LONDON METROPOLITAN UNIVERSITY

Islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CS5004NI Emerging Program Platforms and Technologies

Assessment Weightage & Type

30% Individual Coursework

Year and Semester

2020-21 Autumn

Student Name: Prashanna GC

London Met ID: 19031368

College ID: NP01CP4A190249

Assignment Due Date: May 7th, 2021

Assignment Submission Date: May 7th, 2021

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

Table of Contents

LIST OF FIGURES

LIST OF TABLES

1. Introduction

XML stands for Extensible Markup Language, which is a text-based markup language which is used to describe structured data. XML is profile of an ISO standard SGML which stands for Standard Generalized Markup Language. XML is widely used to store and arrange data, and allows us to create a database information without having a actual database. Simply, it helps to make the webpage smarter and more versatile. XML can be referred as extensible markup language because this is not fixed like a HTML document, instead it is a language that helps to describe other language and it helps to increase the belief of the user's documents. And, also because it allows the user to create their own custom tags based on the user's web application. An XML files is easily used by storing the codes in a simple text file and processing it in a software that is capable of interpreting XML. (tutorialspoint, 2021)

CSS stands for Cascading Style Sheets, which is a simple style sheet language which allows to provide styles about the designing and layout for the web pages by using the CSS properties. It is generally used for HTML and XML web pages and user interfaces. XML Schema is a language for showing the limitation of a XML document. DTD is one of the many XML Schema language which is used to describe the elements, attributes, and data types in a XML schema document. The element in the XML file are separated in blocks with the child and sub-child elements and the data types of the elements and attribute are provided which helps to define the element. (W3C, 2021)

2. XML Content

2.1 XML Code

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<?xml-stylesheet type="text/css" href="catalog_19031368.css"?>
<!DOCTYPE store SYSTEM "catalog_19031368.dtd">
```

```
<store xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="catalog_19031368.xsd">
   <store_details>

      <store_name> SOUND STRONG MUSIC STORE</store_name>

      <logo></logo>


      <quote>
       " Music is the literature of the heart; it commences where speech ends. "
      </quote>


      <background></background>
      <description>
      "Sound Strong Music Store" is a music store which has been working in
selling music albums services for a decade. It has been providing best quality
music albums as soon as the album is released. This music store has been in
demand lately with the quality of albums they provide. Apart from this music
instruments rental service as well as buying musical instruments is also what
Sound Strong has been working on. To manage the heavy influx of customers
and the difficulty with phone call-based booking, this website is for providing the
customer the details of latest music albums as well as for information details of
the music store.

      </description>

   </store_details>



   <heading> SONGS</heading>

   <songs>
      <song_details song_id="S1">
         <song_name>
            <sn>Song Name: </sn>27
         </song_name>
         <album>
            <a>Album: </a> Bloom</album>
         <genre genre_id="G1">
            <gn>Genre: </gn>Hip-Hop/Rap</genre>
         <release_year>
            <ry>Release Year:</ry>2017</release_year>
         <singer>
```

```
        <s>Singers: </s>Machine Gun Kelly</singer>
    <song_writter>
        <sw>Writter: </sw>Machine Gun Kelly</song_writter>
    <song_producer>
        <sp>Producer: </sp>slimXX Baze MGK</song_producer>
    <song_director>
        <sd>Director: </sd>Ben Griffin</song_director>
    <nomination>
        <n>Nomination: </n> None</nomination>
</song_details>


<song_images song_image_id="SI1"></song_images>



<song_details song_id="S2">
    <song_name>
        <sn>Song Name: </sn>Ride</song_name>
    <album>
        <a>Album: </a> BlurryFace</album>
    <genre genre_id="G2">
        <gn>Genre: </gn>Electropop </genre>
    <release_year>
        <ry>Release Year: </ry>2015</release_year>
    <singer>
        <s>Singers: </s>Tyler Joseph, Chris Salih</singer>
    <song_writter>
        <sw>Writter: </sw>Tyler Joseph</song_writter>
    <song_producer>
        <sp>Producer: </sp>Ricky Reed</song_producer>
    <song_director>
        <sd>Director: </sd>Ricky Reed</song_director>
    <nomination>
        <n>Nomination: </n> Billboard Music Award</nomination>
</song_details>

<song_images song_image_id="SI2"></song_images>

<song_details song_id="S3">
    <song_name>
        <sn>Song Name: </sn>SuperHeroes</song_name>
    <album>
        <a>Album: </a> Voices</album>
    <genre genre_id="G3">
        <gn>Genre: </gn>Pop </genre>
```

```
    <release_year>
      <ry>Release Year: </ry>     2014</release_year>
    <singer>
      <s>Singers: </s>Danny O'Donoghue</singer>
    <song_writter>
      <sw>Writter: </sw>Danny and Mark Sheehan</song_writter>
    <song_producer>
      <sp>Producer: </sp>Craig Mazin</song_producer>
    <song_director>
      <sd>Director: </sd> Zack Synder</song_director>

