Web Based Facial Authentication for Login

Mr. G. Srinivasa Rao[1], Chevvakula Siri Lalitha[2], Adisharlapalli Sohan Prasad[3], Bairi Vamshika[4], Bolla Venkatesh[5]

[1,2,3,4,5]*Department of Artificial Intelligence Vidya Jyothi Institute of Technology Hyderabad, India*

***Abstract*—Traditional password-based authentication systems are increasingly vulnerable to phishing attacks, credential theft, and usability limitations, making them unsuitable for modern secure applications. As an alternative, biometric authentication—particularly facial recognition—has gained significant attention due to its convenience and higher resistance to credential-based attacks. However, existing facial authentication systems still face critical challenges, including susceptibility to presentation attacks (spoofing), reduced accuracy in distinguishing highly similar faces, and scalability limitations in large-scale identity matching. This work presents a secure web-based facial authentication system designed as a fully password less identity verification framework. The system extracts high-dimensional facial feature representations using deep learning-based embedding techniques that map facial images into a discriminative vector space for reliable identity comparison. To ensure scalability and fast identity retrieval, the system performs efficient similarity-based matching in a high-dimensional feature space, enabling rapid verification even in large user databases. To address security vulnerabilities, a dynamic active liveness detection mechanism is incorporated, which uses real-time behavioral and motion-based challenges to confirm the physical presence of a live user and effectively mitigate spoofing attempts using static images or pre-recorded videos. The proposed system further enhances robustness against highly similar facial structures by applying strict similarity thresholds for identity verification. A practical implementation is demonstrated through integration with a web application, replacing traditional login mechanisms with a seamless biometric authentication flow.**

***Index Terms*—Facial Recognition, Biometric Authentication, Deep Learning, Face Embeddings, Liveness Detection, Anti-Spoofing, Password less Authentication, Identity Verification, Feature Space Matching, Web Security.**

I. 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 expansion1.

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

Maintaining Visual Quality

- 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.
- Compliance and Standards: Adheres to industry standards and regulations for data protection and privacy.

1.5. KEY FUNCTIONALITIES OF THE SYSTEM:

- The system performs real-time face authentication using a webcam, allowing users to unlock or access the system without passwords or tokens.
- It enables face-based user registration and management, where multiple users can enroll their faces and be recognized during authentication.
- The application provides secure local storage of facial data, ensuring user information is protected and not dependent on external systems.
- It delivers fast and efficient performance through local processing and supports real-time interaction using APIs and live event streaming.

II. LITERATURE SURVEY

1. Title: A Robust and Scalable Approach to Face Identification.

Authors: William Robson Schwartz, Huimin Guo, Larry'S. Davis.

Year:2012

Abstract: This research focuses on developing a scalable and robust face identification system capable of handling large datasets. It introduces efficient feature extraction and matching techniques that improve system performance. The approach addresses challenges related to scalability and computational efficiency in real-time face authentication systems.

2. Title: Deep Face Recognition

Authors: Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman

Year: 2015

Abstract: This paper explores deep convolutional neural networks for large-scale face recognition tasks. It demonstrates how deep learning models can effectively learn discriminative facial features, leading to improved accuracy in face authentication. The work emphasizes the importance of large datasets and training strategies in achieving high-performance systems.

3. Title: Handbook of Face Recognition

Authors: Stan Z. Li, Anil K. Jain

Year:2011

Abstract: This handbook provides an extensive overview of face recognition technologies, including algorithms, system design, and real-world applications. It covers both classical and modern approaches, offering valuable insights into building efficient and reliable face authentication systems.

4. Title: An Introduction to Biometric Recognition

Authors: Anil K. Jain, Arun Ross, Salil Prabhakar

Year:2004

Abstract: This paper provides a comprehensive overview of biometric recognition systems, including face recognition. It discusses system components, performance metrics, and security considerations. The

study highlights the advantages and limitations of biometric-based authentication, offering insights relevant to the design of secure face authentication systems.

5. Title: Presentation Attack Detection Methods for Face Recognition Systems

Authors: Raghavendra Ramachandra, ChristophBusch
Year:2017

Abstract: This paper reviews different methods for detecting presentation attacks in face recognition systems. It evaluates techniques such as texture analysis and motion-based detection to prevent unauthorized access. The work emphasizes enhancing the reliability and security of face authentication systems.

