

CSE3002 Internet and Web Programming
Lab Assessment 6
XML Validation

Submitted by
Patel Vishva Pravinkumar (17BCE0940)
November 1, 2019

Problem Description

Create an XML document with the following

Root Element: real-estate

sub-elements for real-estate: agencies, owners, properties and flats

sub-elements contain the attributes (Specify appropriate attributes of your own with datatypes)

Agency = name, e-mail, phone

Owner = name

Property = name, description, number of flats, features, owner reference

Flat = name, description, features, property reference

- **Ensure well-formedness, ensure its validity by checking with DTD and XML Schema**
- **Use XSL to apply the style with the HTML template for the elements: Owner & Property Address only**
- Use attributes for references, otherwise elements
- Use only CDATA and #PCDATA for content and values

Document Formation

1. Code

- a. XML File (ast6.xml)
- b. DTD File (ast6.dtd)
- c. XML Schema File (ast6.xsd)
- d. XSL Transformation (to HTML) file (ast6.xsl)

2. Output Screenshots

- a. XML file view on Internet Explorer

3. Well Formedness of ast6.xml - screenshots

- a. notepad++
- b. w3school online validator

4. Validation of ast6.xml - screenshots

- a. XML DTD Validation
- b. XML Schema Validation

XML File (ast6.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="ast6.xsl"?>
<!DOCTYPE note SYSTEM "ast6.dtd" >
<!-- root element -->
<real-estate>

<!-- agencies is a child element of root-->
<agencies>
<agency>
<name>Patel Builders</name>
<email>vishvapatel652@gmail.com</email>
<phone>9003362230</phone>
</agency>

<agency>
<name>Vishva Developers</name>
<email>vishvap@gmail.com</email>
<phone>9106848042</phone>
</agency>

</agencies>

<!-- owners is a child element of root -->
<owners>
<owner>
<name>Vishva Patel</name>
</owner>
<owner>
<name>S Rama</name>
</owner>
<owner>
<name>Anand Vishva</name>
</owner>
</owners>

<!-- properties is a child element of root -->
<properties>
<property owner="Vishva Patel"> <!-- owner reference for property -->
<name>Aakash Group</name> <!-- sub-child elements for properties element -->
<desc>Wonderful place in center of city and near VIP road</desc>
<noOfFlat>16</noOfFlat>
<features>
<feature>3 Automatic Elevator</feature>
<feature> Garden</feature>
<feature>Swimming pool</feature>
<feature>Gym</feature>
<feature>24*7 security</feature>
<feature>Inter communication</feature>
```

