**APPENDIX**

REQUIRMENTS.TXT

absl-py==1.3.0

anvil==0.0.2

astunparse==1.6.3

boto3==1.26.16

botocore==1.29.16

cachetools==5.2.0

certifi==2022.9.24

charset-normalizer==2.1.1

contourpy==1.0.6

cycler==0.11.0

facenet-pytorch==2.5.2

flatbuffers==22.11.23

fonttools==4.38.0

gast==0.4.0

google-auth==2.14.1

google-auth-oauthlib==0.4.6

google-pasta==0.2.0

grpcio==1.50.0

h5py==3.7.0

idna==3.4

importlib-metadata==5.1.0

Jinja2==3.1.2

jmespath==1.0.1

keras==2.11.0

keras-vggface==0.6

kiwisolver==1.4.4

libclang==14.0.6

Markdown==3.4.1

MarkupSafe==2.1.1

matplotlib==3.6.2

mtcnn==0.1.1

numpy==1.23.5

oauthlib==3.2.2

opencv-python==4.6.0.66

opt-einsum==3.3.0

packaging==21.3

pandas==1.5.2

Pillow==9.3.0

protobuf==3.19.6

pyasn1==0.4.8

pyasn1-modules==0.2.8

pyparsing==3.0.9

python-dateutil==2.8.2

pytz==2022.6

PyYAML==6.0

requests==2.28.1

requests-oauthlib==1.3.1

rsa==4.9

s3transfer==0.6.0

scipy==1.9.3

six==1.16.0

tensorboard==2.11.0

tensorboard-data-server==0.6.1

tensorboard-plugin-wit==1.8.1

tensorflow==2.11.0

tensorflow-estimator==2.11.0

tensorflow-io-gcs-filesystem==0.28.0

termcolor==2.1.1

torch==1.13.0

torchvision==0.14.0

typing\_extensions==4.4.0

urllib3==1.26.13

Werkzeug==2.2.2

wrapt==1.14.1 zipp==3.11.0

APP.PY

from flask import Flask, render\_template,jsonify,request

app = Flask(\_\_name\_\_)

from PIL import Image

from io import BytesIO

import base64

import cv2

# from matplotlib import pyplot as plt

from mtcnn.mtcnn import MTCNN

from matplotlib.patches import Rectangle

import warnings

warnings.simplefilter('ignore')

detector = MTCNN()

from numpy import asarray

from PIL import Image

import warnings

warnings.filterwarnings('ignore')

from facenet\_pytorch import MTCNN, InceptionResnetV1

import torch

from torchvision import datasets

from torch.utils.data import DataLoader

from PIL import Image

import boto3

import botocore

from PIL import Image

import pandas as pd

import io

import os

from twilio.rest import Client

from threading import Thread

mtcnn = MTCNN(image\_size=240, margin=0, min\_face\_size=20)

resnet = InceptionResnetV1(pretrained='vggface2').eval()

def face\_match(img\_path, data\_path):

# img\_path= location of photo, data\_path= location of data.pt

#getting embedding matrix of the given img

img = Image.open(img\_path).convert('RGB')

mtcnn = MTCNN(image\_size=240, margin=0, min\_face\_size=20)

resnet = InceptionResnetV1(pretrained='vggface2').eval()

face, prob = mtcnn(img, return\_prob=True) # returns cropped face and probability

try:

emb = resnet(face.unsqueeze(0)).detach() # detech is to make required gradient false

saved\_data = torch.load('data.pt') # loading data.pt file

embedding\_list = saved\_data[0] # getting embedding data

name\_list = saved\_data[1] # getting list of names

dist\_list = [] # list of matched distances, minimum distance is used to identify the person

for idx, emb\_db in enumerate(embedding\_list):

dist = torch.dist(emb, emb\_db).item()

dist\_list.append(dist)

idx\_min = dist\_list.index(min(dist\_list))

