



武汉工程科技学院
WUHAN UNIVERSITY OF ENGINEERING SCIENCE

武工院 Web 应用设计大赛

作品设计和开发文档

作品名称: 智学伴 AI 个性化学习平台

作者: 涂维轩 刘富祥 何萧 李娜 杨兴露

指导老师: 申友访

填写日期: 2025年11月28日

填写说明:

1. 本文档为简要文档，不宜长篇大论，需简明扼要，建议设计二级目录，逻辑性强；一级标题可为需求分析、概要设计、详细设计、系统测试、安装及使用、项目总结等；
2. 一级标题采用二号黑体，居中，二级标题采用三号黑体，靠左，根据需要可以设计三级标题，正文一律用五号宋体；
3. 提交文档时，以 PDF 格式提交本文档；
4. 本文档内容是正式参赛内容组成部分，务必真实填写。如不属实，将导致奖项等级降低甚至终止本作品参加比赛。

目录

第一章 需求分析	4
1.1 开发背景	4
1.2 主要功能	4
第二章 概要设计	5
2.1 该系统的层次图	5
2.2.1 前端层	7
2.2.2 API 网关层	7
2.2.3 业务服务层	7
2.2.4 数据访问与存储层	7
第三章 详细设计	8
3.1 界面设计	8
图 3.1.1 PC 端登录界面	8
图 3.1.2 移动端登录界面	9
图 3.1.3 PC 端首页界面	10
图 3.1.4 AI 助手界面	10
图 3.1.5 学习计划界面	10
图 3.1.6 AI 测评界面	11
图 3.1.7 知识图谱界面	11
图 3.1.8 管理后台界面	11
3.2 数据库设计	11
表 3.2.1 用户表	11
表 3.2.2 提示词表	12
表 3.2.3 模型配置表	12
表 3.2.4 测验表	13
表 3.2.5 学习计划表	13
表 3.2.6 AI 会话表	13
表 3.2.7 AI 消息表	13
表 3.2.8 知识图谱文件表	13
表 3.2.9 知识图谱会话表	14
表 3.2.10 测验组卷表	14
表 3.2.11 API 请求日志表	15
第四章 测试报告	16
4.1 测试策略	16
4.2 测试概要	16
表 4.2.1 测试概要表	16
4.3 功能测试	16
4.4 测试执行情况	17
4.4.1 系统登录测试	17
4.4.2 AI 对话测试	18
4.4.3 学习计划生成测试	18
4.4.4 AI 测评智能出卷测试	18
4.4.5 知识图谱测试	20
4.4.6 管理后台测试	20
4.4.7 前后交互测试	20
4.5 测试总结	21
第五章 安装及使用	21
5.1 环境要求	21
5.2 安装教程	22
5.3 使用说明	23
第六章 项目总结	24
6.1 面对困难	26
6.2 开发感悟	26
第七章 参考文献	27

第一章 需求分析

1.1 开发背景

在国家深入实施“人工智能+”行动和教育数字化战略背景下，人工智能正成为推动教育变革的重要力量。当前，学生在学习过程中普遍存在知识结构碎片化、缺乏个性化指导、学习路径不清晰等问题，而教育资源的数字化用率仍有提升空间。

为响应国家“人工智能+教育”发展方向，解决学生学习过程中的痛点，本项目推出智学伴 3.0 AI 个性化学习平台。平台通过多模型融合、知识图谱构建、学习路径规划、智能测评等技术，为学生提供系统化、可视化、个性化的学习支持，助力构建面向未来的智能学习新模式。

面向用户需要复习课程、掌握知识点备考的大中小学学生需要个性化学习规划、内容解析的社会自主学习人士需要 AI 增强课堂教学、学生测评的教育培训机构。

1.2 主要功能

(1)**多模型 AI 支持系统**：支持国内 5 家主流大模型调用，可按需动态切换，确保服务高可用性；支持所有基于 OpenAISDK 的大模型，现已内置 DeepSeek、文心一言、星火、ChatGLM、Moonshot。

(2)**Prompt(提示词)管理系统**：可视化管理 AI 提示词模板，支持版本控制和 A/B 测试；5 分钟 TTL 缓存，提升性能；系统自动注入 Prompt 提高 AI 调用可用性；多场景支持：支持不同场景的 Prompt 模板(系统提示、出题组卷、批改)。

(3)**文件上传与学习计划生成**：用户上传学习资料、笔记 (PDF、Word、PPT、TXT、MD) 文件，AI 自动解析内容并生成个性化学习计划。

(4) **AI 自动评测系统**：AI 根据学习主题或者设置的个性化参数 (学段、考试科目、考试时间等) 自动生成测验题目甚至实现智能组卷，支持在线答题、自动批改和错题讲解。

(5) **知识图谱系统**：将学习内容转化为可视化知识卡片图谱，展示知识点之间的依赖关系，帮助学习者理解知识结构，并以此推荐学习资源。

(6) **学习可视化与成长报告**：多维度 (统计卡片、折线图、饼状图、弱项分析) 展示学习进度和成绩，生成 PDF 学习报告。

(7) **管理后台系统**：数据大屏仪表盘统计展示系统后台信息，支持模型配置、Prompt 管理、

用户管理、系统配置。

第二章 概要设计

智学伴是一个由全国大学生计算机设计大赛参赛团队开发的 AI 个性化学习助手平台。平台深度融合人工智能技术，为学习者提供个性化学习计划生成、智能测评、知识图谱可视化、学习进度分析等全方位学习支持服务。

