## “IT公司人才招聘系统”可行性分析前提

### 3.1 项目要求

#### 3.1.1 功能要求

##### 基础功能

职位信息发布与管理（增删改查）。

简历上传与分类存储（支持PDF/Word格式）。

面试流程管理（安排面试时间、结果记录）。

用户权限管理（企业HR、候选人两种角色）。

##### 扩展功能（选做）

集成AI算法（如简历关键词匹配、候选人智能推荐）优化招聘效率。

支持数据分析与可视化（如招聘周期统计、人才库分析）。

确保数据安全性（加密传输、权限分级管理、备份与恢复机制）。

#### 3.1.2 技术要求

开发语言：c/c++/Java/Python。

数据库：MySQL。

前端：HTML/CSS/JS + Bootstrap。

部署：本地运行或简易服务器。

系统响应时间低于2秒，支持高并发访问（至少1000用户同时在线）。

可扩展性：支持模块化设计，便于后续功能迭代。

基本信息变更验证，数据库访问控制在1S以内。

### 3.2 项目目标

#### 3.2.1 核心目标

为求职者和管理员提供简洁的操作界面，使求职、审阅、录取过程更加方便、快捷、人性化。

优化求职者体验，提供透明化流程（如实时进度反馈）。

掌握可行性分析、需求分析、系统设计等全流程方法。

理解数据库设计与前后端交互逻辑。

#### 3.2.2 进阶目标

建立企业人才数据库，支持长期人才储备和战略规划。

集成AI算法（如简历关键词匹配、候选人智能推荐）优化招聘效率。

#### 3.2.3 课程学习目标

实践软件工程生命周期（需求→设计→编码→测试→文档）。

锻炼团队协作能力（如组员分工：前端、后端、测试）。

### 3.3 项目环境、条件、假定和限制

#### 3.3.1 开发环境

硬件：个人笔记本电脑、Windows系统。

软件：VSCode、MySQL Workbench、Git。

协作平台：使用Github+Git进行团队协作

#### 3.3.2 假设条件

无需真实企业数据，使用模拟数据测试（如虚构的IT公司岗位）。

系统用户量较小（假设最多10人同时使用）。

无需考虑商业级安全性（基础登录验证）。

#### 3.3.3 关键假定

组员具备基础的编程能力（完成过c++课设）。

开发周期为4-6周。

老师提供可行性分析框架指导。

#### 3.3.4. 主要限制

技术限制：不要求使用微服务、分布式架构等高级技术。

时间限制：需在课程截止日期前提交完整文档和代码。

功能限制：以核心功能为主，扩展功能根据时间选做。

### 3.4 可行性研究方法

#### 3.4.1 技术可行性

评估是否能用已学技术实现基础功能。

验证数据库表设计是否合理（如E-R图设计）。

#### 3.4.2 操作可行性

设计简易原型图，验证功能逻辑是否清晰。

组内成员模拟HR和候选人角色，测试系统操作流程。

#### 3.4.3 经济可行性

成本估算：开发成本、运维成本、第三方服务采购费用。

收益预测：通过缩短招聘周期、降低猎头费用计算ROI（投资回报率）。

#### 3.4.4 风险分析

技术风险：组员技术能力不足 → 提前分配任务，预留学习时间。

时间风险：开发进度延迟 → 使用甘特图规划每周任务。