

DevPulse

智能问答助手

软件工程实践团队作业 · 选题说明

01 内容概览

选题名称	DevPulse 智能问答助手
一句话描述	把团队文档、课程资料和常见问题整理成可检索、可追溯、可反馈的知识库，并通过检索增强生成技术提供可信回答。
目标用户	课程班级、学生团队、社团、实验室小组和小型项目组
核心目标	让用户更快获得有依据、可追溯、能持续更新的答案
团队名称	DevPulse

文档结构

章节	内容
02 选题背景	问题来源、用户痛点与项目边界
03 NABCD 分析	需求、做法、好处、竞品与交付方式
04 功能与技术路线	核心功能、扩展功能、RAG 流程与系统架构
05 团队绩效考核	考核原则、维度、计算方式与执行节奏
06 可行性与风险	技术、数据、时间、成本分析与风险应对
07 团队愿景	项目愿景、阶段目标与结语

02 选题背景

问题从哪里来

在课程学习、社团工作和项目协作中，信息通常散落在聊天记录、在线文档、会议纪要、课程资料和网页中。成员遇到问题时，往往需要在多个平台反复搜索，或者直接询问已经处理过同类问题的人。这个过程既消耗个人时间，也消耗团队中其他成员的时间。

三个主要痛点

- **查找成本高：** 同一个问题可能被不同成员重复提问，资料越多，定位越慢。
- **答案难追溯：** 只给结论、不说明来源，使用者难以判断答案是否可信。
- **知识难沉淀：** 问题解决后没有形成可复用的知识资产，人员变动后经验容易流失。

为什么采用检索增强生成

大语言模型可以生成流畅的回答，但如果只依赖模型自身知识，容易出现事实过期、来源不明和内容幻觉。检索增强生成（RAG）先检索与问题相关的文档，再让模型基于检索结果组织答案，能够在一定程度上缓解这些问题。DevPulse 不做通用聊天机器人，而是围绕团队知识库，把“找到资料、理解资料、给出依据”连成一条完整链路。

项目边界

项目要做	项目暂不追求
围绕团队私有资料的检索与问答	替代所有搜索引擎和通用聊天工具
答案引用来源，支持核对原文	让模型在没有依据时强行给出答案
知识库更新、反馈和缺口分析	一次性支持所有复杂企业流程
适合课程组和小型团队的轻量部署	首版就追求大规模高并发

03 NABCD 分析

N: Need, 需求

用户的真实需求不是“再做一个聊天机器人”，而是用更低成本获得有依据、可追溯、能持续更新的答案。需求可以拆解为：支持多格式资料导入；支持自然语言提问；答案需要附带来源片段；支持多轮追问；支持知识更新、删除和权限管理；支持用户反馈并形成改进闭环。

A: Approach, 做法

步骤	做法
1. 知识入库	解析 Markdown、TXT、PDF、Word、网页和问答对，进行清洗、分块、向量化与元数据登记
2. 检索增强	使用向量检索和关键词检索混合召回，再通过重排序筛选最相关的内容
3. 答案生成	将问题、对话历史和检索片段交给大语言模型，要求模型只依据给定材料作答
4. 引用展示	答案下方展示来源文档和段落，点击后可以定位到原文片段
5. 反馈闭环	记录低质量回答和未命中问题，供管理员补文档或调整检索策略
6. 权限管理	管理员维护知识库、成员和访问权限，普通用户只能查看被授权的内容

B: Benefit, 好处

- **对使用者**：不用在多份文档和聊天记录中反复搜索，可以直接获得带来源的回答。
- **对团队**：新成员能够更快了解项目背景，减少重复沟通和重复答疑。
- **对知识管理**：文档、问答和反馈形成闭环，知识库可以持续更新。
- **对工程实践**：团队完整经历需求分析、架构设计、数据处理、接口开发、前端实现、测试和部署。
- **对可信度**：答案有引用来源，用户可以核对原文，降低盲目信任模型输出的风险。

