**SOURCE CODE**

**Urls.py**

from django.contrib import admin

from django.conf.urls import url,include

from .views import home,trnxmngrloginaction,trnxviewtransaction,trnxviewpurchase

from .views import patient,patientregister,doctor,doctorregistration,adminlogin,txmanagerlogin,adminloginaction

from .views import viewadminpatientspage,viewadmindoctorspage,viewadmintransactionspage,logout,activatepatients,activatedoctors

from patients.urls import urlpatterns as patients\_urlpattern

from patients.views import patientlogincheck,patientsendsymptoms,patientsymtomsanalysis,patientsymptomsview,checkandpay,transactionmanagement,patientpurchaseblkmodel

from doctors.views import doctorlogincheck,doctoranalyzesysmptoms,DoctorsSendPriscription,DoctorPriscription,purchaseviewbydoctor,doctorviewtransaction

urlpatterns = [

#url(r'^patients/',include((patients\_urlpattern,'patients'),namespace='patients'),name='patients'),

url(r'^admin/', admin.site.urls),

url(r'^$',home, name="home"),

url(r'^patient/', patient, name="patient"),

url(r'^home/',home, name="home"),

url(r'^patientregister/',patientregister, name="patientregister"),

url(r'^doctor/',doctor, name="doctor"),

url(r'^doctorregistration/',doctorregistration, name="doctorregistration"),

url(r'^adminlogin/',adminlogin, name="adminlogin"),

url(r'^txmanagerlogin/',txmanagerlogin, name="txmanagerlogin"),

url(r'^adminloginaction/',adminloginaction,name="adminloginaction"),

url(r'^viewadminpatientspage/',viewadminpatientspage,name="viewadminpatientspage"),

url(r'^viewadmindoctorspage/',viewadmindoctorspage,name="viewadmindoctorspage"),

url(r'^viewadmintransactionspage/',viewadmintransactionspage,name="viewadmintransactionspage"),

url(r'^logout/',logout,name="logout"),

#url(r'^activatepatients/',activatepatients,name="activatepatients")

url(r'^activatepatients/$', activatepatients, name="activatepatients"),

url(r'^activatedoctors/$', activatedoctors, name="activatedoctors"),

url(r'^trnxmngrloginaction/',trnxmngrloginaction,name="trnxmngrloginaction"),

url(r'^trnxviewtransaction/',trnxviewtransaction,name="trnxviewtransaction"),

url(r'^trnxviewpurchase/',trnxviewpurchase,name="trnxviewpurchase"),

url(r'^patientlogincheck/', patientlogincheck, name="patientlogincheck"),

url(r'^patientsendsymptoms/', patientsendsymptoms, name="patientsendsymptoms"),

url(r'^patientsymtomsanalysis/', patientsymtomsanalysis, name="patientsymtomsanalysis"),

url(r'^patientsymptomsview/',patientsymptomsview, name="patientsymptomsview"),

url(r'^checkandpay/',checkandpay, name="checkandpay"),

url(r'^transactionmanagement/', transactionmanagement, name="transactionmanagement"),

url(r'patientpurchaseblkmodel/',patientpurchaseblkmodel, name="patientpurchaseblkmodel"),

url(r'^doctorlogincheck/',doctorlogincheck, name="doctorlogincheck"),

url(r'^doctoranalyzesysmptoms/',doctoranalyzesysmptoms,name="doctoranalyzesysmptoms"),

url(r'^DoctorsSendPriscription/',DoctorsSendPriscription, name="DoctorsSendPriscription"),

url(r'^DoctorPriscription/',DoctorPriscription,name="DoctorPriscription"),

url(r'^purchaseviewbydoctor/',purchaseviewbydoctor,name="purchaseviewbydoctor"),

url(r'^doctorviewtransaction/',doctorviewtransaction,name="doctorviewtransaction"),

]

