

# **Project Report**

**on**

# **Title of Project**

in partial fulfilment for the award of the degree of

**BACHELOR OF ENGINEERING**

**IN**

**B.E CSE(AI-ML)**

**(101-A)**

**Submitted by:**

Aditya Thakur (24BAI70460)

Sahil (24BAI70588)

Aashutosh(24BAI70052)

Aditya (24BAI70517)

**Under the Guidance of**

Dr. Divneet Singh  
**ACADEMIC UNIT-I**



**Chandigarh University**

**April 2025**

## Index

S. No.	Content	Page No.
1	Project Overview	2
2	Objective and Problem Statement	3
3	Proposed Solution & Methodology	4
4	Key Findings / Results	5
5	Conclusion & Learnings	6
6	References	7
7	Appendix (if required)	8

# 1. Project Overview

This project focuses on developing a **Smart Door Lock System** that integrates Arduino, a keypad, LCD, relay, and servo for hardware-level locking mechanisms, and ESP32 for cloud-based control and monitoring via **Ubidots IoT platform**.

The growing need for smart security systems and the rise of IoT-based automation demand more accessible and secure solutions. This project serves as a cost-effective and robust solution for households, offices, or labs where remote access and real-time notification are essential.

## 2. Objective and Problem Statement

### Problem:

Conventional door locks are prone to physical tampering and lack remote monitoring or control capabilities. In today's digital world, there's a need for smarter, IoT-integrated security systems.

### Objectives:

- Create a secure door lock system with dual control via **Keypad** and **Mobile App (Ubidots)**.
- Use **ESP32** for sending notifications and receiving lock/unlock commands via the internet.
- Display real-time status on a **20x4 LCD**.
- Notify users of access status or changes via Ubidots dashboard or app.

## 3. Proposed Solution & Methodology

### Tools and Materials:

- **Arduino Uno**
- **ESP32**
- **20x4 I2C LCD**
- **4x4 Matrix Keypad**
- **Servo Motor (Door Lock)**
- **Relay Module**
- **Ubidots Platform**

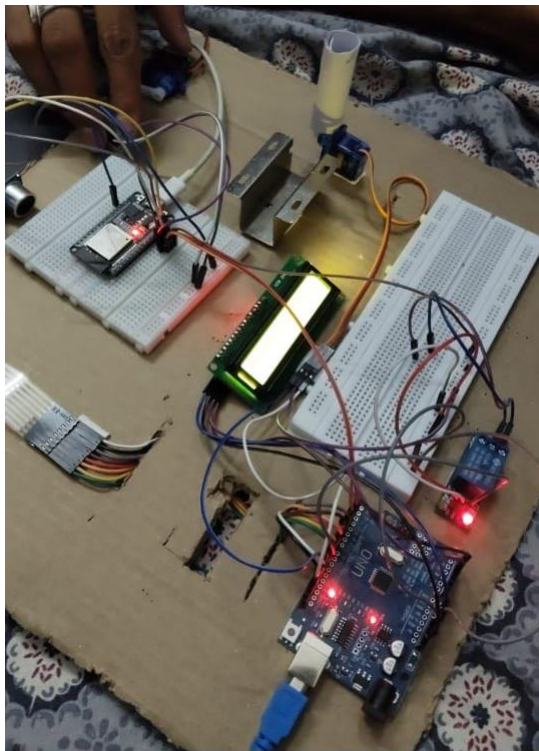
- Jumper Wires, Breadboard, and 5V Power Supply

### Methodology:

1. User inputs password via **Keypad**.
2. If correct, **Arduino** triggers **Servo Motor/Relay** to unlock the door.
3. All actions are mirrored on the **LCD** for feedback.
4. **ESP32** sends status updates to **Ubidots** and listens for lock/unlock commands.
5. Remote access can control the door via the Ubidots dashboard.

