## **一、项目概述**

### **1.1 项目目标**

构建一个功能完善的海外文物知识服务Web平台

实现文物数据的多维度浏览、查询和可视化展示

提供个性化的用户体验和交互功能

实现文物间关联关系的智能推荐

## **二、项目团队组成与分工**

### **2.1 团队成员及分工**

| **角色** | **职责** |
| --- | --- |
| 项目经理 | 负责项目整体规划、进度控制和团队协调 |
| 前端开发 | 负责Web界面设计、交互实现和用户体验优化 |
| 后端开发 | 负责服务器架构设计、API开发和数据处理 |
| 数据库工程师 | 负责数据库设计、优化和知识图谱数据管理 |
| 算法工程师 | 负责推荐算法、知识图谱可视化和数据分析 |
| 测试工程师 | 负责系统测试、质量保证和性能优化 |

### **2.2 团队协作方式**

每周一次团队会议，汇报进度并解决问题

使用Git进行版本控制，采用敏捷开发方法

## **三、项目开发计划**

### **3.1 详细开发计划**

#### **第1周：需求分析与设计**

确定功能需求和用户交互流程

完成UI原型设计

制定技术选型方案

#### **第2-6周：核心功能开发**

实现用户认证和个人信息管理模块

开发文物数据浏览功能(筛选、排序、分页)

实现文物详情页及图片查看功能

开发简单查询和高级查询

实现相关文物推荐算法

开发用户收藏、评论等扩展功能

进行系统演示和培训

第七周

进行功能测试、性能测试和安全测试

优化系统响应速度和用户体验

完善错误处理和日志记录

进行压力测试和负载测试

## **四、技术方案**

### **4.1 技术栈选择**

**JAVA**

**shiro权限框架**

**lucene检索**

### **4.2 系统架构设计**

采用前后端分离的微服务架构：

前端：React单页应用，负责UI展示和用户交互

API网关：统一处理请求路由和认证

业务微服务：

用户服务：处理用户认证和个人信息管理

文物服务：处理文物数据浏览和查询

推荐服务：实现相关文物推荐

可视化服务：生成知识图谱和时间轴可视化数据

数据存储层：

Neo4j存储知识图谱关系

MySQL存储结构化数据

## **五、风险管理**

### **5.1 潜在风险及应对措施**

| **风险** | **应对措施** |
| --- | --- |
| 数据获取不足 | 建立多渠道数据采集方案，优先使用开放数据源 |
| 知识图谱构建延迟 | 提前进行技术验证，简化初期图谱复杂度 |
| 可视化性能问题 | 采用渐进式加载，优化大数据量渲染 |
| 推荐算法效果不佳 | 采用多种推荐策略组合，允许人工调整权重 |
| 团队协作不畅 | 明确分工，加强沟通，使用协作工具 |

## **六、项目成果与交付**

### **6.1 交付物清单**

完整的Web应用系统

源代码及部署文档

数据库设计文档

API接口文档

用户手册

测试报告

项目总结报告

### **6.2 项目成果指标**

支持至少10,000件文物数据的浏览和查询

响应时间：普通页面加载<1秒，复杂查询<3秒

支持并发用户数≥500

推荐准确率≥70%

用户满意度≥85%