Here is a complete, single-file HTML page that provides a clean, intuitive, and modern-looking web portal for your temperature sensor.

It includes:

1. **HTML:** The structure for the dashboard, using "widgets" for each function.
2. **CSS:** Styling to make it look like a professional, modern dashboard (responsive and clean).
3. **JavaScript:** Basic interactivity to make the controls work (e.g., updating the temperature precision and LED status).

You can copy and paste this code into a file named `index.html` and open it in your browser.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sensor Control Panel</title>
    <style>
        /* --- Global Styles --- */
        body {
            font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, Helvetica, Arial, sans-serif;
            background-color: #f4f7f6;
            margin: 0;
            padding: 20px;
            display: flex;
            justify-content: center;
            align-items: flex-start;
            min-height: 100vh;
            color: #333;
        }

        /* --- Main Container --- */
        .container {
            width: 100%;
            max-width: 900px;
            display: grid;
            /* Responsive grid: 1 column on small screens, 2 or 3 on larger ones */
            grid-template-columns: repeat(auto-fit, minmax(280px, 1fr));
            gap: 25px;
        }

        h1 {
            width: 100%;
            text-align: center;
            color: #222;
            margin-bottom: 20px;
            grid-column: 1 / -1; /* Make title span all columns */
        }

        /* --- Widget Card Style --- */
        .widget {
            background-color: #ffffff;
            border-radius: 12px;
            box-shadow: 0 6px 15px rgba(0, 0, 0, 0.07);
            padding: 25px;
            text-align: center;
            transition: transform 0.2s ease, box-shadow 0.2s ease;
        }

        .widget:hover {
            transform: translateY(-3px);
            box-shadow: 0 8px 20px rgba(0, 0, 0, 0.09);
        }

        .widget h2 {
            margin-top: 0;
            margin-bottom: 20px;
            color: #0056b3; /* A nice accent blue */
            border-bottom: 2px solid #eee;
```

```
        padding-bottom: 10px;
    }

    /* --- Temperature Display Widget --- */
    #temp-display {
        font-size: 4.5rem; /* Large, easy-to-read font */
        font-weight: 600;
        color: #1a1a1a;
        margin: 10px 0;
        line-height: 1;
    }

    /* --- LED Control Widget --- */
    .led-control-flex {
        display: flex;
        justify-content: center;
        align-items: center;
        gap: 20px; /* Space between switch and status */
        margin-top: 25px;
    }

    #led-status {
        font-size: 1.2rem;
        font-weight: 500;
        color: #888;
        width: 50px; /* Fixed width to prevent layout shift */
        text-align: left;
    }

    /* The Toggle Switch */
    .toggle-switch {
        position: relative;
        display: inline-block;
        width: 60px;
        height: 34px;
    }

    .toggle-switch input {
        opacity: 0;
        width: 0;
        height: 0;
    }

    .slider {
        position: absolute;
        cursor: pointer;
        top: 0;
        left: 0;
        right: 0;
        bottom: 0;
        background-color: #ccc;
        transition: .4s;
        border-radius: 34px;
    }

    .slider:before {
        position: absolute;
        content: "";
        height: 26px;
        width: 26px;
        left: 4px;
        bottom: 4px;
        background-color: white;
        transition: .4s;
        border-radius: 50%;
```

```
            box-shadow: 0 2px 4px rgba(0,0,0,0.2);
        }

        input:checked + .slider {
            background-color: #28a745; /* Green for 'On' */
        }

        input:checked + .slider:before {
            transform: translateX(26px);
        }

        /* --- Settings Widget --- */
        .settings-control {
            display: flex;
            justify-content: space-between;
            align-items: center;
            margin-top: 25px;
        }

        .settings-control label {
            font-size: 1.1rem;
            font-weight: 500;
            color: #555;
        }

        #precision-select {
            font-size: 1rem;
            padding: 8px 12px;
            border-radius: 6px;
            border: 1px solid #ddd;
            background-color: #fdfdfd;
            cursor: pointer;
        }

        #precision-select:focus {
            outline: none;
            border-color: #0056b3;
            box-shadow: 0 0 0 3px rgba(0,86,179,0.2);
        }

    </style>
</head>
<body>

    <div class="container">
        <h1>Sensor Control Panel </h1>

        <div class="widget">
            <h2>Live Temperature</h2>
            <div id="temp-display">23.4°C</div>
        </div>

        <div class="widget">
            <h2>Device LED</h2>
            <div class="led-control-flex">
                <label class="toggle-switch">
                    <input type="checkbox" id="led-toggle">
                    <span class="slider"></span>
                </label>
                <span id="led-status">Off</span>
            </div>
        </div>

        <div class="widget">
            <h2>Settings</h2>
            <div class="settings-control">
```

```
            <label for="precision-select">Readout Precision:</label>
            <select id="precision-select">
                <option value="0">0 (e.g., 23°C)</option>
                <option value="1" selected>1 (e.g., 23.4°C)</option>
                <option value="2">2 (e.g., 23.45°C)</option>
            </select>
        </div>
    </div>

</div>

<script>
    // Wait for the document to be fully loaded
    document.addEventListener('DOMContentLoaded', () => {

        // --- Element References ---
        const tempDisplay = document.getElementById('temp-display');
        const ledToggle = document.getElementById('led-toggle');
        const ledStatus = document.getElementById('led-status');
        const precisionSelect = document.getElementById('precision-select');

        // --- State Variables ---
        // This is the "true" sensor value. We will format it for display.
        // In a real app, you'd get this from your server (e.g., via fetch or WebSocket).
        let currentTemp = 23.41823;
        let currentPrecision = 1;

        // --- Update Functions ---

        /**
         * Updates the temperature display text based on the current precision.
         */
        function updateTemperatureDisplay() {
            const formattedTemp = currentTemp.toFixed(currentPrecision);
            tempDisplay.textContent = `${formattedTemp}°C`;
        }

        // --- Event Listeners ---

        /**
         * Handle LED toggle change
         */
        ledToggle.addEventListener('change', () => {
            const isChecked = ledToggle.checked;
            ledStatus.textContent = isChecked ? 'On' : 'Off';
            ledStatus.style.color = isChecked ? '#28a745' : '#888';

            // TODO: Send this state to your server/device
            console.log(`LED state changed to: ${ledStatus.textContent}`);
            // Example: fetch('/api/led', { method: 'POST', body: JSON.stringify({ state: isChecked }) });
        });

        /**
         * Handle Precision dropdown change
         */
        precisionSelect.addEventListener('change', () => {
            currentPrecision = parseInt(precisionSelect.value, 10);
            updateTemperatureDisplay(); // Re-format the temperature

            // TODO: You could save this preference locally or on the server
            console.log(`Precision changed to: ${currentPrecision} decimal(s)`);
        });

        // --- Initial Setup ---
        // Set the initial temperature display on page load
```

```
            updateTemperatureDisplay();

            // Set initial LED status text color
            ledStatus.style.color = ledToggle.checked ? '#28a745' : '#888';
        });
    </script>

</body>
</html>
```