**Materials Needed:**

Arduino Uno

20x4 LCD Display with I2C module

Male and Female Headers

8x IR Sensors

Mini Servo Motor (SG-90)

5V 2A Power Adapter

Jumper Wires

Breadboard

**Step-by-Step Installation Guide:**

**1. Prepare the Components:**

a. Solder male headers to the LCD display and female headers to the I2C module (if they're not pre-soldered).

b. Connect the IR sensors to the Arduino using jumper wires. Connect VCC, GND, and the signal (out) pins for each sensor.

c. Connect the servo motor to the Arduino. Connect the servo's power (red) and ground (brown) wires to the 5V and GND pins on the Arduino. Connect the servo's signal (orange or yellow) wire to a digital pin (e.g., D9) on the Arduino.

**2. Connect the I2C LCD:**

a. Connect the I2C module to the LCD display using the male and female headers.

b. Connect the I2C module to the Arduino. Use the SDA and SCL pins on the module and connect them to the corresponding pins on the Arduino (A4 for SDA, A5 for SCL).

**3. Power Supply:**

1. Connect one end of the USB cable to the USB port on computer and the other end of the USB cable to the USB port on the Arduino Uno board.

**4. Code Development:**

a. Write an Arduino sketch that reads data from the IR sensors, displays information on the LCD screen, and controls the servo motor to simulate a parking barrier.

b. Install the required libraries for the LCD and I2C communication.

**5. Mount the Components:**

a. Place the IR sensors at suitable locations in the parking area, ensuring they can detect vehicles.

b. Mount the LCD display in a visible location for monitoring.

c. Mount the servo motor to simulate a parking barrier or gate.

**6. Testing:**

a. Upload the Arduino code to the board.

b. Test the system by moving objects (simulating cars) in front of the IR sensors.

c. Verify that the LCD displays relevant information, such as parking availability, and the servo motor moves to simulate the parking barrier.

**7. Fine-Tuning and Expansion:**

a. Adjust sensor positions and code parameters as needed for optimal performance.