Lab-8

1. Develop XML document that will hold player (Like Cricket) collection with field for player-name, age, batting-average and highest-score, write suitable schema for the XML.

XML :-

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

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

<!DOCTYPE cricket SYSTEM "cricket.dtd">

<cricket>

    <player>

        <player\_name>Virat Kohli</player\_name>

        <balling\_avg>23.45</balling\_avg>

        <highest\_score>183</highest\_score>

    </player>

    <player>

        <player\_name>Rohit Sharma</player\_name>

        <balling\_avg>32.15</balling\_avg>

        <highest\_score>264</highest\_score>

    </player>

    <player>

        <player\_name>MS Dhoni</player\_name>

        <balling\_avg>12.67</balling\_avg>

        <highest\_score>183</highest\_score>

    </player>

    <player>

        <player\_name>Jasprit Bumrah</player\_name>

        <balling\_avg>22.65</balling\_avg>

        <highest\_score>10</highest\_score>

    </player>

    <player>

        <player\_name>Hardik Pandya</player\_name>

        <balling\_avg>29.58</balling\_avg>

        <highest\_score>92</highest\_score>

    </player>

</cricket>

DTD :-

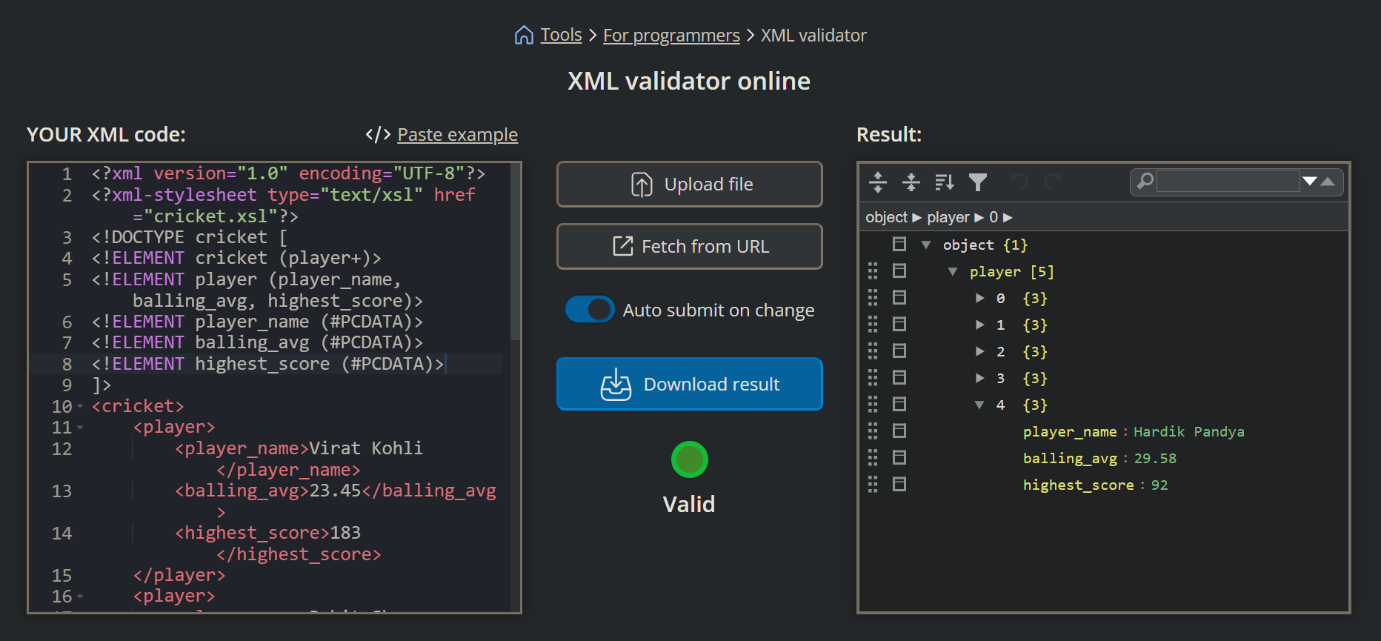
<!ELEMENT cricket (player+)>

<!ELEMENT player (player\_name, balling\_avg, highest\_score)>

<!ELEMENT player\_name (#PCDATA)>

<!ELEMENT balling\_avg (#PCDATA)>

<!ELEMENT highest\_score (#PCDATA)>



XSD :-

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

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

    <xs:element *name*="cricket">

        <xs:complexType>

            <xs:sequence>

                <xs:element *name*="player" *maxOccurs*="unbounded">

                    <xs:complexType>

                        <xs:sequence>

                            <xs:element *name*="player\_name" *type*="xs:string" />

                            <xs:element *name*="balling\_avg" *type*="xs:string" />

                            <xs:element *name*="highest\_score" *type*="xs:string" />

                        </xs:sequence>

                    </xs:complexType>

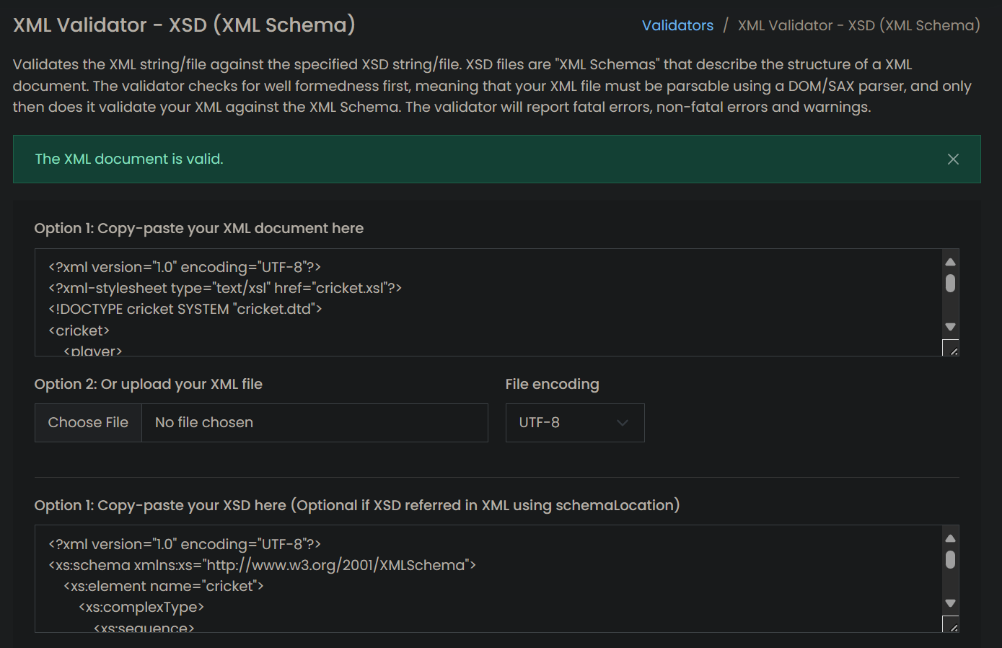
                </xs:element>

            </xs:sequence>

        </xs:complexType>

    </xs:element>

</xs:schema>



XSL :-

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

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

    <xsl:template *match*="/">

        <html>

            <head>

                <title>Cricket Players</title>

                <style>

                    table {

                        width: 50%;

                        border-collapse: collapse;

                        margin: 20px 0;

                    }

                    table, th, td {

                        border: 1px solid black;

                    }

                    th, td {

                        padding: 10px;

                        text-align: center;

                    }

                    th {

                        background-color: #f2f2f2;

                    }

                </style>

            </head>

            <body>

                <h2>Cricket Players Statistics</h2>

                <table>

                    <tr>

                        <th>Player Name</th>

                        <th>Bowling Average</th>

                        <th>Highest Score</th>

                    </tr>

                    <xsl:for-each *select*="cricket/player">

                        <tr>

                            <td><xsl:value-of *select*="player\_name"/></td>

                            <td><xsl:value-of *select*="balling\_avg"/></td>

                            <td><xsl:value-of *select*="highest\_score"/></td>

                        </tr>

                    </xsl:for-each>

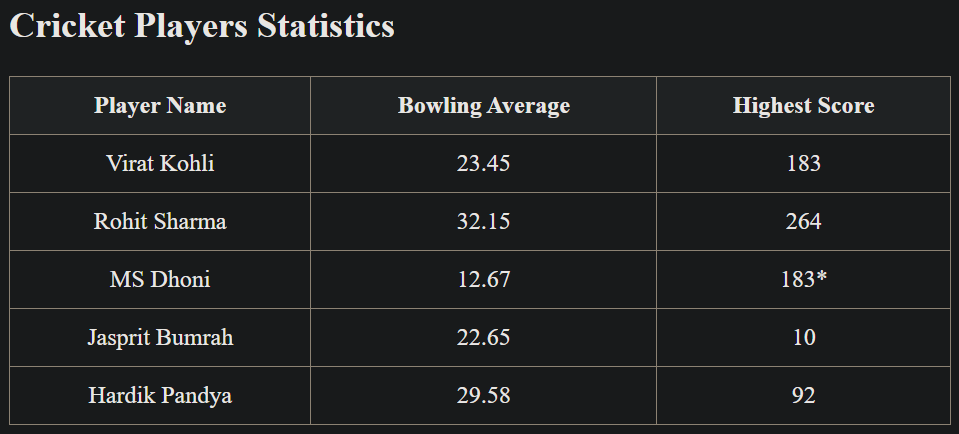
                </table>

            </body>

        </html>

    </xsl:template>

</xsl:stylesheet>



1. Write an XML file to store your semester mark sheet. Write an XSL file using XSLT to display it in XHTML table form.

XML :-

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

<marksheet>

    <student>

        <name>Khushi</name>

        <roll\_number>186147</roll\_number>

        <semester>3</semester>

        <subjects>

            <subject>

                <name>Java</name>

                <marks>88</marks>

                <max\_marks>100</max\_marks>

            </subject>

            <subject>

                <name>DSA</name>

                <marks>92</marks>

                <max\_marks>100</max\_marks>

            </subject>

            <subject>

                <name>Mathematics</name>

                <marks>85</marks>

                <max\_marks>100</max\_marks>

            </subject>

            <subject>

                <name>DBMS</name>

                <marks>89</marks>

                <max\_marks>100</max\_marks>

            </subject>

            <subject>

                <name>Python</name>

                <marks>94</marks>

                <max\_marks>100</max\_marks>

            </subject>

            <subject>

                <name>Dotnet</name>

                <marks>90</marks>

                <max\_marks>100</max\_marks>

            </subject>

