# NANDHA ENGINEERING COLLEGE

**ERODE–638052 (Autonomous)**

**(Affiliated to Anna University, Chennai)**

**DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**

**22AIC14 – INTERNET OF THINGS AND ITS APPLICATIONS**

**MINI PROJECT REPORT ON**

# FINGERPRINT BASED EXAM HALL AUTHENTICATION

**Submitted by**

| REGISTER NUMBER | NAME |
|---|---|
| 22AI012 | DHANUSYA S |
| 22AI038 | PRASSANNAA S M |
| 22AI042 | ROHINI R |

NANDHA ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

## BONAFIDE CERTIFICATE

**This is to certify that the project work entitled "FINGERPRINT BASED EXAM HALL AUTHENTICATION" is the Bonafide work of DHANUSYA S (22AI012), PRASSANNAA S M (22AI038), ROHINI R (22AI042) who carried out the work under my supervision.**

**Signature of the Supervisor**
**Dr. K. Lalitha,**
**Professor,**
**Department of AI & DS,**
**Nandha Engineering College,**
**Erode – 638052.**

**Signature of the HOD**
**Dr. P. Karunakaran,**
**Head of the Department,**
**Department of AI & DS,**
**Nandha Engineering College,**
**Erode – 638052.**

**Submitted for End semester PBL review held on ______________**

# FINGERPRINT-BASED EXAM HALL AUTHENTICATION

## AIM:

To design and implement a Fingerprint-based exam hall authentication system that enhances security and prevents unauthorized access by automating the entry process using biometric verification. This system ensures that only authorized individuals are granted entry, promoting a secure and efficient environment for examination venues.

## SCOPE:

This project provides a solution to a persistent challenge in exam hall security—ensuring accurate and reliable authentication of candidates. The system is designed to be highly effective in various applications, such as school examinations, competitive tests, and secure entrance systems. By leveraging biometric technology, the solution offers real-time authentication and enhances operational efficiency.

## BRIEF HISTORY:

Traditionally, exam hall security relies on manual verification methods like checking ID cards and attendance lists, which were time-consuming, labor-intensive, and susceptible to human error. Basic techniques lacked precision in preventing impersonation and unauthorized entry. However, recent advancements in biometric technologies, especially fingerprint recognition, have transformed exam hall authentication. Automated systems now provide real-time candidate verification, ensuring only authorized individuals can enter. These systems enhance efficiency, minimize human error, and significantly boost security, offering a more streamlined and reliable solution for modern exam management.

## PROPOSED METHODOLOGY:

The methodology for the fingerprint-based exam hall authentication system is as follows:

1. The system is set up by installing a fingerprint sensor (R307) at the entrance of the exam hall to scan candidates' fingerprints. The data collected from the sensor is processed using an Arduino Uno.
2. The scanned fingerprint data is compared with pre-registered fingerprint templates stored in the system's database to verify the candidate’s identity.

3. If the fingerprint matches, the system activates a servo motor to automatically unlock the door, granting access to the exam hall. If the fingerprint does not match, the door remains locked, and the LCD displays a message indicating access denial.
4. The system also provides real-time feedback on the LCD to confirm whether access has been granted or denied, enhancing the security and efficiency of the exam hall entry process.

## COMPONENTS REQUIRED:

| S.NO | HARDWARE | QUANTITY |
|---|---|---|
| 1 | R307 Fingerprint sensor module | 1 |
| 2 | Arduino UNO | 1 |
| 3 | Servo Motor | 1 |
| 4 | LCD 16x2 | 1 |
| 5 | Jumper Wires | As required |
| 6 | USB Cable | 1 |
| 7 | Proximity IR sensor | 1 |

## DESCRIPTION:

The fingerprint-based exam hall authentication system is an IoT-driven solution that employs a fingerprint sensor and an Arduino Uno microcontroller to authenticate candidates in real time. The system works by installing a fingerprint sensor at the entrance of the exam hall to scan the candidate's fingerprint. This data is processed by the Arduino Uno, which compares the fingerprint to pre-registered templates stored in the system’s database.

If the fingerprint matches, access is granted, and the system activates a servo motor to unlock the door. An LCD displays a message such as **"Access granted for ID: [ID Number],"** and the candidate is permitted to enter. If the fingerprint does not match, access is denied, and the LCD shows **"Access denied,"** keeping the door locked. This automated process eliminates the need for manual verification, ensuring only authorized individuals are allowed entry.

