**高考志愿填报助手**

[一、项目需求分析 2](#_Toc142648300)

[（一）立项背景 2](#_Toc142648301)

[（二）用户需求分析 2](#_Toc142648302)

[（三）功能需求分析 2](#_Toc142648303)

[二、模块设计 3](#_Toc142648304)

[（一）、登录注册 3](#_Toc142648305)

[（二）、智能推荐 5](#_Toc142648306)

[（三）、文本智能分析（实体识别） 6](#_Toc142648307)

[（四）、大学专业智能查询（实体查询） 10](#_Toc142648308)

[（五）、查关系（关系查询） 11](#_Toc142648309)

[（六）、大学专业概览（知识概览） 12](#_Toc142648310)

[（七）、AI问答（问答） 13](#_Toc142648311)

[（八）、分数智能分析 14](#_Toc142648312)

[（九）、反馈 15](#_Toc142648313)

[（十）、用户中心 16](#_Toc142648314)

[三、功能实现过程及描述 19](#_Toc142648315)

[（一）数据收集 19](#_Toc142648316)

[（二）数据库搭建 20](#_Toc142648317)

[（三）前端搭建 20](#_Toc142648318)

[（四）后端搭建 20](#_Toc142648319)

[四、运行环境说明 20](#_Toc142648320)

[五、测试 21](#_Toc142648321)

[（一）模块功能测试 21](#_Toc142648322)

[（二）项目上线测试 21](#_Toc142648323)

[六、部署 22](#_Toc142648324)

[（一）服务器配置 22](#_Toc142648325)

[（二）nginx静态资源部署 22](#_Toc142648326)

[（三）uwsgi动态资源部署 22](#_Toc142648327)

[（四）neo4j图关系数据库部署 23](#_Toc142648328)

## 一、项目需求分析

### （一）立项背景

高考志愿填报深刻影响考生个人长远职业发展，从生涯教育理论来看，高中正处于学生的生涯探索时期，高考填报志愿是生涯规划的重要环节，志愿填报对于每一个高考毕业生来说都至关重要。

但是，在当前社会信息繁杂且良莠不齐、有效资源较难获取的背景下，人们在获取、处理和传递信息时,由于不同的知识水平、文化背景、信息来源等因素的影响,导致对于同一信息的理解和解释存在差异和不确定性，这会形成信息的不对称，导致信息差较大。基于上述原因，考生较难获得关于大学、专业、就业的全部信息，常常受到信息壁垒隔阂现象的影响。含有大量专业术语的招生政策措施和录取条件也可能使考生一头雾水，不明所以。这导致对于已经结束高考的考生而言，选择最适合自己的大学、专业是一个难题。

由于网络信息的分散化以及真实性、有效性的问题，考生难以横纵向对比各心仪大学的多项内容，如地理位置、食宿条件、校园环境、招生情况等，面对海量的信息用户只能反复多次使用搜索引擎去获取不同的信息，而由于对招生简章和招生情况的了解不充分也常常无法有效利用自己的高考分数。所以在面对海量的数据的情况下，如何利用现有的平台去迅速了解掌握相关的大学信息是很难做到的。

基于此，我们小组项目以服务高考志愿填报为出发点，以解决考生报考过程中实际遇到的困难为目标，为广大考生提供更加全面便捷的获取大学招生信息、大学校园环境等资源的系统。

### （二）用户需求分析

作为一名高考结束之后等待志愿填报的高考考生，他需要了解他的分数可以上什么大学，根据感兴趣的类型去了解和选择合适自己的专业，了解相关的政策措施及对应的含义，根据他喜爱的知名校友或教师选择对应的学校，和智能AI机器人对话解决自己的疑问。对于用户本身，可能会对词条进行评论和反馈，通过他人的评论去私信关注他人、交流彼此之间的信息，收藏自己感兴趣的词条。