## 4. Key Findings / Results

- Successful dual-control system (Keypad + Mobile via Ubidots).
- Real-time lock status display on LCD.
- Servo/Relay mechanism provides secure physical lock control.
- Cloud integration via ESP32 allows remote commands and notification updates.



## 5. Conclusion & Learnings

This project enhanced our understanding of:

- Embedded Systems and IoT integration.
- Real-time cloud communication using **ESP32** and **Ubidots**.
- Designing secure and scalable smart systems.

### Future Scope:

- Add fingerprint or RFID authentication.
- Use voice assistants for lock control.
- Add camera or motion detection features.

## References

1. Ubidots Documentation
2. [ESP32 with Arduino IDE – Programming Tutorial](#)
3. Arduino Project Hub – Door Lock Systems
4. YouTube and online tech forums for troubleshooting

## 7. Appendix

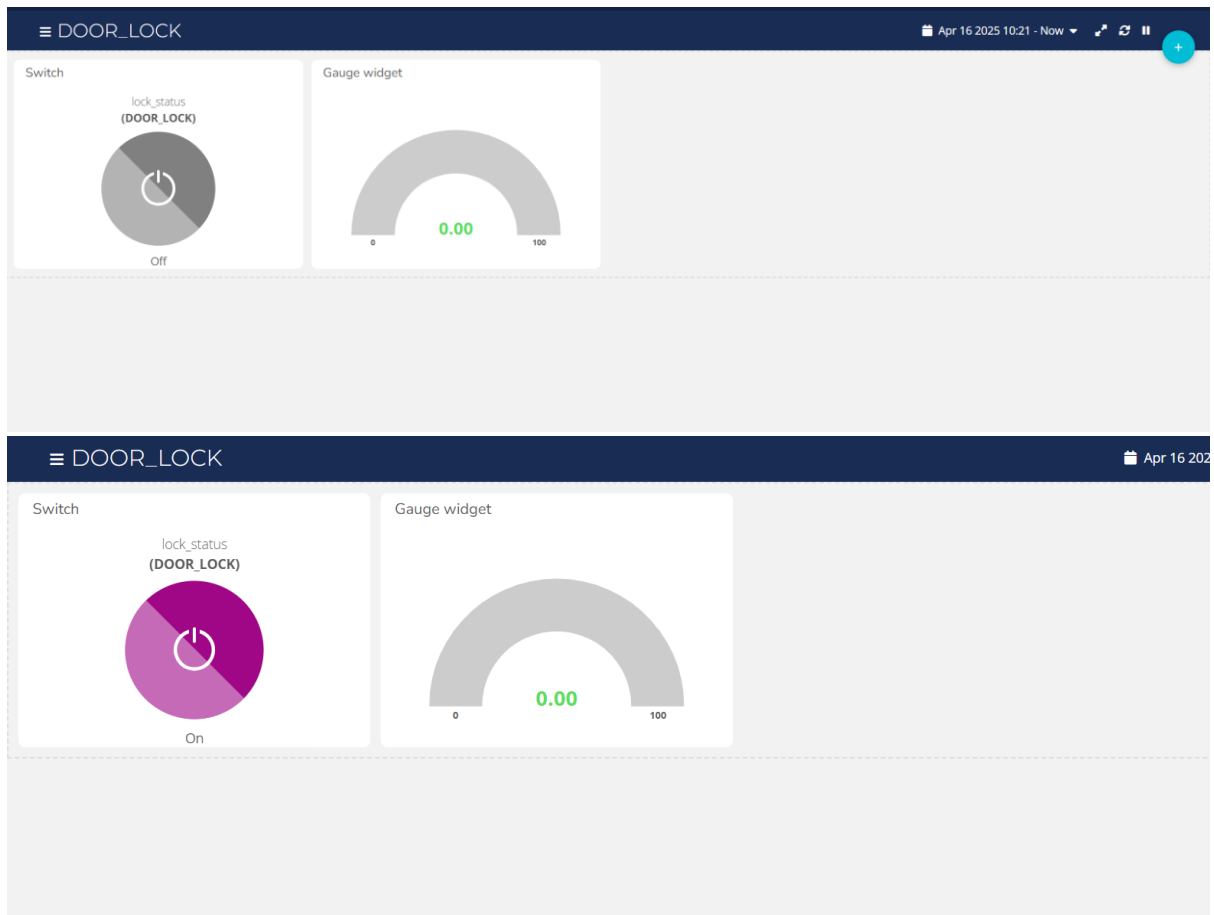
### Arduino + ESP32 Sample Code Snippet:

```
// Sample password verification snippet  
  
if (enteredPassword == "1234") {  
  
    lcd.print("Access Granted");  
  
    digitalWrite(servoPin, HIGH);  
  
    sendStatusToUbidots("unlocked");  
}
```

