```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <style>
        /* Apply styling for the body element */
        body {
            display: flex;
            flex-direction: column;
            align-items: center;
            justify-content: center;
            height: 100vh;
            margin: 0;
        }

        /* Style the speed range input */
        #speedRange {
            width: 80%;
            margin-bottom: 20px;
        }

        /* Style for displaying cursor value */
        #cursorValue {
            font-size: 18px;
            margin-top: 10px;
        }
    </style>
    <title>Adjustable Speed Cursor</title>
</head>
<body>

<!-- Input element for adjusting cursor speed -->
<input type="range" id="speedRange" min="0" max="255"
value="128">
```

```
<!-- Display the current cursor value -->
<p id="cursorValue">Cursor Value: 128</p>

<script>
    // Get references to DOM elements
    const speedRange = document.getElementById('speedRange');
    const cursorValueElement =
document.getElementById('cursorValue');

    // Add event listener to update cursor value and execute
link on input change
    speedRange.addEventListener('input', (e) => {
        // Parse the cursor value as an integer
        const cursorValue = parseInt(e.target.value, 10);
        // Update the displayed cursor value
        updateCursor(cursorValue);
        // Execute the link with the current cursor value
        executeLink(cursorValue);
    });

    // Function to update the displayed cursor value
    function updateCursor(cursorValue) {
        cursorValueElement.textContent = `Cursor Value:
${cursorValue}`;
    }

    // Function to execute a link with the provided cursor value
    function executeLink(cursorValue) {
        // Construct the API URL with the cursor value
        const apiUrl =
`https://api.thingspeak.com/update?api_key=ELW2NF5Q83OGB39G&fiel
d5=${cursorValue}`;

        // Use fetch to make an HTTP request to the provided
link
        fetch(apiUrl)
```

```
            .then(response => {
                // Check for HTTP errors
                if (!response.ok) {
                    throw new Error(`HTTP error! Status:
${response.status}`);
                }
                // Parse the response as text
                return response.text();
            })
            .then(data => {
                // Log success message with the response data
                console.log(`Link executed successfully.
Response: ${data}`);
            })
            .catch(error => {
                // Log error if the link execution fails
                console.error('Error executing link:', error);
            });
    }
</script>

</body>
</html>
```

Output: