

设计文档

一、数据库

本项目中，为实现功能一共在数据库中创建了 11 个表，分别为：

1. blog
2. comment
3. like
4. migration
5. news
6. news_comments
7. news_likes
8. user
9. video
10. video_comment
11. video_like

接下来我会逐个分析这些表的结构及属性。

1. blog 表

目的：存储博客文章的相关信息。

属性：

id：博客文章的唯一标识符，自动递增。

title：博客的标题，最大长度为 255 个字符，使用 UTF-8 编码。

content：博客的内容，使用 TEXT 类型，支持多种字符集。

author_id：作者的 ID，外键，引用 user 表的 id。

created_at：创建时间，默认为当前时间戳。

updated_at：更新时间，默认为当前时间戳，且每次更新时自动更新为当前时间。

likes_count：博客被点赞的次数，默认为 0。

索引：

主键：id。

索引：author_id。

外键：author_id 引用 user(id)，删除时限制 (RESTRICT)，更新时限制 (RESTRICT)。

2. comment 表

目的：存储博客的评论。

属性：

id：评论的唯一标识符，自动递增。

blog_id：博客文章的 ID，外键，引用 blog(id)。

user_id：评论者的 ID，外键，引用 user(id)。

content：评论的内容。

created_at: 评论的时间, 默认为当前时间戳。

索引:

主键: id。

索引: blog_id、user_id。

外键: blog_id 引用 blog(id), user_id 引用 user(id), 删除时限制 (RESTRICT), 更新时限制 (RESTRICT)。

3. like 表

目的: 存储用户对博客的点赞记录。

属性:

id: 点赞记录的唯一标识符, 自动递增。

blog_id: 博客的 ID, 外键, 引用 blog(id)。

user_id: 用户的 ID, 外键, 引用 user(id)。

created_at: 点赞的时间, 默认为当前时间戳。

索引:

主键: id。

唯一索引: blog_id 和 user_id 组合。

索引: user_id。

外键: blog_id 引用 blog(id), user_id 引用 user(id), 删除时限制 (RESTRICT), 更新时限制 (RESTRICT)。

4. migration 表

目的: 记录数据库的迁移历史。

属性:

version: 迁移版本号。

apply_time: 迁移应用的时间戳。

5. news 表

目的: 存储新闻信息。

属性:

id: 新闻的唯一标识符, 自动递增。

title: 新闻的标题。

created_at: 新闻发布的时间, 默认为当前时间戳。

url: 新闻的 URL 链接。

索引:

主键: id。

6. news_comments 表

目的: 存储新闻的评论。

属性:

id: 评论的唯一标识符, 自动递增。

news_id: 新闻的 ID, 外键, 引用 news(id)。

content: 评论的内容。

created_at: 评论的时间, 默认为当前时间戳。

索引:

主键: id。

索引: news_id。

外键: news_id 引用 news(id), 删除时级联删除 (CASCADE), 更新

时限制 (RESTRICT)。

7. news_likes 表

目的: 存储新闻的点赞记录。

属性:

id: 点赞记录的唯一标识符, 自动递增。

news_id: 新闻的 ID, 外键, 引用 news(id)。

user_id: 用户的 ID, 外键, 引用 user(id)。

created_at: 点赞的时间, 默认为当前时间戳。

- 索引:

主键: id。

唯一索引: news_id 和 user_id 组合。

索引: user_id。

外键: news_id 引用 news(id), user_id 引用 user(id), 删除时级联删除 (CASCADE), 更新时限制 (RESTRICT)。

8. user 表

目的: 存储用户信息。

属性:

id: 用户的唯一标识符, 自动递增。

username: 用户名, 唯一。

auth_key: 认证密钥。

password_hash: 密码的哈希值。

password_reset_token: 密码重置令牌, 默认为空。

email: 电子邮件, 唯一。

status: 用户状态 (例如, 激活、禁用等)。

created_at: 账户创建时间。

updated_at: 账户更新时间。

verification_token: 账户验证令牌, 默认为空。

索引:

