Week-5 XML file which will display the Book information

Week-5: **Date:**

AIM: Write an XML file which will display the Book information.

It includes the following:

1) Title of the book
2) Author Name
3) ISBN number
4) Publisher name
5) Edition
6) Price

Write a Document Type Definition (DTD) to validate the above XML file.

Display the XML file as follows.

The contents should be displayed in a table. The header of the table should be in color GREY. And the Author names column should be displayed in one color and should be capitalized and in bold.

Use your own colors for remaining columns.

Use XML schemas XSL and CSS for the above purpose.

DESCRIPTION:

DTD vs XML Schema

The DTD provides a basic grammar for defining an XML Document in terms of the metadata that comprise the shape of the document. An XML Schema provides this, plus a detailed way to define what the data can and cannot contain. It provides far more control for the developer over what is legal, and it provides an Object Oriented approach, with all the benefits this entails.

Many systems interfaces are already defined as a DTD. They are mature definitions, rich and complex. The effort in re-writing the definition may not be worthwhile.

DTD is also established, and examples of common objects defined in a DTD abound on the Internet -- freely available for re-use. A developer may be able to use these to define a DTD more quickly than they would be able to accomplish a complete re-development of the core elements as a new schema.

Finally, you must also consider the fact that the XML Schema is an XML document. It has an XML Namespace to refer to, and an XML DTD to define it. This is all overhead. When a parser examines the document, it may have to link this all in, interpret the DTD for the Schema, load the namespace, and validate the schema, etc., all *before* it can parse the actual XML document in question. If you're using XML as a protocol between two systems that are in heavy use, and need a quick response, then this overhead may seriously degrade performance.

- **Write a Document Type Definition (DTD) to validate the XML file.**

PROGRAM:

XML document (bookstore.xml)

```
<bookstore>
        <book>
                <title>web programming</title>
                <author>chrisbates</author>
                <ISBN>123-456-789</ISBN>
                <publisher>wiley</publisher>
                <edition>3</edition>
                <price>350</price>
        </book>
        <book>
                <title>internet worldwideweb</title>
                <author>ditel&amp;ditel</author>
                <ISBN>123-456-781</ISBN>
                <publisher>person</publisher>
                <edition>3</edition>
                <price>450</price>
        </book>
</bookstore>
```

XML document Validation using DTD

DTD document (bookstore.dtd)

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT bookstore (book+)>
<!ELEMENT book (title,author,ISBN,publisher,edition,price)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT ISBN (#PCDATA)>
```

<!ELEMENT publisher (#PCDATA)>

<!ELEMENT edition (#PCDATA)>

<!ELEMENT price (#PCDATA)>

Bookstore.xml

<!DOCTYPE bookstore SYSTEM "C:\Documents and Settings\Administrator\My Documents\bookstore.dtd">

```
<bookstore>
	<book>
		<title>web programming</title>
		<author>chrisbates</author>
		<ISBN>123-456-789</ISBN>
		<publisher>wiley</publisher>
		<edition>3</edition>
		<price>350</price>
	</book>
	<book>
		<title>internet worldwideweb</title>
		<author>ditel&amp;ditel</author>
		<ISBN>123-456-781</ISBN>
		<publisher>person</publisher>
		<edition>3</edition>
		<price>450</price>
	</book>
</bookstore>
```

XML document Validation using DTD

XML Schema (bookstore.xsd)

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema elementFormDefault="qualified" attributeFormDefault="unqualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
        <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>
<xs:element name="author"  type="xs:string"></xs:element>
<xs:element name="ISBN"    type="xs:string"></xs:element>
<xs:element name="publisher"          type="xs:string"></xs:element>
<xs:element name="edition"  type="xs:int"></xs:element>
<xs:element name="price"     type="xs:decimal"></xs:element>
        </xs:sequence>
</xs:complexType>


</xs:element>

                </xs:sequence>
                </xs:complexType>

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

Bookstore.xml

<bookstore xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:\Documents and Settings\Administrator\My
Documents\bookstore.xsd">

```
    <book>
        <title>web programming</title>
        <author>chrisbates</author>
        <ISBN>123-456-789</ISBN>
        <publisher>wiley</publisher>
        <edition>3</edition>
        <price>350</price>
    </book>
    <book>
        <title>internet worldwideweb</title>
        <author>ditel&amp;ditel</author>
        <ISBN>123-456-781</ISBN>
        <publisher>person</publisher>
        <edition>3</edition>
        <price>450</price>
    </book>
</bookstore>
```

- **Display the XML file as follows.**

PROGRAM:

XML:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="bookstore.xsl"?>

<bookstore>
<book>
  <title>Everyday Italian</title>
  <author>Giada De Laurentiis</author>
  <year>2005</year>
  <price>30.00</price>
</book>
```

