Team: Byte Me

USER KYC PLATFORM

STGI HACKATHON

TODAY'S AGENDA

1 Problem Statement

2 Solution Architecture Overview

3 Solution Workflow Overview

4 Implementation Details

6 Key and Unique Features

7 Resource Limitations

8 Future Scope

9 DEMO

PROBLEM STMT

Key Components

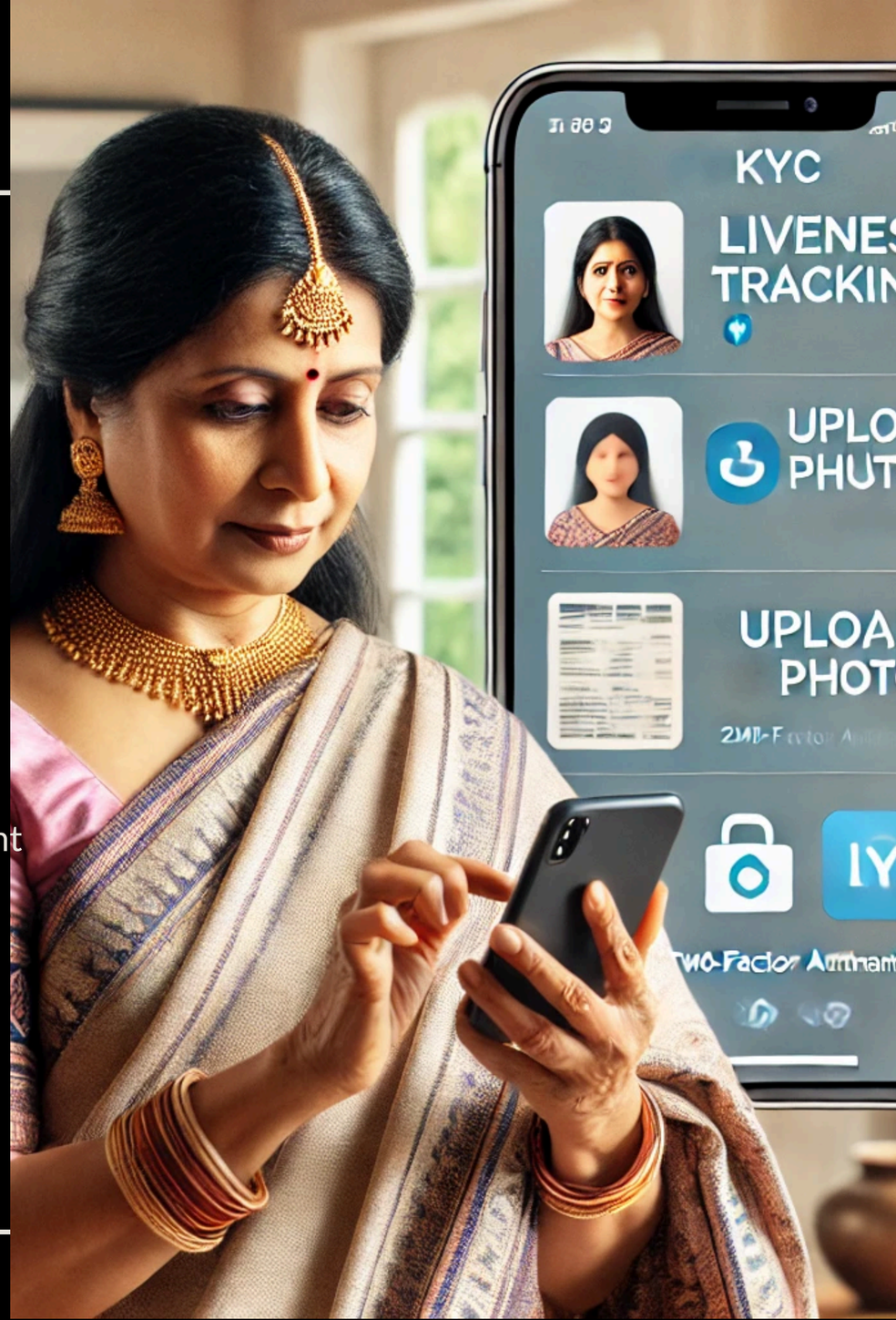
- Liveness Detection
- Facial Recognition
- 2 Step Verification
- Scalibilty

