**Source Code**

**url.py:**

from django.contrib import admin

from django.urls import path

from alzmierdisease import views as alzmr

from patient import views as patient

from doctor import views as doctor

urlpatterns = [

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

path('index/',alzmr.index, name="index"),

path('base/', alzmr.base, name="base"),

path('admin1/', alzmr.adminlogin, name="admin1"),

path('adminloginaction/', alzmr.adminloginaction, name="adminloginaction"),

path('activatepatient/',alzmr.activatepatient,name='activatepatient'),

path('testing/', alzmr.testing, name="testing"),

path('svm/', alzmr.svm, name="svm"),

path('svm1/', alzmr.svm1, name="svm1"),

path('logout/', alzmr.logout, name="logout"),

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

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

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

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

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

path('patientdata/',doctor.doctorviewpatientdata,name='patientdata'),

path('addtreatment/',doctor.addtreatment,name='addtreatment'),

path('report/',doctor.report,name='report'),

path('patientlogin/', patient.patientlogin, name='patientlogin'),

path('patientregister/', patient.patientregister, name='patientregister'),

path('patientlogincheck/',patient.patientlogincheck,name='patientlogincheck'),

path('patientdetails/', patient.patientdetails, name='patientdetails'),

path('symptoms/', patient.patientsymptoms, name='symptoms'),

path('patntsymptms/', patient.patntsymptms, name='patntsymptms'),

if settings.DEBUG:

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

**views.py:**

from django.shortcuts import render

from django.http import HttpResponse

from django.contrib import messages

from docutils.nodes import inline

from patient.models import \*

from patient.forms import \*

from sklearn.model\_selection import train\_test\_split

from sklearn.svm import SVC

from sklearn.metrics import classification\_report, confusion\_matrix

import pandas as pd

from django\_pandas.io import read\_frame

from sklearn.naive\_bayes import GaussianNB

from sklearn import metrics

from sklearn import datasets

#svm packages

import cv2

import numpy as np

import pandas as pd

import os

import matplotlib.pyplot as plt

%matplotlib inline

from sklearn.preprocessing import MinMaxScaler

from datetime import datetime

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

import numpy as np

from sklearn.experimental import enable\_iterative\_imputer

from sklearn.impute import IterativeImputer

from sklearn import preprocessing

from sklearn.model\_selection import train\_test\_split, cross\_validate, KFold, GridSearchCV

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import make\_scorer, accuracy\_score, precision\_score, recall\_score, f1\_score, confusion\_matrix, \

classification\_report

from sklearn.neighbors import KNeighborsClassifier

from helpers.my\_one\_hot\_encoder import MyOneHotEncoder

def index(request):

return render(request,'index.html')

def activatepatient(request):

if request.method == 'GET':

uname = request.GET.get('pid')

print(uname)

status = 'Activated'

print("pid=", uname, "status=", status)

patientModel.objects.filter(id=uname).update(status=status)

qs = patientModel.objects.all()

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

def base(request):

return render(request, "base.html")

def adminlogin(request):

return render(request,"admin/adminlogin.html")

def adminloginaction(request):

if request.method == "POST":

if request.method == "POST":

login = request.POST.get('username')

print(login)

pswd = request.POST.get('password')

if login == 'admin' and pswd == 'admin':

return render(request,'admin/adminhome.html')

else:

messages.success(request, 'Invalid user id and password')

#messages.success(request, 'Invalid user id and password')

return render(request,'admin/adminlogin.html')

def logout(request):

return render(request,'index.html')

**models.py:**

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'

**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 %}