# print(dist\_list)

return (name\_list[idx\_min], min(dist\_list))

except Exception as e:

# raise e

return None

def run\_script():

file = open(r'/path/to/your/training.py', 'r').read()

return exec(file)

def run\_func():

data = { 'some': 'data', 'any': 'data' }

thr = Thread(target=run\_script, args=[app, data])

thr.start()

return thr

@app.route("/")

def index():

return render\_template("title.html")

@app.route('/new\_user')

def new\_user():

return render\_template('register.html')

@app.route('/capture\_image')

def capture\_image():

return render\_template('index.html')

@app.route('/verify\_image',methods=['POST'])

def verify\_image():

# print(request.form)

# data=request.form.to\_dict()

image\_data=request.form.get('filename').split(",")[1]

# print(image\_data)

# image\_data = bytes(image\_data, encoding="ascii")

im = Image.open(BytesIO(base64.b64decode(image\_data)))

im.save('image.png')

response=face\_match('image.png',"data.pt")

print(response)

if response==None:

return render\_template('faceVerification.html')

else:

return render\_template('enterpass.html',res={"user":response})

return jsonify(response)

#register

@app.route('/register',methods=['POST'])

def register():

image\_data=request.form.get('filename').split(",")[1]

# print(image\_data)

user=request.form.get('username')

# image\_data = bytes(image\_data, encoding="ascii")

im = Image.open(BytesIO(base64.b64decode(image\_data)))

os.makedirs('./images/'+user)

im.save('./images/'+user+'/image.png')

# run\_func()

return render\_template('RegisterSuccess.html')

@app.route('/verify\_pass',methods=['POST'])

def verify\_pass():

data=request.form.to\_dict()

user=data['user']

pwd=data['psw']

if user=='Radha Krishna'and pwd=='0801':

return render\_template('payment.html')

else:

return render\_template('verificationfailed.html')

return jsonify(request.form.to\_dict())

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug = True)

TITLE.HTML

<html>

<head>

<meta name="viewport" content="with=device-width, initial-scale=1.0">

<title>Customer Type</title>

<h1>CLOUD BASED ON POS</h1>

<link rel="stylesheet" href="{{url\_for('static', filename='1.css')}}">

</head>

<body>

<h2>SELECT CUSTOMER TYPE</h2>

<section>

<div class="new">

<a href="{{url\_for('new\_user')}}"class="new-user"> NEW USER</a>

</div>

<div class="existing">

<a href="{{url\_for('capture\_image')}}" class="existing-user">EXISTING USER</a>

</div>

</section>

</body>

</html>

REGISTER.HTML

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<!-- <link rel="stylesheet" href="{{url\_for('static',filename='1.css')}}"> -->

<title>Camera</title>

<script type="text/javascript" src="https://unpkg.com/webcam-easy@1.0.5/dist/webcam-easy.min.js"></script>

<style>

a{

padding: 10px;

background-color: orange;

color: white;

text-decoration: none;

} canvas{

display: none;

} button{

cursor: pointer;

} body{

display: flex;

justify-content: center;

align-items: center;

flex-direction: column;

}

.username{

width: 100%;

padding: 15px;

margin: 5px 0 22px 0;

/\* display: inline-block;\*/

border: none;

background: #f1f1f1;

}

</style>

</head>

<body>

<video id="webCam" autoplay playsinlne width="400" height="400"></video>

<canvas id="canvas"></canvas>

<img src="" />

<br/>

<button download id="capture" onClick="takeAPicture()">capture</button>

<form id="myAwesomeForm" method="post" action="{{ url\_for('register') }}">

<input type="text" id="username" class="username" name="username" placeholder="enter your name" hidden />

<input type="text" id="filename" name="filename" hidden/>

<label for="psw"><b>Password</b></label>

<input type="password" placeholder="Enter Password" name="psw" id="psw" required><hr>

<label for="psw-repeat"><b>Repeat Password</b></label>

<input type="password" placeholder="Repeat Password" name="psw-repeat" id="psw-repeat" required>

