**Source Code**

**url.py:**

from django.contrib import admin

from django.urls import path

from django.conf.urls.static import static

from django.conf import settings

from Doctor import views as Doctor

from admins import views as admins

from forensic import views as frnsc

urlpatterns = [

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

path('index/', Doctor.index, name='index'),

path('doctorlogin/',Doctor.doctorlogin,name='doctorlogin'),

path('doctorregister/',Doctor.doctorregister,name='doctorregister'),

path('doctorlogincheck/',Doctor.doctorlogincheck,name='doctorlogincheck'),

path('bodydetails1/',Doctor.bodydetails1,name='bodydetails1'),

path('doctorreport/',Doctor.doctorreport,name='doctorreport'),

path('doctorreport1/',Doctor.doctorreport1,name='doctorreport1'),

path('doctorfinalreport/',Doctor.doctorfinalreport,name='doctorfinalreport'),

path('frnsclogin/', frnsc.frnsclogin, name='frnsclogin'),

path('frnscloginentered/', frnsc.frnscloginentered, name='frnscloginentered'),

path('bodydetails/', frnsc.bodydetails, name='bodydetails'),

path('forensicreport/', frnsc.forensicreport, name='forensicreport'),

path('frnscreport1/', frnsc.frnscreport1, name='frnscreport1'),

path('admin1/', admins.adminlogin, name='admin1'),

path('adminloginentered/', admins.adminloginentered, name='adminloginentered'),

path('doctordetails/', admins.doctordetails, name='doctordetails'),

path('activatedoctor/',admins.activatedoctor,name='activatedoctor'),

path('finalreport/',admins.finalreport,name='finalreport'),

path('adminbodydetails/',admins.adminbodydetails,name='adminbodydetails'),

path('logout/',admins.logout,name='logout'),

]

if settings.DEBUG:

urlpatterns += static(settings.MEDIA\_URL,document\_root=settings.MEDIA\_ROOT)

**views.py:**

from django.shortcuts import render

from django.contrib import messages

from django.http import HttpResponse

from Doctor.models import doctorModel,doctorreportmodel

from Doctor.forms import doctorForm

from forensic.models import \*

from forensic.forms import \*

import speech\_recognition as sr

import pyttsx3

def index(request):

return render(request,'index.html')

def doctorlogin(request):

return render(request,'doctor/doctorlogin.html')

def doctorregister(request):

if request.method == 'POST':

form1 = doctorForm(request.POST)

if form1.is\_valid():

form1.save()

print("succesfully saved the data")

return render(request, 'doctor/doctorlogin.html')

# return HttpResponse("registreration succesfully completed")

else:

print("form not valied")

return HttpResponse("form not valied")

else:

form = doctorForm()

return render(request, "doctor/doctorregister.html", {"form": form})

def doctorlogincheck(request):

if request.method == 'POST':

sname = request.POST.get('email')

print(sname)

spasswd = request.POST.get('upasswd')

print(spasswd)

try:

check = doctorModel.objects.get(email=sname,passwd=spasswd)

# print('usid',usid,'pswd',pswd)

print(check)

request.session['name'] = check.name

print("name",check.name)

status = check.status

print('status',status)

if status == "Activated":

request.session['email'] = check.email

return render(request, 'doctor/doctorpage.html')

else:

messages.success(request, 'doctor is not activated')

return render(request, 'doctor/doctorlogin.html')

except Exception as e:

print('Exception is ',str(e))

pass

messages.success(request,'Invalid name and password')

return render(request,'doctor/doctorlogin.html')

def bodydetails1(request):

qs=bodydetails.objects.all()

return render(request,'doctor/bodydetails1.html',{"object":qs})

def doctorreport(request):

return render(request,'doctor/doctorreport.html')

def doctorreport1(request):

bid=request.POST.get('t1')

qs=bodydetails.objects.filter(bid=bid)

print("qs:",qs)

return render(request,'doctor/doctorreport1.html',{"object":qs})

def doctorfinalreport(request):

r = sr.Recognizer()

print("r:", r)

# Function to convert text to

# speech

def SpeakText(command):

# Initialize the engine

engine = pyttsx3.init()

print("engine:", engine)

engine.say(command)

print("hello:", engine.say(command))

engine.runAndWait()

print(engine.runAndWait())