        </subjects>

    </student>

</marksheet>

DTD :-

<!ELEMENT marksheet (student+)>

<!ELEMENT student (name, roll\_number, semester, subjects)>

<!ELEMENT name (#PCDATA)>

<!ELEMENT roll\_number (#PCDATA)>

<!ELEMENT semester (#PCDATA)>

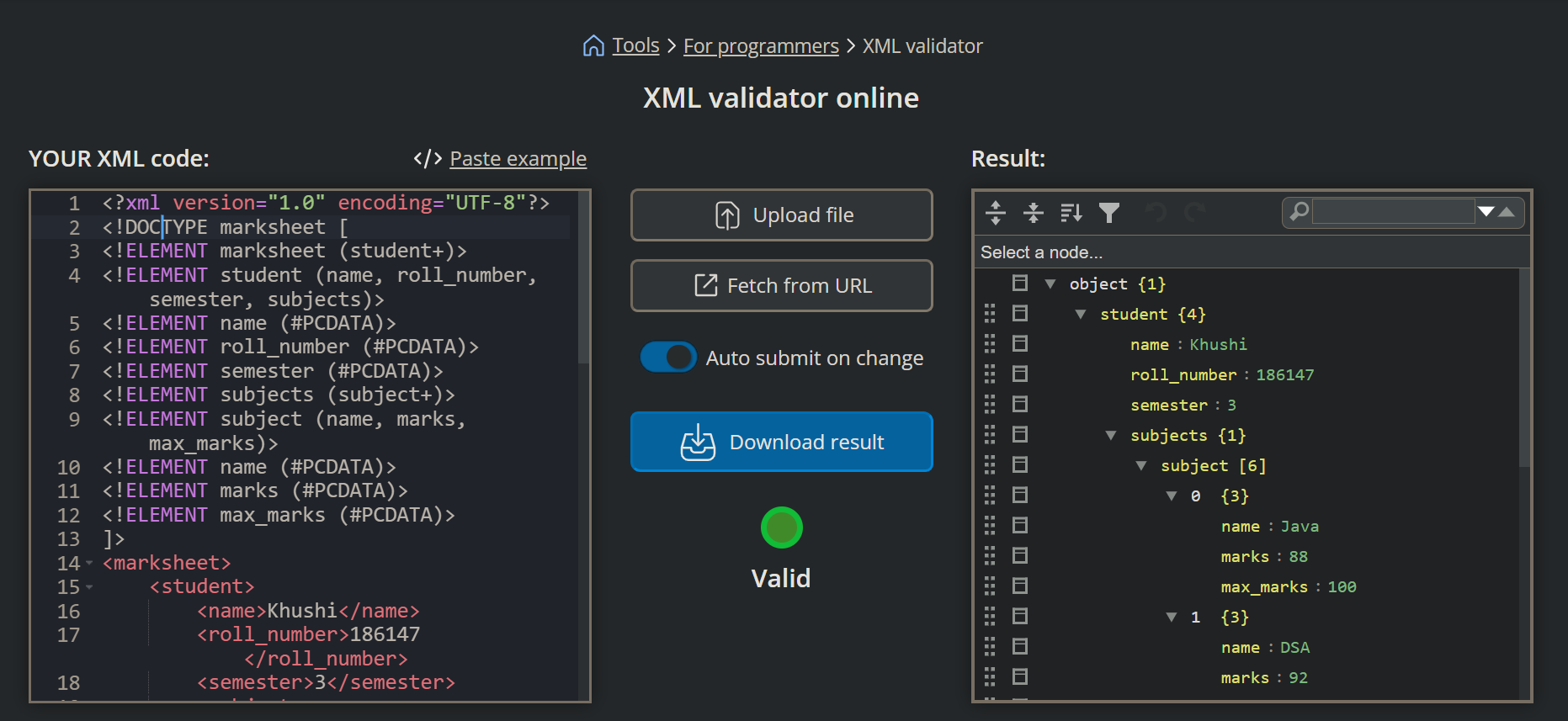
<!ELEMENT subjects (subject+)>

<!ELEMENT subject (name, marks, max\_marks)>

<!ELEMENT name (#PCDATA)>

<!ELEMENT marks (#PCDATA)>

<!ELEMENT max\_marks (#PCDATA)>



XSD :-

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

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

    <xs:element *name*="marksheet">

        <xs:complexType>

            <xs:sequence>

                <xs:element *name*="student">

                    <xs:complexType>

                        <xs:sequence>

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

                            <xs:element *name*="roll\_number" *type*="xs:string"/>

                            <xs:element *name*="semester" *type*="xs:integer"/>

                            <xs:element *name*="subjects">

                                <xs:complexType>

                                    <xs:sequence>

                                        <xs:element *name*="subject" *maxOccurs*="unbounded">

                                            <xs:complexType>

                                                <xs:sequence>

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

                                                    <xs:element *name*="marks" *type*="xs:integer"/>

                                                    <xs:element *name*="max\_marks" *type*="xs:integer"/>

                                                </xs:sequence>

                                            </xs:complexType>

                                        </xs:element>

                                    </xs:sequence>

                                </xs:complexType>

                            </xs:element>

                        </xs:sequence>

                    </xs:complexType>

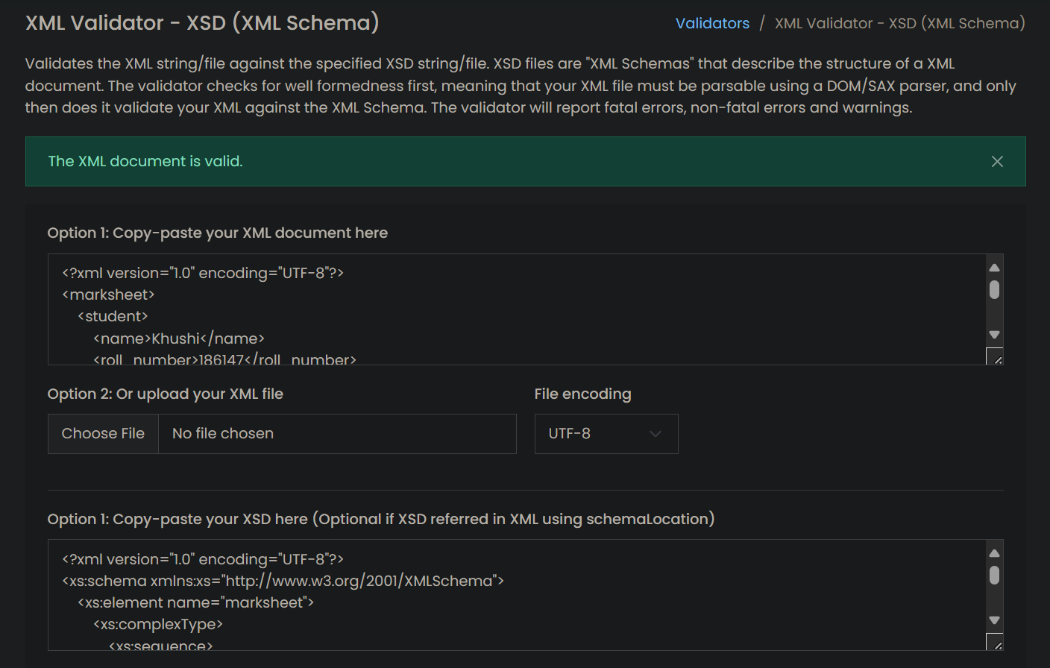
                </xs:element>

            </xs:sequence>

        </xs:complexType>

    </xs:element>

</xs:schema>