2.1 该系统的层次图

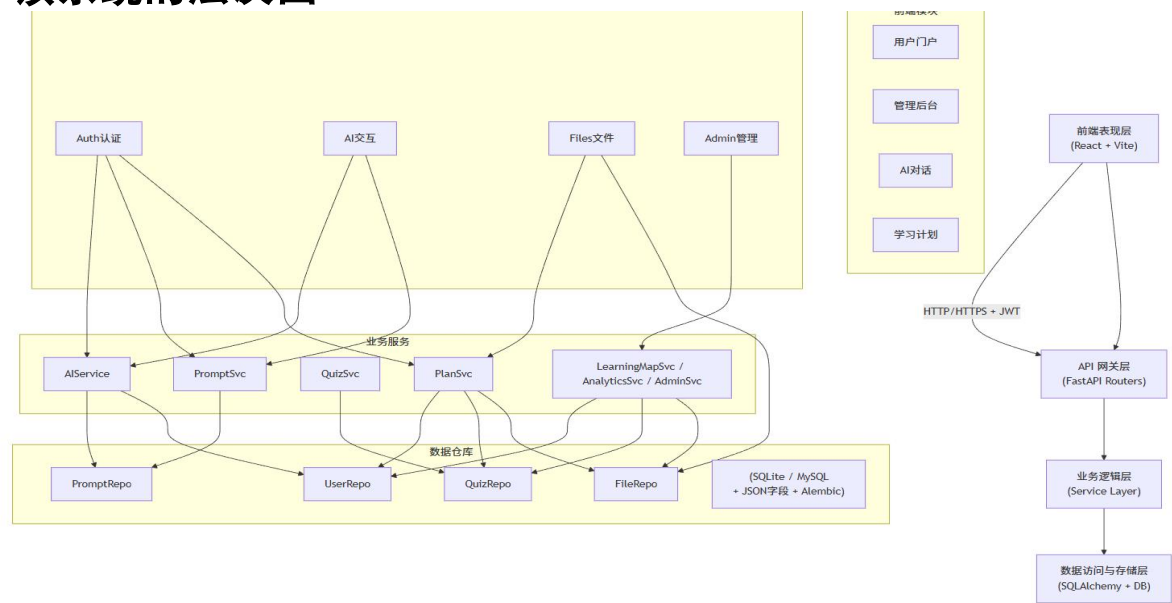


图 2.1 智学伴系统层次图

前端表现层(React + Vite)

页面结构：用户门户、管理后台、AI 对话、学习计划、知识图谱、可视化仪表盘等页面全部由 React 组件组合而成，遵循“页面 → 业务组件 → 原子组件”三段式拆分。

交互规范：统一使用 Tailwind CSS 的设计语言，搭配 Zustand 管理鉴权、当前 Persona、学习图缓存等全局状态；所有请求通过 `src/api/apiClient.js` 发起，内置 JWT 注入与错误拦截，保证端到端一致性。

可视化能力：关键指标采用 Recharts 出图，知识图谱用 `react-force-graph` 提供拖拽、缩放、右键菜单等高级交互，Markdown 内容统一由 `react-markdown + remark-gfm` 渲染并封装成 UI 组件，确保展示层风格一致。

API 网关层（FastAPI 路由）

职责定位：承担 HTTP 入口、参数验证、权限校验、请求日志以及将请求委派到对应 Service

层的职责，不承载任何业务逻辑。

说明： Auth 路由负责注册/登录/JWT 刷新； AI 路由承载 AI 对话、学习计划生成等模型能力； Files 路由处理多格式文件上传与解析； Admin 路由为后台配置（模型、Prompt、用户）提供统一入口；其余如 quiz 、 analytics 、 learning_map 、 plan 等路由均遵循“请求模型 → 依赖注入 → 调用 service → 响应模型”的流程。

业务逻辑层 (Services)

AIService: 封装所有大模型调用链路，负责注入系统 Prompt、加载模型配置、JSON 解析与校验、失败重试和 fallback 策略，同时记录调用日志并统一替换 AI 身份描述。

PromptService: 实现 Prompt 版本管理、缓存、启用/禁用控制，并对外提供 get_system_prompt 等高频接口，保证所有模块获得一致的提示语。

LearningMapService / QuizService / PlanService / AnalyticsService: 分别负责知识图谱构建、AI 测评生成与批改、学习计划生成与持久化、学习进度统计与 PDF 报告生成；每个 Service 内部拆分文件解析、AI 调用、数据清洗、存储、响应装配等子步骤，确保逻辑可测试、可复用。

AdminService / AuthService / BootstrapService: 覆盖管理员配置、权限校验、首启种子数据 等后台治理能力，结合 core/security.py 完成 JWT 签发、密码加密与角色判定。

