Name: **Piyusha Rajendra Supe**

Enrolment Number: **23CO315**

Web Technology Practical 3

Title: Design the XML document to store the information of the employees of any business organization and demonstrate the use of:

a) DTD

b) XML Schema

And display the content in (e.g., tabular format) by using CSS/XSL.

The files are as follows:

[1] Employee.xml – Normal xml file

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="piyushastyle.xsl"?>
<!DOCTYPE PIU_Companyemployee SYSTEM "employee.dtd">
<PIU_Companyemployee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                xsi:schemaLocation="employee.xsd">
   <emp>
      <name>Piyusha Supe</name>
      <age>20</age>
      <phone>1234567890</phone>
      <email>piyu@gmail.com</email>
      <salary>900000</salary>
      <location>Pune</location>
   </emp>
   <emp>
      <name>Swapnakirti</name>
      <age>28</age>
      <phone>7578378</phone>
      <email>sk@gmail.com</email>
      <salary>550002</salary>
      <location>Bangalore</location>
```

```
</emp>
<emp>
    <name>Brielle</name>
    <age>29</age>
    <phone>0987654321</phone>
    <email>bll@gmail.com</email>
    <salary>20000</salary>
    <location>Pune</location>
</emp>
<emp>
    <name>Cindy Ella</name>
    <age>39</age>
    <phone>7248278</phone>
    <email>cindy@gmail.com</email>
    <salary>80000</salary>
    <location>Pune</location>
</emp>
<emp>
    <name>Maxton Parker</name>
    <age>34</age>
    <phone>372478784</phone>
    <email>maxton@gmail.com</email>
    <salary>300000</salary>
    <location>Mumbai</location>
</emp>
<emp>
    <name>Piper Dcosta</name>
    <age>33</age>
    <phone>88987878</phone>
    <email>piper@gmail.com</email>
    <salary>940000</salary>
    <location>Pune</location>
```

```
   </emp>
</PIU_Companyemployee>
```

[2] Employee.dtd – DTD file document type definition

```
<!ELEMENT PIU_Companyemployee (emp+)>
<!ELEMENT emp (name, age, phone, email, salary, location)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT age (#PCDATA)>
<!ELEMENT phone (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT salary (#PCDATA)>
<!ELEMENT location (#PCDATA)>
```

[3] Piyushastyle.xsl – XSL for styling

```
<xsl:stylesheet version = "1.0"
xmlns:xsl = "http://www.w3.org/1999/XSL/Transform">
<xsl:template match = "/">
        <html>
                <body>
                        <h2>PIU Company Employee details</h2>
                        <table border = "1">
                        <tr bgcolor = "lightpink">
                                <th>Name</th>
                                <th>Age</th>
                                <th>Phone</th>
                                <th>Email</th>
                                <th>Salary</th>
                                <th>Location</th>
                        </tr>
                        <xsl:for-each select="PIU_Companyemployee/emp">
                                <tr>
                                        <td><xsl:value-of select = "name"/></td>
```

```
                    <td><xsl:value-of select = "age"/></td>
                    <td><xsl:value-of select = "phone"/></td>
                    <td><xsl:value-of select = "email"/></td>

                    <td><xsl:value-of select = "salary"/></td>
                    <td><xsl:value-of select = "location"/></td>
                </tr>
            </xsl:for-each>
            </table>
        </body>
    </html>
</xsl:template>
</xsl:stylesheet>
```

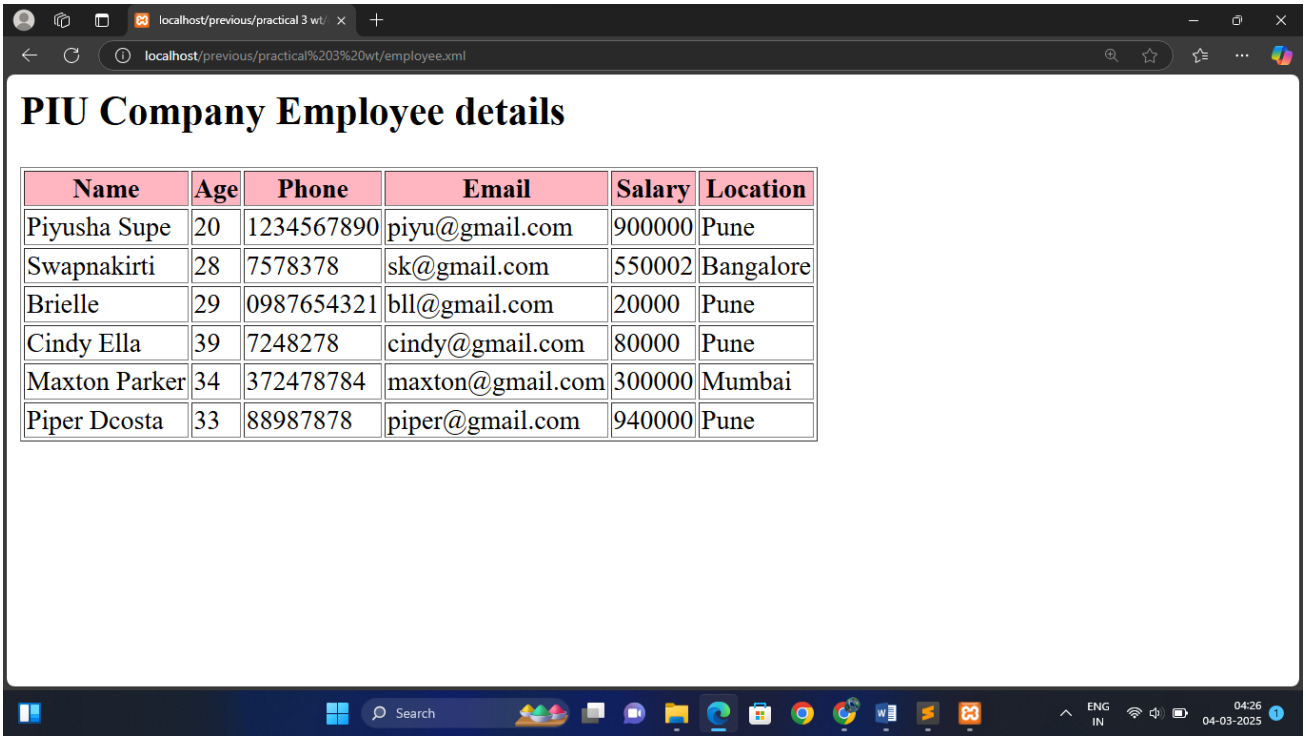
[4] Employee.xsd – Schema file

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="PIU_Companyemployee">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="emp" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="name" type="xs:string"/>
              <xs:element name="age" type="xs:integer"/>
              <xs:element name="phone" type="xs:string"/>
              <xs:element name="email" type="xs:string"/>
              <xs:element name="salary" type="xs:integer"/>
              <xs:element name="location" type="xs:string"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
```

```
        </xs:complexType>
    </xs:element>
</xs:schema>
```

OUTPUT:

The newer versions of browsers do not support running xsl sometimes so my solution was to run it with xampp. As xampp has apache server it interprets the xml and related files easily.



CONCLUSION:

Thus the XML, DTD, XSL, Schema were successfully executed as illustrated above.