AGE AND IDENTITY VERIFICATION SYSTEM

Zynga hackathon

Team name - CODE RESPAWN

Team members - Ishu(01110162022)
Parul Prathana(01710162022)

We are aspiring Machine learning engineers, who have successfully earned certifications in Python-Machine Learning and Computer Vision-Deep Learning each from Centre of Excellence-Anveshan foundation IGDTUW. We are passionate to hone our machine learning skills by engaging in solving real world challenges and builfing meaningful ML applcations.

Solution Summary

- **Purpose:** Web-based system for secure age and identity verification using ID card and live selfie.
- **Workflow:** User uploads ID card, captures selfie via webcam, frontend validates inputs, backend processes images for DOB extraction, face matching, and age calculation, then displays result.
- **Tech Architecture:** Frontend (HTML/CSS/JavaScript) for UI and camera access; Backend (Flask, Python) with OpenCV (face detection), DeepFace (face verification), Tesseract (OCR), and PIL (image processing).
- **Features:** ID card upload, live selfie capture, real-time video preview, form validation, and verification result display with face match and age check.
- **Outcome:** Secure, user-friendly platform ensuring accurate identity and age validation for restricted access systems.
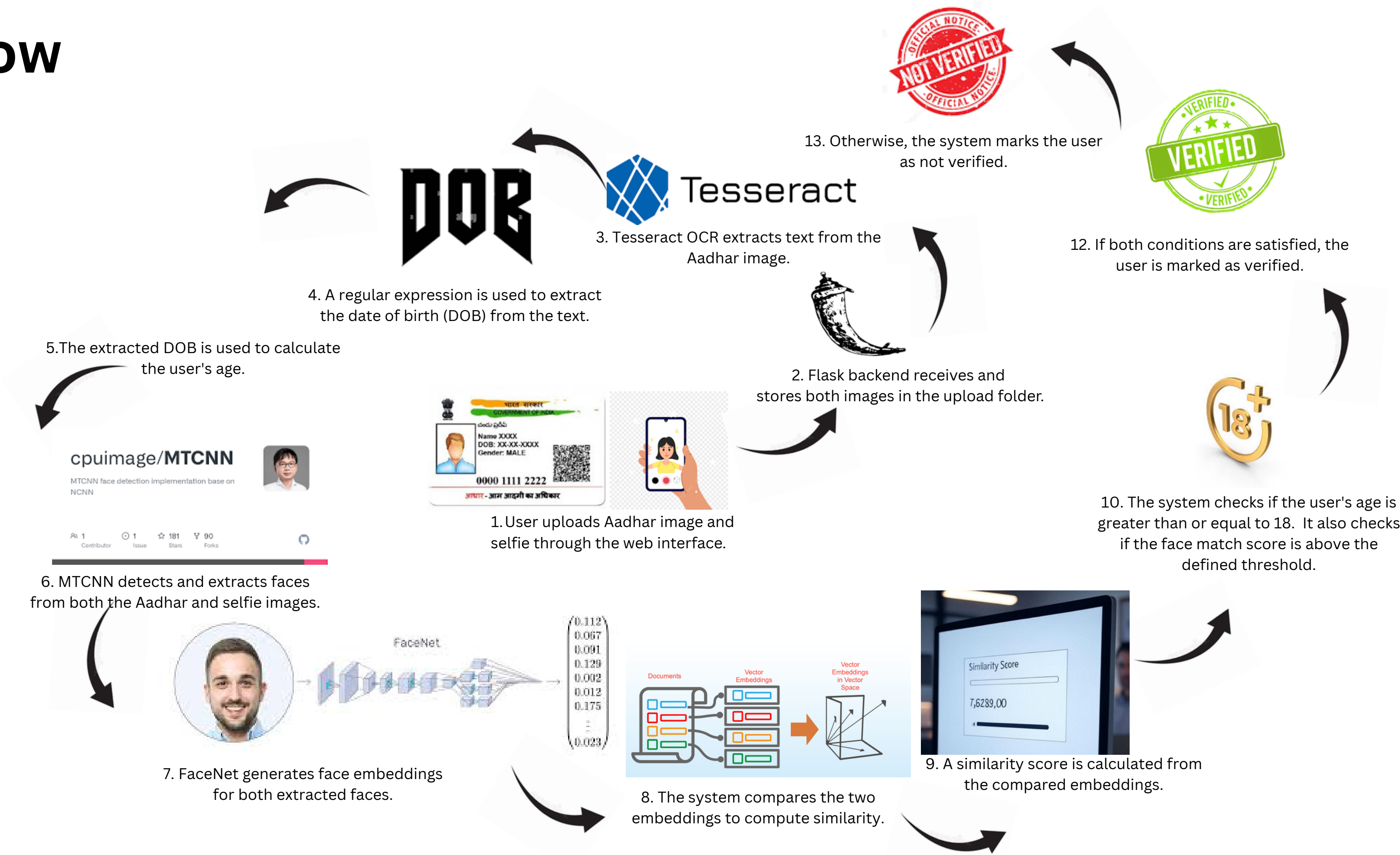
Frontend

