# 个人疫情防控系统报告

Catalog

**[个人疫情防控系统报告 1](#_Toc28881)**

**[1.系统架构 1](#_Toc13392)**

**[2.实验内容: 1](#_Toc15373)**

[3.架构设计: 2](#_Toc24408)

**[4.设计细节 2](#_Toc29286)**

[4.1登录 2](#_Toc32029)

[4.2 用户注册 3](#_Toc8511)

[4.3用户的疫情信息 5](#_Toc30775)

### 1.系统架构

本系统使用web为操作界面,使用bs架构进行服务的搭建

框架 :django

数据库:mysql

操作系统 :windows

pip3 install django

pip3 install mysqlclinet

### 2.实验内容:

（1）登录：用户分为管理员和用户两类角色，通过系统首页输入正确的用户名和密码后登录系统。

（2）用户信息管理：用户信息的增加、删除、修改和查询。其中个人信息应包括个人身份证号、联系电话、现居住地等相关信息；管理员信息可以只体现账号和密码：

（3）核酸检测信息管理：用户可依据不同条件查询、显示核酸检测的信息；管理员可对核酸信息进行维护；

（4）疫苗接种信息管理：用户可依据不同条件查询、显示疫苗的信息；管理员可对疫苗接种信息进行维护；

（5）家庭成员信息管理：可对家庭成员信息（姓名、年龄、与本人关系、联系电话、身份证号等）进行增加、删除、修改和查询；

（6）疫情实时情况管理：用户可实时填报个人疫情实时情况，如现居住地被定为封控区，要求足不出户，解封时间为＊天，以及具体困难等信息；管理员可查询汇总显示当前所有用户的疫情实时信息，并可进行汇总统计显示。

设计思路:

实现了个人疫情的管理系统和疫情中信息的实现,

界面简介和,我们使用web开发可交互的界面,所有的请求都以post的请求进行发送数据到

Django的后端, 用mysqlclinet连接数据库,将数据库的信息载入到django中

## 3.架构设计:

使用 mvc 架构

从view 架构到 module 层到controller

视图层 ->服务层 ->持久层

Web ->django ->mysql

数据库 设计:

class UserInfo(models.Model):

name = models.CharField(max\_length=32)

password = models.CharField(max\_length=64)

age = models.IntegerField(default=2)

class KeSuanInfo(models.Model):

name = models.CharField(max\_length=32)

address = models.CharField(max\_length=32)

age = models.IntegerField(default=2)

number = models.CharField(max\_length=64)

jiating = models.CharField(max\_length=64)

vaccine = models.CharField(max\_length=64)

kesuan= models.CharField(max\_length=64)

jinzhu = models.CharField(max\_length=64)

date = models.DateField(auto\_now\_add=True)

#### 4.设计细节

##### 4.1登录

从系统设计的url的请求路劲

# path('admin/', admin.site.urls),

path('denglu', views.denglu),

path('index/', views.index),

path('user/', views.user\_list),

path('orm/', views.orm),

path('login/', views.loginHello),

# path('ksuan/',views.kesuan)

path('register/', views.register),

path('info/', views.info),

path('update/', views.update),

path('register1/', views.register1),

path('select/',views.select),

path('jiating/',views.jiating)

将用户的登录的函数在请求该路劲的时候被调用

例如login函数:

def loginHello(request):

if request.method == "GET":

name = request.GET.get('username')

password = request.GET.get('password')

print(name)

print(password)

users = UserInfo.objects.filter(name=name)

user = users.first()

print(user.name)

print(user.password)

if name == user.name and password == user.password:

print("登录成功")

# data\_list =KeSuanInfo.objects.filter(name=name).all()

data\_list =KeSuanInfo.objects.all()

return render(request, "user\_list.html",{"data\_list":data\_list})

此函数中,通过网络的get请求 进行判断数据库中密码是否正确,如果正确就会跳转到相应成功登录的界面.

