目录

[摘要 2](#_Toc107495584)

[需求分析 2](#_Toc107495585)

[一、考生情况分析 2](#_Toc107495586)

[二、成绩录入 3](#_Toc107495587)

[三、录用 3](#_Toc107495588)

[四、输出需求 3](#_Toc107495589)

[五、数据流图和数据字典 4](#_Toc107495590)

[概要设计 5](#_Toc107495591)

[一、系统结构设计 5](#_Toc107495592)

[二、运行环境 6](#_Toc107495593)

[三、数据库结构设计 6](#_Toc107495594)

[四、界面设计 7](#_Toc107495595)

[详细设计 8](#_Toc107495596)

[一、数据库部分 8](#_Toc107495597)

[二、前端部分 11](#_Toc107495598)

[三、后端部分 20](#_Toc107495599)

[参考文献 22](#_Toc107495600)

[致谢 22](#_Toc107495601)

[总结 22](#_Toc107495602)

摘要

每年报名参加招聘考试的学生有几千名，我们需要编写一个公务员考试系统，这个系统需要有学生端和管理员端。学生端具有注册、查看自己的信息、下载准考证、查询录取结果的功能。而管理员端具有登陆、授权学生的注册信息、修改和删除学生信息、录入学生的成绩、查看所有考生的录取情况的功能。

我们可以根据这些要求，编写一个可以实现这些功能的系统。该系统由服务端和Web客户端两大部分组成,其中服务端采用了Flask框架进行开发,数据存储使用了关系型数据库My SQL。而Web客户端在应用分层架构层面采用了组件化开发模式。

需求分析

一、考生情况分析

每年报名参加招聘考试的学生有几千名，分为三个专业：法律、行政、财经。考生报名时登记如下内容：姓名、性别、生日、籍贯、报名时间、身份证号码、电话号码、报考专业、学历等。

考生报名后，招聘办要为考生安排考场，编排准考证号、打印准考证。考生分别来自全市各区，考生参加考试的考场一般就近安排。

考场管理：考场有编号、考场地址，假设每个考场可容纳的人数预先确定，一个考场安排考生人数满了，再安排下一考场。同一考场安排专业相同的考生。

准考证号采用11位数字编码，编码规则如下：前两位是年份，三位到五位是专业ID号码，六到七位是月份，以后是现在的时间戳。

二、成绩录入

考生的试卷在每个科目考试结束后按考场分别装订成册　，同一考场的试卷按准考谣等的先后顺序排列。因而，考试成绩的输入是按考场、分科目进行的，同一考场、同一门科目的成绩按准考证号的顺序依次进行输入。

三、录用

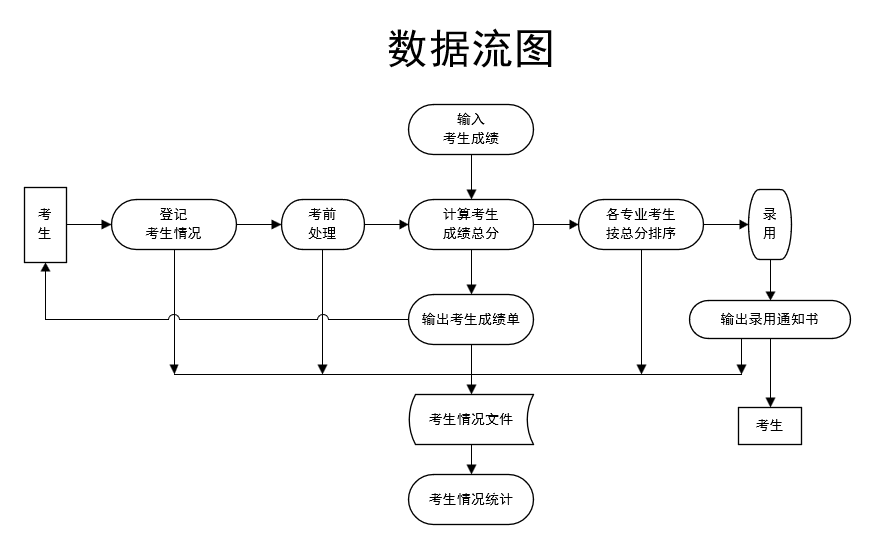
考生成绩输入后，由计算机计算每位考生三门考试科目的成绩总分。然后，三个专业分别将考生按总分从高到低进行排序。