2.2 模块之间的调用关系

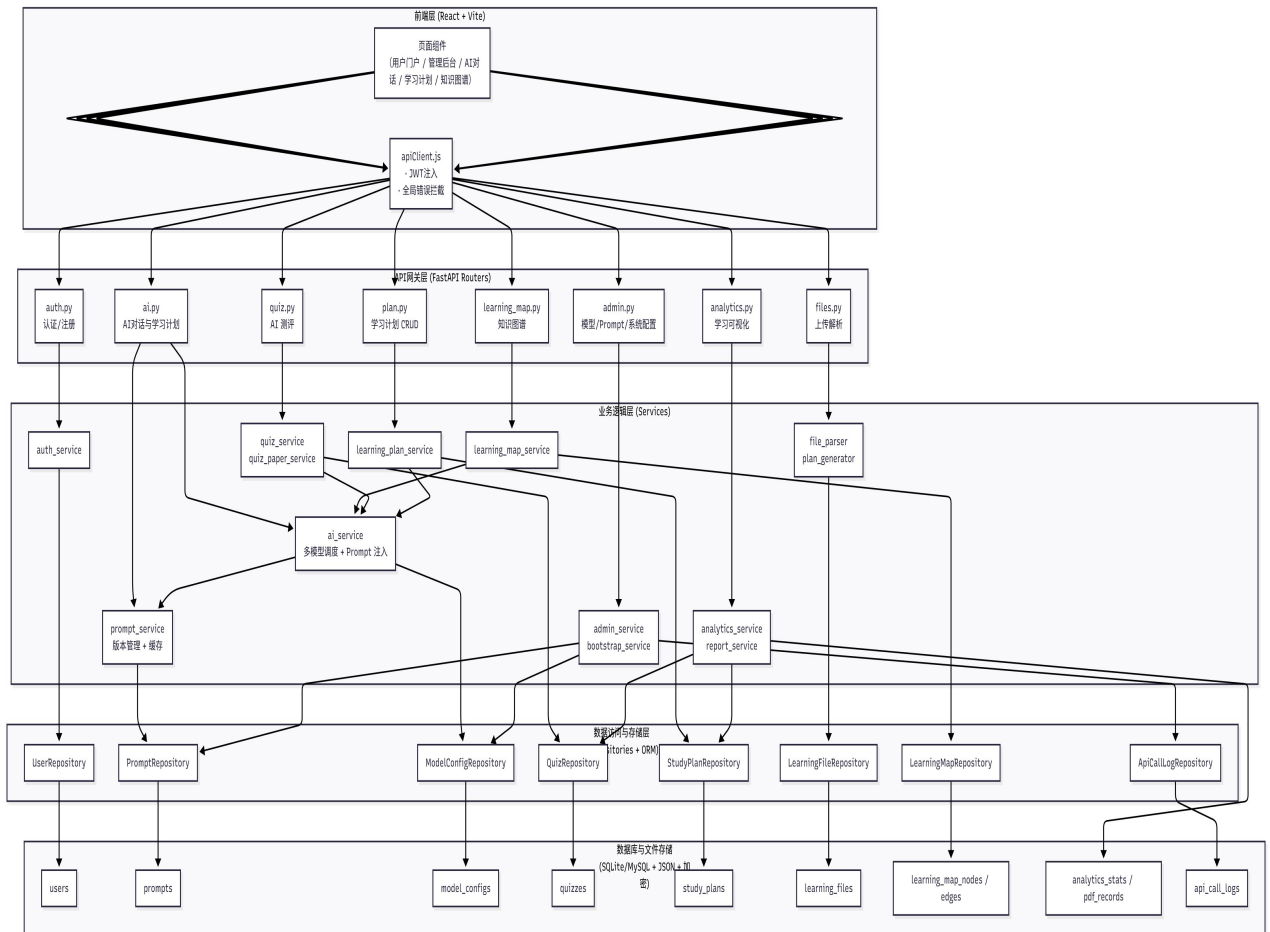


图 2.2 模块调用图

2.2.1 前端层

用户门户、管理后台、AI 对话、学习计划、知识图谱等组件通过统一的 apiClient.js 发起请求

2.2.2 API 网关层

路由文件只是入口，负责请求校验/权限检查，然后把逻辑委托给业务服务；ai.py 是跨功能路由，既可做 AI Chat，也会触发学习计划生成。

2.2.3 业务服务层

ai_service 为学习计划生成，AI 测评服务提供统一的多模型调用与 Prompt 注入。

quiz_service 搭配 quiz_paper_service，负责 AI 测评与组卷；learning_map_service 负责知识图谱；analytics_service/report_service 负责可视化、PDF 报告；admin_service/bootstrap_service 管模型配置、种子数据；prompt_service 管版本与缓存。

2.2.4 数据访问与存储层

每个 Service 都通过对应 Repository 读写数据库，比如：UserRepository → users 表、PromptRepository → prompts 表等。

