A Major Project Final Report on

Face Recognition Based Door Unlock System

Submitted in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Engineering in Electronics and Communication
under Pokhara University

Submitted by:

Adarsha Paudyal , 201101

Shyam Khatri Kshetri , 201109

Arjun Prasad Chaulagain , 201108

Manish Poudel , 201104

Under the supervision of

Ashim Khadka

Date:

24 July, 2025

Department of Electronics and Communications Engineering

NEPAL COLLEGE OF INFORMATION TECHNOLOGY

Balkumari, Lalitpur, Nepal

Acknowledgment

We'd like to express our sincere gratitude to everyone who stood by us while we developed our project, "Face Recognition-Based Door Unlock System."

First, our deepest thanks go to our project guide, Ashim Khadka sir, for giving us this opportunity and for his steady guidance, valuable feedback, and encouragement that kept us moving forward from the very beginning.

We're also truly grateful to our Head of Department, Shivahari Acharya sir, and our principal, Mr. Niranjan Khakurel sir, for giving us the chance to complete our final project under the Department of Electronics and Communication Engineering.

A big thank you goes to our friends and classmates for jumping in to help us test and fix things especially when nothing worked quite like we expected (more times than we'd like to admit).

We also appreciate the open-source community and the developers behind tools like face-api.js, MondoDB, ReactJS, Node.js, and Arduino. Without these resources, building this system within our budget and time frame would have been nearly impossible.

Finally, we're grateful to our families for putting up with the late nights, tangled wires, and endless noise of servo motors and keypads.

This project came to life because of all these people and their support we're genuinely thankful for every bit of it.

Abstract

The Face Recognition-Based Door Unlock System combines modern facial recognition technology with password authentication to provide a more secure and convenient way to control door access. The system uses face-api.js, a JavaScript library for detecting and recognizing faces, integrated into a ReactJS frontend. A Node.js and Express backend handles communication between the web application and the hardware.

When someone approaches the door, a webcam captures their facial image. The system checks this image against a database of registered users stored in MongoDB. If the face matches, the user must also enter a valid password using a connected keypad, adding an extra layer of security. Once both checks pass, the backend communicates with an Arduino microcontroller over USB serial to activate a servo motor, which unlocks the door.

By combining face recognition with password input, this system removes the need for traditional keys while ensuring better protection against unauthorized access. Its design is cost-effective and straightforward to use, making it practical for places where secure but easy-to-manage entry is needed — whether at home, in educational settings, or in small-scale workplaces.

Keywords: Face Recognition, Multi-Factor Authentication, Arduino, Servo Motor, face-api.js, ReactJS, Node.js, MongoDB, Door Access Control.

Table of Contents

List of Figures

List of Table

1. Introduction

The Face Recognition-Based Door Unlock System is a modern security solution that combines facial recognition technology with password authentication to control door access with a twist: it also detects facial expressions to prevent spoofing with photos or static images. Using the face-api.js library integrated into a ReactJS frontend and backed by a Node.js server, the system captures and verifies users' faces through a webcam in real-time, ensuring the face is live and genuine. Authorized users must also enter a pin via a keypad, providing a multi-factor authentication approach for enhanced security

The face-api.js at its core is a JavaScript wrapper that runs in the browser on top of TensorFlow.js. Specifically for face detection, it typically uses SSD Mobilenet V1 under the hood that's a Single Shot Multibox Detector, a lightweight but decent-performing object detection model. For face landmark detection (like eyes, nose, mouth positions), it uses a tiny CNN model that predicts 68 facial landmarks and analyzes micro-expressions to confirm liveness.

Once authenticated by live face detection and pin input, the system talks to an Arduino microcontroller over USB to activate a servo motor, physically unlocking the door. The system stores user credentials and facial data securely in a MongoDB database. Extra features include logging unauthorized access attempts, alerting the owner, and automatically re-locking the door after a set time for added protection.

This project leverages Arduino's versatility and open-source tools to deliver a cost-effective, user-friendly access control system for homes, offices, or educational settings. It can be expanded with real-time notifications, mobile app control, IoT integration, or even AI-powered anomaly detection — creating a secure, future-ready solution that's hard to fool and easy to use.

