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: Priyanka Singh Thakuri

London Met ID: 20048967

College ID: NP01CP4S210331

Assignment Due Date: 5th May 2022

Assignment Submission Date: 5th May 2022

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

Table of Figures:

Table of Tables:

Introduction

A system model for a gift card store has been developed after an interview with the store manager. After collecting all the required details of the store, a system was modelled successfully.

A gift store is situated in London, United Kingdom and it sells digital gift cards or gift vouchers which is a form of payment that can be used to make purchases at retail stores, gas stations, restaurants, and other locations. Some details of store had been noted, some of the are store name, address, telephone number, website address and logo. Along with the store detail, card properties like type, cost, validity, number of users, discount vouchers noted to develop better model for the store.

Developed model meets all the requirement that had been noted during requirement analysis phase. It contains the name and logo of the store at the top of the webpage. It contains the some of the samples of the gift cards with all the properties of respective cards as per the requirements of the store.

An XML document catalog_20048967.xml, a schema file name catalog_20048967.xsd and a CSS file name catalog_20048967.css has been created. The XML file contain the content of the document, the .XSD file are for validation and the .CSS file is for styling the document.

Here, XML file has been created to validate against the Schema file by the use of online XML validator, while CSS has been created to style the webpage to make it more attractive and manageable.

XML

XML (eXtensible Markup Language) is a meta-language (a language used to describe other languages) for defining vocabularies (custom markup languages), which is the key to XML's importance and popularity. XML-based vocabularies (such as XHTML) let you describe documents in a meaningful way.

XML vocabulary documents are like HTML documents in that they are text-based and consist of markup (encoded descriptions of a document's logical structure) and content (document text not interpreted as markup). Markup is evidenced via tags (angle bracket–delimited syntactic constructs), and each tag has a name. Furthermore, some tags have attributes (name/ value pairs). (Jeff, 2019).

XML Code

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<?xml-stylesheet href="catalog_20048967.css" type="text/css"?>
<Store xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="catalog_20048967.xsd">
    <header>
        <!-- Navigation Section of the webpage -->
        <Store_Name>Gift Cards Closet</Store_Name>
        <Logo id="Store_logo"/>
    </header>

    <!-- Title section of the webpage -->
    <Title_1>Our Best</Title_1>
    <Title_2>Sellings....</Title_2>

    <!-- Body section of the webpage -->
    <GiftCards>
        <!-- Card section of webpage -->
        <card cardID="GC01123">
            <cardPicture id="card_1"/>
            <card_name available_Qnty="5" cardtype="Digital">Xbox Live Gold 14 days TRIAL Xbox
Live Key GLOBAL</card_name>
            <e-code>E-Code: XboxGC</e-code>
            <cost currency="USD">Cost: $50</cost>
            <number_of_user>Users: 1K</number_of_user>
            <discount_voucher>Discount: Available</discount_voucher>
            <Validity>Valid Till 6 month of purchase Date</Validity>
            <Delivery_time>Delivery duration: 3-5 Days</Delivery_time>
        </card>

        <card cardID="GC01124">
            <cardPicture id="card_2"/>
            <card_name available_Qnty="2" cardtype="Digital">Amazon Gift Card</card_name>
            <e-code>E-Code: AmazonGC</e-code>
            <cost currency="GBP">Cost: £70</cost>
            <number_of_user>Users: 4K</number_of_user>
            <discount_voucher>Discount: Available</discount_voucher>
```

```
        <Validity>Valid Till 9 month of purchase Date</Validity>
        <Delivery_time>Delivery duration: 1-2 Days</Delivery_time>
    </card>

    <card cardID="GC01125">
        <cardPicture id="card_3"/>
        <card_name cardtype="Physical">HULU gift card</card_name>
        <e-code>E-Code: HULUGC</e-code>
        <cost currency="USD">Cost: 50</cost>
        <number_of_user>Users: 2K</number_of_user>
        <discount_voucher>Discount: Unavailable</discount_voucher>
        <Validity>Valid Till 14 month of purchase Date</Validity>
    </card>

    <card cardID="GC01126">
        <cardPicture id="card_4"/>
        <card_name available_Qnty="1" cardtype="Digital">Garena Free Fire</card_name>
        <e-code>E-code: FreeFireGC</e-code>
        <cost currency="GBP">Cost: £60</cost>
        <Diamond>Offer Diamond: 1080</Diamond>
        <Diamond>Bonus Diamond: 60</Diamond>
        <number_of_user>Users: 1K</number_of_user>
        <discount_voucher>Discount: Unavailable</discount_voucher>
        <Validity>Valid Till 2 month of purchase Date</Validity>
        <Delivery_time>Delivery duration: 4-5 Days</Delivery_time>
    </card>

    <card cardID="GC01127">
        <cardPicture id="card_5"/>
        <card_name cardtype="Physical">PUBG Mobile UC</card_name>
        <e-code>E-code: PUBGGC</e-code>
        <cost currency="USD">Cost: $80</cost>
        <UC>Offer UC: 600</UC>
        <UC>Bonus UC: 60</UC>
        <discount_voucher>Discount: Available</discount_voucher>
        <Validity>Valid Till 16 month of purchase Date</Validity>
        <Delivery_time>Delivery duration: 1-2 Days</Delivery_time>
    </card>

    <card cardID="GC01128">
        <cardPicture id="card_6"/>
        <card_name available_Qnty="2" cardtype="Digital">Valorant Gift Card</card_name>
        <e-code>E-code: ValorantGC</e-code>
        <cost currency="USD">Cost: $100</cost>
        <number_of_user>Users: 4K</number_of_user>
        <discount_voucher>Discount: Available</discount_voucher>
        <Validity>Valid Till 8 month of purchase Date</Validity>
    </card>

    <card cardID="GC01129">
        <cardPicture id="card_7"/>
        <card_name cardtype="Physical">XSplit - 1 Year Premium Key GLOBAL</card_name>
        <e-code>E-code: ASplitGC</e-code>
        <cost currency="USD">Cost: $50</cost>
        <discount_voucher>Discount: Available</discount_voucher>
        <Validity>Valid Till 9 month of purchase Date</Validity>
        <Delivery_time>Delivery duration: 3-5 Days</Delivery_time>
    </card>

    <card cardID="GC01130">
        <cardPicture id="card_8"/>
```

```
            <card_name available_Qnty="5" cardtype="Digital">PlayerUnknown's Battlegrounds -
Survivor Pass 7: Cold Front (DLC) Steam Key GLOBAL</card_name>
            <e-code>E-code: PUBGS7GC</e-code>
            <cost currency="USD">Cost: $60</cost>
            <number_of_user>Users: 8K</number_of_user>
            <discount_voucher>Discount: Unavailable</discount_voucher>
            <Validity>Valid Till 4 month of purchase Date</Validity>
            <Delivery_time>Delivery duration: 1-2 Days</Delivery_time>
        </card>

        <card cardID="GC01131">
            <cardPicture id="card_9"/>
            <card_name available_Qnty="5" cardtype="Physical">Nexon Karma Koin</card_name>
            <e-code>E-code: NexonGC</e-code>
            <cost currency="USD">Cost: $50</cost>
            <number_of_user>Users: 1.6K</number_of_user>
            <discount_voucher>Discount: Unavailable</discount_voucher>
            <Validity>Valid Till 24 month of purchase Date</Validity>
            <Delivery_time>Delivery duration: 8-9 Days</Delivery_time>
            <Status>Active Status: Can't be activated in North Asia</Status>
        </card>

        <card cardID="GC01132">
            <cardPicture id="card_10"/>
            <card_name cardtype="Digital">Twitch Gift Card</card_name>
            <e-code>E-code: TwitchGC</e-code>
            <cost currency="GBP">Cost: £25</cost>
            <discount_voucher>Available</discount_voucher>
            <Validity>Valid Till 8 month of purchase Date</Validity>
            <Delivery_time>Days: 1-5 Days</Delivery_time>
        </card>

        <card cardID="GC01133">
            <cardPicture id="card_11"/>
            <card_name available_Qnty="1" cardtype="Digital">Fortnite - 13500 V-Bucks Gift
Card Key GLOBAL</card_name>
            <e-code>E-code: FortniteGC</e-code>
            <cost currency="GBP">Cost: £30</cost>
            <number_of_user>Users: 2K</number_of_user>
            <discount_voucher>Discount: Available</discount_voucher>
            <Validity>Valid Till 11 month of purchase Date</Validity>
        </card>

        <card cardID="GC01134">
            <cardPicture id="card_12"/>
            <card_name available_Qnty="5" cardtype="Digital">Cyberpunk 2077 GOG Key
GLOBAL</card_name>
            <e-code>E-code: CPunkGC</e-code>
            <cost currency="GBP">Cost: £25</cost>
            <number_of_user>Users: 4K</number_of_user>
            <discount_voucher>Discount: Uvailable</discount_voucher>
            <Validity>Valid Till 3 month of purchase Date</Validity>
            <Status>Active Status: Can only activated in UK and USA</Status>
        </card>

        <card cardID="GC01135">
            <cardPicture id="card_13"/>
            <card_name available_Qnty="8" cardtype="Digital">iTunes Gift Card</card_name>
            <e-code>E-code: iTunesGC</e-code>
            <cost currency="USD">Cost: $25</cost>
            <discount_voucher>Discount: Unavailable</discount_voucher>
```

```
            <Validity>Valid Till 10 month of purchase Date</Validity>
             <Delivery_time>Delivery duration: 3-5 Days</Delivery_time>
        </card>

        <card cardID="GC01136">
            <cardPicture id="card_14"/>
            <card_name available_Qnty="5" cardtype="Digital">Spotify Gift Card</card_name>
            <e-code>E-code: SpotifyGC</e-code>
            <cost currency="USD">Cost: $10</cost>
            <number_of_user>Users: 3K</number_of_user>
            <discount_voucher>Discount: Available</discount_voucher>
            <Validity>Valid Till 6 month of purchase Date</Validity>
             <Delivery_time>Delivery duration: 1-3 Days</Delivery_time>
        </card>

        <card cardID="GC01137">
            <cardPicture id="card_15"/>
            <card_name available_Qnty="5" cardtype="Physical">eBay Gift Card</card_name>
            <e-code>E-code: eBayGC</e-code>
            <cost currency="GBP">Cost: £100</cost>
            <number_of_user>Users: 3K</number_of_user>
            <discount_voucher>Discount: Available</discount_voucher>
            <Validity>Valid Till 12 month of purchase Date</Validity>
            <Status>Active Status: Can't be activated in Asia</Status>
        </card>

    </GiftCards>
    <!-- Summary of the store -->
    <Summary>
        <![CDATA[Rewarding employees and thanking customers is now easier with The Brookfield
Properties Gift Card.
            With thousands of shopping, dining and other entertainment options from which they
can choose, you
            can please your staff and customers with the Brookfield Properties Gift Card.