第三章 详细设计

3.1 界面设计

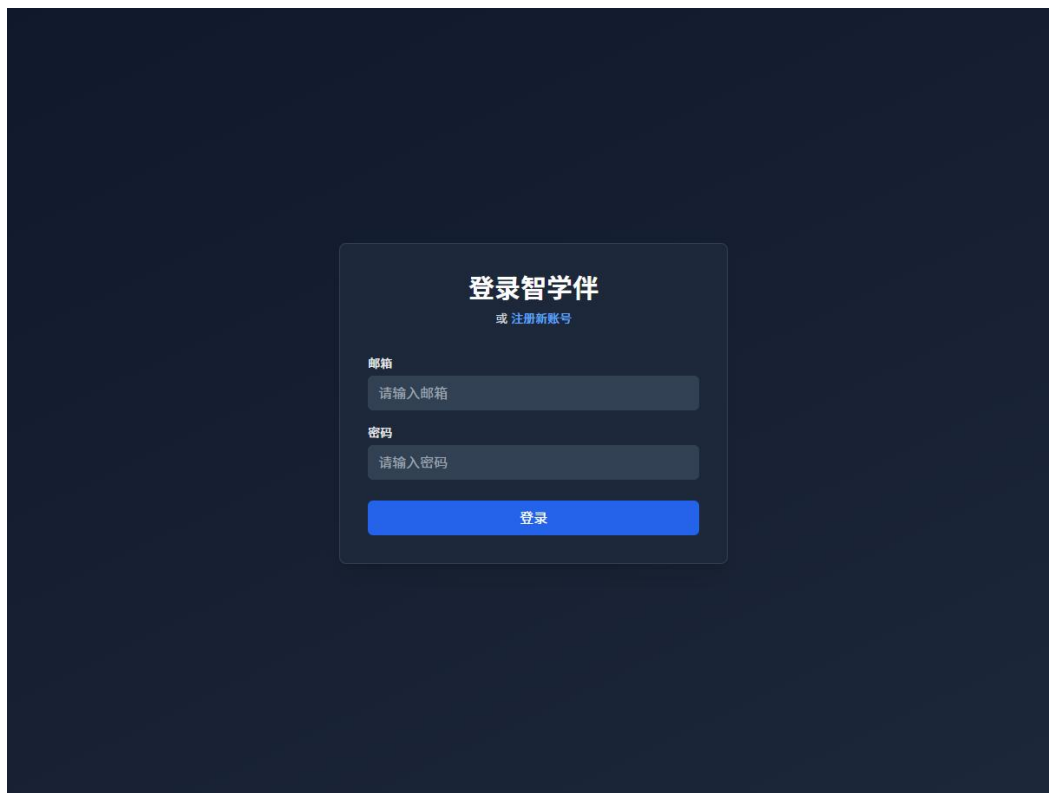


图 3.1.1 PC 端登录界面



图 3.1.2 移动端登录界面



图 3.1.3 PC 端首页界面

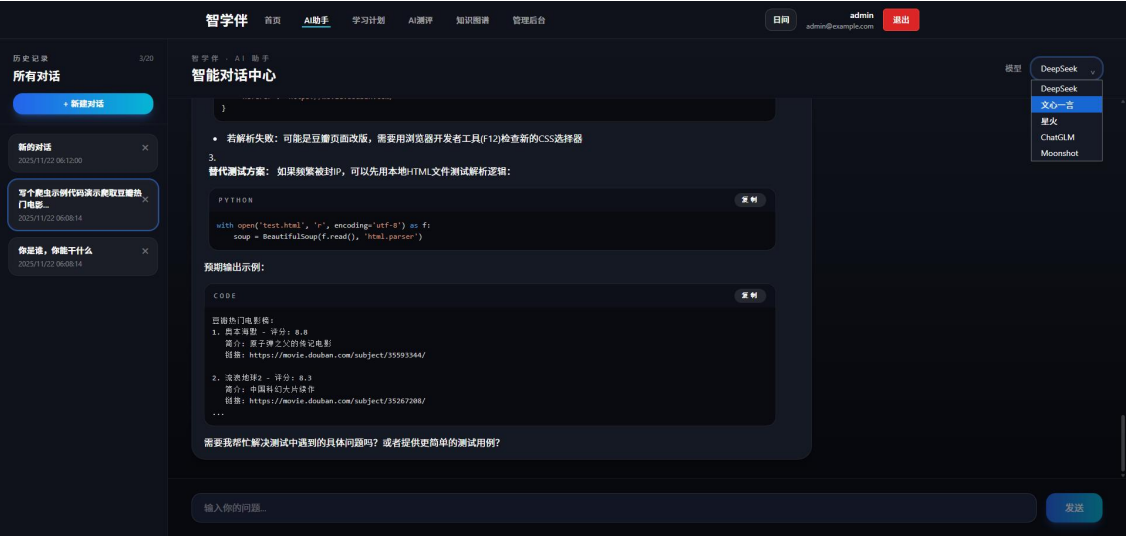


图 3.1.4 AI 助手界面



图 3.1.5 学习计划界面



图 3.1.6 AI 测评界面



图 3.1.7 知识图谱界面



图 3.1.8 管理后台界面

3.2 数据库设计

表 3.2.1 用户表

属性名	数据类型	约束条件	注释
id	Integer	主键，索引	用户 ID
email	String(255))	唯一，非空，索引	用户邮箱
name	String(100))	非空	用户姓名
hashed_password	String(255))	非空	加密密码
role	String(20)	非空，默认 user	用户角色 (admin/user)
created_at	DateTime	默认当前时间	创建时间

表 3.2.2 提示词表

属性名	数据类型	约束条件	注释
id	Integer	主键，索引	Prompt ID
name	String(100)	非空，索引	Prompt 名称
version	Integer	非空，默认 1	版本号
content	Text	非空	Prompt 内容
description	Text	可空	描述
enabled	Boolean	非空，默认 True	是否启用
author	String(100)	可空	创建者
created_at	Date Time	默认当前时间	创建时间
updated_at	Date Time	默认当前时间，更新后自动刷新	更新时间

表 3.2.3 模型配置表

属性名	数据类型	约束条件	注释
id	Integer	主键，索引	配置 ID
provider_name	String(50)	唯一，非空，索引	模型提供商
api_key	Text	可空	加密后的 API 密钥
base_url	String(50)	可空	API 基础 URL
priority	Integer	非空，默认 0	调用优先级
enabled	Booiean	非空，默认 True	是否启用
parsms	JSON	可空	其他参数
created_at	Date Time	默认当前时间	创建时间
updated_at	Date Time	默认当前时间，更新自动刷新	更新时间

表 3.2.4 测验表

属性名	数据类型	约束条件	注释
属性名	数据类型	约束条件	注释
id	Integer	主键，索引	测评 ID
user_id	Integer	非空，索引	用户 ID
topic	Text	可空	测验主题
questions	Text	非空	题目列表 (JSON)
answers	Text	非空	用户答案 (JSON)
score	Integer	非空	得分 0-100
explanations	Text	非空	讲解 (JSON)

表 3.2.5 学习计划表

属性名	数据类型	约束条件	注释
id	Integer	主键，索引	学习计划 ID
user_id	Integer	非空，索引	用户 ID
goal	Text	非空	学习目标
plan_json	Text	非空	计划内容 (JSON)
file_name	Text	可空	上传文件名
created_at	DateTime	默认当前时间	创建时间
updated_at	DateTime	默认当前时间，更新自动刷新	更新时间

