Experiment - 03.

- **Aim:** Design the XML document to store the information of employees of any business organisation and demonstrate use of:
  a) DTD.
  b) XML Schema.
  c) And display the content in (eg. tabular format) by using CSS/XSL.

- **Theory:**

(1) XML:

- XML stands for extensible markup language. It's a markup language much like HTML, but whereas HTML is used to display data, XML is used to store and transport data
- It's a flexible way to create common information formats and share structured data over the internet, making it a common tool in web development, data exchange and various other application some key points of XML:
  1) Markup language
  2) Extensible.
  3) Platform independent.
  4) Human and machine readable.
  5) Heirarchical structure.
  6) Well defined structure.

- Basic XML Syntax -

```
<? xml version ="1.0" encoding = " UTF - 0.8 "?>
<! DOCTYPE root SYSTEM "file. dtd" >
  <root >
          < element attribute = " Value " > Content
           </element>
  </root >
```

(1)

(2). Structure for employee information -

- Root element : <employee >
- Child element : <employee > (Each employee's data).
- Attributes : < employee > has an 'id' attribute.
- Nested elements: <name >, <department >, <position >, <salary > , <contact >
- Contact information: <email> and <phone >

(3). Document type definition (CDTD) -

- DTD is a set of rules that define the structure and allowed elements of an XML document. It ensures that an XML file follows a specific format and maintains consistency.

- Ensures data validity checking whether an XML doument follows the specified format.

- Helps applications understand the expected heirarchy and relationships within an XML file.

- **Types of DTD -**

(1) **Internal:** Defined inside XML document using <! doctype>

Syntax:

```
<? xml version = "1.0" encoding = "UTF - 8" ? >
<! DOCTYPE root [
        <! ELEMENT root ( child1 , child2) >
        <! ELEMENT child1 (# PCDATA) 7
  ]>
<root >
        < child1 > Data 1 </child1 >
         < child2 >Data2 < /child2 >
 </root >
```

#PCDATA - Means elements contain text , parsed character data.

<! ELEMENT > - Specifies elements and their content.

(2) **External DTD:**

```
<! DOCTYPE root SYSTEM " example.dtd" >
```

4] **XML Schema -**

- An XML Schema (XSD - XML Schema definition) is a powerful way to define the structure , data types and rules for an XML document.

- Supports datatypes like integers, decimals, dates, etc.
- Allows reusability of data definitions.
- Provides namespace support for XML document.

- **Basic syntax of XML Schema -**

```
<xs: schema  xmlns:xs = "http:// w3. org /2001/ XMLSchema
        <!--Define elements -->
</xs: schema>
```

- XML Schema components -

1. Element - <xs: element name = "elementName" type = "DT"/>
2. Attribute declaration -
   <xs : attribute name = "attributeName" type = "Datatype"
   use = "required/ optional"/>

3. **Datatypes in XML Schema -**

Data type.	Description.
xs: string	Text content.
xs: integer.	Whole numbers
xs : decimal.	Decimal numbers
xs: boolean.	true or false.
xs: date.	Date format (YYYY - MM - DD)
xs : time.	Time format (HH: MM:SS).

Eg: <xs : element name = "salary" type = "xs: decimal"/>

4. You can also add constraints such as -
   - xs: minLength
   - xs : maxLength.
   - xs : restriction

- Advantages of Schema -

- Supports data types like integer, date, boolean, etc.
- Better validation with contraints like min/max value.
- Reusable components using complex types.
- Supports namespace for handling multiple Schema.
- More readable and scalable than DTD.

5) XSLT : Extensible Stylesheet language Transformations.

- Converts XML to readable formats.

Basic XSLT Syntax:

```
<xsl : stylesheet   version = "1.0"   xmlns : xsl = "http :// www.w3.
  org/ 1999/ XSL/ Transform "/>
          <xsl: template  match ="/ " >
                <html>
                     <body>
                          <table>
                              <xsl : for-each  select = "Root
                                Element / ChildElement" >
                                <tr>
                                   <td> <xsl : value of
                                     select = "element" /> </td
                                </tr>
                             </xsl :for each>
                          </table>
                     </body>
                </html>
          </xsl : template>
     </xsl : stylesheet >
```

* XSLT -

- <xsl: for-each select = "employees/ employee "> – loops through all <employee> elements
- <td> <xsl: value-of select ="element" /> </td> – Extracts all and displays element values inside table cells.

- **Conclusion:** Thus we successfully studied and implemented xml, dtd and XML schema, Stylesheets.

* * * * * * *

8/3/25