LONDON METROPOLITAN UNIVERSITY

Module Code & Module Title

CS5004NA Emerging Programming Platforms and Technologies

Assessment Weightage & Type

50% Individual Coursework

Year and Semester

2019-20 Autumn / 2019-20 Spring

Student Name: Girija Tamang

London Met ID: 18030995

College ID: NP05CP4S190007

Assignment Due Date: 3 June 2020

Assignment Submission Date: 2 June 2020

Submitted To: Pradhumna Dhungana

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

Tables of Tables

Tables of Figures

1. Introduction

This course work is based on XML development. In this course work, there was a scenario where we as an XML developer must develop a framework for the car store to store details of a used car. The model of the system for an online vehicle shop was developed using XML, Schema, and CSS. Firstly, XML is a markup language which describes a set of rules for encoding documents and provides the hardware and software independent way of storing, transporting, and sharing data in both human-readable and machine-readable formats (Data, 1998). An XML schema is a language which is used for describing constraints and validating the structure and content of an XML document (Tutorials Point, 2006). CSS stands for Cascading Style Sheets which is used for describing the presentation of a document in a webpage with different styles, design, layout, and display variations (Data, 1998).

While developing this model system, the system must include the name of the company, its address, telephone number, web address (URL) and logo which should be the image. As it is a vehicle shop, it contains different types of car for sale. Each car has a brand name, car number, description, quantity, and price. A car has information about the manufacturer, car type, color, dimension, etc.

So, as the above-mentioned scenario we will develop an XML system for storing car data. This report includes all the details that were done during the development of this XML document. The entire scenario was first researched, and all the requirements that we needed were included in the document. In this report, you can see the tree diagram, which shows the descriptive structure that describes XML documents. An XML Schema is created for data restriction and validation as per the required scenario. After that CSS was created for displaying car data in an attractive and informative way using the requirements given in the question.

2. XML Content

2.1 Tree diagram

An XML tree diagram is a descriptive structure that describes an XML document with root elements, child elements, and so on.

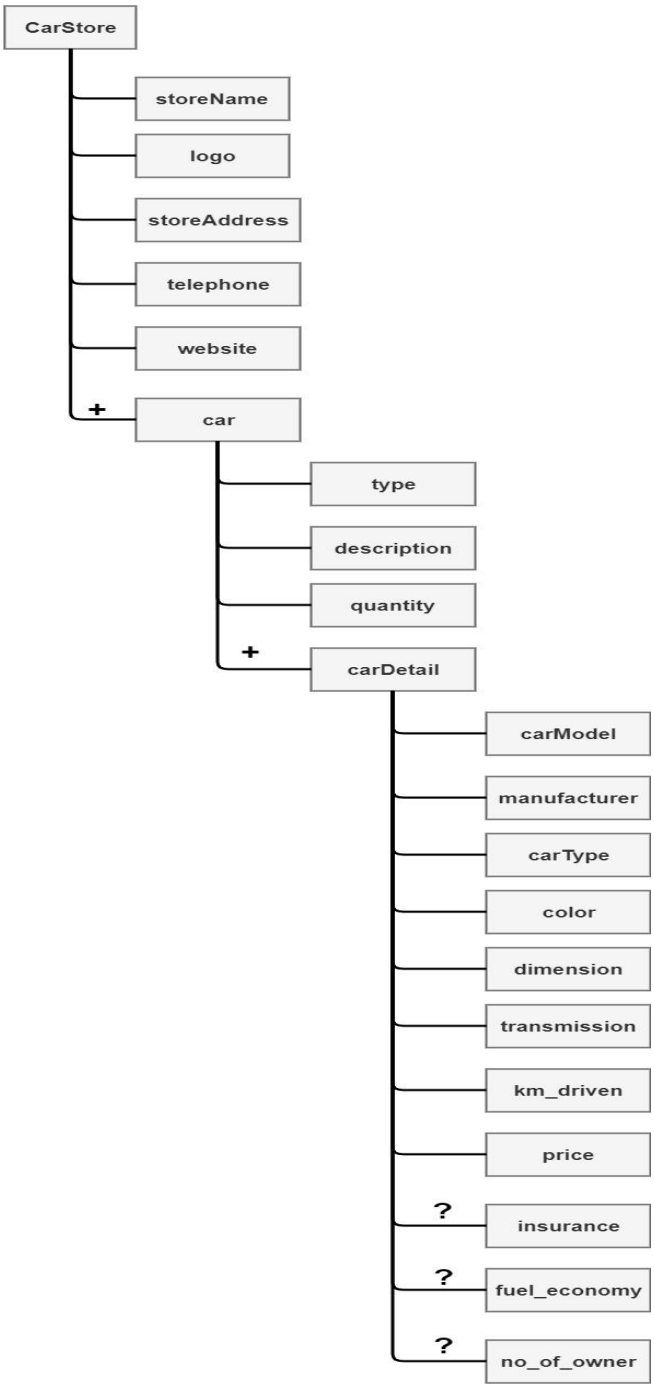


Figure 1: XML tree structure.

2.2 XML

The code of the XML document is pasted below:

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

<?xml-stylesheet type="text/css" href="catalog_18030995.css"?>

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

    <storeName>Heaven Car Shop</storeName>

    <logo> </logo>

    <storeAddress>Dharan-15</storeAddress>

    <telephone>9876543210</telephone>

    <website> www.carstore.com </website>

   <car>

      <type id="ty01">SUV</type>

      <description> Conventionally, a sports utility vehicle (SUV) is a big car built on a body-on-frame chassis, sports increased ground clearance and offers off-roading capabilities to a certain extent. With that said, the aforementioned definition of an SUV has blurred over time and often includes both off-roaders and soft-roaders. </description>

      <quantity>4</quantity>

      <carDetail carid="S01">

        <carModel>Mahindra Scorpio</carModel>

        <manufacturer>Mahindra &amp;amp;Mahindra</manufacturer>

        <carType>Fuel</carType>
```

```
        <color>Black</color>
        <dimension> 4456/1820/1995 mm </dimension>
        <transmission>Manual</transmission>
        <km_driven>54635 </km_driven>
        <price> 1600000 </price>
        <insurance>Yes</insurance>
        <fuel_economy> 15.4  </fuel_economy>
    </carDetail>
    <carDetail carid="S02">
        <carModel>Mahindra Thar</carModel>
        <manufacturer> Mahindra &amp;amp; Mahindra</manufacturer>
        <carType>Fuel</carType>
        <color>Red and Black</color>
        <dimension> 4456/1820/1995 mm </dimension>
        <transmission>Manual</transmission>
        <km_driven> 144635 </km_driven>
        <price> 2000000 </price>
     </carDetail>
    <carDetail carid="S03">
        <carModel>Honda CR-V</carModel>
        <manufacturer>Honda Motor Company</manufacturer>
```

```
            <carType>Hybrid</carType>
            <color>Silver</color>
            <dimension> 4456/1820/1995 mm </dimension>
            <transmission> Automatic </transmission>
            <km_driven>27635 </km_driven>
            <price> 3000000 </price>
            <fuel_economy> 10.4  </fuel_economy>
        </carDetail>
        <carDetail carid="S04">
            <carModel>Renault Duster</carModel>
            <manufacturer> Renault, Dacia </manufacturer>
            <carType>Fuel</carType>
            <color>Blue</color>
            <dimension> 4360/1822/1695 mm </dimension>
            <transmission>Manual</transmission>
            <km_driven>74635 </km_driven>
            <price> 1600000  </price>
            <insurance>Yes</insurance>
        </carDetail>
    </car>
    <car>
```

<type id="ty02"> HATCHBACK </type>

<description> A hatchback is a type of car that has a rear door opening upwards. They usually have a four-door configuration, with the exception of the rear door. Two-door hatchbacks aren't rare though. </description>

<quantity>4</quantity>

<carDetail carid="h01">

<carModel>MINI 3 door Cooper S</carModel>

<manufacturer> Mini (BMW) </manufacturer>

<carType>Fuel</carType>

<color>Black and Yellow </color>

<dimension> 4456/1820/1995 mm </dimension>

<transmission>Automatic</transmission>

<km_driven>195995 </km_driven>

<price> 2200000 </price>

<insurance>Yes</insurance>

<no_of_owner>Second</no_of_owner>

</carDetail>

<carDetail carid="h20">

<carModel>Renault Kwid</carModel>

<manufacturer> Renault </manufacturer>

<carType>Hybrid</carType>

<color>Blue </color>

