Module Code & Module Title

CS5004NI Emerging Programming Platforms and Technologies

Assessment Weightage & Type

30% Individual Coursework

Year and Semester

2020-21 Spring

Student Name: Bijay Bharati

Group: L2C4

London Met ID: 19030824

College ID: NP01CP4A190041

Assignment Due Date: 7th May 2021

Assignment Submission Date: 7th May 2021

Word Count:1880

Contents

Figures

Tables

Introduction

In this coursework we are developing XML to store data about a music shop. The scenario is that we are approached by a music store and provided some guidelines and following the guidelines we must develop a suitable XML structure for the shop to keep record of their music albums they have available.

XML

XML stands for eXtensible Markup Language. It is a language (not a programming language) that uses the markup and can extend. It is derived from Standard Generalized Markup Language (SGML). XML also uses DTDs (Document Type Definitions) to define the XML document structure (Guru99, 2021).

XML is not a programming language and is not used to perform calculations, it is simply used to store and transport data and to display data. CSS can be used to display data along with XML. XML files have .xml extension and can be opened with any text editors.

There are two main uses for XML: One is a way to represent low-level data, for example configuration fi les. The second is a way to add metadata to documents; for example, you may want to stress a particular sentence in a report by putting it in italics or bold (Joe Fawcett, 2012).

XML tags look like HTML tags XML also allows us to create our own custom tags and the tags in XML are case sensitive. There are 2 versions of XML 1.0 and 1.1, the most used version is 1.0. The XML tags are called elements and they can even have attributes. Example:

<Name gender=”m” age=”20> Jon </Name>

Name is an element, gender and age are attributes with certain value.

XML Tree

XML documents are formed as element trees An XML tree starts at a root element and branches from the root to child elements. All elements can have sub elements (child elements) (W3Schools, 2021). The following figure shows the tree structure used by us.

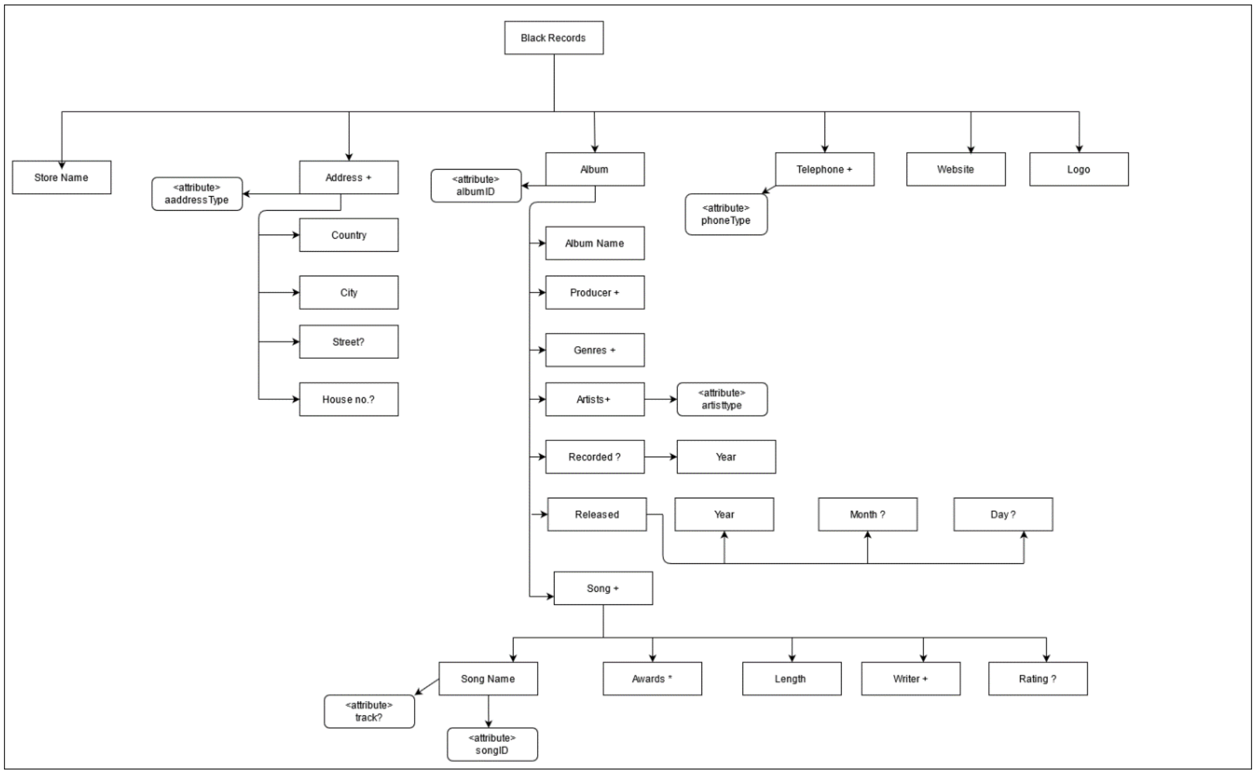


Figure 1: XML tree

Black Records is the root element which is the store name. The tree gives idea about the structure of the document and about the elements and the attributes.

XML Content

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<?xml-stylesheet href=" catalog_19039824.css"?>

<!DOCTYPE BlackRecords SYSTEM "catalog_19039824.dtd">

<BlackRecords xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="catalog_19039824.xsd">

    <StoreName>Black Records Music Store</StoreName>

    <Telephone phoneType="Ncell">9809909090</Telephone>
    <Telephone phoneType="Ntc">9809009090</Telephone>

    <website>https://www.brecords.com</website>
    <Logo>Black Records</Logo>

    <Address addressType="Owned">
        <Country>Nepal</Country>
        <City>Koteshwor</City>
        <Street>Drow</Street>
        <HouseNo>11</HouseNo>
    </Address>

    <Address addressType="Rental">
        <Country>Nepal</Country>
        <City>Bhaktapur</City>
    </Address>

    <Album albumID="A100">
        <AlbumName>Highway to Hell</AlbumName>
        <Producer>Robert John "Mutt" Lange</Producer>
        <Genres>Hard rock</Genres>
        <Artists artistType="Band">AC/DC</Artists>
        <Recorded>1979</Recorded>
        <Released>
            <Year>1979</Year>
            <Month>July</Month>
            <Day>27</Day>
        </Released>
        <Song songID="S001" track="A">
            <SongName>Highway to Hell</SongName>
            <Awards>Most Played Australian Work Overseas</Awards>
            <Length>3:29</Length>
            <Writer>Angus Young</Writer>
            <Writer>Malcolm Young</Writer>
            <Writer>Brian Johnson</Writer>
            <Rating>9</Rating>
        </Song>
        <Song songID="S002" track="A">
            <SongName>Girls Got Rhythm</SongName>
            <Length>3:24</Length>
            <Writer>Angus Young</Writer>
            <Writer>Malcolm Young</Writer>
            <Writer>Brian Johnson</Writer>
            <Rating>8</Rating>
        </Song>
        <Song songID="S003" track="A">
            <SongName>Touch Too Much</SongName>
            <Length>4:28</Length>
            <Writer>Angus Young</Writer>
            <Writer>Malcolm Young</Writer>
            <Writer>Brian Johnson</Writer>
            <Rating>7</Rating>
        </Song>
        <Song songID="S004" track="B">
            <SongName>Shot Down in Flames</SongName>
            <Length>3:23</Length>
            <Writer>Angus Young</Writer>
            <Writer>Malcolm Young</Writer>
            <Writer>Brian Johnson</Writer>
            <Rating>8</Rating>
        </Song>
        <Song songID="S005" track="B">
            <SongName>Night Prowler</SongName>
            <Length>6:18</Length>
            <Writer>Angus Young</Writer>
            <Writer>Malcolm Young</Writer>
            <Writer>Brian Johnson</Writer>
            <Rating>8</Rating>
        </Song>
    </Album>

    <Album albumID="A200">
        <AlbumName>Back in Black</AlbumName>
        <Producer>Robert John "Mutt" Lange</Producer>
        <Genres>Hard rock</Genres>
        <Genres>Heavy metal</Genres>
        <Artists artistType="Band">AC/DC</Artists>
        <Recorded>1980</Recorded>
        <Released>
            <Year>1985</Year>
            <Month>July</Month>
            <Day>25</Day>
        </Released>
        <Song songID="S006" track="A">
            <SongName>Hells Bell</SongName>
            <Length>5:12</Length>
            <Writer>Angus Young</Writer>
            <Writer>Malcolm Young</Writer>
            <Writer>Brian Johnson</Writer>
            <Rating>8</Rating>
        </Song>
        <Song songID="S007" track="A">
            <SongName>Shoot to Thrill</SongName>
            <Length>5:17</Length>
            <Writer>Angus Young</Writer>
            <Writer>Malcolm Young</Writer>
            <Writer>Brian Johnson</Writer>
            <Rating>8</Rating>
        </Song>
        <Song songID="S008" track="A">
```

```
            <SongName>Given the Dog a Bone</SongName>
            <Length>3:30</Length>
            <Writer>Angus Young</Writer>
            <Writer>Malcolm Young</Writer>
            <Writer>Brian Johnson</Writer>
            <Rating>8</Rating>
        </Song>
        <Song songID="S009" track="B">
            <SongName>Back in Black</SongName>
            <Length>4:14</Length>
            <Writer>Angus Young</Writer>
            <Writer>Malcolm Young</Writer>
            <Writer>Brian Johnson</Writer>
            <Rating>8</Rating>
        </Song>
        <Song songID="S010" track="B">
            <SongName>You Shook Me All Night Long</SongName>
            <Length>3:30</Length>
            <Writer>Angus Young</Writer>
            <Writer>Malcolm Young</Writer>
            <Writer>Brian Johnson</Writer>
            <Rating>8</Rating>
        </Song>
    </Album>

