软件设计实践报告

单 位：计算机学院\*网络空间安全学院

班 级：

学 号：

姓 名：

任课教师： 刘新

湘 潭 大 学

2020年9月

**高考志愿智能推荐系统**

**一、项目概述**

**1. 概述**

目前市场上有很多高考志愿填报推荐系统，考生只需要输入自己所在省份以及自己的分数，系统就会给出推荐的若干所学校，一般会按照“冲一冲、稳一稳、保一保”的策略各给出三到四所学校名单，做得好的系统还会给出推荐的专业，以及可能录取的概率（从低到高排列）。

**2. 实现原理**

这种推荐系统实现的原理比较简单，它并不是根据考生的分数来推荐学校的，因为每年的分数变化很大，并不具有多大的参考意义。它是查当年所在省份（因为中国高考是分省录取的）的“一分一段表”，然后根据考生的分数计算出他当年的省内排名，再根据过去3~5年每所大学在本省录取线换算出来的排名给出推荐学校。

比如今年湖南省理科某考生570分，当他来查询推荐学校时，系统会先查今年湖南省的一分一段表，570分的排名在36000名左右。然后再查过去5年各所学校在湖南招生的最低录取排名的平均值计算出来，比如湖南科技大学是42000名左右，湘潭大学是32000名左右，长沙理工是30000名左右，于是系统按照录取概率的大小，分别将长沙理工、湘潭大学和湖南科技大学归入到“冲一冲”、“稳一稳”和“保一保”的档次。

如果考虑到各个专业录取的最低分数排名不同，还需要根据专业给出推荐。不过目前各个系统做到分专业推荐的还很少，因为专业录取分数线搜集很困难。

**3. 开发需求**

本系统分为两个相对独立的部分：一个是信息采集（爬虫）部分，一个是面向用户的业务系统，两个部分之间无直接调用关系，只是会访问同一个数据库。

**3.1 信息采集需求**

**（1）学校基础信息采集**

这部分包括两类学校：本科院校和高职专科学校。

本科院校采集的信息包括学校的名称、招生代码，学校类别（一本、二本、三本，公办、民办，985、211、双一流）学校排名（至少包括武书连和网大排名，以近5年的为准，计算平均值），学校专业排名（主要是第4轮学科排名、武书连排名）

高职专科学校采集的信息主要是学校名称、招生代码和民间的排名。

**（2）招生数据采集**

这部分包括各省（全国有31个省份）近5年（2016-2020）的招生数据。包括：历年的各科最低控制分数线（理科、文科必须采集，艺考、体育生等可暂缓），一分一段线数据，各个学校在本省招生的最低投档线（能采集到平均分和最高分更好）。如果可能的话，采集专业投档线。以上分数线均要转换成排名。

今年各学校在本省的招生名额（分专业人数和总人数）。

**3.2 业务系统需求**

本系统在采集到的高考数据基础上实现高考志愿推荐功能。采用Java Web技术实现。

**（1）管理员角色功能需求**

系统预置至少一个管理员，拥有以下功能：

* 账户管理功能：登录、修改密码
* 其他用户管理功能：包括查看、禁用、启用普通用户，
* 统计功能：查看、统计用户充值缴费情况
* 数据管理功能：搜索、查看、统计、修改、删除、增加采集到的各类数据
* 配置功能：增、删、改、查各个阈值。这里主要是指推荐策略需要用到的数值，包括推荐的学校数目、排名变动的范围

**（2）普通用户（考生）角色功能需求**

* 注册功能：手机号码注册
* 账户管理功能：登录、修改密码、密码找回
* 充值功能：支付宝或者微信充值（暂时做假的，有界面即可）
* 查看推荐学校：输入自己的分数，查看推荐结果，如不满意，可以换一批
* 高校浏览功能：查看各个高校的基本数据和在本省的招生情况
* 评价功能：对推荐结果进行评价
* 交流功能：类似BBS板块，考生之间可以发帖互相交流

