To build and implement the **Smart Parking System** project described in your code, here are the details of **hardware connections, required components, and software setup**:

**Required Components**

**1. Hardware Components**

* **Microcontroller**: ESP32
* **Ultrasonic Sensor**: HC-SR04
* **LEDs**:
  + Green LED (GPIO 2)
  + Red LED (GPIO 4)
  + White LED (GPIO 14) - Represents booking status
* **Buzzer**: Piezoelectric buzzer (GPIO 27)
* **Jumper Wires**: For connections
* **Breadboard**: For prototyping
* **Resistors**: (330Ω or 220Ω for LEDs)
* **Power Supply**: 5V USB power for ESP32

**2. Software Requirements**

* **Arduino IDE**: Install and configure for ESP32
* **WiFi and ThingSpeak Libraries**:
  + Install WiFi.h (pre-installed with ESP32 core)
  + Install ThingSpeak library via Arduino Library Manager
* **ThingSpeak Account**:
  + Create an account at [ThingSpeak](https://thingspeak.com) to get API keys and channel IDs.
* **Web Browser**: To test the HTTP server and view status.

**Hardware Connections**

**1. Ultrasonic Sensor (HC-SR04)**

| **Pin on HC-SR04** | **Connect to ESP32** |
| --- | --- |
| **VCC** | **3.3V** |
| **GND** | **GND** |
| **TRIG** | **GPIO 5** |
| **ECHO** | **GPIO 18** |
|  |  |

**2. LEDs**

| **LED Color** | **ESP32 GPIO Pin** | **Resistor** | **Connection Details** |
| --- | --- | --- | --- |
| **Green** | GPIO 2 | 330Ω | Anode to GPIO, cathode to GND |
| **Red** | GPIO 4 | 330Ω | Anode to GPIO, cathode to GND |
| **White** | GPIO 14 | 330Ω | Anode to GPIO, cathode to GND |
|  |  |  |  |

**3. Buzzer**

| **Buzzer Pin** | **ESP32 GPIO Pin** |
| --- | --- |
| Positive | GPIO 27 |
| Negative | GND |

**Steps to Set Up**

**1. Hardware Setup**

1. Connect the **HC-SR04 sensor** to the ESP32 as described above.
2. Connect the **LEDs** and **buzzer** to their respective GPIO pins via resistors.
3. Mount all components on the breadboard for easy wiring.

**2. ESP32 Configuration**

1. Install **ESP32 board** in Arduino IDE:
   * Go to File > Preferences.
   * Add the URL: https://dl.espressif.com/dl/package\_esp32\_index.json in the Additional Board Manager URLs.
   * Open Tools > Board > Boards Manager, search for **ESP32**, and install it.
2. Install the **ThingSpeak** library:
   * Go to Sketch > Include Library > Manage Libraries.
   * Search for **ThingSpeak** and install it.

**3. ThingSpeak Configuration**

1. Create a ThingSpeak account and a new channel.
2. Add fields:
   * Field 1: **Parking Slot Status** (1 for vacant, 0 for occupied).
   * Field 3: **Booking Status** (1 for booked, 0 for not booked).
3. Note down the **Channel ID** and **Write API Key**.

**4. WiFi Setup**

1. Replace ssid and password in your code with your WiFi credentials:

cpp

Copy code

const char\* ssid = "YourWiFiName";

const char\* password = "YourWiFiPassword";

**5. Testing the System**

1. Upload the code to the ESP32 via Arduino IDE.
2. Open the Serial Monitor to verify the WiFi connection and debug messages.
3. Open a web browser and access the ESP32’s local IP address (shown on the Serial Monitor) to interact with the system:
   * /book: Book the parking slot.
   * /unbook: Unbook the parking slot.
   * /status: Get parking slot status in JSON format.

**Expected Behavior**

* **No vehicle present**: Green LED ON.
* **Vehicle detected**: Red LED ON.
* **Slot booked**:
  + White LED ON.
  + If a vehicle occupies a booked slot, the buzzer will sound.
* **Data** is updated to ThingSpeak every 20 seconds.

