SMART INVIGILATION CODES

❖ Views.py file

```
❖ from django.db.models.query import QuerySet
❖ from django.http.response import HttpResponse, HttpResponseRedirect
❖ from django.shortcuts import render, redirect, reverse, get_object_or_404
❖ from django.contrib import messages
❖ from django.contrib.auth import login, authenticate
❖ from django.contrib.auth.forms import UserCreationForm
❖ from django.contrib.auth.models import User, auth
❖ from django.core.mail import send_mail
❖ from django.conf import settings
❖ from django.contrib.auth.decorators import login_required
❖ from .models import *
❖ from .forms import *
❖
❖ # project 2
❖ from facenet_pytorch import MTCNN
❖ from PIL import Image
❖ from matplotlib import pyplot  as plt
❖ import numpy as np
❖ import math
❖ import requests
❖
❖ import argparse
❖ import torch
❖ import cv2
❖ #from .face import detect_faces
❖ from . import NameFind
❖


❖ import random
❖ # import predFacePoseApp
❖ # Create your views here.
❖
❖ #SEND EMAIL AND SMS
❖
❖ from twilio.rest import Client
❖ from django.core.mail import EmailMultiAlternatives
❖ from django.template.loader import render_to_string
❖ from django.utils.html import strip_tags
❖ from django.core.mail import send_mail
❖ from django.conf import settings
❖
```

```
❖ def homePage(request):
❖
❖     return render(request, 'SmartInvigilationApp/homePage.html')
❖
❖ def signin(request):
❖
❖     if request.method == 'POST':
❖         email = request.POST.get('email')
❖         password = request.POST.get('password')
❖
❖         user = auth.authenticate(email=email, password=password)
❖
❖         if user is not None:
❖             auth.login(request, user)
❖             return redirect('homePage')
❖         else:
❖             messages.info(request, 'Credentials Invalid, Email or Password
   is incorrect')
❖             return redirect('signin')
❖
❖     else:
❖         return render(request, 'SmartInvigilationApp/homePage.html')
❖
❖ def logout(request):
❖     auth.logout(request)
❖     return redirect('homePage')
❖
❖ def SmartInvigilationProject(request):
```

```
    import os
    BASE_DIR = os.path.dirname(os.path.abspath(__file__))
    #C:\Users\DIMOSO JR\Desktop\ProjectWork\SmartInvigilation\SmartInvigilationProject\SmartInvigilationApp
    print(BASE_DIR)

# CODES ZA KUGET USERNAME AND PASSWORD
    # username = request.user
    # camera_no = request.POST.get('camera_no')
    # print(username)
    # print(camera_no)

    # form = InvigilationStaffsForm()
    if request.method == 'POST':
        username = request.user.username
        email = request.user.email
        camera_no = request.POST.get('camera_no')
        to_phone_number = request.user.phone
        #from_phone_number = request.POST.get('from_phone_number')

        save_invigilator = InvigilationStaffs.objects.create(username=username, camera_no=camera_no, email=email, to_phone_number=to_phone_number)
        save_invigilator.save()



        left_offset = 20
        fontScale = 2
        fontThickness = 3
        text_color = (0,0,255)
        lineColor = (255, 255, 0)

        device = torch.device('cuda:0' if torch.cuda.is_available() else 'cpu')
        print(f'Running on device: {device}')

        mtcnn = MTCNN(image_size=160,
                      margin=0,
                      min_face_size=20,
                      thresholds=[0.6, 0.7, 0.7], # MTCNN thresholds
                      factor=0.709,
                      post_process=True,
                      device=device # If you don't have GPU
```

```
                )

        # Landmarks: [Left Eye], [Right eye], [nose], [left mouth], [right
mouth]
        def npAngle(a, b, c):
            ba = a - b
            bc = c - b

            cosine_angle = np.dot(ba,
bc)/(np.linalg.norm(ba)*np.linalg.norm(bc))
            angle = np.arccos(cosine_angle)

            return np.degrees(angle)

        # def visualize(image, landmarks_, angle_R_, angle_L_, pred_):
        #     fig , ax = plt.subplots(1, 1, figsize= (8,8))

        #     leftCount = len([i for i in pred_ if i == 'L'])
        #     rightCount = len([i for i in pred_ if i == 'R'])
        #     frontalCount = len([i for i in pred_ if i == ''])
        #     facesCount = len(pred_) # Number of detected faces (above
the threshold)
        #     ax.set_title(f"Number of detected faces = {facesCount} \n
frontal = {frontalCount}, left = {leftCount}, right = {rightCount}")
        #     for landmarks, angle_R, angle_L, pred in zip(landmarks_,
angle_R_, angle_L_, pred_):

        #         if pred == 'C':
        #             color = 'red'
        #         elif pred == 'R':
        #             color = 'blue'
        #         else:
        #             color = 'green'

        #         point1 = [landmarks[0][0], landmarks[1][0]]
        #         point2 = [landmarks[0][1], landmarks[1][1]]

        #         point3 = [landmarks[2][0], landmarks[0][0]]
        #         point4 = [landmarks[2][1], landmarks[0][1]]

        #         point5 = [landmarks[2][0], landmarks[1][0]]
        #         point6 = [landmarks[2][1], landmarks[1][1]]
        #         for land in landmarks:
        #             pass
        #         #TO PRINT TRIANGLE AND CIRCLES ON A FACE
```

```
❖         #             #ax.scatter(land[0], land[1])
❖         #         # plt.plot(point1, point2, 'y', linewidth=3)
❖         #         # plt.plot(point3, point4, 'y', linewidth=3)
❖         #         # plt.plot(point5, point6, 'y', linewidth=3)
❖         #         #looking_center = int(pred)
❖         #         looking_right = int(math.floor(angle_R))
❖         #         looking_left = int(math.floor(angle_L))
❖
❖         #         plt.text(point1[0], point2[0], f"{pred} \n
   {looking_left}, {looking_right}",
❖         #                  size=20, ha="center", va="center", color=color)
❖         #         ax.imshow(image)
❖         #         fig.savefig(BASE_DIR+'/OutputImages/Output_detection.jpg
   ')
❖         #     return print('Done detect')
❖
❖         def visualizeCV2(frame, landmarks_, angle_R_, angle_L_, pred_):
❖
❖             for landmarks, angle_R, angle_L, pred in zip(landmarks_,
   angle_R_, angle_L_, pred_):
❖
❖                 if pred == 'C':
❖                     color = (0, 255, 0) #Green-BGR
❖                 elif pred == 'Right Profile':
❖                     color = (255, 0, 0)
❖                 else:
❖                     color = (0, 0, 255)
❖
❖                 point1 = [int(landmarks[0][0]), int(landmarks[1][0])]
❖                 point2 = [int(landmarks[0][1]), int(landmarks[1][1])]
❖
❖                 point3 = [int(landmarks[2][0]), int(landmarks[0][0])]
❖                 point4 = [int(landmarks[2][1]), int(landmarks[0][1])]
❖
❖                 point5 = [int(landmarks[2][0]), int(landmarks[1][0])]
❖                 point6 = [int(landmarks[2][1]), int(landmarks[1][1])]
❖
❖                 for land in landmarks:
❖                     pass
❖                     #UKITAKA KUWEKA LINE KWENYE FACE UNCOMMENT BELOW THEN
   TOA PASS HAPO JUU
❖                     # cv2.circle(frame, (int(land[0]), int(land[1])),
   radius=5, color=(0, 255, 255), thickness=-1)
❖
❖
```

```
                # cv2.line(frame, (int(landmarks[0][0]),
int(landmarks[0][1])), (int(landmarks[1][0]), int(landmarks[1][1])),
lineColor, 3)
                # cv2.line(frame, (int(landmarks[0][0]),
int(landmarks[0][1])), (int(landmarks[2][0]), int(landmarks[2][1])),
lineColor, 3)
                # cv2.line(frame, (int(landmarks[1][0]),
int(landmarks[1][1])), (int(landmarks[2][0]), int(landmarks[2][1])),
lineColor, 3)

                text_sizeR, _ = cv2.getTextSize(pred,
cv2.FONT_HERSHEY_PLAIN, fontScale, 4)
                text_wR, text_hR = text_sizeR

                cv2.putText(frame, pred,(point1[0], point2[0]),
cv2.FONT_HERSHEY_PLAIN, fontScale, color, fontThickness, cv2.LINE_AA)



        def predFacePose(frame):

            bbox_, prob_, landmarks_ = mtcnn.detect(frame, landmarks=True)
# The detection part producing bounding box, probability of the detected
face, and the facial landmarks
            angle_R_List = []
            angle_L_List = []
            predLabelList = []

            x,y,width,height = 100,100,200,150
            color = (0,255,0)
            thickness = 2









            if bbox_ is not None and prob_ is not None and landmarks_ is
not None:

                for bbox, landmarks, prob in zip(bbox_, landmarks_,
prob_):
                    if bbox is not None: # To check if we detect a face in
the image
```

```
                        if prob > 0.9:#0.9 # To check if the detected face
has probability more than 90%, to avoid
                            angR = npAngle(landmarks[0], landmarks[1],
landmarks[2]) # Calculate the right eye angle
                            angL = npAngle(landmarks[1], landmarks[0],
landmarks[2])# Calculate the left eye angle
                            angle_R_List.append(angR)
                            angle_L_List.append(angL)
                            
                            
                            
                            
                            
                            if ((int(angR) in range(35, 57)) and
(int(angL) in range(35, 58))):
                                predLabel='' #C 'Frontal'
                                predLabelList.append(predLabel)
                            else:
                                if angR < angL:
                                    LeftAngle = int(angL)
                                    predLabel= f'{LeftAngle}-R'
                                    
                                    print(f"angR (θ2) = {angR} and angL
(θ1) =  {angL}, Hence, Student is Looking Right at Angle {angL} ")
                                    print(" ")
                                    print(" ")
                                    
                                    
                                    if angL > 80: #-80
                                        
                                        #HIZI NI KWA AJILI KUTUMA EMAIL
KWA INVIGILATOR
                                        
                                        
                                        # subject = "SMART INVIGILATION
SYSTEM"
                                        # message = f"Hey {username} Some
of the students are cheating Right, Please! Check your screen"
                                        # from_email =
settings.EMAIL_HOST_USER
                                        # recipient_list = [email]
                                        # send_mail(subject, message,
from_email, recipient_list, fail_silently=True)
                                        
```

```
                                        #MWISHO WA KWA AJILI KUTUMA EMAIL
KWA INVIGILATOR
❖
❖                                       #HIZI NI KWA AJILI YA KUTUMA SMS
KWA INVIGILATOOR
❖
❖
❖
❖                                       # account_sid =
"ACec294a515a22883f25914c532734bf23"
❖                                       # auth_token =
"0d924ac3503fe120262b106a4c7a0bd3"
❖                                       # client = Client(account_sid,
auth_token)
❖                                       # message = client.messages \
❖                                       #             .create(
❖                                       #                 body="Some of
the students are cheating Right, Please! Check your screen",
❖                                       #                 from_='+14068023
763',
❖                                       #                 #to='+2556284315
07'
❖                                       #                 to
=to_phone_number
❖                                       #                         )
❖                                       # print("Some of the students are
cheating Right, Please! Check your screen")
❖
❖
❖
❖                                       #MWISHO WA KUTUMA MSG KWA
INVIGILATOR
❖
❖
❖                                       cv2.circle(frame,