```
        <dimension> 4456/1820/1995 mm </dimension>
        <transmission>Manual</transmission>
        <km_driven>69995 </km_driven>
        <price>  1300000 </price>
        <no_of_owner>Second</no_of_owner>
    </carDetail>
    <carDetail carid="h21">
        <carModel>Maruti Suzuki Alto 800</carModel>
        <manufacturer>Maruti Suzuki </manufacturer>
        <carType>Fuel</carType>
        <color>Black and Yellow </color>
        <dimension> 3445/1515/1475 mm</dimension>
        <transmission>Manual</transmission>
        <km_driven> 951995 </km_driven>
        <price>  2000000 </price>
        <insurance>Yes</insurance>
        <no_of_owner>Second</no_of_owner>
    </carDetail>
    <carDetail carid="h11">
        <carModel>Hyundai Grand i10 </carModel>
        <manufacturer>Hyundai Motor India Ltd.</manufacturer>
```

```
            <carType>Fuel</carType>
            <color>Red and Black </color>
            <dimension> 3765/1660/1520 mm </dimension>
            <transmission>Manual</transmission>
            <km_driven>295995 </km_driven>
            <price> 3000000 </price>
            <insurance>Yes</insurance>
            <no_of_owner>Second</no_of_owner>
        </carDetail>
    </car>
    <car>
        <type id="ty03">COUPE </type>
        <description> Classically, a coupe is known as a two door closed car with a fixed roof. Considered by definition sporty, it typically only gets 2 seats, or a smaller than normal rear. The typical definition of a coupe has evolved over time and differs across manufacturers, and is now also ascribed to four-door cars and coupe-like proportions, primarily for their sporty appeal rather than number of doors. </description>
        <quantity>2</quantity>
        <carDetail carid="C23">
            <carModel>Ford Mustang</carModel>
            <manufacturer> Ford Motor Company</manufacturer>
            <carType>Fuel</carType>
```

```
            <color>Black and Yellow </color>
            <dimension> 4784/2080/1391 mm </dimension>
            <transmission>Automatic</transmission>
            <km_driven>95995 </km_driven>
            <price>  3500000  </price>
            <insurance>Yes</insurance>
            <fuel_economy>      11.8  </fuel_economy>
        </carDetail>
        <carDetail carid="C2">
            <carModel>Audi R8</carModel>
            <manufacturer> Audi </manufacturer>
            <carType>Hybrid</carType>
            <color>Black and Red </color>
            <dimension> 4426/2037/1240 mm</dimension>
            <transmission>Automatic</transmission>
            <km_driven>969951 </km_driven>
            <price> 2200000 </price>
            <fuel_economy>      9.8  </fuel_economy>
        </carDetail>
    </car>
    <car>
```

<type id="ty04"> SEDAN </type>

<description> A sedan (US) or a saloon (UK) of the various types of cars is historically known as a four-door car with a standard boot / trunk. A somewhat technical aspect is that it typically uses a 3-box configuration for each of the categorically used boxes for engine, passenger, and cargo. Sedans have many subtypes such as notchback, fastback, compact, and sub-compact. </description>

<quantity>2 </quantity>

<carDetail carid="S2">

<carModel>Hyundai Elantra</carModel>

<manufacturer>Hyundai Motor Company</manufacturer>

<carType>Fuel</carType>

<color>Black and Red </color>

<dimension> 4620/1800/1465 mm </dimension>

<transmission>Automatic</transmission>

<km_driven>95995 </km_driven>

<price> 2300000 </price>

<insurance>Yes</insurance>

</carDetail>

<carDetail carid="S12">

<carModel>Maruti Suzuki Ciaz</carModel>

<manufacturer> Suzuki </manufacturer>

<carType>Fuel</carType>

```
            <color>Black </color>
            <dimension>4490/1730/1485 mm</dimension>
            <transmission>Automatic</transmission>
            <km_driven>999959 </km_driven>
            <price> 1900000 </price>
        </carDetail>
    </car>
    <car>
        <type id="ty02">CROSSOVER</type>
        <description> A basic, and popular, concept of a crossover would be that it is a vehicle combining the characteristics of an SUV with a hatchback. Crossover vehicles provide soft-roading capabilities and are designed like a car-instead of the body-on-frame structure used in SUVs, they use unibody architecture. </description>
        <quantity>3</quantity>
        <carDetail carid="CR2">
            <carModel>Hyundai i20</carModel>
            <manufacturer>Hyundai Motor Company</manufacturer>
            <carType>Hybrid</carType>
            <color>Silver</color>
            <dimension>3995/1760/1555 mm</dimension>
            <transmission>Manual</transmission>
```

```
        <km_driven>295995 </km_driven>
        <price>  1000000  </price>
        <no_of_owner>third</no_of_owner>
    </carDetail>
    <carDetail carid="CR7">
        <carModel>Volvo S60 Cross Country</carModel>
        <manufacturer>Volvo Cars </manufacturer>
        <carType>Fuel</carType>
        <color>Black and Red </color>
        <dimension>4637/1866/1539 mm</dimension>
        <transmission>Automatic</transmission>
        <km_driven>495995 </km_driven>
        <price> 3600000  </price>
     </carDetail>
    <carDetail carid="C30">
        <carModel>Maruti Suzuki S-Cross</carModel>
        <manufacturer>NEXA</manufacturer>
        <carType>Fuel</carType>
        <color>Black and Blue</color>
        <dimension> 4300/1785/1595 mm</dimension>
        <transmission>Manual</transmission>
```

```
            <km_driven>9598959 </km_driven>
            <price> 1500000  </price>
            <fuel_economy>      10  </fuel_economy>
            <no_of_owner>second</no_of_owner>
        </carDetail>
    </car>
</CarStore>
```

3. Schema Content

The code of the XML Schema is pasted below:

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

<xs:schema attributeFormDefault = " unqualified " elementFormDefault = " qualified "
xmlns:xs = " http://www.w3.org/2001/XMLSchema ">

 <xs:element name="CarStore">

  <xs:complexType>

   <xs:sequence>

    <xs:element type="xs:string" name="storeName"/>

    <xs:element type="xs:string" name="logo"/>

    <xs:element type="xs:string" name="storeAddress"/>

    <xs:element name="telephone" maxOccurs="1">

     <xs:simpleType>

      <xs:restriction base="xs:decimal">

       <xs:pattern value="98[0-9]{8}"/>

      </xs:restriction>

     </xs:simpleType>

    </xs:element>

    <xs:element type="xs:string" name="website"/>

    <xs:element name="car" maxOccurs="unbounded" minOccurs="1">

     <xs:complexType>

      <xs:sequence>
```

```
<xs:element name="type">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute type="xs:string" name="id" use="required"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element type="xs:string" name="description"/>
<xs:element type="xs:positiveInteger" name="quantity"/>
<xs:element name="carDetail" maxOccurs="unbounded" minOccurs="1">
  <xs:complexType>
    <xs:sequence maxOccurs="unbounded">
      <xs:element type="xs:string" name="carModel"/>
      <xs:element type="xs:string" name="manufacturer"/>
      <xs:element name="carType">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="Fuel"/>
            <xs:enumeration value="Hybrid"/>
```

```
            <xs:enumeration value="Electric"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element type="xs:string" name="color"/>
      <xs:element type="xs:string" name="dimension"/>
      <xs:element type="xs:string" name="transmission"/>
      <xs:element type="xs:nonNegativeInteger" name="km_driven"/>
      <xs:element type="xs:positiveInteger" name="price"/>
      <xs:element type="xs:string" name="insurance" minOccurs="0"  />
      <xs:element name="fuel_economy" minOccurs="0" >
        <xs:simpleType>
          <xs:restriction base="xs:decimal">
            <xs:minInclusive value="5"/>
            <xs:maxInclusive value="50"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element type="xs:string" name="no_of_owner" minOccurs="0" />
    </xs:sequence>
    <xs:attribute type="xs:string" name="carid" use="required"/>
```