For local alerts, the system provides real-time feedback on the LCD, notifying the candidate of the authentication result. Additionally, the system can be integrated with a remote monitoring platform, allowing exam administrators to track and monitor entry status from a distance. This feature provides additional convenience and oversight, making it ideal for large-scale or high-security exam environments.

The system is designed with simplicity, security, and efficiency in mind. It is cost-effective, scalable, and easy to deploy, making it suitable for various examination centers. By utilizing fingerprint recognition, the system guarantees accurate, secure, and efficient candidate verification, improving the integrity of the exam process and reducing the risk of fraud or impersonation.

In conclusion, this project addresses the challenge of secure exam hall access and enhances exam management by automating the entry process, improving security, and boosting operational efficiency. Its ability to provide accurate, real-time feedback ensures smooth entry, making the exam experience better for both candidates and administrators.

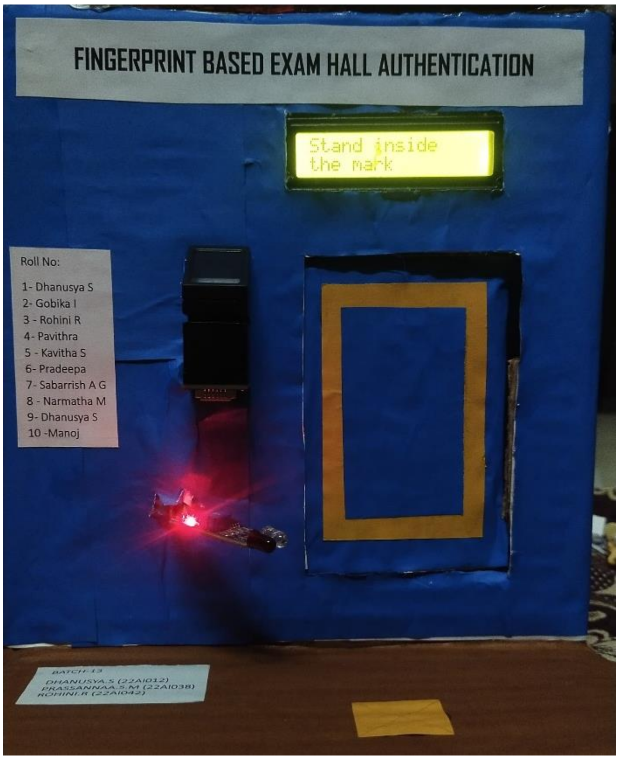
## CODING:

```
#include <LiquidCrystal.h>
#include <Adafruit_Fingerprint.h>
#include <Servo.h>
#include <SoftwareSerial.h>

// Set up the SoftwareSerial connection to the fingerprint sensor
SoftwareSerial mySerial(2, 3); // RX, TX
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
// Initialize Servo Object
Servo servo;
// Initialize the library with the LCD pins
LiquidCrystal lcd(12, 11, 6, 7, 5, 4);

// Define the pin where the IR sensor is connected
const int sensorPin = 10;
bool initMsg;
void setup() {
  // Initialize the serial monitor
  Serial.begin(9600);
  servo.attach(9);      // Attach the servo to pin 9
  servo.write(90);      // Set the servo to 0 degrees initially
  delay(1000);
  // Set up the LCD's columns and rows
  lcd.begin(16, 2);
  delay(1000);
  initMsg = false;
  // Set the sensor pin as input
  pinMode(sensorPin, INPUT);
  Serial.println("Fingerprint sensor initializing...");
  delay(1000);
  // Initialize fingerprint sensor
  finger.begin(57600);
      delay(1000);
  if (finger.verifyPassword()) {
    Serial.println("Found fingerprint sensor!");
    displayMessage("Sensor ready","");
    delay(2000);
    displayMessage("Stand inside ","the mark");
  } else {
    Serial.println("No fingerprint sensor");
    displayMessage("Sensor error","");
    while (1); // Halt the program if sensor not found
  }
}
void displayMessage(String message1, String message2) {
  // Clear the LCD only if the message is different to avoid flicker
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(message1);
    lcd.setCursor(0, 1);
    lcd.print(message2);
```

