|  |
| --- |
| Faculty of Applied Sciences and Technology |
| **XML Data Processing** |
| ITC5202 - Project |
| **Rankit Goyal** |
| **Anupriya Singh** |
| **2/23/2020** |

|  |
| --- |
| This document explains how to process Supplier/Product XML data …………………………. |

Table of Contents

[Question 1 : 2](#_Toc33042134)

[Question 2: 3](#_Toc33042135)

[Question 3 and 4 : XML Structure 4](#_Toc33042136)

[Question 5 : Design XSLT 5](#_Toc33042137)

[Question 6 and 9: XPath and XSLT 6](#_Toc33042138)

[Question 7 and 8: Use JavaScript to process XML data 7](#_Toc33042139)

[Question 10 8](#_Toc33042140)

[Bonus question 9](#_Toc33042141)

[Summary 10](#_Toc33042142)

# Question 1 :

(Describe the answer of this question.)

The XML DOM defines a standard way for accessing and manipulating XML documents. It presents an XML document as a tree-structure. In a node tree, the top node is called the root (or root node) which is orders in this xml document. Every node has exactly one parent, except the root (which has no parent).

A node can have a number of children. In this document, Orders have multiple order elements. Further, Order have many child elements that are customerid, status and item. Item element have child elements(name , price, qty).

Siblings (brothers or sisters) are nodes with the same parent. Item element have sibling of each other and order element have sibling of each other. Instock and itemid are the attributes given to item tag.

**All Elements:** orders,order,customerid,status,item,name,price,qty

Root element : orders

Child element of orders: order

Child elements of order element: customerid, status, item,

Child elements of item element: name , price, qty

Instock and itemid are the attributes given to item tag.

# Question 2:

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

Yes, it is a Well-formed document. The reason is as it follows all the rules given below:

1.Every start tag have an end tag.

2.It contain only one root element.

3.Nesting of elements with each other is proper.

4.Markup characters are properly specified.

5.In each element, name of the attributes is different.

6. All attribute values are enclosed in single or double quotation marks.

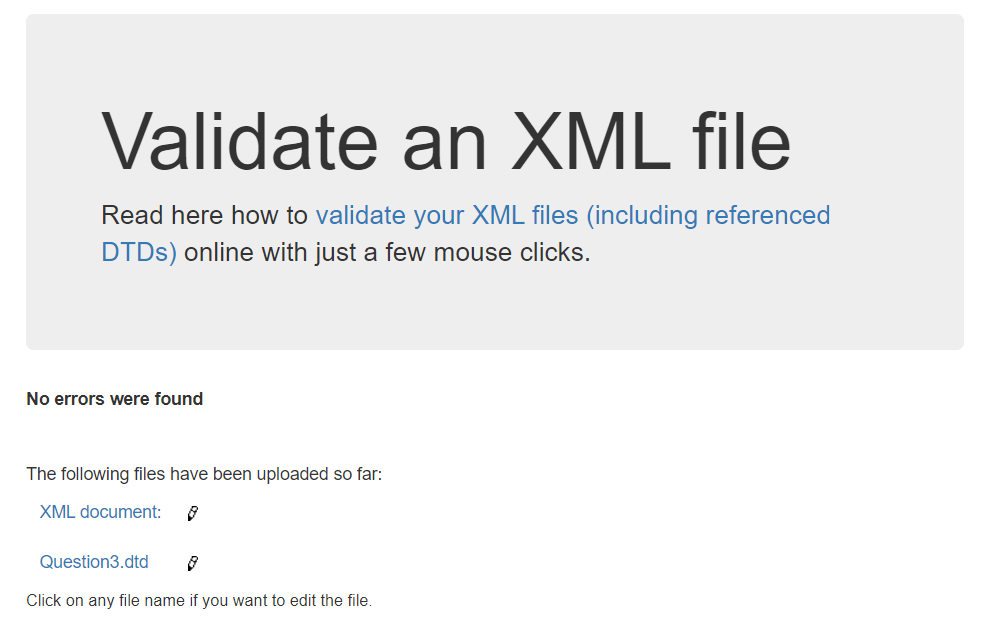
Valid XML Document

It is not a valid document.

The valid document is always a well-formed document but reverse is not true.

The XML document is a said to be valid if DTD is associated with it that is it conforms to the rules of the DTD..

To check whether the document is valid or not ,we use XML validation.com. If it is valid the message is displayed that No error were found as shown in the following screenshot.



# Question 3 and 4 : 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.

Steps to create DTD:

A **DTD** defines the valid building blocks of an XML document. It defines the document structure with a list of validated elements and attributes.

