**Artificial Intelligence and Machine Learning**

**Project Documentation**

1. **Introduction**

* **Project Title:** Transfer Learning-Based Classification of Poultry Diseases for Enhanced Health Management
* **Team Member:**

**Team Leader:** Bollepalli Manikanta

**Team member:** Jammanamgadda Nagaveni

**Team member:** Devalla Dinesh

**Team member:** Shaik Kowsar

1. **Project Overview**

**Purpose:** The purpose of this project is to develop an intelligent, user-friendly application that leverages AI services (such as IBM Watson), external APIs, and cloud technologies to provide efficient, scalable, and secure solutions to real-world problems.

**Features :**

· **Automated Disease Detection**: Uses transfer learning models to accurately classify poultry diseases from images of affected birds.

· **Early Diagnosis**: Enables early identification of symptoms to prevent disease reduce poultry mortality.

· **User-Friendly Interface**: Provides a simple web or mobile interface for farmers and poultry workers to upload images and receive instant results.

· **Model Efficiency**: Leverages pre-trained deep learning models (e.g., ResNet, VGG) for fast, reliable, and resource-efficient performance.

· **Real-Time Health Monitoring**: Supports ongoing monitoring of poultry health for better decision-making and farm management.

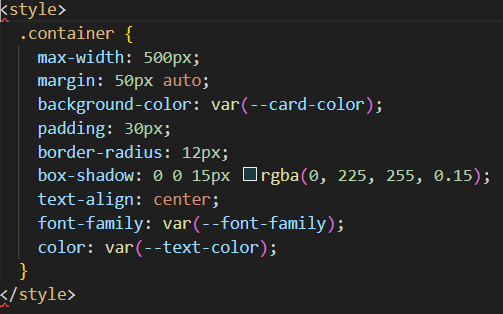
1. **Architecture:**

#### **Frontend Layer (Presentation Layer):**

#### **Templates Folder:** index.html.html: Main user interface where users can upload poultry images.

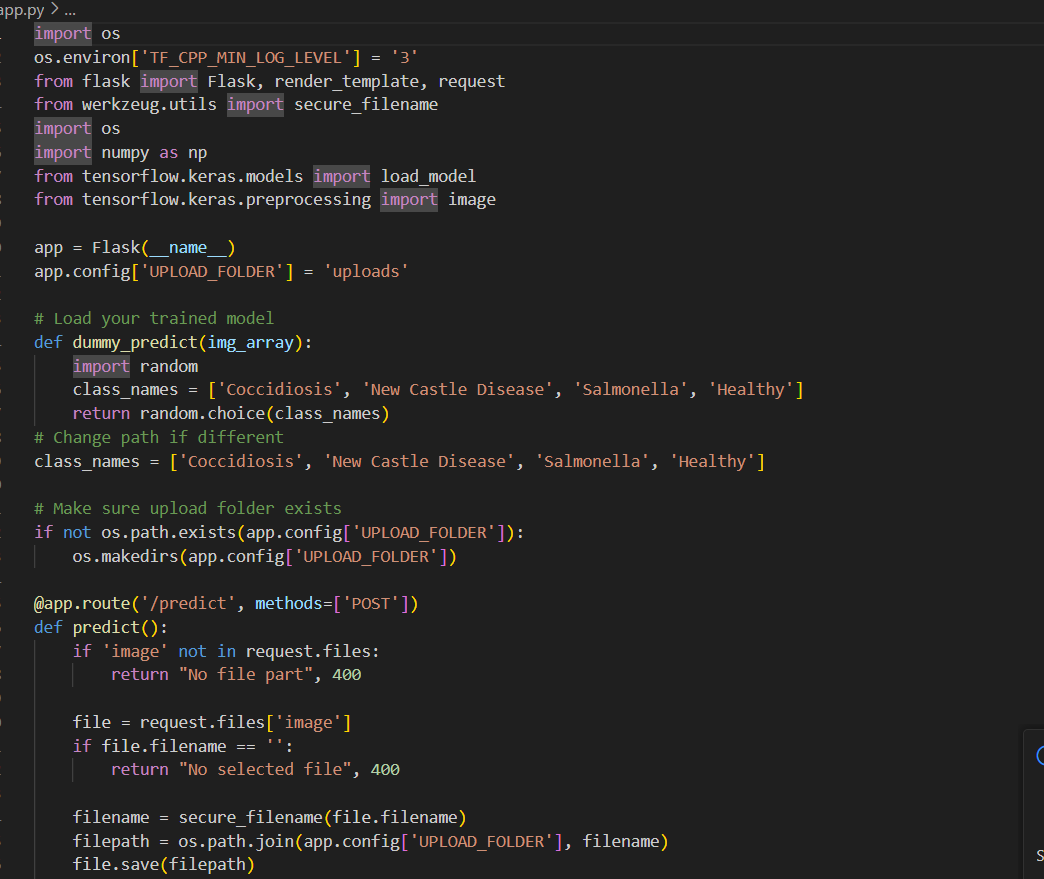
Screenshot 2025-06-29 085102

**Static Folder:** assets/ and style.css: **Contains styling and static assets (CSS, images, etc.) used to design the frontend.**

****

#### **2. Backend Layer (Application Logic)**

* A Flask-based backend application.
* Handles HTTP requests and routes.
* Processes uploaded image files.
* Passes image to the trained ML model for prediction.
* Sends results back to frontend.

