

**数据库课程设计**

**——前台实现文档**

题目： 游戏论坛系统

班级： 计算机科学与技术2班

成员： 黄宜君 202000810067

# 指导老师： 姜秀娥

2022年 6月 15日

目录

[一、引言 4](#_Toc4441)

[二、技术选型 5](#_Toc15332)

[2.1前端技术 5](#_Toc16610)

[2.1.1 React.js 5](#_Toc14582)

[2.1.2 Umi.js 6](#_Toc10690)

[2.1.3 less 6](#_Toc1092)

[2.1.4 antd 7](#_Toc12314)

[2.1.5 TypeScript 8](#_Toc28645)

[2.1后端技术 8](#_Toc21231)

[2.2.1 Node.js 8](#_Toc23844)

[2.2.2 Nest.js 10](#_Toc5055)

[2.2.3 MySql 10](#_Toc25264)

[2.2.4 TypeScript 11](#_Toc1605)

[三. 界面展示 11](#_Toc28731)

[3.1主要界面展示 11](#_Toc25191)

[3.1.1 论坛主界面 11](#_Toc12583)

[3.1.2 板块主界面 12](#_Toc29900)

[3.1.3 登录注册界面 13](#_Toc31521)

[3.1.4 个人信息界面 13](#_Toc4482)

[3.1.5 文章阅读界面 14](#_Toc15025)

[3.1.6 文章发布界面 15](#_Toc224)

[3.1.7用户反馈界面 15](#_Toc5395)

[3.1.8 Swagger后端接口界面 16](#_Toc24677)

[3.2功能界面展示 18](#_Toc29211)

[3.2.1 用户注册功能 18](#_Toc32677)

[3.2.2 用户登录功能 19](#_Toc30193)

[3.2.3 用户个人信息更新功能 20](#_Toc16770)

[3.2.4 文章发布功能 21](#_Toc29153)

[3.2.5 文章评论功能 23](#_Toc3136)

[3.2.6 用户关注功能 24](#_Toc19615)

[四、总结 27](#_Toc26428)

[4.1功能总结 27](#_Toc9641)

[4.2不足与收获 27](#_Toc6983)

# 一、引言

随着科技的发展，基本上所有的具有一定数量数据的机构都开始使用计算机数据库来做管理。几乎所有学校也都已经在使用计算机管理数据的机制，大大减少了学校学生成绩管理的工作量。该课程设计要求设计一个学生成绩的数据库管理系统，数据库中要求包含学生的基本信息，课程基本信息，以及学生所学课程的考试成绩。要方便学生进行成绩查询，通过该课程设计，应该达到把数据库理论知识更加的巩固加深，加强动手能力与实践能力，学以致用，与现实生活中的应用充分的结合起来。

随着网络化席卷全球，从市场本身到商家以及消费者都在日益接受网络带来的变化，网络论坛营销也因此应运而生，并且因其独有的特点正在成为现代营销市场的主流，无论营销环境还是营销方法都是一个转换过程。企业开展论坛营销应该遵循网页策略、产品策略、价格策略、促销策略、渠道策略、网络营销的顾客服务等等策略。论坛营销在中国企业中的应用正逐步走向深入，其中游戏论坛占据了很大的比例。

数据库课程设计旨在提供学生独立设计数据库项目的机会，提升学生对数据库设计实践的能力。本项目是基于数据库课程设计的前后端分离web项目，以【游戏论坛】为题，基本独立实现了游戏论坛的主要功能。

# 二、技术选型

本项目是基于Web的前后端分离项目。出于新型技术尝试的想法，本项目抛弃传统的java后端实现，选择了前后端均为Nodejs+Typescript的模式。

## **2.1前端技术**

### 2.1.1 React.js

### 

React 是一个用于构建用户界面的 JAVASCRIPT 库。React 主要用于构建 UI，很多人认为 React 是 MVC 中的 V（视图）。React 起源于 Facebook 的内部项目，用来架设 Instagram 的网站，并于 2013 年 5 月开源。React 拥有较高的性能，代码逻辑非常简单，越来越多的人已开始关注和使用它。

React是较为先进，更新较为稳定的前端框架。**本项目**主要语法采用React.js，充分利用其声明式渲染、组件化编程等特性。

### 2.1.2 Umi.js

IMG_256

umi，中文可发音为乌米，是一个可插拔的企业级 react 应用框架。umi 以路由为基础的，支持[类 next.js 的约定式路由](https://umijs.org/zh/guide/router.html" \t "https://v2.umijs.org/zh/guide/_blank)，以及各种进阶的路由功能，并以此进行功能扩展，比如[支持路由级的按需加载](https://umijs.org/zh/plugin/umi-plugin-react.html" \l "dynamicimport" \t "https://v2.umijs.org/zh/guide/_blank)。然后配以完善的[插件体系](https://umijs.org/zh/plugin/" \t "https://v2.umijs.org/zh/guide/_blank)，覆盖从源码到构建产物的每个生命周期，支持各种功能扩展和业务需求，目前内外部加起来已有 50+ 的插件。umi 是蚂蚁金服的底层前端框架，已直接或间接地服务了 600+ 应用，包括 java、node、H5 无线、离线（Hybrid）应用、纯前端 assets 应用、CMS 应用等。他已经很好地服务了我们的内部用户，同时希望他也能服务好外部用户。

**本项目**采用Umijs对react环境进行继承和简化，充分利用了其特殊的路由约定、集成的插件体系、良好的TypeScript支持等功能。

### 2.1.3 less

IMG_256

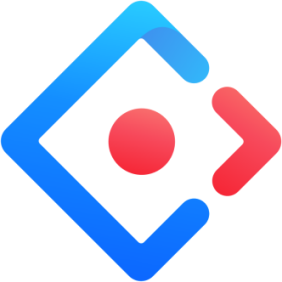
Less （Leaner Style Sheets 的缩写） 是一门向后兼容的 CSS 扩展语言。包含了 Less 语言以及利用 JavaScript 开发的用于将 Less 样式转换成 CSS 样式的 Less.js 工具。

LESS是一个CSS预处理器，可以为网站启用可自定义，可管理和可重用的样式表。 LESS是一种动态样式表语言，扩展了CSS的功能。 LESS也是跨浏览器友好。

CSS预处理器是一种脚本语言，可扩展CSS并将其编译为常规CSS语法，以便可以通过Web浏览器读取。 它提供诸如变量，函数， mixins 和操作等功能，可以构建动态CSS。

**本项目**使用less语法，完成前端界面的编写。

### 2.1.4 antd



## antd 是基于 Ant Design 设计体系的 React UI 组件库，主要用于研发企业级中后台产品。特性有：提炼自企业级中后台产品的交互语言和视觉风格。开箱即用的高质量 React 组件。使用 TypeScript 开发，提供完整的类型定义文件。全链路开发和设计工具体系。数十个国际化语言支持。深入每个细节的主题定制能力。

**本项目**结合antd组件库，实现复杂页面的编写。

### 2.1.5 TypeScript

IMG_256

TypeScript 是 JavaScript 的一个超集，支持 ECMAScript 6 标准（[ES6 教程](https://www.runoob.com/w3cnote/es6-tutorial.html" \t "https://www.runoob.com/typescript/_blank)）。

TypeScript 由微软开发的自由和开源的编程语言。

TypeScript 设计目标是开发大型应用，它可以编译成纯 JavaScript，编译出来的 JavaScript 可以运行在任何浏览器上。

TypeScript 是 JavaScript 的超集，扩展了 JavaScript 的语法，因此现有的 JavaScript 代码可与 TypeScript 一起工作无需任何修改，TypeScript 通过类型注解提供编译时的静态类型检查。

TypeScript 可处理已有的 JavaScript 代码，并只对其中的 TypeScript 代码进行编译。

**本项目**使用Typescript作为前端语言，利用其优秀的接口规范功能提升项目的规范性。

## **2.1后端技术**

### 2.2.1 Node.js

IMG_256

Node.js 是一个开源和跨平台的 JavaScript 运行时环境。 它几乎是任何类型项目的流行工具！

Node.js 在浏览器之外运行 V8 JavaScript 引擎（Google Chrome 的内核）。 这使得 Node.js 的性能非常好。

Node.js 应用程序在单个进程中运行，无需为每个请求创建新的线程。 Node.js 在其标准库中提供了一组异步的 I/O 原语，以防止 JavaScript 代码阻塞，通常，Node.js 中的库是使用非阻塞范式编写的，使得阻塞行为成为异常而不是常态。

当 Node.js 执行 I/O 操作时（比如从网络读取、访问数据库或文件系统），Node.js 将在响应返回时恢复操作（而不是阻塞线程和浪费 CPU 周期等待）。