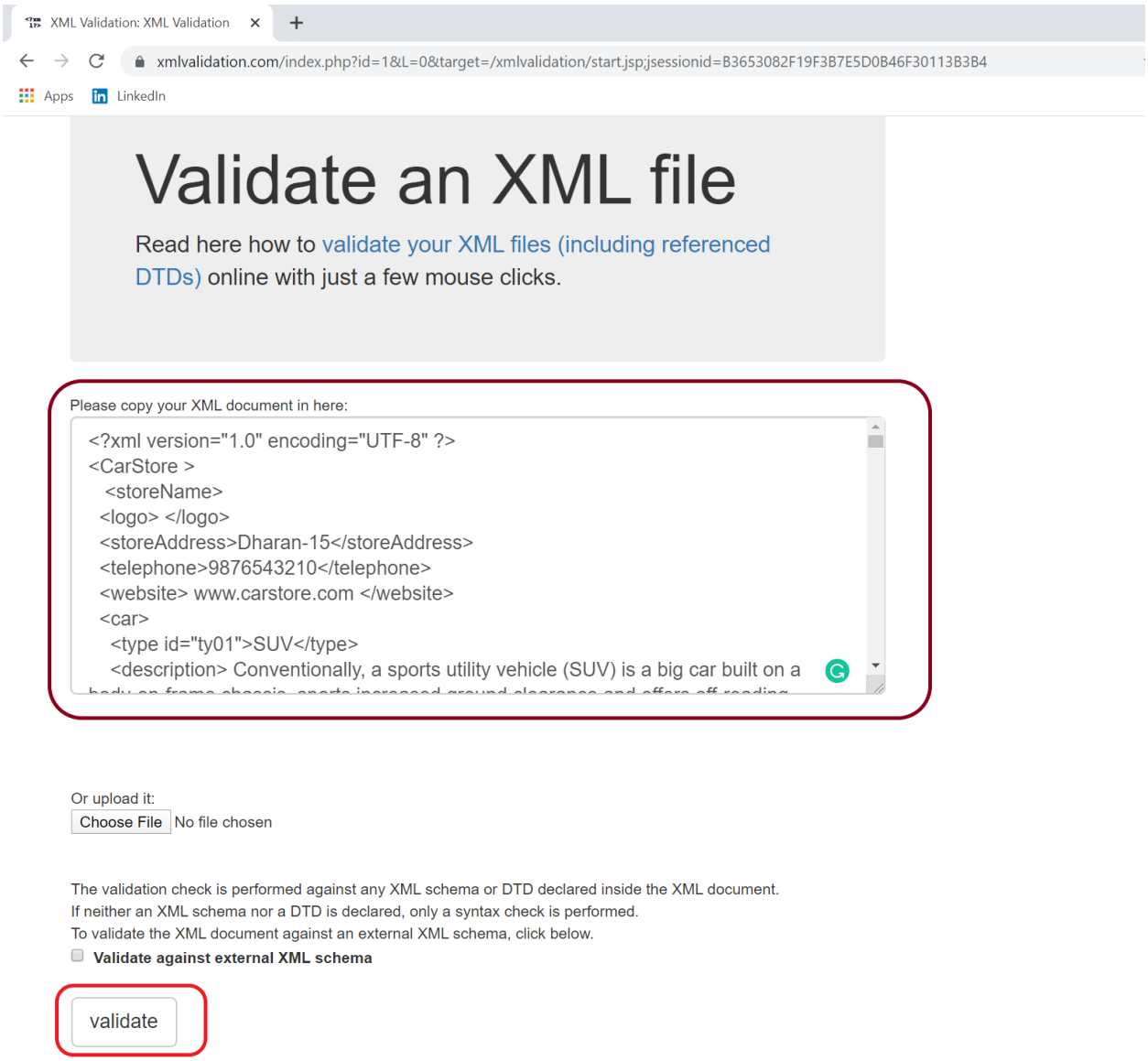
```
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

4. Testing

4.1 : Checking well formedness of XML with the help of an online xml validator.

Action:	Well formedness of xml code was checked in xml validation website.
Expected Result:	No error message will be displayed.
Actual Output:	Message with no error was displayed as per expected result.
Test Result:	Test successful

Table 1: Checking well formedness of XML with the help of an online xml validator.



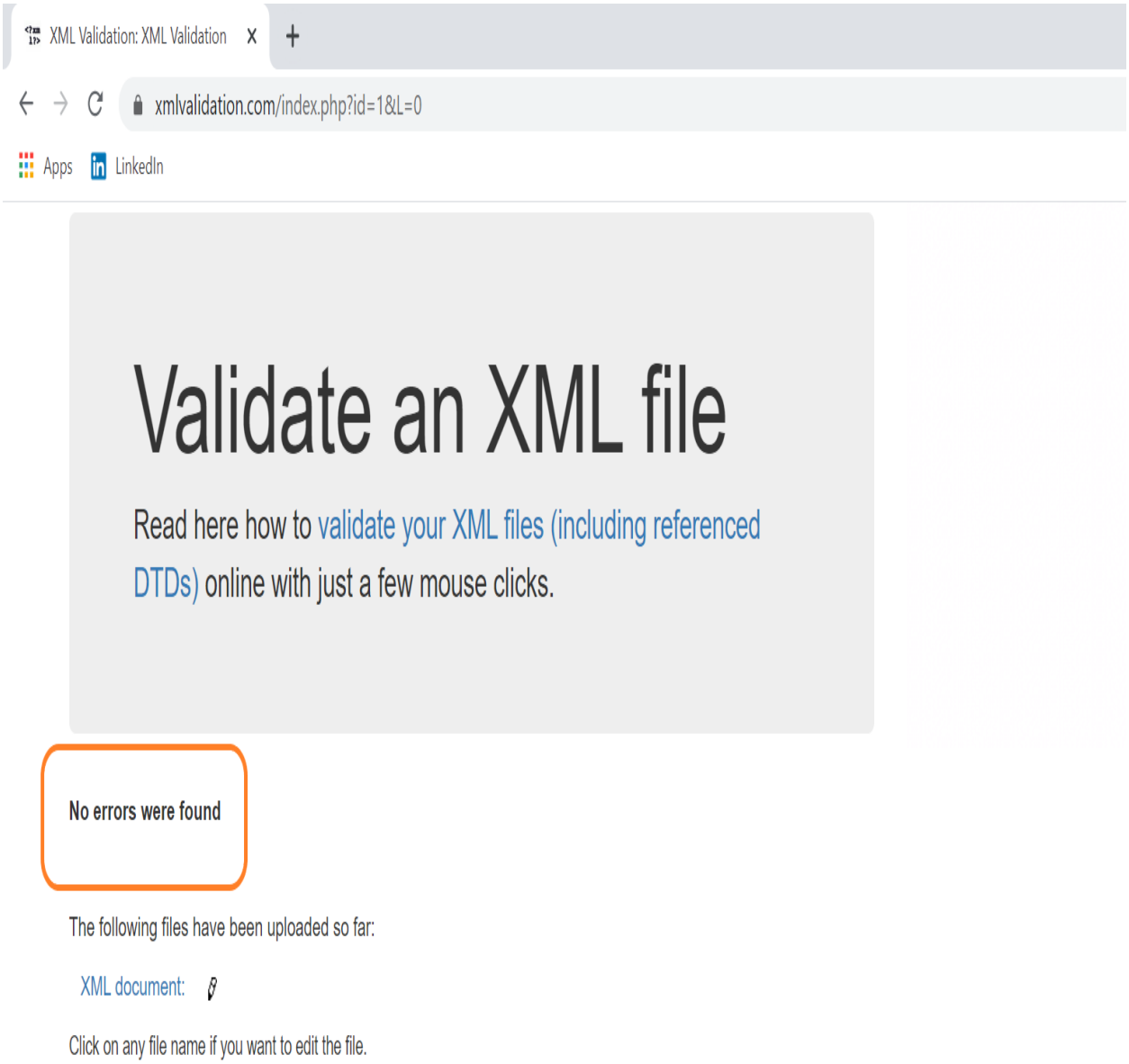


Figure 2: Checking well formedness of XML with the help of an online xml validator.

4.2 : Checking Schema well-formed and validity with the help of an online schema validator.

Action:	Valid and Well-formed schema was checked in W3C XML Schema (XSD) Validation online website.
Expected Result:	No error message will be displayed.
Actual Output:	Message with no error was displayed as per expected result.
Test Result:	Test successful

Table 2 : Checking Schema well-formed and validity with the help of an online schema validator.

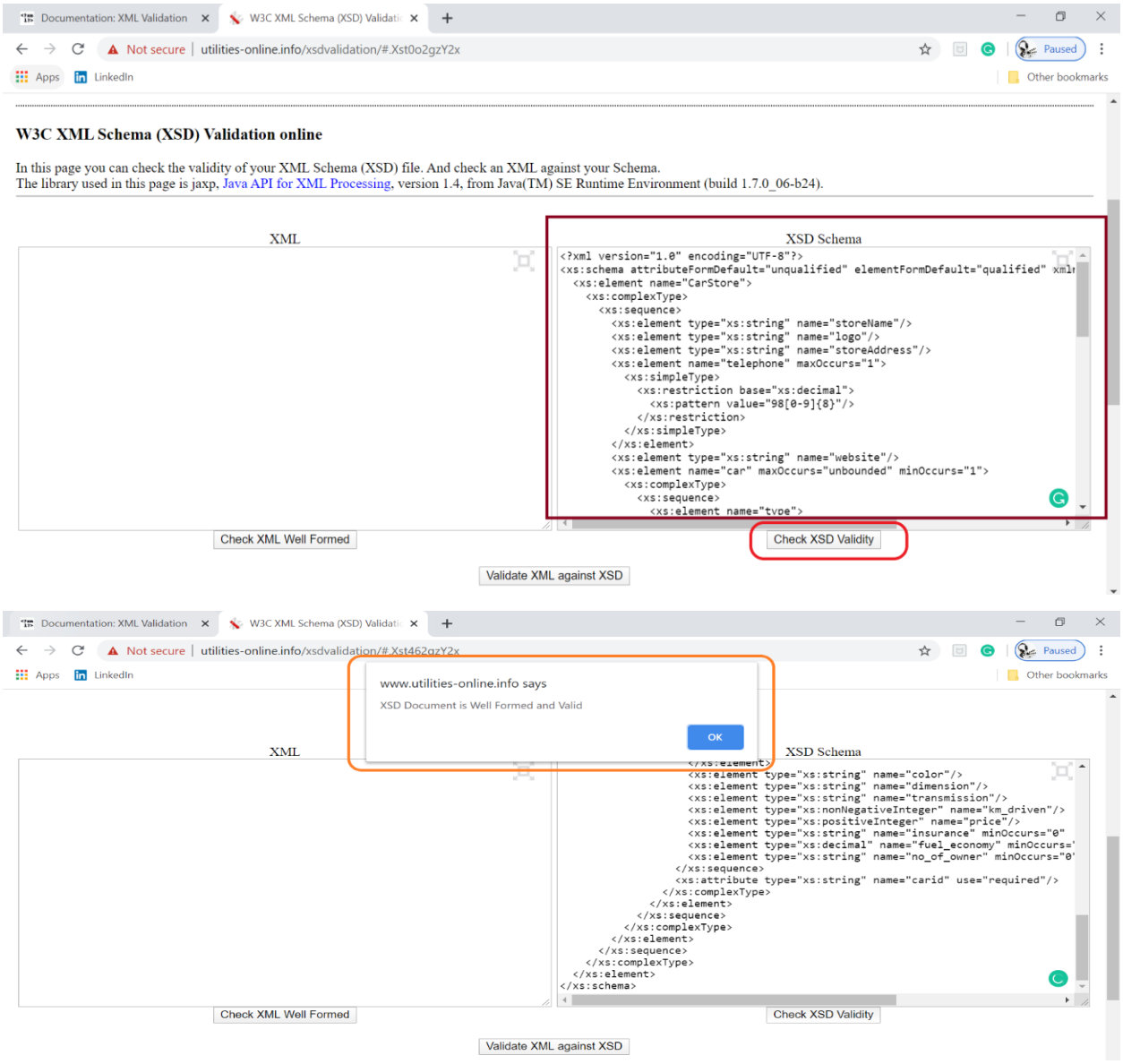
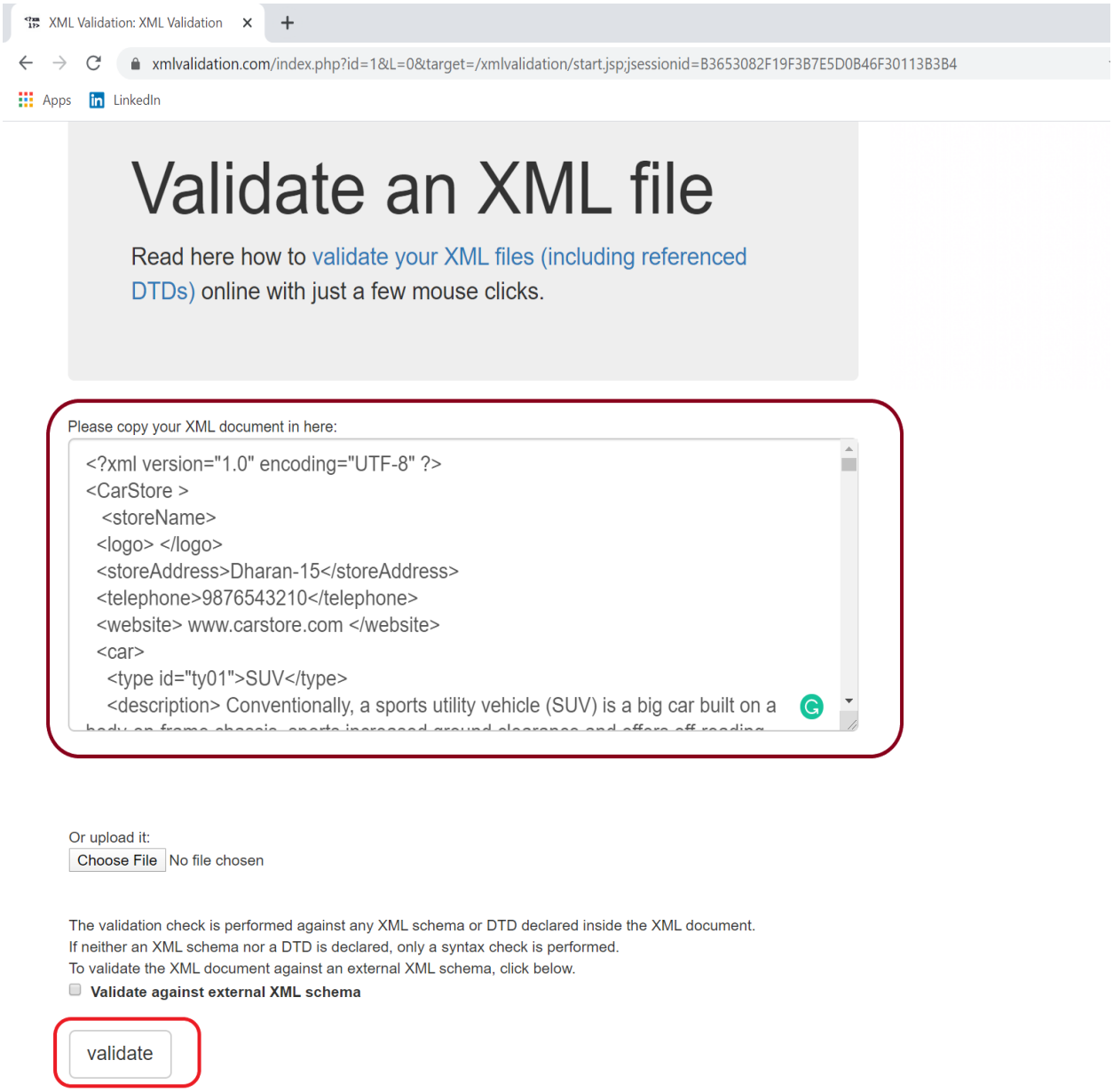


Figure 3: Checking Schema well-formed and validity with the help of an online schema validator.

4.3 : Validating XML and Schema with the help of an online validator.

Action:	All xml and schema code were validated in xml validation website
Expected Result:	No error message will be displayed.
Actual Output:	Message with no error was displayed as per expected result.
Test Result:	Test successful

Table 3: Validating XML and Schema with the help of an online validator.



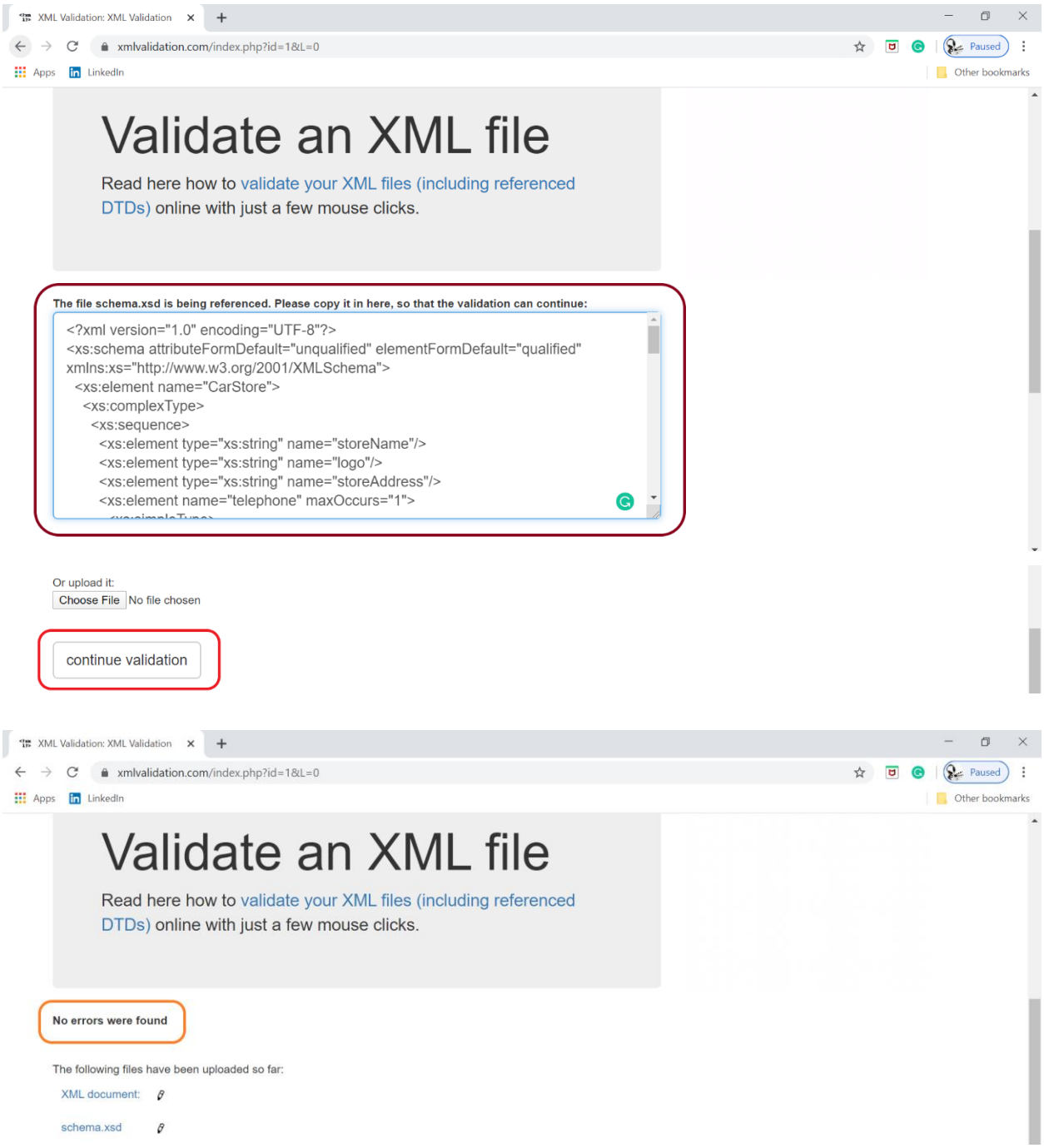


Figure 4: Validating XML and Schema with the help of an online validator.

4.4 : CSS validation with the help of an online CSS validator.

Action:	CSS code was validated in CSS validation website
Expected Result:	No error message will be displayed.
Actual Output:	No error message displayed as per expected result.
Test Result:	Test successful

Table 4: CSS validation with the help of an online CSS validator.

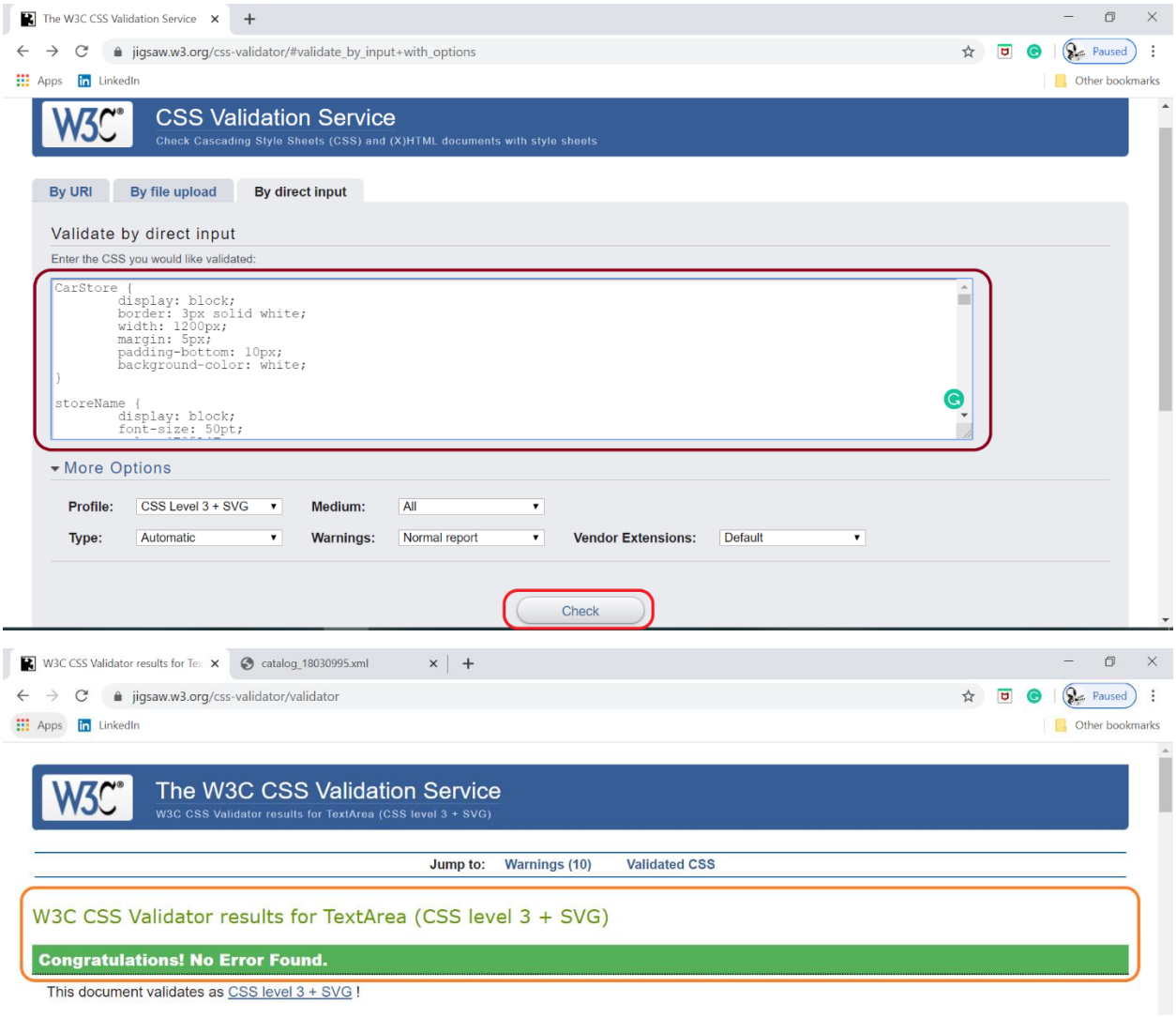


Figure 5: CSS validation with the help of an online css - validator.

4.5 : Running XML without CSS in the browser.

Action:	Browsing XML without CSS in the browser.
Expected Result:	All elements with data will be displayed.
Actual Output:	All elements were displayed as per expected result.
Test Result:	Test successful

Table 5: Running XML without CSS in the browser.

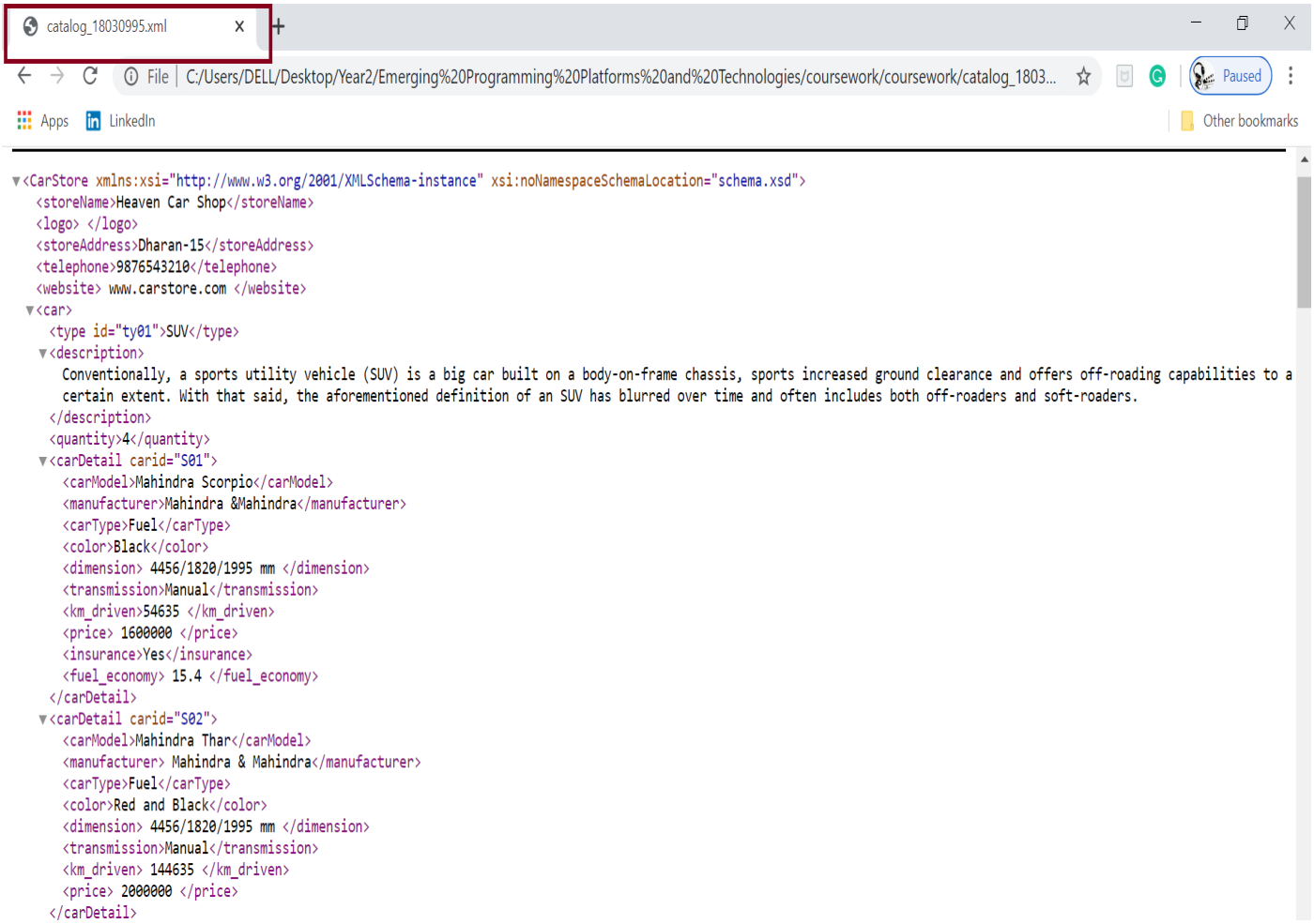