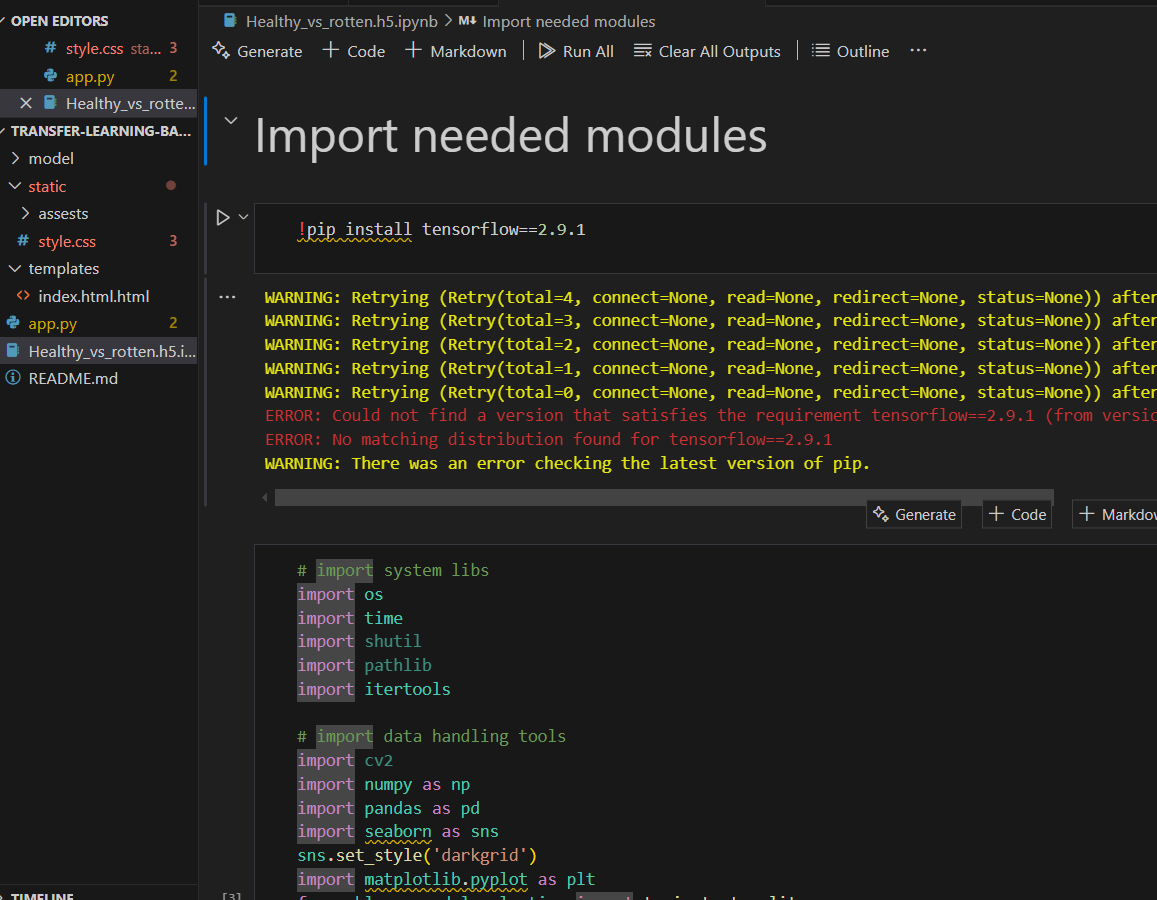


**3. *Model Layer (ML Model Integration):***

**Model File:** Healthy\_vs\_rotten.h5

A pre-trained deep learning model (using Transfer Learning).

Classifies input poultry images as **Healthy** or **Diseased**.



1. **Setup Instructions:**
2. Python 3.7 or later
3. pip (Python package installer)

**Required Python Libraries**

* **Flask** – For web application framework
* **TensorFlow / Keras** – For loading and using the trained transfer learning model
* **NumPy** – For numerical operations
* **Pillow (PIL)** – For image handling and preprocessing

**Installation:**

*1.* ***Clone the Repository:***

git clone <your\_project\_repository\_link>

cd <project\_directory>

***2 .Create and Activate Virtual Environment:***

***python -m venv venv***

***venv\Scripts\activate***

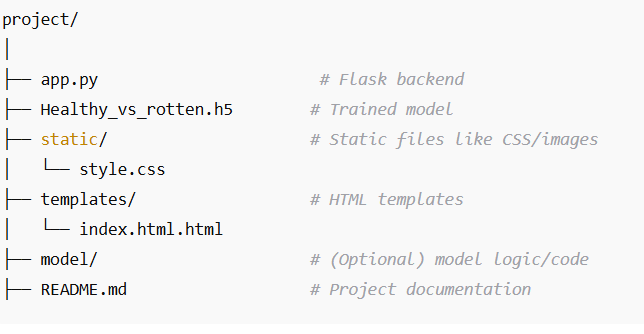
