LONDON METROPOLITAN UNIVERSITY

islington college

(इस्लिङ्टन कलेज)

Module Code & Module Title

CS5004NI Emerging Programming Platforms and Technologies

Assessment Weightage &

Type 30% Individual

Coursework

Year and Semester

2019-20 Autumn / 2020-21 Spring

Student Name:

London Met ID:

College ID:

Assignment Due Date:

Assignment Submission Date:

Title (Where Required):

Word Count (Where Required):

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

2. XML Document.. 1

2.1 List of Data 2

2.1.1 List of Elements.. 2

2.1.2 List of Attribute 3

2.2 Tree Diagram.. 4

2.3 XML Code.. 6

3. Schema Document.. 13

4. Difference between Schema and DTD.. 16

5. Testing 17

5.1 Test 1:.. 17

5.2 Test 2.. 19

5.3 Test 3.. 22

5.4 Test 4.. 24

5.5 Test 5.. 25

6. Coursework Development.. 27

7. Critical Analysis 31

8.Conclusion 34

References.. 34

Appendix 35

CSS.. 35

Table of table

Table 1 list of element 2

Table 2 list of attribute 3

Table 3 Difference between Schema and DTD 16

Table 4 test 1 17

Table 5 test 2 19

Table 6 test 3 22

Table 7 test 4 24

Table 8 test 5 25

Figure of figure

Figure 1 Test1: Inserting XML File 17

Figure 2 Test1: Inserting XSD File 18

Figure 3 Test1: Output 18

Figure 4 before removing offer element in xml 19

Figure 5 : Before Removing offer Element in Website.. 20

Figure 6 : After Removing offer Element in XML File 20

Figure 7 After Removing offer Element in Website.. 21

Figure 8 xml file without css 22

Figure 9 XML file displayed in the browser without CSS 23

Figure 10 xml with css.. 23

Figure 11 xml and css are linked... 24

Figure 12 Hover Effect of Card Panel.. 25

Figure 13 css of background image of cards... 26

Figure 14 css working in the while running the code.. 26

1.Introduction

A model system for a music store named “Gift Card Store” is needed to be developed as an XML developer. After an interview with the store manager, the software requirements are noted for the model system. The details of the store including the Store Name, Address, Telephone Number, Website name and so on. The store has several gift cards of different types like digital and physical cards. With the above-mentioned scenario, an XML document is to be prepared with the additional information. At least 15 data, 5 attributes, and 5 optional elements are to be included in the document. A tree diagram is made to show the structure of the XML document.

An XML document catalog_20048793.xml, a schema file name catalog_20048793.xsd and a CSS file name catalog_20048793.css is to be created. The .XML files contain the content of the document, the .XSD is for validation and the .CSS file is for styling the document. The XML file is to be validated against the Schema files using an online validator https://www.xmlvalidation.com/ and the CSS files require specifications such as a list, at least one border, at least two different font sizes, and font families, two different colors, logo image, and one floating box.

2. XML Document

The Extensible Markup Language (XML) is a simple text-based format for representing structured information: documents, data, configuration, books, transactions, invoices, and much more. It was derived from an older standard format called SGML (ISO 8879), in order to be more suitable for Web use. XML is one of the most widely-used formats for sharing structured information today: between programs, between people, between computers and people, both locally and across networks. If you are already familiar with HTML, you can see that XML is very similar. However, the syntax rules of XML are strict: XML tools will not process files that contain errors, but instead will give you error messages so that you fix them. This means that almost all XML documents can be processed reliably by computer software (w3c, 2022).

2.1 List of Data

2.1.1 List of Elements

List of Elements	Element Type	Element Data	Occurrence
Store_Name	Simple type	Child element	Only once
cost	Simple type	String	One or more
validity	Simple type	String	One or more
offer	Simple type	String	Zero or more
discount	Simple type	String	One or more
user	Simple type	String	One or more
instock	Simple type	String	Zero or one
note	Simple type	String	Zero or one
Refund	Simple type	String	Zero or more
Delivery	Simple type	String	Zero or more
Header	Complex type	String	Only once
Cards	Complex type	Child element	Only once
card	Complex type	Child element	One or more
Footer	Complex type	Child element	Only once
Website	Simple type	String	Only once
Address	Simple type	String	Only once
Telephone	Simple type	String	Only once
Copyright	Simple type	String	Only once

Table 1 list of element

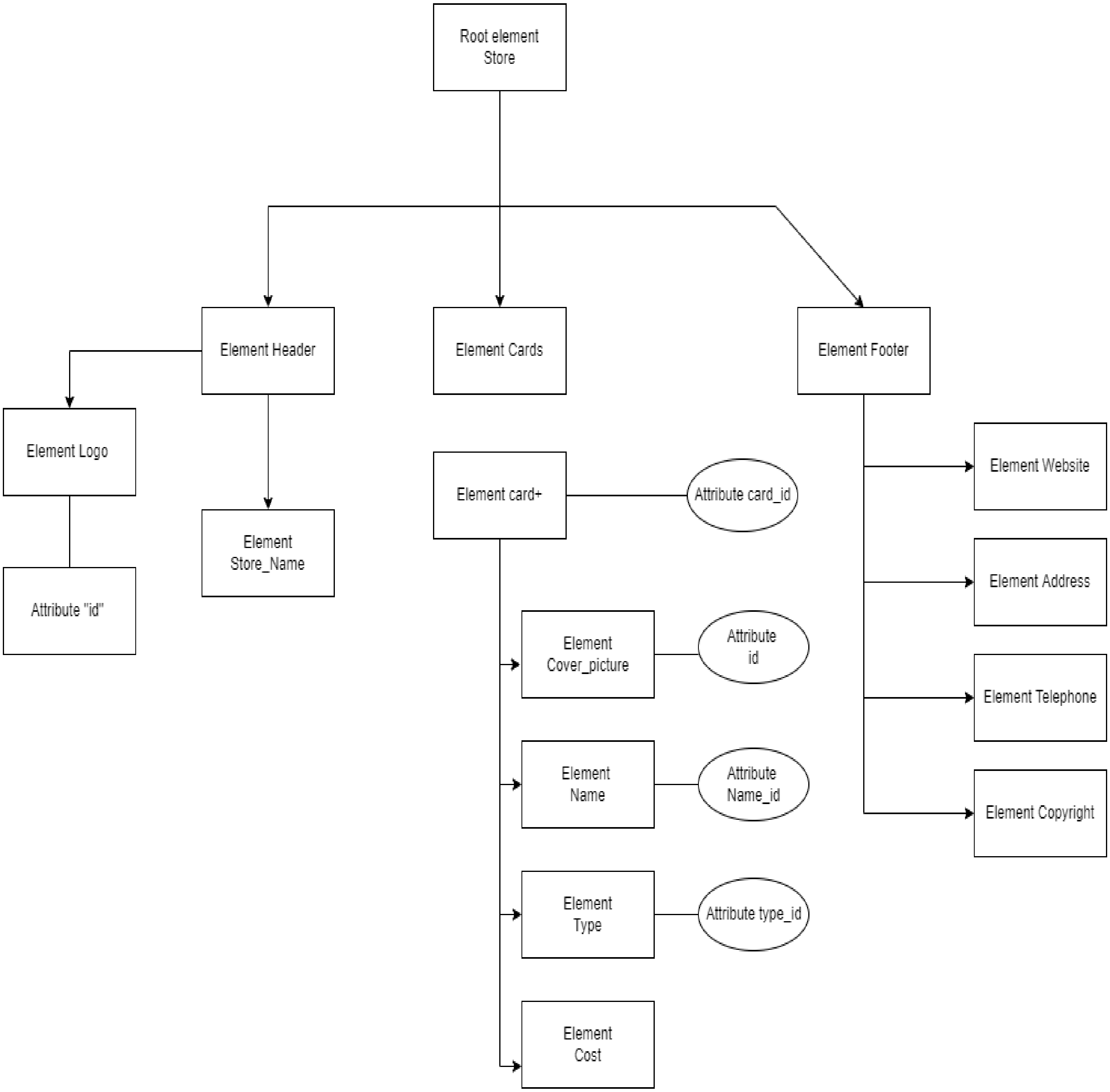
2.1.2 List of Attribute

List of Attributes	Data type	Default	Fixed	Use
id	string			required
Card_id	string			required
Type_id	string			required
Name_id	string			required
Cover_id	string			required

Table 2 list of attribute

2.2 Tree Diagram

The tree structure contains root elements (Store), child elements, and so on. The tree diagram was made with all succeeding branch elements and sub-branch elements starting from the root. The attributes of the branch elements were displayed in the tree diagram. For the tree diagram draw.io website was used to make the tree diagram. The Root element was displayed with its branch elements such as Header, Cards, and Footer. All the Elements of our XML file are displayed in the tree diagram along with their corresponding attributes. The Repeating and Optional element was also displayed by symbols.


Root element
Store
Element Header
Element Cards
Element Footer
Element Logo
Attribute "id"
Element
Store_Name
Element card+
Attribute card_id
Element
Cover_picture
Attribute
id
Element
Name
Attribute
Name_id
Element
Type
Attribute type_id
Element
Cost
Element Website
Element Address
Element Telephone
Element Copyright

2.3 XML Code

```
<?xml version="1.0" standalone="no"?>
<!--

 Coursework 2
    XML
    Author:  Aadesh Gyawali
    Date: 24th April 2021
    Filename: catalog_20048793.xml

-->

<?xml-stylesheet href="catalog_20048793.css" type="text/css"?>

<Store  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="./catalog_20048793.xsd">

    <Header>
        <Logo id="Logo"/>
        <Store_Name>GIFT CARD STORE</Store_Name>
    </Header>

    <Cards>

        <card Card_Id="card_1">
            <Cover_Picture id="cover_1"/>
            <Name Name_Id="name_1">Apple Gift Card</Name>
            <Type type_id="type">Type: physical card</Type>
            <cost>Cost: $50-$400</cost>
            <validity>Validaity: no expiration date</validity>
            <user>User: 5 people</user>
            <offer>Offer: unlimited 3% Daily Cash back on everything you buy at Apple </offer>
            <discount>Discount: up to 10%</discount>
        </card>

        <card Card_Id="card_2">
            <Cover_Picture id="cover_2"/>
            <Name Name_Id="name_1">DSW Gift Card</Name>
            <Type type_id="type">Type: physical card</Type>
            <cost>Cost: $25-$1400</cost>
            <validity>Validaity: no expiration date</validity>
            <instock>Instock: available 13</instock>
            <discount>Discount: up to 20%</discount>
            <note>Note: limited offer for 13 gift cards only</note>
        </card>

        <card Card_Id="card_3">
            <Cover_Picture id="cover_3"/>
            <Name Name_Id="name_1">Starbucks Gift Card</Name>
            <Type type_id="type">Type: physical card</Type>
            <cost>Cost: $400-$500</cost>
            <validity>Validaity: no expiration date</validity>
            <Refund>Refund: No returns or refunds</Refund>
            <offer>Offer: Buy a $20+ Starbucks eGift, Get a $5 Starbucks eGift Card</offer>
            <user>User: more than 1 people</user>
            <discount>Discount: up to 10%</discount>
        </card>

        <card Card_Id="card_4">
            <Cover_Picture id="cover_4"/>
            <Name Name_Id="name_1">Valorant Gift Card</Name>
            <Type type_id="type">Type: digital card</Type>
            <cost>Cost: $25-$100</cost>
            <validity>Validaity: no time limit</validity>
            <user>User: 1 people</user>
            <discount>Discount: up to 10%</discount>
            <note>Note: get exclusive weapons in limited offer </note>
        </card>

        <card Card_Id="card_5">
            <Cover_Picture id="cover_5"/>
            <Name Name_Id="name_1">DoorDash Gift Card</Name>
            <Type type_id="type">Type: physical card</Type>
            <cost>Cost: $25-$500</cost>
            <validity>Validaity: no expiration date</validity>
            <Refund>Refund: No returns or refunds</Refund>
            <user>User: 1 people</user>
            <discount>Discount: up to 20%</discount>
        </card>

        <card Card_Id="card_6">
            <Cover_Picture id="cover_6"/>
            <Name Name_Id="name_1">Visa Gift Card</Name>
            <Type type_id="type">Type: digital card</Type>
            <cost>Cost: $10-$250</cost>
            <validity>Validaity: no expiration date</validity>
            <user>User: single user</user>
            <discount>Discount: up to 10%</discount>
        </card>

        <card Card_Id="card_7">
            <Cover_Picture id="cover_7"/>
            <Name Name_Id="name_1">Google Play Gift Card</Name>
            <Type type_id="type">Type: digital card</Type>
            <cost>Cost: $25-$500</cost>
            <validity>Validaity: no expiration date</validity>
            <user>User: more than 2 people</user>
            <instock>Instock: available 100</instock>
            <discount>Discount: up to 10%</discount>
        </card>

              <card Card_Id="card_8">
                  <Cover_Picture id="cover_8"/>
                  <Name Name_Id="name_1">Steam Gift Card</Name>
                  <Type type_id="type">Type: physical card</Type>
                  <cost>Cost: $5-$100</cost>
                  <validity>Validaity: no expiration date</validity>
                  <instock>Instock: available 43</instock>
                  <user>User:more than 2 people</user>

              </card>

              <card Card_Id="card_9">
                  <Cover_Picture id="cover_9"/>
                  <Name Name_Id="name_1">Spotify Gift Card</Name>
                  <Type type_id="type">Type: digital card</Type>
                  <cost>Cost: $30-$60</cost>
                  <validity>Validaity: up to 12 months</validity>
                  <Delivery>Delivery: by Email</Delivery>
                  <user>User: more than 1 people</user>
                  <discount>Discount: up to 20%</discount>
              </card>

              <card Card_Id="card_10">
                  <Cover_Picture id="cover_10"/>
                  <Name Name_Id="name_1">Whole Foods Market Gift Card</Name>
                  <Type type_id="type">Type: physical card</Type>
                  <cost>Cost: $25-$200</cost>
                  <validity>Validaity: never expire</validity>
                  <Refund>Refund: No returns or refunds</Refund>
                  <user>User: more than 1 people</user>
                  <discount>Discount: up to 20%</discount>
              </card>

              <card Card_Id="card_11">
                  <Cover_Picture id="cover_11"/>
                  <Name Name_Id="name_1">Netflix Gift Card</Name>
                  <Type type_id="type">Type: digital card</Type>
                  <cost>Cost: $25-$200</cost>
                  <validity>Validaity: no expiration date</validity>
                  <user>User: 10 people</user>
                  <Delivery>Delivery: by Email</Delivery>
                  <discount>Discount: up to 20%</discount>
              </card>

              <card Card_Id="card_12">
                  <Cover_Picture id="cover_12"/>
                  <Name Name_Id="name_1">Target Gift Card</Name>
                  <Type type_id="type">Type: physical card</Type>
                  <cost>Cost: $10-$200</cost>
                  <validity>Validaity: no expiration date</validity>
                  <user>User: 1 people</user>
                  <instock>Instock: not available </instock>
                  <discount>Discount: up to 10%</discount>
              </card>

              <card Card_Id="card_13">
                  <Cover_Picture id="cover_13"/>
                  <Name Name_Id="name_1">Bookshop Gift Card</Name>
                  <Type type_id="type">Type: digital card</Type>
                  <cost>Cost: $10-$1000</cost>
                  <validity>Validaity: never expire</validity>
                  <Delivery>Delivery: by Email</Delivery>
                  <instock>Instock: available 63</instock>
                  <discount>Discount: no discount</discount>
              </card>

              <card Card_Id="card_14">
                  <Cover_Picture id="cover_14"/>
                  <Name Name_Id="name_1">Uber e-Gift Card</Name>
```

```
                <Type type_id="type">Type: digital card</Type>
                <cost>Cost: $25-$200</cost>
                <validity>Validaity: no expiration date</validity>
                <user>User: more than people</user>
                <Delivery>Delivery: by Email</Delivery>
                <discount>Discount: up to 4%</discount>
            </card>

            <card Card_Id="card_15">
                <Cover_Picture id="cover_15"/>
                <Name Name_Id="name_1">Chipotle Gift Card </Name>
                <Type type_id="type">Type: physical card</Type>
                <cost>Cost: $25-$75</cost>
                <validity>Validaity: no expiration date</validity>
                <Refund>Refund: No returns or refunds</Refund>
                <discount>Discount: up to 10%</discount>
            </card>

        </Cards>

        <Footer>
            <Website>www.GiftCardStore.com</Website>
            <Address>Address: Greenwich,London</Address>
            <Telephone>Contact: 01-6545654 & 9861062001</Telephone>
            <Copyright>Copyright © 2022 Gift Card Store. All rights reserverd.</Copyright>
        </Footer>

