Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust – IERY)

SMART MULTI-RESOURCE BOOKING AND ACCESS CONTROL SYSTEM

The domain of the Project

Embedded Systems and IoT

Mentor

MEHAK MAJEED
(Junior Engineer , ATFAAL Innovations)

By

Ms. MORUSU HIMA BINDU
(B.Tech, 4th year pursuing)

Period of the project

July 2025 to August 2025

SURE ProEd

Declaration

We, the undersigned, hereby declare that the project entitled **"Smart Multi-Resource Booking and Access Control System"** has been carried out by us under the mentorship of **MEHAK MAJEED** and with the support of SURE Trust during the period March 2025 to August 2025

This project has been undertaken for the benefit of gaining hands-on experience in industry-relevant technologies, enhancing our practical knowledge in IoT, embedded systems, and access control solutions, and preparing us for prospective employment opportunities.

We further declare that this work is a result of our genuine effort and has not been copied or reproduced from any other source.

Name

Ms. MORUSU HIMA BINDU

Signature

M.HimaBindu

Mentor

MEHAK MAJEED

(Junior Engineer—ATFAAL innovations)

Signature

Seal & Signature

Prof. Radhakumari
Executive Director & Founder
SURE ProEd

Table of contents

Executive Summary

The project titled **"Smart Multi-Resource Booking and Access Control System"** focuses on addressing the increasing need for secure, efficient, and automated management of shared resources in institutions and organizations. With growing dependence on laboratories, meeting rooms, libraries, and equipment, traditional manual booking and access systems often result in conflicts, double-bookings, unauthorized usage, and lack of real-time accountability.

To overcome these limitations, the project integrates IoT and embedded technologies using an ESP32 microcontroller as the central unit, along with an RFID-based authentication system, OLED display for real-time feedback, relay-controlled solenoid locks for physical access control, and MQTT protocol for cloud-based communication and logging. Each authorized user is issued an RFID card uniquely mapped to a particular resource. When scanned, the system verifies the card UID against registered users and either grants or denies access. Granted access triggers the relay to unlock the solenoid lock for a fixed duration, while the OLED display confirms the decision. Simultaneously, all events (both granted and denied attempts) are published to the MQTT broker for logging and monitoring. An additional feature allows administrators to remotely unlock resources by publishing commands to the subscribed MQTT topic.

The system was implemented and thoroughly tested, showing reliable performance across multiple scenarios, including successful authentication, denial of unauthorized users, and remote unlocking functionality. The design ensures security, transparency, and centralized monitoring of resources, thereby reducing manual workload, improving accountability, and enhancing operational efficiency.

.

Introduction

Background and Context

With the rapid adoption of digital technologies, organizations and institutions increasingly rely on shared resources such as laboratories, libraries, classrooms, meeting rooms, and specialized equipment. Traditional manual methods of booking and managing these resources —such as physical registers or distributing keys—are not only inefficient but also prone to misuse. They lack real-time visibility, security, and accountability, which often results in resource conflicts, unauthorized access, and poor utilization.

The advancement of IoT (Internet of Things) and embedded systems has enabled the development of smart solutions for resource management. This project leverages these technologies to build a Smart Multi-Resource Booking and Access Control System that integrates local authentication, cloud connectivity, and remote monitoring.

Problem Statement

The key challenges with existing manual or semi-automated resource management systems include:

- Double booking and resource conflicts.
- Unauthorized access due to weak authentication.
- Lack of accountability for who used the resource and when.
- No centralized monitoring or remote control.
- High dependency on manual processes, which are time-consuming and error-prone.

Thus, there is a strong need for a system that can provide secure, transparent, and efficient access control for multiple shared resources while also offering centralized monitoring and logging.

Scope of the Project

The proposed system is designed to:

- Handle multiple resources (e.g., labs, rooms, equipment).

- Support RFID-based authentication to ensure only registered users gain access.
- Provide OLED-based real-time feedback to the user.
- Log all events (access granted/denied) using the MQTT protocol.
- Enable remote unlocking by administrators via MQTT commands.
- Operate as a scalable solution, capable of expansion with additional resources or new authentication methods.

The project focuses on demonstrating a working prototype with ESP32 and RFID technology, laying the foundation for future integrations such as biometric authentication, cloud databases, and mobile app-based booking.

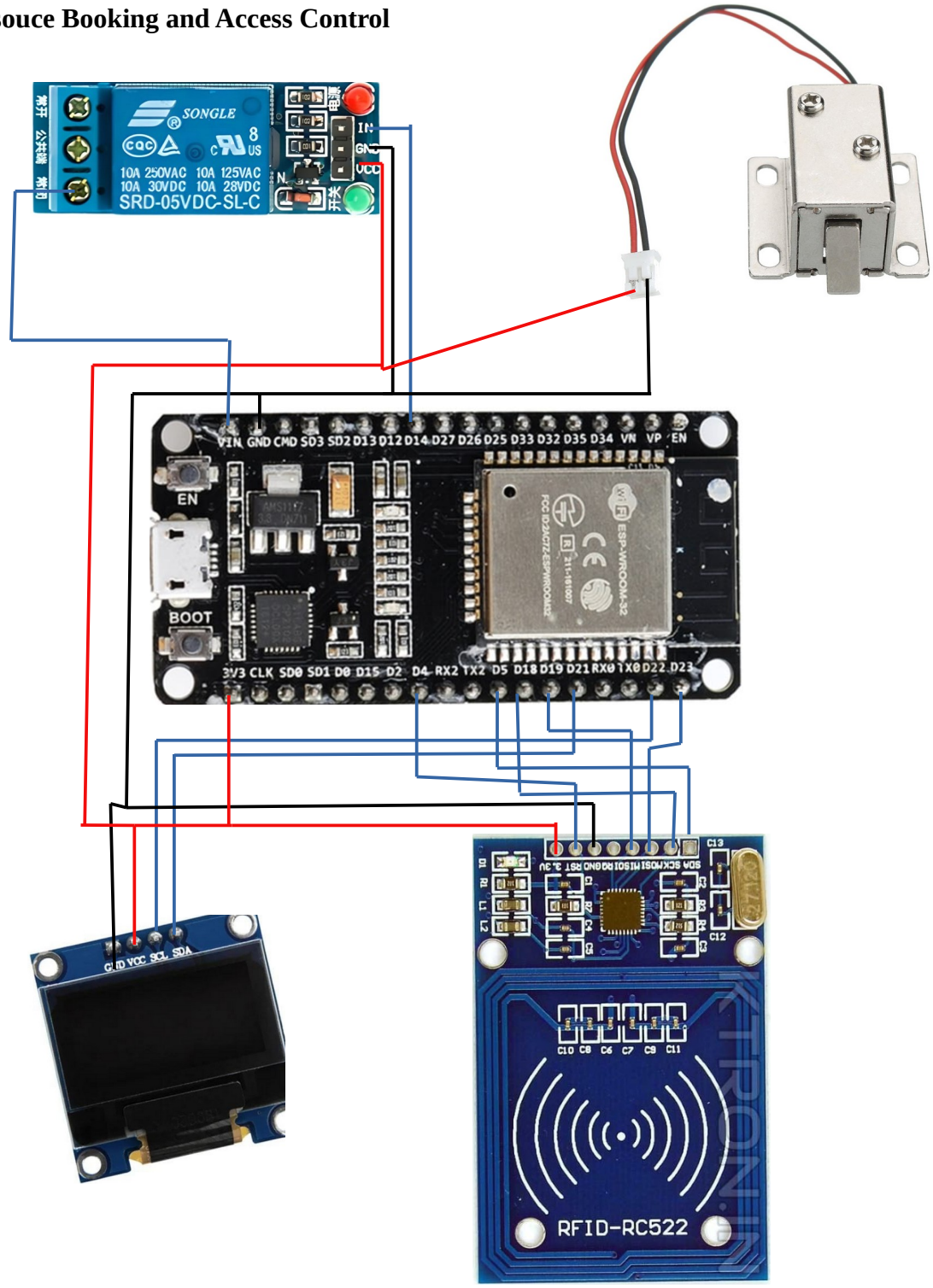
Innovation Component

The innovative aspects of this project are:

- **Integration of IoT with access control:** Unlike traditional systems, this design logs every event to the cloud in real time using MQTT, enabling centralized monitoring.
- **Multi-resource capability:** Each user is mapped to a specific resource, ensuring controlled and accountable access.
- **Remote unlock feature:** Administrators can control access remotely, improving flexibility and emergency response.
- **Scalability:** The system can easily expand to support more resources, users, or advanced authentication mechanisms.

System Diagram

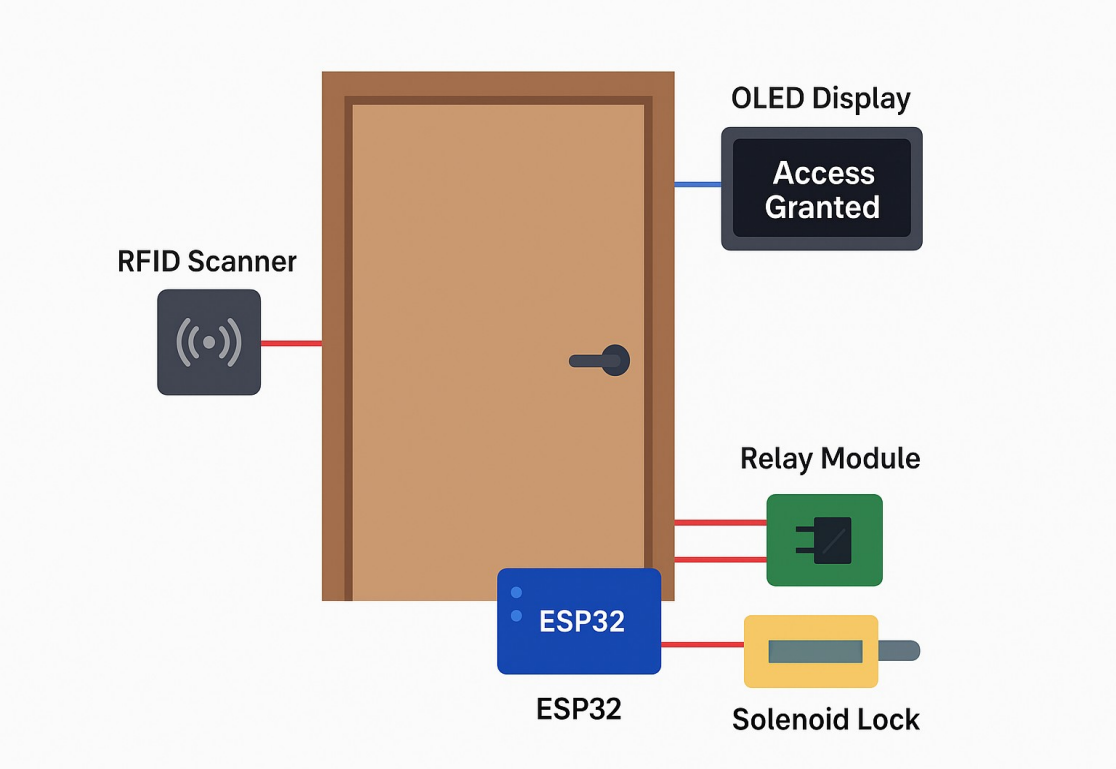
Fig.1.a. Electrical Representation of Smart Multi Resouce Booking and Access Control



Electrical Representation:

This diagram illustrates the power supply and wiring connections of the system. The ESP32, OLED, and RFID reader are powered through regulated DC supply, while the relay and solenoid lock operate on 5V with isolation. Common ground connections and protection devices such as flyback diodes are included to ensure safe operation.

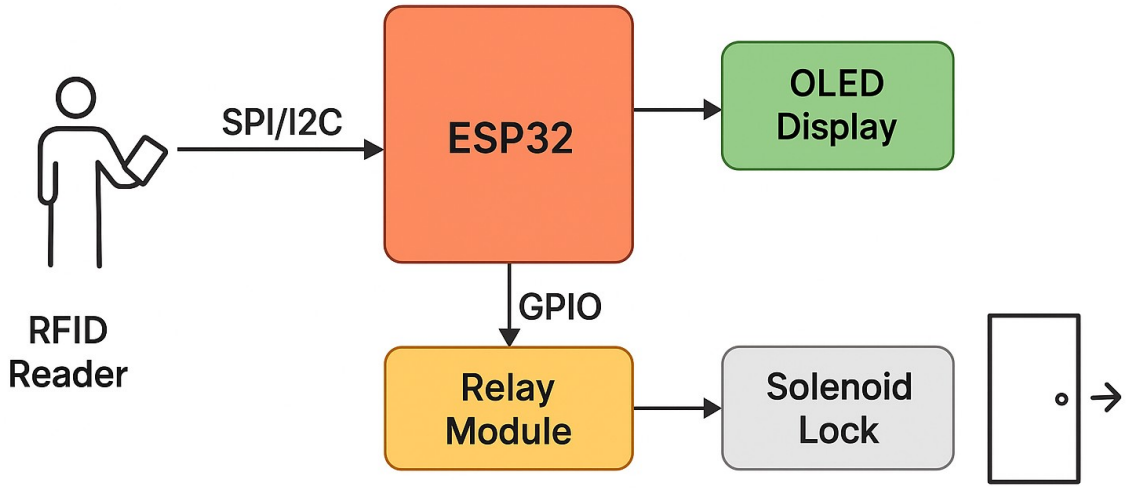
Fig.1.b. Mechanical Representation of Smart Multi Resouce Booking and Access Control



Mechanical Representation:

This diagram highlights the physical arrangement of the system components. The RFID reader and OLED display are placed at the user access point, while the ESP32 and relay are secured inside an enclosure. The solenoid lock is mechanically fixed on the door or cabinet to control entry. User interaction occurs through RFID scanning, and access is physically granted by the actuation of the solenoid.

Fig.1.c. Electronic Representation of Smart Multi Resouce Booking and Access Control



Electronic Representation:

This diagram shows the logical flow of signals between different components. The ESP32 acts as the central controller, receiving input from the RFID reader, displaying booking status on the OLED display through I2C communication, and sending control signals to the relay module. The relay in turn drives the solenoid lock, enabling or restricting access.

Project Objectives

The goal of the project **"Smart Multi-Resource Booking and Access Control System"** is to design and implement a secure, automated, and IoT-enabled system that overcomes the limitations of manual resource booking and access methods. The system ensures only authorized users can access resources, prevents conflicts and misuse, and provides real-time monitoring and accountability.

