WEB LAB-5

2447255

Documentation

- **Transformation and Validation Process**

- **Purpose**

  XSL Stylesheet: The XSLT stylesheet transforms the provided XML data into a user-friendly HTML page. This HTML page is intended to present the location data in a structured and readable format.

  XSD Schema: The XSD schema defines the structure and constraints of the XML data. It ensures that XML documents conform to a specific format and validates that elements meet defined criteria, such as data types and formats.

- **Components**

1. XSL Stylesheet (locations.xsl): Transforms the XML data into HTML.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
        xmlns:html="http://www.w3.org/1999/xhtml">

  <xsl:template match="/locations">
    <html:html>
      <html:head>
        <html:title>Locations</html:title>
        <html:style>
          body {
            font-family: Arial, sans-serif;
            margin: 20px;
            padding: 0;
      color: #333;
    }
    .location {
      border: 1px solid #ddd;
      padding: 10px;
      margin-bottom: 10px;
      border-radius: 5px;
    }
    .location h2 {
      margin: 0;
      font-size: 1.5em;
    }
    .contact, .hours {
      margin-top: 10px;
    }
    .contact p, .hours p {
      margin: 0;
    }
  </html:style>
</html:head>
<html:body>
  <h1>Locations</h1>
  <xsl:for-each select="location">
    <div class="location">
      <h2><xsl:value-of select="name"/></h2>
      <p><strong>Address:</strong> <xsl:value-of select="address"/></p>
      <div class="contact">
        <p><strong>Phone:</strong> <xsl:value-of select="contact/phone"/></p>
        <p><strong>Email:</strong> <xsl:value-of select="contact/email"/></p>
      </div>
      <div class="hours">
```

```
              <p><strong>Weekday Hours:</strong> <xsl:value-of
select="hours/weekday"/></p>

              <p><strong>Weekend Hours:</strong> <xsl:value-of
select="hours/weekend"/></p>

            </div>

          </div>

        </xsl:for-each>

      </html:body>

    </html:html>

  </xsl:template>


</xsl:stylesheet>
```

2. XSD Schema (locations.xsd): Validates the XML data structure and content.

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

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

      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

      xsi:schemaLocation="http://www.w3.org/2001/XMLSchema locations.xsd"

      elementFormDefault="qualified">


  <!-- Define the structure for the 'locations' element -->

  <xs:element name="locations">

    <xs:complexType>

      <xs:sequence>

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

          <xs:complexType>

            <xs:sequence>

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

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

              <xs:element name="contact">

                <xs:complexType>
```

```
                  <xs:sequence>
                    <xs:element name="phone">
                      <xs:simpleType>
                        <xs:restriction base="xs:string">
                          <xs:pattern value="\d{10}"/>
                        </xs:restriction>
                      </xs:simpleType>
                    </xs:element>
                    <xs:element name="email" type="xs:string"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="hours">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="weekday" type="xs:string"/>
                    <xs:element name="weekend" type="xs:string"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
            <xs:attribute name="id" type="xs:integer" use="required"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

- **Validation Scenarios and Issues**

1. Valid XML Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<locations xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:noNamespaceSchemaLocation="locations.xsd">
  <location id="1">
    <name>GymX</name>
    <address>123 Fitness St, Health City</address>
    <contact>
      <phone>1237412984</phone>
      <email>contact@gymx.com</email>
    </contact>
    <hours>
      <weekday>6:00 AM - 10:00 PM</weekday>
      <weekend>8:00 AM - 8:00 PM</weekend>
    </hours>
  </location>
</locations>
```

Result: Passes validation and transformation. Generates an HTML page with the location details correctly formatted.

2. Invalid XML Example (Phone Format Error):

```
<?xml version="1.0" encoding="UTF-8"?>
<locations xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:noNamespaceSchemaLocation="locations.xsd">
  <location id="1">
    <name>GymX</name>
    <address>123 Fitness St, Health City</address>
    <contact>
      <phone>123-4567-890</phone>
      <email>contact@gymx.com</email>
    </contact>
    <hours>
      <weekday>6:00 AM - 10:00 PM</weekday>
      <weekend>8:00 AM - 8:00 PM</weekend>
    </hours>
  </location>
</locations>
```

Validation Issue: Fails schema validation due to the phone number format not matching 1234567890.

3. Invalid XML Example (Missing Required Attribute):

```
<?xml version="1.0" encoding="UTF-8"?>
<locations xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="locations.xsd">
  <location
    <name>ParkFit</name>
    <address>456 Wellness Rd, Fit Town</address>
    <contact>
      <phone>123-456-7890</phone>
      <email>info@parkfit.com</email>
    </contact>
    <hours>
      <weekday>7:00 AM - 9:00 PM</weekday>
      <weekend>9:00 AM - 7:00 PM</weekend>
    </hours>
  </location>
```

```
</locations>
```

Validation Issue: Fails schema validation due to the missing id attribute in the location element.

- **Tools Used:**

1. XML Editor: Tools like Oxygen XML Editor or online XML validators were used to validate the XML against the XSD schema.
2. Web Browser: Used for viewing the HTML transformation result by applying the XSLT stylesheet to the XML data.

- **Documentation:**

1. XSL Stylesheet: Defines how XML data should be transformed into HTML for user-friendly presentation.
2. XSD Schema: Defines the structure, data types, and constraints for the XML data to ensure correctness and validity.

- **Validation Process**:

  XML documents were validated against the XSD schema to ensure compliance with defined rules and constraints. Errors and issues were identified and corrected based on schema validation feedback.