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: Achyut Adhikari

London Met ID:20048811

College ID: NP01CPS210055

Assignment Due Date: 5th May 2022

Assignment Submission Date:5th May 2022

Title (Where Required):

Word Count (Where Required):2399

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

Introduction 1
XML Content 2
XML Attributes: 3
Tree Diagram 4
XML Code 5
XML Schema 9
Testing 12
Test 1: Check if the XML Document is well formed. 12
Test 2: Validate XML against schema 13
Test 3: Check if the CSS is being rendered in the browser 14
Test 4: Check if the hover is working 15
Test 5: Change of background color 17
Coursework Development Process 18
Critical Analysis 23
References 28

Table of Figures

Figure 1 Tree Diagram..... 4
Figure 2 Test 1 12
Figure 3 Test 2 13
Figure 4 Test 3 14
Figure 5 Test 4i..... 15
Figure 6: Test 4(ii) 16
Figure 7 Test 5i..... 17
Figure 8 Test 5ii..... 17
Figure 9 Adding CSS to xml..... 19
Figure 10 without flex..... 23
Figure 11 code without flex..... 24
Figure 12 with flex 24
Figure 13 code with flex..... 25
Figure 14 Error having same icon..... 25
Figure 15 error solved having same icon 25

Table of Tables

Table 1 Test 1..... 12
Table 2 Test 2..... 13
Table 3 Test 3..... 14
Table 4 Test 4..... 15
Table 5 Test 5..... 17

Introduction

The following report is the second evaluation of the module "Emerging Programming Platforms and Technologies." The main goal of this evaluation is to create a Schema validated and XML document for an online gift shop that contains standardized document information. It goes so over XML development process in incredible detail. The document in XML format gives basic knowledge regarding the company, such as its name (Hrtoyu Gifts), The store's address, logo, contact information, and information about its gift card collections are all included. The pricing gift card are even included in the description. The fundamental objectives of

The following are the results of this work:

- To use a markup language and design tools to create an XML document this document includes all of the store's and gift card's data.
- To acquire knowledge and expertise in producing an XML document, as well as to discover how to do it understanding how and why schema must be used to validate it
- To learn further about schema and how it works.

XML Content

XML (Extensible Markup Language) is used to describe data. The XML standard is a flexible way to create information formats and electronically share structured data via the public internet, as well as via corporate networks.
XML is a markup language based on Standard Generalized Markup Language (SGML) used for defining markup languages.
XML's primary function is to create formats for data that is used to encode information for documentation, database records, transactions, and many other types of data. XML data may be used for creating different content types that are generated by building dissimilar types of content -- including web, print and mobile content -- that are based on the XML data. (Peter Loshin, 2022)

XML Declaration is provided below:

<?xml version = “1.0” encoding = “UTF-8”?>

The declaration in XML is not a tag. Meta-data is transferred through xml.

The XML declaration begins with the character sequence <?xml and ends with the character sequence ?>. Note that although this syntax is identical to that for processing instructions, the XML declaration is not considered to be a processing instruction. All XML declarations have a version attribute with a value that must be 1.0 (w3resource, 2020)

Rules of Xml declarations
If the XML declaration is present in the XML, it must be placed as the first line in the XML document.
If the XML declaration is included, it must contain version number attribute.
The Parameter names and values are case-sensitive.
The names are always in lower case.
The order of placing the parameters is important. The correct order is: *version, encoding and standalone.*
Either single or double quotes may be used.
The XML declaration has no closing tag i.e. **</?xml>** (tutorialspoints, 2022)

XML comprises for the following components:

XML Elements: XML elements can be defined as building blocks of an XML. Elements can behave as containers to hold text, elements, attributes, media objects or all of these. Each XML document contains

one or more elements, the scope of which are either delimited by start and end tags, or for empty elements, by an empty-element tag. (Tutorialspoint, 2022)

XML Attributes:

A element of an XML element is an XML attribute. The introduction of an attribute to an XML element enhances the element's attributes by providing it with more measured by the standard**.**

Tree Diagram

An XML document has a self-descriptive structure. It forms a tree structure which is referred as an XML tree. The tree structure makes easy to describe an XML document. A tree structure contains root element (as parent), child element and so on. It is very easy to traverse all succeeding branches and sub-branches and leaf nodes starting from the root. (JavaTpoint, 2022)

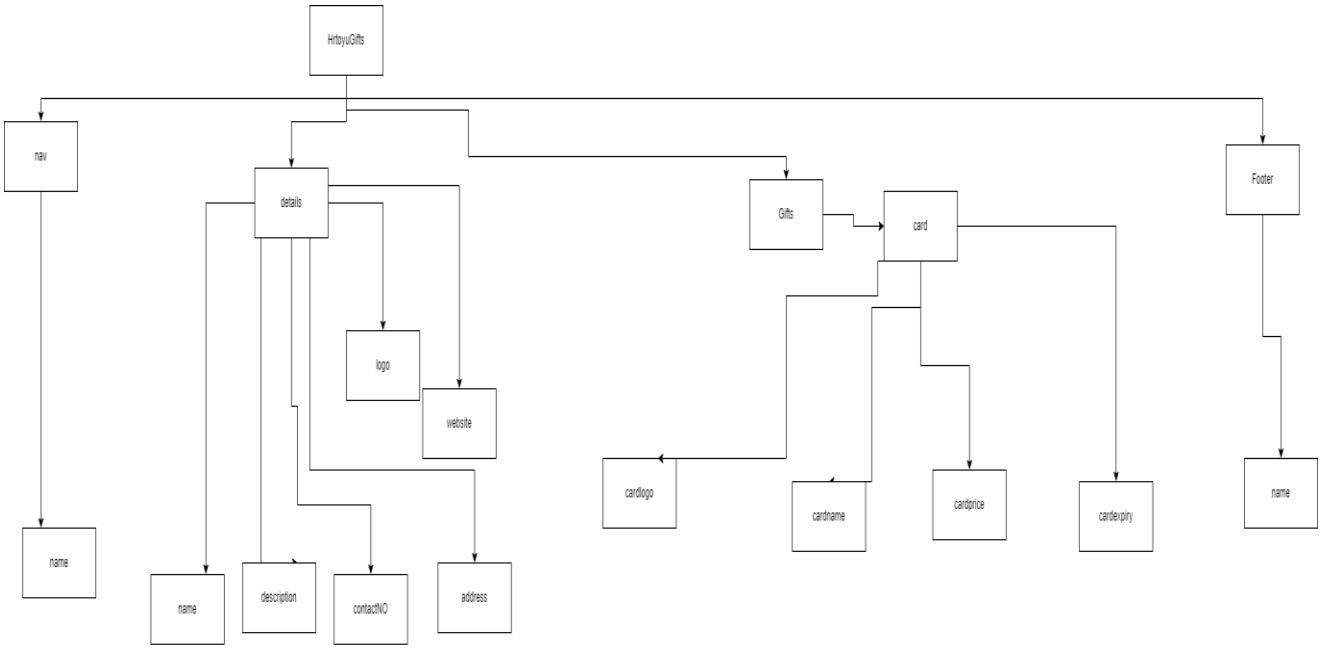

Figure 1 Tree Diagram

XML Code

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="abc.css" type="text/css"?>
<hrtoyu_gifts Storename="HrtoyugiftShop" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="abc.xsd">
  <nav>
    <name>HrtoyuGifts</name>
    <name>Home</name>
    <name>About us</name>
    <name>Contact</name>
  </nav>
    <details>
    <logo/>
    <desc>
    <name>HrtoyuGifts</name>
   <description>Literally meaning Heart to connect, Which sales genuine Gift cards.</description>
    <address>Address: London United Kingdom</address>
    <contactNo>Contact No: +44 2234718002</contactNo>
    <Website>www.hrtoyugifts.de</Website>
  </desc>
</details>

  <gifts>
    <card  cardid="Card01" company="Google" type="Physical" category="discount" NoOfUse="1" Discount="True">
      <cardlogo cardlogoID="01" />
        <CardName>GoogleGiftCard</CardName>
        <Price>10$</Price>
        <Expiry>2025</Expiry>
        <country>Nepal</country>
        <discount>10%</discount>
        <server>local</server>
    </card>
```

```
        <card cardid="Card02" company="Amazon" type="Physical" category="discount" NoOfUse="5" Discount="False">
            <cardlogo cardlogoID="02" />
            <CardName>AmazonGiftCard</CardName>
            <Price>10$</Price>
            <Expiry>2025</Expiry>
            <country>Nepal</country>
            <discount>10%</discount>
            <server>local</server>
        </card>

        <card cardid="Card03" company="Riot" type="Digital" category="Voucher" NoOfUse="3" Discount="False">
            <cardlogo cardlogoID="03" />
            <CardName>RiotGiftCard</CardName>
            <Price>15$</Price>
            <Expiry>2025</Expiry>
            <country>Nepal</country>
            <discount>15%</discount>
            <server>Eu</server>
        </card>

        <card cardid="Card04" company="Steam" type="Digital" category="discount" NoOfUse="4" Discount="True">
            <cardlogo cardlogoID="04" />
            <CardName>SteamGiftCard</CardName>
            <Price>10$</Price>
            <Expiry>2025</Expiry>
            <country>Nepal</country>
            <discount>10%</discount>
            <server>local</server>
        </card>

        <card cardid="Card05" company="Netflix" type="Digital" category="discount" NoOfUse="2" Discount="False">
```

```
        <cardlogo cardlogoID="05" />
            <CardName>NetflixGiftCard</CardName>
            <Price>50$</Price>
            <Expiry>2023</Expiry>
            <country>Nepal</country>
            <discount>10%</discount>
            <server>local</server>
        </card>

          <card  cardid="Card06"  company="tify"  type="Digital"  category="Voucher"  NoOfUse="8"
Discount="True">
        <cardlogo cardlogoID="06" />
            <CardName>tifyGiftCard</CardName>CardName>
            <Price>5$</Price>
            <Expiry>2026</Expiry>
            <country>Nepal</country>
            <discount>10%</discount>
            <server>local</server>
        </card>

       <card  cardid="Card07" company="Discord" type="Digital" category="discount" NoOfUse="50"
Discount="False">
        <cardlogo cardlogoID="07" />
            <CardName>DiscordGiftCard</CardName>
            <Price>15$</Price>
            <Expiry>2028</Expiry>
            <country>Nepal</country>
            <discount>10%</discount>
            <server>local</server>
        </card>

        <card  cardid="Card08" company="Apple" type="Physical" category="discount" NoOfUse="5"
Discount="True">
        <cardlogo cardlogoID="08" />
            <CardName>AppleGiftCard</CardName>
```

```
            <Price>100$</Price>
            <Expiry>2024</Expiry>
            <country>Nepal</country>
            <discount>10%</discount>
            <server>local</server>
        </card>



        <card  cardid="Card09" company="Daraz" type="Physical" category="discount" NoOfUse="4"
Discount="True">
         <cardlogo cardlogoID="09"/>
            <CardName>DarazGiftCard</CardName>
            <Price>100$</Price>
            <Expiry>2025</Expiry>
            <country>Nepal</country>
            <discount>10%</discount>
            <server>local</server>
        </card>
    </gifts>
    <footer>
        <name>Copyright Hryotu © </name>
    </footer>
</hrtoyu_gifts>
```

XML Schema

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
 <xs:element name="hrtoyu_gifts">
  <xs:complexType>
   <xs:sequence>
    <xs:element name="nav">
     <xs:complexType>
      <xs:sequence>
       <xs:element type="xs:string" name="name" maxOccurs="unbounded" minOccurs="0"/>
      </xs:sequence>
     </xs:complexType>
    </xs:element>
    <xs:element name="details">
     <xs:complexType>
      <xs:sequence>
       <xs:element type="xs:string" name="logo"/>
       <xs:element name="desc">
        <xs:complexType>
         <xs:sequence>
          <xs:element type="xs:string" name="name"/>
          <xs:element type="xs:string" name="description"/>
          <xs:element type="xs:string" name="address"/>
          <xs:element type="xs:string" name="contactNo"/>
          <xs:element type="xs:anyURI" name="Website"/>
         </xs:sequence>
        </xs:complexType>
       </xs:element>
      </xs:sequence>
     </xs:complexType>
    </xs:element>
    <xs:element name="gifts">
     <xs:complexType>
      <xs:sequence>
       <xs:element name="card" maxOccurs="unbounded" minOccurs="0">
```

```
            <xs:complexType mixed="true">
              <xs:sequence>
                <xs:element name="cardlogo">
                  <xs:complexType>
                    <xs:simpleContent>
                      <xs:extension base="xs:string">
                        <xs:attribute type="xs:byte" name="cardlogoID" use="optional"/>
                      </xs:extension>
                    </xs:simpleContent>
                  </xs:complexType>
                </xs:element>
                <xs:element type="xs:string" name="CardName"/>
                <xs:element type="xs:string" name="Price"/>
                <xs:element type="xs:short" name="Expiry"/>
                <xs:element type="xs:string" name="country"/>
                <xs:element type="xs:string" name="discount"/>
                <xs:element type="xs:string" name="server"/>
              </xs:sequence>
              <xs:attribute type="xs:string" name="cardid" use="optional"/>
              <xs:attribute type="xs:string" name="company" use="optional"/>
              <xs:attribute type="xs:string" name="type" use="optional"/>
              <xs:attribute type="xs:string" name="category" use="optional"/>
              <xs:attribute type="xs:byte" name="NoOfUse" use="optional"/>
              <xs:attribute type="xs:string" name="Discount" use="optional"/>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="footer">
      <xs:complexType>
        <xs:sequence>
          <xs:element type="xs:string" name="name"/>
        </xs:sequence>
      </xs:complexType>
```

```
      </xs:element>
    </xs:sequence>
    <xs:attribute type="xs:string" name="Storename"/>
   </xs:complexType>
  </xs:element>
</xs:schema>
```

Testing

Test 1: Check if the XML Document is well formed.

Test No	1
Action Performed	Open XML file in browser without linking CSS to it
Expected Result	The structure of the XML document should be displayed
Actual Result	The structure of the XML document was displayed
Test Result	The test was successful

Table 1 Test 1

Figure 2 Test 1

Test 2: Validate XML against schema

Test No	2
Action Performed	XML is validated against the schema in an online tool
Expected Result	No errors should be found
Actual Result	No errors were found
Test Result	The test was successful

Table 2 Test 2

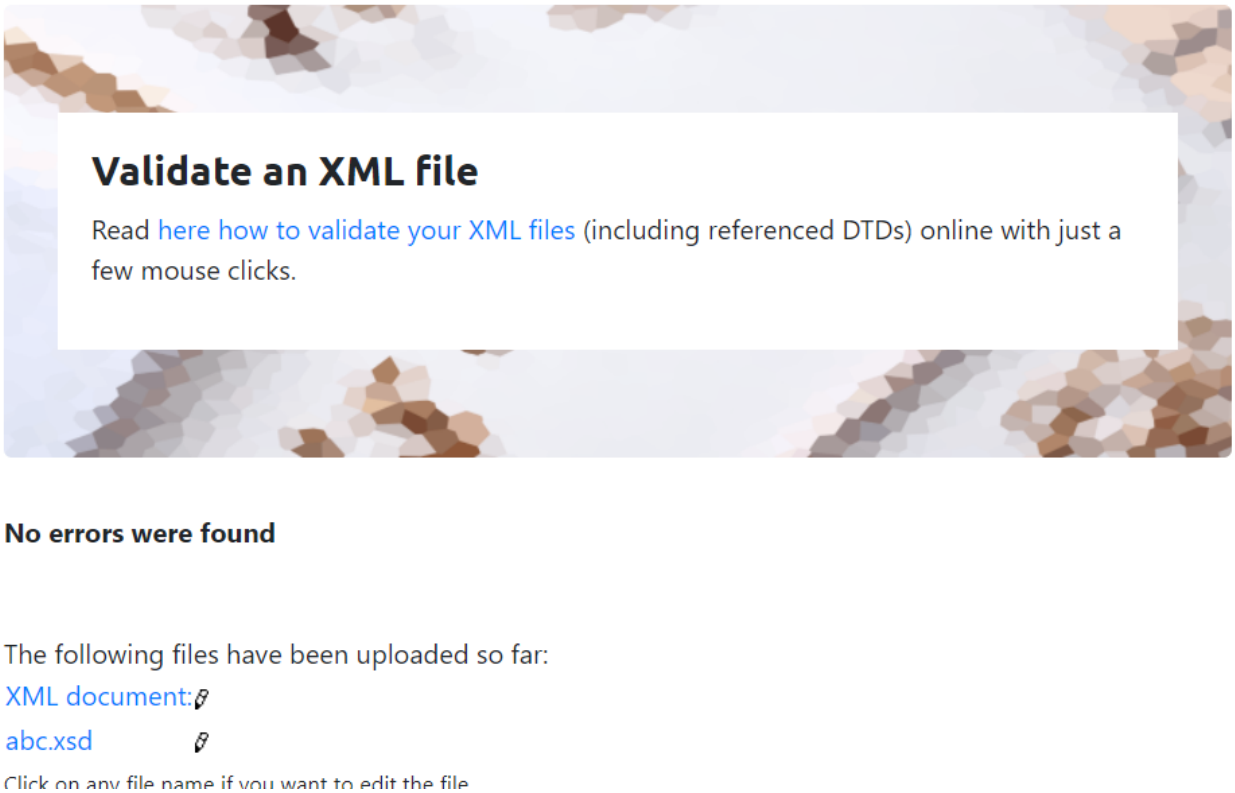

Figure 3 Test 2

Test 3: Check if the CSS is being rendered in the browser

Test No	3
Action Performed	XML is opened in a browser linking the CSS
Expected Result	The document should be rendered as expected
Actual Result	The document was rendered
Test Result	The test was successful