### （三）功能需求分析

能够根据用户给出的分数进行具体分析，给出该考生的排名以及该分数段能够上的专业。

能够根据用户的查询需求，返回对应的大学、专业、政策、学校特色、优秀校友、优秀教师等信息、帮助考生快速了解相关概念。

能够根据用户提供的文本，给出对应的文本分析结果，结果中标记所有的能够被识别的大学、专业、学校特色政策等概念，以及对应的分词结果。

能够根据用户的偏好的内容结合与该用户相似的其他用户的浏览记录，对应返回与之有关系的内容作为推荐，帮助用户拓宽报志愿思路。

能够为用户提供一个更加人性化的、更能精确把控用户需求的并且始终充满关怀的聊天对象（AI），并且支持更多更广泛的相关内容的回复。

能够为用户提供一个可以进行用户间信息交流、交换的平台。

能够为用户提供一个评论词条、反馈词条的功能、

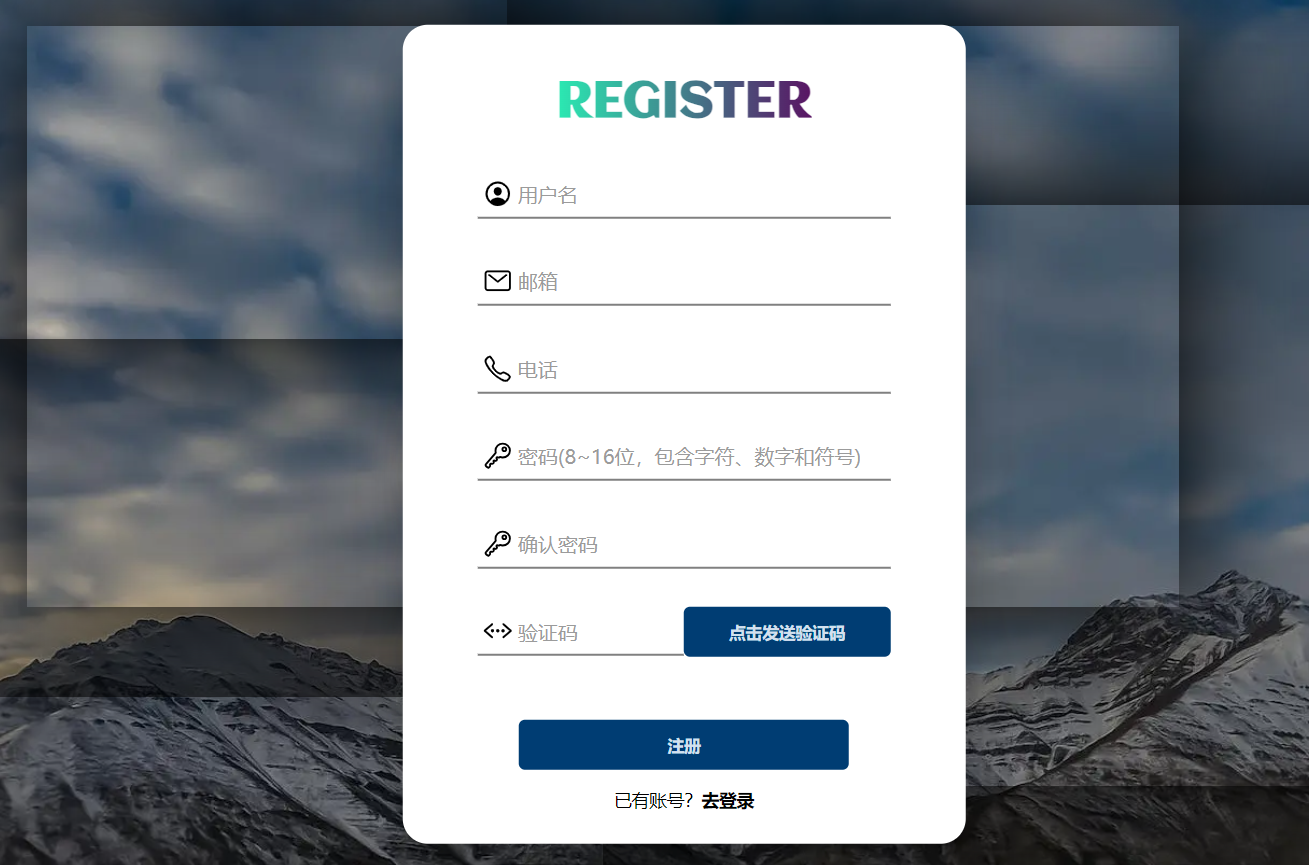
能够为用户提供一个查看历史记录、查看收藏的途径

## 二、模块设计

### （一）、登录注册

**注册账号：**

* 用户填写个人信息（昵称、手机号、邮箱、密码、确认密码、验证码），
* 前端校验表单格式（正则表达式验证手机号、邮箱、密码格式）
* 后端校验手机号、邮箱是否已注册，验证码是否有误
* 注册时，密码hash加密



**用户登录：**

* 用户填写（手机号+密码或者邮箱+密码）
* 前端校验表单格式（正则表达式验证手机号、密码or邮箱、密码）
* 后端校验用户信息，若失败返回失败信息，若成功则创建cookie与session并在请求头中添加cookie以及session\_id来标记用户登录状态，返回个人信息给前端
* 前端对用户个人信息进行本地持久化保存，记录用户登陆状态，用户可在一定时间内免登录进入网站
* 前端同时设置路由导航拦截，每次跳转页面像后端发起同步形式请求，判断该用户是否存在登陆状态，若没有登录则强制跳转到登录界面，禁止其访问。





**忘记密码：**

* 用户填写密码
* 前端校验密码（正则表达式验证密码），请求后端发送验证码
* 用户填写验证码
* 后端校验验证码，不成功返回失败信息，前端跳回登录页面，成功则进行下一步
* 用户填写新密码，后端校验，存储，前端跳回登录界面



### （二）、智能推荐

准备工作：从数据库中读取所有用户的浏览记录，并统计浏览记录三大模块的数量作为模块特征向量

一次过滤：通过欧氏距离和余弦相似度对模块特征向量进行计算得到相似度，进行一次过滤

二次过滤：判断登录用户与该用户实体浏览记录是否有重叠。若无，则筛去实现二次过滤

三次过滤：利用fasttext模型获得每个实体的实体特征向量，并将实体向量乘以时间系数相加得到总和作为每个用户的用户特征向量。通过欧氏距离和余弦相似度对用户特征向量进行计算得到相似度，进行三次过滤

