Facial Recognition

Oodles AI Proprietary and Confidential
https://www.oodles.com

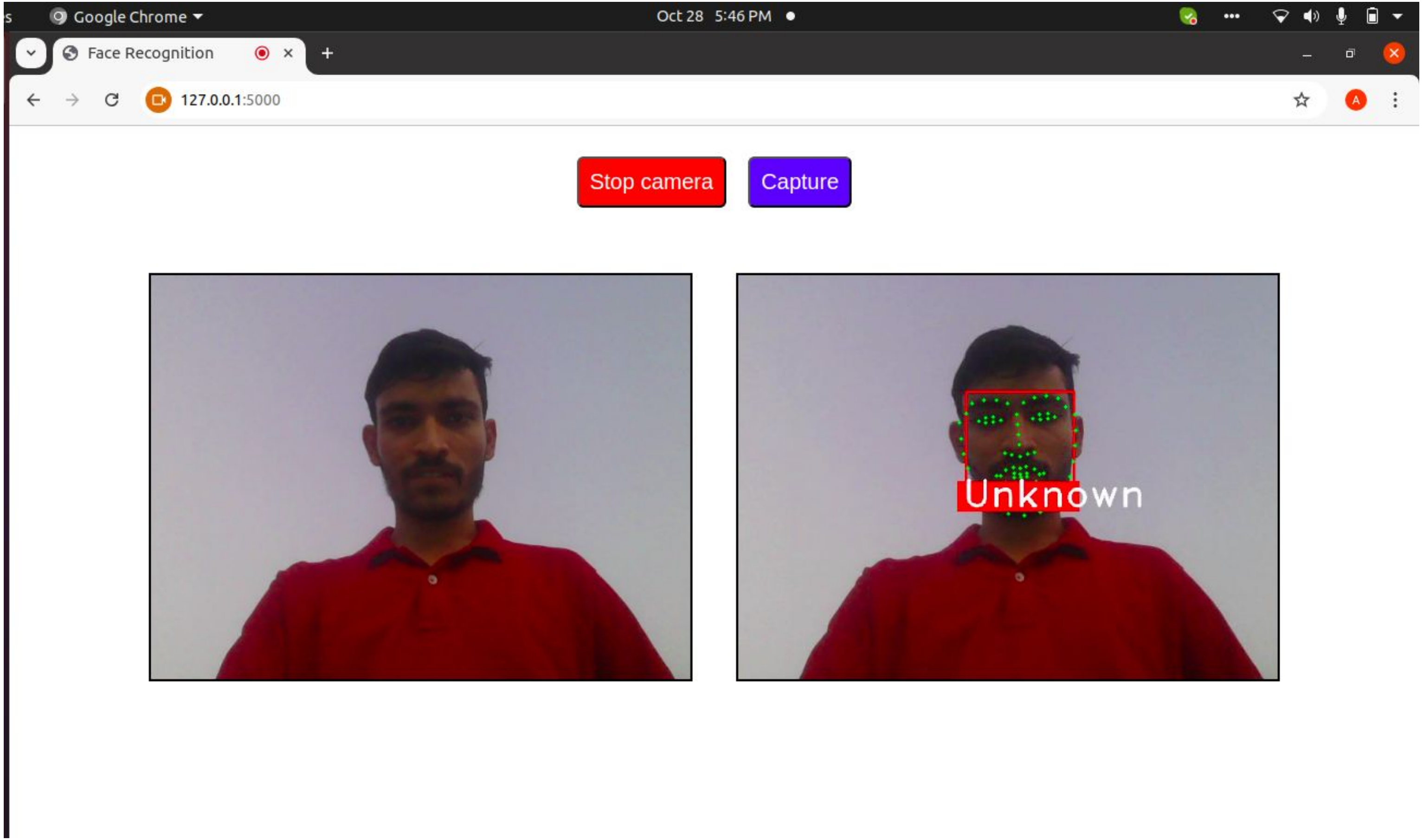
Introduction

- It is a web-based facial recognition application that identifies and verifies faces in real-time.
- The app offers a secure and efficient solution for various use cases, including access control, identity verification, and personalized user experiences.
- The application is built using **HTML, CSS, and JavaScript** for the front end, providing a responsive and interactive user interface.
- The back end is powered by **Flask**, a lightweight and flexible Python web framework, ensuring seamless communication between the client and the server.
- For facial recognition, the app leverages **OpenCV** and **DLIB**, two powerful computer vision libraries. The **dlib_face_recognition_resnet_model_v1.dat** model is utilized to identify and compare facial features with high accuracy.
- The application also employs the **shape_predictor_68_face_landmarks.dat** model for precise facial landmark detection, ensuring robust alignment and recognition even in various lighting conditions and angles.
- The app’s architecture integrates these models and libraries efficiently to process and match faces in real-time, providing quick and reliable results.

