浙江水利水电学院

课程设计报告

（ 2025-2026 学年 1 学期）

|  |  |
| --- | --- |
| 设计题目 | AI智能·学习搭子 |

|  |  |
| --- | --- |
| 姓 名： | 王泽桉 |
| 学 号： | 2022b11043 |
| 专 业： | 软件工程 |
| 班 级： | 软件工程22-2 |
| 教学单位： | 计算机科学与技术学院 |
| 指导教师： | 杨欐 |

2025年9月10日

**总分:** 分 **五级计分制:**

**评阅教师签名：**

**评语：**

**目 录**

[1.设计介绍 2](#_Toc210591890)

**[1.1 目的和意义](#_Toc210591891)** [2](#_Toc210591891)

**[1.2 设计环境与要求](#_Toc210591892)** [2](#_Toc210591892)

[2.设计内容 3](#_Toc210591893)

**[2.1 需求分析](#_Toc210591894)** [3](#_Toc210591894)

[2.1.1 用户画像 3](#_Toc210591895)

[2.1.2 竞品分析 4](#_Toc210591896)

[2.1.3 功能还原 4](#_Toc210591897)

[2.1.4 系统构建 6](#_Toc210591898)

[2.1.5 技术栈确认 6](#_Toc210591899)

**[2.2 详细设计](#_Toc210591900)** [7](#_Toc210591900)

**[2.3 功能实现](#_Toc210591901)** [7](#_Toc210591901)

**[2.3.1](#_Toc210591902)** [7](#_Toc210591902)

[3.设计总结 7](#_Toc210591903)

**[3.1 设计过程碰到的难点与解决方法](#_Toc210591904)** [7](#_Toc210591904)

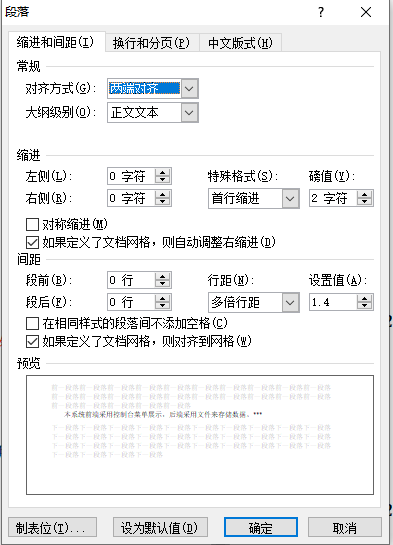
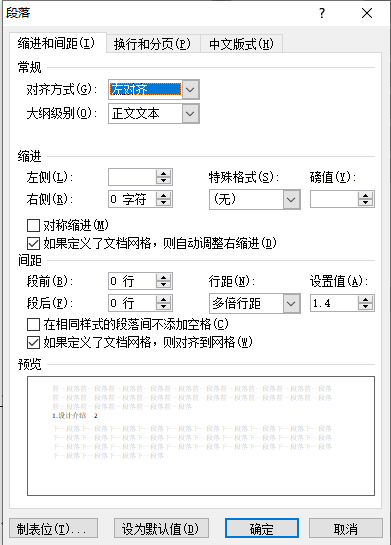
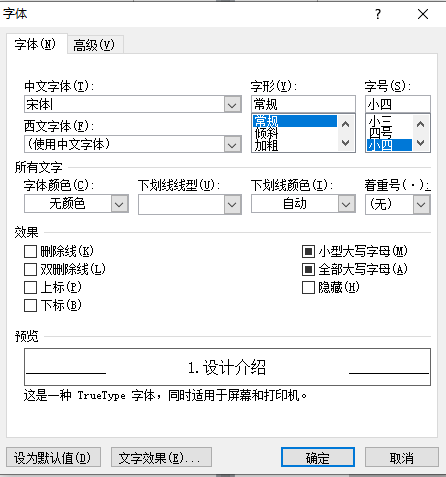
**[3.2 成果与收获](#_Toc210591905)** [8](#_Toc210591905)

**[完成后，删除此段红色文字**

**报告撰写结束后，右键更新域——更新整个目录，同时修改目录格式**

**目录格式：字体：小四，宋体，不加粗，不倾斜；1.4倍行距，段前段后为0**

**正文格式（除设定的标题外）：小四，宋体，段落1.4倍行距，段前段后为0，首行缩进两字符**

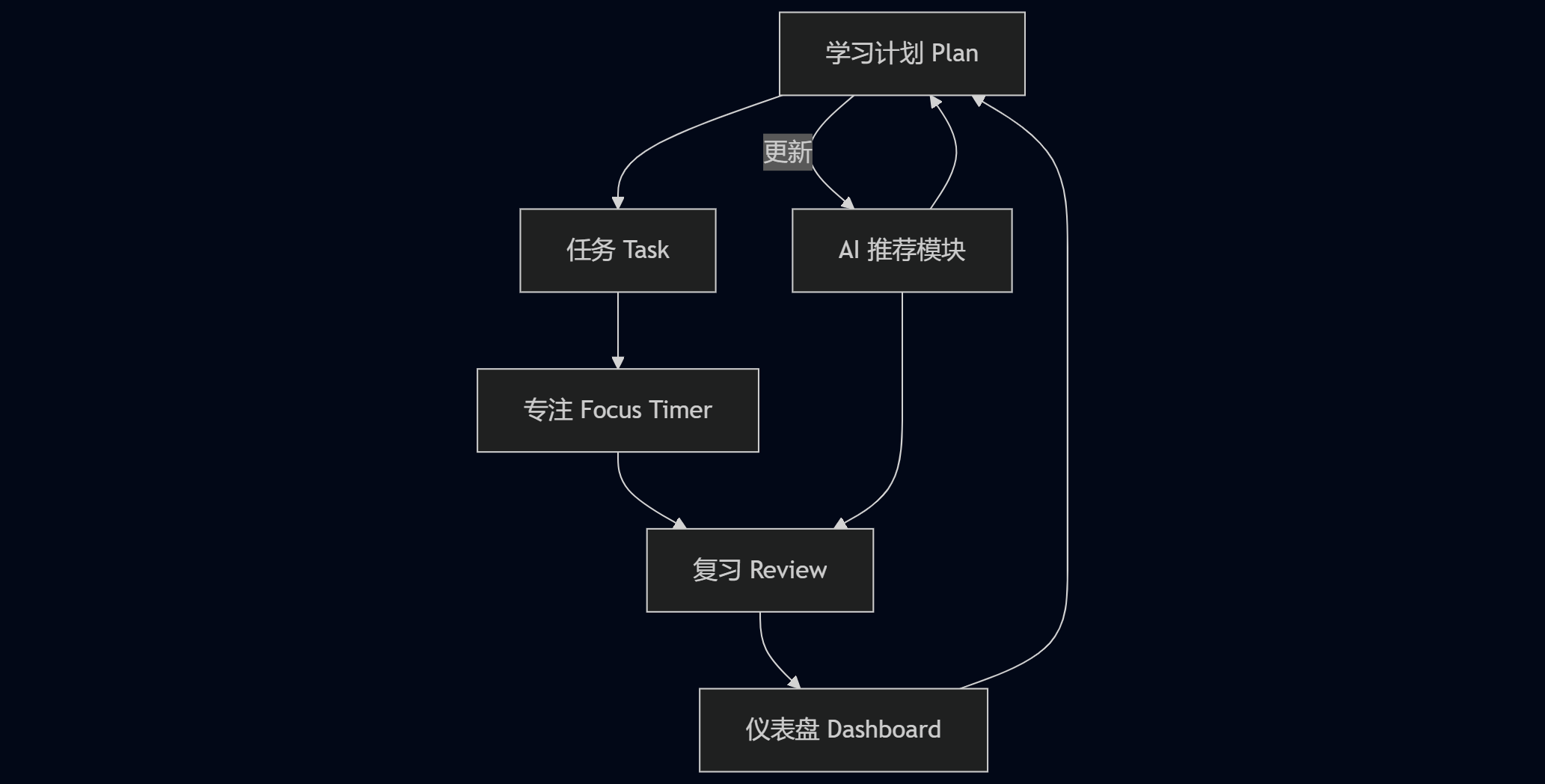
**]**

# **1.设计介绍**

## **1.1 目的和意义**

1.**应对信息过载与知识更新加速**  
在信息爆炸时代，知识更新极快，学习者很难从海量信息中高效筛选并吸收。系统的建设就是为了帮助学习者应对这一挑战，提供更精准的学习资源和路径

2.**突破传统学习模式的局限**  
传统“千人一面”的教学方式，往往无法满足个体的学习节奏和兴趣偏好。而系统的目标是实现差异化、个性化的学习支持，提升学习效率和体验



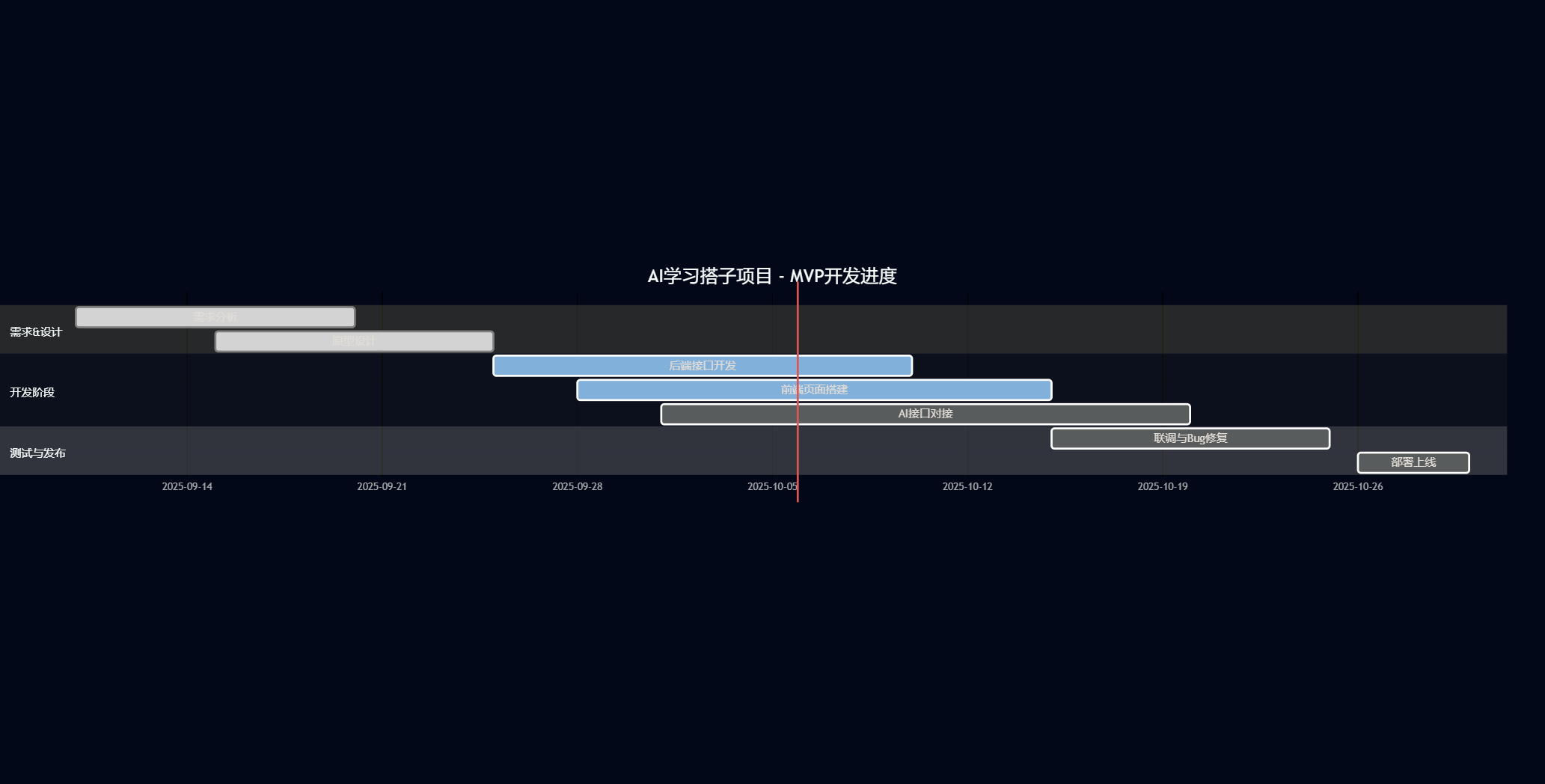
学习计划与专注模块关系

3.**缓解学习过程中的孤独与困境**  
学习者常常面临孤独感、缺乏即时反馈、缺少个性化指导等问题。系统通过 AI 智能搭子功能，能提供陪伴、互动和定制化反馈，减少学习的枯燥感和挫败感