```

```
</Store>
```

3. Schema Document

XML Schema is commonly known as XML Schema Definition (XSD). It is used to describe and validate the structure and the content of XML data. XML schema defines the elements, attributes and data types. Schema element supports Namespaces. It is similar to a database schema that describes the data in a database (tutorialpoint, 2022).

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Store">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Logo">
                <xs:complexType>
                  <xs:attribute name="id" type="xs:string" use="required" />
                </xs:complexType>
              </xs:element>
              <xs:element name="Store_Name" type="xs:string" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="Cards">
          <xs:complexType>
            <xs:sequence>
        <xs:element maxOccurs="unbounded" name="card">
          <xs:complexType>
            <xs:sequence>
              <xs:choice maxOccurs="unbounded">
                <xs:element name="Cover_Picture">
                  <xs:complexType>
                    <xs:attribute name="id" type="xs:string" use="required" />
                  </xs:complexType>
                </xs:element>
                <xs:element name="Name">
                  <xs:complexType>
                    <xs:simpleContent>
                      <xs:extension base="xs:string">
                        <xs:attribute name="Name_Id" type="xs:string" use="required" />
                      </xs:extension>
                    </xs:simpleContent>
                  </xs:complexType>
                </xs:element>
                <xs:element name="Type">
                  <xs:complexType>
                    <xs:simpleContent>
                      <xs:extension base="xs:string">
                        <xs:attribute name="type_id" type="xs:string" use="required" />
                      </xs:extension>
                    </xs:simpleContent>
                  </xs:complexType>
                </xs:element>
                <xs:element name="cost" type="xs:string" />
                <xs:element name="validity" type="xs:string" />
```

```
                      <xs:element name="Refund" type="xs:string" />
                      <xs:element name="instock" type="xs:string" />
                      <xs:element name="user" type="xs:string" />
                      <xs:element name="offer" type="xs:string" />
                      <xs:element name="discount" type="xs:string" />
                      <xs:element name="note" type="xs:string" />
                      <xs:element name="Delivery" type="xs:string" />
                    </xs:choice>
                  </xs:sequence>
                  <xs:attribute name="Card_Id" type="xs:string" use="required" />
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="Footer">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Website" type="xs:string" />
              <xs:element name="Address" type="xs:string" />
              <xs:element name="Telephone" type="xs:string" />
              <xs:element name="Copyright" type="xs:string" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
```

```
  </xs:element>
</xs:schema>
```

4. Difference between Schema and DTD

	DTD	XSD
.1.	DTD are the declarations that define a document type for SGML.	XSD describes the elements in a XML document.
2.	DTD (Document Type Definition) are the declaration that define a document type for SGML.	XSD (XML Schema Definition) describes the elements in an XML document
3.	DTD doesn’t define the order of the child elements	XSD defines an order for the child Element
4.	Validation of Content to Data Types cannot be done in DTD	Validation of Content to Data Types is Possible in XSD
5.	It doesn’t support namespace.	It supports namespace.
6.	It is comparatively harder than XSD.	It is relatively more simpler than DTD.
7.	It doesn’t support datatypes.	It supports datatypes.
8.	SGML syntax is used for DTD.	XML is used for writing XSD.
9.	It is not extensible in nature.	It is extensible in nature.
10.	It doesn’t give us much control on structure of XML document.	It gives us more control on structure of XML document.

Table 3 Difference between Schema and DTD

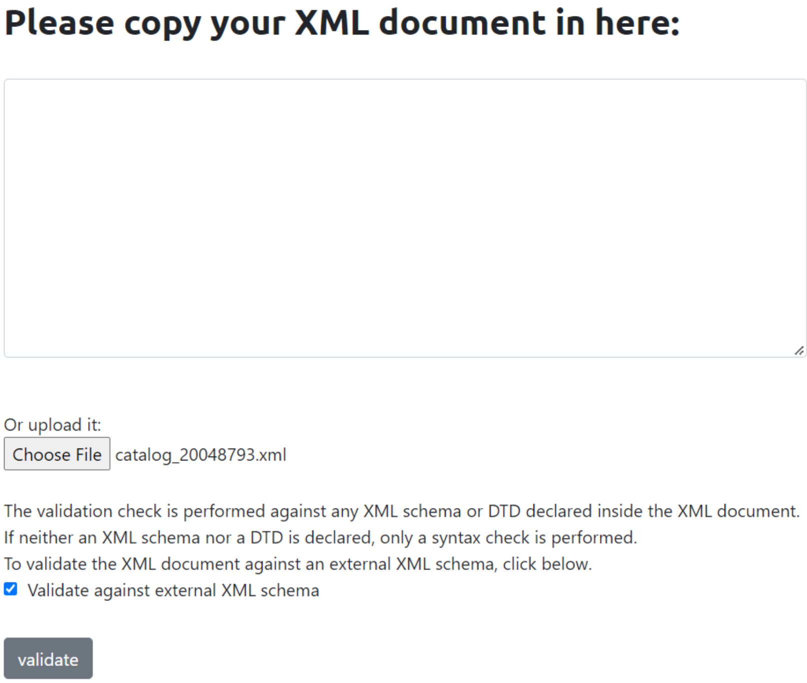
5. Testing

5.1 Test 1:

To Check if XML, XSD is properly formatted and validated or not.

Test no	1
Action	Uploading XML file in the website for validation of its format. Uploading XSD file in the website
Expected Result	No errors may be seen in our documents.
Actual Result	No errors are seen in our documents.
Conclusion	The test was Successful.

Table 4 test 1



Please copy your XML document in here:

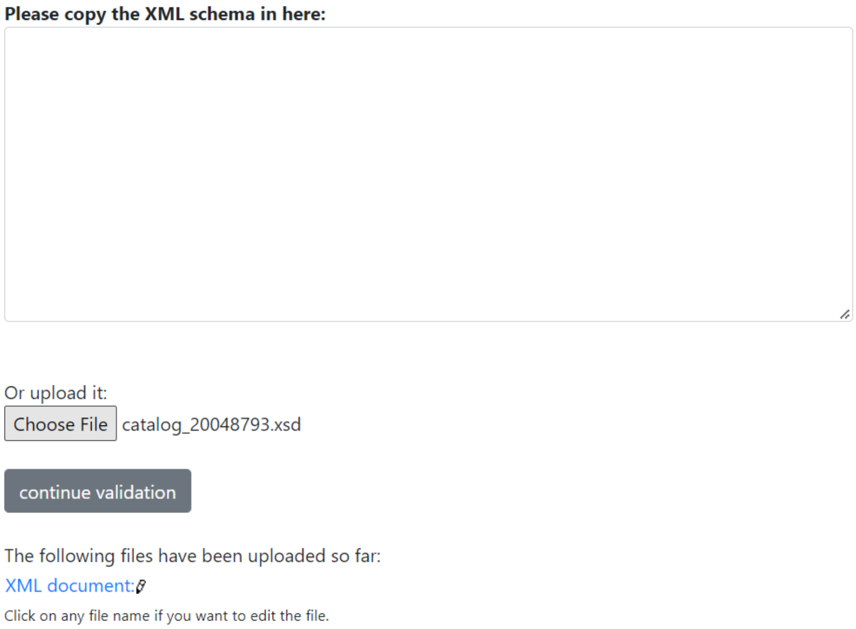
Or upload it:

catalog_20048793.xml

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

☑ Validate against external XML schema

Figure 1 Test1: Inserting XML File



Please copy the XML schema in here:

Or upload it:

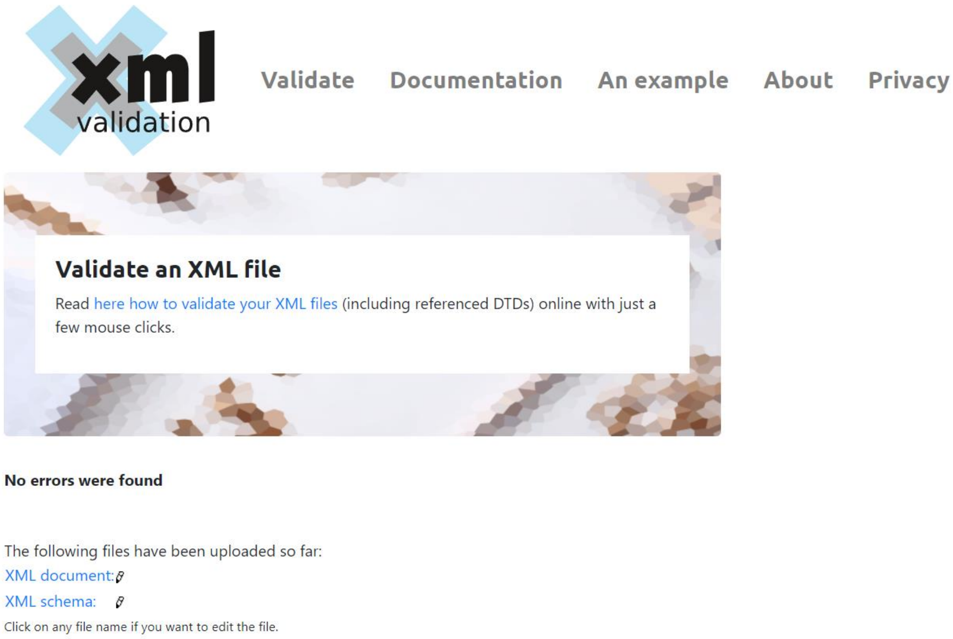
catalog_20048793.xsd

The following files have been uploaded so far:

XML document:

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

Figure 2 Test1: Inserting XSD File



Validate an XML file

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

No errors were found

The following files have been uploaded so far:

XML document:

XML schema:

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

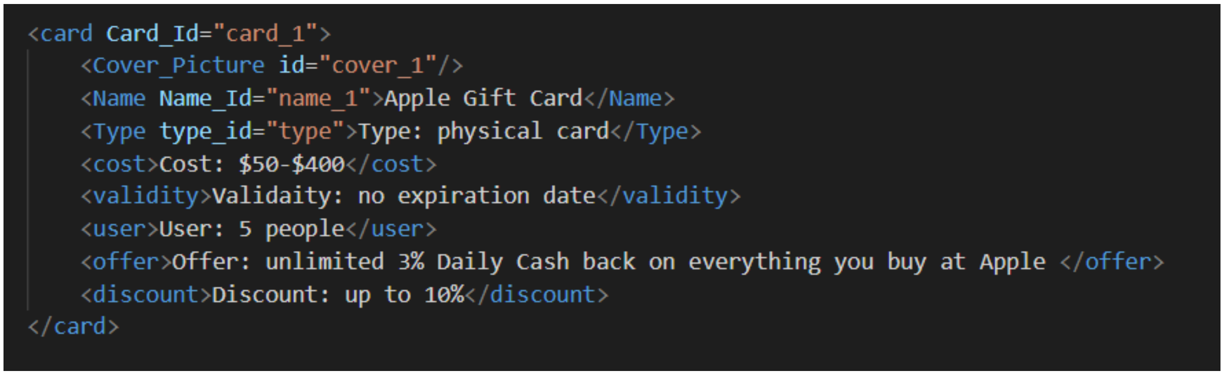
Figure 3 Test1: Output

5.2 Test 2

To check whether the optional element in the XML file would work or not.

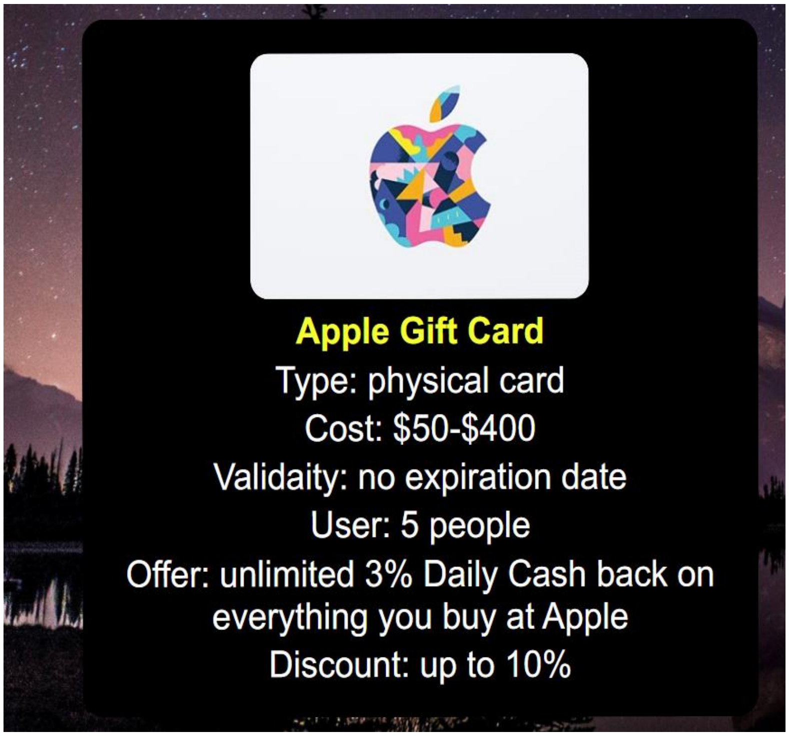
Test no	2
Action	Remove optional element Album from the XML file. Save and run the file.
Expected Result	No errors may be seen in our text editor and browser
Actual Result	No errors may be seen in our text editor and browser
Conclusion	The test was Successful.

Table 5 test 2



