

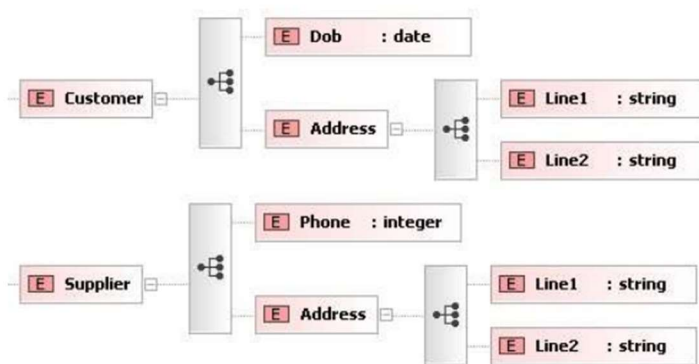
Assignment – 10

XML Lab

Reg No - 2021CA006

Abhinav Salar

1. Write an XML Schema (xsd file) and XML document specifying the following structure.



XML code <>

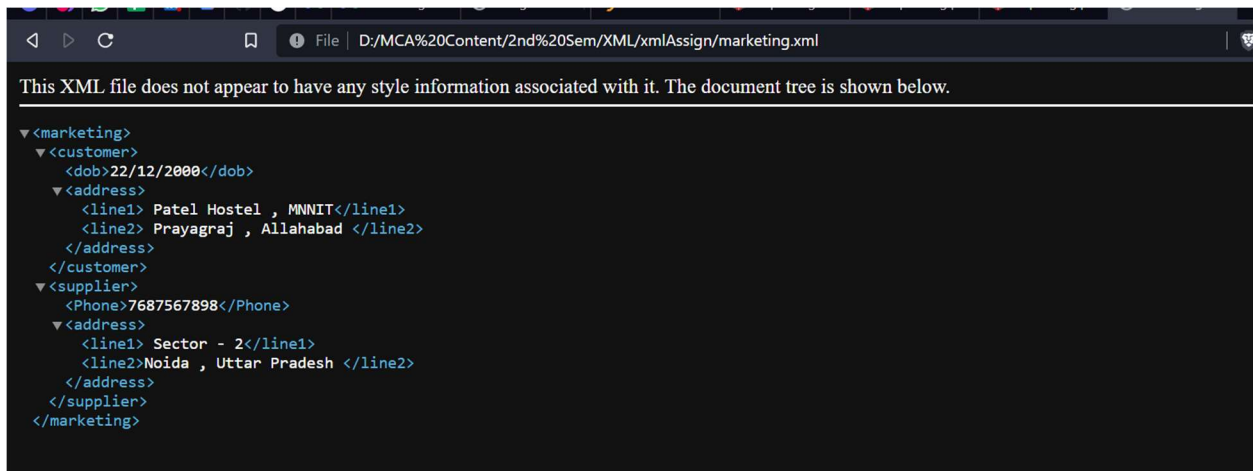
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE marketing SYSTEM "marketing.dtd">
<marketing>
  <customer>
    <dob>22/12/2000</dob>
    <address>
      <line1> Patel Hostel , MNNIT</line1>
      <line2> Prayagraj , Allahabad </line2>
    </address>
  </customer>
```

```

<supplier>
  <Phone>7687567898</Phone>
  <address>
    <line1> Sector - 2</line1>
    <line2>Noida , Uttar Pradesh </line2>
  </address>
</supplier>
</marketing>

```

Output:-



DTD code <>

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

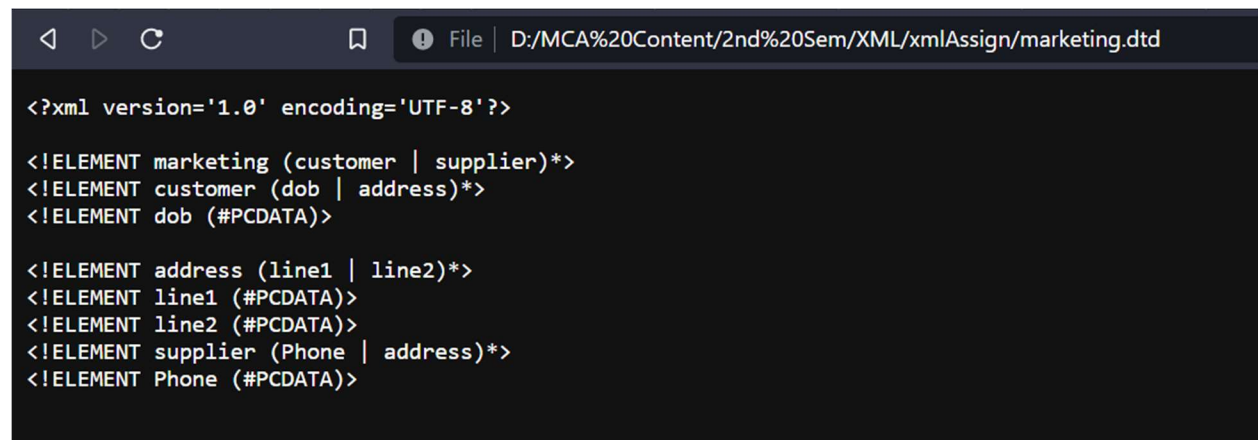
```
<!ELEMENT marketing (customer|supplier)*>
```