HTML

CSS

Backend

OpenCV

Tesseract

pillow

PyTorch

NumPy

deepFace

WorkFlow

13. Otherwise, the system marks the user as not verified.

3. Tesseract OCR extracts text from the Aadhar image.

12. If both conditions are satisfied, the user is marked as verified.

4. A regular expression is used to extract the date of birth (DOB) from the text.

5. The extracted DOB is used to calculate the user's age.

2. Flask backend receives and stores both images in the upload folder.

10. The system checks if the user's age is greater than or equal to 18. It also checks if the face match score is above the defined threshold.

1. User uploads Aadhar image and selfie through the web interface.

6. MTCNN detects and extracts faces from both the Aadhar and selfie images.

7. FaceNet generates face embeddings for both extracted faces.

8. The system compares the two embeddings to compute similarity.

9. A similarity score is calculated from the compared embeddings.

Frontend Features

1. ID Card Upload

Users upload an ID card image (e.g., passport, aadhar) via a file input for backend verification.

2. Live Selfie Capture via Webcam

Users access their webcam to capture a live selfie, ensuring real-time authenticity.

3. Real-Time Video Preview and Selfie Capture

Shows live webcam feed for selfie framing; captures image with a button click.

4. Form Validation for Required Inputs

Checks for ID card and selfie before submission, alerting users if incomplete.

5. Result Display for Verification Status

Shows verification result (e.g., verified, face mismatch) in a clear, styled box.Result display for verification status

Blurry Image Detection : When user takes a selfie in poor lighting or when user has taken a blurry selfie, it is not accepted.
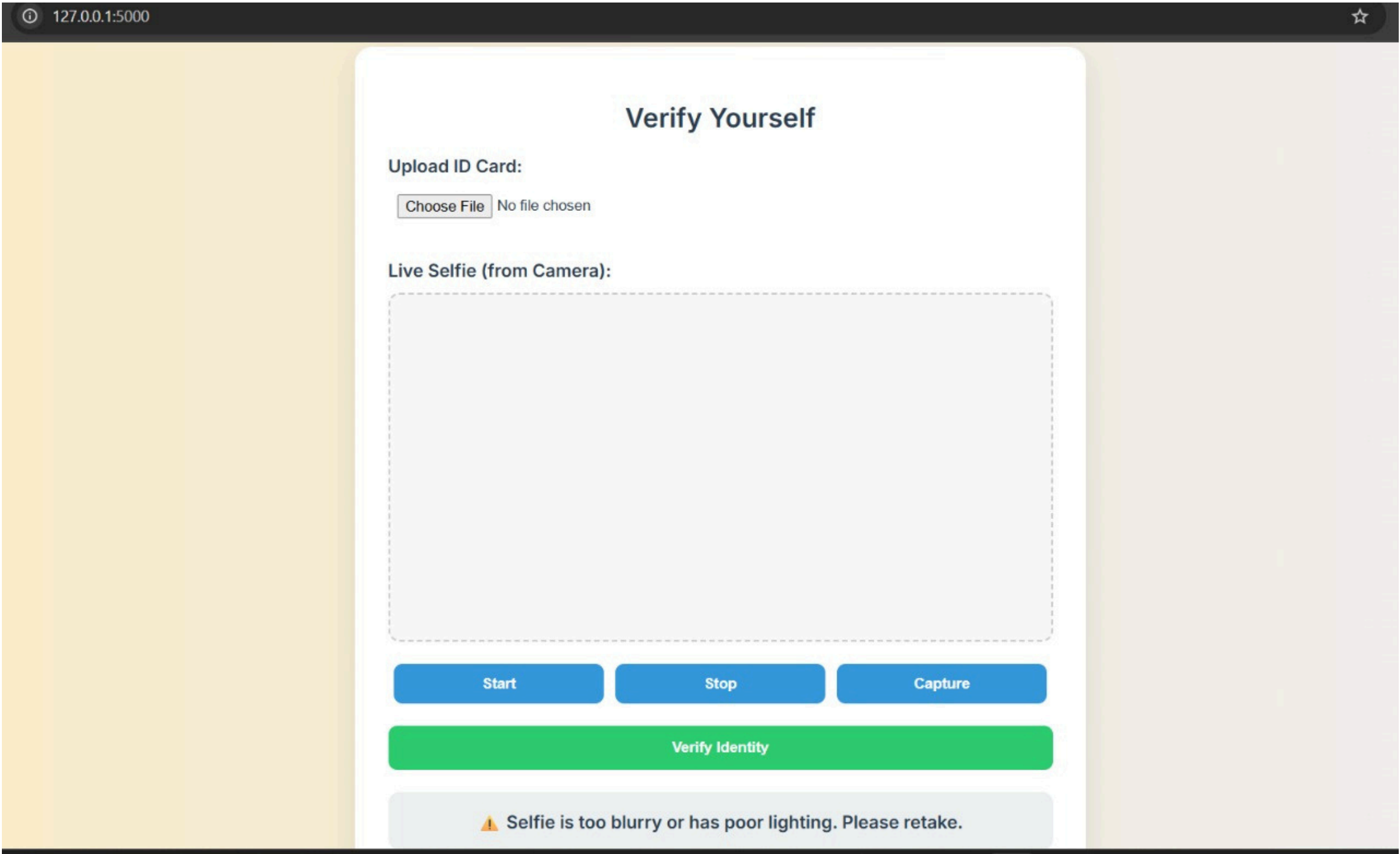
Absence of face in uploaded ID card image : "Face not detected in ID card." is shown when uploaded ID card image does not have photo.

