

Lab Exercise 5

Name: Jethro Jarvis Roy Jyrwa

RollNo: 2447122

1 MCA A

Web Stack Development

Case Study: Transforming XML Data with XSL and Validating with XSD

Given the scenario, I have stored the data for my domain Boston College which is a higher education institute. The Data in the xml format is as follows:

mainCollege.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-model href="college.xsd" type="application/xml"
schematypens="http://www.w3.org/2001/XMLSchema"?>
<?xml-stylesheet href="transform.xsl" type="text/xsl"?>
<College
  xmlns="http://www.bostoncollege.edu.us/college"
  xmlns:bcu="http://www.bostoncollege.edu.us/college"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.bostoncollege.edu.us/college
college.xsd">

  <Administrator AdminID="A001">
    <Prefix>Dr.</Prefix>
    <First_Name>John</First_Name>
    <Middle_Name>H.</Middle_Name>
    <Last_Name>Simon</Last_Name>
    <Phone_No>9878932813</Phone_No>
    <Email>john.simon@bostoncollege.edu.us</Email>
    <Designation>Principal</Designation>
    <Qualification>PhD</Qualification>
    <Date_of_Appointment>2015-08-01</Date_of_Appointment>
    <Password>admin123</Password>
  </Administrator>

  <Department DepartmentID="D001">
    <HoD_ID>T001</HoD_ID>
    <Department_Email>cse@college.edu.us</Department_Email>
    <No_of_Faculty>20</No_of_Faculty>
    <Dept_Library>true</Dept_Library>
    <Location>Building A</Location>
    <Year_of_Estb>2000-01-01</Year_of_Estb>
    <Dept_Website>http://www.bostoncollege.edu.us/cse</Dept_Website>
```

```
<No_of_Research_Papers>100</No_of_Research_Papers>
</Department>

<Programme ProgrammeID="P001">
  <ProgrammeName>Bachelor of Science in Computer Science</ProgrammeName>
  <DepartmentID>D001</DepartmentID>
  <No_of_Students>200</No_of_Students>
  <No_of_Faculty>15</No_of_Faculty>
  <Degree>UG</Degree>
  <No_of_Years>3</No_of_Years>
  <System>Semester</System>
  <Max_Credits>180</Max_Credits>
</Programme>

<Student RollNo="S001">
  <First_Name>Jane</First_Name>
  <Middle_Name>Q.</Middle_Name>
  <Last_Name>Smith</Last_Name>
  <Gender>Female</Gender>
  <Phone_No>9876543210</Phone_No>
  <Email>jane.smith@student.bostoncollege.edu.us</Email>
  <DepartmentID>D001</DepartmentID>
  <ProgrammeID>P001</ProgrammeID>
  <Attendance>85.5</Attendance>
  <Password>student123</Password>
</Student>

<Teacher TeacherID="T001">
  <First_Name>Robert</First_Name>
  <Middle_Name>K.</Middle_Name>
  <Last_Name>Brown</Last_Name>
  <Gender>Male</Gender>
  <Phone_No>2345678901</Phone_No>
  <Email>robert.brown@bostoncollege.edu.us</Email>
  <DepartmentID>D001</DepartmentID>
  <Qualification>PhD</Qualification>
  <NET_Qualified>Yes</NET_Qualified>
  <Designation>Professor</Designation>
  <Date_of_Joining>2012-09-15</Date_of_Joining>
  <Password>teacher123</Password>
</Teacher>

<Admission RegistrationNo="R001">
  <First_Name>Emily</First_Name>
  <Middle_Name>L.</Middle_Name>
  <Last_Name>Johnson</Last_Name>
  <Phone_No>3456789012</Phone_No>
  <Email>emily.johnson@gmail.com</Email>
```

```
<Class10_Perc>90.5</Class10_Perc>
<Class11_Perc>85.0</Class11_Perc>
<Class12_Perc>88.0</Class12_Perc>
<CUET_Score>250.5</CUET_Score>
<ProgrammeID>P001</ProgrammeID>
<Admission_status>Granted</Admission_status>
</Admission>

<Club CCC_ID="C001">
  <CCC_Name>Coding Club</CCC_Name>
  <CCC_Agenda>Promote coding skills</CCC_Agenda>
  <Convener_ID>T001</Convener_ID>
  <President_Name>Jane Smith</President_Name>
  <Secretary_Name>Emily Johnson</Secretary_Name>
  <Treasurer_Name>Robert Brown</Treasurer_Name>
  <Date_of_Estb>2018-02-15</Date_of_Estb>
  <No_of_Members>50</No_of_Members>
</Club>

<Library BookID="B001">
  <Category>Science</Category>
  <DepartmentID>D001</DepartmentID>
  <Borrowable_Status>true</Borrowable_Status>
  <Number_of_copies>5</Number_of_copies>
  <Number_of_Pages>300</Number_of_Pages>
  <Author1>Christopher Maine</Author1>
  <Author2>Jane Smith</Author2>
  <ISBN>978-3-16-148410-0</ISBN>
</Library>