Accuracy

Lightweight

Comparision Logic

Secuirity



SOLUTION ARCHITECTURE OVERVIEW

Liveness Detection Module

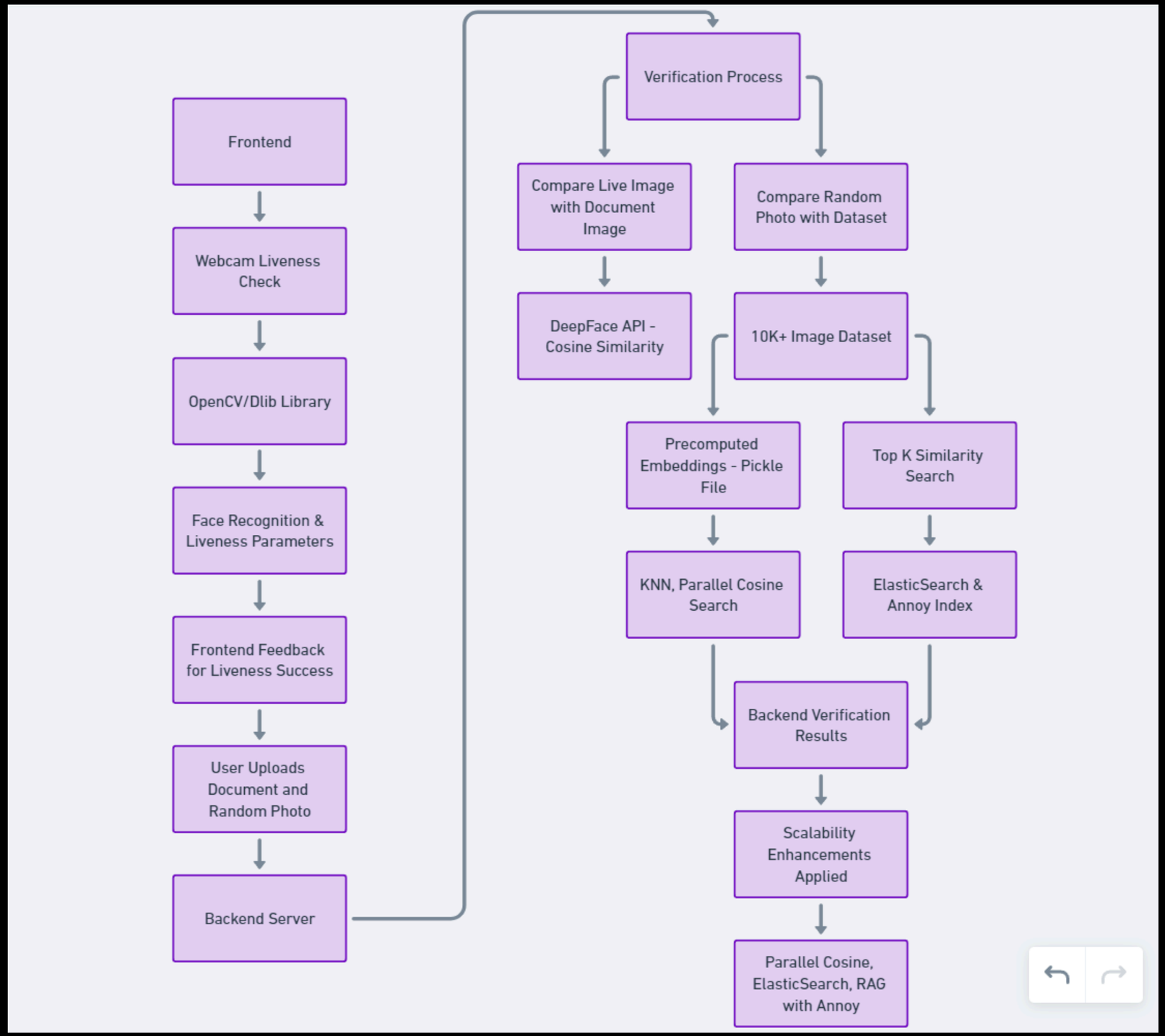
Verifies user presence through webcam by checking for blinking, head movement, and gesture recognition.

Document Verification Module

Extracts the face from the user's ID document and compares it with the live image.

