|  |
| --- |
| Faculty of Applied Sciences and Technology |
| **XML Data Processing** |
| ITC5202 – Project 1 |
|  |
| **Submitted By : Sandeep Das and Rohan Vasudev Patel** |
| **2/24/2022** |

|  |
| --- |
| This document explains how to process Customer/Order XML data |

**Under the Guidance of: Prof Shahdad Shariatmadari**

Table of Contents

[Step 1 : 2](#_Toc96970217)

[Step 4 : Design XSLT 12](#_Toc96970218)

[Step 5 : XPath and XSLT 16](#_Toc96970219)

[Step 6: Use JavaScript to process XML data 21](#_Toc96970220)

[Step 7:](#_Toc96970221)Use JS to implement search by customerID …………………………………………………………………………………………23

Step 8: Replace child Tag with attribute and repeat Step 5 and Step 7 ……………………………………………………………………..30

Summary: ……………………………………………………………………………………………………………………………………34

Declaration: ………………………………………………………………………………………………………………………………….36

# Step 1 :

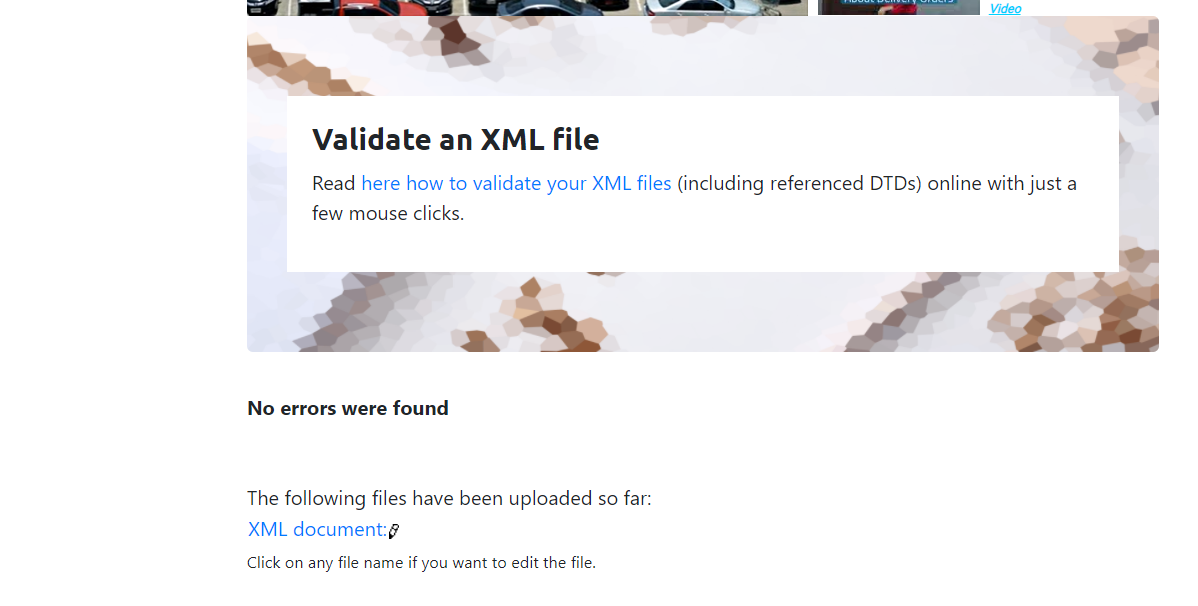
(Describe you answer. How did you prove that the document is well-formed and valid? Add screenshots)

Answer->

1. -this document is well formed because it runs without an any error in the browser as you can see in the below screenshot



1. - this document is valid also because it show no error in the xml validator as you can see in the below screenshot



1. - is there any namespace?

Answer-> yes there is a one default namespace which is written below

xmlns="http://www.adventure-works.com">

**Step 2 and 3 : XML Structure**

(1) Explain the major steps that you take to create DTD. Did you create a .dtd file, or you keep the DTD declaration inside the XML file? Why?

(2) Explain the major steps that you take to create XML Schema.

