LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title
CS5068NI – Cloud Computing & IoT

Fingerprint-Based Door Lock System

Assessment Type
50% Group Report

Semester
2024 Spring

Group Members

London Met ID	Student Name
23047469	Aashish Chaudhari
23047396	Dibya Sitaula
23047473	Evani Raut
23047474	Kritika Bhusal
23047464	Prabesh Sundar Taksari
23047461	Sikum Limbu

Assignment Due Date: 15th May 2025
Assignment Submission Date: 15th May 2025
Submitted to: Mr. Sugat Man Shakya
Word Count: 3322

I confirm that I understand my coursework needs to be submitted online via My Second Teacher under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

IOT Group C.W (1).docx

Group Members

London Met ID

Student Name

23047469

Aashish Chaudhari

23047396

Dibya Sitaula

23047473

Evani Raut

23047474

Kritika Bhusal

23047464

Page 1 of 28 3857 words 135%

12% Overall Similarity

Filters

Match Groups | Sources

Show overlapping sources

1 Internet
www.coursehero.com 2%
5 text blocks 80 matched words

2 Submitted works
islingtoncollege on 2025-03-07 2%
1 text block 63 matched words

3 Submitted works

Abstract

This project primarily concerns the design and implementation of a fingerprint-based smart door lock using IoT technology. The system uses an ESP32-WROOM-32 microcontroller, an AS608 fingerprint sensor, and an SG90 servo motor for secure and reliable access control. The door opens for authorized users via fingerprint authentication while attempts of unauthorized access trigger notifications via the Blynk IoT platform. The system was tested successfully and demonstrated reliable performance in real-time door access control. The project demonstrates the possibility of integrating biometric authentication with IoT connectivity to improve conventional security systems and suggests further enhancements for greater applications in smart homes and commercial areas.

Acknowledgement

We would like to thank everyone who helped and supported us during this project. The guidance we received throughout the process made it easier for us to understand and complete the work successfully.

We are also thankful for the resources and environment that allowed us to learn and apply our knowledge in a practical way. Teamwork and cooperation played a big role in completing this project, and we appreciate the effort and contribution of every team member.

Lastly, we are grateful for the encouragement and support from those around us, which motivated us to do our best.

Table of Contents

Table of Figures

Table of Tables

1. Introduction

The Internet of Things (IoT) is a rapidly expanding technology that connects physical devices from simple household items to complex industrial systems to the internet, enabling them to collect, transmit, and act on data in real time. This growing network of smart devices helps individuals and organizations optimize efficiency, security, and decision-making. With billions of devices already connected across the globe, IoT has become ubiquitous in smart homes, healthcare, agriculture, transport, and especially in security systems (Terra, 2025).

Traditional locking mechanisms have been made increasingly inadequate in the context of escalating security threats over the last several years. These traditional locks can be picked or bypassed easily and offer minimal security from unauthorized access. With advancements in biometrics and IoT, more intelligent and secure access control systems are being developed to mitigate such vulnerabilities (Ferdinando, 2025).

Our project, "Fingerprint Based Door Lock System," is to design and develop a biometric security system using a fingerprint sensor along with a microcontroller, cloud storage, and an email notification system. The system ensures that the door can be unlocked only by registered users using their fingerprint. If unauthorized access is tried and access is denied three consecutive times, an alert email is automatically sent to the owner. This solution offers a smart, user-friendly, and tamper-proof approach that is an improvement on current practice.

1.1. Current Scenario

Security has never been secondary, whether it is in homes, offices, or other establishments. Most places still employ traditional locks, which can be picked or bypassed with ease by duplicate keys or pins. They do not have any alert system or way to know who's trying to open them. Since IoT has gained popularity, smart solutions are replacing outdated methods. In 2016, Venture Scanner estimated that there were 1,428 IoT startups from 46 nations, with a value of over **$25 billion**. That figure increased by another **$3 billion** in a mere three months. In another estimation, the number of IoT devices that would be connected by 2020 was over 24 billion, while the whole market size would be connected by 2020 was over 24 billion, while the whole market size would reach **$6 trillion**. This reflects the increased trust in smart systems like biometric locks for better security and control (Wise, 2025).

1.2. Problem Statement and Project as a Solution

In today's world, most people continue to use traditional door locks to secure their homes and offices. The locks, though, have a few major drawbacks. They are easily opened by duplicate keys, lock-pick tools, or even force. It is not possible to find out who tried to open the door or when it was tried, and in case an intruder tries to enter, the owner is not notified. This makes traditional locks risky and outdated when it comes to modern security needs (Locksmith, 2025).

To solve this problem, our project introduces a Fingerprint-Based Door Lock System that is smarter and safer. It uses a fingerprint reader to guarantee that only authorized users can open the door. All user data are securely stored in the cloud. If someone tries to open the door but fails three times, the system will automatically send an alert email to the owner. Therefore, the owner is in real-time notified of any suspicious entry, and illegal entry is reduced to risk. With the combination of IoT and biometric authentication, our system offers a modern, safe, and efficient solution to day-to-day security problems.

1.3. Aim and Objectives

Aim:

To design and implement a secure, cloud-integrated fingerprint door lock system with the capability to perform biometric access control and real-time intrusion alerting.

Objectives:

- To design a microcontroller-based fingerprint authentication system and fingerprint sensor.
- To Integrate ESP32 microcontroller for Wi-Fi-enabled IoT features.
- To have an alert system that generates an email after three consecutive unauthorized access attempts.

- To design the system to be scalable and compatible with widely used door lock systems.

2. Background

Security issues that have become increasingly important during the past few years have triggered widespread development of improved access control systems. The advancements of modern threats including duplication and theft have made conventional locks with keys outdated. The transition to smart locking systems activated because people needed better protection features and convenient solutions. The popularity of fingerprint-based access in biometric systems has increased because it demonstrates high accuracy together with reliability as well as immunity to duplication. The access control system based on fingerprint recognition creates exclusive security features that minimize unauthorized entry attempts (Cline, 2025).

Smart security systems have undergone major advancements thanks to IoT technologies which integrate into their operations. Blynk operates as a platform for embedded hardware-to-smartphone communication so users can monitor their devices along with managing their embedded systems remotely. Users acquire immediate system alerts in addition to maintaining lock operation through their smartphones as Blynk permits mobile-based system parameter adjustments. Biometric authentication and remote mobile control create two security barriers that let users maintain total control over their security measures from remote locations or on-site installations. A contemporary locking solution is delivered by the proposed system through its integration of IoT capabilities and fingerprint authentication technology (Blynk, 2025).

2.1. System Overview

A smart door lock system functions through the ESP32 microcontroller serving as the axis which processes data while serving as communication conduit. The ESP32 functions as a Wi-Fi-enabled microcontroller which coordinates fingerprint recognition procedures while overseeing hardware operations together with IoT communication functions. The device operates using the AS608 fingerprint sensor to track and authenticate fingerprints. After authorized fingerprint data matches with system-stored data the ESP32 executes the signal triggering the servo motor to unlock or lock the door.

The system design integrates a relay module because it enables future scalability along with flexible operation. While the relay is not required for driving the servo directly it provides a method to switch on high-voltage components when those components need activation. The system becomes more flexible because this enhancement exists. A prototyping circuit composed for testing exists in the breadboard structure and operates using a regulated 9V battery as its power source.

Using Blynk IoT platform users can access and manage the smart lock through their smartphone devices for remote functionality. With Blynk users can view lock status and get notification alerts when unauthorized access attempts happen while they can also activate the lock from anywhere with remote control. Blynk contributes a powerful user experience by allowing remote control access that proves essential for home and office security and property gate access management.

The smart lock system delivers effective access control through the fusion of biometric security limitations with cell phone-based management capabilities.

