智能化在线测试教育平台

详细设计说明书

版本：V1.0

2021年9月20日

目录

[一、总体介绍 3](#_Toc421695287)

[1.1 背景介绍 3](#_Toc1473445236)

[1.2项目简介 3](#_Toc1113561217)

[1.2 使用技术 3](#_Toc1599710176)

[二、 模块介绍 5](#_Toc72195748)

[1. 题目发布与测试案例管理编辑模块 （负责人：徐朔） 5](#_Toc1484943701)

[2. 在线答题与判题模块 （负责人：孙百乐） 7](#_Toc803144308)

[3. 教师与课程管理模块 （负责人：王文婕、廖集秀） 9](#_Toc292333188)

[4. 人脸识别与作弊检测模块 （负责人：孙百乐、徐朔） 13](#_Toc1657367100)

[5.成绩分析与推荐系统模块 （负责人：王文婕、廖集秀） 14](#_Toc876460593)

[三、 部署 15](#_Toc882427403)

# 一、总体介绍

## 背景介绍

在线评测系统（online judge）是指针对系统用户的可提供在线服务的一种系统，或指一个在线的评测系统。通过该系统对程序代码进行编译和执行。具体过程为使用者登陆浏览器，将自己的程序代码提交，系统后台对程序进行编译和执行，之后系统使用设置好的输入和输出来比对，査看程序的正确性，其技术理论依据是软件工程中的黑盒测试。在线评测系统可以帮助教师减少评测作业的工作量，节省时间，并且准确性高，可靠性强。还能使学生及时收到反馈，尤其适合学习编程的同学使用。

现有的编程在线测评系统存在一些不足之处。第一，缺少对作弊行为的检测机制。第二，给学生提供的反馈只有分数信息，过于单调。

所以，我们的项目是在传统在线测试平台中融入智能元素，克服以上缺点。

## 1.2项目简介

本作品在传统在线测试平台（onlinejudge）中融入智能元素，实现了题目发布管理、在线判题、成绩管理与分析、作弊检测、智能推荐等功能。能很好地支撑高校教学过程中的作业发布与质量检测环节。

本项目分为五个模块：一、题目发布与测试案例管理编辑模块。二、在线答题与判题模块。三、教师与课程管理模块。四、人脸识别与作弊检测模块。五、成绩分析与推荐系统模块。

本项目使用多种编程语言，涉及web开发技术（前端、后端）、人工智能技术（人脸识别、推荐系统）、docker容器化等多种技术。

## 使用技术

* django框架：Python语言开发的web框架，与flask相比，更适用于大型项目.
* Linux操作系统和云服务器：使用腾讯云服务器，系统选择centOS 7,命令行操作.
* 数据库Mysql：既使用了django的orm操作数据库，又使用sql语句操作数据库
* Vscode远程开发：通过vscode的ssh插件连接到远程服务器合作写代码
* Python装饰器、多线程：后端登录验证、前端模板渲染、判题机时间限制等代码都需要用装饰器实现.
* 云数据库：购买腾讯云服务器、阿里云云数据库，减轻服务器压力，运行稳定
* 前后端分离架构：部分页面，如风云榜，题目评测都使用了ajax实现前后端分离，数据传输格式为json
* Git,gitee：使用git作为版本迭代工具，gitee为版本发布平台
* 队列算法：巧妙利用mysql数据库的事务管理实现了判题的队列算法，实现了先到先判功能
* 前端框架：如bootstrap，jquery
* Nginx＋gunicorn部署：为了提高网站响应速度，实现负载均衡，我们使用主流的Nginx＋gunicorn在云端部署
* Docker容器技术：我们把判题机放进了docker里，在任何一台能运行docker的电脑上都可以部署判题机，判题机越多，判题速度越快
* 人工智能平台：利用百度AI平台提供外部人脸检测模型接口