To build and implement the **Smart Parking System** project described in your code, here are the details of **hardware connections, required components, and software setup**:

**Required Components**

**1. Hardware Components**

* **Microcontroller**: ESP32
* **Ultrasonic Sensor**: HC-SR04
* **LEDs**:
  + Green LED (GPIO 2)
  + Red LED (GPIO 4)
  + White LED (GPIO 14) - Represents booking status
* **Buzzer**: Piezoelectric buzzer (GPIO 27)
* **Jumper Wires**: For connections
* **Breadboard**: For prototyping
* **Resistors**: (330Ω or 220Ω for LEDs)
* **Power Supply**: 5V USB power for ESP32

**2. Software Requirements**

* **Arduino IDE**: Install and configure for ESP32
* **WiFi and ThingSpeak Libraries**:
  + Install WiFi.h (pre-installed with ESP32 core)
  + Install ThingSpeak library via Arduino Library Manager
* **ThingSpeak Account**:
  + Create an account at [ThingSpeak](https://thingspeak.com) to get API keys and channel IDs.
* **Web Browser**: To test the HTTP server and view status.

**Hardware Connections**

**1. Ultrasonic Sensor (HC-SR04)**

| **Pin on HC-SR04** | **Connect to ESP32** |
| --- | --- |
| **VCC** | **3.3V** |
| **GND** | **GND** |
| **TRIG** | **GPIO 5** |
| **ECHO** | **GPIO 18** |

**2. LEDs**

| **LED Color** | **ESP32 GPIO Pin** | **Resistor** | **Connection Details** |
| --- | --- | --- | --- |
| **Green** | GPIO 2 | 330Ω | Anode to GPIO, cathode to GND |
| **Red** | GPIO 4 | 330Ω | Anode to GPIO, cathode to GND |
| **White** | GPIO 14 | 330Ω | Anode to GPIO, cathode to GND |

**3. Buzzer**

| **Buzzer Pin** | **ESP32 GPIO Pin** |
| --- | --- |
| Positive | GPIO 27 |
| Negative | GND |

**Steps to Set Up**

**1. Hardware Setup**

1. Connect the **HC-SR04 sensor** to the ESP32 as described above.
2. Connect the **LEDs** and **buzzer** to their respective GPIO pins via resistors.
3. Mount all components on the breadboard for easy wiring.

**2. ESP32 Configuration**

1. Install **ESP32 board** in Arduino IDE:
   * Go to File > Preferences.
   * Add the URL: https://dl.espressif.com/dl/package\_esp32\_index.json in the Additional Board Manager URLs.
   * Open Tools > Board > Boards Manager, search for **ESP32**, and install it.
2. Install the **ThingSpeak** library:
   * Go to Sketch > Include Library > Manage Libraries.
   * Search for **ThingSpeak** and install it.

**3. ThingSpeak Configuration**

1. Create a ThingSpeak account and a new channel.
2. Add fields:
   * Field 1: **Parking Slot Status** (1 for vacant, 0 for occupied).
   * Field 3: **Booking Status** (1 for booked, 0 for not booked).
3. Note down the **Channel ID** and **Write API Key**.

**4. WiFi Setup**

1. Replace ssid and password in your code with your WiFi credentials:

cpp

Copy code

const char\* ssid = "YourWiFiName";

const char\* password = "YourWiFiPassword";

**5. Testing the System**

1. Upload the code to the ESP32 via Arduino IDE.
2. Open the Serial Monitor to verify the WiFi connection and debug messages.
3. Open a web browser and access the ESP32’s local IP address (shown on the Serial Monitor) to interact with the system:
   * /book: Book the parking slot.
   * /unbook: Unbook the parking slot.
   * /status: Get parking slot status in JSON format.