```
<card Card_Id="card_1">
    <Cover_Picture id="cover_1"/>
    <Name Name_Id="name_1">Apple Gift Card</Name>
    <Type type_id="type">Type: physical card</Type>
    <cost>Cost: $50-$400</cost>
    <validity>Validaity: no expiration date</validity>
    <user>User: 5 people</user>
    <offer>Offer: unlimited 3% Daily Cash back on everything you buy at Apple </offer>
    <discount>Discount: up to 10%</discount>
</card>
```

Figure 4 before removing offer element in xml



Apple Gift Card

Type: physical card

Cost: $50-$400

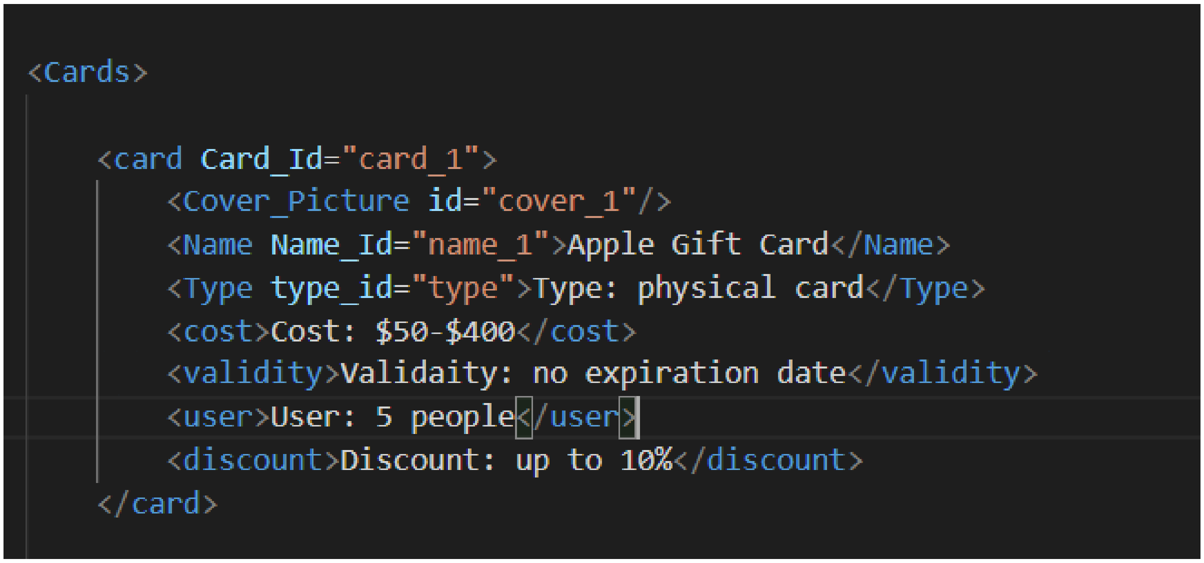
Validaity: no expiration date

User: 5 people

Offer: unlimited 3% Daily Cash back on everything you buy at Apple

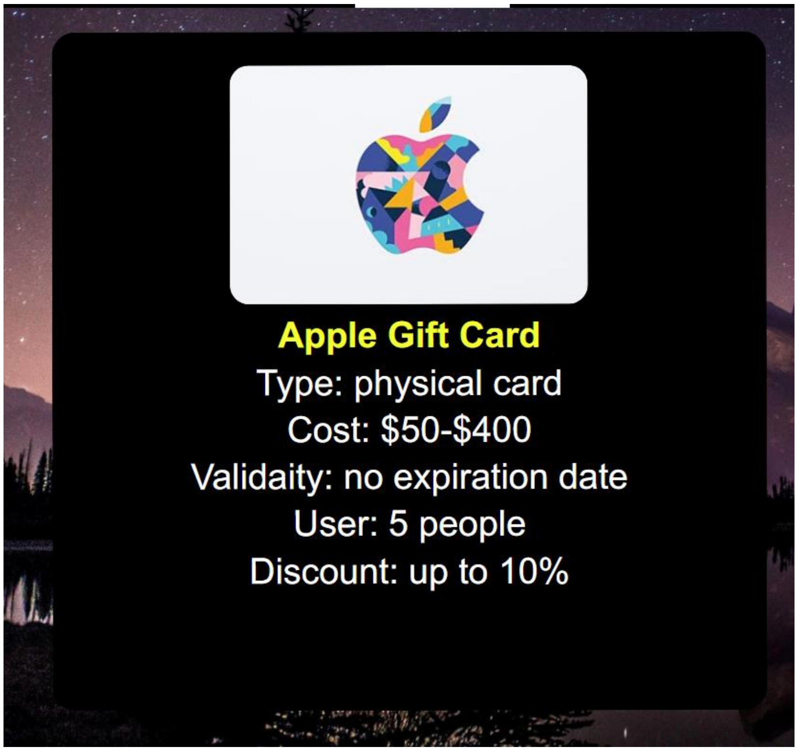
Discount: up to 10%

Figure 5 : Before Removing offer Element in Website



```
<Cards>

    <card Card_Id="card_1">
        <Cover_Picture id="cover_1"/>
        <Name Name_Id="name_1">Apple Gift Card</Name>
        <Type type_id="type">Type: physical card</Type>
        <cost>Cost: $50-$400</cost>
        <validity>Validaity: no expiration date</validity>
        <user>User: 5 people</user>
        <discount>Discount: up to 10%</discount>
    </card>
```

Figure 6 : After Removing offer Element in XML File



Apple Gift Card

Type: physical card

Cost: $50-$400

Validaity: no expiration date

User: 5 people

Discount: up to 10%

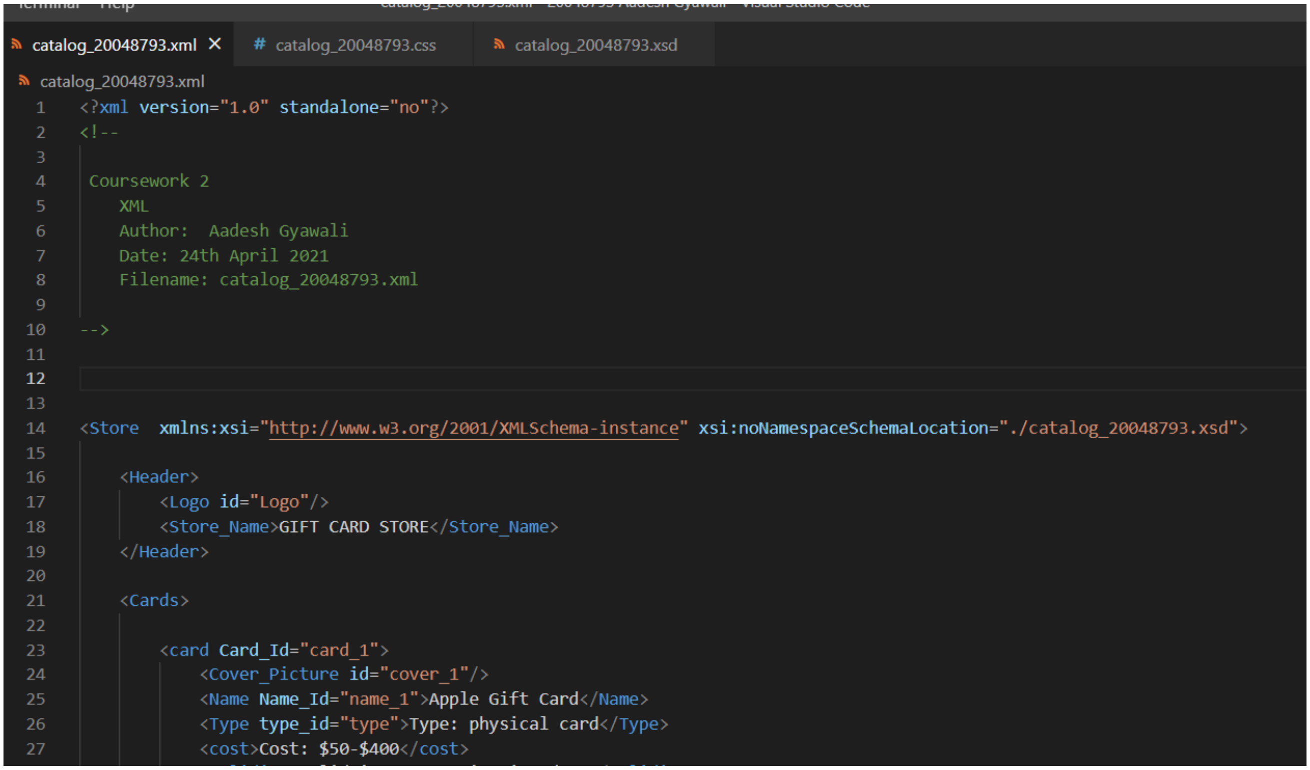
Figure 7 After Removing offer Element in Website

5.3 Test 3

To check whether the CSS file is working or not.

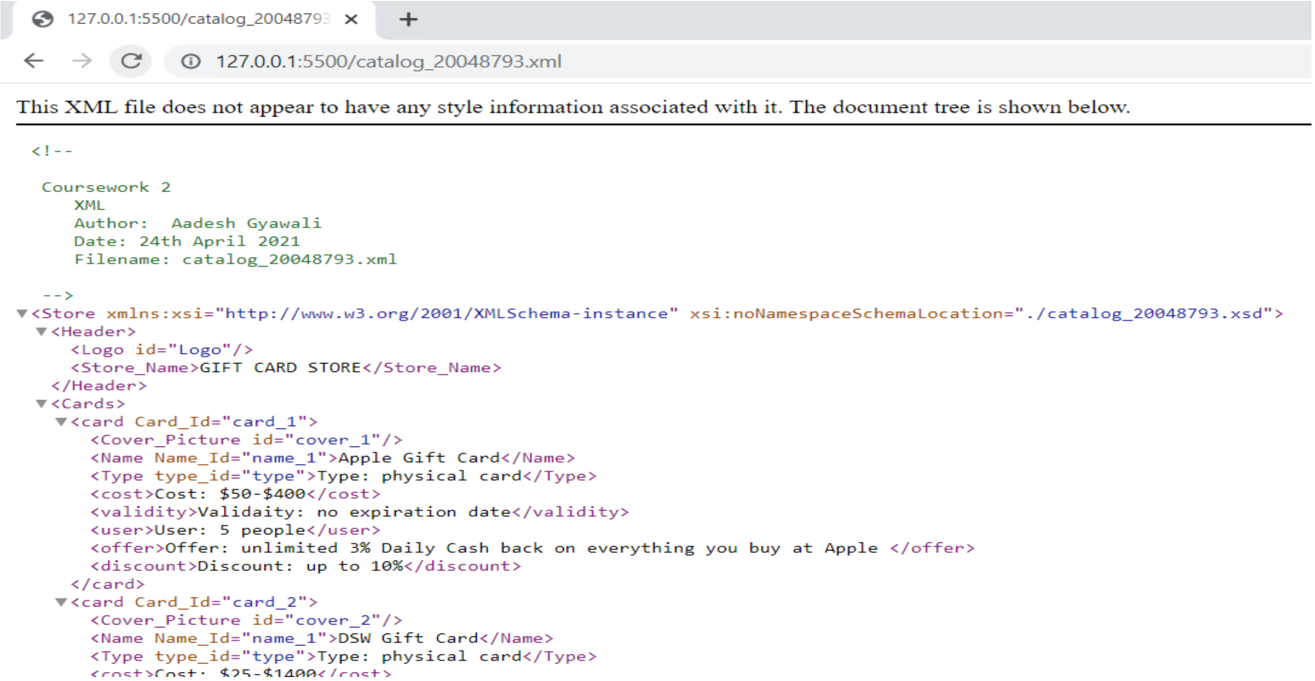
Test no	3
Action	XML file is opened in the browser without linking CSS file. XML file is opened in the browser with linking CSS file
Expected Result	The CSS design was visible in the browser after the linking CSS.
Actual Result	The CSS design was visible in the browser after the linking CSS.
Conclusion	The test was Successful.

Table 6 test 3



