Wiring/Setup Configuration + Explanations on how each parts work separately and together:

Entry IR Sensor: used to detect when a car enters the parking lot, allowing the system to update the slot count and control the gate accordingly

Sensor	Arduino/BB
VCC	5V rail BB
GND	GND rail BB
OUT	Pin 2 Arduino

Steps:

- The loop() function continuously monitors the state of the entry sensor along with other sensors and inputs.
    - Inside the loop() function, there's a conditional statement that checks if the entry sensor is triggered, indicated by sensorValue == HIGH.
        - If the entry sensor is triggered and the exit sensor is not triggered (indicated by exitSensorValue == LOW), the code checks if a car is already entering or exiting to avoid concurrent operations. If no car is currently entering or exiting, it proceeds to update the slot count.
    - If there are available slots (indicated by slot > 0), the code decrements the slot count (slot--) to reflect that a car has entered the parking lot.
        - If there are no available slots, the code displays a message on the LCD indicating that the parking lot is full. It then exits the loop without performing any further actions related to servo movement or RFID scanning.
    - If a slot is available, the code rotates the servo motor to open the gate (servoMotor.write(0)) and records the time when the gate opened (gateOpenTime = millis()).
        - It sets a flag carEntering = true to indicate that a car is currently entering the parking lot.
    - After a certain delay (gateOpenDuration), the code closes the gate by rotating the servo motor back to its initial position (servoMotor.write(90)).

Servo Motor: responsible for controlling the gate barrier, opening it when a car enters or exits the parking lot and closing it after a certain period

Servo	Arduino/BB
PWR	5V rail BB
GND	GND rail BB
Signal	Pin 3 Arduino

- Inside the loop() function, there are conditional statements that check if access is granted and if the entry or exit sensors are triggered.
    - When access is granted and a car is detected by either the entry or exit sensor, the servo motor rotates to open or close the gate barrier.
        - If a car is detected exiting the parking lot (exitSensorValue == HIGH && sensorValue == LOW), and the gate is not already opening or closing (!carEntering && !carExiting), the servo motor rotates to open the gate (servoMotor.write(0)).
        - Similarly, if a car is detected entering the parking lot (sensorValue == HIGH && exitSensorValue == LOW), and the gate is not already opening or closing (!carEntering && !carExiting), the servo motor rotates to open the gate (servoMotor.write(0)).
            - After a specified delay (gateOpenDuration), the servo motor rotates back to its initial position to close the gate (servoMotor.write(90)).
- The gateOpenTime variable is used to record the time when the gate was opened, allowing the system to close it after the specified duration.

Exit IR Sensor: detects when a car is leaving the parking lot, triggers the opening of the gate barrier, and updates the slot count to reflect the availability of parking slots

Sensor	Arduino/BB
VCC	5V rail BB
GND	GND rail BB
OUT	Pin 4 Arduino

- Inside the loop() function, there's a conditional statement that checks if access is granted and if the exit sensor is triggered.
    - When access is granted and the exit sensor detects a car leaving (exitSensorValue == HIGH), the code checks if the entry sensor is not triggered (sensorValue == LOW). This ensures that the exit sensor is triggered before the entry sensor, preventing false readings.
    - If the conditions are met and the gate is not already opening or closing (!carEntering && !carExiting), the slot count is incremented by one (slot++), indicating that a parking slot has become available.
- The servo motor is then activated to open the gate barrier (servoMotor.write(0)), allowing the exiting car to leave the parking lot.
- The gateOpenTime variable is used to record the time when the gate was opened, enabling the system to close it after a specified duration (gateOpenDuration).
- After the gate has been open for the specified duration, the servo motor rotates back to its initial position to close the gate (servoMotor.write(90)).

Parking Spot 1 (IR): monitored by the sensor connected to pin ‘ir_car1’, and its status (empty or full) is displayed on the LCD screen in real-time

Sensor	Arduino/BB

VCC	5V rail BB
GND	GND rail BB
OUT	Pin 5 Arduino

- Inside the Read_Sensor() function, the status of parking spot 1 is read using digitalRead(ir_car1). If the sensor returns a LOW signal (0), it indicates that there is a car present in parking spot 1, and the variable S1 is set to 1. Otherwise, if the sensor returns a HIGH signal (1), it indicates that the parking spot is empty, and S1 is set to 0.
    - The status of parking spot 1 is then displayed on the LCD screen in the Update_LCD() function. If S1 is equal to 1, it indicates that the spot is occupied, and the LCD will display "Full" next to the label for parking spot 1. If S1 is equal to 0, it indicates that the spot is empty, and the LCD will display "Empty" next to the label for parking spot 1.
- This process repeats continuously in the loop() function, allowing the system to constantly monitor the status of parking spot 1 and update the display accordingly.

