Course Code: CT2207	Course Name: Lab: Web Technologies

Name of Students: Badal Wanjari	**Semester/ Section:** 3rd / B
Roll No: 140	**Enroll No: 20011045**

Practical No: 7

Aim: Introduction to XML. Program to demonstrate the use of External and Internal DTD.

Write Theory regarding Internal and External DTD and Program to demonstrate Internal and External DTD.

THEORY:

An XML DTD can be either specified inside the document, or it can be kept in a separate document and then the document can be linked to the DTD document to use it.

Syntax

Basic syntax of a DTD is as follows −

```
<!DOCTYPE element DTD identifier
[
   declaration1
   declaration2
   ........
]>
```

In the above syntax −

- **DTD** starts with <!DOCTYPE delimiter.
- An **element** tells the parser to parse the document from the specified root element.
- **DTD identifier** is an identifier for the document type definition, which may be the path to a file on the system or URL to a file on the internet. If the DTD is pointing to external path, it is called **external subset.**
- The **square brackets [ ]** enclose an optional list of entity declarations called **internal subset**.

- ## Internal DTD

A DTD is referred to as an internal DTD if elements are declared within the XML files. To reference it as internal DTD, *standalone* attribute in XML declaration must be set to **yes**. This means the declaration works independent of external source.

Syntax

The syntax of internal DTD is as shown −

<!DOCTYPE root-element [element-declarations]>

where *root-element* is the name of root element and *element-declarations* is where you declare the elements.

Example

Following is a simple example of internal DTD −

```
<?xml version = "1.0" >
<!DOCTYPE address [
  <!ELEMENT address (name,company,phone)>
  <!ELEMENT name (#PCDATA)>
  <!ELEMENT company (#PCDATA)>
  <!ELEMENT phone (#PCDATA)>
]>
<address>
  <name>Tanmay Patil</name>
  <company>TutorialsPoint</company>
  <phone>(011) 123-4567</phone>
</address>
```

Let us go through the above code −

Start Declaration − Begin the XML declaration with following statement.

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

DTD − Immediately after the XML header, the *document type declaration* follows, commonly referred to as the DOCTYPE −

<!DOCTYPE address [

The DOCTYPE declaration has an exclamation mark (!) at the start of the element name. The DOCTYPE informs the parser that a DTD is associated with this XML document.

DTD Body − The DOCTYPE declaration is followed by body of the DTD, where you declare elements, attributes, entities, and notations −

<!ELEMENT address (name,company,phone)>

<!ELEMENT name (#PCDATA)>

<!ELEMENT company (#PCDATA)>

<!ELEMENT phone_no (#PCDATA)>

Several elements are declared here that make up the vocabulary of the <name> document. <!ELEMENT name (#PCDATA)> defines the element *name* to be of type "#PCDATA". Here #PCDATA means parse-able text data.

End Declaration − Finally, the declaration section of the DTD is closed using a closing bracket and a closing angle bracket (]>). This effectively ends the definition, and thereafter, the XML document follows immediately.

Rules

- The document type declaration must appear at the start of the document (preceded only by the XML header) - it is not permitted anywhere else within the document.
- Similar to the DOCTYPE declaration, the element declarations must start with an exclamation mark.
- The Name in the document type declaration must match the element type of the root element.

- ## External DTD

In external DTD elements are declared outside the XML file. They are accessed by specifying the system attributes which may be either the legal *.dtd* file or a valid URL. To reference it as external DTD, *standalone* attribute in the XML declaration must be set as **no**. This means, declaration includes information from the external source.

Syntax

Following is the syntax for external DTD −

<!DOCTYPE root-element SYSTEM "file-name">

where *file-name* is the file with *.dtd* extension.

Example

The following example shows external DTD usage −

```
<?xml version = "1.0" encoding = "UTF-8" standalone = "no" ?>
<!DOCTYPE address SYSTEM "address.dtd">


<address>
  <name>Tanmay Patil</name>
  <company>TutorialsPoint</company>
  <phone>(011) 123-4567</phone>
</address>
```

The content of the DTD file **address.dtd** are as shown −

```
<!ELEMENT address (name,company,phone)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT company (#PCDATA)>
<!ELEMENT phone (#PCDATA)>
```

Types

You can refer to an external DTD by either using **system identifiers** or **public identifiers**.

System Identifiers

A system identifier enables you to specify the location of an external file containing DTD declarations. Syntax is as follows −

```
<!DOCTYPE name SYSTEM "address.dtd" [...]>
```

As you can see it contains keyword SYSTEM and a URI reference pointing to the location of the document.

Public Identifiers

Public identifiers provide a mechanism to locate DTD resources and are written as below −

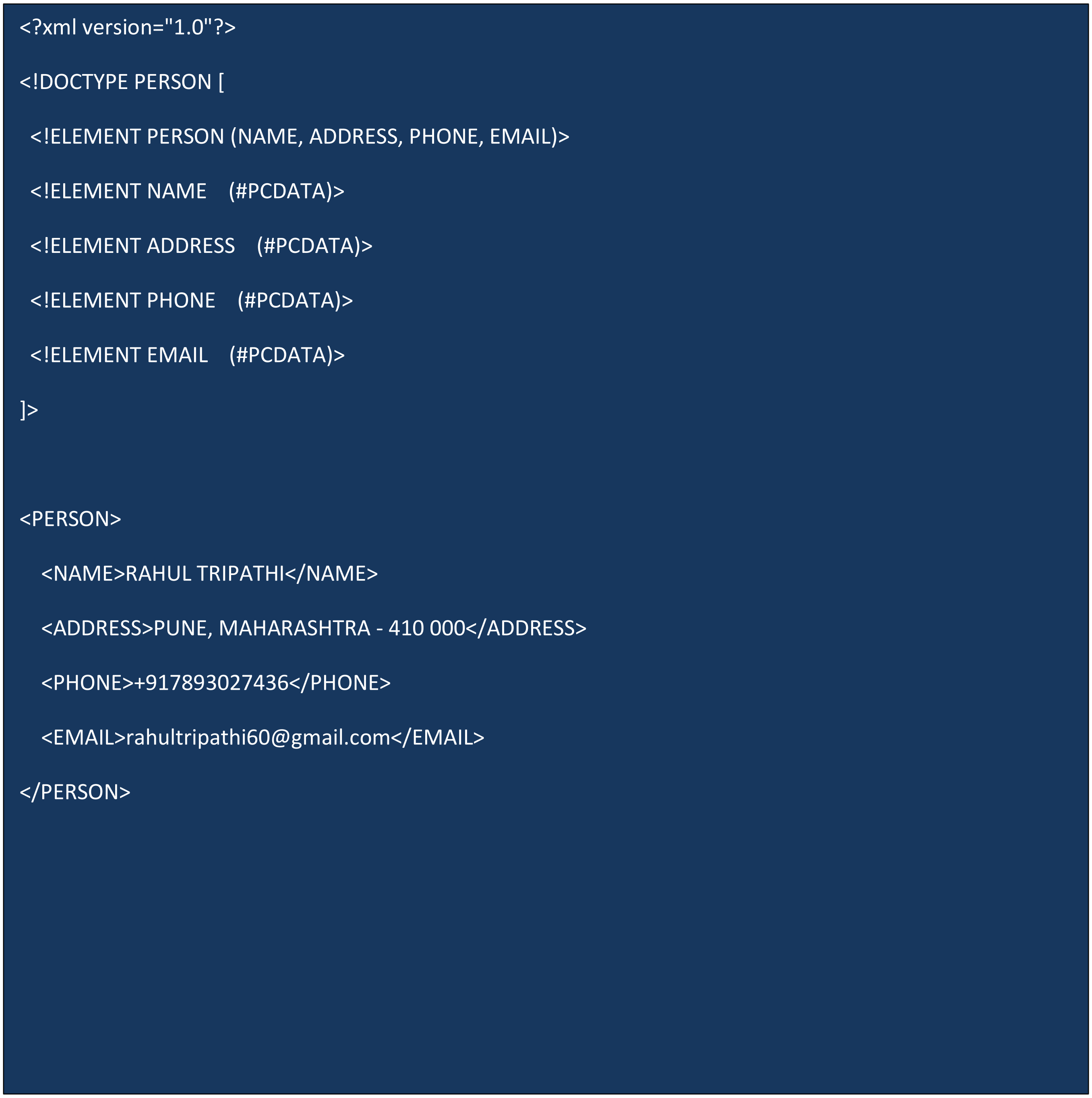
```
<!DOCTYPE name PUBLIC "-//Beginning XML//DTD Address Example//EN">
```

As you can see, it begins with keyword PUBLIC, followed by a specialized identifier. Public identifiers are used to identify an entry in a catalog. Public identifiers can follow any format; however, a commonly used format is called *Formal Public Identifiers, or FPIs.*

OUTPUT SCREEN SHOTS:

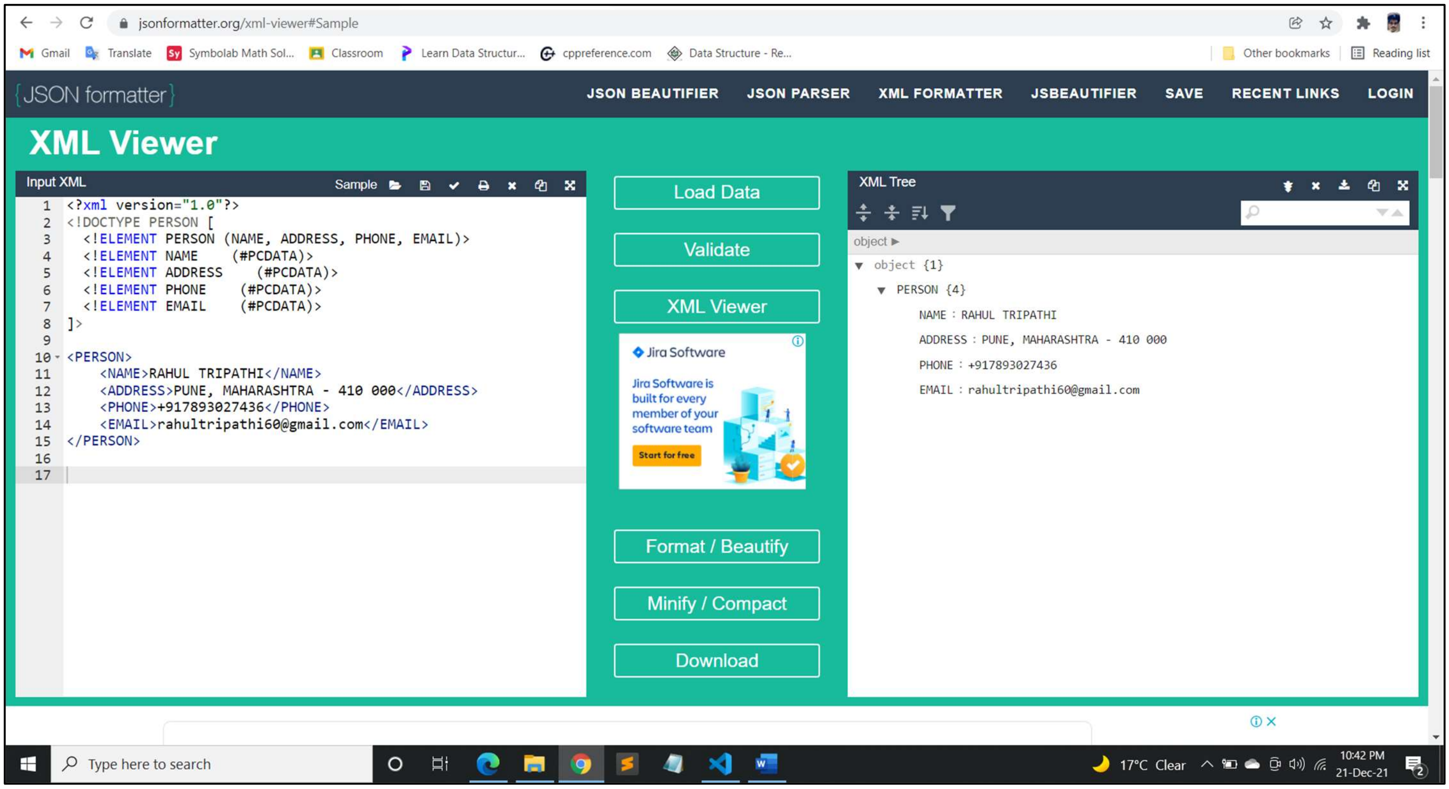
- **Internal DTD**

Program:



```
<?xml version="1.0"?>

<!DOCTYPE PERSON [

 <!ELEMENT PERSON (NAME, ADDRESS, PHONE, EMAIL)>

 <!ELEMENT NAME    (#PCDATA)>

 <!ELEMENT ADDRESS    (#PCDATA)>

 <!ELEMENT PHONE    (#PCDATA)>

 <!ELEMENT EMAIL    (#PCDATA)>

]>


<PERSON>

   <NAME>RAHUL TRIPATHI</NAME>

   <ADDRESS>PUNE, MAHARASHTRA - 410 000</ADDRESS>

   <PHONE>+917893027436</PHONE>

   <EMAIL>rahultripathi60@gmail.com</EMAIL>

</PERSON>
```

Output:



```
<?xml version="1.0"?>
<!DOCTYPE PERSON [
  <!ELEMENT PERSON (NAME, ADDRESS, PHONE, EMAIL)>
  <!ELEMENT NAME    (#PCDATA)>
  <!ELEMENT ADDRESS    (#PCDATA)>
  <!ELEMENT PHONE    (#PCDATA)>
  <!ELEMENT EMAIL    (#PCDATA)>
]>

<PERSON>
    <NAME>RAHUL TRIPATHI</NAME>
    <ADDRESS>PUNE, MAHARASHTRA - 410 000</ADDRESS>
    <PHONE>+917893027436</PHONE>
    <EMAIL>rahultripathi60@gmail.com</EMAIL>
</PERSON>

```

```
▼ object {1}
  ▼ PERSON {4}
      NAME : RAHUL TRIPATHI
      ADDRESS : PUNE, MAHARASHTRA - 410 000
      PHONE : +917893027436
      EMAIL : rahultripathi60@gmail.com
```

- **External DTD**

Program:

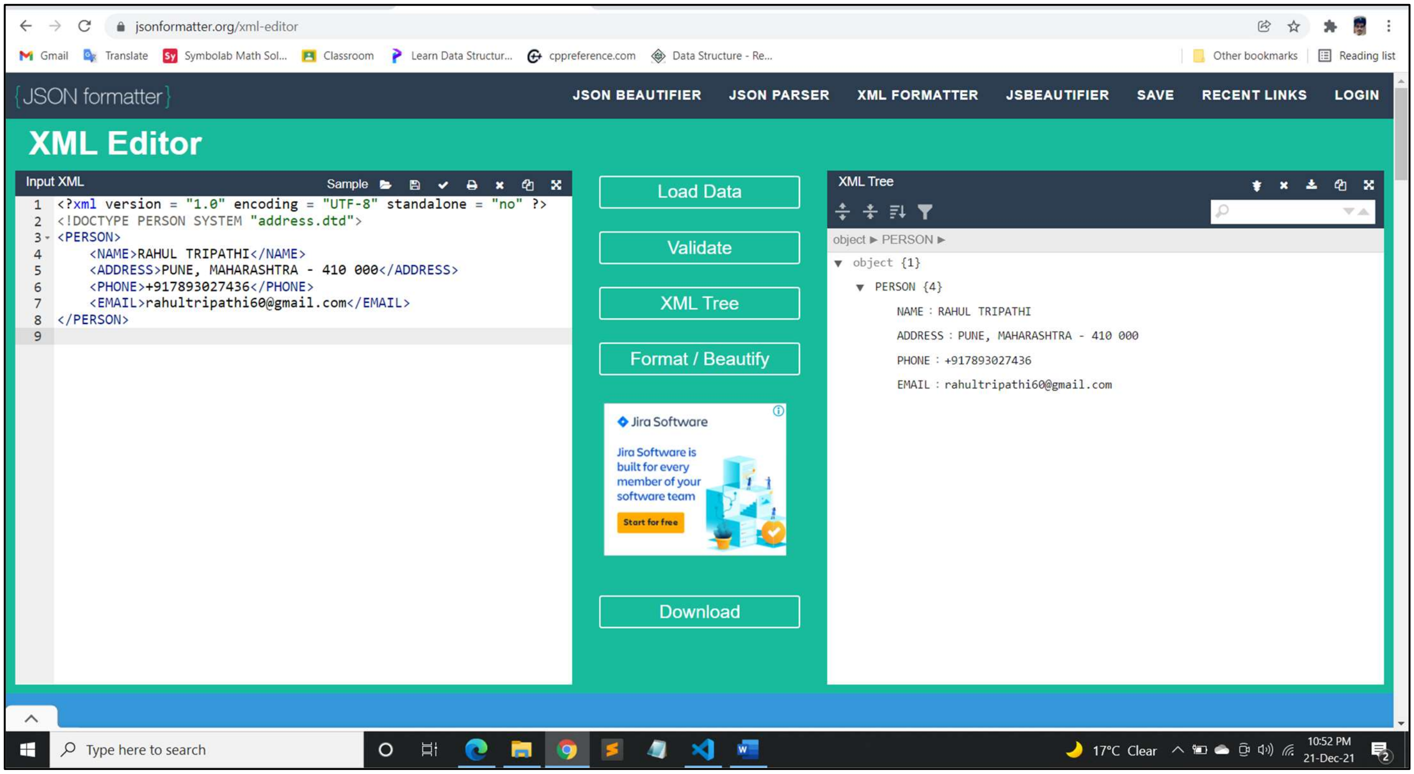
person.xml file:

```
<?xml version = "1.0" encoding = "UTF-8" standalone = "no" ?>

<!DOCTYPE PERSON SYSTEM "info.dtd">

<PERSON>

   <NAME>RAHUL TRIPATHI</NAME>

   <ADDRESS>PUNE, MAHARASHTRA - 410 000</ADDRESS>

   <PHONE>+917893027436</PHONE>

   <EMAIL>rahultripathi60@gmail.com</EMAIL>
```

info.dtd file:

```
<!ELEMENT PERSON (NAME, ADDRESS, PHONE, EMAIL)>

 <!ELEMENT NAME    (#PCDATA)>

 <!ELEMENT ADDRESS    (#PCDATA)>

 <!ELEMENT PHONE    (#PCDATA)>

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

Output:



```
<?xml version = "1.0" encoding = "UTF-8" standalone = "no" ?>
<!DOCTYPE PERSON SYSTEM "address.dtd">
<PERSON>
    <NAME>RAHUL TRIPATHI</NAME>
    <ADDRESS>PUNE, MAHARASHTRA - 410 000</ADDRESS>
    <PHONE>+917893027436</PHONE>
    <EMAIL>rahultripathi60@gmail.com</EMAIL>
</PERSON>
```

```
object ► PERSON ►
▼ object {1}
  ▼ PERSON {4}
       NAME : RAHUL TRIPATHI
       ADDRESS : PUNE, MAHARASHTRA - 410 000
       PHONE : +917893027436
       EMAIL : rahultripathi60@gmail.com
```

CONCLUSION:

Using reference of internet, I have successfully completed Practical-7.