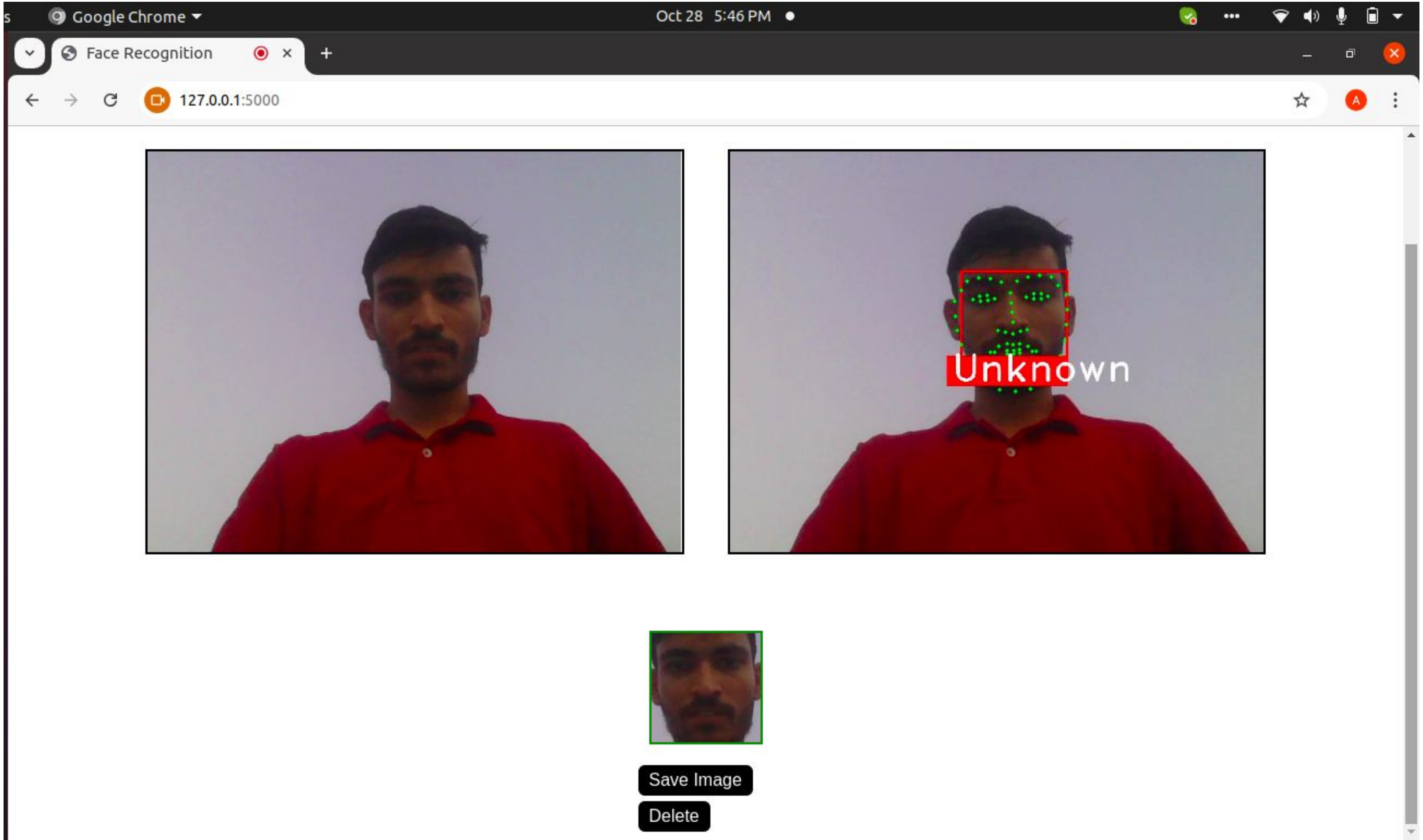
Application Flow

1. Launch the application https://face-reco.oodleslab.com/
2. Click on Start Camera button
3. It will start scanning your face on right screen
4. Click on the capture button to capture the face and save with your name.
5. Once you save, it will display the success popup message and picture will get saved in database.
6. If any user’s face is already saved in DB and upon future scanning it will detect the face and won’t allow to capture again. Only unknown faces can be captured.

