



Islington college
(इस्लिंग्टन कलेज)

Module Code & Module Title

CS5004NI Emerging Programming Platforms and Technology

Assessment Weightage & Type

30% Individual Coursework

Year and Semester

2020-21 Autumn

Student Name: Rusan Maharjan

London Met ID: 19031542

College ID: NP01CP4A190036

Assignment Due Date: 7th May, 2021

Assignment Submission Date: 7th May, 2021

Word Count (Where Required): 5044

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.

Table of Contents

Introduction	1
XML Content.....	6
Tree Diagram	6
XML Content.....	7
Schema Content.....	22
DTD Content	26
Testing	29
Test 1:.....	29
Test 2:.....	31
Test 3:.....	33
Test 4:.....	36
Test 5:.....	39
Test 6:.....	42
How you developed the coursework?	43
Difference between Schema and DTD	45
Critical Evaluation.....	46
Conclusion.....	47
References.....	48

Table of Figures

Figure 1: XML	1
Figure 2: Visual Studio Code	2
Figure 3: CSS (Cascading Style Sheet).....	3
Figure 4: DTD (Document Type Declaration).....	4
Figure 5: Schema	5
Figure 6: Tree Diagram	6
Figure 7: Inserting xml file in validating site	29
Figure 8: No errors found in xml file	30
Figure 9: Inserting DTD file in validating site	31
Figure 10: Error occurs during validating DTD	32
Figure 11: Error occurs due to the code inside red border code.....	32
Figure 12: inserting DTD file in validating site	33
Figure 13: fixed error in DTD.....	34
Figure 14: No errors found validating DTD	35
Figure 15: Inserting schema file in validating site	36
Figure 16: Error occurs during validating schema file	37
Figure 17: Error occurs from code inside red border: wrong tag.....	38
Figure 18: Inserting schema file in validating site	39
Figure 19: Error fixed of schema file	40
Figure 20: No errors found validating schema file.....	41
Figure 21: XML after applying CSS	42
Figure 22: Draw.io	43
Figure 23: Visual Studio Code	44

Table of Tables

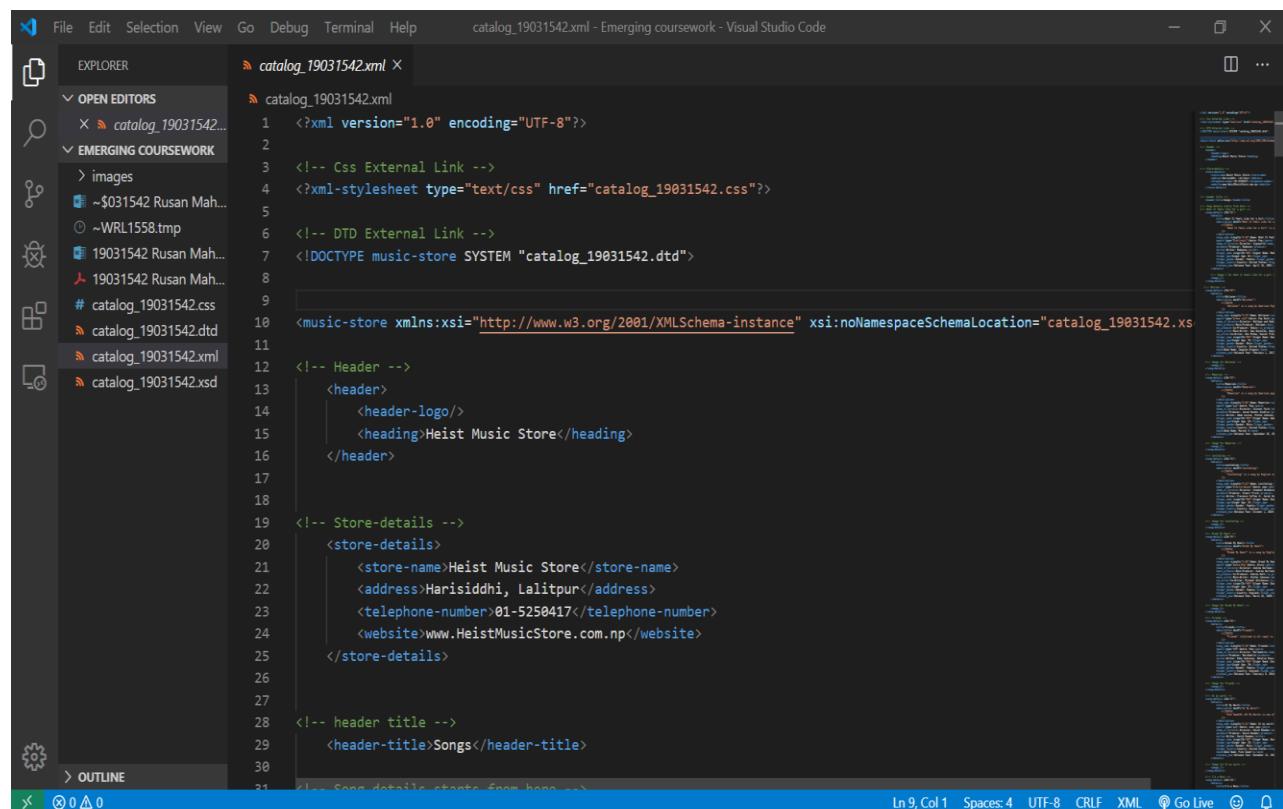
Table 1: Test 1.....	29
Table 2: Test 2.....	31
Table 3: Test 3.....	33
Table 4: Test 4.....	36
Table 5: Test 5.....	39
Table 6: Test 6.....	42

Introduction

This is second coursework of Emerging Programming Platforms and Technologies. We are assigned to model a system for music store. To build a system for music store we have used XML (Extensible Markup Language), CSS (Cascading Style Sheet), DTD (Document Type Definition), XSD (XML Schema Definition). XML is a text-based markup language derived from Standard Generalized Markup Language. XML tag identify the data and are used to store and organize the data, rather than specifying how to display it like HTML tags, which are used to display the data (tutorialspoint, 2021). By using XML, Web agents and robots are more efficient and produce more useful results. To complete this coursework XML, DTD, Schema, CSS are used and draw.io and Visual Studio Code is used.

XML:

XML stands for extensible markup language. A markup language is a set of codes, or tags, that describes the text in a digital document. XML is a more flexible than HTML and makes it possible to conduct complex business over the internet (Roche, 2000).



```

<?xml version="1.0" encoding="UTF-8"?>
<!-- Css External Link -->
<?xml-stylesheet type="text/css" href="catalog_19031542.css"?>
<!-- DTD External Link -->
<!DOCTYPE music-store SYSTEM "catalog_19031542.dtd">
<music-store xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="catalog_19031542.xsd">
    <!-- Header -->
    <header>
        <header-logo/>
        <heading>Heist Music Store</heading>
    </header>
    <!-- Store-details -->
    <store-details>
        <store-name>Heist Music Store</store-name>
        <address>Harisiddhi, Lalitpur</address>
        <telephone-number>01-5250417</telephone-number>
        <website>www.HeistMusicStore.com.np</website>
    </store-details>
    <!-- header title -->
    <header-title>Songs</header-title>
</music-store>

```

Figure 1: XML

Visual Studio Code:

This coursework is done in Visual Studio Code which is a text editor. Visual Studio Code known as VS Code is a free open-source text editor by Microsoft. VS code is available for Windows, Linux and macOS (educative, 2021).

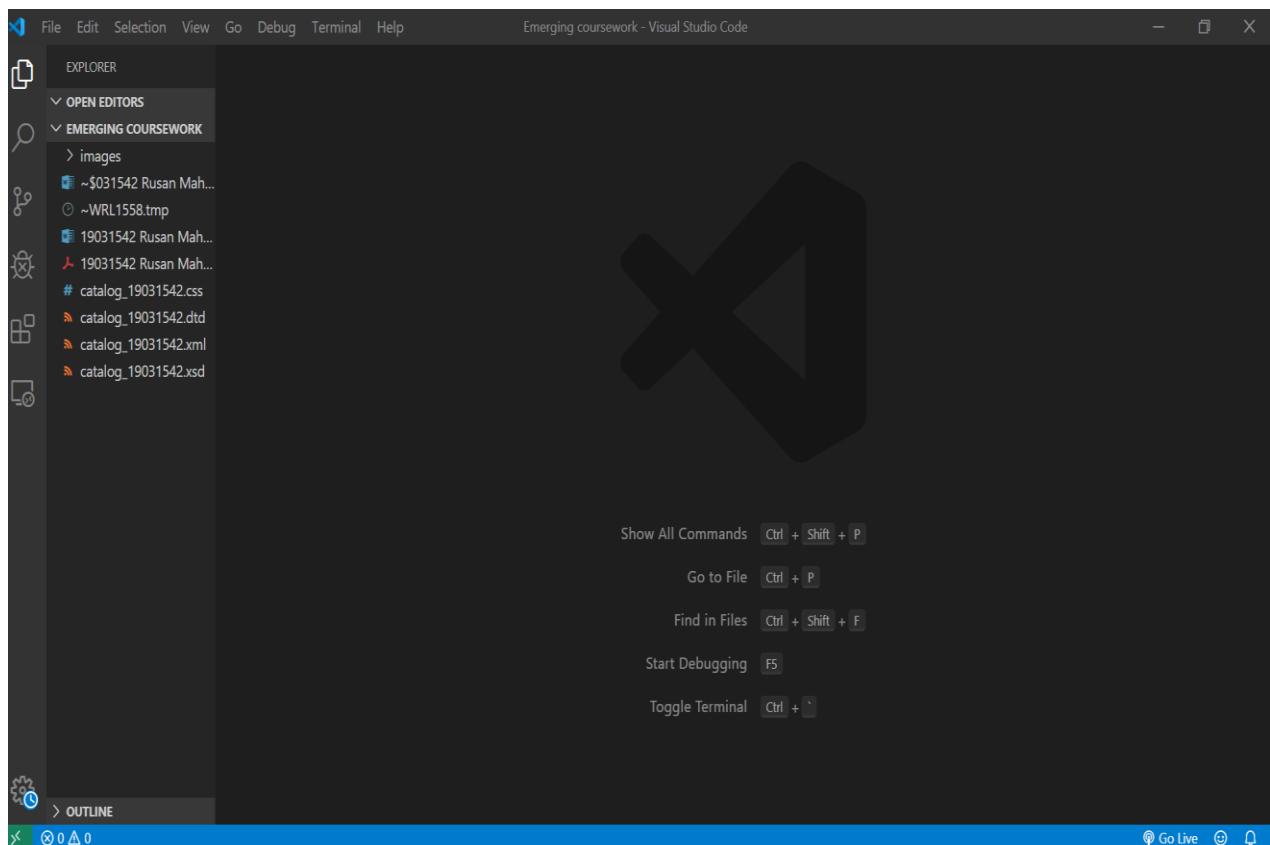
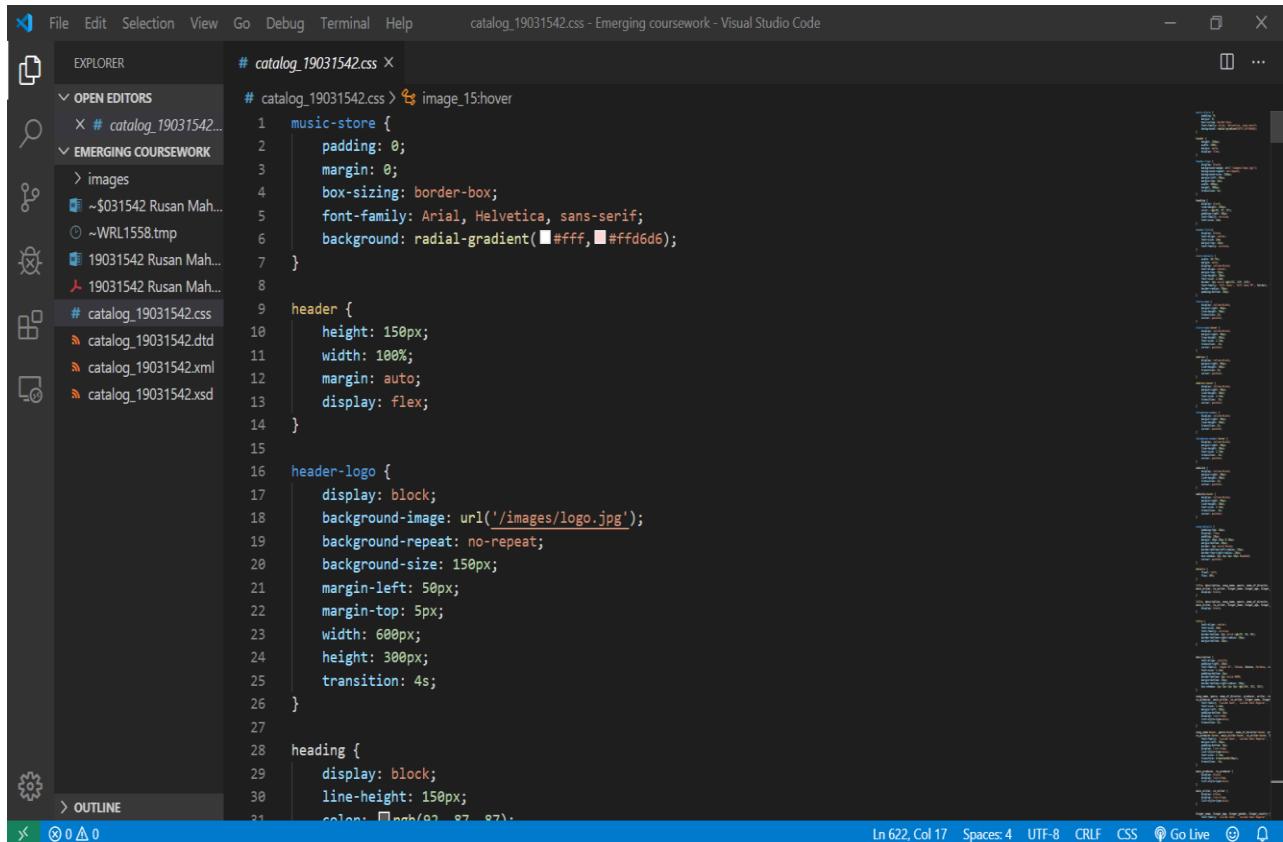