DTD includes the following declarations:

– Element Type Declarations

– Attribute List Declarations

An XML DTD can be either specified inside the document, or it can be kept in a separate document and then the document can be linked to the DTD document to use it.

We have created an external DTD as follows:

<!ELEMENT orders (order+)>

<!ELEMENT order (customerid, status, item+)>

<!ELEMENT customerid (#PCDATA)>

<!ELEMENT status (#PCDATA)>

<!ELEMENT item (name, price, qty)>

<!ELEMENT name (#PCDATA)>

<!ELEMENT price (#PCDATA)>

<!ELEMENT qty (#PCDATA)>

<!ATTLIST item

    itemid (B12 | F32 | M12 | PS93 | Q52 | SD93) #REQUIRED

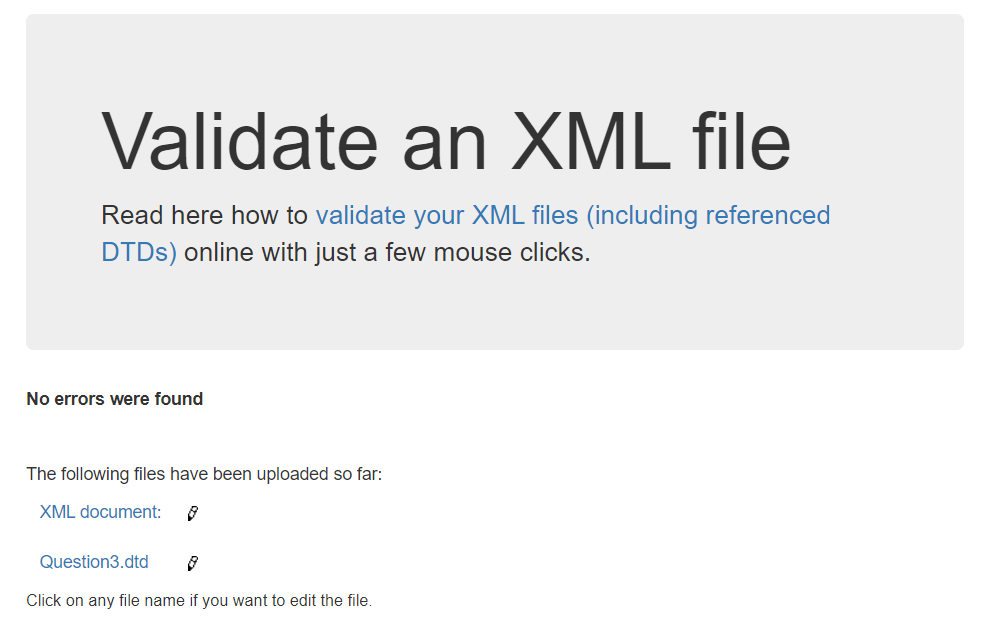
    instock (N | Y) #REQUIRED

>

We link it to the xml file:

<!DOCTYPE orders SYSTEM "Question3.dtd">

Output of validation:



We have created a separate DTD file and linked it with the xml document.

The benefits of using external DTDs is that they can be shared by more than one XML document. You can write a DTD once and multiple documents can refer to it. We can make changes to the central DTD, all documents that rely on the DTD are updated in at the same time.

Steps to create XMLSchema:

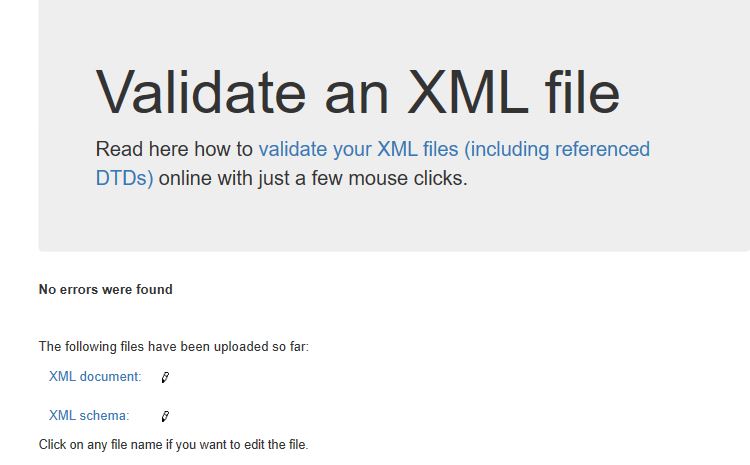
We have generated the schema using the XSD Generator.