补充：可设计成充值用户拥有更多的推荐功能，例如不充值只能使用一次推荐，或者推荐的学校较少。

1. **项目设计与分析**

（本人担任业务组组长）

**补充：本人工作概览**

（1）项目原型图设计

（2）前后端调用接口设计

（3）业务数据库设计

（4）Vue实现考生前端页面

（5）协调前后端开发人员进度

（6）协助大组长维护gitee

注意：源码及其他文档已经全部打包

1. **开发工具与开发环境**

开发设备：Windows10

原型图设计工具：Axure RP 8

数据库：Mysql8.0

数据库管理工具：Navicat Premium 12

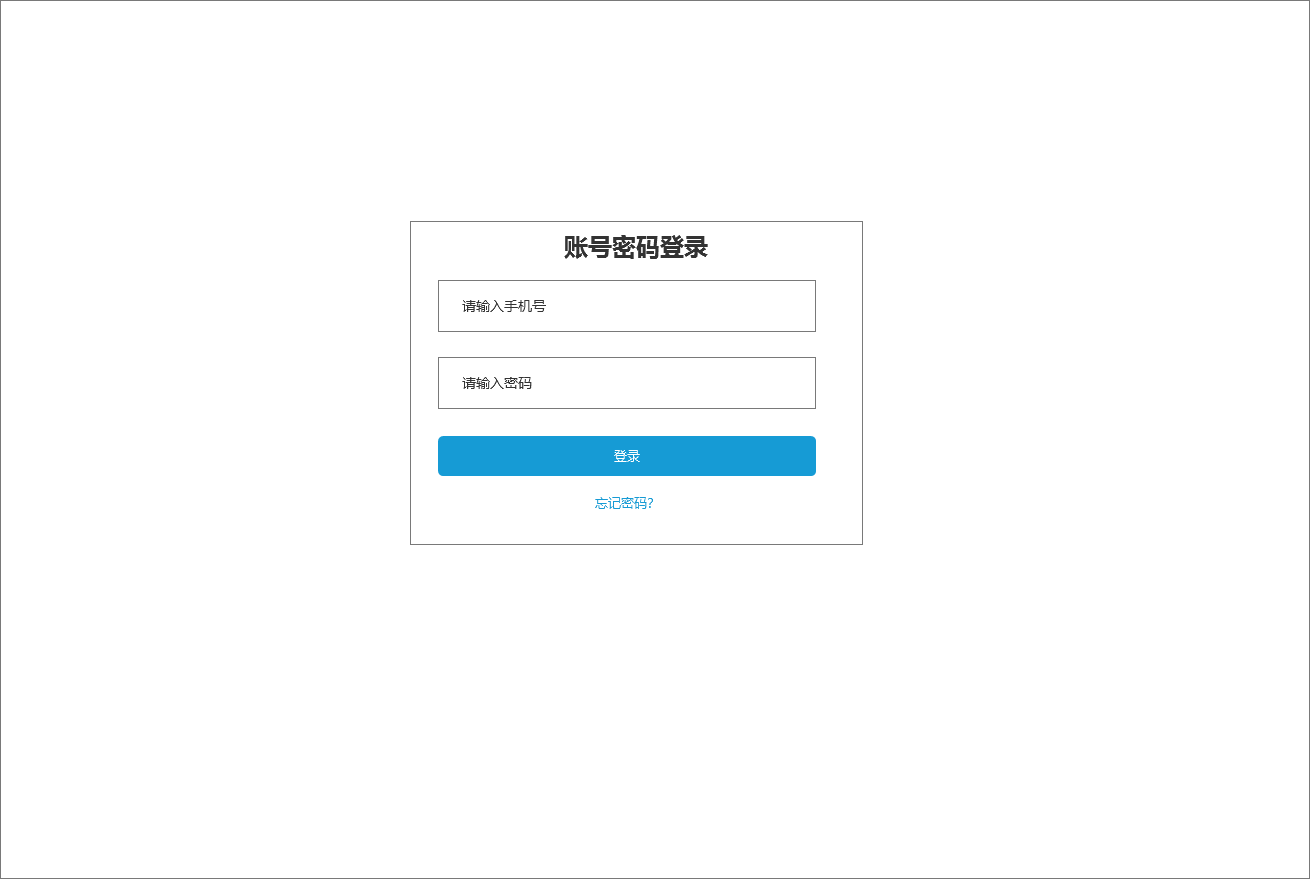
开发环境：Node v12.18.1

前端框架：Vue.js

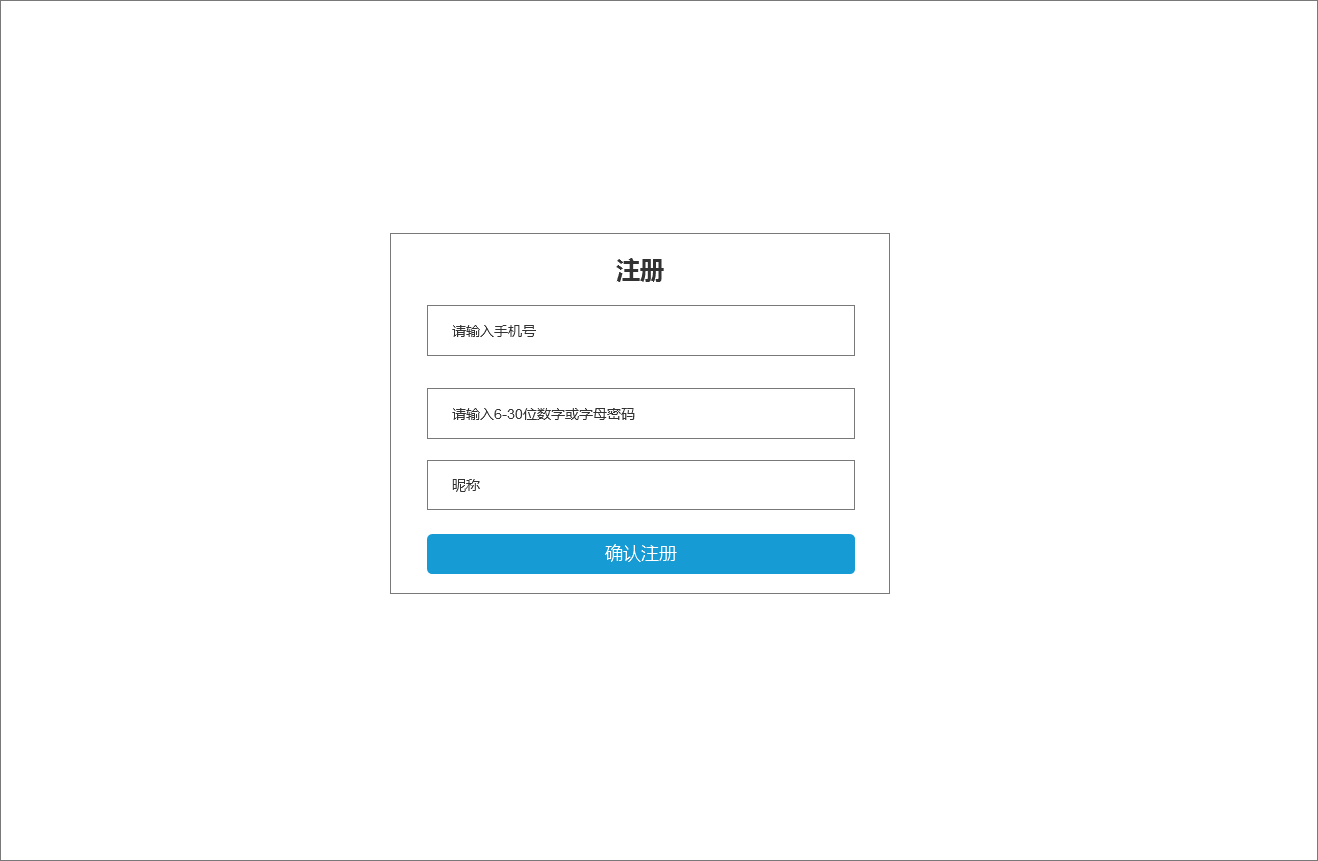
1. **项目页面分析及对应UI原型图**
2. 公共导航栏有首页、社区、剩余查询次数及登录注册等选项
3. 首页主要为考生提供志愿推荐功能，考生输入其总分、省排名、文理科后将跳转至推荐结果页。推荐结果页将显示查询结果以及提供反馈评价功能
4. 社区为考生提供发表言论的功能
5. 剩余查询次数将显示当前考生可用查询次数并提供充值链接
6. 登录与注册