2.2. Design Diagrams

2.2.1. Block Diagram

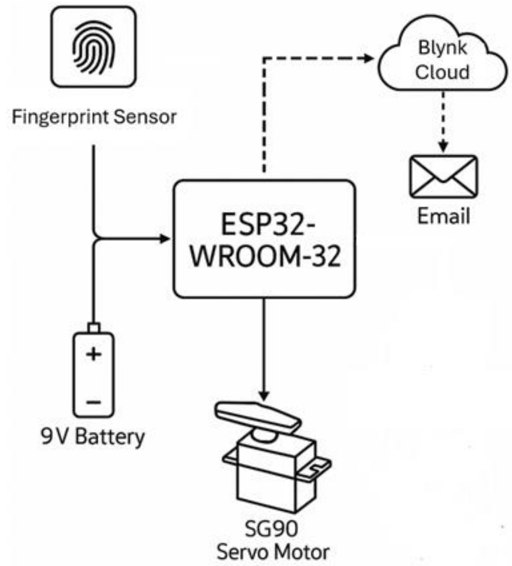

Figure 1: Block Diagram

2.2.2. Hardware Architecture

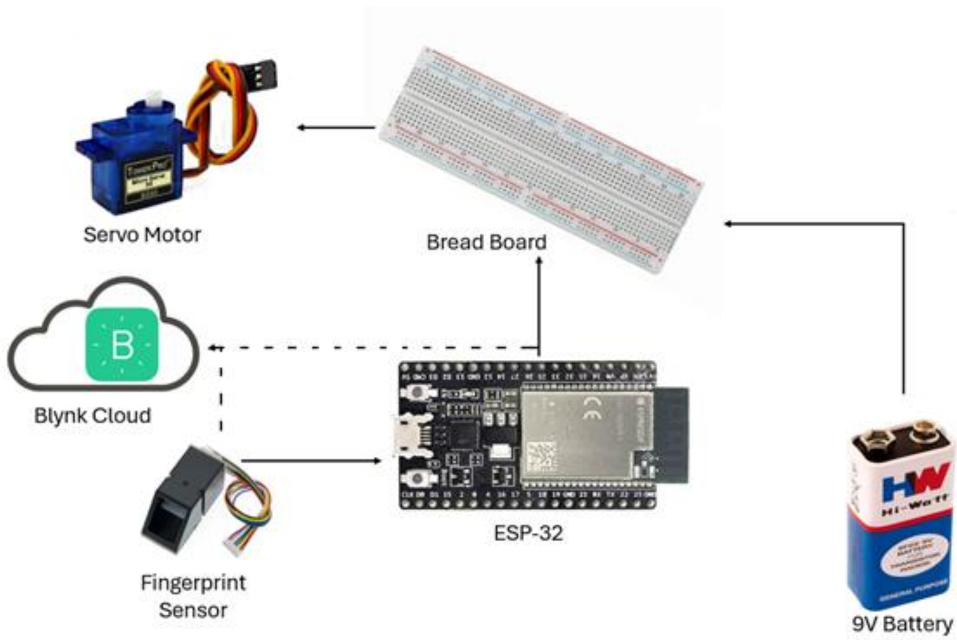

Figure 2: Hardware Architecture of the System

2.2.3. Circuit Diagram

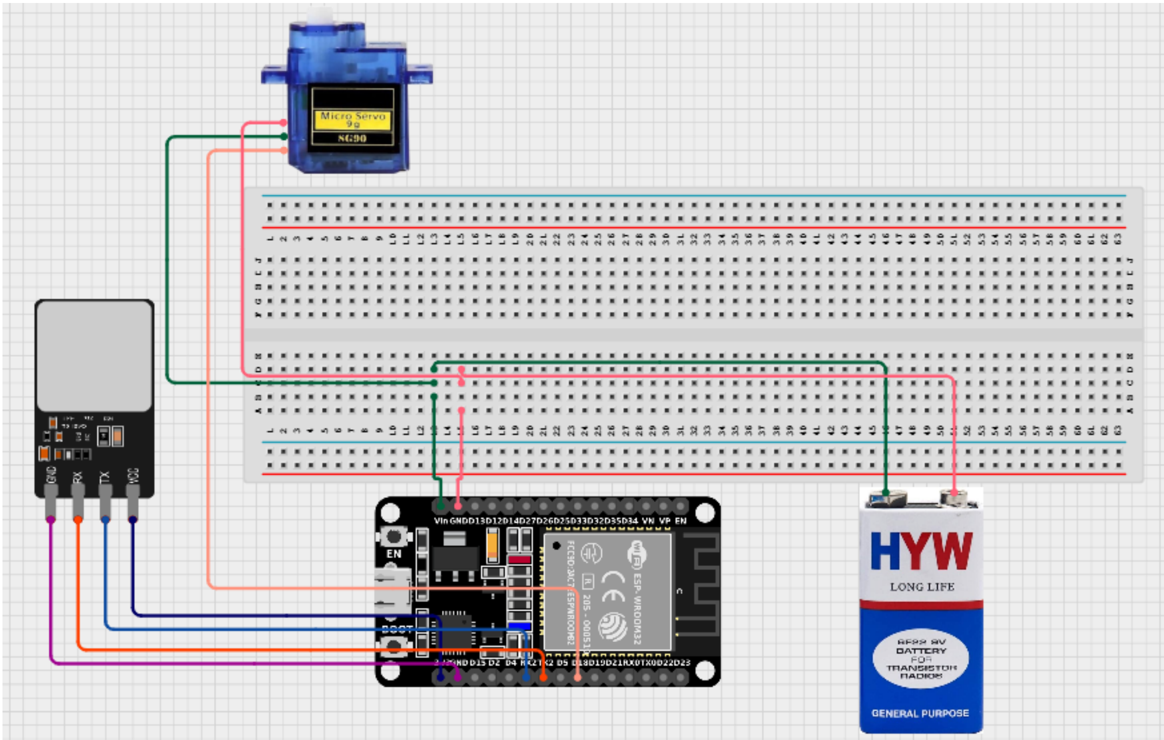

Figure 3: Circuit Diagram

2.2.4. Schematics

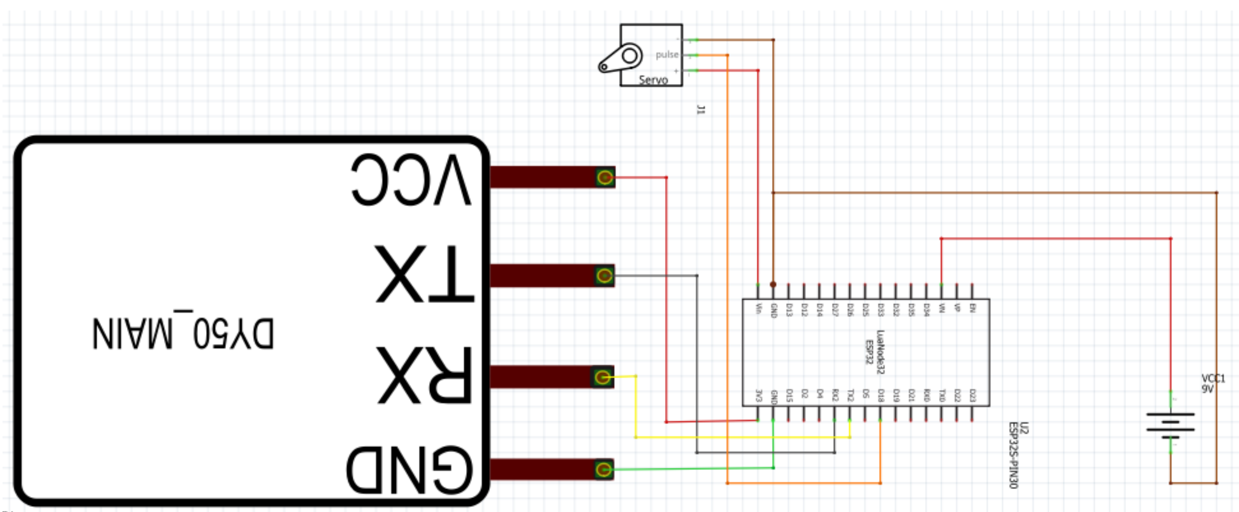

Figure 4: Schematics

2.2.5. Flowchart

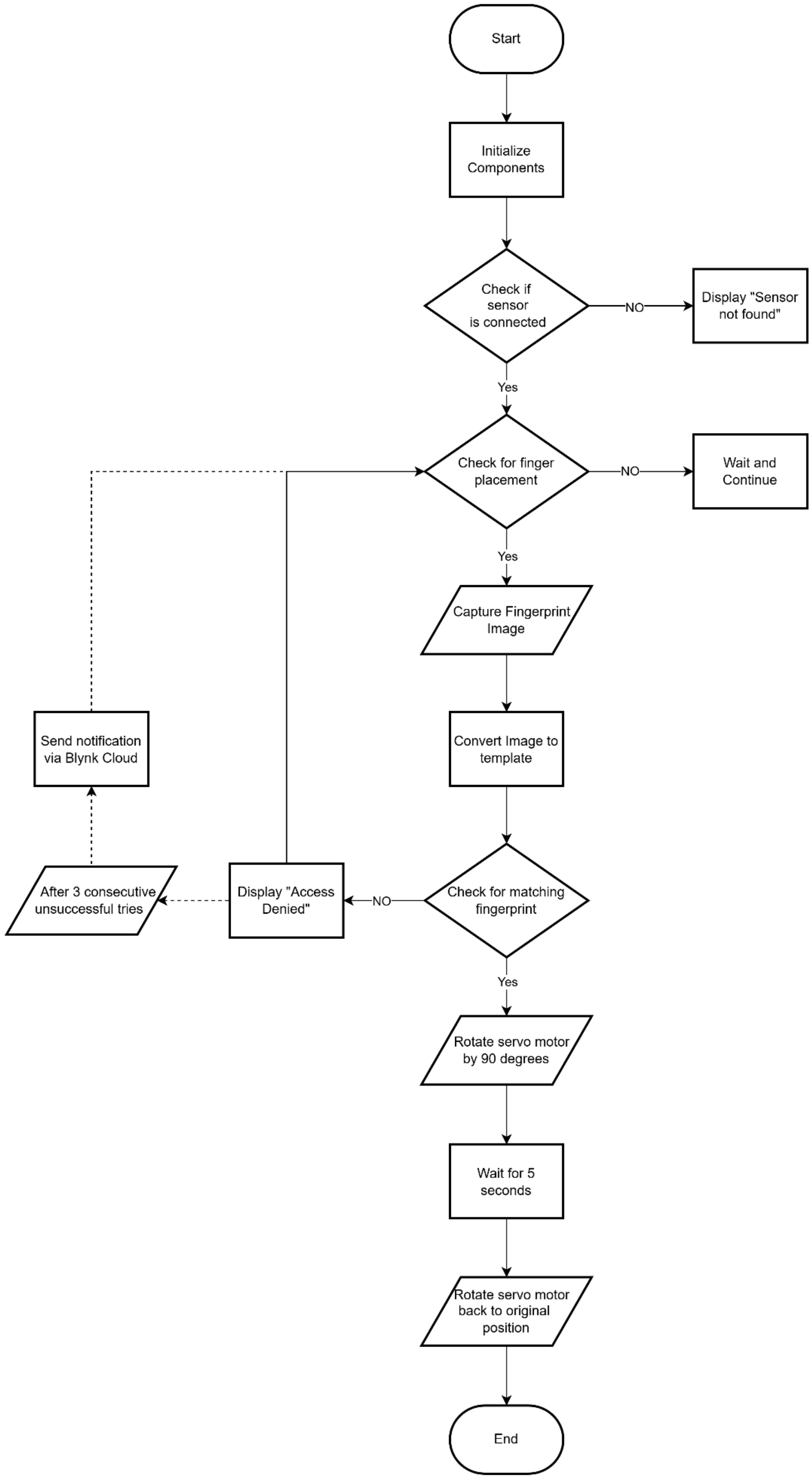

Figure 5: Flowchart

2.3. Requirement Analysis

Different tools serve as key components in this project to simplify work tasks while ensuring all data remains organized. The project implements several primary tools including the following list of tools.

2.3.1. Hardware Requirements

- **AS608 Fingerprint Sensor:** The AS608 Fingerprint Sensor operates as a miniaturized optical biometric module which obtains precise fingerprint illustrations to provide secure authentication processes. Such hardware suits applications in access management and automated presence management and smart locking needs through quick installation with dependable performance (Oku Electronics, 2025).

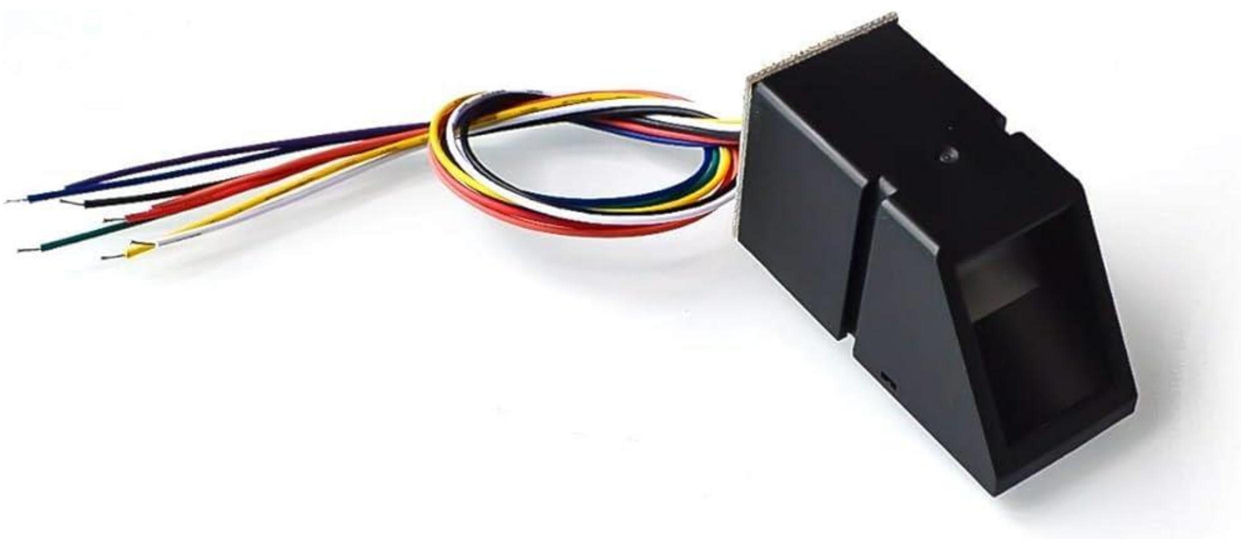

Figure 6: AS608 Optical Fingerprint Sensor

- **ESP32:** The ESP32 features dual-core processing and integrated Wi-Fi and Bluetooth capabilities in its low-cost microcontroller design which suits IoT and smart devices applications. Users can program the ESP32 through Arduino IDE or Micro Python because it includes dual-core processing and multiple input and output connections. The device offers excellent power efficiency and wireless functionality that makes it ideal for real-time automated solutions (RandomNerdTutorials, 2019).