<Research PublicationID="Re001">
  <PublicationName>AI in Education</PublicationName>
  <PublicationSynopsis>A study on the impact of AI on
education</PublicationSynopsis>
  <ResearchArea>Artificial Intelligence</ResearchArea>
  <AuthorID1>T001</AuthorID1>
  <EtAl_Status>true</EtAl_Status>
  <Link>http://www.research.edu/ai-education</Link>
  <Verified_Status>true</Verified_Status>
</Research>

<Alumni AlumniID="AL001">
  <RollNo>S002</RollNo>
  <DepartmentID>D001</DepartmentID>
  <ProgrammeID>P001</ProgrammeID>
  <Year_of_Graduation>2020</Year_of_Graduation>
  <Alum_Email>alex@alumni.bostoncollege.edu.us</Alum_Email>
```

```

        <Alum_Phone_No>9876543210</Alum_Phone_No>
        <Highest_Qualification>MSc</Highest_Qualification>
        <Current_Occupation>Software Engineer</Current_Occupation>
        <Testimonial>The college provided excellent education and
opportunities.</Testimonial>
    </Alumni>

</College>

```

To ensure that the data adheres to a predefined XML schema for validation purposes, the following XSD file has been created specifying each entity as a complexType which has a sequence of elements. Each element has a specific data type and there are also elements which contain specific values within a range or enumeration, done through a restrictionBase. Elements which are of restrictionBase are enclosed as a simpleType. For example: the **Degree** element in the **Programme** entity is a **simpleType** which has a **restriction** of type string containing values in enumerations, namely, "UG","PG","PhD".

The purpose of an XSD file is to provide the data types and domains for the various data which will be entered in the XML file as well as the sequence in which various elements in the XML appear. The data entered in the XML file has to match the specifications given in the corresponding XSD file.

college.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:bcu="http://www.bostoncollege.edu.us/college"
  targetNamespace="http://www.bostoncollege.edu.us/college"
  elementFormDefault="qualified">

  <xs:element name="College" type="bcu:College"></xs:element>

  <xs:complexType name="College">
    <xs:sequence>
      <xs:element name="Administrator" type="bcu:Administrator"
maxOccurs="unbounded"/>
      <xs:element name="Department" type="bcu:Department"
maxOccurs="unbounded"/>
      <xs:element name="Programme" type="bcu:Programme"
maxOccurs="unbounded"/>
      <xs:element name="Student" type="bcu:Student"
maxOccurs="unbounded"/>
      <xs:element name="Teacher" type="bcu:Teacher"
maxOccurs="unbounded"/>
      <xs:element name="Admission" type="bcu:Admission"
maxOccurs="unbounded"/>
      <xs:element name="Club" type="bcu:Club"
maxOccurs="unbounded"/>
    
```

```

        <xs:element name="Library" type="bcu:Library"
maxOccurs="unbounded"/>
        <xs:element name="Research" type="bcu:Research"
maxOccurs="unbounded"/>
        <xs:element name="Alumni" type="bcu:Alumni"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>

<!-- Administrator Element -->
<xs:complexType name="Administrator">
    <xs:sequence>
        <xs:element name="Prefix" type="xs:string"/>
        <xs:element name="First_Name" type="xs:string"/>
        <xs:element name="Middle_Name" type="xs:string"
minOccurs="0"/>
        <xs:element name="Last_Name" type="xs:string"/>
        <xs:element name="Phone_No">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:length value="10"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="Email" type="xs:string"/>
        <xs:element name="Designation" type="xs:string"/>
        <xs:element name="Qualification" type="xs:string"/>
        <xs:element name="Date_of_Appointment" type="xs:date"/>
        <xs:element name="Password" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="AdminID" type="xs:ID" use="required"/>
</xs:complexType>

<xs:complexType name="Department">
    <xs:sequence>
        <xs:element name="HoD_ID" type="xs:string"/>
        <xs:element name="Department_Email" type="xs:string"/>
        <xs:element name="No_of_Faculty" type="xs:integer"/>
        <xs:element name="Dept_Library" type="xs:boolean"/>
        <xs:element name="Location" type="xs:string"/>
        <xs:element name="Year_of_Estb" type="xs:date"/>
        <xs:element name="Dept_Website" type="xs:string"/>
        <xs:element name="No_of_Research_Papers" type="xs:integer"/>
    </xs:sequence>
    <xs:attribute name="DepartmentID" type="xs:ID" use="required"/>
</xs:complexType>

<!-- Programme Element -->

```

```

<xs:complexType name="Programme">
  <xs:sequence>
    <xs:element name="ProgrammeName" type="xs:string"/>
    <xs:element name="DepartmentID" type="xs:string"/>
    <xs:element name="No_of_Students" type="xs:integer"/>
    <xs:element name="No_of_Faculty" type="xs:integer"/>
    <xs:element name="Degree">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="UG"/>
          <xs:enumeration value="PG"/>
          <xs:enumeration value="PhD"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="No_of_Years" type="xs:string"/>
    <xs:element name="System">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="Semester"/>
          <xs:enumeration value="Trimester"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Max_Credits" type="xs:integer"/>
  </xs:sequence>
  <xs:attribute name="ProgrammeID" type="xs:ID" use="required"/>
</xs:complexType>