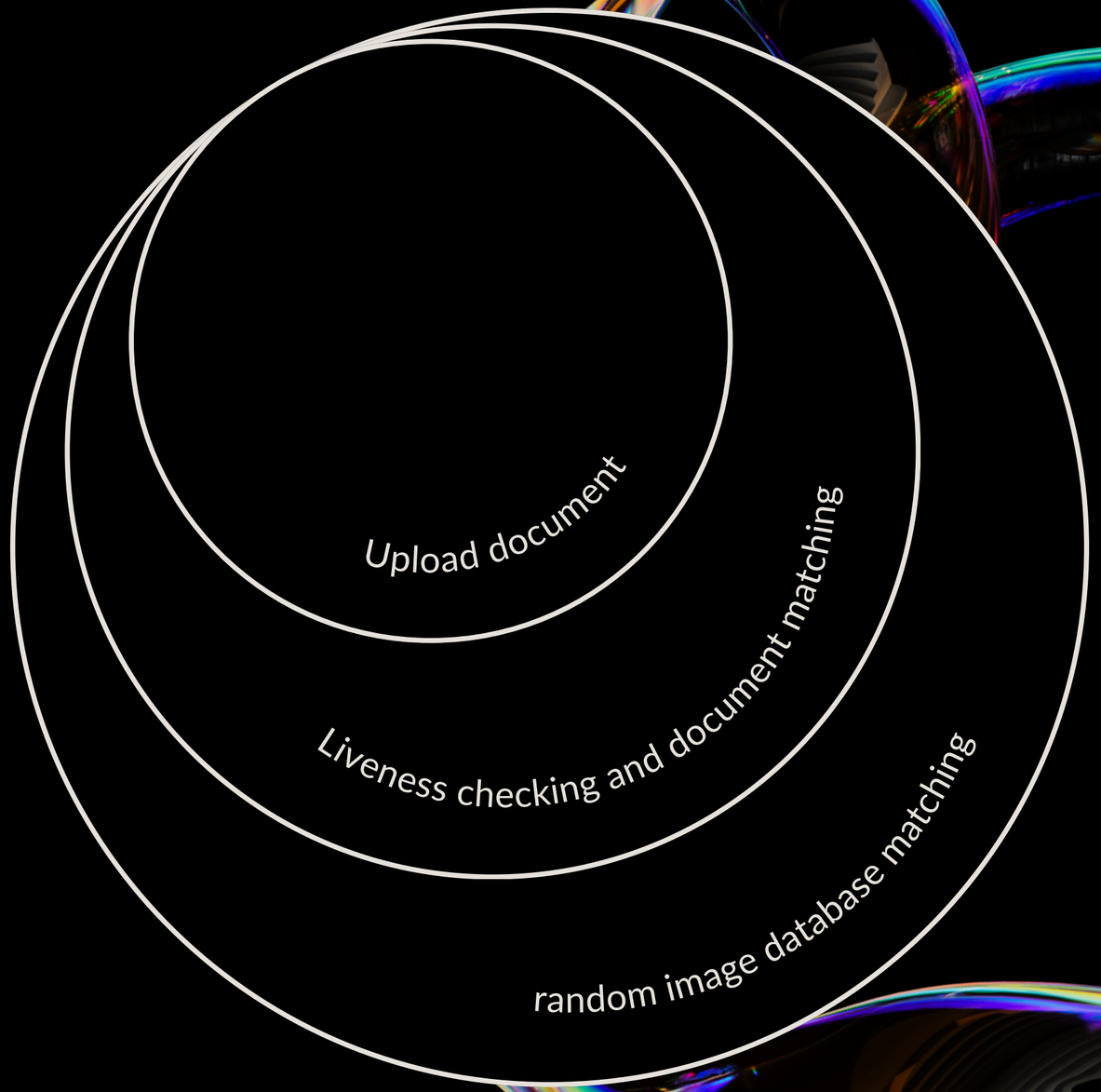
Database Matching Module

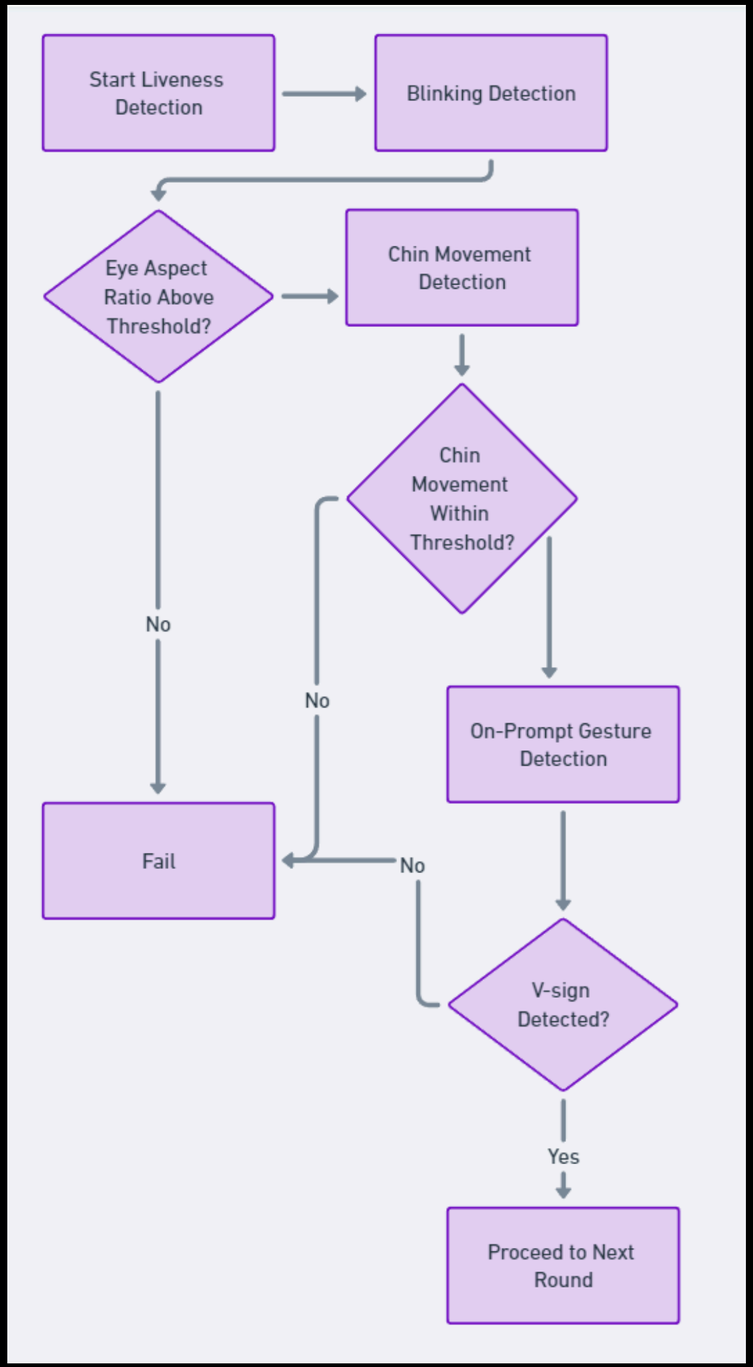
Compares the user's second image with a large dataset of images, using cosine similarity and k-NN to find the top k matches.


Frontend
Webcam Liveness Check
OpenCV/Dlib Library
Face Recognition & Liveness Parameters
Frontend Feedback for Liveness Success
User Uploads Document and Random Photo
Backend Server
Verification Process
Compare Live Image with Document Image
Compare Random Photo with Dataset
DeepFace API - Cosine Similarity
10K+ Image Dataset
Precomputed Embeddings - Pickle File
Top K Similarity Search
KNN, Parallel Cosine Search
ElasticSearch & Annoy Index
Backend Verification Results
Scalability Enhancements Applied
Parallel Cosine, ElasticSearch, RAG with Annoy