    <nomination>
      <n>Nomination: </n>None</nomination>
</song_details>

<song_images song_image_id="SI3"></song_images>


<song_details song_id="S4">
    <song_name>
      <sn>Song Name: </sn>     Perfect</song_name>
    <album>
      <a>Album: </a> Divide</album>
    <genre genre_id="G4">
      <gn>Genre: </gn>Pop</genre>
    <release_year>
      <ry>Release Year: </ry> 2017</release_year>
    <singer>
      <s>Singers: </s> Ed Sheeran</singer>
    <song_writter>
      <sw>Writter:</sw>    Ed Sheeran </song_writter>
    <song_producer>
      <sp>Producer: </sp> Will Hicks</song_producer>
    <song_director>
      <sd>Director: </sd>  Jason Koenig</song_director>

    <nomination>
      <n>Nomination: </n>MTV Music Video Award</nomination>
</song_details>

<song_images song_image_id="SI4"></song_images>

<song_details song_id="S5">
    <song_name>
      <sn>Song Name: </sn>     IDGAF</song_name>
    <album>
```

```
            <a>Album: </a> Dua Lipa: Complete Edition</album>
         <genre genre_id="G5">
            <gn>Genre: </gn>Pop</genre>
         <release_year>
            <ry>Release Year: </ry> 2018</release_year>
         <singer>
            <s>Singers: </s>  Dua Lipa</singer>
         <song_writter>
            <sw>Writter: </sw>> Dua Lipa </song_writter>
         <song_producer>
            <sp>Producer: </sp>  Stephen Kozmeniuk</song_producer>
         <song_director>
            <sd>Director: </sd> Henry Scholfield</song_director>

         <nomination>
            <n>Nomination: </n>Brit Award for British Single with
Mastercard</nomination>
      </song_details>

      <song_images song_image_id="SI5"></song_images>

      <song_details song_id="S6">
         <song_name>
            <sn>Song Name: </sn>Rap God</song_name>
         <album>
            <a>Album: </a> The Marshall Mathers LP 2</album>
         <genre genre_id="G6">
            <gn>Genre: </gn>Hip-Hop/Rap</genre>
         <release_year>
            <ry>Release Year: </ry>2013</release_year>
         <singer>
            <s>Singers: </s>Eminem</singer>
         <song_writter>
            <sw>Writter: </sw>Eminem </song_writter>
         <song_producer>
            <sp>Producer: </sp>DVLP Filthy (co.)</song_producer>
         <song_director>
            <sd>Director: </sd>DVLP Filthy (co.)</song_director>

         <nomination>
            <n>Nomination: </n>  Grammy Award</nomination>
      </song_details>

      <song_images song_image_id="SI6"></song_images>
```

```
<song_images song_image_id="SI7"></song_images>
<song_details song_id="S7">
   <song_name>
      <sn>Song Name: </sn>You and I</song_name>
   <album>
      <a>Album: </a> Midnight Memories</album>
   <genre genre_id="G7">
      <gn>Genre: </gn>Soft Rock</genre>
   <release_year>
      <ry>Release Year: </ry>2013</release_year>
   <singer>
      <s>Singers: </s>Harry Styles and Loius Tomlimson</singer>
   <song_writter>
      <sw>Writter: </sw>     Tyler Joseph </song_writter>
   <song_producer>
      <sp>Producer: </sp>John Ryan</song_producer>
   <song_director>
      <sd>Director: </sd>   Ben Winston</song_director>

   <nomination>
      <n>Nomination: </n> Teen Choice Award</nomination>
</song_details>

<song_details song_id="S8">
   <song_name>
      <sn>Song Name: </sn>Kill this love</song_name>
   <album>
      <a>Album: </a> Kill This Love</album>
   <genre genre_id="G8">
      <gn>Genre: </gn>Hip hop music, K-Pop</genre>
   <release_year>
      <ry>Release Year: </ry>April 4, 2019</release_year>
   <singer>
      <s>Singers: </s>BlackPink</singer>
   <song_writter>
      <sw>Writter: </sw>Teddy Park</song_writter>
   <song_producer>
      <sp>Producer: </sp>Teddy Park and R.Tee</song_producer>
   <song_director>
      <sd>Director: </sd>Seo Hyun-seung</song_director>

   <nomination>
      <n>Nomination:</n> MTV K-Pop Video Music Award </nomination>
</song_details>

<song_images song_image_id="SI8"></song_images>
```

```
    </songs>

  <footer>
      <logo_last></logo_last>
      <address>Location: Kamal Marg, Kamal Pokhari, Kathmandu,
Nepal</address>
      <telephone_number>Tel: 4512929, 980-1022446</telephone_number>
      <social_media><![CDATA[Whatsapp & Viber Number:
9801000400]]></social_media>
      <website_address>Email: info@islington.edu.np</website_address>
      <about_us></about_us>
  </footer>