Table 3 Test 3

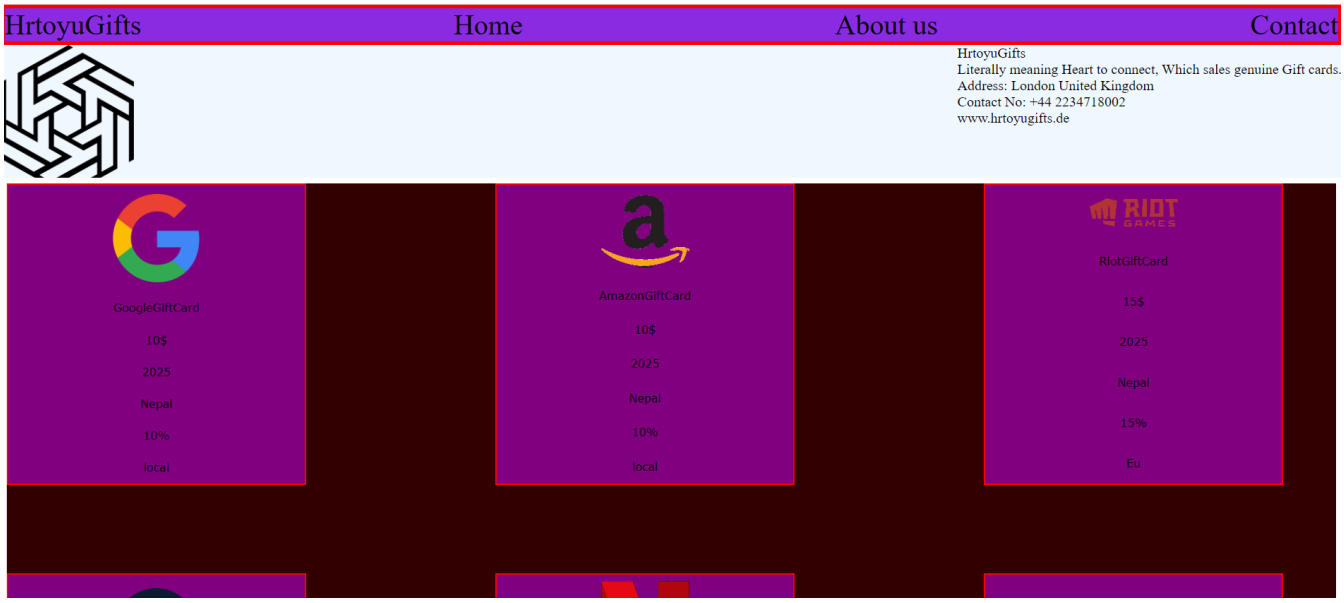

Figure 4 Test 3

Test 4: Check if the hover is working

Test No	4
Action Performed	The pointer is hovered above the card to
Expected Result	The card should change its color
Actual Result	The card changed its color
Test Result	The test was successful

