UNIT V: XML

5.1 Introduction to XML
5.2 uses of XML
5.3 simple XML
5.4 XML key components
5.5 DTD and Schemas
5.6 Using XML with application.
5.7 Transforming XML using XSL and XSLT.

5.1 Introduction to XML

Xml (eXtensible Markup Language) is a mark up language.

XML is designed to store and transport data.

Xml was released in late 90's. it was created to provide an easy to use and store self describing data.

XML became a W3C Recommendation on February 10, 1998.

XML is not a replacement for HTML.

XML is designed to be self-descriptive.

XML is designed to carry data, not to display data.

XML tags are not predefined. You must define your own tags.

XML is platform independent and language independent.

A mark up language is a modern system for highlight or underline a document.

Students often underline or highlight a passage to revise easily, same in the sense of modern mark up language highlighting or underlining is replaced by tags.

The XML language has no predefined tags.

XML simplifies data sharing
XML simplifies data transport
XML simplifies platform changes
XML simplifies data availability

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.

5.2 uses of XML

XML is used in many aspects of web development.

XML is often used to separate data from presentation.

XML Separates Data from Presentation

XML does not carry any information about how to be displayed.

The same XML data can be used in many different presentation scenarios.

Because of this, with XML, there is a full separation between data and presentation.

XML is Often a Complement to HTML

In many HTML applications, XML is used to store or transport data, while HTML is used to format and display the same data.

XML Separates Data from HTML

When displaying data in HTML, you should not have to edit the HTML file when the data changes.

With XML, the data can be stored in separate XML files.

With a few lines of JavaScript code, you can read an XML file and update the data content of any HTML page.

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

<bookstore>

 <book category="cooking">
  <title lang="en">Everyday Italian</title>
  <author>Giada De Laurentiis</author>
  <year>2005</year>
  <price>30.00</price>
 </book>

</bookstore>
```

5.3 simple XML

The syntax rules of XML are very simple and logical. The rules are easy to learn, and easy to use.

1 XML Documents Must Have a Root Element

XML documents must contain one root element that is the parent of all other elements:

```
<root>
  <child>
    <subchild>.....</subchild>
  </child>
</root>
```

In this example <note> is the root element:

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <body>Don't forget me this weekend!</body>
</note>
```

2 The XML Prolog

This line is called the XML prolog:

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

The XML prolog is optional. If it exists, it must come first in the document.
XML documents can contain international characters

To avoid errors, you should specify the encoding used, or save your XML files as UTF-8.UTF-8 is the default character encoding for XML documents.

3 All XML Elements Must Have a Closing Tag

In XML, it is illegal to omit the closing tag. All elements must have a closing tag:

4 XML Tags are Case Sensitive

XML tags are case sensitive. The tag <Letter> is different from the tag <letter>.
Opening and closing tags must be written with the same case:

5 XML Elements Must be Properly Nested

```
<b><i>This text is bold and italic</b></i>
```

XML Attribute Values Must Always be Quoted
XML elements can have attributes in name/value pairs just like in HTML.

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

5.4 XML key components

1. XML document

An XML file begins with the <xml></xml> tags. The data written in between these tags is part of an XML document. XML processing software uses these specific tags as reference points to compute the XML code.

2. XML declaration

An XML file has an opening declaration that defines the XML version used in the document. It initiates the XML processor to parse the XML document. It has the following syntax:

<?xml version=”1.0″ encoding=”UTF-16″ standalone=’yes’?>

3. XML elements

The remaining tags within the XML document, excluding the tags of the above components, are labeled as XML elements.

These contain features such as text, attributes, and other XML file elements. The root element marks the beginning of the XML document.

<NewYearPartyList>
<office><managers><name>Albert</name></managers></office>
</NewYearPartyList>

4. XML attributes

XML attributes refer to the descriptors that provide details of the XML elements. You can write the attribute names and the corresponding values under quotation marks.

<Albert height=“173”>

5. XML content

Data embedded within XML files are referred to as XML content.

6. XML schema

XML schema sets boundaries to the XML file structure. It expresses rules and constraints that need to be obeyed by the XML document.

7. XML parser

XML parser refers to software that validates XML files by evaluating their syntax. Subsequently, it processes or reads XML documents to extract relevant information.

5.5 DTD and Schemas

S.NO.	DTD	XSD
1.	DTD are the declarations that define a document type for SGML.	XSD describes the elements in a XML document.
2.	It doesn't support namespace.	It supports namespace.
3.	It is comparatively harder than XSD.	It is relatively more simpler than DTD.
4.	It doesn't support datatypes.	It supports datatypes.
5.	SGML syntax is used for DTD.	XML is used for writing XSD.
6.	It is not extensible in nature.	It is extensible in nature.
7.	It doesn't give us much control on structure of XML document.	It gives us more control on structure of XML document.
8.	It specifies only the root element.	Any element which is made global can be done as root as markup validation.
9.	It doesn't have any restrictions on data used.	It specifies certain data restrictions.
10.	It is not much easy to learn .	It is simple in learning.
11.	File here is saved as .dtd	File in XSD is saved as .xsd file.
12.	It is not a strongly typed mechanism.	It is a strongly typed mechanism.
13.	It uses #PCDATA which is a string data type.	It uses fundamental and primitive data types.

5.6 Using XML with application.

```
<!DOCTYPE html>
<html>
<body>

<div id='showCD'></div><br>
<input type="button" onclick="previous()" value="<<">
<input type="button" onclick="next()" value=">>">

<script>
var i = 0;
var x;
displayCD(i);

function displayCD(i) {
   var xmlhttp = new XMLHttpRequest();
   xmlhttp.onreadystatechange = function() {
      if (this.readyState == 4 && this.status == 200) {
         myFunction(this, i);
      }
   };
   xmlhttp.open("GET", "cd_catalog.xml", true);
   xmlhttp.send();
}

function myFunction(xml, i) {
   var xmlDoc = xml.responseXML;
   x = xmlDoc.getElementsByTagName("CD");
   document.getElementById("showCD").innerHTML =
   "Artist: " +
   x[i].getElementsByTagName("ARTIST")[0].childNodes[0].nodeValue +
   "<br>Title: " +
   x[i].getElementsByTagName("TITLE")[0].childNodes[0].nodeValue +
   "<br>Year: " +
   x[i].getElementsByTagName("YEAR")[0].childNodes[0].nodeValue;
}

function next() {
if (i < x.length-1) {
  i++;
  displayCD(i);
  }
}
```

```
function previous() {
if (i > 0) {
  i--;
  displayCD(i);
  }
}
</script>

</body>
</html>
```

5.7 Transforming XML using XSL and XSLT.

XSLT stands for Extensible Stylesheet Language Transformation.

XSLT is used to transform XML document from one form to another form.

XSLT uses Xpath to perform matching of nodes to perform these transformation

The result of applying XSLT to XML document could be an another XML document, HTML, text or any another document from technology perspective.

The XSL code is written within the XML document with the extension of (.xsl). In other words, an XSLT document is a different kind of XML document.

Correct Style Sheet Declaration

The root element that declares the document to be an XSL style sheet is <xsl:stylesheet> or <xsl:transform>.

```
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:transform version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

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".

We want to transform the following XML document ("cdcatalog.xml") into XHTML

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="cdcatalog.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>
```

The End