```

```

<xs:complexType name="Student">
  <xs:sequence>
    <xs:element name="First_Name" type="xs:string"/>
    <xs:element name="Middle_Name" type="xs:string"
minOccurs="0"/>
    <xs:element name="Last_Name" type="xs:string"/>
    <xs:element name="Gender" type="xs:string"/>
    <xs:element name="Phone_No">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:length value="10"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Email" type="xs:string"/>
    <xs:element name="DepartmentID" type="xs:string"/>
    <xs:element name="ProgrammeID" type="xs:string"/>
    <xs:element name="Attendance">

```

```
        <!-- Restriction: Attendance percentage must be between 0  
and 100 -->
```

```
        <xs:simpleType>  
            <xs:restriction base="xs:float">  
                <xs:minInclusive value="0"/>  
                <xs:maxInclusive value="100"/>  
            </xs:restriction>  
        </xs:simpleType>  
    </xs:element>  
    <xs:element name="Password" type="xs:string"/>  
</xs:sequence>  
    <xs:attribute name="RollNo" type="xs:ID" use="required"/>  
</xs:complexType>
```

```
<xs:complexType name="Teacher">  
    <xs:sequence>  
        <xs:element name="First_Name" type="xs:string"/>  
        <xs:element name="Middle_Name" type="xs:string"  
minOccurs="0"/>  
        <xs:element name="Last_Name" type="xs:string"/>  
        <xs:element name="Gender" type="xs:string"/>  
        <xs:element name="Phone_No">  
            <xs:simpleType>  
                <xs:restriction base="xs:string">  
                    <xs:length value="10"/>  
                </xs:restriction>  
            </xs:simpleType>  
        </xs:element>  
        <xs:element name="Email" type="xs:string"/>  
        <xs:element name="DepartmentID" type="xs:string"/>  
        <xs:element name="Qualification" type="xs:string"/>  
        <xs:element name="NET_Qualified">  
            <xs:simpleType>  
                <xs:restriction base="xs:string">  
                    <xs:enumeration value="Yes"/>  
                    <xs:enumeration value="No"/>  
                </xs:restriction>  
            </xs:simpleType>  
        </xs:element>  
        <xs:element name="Designation" type="xs:string"/>  
        <xs:element name="Date_of_Joining" type="xs:date"/>  
        <xs:element name="Password" type="xs:string"/>  
    </xs:sequence>  
    <xs:attribute name="TeacherID" type="xs:ID" use="required"/>  
</xs:complexType>
```

```
<!-- Admission Element -->  
<xs:complexType name="Admission">
```

```

        <xs:sequence>
            <xs:element name="First_Name" type="xs:string"/>
            <xs:element name="Middle_Name" type="xs:string"
minOccurs="0"/>
            <xs:element name="Last_Name" type="xs:string"/>
            <xs:element name="Phone_No">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:length value="10"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element name="Email" type="xs:string"/>
            <xs:element name="Class10_Perc">
                <xs:simpleType>
                    <xs:restriction base="xs:float">
                        <xs:minInclusive value="0"/>
                        <xs:maxInclusive value="100"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element name="Class11_Perc" type="xs:float"
minOccurs="0"/>
            <xs:element name="Class12_Perc">
                <xs:simpleType>
                    <xs:restriction base="xs:float">
                        <xs:minInclusive value="0"/>
                        <xs:maxInclusive value="100"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element name="CUET_Score">
                <xs:simpleType>
                    <xs:restriction base="xs:float">
                        <xs:minInclusive value="0"/>
                        <xs:maxInclusive value="300"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element name="ProgrammeID" type="xs:string"/>
            <xs:element name="Admission_status">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="Granted"/>
                        <xs:enumeration value="Pending"/>
                        <xs:enumeration value="Rejected"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:schema>

```



```

        </xs:element>
    </xs:sequence>
    <xs:attribute name="RegistrationNo" type="xs:ID" use="required"/>
</xs:complexType>

<!-- Club/Cell/Committee Element -->
<xs:complexType name="Club">
    <xs:sequence>
        <xs:element name="CCC_Name" type="xs:string"/>
        <xs:element name="CCC_Agenda" type="xs:string"/>
        <xs:element name="Convener_ID" type="xs:string"/>
        <xs:element name="Co_Convener_ID" type="xs:string"
minOccurs="0"/>
        <xs:element name="President_Name" type="xs:string"
minOccurs="0"/>
        <xs:element name="Secretary_Name" type="xs:string"
minOccurs="0"/>
        <xs:element name="Treasurer_Name" type="xs:string"
minOccurs="0"/>
        <xs:element name="Date_of_Estb" type="xs:date"/>
        <xs:element name="No_of_Members" type="xs:integer"/>
    </xs:sequence>
    <xs:attribute name="CCC_ID" type="xs:ID" use="required"/>
</xs:complexType>

<!-- Library Element -->
<xs:complexType name="Library">
    <xs:sequence>
        <xs:element name="Category" type="xs:string"/>
        <xs:element name="DepartmentID" type="xs:string"/>
        <xs:element name="Borrowable_Status" type="xs:boolean"/>
        <xs:element name="Number_of_copies" type="xs:integer"/>
        <xs:element name="Number_of_Pages" type="xs:integer"/>
        <xs:element name="Author1" type="xs:string"/>
        <xs:element name="Author2" type="xs:string" minOccurs="0"/>
        <xs:element name="Author3" type="xs:string" minOccurs="0"/>
        <xs:element name="ISBN" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="BookID" type="xs:ID" use="required"/>
</xs:complexType>

<!-- Research Element -->
<xs:complexType name="Research">
    <xs:sequence>
        <xs:element name="PublicationName" type="xs:string"/>
        <xs:element name="PublicationSynopsis" type="xs:string"/>
        <xs:element name="ResearchArea" type="xs:string"/>
        <xs:element name="AuthorID1" type="xs:string"/>
    </xs:sequence>
</xs:complexType>