III. PROPOSED SYSTEM

3.1. SYSTEM OVERVIEW

- The proposed system is a real-time face authentication application that verifies user identity using facial features captured through a webcam.
- It operates in a fully local environment, ensuring faster response time and improved privacy.
- The system integrates backend services, computer vision techniques, and a web interface into a unified platform.
- It is designed to be modular and scalable, allowing easy extension and integration with other applications.

3.2. FACE AUTHENTICATION PROCESS

- The system captures live video input and performs real-time face detection using computer vision techniques.
- Detected faces are processed and compared with stored facial encodings for authentication.
- Upon successful matching, the system grants access automatically without requiring passwords.
- The process is optimized for low latency and quick response, ensuring smooth user experience.

3.3. USER REGISTRATION AND MANAGEMENT

- New users can register their face data through the webcam interface.
- The system supports multi-user management, allowing addition, deletion, and tracking of users.
- Each user's facial data is uniquely stored and associated with a specific identity.
- A dedicated interface enables easy monitoring and management of registered users.

3.4. ADVANTAGES OF THE PROPOSED SYSTEM

- Eliminates the need for password-based authentication, improving user convenience.
- Ensures high privacy by keeping all facial data within the local system.
- Provides faster performance due to reduced network dependency and real-time processing.
- Offers a complete authentication solution with user management and activity tracking.
- Enhances security through encrypted data storage and controlled access.
- Supports easy integration with other systems using APIs and real-time communication.
- Designed with a modular structure, making it adaptable for future improvements. allows the model to recover the message even after transformations like resizing or cropping.

IV. 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.

4.2. SDLC MODEL:

An SDLC model (Software Development Life Cycle model) is a formalized framework that defines a structured process for the software development lifecycle.

Fig 1: Software development lifecycle

It outlines the distinct phases involved in creating software, from initial conception and planning to final deployment, testing, and ongoing maintenance.

1. 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.

2. Deployment (Server & Client-side):
- Deploy the application on a web server.
- Ensure client-side compatibility across different browsers and devices.

3. 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.

4.3. 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 apture and preprocessing.

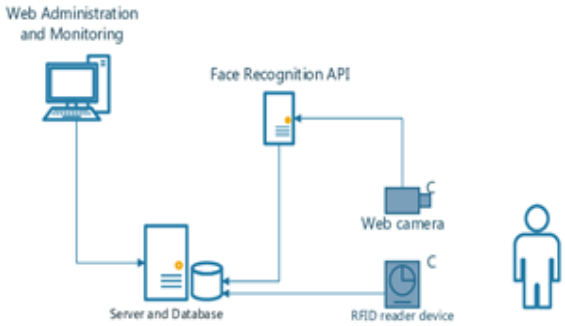


fig-2: System Architecture

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 artifact id 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.

4.4. MODULES DESCRIPTION

In a web-based facial authentication system, there can be several modules that work together
to perform various tasks. Here are some common modules that can be included:

4.4.1. 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.

4.4.2. 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, Face Net, or Convolutional Neural Networks (CNNs) for accurate facial recognition. Utilizes the Face IO API and the fio.js JavaScript library for facial recognition capabilities.

4.4.3. Liveness Detection Module:

Liveness detection is used to ensure that the detected face is from a 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.

4.4.4. Data Storage and Management Module:

This module handles the storage and management official data, including storing the face prints or templates of enrolled individuals, managing databases, and handling data privacy and security.

4.4.5. User Interface Module:

The user interface module provides the interface for users to interact with the facial authentication system. It can include features such as capturing images or videos, displaying authentication results, and providing feedback to the user.

4.4.6. Integration with APIs or SDKs Module:

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

4.5. 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) • Purpose: Segmentation helps determine specific

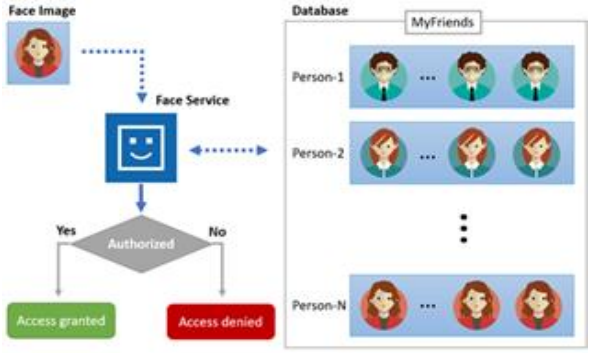


Fig-3: Dataflow model

This diagram illustrates a face recognition system. It begins with a user submitting a face image for identification.

