

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
|  | | CSLR 51 : DBMS LAB-9 | | | | |  | |
|  |  | | | | | | |  |
|  | | | |  |  | | | |
|  | | | | Roll no. : 106119100Name : Rajneesh PandeySection : CSE-B |  | | | |
|  | | |  | | |  | | |

1)

a. Demonstrating DTD:

**1. filename :** *bookstore.dtd*

<!ELEMENT bookstore (book+)>

<!ELEMENT book (title,author,category,isbn,publisher,edition,price)>

<!ELEMENT title (#PCDATA)>

<!ELEMENT author (#PCDATA)>

<!ELEMENT category (#PCDATA)>

<!ELEMENT isbn (#PCDATA)>

<!ELEMENT publisher (#PCDATA)>

<!ELEMENT edition (#PCDATA)>

<!ELEMENT price (#PCDATA)>

**2. filename :** *bookstore.xml*

<?xml version="1.0"?>

<!DOCTYPE bookstore SYSTEM "bookstore.dtd">

    <bookstore>

            <book>

                <title>XML Developer's Guide</title>

                <author>Gambardella, Matthew</author>

                <category>textbook</category>

                <isbn>111111</isbn>

                <publisher>wiley</publisher>

                <edition>second</edition>

                <price>20.00</price>

            </book>

            <book>

                <title>Maeve Ascendant</title>

                <author>Corets, Eva</author>

                <category>textbook</category>

                <isbn>222222</isbn>

                <publisher>tata</publisher>

                <edition>second</edition>

                <price>60.00</price>

            </book>

            <book>

                <title>Midnight Rain</title>

                <author>Ralls, Kim</author>

                <category>fiction</category>

                <isbn>333333</isbn>

                <publisher>O'relilly</publisher>

                <edition>second</edition>

                <price>100.00</price>

            </book>

    </bookstore>

Graphical user interface, text, application

Description automatically generated

b. Demonstrating XSD:

**1. filename :** *bookstore.xml*

<?xml version="1.0" encoding="iso-8859-1"?>

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

    <xs:element name="bookstore">

        <xs:complexType>

            <xs:sequence>

               <xs:element name="book" minOccurs="1" maxOccurs="unbounded" />

            </xs:sequence>

        </xs:complexType>

    </xs:element>

    <xs:element name="book">

        <xs:complexType>

            <xs:sequence>

                <xs:element ref="title" minOccurs="1" maxOccurs="1" />

                <xs:element ref="author" minOccurs="1" maxOccurs="1" />

                <xs:element ref="category" minOccurs="1" maxOccurs="1" />

                <xs:element ref="isbn" minOccurs="1" maxOccurs="1" />

                <xs:element ref="publisher" minOccurs="1" maxOccurs="1" />

                <xs:element ref="edition" minOccurs="1" maxOccurs="1" />

                <xs:element ref="price" minOccurs="1" maxOccurs="1" />

            </xs:sequence>

        </xs:complexType>

    </xs:element>

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

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

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

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

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

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

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

</xs:schema>

**2. filename :** *bookstore.css*

bookstore {

  color: orange;

}

book {

  color: Red;

}

title {

  color: blue;

  font-weight: bold;

  margin-left: 10pt;

  display: block;

}

author {

  color: red;

  font-weight: bold;

  margin-left: 10pt;

}

category {

  color:grey;

  font-weight: bold;

  margin-left: 10pt;

}

isbn {

  color: green;

  font-weight: bold;

  margin-left: 10pt;

}

edition {

  color: red;

  font-weight: bold;

  margin-left: 10pt;

}

publisher {

  color: green;

  font-weight: bold;

  margin-left: 10pt;

}

**3. filename :** *bookstore.xsl*

<?xml version="1.0"?>

<?xml-stylesheet type="text/css" href="bookstore.css"?>

<bookstore xmlns:xsi="http:www.w3.org/2001/xmlschema-instance" xsi:noNamespaceSchemaLocation="bookstore.xsd">

    <book>

        <title>XML Developer's Guide</title>

        <author>Gambardella, Matthew</author>

        <category>textbook</category>

        <isbn>111111</isbn>

        <publisher>wiley</publisher>

        <edition>second</edition>

        <price>20.00</price>

    </book>

    <book>

        <title>Maeve Ascendant</title>

        <author>Corets, Eva</author>

        <category>textbook</category>

        <isbn>222222</isbn>

        <publisher>tata</publisher>

        <edition>second</edition>

        <price>60.00</price>

    </book>

    <book>

        <title>Midnight Rain</title>

        <author>Ralls, Kim</author>

        <category>fiction</category>

        <isbn>333333</isbn>

        <publisher>O'relilly</publisher>

        <edition>second</edition>

        <price>100.00</price>

    </book>

</bookstore>

Graphical user interface, text, application, email

Description automatically generated

c. Demonstrating XSL:

**1. filename :** *bookstore.xsl*

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

    <xsl:template match="/">

        <html>

            <head>

                <title>bookstore</title>

            </head>

            <body>

                <table border="1">

                    <tr>

                        <th>title</th>

                        <th>author</th>

                        <th>category</th>

                        <th>isbn</th>

                        <th>publisher</th>

                        <th>edition</th>

                        <th>price</th>

                    </tr>

                    <xsl:for-each select="/bookstore/book">

                        <tr>

                            <td bgcolor="green">

                                <xsl:value-of select="title" />

                            </td>

                            <td bgcolor="red">

                                <xsl:value-of select="author" />

                            </td>

                            <td bgcolor="grey">

                                <xsl:value-of select="category" />

                            </td>

                            <td bgcolor="cyan">

                                <xsl:value-of select="isbn" />

                            </td>

                            <td bgcolor="yellow">

                                <xsl:value-of select="publisher" />

                            </td>

                            <td bgcolor="silver">

                                <xsl:value-of select="edition" />

                            </td>

                            <td bgcolor="blue">

                                <xsl:value-of select="price" />

                            </td>

                        </tr>

                    </xsl:for-each>

                </table>

            </body>

        </html>

    </xsl:template>

</xsl:stylesheet>

**2. filename :** *bookstore.xml*

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

<bookstore>

    <book>

        <title>XML Developer's Guide</title>

        <author>Gambardella, Matthew</author>

        <category>textbook</category>

        <isbn>111111</isbn>

        <publisher>wiley</publisher>

        <edition>second</edition>

        <price>20.00</price>

    </book>

    <book>

        <title>Maeve Ascendant</title>

        <author>Corets, Eva</author>

        <category>textbook</category>

        <isbn>222222</isbn>

        <publisher>tata</publisher>

        <edition>second</edition>

        <price>60.00</price>

    </book>

    <book>

        <title>Midnight Rain</title>

        <author>Ralls, Kim</author>

        <category>fiction</category>

        <isbn>333333</isbn>

        <publisher>O'relilly</publisher>

        <edition>second</edition>

        <price>100.00</price>

    </book>

</bookstore>

Table

Description automatically generated

2 ) Perform the following queries using XPath :

1. Find the title and price of non-fiction books with a price more than 50 USD

***Xpath query*** : /**bookstore**/**book**[**category**!="fiction" and **price**>50.00]/(**title** | **isbn**)

Graphical user interface, text, application

Description automatically generated

1. Find average price of textbooks

***Xpath query*** : *sum*(/**bookstore**/**book**[**category**="textbook"]/**price**/*number*(*text*())) div *count*(/**bookstore**/**book**[**category**="textbook"]/**price**)

Graphical user interface, text, application, Word

Description automatically generated

1. Find the titles of textbooks on XML

***Xpath query*** : /**bookstore**/**book**[**category**="textbook" and *contains*(**title**, "XML")]/**title**/*text*()

Graphical user interface, text, application

Description automatically generated

1. Perform the following queries using XQuery:
2. Create a new document which contain only the isbn and title of textbooks

***XQuery***

<textbook>  
 {  
 **for** **$book** **in** *doc*("bookstore.xml") // **book**   
 **where** **$book**[**category** = "textbook"]   
 **return** <textbook>{**$book**/**title**}{**$book**/**isbn**}</textbook>  
 }  
 </textbook>

Graphical user interface, application

Description automatically generated

1. Find the title and price of the book with isbn “222222”.

***XQuery :***

**for** **$book** **in** *doc*("bookstore.xml") // **book** **where** **$book**[**isbn**="222222"]  
 **return** <book>{ **$book**/**title**, **$book**/**price**}</book>

Graphical user interface, text, application

Description automatically generated

1. Produce a list of non-fictions with their title and price, sorted by price

***XQuery :***

<nonfiction-list>  
 {   
 **for** **$book** **in** *doc*("bookstore.xml")//**book**, **$title** **in** **$book**/**title**, **$price** **in** **$book**/**price**   
 **where** **$book**/**category**!="fiction" **order by** **$price**/*text*()   
 **return**  
 <nonfiction>{**$title**, **$price**}</nonfiction>  
 }  
</nonfiction-list>

Graphical user interface, text, application

Description automatically generated

1. Find title of the textbook with highest price

***XQuery :***

<textbooks>  
 { **let** **$prices** := *doc*("bookstore.xml")// **book**[**category**="textbook"]/**price** **let** **$max** := *max*(**$prices**)  
 **return**  
 <max-price-textbook **price**=**"**{**$max**}**"**>  
 {**for** **$book** **in** *doc*("bookstore.xml")// **book** **where** **$book**/**price** = **$max** **return** **$book**/**title**}  
 </max-price-textbook>  
 }  
</textbooks>

Graphical user interface, text, application

Description automatically generated