C: Competitors, 竞品

竞品或替代方案	优点	不足与 DevPulse 的差异
通用聊天机器人	门槛低、回答快	不了解团队私有资料，来源不透明，文档更新后无法自动同步
传统关键词搜索	结果可解释、实现简单	依赖关键词匹配，难以理解同义表达和上下文
在线文档搜索	适合人工查找原始资料	需要用户自己阅读和归纳，不能直接生成带引用回答
企业知识库产品	功能完整、管理能力强	成本较高，配置复杂，对课程小组偏重

DevPulse 的差异点不是追求通用能力，而是围绕课程和团队协作场景，做一个轻量、透明、可解释的知识问答工具。用户可以看见答案从哪里来，也可以参与反馈和知识维护。

D: Delivery, 交付

阶段	交付重点	验收方式
第一阶段	需求、原型、数据库和接口设计	设计文档可评审，角色和模块划分清楚
第二阶段	文档导入、分块、向量化和检索	上传资料后可以被检索到
第三阶段	问答、引用、多轮对话和反馈	主链路可演示，答案可以回看来源
第四阶段	测试、部署、演示数据和项目文档	可访问、可复现、可讲解

04 功能与技术路线

核心功能

功能	说明
知识库管理	创建知识库，上传、查看和删除文档
文档处理	解析文本、切分片段、生成向量并记录来源
智能问答	输入自然语言问题，返回结构化答案
来源引用	展示答案引用的文档、段落和相似度信息
多轮对话	保留上下文，支持连续追问
用户反馈	对回答进行有用或无用评价，记录知识缺口

扩展功能

功能	价值
角色与权限	区分管理员、普通成员和访客，控制知识库访问范围
知识缺口分析	统计高频未命中问题和低分回答，提示补充资料
文档版本管理	记录文档更新历史，避免使用过期内容
回答质量评估	用固定问题集测试检索命中率、引用准确率和回答一致性

RAG 主流程



图 1 检索增强问答的五个关键环节

建议技术架构



层次	选型建议	选择理由
前端	Vue 3 + TypeScript	开发效率高，适合课程项目快速迭代
后端	FastAPI	接口清晰，方便接入 Python AI 生态
AI 编排	LangChain 或 LangGraph	便于组合检索、重排、提示词和生成流程
向量检索	Chroma 或 Milvus	先本地轻量运行，后续可以扩展
数据存储	MySQL + Redis	保存业务数据、会话和缓存
模型接口	OpenAI 兼容接口	便于切换不同大模型服务，避免绑定单一厂商
部署	Docker Compose	降低演示和云端部署成本

质量与评估指标

指标	目标或测量方式
检索命中率	固定问题集中，前 5 条检索结果包含正确答案来源的比例
引用准确率	答案引用的片段确实支持对应结论，抽查进行人工确认
回答一致性	同一问题在不同时间重复提问时，核心结论保持稳定
响应时间	普通问题在演示环境中的端到端返回时间
反馈闭环率	低分回答和未命中问题被记录、复核或补充资料的比例

05 团队绩效考核方案

团队绩效考核的目的，是让每个人的投入和贡献被看见，并帮助团队及时调整分工，而不是制造内耗。结合《构建之法》中关于团队绩效管理和成员投入程度的讨论，我们采用“过程记录为主、结果质量并重、互评补充、规则透明”的考核方式。

考核原则

- **公开透明**：考核维度、权重和数据来源在项目开始时向所有成员说明。
- **过程与结果并重**：不仅看最终代码，也看需求、设计、测试、文档、协调和复盘。
- **可追溯**：以任务看板、Git 提交、代码评审、会议记录和交付物为主要依据。
- **可申诉**：成员如果对评分有异议，可以在结果公布后 24 小时内提出，由团队共同复核。

