Module Code & Module Title

CS5002NA SOFTWARE ENGINEERING

Assessment Weightage &

Type 30% Individual

Coursework Year and

Semester 2020-2021 Spring

Student Name: Bhuwan Bhetwal

London Met ID: 19030796

College ID: NP01CP4A190340

Assignment Due Date: 07 MAY 2021

Assignment Submission Date: 07 MAY 2021

Title: Music Store Webpage

Count (Where Required): 3093

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

Acknowledgement

The final outcome of this coursework required a bunch of guidance and supervision from lecturers, tutors and colleagues. I am extremely blessed to have such guidance and assistance from them and I cannot forget to thank them.

I would like to express my thanks of gratitude to my module leader **Mr. Dhurba Sen** sir for providing me an opportunity to complete this wonderful coursework and believing me as capable of this which helped me to know about many new things.

Last but not least, I am thankful to Islington college for providing the hygiene study environment which helped me in successfully completing my coursework.

Table of Contents

Table of Figures

Table of Tables

1 Introduction

XML is a mark-up language that can be used to generate our own identifiers also known as Extensible Mark-up language. It was developed in order to circumvent the restrictions of HTML by the World Wide Web Consortium (W3C). Like HTML, XML is a standard generalized markup language based on SGML. Web architecture was considered for XML. XML was developed to store, transport, and self-description files. Data are stored in a plain-text. This offers an isolated storage, transport and exchange of data for software and hardware. XML can be used as an extensible tool for creating a wide variety of text forms. They are not predefined as HTML tags. It makes data exchange easier. It simplifies the transport of data, facilitates shift of channel and makes it easier to access data. XML also facilitates the expansion or update to new operating systems, apps or browsers without data loss. (W3schools, 2021)

XML will not replace HTML soon, but it offers some common features by successfully introducing new HTML capability features. Inside the scenario, XML will function to facilitate HTML document formation in large websites. XML can be used to save and archive data, to adapt the data management needs and to express almost every form of data in XML documents.

XSD (XML Schema Definition) is a recommendation from the World Wide Web Consortium (W3C) which specified how the elements of an extensible mark-up langue (XML) document are formally defined. This definition will be used to ensure that each object in a document complies with the description of the element to be placed in the text. XSD can also serve as a programming object for generation of XML documents. In comparison to the previous XML, XSD has some benefits including DTD and SOX. XSD is written in XML so that intermediate processing by a parser is not needed. (Lawton, 2021)

The XML document is presented as a tree structure when it is parsed on a browser with all the objects, attributes and values. The style of the text using CSS graphically represents all objects, attributes and values. CSS is used to display XML contents simply and correctly. The whole XML document is designed and designed using it.

2 Tree Diagram

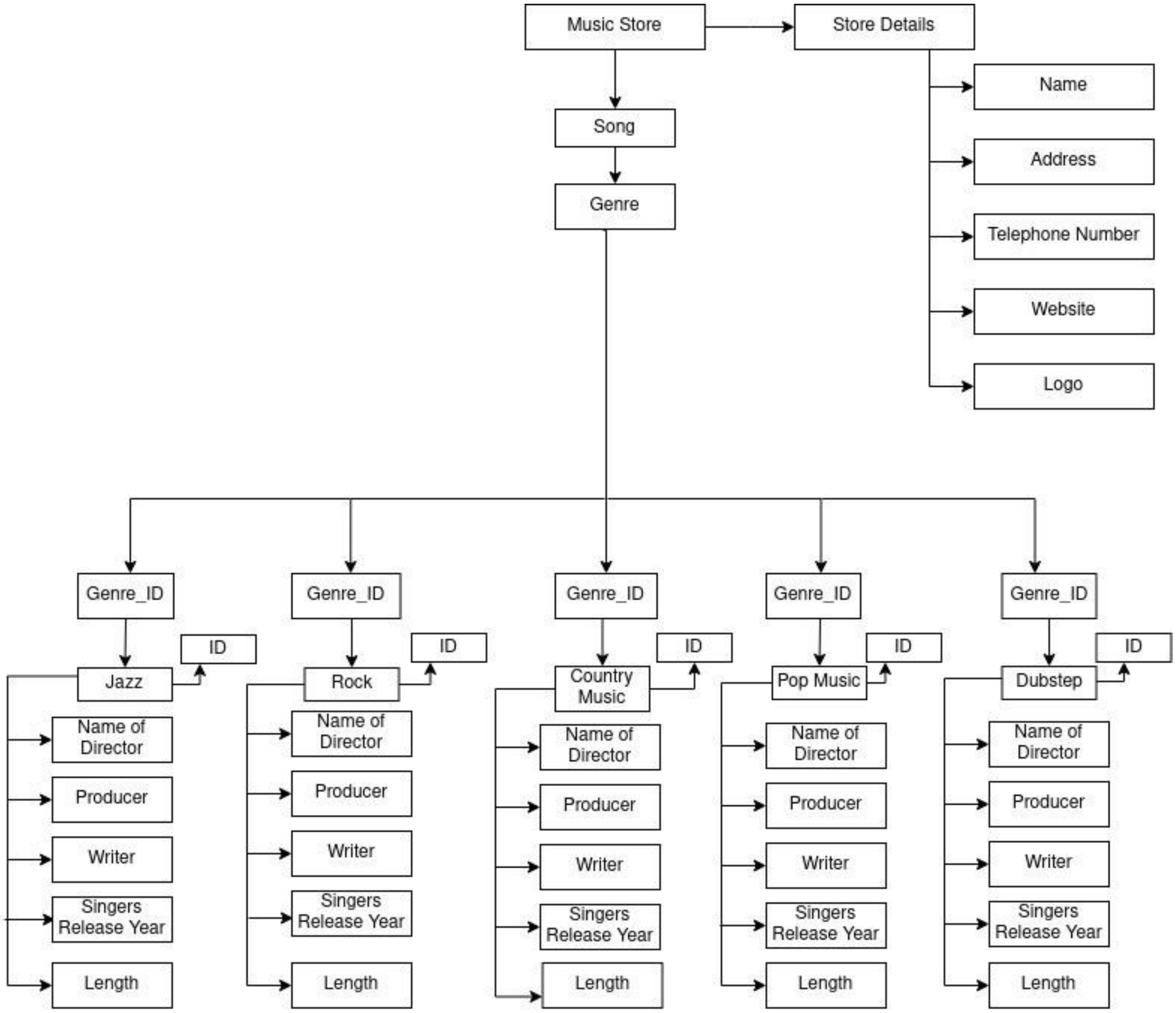


Figure 1 Tree Diagram

3 XML Content

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

<?xml-stylesheet type="text/css" href="catalog_19030796.css" ?>

```
<store xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="catalog_19030796.xsd" >
  <store_detail>
    <name>C-Cords Music</name>
    <address>Dillibazar, Kathmandu</address>
    <telephone>9815955928</telephone>
    <website>ccordsmusic.com.np</website>
    <email>ccordsmusic@gmail.com</email>
    <logo/>
  </store_detail>

  <songs>
    <jazz GenreID="jazz1">
      <song id="J1">
        <name>What a Wonderful World</name>
        <director>Louis Armstrong</director>
        <director_info>
          <age>56</age>
        </director_info>
        <producer>Bob Thiele</producer>
        <writer>George Douglas</writer>
        <released>1967</released>
        <length>2:21</length>
      </song>

      <song id="J2">
        <name>It Don't Mean a Thing (If It Ain't Got That Swing)</name>
        <director>Ellington</director>
        <director_info>
```

```
                <age>50</age>
            </director_info>
            <producer>Duke Ellington</producer>
            <writer>Irving Mills</writer>
            <released>1932</released>
            <length>2:33</length>
        </song>

        <song id="J3">
            <name>Sweet Georgia Brown</name>
            <director>Django Reinhardt</director>
            <director_info>
                <age>49</age>
            </director_info>
            <producer>Maceo Pinkard</producer>
            <writer>Kenneth Casey</writer>
            <released>1925</released>
            <length>4:00</length>
        </song>
    </jazz>

    <rock GenreID="rock1">
        <song id="R1">
            <name>Layla</name>
            <director>Patti Harrison</director>
            <director_info>
                <age>63</age>
            </director_info>
```

```
                <producer>Tom Dowd</producer>
                <writer>Eric Clapton</writer>
                <released>1971</released>
                <length>7:04</length>
            </song>

            <song id="R2">
                <name>Stairway to Heaven</name>
                <director>Led Zeppelin</director>
                <director_info>
                    <age>67</age>
                </director_info>
                <producer>Jimmy Page</producer>
                <writer>Robert Plant</writer>
<released>1971</released>
                <length>8:02</length>
            </song>

            <song id="R3">
                <name>Hey Jude</name>
                <director>Paul McCartney</director>
                <director_info>
                    <age>71</age>
                </director_info>
                <producer>George Martin</producer>
                <writer>Lennon–McCartney</writer>
                <released>1968</released>
                <length>7:11</length>
```

```
        </song>
    </rock>

    <country GenreID="country1">
        <song id="C1">
            <name>10,000 Hours</name>
            <director>Warner Nashville</director>
            <director_info>
                <age>46</age>
            </director_info>
            <producer>Dan Smyers</producer>
            <writer>Shay Mooney</writer>
            <released>2019</released>
            <length>2:47</length>
        </song>

        <song id="C2">
            <name>I Hope</name>
            <director>Gabby Barrett</director>
            <director_info>
                <age>68</age>
            </director_info>
            <producer>Ross Copperman</producer> <writer>Jon Nite</writer>
            <released>2019</released>
            <length>3:23</length>
        </song>

        <song id="C3">
```

```
            <name>The Bones</name>
            <director>Maren Morris </director>
            <director_info>
                <age>56</age>
            </director_info>
            <producer>Greg Kurstin</producer>
            <writer>Laura Veltz</writer>
            <released>2019</released>
            <length>3:17</length>
        </song>
    </country>

    <pop GenreID="pop1">
        <song id="P1">
            <name>Bad Guy</name>
            <director>Dave Meyers</director>
            <director_info>
                <age>69</age>
            </director_info>
            <producer>Finneas O'Connell</producer>
            <writer>Billie Eilish</writer>
            <released>2019</released> <length>3:14</length>
        </song>

        <song id="P2">
            <name>Thriller</name>
            <director>John Landis</director>
            <director_info>
```

```
                <age>73</age>
            </director_info>
            <producer>George Folsey Jr.</producer>
            <writer>John Landis</writer>
            <released>1983</released>
            <length>13:42</length>
        </song>

        <song id="P3">
            <name>Like a Prayer</name>
            <director>Mary Lambert</director>
            <director_info>
                <age>66</age>
            </director_info>
            <producer>Patrick Leonard</producer>
            <writer> Madonna</writer>
            <released>1989</released>
            <length>5:19</length>
        </song>
    </pop>

    <dubstep GenreID="dubstep1">
        <song id="D1">
            <name>Cinema(Skrillex Remix)</name>
            <director>Benny Benassi</director>
            <director_info>
                <age>55</age>
            </director_info>
            <producer>Marco "Benny" Benassi</producer>
```

```xml
        <writer>Gary Baker</writer>
        <released>2011</released>
        <length>3:03 (radio edit)</length>
    </song>

    <song id="D2">
        <name>Promises</name>
        <director>Emil Nava</director>
        <director_info>
            <age>39</age>
        </director_info>
        <producer>Nero</producer>
        <writer>Daniel Stephens</writer>
        <released>2011</released>
        <length>4:17</length>
    </song>

    <song id="D3">
        <name>Scary Monsters and Nice Sprites</name>
        <director>Skrillex</director>
        <director_info>
            <age>75</age>
        </director_info>
        <producer>Bare Noize</producer>
        <writer>Skrillex</writer>
        <released>2010</released>
        <length>44:02</length>
    </song>
```

```
        </dubstep>
    </songs>
</store>
```

4 XSD Content

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

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

    <!-- Simple elements -->
    <xs:element name="name" type="xs:string" />
    <xs:element name="address" type="xs:string" />
    <xs:element name="telephone" type="xs:string" />
    <xs:element name="website" type="xs:string" />
    <xs:element name="email" type="xs:string" />
    <xs:element name="logo" type="xs:anyURI" />
    <xs:element name="director" type="xs:string" />
    <xs:element name="producer" type="xs:string" />
    <xs:element name="writer" type="xs:string" />
    <xs:element name="released" type="xs:integer" />
    <xs:element name="length" type="xs:string" />

    <!-- Simple types -->
    <xs:element name="age">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:enumeration value="56"/>
                <xs:enumeration value="50"/>
                <xs:enumeration value="49"/>
                <xs:enumeration value="63"/>
                <xs:enumeration value="67"/>
                <xs:enumeration value="71"/>
```

```
      <xs:enumeration value="46"/>
      <xs:enumeration value="68"/>
      <xs:enumeration value="56"/>
      <xs:enumeration value="69"/>
      <xs:enumeration value="73"/>
      <xs:enumeration value="66"/>
      <xs:enumeration value="55"/>
      <xs:enumeration value="39"/>
      <xs:enumeration value="75"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

<!-- Attributes -->
<xs:attribute name="id" type="xs:ID" />
<xs:attribute name="GenreID" type="xs:ID" />

<!-- Complex types -->
<xs:element name="director_info">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="age" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="song">
  <xs:complexType>
```

```
      <xs:sequence>
        <xs:element ref="name" />
        <xs:element ref="director" />
        <xs:element ref="director_info" />
        <xs:element ref="producer" />
        <xs:element ref="writer" />
        <xs:element ref="released" />
        <xs:element ref="length" />
      </xs:sequence>
      <xs:attribute ref="id" use="required" />
    </xs:complexType>
  </xs:element>

  <xs:element name="jazz">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="song" minOccurs="0" maxOccurs="unbounded" />
      </xs:sequence>
      <xs:attribute ref="GenreID" use="required" />
    </xs:complexType>
  </xs:element>

  <xs:element name="rock">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="song" minOccurs="0" maxOccurs="unbounded" />
      </xs:sequence>
      <xs:attribute ref="GenreID" use="required" />
    </xs:complexType>
</xs:element>

<xs:element name="country">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="song" minOccurs="0" maxOccurs="unbounded" />
        </xs:sequence>
        <xs:attribute ref="GenreID" use="required" />
    </xs:complexType>
</xs:element>

<xs:element name="pop">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="song" minOccurs="0" maxOccurs="unbounded" />
        </xs:sequence>
        <xs:attribute ref="GenreID" use="required" />
    </xs:complexType>
</xs:element>

<xs:element name="dubstep">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="song" minOccurs="0" maxOccurs="unbounded" />
        </xs:sequence>
        <xs:attribute ref="GenreID" use="required" />
    </xs:complexType>
```

```
</xs:element>

<xs:element name="store_detail">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="name" />
            <xs:element ref="address" />
            <xs:element ref="telephone" />
            <xs:element ref="website" />
            <xs:element ref="email" />
            <xs:element ref="logo" />
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="songs">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="jazz" />
            <xs:element ref="rock" />
            <xs:element ref="country" />
            <xs:element ref="pop" />
            <xs:element ref="dubstep" />
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="store">        <xs:complexType>
```

```
      <xs:sequence>
        <xs:element ref="store_detail" />
        <xs:element ref="songs" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

