app.py

import os

import numpy as np

from flask import Flask, render\_template, request

from keras.preprocessing.image import load\_img, img\_to\_array

from tensorflow.keras.applications.resnet50 import preprocess\_input

import tensorflow as tf

app = Flask(\_name\_)

# ✅ Load the ResNet50-trained model

model = tf.keras.models.load\_model("best\_model.h5")

# Class labels (in training order)

class\_labels = ['Coccidiosis', 'Healthy', 'New Castle Disease', 'Salmonella']

# Folder for uploaded images

UPLOAD\_FOLDER = os.path.join('static', 'uploads')

os.makedirs(UPLOAD\_FOLDER, exist\_ok=True)

@app.route('/')

def home():

return render\_template('index.html')

@app.route('/about')

def about():

return render\_template('about.html')

@app.route('/contact')

def contact():

return render\_template('contact.html')

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

def predict():

predicted\_class = None

uploaded\_image\_path = None

if request.method == 'POST':

file = request.files['pc\_image']

if file:

filename = file.filename

uploaded\_image\_path = os.path.join(UPLOAD\_FOLDER, filename)

file.save(uploaded\_image\_path)

# ✅ Preprocess image for ResNet50

img = load\_img(uploaded\_image\_path, target\_size=(224, 224))

img\_array = img\_to\_array(img)

img\_array = np.expand\_dims(img\_array, axis=0)

img\_array = preprocess\_input(img\_array)

# ✅ Predict

prediction = model.predict(img\_array)

predicted\_class = class\_labels[np.argmax(prediction)]

return render\_template(

'predict.html',

predict=predicted\_class,

uploaded\_image=uploaded\_image\_path if predicted\_class else None

)

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

app.run(debug=True)