```
</features>
</property>
</properties>
```

```
<!-- flats is a child element of root -->
<flats>
<flat property="Akash Group"> <!-- reference to property for flat element -->
<name> Akash East Point</name>
<desc> Amazing architecture with modern facilities</desc>
<features>
<feature> 5 BHK</feature>
<feature> Centralized AC</feature>
<feature> 4 bathrooms</feature>
<feature> Green environment</feature>
<feature> happy home</feature>
</features>
</flat>
</flats>
</real-estate>
```

DTD File (ast6.dtd)

<!ELEMENT real-estate (agencies, owners, properties, flats)>

<!ELEMENT agencies (agency*)>

<!ELEMENT owners (owner*)>

<!ELEMENT properties (property*)>

<!ELEMENT flats (flat*)>

<!ELEMENT agency (name, email, phone)>

<!ELEMENT owner (name)>

<!ELEMENT property (name, desc, noOfFlat, features?)>

<!ATTLIST property owner CDATA #REQUIRED>

<!ELEMENT flat (name, desc, features)>

<!ATTLIST flat property CDATA #REQUIRED>

<!ELEMENT name (#PCDATA)>

<!ELEMENT email (#PCDATA)>

<!ELEMENT phone (#PCDATA)>

<!ELEMENT noOfFlat (#PCDATA)>

<!ELEMENT desc (#PCDATA)>

<!ELEMENT features (feature+)>

<!ELEMENT feature (#PCDATA)>

XML Schema File (ast6.xsd)

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="real-estate">
    <xs:annotation>
      <xs:documentation>root element</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="agencies">
          <xs:annotation>
            <xs:documentation>agencies is a child element of root</xs:documentation>
          </xs:annotation>
          <xs:complexType>
            <xs:sequence>
              <xs:element name="agency" maxOccurs="unbounded" minOccurs="0">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:string" name="name"/>
                    <xs:element type="xs:string" name="email"/>
                    <xs:element type="xs:long" name="phone"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="owners">
          <xs:annotation>
            <xs:documentation>owners is a child element of root</xs:documentation>
          </xs:annotation>
          <xs:complexType>
            <xs:sequence>
              <xs:element name="owner" maxOccurs="unbounded" minOccurs="0">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:string" name="name"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="properties">
          <xs:annotation>
            <xs:documentation>properties is a child element of root</xs:documentation>
          </xs:annotation>
          <xs:complexType>
            <xs:sequence>
```

```

<xs:element name="property">
  <xs:complexType>
    <xs:sequence>
      <xs:element type="xs:string" name="name">
        <xs:annotation>
          <xs:documentation>owner reference for property</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element type="xs:string" name="desc">
        <xs:annotation>
          <xs:documentation>sub-child elements for properties
element</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element type="xs:byte" name="noOfFlat"/>
      <xs:element name="features">
        <xs:complexType>
          <xs:sequence>
            <xs:element type="xs:string" name="feature" maxOccurs="unbounded"
minOccurs="0"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:sequence>
        <xs:attribute type="xs:string" name="owner"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="flats">
  <xs:annotation>
    <xs:documentation>flats is a child element of root</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="flat">
        <xs:complexType>
          <xs:sequence>
            <xs:element type="xs:string" name="name">
              <xs:annotation>
                <xs:documentation>reference to property for flat
element</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element type="xs:string" name="desc"/>
            <xs:element name="features">
              <xs:complexType>
                <xs:sequence>