(3) How did you validate them? Add screenshots.

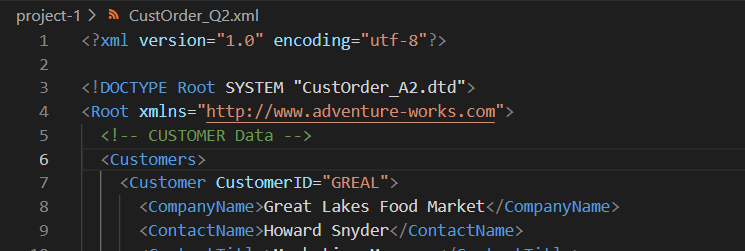
(4) Compare the DTD and Schema and show how DTD declaration are matched with Schema.

Answer-1)

I have created the external dtd which is shown below and the reason I choose external dtd between external and internal dtd is you can link the external dtd to any file apart from this the actual file is already big and if you add internal then it will become more long and when you are doing the other question then there is chance that you edit the dtd unintentionally so it is better to use external and it also look clear.so that’s the reason I have used external dtd

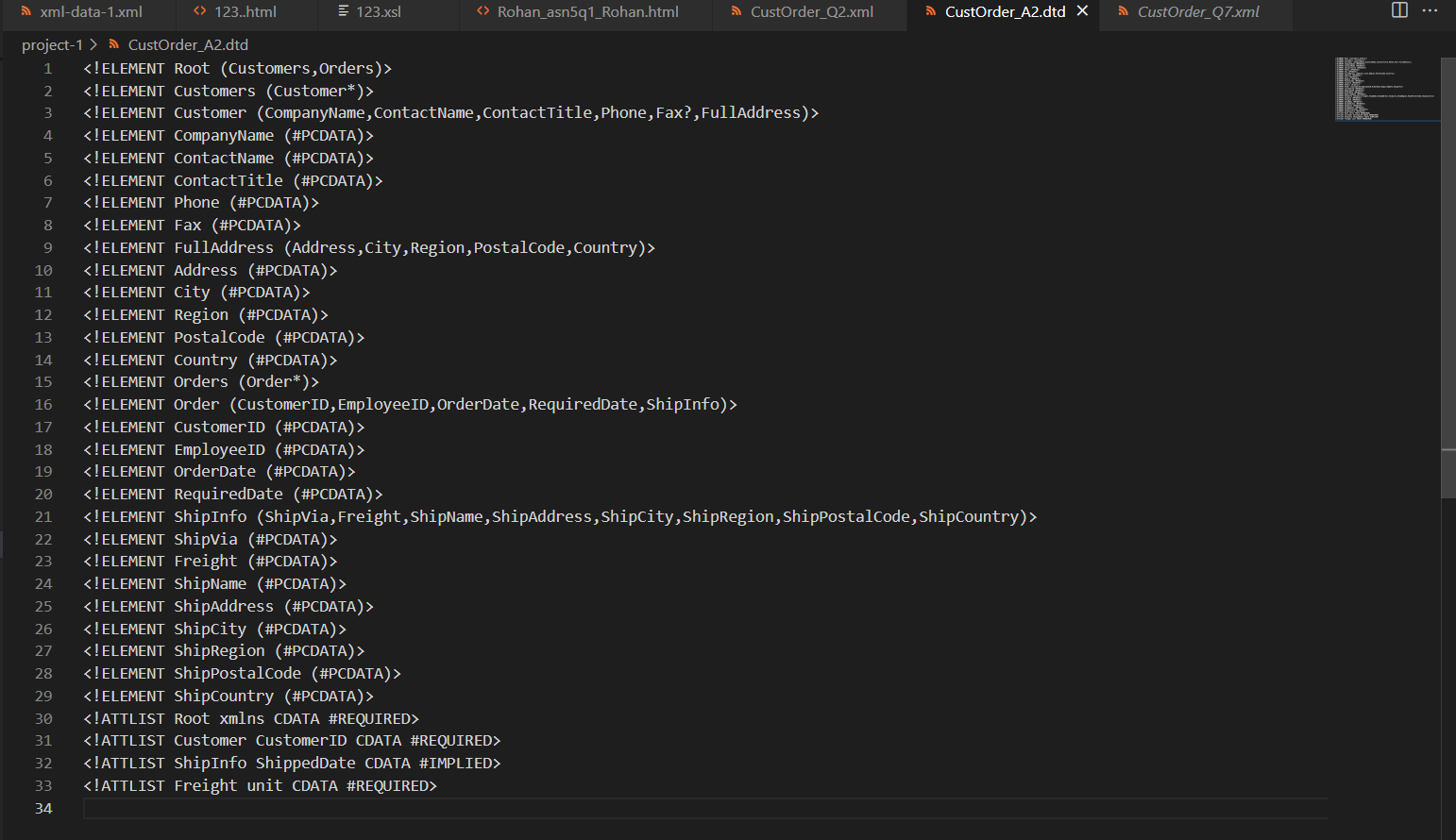
Code

Link to the external dtd



Link to external DTD

EXTERNAL DTD

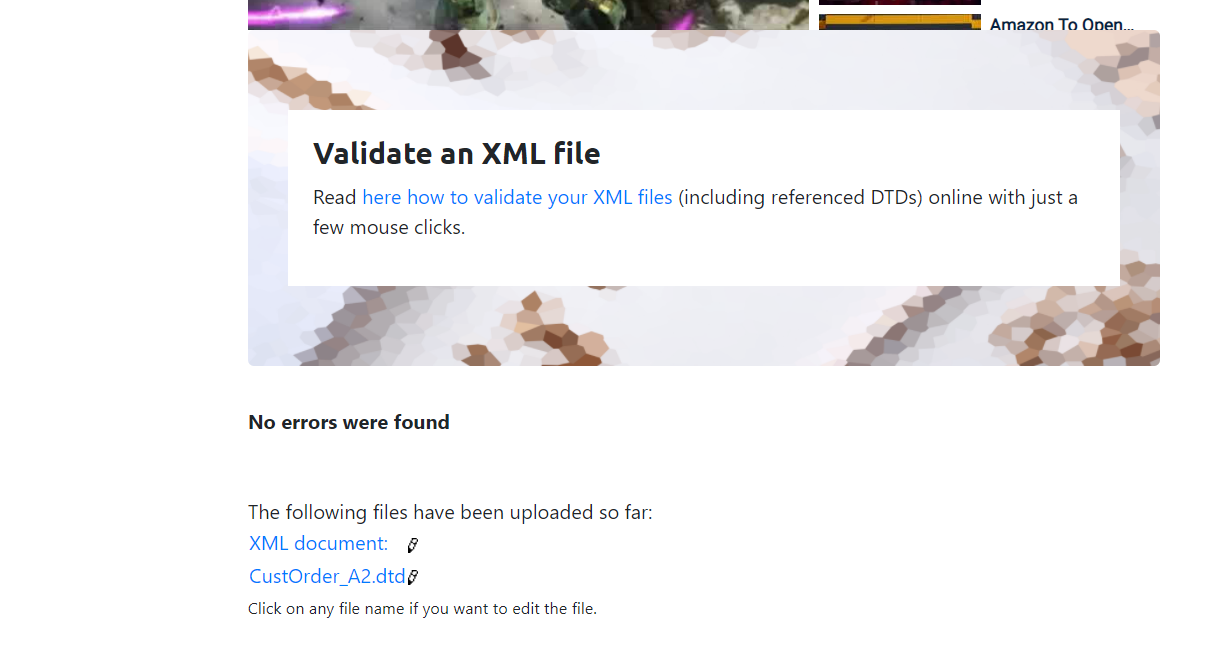


For the attribute

For the orders element

For the customers element

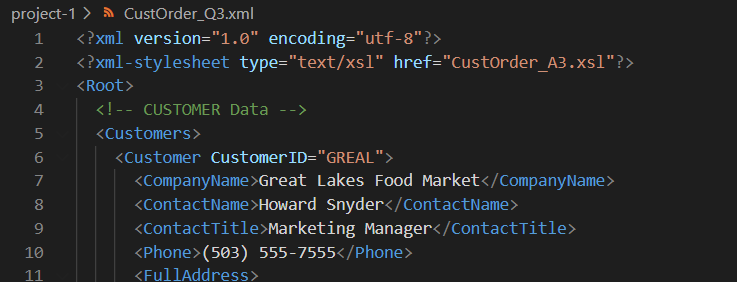
Output



1. Explain the major steps that you take to create XML Schema.

