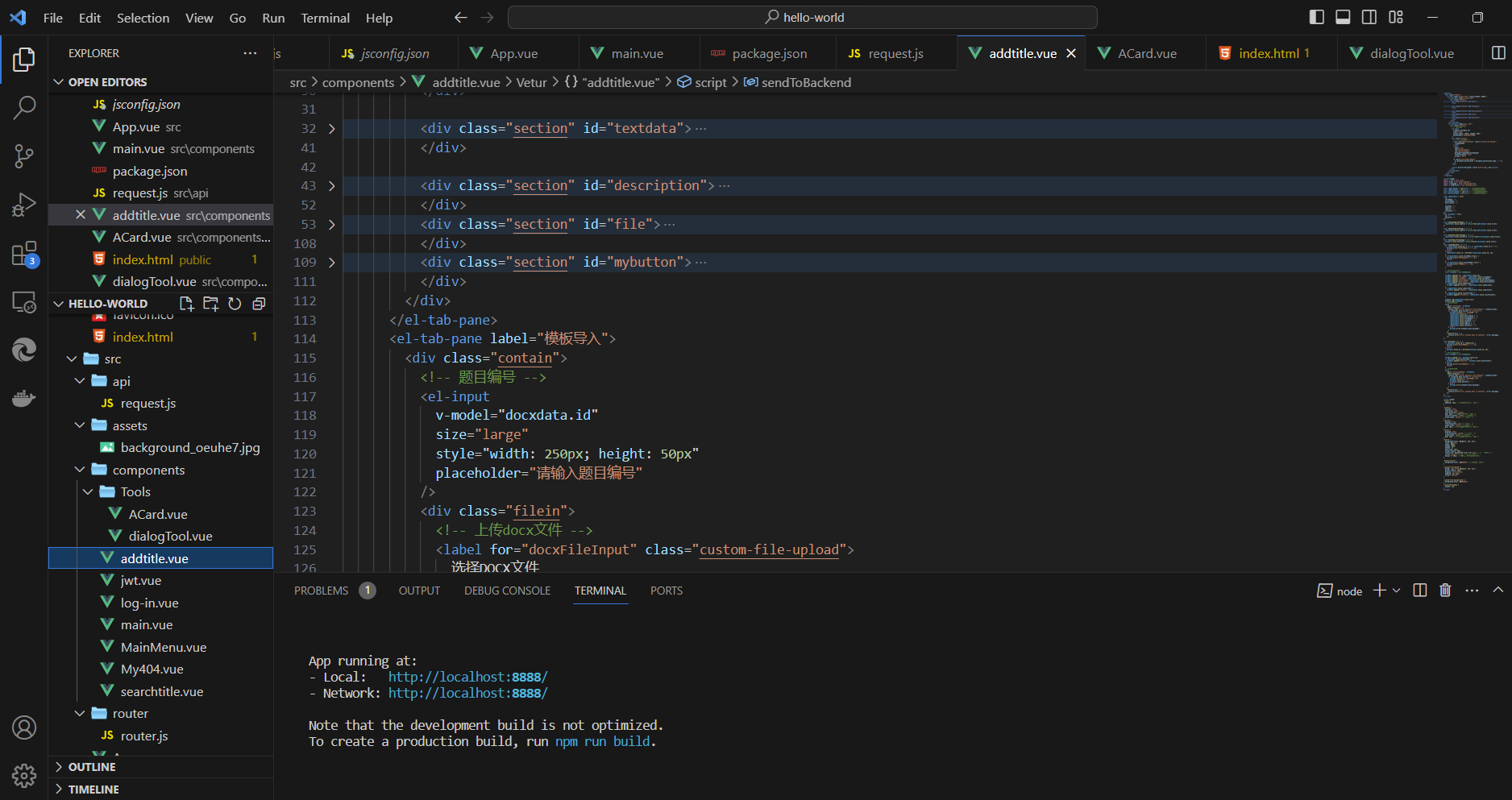
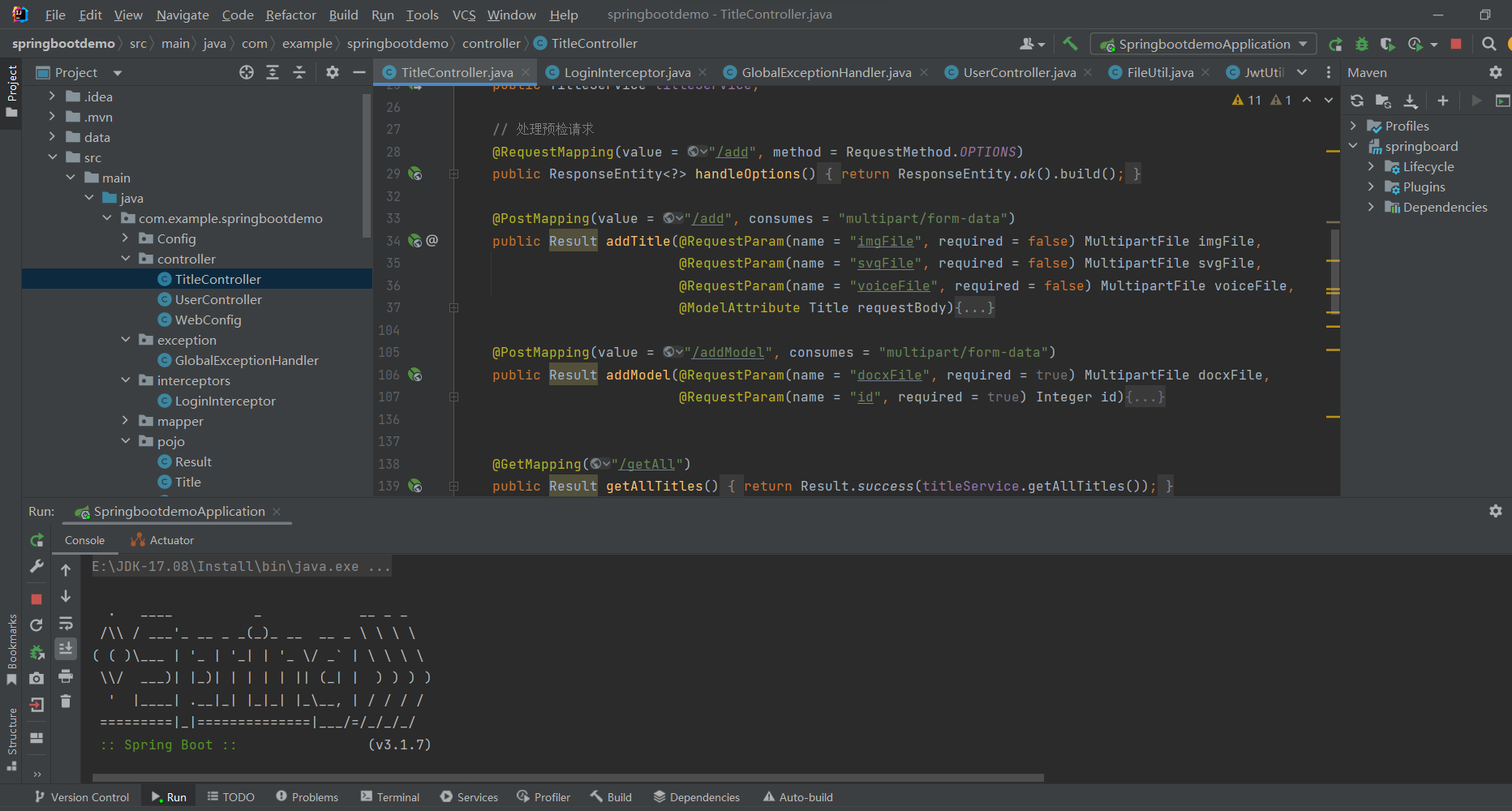
利用 Vue Router 实现路由导航。