```
<?xml version="1.0" standalone="no"?>
<!--

 Coursework 2
    XML
    Author:  Aadesh Gyawali
    Date: 24th April 2021
    Filename: catalog_20048793.xml

-->



<Store  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="./catalog_20048793.xsd">

    <Header>
        <Logo id="Logo"/>
        <Store_Name>GIFT CARD STORE</Store_Name>
    </Header>

    <Cards>

        <card Card_Id="card_1">
            <Cover_Picture id="cover_1"/>
            <Name Name_Id="name_1">Apple Gift Card</Name>
            <Type type_id="type">Type: physical card</Type>
            <cost>Cost: $50-$400</cost>
```

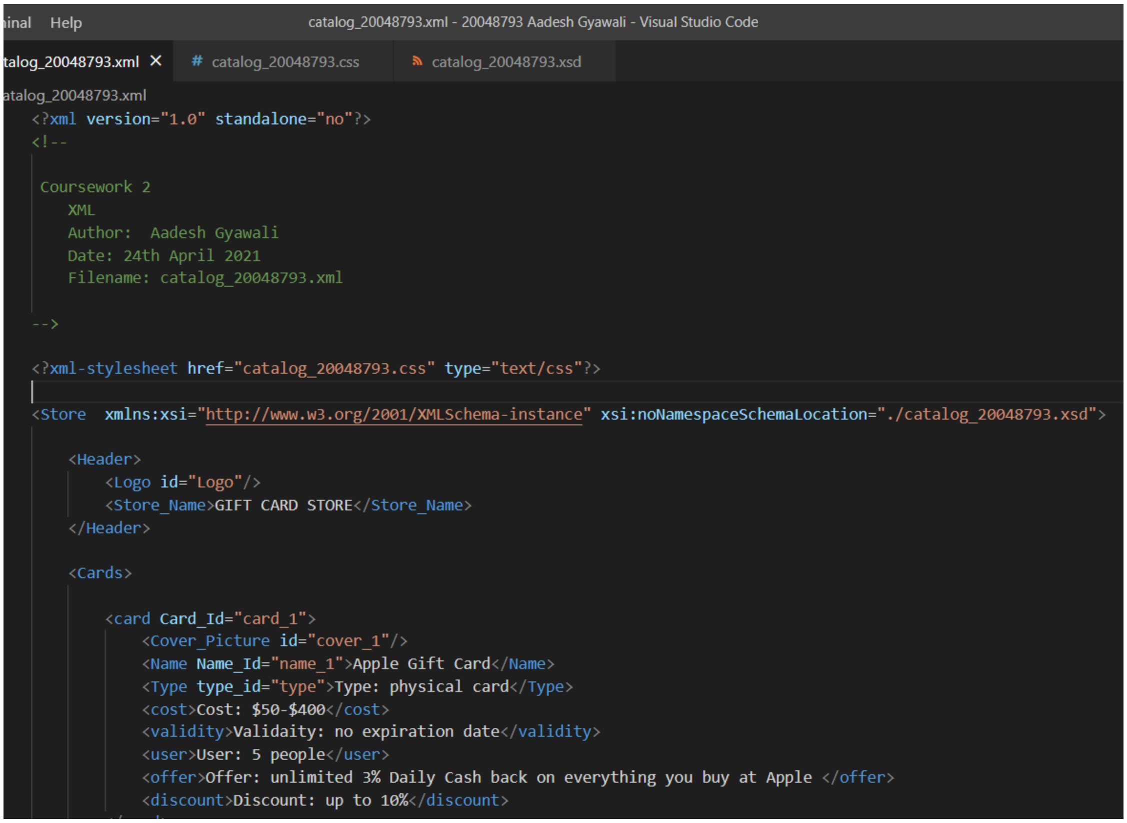
Figure 8 xml file without css


127.0.0.1:5500/catalog_2004879
127.0.0.1:5500/catalog_20048793.xml

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

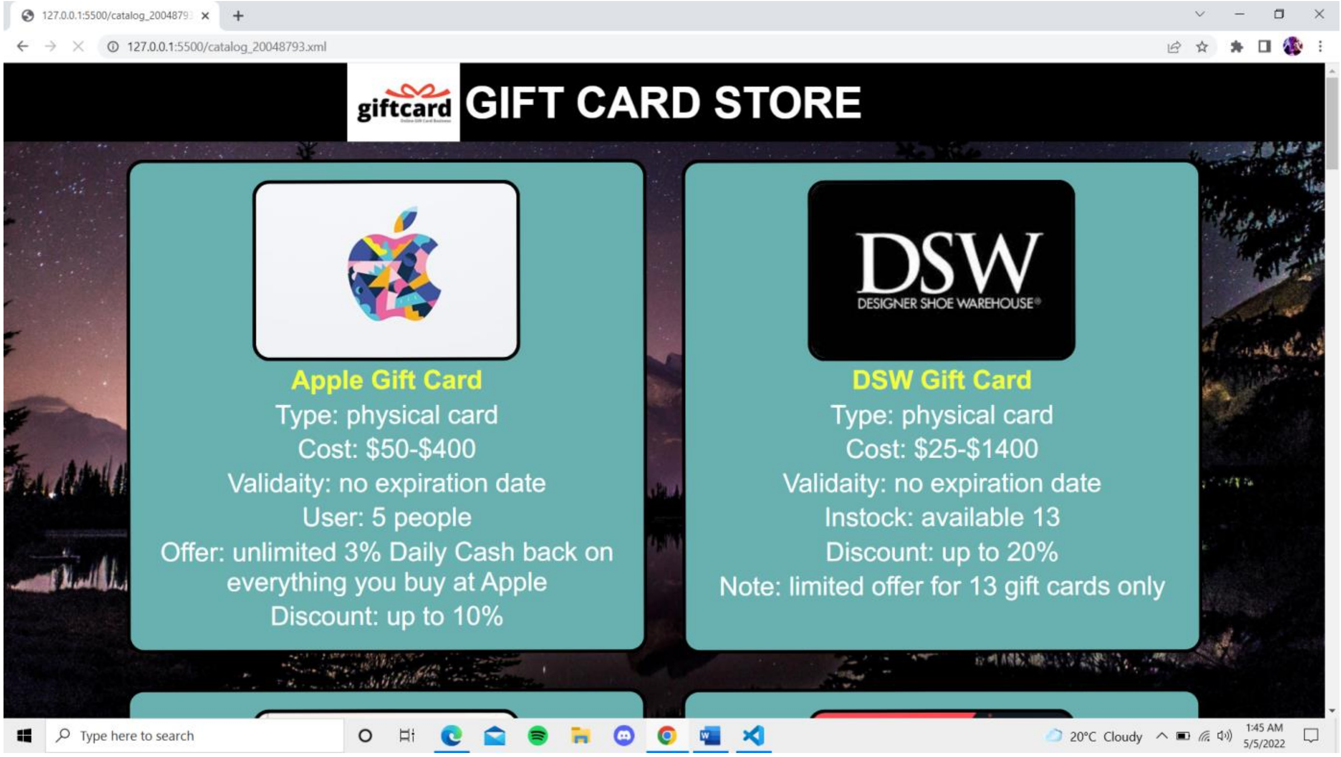
```
<!--

 Coursework 2
    XML
    Author:  Aadesh Gyawali
    Date: 24th April 2021
    Filename: catalog_20048793.xml

 -->
▼<Store xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="./catalog_20048793.xsd">
  ▼<Header>
     <Logo id="Logo"/>
     <Store_Name>GIFT CARD STORE</Store_Name>
   </Header>
  ▼<Cards>
    ▼<card Card_Id="card_1">
       <Cover_Picture id="cover_1"/>
       <Name Name_Id="name_1">Apple Gift Card</Name>
       <Type type_id="type">Type: physical card</Type>
       <cost>Cost: $50-$400</cost>
       <validity>Validaity: no expiration date</validity>
       <user>User: 5 people</user>
       <offer>Offer: unlimited 3% Daily Cash back on everything you buy at Apple </offer>
       <discount>Discount: up to 10%</discount>
     </card>
    ▼<card Card_Id="card_2">
       <Cover_Picture id="cover_2"/>
       <Name Name_Id="name_1">DSW Gift Card</Name>
       <Type type_id="type">Type: physical card</Type>
       <cost>Cost: $25-$1400</cost>
```

Figure 9 XML file displayed in the browser without CSS



```
<?xml version="1.0" standalone="no"?>
<!--

 Coursework 2
    XML
    Author:  Aadesh Gyawali
    Date: 24th April 2021
    Filename: catalog_20048793.xml

-->

<?xml-stylesheet href="catalog_20048793.css" type="text/css"?>

<Store  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="./catalog_20048793.xsd">

    <Header>
        <Logo id="Logo"/>
        <Store_Name>GIFT CARD STORE</Store_Name>
    </Header>