Figure 2: Visual Studio Code

CSS (Cascading Style Sheet):

Cascading Style Sheet is fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable. Using CSS, you can control the colour of the text, font styles, font families, layout designs and variety or other effects (tutorialsPoint, 2021).

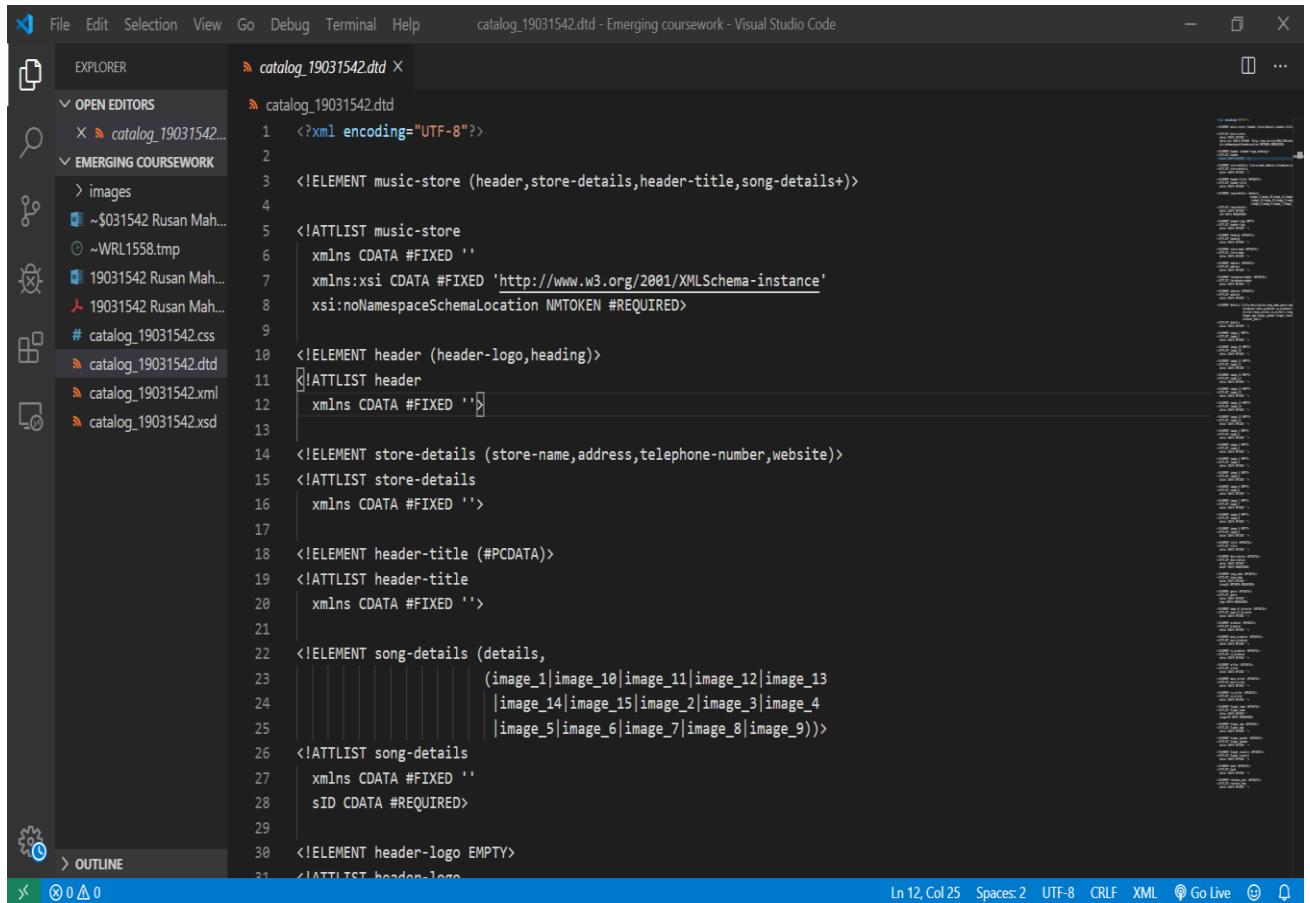


```
# catalog_19031542.css
# catalog_19031542.css > image_15:hover
1   music-store {
2     padding: 0;
3     margin: 0;
4     box-sizing: border-box;
5     font-family: Arial, Helvetica, sans-serif;
6     background: radial-gradient( #fff, #ffd6d6);
7   }
8
9   header {
10    height: 150px;
11    width: 100%;
12    margin: auto;
13    display: flex;
14  }
15
16  header-logo {
17    display: block;
18    background-image: url('/images/logo.jpg');
19    background-repeat: no-repeat;
20    background-size: 150px;
21    margin-left: 50px;
22    margin-top: 5px;
23    width: 600px;
24    height: 300px;
25    transition: 4s;
26  }
27
28  heading {
29    display: block;
30    line-height: 150px;
31    color: #f0f0f0;
32  }
```

Figure 3: CSS (Cascading Style Sheet)

DTD (Document Type Declaration):

The XML Document Type Declaration, commonly known as DTD, is a way to describe XML language precisely. An XML DTD can be either specified inside the document, or it can be kept in a separate document and then linked separately (tutorialspoint, 2021).



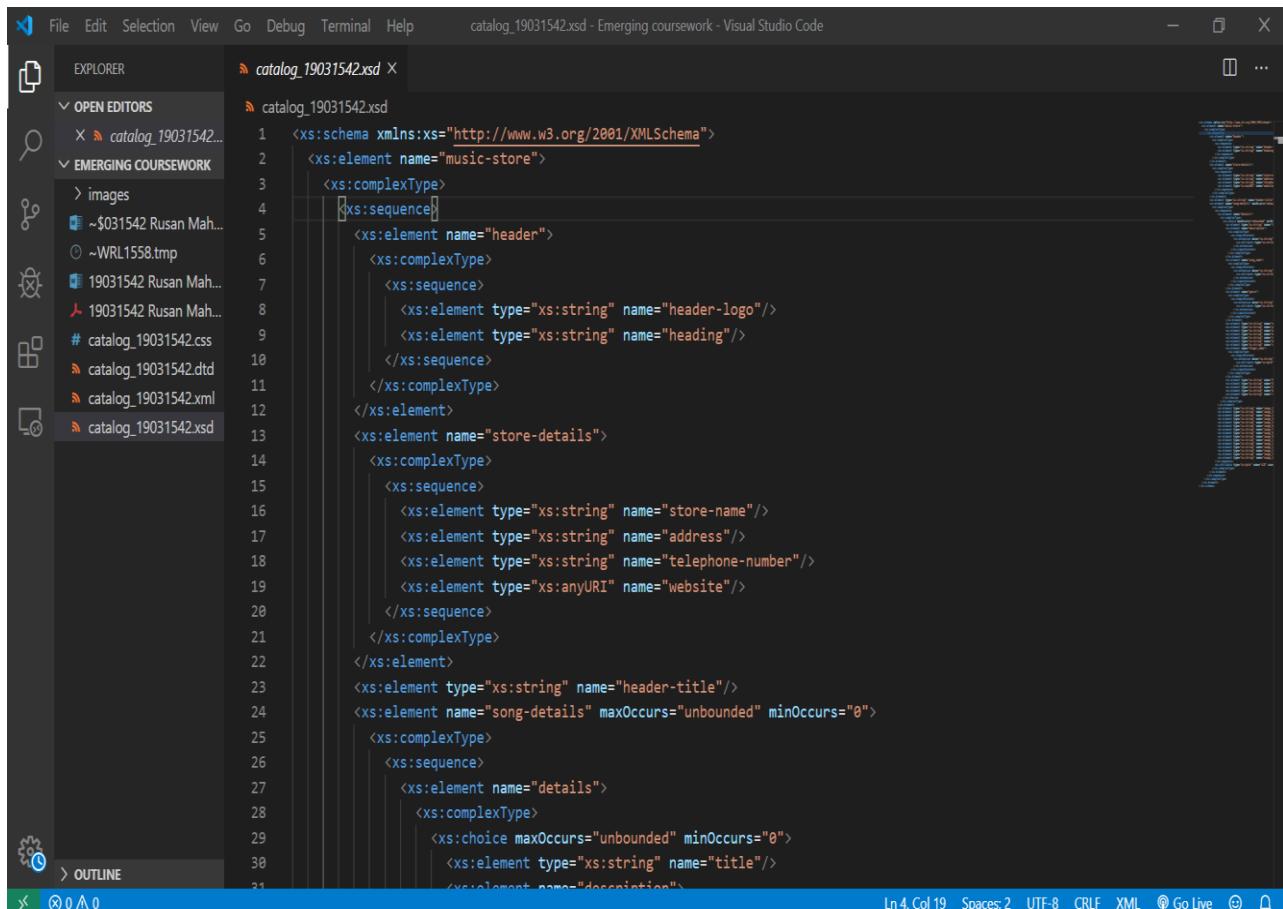
A screenshot of the Visual Studio Code interface. The title bar reads "catalog_19031542.dtd - Emerging coursework - Visual Studio Code". The left sidebar shows the "EXPLORER" view with files like "catalog_19031542.dtd", "catalog_19031542.css", "catalog_19031542.xml", and "catalog_19031542.xsd". The main editor area displays the XML DTD code:

```
<?xml encoding="UTF-8"?>
<!ELEMENT music-store (header,store-details,header-title,song-details+)>
<!ATTLIST music-store
      xmlns CDATA #FIXED ''
      xmlns:xsi CDATA #FIXED 'http://www.w3.org/2001/XMLSchema-instance'
      xsi:noNamespaceSchemaLocation NMTOKEN #REQUIRED>
<!ELEMENT header (header-logo,heading)>
<!ATTLIST header
      xmlns CDATA #FIXED ''>
<!ELEMENT store-details (store-name,address,telephone-number,website)>
<!ATTLIST store-details
      xmlns CDATA #FIXED ''>
<!ELEMENT header-title (#PCDATA)>
<!ATTLIST header-title
      xmlns CDATA #FIXED ''>
<!ELEMENT song-details (details,
      (image_1|image_10|image_11|image_12|image_13
      |image_14|image_15|image_2|image_3|image_4
      |image_5|image_6|image_7|image_8|image_9))>
<!ATTLIST song-details
      xmlns CDATA #FIXED ''
      sID CDATA #REQUIRED>
<!ELEMENT header-logo EMPTY>
<!ATTLIST header-logo
```

Figure 4: DTD (Document Type Declaration)

Schema (XSD):

XML Schema is commonly known as XML Schema Definition (XSD). It is used to describe and validate the structure and the content of XML data. XML schema defines the elements, attributes and data types. Schema element supports Namespaces (tutorialspoint, 2021).