使用 Vuex 进行状态管理，确保组件之间的数据共享。

后端（Spring Boot）

提供 RESTful API，为前端提供数据支持。



#### 3.2 题目输入

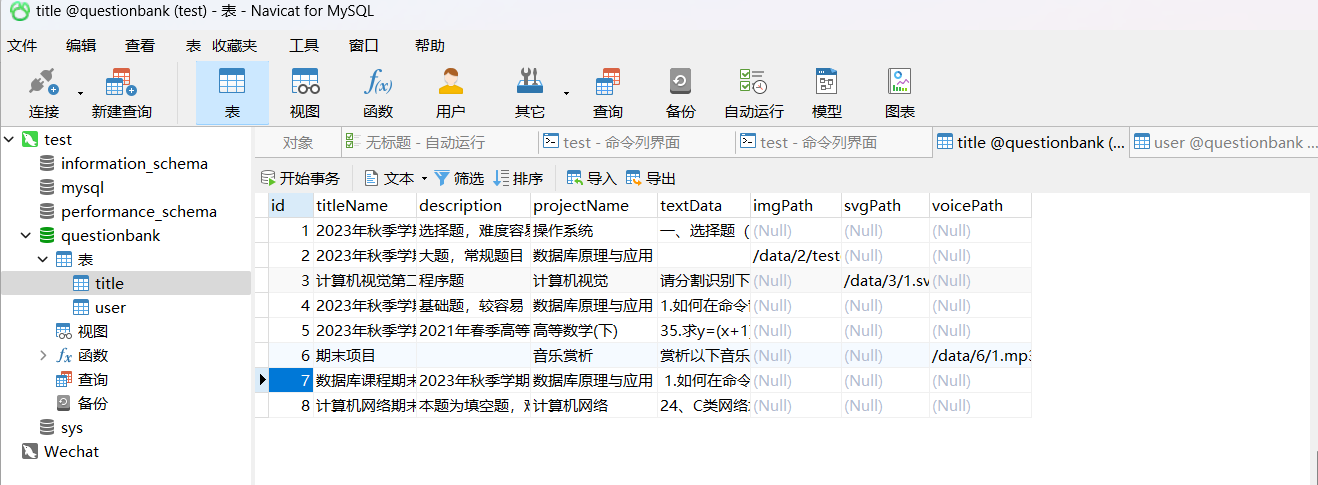
##### 3.2.1 直接输入

前端提供表单页面，用户可以直接在页面上输入题目信息。

后端接收表单数据，将题目信息保存到数据库中。



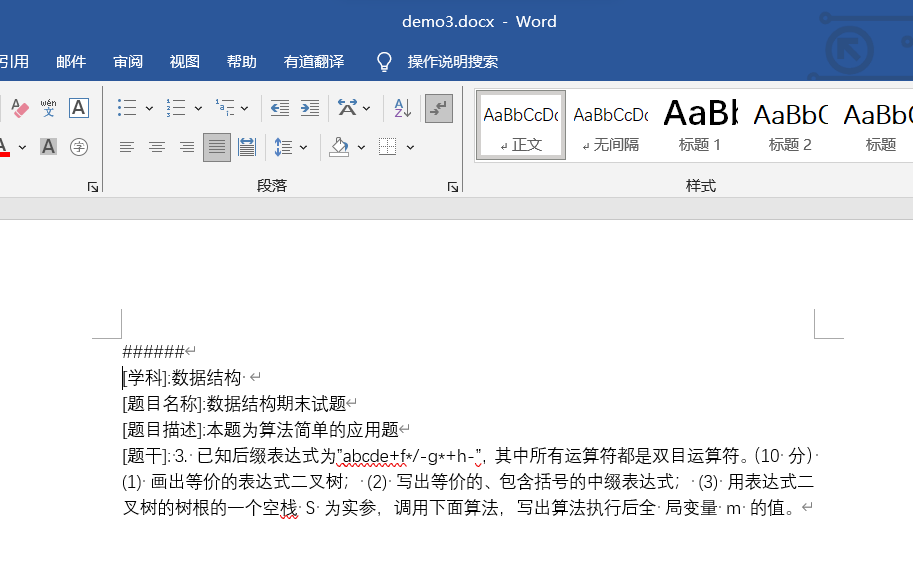




##### 3.2.2 模板导入

前端提供上传功能，用户可以上传题目模板文件。

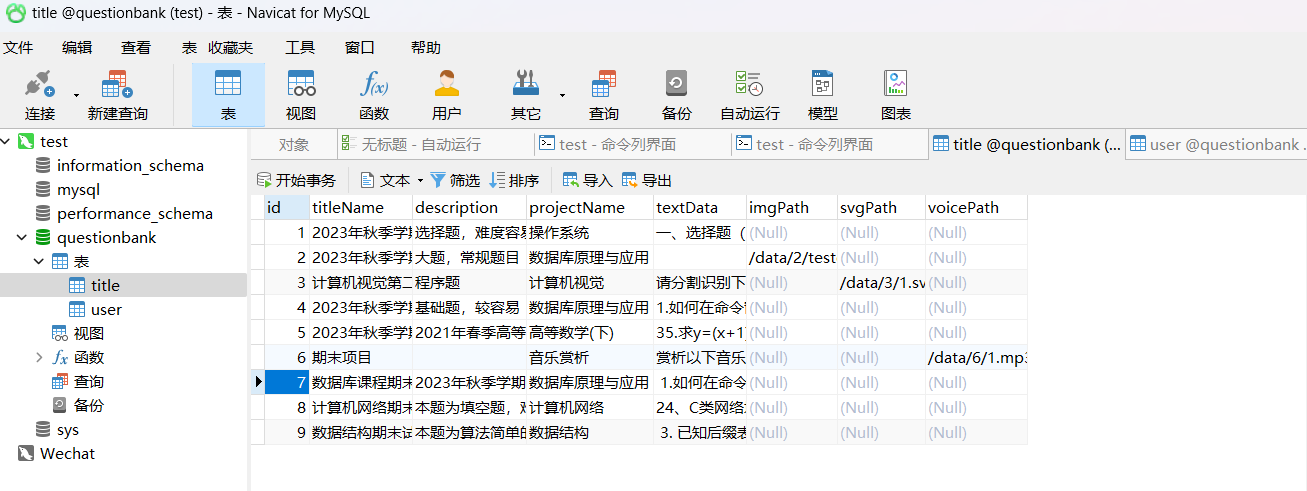
后端解析上传的文件，将题目信息提取并保存到数据库中。主要调用Apache POI来解析docx文档，由于缺少库函数，图像和SVG图形和MP3文件无法使用模板导入，仅支持文本的docx文件导入

