St. Francis Institute of Technology Borivali (West), Mumbai-400103
(Autonomous Institute)
Department of Information Technology
Academic Year: 2024-25

Class: TE-ITA/B Semester: VI

Subject: Web Lab

Experiment –10: To Implement AJAX to Fetch and Display Data Dynamically on User Interaction.

Aim: Implement AJAX to Fetch and Display Data Dynamically on User Interaction.

1. **Objectives:** Aim of this experiment is that, the students will be able
   - To explain readystate property with state and status
   - Read and understand GET and POST method
   - Environment setup
   - To understand how we implement AJAX with other method
2. **Outcomes:** After study of this experiment, the students will be able
   - To Implement AJAX with different method
   - To implement GET and POST method.
   - To handle Data Dynamically on User Interaction.
3. **Prerequisite:** Basic understanding of CSS, HTML, DOM, text editor and execution of programs, Basic understanding of database, MongoDB commands, data types
4. **Requirements:** Personal Computer, Windows operating system, AJAX, browser, Internet Connection, google doc.
5. **Pre-Experiment Exercise:**
   Brief Theory: Refer shared material
6. **Laboratory Exercise**
   A. **Procedure:**
      a. **Answer the following:**
         - What are some of the advantages of AJAX?
         - What is AJAX?
         - Explain how to implement AJAX to fetch data
      b. **Attach screenshots:**
         - AJAX code and output with your own comments.
         - Attach all screenshots.
7. **Post-Experiments Exercise**
   A. **Extended Theory:**
      Nil
   B. **Questions:**
      - What are some features of AJAX?
      - Explain how to implement AJAX to display data dynamically on user interaction.
   C. **Conclusion:**
      - Write what was performed in the experiment.
      - Write the significance of the topic studied in the experiment.
   D. **References:**

      1. https://moodledev.io/docs/4.5/guides/javascript/ajax
      2. https://developer.mozilla.org/en-US/docs/Glossary/AJAX

Tanmay Bhatkar TE INFT A Roll No=>9

Answer the following:

What are some of the advantages of AJAX?

- **Improved User Experience:**
  - No Page Reloads: With AJAX, only specific parts of the page are updated rather than refreshing the entire page. This leads to faster interactions and a smoother experience.
  - Real-time Interactions: You can create more interactive features like auto-suggest, live data updates, and chat applications, without waiting for a full page reload.
- Faster Performance:
  - Reduced Server Load: AJAX reduces the need to reload the entire page, reducing the number of server requests. Only the necessary data is requested, which can improve both client-side and server-side performance.
  - Asynchronous Operations: AJAX requests are handled asynchronously, meaning that the browser can continue other tasks while the server processes the request, resulting in faster response times.
- Bandwidth Efficiency:
  - Partial Data Loading: Since only the necessary data is fetched, rather than reloading a full page, less bandwidth is consumed. This can be a significant advantage, especially on mobile networks.
- Better User Interface:
  - AJAX allows for creating highly interactive elements like updating parts of the page, refreshing data, or loading new content without disturbing the user's current activity.
- Reduced Latency:
  - Faster Updates: Since the page doesn't need to reload, the time between requesting and receiving data can be minimized, resulting in near-instant updates.

What is AJAX?

AJAX (Asynchronous JavaScript and XML) is a technique used in web development to create dynamic and interactive web pages. It allows a web page to request data from a server asynchronously (in the background) without having to reload the entire page. This leads to a smoother and faster user experience as only parts of the web page are updated.

AJAX typically uses JavaScript, XMLHttpRequest (or the Fetch API), and a web server to fetch and send data. It can handle a variety of data formats, including JSON, XML, HTML, and plain text.

Explain how to implement AJAX to fetch data

1. Using the XMLHttpRequest Object:

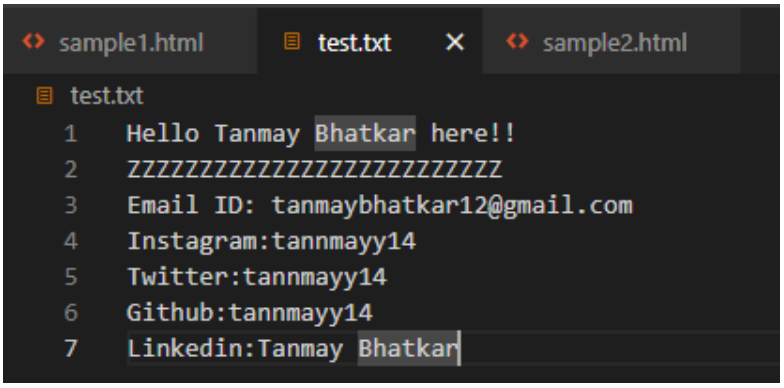
Here’s a basic example using the XMLHttpRequest object.

```
// Step 1: Create a new XMLHttpRequest object
var xhr = new XMLHttpRequest();

// Step 2: Configure it (GET method, URL, asynchronous)
xhr.open("GET", "https://api.example.com/data", true);

// Step 3: Set up a function that will run when the request completes
xhr.onreadystatechange = function() {
   // Step 4: Check if the request is successful (status 200)
   if (xhr.readyState === 4 && xhr.status === 200) {
      // Parse the response data (assuming JSON)
      var data = JSON.parse(xhr.responseText);

      // Do something with the data (e.g., update the DOM)
      console.log(data);
   }
};
// Step 5: Send the request
xhr.send();
```