4.**推动教育与人工智能的深度融合**  
通过该系统，可以更好地探索 AI 在教育场景中的应用价值，不仅为学习者赋能，也为教育模式创新提供实践意义

## **1.2 设计环境与要求**

**一、设计环境**



开发迭代甘特图

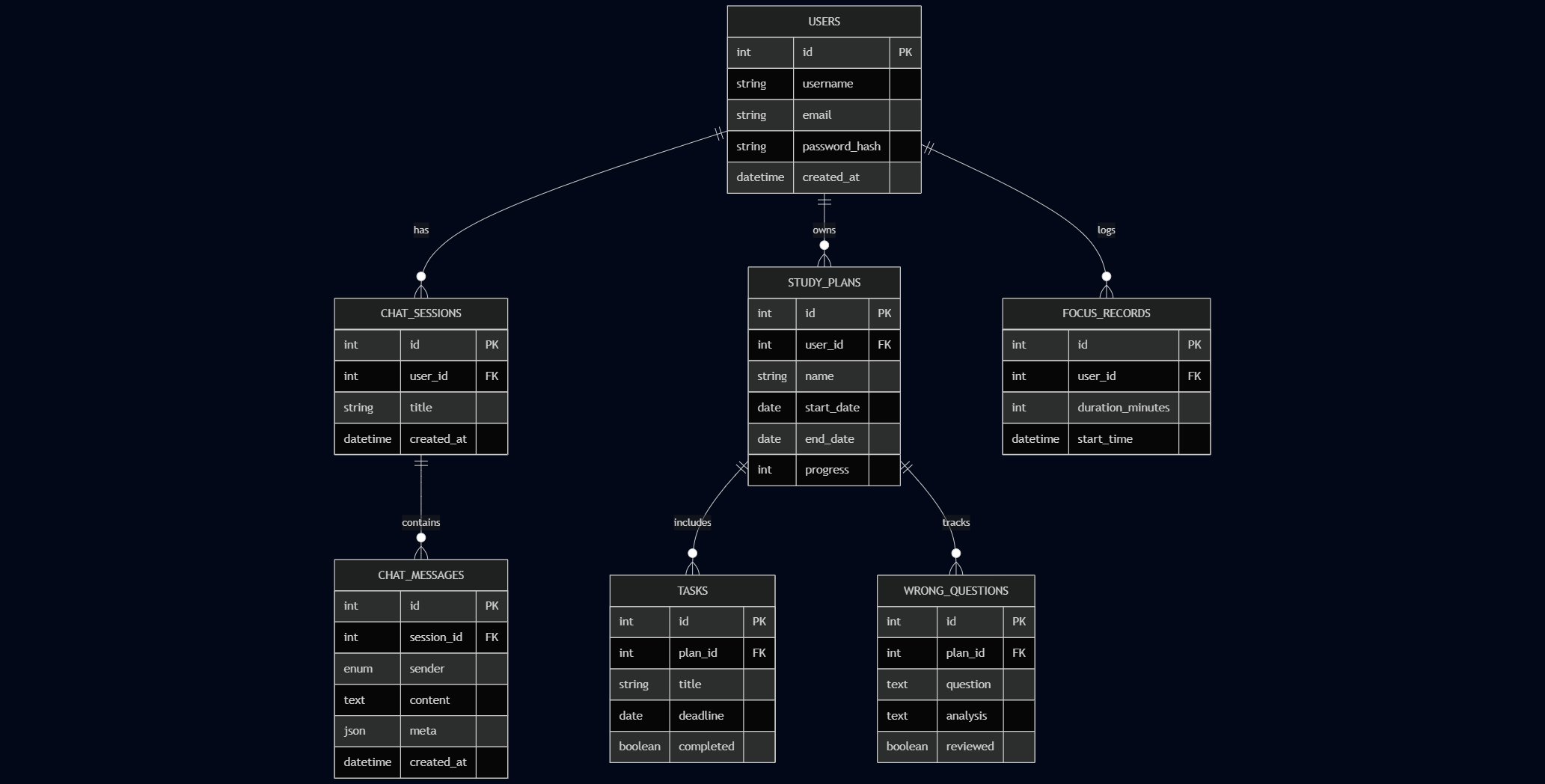
1. **开发框架与语言**

**后端**：Spring Boot（基于 Java），用于实现业务逻辑、用户管理、接口服务

**前端**：Vue.js（渐进式框架），用于构建交互式页面和学习场景的可视化界面

**数据库**：MySQL，存储用户信息、学习记录、任务配置等结构化数据

**知识向量检索**：阿里巴巴云端知识库，提供向量化语义检索功能，支撑个性化推荐与智能答疑



数据库逻辑结构

1. **运行环境**

**服务器环境**：Windows11

**依赖环境**：JDK 17+、Node.js 16+、Nginx（前后端分离部署）

**云平台**：阿里云，利用其 **知识库向量检索服务** 与 **对象存储（OSS）**

1. **AI 支持**

调用Qwen3大语言模型 API，结合向量数据库，实现个性化学习支持

支持知识库的增量更新与语义匹配，保证学习资源实时性

**二、设计要求**

1. **功能性要求**

**个性化推荐**：基于学习者的行为数据和知识库向量检索，提供定制化学习路径与资源

**互动陪伴**：解决学习孤独感，提供 AI 智能搭子功能，实现即时问答与激励反馈

**学习记录管理**：支持学习进度追踪、错题本、知识点复盘

**多端适配**：支持 Web 端为主，未来可扩展至移动端

1. **性能要求**

系统并发能力强，能够同时支持多人学习交互

向量检索延迟需控制在 **200ms** 以内，保证用户体验

系统可扩展，支持知识库规模化扩充

1. **安全性要求**

用户信息加密存储（密码哈希化、通信采用 HTTPS）

数据访问权限控制，确保知识库与用户学习数据的安全

1. **可维护性要求**

前后端分离架构，方便模块化维护与迭代开发

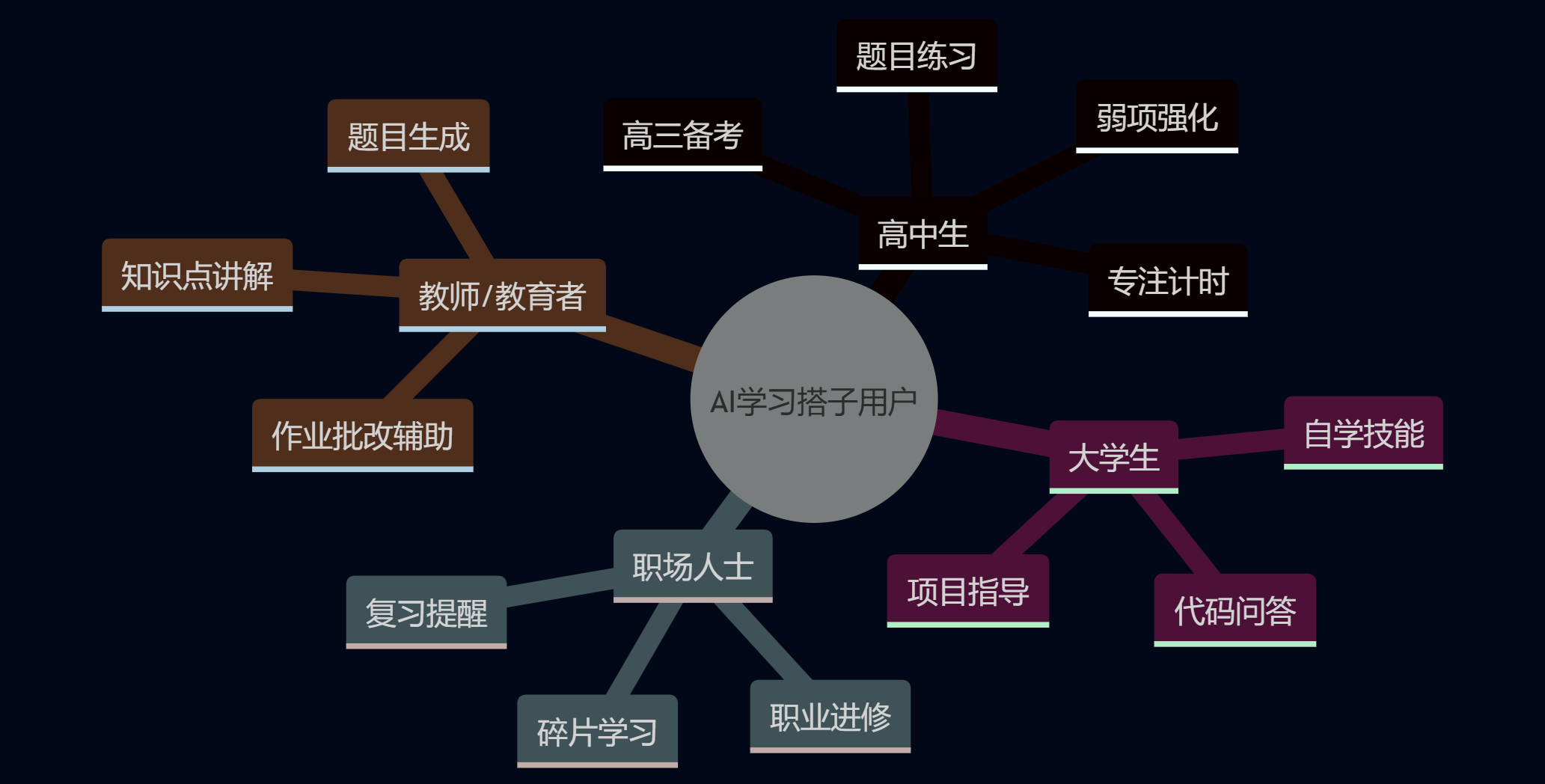
日志监控与错误报警机制，便于运维

代码风格统一，使用 Github 进行版本管理

# **2.设计内容**

## **2.1 需求分析**

### 2.1.1 用户画像



用户画像分类图

1. **李华 — 高三备考生（主要用户）**

年龄：17–18，学习目标：高考冲刺

痛点：题量大、时间紧、需要针对弱项练习、需要随时问答与错题分析

典型场景：在晚自习/回家路上，用手机问题、做定时“专注”练习（Pomodoro 风格），查看错题回顾与学习计划

1. **小王 — 大学自学者（偏技能/编程）**

年龄：20–24，目标：自学编程、完成小项目

痛点：缺少即时代码/概念讲解，容易走弯路

场景：在图书馆/寝室用“聊天”向 AI 咨询概念或代码思路，保存学习笔记

1. **张姐 — 职场进修者**

年龄：28–35，目标：职业证书或技能提升

痛点：时间碎片化、需要高效复习与练习题库

场景：利用“工作”视图安排学习任务与复习提醒，AI 帮忙把长文本浓缩成考点

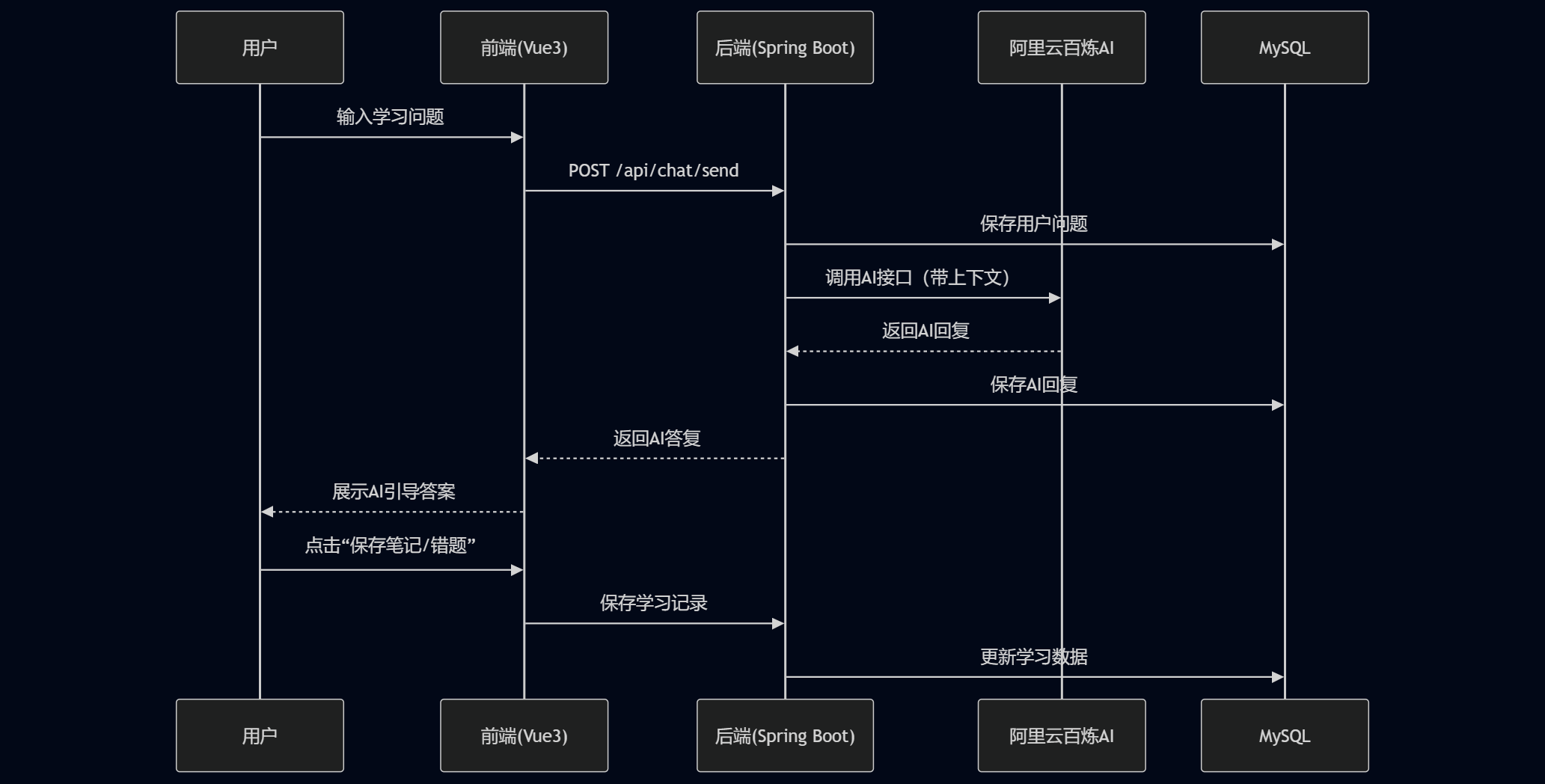
1. **林老师 — 终身学习/教师（次要用户）**

年龄：35+，目标：备课、布置作业与判题效率

场景：用 AI 生成练习题、参考答案、讲义摘要（教师端功能可作为中长期拓展）

这些画像将直接影响功能优先级（例如：高三生优先要题目练习、错题本、专注模式；大学生更需要开放问答与笔记导出）

### 2.1.2 竞品分析



核心业务流程图