            The Brookfield Properties Gift Card is valid virtually anywhere United Kingdom
Express® Cards are accepted within the U.K.,
            like movie theaters, restaurants, retail shops and more, including those retailers
at the more than 100 Brookfield Properties
            locations nationwide.

            Use Gift Card only at US merchants that accept American Express® Cards, except for
recurring payment. No ATM use. Funds do not expire.
            Not redeemable for cash, except where required by law. Additional limitations
apply. See Cardholder Agreement for complete details.
            Gift Card is issued by UK Express Prepaid Card Management Corporation.]]>

    </Summary>

    <!-- Footer section of the webpage -->
    <footer>
        <website>http://www.giftcardscloset.com</website>
        <Address>Address: The Wheatsheaf, Briggate,Elland HX5 9HG, United Kingdom</Address>
        <Telepnohe_number>Contact Us: 01-6547789 & 9841862549</Telepnohe_number>
        <Email_Address>Email ID: www.giftcardscloset@gmail.com</Email_Address>
        <Copyright>Copyright © 2022  Gift Cards Closet. All rights reserverd.</Copyright>
    </footer>

</Store>
```

List of Data

It contains the list of all the data present in XML document. Data like elements and attributes and its properties are present.

List of Elements

List of Elements	Element Type	Element Data	Occurrence
Header	Complex Type	Child Elements	Only once
Store_Name	Simple Type	String	Only once
Logo	Complex Type	Empty Element	Only once
Title_1	Simple Type	String	Only once
Title_2	Simple Type	String	Only once
GiftCards	Complex Type	Child Elements	Only once
card	Complex Type	Child elements	One or more
cardPicture	Complex Type	Empty Element	One or more
card_name	Complex Type	String	One or more
e-code	Simple Type	String	One or more
cost	Complex Type	String	One or more
Number_of_user	Simple Type	String	Zero or more
discount_voucher	Simple Type	String	One or more
validity	Simple Type	String	One or more
Delivery_time	Simple Type	String	Zero or more
Diamond	Simple Type	String	Zero or more
UC	Simple Type	String	Zero or more
Status	Simple Type	String	Zero or more
Summary	Simple Type	String	Only once
footer	Complex Type	Child Elements	Only once
website	Simple Type	String	Only once
Telephone_number	Simple Type	String	Only once

Email_Address	Simple Type	String	Only once
Copyright	Simple Type	String	Only once

Table 1: List of Elements

List of Attributes

List of Attributes	Data type	Default	Fixed	Use
L_id	String	-	-	Required
cardID	String	-	-	Required
id	String	-	-	Required
available_qnty	positiveInteger	1	-	Optional
cardtype	String	-	-	Required
currency	String	-	-	Required

Table 2: List of Attributes

XML Tree Structure

XML documents have a self-describing structure. It forms a tree structure called an XML tree. The tree structure contains root elements (as parents), child elements, and so on. Starting from the root and traversing all subsequent branches, subbranches, and leaf nodes is very easy.

XML file has a tree structure, wherein the foundation detail is on the top and the kid factors are related to the foundation factors, the equal way, how leaves are related to tree thru branches. (Rudd, 2020).

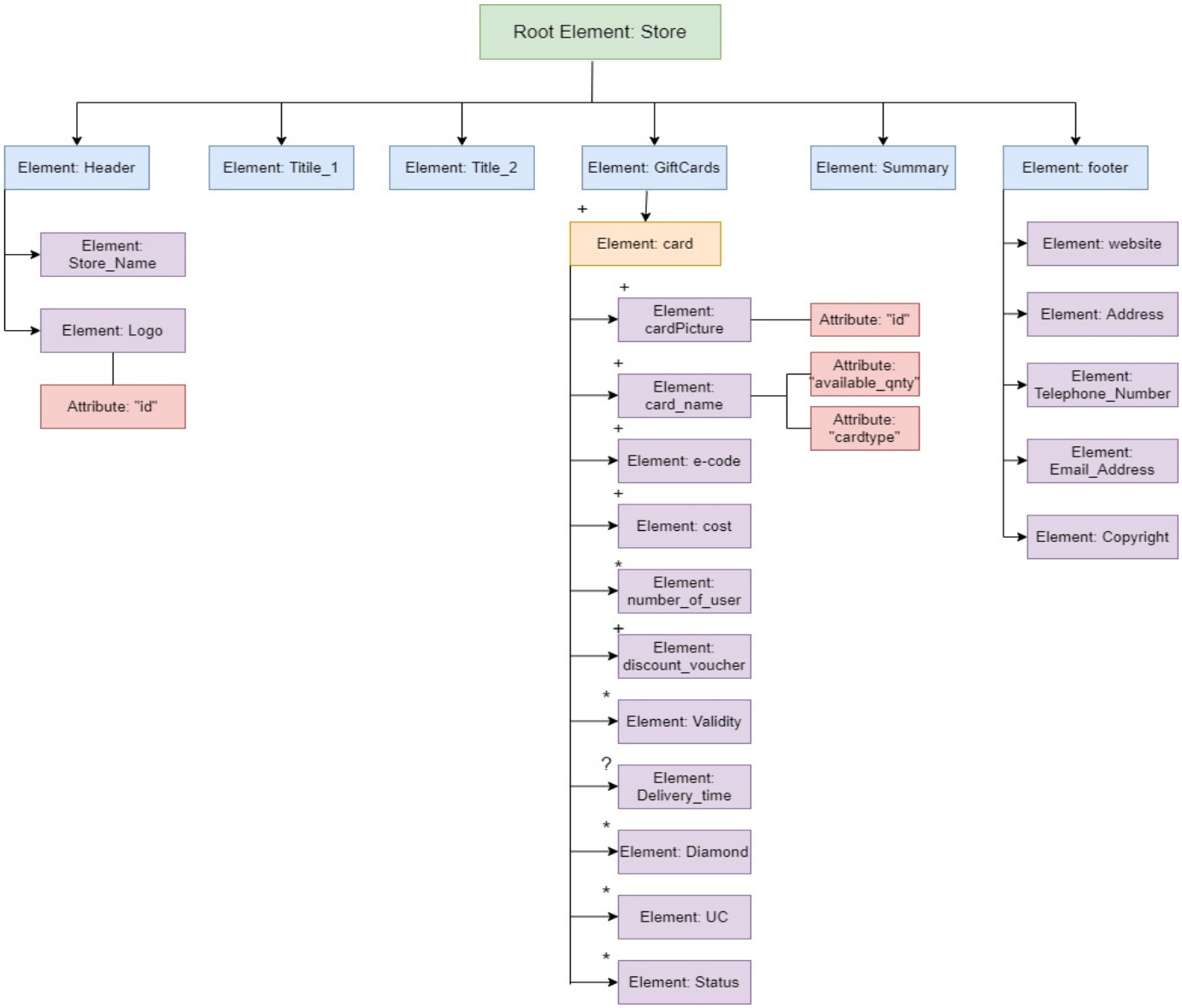

Figure 1: XML Tree structure

Schema

An XML Schema is an XML document that contains a formal description of what it is. Consists of valid XML documents. The schema of the W3CXML schema language is An XML schema written in a special syntax recommended by the W3C.

XML documents written by the schema are called instance documents. when The document is considered to meet all the restrictions specified in the schema The schema is valid. The schema document is associated with the instance document In one of the following ways:

- The element's xsi: schemaLocation attribute contains a list of namespaces. The URL of the schema used within that element and used to validate the elements and attributes of those namespaces.
- xsi: The noNamespaceSchemaLocation attribute contains the URL of the schema Used to validate elements that are not in the namespace. • The validation parser can instruct you to validate a particular document. Ignore the explicitly provided schema and hints in it the document itself.

XML Schema of the Store:

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

<!-- schema declaration -->
<xs:schema 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="Store_Name" type="xs:string"/>
                            <xs:element name="Logo">
                                <xs:complexType>
                                    <xs:simpleContent>
                                        <xs:extension base="xs:string">
                                            <xs:attribute type="xs:string" name="id"/>
                                        </xs:extension>
                                    </xs:simpleContent>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="Title_1" type="xs:string"/>
                <xs:element name="Title_2" type="xs:string"/>
                <xs:element name="GiftCards" >
                    <xs:complexType>
```

```
                        <xs:sequence>
                            <xs:element name="card" minOccurs="1" maxOccurs="unbounded">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="cardPicture">
                                            <xs:complexType>
                                                <xs:simpleContent>
                                                    <xs:extension base="xs:string">
                                                        <xs:attribute type="xs:string"
name="id" use="optional"/>
                                                    </xs:extension>
                                                </xs:simpleContent>
                                            </xs:complexType>
                                        </xs:element>
                                        <xs:element name="card_name">
                                            <xs:complexType>
                                                <xs:simpleContent>
                                                    <xs:extension base="xs:string">
                                                        <xs:attribute name="available_Qnty"
type="xs:string" use="optional"/>
                                                        <xs:attribute name="cardtype"
type="xs:string" use="optional"/>
                                                    </xs:extension>
                                                </xs:simpleContent>
                                            </xs:complexType>
                                        </xs:element>
                                        <xs:element name="e-code" type="xs:string"/>
                                        <xs:element name="cost">
                                            <xs:complexType>
                                                <xs:simpleContent>
                                                    <xs:extension base="xs:string">
                                                        <xs:attribute name="currency"
use="required">
                                                            <xs:simpleType>
                                                                <xs:restriction
base="xs:string">
                                                                    <xs:enumeration
value="USD"/>
                                                                    <xs:enumeration
value="GBP"/>
                                                                </xs:restriction>
                                                            </xs:simpleType>
                                                        </xs:attribute>
                                                    </xs:extension>
                                                </xs:simpleContent>
                                            </xs:complexType>
                                        </xs:element>
                                        <xs:element type="xs:string" name="Diamond"
maxOccurs="unbounded" minOccurs="0"/>
                                        <xs:element type="xs:string" name="number_of_user"
minOccurs="0"/>
                                        <xs:element type="xs:string" name="UC"
maxOccurs="unbounded" minOccurs="0"/>
                                        <xs:element type="xs:string" name="discount_voucher"/>
                                        <xs:element type="xs:string" name="Validity"/>
                                        <xs:element type="xs:string" name="Delivery_time"
minOccurs="0"/>
                                        <xs:element type="xs:string" name="Status"
minOccurs="0"/>
                                    </xs:sequence>
```

```
                                    <xs:attribute name="cardID" type="xs:string"
use="required"/>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="Summary" type="xs:string"/>
                <xs:element name="footer">
                                <xs:complexType>
                                    <xs:sequence>
                                       <xs:element name="website">
                                        <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                                <xs:pattern value="http://.+"/>
                                            </xs:restriction>
                                        </xs:simpleType>
                                       </xs:element>
                                       <xs:element name="Address" type="xs:string"/>
                                       <xs:element name="Telepnohe_number" type="xs:string"/>
                                       <xs:element name="Email_Address">
                                        <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                                <xs:pattern value="[^@]+@[^\.]+\..+"/>
                                            </xs:restriction>
                                        </xs:simpleType>
                                       </xs:element>
                                       <xs:element name="Copyright" type="xs:string"/>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

CSS

Fondly called CSS, cascading stylesheets are a simply designed language aimed at simplifying the process of making web pages visible. You can use CSS to apply styles to your web pages. More importantly, CSS allows you to do this independently of the HTML that makes up each web page. CSS is easy to learn and understand, but it gives you a lot of control over the display of your HTML documents. (DivyanshGupta1, 2021).

CSS of the Store:

```
/*

 Coursework 2
 CSS
 Author: Priyanka Singh Thakuri
 Date: 27th April 2022
 Filename: catalog_20048967.css

*/

/* CSS File for styling our Document */
@import
url('https://fonts.googleapis.com/css2?family=Poppins:ital,wght@1,400;1,500;1,600;1,700&family
=Roboto:wght@300;400;500;700&display=swap');

/* The whole Website section */
Store{
    background-color: rgb(15, 82, 186);
    display: block;
    color: #F4F4F4;
    background-image: url(./images/background1.jpg);
}

/* Styling Title section of the webpage */
Title_1, Title_2{
    font-size: 35px;
    margin-top: 5px;
    color: #f4f4f4;
    display: flex;
         flex-direction: column;
         align-items: center;
    justify-content: space-around;
    font-family: Papyrus;
}

Title_2{
    font-size: 45px;
    color: cyan;
    width: 100%;
    text-shadow: 3px 3px 4px cyan;
}

Title_1{
    width: 80%;
```

```
}

/* Defining Header Section */
header
{
    display: flex;
    align-items: center;

     justify-content: space-around;
    background-color: black;
    height: 12vh;
    max-width: 100%;
    /* font-family: 'Roboto', sans-serif; */
    font-family: Papyrus;
    text-shadow: 3px 3px 3px greenyellow;
    font-size: 25px;
    font-weight: 600;
    color: cyan;
         position: relative;
 }

 /* Styling the Storename of navigation section */
.Store_Name{

    margin-left: 5vw;
    margin-bottom: 20vh;
    position: absolute;
    display: flex;
    width: 80%;
}

/* Styling the Logo of navigation section */
Logo{
    background-image: url(images/images.png);
    background-size: 32%;
    background-repeat: no-repeat;
    height: 15vh;
    width: 30vw;
         position:absolute;
    left: 10vw;
}

/* Styling the Giftcard section */
GiftCards
{
    display: flex;
    flex-wrap: wrap;
    justify-content: center;
}

/* Floating Box for Song Information */
card
{
    border: 4px groove black;
    font-family: 'Roboto', sans-serif;
         display: flex;
         flex-direction: column;
         align-items: center;
         background: rgb(5, 5, 81);
         box-shadow: 0px 4px 4px grey;
         border-radius: 20px;
```

```
        transform: rotate(-0.1deg);
        margin: 20px;
        width: 350px;
        height: 600px;
        padding: 20px;
        color: #f4f4f4;
    grid-gap: 10px;
        transition: background-color 0.4s ease-out;
        text-align: center;
    font-size: 15px;
    transition: 0.5s;
}

/* hover effect of the cards */
card:hover{
    height: 450px;
    width: 250px;
    transform: scale(1.5);
    z-index: 5;
}

card_description{
    font-size: 10px;
}

card_name{
    font-size: 25px;
    color: #FFFF00;
}

card{
    cursor: pointer;
}

/* Embedding the picture of the cards */
#card_1 {
        background-image: url(images/Card_1.jpg);

    background-size: cover;
    background-position: center;
}

#card_2 {
        background-image: url(images/Card_2.jpg);
    background-size: cover;
    background-position: center;
}
#card_3 {
        background-image: url(images/Card_3.webp);
    background-size: cover;
    background-position: center;
}

#card_4 {
        background-image: url(images/card_4.jpg);
    background-size: cover;
    background-position: center;
}
#card_5 {
        background-image: url(images/Card_5.png);
    background-size: cover;
```

```
    background-position: center;
}

#card_6 {
        background-image: url(images/Card_6.jpg);
    background-size: cover;
    background-position: center;
}
#card_7 {
        background-image: url(images/Card_7.webp);
    background-size: cover;
    background-position: center;
}

#card_8 {
        background-image: url(images/Card_8.webp);
    background-size: cover;
    background-position: center;
}
#card_9 {
        background-image: url(images/Card_9.webp);
    background-size: cover;
    background-position: center;
}

#card_10 {
        background-image: url(images/Card_10.jpg);
    background-size: cover;
    background-position: center;
}
#card_11 {
        background-image: url(images/Card_11.webp);
    background-size: cover;
    background-position: center;
}

#card_12 {
        background-image: url(images/Card_12.png);
    background-size: cover;
    background-position: center;
}
#card_13 {
        background-image: url(images/Card_13.png);
    background-size: cover;
    background-position: center;
}

#card_14 {
        background-image: url(images/Card_14.png);
    background-size: cover;
}
#card_15 {
        background-image: url(images/Card_15.jpg);
    background-size: cover;
    background-position: center;
}

/* Styling the pictures of the card section */
cardPicture{
    padding: 0.6rem 1rem;
    display: 100%;
```

```
    width: 100%;
    height: 60vh;

}

/* Listing the Diaminds and UC of the game cards */
UC,Diamond{
    display: list-item;
    list-style-type: square;
    margin-left: 5%;
}


/* Styling the Summary of Body section */
Summary{
    font-size: 20px;
    display:inline-block;
    color: rgb(5, 5, 81);
    text-align: center;
    padding-top: 2vh;
    padding-bottom: 2vh;
    background-color: #F7EDE2;
}

/* Defining Footer Section */
footer
{
    padding: 10px;
    margin-top: 5px;
    background-color: black;
    color: #f4f4f4;
    display: flex;
         flex-direction: column;
         align-items: center;
    justify-content: space-around;

}

/* Padding for Footer */
footer website {
         padding-top: -19px;
}

/* Styling the Footer section */
Address,Telepnohe_number,website,Email_Address
{
    padding-bottom: 10px;
    margin-left: -80px;
    font-family: 'Roboto', sans-serif;
    font-size: 21px;
         color:#FFFF00;
}

Copyright
{
    padding-top: 5vh;
    padding-bottom: 10px;
    margin-left: -80px;
    font-family: 'Roboto', sans-serif;
    font-size: 21px;
}
```

Testing

Testing is a method to check whether the actual program product matches expected requirements and to ensure that program product is Defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of testing is to identify errors, gaps or missing requirements in contrast to actual requirements. (Thomas, 2022).

Test 1: Run XML in browser

Test 1	
Action	Run XML in browser
Expected Result	The XML document should run in browser.
Actual Result	The XML document run in browser.
Conclusion	The test was successful.

Table 3: Test 1 Running XML in browser.