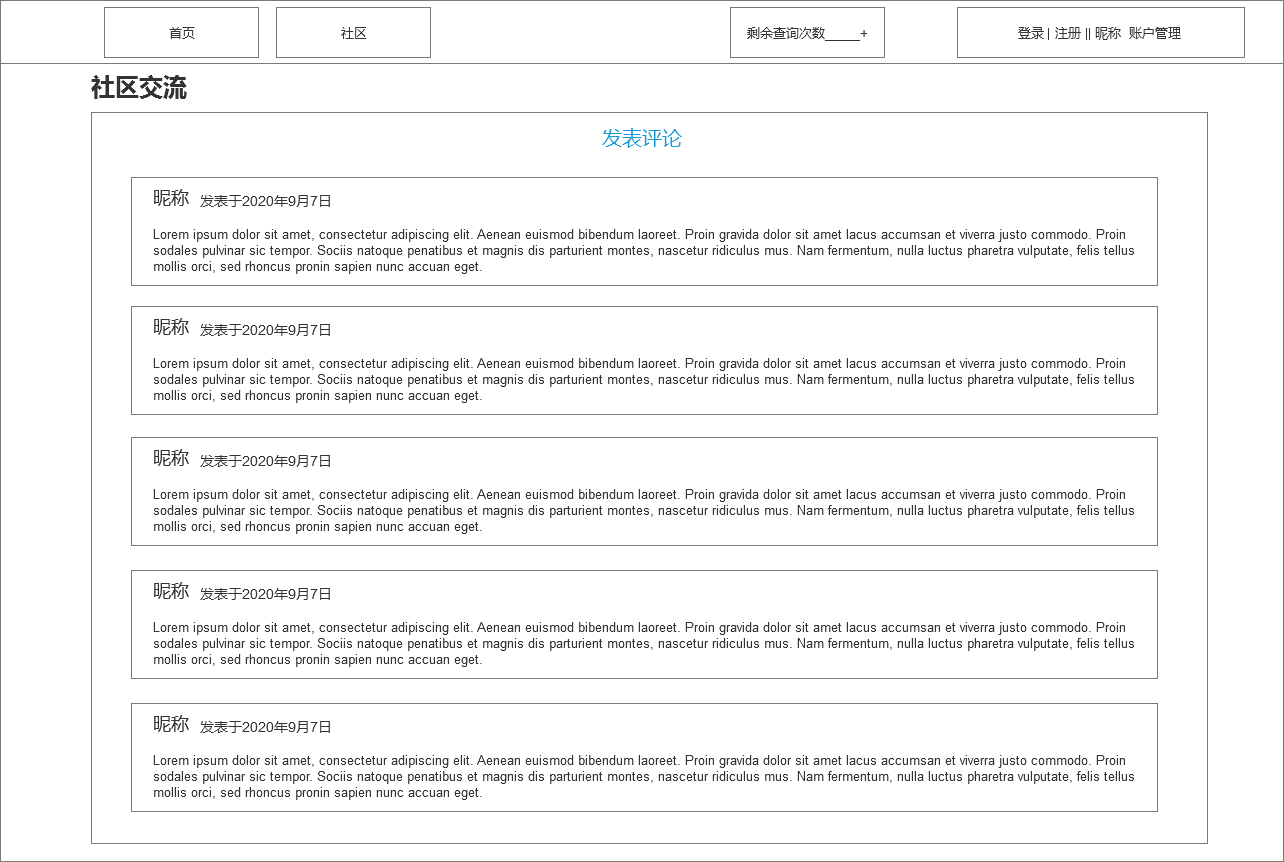
首页（原型图）



登录（原型图）



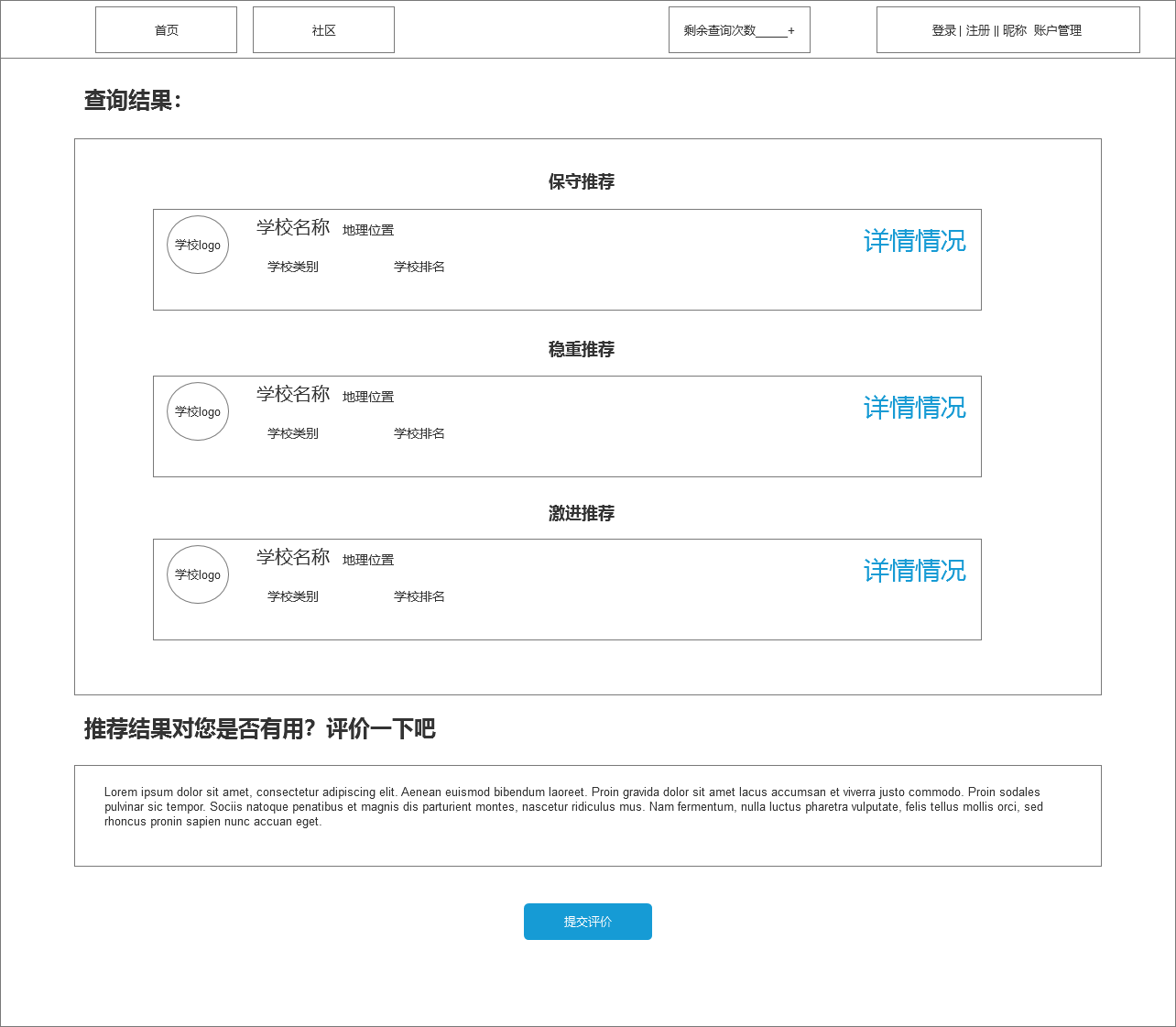
注册（原型图）



社区（原型图）



高校浏览（原型图）



推荐结果（原型图）

1. **前后端接口设计**

设计遵循restful风格：名词+http方法+参数

|  |  |
| --- | --- |
| http方法 | 作用 |
| Post | 创建资源 |
| Get | 获取资源 |
| Put | 更新资源 |
| Delete | 删除资源 |

资源分为：university(高校)、users（用户）、account（账户）、posts（帖子）、costs（费用）

**高校：universities**

|  |  |
| --- | --- |
| 获取高校信息 | |
| 协议： | HTTP |
| 终端路径： | Protocol:host:port/universities |
| URI设计： | Protocol:host:port/universities |
| 方法： | GET |
| 内容类型： | Application/json |
| 查询参数： | 无 |
| HTTP状态码： | 1. 400 、500 |
| 表现形式： | Json |

|  |  |
| --- | --- |
| 按分数、省内排名、分科查询高校推荐信息 | |
| 协议： | HTTP |
| 终端路径： | Protocol:host:port/universities |
| URI设计： | Protocol:host:port/universities?score=<score>&&rank=<rank>&&branch=<branch> |
| 方法： | GET |
| 内容类型： | Application/json |
| 查询参数： | score、rank、branch |
| HTTP状态码： | 200、400 、500 |
| 表现形式： | Json |

|  |  |
| --- | --- |
| 按省、市浏览高校信息 | |
| 协议： | HTTP |
| 终端路径： | Protocol:host:port/universities |
| URI设计： | Protocol:host:port/universities?province=<province>&&city=<city> |
| 方法： | GET |
| 内容类型： | Application/json |
| 查询参数： | province、city |
| HTTP状态码： | 200、400 、500 |
| 表现形式： | Json |

|  |  |
| --- | --- |
| 按名字搜索浏览高校信息 | |
| 协议： | HTTP |
| 终端路径： | Protocol:host:port/universities |
| URI设计： | Protocol:host:port/universities?university\_name=<university\_name> |
| 方法： | GET |
| 内容类型： | Application/json |
| 查询参数： | university\_name |
| HTTP状态码： | 200、400 、500 |
| 表现形式： | Json |