# Loop infinitely for user to

# speak

SpeakText("report submitted")

if request.method == 'POST':

name1=request.POST.get('t1')

print("name",name1)

bid=request.POST.get('bid')

age=request.POST.get('age')

gender=request.POST.get('gender')

ldh=request.POST.get('ldh')

print("ldh",type(ldh))

ast=request.POST.get('ast')

print("ast",type(ast))

triglycerides=request.POST.get('triglycerides')

phlevel=request.POST.get('phlevel')

rpt=int(ast)+int(triglycerides)+int(ldh)

print("rpt:",type(rpt))

phlevel=int(phlevel)

if rpt<=500 and phlevel==7:

pmi='24hrs'

elif rpt>=550 and phlevel<7 and phlevel>=5:

pmi='48hrs'

elif rpt>=550 and phlevel<5 and phlevel>=4:

pmi='72hrs'

else:

pmi='more than 72hrs'

pmi=str(pmi)

print(type(pmi))

ldh=str(ldh)

ast=str(ast)

triglycerides=str(triglycerides)

doctorreportmodel.objects.create(name=name1,bid=bid,age=age,gender=gender,ldh=ldh,ast=ast,triglycerides=triglycerides, phlevel=phlevel,pmi=pmi)

return render(request, 'doctor/doctorreport1.html')

**models.py:**

from django.db import models

# Create your models here.

class doctorModel(models.Model):

name = models.CharField(max\_length=50)

email = models.EmailField()

passwd = models.CharField(max\_length=40)

cwpasswd = models.CharField(max\_length=40)

mobileno = models.CharField(max\_length=50, default="", editable=True)

specialization = models.CharField(max\_length=50, default="", editable=True)

status = models.CharField(max\_length=40, default="", editable=True)

def \_\_str\_\_(self):

return self.email

class Meta:

db\_table='doctorregister'

class doctorreportmodel(models.Model):

name = models.CharField(max\_length=50)

bid = models.CharField(max\_length=40, default="",editable=True,unique=True)

age = models.CharField(max\_length=40)

gender = models.CharField(max\_length=40)

ldh = models.CharField(max\_length=40)

ast = models.CharField(max\_length=40)

triglycerides = models.CharField(max\_length=40)

phlevel = models.CharField(max\_length=40)

pmi = models.CharField(max\_length=40)

def \_\_str\_\_(self):

return self.bid

class Meta:

db\_table='doctorreport'

**forms.py:**

from django import forms

from Doctor.models import \*

from django.core import validators

class doctorForm(forms.ModelForm):

name = forms.CharField(widget=forms.TextInput(), required=True, max\_length=100,)

passwd = forms.CharField(widget=forms.PasswordInput(), required=True, max\_length=100)

cwpasswd = forms.CharField(widget=forms.PasswordInput(), required=True, max\_length=100)

email = forms.CharField(widget=forms.TextInput(),required=True)

mobileno= forms.CharField(widget=forms.TextInput(), required=True, max\_length=10,validators=[validators.MaxLengthValidator(10),validators.MinLengthValidator(10)])

specialization= forms.CharField(widget=forms.TextInput(), required=True, max\_length=100)

status = forms.CharField(widget=forms.HiddenInput(), initial='waiting', max\_length=100)

def \_\_str\_\_(self):

return self.email

class Meta:

model=doctorModel

fields=['name','passwd','cwpasswd','email','mobileno','specialization','status']

**doctorlogin.html:**

{% extends 'base.html' %}

{% load static %}

{% block contents %}

<section id="intro" class="clearfix">

<div class="container d-flex h-100">

<div class="row justify-content-center align-self-center">

<div class="col-md-6 intro-info order-md-first order-last">

<form action="/doctorlogincheck/" method="POST">

{% csrf\_token %}<center>

<table><tr><td style="color:blue">

<!-- <input type="text" name="uname" placeholder="student Login" ><br/><br/>-->

<input type="email" name="email" placeholder="Email"><br/>

<input type="password" name="upasswd" placeholder="Password" ><br/><br/>

</td></tr></table> <input type="submit" style="color:#FF4500" value="submit"><br/>

<br><br>New Doctor Register Here ?<a href="{% url 'doctorregister' %}" style="color:blue">Click Here</a>

</center> </form> </h2>

{% if messages %}

{% for message in messages %}

<font color='darkred'> {{ message }}</font>

{% endfor %}

{% endif %}

<!--<div>

<a href="#about" class="btn-get-started scrollto">Get Started</a>

</div>-->

</div>

<div class="col-md-6 intro-img order-md-last order-first">

<img src="{% static 'img/intro-img.svg' %}" alt="" class="img-fluid"></br></br>

</div>

</div>

</div>

</section><!-- #intro -->

{% endblock %}