主键: id。

唯一索引: username、email、password_reset_token。

9. video 表

目的: 存储视频信息。

属性:

id: 视频的唯一标识符, 自动递增。

title: 视频的标题。

description: 视频的描述, 可为空。

url: 视频的 URL 链接。

cover_image: 视频封面图片的 URL, 可为空。

created_at: 视频上传时间, 默认为当前时间戳。

索引:

主键: id。

10. video_comment 表

目的: 存储视频的评论。

属性:

id: 评论的唯一标识符, 自动递增。
video_id: 视频的 ID, 外键, 引用 video(id)。
user_id: 评论者的 ID, 外键, 引用 user(id)。
content: 评论内容。
created_at: 评论时间, 默认为当前时间戳。

索引:

主键: id。

索引: video_id、user_id。

外键: video_id 引用 video(id), user_id 引用 user(id), 删除时限制 (RESTRICT), 更新时限制 (RESTRICT)。

11. video_like 表

目的: 存储视频的点赞记录。

属性:

id: 点赞记录的唯一标识符, 自动递增。
video_id: 视频的 ID, 外键, 引用 video(id)。
user_id: 用户的 ID, 外键, 引用 user(id)。
created_at: 点赞时间, 默认为当前时间戳。

索引:

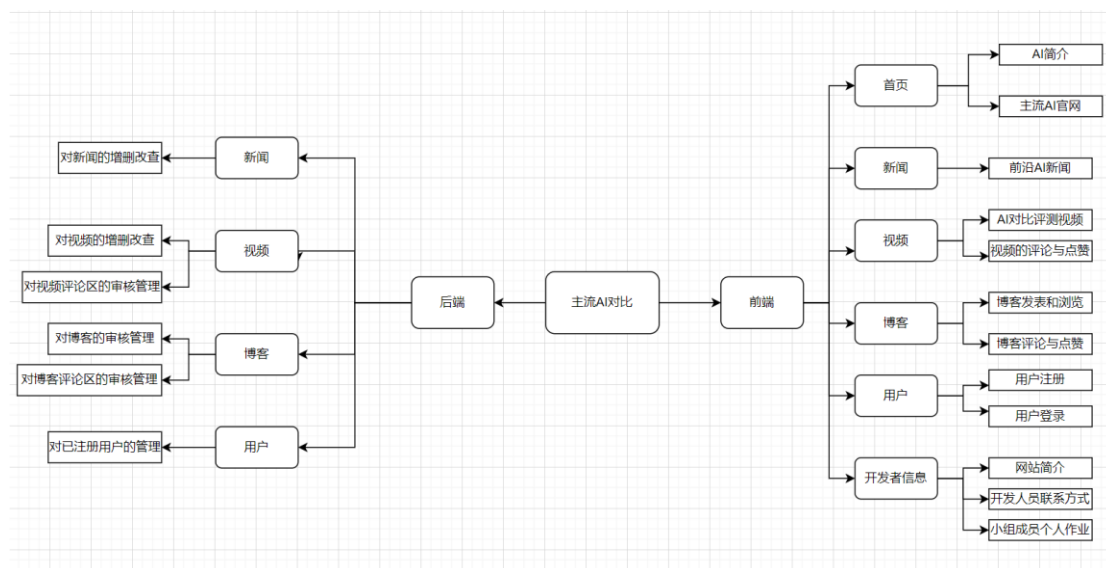
主键: id。

唯一索引: video_id 和 user_id 组合。

索引: user_id。

外键: video_id 引用 video(id), user_id 引用 user(id), 删除时限制 (RESTRICT), 更新时限制 (RESTRICT)。