最终推荐列表：对筛选过的用户的所有实体所对应用户的相似度相加得到一个权重值，比较排序后得到最终的推荐列表



### （三）、文本智能分析（实体识别）

用户提供一段不限字数的文本

后端对文本进行分词以及实体识别

前端接受后端返回文本和被标记的实体，动态渲染带有超链接的内容文本

所有文本均以本地化形式临时存储，方便用户多次点击查询其中的内容信息



**详情页面：**

根据传入的实体名称以及实体标签，选择对应的详情模板进行渲染

**大学模板：**

提供点击星星可以添加到收藏页面中功能

根据大学名称，匹配对应的学校风景、学校信息（邮箱、电话等）、食宿情况、食宿图片、学校简介、基本信息、排名信息等

根据对应的大学名称、匹配该大学所有的专业信息，生成概念图云、大学信息详情表

根据对应的实体（名称、标签），匹配对应的评论内容



**百度词条模板：**

根据对应的名称与类别，匹配对应的词条信息、词条简介、

若类别是人：返回所属院校信息，同校师生信息，并生成图云

其他则搜索相关概念并返回

根据对应的实体（名称、标签），匹配对应的评论内容



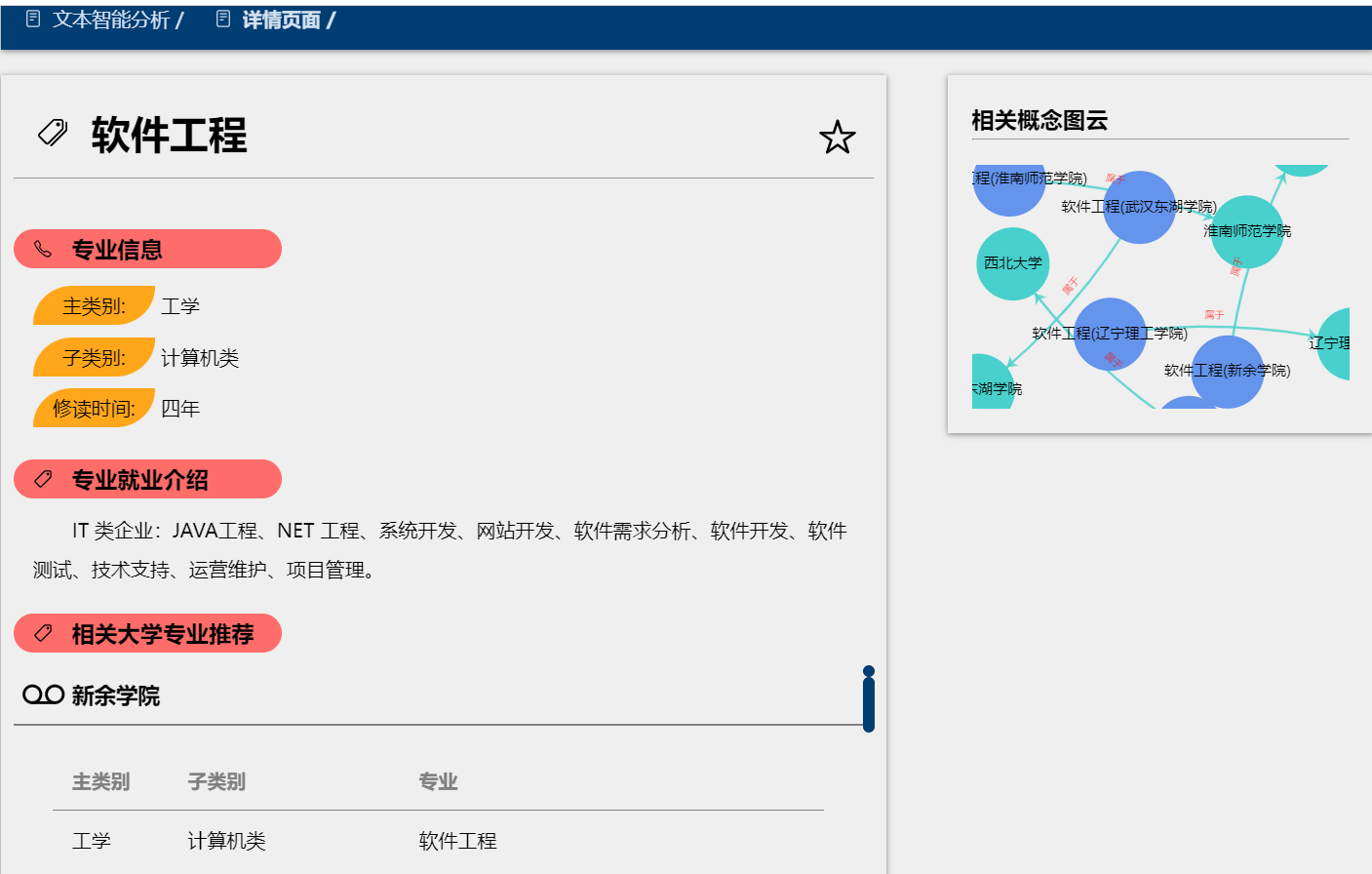


**专业模板：**

根据对应的名称与类别，返回专业信息、以及相关专业推荐