(int(landmarks[1][0]), int(landmarks[1][1])), radius=50, color=(0, 0,
255), thickness=5)
❖                                       predLabel= f'{LeftAngle}'
#'Cheating L'
❖
❖                                       print("LEFT ANGLE IS > 80:  THEN;
")
❖
❖
❖
```

```
                                        print(f"Student is Cheating Right
at Angle {angL} ")
                                        print(" ")
                                        print(" ")


                                        # KWA AJILI YA KUCHORA PEMBE TATU
                                        # cv2.line(frame,
(int(landmarks[0][0]), int(landmarks[0][1])), (int(landmarks[1][0]),
int(landmarks[1][1])), lineColor, 3)
                                        # cv2.line(frame,
(int(landmarks[0][0]), int(landmarks[0][1])), (int(landmarks[2][0]),
int(landmarks[2][1])), lineColor, 3)
                                        # cv2.line(frame,
(int(landmarks[1][0]), int(landmarks[1][1])), (int(landmarks[2][0]),
int(landmarks[2][1])), lineColor, 3)


                                else:
                                    RightAngle = int(angR)
                                    predLabel=f'{RightAngle}-L'


                                    print(f"angR (θ2) = {angR} and angL
(θ1) =  {angL}, Hence, Student is Looking Left at Angle {angR} ")
                                    print(" ")


                                    if angR > 80: #80

                                        #HIZI NI KWA AJILI KUTUMA EMAIL
KWA INVIGILATOR


                                        # subject = "SMART INVIGILATION
SYSTEM"
                                        # message = f"Hey {username} Some
of the students are cheating Left, Please! Check your screen"
                                        # from_email =
settings.EMAIL_HOST_USER
                                        # recipient_list = [email]
                                        # send_mail(subject, message,
from_email, recipient_list, fail_silently=True)

```

```
                                        #MWISHO KWA AJILI KUTUMA EMAIL KWA
INVIGILATOR

                                        #HIZI NI KWA AJILI YA KUTUMA SMS
KWA INVIGILATOOR



                                        # account_sid =
"ACec294a515a22883f25914c532734bf23"
                                        # auth_token =
"0d924ac3503fe120262b106a4c7a0bd3"
                                        # client = Client(account_sid,
auth_token)
                                        # message = client.messages \
                                        #             .create(
                                        #                 body="Some of
the students are cheating Left, Please! Check your screen",
                                        #                 from_='+14068023
763',
                                        #                 #to='+2556284315
07'
                                        #                 to
=to_phone_number
                                        #                         )
                                        # print("Some of the students are
cheating Left, Please! Check your screen")

                                        #MWISHO KWA AJILI YA KUTUMA SMS
KWA INVIGILATOOR



                                        cv2.circle(frame,
(int(landmarks[0][0]), int(landmarks[1][1])), radius=50, color=(0, 0,
255), thickness=5)
                                        predLabel=f'{RightAngle}'
#'Cheating R'

                                        print("RIGHT ANGLE IS > 80: THEN;
")


```

```
                                        print(f"Student is Cheating Left
at Angle {angR} ")
                                        print(" ")
                                        print(" ")


                                        # KWA AJILI YA KUCHORA PEMBE TATU
                                        # cv2.line(frame,
(int(landmarks[0][0]), int(landmarks[0][1])), (int(landmarks[1][0]),
int(landmarks[1][1])), lineColor, 3)
                                        # cv2.line(frame,
(int(landmarks[0][0]), int(landmarks[0][1])), (int(landmarks[2][0]),
int(landmarks[2][1])), lineColor, 3)
                                        # cv2.line(frame,
(int(landmarks[1][0]), int(landmarks[1][1])), (int(landmarks[2][0]),
int(landmarks[2][1])), lineColor, 3)

                                        #cv2.rectangle(frame,(x,y),(x +
width, y + height), color, thickness)

                                    predLabelList.append(predLabel)
                            else:
                                print('The detected face is Less then the
detection threshold')
                                continue
                        else:
                            print('No face detected in the image')
                            continue
                    # FACE YA MTU INAONEKANA KWENYE CAMERA
                    # VALUES ZAKE NDO HIZI KWA ANGLES ZOTE

                    # print(f"right {angle_R_List} ")
                    # print(f"left {angle_L_List} ")
                    # print(f"center {predLabelList} ")

                    return landmarks_, angle_R_List, angle_L_List,
predLabelList
                else:
                    # KAMAHAMNA MTU KWENYE CAMERA IKASOME HIZI DEFAULT
                    # VALUES ILI KUREMOVE ERROR INAYOSEMA "CAN NOT UNPACK
NONETYPE OBJECT"
                    angle_R_List = [41.499546, 38.9971]
                    angle_L_List= [44.377758, 45.907673]
                    predLabelList= ['', '']
                    # bbox_ = [[-76.825165 345.9478    87.617516 547.3037  ]
```

```
                #          [302.04672  163.34067  380.0983   259.25977 ]
                #          [120.40265  130.62308  186.46005  210.27997 ]]
                landmarks_ = [[[-34.039093,   421.1461    ],
                               [ 16.470798,   427.12183   ],
                               [-35.07744,    464.84344   ],
                               [-41.21428,    498.56042   ],
                               [ -0.88659286, 504.5458    ]],

                              [[324.0439,     198.46864   ],
                               [360.09473,    200.7623    ],
                               [339.04395,    216.24658   ],
                               [324.7361,     238.5289    ],
                               [353.1474,     240.4968    ]],

                              [[133.74072,    161.84549   ],
                               [163.7682,     158.71199   ],
                               [145.17691,    176.29892   ],
                               [138.92558,    193.42422   ],
                               [164.13974,    191.53346   ]]]
                # prob_ =[0.9998667, 0.9999882, 0.9963965]
                #return landmarks_, angle_R_List, angle_L_List,
   predLabelList
                print("There is no face to detect")
                # print("None")
                # print(f"r is {angle_R_List} ")
                # print(f"l is {angle_L_List} ")
                # print(f"prob_ is {predLabelList} ")

                return landmarks_, angle_R_List, angle_L_List,
   predLabelList


        def predFacePoseApp():



            source = 0

            mysource = camera_no #now naingiza video path lkn km unaingiza
   no?
            #mysource = int(camera_no)
            #print(camera_no)
            # Create a video capture object from the VideoCapture Class.
            #video_cap = cv2.VideoCapture(BASE_DIR+"/videos/4.mp4")
            video_cap = cv2.VideoCapture(BASE_DIR+mysource)
```

```
❖
❖             #km unaingiza no path itakuwa
❖             #video_cap = cv2.VideoCapture(mysource)
❖
❖             # Create a named window for the video display.
❖             win_name = 'SMART INVIGILATION SYSTEM'
❖             cv2.namedWindow(win_name, cv2.WINDOW_NORMAL)
❖             # video_cadesired_width = 1400
❖             # desired_height = 800
❖             cv2.setWindowProperty(win_name, cv2.WND_PROP_FULLSCREEN,
   cv2.WINDOW_FULLSCREEN)
❖             cv2.resizeWindow(win_name, 1400, 1000)
❖             #dim = (video_cadesired_width, desired_height)
❖             left_offset = 20
❖             fontScale = 2
❖             fontThickness = 3
❖             text_color = (255,0,0) #(0,0,255)
❖
❖             # #Mwisho wa full screen Model
❖
❖
❖             #MWANZO  KWA AJILI YA KURECODI VIDEO
❖
❖             random.seed()
❖             random_number = random.randint(1,100)
❖             print(random_number)
❖
❖             #video_cap = cv2.VideoCapture(BASE_DIR+"/videos/6.mp4")
❖             fourcc=cv2.VideoWriter_fourcc('M','J','P','G')
❖             capturing_win_name = 'CAPTURING LIVE VIDEO'
❖             cv2.namedWindow(capturing_win_name, cv2.WINDOW_NORMAL)
❖             # video_cadesired_width = 1400
❖             # desired_height = 800
❖             cv2.setWindowProperty(capturing_win_name,
   cv2.WND_PROP_FULLSCREEN, cv2.WINDOW_FULLSCREEN)
❖             cv2.resizeWindow(capturing_win_name, 1400, 1000)
❖             # out=cv2.VideoWriter(BASE_DIR+'/saved-
   media/my.mp4',fourcc,20.0,(640,480))
❖
❖             if (video_cap.isOpened() == False):
❖                 print("Unable to read camera")
❖
❖             frame_width = int(video_cap.get(3))
❖             frame_height = int(video_cap.get(4))
```

```
❖             out=cv2.VideoWriter(BASE_DIR+f'/saved-
media/SmartInvigilationVideo-
{random_number}.avi',fourcc,10,(frame_width,frame_height))
❖
❖             #MWISHO  KWA AJILI YA KURECODI VIDEO
❖
❖
❖             face_cascade =
cv2.CascadeClassifier(BASE_DIR+'/cascades/data/haarcascade_frontalface_def
ault.xml')
❖             eye_cascade =
cv2.CascadeClassifier(BASE_DIR+'/cascades/data/haarcascade_eye.xml')
❖
❖             #MWANZO WA KUCAPTURE IMAGES
❖
❖             load =
cv2.CascadeClassifier(BASE_DIR+'/cascades/data/haarcascade_frontalface_def
ault.xml')
❖
❖             #cap=cv2.VideoCapture("videos/4.mp4")
❖
❖             random.seed()
❖             random_number = random.randint(1,100)
❖             print(random_number)
❖
❖             # name = input("Enter Username : ")
❖             # id = input("Enter Unique id (maxLen = 4) : ")
❖
❖             val=0
❖
❖             #MWISHO WA KUCAPTURE IMAGES
❖
❖
```

```
❖             while True:
❖
❖                 # #MWANZO WA KUCAPTURE IMAGES
❖                 # status,frame = video_cap.read()
❖                 # gray = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
❖                 # faces = load.detectMultiScale(gray,1.3,5)
❖                 # eyes = load.detectMultiScale(gray,1.3,5)
❖
❖                 # for (x,y,w,h) in faces:
❖                 #     val=val+1
❖                 #     cv2.imwrite(BASE_DIR+"/CheatingStudentsCapturedImage
   s/"+str(random_number)+"."+str(val)+".jpg",gray[y:y+h,x:x+w])
❖                 #     #cv2.imwrite(BASE_DIR+"/Data/"+str(val)+".jpg",gray[
   y:y+h,x:x+w])
❖
❖                 #     cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2)
❖                 #     cv2.waitKey(50)
❖
❖
❖
❖                 # #cv2.imshow('FaceDetect',frame)
❖                 # cv2.waitKey(1)
❖
❖                 # # if(val >= 10):
❖                 # #     break
❖                 # if cv2.waitKey(1) & 0xFF == ord('q'):
❖                 #     break
❖
❖
❖                 # #MWISHO WA KUCAPTURE IMAGES
❖
❖
❖
❖
❖
❖                 #MWANZO  KWA AJILI YA KURECODI VIDEO
❖
❖                 ret, frame = video_cap.read()
❖                 if (ret==True):
❖                     cv2.flip(frame,180)
❖                     out.write(frame)
```

```
❖
❖                   # Display the resulting frame
❖                       cv2.imshow(capturing_win_name, frame)
❖                       if cv2.waitKey(1) & 0xFF == ord('q'):
❖                           break
❖                   else:
❖                       continue
❖
❖                   #MWISHO  KWA AJILI YA KURECODI VIDEO
❖
❖
❖
❖                   # Read one frame at a time using the video capture object.