<hr>

<button type="submit" id="registerbtn">CREATE PASSWORD</button> <!-- Filename -->

<button type="submit" id="submit" hidden>Sumbit</button>

</form>

<script>

const webCamElement = document.getElementById("webCam");

const canvasElement = document.getElementById("canvas");

const webcam = new Webcam(webCamElement,"user",canvasElement);

webcam.start();

function takeAPicture()

{

let vcanvas=document.getElementById('webCam')

console.log(vcanvas.style.display)

if(vcanvas.style.display!=='none')

{

let picture =webcam.snap();

document.getElementById('webCam').style.display="none"

document.querySelector("img").src =picture;

document.querySelector("img").height = "400";

document.querySelector("img").width = "400";

document.getElementById("filename").value=picture;

document.getElementById("capture").innerHTML ="Recapture";

document.getElementById("submit").style.display="block"

document.getElementById("username").style.display="block"

}

else{

document.getElementById('webCam').style.display="block"

document.querySelector("img").src ="";

document.querySelector("img").height = "0";

document.querySelector("img").width = "0";

document.getElementById("capture").innerHTML ="capture";

document.getElementById("submit").style.display="none"} }</script>

</body>

</html>

INDEX.HTML

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Camera</title>

<script type="text/javascript" src="https://unpkg.com/webcam-easy@1.0.5/dist/webcam-easy.min.js"></script>

<style>

a{

padding: 10px;

background-color: orange;

color: white;

text-decoration: none;

}

canvas{

display: none;

}

button{

cursor: pointer;

}

body{

display: flex;

justify-content: center;

align-items: center;

flex-direction: column;

}

</style>

</head>

<body>

<video id="webCam" autoplay playsinlne width="400" height="400"></video>

<canvas id="canvas"></canvas>

<img src="" />

<br/>

<button download id="capture" onClick="takeAPicture()">capture</button>

<form id="myAwesomeForm" method="post" action="{{ url\_for('verify\_image') }}">

<input type="text" id="filename" name="filename" hidden/> <!-- Filename -->

<button type="submit" id="submit" hidden>Sumbit</button>

</form>

<script>

const webCamElement = document.getElementById("webCam");

const canvasElement = document.getElementById("canvas");

const webcam = new Webcam(webCamElement,"user",canvasElement);

webcam.start();

function takeAPicture()

{

let vcanvas=document.getElementById('webCam')

console.log(vcanvas.style.display)

if(vcanvas.style.display!=='none')

{

let picture =webcam.snap();

document.getElementById('webCam').style.display="none"

document.querySelector("img").src =picture;

document.querySelector("img").height = "400";

document.querySelector("img").width = "400";

document.getElementById("filename").value=picture;

document.getElementById("capture").innerHTML ="Recapture";

document.getElementById("submit").style.display="block"

}

else{

document.getElementById('webCam').style.display="block"

document.querySelector("img").src ="";

document.querySelector("img").height = "0";

document.querySelector("img").width = "0";

document.getElementById("capture").innerHTML ="capture";

document.getElementById("submit").style.display="none"