**Models.py**

from django.db import models

class patientregistrationmodel(models.Model):

name = models.CharField(max\_length=100)

loginid = models.CharField(max\_length=100)

password = models.CharField(max\_length=100)

mobile = models.CharField(max\_length=100)

email = models.CharField(max\_length=100)

locality = models.CharField(max\_length=100)

address = models.CharField(max\_length=100)

city = models.CharField(max\_length=100)

state = models.CharField(max\_length=100)

authkey = models.CharField(max\_length=100)

status = models.CharField(max\_length=100)

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

return self.email

class docotrtregistrationmodel(models.Model):

doctorname = models.CharField(max\_length=100)

loginid = models.CharField(max\_length=100)

password = models.CharField(max\_length=100)

mobile = models.CharField(max\_length=100)

emailid = models.CharField(max\_length=100)

locality = models.CharField(max\_length=100)

address = models.CharField(max\_length=100)

city = models.CharField(max\_length=100)

state = models.CharField(max\_length=100)

authkey = models.CharField(max\_length=100)

status = models.CharField(max\_length=100)

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

return self.emailid

main **View.py**

from django.shortcuts import render,redirect,HttpResponseRedirect,HttpResponse

from django.contrib import messages

from .forms import patientregistrationform,doctorregistrationform

from .models import patientregistrationmodel,docotrtregistrationmodel

from random import randint

from django.db.models import Sum,Max

from patients.models import transactionsstore

from django.db.models import Sum,Max

from patients.models import transactionsstore

from doctors.models import doctorreplaysysmptoms

def home(request):

template = 'home.html'

context = {}

return render(request,template,context)

def patient(request):

template = 'patient.html'

context = {}

return render(request,template,context)

def doctor(request):

template = 'doctor.html'

context = {}

return render(request,template,context)

def patientregister(request):

if request.method=='POST':

form = patientregistrationform(request.POST)

if form.is\_valid():

print('Am Not Human but no humanity')

form.save()

messages.success(request, 'You have been successfully registered')

return HttpResponseRedirect('patient')

else:

print("Invalid form")

else:

form = patientregistrationform()

return render(request,'patientregister.html',{'form':form})

def doctorregistration(request):

if request.method=='POST':

form = doctorregistrationform(request.POST)

if form.is\_valid():

form.save()

messages.success(request, 'You have been successfully registered')

return HttpResponseRedirect('doctor')

else:

print("Invalid doctor Form")

else:

form = doctorregistrationform()

return render(request,'doctorregister.html',{'form':form})

def adminlogin(request):

return render(request,'adminlogin.html')

def txmanagerlogin(request):

return render(request,'trxlogin.html')

def trnxmngrloginaction(request):

if request.method == "POST":

if request.method == "POST":

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

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

if usid == 'txmanager' and pswd == 'txmanager':

return render(request,'admins/tranxhome.html')

else:

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

return render(request,'trxlogin.html')

def adminloginaction(request):

if request.method == "POST":

if request.method == "POST":

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

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

if usid == 'admin' and pswd == 'admin':

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

else:

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

return render(request,'adminlogin.html')

def viewadminpatientspage(request):

patientdata = patientregistrationmodel.objects.all()

#return HttpResponse("Redirect to Admin View Patients")

return render(request,'admins/viewppatientsdata.html',{'object':patientdata})

def viewadmindoctorspage(request):

docotrtdata = docotrtregistrationmodel.objects.all()

#return HttpResponse("Redirect to Admin View Patients")

return render(request,'admins/viewdoctordata.html',{'object':docotrtdata})

#return render(request,'adminactivateDoctors.html')

def viewadmintransactionspage(request):

ledbal = transactionsstore.objects.aggregate(Sum('ledgerbalance'))

x = ledbal.get("ledgerbalance\_\_sum")

x = round(x,2)

print("Total Ledger Balance ",x)

id = request.session['docid']

obj= transactionsstore.objects.last()

print("The Last Transactin ID ",obj)

print("Latest Ledger Balance ",obj.ledgerbalance)

userdata = transactionsstore.objects.all()

lststate = {

'ledbalance':x

}

return render(request,"admins/viewadmintransactionspage.html",{'object':userdata,'dph':lststate,'dpdet':obj})