```

```

        <xs:element name="AuthorID2" type="xs:string" minOccurs="0"/>
        <xs:element name="AuthorID3" type="xs:string" minOccurs="0"/>
        <xs:element name="EtAl_Status" type="xs:boolean"/>
        <xs:element name="Link" type="xs:string"/>
        <xs:element name="Verified_Status" type="xs:boolean"/>
    </xs:sequence>
    <xs:attribute name="PublicationID" type="xs:ID" use="required"/>
</xs:complexType>

<!-- Alumni Element -->
<xs:complexType name="Alumni">
    <xs:sequence>
        <xs:element name="RollNo" type="xs:string"/>
        <xs:element name="DepartmentID" type="xs:string"/>
        <xs:element name="ProgrammeID" type="xs:string"/>
        <xs:element name="Year_of_Graduation" type="xs:integer"/>
        <xs:element name="Alum_Email" type="xs:string"/>
        <xs:element name="Alum_Phone_No">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:length value="10"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="Highest_Qualification" type="xs:string"/>
        <xs:element name="Current_Occupation" type="xs:string"/>
        <xs:element name="Testimonial" type="xs:string"
minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="AlumniID" type="xs:ID" use="required"/>
</xs:complexType>

</xs:schema>

```

The XSL stylesheet which performs data transformation from the unreadable xml format to a readable format in HTML. The output is displayed is a HTML table.

The purpose of the XSL file is to provide a template using HTML tags and CSS styles into which the XML data will be loaded into so that it is in a human readable format.

transform.xsl

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0"
    xmlns:bcu="http://www.bostoncollege.edu.us/college">
    <xsl:output method="html" indent="yes"/>

```

```

<xsl:template match="bcu:College">
  <html>
    <head>
      <title>Boston College</title>
      <style>
        table {
          width: 100%;
          border-collapse: collapse;
        }
        th, td {
          border: 1px solid black;
          padding: 8px;
          text-align: left;
        }
        th {
          background-color: #f2f2f2;
        }
      </style>
    </head>
    <body>
      <h1>Boston College</h1>

      <h2>Administrators</h2>
      <table>
        <tr>
          <th>AdminID</th>
          <th>Prefix</th>
          <th>First Name</th>
          <th>Middle Name</th>
          <th>Last Name</th>
          <th>Phone No</th>
          <th>Email</th>
          <th>Designation</th>
          <th>Qualification</th>
          <th>Date of Appointment</th>
        </tr>
        <xsl:for-each select="bcu:Administrator">
          <tr>
            <td><xsl:value-of select="@AdminID" /></td>
            <td><xsl:value-of select="bcu:Prefix" /></td>
            <td><xsl:value-of select="bcu:First_Name" /></td>
            <td><xsl:value-of select="bcu:Middle_Name" /></td>
            <td><xsl:value-of select="bcu:Last_Name" /></td>
            <td><xsl:value-of select="bcu:Phone_No" /></td>
            <td><xsl:value-of select="bcu:Email" /></td>
            <td><xsl:value-of select="bcu:Designation" /></td>

```

```

                <td><xsl:value-of select="bcu:Qualification"
/></td>
                <td><xsl:value-of select="bcu:Date_of_Appointment"
/></td>
            </tr>
        </xsl:for-each>
    </table>

    <h2>Students</h2>
    <table>
        <tr>
            <th>RollNo</th>
            <th>First Name</th>
            <th>Middle Name</th>
            <th>Last Name</th>
            <th>Gender</th>
            <th>Phone No</th>
            <th>Email</th>
            <th>DepartmentID</th>
            <th>ProgrammeID</th>
            <th>Attendance</th>
        </tr>
        <xsl:for-each select="bcu:Student">
            <tr>
                <td><xsl:value-of select="@RollNo" /></td>
                <td><xsl:value-of select="bcu:First_Name" /></td>
                <td><xsl:value-of select="bcu:Middle_Name" /></td>
                <td><xsl:value-of select="bcu:Last_Name" /></td>
                <td><xsl:value-of select="bcu:Gender" /></td>
                <td><xsl:value-of select="bcu:Phone_No" /></td>
                <td><xsl:value-of select="bcu:Email" /></td>
                <td><xsl:value-of select="bcu:DepartmentID"
/></td>
                <td><xsl:value-of select="bcu:ProgrammeID" /></td>
                <td><xsl:value-of select="bcu:Attendance" /></td>
            </tr>
        </xsl:for-each>
    </table>

    <h2>Teachers</h2>
    <table>
        <tr>
            <th>TeacherID</th>
            <th>First Name</th>
            <th>Middle Name</th>
            <th>Last Name</th>
            <th>Gender</th>
            <th>Phone No</th>

```

