**Project Idea: Remote Access Auto Door Lock**

* **Problem**: In today's fast-paced world, people often forget to lock or unlock their doors when needed. This can lead to security risks or inconvenience, such as someone needing access to the house (like guests, delivery personnel, or family members) when you're not home.
* **Solution**: Develop an IoT-based **Auto Door Lock** system that can be controlled via a mobile app, allowing users to lock or unlock their doors remotely. This solution would enhance home security, provide convenience, and allow users to monitor their door status from anywhere.

**Key Features:**

1. **Remote Access Control**: Lock and unlock the door from anywhere using a mobile app.
2. **Real-Time Notifications**: Send notifications whenever the door is locked or unlocked, or if unauthorized access is detected.
3. **Manual Override**: A keypad or RFID tag for local access in case the mobile app or internet connection fails.
4. **Security**: Encrypt communication between the mobile app and the door lock for enhanced security.
5. **Power Management**: Battery backup or low power consumption to ensure the system remains functional during power outages.
6. **Integration with Sensors**: PIR or Infrared sensors can detect motion near the door to trigger alerts.

**Required Components:**

* **ESP32**: Microcontroller with built-in Wi-Fi for remote communication.
* **Servo Motor or Electric Latch**: For controlling the physical lock mechanism.
* **Mobile App (App)**: Interface for controlling the lock from a smartphone.
* **Relay Module**: To control the electric lock or motor.
* **RFID Module (optional)**: For local access without needing the app.
* **PIR Sensor**: To detect movement near the door.
* **Power Supply (Battery/Adapter)**: To ensure the system operates continuously.
* **Encryption Module**: For secure communication between the app and lock.
* **Google Firebase/ThingSpeak**: To manage the backend and database for real-time data storage.
* **Buzzer**: Alerts in case of unauthorized access attempts.

**Steps:**

1. **Hardware Setup**:
   * Use **ESP32** to interface with the **servo motor** or **electric latch**.
   * Use **PIR sensor** to detect motion near the door for enhanced security.
   * Set up **relay modules** to control power to the lock.
   * Integrate **RFID** for local access as a backup.
2. **Mobile App Development**:
   * Create a mobile app to communicate with the ESP32 over Wi-Fi.
   * The app will send commands to lock/unlock the door.
   * Add features like viewing door status and receiving real-time notifications.
   * Use **Google Firebase** for remote control and to store logs of door activity.
3. **Software Development**:
   * Program the ESP32 to listen for commands from the mobile app and control the lock mechanism.
   * Implement encryption for secure communication.
   * Add logic for the **PIR sensor** to trigger an alert if motion is detected near the door.
4. **Testing & Integration**:
   * Test the entire system locally first (manual locking/unlocking).
   * Test the remote access feature through the app.
   * Test power failure scenarios and ensure the system operates on backup power.

**Auto Door Lock System with Remote Access Control**

**Features:**

1. **Compact and Minimalistic Design**:
   * The system will be designed to be as unobtrusive as possible, integrating into existing door hardware with minimal space requirements.
2. **Power Management**:
   * **Mains Power**: The primary power source will be mains electricity.
   * **Battery Backup**: If the building lacks an electric source, the system will automatically switch to battery power. The **18650 Li-ion batteries** you have will be used for backup, ensuring the lock works even during power outages.
3. **Mobile App Control** (for both Android and iOS):
   * A mobile app will allow users to lock and unlock the door from anywhere using **Wi-Fi**.
   * Free platforms like **Blynk** (with some limitations) or **Google Firebase** will be used for backend communication.
   * **Notifications**: The app will send push notifications for both successful lock/unlock actions and any unauthorized access attempts detected by the **RFID reader** or **PIR sensor**.
4. **RFID Reader**:
   * An **RFID reader (RC522)** will be integrated to provide local access without the mobile app. This allows users to unlock the door by scanning an RFID card, serving as a backup in case of a network or app failure.
5. **Security**:
   * **Encryption**: Communication between the mobile app and the lock will be encrypted for data security. We'll implement standard encryption protocols such as AES or SSL over **Google Firebase** to ensure that your commands and door status are secure.
   * **Unauthorized Access Detection**: The **PIR sensor** and **RFID reader** will detect motion near the door and trigger alerts if unauthorized access is attempted.
6. **Free Platforms**:
   * We'll use **Google Firebase** for database management (storing logs and sending data between the mobile app and ESP32).
   * **Blynk** (free version) can also be used to create a basic mobile app interface for controlling the lock and receiving notifications.
7. **Environment**:
   * The system will be suitable for both **home** and **office environments**, providing flexibility for different users and scenarios.

**Components (from your list):**

* **NODEMCU ESP32**: Acts as the controller for the lock and communicates with the mobile app via Wi-Fi.
* **RFID-RC522 Reader**: For local access using RFID tags.
* **Servo Motor (SG90)**: To physically lock and unlock the door.
* **PIR Sensor (HC-SR501)**: For detecting motion near the door and triggering security alerts.
* **Relay Module**: To control the servo motor or electric latch for door locking.
* **Battery Holder & 18650 Batteries**: For providing backup power in case of mains failure.
* **Breadboard & Jumper Wires**: For prototyping and connecting components.

**Project Workflow:**

1. **Hardware Setup**:
   * Connect the **ESP32** to the **relay module** and **servo motor** to control the lock mechanism.
   * Integrate the **RFID reader** to provide local access and the **PIR sensor** to detect motion.
   * Set up a **battery backup** system using your 18650 batteries.
   * Mount the components inside a compact, secure enclosure.
2. **Mobile App Development**:
   * Use **Google Firebase** to handle user authentication, door control commands, and notifications.
   * Develop a **cross-platform mobile app** using **Blynk** or other free platforms. The app will allow users to lock/unlock the door, view door status, and receive notifications.
3. **Software Development**:
   * Write the firmware for the **ESP32** to receive commands from the app and control the lock.
   * Implement **secure communication** between the app and the ESP32 (using encryption).
   * Add logic for the **PIR sensor** to send notifications if motion is detected.
4. **Testing**:
   * Test local access via **RFID tags**.
   * Test remote access via the **mobile app**.
   * Simulate network outages to test the **manual override** functionality with the **RFID reader**.
   * Test power failures to ensure smooth transitions between mains power and the battery backup.