# return HttpResponse("Redirect to Transaction Page")

#return render(request,'adminaviewtransactions.html')

def logout(request):

return render(request,'home.html')

def activatepatients(request):

if request.method=='GET':

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

authkey = random\_with\_N\_digits(8)

status = 'activated'

print("PID = ",pid,authkey,status)

patientregistrationmodel.objects.filter(id=pid).update(authkey=authkey , status=status)

patientdata = patientregistrationmodel.objects.all()

#return HttpResponse("Redirect to Admin View Patients")

return render(request,'admins/viewppatientsdata.html',{'object':patientdata})

def activatedoctors(request):

if request.method=='GET':

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

authkey = random\_with\_N\_digits(8)

status = 'activated'

print("PID = ",pid,authkey,status)

docotrtregistrationmodel.objects.filter(id=pid).update(authkey=authkey , status=status)

docotordata = docotrtregistrationmodel.objects.all()

#return HttpResponse("Redirect to Admin View Patients")

return render(request,'admins/viewdoctordata.html',{'object':docotordata})

def random\_with\_N\_digits(n):

range\_start = 10\*\*(n-1)

range\_end = (10\*\*n)-1

return randint(range\_start, range\_end)

def patientsymptomsview(request):

patientsysmptoms = patientsymptomsanalysis.objects.all()

#return HttpResponse("Redirect to Admin View Patients")

return render(request,'patients/viewppatientsdata.html',{'object':patientsysmptoms})

def trnxviewtransaction(request):

if request.method=='GET':

ledbal = transactionsstore.objects.aggregate(Sum('ledgerbalance'))

x = ledbal.get("ledgerbalance\_\_sum")

x = round(x,2)

print("Total Ledger Balance ",x)

id = request.session['docid']

obj= transactionsstore.objects.last()

print("The Last Transactin ID ",obj)

print("Latest Ledger Balance ",obj.ledgerbalance)

userid = request.session['userid']

userdata = transactionsstore.objects.filter(docid=id)

lststate = {

'ledbalance':x

}

return render(request,"admins/trnxviewtransac.html",{'object':userdata,'dph':lststate,'dpdet':obj})

def trnxviewpurchase(request):

docdataset = doctorreplaysysmptoms.objects.filter(status='purchase')

return render(request,"admins/trnxviewpurchase.html",{'object':docdataset})

Patients **View.py**

from django.shortcuts import render,redirect,HttpResponseRedirect,HttpResponse

from django.contrib import messages

from random import randint

from converginblockchain.models import patientregistrationmodel

from doctors.models import doctorreplaysysmptoms

from django.db.models import Sum,Max

from nltk.corpus import wordnet

from .models import patientsymptomsanalysis,blkchainapproach,transactionsstore

import datetime

import time

import random

# Create your views here.

class NaiveBayes:

def \_\_init\_\_(self, name, symptoms):

self.name = name

self.symptoms = symptoms

known\_diseases = [

NaiveBayes('cold', set("sorethroat|runnynose|congestion|cough|aches".split("|"))),

NaiveBayes('flu', set("fever|headache|muscleaches|returningfever".split("|"))),

NaiveBayes('ebola', set("tiredness|death|bruisingover90%fbody|blackblood".split("|"))),

NaiveBayes('spondylosis', set("Tingling|numbness|weakness|Abnormalreflexes|musclespasms".split("|"))),

NaiveBayes('alcohol', set("antisocialbehaviour|impulsivity|self-harm|loneliness".split("|"))),

NaiveBayes('stroke', set("Numbness|arm|Confusion|Difficultyspeaking|difficultywalking|slurredspeech".split("|"))),

NaiveBayes('respiratory', set("phlegm|fever|difficultybreathing|abluetinttotheskin|chestpain|wheezing".split("|"))),

NaiveBayes('pulmonary', set("dyspnea|Fatigue|faintingspells|Chestpressure|Swelling".split("|"))),