5 CSS Content

```
store {
        position: relative;    display: block;        margin:
auto;   background-image:
url("./Resources/music1.jpg");      background-repeat:
no-repeat;     background-size: 105% 99%;      opacity:
1;
}

logo {
        background-image:
url("./Resources/logo.png");        display: block;
top: 0px;      position: absolute;    background-size:
220px 121px;           background-position: absolute;
background-repeat: no-repeat;     width: 80%;   height:
100%;          margin-left: 5em;
}

store_detail {
        height: 10%;           width:
100%;          background-color:
#42bfcf; display: block;

        color: whitesmoke;   text-shadow:
black 5px 0px 5px;   border-bottom:
#24c215 4px solid;   top: 20%;     left:
20%;   padding-top: 10px;
   padding-bottom: 10px;
```

```
}

store_detail name, address, telephone, website ,email{
display: inline-block;          font-size: 25;
   font-family: "Lucida Console", "Courier New", monospace;
font-weight: bold;     padding: 5px;          margin-left: 80%;
}

songs {
        height: 80%;
width: 65%;   display:
block;          margin:
auto;
   font-size: 35;
        font-weight: bold;
}
song * {        display: list-
item;   margin-top: 5px;
margin-bottom: 5px;
text-align: left;         font-
size: 15;       color:
whitesmoke;
}

song name {     font-
family: monospace;
font-size: 40px;
display: block;
```

```
   text-decoration-line: overline underline;         text-
decoration-style: solid;       font-weight: bolder;
font-family: Impact, 'Arial Narrow Bold', sans-serif;
}

jazz::before {     font-family:
monospace;  font-size: 50px;        font-
style: bold;     text-shadow: black 5px
0px 5px;        margin-bottom: 5px;
color: #6b86b6; display: block;

         text-align: center;
   content: "Jazz";     background-
position-x: center;
}

rock::before {     font-family:
monospace;  font-size: 50px;        font-
style: bold;     text-shadow: black 5px
0px 5px;        margin-bottom: 5px;
color: #ac2222;       display: block;
text-align: center;     content: "ROCK";
background-position-x: center;


}

country::before {     font-family:
monospace;  font-size: 50px;        font-
style: bold;     text-shadow: black 5px
```

```
0px 5px;       margin-bottom: 5px;
color: #459211;      display: block;
          text-align: center;
   content: "COUNTRY MUSIC";     background-
position-x: center;
}

pop::before {     font-family:
monospace;  font-size: 50px;        font-
style: bold;     text-shadow: black 5px
0px 5px;       margin-bottom: 5px;
color: #b6c52f;        display: block;
text-align: center;     content: "POP
MUSIC";     background-position-x:
center;
}


dubstep::before {     font-family:
monospace;  font-size: 50px;        font-
style: bold;     text-shadow: black 5px
0px 5px;       margin-bottom: 5px;
color: #9c266f;        display: block;
text-align: center;     content:
"DUBSTEP";     background-position-x:
center;

}
```

```
jazz, rock,  country,
pop, dubstep {
position: relative;
   display: block;     background-
position-x: center;
         padding: 20px;        margin-top: 4em;
margin-bottom: 4em;         text-align: center;
background-color: rgba(255, 255, 255, .2);
}

song {           position:
relative;        display:
inline-block;   margin: 1em
3em;   margin-top: 2px;
vertical-align: top;
}

length {
         background-color: black;
} jazz{           background-
image:
url(https://thecaesura.files.wordpress.com/2016/06/dixie_jazz_5.jpg);
box-shadow: 5px 15px 40px black;        border-radius: 20px;
}
rock{
         background-image:url(images/rock.PNG);
box-shadow: 5px 15px 40px black;        border-radius:
20px;
```

```
}
country{
        background-image:                                   url(https://grizzlyrose.com/wp-
content/uploads/2016/05/Birthplace-of-Country-Music.jpg);
box-shadow: 5px 15px 40px black;        border-radius:
20px;
}
pop{    background-image:
url(https://img3.stockfresh.com/files/n/nicemonkey/m/62/323252_stock-photo-
silhouetteconcert-crowd.jpg);
        height: 460px;          box-shadow:
5px 15px 40px black;        border-radius:
20px;
}
dubstep{        background-image:
url(images/dubstep.PNG); box-shadow: 5px 15px
40px black; border-radius: 20px;


 }
store_detail{
 background-image:
url(https://thumbs.gfycat.com/AcclaimedHeartfeltGoatsize_restricted.gif);
}


name::after{
        content: '';     display:
block;          width: 0;
```

```
height: 4px;   background:
#00B1FF;      transition: width
.4s;

  }
  name:hover::after{
width: 100%;
          transition: width .4s;

  }

  address::after{
          content: '';
display: block;
width: 0; height: 4px;
background: #00B1FF;
transition: width .4s;

}
  address:hover::after{
          width: 100%;
          transition: width .4s;

  }

  telephone::after{
          content: '';     display:
block;           width: 0;
```

```
height: 4px;   background:
#00B1FF;      transition: width
.4s;

}
  telephone:hover::after{
            width: 100%;
            transition: width .4s;

  }

  website::after{
            content: '';
            display: block;
            width: 0;
            height: 4px;
            background: #00B1FF;
transition: width .4s;

}
  website:hover::after{
            width: 100%;
            transition: width .4s;

  }

  email::after{
        content: '';     display:
block;            width: 0;
height: 4px;   background:
#00B1FF;      transition: width
.4s;

}
  email:hover::after{
width: 100%;
        transition: width .4s;

  }
```