    <Album albumID="A300">
        <AlbumName>Thriller</AlbumName>
        <Producer>Quincy Jones</Producer>
        <Genres>Pop</Genres>
        <Genres>Post disco</Genres>
        <Genres>Funk</Genres>
        <Artists artistType="Single"> Michael Jackson</Artists>
        <Released>
            <Year>1982</Year>
            <Month>November</Month>
            <Day>30</Day>
        </Released>
        <Song songID="S011" track="A">
            <SongName>Wanna Be Startin' Somethin</SongName>
            <Length>6:02</Length>
            <Writer>Michael Jackson</Writer>
            <Rating>8</Rating>
        </Song>
        <Song songID="S012" track="A">
            <SongName>Thriller</SongName>
            <Awards>Grammy Award for Best Music Video</Awards>
            <Awards>MTV Video Music Award for Best Art Direction</Awards>
            <Length>5:57</Length>
            <Writer>Temperton</Writer>
            <Rating>9</Rating>
        </Song>
        <Song songID="S013" track="B">
            <SongName>Beat It</SongName>
            <Length>4:18</Length>
            <Writer>Michael Jackson</Writer>
            <Rating>9</Rating>
        </Song>
        <Song songID="S014" track="B">
            <SongName>Beat It</SongName>
            <Awards>American Music Award for Favorite Soul/R&amp;B
Video</Awards>
            <Awards>American Music Award for Favorite Pop/Rock Video</Awards>
            <Length>4:18</Length>
            <Writer>Michael Jackson</Writer>
            <Rating>9</Rating>
        </Song>
    </Album>

    <Album albumID="A400">
        <AlbumName>Nevermind</AlbumName>
        <Producer>Butch Vig</Producer>
        <Producer>Nirvana</Producer>
        <Genres>Grunge</Genres>
        <Genres>Alternative rock</Genres>
        <Artists artistType="Band">Nirvana</Artists>
        <Recorded>1990</Recorded>
        <Released>
            <Year>1991</Year>
        </Released>
        <Song songID="S015">
            <SongName>Smells Like Teen Spirit</SongName>
            <Awards>MTV Video Music Award for Best New Artist</Awards>
            <Awards>NME Award for Best Single Ever</Awards>
            <Length>5:01</Length>
            <Writer>Kurt Cobain</Writer>
            <Writer>Krist Novoselic</Writer>
            <Writer>Dave Grohl</Writer>
            <Rating>9</Rating>
        </Song>
        <Song songID="S016">
            <SongName>In Bloom</SongName>
            <Awards>MTV Video Music Award for Best Alternative Video</Awards>
            <Length>4:14</Length>
            <Writer>Kurt Cobain</Writer>
            <Rating>9</Rating>
        </Song>
        <Song songID="S017">
            <SongName>Breed</SongName>
            <Awards>MTV Video Music Award for Best Alternative Video</Awards>
            <Length>3:03</Length>
            <Writer>Kurt Cobain</Writer>
            <Rating>7</Rating>
        </Song>
        <Song songID="S018">
            <SongName>Coms as You Are</SongName>
            <Length>3:39</Length>
            <Writer>Kurt Cobain</Writer>
            <Rating>8</Rating>
        </Song>
    </Album>
</BlackRecords>
```

DTD

DTD stands for Document Type Definition. It defines the legal building blocks of an XML document. It is used to define document structure with a list of legal elements and attributes (JavatPoint, 2018).

We can write DTD in the same file or use DTD from a different file, but we need to specify the location of the file we are using DTD files have .dtd extensions. We declare elements, attributes and entities in a DTD file.

Validation

A valid document includes a document type declaration that identifies the DTD that the document satisfies. * The DTD lists all the elements, attributes, and entities the document uses and the contexts in which it uses them. The DTD may list items the document does not use as well. Validity operates on the principle that everything not permitted is forbidden. Everything in the document must match a declaration in the DTD. If a document has a document type declaration and the document satisfies the DTD that the document type declaration indicates, then the document is said to be valid. If it does not, it is said to be invalid (Means, 2004).

DTD Content

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

<!ELEMENT BlackRecords (StoreName,Telephone+,website,Logo,Address+,Album+)>
<!ATTLIST BlackRecords xmlns:xsi CDATA #REQUIRED
xsi:noNamespaceSchemaLocation CDATA #REQUIRED>

<!ELEMENT StoreName (#PCDATA)>


<!ELEMENT Telephone (#PCDATA)>
<!ATTLIST Telephone phoneType (Ncell|Ntc) #REQUIRED>

<!ELEMENT website (#PCDATA)>

<!ELEMENT Logo (#PCDATA)>


<!ELEMENT Address (Country,City,Street?,HouseNo?)>
<!ATTLIST Address addressType (Owned|Rental) #REQUIRED>

<!ELEMENT Album
(AlbumName,Producer+,Genres+,Artists,Recorded?,Released,Song+)>
<!ATTLIST Album albumID ID #REQUIRED>

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

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

<!ELEMENT Street (#PCDATA)>

<!ELEMENT HouseNo (#PCDATA)>

<!ELEMENT AlbumName (#PCDATA)>

<!ELEMENT Producer (#PCDATA)>

<!ELEMENT Genres (#PCDATA)>

<!ELEMENT Artists (#PCDATA)>
<!ATTLIST Artists artistType (Band|Single) #REQUIRED>

<!ELEMENT Recorded (#PCDATA)>

<!ELEMENT Released (Year,(Month,Day)?)>

<!ELEMENT Song (SongName,Awards*,Length,Writer+,Rating)>
<!ATTLIST Song songID ID #REQUIRED track (A|B) #IMPLIED>

<!ELEMENT Year (#PCDATA)>

<!ELEMENT Month (#PCDATA)>

<!ELEMENT Day (#PCDATA)>

<!ELEMENT SongName (#PCDATA)>

<!ELEMENT Awards (#PCDATA)>

<!ELEMENT Length (#PCDATA)>

<!ELEMENT Writer (#PCDATA)>

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

XML Schema

XML Schemas express shared vocabularies and allow machines to carry out rules made by people. They provide a means for defining the structure, content, and semantics of XML documents. in more detail (W3, 2004).

We can apply schema .xsd file to our xml document from the root tag.

The <schema> element is the root element within an XML Schema, and it enables you to declare namespace information as well as defaults for declarations throughout the document. You can also include a version attribute that helps to identify the XML Schema and the version of your vocabulary (Joe Fawcett, 2012).

Schemas also have data types which we can use to give more clarity to our documents. The most used data types are string, token, byte, short, long, integer etc.

XML Schema Content

Schema for the coursework is presented below:

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

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="BlackRecords">
    <xs:complexType>
      <xs:sequence>
        <xs:element type="xs:string" name="StoreName"/>
        <xs:element name="Telephone" maxOccurs="unbounded" minOccurs="1">
          <xs:complexType>
            <xs:simpleContent>
              <xs:extension base="xs:long">
                <xs:attribute type="xs:string" name="phoneType"
use="required"/>
              </xs:extension>
            </xs:simpleContent>
          </xs:complexType>
        </xs:element>
        <xs:element type="xs:string" name="website"/>
        <xs:element type="xs:string" name="Logo"/>
        <xs:element name="Address" maxOccurs="unbounded" minOccurs="1">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="Country"/>
              <xs:element type="xs:string" name="City"/>
              <xs:element type="xs:string" name="Street" minOccurs="0"/>
              <xs:element type="xs:byte" name="HouseNo" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute type="xs:string" name="addressType"
use="required"/>
          </xs:complexType>
        </xs:element>
        <xs:element name="Album" maxOccurs="unbounded" minOccurs="1">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="AlbumName"/>
              <xs:element type="xs:string" name="Producer"
maxOccurs="unbounded" minOccurs="1"/>
              <xs:element type="xs:string" name="Genres"
maxOccurs="unbounded" minOccurs="1"/>
              <xs:element name="Artists">
                <xs:complexType>
                  <xs:simpleContent>
                    <xs:extension base="xs:string">
                      <xs:attribute type="xs:string" name="artistType"
use="required"/>
                    </xs:extension>
                  </xs:simpleContent>
                </xs:complexType>
              </xs:element>
              <xs:element type="xs:short" name="Recorded" minOccurs="0"/>
              <xs:element name="Released">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:short" name="Year"/>
                    <xs:element type="xs:string" name="Month" minOccurs="0"/>
                    <xs:element type="xs:byte" name="Day" minOccurs="0"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="Song" maxOccurs="unbounded" minOccurs="0">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:string" name="SongName"/>
                    <xs:element type="xs:string" name="Awards"
maxOccurs="unbounded" minOccurs="0"/>
                    <xs:element type="xs:string" name="Length"/>
                    <xs:element type="xs:string" name="Writer"
maxOccurs="unbounded" minOccurs="1"/>
                    <xs:element type="xs:byte" name="Rating"/>
                  </xs:sequence>
                  <xs:attribute type="xs:string" name="songID"
use="required"/>
                  <xs:attribute type="xs:string" name="track"
use="optional"/>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
            <xs:attribute type="xs:string" name="albumID" use="required"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

CSS

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media CSS is among the core languages of the open web and is standardized across Web browsers according to W3C specifications (Mozilla, 2021).

CSS Content

The CSS used in this coursework is provided below.

