Experiment No. 04

Title: Create XML document and display using XSLT

Batch:A2   **Roll No.:16010421063**   **Experiment No.:4**

Aim: To Create XML document and display using XSLT

Resources needed: Notepad++, Web Browser

Theory:

XML stands for E**X**tensible **M**arkup **L**anguage. XML is a software- and hardware independent tool for storing and transporting data. XML was designed to be both human- and machine-readable. XML was designed to be self-descriptive. XML is just information wrapped in tags. XML was designed to carry data - with focus on what data is. HTML was designed to display data - with focus on how data looks. The XML language has no predefined tags, the author must define both the tags and the document structure.

XML stores data in plain text format. This provides a software- and hardware-independent way of storing, transporting, and sharing data. XML also makes it easier to expand   or upgrade to new operating systems, new applications, or new browsers, without losing data. With XML, data can be available to all kinds of "reading machines" like people, computers, voice machines, news feeds, etc. XML is often used to separate data from presentation. XML does not carry any information about how to be displayed.

Example catalog.xml:-

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

<?xml-stylesheet type="text/xsl" href="catalog.xsl"?>

<catalog>

    <cd>

        <title>Empire Burlesque</title>

        <artist>Bob Dylan</artist>

        <country>USA</country>

        <company>Columbia</company>

        <price>10.90</price>

        <year>1985</year>

    </cd>

</catalog>
```

First line defines version. The next line describes root element which is "catalog" in this case. The last line defines the end of root element.

XML Syntax Rules

- XML documents form a tree structure that starts at "the root" and branches to "the leaves". XML documents must contain one **root** element that is the parent of all other elements.
- **<?xml version="1.0" encoding="UTF-8"?>** This line is called the XML **prolog.** The XML prolog is optional. If it exists, it must come first in the document.
- All XML Elements Must Have an Opening and Closing Tag.ie, Each element is enclosed in <> brackets.
- XML tags are case sensitive. Tags should all be written in lower case. · Empty tags should be written in an XML style that still displays properly in browsers, with a trailing space and slash before the closing bracket (e.g.
). · In XML, all elements must be properly nested within each other.
- XML comment syntax is similar to that of HTML. <!-- This is a comment --> · XML elements can have attributes in name/value pairs just like in HTML. XML requires that all XML attributes must have a value.

      <student active="true">

      <name>Sam</name>

      <grade>A</grade>

      </student>

- In XML, the attribute values must always be quoted.

      <note date="12/11/2007">
          </note>

- Elements that are included in another element are called nested element.

- The naming rules for XML elements:

      ☐ it consist letters, numbers, and other characters
      ☐ Names must not start with a number or punctuation character
      ☐ Names must not start with the letters xml (or XML, or Xml, etc)
      ☐ Names cannot contain spaces

XSL - More Than a Style Sheet Language

External style sheet language standard specifies how to display an XML document. Using XSL it is possible to:

1. Transfer an XML document to HTML.
2. Filter and sort XML data.

To transfer the above catalog.XML document to HTML.

catalog.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>

<table border="1">

<tr bgcolor="#9acd32">

<th>Title</th>

<th>Artist</th>

</tr>

<xsl:for-each select="catalog/cd[artist='Bob Dylan']">

<tr>

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

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

</tr>

</xsl:for-each>

</table>

</body>

</html>

</xsl:template>

</xsl:stylesheet>
```

The first three lines are standard lines for any xsl document. Next line indicates table format in HTML. Since an XSL style sheet is an XML document, it always begins with the XML declaration: **<?xml version="1.0" encoding="UTF-8"?>**. The next element, **<xsl:stylesheet>**, defines that this document is an XSLT style sheet document (along with the version number and XSLT namespace attributes). To get access to the XSLT elements, attributes and features we must declare the XSLT namespace at the top of the document. The

xmlns:xsl="http://www.w3.org/1999/XSL/Transform" points to the official W3C XSLT namespace. If you use this namespace, you must also include the attribute version="1.0".

Document Type Definition (DTD)

Markup languages require a Document Type Definition which defines the elements that are allowed in the document. The DTD also defines how elements may be used with relationship to each other. It define how many and which elements may be included inside another element. The DTD is a text file written in a specific format to define the document. The DTD is based on the Standardized Generalized Markup Language (SGML). SGML is the parent language of all markup languages.

XML uses document type definition or an XML schema to describe the data. XML documents are used as self describing and simple syntax. Although XML may use a DTD, it is not required for those documents that are considered "well formed". This is an XML document with a Document Type Definition:

<?xml version="1.0"?>

<!DOCTYPE note [

<!ELEMENT note (to,from,heading,body)>

<!ELEMENT to (#PCDATA)>

<!ELEMENT from (#PCDATA)>

<!ELEMENT heading (#PCDATA)>

<!ELEMENT body (#PCDATA)>

]>

<note>

<to>Tove</to>

<from>Jani</from>

<heading>Reminder</heading>

<body>Don't forget me this weekend!</body>

</note>

Displaying XML

Raw XML files can be viewed in all major browsers. Don't expect XML files to be displayed as HTML pages. XML files can be displayed using CSS. Unlike HTML, XML does not predefine display properties for specific elements. Therefore, XML requires a separate stylesheet that specifies how the XML data should be displayed. This separation of XML

content from its presentation allows the content to be easily repurposed.

The easiest way to display these documents is to create XML documents based on the XHTML DTD and to let the browser display them as HTML.

