**EXPERIMENT NUMBER- 07**

**AIM**

* To write a DTD for the given XML file.

**DESCRIPTION**

* For the given XML file, we will create an external DTD file and import that file to the given XML file.
* XML is a markup language similar to HTML, but without predefined tags to use. Instead, we will define our tags designed specifically for our needs. We use XML to store data in a format that can be stored, searched, and shared. Here we use DTD for the document structure. DTD defines the valid building blocks of an XML document. It defines the document structure with a list of validated elements and attributes.

**PROGRAM(S)**

* **XML File**

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

<!DOCTYPE stock SYSTEM "1.dtd">

<stock>

<new-car>

<model>Fiat</model>

<price>12000</price>

</new-car>

<used-car>

<model>Fiat Bravo</model>

<price>4000</price>

<mileage>1000</mileage>

<condition>Good</condition>

</used-car>

<used-car>

<model>Ferrari</model>

<price>400000</price>

<mileage>100</mileage>

</used-car>

</stock>

* **DTD File**

<!ELEMENT stock (new-car,(used-car)+)>

<!ELEMENT new-car (model,price)>

<!ELEMENT used-car (model,price,mileage,(condition)?)>

<!ELEMENT model (#PCDATA)>

<!ELEMENT price (#PCDATA)>

<!ELEMENT mileage (#PCDATA)>

<!ELEMENT condition (#PCDATA)>

**RESULTS/OUTPUT**

****

**CONCLUSION**

* The DTD file for the given XML code is valid.

**EXPERIMENT NUMBER- 08**

**AIM**

* To write a DTD and XSD files for the given XML code.

**DESCRIPTION**

* For the given XML file, we will create an external DTD file and import that file to the given XML file. Similarly, we will create an XSD file and link it to the XML file.
* XML is a markup language similar to HTML, but without predefined tags to use. Instead, we will define our tags designed specifically for our needs. We use XML to store data in a format that can be stored, searched, and shared. Here we use DTD for the document structure. DTD defines the valid building blocks of an XML document. It defines the document structure with a list of validated elements and attributes. XSD, known as XML Schema Definition, is a W3C recommendation that specifies how to formally describe the elements in an XML document. The purpose of an XML Schema is to define the legal building blocks of an XML document. DTD is mainly used to define the structure of an XML file, whereas XSD is mainly used to describe the structure and contents of an XML file. Also, DTD is not extensible, whereas XSD is extensible.

**PROGRAM(S)**

* **XML File**

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

<!DOCTYPE bookstore SYSTEM "2.dtd">

<bookstore>

<book price="125" ISBN="string" publicationdate="2016-02-27">

<title>String</title>

<author>

<first-name>string</first-name>

<last-name>string</last-name>

</author>

<genre>string</genre>

</book>

<book ISBN="string" price="124">

<title>stirng</title>

<author>

<first-name>string</first-name>

<last-name>string</last-name>

</author>

</book>

</bookstore>

* **DTD File**

<!ELEMENT bookstore (book)+>

<!ELEMENT book (title,author,(genre)?)>

<!ELEMENT author (first-name,last-name)>

<!ELEMENT title (#PCDATA)>

<!ELEMENT first-name (#PCDATA)>

<!ELEMENT last-name (#PCDATA)>

<!ELEMENT genre (#PCDATA)>

<!ATTLIST book price CDATA #REQUIRED>

<!ATTLIST book ISBN CDATA #REQUIRED>

<!ATTLIST book publicationdate CDATA #IMPLIED>

* **XSD File**

<?xml version="1.0"?>

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

targetNamespace="https://www.w3schools.com"

xmlns="https://www.w3schools.com"

elementFormDefault="qualified">

<xs:element name="bookstore">

<xs:complexType>

<xs:sequence>

<xs:element name="book" maxOccurs="unbounded">

<xs:complexType>

<xs:sequence>

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

<xs:element name="author">

<xs:complexType>

<xs:sequence>

<xs:element name="first-name" type="xs:string"/>

<xs:element name="last-name" type="xs:string"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="genre" type="xs:string" minOccurs="0"/>

</xs:sequence>

<xs:attribute name="price" type="xs:decimal" use="required"/>

<xs:attribute name="ISBN" type="xs:string" use="required"/>

<xs:attribute name="publicationdate" type="xs:date" use="optional"/>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:schema>

**RESULTS/OUTPUT**





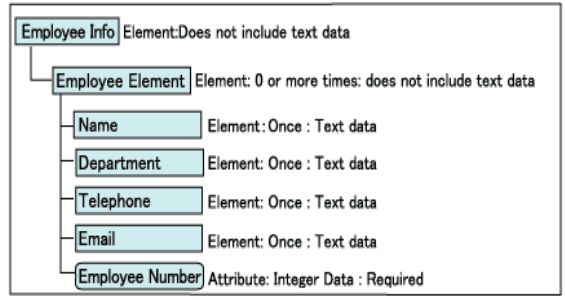
**CONCLUSION**

* DTD and XSD files are valid for the given XML file.

**EXPERIMENT NUMBER- 09**

**AIM**

* To write DTD, XSD, and XML files for the given structure.



**DESCRIPTION**

* For the given XML file, we will create an external DTD file and import that file to the given XML file. Similarly, we will create an XSD file and link it to the XML file.
* XML is a markup language similar to HTML, but without predefined tags to use. Instead, we will define our tags designed specifically for our needs. We use XML to store data in a format that can be stored, searched, and shared. Here we use DTD for the document structure. DTD defines the valid building blocks of an XML document. It defines the document structure with a list of validated elements and attributes. Now XSD known as XML Schema Definition is a W3C

**PROGRAM(S)**

* **XML File**

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

<!DOCTYPE EmployeeInfo SYSTEM "3.dtd">

<EmployeeInfo>

<employee empno="166">

<name>string</name>

<department>stirng</department>

<telephone>string</telephone>

<email>string</email>

</employee>

</EmployeeInfo>

* **DTD File**

<!ELEMENT EmployeeInfo (employee)>

<!ELEMENT employee (name,department,telephone,email)>

<!ELEMENT name (#PCDATA)>

<!ELEMENT department (#PCDATA)>

<!ELEMENT telephone (#PCDATA)>

<!ELEMENT email (#PCDATA)>

<!ATTLIST employee empno CDATA #REQUIRED>

* **XSD File**

<?xml version="1.0"?>

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

targetNamespace="https://www.w3schools.com"

xmlns="https://www.w3schools.com"

elementFormDefault="qualified">

<xs:element name="employeeinfo">

<xs:complexType>

<xs:sequence>

<xs:element name="employeeelement">

<xs:complexType>

<xs:sequence>

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

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

<xs:element name="telephone" type="xs:integer"/>

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

<xs:element name="employeeno" type="xs:integer"/>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:schema>

**RESULTS/OUTPUT**





**CONCLUSION**

* The DTD and XSD files for the given XML file are valid.