搜索相关概念，生成图云

根据对应的实体（名称、标签），匹配对应的评论内容，同时用户可以在这里留言，看其他用户的留言信息，或者点击用户头像跳转到对应用户的主页面



**评论区**

评论区可以查看其他用户对该词条的评论与补充，如果你感兴趣，可以悬浮其头像上，点击进入他的主页





个人主页根据用户id以及其隐私设置，返回对应用户的详细信息，隐私设置为保密的则显示未公开

同时，若收藏的隐私为公开，则会显示对应用户的收藏记录，点击记录可跳转到对应的详情界面。

点击私信可以与对方开展聊天，点击关注可以关注对方

好友列表在导航栏右上角，用户头像处获得，点击好友可以跳转对应用户主页



### （四）、大学专业智能查询（实体查询）

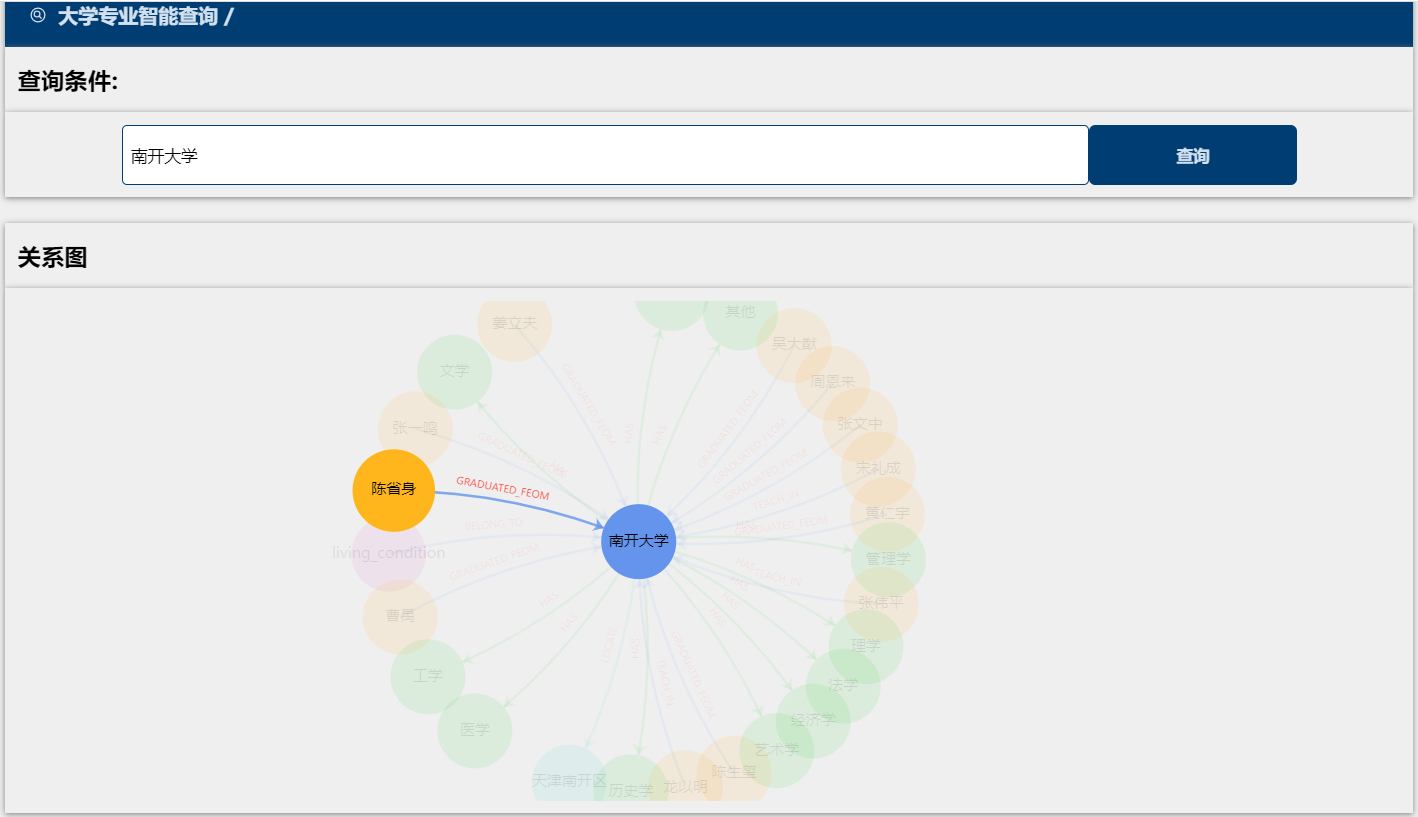
用户输入想要搜索的大学、专业

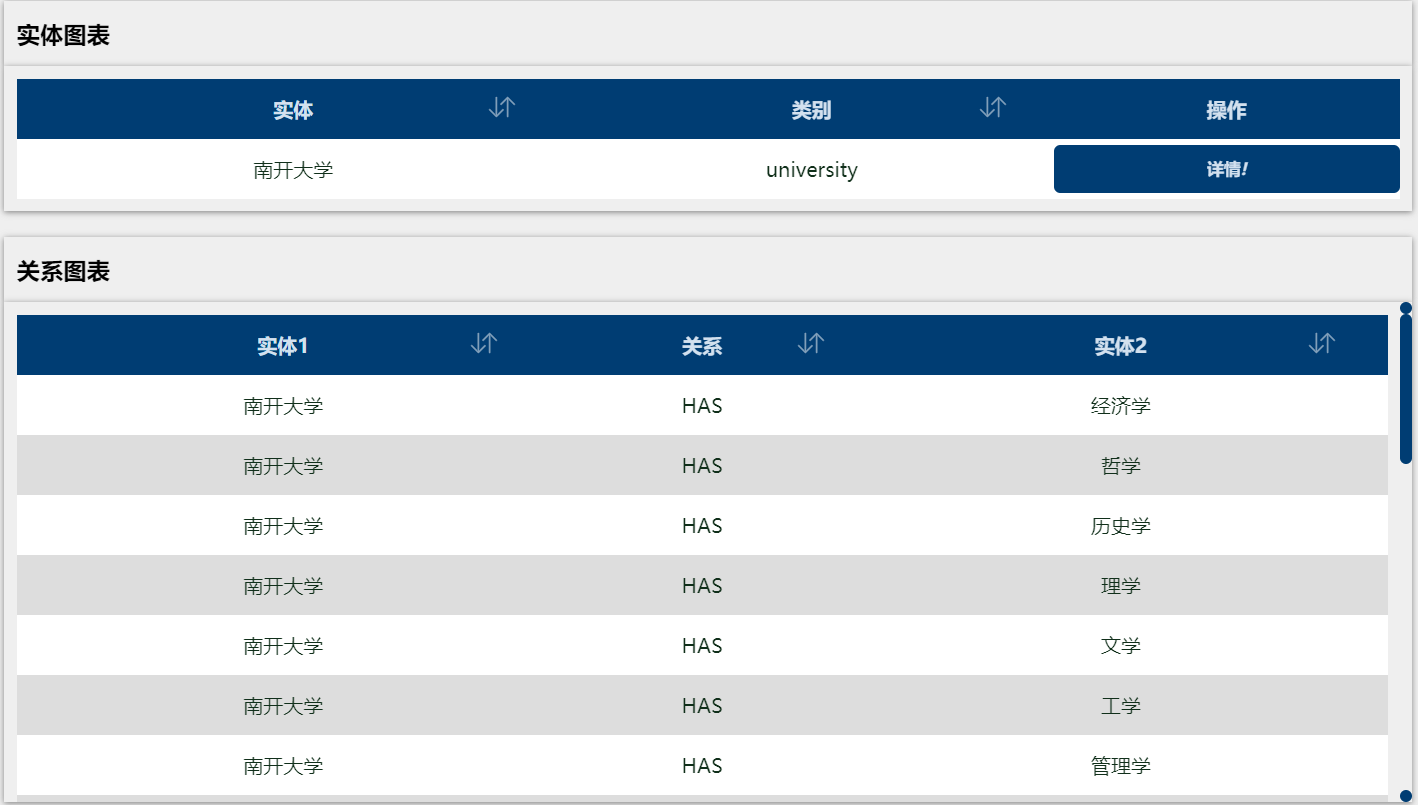
后端分词识别其中的实体，根据识别到的实体搜索对应存在关系的实体与内容，返回前端

前端生成关系图：关系图包含用户输入实体内容以及与之有关的实体的内容

生成实体图表：显示所有识别到的实体，同时点击详情可以跳转到对应词条的详情页面

生成关系图表：对关系图通过表格形式进行更加清晰的展示





