

浙江大学

本科实验报告

课程名称: B/S 体系软件设计

姓 名: 彭子帆

学 院: 计算机科学与技术学院

系: 软件工程

专 业: 软件工程

学 号: 3170105860

指导教师: 胡晓军

2020 年 05 月 23 日

浙江大学实验报告

课程名称: B/S 体系软件设计 实验类型: 软件系统开发

实验项目名称: 调查问卷网站

学生姓名: 彭子帆 专业: 软件工程 学号: 3170105860

同组学生姓名: 无 指导老师: 胡晓军

实验地点: 疫情期间于家中 实验日期: 2020 年 3 月 1 日-5 月 23 日

1. 实验目的

任选一种技术实现一个调查问卷的网站

2. 实验要求

需要实现的基本功能如下:

1. 实现用户注册、登录功能, 用户注册时需要填写必要的信息并验证, 如用户名、密码要求在 6 字节以上, email 的格式验证, 并保证用户名和 email 在系统中唯一。
2. 用户登录后可以设计问卷, 一个问卷由标题、问卷说明和多个问卷项目组成, 提供一个界面来动态设计, 问卷项目需要最少需要实现以下内容
 - 单选
 - 多选
 - 文本填写 (支持单行、多行)
 - 数字填写 (支持定义数字类型: 整数、小数)
 - 评分 (这是单选的特殊形式, 界面表示有所不同)

请您对此项目进行评价

☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7

非常不满意

非常满意

- 级联选择 (对于单选的每个选择, 可以动态显示其他一个或多个问卷项目)

1 是否在校

☐ 是

☐ 否

如用户选“是”，就显示一些项目，选“否”就显示另外一些项目

3. 问卷设计完成后生成填写链接，通过分享链接由他人填写，分享时可以设定填写周期和填写方式，填写方式支持以下几种

- 仅限注册用户
- 无需注册，可填写 n 次
- 无需注册，每天可填写 n 次

自行设计如何确定同一人。

4. 保存填写结果，设计一个界面展现填写结果，要求展示界面直观清晰，有一定的统计量（如填报人数、起止时间等），对于数字类型的问卷项目支持汇总计算。

增强功能：

5. 样式适配手机端，能够在手机浏览器/微信等应用内置的浏览器中友好显示，支持额外的问卷项目

- 地理位置

一些可能需要的账号与密码：

阿里云服务器：[47.94.46.115](#)

