

基于开源 AI 大模型的教学实训智能体软件开发指南

第十四届中国软件杯 A7 赛题分析与实现方案

1 实现目标

本项目旨在构建一套基于开源大语言模型的教学实训智能体系统，具备以下核心能力：

- 教师端：智能备课设计、自动考核生成、学情数据分析；
- 学生端：基于知识库的学习问答助手、实时练习评测、错题解析与资源推荐；
- 管理端：用户管理、教学资源管理、学习大数据可视化分析。

本系统需基于开源大模型（如 ChatGLM 等）完成本地部署或 API 接入，结合教材内容构建知识库并实现问答增强（RAG）能力。

2 赛题分析与难点解析

2.1 任务拆解

模块	关键能力
教师端	自动备课、题目生成、答案标注与知识点分析
学生端	学习问答助手、随练推题、练习纠错与反馈
管理端	用户与资源管理、教学效率与学习成效可视化

2.2 技术路径关键词

- 自然语言处理（NLP）与内容生成（NLG）
- 大模型集成与本地知识库结合（RAG 架构）
- 文本向量化与相似度检索（BGE/FAISS）

- 教育数据分析与知识点归因
- 快速前端交互界面构建 (Streamlit/Vue)

2.3 关键难点分析

1. 大模型与本地知识结合：向量召回 + 大模型生成，需构造高质量 Prompt 模板；
2. 知识库构建：文本清洗与分块、向量化与索引管理；
3. 内容生成的溯源与可控性：确保题目和答案具备对应教材出处；
4. 系统集成：模块解耦与前后端数据流整合；
5. 时间能力匹配：对学生选手要求设计最小可行原型 (MVP) 路径。

3 开发路径分解

3.1 项目技术架构

建议采用模块化分层架构，前后端分离部署：

前端界面 ← Streamlit/Vue + ECharts

↓

后端服务 (FastAPI)

模型API封装/知识召回/逻辑控制等

↓

向量数据库 ← 课程内容向量化
(FAISS)

↓

本地部署模型 ← ChatGLM等

3.2 推荐技术栈

- 大模型：ChatGLM3 / Baichuan / MiniCPM
- 向量模型：bge-small-zh / m3e-small
- 向量库：FAISS
- 后端：FastAPI / Flask
- 前端：Streamlit / Vue3 + ElementPlus
- 数据库：SQLite 或 PostgreSQL

4 功能模块实现步骤

4.1 知识库构建

1. 提取教材内容：使用 PyMuPDF 读取 PDF 文本；
2. 文本切块：按段落或节块划分，大小 200-500 字；
3. 向量生成：调用中文 Embedding 模型生成向量；
4. 向量索引：用 FAISS 构建召回系统。

4.2 大模型调用

- 使用 ChatGLM 等模型进行本地部署；
- 构造 Prompt 模版，将召回段落融合进输入；
- 接收模型输出，返回回答或生成内容。

4.3 教师端模块

- 备课设计：根据课程大纲生成授课建议；
- 考核生成：基于知识点出题（含编程题）；
- 学情分析：对学生答题数据进行聚类与归因分析。

4.4 学生端模块

- 学习问答：输入问题，召回段落 + 模型回答；
- 练习系统：推送适应性题目，评测与反馈；
- 错题推荐：输出改错建议与相关章节。

4.5 管理与可视化模块

- 用户管理：支持不同权限用户注册与登陆；
- 资源管理：教师上传与导出教学资源；
- 数据大屏：展示使用频率、正确率、薄弱点等指标。

5 学生实战路线与提交材料建议

5.1 推荐 MVP 路径

- 优先实现“学生问答助手”与“教师出题助手”两个模块；
- 使用 Streamlit 开发快速界面，避免复杂网页工程；
- 使用静态结果模拟未完成模块的逻辑与数据流。

5.2 提交材料结构

1. 源码目录（含模型接口、知识库、前端页面）
2. 系统设计说明文档（PDF）
3. 使用手册（Word/PDF）
4. 项目展示 PPT
5. 演示视频（不超过 7 分钟）

5.3 建议开发时间表（7 周）

- 第 1 周：模型选型、课程资料清洗；
- 第 2 周：知识库构建、向量召回实现；
- 第 3 周：RAG 模块 + 问答系统搭建；

- 第 4 周：教师端功能开发；
- 第 5 周：学生端练习与评测模块；
- 第 6 周：管理端与数据可视化开发；
- 第 7 周：打包文档、录制视频、提交成果。

赛题分析 by 谢仕鹏老师