4.6. UML DESIGN

The Unified Modeling Language (UML) is a standard language for writing software blueprints. 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. Modeling yields 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

4.6.1. 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.

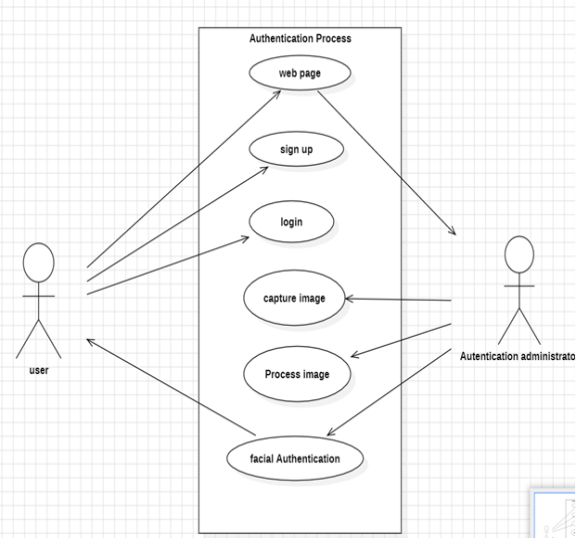


Fig-4: Use case diagram

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.

4.6.2. 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 of various requirements in an application.

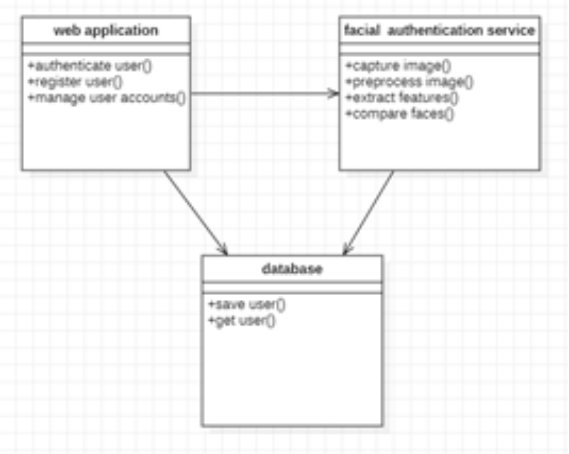


Fig-5: Class diagram

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.

4.6.3. 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 of the system. Activity diagrams are used to represent the sequence of activities performed by actors.

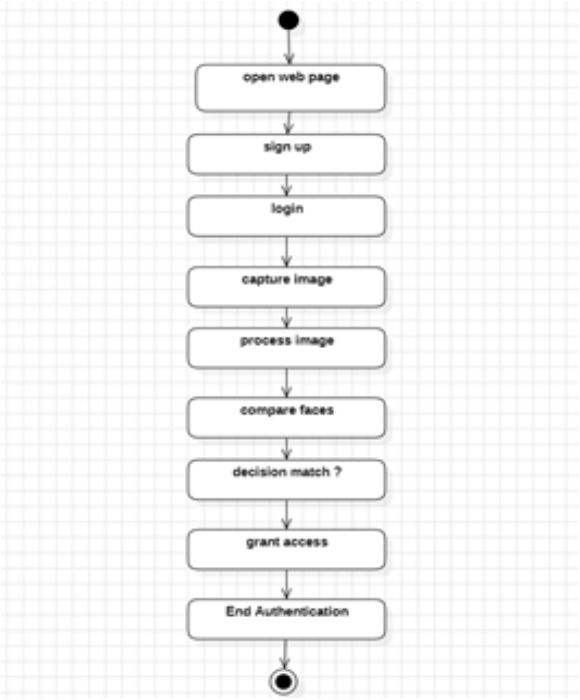


Fig-6: Activity Diagram

4.6.4. 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 communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time.

