|  |
| --- |
| Faculty of Applied Sciences and Technology |
| **XML Data Processing** |
| ITC5202 - Project |
|  |
| **Leontii Polovinko, Gregorit Tyapkin** |
| **3/6/21** |

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

Table of Contents

[Question 1 : 2](#_Toc65610936)

[Question 2 and 3 : XML Structure 3](#_Toc65610937)

[Question 4 : Design XSLT 4](#_Toc65610938)

[Question 5 and 8: XPath and XSLT 5](#_Toc65610939)

[Question 6 and 7: Use JavaScript to process XML data 6](#_Toc65610940)

[Question 9 7](#_Toc65610941)

[Bonus question 8](#_Toc65610942)

[Summary 9](#_Toc65610943)

# Question 1 :

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

The product.xml document was not both well-formed and valid. It contained two id attributes which were not wrapped in quotes.





Now it is well-formed, because it follows all rules to be well-formed: has one root element, all opened tags have close tags, tags are written correctly, all attributes are wrapped in double quotes. Also, after fixing omitting of double quotes it became valid and validator approved it.

Изображение выглядит как текст

Автоматически созданное описание

# Question 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.

1) To create DTD, I have created stand-alone dtd file called question2.dtd because I am used to dividing projects into separates files which makes project more readable and easier to understand. To create proper dtd file I have properly written the xml document tags hierarchy including all tags and attributes. Then I have linked this file in the question2.xml document via !DOCTYPE tag.

2) To create schema, I have used freeformatter. For this, I just copied the xml file and put it into XSD generator which generated the schema for me. Then I put generated code into a question3.xsd file. After that I have linked question3.xsd in the root element of question3.xml file.

3) For validation I have used xml validator. To validate DTD file, I have firstly uploaded question2.xml file and then uploaded question2.dtd file.

Validator output:

Изображение выглядит как текст

Автоматически созданное описание

To validate schema, I have firstly uploaded question3.xml file and then uploaded question3.xsd file.

Validator output:

Изображение выглядит как текст

Автоматически созданное описание

4)

# Question 4 : Design XSLT

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

To create XSLT, I have created called question4.xslt. Then I created table element to display a proper information from xml document. Via foreach loop I went through all product elements and put appropriate information into td tags of a table. To get a value of an element I just used simple XPath syntaxis (put a path of an element into the select attribute of xsl:value-of element). To get values of attributes I have used “@” sign and attribute name. This allowed me to display an information about all products row by row in a table format.

Output:

Изображение выглядит как стол

Автоматически созданное описание

# Question 5 and 8: 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)

To design XPath expression, I have used XPath tester on freeformatter website. Then, I wrote the path to all product\_details elements of product elements which have instock attribute equal to “N”. To get the root element I wrote double slashes. To get products which are not in stock I wrote product[@instock=’N’] which allows to select all element which have attribute “instock” equal to “N”.