```
BlackRecords {
    width: 100%;
    background: linear-gradient(-45deg, #ee7752, #e73c7e, #23a6d5, #23d5ab);
    background-size: 500% 500%;
    animation: gradient 15s ease infinite;
    display: flex;
    flex-direction: column;
    flex-wrap: wrap;
    justify-content: space-evenly;
    align-items: stretch;
}

@keyframes gradient {
    0% {
        background-position: 0% 50%;
    }
    50% {
        background-position: 100% 50%;
    }
    100% {
        background-position: 0% 50%;
    }
}

StoreName {
    order: 0;
    background-image: url(music-regular-96white.png);
    background-repeat: no-repeat;
    flex-grow: 1;
    flex-shrink: 1;
    width: 100%;
    height: 90px;
    color: blanchedalmond;
    text-decoration: overline underline;
    text-align: center;
    font-size: 33px;
    font-family: 'Calibri', Times, serif;
    padding-top: 20px;
}
```

```
Album {
    display: flex;
    flex-direction: column;
    flex-wrap: nowrap;
    justify-content: space-evenly;
    align-items: baseline;
    padding: 20px;
    flex-grow: 1;
    flex-shrink: 1;
    border: 3px black solid;
}

Producer::before {
    content: "Producer:- ";
}

Recorded::before {
    content: "Recorded:- ";
}

Released::before {
    content: "Released:- ";
}

Artists {
    font-size: 20px;
    padding-top: 7px;
    padding-bottom: 7px;
    font-family: 'Brush Script MT';
}

AlbumName {
    font-size: 32px;
    font-family: Helvetica, Arial, sans-serif;
    padding-bottom: 12px;
    text-decoration: underline;
}

Awards {
    display: block;
    font-style: italic;
}

Song {
    padding: 5px;
}

SongName {
    display: block;
    font-family: 'Courier New', Courier, monospace;
    color: beige;
}

Rating {
    font-size: large;
}
```

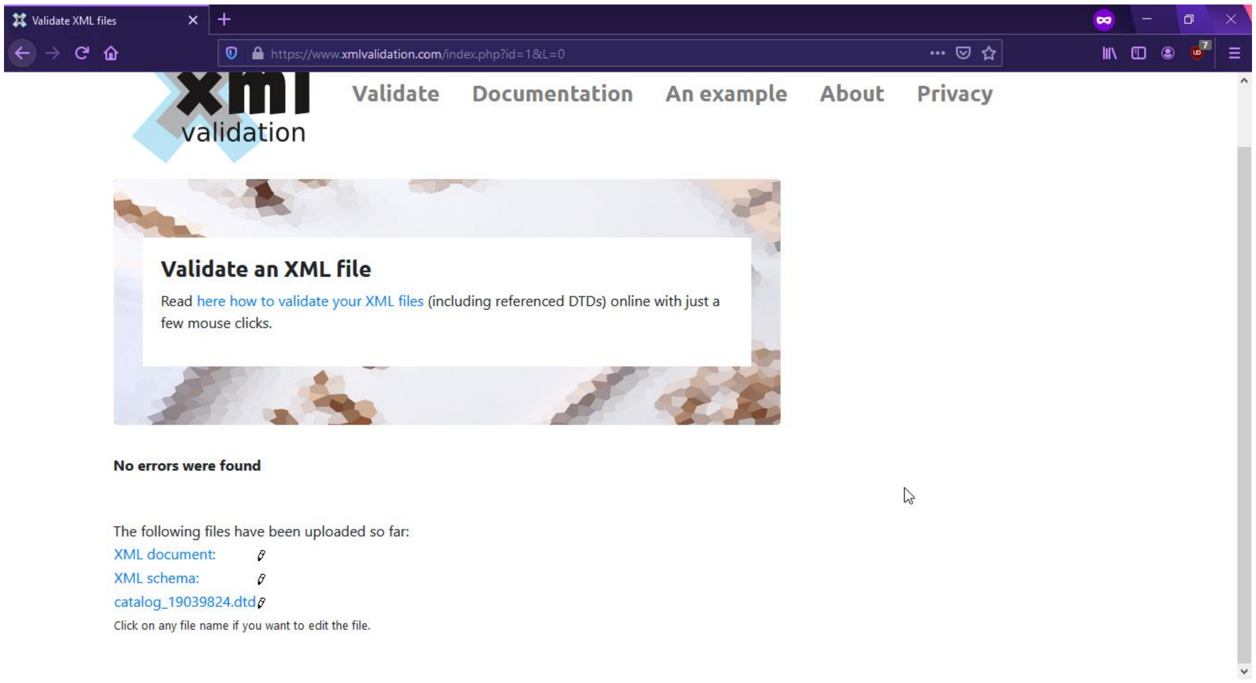
```
Length {
    font-family: monospace;
    font-size: 17px;
    color: bisque;
}

Logo {
    order: 1;
    padding: 10px;
}

Logo::after {
    content: " Contacts and Locations:";
}

Telephone {
    padding: 10px;
    order: 2;
}

website {
    order: 3;
    padding: 10px;
}

Address {
    order: 4;
    padding: 10px;
}
```

Testing

Test 1 Validating XML

Objective:	To validate XML files.
Action:	Online validation tool xmlvalidation.com was used to validate the xml, dtd and schema file.
Expected result:	There should be no errors when validating.
Actual result:	There were no errors.
Conclusion:	Test is successful.

Table 1: Test 1



Validate an XML file

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

No errors were found

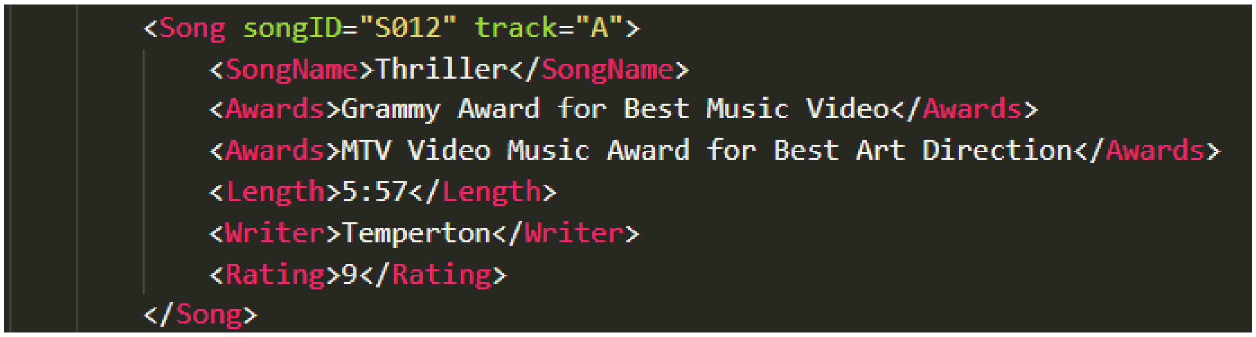
The following files have been uploaded so far:
XML document:
XML schema:
catalog_19039824.dtd
Click on any file name if you want to edit the file.

Figure 2: Validation Successful

Test 2 Validating optional elements and attributes

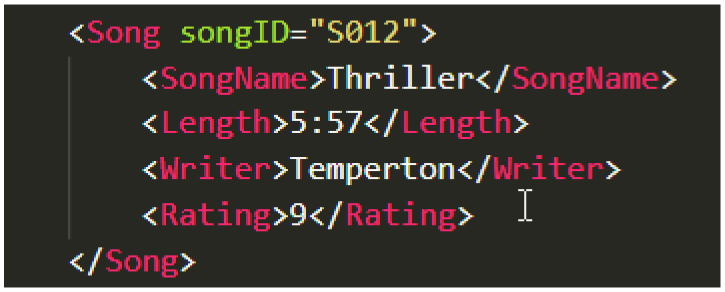
Objective:	To validate XML files by removing optional elements and attributes.
Action:	Validation was done after removing some elements.
Expected result:	There should be no errors when validating.
Actual result:	There were no errors.
Conclusion:	Test is successful.

Table 2: Test 2



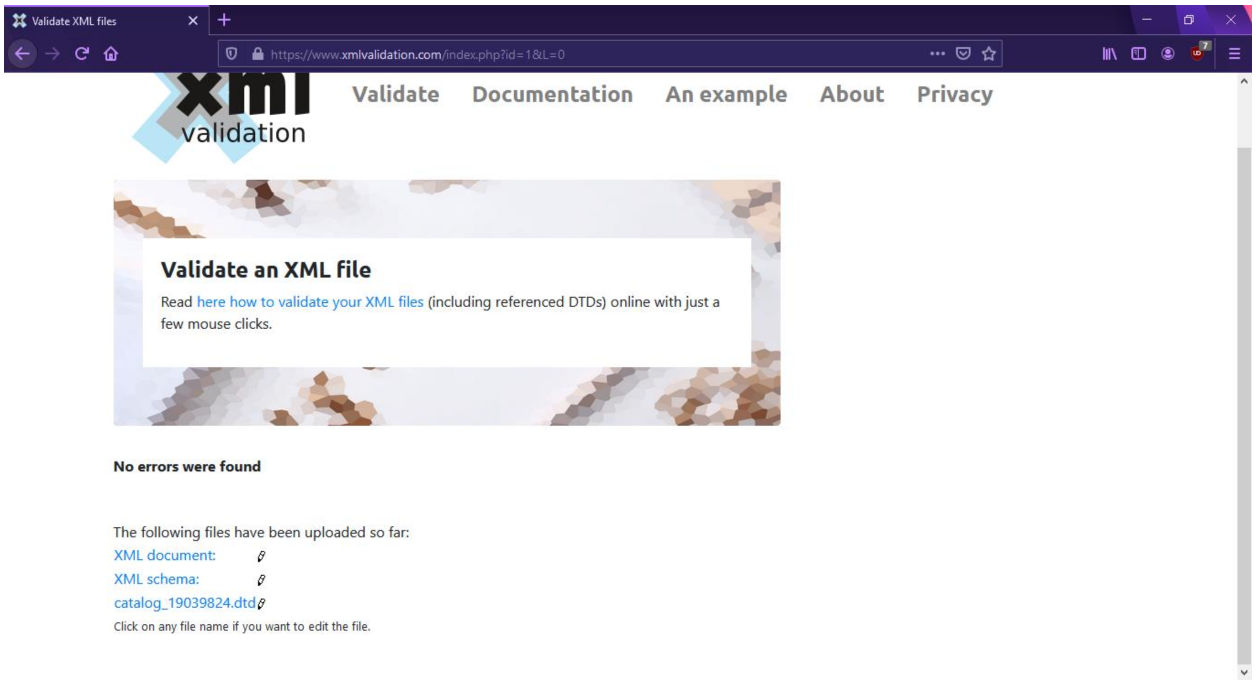
```
<Song songID="S012" track="A">
    <SongName>Thriller</SongName>
    <Awards>Grammy Award for Best Music Video</Awards>
    <Awards>MTV Video Music Award for Best Art Direction</Awards>
    <Length>5:57</Length>
    <Writer>Temperton</Writer>
    <Rating>9</Rating>
</Song>
```

Figure 3: Original:



```
<Song songID="S012">
    <SongName>Thriller</SongName>
    <Length>5:57</Length>
    <Writer>Temperton</Writer>
    <Rating>9</Rating>
</Song>
```

Figure 4: Some elements and attributes removed



Validate an XML file

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

No errors were found

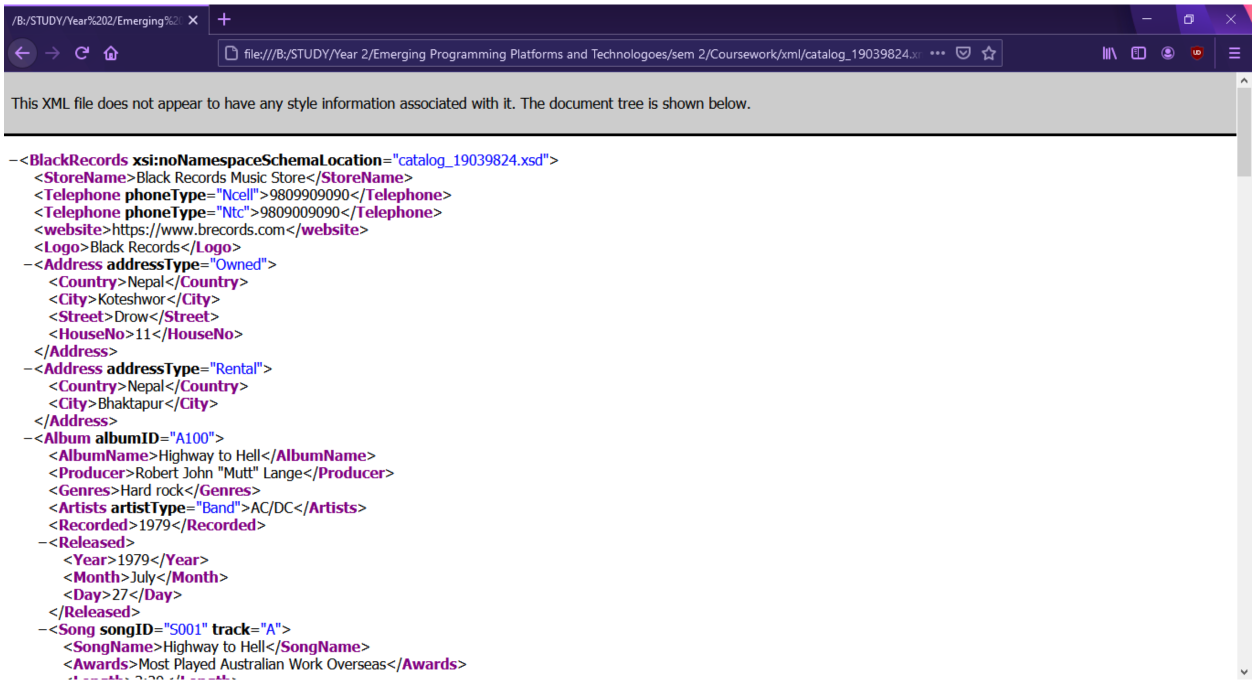
The following files have been uploaded so far:
XML document:
XML schema:
catalog_19039824.dtd
Click on any file name if you want to edit the file.

Figure 5: Validation after removing optional elements

Test 3 Opening the file without and with CSS

Objective:	To see if CSS makes difference.
Action:	The file was opened without CSS and with CSS.
Expected result:	The browser should say the document has no style associated with it when opening without CSS and should display the XML document as in text editor, when CSS is applied, the XML document should be displayed as specified in the CSS.
Actual result:	The document was displayed as in text editor without CSS, with CSS it was displayed with some formatting and style.
Conclusion:	Test is successful.

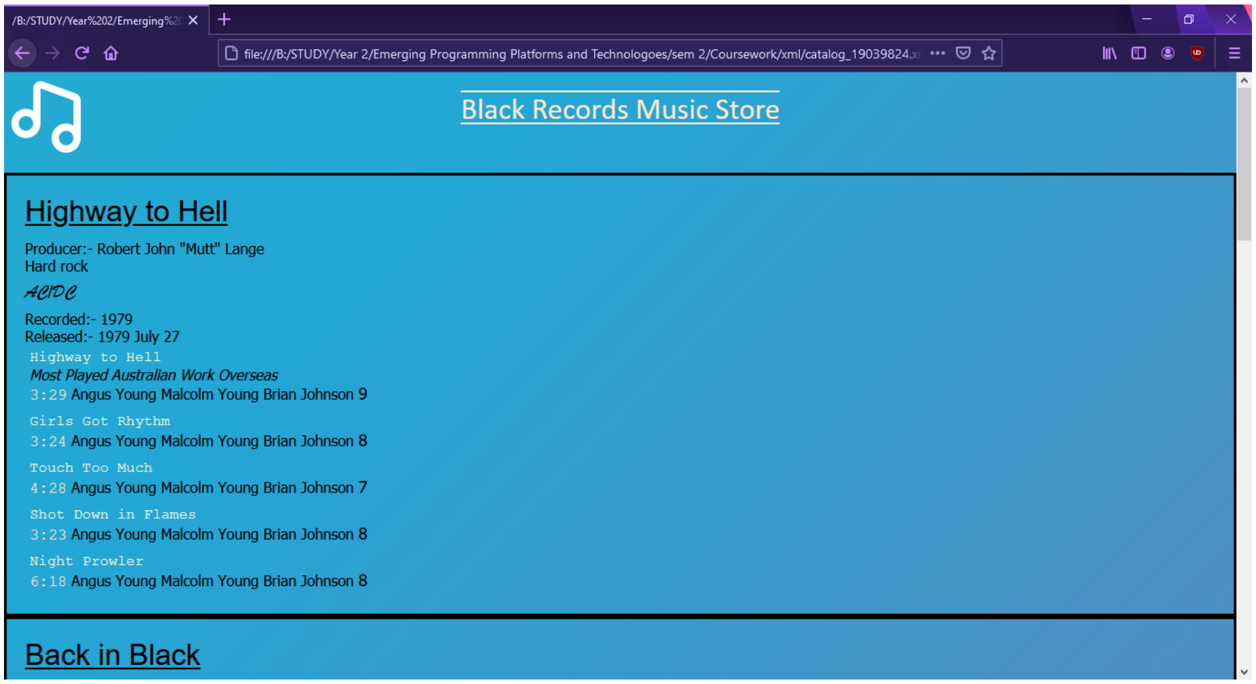
Table 3: Test 3


/B:/STUDY/Year%202/Emerging%2
file:///B:/STUDY/Year 2/Emerging Programming Platforms and Technologoes/sem 2/Coursework/xml/catalog_19039824.x

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

```
−<BlackRecords xsi:noNamespaceSchemaLocation="catalog_19039824.xsd">
    <StoreName>Black Records Music Store</StoreName>
    <Telephone phoneType="Ncell">9809909090</Telephone>
    <Telephone phoneType="Ntc">9809009090</Telephone>
    <website>https://www.brecords.com</website>
    <Logo>Black Records</Logo>
  −<Address addressType="Owned">
      <Country>Nepal</Country>
      <City>Koteshwor</City>
      <Street>Drow</Street>
      <HouseNo>11</HouseNo>
    </Address>
  −<Address addressType="Rental">
      <Country>Nepal</Country>
      <City>Bhaktapur</City>
    </Address>
  −<Album albumID="A100">
      <AlbumName>Highway to Hell</AlbumName>
      <Producer>Robert John "Mutt" Lange</Producer>
      <Genres>Hard rock</Genres>
      <Artists artistType="Band">AC/DC</Artists>
      <Recorded>1979</Recorded>
    −<Released>
        <Year>1979</Year>
        <Month>July</Month>
        <Day>27</Day>
      </Released>
    −<Song songID="S001" track="A">
        <SongName>Highway to Hell</SongName>
        <Awards>Most Played Australian Work Overseas</Awards>
```

Figure 6: Without CSS


/B:/STUDY/Year%202/Emerging%2
file:///B:/STUDY/Year 2/Emerging Programming Platforms and Technologoes/sem 2/Coursework/xml/catalog_19039824.x

Black Records Music Store

Highway to Hell

Producer:- Robert John "Mutt" Lange
Hard rock
ACDC
Recorded:- 1979
Released:- 1979 July 27
Highway to Hell
Most Played Australian Work Overseas
3:29 Angus Young Malcolm Young Brian Johnson 9
Girls Got Rhythm
3:24 Angus Young Malcolm Young Brian Johnson 8
Touch Too Much
4:28 Angus Young Malcolm Young Brian Johnson 7
Shot Down in Flames
3:23 Angus Young Malcolm Young Brian Johnson 8
Night Prowler
6:18 Angus Young Malcolm Young Brian Johnson 8

Back in Black

Figure 7: With CSS

NOTE:(when the CSS is applied, for information, the album name is in top and underlined, followed by producer and genre then the artist name, the white colored title represent song name, italics represent award, the whitish numbers represent song length, the names after the length represent writers and the slightly large numbers represent the ratings.)

Test 4 Checking animation

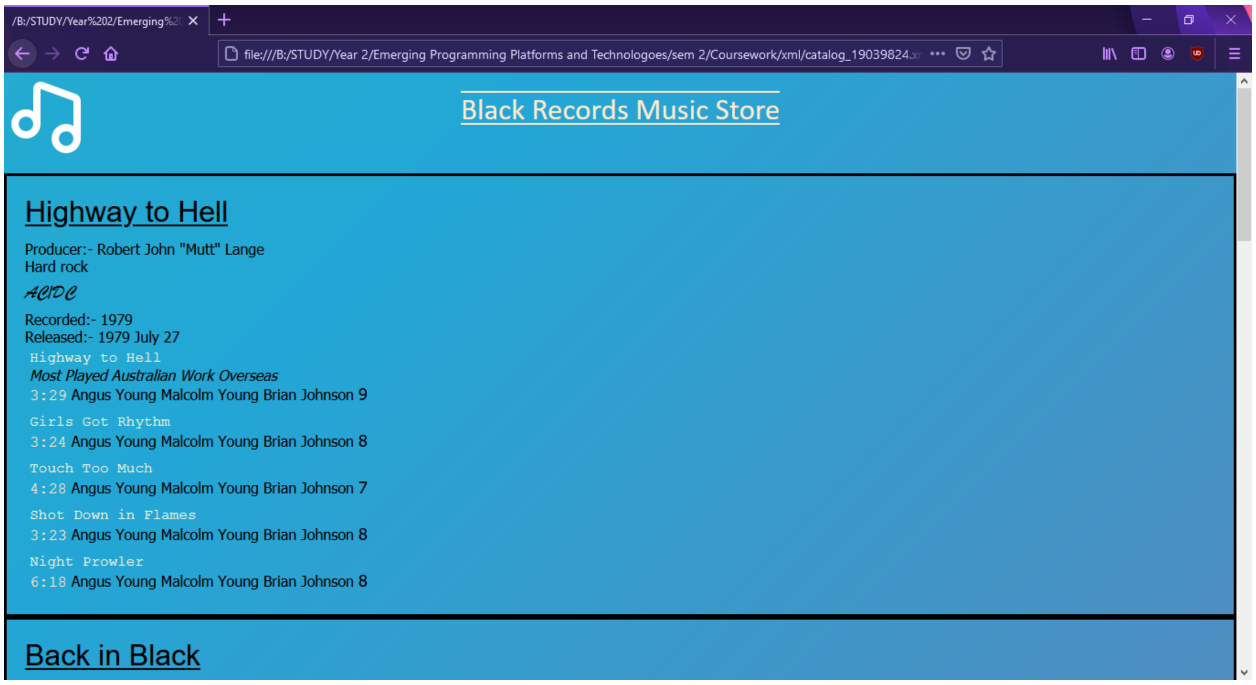
Objective:	To see if animation works.
Action:	The file was opened in a browser that supports it.
Expected result:	The background color should change to as time passes.
Actual result:	The background color kept changing.
Conclusion:	Test is successful.

Table 4: Test 4:

The following CSS was used

