# 目录

• 大学生竞赛信息聚合与订阅平台 - 项目报告

• 元数据

• 项目概述

• 项目背景

• 技术栈

• 系统架构

• 主要组件

• 数据流

• 功能特性

• 实现细节

• 关键算法

• 挑战与解决方案

• 测试与评估

• 测试结果

• 性能评估

• 结论与展望

• 未来工作

---

# 大学生竞赛信息聚合与订阅平台 - 项目报告

本报告由系统自动生成，包含项目的基本信息和详细说明。

## 元数据

\*\*作者\*\*: 测试作者

\*\*日期\*\*: 2025-07-06

\*\*项目ID\*\*: test-001

\*\*生成时间\*\*: 2025-07-06 17:13:37

# 项目概述

本项目是一个Web应用，主要目标是为大学生提供竞赛信息聚合与订阅服务。

项目开始于2023-01-01，计划完成于2023-12-31。

## 项目背景

随着高校竞赛数量的增加，学生需要一个集中的平台来获取和管理竞赛信息。

# 技术栈

本项目使用了以下技术栈：

前端：React, TypeScript, Ant Design

后端：Python, FastAPI, SQLAlchemy

数据库：PostgreSQL

部署：Docker, Nginx

其他工具：Git, GitHub Actions

# 系统架构

本系统采用前后端分离架构，后端使用FastAPI提供RESTful API，前端使用React构建用户界面。

### 主要组件

前端：用户界面、状态管理、API调用

后端：API服务、业务逻辑、数据访问

数据库：数据存储

缓存：提高性能

### 数据流

用户通过前端界面发起请求

请求经过API网关到达后端服务

后端服务处理请求并访问数据库

数据返回给前端展示

# 功能特性

竞赛信息浏览与搜索

个性化竞赛推荐

竞赛订阅与提醒

团队协作功能

竞赛资源共享

# 实现细节

系统实现采用模块化设计，主要包括用户管理、竞赛管理、订阅管理、推荐系统等模块。

### 关键算法

基于协同过滤的推荐算法

全文搜索算法

自动标签提取算法

### 挑战与解决方案

挑战1：数据量大导致查询性能下降

解决方案：引入缓存机制和索引优化

挑战2：个性化推荐准确性不足

解决方案：结合内容和协同过滤的混合推荐算法

# 测试与评估

采用单元测试、集成测试和端到端测试相结合的测试策略。

### 测试结果

单元测试覆盖率：85%

API测试通过率：95%

端到端测试通过率：90%

### 性能评估

页面加载时间：<2秒

API响应时间：<200ms

并发用户支持：1000+

# 结论与展望

本项目成功构建了一个功能完善、性能良好的大学生竞赛信息聚合与订阅平台，满足了用户的核心需求。

### 未来工作

引入AI助手功能

添加移动应用支持

扩展国际竞赛资源

优化推荐算法