❖                   has_frame, frame = video_cap.read()
❖                   if not has_frame:
❖                       #break
❖                       print("It is not possible to read image from the
   camera")
❖                       continue
❖
❖
❖
❖
❖                   #DRAWING BOX ON STUDENT'S FACE
❖
❖                   faces = face_cascade.detectMultiScale(frame, 1.3,
   5)
❖                   for (x, y, w, h) in
   faces:
❖                       NameFind.draw_box(frame, x, y, w, h)
❖
❖                   #cv2.imshow('Face Detection Using Haar-Cascades ',
   frame)
❖                   if cv2.waitKey(1) & 0xFF ==
   ord('q'):
❖                       break
❖
❖                   #MWISHO WA  DRAWING BOX
❖
❖                   landmarks_, angle_R_List, angle_L_List, predLabelList =
   predFacePose(frame)
❖
❖                   # print(angle_R_List)
```

```
                # print(angle_L_List)
                # angle_L_List and angle_R_List are list inwhich
                # the left and right angles are appended

                # Annotate each video frame.
                visualizeCV2(frame, landmarks_, angle_R_List,
angle_L_List, predLabelList)

                #To draw a graphy
                #visualize(frame, landmarks_, angle_R_List, angle_L_List,
predLabelList)

                for Left_angle in angle_R_List:
                    if Left_angle > 80:
                        #MWANZO WA KUCAPTURE IMAGES

                        gray2 = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
                        faces2 = load.detectMultiScale(gray2,1.3,5)


                        for (x,y,w,h) in faces2:
                            val=val+1
                            cv2.imwrite(BASE_DIR+"/LeftCapturedImages/"+st
r(Left_angle)+"."+str(val)+".jpg",gray2[y:y+h,x:x+w])


                        # if(val >= 3):
                        #     break
                        # if cv2.waitKey(1) & 0xFF == ord('q'):
                        #     break
                        #MWISHO WA KUCAPTURE IMAGES

                for Right_angle in angle_L_List:
                    if Right_angle > 80:
                        #MWANZO WA KUCAPTURE IMAGES

                        gray2 = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
                        faces2 = load.detectMultiScale(gray2,1.3,5)


                        for (x,y,w,h) in faces2:
                            val=val+1
                            cv2.imwrite(BASE_DIR+"/RightCapturedImages/"+s
tr(Left_angle)+"."+str(val)+".jpg",gray2[y:y+h,x:x+w])

```

```
                        # if(val >= 3):
                        #     break
                        # if cv2.waitKey(1) & 0xFF == ord('q'):
                        #     break
                        #MWISHO WA KUCAPTURE IMAGES









                cv2.imshow(win_name, frame)

                key = cv2.waitKey(1)

                # You can use this feature to check if the user selected the `q` key to quit the video stream.
                if key == ord('Q') or key == ord('q') or key == 27:
                    # Exit the loop.
                    break

            video_cap.release()
            cv2.destroyWindow(win_name)



        messages.success(request, f"Invigilation Completed Successfully By - {username} ")
        predFacePoseApp()



        return redirect('starting_page')


                #return HttpResponse("welll")
                #return render(request, 'SmartInvigilationApp/homePage.html')


@login_required(login_url='homePage')
def starting_page(request):

    return render(request, 'SmartInvigilationApp/starting_page.html')
```

```
❖
❖  @login_required(login_url='homePage')
❖  def wifi_page(request):
❖
❖      return render(request, 'SmartInvigilationApp/wifi_page.html')
❖

❖  def record_video(request):
❖      import numpy as np
❖      import cv2
❖      import os
❖      import random
❖
❖      # username = request.user
❖      # camera_no = request.POST.get('camera_no')
❖      # print(username)
❖      # print(camera_no)
❖
❖
❖      if request.method == 'POST':
❖          username = request.user.username
❖          camera_no = request.POST.get('camera_no')
❖
❖          save_invigilator =
   InvigilationStaffs.objects.create(username=username, camera_no=camera_no)
❖          save_invigilator.save()
❖
❖          BASE_DIR = os.path.dirname(os.path.abspath(__file__))
❖          win_name = 'SMART INVIGILATION SYSTEM'
❖          cv2.namedWindow(win_name, cv2.WINDOW_NORMAL)
❖          # video_cadesired_width = 1400
❖          # desired_height = 800
❖          cv2.setWindowProperty(win_name, cv2.WND_PROP_FULLSCREEN,
   cv2.WINDOW_FULLSCREEN)
❖          cv2.resizeWindow(win_name, 1400, 1000)
❖
❖          #username = request.user
❖
❖          mysource = camera_no #now naingiza video path lkn km unaingiza no?
❖          #mysource = int(camera_no)
❖          #print(camera_no)
❖          # Create a video capture object from the VideoCapture Class.
❖          #video_cap = cv2.VideoCapture(BASE_DIR+"/videos/4.mp4")
❖          mycamera = cv2.VideoCapture(BASE_DIR+mysource)
❖
```

```
❖
❖
❖
❖         #km unaingiza no path itakuwa
❖         #mycamera = cv2.VideoCapture(mysource)
❖
❖         random.seed()
❖         random_number = random.randint(1,100)
❖         print(random_number)
❖


❖         fourcc=cv2.VideoWriter_fourcc('M','J','P','G')
❖         # out=cv2.VideoWriter(BASE_DIR+'/saved-
  media/my.mp4',fourcc,20.0,(640,480))
❖
❖         if (mycamera.isOpened() == False):
❖             print("Unable to read camera")
❖
❖         frame_width = int(mycamera.get(3))
❖         frame_height = int(mycamera.get(4))
❖         out=cv2.VideoWriter(BASE_DIR+f'/saved-
  media/SmartInvigilationVideo-
  {random_number}.avi',fourcc,10,(frame_width,frame_height))
❖
❖         while(True):
❖             # mycamerature frame-by-frame
❖             ret, frame = mycamera.read()
❖             if (ret==True):
❖                 cv2.flip(frame,180)
❖                 out.write(frame)
❖
❖             # Display the resulting frame
❖                 cv2.imshow(win_name, frame)
❖                 if cv2.waitKey(1) & 0xFF == ord('q'):
❖                     break
❖             else:
❖                 break
❖
❖         # When everything done, release the mycamerature
❖         mycamera.release()
❖         cv2.destroyAllWindows()
❖
❖         messages.success(request, f"Video Recorded Successfully By
  {username} ")
❖         return redirect('recording_video_page')
```

```
❖
❖       messages.success(request, f"Error! Form is not valid")
❖       return redirect('recording_video_page')
❖
❖       #return render(request,
    'SmartInvigilationApp/recording_video_page.html')
❖

❖   @login_required(login_url='homePage')
❖   def recording_video_page(request):
❖
❖       return render(request,
    'SmartInvigilationApp/recording_video_page.html')
❖



❖
❖
```

- Urls.py file

```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.homePage, name="homePage"),
    path('signin/', views.signin, name="signin"),
    path('logout/', views.logout, name='logout'),

   ## path('WebcamInvigilation/', views.WebcamInvigilation,
name='WebcamInvigilation'),
    path('SmartInvigilationProject/', views.SmartInvigilationProject,
name='SmartInvigilationProject'),
    path('record_video/', views.record_video, name='record_video'),
    path('recording_video_page/', views.recording_video_page,
name='recording_video_page'),

    path('starting_page/', views.starting_page, name='starting_page'),
    path('wifi_page/', views.wifi_page, name='wifi_page'),

]
```

❖ Models.py file

```
from django.db import models
from datetime import datetime, date
from django.contrib.auth.models import AbstractBaseUser, BaseUserManager
from django.conf import settings
from django.contrib.auth.models import User
from django.utils import timezone
# Create your models here.


class MyUserManager(BaseUserManager):
    def create_user(self, email, username, phone, password=None):
        if not email:
            raise ValueError("email is required")
        if not username:
            raise ValueError("Your user name is required")
        if not phone:
            raise ValueError("Your phone number is required")



        user=self.model(
            email=self.normalize_email(email),
            username=username,
            phone=phone,


        )
        user.set_password(password)
        user.save(using=self._db)
        return user
    def create_superuser(self, email, username, phone, password=None):
        user=self.create_user(
            email=self.normalize_email(email),
            username=username,
            phone=phone,
            password=password,

        )
        user.is_admin=True
        user.is_staff=True

        user.is_superuser=True
```

```
        user.save(using=self._db)
        return user




class MyUser(AbstractBaseUser):
    email=models.EmailField(verbose_name="email", max_length=100, unique=True)
    first_name=models.CharField(verbose_name="first name", max_length=100,
unique=False)
    username=models.CharField(verbose_name="user name", max_length=100,
unique=True)
    middle_name=models.CharField(verbose_name="middle name", max_length=100,
unique=False)
    last_name=models.CharField(verbose_name="last name", max_length=100,
unique=False)
    company_name=models.CharField(verbose_name="company name", max_length=100,
unique=False)
    phone=models.CharField(default="+255746244743", verbose_name="phone",
max_length=13, unique=True)
    profile_image = models.ImageField(upload_to='get_profile_image_filepath',
blank=True, null=True)
    date_joined=models.DateTimeField(verbose_name="date joined",
auto_now_add=True)
    last_login=models.DateTimeField(verbose_name="last login", auto_now=True)
    is_admin=models.BooleanField(default=False)
    is_active=models.BooleanField(default=True)
    is_staff=models.BooleanField(default=True)
    is_superuser=models.BooleanField(default=False)
    hide_email = models.BooleanField(default=True)



    USERNAME_FIELD="email"
    REQUIRED_FIELDS=['username','phone']

    objects=MyUserManager()

    def __str__(self):
        return self.username



    def has_perm(self, perm, obj=None):
```

```
        return True

    def has_module_perms(self, app_label):
        return True


class InvigilationStaffs(models.Model):

    username = models.CharField(max_length=200,blank=False,null=False)
    email = models.EmailField(default="juniordimoso8@gmail.com",
max_length=100,blank=False,null=False)
    camera_no = models.CharField(max_length=200,default="/videos/S1.mp4",
blank=False,null=False)
    to_phone_number = models.CharField(default="+255746244743",
max_length=13,blank=True,null=True)
    created = models.DateTimeField(auto_now_add=True)
    updated = models.DateTimeField(auto_now=True)

    def __str__(self):
        return self.username
```

- Admin.py file