**用户：users**

|  |  |
| --- | --- |
| 用户登录 | |
| 协议： | HTTP |
| 终端路径： | Protocol:host:port/users/login |
| URI设计： | Protocol:host:port/users/login |
| 方法： | POST |
| 内容类型： | Application/json |
| 查询参数： | 无 |
| HTTP状态码： | 200、400 、500 |
| 表现形式： | Json |

|  |  |
| --- | --- |
| 用户注册 | |
| 协议： | HTTP |
| 终端路径： | Protocol:host:port/users/register |
| URI设计： | Protocol:host:port/users/register |
| 方法： | POST |
| 内容类型： | Application/json |
| 查询参数： | 无 |
| HTTP状态码： | 200、400 、500 |
| 表现形式： | Json |

|  |  |
| --- | --- |
| 用户找回密码 | |
| 协议： | HTTP |
| 终端路径： | Protocol:host:port/users/findpwd |
| URI设计： | Protocol:host:port/users/findpwd |
| 方法： | POST |
| 内容类型： | Application/json |
| 查询参数： | 无 |
| HTTP状态码： | 200、400 、500 |
| 表现形式： | Json |

|  |  |
| --- | --- |
| 用户重置密码 | |
| 协议： | HTTP |
| 终端路径： | Protocol:host:port/users/{id}/resetpwd |
| URI设计： | Protocol:host:port/users/{id}/resetpwd |
| 方法： | POST |
| 内容类型： | Application/json |
| 查询参数： | 无 |
| HTTP状态码： | 200、400 、500 |
| 表现形式： | Json |

**账号：accouts**

|  |  |
| --- | --- |
| 用户更新账户信息 | |
| 协议： | HTTP |
| 终端路径： | Protocol:host:port/accouts |
| URI设计： | Protocol:host:port/accouts/{user\_id} |
| 方法： | POST |
| 内容类型： | Application/json |
| 查询参数： | 无 |
| HTTP状态码： | 200、400 、500 |
| 表现形式： | Json |

**帖子/评论：posts**

|  |  |
| --- | --- |
| 用户发表评论/帖子 | |
| 协议： | HTTP |
| 终端路径： | Protocol:host:port/posts |
| URI设计： | Protocol:host:port/posts/{user\_id} |
| 方法： | POST |
| 内容类型： | Application/json |
| 查询参数： | 无 |
| HTTP状态码： | 200、400 、500 |
| 表现形式： | Json |

|  |  |
| --- | --- |
| 获取帖子 | |
| 协议： | HTTP |
| 终端路径： | Protocol:host:port/posts |
| URI设计： | Protocol:host:port/posts |
| 方法： | GET |
| 内容类型： | Application/json |
| 查询参数： | 无 |
| HTTP状态码： | 200、400 、500 |
| 表现形式： | Json |

**费用表：costs**

|  |  |
| --- | --- |
| 用户充值 | |
| 协议： | HTTP |
| 终端路径： | Protocol:host:port/costs |
| URI设计： | Protocol:host:port/costs/{user\_id} |
| 方法： | POST |
| 内容类型： | Application/json |
| 查询参数： | 无 |
| HTTP状态码： | 200、400 、500 |
| 表现形式： | Json |

## 管理系统前端与后端接口说明：

**高校：universities**

|  |  |
| --- | --- |
| 获取高校信息 | |
| 协议： | HTTP |
| 终端路径： | Protocol:host:port/universities |
| URI设计： | Protocol:host:port/universities |
| 方法： | GET |
| 内容类型： | Application/json |
| 查询参数： | 无 |
| HTTP状态码： | 1. 400 、500 |
| 表现形式： | Json |

|  |  |
| --- | --- |
| 修改高校信息 | |
| 协议： | HTTP |
| 终端路径： | Protocol:host:port/universities |
| URI设计： | Protocol:host:port/universities |
| 方法： | POST |
| 内容类型： | POST内容体 |
| 查询参数： | 无 |
| HTTP状态码： | 1. 400 、500 |
| 表现形式： | Json |

|  |  |
| --- | --- |
| 更改高校推荐项信息 | |
| 协议： | HTTP |
| 终端路径： | Protocol:host:port/universities |
| URI设计： | Protocol:host:port/universities |
| 方法： | POST |
| 内容类型： | POST内容体 |
| 查询参数： | 无 |
| HTTP状态码： | 1. 400 、500 |
| 表现形式： | Json |

|  |  |
| --- | --- |
| 删除高校信息 | |
| 协议： | HTTP |
| 终端路径： | Protocol:host:port/universities |
| URI设计： | Protocol:host:port/universities?universities\_id=<universities\_id> |
| 方法： | DELETE |
| 内容类型： | 无 |
| 查询参数： | universities\_id |
| HTTP状态码： | 1. 400 、500 |
| 表现形式： | Json |

**用户：users**

|  |  |
| --- | --- |
| 获取人员信息 | |
| 协议： | HTTP |
| 终端路径： | Protocol:host:port/users |
| URI设计： | Protocol:host:port/users |
| 方法： | DELETE |
| 内容类型： | 无 |
| 查询参数： | universities\_id |
| HTTP状态码： | 1. 400 、500 |
| 表现形式： | Json |

**帖子/评论：posts**

|  |  |
| --- | --- |
| 获取帖子 | |
| 协议： | HTTP |
| 终端路径： | Protocol:host:port/posts |
| URI设计： | Protocol:host:port/posts |
| 方法： | GET |
| 内容类型： | Application/json |
| 查询参数： | 无 |
| HTTP状态码： | 200、400 、500 |
| 表现形式： | Json |

**费用表：costs**

|  |  |
| --- | --- |
| 获取缴费记录 | |
| 协议： | HTTP |
| 终端路径： | Protocol:host:port/costs |
| URI设计： | Protocol:host:port/costs |
| 方法： | GET |
| 内容类型： | Application/json |
| 查询参数： | 无 |
| HTTP状态码： | 200、400 、500 |
| 表现形式： | Json |

1. **业务数据库设计**

共建立五张表，分别是**高校信息表(university)**、**用户表(users)**、**账户表(accouts)、发帖表(posts)**、**费用表(costs)**

