**Source Code :-**

from django.shortcuts import render

from django.template import RequestContext

from django.contrib import messages

from django.http import HttpResponse

from hashlib import sha256

import time

import pickle

import json

from datetime import datetime

import BC

from BC.Blockchain import Blockchain

from datetime import date

def ViewChain(request):

if request.method == 'GET':

with open('BC\_DB.txt', 'rb') as input:

blockchain = pickle.load(input)

input.close()

output = ''

output+='<table align=\"center\" border=\"1\"><tr><th><font size=\"3\" color=\"black\">Transaction No</th>'

output+='<th><font size=\"3\" color=\"black\">From Peer</th>'

output+='<th><font size=\"3\" color=\"black\">To Peer</th>'

output+='<th><font size=\"3\" color=\"black\">Coin</th>'

output+='<th><font size=\"3\" color=\"black\">Transaction Date</th>'

output+='</tr>'

print(len(blockchain.translist))

for i in range(len(blockchain.translist)):

b = blockchain.translist[i]

print(b)

arr = b.split(",")

output+='<tr><td><font size=\"3\" color=\"black\">'+str(i)+'</td>'

output+='<td><font size=\"3\" color=\"black\">'+str(arr[0])+'</td>'

output+='<td><font size=\"3\" color=\"black\">'+str(arr[1])+'</td>'

output+='<td><font size=\"3\" color=\"black\">'+str(arr[2])+'</td>'

output+='<td><font size=\"3\" color=\"black\">'+str(arr[3])+'</td></tr>' output+='<tr></tr><tr></tr><tr></tr><tr></tr><tr></tr></table><br/><br/><br/><br/><br/><br/><br/><br/>'

context= {'data':output}

return render(request, 'ViewChain.html', context)

def Transactions(request):

if request.method == 'GET':

with open('BC\_DB.txt', 'rb') as input:

blockchain = pickle.load(input)

input.close()

output = ''

output+='<tr><td><font size=\"3\" color=\"black\">From&nbsp;Peer</td><td><select name=\"t1\">'

for i in range(len(blockchain.chain)):

b = blockchain.chain[i]

if b.index > 0:

output+='<option value='+b.transactions[0]+'>'+b.transactions[0]+'</option>'

output+='</select></td></tr><tr><td><font size=\"3\" color=\"black\">To&nbsp;Peer</td><td><select name=\"t2\">'

for i in range(len(blockchain.chain)):

b = blockchain.chain[i]

if b.index > 0:

output+='<option value='+b.transactions[0]+'>'+b.transactions[0]+'</option>'

output+='</select></td></tr><tr><td><font size=\"3\" color=\"black\">Coins</td><td><input type=\"text\" name=\"t3\"></td></tr>'

output+='<tr><td></td><td><input type=\"submit\" value=\"Submit Transaction\"></td></td></tr></table>'

context= {'data':output}

return render(request, 'Transactions.html', context)

def TransactionsSubmit(request):

if request.method == 'POST':

frompeer = request.POST.get('t1', False)

topeer = request.POST.get('t2', False)

coin = request.POST.get('t3', False)

with open('BC\_DB.txt', 'rb') as input:

blockchain = pickle.load(input)

input.close()

today = date.today()

output = frompeer+","+topeer+","+coin+","+str(today)

blockchain.addTransaction(output)

with open('BC\_DB.txt', 'wb') as outputs:

pickle.dump(blockchain, outputs, pickle.HIGHEST\_PROTOCOL)

outputs.close()

output = 'Transaction complete between '+frompeer+' and '+topeer+' coins '+coin

context= {'data':output}

return render(request, 'Transactions.html', context)

def BlockAdded(request):

if request.method == 'POST':

name = request.POST.get('t1', False)

with open('BC\_DB.txt', 'rb') as input:

blockchain = pickle.load(input)

input.close()

peer = ' '

for i in range(len(blockchain.peer)):

block = blockchain.peer[i]

#arr = block.split(",")

if name == block:

peer = block

blockchain.add\_new\_transaction(peer)

blockchain.mine()

break;

blockchain.peer.remove(peer)

with open('BC\_DB.txt', 'wb') as outputs:

pickle.dump(blockchain, outputs, pickle.HIGHEST\_PROTOCOL)

outputs.close()

output = ''

output+='<tr><td><font size=\"3\" color=\"black\">Choose&nbsp;Peer&nbsp;Name</td><td><select name=\"t1\">'

for i in range(len(blockchain.peer)):

block = blockchain.peer[i]