```
<!ELEMENT customer (dob|address)*>
```

```
<!ELEMENT dob (#PCDATA)>
```

```
<!ELEMENT address (line1|line2)*>
<!ELEMENT line1 (#PCDATA)>
<!ELEMENT line2 (#PCDATA)>
<!ELEMENT supplier (Phone|address)*>
<!ELEMENT Phone (#PCDATA)>
```

Output:-

A screenshot of a text editor window with a dark background. The title bar shows a file path: "D:/MCA%20Content/2nd%20Sem/XML/xmlAssign/marketing.dtd". The editor contains the following XML DTD code:

```
<?xml version='1.0' encoding='UTF-8'?>
<!ELEMENT marketing (customer | supplier)*>
<!ELEMENT customer (dob | address)*>
<!ELEMENT dob (#PCDATA)>

<!ELEMENT address (line1 | line2)*>
<!ELEMENT line1 (#PCDATA)>
<!ELEMENT line2 (#PCDATA)>
<!ELEMENT supplier (Phone | address)*>
<!ELEMENT Phone (#PCDATA)>
```

XSD code <>

```
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
</xs:schema>

<xs:element name="marketing">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="customer">
```

```
<xs:complexType>
  <xs:sequence>
    <xs:element name="dob" type="date">
    <xs:element name="Address" type="AddressType">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Line1" type="String">
          <xs:element name="Line2" type="String">
        </xs:sequence>
      </xs:complexType>
    </xs:sequence>
  </xs:complexType>
</xs:sequence>
</xs:complexType>
<xs:element name="suppiler">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="phone" type="int">
      <xs:element name="Address" type="AddressType">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Line1" type="String">
            <xs:element name="Line2" type="String">
          </xs:sequence>
        </xs:complexType>
      </xs:sequence>
    </xs:sequence>
  </xs:complexType>
</xs:sequence>
```

</xs:complexType>

</xs:sequence>

</xs:complexType>

Output:-

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
</xs:schema>
<xs:element name="marketing">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="customer">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="dob" type="date">
            <xs:element name="Address" type="AddressType">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="Line1" type="String">
                  <xs:element name="Line2" type="String">
                </xs:sequence>
              </xs:complexType>
            </xs:sequence>
          </xs:complexType>
        </xs:sequence>
      </xs:complexType>
      <xs:element name="supplier">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="phone" type="int">
            <xs:element name="Address" type="AddressType">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="Line1" type="String">
                  <xs:element name="Line2" type="String">
                </xs:sequence>
              </xs:complexType>
            </xs:sequence>
          </xs:complexType>
        </xs:sequence>
      </xs:complexType>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

2. Write an XML Schema (xsd file) document specifying the structure of the XML document below.

```
<?xml version="1.0" encoding="UTF-8"?>
<books
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="books.xsd">
  <book>
    <author auid="123">
      <name>
        <fname>Abbey</fname>
        <lname>Deitel</lname>
      </name>
      <contact>
        <email>aaa@bbb.com</email>
        <phone>562-1119999</phone>
      </contact>
    </author>
    <bookinfo>
      <title>C++ Programming</title>
      <edition>1st Edition</edition>
```

```
      <isbn>123456</isbn>
      <price>50.90</price>
    </bookinfo>
  </book>
  <book>
    <author auid="623">
      <name>
        <fname>John</fname>
        <lname>Nguyen</lname>
      </name>
      <contact>
        <email>eee@bbb.com</email>
        <phone>562-2229999</phone>
      </contact>
    </author>
    <bookinfo>
      <title>Java Programming</title>
      <edition>3th Edition</edition>
      <isbn>563456</isbn>
      <price>80.00</price>
    </bookinfo>
  </book>
</books>
```

Schema requirements:

- a) Define data types using simpleType and complexType
- b) data types:
 - price - decimal
 - isbn - positiveInteger
 - author id – string
 - example-001, 349 [use pattern]
 - email-string
 - example- mike_12@abc.com, sam.2g@xyz.ac.in [use pattern]
 - phone number-string
 - example- 0532-1278912, 011-2345200 [use pattern]
- c) Minimum number of books: 1 to unbounded
- d) Author id is required.
- e) The child elements must appear in sequence.

DTD code<>

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