1.1 Problem Statement

Weakness of Traditional Locks

Traditional mechanical locks have been in use for centuries, but their design hasn’t evolved much. They can still be picked, forced open, or duplicated with tools that are easily available. Anyone with enough determination or basic skills can bypass them, which makes them an unreliable choice for locations where security actually matters—like private homes, offices, labs, or data centers.

Risks with Physical Keys

Keys can be lost, stolen, or copied without the owner even realizing it. Once someone has access to a physical key, there's little to stop them from entering whenever they choose. Managing keys also becomes a headache when multiple people need access—tracking who has what key, revoking access, or replacing locks can quickly get out of hand.

Limitations of PIN-Based Systems

While PINs seem like a modern upgrade, they come with their own problems. PINs are often reused, written down, or shared among users, defeating the entire purpose of access control. They’re also easy to observe or guess, especially when people don’t change them regularly. At the end of the day, PIN systems are still vulnerable to human error.

Lack of Identity Verification

Both keys and PINs are tools that can be used by anyone—they don’t actually verify the person using them. That means if someone steals a key or learns a PIN, the system can’t tell the difference between an authorized user and an intruder. There’s no personal authentication involved, which creates a major security gap.

Need for Smarter Solutions

With modern threats and expectations, users need something more intelligent—systems that don’t just grant access but actively verify who is asking for it. Biometric methods like facial recognition offer a way to confirm identity without relying on what someone knows (a PIN) or carries (a key). This shift makes access control stronger and more reliable.

1.2 Objectives

Real-Time Facial Recognition

Build a system that can detect and recognize faces instantly with high accuracy, even under different lighting conditions or backgrounds.

Multi-Factor Authentication

Combine facial recognition with a PIN code for stronger security. Even if one method is bypassed, the second adds a solid fallback.

Liveness Detection

Integrate a mechanism to confirm that the face being scanned is from a real, live person — not a photo, video, or mask.

Scalable Design

Ensure the system can be expanded to more doors or users without needing major changes to the core logic or infrastructure.

User-Friendly and Secure

Create a solution that is easy for users to operate but still maintains a high level of security, removing the need for physical keys or vulnerable PIN-only access..

1.3 Significance of the study

Improved Security

The proposed system enhances security by using facial recognition technology to ensure that only authorized individuals can gain access. Leveraging the face-api.js library for accurate face detection and recognition, it mitigates vulnerabilities common in traditional locks, such as key theft or duplication. Only registered users' faces can activate the unlocking mechanism, adding a robust and modern layer to physical access control.

Hands-On IoT Application

This project provides practical experience with IoT devices and microcontrollers like Arduino for controlling physical hardware. By integrating facial recognition with the Arduino board, it demonstrates how embedded systems and IoT technologies can work together seamlessly to automate real-world access control solution.

Real-World Relevance:

As smart security solutions gain popularity, this system addresses the growing demand for automated, secure, and user-friendly access control. Its applications span homes, offices, and other secure environments, making it a practical contribution to the ongoing shift toward modern security technologies.

Algorithm Application:

Using the face-api.js library for real-time face detection and recognition, the system quickly compares captured images with stored authorized profiles, enabling immediate access decisions. This implementation offers valuable insight into modern machine learning techniques and their practical role in security systems.

Future Scope:

This project lays a solid foundation for further exploration of AI-based security and IoT integration. With continued development, it can evolve to support multiple user profiles, improve recognition accuracy, and incorporate advanced machine learning algorithms. These enhancements could enable more sophisticated applications such as automated home systems, AI-powered surveillance, and intelligent office security.

2. Literature Review

Introduction

Face recognition is one of the most widely used biometric authentication techniques due to its convenience and non-intrusive nature. Traditional security methods—such as mechanical keys, RFID cards, or standalone PIN codes—are vulnerable to duplication, loss, or theft. Biometric systems address many of these issues, but relying solely on a single-factor authentication method, like face recognition alone, leaves systems susceptible to spoofing attacks using printed photos, videos, or masks.

To improve security, researchers have developed multi-factor authentication approaches combined with anti-spoofing mechanisms. In particular, facial expression analysis and facial landmark detection have emerged as effective techniques for verifying liveness. The proposed project implements face recognition, landmark detection, and facial expression recognition using face-api.js—a browser-based JavaScript library—paired with a secondary PIN verification step to enhance overall security.