排序后的考生名单供用人单位录用时做参考。被某单位录用的考生，应在供录用的名单中去除，同时添加到已录用名单中。

四、输出需求

1. 考生的准考证（含准考证号、姓名、性别、身份证号码、籍贯电话号码、专业等）。
2. 发给每位考生的考试成绩单。
3. 招聘考试办公室留存的按准考证号顺序排列的成绩表。
4. 三个专业分别按总分从高到低排序的考生成绩表。
5. 发给每位考生的录用通知书（含准考证号、姓名、性别、报考时间、专业、录用单位等）。
6. 录用名单（含准考证号、姓名、性别、专业、录用单位等）。

五、数据流图和数据字典



本系统数据字典如下：

1. 数据项定义

专业代号=[177=财经|178=行政|179=法律]

准考证编排规则：1-2位是年份，第3-5是所在专业代号，6-11是当时的时间戳。

考生=准考证号+姓名+性别+出生日期+地址+1{课程名+成绩}3+总分+名次+录用否+录用单位。

推考证=准考证号+姓名+专业+考场号+考场地址。

考生文件分为两种：一种按准考证号码的顺序排列，另一种按考生成绩总分由高到低排列。录用通知书-准考证号+专业+姓名+录用单位+地址。

考生成绩单=准考证号+姓名+专业+1{课程名+成绩}3+总分+地址。

1. 处理算法

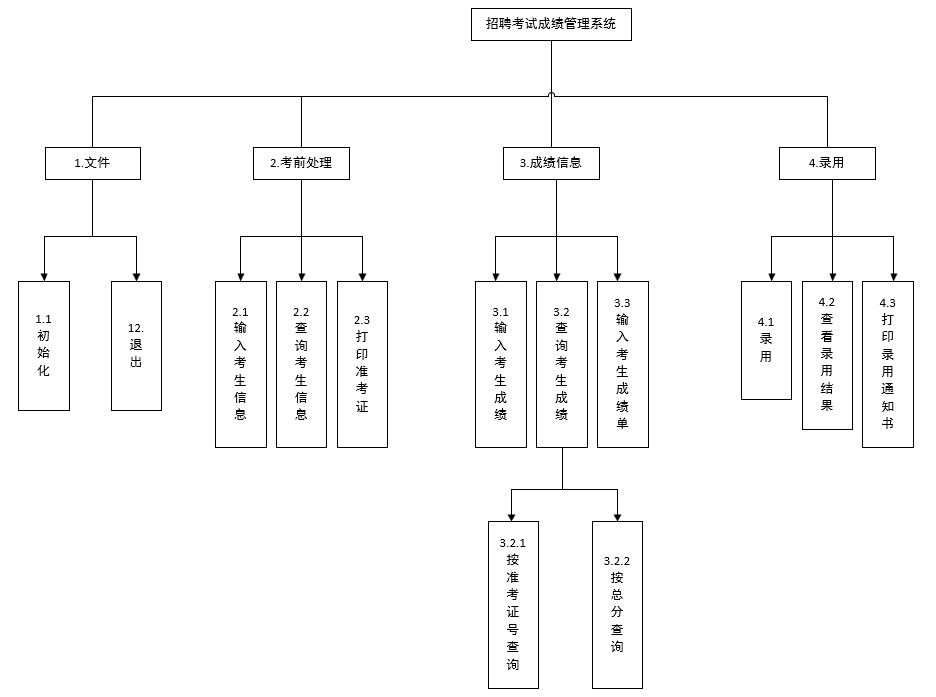
排序：输出三个专业的考生分别按总分由高到低的次序排序的名单，供录用参考。

录用原则：各专业按考生成绩总分从高分到低分的次序录用，总分相同时专业课成绩高的优先录用，如果被录取的人数已满，则不被录用。

概要设计

一、系统结构设计

为进行结构设计，首先把复杂的功能分解为比较简单的功能，此时数据流图也进一步细化。通常一个模块完成一个适当的功能。分析员应把模块组织成层次结构，顶层模块调用它的下一层模块，下一层模块再调用其下层模块，依次向下调用，最下层的模块能完成某个功能。软件的结构可用层次图或结构图来描绘。招聘考试成绩管理系统的HIPO图如下。