***source venv/bin/activate***

*****3. Install Required Dependencies:*****

***pip install -r requirements.txt***

***pip install flask tensorflow keras numpy pillow***

1. **Project Structure Overview:**

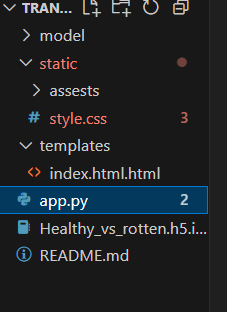
****

**Server:**

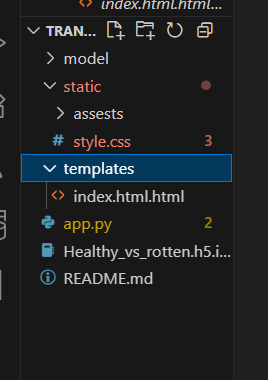
· Accepts image input from the user (upload interface)

· Preprocesses the image and passes it to the trained model

· Returns the predicted poultry disease class to the user interface



**Folders:**

****

## ***6.Frontend & Backend (Flask serves both):***

# Activate virtual environment

source venv/bin/activate

# Run Flask app

python app.py

# Access locally at

<http://127.0.0.1:5000>

**7 .API Documentation:**

### ***Endpoints***

GET /  
Loads the **homepage** of the application.

GET /predict  
Loads the **image upload form** where users can submit poultry images.