SOLUTION WORKFLOW

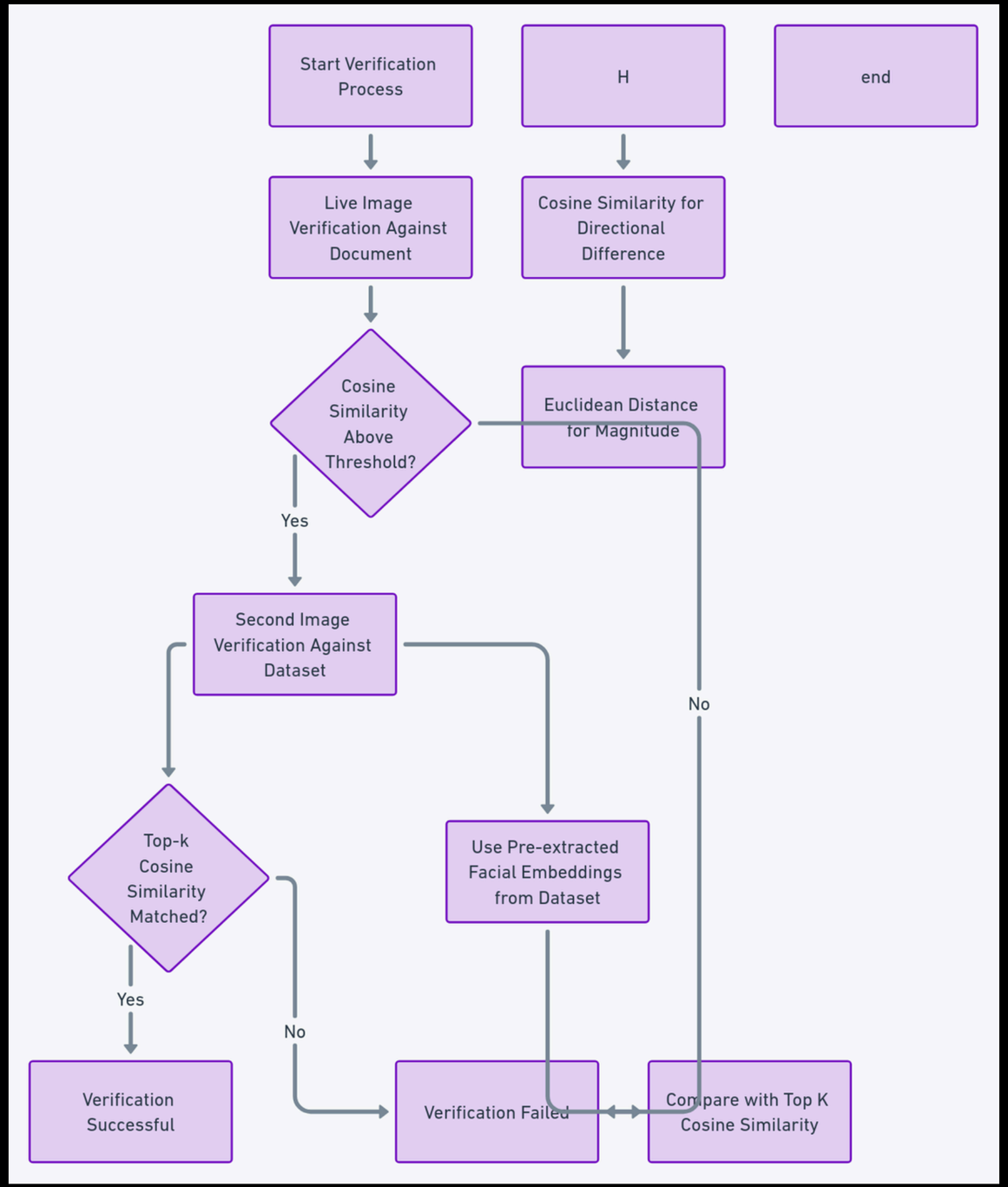
Understanding the market size is important for us. In the US, there are about 32 million small businesses. We're aiming at industries like technology, e-commerce, and professional services, which are about 30% of all small businesses. That means we're looking at around 9.6 million potential customers. Our goal is to get about 5% of them, which would be roughly 480,000 businesses. This helps us know who to target and plan our growth strategy.




Start Liveness Detection
Blinking Detection
Eye Aspect Ratio Above Threshold?
Chin Movement Detection
Chin Movement Within Threshold?
No
No
On-Prompt Gesture Detection
Fail
No
V-sign Detected?
Yes
Proceed to Next Round

Task 1

Task 2


Start Verification Process
H
end
Live Image Verification Against Document
Cosine Similarity for Directional Difference
Cosine Similarity Above Threshold?
Euclidean Distance for Magnitude
Yes
Second Image Verification Against Dataset
No
Top-k Cosine Similarity Matched?
Use Pre-extracted Facial Embeddings from Dataset
Yes
No
Verification Successful
Verification Failed
Compare with Top K Cosine Similarity

IMPLEMENTATION DETAILS

LIVENESS AND LANDMARK RECOGNITION

Liveness detection with OpenCV and Dlib tracks blinks (EAR), head movements, and on-prompt gestures (V-sign) to verify real-time user presence. API thresholds are optimized; depth detection was planned but limited by GPU constraints.

DOC VS WEB IMAGE VERIFY

We use the DeepFace library (VGG-Face, FaceNet, ArcFace) to compare live and ID document images using cosine similarity and Euclidean distance; verification is confirmed if the similarity score exceeds a set threshold.

DOCUMENT UPLOAD AND CROPPING

We use OpenCV and facial landmark detection to automatically crop the face from the ID document for accurate comparison with the live image.The second image is stored for further verification during the database comparison step.

DATABASE COMPARISION RANDOM IMAGE

The second random image is compared against a large dataset (10,000+ images) using VGG-Face embeddings stored in a pickle file for quick comparison. Top-K Cosine Similarity is applied to find the closest matches, reducing time complexity from $O(n^2)$ to $O(n)$.

OUR INNOVATIVE SOLUTIONS

Real-Time Liveness Detection with Multi-Factor Check

The platform uses a combination of blinking detection, head movement tracking, and on-prompt gesture recognition (V-sign) to ensure the user is physically present and not using a static image or pre-recorded video.

Cosine Similarity-Based Facial Verification

The system uses Cosine Similarity and Euclidean distance to compare the live image with the ID document photo, ensuring precise and reliable face matching by focusing on directional differences in the facial vectors.

