



islington college  
(इस्लिङ्टन कलेज)

**Module Code & Module Title**

**CS5004NI Emerging Programming Platforms and Technologies**

**Assessment Weightage & Type**

**30% Individual Coursework**

**Year and Semester**

**2020-21 Autumn**

**Student Name: Yam Narayan Basnet**

**London Met ID: 19031841**

**College ID: NP01CP4A190096**

**Assignment Due Date: 7<sup>th</sup> May 2021**

**Assignment Submission Date: 7<sup>th</sup> May 2021**

*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

1. Introduction .....	1
1.1 XML .....	1
1.2 XSD .....	1
1.3 DTD .....	1
1.4 CSS .....	1
2. XML Content .....	3
2.1 Tree Diagram .....	3
2.2 XML Insights .....	4
3. Schema Content .....	10
4. DTD Content .....	13
5. Testing .....	15
i. Test 1.....	15
Test 1.1 .....	15
Test 1.2 .....	16
Test 1.3 .....	18
ii. Test 2.....	19
iii. Test 3 .....	21
iv. Test 4 .....	23
v. Test 5.....	25
6. Development of the Coursework.....	26
7. Difference between DTD and XSD .....	30
8. Critical Evaluation .....	32
9. Conclusion .....	38
10. References .....	39

11. Appendix .....	40
--------------------	----

## List of Tables

Table 1 Test 1.1 .....	16
Table 2 Test 1.2 .....	17
Table 3 Test 1.3 .....	18
Table 4: Test 2 .....	19
Table 5:Test 3 .....	21
Table 6 Test 4 .....	23
Table 7 Test 5 .....	25
Table 8 Difference between DTD and XSD .....	31

## List of Figures

Figure 1 Tree Diagram of YamlInc.....	3
Figure 2: Test A   Testing the formatting of XML file .....	15
Figure 3: Test B   Testing the formatting of DTD file .....	16
Figure 4: Test C   Testing the formatting of external schema (XSD) file .....	18
Figure 5: Test 1.3   result of XML, DTD and XSD file formatting.....	19
Figure 6:Test 2   Before removing Group element.....	20
Figure 7:Test 2   After removing Group element.....	20
Figure 8:Test 3   XML file without CSS .....	21
Figure 9: Test 3   XML file displayed in browser without CSS.....	22
Figure 10: Test 3   External CSS file linked in the XML file.....	22
Figure 11: Test 3   browser rendering CSS and its properties when XML file is shown.	23
Figure 12: Task 4   Before hovering .....	24
Figure 13: Task 4   After hovering the Song Title.....	24
Figure 14 Test 5 .....	25
Figure 15 VS Code workspace .....	26
Figure 16 Figma workspace .....	27
Figure 17 draw.io workspace .....	28
Figure 18 Validator requesting for files. ....	29

Figure 19 Validated successfully without any errors.....	29
Figure 21 Critical Evaluation 1.1 .....	32
Figure 22 Critical Evaluation 1.2 .....	32
Figure 23 Critical Evaluation 1.3 .....	33
Figure 24 Critical Evaluation 1.4 .....	33
Figure 25 Critical Evaluation 1.5 .....	33
Figure 26 Critical Evaluation 2.1 .....	34
Figure 27 Critical Evaluation 2.2 .....	34
Figure 28 Critical Evaluation 2.3 .....	35
Figure 29 Critical Evaluation 2.4 .....	35
Figure 30 Critical Evaluation 2.5 .....	35
Figure 31 Critical Evaluation 3.1 .....	36
Figure 32 Critical Evaluation 3.2 .....	36
Figure 33 Critical Evaluation 3.3 .....	37
Figure 34 Critical Evaluation 3.4 .....	37
Figure 35 Critical Evaluation 3.5 .....	37

## 1. Introduction

According to the coursework, a system based on XML for a music store that sells music is required. The music store is named as YamlInc. YamlInc is a music store that has been selling songs since 90s. It is located in Baluwatar, Kathmandu and has been providing service for thousands and millions of happy customers.

### 1.1 XML

XML stands for Extensible Markup Language and is a text-based markup language which is derived from Standard Generalized Markup Language (tutorialspoint, 2021) . XML was initially designed to store and transport data where it also secured its place on being designed as self-descriptive (w3schools, 2021). It is a World Wide Web Consortium (W3C) recommendation.

### 1.2 XSD

XSD (XML Schema Definition) is a World Wide Web Consortium recommendation that specifies how to formally describe the elements present in an XML (Extensible Markup Language) document. XSD can also be used for constructing XML documents that can be treated as programming objects. (Lawton, 2021). It doesn't require parser to process and specifically checks for the correctness of the structure of XML file.

### 1.3 DTD

DTD stands for Document Type Definition and it defines the structure and legal attributes and elements of an XML document. With the help of DTD, independent groups of people can agree on a common standard DTD for exchanging data. An application can use a DTD to verify whether the XML data is valid or not. (w3schools, 2021). It can be specified within a document or outside a document. It primarily checks the grammar and validity of an XML document.

### 1.4 CSS

CSS stands for Cascading Style Sheets and is a stylesheet language which is used to describe or to enhance the presentation of a document written in XML or HTML. CSS describes how elements should be rendered on screen, on paper, or on other media. In

simple words, CSS is used to make an XML or HTML document attractive and presentable. (Docs, 2021)

## 2. XML Content

### 2.1 Tree Diagram

The tree diagram or a tree structure is often referred to as XML Tree and plays a vital role to describe an XML document effortlessly. The tree structure contains a root element i.e. parent elements, child elements and so on. We can get to know the progressive branches and sub-branches initiated from the root. The parsing starts at the root then moves down the branch and so on to the leaf nodes. (tutorialspoint, 2021)

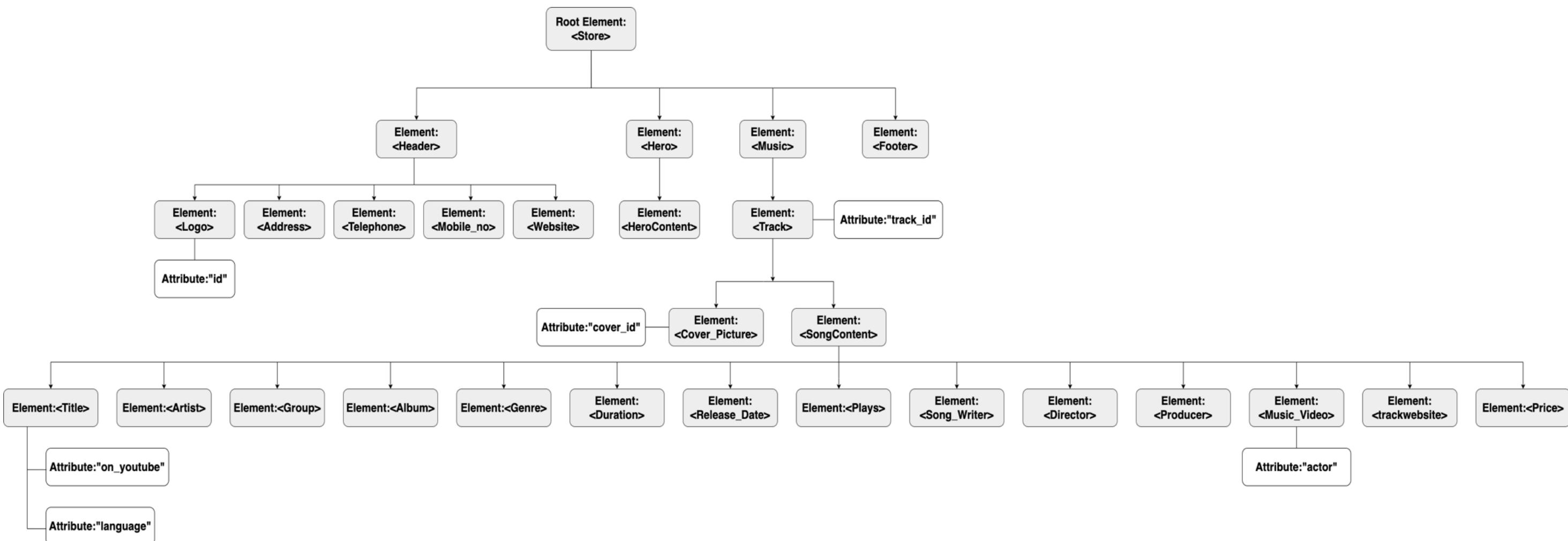


Figure 1 Tree Diagram of YamInc

As we can see the above tree diagram of YamInc Music Store, the root element is Store and it branches out four different sub-branches and so on. The visualization of the tree helps to easily understand the XML. Here in the tree diagram the parents extend their children and the elements and attributes are well presented to the place where they belong.

## 2.2 XML Insights

```

3. <?xml version="1.0" encoding="UTF-8"?>
4. <?xml-stylesheet href="catalog_19031841.css" type="text/css"?>
5. <!DOCTYPE Store SYSTEM "./catalog_19031841.dtd">
6.
7. <!--
8.     Coursework 2
9.     XML
10.    Author: Yam Narayan Basnet
11.    Date: 2nd May 2021
12.    Filename: catalog_19031841.xml
13. -->
14. <!-- Store is the root element -->
15. <Store xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="catalog_19031841.xsd">
16.
17.     <!-- Header consists of logo and store information -->
18.     <Header>
19.         <Logo id="Img"/>
20.         <Address>Baluwatar, Kathmandu</Address>
21.         <Telephone>01-4018293</Telephone>
22.         <Mobile_no>9813245686</Mobile_no>
23.         <Website>www.yaminmusic.com</Website>
24.     </Header>
25.     <!-- Hero element consists of a tag line -->
26.     <Hero>
27.         <HeroContent>Blessing your ears since the 90s</HeroContent>
28.     </Hero>
29.     <!-- Music element consists of Track and Song information -->
30.     <Music>
31.         <Track track_id="T1">
32.             <Cover_Picture cover_id="Cover_1"/>
33.             <!-- Song content consists of the information of the song -->
34.             <SongContent>
35.                 <Title on_youtube="Yes" language="English">Bohemian
   Rhapsody</Title>
36.                 <Artist>Artist: Freddie Mercury</Artist>
37.                 <Group>Group: Queen</Group>
38.                 <Album>Album: Live Killers</Album>
39.                 <Genre>Genre: Rock</Genre>
40.                 <Duration>Duration: 5:55</Duration>
41.                 <Release_Date>Released Year: 1975</Release_Date>
42.                 <Plays>Plays: 1,483,289,766</Plays>
43.                 <Song_Writer>Song Writer: Freddie Mercury</Song_Writer>
44.                 <Producer>Producer: Roy Thomas Baker</Producer>
45.                 <Music_Video>Music Video: Yes</Music_Video>
46.                 <trackwebsite>Website: https://www.queenonline.com/</trackwebsite>
```

```

47.          <Price>$12.99</Price>
48.      </SongContent>
49.  </Track>
50.  <Track track_id="T2">
51.      <Cover_Picture cover_id="Cover_2"/>
52.      <SongContent>
53.          <Title on_youtube="Yes" language="English">Heather</Title>
54.          <Artist>Artist: Conan Grey</Artist>
55.          <Album>Album: Kid Crow</Album>
56.          <Genre>Genre: Folk</Genre>
57.          <Duration>Duration: 3:18</Duration>
58.          <Release_Date>Released Year: 2020</Release_Date>
59.          <Plays>Plays: 583,789,166</Plays>
60.          <Song_Writer>Song Writer: Conan Grey</Song_Writer>
61.          <Producer>Producer: Daniel Nigro</Producer>
62.          <Music_Video>Music Video: Yes</Music_Video>
63.          <Price>$10.99</Price>
64.      </SongContent>
65.  </Track>
66.  <Track track_id="T3">
67.      <Cover_Picture cover_id="Cover_3"/>
68.      <SongContent>
69.          <Title on_youtube="Yes" language="English">Driver's license</Title>
70.          <Artist>Artist: Olivia Rodrigo</Artist>
71.          <Album>Album: Driver's license</Album>
72.          <Genre>Genre: Pop</Genre>
73.          <Duration>Duration: 4:02</Duration>
74.          <Release_Date>Released Year: 2021</Release_Date>
75.          <Plays>Plays: 239,989,766</Plays>
76.          <Song_Writer>Song Writer: Olivia Rodrigo <! [CDATA[ & ]]> Daniel
    Nigro</Song_Writer>
77.          <Producer>Producer: Daniel Nigro</Producer>
78.          <Music_Video actor ="Olivia Rodrigo">Music Video: Yes</Music_Video>
79.          <Price>$8.99</Price>
80.      </SongContent>
81.  </Track>
82.  <Track track_id="T4">
83.      <Cover_Picture cover_id="Cover_4"/>
84.      <SongContent>
85.          <Title language="English">Everything I wanted</Title>
86.          <Artist>Artist: Billie Eillish</Artist>
87.          <Album>Album: Everything I wanted</Album>
88.          <Genre>Genre: Pop</Genre>
89.          <Duration>Duration: 4:05</Duration>
90.          <Release_Date>Released Year: 2019</Release_Date>
91.          <Plays>Plays: 2,193,659,195</Plays>
92.          <Song_Writer>Song Writer: Billie Eillish</Song_Writer>
93.          <Music_Video actor="Billie Eillish">Music Video: Yes</Music_Video>

```

```

94.          <Price>$9.99</Price>
95.          </SongContent>
96.      </Track>
97.      <Track track_id="T5">
98.          <Cover_Picture cover_id="Cover_5"/>
99.          <SongContent>
100.              <Title language="English">Pillow Talk</Title>
101.              <Artist>Artist: Zayn Malik</Artist>
102.              <Album>Album: Mind Of Mine</Album>
103.              <Genre>Genre: RnB</Genre>
104.              <Duration>Duration: 3:23</Duration>
105.              <Release_Date>Released Year: 2016</Release_Date>
106.              <Plays>Plays: 1,893,679,206</Plays>
107.              <Song_Writer>Song Writer: Zayn Malik</Song_Writer>
108.              <Producer>Producer: Levi Lennon</Producer>
109.              <Music_Video actor= "Gigi Hadid">Music Video:
    Yes</Music_Video>
110.          <Price>$15.99</Price>
111.          </SongContent>
112.
113.      </Track>
114.      <Track track_id="T6">
115.          <Cover_Picture cover_id="Cover_6"/>
116.          <SongContent>
117.              <Title language="Hindi">Vaaste</Title>
118.              <Artist>Artist: Dhvani Bhanushali <![CDATA[ & ]]> Nikhil D'
    Souza</Artist>
119.              <Genre>Genre: Pop</Genre>
120.              <Duration>Duration: 4:06</Duration>
121.              <Release_Date>Released Year: 2019</Release_Date>
122.              <Plays>Plays: 988,189,155</Plays>
123.              <Song_Writer>Song Writer: Arafar Mehmood</Song_Writer>
124.              <Producer>Producer: Bhusan Kumar</Producer>
125.              <Music_Video actor= "Dhvani Bhanushali">Music Video:
    Yes</Music_Video>
126.          <Price>$5.99</Price>
127.          </SongContent>
128.      </Track>
129.      <Track track_id="T7">
130.          <Cover_Picture cover_id="Cover_7"/>
131.          <SongContent>
132.              <Title on_youtube="Yes" language="Hindi">Pachtaoge</Title>
133.              <Artist>Artist: Arijit Singh</Artist>
134.              <Album>Album: Pachtaoge</Album>
135.              <Genre>Genre: Pop</Genre>
136.              <Duration>Duration: 4:30</Duration>
137.              <Release_Date>Released Year: 2019</Release_Date>
138.              <Plays>Plays: 2,623,189,466</Plays>

```

```

139.          <Song_Writer>Song Writer : Jaani</Song_Writer>
140.          <Music_Video>Music Video: Yes</Music_Video>
141.          <trackwebsite>Website:
142.              https://twitter.com/TheArijitSingh</trackwebsite>
143.          <Price>$4.99</Price>
144.          </SongContent>
145.          </Track>
146.          <Track track_id="T8">
147.              <Cover_Picture cover_id="Cover_8"/>
148.              <SongContent>
149.                  <Title language="English">The Lady in Red</Title>
150.                  <Artist>Artist: Chris de Burgh</Artist>
151.                  <Album>Album: Into the Light</Album>
152.                  <Genre>Genre: Soft rock</Genre>
153.                  <Duration>Duration: 4:17</Duration>
154.                  <Release_Date>Released Year: 1986</Release_Date>
155.                  <Plays>Plays: 2,203,489,266</Plays>
156.                  <Song_Writer>Song Writer: Chris de Burgh</Song_Writer>
157.                  <Producer>Producer: Paul Hardiman</Producer>
158.                  <Music_Video>Music Video: No</Music_Video>
159.                  <Price>$22.99</Price>
160.                  </SongContent>
161.          </Track>
162.          <Track track_id="T9">
163.              <Cover_Picture cover_id="Cover_9"/>
164.              <SongContent>
165.                  <Title language="English">2002</Title>
166.                  <Artist>Artist: Anne Marie</Artist>
167.                  <Album>Album: Speak Your Mind</Album>
168.                  <Genre>Genre: Pop</Genre>
169.                  <Duration>Duration: 3:07</Duration>
170.                  <Release_Date>Released Year: 2018</Release_Date>
171.                  <Plays>Plays: 623,189,466</Plays>
172.                  <Song_Writer>Song Writer: Anne Marie</Song_Writer>
173.                  <Producer>Producer: Steve Mac</Producer>
174.                  <Music_Video>Music Video: Yes</Music_Video>
175.                  <Price>$11.99</Price>
176.                  </SongContent>
177.          </Track>
178.          <Track track_id="T10">
179.              <Cover_Picture cover_id="Cover_10"/>
180.              <SongContent>
181.                  <Title language="English">Shape of You</Title>
182.                  <Artist>Artist: Ed Sheeran</Artist>
183.                  <Album>Album: %</Album>
184.                  <Genre>Genre: Pop</Genre>
185.                  <Duration>Duration: 3:53</Duration>
186.                  <Release_Date>Released Year: 2017</Release_Date>

