

iryan-blog 前后端系统开发过程记录

文档

项目名称：iryan - i see u 个人博客系统

开发周期：2026 年 7 月 — 2026 年 8 月

开发工具：Trae (AI 辅助编程)

技术栈：React 19 + TypeScript + Vite + Tailwind CSS 4 / Node.js + Express + Sequelize / MySQL

一、项目概述

1.1 项目背景

本项目使用 Trae AI 编程工具，通过 Vibe Coding 方式开发一个功能完整、安全可靠的个人博客系统 "iryan - i see u"。项目采用前后端分离的 monorepo 架构，覆盖用户认证、内容管理、交互功能、管理后台等完整功能模块，并成功部署至生产服务器（域名 iryan.icu）。

1.2 功能模块

1. 用户认证系统：注册（含邮箱验证码）、登录（邮箱/用户名）、JWT 双令牌、个人资料、头像上传、修改密码、忘记密码
2. 内容管理系统：Markdown 编辑器（含 Word 导入）、文章 CRUD、分类与标签、搜索、定时发布、自定义链接（slug）、自动摘要
3. 交互功能：评论（含匿名与游客审核）、点赞、用户主页、作者主页
4. 创作中心：作者/管理员专属，文章管理、发布、编辑
5. 管理后台：用户/文章/分类/评论/标签/社交链接管理、主页板块设置、SMTP 配置、邮件模板定制、平台链接配置
6. 首页设计：导航菜单、搜索、深色模式、主题色定制、社交链接、平台链接板块、页面切换动画

1.3 技术栈

层次	技术选型	说明
前端	React 19 + TypeScript + Vite + Tailwind CSS 4	Zustand 状态管理、React Router 6、Axios、react-markdown、mammoth
后端	Node.js + Express + TypeScript + Sequelize	JWT、bcrypt、nodemailer、multer、express-validator、helmet
数据库	MySQL 5.7	Sequelize ORM，参数化查询防 SQL 注入
部署	Nginx + PM2	反向代理、静态文件服务、进程管理，端口 3002

二、开发流程总览

项目按照“需求分析 → 规格设计 → 项目初始化 → 数据库设计 → 后端模块 → 前端模块 → 安全加固 → 部署上线 → 功能完善”的流程推进，共经历 13 个核心任务与多轮迭代优化。

阶段	主要内容	状态
需求分析与规格设计	使用 /spec 命令创建 spec.md、tasks.md、checklist.md，明确技术栈与 10 个核心任务，后补充至 13 个任务	完成
项目初始化	创建 monorepo 结构（client/server/shared），配置依赖与 Git	完成
数据库设计	11 个模型文件、关联关系、同步脚本	完成
后端模块开发	认证、文章、分类、标签、评论、点赞、用户、管理员、首页等 13 个路由模块	完成
前端模块开发	14 个页面 + 20+ 组件，含认证 UI、内容管理 UI、管理后台	完成
安全加固	JWT、bcrypt、helmet、速率限制、输入清理、CORS、API 文档、部署文档	完成
部署上线	宝塔面板 + Nginx + PM2，域名 iryan.icu，端口 3002	完成
功能完善与优化	分页、定时发布、图片压缩、评论审核、作者主页、编辑器升级、主题定制等	完成

三、关键 Prompt 记录

3.1 需求分析与规格设计阶段

序号	Prompt 要点	实现结果
1	使用 /spec 命令开发个人博客系统，明确技术栈、核心功能模块、UI/UX、安全、交付物要求	生成 spec.md、tasks.md、checklist.md 三份规格文档
2	补充首页设计（导航/搜索/深色模式/社交链接/版权）、文档编辑器（Markdown/表格/图片/附件）、管理员功能需求	规格文档扩展至 13 个任务、16 项检查点