    <Cards>

        <card Card_Id="card_1">
            <Cover_Picture id="cover_1"/>
            <Name Name_Id="name_1">Apple Gift Card</Name>
            <Type type_id="type">Type: physical card</Type>
            <cost>Cost: $50-$400</cost>
            <validity>Validaity: no expiration date</validity>
            <user>User: 5 people</user>
            <offer>Offer: unlimited 3% Daily Cash back on everything you buy at Apple </offer>
            <discount>Discount: up to 10%</discount>
```

Figure 10 xml with css



GIFT CARD STORE

Apple Gift Card
Type: physical card
Cost: $50-$400
Validaity: no expiration date
User: 5 people
Offer: unlimited 3% Daily Cash back on everything you buy at Apple
Discount: up to 10%

DSW Gift Card
Type: physical card
Cost: $25-$1400
Validaity: no expiration date
Instock: available 13
Discount: up to 20%
Note: limited offer for 13 gift cards only

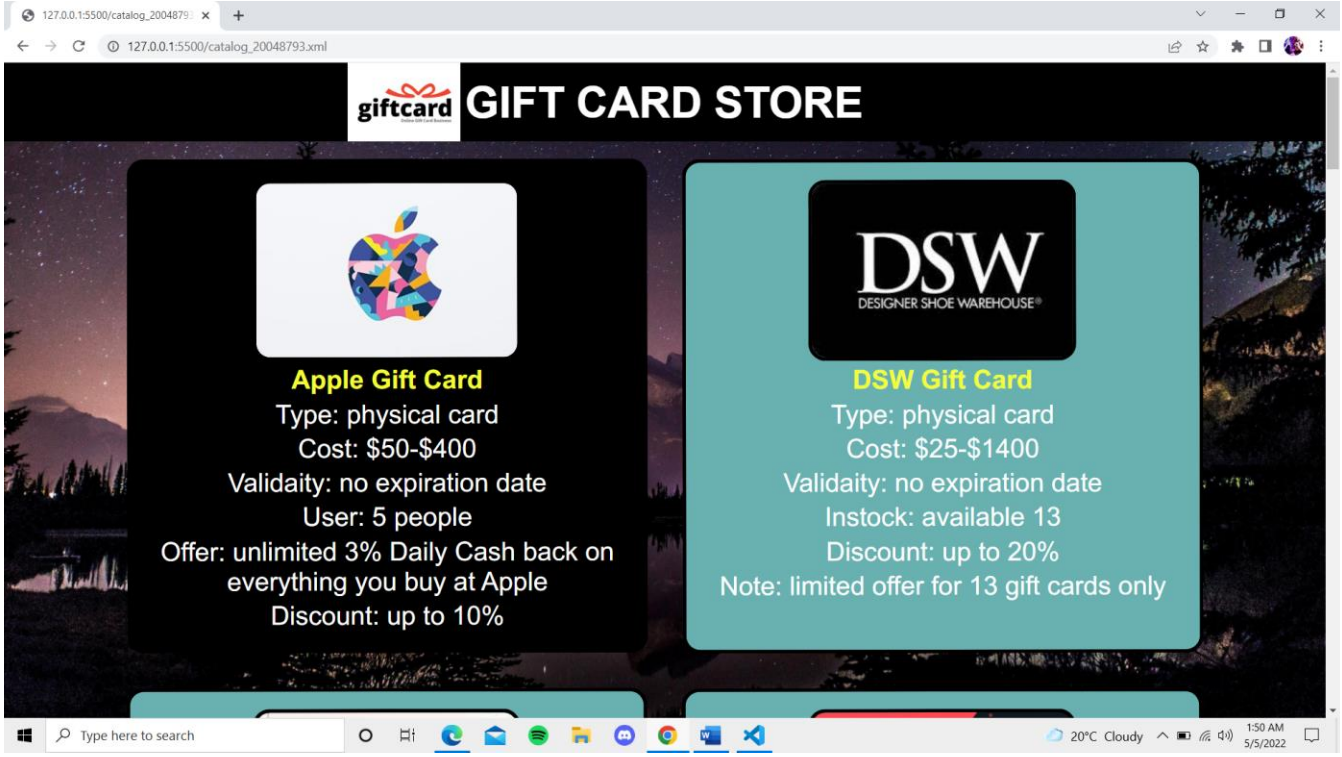
Figure 11 xml and css are linked

5.4 Test 4

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

Test no	4
Action	Catalog_20048793 was opened in the browse. The cursor hovered around the card Panel of Apple Gift Card
Expected Result	The hover effected can be seen by the change of color on card panel.
Actual Result	The hover effected can be seen by the change of color on card panel.
Conclusion	The test was Successful.

Table 7 test 4



GIFT CARD STORE

Apple Gift Card
Type: physical card
Cost: $50-$400
Validaity: no expiration date
User: 5 people
Offer: unlimited 3% Daily Cash back on everything you buy at Apple
Discount: up to 10%

DSW Gift Card
Type: physical card
Cost: $25-$1400
Validaity: no expiration date
Instock: available 13
Discount: up to 20%
Note: limited offer for 13 gift cards only

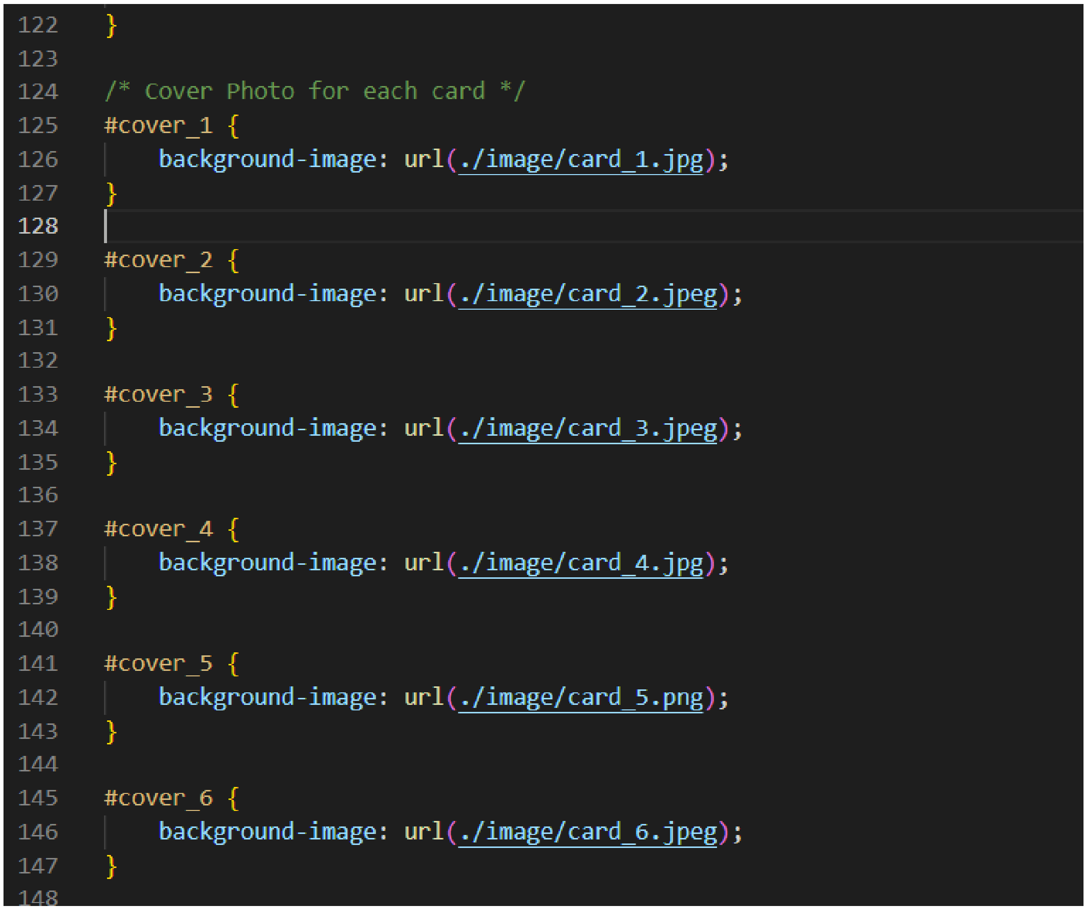
Figure 12 Hover Effect of Card Panel

5.5 Test 5

To check the back ground image of card is working

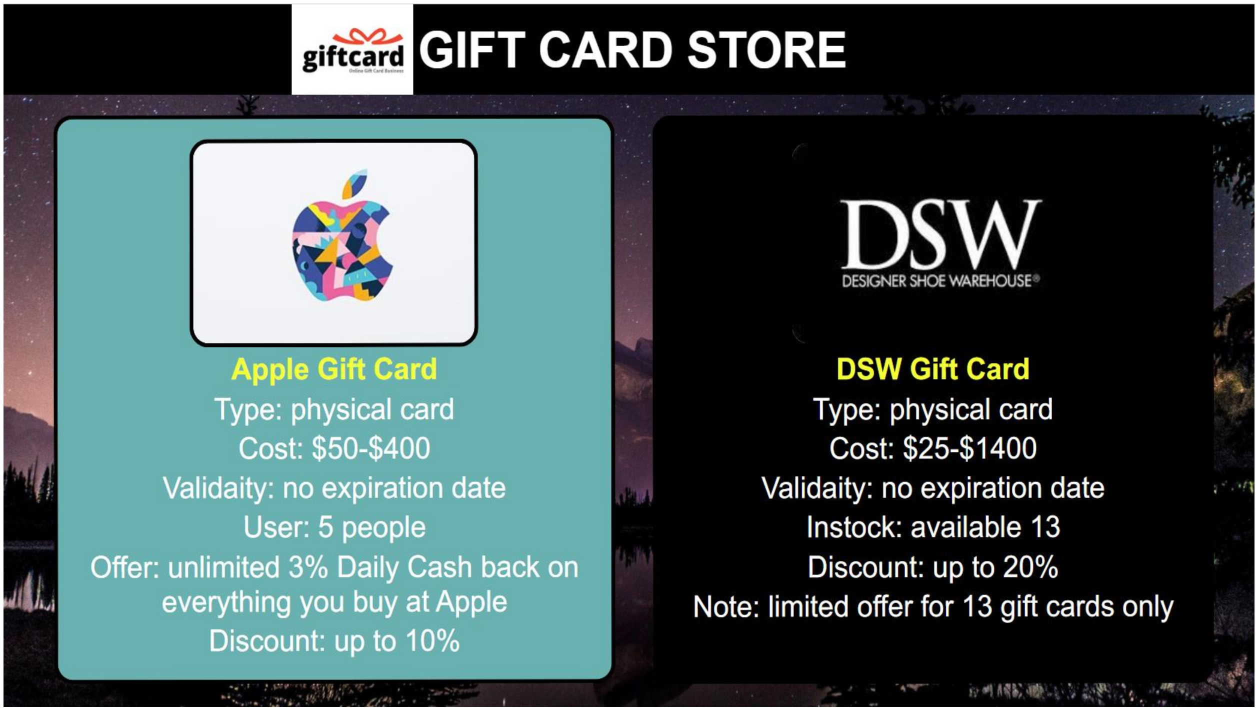
Test no	5
Action	opening the catalog_20048793.css and checking image is working
Expected Result	No errors may be seen in our text editor and browser
Actual Result	No errors may be seen in our text editor and browser
Conclusion	The test was Successful.

Table 8 test 5



```css
}

/* Cover Photo for each card */
#cover_1 {
    background-image: url(./image/card_1.jpg);
}