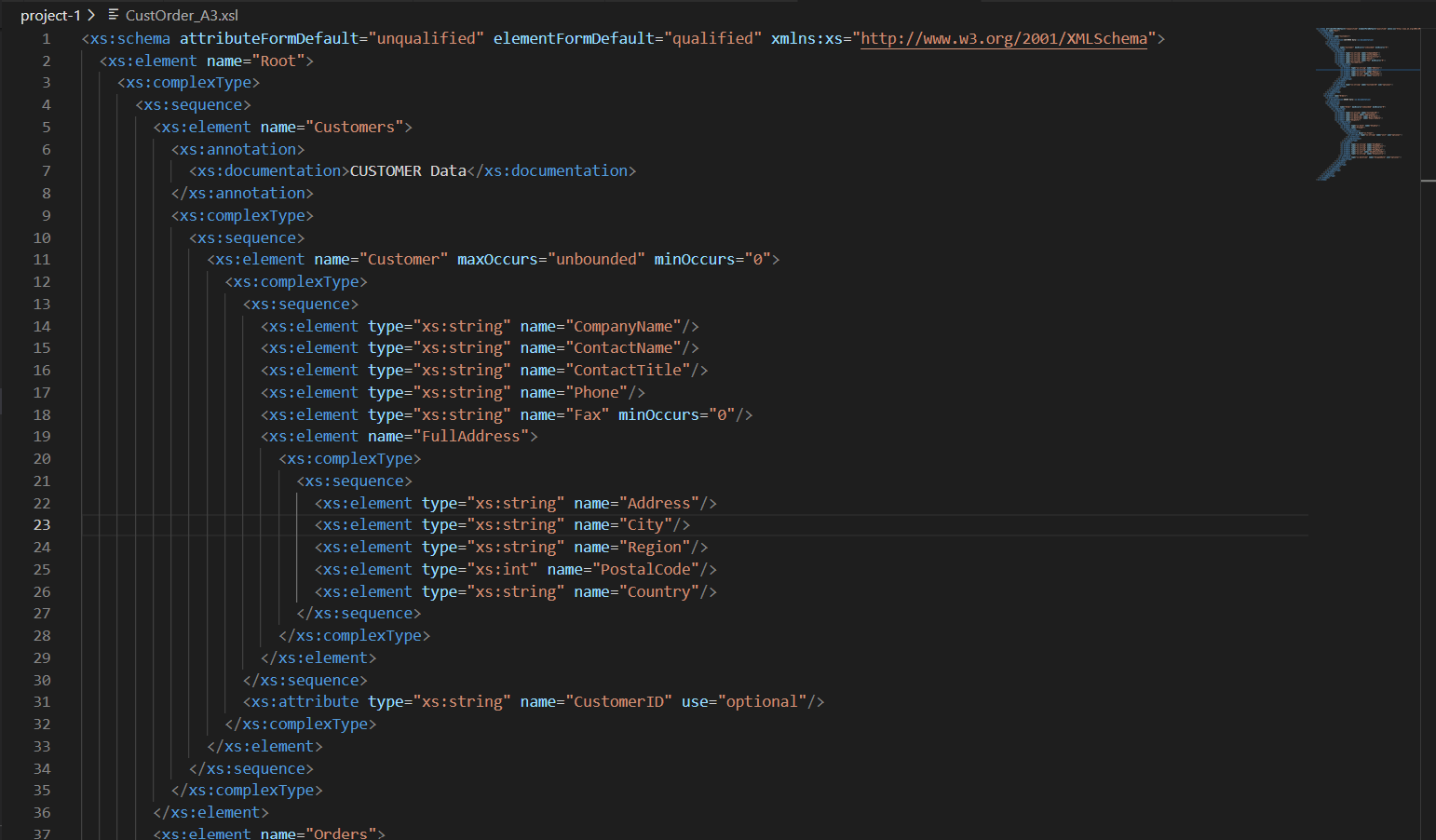
Answer->

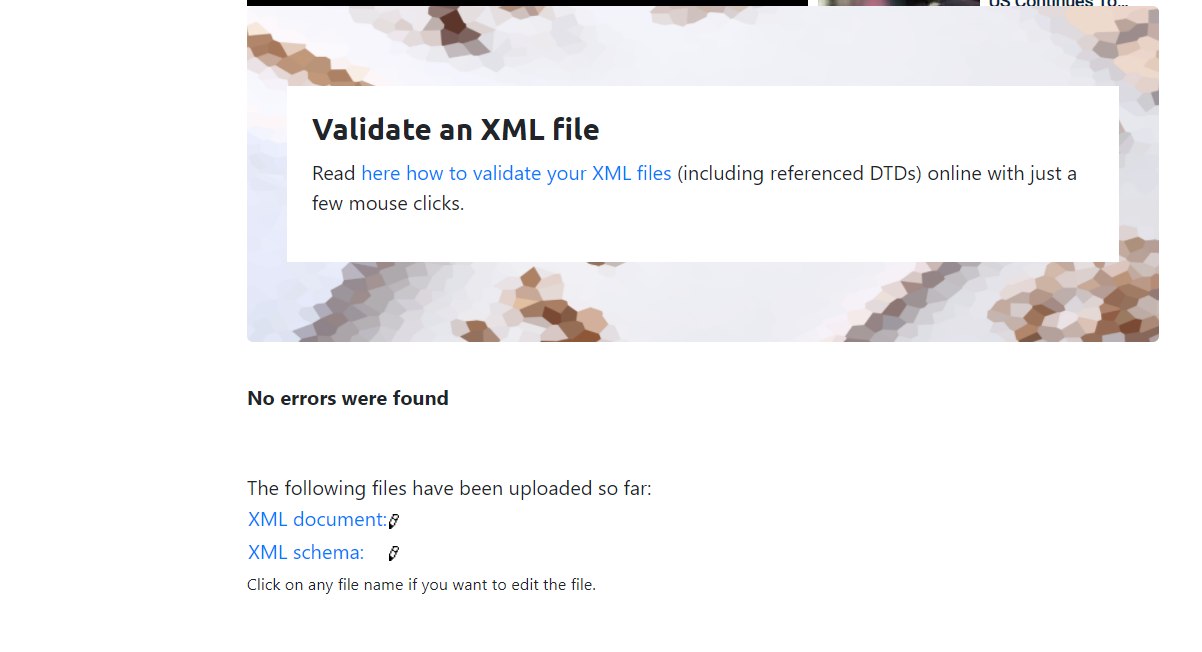
Link to xsl file



Link to xsl file

Xsl file



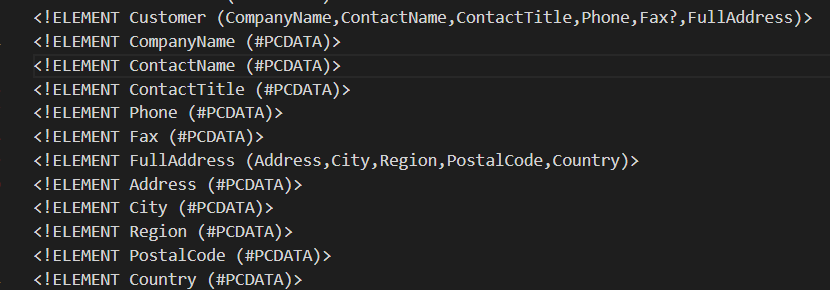
1. How did you validate them? Add screenshots.
2. 
3. Compare the DTD and Schema and show how DTD declaration are matched with Schema.