**users用户信息表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名称** | **字段类型** | **约束** | **是否为空** | **字段释义** | **备注** |
| id | int(11) | 主键 | 否 | 用户主键 |  |
| username | varchar(64) |  | 否 | 用户名 | 不能重复 |
| password | varchar(64) |  | 否 | 用户密码 |  |
| state | Int(11) |  | 否 | 0表示管理员；  1表示普通用户；  2表示vip |  |

**accouts用户信息表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名称** | **字段类型** | **约束** | **是否为空** | **字段释义** | **备注** |
| id | int(11) | 主键 | 否 | 用户主键 |  |
| username | varchar(64) |  | 否 | 用户名 | 不能重复 |
| phone | Int(11) |  | 否 | 电话号码 |  |
| email | varchar(64) |  | 否 | 邮箱 |  |

**posts发帖表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名称** | **字段类型** | **约束** | **是否为空** | **字段释义** | **备注** |
| id | int(11) | 主键 | 否 | 用户主键 |  |
| username | varchar(64) |  | 否 | 用户名 | 不能重复 |
| time | varchar(64) |  | 否 | 发表时间 |  |
| content | varchar(64) |  | 否 | 发帖内容 |  |

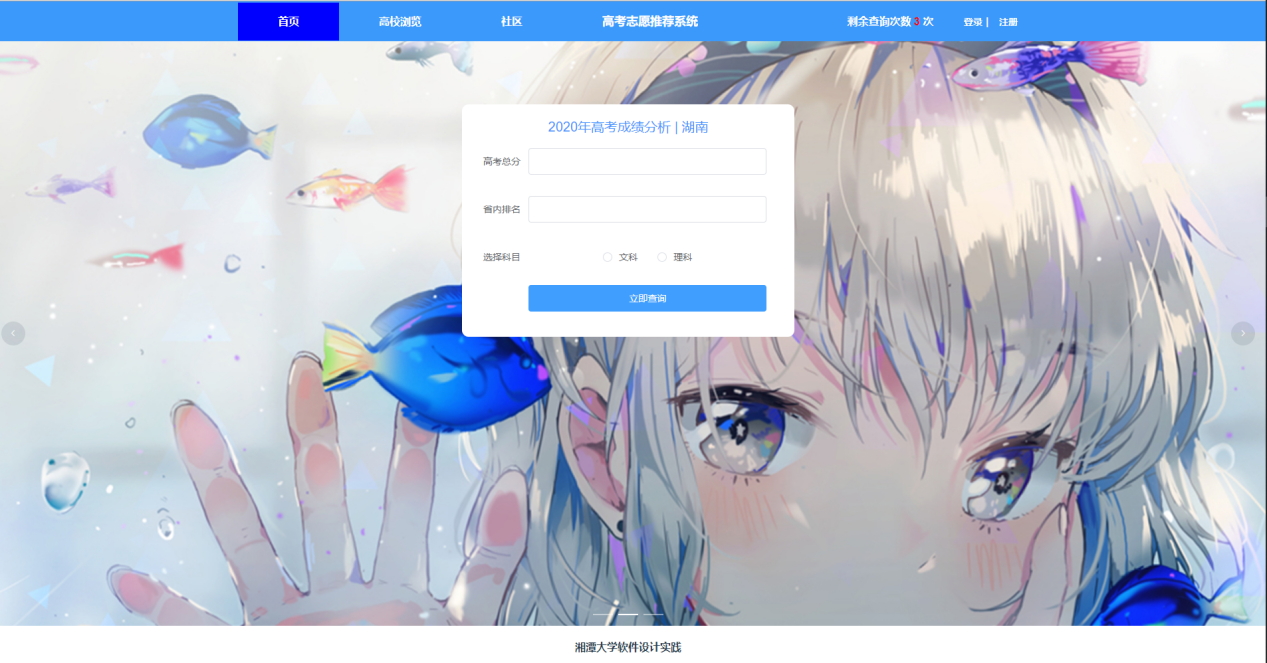
**costs费用表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名称** | **字段类型** | **约束** | **是否为空** | **字段释义** | **备注** |
| id | int(11) | 主键 | 否 | 用户主键 |  |
| username | varchar(64) |  | 否 | 用户名 | 不能重复 |
| time | varchar(64) |  | 否 | 消费时间 |  |
| money | int(11) |  | 否 | 充值金额 |  |