**Expected Behavior** **To build and implement the Smart Parking System project described in your code, here are the details of hardware connections, required components, and software setup:**

**Required Components**

**1. Hardware Components**

**Microcontroller: ESP32**

**Ultrasonic Sensor: HC-SR04**

**LEDs:**

**Green LED (GPIO 2)**

**Red LED (GPIO 4)**

**White LED (GPIO 14) - Represents booking status**

**Buzzer: Piezoelectric buzzer (GPIO 27)**

**Jumper Wires: For connections**

**Breadboard: For prototyping**

**Resistors: (330Ω or 220Ω for LEDs)**

**Power Supply: 5V USB power for ESP32**

**2. Software Requirements**

**Arduino IDE: Install and configure for ESP32**

**WiFi and ThingSpeak Libraries:**

**Install WiFi.h (pre-installed with ESP32 core)**

**Install ThingSpeak library via Arduino Library Manager**

**ThingSpeak Account:**

**Create an account at ThingSpeak to get API keys and channel IDs.**

**Web Browser: To test the HTTP server and view status.**

**Hardware Connections**

**1. Ultrasonic Sensor (HC-SR04)**

**Pin on HC-SR04 Connect to ESP32**

**VCC 3.3V**

**GND GND**

**TRIG GPIO 5**

**ECHO GPIO 18**

**2. LEDs**

**LED Color ESP32 GPIO Pin Resistor Connection Details**

**Green GPIO 2 330Ω Anode to GPIO, cathode to GND**

**Red GPIO 4 330Ω Anode to GPIO, cathode to GND**

**White GPIO 14 330Ω Anode to GPIO, cathode to GND**

**3. Buzzer**

**Buzzer Pin ESP32 GPIO Pin**

**Positive GPIO 27**

**Negative GND**

**Steps to Set Up**

**1. Hardware Setup**

**Connect the HC-SR04 sensor to the ESP32 as described above.**

**Connect the LEDs and buzzer to their respective GPIO pins via resistors.**

**Mount all components on the breadboard for easy wiring.**

**2. ESP32 Configuration**

**Install ESP32 board in Arduino IDE:**

**Go to File > Preferences.**

**Add the URL: https://dl.espressif.com/dl/package\_esp32\_index.json in the Additional Board Manager URLs.**

**Open Tools > Board > Boards Manager, search for ESP32, and install it.**

**Install the ThingSpeak library:**

**Go to Sketch > Include Library > Manage Libraries.**

**Search for ThingSpeak and install it.**

**3. ThingSpeak Configuration**

**Create a ThingSpeak account and a new channel.**

**Add fields:**

**Field 1: Parking Slot Status (1 for vacant, 0 for occupied).**

**Field 3: Booking Status (1 for booked, 0 for not booked).**

**Note down the Channel ID and Write API Key.**

**4. WiFi Setup**

**Replace ssid and password in your code with your WiFi credentials:**

**cpp**

**Copy code**

**const char\* ssid = "YourWiFiName";**

**const char\* password = "YourWiFiPassword";**

**5. Testing the System**

**Upload the code to the ESP32 via Arduino IDE.**

**Open the Serial Monitor to verify the WiFi connection and debug messages.**

**Open a web browser and access the ESP32’s local IP address (shown on the Serial Monitor) to interact with the system:**

**/book: Book the parking slot.**

**/unbook: Unbook the parking slot.**

**/status: Get parking slot status in JSON format.**

**Expected Behavior**

**No vehicle present: Green LED ON.**

**Vehicle detected: Red LED ON.**

**Slot booked:**

**White LED ON.**

**If a vehicle occupies a booked slot, the buzzer will sound.**

**Data is updated to ThingSpeak every 20 seconds.**

* **No vehicle present**: Green LED ON.
* **Vehicle detected**: Red LED ON.
* **Slot booked**:
  + White LED ON.
  + If a vehicle occupies a booked slot, the buzzer will sound.
* **Data** is updated to ThingSpeak every 20 seconds.