Pollen Grain Classification — Flask Web App

A deep learning-powered web application that classifies pollen grain images using a trained CNN model, built with Flask, Keras, and Bootstrap UI.

# Project Structure

POLLEN\_GRAIN/  
├── app.py # Main Flask application  
├── cnn.keras # Trained model file  
├── flask/  
│ ├── static/  
│ │ └── images/ # Background or style images  
│ ├── uploads/ # User-uploaded images  
│ └── templates/  
│ ├── index.html # Home page  
│ ├── prediction.html # Result page  
│ └── logout.html # Logout/dummy exit  
├── .venv/ # Virtual environment (optional)  
└── README.md # Project documentation

# Requirements

Make sure you’re using Python 3.9 or compatible. Install dependencies:

pip install flask tensorflow keras pillow numpy

# How to Run

1. Activate your virtual environment (if you use one):

.venv\Scripts\activate

2. Start Flask:

flask --app app run

3. Open in browser: http://127.0.0.1:5000/

# How It Works

1. User uploads a pollen image via the web UI.  
2. Image is resized and preprocessed.  
3. The CNN model (cnn.keras) predicts the class.  
4. The result is rendered back to the user.

# Model Info

- Trained using Keras Sequential CNN.  
- Input shape: (128, 128, 3)  
- Output: Softmax over pollen grain classes  
- Example class labels: ['Dandelion', 'Rose', 'Lily', 'Orchid']

# Flask Prediction Code Snippet

img = Image.open(filepath).convert('RGB')  
img = img.resize((128, 128))  
img\_array = np.expand\_dims(np.array(img) / 255.0, axis=0)  
prediction = model.predict(img\_array)  
result = np.argmax(prediction)

# UI Features

- Clean Bootstrap-based layout  
- Upload form  
- Background image support  
- Prediction result display