The following line of code is added to the xml file to link XML Schema to it:

<orders xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespaceSchemaLocation="Question4.xsd">

Output of validation:



# Question 5 : Design XSLT

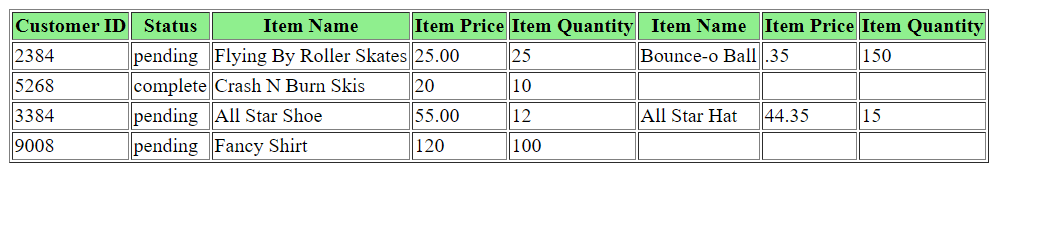
(Describe the major steps for designing the XSLT. Add screenshot of the output)

XSLT stands for Extensible Stylesheet Language Transformations and is used to define the transformation rules to be applied on the target XML document. XSLT Processor takes the XSLT stylesheet and applies the transformation rules on the target XML document and then it generates a formatted document in the form of XML, HTML, or text format. This formatted document is then utilized by XSLT formatter to generate the actual output which is to be displayed to the end-user.

We added the XSLT to the XML document as follows

<?xml-stylesheet type="text/xsl" href="Question5.xsl"?>

Output



# Question 6 and 9: XPath and XSLT

(Describe the major steps for designing the XPath and XSLT. How did you test the XPath? How did you use XPath in the XSLT?

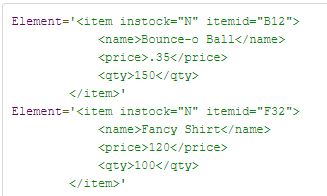
Add screenshot of the XPath testing and the output of XSLT)

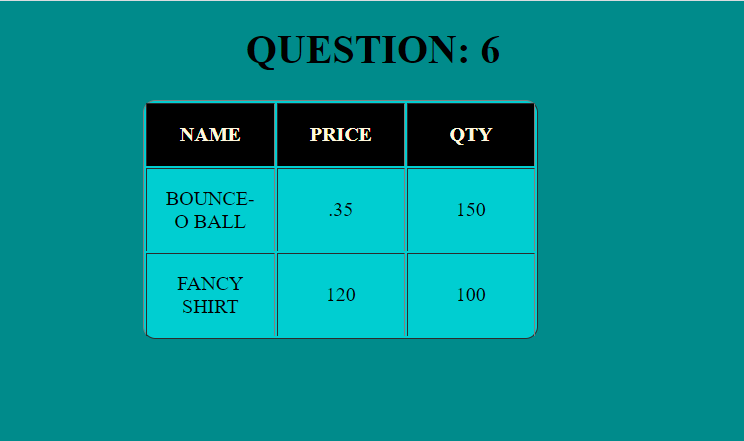
XPath uses path expressions to select nodes or node-sets in an XML document. The node is selected by following a path or steps.

We have used XPath Tester to test the XPath. For example to display all the order items which are not in stock ,following XPath expression is used in order.xml document.