```

        <th>Email</th>
        <th>DepartmentID</th>
        <th>Qualification</th>
        <th>NET Qualified</th>
        <th>Designation</th>
        <th>Date of Joining</th>
    </tr>
    <xsl:for-each select="bcu:Teacher">
        <tr>
            <td><xsl:value-of select="@TeacherID"/></td>
            <td><xsl:value-of select="bcu:First_Name"/></td>
            <td><xsl:value-of select="bcu:Middle_Name"/></td>
            <td><xsl:value-of select="bcu:Last_Name"/></td>
            <td><xsl:value-of select="bcu:Gender"/></td>
            <td><xsl:value-of select="bcu:Phone_No"/></td>
            <td><xsl:value-of select="bcu:Email"/> </td>
            <td><xsl:value-of select="bcu:DepartmentID"/>
        </td>

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

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

            <td><xsl:value-of select="bcu:Designation"/> </td>
            <td><xsl:value-of select="bcu:Date_of_Joining"/>
        </td>

        </tr>
    </xsl:for-each>
</table>

<h2>Departments</h2>
<table>
    <tr>
        <th>DepartmentID</th>
        <th>HoD_ID</th>
        <th>Department Email</th>
        <th>No of Faculty</th>
        <th>Dept Library</th>
        <th>Location</th>
        <th>Year of Establishment</th>
        <th>Dept Website</th>
        <th>No of Research Papers</th>
    </tr>
    <xsl:for-each select="bcu:Department">
        <tr>
            <td><xsl:value-of select="@DepartmentID" /></td>
            <td><xsl:value-of select="bcu:HoD_ID" /></td>
            <td><xsl:value-of select="bcu:Department_Email"
/></td>

```

```

        <td><xsl:value-of select="bcu:No_of_Faculty"
/></td>
        <td><xsl:value-of select="bcu:Dept_Library"
/></td>
        <td><xsl:value-of select="bcu:Location" /></td>
        <td><xsl:value-of select="bcu:Year_of_Estb"
/></td>
        <td><xsl:value-of select="bcu:Dept_Website"
/></td>
        <td><xsl:value-of
select="bcu:No_of_Research_Papers" /></td>
    </tr>
</xsl:for-each>
</table>

<h2>Programmes</h2>
<table>
    <tr>
        <th>ProgrammeID</th>
        <th>Programme Name</th>
        <th>DepartmentID</th>
        <th>No of Students</th>
        <th>No of Faculty</th>
        <th>Degree</th>
        <th>No of Years</th>
        <th>System</th>
        <th>Max Credits</th>
    </tr>
    <xsl:for-each select="bcu:Programme">
        <tr>
            <td><xsl:value-of select="@ProgrammeID" /></td>
            <td><xsl:value-of select="bcu:ProgrammeName"
/></td>
            <td><xsl:value-of select="bcu:DepartmentID"
/></td>
            <td><xsl:value-of select="bcu:No_of_Students"
/></td>
            <td><xsl:value-of select="bcu:No_of_Faculty"
/></td>
            <td><xsl:value-of select="bcu:Degree" /></td>
            <td><xsl:value-of select="bcu:No_of_Years" /></td>
            <td><xsl:value-of select="bcu:System" /></td>
            <td><xsl:value-of select="bcu:Max_Credits" /></td>
        </tr>
    </xsl:for-each>
</table>

<h2>Admissions</h2>

```

```

<table>
  <tr>
    <th>Registration No</th>
    <th>First Name</th>
    <th>Middle Name</th>
    <th>Last Name</th>
    <th>Phone No</th>
    <th>Email</th>
    <th>Class 10 Percentage</th>
    <th>Class 11 Percentage</th>
    <th>Class 12 Percentage</th>
    <th>CUET Score</th>
    <th>ProgrammeID</th>
    <th>Admission Status</th>
  </tr>
  <xsl:for-each select="bcu:Admission">
    <tr>
      <td><xsl:value-of select="@RegistrationNo" /></td>
      <td><xsl:value-of select="bcu:First_Name" /></td>
      <td><xsl:value-of select="bcu:Middle_Name" /></td>
      <td><xsl:value-of select="bcu:Last_Name" /></td>
      <td><xsl:value-of select="bcu:Phone_No" /></td>
      <td><xsl:value-of select="bcu:Email" /></td>
      <td><xsl:value-of select="bcu:Class10_Perc" /></td>
      <td><xsl:value-of select="bcu:Class11_Perc" /></td>
      <td><xsl:value-of select="bcu:Class12_Perc" /></td>
      <td><xsl:value-of select="bcu:CUET_Score" /></td>
      <td><xsl:value-of select="bcu:ProgrammeID" /></td>
      <td><xsl:value-of select="bcu:Admission_status" /></td>
    </tr>
  </xsl:for-each>
</table>

<!-- Display Club Data -->
<h2>Clubs</h2>
<table>
  <tr>
    <th>CCC_ID</th>
    <th>CCC Name</th>
    <th>CCC Agenda</th>
    <th>Convener ID</th>
    <th>President Name</th>
    <th>Secretary Name</th>
    <th>Treasurer Name</th>
  </tr>

```