Lab Exercise:

Test.txt:

Hello Tanmay Bhatkar here!!
ZZZZZZZZZZZZZZZZZZZZZZZZZZ
Email ID: tanmaybhatkar12@gmail.com
Instagram:tannmayy14
Twitter:tannmayy14
Github:tannmayy14
Linkedin:Tanmay Bhatkar



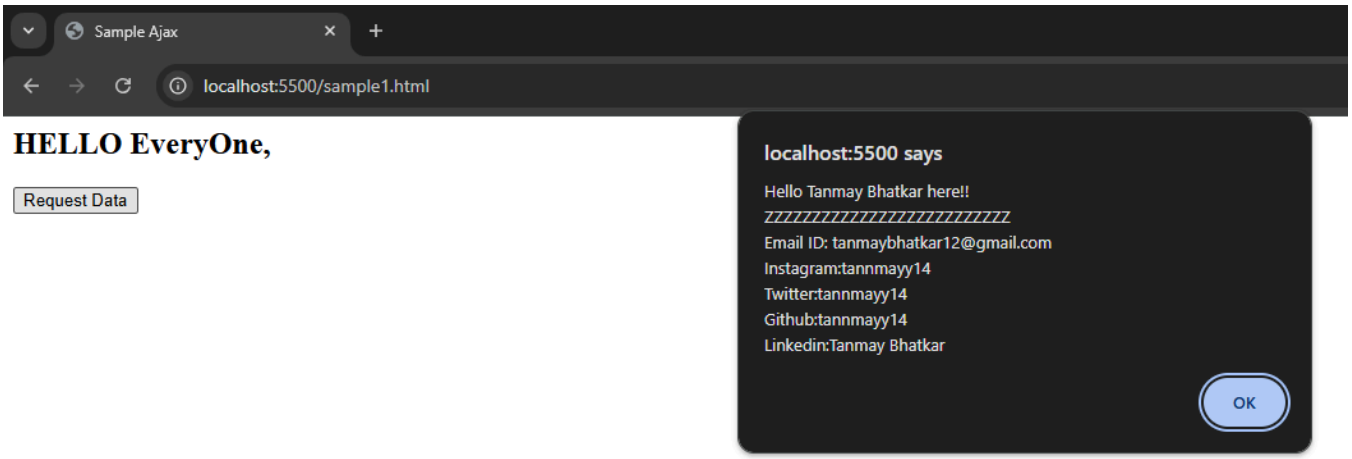
SAMPLE1.html:

```
<html>
  <head>
    <title>Sample Ajax</title>
   <script type="text/javascript"> //This specifies the MIME type (Multipurpose Internet Mail Extensions) of the script.
    var request = new XMLHttpRequest(); //new AJAX request object, which allows us to communicate with a server without reloading the page.
    function requestData() {  //This function is called when the button is clicked.
      request.onload = function () {   //request.onload is triggered when the server responds successfully.
        alert(this.responseText); //use for pop up msg we get responce of request. The response data is displayed in an alert box.
         //document.getElementById("test").innerHTML=this.responseText;

        document.getElementById("txtFileData").innerHTML = this.responseText;  //head.innerHTML = this.responseText;
      }
      request.open('GET', 'test.txt', true); //3 methods GET,POST,PUT.'GET': The request method (fetches data).'test.txt': The file to fetch.true: Asynchronous request (does not block execution).
      request.send(); //This sends the request to fetch test.txt.
    }
    </script>
  </head>
  <body>
  <div id="container">
    <h2 id="heading">HELLO EveryOne,</h2>
        <div id="btnDiv">
      <button class="btn" onclick="requestData()">
        Request Data
      </button>
    </div>
    <br>
    <div id="txtFileData"></div>
  </div>
  </body>
</html>
```

HELLO EveryOne,

Request Data

Hello Tanmay Bhatkar here!! ZZZZZZZZZZZZZZZZZZZZZZZZZZ Email ID: tanmaybhatkar12@gmail.com Instagram:tannmayy14 Twitter:tannmayy14 Github:tannmayy14 Linkedin:Tanmay Bhatkar



Sample2.html

```
<html>
  <head>
    <title>Sample Ajax</title>
        <script type="text/javascript">
     function ajaxfunction()
     {
        var ajax= new XMLHttpRequest();
        ajax.onreadystatechange=function()
        {
          if (this.readyState==4 && this.status==200)
          {
           alert(this.responseText);  //use for pop up msg we get responce of request
           //document.getElementById("test").innerHTML=this.responseText;
          }
        };
          ajax.open("GET","test.txt",true) //3 methods GET,POST,PUT

          ajax.send();
     }
    </script>
  </head>
  <body>
    <button type="button"  onclick="ajaxfunction()">Call Ajax</button>
     <div id="test"></div>
      </body>
</html>
```