Facial Embedding Storage in Pickle Files

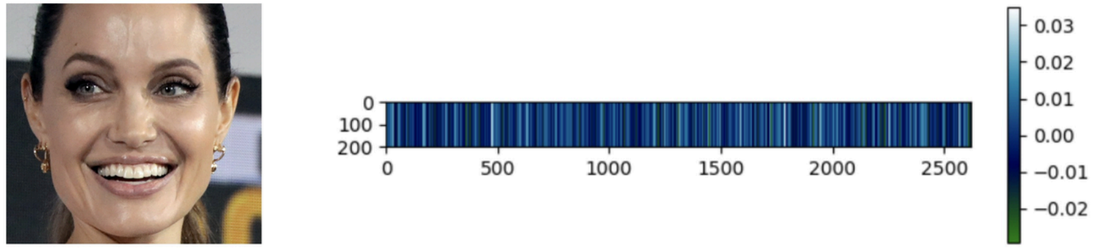
Pre-extracted facial embeddings are stored in pickle files for faster future comparisons, reducing the need for repeated real-time embedding generation and improving system efficiency.

Parallel Processing for Scalability

Utilized parallel processing to speed up the facial recognition process, allowing multiple image comparisons simultaneously and improving system scalability.

```
{
    "verified": true,
    "distance": 0.25638097524642944,
    "max_threshold_to_verify": 0.40,
    "model": "VGG-Face",
    "similarity_metric": "cosine"
}
```

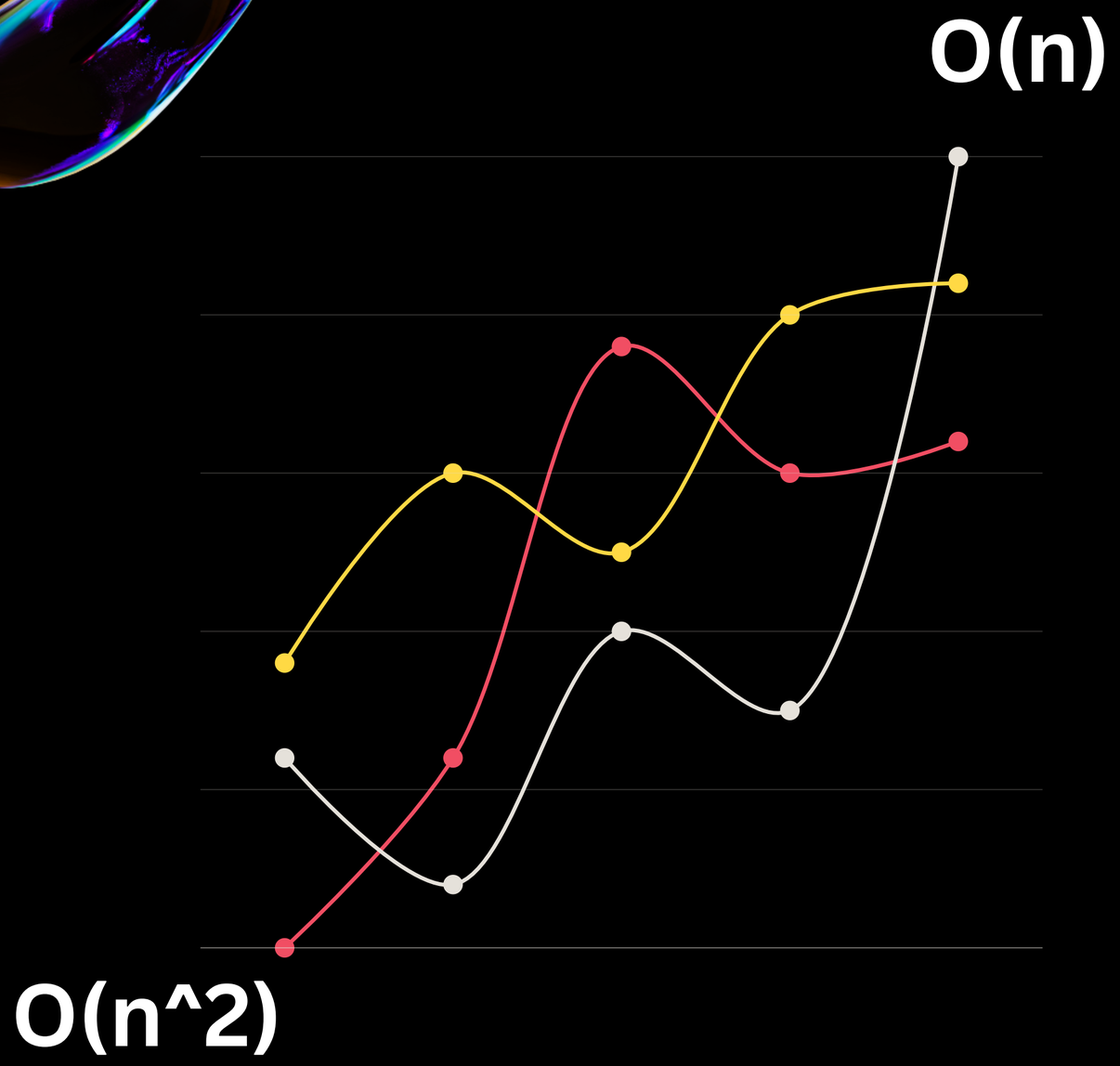

Deepface function ‘verify’ for matching using cosine values



Deepface function ‘represent’ for representing images as embeddings

Model	Measured Score	Declared Score
Facenet512	98.4%	99.6%
Human-beings	97.5%	97.5%
Facenet	97.4%	99.2%
Dlib	96.8%	99.3 %
VGG-Face	96.7%	98.9%
ArcFace	96.7%	99.5%
GhostFaceNet	93.3%	99.7%
SFace	93.0%	99.5%
OpenFace	78.7%	92.9%
DeepFace	69.0%	97.3%
DeepID	66.5%	97.4%

Why we preferred deepface over other computational models



LIGHTWEIGHT

Our solution is lightweight due to its efficient architecture that minimizes computational overhead while maintaining high accuracy.