表 3.2.6 AI 会话表

属性名	数据类型	约束条件	注释
id	Integer	主键，索引	会话 ID
user_id	Integer	外键→users.id，非空，索引	用户 ID
title	String(200)	非空	会话标题
created_at	DateTime	默认当前时间	创建时间
updated_at	DateTime	默认当前时间，更新自动刷新	更新时间

表 3.2.7 AI 消息表

属性名	数据类型	约束条件	注释
id	Integer	主键，索引	消息 ID
session_id	Integer	外 键 → chat_sessions.id, 非空，索引	会话 ID
role	String(20)	非空	消 息 角 色 (user/ai)
content	Text	非空	消息内容
provider	String(50)	可空	AI 提供商
created_at	DateTime	默认当前时间	创建时间

表 3.2.8 知识图谱文件表

user_id	Integer	外键→user.id, 非空, 索引	用户 ID
title	String(200)	非空	会话标题
created_at	Date Time	默认当前时间	创建时间
update_at	Date Time	默认当前时间, 更新自动刷新	更新时间

表 3.2.9 知识图谱会话表

属性名	数据类型	约束条件	注释
id	Integer	主键, 索引	图谱会话 ID
user_id	Integer	非空, 索引	用户 ID
topic	String(255)	可空	主题
provider	String(64)	可空	使用模型
file_id	Integer	外键→learning_map_files.id	来源文件
source_preview	Text	可空	文本摘录
created_at	DateTime	默认当前时间	创建时间

表 3.2.10 知识节点表

属性名	数据类型	约束条件	注释
id	Integer	主键, 索引	知识节点 ID
user_id	Integer	非空, 索引	用户 ID
session_id	Integer	外键→learning_map_sessions.id, 非空, 索引	会话 ID
file_id	Integer	外键→learning_map_files.id	文件 ID
title	String(255)	非空	知识点名称
description	Text	可空	描述
level	String(64)	可空	难度层级
mastery	String(32)	默认 unknown	掌握度
example	Text	可空	典型例题
resources	Text	可空	资源列表 (JSON)
created_at	DateTime	默认当前时间	创建时间

表 3.2.10 测验组卷表

属性名	数据类型	约束条件	注释
id	Integer	主键, 索引	试卷 ID
user_id	Integer	非空, 索引	用户 ID
title	String(200)	非空	试卷标题
subject	String(50)	可空	科目

属性名	数据类型	约束条件	注释
grade_level	String(20)	可空	学段
total_questions	Integer	非空, 默认 20	总 题 数
difficulty_distribution	JSON	可空	难 度 分布
question_type_distribution	JSON	可空	题 型 分布
knowledge_points	JSON	可空	知 识 点列表
questions	JSON	非空	题 目 列表
answer_key	JSON	可空	标 准 答案
paper_type	String(20)	非 空 , 默 认 custom	试 卷 类型
time_limit	Integer	可空	考 试 时长
total_score	Integer	非 空 , 默 认 100	总分
is_template	Boolean	默认 False	是 否 模板
is_published	Boolean	默认 False	是 否 发布
created_at	DateTime	默 认 当 前 时 间	创 建 时间
updated_at	DateTime	默 认 当 前 时 间 , 更 新 自 动 刷 新	更 新 时间

表 3.2.11 API 请求日志表

属性名	数据类型	约束条件	注释
id	Integer	主键, 索引	日志 ID
provider	String(64)	可空	模型提供商
source	String(32)	非 空 , 默 认 user	调用来源 (user/admin/test 等)
success	Boolean	默认 True	是否成功
created_at	DateTime	默 认 当 前 时 间 , 索 引	创建时间

关键技术:

- 1.使用了 md5 算法对密码进行了加密处理使数据库不会显示真实的密码
- 2.使用了 springboot+mybatis-plus+jQuery+bookstrap4+jQuery ui+thymeleaf 等技术。

第四章 测试报告

4.1 测试策略

多层测试： 后端采用 Pytest 分层（集成 / 端到端），前端以 Jest + React Testing Library + Cypress 组合，配合静态检查（ESLint、TypeScript、Ruff）

数据隔离： 所有 Pytest 用例使用内存 SQLite (sqlite:///memory:) 与临时文件夹，在 fixture 中统一建表/清理，确保测试互不污染

4.2 测试概要

表 4.2.1 测试概要表

测试层级	目标/范围	主要用例	依赖与工具	结果记录
单元测试 (pytest -q)	service/AI mock、utils、repositories	成功路径、非法输入、重试/回退	pytest + sqlite:///memory: + mock provider	backend/tests/*.py
集成测试	router → service → repo → DB	上传解析、AI JSON 校验、JWT 鉴权	FastAPI TestClient、内存 SQLite、假 JWT	backend/tests/test_*.route.r.py
前端组件测试	Recharts、force-graph、Zustand store	节点渲染、图表数据绑定、状态切换	Vitest + React Testing Library + MSW	frontend/src/__tests__/*.test.tsx
端到端 API 测试	核心业务流程	Learning Map、Exam、Teacher Persona、Read-Along、Mistake、Path	pytest + httpx + mock storage	tests/e2e/test_workflows.py
手动/探索测试	UI 主题、交互细节、上传进度	蓝白主题一致性、错误提示、SSE	浏览器 + Mock 后端	QA checklist/wiki