Table 4 Test 4

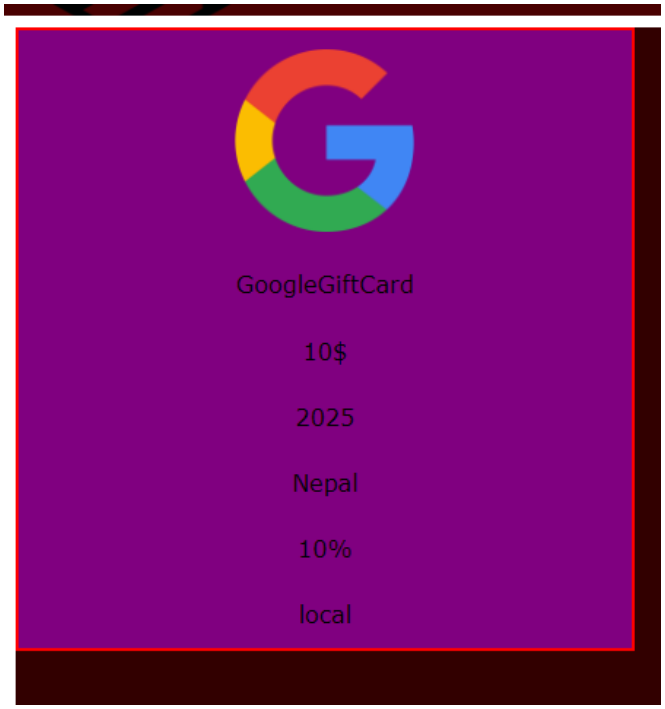

Figure 5 Test 4i

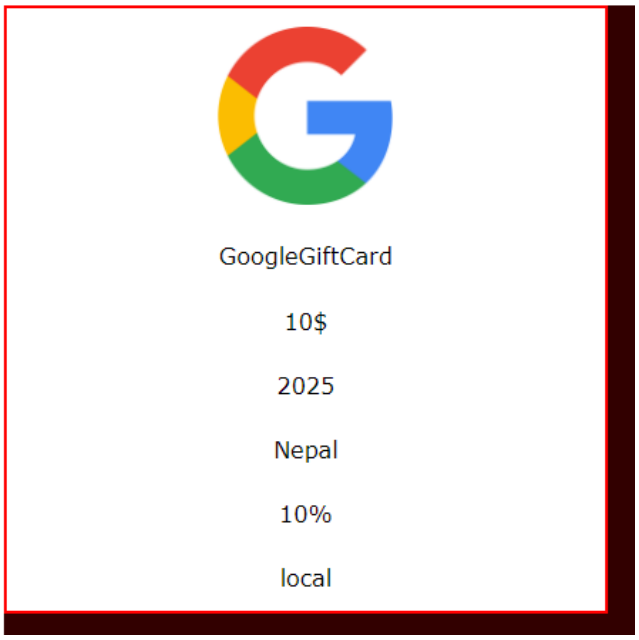

Figure 6: Test 4(ii)

Test 5: Change of background color

Test No	5
Action Performed	XML is opened in a browser linking the CSS
Expected Result	The color of the navigation should be changed
Actual Result	The color of the navigationchanged
Test Result	The test was successful

Table 5 Test 5

Figure 7 Test 5i

Figure 8 Test 5ii

Coursework Development Process

There were several criteria and conditions to meet in order to complete this job. For this project, we needed to develop a Schema-validated XML document that contained standardized document information for an online gift store.Before starting, we did a lot of research and study on XML and Schema the programming process, using the resources provided by the teachers and the internet To construct The an in-depth and extensive study of the XML document, the proper structure for the XML document requirements was carried out. We must first define what an XML document is.We'll need to know what components and characteristics we'll use. A visual representation of the structure It was necessary to create a tree of the XML document, which could be done by usingdiagram. As may be seen in this branch of the tree in the above figure.

As we known XML Schema Definition is yet another name for XML Schema (XSD). It's used to define and validate XML data's structure and content. The components, properties, and data types are defined by the XML standard. Namespaces are supported by the Schema element. It's comparable to a database schema, which explains how the data in a database is organized.so we created a xml document using the structure as per as the visualization in the diagram.likewise external file was created and used to validate the xml content . then the validation is done using online tools known as XML validation tool.

Then the Css was added to the xml where the style was used. CSS helps to describes how the components in the markup language should appear on the screen. Changing the component look attractive .

```
    color: black;
    height: 338px;
    width: 335px;
    border: 2px solid red;
    margin-right: 60px;
    margin-bottom: 100px;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: space-around;
}

cardlogo[cardlogoID="01"] {
    background-image: url("css/Google__G__Logo.svg.png");
    background-size: 100px;
    height: 100px;
    width: 100px;
}

cardlogo[cardlogoID="02"] {
    background-image: url("css/amazon_PNG5.png");
    background-size: 97px;
    height: 82px;
    width: 100px;
}
```

Figure 9 Adding CSS to xml

Following the verification of all codes, testing was conducted to confirm that all criteria were met, and the project was documented in compliance with the standards, including all research and conclusions. To construct this project, a variety of tools were used to achieve diverse goals. To satisfy all of the needs and criteria, the following tools were used.

• Diagrams.net: diagrams.net/draw.io is an open source technology stack for building diagramming applications, and the world's most widely used browser-based end-user diagramming software.
diagrams.net is a trademark and draw.io is a registered trademark of JGraph Ltd. JGraph Ltd is a company registered in England that develops and owns the software, runs the diagrams.net and draw.io sites and owns the diagrams.net and draw.io brands.
We promise you we won't hide your data from you and will always give you some way to open and edit that data, at no cost. When companies pay us money it should be because we add value, not because they are locked in.
We are disrupting the industry with a new business model that makes reasonable profits to pay a dedicated, professional software team, but doesn't use artificial scarcity to produce a bloated sales-centric company with matching revenues (Diagrams.net, 2022)

• Visual Studio Code: Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity). Begin your journey with VS Code with these introductory videos. (Visual Studio code, 2022)

• Microsoft Word: Microsoft word is a word processor software developed by Microsoft in 1983. It is the most used word processor software. It is used to create professional quality documents, letters, reports, resumes, etc and allows you to edit or modify your new or existing document.

Difference between schema and DTD

The main difference between DTDs and XML Schemas is that XML Schemas use an XML-based syntax, whereas DTDs use a special syntax that dates back to SGML DTDs. The converse is true for XML Schema, which is verbose but also uses tags and XML, making the syntax of XML Schema less scary to XML authors. Although DTDs are frequently chastised for requiring users to learn new grammar, the syntax is extremely simple. XML is used to describe and validate XML data's structure and content. The components, properties, and data types are defined by an XML schema. A set of objects are supported by the schema element. It is like a database schema, which explains how the data in a database is organized.