二、运行环境

|  |  |  |  |
| --- | --- | --- | --- |
| 分类 | 名称 | 版本 | 语种 |
| 操作系统 | Windows11 | 21H2 | 简体中文 |
| 数据平台 | MySQL | 8.0.16 | 英文 |
| 前端应用 | Html | 5 | 英文 |
|  | CSS | 3 | 英文 |
|  | Bootstrap | v4 | 简体中文 |
|  | JS | ES11 | 英文 |
|  | VUE | 2.0 | 英文 |
| 后端应用 | Python | 3.8.8 | 英文 |
|  | Flask | 2.1.2 | 英文 |
|  | Reprotlab | 3.6.10 | 英文 |
|  | Pymysql | 1.0.2 | 英文 |
| 浏览器 | Chrome | 102.0.5505.115 | 简体中文 |
| 编辑器 | VS Code | 1.67.1 | 简体中文 |
|  | Pycharm | 2022.1.3 | 英文 |

三、数据库结构设计

考虑系统数据安全性，进入本系统时要输入用户账号、密码，非法用户不能进入。因而要建立管理员数据表。

在考前处理时，要为考生安排考场，编排准考证号。编写程序让计算机自动生成准考证号。

考生数据表里存放考生的所有信息，根据需求分析阶段得到的数据字典，决定其所含的字段。

考生录用情况可建立两个不同的数据表来存放数据。如果考虑到数据量太大，也可用考生数据表来生成录用查询视图。

因为需要开启和关闭注册时间，所以有独立的数据表来存储可注册时间。本系统建立的数据库文件共含6个数据表，其中对于录用考生可以不建数据表。各数据表所含的字段如下：

1. 管理员：id、账号、密码。
2. 学生：学生准考证号、姓名、性别、生日、籍贯、报名时间、身份证号码、电话号码、专业、学历、成绩、总成绩、授权信息。
3. 学生登陆：准考证号、手机号、密码。
4. 公司标准：公司ID、专业、录取分数线、录入总人数。
5. 开放注册时间：ID、开始日期、结束日期、结束时间。
6. 录取考生：准考证号、姓名、性别、专业、总分、录用单位。
7. 未录取考生：准考证号、姓名、性别、专业、总分、分数线。
8. 考试地点：地点ID、省市、学校、教学楼、考场号、座位号。

四、界面设计

1. 管理员具有登陆系统，只有登陆才可以操作数据，以保证学生数据安全。
2. 管理员具有汇总查看学生信息，修改和删除某个学生的信息， 并对没有成绩的学生进行成绩录入。
3. 管理员具有统一查看学生的录取情况的界面。
4. 学生端分为注册和登陆，如果有账号，即可直接登陆查看信息，反之需要进入注册页面进行注册。注册完毕后，需要管理员进行审核才可以下载准考证和参加考试。
5. 如果学生忘记了自己的密码，即可输入身份证和手机号来找回自己的密码，查看自己的信息。
6. 已被批准的学生可以下载准考证，考完后可以查看自己的录取结果，并下载成绩单的功能。

详细设计

一、数据库部分

1. 管理员登陆

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 键名 | 类型 | 长度 | 约束 | 作用 |
| Id | int | 11 | 唯一、不为空、主键 | 管理员ID |
| account | varchar | 20 | NONE | 管理员账号 |
| password | varchar | 20 | NONE | 管理员密码 |
| permission | int | 11 | 默认0 | 管理员权限 |

1. 学生注册表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 键名 | 类型 | 长度 | 约束 | 作用 |
| studentId | bigint | 20 | 唯一、不为空、主键 | 准考证号 |
| name | varchar | 10 | 不为空 | 姓名 |
| sex | varchar | 4 | 仅限男女、默认男 | 性别 |
| number | varchar | 18 | 必须18位 | 身份证 |
| numberP | varchar | 11 | 必须11位、不为空 | 电话号 |
| birthday | varchar | 10 | xxxx-xx-xx | 生日 |
| province | varchar | 10 | NONE | 籍贯 |
| education | varchar | 6 | NONE | 学历 |
| profession | varchar | 10 | 不为空 | 专业 |
| date | varchar | 11 | NONE | 注册日期 |
| score | varchar | 64 | 列表 | 三项成绩 |
| sumScore | bigint | 20 |  | 总成绩 |
| allow | int | 11 | 默认0 | 授权 |