```
<!ELEMENT books (book)+>
```

```
<!ELEMENT book (author | bookinfo)+>
```

```
<!ELEMENT author (name | contact)+>
```

```
<!ELEMENT name (fname | lname)>
```

```
<!ELEMENT fname (#PCDATA)>
```

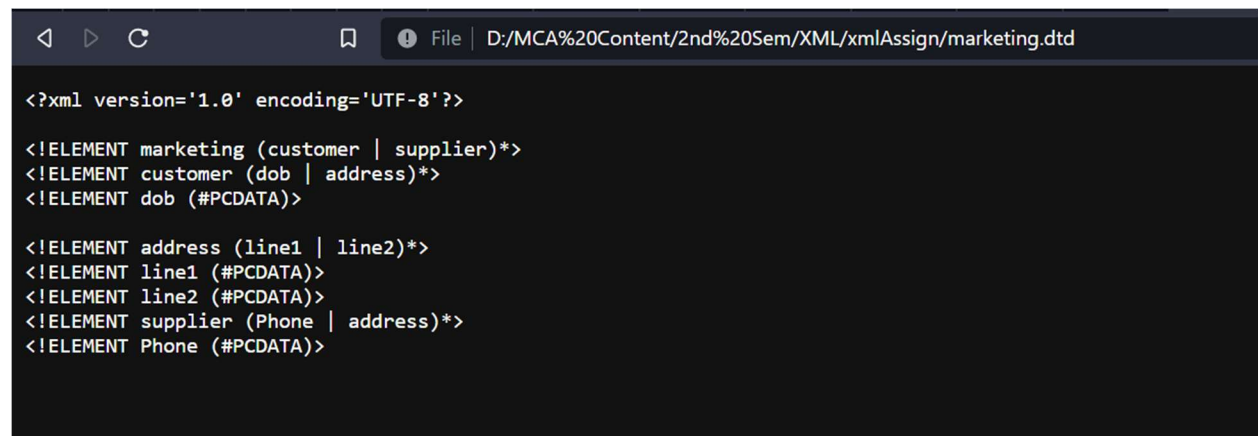
```
<!ELEMENT lname (#PCDATA)>
```

```
<!ELEMENT contact (email | phone)+>
```

```
<!ELEMENT email (#PCDATA)>
```

```
<!ELEMENT phone (#PCDATA)>
<!ELEMENT bookinfo (title | edition | isbn | price)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT edition (#PCDATA)>
<!ELEMENT isbn (#PCDATA)>
<!ELEMENT price (#PCDATA)>
<!ATTLIST author auid CDATA #REQUIRED>
```

Output:-

A screenshot of a text editor window with a dark background. The title bar shows the file path: D:/MCA%20Content/2nd%20Sem/XML/xmlAssign/marketing.dtd. The editor contains the following XML DTD code:

```
<?xml version='1.0' encoding='UTF-8'?>
<!ELEMENT marketing (customer | supplier)*>
<!ELEMENT customer (dob | address)*>
<!ELEMENT dob (#PCDATA)>
<!--
<!ELEMENT address (line1 | line2)*>
<!ELEMENT line1 (#PCDATA)>
<!ELEMENT line2 (#PCDATA)>
-->
<!ELEMENT supplier (Phone | address)*>
<!ELEMENT Phone (#PCDATA)>
```

xsd code<>

```
<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="books">
  <xs:complexType>
```



```

<xs:sequence>
  <xs:element name="book" minOccurs="1">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="author">
          <xs:complexType>
            <xs:sequence>
              <xs:extension base="xs:string">
                <xs:attribute name="auid" use="required" type="xs:ID"/>
              </xs:extension>
            <xs:element name="name">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="fname" type="xs:string" />
                  <xs:element name="lname" type="xs:string"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          <xs:element name="contact">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="email">
                  <xs:simpleType>
                    <xs:restriction base="xs:string">

```

<xs:pattern value="/^([a-z\d\._]+)@([a-z]+)\.([a-z]{2,8})?\$/"/>

<xs:pattern value="/^([a-z\d\._]+)@([a-z]+)\.([a-z]{2,8})\.([a-z]{2,8})?\$/"/>

</xs:restriction>

</xs:simpleType>

</xs:element>

<xs:element name="phone">

<xs:simpleType>

<xs:restriction base="xs:nonNegativeInteger">

<xs:pattern value="/^([0-9]{3})-([0-9]{7})\$/"/>

<!-- <xs:pattern value="/^\d{4}-\d{6}\$/"/> -->

</xs:restriction>

</xs:simpleType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="bookinfo">

<xs:complexType>

<xs:sequence>

<xs:element name="title" type="xs:string"/>

```
        <xs:element name = "edition" type = "xs:string"/>
        <xs:element name = "isbn" type = "xs:positiveInteger"/>
        <xs:element name = "price" type = "xs:decimal"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

Output:-

```
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
<xs:element name="marketing">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="customer">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="dob" type="xs:date">
            </xs:element>
            <xs:element name="Address">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="Line1" type="xs:string">
                  </xs:element>
                  <xs:element name="Line2" type="xs:string">
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="supplier">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="phone" type="xs:nonNegativeInteger">
            </xs:element>
            <xs:element name="Address">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="Line1" type="xs:string">
                  </xs:element>
                  <xs:element name="Line2" type="xs:string">
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```