NaiveBayes('bronchus', set("Coughwithblood|Wheezing|Shortnessofbreath|Chestpain|Flushing".split("|"))),

NaiveBayes('Diabetes', set("thirstandhunger|urination|Weightlossorgain|Fatigue|Nausea|Blurredvision".split("|"))),

NaiveBayes('Alzheimer', set("Memoryloss|Visionloss|Misplacingitems|Difficultymakingdecisions|meaninglessrepetition ".split("|"))),

NaiveBayes('Dehydration', set("vomiting|sweating|Individuals|drymouth|lethargy|dizziness".split("|"))),

NaiveBayes('Tuberculosis', set("Coughing|Chestpain|weightloss|Fatigue|Fever|Night sweats|Chills".split("|"))),

NaiveBayes('Cirrhosis', set("jaundice|Weakness|Lossofappetite|Itching|Easybruising|darkurine".split("|"))),

NaiveBayes('Plague', set("diarrhoea|nausea|nausea|malaise|delirium|shortnessofbreath|tenderlymphnode".split("|"))),

]

def patientlogincheck(request):

if request.method == "POST":

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

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

try:

check = patientregistrationmodel.objects.get(loginid=usid, password=pswd)

request.session['userid'] = check.id

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

print("patient id ",check.id)

status = check.status

if status == "activated":

return render(request,'patients/patientpage.html')

else:

messages.success(request, 'Your Account Not at activated')

return render(request,'patient.html')

return render(request,'patients/patientpage.html')

except:

pass

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

return render(request,'patient.html')

def patientsendsymptoms(request):

return render(request,'patients/patientsendsymptoms.html')

def patientsymtomsanalysis(request):

if request.method == "POST":

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

userid = request.session['userid'] #request.POST.get('id')

username = ''

email = ''

ts = time.time()

st = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d %H:%M:%S')

try:

check = patientregistrationmodel.objects.get(id=userid)

username = check.name

email = check.email

storsympto = symptoms

#print("patient id ",check.email,symptoms)

symptoms = symptoms.lower()

symptoms = symptoms.split(",")

possible = []

for symptom in symptoms:

for disease in known\_diseases:

if symptom in disease.symptoms:

possible.append(disease.name)

if possible:

#print("You may have these diseases:",len(possible) )

#print(\*possible)

for x in possible:

print('Disease is = ',x)

syn = wordnet.synsets(x)

description = ''

#print(type(syn))

if len(syn)!=0:

description = syn[0].definition()

#print(\*syn[0].examples())

print(description)

#var = patientsymptomsanalysis.objects.create(patintid=userid,patinetname=username,email=email,patinetallsymptoms=storsympto,diseasname=x,descriptions=description,createdon=st)

#print('Var ve type ',type(var))

else:

description = 'No Data found'

patientsymptomsanalysis.objects.create(patintid=userid,patinetname=username,email=email,patinetallsymptoms=storsympto,diseasname=x,descriptions=description)

print('Desc',description)

else:

messages.success(request,"Good news! You're going to have a disease named after you!")

messages.success(request, 'Thanking you for sending your sysmptoms we will get back you soon')

return render(request,'patients/patientsendsymptoms.html')

except Exception as e:

#pass

print(str(e))

messages.success(request, 'There is a problam in analysing your sysmptoms')

return render(request,'patients/patientsendsymptoms.html')

def patientsymptomsview(request):

userid = request.session['userid']

patientsysmptoms = patientsymptomsanalysis.objects.filter(patintid=userid)

return render(request,"patients/patientsymptomsview.html",{'object':patientsysmptoms})

def checkandpay(request):

if request.method=='GET':

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

#sysmpotmsid = int(sysid)

#reqDate = request.GET.get('createdate')

#print("SysmpID ",sysid)