```
▼<Store>
  ▼<header>
     <!--  Navigation Section of the webpage  -->
     <Store_Name>Gift Cards Closet</Store_Name>
     <Logo id="Store_logo"/>
   </header>
   <!--  Title section of the webpage  -->
   <Title_1>Our Best</Title_1>
   <Title_2>Sellings....</Title_2>
   <!--  Body section of the webpage  -->
  ▼<GiftCards>
     <!--  Card section of webpage  -->
    ▼<card cardID="GC01123">
       <cardPicture id="card_1"/>
       <card_name available_Qnty="5" cardtype="Digital">Xbox Live Gold 14 days TRIAL Xbox Live Key GLOBAL</card_name>
       <e-code>E-Code: XboxGC</e-code>
       <cost currency="USD">Cost: $50</cost>
       <number_of_user>Users: 1K</number_of_user>
       <discount_voucher>Discount: Available</discount_voucher>
       <Validity>Valid Till 6 month of purchase Date</Validity>
       <Delivery_time>Delivery duration: 3-5 Days</Delivery_time>
     </card>
    ▼<card cardID="GC01124">
       <cardPicture id="card_2"/>
       <card_name available_Qnty="2" cardtype="Digital">Amazon Gift Card</card_name>
       <e-code>E-Code: AmazonGC</e-code>
       <cost currency="GBP">Cost: £70</cost>
       <number_of_user>Users: 4K</number_of_user>
       <discount_voucher>Discount: Available</discount_voucher>
       <Validity>Valid Till 9 month of purchase Date</Validity>
       <Delivery_time>Delivery duration: 1-2 Days</Delivery_time>
     </card>
    ▼<card cardID="GC01125">
       <cardPicture id="card_3"/>
       <card_name cardtype="Physical">HULU gift card</card_name>
       <e-code>E-Code: HULUGC</e-code>
       <cost currency="USD">Cost: 50</cost>
       <number_of_user>Users: 2K</number_of_user>
       <discount_voucher>Discount: Unavailable</discount_voucher>
       <Validity>Valid Till 14 month of purchase Date</Validity>
     </card>
    ▼<card cardID="GC01126">
       <cardPicture id="card_4"/>
       <card_name available_Qnty="1" cardtype="Digital">Garena Free Fire</card_name>
       <e-code>E-code: FreeFireGC</e-code>
       <cost currency="GBP">Cost: £60</cost>
```

Figure 2: Running XML in browser

Test 2: Validate XML and Schema.

Test 2	
Action	Validating XML and Schema.
Expected Result	The XML and Schema should validate without any errors.
Actual Result	The XML and Schema validate without any errors.
Conclusion	The test was successful.

Table 4: Test 2 Validating XML and Schema

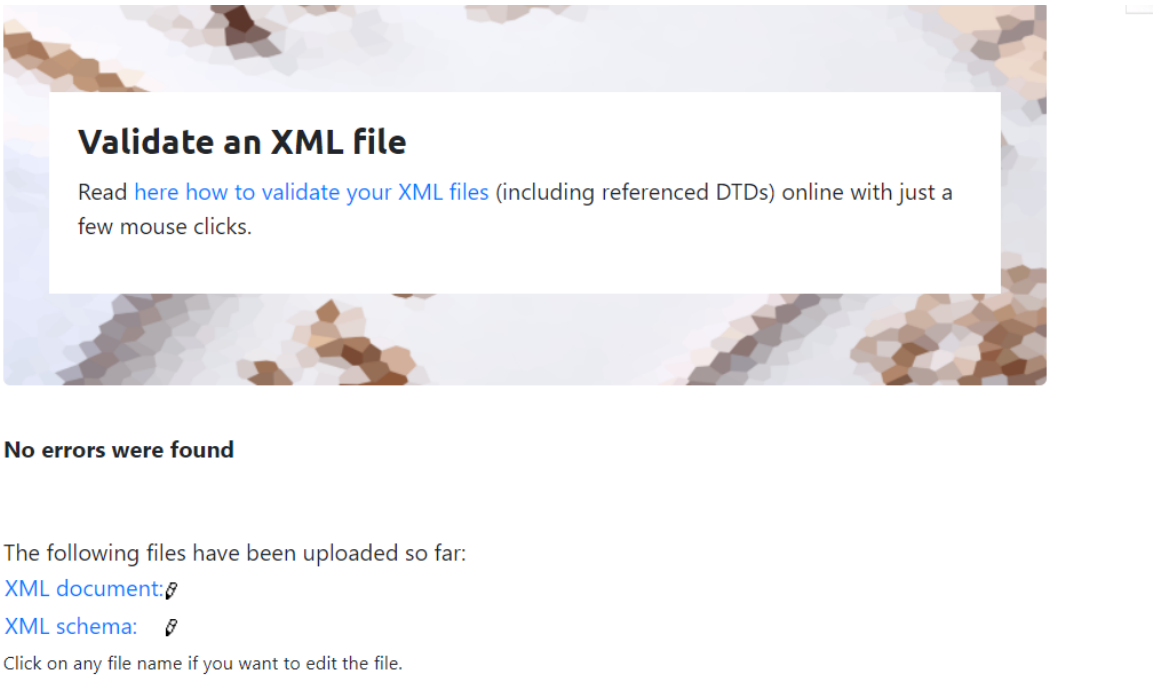

Figure 3: Test 2 Validating XML and Schema

Test 3: Validate XML and CSS

Test 3	
Action	Validating XML and CSS
Expected Result	The XML and CSS should validate in browser.
Actual Result	The XML and CSS was validated.
Conclusion	The test was successful.

Table 5: Test 3 Validating XML and CSS

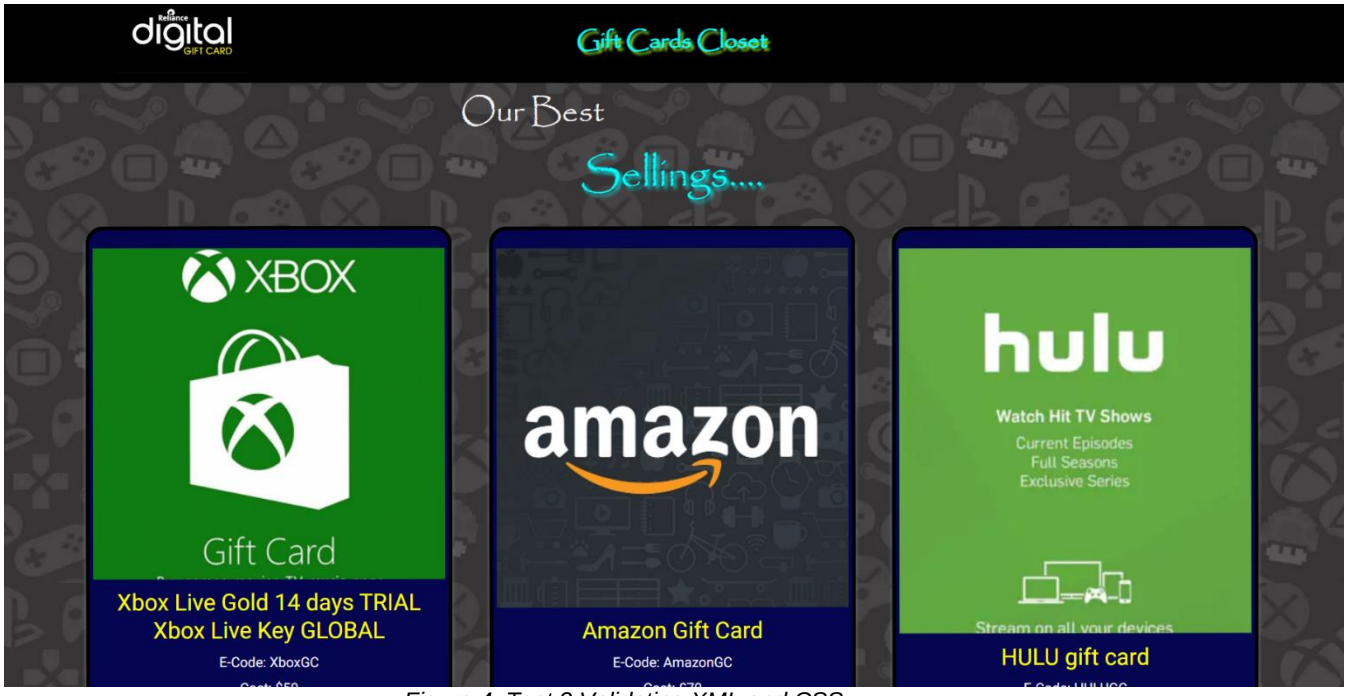

Figure 4: Test 3 Validating XML and CSS

Test 4: Run XML without flexbox in CSS.

Test 4	
Action	Running XML without flexbox in CSS.
Expected Result	The XML should run but the cards picture and properties should be misplaced.
Actual Result	The XML run successfully with the absence of cards picture and properties was misplaced.
Conclusion	The test was successful.

Table 6: Test 4 Running XML without flexbox in CSS.

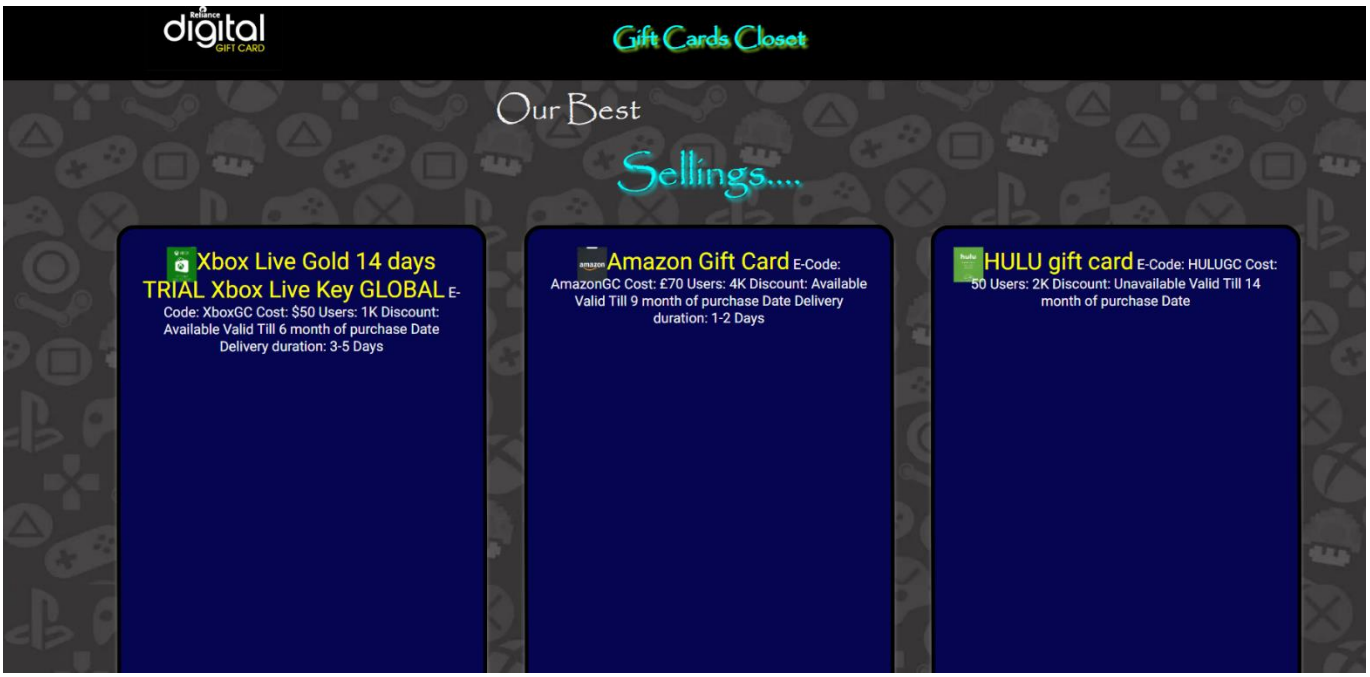

Figure 5: Test 4 Running XML without flexbox in CSS.

Test 5: Load Images from website

Test 5	
Action	Loading Images from website
Expected Result	The images should load from the website.
Actual Result	The images were loaded from the website.
Conclusion	The test was successful.

Table 7: Test 5 Loading Images from website

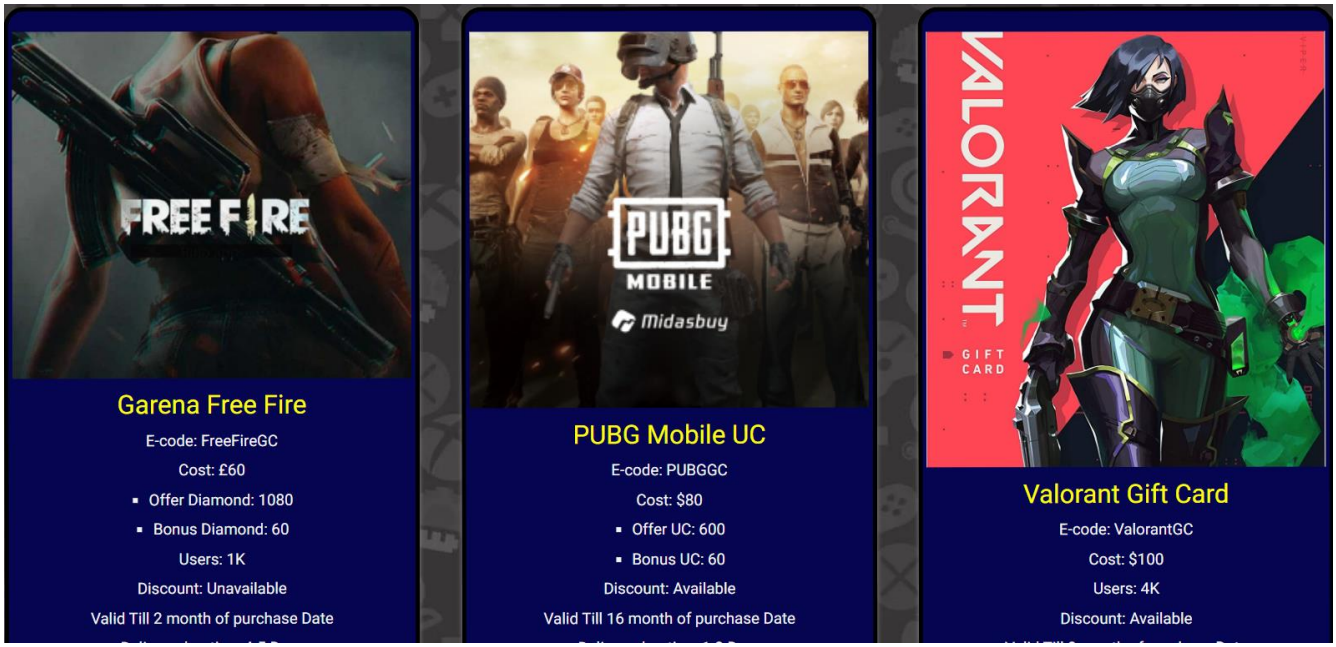

Figure 6: Loading Images from website

Difference between Schema and DTD

DTD or document type definitions and XML Schema (also known as XSD) are two ways to describe the structure and content of an XML document. Since the DTD is the older of the two, there are limits that XML Schema has tried to improve. The first difference between DTDs and XML Schemas is namespace recognition. There is an XML schema, but the DTD is not. Namespace recognition removes the ambiguity that can result from having a particular element and attribute from multiple XML vocabularies by assigning a namespace that places the element or attribute in context.

Part of the reason why XML Schemas are namespaced but DTDs are not namespaced is the fact that XML Schemas are written in XML and DTDs are not. Therefore, like any other XML document, XML Schema can be processed programmatically. Unlike DTDs, XML Schemas are written in XML, so you don't even have to learn another language.

Another important advantage of XML Schema is the ability to implement strong typing. XML Schema can also define the data type of a particular element and limit it to a particular length or value. This feature ensures that the data stored in the XML document is correct. The DTD doesn't have powerful typing capabilities and there is no way to validate the content of a data type. XML Schema has a wealth of derived and built-in data types for validating content. This provides the above benefits. Also, there are unified data types, but older XML parsers often fail because all processors and validators must support these data types.

One property of a DTD that is often considered both an advantage and a disadvantage is the ability to define a DTD inline that is missing from the XML Schema. This is good for working with small files because you can include both the content and the schema in the same document, but for large documents it is a disadvantage because you drag the content each time you fetch the schema. There is a possibility. This can result in significant overhead that can impact performance.

Summary:

1. XML Schema is namespace aware, while DTD is not.
2. XML Schemas are written in XML, while DTDs are not.
3. XML Schema is strongly typed, while DTD is not.
4. XML Schema has a wealth of derived and built-in data types that are not available in DTD.
5. XML Schema does not allow inline definitions, while DTD does. (Joan, 2011).

Coursework Development

An XML document catalog_20048967.xml, a schema file name catalog_20048967.xsd and a CSS file name catalog_20048967.css has been created. The XML file contain the content of the document, the .XSD file are for validation and the .CSS file is for styling the document. Here, XML file has been created to validate against the Schema file by the use of online XML validator, while CSS has been created to style the webpage to make it more attractive and manageable.

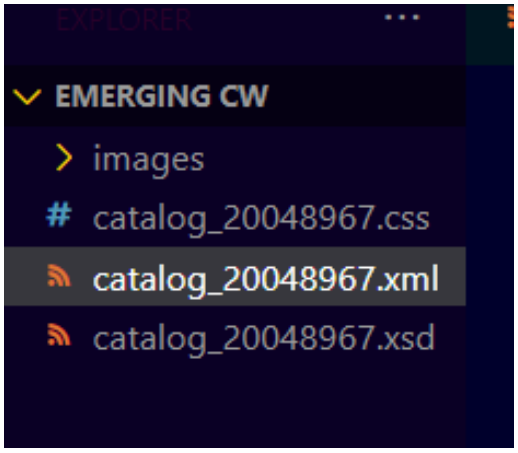

Figure 7: XML, Schema, CSS and Images files in Vs code

According to the provided situations and conditions, a system model for gift card store was developed by the use of Vs code for the purpose of XML, XSD and CSS, while draw.io was used to make the tree diagram of XML document and MS word was used to prepare the document of this coursework.

At first, the XML document was prepared, XML contains the root element store which warp the whole elements of XML documents. It contains the unique elements like Store_Name, Logo, Title_1, Title_2, GiftCards, card and so on. Some unique attributes like id, cardID, available_Qnty, cardtype, currency and optional elements like Delivery_time, Diamond, UC, Status, Title_1 are also present in this document. The sample of XML document is shown below:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<?xml-stylesheet href="catalog_20048967.css" type="text/css"?>
<Store xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="catalog_20048967.xsd">
    <header>
        <!-- Navigation Section of the webpage -->
        <Store_Name>Gift Cards Closet</Store_Name>
        <Logo id="Store_logo"/>
    </header>

    <!-- Title section of the webpage -->
    <Title_1>Our Best</Title_1>
    <Title_2>Sellings....</Title_2>

    <!-- Body section of the webpage -->
    <GiftCards>
        <!-- Card section of webpage -->
        <card cardID="GC01123">
            <cardPicture id="card_1"/>
            <card_name available_Qnty="5" cardtype="Digital">Xbox Live Gold 14 days TRIAL Xbox
Live Key GLOBAL</card_name>
            <e-code>E-Code: XboxGC</e-code>
            <cost currency="USD">Cost: $50</cost>
            <number_of_user>Users: 1K</number_of_user>
            <discount_voucher>Discount: Available</discount_voucher>
            <Validity>Valid Till 6 month of purchase Date</Validity>
            <Delivery_time>Delivery duration: 3-5 Days</Delivery_time>
        </card>

        <card cardID="GC01124">
            <cardPicture id="card_2"/>
            <card_name available_Qnty="2" cardtype="Digital">Amazon Gift Card</card_name>
            <e-code>E-Code: AmazonGC</e-code>
            <cost currency="GBP">Cost: £70</cost>
            <number_of_user>Users: 4K</number_of_user>
            <discount_voucher>Discount: Available</discount_voucher>
            <Validity>Valid Till 9 month of purchase Date</Validity>
            <Delivery_time>Delivery duration: 1-2 Days</Delivery_time>
        </card>
```