Parking Spot 2 (IR): monitored by the sensor connected to pin ‘ir_car2’, and its status (empty or full) is displayed on the LCD screen in real-time

Sensor	Arduino/BB
VCC	5V rail BB
GND	GND rail BB
OUT	Pin 6 Arduino

- Inside the Read_Sensor() function, the status of parking spot 2 is read using digitalRead(ir_car2). If the sensor returns a LOW signal (0), it indicates that there is a car present in parking spot 2, and the variable S2 is set to 1. Otherwise, if the sensor returns a HIGH signal (1), it indicates that the parking spot is empty, and S2 is set to 0.
    - The status of parking spot 2 is then displayed on the LCD screen in the Update_LCD() function. If S2 is equal to 1, it indicates that the spot is occupied, and the LCD will display "Full" next to the label for parking spot 2. If S2 is equal to 0, it indicates that the spot is empty, and the LCD will display "Empty" next to the label for parking spot 2.
- This process repeats continuously in the loop() function, allowing the system to constantly monitor the status of parking spot 2 and update the display accordingly.

Parking Spot 3 (IR): monitored by the sensor connected to pin ‘ir_car3’, and its status (empty or full) is displayed on the LCD screen in real-time

Sensor	Arduino/BB
VCC	5V rail BB
GND	GND rail BB
OUT	Pin 7 Arduino

- Inside the Read_Sensor() function, the status of parking spot 3 is read using digitalRead(ir_car3). If the sensor returns a LOW signal (0), it indicates that there is a car present in parking spot 3, and the variable S3 is set to 1. Otherwise, if the sensor returns a HIGH signal (1), it indicates that the parking spot is empty, and S3 is set to 0.
    - The status of parking spot 3 is then displayed on the LCD screen in the Update_LCD() function. If S3 is equal to 1, it indicates that the spot is occupied, and the LCD will display "Full" next to the label for parking spot 3. If S3 is equal to 0, it indicates that the spot is empty, and the LCD will display "Empty" next to the label for parking spot 3.
- This process repeats continuously in the loop() function, allowing the system to constantly monitor the status of parking spot 3 and update the display accordingly.

Parking Spot 4 (IR): monitored by the sensor connected to pin 'ir_car4', and its status (empty or full) is displayed on the LCD screen in real-time

Sensor	Arduino/BB
VCC	5V rail BB
GND	GND rail BB
OUT	Pin 8 Arduino

- Inside the Read_Sensor() function, the status of parking spot 2 is read using digitalRead(ir_car2). If the sensor returns a LOW signal (0), it indicates that there is a car present in parking spot 2, and the variable S2 is set to 1. Otherwise, if the sensor returns a HIGH signal (1), it indicates that the parking spot is empty, and S2 is set to 0.
    - The status of parking spot 2 is then displayed on the LCD screen in the Update_LCD() function. If S2 is equal to 1, it indicates that the spot is occupied, and the LCD will display "Full" next to the label for parking spot 2. If S2 is equal to 0, it indicates that the spot is empty, and the LCD will display "Empty" next to the label for parking spot 2.
- This process repeats continuously in the loop() function, allowing the system to constantly monitor the status of parking spot 2 and update the display accordingly.

LCD Display (IR):

I2C	Arduino/BB
VCC	5V rail BB
GND	GND rail BB
SDA	A4 (analog in)
SCL	A5 (analog in)

RFID module:

RFID	Arduino/BB
SDA	Pin 10 Arduino

SCK	Pin 13 Arduino
MOSI	Pin 11 Arduino
MISO	Pin 12 Arduino
GND	GND Arduino
RST	Pin 9 Arduino
3.3V	3.3V Pin Arduino

Code Info:

1. Setup Phase:
    - The system initializes by setting up serial communication, defining pin configurations for various sensors (IR sensors, exit sensor, car presence sensors), initializing the servo motor and LCD display, and initializing the MFRC522 RFID reader.
    - The LCD display is cleared and initialized to show a system message for a brief period.
2. Main Loop:
    - The main loop continuously executes two primary tasks: monitoring sensor inputs and checking RFID card authentication.
3. Sensor Monitoring:
    - The system continuously monitors the status of various sensors, including IR sensors (for detecting cars in parking spots), an exit sensor (for detecting cars leaving the parking area), and RFID card readers.
    - Depending on the sensor inputs, the system updates the slot count, controls the servo motor to open/close the parking gate, and updates the LCD display to show the current parking status.
4. RFID Card Authentication:
    - When a new RFID card is detected, the system reads its unique identifier (UID) and compares it with a list of authorized UIDs stored in the authorizedTags array.
    - If the UID matches one of the authorized UIDs, access is granted, and the gate opens. Otherwise, access is denied.
5. Gate Control:
    - The system controls the servo motor to open the gate when a car is exiting or entering, based on sensor inputs and access authorization.
    - After a predefined delay (gateOpenDuration), the gate closes automatically.
6. LCD Display:
    - The LCD display continuously updates to show the number of available parking slots (slot), as well as the status of individual parking spots (full or empty).
7. Functions:

- Several helper functions are defined to simplify code readability and organization. These include functions to read sensor inputs (Read_Sensor), update the LCD display (Update_LCD), and compare RFID UIDs (compareUid)