

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 applications. Some key points of XML:

- 1) Markup language
- 2) Extensible.
- 3) Platform independent.
- 4) Human and machine readable.
- 5) Hierarchical structure.
- 6) Well defined structure.

- Basic XML Syntax -

```

<? xml version = "1.0" encoding = "UTF-8"?>
<!DOCTYPE root SYSTEM "file.dtd">
<root>
    <element attribute = "value" > Content
    </element>
</root>

```

(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 (DTD) -

- 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 document follows the specified format.
- Helps applications understand the expected hierarchy 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) >
]
<root>
    <child1> Data1 </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 -

```

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

```

- XML Schema components -

1. Element - `<xsd:element name = "elementName" type = "DT"/>`

2. Attribute declaration -

```

<xsd:attribute name = "attributeName" type = "Datatype"
    use = "required/optional"/>

```

3. Datatypes in XML Schema -

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

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

4. You can also add constraints such as -

- `xsd:minLength`
- `xsd:maxLength`.
- `xsd:restriction`

Advantages of Schema -

- Supports data types like integer, date, boolean, etc.
- Better validation with constraints 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>

```

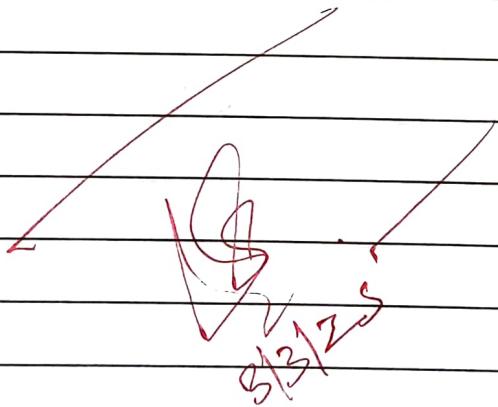
17.

12

* 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.



* * * * *