```

        <th>Date of Establishment</th>
        <th>No of Members</th>
    </tr>
    <xsl:for-each select="bcu:Club">
        <tr>
            <td><xsl:value-of select="@CCC_ID" /></td>
            <td><xsl:value-of select="bcu:CCC_Name" /></td>
            <td><xsl:value-of select="bcu:CCC_Agenda" /></td>
            <td><xsl:value-of select="bcu:Convener_ID" /></td>
            <td><xsl:value-of select="bcu:President_Name"
/></td>

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

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

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

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

        </tr>
    </xsl:for-each>
</table>

<h2>Library</h2>
<table>
    <tr>
        <th>BookID</th>
        <th>Category</th>
        <th>DepartmentID</th>
        <th>Borrowable Status</th>
        <th>Number of Copies</th>
        <th>Number of Pages</th>
        <th>Author1</th>
        <th>Author2</th>
        <th>ISBN</th>
    </tr>
    <xsl:for-each select="bcu:Library">
        <tr>
            <td><xsl:value-of select="@BookID" /></td>
            <td><xsl:value-of select="bcu:Category" /></td>
            <td><xsl:value-of select="bcu:DepartmentID"
/></td>

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

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

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

```



```

        <td><xsl:value-of select="bcu:Author1" /></td>
        <td><xsl:value-of select="bcu:Author2" /></td>
        <td><xsl:value-of select="bcu:ISBN" /></td>
    </tr>
</xsl:for-each>
</table>

<h2>Research</h2>
<table>
    <tr>
        <th>PublicationID</th>
        <th>Publication Name</th>
        <th>Publication Synopsis</th>
        <th>Research Area</th>
        <th>AuthorID1</th>
        <th>EtAl Status</th>
        <th>Link</th>
        <th>Verified Status</th>
    </tr>
    <xsl:for-each select="bcu:Research">
        <tr>
            <td><xsl:value-of select="@PublicationID" /></td>
            <td><xsl:value-of select="bcu:PublicationName"
/></td>
            <td><xsl:value-of select="bcu:PublicationSynopsis"
/></td>
            <td><xsl:value-of select="bcu:ResearchArea"
/></td>
            <td><xsl:value-of select="bcu:AuthorID1" /></td>
            <td><xsl:value-of select="bcu:EtAl_Status" /></td>
            <td><xsl:value-of select="bcu:Link" /></td>
            <td><xsl:value-of select="bcu:Verified_Status"
/></td>
        </tr>
    </xsl:for-each>
</table>

<h2>Alumni</h2>
<table>
    <tr>
        <th>AlumniID</th>
        <th>RollNo</th>
        <th>DepartmentID</th>
        <th>ProgrammeID</th>
        <th>Year of Graduation</th>
        <th>Alumni Email</th>
        <th>Alumni Phone No</th>
        <th>Highest Qualification</th>
    </tr>

```

```

        <th>Current Occupation</th>
        <th>Testimonial</th>
    </tr>
    <xsl:for-each select="bcu:Alumni">
        <tr>
            <td><xsl:value-of select="@AlumniID" /></td>
            <td><xsl:value-of select="bcu:RollNo" /></td>
            <td><xsl:value-of select="bcu:DepartmentID"
/></td>
            <td><xsl:value-of select="bcu:ProgrammeID" /></td>
            <td><xsl:value-of select="bcu:Year_of_Graduation"
/></td>
            <td><xsl:value-of select="bcu:Alum_Email" /></td>
            <td><xsl:value-of select="bcu:Alum_Phone_No"
/></td>
            <td><xsl:value-of
select="bcu:Highest_Qualification" /></td>
            <td><xsl:value-of select="bcu:Current_Occupation"
/></td>
            <td><xsl:value-of select="bcu:Testimonial" /></td>
        </tr>
    </xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

The XSL document transforms the data in the XML file into a readable format In HTML, however we must also validate the data that it matches that given in the XSD file. To do this the following program is used for both transformation and validation:

validate.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Boston College</title>

    <script src="xmllint.js"> </script>
</head>
<body>
    <div id="output"></div>

    <script>

```

```

function loadXMLDoc(filename) {
    return new Promise(function(resolve, reject) {
        var xhttp = new XMLHttpRequest();
        xhttp.onreadystatechange = function() {
            if (this.readyState == 4 && this.status == 200) {
                resolve(this.responseText);
            } else if (this.readyState == 4) {
                reject(new Error('Error loading document: ' +
filename));
            }
        };
        xhttp.open("GET", filename, true);
        xhttp.send();
    });
}

function transformXML(xml, xslt) {
    if (window.XSLTProcessor) {
        var xsltProcessor = new XSLTProcessor();
        xsltProcessor.importStylesheet(xslt);
        var resultDocument = xsltProcessor.transformToFragment(xml,
document);
        document.getElementById("output").appendChild(resultDocument);
    } else {
        alert("XSLTProcessor not supported in this browser.");
    }
}

function validateXML(xmlStr, xsdStr) {
    var result = xmllint.validateXML({ xml: xmlStr, schema: xsdStr });
    if (result.errors) {
        console.log("XML is invalid.", result.errors);
        alert("XML is invalid: " + result.errors.join("\n"));
    } else {
        console.log("XML is valid.");
        alert("XML is valid.");
    }
}

async function processXML() {
    try {
        var xmlStr = await loadXMLDoc("mainCollege.xml");
        console.log("Loaded XML:", xmlStr);
        var xsdStr = await loadXMLDoc("college.xsd");
        console.log("Loaded XSD:", xsdStr);
        var xslStr = await loadXMLDoc("transform.xsl");
        console.log("Loaded XSLT:", xslStr);
    }
}

```