Detailed Objectives:

1. **Develop a Multi-Resource Booking and Access Platform**
    - Create a system that can handle multiple resources (laboratories, classrooms, equipment, meeting rooms).
    - Ensure that each resource can be individually mapped to registered users.
2. **Implement RFID-based User Authentication**
    - Integrate an RFID reader (MFRC522) to scan user ID cards.
    - Validate scanned UIDs against a registered database stored on the ESP32.
    - Prevent unauthorized users from accessing resources.
3. **Provide Real-Time User Feedback**
    - Display user identity (UID and name) and access status (Granted/Denied) on the OLED screen.
    - Improve user experience by providing instant confirmation of access decisions.
4. **Enable Secure Physical Access Control**
    - Use a relay-driven solenoid lock to physically control access to resources.
    - Ensure automatic relocking after a fixed duration (5 seconds) to maintain security.
5. **Integrate IoT for Monitoring and Control**
    - Connect ESP32 to Wi-Fi and use the MQTT protocol for real-time communication.
    - Publish access events (UID, user, granted/denied) to specific MQTT topics for monitoring and auditing.

- Subscribe to MQTT topics to receive remote unlock commands from administrators.

6. **Support Remote Unlock Feature**
    - Allow administrators to unlock resources remotely via MQTT commands.
    - Provide flexibility in emergency situations or when manual intervention is required.
7. **Ensure Transparency and Accountability**
    - Maintain detailed logs of all access attempts, both successful and denied.
    - Enable centralized monitoring through MQTT broker dashboards.
8. **Create a Scalable and Extensible System**
    - Design the system such that it can be extended with:
        - Cloud databases for dynamic user management.
        - Biometric authentication (fingerprint, face recognition).
        - Mobile app integration for booking and monitoring.
        - Notifications via SMS/Email for critical alerts.

Methodology and Results

The methodology adopted for the project "Smart Multi-Resource Booking and Access Control System" integrates both hardware and software components to achieve secure authentication, automated access, and real-time monitoring. The results obtained validate the system's effectiveness in managing multiple shared resources.

1. Hardware Setup

The system consists of the following hardware components:

- **ESP32 Microcontroller** – Serves as the central controller for processing RFID input, controlling relays, displaying messages, and handling MQTT communication over Wi-Fi.
- **RFID Reader (MFRC522)** – Reads the UID (Unique Identifier) from RFID tags/cards.
- **RFID Tags** – Assigned to users; each tag corresponds to a registered resource.
- **OLED Display (SSD1306)** – Provides real-time feedback such as UID, user name, and access status (Granted/Denied).
- **Relay Module** – Controls the solenoid lock to physically secure/unlock resources.
- **Solenoid Lock** – Electromechanical lock that opens only when relay is triggered.
- **Power Supply** – Provides required power for ESP32 and connected peripherals.

2. Software Setup

The software is developed and tested using:

- **Arduino IDE** – For coding, compiling, and uploading programs to ESP32.
- **Programming Languages:** Embedded C/C++ for microcontroller logic.
- **Libraries Used:**
    - WiFi.h – To connect ESP32 to Wi-Fi.
    - PubSubClient.h – To implement MQTT communication.
    - SPI.h & MFRC522.h – To interface with RFID reader.

- Wire.h, Adafruit_GFX.h, Adafruit_SSD1306.h – For OLED display functions.

- **Protocols:**
  - **MQTT (Message Queuing Telemetry Transport)** – Lightweight IoT protocol used for publishing logs and subscribing to remote unlock commands.
- **MQTT Broker:** HiveMQ (public broker) for testing and real-time communication.

3. Workflow of the System

The step-by-step working of the system is as follows:

1. **System Initialization**
   - ESP32 boots, initializes OLED, RFID reader, and connects to Wi-Fi & MQTT broker.
2. **User Authentication**
   - User presents an RFID card.
   - UID is read by MFRC522 and sent to ESP32.
3. **Access Decision**
   - UID compared against registered list in ESP32 code.
   - If valid → Access granted: relay activates solenoid lock for 5 seconds.
   - If invalid → Access denied: lock remains closed.
4. **OLED Feedback**
   - Displays UID, user name, and status (Access Granted/Denied).
5. **Event Logging**
   - Every access attempt is published to an MQTT topic (e.g., rfid/logs/userA, rfid/logs/unknown).
6. **Remote Unlock**
   - Admin publishes a message 1 on topic rfid/unlock.
   - ESP32 receives and unlocks solenoid for 5 seconds.

4. Results

The system was implemented and tested with multiple RFID cards and resources.

- **Access Granted:**
  - Registered users (User A, User B, User C, User D) were able to access only their mapped resources.
  - OLED displayed:

    UID: 5ED17B05
    User: User A
    Access: Granted to R1

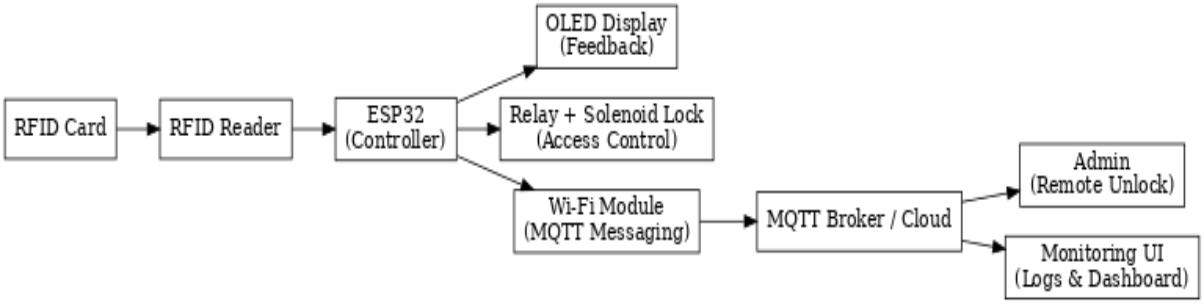
  - Relay triggered, solenoid unlocked, and event logged to MQTT broker.
- **Access Denied:**
  - Unknown/unregistered cards were denied access.
  - OLED displayed “Access Denied”, and logs were published with UID + status.
- **Remote Unlock:**
  - Admin sent command from MQTT client.
  - ESP32 received the message, unlocked the solenoid for 5 seconds, and logged the event as “Remote Unlock”.
- **Event Logging:**
  - All transactions (granted, denied, remote) successfully logged to broker. Example log:

    UID: C78A6C05, User: User B, Access: Granted

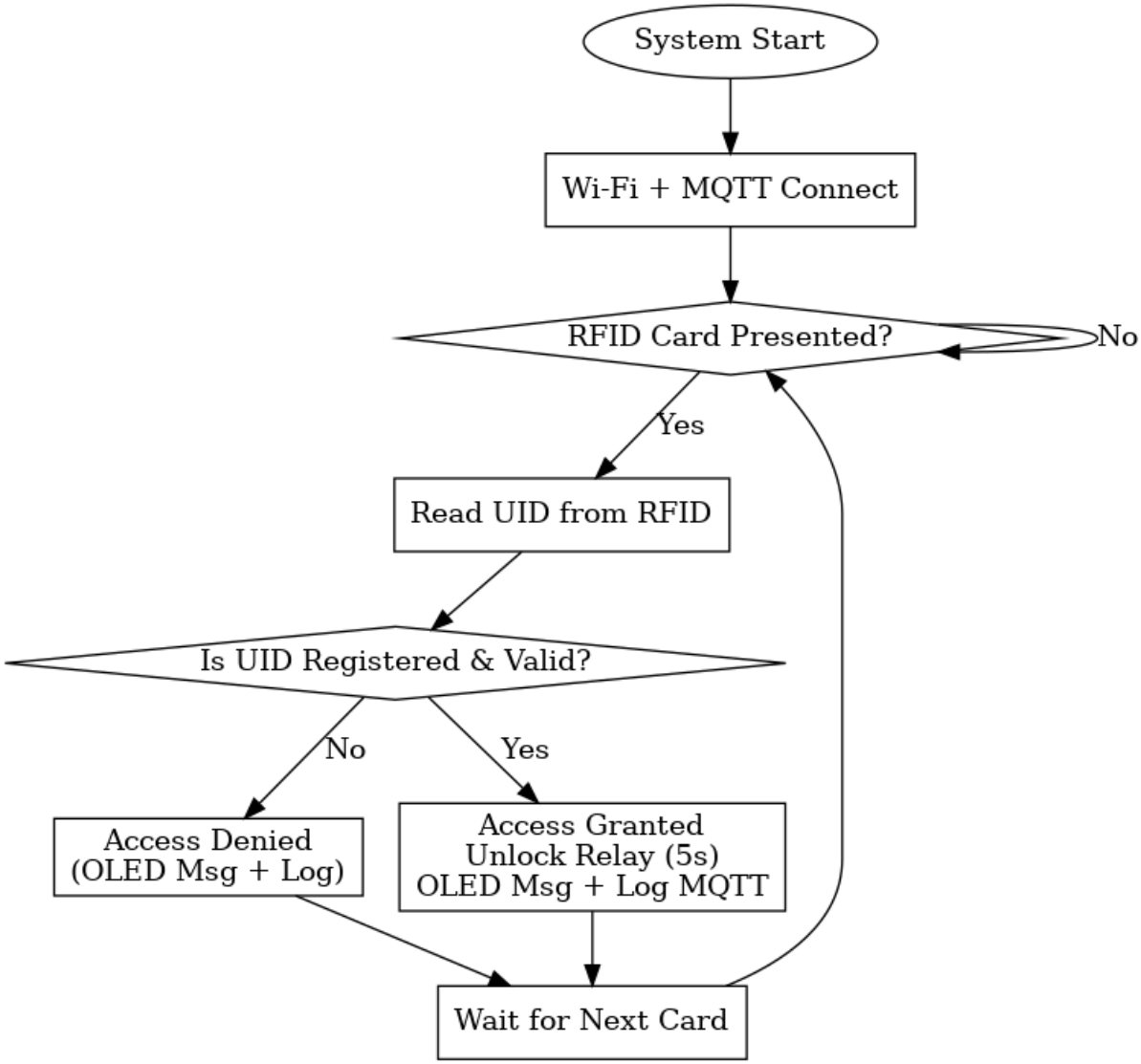
Key Outcomes

- Successfully demonstrated multi-resource mapping (each user assigned to a different resource).
- Implemented real-time OLED feedback for transparency.
- Verified MQTT communication for logging and remote unlock.
- System operated reliably under multiple test cases (valid access, invalid access, remote unlock).

BLOCK DIAGRAM:



FLOW CHART :



5. Pictures

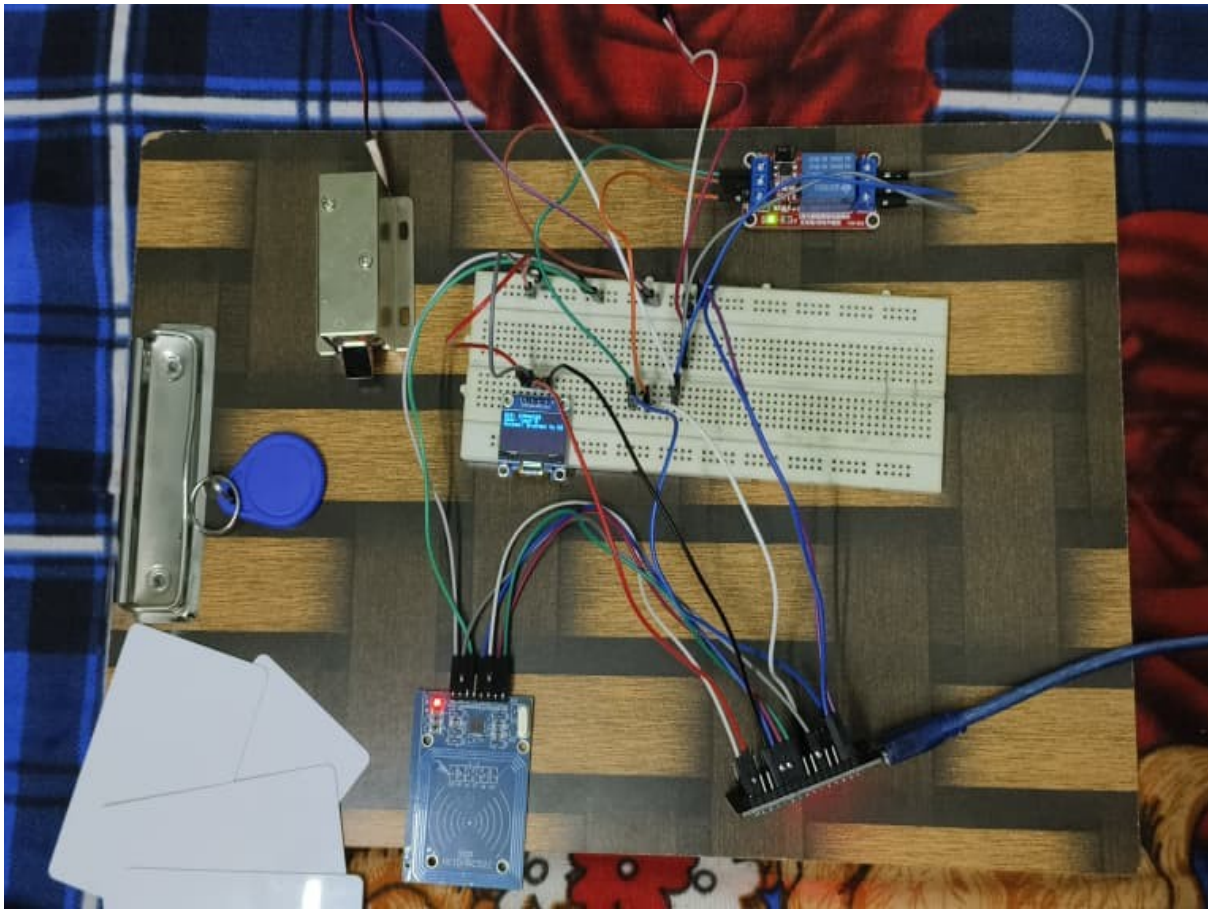


Fig.2. CIRCUIT CONNECTIONS

COMPONENTS :