**作业帮（国内）**

亮点：已经把 AI 用在学习机/作业批改/视频答疑等场景，2024年报告过大量自动批改与高频视频答疑数据，侧重“中小学生作业辅导与题库”

**猿辅导 / 小猿（国内）**

亮点：基于自有数据构建知识图谱与大模型（大量教学视频/题库支持），推出小猿学练机与 AI 驱动功能，强调练习路径个性化与多模态交互

**Khanmigo（Khan Academy，国际）**

亮点：作为非营利教育组织的 AI 辅助教学（Khanmigo），采取引导式教学（不直接给出答案而是引导学生思考），重视学习伦理与教育效果。适合以“引导+练习”为核心的产品策略参考

**Quizlet（国际）**

亮点：强大的记忆练习（闪卡）生态，正在把 AI 用于内容生成与个性化课程（Course-Powered）以增强协作学习

**ChatGPT / 大型通用模型（背景）**

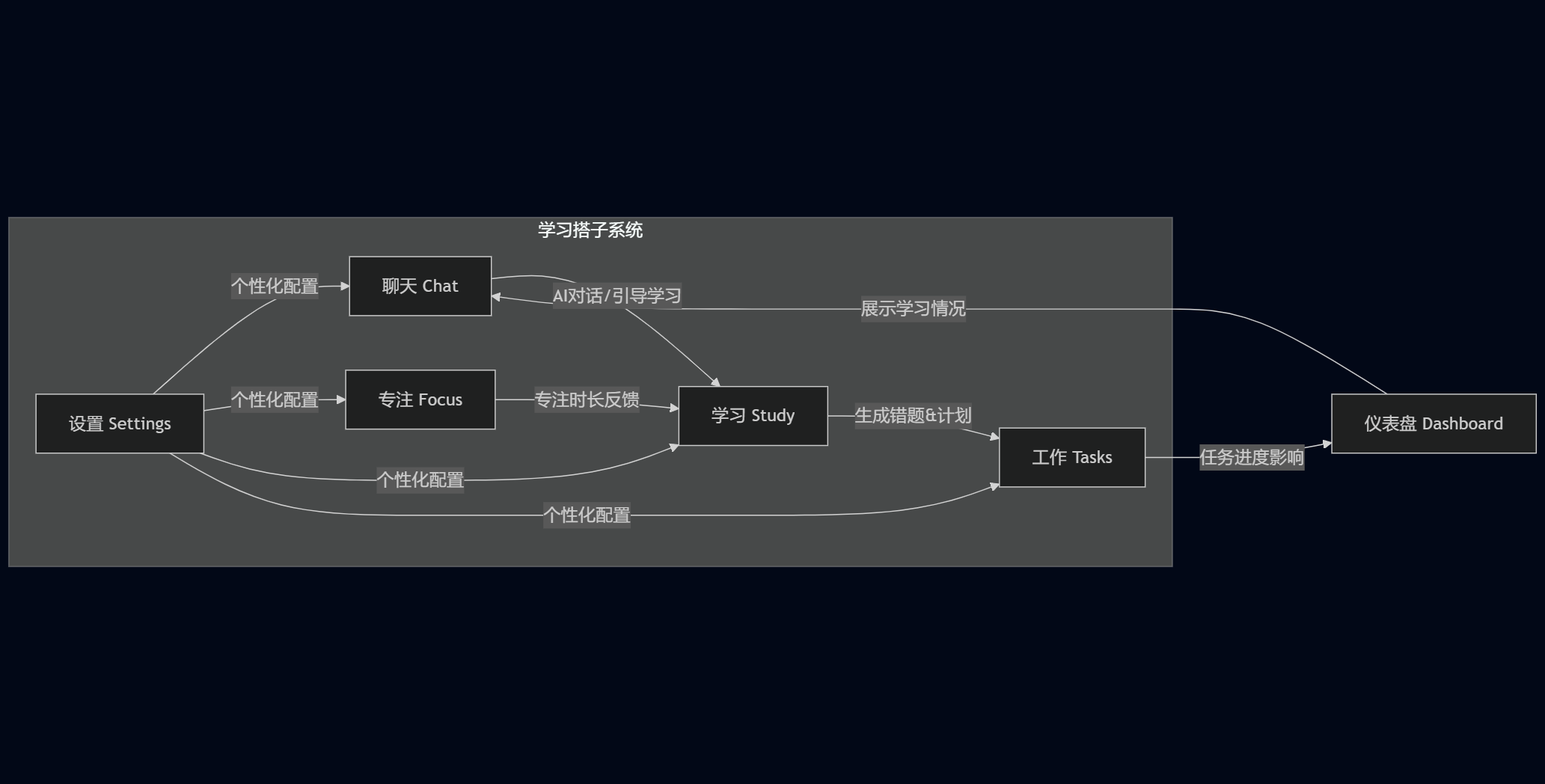
趋势：多家通用 AI 服务正加入“学习/教学模式”（如 ChatGPT 的 Study Mode），体现行业方向：把大模型从“答案引擎”转为“教学引导器”。这说明产品设计上需要把“引导式学习（Socratic）”与“直接答疑”做权衡

**竞品给我们的启示（机会）**

差异化策略：聚焦“陪伴式学习 + 轻量化社交（学习搭子） + 目标导向的学习计划”，而非单纯题库或单一答疑工具

技术上：竞品倾向把 AI 用在题目生成、个性化练习安排与多模态（语音/图像）交互。我们可以从“对话 + 学习计划 + 专注模式 + 进度反馈”切入，形成差异化（不需要一次性具备超大题库，先以智能陪伴与适配练习为 MVP）

### 2.1.3 功能还原



功能模块关系图

**A. 聊天（Chat） — 核心 MVP 功能之一**

**目标**：实现多轮上下文感知的学习对话，能回答问题、引导学习并生成练习题或讲解要点

**用户故事**

作为学习者，我希望能在聊天中输入问题并得到分步引导的解题思路（而不是直接给出答案），以加深理解

作为学习者，我希望能保存有价值的对话为笔记/题目，方便复习

**功能细分**

多轮对话上下文追踪（会话 ID 管理）

分步引导模式（可切换“直答/引导”）

会话历史查看、标星与保存为“知识卡”

后端调用阿里云百炼/AI 接口生成回复并做缓存（配置化 prompt 与温度）

**验收标准**

用户发送问题 → 在 3s 内返回 AI 初次响应（具体延时依赖外部 AI 接口）

会话历史可以分页/检索，用户可标记并导出为文本/JSON

**示例 API（REST）**

POST /api/chat/send

body: { userId, sessionId?, message, mode: "guide"|"direct" }

返回: { replyId, text, aiMeta:{model, promptUsed, tokens?} }

GET /api/chat/history?userId=&sessionId=&page=

POST /api/chat/save-note（把会话片段保存为笔记）

**数据库（MySQL）表设计示例（简要）**

users（id, username, hashed\_password, email, created\_at）

chat\_sessions（id, user\_id, title, created\_at, last\_active）

chat\_messages（id, session\_id, sender:ENUM('user','ai'), content TEXT, meta JSON, created\_at）

