**PARKBAI: READ.ME**

***RFID SCANNER (REGISTRATION):***

*# RFID Based Access Control System with Firebase Integration*

*This Arduino sketch is designed to create an RFID-based access control system using an ESP8266 module. The system integrates with Firebase for real-time monitoring and management of RFID cards.*

*## Components Used*

*- ESP8266WiFi library*

*- Wire library*

*- LiquidCrystal\_I2C library*

*- SPI library*

*- RFID library*

*- FirebaseESP8266 library*

*## Installation*

*1. Install the required libraries:*

*- ESP8266WiFi*

*- Wire*

*- LiquidCrystal\_I2C*

*- SPI*

*- RFID*

*- FirebaseESP8266 (Install this library from the Arduino Library Manager)*

*2. Update Firebase Credentials:*

*- Replace `FIREBASE\_HOST` with your Firebase Realtime Database URL.*

*- Replace `FIREBASE\_AUTH` with your Firebase Secret Key.*

*3. Update WiFi Credentials:*

*- Set your WiFi SSID and password in the `ssid` and `pass` variables.*

*4. Hardware Configuration:*

*- Connect RFID reader SDA pin to D4 and RST pin to D3.*

*5. Upload the sketch to your ESP8266 board.*

*## Usage*

*1. Open the Arduino IDE Serial Monitor.*

*2. Connect to WiFi:*

*- The sketch will attempt to connect to the specified WiFi network.*

*- The ESP8266 board's MAC address and IP address will be displayed upon successful connection.*

*3. RFID Card Registration:*

*- Tap a new RFID card near the reader to register it in the system.*

*- The RFID card number will be displayed on the Serial Monitor.*

*- The card will be registered in the Firebase database.*

*4. Access Control:*

*- If a registered RFID card is detected, the LCD will display "RFID CARD REGISTERED!"*

*- The RFID card number will be logged in the Firebase database under `/ADMIN/RFID\_CARDS/`.*

*## Notes*

*- Ensure the libraries and dependencies are properly installed before uploading the sketch.*

*- Make sure to handle your Firebase credentials securely.*

*Feel free to modify the sketch as needed for your specific use case.*

***RFID SCANNER WITH SERVO:***

*# RFID Based Parking System with Firebase Integration*

*This Arduino sketch is designed to create a RFID-based parking system using an ESP8266 module. The system integrates with Firebase for real-time monitoring and management of parking transactions. Additionally, it utilizes webhooks to send SMS notifications for parking events.*

*## Components Used*

*- Arduino IDE*

*- ESP8266WiFi library*

*- ESP8266Webhook library*

*- NTPClient library*

*- WiFiUdp library*

*- FirebaseESP8266 library*

*- Wire library*

*- LiquidCrystal\_I2C library*

*- SPI library*

*- RFID library*

*- TimeLib library*

*- ArduinoJson library*

*- Servo library*

*## Installation*

*1. Install the required libraries:*

*- ESP8266WiFi*

*- ESP8266Webhook*

*- NTPClient*

*- WiFiUdp*

*- FirebaseESP8266*

*- Wire*

*- LiquidCrystal\_I2C*

*- SPI*

*- RFID*

*- TimeLib*

*- ArduinoJson*

*- Servo*

*2. Update Firebase Credentials:*

*- Replace `FIREBASE\_HOST` with your Firebase Realtime Database URL.*

*- Replace `FIREBASE\_AUTH` with your Firebase Secret Key.*

*3. Update WiFi Credentials:*

*- Set your WiFi SSID and password in the `ssid` and `pass` variables.*

*4. Update Webhooks Key and Event:*

*- Replace `KEY` and `EVENT` with your specific values for webhooks integration.*

*5. Hardware Configuration:*

*- Connect RFID reader SDA pin to D4 and RST pin to D3.*

*- Connect I2C LCD to D1 (SCL) and D2 (SDA).*

*- Connect Servo motor to D8 for toll gate control.*

*6. Upload the sketch to your ESP8266 board.*

*## Usage*

*1. Open the Arduino IDE Serial Monitor.*

*2. Connect to WiFi:*

*- The sketch will attempt to connect to the specified WiFi network.*

*- The ESP8266 board's MAC address and IP address will be displayed upon successful connection.*

*3. RFID Card Detection:*

*- Tap an RFID card near the reader to initiate parking transactions.*

*- The system will check the access, calculate parking fees, and update the Firebase database accordingly.*

*- SMS notifications will be sent using webhooks.*

*4. Servo-controlled Toll Gate:*

*- The servo motor is used to control the toll gate, simulating the opening and closing of the gate.*

*## Notes*

*- Ensure the libraries and dependencies are properly installed before uploading the sketch.*

*- Make sure to handle your Firebase credentials securely.*

*- Adjust the servo motor connection as per your hardware setup.*

*Feel free to modify the sketch as needed for your specific use case.*

***PARKING SENSORS:***

*# Parking Space Monitoring System with Firebase Integration*

*This Arduino sketch is designed to create a parking space monitoring system using an ultrasonic sensor (HC-SR04) with an ESP8266 module. The system uses Firebase for real-time monitoring and management of parking space occupancy.*

*## Components Used*

*- Arduino IDE*

*- ESP8266WiFi library*

*- FirebaseESP8266 library*

*- NewPing library*

*## Installation*

*1. Install the required libraries:*

*- ESP8266WiFi*

*- FirebaseESP8266*

*- NewPing*

*2. Update Firebase Credentials:*

*- Replace `FIREBASE\_HOST` with your Firebase Realtime Database URL.*

*- Replace `FIREBASE\_AUTH` with your Firebase Secret Key.*

*- Modify `ownerUID` with your specific owner's UID.*

*3. Update WiFi Credentials:*

*- Set your WiFi SSID and password in the `ssid` and `password` variables.*

*4. Hardware Configuration:*

*- Connect the HC-SR04 ultrasonic sensor's TRIGGER pin to D5 and ECHO pin to D6.*

*5. Upload the sketch to your ESP8266 board.*

*## Usage*

*1. Open the Arduino IDE Serial Monitor.*

*2. Connect to WiFi:*

*- The sketch will attempt to connect to the specified WiFi network.*

*- The ESP8266 board's MAC address and IP address will be displayed upon successful connection.*

*3. Parking Space Monitoring:*

*- The system uses an ultrasonic sensor to measure distance.*

*- The sketch monitors parking space occupancy based on distance readings.*

*- Updates the Firebase database with parking space status (OCCUPIED, IMPROPER PARK, VACANT).*

*4. Adjust Thresholds:*

*- Modify the `detectOccupied`, `detectImproperPark1`, `detectImproperPark2`, `detectImproperPark3`, and `detectVacant` values based on your specific setup.*

*## Notes*

*- Ensure the libraries and dependencies are properly installed before uploading the sketch.*

*- Make sure to handle your Firebase credentials securely.*

*- Adjust the sensor thresholds according to your parking space dimensions and requirements.*

*Feel free to modify the sketch as needed for your specific use case.*