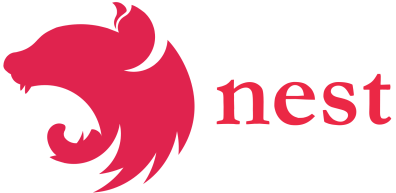
这允许 Node.js 使用单个服务器处理数千个并发连接，而​​不会引入管理线程并发（这可能是错误的重要来源）的负担。

Node.js 具有独特的优势，因为数百万为浏览器编写 JavaScript 的前端开发者现在无需学习完全不同的语言，就可以编写除客户端代码之外的服务器端代码。

在 Node.js 中，可以毫无问题地使用新的 ECMAScript 标准，因为你不必等待所有用户更新他们的浏览器，你负责通过更改 Node.js 版本来决定使用哪个 ECMAScript 版本，你还可以通过运行带有标志的 Node.js 来启用特定的实验性功能。

**本项目**利用Nodejs集成Express实现后端服务器的功能。

### 2.2.2 Nest.js



Nest (NestJS) 是一个用于构建高效、可扩展的 [Node.js](https://nodejs.org/" \t "https://nestjs.bootcss.com/_blank) 服务器端应用程序的开发框架。它利用 JavaScript 的渐进增强的能力，使用并完全支持 [TypeScript](http://www.typescriptlang.org/" \t "https://nestjs.bootcss.com/_blank) （仍然允许开发者使用纯 JavaScript 进行开发），并结合了 OOP （面向对象编程）、FP （函数式编程）和 FRP （函数响应式编程）。

在底层，Nest 构建在强大的 HTTP 服务器框架上，例如 [Express](https://expressjs.com/" \t "https://nestjs.bootcss.com/_blank) （默认），并且还可以通过配置从而使用 [Fastify](https://github.com/fastify/fastify" \t "https://nestjs.bootcss.com/_blank) ！

Nest 在这些常见的 Node.js 框架 (Express/Fastify) 之上提高了一个抽象级别，但仍然向开发者直接暴露了底层框架的 API。这使得开发者可以自由地使用适用于底层平台的无数的第三方模块。

**本项目**使用nestjs框架，实现后端服务器功能实现的工程化。

### 2.2.3 MySql



MySQL 是最流行的关系型数据库管理系统，在 WEB 应用方面 MySQL 是最好的 RDBMS(Relational Database Management System：关系数据库管理系统)应用软件之一。MySQL数据库 MySQL 是一个关系型数据库管理系统，由瑞典 MySQL AB 公司开发，目前属于 Oracle 公司。MySQL 是一种关联数据库管理系统，关联数据库将数据保存在不同的表中。

### 2.2.4 TypeScript

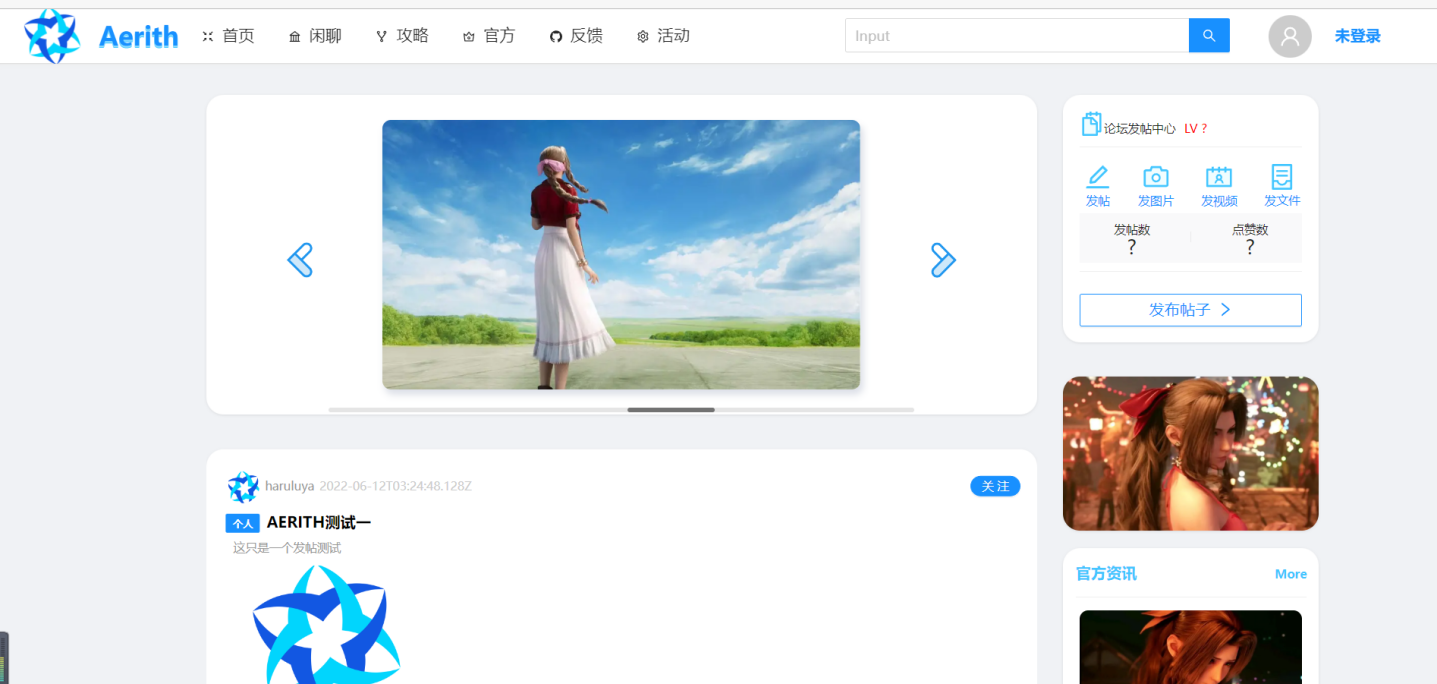
IMG_256

基于Nodejs的前后端实现意味着后端也完全可以抛弃传统的JAVA语言。基于对新型技术的渴望，本项目选择使用Typescript作为后端语言。

# 三. 界面展示

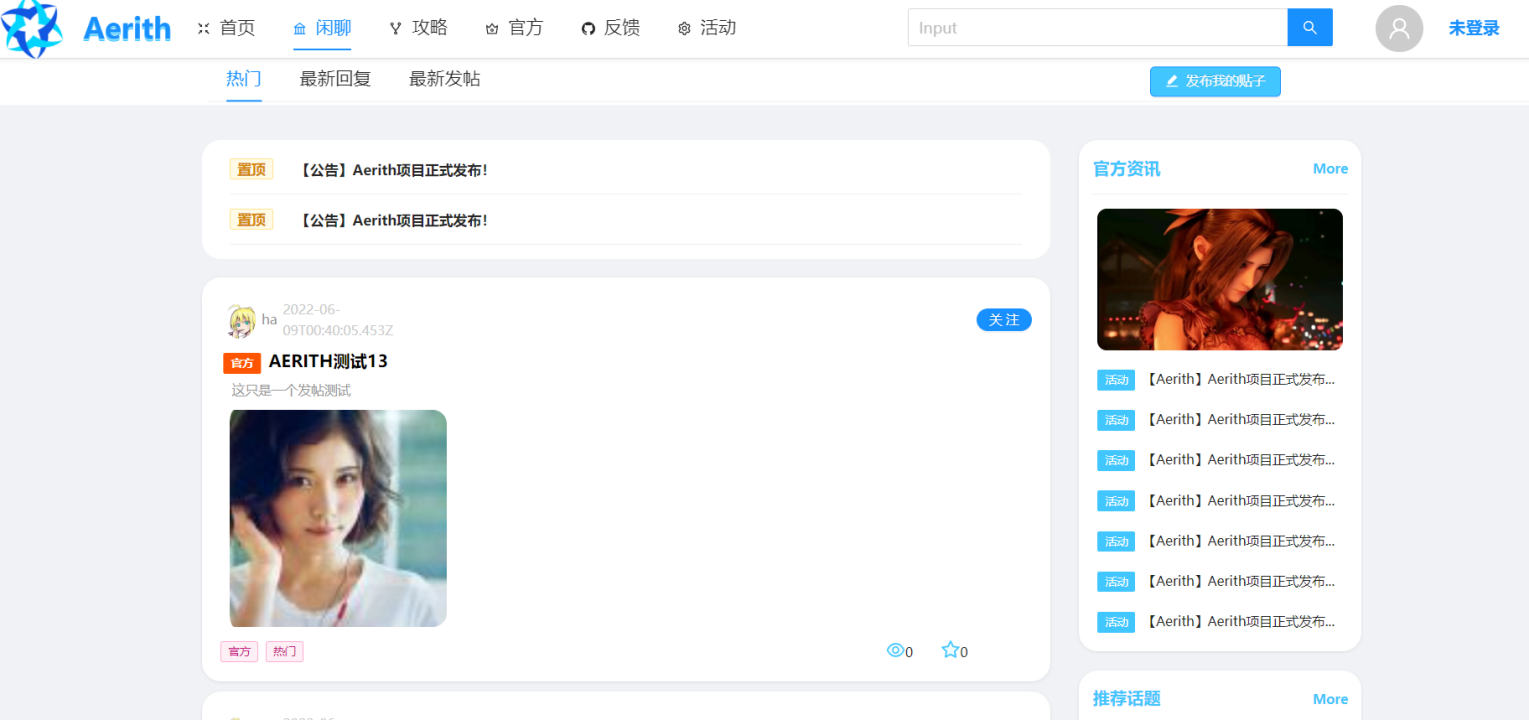
## **3.1主要界面展示**

### 3.1.1 论坛主界面



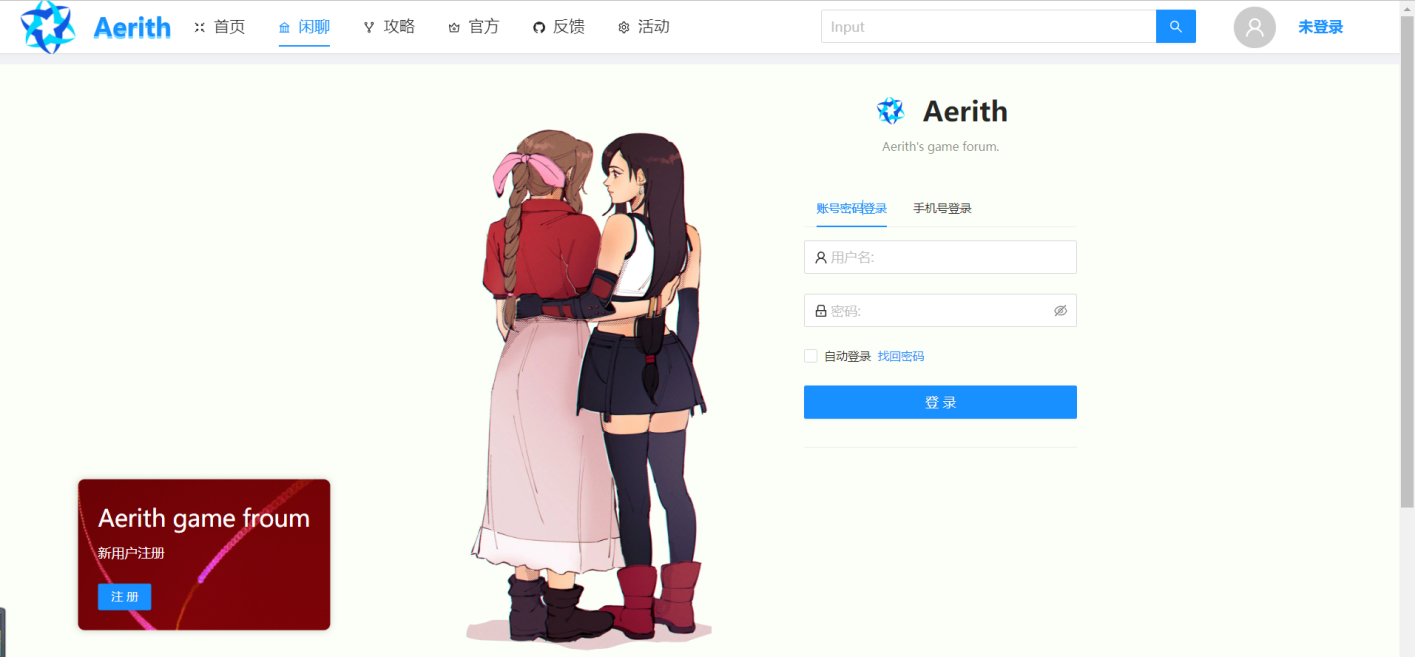
包括轮播图、部分个人信息、活动推荐、热门帖子的显示。

### 3.1.2 板块主界面



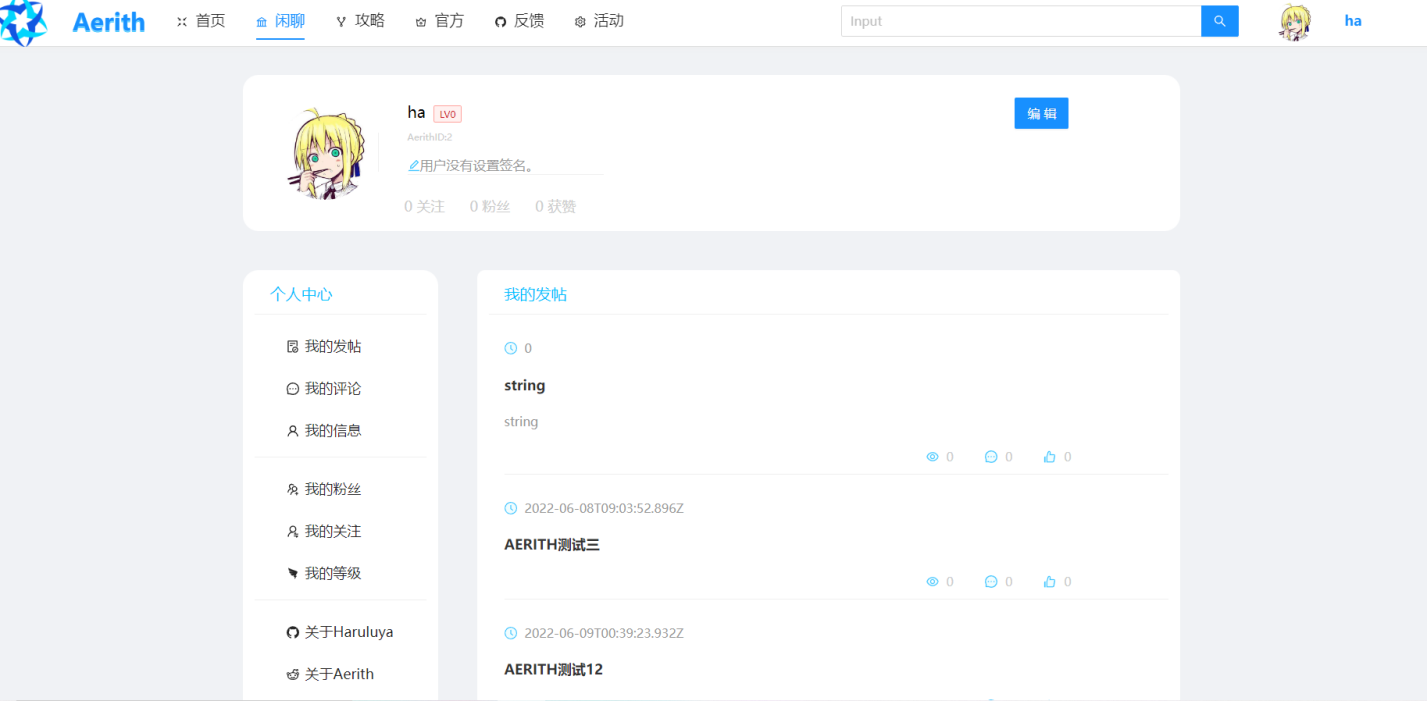
包括置顶信息、所属板块文章和活动推荐的显示。

### 3.1.3 登录注册界面



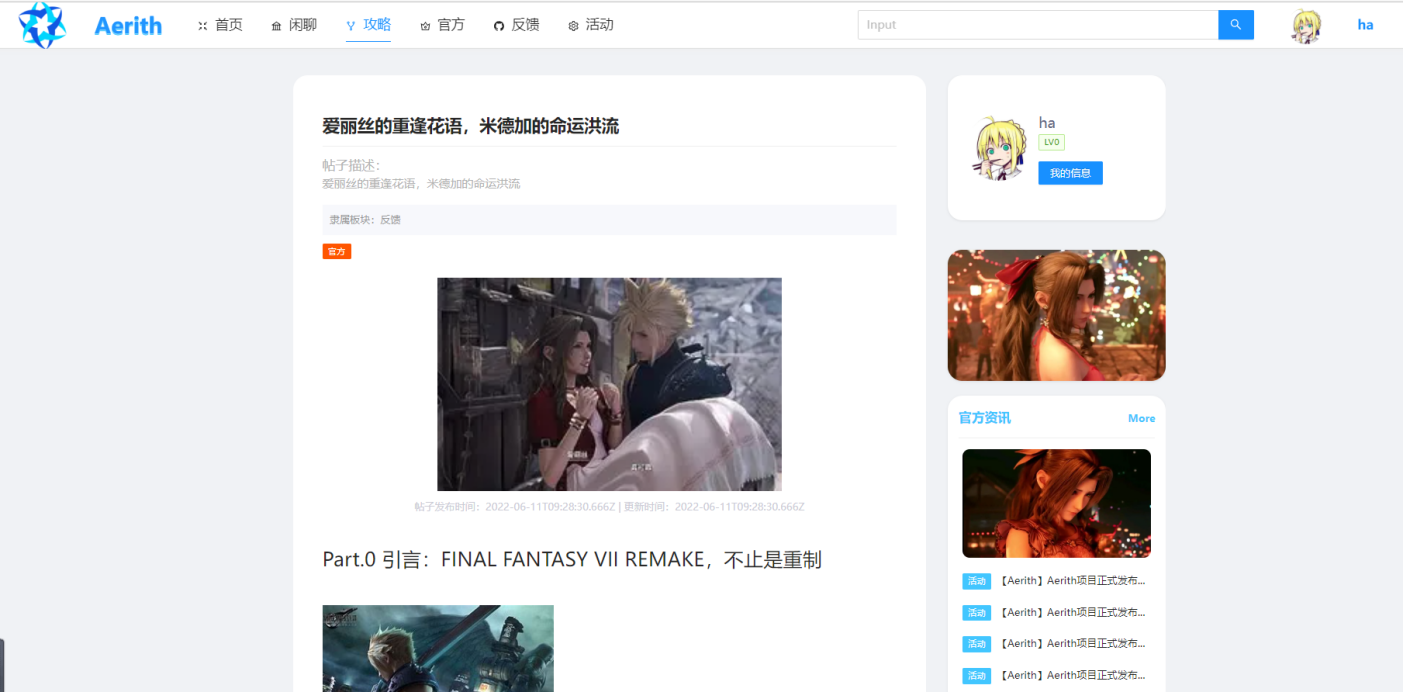
包括登录、注册表单的显示。

