**Practical 6**

**Aim**: To implement a program using Ajax.

**Theory:**

AJAX = Asynchronous JavaScript and XML.

AJAX is not a programming language.

AJAX just uses a combination of:

-A browser built-in XMLHttpRequest object (to request data from a web server)

-JavaScript and HTML DOM (to display or use the data)

AJAX is a misleading name. AJAX applications might use XML to transport data, but it is equally common to transport data as plain text or JSON text.

AJAX allows web pages to be updated asynchronously by exchanging data with a web server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.

1. An event occurs in a web page (the page is loaded, a button is clicked)
2. An XMLHttpRequest object is created by JavaScript
3. The XMLHttpRequest object sends a request to a web server
4. The server processes the request
5. The server sends a response back to the web page
6. The response is read by JavaScript
7. Proper action (like page update) is performed by JavaScript

XMLHttpRequest Object Methods

-abort()

Cancels the current request.

-getAllResponseHeaders()

Returns the complete set of HTTP headers as a string.

-getResponseHeader( headerName )

Returns the value of the specified HTTP header.

-open( method, URL )

open( method, URL, async )

open( method, URL, async, userName )

open( method, URL, async, userName, password )

Specifies the method, URL, and other optional attributes of a request.

The method parameter can have a value of "GET", "POST", or "HEAD". Other

HTTP methods such as "PUT" and "DELETE" (primarily used in REST

applications) may be possible.

-send( content )

Sends the request.

-setRequestHeader( label, value )

Adds a label/value pair to the HTTP header to be sent.

XMLHttpRequest Properties

-onreadystatechange

An event handler for an event that fires at every state change. -readyState

The readyState property defines the current state of the XMLHttpRequest object.

The following table provides a list of the possible values for the readyState property



readyState = 0 After you have created the XMLHttpRequest object, but before you have called the open() method.

readyState = 1 After you have called the open() method, but before you have called send().

readyState = 2 After you have called send().

readyState = 3 After the browser has established a communication with the server, but before the server has completed the response.

readyState = 4 After the request has been completed, and the response data has been completely received from the server. -responseText

Returns the response as a string.

-responseXML

Returns the response as XML. This property returns an XML document object,

which can be examined and parsed using the W3C DOM node tree methods and

properties.

-status

Returns the status as a number (e.g., 404 for "Not Found" and 200 for "OK").

-statusText

Returns the status as a string (e.g., "Not Found" or "OK").

**INPUT**

1. **Program using Ajax for making server request.**

<html>

<body>

<h1>The XMLHttpRequest Object</h1>

<div id="demo">

<button type="button" onclick="loadDoc()">Change Content</button>

</div>

<script>

function loadDoc()

{

var xhttp = new XMLHttpRequest();

xhttp.onreadystatechange = function()

{

if(this.readyState == 4 && this.status == 200)

{

document.getElementById("demo").innerHTML =this.responseText;

}

};

xhttp.open("GET", "data.txt", true);

xhttp.send();

}

</script>

</body>

</html>

**OUTPUT**





**INPUT**

1. **Write a program of Ajax with PHP for displaying the multiplication table of the number entered.**

**multiplication.html**

<html>

<body>

<input type="text" id="num">

<div id="demo">

<button type="button" onclick="loadDoc()">Change Content</button>

</div>

<script>

function loadDoc() {

var num=document.getElementById("num").value; var xhttp = new XMLHttpRequest(); xhttp.onreadystatechange = function() {

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

document.getElementById("demo").innerHTML =this.responseText; } };

xhttp.open("GET", "mul.php?q="+num, true);

xhttp.send(); }

</script>

</body>

</html>

**mul.php**

<?php

$num=$\_GET["q"];

for($i=1;$i<=10;$i++)

{

$res=$num\*$i;

echo $num."X".$i."=".$res."</br>";

}

?>

**OUTPUT**

 