```

```

186.          <Plays>Plays: 3,523,289,566</Plays>
187.          <Song_Writer>Song Writer: Ed Sheeran</Song_Writer>
188.          <Producer>Producer: Steve Mac</Producer>
189.          <Music_Video>Music Video: Yes</Music_Video>
190.          <Price>$10.99</Price>
191.          </SongContent>
192.      </Track>
193.      <Track track_id="T11">
194.          <Cover_Picture cover_id="Cover_11"/>
195.          <SongContent>
196.              <Title language="English">Hey Jude</Title>
197.              <Artist>Artist: Paul McCartney</Artist>
198.              <Group>Group: The Beatles</Group>
199.              <Album>Album: Hey Jude</Album>
200.              <Genre>Genre: Pop rock</Genre>
201.              <Duration>Duration: 7:11</Duration>
202.              <Release_Date>Released Year: 1968</Release_Date>
203.              <Plays>Plays: 2,223,279,766</Plays>
204.              <Song_Writer>Song Writer: Lennon–McCartney</Song_Writer>
205.              <Producer>Producer: George Martin</Producer>
206.              <Music_Video>Music Video: Yes</Music_Video>
207.              <Price>$8.99</Price>
208.              </SongContent>
209.          </Track>
210.          <Track track_id="T12">
211.              <Cover_Picture cover_id="Cover_12"/>
212.              <SongContent>
213.                  <Title language="English">Cheerleader</Title>
214.                  <Artist>Artist: Omi</Artist>
215.                  <Album>Album: Me 4 U</Album>
216.                  <Genre>Genre: Reggae fusion</Genre>
217.                  <Duration>Duration: 2:58</Duration>
218.                  <Release_Date>Released Year: 2012</Release_Date>
219.                  <Plays>Plays: 303,989,866</Plays>
220.                  <Song_Writer>Song Writer: Omar Samuel Pasley</Song_Writer>
221.                  <Producer>Producer: Feliz Jaehn</Producer>
222.                  <Music_Video>Music Video: Yes</Music_Video>
223.                  <Price>$7.99</Price>
224.                  </SongContent>
225.          </Track>
226.          <Track track_id="T13">
227.              <Cover_Picture cover_id="Cover_13"/>
228.              <SongContent>
229.                  <Title language="English">Someone You Loved</Title>
230.                  <Artist>Artist: Lewis Capaldi</Artist>
231.                  <Album>Album: Divinely Uninspired to a Hellish Extent</Album>
232.                  <Genre>Genre: Pop</Genre>
233.                  <Duration>Duration: 3:02</Duration>

```

```

234.          <Release_Date>Released Year: 2012</Release_Date>
235.          <Plays>Plays: 1,203,589,886</Plays>
236.          <Song_Writer>Song Writer: Lewis Capaldi</Song_Writer>
237.          <Producer>Producer: TMS</Producer>
238.          <Music_Video>Music Video: Yes</Music_Video>
239.          <Price>$14.99</Price>
240.          </SongContent>
241.      </Track>
242.      <Track track_id="T14">
243.          <Cover_Picture cover_id="Cover_14"/>
244.          <SongContent>
245.              <Title language="English">All I Want</Title>
246.              <Artist>Artist: Steve Garrigan</Artist>
247.              <Group>Group: Kodaline</Group>
248.              <Album>Album: In a Perfect World</Album>
249.              <Genre>Genre: Rock</Genre>
250.              <Duration>Duration: 3:58</Duration>
251.              <Release_Date>Released Year: 2013</Release_Date>
252.              <Plays>Plays: 1,202,979,966</Plays>
253.              <Song_Writer>Song Writer: James Flannigan</Song_Writer>
254.              <Director>Stevie Russell</Director>
255.              <Producer>Producer Stephan Harris</Producer>
256.              <Music_Video actor="Bryan Hussel">Music_Video:
    Yes</Music_Video>
257.          <Price>$9.99</Price>
258.          </SongContent>
259.      </Track>
260.      <Track track_id="T15">
261.          <Cover_Picture cover_id="Cover_15"/>
262.          <SongContent>
263.              <Title language="English">Changes</Title>
264.              <Artist>Artist: XXXTentacion</Artist>
265.              <Genre>Genre: RnB</Genre>
266.              <Duration>Duration: 2:02</Duration>
267.              <Release_Date>Released Year: 2018</Release_Date>
268.              <Plays>Plays: 903,689,766</Plays>
269.              <Song_Writer>Song Writer: Jahseh Onfroy</Song_Writer>
270.              <Producer>Producer: Cunningham</Producer>
271.              <Music_Video>Music Video: No</Music_Video>
272.              <Price>$7.99</Price>
273.              </SongContent>
274.          </Track>
275.      </Music>
276.      <!-- Footer of the site -->
277.      <Footer>YamInc Pvt. Ltd. | Copyright &#xA9; 2021</Footer>
278.  </Store>

```

### 3. Schema Content

```
4. <?xml version="1.0" encoding="UTF-8"?>
5. <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
   elementFormDefault="qualified">
6.   <xs:element name="Store">
7.     <xs:complexType>
8.       <xs:sequence>
9.         <xs:element ref="Header"/>
10.        <xs:element ref="Hero"/>
11.        <xs:element ref="Music"/>
12.        <xs:element ref="Footer"/>
13.       </xs:sequence>
14.     </xs:complexType>
15.   </xs:element>
16.   <xs:element name="Header">
17.     <xs:complexType>
18.       <xs:sequence>
19.         <xs:element ref="Logo"/>
20.         <xs:element ref="Address"/>
21.         <xs:element ref="Telephone"/>
22.         <xs:element ref="Mobile_no"/>
23.         <xs:element ref="Website"/>
24.       </xs:sequence>
25.     </xs:complexType>
26.   </xs:element>
27.   <xs:element name="Logo">
28.     <xs:complexType>
29.       <xs:attribute name="id" use="required" type="xs:NCName"/>
30.     </xs:complexType>
31.   </xs:element>
32.   <xs:element name="Address" type="xs:string"/>
33.   <xs:element name="Telephone" type="xs:NMTOKEN"/>
34.   <xs:element name="Mobile_no" type="xs:integer"/>
35.   <xs:element name="Website" type="xs:NCName"/>
36.   <xs:element name="Hero">
37.     <xs:complexType>
38.       <xs:sequence>
39.         <xs:element ref="HeroContent"/>
40.       </xs:sequence>
41.     </xs:complexType>
42.   </xs:element>
43.   <xs:element name="HeroContent" type="xs:string"/>
44.   <xs:element name="Music">
45.     <xs:complexType>
46.       <xs:sequence>
47.         <xs:element maxOccurs="unbounded" ref="Track"/>
48.       </xs:sequence>
```

```

49.      </xs:complexType>
50.    </xs:element>
51.    <xs:element name="Track">
52.      <xs:complexType>
53.        <xs:sequence>
54.          <xs:element ref="Cover_Picture"/>
55.          <xs:element ref="SongContent"/>
56.        </xs:sequence>
57.        <xs:attribute name="track_id" use="required" type="xs:NCName"/>
58.      </xs:complexType>
59.    </xs:element>
60.    <xs:element name="Cover_Picture">
61.      <xs:complexType>
62.        <xs:attribute name="cover_id" use="required" type="xs:NCName"/>
63.      </xs:complexType>
64.    </xs:element>
65.    <xs:element name="SongContent">
66.      <xs:complexType>
67.        <xs:sequence>
68.          <xs:element ref="Title"/>
69.          <xs:element ref="Artist"/>
70.          <xs:element minOccurs="0" ref="Group"/>
71.          <xs:element minOccurs="0" ref="Album"/>
72.          <xs:element ref="Genre"/>
73.          <xs:element ref="Duration"/>
74.          <xs:element ref="Release_Date"/>
75.          <xs:element ref="Plays"/>
76.          <xs:element ref="Song_Writer"/>
77.          <xs:element minOccurs="0" ref="Director"/>
78.          <xs:element minOccurs="0" ref="Producer"/>
79.          <xs:element ref="Music_Video"/>
80.          <xs:element minOccurs="0" ref="trackwebsite"/>
81.          <xs:element ref="Price"/>
82.        </xs:sequence>
83.      </xs:complexType>
84.    </xs:element>
85.    <xs:element name="Title">
86.      <xs:complexType mixed="true">
87.        <xs:attribute name="language" use="required" type="xs:NCName"/>
88.        <xs:attribute name="on_youtube" type="xs:NCName"/>
89.      </xs:complexType>
90.    </xs:element>
91.    <xs:element name="Artist" type="xs:string"/>
92.    <xs:element name="Group" type="xs:string"/>
93.    <xs:element name="Album" type="xs:string"/>
94.    <xs:element name="Genre" type="xs:string"/>
95.    <xs:element name="Duration" type="xs:string"/>
96.    <xs:element name="Release_Date" type="xs:string"/>

```

```
97.  <xs:element name="Plays" type="xs:string"/>
98.  <xs:element name="Song_Writer" type="xs:string"/>
99.  <xs:element name="Director" type="xs:string"/>
100.   <xs:element name="Producer" type="xs:string"/>
101.   <xs:element name="Music_Video">
102.     <xs:complexType mixed="true">
103.       <xs:attribute name="actor"/>
104.     </xs:complexType>
105.   </xs:element>
106.   <xs:element name="trackwebsite" type="xs:string"/>
107.   <xs:element name="Price" type="xs:string"/>
108.   <xs:element name="Footer" type="xs:string"/>
109. </xs:schema>
```