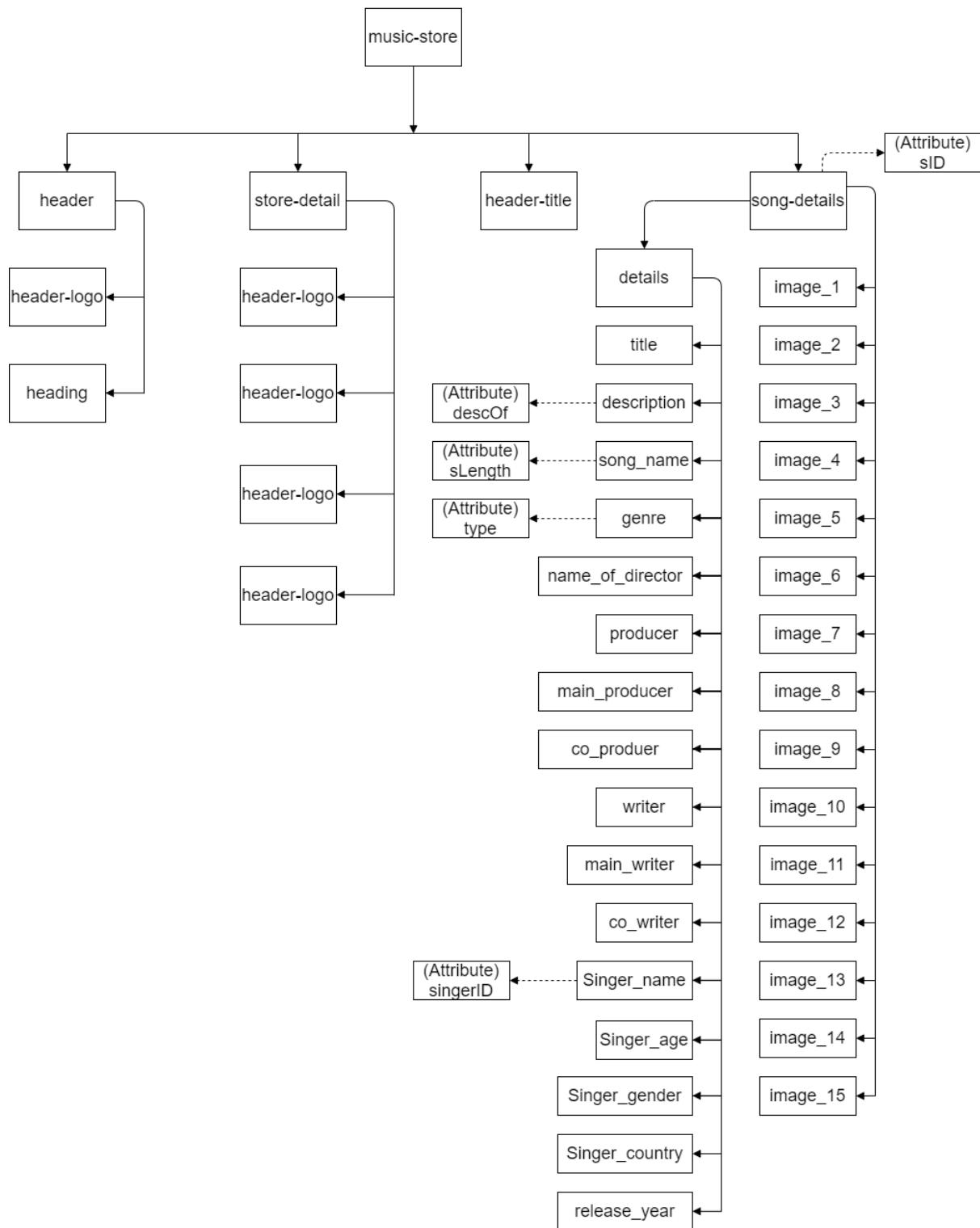
```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="music-store">
<xs:complexType>
<xs:sequence>
<xs:element name="header">
<xs:complexType>
<xs:sequence>
<xs:element type="xs:string" name="header-logo"/>
<xs:element type="xs:string" name="heading"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="store-details">
<xs:complexType>
<xs:sequence>
<xs:element type="xs:string" name="store-name"/>
<xs:element type="xs:string" name="address"/>
<xs:element type="xs:string" name="telephone-number"/>
<xs:element type="xs:anyURI" name="website"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element type="xs:string" name="header-title"/>
<xs:element name="song-details" maxOccurs="unbounded" minOccurs="0">
<xs:complexType>
<xs:sequence>
<xs:element name="details">
<xs:complexType>
<xs:choice maxOccurs="unbounded" minOccurs="0">
<xs:element type="xs:string" name="title"/>
<xs:element name="description"/>
```

Figure 5: Schema

XML Content

Tree Diagram

Tree Diagram

**Figure 6: Tree Diagram**

XML Content

XML stands for Extensible Markup Language. It is a text-based markup language derived from Standard Generalization Markup Language (SGML). XML tag recognize the data and it is used to store and organize data. XML has three important characteristics which make it useful in a variety of systems and solutions: XML is extensible, XML carries the data, does not present it, and XML is a public standard (tutorialspoint, 2021).

The XML code of this coursework is given below:

XML Code:

```
1. <?xml version="1.0" encoding="UTF-8"?>
2.
3. <!-- Css External Link -->
4. <?xml-stylesheet type="text/css" href="catalog_19031542.css"?>
5.
6. <!-- DTD External Link -->
7. <!DOCTYPE music-store SYSTEM "catalog_19031542.dtd">
8.
9.
10.<music-store      xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
    xs:noNamespaceSchemaLocation="catalog_19031542.xsd">
11.
12.<!-- Header -->
13.  <header>
14.    <header-logo/>
15.    <heading>Heist Music Store</heading>
16.  </header>
17.
18.
19.<!-- Store-details -->
20.  <store-details>
21.    <store-name>Heist Music Store</store-name>
22.    <address>Harisiddhi, Lalitpur</address>
```

```
23.    <telephone-number>01-5250417</telephone-number>
24.    <website>www.HeistMusicStore.com.np</website>
25.    </store-details>
26.
27.
28. <!-- header title -->
29.    <header-title>Songs</header-title>
30.
31. <!-- Song details starts from here -->
32. <!-- What it feels like for a girl -->
33.    <song-details sID="01">
34.        <details>
35.            <title>What It Feels Like for a Girl</title>
36.            <description desOf="What It Feels Like for a Girl">
37.                <![CDATA[
38.                    "What It Feels Like for a Girl" is a song recorded by American
39.                    Singer Madonna for her eighth studio album Music. It was released as the third
40.                    and final single from the album by Maveric Records.
41.                ]]>
42.            </description>
43.            <song_name sLength="4:35">Name: What It Feels Like For a
44.            Girl</song_name>
45.            <genre type="Electronic">Genre: Pop</genre>
46.            <name_of_director>Director: Sigsworth</name_of_director>
47.            <producer>Producer: Madonna</producer>
48.            <writer>Writer: Madonna</writer>
49.            <Singer_name singerID="001">Signer Name:
50.            Madonna</Singer_name>
51.            <Singer_age>Singer Age: 62</Singer_age>
52.            <Singer_gender>Gender: Female</Singer_gender>
53.            <Singer_country>Country: United States</Singer_country>
54.            <release_year>Release Year: April 16, 2001</release_year>
55.        </details>
```

```
53.      <!-- Image 1 for what it feels like for a girl -->
54.      <image_1/>
55.      </song-details>
56.
57.      <!-- Beliver -->
58.      <song-details sID="02">
59.          <details>
60.              <title>Believer</title>
61.              <description desOf="Believer">
62.                  <![CDATA[
63.                      "Believer" is a song by American Pop Rock band Imagine Dragons.
The song was released on February 1, 2017 through Interscope Records and
Kidinakorner as the lead single from the band's third studio album.
64.                  ]]>
65.          </description>
66.          <song_name sLength="3:23">Name: Believer</song_name>
67.          <genre type="Arena rock">Genre: Pop Rock</genre>
68.          <name_of_director>Director: Mattman and Robin</name_of_director>
69.          <main_producer>Main-Producer: Mattman</main_producer>
70.          <co_producer>Co-Producer: Robin</co_producer>
71.          <main_writer>Main-Writer: Dan Reynolds, Wayne
Sermon</main_writer>
72.          <co_writer>Co-Writer: Ben McKee, Daniel Platzman</co_writer>
73.          <Singer_name singerID="002">Singer Name: Dan
Reynolds</Singer_name>
74.          <Singer_age>Singer Age: 33</Singer_age>
75.          <Singer_gender>Gender: Male</Singer_gender>
76.          <Singer_country>Country: United States</Singer_country>
77.          <band>Band Name: Imagine Dragons</band>
78.          <release_year>Release Year: February 1, 2017</release_year>
79.      </details>
80.
81.      <!-- Image for Believer -->
82.      <image_2/>
```

```
83. </song-details>
84.
85. <!-- Memories -->
86. <song-details sID="03">
87.   <details>
88.     <title>Memories</title>
89.     <description desOf="Memories">
90.       <![CDATA[
91.         "Memories" is a song by American pop band Maroon 5, released
92.         through 222 on September 20, 2019. Lyrically, the song pays homepage to the
93.         memories of a lobed one who has since passed.
94.       ]]>
95.     </description>
96.     <song_name sLength="3:05">Name: Memories</song_name>
97.     <genre type="pop">Genre: Pop</genre>
98.     <name_of_director>Director: Vincent Ford</name_of_director>
99.     <producer>Producer: Jacod Kasher Hindlin</producer>
100.    <writer>Writer: Adam Levine, Stefan Johnson, Jordan
101.      Johnson</writer>
102.    <Singer_name singerID="003">Singer Name: Adam
103.      Levine</Singer_name>
104.    <Singer_age>Singer Age: 42</Singer_age>
105.    <Singer_gender>Gender: Male</Singer_gender>
106.    <Singer_country>Country: United States</Singer_country>
107.    <band>Band Name: Maroon 5</band>
108.    <release_year>Release Year: September 20, 2019</release_year>
109.  </details>
110.
111. <!-- Levitating -->
112. <song-details sID="04">
```

```
113.    <details>
114.        <title>Levitating</title>
115.        <description desOf="Levitating">
116.        <![CDATA[
117.            "Levitating" is a song by English singer Dua Lipa from her second
118.            studio album, Future Nostalgia (2020). The song was written by Lipa, Clarence
119.            Coffee Jr., Sarah Hudson and Stephen Kozmeniuk, with the image of Mike
120.            Myers in Austin Powers in mind.
121.        ]]>
122.        </description>
123.        <song_name sLength="3:41">Name: Levitating</song_name>
124.        <genre type="Electro-disco">Genre: pop</genre>
125.        <name_of_director>Director: Stephen
126.            Kozmeniuk</name_of_director>
127.        <producer>Producer: Stuart Price</producer>
128.        <writer>Writer: Clarence Coffee Jr, Sarah Hudson</writer>
129.        <Singer_name      singerID="004">Singer      Name:      Dua
130.            Lipa</Singer_name>
131.        <Singer_age>Singer Age: 25</Singer_age>
132.        <Singer_gender>Gender: Female</Singer_gender>
133.        <Singer_country>Country: England</Singer_country>
134.        <release_year>Release Year: October 1, 2020</release_year>
135.    </details>
136.    <!-- Image for Levitating -->
137.    <image_4/>
138.    </song-details>
139.    <!-- Break My Heart -->
140.    <song-details sID="05">
141.        <details>
```

142. "Break My Heart" is a song by English singer Dua Lipa from her second studio album, Future Nostalgia (2020). The song was written by Lipa, Ali Tamposi, Stefan Johnson, Jordan K. Johnson, and Andrew Watt, with the latter of the five handling production alongside the Monsters & Strangerz.