3.2 功能开发与升级阶段

序号	Prompt 要点	实现结果
3	文章列表分页功能升级：页码导航、每页 10 篇、边界条件、加载状态、响应式	Pagination 组件升级 + ArticleListPage 优化
4	管理员修改用户身份不生效问题排查与修复	修复 handleSave 闭包与错误提示
5	修复作者无法编辑/删除自己文章、保存提示“保存失败”但已写入的问题；优化创作中心；编辑器纯文本/Markdown 切换	权限检查增加 isOwner、错误信息提取、创作中心入口
6	图片压缩（超 600KB 自动压缩至 200-500KB）、图片上传显示修复、定时发布功能	compressArticleImage、URL 路径修复、scheduled 状态 + 定时任务
7	修复创作中心与列表时间显示不一致；无封面自动提取内容首图；编辑器封面设置	published_at 统一、extractFirstImage、封面 UI
8	时区问题、评论权限与匿名评论、封面 UI 紧凑化、纯文本模式实现	统一北京时区、is_anonymous、纯文本模式隐藏预览
9	评论功能、匿名评论、封面尺寸、图片显示、时间显示 5 个问题逐一修复	登录用户评论默认 approved、helmet CORP 修复、formatDate 统一
10	文章访问权限统一为公开，移除详情页登录拦截	like status 接口改用 optionalAuth

序号	Prompt 要点	实现结果
11	新文章无法保存、发布日期修改失效；编辑器升级（下划线、颜色、对齐、快捷键）	category_id 支持 null、编辑器 12 项功能升级
12	文章列表“最新排序”模式文章不全、分页不显示问题	排序切换重置页码 + URL 同步
13	创作中心优化：数据同步、文章链接（slug）、自动摘要、页面权限、动画开关	自动保存、migrate-slugs 脚本、extractExcerpt、路由守卫、AnimationToggle
14	首页新增平台链接板块（升级“关注我”模块）	PlatformLinks 组件 + 后台配置
15	评论逻辑：注册用户默认通过审核、游客待审核；社交链接验证	游客评论 pending + 审核提示
16	整合“关注我”与社交平台链接为统一配置模块；全面审查文档	/api/links 统一接口、文档重写
17	更新首页与底部平台链接样式	数据库新增多个平台链接、PlatformLinks 6 列网格
18	底部样式优化、关于页面更新、首页标语更新	Footer 重设计、联系邮箱 ryan9264@qq.com、新标语

3.3 系统维护与安全阶段

序号	Prompt 要点	实现结果
19	账户安全检查、登录错误提示优化、文章显示异常排查	密码重置 + 审计日志、axios 拦截器修复、Cloudflare 缓存处理
20	批量删除测试文章、示例文章标识、分类标签整理、权限功能测试	删除 10 篇测试文章、6 篇示例文章加 [示例] 前缀、4 分类 19 标签
21	个人中心修改密码/资料、管理员重置密码功能	密码强度检测、确认机制、临时密码 + 日志 + 邮件通知
22	邮箱验证码注册功能：发送验证码、验证后注册、频率限制	send-register-code、15 分钟有效期、一次性验证码

序号	Prompt 要点	实现结果
23	邮件模板保存失败修复并预设模板	创建 email_templates 表、预设 3 个模板
24	测试邮件成功但注册邮件失败问题排查	SMTP 配置来源统一为数据库 settings 表

四、AI 交互日志

4.1 需求分析与规格设计

1. 我通过 /spec 命令下达完整需求，Trae 生成三份规格文档并确认技术栈选型 (React + Express + MySQL)
2. 补充首页设计、文档编辑器、管理员功能需求，Trae 将任务从 10 个扩展至 13 个，新增 Attachments、Home_Section、Social_Links 表
3. 规格文档获批准后，Trae 创建任务跟踪并进入开发阶段

4.2 项目初始化与数据库设计

1. Trae 创建 monorepo 项目结构 (client/server/shared)，安装前端 (React + Vite + Tailwind) 与后端 (Express + Sequelize) 依赖
2. 创建 11 个模型文件 (User、Article、Category、Tag、Comment、Like 等)，定义关联关系与唯一联合索引
3. TypeScript 编译与 Vite 构建验证通过，任务 1、2 完成