1. 学生登陆

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 键名 | 类型 | 长度 | 约束 | 作用 |
| studentId | bigint | 20 | 学生表外键、主键 | 准考证号 |
| numberP | varchar | 18 | 不为空 | 电话 |
| password | varchar | 30 | 不为空 | 密码 |

1. 公司信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 键名 | 类型 | 长度 | 约束 | 作用 |
| conditionId | int | 11 | 主键、不为空 | 公司ID |
| profession | varchar | 10 | 不为空 | 录取专业 |
| fractions | int | 11 | NONE | 录取分数线 |
| nums | int | 11 | NONE | 录取总人数 |

1. 已录取学生表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 键名 | 类型 | 长度 | 约束 | 作用 |
| studentId | bigint | 20 | 学生表外键、主键 | 准考证号 |
| name | varchar | 10 | 不为空 | 姓名 |
| profession | varchar | 10 | 不为空 | 专业 |
| sumScore | bigint | 20 | 不为空 | 总成绩 |
| conditionId | int | 11 | 不为空 | 录取公司id |

1. 未录取学生表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 键名 | 类型 | 长度 | 约束 | 作用 |
| studentId | bigint | 20 | 学生表外键、主键 | 准考证号 |
| name | varchar | 10 | 不为空 | 姓名 |
| profession | varchar | 10 | 不为空 | 专业 |
| sumScore | bigint | 20 | 不为空 | 总成绩 |
| fractions | bigint | 20 | 不为空 | 分数线 |

1. 考试地点

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 键名 | 类型 | 长度 | 约束 | 作用 |
| placeId | bigint | 20 | 主键、不为空 | 考点ID |
| province | varchar | 10 | 不为空 | 省市 |
| school | varchar | 10 | 不为空 | 学校 |
| classroom | varchar | 10 | 不为空 | 教室 |
| examRoom | varchar | 10 | 不为空 | 考场号 |
| seat | int | 11 | 不为空 | 座位号 |

1. 可注册时间

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 键名 | 类型 | 长度 | 约束 | 作用 |
| id | int | 11 | 主键 | 时间id |
| dateStart | date |  | 不为空 | 开始日期 |
| dateEnd | date |  | 不为空 | 结束日期 |
| time | time |  | 不为空 | 结束时间 |

二、前端部分

前端部分主要分为学生端和管理员端。界面只要以简洁为主，使用了bootstrap框架中的网格系统和CSS样式来布局。

学生端简介：

学生端页面有：主界面、考生注册界面、考生查询信息界面、查看考生录取情况界面、考生信息界面组成。

1．学生端主界面：分别有注册考生信息、查询考生信息、查询考生录取结果等功能。

2．考生注册界面：有姓名、性别、身份证号、专业等注册选项，当考生注册成功后将会跳转到考生信息界面,即可查看考生准考证号和考生详细信息。

3. 考生在忘记密码时可以通过找回密码按钮用身份证和电话号去申请之前的密码。

4．考生查询信息界面：使用姓名和身份证去查询，查询成功跳转到学生信息界面。即可查看考生准考证号和考生详细信息。

5．查看考生录取情况界面：考生在此界面输入准考证号来查询考生录取情况，在输入考生号准确无误后将会跳转到考生录取结果详细界面，来查看是否录取，并且可以下载录取结果pdf文档。

管理员端：

管理员端页面有： 管理员主界面，管理员登录界面、查看考生信息界面、考生成绩录入界面、查看考生录取情况界面组成。管理员非为超级管理员和普通管理员，超级管理员什么都可以操作，普通管理员不能录取考生成绩。

1. 管理员主界面：分别有查看考生信息、考生成绩录入、查看考生录取情况。

1. 管理员登陆界面：在进去管理员主界面时如果没有登陆，将会跳转到登录界面，登录完成后才会跳转到主界面。

3．查看考生信息：将考生注册过后的详细信息将会以面板的方式呈现，可以对考生授权，删除，修改，查询等操作。

4. 考生成绩录取界面：主要时录入考生成绩，在输入考生号后就可以输入对应的成绩然后录入。

1. 查看考生录取情况：当考生成绩出来后即可查看考生录取情况。

**学生端页面**

1. **主页**

**介绍：**学生页面首页主要以选择功能的界面展示，考生可以通过注册信息去注册考生通行证并登录报名，查询考试信息可以在考生忘记信息时查询个人信息，如：查询S准考证号、注册时间等。

**实现功能：**

通过点击按钮实现页面的跳转。

**2) 登录考生信息**

**介绍：**

注册考生信息页面分为考生的账号登录、找回密码、和注册组成、考生在没有注册报名考试项目时可以在注册界面注册，在注册完成后考生就可以登录查看自己个人信息，当考生忘记自己密码时可以选择找回密码去获取自己之前的密码。