#### 3.3 题目存储

使用 MySQL 数据库存储题目信息。

设计数据库表结构，包括题目内容、类型、附件路径等字段。



#### 3.4 题目浏览

前端实现题目浏览页面，展示数据库中的题目列表。

后端提供 API，前端调用 API 获取题目数据。

附加文件将下载到后端的静态资源下

前端根据获取的数据展示给用户

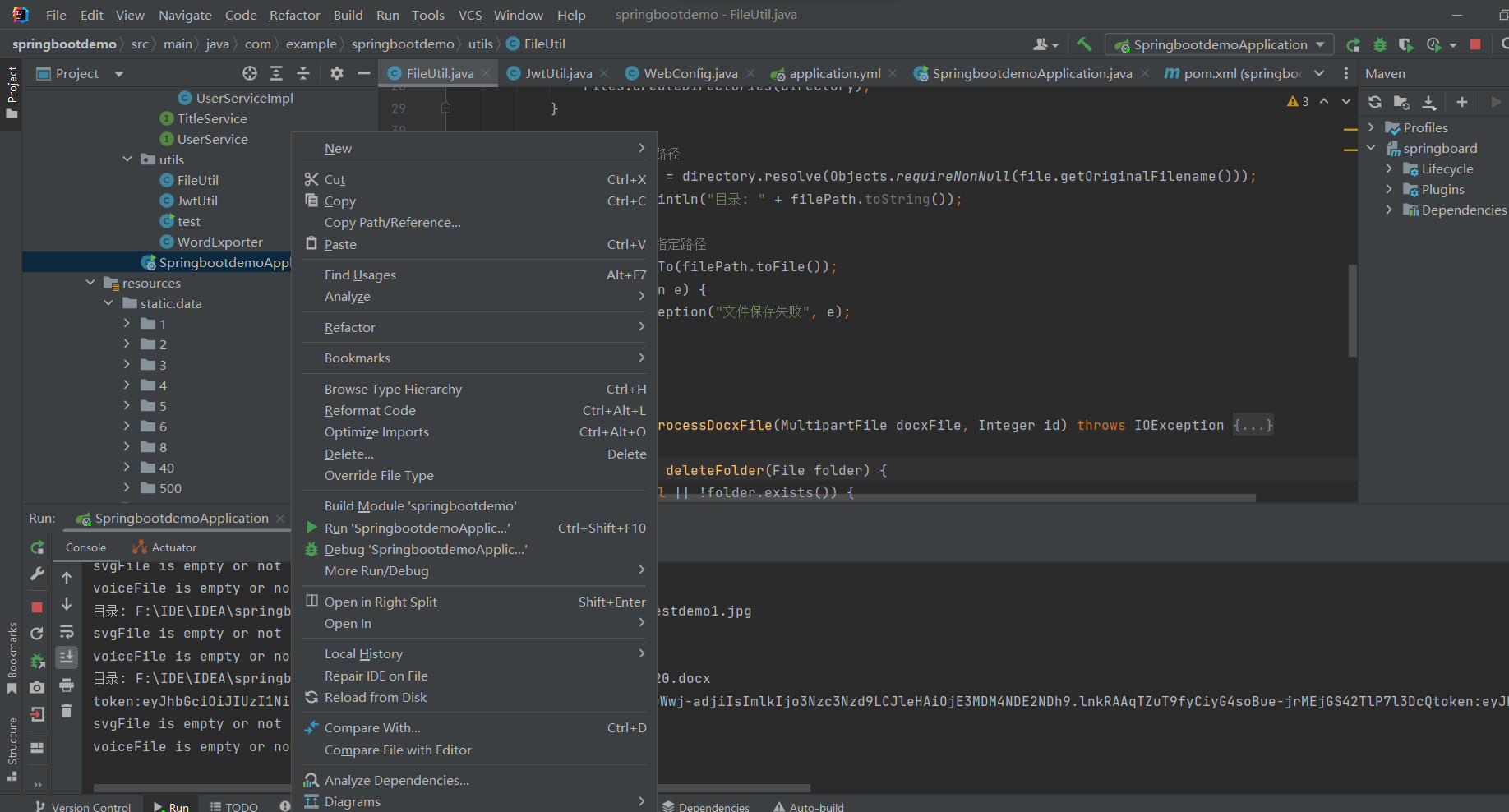


用户可以点击查看详情按钮获取题目完整信息（若是刚上传的带有附加文件的题目，需要在后端重新编译启动加载静态资源后才能正常看到附件文件）

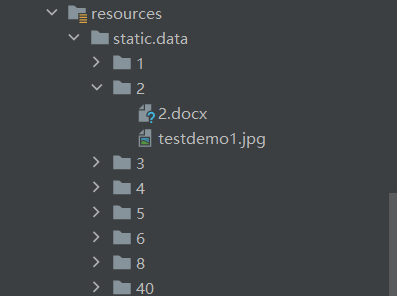




(重新启动启动类)



