| **Experiment No. – 6** | | | | |
| --- | --- | --- | --- | --- |
| **Date of Performance:** | **12/2/25** | | | |
| **Date of Submission:** | **19/3/25** | | | |
| Program Execution/  formation/  correction/  ethical practices  (06) | Timely  Submission  (01) | Viva  (03) | Experiment  Total (10) | Sign with Date |
|  |  |  |  |  |

**6.1Aim :** Web application using AJAX.

**6.2 Course Outcome:** Develop Rich Internet Application using proper choice of Framework.

**6.3 Learning Objectives:** To explore the use of Asynchronous JavaScript and XML (AJAX) in modern web applications, understand how AJAX can be used for partial page updates, and differentiate between synchronous and asynchronous communication in web development.

**6.4 Requirement:**

- Web Browser (Chrome/Firefox)  
- Text Editor (Visual Studio Code / Notepad++)  
- Internet Connection  
- Localhost Server (XAMPP/WAMP or Node.js based server)

**6.5 Theory :**

**Introduction to AJAX:**  
AJAX (Asynchronous JavaScript and XML) is a technique used in web development to create asynchronous web applications. It allows web pages to update asynchronously by exchanging small amounts of data with the server behind the scenes.  
  
**Key Features:**- Sends and retrieves data asynchronously  
- Improves user experience  
- Supports JSON, XML, HTML, and plain text  
  
**Components of AJAX:**1. XMLHttpRequest Object  
2. JavaScript  
3. Server-side Script  
  
**Working of AJAX:**- Event triggers JavaScript to send a request  
- Server processes the request  
- Response is sent and DOM is updated  
  
**Advantages:**- Reduces server traffic  
- Faster page updates  
- Enhanced UX  
  
**Disadvantages:**- Depends on JavaScript  
- SEO limitations

**6.6 Procedure:**

Implementation Steps:  
1. Create folder and open in VS Code  
2. Create index.html and data.php  
3. Use JavaScript to send request via XMLHttpRequest  
4. Handle server response  
5. Display data dynamically on webpage

**6.7 Program and Output:**

data.txt:

Hello! This text was loaded using AJAX without reloading the page.

Index.html:

<!DOCTYPE html>

<html>

<head>

    <title>AJAX Example</title>

    <script>

        function fetchData() {

            // Create a new XMLHttpRequest object

            var xhr = new XMLHttpRequest();

            // Configure it: GET-request for data.txt (a sample server file)

            xhr.open("GET", "data.txt", true); // true = asynchronous

            // What to do when response is ready

            xhr.onreadystatechange = function() {

                if (xhr.readyState == 4 && xhr.status == 200) {

                    // Update the HTML content

                    document.getElementById("output").innerHTML = xhr.responseText;

                }

            };

            // Send the request

            xhr.send();