```
    background: linear-gradient(-45deg, #ee7752, #e73c7e, #23a6d5, #23d5ab);
    background-size: 500% 500%;
    animation: gradient 15s ease infinite;
@keyframes gradient {
    0% {
        background-position: 0% 50%;
    }
    50% {
        background-position: 100% 50%;
    }
    100% {
        background-position: 0% 50%;
    }
}
```

The background is set to gradient of different colors and animation name, time, duration etc. is defined then it is specified how to move.



file:///B:/STUDY/Year 2/Emerging Programming Platforms and Technologoes/sem 2/Coursework/xml/catalog_19039824.x

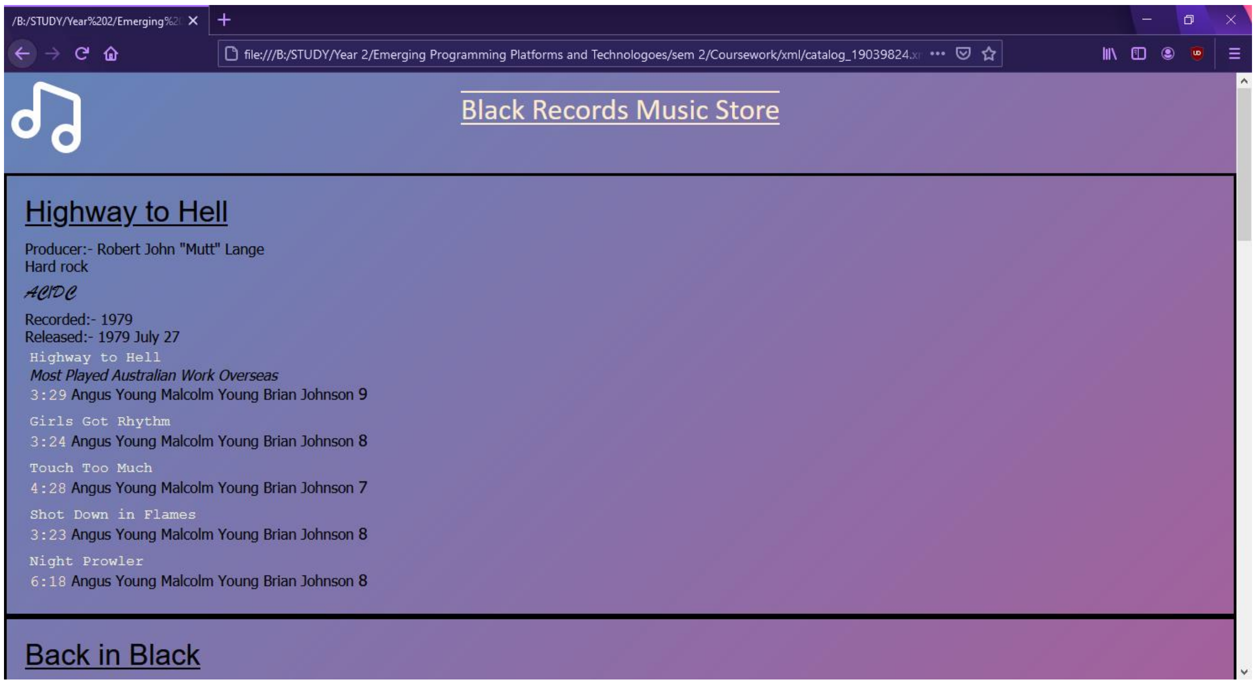
Black Records Music Store

Highway to Hell

Producer:- Robert John "Mutt" Lange
Hard rock
ACDC
Recorded:- 1979
Released:- 1979 July 27
Highway to Hell
Most Played Australian Work Overseas
3:29 Angus Young Malcolm Young Brian Johnson 9
Girls Got Rhythm
3:24 Angus Young Malcolm Young Brian Johnson 8
Touch Too Much
4:28 Angus Young Malcolm Young Brian Johnson 7
Shot Down in Flames
3:23 Angus Young Malcolm Young Brian Johnson 8
Night Prowler
6:18 Angus Young Malcolm Young Brian Johnson 8

Back in Black

Figure 8: At the beginning



file:///B:/STUDY/Year 2/Emerging Programming Platforms and Technologoes/sem 2/Coursework/xml/catalog_19039824.x

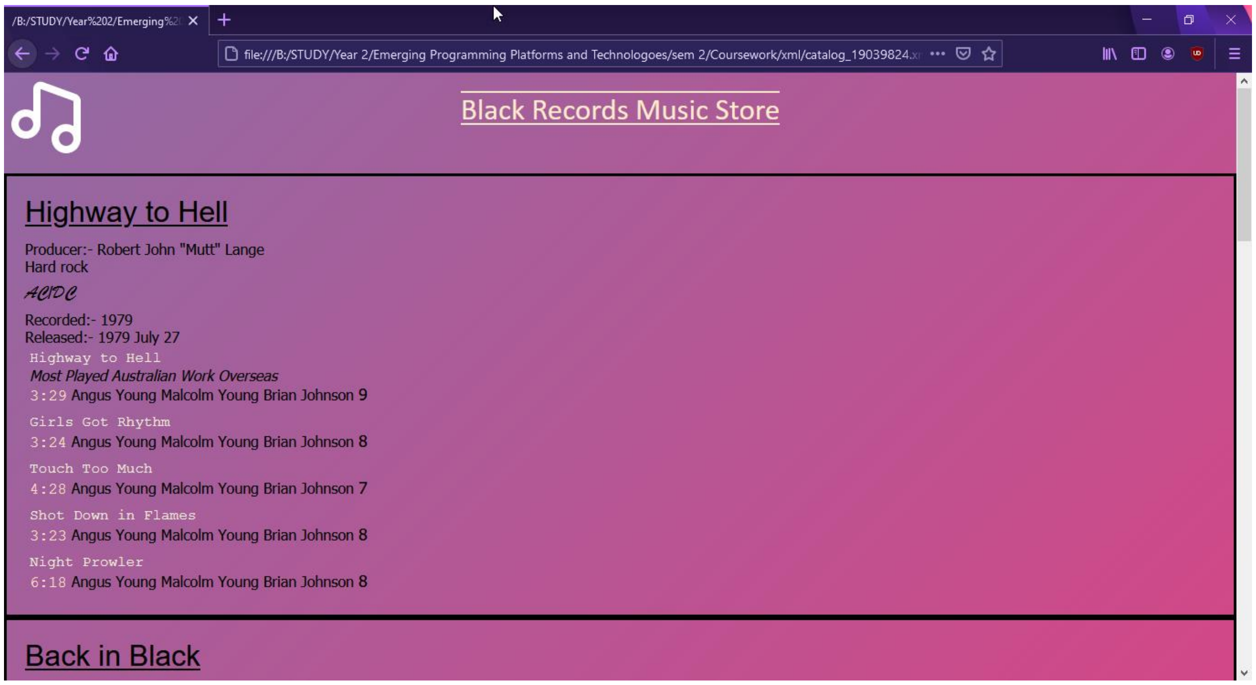
Black Records Music Store

Highway to Hell

Producer:- Robert John "Mutt" Lange
Hard rock
ACDC
Recorded:- 1979
Released:- 1979 July 27
Highway to Hell
Most Played Australian Work Overseas
3:29 Angus Young Malcolm Young Brian Johnson 9
Girls Got Rhythm
3:24 Angus Young Malcolm Young Brian Johnson 8
Touch Too Much
4:28 Angus Young Malcolm Young Brian Johnson 7
Shot Down in Flames
3:23 Angus Young Malcolm Young Brian Johnson 8
Night Prowler
6:18 Angus Young Malcolm Young Brian Johnson 8

Back in Black

Figure 9: After some time 1


/B:/STUDY/Year%202/Emerging%2
file:///B:/STUDY/Year 2/Emerging Programming Platforms and Technologoes/sem 2/Coursework/xml/catalog_19039824.x

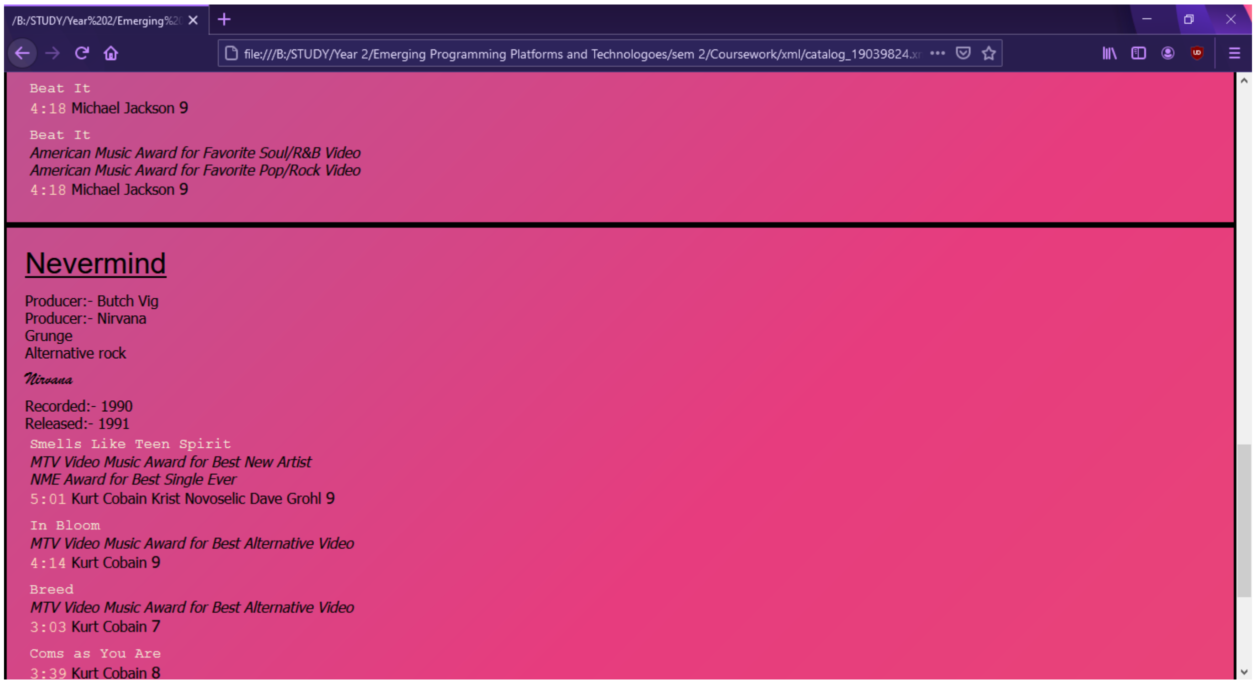