</store>
```

2.1 Tree Diagram

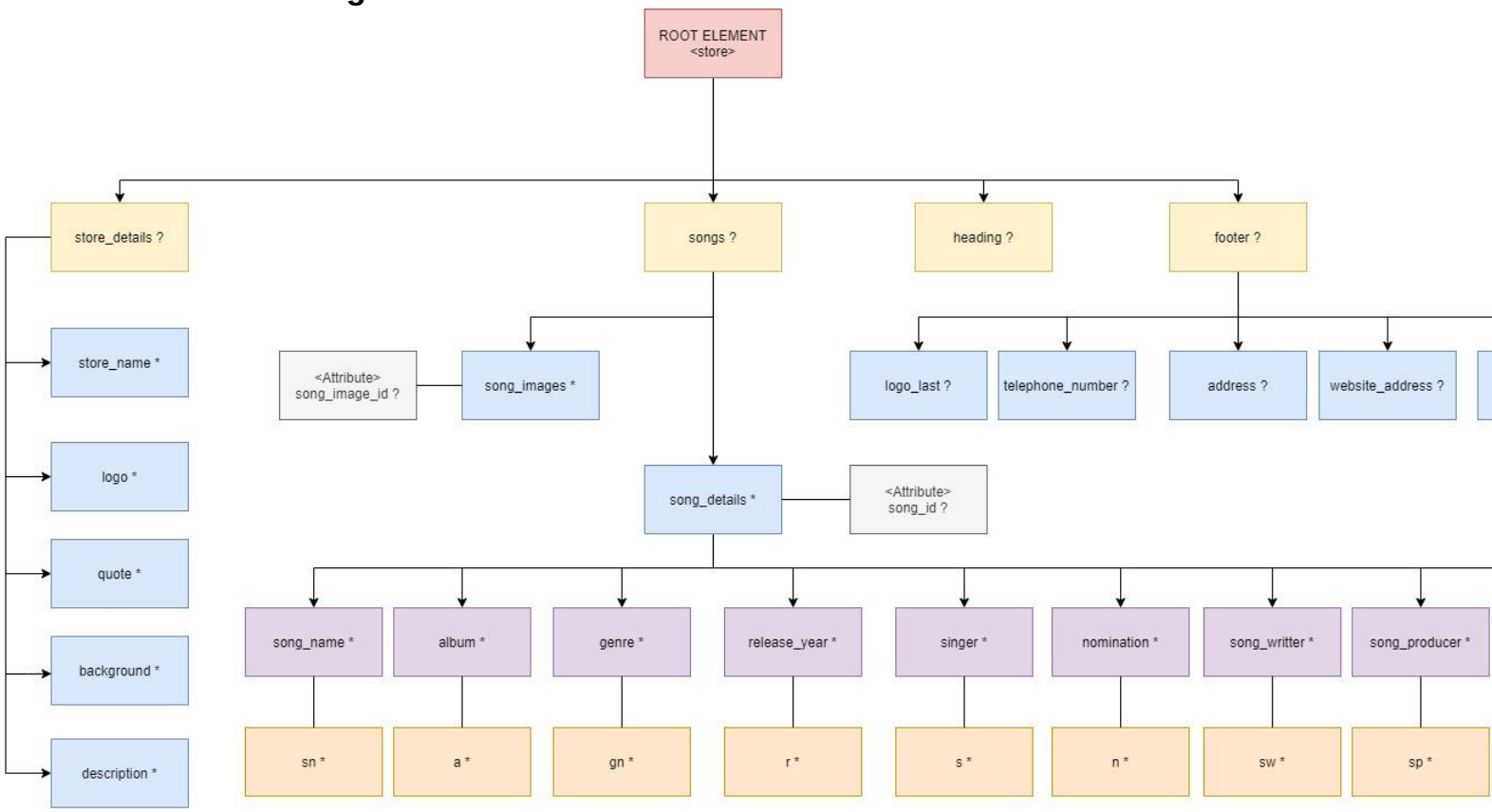

Figure 1: Tree Diagram

3. Schema Content

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="store" type="storeType">
    <xs:annotation>
      <xs:documentation>&lt;!DOCTYPE store SYSTEM
"XML.dtd"></xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:complexType name="song_nameType" mixed="true">
    <xs:sequence>
      <xs:element type="xs:string" name="sn"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="storeType">
```

```
16    <xs:sequence>
17      <xs:element type="store_detailsType" name="store_details"/>
18      <xs:element type="xs:string" name="heading"/>
19      <xs:element type="songsType" name="songs"/>
20      <xs:element type="footerType" name="footer"/>
21    </xs:sequence>
22  </xs:complexType>
23  <xs:complexType name="release_yearType" mixed="true">
24    <xs:sequence>
25      <xs:element type="xs:string" name="ry"/>
26    </xs:sequence>
27  </xs:complexType>
28  <xs:complexType name="song_directorType" mixed="true">
29    <xs:sequence>
30      <xs:element type="xs:string" name="sd"/>
31    </xs:sequence>
32  </xs:complexType>
33  <xs:complexType name="song_detailsType" mixed="true">
34    <xs:sequence>
35      <xs:element type="song_nameType" name="song_name" minOccurs="0"/>
36      <xs:element type="albumType" name="album" minOccurs="0"/>
37      <xs:element type="genreType" name="genre" minOccurs="0"/>
38      <xs:element type="release_yearType" name="release_year"
    minOccurs="0"/>
39      <xs:element type="singerType" name="singer" minOccurs="0"/>
40      <xs:element type="song_writterType" name="song_writter" minOccurs="0"/>
41      <xs:element type="song_producerType" name="song_producer"
    minOccurs="0"/>
42      <xs:element type="song_directorType" name="song_director"
    minOccurs="0"/>
43      <xs:element type="nominationType" name="nomination" minOccurs="0"/>
44    </xs:sequence>
45    <xs:attribute type="xs:string" name="song_id" use="optional"/>
46    <xs:attribute type="xs:string" name="song_image_id" use="optional"/>
47  </xs:complexType>
48  <xs:complexType name="singerType" mixed="true">
49    <xs:sequence>
50      <xs:element type="xs:string" name="s"/>
51    </xs:sequence>
52  </xs:complexType>
53  <xs:complexType name="song_imagesType" mixed="true">
54    <xs:sequence>
55      <xs:element type="song_nameType" name="song_name" minOccurs="0"/>
56      <xs:element type="albumType" name="album" minOccurs="0"/>
57      <xs:element type="genreType" name="genre" minOccurs="0"/>
```

```
58      <xs:element type="release_yearType" name="release_year"
    minOccurs="0"/>
59      <xs:element type="singerType" name="singer" minOccurs="0"/>
60      <xs:element type="song_writterType" name="song_writter" minOccurs="0"/>
61      <xs:element type="song_producerType" name="song_producer"
    minOccurs="0"/>
62      <xs:element type="song_directorType" name="song_director"
    minOccurs="0"/>
63      <xs:element type="nominationType" name="nomination" minOccurs="0"/>
64    </xs:sequence>
65    <xs:attribute type="xs:string" name="song_image_id" use="optional"/>
66    <xs:attribute type="xs:string" name="song_id" use="optional"/>
67  </xs:complexType>
68  <xs:complexType name="genreType" mixed="true">
69    <xs:sequence>
70      <xs:element type="xs:string" name="gn"/>
71    </xs:sequence>
72    <xs:attribute type="xs:string" name="genre_id" use="optional"/>
73  </xs:complexType>
74  <xs:complexType name="songsType">
75    <xs:choice maxOccurs="unbounded" minOccurs="0">
76      <xs:element type="song_detailsType" name="song_details"/>
77      <xs:element type="song_imagesType" name="song_images"/>
78    </xs:choice>
79  </xs:complexType>
80  <xs:complexType name="albumType" mixed="true">
81    <xs:sequence>
82      <xs:element type="xs:string" name="a"/>
83    </xs:sequence>
84  </xs:complexType>
85  <xs:complexType name="store_detailsType">
86    <xs:sequence>
87      <xs:element type="xs:string" name="store_name"/>
88      <xs:element type="xs:string" name="logo"/>
89      <xs:element type="xs:string" name="quote"/>
90      <xs:element type="xs:string" name="background"/>
91      <xs:element type="xs:string" name="description"/>
92    </xs:sequence>
93  </xs:complexType>
94  <xs:complexType name="song_producerType" mixed="true">
95    <xs:sequence>
96      <xs:element type="xs:string" name="sp"/>
97    </xs:sequence>
98  </xs:complexType>
99  <xs:complexType name="song_writterType" mixed="true">
100       <xs:sequence>
```

```
101         <xs:element type="xs:string" name="sw"/>
102       </xs:sequence>
103     </xs:complexType>
104     <xs:complexType name="footerType">
105       <xs:sequence>
106         <xs:element type="xs:string" name="logo_last"/>
107         <xs:element type="xs:string" name="address"/>
108         <xs:element type="xs:string" name="telephone_number"/>
109         <xs:element type="xs:string" name="social_media"/>
110         <xs:element type="xs:string" name="website_address"/>
111         <xs:element type="xs:string" name="about_us"/>
112       </xs:sequence>
113     </xs:complexType>
114     <xs:complexType name="nominationType" mixed="true">
115       <xs:sequence>
116         <xs:element type="xs:string" name="n"/>
117       </xs:sequence>
118     </xs:complexType>
119   </xs:schema>
```