考核维度

考核维度	分值	主要观察点
任务完成度	35	是否按约定完成分配任务，是否主动同步进度和风险
成果质量	25	代码、文档、设计、测试和演示材料的质量与可维护性
协作与沟通	20	是否及时回复、主动帮助、积极参与讨论、尊重他人意见
过程规范	10	是否遵守 Git 规范、会议记录、命名约定和交付要求
额外贡献	10	关键问题攻关、额外测试、知识分享、组织协调等

数据来源与计算方式

- 任务完成度、成果质量和过程规范以任务看板、Git 提交记录、代码评审和交付物检查为依据。
- 协作与沟通采用成员互评，去掉一个最高分和一个最低分后取平均；人数较少时直接取平均。
- 额外贡献由成员自荐或他人提名，团队讨论后确认。
- 个人绩效得分 = 任务完成度 + 成果质量 + 协作与沟通 + 过程规范 + 额外贡献。
- 当需要按贡献度分配团队成绩时，个人系数 = 个人绩效得分 ÷ 团队平均绩效得分。
- 个人系数限制在 0.8 至 1.2 之间，再按比例归一化，使所有成员系数之和等于团队人数。

示例：如果团队最终成绩为 90 分，某成员归一化后的个人系数为 1.05，则该成员成绩为 94.5 分；如果另一名成员系数为 0.95，则成绩为 85.5 分。具体是否启用贡献度分配，以老师后续作业要求为准。

执行节奏

节奏	执行方式
每周短会	确认本周任务、负责人和截止时间
会议记录	记录结论、待办和风险，发到团队群
双周互评	重点看协作问题和任务阻塞，不等到期末才反馈
阶段复盘	记录做得好的地方和需要改进的地方

06 可行性与风险

可行性分析

维度	判断	依据
技术可行性	较高	文档解析、向量检索和大模型调用都有成熟组件
数据可行性	较高	课程资料、项目文档和常见问答可以作为初始知识库
时间可行性	中等	先做单知识库、单轮问答和引用展示，再逐步扩展
成本可行性	较高	本地向量库配合按量调用的大模型接口，成本可控

主要风险与应对

风险	影响	应对措施
模型产生无依据回答	答案不可信	强制基于检索片段回答，展示来源，低置信度时提示核对原文
文档解析效果差	检索不到有效内容	支持多种解析器，保留原文页码或段落，允许人工修正
检索结果不相关	回答偏离问题	采用混合召回、重排序和固定测试集验证
接口延迟或失败	体验下降	增加超时、重试、缓存和降级提示
隐私与权限问题	资料泄露	知识库分级授权，敏感文档不入公共库，接口统一鉴权
团队进度不一致	交付延期	每周检查任务看板，提前暴露风险，关键模块设置备份负责人

迭代计划

阶段	主要任务	阶段成果
第一阶段	需求分析、原型、架构、数据库和接口设计	可评审的选题材料与设计文档
第二阶段	文档上传、解析、分块、向量化和检索	可检索的知识库
第三阶段	问答、引用、多轮对话和反馈	可演示的问答主链路

阶段	主要任务	阶段成果
第四阶段	测试、部署、性能优化和演示准备	可访问的课程项目成品

07 团队愿景与结语

团队愿景

我们希望 DevPulse 智能问答助手从课程项目出发，逐步成为可复用的知识问答基础设施：把分散在文档、网页和聊天记录中的知识整理成可信、可追溯的答案，让使用者少找资料、少重复提问，把时间留给判断和创造。项目初期先完成知识导入、检索增强问答、答案引用和反馈闭环，保证回答有依据、错误可定位；后续扩展多轮对话、权限管理和知识缺口分析，服务班级、社团和小型团队。我们也希望每位成员在真实协作中提升需求分析、工程实现、测试和表达能力。

我们想证明什么

- AI 不只是生成答案的工具，更应该帮助团队沉淀知识、建立信任并持续改进。
- 一个课程项目也可以从真实问题出发，形成清晰的需求、可验证的设计和可演示的成果。
- 团队协作的价值，不只在把任务做完，更在于把方法和经验留下来。

DevPulse 智能问答助手 • 先让答案有出处，再让回答更聪明。