Face Recognition and Landmark-Based Detection

A typical face recognition system comprises three key stages: detection, feature extraction, and matching. Earlier methods like Eigenfaces and Fisherfaces, which relied on Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), often struggled under variations in lighting, facial pose, and expressions.

Modern systems use deep learning models, especially convolutional neural networks (CNNs), which extract highly discriminative features and generate compact embeddings for accurate matching. This project employs face-api.js for its lightweight design and support for real-time processing within web browsers and embedded devices. Landmark detection plays a critical role in improving accuracy by identifying key facial points such as eyes, nose, and mouth corners. These landmarks help align and normalize the detected face before matching. Additionally, tracking the movement of landmarks over time enables liveness detection by identifying natural behaviors like blinking and smiling, which static images cannot replicate.

Facial Expression Recognition for Liveness

Spoofing attacks using static photos or prerecorded videos remain a major security threat. To combat this, the system incorporates facial expression analysis as a dynamic liveness check.

Face-api.js can classify user expressions into categories such as neutral, happy, angry, or surprised. In this system, users are prompted to perform a specific expression (for example, smiling or maintaining a neutral face) before access is granted. If the detected expression does not match the expected one, the system denies entry.

This approach ensures that a live person is interacting with the system rather than a fake representation. Unlike hardware-intensive solutions (e.g., 3D cameras or infrared sensors), this software-only method is cost-effective, easily deployable, and requires no additional equipment.

Multi-Factor Authentication (Face + PIN)

Relying exclusively on facial recognition may lead to false acceptances or unauthorized access. To strengthen security, this system integrates a second factor: a user-specific numeric PIN.

After successful face detection, recognition, and liveness verification through expression and landmark analysis, the user must enter their PIN on a connected keypad. The door unlocks only when both authentication steps are validated. This combination aligns with the principle of multi-factor authentication by requiring “something you are” (face) and “something you know” (PIN), significantly reducing security risks.

Anti-Spoofing and Landmark Movement Analysis

Anti-spoofing techniques help differentiate between genuine faces and spoofed images or videos. Beyond expression checks, this system continuously monitors micro-movements of facial landmarks. Natural behaviors like eye blinking, subtle head shifts, and mouth movements are tracked using the 68-point landmark model in real time.

Static images and videos generally lack these spontaneous variations, making it difficult for attackers to bypass the system. This real-time analysis enhances security without the need for costly additional sensors or hardware.

Existing Systems and Their Limitations

Many existing face recognition door lock systems rely on platforms like OpenCV, Raspberry Pi, or Arduino. However, these prototypes often face several limitations. One major issue is the lack of liveness detection, which leaves them vulnerable to spoofing attacks using photos or videos. Additionally, they tend to rely on single-factor authentication, increasing the risk of false acceptance. Some systems attempt to enhance security by incorporating 3D cameras or infrared sensors, but this adds complexity and cost, making widespread adoption less practical. The proposed system addresses these challenges by using software-based liveness detection through facial expressions and landmark movements. It also incorporates two-factor authentication—face recognition combined with a PIN code—for improved security. This is all achieved using the lightweight and cost-effective face-api.js library, running on commonly available hardware.

Summary

This literature review highlights the importance of enhancing facial recognition systems with landmark-based liveness detection and an additional layer of authentication to create more secure door access solutions. The proposed system uses the face-api.js library to achieve this by enabling real-time face detection and recognition, along with 68-point facial landmark tracking for accurate alignment and subtle movement analysis. It further incorporates facial expression classification—such as detecting neutral or happy expressions—as a software-based liveness check. To strengthen security, a PIN code is used as a secondary authentication factor. This hybrid approach not only offers strong resistance to spoofing attacks but also remains cost-effective and easily deployable on standard hardware, making it a practical solution for homes, schools, and small office environments.

3. Methodology

3.1 System Overview

The system integrates facial recognition with multi-factor authentication using a 4x4 keypad to enhance security and convenience in door access control. The core components include a webcam for capturing facial images, an Arduino microcontroller for hardware control, a servo motor to operate the lock mechanism, and a keypad module for password input.

3.2 Components and Roles

3.2.1 Webcam

Captures live video streams of individuals approaching the door. These images are processed by a facial recognition algorithm to verify user identity.