**B. 专注（Focus） — 学习习惯培养**

**目标**：提供定时/专注机制（例如 Pomodoro），并在专注结束后自动记录学习时长与打卡

**功能**

开始/暂停/结束专注任务（持续时长、任务标签）

专注统计（日/周/月）并与学习计划关联

专注时禁止/屏蔽通知（前端控制）

**验收**

用户能创建一个专注任务并在完成后查看累计时长与任务清单

**C. 学习（Study） — 内容与练习管理**

**目标**：管理学习计划、题库、错题本与进度追踪

**功能**

创建/编辑学习计划（目标、起止日期、每天计划时长）

系统根据计划每天推荐练习（基于错题、知识点）——此处 AI 负责“推荐题目/习题难度调整”逻辑（调用阿里云 AI 做能力评估与题目生成或筛选）

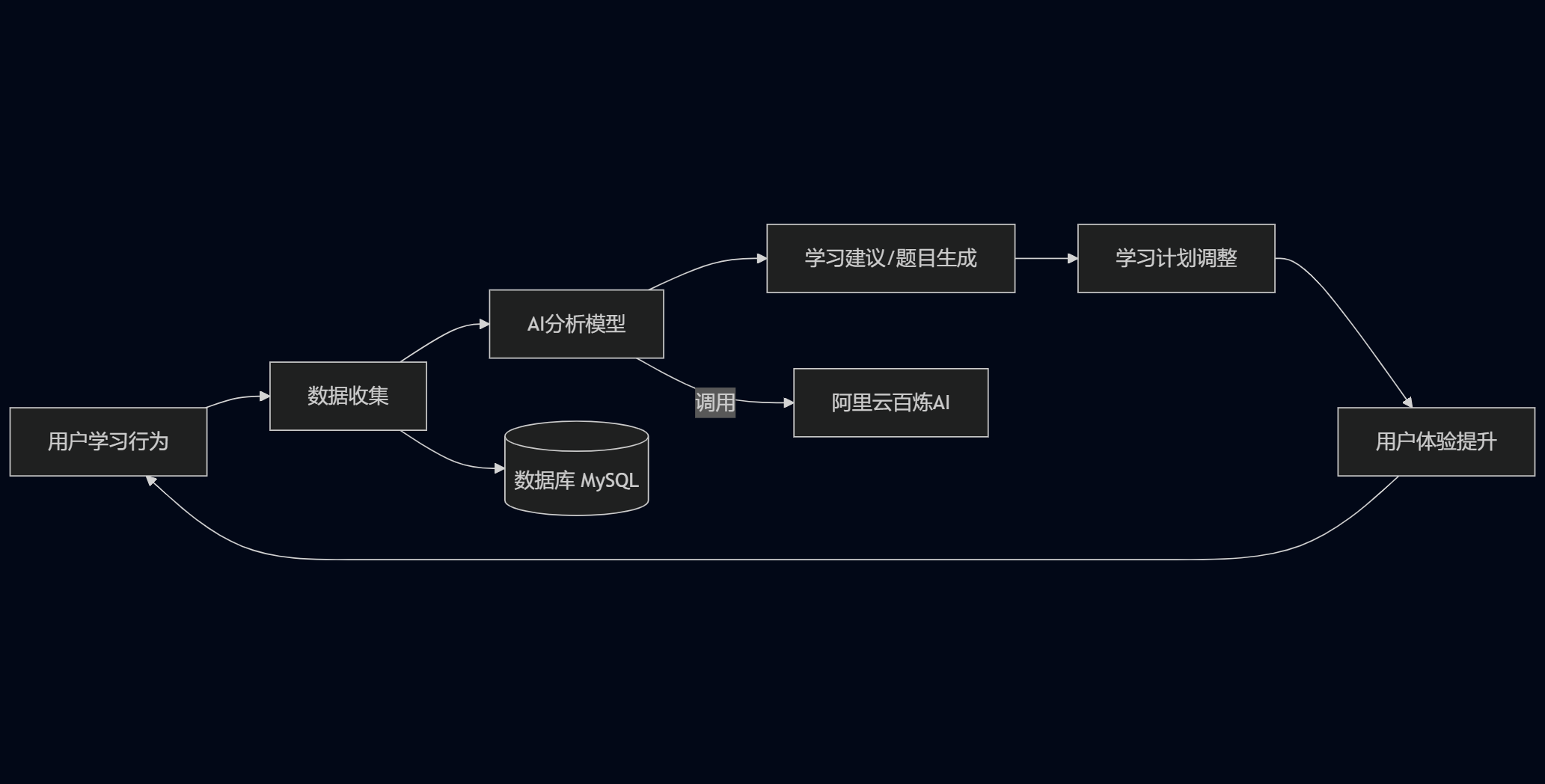
AI学习搭子项目

错题本（记录错题、AI 给出逐题解析、复习提醒）

学习数据仪表盘（完成率、正确率、弱点分布）

**验收**

学习计划可被创建并生成当天推荐练习列表；错题复习能够标记“已掌握”



学习行为与数据闭环图

**D. 工作（Tasks / Productivity）**

**目标**：把“工作”视为学习内的任务管理（任务、提醒、待办）

**功能**

任务创建（deadline、优先级、关联学习计划）

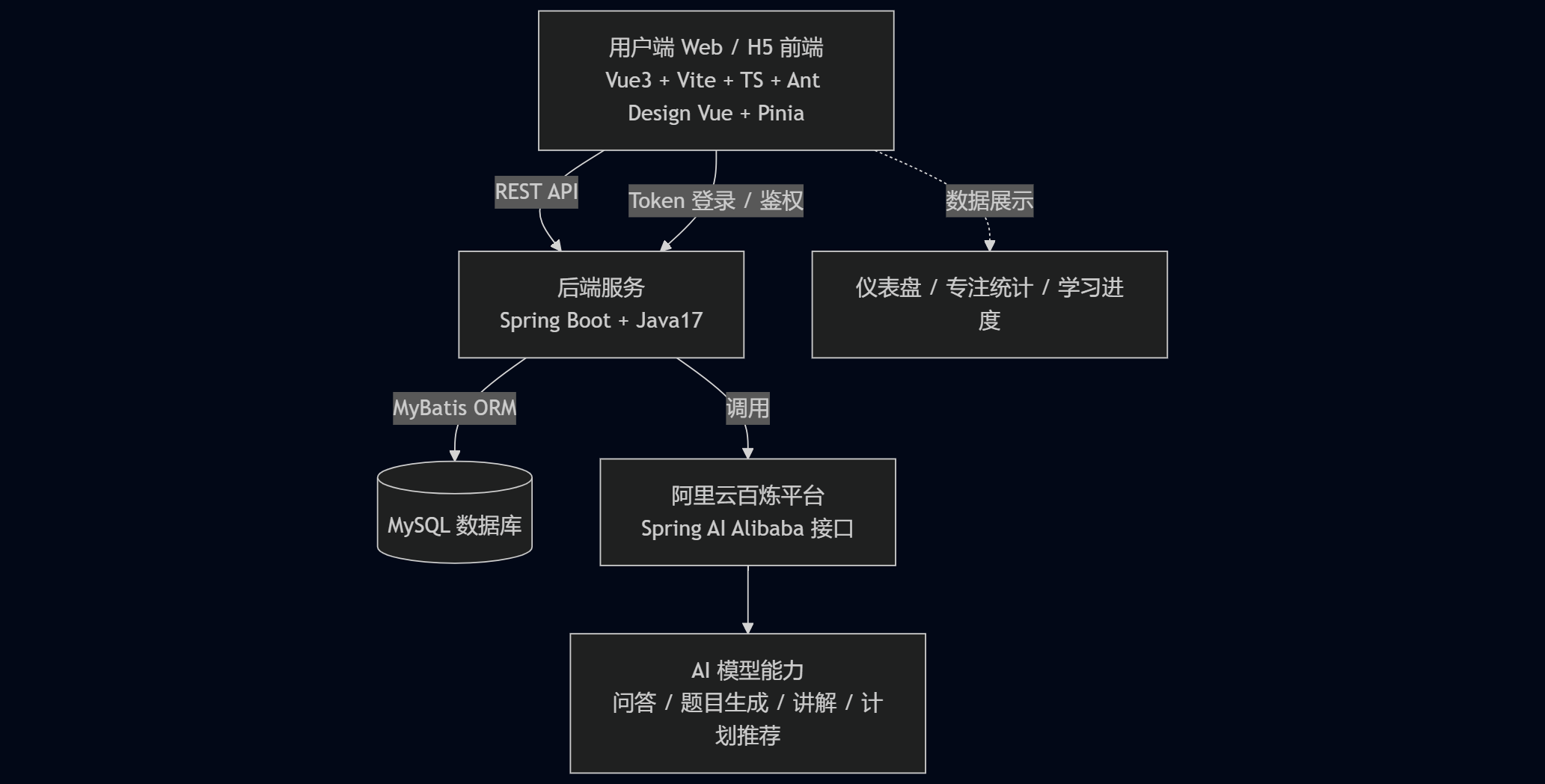
日历/日程视图（可在未来迭代）

任务完成触发自动奖励/日统计

**E. 设置（Settings）**

账户管理、隐私与数据导出、AI 模式切换（直接答/引导）、通知偏好、主题/外观等

### 2.1.4 系统构建



项目总体结构（系统架构图）

**总体架构**

**客户端（SPA, Vue 3 / TypeScript）**：负责 UI、状态管理（Pinia）、路由（Vue Router），与后端通过 RESTful API 通信（项目现状已如此组织）

**后端（Spring Boot）**：提供 REST 接口、业务逻辑、持久化（MyBatis 对接 MySQL），并作为 AI 中间层负责与阿里云百炼平台（通过 Spring AI Alibaba 或自定义 SDK）交互以获取模型响应与题目生成

**数据库（MySQL）**：存储用户、会话、学习计划、题目记录、错题与统计数据（见上面数据表示例）

**AI 服务**：通过后端调用阿里云的 AI 接口（API Key 等配置保存在后端配置文件/环境变量中，走 HTTPS），后端负责 prompt 管理、上下文拼接与对 AI 响应的后处理（例如抽取题目、生成解析）

**关键交互**

1. 用户前端发送问题 POST /api/chat/send → 后端接收并写入 chat\_messages（user）
2. 后端构建 prompt（包含必要上下文）并调用阿里云 AI 接口（同步或异步）
3. AI 返回响应 → 后端将 AI 内容写入 chat\_messages（ai），并返回给前端展示
4. 若用户“保存为错题”，后端在 problem\_records 表插入条目并更新学习计划统计