```

        validateXML(xmlStr, xsdStr);

        var parser = new DOMParser();
        var xml = parser.parseFromString(xmlStr, "application/xml");
        var xsl = parser.parseFromString(xslStr, "application/xml");

        transformXML(xml, xsl);
    } catch (error) {
        console.error("Error processing XML:", error.message);
    }
}

window.onload = function() {
    processXML();
};
</script>
</body>
</html>

```

The above script is used to transform and validate the data in our XML file against our XSD file.

The **loadXMLDoc** is used to load the data in an XML file using XMLHttpRequest (XHR). First the file is opened using GET method and the file is sent to the XHR object. The function returns Promise object which can either be resolved or reject. If the XHR object is in the 200 status and ready state 4 then the Promise object is resolved and the responseText is displayed else if the Promise is not in the 200 status and only in the ready state 4 then it is Rejected and an error is displayed.

The **transformXML** function performs data transformation on an XML document using a provided XSL stylesheet. It uses the XSLTProcessor interface available in modern web browsers to apply the stylesheet and transform the XML.

The **validateXML** function validates an XML document against an XML Schema Definition (XSD) to ensure it conforms to the rules specified in the XSD. This function uses the xmllint library which is included in the same directory as a js file; which provides XML parsing and validation capabilities.

The **processXML** function is an asynchronous function designed to handle XML processing tasks, which calls the user defined functions to load, transform and validate the XML data. The processXML function is called whenever the file is loaded.

Various scenarios where the solution is tested is given below through screenshots:

1. A scenario where the phone number is more than 10 digits as we have restricted the length to 10 digits. We get the following alert:

127.0.0.1:5500 says

XML is invalid: file_0.xml:15: element Phone_No: Schemas validity error : Element '{http://www.bostoncollege.edu.us/college}Phone_No': [facet 'length'] The value has a length of '13'; this differs from the allowed length of '10'.
file_0.xml:15: element Phone_No: Schemas validity error : Element '{http://www.bostoncollege.edu.us/college}Phone_No': '9878932813123' is not a valid value of the local atomic type.

OK

2. A scenario where we enter letters in an element which has a date datatype. We get an alert:

127.0.0.1:5500 says

XML is invalid: file_0.xml:19: element Date_of_Appointment: Schemas validity error : Element '{http://www.bostoncollege.edu.us/college}Date_of_Appointment': '2015-08-01DOFE' is not a valid value of the atomic type 'xs:date'.

OK

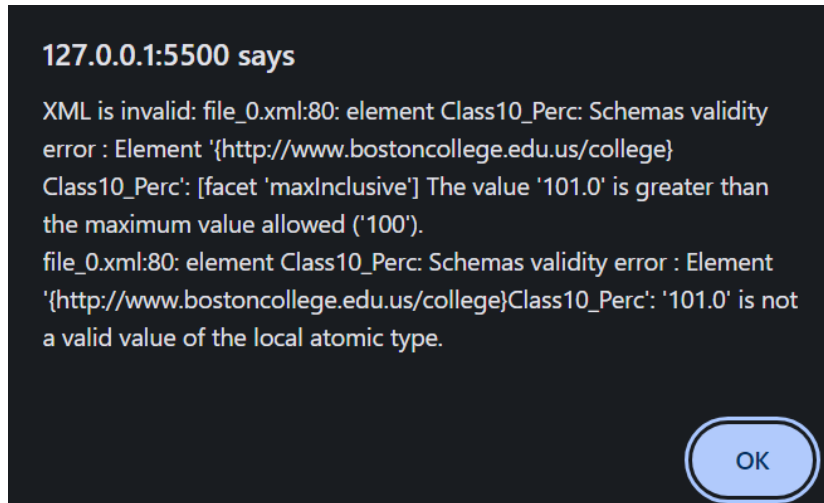
3. A scenario where we enter a value not specified in an enumeration. For example, in the System element of the Programme entity, the enumerated value are semester and trimester. I will enter "Annual" in the XML Data and get the following alert:

127.0.0.1:5500 says

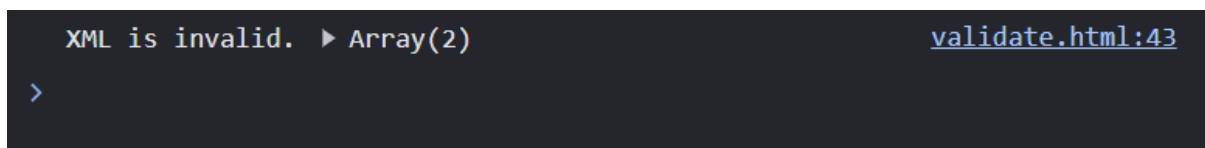
