|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Session** | **2024-25 (ODD)** | | **Course Name** | **Web Technology Lab** | |
| **Semester** | **3** | | **Course Code** | **23CT1301** | |
| **Roll No** | **53** | | **Name of Student** | **Kartik Sheshraoji Tale** | |
|  |  | |  |  |  |
| Practical Number | | 6 | | | |
| Course Outcome | | 1. Understand various internet technologies. 2. Design the web pages using HTML and CSS. 3. Implement the XML technology to store the data. 4. Develop the interactive web pages using JavaScript. | | | |
| Aim | | Parsing a file cd\_catalog.xml and use the response as an XML DOM  objects and extracts the info from it with JavaScript. | | | |
| Problem Definition | | Parse an external XML file (cd\_catalog.xml) using JavaScript, convert it into an XML DOM, extract <CD> details, and display them in an HTML table. | | | |
| Theory  (100 words) | | XML (Extensible Markup Language) is a way to store and share data in a structured format. It is easy to read by both humans and machines. To use XML in a webpage, we can parse it into an XML DOM (Document Object Model). The DOM lets JavaScript read and work with different parts of the XML file, like elements and values. We can load the XML file using XMLHttpRequest and then access data using functions such as getElementsByTagName(). Finally, the extracted information can be shown in a proper HTML table, making the data clear and user-friendly | | | |
| Procedure and Execution  (100 Words) | | Step for Implementation:   1. Create cd\_catalog.xml with CD details. 2. Save HTML file (index.html) with JavaScript code to load and parse XML. 3. Use XMLHttpRequest to fetch and process XML into DOM. 4. Extract data using getElementsByTagName and build an HTML table. 5. Run project on a local server (e.g., Python http.server or VS Code Live Server). 6. Open in browser → Click **Load Catalog** → View table with CD info. | | | |
| Code:  Index.html :  <!DOCTYPE html>  <html lang="en">  <head>    <meta charset="UTF-8">    <title>CD Catalog</title>    <style>      body {        font-family: Arial, sans-serif;        text-align: center;        margin: 20px;      }      table {        border-collapse: collapse;        width: 80%;        margin: 20px auto;      }      table, th, td {        border: 1px solid black;      }      th, td {        padding: 10px;        text-align: center;      }      th {        background-color: #f2f2f2;      }      button {        padding: 10px 20px;        background: #4CAF50;        border: none;        color: white;        cursor: pointer;        border-radius: 5px;      }      button:hover {        background: #45a049;      }    </style>  </head>  <body>    <h2>CD Catalog from XML</h2>    <button onclick="loadXMLDoc()">Load Catalog</button>    <br><br>    <div id="tableContainer"></div>    <script>      function loadXMLDoc() {        let xhttp = new XMLHttpRequest();        xhttp.onreadystatechange = function() {          if (this.readyState === 4 && this.status === 200) {            cdDetails(this);          }        };        xhttp.open("GET", "cd\_catalog.xml", true);        xhttp.send();      }      function cdDetails(xml) {        let xmlDoc = xml.responseXML;        let cds = xmlDoc.getElementsByTagName("CD");        let table = "<table><tr><th>Title</th><th>Artist</th><th>Country</th><th>Company</th><th>Price</th><th>Year</th></tr>";        for (let i = 0; i < cds.length; i++) {          table += "<tr>" +            "<td>" + cds[i].getElementsByTagName("TITLE")[0].childNodes[0].nodeValue + "</td>" +            "<td>" + cds[i].getElementsByTagName("ARTIST")[0].childNodes[0].nodeValue + "</td>" +            "<td>" + cds[i].getElementsByTagName("COUNTRY")[0].childNodes[0].nodeValue + "</td>" +            "<td>" + cds[i].getElementsByTagName("COMPANY")[0].childNodes[0].nodeValue + "</td>" +            "<td>" + cds[i].getElementsByTagName("PRICE")[0].childNodes[0].nodeValue + "</td>" +            "<td>" + cds[i].getElementsByTagName("YEAR")[0].childNodes[0].nodeValue + "</td>" +            "</tr>";        }        table += "</table>";        document.getElementById("tableContainer").innerHTML = table;      }    </script>  </body>  </html>  cd\_catalog.xml :  <?xml version="1.0" encoding="UTF-8"?>  <CATALOG>    <CD>      <TITLE>Empire Burlesque</TITLE>      <ARTIST>Bob Dylan</ARTIST>      <COUNTRY>USA</COUNTRY>      <COMPANY>Columbia</COMPANY>      <PRICE>10.90</PRICE>      <YEAR>1985</YEAR>    </CD>    <CD>      <TITLE>Hide your heart</TITLE>      <ARTIST>Bonnie Tyler</ARTIST>      <COUNTRY>UK</COUNTRY>      <COMPANY>CBS Records</COMPANY>      <PRICE>9.90</PRICE>      <YEAR>1988</YEAR>    </CD>    <CD>      <TITLE>Greatest Hits</TITLE>      <ARTIST>Dolly Parton</ARTIST>      <COUNTRY>USA</COUNTRY>      <COMPANY>RCA</COMPANY>      <PRICE>9.90</PRICE>      <YEAR>1982</YEAR>    </CD>  </CATALOG> | | | |
| Output: | | | |
| Output Analysis | | When the **Load Catalog** button is clicked, the XML file is read by JavaScript. The program extracts CD details like title, artist, country, company, price, and year. These details are displayed in an HTML table. The output shows that XML data can be parsed and shown clearly on a webpage. | | | |
| Link of student Github profile where lab assignment has been uploaded | | <https://github.com/kartiktale12/WebTechnology> | | | |
| Conclusion | | This experiment shows how XML data can be read and parsed using JavaScript. The CD details from the XML file are extracted and displayed in a clear HTML table. It proves that XML DOM and JavaScript can be used to present external data effectively on a web page. | | | |
| Plag Report (Similarity index < 12%) | | 10% | | | |
| Date | | 8/09/2025 | | | |