XSL :-

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

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

    <xsl:output *method*="html" *doctype-public*="-//W3C//DTD XHTML 1.0 Strict//EN" *doctype-system*="http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd" *indent*="yes"/>

    <xsl:template *match*="/">

        <html *xmlns*="http://www.w3.org/1999/xhtml">

            <head>

                <title>Semester Mark Sheet</title>

                <style>

                    table {

                        width: 60%;

                        border-collapse: collapse;

                    }

                    th, td {

                        border: 1px solid black;

                        padding: 8px;

                        text-align: center;

                    }

                    th {

                        background-color: #f2f2f2;

                    }

                </style>

            </head>

            <body>

                <h1>Semester Mark Sheet</h1>

                <p><strong>Student Name:</strong> <xsl:value-of *select*="marksheet/student/name"/></p>

                <p><strong>Roll Number:</strong> <xsl:value-of *select*="marksheet/student/roll\_number"/></p>

                <p><strong>Semester:</strong> <xsl:value-of *select*="marksheet/student/semester"/></p>

                <table>

                    <tr>

                        <th>Subject</th>

                        <th>Marks Obtained</th>

                        <th>Max Marks</th>

                    </tr>

                    <xsl:for-each *select*="marksheet/student/subjects/subject">

                        <tr>

                            <td><xsl:value-of *select*="name"/></td>

                            <td><xsl:value-of *select*="marks"/></td>

                            <td><xsl:value-of *select*="max\_marks"/></td>

                        </tr>

                    </xsl:for-each>

                </table>

            </body>

        </html>

    </xsl:template>

</xsl:stylesheet>



1. Develop an XML document that will hold a book collection with fields for book name, author, ISBN number and quantity. Write suitable schema for the XML.