4.3 后端模块开发

1. 认证模块：注册、登录、JWT 双令牌、邮箱验证、头像上传、速率限制
2. 内容模块：文章 CRUD、分类、标签、附件、图片上传 (10MB、SVG 支持)
3. 交互模块：评论 (审核)、点赞 (切换/状态/计数)、用户资料与活动
4. 管理模块：用户/文章/分类管理、首页板块与社交链接管理
5. 多次处理 Express 5 的 req.params 类型兼容问题，通过类型注解与 tsx 运行方式解决

4.4 前端模块开发

1. 搭建路由与布局（Header、Footer、MainLayout），实现深色模式与响应式菜单
2. 开发认证 UI（登录、注册、个人资料）、内容管理 UI（编辑器、文章列表/详情）、管理后台页面
3. 创建通用组件（Modal、DataTable、StatusBadge、ErrorBoundary、Skeleton 等）
4. 使用 mock 数据先行开发，后续统一替换为真实 API

4.5 安全加固与文档

1. 配置 helmet 安全头（CSP、HSTS、XSS 过滤）、CORS、输入清理中间件（XSS 防护）
2. 集中管理速率限制器（全局 100/min、登录 10/min、注册 5/h、验证码 1/5min）
3. 创建 ErrorBoundary、Loading、Skeleton 组件，图片懒加载与 React.memo 性能优化
4. 编写 API.md、DEPLOY.md、README.md 等完整文档

4.6 部署上线

1. 在宝塔面板部署，处理目录权限（sudo）、TypeScript 类型错误、前端构建问题
2. 发现端口 3001 被其他项目占用，与用户确认后切换到 3002 端口
3. 配置 Nginx 反向代理（iryan.icu 域名）、PM2 进程管理（iryan-blog）
4. 修复 Nginx 正则 location 优先级导致头像无法访问的问题

4.7 功能完善与迭代优化

1. 文章列表分页、定时发布、图片压缩、封面图提取、评论审核、作者主页等功能迭代
2. 编辑器升级：Word 导入（mammoth）、文本样式、对齐方式、快捷键
3. 主题定制：8 种预设主题色、深色模式、动画开关

4. 平台链接板块、社交链接整合、6 平台配置

5. 多轮 bug 修复与数据一致性校验，最终通过 16/16 项检查点验证

五、遇到的问题及解决方案

5.1 认证与权限类

问题	根因	解决方案
注册后无法登录	注册接口不返回 JWT token，前端解构失败	注册时自动生成并返回 token，注册后直接登录
密码验证规则前后端不一致	前端最少 6 位，后端最少 8 位且需含字母数字	统一为最少 6 位，与前端一致
管理员无法登录	数据库密码哈希被测试覆盖，与用户密码不匹配	更新 bcrypt 哈希为正确密码
作者无法编辑/删除自己文章	文章详情路由对非公开文章仅允许管理员查看	权限检查增加 isOwner 判断，作者可查看自己的草稿
未登录访问文章详情被重定向登录页	点赞状态接口使用 authenticate 返回 401，触发拦截器跳转	改用 optionalAuth，未登录返回 liked:false
用户管理缺少“作者”角色	前后端多处角色类型限制为 user/admin	6 处文件添加 author 角色支持
登录失败页面刷新	axios 拦截器对登录 API 的 401 也做页面跳转	拦截器排除登录 API 的 401 跳转
密码重置提示“请求参数验证失败”	前后端密码特殊字符正则不一致	统一使用 [^A-Za-z0-9] 正则

5.2 数据与接口类

问题	根因	解决方案
保存文章提示“保存失败”但实际已写入	前端未校验返回值且错误信息固定	增加返回数据验证与后端错误信息提取
新文章无法保存	category_id 验证器不接受 null 值	验证器改为 optional({ values: 'null' })

