项目：胃炎

涉及：爬虫、开发、数学分析，数据挖掘

1. 界面的开发--->前端
2. 后面逻辑
3. 收集的数据进行数据分析
4. 从网上抓取关于胃炎的文献
5. 对现有的数据进行数据挖掘

模块：

框架：

前端：juqery mobile （jqm）

后端:django

爬虫+自动化测试

数据分析：pandas

数据挖掘

1. 对项目开发前的分析

问卷调查(病人和非病人做）

涉及到的设计维度有熟悉。

胃炎病人身高和体重：肥胖指数

BMI指数算法：体重/身高的平方

调查维度：

1. 姓名 2、性别 3、年龄、4、父母兄弟遗传性问题

5、身高和体重计算BMI 6、是否吸烟 7、是否饮酒8、是否幽门螺杆菌感染9、是否长期生活不规律10、是否大地物理变化导致（气候变化，环境变化）11、是否其他脏器病变导致

1. jqm框架（移动端）

1、需要解决的问题

1. jqm怎么用?
2. Jqm的九宫格怎么去做
3. jqm调查问卷中的一些做法

2、关于版本号的说明

jquery 版本号1.7.1

Jquery mobile 版本号 1.3.2

注意:(简单做一下并发,这个项目不是主要处理的问题)

Jqm 需要jquery,js文件,css文件

1. jqm基本页面制作

(移动端页面不用考虑标签，div)

1. 引入js/css文件
2. jqm把所有的移动端的界面都用data-role=”page”
3. jqm把移动端页面分为顶部、中部和底部.

顶部 data-role=header

中部 data-role=content

底部 data-role=footer

改变显示的样式:data-theme=”a,b,c,d,e”

固定顶部在顶端，底部在底端， data-position=”fixed”

把底部变导航：

data-role=”navbar”中间嵌套ul li

加图标 data-icon=”图标名”

所有jqm共有框架:

<!DOCTYPE html>  
<html lang**="en"**>  
<head>  
 <meta charset**="UTF-8"**>  
 <title>**胃炎检测**</title>  
 <link rel**="stylesheet"** href**="css/jquery.mobile.min.css"**/>  
 <script src**="js/jquery-1.7.1.min.js"**></script>  
 <script src**="js/jquery.mobile.min.js"**></script>  
</head>  
<body>  
 <div data-role**="page"**>  
 <div data-role**="header"** data-theme**="b"** data-position**="fixed"**>  
 <a href**="#"**>**返回**</a>  
 <h1>胃炎**分析**</h1>  
 </div>  
 <div data-role**="content"**>  
  
 </div>  
 <div data-role**="footer"** data-theme**="b"** data-position**="fixed"**>  
 <div data-role**="navbar"**>  
 <ul >  
 <li><a href**="#"** data-icon**="home"**>**首页**</a></li>  
 <li><a href**="#"** data-icon**="gear"**>**设置**</a></li>  
 <li><a href**="#"** data-icon**="star"**>**推荐**</a></li>  
 <li><a href**="#"** data-icon**="grid"**>**收藏**</a></li>  
 </ul>  
 </div>  
 </div>  
 </div>  
</body>  
</html>

1. 做内容可以只做content部分，其它部分复制粘贴即可。
2. jqm实现九宫格

“ui-grid”里面嵌套”ui-block”

Class=”ui-grid-a”一行两个元素

Class=”ui-grid-b”一行三个元素

Class=”ui-grid-c”一行四个元素

每一行都按ui-block-a ui-block-b ui-block-c

遇到ui-block-a换行

要求一行三个图标，

1. 调查问卷

结合九宫格技术进地布局

两个单选按钮组合在一起,data-role=”controlgroup”

水平放在一起 data-role=”horizontal”

表单中的任何一个元素，取一个name名字.

