**功能**

**个性化模拟面试系统**

* 基于RAG查询面经嵌入知识库，大模型生成面试问题

- 经典面试题4000+，QA对应且通过大量用户使用（小林coding，cs-notes，csdn等），认证干净可靠。

* 结合简历中的技能/项目经历生成针对性问题

- 使用embedding ai解析简历文档，生成文档向量存入向量数据库，调用deepseek api前先使用embedding ai根据意图检索向量数据库中的内容，结果发回deepseek提问。

* 支持以对话形式自动化生成与修改prompt，并支持上传面经知识，并可调整面试官的各项封装好的设置（如关注简历的程度，发散程度，是否压力面等），生成自定义的大模型面试官

- 关注简历的程度: 0-100%，步长为5%

- 发散程度: 0-100%，步长为5%

- 压力面试程度: 0-100%，步长为5%

- 每个设置项至少提供5个不同的调整级别。

* 使用ai agent技术，使模拟面试官能够调用项目内其他功能，提供更加便捷，逼真的面试体验（如在进行代码能力测试时调用内置的代码编辑评测系统，调用api实时对面试者不同方面的评分进行修改等）
* 支持语音输入，锻炼面试能力

使用paraformer进行语音识别。

**AI代码评测系统（在线IDE+OJ+AI）**

* 支持在线代码编辑（在线IDE）
* 根据已有的OJ题库对DeepSeek R1进行LoRA微调，优化出题与评测表现
* 爬取leetcode免费题目3237道的json源文件，包含题库+难度+标签+解答，作为知识库来使用LoRA微调优化DeepSeek R1的出题与评测表现
* 训练Loss收敛阈值且模型在训练集上的损失值需稳定收：Loss ≤ 0.4（初始Loss约1.2），每100步记录Loss曲线，最终Loss需在最后10%训练步中波动范围≤5%
* 题目生成多样性，微调后生成的题目需覆盖LeetCode题库的标签分布（如算法、数据结构等）：标签覆盖率 ≥ 95%，抽样生成100道题目，统计标签分布与原题库的余弦相似度≥0.9
* 利用ai agent技术，根据面试系统的ai意向进行出题（如面试到代码能力测试部分时，面试系统ai调用该系统，并说明出涉及哪一部分知识点，什么难度的题目，由该系统ai进行出题）
* 点击提交后，根据评测机核对答案是否正确，并通过ai对代码思路，时间复杂度，代码规范性等进行评价。

**社区分享系统**

* 账户注册与登录
* 保存自定义的模拟面试官，面试记录（可从此处直接跳转到与选定的模拟面试官，以及面试记录的特定位置继续面试过程）

- 包含初始模拟面试官提示词及外部知识库及面试记录100+

* 关注，查看其他人的面试经验贴

- 包含初始面试经验帖100+

* 帖子里可以分享分享自定义的模拟面试官，以及面试记录，可以保存到自己的库中，并对话

**主要分工如下**

**1. 个性化模拟面试系统**

| **子任务** | **技术栈/工具** |  | **负责角色** | **关键要求** |
| --- | --- | --- | --- | --- |
| **1.1 RAG知识库集成** | Python/FAISS、LangChain |  | 算法工程师 | 嵌入4000+认证面试题，支持QA检索，确保数据干净可靠 |
| **1.2 简历解析与匹配** | SpaCy/DeepSeek API |  | 算法工程师 | 解析简历生成向量数据库，支持意图检索与针对性提问 |
| **1.3 自定义面试官配置开发** | React/Ant Design |  | 前端工程师 | 实现三项设置（关注简历、发散、压力面），每项提供5级调整 |
| **1.4 AI Agent交互逻辑** | Python/FastAPI |  | 后端工程师 | 支持对话式Prompt生成，调用代码评测系统API实时评分 |
| **1.5 语音输入模块** | Paraformer/Web Speech API |  | 全栈工程师 | 集成语音识别，延迟≤1秒，准确率≥90% |

**2. AI代码评测系统**

| **子任务** | **技术栈/工具** | **负责角色** | **关键要求** |
| --- | --- | --- | --- |
| **2.1 OJ题库整合** | MySQL/Redis | 后端工程师 | 爬取并清洗3237道LeetCode题目，剔除重复率≤3% |
| **2.2 DeepSeek R1微调** | PyTorch/LoRA | 算法工程师 | Loss≤0.4，标签覆盖率≥95%，题目重复率≤3% |
| **2.3 实时评测机开发** | Docker/Judge0 | 后端工程师 | 支持代码编译与结果比对，响应延迟P99≤2秒 |
| **2.4 AI代码评价模块** | CodeBERT/规则引擎 | 算法工程师 | 代码规范性、时间复杂度评价与人工评分一致性≥85% |
| **2.5 在线IDE开发** | React/Monaco | 前端工程师 | 支持代码高亮、自动补全，集成提交按钮与评测反馈 |

**3. 社区分享系统**

| **子任务** | **技术栈/工具** | **负责角色** | **关键要求** |
| --- | --- | --- | --- |
| **3.1 用户账户系统** | Spring Security/JWT | 后端工程师 | 支持注册/登录、权限管理，初始用户库≥1000 |
| **3.2 内容管理模块** | MongoDB/Elasticsearch | 后端工程师 | 支持保存100+面试官配置与1000+经验帖，支持收藏与跳转 |
| **3.3 社区交互功能** | WebSocket/Redis | 全栈工程师 | 实现关注、评论、复用配置功能，互动率≥20% |
| **3.4 数据埋点与统计** | ELK/Grafana | 数据分析师 | 用户留存率≥60%，行为数据可视化 |

**4.系统测试与文档撰写**

| **子任务** | **技术栈/工具** | **负责角色** | **关键要求** |
| --- | --- | --- | --- |
| **4.1 端到端测试** | Cypress/Selenium | 测试工程师 | 覆盖核心流程（面试→评测→分享） |
| **4.2 文档撰写** | Word |  | 覆盖所有功能模块、技术实现细节、量化指标，避免遗漏关键逻辑或依赖关系。 |