Then, XML tree diagram was draw with the help of draw.io, it contains the root element in the top of the diagram then the child elements along with attributes were shown. The tree diagram of XML document is shown below:

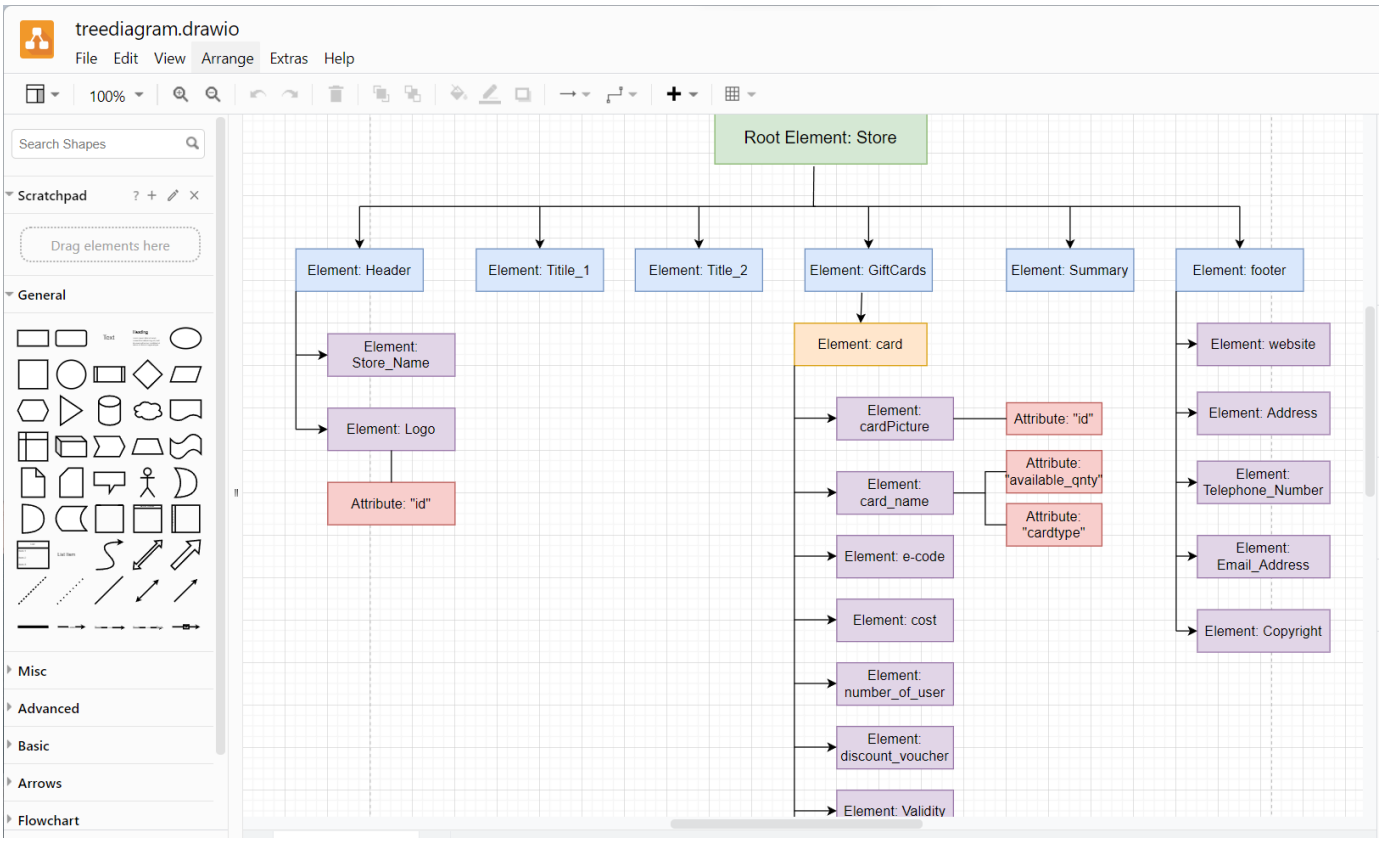

Figure 8: Evidence of tree diagram

Based on the structure of the xml file, external schema file was created named catalog_20048967.xsd where applicable, user-derived data types is included in the schema file to define values for XML elements and attributes. A Russian Doll Design has been used for the schema. This design contains a set of nested declarations. This design makes it easy to associate a schema with an instance document, but it can be confusing and difficult to maintain.