4.3 功能测试

认证与权限： 验证注册、登录、刷新 Token、不同角色访问控制，并模拟 JWT 过期/伪造。

AI 多模型流程： 测试不同 provider 配置、fallback 策略、系统 Prompt 注入、模型签名剥离、JSON 解析失败后的重试。

文件上传与解析：覆盖 PDF/DOCX/PPTX/TXT/MD、编码差异、超大小、空文档、重复上传等场景，确认 learning_files 记录与截断标记。

知识图谱：确保生成 ≥ 6 节点、8 条边，校验节点/边入库、历史检索权限、前端 graph API JSON。

AI 测评与学习计划：完整验证题目生成、答题提交、错题讲解、学习计划 CRUD、PDF 报告导出、弱项分析逻辑。

管理后台：Prompt 版本管理、模型配置加密写入/读取、系统诊断、调用日志过滤。

4.4 测试执行情况

4.4.1 系统登录测试

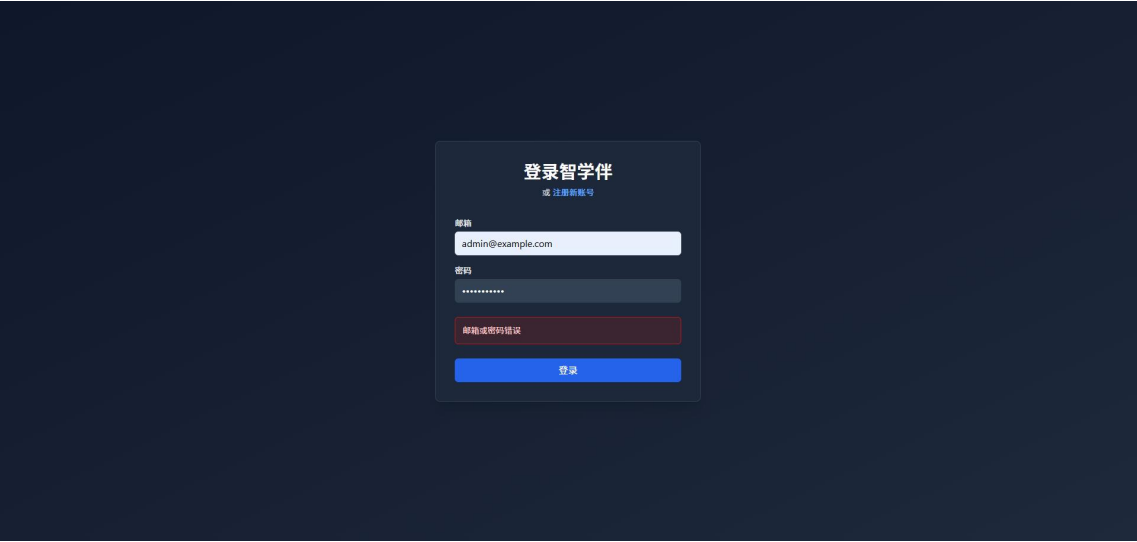


图 4.4.1 登录测试图

4.4.2 AI 对话测试



图 4.4.2 AI 对话测试图

4.4.3 学习计划生成测试

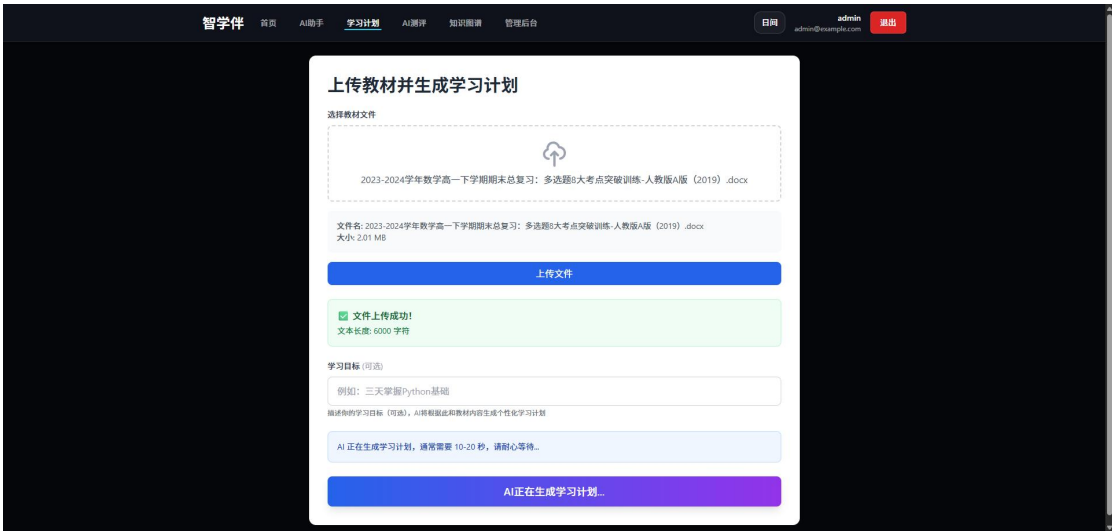


图 4.4.3 学习计划生成测试图

4.4.4 AI 测评智能出卷测试



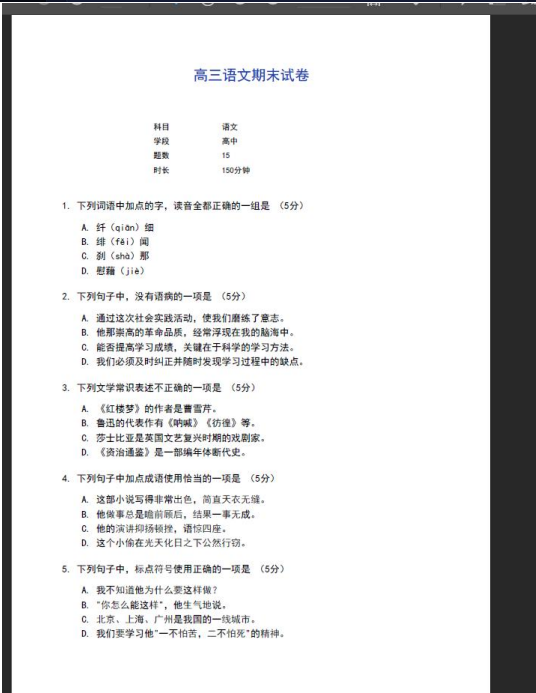


图 4.4.4 AI 测评智能出卷导出测试图

4.4.5 知识图谱测试

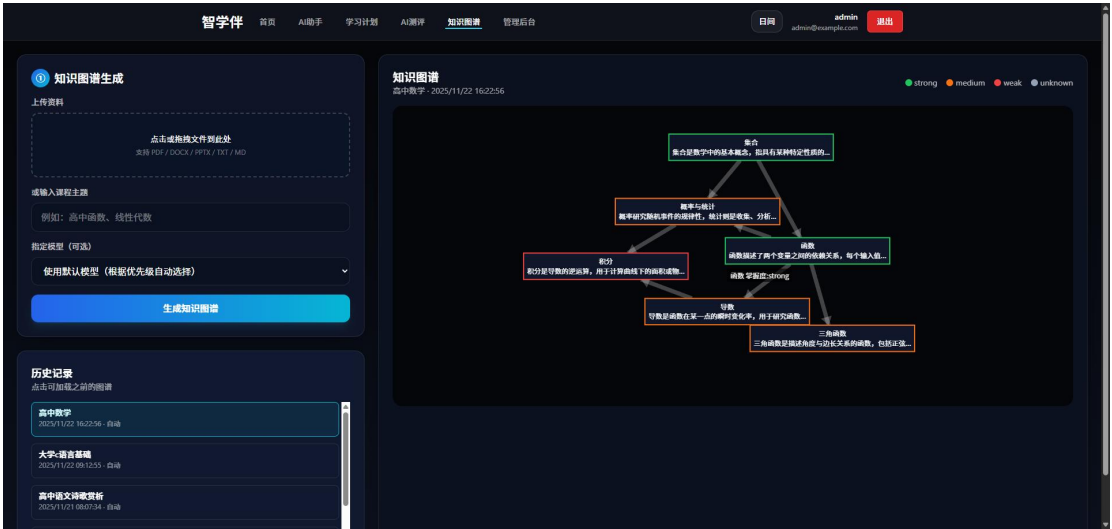


图 4.4.5 知识图谱测试图

4.4.6 管理后台测试



图 4.4.6 管理后台测试图

4.4.7 前后交互测试

表 4.4.7 交互测试表

接口/页面	前端动作	预期后端响应	验证点	状态
/api/v1/learning-map/upload-file	上传 markdown/docx , 展示 force-graph	JSON nodes ≥ 8 、 edges ≥ 10 , 含 mastery	节点颜色映射、点击详情弹窗	通过
/api/v1/exam/upload	上传 PDF, 显示题目	questions[]、exam_analysis	题目可编辑, 图表与	通过

	列表+饼图		JSON 一致	
/api/v1/teacher/personas + AI Chat	新建 persona、在聊天侧栏切换	Persona CRUD JSON, AI 回复风格变更	Markdown 预览、语气对比	通过
/api/v1/read-along/upload & /sections/{id}	上传文章, 点击段落请求讲解	sections[] + explanation	段落高亮、loading/错误提示	通过