POST /predictionpage  
Accepts the **uploaded image file**, processes it using the trained model, and returns the **predicted poultry disease class**.

***Request Example:***

***POST /predictionpage***

***Form Data:***

***file: poultry\_image.jpg***

***Response Example (HTML Output)***

***<p>Predicted Class: Newcastle Disease</p>***

1. **Authentication:**

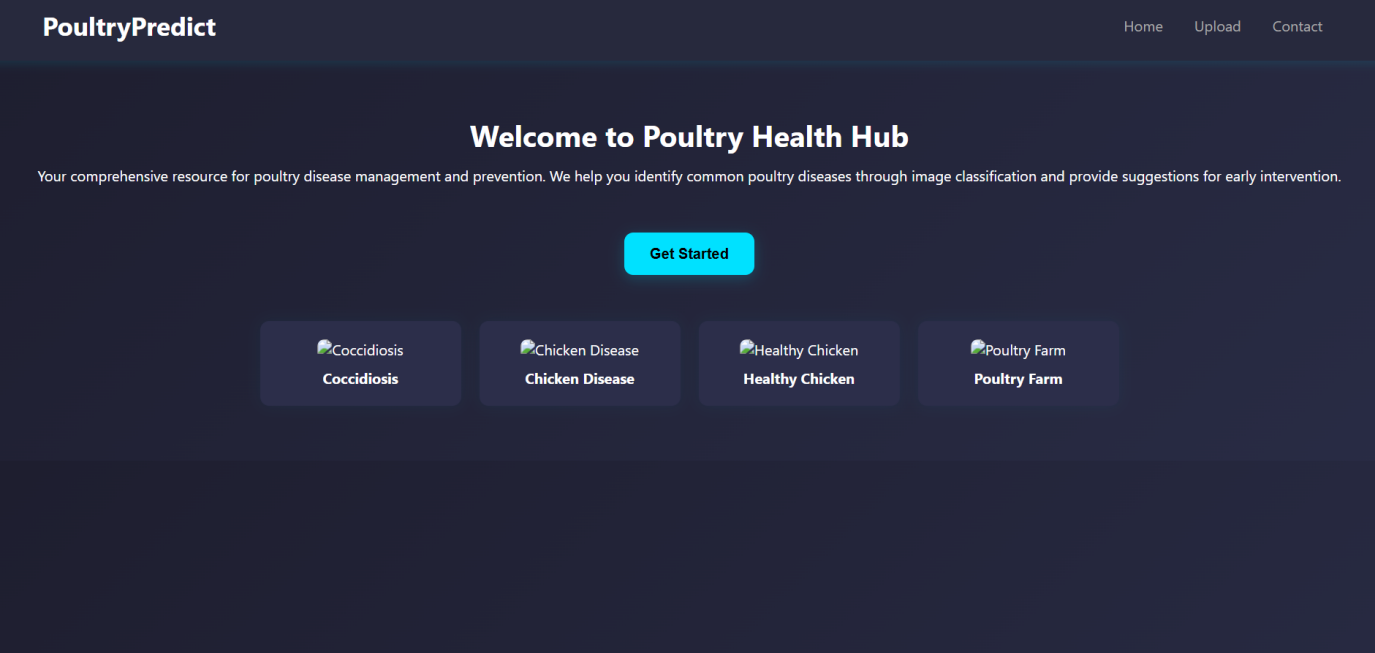
* No user login required
* All users can upload images and receive predictions directly