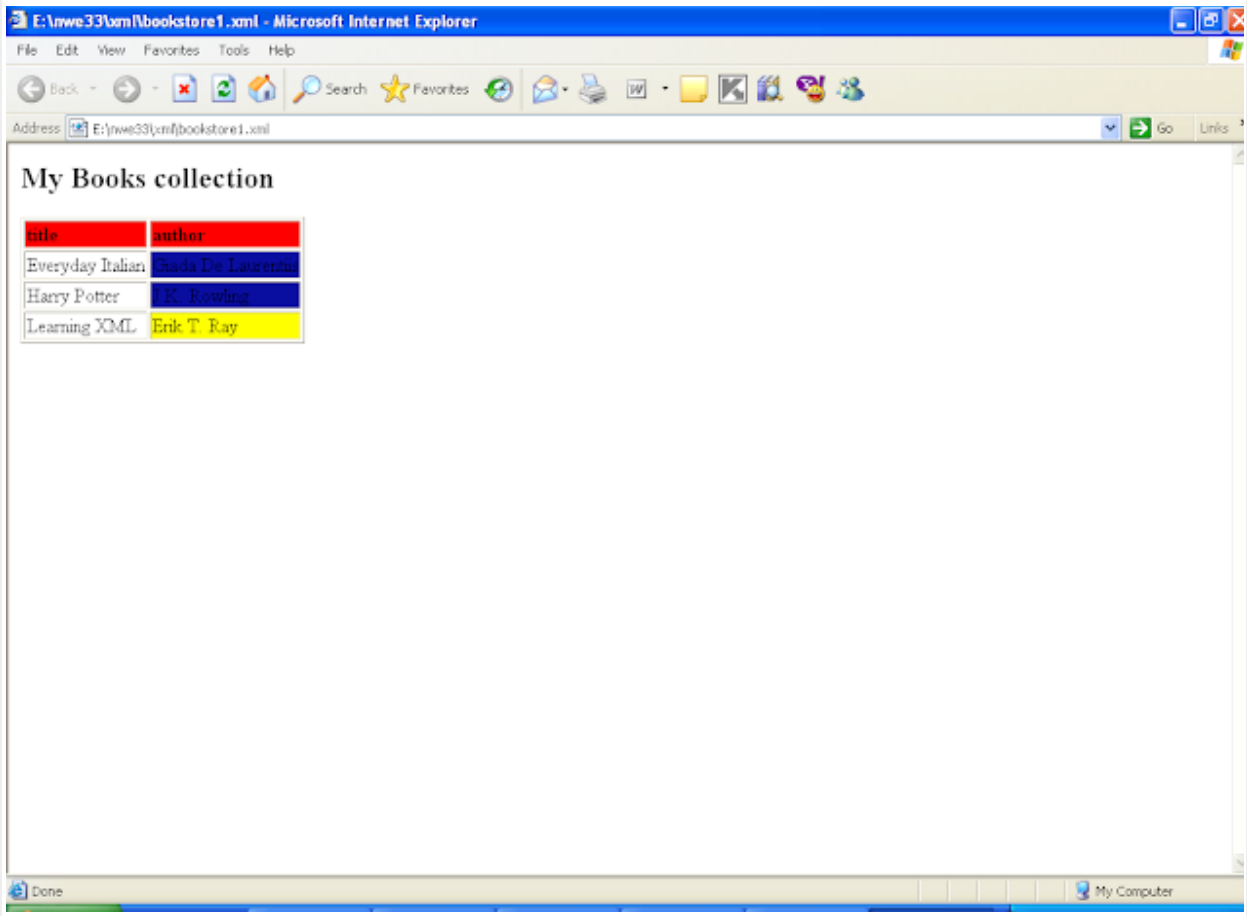
```
<book>
  <title>Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
</book>
<book>
  <title>Learning XML</title>
  <author>Erik T. Ray</author>
  <year>2003</year>
  <price>39.95</price>
</book>
</bookstore>
```

XSL:

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<body>
<h2> My Books collection</h2>
<table border="1">
<tr bgcolor="red">
<th align="left">title</th>
<th align="left">author</th>
</tr>
<xsl:for-each select="bookstore/book">
<tr>
<td><xsl:value-of select="title"/></td>
<xsl:choose>
<xsl:when test="price &gt; 30">
<td bgcolor="yellow"><xsl:value-of select="author"/></td>
</xsl:when>
<xsl:when test="price &gt; 10">
<td bgcolor="magneta"><xsl:value-of select="author"/></td>
</xsl:when>
<xsl:otherwise>
<td><xsl:value-of select="author"/></td>
</xsl:otherwise>
</xsl:choose>
</tr>
</xsl:for-each>
```

</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

OUTPUT:



RESULT: Thus the XML stylesheets are successfully used to display the content in a table format.