Answer-> first lets compare the element how both dtd and xsl declare element

In dtd elements are declared like this

<!ELEMENT ELEMENT NAME (CHILD1,CHILD2)>

For example here we have declared the customer element like this in dtd



Where as in xsl we have declared the same element like this



In the xsl we specify the element one by one by using the xs:sequence here fax is used only one time so in xsl they use minOccurs where as in dtd we add one questionmark after that as it is seen in the above dtd screenshot

# Step 4 : Design XSLT

Write an XSLT document to transform order.xml into an HTML document that displays all "**Customer's Data**" in a HTML table format.

**CustOrder\_Q4.xsl**

Text

Description automatically generated

**CustOrder\_Q4.xml**





Text

Description automatically generated

Display internal tag values of Address attributes.

Display other tag values inside loop

To get the customer ID attribute

We will loop through all customers here by using this XPath inside for-each statement

Table

Description automatically generated

# Step 5 : XPath and XSLT

Write an XPATH query to show all orders that have not shipped yet (don't have "ShippedDate") Graphical user interface, application

Description automatically generated

Omit the ShipInfo elements that don’t have the ShippedDate attribute

Design a new XSLT file that displays all "**Orders**" info (CustomerID, EmployeeID, OrderDate and RequiredDate), but displays these items (which you selected with XPATH) in red color.

Text

Description automatically generated





Text

Description automatically generated

If the elements have ShipDate attribute, then display as a normal font

Apply class redText to the selective elements to display them in red font

Check if the ShipInfo tag does not have the ShippedDate attribute

Go to the xpath /Root/Orders/Order tag and loop on each element

Table

Description automatically generated

# Step 6: Use JavaScript to process XML data

You are going to redesign Step5 using JavaScript. You need Develop a proper JavaScript that shows all orders' info (CustomerID, EmployeeID, OrderDate and RequiredDate) that have not shipped yet (don't have "ShippedDate")Text

Description automatically generated

If the order/ShipInfo does not have the attribute, then apply the redText class for changing font colour to red.

Check if the order/ShipInfo has an attribute with the name ShippedDate or not. If the attribute is present, then don’t apply the font style

Block 1 : getting the XMLResopnse from XHR object



Table

Description automatically generated

# Step 7: Using Step7.html, you need to develop a proper JavaScript function which will be invoked when user clicks on the “Search” button. The JavaScript function, reads the CustomerID from webform, and processes the CustOrder.xml data and displays all Customer info and related orders that their CustomerID is matched with the given CustomerID. (display these fields: CompanyName, Country, Region, OrderDate, RequiredDate, ShippedDate, ShipAddress)

# • Use your creativity to display Customer and Order data properly to look like a report.

# • You need to add partial search to this program, so, if user enters “L”, the program should displays CustomerID’s has ‘L’ (or starts with ‘L’).

Graphical user interface, table

Description automatically generated

Code:

Text

Description automatically generated

**Fetching the XML file and storing its data into variables**

Creating The Skeleton And Adding Callback Functions

To Generate Report

Functions:

Text

Description automatically generated

If there is a search string present, only then append the option for export report to pdf

Check if order has a shipped date by checking length of attributes of ShipInfo tag inside Order tag, if not then populate table with Not yet shipped.

Appending output

Loop through the customers array where customer data is saved

If the customerId attribute of the current customer in loop includes the search id and if the search id is not blank

Then,

Inner loop orders array and check if any order has a customerId tag whose value matches to outer loop current customer, if yes then populate table by appending text to output string.

Set the element with result id to blank whenever the function is called.

Set the output text to blank whenever the function is called.

