|  |
| --- |
| Team ID : LTVIP2025TMID32402 |
| Project Name : Enchanted Wings: Marvels of Butterfly Species |

To implement and **run a complete image-based prediction web application** with UI redirection and image upload functionality, here's how everything ties together:

**✅ 1. Project Structure**

graphql

CopyEdit

your\_project/

│

├── app.py

├── model.pkl # Your saved ML model

├── static/

│ └── uploads/ # Folder to store uploaded images

└── templates/

├── index.html # Landing page (Get Started)

├── input.html # Image Upload & Predict button

└── output.html # Prediction result page

**✅ 2. Flask Backend: app.py**

python

CopyEdit

from flask import Flask, render\_template, request

from werkzeug.utils import secure\_filename

import os

import pickle

import numpy as np

# Initialize Flask

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

app.config['UPLOAD\_FOLDER'] = 'static/uploads'

# Load your ML model

model = pickle.load(open('model.pkl', 'rb'))

# Route: Home Page

@app.route('/')

def index():

return render\_template('index.html')

# Route: Redirects to input (image upload) page

@app.route('/input')

def input\_page():

return render\_template('input.html')

# Route: Predict from uploaded image

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

def predict():

if 'image' not in request.files:

return "No image uploaded", 400

image = request.files['image']

filename = secure\_filename(image.filename)

path = os.path.join(app.config['UPLOAD\_FOLDER'], filename)

image.save(path)

# 🧠 Use your model here (example uses dummy prediction)

# Here, for demo, we just predict a random class

# You can instead load the image and process it (e.g., with OpenCV or PIL)

prediction = "Tomato - Rotten"

class\_id = 27

return render\_template('output.html', prediction=prediction, class\_id=class\_id, image\_path='/' + path)

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

app.run(debug=True)