### （五）、查关系（关系查询）

用户输入要查找的实体和关系

后端根据用户的输入进行处理，分为四类

只有一个实体、

一个实体一个关系

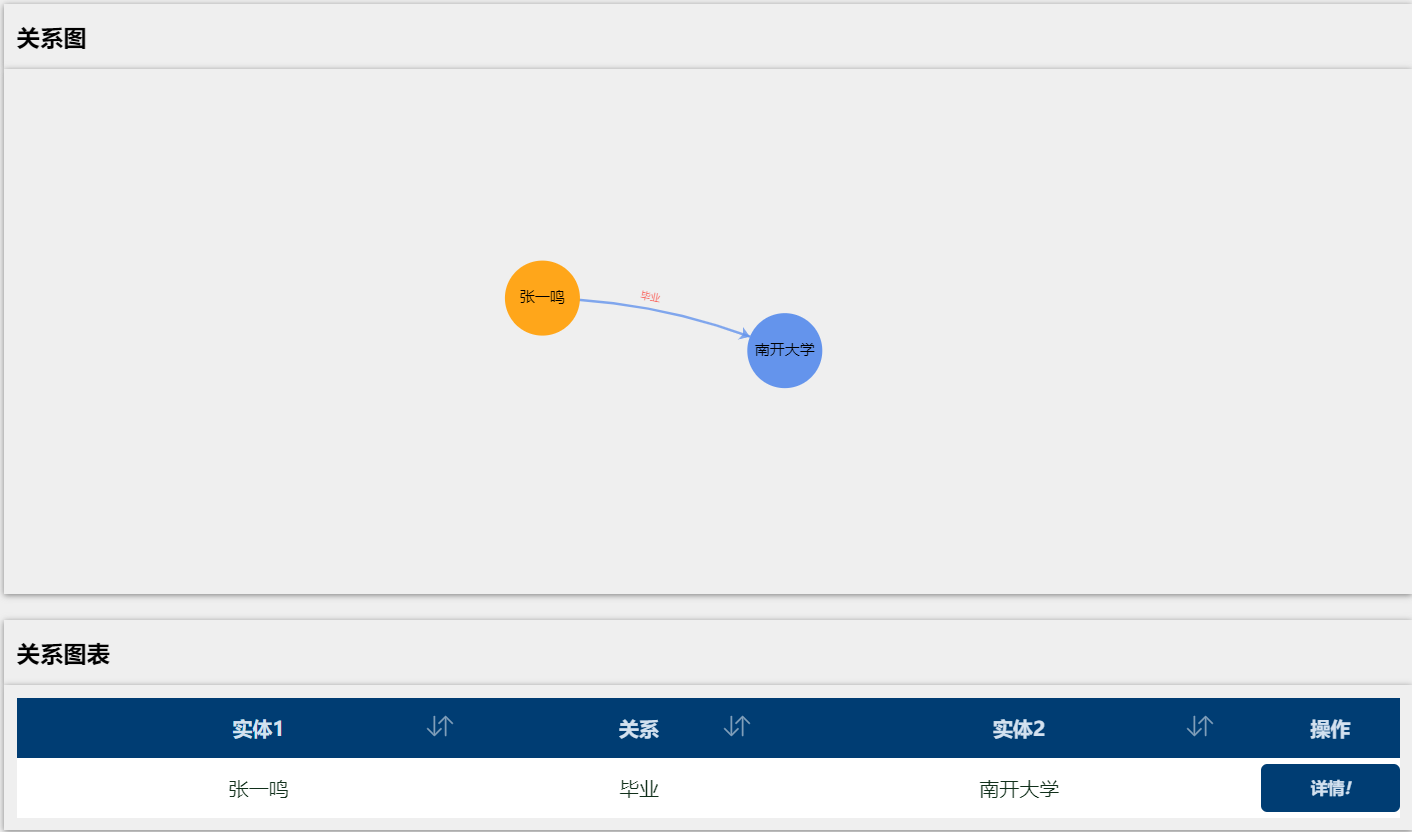
两个实体

两个实体、一个关系

对应每一类进行实体识别与查找，然后返回给前端

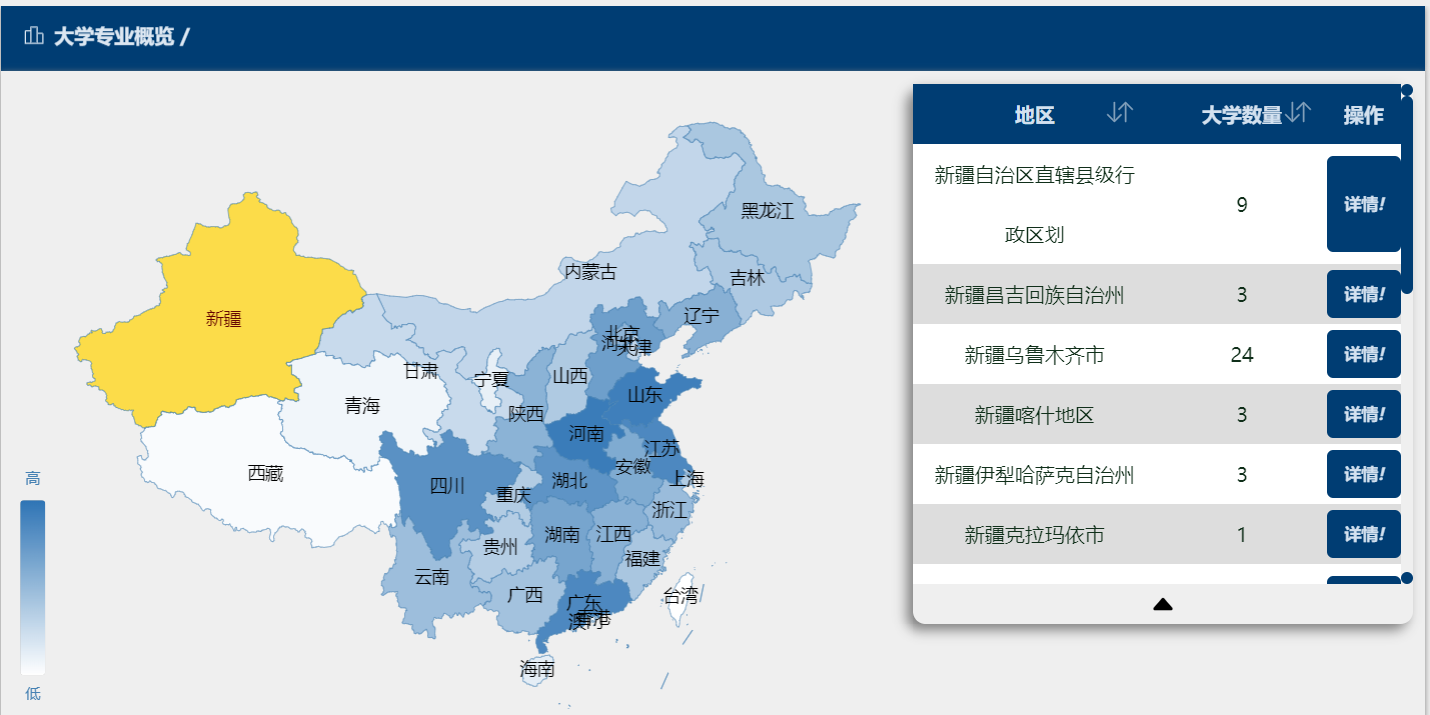
前端对后端返回的数据进行展示，展示内容包括关系图以及关系图表。

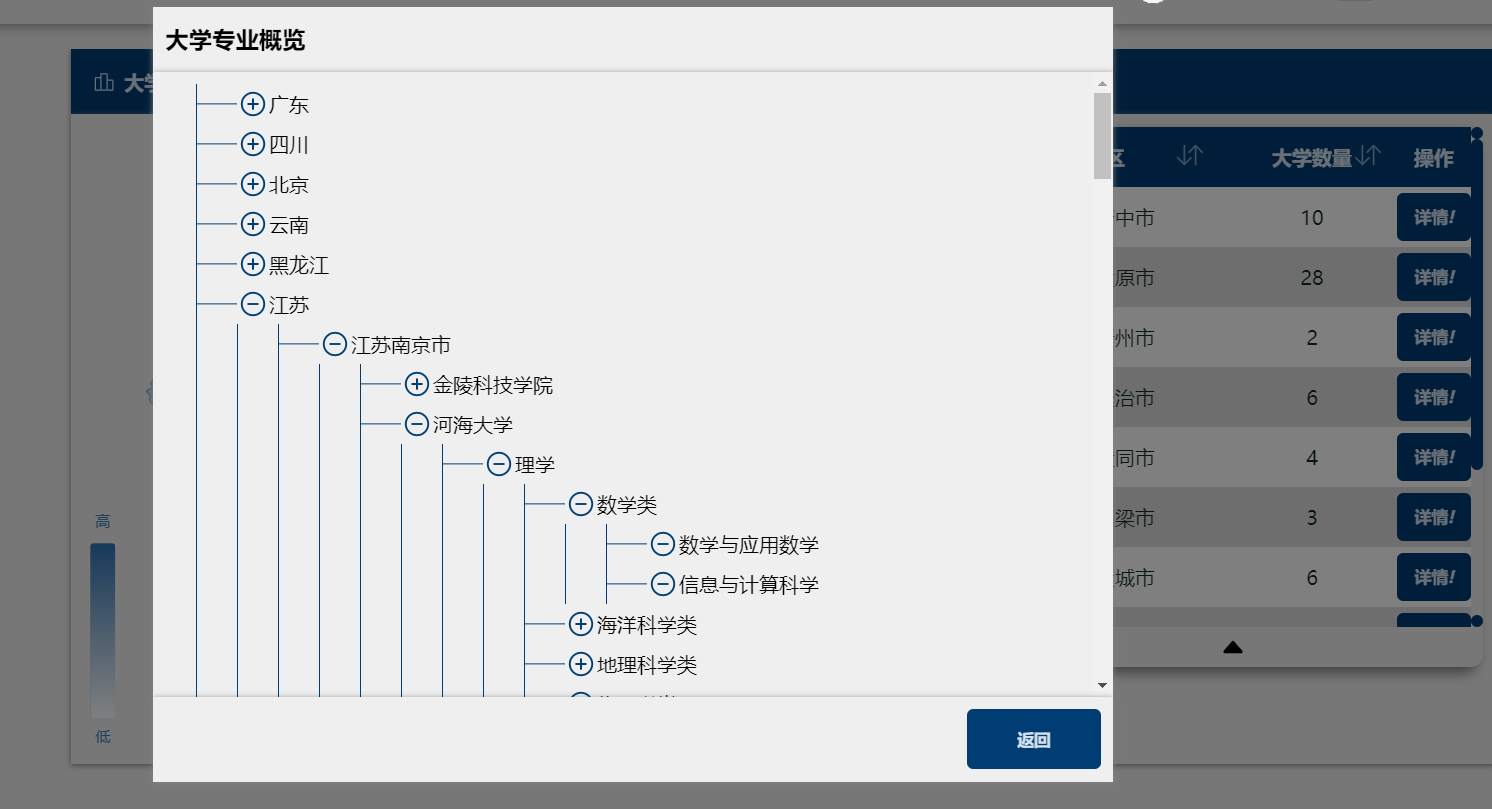




### （六）、大学专业概览（知识概览）

大学专业概览涵盖省市、大学、专业多个概念，提供一个总览的功能。用户可以点击地图的对应省份，查看对应省份的地区所拥有的大学数量，点击右边的详情则可以查看具体的大学专业概览。





### （七）、AI问答（问答）

考虑大语言模型的部署困难，我们通过使用大模型的API接口，来近似达到部署大语言模型的效果。（本次使用阿里云的通义千问的API）

首先调用实体识别函数识别语句中所有的实体名称，记录其名称与类型；之后使用正则表达式匹配出类似于 省/分数/排名/专业 等标签名称；再使用find函数判断该句中是否含有某种关系词，如'拥有'、'属于'、'去' 等。

