|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Experiment - 06** | | | | |
| **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** |
|  |  |  |  |  |

**Experiment No. 6**

**Title. Web application using ajax**

**1.1 Aim:** To design and develop a web application using AJAX

**1.2 Course Outcome:** Develop rich internet application using different frameworks

**1.3 Learning Objectives:** Understand how AJAX works in the background without refreshing the page.

Create a simple client-server communication using AJAX.

**1.4 Requirement:**

Code editor (e.g., VS Code)

Basic HTTP server (e.g., Live Server or Python SimpleHTTPServer)

1.5 Related Theory:

AJAX (Asynchronous JavaScript and XML) is a technique for creating dynamic and fast web applications by allowing data to be fetched or sent in the background without reloading the full page. AJAX uses JavaScript to send HTTP requests and handle responses from the server.

Key Concepts:

* XMLHttpRequest object (or fetch()) is used to make asynchronous HTTP requests.
* AJAX supports formats like JSON, XML, or plain text.
* Common HTTP methods: GET (fetch data), POST (send data)

**1.6 Procedure:**

1. Create an HTML file with a form or button to trigger an AJAX request.
2. Use JavaScript to send a request to the server using XMLHttpRequest or fetch().
3. On the server-side, create a script (PHP, Node.js, or JSON file) that returns data.
4. Handle the server response in the JavaScript code and update the webpage dynamically.
5. Test the application by running it on a local server.

**1.7 Program and Output:**

**Program :**

index.html

/<!DOCTYPE html>

<html>

<head>

<title>AJAX Web Application</title>

</head>

<body>

<h2>AJAX Example - Get User Info</h2>

<button onclick="loadUser()">Get User Info</button>

<div id="result"></div>

<script>

function loadUser() {

const xhr = new XMLHttpRequest();

xhr.open("GET", "user.json", true);

xhr.onload = function() {

if (this.status === 200) {

const user = JSON.parse(this.responseText);

document.getElementById("result").innerHTML =

`<p><strong>Name:</strong> ${user.name}</p>

<p><strong>Email:</strong> ${user.email}</p>`;

}

};

xhr.send();

}

</script>

</body>

</html>

user.json

json

Copy

Edit

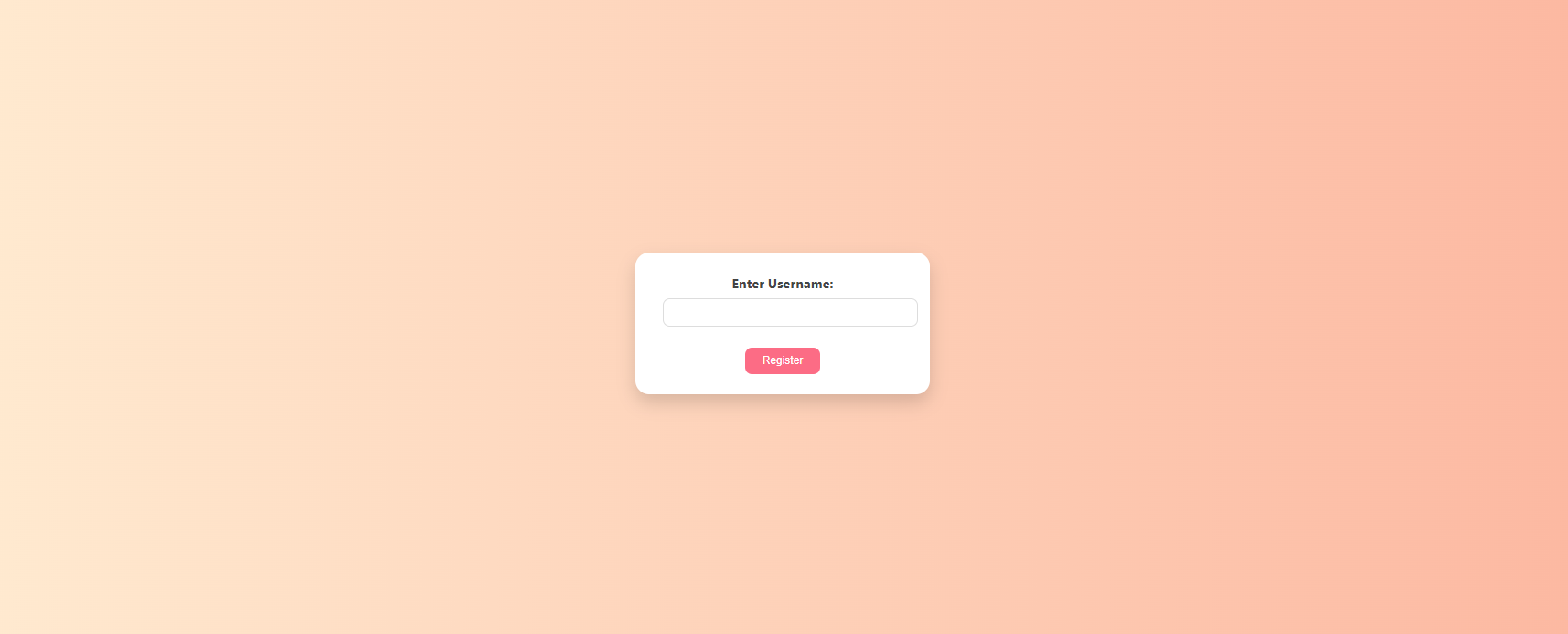
{

"name": "CharudattaThorve",

"email": "charu@example.com"

}

**Output**

****

**1.8 Conclusion:**

The experiment demonstrated how AJAX can be used to fetch data from a server and update the web page without a reload. This technique improves the responsiveness and performance of web applications. Students also learned how to send asynchronous requests using JavaScript and handle JSON responses effectively.

**1.9 Questions:**

**1. What is AJAX?** **Answer:** AJAX (Asynchronous JavaScript and XML) is a technique for creating fast and dynamic web pages by exchanging data with a web server asynchronously, without reloading the page.

**2. What are the main components used in AJAX?** **Answer:**

* JavaScript
* XMLHttpRequest or Fetch API
* HTML/CSS
* A server-side resource (like JSON, PHP, etc.)

**3. What are the advantages of using AJAX?** **Answer:**

* Faster user interaction
* No full page reload
* Better user experience
* Asynchronous data loading