AN

INDUSTRIAL ORIENTED MINI PROJECT REPORT

ON

**WEB BASED FACIAL AUTHENTICATION FOR LOGIN**

Submitted to JNTUH in the partial fulfillment of the Academic Requirements for the award of the degree of

**BACHELOR OF TECHNOLOGY**

IN

**COMPUTER SCIENCE AND ENGINEERING**

BY

**R.ANJALI (21QM1A05A0)**

**UNDER THE GUIDANCE OF**

**Mrs MANCHALA MANASA**

**Assistant Professor**

**Department of Computer Science & Engineering**

****

**DEPARTMENT COMPUTER SCIENCE AND ENGINEERING**

**K.G REDDY COLLEGE OF ENGINEERING & TECHNOLOGY**

**(Approved by AICTE, Affiliated to JNTU Hyderabad)**

Chilkur (V), Moinabad Mandal, R.R. Dist-501504, Ph: 9247033008,

9000633008.Website: www.kgr.ac.in Batch: 2021-2025

****

**CERTIFICATE**

This is to certify that the project report entitled “**WEB BASED FACIAL RECOGNITION SYSTEM FOR LOGIN**” that is being submitted by  **R.ANJALI (21QM1A05A0)** under the guidance of **Mrs M.MANASA** with fulfillment for the award of the Degree of **Bachelor of Technology in Computer Science and Engineering** to the Jawaharlal Nehru Technological University isa record of bonafide work carried out by his under my guidance and supervision. The results embodied in this project report have not been submitted to any other University or Institute for the award of any graduation degree.

**INTERNAL GUIDE HEAD OF DEPARTMENT**

**Dr. M. MANASA Dr. L. RAGHU KUMAR**

**Associate Professor Hod & Associate Professor**

**Department Of CSE Department Of CSE**

**EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

This is to place on our record my appreciation and deep gratitude to the persons without whose support this project would never been this successful.We would like to express our profound sense of gratitude and thanks to our Director **Dr. ROHIT KANDAKATLA** of KGRCET for his guidance, encouragement and for all the facilities to complete this project. We have immense pleasure in expressing thanks and deep sense of gratitude valuable time with us and laying down his valuable suggestions to complete the project successfully on time.

It is with immense please that we would like to express our indebted gratitude to **Dr.S.SAI SATYANARAYANA REDDY** , Principal, KG Reddy College of Engineering & Technology, for providing a great support and for giving us the opportunity of doing the project.

At the same time, we feel elated to the**, Dr.L.RAGHU KUMAR**, Associate Professor & Head of the Department, CSE, KG Reddy College of Engineering & Technology, for inspiring us all the way and for arranging all the facilities and resources needed for our project.

We would like to take this opportunity to thank our internal guide **Mrs. M.MANASA**, Assistant Professor, Department of CSE, KG Reddy College of Engineering & Technology, who has guided us a lot and encouraged us in every step of the project work. His valuable moral support and guidance throughout the project helped us to a greater extent.

We would like to take this opportunity to specially thank our PRC Coordinator, **Mr. Surendra Tripathi**, **Assistant Professor, Department of CSE, KG Reddy College of Engineering & Technology**, who guided us in our project.

Finally, we express our sincere gratitude to all the members of the faculty of Department of Computer Science and Engineering, our friends and our families who contributed their valuable advice and helped us to complete the project successfully.

**Team Member**

RANKIREDDI ANJALI – 21QM1A05A0

**DECLARATION**

This is to certify that the major project titled “**WEB BASED FACIAL AUTHENTICATION FOR LOGIN**” is a bonafied work done by us in fulfillment of therequirements for the award of the degree **Bachelor of Technology** in **Computer Science and Engineering** and submitted to the **Department of CSE**, KG ReddyCollege of Engineering and Technology, Chilkur, Moinabad, Hyderabad.

We also declare that this project is a result of our own effort and has not been copiedor intimated from any source. Citations from any websites are mentioned in thebibliography. This work was not submitted earlier at any other university for the awardof any degree.

**Team Member**

RANKIREDDI ANJALI – 21QM1A05A0

**INDEX PAG**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **CONTENTS** | **PAGE NO** |
| 1 | INTRODUCTION | 6-8 |
| 2 | SYSTEM ANALYSIS | 9-13 |
| 3 | SOFTWARE REQUIREMENT & SPECIFICATIONS | 14-15 |
| 4 | SYSTEM DESIGN | 16-25 |
| 5 | SYSTEM IMPLEMENTATION | 26-29 |
| 6 | SYSTEM TESTING | 30-31 |
| 7 | OUTPUT SCREENS | 32-39 |
| 8 | CONCLUSION | 40 |
| 9 | FUTURE ENHANCEMENT | 40 |
| 10 | REFERENCES | 41 |

|  |  |  |
| --- | --- | --- |
| **Figure No** | **Figure** | **Page No** |
| **1** | Use-case diagram | 18 |
| **2** | Class diagram | 19 |
| **3** | Activity diagram | 21 |
| **4** | Sequence diagram | 22 |
| **5** | Collaboration diagram | 24 |
| **6** | Deployment diagram | 25 |
| **7** | Output Screen | 32 |

**OF FIGURES**

**ABSTRACT**

The web-based facial authentication system represents a cutting-edge solution for secure user verification in web applications. Harnessing advanced computer vision and machine learning techniques, the system captures and analyzes facial features to establish a user's identity.

After initiating the process, the system captures and pre-processes facial images, extracting unique

facial attributes and landmarks. These features are then matched against pre-enrolled templates stored in a secure database, leveraging robust algorithms such as Eigen faces or Convolutional Neural Networks (CNNs) for accurate verification. To enhance security, the system incorporates anti spoofing measures, including liveness detection, to thwart fraudulent attempts.With a web based architecture, the system seamlessly integrates into various web platforms , providing a user friendly and password-free experience. cost-effectiveness, and global availability, enabling rapid deployment and streamlined operations. Comprehensive testing confirms the system's high accuracy, adaptability, and resilience against potential threats, making it a robust and efficient solution for modern web-based applications. In this hands-on project, we will be using FaceIO API to implement facial recognition authentication in a React web application.With the help of the fio.js JavaScript library.

**1. INTRODUCTION**

**1.1 PURPOSE**

The purpose of this project is to enhance security and user experience in web-based authentication systems by replacing traditional password-based methods with facial recognition technology. Traditional systems are vulnerable to issues such as weak passwords, password reuse, and phishing attacks. By utilizing unique facial data, this project aims to provide a robust and secure authentication method. Additionally, it seeks to simplify the login process, making it quicker and more user-friendly by eliminating the need for complex passwords. The system is designed to work with standard webcams, ensuring no need for additional hardware. Ultimately, this project aims to mitigate the risks associated with password-based systems, offering a secure, efficient, and convenient alternative for user authentication.

**1.2 SCOPE**

The scope of this project includes the development of a web-based application that uses the Face IO API to implement facial recognition authentication. This involves creating a React web application and integrating the fio.js JavaScript library for facial recognition.

The system finds applications in finance, healthcare, education, and travel, offering enhanced security

and convenience. Additionally, it enables personalized experiences in social media, customer service,

and digital marketing. As technology evolves, the scope may extend to fraud detection, smart homes,

and targeted advertising. However, privacy and data security are critical considerations for

responsible implementation across diverse industries and domain. The system can provide enhanced

security, improved user experience, regulatory compliance, integration with existing systems, and

potential for future expansion.

**1.3 GOALS**

The goals of this project are:

* To enhance security by providing an additional layer of authentication.
* To improve user experience by allowing users to log in without entering complex passwords.
* To develop a seamless and convenient authentication system that is fast and user-friendly.
* Facilitate Integration and Scalability Offer easy integration with existing applications and scalable architecture for growing user bases.

**1.4 FEATURES OF OUR PROJECT**

* Enhanced Security: Provides an additional layer of security compared to traditional password-based systems.
* Improved User Experience: Offers a seamless and convenient login process by simply presenting the face for authentication.
* Accuracy and Speed: Typically takes only a few seconds to verify a person's identity,

making it a swift authentication method.

* User-Friendly: Most people are familiar with the concept of facial recognition from using it on their smartphones, making it easy to adopt.
* Integration and Scalability: Easily integrates with existing web applications and supports scalable architecture for growing user bases.
* Multi-Factor Authentication (MFA): Supports layered security by combining facial recognition with other authentication factors.
* Audit and Monitoring: Includes activity logs and real-time alerts for monitoring and enhancing security.
* Accessibility Features: Designed to be inclusive and support users with disabilities.
* Compliance and Standards: Adheres to industry standards and regulations for data protection and privacy.

This information outlines the key aspects and intentions of the web-based facial authentication system project, highlighting its purpose, scope, goals, and main features.

**2. SYSTEM ANALYSIS**

**2.1 EXISTING SYSTEM**

**Vulnerability To Password-Related Attacks:**

* Traditional web-based authentication systems relying on passwords are susceptible to various attacks such as brute force attacks, dictionary attacks, and credential stuffing.
* Weak or reused passwords can be easily compromised, leading to unauthorized access.

**User inconvenience:**

* Users often struggle with password management, including remembering complex passwords or constantly resetting forgotten passwords.
* Users value their privacy, so systems should implement strong privacy protections, including secure storage of facial data and transparent policies regarding data usage.