**登录功能实现：**

**使用Vue获取账号文本框中的内容和密码文本框中的内容进行获取，在点击登录按钮时，首先判断文本框中的内容是否为空，如果为空则弹出‘请输入信息’，如果不为空，则将获取的内容文本发送请求到后端，然后后端接收到值并进行处理然后返回一些参数，如：返回值为200，则账户和密码正确，如果返回201说明密码错误，如果返回202说明没有账号。在密码正确时，会跳转到考生消息页面。**

**3) 注册考生信息**

**简介：**

**在首次登录时没有考试信息时可以通过注册来实现，注册中有姓名、性别、身份证、生日、省份、电话号、密码、确认密码、学历、专业等。在全部填完后则可登录。**

**注册功能实现：**

**使用Vue通过获取文本中的内容放入到Vue中的data中，当点击注册按钮后，如果填写密码和确认密码不一致，则显示提示，如果身份证号码无效，则显示提示，如果有一天信息为空，则显示提示‘信息不能为空’，如果信息正确，则向后端发送请求，将data数据发送过去，后端会返回一些数据，如果返回200说明注册成功，如果返回201说明身份证号已注册，如果返回202说明信息不能为空，如果显示203说明注册时间以过，如果显示204说明注册没有开始，如果显示205说明电话号已存在。当注册成功后会有一个2秒计时器，在过两秒后会跳转到学生详细界面。**

**准考证号自动生成功能实现：**

**准考证号一共有**11位，连两位为年如：22，第三位到第五位为专业号如：176为法律，177为行政，178财经， **六到十一为时间戳生成的随机数。就生成了一个随机的学号。如**22179877126。

1. **找回密码**

**介绍：**

**当我们忘记密码是，我们可以通过使用找回密码去找到我们之前使用的密码。**

**找回密码功能实现：**

**使用Vue将身份证号中的文本框内容和电话号码文本框内容获取并放入到Vue中的data中。然后在点击找回密码按钮后将身份证号和电话号码发送到后端，然后后端经过处理后返回数据，如果返回200，说明找回成功，在文本框下方会显示之前密码，如果返回201则说明没有该学生。**

1. **登录考生信息详细页面**

**介绍：**

考生在登录页面登录成功后会跳转到考生信息详细页面，在此页面考生能看到报考的专业、准考证号、姓名、性别、身份证号和信息审核，在刚报考完成的考生信息中的信息审核显示审核中，在审核通过后，信息审核内容显示已通过，并且显示考试地点和报考时间等信息，并可以下载准考证。

**考生信息功能：**

在考生登录成功后会将考生id传到考生详细页面，在详细页面中获取此考生详细信息，然后将考生详细信息通过Vue数据渲染显示在页面上。当考生审核通过后，程序通过js将考生考试地点、考试时间会显示在考生详细页面上，并且左边栏会有一个下载中考证的按钮，点击下载中考即可，

**准考证功能实现：**

在前端传递此学生id给后端，后端接受到id后返回一个pdf文件即可下载。

1. **查询考生信息**

**介绍：**

考生还可以在查询考生信息中登录查看考生信息，用姓名和身份证号的方式去查看，查询成功后也会进入考生详细信息查询界面。

**功能实现：**

先回去姓名和身份证号

1. **查询考生信息**

**介绍：**

当考试录取结果出来后，考生可以通过用准考证号去录取信息查询页面去查看考生录取结果，并可以打印录取信息。

**录取结果查询页面功能实现：**

获取准考证号中的文本，当点击查询按钮后将准考证号发送给后端，然后返回数值，如果返回200说明准考证号正确并有录取结果并且跳转到录取结果界面，如果返回201说明学号不存在并在页面显示提示框我‘学生没有成绩’，如果返回202说明学号不存在并提示‘该准考证号不存在’，如果返回203说明学号错误并提示‘请输入正确的准考证号’，如果没有输入则提示‘未输入学号’。