Black Records Music Store

Highway to Hell

Producer:- Robert John "Mutt" Lange
Hard rock
ACDC
Recorded:- 1979
Released:- 1979 July 27
Highway to Hell
Most Played Australian Work Overseas
3:29 Angus Young Malcolm Young Brian Johnson 9
Girls Got Rhythm
3:24 Angus Young Malcolm Young Brian Johnson 8
Touch Too Much
4:28 Angus Young Malcolm Young Brian Johnson 7
Shot Down in Flames
3:23 Angus Young Malcolm Young Brian Johnson 8
Night Prowler
6:18 Angus Young Malcolm Young Brian Johnson 8

Back in Black

Figure 10: After some time 2


/B:/STUDY/Year%202/Emerging%2
file:///B:/STUDY/Year 2/Emerging Programming Platforms and Technologoes/sem 2/Coursework/xml/catalog_19039824.x

Beat It
4:18 Michael Jackson 9
Beat It
American Music Award for Favorite Soul/R&B Video
American Music Award for Favorite Pop/Rock Video
4:18 Michael Jackson 9

Nevermind

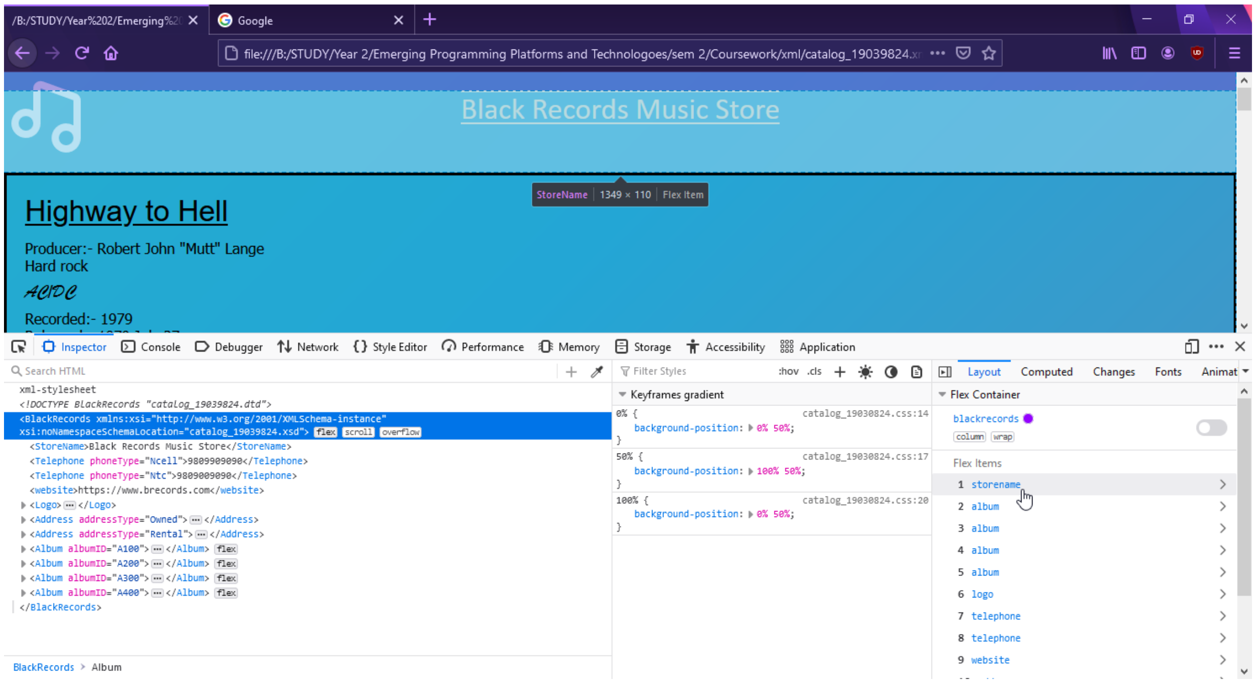
Producer:- Butch Vig
Producer:- Nirvana
Grunge
Alternative rock
Nirvana
Recorded:- 1990
Released:- 1991
Smells Like Teen Spirit
MTV Video Music Award for Best New Artist
NME Award for Best Single Ever
5:01 Kurt Cobain Krist Novoselic Dave Grohl 9
In Bloom
MTV Video Music Award for Best Alternative Video
4:14 Kurt Cobain 9
Breed
MTV Video Music Award for Best Alternative Video
3:03 Kurt Cobain 7
Coms as You Are
3:39 Kurt Cobain 8

Figure 11: After some time 3

Test 5 Flex box

Objective:	To see flex box is used.
Action:	The file was opened in browser and inspected.
Expected result:	Flex items should be visible.
Actual result:	Flex items were visible.
Conclusion:	Test is successful.

Table 5: Test 5


/B:/STUDY/Year%202/Emerging%2
Google
file:///B:/STUDY/Year 2/Emerging Programming Platforms and Technologoes/sem 2/Coursework/xml/catalog_19039824.x
Black Records Music Store
StoreName 1349 × 110 Flex Item
Highway to Hell
Producer:- Robert John "Mutt" Lange
Hard rock
ACDC
Recorded:- 1979
Inspector
Console
Debugger
Network
Style Editor
Performance
Memory
Storage
Accessibility
Application
Search HTML
Filter Styles
Layout
Computed
Changes
Fonts
Animat
BlackRecords > Album

