

# Senior Design Project 2023- 2024

Mandana Zooyousefin

Muğla Sıtkı Koçman University

Advisor:

DR.Tuğba ÖNAL SÜZEK

Department of computer engineering



## Automatic Attendance System Using Image Processing





# Goal

- Create web application for attendance system using face

# Libraries

OpenCV

Face  
Recognition





# How does the codes work

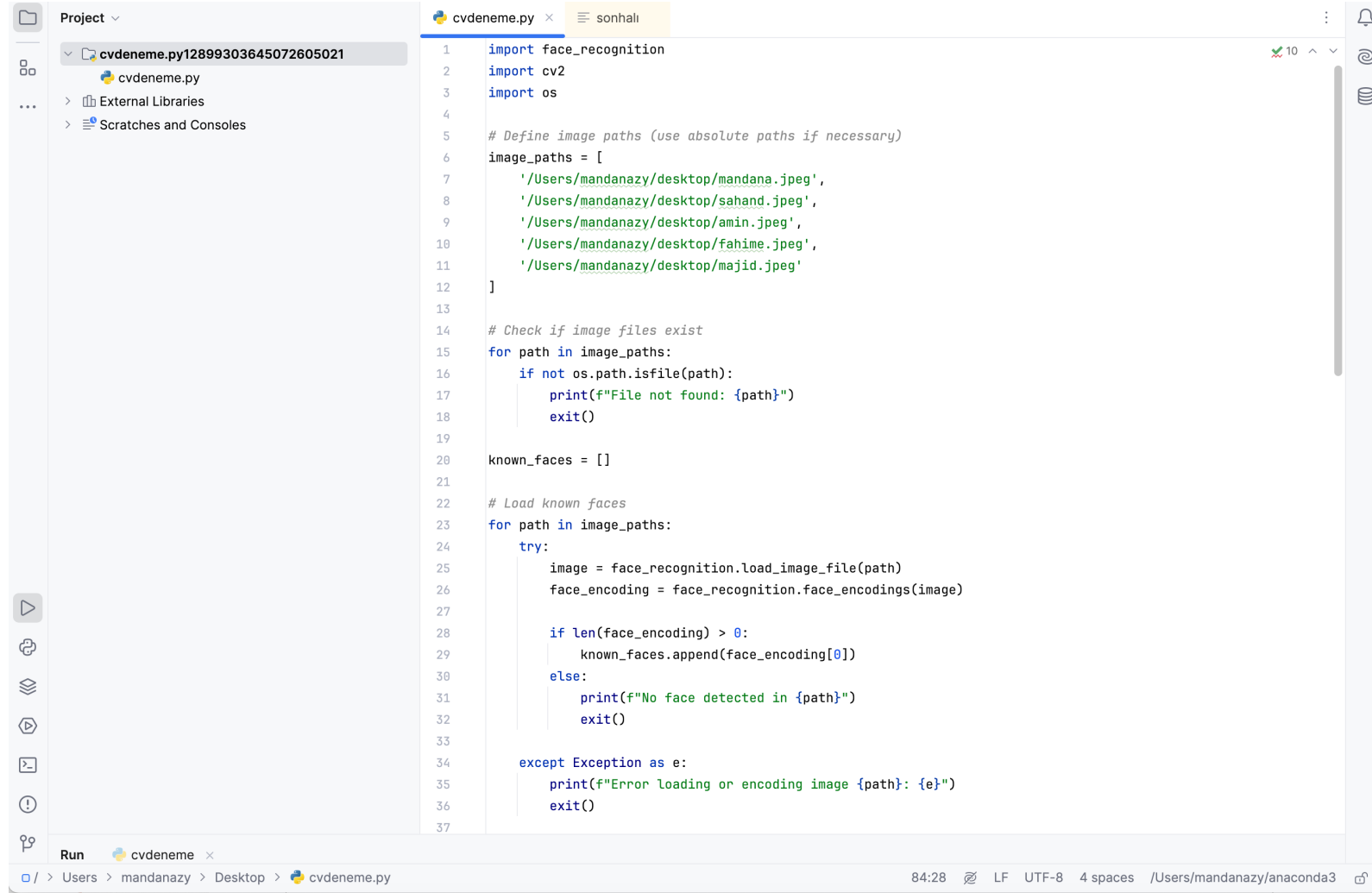
For this attendance system, the codes, recognize the face that show to the camera and and find the face between photoes of students