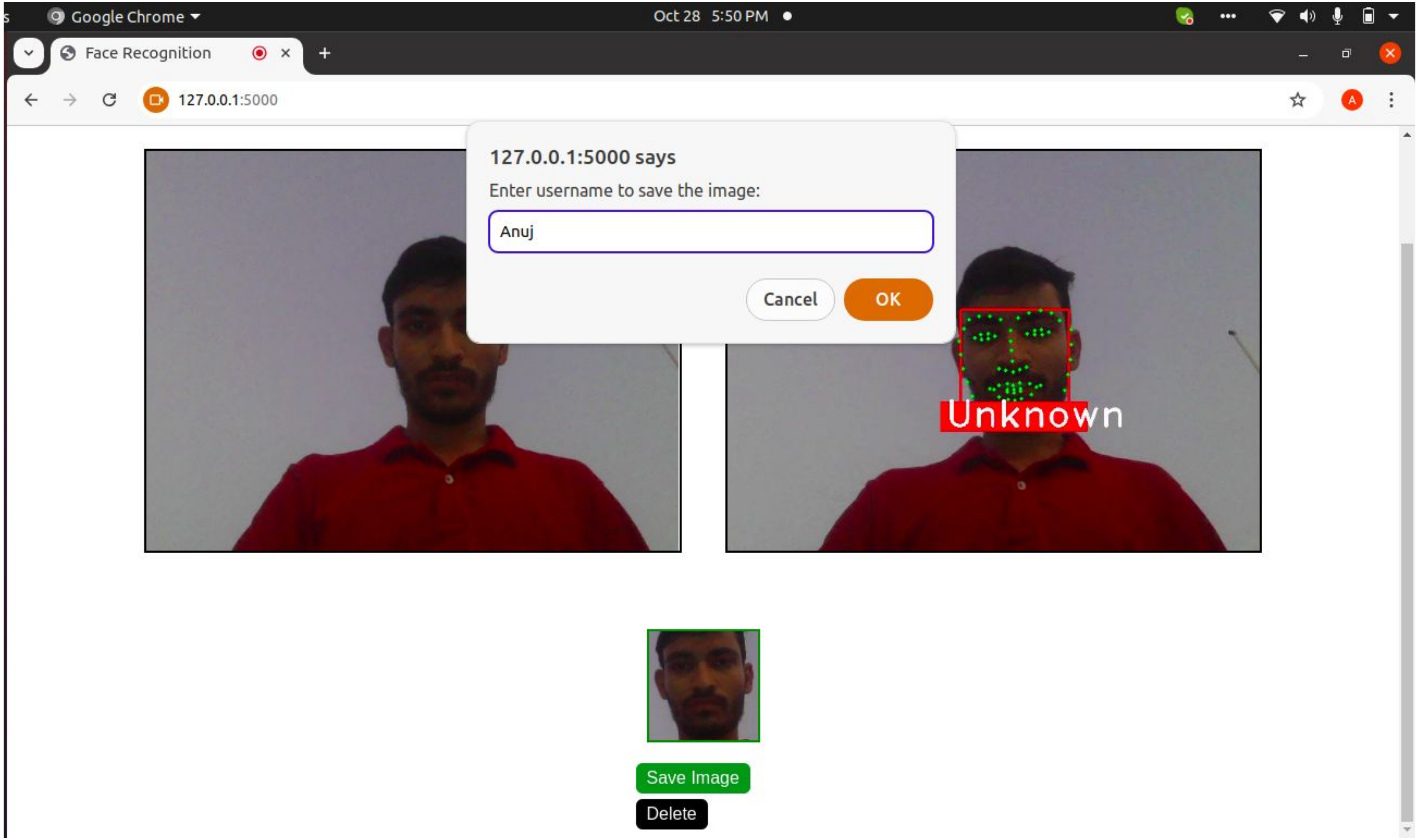
Analysing and Scanning Face



Face Captured



Save Captured Face

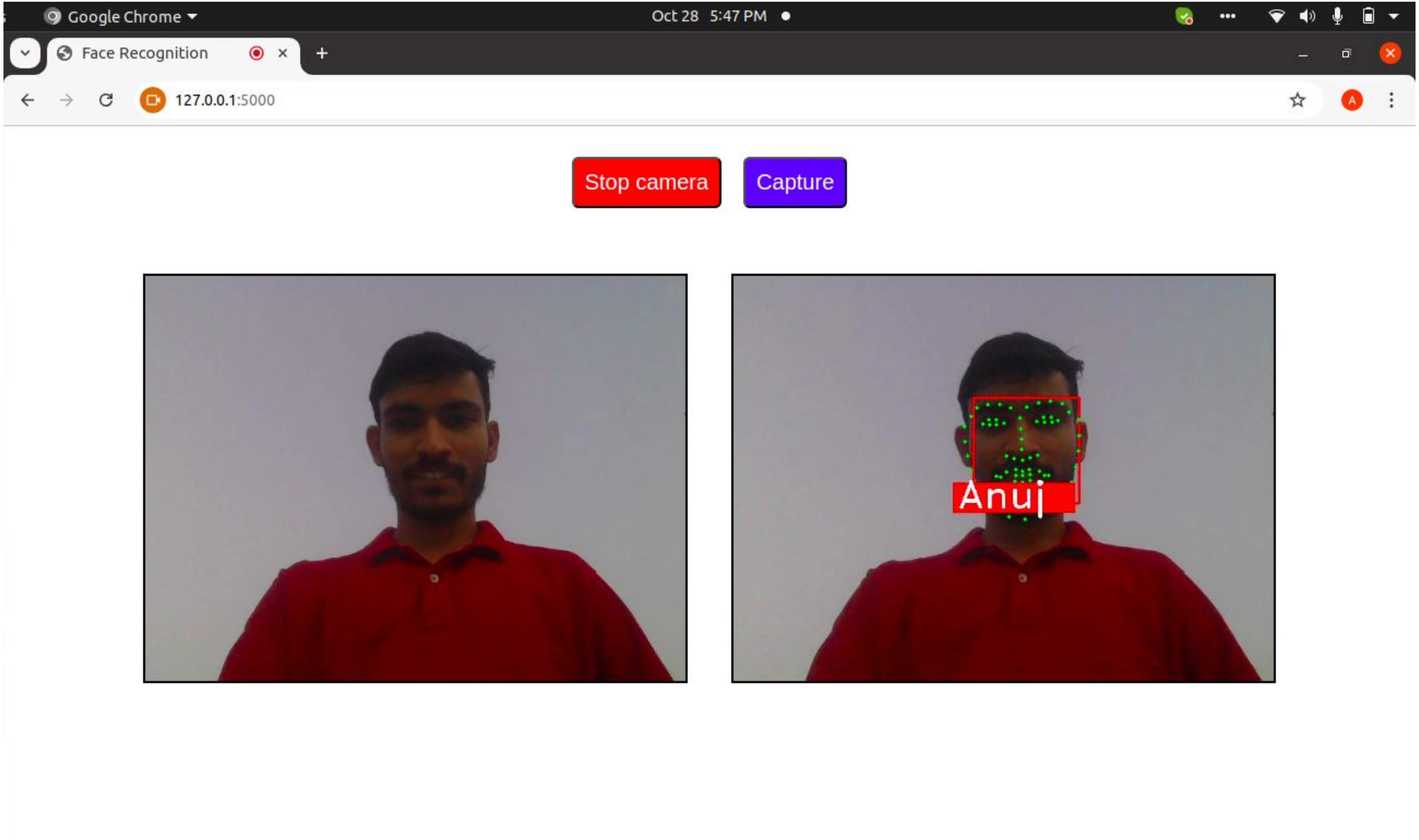


127.0.0.1:5000 says

Enter username to save the image:

Anuj

Face Detected Upon Future Scanning



Tech Stack

Technology	Version	Description
HTML, CSS, JavaScript	-	Used for all UI elements and front end related tasks.
Flask	3.0	Used to handle server-side logic, manages API endpoints, and communicates with the front end.
Python	3.8.10	Powers the back-end development with Flask and integrates libraries like OpenCV and Dlib for efficient image processing and facial recognition functionalities.
OpenCV	4.10	Used for real-time image and video processing, crucial for detecting and analyzing facial features.
Dlib	-	library that supports facial recognition and landmark detection, enabling precise face alignment and recognition.
dlib_face_recognition_resnet_model_v1.dat	-	A pre-trained deep learning model used to identify and verify faces with high accuracy
shape_predictor_68_face_landmarks.dat	-	A facial landmark detection model that identifies key points on the face, ensuri accurate alignment and analysis.