

```

#include <Servo.h>
#include <LiquidCrystal_I2C.h>
#include <SPI.h>
#include <MFRC522.h>
#include <Wire.h>
#include "HX711.h"

// Define connections
#define DT 8 // Data pin
#define SCK 2 // Clock pin

HX711 scale; // Create HX711 instance
float weight = 0.0; // Variable to store weight

Servo motor1;
Servo motor2;

#define BUZZER A0 // A0 as digital pin

#define ir1 7
#define ir2 4
#define ir3 3

unsigned int totalslots = 2;
unsigned int e = totalslots;
LiquidCrystal_I2C lcd(0x27, 16, 2);

#define SS_PIN 10
#define RST_PIN 9
MFRC522 mfrc522(SS_PIN, RST_PIN);

int a = 34, b = 400;

void setup() {
    motor1.attach(6);
    motor2.attach(5);

    pinMode(ir1, INPUT);
    pinMode(ir2, INPUT);
    pinMode(ir3, INPUT);

    pinMode(BUZZER, OUTPUT);

    motor1.write(90);

```

```

motor2.write(00);

lcd.init();
lcd.backlight();

Serial.begin(9600);
SPI.begin();
delay(500);
mfr522.PCD_Init();

lcd.clear();
lcd.setCursor(0, 0);
lcd.print(" Car Parking");
lcd.setCursor(0, 1);
lcd.print(" System");
delay(1000);

updateDisplay();

Serial.println("Initializing HX711...");

scale.begin(DT, SCK); // Initialize HX711

if (scale.is_ready()) {
    Serial.println("HX711 is ready.");
} else {
    Serial.println("HX711 NOT found. Check wiring!");
    while (1); // Stop execution if HX711 is not detected
}

Serial.println("Remove any weight. Taring...");
delay(2000);

scale.set_scale(500); // Replace 500.0 with your calculated scale factor
scale.tare(); // Reset to zero
Serial.println("Calibration Done.");
}

void loop() {
    int ir1_value = digitalRead(ir1);
    int ir2_value = digitalRead(ir2);
    int ir3_value = digitalRead(ir3);

    if (scale.is_ready()) {

```

```

    weight = scale.get_units(5); // Store weight in variable
    delay(1000);
    Serial.print("Weight: ");
    Serial.print(weight, 2); // Print with 2 decimal places
    Serial.println(" g");
    delay(2000);

} else {
    Serial.println("HX711 not ready...");
}

if(weight > 150){
    tone(BUZZER, 2000); // Play 1000Hz tone
    delay(200);
    noTone(BUZZER); // Stop tone
    delay(500);
}

delay(500);
if (ir1_value == LOW && ir2_value == HIGH && e > 0 && (weight < 300)) {
    Serial.println("IR Sensor triggered. Waiting for RFID...");

    while (!mfrc522.PICC_IsNewCardPresent() || !mfrc522.PICC_ReadCardSerial()) {
        Serial.println("Waiting for RFID Card...");
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("RFID Card ?");
        delay(1500);
    }

    Serial.print("UID tag: ");
    String content = "";

    for (byte i = 0; i < mfrc522.uid.size; i++)
    {
        Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
        Serial.print(mfrc522.uid.uidByte[i], HEX);
        content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
        content.concat(String(mfrc522.uid.uidByte[i], HEX));
    }
    Serial.println();
    content.toUpperCase();

    if (content.substring(1) == "7E 4A 32 02") { // Card 1

```

```

Serial.println("Card 1 Detected");

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Available Amount:");
lcd.setCursor(0, 1);
lcd.print(a);
delay(3000);

if (a >= 35) {
  a -= 35; // Deduct balance
  Serial.print("New Balance for Card 1: ");
  Serial.println(a);

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Balance:");
  lcd.setCursor(0, 1);
  lcd.print(a);
  //delay(1000);

  if (a >= 0) {
    motor1.write(0);
    delay(500);
  }
  e=e-1;

} else {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Insufficient");
  lcd.setCursor(0, 1);
  lcd.print("amount");
  delay(1000);

  tone(BUZZER, 1000); // Play 1000Hz tone
  delay(200);
  noTone(BUZZER); // Stop tone
  delay(500);
}

}

else if (content.substring(1) == "03 C4 12 DA") { // Card 2
  Serial.println("Card 2 Detected");
}

```

```

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Available Amount:");
lcd.setCursor(0, 1);
lcd.print(b);
delay(3000);

if (b >= 35) {
  b -= 35; // Deduct balance
  Serial.print("New Balance for Card 2: ");
  Serial.println(b);

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Balance:");
  lcd.setCursor(0, 1);
  lcd.print(b);
  //delay(1000);

  if(b >= 0){
    motor1.write(0);
    delay(500);
  }
  e=e-1;

} else {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Insufficient");
  lcd.setCursor(0, 1);
  lcd.print("amount");
  delay(2000);

  tone(BUZZER, 1000); // Play 1000Hz tone
  delay(200);
  noTone(BUZZER); // Stop tone
  delay(500);
}

} else {
  Serial.println("Unknown Card");
  lcd.clear();

```

```

        lcd.setCursor(0, 0);
        lcd.print("Unauthorized");
        delay(2000);
    }

    mfrc522.PICC_HaltA();
    mfrc522.PCD_StopCrypto1();
}

else if (ir2_value == LOW && ir1_value == HIGH && e >= 0) {
    motor1.write(90);
    delay(1000);

    updateDisplay();
}
if (ir3_value == LOW && ir1_value == HIGH && ir2_value==HIGH && e >= 0) {
    motor2.write(90);
    delay(4000);
    motor2.write(0);
    e=e+1;
    if(e>2){
        e=2;
    }
    updateDisplay();
}
}

void updateDisplay() {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Parking Slots:");
    lcd.setCursor(0, 1);
    lcd.print("Available: ");
    lcd.print(e);
}

```