4. DTD Content

```
<!ELEMENT store (store_details?,heading?,songs?,footer?)>


<!ELEMENT store_details (store_name?,logo?,quote?,background?,description?)>



<!ELEMENT store_name (#PCDATA)>


<!ELEMENT logo EMPTY>


<!ELEMENT quote (#PCDATA)>


<!ELEMENT background EMPTY>


<!ELEMENT description (#PCDATA)>
```

```
<!ELEMENT heading (#PCDATA)>

<!ELEMENT songs (song_details|song_images)*>

<!ELEMENT song_details
(song_name|album|genre|release_year|singer|song_writter|song_producer|song_direct
or|nomination)*>
<!ATTLIST song_details
song_id CDATA #IMPLIED
>

<!ELEMENT song_name (#PCDATA|sn)*>

<!ELEMENT sn (#PCDATA)>

<!ELEMENT album (#PCDATA|a)*>

<!ELEMENT a (#PCDATA)>

<!ELEMENT genre (#PCDATA|gn)*>
<!ATTLIST genre
genre_id CDATA #IMPLIED
>
```

```
<!ELEMENT gn (#PCDATA)>

<!ELEMENT release_year (#PCDATA|ry)*>


<!ELEMENT ry (#PCDATA)>


<!ELEMENT singer (#PCDATA|s)*>


<!ELEMENT s (#PCDATA)>


<!ELEMENT song_writter (#PCDATA|sw)*>

<!ELEMENT sw (#PCDATA)>


<!ELEMENT song_producer (#PCDATA|sp)*>


<!ELEMENT sp (#PCDATA)>


<!ELEMENT song_director (#PCDATA|sd)*>
```

```
<!ELEMENT sd (#PCDATA)>

<!ELEMENT nomination (#PCDATA|n)*>

<!ELEMENT n (#PCDATA)>

<!ELEMENT song_images EMPTY>
<!ATTLIST song_images
song_image_id CDATA #IMPLIED
>

<!ELEMENT footer
(logo_last|address|telephone_number|social_media|website_address|about_us)*>

<!ELEMENT logo_last EMPTY>

<!ELEMENT address (#PCDATA)>

<!ELEMENT telephone_number (#PCDATA)>

<!ELEMENT social_media (#PCDATA)>
```

```
<!ELEMENT website_address (#PCDATA)>


<!ELEMENT about_us EMPTY>
```