        }

    </script>

</head>

<body>

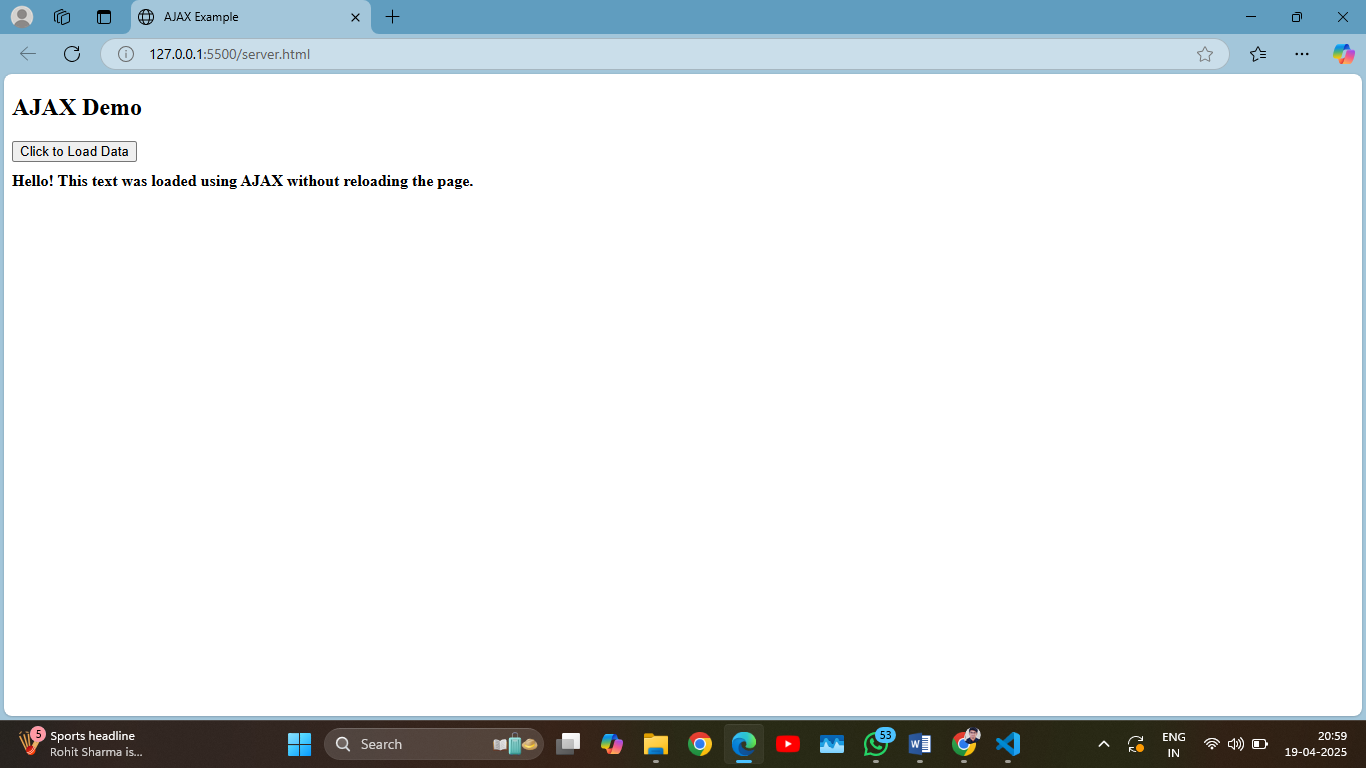
    <h2>AJAX Demo</h2>

    <button onclick="fetchData()">Click to Load Data</button>

    <div id="output" style="margin-top:10px; font-weight:bold;"></div>

</body>

</html>

****

**6.8 Conclusion**

In this experiment, we implemented AJAX-based communication to load server-side content without refreshing the entire webpage. We learned how asynchronous operations enhance user experience and interactivity in web applications.

**6.9 Questions:**

**1. What is AJAX and how does it improve web applications?**  
**Answer:**  
AJAX (Asynchronous JavaScript and XML) is a web development technique used to create dynamic and interactive web applications. It allows web pages to communicate with the server asynchronously without reloading the entire page. This improves user experience by making web apps faster, more responsive, and seamless, as only parts of the page get updated when needed.

**2. What are the main components involved in AJAX communication?**  
**Answer:**  
The main components of AJAX are:

* **HTML/XHTML**: For structure of the web page
* **CSS**: For styling
* **JavaScript**: For handling AJAX logic
* **XMLHttpRequest Object**: Core of AJAX, used to send and receive data from a server
* **DOM (Document Object Model)**: To dynamically update parts of the web page based on server response
* **Server-side script**: Processes the request (e.g., PHP, Node.js, Python)

**3. What is the difference between synchronous and asynchronous requests?**  
**Answer:**

* **Synchronous requests** block the execution of code until the server response is received. This can freeze the webpage and make it unresponsive.
* **Asynchronous requests** allow the browser to continue processing while the request is being handled in the background, improving performance and user experience.