**2.2 PROPOSED SYSTEM**

* **Enhanced security**: Facial authentication provides an additional layer of security compared to traditional password-based systems.
* I**mproved user experience**:Facial authentication offers a seamless and convenient user experience. Users can simply present their face for authentication, eliminating the need to remember and enter complex passwords.This results in quicker and more user-friendly authentication processes.

**2.3 OVERALL DESCRIPTION**

The existing system uses traditional password-based authentication, which is vulnerable to attacks and inconvenient for users who struggle with password management. These systems can be costly to set up and integrate, and storing facial data raises privacy concerns.

The proposed system introduces web-based facial authentication to enhance security and user experience. This system provides an additional security layer and a seamless, user-friendly authentication process, eliminating the need for complex passwords. It ensures quick and accurate identity verification, leveraging familiar facial recognition technology from smartphones. This approach addresses the shortcomings of the existing system, offering a secure, efficient, and user-friendly solution.

**2.4 FEASIBILITY STUDY**

The feasibility study evaluates the practicality of implementing the web-based facial authentication system. It considers various aspects such as technical, economic, operational, and legal feasibility to ensure the project's success.

Technical Feasibility:

* Technology Availability: The system utilizes existing technologies such as facial recognition algorithms, web development frameworks, and APIs (e.g., Face IO API).
* System Requirements: The required hardware and software are readily available and compatible with current technology standards.

Economic Feasibility:

* Cost-Benefit Analysis: The initial investment in developing and deploying the system is justified by the anticipated benefits, including enhanced security and improved user experience.
* Resource Allocation: Adequate resources, including budget and personnel, are allocated to ensure the project's completion within the stipulated timeframe and budget.

Operational Feasibility:

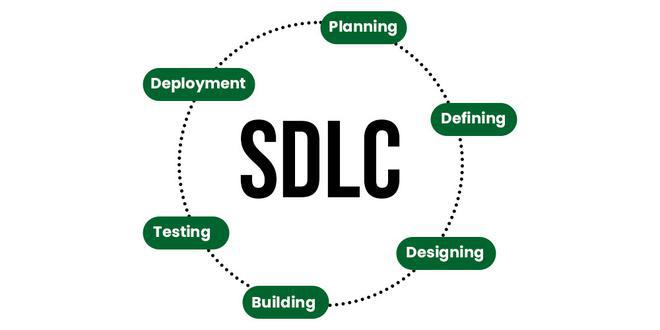
* User Acceptance: Users are likely to accept and adapt to the new authentication system due to its convenience and familiarity with facial recognition technology.
* Support and Training: Sufficient support and training will be provided to users and administrators to ensure smooth operation and maintenance of the system.

Legal Feasibility:

* Data Protection Regulations: The system complies with relevant data protection and privacy regulations, ensuring that users' facial data is securely stored and processed.
* Legal Constraints: No significant legal barriers are anticipated that would hinder the development or deployment of the system.

**2.5.SDLC MODEL**

An **SDLC model** (Software Development Life Cycle model) is a **formalized framework** that defines a **structured process for the software development lifecycle.** It outlines the **distinct phases** involved in creating software, from initial conception and planning to final deployment, testing, and ongoing maintenance.

****

**Fig-1:Software development lifecycle**

1. **Requirement Analysis :**
   * Identify user needs and security requirements.
   * Determine the specifications for the facial authentication feature.
2. **System Design :**
   * Design the architecture of the web application and the integration of the FaceIO API.
   * Plan the database schema and define the workflow for authentication.
3. **Implementation (Web App & FaceIO API Integration) :**
   * Develop the web application using React.
   * Integrate the FaceIO API for facial recognition functionality.
4. **Testing (Unit, Integration, Functional) :**
   * Perform unit testing on individual components.
   * Conduct integration testing to ensure components work together.
   * Execute functional testing with real facial images to validate overall system performance.
5. **Deployment (Server & Client-side):**
   * Deploy the application on a web server.
   * Ensure client-side compatibility across different browsers and devices.
6. **Maintenance:**
   * Regularly update the system for security patches and feature enhancements.
   * Monitor system performance and user feedback to make necessary improvements.

This model visually represents the key phases in developing and maintaining a secure, user-friendly web-based facial authentication system.

**3.SOFTWARE REQUIREMENT & SPECIFICATIONS**

**3.1.SOFTWARE REQUIREMENTS**

* **Operating System :** Windows 11 or Latest One
* **Text Editor :** Vs Code or Any One
* **Browser :** Chrome or Any Latest One
* **Others :** Node Js

3.2.**HARDWARE REQUIREMENTS**

* **Processor :** i5 11th Gen or Above
* **RAM** : 4GB or Above
* **Storage** : 10GB or Above

**3.3.COMMUNICATION INTERFACES**

* **User Interface**: Web interface for user interaction, initiating authentication process, granting camera access, providing feedback.
* **FaceIO API Integration**: Utilizing fio.js library to communicate with FaceIO API for facial recognition capabilities, image capture, processing, and matching.
* **Backend Server**: Handling user management, facial data storage/retrieval, application logic via HTTP/HTTPS communication.
* **Database Interface:** Secure storage and retrieval of user information, facial data templates.
* **External Integration**: Potential integration with identity management, access control, logging systems using REST APIs, messaging.
* **Admin Interface**: Management and monitoring interface for administrators, user management, configuration, logs, reports.
* **Compliance Interface**: Communication with regulatory bodies for data protection, privacy compliance, data sharing, auditing.

**4.SYSTEM DESIGN**

**4.1 DESIGN OVERVIEW**

The design of the web-based facial authentication system focuses on enhancing security and user experience by using facial recognition technology. It replaces traditional password-based methods with a more secure and user-friendly approach. The system integrates key components such as a user interface for interaction, an authentication server for processing requests, a database server for storing user data, and a facial recognition engine for image capture and feature extraction. This holistic design ensures seamless, efficient, and secure authentication for users.

The overview design of the web-based facial authentication system focuses on providing enhanced security and user experience. By utilizing facial recognition technology, the system aims to replace traditional password-based authentication methods, which are often vulnerable to various attacks and user management challenges. The key components of the system include:

* **User Interface (Web Browser)**: This is where users interact with the system to perform authentication.
* **Authentication Server**: Responsible for handling authentication requests and validating user identities.
* **Database Server**: Stores user data, including facial features and authentication logs.
* **Facial Recognition Engine**: Processes images, extracts facial features, and compares them for authentication purposes.

This architecture ensures that the system operates efficiently and securely, providing a seamless authentication process for users.

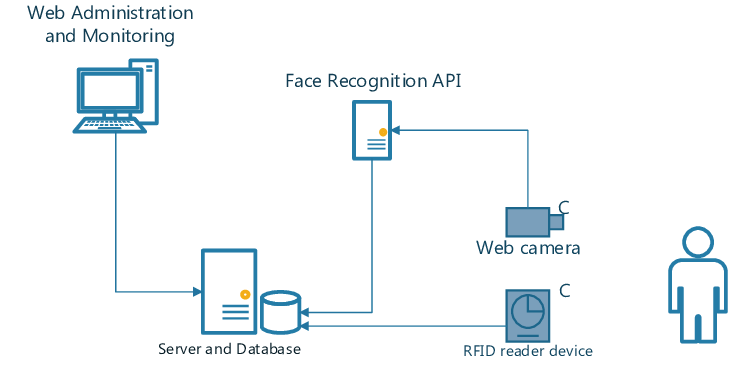
**4.2 SYSTEM ARCHITECTURE**

1. **User Interface (Web Browser):**
   * The interface through which users interact with the system.
2. **Authentication Server:**
   * Handles the authentication requests from the user interface.
   * Manages communication with other system components for user authentication.
3. **Database Server:**
   * Stores user data, including facial images and other relevant information.

**4. Facial Recognition Engine:**

* + The core component responsible for processing and verifying facial images.

Includes sub-modules for image capture and preprocessing.



**fig-2: System Architecture**

The above image shows a diagram of a web administration and monitoring system with a face recognition app, a web camera, a server and database, and a rfid reader device.

* A web camera captures a face image of a person.
* The face image is then sent to a face recognition API, likely running on a server.
* The face recognition API compares the face image to a database.
* Based on the comparison, the system grants or denies access. The database might contain authorized users’ faces.
* An RFID reader device might also be used to grant or deny access.

**4.3 MODULES DESCRIPTION**

In a web-based facial authentication system, there can be several modulesthat work together

to perform varioustasks. Here are some common modules that can be included:

**1. Face Detection Module:** This module is responsible for detecting the presence of faces in captured images or video streams from the user's device camera. Identifies the location and boundaries of faces within the input data. May utilize algorithms like Viola-Jones or Multi-Task Convolutional Neural Networks (MTCNN) for accurate face detection.

**2. Face Recognition Module**: Performs the core task of recognizing and identifying individuals based on their unique facial features. Extracts facial attributes and landmarks from detected faces. Compares extracted features against pre-enrolled facial data templates stored in the database. Leverages robust algorithms such as Eigenfaces, FaceNet, or Convolutional Neural Networks (CNNs) for accurate facial recognition. Utilizes the FaceIO API and the fio.js JavaScript library for facial recognition capabilities.

