

Internet and Web Programming

Lab Assignment-6

-17BCE2416

Navya RG

Write a program to perform data storage and parsing using XML and Schema Validation.

Employees.xml:

```
<?xml version="1.0"?>

<employees
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="Employees.xsd">

  <employee>

    <name>

      <fname>Jane</fname>

      <lname>Doe</lname>

    </name>

    <dob>1999-07-19</dob>

    <age>20</age>

    <email>jane.doe@xmail.com</email>

    <mbi>9876543210</mbi>

    <designation>Java Developer</designation>

    <experience>3.4</experience>

  </employee>

  <employee>

    <name>

      <fname>John</fname>

      <lname>Smith</lname>

    </name>

    <dob>1994-03-24</dob>

    <age>25</age>

    <email>john.smith@xmail.com</email>
```

<mb1>9121212121</mb1>

<designation>Senior Technical Consultant</designation>

<experience>6.2</experience>

</employee>

<employee>

<name>

<fname>Alice</fname>

<lname>Pinkman</lname>

</name>

<dob>1990-08-02</dob>

<age>33</age>

<email>alice.pinkman@xmail.com</email>

<mb1>9898989898</mb1>

<designation>General Manager</designation>

<experience>10</experience>

</employee>

<employee>

<name>

<fname>Bob</fname>

<lname>Driscoll</lname>

</name>

<dob>1991-10-16</dob>

<age>32</age>

<email>bob.driscoll@xmail.com</email>

<mb1>9999999999</mb1>

<designation>Chief Executive Officer</designation>

<experience>9</experience>

</employee>

</employees>

Employees.xsd:

```
<?xml version="1.0"?>
<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="name">
                <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="age">
                <xs:simpleType>
                  <xs:restriction base="xs:integer">
                    <xs:minInclusive value="18"/>
                    <xs:maxInclusive value="65"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:element name="email">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:pattern value="[a-z][a-z0-9_\.]*(@)[a-z]*(\.)([a-z])*"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

<xs:element name="mbl">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[789][0-9]{9}" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>

<xs:element name="designation">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Intern Consulting" />
      <xs:enumeration value="Java Developer" />
      <xs:enumeration value="Python Developer" />
      <xs:enumeration value="UI/UX Designer" />
      <xs:enumeration value="Database Support" />
      <xs:enumeration value="Technical Consultant" />
      <xs:enumeration value="Senior Technical Consultant" />
      <xs:enumeration value="Lead Technical Consultant" />
      <xs:enumeration value="Branch Manager" />
      <xs:enumeration value="General Manager" />
      <xs:enumeration value="Chief Executive Officer" />
      <xs:enumeration value="Chief Operations Officer" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>

<xs:element name="experience">
  <xs:simpleType>
    <xs:restriction base="xs:decimal">
      <xs:minInclusive value="0.0" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>

```

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:schema>

Online XML Parser output:

The screenshot shows a web browser window with the URL `freeformatter.com/xml-validator-xsd.html`. The page has a dark header with the site name and navigation links. A sidebar on the left lists various tools under categories like Validators, Encoders & Decoders, Code Minifiers / Beautifier, Converters, Cryptography & Security, and String Escaper & Utilities. The main content area displays a green message: "The XML document is valid." with a "Close" button. Below this, there are sections for "XML Input" and "XSD Input (Optional if XSD referred in XML using schemaLocation)". The XML Input section shows two options: "Option 1: Copy-paste your XML document here" and "Option 2: Or upload your XML document". The XSD Input section also shows two options: "Option 1: Copy-paste your XSD document here" and "Option 2: Or upload your XSD document". A "VALIDATE XML" button is visible at the bottom of the input sections. The XML document being validated is:

```
<?xml version='1.0'?>
<employees
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xsi:schemaLocation='Employees.xsd'>
  <employee>
    <name>
```

index.html:

<!DOCTYPE html>

<html>

<head>

<title>Data storage and parsing using XML and Schema Validation</title>

<link rel="stylesheet" type="text/css" href="style.css">

</head>

<body>

<button onclick="display()">Display data</button>

<p id="demo"></p>

<script type="text/javascript">

function display(){

var xhttp = new XMLHttpRequest();

xhttp.onreadystatechange = function(){

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

myFunction(this);

}

```

    };
    xhttp.open("GET", "Employees.xml", true);
    xhttp.send();

    function myFunction(xml) {
        var xmlDoc = xml.responseXML;
        var table="<br><table><tr><th>First name</th><th>Last
name</th><th>DOB</th><th>Age</th><th>E-
mail</th><th>Phone</th><th>Designation</th><th>Experience</th></tr>";
        for(var i=0; i<4; i++){
            var fname = xmlDoc.getElementsByTagName("fname")[i].childNodes[0].nodeValue;
            var lname = xmlDoc.getElementsByTagName("lname")[i].childNodes[0].nodeValue;
            var dob = xmlDoc.getElementsByTagName("dob")[i].childNodes[0].nodeValue;
            var age = xmlDoc.getElementsByTagName("age")[i].childNodes[0].nodeValue;
            var email = xmlDoc.getElementsByTagName("email")[i].childNodes[0].nodeValue;
            var mbl = xmlDoc.getElementsByTagName("mbl")[i].childNodes[0].nodeValue;
            var designation =
xmlDoc.getElementsByTagName("designation")[i].childNodes[0].nodeValue;
            var experience =
xmlDoc.getElementsByTagName("experience")[i].childNodes[0].nodeValue;

            table+="<tr><td>"+fname+"</td><td>"+lname+"</td><td>"+dob+"</td><td>"+age+"</td><td>"+em
ail+"</td><td>"+mbl+"</td><td>"+designation+"</td><td>"+experience+"</td></tr>";
        }
        table+="</table>";
        document.getElementById("demo").innerHTML = table;
    }
}
</script>
</body>
</html>

```