**录取结果页面功能实现：**

在输入有效的学号后将成功跳转到录取结果界面，并将准考证号通过跳转页面的方式传递到录取结果界面，然后只需要将准考证号传递到后端，后端会返回此考生的有关信息，然后通过Vue将数据渲染到页面即可显示内容。

**打印准考证号功能实现：**

当第一次进去此页面时会将学生准考证号传递到后端，然后后端结果处理后返回一个pdf文件，之后考生只需要点击下载录取结果pdf即可。

**管理员端详细介绍**

**1）主页**

**介绍：**

管理员端首页主要以选择功能的界面展示，如点击查看考生、考生成绩录取、查看考生录取信息等按钮。

**功能实现：**

在html页面中添加a标签实现页面跳转。

1. **查看考生信息页面**

**介绍：**

在这页面管理员可以查看考生的详细信息，如考好、姓名、性别、专业、身份证号、日期、电话、学历、授权等信息。也可以对学生批量删除授权和修改等功能，拥有查询考生考号和查看姓名等功能，也可以对考生进行专业课的分类，如选择法律专业，学生信息框中只显示法律专业学生。授权功能是对学生的信息进行审核，如果学生信息不符合可以对此学生信息授权不通过，也就无法考试。

**授权功能实现：**

我们在数据库中添加一字段数据为0和1，，通过为1，未通过为0，在考生刚注册信息时授权默认为0。在js中判断是否选中，如果选中某条数据，将会请求后端并将授权对应的0或1传递到后端然后后端会返回数据为1，并修改授权内容将1渲染到对应的数据中，使用Vue框架实现页面不刷新来渲染数据。

**删除考生数据功能实现：**

在选中一些数据后点击删除按钮后，会弹出提示框为‘确认删除吗？’，如果点击确定，则会请求后端，并将学生id传递到后端然后经过处理后会删除学生数据，由于页面使用Vue写的，所以可以实时刷新渲染数据。这样就完成了删除学生功能。

**修改功能：**

修改功能只能选择一个学生，如果没选或多选会提示信息‘请选择一个学生’，在选择一个学生时会跳转到修改页面，并将学生id传递过去。

**搜索功能：**

通过Vue实现搜索功能，可以实时提示搜索的内容。

**专业筛选：**

将选择的数据传递到后端，然后经过处理返回筛选的每条数据，这样就实现了筛选功能。

1. **成绩录取**

**介绍：**

管理员可以动过学生考号去录取成绩。

**功能：**

用Vue获取成绩内容和考号，然后请求后端将对应学生的考号赋值成绩即可。

1. **考生信息修改**

**介绍：**

管理员可以通过此页面完成考生信息修改，如修改编号、姓名、专业、地区、性别、学历、电话等信息。

**功能实现：**

用Vue获取文本中的值然后请求后端，并将所有的内容发送过去，后端在处理成功后返回提示‘修改成功’。如果身份证、电话号、错误或者未填都会提示对应的信息

1. **考生录取结果**

**介绍：**

管理员可以查看考生录取情况，录取那个公司，报考专业，录取专业等信息。

**功能实现：**

用Vue获取后端传递的数据，将数据渲染到页面即可获取相关内容。

1. **管理员登录退出**

**功能实现：**

在点击管理员退出后会清除cookie，从而实现退出管理员登录系统功能。

1. **管理员账号登录**

**介绍：**

不同权限的管理员有不同权限的操作，如普通管理员不可以录取成绩，超级管理员可以录取成绩并能删除、修改、授权等功能。

**功能实现：**

给每个功能模块进行判断，超级管理员为0，普通管理员为1，判断如果为0则可以进行操作，如果为1，则不能进行操作，并提示‘权限不足’。

三、后端部分

后端使用Python中的Flask库作为数据交互平台，目的是为了处理前端传过来的数据。PDF使用Reportlab库制作。使用pymysql连接MySQL数据库。

1. 登陆模块

连接mysql的admin表，利用前端传来的数据，查询在此数据表中的某个唯一的字段，如果没有，返回一个异常，如果有，再次判断密码是否一致，如果不一致，返回密码错误，否则返回登陆成功。学生登陆也在该模块中，如果前端传来的数据是电话号码，则连接account\_s表，获取其中所匹配的字段，判断密码是否一致，从而登陆。