143.]]>

144. </description>

145. <song_name sLength="3:48">Name: Break My Heart</song_name>

146. <genre type="Dance-Pop">Genre: Disco</genre>

147. <name_of_director>Director: Andrew Wortman</name_of_director>

148. <main_producer>Main-Producer: Andrew Wortman</main_producer>

149. <co_producer>Co-Producer: Andrew Watt</co_producer>

150. <main_writer>Main-Writer: Stefan Johnson</main_writer>

151. <co_writer>Co-Writer: Michael Hutchence</co_writer>

152. <Singer_name singerID="005">Singer Name: Dua Lipa</Singer_name>

153. <Singer_age>Singer Age: 25</Singer_age>

154. <Singer_gender>Gender: Female</Singer_gender>

155. <Singer_country>Country: England</Singer_country>

156. <release_year>Release Year: March 25, 2020</release_year>

157. </details>

158.

159. <!-- Image for Break My Heart -->

160. <image_5/>

161. </song-details>

162.

163. <!-- Friends -->

164. <song-details sID="06">

165. <details>

166. <title>Friends</title>

167. <description desOf="Friends">

168. <![CDATA[

169. "Friends" (stylized in all caps) is a song by American record producer Marshmello and English singer Anne-Marie. It was written and

produced by Marshmello, with additional writing from Anne-Marie, Eden Anderson, Richard Boardman, Jasmine Thompson, Nat Dunn, Sarah Blanchard and Pablo Bowman.

```
170.      ]]>
171.      </description>
172.      <song_name sLength="3:26">Name: Friends</song_name>
173.      <genre type="EDM">Genre: Pop</genre>
174.      <name_of_director>Director: Marshmello</name_of_director>
175.      <producer>Producer: Marshmello</producer>
176.      <writer>Writer: Eden Andreson, Natalie Dunn</writer>
177.      <Singer_name      singerID="006">Singer      Name:      Anne
         Marie</Singer_name>
178.      <Singer_age>Singer Age: 30</Singer_age>
179.      <Singer_gender>Gender: Female</Singer_gender>
180.      <Singer_country>Country: England</Singer_country>
181.      <release_year>Release Year: February 9, 2018</release_year>
182.      </details>
183.
184.      <!-- Image for Friends -->
185.      <image_6/>
186.      </song-details>
187.
188.      <!-- At my worst -->
189.      <song-details sID="07">
190.          <details>
191.              <title>At My Worst</title>
192.              <description desOf="At My Worst">
193.                  <![CDATA[
194.                      Pink Sweat's "At My Worst" is one of his latest hits, and it's
             already racked up more than 34 million Spotify plays to date. The song, which
             is produced by John Hill and Pink Sweat$, appears on his 2020 EP, The
             Prelude. On the track, the Philly native sings about finding a partner who will
             stick with him despite his flaws.
195.                  ]]>
```

```
196.      </description>
197.      <song_name sLength="3:12">Name: At my worst</song_name>
198.      <genre type="pop">Genre: soul pop</genre>
199.      <name_of_director>Director: David Bowden</name_of_director>
200.      <producer>Producer: David Bowden</producer>
201.      <writer>Writer: David Bowden</writer>
202.      <Singer_name      singerID="007">Singer      Name:      David
          Bowden</Singer_name>
203.      <Singer_age>Singer Age: 29</Singer_age>
204.      <Singer_gender>Gender: Male</Singer_gender>
205.      <Singer_country>Country: United States</Singer_country>
206.      <band>Band Name: Pink Sweat's</band>
207.      <release_year>Release Year: November 12, 2020</release_year>
208.      </details>
209.
210.      <!-- Image for At my worst -->
211.      <image_7/>
212.      </song-details>
213.
214.      <!-- I'm a Mess -->
215.      <song-details sID="08">
216.      <details>
217.          <title>I'm a Mess</title>
218.          <description desOf="I'm a Mess">
219.              <![CDATA[
220.                  "I'm a Mess" is a song recorded by American singer Bebe Rexha
          for her debut studio album, Expectations (2018). It was released as the first and
          only single from the album on June 15, 2018, following an early radio release
          in the United States.
221.              ]]>
222.          </description>
223.          <song_name sLength="3:16">Name: I'm a Mess</song_name>
224.          <genre type="Pop">Genre: Pop Music</genre>
225.          <name_of_director>Director: Shelly Peiken</name_of_director>
```

```
226.      <main_producer>Main-Producer: Jussifer</main_producer>
227.      <co_producer>Co-Producer: Devon Corey</co_producer>
228.      <writer>Writer: Justin Tranter</writer>
229.      <Singer_name      singerID="008">Singer      Name:      Bebe
          Rexha</Singer_name>
230.      <Singer_age>Singer Age: 31</Singer_age>
231.      <Singer_gender>Gender: Female</Singer_gender>
232.      <Singer_country>Country: United State</Singer_country>
233.      <release_year>Release Year: June 15, 2018</release_year>
234.      </details>
235.
236.      <!-- Image for I'm a Mess -->
237.      <image_8/>
238.      </song-details>
239.
240.      <!-- Psycho -->
241.      <song-details sID="09">
242.      <details>
243.          <title>Psycho</title>
244.          <description desOf="Psycho">
245.              <![CDATA[
246.                  "Psycho" is a song recorded by American rapper and singer Post
          Malone, featuring vocals from American singer Ty Dolla Sign. It was released
          through Republic Records on February 23, 2018, as the third single from
          Malone's second studio album.
247.              ]]>
248.          </description>
249.          <song_name sLength="3:57">Name: Psycho</song_name>
250.          <genre type="Rock">Genre: Rock</genre>
251.          <name_of_director>Director: Tyrone Griffin</name_of_director>
252.          <producer>Producer: Louis Bell, Post Malone</producer>
253.          <writer>Writer: Austin Post, Louis Bell</writer>
254.          <Singer_name      singerID="009">Singer      Name:      Post
          Malone</Singer_name>
```

```
255.      <Singer_age>Singer Age: 25</Singer_age>
256.      <Singer_gender>Gender: Male</Singer_gender>
257.      <Singer_country>Country: United States</Singer_country>
258.      <release_year>Release Year: February 23, 2018</release_year>
259.      </details>
260.
261.      <!-- Image for Psycho -->
262.      <image_9/>
263.      </song-details>
264.
265.      <!-- Lonely -->
266.      <song-details sID="10">
267.          <details>
268.              <title>Lonely</title>
269.              <description desOf="Lonely">
270.                  <![CDATA[
271.                      "Lonely" (also known as "Mr. Lonely") is a song by Senegalese-
272.                        American singer and rapper Akon; it appears on his debut album, Trouble.
273.                  ]]>
274.              </description>
275.              <song_name sLength="4:25">Name: Lonely</song_name>
276.              <genre type="pop">Genre: Jazz</genre>
277.              <name_of_director>Director: Bobby Vinton</name_of_director>
278.              <producer>Producer: Akon</producer>
279.              <writer>Writer: Bobby Vinton, Gene Allan</writer>
280.              <Singer_name singerID="010">Singer Name: Akon</Singer_name>
281.              <Singer_age>Singer Age: 48</Singer_age>
282.              <Singer_gender>Gender: Male</Singer_gender>
283.              <Singer_country>Country: United States</Singer_country>
284.              <release_year>Release Year: April 19, 2005</release_year>
285.          </details>
286.          <!-- Image for Lonely -->
287.          <image_10/>
```

```
288.    </song-details>
289.
290.    <!-- Closer -->
291.    <song-details sID="11">
292.        <details>
293.            <title>Closer</title>
294.            <description desOf="Closer">
295.                <![CDATA[
296.                    "Closer" is a song by American DJ duo the Chainsmokers
297.                    featuring American singer Halsey. Andrew Taggart (one half of the
298.                    Chainsmokers) also provides his vocals in the song. It was released on July 29,
299.                    2016, through Disruptor Records and Columbia Records.
300.                ]]>
301.            </description>
302.            <song_name sLength="4:07">Name: Closer</song_name>
303.            <genre type="Future bass">Genre: Pop</genre>
304.            <name_of_director>Director: Joe King</name_of_director>
305.            <producer>Producer: Ashley Frangipane</producer>
306.            <writer>Writer: Shaun Frank, Isaac Slade</writer>
307.            <Singer_name singerID="011">Singer Name: Chainsmoker and
308.                Halsey</Singer_name>
309.            <Singer_age>Singer Age: 34 and 26</Singer_age>
310.            <Singer_gender>Gender: Male and Female</Singer_gender>
311.            <Singer_country>Country: United States</Singer_country>
312.            <release_year>Release Year: July 29, 2016</release_year>
313.        </details>
314.
315.    <!-- Image for Closer -->
316.    <image_11/>
317.    </song-details>
```

```
318.      <title>Dance Monkey</title>
319.      <description desOf="Dance Monkey">
320.          <![CDATA[
321.              "Dance Monkey" is a song by Australian singer Tones and I,
            released on 10 May 2019 as the second single (first in the US) from Tones and
            I's debut EP The Kids Are Coming. The song was produced and mixed by
            Konstantin Kersting.
322.          ]]>
323.      </description>
324.      <song_name sLength="3:29">Name: Dance Monkey</song_name>
325.      <genre type="Electric Pop">Genre: Pop</genre>
326.      <name_of_director>Director: Toni Watson</name_of_director>
327.      <producer>Producer: Konstanitin Kersting</producer>
328.      <writer>Writer: Toni Watson</writer>
329.      <Singer_name      singerID="012">Singer      Name:      Toni
            Watson</Singer_name>
330.      <Singer_age>Singer Age: 22</Singer_age>
331.      <Singer_gender>Gender: Male</Singer_gender>
332.      <Singer_country>Country: Australia</Singer_country>
333.      <release_year>Release Year: 10 May, 2019</release_year>
334.      </details>
335.
336.      <!-- Image for Dance Monkey -->
337.      <image_12/>
338.      </song-details>
339.
340.      <!-- Sweet But Psycho -->
341.      <song-details sID="13">
342.          <details>
343.              <title>Sweet But Psycho</title>
344.              <description desOf="Sweet but Psycho">
345.                  <![CDATA[
346.                      "Sweet but Psycho" is a song by American singer Ava Max,
                        released on August 17, 2018, through Atlantic Records as the first single from
```