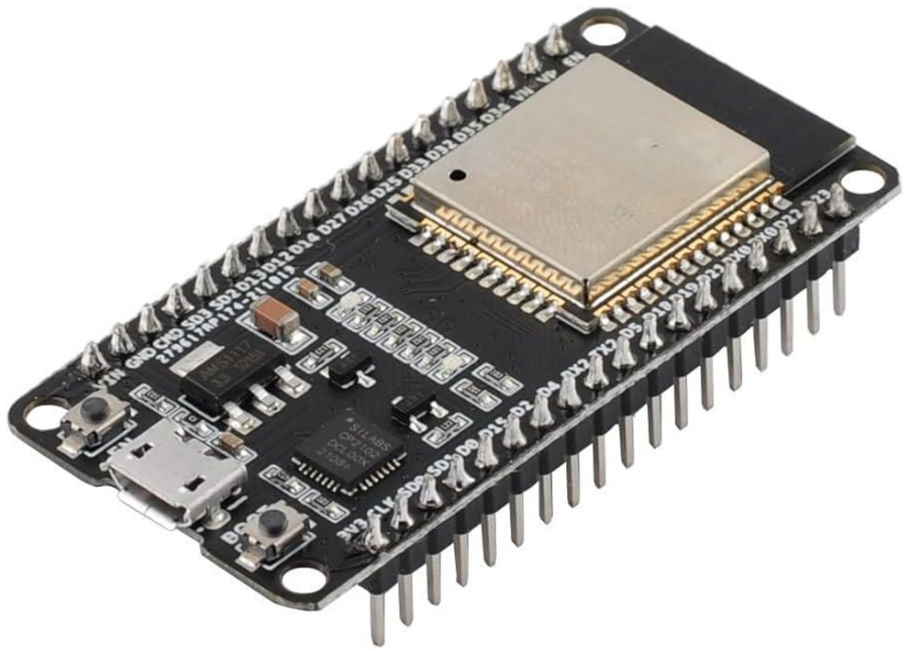

Figure 7: ESP32-WROOM-32

- **Servo Motor:** The servo motor serves as a precise motor which uses feedback control for precise angle rotation. The small yet powerful motor delivers high torque which finds applications in robotics as well as RC toys along with automation systems. This mechanism works under electric pulse control with DC or AC power input (Apoorve, 2025).

Figure 8: Servo motor

- **9v Battery:** The compact 9V battery functions as an elementary power source for operating small electronic devices as well as circuits. The device generates a reliable 9-volt output which suits microcontroller and sensor needs in portable projects (MicroBattery, 2025).

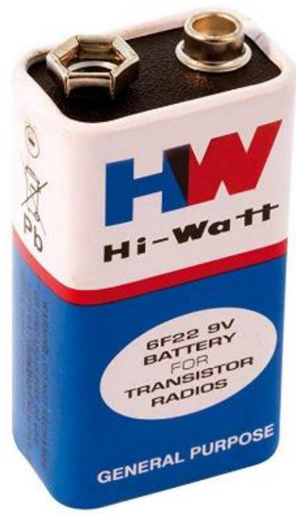

Figure 9: 9V Battery

- **Breadboard:** The breadboard serves as a testing platform for electronic circuit design during prototyping because it lacks any need for soldering. The breadboard contains a matrix of holes that use embedded spring mechanism to maintain electrical components and wiring. Due to its design with holes and internal spring clips the breadboard serves perfectly for circuit design prototyping purposes (Modern Practical Healthcare Issues in Biomedical Instrumentation, 2022).

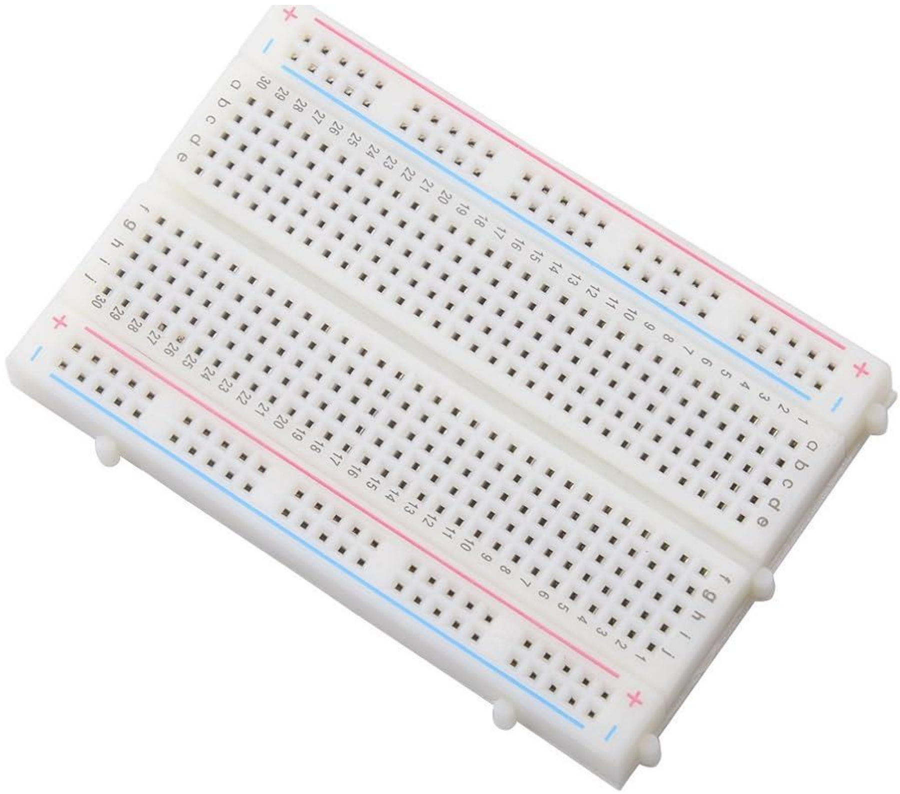

Figure 10: Breadboard

2.3.2. Software Requirements

- **MS Word:** Microsoft Word serves as a widely used text processing solution from Microsoft Corporation. Users can utilize this software to develop modify schedule records alongside presentation features that enhance document appearance together with text sharing capabilities. The software enables users to create written content through its basic functionality which includes writing letters and reports and preparing resumes. People from education, business and personal settings often use this application because of its convenient features (GeekFoeGeeks, 2025).

- **Arduino IDE:** The Arduino IDE serves as a platform that enables users to develop and transmit code to Arduino board systems through its writing and compilation functions. The Arduino IDE provides a straightforward interface during which users write C/C++ code through its code editor and toolbar and message area. Built-in libraries together with serial communication form integral elements of the IDE for debugging purposes. This software application functions as an essential tool that helps users develop and check projects which use Arduino boards (GeekForGeeks, 2025).
- **Blynk IOT Cloud:** Blynk is as an IoT platform which gives developers all necessary tools to manufacture and execute connected devices across any size scale. The platform provides users with capabilities to link hardware components to cloud services together with a framework to create mobile and web applications. Through its real-time data analysis features Blynk enables users to manage devices remotely and send notifications simultaneously which works effectively with personal projects as well as high-scale commercial applications (Blynk, 2025).
- **Fritzing:** Fritzing operates as an open-source application which helps users develop electronic project designs during prototyping. Through this platform users can construct schematics followed by designing wiring diagrams while working on their own custom-made PCBs. Users can make and distribute personal project elements through Fritzing software. The professional tool has strong usage in documenting electronic projects while offering teaching and sharing capabilities (SparkFun Electronics, 2025).

- **Draw.io:** Draw.io functions as a cost-free web-based visualization software for drafting flowcharts and mind maps with additional features for organization chart development and other visual representations. The tool saves all projects directly to Google Drive while providing simple sharing functionalities. Users can create designs via drag-and-drop functions in Draw.io while benefiting from different visual chart templates (Paraschiv, 2023).

3. Development

The development process of this Fingerprint-Based Door Lock System with Cloud Notification was executed methodically in organized successive stages. Each stage encompassed certain specific activities, for instance, planning, implementation, and testing.

3.1. Step 1: Planning and Design

The problem became apparent in this initial phase — that there were no cheap and real time security systems that would signal the owner's unauthorized access. On this basis, a plan was generated for the creation of a fingerprint-based door lock system using readily available components interfaced with IoT for cloud notifications.

The connections between the fingerprint sensor, relay module, servo motor, and ESP32 microcontroller were mapped out in a schematic diagram for hardware visualization. A different type of diagram was then created that detailed the steps taken to determine system response for valid and invalid fingerprint attempts.

Deliverables accomplished:

- Finalization of the problem statement.
- Project objectives and scope definition.
- Identification of hardware and software requirements.
- Drafting of workflow chart and hardware schematic.

3.2. Step 2: Resource Collection

After the planning stage, the hardware and software resources required were acquired.

Hardware:

- ESP32-WROOM-32 microcontroller
- AS608 optical fingerprint sensor
- SG90 servo motor
- 9V battery
- Breadboard and jumper wires

Software:

- Arduino IDE
- Blynk IoT platform
- Necessary Arduino libraries (Adafruit Fingerprint Sensor Library, Blynk Library, ESP32Servo)

The Blynk application was installed on a smartphone to enable cloud-based notifications.

3.3. Step 3: Hardware Assembly

The components were assembled on a breadboard according to the schematic diagram.

Connections:

Device	Device Pin	ESP32 Pin
AS608 Fingerprint Sensor	VCC (Red Wire)	3.3V
	GND (Green Wire)	GND
	TX (Black Wire)	GPIO16 (RX2)
	RX (Yellow Wire)	GPIO17 (TX2)
Servo Motor	VCC (Red Wire)	5V
	GND (Brown Wire)	GND

	Signal (Orange Wire)	GPIO18 (D18)
9v Battery	Positive (+)	5V
	Negative (-)	GND

Table 1: Connections between all the hardware components and the ESP32

Physical locking and unlocking were simulated by attaching the servo to a mock door latch.

Phase 1: Powering the ESP-32

The ESP-32 was connected to the computer to supply power to the ESP-32 which acted as the basic setup of the microcontroller for the initiation of the project. The ESP-32 was connected to a breadboard to allow proper connections between the sensors and actuators used in the project along with the backup battery.