# 模块介绍

## 题目发布与测试案例管理编辑模块 （负责人：徐朔）

API介绍：

createtype(题目创建页面)：

|  |  |  |
| --- | --- | --- |
| url | create/common/ | create/select |
| 功能 | 普通评测题录入页面，将录入内容储存至数据库 | 选择题录入页面，将录入内容储存至数据库 |
| 说明 | 录入后点击提交会将向该url发送含题目内容的post请求，并于后端储存至数据库中 | 录入后点击提交会将向该url发送含题目内容的post请求，并于后端储存至数据库中 |
| 备注 | 需要教师权限 | 需要教师权限 |

testcases(题目内容编辑)：

|  |  |  |
| --- | --- | --- |
| url | create/testcases/ | create/selectOptions/ |
| 功能 | 编辑普通评测题测试案例数量以及输入输出的内容 | 编辑选择题选项以及正确选项 |
| 传递内容 | 生成的测试案例输入列表以及测试案例输出列表 | 生成选项列表以及答案列表 |
| 备注 | 弹出小窗，输入内容后生成列表通过前端返回内容至父页面，并在父页面显示 | 弹出小窗，输入内容后生成列表通过前端返回内容至父页面，并在父页面显示 |

edittype(编辑已录入的题目)：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| url | create/editCommon | create/editSelect | create/editTestcaseList | create/deleteProblem |
| 功能 | 编辑已录入普通评测题页面 | 编辑已录入的选择题页面 | 显示已录入的题目列表，可选择修改或删除 | 删除题目 |
| 说明 | 可修改已经录入的普通评测题目 | 可修改已经录入的选择题 |  |  |
| 备注 | 需要教师权限 | 需要教师权限 | 需要教师权限 | 需要教师权限 |

测试案例在数据库中的存储形式：

例如：有四组测试，每组的输入变量有两个，为inputx1和inputx2，

则存储形式为：

[[input11,input12],[input21,input22],[input31,input],[input,input32],[input41,input42]]

四组输出为：

[output,output,output,output,output]

数据库：

表名：question

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | pro\_id | pro\_type | problem |
| 说明 | 题目唯一id | 题目类型 | 相应题目表内的唯一id |
| 类型 | int | char | int |
| 备注 | 自增 | 对应每个题目类型的表 |  |

表名：question\_common

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | id | title | diffcult | point | content |  |
| 说明 | 题目id | 题目标题 | 难度 | 知识点 | 题目内容 |  |
| 类型 | int | char | char | char | char |  |
| 备注 | 递增 |  | 列表形式，储存难度与对应标签颜色 | 用|分隔开 |  |  |
| 字段 | pre\_code | testcase\_num | inputs | outputs | chapter\_id | course\_id |
| 说明 | 预置代码 | 测试案例数量 | 输入内容 | 输出内容 | 章节id | 课程id |
| 类型 | str | int | char | char | 外键 | 外键 |
| 备注 |  |  | 列表形式，储存测试案例输入信息 | 列表形式，储存测试案例输出信息 | 链接chapter表 | 链接course表 |

表名：question\_select

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | id | title | diffcult | point | content |
| 说明 | 题目id | 题目标题 | 难度 | 知识点 | 题目内容 |
| 类型 | int | char | char | char | char |
| 备注 | 递增 |  | 列表形式，储存难度与对应标签颜色 | 用|分隔开 |  |
| 字段 | testcase\_num | inputs | outputs | chapter\_id | course\_id |
| 说明 | 测试案例数量 | 输入内容 | 输出内容 | 章节id | 课程id |
| 类型 | int | char | char | 外键 | 外键 |
| 备注 |  | 列表形式，储存选择题选项 | 列表形式，储存选择题答案 | 链接chapter表 | 链接course表 |

## 在线答题与判题模块 （负责人：孙百乐）

api说明：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能 | 打开题目列表 | 打开一道普通编程题目的在线评测页面 | 打开一道选择题的在线评测页面 | 评测一道题目 | 人脸作弊检测 |
| 路由 | Onlinejudge/list\_problems/ | Onlinejudge/show\_common\_problem/ | Onlinejudge/show\_select\_problem/ | Onlinejudge/Judge/ | Onlinejudge/verify\_face/ |
| 参数及说明 | course\_id：  数据库中的课程编号；course\_id=all表示获取全部课程 | Id：  一道普通编程题目在数据库中的id，为数字类型; | Id：  一道选择题在数据库中的id，为数字类型; | sourceCode:用户输入的源代码，字符串类型；qsid：对应题目id；type：题目的类型； | Base64Img:拍摄照片的base64编码。 |
| 返回结果 | 返回所选课程下的题目列表页面。 | 返回这道普通编程题目的在线评测页面。 | 返回这道选择题的在线评测页面。 | 返回包含题目分数和报错日志的json信息，例如{"score":100,  errorMsg":"Answer right"} | 返回人脸作弊检测的结果，格式为json。 |
| 备注 | 需登录验证 | 需登录验证 | 需登录验证 | 主要由判题机调用，需登录验证 | 需登录验证 |