her debut studio album, Heaven & Hell (2020). The song was written by Max, Madison Love, Tix, Cook Classics, and the producer Cirkut. It is a pop, dance-pop, electropop, and synth-pop song with lyrics about the perception of a woman.

```
347.          ]]>
348.      </description>
349.      <song_name      sLength="3:08">Name:      Sweet      but
350.          Psycho</song_name>
351.          <genre type="Electric pop">Genre: Jazz</genre>
352.          <name_of_director>Director: Andreas Andresen</name_of_director>
353.          <producer>Producer: Cirkut</producer>
354.          <writer>Writer: William Lobban-Bean</writer>
355.          <Singer_name      singerID="013">Singer      Name:      Ava
356.          Max</Singer_name>
357.          <Singer_age>Singer Age: 27</Singer_age>
358.          <Singer_gender>Gender: Female</Singer_gender>
359.          <Singer_country>Country: United States</Singer_country>
360.          <release_year>Release Year: August 17, 2018</release_year>
361.          </details>
362.          <!-- Image for Sweet But Psycho -->
363.          <image_13/>
364.          </song-details>
365.          <!-- Rap God -->
366.          <song-details sID="14">
367.          <details>
368.          <title>Rap God</title>
369.          <description desOf="Rap God">
370.          <![CDATA[
371.              "Rap God" is a song by American rapper Eminem. The song
372.              premiered via YouTube on October 14, 2013, and was released in the United
373.              States on October 15, 2013, as the third single from Eminem's eighth studio
374.              album, The Marshall Mathers LP 2 (2013).
```

```
372.          ]]>
373.      </description>
374.      <song_name sLength="6:07">Name: Rap God</song_name>
375.      <genre type="rap">Genre: Pop</genre>
376.      <name_of_director>Director: Narsgakk Natgers</name_of_director>
377.      <producer>Producer: Filthy.co</producer>
378.      <main_writer>Main-Writer: Stephen Hacker</main_writer>
379.      <co_writer>Co-Writer: Douglas Davis</co_writer>
380.      <Singer_name           singerID="014">Singer           Name:
381.          Eminem</Singer_name>
382.          <Singer_age>Singer Age: 48</Singer_age>
383.          <Singer_gender>Gender: Male</Singer_gender>
384.          <Singer_country>Country: United States</Singer_country>
385.          <release_year>Release Year: October 15, 2013</release_year>
386.
387.      <!-- Image of Rap God -->
388.      <image_14/>
389.  </song-details>
390.
391.  <!-- Pretty Girl -->
392.  <song-details sID="15">
393.      <details>
394.          <title>Pretty Girl</title>
395.          <description desOf="Pretty Girl">
396.              <![CDATA[
397.                  "Pretty Girl" is a song by American singer and songwriter Maggie
398.                  Lindemann. It is her first major-label single, and was released on September
399.                  30, 2016, through 300 Entertainment. The song was written by Lindemann,
400.                  Sasha Sloan, and Sean Myer, with the production being handled by Jayson
401.                  DeZuzio.
402.          ]]>
403.      </description>
404.      <song_name sLength="3:20">Name: Pretty Girl</song_name>
```

```
401.      <genre type="Pop">Genre: Cheat Codes X Cade remix</genre>
402.      <name_of_director>Director: Sasha Sloan</name_of_director>
403.      <producer>Producer: Jayson DeZuzio</producer>
404.      <writer>Writer: Maggie Linderman</writer>
405.      <Singer_name    singerID="015">Singer    Name:    Maggie
          Linderman</Singer_name>
406.      <Singer_age>Singer Age: 22</Singer_age>
407.      <Singer_gender>Gender: Female</Singer_gender>
408.      <Singer_country>Country: United States</Singer_country>
409.      <release_year>Release Year: September 30, 2016</release_year>
410.      </details>
411.
412.      <!-- Image for Pretty Girl -->
413.      <image_15/>
414.      </song-details>
415. </music-store>
416.
417.
```

Schema Content

XSD stands for XML Schema Definition and it is a way to describe the structure of a XML document. It defines the rules for all the attributes and elements in a XML document. XSD checks the correctness of the structure of the XML file (GeeksforGeeks, 2020).

The XSD code is given below:

Schema Code:

```
1. <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
2.   <xs:element name="music-store">
3.     <xs:complexType>
4.       <xs:sequence>
5.         <xs:element name="header">
6.           <xs:complexType>
7.             <xs:sequence>
8.               <xs:element type="xs:string" name="header-logo"/>
9.               <xs:element type="xs:string" name="heading"/>
10.            </xs:sequence>
11.          </xs:complexType>
12.        </xs:element>
13.        <xs:element name="store-details">
14.          <xs:complexType>
15.            <xs:sequence>
16.              <xs:element type="xs:string" name="store-name"/>
17.              <xs:element type="xs:string" name="address"/>
18.              <xs:element type="xs:string" name="telephone-number"/>
19.              <xs:element type="xs:anyURI" name="website"/>
20.            </xs:sequence>
21.          </xs:complexType>
22.        </xs:element>
23.        <xs:element type="xs:string" name="header-title"/>
24.        <xs:element name="song-details" maxOccurs="unbounded"
   minOccurs="0">
```

```
25.      <xs:complexType>
26.        <xs:sequence>
27.          <xs:element name="details">
28.            <xs:complexType>
29.              <xs:choice maxOccurs="unbounded" minOccurs="0">
30.                <xs:element type="xs:string" name="title"/>
31.                <xs:element name="description">
32.                  <xs:complexType>
33.                    <xs:simpleContent>
34.                      <xs:extension base="xs:string">
35.                        <xs:attribute type="xs:string" name="desOf" use="optional"/>
36.                      </xs:extension>
37.                    </xs:simpleContent>
38.                  </xs:complexType>
39.                </xs:element>
40.                <xs:element name="song_name">
41.                  <xs:complexType>
42.                    <xs:simpleContent>
43.                      <xs:extension base="xs:string">
44.                        <xs:attribute type="xs:string" name="sLength"
   use="optional"/>
45.                      </xs:extension>
46.                    </xs:simpleContent>
47.                  </xs:complexType>
48.                </xs:element>
49.                <xs:element name="genre">
50.                  <xs:complexType>
51.                    <xs:simpleContent>
52.                      <xs:extension base="xs:string">
53.                        <xs:attribute type="xs:string" name="type" use="optional"/>
54.                      </xs:extension>
55.                    </xs:simpleContent>
56.                  </xs:complexType>
57.                </xs:element>
```

```
58.      <xs:element type="xs:string" name="name_of_director"/>
59.      <xs:element type="xs:string" name="producer"/>
60.      <xs:element type="xs:string" name="writer"/>
61.      <xs:element type="xs:string" name="main_producer"/>
62.      <xs:element type="xs:string" name="co_producer"/>
63.      <xs:element type="xs:string" name="main_writer"/>
64.      <xs:element type="xs:string" name="co_writer"/>
65.      <xs:element name="Singer_name">
66.          <xs:complexType>
67.              <xs:simpleContent>
68.                  <xs:extension base="xs:string">
69.                      <xs:attribute type="xs:byte" name="singerID"
70.                          use="optional"/>
71.                  </xs:extension>
72.              </xs:simpleContent>
73.          </xs:complexType>
74.      </xs:element>
75.      <xs:element type="xs:string" name="Singer_age"/>
76.      <xs:element type="xs:string" name="Singer_gender"/>
77.      <xs:element type="xs:string" name="Singer_country"/>
78.      <xs:element type="xs:string" name="band"/>
79.      <xs:element type="xs:string" name="release_year"/>
80.  </xs:choice>
81.  </xs:complexType>
82. </xs:element>
83. <xs:element type="xs:string" name="image_1" minOccurs="0"/>
84. <xs:element type="xs:string" name="image_2" minOccurs="0"/>
85. <xs:element type="xs:string" name="image_3" minOccurs="0"/>
86. <xs:element type="xs:string" name="image_4" minOccurs="0"/>
87. <xs:element type="xs:string" name="image_5" minOccurs="0"/>
88. <xs:element type="xs:string" name="image_6" minOccurs="0"/>
89. <xs:element type="xs:string" name="image_7" minOccurs="0"/>
90. <xs:element type="xs:string" name="image_8" minOccurs="0"/>
91. <xs:element type="xs:string" name="image_9" minOccurs="0"/>
```

```
91.      <xs:element type="xs:string" name="image_10" minOccurs="0"/>
92.      <xs:element type="xs:string" name="image_11" minOccurs="0"/>
93.      <xs:element type="xs:string" name="image_12" minOccurs="0"/>
94.      <xs:element type="xs:string" name="image_13" minOccurs="0"/>
95.      <xs:element type="xs:string" name="image_14" minOccurs="0"/>
96.      <xs:element type="xs:string" name="image_15" minOccurs="0"/>
97.      </xs:sequence>
98.      <xs:attribute type="xs:byte" name="sID" use="optional"/>
99.      </xs:complexType>
100.     </xs:element>
101.     </xs:sequence>
102.     </xs:complexType>
103.   </xs:element>
104. </xs:schema>
105.
106.
```

DTD Content

DTD stands for Document Type Definition and it is a document which defines the structure of an XML document. It is used to describe the attributes of XML language precisely. There are two types of DTD: External DTD and Internal DTD. It checks that a XML document has a valid structure or not (GeeksforGeeks, 2020).

The DTD Code is given below:

DTD Code:

```
1. <!ELEMENT music-store (header,store-details,header-title,song-details+)>
2.
3. <!ATTLIST music-store
4.   xmlns:xs CDATA #FIXED 'http://www.w3.org/2001/XMLSchema-instance'
5.   xs:noNamespaceSchemaLocation NMTOKEN #REQUIRED>
6.
7. <!ELEMENT header (header-logo,heading)>
8. <!ELEMENT store-details (store-name,address,telephone-number,website)>
9. <!ELEMENT header-title (#PCDATA)>
10.<!ELEMENT song-details
     (details,(image_1?,image_2?,image_3?,image_4?,image_5?,image_6?,image
     _7?,image_8?,image_9?,image_10?,
     image_11?,image_12?,image_13?,image_14?,image_15?))>
11.
12.
13.<!ATTLIST song-details sID CDATA #REQUIRED>
14.
15.<!ELEMENT header-logo EMPTY>
16.<!ELEMENT heading (#PCDATA)>
17.<!ELEMENT store-name (#PCDATA)>
18.<!ELEMENT address (#PCDATA)>
19.<!ELEMENT telephone-number (#PCDATA)>
20.<!ELEMENT website (#PCDATA)>
21.
```

22.<!ELEMENT	details
(title,description,song_name,genre,name_of_director,producer*,main_produc	
er*,co_producer*,	
23.	
writer*,main_writer*,co_writer*,Singer_name,Singer_age,Singer_gender,Sing	
er_country,band*,release_year)>	
24.	
25.<!ELEMENT image_1 EMPTY>	
26.<!ELEMENT image_2 EMPTY>	
27.<!ELEMENT image_3 EMPTY>	
28.<!ELEMENT image_4 EMPTY>	
29.<!ELEMENT image_5 EMPTY>	
30.<!ELEMENT image_6 EMPTY>	
31.<!ELEMENT image_7 EMPTY>	
32.<!ELEMENT image_8 EMPTY>	
33.<!ELEMENT image_9 EMPTY>	
34.<!ELEMENT image_10 EMPTY>	
35.<!ELEMENT image_11 EMPTY>	
36.<!ELEMENT image_12 EMPTY>	
37.<!ELEMENT image_13 EMPTY>	
38.<!ELEMENT image_14 EMPTY>	
39.<!ELEMENT image_15 EMPTY>	
40.	
41.<!ELEMENT title (#PCDATA)>	
42.	
43.<!ELEMENT description (#PCDATA)>	
44.<!ATTLIST description desOf CDATA #REQUIRED>	
45.	
46.<!ELEMENT song_name (#PCDATA)>	
47.<!ATTLIST song_name sLength NMOKEN #REQUIRED>	
48.	
49.<!ELEMENT genre (#PCDATA)>	
50.<!ATTLIST genre type CDATA #REQUIRED>	
51.	

```
52.<!ELEMENT name_of_director (#PCDATA)>
53.<!ELEMENT producer (#PCDATA)>
54.<!ELEMENT main_producer (#PCDATA)>
55.<!ELEMENT co_producer (#PCDATA)>
56.<!ELEMENT writer (#PCDATA)>
57.<!ELEMENT main_writer (#PCDATA)>
58.<!ELEMENT co_writer (#PCDATA)>
59.
60.<!ELEMENT Singer_name (#PCDATA)>
61.<!ATTLIST Singer_name singerID CDATA #REQUIRED>
62.
63.<!ELEMENT Singer_age (#PCDATA)>
64.<!ELEMENT Singer_gender (#PCDATA)>
65.<!ELEMENT Singer_country (#PCDATA)>
66.<!ELEMENT band (#PCDATA)>
67.<!ELEMENT release_year (#PCDATA)>
68.
```

Testing

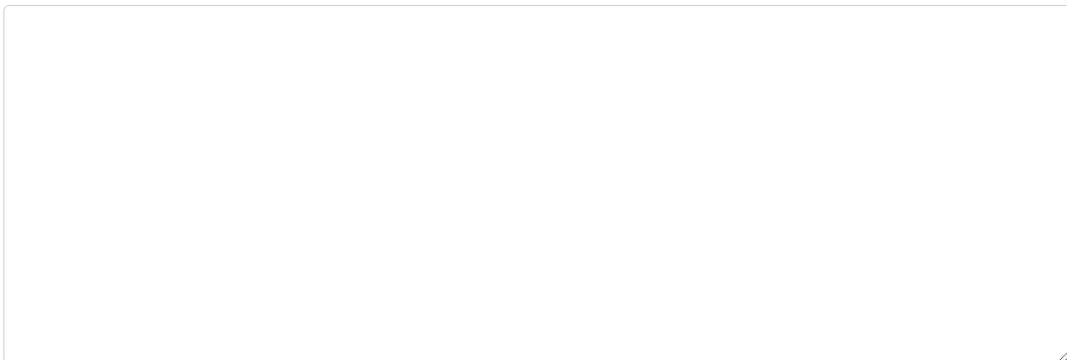
Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software produce is Defect free. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements. Some prefer saying software testing as a White Box and Black Box Testing (Guru99, 2021).