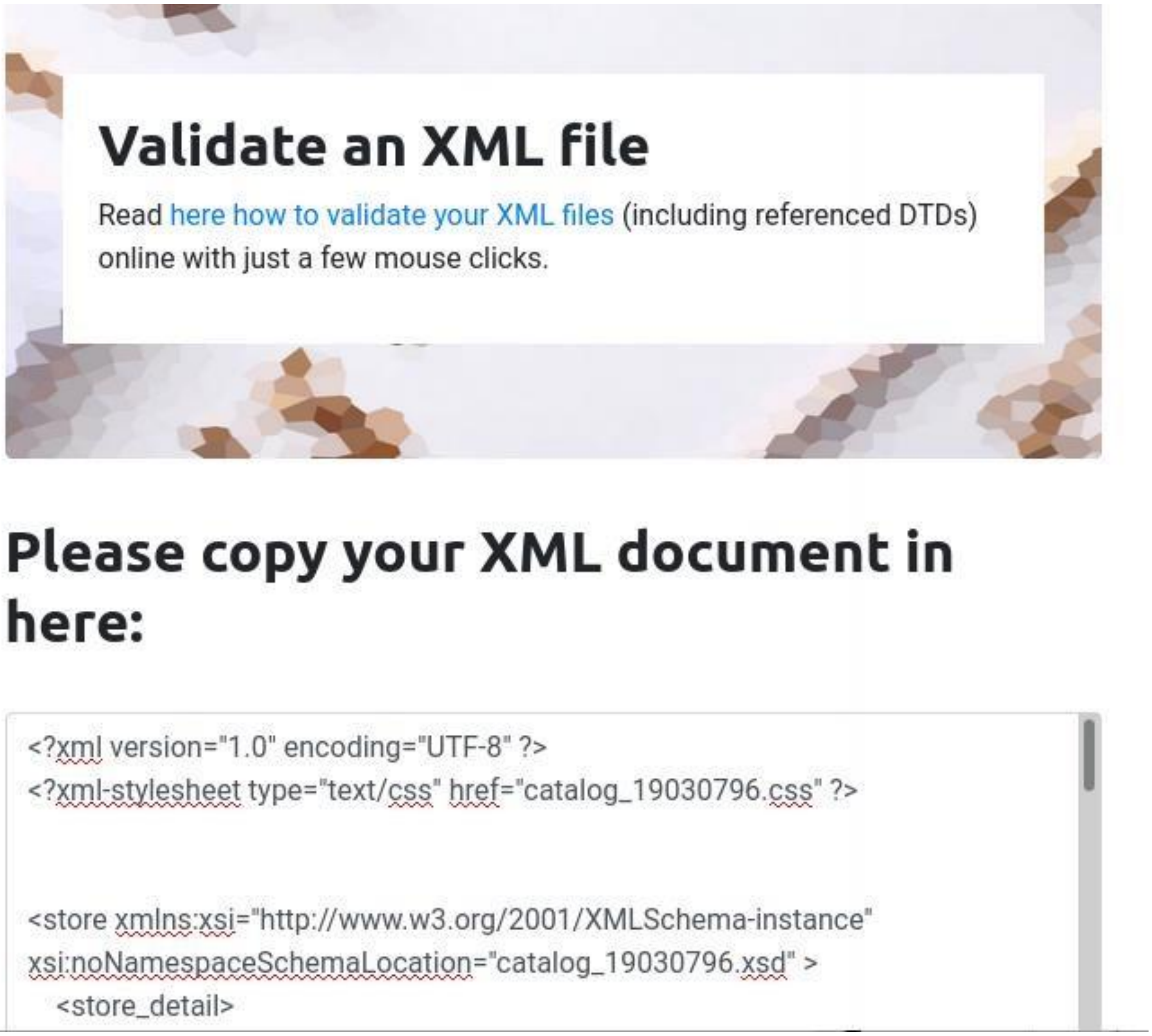
6 DTD Content

```
    <!ELEMENT store (store_detail,songs)>
<!ATTLIST store xmlns:xsi CDATA #REQUIRED>
<!ATTLIST store xsi:noNamespaceSchemaLocation CDATA #REQUIRED>
<!ELEMENT store_detail (Name,address,telephone,website,email,logo)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT address (#PCDATA)>
<!ELEMENT telephone (#PCDATA)>
<!ELEMENT website (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT logo EMPTY>
<!ELEMENT songs (jazz,rock,country,pop,dubstep)>
<!ELEMENT jazz (song)+>
<!ATTLIST jazz GenreID ID #REQUIRED>
<!ELEMENT rock (song)+>
<!ATTLIST rock GenreID ID #REQUIRED>
```

```
<!ELEMENT country (song)+>
<!ATTLIST country GenreID ID #REQUIRED>
<!ELEMENT pop (song)+>
<!ATTLIST pop GenreID ID #REQUIRED>
<!ELEMENT dubstep (song)+>
<!ATTLIST dubstep GenreID ID #REQUIRED>
<!ELEMENT name (#PCDATA)>
<!ELEMENT song (name,director,director_info,producer,writer,released,length)>
<!ATTLIST song id ID #REQUIRED>
<!ELEMENT director (#PCDATA)>
<!ELEMENT director_info (age)>
<!ELEMENT age (#PCDATA)>
<!ELEMENT producer (#PCDATA)>
<!ELEMENT writer (#PCDATA)>
<!ELEMENT released (#PCDATA)>
<!ELEMENT length (#PCDATA)>
```

7 Testing

7.1 Test 1



Validate an XML file

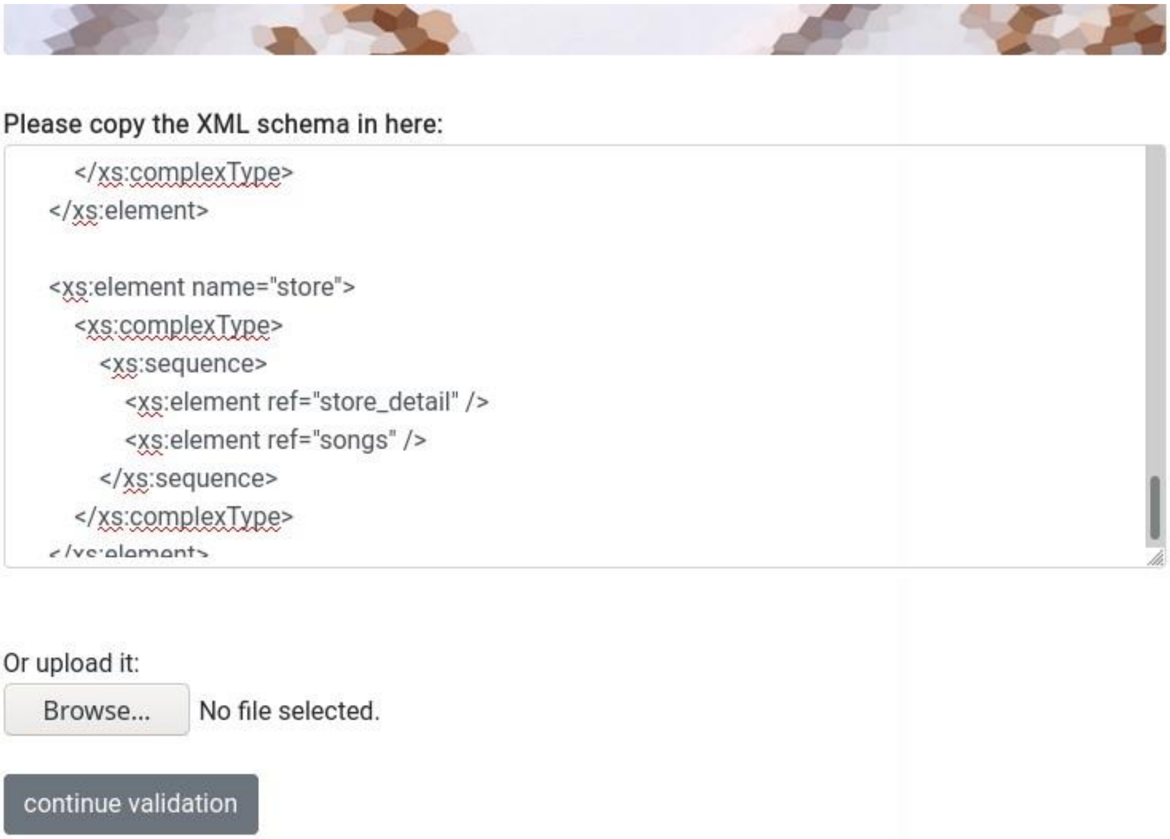
Read here how to validate your XML files (including referenced DTDs) online with just a few mouse clicks.

Please copy your XML document in here:

```
<?xml version="1.0" encoding="UTF-8" ?>
<?xml-stylesheet type="text/css" href="catalog_19030796.css" ?>


<store xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="catalog_19030796.xsd" >
   <store_detail>
```

Figure 2 Uploading XML file for validating.