```
xml-stylesheet
<!DOCTYPE BlackRecords "catalog_19039824.dtd">
<BlackRecords xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="catalog_19039824.xsd"> flex scroll overflow
  <StoreName>Black Records Music Store</StoreName>
  <Telephone phoneType="Ncell">9809909090</Telephone>
  <Telephone phoneType="Ntc">9809009090</Telephone>
  <website>https://www.brecords.com</website>
 ▸<Logo> ⋯ </Logo>
 ▸<Address addressType="Owned"> ⋯ </Address>
 ▸<Address addressType="Rental"> ⋯ </Address>
 ▸<Album albumID="A100"> ⋯ </Album> flex
 ▸<Album albumID="A200"> ⋯ </Album> flex
 ▸<Album albumID="A300"> ⋯ </Album> flex
 ▸<Album albumID="A400"> ⋯ </Album> flex
</BlackRecords>
```

```
Keyframes gradient
0% {                                  catalog_19030824.css:14
  background-position: ▸ 0% 50%;
}
50% {                                 catalog_19030824.css:17
  background-position: ▸ 100% 50%;
}
100% {                                catalog_19030824.css:20
  background-position: ▸ 0% 50%;
}
```

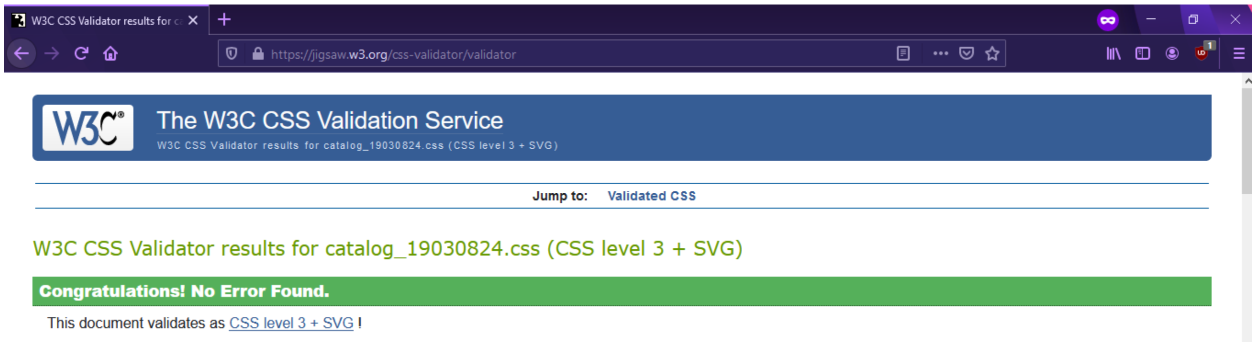
```
Flex Container
blackrecords
column wrap
Flex Items
1 storename
2 album
3 album
4 album
5 album
6 logo
7 telephone
8 telephone
9 website
```

Figure 12: Flex items visible

Test 6 CSS validation

Objective:	To see if there are errors in CSS.
Action:	The file was tested online in CSS validator.
Expected result:	There should be no errors.
Actual result:	There were no errors.
Conclusion:	Test is successful.

Table 6: Test 6


W3C CSS Validator results for c
https://jigsaw.w3.org/css-validator/validator
Jump to: Validated CSS

The W3C CSS Validation Service
W3C CSS Validator results for catalog_19030824.css (CSS level 3 + SVG)

W3C CSS Validator results for catalog_19030824.css (CSS level 3 + SVG)

Congratulations! No Error Found.

This document validates as CSS level 3 + SVG !

Figure 13: CSS is valid

DTD vs Schema

Both DTD and Schema are used to validate our XML documents but there are still differences between them.

- XML Schemas are created using basic XML, whereas DTDs utilize a separate syntax (Joe Fawcett, 2012).
- XML Schemas fully support the Namespace Recommendation (Joe Fawcett, 2012).
- XML Schemas enable you to validate text element content based on built-in and user-defined data types (Joe Fawcett, 2012).
- XML Schemas enable you to create complex and reusable content models more easily (Joe Fawcett, 2012).
- XML Schemas enable the modeling of programming concepts such as object inheritance and type substitution (Joe Fawcett, 2012).

DTDs do not provide fine control over the format and data types of element and attribute values. Other than the various special attribute types (ID, IDREF, ENTITY, NMTOKEN, and so forth), once an element or attribute has been declared to contain character data, no limits may be placed on the length, type, or format of that content (Means, 2004).

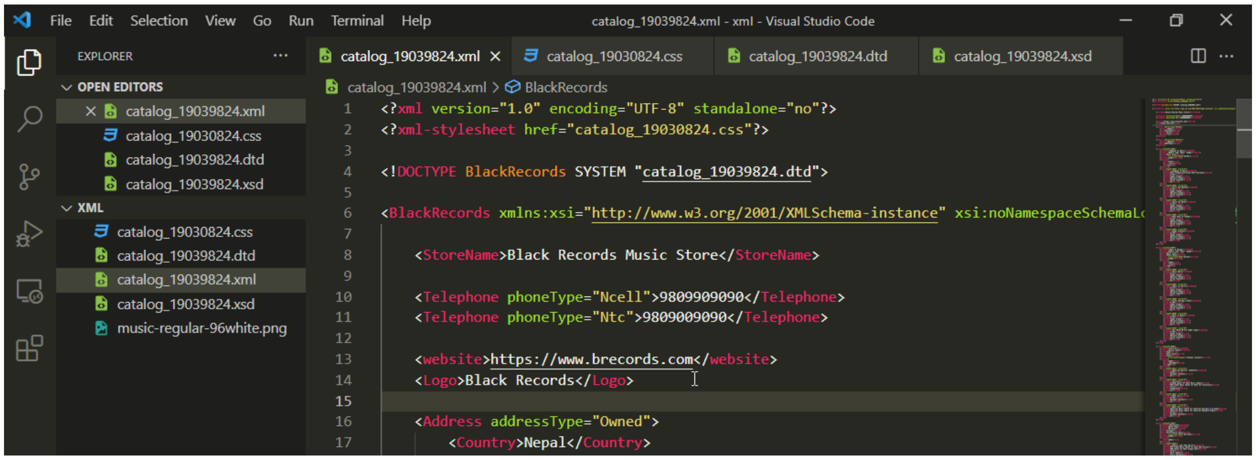
Schemas can enforce much more specific rules about the contents of elements and attributes than DTDs can. In addition to a wide range of built-in simple types (such as string, integer, decimal ,and date Time), the schema language provides a framework for declaring new data types, deriving new types from old types, and reusing types from other schemas (Means, 2004).

schemas can place more explicit restrictions on the number and sequence of child elements that can appear in a given location. This is even true when elements are mixed with character data, unlike the mixed content supported by DTDs (Means, 2004).

Development

For the development, completely free software and tools were used. For tree diagram, https://app.diagrams.net/ was used, this website allows us to create diagrams for free, for validating xml, xsd, dtd and css documents, https://www.xmlvalidation.com/ and https://jigsaw.w3.org/css-validator/#validate_by_upload were used respectively.

To develop all the documents, Visual Studio Code was used, it is a free lightweight source code editor that provides a lot of useful features like code highlighting, formatting and plugins and extensions to make workflow easy and efficient. Visual Studio Code can be downloaded here https://code.visualstudio.com/Download.

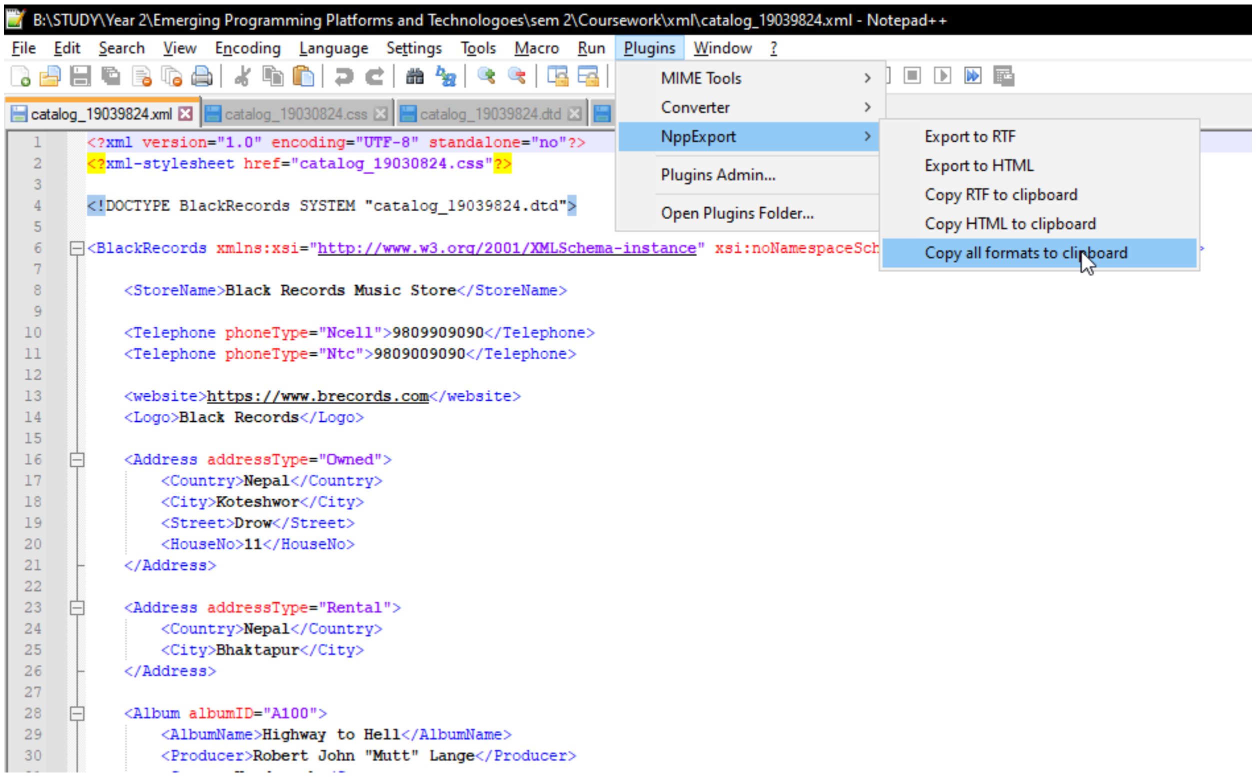


```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<?xml-stylesheet href="catalog_19030824.css"?>

<!DOCTYPE BlackRecords SYSTEM "catalog_19039824.dtd">

<BlackRecords xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaL

    <StoreName>Black Records Music Store</StoreName>

    <Telephone phoneType="Ncell">9809909090</Telephone>
    <Telephone phoneType="Ntc">9809009090</Telephone>

    <website>https://www.brecords.com</website>
    <Logo>Black Records</Logo>

    <Address addressType="Owned">
        <Country>Nepal</Country>
```

Figure 14: Using Visual Studio Code

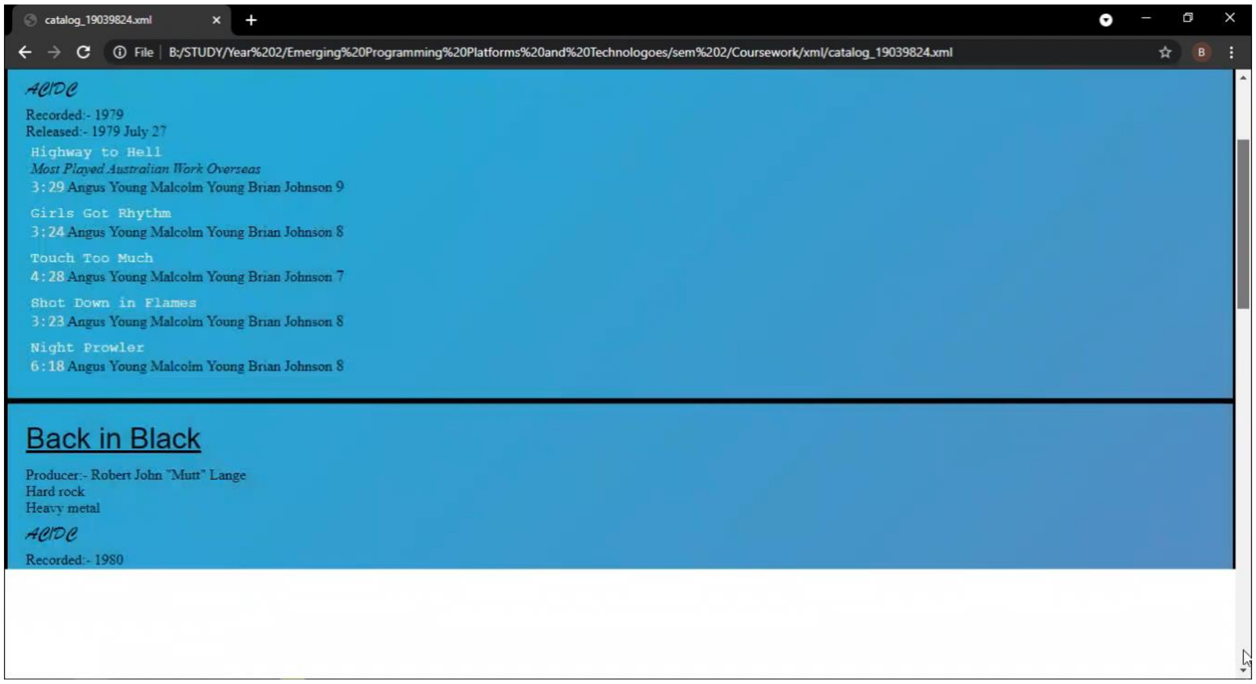
Notepad++ was also, it is also a source code editor, it was used to format and copy code to word document. Notepad++ can be found here https://notepad-plus-plus.org/downloads/.

Firefox browser was used to view the xml file, as in my experience, I found, my css worked best in Firefox. The browser can be found here https://www.mozilla.org/en-US/firefox/all/#product-desktop-release.


B:\STUDY\Year 2\Emerging Programming Platforms and Technologoes\sem 2\Coursework\xml\catalog_19039824.xml - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
catalog_19039824.xml
catalog_19030824.css
catalog_19039824.dtd
MIME Tools
Converter
NppExport
Plugins Admin...
Open Plugins Folder...
Export to RTF
Export to HTML
Copy RTF to clipboard
Copy HTML to clipboard
Copy all formats to clipboard

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<?xml-stylesheet href="catalog_19030824.css"?>

<!DOCTYPE BlackRecords SYSTEM "catalog_19039824.dtd">

<BlackRecords xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSch

    <StoreName>Black Records Music Store</StoreName>

    <Telephone phoneType="Ncell">9809909090</Telephone>
    <Telephone phoneType="Ntc">9809009090</Telephone>

    <website>https://www.brecords.com</website>
    <Logo>Black Records</Logo>

    <Address addressType="Owned">
        <Country>Nepal</Country>
        <City>Koteshwor</City>
        <Street>Drow</Street>
        <HouseNo>11</HouseNo>
    </Address>

    <Address addressType="Rental">
        <Country>Nepal</Country>
        <City>Bhaktapur</City>
    </Address>

    <Album albumID="A100">
        <AlbumName>Highway to Hell</AlbumName>
        <Producer>Robert John "Mutt" Lange</Producer>
```