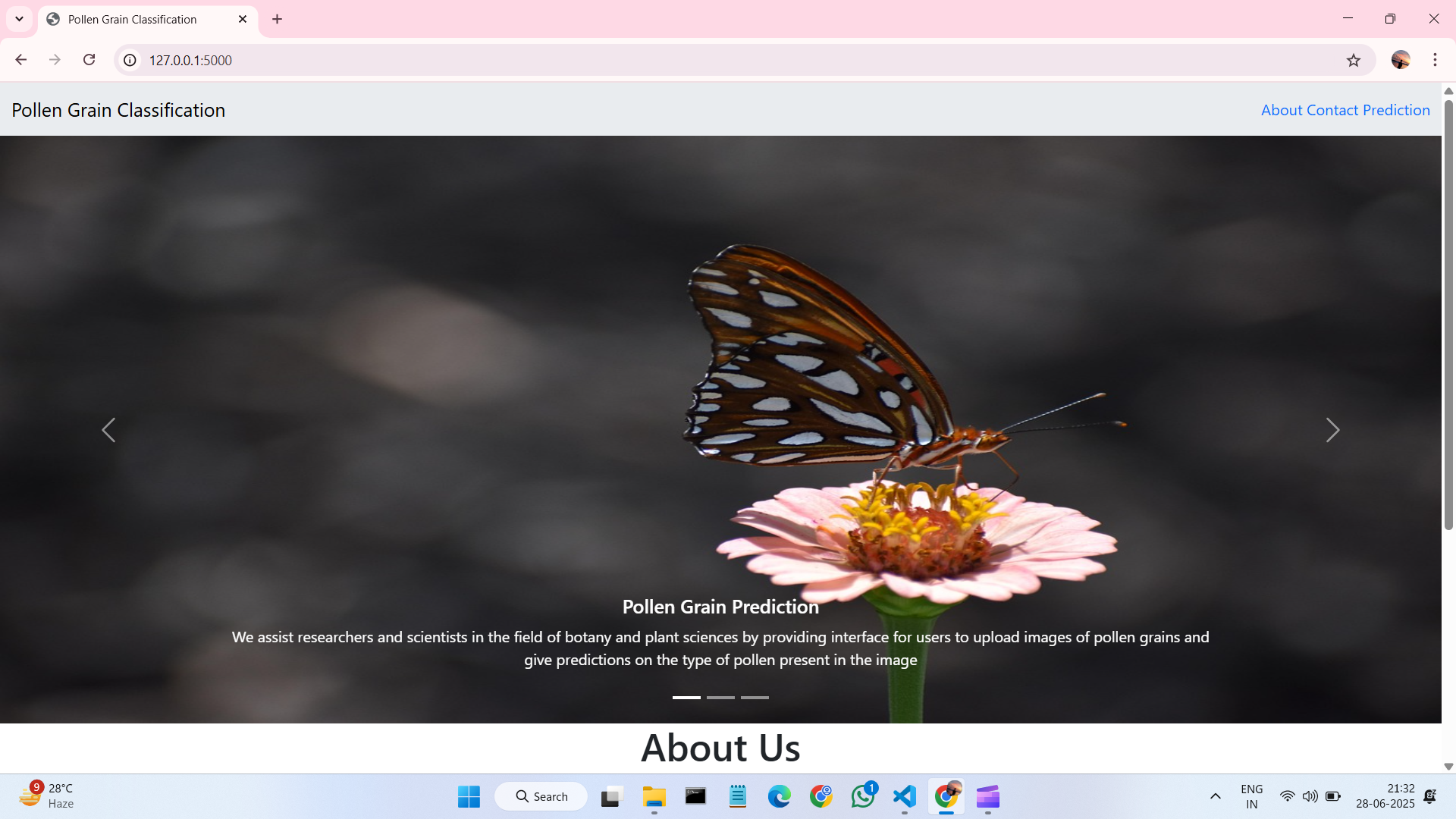
# Notes

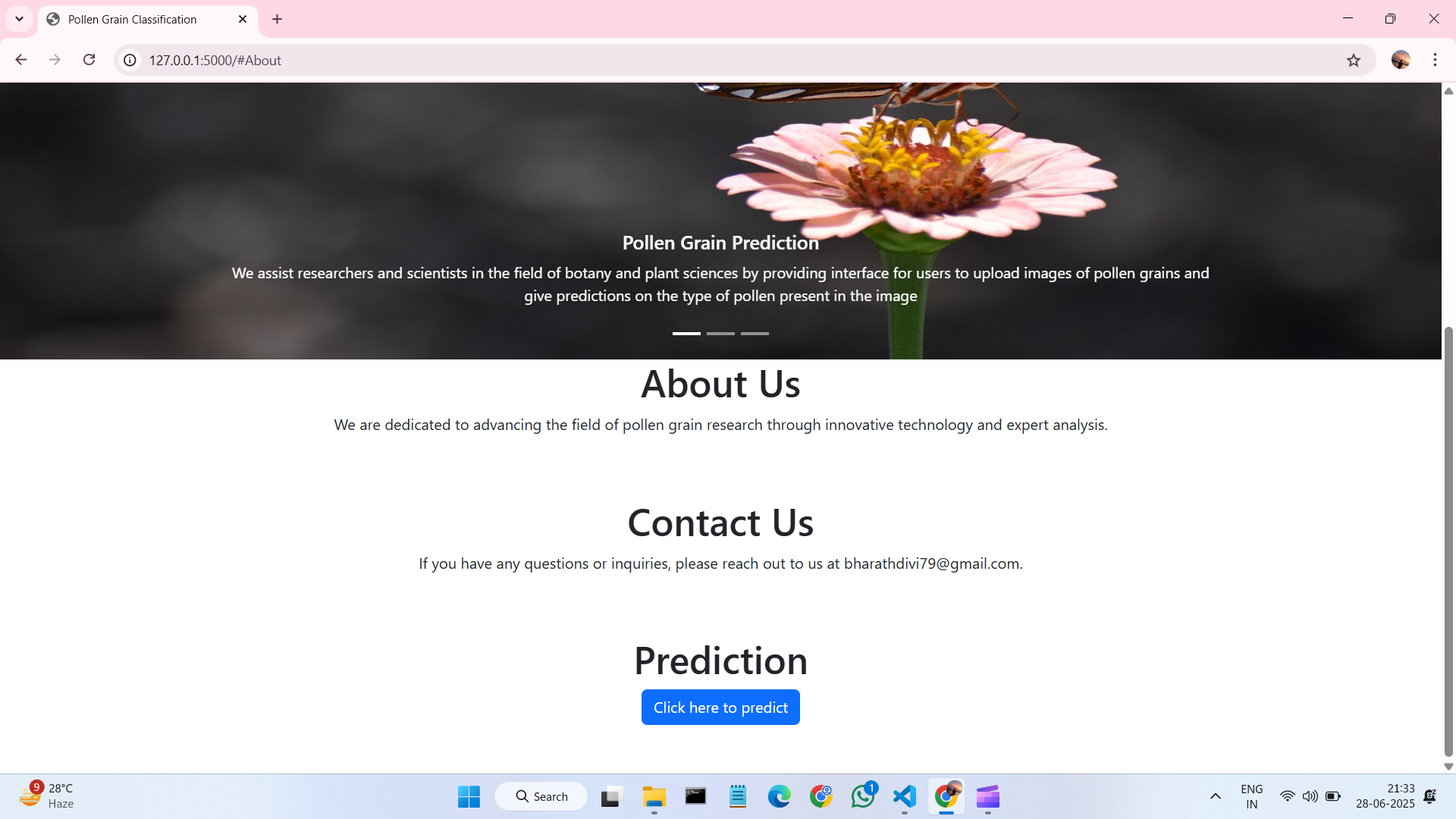
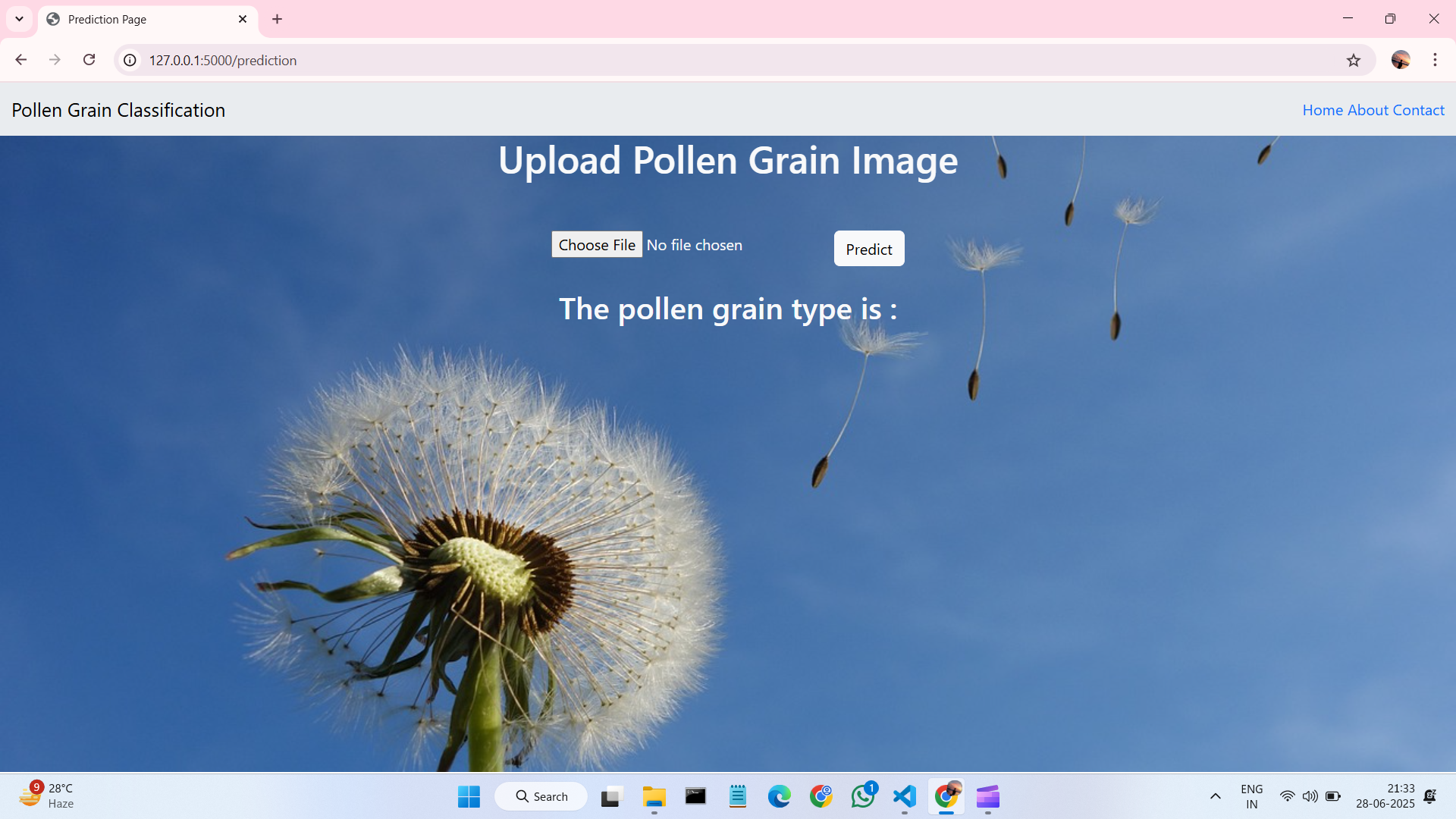
- Make sure uploads/ folder exists or auto-create it in code.  
- Ensure cnn.keras matches input size (128x128 or trained size).  
- Convert grayscale images to RGB before predicting.