Test 1:

Test No	1
Objective:	Check XML is well form or not
Action:	Checking XML document well form or not
Expected Result:	XML should be well formed
Actual Output:	XML is well formed
Conclusion:	Test is successful.

Table 1: Test 1

Please copy your XML document in here:



Or upload it:

catalog_19031542.xml

The validation check is performed against any XML schema or DTD declared inside the XML document. If neither an XML schema nor a DTD is declared, only a syntax check is performed.

To validate the XML document against an external XML schema, click below.

Validate against external XML schema

validate

Figure 7: Inserting xml file in validating site

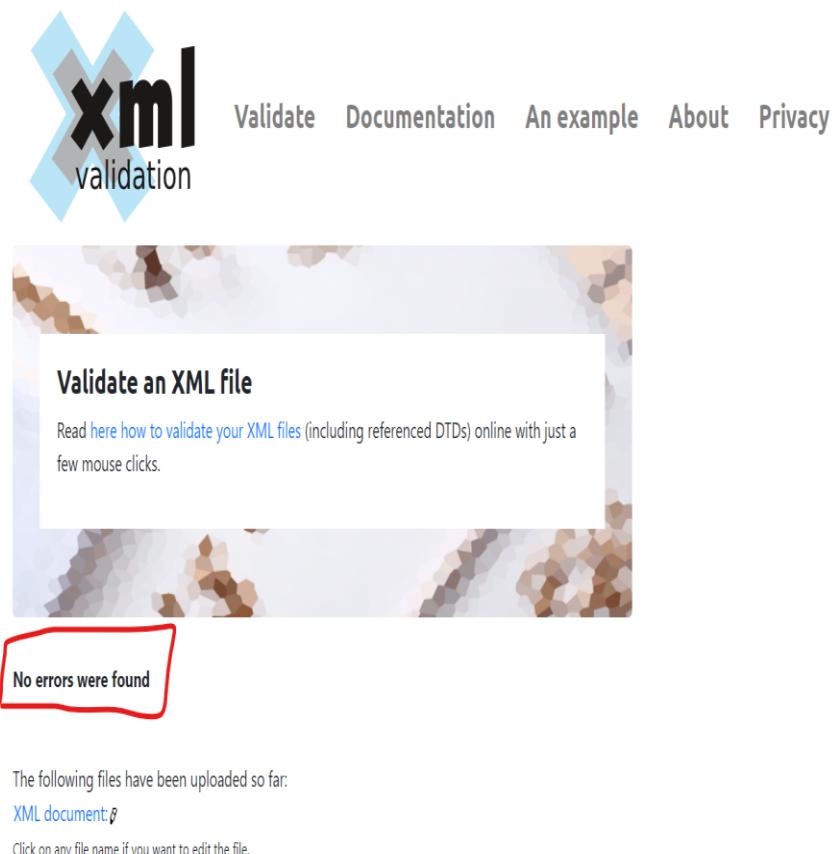


Figure 8: No errors found in xml file

Test 2:

Test No	2
Objective:	Check DTD is error free
Action:	Checking DTD is error free
Expected Result:	DTD should be error free
Actual Output:	Error Displayed
Conclusion:	Test is unsuccessful.

Table 2: Test 2

The file catalog_19031542.dtd is being referenced. Please copy it in here, so that the validation can continue:

Or upload it:

[continue validation](#)

The following files have been uploaded so far:

[XML document](#)

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

Figure 9: Inserting DTD file in validating site

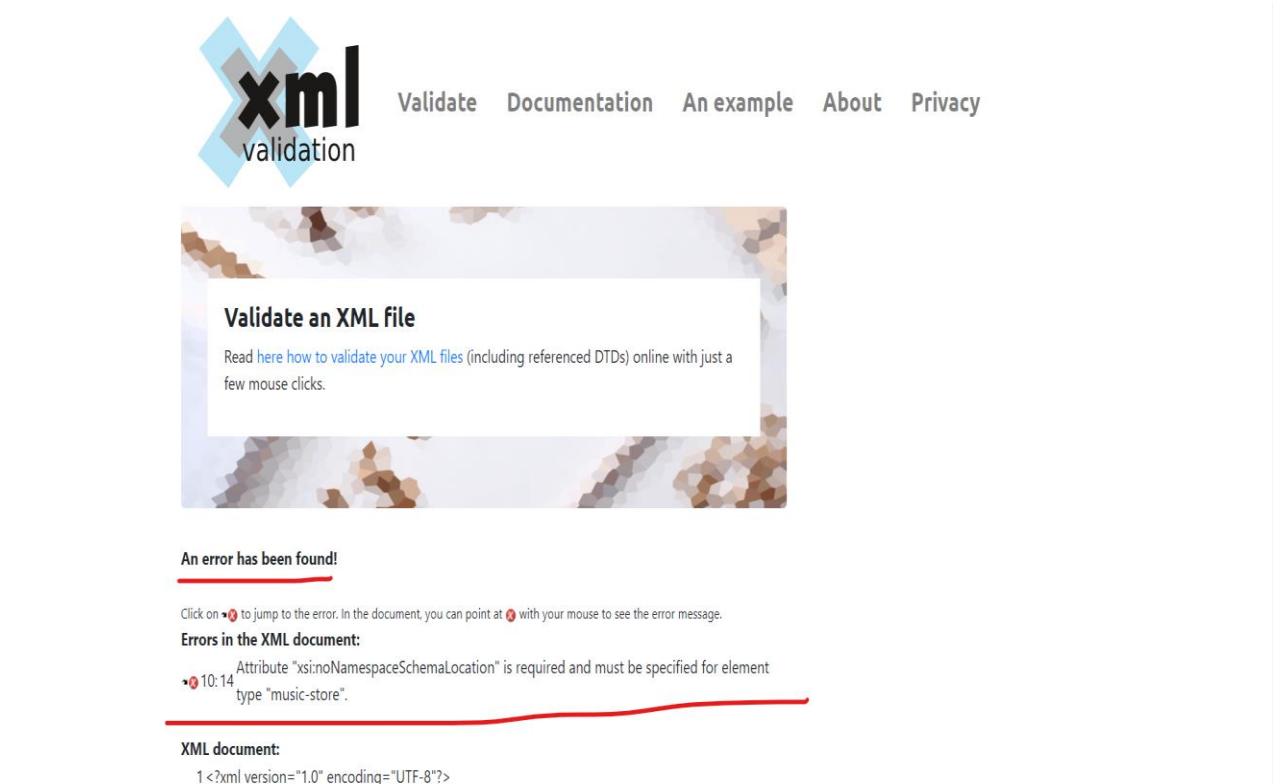


Figure 10: Error occurs during validating DTD

The screenshot shows the Visual Studio Code interface with a dark theme. The left sidebar shows an "EXPLORER" view with files like "catalog_19031542.xml", "catalog_19031542.xsd", and "catalog_19031542.dtd". The main editor window displays the content of "catalog_19031542.dtd". A red border highlights a specific section of the XML code where an error occurred. The code within the red box is:

```

5   <!ATTLIST music-store
6     xmlns CDATA #FIXED ''
7     xmlns:xsi CDATA #FIXED 'http://www.w3.org/2001/XMLSchema-instance'
8     xsi:noNamespaceSchemaLocation NMTOKEN #REQUIRED>

```

Figure 11: Error occurs due to the code inside red border code

Test 3:

Test No	3
Objective:	Check DTD is error free
Action:	Checking DTD is error free
Expected Result:	DTD should be error free
Actual Output:	DTD is error free
Conclusion:	Test is successful.

Table 3: Test 3

The file catalog_19031542.dtd is being referenced. Please copy it in here, so that the validation can continue:

Or upload it:
 catalog_19031542.dtd

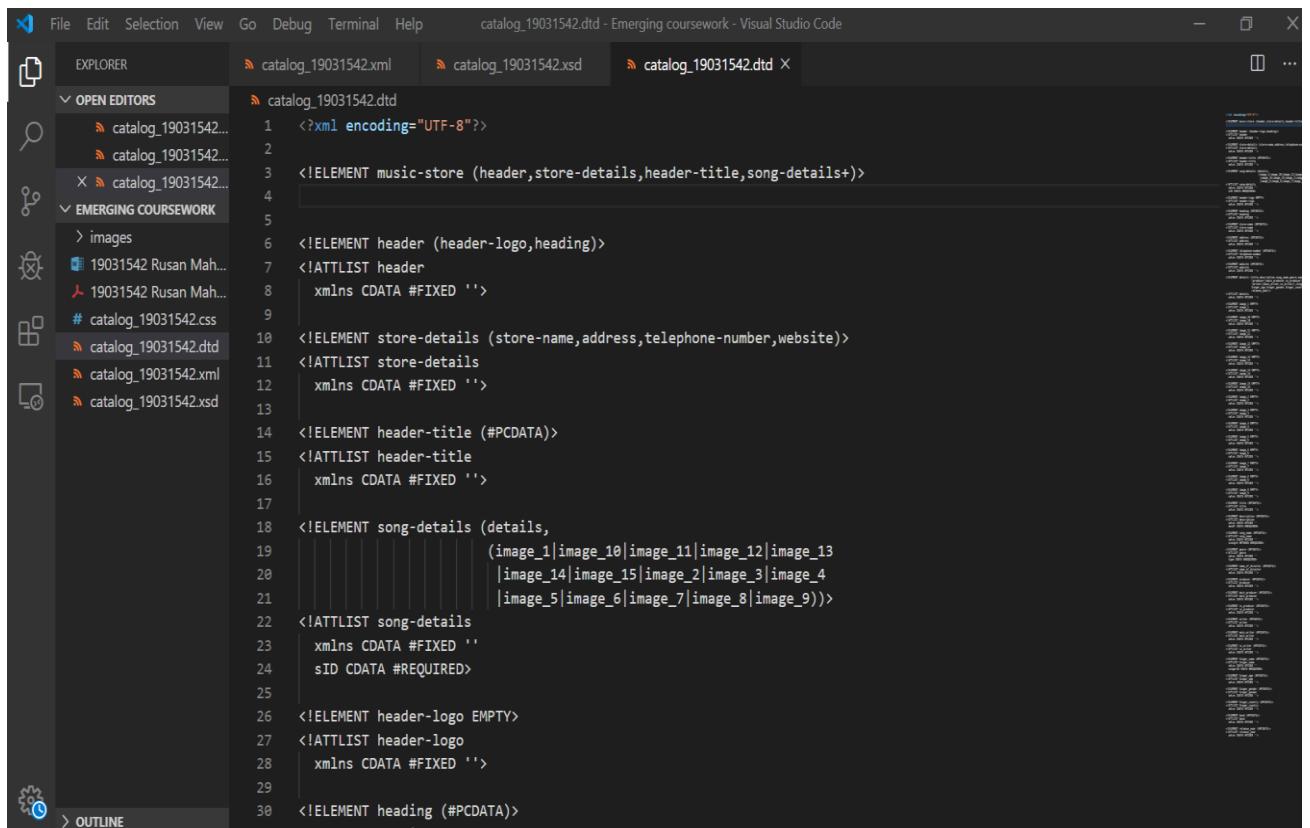
[continue validation](#)

The following files have been uploaded so far:

XML document:

[Click on any file name if you want to edit the file](#)

Figure 12: inserting DTD file in validating site



```
<?xml encoding="UTF-8"?>
<!ELEMENT music-store (header,store-details,header-title,song-details+)>
<!ELEMENT header (header-logo,heading)>
<!ATTLIST header
      xmlns CDATA #FIXED ''>
<!ELEMENT store-details (store-name,address,telephone-number,website)>
<!ATTLIST store-details
      xmlns CDATA #FIXED ''>
<!ELEMENT header-title (#PCDATA)>
<!ATTLIST header-title
      xmlns CDATA #FIXED ''>
<!ELEMENT song-details (details,
      (image_1|image_10|image_11|image_12|image_13
      |image_14|image_15|image_2|image_3|image_4
      |image_5|image_6|image_7|image_8|image_9))>
<!ATTLIST song-details
      xmlns CDATA #FIXED ''
      sID CDATA #REQUIRED>
<!ELEMENT header-logo EMPTY>
<!ATTLIST header-logo
      xmlns CDATA #FIXED ''>
<!ELEMENT heading (#PCDATA)>
```