}

}

</script>

</body>

</html>

FACEVERIFICATION.HTML

<html>

<head>

<meta name="viewport" content="with=device-width, initial-scale=1.0">

<title>Face VERIFICATION FAILED</title>

<link rel="stylesheet" href="1.css">

<link rel="stylesheet" href="{{url\_for('static', filename='1.css')}}">

</head>

<body>

<section>

<div class="uploaded">

<h3> face VERIFICATION FAILED :(</h3>

</div>

<a href="{{url\_for('index')}}">Restart</a>

</section>

</body>

</html>

REGISTRATIOIN SUCCESSFUL.HTML

<html>

<head>

<meta name="viewport" content="with=device-width, initial-scale=1.0">

<title>Face Registration Success</title>

<link rel="stylesheet" href="1.css">

<link rel="stylesheet" href="{{url\_for('static', filename='1.css')}}">

</head>

<body>

<section>

<div class="uploaded">

<h3> Face Registration Success :)</h3>

</div>

<a href="{{url\_for('index')}}">Home</a>

</section>

</body>

</html>

PAYMENT.HTML

<html>

<head>

<meta name="viewport" content="with=device-width, initial-scale=1.0">

<title>Account Details</title>

<link rel="stylesheet" href="1.css">

<link rel="stylesheet" href="{{url\_for('static', filename='1.css')}}">

</head>

<body>

<section>

<form>

<div class="Enter">

<input type="EnterAccountID" placeholder="EnterAccountID" name="EnterAccountID" id="EnterAccountID" required>

<input type="EnterAmount" placeholder="EnterAmount" name="EnterAmount" id="EnterAmount" required>

<input type="EnterCustomerPhoneno" placeholder="EnterCustomerPhoneno" name="EnterCustomerPhoneno" id="EnterCustomerPhoneno" required>

<button type="submit" class="pay">pay</button>

</div>

</form>

</section>

</body>

</html>

VERIFICATIONFAIL.HTML

<html>

<head>

<meta name="viewport" content="with=device-width, initial-scale=1.0">

<title>PASSWORD VERIFICATION FAILED</title>

<link rel="stylesheet" href="1.css">

<link rel="stylesheet" href="{{url\_for('static', filename='1.css')}}">

</head>

<body>

<section>

<div class="uploaded">

<h3> PASSWORD VERIFICATION FAILED :(</h3>

</div>

</section>

</body>

</html>

CUSTOMERDETAILS.HTML

<html>

<head>

<meta name="viewport" content="with=device-width, initial-scale=1.0">

<title>UPLOADED</title>

<link rel="stylesheet" href="1.css">

<link rel="stylesheet" href="{{url\_for('static', filename='1.css')}}">

</head>

<body>

<section>

<div class="uploaded">

<h3> CUSTOMER DETAILS ARE UPLOADED SUCESSFULLY</h3>

</div>

</section>

</body>

</html>

1.CSS

\*{

margin: 0;

padding: 0;

font-family: 'Poppins', sans-serif;

}

body{

min-height: 50vh;

background: #fff;

}

.header{

min-height: 100vh;

width: 100%;

background-position: center;

background-size: cover;

position: relative;

}

h1{

font-family: sans-serif;

text-align: left;

font-size: 30px;

color: black;

background: skyblue;

}

h2{

font-family: sans-serif;

font-size: 40px;

color: black;

font-style: italic;

text-align: center;

padding: 12px 34px;

margin: 8px 0;

}

.new{

text-align: center;

font-size: 20px;

color: bisque;

}

.new-user{

display: inline-block;

text-decoration: none;

color: blue;

padding: 12px 34px;

font-size: 20px;

background: transparent;

position: relative;

cursor: pointer;

text-align: center;

background-color: #04AA6D;

color: white;

padding: 16px 20px;

margin: 8px 0;

}

.new-user:hover{

background: #04AA6D;

transition: 1s;

}

.existing{

text-align: center;

font-size: 20px;

color: #04AA6D;

}

.existing-user{

display: inline-block;

text-decoration: none;

color: blue;

padding: 12px 34px;

font-size: 20px;

background: transparent;

position: relative;

cursor: pointer;

text-align: center;

background-color: #04AA6D;

color: white;

padding: 16px 20px;

margin: 8px 0;

}

.existing-user:hover{

background: #04AA6D;

opacity: 0.8;

transition: 1s;

}

/\*-----capture------\*/

.capture1{

text-align: center;

font-size: 20px;

color: bisque;

}

.capture{

display: inline-block;

text-decoration: none;

color: blue;

padding: 12px 34px;

font-size: 20px;

background: transparent;

position: relative;

cursor: pointer;

text-align: center;