#print(type(sysmpid),type(sysmpotmsid))

check = doctorreplaysysmptoms.objects.get(sysid=sysid)

docname = check.doctorname

print("Doctor Name ",docname)

return render(request,"patients/checkandpay.html",{'object':check})

def transactionmanagement(request):

if request.method=='POST':

docname = request.POST.get('doctorname')

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

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

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

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

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

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

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

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

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

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

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

ledbalance = float(price)/10

expiredate = month+'/'+year

trnxid = random.randint(100000000000,999999999999)

print('Post Method Workd Fine ',docname,"= ",docid)

print('Thanking you for Purchase ',trnxid ,'Expire Date ',expiredate)

blkchainapproach.objects.create(docname=docname,docid=int(docid),patientname=patientname,patientid=int(patientid),disease=dieses,price=float(price),sysmptid=int(sysid),ledgerbalance=ledbalance,tranxid=trnxid )

transactionsstore.objects.create(docid=int(docid),patientid=int(patientid),nameoncard=nameoncard,cvv=int(cvv),cardnumber=cardnumber,expiredate=expiredate,tranxid=trnxid,price=float(price),ledgerbalance=ledbalance)

#Update the payment section

doctorreplaysysmptoms.objects.filter(sysid=sysid).update(status='purchase')

ledbal = transactionsstore.objects.aggregate(Sum('ledgerbalance'))

print("Led balance ",ledbal)

#patientsymptomsanalysis.objects.filter(id=sysid).update(status='given',docname=doctorname)

#patientsymptomsanalysis.objects.filter(id=sysid).update(status='given',docname=doctorname)

#patientsymptomsanalysis.objects.filter(id=sysid).update(status='given',docname=doctorname)

#patientsymptomsanalysis.objects.filter(id=sysid).update(status='given',docname=doctorname)

messages.success(request,"Thnking you for Purchase keep updates for our news at Dp")

#print("Return Data type ",type(ledbal))

x = ledbal.get("ledgerbalance\_\_sum")

x = round(x,2)

ctx = {

'tranxid': trnxid,

'ledbala': x

}

print('Response Dictonary ',ctx)

return render(request,"patients/paidsheet.html",{'object':ctx})

#return HttpResponse('Its Works')

def patientpurchaseblkmodel(request):

if request.method=='GET':

ledbal = transactionsstore.objects.aggregate(Sum('ledgerbalance'))

x = ledbal.get("ledgerbalance\_\_sum")

x = round(x,2)

print("Total Ledger Balance ",x)

obj= transactionsstore.objects.last()

print("The Last Transactin ID ",obj)

print("Latest Ledger Balance ",obj.ledgerbalance)

userid = request.session['userid']

userdata = transactionsstore.objects.filter(patientid=userid)

lststate = {

'ledbalance':x

}

return render(request,"patients/patientpurchaseblkmodel.html",{'object':userdata,'dph':lststate,'dpdet':obj})

**base.html**

{% load static %}

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<meta name="description" content="">

<meta name="author" content="Mark Otto, Jacob Thornton, and Bootstrap contributors">

<meta name="generator" content="Jekyll v3.8.5">

<title>BlockChain Machine Learning · Bootstrap</title>

<link rel="canonical" href="https://getbootstrap.com/docs/4.3/examples/jumbotron/">

<!-- Bootstrap core CSS -->

<link href=" {% static 'css/bootstrap.min.css' %}" rel="stylesheet" >

<style>

.bd-placeholder-img {

font-size: 1.125rem;

text-anchor: middle;

-webkit-user-select: none;

-moz-user-select: none;

-ms-user-select: none;

user-select: none;

}

@media (min-width: 768px) {

.bd-placeholder-img-lg {

font-size: 3.5rem;

}

}

</style>

<!-- Custom styles for this template -->

<link href=" {% static 'css/jumbotron.css'%}" rel="stylesheet">

</head>

<body>

<nav class="navbar navbar-expand-md navbar-dark fixed-top bg-dark">

<a class="navbar-brand" href="#">Converging Blockchain and Machine Learning for Healthcare</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarsExampleDefault" aria-controls="navbarsExampleDefault" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarsExampleDefault">