- KNN with Top-K Cosine Similarity reduces time complexity.
- Parallel searching speeds up database queries.
- Pre-extracted embeddings avoid unnecessary real-time computations.

RESOURCE LIMITATIONS

Depth Detection:

We planned to implement depth detection to prevent spoofing with pre-recorded videos but faced GPU constraints.

AI-based Fake Image Detection:

We were unable to incorporate AI algorithms for detecting deepfakes and AI-generated images.

Advanced Liveness Detection (e.g., Micro-expressions):

We prioritize building strong relationships and understanding our clients' needs, leading to long-term partnerships based on trust and satisfaction.

Parallel Processing for Database Matching:

While planned, full implementation of parallel processing for faster image comparison in large datasets was not completed.

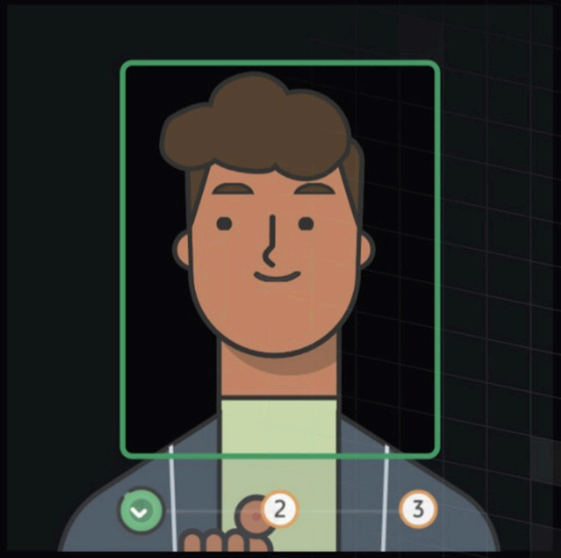
FUTURE SCOPES AND PLANS

	Liveness Detection	Verification Using DeepFace (Both Steps)	Scalability
Accuracy	- Use 3D depth features for better spoof detection.	- Integrate ResNet-50 for more accurate facial recognition.	- Implement K-means clustering for better accuracy in large datasets.
Performance	- Leverage TensorFlow.js for client-side liveness checks.	- Use ONNX to optimize model inference speed across different platforms.	- Apply Elasticsearch with Annoy for faster database lookups.
Security	- Utilize FaceNet SecureNet for encrypting liveness data in transit.	- Implement homomorphic encryption for facial embeddings in DeepFace	Not Yet
Additional Services	Integrate Azure Face API for real-time validation.	Not yet	Not yet

KYC platform

Upload

Dummy



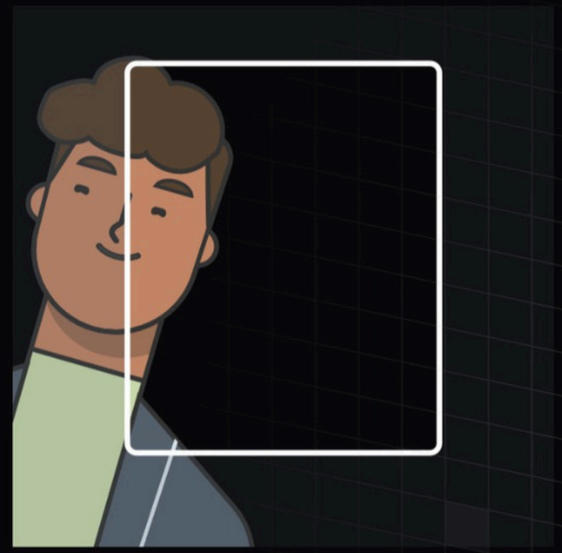
Please complete your KYC process now for a seamless experience.