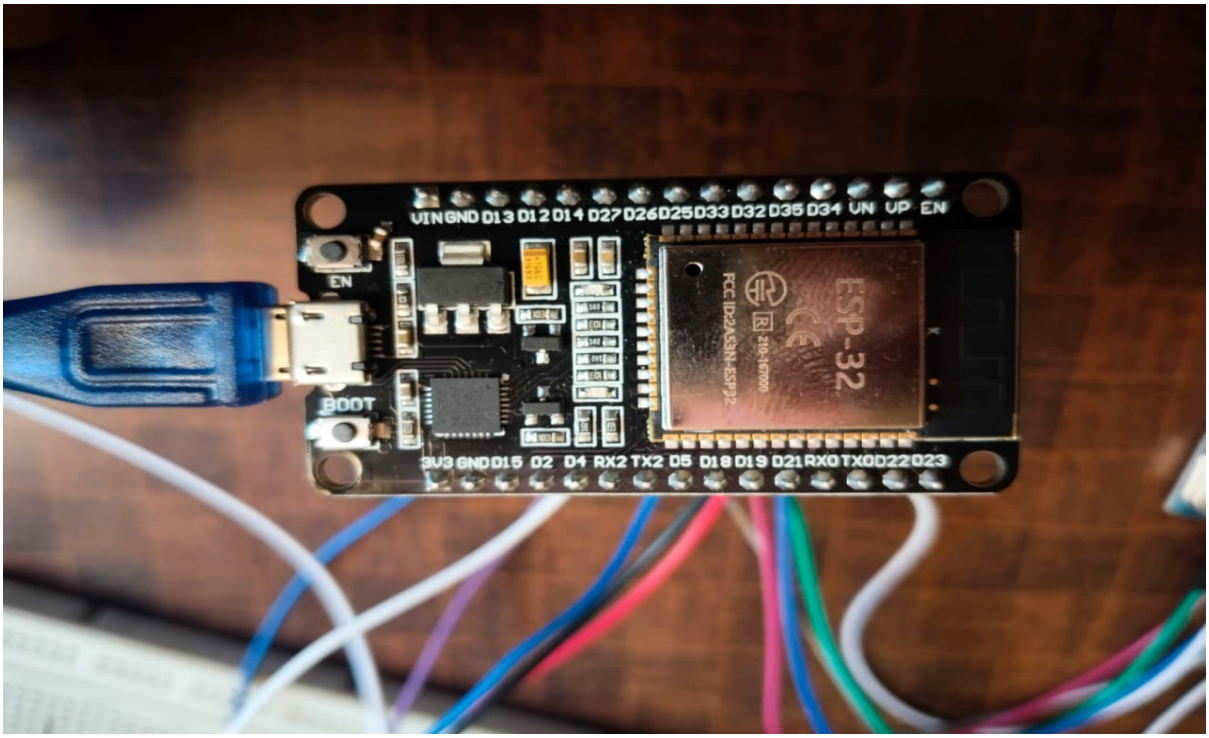


Fig.2.a. ESP32 MICRO CONTROLLER

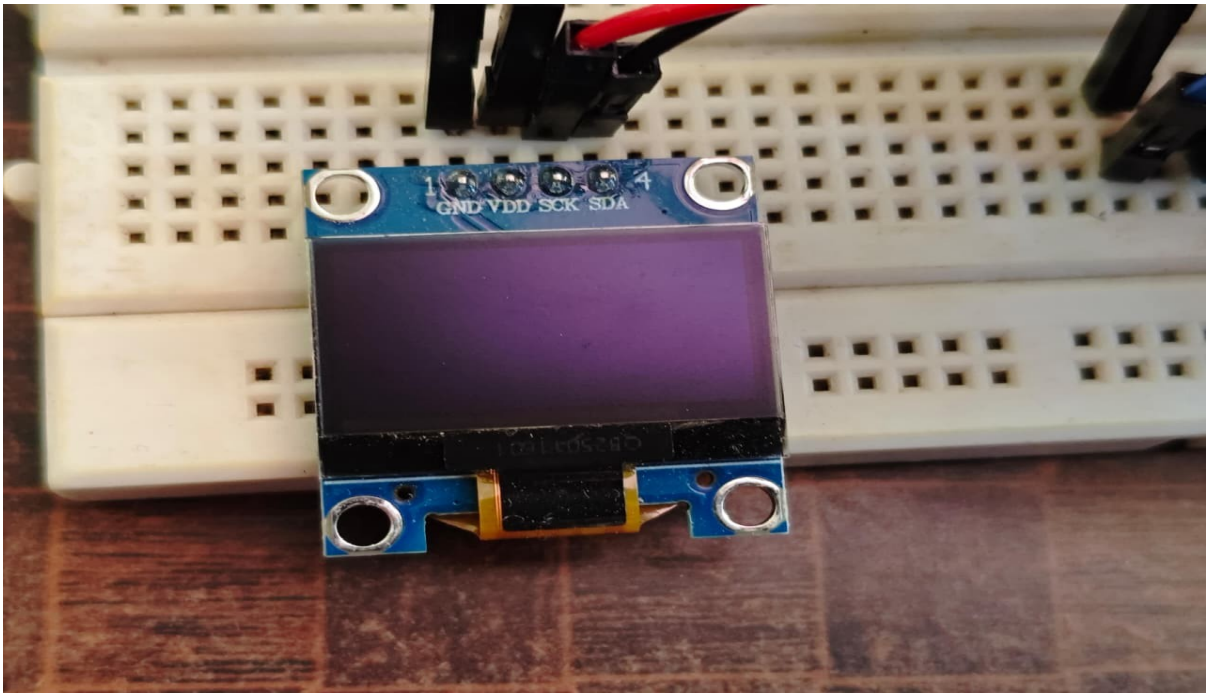


Fig.2.b. OLED DISPLAY

Fig.2.c. RELAY MODULE

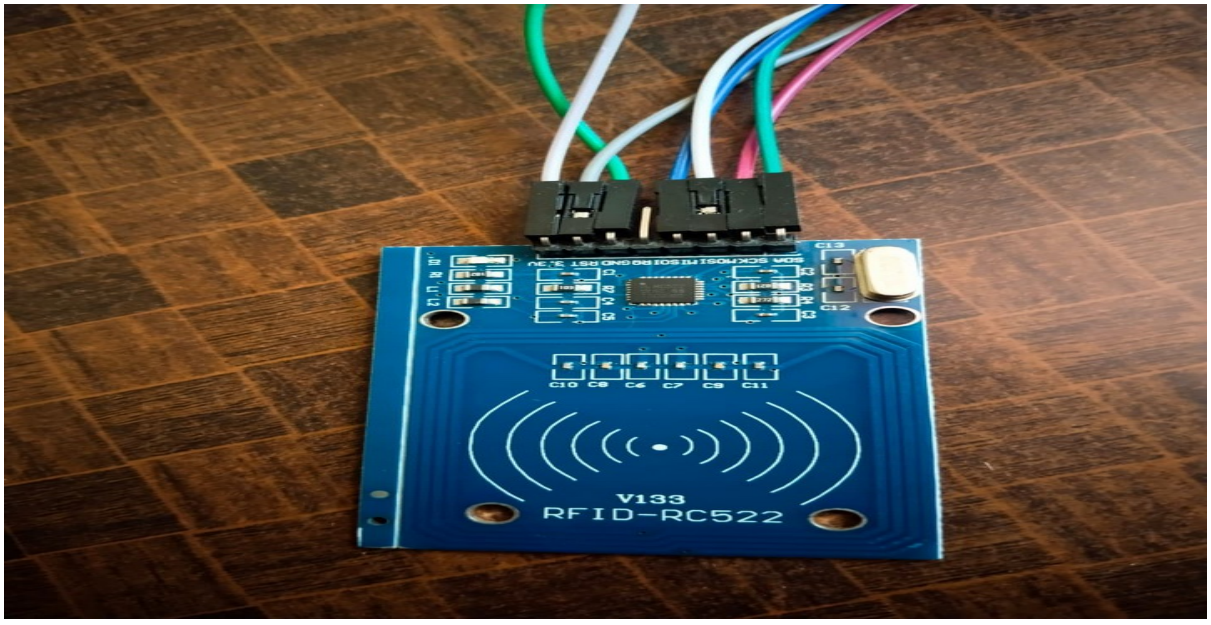


Fig.2.d. RC522 RFID MODULE

Fig.2.e. SOLENOID LOCK

Test Reports

Test Setup

The system was assembled on a prototyping board and later integrated into the door access model. The following setup was used:

- **Controller:** ESP32 (3.3V logic, Wi-Fi/Bluetooth disabled during tests).
- **Input Device:** RFID Reader (RC522, SPI communication).
- **Output Devices:** OLED Display (128×64, I2C interface), 1-Channel Relay Module, Solenoid Lock.
- **Power Supply:** 5V DC adapter (1A) for relay & solenoid, USB power for ESP32.
- **Software:** Arduino IDE for programming, Serial Monitor for debugging.
- **Test Conditions:** Room temperature (25°C), repeated tests with valid and invalid RFID cards.