```
from django.contrib import admin
from .models import *
from django.contrib.auth.admin import UserAdmin as BaseUserAdmin
# Register your models here.

class MyUserAdmin(BaseUserAdmin):
    list_display=('username', 'email', 'phone','date_joined', 'last_login',
'is_admin', 'is_active')
    search_fields=('email', 'phone', 'username')
    readonly_fields=('date_joined', 'last_login')
    filter_horizontal=()
    list_filter=('last_login',)
    fieldsets=()

    add_fieldsets=(
        (None,{
            'classes':('wide'),
```

```
            'fields':('email', 'username','first_name', 'middle_name',
'last_name', 'company_name', 'phone', 'profile_image', 'password1', 'password2'),
        }),
    )

    ordering=('email',)

class InvigilationStaffsAdmin(admin.ModelAdmin):
    list_display = ['username','camera_no','email','to_phone_number',
'created','updated']
    list_filter=['created','updated']


admin.site.register(MyUser, MyUserAdmin)
admin.site.register(InvigilationStaffs,InvigilationStaffsAdmin)
```

- Forms.py file

```
from .models import *
from django.forms import ModelForm
from django import forms


class InvigilationStaffsForm(forms.ModelForm):

    class Meta:
        model = InvigilationStaffs
        fields ='__all__'
```

- NameFind.py file

```
#     ----------- FUNCTION TO READ THE FILE AND ADD THE NAMES AND IDs IN
TO TUPLES

import cv2
import math
import  time
```

```
import datetime
from django.utils import timezone
import os
BASE_DIR = os.path.dirname(os.path.abspath(__file__))

now_time = datetime.datetime.now()

face =
cv2.CascadeClassifier(BASE_DIR+'/cascades/data/haarcascade_frontalface_def
ault.xml')
glass_cas =
cv2.CascadeClassifier(BASE_DIR+'/cascades/data/haarcascade_eye_tree_eyegla
sses.xml')

WHITE = [255, 255, 255]

def FileRead():
    Info = open(BASE_DIR+"/Names.txt", "r")                        # Open
th text file in readmode
    NAME = []                                              # The tuple to
store Names
    while (True):                                          # Read all the
lines in the file and store them in two tuples
        Line = Info.readline()
        if Line == '':
            break
        NAME.append (Line.split(",")[1].rstrip())

    return NAME                                         # Return the two
tuples

Names = FileRead()                                     # Run the above
Function to get the ID and Names Tuple

#     ------------------- FUNCTION TO FIND THE NAME  ---------------------
-------------------------------------

def ID2Name(ID, conf):
    if ID > 0:
        #NameString = "Name: " + Names[ID-1] + " Distance: " +
(str(round(conf)) )
        NameString = "Name: " + Names[ID-
1]                                  # Find the Name using the index of the
ID
    else:
```

```
❖         NameString = " Face Not Recognised "  # Find the Name using the
index of the ID
❖
❖     return NameString
❖
❖ #     ------------------- THIS FUNCTION READ THE FILE AND ADD THE NAME TO
THE END OF THE FILE  -----------------
❖
❖ def AddName():
❖     Name = input('Enter Your Name ')
❖     Info = open(BASE_DIR+"/Names.txt", "r+")
❖     ID = ((sum(1 for line in Info))+1)
❖     Info.write(str(ID) + "," + Name + "\n")
❖     print ("Name Stored in " + str(ID))
❖     Info.close()
❖     return ID
❖
❖ #     ------------------- DRAW THE BOX AROUND THE FACE, ID and
CONFIDENCE  ----------------------------------
❖
❖ def DispID(x, y, w, h, NAME, Image):
❖
❖     #  ------------------------------- THE POSITION OF THE ID BOX  -----
----------------------------------------
❖
❖     Name_y_pos = y - 10
❖     Name_X_pos = x + w//2 - (len(NAME)*7//2)
❖
❖     if Name_X_pos < 0:
❖         Name_X_pos = 0
❖     elif (Name_X_pos +10 + (len(NAME) * 7) > Image.shape[1]):
❖           Name_X_pos= Name_X_pos - (Name_X_pos +10 + (len(NAME) * 7) -
(Image.shape[1]))
❖     if Name_y_pos < 0:
❖         Name_y_pos = Name_y_pos = y + h + 10
❖
❖  #  ----------------------------------    THE DRAWING OF THE BOX AND
ID   ---------------------------------
❖
❖     draw_box(Image, x, y, w, h)
❖
❖
❖     cv2.rectangle(Image, (Name_X_pos-10, Name_y_pos-25), (Name_X_pos +10 +
(len(NAME) * 7), Name_y_pos-1), (0,0,0), -2)           # Draw a Black
Rectangle over the face frame
```

```
    cv2.rectangle(Image, (Name_X_pos-10, Name_y_pos-25), (Name_X_pos +10 +
(len(NAME) * 7), Name_y_pos-1), WHITE, 1)
    cv2.putText(Image, NAME, (Name_X_pos, Name_y_pos - 10),
cv2.FONT_HERSHEY_DUPLEX, .4, WHITE)                         # Print the
name of the ID

def draw_box(Image, x, y, w, h):
    cv2.line(Image, (x, y), (x + (w//5) ,y), WHITE, 2)
    cv2.line(Image, (x+((w//5)*4), y), (x+w, y), WHITE, 2)
    cv2.line(Image, (x, y), (x, y+(h//5)), WHITE, 2)
    cv2.line(Image, (x+w, y), (x+w, y+(h//5)), WHITE, 2)
    cv2.line(Image, (x, (y+(h//5*4))), (x, y+h), WHITE, 2)
    cv2.line(Image, (x, (y+h)), (x + (w//5) ,y+h), WHITE, 2)
    cv2.line(Image, (x+((w//5)*4), y+h), (x + w, y + h), WHITE, 2)
    cv2.line(Image, (x+w, (y+(h//5*4))), (x+w, y+h), WHITE, 2)

# ---------------     SECOND ID BOX      ----------------------
def DispID2(x, y, w, h, NAME, Image):

#  ------------------------------- THE POSITION OF THE ID BOX  ---------
----------------------------------------

    Name_y_pos = y - 40
    Name_X_pos = x + w/2 - (len(NAME)*7/2)

    if Name_X_pos < 0:
        Name_X_pos = 0
    elif (Name_X_pos +10 + (len(NAME) * 7) > Image.shape[1]):
          Name_X_pos= Name_X_pos - (Name_X_pos +10 + (len(NAME) * 7) -
(Image.shape[1]))
    if Name_y_pos < 0:
        Name_y_pos = Name_y_pos = y + h + 10

 #  ----------------------------------    THE DRAWING OF THE BOX AND
ID   ---------------------------------
    cv2.rectangle(Image, (Name_X_pos-10, Name_y_pos-25), (Name_X_pos +10 +
(len(NAME) * 7), Name_y_pos-1), (0,0,0), -2)          # Draw a Black
Rectangle over the face frame
    cv2.rectangle(Image, (Name_X_pos-10, Name_y_pos-25), (Name_X_pos +10 +
(len(NAME) * 7), Name_y_pos-1), WHITE, 1)
    cv2.putText(Image, NAME, (Name_X_pos, Name_y_pos - 10),
cv2.FONT_HERSHEY_DUPLEX, .4, WHITE)                         # Print the
name of the ID

# ---------------     THIRD ID BOX      ----------------------
```

```
❖ def DispID3(x, y, w, h, NAME, Image):
❖
❖ #  ---------------------------------- THE POSITION OF THE ID BOX  ---------
   ---------------------------------------
❖
❖     Name_y_pos = y - 70
❖     Name_X_pos = x + w/2 - (len(NAME)*7/2)
❖
❖     if Name_X_pos < 0:
❖         Name_X_pos = 0
❖     elif (Name_X_pos +10 + (len(NAME) * 7) > Image.shape[1]):
❖           Name_X_pos= Name_X_pos - (Name_X_pos +10 + (len(NAME) * 7) -
   (Image.shape[1]))
❖     if Name_y_pos < 0:
❖         Name_y_pos = Name_y_pos = y + h + 10
❖
❖  #  ----------------------------------    THE DRAWING OF THE BOX AND
   ID   ---------------------------------------
❖     cv2.rectangle(Image, (Name_X_pos-10, Name_y_pos-25), (Name_X_pos +10 +
   (len(NAME) * 7), Name_y_pos-1), (0,0,0), -2)           # Draw a Black
   Rectangle over the face frame
❖     cv2.rectangle(Image, (Name_X_pos-10, Name_y_pos-25), (Name_X_pos +10 +
   (len(NAME) * 7), Name_y_pos-1), WHITE, 1)
❖     cv2.putText(Image, NAME, (Name_X_pos, Name_y_pos - 10),
   cv2.FONT_HERSHEY_DUPLEX, .4, WHITE)                        # Print the
   name of the ID
❖
❖ def DrawBox(Image, x, y, w, h):
❖     cv2.rectangle(Image, (x, y), (x + w, y + h), (255, 255, 255), 1)     #
   Draw a rectangle arround the face
❖
❖ # ---------------------------- THIS FUNCTION TAKES IN SPEC CASCADE, FACE
   CASCADE AND AN IMAGE
❖ # ------------------------- IT RETURNS A CROPPED FACE AND IF POSSIBLE
   STRAIGHTENS THE TILT OF THE HEAD
❖
❖ def DetectEyes(Image):
❖     Theta = 0
❖     rows, cols = Image.shape
❖     glass =
   glass_cas.detectMultiScale(Image)
         # This ditects the eyes
❖     for (sx, sy, sw, sh) in glass:
```

```
        if glass.shape[0] == 2:                                                 # The Image should have 2 eyes
            if glass[1][0] > glass[0][0]:
                DY = ((glass[1][1] + glass[1][3] / 2) - (glass[0][1] + glass[0][3] / 2))    # Height diffrence between the glass
                DX = ((glass[1][0] + glass[1][2] / 2) - glass[0][0] + (glass[0][2] / 2))    # Width diffrance between the glass
            else:
                DY = (-(glass[1][1] + glass[1][3] / 2) + (glass[0][1] + glass[0][3] / 2))   # Height diffrence between the glass
                DX = (-(glass[1][0] + glass[1][2] / 2) + glass[0][0] + (glass[0][2] / 2))   # Width diffrance between the glass

            if (DX != 0.0) and (DY != 0.0):                                     # Make sure the the change happens only if there is an angle
                Theta = math.degrees(math.atan(round(float(DY) / float(DX), 2)))            # Find the Angle
                print ("Theta  " + str(Theta))

                M = cv2.getRotationMatrix2D((cols / 2, rows / 2), Theta, 1)               # Find the Rotation Matrix
                Image = cv2.warpAffine(Image, M, (cols, rows))
                # cv2.imshow('ROTATED', Image)                                          # UNCOMMENT IF YOU WANT TO SEE THE

                Face2 = face.detectMultiScale(Image, 1.3, 5)                            # This detects a face in the image
                for (FaceX, FaceY, FaceWidth, FaceHeight) in Face2:
                    CroppedFace = Image[FaceY: FaceY + FaceHeight, FaceX: FaceX + FaceWidth]
                    return CroppedFace

def tell_time_passed():
    print ('TIME PASSED ' + str(round(((time.clock() - now_time)/60), 2)) + ' MINS')
```

- Base.html file

```
{% load static %}

<!DOCTYPE html>
<html>
<head>
  <title></title>

   <!--HII NI KWA AJILI YA ICONS ZOTE--->
 <link href="{% static 'SmartInvigilationApp/assets/vendor/fontawesome-
free/css/all.min.css' %}" rel="stylesheet">



        <!---jqeury css link---->
     <link rel="stylesheet" href="{% static
'SmartInvigilationApp/assets/jqueryui/jquery-ui.css' %}">
    <link rel="stylesheet" href="{% static
'SmartInvigilationApp/assets/jqueryui/jquery-ui.structure.min.css' %}">
    <link rel="stylesheet" href="{% static
'SmartInvigilationApp/assets/jqueryui/jquery-ui.structure.css' %}">
    <link rel="stylesheet" href="{% static
'SmartInvigilationApp/assets/jqueryui/jquery-ui.theme.min.css' %}">