XML :-

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

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

<!DOCTYPE library SYSTEM "library.dtd">

<library>

    <book>

        <title>Applied Physics</title>

        <author>Dattu Joshi</author>

        <isnb>12032</isnb>

        <quantity>10</quantity>

    </book>

    <book>

        <title>Computer Organization</title>

        <author>Ramesh Gaonker</author>

        <isnb>12032</isnb>

        <quantity>10</quantity>

    </book>

    <book>

        <title>Data Science for Beginners</title>

        <author>Corman</author>

        <isnb>12032</isnb>

        <quantity>10</quantity>

    </book>

    <book>

        <title>Applied Mathamatics</title>

        <author>Arwin Kreyzik</author>

        <isnb>12032</isnb>

        <quantity>10</quantity>

    </book>

    <book>

        <title>Combinitorial Methods</title>

        <author>Rozen</author>

        <isnb>12032</isnb>

        <quantity>10</quantity>

    </book>

</library>

DTD :-  
<!ELEMENT library (book\*)>

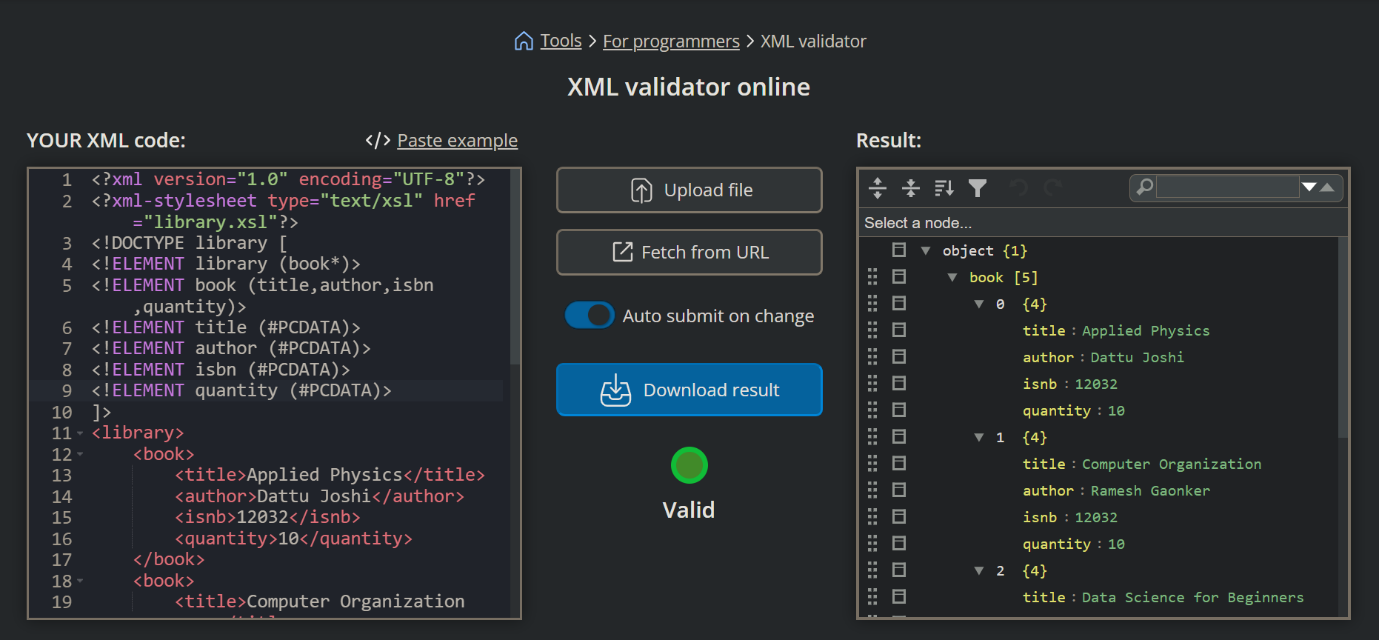
<!ELEMENT book (title,author,isbn,quantity)>

<!ELEMENT title (#PCDATA)>

<!ELEMENT author (#PCDATA)>

<!ELEMENT isbn (#PCDATA)>

<!ELEMENT quantity (#PCDATA)>



XSD :-

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

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

    <xs:element *name*="library">

        <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" *type*="xs:string"/>

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

                            <xs:element *name*="quantity" *type*="xs:integer"/>

                        </xs:sequence>

                    </xs:complexType>

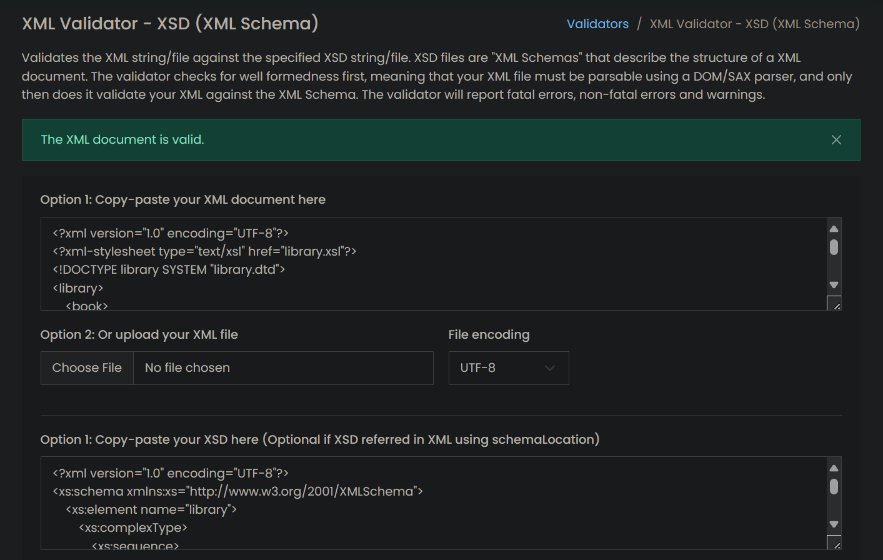
                </xs:element>

            </xs:sequence>

        </xs:complexType>

    </xs:element>

</xs:schema>



XSL :-

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

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

    <xsl:output *method*="html" *doctype-public*="-//W3C//DTD XHTML 1.0 Strict//EN" *doctype-system*="http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd" *indent*="yes"/>