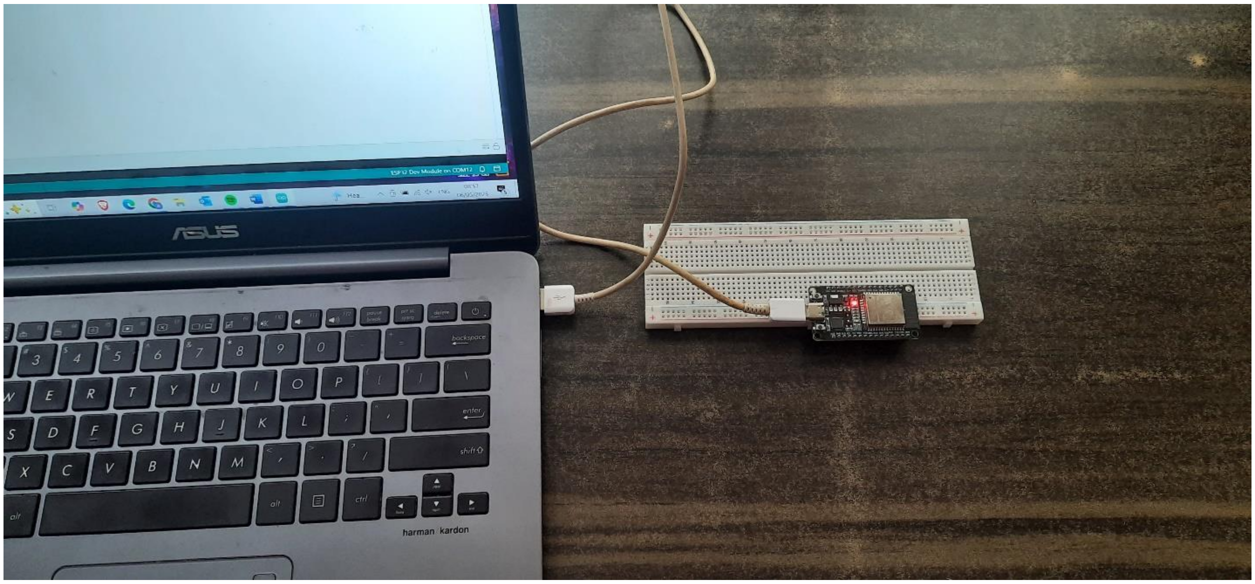

Figure 11: Connection between the ESP32 and a computer

Phase 2: Connecting the Fingerprint sensor

The AS608 Fingerprint sensor was added to the setup as the primary sensor of the project. Although there are 8 pins on the sensor, only 4 pins are used because the remaining 4 pins are used for extra functionality, futureproofing or compatibility with other systems which were not the requirement for our project. The connection of the AS608 Fingerprint Sensor to the ESP-32 are stated clearly in the Connections table above.

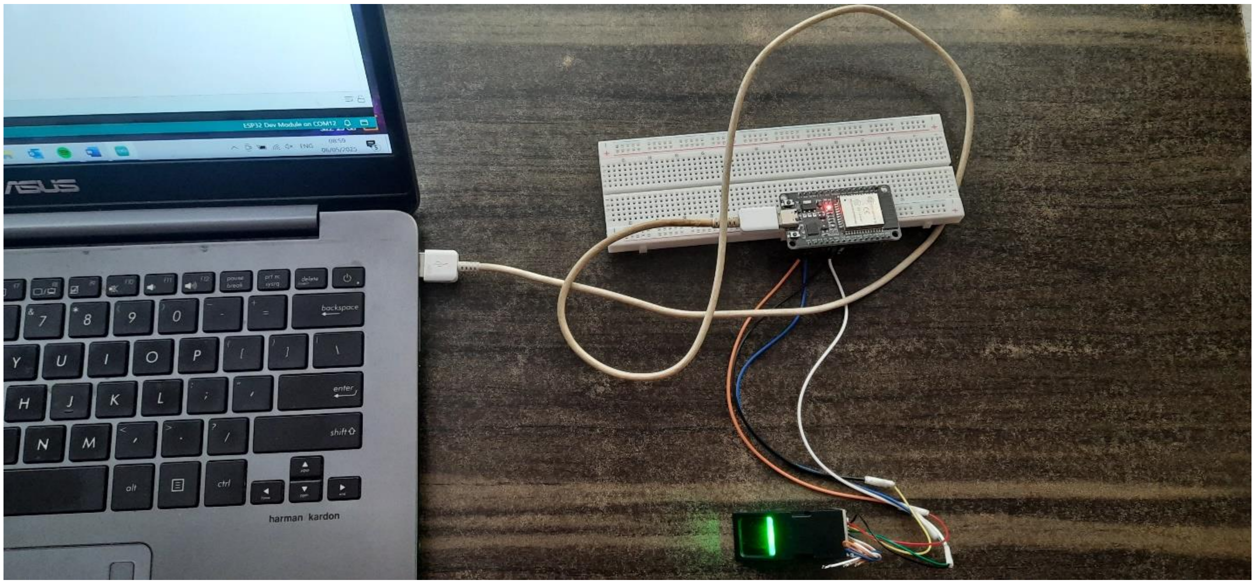

Figure 12: Adding the fingerprint sensor to the setup

Phase 3: Adding the Servo motor

The Servo Motor was added to the setup as the primary actuator of the project. When a verified fingerprint was detected by the AS608 Fingerprint sensor, the servo motor was supposed to rotate by 90 degrees. The 3 pins of the Servo motor were connected to the breadboard as shown in the Connections table above.

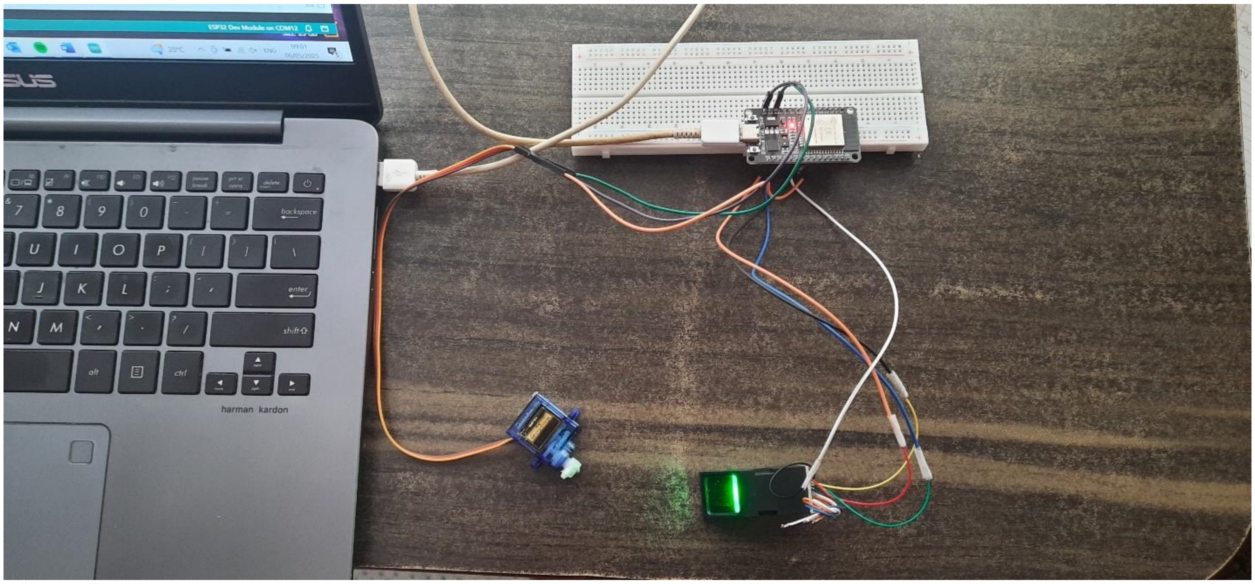

Figure 13: Adding the servo motor as the actuator for the project

Phase 4: Connecting the 9-volt battery

The whole setup was completed with the addition of a 9v battery which acted as the backup battery when the ESP-32 was not connected to the computer. The connection between the 9v battery and the breadboard is shown in the Connections table above.

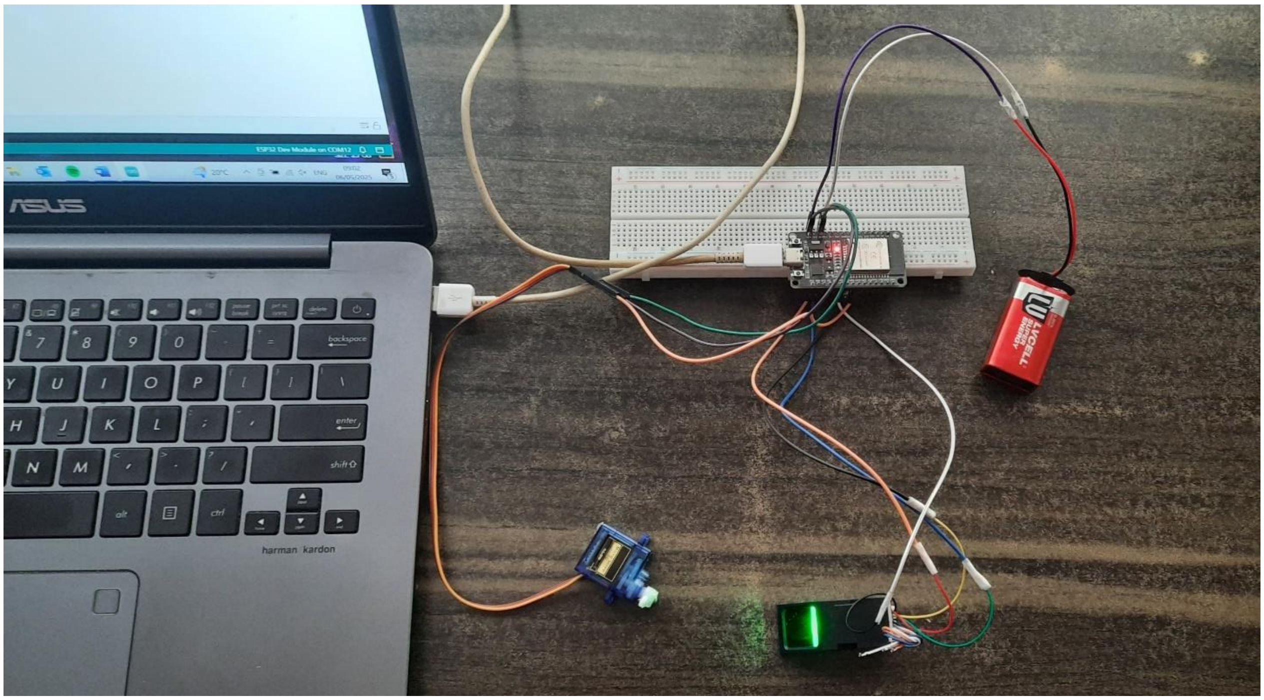

Figure 14: Adding the backup power battery to complete the setup

3.4. Step 4: System Programming

The principal project firmware was developed and uploaded to the ESP32. This code contains:

- Fingerprint detection logic.
- The methods for door unlocking and locking through relay and servo mechanisms.
- The failed Attempt Counter tracks unauthorized attempts for notification in the cloud.
- Integration with Blynk to send alerts after three consecutive failed attempts.