账号：root

密码：5860@zju

MySQL 数据库：47.94.46.115:3306

用户名：root

密码：123456

Redis 数据库：47.94.46.115:2020

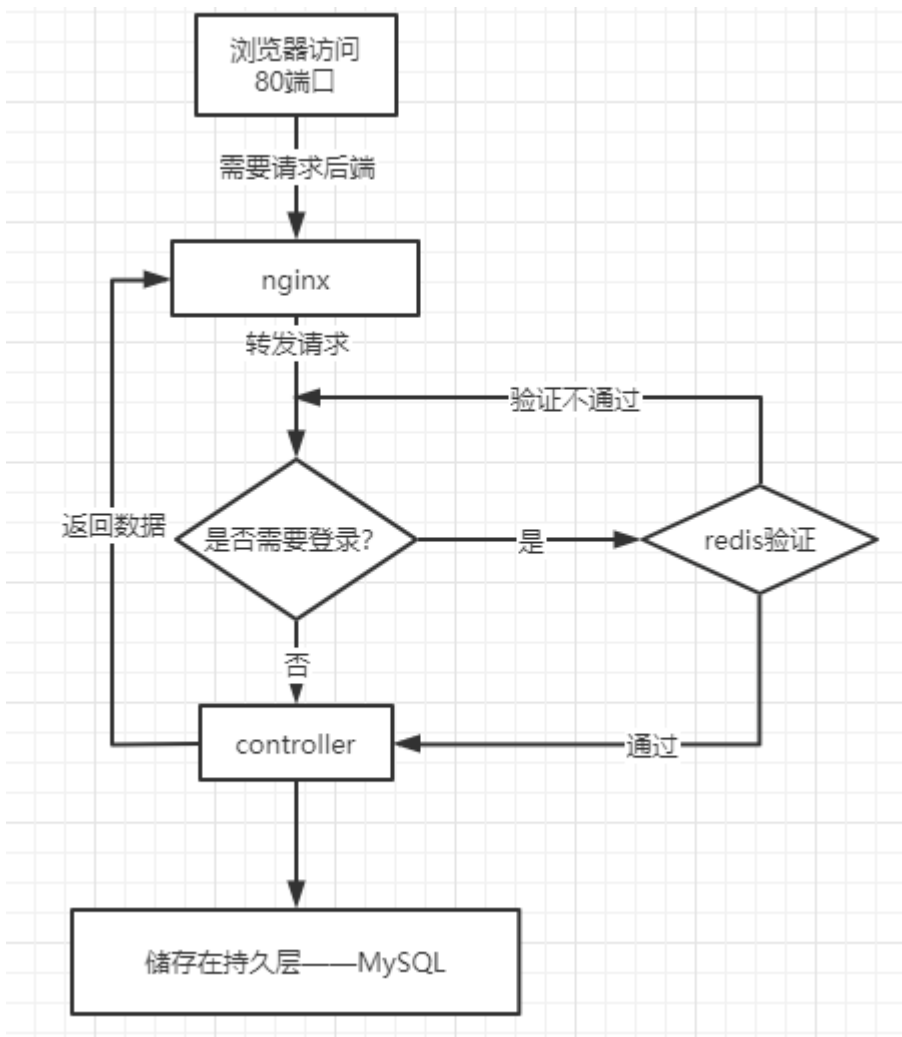
数据库 name：redis

3. 实验过程

整体架构思路：

前后端分离。前端采用 element-ui 以及，vue.js 并通过 nginx 转发请求到后端端口，后端采用 spring boot + mybatis 框架。Redis、Mysql、nginx 以及后端服务都通过 docker 部署在服务器 47.94.46.115 的不同端口上。

整体流程如下图：



3.1 购买并搭建服务器

疫情期间阿里云学生服务器免费领取：

实例ID/名称	标签	监控	可用区	IP地址	状态	网络类型	配置	付费方式	操作
i-2zeir6up2906ad31pzh3			华北 2 可用区 H	47.94.46.115(公)	运行中	专有网络	2 vCPU 4 GiB (I/O优化) ecs.t5-c1m2.large 1Mbps	包年包月 2020年11月4日 23:59 到期	管理 远程连接 升降配 续费 更多

配置信息	升降配	更多▼
CPU: 2核		
内存: 4 GiB		
实例类型: I/O优化		
操作系统: CentOS 7.7 64位		
弹性网卡: eni-2zefxpd65yi9x0p17j99 		
公网IP: 47.94.46.115 		
弹性公网IP: - 		
私有IP: 172.17.49.45 		
辅助私网IP: 管理辅助私网IP		
带宽计费方式: 按固定带宽		
当前使用带宽: 1Mbps		

3.2 Docker 的安装

卸载旧版本:

```
sudo apt-get remove docker docker-engine docker.io containerd runc
```

安装新版本, 由于阿里云自动启用阿里的仓库因此不用配置 ubuntu 的仓库:

```
sudo apt-get install docker-ce docker-ce-cli containerd.io
```

检查安装完毕

```
[root@iZ2zeir6up2906ad3lpzh3Z ~]# docker --version
Docker version 19.03.8, build afacb8b
```

由于镜像源位于国外, 安装其他软件较慢, 配置 docker 镜像加速, 找到阿里云的私人镜像加速:

镜像加速器

加速器

使用加速器可以提升获取Docker官方镜像的速度

加速器地址

<https://xi3mk5ux.mirror.aliyuncs.com>

复制

1. 安装 / 升级 Docker 客户端

推荐安装 1.10.0 以上版本的 Docker 客户端，参考文档 [docker-ce](#)

2. 配置镜像加速器

针对 Docker 客户端版本大于 1.10.0 的用户

您可以通过修改 daemon 配置文件 `/etc/docker/daemon.json` 来使用加速器

```
sudo mkdir -p /etc/docker
```

```
sudo tee /etc/docker/daemon.json <<-'EOF'
```

```
{  
  "registry-mirrors": ["https://xi3mk5ux.mirror.aliyuncs.com"]  
}
```

EOF

```
sudo systemctl daemon-reload
```

```
sudo systemctl restart docker
```

3.3 MySQL 的安装部署

拉取镜像

```
docker pull mysql:latest
```

运行容器

```
docker run -itd --name mysql -p 3306:3306 -e MYSQL_ROOT_PASSWORD=123456 mysql
```