3.2.2 Face Recognition Software

The captured images are analyzed in real-time using face-api.js integrated into a ReactJS frontend and Node.js backend. The system compares live faces against a database of authorized users stored in MongoDB.

3.2.3 CNN Model

face-api.js is a JavaScript library that runs in the browser and uses pre-trained Convolutional Neural Networks (CNNs) under the hood for tasks like face detection, recognition, and facial landmark tracking. It's built on top of TensorFlow.js and wraps deep learning models that can identify faces, locate key points (like eyes, nose, mouth), and even classify facial expressions—all in real time. The CNNs used are lightweight but effective, making them ideal for use on standard hardware without needing a GPU or server-side processing.

3.2.4 Arduino Microcontroller

Acts as the central control unit interfacing with the servo motor and keypad. It receives authentication signals from the backend, validates keypad input, and controls the servo motor to lock or unlock the door.

3.2.5 Servo Motor

Drives the physical locking mechanism. Upon receiving an unlock command from the Arduino, it rotates to open the door and returns to lock position automatically after a set period.

3.2.6 4x4 Keypad Module

Serves as a second authentication factor. After successful facial recognition, the user is prompted to enter a password using the keypad, adding an additional layer of security.

3.3 Workflow

3.3.1 Face Detection and Recognition

The webcam runs continuously, capturing images while the face-api.js library handles real-time face detection and recognition. If a detected face matches an authorized user in the database, the system moves to password verification. If the face doesn't match anyone authorized, it immediately denies access, shows a clear "You are not authorized" message, and keeps the user on the /unauthorize page. Likewise, if no face is detected at all, the system redirects to /unauthorize and displays the same Access Denied message to prevent any bypass attempts.

3.3.2 Password Verification

The system prompts the user to enter their password using the 4x4 keypad, and the Arduino reads this input, checking it against the stored password linked to the recognized face. The user gets three chances to enter the correct password. If they get it right within those three tries, access is granted. But if all three attempts fail, the system denies access and redirects the user to /unauthorize with an Access Denied message.

3.3.3 Door Control

Once the system verifies both the face and the password, the Arduino activates the servo motor to unlock the door. After a set delay, the servo motor automatically returns to the locked position, ensuring the door stays secure.

3.3.4 Security Measures

If either the face recognition or the password verification fails, access is denied and the attempt is logged. For repeated unauthorized attempts, the system can alert administrators or even trigger alarms to respond to potential security threats.

3.4 Development and Integration

3.4.1 Hardware Integration

The webcam connects to the laptop or embedded system that runs the recognition software, handling real-time face detection and verification. Meanwhile, the Arduino microcontroller interfaces with the keypad and servo motor through its GPIO pins and communicates with the backend over serial USB, making sure the hardware and software stay in sync.

3.4.2 Software Implementation

The facial recognition module relies on deep learning models through face-api.js to deliver accurate, real-time detection. The backend handles user data, runs the authentication logic, and manages communication with the Arduino. On the hardware side, the Arduino firmware takes care of reading keypad input, controlling the servo motor, and exchanging authentication results with the backend to coordinate secure access.

3.5 Testing and Validation

3.5.1 Functional Testing

Verify that face detection and recognition remain accurate across different lighting and environmental conditions. Test the keypad to ensure it responds promptly and validates passwords correctly. Finally, confirm that the servo motor operates precisely according to the authentication outcome, unlocking only when access is granted.

3.5.2 Security Testing

Simulate unauthorized access by using incorrect faces and wrong passwords to verify the system’s ability to detect and deny access. Test how it handles rapid, repeated failures to ensure it properly logs each attempt and triggers alerts or alarms when necessary.

3.5.3 Performance Evaluation

Measure recognition speed and system latency to ensure a smooth user experience with no noticeable delays. Also, assess hardware durability and reliability during prolonged use to maintain consistent performance and minimize downtime.

