

# Sistem de Parcare Rezidentiala Intelligent IoT

## README.md

### Rezumat

Acest proiect implementeaza un sistem de parcare inteligent destinat unei parcare rezidentiale mici, utilizand tehnologie IoT si conectivitate cloud. Solutia include un microcontroller ESP32 pentru gestionarea hardware-ului local si un backend gazduit in cloud pentru procesarea datelor si gestionarea rezervarilor. Utilizatorii pot accesa parcare prin scanarea unui card RFID si pot rezerva locuri de parcare in timp real prin intermediul unei aplicatii web. Sistemul integreaza senzori ultrasonici, LED-uri, un ecran LCD si un servo motor pentru o interactiune eficienta si clara cu utilizatorii.

### Caracteristici

- **Autentificare RFID:** Accesul utilizatorilor in parcare se face prin scanarea unui card RFID autorizat.
- **Gestionare locuri de parcare:** Monitorizarea si gestionarea locurilor in timp real, afisand starea fiecarui loc (liber, rezervat, ocupat).
- **Rezervare prin aplicatie web:** Utilizatorii pot rezerva locuri de parcare printr-un dashboard web.
- **Feedback vizual:** LED-uri care indica starea locurilor de parcare si un ecran LCD pentru informatii in timp real.
- **Control securizat:** Blocarea accesului daca parcare este plina sau cardul nu este autorizat.

### Componentele Sistemului

#### Hardware:

- **ESP32:** Microcontroller principal cu conectivitate Wi-Fi.
- **Cititor RFID MFRC-522:** Pentru autentificarea utilizatorilor la intrarea in parcare.
- **Senzori Ultrasonici HC-SR04:** Pentru detectarea prezentei vehiculelor in locurile de parcare.
- **Servo Motor SG90:** Pentru controlul barierei de acces.
- **LCD I2C (16x2):** Pentru afisarea informatiilor legate de starea locurilor si a mesajelor de sistem.
- **LED-uri:** Indica vizual starea locurilor de parcare (verde = liber, galben = rezervat, rosu = ocupat).
- **Buton Push:** Pentru ridicarea manuala a barierei la iesire.

#### Software:

- **Cod ESP32:** Gestionarea senzorilor, controlul barierei si comunicatia cu backend-ul prin MQTT.
- **Backend Python (FastAPI):** Logica serverului pentru gestionarea rezervarilor, autentificarii si monitorizarii starii locurilor.

- **Frontend React:** Dashboard web pentru utilizatori si administratori.

## Fluxul de Functionare

### 1. Autentificare la intrare:

- Utilizatorul scaneaza cardul RFID la bariera.
- Sistemul verifica cardul in baza de date si statusul locurilor de parcare.
- Bariera se ridica daca utilizatorul are acces si exista un loc disponibil.

### 2. Monitorizare locuri:

- Senzorii ultrasonici detecteaza vehiculele si actualizeaza starea locurilor in timp real.
- Statusurile sunt afisate pe LCD si actualizate in aplicatia web.

### 3. Rezervare prin aplicatie web:

- Utilizatorul acceseaza dashboard-ul pentru a vedea locurile disponibile.
- Rezerva un loc introducand UID-ul cardului RFID si selectand locul dorit.
- Locul devine rezervat si LED-ul asociat se aprinde galben.

### 4. Blocare acces daca parcare este plina:

- Daca toate locurile sunt ocupate sau rezervate, accesul este refuzat si utilizatorul este informat prin mesaj pe LCD.

---

## Documentatie Preliminara

### 1. Introducere

#### Scopul Proiectului

Proiectul urmareste dezvoltarea unui sistem de parcare inteligent pentru gestionarea eficienta a accesului si a locurilor intr-o parcare rezidentiala. Utilizand tehnologia IoT, sistemul ofera o solutie moderna pentru monitorizarea si controlul parcarii, reducand timpul necesar pentru gasirea unui loc si asigurand securitatea accesului.

#### Obiective

- Implementarea unui sistem de autentificare prin RFID.
- Monitorizarea si gestionarea in timp real a locurilor de parcare.
- Dezvoltarea unei aplicatii web pentru rezervarea si vizualizarea statusului locurilor.
- Securizarea comunicatiei intre componente folosind protocoale criptate (TLS).
- Afişarea informatiilor relevante pentru utilizatori prin LCD si LED-uri.

---

## 2. Arhitectura

#### Descriere Generala a Sistemului

Sistemul este alcatuit din urmatoarele componente principale:

## 1. Hardware:

- **ESP32**: Responsabil pentru gestionarea senzorilor, controlul barierei si conectivitatea cu backend-ul.
- **RFID MFRC-522**: Detecteaza UID-ul cardurilor pentru autentificarea utilizatorilor.
- **Senzori Ultrasonici HC-SR04**: Detecteaza prezenta vehiculelor in locurile de parcare.
- **Servo Motor SG90**: Controleaza ridicarea si coborarea barierei.
- **LCD I2C (16x2)**: Afiseaza informatii despre locurile disponibile si mesaje pentru utilizatori.
- **LED-uri**: Indicatori vizuali ai starii locurilor (liber, rezervat, ocupat).

## 2. Software:

- **ESP32 Firmware**: Cod pentru gestionarea senzorilor, controlul barierei si comunicatia cu backend-ul prin MQTT.
- **Backend Python (FastAPI)**: Logica aplicatiei pentru autentificare, rezervari si procesare de date.
- **Frontend React**: Dashboard pentru utilizatori si administratori.

## 3. Rețea:

- **Wi-Fi**: Comunicatia dintre ESP32 si backend.
- **MQTT**: Protocol pentru transmiterea datelor intre ESP32 si backend.
- **HTTPS**: Securizarea comunicatiei intre frontend si backend.

## Fluxul de Date

1. Utilizatorul scaneaza cardul RFID.
2. ESP32 trimite UID-ul catre backend prin MQTT.
3. Backend-ul valideaza cardul si verifica statusul locurilor in baza de date.
4. Daca accesul este permis, ESP32 ridica bariera si actualizeaza statusul locului rezervat sau liber.
5. Datele despre statusul locurilor sunt afisate pe LCD si actualizate in dashboard-ul web.

## Topologia Rețelei

Sistemul utilizeaza o topologie star (stea):

- **ESP32** este nodul care gestioneaza toate componentele hardware si comunica cu backend-ul.
- **Backend-ul** este gazduit in cloud si serveste ca punct central pentru procesarea datelor si gestionarea rezervarilor.
- **Frontend-ul** este accesibil printr-un browser si afiseaza informatii in timp real.

## Alegerea Protocoalelor de Comunicare

1. **MQTT**: Protocol usor pentru comunicatia intre ESP32 si backend, ideal pentru transmisii in timp real cu consum redus de resurse.
2. **HTTPS/TLS**: Folosit pentru securizarea comunicatiei intre aplicatia web si backend.