5. Testing

5.1. Test 1

Test No	1
Objective	To validate the XML document.
Action	The XML Code of the webpage is copied and pasted in XML Validation website.
Expected Result	A message saying “No errors were found” should be displayed.
Actual Result	A message saying “No errors were found” was displayed.
Test Result	Test Successful.

Table 1: To validate the XML document

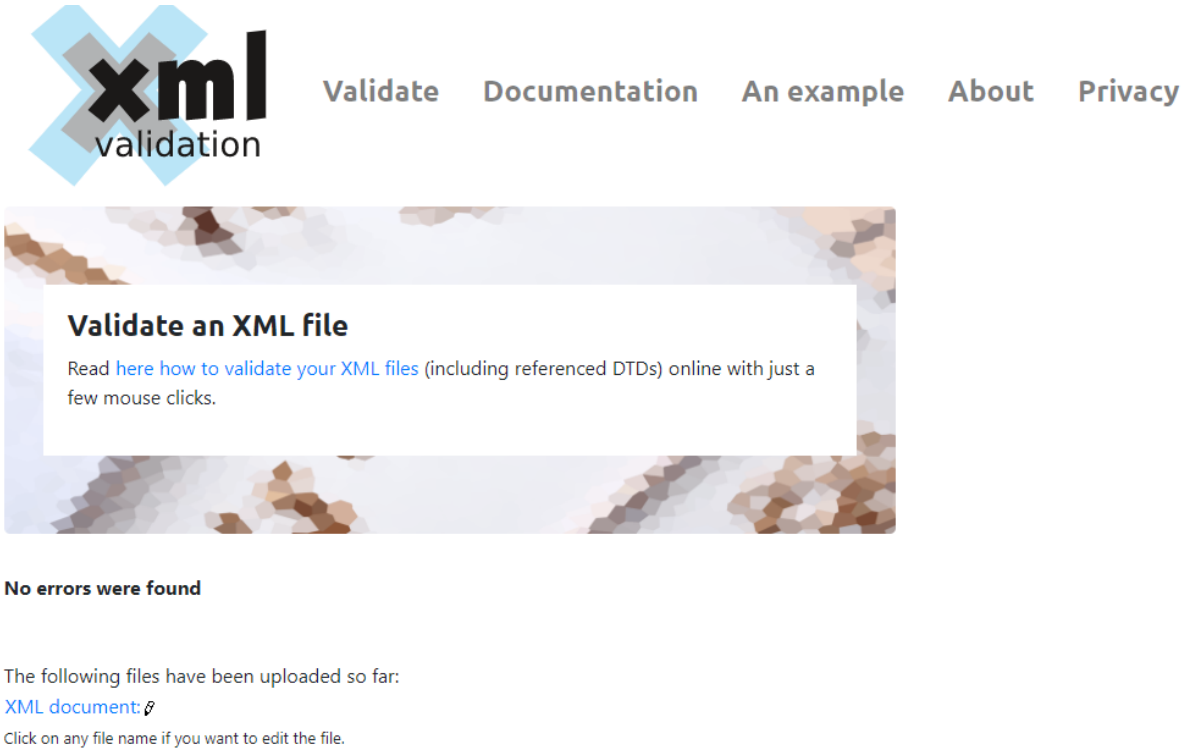

Figure 2:XML Validation without errors

5.2. Test 2

Test No	2
Objective	To validate the XML file against Schema file.
Action	The XML Code of the webpage including the schema file code is copied and pasted in XML Validation website.
Expected Result	A message saying “No errors were found” should be displayed.
Actual Result	A message saying “No errors were found” was displayed.
Test Result	Test Successful.

Table 2: To validate the XML file against Schema file.