完成上诉步骤后，开始进行分类：如果没有任何实体，如问候之类的，直接交给 API 处理；如果有一个实体，判断该句中是否有 某类标签 或 关系词，如果有某类标签或关系词，则将其作为参数传入（在 Cypher 语句中，标签直接插入，关系词使用 any 函数插入）；然后分别调用不同的函数进行处理；如果有两个实体，同第二种情况类似，分两种情况；如果有三个实体，则寻找任意两个实体之间的最短路径并融合（运算量极大，舍弃）；当实体数量大于等于3时，直接交由真正的AI处理。

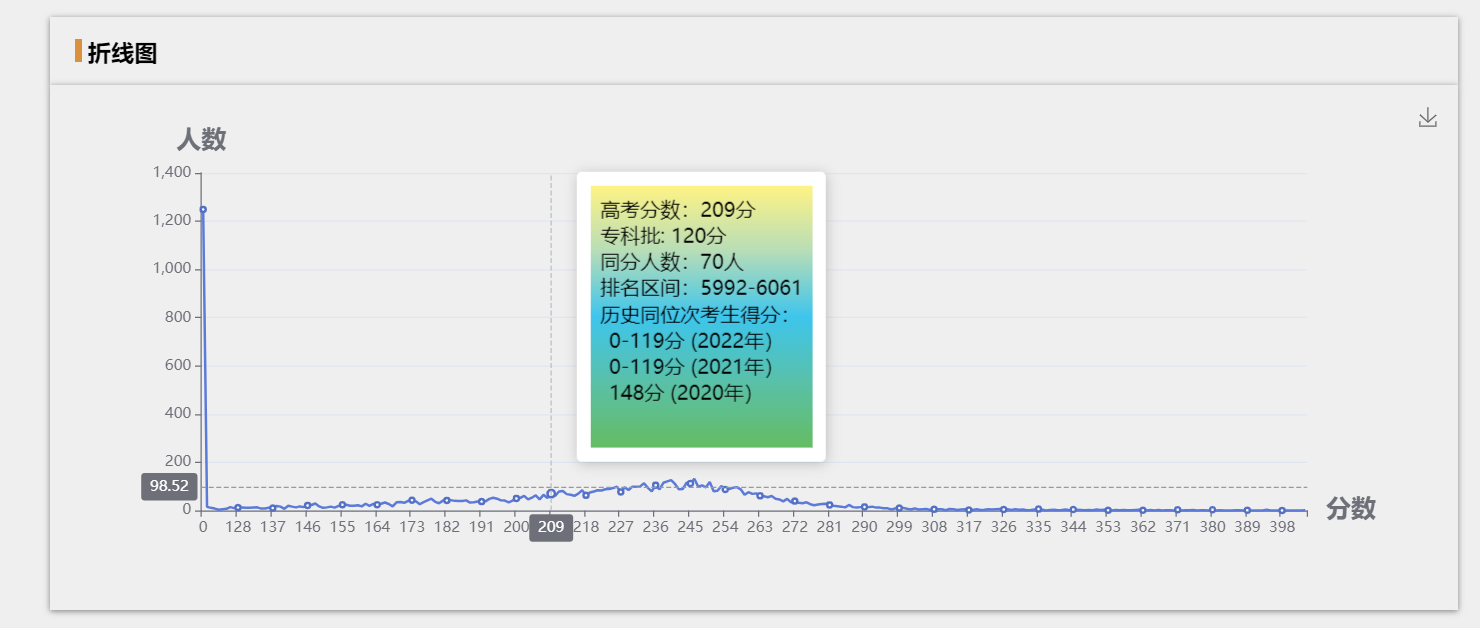
这样，我们即实现了对志愿填报垂直领域知识的精准回答，又不缺少一个AI所具备的真正的聊天功能，实现了大模型与专业领域的巧妙结合。