## 4. DTD Content

```

5. <!--
6.     Coursework 2
7.     XML
8.     Author: Yam Narayan Basnet
9.     Date: 2nd May 2021
10.    Filename: catalog_19031841.xml
11.   -->
12.
13.<!ELEMENT Store (Header, Hero, Music, Footer)>
14.<!ELEMENT Header (Logo, Address, Telephone, Mobile_no, Website)>
15.<!ELEMENT Logo EMPTY>
16.<!ELEMENT Address (#PCDATA)>
17.<!ELEMENT Telephone (#PCDATA)>
18.<!ELEMENT Mobile_no (#PCDATA)>
19.<!ELEMENT Website (#PCDATA)>
20.<!ELEMENT Hero (HeroContent)>
21.<!ELEMENT HeroContent (#PCDATA)>
22.<!ELEMENT Music (Track+)>
23.<!ELEMENT Track (Cover_Picture, SongContent)>
24.<!ELEMENT SongContent (Title, Artist, Group?, Album?, Genre,
25.                               Duration, Release_Date, Plays, Song_Writer, Director?,
26.                               Producer?, Music_Video,trackwebsite?, Price)>
27.<!ELEMENT Cover_Picture (#PCDATA)>
28.<!ELEMENT Title (#PCDATA)>
29.<!ELEMENT Artist (#PCDATA)>
30.<!ELEMENT Group (#PCDATA)>
31.<!ELEMENT Album (#PCDATA)>
32.<!ELEMENT Genre (#PCDATA)>
33.<!ELEMENT Duration (#PCDATA)>
34.<!ELEMENT Release_Date (#PCDATA)>
35.<!ELEMENT Plays (#PCDATA)>
36.<!ELEMENT Song_Writer (#PCDATA)>
37.<!ELEMENT Director (#PCDATA)>
38.<!ELEMENT Producer (#PCDATA)>
39.<!ELEMENT Music_Video (#PCDATA)>
40.<!ELEMENT Price (#PCDATA)>
41.<!ELEMENT trackwebsite (#PCDATA)>
42.<!ELEMENT Footer (#PCDATA)>
43.
44.<!ATTLIST Store
45. xmlns:xsi CDATA #REQUIRED
46. xsi:noNamespaceSchemaLocation CDATA #REQUIRED
47.>
48.<!ATTLIST Logo id CDATA #REQUIRED>
49.<!ATTLIST Track track_id ID #REQUIRED>
50.<!ATTLIST Cover_Picture cover_id ID #REQUIRED>
```

```
51.<!ATTLIST Title on_youtube CDATA #IMPLIED language CDATA #REQUIRED>
52.<!ATTLIST Music_Video actor CDATA #IMPLIED>
```

## 5. Testing

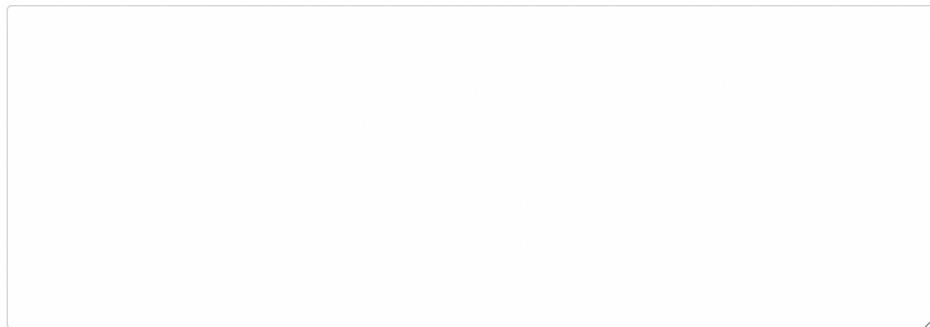
In the testing process format of XML, DTD and Schema (XSD) files are validated through the use of online validator website named as <http://www.xmlvalidation.com> and checked whether XML and CSS files are rendered in the browser or not.

### i. Test 1

**Objective: To Check if XML, DTD and XSD are properly formatted and validated or not.**

#### Test 1.1

**Please copy your XML document in here:**



Or upload it:

Choose File No file chosen

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**

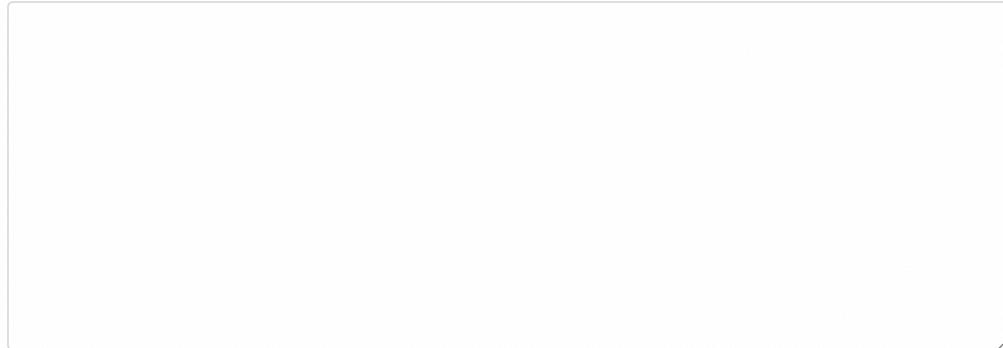
Figure 2: Test A | Testing the formatting of XML file

<b>Test No.</b>	1.1
<b>Action</b>	Uploading XML file in the validator website.
<b>Expected Result</b>	Website is expected to ask for DTD file that was referenced in XML.
<b>Actual Result</b>	Website asks for the referenced DTD file.
<b>Conclusion</b>	The test was Successful.

Table 1 Test 1.1

## Test 1.2

The file catalog\_19031841.dtd is being referenced. Please copy it in here, so that the validation can continue:



Or upload it:

No file chosen

The following files have been uploaded so far:

[XML document:](#) 

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

Figure 3: Test B | Testing the formatting of DTD file

<b>Test No.</b>	<b>1.2</b>
<b>Action</b>	DTD file is being uploaded on the website for validation of its format.
<b>Expected Result</b>	Website asks for Schema referenced in XML since, “Validate against external XML is checked” in Task 1.1.
<b>Actual Result</b>	Website asks for Schema referenced in XML
<b>Conclusion</b>	The test was Successful.

Table 2 Test 1.2

**Test 1.3**

Please copy the XML schema in here:

Or upload it:

Choose File No file chosen

**continue validation**

The following files have been uploaded so far:

XML document: [∅](#)

[catalog\\_19031841.dtd](#) [∅](#)

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

Figure 4: Test C | Testing the formatting of external schema (XSD) file

<b>Test No.</b>	1.3
<b>Action</b>	Uploading external Schema (XSD) file in the website.
<b>Expected Result</b>	Website would display “no errors were found” in the formatting of XML, DTD and XSD files.
<b>Actual Result</b>	Website displays “no errors were found”
<b>Conclusion</b>	The test was Successful.

Table 3 Test 1.3

**No errors were found**

The following files have been uploaded so far:

XML document: [catalog.xml](#)  
XML schema: [catalog\\_19031841.xsd](#)  
[catalog\\_19031841.dtd](#)

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

Figure 5: Test 1.3 | result of XML, DTD and XSD file formatting

**ii. Test 2**

**Objective: To check whether the optional element in xml file would work or not.**

Table 4: Test 2

<b>Test No.</b>	2
<b>Action</b>	Removing optional element, Album from the XML file and check whether it would show any error or not
<b>Expected Result</b>	After performing following action error shouldn't be shown
<b>Actual Result</b>	Error was not shown in the text editor as well as browser.

<b>Conclusion</b>	The test was Successful.
-------------------	--------------------------

```
<Track track_id="T1">
    <Cover_Picture cover_id="Cover_1"/>
    <SongContent>
        <Title on_youtube="Yes" language="English">Bohemian Rhapsody</Title>
        <Artist>Artist: Freddie Mercury</Artist>
        <Group>Group: Queen</Group>
        <Album>Album: Live Killers</Album>
        <Genre>Genre: Rock</Genre>
```

Figure 6:Test 2 | Before removing Group element

```
<Track track_id="T1">
    <Cover_Picture cover_id="Cover_1"/>
    <SongContent>
        <Title on_youtube="Yes" language="English">Bohemian Rhapsody</Title>
        <Artist>Artist: Freddie Mercury</Artist>
        <Album>Album: Live Killers</Album>
        <Genre>Genre: Rock</Genre>
        <Duration>Duration: 5:55</Duration>
```

Figure 7:Test 2 | After removing Group element

### iii. Test 3

**Objective:** To check whether the CSS file is working or not.

Table 5:Test 3

<b>Test No.</b>	3
<b>Action</b>	XML file is opened in the browser without linking CSS file and vice versa.
<b>Expected Result</b>	After performing following action CSS design must be seen after linking.
<b>Actual Result</b>	CSS design was rendered in the browser
<b>Conclusion</b>	The test was Successful.

```
<?xml version="1.0" encoding="UTF-8"?>
|
<!DOCTYPE Store SYSTEM "./catalog_19031841.dtd">

<!--
    Coursework 2
    XML
    Author: Yam Narayan Basnet
    Date: 2nd May 2021
    Filename: catalog_19031841.xml
-->
```

Figure 8:Test 3 | XML file without CSS

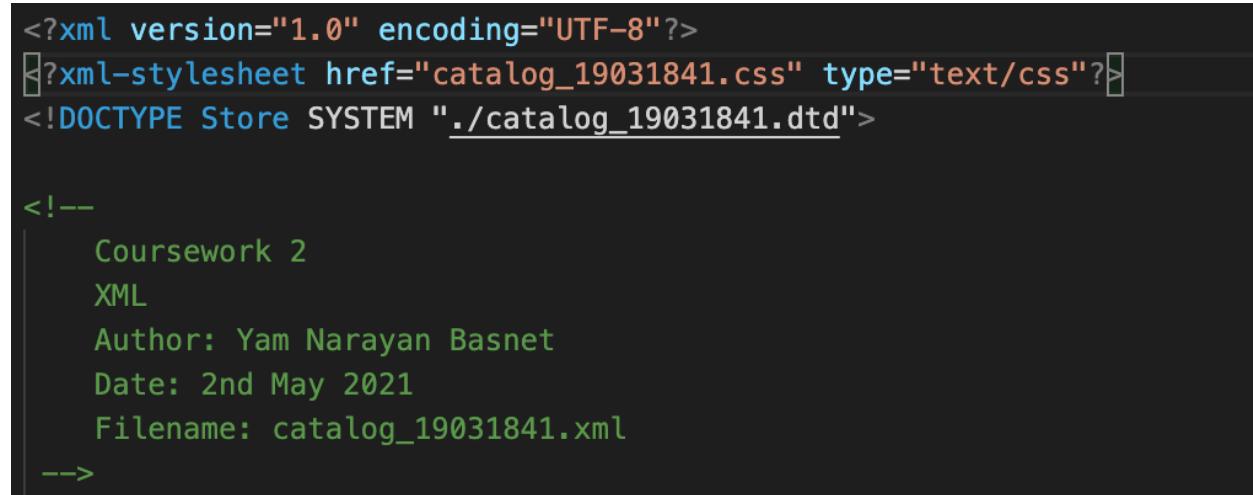
---

This XML file does not appear to have any style information associated with it. The document tree is shown below.