Figure 6: Running XML without CSS in the browser.

4.6 : Running XML with CSS in the browser.

Action:	Browsing XML with CSS in the browser.
Expected Result:	The XML document will display car details in an informative way.
Actual Output:	Car details displayed as per expected result.
Test Result:	Test successful

Table 6: Running XML with CSS in the browser.

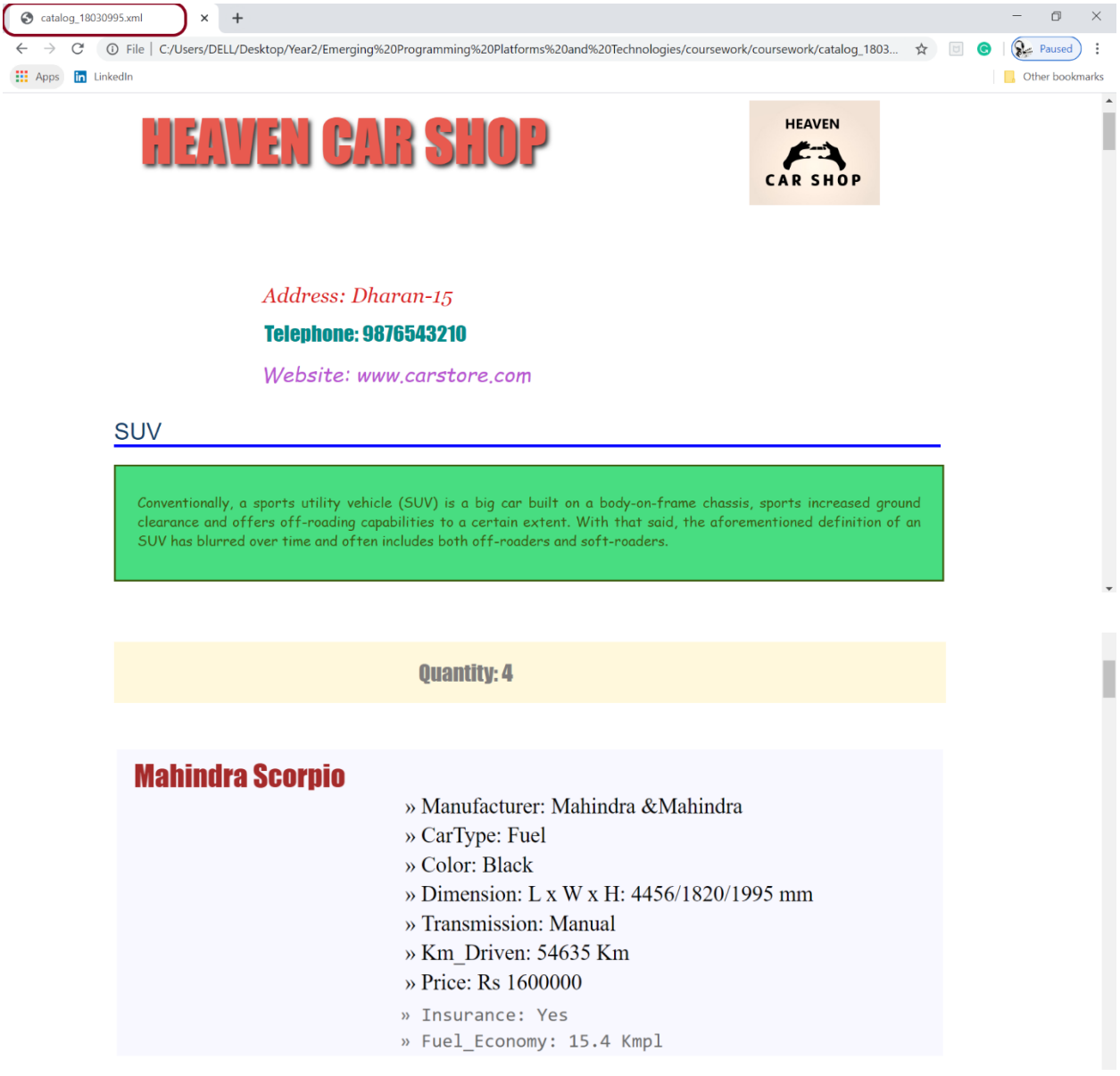


Figure 7: Running XML with CSS in the browser.

4.7 : Validating XML document with insufficient value in telephone element.

Action:	While inserting value, the telephone element was filled with insufficient value.
Expected Result:	The input error will be displayed in the xml document.
Actual Output:	Error message displayed as per expected result.
Test Result:	Test successful

Table 7: Validating XML document with insufficient value in telephone element.

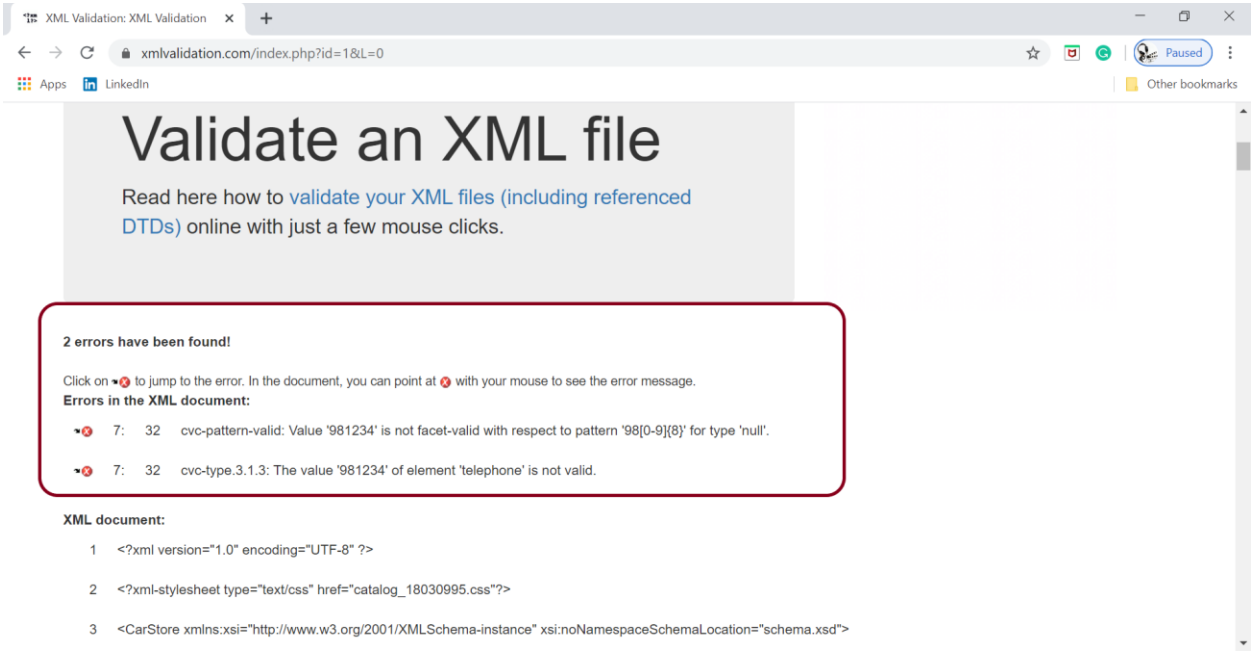


Figure 8 : Validating XML document with insufficient value in telephone element.

4.8 : Validating XML document with string value in price element.

Action:	While inserting value, the price element was filled with string value.
Expected Result:	The input value invalid message will be displayed.
Actual Output:	Error message displayed as per expected result.
Test Result:	Test successful

Table 8 : Validating XML document with string value in price element.

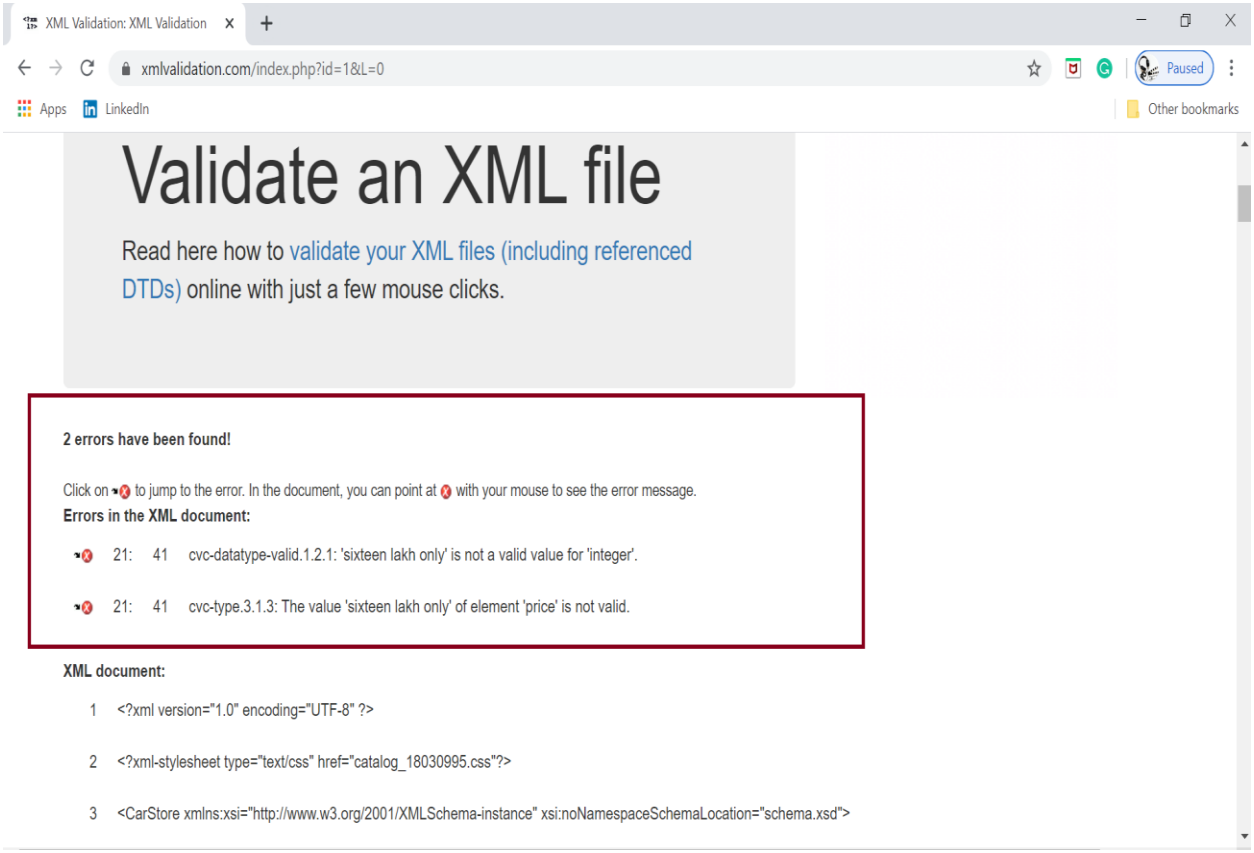


Figure 9: Validating XML document with string value in price element.

4.9 : Validating XML document with different value in carType element.

Action:	While inserting a value, the carTpye element was filled with another value.
Expected Result:	The input value invalid message will be displayed.
Actual Output:	Error message displayed as per expected result.
Test Result:	Test successful

Table 9: Validating XML document with different value in carType element.

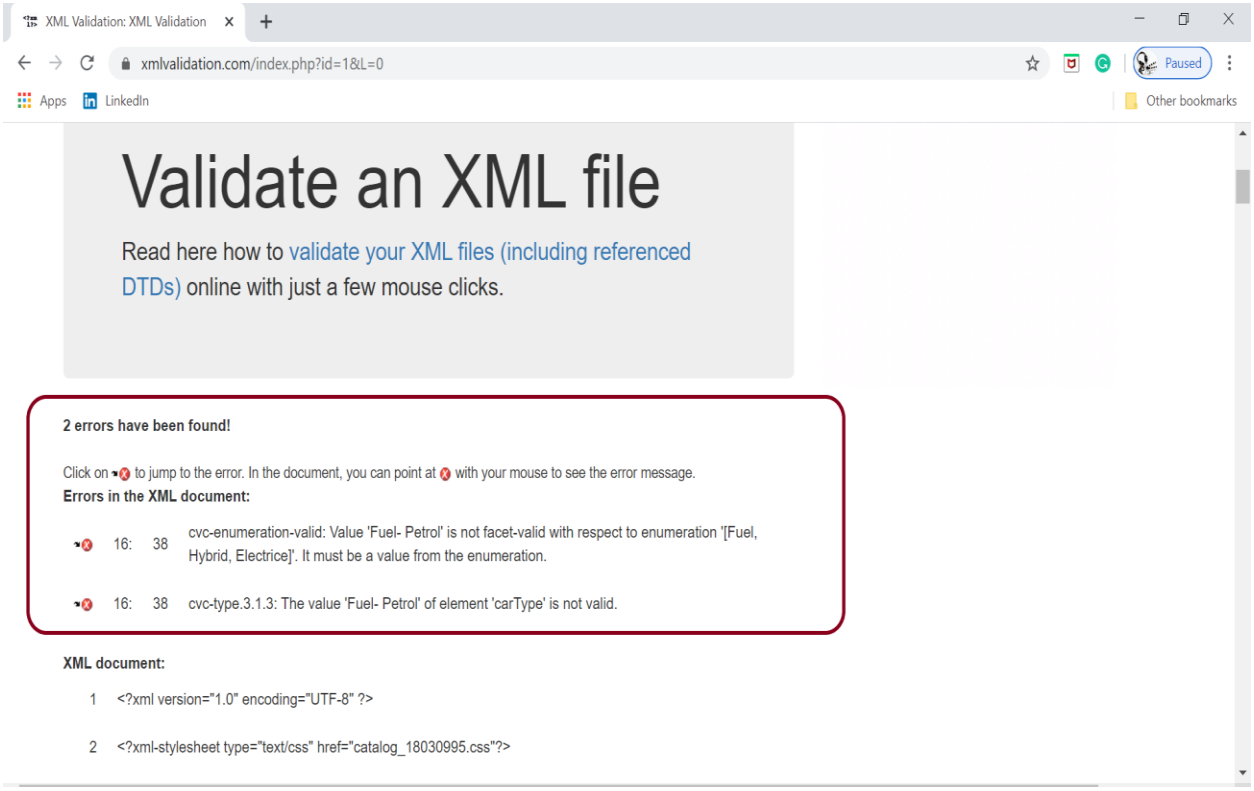


Figure 10: Validating XML document with different value in carType element.

4.10 : Validating XML document with greater value in fuel_economy element.

Action:	While inserting value, the fuel_economy element was filled with greater value.
Expected Result:	The input value invalid message will be displayed.
Actual Output:	Error message displayed as per expected result.
Test Result:	Test successful

Table 10: Validating XML document with greater value in fuel_economy element.

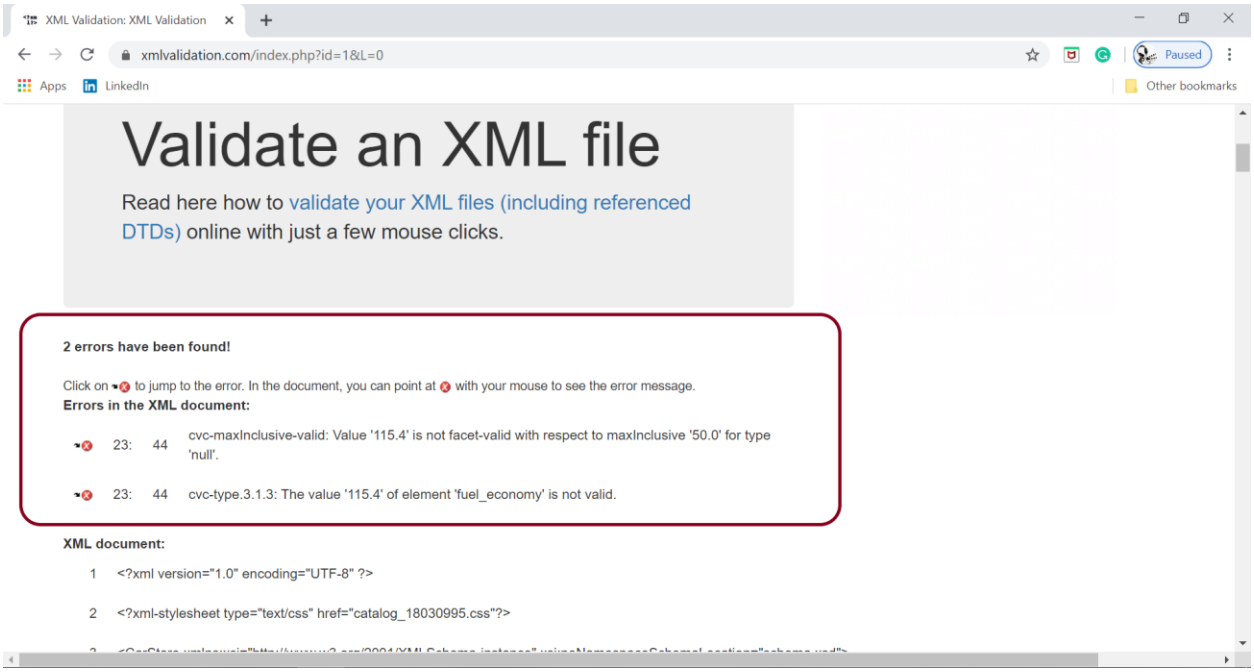


Figure 11: Validating XML document with greater value in fuel_economy element.

5. Coursework Development