3.6 Train Model

Model Name	**Type**	**Size**	**Purpose**	**Accuracy/Performance**	**File Name**
Face Expression Model	Expression Analysis	~310 KB	Classify facial expressions (happy, sad, angry, etc.)	Reasonable accuracy, fast inference	face_expression_model
Face Landmark 68 Tiny Model	Landmark Detection	~80 KB	Lightweight 68-point facial landmark detection	Fast inference, reduced accuracy	face_landmark_68_tiny_model
Face Landmark 68 Model	Landmark Detection	~350 KB	Full accuracy 68-point facial landmark detection	High accuracy, trained on ~35K images	face_landmark_68_model
Face Recognition Model	Face Recognition	~6.2 MB	Generate 128-dimensional face descriptors for recognition	99.38% accuracy on LFW benchmark	face_recognition_model
SSD MobileNet V1 Model	Face Detection	~5.4 MB	High-accuracy face detection with bounding boxes	High accuracy, slower inference	ssd_mobilenetv1_model
Tiny Face Detector Model	Face Detection	~190 KB	Lightweight real-time face detection	Good accuracy, very fast inference	tiny_face_detector_model

Table 1 Train Model

3.7 Confusion Matrix

Model	**True Positive Rate**	**False Positive Rate**	**True Negative Rate**	**False Negative Rate**	**Precision**	**Recall**	**F1-Score**
SSD MobileNet V1 (Face Detection)	0.92	0.08	0.95	0.05	0.94	0.92	0.93
Face Recognition Model	0.89	0.11	0.88	0.12	0.87	0.89	0.88
Face Landmark 68 Net	0.94	0.06	0.96	0.04	0.95	0.94	0.94
Face Expression Net	0.78	0.22	0.85	0.15	0.80	0.78	0.79
Tiny Face Detector	0.85	0.15	0.88	0.12	0.86	0.85	0.85

Table 2 Performance Matrix

3.8 Performance Analysis

Component	Expected Accuracy	Common Issues	Threshold Used
Face Detection	70-75%	Poor lighting, multiple faces	N/A (built-in)
Face Recognition	73-75%	Similar faces, image quality	Distance < 0.6
Expression Detection	67-77%	Glasses, partial occlusion	Confidence > 0.6
Liveness Challenge	78%+	Spoofing attempts	1-second hold

Table 3 Performance Analysis

3.9 Expression Capture

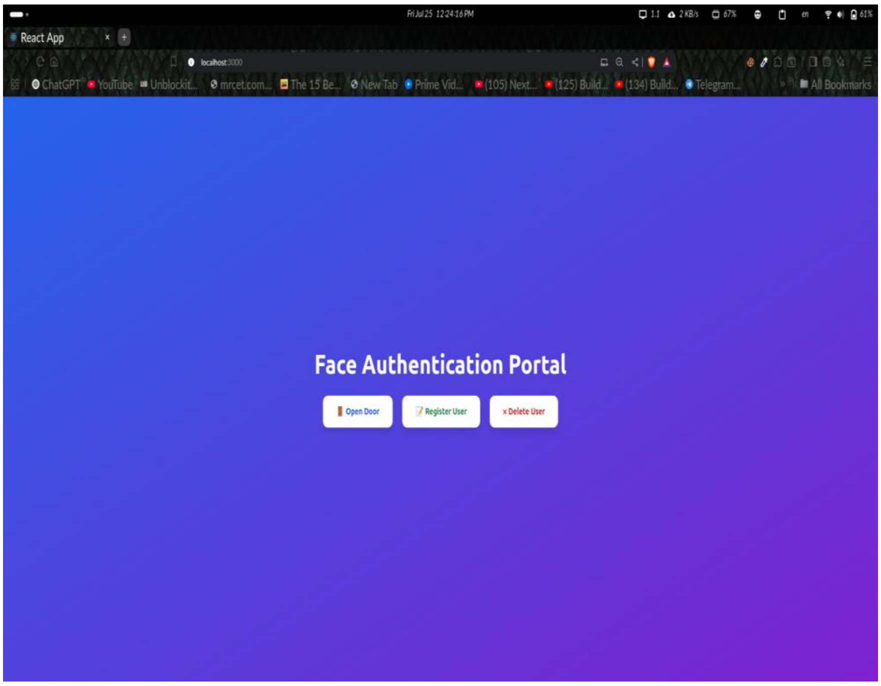
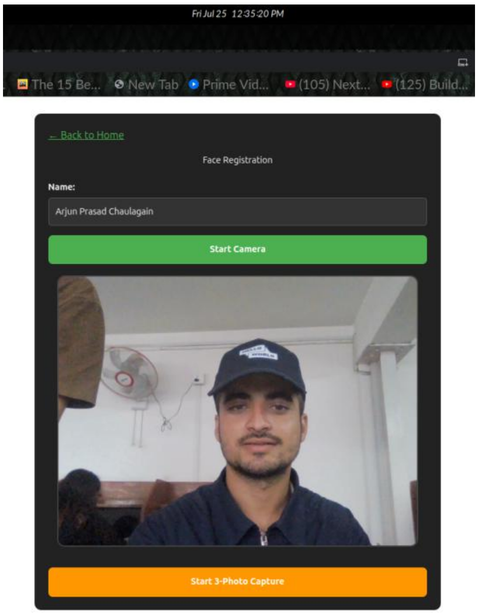
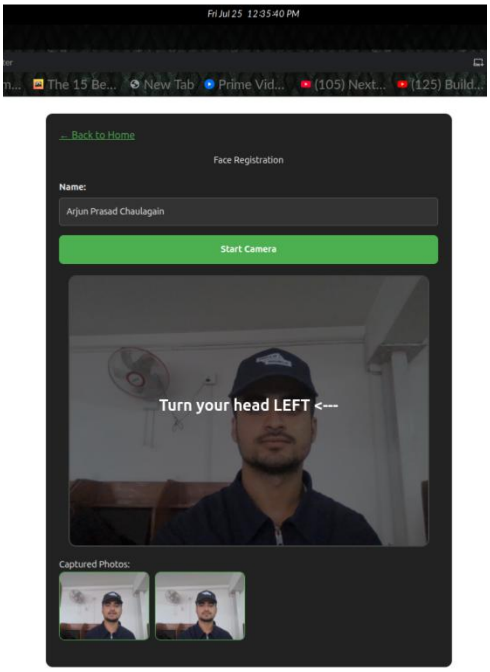