---

```
<!-- <?xml-stylesheet href="catalog_19031841.css" type="text/css"?> -->
<!--
Coursework 2
XML
Author: Yam Narayan Basnet
Date: 2nd May 2021
Filename: catalog_19031841.xml
-->
<!-- Store is the root element -->
<Store xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="catalog_19031841.xsd">
<!-- Header consists of logo and store information -->
<Header>
<Logo id="Img"/>
<Address>Baluwatar, Kathmandu</Address>
<Telephone>01-4018293</Telephone>
<Mobile_no>9813245686</Mobile_no>
<Website>www.yaminmusic.com</Website>
</Header>
<!-- Hero element consists of a tag line -->
<Hero>
<HeroContent>Blessing your ears since the 90s</HeroContent>
</Hero>
<Music>
<Track track_id="T1">
<Cover_Picture cover_id="Cover_1"/>
<SongContent>
<Title on_youtube="Yes" language="English">Bohemian Rhapsody</Title>
<Artist>Artist: Freddie Mercury</Artist>
<Group>Group: Queen</Group>
```

Figure 9: Test 3 | XML file displayed in browser without CSS



```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="catalog_19031841.css" type="text/css"?>
<!DOCTYPE Store SYSTEM "./catalog_19031841.dtd">

<!--
Coursework 2
XML
Author: Yam Narayan Basnet
Date: 2nd May 2021
Filename: catalog_19031841.xml
-->
```

Figure 10: Test 3 | External CSS file linked in the XML file

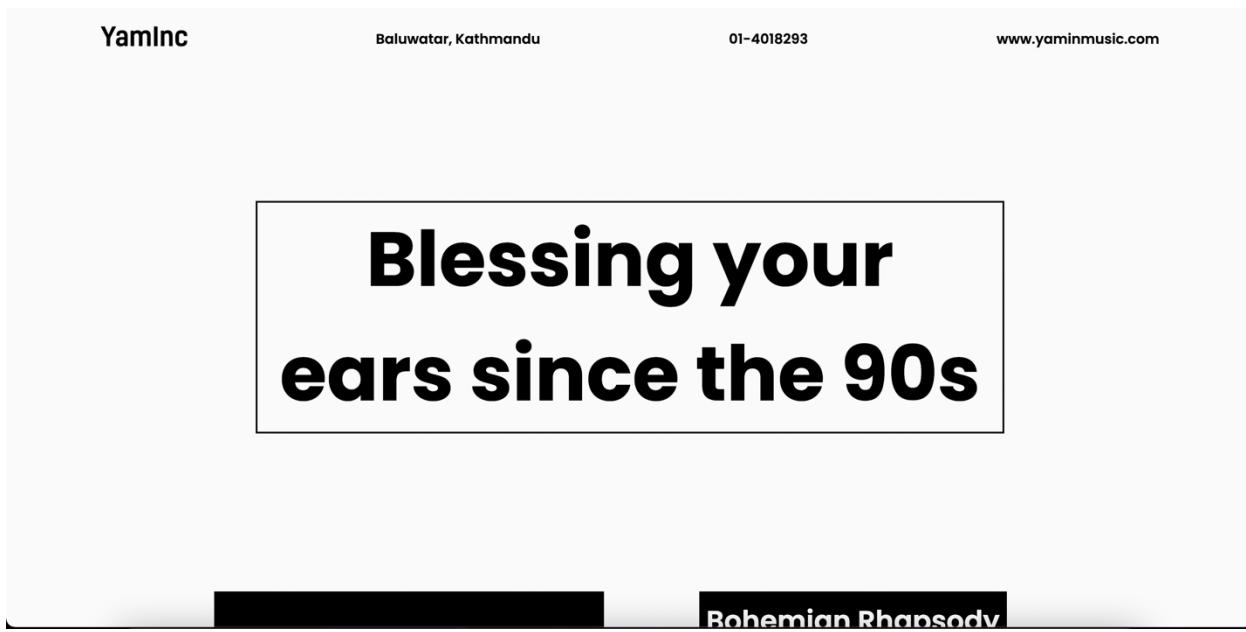


Figure 11: Test 3 | browser rendering CSS and its properties when XML file is shown

#### iv. Test 4

**Objective:** To check whether the :hover effect in CSS working or not.

<b>Test No.</b>	4
<b>Action</b>	XML file is opened in the browser and cursor is hovered around Song Title.
<b>Expected Result</b>	After performing following action Song Title should change its color with an ease transition.
<b>Actual Result</b>	Mentioned hover effects were seen in the browser
<b>Conclusion</b>	The test was Successful.

Table 6 Test 4

**Driver's license**

Artist: Olivia Rodrigo  
Album: Driver's license  
Genre: Pop  
Duration: 4:02  
Released Year: 2021  
Plays: 239,989,766  
Song Writer: Olivia Rodrigo & Daniel Nigro  
Producer: Daniel Nigro  
Music Video: Yes

**\$8.99**

**Everything I wanted**

Artist: Billie Eilish  
Album: Everything I wanted

Figure 12: Task 4 | Before hovering

**Driver's license**

Artist: Olivia Rodrigo  
Album: Driver's license  
Genre: Pop  
Duration: 4:02  
Released Year: 2021  
Plays: 239,989,766  
Song Writer: Olivia Rodrigo & Daniel Nigro  
Producer: Daniel Nigro  
Music Video: Yes

**\$8.99**

**Everything I wanted**

Artist: Billie Eilish  
Album: Everything I wanted

Figure 13: Task 4 | After hovering the Song Title

### v. Test 5

**Objective:** To check

<b>Test No.</b>	5
<b>Action</b>	To check whether XML file references DTD file or not.
<b>Expected Result</b>	DTD file would reference and validate with the XML file.
<b>Actual Result</b>	DTD file is referenced in XML file and validates as shown in Test 1.2 and 1.3
<b>Conclusion</b>	The test was successful

Table 7 Test 5

```
<?xml version="1.0" encoding="UTF-8"?>
<!xmlstylesheet href="catalog_19031841.css" type="text/css"?>
<!DOCTYPE Store SYSTEM "./catalog_19031841.dtd">

<!--
    Coursework 2
    XML
    Author: Yam Narayan Basnet
    Date: 2nd May 2021
    Filename: catalog_19031841.xml
-->
```

Figure 14 Test 5

## 6. Development of the Coursework

The coursework comprises of XML document, DTD, Schema (XSD) and CSS files. Arranging those files based on their similarities, programming on individual files, creating DTD and Schema for the XML document was not an easy job. This is where VS Code (Visual Studio Code) came in handy. Visual Studio Code (VS Code) is a code editor where you can easily arrange files and debug the program. It shows error in a pretty convenient way that makes it very reliable to debug and deploy your application. Some of the key feature of VS Code that came in handy while development of this coursework was the live server where you can go live with your XML file and it refreshes every time you save your file. This feature saved a lot of time and effort. Since I own a m1 powered MacBook, the VS Code is very optimized and smooth for it. Below is the layout of a VS Code where we can see how simple and attractive the workspace would be.

```

catalog_19031841.xml - XML
catalog_19031841.xml > Store
catalog_19031841.xml > Store
catalog_19031841.dtd
catalog_19031841.css
catalog_19031841.xsd

catalog_19031841.xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="catalog_19031841.css" type="text/css"?>
<!DOCTYPE Store SYSTEM "./catalog_19031841.dtd">
<!--
    Coursework_2
    XML
    Author: Yam Narayan Basnet
    Date: 2nd May 2021
    Filename: catalog_19031841.xml
-->
<!-- Store is the root element -->
<Store xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="catalog_19031841.xsd">
    <!-- Header consists of logo and store information -->
    <Header>
        <Logo id="Img"/>
        <Address>Baluwatar, Kathmandu</Address>
        <Telephone>01-4018293</Telephone>
        <Mobile_no>9813245686</Mobile_no>
        <Website>www.yamimusic.com</Website>
    </Header>
    <!-- Hero element consists of a tag line -->
    <Hero>
        <HeroContent>Blessing your ears since the 90s</HeroContent>
    </Hero>
    <!-- Music element consists of Track and Song information -->
    <Music>
        <Track track_id="T1">
            <Cover_Picture cover_id="Cover_1"/>
            <SongContent>
                <Title on_youtube="Yes" language="English">Bohemian Rhapsody</Title>
                <Artist>Artist: Freddie Mercury</Artist>
                <Group>Group: Queen</Group>
                <Album>Album: Live Killers</Album>
                <Genre>Genre: Rock</Genre>
                <Duration>Duration: 5:55</Duration>
                <Release_Date>Released Year: 1975</Release_Date>
            </SongContent>
        </Track>
    </Music>
</Store>

```

Figure 15 VS Code workspace

The other thing that primarily played a vital role in the development of this coursework was ‘Figma’. Figma is digital design and prototyping tool. It is primarily used for UX & UI Design, but it was used for wireframing the idea in this coursework. The XML document of this coursework would be quite common between the students as the questions provided are same but what makes one excel is a beautiful CSS design. Personally, I prefer minimalistic design, so I started building a wireframe in Figma. Black and white was the primary theme color and logo was designed in a free logo maker site. Images were extracted respective to their song covers. Figma was a great help for wireframing the concept of the site. Below is a gist of what actually goes on in Figma.