The development of the course work begins with the creation of XML tree diagrams according to the requirements of the car store. Then an XML for the store was developed by defining its prolog on top of which its version and character encoding are defined. To have proper and valid XML, a root element was initialized along with its various required child and sub-child elements. According to the requirement of the car store, specific data were filled in the elements.

An XML Schema was developed by describing the structure of a car store's XML document. The schema was defined for all the elements and attributes that can appear in a document with its related data type. Different element restrictions, data formats, and optional elements were declared in the schema to validate the correctness of data in XML documents.

Then, CSS was developed for the system to define styles for an XML element, including the design, layout, and variations to display on a web browser. In CSS, different font families, sizes, and colors were used for elements. A list and border were created, the image was used for the logo and the floating box was also created to make the XML document look attractive.

Many tools have been used in this coursework to successfully complete the entire project. The tool used for this coursework were: -

MS Word

Microsoft Word is a visual word processing program developed by a Microsoft computer company in which users can write (Janssen & Janssen, 2019). It allows users to type and save documents. MS Word is used for creating a report that describes this project. Its features have made it easier to make a report. I have created tables, inserted pictures, formatting the references, layout of the page, and so on.

Sublime Text Editor

Sublime Text Editor is a popular text editor that is commonly used by developers which includes wide-ranging features such as Syntax Highlight, Auto Indentation, File Type Recognition, Sidebar, Macros, Plug-in and Packages that make working with code base simple (Tutorials Point, 2006).This text editor is used for writing the code of XML, Schema and CSS for developing the model system for online used car shop for storing car details.

Draw.io

Draw.io is a free web browser diagramming application that allows users to create and share diagrams and it offers an intuitive interface with drag & drop features, customizable models and a vast catalog of shapes .Draw.io is used for creating tree diagram that describes about the structure of xml document.

Snipping Tool