    <xsl:template *match*="/">

        <html *xmlns*="http://www.w3.org/1999/xhtml">

            <head>

                <title>Semester Mark Sheet</title>

                <style>

                    table {

                        width: 60%;

                        border-collapse: collapse;

                    }

                    th, td {

                        border: 1px solid black;

                        padding: 8px;

                        text-align: center;

                    }

                    th {

                        background-color: #f2f2f2;

                    }

                </style>

            </head>

            <body>

                <h1>Semester Mark Sheet</h1>

                <p><strong>Student Name:</strong> <xsl:value-of *select*="marksheet/student/name"/></p>

                <p><strong>Roll Number:</strong> <xsl:value-of *select*="marksheet/student/roll\_number"/></p>

                <p><strong>Semester:</strong> <xsl:value-of *select*="marksheet/student/semester"/></p>

                <table>

                    <tr>

                        <th>Subject</th>

                        <th>Marks Obtained</th>

                        <th>Max Marks</th>

                    </tr>

                    <xsl:for-each *select*="marksheet/student/subjects/subject">

                        <tr>

                            <td><xsl:value-of *select*="name"/></td>

                            <td><xsl:value-of *select*="marks"/></td>

                            <td><xsl:value-of *select*="max\_marks"/></td>

                        </tr>

                    </xsl:for-each>

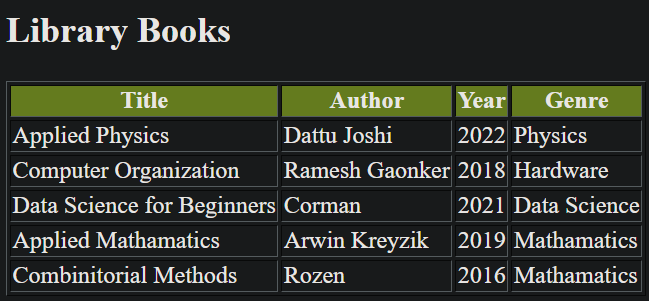
                </table>

            </body>

        </html>

    </xsl:template>

</xsl:stylesheet>



1. Write an XSL file to print list of products which has price< 300Rs. For item.xml file in tabular structure.

XML :-

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

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

<items>

    <item>

        <name>Notebook</name>

        <price>250</price>

    </item>

    <item>

        <name>Pencil</name>

        <price>20</price>

    </item>

    <item>

        <name>Backpack</name>

        <price>500</price>

    </item>

    <item>

        <name>Water Bottle</name>

        <price>150</price>

    </item>

    <item>

        <name>Pen</name>

        <price>100</price>

    </item>

    <item>

        <name>Smartphone Cover</name>

        <price>299</price>

    </item>

</items>

DTD :-

<!ELEMENT items (item+)>

<!ELEMENT item (name, price)>

<!ELEMENT name (#PCDATA)>

<!ELEMENT price (#PCDATA)>

XSD :-

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

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

    <xs:element *name*="items">

        <xs:complexType>

            <xs:sequence>

                <xs:element *name*="item" *maxOccurs*="unbounded">

                    <xs:complexType>

                        <xs:sequence>

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

                            <xs:element *name*="price" *type*="xs:decimal"/>

                        </xs:sequence>

                    </xs:complexType>

                </xs:element>

            </xs:sequence>

        </xs:complexType>

    </xs:element>

</xs:schema>

XSL :-

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

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

    <xsl:template *match*="/items">

        <html>

            <head>

                <title>Products List (Price &lt; 300 Rs)</title>

            </head>

            <body>

                <h2>Products with Price Less than 300 Rs</h2>

                <table *border*="1">

                    <tr>

                        <th>Product Name</th>

                        <th>Price (Rs)</th>

                    </tr>

                    <xsl:for-each *select*="item[price &lt; 300]">

                        <tr>

                            <td><xsl:value-of *select*="name"/></td>

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

                        </tr>

                    </xsl:for-each>

                </table>

            </body>

        </html>

    </xsl:template>

</xsl:stylesheet>