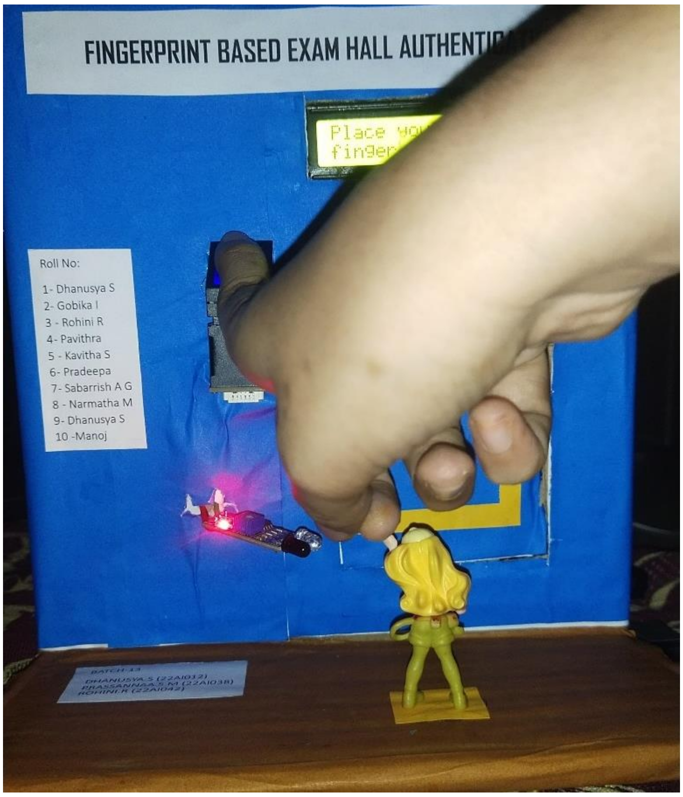
```
}
void closingCountDown(){
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Door closes in:");
    for(int i = 5; i > 0; i--){
    lcd.setCursor(0, 1)
    lcd.print(String(i) + " Seconds");
    delay(1000);
    }
}
void loop() {
  // Read the IR sensor value
  int sensorValue = digitalRead(sensorPin);
  if(!initMsg){
   displayMessage("Stand inside ","the mark");
   initMsg = true;
  }
  while (sensorValue == LOW)
  {
    unsigned long lastPollTime = 0;
    const unsigned long pollInterval = 200; // Poll every 200ms
    bool detected;
    uint8_t d;
    displayMessage("Place your ","finger");
    do{
      delay(1000);
      d = finger.getImage();
      detected = d == FINGERPRINT_OK;
      sensorValue = digitalRead(sensorPin);
    } while(d != FINGERPRINT_OK && sensorValue == LOW);
    if(detected)
    {
      // Convert the image
      uint8_t p = finger.image2Tz();
      if (p == FINGERPRINT_OK){
        // Search for a match
        p = finger.fingerSearch();
        if (p == FINGERPRINT_OK) {
          String matchMessage = "ID: " + String(finger.fingerID);
          displayMessage("Access Granted",matchMessage);
          delay(3000);
          servo.write(0);      // Unlock the door
          displayMessage("Enter the hall","");
          delay(3000);
          closingCountDown();        // Keep door open for 10 seconds
          servo.write(90);            // Closes the door
          lcd.clear();
          initMsg = false;
          delay(1500);          // waits for 3 seconds
        } else {
          displayMessage("No match found","Access Denied");
          servo.write(90);
          delay(1500);
          initMsg = false;
        }
      }
    }
  }
}
```