数据库：

**表名：Queue**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | id | pro\_id | type0 | ans\_num | inputs | outputs |
| 说明 | 序号 | 题目唯一id | 题目类型 | 测试案例数量 | 测试案例中的输入值 | 测试案例中的输出值 |
| 类型 | int | int | char | int | char | char |
| 备注 | 自增 | 与question\_common和question\_select中的id保持一致 | common或select | 此字段往往不直接使用 | 字符串列表，有特定格式 | 字符串列表，有特定格式 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | statue | time | sourceCode | score | msg | userid |
| 说明 | 状态 | 做题时间 | 用户源码 | 分数 | 错误提示 | 用户id |
| 类型 | int | date | char | int | char | int |
| 备注 | -1表示未拉取，0表示已拉取正在判题，1表示已判完 | 自动填入 | 从页面输入 | [0,100] | 字符串列表 | 用户的唯一标记 |

html页面模板：

|  |  |  |  |
| --- | --- | --- | --- |
| 文件名 | problemList.html | showCommonProblem.html | showSelectProblem.html |
| 说明 | 题目列表页面 | 普通编程题目页面 | 选择题页面 |

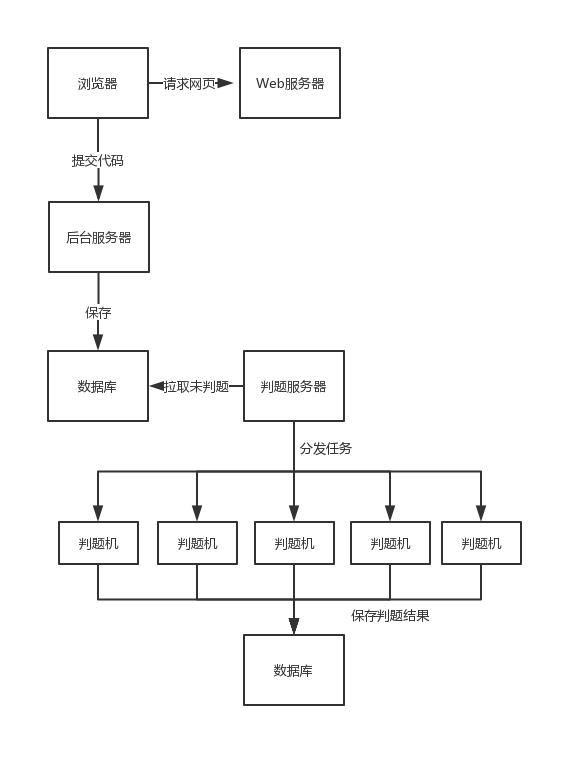
判题机：

利用了数据库的事务管理，实现队列判题和高并发。用户相当于生产者，不断地向数据库提交待测评列表（即上述定义的Queue表），测评服务器相当于一个消费者，不断的从数据库中获取未判提交列表，判题机拉取到最早提交的未判题以及我们制定好格式的输入输出内容进行判题，判题完成后将判题结果记录到数据库表。

判题机被封装到docker容器中，运行多个容器能实现多个判题机同时运行，且这些判题机可以在任意的机器上运行

为防止多线程资源竞争造成冲突，加入互斥锁（mutex）。

判题机流程图：



## 教师与课程管理模块 （负责人：王文婕、廖集秀）

数据库：

**表名：organization**

|  |  |  |
| --- | --- | --- |
| 字段 | name | email |
| 说明 | 组织名称 | 组织邮箱 |
| 类型 | text | email |
| 备注 |  |  |