#arr = block.split(",")

output+='<option value='+block+'>'+block+'</option>'

output+='</select></td></tr><tr><td></td><td><input type=\"submit\" value=\"Add To Block\"></td></td></tr></table>'

output+='<table align=\"center\" border=\"1\"><tr><th><font size=\"3\" color=\"black\">Block No</th>'

output+='<th><font size=\"3\" color=\"black\">Block Name</th>'

output+='<th><font size=\"3\" color=\"black\">Previous Proof Hash</th>'

output+='<th><font size=\"3\" color=\"black\">New Hash</th>'

output+='<th><font size=\"3\" color=\"black\">Block Created Time</th>'

output+='</tr>'

for i in range(len(blockchain.chain)):

b = blockchain.chain[i]

if b.index > 0:

output+='<tr><td><font size=\"3\" color=\"black\">'+str(b.index)+'</td>'

output+='<td><font size=\"3\" color=\"black\">'+str(b.transactions)+'</td>'

output+='<td><font size=\"3\" color=\"black\">'+str(b.previous\_hash)+'</td>'

output+='<td><font size=\"3\" color=\"black\">'+str(b.hash)+'</td>'

output+='<td><font size=\"3\" color=\"black\">'+str(str(datetime.fromtimestamp(b.timestamp)))+'</td></tr>' output+='<tr></tr><tr></tr><tr></tr><tr></tr><tr></tr></table><br/><br/><br/><br/><br/><br/><br/><br/>'

context= {'data':output}

return render(request, 'AddToBlock.html', context)

def AddToBlock(request):

if request.method == 'GET':

output = ''

output+='<tr><td><font size=\"3\" color=\"black\">Choose&nbsp;Peer&nbsp;Name</td><td><select name=\"t1\">'

with open('BC\_DB.txt', 'rb') as input:

blockchain = pickle.load(input)

input.close()

for i in range(len(blockchain.peer)):

block = blockchain.peer[i]

#arr = block.split(",")

output+='<option value='+block+'>'+block+'</option>'

output+='</select></td></tr><tr><td></td><td><input type=\"submit\" value=\"Add To Block\"></td></td></tr></table>'

output+='<table align=\"center\" border=\"1\"><tr><th><font size=\"3\" color=\"black\">Block No</th>'

output+='<th><font size=\"3\" color=\"black\">Block Name</th>'

output+='<th><font size=\"3\" color=\"black\">Previous Proof Hash</th>'

output+='<th><font size=\"3\" color=\"black\">New Hash</th>'

output+='<th><font size=\"3\" color=\"black\">Block Created Time</th>'

output+='</tr>'

for i in range(len(blockchain.chain)):

b = blockchain.chain[i]

if b.index > 0:

output+='<tr><td><font size=\"3\" color=\"black\">'+str(b.index)+'</td>'

output+='<td><font size=\"3\" color=\"black\">'+str(b.transactions)+'</td>'

output+='<td><font size=\"3\" color=\"black\">'+str(b.previous\_hash)+'</td>'

output+='<td><font size=\"3\" color=\"black\">'+str(b.hash)+'</td>'

output+='<td><font size=\"3\" color=\"black\">'+str(str(datetime.fromtimestamp(b.timestamp)))+'</td></tr>'

output+='<tr></tr><tr></tr><tr></tr><tr></tr><tr></tr></table><br/><br/><br/><br/><br/><br/><br/><br/>'

context= {'data':output}

return render(request, 'AddToBlock.html', context)

def index(request):

if request.method == 'GET':

return render(request, 'index.html', {})

def AddPeer(request):

if request.method == 'GET':

blockchain = Blockchain()

#with open('BC\_DB.txt', 'wb') as outputs:

# pickle.dump(blockchain, outputs, pickle.HIGHEST\_PROTOCOL)

#outputs.close()

with open('BC\_DB.txt', 'rb') as input:

blockchain = pickle.load(input)

input.close()

output = ''

output+='<table align=\"center\" border=\"1\"><tr><th><font size=\"3\" color=\"black\">Added Peer Details</th></tr>'

for i in range(len(blockchain.peer)):

block = blockchain.peer[i]

output+='<tr><td><font size=\"3\" color=\"black\">'+block+'</td></tr>' output+='<tr></tr><tr></tr><tr></tr><tr></tr><tr></tr></table><br/><br/><br/><br/><br/><br/><br/><br/>'

context= {'data':output}

return render(request, 'AddPeer.html', context)

def AddPeerAction(request):

if request.method == 'POST':

name = request.POST.get('t1', False)

#blockchain = Blockchain()

with open('BC\_DB.txt', 'rb') as input:

blockchain = pickle.load(input)

input.close()

x = name

blockchain.addPeer(x)

with open('BC\_DB.txt', 'wb') as outputs:

pickle.dump(blockchain, outputs, pickle.HIGHEST\_PROTOCOL)

outputs.close()

output = ''

output+='<table align=\"center\" border=\"1\"><tr><th><font size=\"3\" color=\"black\">Added Peer Details</th></tr>'

for i in range(len(blockchain.peer)):

block = blockchain.peer[i]

output+='<tr><td><font size=\"3\" color=\"black\">'+block+'</td></tr>'

output+='<tr></tr><tr></tr><tr></tr><tr></tr><tr></tr></table><br/><br/><br/><br/><br/><br/><br/><br/>'

context= {'data':output}

return render(request, 'AddPeer.html', context)