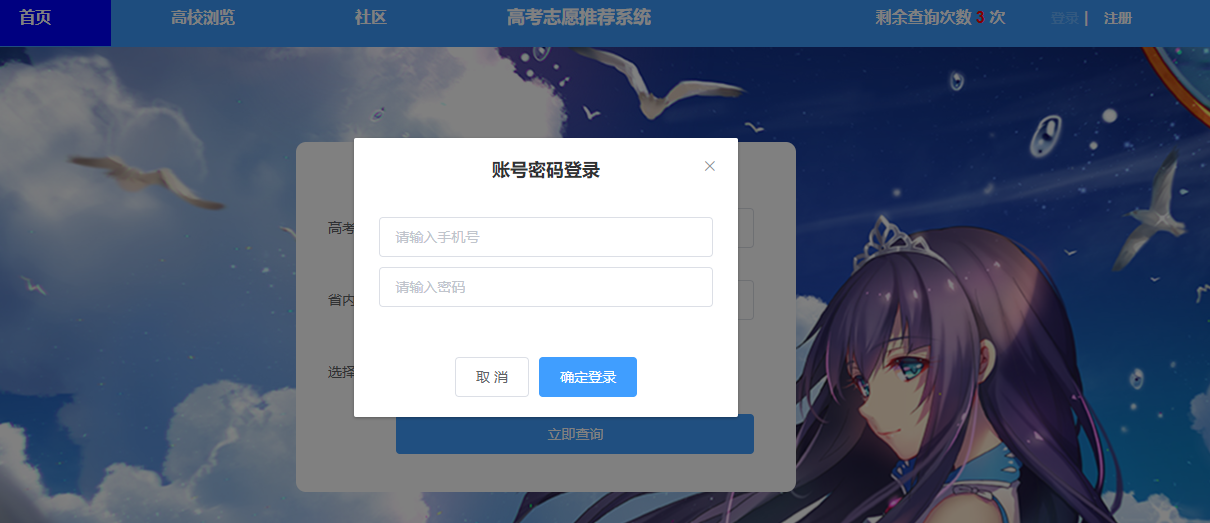
**unversitis高校表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名称** | **字段类型** | **约束** | **是否为空** | **字段释义** | **备注** |
| id | int(11) | 主键 | 否 | 高校编码 |  |
| unversity\_name | varchar(64) |  | 否 | 高校名称 | 不能重复 |
| location | varchar(64) |  | 否 | 位置 |  |
| category | varchar(64) |  | 否 | 类别 |  |
| photo | varchar(64) |  | 否 | 高校logo |  |
| wen\_average\_score | int(11) |  |  | 文科平均分 |  |
| li\_average\_score | int(11) |  |  | 理科平均分 |  |
| content | varchar(64) |  |  | 学校简介 |  |
| rank0 | int(11) |  |  | 武书连排名 |  |
| rank1 | int(11) |  |  | 四轮学科排名 |  |
| low\_score | int(11) |  |  | 最低分数线 |  |

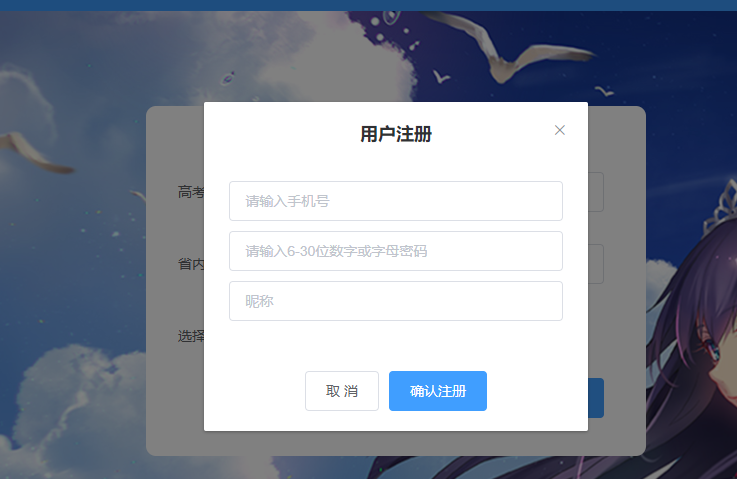
1. **Vue实现考生前端页面**
2. **产品展示**



首页



登录页面



注册页面



高校浏览页面



社区交流页面

1. **技术分析**

考生前端页面实现采用Vue技术栈。

1. 基础框架为vue.js；

Vue (读音 /vjuː/，类似于 view) 是一套用于构建用户界面的渐进式框架。与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与[现代化的工具链](https://cn.vuejs.org/v2/guide/single-file-components.html)以及各种[支持类库](https://github.com/vuejs/awesome-vue" \l "libraries--plugins" \t "https://cn.vuejs.org/v2/guide/_blank)结合使用时，Vue 也完全能够为复杂的单页应用提供驱动。

1. 项目的状态管理采用vuex;

Vuex 是一个专为 Vue.js 应用程序开发的状态管理模式。它采用集中式存储管理应用的所有组件的状态，并以相应的规则保证状态以一种可预测的方式发生变化。Vuex 也集成到 Vue 的官方调试工具 [devtools extension](https://github.com/vuejs/vue-devtools" \t "https://vuex.vuejs.org/zh/_blank)，提供了诸如零配置的 time-travel 调试、状态快照导入导出等高级调试功能。

1. 前端页面路由采用vue-router;

Vue Router 是 [Vue.js](http://cn.vuejs.org/" \t "https://router.vuejs.org/zh/_blank) 官方的路由管理器。它和 Vue.js 的核心深度集成，让构建单页面应用变得易如反掌。包含的功能有：嵌套的路由/视图表、模块化的、基于组件的路由配置、路由参数、查询、通配符、基于 Vue.js 过渡系统的视图过渡效果、细粒度的导航控制、带有自动激活的 CSS class 的链接、HTML5 历史模式或 hash 模式，在 IE9 中自动降级、自定义的滚动条行为。

1. 除了使用基础的html,css,javascript外，还引入了element-ui的vue样式库。

element-ui是一个ui库，它不依赖于vue。但是却是当前和vue配合做项目开发的一个比较好的ui框架

1. 数据交互使用axios请求插件；

Axios 是一个基于 promise 的 HTTP 库，可以用在浏览器和 node.js 中。它具有如下特性：从浏览器中创建 [XMLHttpRequests](https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest" \t "http://www.axios-js.com/zh-cn/docs/_blank)、从 node.js 创建 [http](http://nodejs.org/api/http.html" \t "http://www.axios-js.com/zh-cn/docs/_blank) 请求、持 [Promise](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise" \t "http://www.axios-js.com/zh-cn/docs/_blank) API、拦截请求和响应、转换请求数据和响应数据、取消请求、自动转换 JSON 数据、客户端支持防御 [XSRF](http://en.wikipedia.org/wiki/Cross-site_request_forgery" \t "http://www.axios-js.com/zh-cn/docs/_blank)。