**表名：course**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | cou\_id | course\_name | tech\_id | course\_info | course\_resource |
| 说明 | 课程唯一 | 课程名称 | 教师id | 课程简介 | 课程资源 |
| 类型 | char | char | int | text | text |
| 备注 |  |  | 外键，链接teacher表 |  |  |

**表名：teacher**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | tech\_id | teacher\_name | teacher\_pwd | teacher\_email | teacher\_info | org |
| 说明 | 教师唯一id | 教师名称 | 教师登录密码 | 教师绑定邮箱 | 教师简介 | 所属组织(例如中国石油大学（华东）） |
| 类型 | char | char | char | email | text | int |
| 备注 |  |  |  |  |  | 外键，链接organization表 |

**表名：student**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | stu\_id | student\_name | student\_pwd | student\_email | cls\_id | org |
| 说明 | 学生唯一id | 学生名称 | 学生登录密码 | 学生绑定邮箱 | 学生所在班级 | 所属组织(例如中国石油大学（华东）） |
| 类型 | char | char | char | email | char | int |
| 备注 |  |  |  |  |  | 外键，链接organization表 |

**表名：studentcourse**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | studentcourseid | cou\_id | stu\_id | group |
| 说明 | 自动生成的id | 课程id | 学生id | 小组号 |
| 类型 | int | char | char | char |
| 备注 |  | 外键，链接course表 | 外键，链接student表 | 由教师进行分组 |

**表名：chapter**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | chapterid | cha\_id | chapter\_name | chapter\_info | cour\_id |
| 说明 | 章节唯一id，自动生成 | 章节id，第一章第二章等 | 章节名称 | 章节简介 | 所属课程id |
| 类型 | int | char | char | text | char |
| 备注 |  |  |  |  | 外键，链接course表 |

**表名：scoreinfo**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | scoreid | stu\_id | problem\_score | proid | time | state |
| 说明 | 唯一id，自动生成 | 学生id | 该题的分数 | 题目号 | 提交此题的时间 | 做题的状态 |
| 类型 | int | char | char | int | datetime | char |
| 备注 |  | 外键，链接studentcourse表 |  | 对应question表中的序号 |  | 显示判断结果，是否有可能作弊 |

html学生个人中心页面模板：

登录时长计时（左上角）：显示此次的登录时长。

在用户登录时记录下用户登录的时间，在用户的操作页面用JavaScript自制即使器，以现在的时间为基准减去登录时间，每秒进行更新，展示在个人中心主页上。

自动登出：用户在登录后每隔一个小时会进行登录检测，即跳出弹框提醒，十秒内未点击，则判定用户已不在使用，执行自动登出。若用户关闭浏览器，则会在关闭浏览器的同时进行登出。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 文件名 | a.html | lis.html | homme.html | choose\_0.html | showchapter.html | showscore.html |
| 说明 | 学生个人主页，导航到课程信息和智能推荐界面 | 智能推荐题目，根据该同学的专业和错题进行针对性推荐 | 所选课程的显示界面（已选课程不显示） | 选课界面，显示可以选择的所有课程 | 章节界面，显示该课程的所有章节，点击进入该章节题目的模块 | 详细成绩显示界面，包括了此题的题目内容，历史最高分和当前得分 |

html教师中心页面模板:

管理员模块：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 功能 | 教师信息修改、删除 | 教师信息展示 | 各科课程录入 | 批量导入教师信息 | 各科题库录入 | 学生信息修改、删除 |
| 介绍 | 为教师进行调节教学行政班，或者为教师删除该门课程 | 展示所有教师的任课情况 | 录入新的课程 | 根据表格批量导入教师所授课程及其行政班 | 录入题目 | 为学生退课及调课 |

普通教师模块

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 功能 | 学生信息录入 | 学生信息展示 | 批量导入 | 成绩录入 | 成绩查看 | 学生分组 |
| 介绍 | 如果有学生没有经过教务系统（管理员允许）选择了这门课，可以通过与任课教师沟通协商获取教学资源 | 展示该教师所受课程及对应课程的所有学生 | 按照文件批量导入学生及对应分组 | 对应课程按照分组进行打分 | 按照所受课程进行成绩查看 | 为落单学生进行分组，或者为分组不满意同学进行调组 |