3.4 Redis 的安装部署

拉取 image

```
docker pull redis:latest
```

运行容器

```
docker run -itd --name redis -p 2020:6379 redis:latest myredis
```

3.5 docker compose 的安装与使用

Compose 是用于定义和运行多容器 Docker 应用程序的工具。通过 Compose，您可以使用 YAML 文件来配置应用程序需要的所有服务。然后，使用一个命令，就可以从 YAML 文件配置中创建并启动所有服务。本次实验中只是简单的使用了 docker compose

Compose 的安装：

```
$ sudo curl -L "https://github.com/docker/compose/releases/download/1.24.1/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
$ sudo chmod +x /usr/local/bin/docker-compose
$ sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
```

前端的 docker-compose.yml 文件

(与 nginx.conf 和 vue 的 npm run build 生成的 dist 都同级放在同一文件夹中)

```
version: '3'
services:
  # 服务名称
  nginx:
    # 镜像:版本
    image: nginx:latest
    container_name: frontend
    # 映射容器 80 端口到本地 80 端口
    ports:
      - "80:80"
    # 数据卷 映射本地文件到容器
    volumes:
      # 映射 nginx.conf 文件到容器的/etc/nginx/conf.d 目录并覆盖 default.conf 文件
      - ./nginx.conf:/etc/nginx/conf.d/default.conf
      # 映射 dist 文件夹到容器的/usr/share/nginx/html 文件夹
      - ./dist:/usr/share/nginx/html
    # 覆盖容器启动后默认执行的命令。
    command: /bin/bash -c "nginx -g 'daemon off;'"
```

后端的 docker-compose.yml 文件

```
version: '3'
services:
  # 服务名称
  java:
    # 镜像:版本
    image: openjdk:latest
    container_name: backend
    # 映射容器 8080 端口到本地 8080 端口
    ports:
```

```
- "8080:8080"
volumes:
  - ./usr/share
# 覆盖容器启动后默认执行的命令。
command: /bin/bash -c "java -jar /usr/share/questionnaire-0.1.jar"
```

3.6 Nginx 与前端的安装部署

首先和刚刚相同都进行拉取

```
$ docker pull nginx:latest
```

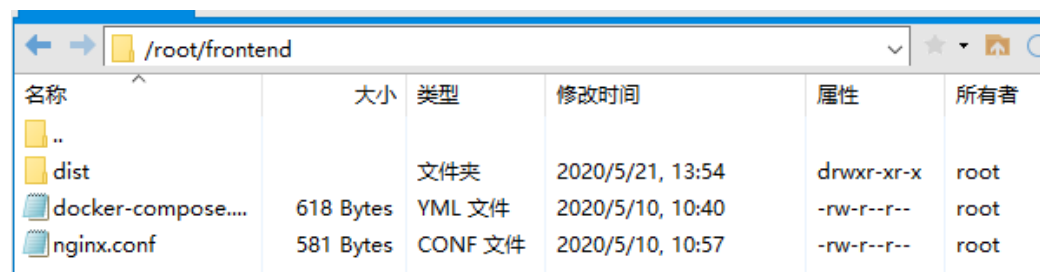
然后到下面图中的 frontend 文件夹中

```
$ docker-compose up -d
```

```
[root@iZ2zeir6up2906ad3lpzh3Z frontend]# docker-compose up -d
Starting frontend ... done
```



名称	大小	类型	修改时间	属性	所有者
..					
backend		文件夹	2020/5/5, 11:16	drwxr-xr-x	root
frontend		文件夹	2020/5/23, 20:25	drwxr-xr-x	root



名称	大小	类型	修改时间	属性	所有者
..					
dist		文件夹	2020/5/21, 13:54	drwxr-xr-x	root
docker-compose....	618 Bytes	YML 文件	2020/5/10, 10:40	-rw-r--r--	root
nginx.conf	581 Bytes	CONF 文件	2020/5/10, 10:57	-rw-r--r--	root

3.7 后端的部署

与前端同理，先要拉取镜像

```
$ docker pull openjdk
```

然后到下面的 backend 文件夹中

```
$ docker-compose up -d
```


/root/backend					
名称	大小	类型	修改时间	属性	所有者
..					
docker-compose....	373 Bytes	YML 文件	2020/5/8, 13:24	-rw-r--r--	root
questionnaire-0.1.j...	35.85MB	Executabl...	2020/5/23, 1:57	-rw-r--r--	root