### 3.1.4 个人信息界面



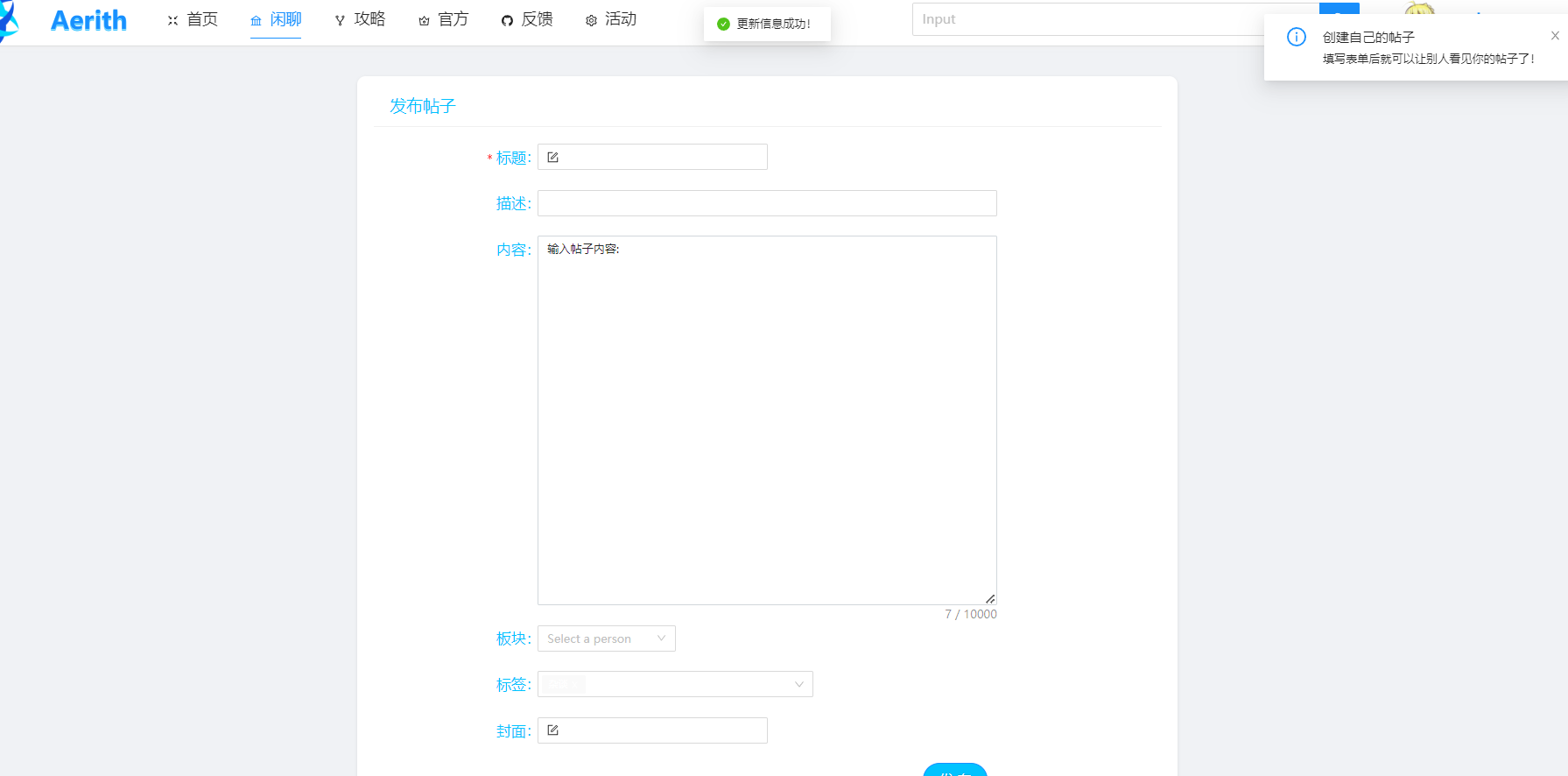
包括个人发帖、评论、身份信息、关注、粉丝、项目信息的展示。

### 3.1.5 文章阅读界面



包括文章标题、模块、内容、作者信息、评论、活动推荐等信息。

### 3.1.6 文章发布界面



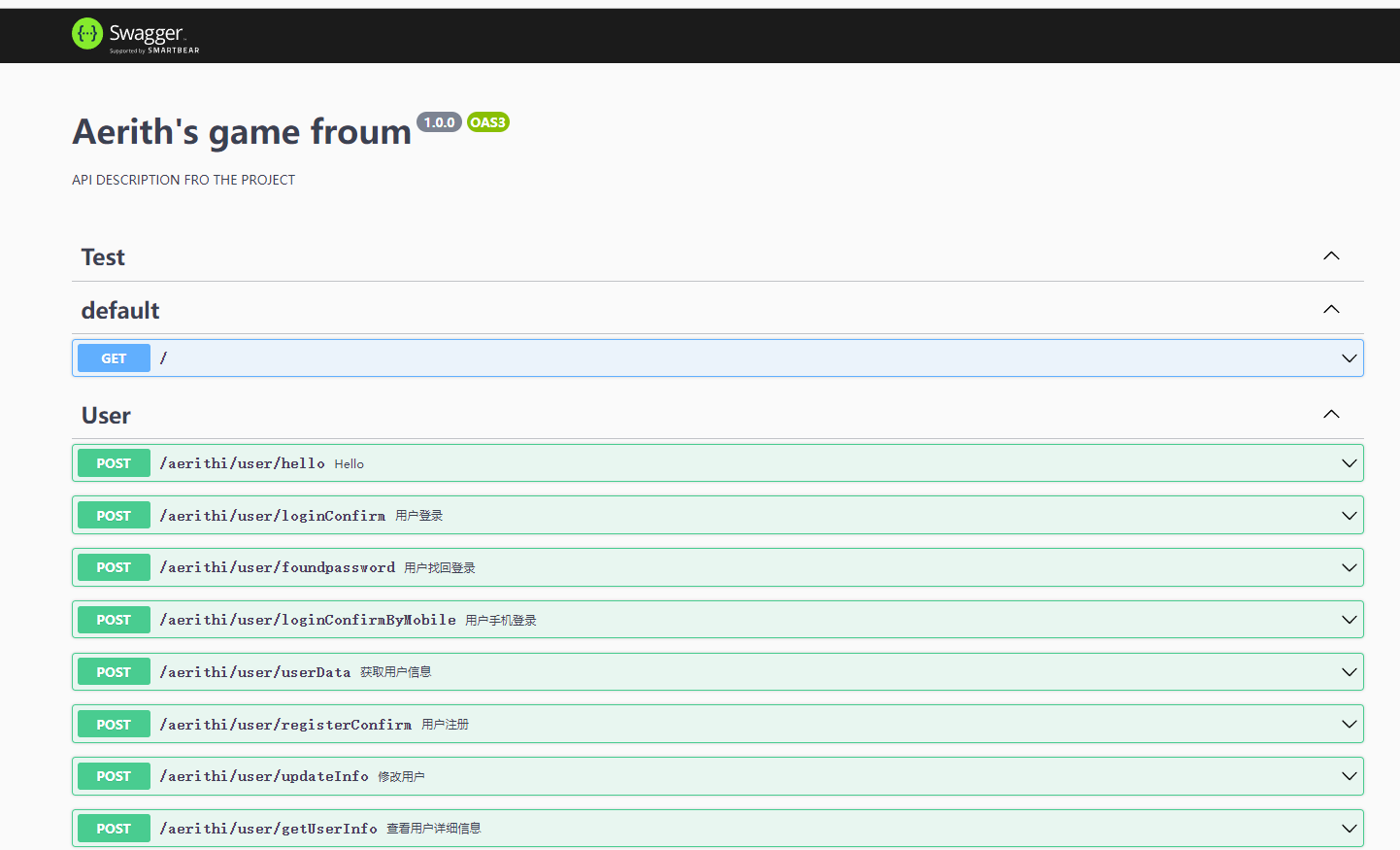
包括文章内容表单的填写界面。

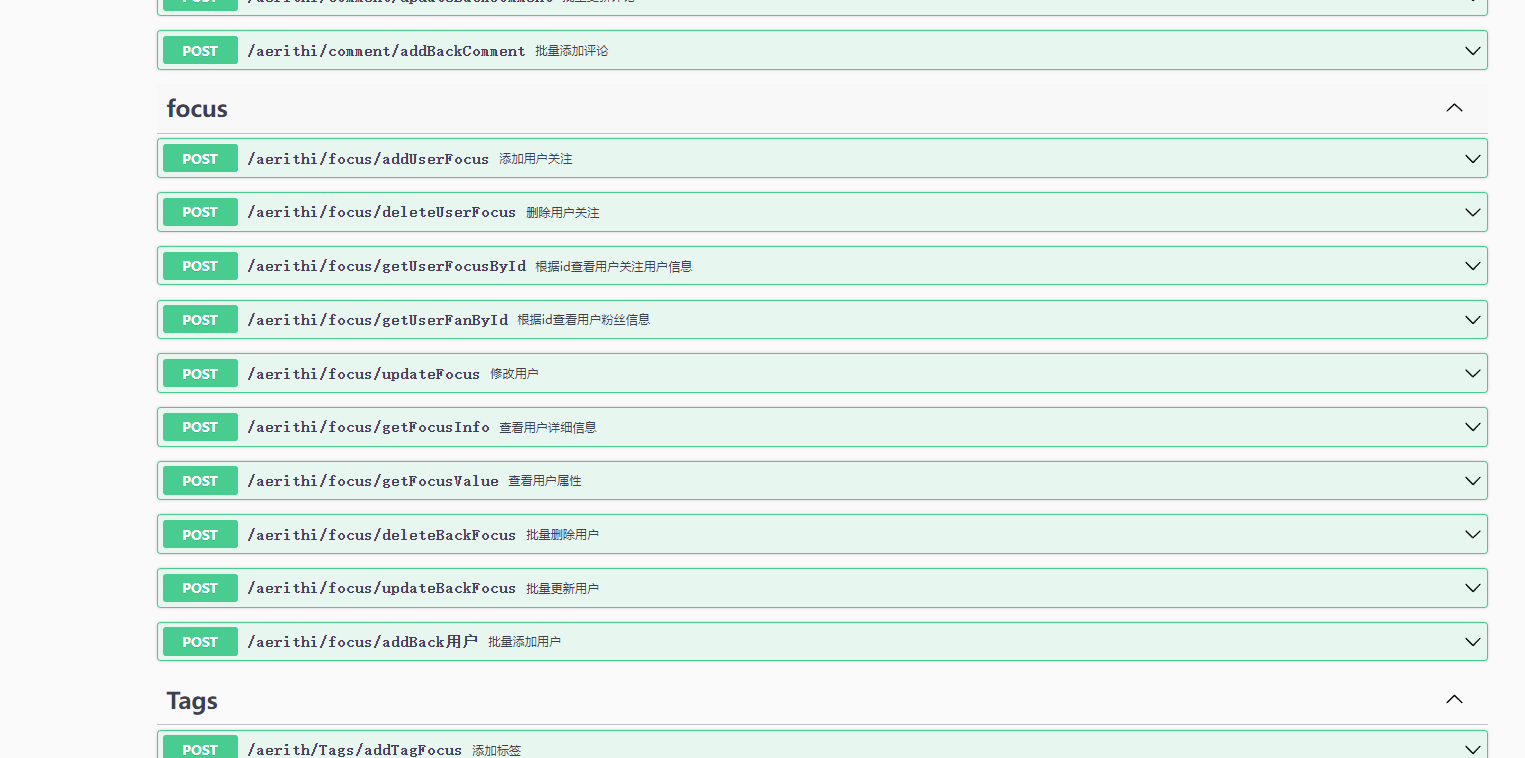
### 3.1.7用户反馈界面

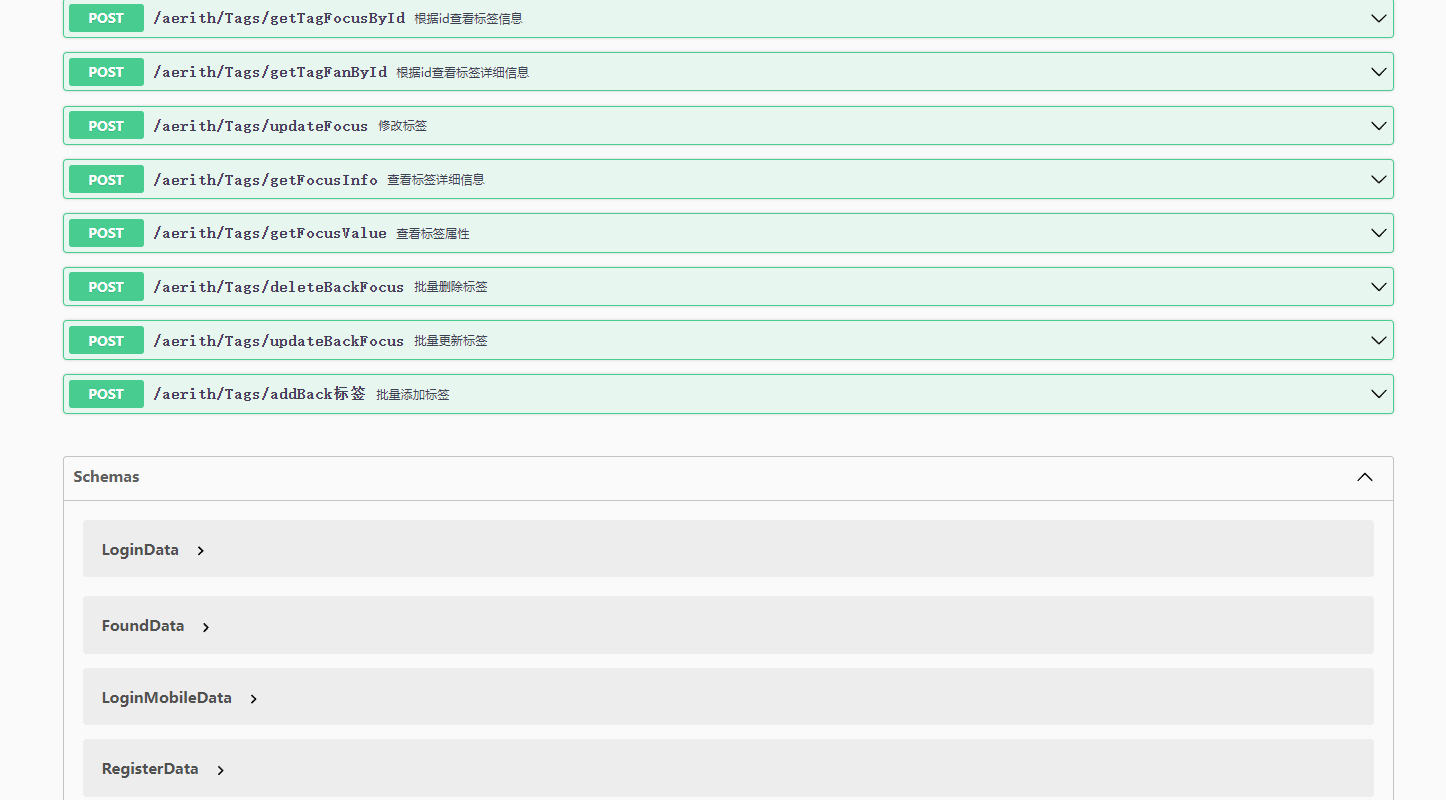


包括用户的反馈内容和处理情况展示。

### 3.1.8 Swagger后端接口界面





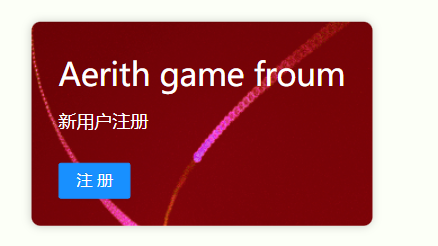