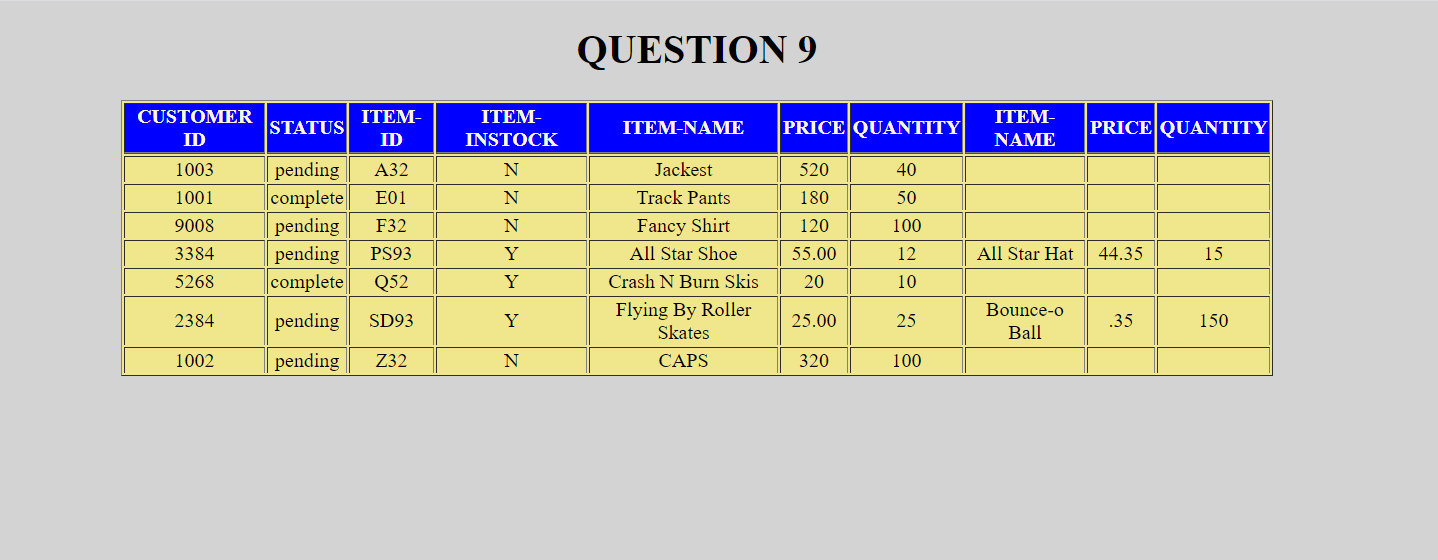
//orders/order/item[@instock='N']

Output for the XPath testing:



Output of Question 6 using XPath in it: 

Output of Question 9 using XPath in it:



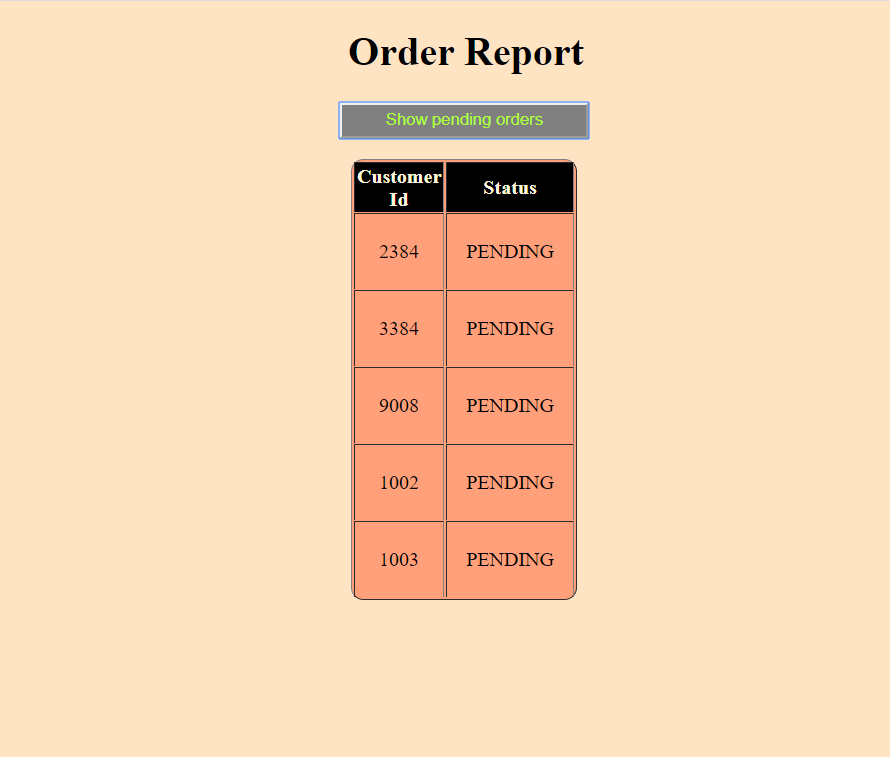
# Question 7 and 8: Use JavaScript to process XML data

(Describe the major steps for designing the JavaScript function(s), how you test this program, add some screenshots of the output )

JavaScript functions are used to add various types of functionality to the application to make it user interactive.