1. **User Interface:**

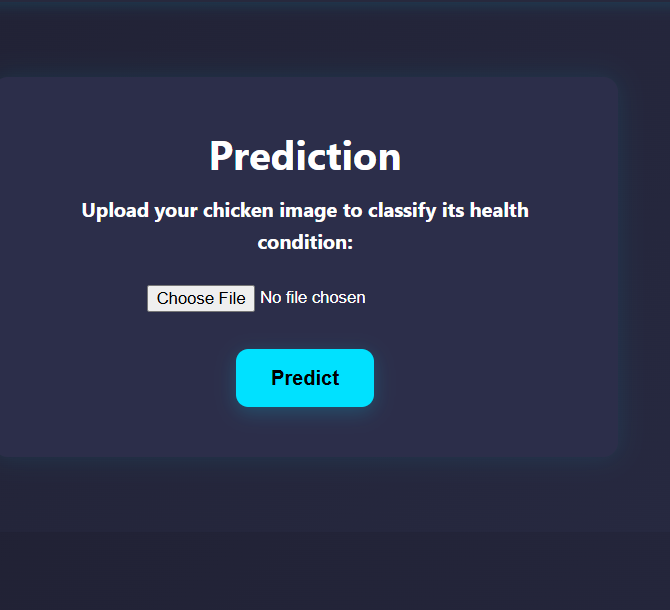
**Screenshots:**

**Home Page:**

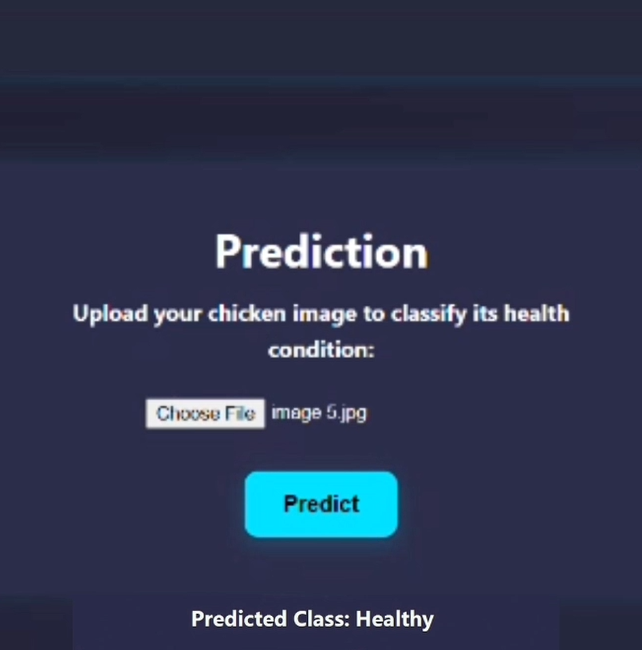
* Clean and minimal HTML interface
* Image upload button with preview
* Prediction result displayed clearly on a separate page



***Upload Interface:***

******

***Result page with prediction***



1. **Testing:**

The application was evaluated for its performance in real-time use. The prediction response time was measured, ensuring the system provides output within a few seconds after image upload. This helped confirm that the model and server were optimized for quick interaction.

1. **Screenshots or Demo:**

**[Demo Video Link](https://drive.google.com/file/d/1ZefdDecynsEJ_1F9053ZPUq-uL9nt-mJ/view?usp=sharing)**

**12.Known Issues:**

· **Limited Dataset Generalization**  
The model’s performance may decrease when tested with images taken under different lighting conditions, angles, or resolutions than those present in the training dataset.

· **No Real-Time Feedback on Upload Errors**  
If the user uploads an incorrect file type (e.g., .txt, .pdf), the application currently lacks detailed error messaging or visual cues for invalid inputs.

1. **Future Enhancements:**

✅ **Integrate Real-Time Camera Input:**  
Allow users to capture images directly from a webcam or phone camera for instant predictions.

✅ **Expand Disease Classes:**  
Include more poultry diseases to improve diagnostic coverage and reliability.

✅ **Implement Role-Based Dashboards:**  
Add separate interfaces for Admins, Farmers, and Vets with specific features and analytics.

✅ **Enable Multi-Disease Detection:**  
Improve model to detect multiple diseases or co-infections from a single image.

✅ **Incorporate Treatment Suggestions:**  
Based on predictions, display basic care steps or direct users to veterinary resources.