The incomplete and little part of designed schema file is shown below:

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

<!-- schema declaration -->
<xs:schema 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="Store_Name" type="xs:string"/>
                            <xs:element name="Logo">
                                <xs:complexType>
                                    <xs:simpleContent>
                                        <xs:extension base="xs:string">
                                            <xs:attribute type="xs:string" name="id"/>
                                        </xs:extension>
                                    </xs:simpleContent>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="Title_1" type="xs:string"/>
                <xs:element name="Title_2" type="xs:string"/>
                <xs:element name="GiftCards" >
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="card" minOccurs="1" maxOccurs="unbounded">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="cardPicture">
                                            <xs:complexType>
                                                <xs:simpleContent>
                                                    <xs:extension base="xs:string">
                                                        <xs:attribute type="xs:string" name="id"
use="optional"/>

                                                    </xs:extension>
                                                </xs:simpleContent>
                                            </xs:complexType>
                                        </xs:element>
                                        <xs:element name="card_name">
                                            <xs:complexType>
                                                <xs:simpleContent>
                                                    <xs:extension base="xs:string">
                                                        <xs:attribute name="available_Qnty"
type="xs:string" use="optional"/>

                                                        <xs:attribute name="cardtype"
type="xs:string" use="optional"/>

                                                    </xs:extension>
                                                </xs:simpleContent>
                                            </xs:complexType>
                                        </xs:element>
                                        <xs:element name="e-code" type="xs:string"/>
                                        <xs:element name="cost">
                                            <xs:complexType>
                                                <xs:simpleContent>
                                                    <xs:extension base="xs:string">
                                                        <xs:attribute name="currency"
use="required">

                                                            <xs:simpleType>
```

After the creation of schema and XML documents, the validation of XML document against Schema was done through online XML validator. Different errors were encountered during the creation of schema. But after many tries, finally XML was validated against Schema. The evidence of validation with error and no error is shown below:

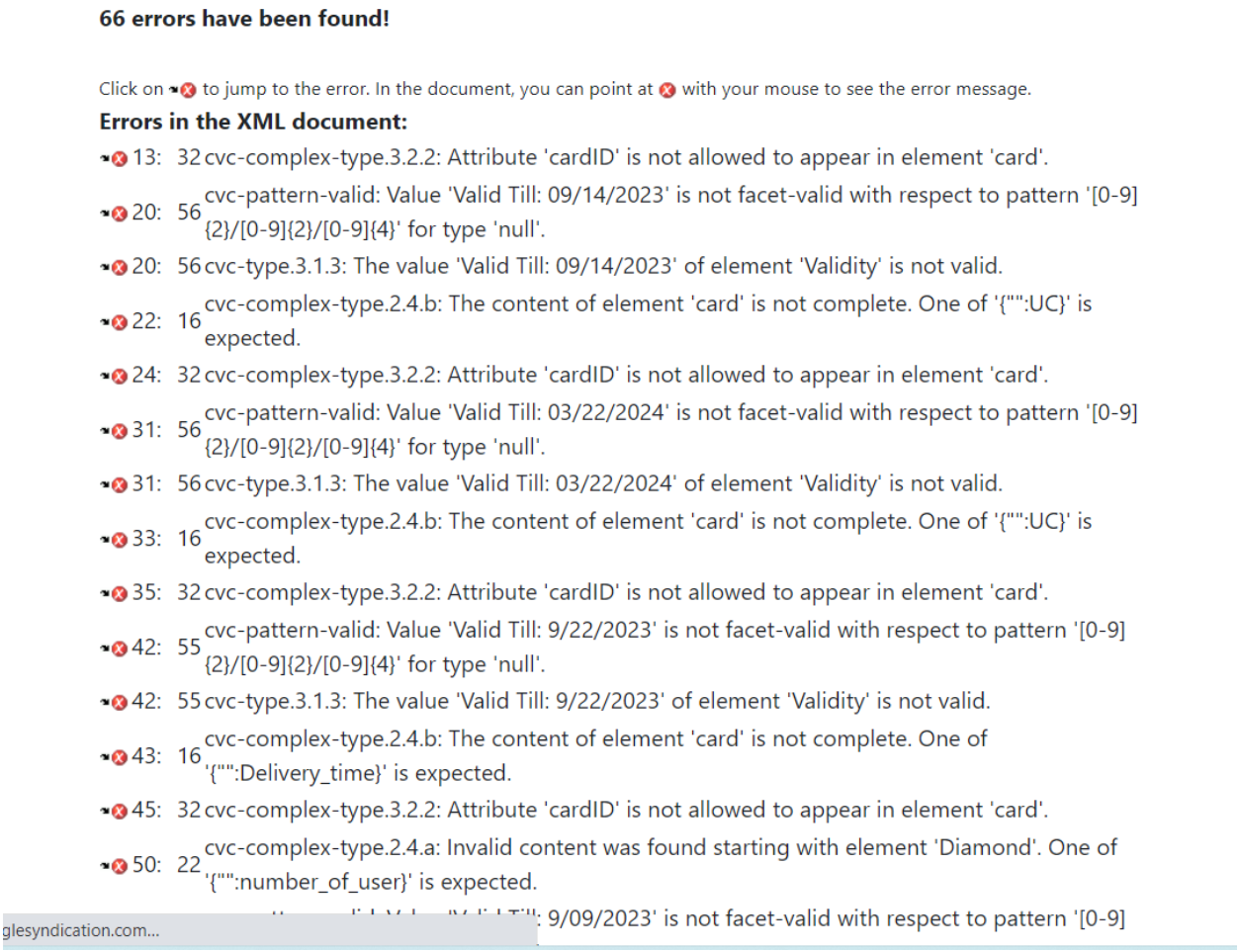

Figure 9: Validation with errors

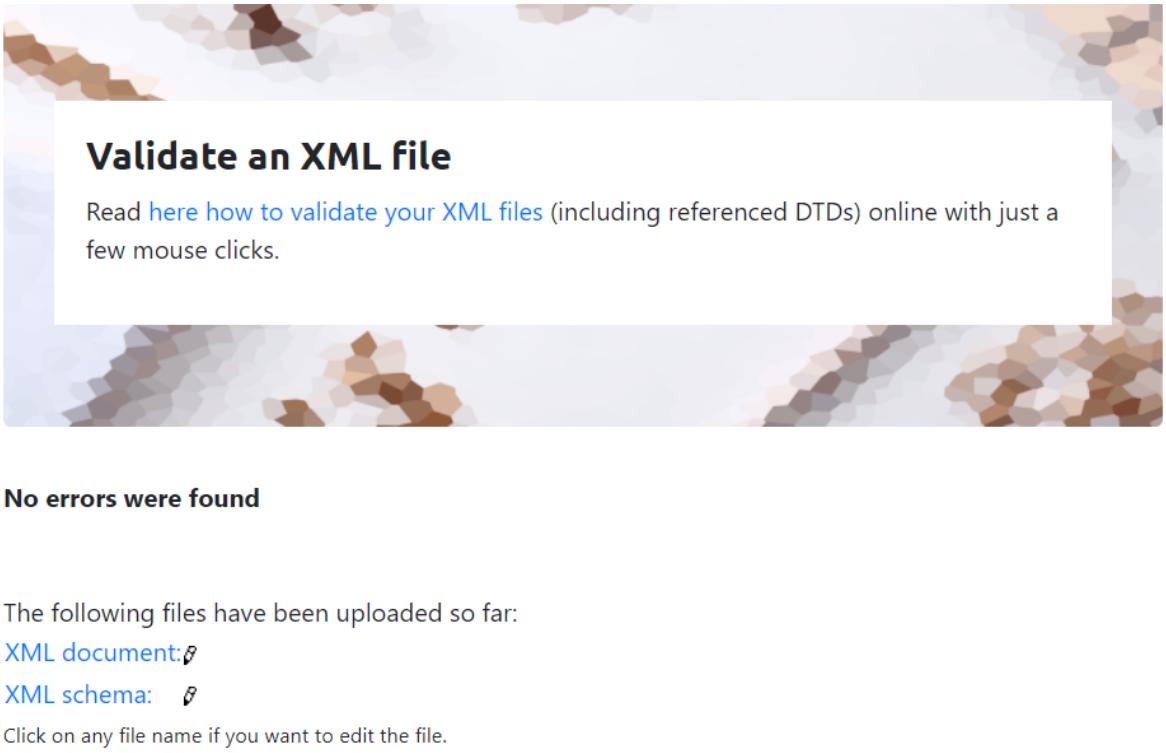

Figure 10: Validation with no errors

A CSS file called catalog_20048967.css was created to render the XML data file in a web browser. The CSS styled the webpage and made it more attractive and manageable. The CSS file meets all the criteria mentioned in coursework requirements. It contains the list for the UC and Diamonds of PUBG and Freefire game. Different font family like 'Roboto', sans-serif, Papyrus has been used, along with this different colours like greenyellow,cyan, rgb(15, 82, 186), rgb(5, 5, 81) has been implemented in webpage. A hover effect for the cards also has been added. Images for cards and logo also has been added to CSS. The figure of XML document after CSS is shown below:

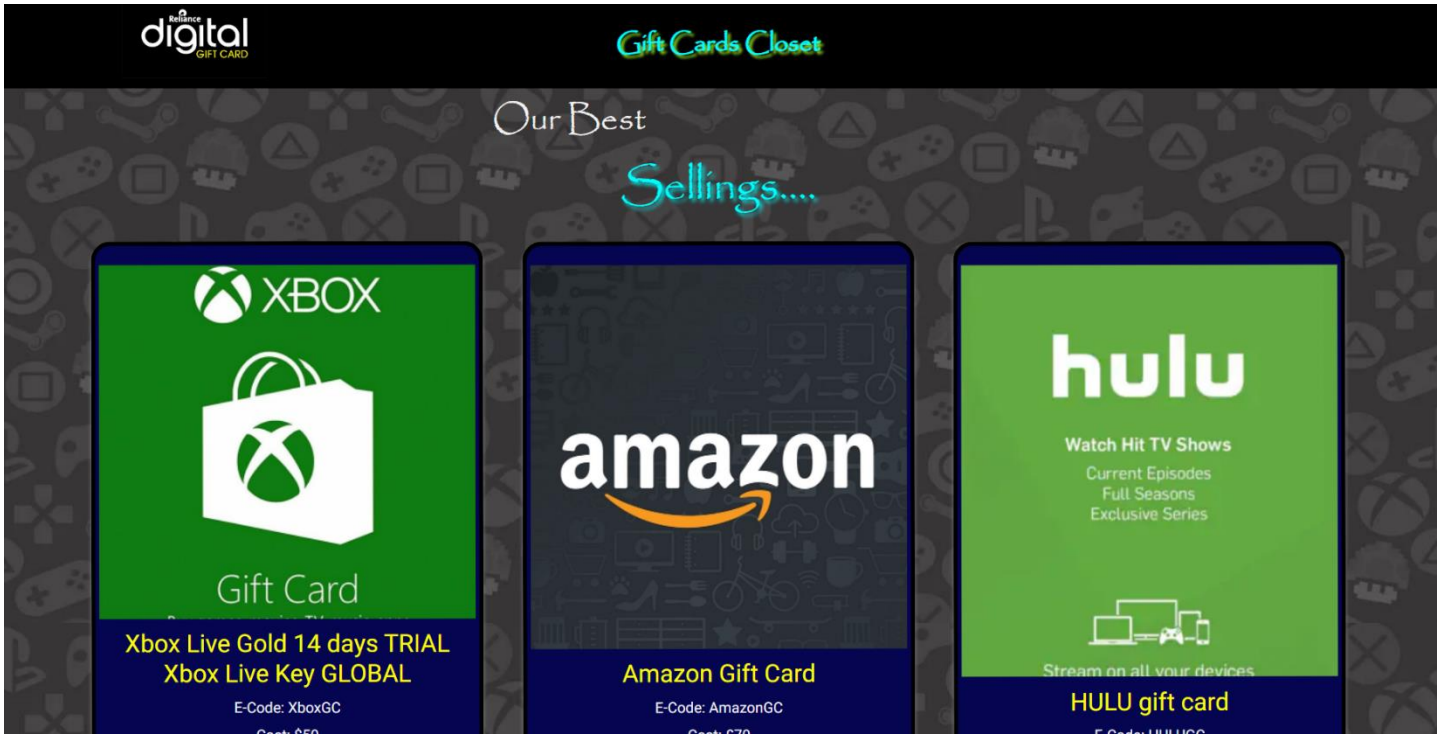

Figure 11: XML Document after CSS

After the completion of development part, a reported was prepared in MS word where all the contents of the development parts are included along with its description. In, this way the coursework was developed successfully within the deadline.

Critical Analysis

Although the coursework has been completed successfully, there were many problems that had raised during the development phase. Many errors were encountered to complete the coursework. Much research was done to solve the errors and also guidance from module teacher had been received to complete this coursework.

Some of the errors that were encountered are:

First error was encountered while validating the XML document.

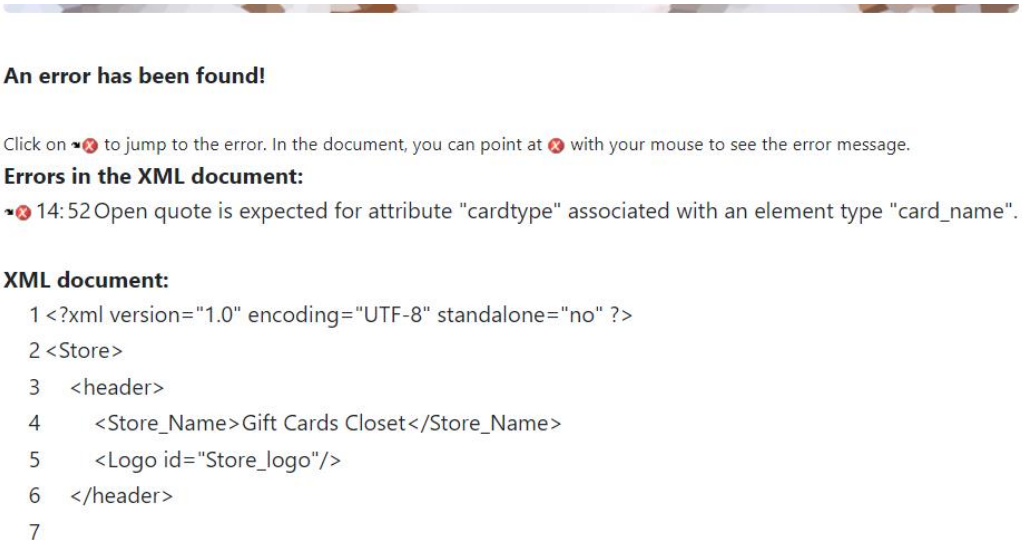

Figure 12: error in XML during validation

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<Store>
    <header>
        <Store_Name>Gift Cards Closet</Store_Name>
        <Logo id="Store_logo"/>
    </header>

    <Title_1>Our Best</Title_1>
    <Title_2>Sellings....</Title_2>

    <GiftCards>
        <card cardID="GC01123">
            <cardPicture id="card_1"/>
            <card_name available_Qnty="5" cardtype=>Xbox Live Gold 14 days TRIAL Xbox Live Key GLOBAL</card_name>
            <e-code>E-Code: XboxGC</e-code>
            <cost currency="USD">Cost: $50</cost>
            <number_of_user>Users: 1K</number_of_user>
            <discount_voucher>Discount: Available</discount_voucher>
            <Validity>Valid Till: 09/14/2023</Validity>
            <Delivery_time>Delivery duration: 3-5 Days</Delivery_time>
        </card>
```