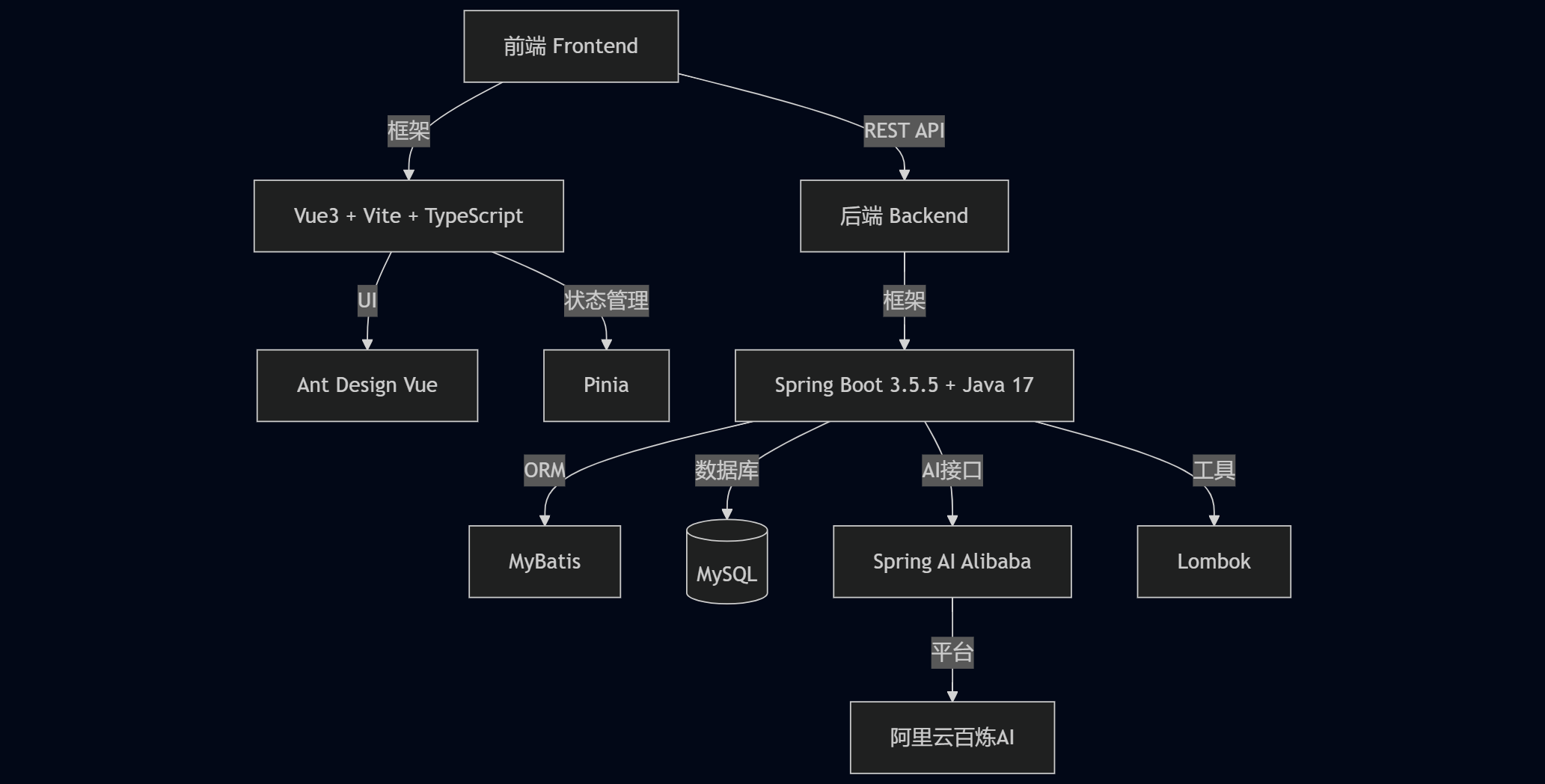
**配置与安全**

AI 密钥与配置放在后端（application.yml）通过环境变量注入，敏感信息不进仓库

认证采用 token-based 机制（前端持有 token，后端校验）。（项目 router 中已包含认证路由配置，提示已有认证需求）

数据隐私：对话日志/错题属于用户敏感学习数据，导出/清除应提供用户操作权限并有审计记录

### 2.1.5 技术栈确认



技术栈确认图

**前端**

Vue 3（Composition API）— SPA 框架

TypeScript — 静态类型

Vue Router — 路由

Pinia — 状态管理

Ant Design Vue — UI 组件库

**后端**Java 17（运行时）

Spring Boot 3.5.5（核心后端框架）

Spring Web (MVC) — REST API

MyBatis Spring Boot Starter — ORM/Mapper 层

MySQL Connector/J — 与 MySQL 连接

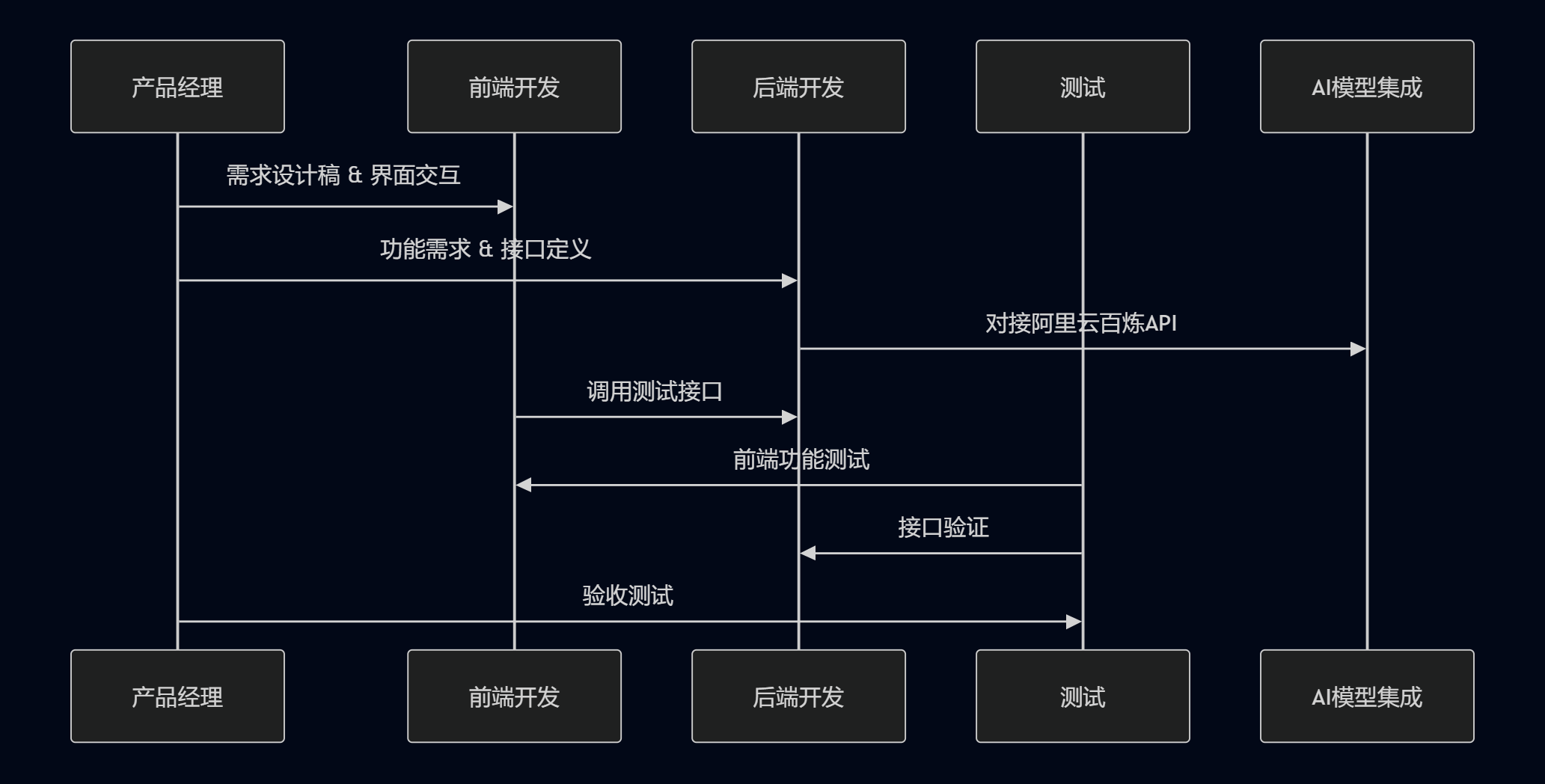
Spring AI Alibaba / 阿里云百炼平台集成（用于 AI 能力调用）

Lombok — 简化实体/DTO/VO 等 Java 样板代码

**开发 / 构建说明**

前端：npm install / npm run dev / npm run build

后端：Maven 构建：mvn clean install / mvn spring-boot:run



项目开发泳道图

## **2.2 详细设计**

简要描述主要功能的详细设计，包括类图（包括类之间的关系）、顺序图等等

2.2.1 前端详细设计

2.2.1.1 引言

2.2.1.1.1 编写目的

本章节旨在详细阐述 AI学习搭子系统前端部分 的设计思路、架构方案与实现细节。通过对页面结构、功能模块、界面设计、交互逻辑、状态管理及数据通信流程的系统说明，使读者能够清晰理解前端系统的整体构成与技术实现路径。

本章节主要服务于以下读者对象：

（1）前端开发人员：用于指导具体的组件开发、接口联调与样式实现；

（2）后端开发人员：了解前端数据调用方式与交互逻辑，以便于接口设计与联调；

（3）测试人员：理解系统功能结构与模块边界，用于制定测试方案与验证标准；

（4）项目管理者与设计人员：掌握前端架构全貌，便于后续功能扩展与版本规划。

本章节在系统设计文档体系中处于承上启下的关键位置，承接《系统概要设计》中提出的前端功能需求与总体架构思路，为后续的“功能实现”“测试设计”等章节提供具体的技术依据与可操作指南。通过该文档，可确保前端开发过程的一致性、可维护性与可复用性，并在项目后续迭代中提供技术追溯的依据。

2.2.1.1.2 背景

（1）待开发软件系统的名称：AI学习搭子系统

（2）任务提出者：

本系统的开发源于当下高校学生在自学过程中普遍存在的痛点：学习孤独感强、缺乏有效的知识协作与反馈机制、学习资源分散且难以高效利用。随着大语言模型（LLM）技术的迅速发展，智能化学习助手逐渐成为可能。因此，本项目旨在构建一个基于LLM的智能学习伙伴平台，通过自然语言交互、知识问答、个性化学习推荐等功能，帮助用户更高效地学习知识、巩固技能，并与“AI学习搭子”进行深度对话和互动式学习。

（3）开发者：

由本项目小组独立开发完成。开发团队包括项目负责人（项目经理）、前端开发人员（负责界面设计实现与交互逻辑）、后端开发人员（负责接口设计与数据处理）、算法与模型工程师（负责大语言模型的接入与优化）等成员。其中，前端部分采用 Vue 全家桶（Vue3 + Vue Router + Pinia）\*\* 开发，结合 Ant Design Vue 组件库、Axios 网络请求库与 Vite 构建工具，实现现代化的界面交互、良好的响应式体验与高效的前后端通信。

（4）用户：

主要用户群体为高校学生、自学者及对人工智能辅助学习感兴趣还有为学习和就业焦虑的用户。系统为用户提供学习问答、学习记录归档、AI学习对话、知识分析与总结等功能，帮助用户在个性化场景中实现智能化学习。

（5）运行该程序系统的计算中心：

系统部署于云服务器环境（如阿里云或腾讯云）或本地，通过后端 API 与前端 Web 客户端交互。计算中心提供必要的服务器计算资源、数据库服务及网络环境支持，确保系统在高并发访问下仍具备良好的稳定性、安全性与响应性能。系统前端通过 HTTPS 协议访问后端服务，实现用户端的安全、流畅体验。

2.2.1.1.3 定义

（1）LLM：Large Language Model，大语言模型，如 ChatGPT、GLM 等。

（2）前端：系统中负责与用户直接交互的部分，实现界面显示、用户输入、数据可视化。

（3）接口 API：前后端之间通过 HTTP 进行数据通信的约定。

2.2.1.1.4 参考资料

（1）Ant Design Vue 文档 https://antdv.com

2.2.1.1.5 成本分析

本节从前端视角，对人力成本、时间成本与工具/运行成本进行说明。

1. 人力成本：前端开发由 1 名开发者承担，覆盖 UI 设计与实现、页面逻辑、状态管理、接口联调与部署。
2. 时间成本：基于 Vite + Ant Design Vue，依赖轻、构建快，上手与迭代成本低。
3. 工具与运行成本：开发工具（VS Code、Node.js、Git、浏览器调试工具）；运行环境（Nginx 或 Vite Preview 部署）。

2.2.1.1.6 进度计划

| **阶段** | **时间范围** | **主要任务** | **产出物** |
| --- | --- | --- | --- |
| 阶段一 | 第 1–2 天 | 环境搭建、UI 规范确定、组件结构规划 | 页面原型、组件结构图 |
| 阶段二 | 第 3–7 天 | 功能开发（聊天、番茄钟、学业/职业设置等核心页面） | 可运行的前端页面 |
| 阶段三 | 第 8–10 天 | 数据联调、状态管理完善、用例与边界处理 | 端到端功能闭环 |
| 阶段四 | 第 11–13 天 | 响应式适配、性能优化、打包与部署 | 前端最终版本与部署说明 |

