1. **Arduino Code**:
    - Update the Arduino code to handle JSON responses for easy parsing in JavaScript.

Explanation:

1. **Wi-Fi Setup**:
    - The ESP8266 connects to the specified Wi-Fi network using the SSID and password provided.
2. **Server Setup**:
    - A web server is started on port 80.
    - The server responds to two types of requests: the root page `/` and the data endpoint `/data`.
3. **HTML and JavaScript**:
    - The HTML response contains a simple webpage with placeholders (`<span>` elements) for the sensor data.
    - JavaScript fetches new data from the `/data` endpoint every second and updates the webpage.

Accessing the Web Dashboard:

1. **Find the IP Address**:
    - After the ESP8266 connects to the Wi-Fi network, note the IP address printed to the serial monitor (e.g., `192.168.1.100`).
2. **Open Web Browser**:
    - Open a web browser on any device connected to the same Wi-Fi network.
    - Enter the IP address of the ESP8266 in the address bar (e.g., `http://192.168.1.100`).
3. **View Data**:
    - The webpage will display the current pressure, temperature, date, and time.
    - The data will automatically update every second.

This setup allows you to create a simple and continuously updating web dashboard using the NodeMCU ESP8266 and the BMP180 and Tiny RTC modules.

Connections:

1. **BMP180 Connections**:
    - **BMP180 VIN** to **NodeMCU 3V3**
    - **BMP180 GND** to **NodeMCU GND**
    - **BMP180 SCL** to **NodeMCU D1 (GPIO5)**
    - **BMP180 SDA** to **NodeMCU D2 (GPIO4)**
2. **Tiny RTC Connections**:
    - **RTC VCC** to **NodeMCU 3V3**
    - **RTC GND** to **NodeMCU GND**
    - **RTC SCL** to **NodeMCU D1 (GPIO5)**
    - **RTC SDA** to **NodeMCU D2 (GPIO4)**