     <link rel="stylesheet" href="{% static
'SmartInvigilationApp/assets/jqueryui/jquery-ui.theme.css' %}">

 <!-- Vendor CSS Files ----->
    <link href="{% static 'SmartInvigilationApp/assets/css/animate.min.css' %}"
rel="stylesheet">

    <link href="{% static 'SmartInvigilationApp/assets/css/bootstrap.min.css' %}"
rel="stylesheet">
    <link href="{% static 'SmartInvigilationApp/assets/vendor/bootstrap-
icons/bootstrap-icons.css' %}" rel="stylesheet">

    <link href="{% static 'SmartInvigilationApp/assets/vendor/fontawesome-
free/css/all.min.css' %}" rel="stylesheet">



    <style type="text/css">

  /*STYLES FOR HOME BUTTONS*/
```

```
*{
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    }
body{
    background-image:linear-gradient(rgba(0,0,0,0.7),rgba(0,0,0,0.7)),url("{%
static 'SmartInvigilationApp/assets/img/1q.jpeg' %}");
    background-size: cover;
}
.buttons-container{
    display: flex;
    justify-content: center;
    align-items: center;
    /*min-height: 100vh;*/
    margin-bottom: 2cm;
    margin-top: 20px;
    /*background: #333;*/

}
.buttons{
    display: flex;
    justify-content: center;
    align-items: center;
    flex-wrap: wrap;
    max-width: 800px;
    gap:50px;
}
.buttons label{
    position: relative;
    width: 190px;
    height: 150px;
    display: flex;
    justify-content: center;
    align-items: center;
    cursor: pointer;
    padding: 5px;
}
.buttons label input{
    appearance:none;

}
.buttons label span{
```

```
    position: absolute;
    width: 100%;
    height: 100%;
    background: linear-gradient(#555353, #363535, #303030);
    border: 10px solid #222;
    border-radius: 6px;
    box-shadow: inset 0 5px 1px rgba(0,0,0,0.35), 0 5px 5px rgba(0,0,0,0.5), 0
15px 25px rgba(0,0,0,0.35);

}
.buttons label input:checked ~ span{
    box-shadow: inset 0 5px 1px rgba(0,0,0,0.35), inset 0 5px 5px
rgba(0,0,0,0.5), inset 0 15px 25px rgba(0,0,0,0.35);

}
.buttons label span::before{
    content: '';
    position: absolute;
    inset:5px 3px;
    border-top: 1px solid #ccc;
    filter: blur(2px);

}
.buttons label i{
    position: relative;
    z-index: 10;
    font-size: 1.5rem;
    color: white;
}
.buttons label input:checked ~ i{
    color: red;
    text-shadow: 0 0 5px #219cf3, 0 0 8px #219cf3;
}




/*STYLES FOR LOGIN PAGE*/


.login{
  /*position: absolute;*/
  top: 0%;
  left: 0%;
```

```
  /*width: 400px;*/
  padding: 40px;
 /* transform: translate(-50%, -50%);*/
  background: linear-gradient(#141e30, #243b55);
  box-sizing: border-box;
  box-shadow: 0 15px 25px rgba(0,0,0,.6);
  border-bottom: 1.50px solid #03e9f4;
  border-radius: 10px;

}
.login h2{
  margin: 0 0 30px;
  padding: 0;
  color: #03e9f4;
  text-align: center;
}


.login .user-box{
  position: relative;
}
.login .user-box input{
  width: 100%;
  padding: 10px;
  font-size: 17px;
  color: white;
  margin-bottom: 30px;
  border: none;
  border-bottom: 1px solid #03e9f4;
  outline: none;
  background: transparent;
}
.login .user-box label{
  position: absolute;
  top:0;
  left: 0;
  padding: 10px 0;
  color: #03e9f4;
  pointer-events: none;
  font-size: 16px;
  transition: 0.5s;
}
.login .user-box input:focus ~ label,
.login .user-box input:valid ~ label{
  top: -20px;
```

```
  left: 0;
  color: #03e9f4;
  font-size: 12px;
}
.login p{
  color: darkgray;
  cursor: pointer;
}
.login p span{
  color: #03e9f4;
  float: right;
}
.login form .link-btn{
  position: relative;
  display: inline-block;
  padding: 10px 20px;
  color: #03e9f4;
  font-size: 16px;
  text-decoration: none;
  text-transform: uppercase;
  overflow: hidden;
  transition: 0.5s;
  margin-top: 40px;
  letter-spacing: 4px;
}
.login .link-btn:hover{
  background: #03e9f4;
  color: #fff;
  border-radius: 5px;
  box-shadow: 0 0 5px #03e9f4,
              0 0 25px #03e9f4,
              0 0 50px #03e9f4,
              0 0 100px #03e9f4;
}

input[type="submit"]{

    position: relative;
  display: inline-block;
  padding: 10px 20px;
  color: #03e9f4;
  font-size: 16px;
  text-decoration: none;
  text-transform: uppercase;
  overflow: hidden;
```

```
  transition: 0.5s;
  margin-top: 40px;
  letter-spacing: 4px;



}
/*input[type="submit"]{
    background: #03e9f4;
  color: #fff;
  border-radius: 5px;
  box-shadow: 0 0 5px #03e9f4,
              0 0 25px #03e9f4,
              0 0 50px #03e9f4,
              0 0 100px #03e9f4;

    }*/
.login .link-btn span{
  position: absolute;
  display: block;
}
.login .link-btn span:nth-child(1){
  top: 0;
  left: -100%;
  width: 100%;
  height: 2px;
  background: linear-gradient(90deg, transparent, #03e9f4);
  animation: animatedbtn1 1s linear infinite;
}
@keyframes animatedbtn1{
  0%{
    left: -100%;
  }
  50%,100%{
    left: 100%;
  }
}
.login .link-btn span:nth-child(2){
  top: -100%;
  right: 0;
  width: 2px;
  height: 100%;
  background: linear-gradient(90deg, transparent, #03e9f4);
  animation: animatedbtn2 1s linear infinite;
  animation-delay: .25s;
```

```
}
@keyframes animatedbtn2{
  0%{
    top: -100%;
  }
  50%,100%{
    top: 100%;
  }
}

.login .link-btn span:nth-child(3){
  bottom: 0;
  right: -100%;
  width: 100%;
  height: 2px;
  background: linear-gradient(90deg, transparent, #03e9f4);
  animation: animatedbtn3 1s linear infinite;
  animation-delay: .5s;
}
@keyframes animatedbtn3{
  0%{
    right: -100%;
  }
  50%,100%{
    right: 100%;
  }
}

.login .link-btn span:nth-child(4){
  bottom: -100%;
  left: 0;
  width: 2px;
  height: 100%;
  background: linear-gradient(90deg, transparent, #03e9f4);
  animation: animatedbtn4 1s linear infinite;
  animation-delay: .75s;
}
@keyframes animatedbtn4{
  0%{
    bottom: -100%;
  }
  50%,100%{
    bottom: 100%;
  }
}
```

```
@media screen and (max-width:430px){
  .login{
    width: 90%;
  }
}




    </style>

</head>
<body>


  {% block content %}


  {% endblock %}




<script src="{% static 'SmartInvigilationApp/assets/js/bootstrap.min.js'
%}"></script>

 <!--jquery code for autocomplete javascriptcode---->
<script src="{% static
'SmartInvigilationApp/assets/jqueryui/external/jquery/jquery.js' %}"
type="text/javascript"></script>
<script src="{% static 'SmartInvigilationApp/assets/jqueryui/jquery-ui.js' %}"
type="text/javascript"></script>




    <!-- ------------ AOS js Library  ------------------------- -->
    <script src="{% static 'SmartInvigilationApp/assets/js/aos.js' %}"></script>


</body>
</html>
```

❖ HomePage.html file

```
{% extends 'SmartInvigilationApp/base.html' %}

{% load static %}

{% block content %}
<!DOCTYPE html>
<html>
<head>
  <title></title>

  <style type="text/css">
     *{
       margin: 0;
       padding: 0;
       box-sizing: border-box;
       font-family: sans-serif;
       font-size: 16px;

     }
     ul li{
       list-style: none;
     }

 h5{
  font-size: 16px;
  font-family: sans-serif;
 }


  </style>

</head>
<body>

<div class="container-fluid">


<center>
    <h1 style="color: white;font-weight: bold;margin-top: 20px;">SMART
INVIGILATION SYSTEM</h1>

{% if user.is_authenticated %}
```

```
    {% if user.is_superuser %}
    <a href="/admin">
        <button type="button" class="btn btn-primary" data-bs-dismiss="modal"
style="font-size: 18px;margin-right: 20px;">Admin</button></a>
 {% else %}
 <h1 style="color: white;font-weight: bold;margin-top: 10px;font-size:
20px;color: red;">Welcome {{user.username}}</h1>

 {% endif %}


  {% else %}
   <h1 style="color: white;font-weight: bold;margin-top: 10px;font-size:
20px;color: red;">To start invigilation click login button to be verified!!</h1>
  {% endif %}

        {% if messages %}
{% for message in messages %}


          <div class="alert alert-warning alert-dismissible fade show"
role="alert">
  <strong style="color: red;font-size: 16px;">{{message}}</strong>
  <button type="button" class="btn-close" data-bs-dismiss="alert" aria-
label="Close"></button>
</div>
    {% endfor %}
  {% endif %}

  </center>


<!-- KAMA USER AMIJISAJILI ZINAANZIA HAPA -->

{% if user.is_authenticated %}


<div class="buttons-container">



<!-- <center>
    <h1 style="color: white;font-weight: bold;">SMART INVIGILATION<br>
SYSTEM</h1>
  </center> -->
```

```
<div class="buttons"><!-- mwanzo wa buttons-div -->

<label>
    <input type="checkbox" name="check">
    <span data-bs-toggle="modal" data-bs-target="#record"></span>
    <i class="fa fa-camera icon" data-bs-toggle="modal" data-bs-target="#record">
RECORD</i>
  </label>

  <label>
    <input type="checkbox" name="check">
    <span data-bs-toggle="modal" data-bs-target="#startexternal"></span>
    <i class="fa fa-camera icon" data-bs-toggle="modal" data-bs-
target="#startexternal"> CAMERA</i>
  </label>


  <label>
    <input type="checkbox" name="check">
    <span data-bs-toggle="modal" data-bs-target="#startwifi"></span>
    <i class="fa fa-wifi icon" data-bs-toggle="modal" data-bs-
target="#startwifi"> WIFI</i>
  </label>


<a href="{% url 'logout' %}" style="text-decoration: none;">
  <label>
    <!-- <input type="checkbox" name="check"> -->
    <span></span>
    <i class="fa fa-user icon"> LOGOUT</i>
  </label></a>

  <label>
    <input type="checkbox" name="check">
    <span data-bs-toggle="modal" data-bs-target="#login2"></span>
    <i class="fa fa-user icon" data-bs-toggle="modal" data-bs-target="#login2">
LOGIN</i>
  </label>

  <label>
    <input type="checkbox" name="check">
    <span data-bs-toggle="modal" data-bs-target="#help"></span>
    <i class="fa fa-moon icon" data-bs-toggle="modal" data-bs-target="#help">
HELP</i>
```

```
  </label>



</div><!-- mwisho wa buttons div -->



</div><!-- mwisho wa butons-container -->


<!-- KAMA USER AMIJISAJILI ZINAISHIA HAPA -->


{% else %}


<!-- KAMA USER HAJAJISAJILI ZINAANZIA HAPA -->


<div class="buttons-container">



<!-- <center>
    <h1 style="color: white;font-weight: bold;">SMART INVIGILATION<br>
SYSTEM</h1>
  </center> -->

<div class="buttons"><!-- mwanzo wa buttons-div -->

<label>
    <input type="checkbox" name="check">
    <span data-bs-toggle="modal" data-bs-target="#webcam"></span>
    <i class="fa fa-camera icon" data-bs-toggle="modal" data-bs-target="#webcam">
RECORD</i>
  </label>

  <label>
    <input type="checkbox" name="check">
```

```
    <span data-bs-toggle="modal" data-bs-target="#external"></span>
    <i class="fa fa-camera icon" data-bs-toggle="modal" data-bs-
