HTML code to open link upon pressing button in the background without redirecting:

```
<!-- Declaring the document type and indicating that it's HTML -->
<!DOCTYPE html>
<!-- Opening the HTML document and specifying the language as English -->
<html lang="en">
<head>
  <!-- Metadata specifying the character encoding for the document -->
  <meta charset="UTF-8">
  <!-- Metadata specifying the viewport settings for responsive design -->
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- Title of the webpage -->
  <title>Background Request Button</title>
</head>
<body>

<!-- Button element with an ID for JavaScript interaction -->
<button id="sendRequestButton">Send Request</button>

<script>
// Adding an event listener to the button for the 'click' event
document.getElementById("sendRequestButton").addEventListener("click", function() {
  // Creating a new XMLHttpRequest object for making HTTP requests
  var xhr = new XMLHttpRequest();
  // Opening a GET request to the specified URL in asynchronous mode
  xhr.open("GET",
"https://api.thingspeak.com/update?api_key=DF996D42ETNPDB48&field1=0", true);
  // Sending the HTTP request
  xhr.send();
});
</script>

</body>
</html>
```

HTML Code for viewers side:

```
<!DOCTYPE html> <!-- Declaration of the document type -->
<html lang="en"> <!-- Opening of the HTML document with language set to English -->
<head>
  <meta charset="UTF-8"> <!-- Metadata specifying the character encoding of the document -->
  <meta name="viewport" content="width=device-width, initial-scale=1.0"> <!-- Metadata
specifying viewport settings for responsive design -->
  <title>ThingSpeak Value Display</title> <!-- Title of the webpage -->
  <style>
    /* CSS style to increase the font size */
    #lectureDisplay {
      font-size: 20px; /* Adjust the font size as needed */
    }
  </style>
</head>
<body>

<div id="lectureDisplay">Value will be displayed here</div> <!-- Placeholder for displaying the
fetched value -->

<script>
// Function to fetch and update the value
function fetchAndUpdateValue() {
  // Creating an XMLHttpRequest object for making HTTP requests
  var xhr = new XMLHttpRequest();
  // Event listener for changes in the XMLHttpRequest state
  xhr.onreadystatechange = function() {
    // Checking if the request is completed
    if (xhr.readyState === XMLHttpRequest.DONE) {
      // Checking if the response status is OK
      if (xhr.status === 200) {
        // Parsing the response text to an integer
        var value = parseInt(xhr.responseText.trim());
        // Calling the updateDisplay function with the fetched value
        updateDisplay(value);
      } else {
        // Logging an error message if fetching fails
        console.error('Failed to fetch value');
      }
    }
  };
  // Opening a GET request to fetch the value from ThingSpeak
```

```
  xhr.open("GET",
"https://api.thingspeak.com/channels/2461422/fields/1/last.txt?api_key=KNOM94FGO8LB40P3"
, true);
  // Sending the HTTP request
  xhr.send();
}

// Function to update the display based on the fetched value
function updateDisplay(value) {
  // Getting the element with the ID "lectureDisplay"
  var lectureDisplay = document.getElementById("lectureDisplay");
  // Using a switch statement to determine the display text based on the value
  switch (value) {
    case 100:
      lectureDisplay.textContent = "Lecture 1";
      break;
    case 200:
      lectureDisplay.textContent = "Lecture 2";
      break;
    case 300:
      lectureDisplay.textContent = "Lecture 3";
      break;
    case 400:
      lectureDisplay.textContent = "Lecture 4";
      break;
    case 500:
      lectureDisplay.textContent = "Lecture 5";
      break;
    default:
      lectureDisplay.textContent = "Unknown Lecture";
      break;
  }
}

// Fetch value every second
setInterval(fetchAndUpdateValue, 1000); // Setting up an interval to fetch and update the value
every second
</script>

</body>
</html>
```

Algorithm: Fetch and Display ThingSpeak Value

1. Start

2. Define the HTML structure:
   2.1. Declare the document type as HTML5.
   2.2. Open the HTML document with language set to English.
   2.3. Set the character encoding to UTF-8.
   2.4. Define viewport settings for responsive design.
   2.5. Set the title of the webpage.
   2.6. Style the display area to increase font size.
   2.7. Create a div element with an ID "lectureDisplay" to display the fetched value.
   2.8. Include JavaScript code within the <script> tag.

3. Define JavaScript functions:
   3.1. fetchAndUpdateValue():
      3.1.1. Create an XMLHttpRequest object (xhr) for making HTTP requests.
      3.1.2. Set up an event listener for changes in the XMLHttpRequest state.
      3.1.3. When the request state changes:
         - Check if the request is completed.
         - Check if the response status is OK (200).
         - Parse the response text to an integer value.
         - Call the updateDisplay function with the fetched value.
         - Log an error message if fetching fails.
      3.1.4. Open a GET request to fetch the value from ThingSpeak.
      3.1.5. Send the HTTP request.

   3.2. updateDisplay(value):
      3.2.1. Get the element with the ID "lectureDisplay".
      3.2.2. Use a switch statement to determine the display text based on the fetched value.
         - If value is 100, display "Lecture 1".
         - If value is 200, display "Lecture 2".
         - If value is 300, display "Lecture 3".
         - If value is 400, display "Lecture 4".
         - If value is 500, display "Lecture 5".
         - Otherwise, display "Unknown Lecture".

4. Fetch and update value every second:
   4.1. Set an interval to call fetchAndUpdateValue function every 1000 milliseconds (1 second).

5. End