**项目采用NestJS集成Swagger完成后端接口文档，严格遵守工业项目的流程。**

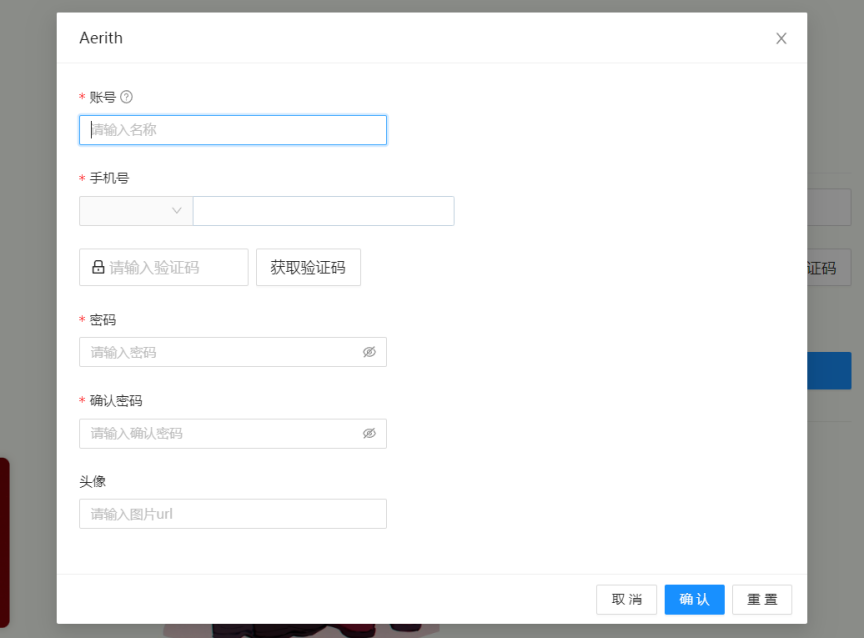
## **3.2功能界面展示**

### 3.2.1 用户注册功能

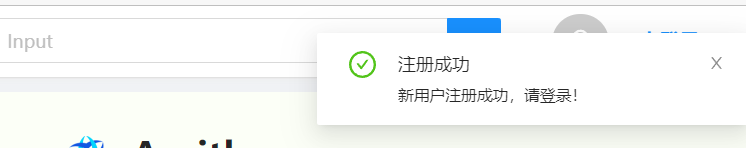
**注册界面入口**



**填写注册表单**



**注册成功**

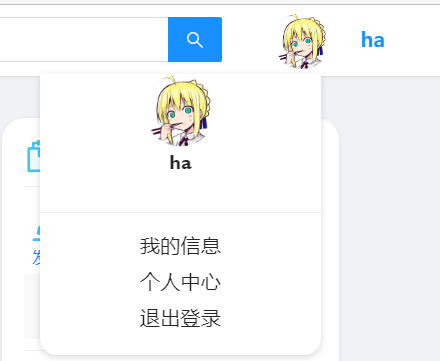


### 3.2.2 用户登录功能

用户可选择账号密码登录或手机号登录，手机号登录以发送验证码的形式完成。

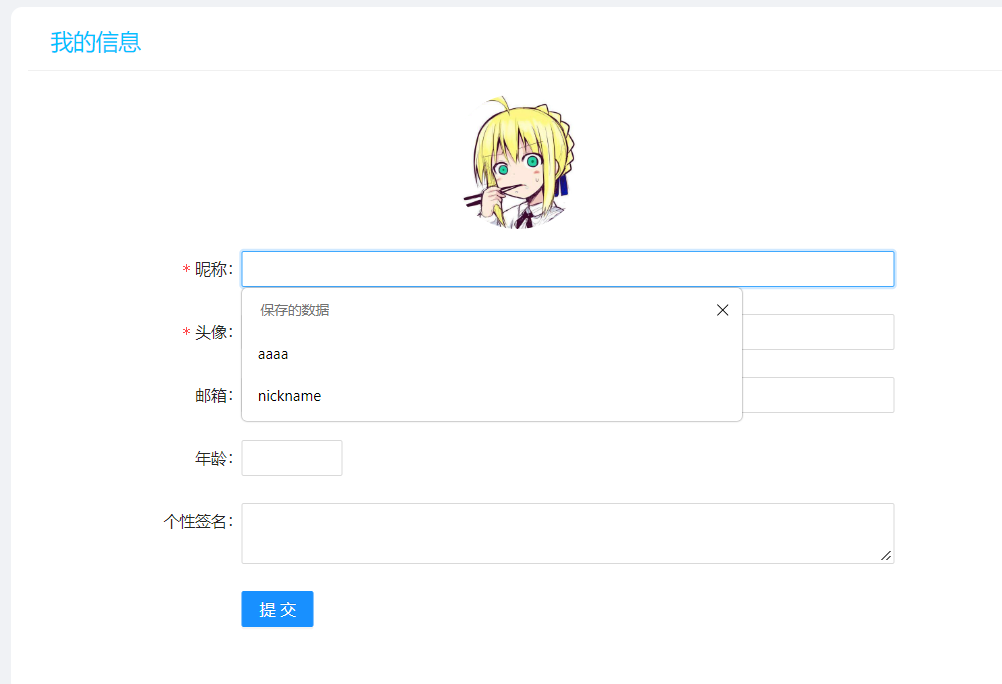


登录成功。

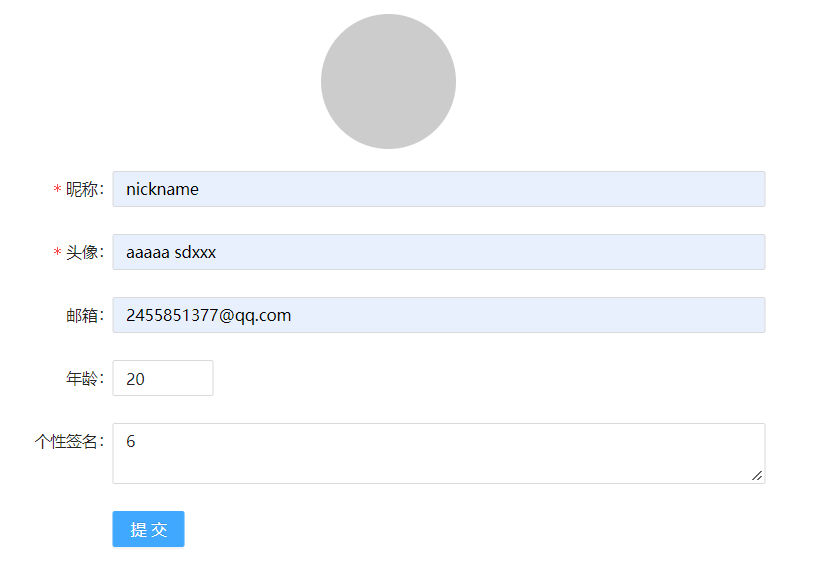


### 3.2.3 用户个人信息更新功能

填写用户信息更新表单：



更新成功：



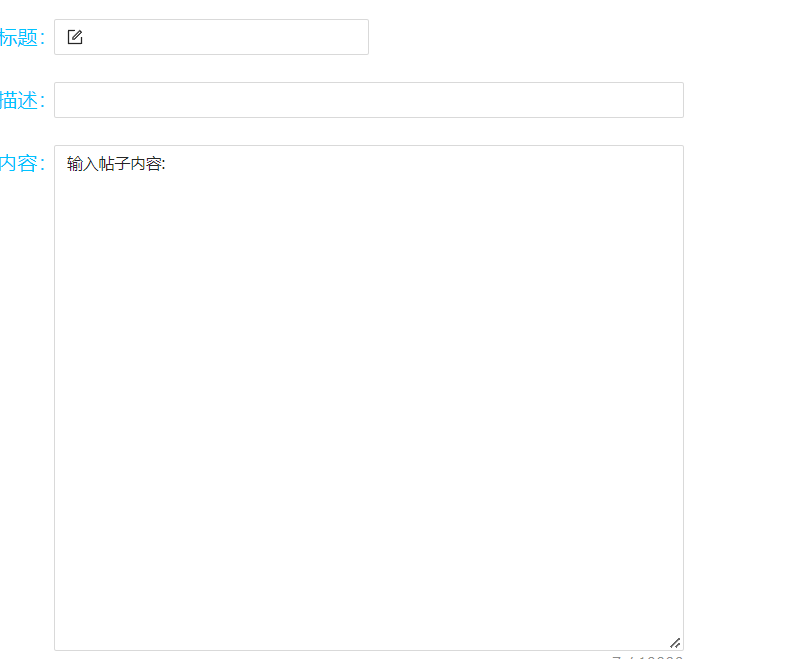
### 3.2.4 文章发布功能

多个发布界面入口：

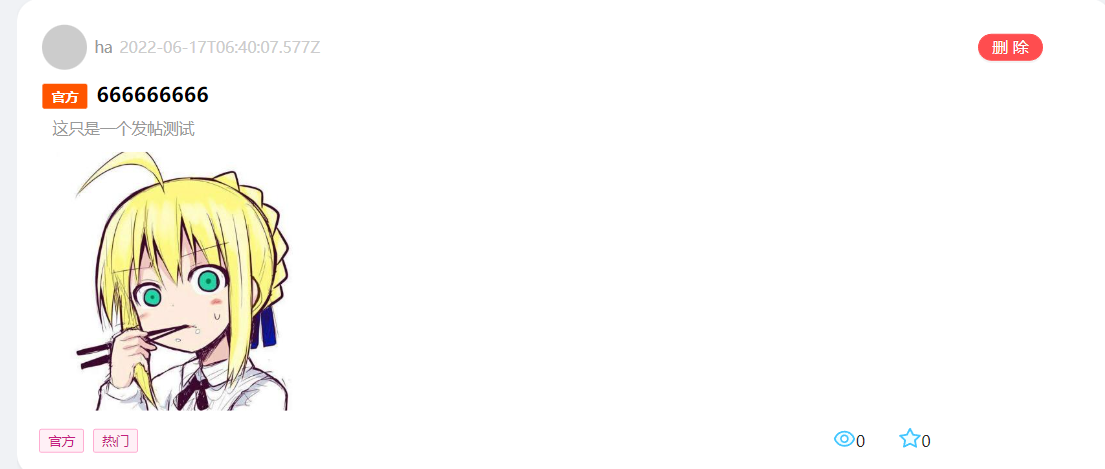




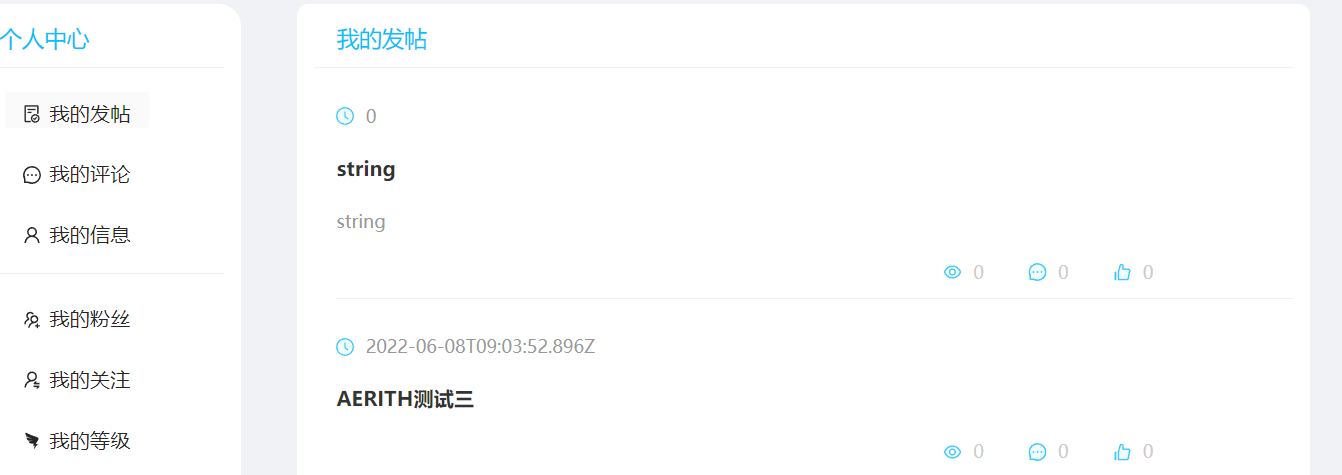
填写文章发布表单：



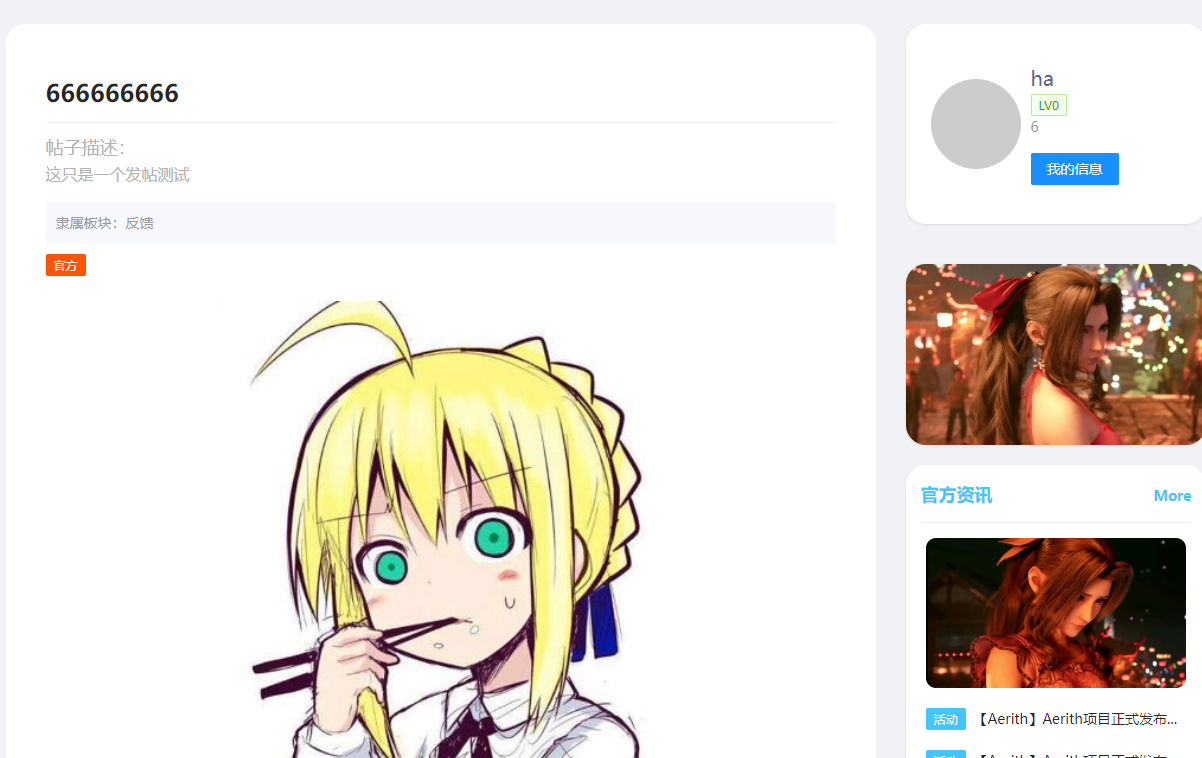
发布成功后模板页面、主页显示



发布成功后我的发帖界面显示：

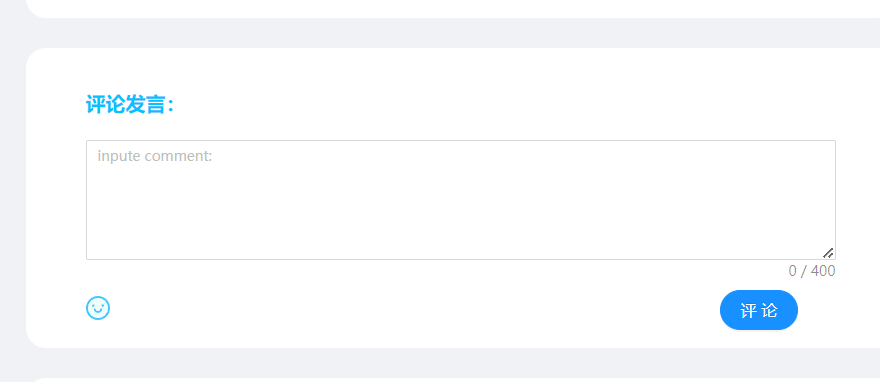


文章详情内容渲染：

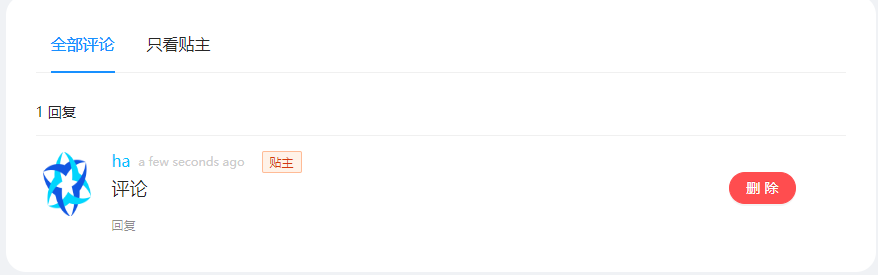


### 3.2.5 文章评论功能

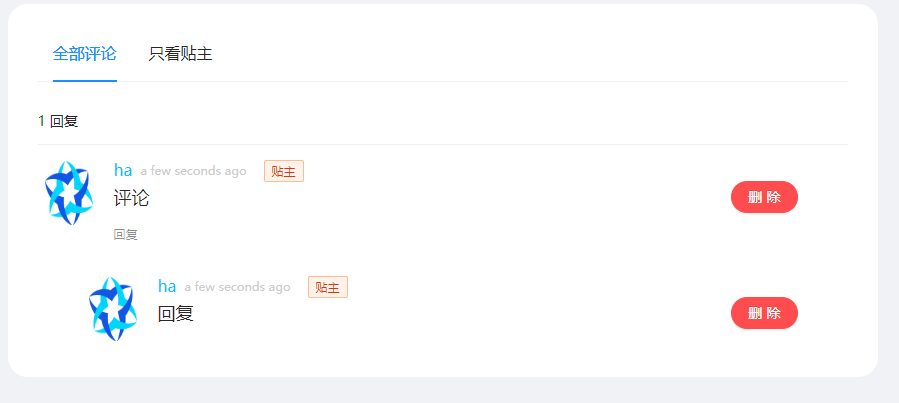
文章详情下评论表单填写：



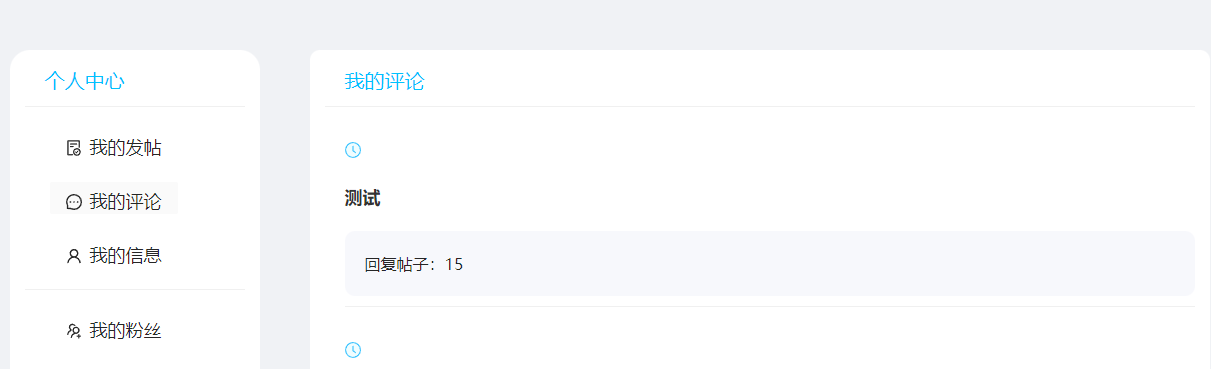
文章详情下评论显示：



评论评论，生成二级评论：



我的信息中我的评论查询历史评论：