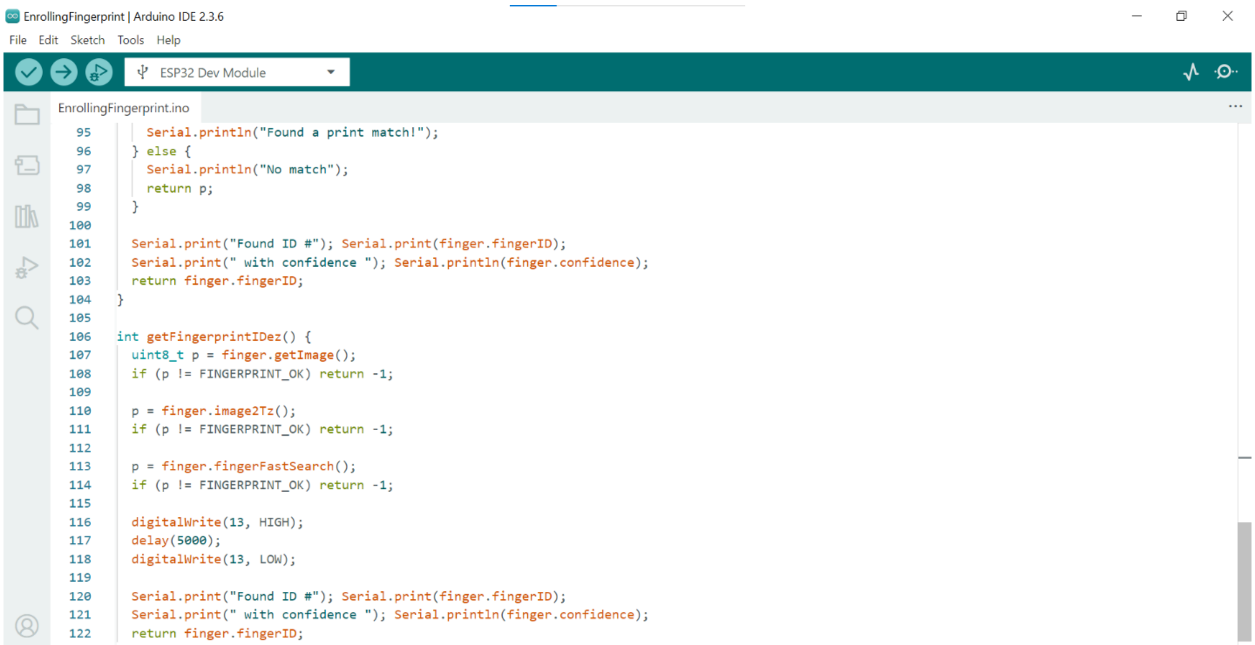

Figure 15: Code to Enroll Fingerprint

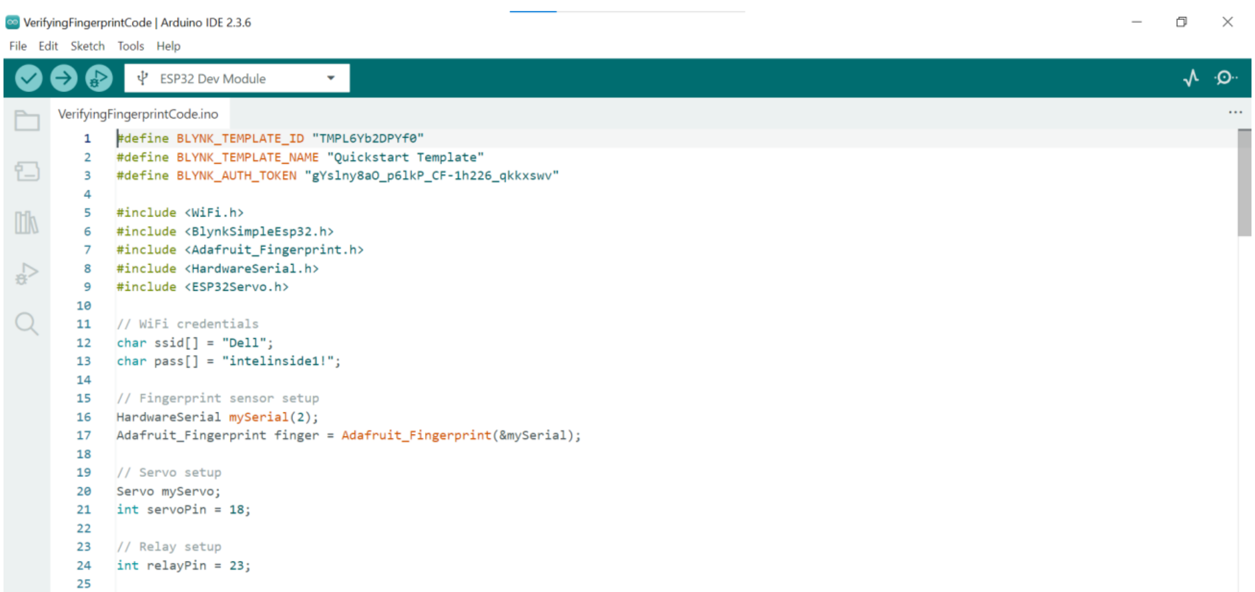

Figure 16: Code to Verify Fingerprint

3.5. Step 5: Fingerprint Enrollment

An exclusive Arduino sketch was uploaded onto the very ESP32 to enroll the approved fingerprints into the internal database of the AS608 sensor.

Process:

- Run the enrollment code from the Adafruit Fingerprint Sensor library.
- Follow serial monitor instructions for adding fingerprints with unique IDs.
- The store enrolled fingerprints inside the flash memory of the sensor.

3.6. Step 6: Blynk Cloud setup

A program has been designed for the Blynk application with:

- **Device Type**: ESP32
- **Authentication Token**: This was generated and is included in the Arduino code.
- **Email Notification Setup**: Developed an event-triggered notification that would alert the owner on reaching three failed attempts.

3.7. Step 7: System Testing and Debugging

The entire system was subjected to tests in different situations:

- Authorized fingerprint unlocking.
- Unauthorized fingerprint detection and attempt-counting.
- Cloud email alert after three failed attempts.
- Auto-locking after 5 sec.

Debugging and verification were performed using Serial Monitor outputs and Blynk notification logs.

4. Result and Findings

4.1. Result

The fingerprint-based authentication system was successfully developed, meeting all the intended objectives. The system accurately captures fingerprint data during the enrollment process and stores it securely for future use. During fingerprint verification, the system correctly matches the scanned fingerprint with the stored data and grants or denies access based on the match. If an unverified fingerprint is scanned, the system sends an email notification to the registered user, alerting them of the failed attempt. Additionally, the system provides real-time feedback, displaying clear messages on the serial monitor at each step of the enrollment and verification process to ensure smooth operation and transparency.

4.2. Findings

In this section, the expected results and the methods used to test the fingerprint authentication system were outlined. The tests were set up to check how well different parts of the system work, such as fingerprint enrollment, verification, and the email notification feature. The following tests were planned:

4.2.1. Test 1: Fingerprint enrollment Code Verification

Test	1
Objective:	To check whether the fingerprint enrollment code works properly after uploading it.
Activity:	Connect the ESP32 to the laptop and upload the enrollment code.
Expected Result:	The code should run without any issues and ask to place a finger on the sensor.
Actual Result:	The code ran successfully and displayed the message to place a finger.
Conclusion:	The test was successful.

Table 2: Test 1

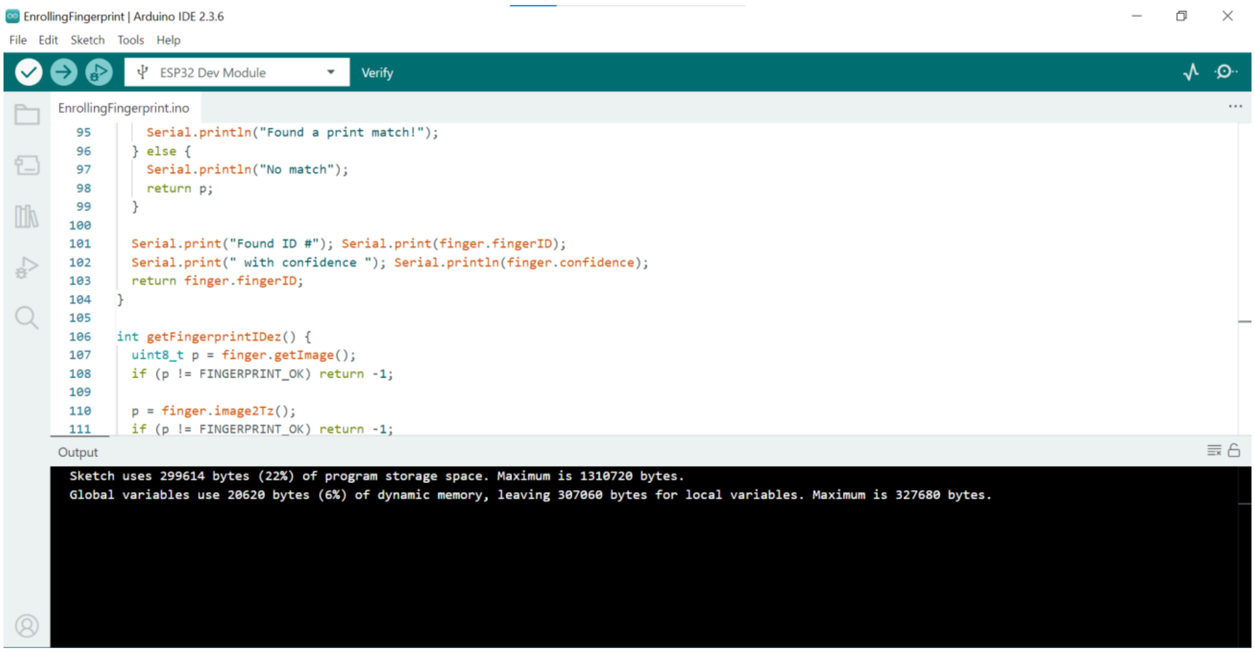

Figure 17: Successful fingerprint enrollment code verification

4.2.2. Test 2: Fingerprint Verification Code Verification

Test	2
Objective:	To verify if the fingerprint verification code works properly after uploading.
Activity:	Upload the verification code and monitor serial output.
Expected Result:	The code should run smoothly and prompt for a fingerprint verification.
Actual Result:	The code executed correctly, and the prompt appeared.
Conclusion:	The test was successful.