Get Started

KYC platform

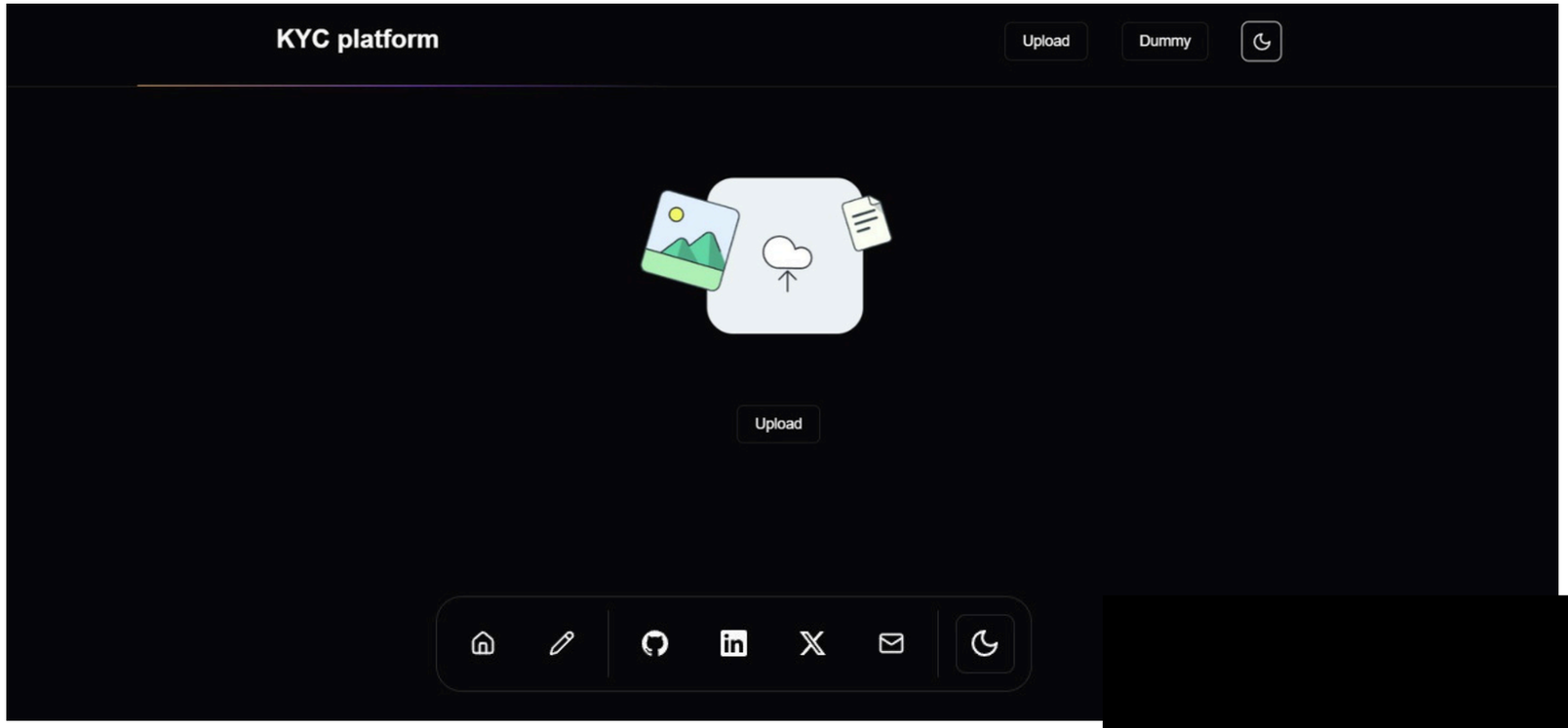
Upload

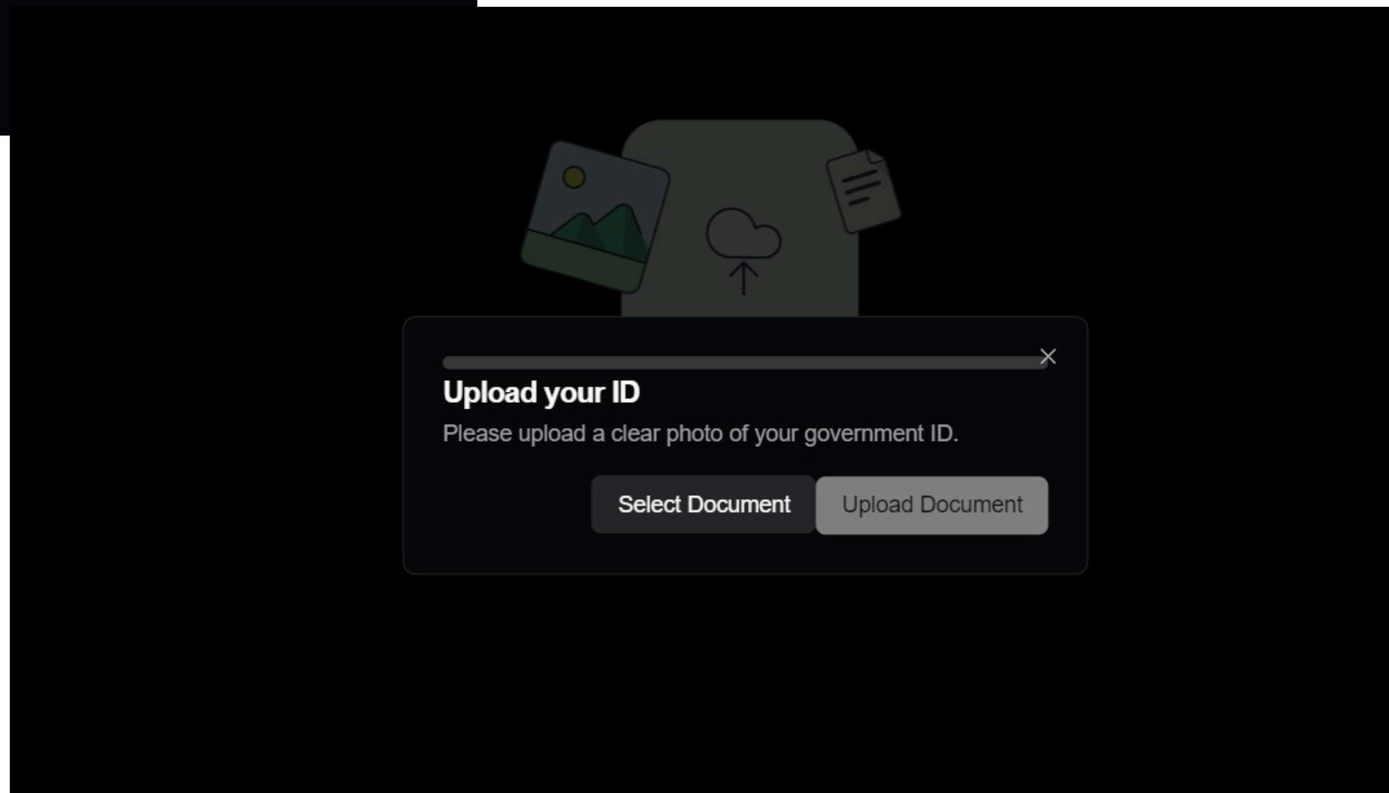
Dummy

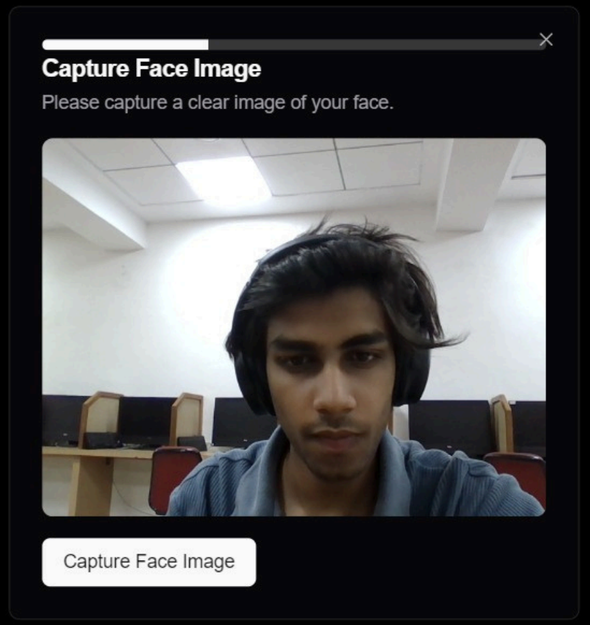


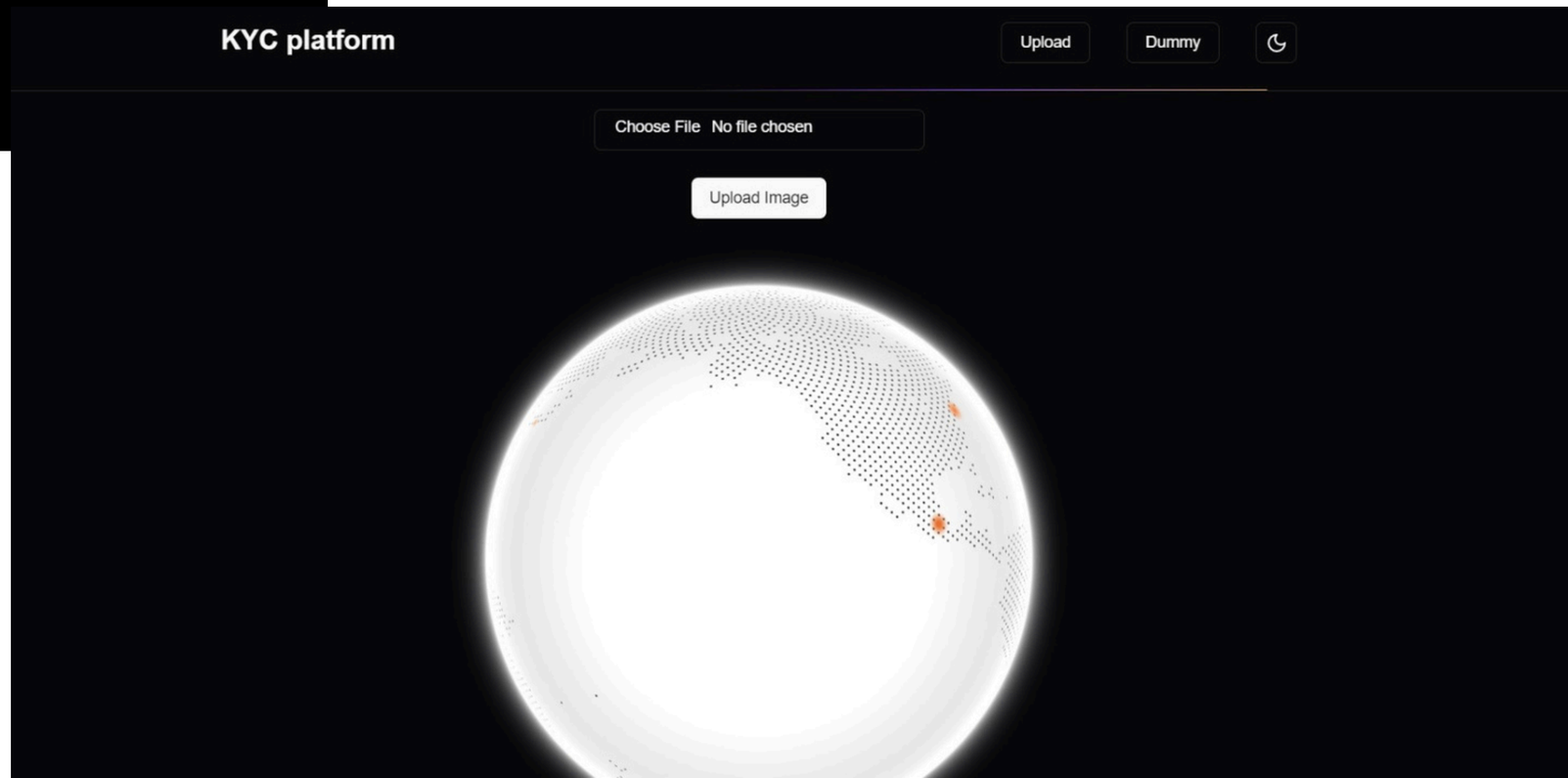
Please complete your KYC process now for a seamless experience.

Get Started


KYC platform
Upload
Dummy
Upload


Upload your ID
Please upload a clear photo of your government ID.
Select Document
Upload Document


Capture Face Image
Please capture a clear image of your face.
Capture Face Image


KYC platform
Upload
Dummy
Choose File No file chosen
Upload Image

Team Name: Byte Me

THANK YOU

Arav Gupta
Sarthak Kapila
Armaan Verma
Shauray Dhingra
Lakshya Vema

Please refer to the demo video as well as the documentation provided with th repo.