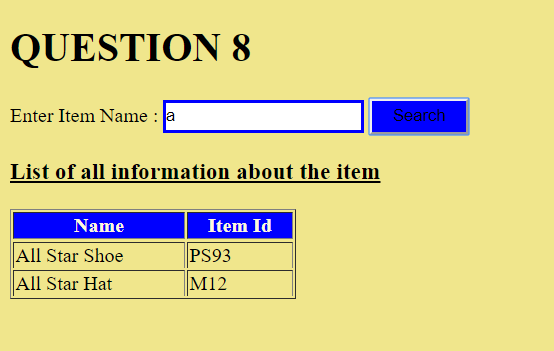
* A JavaScript function is defined with the function keyword, followed by a name, followed by parentheses ().
* Function names can contain letters, digits, underscores, and dollar signs.
* The parentheses may include parameter names separated by commas:
* The code to be executed, by the function, is placed inside curly brackets: {}

Question 7



Question 8

When user enters the name of the item or one or more characters and user clicks on the search button ,then it will search the xml file for corresponding data and displays the results. In a table format.



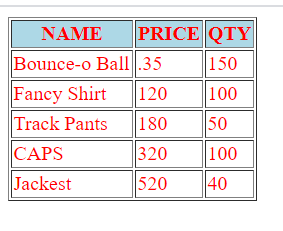
# Question 10

(Describe the major steps for designing the JavaScript function(s), how you test this program, add some screenshots of the output)

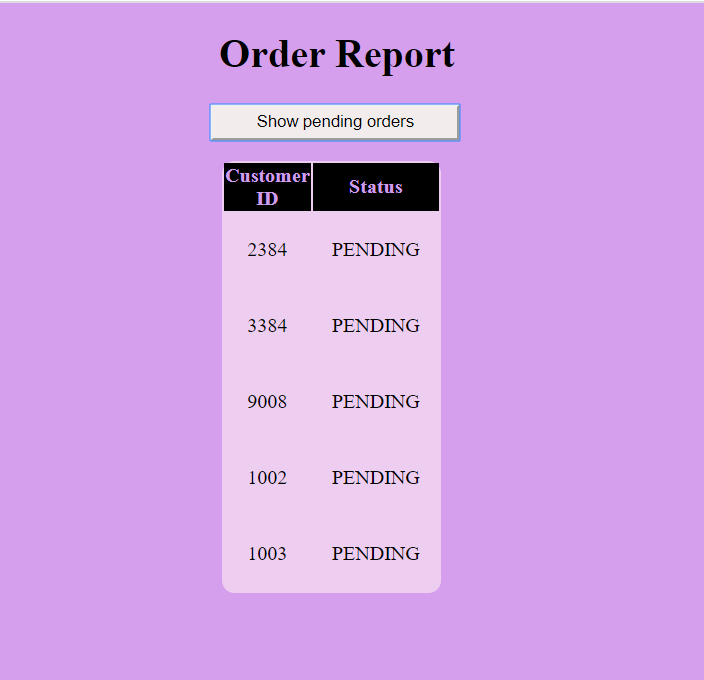
In Question7 ,customerid is used as element but in Modified version of Question7 ,it is used as an attribute in the order tag. The getAttribute method is used to get the customerid as attribute whereas in Question7 getElementsByTagName is used to get the customerid. It is easier with XSLT to process customerid as compare to JS because we just have to write XPATH expression to get it.

We provide all the functionality in the show\_orders() function. When user click on the button named show pending orders the function is called to display output as shown:

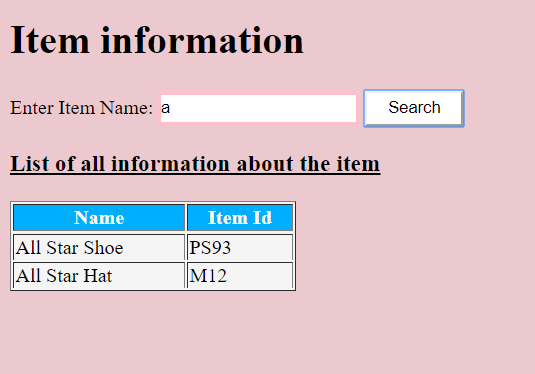
**Question 6 Modified**

****

**Question 7 Modified**



**Question 8 Modified**



As the above screenshot is similar to question8 having a search option .When the user enter a word which is present in the document it will display the result.

# Summary

(Describe how did you divide the work, share your feedback about this project like new points that you learn, challenges, …)

All the questions are equally shared among us and if there is some error we helped each other**.** **T**he last 10th question was qui**te** tricky and so we both did it together. Working on this project and stepping this to the final stage was super challenging job.

The most important was the managing things in the limited time and how to distribute the workload with each member,

Project made me realize how we can put the xml,xsl and javascript into much more practical way. Project helped us to explore new outcomes and what new things we can add using these languages. Problem was when fetching the data inside the certain tags , when there was sub tags but then we used certain method to overcome the problem.