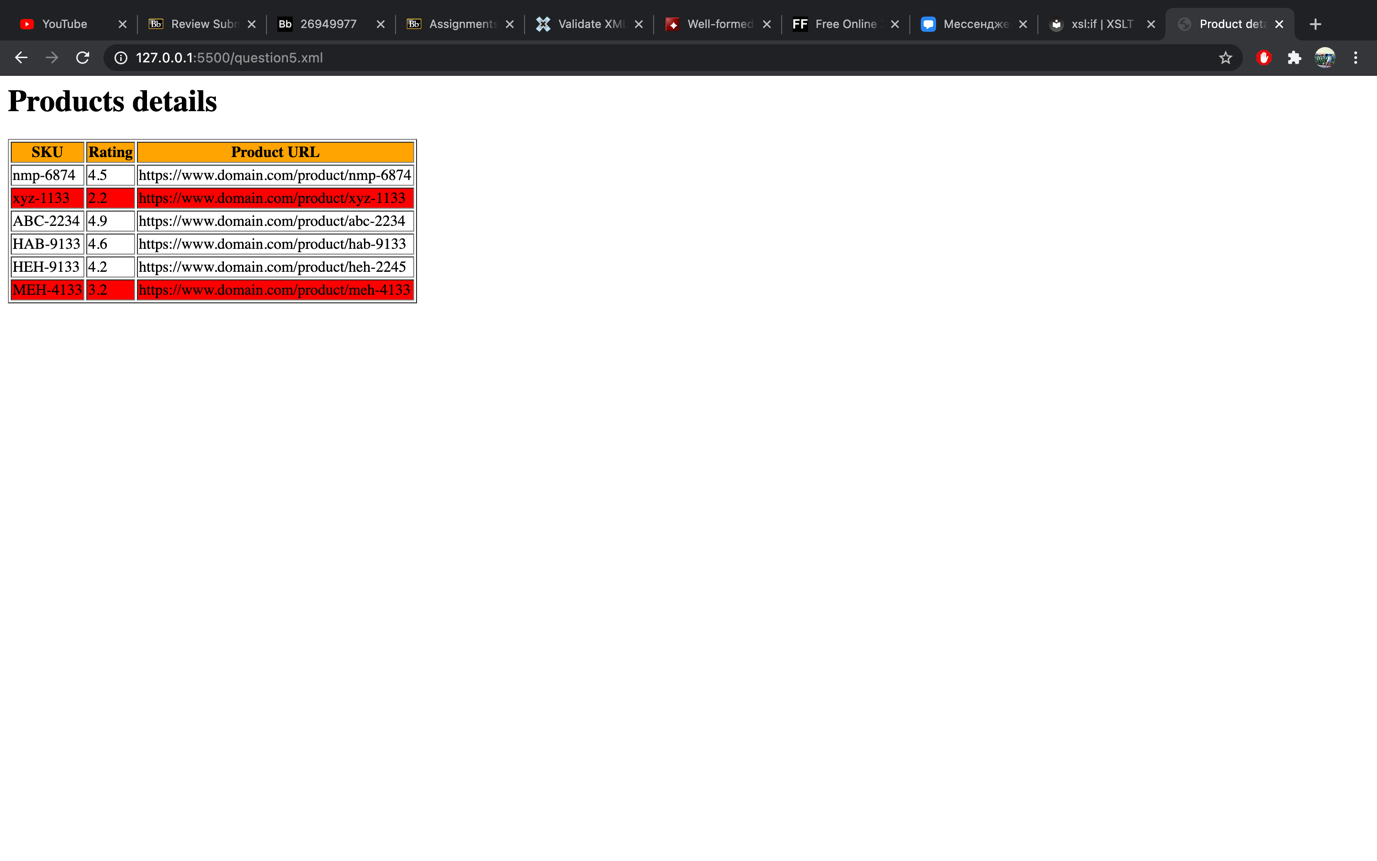
The XPath expression and the result:

Изображение выглядит как текст

Автоматически созданное описание

To display product\_detail elements in a browser, I created xsl document called question5.xsl and linked it in question5.xml document. In question5.xsl file I have used table element to display an information in a table format. Then, I used foreach loop to go through all product elements. Within the loop I have used xsl:if tag to check the “instock” attribute value of a product element. If it was equal to ‘N’ I colored the row of a table in red and display an information. If it was not equal to ‘N’ I just display the information.

Output:

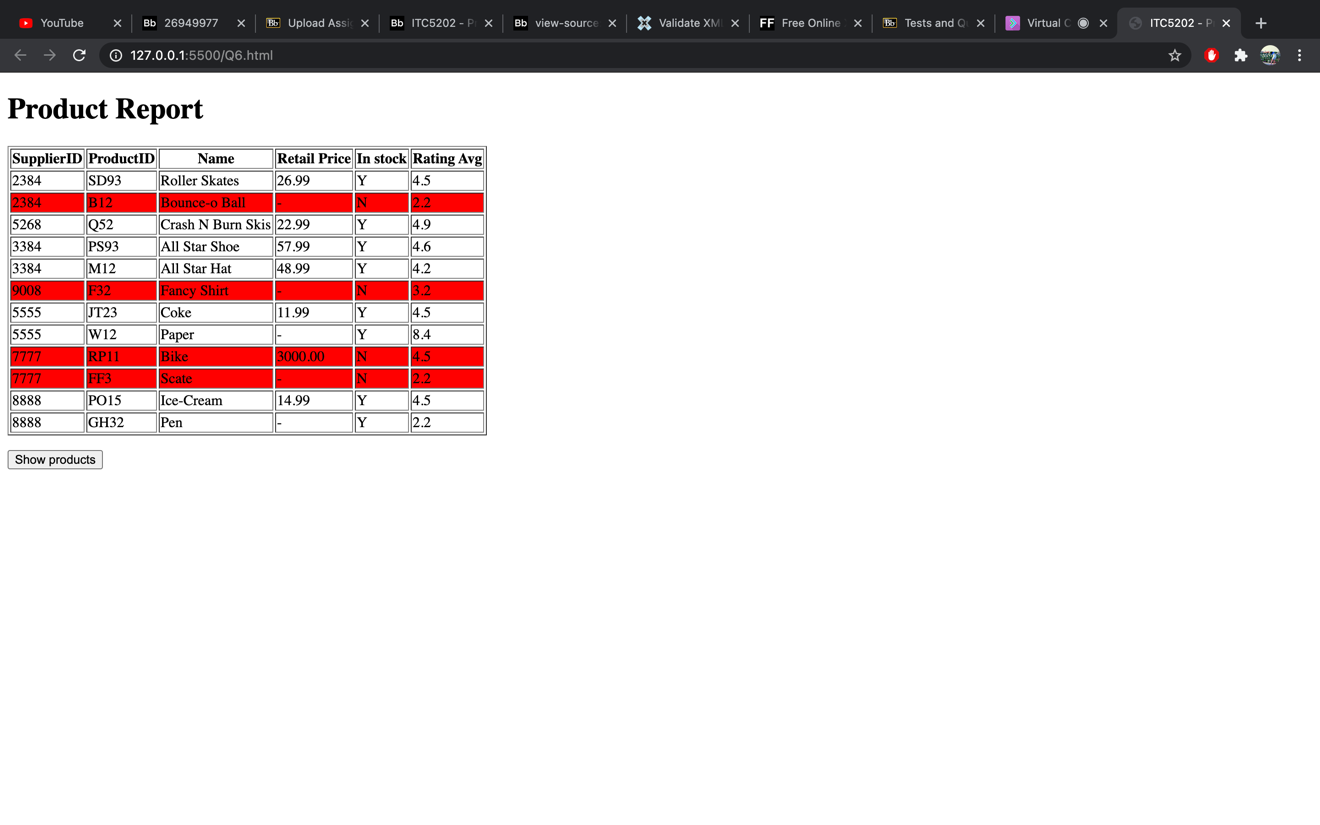


# Question 6 and 7: Use JavaScript to process XML data

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

To display all products in question 6 I have created question6.js file. Also, I copied information from product.xml to question6.xml and added 3 suppliers with 2 products each. To display information in a table format I have created the variable called output and put appropriate information into it. Then I put all products into “products” variable and using for loop I went through