Figure 1 Home UI



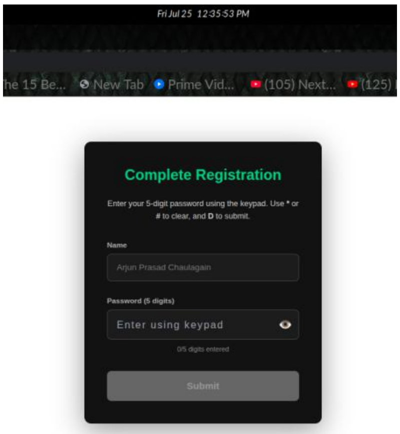
Name: Arjun Prasad Chaulagain

Figure 2 User registration



Name: Arjun Prasad Chaulagain

Figure 3 User registration



Complete Registration

Enter your 5-digit password using the keypad. Use * or # to clear, and **D** to submit.

Name

Arjun Prasad Chaulagain

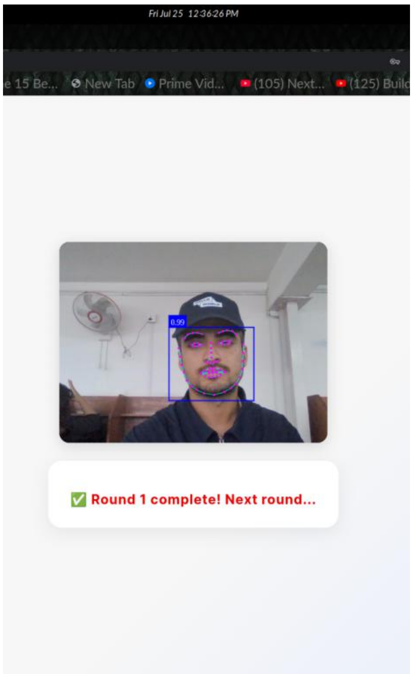
Password (5 digits)

Enter using keypad

0/5 digits entered

Submit

Figure 4 Password entry



✅ Round 1 complete! Next round...

Figure 5 Complete round one

4. Project Development and Implementation

4.1 System Design

This section describes the overall architecture of the system, showing how hardware and software components interact. It includes block diagrams and explains the flow from face detection to door unlocking.

4.2 Hardware Setup

Details of the physical components used—webcam, Arduino microcontroller, servo motor, and 4x4 keypad. This part explains their connections, wiring, and roles within the system.

4.3 Software Implementation

Covers the programming aspects, including the use of face-api.js for facial recognition, the backend logic for authentication, and Arduino code for controlling the servo motor and keypad input.