4.2 用户注册

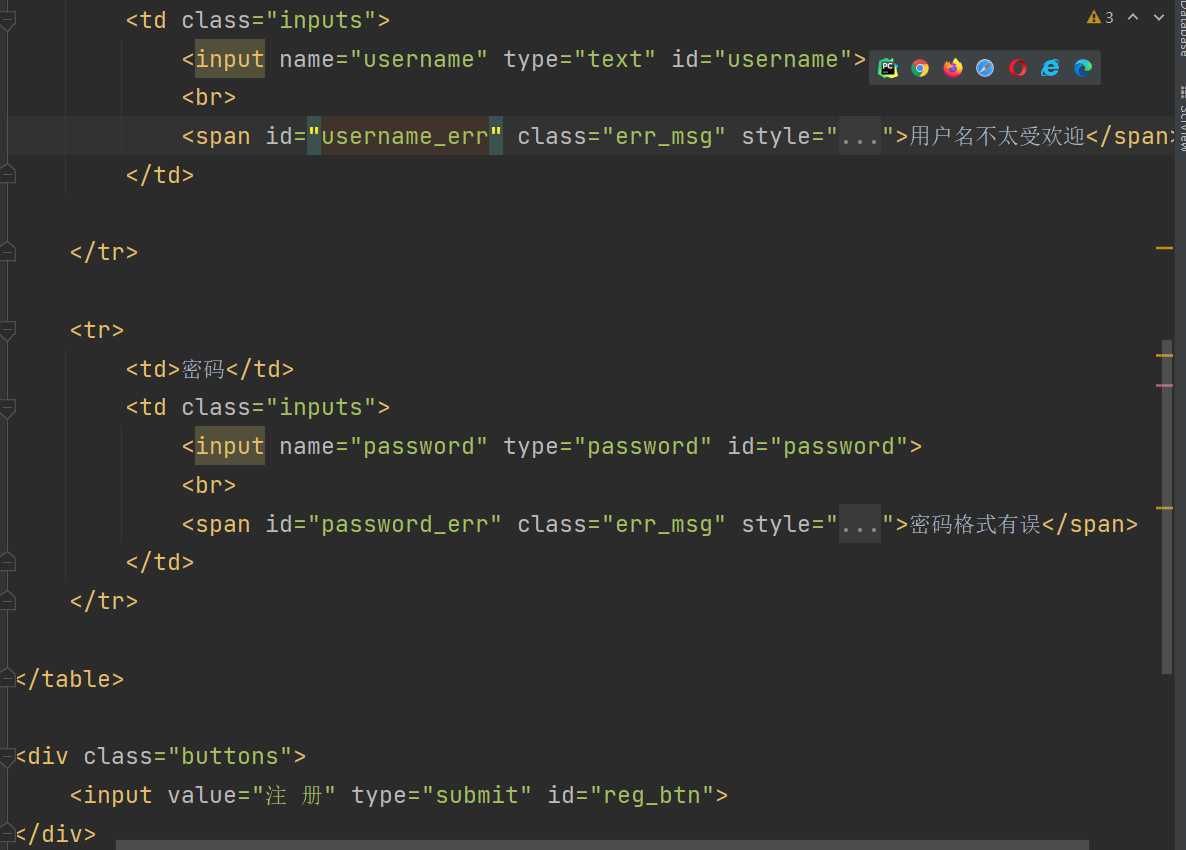
Register 的url路劲

def register(request):

return render(request, "register.html")

当我们选择 register 的登录页面之后,返回登录页面至浏览器

在用户输入信息之后 ,点击提交时,将信息发送至后端,保存进数据库



<form id="reg-form" action="/register1" method="get">

Action地址就是提交处理的系统路劲地址

/register1

def register1(request):

if request.method == "GET":

name = request.GET.get("username")

password = request.GET.get("password")

# address = request.POST.get("adress")

# age = request.POST.get("age")

# number = request.POST.get("number")

# jiating = request.POST.get("jiating")

# vaccine = request.POST.get("vaccine")

# kesuan = request.POST.get("kesuan")

# jinzhu = request.POST.get("jinzhu")

UserInfo.objects.create(name=name,password=password)

# data = {

# 'flag': 1,

# 'code': '200',

# 'message': '获取成功!'

# }

return render(request, "login.html")

之后就返回到登录页面进行登录

##### 4.3用户核酸的疫情信息

将用户的所有信息都写入到web中

if request.method == "GET":

name = request.GET.get('username')

password = request.GET.get('password')

print(name)

print(password)

users = UserInfo.objects.filter(name=name)

user = users.first()

print(user.name)

print(user.password)

if name == user.name and password == user.password:

print("登录成功")

# data\_list =KeSuanInfo.objects.filter(name=name).all()

data\_list =KeSuanInfo.objects.all()

return render(request, "user\_list.html",{"data\_list":data\_list})

##### **把所有用户的所有信息都进行展示**

##### 4.4用户核酸信息的删除和修改

KeSuanInfo.objects.filter(id=id).update(kesuan='ok', is\_active=True)

KeSuanInfo.objects.filter(id=id).delete()

调用内部自带的update和delect方法

##### 4.5 信息关联

用户信息中存在 核酸信息,将这两个信息进行关联,之后可以进行查询

##### 4.6 家庭信息查询

在web端制作好查询按钮

返回web端的家庭查询页面

def jiating(request):

return render(request, "jiating.html")

<table class="table">

<caption>查询用户家庭的疫苗检测情况</caption>

<form action="/select" method="get" class="form-horizontal" role="form">

<div class="form-group">

<label for="firstname" class="col-sm-2 control-label">右边是输入查询家庭个人防疫信息</label>

<div class="col-sm-10">

<input type="text" class="form-control" id="username"

name="username"

placeholder="请输入名字">

</div>

</div>

<div class="form-group">

<div class="col-sm-offset-2 col-sm-10">

<div class="checkbox">

<label>

<input type="checkbox"> 请记住我

</label>

</div>

</div>

</div>

<div class="form-group">

<div class="col-sm-offset-2 col-sm-10">

<input type="submit" value="搜索" class="btn btn-default">搜索该用户</input>

</div>

</div>

##### 4.7家庭信息的删除和修改

在信息和渲染还有删除,提交不同的路劲,在不同的路径下进行删除和修改

KeSuanInfo.objects.filter(id=id).update(kesuan='ok', is\_active=True)

KeSuanInfo.objects.filter(id=id).delete()

## 心得和体会

总结 :关于计算机的程序设计,对于对语言的熟练运用,对python语言语法的熟悉,将数据一个个保存起来,最重要的设计在于,如何把其中算法字符与数字分离开来,对程序进行细致的设计,分析每一个细节

体会:学好框架,熟练的使用框架进行各种数据的变换格式,有利于我们对程序的掌握.

主动上机，多多实践，提高动手能力。通过上机实践来验证所学的理论，通过在机器上查找错误来巩固概念，通过提高上机能力来促进理论学习，开阔编程思路，提高反映速度，提高分析问题解决问题的能力，锻炼心理素质，提高心理承受能力。