A document type definition is the full form of DTD. A document type definition (DTD) is a set of markup declarations for an SGML-family markup language that defines a document type (SGML, XML, HTML). The valid building pieces of an XML document are defined by a DTD. It contains a list of validated components and attributes that describe the document structure. (Askany, 2022)

The key differences between Schema and DTD are as follows:

Parameters of Comparison	XML Schema	XML DTD
Definition	XML Schema helps in the generation of XML documents and defines rules of attributes and elements of an XML document.	DTD describes the attributes and structure of an XML document and languages.
Namespace	XML Schema supports namespace.	XML DTD doesn't support namespace.
Structure Control	It gives better structure control of any XML document.	It gives less structure control of any XML document compared to Schema.
Extensible	XML Schema is extensible	XML DTD is not extensible
Data Types	The schema supports Data Types.	XML DTD does not support Data Types.

To summarize, the following are the key differences between XML Schema and DTD:

1. DTD is generally used to specify the structure of an XML file, whereas Schema is mostly used to describe the structure and contents of an XML file.

2. DTD does not support namespace, although Schema does.

3. Datatypes are not supported by DTD, but they are for elements and attributes in Schema.

4. A DTD offers you less power over XML than a Schema provides.

5. DTD is taken from SGML (Standard Generalized Markup Language) syntax, whereas schema is defined in XML.

6. The DTD is not extendable; however, the Schema is.

7. In XML Schema, there are various derived and built-in data types that are not included in DTD.

8. DTD is not strongly typed, whereas XML Schema is.

9. Insert entries are not permitted in XML Schema but are permitted in DTDs.

Critical Analysis

Several problems developed throughout the development of this curriculum, which were addressed after extensive study and research. Since this was my first time working on an XML project, I anticipated I'd hit some obstacles. Most of the issues are caused by discovered while validating the XML and applying CSS to the XML document full. The problems that were faced, as well as how they were solved, are detailed in this narrative. This critical analysis of the curriculum is given with sufficient evidence development. The first mistake I made was when referencing an external XML document in the XML document. Schema. I got an error when I tried to validate the XML document against the schema the mistake that follows.

The representation of flexbox on the screen while rendering CSS in the browser was one of the most difficult challenges for me to fix. For a more aesthetically pleasing presentation in the browser, I used the flex display property to display the gift cards element horizontally in the screen, as shown.

Figure 10 without flex

```css
card {
    background-color: purple;
    color: black;
    height: 338px;
    width: 335px;
    border: 2px solid red;
    margin-right: 60px;
    margin-bottom: 100px;
    display: block;
    flex-direction: column;
    align-items: center;
    justify-content: space-around;
}
```

Figure 11 code without flex

Figure 12 with flex

```
card {
    background-color: purple;
    color: black;
    height: 338px;
    width: 335px;
    border: 2px solid red;
    margin-right: 60px;
    margin-bottom: 100px;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: space-around;
}
```

Figure 13 code with flex

Similary,also one of the challenging issues was getting same icon in every card. this issue was resolved by targeting each elements using their attributes