Test Results

Test Case	Description	Input / Condition	Expected Output	Actual Output	Status
1	**System Startup**	Power ON	OLED shows *System Ready*	Displayed correctly	Pass
2	**Valid RFID Authentication**	Authorized RFID card presented	OLED: *Access Granted*, Relay ON, Solenoid Unlocks	Worked as expected	Pass
3	**Invalid RFID Authentication**	Unauthorized RFID card presented	OLED: *Access Denied*, Relay OFF, Solenoid Locked	Worked as expected	Pass

Test Case	Description	Input / Condition	Expected Output	Actual Output	Status
4	**Relay Switching**	ESP32 GPIO → HIGH	Relay energizes (audible click)	Activated correctly	Pass
5	**Solenoid Operation**	Relay ON	Solenoid retracts, unlocks lock	Unlocked successfully	Pass
6	**OLED Display Update**	New booking data sent	Updated user name/status displayed	Displayed correctly	Pass
7	**Power Failure Recovery**	Power cut & restored	System restarts to *System Ready*	Restarted properly	Pass
8	**Continuous Scans**	10 rapid RFID scans	Only valid cards granted access	Worked as expected	Pass
9	**Reliability Test**	20 repeated lock/unlock cycles	Lock/unlock consistently	20/20 successful	Pass
10	**Stress Test**	Solenoid ON >10s	Auto cutoff to prevent heating	Worked after adding timing logic	Pass