The Snipping tool is a screenshot app included in and later versions of Windows Vista that allows the user to take all or part of a screenshot in various ways and save that image. Snipping tool is used for taking screenshot images for developing testing reports for this project.

6. Limitation of DTD and CSS

6.1 : Limitation of DTD:

A Document Type Specification (DTD) is a specific document that defines and restricts the meaning or collection of statements which follow the rules of the Standard Generalized Markup Language (SGML) or the Extensible Markup Language (XML), a subset of SGML. The valid build blocks of an XML document are specified by a DTD. This specifies the structure of the text, with a list of validated elements and attributes (Rouse, 2010).

It has many benefits and inconveniences. Some of its benefits are that we can define our own XML file format and it helps us check the validity of XML files. The limitations of DTD are listed below:

- We need to learn different DTD syntax because it is not written in XML syntax and only supports text string data type.
- In DTD, data typing is not possible, the modularity of code is limited, and inheritance is not allowed.
- Larger DTDs are incredibly hard to recognize and sustain.
- DTD has limited ability to monitor white space and Namespaces are not provided.
- There are no default values for the elements, and default attributes must be set when declared.
- Documentation support is limited because the standardized documentation features available for schema notation cannot be used (sharat, 2007).

6.2 : Limitation of CSS

Cascading Style Sheets (CSS) is a style sheet language used to describe a document written in a markup format, such as HTML. CSS describes styles for your web pages including the architecture, layout and display variations for various devices and screen sizes. The limitation of CSS are as follows:

- CSS has various levels such as CSS, CSS 2, CSS 3. Because of this non-developers and beginners are confused.
- CSS has an issue with fragmentation, as though one CSS fits perfectly with one browser but does not work with another.
- Developers need to check compatibility of CSS that runs over many browsers.
- In CSS, lack of security is a concern.
- CSS cannot perform any logical operations like if/else, for/while, +/-, etc.
- CSS can’t request a webpage (SAHU, 2018).

7. Critical Evaluation