通过 docker 安装的一些镜像如下:

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
openjdk	latest	db8922a53903	2 weeks ago	497MB
redis	latest	f9b990972689	3 weeks ago	104MB
mysql	latest	a7a67c95e831	3 weeks ago	541MB
nginx	latest	602e111c06b6	4 weeks ago	127MB
eolinker/eolinker-api-management-system	latest	e89dadd3a32c	23 months ago	1.87GB

最后, 通过 docker 部署后的一些容器结果如下:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
815adea63f18	redis	"docker-entrypoint.s..."	13 days ago	Up 13 days	0.0.0.0:2020->6379/tcp	redis
33eca908e507	nginx:latest	"/bin/bash -c 'nginx..."	2 weeks ago	Up 19 hours	0.0.0.0:80->80/tcp	frontend
0be53b410697	openjdk:latest	"/bin/bash -c 'java ..."	2 weeks ago	Up 18 hours	0.0.0.0:8080->8080/tcp	backend
126ad592dd75	mysql	"docker-entrypoint.s..."	2 weeks ago	Up 2 weeks	0.0.0.0:3306->3306/tcp, 33060/tcp	mysql

3.8 接口的设定

通过 spring 的 swagger 插件, 定义在网站: <http://47.94.46.115:8080/swagger-ui.html>

API 问卷调查 1.0.0

[Base URL: 47.94.46.115:8080/]