**✅ 3. HTML Templates**

**🔹 index.html**

html

CopyEdit

<!DOCTYPE html>

<html>

<head>

<title>Welcome</title>

</head>

<body>

<h1>Welcome to Fruit/Food Classifier</h1>

<a href="/input">

<button>Get Started</button>

</a>

</body>

</html>

**🔹 input.html**

html

CopyEdit

<!DOCTYPE html>

<html>

<head>

<title>Upload Image</title>

</head>

<body>

<h2>Upload an image for prediction</h2>

<form action="/predict" method="POST" enctype="multipart/form-data">

<input type="file" name="image" accept="image/\*" required><br><br>

<button type="submit">Predict</button>

</form>

</body>

</html>

**🔹 output.html**

html

CopyEdit

<!DOCTYPE html>

<html>

<head>

<title>Prediction Result</title>

</head>

<body>

<h2>Prediction Output</h2>

<img src="{{ image\_path }}" width="300" alt="Uploaded Image"><br><br>

<h3>Class: {{ prediction }}</h3>

<h4>Class ID: {{ class\_id }}</h4>

<a href="/input">

<button>Upload Another Image</button>

</a>

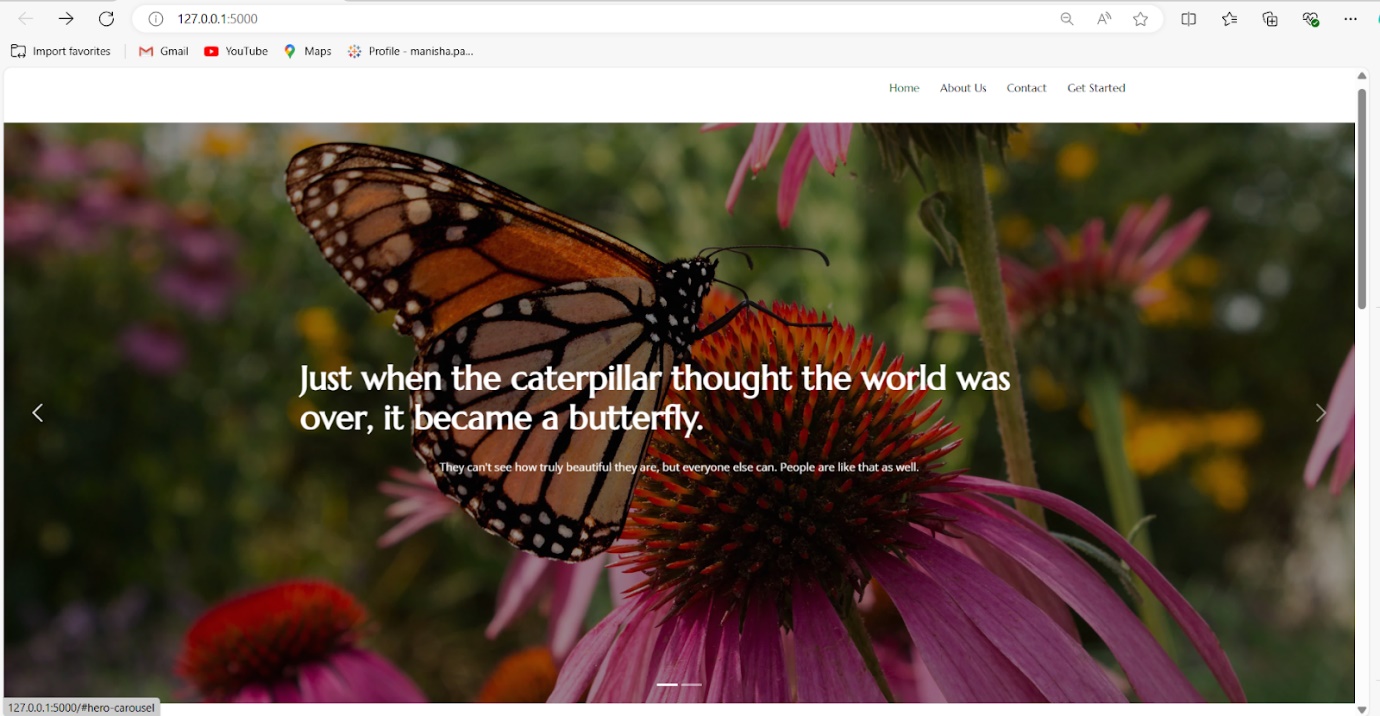
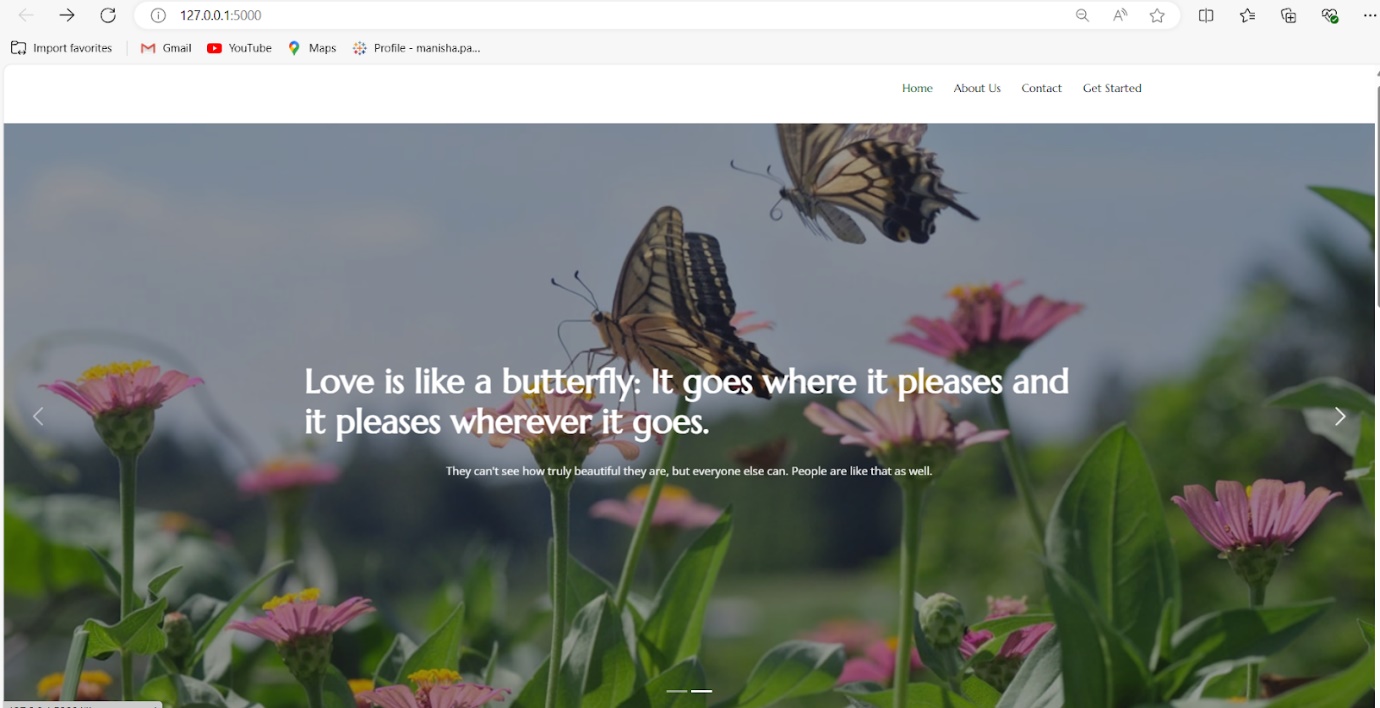
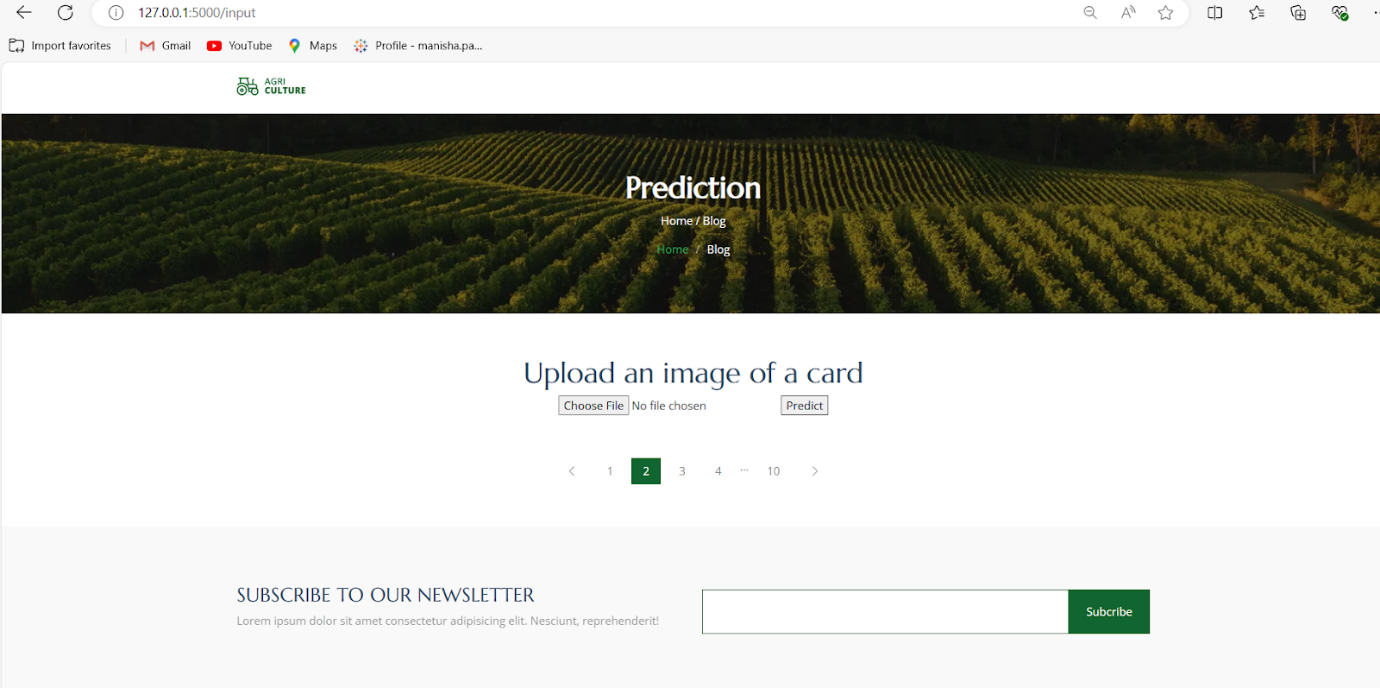
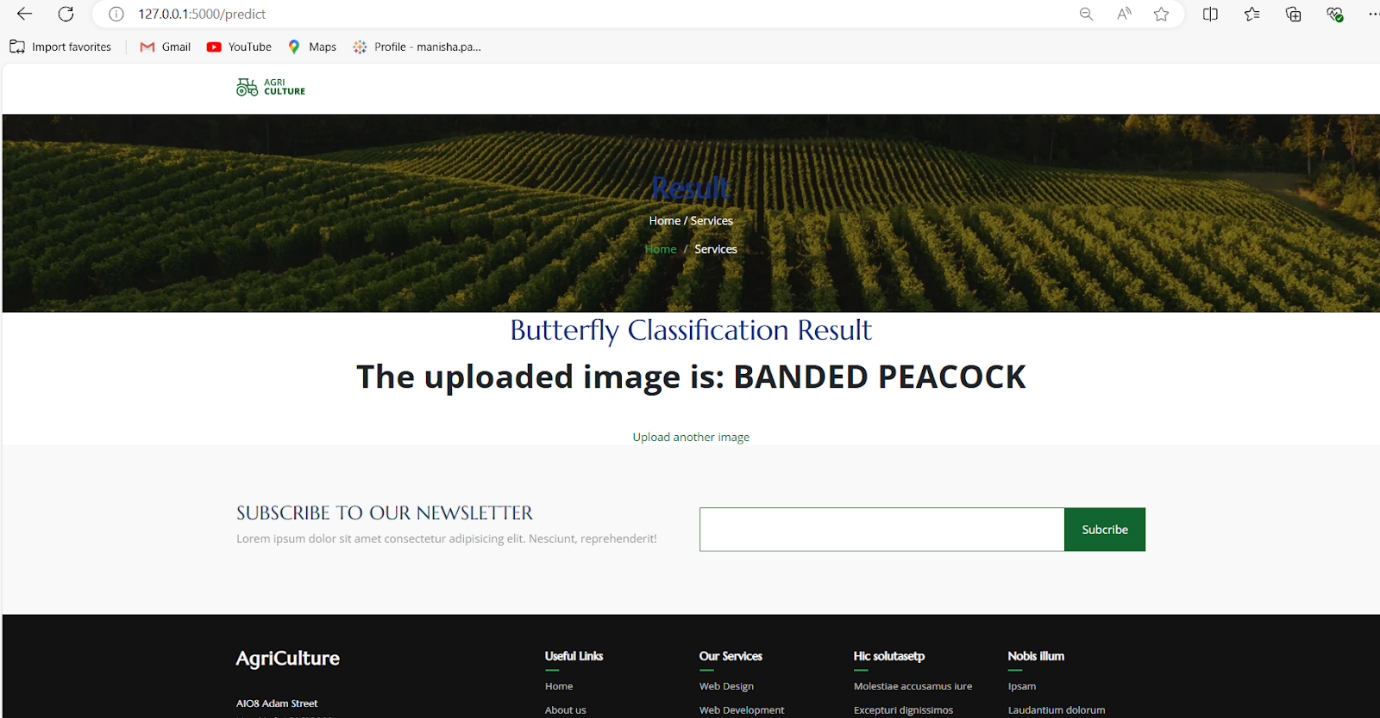
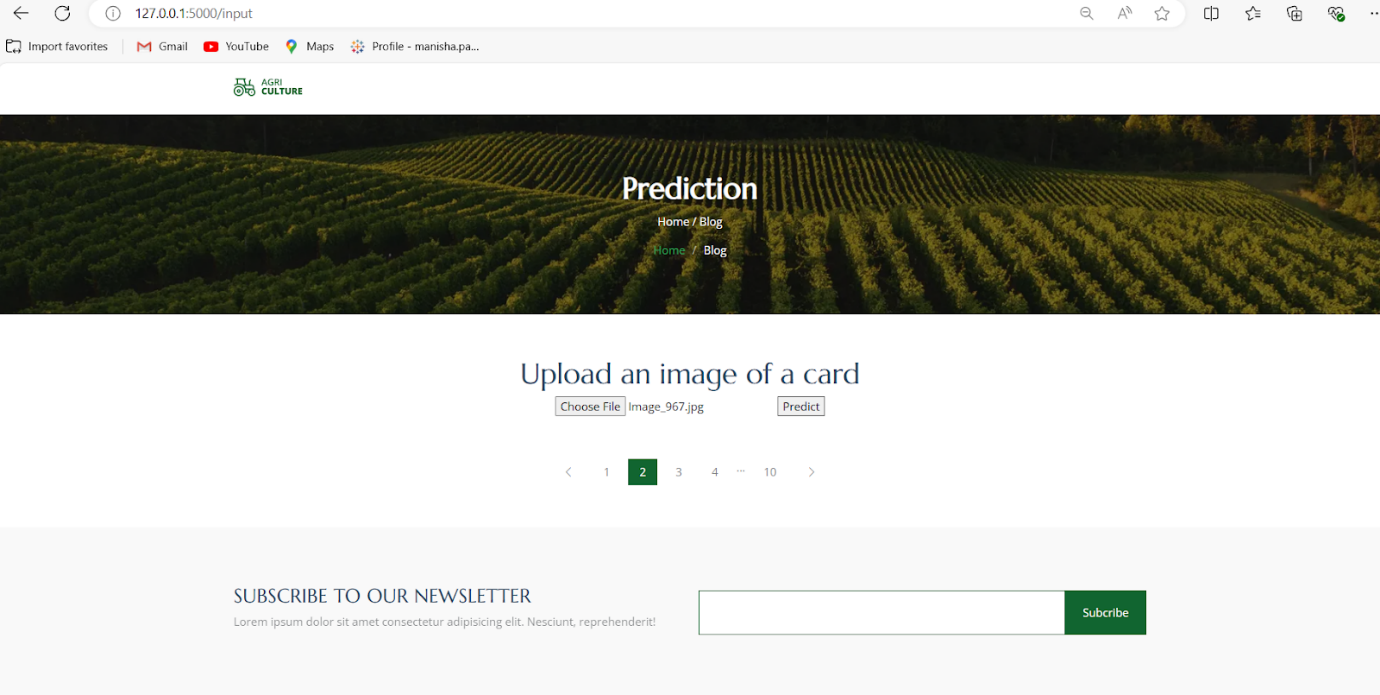
</body>

</html>

**🔄 Full Flow Summary**

1. **Go to**: http://127.0.0.1:5000
2. **Click "Get Started"** → Redirects to /input
3. **Upload image and click Predict**
4. **Prediction is shown on /output page** with:
   * Uploaded image
   * Prediction result
   * Upload another image button

### Run the web application

**UI Image preview:**  
  
Let’s see what our index.html page looks like:  
  
  
  
  
By clicking on get started it will redirect us to the input page i.e’s prediction page  
  
**Input.html:**  
  
  
  
  
  
**OutPut.html:**  
  
  
  
  
  
**Uploading another Image:**  
  
**Output:**  