After numerous efforts and hardship, I was able to complete this coursework which was related to developing a model system for an online used vehicle shop using XML. Developing this coursework as per the scenario was a major challenge for me, I had faced several obstacles during the completion of this course work. While I was executing the first task, as per the requirements creating the XML document was not easy because the appropriate data and different roots, child, and sub-child elements for any company are hard to find and various errors in the code occur while implementing the requirements. Similarly, during the construction of the XML tree diagram, it was hard to design document structure with optional child elements and other elements with its occurrence in XML documents.

Developing schema was the most challenging part of this coursework. For learning the concept of schema and its importance, we must take an online class conducted by college due to COVID-19. While joining online classes different difficulties were faced like net problems and not understanding clearly and consulting with teacher for problems was difficult. While developing a schema for this coursework, the problems were defining the elements and attributes that appear in a document as compulsory or optional, validating data types for each element and attributes. Creating data restriction according to requirements, case sensitive elements problem was also one of the main problems while creating a schema. A small mistake in elements and data brings errors in both schema and XML. Likewise, other several problems were faced while creating the schema but I was successful in completing the schema with the guidance of the slide present in the google classroom, a lot of research, and the help of the module leader.

Finally, creating CSS was not that much challenging as developing Schema. The challenging part of CSS was to make an attractive system by adding different CSS styles, font families, font sizes, creating lists and displaying the logo of the store. All the problems of CSS were solved with the help of slides and online class videos. Doing this coursework was a great experience, I have learned about the basic concept of XML, schema, and CSS.

8. Conclusion

This coursework was related about developing an XML system for an online used vehicle shop for storing car details. This course work finally helped us to build a framework for any organization using XML, Schema and CSS.I learned how to create an XML document for any organization that has specific elements of root, child, and sub child. A document structure was drawn with optional elements and other required elements with its occurrence in XML documents.

