WT Assignment 3

Neeti Kurulkar

XML document with DTD, Schema and XSL.

Code:

1. Index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Employee Data</title>

<script>

*function* loadXML() {

*let* xhr = *new* XMLHttpRequest();

xhr.open("GET", "employees.xml", *true*);

xhr.responseType = "document"; // Ensure XML response

xhr.overrideMimeType("text/xml"); // Force XML format

xhr.onreadystatechange = *function* () {

*if* (xhr.readyState === 4) {

*if* (xhr.status === 200) {

*let* xml = xhr.responseXML;

*if* (!xml) {

console.error("XML failed to load.");

*return*;

}

console.log("XML loaded:", xml);

// Load XSL file

*let* xsl = *new* XMLHttpRequest();

xsl.open("GET", "employees.xsl", *true*);

xsl.responseType = "document"; // Ensure XSL response

xsl.overrideMimeType("text/xml");

xsl.onreadystatechange = *function* () {

*if* (xsl.readyState === 4) {

*if* (xsl.status === 200) {

*let* xslDoc = xsl.responseXML;

*if* (!xslDoc) {

console.error("XSL failed to load.");

*return*;

}

console.log("XSL loaded:", xslDoc);

// Apply XSLT transformation

*let* xsltProcessor = *new* XSLTProcessor();

xsltProcessor.importStylesheet(xslDoc);

*let* resultDocument = xsltProcessor.transformToFragment(

xml,

document

);

document.getElementById("output").innerHTML = ""; // Clear previous content

document

.getElementById("output")

.appendChild(resultDocument);

} *else* {

console.error("Error loading XSL:", xsl.status);

}

}

};

xsl.send();

} *else* {

console.error("Error loading XML:", xhr.status);

}

}

};

xhr.send();

}

</script>

</head>

<body onload="loadXML()">

<div id="output"></div>

</body>

</html>

1. Employees.dtd

<!ELEMENT employees (employee+)>

<!ELEMENT employee (empName, DOB, contactno, address, gender, designation, Dateofjoin, salary, department)>

<!ELEMENT empName (fname, lname)>

<!ELEMENT fname (#PCDATA)>

<!ELEMENT lname (#PCDATA)>

<!ELEMENT DOB (#PCDATA)>

<!ELEMENT contactno (#PCDATA)>

<!ELEMENT address (#PCDATA)>

<!ELEMENT gender (#PCDATA)>

<!ELEMENT designation (#PCDATA)>

<!ELEMENT Dateofjoin (#PCDATA)>

<!ELEMENT salary (#PCDATA)>

<!ELEMENT department (#PCDATA)>

1. Exmployees.xsd

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

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="employees">

<xs:complexType>

<xs:sequence>

<xs:element name="employee" maxOccurs="unbounded">

<xs:complexType>

<xs:sequence>

<xs:element name="empName">

<xs:complexType>

<xs:sequence>

<xs:element name="fname" type="xs:string"/>

<xs:element name="lname" type="xs:string"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="DOB" type="xs:date"/>

<xs:element name="contactno" type="xs:string"/>

<xs:element name="address" type="xs:string"/>

<xs:element name="gender">

<xs:simpleType>

<xs:restriction base="xs:string">

<xs:enumeration value="Male"/>

<xs:enumeration value="Female"/>

</xs:restriction>

</xs:simpleType>

</xs:element>

<xs:element name="designation" type="xs:string"/>

<xs:element name="Dateofjoin" type="xs:date"/>

<xs:element name="salary" type="xs:decimal"/>

<xs:element name="department" type="xs:string"/>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:schema>

1. Employees.xml

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

<!DOCTYPE employees SYSTEM "employees.dtd">

<employees>

<employee>

<empName>

<fname>John</fname>

<lname>Doe</lname>

</empName>

<DOB>1990-05-21</DOB>

<contactno>9876543210</contactno>

<address>123, ABC Street, New York</address>

<gender>Male</gender>

<designation>Software Engineer</designation>

<Dateofjoin>2015-06-01</Dateofjoin>

<salary>75000.50</salary>

<department>IT</department>

</employee>

<employee>

<empName>

<fname>Jane</fname>

<lname>Smith</lname>

</empName>

<DOB>1992-08-15</DOB>

<contactno>9123456789</contactno>

<address>456, XYZ Avenue, California</address>

<gender>Female</gender>

<designation>Data Scientist</designation>

<Dateofjoin>2018-09-20</Dateofjoin>

<salary>85000.75</salary>

<department>Data Science</department>

</employee>

<employee>

<empName>

<fname>Michael</fname>

<lname>Johnson</lname>

</empName>

<DOB>1985-12-10</DOB>

<contactno>8765432109</contactno>

<address>789, LMN Road, Texas</address>

<gender>Male</gender>

<designation>Project Manager</designation>

<Dateofjoin>2010-04-15</Dateofjoin>

<salary>95000.00</salary>

<department>Management</department>

</employee>

<employee>

<empName>

<fname>Emily</fname>

<lname>Williams</lname>

</empName>

<DOB>1995-03-25</DOB>

<contactno>9234567890</contactno>

<address>321, PQR Lane, Florida</address>

<gender>Female</gender>

<designation>UX Designer</designation>

<Dateofjoin>2019-07-10</Dateofjoin>

<salary>70000.25</salary>

<department>Design</department>

</employee>

<employee>

<empName>

<fname>David</fname>

<lname>Brown</lname>

</empName>

<DOB>1988-07-30</DOB>

<contactno>9012345678</contactno>

<address>654, STU Street, Illinois</address>

<gender>Male</gender>

<designation>Cybersecurity Analyst</designation>

<Dateofjoin>2014-11-05</Dateofjoin>

<salary>88000.60</salary>

<department>Cybersecurity</department>

</employee>

</employees>

1. Employees.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>

<head>

<title>Employee Details</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

text-align: center;

}

table {

width: 80%;

margin: 20px auto;

border-collapse: collapse;

box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.2);

background: #ffffff;

}

th, td {

border: 1px solid #ddd;

padding: 12px;

text-align: left;

}

th {

background: #4CAF50;

color: white;

}

tr:nth-child(even) {

background: #f2f2f2;

}

tr:hover {

background: #ddd;

cursor: pointer;

}

</style>

</head>

<body>

<h2>Employee Details</h2>

<table border="1">

<tr>

<th>First Name</th>

<th>Last Name</th>

<th>DOB</th>

<th>Contact No</th>

<th>Address</th>

<th>Gender</th>

<th>Designation</th>

<th>Date of Joining</th>

<th>Salary</th>

<th>Department</th>

</tr>

<xsl:for-each select="employees/employee">

<tr>

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

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

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

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

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

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

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

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

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

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

</tr>

</xsl:for-each>

</table>

</body>

</html>

</xsl:template>

</xsl:stylesheet>

Output:

A screenshot of a computer

AI-generated content may be incorrect.