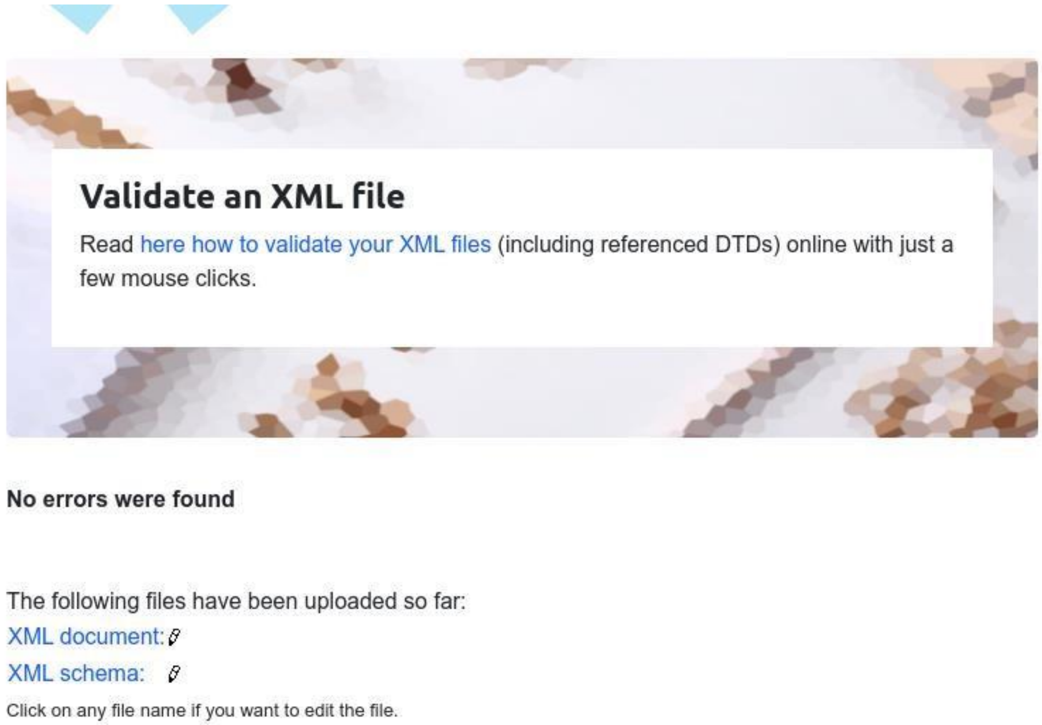
Please copy the XML schema in here:

```
    </xs:complexType>
  </xs:element>

  <xs:element name="store">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="store_detail" />
        <xs:element ref="songs" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```

Or upload it:

Figure 3 Uploading XSD file for validating.



Validate an XML file

Read here how to validate your XML files (including referenced DTDs) online with just a few mouse clicks.

No errors were found

The following files have been uploaded so far:

XML document:

XML schema:

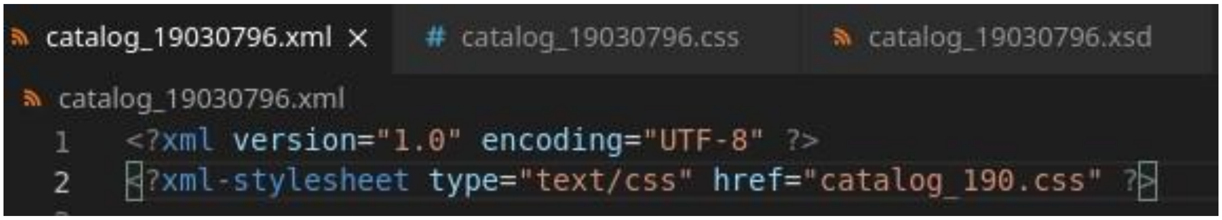
Click on any file name if you want to edit the file.

Figure 4 XML document Validating.

Objective	Validation of XML file against XSD schema document.
Action	Validating the XML file using online XML validator.
Expected outcome	File should be validated without any errors.
Actual Outcome	File was validated with no errors present in the XML file.
Conclusion	The test was successful.

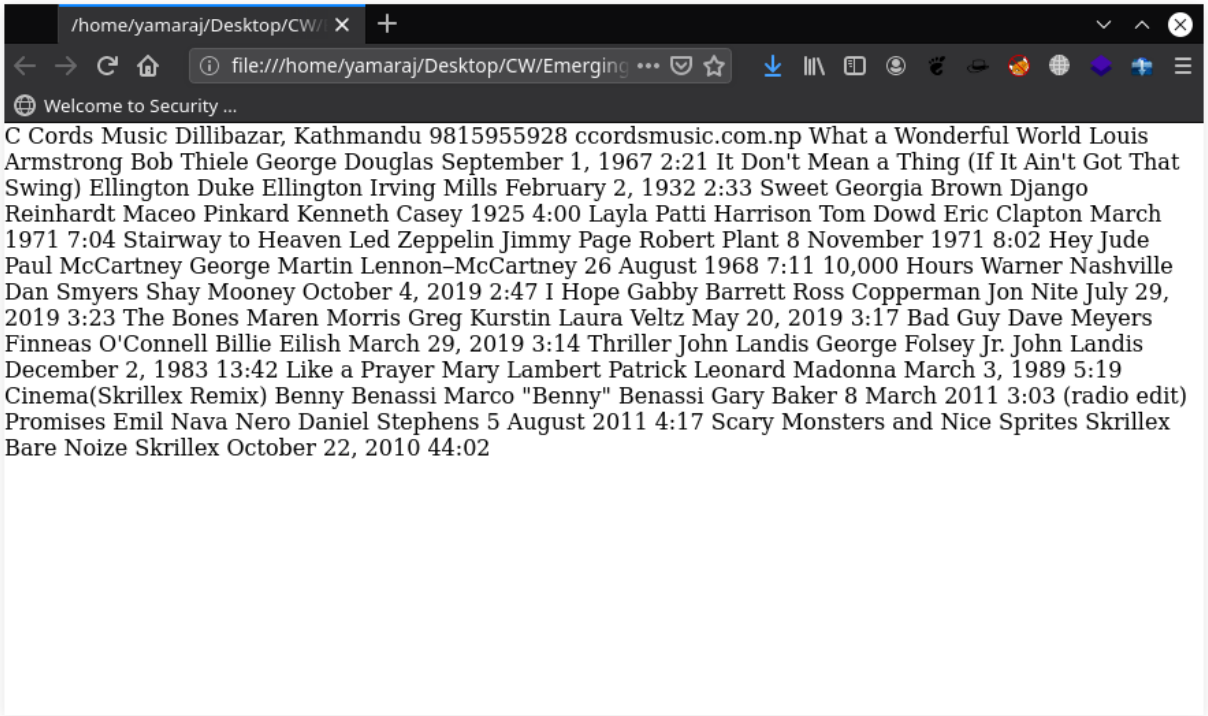
Table 1 Test 1

7.2 Test 2



