

Project Name: BeakyBotX (Framework)

Team Members: Mahalakshmi M, Raja Dharsini J, Deepikaa E

Project Objective

BeakyBotX is a web-based AI bird recognition system that allows users to identify bird species in real-time using a camera-enabled website. The goal is to make bird identification simple and accessible for bird watchers, researchers, and conservationists.

Users can capture a bird image via a mobile or PC camera, and an AI model will analyze the image, identify the bird species, and display the results. This eliminates the need for manual species identification and provides an automated, fast, and accurate solution.

Workflow of BeakyBotX

1 User Opens the Website

The user visits the **BeakyBotX web app** on a **mobile or PC browser**.

2 Website Accesses Camera

- Using **JavaScript (getUserMedia API)**, the website requests access to the **user's camera**.
- The **live camera feed** is displayed on the website.

3 User Takes a Photo of a Bird

- The user **clicks a button** to capture an image from the live feed.
- The captured image is **converted into a format (Base64/Blob)** for processing.

4 Website Sends Image to the AI Model (Backend - Python)

- The image is sent to a **Python Flask API** that runs a **pre-trained deep learning model**.
- The model processes the image using **OpenCV & TensorFlow**.

5 AI Model Detects the Bird Species

- The AI model **analyzes the bird's features** (shape, color, size, etc.).
- It predicts the **species** using a **CNN (e.g., MobileNetV2)**.
- The **predicted bird name** is returned as a **JSON response** to the website.

6 Website Displays the Bird Name & Image

- The website **receives the result** from the AI model.
- The **bird species name and image** are displayed on the screen.

Technologies & Tools Used

1. Frontend (Website) - User Interface

The frontend will be a **simple web page** that captures an image from the **camera** and sends it to the AI model for bird detection.

Technologies Used:

- **HTML** → For structuring the webpage
- **JavaScript** → For camera access & sending the image to the backend
- **getUserMedia API** → To access the camera on mobile & PC
- **Fetch API** → To send the captured image to the backend for processing

2.Backend (AI Model & Image Processing)

The backend is responsible for **handling image processing, running the AI model, and returning the bird species name.**

Technologies Used:

- **Python** → Backend programming language
- **Flask (or FastAPI)** → To create the API that receives images & runs the AI model
- **OpenCV** → For image preprocessing (resizing, color adjustments)
- **NumPy** → For handling image data
- **TensorFlow/Keras** → To train and deploy the deep learning model
- **Pandas** → For handling bird species data

3.Machine Learning & AI Model

We need a deep learning model to **classify bird species** based on the uploaded image.

Technologies Used:

- **CNN (Convolutional Neural Networks)** → Deep learning model for image classification
- **Pre-trained models** MobileNetV2 for transfer learning
- **TensorFlow/Keras** → To train and optimize the model
- **Scikit-learn** → For model evaluation (accuracy, precision, recall)

4.Deployment & Hosting

Local Hosting (For Testing)

- Flask development server (<http://127.0.0.1:5000>)