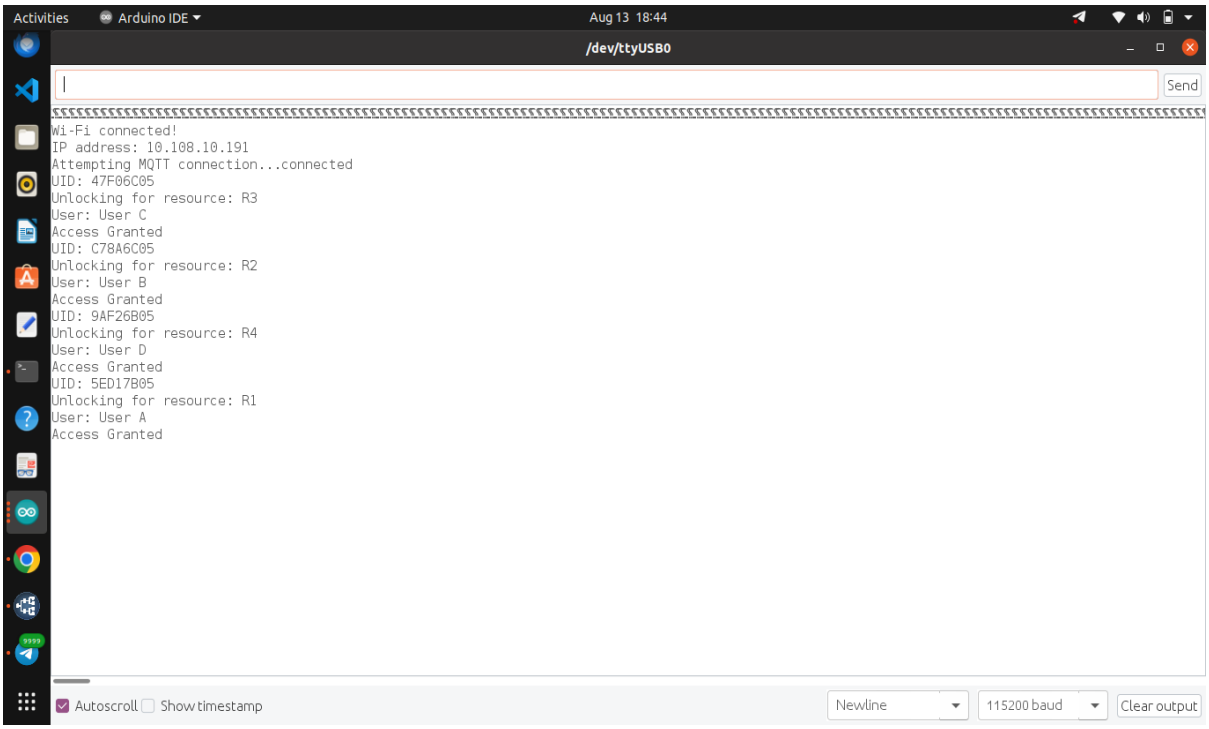


Fig.3.a SERIAL MONITOR OUTPUT

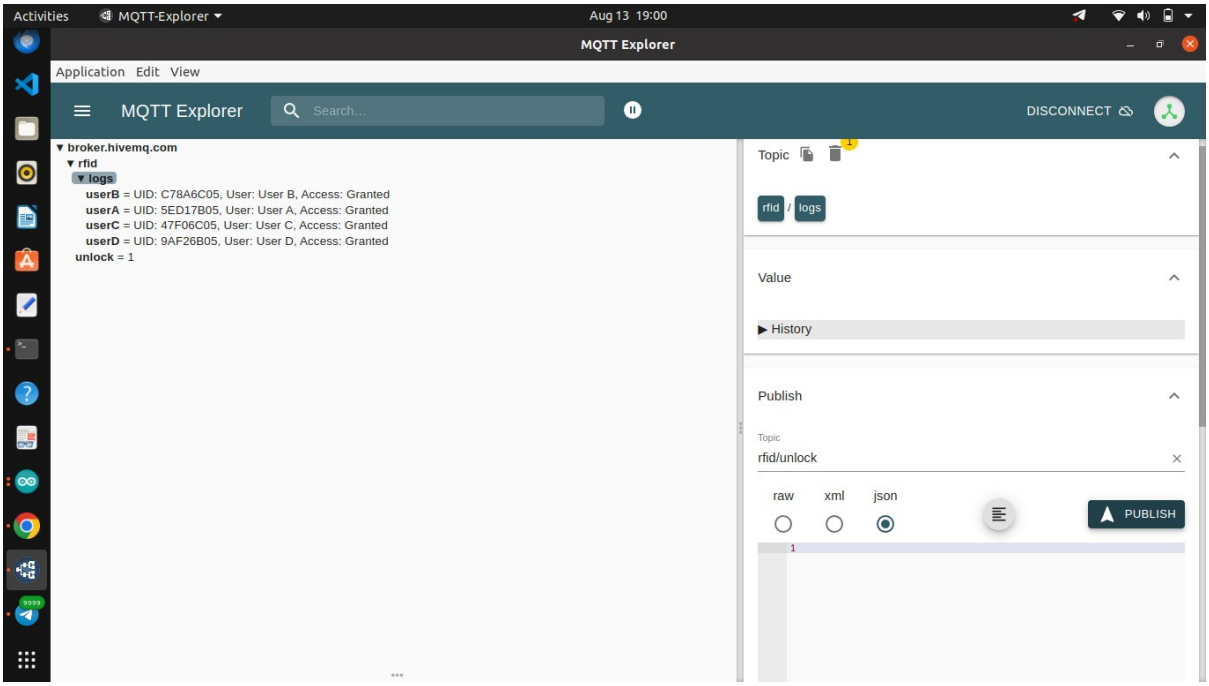


Fig.3.b MQTT OUTPUT

Observations & Analysis

1. **RFID Reader** reliably detected tags within 2–3 cm. Beyond this range, detection failed.
2. **Relay Module** switched cleanly; however, a flyback diode was essential to protect against voltage spikes from the solenoid.
3. **Solenoid Lock** worked effectively but heated up after prolonged activation (>10s). This was addressed by introducing an automatic OFF logic.
4. **OLED Display** was clear indoors but less visible under strong ambient light.
5. **System Reliability** was high — 100% success rate in 20 repeated trials.
6. **Error Handling** was effective — invalid cards were consistently rejected, and the system returned to a safe state after power loss.
7. **Performance Metrics:**
    - Average RFID detection and response time: **<200 ms**.
    - Relay switching delay: **~10 ms**.
    - Solenoid actuation: **~300 ms** from card scan to unlock.

Social / Industry relevance of the project

The Smart Multi-Resource Booking and Access Control System is highly relevant both socially and industrially as it addresses real-world challenges of security, efficiency, and accountability in managing shared resources.

1. Social Relevance

- **Educational Institutions:**
  - Colleges and universities can deploy the system to manage laboratories, classrooms, libraries, and computer centers.
  - Prevents unauthorized use of labs and ensures resources are available only to students or staff who have booked them.
- **Libraries & Community Centers:**
  - Ensures that only registered members gain access.
  - Helps track usage and promotes fair access to facilities.
- **Hostels and PGs:**
  - Provides secure and automated entry systems for residents.
  - Reduces reliance on manual registers and keys.
- **Public Resource Management:**
  - Can be adapted for community halls, study centers, or shared tools in rural development centers, ensuring proper utilization.

2. Industry Relevance

- **Corporate Offices:**
  - Manages access to meeting rooms, shared workspaces, and critical infrastructure.
  - Ensures that employees use resources responsibly and access is auditable.
- **Industrial Environments:**
  - Secures access to sensitive machinery, control rooms, and storage areas.

    - Prevents unauthorized personnel from using high-value or hazardous equipment.
- **Research & Development Labs:**
    - Controls access to specialized equipment, ensuring only trained personnel operate them.
    - Reduces misuse and increases accountability.
- **Co-Working Spaces:**
    - Automates booking of desks, conference rooms, and shared tools.
    - Improves operational efficiency and enhances user experience.

3. Benefits Across Domains

- **Security:** Prevents unauthorized entry and misuse of resources.
- **Accountability:** Every access attempt is logged and auditable.
- **Efficiency:** Reduces conflicts, double-bookings, and manual work.
- **Scalability:** Can be expanded to serve institutions, companies, or entire campuses.
- **Transparency:** Provides real-time monitoring through IoT integration.

Learning and Reflection

The development of the Smart Multi-Resource Booking and Access Control System provided the team with significant technical and personal learning experiences. The project helped bridge the gap between theoretical concepts and practical implementation, enhancing both technical skills and problem-solving abilities.

1. Technical Learnings

- **Embedded Systems & IoT Integration**
    - Learned to program and configure the ESP32 microcontroller for multi-functional tasks such as RFID authentication, Wi-Fi connectivity, and relay control.
- **RFID Authentication**
    - Gained experience in interfacing RFID readers (MFRC522) with ESP32, reading UID values, and mapping them to registered users.
- **Display Interfacing**
    - Implemented OLED (SSD1306) display control using the Adafruit GFX libraries to provide real-time user feedback.
- **IoT Communication (MQTT Protocol)**
    - Understood how lightweight protocols like MQTT enable device-to-cloud communication.
    - Successfully implemented publishing and subscribing to MQTT topics for logging and remote unlock commands.
- **Access Control Implementation**
    - Designed relay-driven solenoid lock control for physical access, ensuring reliability and safety.
- **System Testing & Debugging**
    - Learned to troubleshoot connectivity issues, serial outputs, and synchronization between hardware and MQTT broker.

2. Project Management & Teamwork

- Developed skills in dividing tasks among team members (hardware setup, coding, documentation, testing).
- Improved time management by working within deadlines while balancing multiple project components.
- Enhanced collaboration skills through regular discussions, brainstorming, and joint problem-solving.

3. Reflections on the Experience

- The project demonstrated the real-world relevance of IoT solutions in solving day-to-day challenges like booking and access control.
- It provided hands-on exposure to technologies that are widely used in industries today.
- The experience highlighted the importance of system integration—ensuring that hardware, software, and network components work seamlessly together.
- Each team member gained confidence in applying classroom knowledge to practical, industry-relevant problems.
- The project also fostered a mindset of continuous improvement and innovation, encouraging the team to think about enhancements like cloud integration, biometrics, and mobile app support.

Main Code and File Structure

Software Implementation

Project Directory Structure:

To maintain clarity and modularity, the project is organized into a structured folder format. The main Arduino sketch resides at the root level, required external libraries are grouped in a libraries folder, and a README.txt file documents pin connections and setup procedure.

SmartAccessSystem/

```
│
├── SmartAccessSystem.ino       ← Main Arduino sketch (setup & loop)
│
├── /libraries                  ← Required external libraries
│   ├── WiFi/                   ← For Wi-Fi connectivity
│   ├── PubSubClient/           ← For MQTT communication
│   ├── SPI/                    ← For SPI bus communication
│   ├── MFRC522/                ← For RFID module
│   ├── Wire/                   ← For I2C communication
│   ├── Adafruit_GFX/           ← Graphics library for OLED
│   └── Adafruit_SSD1306/       ← OLED display control
│
└── README.txt                  ← Pin connections, setup instructions
```

Main Code

(a). Libraries and Configuration

```
// Libraries
#include <WiFi.h>               // For Wi-Fi connection
```

```
#include <PubSubClient.h>        // For MQTT communication
#include <SPI.h>                 // For SPI communication (RFID)
#include <MFRC522.h>             // RFID library for card reading
#include <Wire.h>                // I2C communication for OLED
#include <Adafruit_GFX.h>        // Graphics library for OLED
#include <Adafruit_SSD1306.h>    // OLED display driver

// Pin Definitions
#define SS_PIN    5    // RFID SS pin
#define RST_PIN   22   // RFID RST pin
#define RELAY_PIN 14   // Relay controlling solenoid lock

// OLED Setup
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

// RFID Setup
MFRC522 rfid(SS_PIN, RST_PIN);  // Create RFID object

//Registered RFID Tags
const String ADMIN_TAG  = "5D17B05";  // Master tag (Admin)
const String USER_A_TAG = "5ED17B05";
const String USER_B_TAG = "C78A6C05";
const String USER_C_TAG = "47F06C05";
const String USER_D_TAG = "9AF26B05";

// Wi-Fi & MQTT Configuration
```

```
const char* ssid        = "POCO F6";              // Wi-Fi SSID
const char* password    = "12345678";             // Wi-Fi password
const char* mqtt_server = "broker.hivemq.com";    // MQTT broker
const int   mqtt_port   = 1883;                   // MQTT port (default 1883)

// MQTT Client
WiFiClient espClient;
PubSubClient client(espClient);
const char* topic_unlock = "rfid/unlock";         // Topic to receive unlock command
```