Figure 13: fixed error in DTD

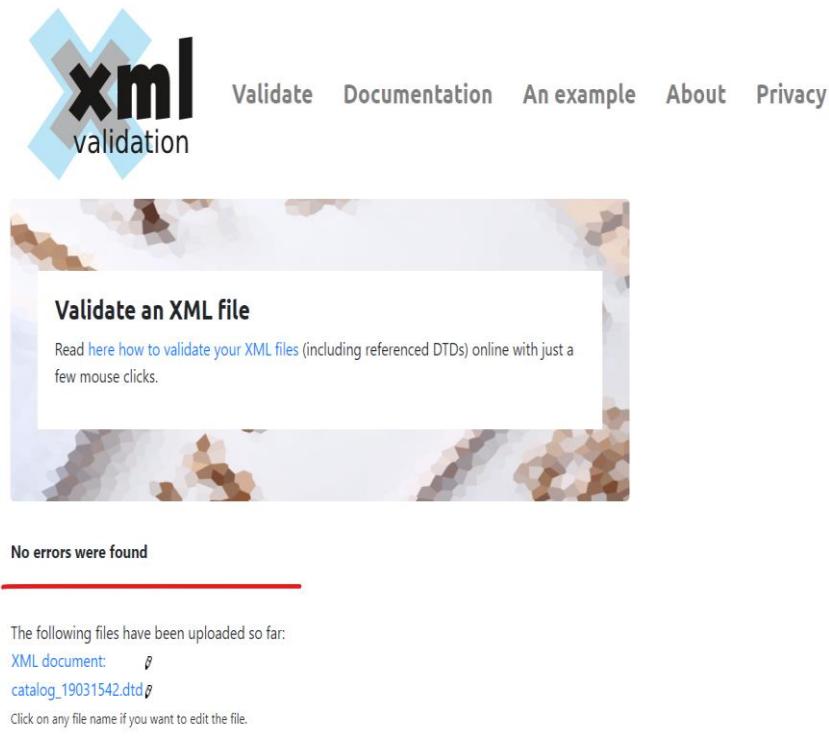


Figure 14: No errors found validating DTD

Test 4:

Test No	4
Objective:	Check Schema is error free
Action:	Checking Schema is error free
Expected Result:	Schema should be error free
Actual Output:	Error Displayed
Conclusion:	Test is unsuccessful.

Table 4: Test 4

The screenshot shows a web page titled "Validate an XML file". It includes instructions on how to validate XML files online. Below this, there is a large text input area labeled "Please copy the XML schema in here:" followed by a smaller text input area labeled "Or upload it:". A red box highlights the "Choose File" button next to the text input, which contains the file name "catalog_19031542.xsd". At the bottom left is a "continue validation" button.

Figure 15: Inserting schema file in validating site

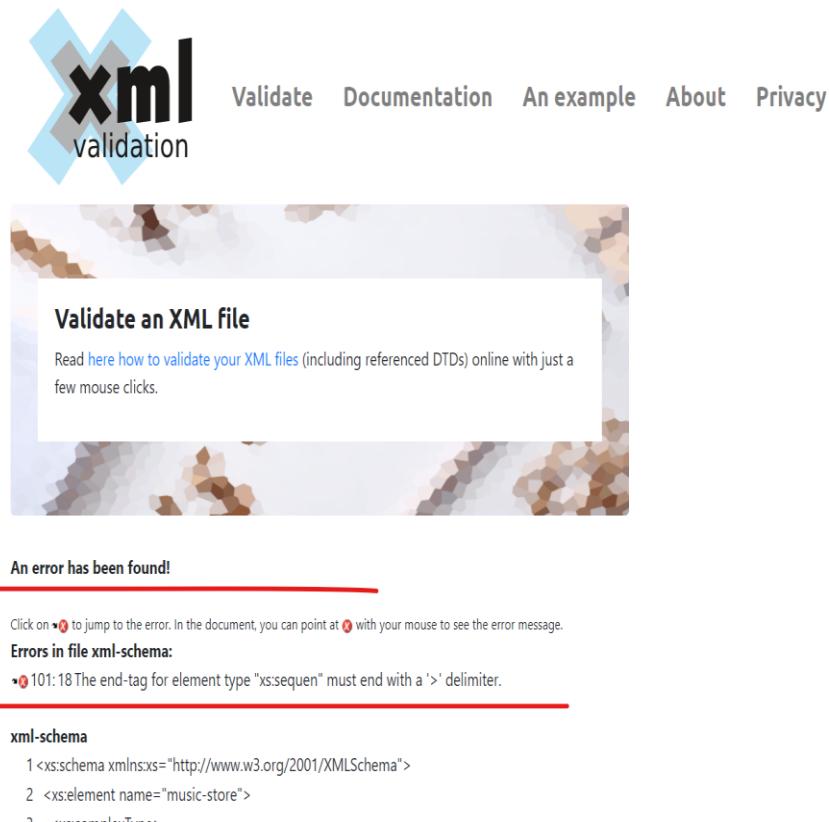


Figure 16: Error occurs during validating schema file

The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Editor Bar:** catalog_19031542.xsd - Emerging coursework - Visual Studio Code.
- Explorer:** Shows files like catalog_19031542.xml, catalog_19031542.xsd, catalog_19031542.dtd, and catalog_19031542.css.
- EMERGING COURSEWORK:** A folder containing images, 19031542 Rusan Mah..., 19031542 Rusan Mah..., catalog_19031542.css, catalog_19031542.dtd, catalog_19031542.xml, and catalog_19031542.xsd.
- Editor Content:** XML schema code for a "music-store".

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="music-store">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="header">
        <xs:complexType>
          <xs:sequence>
            <xs:element type="xs:string" name="header-logo"/>
            <xs:element type="xs:string" name="heading"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="store-details">
        <xs:complexType>
          <xs:sequence>
            <xs:element type="xs:string" name="store-name"/>
            <xs:element type="xs:string" name="address"/>
            <xs:element type="xs:string" name="telephone-number"/>
            <xs:element type="xs:anyURI" name="website"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element type="xs:string" name="header-title"/>
      <xs:element name="song-details" maxOccurs="unbounded" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="details">
              <xs:complexType>
                <xs:choice maxOccurs="unbounded" minOccurs="0">
                  <xs:element type="xs:string" name="title"/>
                  <xs:element name="description"/>
                </xs:choice>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```
- Issues:** A red border highlights the word "sequen" at line 4, column 17, indicating a spelling error.
- Status Bar:** Ln 4, Col 17, Spaces: 2, UTF-8, CRLF, XML, Port: 5500.

Figure 17: Error occurs from code inside red border: wrong tag

Test 5:

Test No	5
Objective:	Check Schema is error free
Action:	Checking Schema is error free
Expected Result:	Schema should be error free
Actual Output:	Schema is error free
Conclusion:	Test is successful.

Table 5: Test 5

The screenshot shows a web page titled "Validate an XML file". It includes a link to "here how to validate your XML files (including referenced DTDs) online with just a few mouse clicks". Below this is a large text input area with the placeholder "Please copy the XML schema in here:". At the bottom left, there is a red rectangular box highlighting a "Choose File" button labeled "catalog_19031542.xsd". A "continue validation" button is located at the bottom right.

Figure 18: Inserting schema file in validating site

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** catalog_19031542.xsd - Emerging coursework - Visual Studio Code.
- Explorer:** Shows files like catalog_19031542.xml, catalog_19031542.xsd, catalog_19031542.dtd, catalog_19031542.css, catalog_19031542.xml, and catalog_19031542.xsd.
- Editor:** Displays the XML schema code. The cursor is at line 4, column 19, under the word "sequence".