**3. Liveness Detection Module**: Liveness detection is used to ensure that the detected face is froma live person and not a photograph or video. This module can employ various techniques, such as analyzing facial movements or requesting the user to perform specific actions.

**4. Data Storage and Management Module :** This module handles the storage and management offacial data, including storing the face prints or templates of enrolled individuals, managing databases, and handling data privacy and security.

**5. User Interface Module**: The user interface module provides the interface for users to interactwith the facial authentication system. It can include features such as capturing images or videos, displaying authentication results, and providing feedback to the user.

**6. Integration with APIs or SDKs Module:** Depending on the requirements of the system, additional modules may be needed to integrate with external APIs or softwaredevelopment kits (SDKs) for specific functionalities, such as accessing facial recognition algorithms or utilizing pre-trained models.

**4.4 DFD DESIGN**

The Data Flow Diagram (DFD) outlines the flow of data within the system:

Context Level DFD (Level 0):

* Process : Web-Based Facial Authentication System
* **External Entities :**
  + User (Web Browser)
  + Facial Recognition Engine
  + User Database
* **Data Flows :**
  + User -> System: Login request (username/password, facial image)
  + System -> Facial Recognition Engine: Facial image for recognition
  + Facial Recognition Engine -> System: Recognition result (match/no match)
  + System -> User Database: Username for authentication (if recognition successful)
  + User Database -> System: Authentication result (authorized/unauthorized)
  + System -> User: Login response (success/failure message)

**Level 1 DFD :**

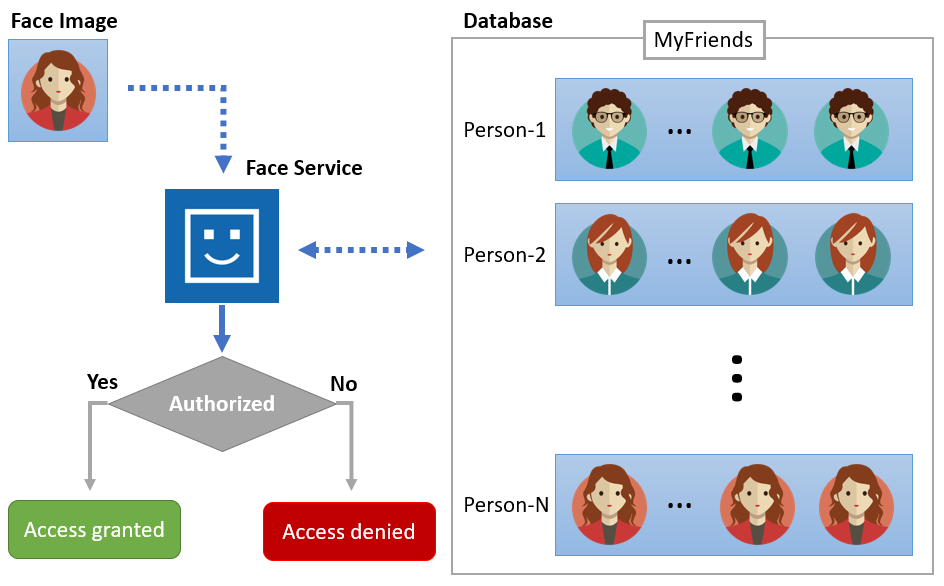
* **Process:** Web-Based Facial Authentication System

**Data Stores :**

* User Database: Stores user credentials (username, password, hashed facial features)
* Session Data (optional): Stores temporary user session information after successful login

**Data Flows :**

* User -> System: Login request (username/password, facial image)
* System -> User Database: Username and password for verification
* User Database -> System: Authentication result (valid credentials/invalid credentials)
* (If invalid credentials) System -> User: Login failure message
* System -> Facial Recognition Engine: Facial image for recognition (if credentials valid)
* Facial Recognition Engine -> System: Recognition result (match/no match, confidence score)
  + (If no match or low confidence) System -> User: Login failure message
* System -> User Database: User ID (if recognition successful)
* User Database -> System: User information and access rights (optional)
* System -> Session Data (optional): Create user session data (if successful)
* System -> User: Login success message (and potentially user-specific information)



**Fig-3:Dataflow model**

This diagram illustrates a face recognition system. It begins with a user submitting a face image for identification. The system then compares this image to a database likely containing recognized faces, possibly titled MyFriends. Based on this comparison, the system grants access if there's a match, or denies it if the face isn't recognized.

**4.5.UML DESIGN**

The Unified Modeling Language (UML) is a standard language for writing softwareblueprints. It can depict various aspects of the system, including its structure, behavior, and the way different parts interact.

The UML is a language for:

• Visualizing

• Specifying

• Constructing

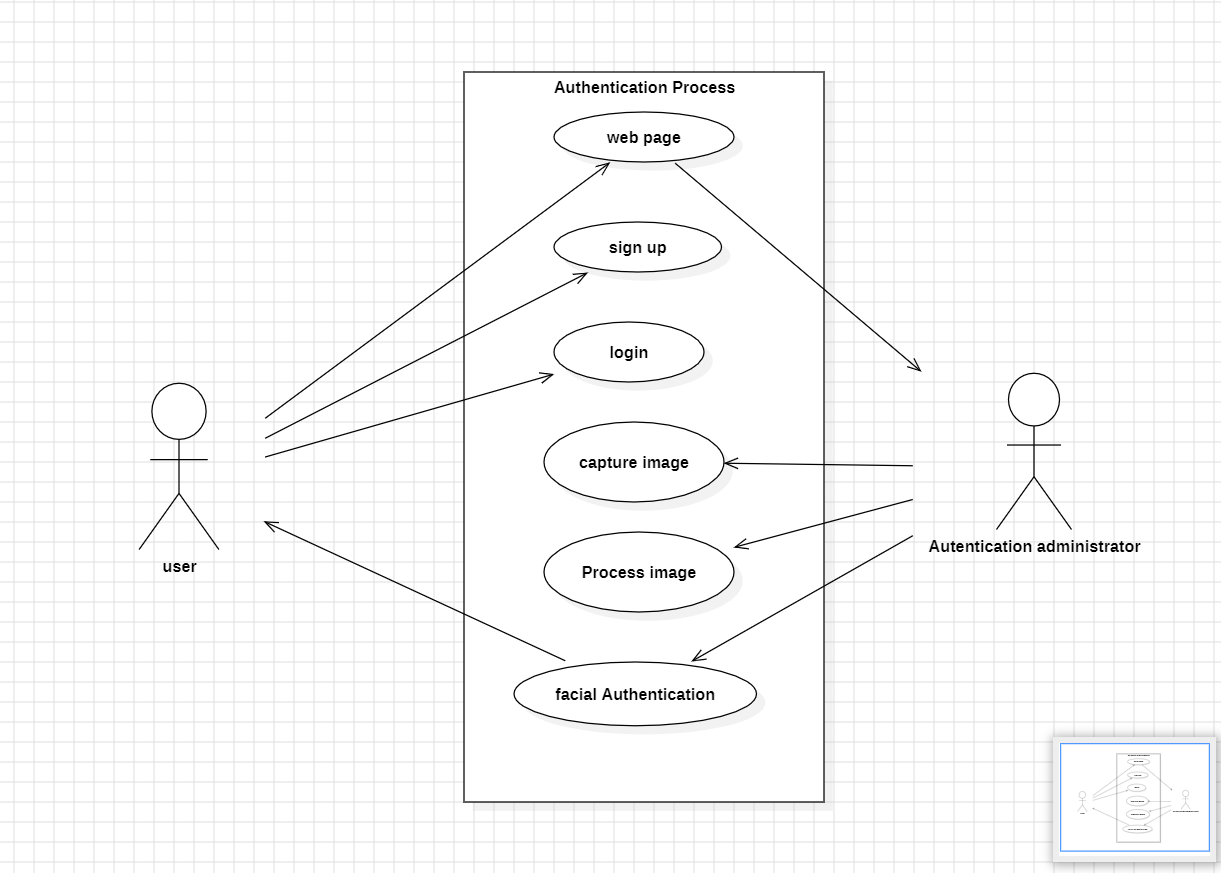
• Documenting the artifacts of a software-intensive system.

The UML is a language that provides vocabulary and the rules for combining words in that vocabulary for the purpose of communication. A modeling language is a language whose vocabulary and rules focus on the conceptual and physical representation of a system. Modelingyields an understanding of a system.

There are two broad categories of diagrams and they are again divided into structural diagrams and behavioral diagrams. The structural diagrams represent the static aspect of the system. The four structural diagrams are class diagram, object diagram, component diagram, deployment diagram. Behavioral diagrams basically capture the dynamic aspect of a system. Types of behavioral diagrams are use case diagrams, sequence diagrams, collaboration diagrams, state chart diagrams, and activity diagrams. These UML diagrams help define the system's functionalities, structure, behavior, interactions, architecture, and deployment aspects, providing a comprehensive visual representation of the web-based facial authentication system's design. Some of the frequently used use case diagrams in software development are as below

**A. USE-CASE DIAGRAM :**

A use-case is defined as a set ofsequences of actions performed by an actor to achieve a specificresult. A use case contains an actor. An actor refers to various people that use the system.

