# 7技术可行性(技术风险评价)

因本项目为课程作业，面临着技术栈较基础、团队经验有限、项目周期较短的问题，结合实际，提出以下关键风险点及应对评估。

1. 技术选型风险
   1. 问题：技术栈选择不当可能导致开发受阻
   2. 具体风险：
      1. 前端框架：若团队只学过基础HTML/CSS/JS，直接使用Vue.js/React可能学习成本过高，会拖慢项目进程
      2. 后端复杂度：后端搭建可以选择Spring Boot或者Express，但是团队中大部分同学没有系统学习过Java，强行使用Spring Boot可能会加重团队负担，阻塞项目进程
      3. 数据库操作：在本次项目进展中，是否要引入ORM工具，如果手动写SQL一方面效率较低另一方面容易出错
   3. 应对建议：
      1. 简化技术栈：用更为熟悉的语言（Node.js+Express）快速搭建后端，避免追求“主流框架”
      2. 优先使用学过的技术：我们在数据库课程中已经学过了MySQL语言，在数据库操作中将直接使用MySQL来降低学习成本
2. 功能实现风险
   1. 问题：需求范围过大导致无法按期完成
   2. 高风险功能
      1. 人才匹配算法：尝试实现“智能推荐”可能需要复杂算法（如协同过滤等），难度较大
      2. 即时通讯（留言管理）：WebSocket实现难度较高，且需处理消息持久化
      3. 文件上传（简历/企业资料）：若为接触过云存储，本地存储可能引发路径问题或者服务器过载
   3. 应对建议：
      1. 删掉非核心功能：将“智能推荐”简化为标签筛选，用“站内信”代替即时通讯
      2. 使用现有的解决方案：小型文件上传改用Base64编码直接存到数据库；留言管理用普通表单提交，页面刷新实现
3. 团队协作风险
   1. 问题：多人开发时代码合并冲突、接口定义混乱。如前端同学等待后端API，双方对参数格式理解不一致，又或者数据库字段修改后未同步给全组，导致查询报错
   2. 应对建议：
      1. 定义接口文档：用Markdown或Swagger明确API路径、参数、返回值（哪怕手写截图发群里）
      2. 分模块开发：按角色分配独立功能（例如：A负责个人模块，B负责企业模块），减少交叉修改
      3. 每日同步进度：用腾讯会议10分钟快速对齐今日完成/卡点（避免憋大招最后集成失败）
4. 部署与测试风险
   1. 问题：本地运行正常，但部署后出现兼容性问题
   2. 常见坑点：
      1. 前端路由在Nginx配置错误导致404。
      2. 数据库连接未设置为远程访问权限。
      3. 未处理跨域问题（CORS），前端请求被拦截。
   3. 应对建议
      1. 尽早部署调试：不要等到最后一周才部署，用Heroku/Vercel等免费平台快速上线测试
      2. 记录部署步骤：写清依赖安装命令（如npm install）、端口号配置等，避免队友环境不一致
      3. 基础测试方案：至少覆盖核心功能（用户注册、简历提交、岗位发布），用Postman测试所有API，导出测试结果作为报告附件
5. 时间管理风险
   1. 问题：低估开发周期，最后熬夜赶工
   2. 应对建议：
      1. 按优先级进行开发：
         1. 核心功能：登录注册、简历管理、岗位发布（占70%时间）
         2. 加分功能：页面美化、数据图表（占20%时间）。
         3. 可以放弃功能：第三方登录及复杂权限控制。
      2. 设定阶段性目标并及时进行复盘

# 8法律可行性

1. 数据隐私保护简化措施
   1. 模拟数据原则：所有测试账号的简历信息（如姓名、电话、学历）均采用虚构数据（如张三+虚拟手机号）及小组内成员信息，从源头规避《个人信息保护法》合规风险
   2. 本地化部署：系统部署于校园网，不接入公网，数据仅限教师评审组访问，降低泄露风险
   3. 基础权限控制：实现用户角色隔离（个人/企业/管理员），企业账号仅可查看简历中的非敏感字段（如专业、技能），隐藏联系方式（需通过平台站内信沟通）
2. 内容审核责任弱化说明
   1. 标注免责说明：在系统登录页显著位置注明“本系统为教学演示项目，所有招聘信息及简历内容均为模拟数据，不构成真实求职行为”
   2. 教师参与审核：将“客服管理人员模块”的会员审核权限交由指导教师账号执行，人工审核企业提交的模拟招聘信息（如岗位描述、薪资范围），避免出现违法或不道德内容（如传销、歧视性条款）
   3. 功能限制：关闭企业用户“人才查询”模块的简历下载功能，仅允许在线预览，防止数据滥用风险。
3. 知识产权合规
   1. 使用开源组件：技术栈优先选择MIT/ Apache协议的开源框架（如Spring Boot、Vue.js），避免商业软件版权纠纷
   2. 简历模板说明：若需内置简历模板，直接引用高校就业指导中心公开模板，避免原创性争议。
4. 其余潜在风险
   1. 代码引用规范：在文档中明确标注参考的第三方代码片段来源（如CSDN技术博客、GitHub开源项目），遵守学校学术诚信规定，避免代码抄袭争议。
   2. 功能简化合理性：说明“因课程周期限制，本系统暂不实现在线支付、人脸识别等高风险功能”，聚焦核心业务流程合规性。