## 人脸识别与作弊检测模块 （负责人：孙百乐、徐朔）

人脸检测：

后端api：

url：onlinejudge/verify\_face/

类型：post

参数：data:{base64Img:\*摄像头所拍摄人脸的base64字符串码\*}

前端逻辑：

进入题目页面=>请求用户打开摄像头=>若用户不允许，不予显示题目页面=>若用户允许，显示题目信息，并每隔1分钟对用户抓拍一次，将抓拍照片的base64Img上传至服务器=>服务器判断后返回结果，示例结果：{

"face\_num": 1,

"face\_list": [

{

"face\_token": "9738e07b774399fa430a2fd9bd6cc437",

"location": {

"left": 179.21,

"top": 114.69,

"width": 218,

"height": 217,

"rotation": -3

},

"face\_probability": 1,

"angle": {

"yaw": -5.08,

"pitch": -15.1,

"roll": -5.24

}

}

]

}；

其中face\_num（人脸数量）和face\_token（人脸token）为判断作弊的重要依据。与上一张图片比对，若人脸消失或者两张人脸不一致，都可判为答题异常。

人脸检测模型提供自百度AI开放平台。

时长分析：

根据用户两次提交的时长间隔来判断是否

## 5.成绩分析与推荐系统模块 （负责人：王文婕、廖集秀）

数据来源

1. 用户集：登录平台并完善信息的所有用户，字段有学号、姓名、专业。其中，主要利用专业字段进行用户推荐相关性的分析。
2. 错题集：记录下用户错题的信息，包括题号、时间、所属知识点。
3. 题集、知识点集：管理员或教师录入的所有题目或知识点。