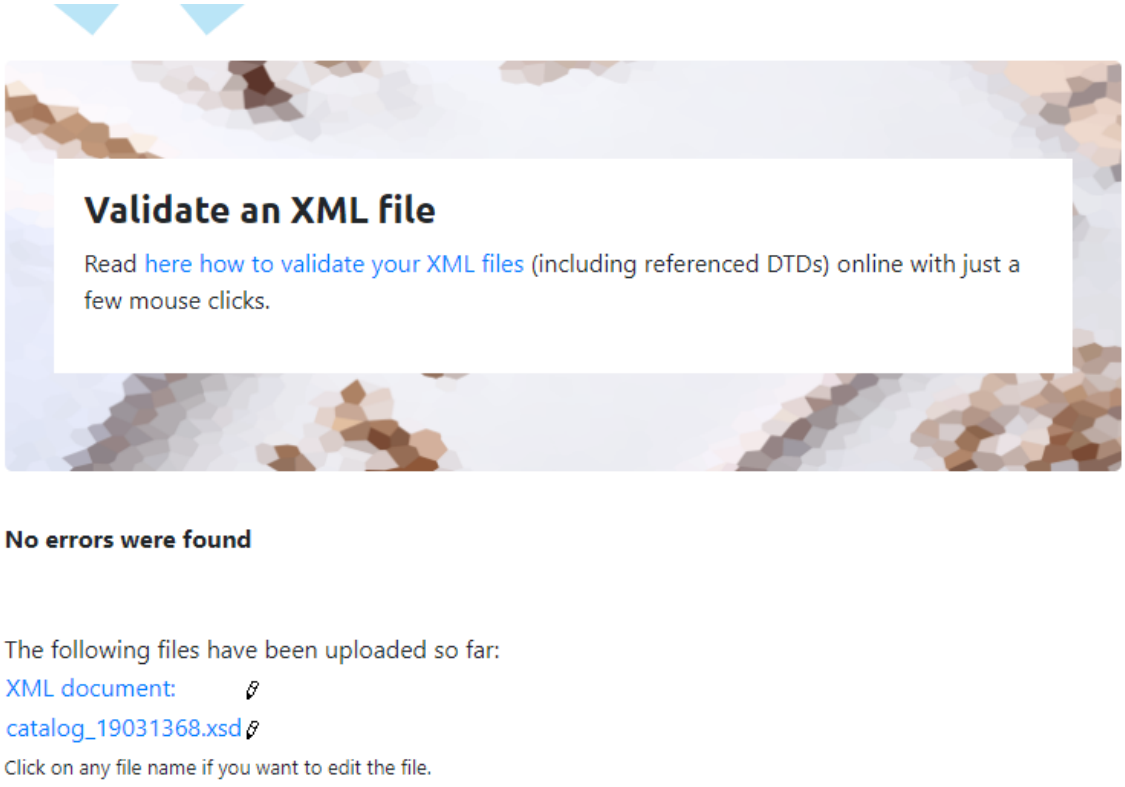

Figure 3: XML Validation of Schema file without errors

5.3. Test 3

Test No	3
Objective	To link CSS file with XML file.
Action	The syntax to link CSS file is written in XML file.
Expected Result	The CSS properties provided in CSS file should work when running XML file.
Actual Result	The CSS properties provided in CSS file worked when running XML file.
Test Result	Test Successful

Table 3: To link CSS file with XML file.

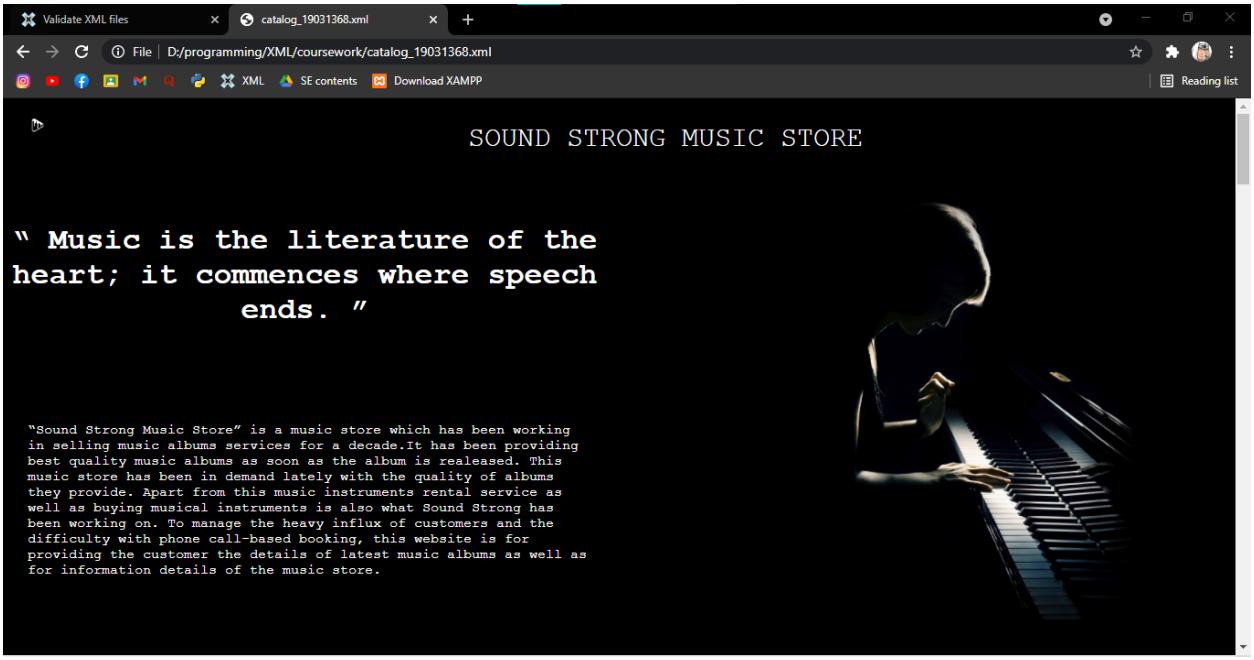

Figure 4: XML file with CSS.