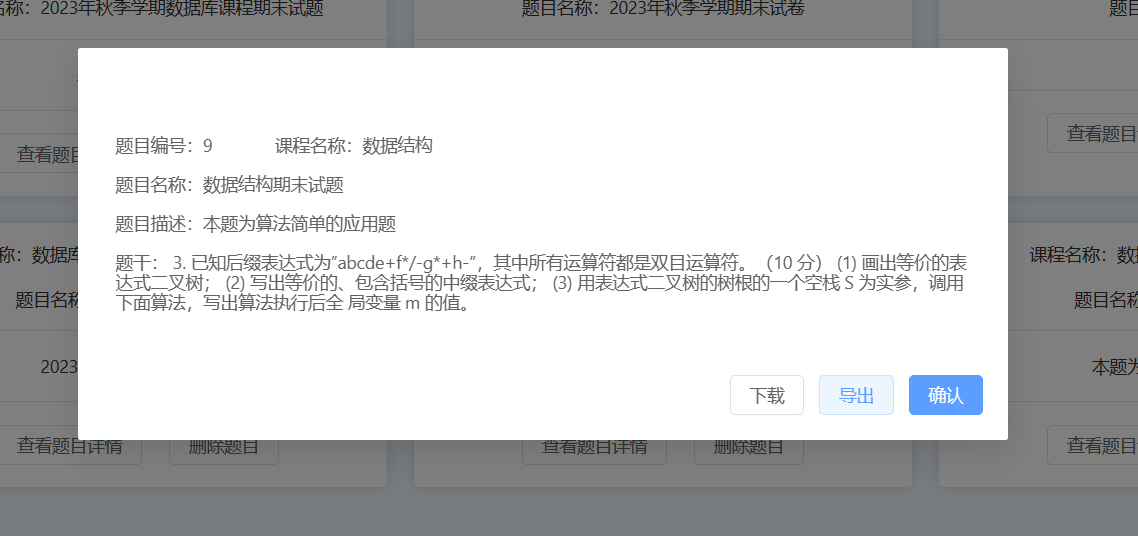
（静态资源位置）



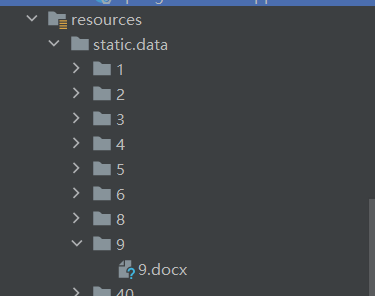
#### 3.5 选择题目并输出

实现浏览功能，用户可以根据条件选择题目。

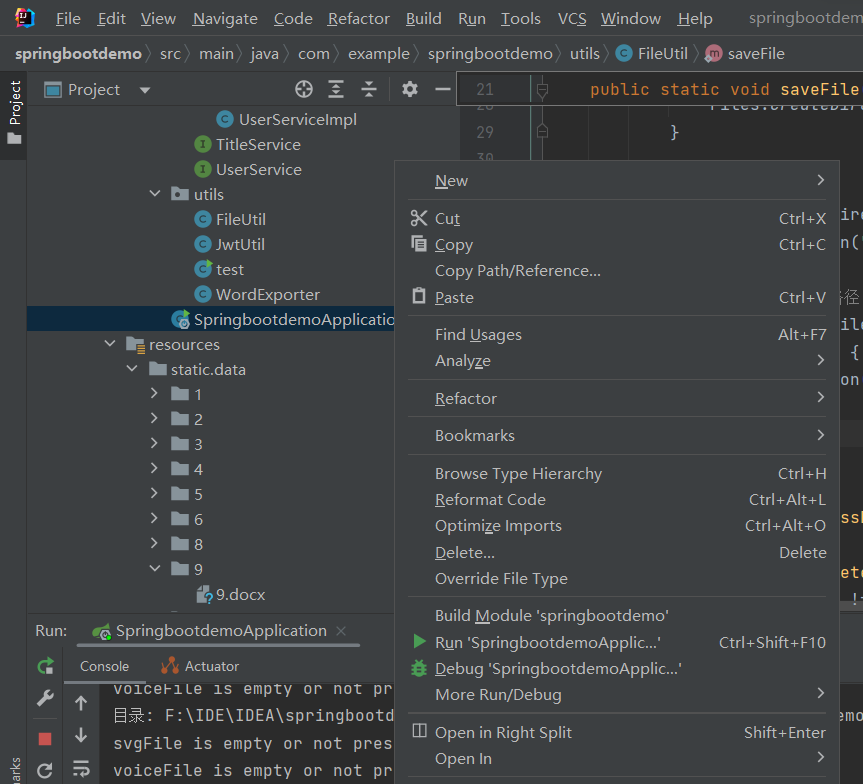
用户导出题目后后端生成Word文档存放在静态资源里，重新编译静态资源后前端用户点击下载按钮可下载生成的文档