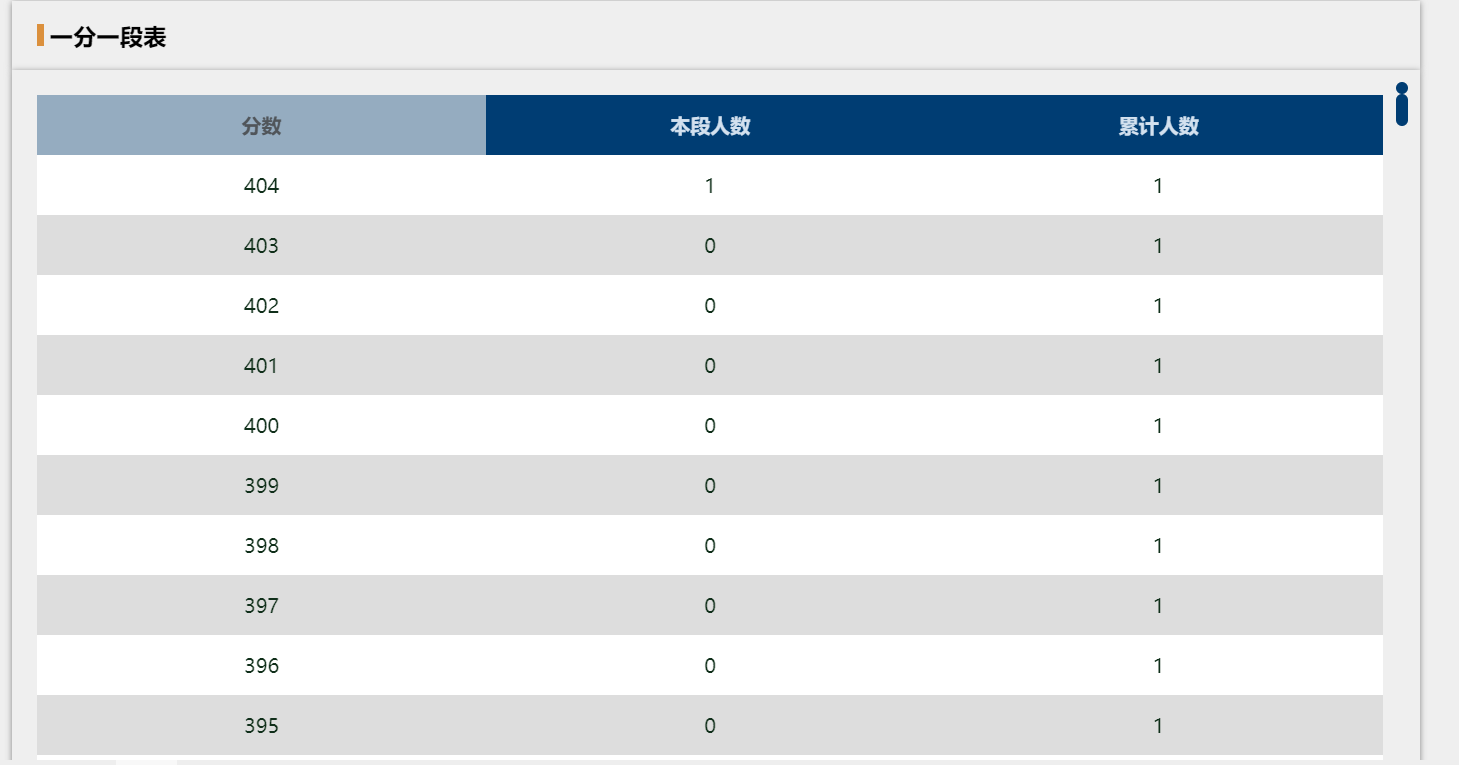
### （八）、分数智能分析

用户选择自己的所在地区、年份、考生类别、成绩类型，

后端对用户选择进行处理，返回该地区该年份该考生类型该成绩类型的一分一段折线图以及具体的一分一段表



在折线图中，用户可以查看同分段人数、排名、历史同分段排名等等有用的可以直观对比的信息



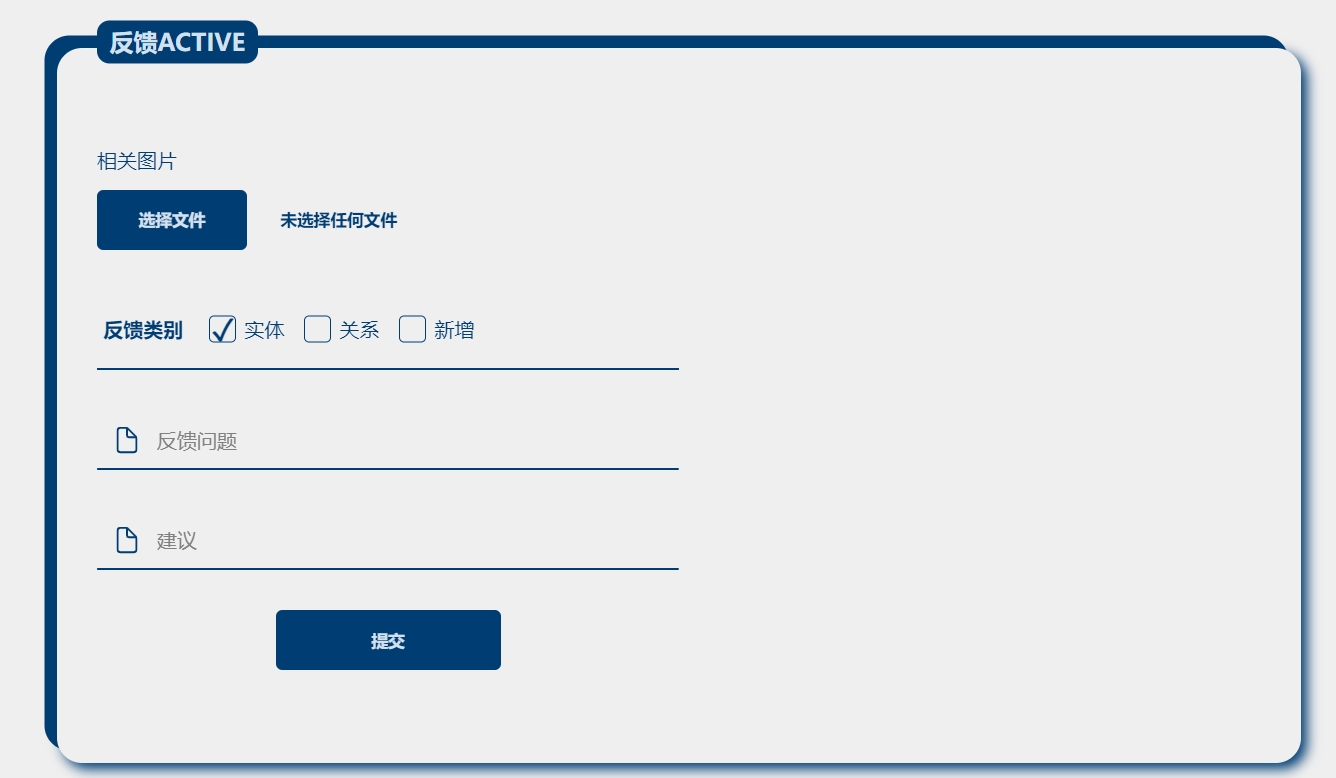
在一分一段表中，用户可以具体查看该省份具体的一分一段

同时，用户输入自己的分数，便可以查询自己的分数在该年份可以上哪些学校



### （九）、反馈

用户可以选择反馈类型，选择反馈上传内容、填写反馈问题、反馈建议等等必要内容



同时，用户提交后，可以在处理栏查看管理人员处理问题的进度



### （十）、用户中心

**个人主页**

个人主页根据用户id以及其隐私设置，返回对应用户的详细信息，隐私设置为保密的则显示未公开

同时，若收藏的隐私为公开，则会显示对应用户的收藏记录，点击记录可跳转到对应的详情界面。



**个人信息**

信息修改有三个方面，

基本信息修改：修改头像图片、用户名、性别、电话等等隐私信息

密码修改：修改密码

隐私修改：修改是否公开的权限，选择不公开则在主页不会显示这些字段



**历史记录**

在历史记录中，所有的历史记录是根据每一天进行分组显示，用户可以根据时间进行查找，然后点击对应的历史记录跳转到对应的详情页面中



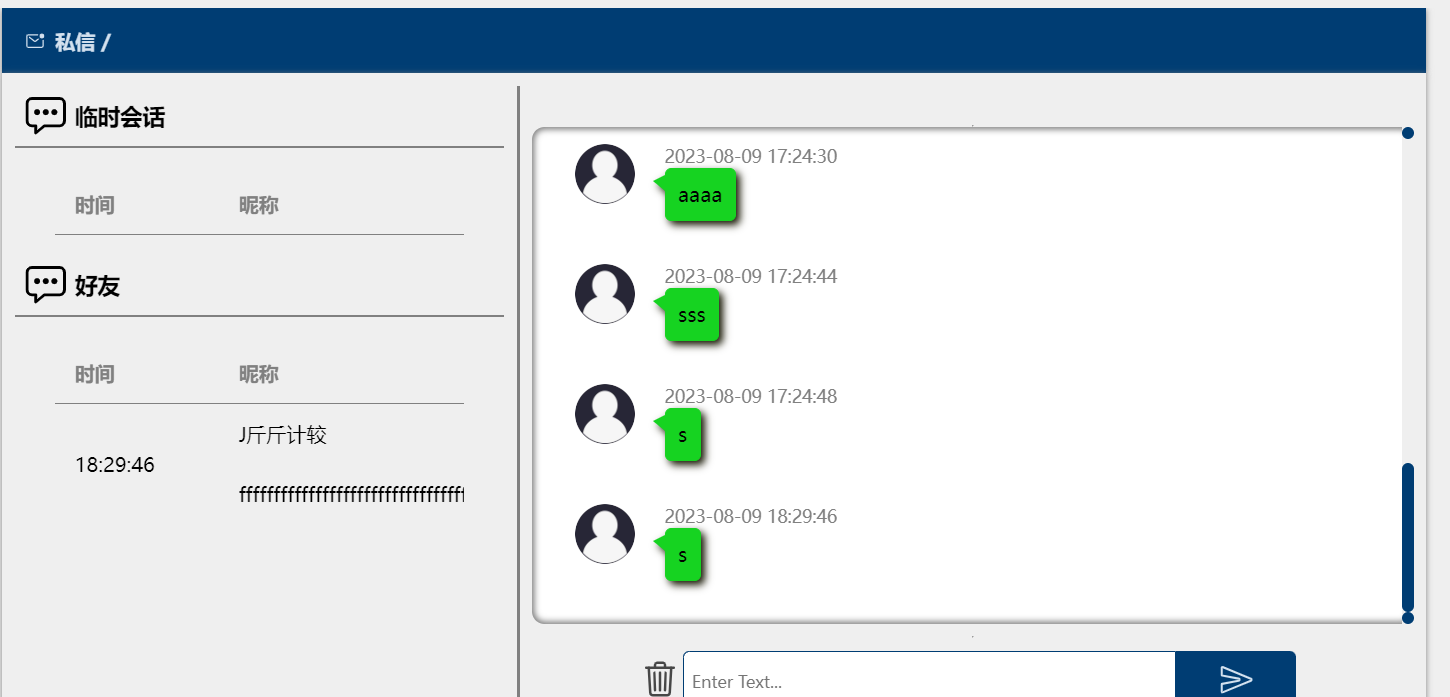
**收藏记录**

在历史记录中，所有的收藏记录是根据收藏类别分组显示，用户可以根据所属类别进行查找，然后点击对应的收藏记录跳转到对应的详情页面中



**私信**

在私信这里，用户可以与之前添加聊天的对象进行聊天



若是用户关注的对象，则会出现在好友列表，若是对方关注的用户，则其会出现在临时会话列表

## 三、功能实现过程及描述

### （一）数据收集

本次需要大学志愿填报服务领域的相关数据

数据源采用：

百度百科：<https://baike.baidu.com>、

掌上高考：<https://gaokao.cn/>、

数据获取：

通过使用requests、request-html、selenium库、scrapy框架对数据进行爬取

爬取过程：

通过分模块爬取，我们可以直接获取对应label下的对应实体，而无需去额外分辨爬取的实体的label

（1）贾泽栋（大学、人（优秀学生、教师）、城市）

1.掌上高考导出大学索引->百度百科爬取大学详情->百度百科爬取对应大学的优秀教师、学生

2.掌上高考导出大学索引->百度百科爬取大学详情->获取所有大学对应的城市，构建城市表与索引

3.掌上高考导出大学索引->爬取对应大学详情

（2）宋希杰（特殊政策、学校政策）

1.掌上高考导出大学索引->百度百科爬取学校政策

2.掌上高考导出大学索引->百度百科爬取特殊政策（211、985、珠峰等等）

（3）黄栩铭（省份、生活条件、一分一段表）

1.掌上高考导出大学索引->百度百科爬取一分一段-》获取省份信息

2.掌上高考导出大学索引->获取对应的食宿介绍

（4）郝浩（专业、专业排名、就业去向等等）

1.掌上高考导出专业详情

2.掌上高考导出大学索引->爬取大学对应专业的详情

数据处理

通过对数据进行查漏补缺，尽最大可能获取最全面的数据，同时对数据进行去重，对两个数据源的数据进行合并，最终形成最后的数据

### （二）数据库搭建

MYSQL表设计

（1）.知识图谱关系数据库设计

考虑大学、专业、学科、分数线、省市等等进行需求设计，构建知识图谱数据库

（2）.用户系统管理数据库设计

考虑要存储用户信息、存储用户历史记录、收藏记录、用户权限、会话、会话消息等需求设计、构建用户系统管理数据库

NEO4J设计

根据知识图谱关系数据库，形成各节点、以及搭建不同节点之间的关系。

详情见数据库文件夹

### （三）前端搭建

资源收集、UI设计、主题设计

利用css变量实现主题切换功能

按照模块设计思路，进行各模块页面搭建

前端权限管理，利用路由导航拦截实现，实现保持登录、免登录功能、同时实时监测用户登陆状态，避免用户越权访问

### （四）后端搭建

各页面接口实现

利用session存储用户id，既实现检测用户登陆状态、避免非用户人员调用接口访问导致信息泄露，又能够迅速搜索用户相关信息，减少向外暴露的参数量，避免注入风险