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="music-store">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="header">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="header-logo"/>
              <xs:element type="xs:string" name="heading"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="store-details">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="store-name"/>
              <xs:element type="xs:string" name="address"/>
              <xs:element type="xs:string" name="telephone-number"/>
              <xs:element type="xs:anyURI" name="website"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element type="xs:string" name="header-title"/>
        <xs:element name="song-details" maxOccurs="unbounded" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="details">
                <xs:complexType>
                  <xs:choice maxOccurs="unbounded" minOccurs="0">
                    <xs:element type="xs:string" name="title"/>
                    <xs:element name="description"/>
```
- Bottom Status Bar:** Ln 4, Col 19, Spaces: 2, UTF-8, CRLF, XML, Port: 5500.

Figure 19: Error fixed of schema file

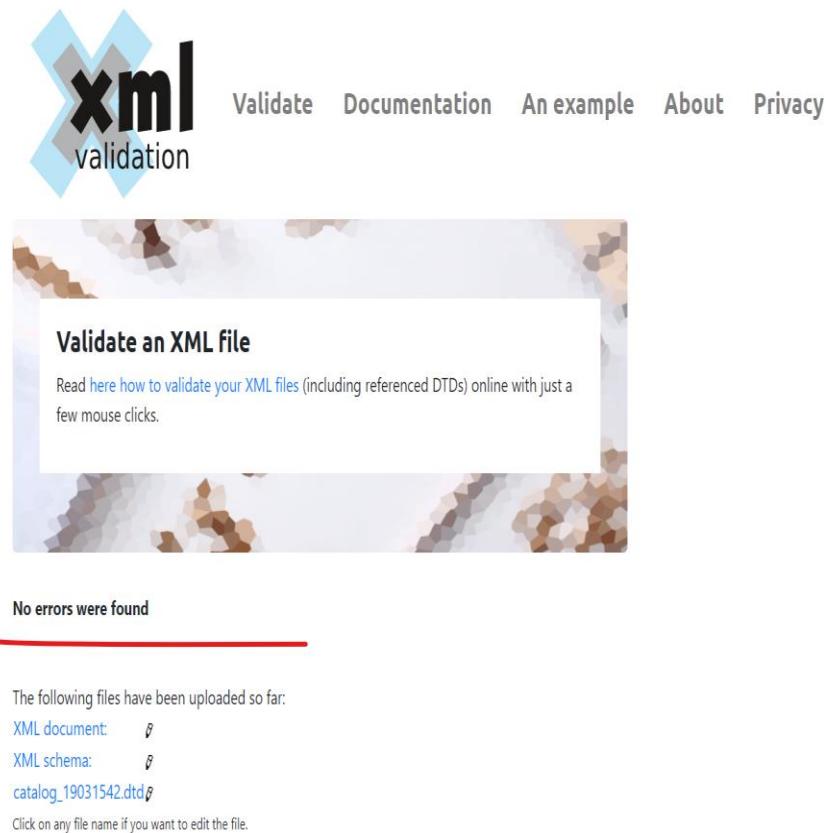


Figure 20: No errors found validating schema file

Test 6:

Test No	6
Objective:	Check XML after applying CSS
Action:	Check XML after applying CSS
Expected Result:	XML should be run with design in it
Actual Output:	XML run with design in it
Conclusion:	Test is successful.

Table 6: Test 6

The screenshot shows a web page with a pink header and footer. The header features a music note icon and the text 'Heist Music Store'. The footer contains contact information: 'Heist Music Store' (address), 'Harisiddhi, Lalitpur', '01-5250417', and 'www.HeistMusicStore.com.np'. The main content area is titled 'Songs' and lists a single song entry:

What It Feels Like for a Girl

"What It Feels Like for a Girl" is a song recorded by American Singer Madonna for her eighth studio album Music. It was released as the third and final single from the album by Maveric Records.

- Name: What It Feels Like For a Girl
- Genre: Pop
- Director: Sigsworth
- Producer: Madonna
- Writer: Madonna
- Signer Name: Madonna
- Singer Age: 62
- Gender: Female
- Country: United States
- Release Year: April 16, 2001

To the right of the text is a circular image of Madonna in a white top and black pants, standing next to a red car. The word 'MADONNA' is written in large letters at the bottom of the image.

Figure 21: XML after applying CSS

How you developed the coursework?

I have studied the given question in the coursework and planned for starting this coursework. I have made a list of tasks to complete this coursework. After completely studying the scenario given in the coursework, I started developing system using XML (Extensible Markup Language), DTD (Document Type Definition) and Schema/XSD (XML Schema Definition). After all coding files are completed, I started validating in online site www.xmlvalidation.com. Some errors occur on DTD and Schema and for the correction took references from different websites, lecture slides, watched tutorial video and have a console with friends of the problem overcame the error. After all coding are correct, I started doing documentation for this coursework. At first, I make a tree diagram of the XML and started writing introduction for this coursework. In this coursework we are assigned to do minimum 5 testing and while doing testing found error in XML document.

For this coursework work I have used 2 tools: draw.io and visual studio code. And 1 website www.xmlvalidation.com is used for validating XML, DTD and Schema content.

Draw.io:

Draw.io is a software for making diagrams and charts. The software allows you to choose from an automatic layout function, or create a custom layout. Draw.io has options for storing saved charts in the cloud, on a server, or network storage at a data center, depending on your needs (Computer Hope, 2020).

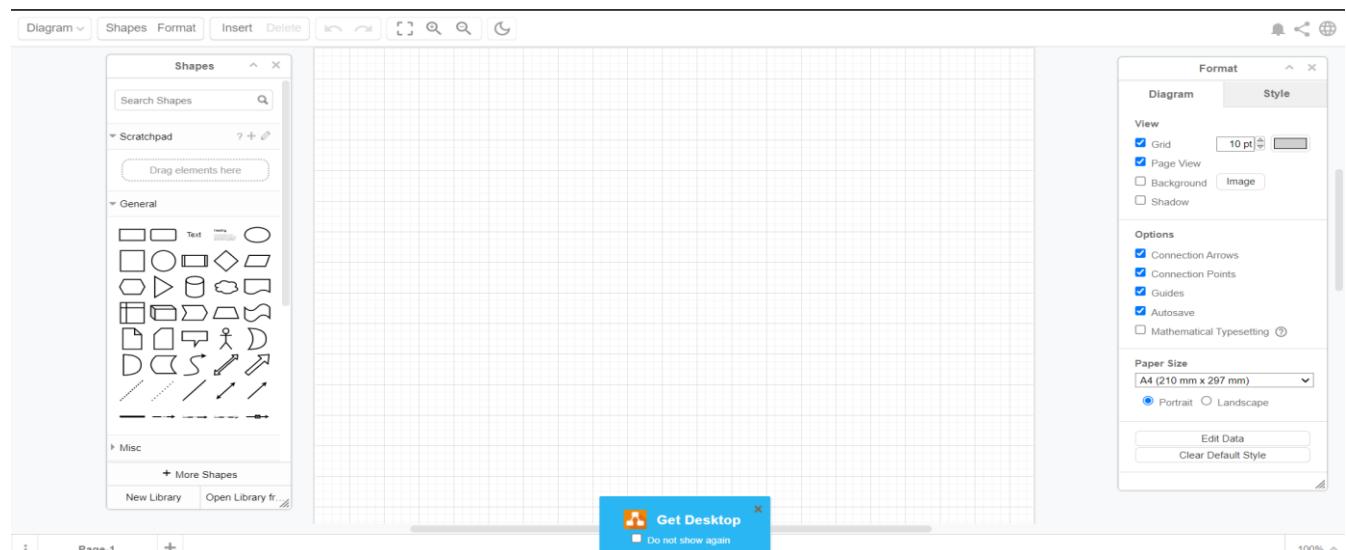


Figure 22: Draw.io

Visual Studio Code:

Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like intelliSense code completion and debugging. Visual Studio Code supports macOS, Linux, and Windows and so you can hit the ground running, no matter the platform (Microsoft, 2021).

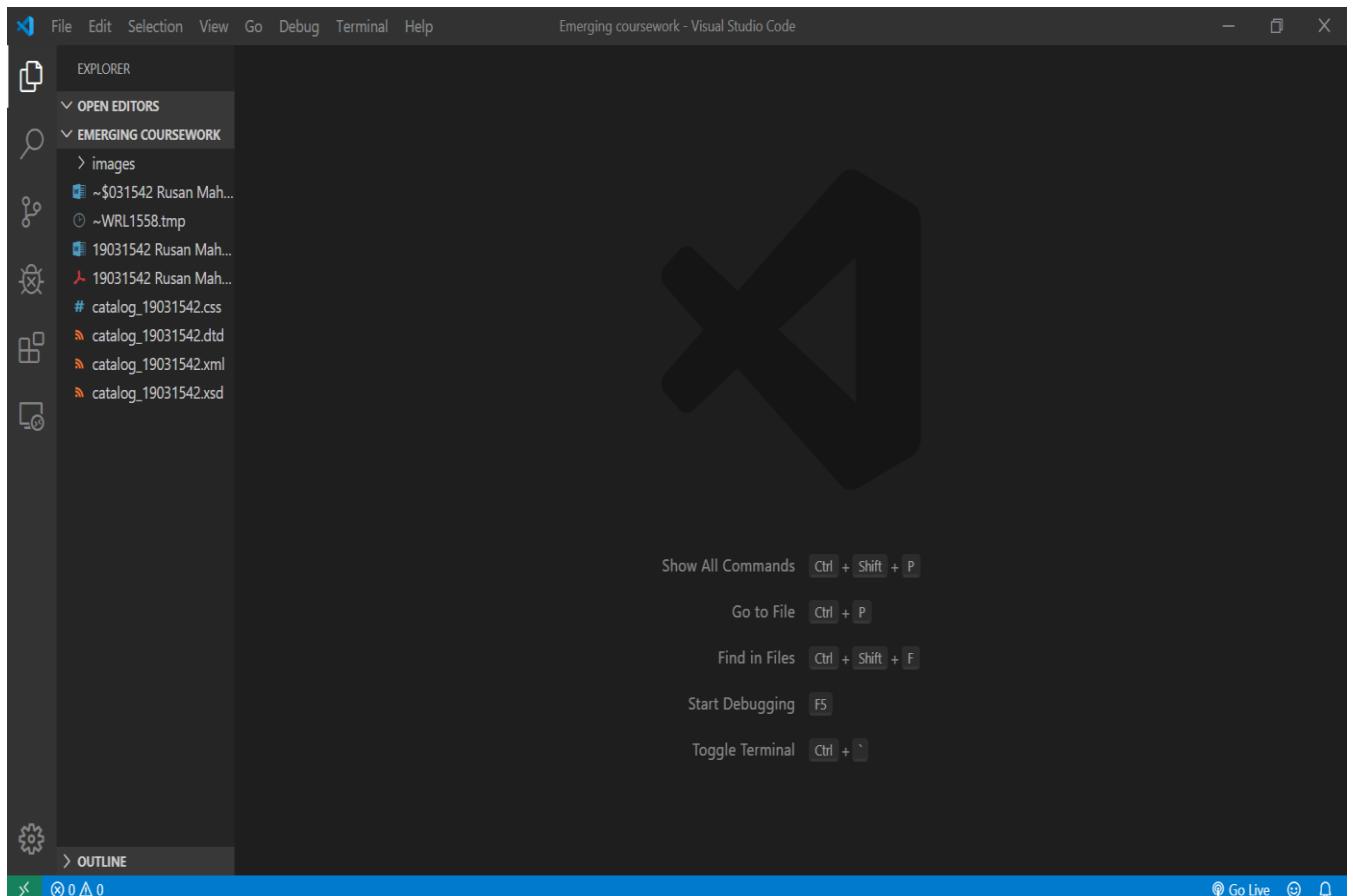


Figure 23: Visual Studio Code

Difference between Schema and DTD

DTD stands for Document Type Definition and it is a document which defines the structure of an XML document. XSD stands for XML Schema Definition and it is a way to describe the structure of XML document (GeeksforGeeks, 2020). The difference between Schema and DTD are:

DTD	Schema (XSD)
DTD are the declarations that define a document type for SGML.	XSD describes the elements in a XML document.
It does not support namespace.	It supports namespace.
It is comparatively harder than XSD.	It is relatively more-simpler than DTD.
It does not support datatypes	It supports datatypes.
SGML syntax is used for DTD	XML is used for writing XSD.
It is not extensible in nature.	It is extensible in nature.
It does not give us much control on structure of XML document (GeeksforGeeks, 2020).	It gives us more control on structure of XML document (GeeksforGeeks, 2020).
DTD does not support namespace.	XSD supports namespace.

Critical Evaluation

This coursework is about making a music store system. It took a lot of time to complete this coursework. It was very difficult to complete this coursework. For making XML document it was easy but designing it through CSS is a bit complex. I have build a music system in which both programmer and beginner programmer can understand code. As per the requirement of coursework, I have researched and put 15 songs, 5 attribute and 5 optional elements. The main problem I have faced in this coursework is to design XML file with CSS. I have faced many problems in this coursework. Some problems are: Image not shown during CSS design, error in DTD and Schema.

DTD and Schema are the one which is most difficult part to understand in this coursework. I have researched in different websites and watch lecture video make DTD and Schema completely error free. To compete DTD it took 3 to 4 hours to make it completely error free. Another problem I have faced is styling XML content with CSS. I was confused how to use floating box and design the system. I took reference from another site for design and for some other styles I took reference from w3School and completed the coding full error free.

Conclusion

In conclusion, I have researched and completed the coursework with full dedication. Visual Studio Code is used to build the music store system. At first, it was difficult for me to do DTD and Schema and by taking references from workshop projects and internet I have completed this project. This project has given us confidence in our XML, DTD and Schema coding and research skills. In this project I have tried different types of Schema but Russian Doll is easiest Schema.

This module helped me to understand about XML and use of DTD and Schema in project. Also learned to draw tree diagram of the system. From my point of view, the purpose of this coursework is to learn how to use XML and learn why it is used in real world. I learned that XML is used to provide a software and hardware-independent way of storing, transporting and sharing data. I have also learned how to use CSS in XML document and how to design it properly. Some errors were occurred during developing program and for the correction took references from different websites, lecture slides, watched tutorial video and have a console with friends of the problem overcame during this coursework.

References

- Computer Hope, 2020. *Draw.io*. [Online] Available at: <https://www.computerhope.com/jargon/d/drawio.htm> [Accessed 05 04 2021].
- educative, 2021. *What is Visual studio Code?*. [Online] Available at: <https://www.educative.io/edpresso/what-is-visual-studio-code> [Accessed 04 05 2021].
- GeeksforGeeks, 2020. *Difference between Document Type Definition (DTD) and XML Schema Definition (XSD)*. [Online] Available at: <https://www.geeksforgeeks.org/difference-between-document-type-definition-dtd-and-xml-schema-definition-xsd/> [Accessed 28 04 2021].
- Guru99, 2021. *What is Software Testing? Definition, Basics & Types*. [Online] Available at: <https://www.guru99.com/software-testing-introduction-importance.html> [Accessed 28 04 2021].
- Microsoft, 2021. *Why did we build Visual Studio Code?*. [Online] Available at: <https://code.visualstudio.com/docs/editor/whyvscode> [Accessed 04 05 2021].
- Roche, E., 2000. *What is XML?*. [Online] Available at: <https://hbr.org/2000/07/explaining-xml> [Accessed 04 05 2021].
- tutorialspoint, 2021. *What is CSS?*. [Online] Available at: https://www.tutorialspoint.com/css/what_is_css.htm [Accessed 04 05 2021].
- tutorialspoint, 2021. *XML - DTDs*. [Online] Available at: https://www.tutorialspoint.com/xml/xml_dtds.htm [Accessed 04 05 2021].
- tutorialspoint, 2021. *XML - Overview*. [Online] Available at: https://www.tutorialspoint.com/xml/xml_overview.htm [Accessed 28 04 2021].

tutorialspoint, 2021. *XML - Overview.* [Online]
Available at: https://www.tutorialspoint.com/xml/xml_overview.htm
[Accessed 04 05 2021].

tutorialspoint, 2021. *XML - Schemas.* [Online]
Available at: https://www.tutorialspoint.com/xml/xml_schemas.htm
[Accessed 04 05 2021].