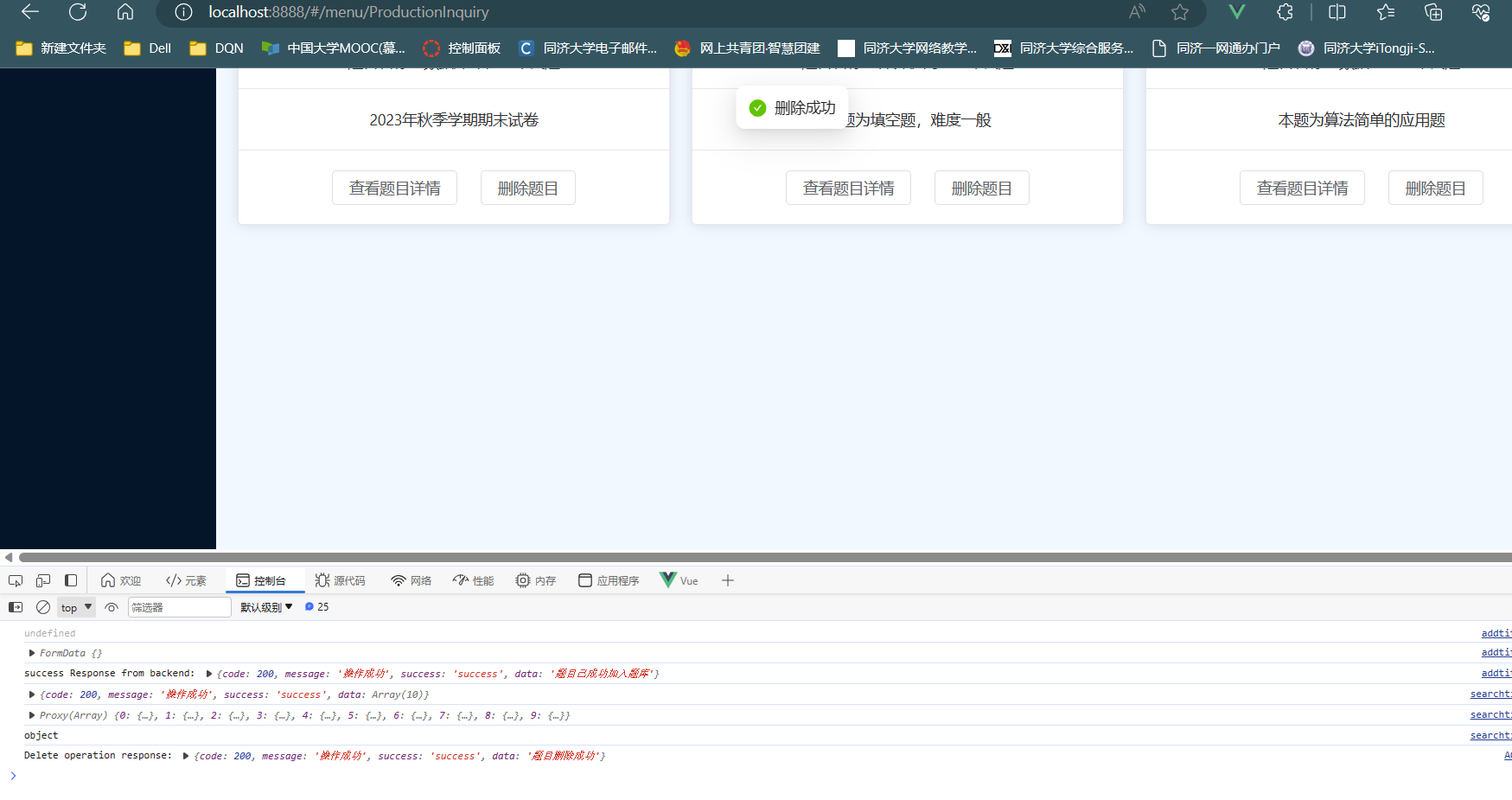
此时重新启动后端前端可下载导出的docx文件（也可开启热部署，后端在静态资源变动后自动重新启动）

#### 3.6删除题目