target="#external"> CAMERA</i>
  </label>


  <label>
    <input type="checkbox" name="check">
    <span data-bs-toggle="modal" data-bs-target="#wifi"></span>
    <i class="fa fa-wifi icon" data-bs-toggle="modal" data-bs-target="#wifi">
WIFI</i>
  </label>



  <label>
    <!-- <input type="checkbox" name="check"> -->
    <span data-bs-toggle="modal" data-bs-target="#logout2"></span>
    <i class="fa fa-user icon" data-bs-toggle="modal" data-bs-target="#logout2">
LOGOUT</i>
  </label>

  <label>
    <input type="checkbox" name="check">
    <span data-bs-toggle="modal" data-bs-target="#login"></span>
    <i class="fa fa-user icon" data-bs-toggle="modal" data-bs-target="#login">
LOGIN</i>
  </label>

  <label>
    <input type="checkbox" name="check">
    <span data-bs-toggle="modal" data-bs-target="#help"></span>
    <i class="fa fa-moon icon" data-bs-toggle="modal" data-bs-target="#help">
HELP</i>
  </label>




</div><!-- mwisho wa buttons div -->



</div><!-- mwisho wa butons-container -->
```

```
<!-- KAMA USER HAJAJISAJILI ZINAISHIA HAPA -->



{% endif %}









<!-- ------------------------------------------------------------ -->

<!-- Modal  for LOGIN BUTTON -->


<div class="modal fade" id="login" data-bs-backdrop="static" data-bs-
keyboard="false" tabindex="-1" aria-labelledby="staticBackdropLabel" aria-
hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="staticBackdropLabel"> SMART INVIGILATION
SYSTEM </h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>
      </div>
      <div class="modal-body" style="font-size: 22px;">


         <div class="login">
    <h2>LOGIN HERE</h2>
    <form method="POST" action="{% url 'signin' %}">
        {% csrf_token %}

      <div class="user-box">
        <input type="email" name="email" required>
        <label>Email</label>
      </div>
```

```
      <div class="user-box">
        <input type="password" name="password" required>
        <label>Password</label>
      </div>
     <!--  <p><input type="checkbox">Remember me <span>Forgot?</span></p> -->

        <a href="" class="link-btn">
        <span></span>
        <span></span>
        <span></span>
        <span></span>



        <button type="submit"  style="background-color: green;color:
white;padding: 10px;padding-right: 20px;padding-left: 20px;font-size:
20px;border-radius: 10px;">Login</button>
      </a>

     <!--  <div class="options-02">
                    <p>You don't have an account? <a href="" style="font-size:
18px;color: lightblue;font-weight: bold;text-decoration: none;">Sign Up</a></p>
                </div> -->
    </form>
  </div>

</div>
<!-- mwisho wa modal-body -->


      <div class="modal-footer">
        <button type="button" class="btn btn-danger" data-bs-dismiss="modal"
style="font-size: 18px;">Close</button>




            </div>
    </div>
  </div>
</div>
<!--MWISHO WA  Modal  for LOGIN BUTTON -->
```

```
<!-- ------------------------------------------------------------ -->

<!-- Modal  for HELP BUTTON -->


<div class="modal fade" id="help" data-bs-backdrop="static" data-bs-
keyboard="false" tabindex="-1" aria-labelledby="staticBackdropLabel" aria-
hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">

      <div class="modal-header">
        <h5 class="modal-title" id="staticBackdropLabel">WELCOME TO SMART
INVIGILATION SYSTEM </h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>
      </div>


      <div class="modal-body" style="font-size: 18px;background-
color:rgb(5,5,49);color: white;font-family: sans-serif;">

        <ul>
          <li><span style="color: red">Step 1:</span> Login using a correct
password and username.</li>
          <li><span style="color: red">Step 2:</span> Click Camera button to
start invigilation by using Laptop or External  camera to capture student's
faces</li>
          <li><span style="color: red">Step 3:</span> Click Record button to
start recording all activities during exam</li>
          <li><span style="color: red">Step 4:</span> Click Wi-Fi button to start
invigilation by using Wi-fi camera to capture student's faces</li>
          <li><span style="color: red">Step 5:</span> Click Logout button to
logout from the system</li>
        </ul>
        </div>


      <div class="modal-footer">
        <button type="button" class="btn btn-danger" data-bs-dismiss="modal"
style="font-size: 18px;">Close</button>
        </div>
```

```
    </div>
  </div>
</div>
<!--MWISHO WA  Modal  for HELP BUTTON -->




<!-- ------------------------------------------------------------ -->

<!-- Modal  for WEBCAM NOT LOGIN BUTTON -->


<div class="modal fade" id="webcam" data-bs-backdrop="static" data-bs-
keyboard="false" tabindex="-1" aria-labelledby="staticBackdropLabel" aria-
hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">

      <div class="modal-header">
        <h5 class="modal-title" id="staticBackdropLabel">WELCOME TO SMART
INVIGILATION SYSTEM </h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>
      </div>


      <div class="modal-body" style="font-size: 18px;background-
color:rgb(5,5,49);color: white;font-family: sans-serif;">

        <ul>
         <li>Sorry!!!! <br> To start Invigilation of students you have to login
first.</li>
        </ul>
        </div>


      <div class="modal-footer">
```

```
        <button type="button" class="btn btn-danger" data-bs-dismiss="modal"
style="font-size: 18px;">Close</button>
        </div>




    </div>
  </div>
</div>
<!--MWISHO WA  Modal  for WEBCAM NOT LOGIN BUTTON -->




<!-- ------------------------------------------------------------ -->

<!-- Modal  for EXTERNAL NOT LOGIN BUTTON -->


<div class="modal fade" id="external" data-bs-backdrop="static" data-bs-
keyboard="false" tabindex="-1" aria-labelledby="staticBackdropLabel" aria-
hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">

      <div class="modal-header">
        <h5 class="modal-title" id="staticBackdropLabel">WELCOME TO SMART
INVIGILATION SYSTEM </h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>
      </div>


      <div class="modal-body" style="font-size: 18px;background-
color:rgb(5,5,49);color: white;font-family: sans-serif;">

        <ul>
         <li>Sorry!!!! <br> To start Invigilation of students you have to login
first.</li>
        </ul>
        </div>
```

```
      <div class="modal-footer">
        <button type="button" class="btn btn-danger" data-bs-dismiss="modal"
style="font-size: 18px;">Close</button>
        </div>




    </div>
  </div>
</div>
<!--MWISHO WA  Modal  for EXTERNAL NOT LOGIN BUTTON -->









<!-- ------------------------------------------------------------ -->

<!-- Modal  for WIFI NOT LOGIN BUTTON -->


<div class="modal fade" id="wifi" data-bs-backdrop="static" data-bs-
keyboard="false" tabindex="-1" aria-labelledby="staticBackdropLabel" aria-
hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">

      <div class="modal-header">
        <h5 class="modal-title" id="staticBackdropLabel">WELCOME TO SMART
INVIGILATION SYSTEM </h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>
      </div>


      <div class="modal-body" style="font-size: 18px;background-
color:rgb(5,5,49);color: white;font-family: sans-serif;">
```

```
        <ul>
         <li>Sorry!!!! <br> To start Invigilation of students you have to login
first.</li>
        </ul>
        </div>


      <div class="modal-footer">
        <button type="button" class="btn btn-danger" data-bs-dismiss="modal"
style="font-size: 18px;">Close</button>
        </div>




    </div>
  </div>
</div>
<!--MWISHO WA  Modal  for WIFI NOT LOGIN BUTTON -->







<!-- ------------------------------------------------------------ -->

<!-- Modal  for LOGOUT2 NOT LOGIN BUTTON -->


<div class="modal fade" id="logout2" data-bs-backdrop="static" data-bs-
keyboard="false" tabindex="-1" aria-labelledby="staticBackdropLabel" aria-
hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">

      <div class="modal-header">
        <h5 class="modal-title" id="staticBackdropLabel">WELCOME TO SMART
INVIGILATION SYSTEM </h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>
      </div>
```

```
      <div class="modal-body" style="font-size: 18px;background-
color:rgb(5,5,49);color: white;font-family: sans-serif;">

        <ul>
         <li>Sorry!!!! <br> You have to log in first.</li>
        </ul>
        </div>


      <div class="modal-footer">
        <button type="button" class="btn btn-danger" data-bs-dismiss="modal"
style="font-size: 18px;">Close</button>
        </div>




    </div>
  </div>
</div>
<!--MWISHO WA  Modal  for LOGOUT2 NOT LOGIN BUTTON -->







<!-- ------------------------------------------------------------ -->

<!-- Modal  for LOGIN2 NOT LOGIN BUTTON -->


<div class="modal fade" id="login2" data-bs-backdrop="static" data-bs-
keyboard="false" tabindex="-1" aria-labelledby="staticBackdropLabel" aria-
hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">

      <div class="modal-header">
        <h5 class="modal-title" id="staticBackdropLabel">WELCOME TO SMART
INVIGILATION SYSTEM </h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>
      </div>
```

```
      <div class="modal-body" style="font-size: 18px;background-
color:rgb(5,5,49);color: white;font-family: sans-serif;">

        <ul>
         <li>Sorry!!!! <br> You have already authenticated as {{user.username}}
in this system.</li>
        </ul>
        </div>


      <div class="modal-footer">
        <button type="button" class="btn btn-danger" data-bs-dismiss="modal"
style="font-size: 18px;">Close</button>
        </div>




    </div>
  </div>
</div>
<!--MWISHO WA  Modal  for LOGIN2 NOT LOGIN BUTTON -->









<!-- ------------------------------------------------------------ -->

<!--MWANZO WA  Modal   TO START RECORDING BUTTON -->


<div class="modal fade" id="record" data-bs-backdrop="static" data-bs-
keyboard="false" tabindex="-1" aria-labelledby="staticBackdropLabel" aria-
hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">

      <div class="modal-header">
        <h5 class="modal-title" id="staticBackdropLabel">WELCOME TO SMART
INVIGILATION SYSTEM </h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>
```

```
      </div>


      <div class="modal-body" style="font-size: 18px;background-
color:rgb(5,5,49);color: white;font-family: sans-serif;">