<http://47.94.46.115:8080/v2/api-docs>

API 问卷调查项目的前后端交互的接口

answer-controller Answer Controller

- POST** /answer/{paperid} 回答问卷
- GET** /result/{paperid} 获取问卷的结果

basic-error-controller Basic Error Controller

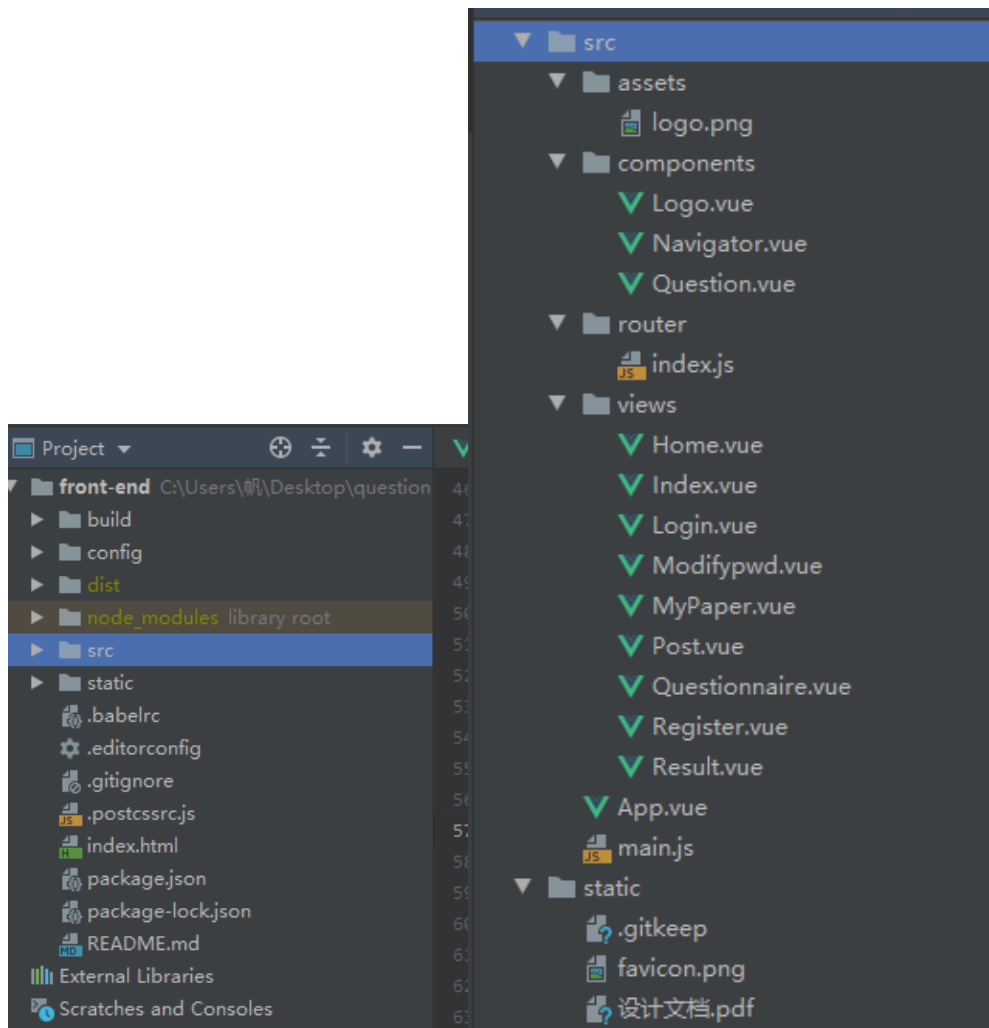