Developing schema as per the required scenario was the most difficult part where schema describes and validates all the structure and the content of xml data according to shop requirement. While developing the schema I have learned to define elements, different data types and attributes. I have also used schema indicators and restrictions for accepting specific types of values for some XML elements and attributes. Likewise, l learned to style the XML document with the help of CSS using border, different font families, font styles, colors, layouts etc. With the use of CSS an XML document looks attractive and displays car details in an informative way.

Developing this coursework as per the scenario was a major challenge for me, I had faced several obstacles during the completion of this course work. With the help of studying material present in the google classroom and module leader, all the questions and problems that emerged during the course work were easily tackled. I had done a lot of research and study for completing this coursework. This coursework provided a very good understanding of XML, Schema and CSS. This coursework makes me adequate experience in designing and implementing framework using xml, schema, and CSS for developing a model system for an online used vehicle shop and providing knowledge and understanding the value of necessary research and problem solving.

References

Data, R., 1998. *W3Schools.* [Online]
Available at: https://www.w3schools.com/css/css_intro.asp
[Accessed 21 May 2020].

Data, R., 1998. *w3schools.com.* [Online]
Available at: https://www.w3schools.com/xml/xml_whatis.asp
[Accessed 20 May 2020].

Janssen, D. & Janssen, C., 2019. *Technopedia.* [Online]
Available at: https://www.techopedia.com/definition/3840/microsoft-word
[Accessed 25 May 2020].

Rouse, M., 2010. *WhatIs.com.* [Online]
Available at: https://whatis.techtarget.com/definition/Document-Type-Definition-DTD
[Accessed 23 May 2020].

SAHU, P., 2018. *Advantages and Disadvantages of CSS Everyone Should Know.* [Online]
Available at: https://www.motocms.com/blog/en/advantages-and-disadvantages-of-css/
[Accessed 23 May 2020].

sharat, 2007. *What are the disadvantages of DTD.* [Online]
Available at: https://sharat.wordpress.com/2007/05/18/what-are-the-disadvantages-of-dtd/
[Accessed 23 May 2020].

Tutorials Point, 2006. *Sublime Text Tutorial.* [Online]
Available at: https://www.tutorialspoint.com/sublime_text/index.htm
[Accessed 25 May 2020].

Tutorials Point, 2006. *Tutorials Point Simple Easy Learning.* [Online]
Available at: https://www.tutorialspoint.com/xml/xml_schemas.htm
[Accessed 20 May 2020].