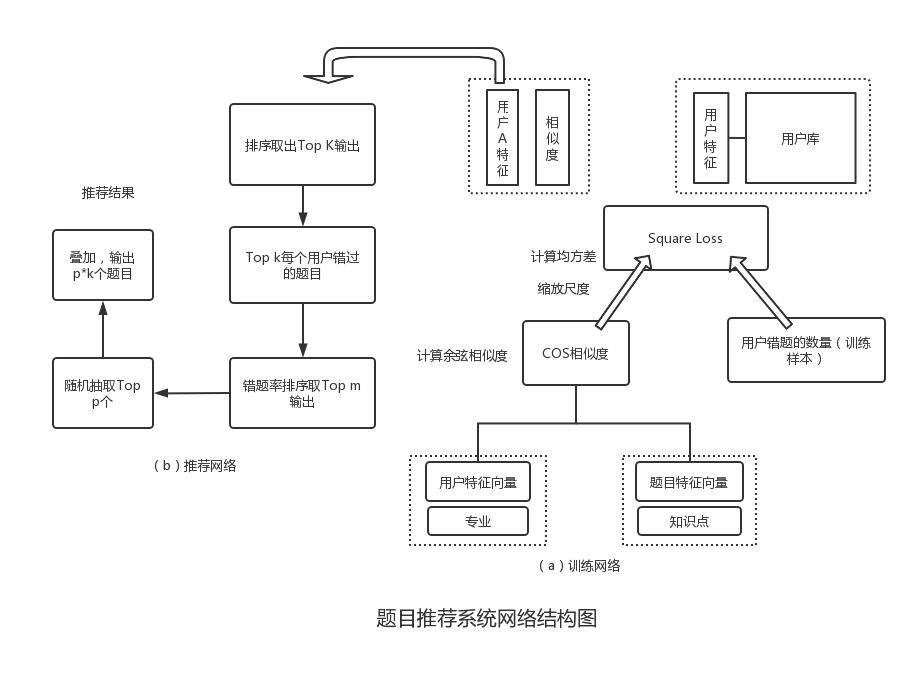
实现思路

1、给定用户ID，索引该用户的特征向量。计算该用户的特征向量与用户库中的其他用户特征向量的余弦相似度。

2、对相似度计算结果进行排序，输出排序前k个用户的原始信息和特征向量。 根据相似度排名前k个用户的ID调用评分数据，输出该k个用户做错过的所有题目。

3、从相似度最高的k个用户中随机抽取m个，取抽取的每个用户错题率最高的p个题目进行推荐。引入随机因素能够使每次推荐的结果不一样。

4、输出 p \* k个题目，作为推荐结果。



实现结果

在界面上为用户推荐适合个人特点的题目（按照专业及错题率进行比对），点击即可作答，实现高效率提升自身水平，做到哪里薄弱学哪里。

风云榜

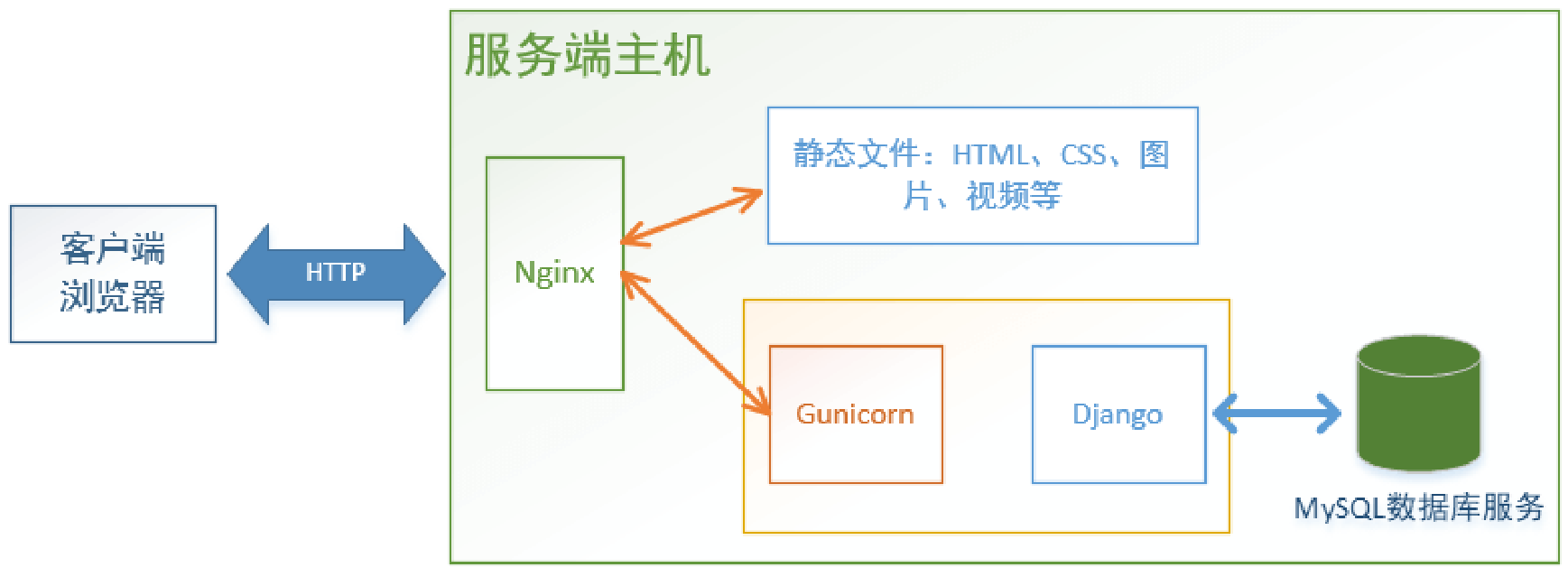
以AC为主要关键字，正确率为次要关键字，显示前十名同学的排名。

处理流程：从数据库中提取数据，使用Bar来绘制图像，最后运用ajax展示到前端。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 分数 | AC | total | 正确率 |
| 说明 | 每一道题的最高分之和 | 拿到满分的题目数量 | 做过的总题数（在拿到此题的满分后此题不再计数） | =AC/total |

# 部署

我们采用部署结构如图：



使用nginx做反向代理，gunicorn作为django的服务器，后台数据库使用MySQL

在我们提交的源码中，我们把gunicorn服务器和django放入容器中，mysql数据库使用阿里云数据库远程连接。

详情请见《智能化在线测试教育平台安装部署说明书》

为您提供测试账号

账号：123456

密码：123456

解决chrome浏览器打不开摄像头的问题：https://www.leyoubaloy.xyz/?p=961