#cover_2 {
    background-image: url(./image/card_2.jpeg);
}

#cover_3 {
    background-image: url(./image/card_3.jpeg);
}

#cover_4 {
    background-image: url(./image/card_4.jpg);
}

#cover_5 {
    background-image: url(./image/card_5.png);
}

#cover_6 {
    background-image: url(./image/card_6.jpeg);
}
```

Figure 13 css of background image of cards



GIFT CARD STORE

Apple Gift Card
Type: physical card
Cost: $50-$400
Validaity: no expiration date
User: 5 people
Offer: unlimited 3% Daily Cash back on everything you buy at Apple
Discount: up to 10%

DSW Gift Card
Type: physical card
Cost: $25-$1400
Validaity: no expiration date
Instock: available 13
Discount: up to 20%
Note: limited offer for 13 gift cards only

Figure 14 css working in the while running the code

6. Coursework Development

An xml file catalog_20048793.xml, a schema file catalog_20048793.xsd and css file name catalog_20048793.css was developed. The .xml file contain the document, the .xsd contain the schema which is used for validation and css file for styling the documentation.

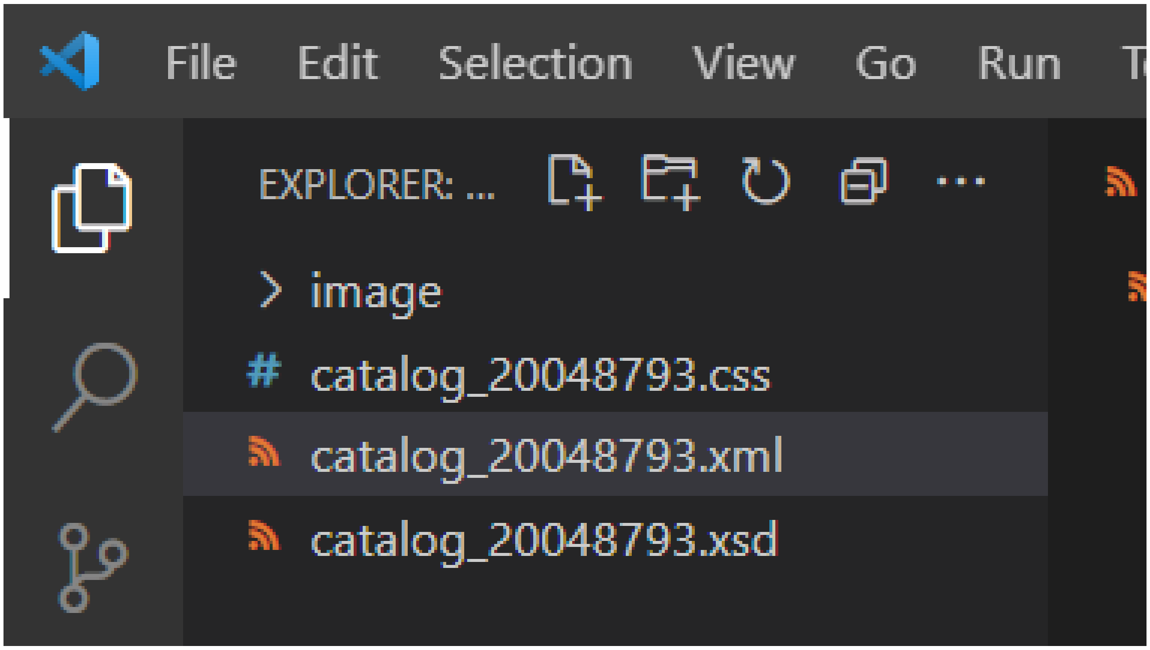
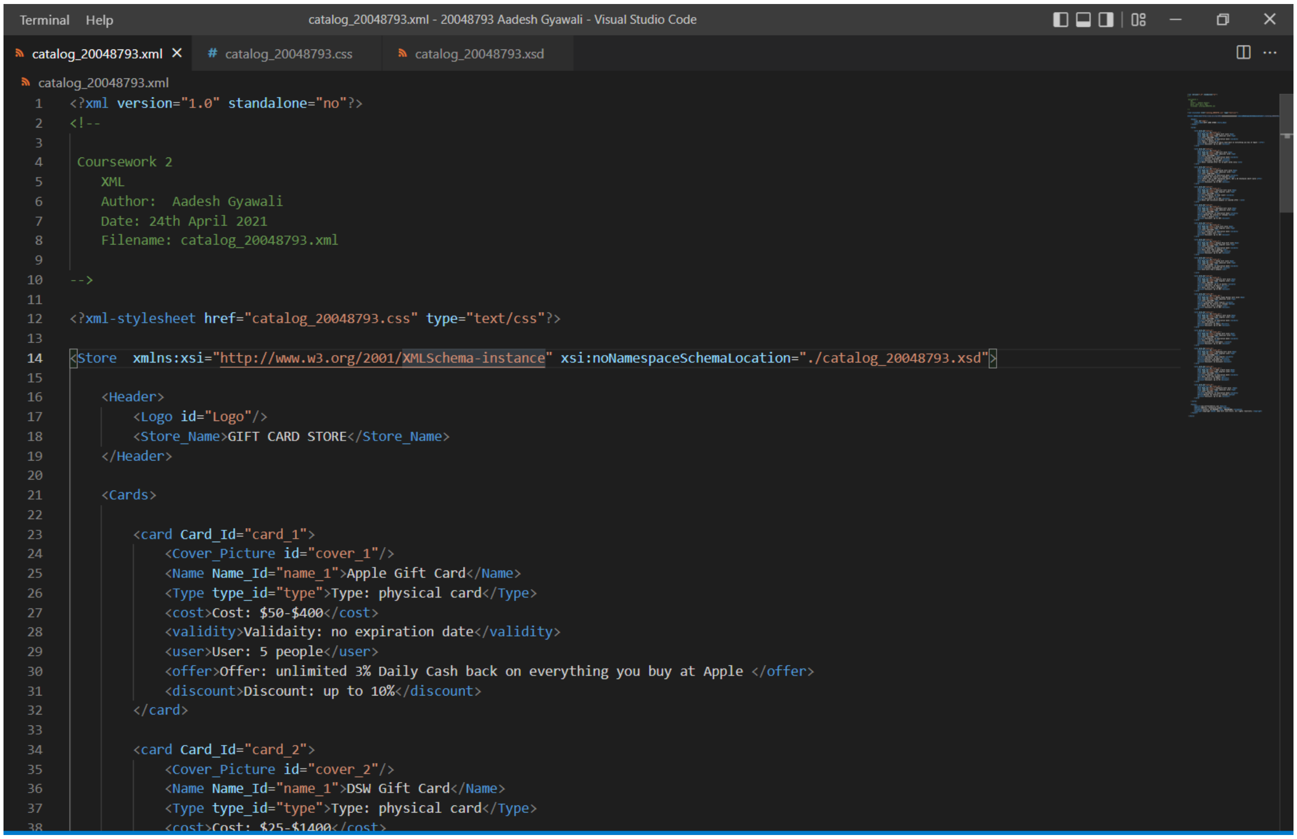


image
catalog_20048793.css
catalog_20048793.xml
catalog_20048793.xsd

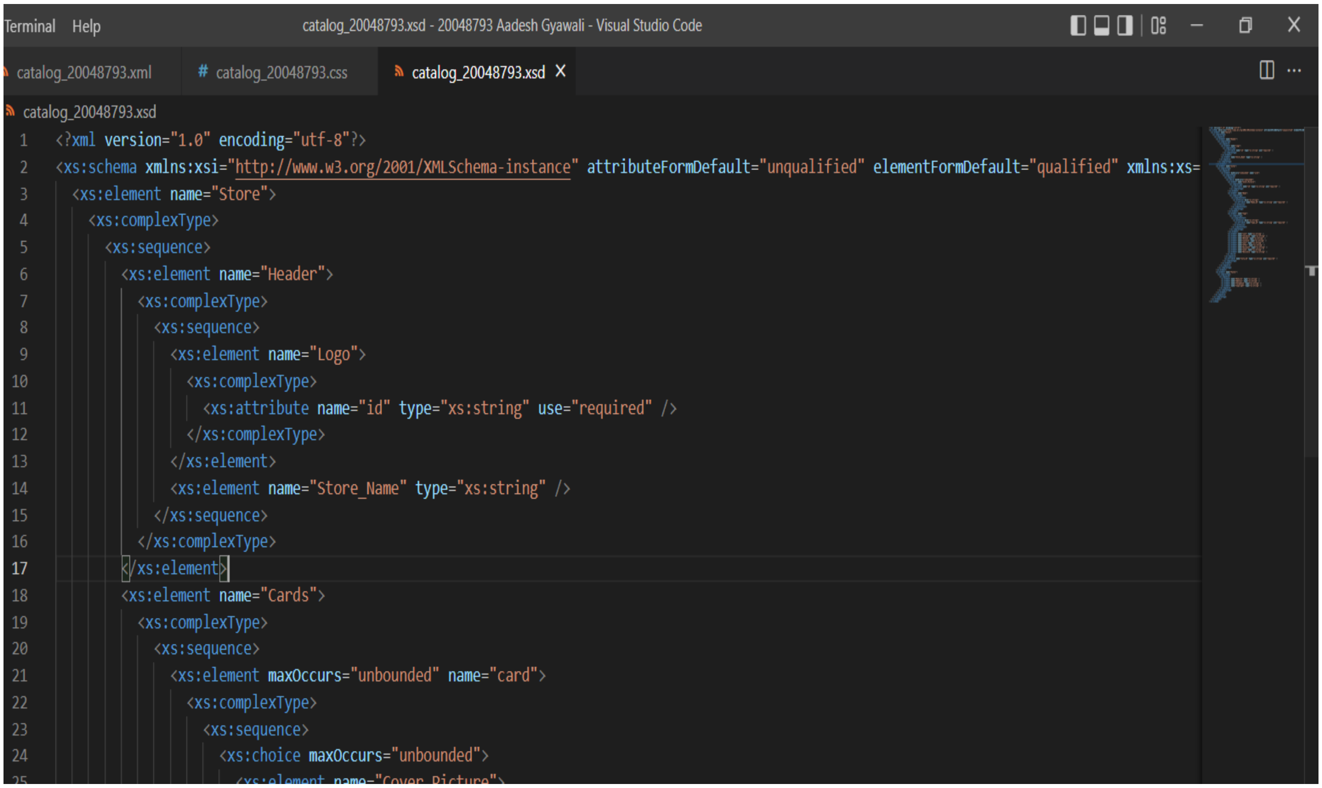
The course work was developed in the code editor VS code What is VS Code used for? Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as Visual Studio IDE (Visual Studio Code, 2022).

In the xml file a single root element is required to enclose the other tags, in the document. The root contain the subtags header, cards, footer. The header contain the logo and name of the gift card store. The cards contains the 15 different cards details. And the footer contains the website, address, telephone, etc.



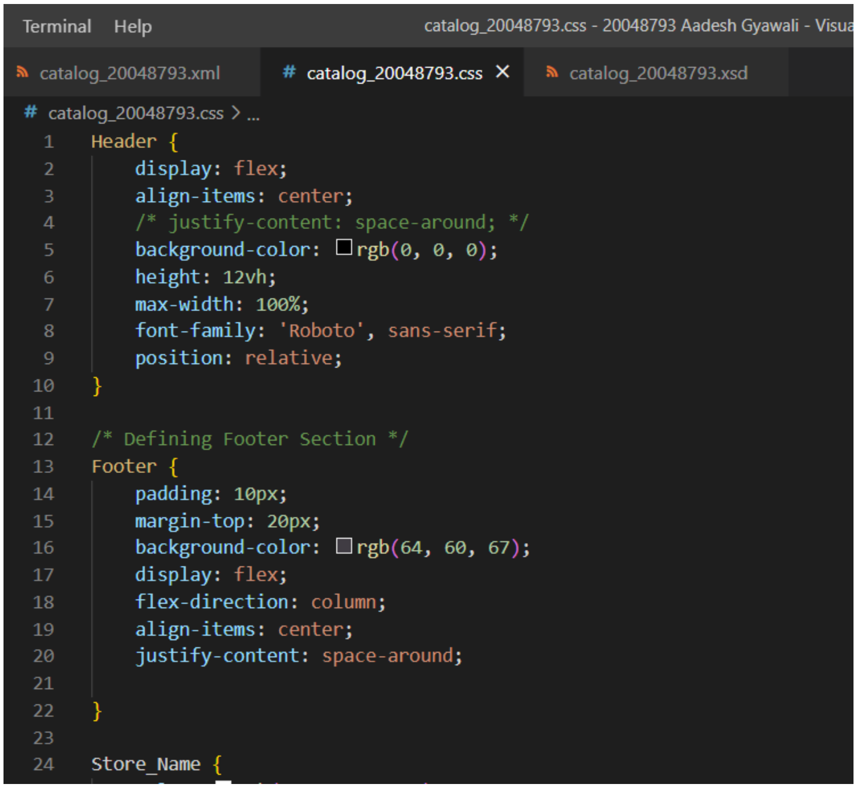
```xml
<?xml version="1.0" standalone="no"?>
<!--

 Coursework 2
    XML
    Author:  Aadesh Gyawali
    Date: 24th April 2021
    Filename: catalog_20048793.xml

-->

<?xml-stylesheet href="catalog_20048793.css" type="text/css"?>

<Store  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="./catalog_20048793.xsd">

    <Header>
        <Logo id="Logo"/>
        <Store_Name>GIFT CARD STORE</Store_Name>
    </Header>

    <Cards>

        <card Card_Id="card_1">
            <Cover_Picture id="cover_1"/>
            <Name Name_Id="name_1">Apple Gift Card</Name>
            <Type type_id="type">Type: physical card</Type>
            <cost>Cost: $50-$400</cost>
            <validity>Validaity: no expiration date</validity>
            <user>User: 5 people</user>
            <offer>Offer: unlimited 3% Daily Cash back on everything you buy at Apple </offer>
            <discount>Discount: up to 10%</discount>
        </card>

        <card Card_Id="card_2">
            <Cover_Picture id="cover_2"/>
            <Name Name_Id="name_1">DSW Gift Card</Name>
            <Type type_id="type">Type: physical card</Type>
            <cost>Cost: $25-$1400</cost>
```

in the Schema file, the Schema Elements and Attributes were defined. The default and fixed values for Simple Elements, Repeating Elements and Optional Elements were Declared. The default and fixed values for Attributes were declared. The store was the root element for the XSD file. we are using namespace http://www.w3.org/2001/XMLSchema as a namespace target namespace xs is created.



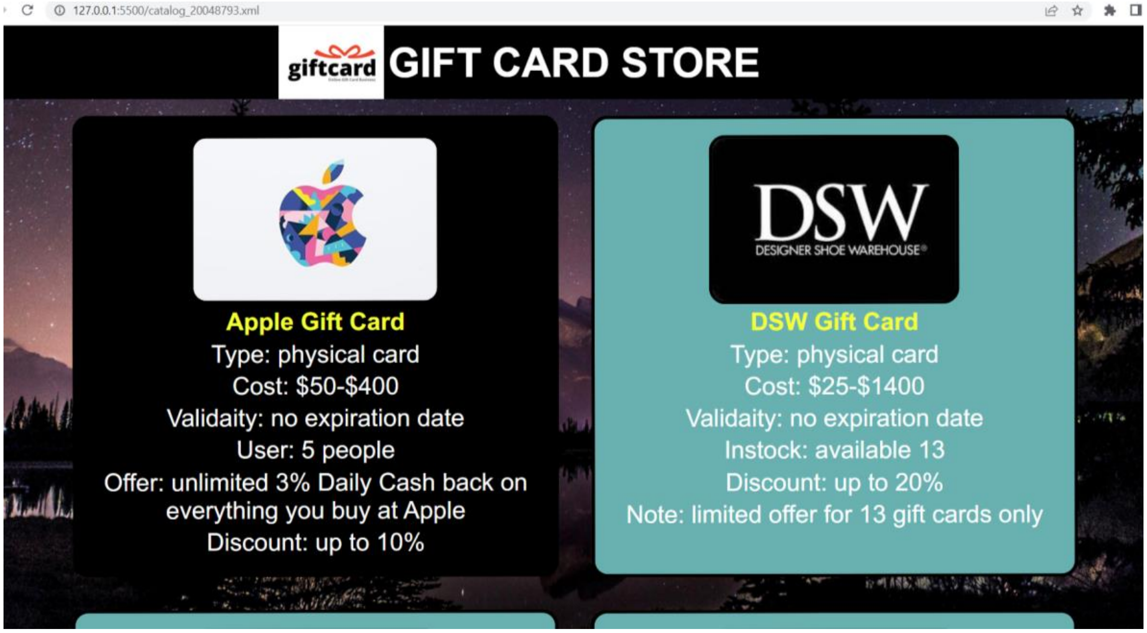
```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs=
  <xs:element name="Store">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Logo">
                <xs:complexType>
                  <xs:attribute name="id" type="xs:string" use="required" />
                </xs:complexType>
              </xs:element>
              <xs:element name="Store_Name" type="xs:string" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="Cards">
          <xs:complexType>
            <xs:sequence>
              <xs:element maxOccurs="unbounded" name="card">
                <xs:complexType>
                  <xs:sequence>
                    <xs:choice maxOccurs="unbounded">
                      <xs:element name="Cover_Picture">
```

Css also included while developing the project. The CSS file contains a list of the genre, border of the song panel, different font sizes, and two different font family Poppins and Roboto. The CSS file contains a list of the genre, border of the song panel, different font sizes, and two different font family Poppins and Roboto.


Terminal
Help
catalog_20048793.css - 20048793 Aadesh Gyawali - Visua
catalog_20048793.xml
catalog_20048793.css
catalog_20048793.xsd
catalog_20048793.css > ...