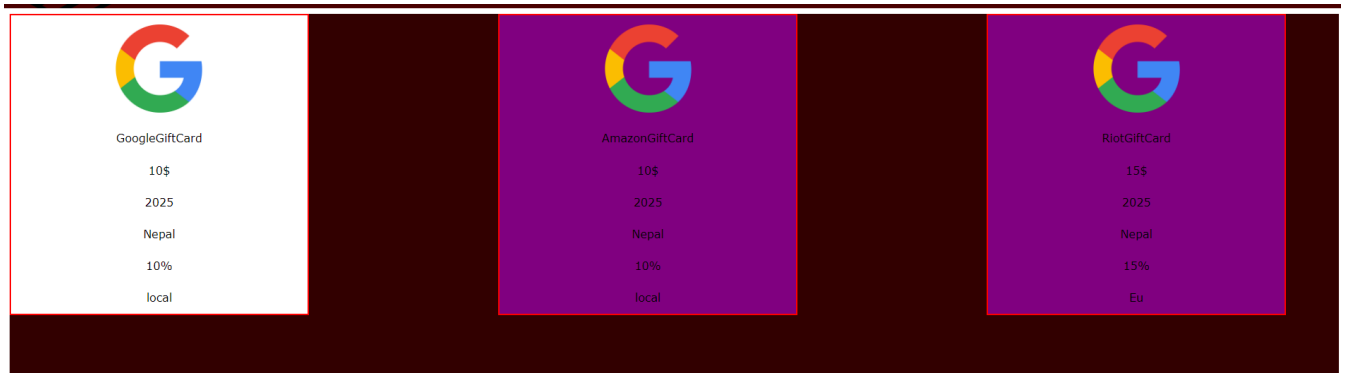

Figure 14 Error having same icon

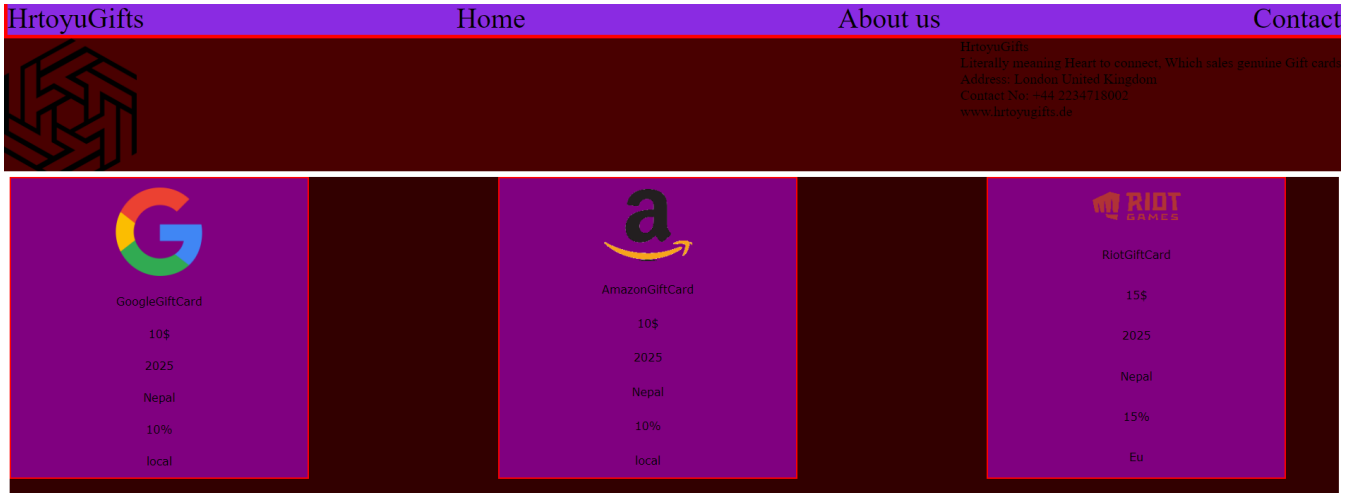

Figure 15 error solved having same icon

In this way, all problems were solved along the road with the help of the web and study materials provided by our professors, allowing us to enhance our problem-solving skills and acquire experience dealing with similar scenarios. This training enabled us to develop my knowledge of XML, CSS, and schemas. After analyzing the circumstances and consulting references, the development project was simple and clear to comprehend and complete. Furthermore, in a short amount of time, efficiently creating a web page integrating skills and information obtained in the classroom and on the internet has shown to be quite beneficial in the long term.

Conclusion

In a word, we can use XML to store the data and transfer to other systems that require it. We use XML to create web forms, and here's an example of how it's done in practice. Since XML eliminates all coding and leaves only the basic data, it works. This makes it simple to transfer data from the forms with other systems and apps. We can use Schema to check and improve XML because it isn't pre-formatted. You will have gained a lot of knowledge and expertise in constructing XML documents, as well as a grasp of how and why to validate them using Schema after finishing this project. This development process has provided a plethora of useful knowledge, which has been put to good use.

References

Askany, 2022. *Difference Between XML Schema And DTD (With Table).* [Online] Available at: https://askanydifference.com/ [Accessed 05 04 2022].

Diagrams.net, 2022. *About us.* [Online] Available at: https://www.diagrams.net/about [Accessed 05 04 2022].

JavaTpoint, 2022. *XML Tree Structure.* [Online] Available at: https://www.javatpoint.com/ [Accessed 05 04 2022].

Peter Loshin, D. L. ,. M. G., 2022. *XML.* [Online] Available at: https://www.techtarget.com/ [Accessed 05 04 2022].

Tutorialspoint, 2022. *Xml Elements.* [Online] Available at: https://www.tutorialspoint.com/ [Accessed 05 04 2022].

tutorialspoints, 2022. *XML declaration.* [Online] Available at: https://www.tutorialspoint.com/ [Accessed 05 04 2022].

Visual Studio code, 2022. *Getting Started.* [Online] Available at: https://code.visualstudio.com/ [Accessed 05 05 2022].

w3resource, 2020. *XML declarations.* [Online] Available at: https://www.w3resource.com/ [Accessed 05 04 2022].