# To Do / Improvements

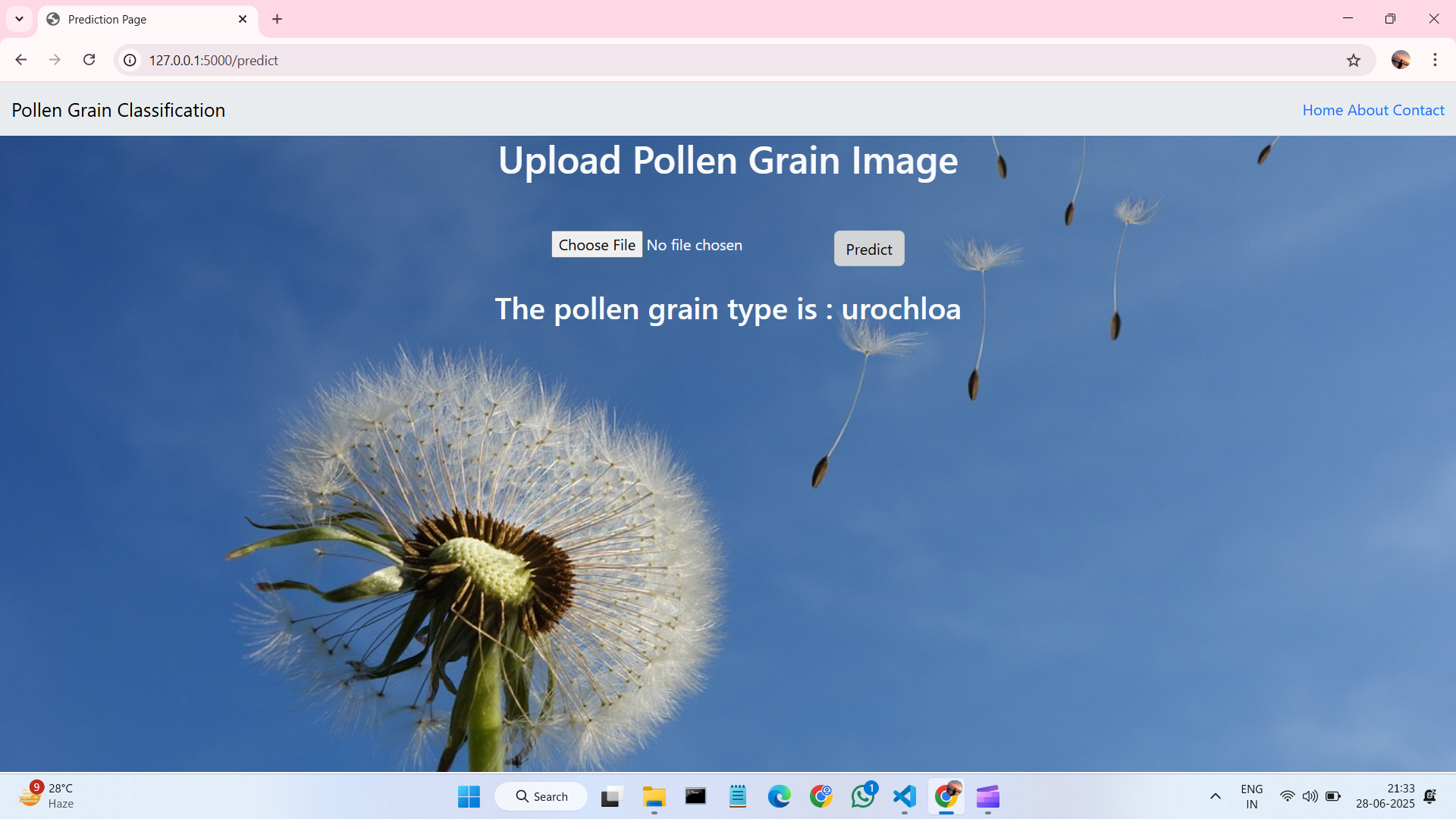
- Add multiple class label support (with label mapping).  
- Display uploaded image on prediction page.  
- Add history of predictions or logs.

# Output:

**Home Page:**

**About Page:**

**Prediction Page:**

**Final Result:**

**For More Details (video) –**

**https://drive.google.com/drive/folders/1hHfkDFlwJLPdZzx1Tjl3V27fdfEIQA\_A?usp=sharing**