二、模块图



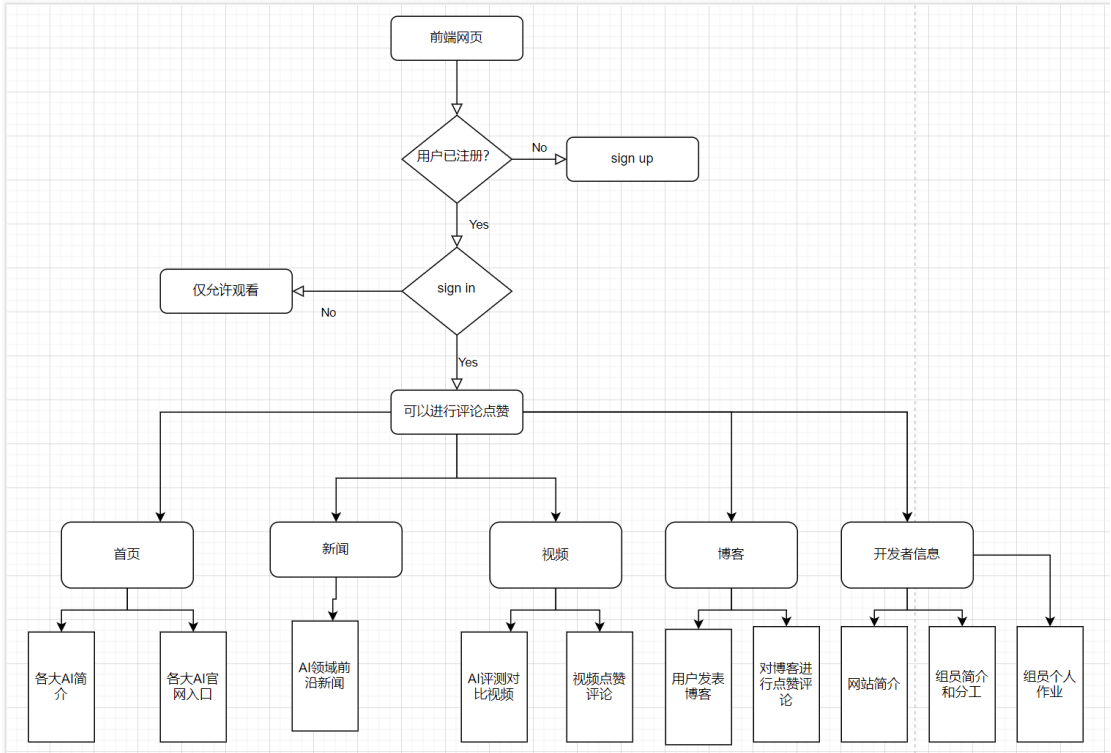
本项目主要分为两个大模块:

前端功能, 主要面向用户提供信息展示和交互功能。前端包括以下模块: 首页模块提供 AI 简介和主流 AI 官网链接; 新闻模块用于展示前沿 AI 新闻; 视频

模块提供 AI 对比评测视频，支持用户评论与点赞；博客模块允许用户发表和浏览博客，并支持评论与点赞功能；用户模块实现用户注册和登录；开发者信息模块展示网站简介、开发者联系方式和小组成员的个人作业。

后端功能主要负责网站内容的管理与审核。新闻模块支持新闻内容的增删改查；视频模块负责管理视频内容，并对视频评论区进行审核；博客模块实现对博客内容及其评论区的审核与管理；用户模块负责管理已注册用户信息。这些后端功能与前端展示协同运作，确保网站内容安全、规范，为用户提供流畅的交互体验。

三、流程图



用户首先进入前端网页，系统会判断用户是否已注册账号。如果用户未注册，则需进行注册（Sign up）；若已注册，则可通过登录（Sign in）进入网站。若用户未登录，则仅允许其浏览内容；登录后，用户不仅可以浏览网站内容，还可进行评论与点赞等交互操作。

网站内容包括多个模块：首页展示各类 AI 简介及官网入口；新闻模块提供 AI 领域的前沿资讯；视频模块展示 AI 评测对比视频，并支持视频点赞与评论；博客模块允许用户发表博客并参与评论互动；开发者信息模块包含网站简介、开发团队介绍及小组成员的个人作业。