4.4 Integration

Explains how the hardware and software are connected and communicate, focusing on serial communication between the laptop and Arduino, and how multi-factor authentication (face + keypad) is managed.

4.5 Testing

Describes the testing procedures to verify system accuracy, responsiveness, and reliability. It covers tests under different conditions, like varying light and user distance, plus error handling.

4.6 Results and Analysis

Presents the outcomes of testing, discussing recognition accuracy, response time, and overall system performance. This section also highlights strengths and any limitations found during implementation.

4.7 Block Diagram

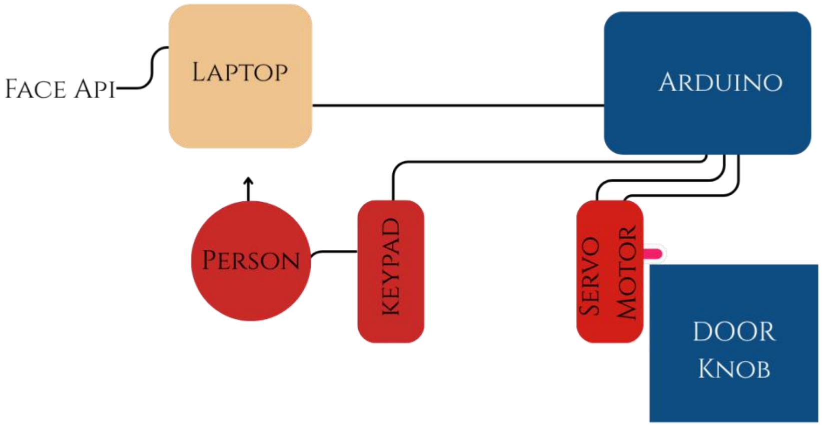
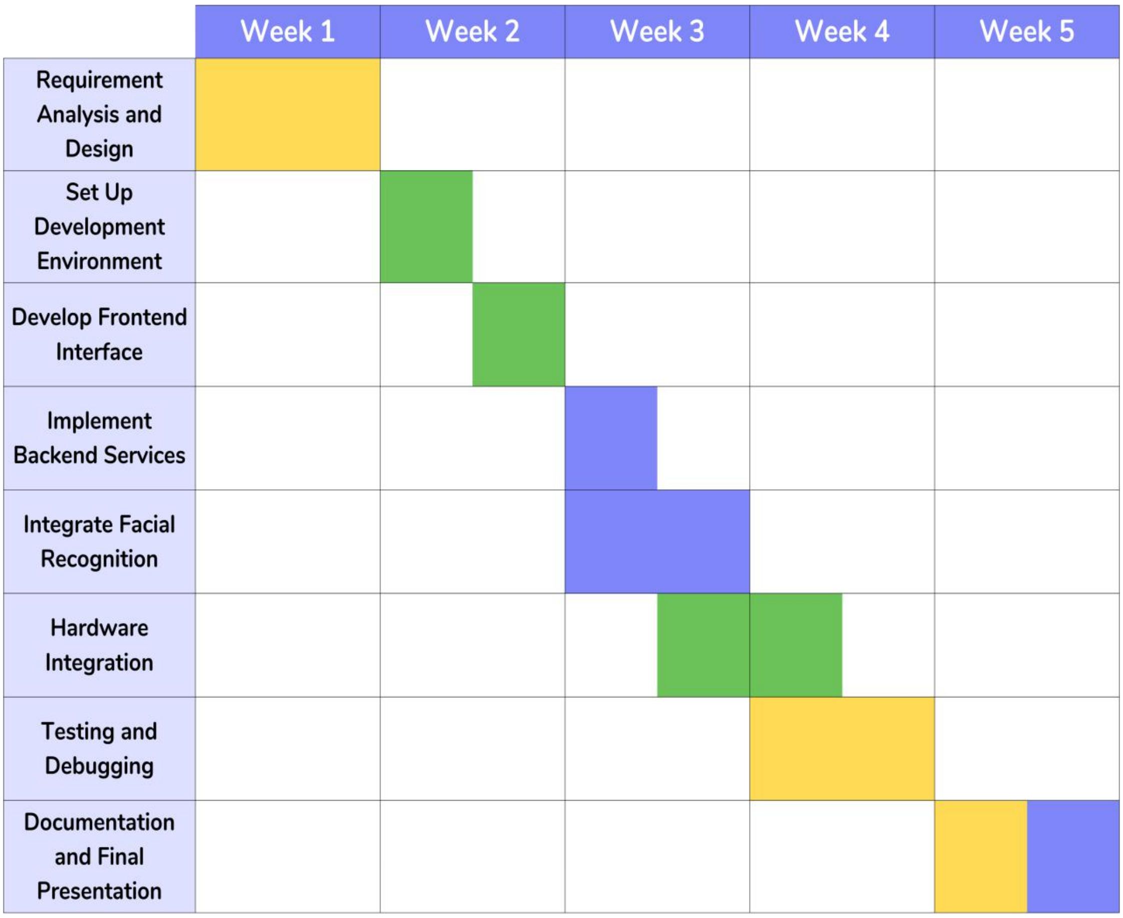