        <ul>
         <li>Hey {{user.username}}, <br> Are you sure you want to start Recording
a Video ???</li>
        </ul>
        </div>


      <div class="modal-footer">
        <a href="{% url 'recording_video_page' %}">
        <button type="button" class="btn btn-primary" data-bs-dismiss="modal"
style="font-size: 18px;margin-right: 20px;">Confirm</button></a>

        <button type="button" class="btn btn-danger" data-bs-dismiss="modal"
style="font-size: 18px;">Close</button>
        </div>




    </div>
  </div>
</div>
<!--MWISHO WA  Modal   TO START WEBCAM BUTTON -->




<!-- ------------------------------------------------------------ -->

<!--MWISHO WA  Modal   TO START EXTERNAL BUTTON -->


<div class="modal fade" id="startexternal" data-bs-backdrop="static" data-bs-
keyboard="false" tabindex="-1" aria-labelledby="staticBackdropLabel" aria-
hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
```

```
      <div class="modal-header">
        <h5 class="modal-title" id="staticBackdropLabel">WELCOME TO SMART
INVIGILATION SYSTEM </h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>
      </div>


      <div class="modal-body" style="font-size: 18px;background-
color:rgb(5,5,49);color: white;font-family: sans-serif;">

        <ul>
         <li>Hey {{user.username}}, <br> Are you sure you want to start
Invigilation of students???</li>
        </ul>
        </div>


      <div class="modal-footer">
        <a href="{% url 'starting_page' %}">
        <button type="button" class="btn btn-primary" data-bs-dismiss="modal"
style="font-size: 18px;margin-right: 20px;">Confirm</button></a>

        <button type="button" class="btn btn-danger" data-bs-dismiss="modal"
style="font-size: 18px;">Close</button>
        </div>




    </div>
  </div>
</div>
<!--MWISHO WA  Modal   TO START EXTERNAL BUTTON -->









<!-- ------------------------------------------------------------ -->

<!--MWISHO WA  Modal   TO START WIFI BUTTON -->
```

```
<div class="modal fade" id="startwifi" data-bs-backdrop="static" data-bs-
keyboard="false" tabindex="-1" aria-labelledby="staticBackdropLabel" aria-
hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">

      <div class="modal-header">
        <h5 class="modal-title" id="staticBackdropLabel">WELCOME TO SMART
INVIGILATION SYSTEM </h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>
      </div>


      <div class="modal-body" style="font-size: 18px;background-
color:rgb(5,5,49);color: white;font-family: sans-serif;">

        <ul>
         <li>Hey {{user.username}}, <br> Are you sure you want to start
Invigilation of students using Wi-Fi camera ???</li>
        </ul>
        </div>


      <div class="modal-footer">
        <a href="{% url 'wifi_page' %}">
        <button type="button" class="btn btn-primary" data-bs-dismiss="modal"
style="font-size: 18px;margin-right: 20px;">Confirm</button></a>

        <button type="button" class="btn btn-danger" data-bs-dismiss="modal"
style="font-size: 18px;">Close</button>
        </div>




    </div>
  </div>
</div>
<!--MWISHO WA  Modal   TO START WIFI BUTTON -->
```

```
</div><!-- mwisho wa container -->
</body>
</html>
{% endblock %}
```

- StartingPage.html file

```
{% extends 'SmartInvigilationApp/base.html' %}

{% load static %}

{% block content %}

<!DOCTYPE html>
<html>
<head>
    <title></title>
    <style type="text/css">
        *{
            margin: 0;
            padding: 0;
            box-sizing: border-box;
            font-family: 'poppins',sans-serif;
        }
        body{
            display: flex;
            justify-content: center;
            align-items: center;
            min-height: 100vh;
```

```
            background: #333;
        }
        .container{
            position: relative;
            display: flex;
            padding: 80px 40px;
            justify-content: center;
            align-items: center;
            flex-wrap: wrap;
            gap:80px;
            background-color: #333;
        }
        .card{
            position: relative;
            width: 350px;
            height: 180px;
            background: #333;
            transition: 0.5s;

        }
        .card:hover{
            height: 450px;

        }
        .card .lines{
            position: absolute;
            inset:0;
            background: #000;
            overflow: hidden;
        }
        .card .lines::before{
            content: '';
            position: absolute;
            top: 50%;
            left: 50%;
            width: 600px;
            height: 120px;
            background: linear-
gradient(transparent,#45f3ff,#45f3ff,#45f3ff,transparent);
            animation: animate 4s linear infinite;
            animation-play-state: paused;
        }
        .card:hover .lines::before{
            animation-play-state: running;
```

```
        }
        @keyframes animate{
            0%{
                transform: translate(-50%,-50%) rotate(0deg);
            }
            100%{
                transform: translate(-50%,-50%) rotate(360deg);
            }

        }

        .card .lines::after{
            content: '';
            position: absolute;
            inset:3px;
            background: #292929;

        }
        .card .imgBx{
            position: absolute;
            width: 150px;
            height: 150px;
            top: -60px;
            left: 50%;
            transform: translateX(-50%);
            background: #000;
            transition: 0.5s;
            z-index: 10;
            overflow: hidden;
            display: flex;
            justify-content: center;
            align-items: center;
        }
        .card:hover .imgBx{
            top: 25px;
            width: 200px;
            height: 200px;

        }
        .card .imgBx::before{
            content: '';
            position: absolute;
            top: 50%;
            left: 50%
            width:500px;
```

```
            height: 150px;
            transform: translate(-50%,-50%);
            background: linear-
gradient(transparent,#ff3c7b,#ff3c7b,#ff3c7b,transparent);
            animation: animate2 6s linear infinite;
            animation-play-state: paused;
        }
        .card hover .imgBx::before{
            animation-play-state: running;
        }
        @keyframes animate2{
            0%{
                transform: translate(-50%,-50%) rotate(360deg);
            }
            100%{
                transform: translate(-50%,-50%) rotate(0deg);
            }

        }

        .card .imgBx::after{
            content: '';
            position: absolute;
            inset:3px;
            background: #292929;

        }

        .card .imgBx img{
            position: absolute;
            width: 100px;
            z-index: 1;
            /*filter: invert(1);*/
            opacity: 1;
            transition: 0.5s;


        }
        .card:hover .imgBx img{
            opacity: 1;
        }
        .card .content{
            position: absolute;
            width: 100%;
            height: 100%;
```

```
        display: flex;
        justify-content: center;
        align-items: flex-end;
        overflow: hidden;
    }
    .card .content .details{
        padding: 30px 20px;
        text-align: center;
        width: 100%;
        transition: 0.5s;
        transform: translateY(145px);
    }
    .card:hover .content .details{
        transform: translateY(0px);

    }
    .card .content .details h2{
        font-size: 1.5rem;
        font-weight: 500;
        color: #45f3ff;
        line-height: 1.2em;
    }
    .card .content .details input{
        color: black;
        opacity: 0;
        transition: 0.5s;
        border-radius: 10px;
        padding: 10px;
        font-weight: bold;

    }

    .card .content .details a{
        display: inline-block;
        padding: 8px 15px;
        background: #45f3ff;
        color: #292929;
        margin-top: 10px;
        font-weight: 500;
        text-decoration: none;
        opacity: 0;
        transition: 0.5s;
    }

    .card:hover .content .details input,
```

```
        .card:hover .content .details a
        {
            opacity: 1;

        }
        ::placeholder{
            color: red;
        }


    </style>
</head>
<body>

<form method="POST" action="{% url 'SmartInvigilationProject' %}">
        {% csrf_token %}

        <center>
             {% if messages %}
{% for message in messages %}


          <div class="alert alert-warning alert-dismissible fade show"
role="alert">
  <strong style="color: red;font-size: 20px;">{{message}}</strong>
  <button type="button" class="btn-close" data-bs-dismiss="alert" aria-
label="Close"></button>
</div>
    {% endfor %}
  {% endif %}
        </center>


        <h1 style="color: white;font-family: sans-serif;font-size: 22px;text-
align: center;margin-top:20px;">INVIGILATOR INTERFACE</h1>

<div class="container">
```

```
<div class="card"><!-- mwanzo wa card -->

        <div class="lines"></div><!-- div ya lines -->


        <div class="imgBx"><!-- mwanzo wa imgBx -->
            <img src="{% static 'SmartInvigilationApp/assets/img/1q.jpeg' %}">
        </div><!-- mwisho wa imgBx -->


        <div class="content"><!-- mwanzo wa content -->

            <div class="details"><!-- mwanzo wa details -->
                <h2>Camera</h2>
                <p>
  <input type="text" value="/videos/S1.mp4" class="inputi" name="camera_no"
placeholder="Enter Camera Number" required>
</p>
    <a href="#" style="background-color: #333;color: #333;">Read
More</a>
            </div><!-- mwisho wa details -->
        </div><!-- mwisho wa content -->




    </div><!-- mwisho wa card -->









<div class="card"><!-- mwanzo wa card -->

        <div class="lines"></div><!-- div ya lines -->
```

```
        <div class="imgBx"><!-- mwanzo wa imgBx -->
            <img src="{% static 'SmartInvigilationApp/assets/img/3q.jpeg' %}">
        </div><!-- mwisho wa imgBx -->


        <div class="content"><!-- mwanzo wa content -->

            <div class="details"><!-- mwanzo wa details -->
                <h2>START</h2>
                <p style="color: white">
                    Hey <span style="color: red;font-weight:
bold;">{{user.username}}</span>, now you can turn on a Camera to start
Invigilation.