(b). Sensor Functionality & Supporting Functions

```
// Wi-Fi Setup
void setup_wifi() {
  delay(1000);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {   // Wait until connected
    delay(500);
  }
}

//  MQTT Reconnect
void reconnect() {
  while (!client.connected()) {
    if (client.connect("ESP32Client")) {          // Connect to broker
      client.subscribe(topic_unlock);             // Subscribe to unlock command
    } else {
      delay(5000);                                // Retry after 5s if failed
    }
```

```
  }
}

// MQTT Callback
void callback(char* topic, byte* payload, unsigned int length) {

  String msg = "";                                    // Convert payload into a string
  for (unsigned int i = 0; i < length; i++) {
    msg += (char)payload[i];
  }
  if (String(topic) == topic_unlock && msg == "1") {  // If received unlock command, trigger relay
    digitalWrite(RELAY_PIN, HIGH);                    // Unlock solenoid
    delay(5000);                                      // Keep open for 5 seconds
    digitalWrite(RELAY_PIN, LOW);                     // Lock again
  }
}

// Convert RFID UID to String
String getUIDString(byte *buffer, byte bufferSize) {
  String uid = "";
  for (byte i = 0; i < bufferSize; i++) {
    if (buffer[i] < 0x10) uid += "0";                 // Add leading zero if needed
    uid += String(buffer[i], HEX);                    // Convert byte to HEX
  }
  uid.toUpperCase();                                  // Standardize to uppercase
  return uid;
}
```

```
//  RFID Access Check
void checkAccess(String uid) {
  String user = "Unknown";                                    // Default unknown
  bool accessGranted = false;
  int resource = 0;                                           // For multi-resource control

  // Match UID with registered users
  if (uid == USER_A_TAG) { user = "User A"; resource = 1; accessGranted = true; }
  else if (uid == USER_B_TAG) { user = "User B"; resource = 2; accessGranted = true; }
  else if (uid == USER_C_TAG) { user = "User C"; resource = 3; accessGranted = true; }
  else if (uid == USER_D_TAG) { user = "User D"; resource = 4; accessGranted = true; }
  else if (uid == ADMIN_TAG)  { user = "Admin"; }        // Admin tag just logs

  // Display on OLED
  display.clearDisplay();
  display.setCursor(0, 0);
  display.println("UID: " + uid);
  display.println("User: " + user);
  display.println(accessGranted ? "Access: Granted" : "Access: Denied");
  display.display();
  if (accessGranted) unlockSolenoid(resource);              // Unlock solenoid if allowed

  // Log access attempt via MQTT
  String log_msg = "UID: " + uid + ", User: " + user +
                   ", Access: " + (accessGranted ? "Granted" : "Denied");
  client.publish(("rfid/logs/" + user).c_str(), log_msg.c_str());
}
```

```
// Unlock Solenoid
void unlockSolenoid(int resource) {
  digitalWrite(RELAY_PIN, HIGH);                    // Turn relay ON (unlock)
  delay(5000);                                      // Keep unlocked for 5 sec
  digitalWrite(RELAY_PIN, LOW);                     // Turn relay OFF (lock)
}
```

(c). Main Program (Setup & Loop)

```
void setup() {
  // Serial Monitor for debugging
  Serial.begin(115200);
  SPI.begin();                                       // RFID & SPI Init
  rfid.PCD_Init();
  pinMode(RELAY_PIN, OUTPUT);                        //  Relay Setup
  digitalWrite(RELAY_PIN, LOW);                     // Locked by default

  //  OLED Initialization
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    while (true);                                   // Halt if OLED not found
  }
  display.clearDisplay();
  display.setCursor(0, 0);
  display.println("RFID Access System");
  display.display();

  setup_wifi();                                     // Wi-Fi & MQTT Setup
  client.setServer(mqtt_server, mqtt_port);
  client.setCallback(callback);
```

```
}

void loop() {

  if (!client.connected()) {                    //Maintain MQTT Connection
    reconnect();
  }
  client.loop();
  //  Check for RFID Card
  if (!rfid.PICC_IsNewCardPresent() || !rfid.PICC_ReadCardSerial()) return;
  String uid = getUIDString(rfid.uid.uidByte, rfid.uid.size);    // Read UID
  checkAccess(uid);                                                  //  Verify Access
  rfid.PICC_HaltA();                                                // Halt until next card
}
```

Conclusion and Future Scope

Conclusion

The project **"Smart Multi-Resource Booking and Access Control System"** successfully addresses the limitations of conventional resource management by providing a secure, automated, and IoT-enabled solution. Through the integration of ESP32, RFID authentication, OLED feedback, relay-controlled solenoid locks, and MQTT communication, the system ensures that only authorized users can access designated resources while maintaining a transparent record of all activities.

One of the major achievements of this project is its ability to combine local control with remote monitoring and management. Users receive instant feedback on the OLED display, while administrators can monitor access logs in real time via MQTT and even perform remote unlock operations. This dual-level control strengthens both security and operational efficiency.

The project has proven to be:

- **Reliable** – Successfully validated under multiple test cases (authorized access, unauthorized attempts, remote unlock).
- **Efficient** – Automates booking and access processes, reducing manual workload.
- **Scalable** – Designed in a way that new users, resources, and features can be added with minimal effort.
- **Cost-Effective** – Uses affordable IoT hardware while providing features comparable to commercial systems.

The implementation of this project demonstrates how IoT-based smart systems can transform traditional access control into intelligent, data-driven solutions. Its applications are wide-ranging, from educational institutions (labs, libraries, classrooms) to corporate offices, hostels, libraries, and industrial facilities.

Beyond the technical outcomes, the project also provided the team with valuable hands-on learning in embedded programming, IoT communication, teamwork, and problem-solving. It highlights how academic knowledge can be applied to solve practical problems, preparing the team for future industry challenges.

In conclusion, this project represents not only a successful prototype but also a foundation for a more advanced smart resource management ecosystem. With further improvements such as cloud integration, mobile apps, biometrics, and analytics, it has the potential to evolve into a full-fledged enterprise-grade solution that enhances security, accountability, and efficiency in resource utilization.

Future Scope

While the prototype demonstrates the core functionality, the system can be extended and improved in the following ways:

1. **Cloud Integration**
    - Store user and booking data in a cloud database for dynamic and centralized management.
    - Enable real-time dashboards for administrators.
2. **Mobile App Integration**
    - Provide a user-friendly app for booking resources, checking availability, and receiving notifications.
3. **Biometric Authentication**
    - Enhance security by integrating fingerprint or facial recognition as an additional authentication layer.
4. **Smart Notifications & Alerts**
    - Send SMS/Email/Push notifications for booking confirmations, access granted/denied, and unauthorized attempts.
5. **Data Analytics**
    - Analyze usage data to provide reports on resource utilization, peak hours, and booking trends.
6. **Scalability**
    - Expand the system to handle multiple resources across large campuses, office buildings, or industries.
7. **Energy Efficiency & Offline Mode**
    - Implement low-power modes and offline caching of RFID data to improve system reliability during network outages.