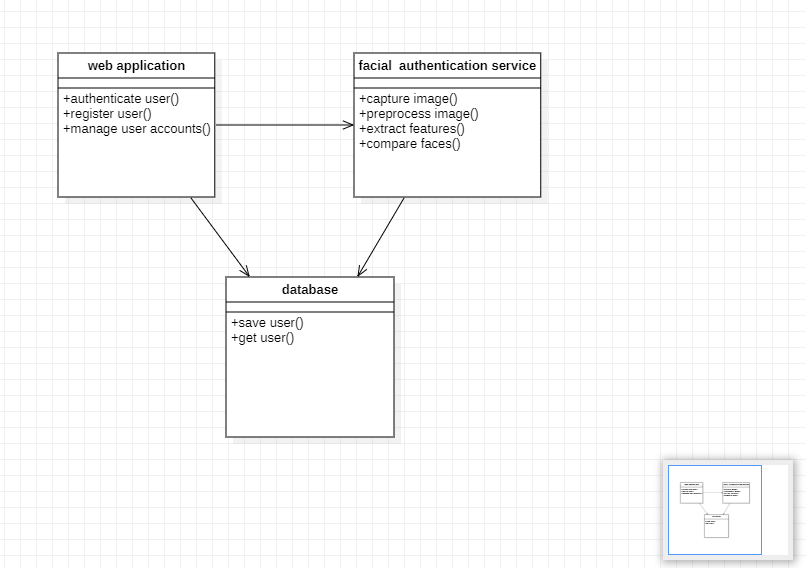
**Fig-4: Use case diagram**

The above use case diagram shows that :

1. The process starts with a user on a web page.
2. The user can then choose between signing up for an account or logging in to an existing account.
3. If the user opts to sign up, they will need to enter their email address and create a password.
4. If the user chooses to login, they will need to enter their email address and password.
5. Once the user submits their information, the system will process it and authenticate the user.
6. After successful authentication, the user will be granted access.

**B. CLASS DIAGRAM :**

In software engineering, a class diagram in UML is a type of static structure diagram that describes the structure of a system by showing the system's classes, attributes, operations (or methods), and relationships among objects. It represents the functionality ofvarious requirements in an application.

**Fig-5: Class diagram**

The above class diagram sows that the web application has three functionalities listed:

1. Authenticate user: This likely refers to a process where a user submits their login credentials.
2. Register user: This signifies a process where a new user creates an account with the web application.
3. Manage user accounts: This suggests managing existing user accounts, possibly including activities like updating passwords or deleting accounts.

The facial authentication service has four functionalities listed:

1. Capture image: This likely refers to capturing a facial image via a webcam or camera.
2. Preprocess image: This signifies getting the image ready for facial recognition, possibly including resizing or adjusting lighting.
3. Extract features: This refers to extracting identifying data from the facial image, like distances between key points on the face.
4. Compare faces: This signifies comparing the extracted facial data to a database of faces, likely to identify the person in the captured image.

Text labels at the bottom center depict a database. Arrows show how the facial recognition service interacts with the database:

1. Save user: This signifies storing facial data in the database, likely associated with a registered user.
2. Get user: This signifies retrieving facial data from the database, possibly to compare it to a captured face image.

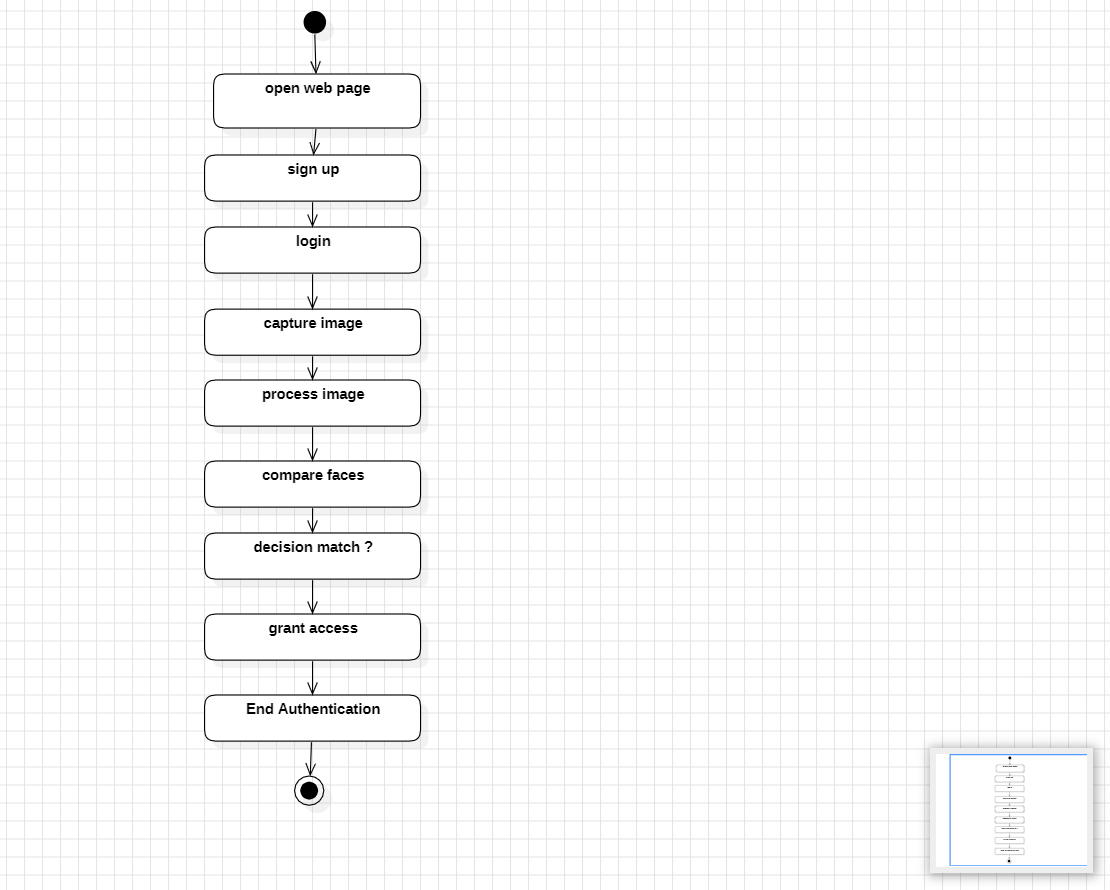
**C. ACTIVITY DIAGRAM :**

An activity diagram is like a flow Machine showing the flow control from one activity to another. In UML it is used to describe the dynamic aspects ofthe system. Activity diagrams areused to represent the sequence ofactivities performed by actors**.**

The activity diagram depicts a web application login process with facial recognition. Here's a simplified overview:

1. A user opens the web application.
2. They choose to either sign up for a new account or log in to an existing one.
3. For signup, the user likely captures a facial image, possibly with a webcam. This image is stored along with their account information.
4. For login, the system captures a facial image of the user attempting to access the account.
5. This captured image is then compared to a database containing facial data from registered users.
6. If the facial features match someone in the database, access is granted.
7. If there's no match, access is denied.

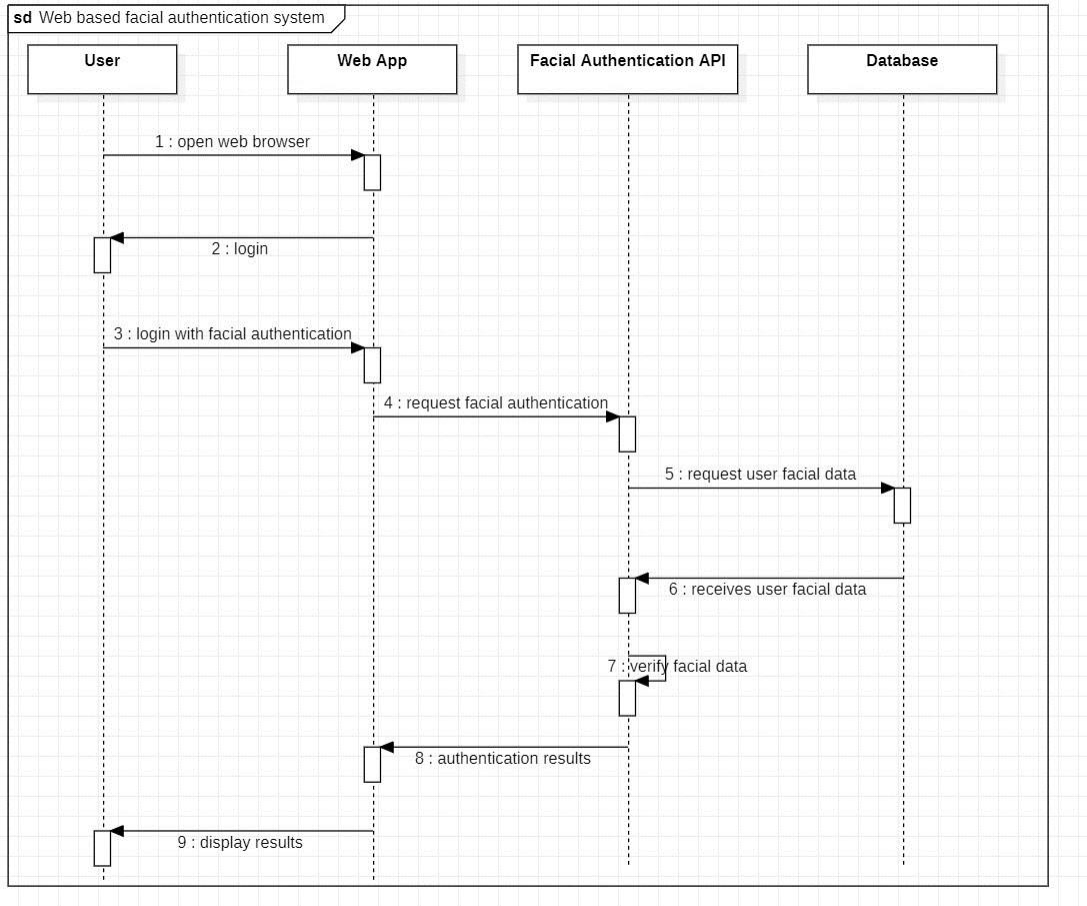
Activity diagram for above discription:



**Fig-6: Activity Diagram**

**D. SEQUENCE DIAGRAM :**

The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communicationbetween any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time.