删除题目会删除数据库的内容和静态资源里的对应题目编号目录下的所有文件(递归删除文件)

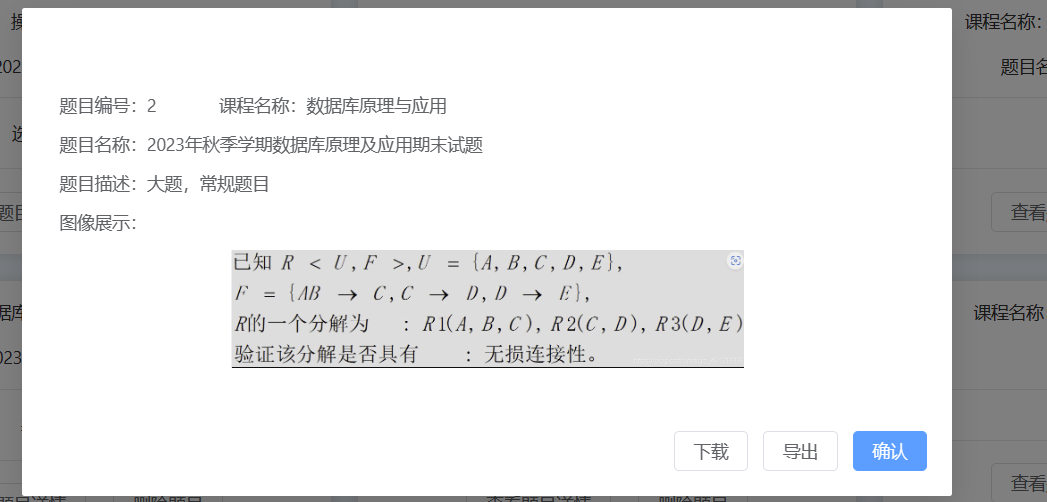
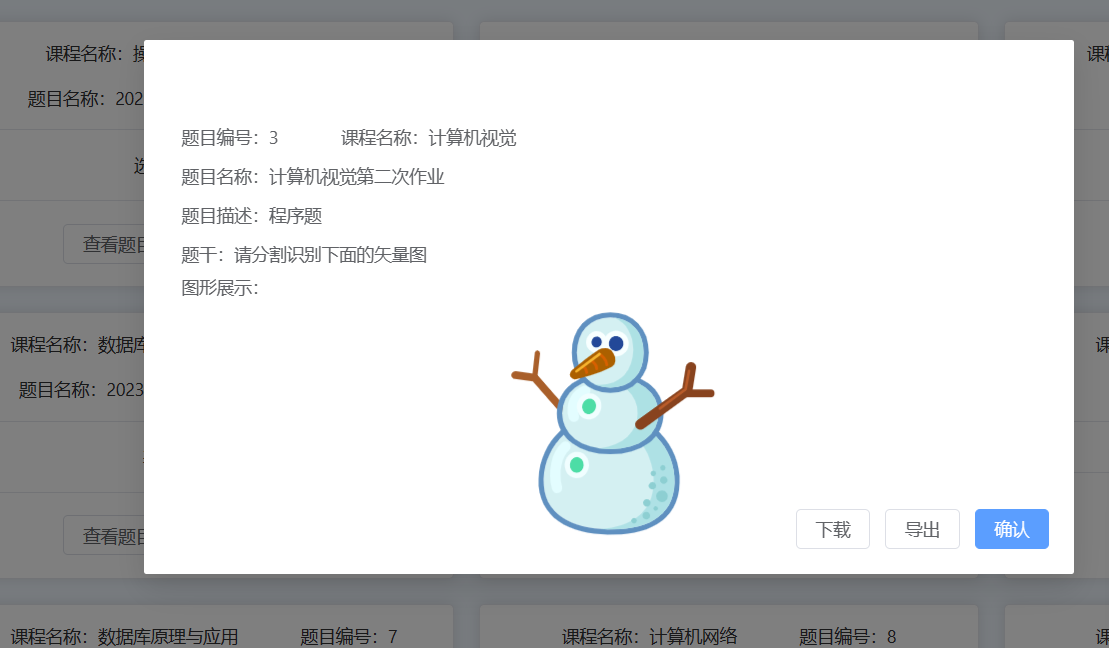