---

# Language

- PyCharm





```
1 import face_recognition
2 import cv2
3 import os
4
5 # Define image paths (use absolute paths if necessary)
6 image_paths = [
7     '/Users/mandanazy/desktop/mandana.jpeg',
8     '/Users/mandanazy/desktop/sahand.jpeg',
9     '/Users/mandanazy/desktop/amin.jpeg',
10    '/Users/mandanazy/desktop/fahime.jpeg',
11    '/Users/mandanazy/desktop/majid.jpeg'
12 ]
13
14 # Check if image files exist
15 for path in image_paths:
16     if not os.path.isfile(path):
17         print(f"File not found: {path}")
18         exit()
19
20 known_faces = []
21
22 # Load known faces
23 for path in image_paths:
24     try:
25         image = face_recognition.load_image_file(path)
26         face_encoding = face_recognition.face_encodings(image)
27
28         if len(face_encoding) > 0:
29             known_faces.append(face_encoding[0])
30     except:
31         print(f"No face detected in {path}")
32         exit()
33
34 except Exception as e:
35     print(f"Error loading or encoding image {path}: {e}")
36     exit()
37
```

Run cvdeneme

/ > Users > mandanazy > Desktop > cvdeneme.py 84:28 LF UTF-8 4 spaces /Users/mandanazy/anaconda3