2.2.1.2 系统结构设计

2.2.1.2.1 前端整体架构

前端采用组件化开发思想，整体结构分为以下层次：

1. 视图层（View）：负责用户界面展示，如登录页、学习搭子页、个人中心页。
2. 逻辑层（Logic）：通过 Pinia 管理全局状态（如用户信息、聊天记录）。
3. 数据层（Data）：通过 Axios 与后端接口交互，获取或提交数据。
4. 路由层（Router）：通过 Vue Router 控制页面跳转与权限验证。

目录结构示意：

AI\_LearningPartner/

├── readme.md # 项目概要说明

├── package.json # 顶层依赖

└── vue-project/ # 核心前端项目目录

├── package.json # 完整前端依赖与脚本

├── index.html # 应用入口 HTML

├── vite.config.ts # Vite 配置

├── src/

│ ├── App.vue # 根组件

│ ├── main.ts # 应用入口 TS

│ ├── router/ # 路由配置（懒加载）

│ ├── stores/ # Pinia 状态管理

│ └── views/ # 页面视图组件

│ ├── AppLayout.vue # 应用主布局

│ ├── Auth.vue # 登录/注册页

│ ├── ChatView.vue # 主页（聊天答疑）

│ ├── FocusView.vue # 番茄钟与目标页

│ ├── WorkView.vue # 职业规划页

│ ├── UserSettingsView.vue # 个人设置页

│ └── StudyView.vue # 学业规划页

└── src/api/ # API 封装

2.2.1.2.2 功能模块设计部署图

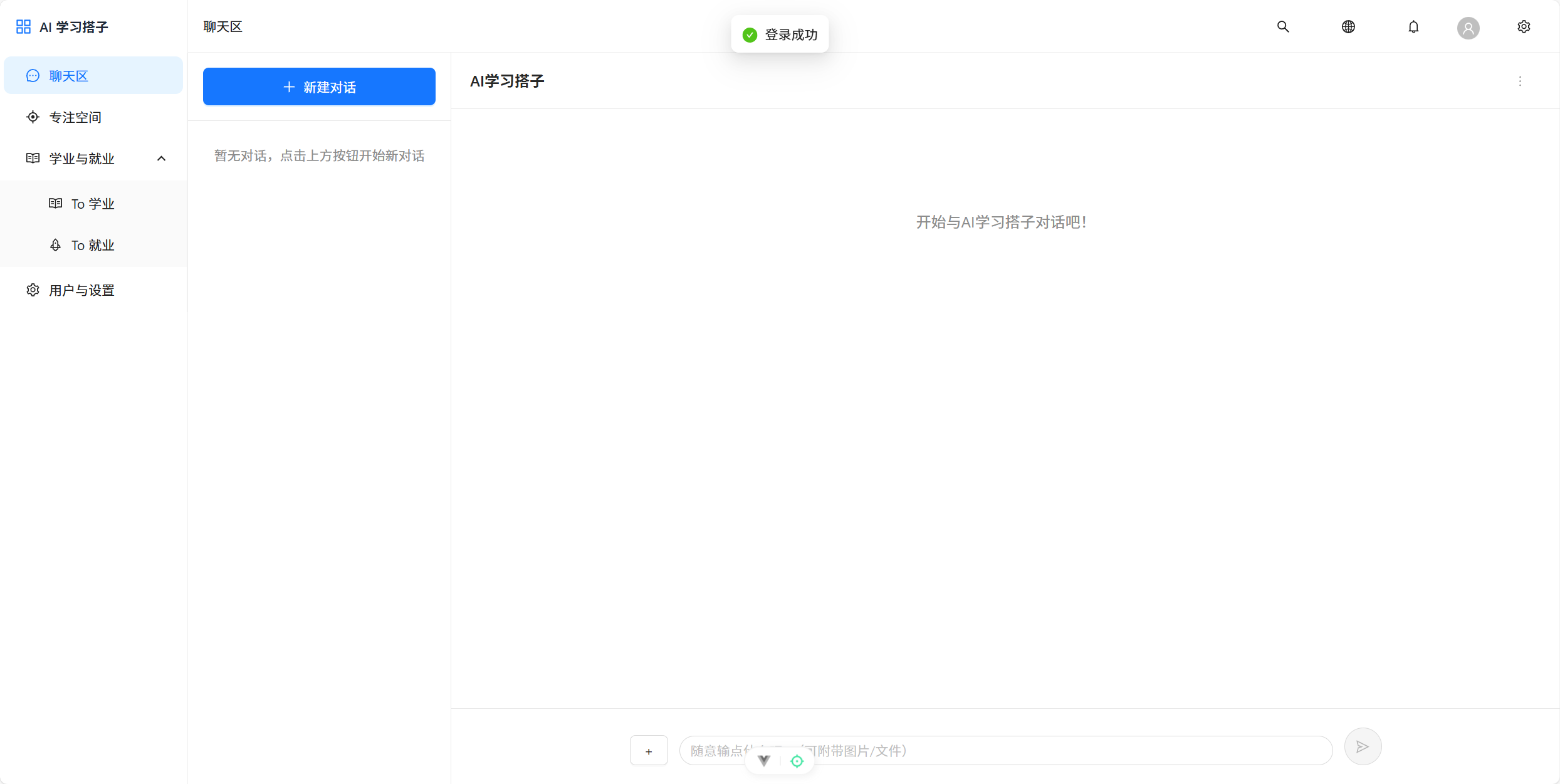