5.4. Test 4

Test No	4
Objective	To check whether the XML file is linked with CSS file or not.
Action	The syntax to link CSS file is removed from the XML file.
Expected Result	The XML file should not have any CCS properties applied in CSS file.
Actual Result	The XML file do not have any CCS properties applied in CSS file.
Test Result	Test Successful

Table 4: To check whether the XML file is linked with CSS file or not.

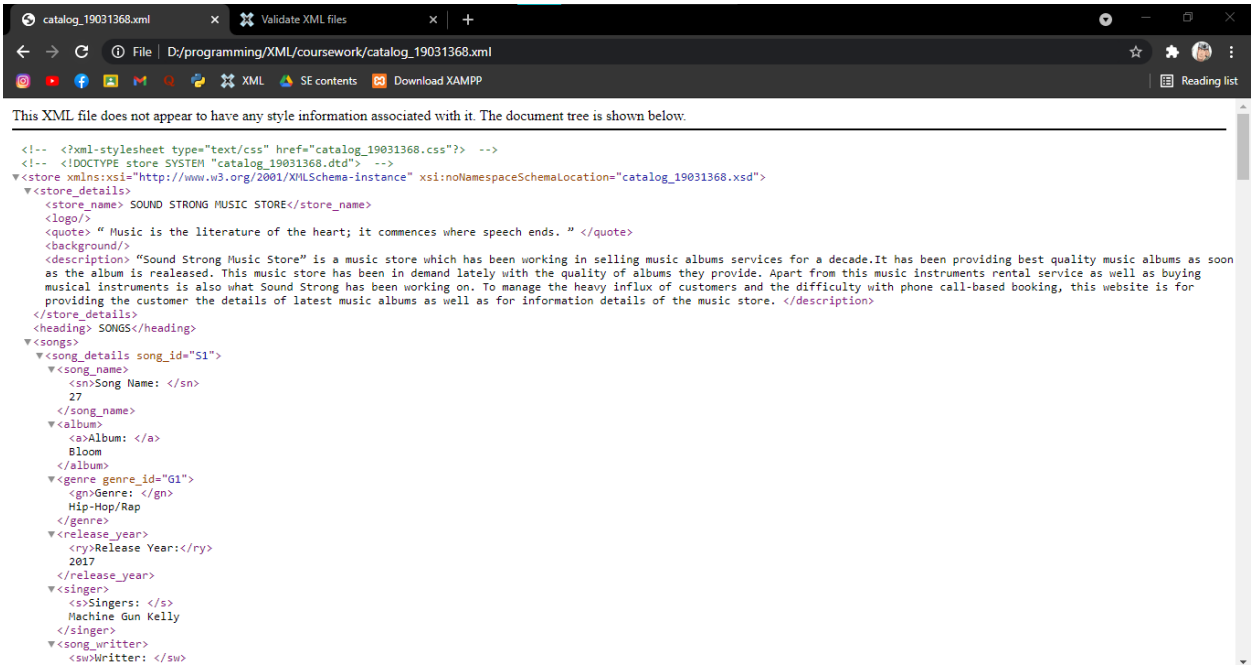

Figure 5: XML file without CSS properties

5.5. Test 5

Test No	5
Objective	To validate the XML file against DTD file.
Action	The XML Code of the webpage including the DTD file code is copied and pasted in XML Validation website.
Expected Result	A message saying "No errors were found" should be displayed.
Actual Result	A message saying "No errors were found" was displayed.
Test Result	Test Successful.

Table 5: To validate the XML file against DTD file.

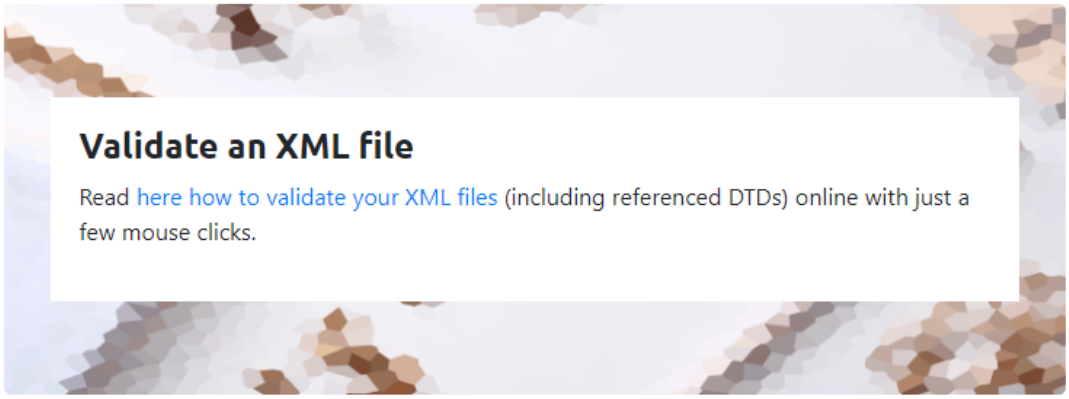

Figure 6: XML Validation against DTD file without errors.