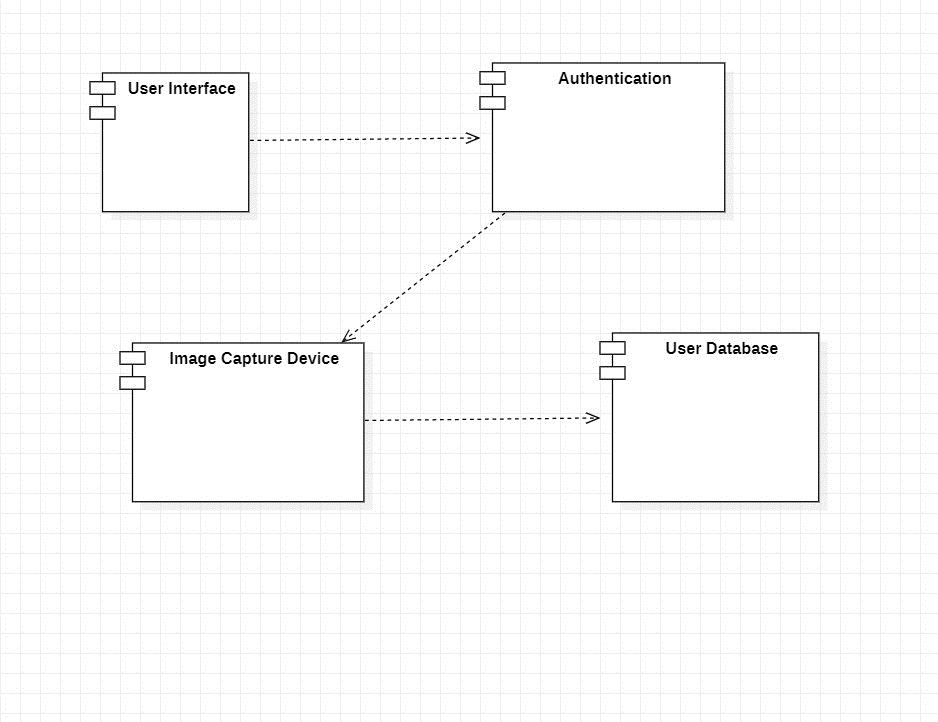
**Fig-7: Sequence Diagram**

The sequence diagram of a web-based facial authentication system. It includes the process :

1. The process starts with a user opening a web browser (1).
2. The user then logs in to the web application, potentially using facial authentication (2, 3).
3. After a successful login, the web application requests facial authentication from the facial authentication API (4).
4. The facial authentication API then requests the user's facial data from the database (5).
5. The database retrieves the user's facial data and sends it back to the facial authentication API (6).
6. The facial authentication API verifies the user's facial data (7).
7. Once the verification is complete, the API sends the authentication results back to the web application (8).
8. Finally, the web application displays the results to the user (9).

**E. COMPONENT DIAGRAM :**

A component diagram is used to break down a large object-oriented system into smaller components,so asto make themmore manageable. Itmodels the physical view ofa system suchas executables, files, libraries, etc. that resides within the node. It visualizes the relationships as well as the organizationbetweenthe components present inthesystem. It helps in forming an executable system.



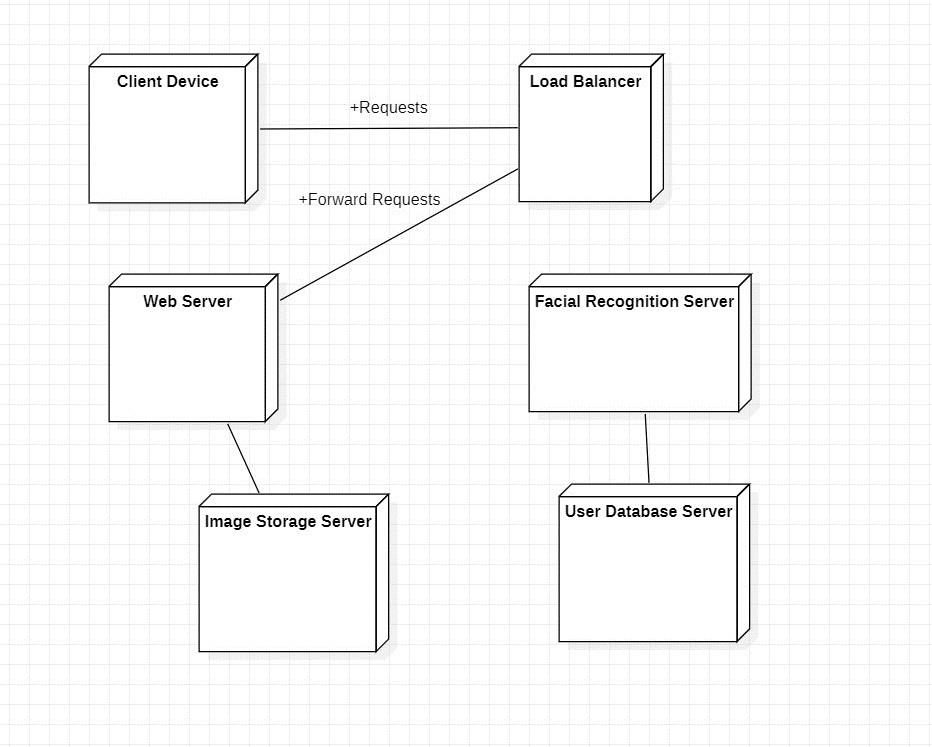
**Fig-8 : Component Diagram**

The component diagram includes :

1. User Interface
2. Authentication
3. Image capture device
4. User database

**F. DEPLOYMENT DIAGRAM :**

The deployment diagram represents the deployment view ofa system. It is related to thecomponent diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists ofnodes. Nodes are nothing but physical hardware used to deploythe application.



**Fig-9 : Deployment diagram**

The deployment diagram of a web application with facial recognition. Here's a breakdown of the system:

* **Client Device:** This represents the user's device, such as a laptop or smartphone, used to access the web application.
* **Load Balancer:** This distributes incoming traffic among multiple web servers, ensuring efficient use of resources and high availability for the web application.
* **Web Server:** This runs the web application code and delivers content to the client device. There can be multiple web servers for redundancy and scalability.
* **Facial Recognition Server:** This server performs facial recognition tasks. It likely receives captured facial images from the web server, compares them to a database of facial data, and returns verification results.
* **User Database Server:** This stores user information, likely including account credentials and potentially facial data for registered users.
* **Image Storage Server:** This server stores facial images captured by the system.

**4.5.DB DESIGN**

The database design for the web-based facial authentication system focuses on efficiently managing user data, including facial recognition information. The key components of the database design are as follows

1. **User Table :** 
   * user\_id (Primary Key) : A unique identifier for each user.
   * Username : The user's chosen username.
   * Password : The user's password (hashed for security).
   * Email : The user's email address.
2. **Facial Data Table :**
   * facial\_data\_id (Primary Key) : A unique identifier for each facial data record.
   * user\_id (Foreign Key) : Links to the user's unique identifier.
   * facial\_features : Stores the extracted facial features used for authentication.
   * image\_path : The file path of the user's facial image.
3. **Authentication Log Table :**
   * log\_id (Primary Key) : A unique identifier for each authentication attempt.
   * user\_id (Foreign Key) : Links to the user's unique identifier.
   * Timestamp : The date and time of the authentication attempt.
   * Status : The result of the authentication attempt (e.g., success, failure).

This database design ensures that user data is securely stored and easily accessible for authentication purposes, while maintaining the integrity and confidentiality of sensitive information.

**5. SYSTEM IMPLEMENTATION**

**5.1 RUNNING APPLICATION**

To run the application, ensure that all dependencies are installed and the development environment is properly configured. Use the following steps:

1. Install Dependencies: Use npm install to install all necessary packages.
2. Start the Development Server: Run npm start to launch the development server.
3. Access the Application: Open a web browser and navigate to http://localhost:3000 to access the web-based facial authentication system.