个性推荐算法实现，采用时间加权以及三阶段协同过滤方法，计算出与该用户喜好最相近的一批实体数据，然后根据实体数据进行个性推荐。

分词提取、词典形成、分词模型搭建

大语言模型调试、提问词调参、接口书写

## 四、运行环境说明

**平台:** windows10、 Intel i7-10875H、2060ti

**浏览器**: Chrome:115.0.5790.110

npm: axios1.4.0、core-js3.8.3、echarts5.4.2、gsap3.12.2、js-pinjin0.2.4、vue3.2.13、vue-router4.0.3、vuex4.0.0

python: django4.2、django-cors-headers4.2.0、django-redis4.3.0、py2neo2021.2.3、mysqlclient2.2.0、requests2.31.0、requests-html0.10.0、selenium4.10.0、pymysql1.1.0、jieba0.42.1、fasttext

（注意mysqlclient用conda安装不会报错）

（fasttext请下载对应的whl进行本地安装）

## 五、测试

### （一）模块功能测试

**个性推荐：**

是否可以实现设计的功能，根据用户的历史记录通过个性推荐算法进行相关内容推荐

**文本识别：**

是否可以根据给定的文本进行实体识别，是否可以点击对应实体显示对应详情内容

**实体查询：**

是否可以根据输入的内容显示对应的关系与图表

**关系查询：**

是否可以根据输入的实体关系查找显示对应的关系图表

**大学专业概览：**

是否可以正常显示所有地区的大学数量，是否可以点击某一地区显示对应地区的大学树状图

**AI问答：**

是否可以根据用户的输入识别实体、关系，然后生成对应的回答

**分数智能查询：**

是否可以根据输入的地区、年份、类别生成对应的折线图、一分一段表、

是否可以根据用户输入的分数对应的推荐相关大学

**用户中心：**

个人主页：是否可以根据用户隐私设置选择性显示对应的信息

信息修改：是否可以修改用户信息、修改密码、修改隐私

历史记录、收藏记录：是否可以正常显示用户记录并点击跳转对应的页面

私信：是否可以正常与对面的好友进行聊天

### （二）项目上线测试

项目部署到服务器后

测试项目整体功能的运行情况（路由跳转是否正常、接口请求是否正常等等）

测试项目各部分的资源加载速度（图片加载速度是否正常、是否需要考虑速度优化）

## 六、部署

### （一）服务器配置

2 vCPU 2 GiB ubuntu **开放**80

2 vCPU 2 GiB centos **开放**7474、7687

### （二）nginx静态资源部署

Ubuntu，安装部署nginx

监听

```

server {

listen 80;

server\_name [ip/域名];

gzip on;

gzip\_types application/xml application/json text/css text/javascript application/javascript;

gzip\_vary on;

gzip\_buffers 32 4k;

gzip\_comp\_level 6;

gzip\_min\_length 100;

location / {

root [path];

index index.html index.htm;

}

location /api/ {

include uwsgi\_params;

uwsgi\_pass 0.0.0.0:8000;

}

```

### （三）uwsgi动态资源部署

安装uwsgi，注意请用conda安装，不会报错

chdir=[path to/CH-Backend/backend]

socket=0.0.0.0:8000

module=AK\_Graph\_Backend.wsgi:application

processes=2

threads=4

pidfile=[path to uwsgi.pid]

vacuum=true

daemonize=[path to uwsgi.log]

### （四）neo4j图关系数据库部署

下载neo4j的tar.gz压缩包

配置java17环境

将neo4j加入环境变量

neo4j-admin 导入dump文件

neo4j start后台挂起服务