Figure 13: Error in XML

The cause of this error was due to empty input of cardtype attribute in Card_1. After the error message received from the validator which was 14:52Open quote is expected for attribute "cardtype" associated with an element type "card_name", a code was viewed and was debugged. After debugging, the error was solved.

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<Store>
    <header>
        <Store_Name>Gift Cards Closet</Store_Name>
        <Logo id="Store_logo"/>
    </header>

    <Title_1>Our Best</Title_1>
    <Title_2>Sellings....</Title_2>

    <GiftCards>
        <card cardID="GC01123">
            <cardPicture id="card_1"/>
            <card_name available_Qnty="5" cardtype="Digital">Xbox Live Gold 14 days TRIAL Xbox Live Key GLOBAL</card_name>
            <e-code>E-Code: XboxGC</e-code>
            <cost currency="USD">Cost: $50</cost>
            <number_of_user>Users: 1K</number_of_user>
            <discount_voucher>Discount: Available</discount_voucher>
            <Validity>Valid Till: 09/14/2023</Validity>
            <Delivery_time>Delivery duration: 3-5 Days</Delivery_time>
        </card>
```

Figure 14: Correction in XML document

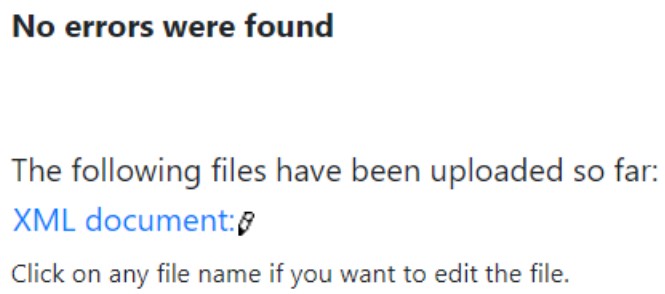

Figure 15: No error message after XML validation

Second errors were encountered during the validation of XML against Schema. There were too many errors which was solved one by one. The below figure shows the errors found during validation:

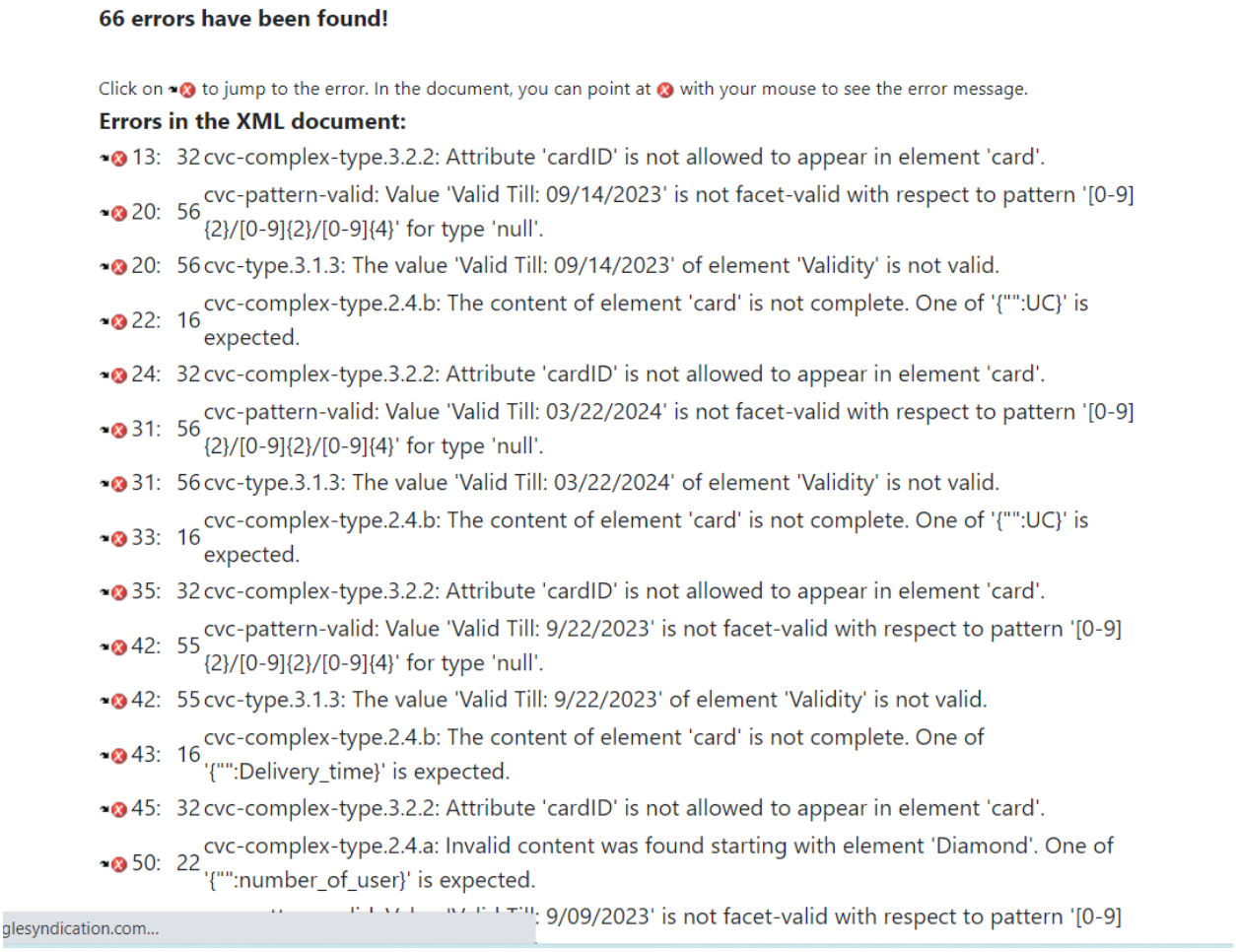

Figure 16: Error during validation of XML against Schema

All of this error has been raised due to invalid datatype of validity, Diamond, UC elements. And other errors like error of cardID was due to undeclared attributes of element card in Schema.

These errors were also solved by declaring the attributes, placing right datatype of elements and so on. The validation of XML against Schema with no error is shown below:

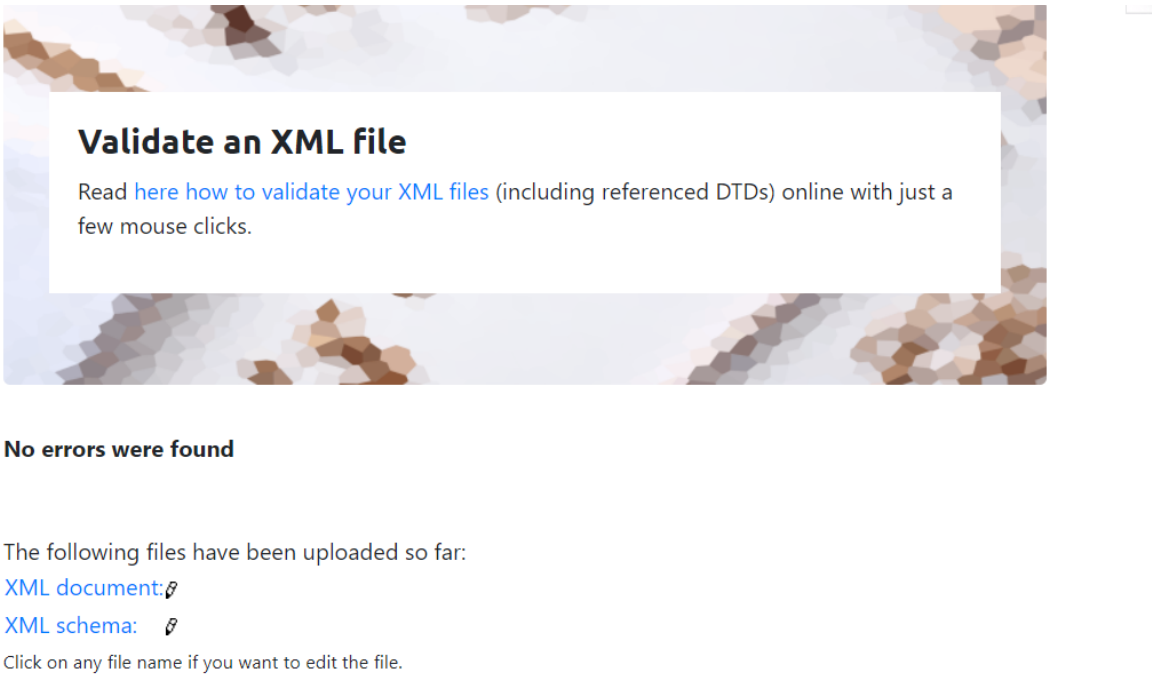

Figure 17: Validation of XML against Schema with no errors

Third error was also encountered during the validation of XML against Schema. This error was due to mistake in spelling of enumeration in schema. This error took place during correcting the format of the schema after the successful validation. And while validating the Schema for second time after correcting the format, this error was encountered.