/api/v1/mistake/analyze	批量提交错题	cause 分类 + remedial tasks + map nodes	跳转到知识图谱节点	通过
/api/v1/path/generate	输入目标/时间, 展示时间轴	path_json + timeline_json	列表顺序 + Recharts 甘特	通过
通用 JWT 流程	sessionStorage 注入 token 调用 apiClient	401/403 处理、成功返回数据	CORS、错误提示组件	通过
SSE/轮询 (Read-Along 可选)	监听讲解流/轮询	streaming 文本或 JSON	断线重连、loading 状态同步	待实现

4.5 测试总结

质量结论：关键业务链路已通过系统性验证，AI 调用、权限控制、文件解析等高风险点具有冗余保护。

已知风险：高并发 / 长文本场景需更多压测；知识图谱与测评对第三方 AI 稳定性敏感；移动端 UI 覆盖仍有限。

后续计划：补齐剩余 Service 的单测（整体覆盖率目标 90%）、引入 Playwright 多浏览器回归、使用 K6/Locust 建立性能基线、沉淀测试数据/脚本以支持展示。

第五章 安装及使用

5.1 环境要求

系统要求: Windows 10+/macOS/Ubuntu 24+

后端: Python3.10+、pip、虚拟环境工具 (venv/conda)、SQLite 或 MySQL 8+/PostgreSQL 14+

前端: Node.js 18+、npm/pnpm、浏览器

网络要求: 可正常访问 DeepSeek、文心一言、星火大模型、ChatGLM、Moonshot 等模型

5.2 安装教程

1. 下载源代码: `git clone https://github.com/YEYUbaka/zhixueban.git`

```
cd zhixueban
```

2. 后端初始化:

```
cd backend
```

```
python -m venv venv
```

```
.\venv\Scripts\activate
```

```
pip install -r requirements.txt
```

```
cp .env.example .env # 按需修改数据库、AI 配置、密钥
```

```
uvicorn main:app --reload --port 8000
```

3. 前端初始化

```
cd frontend
```

```
npm install
```

```
npm run dev -- --host 0.0.0.0 --port 5173
```