2.2.1.2.3 界面设计（包括美学设计）



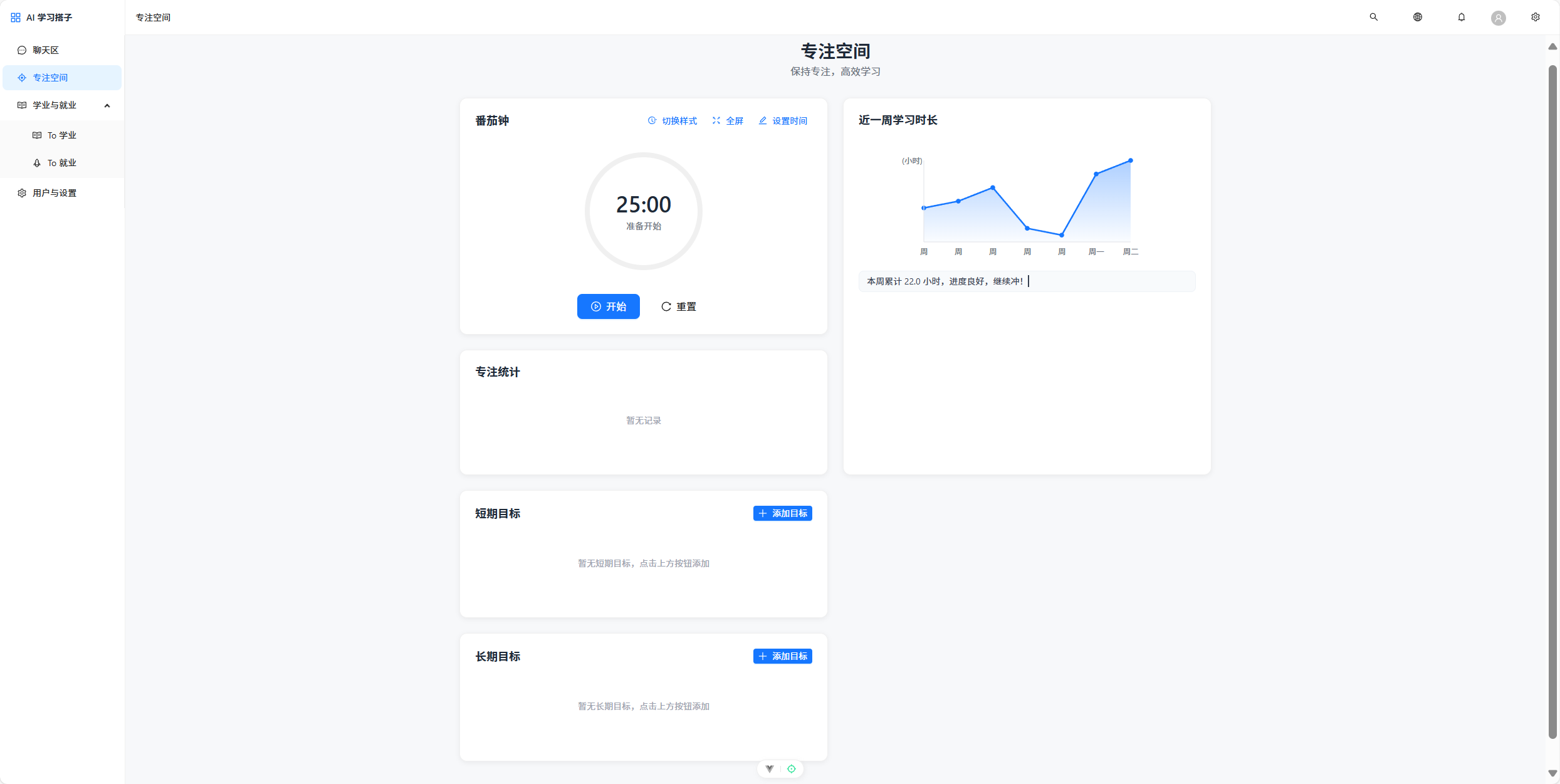
图 3-2-1登录界面图



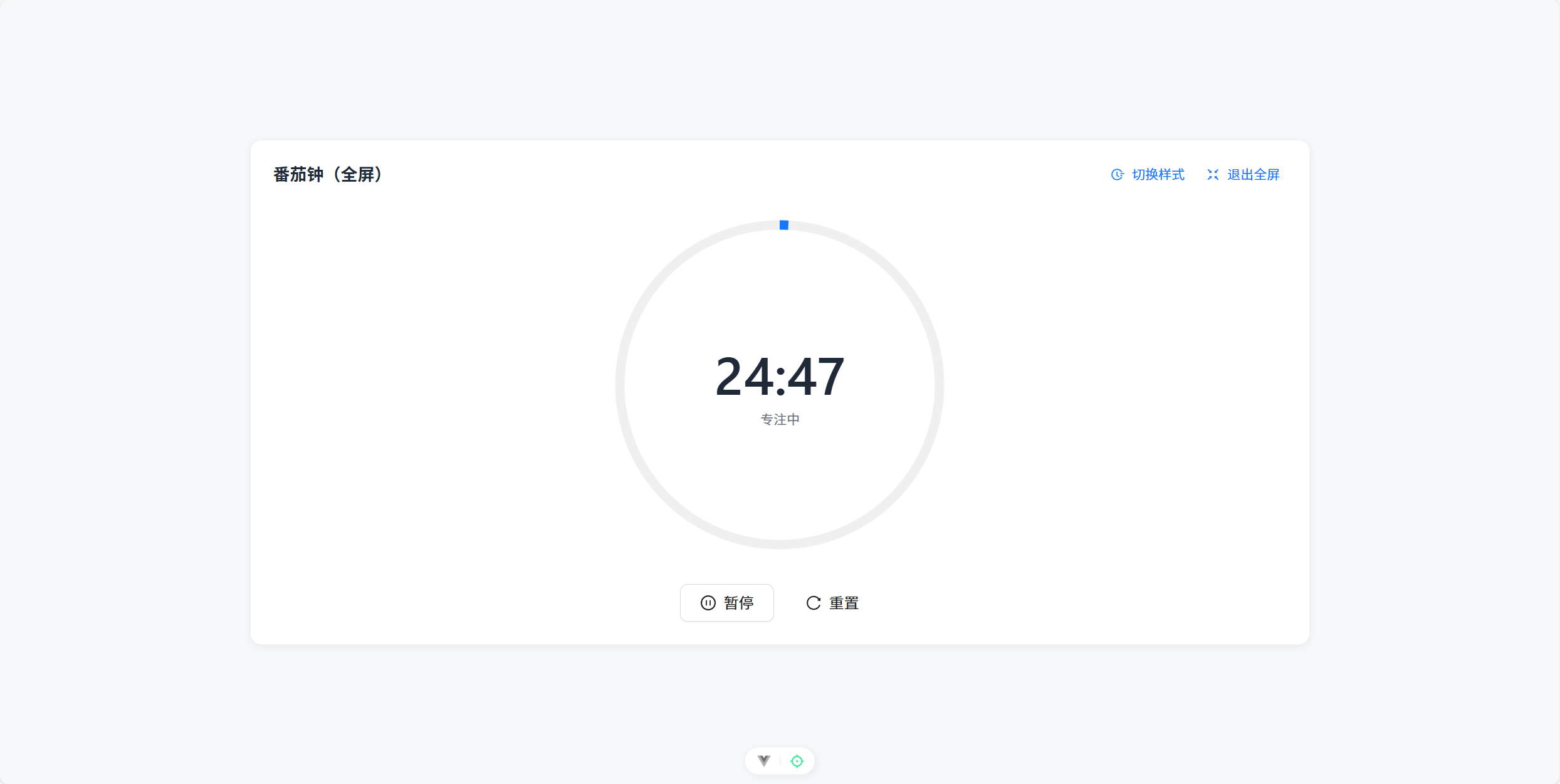
图 3-2-3 注册界面

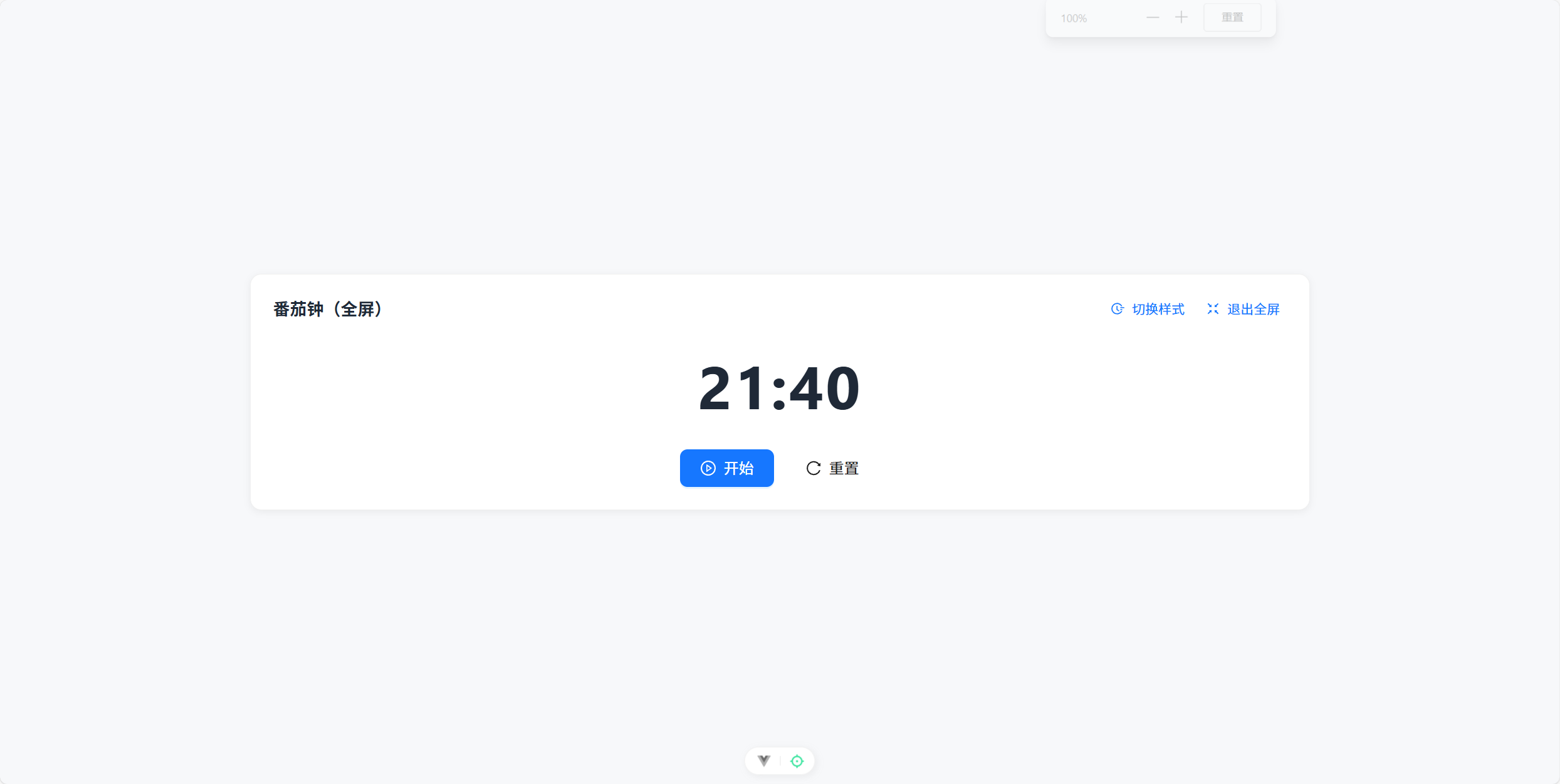


主页面（聊天页面）



专注页面

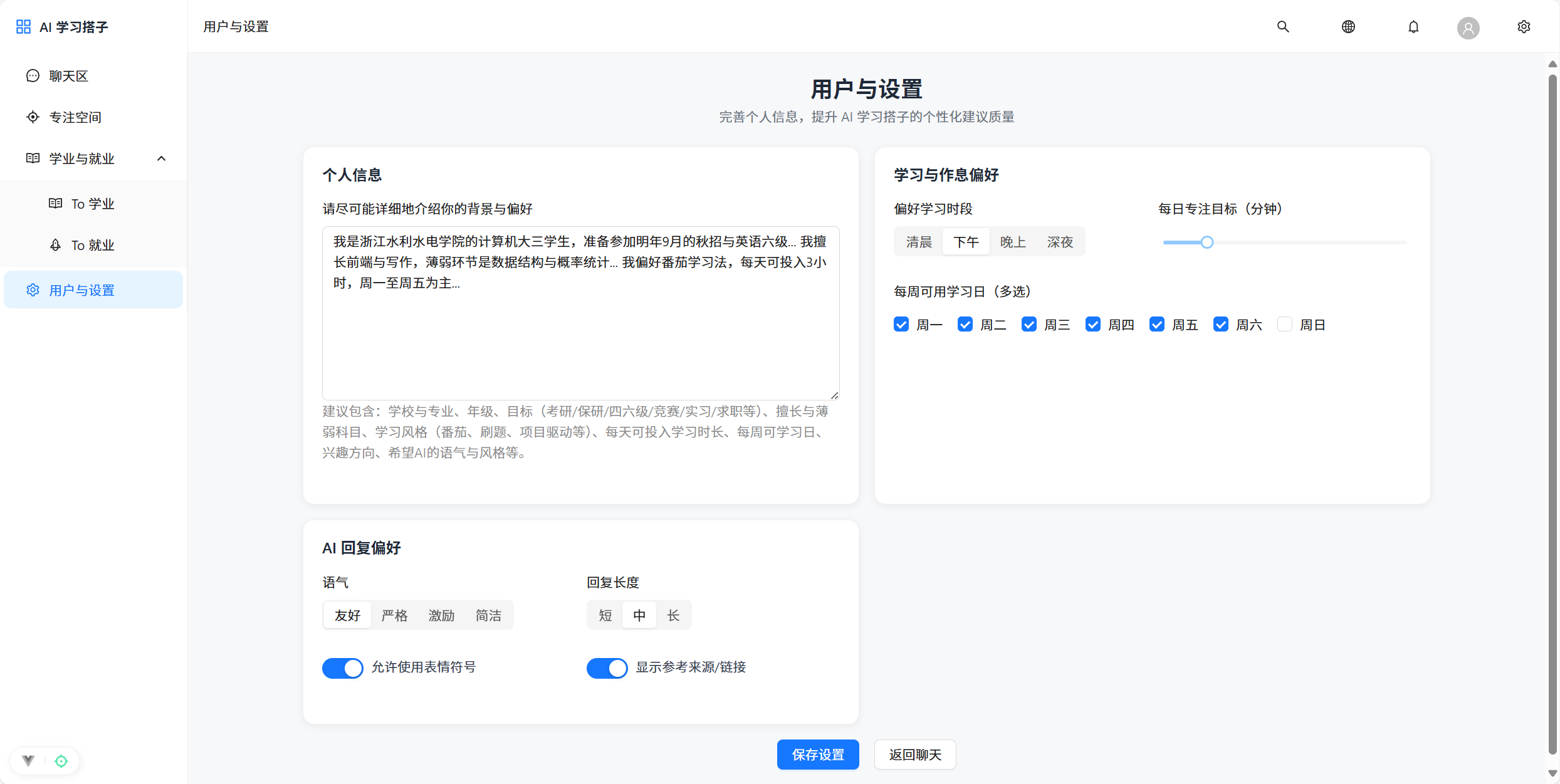






学业页面

就业页面



设置页面

2.2.1.3 功能模块设计

2.2.1.3.1 登录注册模块

2.2.1.3.1.1 程序描述

基于 Ant Design Vue 的登录/注册双 Tab 界面（a-tabs），提供邮箱+密码登录与注册能力。前端完成表单校验、加载态与错误提示；与后端交互后在成功时将用户信息写入本地并跳转主界面。

2.2.1.3.1.2 功能

（1）登录：邮箱+密码登录，支持“记住我”；

（2）注册：邮箱+密码+确认密码，前端校验两次密码一致；

（3）统一的加载按钮态与失败提示；

（4）成功后跳转聊天页并持久化最小用户信息。

2.2.1.3.1.3 输入项

（1）登录：邮箱、密码、记住我；

（2）注册：邮箱、密码、确认密码。

2.2.1.3.1.4 输出项

（1）登录/注册成功或失败的状态与提示；

（2）成功后进入主界面；

（3）本地持久化的用户信息（如id, email，信息不展示在前端）。

2.2.1.3.1.5 接口

（1）GET http://localhost:8080/login?email=...&password=...

（2）GET http://localhost:8080/register?email=...&password=...