**5.2 CONFIGURING DATABASE**

Configuring the database involves setting up the database schema and connecting the application to the database. Follow these steps:

1. Database Setup: Create the required tables as per the database design, including User, Facial Data, and Authentication Log tables.
2. Configuration File: Update the configuration file (config.js or equivalent) with the database connection details (host, port, username, password, and database name).
3. Database Connection: Ensure the application can connect to the database using the configured settings. Verify the connection before proceeding with other development tasks.

**5.3 CODING :**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<link rel="icon" type="image/svg+xml" href="/vite.svg" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<style>body

{

background: linear-gradient(rgba(0,0,0,0.4),rgba(0,0,0,0.8))url(1.jpg);

}

</style>

<title>Face Authentication</title>

</head>

<body>

<div id="root"></div>

<script type="module" src="/src/main.jsx"></script>

<!-- fio.js recognition script -->

<div id="faceio-modal"></div>

<script src="https://cdn.faceio.net/fio.js"></script>

</body>

</html>

import React from 'react'

import ReactDOM from 'react-dom/client'import App

from './App'

import './index.css'16

ReactDOM.createRoot(document.getElementById('root')).render(

<React.StrictMode>

<App />

</React.StrictMode>

)

import { useState } from 'react';

import { useEffect } from 'react'

import './App.css'

function App()

{ let faceio;

useEffect(() => {

faceio = new faceIO("fioaca25");

}, []);

const handleSignIn = async () => {

try

{

let response = await faceio.enroll({

locale: "auto", payload: {

email: "example@gmail.com",

pin: "12345",

},

});

console.log(` Unique Facial ID: ${response.facialId}Enrollment

Date: ${response.timestamp}

Gender: ${response.details.gender}

Age Approximation: ${response.details.age}`);

} catch (error)

{ console.log(error);

}};17

const handleLogIn = async () => {

try {

let response = await faceio.authenticate

({locale: "auto",

});

console.log(` Unique Facial ID: ${response.facialId}

PayLoad: ${response.payload});

} catch (error) { console.log(error);

}

};

return (

<div className="App">

<section>

<h1>Web Based Facial Authentication System</h1>

<button on Click={handleSignIn}>Sign-Up</button>

<button on Click={handleLogIn}>Log-in</button>

</section>

</div>)

}

import { defineConfig } from 'vite'

import react from '@vitejs/plugin-react'

// https://vitejs.dev/config/

export default defineConfig({

plugins: [react()],

base: "/WebFacialAuth/"

})

**6. SYSTEM TESTING**

**6.1 TESTING INTRODUCTION**

**Testing** is the **systematic process of evaluating a system or component with the intent to identify any inconsistencies between the actual and the expected results.** It aims to expose defects, errors, or areas where the system does not meet its intended functionality, security, or usability requirements.

Web-based facial authentication systems offer a convenient and potentially secure login method. However, ensuring their reliability and security requires thorough testing.

**6.2 UNIT TESTING**

Unit testing involves testing individual components and modules, such as the facial image capture, feature extraction, and comparison, using mock data and scenarios. This helps identify and fix issues at an early stage in development.

**6.3 WHITE BOX TESTING**

White box testing, also known as clear or glass box testing, involves testing the internal structures or workings of an application. The tester needs to have knowledge of the internal logic of the system and code.

**6.4 BLACK BOX TESTING**

Black box testing focuses on testing the functionality of the application without knowledge of the internal code structure. This ensures that the system meets user requirements and functions as expected.

**6.5 INTEGRATION TESTING**

Integration testing tests the integration of different components and modules to ensure they work seamlessly together. This helps verify that the system's parts are correctly integrated and communicate properly.

**6.6 TEST CASES**

Example Test Cases:

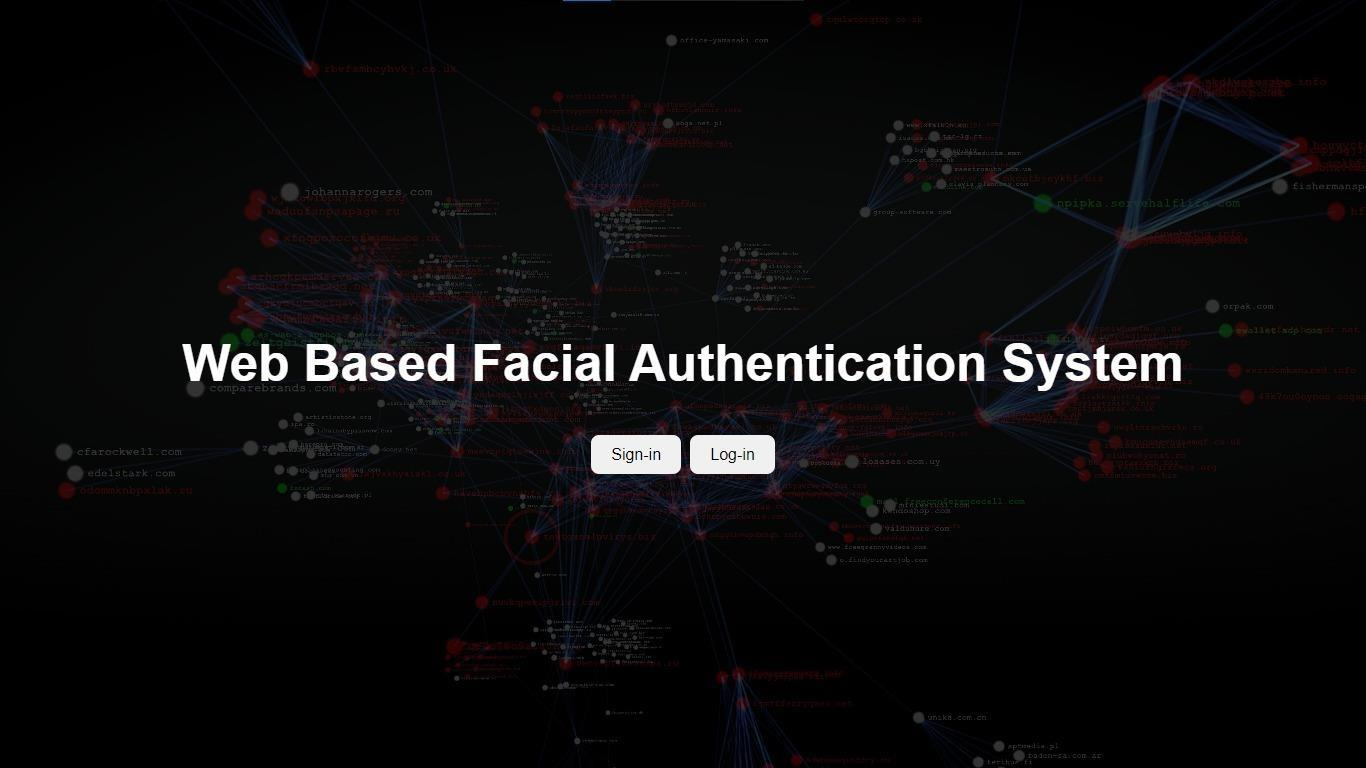
1. **Facial Image Capture:**
   * Verify that the system can capture a facial image from the user's webcam.
   * Ensure the image quality is sufficient for further processing.
2. **Feature Extraction:**
   * Check that the system accurately extracts facial features from the captured image.
   * Validate that the extracted features are correctly formatted and stored.
3. **Authentication:**
   * Test the system's ability to authenticate a user based on the captured facial features.
   * Confirm that the authentication process is quick and accurate.
4. **Integration:**
   * Ensure that all modules (image capture, feature extraction, and authentication) work together seamlessly.
   * Test the entire authentication flow from image capture to successful login.

These tests ensure the web-based facial authentication system is reliable, secure, and user-friendly.