1. 获得所有学生

接收到前端的请求后，直接查询学生表中的数据，返回给前端，从而获得所有学生信息。

1. 获得某个特定的学生

接收前端请求的条件，在学生表中筛选匹配条件的某个人，返回给前端，如果没有，则返回没有这个学生。

1. 获得所有已经录取和未被录取的学生

接收到前端的请求后，在matriculate和unmatriculate中查询所有的值，并依次返回给前端。

1. 学生注册

接收前端发送的信息参数后，先对注册时间进行查看，如果在注册时间段外，是不能够注册的，返回注册时间已过或未开始。如果在注册时间段内，获得学生的每个参数信息，检测数据库中身份证号码和电话号码是否已被占用，如果是，返回身份证或电话号已被注册。如果正常，则录入成功。

1. 学生注册审核

管理员登陆后，在学生管理页面中，会检测学生的授权信息，选择一位或多位学生后发送。接收到多位学生id后，依次处理每位学生的信息，将学生表中的授权信息改为已授权。

1. 管理员对成绩的插入

在插入成绩之前，先检测该学生是否已被审核通过，如果未被通过，是不可以录入成绩的。如果是通过的，就检测是否已经录入过成绩，

如果录入过，就不能再次录入，或者让更高级管理员修改成绩，如果没有，就可以成功录入。

1. 学生录取处理

当录入成绩后，开始处理学生录取情况，首先获得该学生的专业，获取对应公司的录取分数线和录取人数，如果满足了条件，则添加到已录取学生的表中，如果不满足，添加到未录取学生表。

1. 处理准考证PDF和成绩单PDF

处理准考证PDF时，接收要处理的学生ID，根据学生所在的省市，分配最近的考试学校，随机分配考场和座位号。获取考试的时间和学生的数据，将所有的数据，对接到处理准考证PDF的接口中，将处理完的文件地址，返回给前端。

处理成绩单PDF时，接收学生的ID，在两个录取表中查询此ID，应为学生ID是唯一的，所有只有一个返回值，将该数据对接到处理成绩单的接口中，再将处理完的文件，返回给前端。

参考文献

1. 郝兴伟．Web技术导论(第四版)．北京: 清华大学出版社，2018(2021.5重印)．
2. 魏江江．实用软件工程(第四版)．北京: 清华大学出版社，2020(2021.7重印)．
3. Vue官方中文文档．https://cn.vuejs.org．
4. Axios官方中文文档．https://axios-http.com/zh/docs/intro．
5. Flask官方中文文档．https://dormousehole.readthedocs.io．
6. Reportlab电子书．https://www.reportlab.com/docs/reportlab-userguide.pdf．

致谢

首先，非常感谢软件工程这个课程给我这次宝贵的实践机会，我体会到这学期的每门课程的作用，每一门课程都有自己的作用，同时也感谢老师的辛勤指导，辛苦了！在开发项目时给予了我们很大的帮助。展现在自己面前的是一片任自己驰骋的沃土，耕种梦想，成就未来。我们将继续学习，深入实践，不断提升自我，多反思，多总结，吸取经验，为以后打下基础。

总结

经过之前的学习，对程序设计有了⼀定的认识与理解。在校期间，⼀直都是学习理论知识，没有机会去参与项⽬的开发。所以说实话，这次实训，软件项⽬开发对我来说是⽐较抽象的，⼀个完整的项⽬要怎么分⼯以及完成该项⽬所要的步骤也不是很明确。⽽经过这次实训，让我明⽩了⼀个完整项⽬的开发，必须由团队来分⼯合作。

除此之外，我还学到了如何与⼈相处，如何和⼈更好的交流，我们组成⼀个团队⼤家⼀起开发⼀个项⽬，⼤家的交流沟通显得尤为重要，如何将⾃⼰的想法清楚明⽩的告诉队友，如何提出⾃⼰想法的同时⼜不伤害其他的队友的⾯⼦，这些在我的实训⽣活中都有⼀些体会。可是说，第⼀次亲⾝体会理论与实际相结合，让我⼤开眼界。 也是对以前学习的⼀个初审吧，相信这次实训多我以后的学习、⼯作也将会有很⼤的影响，在实训的这段时间⾥这些宝贵的经验将会成为我以后⼯作的基⽯。