This ensures that your documents can both be displayed in a Web browser and still be processed by XML tools.

Procedure:
1.   Open text editor and write XML code with the use of tags mentioned.
2. Save file with .xml extension.
3. Open the text editor and write XSL code and transfer it to XML .
4. Save file with .xsl extension.
5. Run saved XML file in browser.

Activity:
Design a catalog specific to your website using XML and display it using XSLT.

Results: (Program printout with output)

css

```css
Spacecrafts {
    font-size: 80%;
    margin: 0.5em;
    font-family: Verdana;
    display: block;
    background-color: black;
}
properties {
    display: block;
    border: 6px solid rgb(252, 252, 252);
    margin: 3.5em;
    padding: 3.5em;
    background-color: transparent;
}
h2 {
    display: block;
    font-weight: bolder;
    text-align: left;
    color: rgb(213, 173, 240);
}
Name,
Launchdate,
Launchvehicle,
OrbitType,
Application {
    display: block;
```

```css
        text-align: center;
}
Name {
    display: block;
    font-size: 1.5vw;
    text-align: left;
    color: rgb(241, 202, 202);
}
Launchdate {
    display: block;
    font-size: 1.5vw;
    text-align: left;
    color: rgb(241, 202, 202);
}
Launchvehicle {
    display: block;
    font-size: 1.5vw;
    text-align: left;
    color: rgb(241, 202, 202);
}
OrbitType {
    display: block;
    font-size: 1.5vw;
    text-align: left;
    color: rgb(241, 202, 202);
}
Application {
    display: block;
    font-size: 1.5vw;
    text-align: left;
    color: rgb(241, 202, 202);
}
```

XML

```xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="Spacecrafts.css"?>
<Spacecrafts>
<properties>
<Name>Name : EOS-04</Name>
<Launchdate>Launchdate : Feb 14, 2022 </Launchdate>
<Launchvehicle>Launchvehicle : PSLV-C52/EOS-04 Mission </Launchvehicle>
<OrbitType>OrbitType : GTO</OrbitType>
<Application>Application : Earth Observation </Application>
</properties>
<properties>
<Name>Name : EOS-03</Name>
```

```xml
<Launchdate>Launchdate : Aug 12, 2021 </Launchdate>
<Launchvehicle>Launchvehicle : GSLV-F10 / EOS-03 </Launchvehicle>
<OrbitType>OrbitType : GTO </OrbitType>
<Application>Application : Earth Observation </Application>
</properties>
<properties>
<Name>Name : CMS-01</Name>
<Launchdate>Launchdate : Dec 17, 2020 </Launchdate>
<Launchvehicle>Launchvehicle : PSLV-C50/CMS-01 </Launchvehicle>
<OrbitType>OrbitType : GSO </OrbitType>
<Application>Application : Communication </Application>
</properties>
<properties>
<Name>Name: GSAT-30</Name>
<Launchdate>Launchdate : Jan 17, 2020 </Launchdate>
<Launchvehicle>Launchvehicle : Ariane-5 VA-251 </Launchvehicle>
<OrbitType>OrbitType : GSO </OrbitType>
<Application>Application : Communication </Application>
</properties>
</Spacecrafts>
```

XSL

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<html>
<head>
<title> Spacecrafts </title>
<style>
body{
/* The image used */
background:lineargradient(rgba(180,180,180,0.5),rgba(200,200,200,0.5)),
url("valley.jpg");
min-height: 300px;
/* Center and scale the image nicely */
background-position: center;
background-repeat: no-repeat;
background-size: cover;
margin:0;
}
h2{font-size: 69px;
font-weight: bold;
font-family:'Poppins', Arial, sans-serif;
color: darkgreen;
margin-top: 140px;
margin-bottom: 17px;
display:center;
```

```
text-align: center;
}
table{
margin:30px;
}
</style>
</head>
<body>
<h2> Spacecrafts </h2>
<p> Lorem Ipsum is simply dummy text of the printing and typesetting
industry.
Lorem Ipsum has been the industry's standard dummy text ever since the
1500s,
when an unknown printer took a galley of type and scrambled it to make a
type
specimen book. It has survived not only five centuries, but also the leap
into
electronic typesetting, remaining essentially unchanged. It was
popularised in
the 1960s with the release of Letraset sheets containing Lorem Ipsum
passages,
and more recently with desktop publishing software like Aldus PageMaker
including versions of Lorem Ipsum.</p>
<table border = "1" >
<tr>
<th style="text-align:center;">Name</th>
<th style="text-align:center;">Launchdate</th>
<th style="text-align:center;">Launchvehicle</th>
<th style="text-align:center;">OrbitType</th>
<th style="text-align:center;">Application</th>
</tr>
<xsl:for-each select="Spacecrafts/properties">
<tr>
<td><xsl:value-of select="Name"/></td>
<td><xsl:value-of select="Launchdate"/></td>
<td><xsl:value-of select="Launchvehicle"/></td>
<td><xsl:value-of select="OrbitType"/></td>
<td><xsl:value-of select="Application"/></td>
</tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:styleshee>
```

Outcomes:

CO 2 Create Web pages using HTML 5 and CSS.

Conclusion: (Conclusion to be based on the outcomes achieved)

Implemented XML

Grade: AA / AB / BB / BC / CC / CD

/DD Signature of faculty in-charge with

date **References:**

Books/ Journals/ Websites:
1. http://www.w3schools.com
2. http://nwalsh.com/docs/tutorials/xsl/xsl/slides.html