If the search input is not empty, only then add table element into output field

GeneratePDF function:

   References : https://codingshiksha.com/javascript/jspdf-tutorial-to-export-html-table-to-excel-and-pdf-document-using-jspdf-autotable-library-in-javascript-full-tutorial-for-beginners/

Text

Description automatically generated

Creating the PDF and downloading it

Using jsPDF library method to convert the table to JSON

**Creating object of jsPDF**

Fetching the table into elem variable

# Step 8:

Modify the CustOrder.xml by removing the element ( as a child of ) and add it as attribute to the element. save it as order\_modified.xml:



Step 5 on the modified XML

Graphical user interface, application

Description automatically generated

The modified xsl is the one on the right hand side in the image above. These changes were made to display the content when CustomerID is made an attribute of Order.

Table

Description automatically generated

Step 7 on the modified XML

# 

The changes that were made was to replace getElementsByTagName("CustomerID")[0].innerHTML to attributes[0].value

Our opinion

*Attribute vs Element:*

While parsing the XML we have used both attributes and elements to fetch our data. Our mutual understanding is that element tag names are much handy and robust to use while parsing XML. These are JS functions that help us find an element object irrespective of its cascading. Contrary to that, while fetching attributes, we can get errors if the attribute position is not mentioned correctly. If the attribute is not mandatory and is missed for an element, a proper exception handling must be done to prevent runtime errors.

*JS vs XSLT:*

Definitely JS has the winner point here, provided the ease of syntax usage. JS also offers multiple internal libraries that makes data manipulation much easier. Added to that, we also have the flexibility to add external libraries easily. Though libraries can be added in an XSL as well but the less amount of code and syntactical ethics makes JS stand out in terms of accessibility.

A screenshot of a computer

Description automatically generated

# Summary

This project is a joint effort of me and my team member Mr. Rohan Vasudev Patel.

We divided the work equally among ourselves.

**The first three steps and 8th step have been solved by Rohan:**

Step 1) Try to explore and understand the structure of the given XML data. Is the XML file a well-form document? Is it a valid document? Is there any namespace? .

Step 2) Design the DTD for this document and link it to the given XML file.

Step 3) Design XML Schema for this document and link it to the given XML file.

Step 8) Complete the followings:

 Modify the CustOrder.xml by removing the <CustomerID> element ( as a child of <Order> ) and

add it as attribute to the <order> element. save it as order\_modified.xml:

 Based on the new xml data file, redo the Step 5 and 7.

 From your understanding as developer, explain your idea about coding/processing XML element vs

attribute. Which one is easier to develop with JS or XSLT, process CustomerID as element-data or

attribute-data? Explain your answer based on your observation through this project.

**The 4th, 5th, 6th and 7th step have been solved by Sandeep :**

Step 4) Write an XSLT document to transform order.xml into an HTML document that displays all "**Customer's Data**" in a HTML table format.

Step 5) Write an XPATH query to show all orders that have not shipped yet (don't have "ShippedDate")

Design a new XSLT file that displays all "**Orders**" info (CustomerID, EmployeeID, OrderDate and RequiredDate), but displays these items (which you selected with XPATH) in red color.

Step 6) You are going to redesign Step5 using JavaScript. You need Develop a proper JavaScript that shows all orders' info (CustomerID, EmployeeID, OrderDate and RequiredDate) that have not shipped yet (don't have "ShippedDate")

Step 7) Using Step7.html, you need to develop a proper JavaScript function which will be invoked when user clicks on the “Search” button. The JavaScript function, reads the CustomerID from webform, and processes the CustOrder.xml data and displays all Customer info and related orders that their CustomerID is matched with the given CustomerID. (display these fields: CompanyName, Country, Region, OrderDate, RequiredDate, ShippedDate, ShipAddress)

• Use your creativity to display Customer and Order data properly to look like a report.

• You need to add partial search to this program, so, if user enters “L”, the program should displays CustomerID’s has ‘L’ (or starts with ‘L’).

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* ITC5202 – Project1 \* We declare that this assignment is our own work in accordance with Humber Academic \* Policy.

\* No part of this assignment has been copied manually or electronically from any other \* source (including web sites) or distributed to other students. \*

\* Names: Sandeep Das, Rohan Vasudev Patel

\* Student ID: N01472825 , N01469929

\* Date: 28th Feb 2022

\* \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*