XML is invalid: file_0.xml:41: element System: Schemas validity error : Element '{http://www.bostoncollege.edu.us/college}System': [facet 'enumeration'] The value 'Annual' is not an element of the set {'Semester', 'Trimester'}.
file_0.xml:41: element System: Schemas validity error : Element '{http://www.bostoncollege.edu.us/college}System': 'Annual' is not a valid value of the local atomic type.

OK

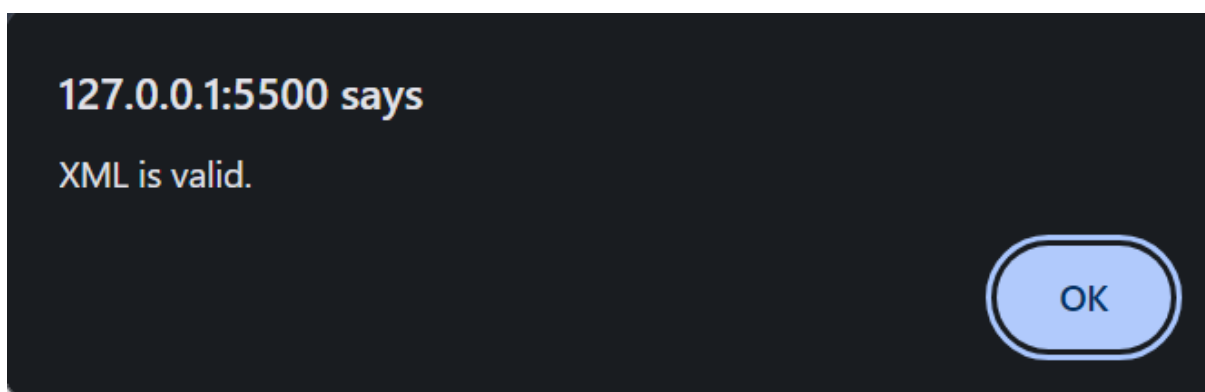
4. A scenario where we enter a value which is not within a specified range. In the Admission entity, the Class10_Perc element has a range from 0 to 100. If we enter any value outside this year we get the following alert:



The above scenarios have successfully tested the validation process. Whenever there is incorrect data, the console will display the following message:



The Final Output after the XML data has been transformed and validated is as follows:



Boston College

Administrators

AdminID	Prefix	First Name	Middle Name	Last Name	Phone No	Email	Designation	Qualification	Date of Appointment
A001	Dr.	John	H.	Simon	9878932813	john.simon@bostoncollege.edu.us	Principal	PhD	2015-08-01

Students

RollNo	First Name	Middle Name	Last Name	Gender	Phone No	Email	DepartmentID	ProgrammeID	Attendance
S001	Jane	Q.	Smith	Female	9876543210	jane.smith@student.bostoncollege.edu.us	D001	P001	85.5

Teachers

TeacherID	First Name	Middle Name	Last Name	Gender	Phone No	Email	DepartmentID	Qualification	NET Qualified	Designation	Date of Joining
T001	Robert	K.	Brown	Male	2345678901	robert.brown@bostoncollege.edu.us	D001	PhD	Yes	Professor	2012-09-15

Departments

DepartmentID	HoD_ID	Department Email	No of Faculty	Dept Library	Location	Year of Establishment	Dept Website	No of Research Papers
D001	T001	cse@college.edu.us	20	true	Building A	2000-01-01	http://www.bostoncollege.edu.us/cse	100

Programmes

ProgrammeID	Programme Name	DepartmentID	No of Students	No of Faculty	Degree	No of Years	System	Max Credits
P001	Bachelor of Science in Computer Science	D001	200	15	UG	3	Semester	180

Admissions

Registration No	First Name	Middle Name	Last Name	Phone No	Email	Class 10 Percentage	Class 11 Percentage	Class 12 Percentage	CUET Score	ProgrammeID	Admission Status
R001	Emily	L.	Johnson	3456789012	emily.johnson@gmail.com	90.5	85.0	88.0	250.5	P001	Granted

Clubs

CCC_ID	CCC Name	CCC Agenda	Convener ID	President Name	Secretary Name	Treasurer Name	Date of Establishment	No of Members
C001	Coding Club	Promote coding skills	T001	Jane Smith	Emily Johnson	Robert Brown	2018-02-15	50

Library

BookID	Category	DepartmentID	Borrowable Status	Number of Copies	Number of Pages	Author1	Author2	ISBN
B001	Science	D001	true	5	300	Christopher Maine	Jane Smith	978-3-16-148410-0

Research

PublicationID	Publication Name	Publication Synopsis	Research Area	AuthorID1	EtAI Status	Link	Verified Status
Re001	AI in Education	A study on the impact of AI on education	Artificial Intelligence	T001	true	http://www.research.edu/ai-education	true

Alumni

AlumniID	RollNo	DepartmentID	ProgrammeID	Year of Graduation	Alumni Email	Alumni Phone No	Highest Qualification	Current Occupation	Testimonial
AL001	S002	D001	P001	2020	alex@alumni.bostoncollege.edu.us	9876543210	MSc	Software Engineer	The college provided excellent education and opportunities.
