**IMPLEMENTATION OF AJAX, XML, XSL.**

**OBJECTIVE**

To implement and familiarize with Ajax, XML, XSL.

**THEORY**

**Ajax**

Ajax is a set of web development techniques using many web technologies on the client side to create asynchronous web applications. With Ajax, web applications can send and retrieve data from a server asynchronously (in the background) without interfering with the display and behavior of the existing page. By decoupling the data interchange layer from the presentation layer, Ajax allows web pages and, by extension, web applications, to change content dynamically without the need to reload the entire page. In practice, modern implementations commonly utilize JSON instead of XML.

Ajax is not a single technology, but rather a group of technologies. HTML and CSS can be used in combination to mark up and style information. The webpage can then be modified by JavaScript to dynamically display—and allow the user to interact with—the new information. The built-in XMLHttpRequest object, or since 2017 the new "fetch()" function within JavaScript, is commonly used to execute Ajax on webpages allowing websites to load content onto the screen without refreshing the page. Ajax is not a new technology, or different language, just existing technologies used in new ways.

**XML**

Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. The World Wide Web Consortium's XML 1.0 Specification of 1998 and several other related specifications—all of them free open standards—define XML.

The design goals of XML emphasize simplicity, generality, and usability across the Internet. It is a textual data format with strong support via Unicode for different human languages. Although the design of XML focuses on documents, the language is widely used for the representation of arbitrary data structures such as those used in web services.

**XSL**

XSL (extensible Stylesheet Language) is a styling language for XML. XSLT stands for XSL Transformations.

**ACTIVITIES**

First, we setup for xampp. Then we created index.html, script.js, and assets folder with bookstore.xml and bookstore.xsl. In HTML file we link the script and make a button named ‘Get Data’. From the script we open bookstore.xml as request, then the bookstore.xml linked the bookstore.xsl. Finally, we get the data using Ajax, XML and XSL.

**SOURCE CODE**

**HTML:**

<!DOCTYPE html>

<html>

<head>

<title>AJAX Example</title>

<script type="text/javascript" src="script.js"></script>

</head>

<body>

<button id="get-data">Get Data</button>

<div class="result"></div>

</body>

</html>

**JS:**

function getData(){

var xmlHttp = new XMLHttpRequest();

xmlHttp.open('GET','assets/bookstore.xml',true);

xmlHttp.send();

xmlHttp.onreadystatechange = function(){

if(xmlHttp.readyState == 4){

// document.getElementsByClassName('result')[0].innerHTML = xmlHttp.responseXML;

var xmlDoc = xmlHttp.responseXML;

var text ='';

var titles = xmlDoc.getElementsByTagName('title');

for(var i=0; i< titles.length;i++){

text+= titles[i].innerHTML+'</br>';

}

// alert(text);

document.getElementsByClassName('result')[0].innerHTML=text;

}

}

}

function myFunction(){

var button = document.getElementById('get-data');

button.addEventListener('click',getData);

}

document.addEventListener('DOMContentLoaded',myFunction);

**XML**

<?xml version="1.0" encoding="UTF-8"?>

<?xml-stylesheet type="text/xsl" href="bookstore.xsl"?>

<bookstore>

<book>

<title>Web Technology</title>

<price>$100</price>

<author>Krishna Raj Giri</author>

<isbn>1234</isbn>

</book>

<book>

<title>Javascript</title>

<price>$18</price>

<author>Leo Tokyo Lee</author>

<isbn>1235</isbn>

</book>

<book>

<title>Machine Learning</title>

<price>$120</price>

<author>Sung Jung Karki</author>

<isbn>1236</isbn>

</book>

</bookstore>

**XSL**

<?xml version="1.0" encoding="UTF-8"?>

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">

<html>

<body>

<h2>Book List</h2>

<table border="2">

<tr>

<td>Title</td>

<td>Price</td>

</tr>

<xsl:for-each select="bookstore/book">

<tr>

<td><xsl:value-of select="title"/></td>

<td><xsl:value-of select="price"/></td>

</tr>

</xsl:for-each>

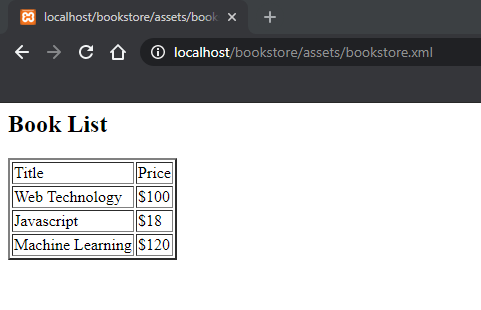
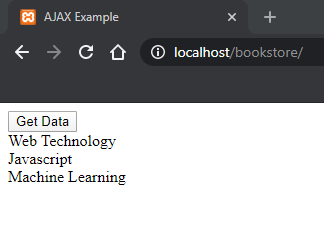
</table>

</body>

</html>

</xsl:template></xsl:stylesheet>

**OUTPUT**

****

**SUMMARY/PROBLEM FACED**

We are able to get data using Ajax, XML and XSL.

The main problem faced was to link the files between them itself.

**LAB INSTRUCTOR COMMENTS**