Table 3: Test 2

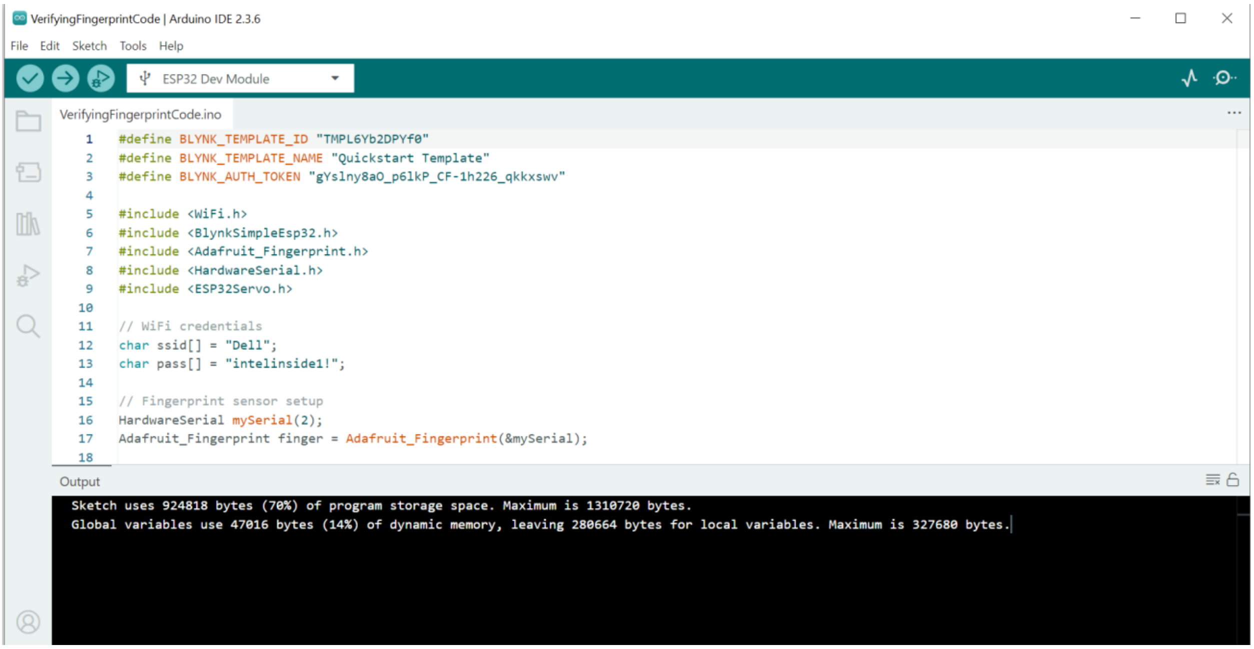

Figure 18: Successful fingerprint verification code verification

4.2.3. Test 3: Fingerprint Enrollment Test

Test	3
Objective:	To check whether the fingerprint can be successfully enrolled in the system.
Activity:	Scan a new fingerprint using the sensor and register it to the system.
Expected Result:	The fingerprint should be scanned, processed, and saved with a user ID.
Actual Result:	The fingerprint was successfully scanned and stored in the sensor's memory with a specific ID.
Conclusion:	The test was successful.

Table 4: Test 3

Figure 19: Enrolling a fingerprint

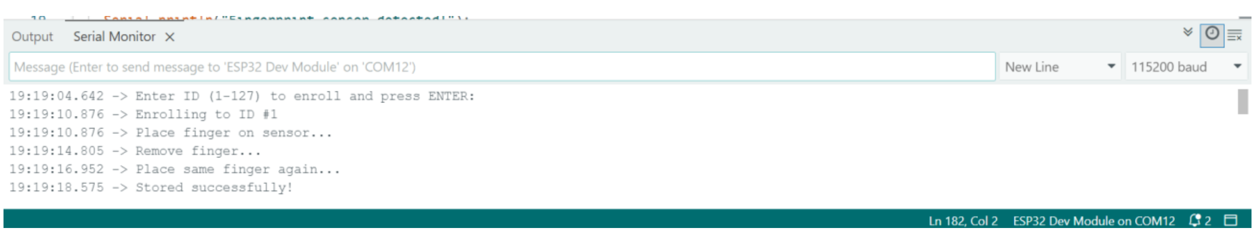

Figure 20: Successful Enrollment of a fingerprint

4.2.4. Test 4: Fingerprint Verification Test

Test	4
Objective:	To verify whether the fingerprint recognition system correctly identifies an enrolled fingerprint.
Activity:	Place a previously enrolled fingerprint on the sensor to initiate the matching process.
Expected Result:	The fingerprint should match, and the servo motor should rotate
Actual Result:	The system successfully identified the fingerprint, and the servo motor rotated 90 degrees.
Conclusion:	The test was successful.

Table 5: Test 4

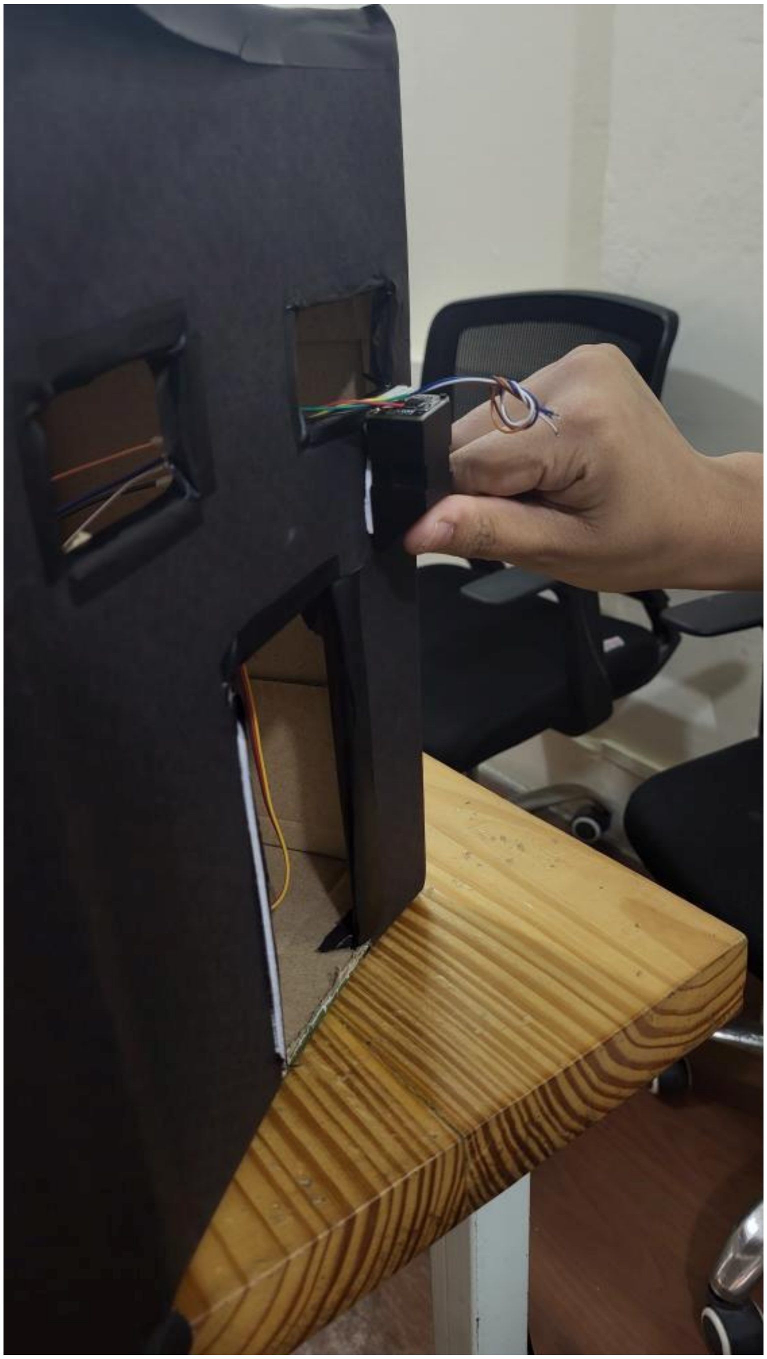

Figure 21: Unlocking the door

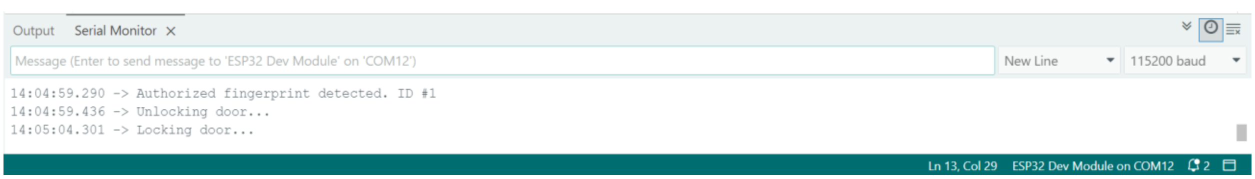

Figure 22: Successful unlocking of door

4.2.5. Test 5: Email Notification Test

Test	5
Objective:	To check whether an email notification is sent when an unverified fingerprint is detected.
Activity:	Scan a fingerprint that is not enrolled and check the registered email
Expected Result:	An email alert should be sent notifying that the fingerprint is unverified.
Actual Result:	The system successfully sent an email notification.
Conclusion:	The test was successful.