```css
Header {
    display: flex;
    align-items: center;
    /* justify-content: space-around; */
    background-color: rgb(0, 0, 0);
    height: 12vh;
    max-width: 100%;
    font-family: 'Roboto', sans-serif;
    position: relative;
}

/* Defining Footer Section */
Footer {
    padding: 10px;
    margin-top: 20px;
    background-color: rgb(64, 60, 67);
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: space-around;

}

Store_Name {
```

After combining all the element of .xml ,,xsd and css we get the final product.



GIFT CARD STORE

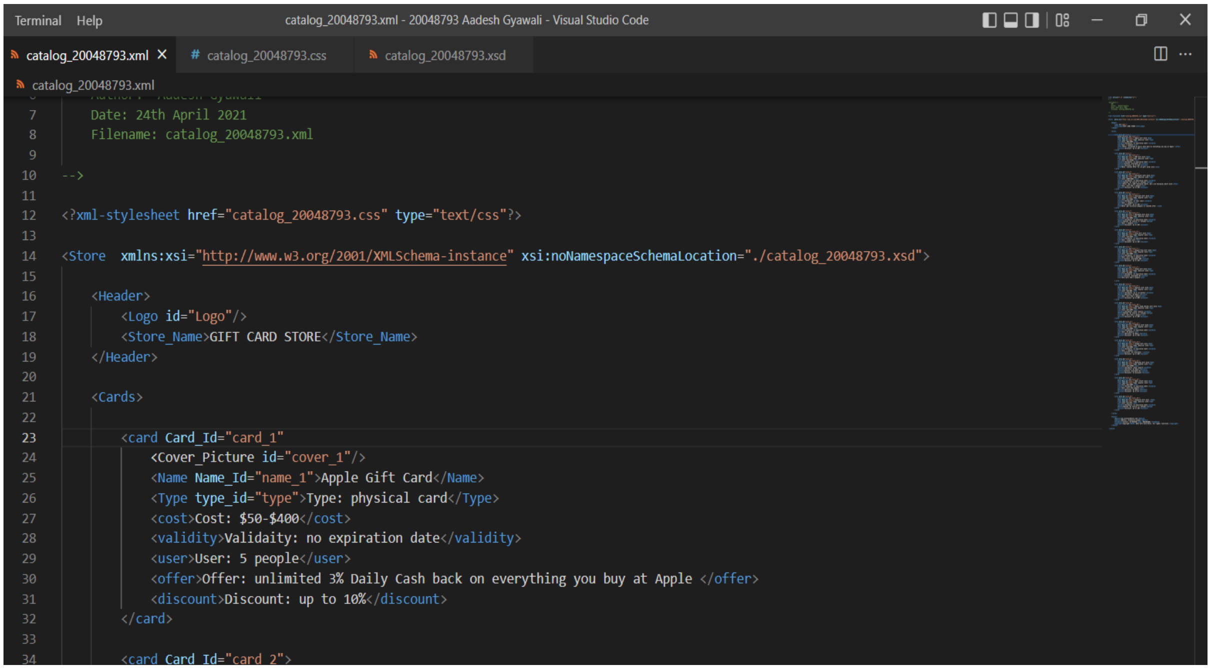
Apple Gift Card
Type: physical card
Cost: $50-$400
Validaity: no expiration date
User: 5 people
Offer: unlimited 3% Daily Cash back on everything you buy at Apple
Discount: up to 10%

DSW Gift Card
Type: physical card
Cost: $25-$1400
Validaity: no expiration date
Instock: available 13
Discount: up to 20%
Note: limited offer for 13 gift cards only

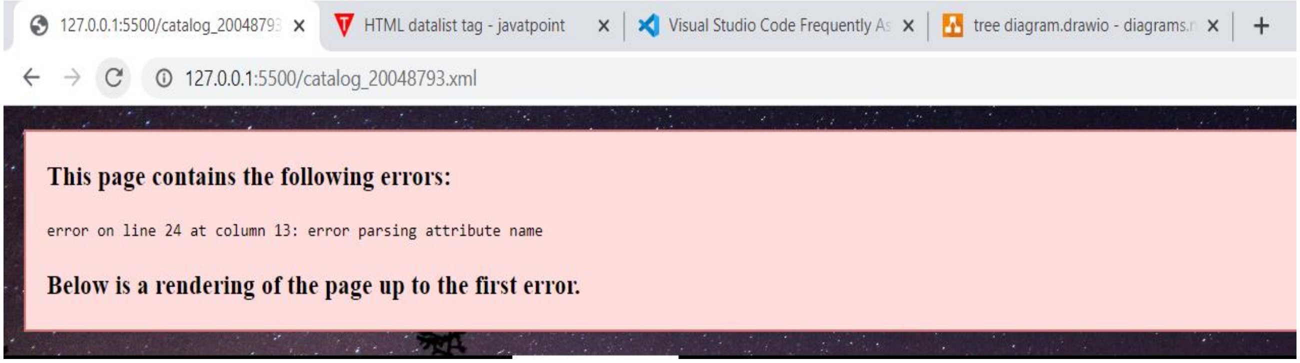
7. Critical Analysis

The coursework requires us to develop an XML document that includes details about a store. Due to the Coursework, I have gained skills and knowledge in the new field of XML and Schema. Now I can make a website that stores data in an XML file, which validation is verified by Schema. Many errors were encountered during the completion of the coursework, the errors were dealt with with proper research and guidance from our module teachers.

Some of the errors that were encountered are:



```
7       Date: 24th April 2021
8       Filename: catalog_20048793.xml
9
10  -->
11
12  <?xml-stylesheet href="catalog_20048793.css" type="text/css"?>
13
14  <Store  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="./catalog_20048793.xsd">
15
16      <Header>
17          <Logo id="Logo"/>
18          <Store_Name>GIFT CARD STORE</Store_Name>
19      </Header>
20
21      <Cards>
22
23          <card Card_Id="card_1"
24              <Cover_Picture id="cover_1"/>
25              <Name Name_Id="name_1">Apple Gift Card</Name>
26              <Type type_id="type">Type: physical card</Type>
27              <cost>Cost: $50-$400</cost>
28              <validity>Validaity: no expiration date</validity>
29              <user>User: 5 people</user>
30              <offer>Offer: unlimited 3% Daily Cash back on everything you buy at Apple </offer>
31              <discount>Discount: up to 10%</discount>
32          </card>
33
34          <card Card_Id="card_2">
```

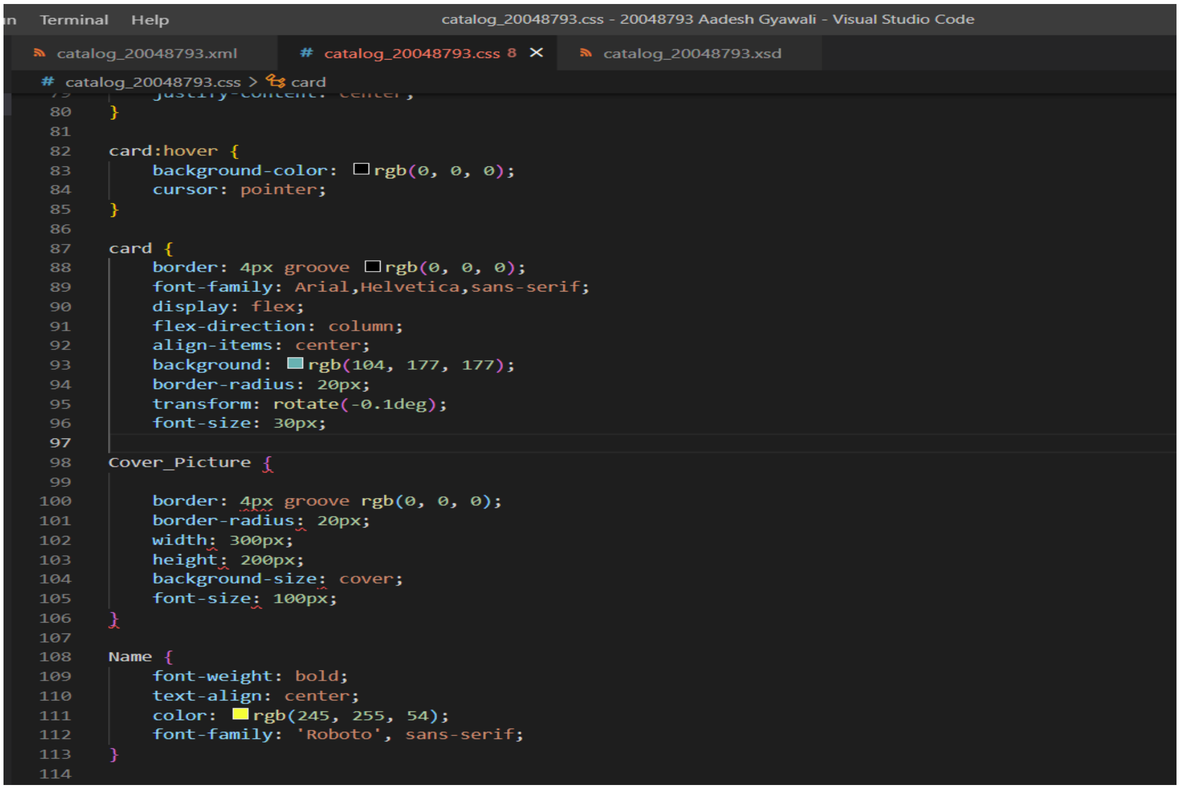

127.0.0.1:5500/catalog_2004879
HTML datalist tag - javatpoint
Visual Studio Code Frequently As
tree diagram.drawio - diagrams.n
127.0.0.1:5500/catalog_20048793.xml

This page contains the following errors:

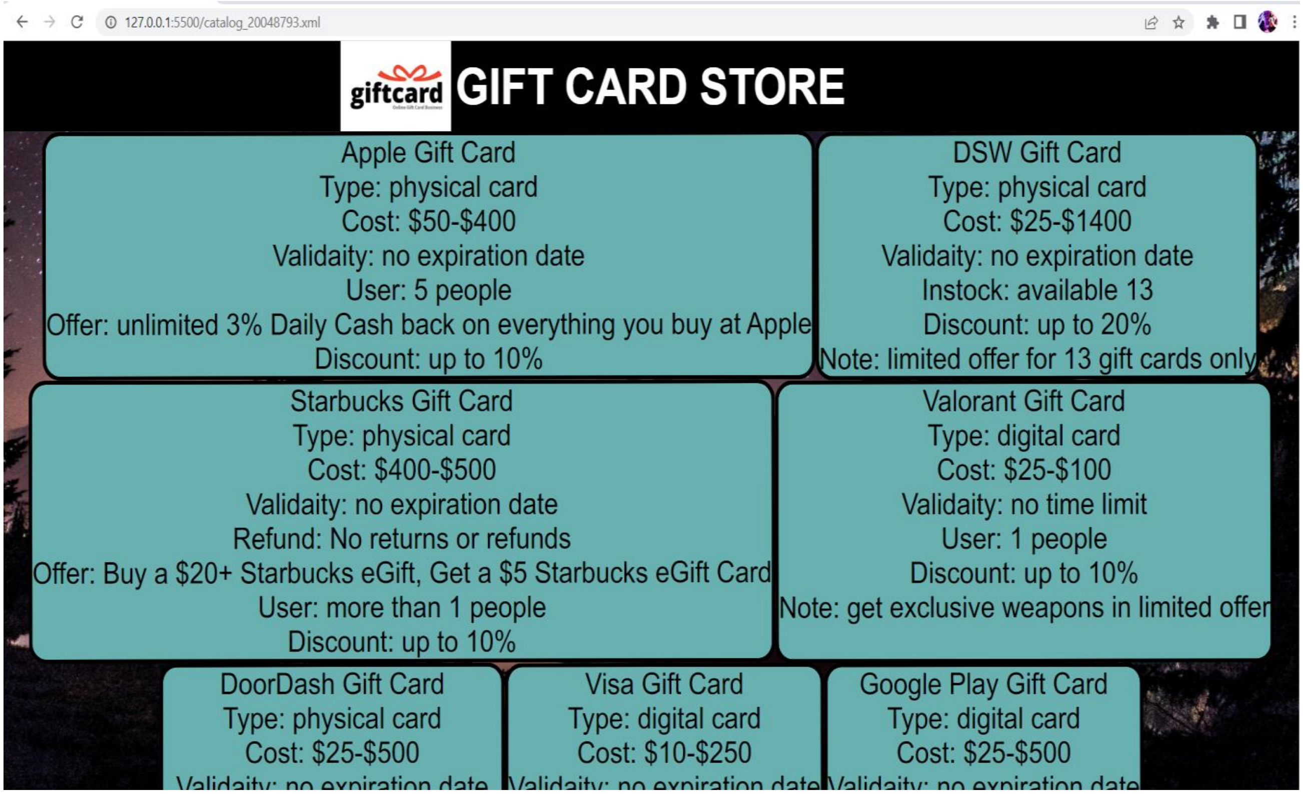
```
error on line 24 at column 13: error parsing attribute name
```

Below is a rendering of the page up to the first error.

While styling the .xml with the help of .css at first css showing some error.


n Terminal Help
catalog_20048793.css - 20048793 Aadesh Gyawali - Visual Studio Code
catalog_20048793.xml
# catalog_20048793.css 8 ×
catalog_20048793.xsd
# catalog_20048793.css > card