6. Coursework Development

The motive of this coursework was to work as an XML developer and to build a web system for a music store which should have the information of different music albums available for marketing purposes. The web system according to the coursework had requirements like to include store name, address, telephone number, website address (URL), and logo image in the website. The website should consist detailed information of the songs provided in it and according to the requirement of the coursework, this all information where to be separated into different elements and necessary attributes were provided to store the element ID.

Some tools used in the coursework are as following:

i) Draw.io

Draw.io is a free online diagram software which provides simple tools to create flowchart, network diagram, UML online, ER diagram etc. It helps to import the diagram files in different extensions like .vsdx files, gliffy files, Lucid chart files etc. In this coursework, this

software is used to create the tree view diagram of the elements and attributed used in the XML document.

ii) Sublime Text

Sublime Text is a source code editor. For this coursework, I used sublime code editor for all coding of XML, XSD, DTD and CSS file. The file here involved all the required coding for building the website.

iii) Google Chrome

Google Chrome is the most used and popular web browser developed by google. In this coursework, I used google chrome to run my XML file as well as for validation, I visited xmlvalidation.com website through chrome. Any kind of errors related the code during the validation is obtained from this website and the respective errors were solved in the codes.

7. Difference between Schema and DTD

DTD	SCHEMA
1. DTD stands for Document Type Definition.	1. Schema/XSD stands for Extensible Schema Definition.
2. DTD doesn't support namespace.	2. Schema supports namespace.

3. DTD has pre-defined elements for root tag, attributes, and variables.	3. Schema doesn't have pre-defined elements for root tag, attributes, and variables.
4. DTD supports only one datatype i.e. string datatype.	4. Schema supports every primitive or inbuilt datatypes.
5. DTD doesn't support code reusability.	5. Schema supports code reusability.
6. DTD document uses SGML syntaxes.	6. Schema document uses XML syntaxes.
7. Constraints i.e. elements length cannot be set in a DTD file.	7. Constraints i.e. elements length can be set in a Schema file.
8. User can use inline definitions in a DTD file.	8. User cannot use inline definitions in a Schema file.
9. DTD is used to describe the attributes of element of an XML file.	9. Schema is used to describe the rules for attributes of an element.
10. DTD can be both internal and external.	10. Schema can only be external.

(ATechDaily, 2020-21)

Table 6: Difference between DTD and Schema

8. Critical Evaluation

In this coursework, we were asked to design a website using XML and XSS for a music store which holds the information of the latest music albums. The design of the web depends on the user choice. The web system for the music store consists of the information of different music albums available for marketing purposes. In this website, according to the requirement of the coursework logo, title and description is given at the very beginning. Then the details about the song with image and hover animation is given. For the development of the website, the XML file contained different elements which holds respective data. The root element, parent elements, child elements, sub-child elements and attributes were respective placed. The CSS file with CSS properties was linked to the XML file for the design purpose.

The most difficulties I faced during this coursework was related to validation of XML file against schema file and validation of XML file against DTD file. To build the XML file and use the CSS properties for the design purpose was not that difficult for me since using CSS properties is as same as while working with HTML and I was quite familiar with HTML. I faced a lot of errors during validation. I used venetian blind method to build the XSD file, during this I was unable to sort out the placement of the parent elements, child elements and sub-child elements. The attributeGroup reference was also not in the correct placements. Many simple errors were also obtained during the validation process. But, with the help of proper research throughout the internet and with the help of friends and tutors via discord and hangout, I was able to remove all the errors while validating XML file. And, lastly the report section of the coursework was started. The report section did not have much difficulty because simple testing and theories were only the requirement of the report section.

9. Conclusion

This coursework was all about designing a website using XML and CSS, then applying schema and DTD content in the XML document. The website requirement to

show detail information of music store which consists information of different songs of different singers. The designing of the web using XML and CSS was quite easy and fun. But the main purpose of this coursework was to create a schema and DTD, then validating the XML document against schema and DTD.

There were many errors which obtained during this coursework, but most I face difficulty was during validation process. This was new topic for me, so it took a lot of time to solve the errors. The physical workshop classes were a lot of help for this coursework. I also had to browse over the internet and research for this validation. And, I also discussed some queries related to the validation error with the module leader via hangout. At the end, I was able to submit the coursework in time.

10. References

ATechDaily, 2020-21. *Difference-between-DTD-and-XSD-schema.* [Online]
Available at: https://www.atechdaily.com/posts/Difference-between-DTD-and-XSD-schema

tutorialspoint, 2021. *xml_overview.* [Online]
Available at: https://www.tutorialspoint.com/xml/xml_overview.htm

W3C, 2021. *schema.* [Online]
Available at: https://www.w3.org/standards/xml/schema