<ul class="navbar-nav mr-auto">

<li class="nav-item ">

<a class="nav-link" href="{% url 'home' %}">Home </a>

</li>

<li class="nav-item">

<a class="nav-link" href="{% url 'adminlogin' %}">Admin</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{% url 'patient' %}">Patients</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{% url 'doctor' %}">Doctors</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{% url 'txmanagerlogin' %}">Transaction Manager</a>

</li>

</ul>

<form class="form-inline my-2 my-lg-0">

<input class="form-control mr-sm-2" type="text" placeholder="Search" aria-label="Search">

<button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>

</form>

</div>

</nav>

<main role="main">

{% block contents %}

<!-- Main jumbotron for a primary marketing message or call to action -->

{% endblock %}

<hr>

</div> <!-- /container -->

</main>

<footer class="container">

<p>&copy; Company 2017-2019</p>

</footer>

<script src="{% static 'js/jquery-3.3.1.slim.min.js' %}" ></script>

<script>window.jQuery || document.write('<script src="/docs/4.3/assets/js/vendor/jquery-slim.min.js"><\/script>')</script><script src="{% static 'js/bootstrap.bundle.min.js' %}" ></script></body>

</html>

**AdminTransaction.html**

{% extends 'trnxbase.html' %}

{% block contents %}

<style>

table {

border-collapse: collapse;

}

table, th, td {

border: 1px solid black;

}

</style>

<div class="container">

<!-- Example row of columns -->

<h2>View All Transaction </h2>

<table border="2px" >

<tr>

<th style="color: darkblue">S.No</th>

<th style="color: darkblue">Doctor ID</th>

<th style="color:darkblue">Name on Card</th>

<th style="color:darkblue">CVV</th>

<th style="color:darkblue">Card Number</th>

<th style="color:darkblue">Expire Date</th>

<th style="color:darkblue">Transaction Id</th>

<th style="color:darkblue">Price </th>

<th style="color:darkblue">Ledger Ammount</th>

</tr>

{% for i in object %}

<tr>

<td style="color:brown">{{forloop.counter}}</td>

<td style="color:darkolivegreen">{{i.docid}}</td>

<td style="color:brown">{{i.nameoncard}}</td>

<td style="color:brown">{{i.cvv}}</td>

<td style="color:brown">{{i.cardnumber}}</td>

<td style="color:brown">{{i.expiredate}}</td>

<td style="color:brown">{{i.tranxid}}</td>

<td style="color:brown">{{i.price}}</td>

<td style="color:brown">{{i.ledgerbalance}}</td>

{% endfor %}

</table>

<h3>Total Ledger Balance <font style="color: blueviolet">{{dph.ledbalance}}</font> </h3><br/>

<h3>Last Transaction Details </h3>

<table>

<tr>

<th style="color:blueviolet">S.No</th>

<th style="color: crimson">Doctor ID</th>

<th style="color:darkblue">Name on Card</th>

<th style="color:darkblue">CVV</th>

<th style="color:darkblue">Card Number</th>

<th style="color:darkblue">Expire Date</th>

<th style="color:darkblue">Transaction Id</th>

<th style="color:darkblue">Price </th>

<th style="color:darkblue">Ledger Ammount</th>

</tr>

<tr>

<td style="color:brown">1</td>

<td style="color:darkolivegreen">{{dpdet.docid}}</td>

<td style="color:crimson">{{dpdet.nameoncard}}</td>

<td style="color:crimson">{{dpdet.cvv}}</td>

<td style="color:crimson">{{dpdet.cardnumber}}</td>

<td style="color:crimson">{{dpdet.expiredate}}</td>

<td style="color:crimson">{{dpdet.tranxid}}</td>

<td style="color:crimson">{{dpdet.price}}</td>

<td style="color:crimson">{{dpdet.ledgerbalance}}</td>

</table>

<div class="row">

<div class="col-md-4">

<p>

</p>

</div>

</div>

{% endblock %}