}

```
1  #include <Wire.h>
2  #include <LiquidCrystal_I2C.h>
3  #include <Keypad.h>
4  #include <Servo.h>
5  #include <SoftwareSerial.h>
6
7  // LCD setup
8  LiquidCrystal_I2C lcd(0x27, 20, 4);
9
10 // Servo
11 Servo lockServo;
12
13 // Keypad setup
14 const byte ROWS = 4;
15 const byte COLS = 4;
16 char keys[ROWS][COLS] = {
17   {'1','2','3','A'},
18   {'4','5','6','B'},
19   {'7','8','9','C'},
20   {'*','0','#','D'}
21 };
22 byte rowPins[ROWS] = {9, 8, 7, 6};
23 byte colPins[COLS] = {5, 4, 13, 12};
24 Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);
25
26 // Servo and relay pins
27 const int servoPin = 11;
28 const int relayPin = 10;
29
30 // Password logic
31 const String correctPassword = "1346";
32 String enteredPassword = "";
33
34 // SoftwareSerial to communicate with ESP32 (ESP32 RX <-- Arduino TX (pin 2), ESP32 TX --> Arduino RX (pin 3))
35 SoftwareSerial espSerial(2, 3); // TX, RX
36
37 void setup() {
38   lcd.init();
39   lcd.backlight();
40   lcd.setCursor(0, 0);
41   lcd.print("Enter Password:");
42
43   lockServo.attach(servoPin);
44   lockServo.write(0); // Lock position
45
46   pinMode(relayPin, OUTPUT);
47   digitalWrite(relayPin, LOW); // Lock initially
48
49   espSerial.begin(9600); // Start communication with ESP32
50   Serial.begin(9600);    // For debugging
51 }
52
53 void loop() {
54   char key = keypad.getKey();
55
56   if (key) {
57     if (key == '#') {
58       if (enteredPassword == correctPassword) {
59         lcd.clear();
60         lcd.setCursor(0, 0);
61         lcd.print("Access Granted");
62
63         lockServo.write(90); // Unlock
64         digitalWrite(relayPin, HIGH);
65
```

```
66     delay(5000);
67
68     lockServo.write(0); // Lock back
69     digitalWrite(relayPin, LOW);
70
71     lcd.clear();
72     lcd.print("Enter Password:");
73   } else {
74     lcd.clear();
75     lcd.setCursor(0, 0);
76     lcd.print("Wrong Password");
77
78     // Notify ESP32 about wrong password attempt
79     espSerial.println("wrong_attempts");
80     Serial.println("wrong_attempts sent to ESP32"); // Debug message
81
82     delay(2000);
83     lcd.clear();
84     lcd.print("Enter Password:");
85   }
86   enteredPassword = "";
87   } else if (key == '*') {
88     enteredPassword = "";
89     lcd.clear();
90     lcd.print("Enter Password:");
91   } else {
92     if (enteredPassword.length() < 16) {
93       enteredPassword += key;
94       lcd.setCursor(0, 1);
95       lcd.print(enteredPassword);
96     }
97   }
98 }
```



- Extra content like code snippets, supplementary material, etc.