Table 6: Test 5

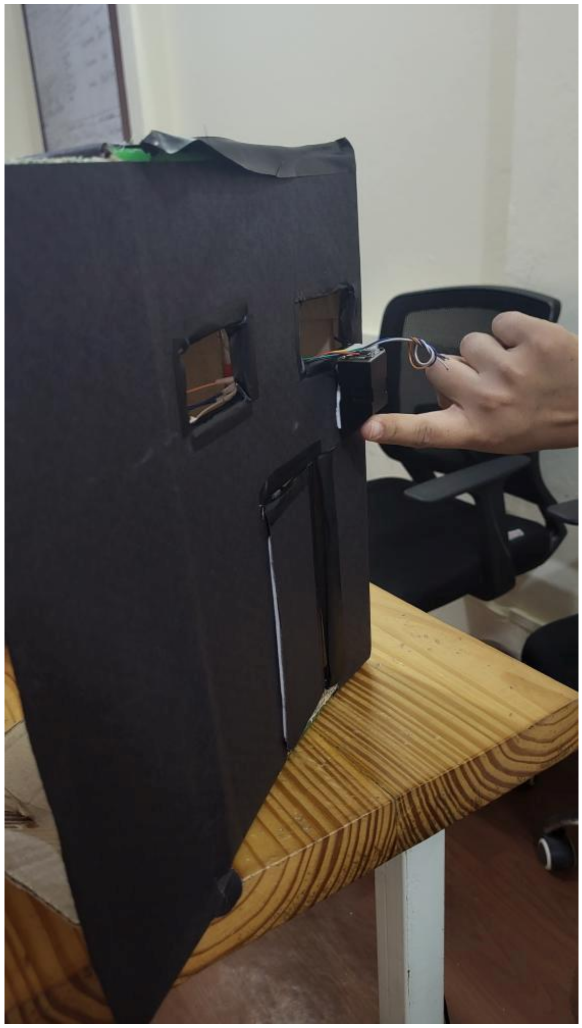

Figure 23: Unverified fingerprint not unlocking the door

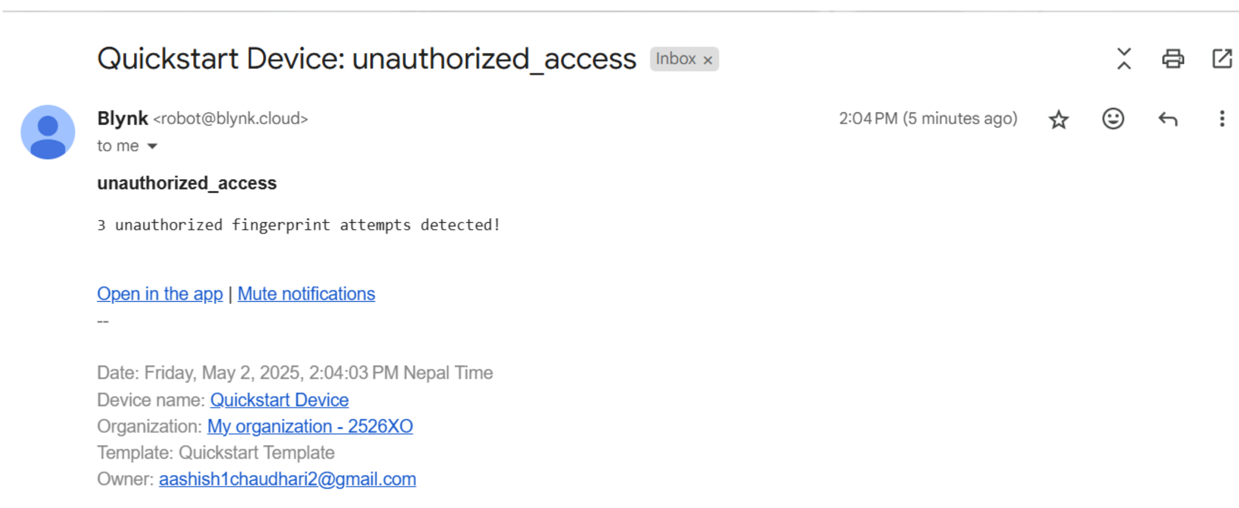

Figure 24: Successful email notification

5. Future Works

There are still a few improvements that can be made to this project to make it more effective, user-friendly, and long-lasting. Some possible ideas for future work include:

- **Adding a cloud-based database:** Right now, the fingerprint data is stored locally. In the future, it can be saved to a secure cloud platform so that data can be accessed and managed from anywhere.
- **Alternative Login Method:** Sometimes the fingerprint sensor might fail to detect a fingerprint due to sensor issues or worn fingerprints. In such cases, a manual override or PIN-based login option can be added as a backup.
- **Face Recognition Integration:** Along with fingerprints, a face recognition feature can be added for extra security, creating a two-step verification method.
- **Web Dashboard for Admins:** A simple web page can be created where the admin can view entry logs, check user data, and manage alerts without connecting to the Arduino every time.
- **Battery Status Display**: A feature can be added to show the current battery level of the device, helping users keep track of power usage and recharge it on time to avoid unexpected shutdowns.

Use Cases

1. **Office Entry and Attendance System**

   This system is used at office entrances to help manage employee check-ins. When someone scans a registered fingerprint, the door opens, and their attendance is marked at the same time. If the fingerprint isn't recognized, the system sends an email to HR or the admin team right away, so they can deal with any unauthorized access attempts.

2. **Schools and Colleges**

   This system can be set up at the entrance of schools or colleges to keep track of student attendance and prevent outsiders from entering. Only fingerprints of registered students will be accepted, making sure attendance is recorded correctly. If someone tries to enter with an unregistered fingerprint, the system alerts the school staff to help maintain safety on campus.

3. **Data Centres and Secure Labs**

   In places that need strong security, like data centres or research labs, this system ensures that only authorized people can enter. Access is granted through verified fingerprints, and if someone tries to enter with an unregistered fingerprint, the system instantly sends an alert email to the security team to stop unauthorized access.

4. **Hostels and Dorms**

   In hostels or dormitories, this system ensures that only registered residents can gain entry. It improves security and keeps track of who enters. If an unregistered fingerprint is detected, the system sends an email alert, adding an extra level of security, especially during late hours.

5. **Private Home Security**

   This system can be set up at home entrances to provide secure access. Family members and trusted guests can be registered, and if an unrecognized person tries to enter, the system immediately sends an email alert to the homeowner, ensuring real-time security and peace of mind.

6. Conclusion

The fingerprint door lock system being based on biometrics was hence designed, developed, and implemented using IoT technologies. The implementation of the ESP32-WROOM-32 microcontroller, AS608 fingerprint sensor, SG90 servo motor, and the Songle 5V relay module facilitated a secure mechanism for access control. The system enabled real-time fingerprint authentication to grant access to an authorized user of the system and deny access to an unauthorized attempt.

In addition, the project demonstrated the use of IoT connectivity for the enhancement of traditional security systems. Blynk system notifications were used for cloud notifications so that the system would generate an alert whenever it detected more than a given threshold of failed authentication attempts. The presence of this feature enhanced the overall security and monitoring capacity of the system, providing users with updates in real-time and greater control over their premises.

From the very idea of the project, ensuring that biometric authentication can be integrated with the IoT infrastructure for augmenting access control proves feasible and practical. Enhancements worth considering in the future would include a mobile app interface, integration with smart home ecosystems, and features for remote door control. Such enhancements will increase the usability of this system toward applying in wider residential as well as commercial applications.

Bibliography

Apoorve, 2025. *What is a Servo Motor? - Understanding Basics of Servo Motor Working.* [Online] Available at: https://circuitdigest.com/article/servo-motor-working-and-basics [Accessed 4 May 2025].

Blynk, 2025. *Welcome to Blynk Documentation.* [Online] Available at: https://docs.blynk.io/en [Accessed 4 May 2025].

Cline, R., 2025. *Biometric & Facial Recognition Access Control System Trends in 2025.* [Online] Available at: https://butterflymx.com/blog/facial-recognition-access-control-system/ [Accessed 6 May 2025].

Ferdinando, J., 2025. *Access Control Systems vs. Traditional Lock-and-Key Methods.* [Online] Available at: https://www.buildingsecurity.com/blog/access-control-systems-vs-lock-and-key/ [Accessed 6 May 2025].

GeekFoeGeeks, 2025. *Introduction to Microsoft Word.* [Online] Available at: https://www.geeksforgeeks.org/introduction-to-microsoft-word/ [Accessed 4 May 2025].

GeekForGeeks, 2025. *Arduino Integrated Development Environment (IDE) v1.* [Online] Available at: https://www.geeksforgeeks.org/arduino-integrated-development-environment-ide-v1/
[Accessed 4 May 2025].

Locksmith, 2025. *What Are the Problems with Traditional Locks?.* [Online] Available at: https://www.lardnerlocksmiths.com/what-are-the-problems-with-traditional-locks/ [Accessed 6 May 2025].

MicroBattery, 2025. *Battery Bios: Everything You Need To Know About The 9V Battery.* [Online] Available at: https://www.microbattery.com/blog/post/battery-bios:-everything-you-need-to-know-about-the-9v-battery/
[Accessed 4 May 2025].

Modern Practical Healthcare Issues in Biomedical Instrumentation, 2022. *Breadboard.* [Online] Available at: https://www.sciencedirect.com/topics/engineering/breadboard [Accessed 4 May 2025].

Oku Electronics, 2025. *AS608 Optical Fingerprint Module.* [Online] Available at: https://www.okuelectronics.com/store/sensors-modules-shields/as608-optical-fingerprint-module/#:~:text=Sensors%2C%20Modules%20%26%20Shields-,The%20AS608%20Fingerprint%20Reader%20Sensor%20Module%20is%20a%20cutting%2Dedge,fingerprint%20images%20with%20excep
[Accessed 4 May 2025].

Paraschiv, L., 2023. *Draw.io online – a step-by-step guide for users.* [Online] Available at: https://fotc.com/blog/draw-io-online-guide/ [Accessed 4 May 2025].

RandomNerdTutorials, 2019. *Getting Started with the ESP32 Development Board.* [Online] Available at: https://randomnerdtutorials.com/getting-started-with-esp32/ [Accessed 4 May 2025].

RandomNerdTutorials, 2019. *Guide for Relay Module with Arduino.* [Online] Available at: https://randomnerdtutorials.com/guide-for-relay-module-with-arduino/ [Accessed 4 May 2025].

SparkFun Electronics, 2025. *What is Fritzing?.* [Online] Available at: https://learn.sparkfun.com/tutorials/make-your-own-fritzing-parts/what-is-fritzing#:~:text=What%20is%20Fritzing%3F,very%20professional%2Dlooking%20wiring%20diagrams
[Accessed 4 May 2025].

Terra, J., 2025. *Real-World IoT Applications.* [Online] Available at: https://www.simplilearn.com/iot-applications-article [Accessed 1 May 2025].

Wise, J., 2025. *16 NEW Internet of Things Statistics in 2025*. [Online] Available at: https://earthweb.com/blog/internet-of-things-statistics/ [Accessed 6 May 2025].