```

```
        <xs:element type="xs:string" name="feature" maxOccurs="unbounded"
minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
    <xs:attribute type="xs:string" name="property"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

XSL Transformation (to HTML) file (ast6.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>

  <body>
  <!-- agency elements display-->
  <h1>Agency Details</h1>
  <table border="2" bgcolor="yellow">
    <tr>
      <th>Agency Name</th>
      <th>Email ID</th>
      <th>Phone</th>
    </tr>
    <xsl:for-each select="real-estate/agencies/agency">
      <tr>
        <td><xsl:value-of select="name"/></td>
        <td><xsl:value-of select="email"/></td>
        <td><xsl:value-of select="phone"/></td>

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

  <!-- owner elements display-->
  <h1>Owner Details</h1>
  <table border="2" bgcolor="lightgreen">
    <tr>
      <th>Owner Name</th>
    </tr>
    <xsl:for-each select="real-estate/owners/owner">
      <tr>
        <td><xsl:value-of select="name"/></td>
      </tr>
    </xsl:for-each>
  </table>

  <!-- property elements display-->
  <h1>Property Details</h1>
  <table border="2" bgcolor="lightblue">
    <tr>
      <th>Property Name</th>
      <th>Description</th>
      <th>Number of Flats</th>
      <th>Features Highlights</th>
    </tr>
    <xsl:for-each select="real-estate/properties/property">
```



```

        <tr>
            <td><xsl:value-of select="name"/></td>
            <td><xsl:value-of select="desc"/></td>
            <td><xsl:value-of select="noOfFlat"/></td>
            <td><xsl:value-of select="features"/></td>
        </tr>
    </xsl:for-each>
</table>

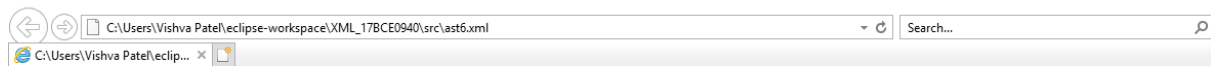
    <!-- flat info -->
    <h1> Flat Details</h1>
    <table border="2" bgcolor="pink">
        <tr>
            <th>Flat Name</th>
            <th>Description</th>
            <th>Features</th>
        </tr>
        <xsl:for-each select="real-estate/flats/flat">
            <tr>
                <td><xsl:value-of select="name"/></td>
                <td><xsl:value-of select="desc"/></td>
                <td><xsl:value-of select="features"/></td>
            </tr>
        </xsl:for-each>
    </table>

</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

Output Screenshots

XML file view on Internet Explorer



Agency Details

Agency Name	Email ID	Phone
Patel Builders	vishvapatel652@gmail.com	9003362230
Vishva Developers	vishvap@gmail.com	9106848042

Owner Details

Owner Name
Vishva Patel
S Rama
Anand Vishva

Property Details

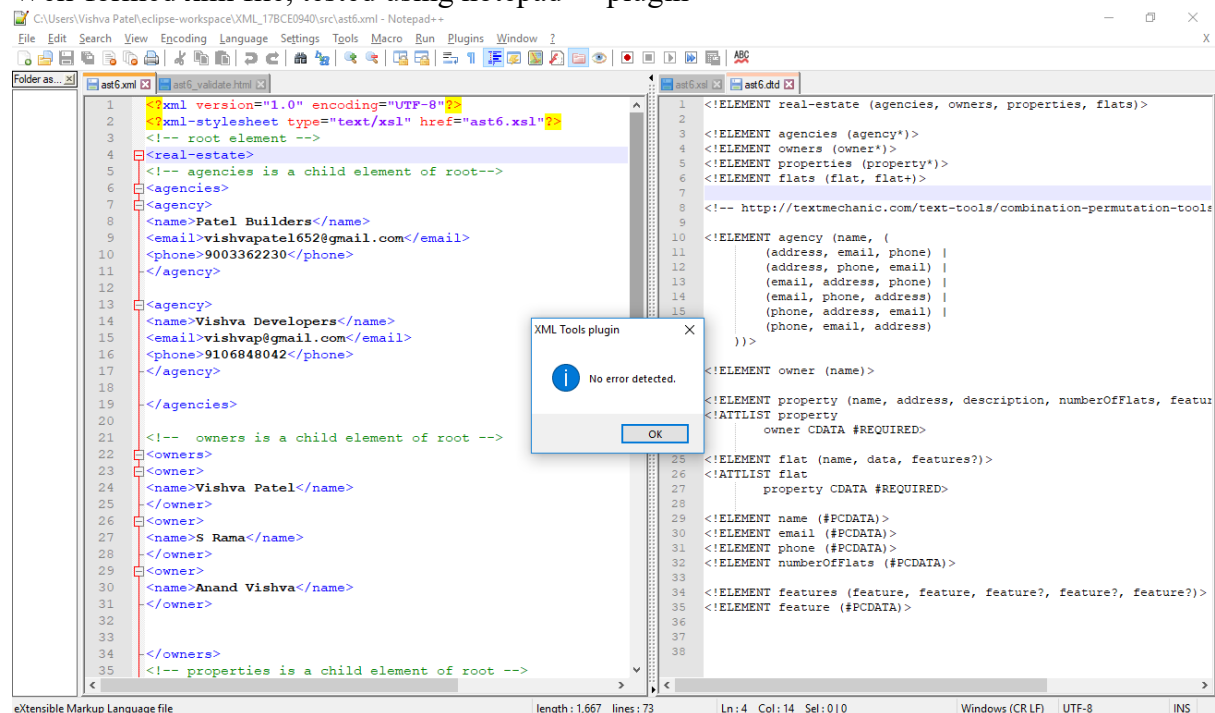
Property Name	Description	Number of Flats	Features Highlights
Aakash Group	Wonderful place in center of city and near VIP road	16	3 Automatic Elevator Garden Swimming pool Gym 24*7 security Inter communication

Flat Details

Flat Name	Description	Features
Akash East Point	Amazing architecture with modern facilities	5 BHK Centralized AC 4 bathrooms Green environment happy home

Well Formedness of ast6.xml

Well-formed xml file, tested using notepad++ plugin



The screenshot shows the W3Schools XML Validator interface. The browser's address bar displays the URL 'w3schools.com/xml/xml_validator.asp'. On the left, a navigation menu lists various XML topics, with 'XML Validator' selected and highlighted in green. The main content area is divided into two sections. The top section displays a sample XML document:

```
<?xml version='1.0' encoding='UTF-8' />
<heading>Reminder</heading>
<body>Don't forget me</body>
</note>
```

 A message box from 'www.w3schools.com' states 'No errors found'. Below this, there is a text input area for a custom XML document, containing an example for a real-estate agency. A 'Check XML' button is positioned next to the input area. The bottom section, titled 'Valid XML Documents', provides an explanation of XML validity, distinguishing between 'well formed' and 'valid' documents.

XML Validator

w3schools.com/xml/xml_validator.asp

HTML CSS JAVASCRIPT SQL PYTHON

XML Tutorial

XML HOME

XML Introduction

XML How to use

XML Tree

XML Syntax

XML Elements

XML Attributes

XML Namespaces

XML Display

XML HttpRequest

XML Parser

XML DOM

XML XPath

XML XSLT

XML XQuery

XML XLink

XML Validator

XML DTD

XML Schema

XML Server

XML Examples

XML Quiz

XML Certificate

XML Validator

www.w3schools.com says

No errors found

OK

Try to syntax-check your own XML :

```
<real-estate>
<!-- agencies is a child element of root-->
<agencies>
<agency>
<name>Patel Builders</name>
<email>vishvapatel652@gmail.com</email>
<phone>9003362230</phone>
</agency>
</real-estate>
```

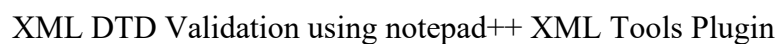
Valid XML Documents

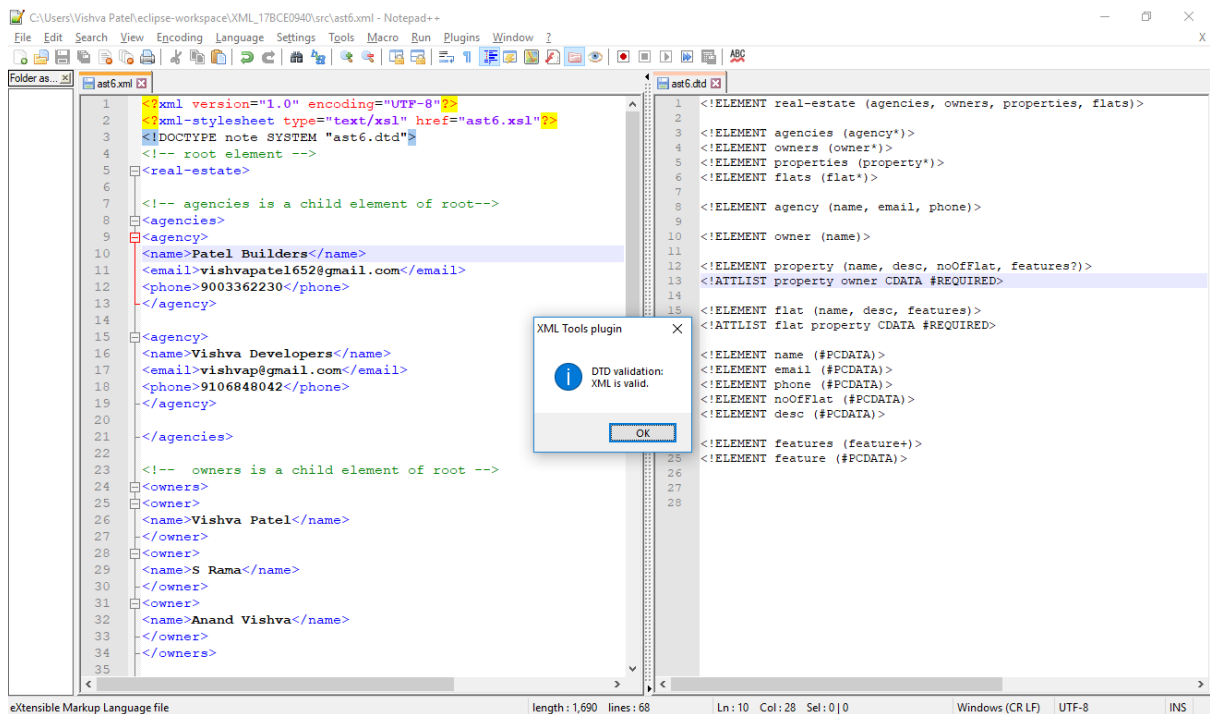
A "well formed" XML document is not the same as a "valid" XML document.

A "valid" XML document must be well formed. In addition, it must conform to a document type definition.

There are two different document type definitions that can be used with XML:

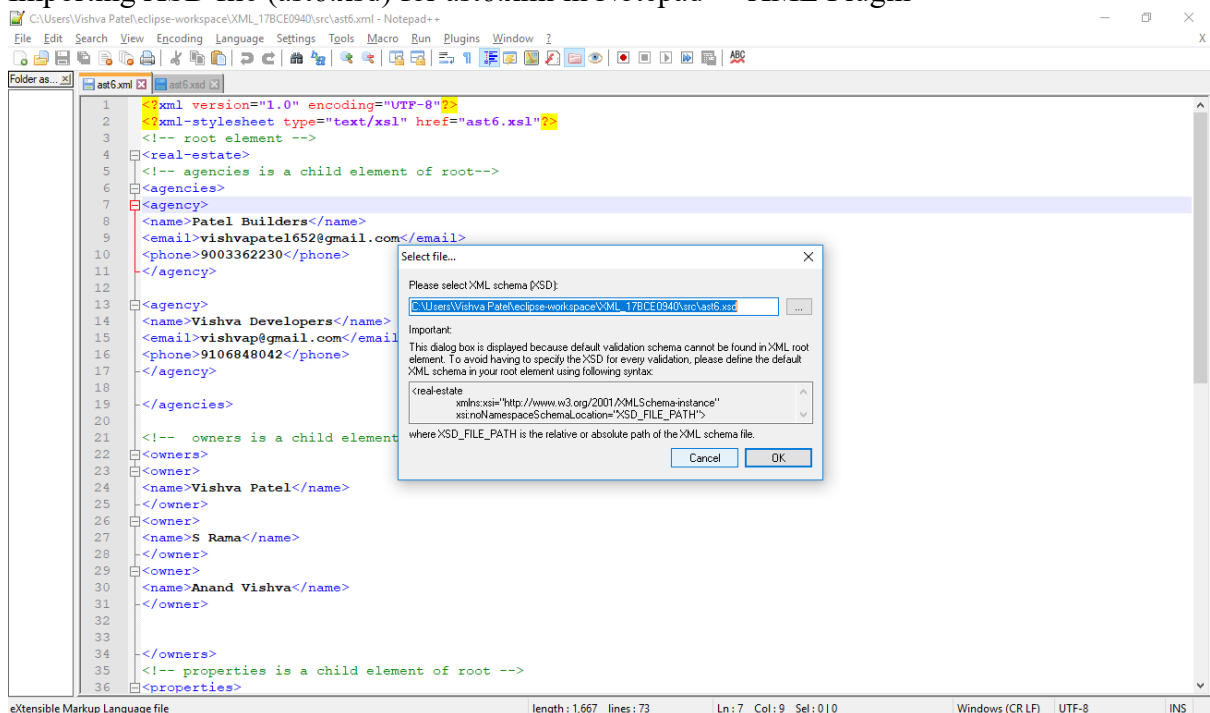
Installing XML Tools plugin in notepad++





XML Schema Validation

Importing XSD file (ast6.xsd) for ast6.xml in Notepad++ XML Plugin



Ast6.xml is validated through XML Schema