### 3.2.6 用户关注功能

多个用户关注入口：



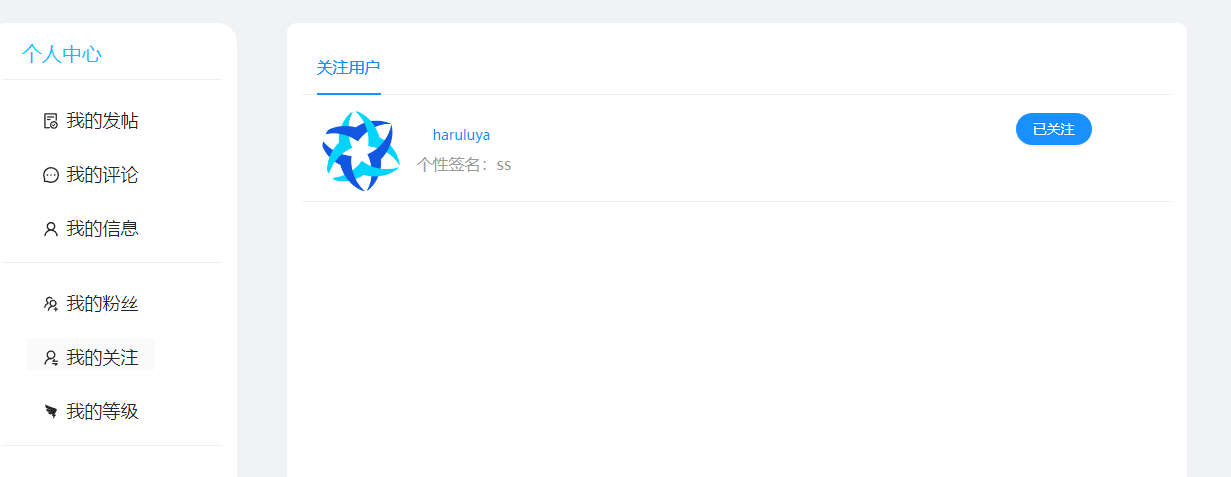




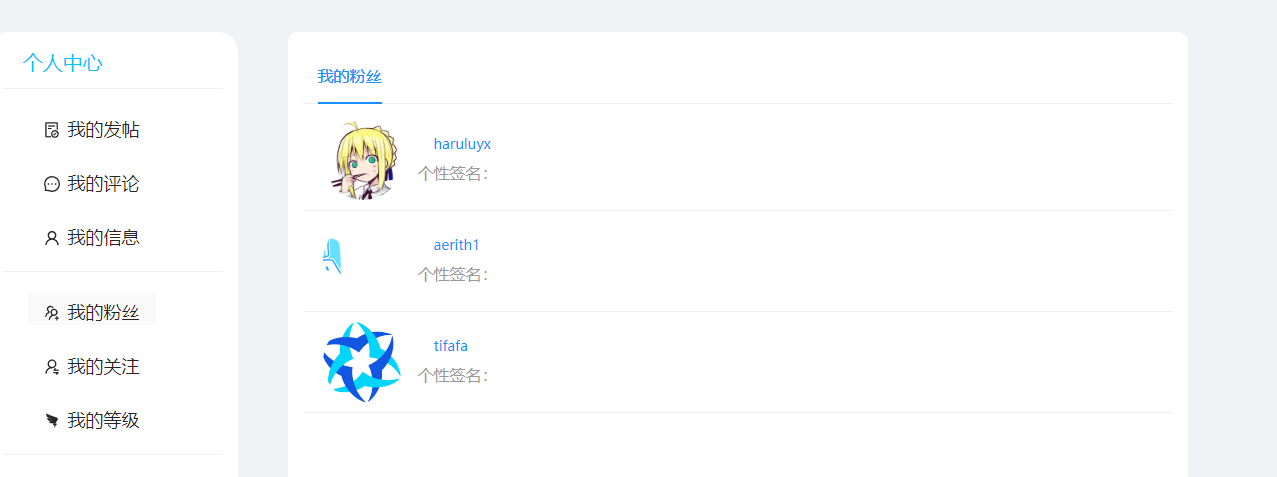
用户关注：



关注后在我的关注中查询：



他人关注自己后在我的粉丝中查询：



# 四、总结

## **4.1功能总结**

本项目以【游戏论坛】为题，基于先进的web框架Umijs、Nestjs严格的Typescript语法和规范、利用Swagger规范和可视化后端接口文档、以整洁统一的UI界面实现了完整的用户登录、用户注册、用户管理、文章浏览、文章搜索、板块分类、标签查找、Markdown文字渲染、文章发布、文章评论、用户关注、个人信息管理、浏览量统计，最新、最热文章统计等功能。

## **4.2不足与收获**

鉴于时间限制，对于【游戏论坛】的选题来说，本项目功能并非完善，无法作为一个完整的上线项目。但作为数据库课程设计，主要实践价值在于对数据库、当前火热的前端、后端等知识的巩固和实践，大型软件的实现能力的提高。本项目对于学生来说意义非凡，首先提供了学生独立实现大型项目，将框架理论付诸实现的机会。同时也让学生充分理解，当今Web技术的发展程度和发展方向、发展价值。从框架上来说，本项目完全抛弃传统JS+Java的前后端分离模式，采用Typescript+Nodejs实现，在功能实现上能够几乎完全地代替前者，揭示了未来Web前后端的多样化。从项目架构来看，此项目虽然作为“前后端分离”实现，但由于时间和项目大小原因，并没有体现出前后端分离的优势。这提醒者我，在未来进一步的Web学习方向上，多样化的框架并非是有能力者需要重点投入的方向，界面的美化和特效的叠加能让开发者更加有成就感，但关于数据库设计、中间件和多并发、分布式、数据流等知识对我来说才是至关重要的。

本项目从开题以来历经坎坷，最终谢幕。它也许不会是我做的最好、最用心、最长久的项目之一，但作为大学的一个里程碑项目，我为我从赋予名称【Aerith】开始独立一步一步最终完工演示而感到莫大的成就满足。