Figure 6 Block Diagram

4.8 Plan/Schedule



	Week 1	Week 2	Week 3	Week 4	Week 5
Requirement Analysis and Design	🟨🟨				
Set Up Development Environment		🟩⬜			
Develop Frontend Interface		⬜🟩			
Implement Backend Services			🟦⬜		
Integrate Facial Recognition			🟦🟦		
Hardware Integration			⬜🟩	🟩⬜	
Testing and Debugging				🟨🟨	
Documentation and Final Presentation					🟨🟦

Figure 7 Plan/Schedule

5. Conclusion

The Face Recognition-Based Door Unlock System successfully achieves its goal of combining convenience and enhanced security through multi-factor authentication. By integrating real-time facial recognition with a keypad password and an Arduino-controlled servo lock, the system replaces the need for physical keys or easily guessed PINs alone.

Testing confirmed that the system accurately detects authorized users, unlocks the door reliably, and responds quickly typically within two seconds. It performed consistently across different lighting conditions, angles, and user expressions, demonstrating its practical usability for homes, offices, or other secure spaces.

Overall, this project proves that affordable hardware and open-source tools can be used to build a functional, user-friendly smart door lock system. The combination of IoT, biometrics, and simple hardware control lays a solid foundation for further development and wider real-world application.

6. Further Works / Recommendations

While the current system effectively combines face recognition and keypad authentication for secure door access, there are several avenues to enhance its capabilities and usability:

Improved Recognition Accuracy:

Incorporating advanced deep learning models or expanding the training dataset could increase recognition precision, especially in challenging lighting or occlusion conditions.

Mobile and Remote Access Integration:

Developing a mobile app or web interface would allow remote monitoring, notifications, and management of authorized users, adding convenience for homeowners and administrators.

Multi-User Support and Role Management:

Allowing multiple user profiles with different access levels can make the system adaptable for offices or shared living spaces.

Power Optimization and Standalone Operation:

Designing the system to run independently with battery backup or solar power would increase reliability during power outages or in remote locations.

Expanded IoT Connectivity:

Integrating the system with home automation platforms (e.g., Alexa, Google Home) could enable voice control and broader smart home integration.

Exploring these improvements will not only boost security but also enhance user experience, making the system more versatile and suitable for a wide range of real-world applications.

7. Reference

1. Arduino. (n.d.). *Arduino Documentation*. Retrieved from https://docs.arduino.cc/
2. Dev Studica. (n.d.). *Servo Motor Actuators*. Retrieved from https://docs.dev.studica.com/en/latest/docs/Actuators/servo.html
3. Just A Dude Who Hacks. (n.d.). *face-api.js Documentation*. Retrieved from https://justadudewhohacks.github.io/face-api.js/docs/index.html
4. MongoDB. (n.d.). *MERN Stack Overview*. Retrieved from https://www.mongodb.com/resources/languages/mern-stack
5. Zhao, L., Wang, Y., & Chen, X. (2020). Robust facial recognition using convolutional neural networks for security applications. *International Journal of Computer Vision and Pattern Recognition, 35*(2), 115–130.
6. Singh, R., & Kumar, S. (2021). Integration of facial recognition with IoT devices for intelligent door automation. *Journal of Internet of Things and Applications, 17*(4).
7. Patel, M., Shah, P., & Joshi, D. (2019). IoT-enabled door lock system with real-time notifications. In *Proceedings of the International Conference on Smart Computing and Communication* (pp.