```css
    justify-content: center;
}

card:hover {
    background-color: □rgb(0, 0, 0);
    cursor: pointer;
}

card {
    border: 4px groove □rgb(0, 0, 0);
    font-family: Arial,Helvetica,sans-serif;
    display: flex;
    flex-direction: column;
    align-items: center;
    background: ■rgb(104, 177, 177);
    border-radius: 20px;
    transform: rotate(-0.1deg);
    font-size: 30px;

Cover_Picture {

    border: 4px groove rgb(0, 0, 0);
    border-radius: 20px;
    width: 300px;
    height: 200px;
    background-size: cover;
    font-size: 100px;
}

Name {
    font-weight: bold;
    text-align: center;
    color: ■rgb(245, 255, 54);
    font-family: 'Roboto', sans-serif;
}
```



Apple Gift Card
Type: physical card
Cost: $50-$400
Validaity: no expiration date
User: 5 people
Offer: unlimited 3% Daily Cash back on everything you buy at Apple
Discount: up to 10%

DSW Gift Card
Type: physical card
Cost: $25-$1400
Validaity: no expiration date
Instock: available 13
Discount: up to 20%
Note: limited offer for 13 gift cards only

Starbucks Gift Card
Type: physical card
Cost: $400-$500
Validaity: no expiration date
Refund: No returns or refunds
Offer: Buy a $20+ Starbucks eGift, Get a $5 Starbucks eGift Card
User: more than 1 people
Discount: up to 10%

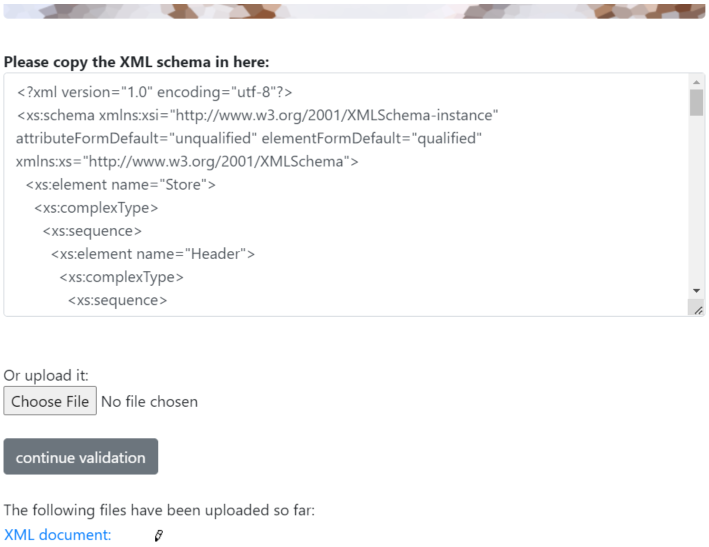
Valorant Gift Card
Type: digital card
Cost: $25-$100
Validaity: no time limit
User: 1 people
Discount: up to 10%
Note: get exclusive weapons in limited offer

DoorDash Gift Card
Type: physical card
Cost: $25-$500
Validaity: no expiration date

Visa Gift Card
Type: digital card
Cost: $10-$250
Validaity: no expiration date

Google Play Gift Card
Type: digital card
Cost: $25-$500
Validaity: no expiration date

Also while validating xml and xsd in the validator many error were seen.



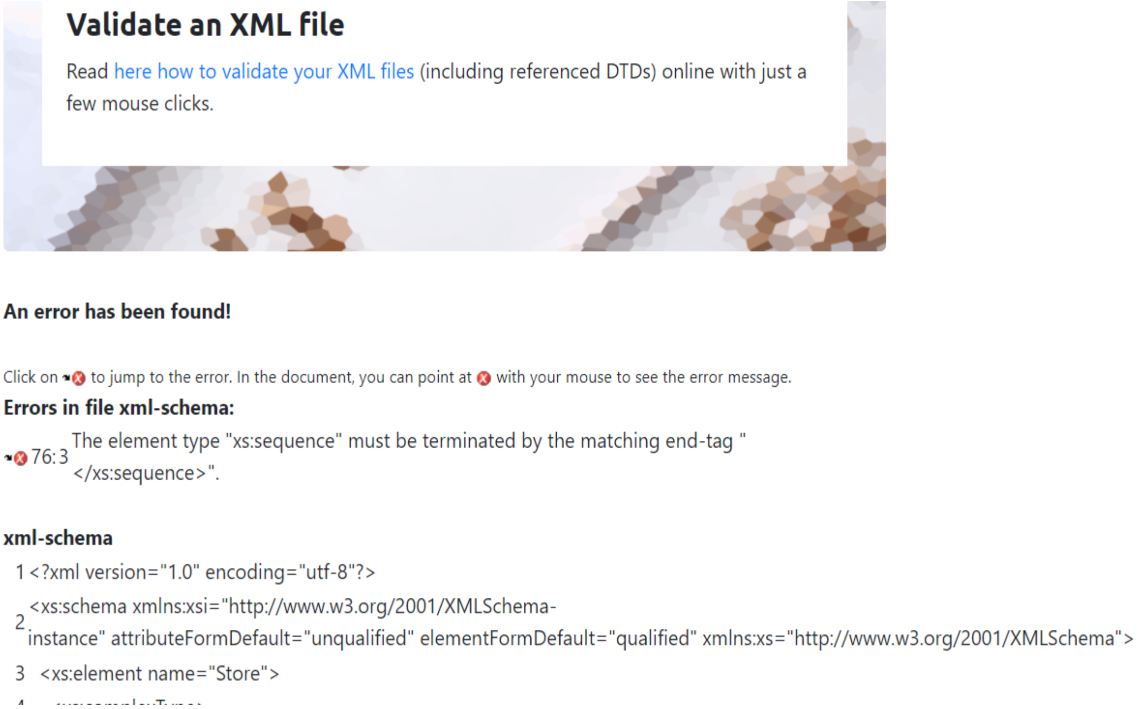
Please copy the XML schema in here:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Store">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
```

Or upload it:

The following files have been uploaded so far:

XML document:



Validate an XML file

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

An error has been found!

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

Errors in file xml-schema:

76:3 The element type "xs:sequence" must be terminated by the matching end-tag "</xs:sequence>".

xml-schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Store">
```

8.Conclusion

A model system for a gift card store named “Gift Card Store” was developed and the software requirements were noted for the model system. The details of the store included the Store Name, Address, Telephone Number, Website Address, Singers, release year, and so on. The gift card has properties like type, cost, validity, number of users, discount vouchers and so on. At least 15 data, 5 attributes, and 5 optional elements were included in the document. A tree diagram was made to show the structure of the XML document. An XML document catalog_20048793.xml, a schema file name catalog_20048793.xsd and a CSS file name catalog_20048793.css was created.

The. XML files contained the content of the document, the.XSD and validation and the. CSS file was for styling the document. The XML file was validated against the Schema files using an online validator http://www.xmlvalidation.com/ . The CSS file contained a list of the genre, border of the song panel, different font sizes, and two different font family Poppins and Roboto. Different rgb color were implemented. With the helpful guidance from our module teachers. I was successful development the xml file .

References

tutorialpoint, 2022. *tutorialpoint.* [Online]
Available at: tutorialpoint
[Accessed 4 may 2022].
Visual Studio Code, 2022. [Online]
Available at: https://code.visualstudio.com/docs/supporting/FAQ
[Accessed 5 may 2022].
w3c, 2022. *w3c.* [Online]
Available at: https://www.w3.org/standards/xml/core
[Accessed 4 may 2022].

Appendix

CSS

```css
Header {
    display: flex;
    align-items: center;
    /* justify-content: space-around; */
    background-color: rgb(0, 0, 0);
    height: 12vh;
    max-width: 100%;
    font-family: 'Roboto', sans-serif;
    position: relative;
}

/* Defining Footer Section */
Footer {
    padding: 10px;
    margin-top: 20px;
    background-color: rgb(64, 60, 67);
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: space-around;

}

Store_Name {
    color: rgb(255, 255, 255);
    font-size: 50px;
    position: absolute;
    right: 35%;
    font-family: 'Poppins', sans-serif;
    font-weight: 600;
}

Copyright {
    padding-bottom: 10px;
    margin-left: -80px;
    font-family: 'Roboto', sans-serif;
    font-size: 25px;
    color: rgb(0, 0, 0);
}

Address,Telephone,Website {
    padding-bottom: 10px;
    margin-left: -80px;
    font-family: 'Roboto', sans-serif;
    font-size: 25px;
    color: rgb(0, 0, 0);
}

/*yo kinaa hoo hauuu
/* Padding for Footer */
Footer Website {
    padding-top: -19px;
}

Logo {
    background-image: url(./image/logo.jpg);
    background-size: cover;
    height: 90px;
    width: 130px;
    position: absolute;
    left: 26%;
}

Store{

    background-image: url(./image/background.jpg);
    background-color: #cccccc;
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-position: center;
    background-size: cover;
    position: relative;

}
```

```

Cards {
    display: flex;
    flex-wrap: wrap;
    justify-content: center;
}

card:hover {
    background-color: rgb(0, 0, 0);
    cursor: pointer;
}

card {
    border: 4px groove rgb(0, 0, 0);
    font-family: Arial,Helvetica,sans-serif;
    display: flex;
    flex-direction: column;
    align-items: center;
    background: rgb(104, 177, 177);
    border-radius: 20px;
    transform: rotate(-0.1deg);
    font-size: 30px;
    margin: 20px;
    width: 550px;
    height: 520px;
        padding: 20px;
        color: #ffffff;
        grid-gap: 5px;
        transition: background-color 0.4s ease-out;
        text-align: center;
    }

    Cover_Picture {

        border: 4px groove rgb(0, 0, 0);
        border-radius: 20px;
        width: 300px;
        height: 200px;
        background-size: cover;
        font-size: 100px;
    }

    Name {
        font-weight: bold;
        text-align: center;
        color: rgb(245, 255, 54);
        font-family: 'Roboto', sans-serif;
    }

    /* Cover Photo for each card */
    #cover_1 {
        background-image: url(./image/card_1.jpg);
    }

    #cover_2 {
        background-image: url(./image/card_2.jpeg);
    }

    #cover_3 {
        background-image: url(./image/card_3.jpeg);
    }

    #cover_4 {
        background-image: url(./image/card_4.jpg);
    }

    #cover_5 {
        background-image: url(./image/card_5.png);
    }

    #cover_6 {
        background-image: url(./image/card_6.jpeg);
    }

    #cover_7 {
        background-image: url(./image/card_7.png);
    }

    #cover_8 {
        background-image: url(./image/card_8.jpeg);
    }

    #cover_9 {
        background-image: url(./image/card_9.png);
    }

    #cover_10 {
        background-image: url(./image/card_10.jpeg);
    }

    #cover_11 {
        background-image: url(./image/card_11.jpeg);
    }

    #cover_12 {
        background-image: url(./image/card_12.jpeg);
    }

    #cover_13 {
        background-image: url(./image/card_13.webp);
    }

    #cover_14 {
        background-image: url(./image/card_14.png);
    }

    #cover_15 {
        background-image: url(./image/card_15.jpg);
    }

```