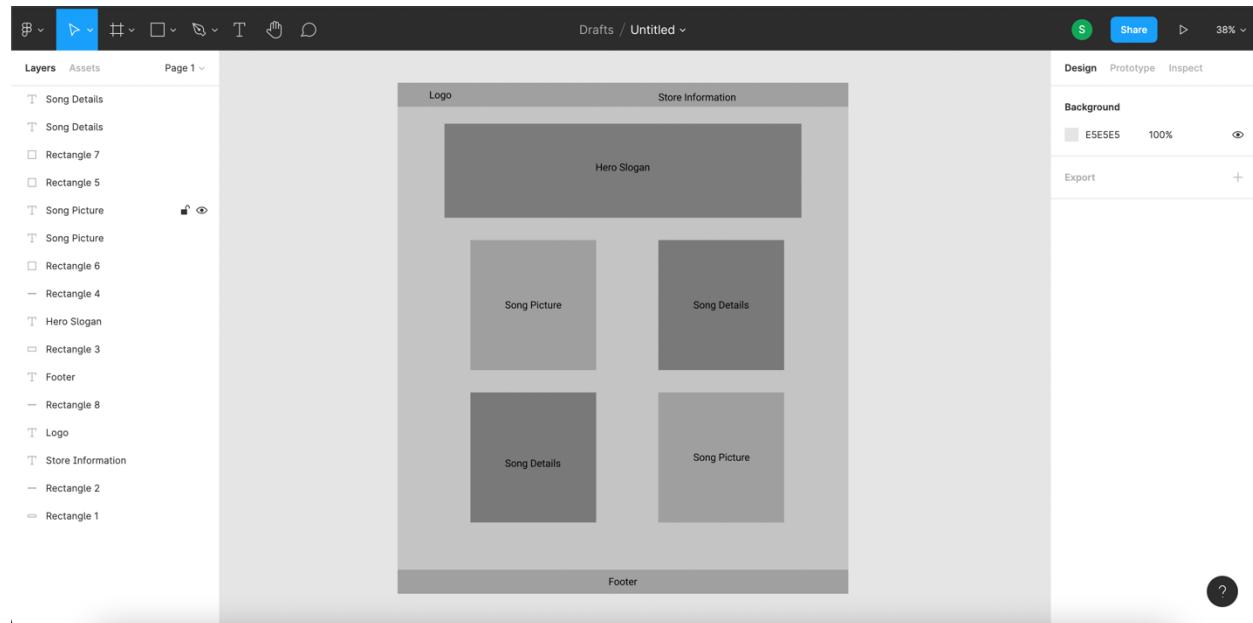


Figure 16 Figma workspace

Besides the programming part of the coursework, for documenting the tree diagram of the XML document, draw.io tool was used. draw.io is a free online diagramming software application for building diagramming applications and also produces web-based diagramming technology and integrated with Google Drive and Dropbox (crunchbase, 2021). draw.io has been a great companion and always a first option when it comes to drawing figures. It is simple, professional and builds elegant looking design with just drag and drop skills. Below is the picture of what actually goes on draw.io.

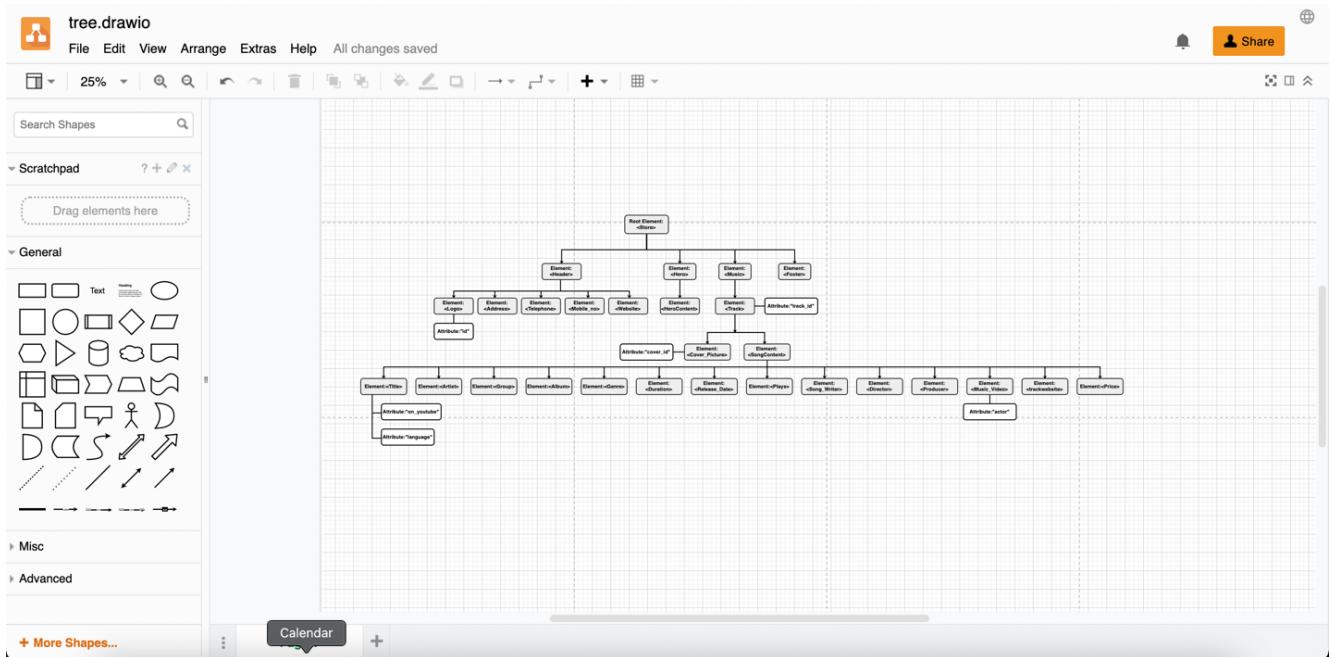


Figure 17 draw.io workspace

Having a revised look on XML Document, it is comprised of XML, DTD, Schema, and CSS where the coursework had assigned to create minimum of 15 songs with their information and a music store selling it. The music store was named YamlInc, and it has a total of 15 songs with their information. At least 5 optional elements and attributes are included that makes a total of 14 Song elements i.e., Song information. Post completion of the program, the document was validated using an online site, <http://www.xmlvalidation.com> where it validates whether the XML document has an error or not. Below is the picture of xml validator asking for files and success message with no errors.

Please copy your XML document in here:

Or upload it:

No file chosen

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

Figure 18 Validator requesting for files.

### No errors were found

The following files have been uploaded so far:

XML document: [∅](#)

XML schema: [∅](#)

[catalog\\_19031841.dtd](#) [∅](#)

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

Figure 19 Validated successfully without any errors.

Therefore, these components coordinate to complete this coursework. Each and every mentioned component are equally important in integrating in completion of coursework. The External DTD file defined the structure and legal elements and attributes of the XML document. In the DTD the Elements and Attributes were declared. The DTD attribute value like #REQUIRED, #IMPLIED, were used. The DTD Element like Child Element, Only One Occurrence, either/or, etc. were declared.

## 7. Difference between DTD and XSD

Some of the differences between Document Type Definition (DTD) and XML Schema Definition (XSD) are shown in the table below:

Document Type Definition (DTD)	XML Schema Definition (XSD)
1. DTD are the proclamation that define a document type for SGML.	1. XSD describes the elements in an XML document.
2. DTD doesn't support datatypes.	2. XML supports datatypes.
3. It is not extensible in nature.	3. It is extensible in nature.
4. It doesn't provide control on structure of an XML document.	4. It provides comparatively abundant control on structure of an XML document.
5. It is comparatively complex than XSD.	5. It is simpler in comparison to DTD.
6. It doesn't support namespace.	6. It supports namespace.
7. It uses SGML syntax.	7. XML is used for writing XSD.
8. DTD predates and is not as optimized as XSD for XML.	8. XSD is based on XML and is optimized and provides powerful means by which to define XML document structure.

9. DTD doesn't define order for child elements.	9. XSD defines order for child elements.
10. DTD has limited capabilities in describing the document due to its format in terms of how many elements can nest within other elements.	10. Document format can be precisely defining in schema with the features such as minOccurs and maxOccurs to postulate the number of times an element can occur within a particular context.

*Table 8 Difference between DTD and XSD*

Table cited from (geeksforgeeks, 2020).

## 8. Critical Evaluation

During the development of this coursework various problems and hardships were faced. This is because the topic that we're dealing with is completely new and majority part of the topic were to be researched. Below are some of the hardships that were faced during the development of this coursework and was critically evaluated.