#### 3.7 多种题目形式

考虑使用富文本编辑器，支持多媒体内容的输入。

对于语音、图形、图像题目，需要存储对应的文件路径，并在前端展示或下载。

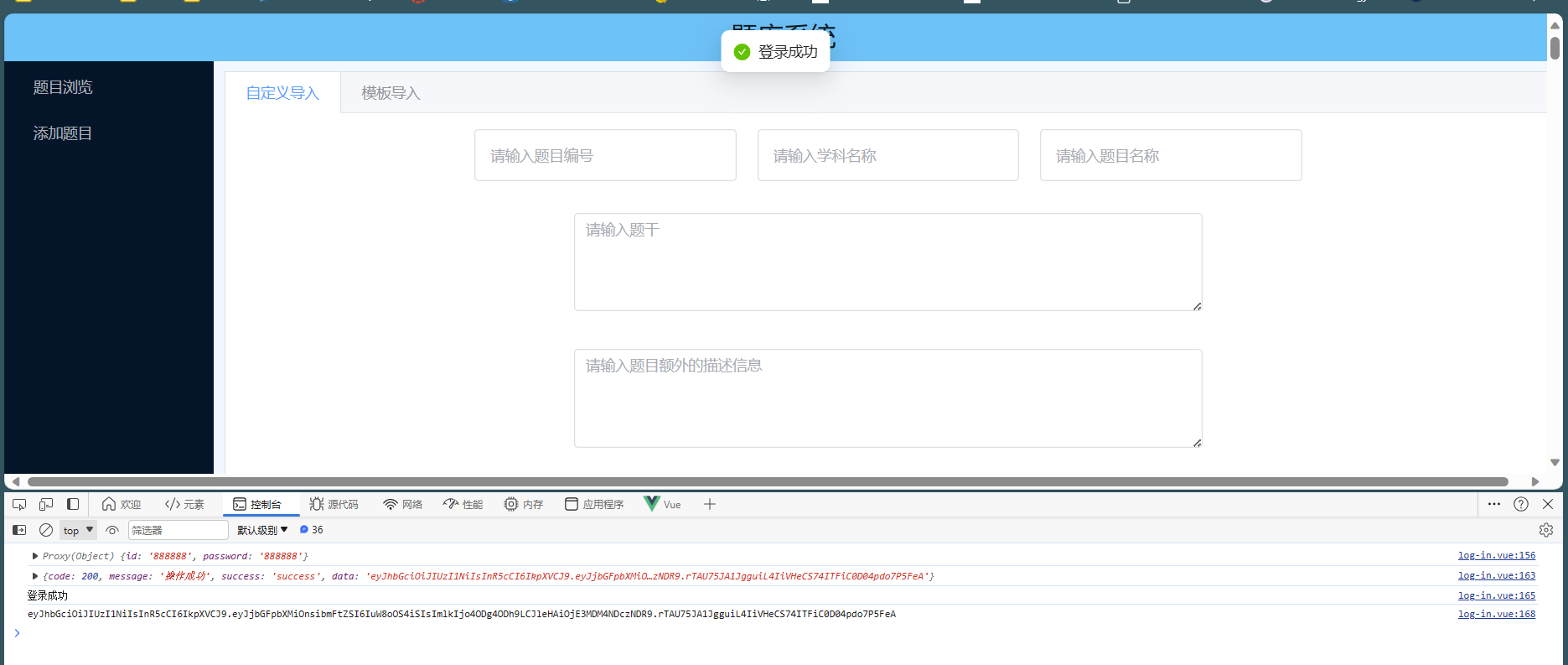
  

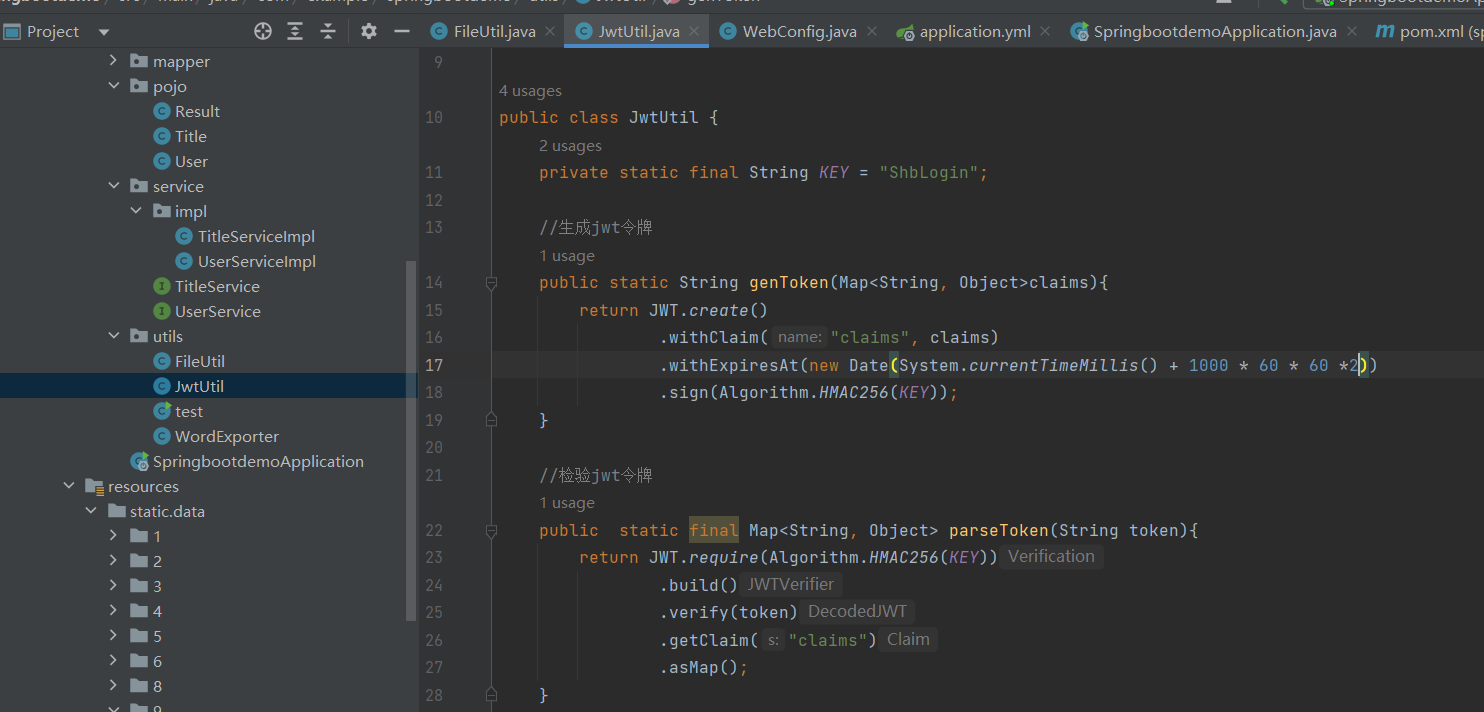
#### 3.8 安全性

实施身份验证和授权机制，确保只有授权用户能够进行操作。成功登陆后返回一个时效性的jwt令牌，在请求头添加后相应拦截器拦截请求，检测是否有有效的token令牌，没有则返回401异常状态码。

对用户输入进行有效性验证，防止恶意输入和攻击。

下图为生成的jwt令牌





#### 3.9 性能优化

对前端进行性能优化，如懒加载、缓存等。

对静态资源存储的优化，例如前后端部署到服务器上，通过绝对路径寻找文件，这样每次改动静态资源文件后就不需要重新编译启动后端。

对数据库进行索引和查询性能优化。

### 四、小结及回顾

在完成题库系统的开发过程中，我首先进行了充分的工作需求分析，明确了系统的功能模块、用户管理、题目录入、题目浏览与管理、输出模块、多媒体题目支持等方面的要求。通过这一步骤，我明确了系统的整体框架和主要功能点，为后续的开发工作奠定了基础。

接着，我通过实现解析，采用了前后端分离的架构，使用Vue 3作为前端框架，Spring Boot作为后端框架，MySQL作为数据库。在前端，我实现了图形界面、题目输入、题目存储、题目浏览、选择题目并输出、多种题目形式等功能。在后端，我设计了RESTful API，实现了用户管理、题目存储、安全性控制、性能优化等关键功能。

在功能的具体实现中，我使用了各种工具和库，如Vue Router、Axios、Element Plus和Ant Design Vue的UI框架等。在多媒体题目的支持中，我使用了Apache POI来解析docx文档，确保了系统对多种题目形式的良好支持。

为了确保系统的安全性，我实施了身份验证和授权机制，采用Token鉴权方式，有效防范了未授权的访问。在性能方面，我以后可以对前端进行懒加载、缓存等方面的优化，并对数据库进行了索引和查询性能的优化。

最后，在小结及回顾中，我可以对整个开发过程进行总结，包括所取得的成果、遇到的挑战以及未来可能的改进和扩展方向。这有助于对项目进行全面的评估，并为后续的维护和升级工作提供指导。