1. django前后端分离
2. 写接口的目的

后台存储,返回json数据。

1. 版本：

　　　django 3.0.3

mysql 5.7.25

djangorestframework 3.10.3

1. 创建django项目

django-admin startproject myGastritis

1. 创建应用:django-admin startapp myRadang
2. 创建数据库

**gastricism**

create database myRadang default charset=utf8;

1. restful接口，安装djangorestframework
2. 第一步：settings.py设置
3. INSTALLED\_APP　(应用,rest\_framework)
4. DATABASES　（数据库相关配置）

'default': {  
 'ENGINE': 'django.db.backends.mysql',  
 'HOST':'localhost',  
 'PORT':3306,  
 'USER':'root',  
 'PASSWORD':'admin',  
 'NAME': 'gastricism',  
}

1. 静态

STATICFILES\_DIRS=[  
 os.path.join(BASE\_DIR,**"static"**)  
]

如果无法识别static路径，那么可以在urls.py文件中设置static路径来匹配文件

*#如果setting文件中debug=False，那么就遍历整个static文件来匹配合适的静态文件***if** settings.DEBUG==**False**:  
 urlpatterns +=[  
 url(**r'^static/(?P<path>.\*)$'**,static.serve,  
 {**'document\_root'**:settings.STATIC\_L},name=**'static'**)  
 ]

1. 模板

TEMPLATES=

DIRS=[os.path.join(BASE\_DIR,”templates”)]

1. 设计models

定义RadangInfo类,继承于models.Model

username=models.CharField(max\_length=20)  
sex=models.BooleanField(default=False)  
cardid=models.CharField(max\_length=18)  
tall=models.DecimalField(max\_digits=6,decimal\_places=2)  
weight=models.DecimalField(max\_digits=6,decimal\_places=2)  
smoke=models.BooleanField(default=True)  
drink=models.BooleanField(default=True)  
ill=models.BooleanField(default=False)

HP\_infection=models.BooleanField(default=**False**) *#是否幽门螺杆菌感染*irregular\_life=models.BooleanField(default=**False**) *#是否长期不规律生活*environment\_change=models.BooleanField(default=**False**) *#是否大地物理变化导致（气候变化，环境变化）*Organ\_change=models.BooleanField(default=**False**) #*是否其他脏器病变导致*

９、同步数据库

Python manage.py makemigrations

Python manage.py migrate

如果有报错，需要安装pymysql

放在项目下的\_\_init\_\_.py文件中

import pymysql  
pymysql.install\_as\_MySQLdb()

但是如果使用得mysqlclient，那么可以注释上面的命令

10，引入rest\_framework目的,序列化

定义序列化的类，类继承于serializers.ModelSerializer

类中指明序列化的模型，指明返回前端的json数据输出维度有哪些.

有元类来指明上述内容

**class** MyRadangSerializer(serializers.ModelSerializer):  
 *'''  
 要序列的是哪个models,json里面显示是哪些字段  
 \_\_all\_\_所有字段  
 写元类  
 '''* **class** Meta:  
 model = RadangInfo  
 fields = **"\_\_all\_\_"**

**11，后台存储，前端返回json数据**

**view中的方法变成类的形式.类继承APIView**

**我们定义了一个RandangView，继承于APIView**

**提供了两个方法：get post(get与post区别)**

**请求-------->响应**

1、用radang变量存储MyRandangSerializer序列化的模型类，模型类中的参数就是接收的数据  
2、用radang.is\_valid()判断一下数据是否有效  
3、用radang.save()完成数据的存储  
4、返回Response,这是Views函数要求.  
5、这里要判断radang是否有效，else返回Response的错误  
'''  
radang=MyRadangSerializer(data=request.data)  
*#如果数据是有效的***if** radang.is\_valid():  
 *#保存数据* radang.save()  
 *#返回数据* **return** Response(radang.data)  
**else**:  
 **return** Response(radang.errors)

**12、写路由规则，通过地址找到方法的映射。**

**urls.py中写字相应的路由规则。**

前端做表单提交，需要有form

属性: action=”/blood/” method=”post”

(1)用form表单提交,得到rest\_framework原生的apiview界

(2)手写form表单提交

/\*jquery可以用$("#")提取id元素,发生的click点击事件\*/  
$("#btn").click(function(){  
 /\*#form是form的id名，取表单数据，序列化 serilizer()\*/  
 data=$("#myform").serialize()  
 /\*用post方法,地址要用”/blood/“\*/  
 $.post('',data,function(res){  
 *console*.log(res)  
 })  
})

1. 爬取外文胃炎文章（正在创建中）
2. 数据分析胃炎数据（正在创建中）
3. 数据挖掘（正在创建中）