CODE:

app.py

from flask import Flask, render template, request, jsonify from tensorflow.keras.models import load model

from tensorflow.keras.preprocessing.image import load img, img to\_array

import numpy

app Flask( name model load model('poultry model.h5')

labels = ['Coccidiosis', 'Healthy', 'New Castle Disease', 'Salmonella']

@app.route('/")

def home(): return render template('index.html')

@app.route('/predict', methods=["POST']) def predict():

if 'image not in request.files: return jsonify({'error': image uploaded'})

file request.files['image

img = load\_img(file, target size=(224, 224)) img\_array = img\_to\_array(img)

img array = np.expand dims(img array, axis=0)

prediction model.predict(img\_array) predicted\_label = labels[np.argmax(prediction)]

return jsonify(('prediction': predicted\_label})

name main app.run(debug=True)

index.html

<html>

<body>

<script>

async function uploadImage()

const input = document.getElementById('imageInput'); const formData = new FormData(); formData.append('image', input.files [0]);

const res = await fetch('/predict', (

method: 'POST',

body: formData

});

const data await res.json();

document.getElementById('result').innerText "Prediction:data.prediction;

</script> </body>

</html>

PROJECT STRUCTURE:

Poultry\_Disease\_Classification/

dataset/

|healthy/

| coccidiosis/

| newcastle\_disease/

|avian\_influenza/

# Contains poultry images in folders

# Jupyter notebooks for each phase

-notebooks/

| -1\_data\_preprocessing.ipynb

| 2\_model\_building.ipynb

| 3\_model\_training.ipynb

| -4\_model\_evaluation.ipynb

| 5\_inference.ipynb

models/

|best\_model.h5

#Trained model files

flask\_app/

app.py

utils.py

# Web app backend

# HTML files for web UI

-templates/

| index.html

static/

# Images, CSS for web app

|

sample\_images/

docs/

# Reports and presentations

Final\_Report.pdf

| PPT\_Presentation.pptx

Abstract.txt

requirements.txt

README.md

Python dependencies

#

# Project overview

gitignore