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

Below is the complete **Python Flask application code** that:

* Loads a saved machine learning model
* Uses Flask to serve the frontend
* Retrieves input values from the HTML form using POST method
* Passes them to the model
* Displays prediction on the output page

**✅ app.py — Full Backend Code**

python

CopyEdit

# Import the libraries

from flask import Flask, render\_template, request

import numpy as np

import pickle

# Load the saved model

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

# Initialize the Flask app

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

# Route to index page

@app.route('/')

def index():

return render\_template('index.html')

# Route to predict function

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

def output():

# Retrieve values from form using POST

feature1 = float(request.form['feature1'])

feature2 = float(request.form['feature2'])

feature3 = float(request.form['feature3'])

feature4 = float(request.form['feature4'])

# Store the values in an array (modify depending on your model's input)

features = np.array([[feature1, feature2, feature3, feature4]])

# Pass input to the model and get prediction

prediction = model.predict(features)

# Render result in output.html

return render\_template('output.html', prediction\_text=f"Prediction Result: {prediction[0]}")

# Main function

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)

**🧠 Assumptions**

* The model expects 4 float inputs.
* You have model.pkl saved in the same directory.
* HTML files (index.html, output.html) are in the templates/ folder.

**✅ index.html (inside templates/)**

html

CopyEdit

<!DOCTYPE html>

<html>

<head>

<title>Model Input Page</title>

</head>

<body>

<h1>Enter Features for Prediction</h1>

<form action="/predict" method="post">

<label>Feature 1:</label>

<input type="text" name="feature1" required><br><br>

<label>Feature 2:</label>

<input type="text" name="feature2" required><br><br>

<label>Feature 3:</label>

<input type="text" name="feature3" required><br><br>

<label>Feature 4:</label>

<input type="text" name="feature4" required><br><br>

<input type="submit" value="Predict">