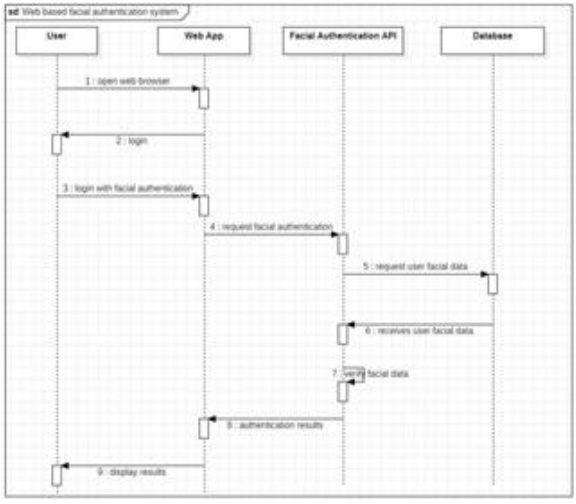


Fig-7: Sequence Diagram

4.6.5. E. COMPONENT DIAGRAM:

A component diagram is used to break down a large object-oriented system into smaller components, so also make them more manageable. It models the

physical view of a system such as executables, files, libraries, etc. that resides within the node. It visualizes the relationships as well as the organization between the components present in the system. It helps in forming an executable system

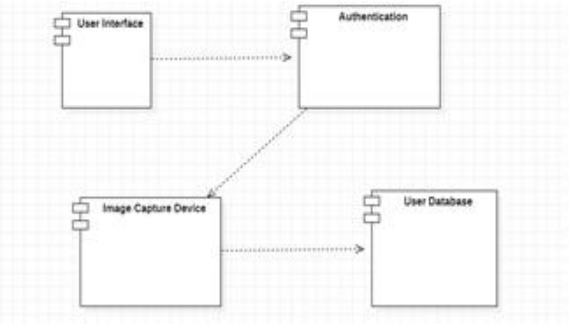


Fig-8: Component Diagram

The component diagram includes:

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

4.6.6. DEPLOYMENT DIAGRAM:

The deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware used to deploy the application.

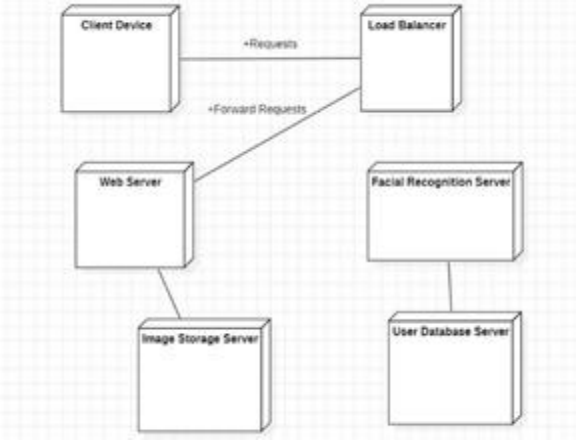


Fig-9: Deployment diagram

4.6.7. OBJECT DIAGRAM:

- The diagram represents the interaction between client, server, and authentication components, where the client sends a base64 image to the server for face authentication.
- The authentication engine processes the image, performs face detection and matching using defined logic, and interacts with encryption and storage modules to securely handle facial data.