```
<xs:simpleContent>
    <xs:extension base="xs:string">
        <xs:attribute name="currency">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:enumation value="USD"/>
                    <xs:enumuration value="GBP"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:extension>
```

Figure 18: Error in Schema

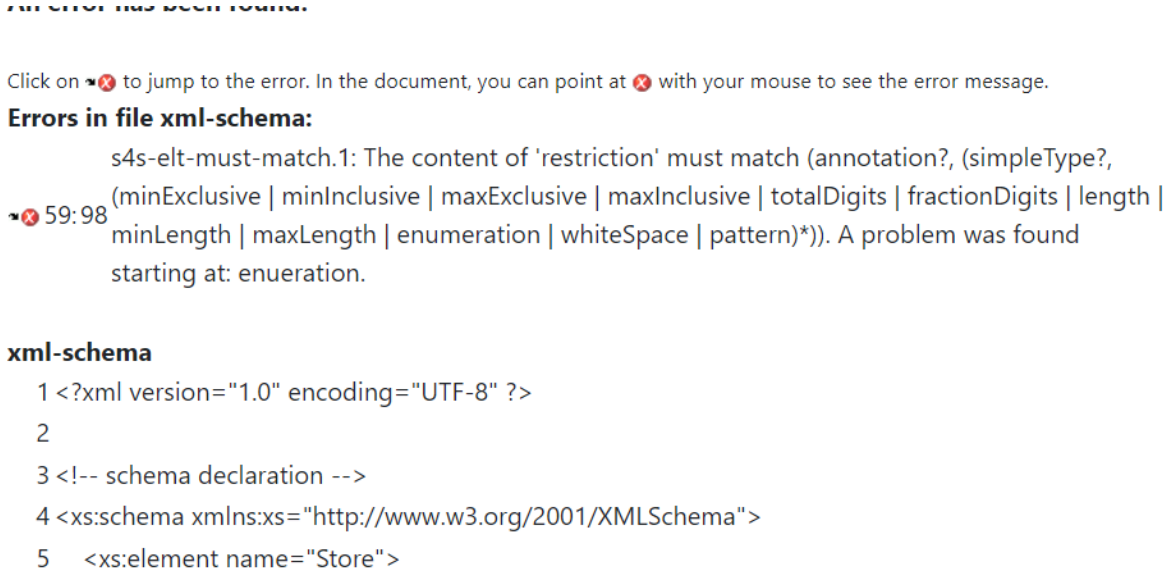

Figure 19: Schema error message during validation

This error was solved by correcting the spelling of enumeration. After the problem was solved the XML document was successfully validate against Schema.

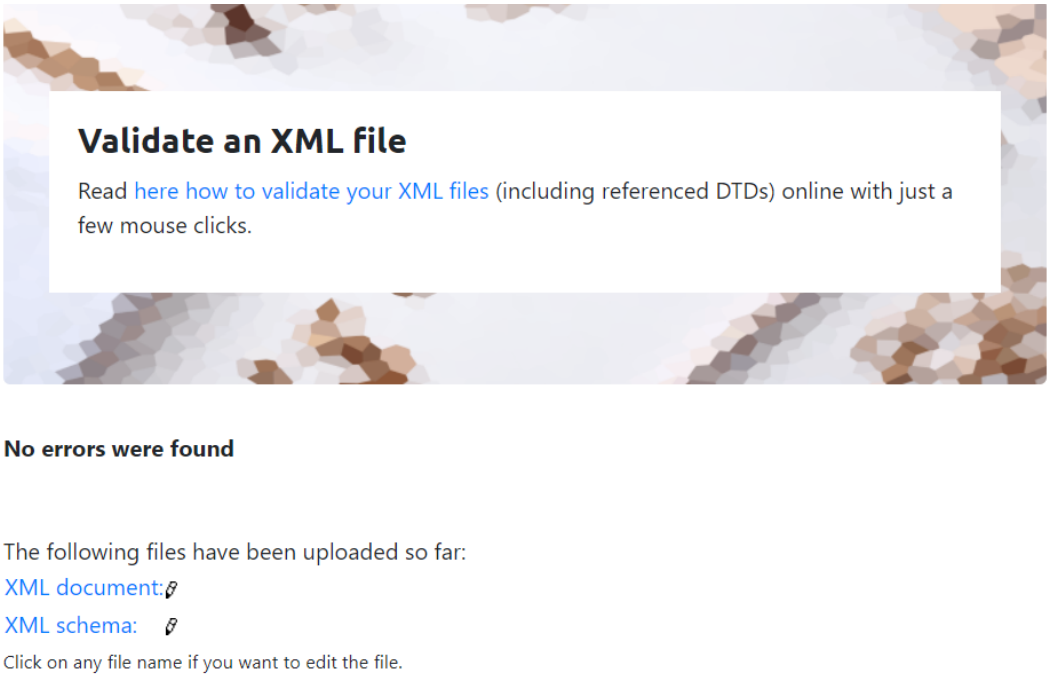

Figure 20: Successful Validation

Conclusion

A model system for a gift card store was developed which meets all the requirement mentioned by the store manager. The details of the store include store name, website, logo, their best selling's, address, telephone number and many other. It's best selling's includes the giftcard of Xbox, PUBG, Freefire, Twitch, itunes and so on. It also contains the properties of the giftcard like validity, card name, users of card, cost, e-code, type of card and others.

The whole development includes XML, Schema, CSS and some pictures for webpage. The XML contains the properties of cards, details of store and cards. While the schema contains the validation of the elements and attributes of XML. CSS is for the styling purpose of the webpage; it has styled the webpage of store in very attractive way.

During the phase od development and documentation different problems were raised and also those problems were solved with the guidance of respective tutor and research from different websites, books and YouTube videos. This coursework provides the knowledge of XML, schema and CSS. A brief knowledge was gained with the help of this coursework. Not only this, this coursework has provide the knowledge about the way of research through different online platform and also through some physical platform like books, journals, and others.

In an overview, it was a great experience, and we could learn so many things that would surely help for future projects and industry-level contributions.

References

DivyanshGupta1, 2021. *geeksforgeeks.* [Online]
Available at: www.geeksforgeeks.org/css-introduction/
[Accessed 4 May 2022].

Jeff, F., 2019. *Java XML and JSON: Document Processing for Java SE.* 2nd ed. s.l.:Apress.

Joan, B., 2011. *Differenve Between.net.* [Online]
Available at: http://www.differencebetween.net/
[Accessed 4 May 2022].

Rudd, A., 2020. *NBCCOMEDYPLAYGROUND.* [Online]
Available at: www.nbccomedyplayground.com/what-is-xml-tree-structure/
[Accessed 4 May 2022].

Thomas, H., 2022. *Guru99.* [Online]
Available at: www.guru99.com/software-testing-introduction-importance.html
[Accessed 4 May 2022].