</form>

</body>

</html>

**✅ output.html (inside templates/)**

html

CopyEdit

<!DOCTYPE html>

<html>

<head>

<title>Prediction Result</title>

</head>

<body>

<h1>{{ prediction\_text }}</h1>

<a href="/">Back to Home</a>

</body>

</html>

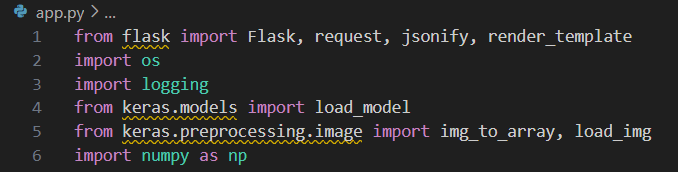
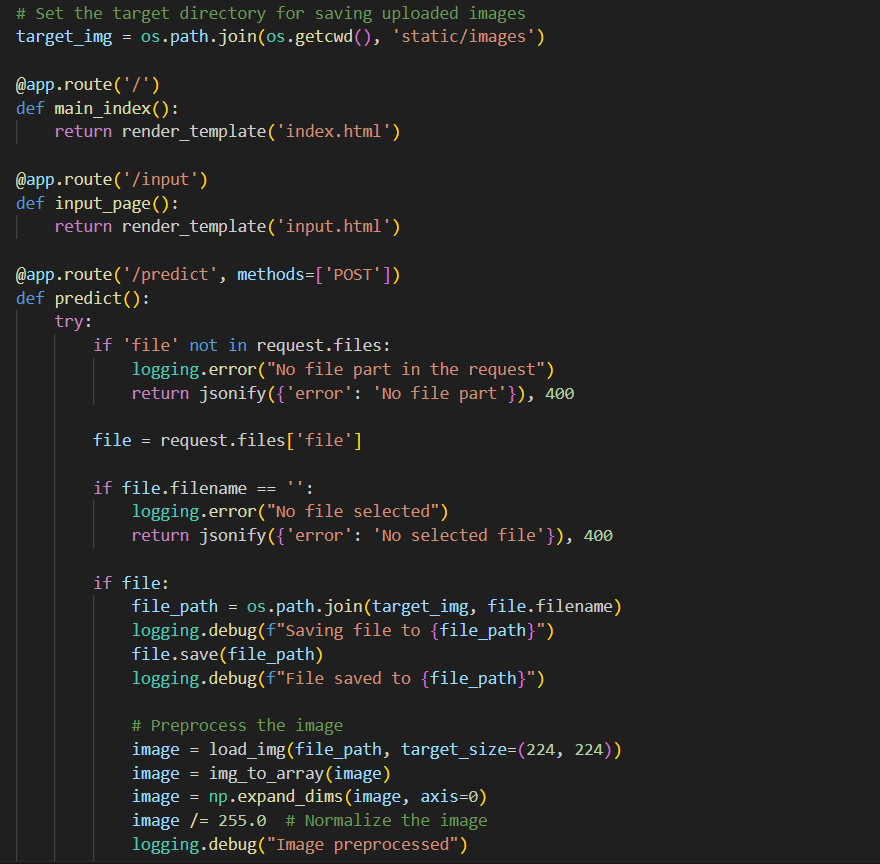
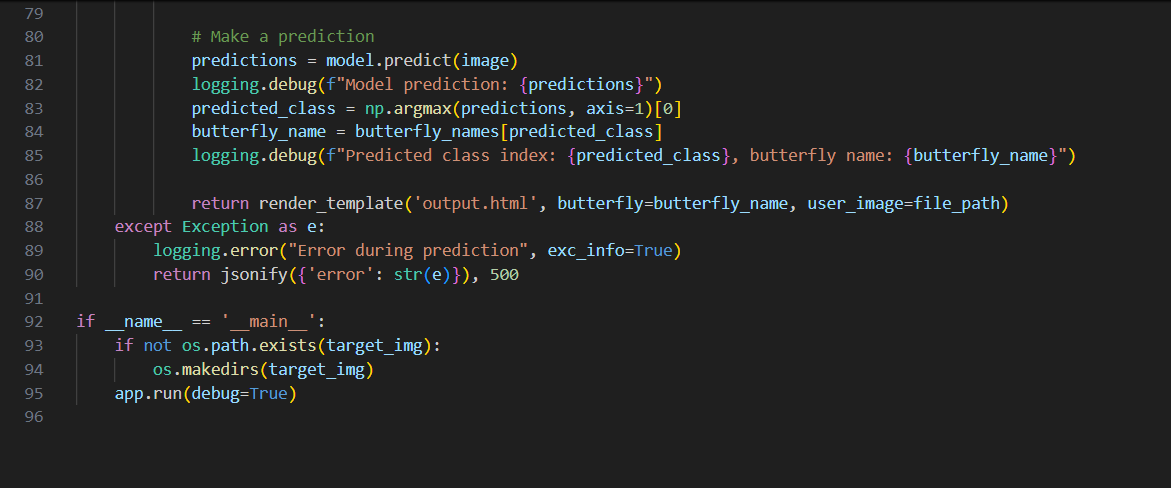
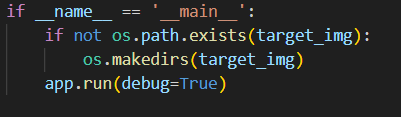
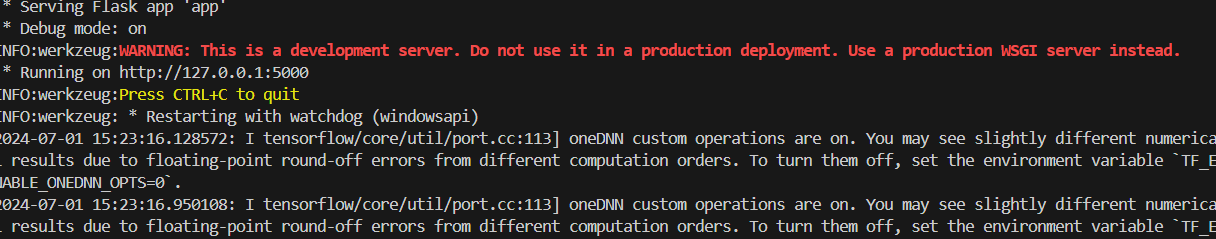
**🚀 To Run the App**

1. **Open Anaconda Prompt**
2. Navigate to the folder containing app.py
3. Run the script:

nginx

CopyEdit

python app.py

1. Open your browser and go to:  
   👉 [**http://127.0.0.1:5000**](http://127.0.0.1:5000)
2. **Build Python code:**
3. Import the libraries
4. 
5. Load the saved model. Importing the Flask module in the project is mandatory. An object of the Flask class is our WSGI application. The Flask constructor takes the name of the current module (\_\_name\_\_) as argument.
6. Here we will be using the declared constructor to route to the HTML page which we have created earlier.
7. In the above example, the ‘/’ URL is bound with the index.html function. Hence, when the index page of the web server is opened in the browser, the html page will be rendered. Whenever you enter the values from the html page the values can be retrieved using POST Method.
8. Retrieves the value from UI:
9. 
10. 
11. 
12. Here we are routing our app to the output() function. This function retrieves all the values from the HTML page using a Post request. That is stored in an array. This array is passed to the model. Predict () function. This function returns the prediction. This prediction value will rendered to the text that we have mentioned in the output.html page earlier.
13. Main Function:
14. 
15. Open Anaconda prompt from the start menu
16. Navigate to the folder where your Python script is.
17. Now type the “app.py” command
18. Navigate to the local host where you can view your web page.
19. Click on the inspect button from the top right corner, enter the inputs, click on the predict button, and see the result/prediction on the web.
20. 
21. Now, Go the web browser and write the localhost url (http://127.0.0.1:5000) to get the below results