The files are supposed to cooperate with each other but when the XML and DTD files are not correlated, it throws out an error.

```
<Song_Writer>Song Writer: Freddie Mercury</Song_Writer>
<Producer>Producer: Roy Thomas Baker</Producer>
<Music_Video>Music Video: Yes</Music_Video>
<trackwebsite>Website: https://www.queenonline.com/</trackwebsite>
<Price>$12.99</Price>
```

Figure 20 Critical Evaluation 1.1

Above error was shown in the project while development. The key feature of VS Code allows us to hover up on the effect and it shows the cause of error. So we see what caused the error by simply hovering over it.



The screenshot shows an XML file in VS Code. The code is identical to Figure 20. A mouse cursor is hovering over the opening tag of the 'Music\_Video' element. A tooltip appears, displaying the error message: "Attribute 'actor' is required and must be specified for element type 'Music\_Video'. xml(MSG\_REQUIRED\_ATTRIBUTE\_NOT\_SPECIFIED)". Below the tooltip, there is a link "View Problem (F8)" and the text "No quick fixes available".

```
<Genre>Genre: Rock</Genre>
<Music_Video>
<!-- Attribute "actor" is required and must be specified for element type
      "Music_Video". xml(MSG_REQUIRED_ATTRIBUTE_NOT_SPECIFIED) -->
<View Problem (F8) No quick fixes available
<Music_Video>Music Video: Yes</Music_Video>
<trackwebsite>Website: https://www.queenonline.com/</trackwebsite>
<Price>$12.99</Price>
```

Figure 21 Critical Evaluation 1.2

It says that an attribute “actor” is required and must be specified for element type. I wondered because not every song has a music video and even if it does, it doesn’t always have an actor on it. Then I realized that if declaring an attribute on DTD was incorrect.

```
<!ATTLIST Track track_id ID #REQUIRED>
<!ATTLIST Cover_Picture cover_id ID #REQUIRED>
<!ATTLIST Title on_youtube CDATA #IMPLIED language CDATA #REQUIRED>
<!ATTLIST Music_Video actor CDATA #REQUIRED>
```

*Figure 22 Critical Evaluation 1.3*

The above screenshot shows the attribute declaration in the DTD and we can clearly see that I mistakenly declared the attribute “actor” as #REQUIRED where required actually means the following attribute must be specified. The correct declaration of that attribute is #IMPLIED where it doesn’t necessarily have to be specified or in other words is declared as an optional attribute.

```
<!ATTLIST Track track_id ID #REQUIRED>
<!ATTLIST Cover_Picture cover_id ID #REQUIRED>
<!ATTLIST Title on_youtube CDATA #IMPLIED language CDATA #REQUIRED>
<!ATTLIST Music_Video actor CDATA #IMPLIED>
```

*Figure 23 Critical Evaluation 1.4*

The attribute’s requirement had been changed to Implied where it eases the specification of a particular attribute or in our case, “actor” attribute on Music\_Video element.

```
<Song_Writer>Song Writer: Freddie Mercury</Song_Writer>
<Producer>Producer: Roy Thomas Baker</Producer>
<Music_Video>Music Video: Yes</Music_Video>
<trackwebsite>Website: https://www.queenonline.com/</trackwebsite>
<Price>$12.99</Price>
<Comments>
```

*Figure 24 Critical Evaluation 1.5*

As we can see in the above figure, no errors were seen after we changed the declaration of the attribute.

Another error faced whilst development of the coursework was simple yet had to be critically evaluated. It was also on DTD declaration and XML file.

```

<!-- Store is the root element -->
<Store xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="catalog_19031841.xsd">

    <!-- Header consists of logo and store information -->
    <Header>
        <Logo id="Img"/>
        <Address>Baluwatar, Kathmandu</Address>
        <Telephone>01-4018293</Telephone>
        <Mobile_no>9813245686</Mobile_no>
        <Website>www.yaminmusic.com</Website>
    </Header>
    <!-- Hero element consists of a tag line -->
    <Hero>
        <HeroContent>Blessing your ears since the 90s</HeroContent>
    </Hero>
    <!-- Music element consists of Track and Song information -->

```

Figure 25 Critical Evaluation 2.1

Above shown screenshot contains an error on the root element ‘Store’. Again, using the VS Code feature, we hover on the error and it says as shown in the figure below.

The screenshot shows a portion of an XML file in VS Code. The root element is 'Store'. A tooltip is displayed over the 'Store' element, containing the following text:

```

< Store
Coursework 2 XML Author: Yam Narayan Basnet Date: 2nd May 2021 Filename:
catalog_19031841.xml
Source: catalog_19031841.dtd
The content of element type "Store" must match "(Header,Music,Footer)". xml(MSG_CONTENT_INVALID)
View Problem (F8) No quick fixes available

```

The 'Store' element is highlighted with a red underline, indicating a validation error. The tooltip also includes the XML declaration and some internal comments.

Figure 26 Critical Evaluation 2.2

The error says that the element type “Store” must match “(Header, Music, Footer)”. I got an idea that the error had to be related with the DTD then I switched the file to see what the DTD has to do.

```

<!ELEMENT Store (Header, Music, Footer)>
<!ELEMENT Header (Logo, Address, Telephone, Mobile_no, Website)>
<!ELEMENT Logo EMPTY>
<!ELEMENT Address (#PCDATA)>

```

*Figure 27 Critical Evaluation 2.3*

After taking the look on the DTD, I realized that I had improvised the XML file by creating a new element 'Hero'. Hero was created to enhance the beauty of the website by putting a tagline in front of the page and was the child element of the parent element 'Store'. The child element Hero was included in the element declaration of the store as shown in the figure below.

```
<!ELEMENT Store (Header, Hero, Music, Footer)>
<!ELEMENT Header (Logo, Address, Telephone, Mobile_no, Website)>
<!ELEMENT Logo EMPTY>
<!ELEMENT Address (#PCDATA)>
```

*Figure 28 Critical Evaluation 2.4*

```
<!-- Store is the root element -->
<Store xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="catalog_19031841.xsd">

    <!-- Header consists of logo and store information -->
    <Header>
        <Logo id="Img"/>
        <Address>Baluwatar, Kathmandu</Address>
        <Telephone>01-4018293</Telephone>
        <Mobile_no>9813245686</Mobile_no>
        <Website>www.yaminmusic.com</Website>
    </Header>
    <!-- Hero element consists of a tag line -->
    <Hero>
```

*Figure 29 Critical Evaluation 2.5*

After adding the Hero in the parent element, the error was solved and no more displayed.

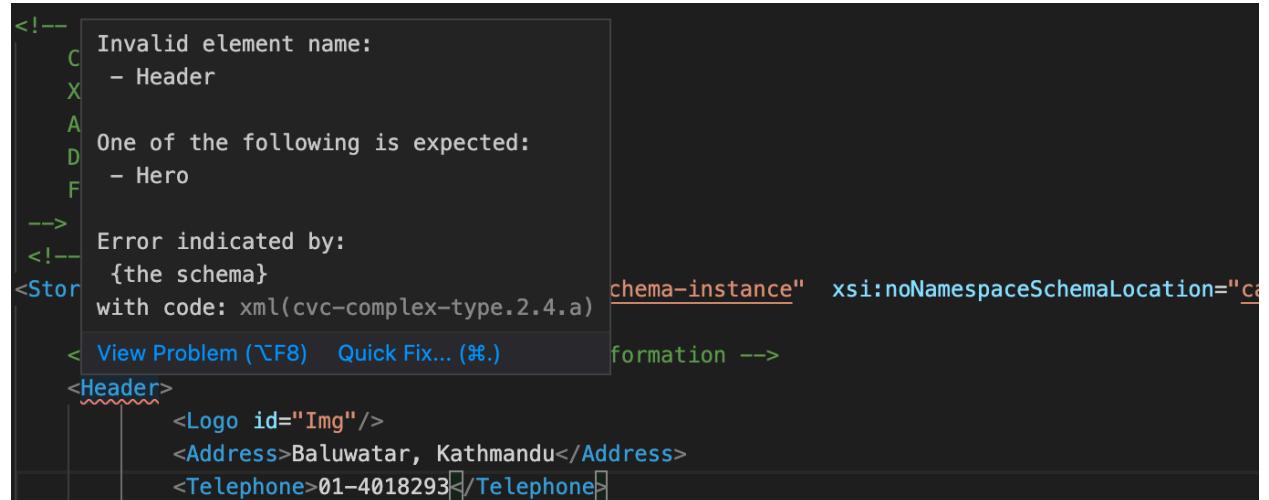
Another problem faced involved with the XSD portion of the coursework.

```

<!-- Header consists of logo and store information -->
<Header>
    <Logo id="Img"/>
    <Address>Baluwatar, Kathmandu</Address>
    <Telephone>01-4018293</Telephone>
    <Mobile_no>9813245686</Mobile_no>
    <Website>www.yaminmusic.com</Website>
</Header>
<!-- Hero element consists of a tag line -->
-->
```

Figure 30 Critical Evaluation 3.1

As shown in the screenshot above, an error was displayed in the header element, then it is hovered and shows the following error.