                </p>
                <button style="border-radius: 10px;padding: 10px;background:
#45f3ff;color: black;font-size: 22px;">Start</button>

            </div><!-- mwisho wa details -->
        </div><!-- mwisho wa content -->




    </div><!-- mwisho wa card -->







</div>
</form>
</body>
</html>

{% endblock %}
```

❖ RecordingVideoPage.html

```
❖ {% extends 'SmartInvigilationApp/base.html' %}
❖
❖ {% load static %}
❖
❖ {% block content %}
❖
❖ <!DOCTYPE html>
❖ <html>
❖ <head>
❖     <title></title>
❖     <style type="text/css">
❖         *{
❖             margin: 0;
❖             padding: 0;
❖             box-sizing: border-box;
❖             font-family: 'poppins',sans-serif;
❖         }
❖         body{
❖             display: flex;
❖             justify-content: center;
❖             align-items: center;
❖             min-height: 100vh;
❖             background: #333;
❖         }
❖         .container{
❖             position: relative;
❖             display: flex;
❖             padding: 80px 40px;
❖             justify-content: center;
❖             align-items: center;
❖             flex-wrap: wrap;
❖             gap:80px;
❖             background-color: #333;
❖         }
❖         .card{
❖             position: relative;
❖             width: 350px;
❖             height: 180px;
❖             background: #333;
❖             transition: 0.5s;
❖
❖         }
❖         .card:hover{
❖             height: 450px;
```

```
❖
❖         }
❖         .card .lines{
❖             position: absolute;
❖             inset:0;
❖             background: #000;
❖             overflow: hidden;
❖         }
❖         .card .lines::before{
❖             content: '';
❖             position: absolute;
❖             top: 50%;
❖             left: 50%;
❖             width: 600px;
❖             height: 120px;
❖             background: linear-
   gradient(transparent,#45f3ff,#45f3ff,#45f3ff,transparent);
❖             animation: animate 4s linear infinite;
❖             animation-play-state: paused;
❖         }
❖         .card:hover .lines::before{
❖             animation-play-state: running;
❖
❖         }
❖         @keyframes animate{
❖             0%{
❖                 transform: translate(-50%,-50%) rotate(0deg);
❖             }
❖             100%{
❖                 transform: translate(-50%,-50%) rotate(360deg);
❖             }
❖
❖         }
❖
❖         .card .lines::after{
❖             content: '';
❖             position: absolute;
❖             inset:3px;
❖             background: #292929;
❖
❖         }
❖         .card .imgBx{
❖             position: absolute;
❖             width: 150px;
❖             height: 150px;
```

```
❖             top: -60px;
❖             left: 50%;
❖             transform: translateX(-50%);
❖             background: #000;
❖             transition: 0.5s;
❖             z-index: 10;
❖             overflow: hidden;
❖             display: flex;
❖             justify-content: center;
❖             align-items: center;
❖         }
❖         .card:hover .imgBx{
❖             top: 25px;
❖             width: 200px;
❖             height: 200px;
❖
❖         }
❖         .card .imgBx::before{
❖             content: '';
❖             position: absolute;
❖             top: 50%;
❖             left: 50%
❖             width:500px;
❖             height: 150px;
❖             transform: translate(-50%,-50%);
❖             background: linear-
  gradient(transparent,#ff3c7b,#ff3c7b,#ff3c7b,transparent);
❖             animation: animate2 6s linear infinite;
❖             animation-play-state: paused;
❖         }
❖         .card hover .imgBx::before{
❖             animation-play-state: running;
❖         }
❖         @keyframes animate2{
❖             0%{
❖                 transform: translate(-50%,-50%) rotate(360deg);
❖             }
❖             100%{
❖                 transform: translate(-50%,-50%) rotate(0deg);
❖             }
❖
❖         }
❖
❖         .card .imgBx::after{
❖             content: '';
```

```
            position: absolute;
            inset:3px;
            background: #292929;

        }

        .card .imgBx img{
            position: absolute;
            width: 100px;
            z-index: 1;
            /*filter: invert(1);*/
            opacity: 1;
            transition: 0.5s;

        }
        .card:hover .imgBx img{
            opacity: 1;
        }
        .card .content{
            position: absolute;
            width: 100%;
            height: 100%;
            display: flex;
            justify-content: center;
            align-items: flex-end;
            overflow: hidden;
        }
        .card .content .details{
            padding: 30px 20px;
            text-align: center;
            width: 100%;
            transition: 0.5s;
            transform: translateY(145px);
        }
        .card:hover .content .details{
            transform: translateY(0px);

        }
        .card .content .details h2{
            font-size: 1.5rem;
            font-weight: 500;
            color: #45f3ff;
            line-height: 1.2em;
        }
        .card .content .details input{
```

```
            color: black;
            opacity: 0;
            transition: 0.5s;
            border-radius: 10px;
            padding: 10px;
            font-weight: bold;

        }

        .card .content .details a{
            display: inline-block;
            padding: 8px 15px;
            background: #45f3ff;
            color: #292929;
            margin-top: 10px;
            font-weight: 500;
            text-decoration: none;
            opacity: 0;
            transition: 0.5s;
        }

        .card:hover .content .details input,
        .card:hover .content .details a
        {
            opacity: 1;

        }
        ::placeholder{
            color: red;
        }

    </style>
</head>
<body>

<form method="POST" action="{% url 'record_video' %}">
        {% csrf_token %}

        <center>
             {% if messages %}
{% for message in messages %}


          <div class="alert alert-warning alert-dismissible fade show"
role="alert">
```

```
❖    <strong style="color: red;font-size: 20px;">{{message}}</strong>
❖    <button type="button" class="btn-close" data-bs-dismiss="alert" aria-
   label="Close"></button>
❖  </div>
❖      {% endfor %}
❖    {% endif %}
❖          </center>
❖
❖          <h1 style="color: white;font-family: sans-serif;font-size:
   22px;text-align: center;margin-top:20px;">INVIGILATOR INTERFACE - TAKING
   LIVE VIDEO</h1>
❖
❖  <div class="container">
❖
❖
❖
❖
❖
❖  <div class="card"><!-- mwanzo wa card -->
❖
❖          <div class="lines"></div><!-- div ya lines -->
❖
❖          <div class="imgBx"><!-- mwanzo wa imgBx -->
❖              <img src="{% static 'SmartInvigilationApp/assets/img/1q.jpeg'
   %}">
❖          </div><!-- mwisho wa imgBx -->
❖
❖          <div class="content"><!-- mwanzo wa content -->
❖
❖              <div class="details"><!-- mwanzo wa details -->
❖                  <h2>Camera</h2>
❖                  <p>
❖    <input type="text" value="/videos/S1.mp4" class="inputi"
   name="camera_no" placeholder="Enter Camera Number" required>
❖  </p>
❖      <a href="#" style="background-color: #333;color: #333;">Read
   More</a>
❖              </div><!-- mwisho wa details -->
```

```
        </div><!-- mwisho wa content -->
    </div><!-- mwisho wa card -->
<div class="card"><!-- mwanzo wa card -->
        <div class="lines"></div><!-- div ya lines -->
        <div class="imgBx"><!-- mwanzo wa imgBx -->
            <img src="{% static 'SmartInvigilationApp/assets/img/3q.jpeg' %}">
        </div><!-- mwisho wa imgBx -->
        <div class="content"><!-- mwanzo wa content -->
            <div class="details"><!-- mwanzo wa details -->
                <h2>START</h2>
                <p style="color: white">
                    Are you sure you want to start Invigilation ??.
                </p>
                <button style="border-radius: 10px;padding: 10px;background: #45f3ff;color: black;font-size: 22px;">Start</button>
            </div><!-- mwisho wa details -->
        </div><!-- mwisho wa content -->
    </div><!-- mwisho wa card -->
```

```
❖
❖  </div>
❖  </form>
❖  </body>
❖  </html>
❖
❖  {% endblock %}
```

- WifiPage.html

```
{% extends 'SmartInvigilationApp/base.html' %}

{% load static %}

{% block content %}

<!DOCTYPE html>
<html>
<head>
    <title></title>
    <style type="text/css">
        *{
            margin: 0;
            padding: 0;
            box-sizing: border-box;
            font-family: 'poppins',sans-serif;
        }
        body{
            display: flex;
            justify-content: center;
            align-items: center;
            min-height: 100vh;
            background: #333;
        }
        .container{
            position: relative;
            display: flex;
            padding: 80px 40px;
            justify-content: center;
            align-items: center;
```

```
            flex-wrap: wrap;
            gap:80px;
            background-color: #333;
        }
        .card{
            position: relative;
            width: 350px;
            height: 180px;
            background: #333;
            transition: 0.5s;

        }
        .card:hover{
            height: 450px;

        }
        .card .lines{
            position: absolute;
            inset:0;
            background: #000;
            overflow: hidden;
        }
        .card .lines::before{
            content: '';
            position: absolute;
            top: 50%;
            left: 50%;
            width: 600px;
            height: 120px;
            background: linear-
gradient(transparent,#45f3ff,#45f3ff,#45f3ff,transparent);
            animation: animate 4s linear infinite;
            animation-play-state: paused;
        }
        .card:hover .lines::before{
            animation-play-state: running;

        }
        @keyframes animate{
            0%{
                transform: translate(-50%,-50%) rotate(0deg);
            }
            100%{
                transform: translate(-50%,-50%) rotate(360deg);
            }
```

```
        }

        .card .lines::after{
            content: '';
            position: absolute;
            inset:3px;
            background: #292929;

        }
        .card .imgBx{
            position: absolute;
            width: 150px;
            height: 150px;
            top: -60px;
            left: 50%;
            transform: translateX(-50%);
            background: #000;
            transition: 0.5s;
            z-index: 10;
            overflow: hidden;
            display: flex;
            justify-content: center;
            align-items: center;
        }
        .card:hover .imgBx{
            top: 25px;
            width: 200px;
            height: 200px;

        }
        .card .imgBx::before{
            content: '';
            position: absolute;
            top: 50%;
            left: 50%
            width:500px;
            height: 150px;
            transform: translate(-50%,-50%);
            background: linear-
gradient(transparent,#ff3c7b,#ff3c7b,#ff3c7b,transparent);
            animation: animate2 6s linear infinite;
            animation-play-state: paused;
        }
        .card hover .imgBx::before{
```

```
        animation-play-state: running;
    }
    @keyframes animate2{
        0%{
            transform: translate(-50%,-50%) rotate(360deg);
        }
        100%{
            transform: translate(-50%,-50%) rotate(0deg);
        }

    }

    .card .imgBx::after{
        content: '';
        position: absolute;
        inset:3px;
        background: #292929;

    }

    .card .imgBx img{
        position: absolute;
        width: 100px;
        z-index: 1;
        /*filter: invert(1);*/
        opacity: 1;
        transition: 0.5s;


    }
    .card:hover .imgBx img{
        opacity: 1;
    }
    .card .content{
        position: absolute;
        width: 100%;
        height: 100%;
        display: flex;
        justify-content: center;
        align-items: flex-end;
        overflow: hidden;
    }
    .card .content .details{
        padding: 30px 20px;
        text-align: center;
```

```
            width: 100%;
            transition: 0.5s;
            transform: translateY(145px);
        }
        .card:hover .content .details{
            transform: translateY(0px);

        }
        .card .content .details h2{
            font-size: 1.5rem;
            font-weight: 500;
            color: #45f3ff;
            line-height: 1.2em;
        }
        .card .content .details input{
            color: black;
            opacity: 0;
            transition: 0.5s;
            border-radius: 10px;
            padding: 10px;
            font-weight: bold;

        }

        .card .content .details a{
            display: inline-block;
            padding: 8px 15px;
            background: #45f3ff;
            color: #292929;
            margin-top: 10px;
            font-weight: 500;
            text-decoration: none;
            opacity: 0;
            transition: 0.5s;
        }

        .card:hover .content .details input,
        .card:hover .content .details a
        {
            opacity: 1;

        }
        ::placeholder{
            color: red;
        }
```

```
    </style>
</head>
<body>

<form method="POST" action="{% url 'SmartInvigilationProject' %}">
        {% csrf_token %}

        <center>
             {% if messages %}
{% for message in messages %}


          <div class="alert alert-warning alert-dismissible fade show"
role="alert">
  <strong style="color: red;font-size: 20px;">{{message}}</strong>
  <button type="button" class="btn-close" data-bs-dismiss="alert" aria-
label="Close"></button>
</div>
    {% endfor %}
  {% endif %}
        </center>


        <h1 style="color: white;font-family: sans-serif;font-size: 22px;text-
align: center;margin-top:20px;">INVIGILATOR INTERFACE <br>
    <span style="color: red;margin-top: 25px;font-size: 20px;">WI-FI
CAMERA</span>  </h1>

<div class="container">
```

```
<div class="card"><!-- mwanzo wa card -->

        <div class="lines"></div><!-- div ya lines -->


        <div class="imgBx"><!-- mwanzo wa imgBx -->
            <img src="{% static 'SmartInvigilationApp/assets/img/1q.jpeg' %}">
        </div><!-- mwisho wa imgBx -->


        <div class="content"><!-- mwanzo wa content -->

            <div class="details"><!-- mwanzo wa details -->
                <h2>Camera Address</h2>
                <p>
  <input type="text" value="https://192.168.1.40" class="inputi" name="camera_no"
placeholder="Enter Camera Ip address" required>
</p>
    <a href="#" style="background-color: #333;color: #333;">Read
More</a>
            </div><!-- mwisho wa details -->
        </div><!-- mwisho wa content -->




    </div><!-- mwisho wa card -->









<div class="card"><!-- mwanzo wa card -->

        <div class="lines"></div><!-- div ya lines -->


        <div class="imgBx"><!-- mwanzo wa imgBx -->
            <img src="{% static 'SmartInvigilationApp/assets/img/3q.jpeg' %}">
        </div><!-- mwisho wa imgBx -->
```

```
        <div class="content"><!-- mwanzo wa content -->

            <div class="details"><!-- mwanzo wa details -->
                <h2>START</h2>
                <p style="color: white">
                    Hey <span style="color: red;font-weight:
bold;">{{user.username}}</span>, now you can turn on a Camera to start
Invigilation.
                </p>
                <button style="border-radius: 10px;padding: 10px;background:
#45f3ff;color: black;font-size: 22px;">Start</button>

            </div><!-- mwisho wa details -->
        </div><!-- mwisho wa content -->




    </div><!-- mwisho wa card -->









</div>
</form>
</body>
</html>

{% endblock %}
```