Appendix

Code for Enrolling Fingerprint

```
#include <Adafruit_Fingerprint.h>


// Use HardwareSerial for ESP32 UART2
HardwareSerial mySerial(2);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);


void setup() {
  Serial.begin(115200);
  while (!Serial);
  delay(100);
  Serial.println("\n\nAdafruit Fingerprint system (ESP32)");


  // Initialize UART2: RX=16, TX=17
  mySerial.begin(57600, SERIAL_8N1, 16, 17);
  finger.begin(57600);
  delay(5);


  if (finger.verifyPassword()) {
    Serial.println("Fingerprint sensor detected!");
  } else {
    Serial.println("Could not find fingerprint sensor :(");
```

```
    while (1) { delay(1); }
  }

  pinMode(13, OUTPUT);
  Serial.println("Commands: e=Enroll | r=Read | s=Show Total | d=Delete");
}

void loop() {
  if (Serial.available()) {
    char cmd = Serial.read();

    if (cmd == 'e') {
      // Enrollment
      flushSerialBuffer();
      Serial.println("Enter ID (1-127) to enroll and press ENTER:");
      int id = readNumberInput();
      if (id > 0 && id < 128) {
        Serial.print("Enrolling to ID #"); Serial.println(id);
        enrollFingerprint(id);
      } else {
        Serial.println("Invalid ID! Must be between 1-127");
      }
```

```
  }
  else if (cmd == 'r') {
    // Verification
    getFingerprintID();
  }
  else if (cmd == 's') {
    // Show count
    finger.getTemplateCount();
    Serial.print("Total stored fingerprints: ");
    Serial.println(finger.templateCount);
  }
  else if (cmd == 'd') {
    // Deletion
    flushSerialBuffer();
    Serial.println("Enter ID to delete and press ENTER:");
    int id = readNumberInput();
    if (id >= 0 && id < 128) {
      deleteFingerprint(id);
    } else {
      Serial.println("Invalid ID! Must be between 0-127");
    }
  }
```

```
  }
}

// Helper function to clear serial buffer
void flushSerialBuffer() {
  while (Serial.available()) {
    Serial.read();
    delay(2);
  }
}

// Helper function to read numeric input
int readNumberInput() {
  while (!Serial.available());
  String input = Serial.readStringUntil('\n');
  input.trim();
  return input.toInt();
}

void enrollFingerprint(int id) {
  int p = -1;
```

```
// First fingerprint
Serial.println("Place finger on sensor...");
while (p != FINGERPRINT_OK) {
  p = finger.getImage();
  switch (p) {
    case FINGERPRINT_OK:
      break;
    case FINGERPRINT_NOFINGER:
      continue;
    default:
      Serial.println("Error capturing image");
      return;
  }
}

p = finger.image2Tz(1);
if (p != FINGERPRINT_OK) {
  Serial.println("Error converting image 1");
  return;
}

Serial.println("Remove finger...");
```

```
delay(2000);

// Wait for finger removal
while (finger.getImage() != FINGERPRINT_NOFINGER);

// Second fingerprint
Serial.println("Place same finger again...");
p = -1;
while (p != FINGERPRINT_OK) {
  p = finger.getImage();
  if (p == FINGERPRINT_NOFINGER) continue;
  if (p != FINGERPRINT_OK) {
    Serial.println("Error capturing image 2");
    return;
  }
}

p = finger.image2Tz(2);
if (p != FINGERPRINT_OK) {
  Serial.println("Error converting image 2");
  return;
}
```

```
  // Create model
  p = finger.createModel();
  if (p != FINGERPRINT_OK) {
    Serial.println("Fingerprints didn't match");
    return;
  }

  // Store model
  p = finger.storeModel(id);
  if (p == FINGERPRINT_OK) {
    Serial.println("Stored successfully!");
  } else {
    Serial.println("Failed to store template");
  }
}

void getFingerprintID() {
  uint8_t p = finger.getImage();
  if (p != FINGERPRINT_OK) {
    Serial.println("No finger detected");
    return;
```

```
  }

  p = finger.image2Tz();
  if (p != FINGERPRINT_OK) {
    Serial.println("Image conversion failed");
    return;
  }

  p = finger.fingerFastSearch();
  if (p == FINGERPRINT_OK) {
    Serial.print("Found ID #"); Serial.print(finger.fingerID);
    Serial.print(" with confidence "); Serial.println(finger.confidence);
    digitalWrite(13, HIGH);  // Optional: Activate lock/unlock mechanism
    delay(3000);
    digitalWrite(13, LOW);
  } else {
    Serial.println("No match found");
  }
}

void deleteFingerprint(int id) {
  uint8_t p = finger.deleteModel(id);
```

```
  if (p == FINGERPRINT_OK) {
    Serial.print("Deleted fingerprint ID #");
    Serial.println(id);
  } else {
    Serial.println("Failed to delete (invalid ID?)");
  }
}
```

Code for Verifying Fingerprint

```
#define BLYNK_TEMPLATE_ID "TMPL6Yb2DPYf0"
#define BLYNK_TEMPLATE_NAME "Quickstart Template"
#define BLYNK_AUTH_TOKEN "gYslny8aO_p6lkP_CF-1h226_qkkxswv"

#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
#include <Adafruit_Fingerprint.h>
#include <HardwareSerial.h>
#include <ESP32Servo.h>

// WiFi credentials
char ssid[] = "Islington College";
char pass[] = "I$LiNGT0N2025";
```

```
// Fingerprint sensor setup
HardwareSerial mySerial(2);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

// Servo setup
Servo myServo;
int servoPin = 18;

// Relay setup
int relayPin = 23;

// Failed attempt counter
int failedAttempts = 0;
const int maxAttempts = 3;

void setup() {
  Serial.begin(9600);

  // Connect to WiFi
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
```

```
  Serial.print(".");
 }
 Serial.println("\nWiFi connected");

 // Start Blynk
 Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);

 // Initialize servo
 myServo.attach(servoPin);
 myServo.write(0); // Locked

 // Initialize relay
 pinMode(relayPin, OUTPUT);
 digitalWrite(relayPin, LOW); // Relay off

 // Initialize fingerprint sensor
 mySerial.begin(57600, SERIAL_8N1, 16, 17);
 Serial.println("\nFingerprint Door Unlock System");

 if (finger.verifyPassword()) {
  Serial.println("Found fingerprint sensor!");
 } else {
```

```
    Serial.println("Did not find fingerprint sensor :(");
    while (1) { delay(1); }
  }

  Serial.println("System ready. Waiting for valid fingerprint...");
}

void loop() {
  Blynk.run(); // Keep Blynk connection alive

  int fingerprintID = getFingerprintID();
  delay(50);

  if (fingerprintID != -1) {
    Serial.print("Authorized fingerprint detected. ID #");
    Serial.println(fingerprintID);
    failedAttempts = 0;

    unlockDoor();
    delay(5000);
    lockDoor();
  }
```

```
}


int getFingerprintID() {
  if (finger.getImage() != FINGERPRINT_OK) return -1;
  if (finger.image2Tz() != FINGERPRINT_OK) return -1;
  if (finger.fingerFastSearch() != FINGERPRINT_OK) {
    failedAttempts++;
    Serial.print("Failed attempts: ");
    Serial.println(failedAttempts);

    if (failedAttempts >= maxAttempts) {
      Serial.println("Max attempts reached! Sending alert...");
      Blynk.logEvent("unauthorized_access", "3 unauthorized fingerprint attempts detected!");
      failedAttempts = 0;
    }
    return -1;
  }

  return finger.fingerID;
}


void unlockDoor() {
```

```
  Serial.println("Unlocking door...");
  digitalWrite(relayPin, HIGH);
  myServo.write(90);
}

void lockDoor() {
  Serial.println("Locking door...");
  myServo.write(0);
  digitalWrite(relayPin, LOW);
}
```

Individual Contribution

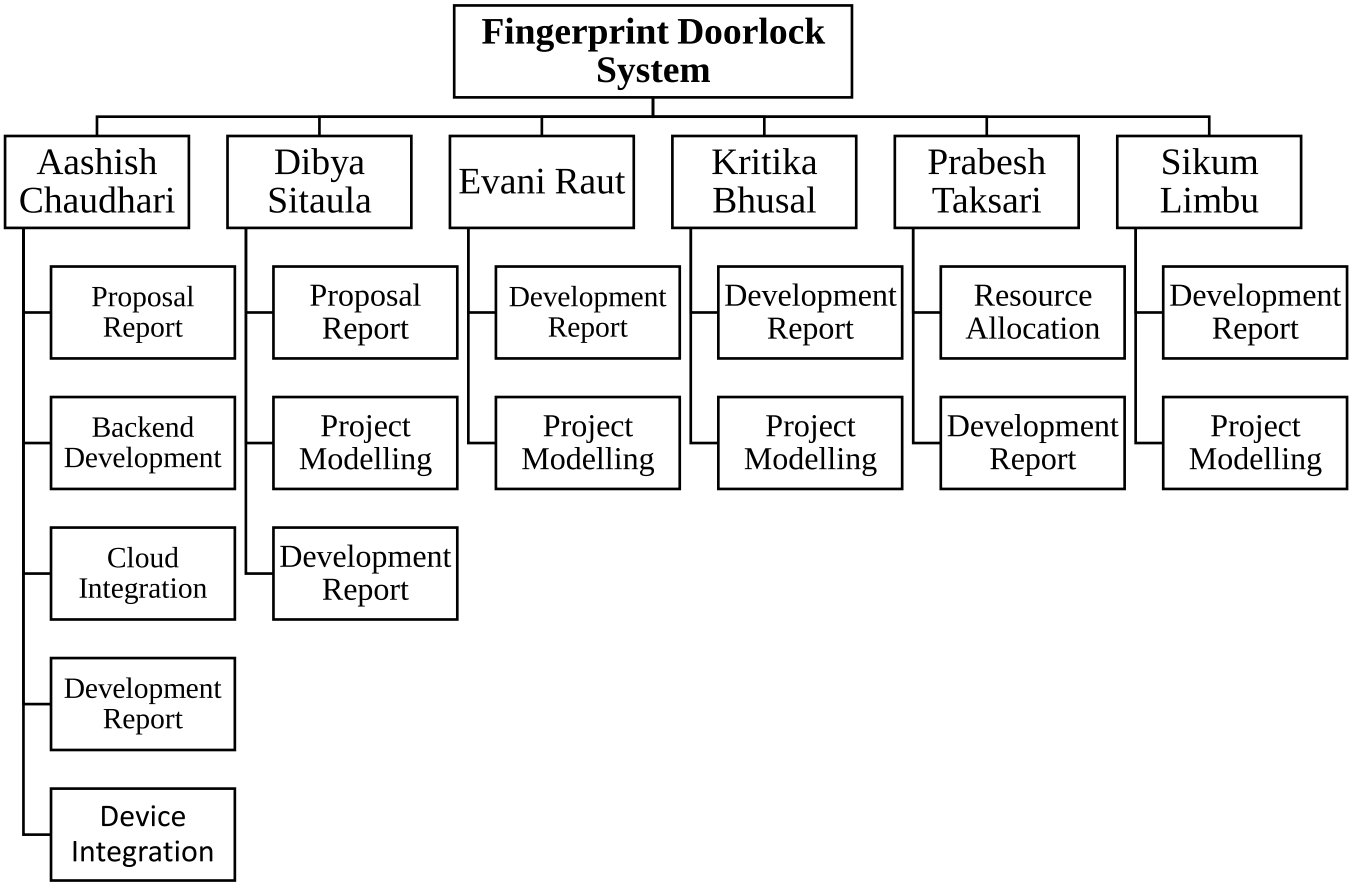

Figure 25: Work Breakdown Structure

Team Member's Name	Contribution
Aashish Chaudhari	**Proposal Report:** Introduction, Aim & Objectives, Conclusion **Backend Development, Cloud Integration & Device Integration:** Testing and Debugging the code, connecting all the hardware components and integrating the whole system with the Blynk Cloud **Development Report:** Design Diagrams and Development phases
Dibya Sitaula	**Proposal Report:** Background, Requirement Analysis

	Development Report: Abstract, Acknowledgement, Background, System Overview, Circuit Diagram **Project Modelling:** Designing a cardboard base creating a secure structure to hold all components in place while adding design elements and a paint finish to make it both functional and visually appealing
Evani Raut	**Development Report:** Result, Testing, Future works, Use cases, Conclusion **Project Modelling:** Designing a cardboard base, creating a secure structure to hold all components in place while adding design elements and a paint finish to make it both functional and visually appealing.
Kritika Bhusal	**Development Report**: Introduction, Current scenario, Problem statement and project as a solution, Aims and objectives. **Project Modelling:** Designing a cardboard base creating a secure structure to hold all components in place while adding design elements and a paint finish to make it both functional and visually appealing
Prabesh Sundar Taksari	**Resource Allocation:** Selected and managed essential hardware components to ensure efficient and smooth system development. **Development Report:** Design Diagram and Software Requirements
Sikum Limbu	**Development Report:** Hardware Requirements **Project Modelling:** Designing a cardboard base creating a secure structure to hold all components in place while adding design elements and a paint finish to make it both functional and visually appealing

Table 7: Individual Contribution