paper-controller Paper Controller

user-controller User Controller

Models

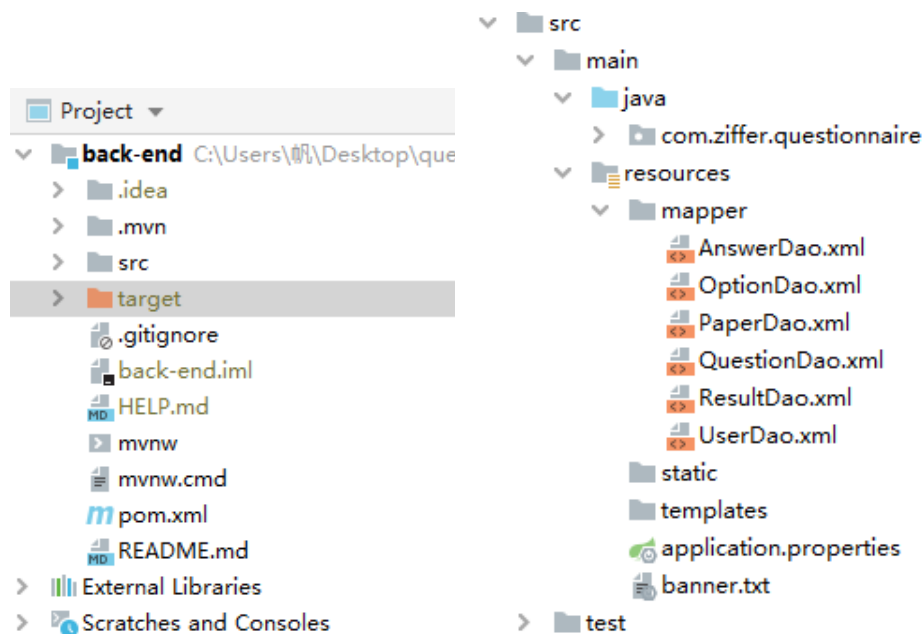
3.9 前端代码的开发

前端采用 vue.js 框架结合 element-ui 进行开发，具体代码在文件夹中：

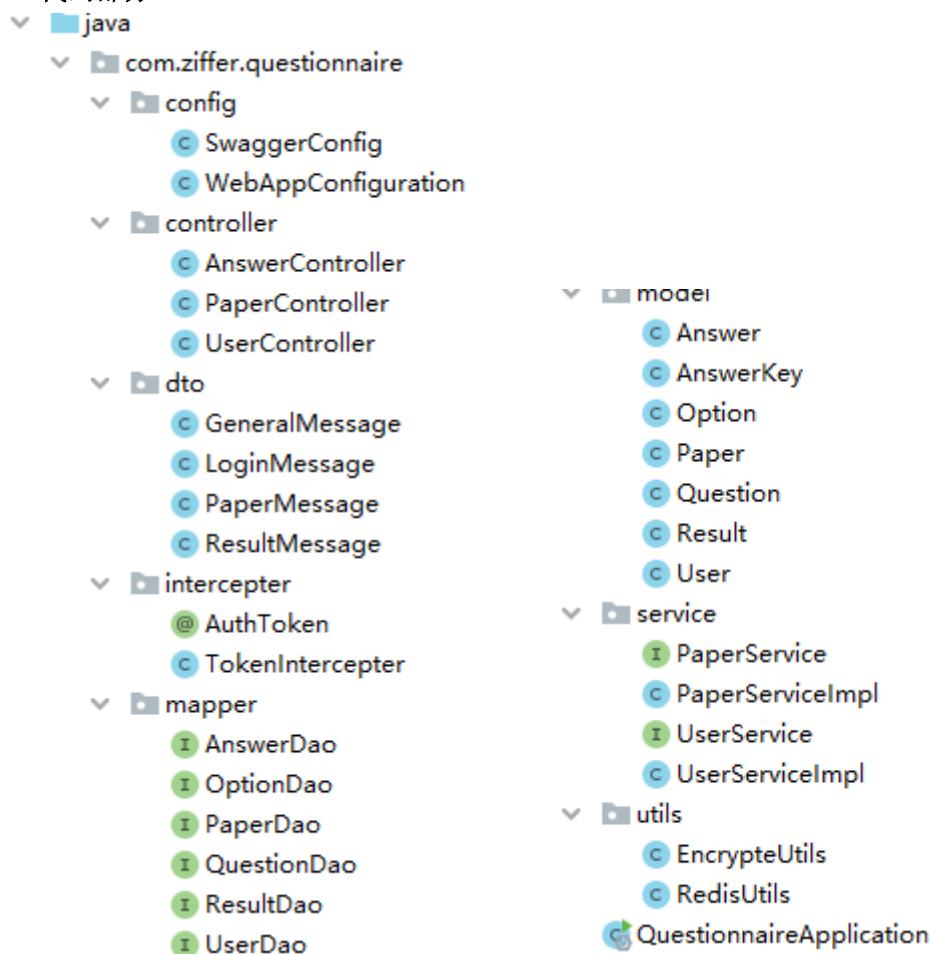


3.10 后端代码的开发