}

.capture:hover{

background: orange;

transition: 1s;

opacity: 0.8;

}

.upload1{

text-align: center;

font-size: 20px;

color: orange;

}

.upload{

display: inline-block;

text-decoration: none;

color: blue;

padding: 12px 34px;

font-size: 20px;

background: transparent;

position: relative;

cursor: pointer;

text-align: center;

}

.upload:hover{

background: orange;

transition: 1s;

}

/\*-------password------\*/

/\*

.password{

text-align: center;

font-size: 25px;

color: bisque;

}

.pass{

display: inline-block;

text-decoration: none;

color: blue;

padding: 12px 34px;

font-size: 13px;

background: transparent;

position: relative;

cursor: pointer;

text-align: center;

font-size: 50px;

}

.pass:hover{

background: pink;

transition: 0s;

}

\*/

.password{

padding: 16px;

}

/\* Full-width input fields \*/

input[type=text], input[type=password] {

width: 100%;

padding: 15px;

margin: 5px 0 22px 0;

display: inline-block;

border: none;

background: #f1f1f1;

}

input[type=text]:focus, input[type=password]:focus {

background-color: #ddd;

outline: none;

}

/\* Overwrite default styles of hr \*/

hr {

border: 1px solid #f1f1f1;

margin-bottom: 25px;

}

/\* Set a style for the submit/register button \*/

.registerbtn {

background-color: orange;

color: white;

padding: 16px 20px;

margin: 8px 0;

border: none;

cursor: pointer;

width: 100%;

opacity: 0.9;

}

.registerbtn:hover {

opacity:1;

}

/\*---------uploaded--------\*/

h3{

text-align: center;

font-family: sans-serif;

color: forestgreen;

font-style: italic;

font-size: 30px;

font-weight: 100;

}

/\*-------CAPTURING--------\*/

.capture12{

text-align: center;

font-size: 20px;

}

.capture11{

display: inline-block;

text-decoration: none;

color: black;

padding: 12px 34px;

font-size: 20px;

background: transparent;

position: relative;

cursor: pointer;

text-align: center;

}

.capture11:hover{

background: orange;

transition: 1s;

opacity: 0.8;

}

.verify1{

text-align: center;

font-size: 20px;

}

.verify{

display: inline-block;

text-decoration: none;

color: black;

padding: 12px 34px;

font-size: 20px;

background: transparent;

position: relative;

cursor: pointer;

text-align: center;

}

.verify:hover{

background: orange;

transition: 1s;

opacity: 0.8;

}

/\*--------payment------\*/

.password{

padding: 16px;

}

/\* Full-width input fields \*/

input[type=text], input[type=EnterAccountID] {

width: 100%;

padding: 15px;

margin: 5px 0 22px 0;

display: inline-block;

border: none;

background: #f1f1f1;

}

input[type=text]:focus, input[type=EnterAccountID]:focus {

background-color: #ddd;

outline: none;

}

input[type=text], input[type=EnterAmount] {

width: 100%;

padding: 15px;

margin: 5px 0 22px 0;

display: inline-block;

border: none;

background: #f1f1f1;

}

input[type=text]:focus, input[type=EnterAmount]:focus {

background-color: #ddd;

outline: none;

}

input[type=text], input[type=EnterCustomerPhoneno] {

width: 100%;

padding: 15px;

margin: 5px 0 22px 0;

display: inline-block;

border: none;

background: #f1f1f1;

}

input[type=text]:focus, input[type=EnterCustomerPhoneno]:focus {

background-color: #ddd;

outline: none;

}

/\* Overwrite default styles of hr \*/

hr {

border: 1px solid #f1f1f1;

margin-bottom: 25px;

}

/\* Set a style for the submit/register button \*/

.pay{

background-color: orange;

color: white;

padding: 16px 20px;

margin: 8px 0;

border: none;

cursor: pointer;

width: 100%;

opacity: 0.9;

}

.pay:hover {

opacity:1;

}