```
<?xml version="1.0" encoding="UTF-8" ?>
<?xml-stylesheet type="text/css" href="catalog_190.css" ?>
```

Figure 5 Linking invalid CSS file


/home/yamaraj/Desktop/CW/
file:///home/yamaraj/Desktop/CW/Emerging
Welcome to Security ...

C Cords Music Dillibazar, Kathmandu 9815955928 ccordsmusic.com.np What a Wonderful World Louis Armstrong Bob Thiele George Douglas September 1, 1967 2:21 It Don't Mean a Thing (If It Ain't Got That Swing) Ellington Duke Ellington Irving Mills February 2, 1932 2:33 Sweet Georgia Brown Django Reinhardt Maceo Pinkard Kenneth Casey 1925 4:00 Layla Patti Harrison Tom Dowd Eric Clapton March 1971 7:04 Stairway to Heaven Led Zeppelin Jimmy Page Robert Plant 8 November 1971 8:02 Hey Jude Paul McCartney George Martin Lennon–McCartney 26 August 1968 7:11 10,000 Hours Warner Nashville Dan Smyers Shay Mooney October 4, 2019 2:47 I Hope Gabby Barrett Ross Copperman Jon Nite July 29, 2019 3:23 The Bones Maren Morris Greg Kurstin Laura Veltz May 20, 2019 3:17 Bad Guy Dave Meyers Finneas O'Connell Billie Eilish March 29, 2019 3:14 Thriller John Landis George Folsey Jr. John Landis December 2, 1983 13:42 Like a Prayer Mary Lambert Patrick Leonard Madonna March 3, 1989 5:19 Cinema(Skrillex Remix) Benny Benassi Marco "Benny" Benassi Gary Baker 8 March 2011 3:03 (radio edit) Promises Emil Nava Nero Daniel Stephens 5 August 2011 4:17 Scary Monsters and Nice Sprites Skrillex Bare Noize Skrillex October 22, 2010 44:02

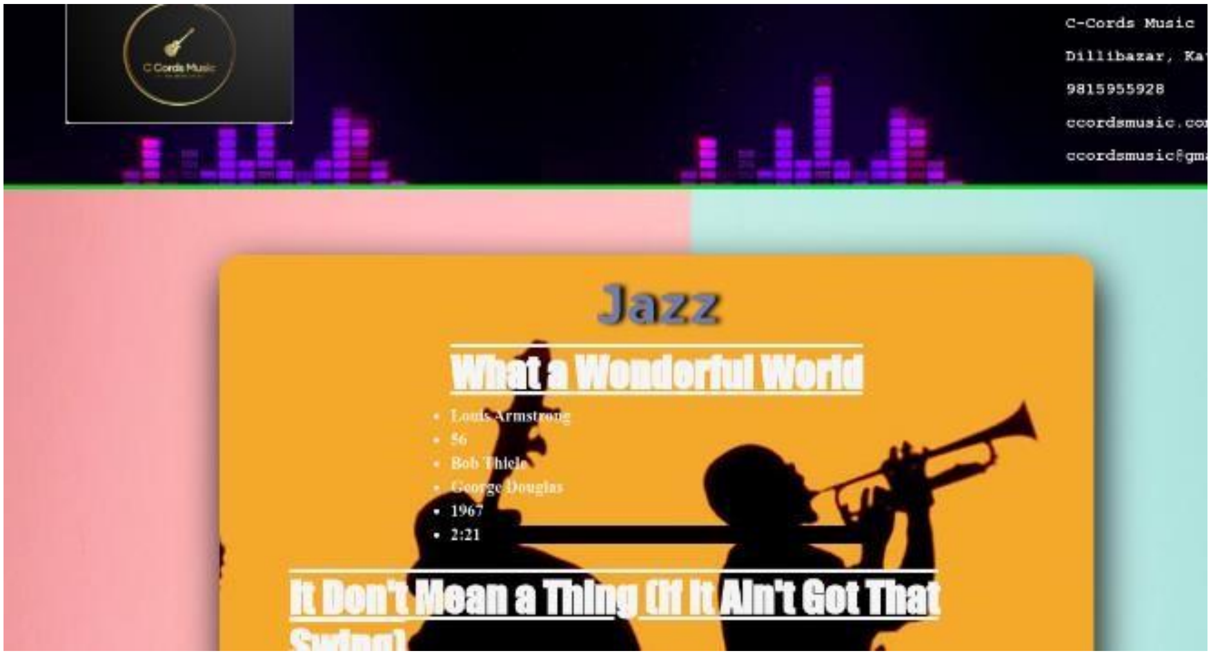
Figure 6 Displayed Error Message

Objective	Testing the CSS file reference format if it works or not.
Action	Linking Invalid CSS file in XML file and opened in the browser.

Expected outcome	XML file should be shown without proper format.
Actual Outcome	XML file was shown without proper format.
Conclusion	The test was successful.

Table 2 Test 2

7.3 Test 3



C-Cords Music

Dillibazar, Ka

9815955928

ccordsmusic.co

ccordsmusic@gm

Jazz

What a Wonderful World

- Louis Armstrong
- 56
- Bob Thiele
- George Douglas
- 1967
- 2:21

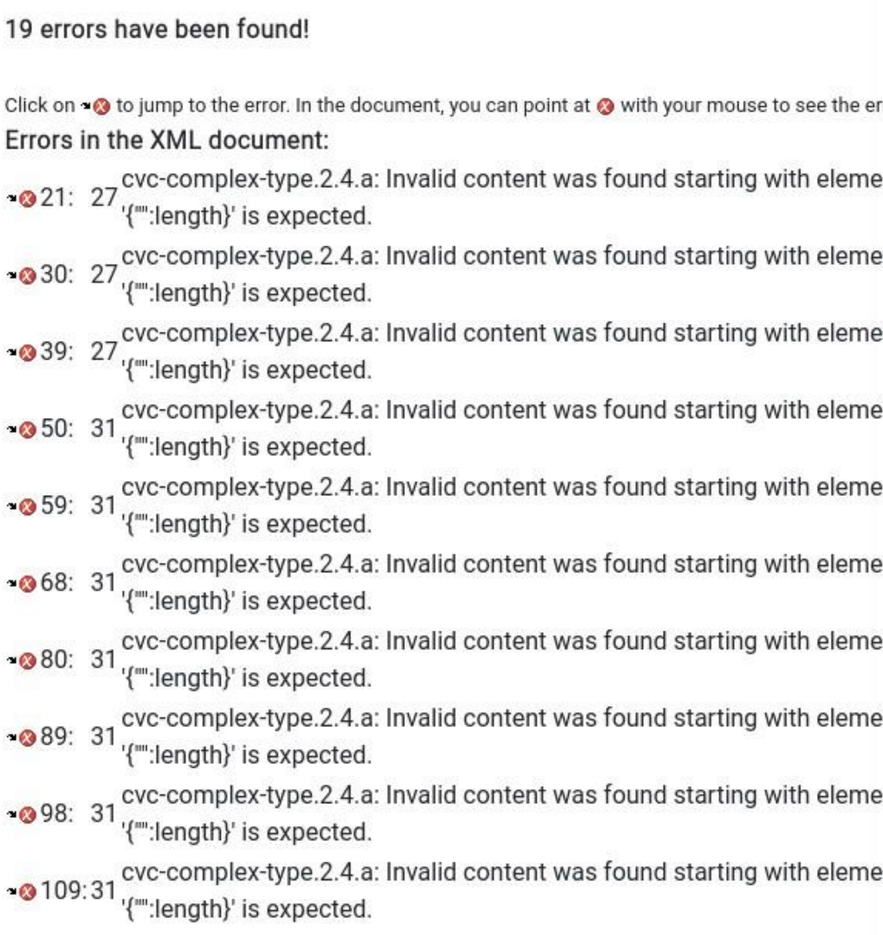
It Don't Mean a Thing (If It Ain't Got That Swing)

Figure 7: Loading XML in the browser.

Objective	Testing whether XML file can be displayed in the browser.
Action	The XML file was loaded in the firefox browser.
Expected outcome	The XML file should be rendered and displayed.
Actual Outcome	The XML file was rendered and displayed.
Conclusion	The test was successful.

Table 3 Test 3

7.4 Test 4



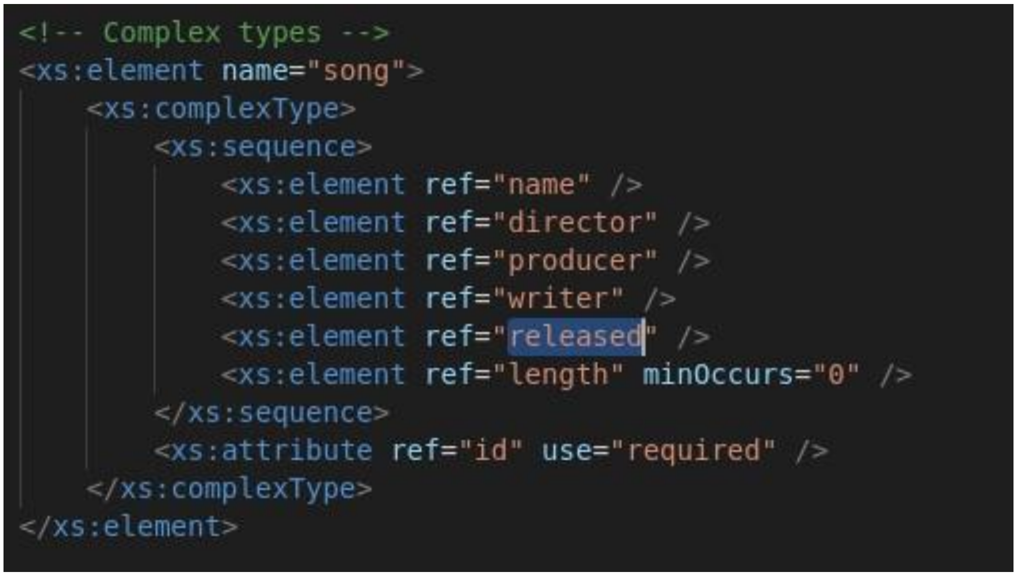
19 errors have been found!

Click on ↘⊗ to jump to the error. In the document, you can point at ⊗ with your mouse to see the error message.

Errors in the XML document:

```
↘⊗21:  27 cvc-complex-type.2.4.a: Invalid content was found starting with element 'released'. One of
          '{"":length}' is expected.
↘⊗30:  27 cvc-complex-type.2.4.a: Invalid content was found starting with element 'released'. One of
          '{"":length}' is expected.
↘⊗39:  27 cvc-complex-type.2.4.a: Invalid content was found starting with element 'released'. One of
          '{"":length}' is expected.
↘⊗50:  31 cvc-complex-type.2.4.a: Invalid content was found starting with element 'released'. One of
          '{"":length}' is expected.
↘⊗59:  31 cvc-complex-type.2.4.a: Invalid content was found starting with element 'released'. One of
          '{"":length}' is expected.
↘⊗68:  31 cvc-complex-type.2.4.a: Invalid content was found starting with element 'released'. One of
          '{"":length}' is expected.
↘⊗80:  31 cvc-complex-type.2.4.a: Invalid content was found starting with element 'released'. One of
          '{"":length}' is expected.
↘⊗89:  31 cvc-complex-type.2.4.a: Invalid content was found starting with element 'released'. One of
          '{"":length}' is expected.
↘⊗98:  31 cvc-complex-type.2.4.a: Invalid content was found starting with element 'released'. One of
          '{"":length}' is expected.
↘⊗109:31 cvc-complex-type.2.4.a: Invalid content was found starting with element 'released'. One of
          '{"":length}' is expected.
```

Figure 8 Error message displayed



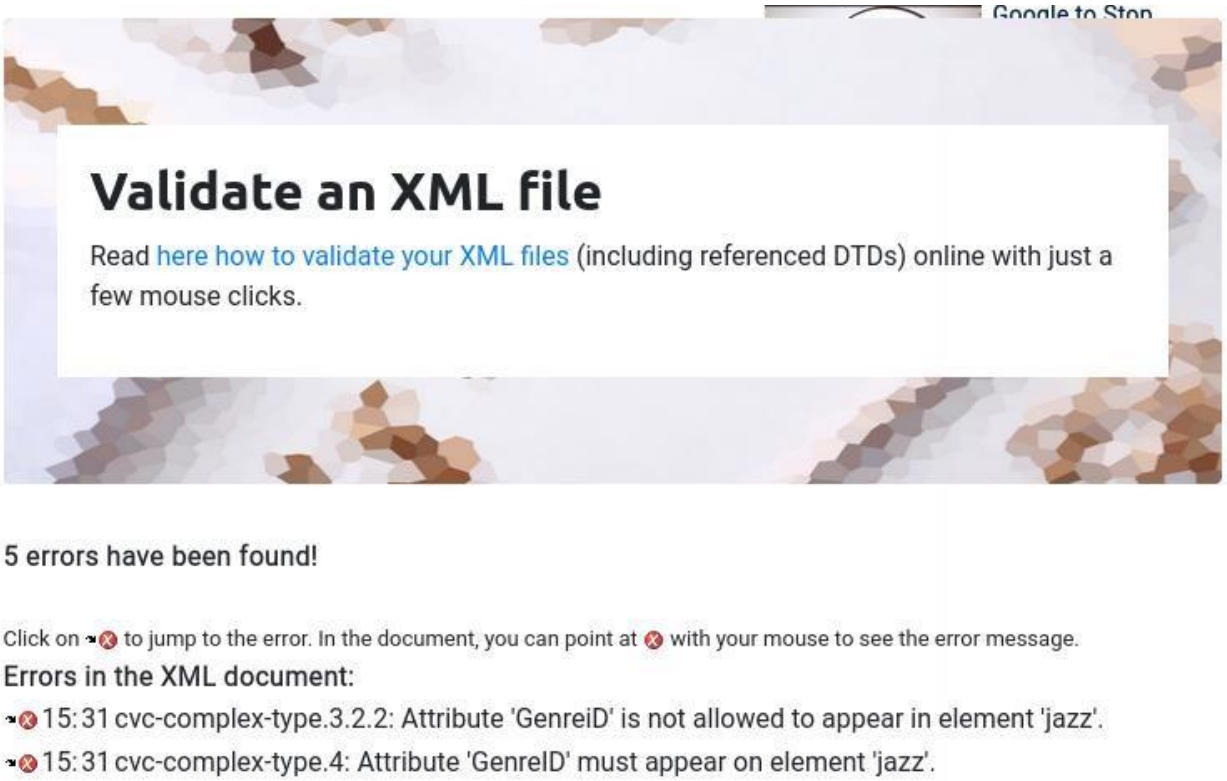
```
<!-- Complex types -->
<xs:element name="song">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="name" />
            <xs:element ref="director" />
            <xs:element ref="producer" />
            <xs:element ref="writer" />
            <xs:element ref="released" />
            <xs:element ref="length" minOccurs="0" />
        </xs:sequence>
        <xs:attribute ref="id" use="required" />
    </xs:complexType>
</xs:element>
```

Figure 9 passing invalid element in XML file

Objective	Testing if XML file can be validated or not when the content (released) is not specified in XSD file.
Action	Using invalid child element instead of valid element as stated in XSD.
Expected outcome	There should be an error indicating invalid element
Actual Outcome	Error message was displayed indicating invalid element
Conclusion	The test was successful.

Table 4 Test 4

7.5 Test 5



Google to Stop

Validate an XML file

Read here how to validate your XML files (including referenced DTDs) online with just a few mouse clicks.

5 errors have been found!

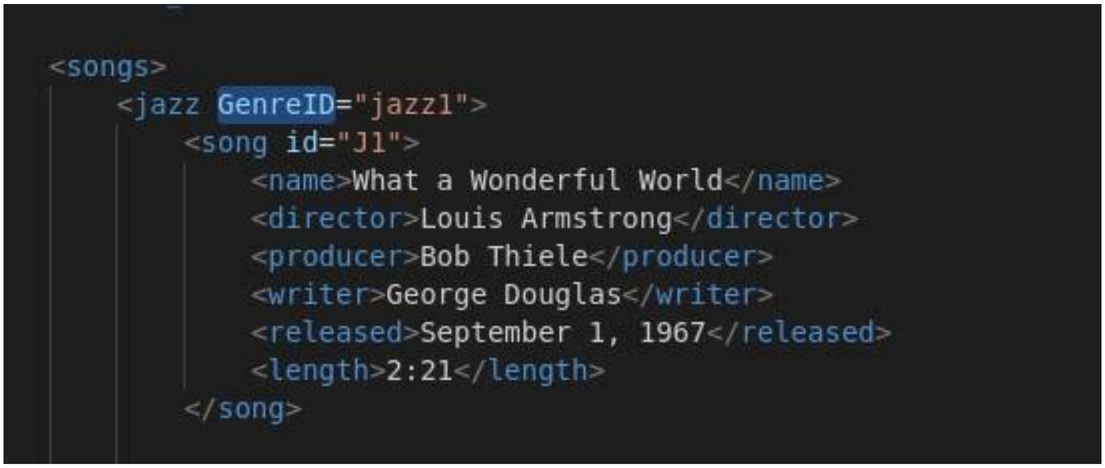
Click on ↘⊗ to jump to the error. In the document, you can point at ⊗ with your mouse to see the error message.

Errors in the XML document:

↘⊗ 15: 31 cvc-complex-type.3.2.2: Attribute 'GenreiD' is not allowed to appear in element 'jazz'.

↘⊗ 15: 31 cvc-complex-type.4: Attribute 'GenreID' must appear on element 'jazz'.

Figure 10 Error message displayed