4. 数据库准备

开发模式默认 SQLite，首次启动自动建表；

切换 MySQL/PostgreSQL 时更新 .env 的 DATABASE_URL，执行 Alembic 迁移

或 `python main.py --init-db`；

可选运行 `backend/generate_sample_files.py` 生成功能演示数据。

5.3 使用说明

1. 访问入口

前端：<http://localhost:5173>

后端 API 文档：<http://localhost:8000/docs> (Swagger) 或 `/redoc`

2. 账号与权限

第一个注册用户自动成为管理员；

管理后台路径 `/admin/dashboard`，可配置模型、Prompt、系统参数。

3. 典型流程

上传教材 → 解析生成学习计划 → 查看知识图谱 → AI 自动测评 → 导出学习报告；

AI Chat 页面可切换教师人格、选择模型、查看 Markdown/LaTeX 渲染结果。

4. 常用命令

后端测试: `cd backend && pytest -q`

前端测试: `cd frontend && npm test`

代码质量: `ruff check --fix && npm run lint`

5. 部署建议

推荐 Docker Compose 或前后端独立容器, 配 Nginx 反向代理 + HTTPS;

生产环境务必替换 .env 的 SECRET_KEY/ENCRYPTION_KEY, 并将 AI API Key 写入数据库 加密存储;

配置日志轮转、定时备份 uploads/ 与 reports/ 目录。

第六章 项目总结

6.1 任务分配

为了确保项目的专业化、高质量推进, 我们团队五名成员按各自技能特点进行了明确分工, 各司其职、协同开发:

1. 涂维轩 —— 负责整体架构设计、后端核心开发 (组长)

负责项目整体方案设计、系统架构制定、技术选型

主导 FastAPI 后端框架、数据库模型、AI 多模型调用体系、知识图谱/路径规划算法实现

负责后端统一 AI 调度核心: AIService、ModelRegistry

负责服务器部署、性能优化和线上环境运维

2. 刘富祥 —— 前端工程、UI/UX 体验设计负责人

负责 Web 前端架构 (Vite + React + Tailwind) 与组件化设计

开发 AI 聊天界面、学习仪表盘、可视化图表、文件上传页面

主导知识图谱可视化与交互优化 (react-force-graph)

深度参与用户体验设计, 保证界面美观、易用、统一风格

配合后端联调 API, 完成整体交互逻辑

3. 李娜—— 文档撰写负责人 & 管理后台界面协作开发

全权负责大赛所需文档: 立项分析、需求文档、开发文档、测试文档撰写

协助前端开发后台管理模块 (模型管理、Prompt 管理)

参与测试工作, 补充测试用例、用户反馈与报告整理

4. 杨兴露—— 测试与内容体系负责人 (学习计划/测评体系)

负责学习计划、测评系统的业务逻辑测试和功能体验优化

负责资料收集、样例文档整理 (学习材料、试题等), 确保 AI 解析准确性

负责展示材料、PPT 内容与整体视觉风格制定

参与撰写系统展示与答辩材料, 整理项目亮点

对学习路径规划、弱项诊断等模块提出可行性改进意见

5. 何萧—— AI Prompt 体系设计、答辩展示负责人

负责 20+ 类 AI Prompt (教师人格、真题分析、测评生成、知识提取等) 的设

计优化

负责答辩内容的推进与演练，制定演示逻辑与流程

协助文档整理，参与界面内容文案、AI 介绍文本与项目亮点设计

6.1 面对困难

在从零开发这套“智学伴”系统的过程中，我们反反复复遇到模型接口超时、内网映射受限、多人协作规范不统一等棘手问题。每一次后台 500、前端白屏、配置冲突，都逼着我们重新梳理分层架构、加固超时策略、统一 Prompt/模型治理；那些看似毫无头绪的日志、被 Vite 拦下的请求，都成为我们打磨工程化细节的催化剂。也正是在一次次“卡住—排查—修复”的循环里，我们练就了快速定位问题的耐心和韧性，把困难转化为架构更稳定、体验更友好的推动力。这段旅程告诉我们：困难不可避免，但只要保持冷静、专注拆解，它就会成为项目成长的助燃剂。

6.2 开发感悟

项目开发历时数周，我们团队从最初的构想，到架构搭建，到不断打磨功能细节，深刻体会到一次完整软件项目的复杂性与价值。在这个过程中，我们有以下体会：

1. 团队合作比技术更重要，每个人都有不同的特长：有人擅长后端架构，有人擅长设计与前端，有人写文档、有人做测试，有人优化 Prompt。大家分工明确，又彼此补位，形成了高效的协作链，让项目能在短时间内稳定成型。

2. AI 不是万能的，工程化才是关键 AI 能解决很多问题，但要真正落地成一个教育产品，需要：数据结构规划 AI 调用策略与 fallback Prompt 工程，前后端联动，用户体验设计，我们体会到工程化的重要性远比预想更高。

3. 需求会不断变化，系统要能持续迭代，智学伴从最初的“AI 问答工具”，不断发展为：知识图谱，学习路径规划，弱项诊断，真题解析，AI 教师人格，Dashboard 学习报告，每一项新增功能都是需求驱动，体系也越来越成熟。

4. 文档与规范是不可或缺的一环

通过统一文档规则、统一项目结构，我们也理解到专业项目与学生作品之间最大的差别不是功能多少，而是规范程度。

5. 最终收获的不只是作品，而是能力，项目让我们真正掌握了：Web 系统工程化开发流程，AI 能力集成与提示词工程，团队协作与项目管理，教育场景需求分析与产品化思维，这份作品不仅是比赛成果，也是我们进入更高技术阶梯的重要实践。

第七章 参考文献

FastAPI 文档: <https://fastapi.tiangolo.com/>

SQLAlchemy 文档: <https://docs.sqlalchemy.org/>

React / Vite / Tailwind 官方指南:

<https://react.dev/>

<https://vitejs.dev/>

<https://tailwindcss.com/>

Recharts:

<https://recharts.org/>

React-force-graph:

<https://github.com/vasturiano/react-force-graph>

ReportLab 用户手册:

<https://www.reportlab.com/docs/reportlab-userguide.pdf>

PyMuPDF 文档:

<https://pymupdf.readthedocs.io/>