The screenshot shows an XML editor interface with a dark theme. A tooltip or error message is displayed over the 'Header' element. The message reads:

- Invalid element name:
- Header
- One of the following is expected:
- Hero

The XML code visible in the editor includes:

```

<!--
C   Invalid element name:
X   - Header
A
D
F
-->
<!--
S   One of the following is expected:
  - Hero
  Error indicated by:
  {the schema}
  with code: xml(cvc-complex-type.2.4.a)
< View Problem (⌘F8) Quick Fix... (⌘.) formation -->
<Header>
    <Logo id="Img"/>
    <Address>Baluwatar, Kathmandu</Address>
    <Telephone>01-4018293</Telephone>
```

Figure 31 Critical Evaluation 3.2

It says that the error is indicated by the schema and says Header is an invalid element name and Hero is expected. The having a closer look in the schema, the positioning of Hero and Header element was interchanged and that caused the error as it breached the sequence of the XML file.

```
<xs:complexType>
  <xs:sequence>
    <xs:element ref="Hero"/>
    <xs:element ref="Header"/>
    <xs:element ref="Music"/>
    <xs:element ref="Footer"/>
  </xs:sequence>
```

Figure 32 Critical Evaluation 3.3

```
<xs:complexType>
  <xs:sequence>
    <xs:element ref="Header"/>
    <xs:element ref="Hero"/>
    <xs:element ref="Music"/>
    <xs:element ref="Footer"/>
  </xs:sequence>
</xs:complexType>
```

Figure 33 Critical Evaluation 3.4

```
<!-- Header consists of logo and store information -->
<Header>
  <Logo id="Img"/>
  <Address>Baluwatar, Kathmandu</Address>
  <Telephone>01-4018293</Telephone>
  <Mobile_no>9813245686</Mobile_no>
  <Website>www.yaminmusic.com</Website>
</Header>
<!-- Hero element consists of a tag line -->
```

Figure 34 Critical Evaluation 3.5

The position was interchanged and kept in sequence that caused the editor to no longer display the error.

## 9. Conclusion

The coursework required numerous effort and patience as it was based on various topics that were quite sensitive regarding Emerging Programming platforms and technologies Module. Amidst the COVID-19, and the while lockdown situation, there was some pressure while finalizing the coursework. The main objective of this coursework was to build an XML document including an attractive CSS that resembles a Music store selling songs. YamlInc was the store's name and it had 15 songs on sale.

The relation between the requirement of this coursework and our Module is that developing an XML document like this requires integration of many components like a proper knowledge on Elements, Attributes, DTDs, Schemas, and many more. The most exciting part of this coursework was the CSS. It nurtured my skills and helped me excel in designing aesthetics with minimalism. Personally, I used to find those quite difficult, but with proper guidance and help by my teacher even in the time of pandemic, was a great yardstick for me. The website was developed using XML and CSS and the methods were straight forward and not so overwhelming. The website for YamlInc is enough as per the requirement as it consists of decent level of information regarding the songs.

While completion of this coursework, there were many challenges faced as the code were incorrect in some ways and that led my confidence go low a bit but with the help of my teachers and their honest feedbacks, I could redo my programs and correct them that helped me complete this coursework. Overall, it was a great experience, and we could learn so many things that would surely help me for my personal projects and industry level contributions.

## 10. References

tutorialspoint, 2021. *XML - Overview*. [Online]

Available at: [https://www.tutorialspoint.com/xml/xml\\_overview.htm](https://www.tutorialspoint.com/xml/xml_overview.htm)

[Accessed 2 May 2021].

w3schools, 2021. *Introduction to XML*. [Online]

Available at: [https://www.w3schools.com/xml/xml\\_whatis.asp](https://www.w3schools.com/xml/xml_whatis.asp)

[Accessed 2 May 2021].

Lawton, G., 2021. *TechTarget*. [Online]

Available at: <https://whatis.techtarget.com/definition/XSD-XML-Schema-Definition>

[Accessed 2 May 2021].

w3schools, 2021. *DTD Tutorial*. [Online]

Available at: [https://www.w3schools.com/xml/xml\\_dtd\\_intro.asp](https://www.w3schools.com/xml/xml_dtd_intro.asp)

[Accessed 2 May 2021].

Docs, M. W., 2021. *CSS: Cascading Style Sheets*. [Online]

Available at: <https://developer.mozilla.org/en-US/docs/Web/CSS>

[Accessed 2 May 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 2 May 2021].

tutorialspoint, 2021. *XML - Tree Structure*. [Online]

Available at: [https://www.tutorialspoint.com/xml/xml\\_tree\\_structure.htm](https://www.tutorialspoint.com/xml/xml_tree_structure.htm)

[Accessed 4 May 2021].

crunchbase, 2021. *draw.io*. [Online]

Available at: <https://www.crunchbase.com/organization/draw-io>

[Accessed 4 May 2021].

## 11. Appendix

```
@import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700;800&family=Raleway:wght@400;500;600;700;800&display=swap');

*{
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    background-color: #fafafa;
}

Store{
    display: flex;
    flex-direction: column;
}

/* Header content */
Header{
    /* background-color: blue; */
    display: list-item;
    height: 10vh;
    display: flex;
    justify-content: space-around;
    font-family: 'Poppins', sans-serif;
    align-items: center;
}

Header Address, Telephone, Website{
    font-weight: 600;
}

Header Logo{
    background: url('./assets/logo.svg');
    height: 100px;
    margin-top: 4.4rem;
    width: 100px;
    background-repeat: no-repeat;
}

Header Mobile_no{
    display: none;
}
/* styling for hero tag */
Hero{
    height: 80vh;
    font-family: 'Poppins', sans-serif;
```

```
color: black;
display: flex;
justify-content: center;
align-items: center;
}

Hero HeroContent{
    padding: 0 20px !important;
    font-weight: 700;
    width: 60vw;
    text-align: center;
    font-size: 5.5rem;
    border:2px solid black;
}

/* styling for the floating body */
Music{
    /* background-color: #c8c6a7; */
    display: flex;
    flex-direction: column;
    padding: 0 15rem;
    padding-top: 30px;
    font-family:'Poppins', sans-serif !important ;
    /* color:white; */
}

Music Track{
    display: flex;

    position: relative;
    margin-bottom: 50px;
}

Music Track:nth-child(odd){
    flex-direction: row;
}

Music Track:nth-child(odd) SongContent{
    min-width: 400px;
    /* height:100%; */
    position: absolute;
    top:0;
    right:0;
}
```

```
Music Track:nth-child(even){  
    flex-direction: row-reverse;  
  
}  
  
Music Track:nth-child(even) SongContent{  
    position:absolute;  
    left:0;  
    top:0;  
}  
  
/* styling for the cover picture */  
Music Track Cover_Picture{  
  
    min-height: 500px;  
    min-width: 500px;  
    max-height: 501px;  
    max-width: 501px;  
    background:black;  
    background-size: 90% !important;  
    background-repeat: no-repeat !important;  
}  
  
/* styling for image card */  
  
Music Track:nth-child(1) Cover_Picture{  
    background:url("./assets/bohemian.png");  
}  
  
Music Track:nth-child(2) Cover_Picture{  
    background:url("./assets/heather.jpeg");  
}  
Music Track:nth-child(3) Cover_Picture{  
    background:url("./assets/driverslicence.jpeg");  
}  
Music Track:nth-child(4) Cover_Picture{  
    background:url("./assets/everythingiwanted.jpeg");  
    background-position: bottom;  
}  
  
Music Track:nth-child(5) Cover_Picture{  
    background:url("./assets/pillotalk.jpeg");  
}  
Music Track:nth-child(6) Cover_Picture{  
    background:url("./assets/vaaste.jpeg");  
}  
Music Track:nth-child(7) Cover_Picture{  
    background:url("./assets/pachtaoge.jpg");  
}
```

```
}

Music Track:nth-child(8) Cover_Picture{
    background:url("./assets/ladyinred.jpg");
}

Music Track:nth-child(9) Cover_Picture{
    background:url("./assets/2002.jpeg");
}

Music Track:nth-child(10) Cover_Picture{
    background:url("./assets/shapeofyou.jpeg");
}

Music Track:nth-child(11) Cover_Picture{
    background:url("./assets/heyjude.jpeg");
}

Music Track:nth-child(12) Cover_Picture{
    background:url("./assets/cheerleader.jpeg");
}

Music Track:nth-child(13) Cover_Picture{
    background:url("./assets/someoneyouloved.jpeg");
}

Music Track:nth-child(14) Cover_Picture{
    background:url("./assets/alliwant.jpeg");
}

Music Track:nth-child(15) Cover_Picture{
    background:url("./assets/changes.jpeg");
}

SongContent{
    display: flex;
    flex-direction: column;
    justify-content: center;
}

SongContent Title, Artist, Group, Album, Genre, Duration,
Release_Date, Plays, Song_Writer, Director, Producer,
Music_Video, Price, trackwebsite{
    font-family:'Raleway', sans-serif;
    padding: 6px;
    font-weight: 600;
}
```

```
SongContent Title{
    font-family:'Poppins', sans-serif;
    font-size: 2rem;
    color: white;
    width: fit-content;
    background-color: black;
    padding: 10px 8px;
    cursor: pointer;
    transition: background-color 0.5s ease-in-out;
}

SongContent Title:hover{
    background-color: #fafafa;
    color: black;
    font-family:'Poppins', sans-serif;
    font-weight: 700;
}

SongContent Price{
    font-family:'Poppins', sans-serif;
    font-size: 2rem;
}

/* styling for footer */
Footer{
    height: 12vh;
    background-color: black;
    color: #f4f4f4;
    font-family:'Poppins', sans-serif;
    display: flex;
    font-weight: 500;
    justify-content: center;
    align-items: center;
}
```