```
<songs>
    <jazz GenreID="jazz1">
        <song id="J1">
            <name>What a Wonderful World</name>
            <director>Louis Armstrong</director>
            <producer>Bob Thiele</producer>
            <writer>George Douglas</writer>
            <released>September 1, 1967</released>
            <length>2:21</length>
        </song>
```

Figure 11 Inserting invalid attribute in XSD file.

Objective	Testing schema specifications.
Action	Inserting an invalid attribute in the XML document which is not specified in the XSD file

Expected outcome	Validator should show that the XML contains an unknown attribute.
Actual Outcome	Validator show that there was an unknown attribute present in XML file.
Conclusion	The test was successful.

Table 5 Test 5

8 Difference between Schema and DTD

Here is the difference between DTD and Schema.

DTD	Schema
DTD is the Document Type Definition document which is a document that describes an XML document structure.	XSD is the definition of the XML Schema and a way to explain the XML format structure.
It is used to accurately classify the XML attributes	The rules for all attributes and elements in an XML document are defined by schema.
Two forms may be classified: internal DTD and external DTD. It may be entered in or out of a document.	The XML documents can also be developed. It also monitors the document's language. There is no need for an analyser to process
DTD primarily verifies XML text syntax and validity. It verifies whether a correct structure is present in an XML document.	XSD verifies if the XML file structure is valid.
Namespace is not supported.	Namespace is supported.
Datatypes are not supported.	Datatypes are supported.
DTD uses SGML syntax.	For writing XSD, XML is used.
It is not extensible in nature.	It is extensible in nature.
It gives us no influence over the XML document structure.	It allows one to have more influence over the XML document structure.
XML schemes characterize order of child element and number.	XSD does not characterize order of child element and number.

Table 6: Difference between DTD and Schema.

(Pearson, 2021)

(GeeksforGeeks, 2021)

9 Coursework Development

At first vast research was done in XML and CSS. Then the development process was proceeded in Visual Studio Code which is an editor for the development process. The design was created and analysed from Tree diagram. Tree diagram was created in draw.io. After the diagram was created, the XML code was written which was later linked with schema and CSS. After the completion of the XML the XSD code was written which managed the elements and attributes of the XSD. Then for the design CSS was used and implemented in XML file. The code was validated using an online validation tool. The next step was, performing the testing. The code was executed which resulted various errors, which was solved manually. The code was executed successfully with no errors. After the completion of the development process. The Report was initiated.

9.1 Tools Used

9.1.1 Visual Studio Code

Visual Studio Code, a simplified programming editor with development support, such as debugging, working flow and version control. It aims to provide the tools; a developer needs for a fast code-build-debug cycle. The entire coding aspect of this course was achieved with visual code studio support, whether it is XML, XSD or CSS. The VS Code function was useful because it had a super interactive debugging feature so we could go through source code, examine the variable, and run commands on the screen. VS Code already has excellent web technology tools, so creating the CSS file was very useful. (Visual Studio Code)

9.1.2 Draw.io

Draw.io is a free software as well as web application in order to create charts and diagrams. Draw.io provides and elegant drag and drop GUI, prototypes for custom diagrams and library types. Draw.io has options for storing diagrams in the cloud, on a computer or in a data center, based on the requirements. Users can generate a variety of diagrams and edit them, including flowcharts, ER diagrams, UMLs, etc. This program has been used to draw the XML text Tree diagram. Working on draw.io was simple and its different features made it interactive for this device users. (Computer Hope, 2021)

10 Critical Analysis

This has been our first task on XML. With a previous year's HTML history, I understood XML more clearly. Though I knew HTML, my XML work was entirely different, it did not help me completely. The XML language was not similar to HTML which created a lot of confusions initially. During the completion of the assignment, many issues and complications arose and various action were taken to overcome the problems and obstacles which are as follows:

- At first XSD file was developed, it was the initial challenge of the work. It was done in order validate the XML file. The elements and attributes in XSD were managed according to the XML document. The misplaced of the elements caused various errors which were manually solved later. This helped in overcoming the first challenge.
- The very next challenge was working on with CSS. The website looks and designed need to be attractive. All the text and contents need to be displayed on the webpage in various forms, which was a major challenge. Each factor was designed in various styles which was a difficult task. After various try the CSS file was successfully created in no time. The website designed was done attractively.

11 Conclusion

The development work was successfully completed and a website was developed for the sound strong music store. A brief report was written to clarify the design and execution of the course. The main aim of the coursework was to create an online system for a music store named sound strong. Before beginning the development process, a tree diagram was drawn for the definition of the structure of data for the XML Document. The XML file was then developed following all the requirements as given in the question. The document was filled by some of my extra information too. A schema XSD file was also created as per the structure of the XML document which was later connected to the XML document. Afterwards, these documents were validated by using an online XML validator. The validation step was time consuming as many errors were discovered while validating the files. For the design and display of the webpage CSS file was created which helped to render the XML data in beautiful manner.

Various research and learning was done in order to create the document which met all the criteria of the question. This was the primary part of the documentation. Numerous websites were visited, journals and books also assisted a lot. Lecture slides as well as other study resources gave knowledge regarding XML, XSD, DTD and CSS.

They were really helpful as they helped me a lot in coding portion. Frequent interaction with my module leader i was able to successfully complete the given work on time. The work contained a lot of obstacles and had many challenges. The work would be incomplete without the guidance of my tutor as well as my seniors. The advice and resources provided by the tutor helped me do my work without any confusion.

In Summary, I learned about XML, XSD, DTD, and CSS in this course. The coursework was a great help on sharpening my skills on XML creation. While this task was very challenging and a lot of work was required, since it was my first task with XML, I managed to complete it successfully without any errors. I also discovered various new functions of the Microsoft word which made me do my reports in easier manner. In my potential career as a successful programmer, the skills I have gathered from this course work will certainly be included.

12 References

Computer Hope, 2021. *What is Draw.io?.* [Online] Available at: https://www.computerhope.com/jargon/d/drawio.htm [Accessed 06 05 2021].

GeeksforGeeks, 2021. *Difference between Document Type Definition (DTD) and XML Schema Definition (XSD) - GeeksforGeeks.* [Online] Available at: https://www.geeksforgeeks.org/difference-between-document-typedefinition-dtd-and-xml-schema-definition-xsd/ [Accessed 06 05 2021].

Lawton, G., 2021. *What is XSD (XML Schema Definition)? - Definition from WhatIs.com.* [Online] Available at: https://whatis.techtarget.com/definition/XSD-XML-SchemaDefinition#:~:text=XSD%20(XML%20Schema%20Definition)%20is,Markup%20Languag e%20(XML)%20document [Accessed 06 05 2021].

Pearson, 2021. *Differences Between DTDs and Schema | Converting a DTD into a Schema | InformIT.* [Online]

Available at: https://www.informit.com/articles/article.aspx?p=24614&seqNum=3#:~:text=The%20critical%20difference%20between%20DTDs,syntax%20itself%20is%20quite%20terse [Accessed 06 05 2021].

Visual Studio Code, 2021. *Why Visual Studio Code?.* [Online] Available at: https://code.visualstudio.com/docs/editor/whyvscode [Accessed 06 05 2021].

W3schools, 2021. *XML Tutorial.* [Online] Available at: https://www.w3schools.com/xml/ [Accessed 06 05 2021].

13 Bibliography

Cole, Timothy W, et al. Coding with XML for Efficiencies in Cataloging : Practical Applications of XSD, XSLT, and XQuery. London, Facet Publishing, 2018.

Liu, Mengchi. "DTD Schema: A Simple but Powerful XML Schema Language." International Journal of Web Information Systems, vol. 4, no. 4, 21 Nov. 2008, pp. 465–483,

10.1108/17440080810919495. Accessed 2 May 2019.

Williamson, Heather. Xml. Berkeley, Calif. ; London, Osborne, 2001.