后端采用 Spring Boot+MyBatis 框架并使用了一些插件如 lombok 等进行开发，文件目录如下：



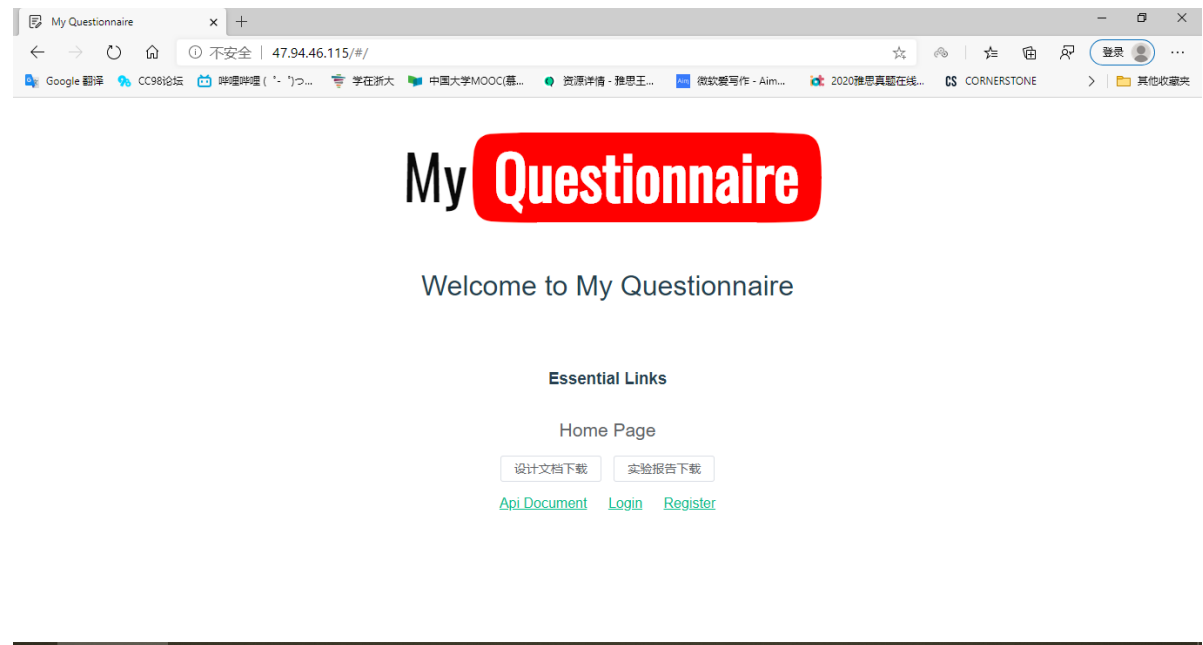
Java 代码部分：



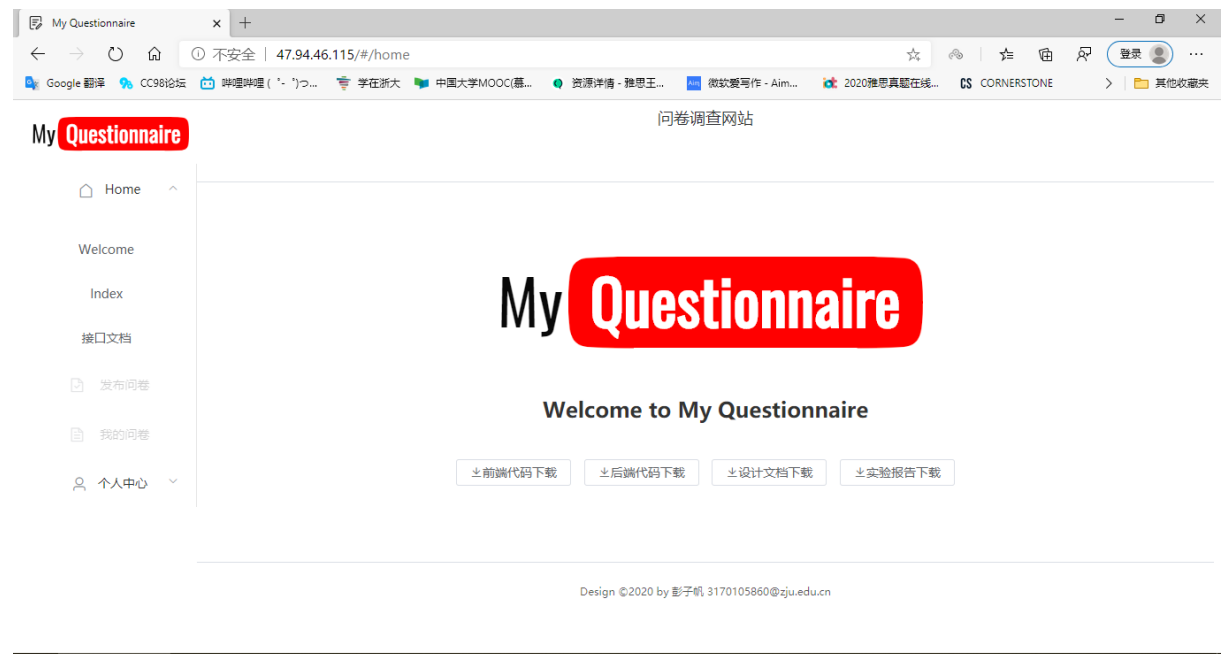
4 实验结果

最后成果的网站如下: <http://47.94.46.115/#/>

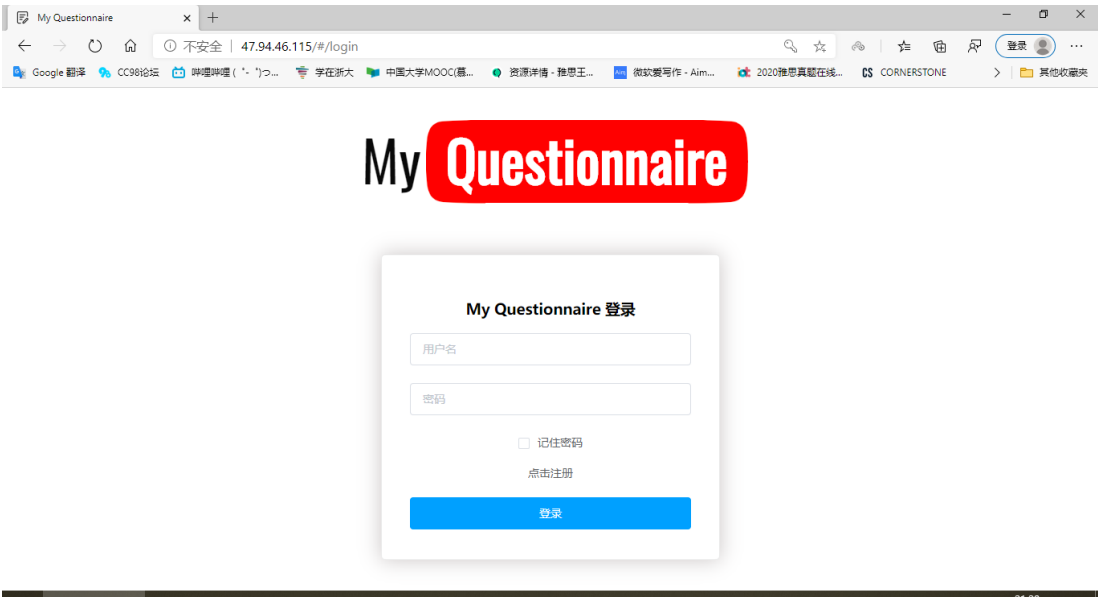
Index 页:



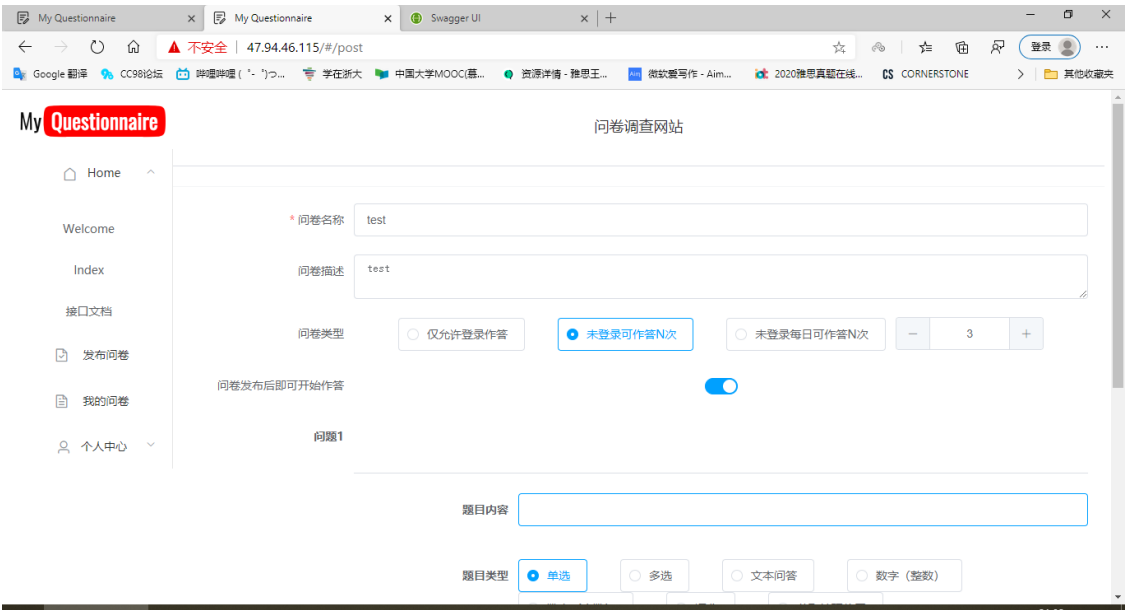
HomePage:



登录、注册与修改密码类似：



发布问卷：



问题1

题目内容

单选

题目类型

单选

多选

文本问答

数字 (整数)

数字 (小数)

评分

获取地理位置

选项1

单选项1

选项2

单选项2

选项3

单选项3

+

添加选项

我的所有问卷：

My Questionnaire

Home

Welcome

Index

接口文档

发布问卷

我的问卷

个人中心

修改密码

问卷调查网站

问卷标题	序号	模式	可填写的次数	状态	描述	链接	操作
test	1	未注册可答N次	3	open	test	http://47.94.46.115/#/ques	查看结果 删除 关闭

回答页面：

test

test

* 多选

☒ 多选选项1 ☒ 多选选项2

* 获取地理位置

浙江省杭州市

* 数字 (小数)

-

0.30

+

* 单选

☒ 单选选项1 ☐ 单选选项2 ☐ 单选选项3

* 评分

★ ★ ★ ☆ ☆

* 文本问答

你好

* 数字 (整数)

-

2

+

立即提交

获取问卷结果界面：

序号	用户名	IP地址	提交时间	单选
1	admin123	60.186.216.234	Sat May 23 13:36:59 GMT 2020	单选选项2
2	admin123	60.186.216.234	Sat May 23 13:36:39 GMT 2020	单选选项1

My Questionnaire 问卷调查网站						
单选	多选	文本问答	数字（整数）	数字（小数）	评分	获取地址
单选选项2	多选选项1;	你好	6	0.4	4.5	浙江會計學院
单选选项1	多选选项1;多选选项2;	你好	2	0.3	3	浙江會計學院
			8	0.7	7.5	