Start

Stop

Capture

Verify Identity

❌ Face not detected in ID card.
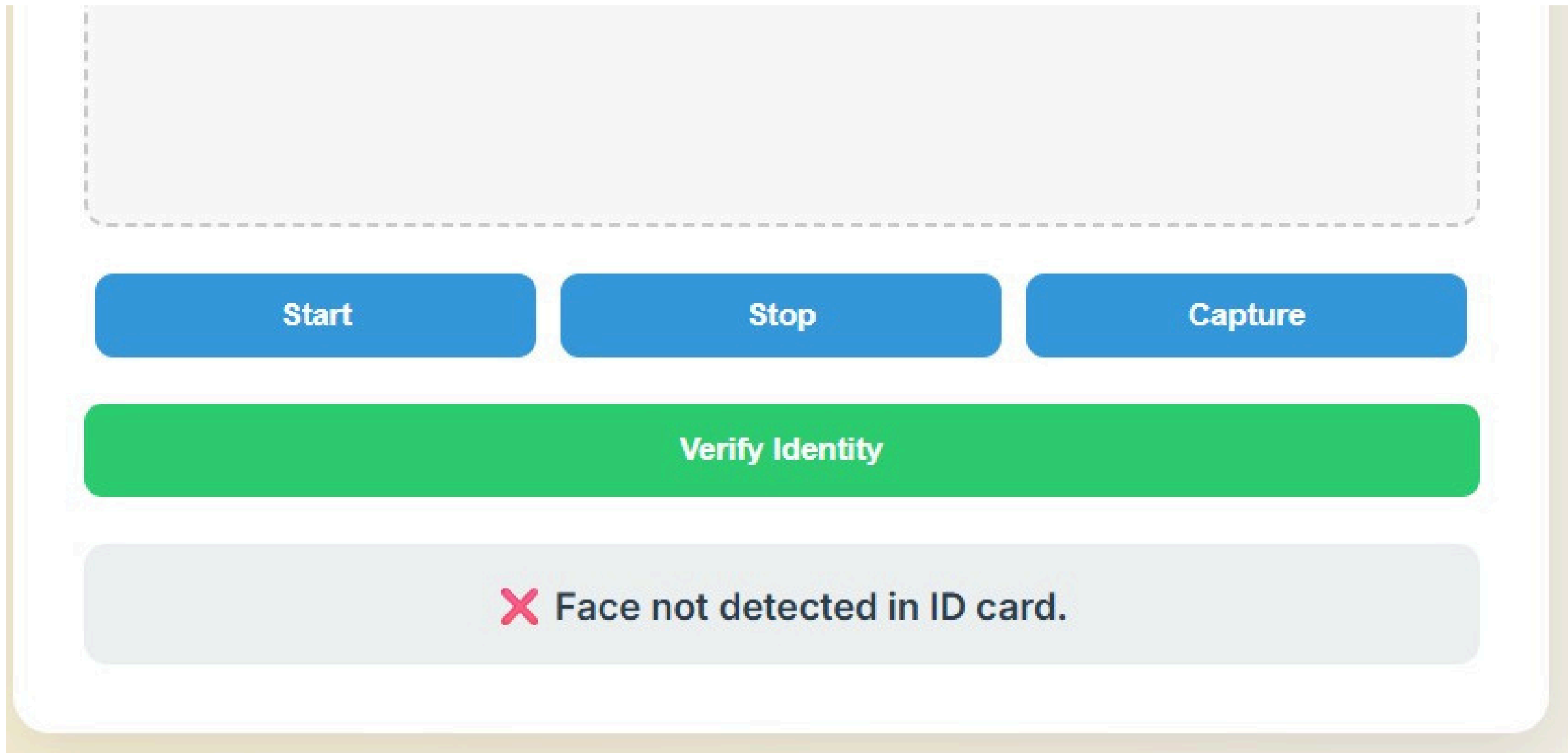
Verification score: A numerical value indicating 66% match
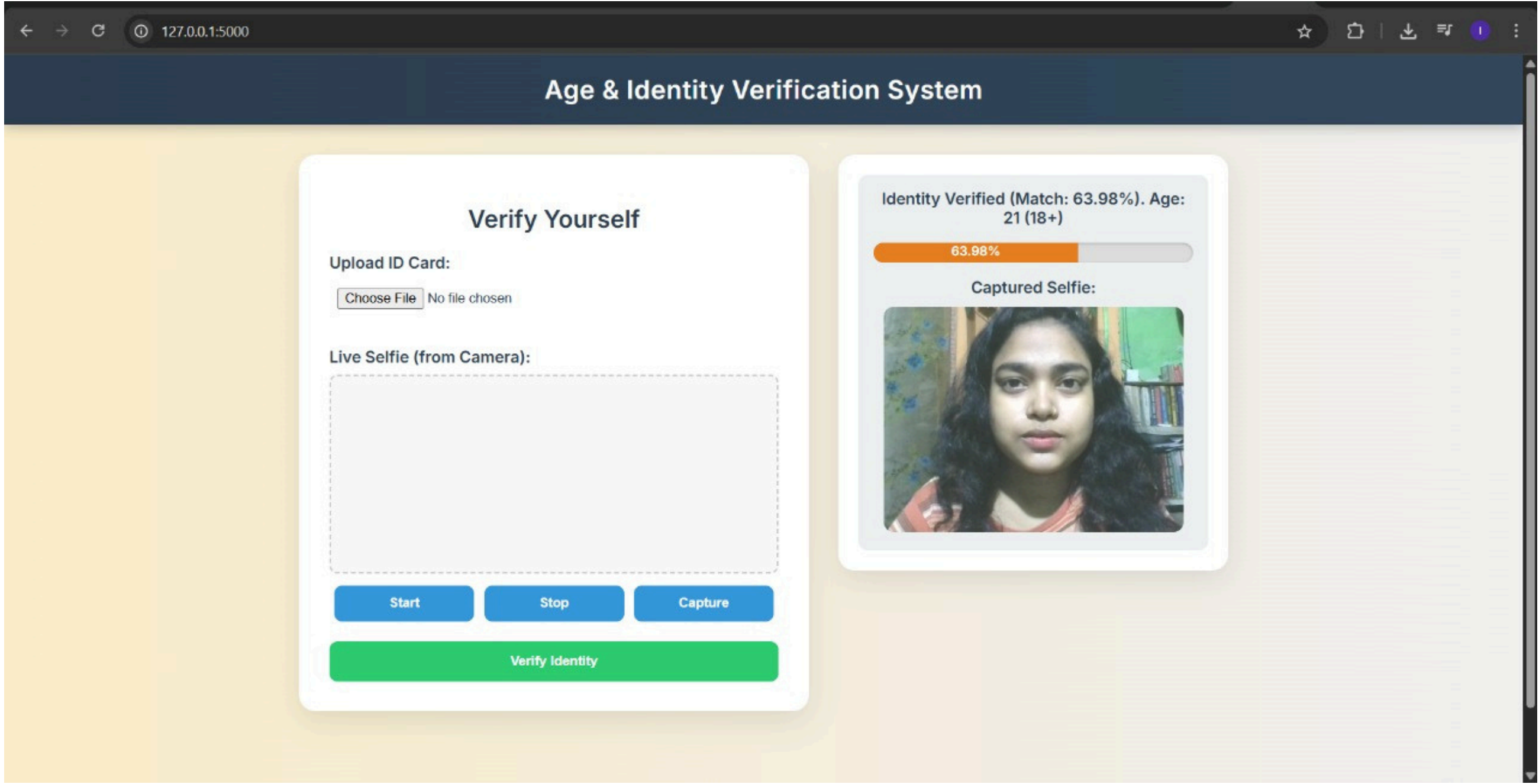
Backend Features

- Processing:
  - OCR for DOB extraction from ID card
  - Face detection and cropping
  - Face verification using DeepFace
  - Age calculation from extracted DOB

- **File Handling:** Secure upload and storage of images
- **Output:** Verification result (age, face match status)

Features of Complete Project

- **Secure Verification:** Face matching and age validation
- **User-Friendly:** Intuitive interface, real-time feedback
- **Accurate:** Robust OCR and face recognition
- **Responsive:** Mobile and desktop compatibility
- **Real-Time:** Live selfie capture for authenticity