

***iKNOW***

iKNOW大学生竞赛平台

交付文档

夏侯惇 2015013212

马福崑 2015013186

陈　超 2015103222

胡　哲 2015103187

一．产品名称

　　iKNOW大学生竞赛平台

二．网站链接&源代码链接地址

　　网站链接：[http://118.89.240.107](http://118.89.240.107/)

　　源代码链接网址：<https://gitlab.thsecourse.com/iKNOW-CPCS/iKNOW-Project>

　　后台管理员账号：iKNOW, 密码：iKNOWCMH

三．产品目标

　　1．系统功能

　　iKNOW大学生竞赛平台，致力于实现当前市场现有竞赛平台网站所具有的基本功能,并

在此基础上充分考虑用户需求,开发出新的功能,以提高竞争力。目前，已经实现了所有项目之初所定下的功能：

基本功能：

* 完备的注册，登录，包括参赛者和组织者两种类型
* 组织者组织信息设置，参赛者个人信息设置
* 组织者可以创建一个比赛，设置比赛阶段，设置个人赛和团体赛
* 组织者可以设置阶段报名表，上传附件等
* 参赛者可以查看比赛信息，报名比赛
* 参赛者创建队伍或加入队伍
* 参赛者可以发起事件（报名，退赛，申请队长，申请移交队长，申请编辑资料，申诉）
* 组织者可以管理参赛者，对违规参赛者予以处罚
* 组织者可以处理事件（队长报名，队员报名，退赛，移交队长，申请队长，申请编辑资料，申诉）
* 组织者可以上传比赛试题，设置参赛者提交方式
* 参赛者可以下载比赛试题，提交结果
* 组织者可以在线批改试题
* 参赛者可以查看批改结果

亮点功能：

* 完备的邮箱验证和找回密码功能，保证用户账号安全性
* 官方教程和学友分享，比赛兼学习平台
* 私信，用户之间可以方便交流
* 动态报名表，组织者可以根据自己的实际需求设置报名表
* 功能强大的选人组件，方便参赛者导入和指定对象发送通知
* 个性化主页，方便访问他人空间和私信对方

　　2. 性能方面

　　咋办啊？

四．开发组织管理

　　1. 开发过程

　　第一次迭代(10.27-11.9)：准备开发环境,完善所需配置;学习Vue.js.实现基本的注册、登录功能,包括前端对应页面及后端对应API,对应数据模型.

　　第二次迭代(11.10-11.23):　完善注册、登录功能。实现首页、个人/组织主页、官方教程、学友分享等页面及其所需要的API、数据模型;实现教程、私信系统的部分功能。

　　第三次迭代(11.24-12.7):　实现基本的比赛管理功能，包括：创建比赛功能，创建相应阶段，设置报名表，附件上传．实现通知，公告，比赛详情页等页面及后端对应API，对应数据模型

　　第四次迭代(12.8-12.21):　实现阶段管理的参赛者管理，事件处理，过程管理，成绩管理，参赛者报名(分为队伍和个人)，附件下载和上传等功能或页面及其所需要的API，数据模型等．实现访问他人主页．搜索功能．

　　第五次迭代(12.22-1.5):　实现比赛详情页面的排行榜和讨论区，完善官方教程和学友分享，修补尚未实现前后端交互的部分，包括参赛者管理，公告通知，搜索等，完善后端对应API，数据模型．整理代码风格，进行功能测试，部署．

2. 成员分工

大体上，前端由侯建国和陈超负责，后端由马福崑和胡哲负责，前后端交互由双方共同完成，每个人在小组内贡献度均等．

3. 开发环境

Ubuntu16.04 LTS

4. 配置管理

前端部分：　node.js　9.0.0

　　　　　　npm 5.5.1

　　　　　 vue.js 2.9.1

后端部分：

　　　　　　python　3.5.2

　　　　　 django 1.11.7

　　　　　 mysql 5.7.20

　　　　 django-rest-framework 3.7.3

五. 系统设计

　　1. 前端交互

　　前端模块的相关内容列表如下：

　　views: 每一个vue文件是一个页面，包括templete, script, style三分，完成了平台主要页面的设计.

　　components: 页面步局中的一些组件设计,包括导航栏，个人信息设置，比赛报名表等．

　　router: 提供了页面跳转的路由设置．

　　2. 后端模块

　　后端模块的相关内容列表如下：

　　backend: 对各个子模块进行统筹管理，包含数据模型的定义

　　userpage: 提供与用户相关的API,其中实现了需要的序列化器、权限控制及URL配置

　　tutorial:提供与教程相关的API,实现了需要的序列化器、权限控制及URL配置

　　competition:提供与比赛相关的API,实现了需要的序列化器、权限控制及URL配置;

　　3. 接口规范

　　前后端之间的接口均为RESTful API

　　4. 数据库设计

五．重点和难点问题及其解决办法

１．页面的刷新问题

　在前端页面的跳转过程中，经常出现页面跳转后数据没有更新的问题．但是一刷新页面就会显示更新后的数据．由于这是一个普遍存在的问题，因此我们对其做了重点解决．该问题出现的原因是页面created之后，当我们再次回到该页面时，页面就不再重新创建，而是保持原状．因此，我们设置了watch，用于监视路由，一旦发现路由跳转到指定页面，则触发相关函数，从后端获取数据，从而解决了该问题．

２．报名表的动态设计

　　报名表写死肯定不太好，因此我们考虑使用动态报名表．组织者可以按照自己的需求添加相应的组件．考虑到这个原因，我们对每一个组件都设置三个属性：标签名，组件类型，值．然后将用户选择的组件加入一个数组，整个传给后端作为报名表数组．当后端传给前端时，前端根据报名表数组每一项的组件类型选择指定组件，并在组件前标注标签名．这样便实现了报名表的动态设计．

六．测试总结

1. 前端测试

　　前端测试我们将网站的整个流程中经过的每一个功能均进行了手动测试，并对结果进行了截图保存．对测试出现的问题及时记录．主要的测试项包括：

* 注册，登录
* 组织主页，个人主页
* 官方教程，学友分享(创建教程，查看教程，删除教程，修改教程，收藏教程等)
* 私信
* 搜索
* 创建比赛
* 创建比赛阶段
* 设置报名表
* 附件上传
* 试题上传
* 参赛者提交设置
* 他人主页

七．系统部署