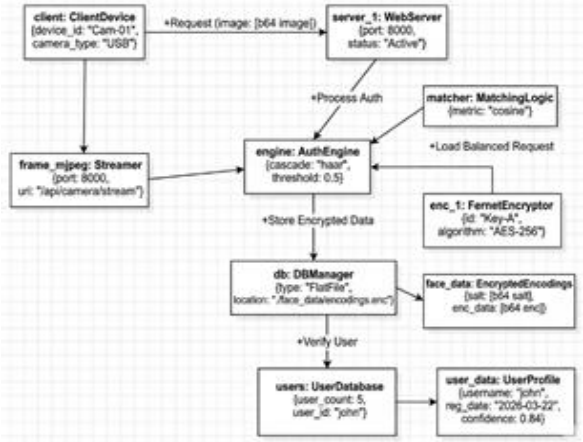


Fig:9 Object Diagram

4.6.8. STATE DIAGRAM

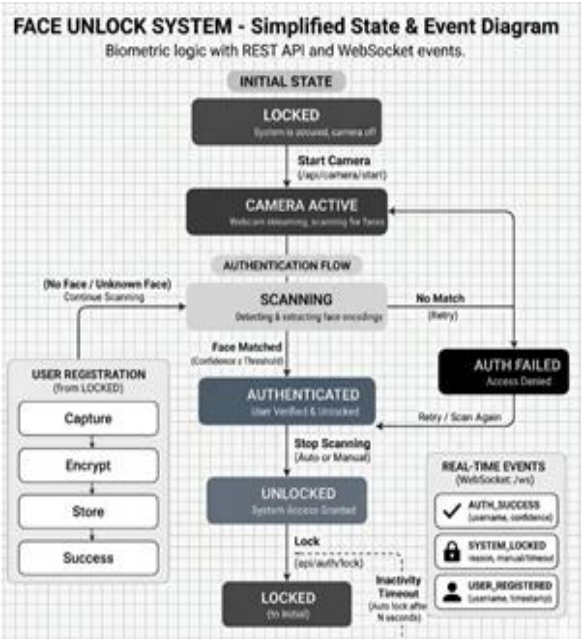


Fig:10 State Diagram

- The diagram illustrates the complete flow of face authentication, starting from the locked state, activating the camera, scanning the face, and transitioning to authenticated or failed states based on matching results.
- It shows the system behavior and transitions, including user registration, real-time authentication, auto-lock after inactivity, and

event handling through APIs and WebSocket communication.

4.6.9. PACKAGE DIAGRAM

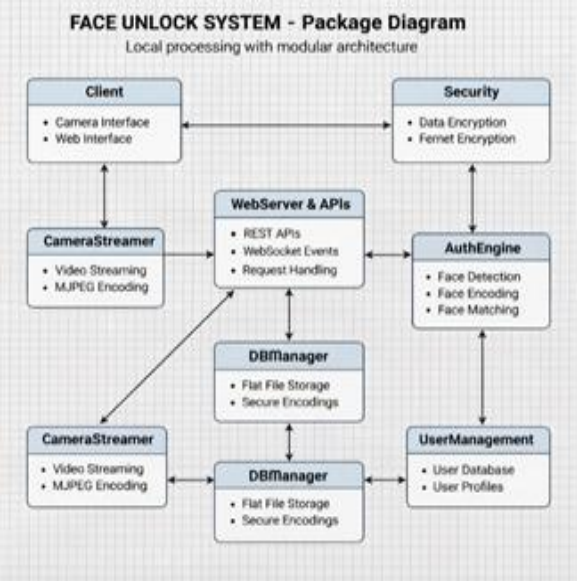
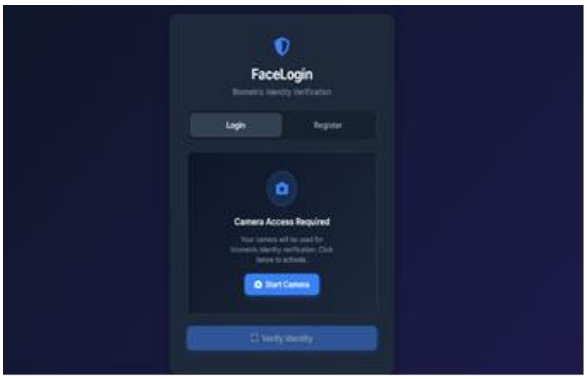


Fig 11: Package Diagram

- The diagram represents the modular architecture of the system, where components like Client, Web Server, Authentication Engine, Security, and Database are organized into separate packages for better structure and maintainability.
- It shows the interaction between different packages, where the Web Server handles requests, the Auth Engine processes face authentication, the Security module manages encryption, and the Database stores user and face data.

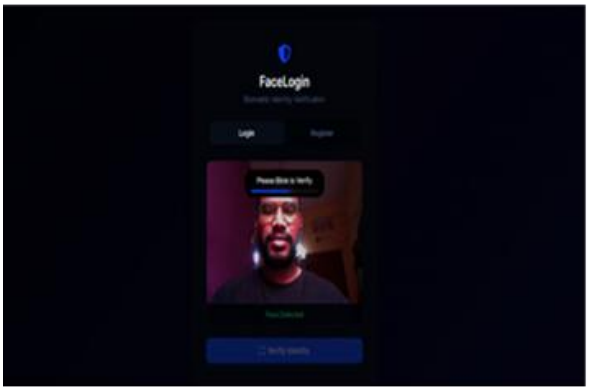
V. RESULTS

5.1. APP INITIAL LOGIN PAGE

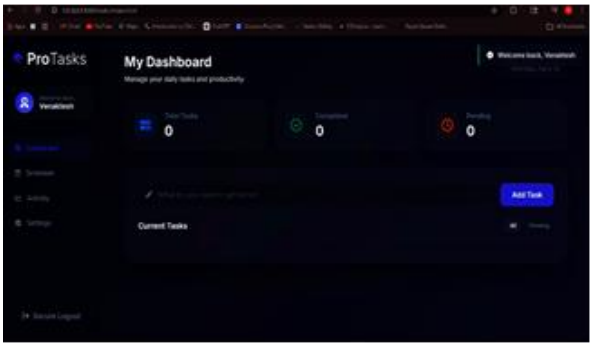


5.2. ENROLL IDENTITY

5.3. LOGIN – VIA - EYES BLINKING GEASTURE FOR TESTING LIVENESS:

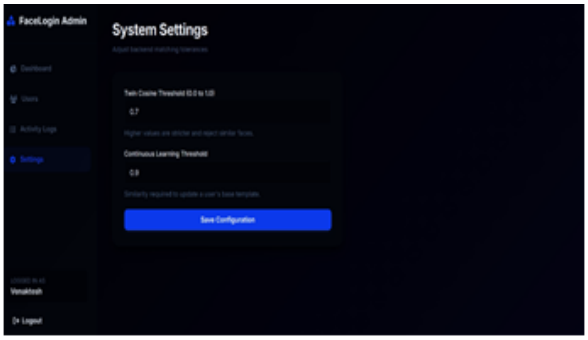


5.4. DASHBOARD:

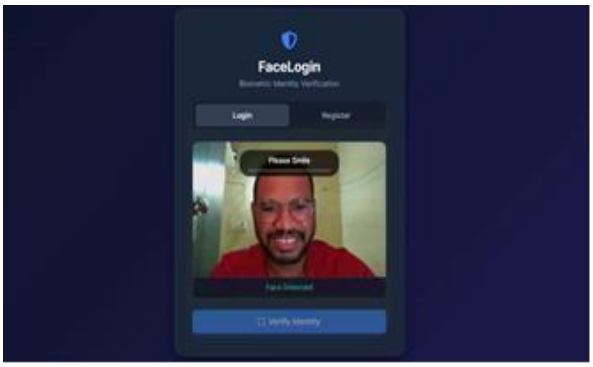


5.5 ACTIVITY LOGS

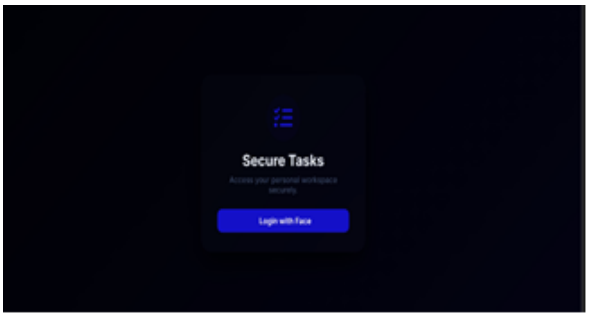
5.6 APP SETTINGS



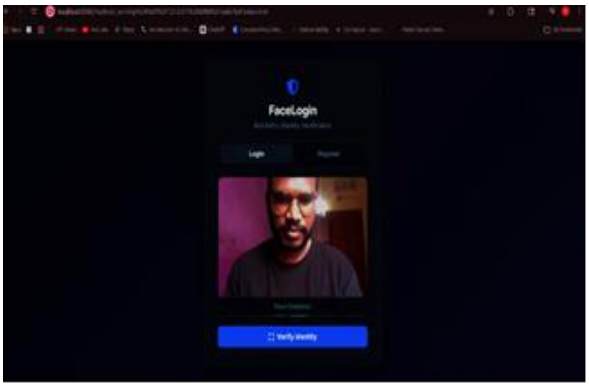
5.7 LOGIN - VIA – SMILE GEASTURE



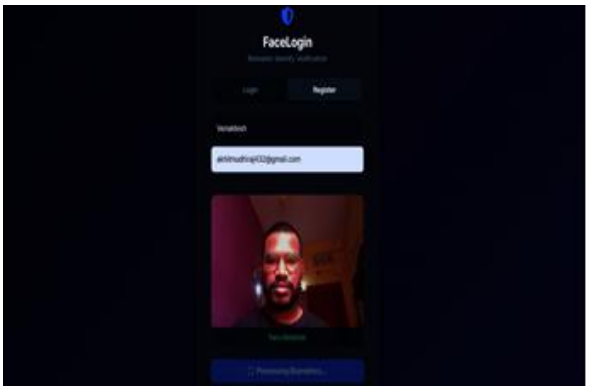
5.8 DETECTING SPOOFING



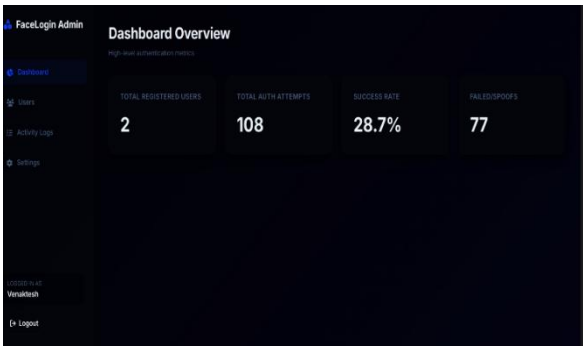
5.9 TODO APP – TEST & CONNECT



5.10. TODO APP- REDIRECT TO FACE AUTH APP FOR AUTHENTICATION



5.11 USER AUTHENTICATED SUCCESSFULLY AND REDIRECTING TO TODO APP DASHBOARD



VI. CONCLUSION

The web-based facial authentication system represents a significant advancement in the field of user authentication by addressing the security and usability limitations associated with traditional password-based methods. As discussed throughout the project, conventional authentication mechanisms rely heavily on user-created passwords, which are often weak, reused across multiple platforms, and vulnerable to various cyberattacks. By leveraging facial recognition as a biometric authentication technique, the proposed system provides a more secure, reliable, and user-friendly alternative for verifying user identities in web applications.

The theoretical analysis and literature survey demonstrate that facial recognition has evolved considerably with advancements in computer vision and deep learning techniques. Modern facial authentication systems are capable of accurately identifying individuals despite variations in lighting

conditions, facial expressions, and pose. The integration of facial recognition into a web-based environment further enhances accessibility and scalability, allowing users to authenticate themselves using standard cameras without the need for specialized hardware. This makes the system suitable for a wide range of real-world applications, including finance, healthcare, education, and enterprise systems. Security considerations form a critical aspect of the proposed system. Facial