Top of Form

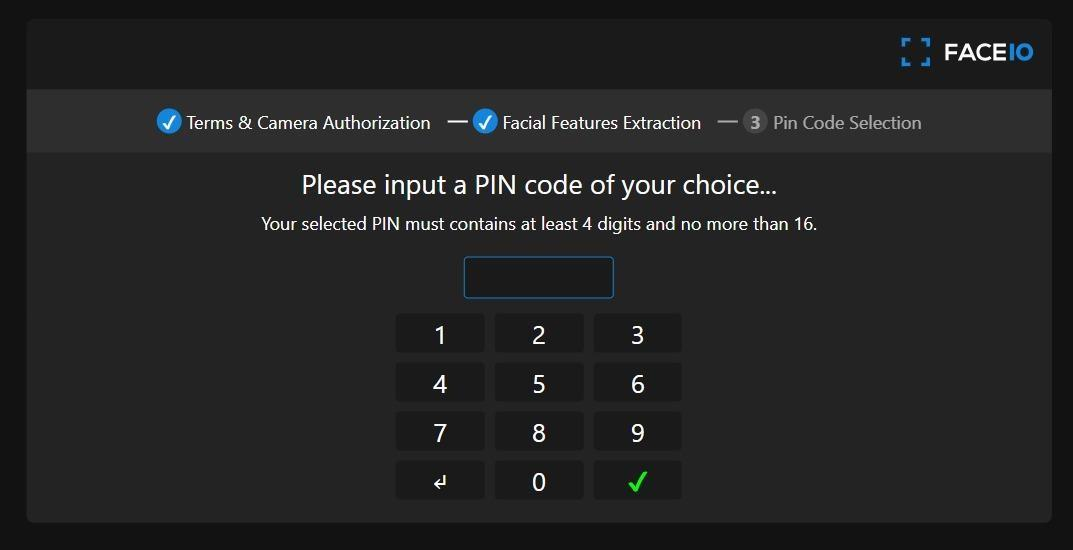
**7. OUTPUT SCREENS**

**1. Sign in /Login page:**



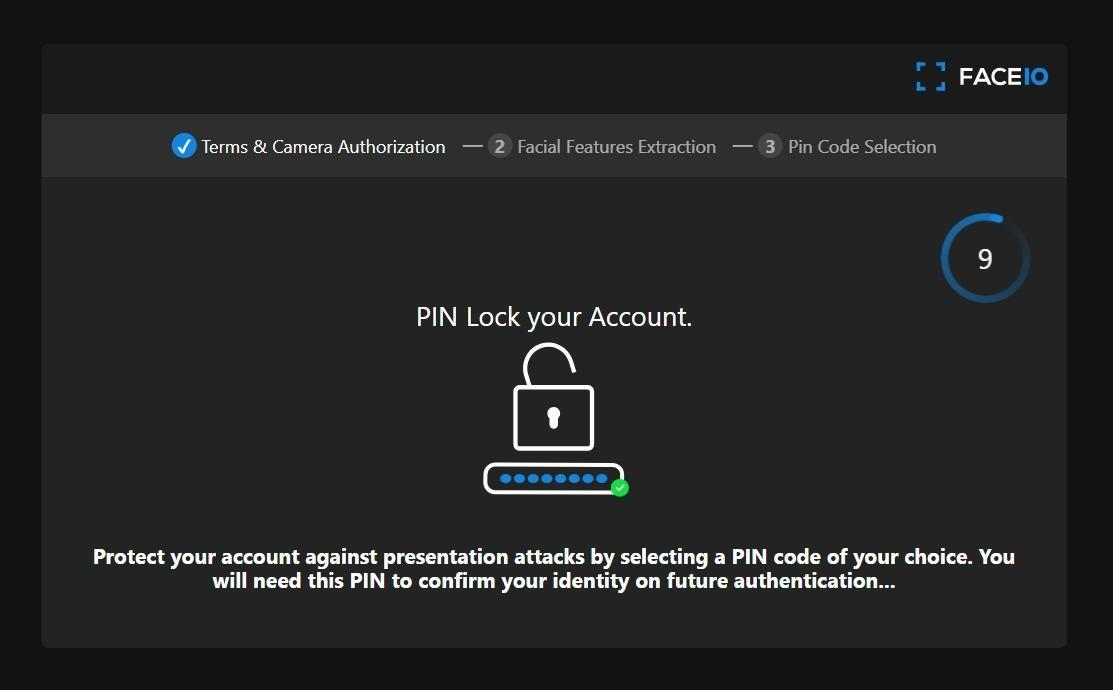
**FIG-10: Sign in /Login page**

**2. Passcode/Pin Page:**



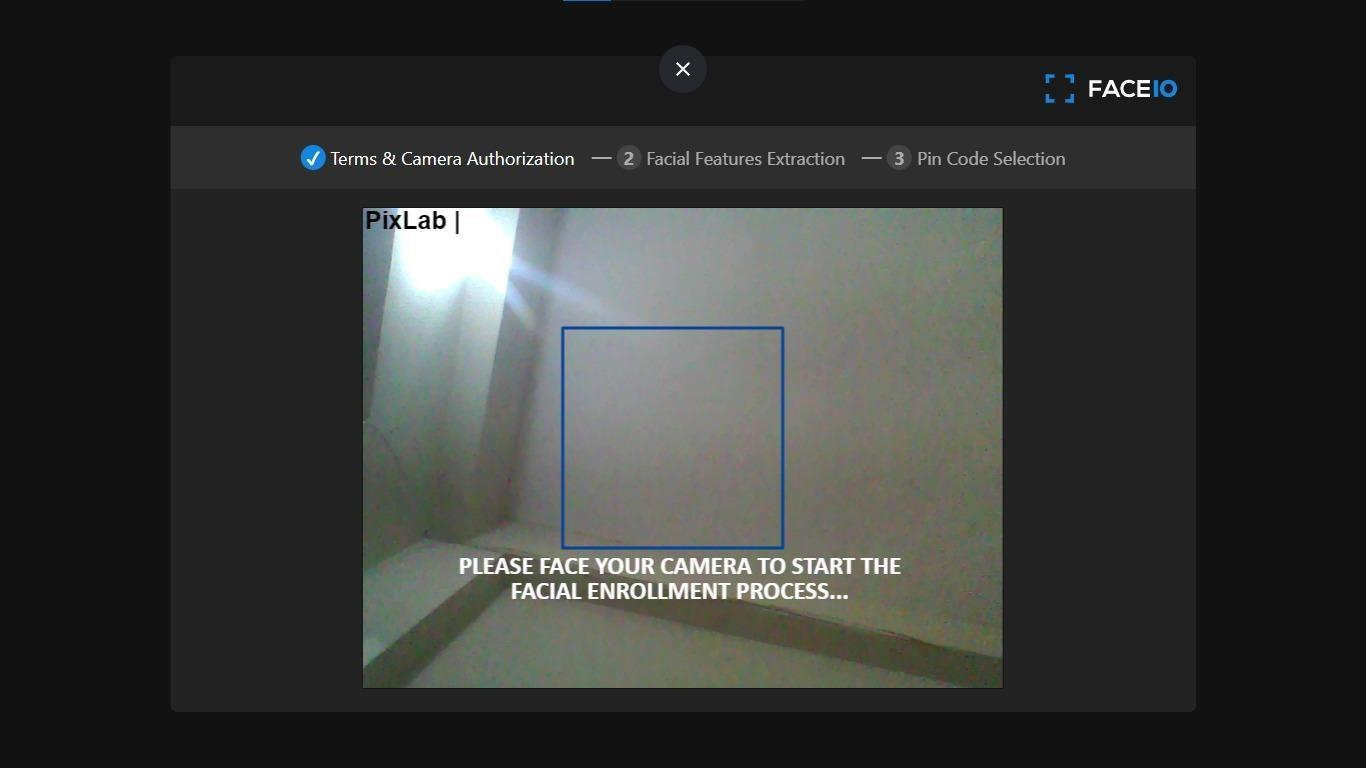
**FIG-11:Passcode/Pin Page**

**3. Signed up Page:**



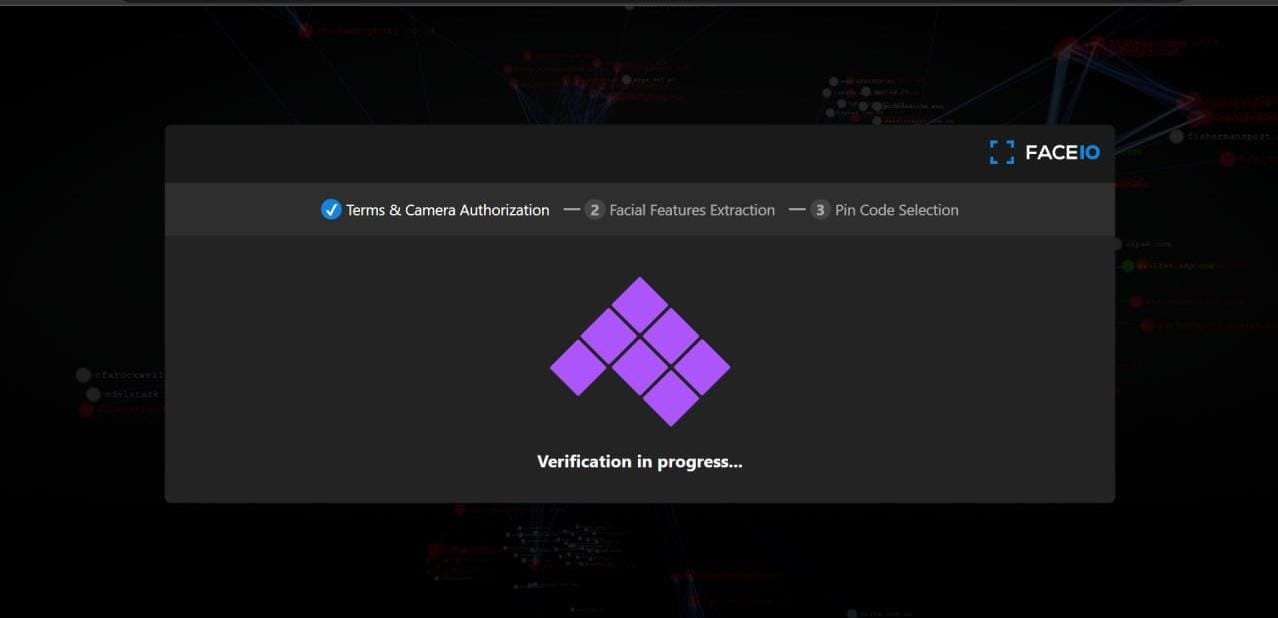
**FIG-12: Signed up Page**

**4. Camera Authorizing Page:**



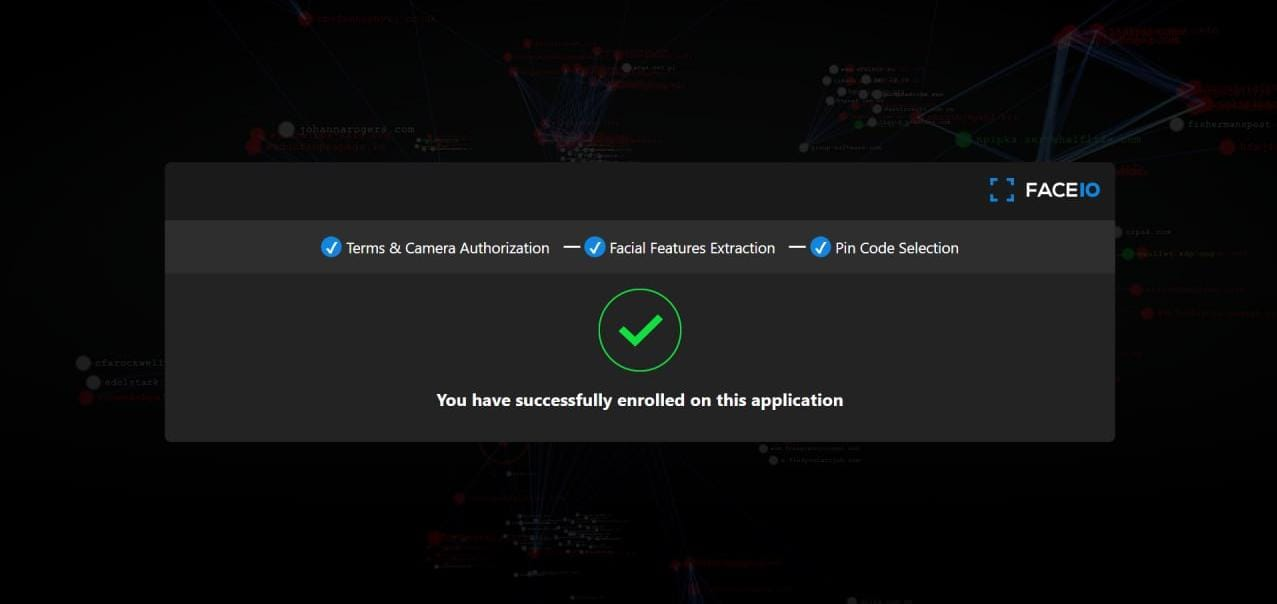
**FIG-13: Camera Authorizing Page**

**5. Verification in Progress :**



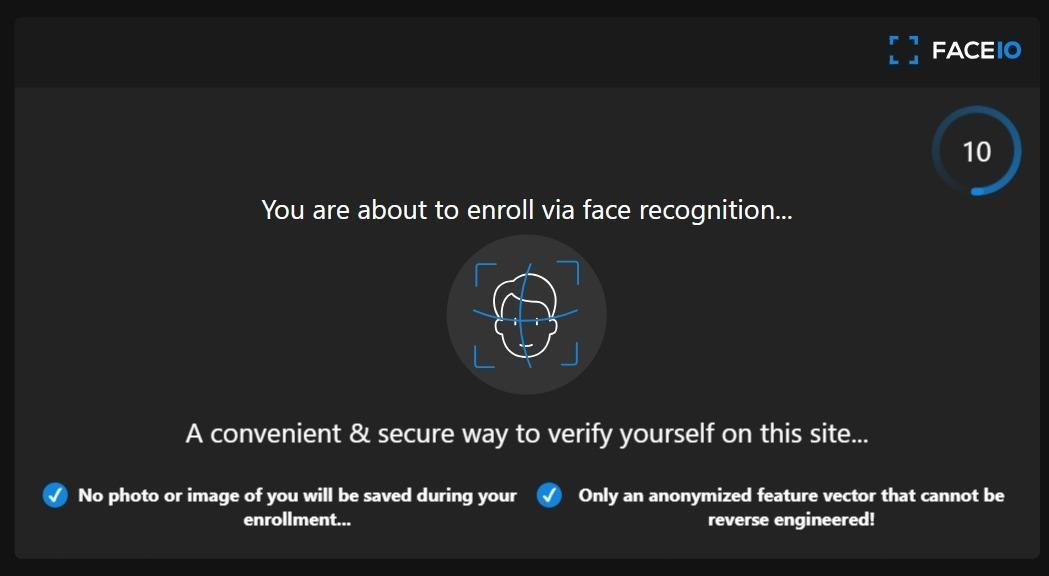
**FIG-14:Verification in Progress**

**6. Successfully Enrolled :**



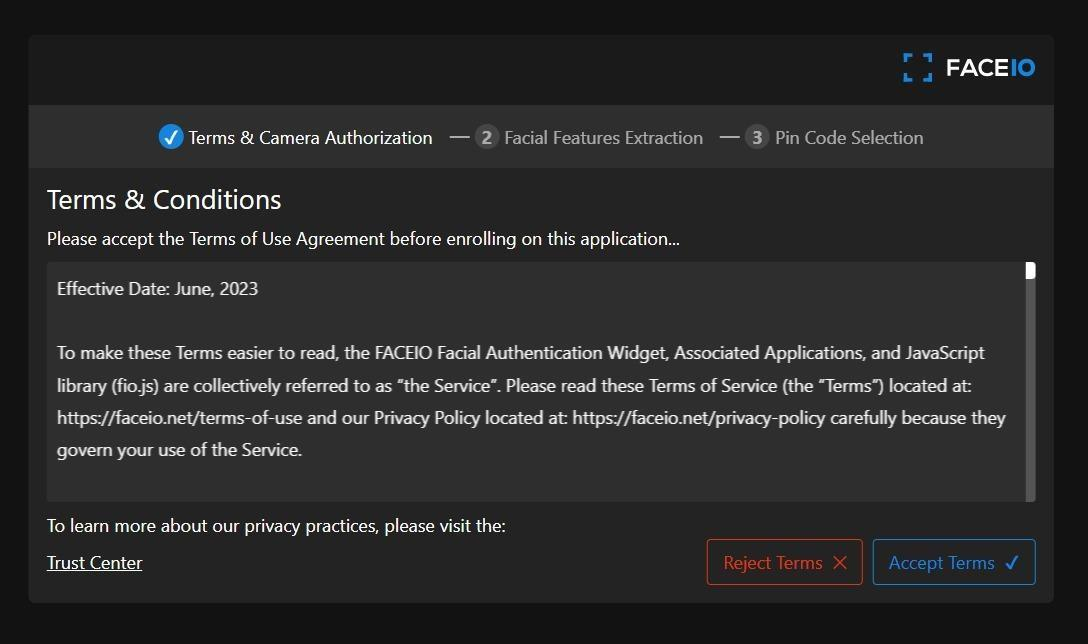
**FIG-15: Successfully Enrolled**

**7.Enrolling via face recognition Page:**



**FIG-16: Enrolling via face recognition Page**

**8.Terms and Conditions Page:**



**FIG-17: Terms and Conditions Page**

**8. CONCLUSION**

The developed web-based facial authentication system represents a significant advancement in user authentication for web applications. By utilizing advanced facial recognition technology, it offers a secure, convenient, and innovative method for verifying user identities. This system addresses the common vulnerabilities associated with traditional password-based systems, such as brute force attacks and password management issues, by providing a robust alternative.

The system architecture ensures that all components work together harmoniously, enhancing both security and user experience. The database design supports efficient data management, ensuring the integrity and security of sensitive user information. Additionally, the clear data flow within the system guarantees reliable and transparent authentication processes.