2.2.1.3.1.6 逻辑流程

（1）用户选择登录或注册并填写表单；

（2）前端进行必填项与一致性校验（注册需校验两次密码一致）；

（3）发起认证请求并显示加载状态；

（4） 成功：保存必要的用户会话信息并跳转至主界面；

（5） 失败：提示错误信息并结束加载。

2.2.1.3.1.7 UI 与组件实现

（1）结构：卡片式容器 + 页签切换（登录/注册）；

（2）表单：垂直布局，包含邮箱、密码、确认密码等输入项与校验提示；

（3）行为：提交时显示加载状态；登录提供“记住我”和“忘记密码”入口；

（4）响应式：在手机、平板与桌面端保持良好布局与触控体验。

2.2.1.3.2 主页（聊天答疑）模块

2.2.1.3.2.1 程序描述

用户输入学习相关问题，系统通过后端转发/调用 LLM，返回智能回答并在前端展示多轮对话。支持文件/图片上传与预览，显示“正在输入”状态；AI 回复采用服务端事件或等效机制进行流式增量展示（前端按块更新）。

2.2.1.3.2.2 功能

（1）支持多轮对话与历史记录加载；

（2）输入框支持回车发送与快捷键；

（3）对话消息自动滚动到最新；

（4）显示消息状态（发送中 / 已送达 / 失败）；

（5）支持附件上传与图片预览；

（6）模拟流式回复与“正在输入”占位。

2.2.1.3.2.3 输入项

（1）用户文本输入；

（2）可选：图片/文件附件；

2.2.1.3.2.4 输出项

（1）AI 返回的文本；

（2）更新的会话预览（最后一条消息与时间）。

2.2.1.3.2.5 接口

（1）GET http://localhost:8080/conversation/list：获取会话列表

（2）GET http://localhost:8080/conversation/{id}/messages：获取指定会话消息

（3）POST http://localhost:8080/conversation/create：创建新会话并返回会话ID

（4）DELETE http://localhost:8080/conversation/{id}：删除会话

（5）POST http://localhost:8080/conversation/{id}/message：保存一条消息

（6）POST http://localhost:8080/aliyunbailian/application/chat（返回 text/event-stream）：AI 流式回复

2.2.1.3.2.6 逻辑流程

（1）打开页面时加载会话列表；选中会话后加载消息列表；

（2）用户输入消息（可附带附件）并发送；

（3）若当前未存在服务端会话，则先创建后获取会话ID；

（4）将用户消息追加到界面并持久化到后端；

（5） 发起 AI 对话请求（采用事件流或等效机制），逐步接收增量文本并更新界面；

（6）AI 回复完成后持久化到后端，并更新会话预览（摘要与时间）；

（7）请求异常时提示错误，保留本地消息以便用户重试；删除会话时同步更新界面状态。

2.2.1.3.2.7 UI 与组件实现

（1）布局：左侧为会话列表区（含新建、删除、最近一条摘要与时间），右侧为聊天区（头部、消息列表、输入区）；

（2）消息展示：区分用户与助手两种消息样式，支持时间戳与附件预览；

（3）输入区：提供文本输入、附件选择与发送操作；发送时禁用或显示进度，减少重复提交；

（4）正在回复：在助手回复生成期间显示“正在输入”指示，提升反馈感；

（5）空态与异常：在无会话/无消息/请求失败时提供空态与提示文案；

（6）响应式：适配手机、平板与桌面端，在窄屏时优先展示核心区域并简化列表信息。

2.2.1.3.3 番茄钟与目标模块（Focus）

2.2.1.3.3.1 程序描述

提供圆形番茄钟计时器、目标管理与专注统计展示，基于用户最近一周的专注时长生成个性化评价与建议。

2.2.1.3.3.2 功能

（1）圆形番茄钟：开始/暂停/重置；

（2）目标管理：短期/长期目标展示与添加；

（3）专注统计：饼图展示不同内容时长占比；

（4）个性化建议：每日定时生成提示与建议。

2.2.1.3.3.3 输入项

（1）番茄钟参数与目标项。

2.2.1.3.3.4 输出项

（1）当日专注用时、分布占比与AI建议文案。

2.2.1.3.3.5 接口

（1）GET http://localhost:8080/focus/show-goal：获取目标列表（短期/长期）

（2）GET http://localhost:8080/focus/short-goal：创建短期目标（参数：标题、内容）

（3）GET http://localhost:8080/focus/long-goal：创建长期目标（参数：标题、内容）

（4）GET http://localhost:8080/focus/completed：更新目标完成状态（参数：目标ID、是否完成）

（5）GET http://localhost:8080/focus/delete-goal：删除目标（参数：目标ID）

2.2.1.3.3.6 逻辑流程

（1）进入页面加载近一周学习时长与目标列表；

（2）用户设置番茄钟时长并开始计时，暂停/继续/重置受控；

（3）计时结束记录一次完成情况（含本次专注主题）并更新统计；

（4）用户在目标区新增短期/长期目标，或切换完成状态、删除目标；

（5）相关操作同步持久化到后端，界面实时刷新；

（6）异常时提示并允许用户重试或稍后处理。

2.2.1.3.3.7 UI 与组件实现

（1）布局：上方为圆形计时器与控制区（开始/暂停/重置），下方为目标管理与统计图表区；

（2）计时器：以显著的时间显示为核心，提供清晰的状态与交互反馈；

（3）目标管理：分短期/长期列表展示，支持新增与状态切换，提供空态提示；

（4）统计展示：采用图表展示不同内容类别的专注占比与趋势；

（5）提示与建议：在指定时间提醒用户复盘，并展示个性化建议文案；

（6）响应式：在移动端优先显示计时核心与今日目标，图表与建议按需折叠。

2.2.1.3.5 学业模块

2.2.1.3.5.1 程序描述

展示学期课程、重要考试/赛事信息，并基于个人情况推荐学习视频与建议。

2.2.1.3.5.2 功能

（1）课表/考试日期上传组件（图片或常见课表格式）；

（2）本学期课程展示（mock 数据）；

（3）个性化视频推荐列表（含图片、标题、链接）；

（4）重要考试/赛事推送与 AI 建议。

2.2.1.3.5.3 输入项

（1）考试新增：名称、日期（及可选提醒标记）。

2.2.1.3.5.4 输出项

（1）近期考试列表（名称、日期、剩余天数等）；

（2）学业概览与提醒文案；

（3）个性化视频推荐条目（标题、封面、链接）。

2.2.1.3.5.5 接口

（1）GET http://localhost:8080/exam/list：获取近期考试列表

（2）DELETE http://localhost:8080/exam/delete/{examId}：删除考试

（3）POST http://localhost:8080/exam/add：新增考试

2.2.1.3.5.6 逻辑流程

（1）进入页面后拉取近期考试列表与学业概览；

（2）用户可新增自定义考试或删除已有考试，成功后刷新列表；

（3）展示个性化视频推荐与学业提醒，辅助学习规划。

2.2.1.3.5.7 UI 与组件实现

（1）布局：上方为学业概览与提醒，中间为课程与考试区，下方为视频推荐与资讯；

（2）课表上传：支持文件/图片上传并解析结果回显；

（3）列表展示：课程、考试以分组列表呈现，突出日期与重要性；

（4）推荐区：以卡片形式展示视频封面、标题与外链；

（5）空态与异常：在无数据或加载异常时提供指引与重试；

（6）响应式：在移动端按模块折叠展示核心信息。

2.2.1.3.6 就业模块

2.2.1.3.6.1 程序描述

支持简历上传、岗位目标设定，生成职业匹配度与学习路径建议，。

2.2.1.3.6.2 功能

（1）简历上传 + 模拟“AI 简历建议与修改”；

（2）目标岗位设定；

（3）技能差距分析与提升建议；

2.2.1.3.6.3 输入项

（1）理想岗位：岗位名称文本；

（2）简历上传：文件。

2.2.1.3.6.4 输出项

（1）本地保存的简历列表；

（2）岗位输入框当前值；

（3）AI 就业建议列表。

2.2.1.3.6.5 接口

当前页面不依赖后端接口，使用本地存储保存与读取简历；后续可接入后端存储与简历解析/打分服务。

2.2.1.3.6.6 逻辑流程

（1）进入页面后从本地存储加载已保存的简历列表；

（2）用户可填写理想岗位文本以驱动后续内容呈现；

（3）用户选择简历文件后可保存至本地存储，并在列表中展示下载/删除入口；

（4）页面右侧显示 AI 就业建议，支持按岗位定制（可扩展为服务端）；

（5）后续可扩展：将本地保存替换为后端存储，并新增简历解析、评分与岗位匹配度评估。

2.2.1.3.6.7 UI 与组件实现

（1）布局：顶部为个人目标与进度，中部为简历与评估区，底部为岗位推荐与资讯；

（2）简历区：支持上传与结果反馈，提供下载/查看入口；

（3）评估区：以问卷或结果卡片展示，突出匹配度与建议要点；

（4）岗位与路径：采用列表/时间轴展示能力差距与学习路径；

（5）空态与异常：无简历/评估时给出引导，异常时提示重试；

（6）响应式：在移动端优先呈现目标与建议摘要。

2.2.1.3.7 个人设置模块

2.2.1.3.7.1 程序描述

设置用户个人信息、头像与 AI 身份配置。

2.2.1.3.7.2 功能

（1）学校/专业等信息设置；

（2）用户与 AI 的个性化头像设置；

（3）AI 身份与风格设定。

2.2.1.3.7.3 输入项

（1）个人信息：自由文本简介；

（2）学习与作息：偏好学习时段、每日专注目标时长、每周可用学习日（多选）；

（3）AI 回复偏好：语气、回复长度、是否允许表情、是否显示参考来源。

2.2.1.3.7.4 输出项

（1）已保存的个人设置（上述各项的当前值）；

2.2.1.3.7.5 接口

（1）GET http://localhost:8080/set/show：加载用户设置

（2）POST http://localhost:8080/set/save：保存用户设置（请求体为 JSON，包含个人信息、作息与偏好相关字段）

2.2.1.3.7.6 逻辑流程

（1）进入页面后请求加载用户设置，填充到各输入项；

（2）用户修改信息与偏好后点击保存，提交到后端；

（3）成功时提示保存成功并保留当前值，失败时提示错误并允许重试；

（4）后续可扩展：未登录/未授权的统一处理与跳转。

2.2.1.3.7.7 UI 与组件实现

（1）布局：以分区卡片形式组织基础资料、头像与偏好设置；

（2）表单：提供必要的输入项与校验，支持保存与重置；

（3）头像与形象：支持图片选择与预览；

（4）偏好：以开关/单选等形式设置个性化行为；

（5）空态与异常：在首配或保存失败时给出提示与重试；

（6）响应式：在小屏幕上按区块纵向排列并保留操作按钮的易触达性。

2.2.1.4 界面与交互设计

2.2.1.4.1 风格设计

整体遵循 Ant Design 设计语言，主色调为蓝色（#1677ff）与白色，现代简约，强调留白与信息层级；保证手机、平板与 PC 的统一视觉与交互。

2.2.1.4.2 UI 框架

Ant Design Vue + 自定义样式（Less/SCSS），图标使用 Ant Design Icons；主题色与组件尺寸统一配置。

2.2.1.4.3 布局与响应

（1）响应式布局：桌面端左侧导航 + 主内容区域，移动端底部导航或汉堡菜单；（2）关键交互（发送消息、上传/预览、删除）提供动画反馈与二次确认（删除时）。

2.2.1.4.4 交互细节

（1）登录成功自动保存 token 并跳转聊天页；

（2）聊天支持回车发送、AI “正在输入”占位、消息自动滚动；

（3）附件上传过程显示进度与预览；

（4）番茄钟开始/暂停/重置，结束提醒并更新统计；

2.2.1.5 性能与测试

2.2.1.5.1 性能优化

（1）路由组件懒加载（Vue Router 动态 import）；

（2）对话/列表采用分页或虚拟滚动；

（3）输入与搜索使用防抖/节流；

（4）Vite 构建优化与按需引入 AntD 组件。

2.2.1.5.2 测试计划

（1）类型检查：vue-tsc；

（2）单元测试：关键组件（Auth、ChatMessage、FocusTimer）渲染与行为（Vitest + Vue Test Utils）；

（3）集成/接口测试：模拟 API 返回，验证加载、错误重试与空态；

（4）用户测试：小规模可用性测试，收集问题与建议。

2.2.1.5.3 尚未解决的问题

（1）极端长对话的滚动与渲染性能仍需优化；

（2）多媒体消息（音频/文件）处理链路依赖后端与存储策略；

（3）流式回复需要后端 SSE/WebSocket 支持。

## **2.3 功能实现**

主要功能的实现，可以按以下示例方式进行说明：

### **2.3.1**

，其核心功能实现的伪代码如下所示：

|  |
| --- |
|  |

本功能的界面实现如下：

# **3.设计总结**

## **3.1 设计过程碰到的难点与解决方法**

*不能少于5点，撰写时删除此行*

（1）问题1：

解决方案：

（2）问题2：

解决方案：

（3）问题3：

解决方案：

（4）问题4：

解决方案：

（5）问题5：

解决方案：

（6）问题6：

解决方案：

（7）问题7：

解决方案：

（8）问题8：

解决方案：

（9）问题9：

解决方案：

（10）问题10：

解决方案：

## **3.2 成果与收获**

*不能少于600字，，撰写时删除此行*

|  |
| --- |
| 评语：报告撰写选择一项。  成绩：选择一项。 |