Demo  
codes

```

mirror_mod = modifier_ob.modifiers[0]
#set mirror object to mirror
mirror_mod.mirror_object = mirror_ob

#operation == "MIRROR_X":
mirror_mod.use_x = True
mirror_mod.use_y = False
mirror_mod.use_z = False
#operation == "MIRROR_Y":
mirror_mod.use_x = False
mirror_mod.use_y = True
mirror_mod.use_z = False
#operation == "MIRROR_Z":
mirror_mod.use_x = False
mirror_mod.use_y = False
mirror_mod.use_z = True

#selection at the end -add
mirror_ob.select= 1
modifier_ob.select=1
context.scene.objects.active = mirror_ob
print("Selected" + str(modifier_ob.name))
mirror_ob.select = 0
one = bpy.context.selected_objects[0]
data.objects[one.name].select=True

print("please select exactly one object")

--- OPERATOR CLASSES ---

bpy.types.Operator(
    name="X mirror to the selected object",
    bl_idname="object.mirror_mirror_x",
    bl_label="Mirror X"
):
    """Mirror X"""

    @classmethod
    def poll(cls, context):
        if context.active_object is not None:

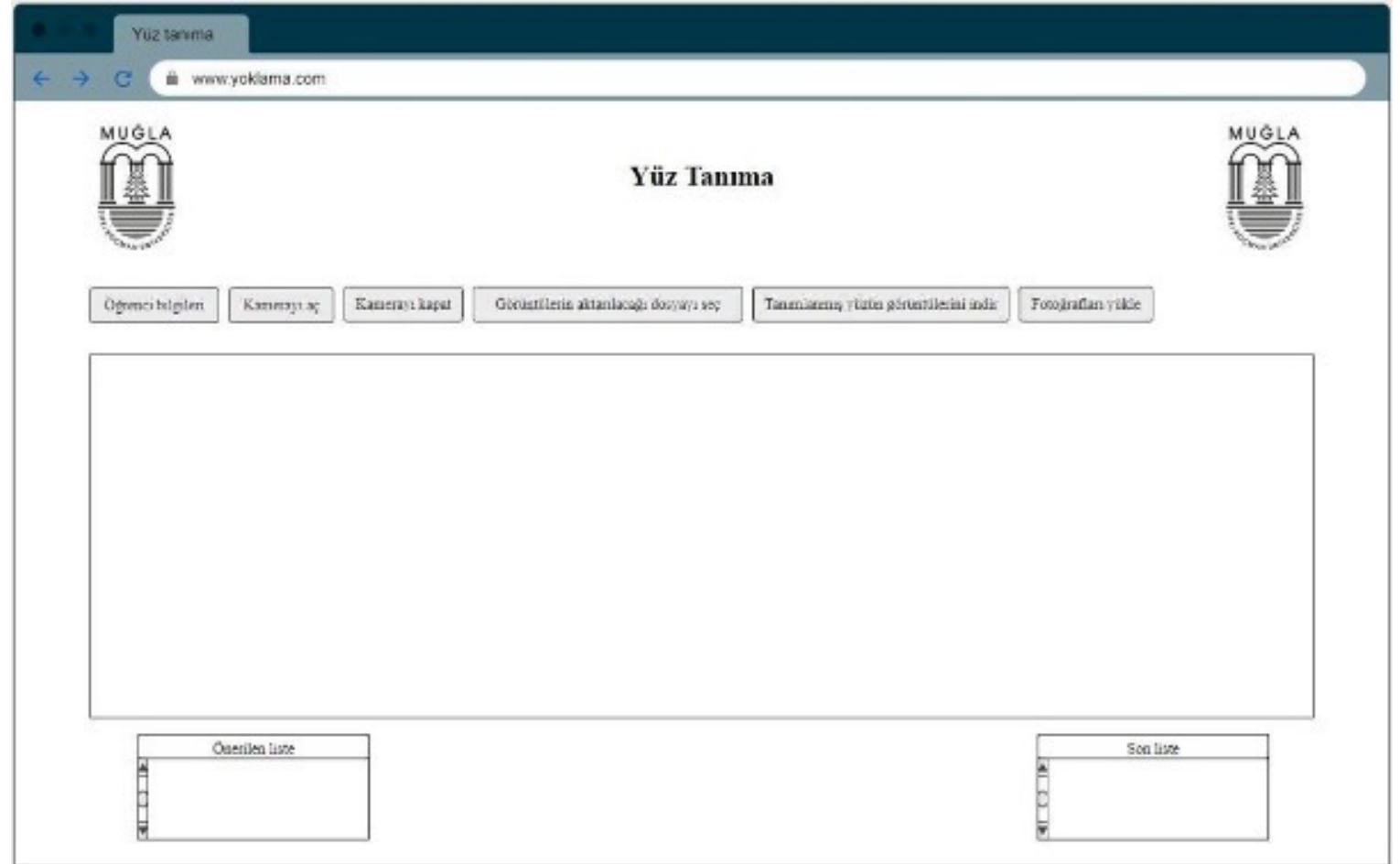
```

## Code Description

- Five images are specified and saved. When the code starts working, the camera is turned on and records the face image, and then recognizes the face from the saved images.



# Graphical Usage Interface



## Student's registering system

Öğrenci Kayıt

www.yokdama.com/kayit

MUGLA  
Öğrenci Kayıt Sistemi

Öğrenci adı/soyadı

E-posta

Öğrenci Numarası

11/20/2023

Yüz tanıma sistemi için  
Fotoğrafınızı ekleyin  
veya  
Kamerayı açın

+ K

GERİ DÖN KAYIT

A hand is visible in the background, reaching towards a computer mouse. In the foreground, a large, three-dimensional '@' symbol made of cardboard stands on a wooden surface. The entire scene is overlaid with a purple-to-orange gradient.

Thank you!

Email:  
[zooyousefinmandana@gmail.com](mailto:zooyousefinmandana@gmail.com)