## OUTPUT SCREENSHOT:

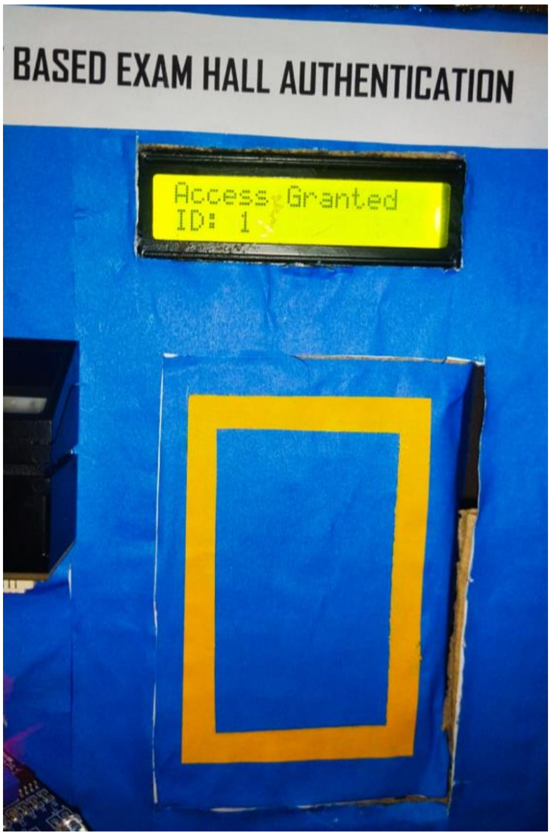
Stand Inside The Mark



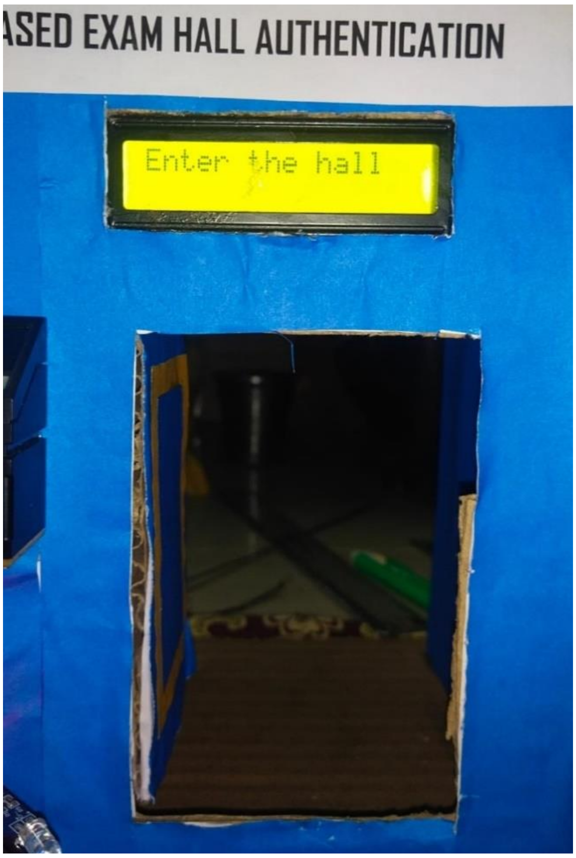
Placing a finger



Access Granting



Entering the exam hall



**Access Denied**

**Door closing time**

## PROTOCOLS USED:

### 1. Communication Protocols

UART (Universal Asynchronous Receiver-Transmitter):
Used for communication between the fingerprint sensor (e.g., R307) and the microcontroller (Arduino).

I2C (Inter-Integrated Circuit):
For communication between components such as the LCD and the microcontroller.

### 2. Control Protocols

PWM (Pulse Width Modulation):
Used to control the servo motor for opening and closing the door.

## LIMITATIONS:

1. **Device Dependency**: The system relies on a fingerprint sensor and an LCD for operation, which might require regular maintenance and calibration.
2. **Power Supply**: Continuous system operation, including the servo motor and sensors, may lead to higher energy consumption.
3. **Limited Access Control**: The system may not accommodate individuals without registered fingerprints, posing challenges in certain scenarios.
4. **Operational Downtime**: If any component, such as the fingerprint sensor or servo motor, malfunctions, the system may face downtime, disrupting the authentication process.
5. **Cost Constraints**: Implementing the system across multiple exam centers could be costly, especially for budget-restricted institutions.

## FUTURE ENHANCEMENTS:

1. **AI Analytics**: Implement advanced AI algorithms to enhance fingerprint matching accuracy and identify potential errors or anomalies during authentication.
2. **Camera Integration**: Incorporate a camera-based system to monitor and prevent unauthorized entry by detecting if more than one person attempts to enter simultaneously.
3. **Cloud Connectivity**: Enable cloud-based integration to provide administrators with real-time monitoring, access logs, and advanced analytics for better management.
4. **Enhanced Security Features**: Add encryption protocols to ensure secure storage and transmission of fingerprint and user data.
5. **Voice Guidance**: Introduce voice prompts for step-by-step guidance, making the system more accessible and user-friendly.
6. **Real-Time Alerts**: Add a notification system to alert administrators of any unauthorized attempts or system malfunctions instantly.

## CONCLUSION:

The fingerprint-based exam hall authentication system offers a modern, efficient, and secure solution to the challenges of candidate verification in examination environments. By leveraging IoT and biometric technology, the system automates the entry process, ensuring that only authorized individuals gain access to the exam hall. This eliminates the risks of impersonation and human error while significantly improving operational efficiency.