authentication introduces an additional layer of security that significantly reduces the risks associated with password theft, phishing attacks, and unauthorized access. The inclusion of liveness detection mechanisms strengthens the system's resistance to spoofing attacks, ensuring that authentication is performed only by genuine users. Furthermore, the emphasis on secure storage and handling of facial data highlights the importance of privacy preservation and compliance with data protection standards.

From a usability perspective, the system enhances the overall user experience by simplifying the login process. Users are no longer required to remember complex passwords or undergo frequent password reset procedures. Instead, authentication is achieved through a natural and intuitive interaction, which increases user acceptance and reduces cognitive burden. This balance between security and convenience is a key advantage of facial authentication systems over traditional methods.

REFERENCES

[1] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015.

[2] Y. Taigman, M. Yang, M. A. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2014.

[3] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[4] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, 1997.

[5] W. R. Schwartz, H. Guo, and L. S. Davis, "A robust and scalable approach to face identification," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2012.

[6] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2015.

[7] K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Trans. Circuits Syst. Video Technol.*, 2004.

[8] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*. New York, NY: Springer, 2009.

[9] S. Z. Li and A. K. Jain, *Handbook of Face Recognition*, 2nd ed. New York, NY: Springer, 2011.

[10] Hadid, N. Evans, S. Marcel, and J. Fierrez, "Biometric spoofing and countermeasures," *IEEE Signal Processing Magazine*, 2015.

[11] R. Ramachandra and C. Busch, "Presentation attack detection methods for face recognition systems," *ACM Computing Surveys*, 2017.

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2009.

[13] Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.