1. 使用webpack进行代码打包；

[Webpack](https://www.leiue.com/tags/webpack" \o "更多关于 Webpack 的文章" \t "https://www.leiue.com/_blank) 是一个开源的前端打包工具。Webpack 提供了[前端开发](https://www.leiue.com/tags/%e5%89%8d%e7%ab%af%e5%bc%80%e5%8f%91" \o "更多关于 前端开发 的文章" \t "https://www.leiue.com/_blank)缺乏的模块化开发方式，将各种静态资源视为模块，并从它生成优化过的代码。 Webpack 可以从终端、或是更改 webpack.config.js 来设置各项功能。

1. 使用npm包管理器进行代码管理；

NPM是随同NodeJS一起安装的包管理工具，能解决NodeJS代码部署上的很多问题。例如：允许用户从NPM服务器下载别人编写的第三方包到本地使用。

允许用户从NPM服务器下载并安装别人编写的命令行程序到本地使用。允许用户将自己编写的包或命令行程序上传到NPM服务器供别人使用

1. 测试浏览器使用chrome;

目前前端开发用的最多的浏览器，js,css的特性兼容较好。

1. 通过Vue Devtools进行调试。

vue-devtools是一款基于chrome游览器的插件，用于调试vue应用，这可以极大地提高我们的调试效率。

管理前端页面使用Vue-element-admin脚手架开发。

这是一个集成大部分功能的后台管理系统框架，也就是说大多数的组件都帮我们写好了，它是基于vue-element进行开发布局的。

1. **模块分析**

根据原型图设计，我将页面分为模块与组件，且模块可以在页面中多次使用，组件可以被模块多次使用。

|  |  |  |
| --- | --- | --- |
| 名称： | 分类： | 作用： |
| 导航栏 | 模块 | 公共导航，可在不同页面复用。 |
| 登录 | 组件 | 用于登录功能 |
| 注册 | 组件 | 用于注册功能 |
| 找回密码 | 组件 | 用于找回密码功能 |
| 底部栏 | 模块 | 可复用的footer |
| 轮播图 | 组件 | 产生轮播效果 |
| 志愿查询框 | 组件 | 用于录入考生分数及相关信息 |
| 评论区 | 组件 | 用于展示评论相关内容 |
| 高校推荐区 | 组件 | 用于展示高校信息 |

1. **源码分析**
2. 项目目录：



public:公共资源目录

src:存放主要代码

src/assets:存放静态资源

src/components:存放可复用的组件

src/router:项目路由文件

src/views:存放视图模块

src/App.vue:项目页面入口

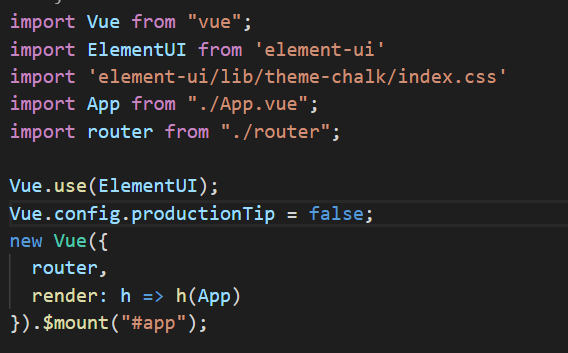
src/main.js:项目主要js文件入口

src/store.js:用于管理数据

package.json：项目依赖目录

1. 模块源码：

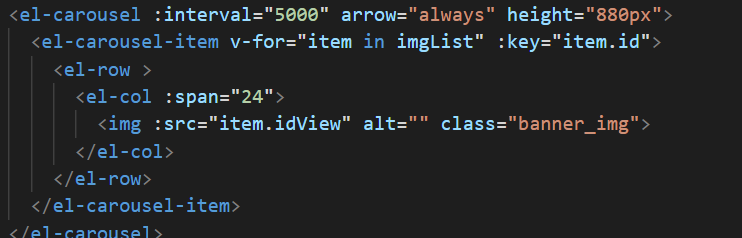
Main.js:页面和路由并将其挂在到根节点上。



Router/index.js:用于配置不同的页面路由。由于vue是单页面应用，所以需要路由文件来匹配对应的渲染视图模块。



el-carousel:轮播图组件：



数据：imgList

imgList: [

{id: 0, idView: require('../assets/紫发.jpg')},

{id: 1, idView: require('../assets/水彩01.jpg')},

{id: 2, idView: require('../assets/水彩02.jpg')}

],

查询组件：



数据：

form: {

score:"",

rank:"",

subject:""

}

UniversityPost.vue高校信息区组件：



数据：

props:{

universityName:String,

location:String,

rank:String,

leibie:String

}

UserPost.vue用户帖子组件：



数据：

props:{

name:String,

time:String,

words:String,

}

1. **项目运行情况及个人心得**

运行情况：

前端页面在分辨率大于1024px\*980px的机器上运行正常,在其他分辨率下会出现错版，页面的兼容性还存在问题，需要后期修改，其他一切正常！

个人心得：

通过此次训练，让我加深了对一个完整项目的理解，从原型图到接口设计，从数据库到页面建构，通过不断的查阅资料和观看视频，建立起了一个相对完善的系统，把一个项目从构思到建立的大部分流程都走了一遍。收获成果丰富。

1. **参考文献**

RESTful API开发实战----Sanjay Patni著、郭理勇（译）

Vue.js教程 <https://cn.vuejs.org/v2/guide/>

Vue-router实战视频 <https://vueschool.io/courses/vue-router-for-everyone?friend=vuejs>

Element:[https://element.eleme.cn/2.0/#/zh-CN/component/installation](https://element.eleme.cn/2.0/" \l "/zh-CN/component/installation)