问题	根因	解决方案
发布日期修改失效	前端清除后发送 undefined 被 JSON 丢弃	前端发送 null，后端验证器支持 null
文章列表排序切换后文章不全/分页不显示	切换排序未重置页码，页码停留在空页	排序按钮重置页码 + URL 参数同步
首页与文章页数据不一致	各页面数据加载逻辑与分页/状态筛选不一致	统一服务端分页与 scheduled 状态筛选
分类篇数统计显示 0 篇	前端硬编码 article_count: 0 覆盖真实数据	移除硬编码，直接使用后端返回数据
浏览计数不显示真实数据	Sequelize increment() 不更新内存对象	increment 后调用 reload() 返回最新计数
点赞数据不持久化	likes 表与 articles.like_count 数据不一致	同步 SQL 对齐计数，toggleLike 后 reload()
评论功能无法使用	评论默认状态 pending，查询只返回 approved	默认状态改为 approved，提交后立即可见
API 响应格式不匹配导致数据加载失败	前端用 ApiResponse 包装解析，后端返回原始数据	移除 ApiResponse 包装，匹配后端真实格式
邮件模板保存失败	email_templates 表未在数据库创建	创建表并预设 3 个邮件模板
注册邮件无法发送	email.ts 读取 .env 示例值而非数据库配置	统一从数据库 settings 表读取 SMTP 配置

5.3 部署与环境类

问题	根因	解决方案
端口 3001 被占用	其他项目（BlogSystem/PromptB）占用	切换到 3002 端口
后台报错无法访问	express-rate-limit 检测到 X-Forwarded-For 但未启用 trust proxy	添加 app.set('trust proxy', 1)
头像无法通过 Nginx 访问	正则 location 优先级高于前缀匹配 /uploads/	添加 ^~ 修饰符使前缀匹配优先

问题	根因	解决方案
前端加载旧版本	Nginx expires 30d 缓存 index.html	调整缓存配置，开启 Cloudflare 开发模式
登录 API 返回 500	数据库 users 表缺少 theme_color 列	添加 theme_color 列
目录权限不足	服务器目录权限限制	使用 sudo 权限修复目录与文件权限

5.4 前端显示类

问题	根因	解决方案
深色模式不生效	Tailwind v4 dark: 变体默认依赖媒体查询而非 .dark 类	添加 @custom-variant dark 指令
代码块白色文字不可见	highlight.js 白色背景覆盖代码块深色背景	添加 .prose pre code.hljs 背景透明规则
板块内容显示反引号符号	内容预览直接渲染原始 Markdown 文本	创建 MarkdownContent 组件渲染内联代码
文章卡片单列显示	StaggerList 外层 div 破坏 CSS Grid 布局	重构使用 cloneElement，不增加额外 DOM 节点
首页空白	App.tsx 使用 ToastProvider 但未导入	添加缺失的 import 语句
文章图片无法显示	helmet 配置 CORP/COEP 阻止跨域资源加载	crossOriginResourcePolicy 改为 cross-origin
图片 URL 双重重叠	URL 生成兼容 ./uploads 与 uploads 两种路径	修复路径拼接逻辑

5.5 其他问题

问题	根因	解决方案
图片上传后占位符无法替换	async 回调中 content 变量过时	使用 contentRef 实时追踪最新内容
发布时间时区偏差	前后端时区处理不一致	统一 UTC+8 时区，前端做 UTC/本地转换
旧头像占用存储	更换头像后旧文件未删除	实现四层安全防护的自动删除功能

问题	根因	解决方案
定时文章出现在公开列表	scheduled 状态未过滤	列表接口过滤 scheduled 状态

六、经验总结

1. Vibe Coding 模式下，规格文档（spec/tasks/checklist）是保证项目质量的关键，需求变更需同步更新规格文档
2. 前后端分离项目需严格统一 API 响应格式与类型定义，避免响应格式不匹配导致的数据加载失败
3. 部署环境与开发环境差异（端口占用、Nginx 配置、代理头、缓存）是常见问题源，需提前规划端口与代理配置
4. 数据库变更（新增列、新表）需同步执行迁移，否则会导致接口 500 错误
5. 安全加固应贯穿开发全程：JWT、bcrypt、helmet、速率限制、输入清理、参数化查询缺一不可
6. AI 辅助开发中，清晰的 Prompt 描述（含具体验收标准）能显著提升实现质量与迭代效率