Figure 15: Using Notepad++ to copy code.


catalog_19039824.xml
File | B:/STUDY/Year%202/Emerging%20Programming%20Platforms%20and%20Technologoes/sem%202/Coursework/xml/catalog_19039824.xml

AC/DC
Recorded:- 1979
Released:- 1979 July 27
Highway to Hell
Most Played Australian Work Overseas
3:29 Angus Young Malcolm Young Brian Johnson 9
Girls Got Rhythm
3:24 Angus Young Malcolm Young Brian Johnson 8
Touch Too Much
4:28 Angus Young Malcolm Young Brian Johnson 7
Shot Down in Flames
3:23 Angus Young Malcolm Young Brian Johnson 8
Night Prowler
6:18 Angus Young Malcolm Young Brian Johnson 8

Back in Black
Producer:- Robert John "Mutt" Lange
Hard rock
Heavy metal
AC/DC
Recorded:- 1980

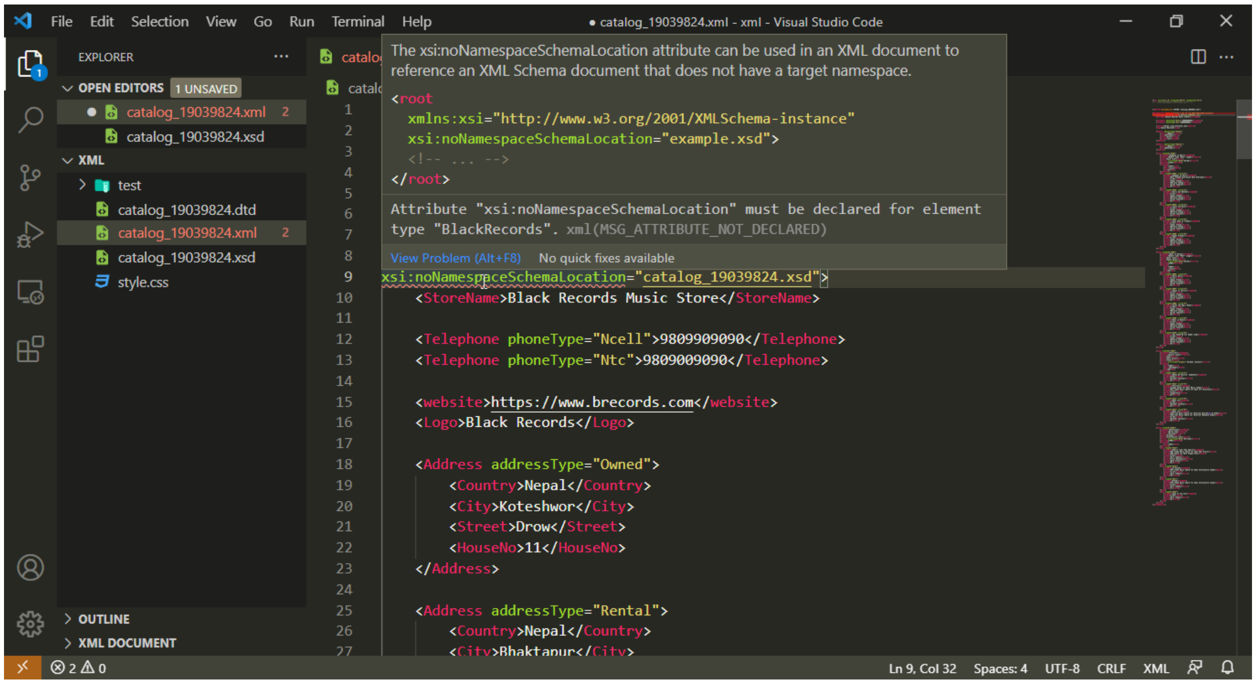
Figure 16: Scrolling very fast in chrome

I encountered problems like in the figure above where the content would not be visible immediately when scrolling rapidly in Chrome browser but I did not face such problems in Firefox browser so, I used Firefox browser during testing and development. Here is a small clip showing the effect.

https://drive.google.com/file/d/105Qvc8wIaOAqC5gXdTJqY9E-SG_PldBK/view?usp=sharing.

Critical Analysis

The development phase of the coursework was mostly a hassle-free process, the xml, dtd, xsd and css documents were easily developed but I did fac problems when adding schema to my xml document. I encountered the following error.



The xsi:noNamespaceSchemaLocation attribute can be used in an XML document to reference an XML Schema document that does not have a target namespace.

```
<root
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="example.xsd">
  <!-- ... -->
</root>
```

```
Attribute "xsi:noNamespaceSchemaLocation" must be declared for element
type "BlackRecords". xml(MSG_ATTRIBUTE_NOT_DECLARED)
```

View Problem (Alt+F8) No quick fixes available

```
xsi:noNamespaceSchemaLocation="catalog_19039824.xsd">
    <StoreName>Black Records Music Store</StoreName>

    <Telephone phoneType="Ncell">9809909090</Telephone>
    <Telephone phoneType="Ntc">9809009090</Telephone>

    <website>https://www.brecords.com</website>
    <Logo>Black Records</Logo>

    <Address addressType="Owned">
        <Country>Nepal</Country>
        <City>Koteshwor</City>
        <Street>Drow</Street>
        <HouseNo>11</HouseNo>
    </Address>

    <Address addressType="Rental">
        <Country>Nepal</Country>
        <City>Bhaktapur</City>
```

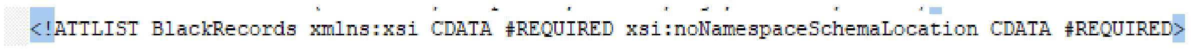
Figure 17: Error encountered

I read the error message and I had no idea what it meant, as during classes, I had not encountered the error. I saw the example in the error and compared to my code, it seemed correct to me.

The first thing I tried was use a different editor, I used Notepad++, it did not show errors, but when validating online, it showed the same error again. I thought that dtd is causing the error, so I removed the dtd document and it did not show error, but since the coursework requires dtd, I added the dtd document back and error occurred again.

Now, I knew that there must be something wrong with dtd, I searched online and found somewhere that these types of error occur when there is conflict in dtd and schema, but my dtd and schema were correct when validating separately, so I again checked the way I implemented schema to my xml.

I had added the xsd from root tag but, I has not added dtd for root tag, this had caused the error for me, after adding the following line to my dtd document, the error was solved.



```
<!ATTLIST BlackRecords xmlns:xsi CDATA #REQUIRED xsi:noNamespaceSchemaLocation CDATA #REQUIRED>
```

Figure 18: Correcting error

I learned from this error, that nothing should be neglected as even a small thing can cause a lot of errors. I also learned that there should be a process to do any task especially find and identify errors, haphazardly typing the error message does not solve the error, we must have understanding of how the system works and what causes the error to solve the error, there should be a rational method.

I also learnt that some tools are better for certain tasks and learnt to use the proper tools of the tools that provided best outcomes during the development phase of the coursework.

Conclusion

The coursework has been completed and it was a great learning opportunity, not only did I learn new things but also got revised on older topics already learnt in class. The current situation of COVID did not allow us to be at college to do coursework by interacting with friends and teachers but the video materials provided were huge help to complete the coursework on time.

References

Guru99. (2021) *Guru99* [Online]. Available from: https://www.guru99.com/xml-tutorials.html [Accessed 5 May 2021].

JavatPoint. (2018) *JavatPoint* [Online]. Available from: https://www.javatpoint.com/xml-dtd [Accessed 5 May 2021].

Joe Fawcett, L.Q.D.A. (2012) *Beginning XML*. 5th ed. Indianapolis: John Wiley & Sons, Inc.

Means, E.R.H.a.W.S. (2004) *XML in a Nutshell*. 3rd ed. Sebastopol, CA: O'ReillyMedia,Inc.

Mozilla. (2021) *MDN Web Docs* [Online]. Available from: https://developer.mozilla.org/en-US/docs/Web/CSS [Accessed 5 May 2021].

W3. (2004) *W3* [Online]. Available from: https://www.w3.org/XML/Schema.html [Accessed 5 May 2021].

W3Schools. (2021) *W3Schools* [Online]. Available from: https://www.w3schools.com/xml/xml_tree.asp [Accessed 5 May 2021].