This web-based facial authentication system not only improves security and user experience but also sets the stage for future enhancements. These may include integrating multi-factor authentication, refining machine learning algorithms for better accuracy, and expanding compatibility with mobile devices. The system's design and functionality make it a promising solution for modern web applications, providing a reliable and user-friendly authentication method.

**9. FUTURE ENHANCEMENT**

The development of the web-based facial authentication system marks a significant improvement in user authentication for web applications. However, there are several potential enhancements that could further improve the system. For now we did only this for only single web page further it can be implemented with the App with facial authentication.

AI and deep learning for accuracy, cross-platform compatibility. Blockchain can decentralize facial data storage, while AR can offer real-time feedback during authentication.

Regular updates can ensure compliance with data protection regulations. Enhanced AI can reduce false positives and negatives, and scalability can support more users.

Advanced encryption can protect data, while user privacy controls, real-time monitoring, and integration with third-party applications can expand utility across industries.

**10.REFERENCES**

Code for Face net model is based on the assignment from Convolutional Neural Networks Specialization by Deeplearning.ai on Coursera.

• <https://www.coursera.org/learn/convolutional-neural-networks/home/welcome>

• Florian Schroff, Dmitry Kalenichenko, James Philbin (2015).Face Net: A Unified Embedding for Face Recognition and Clustering

• Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, Lior Wolf (2014). Deep Face: Closing the gap to human-level performance in face verification

• The pretrained model used is inspired by Victor Sy Wang's implementation and was loaded using his code: https://github.com/ Code iwantooxxoox/Keras-OpenFace.

• A lot of inspiration from the official FaceNet GitHub repository:https://github.com/davidsandberg/facenet