all products. If product has attribute “instock” equals to ‘N’ it has red background color which is set by “style” attribute it <td> tag. Also, going through all products I got all needed information via getAttribute and getElementsByTagName methods. Some products do not contain Retail\_price and Rating\_avg elements, therefore I checked if getElementsByTagName with appropriate argument returns something or not, and if it does not return anything, I put “-” in the field of the table.

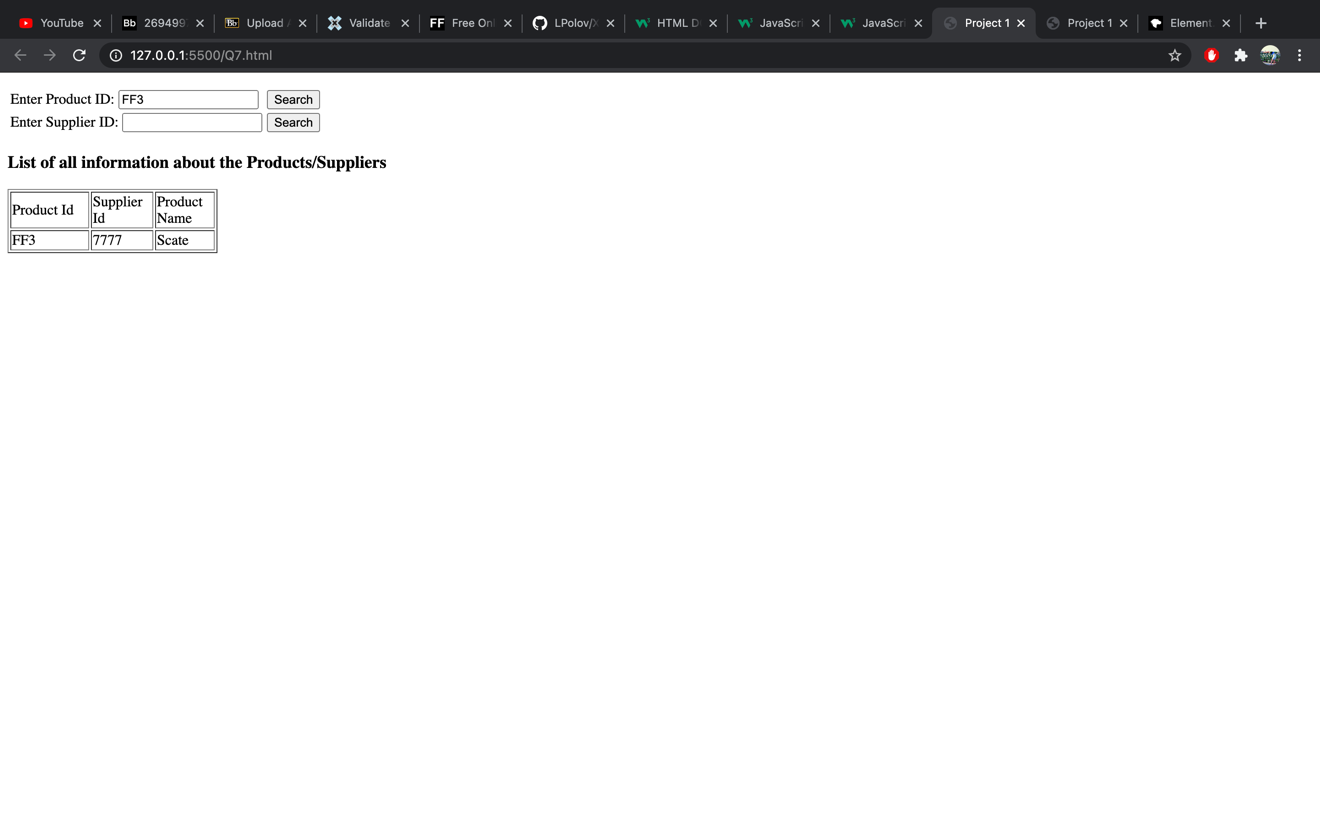
Output:

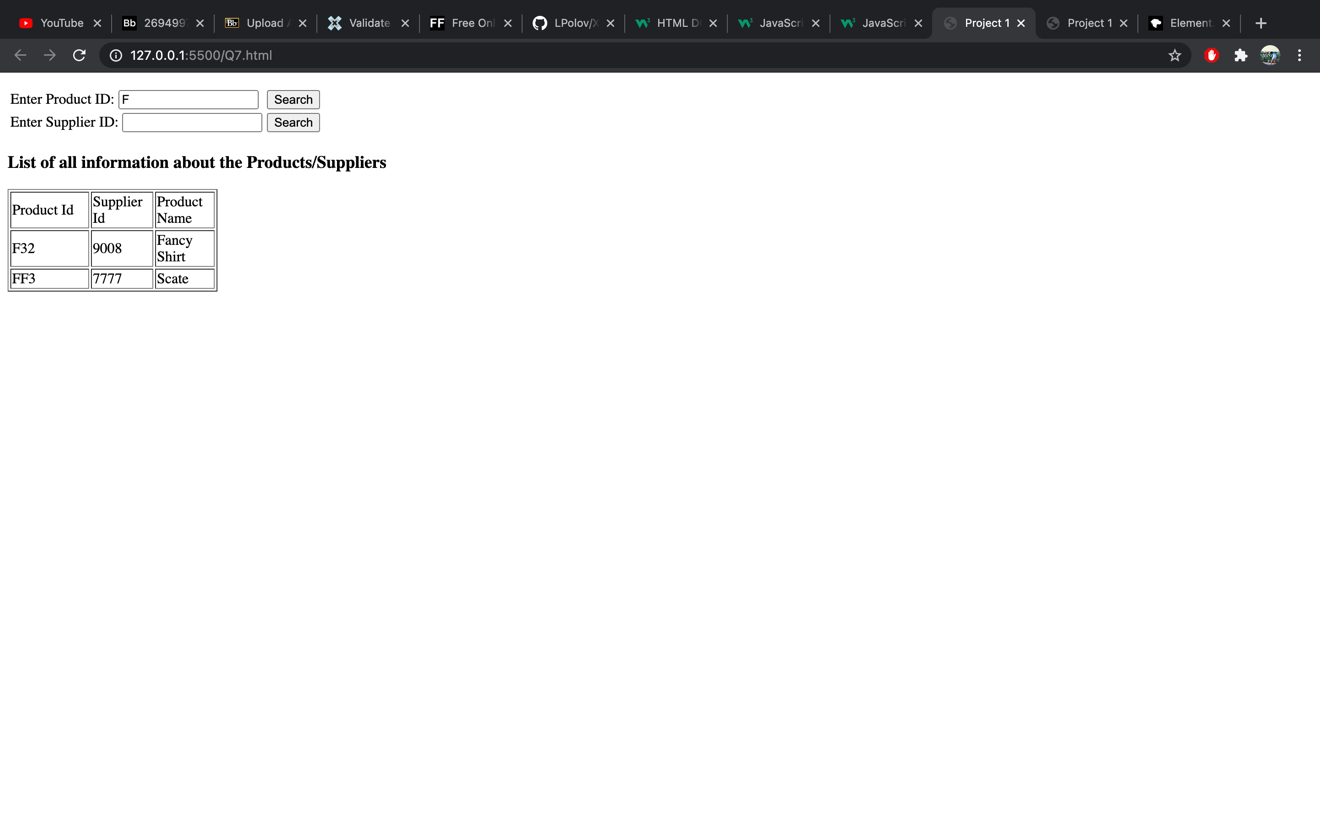


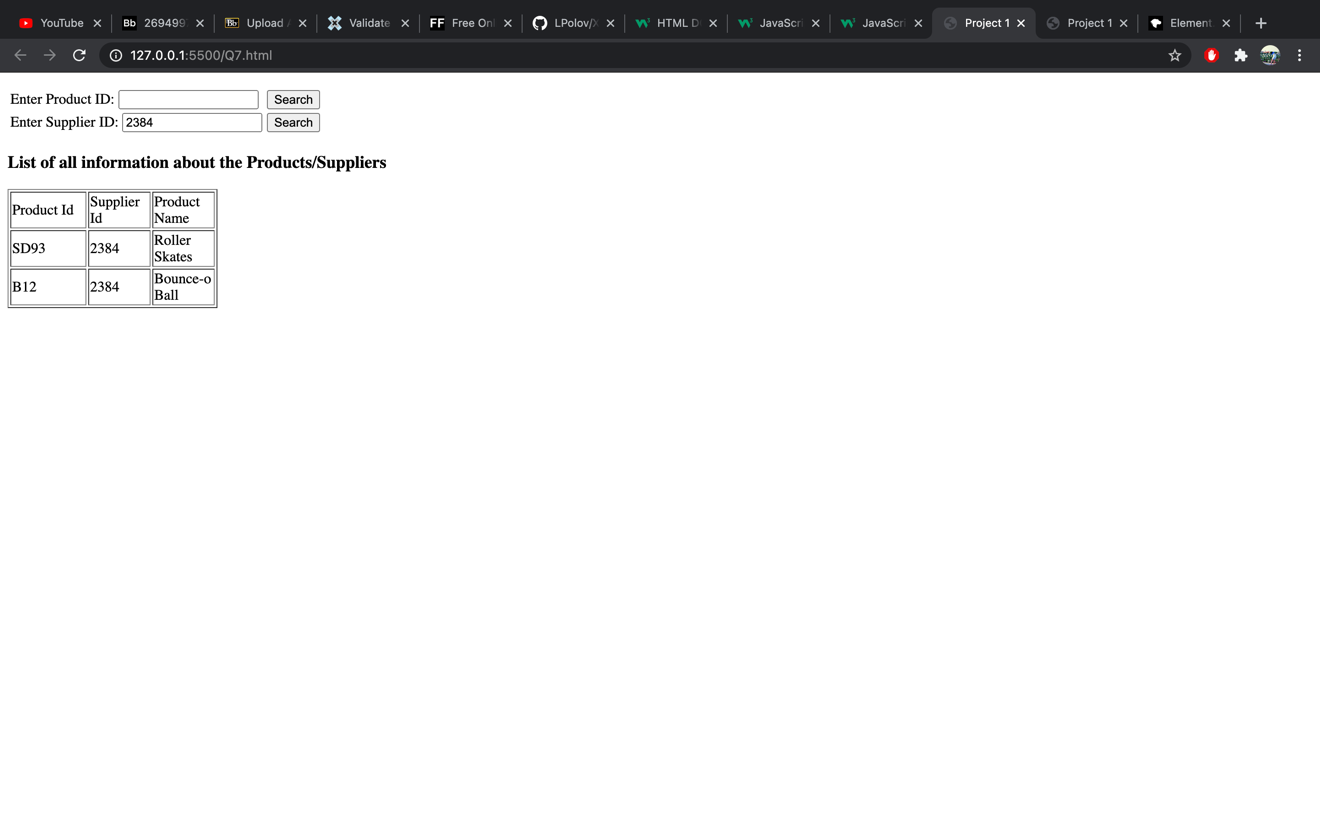
To implement searching in question 7 I have created a file question7.js which contains two functions. First is called searchProductID, this function gets text entered by the user into “Enter Product ID” field. Then, it gets all supplier elements and goes through them using for loop. Within this loop it gets all products of each supplier and goes through them using nested for loop. Then, it checks if current product id is equal to entered product id and if it is it take an appropriate information and puts it into output variable. Includes() function allows to implement partial search because it checks if this substring is contained in the productid attribute. If user does not enter any information and clicks on “Search” button it will display an information about all products due to include() function. When all loops are finished, output variable is set into innerHTML of a table which is used to display data.

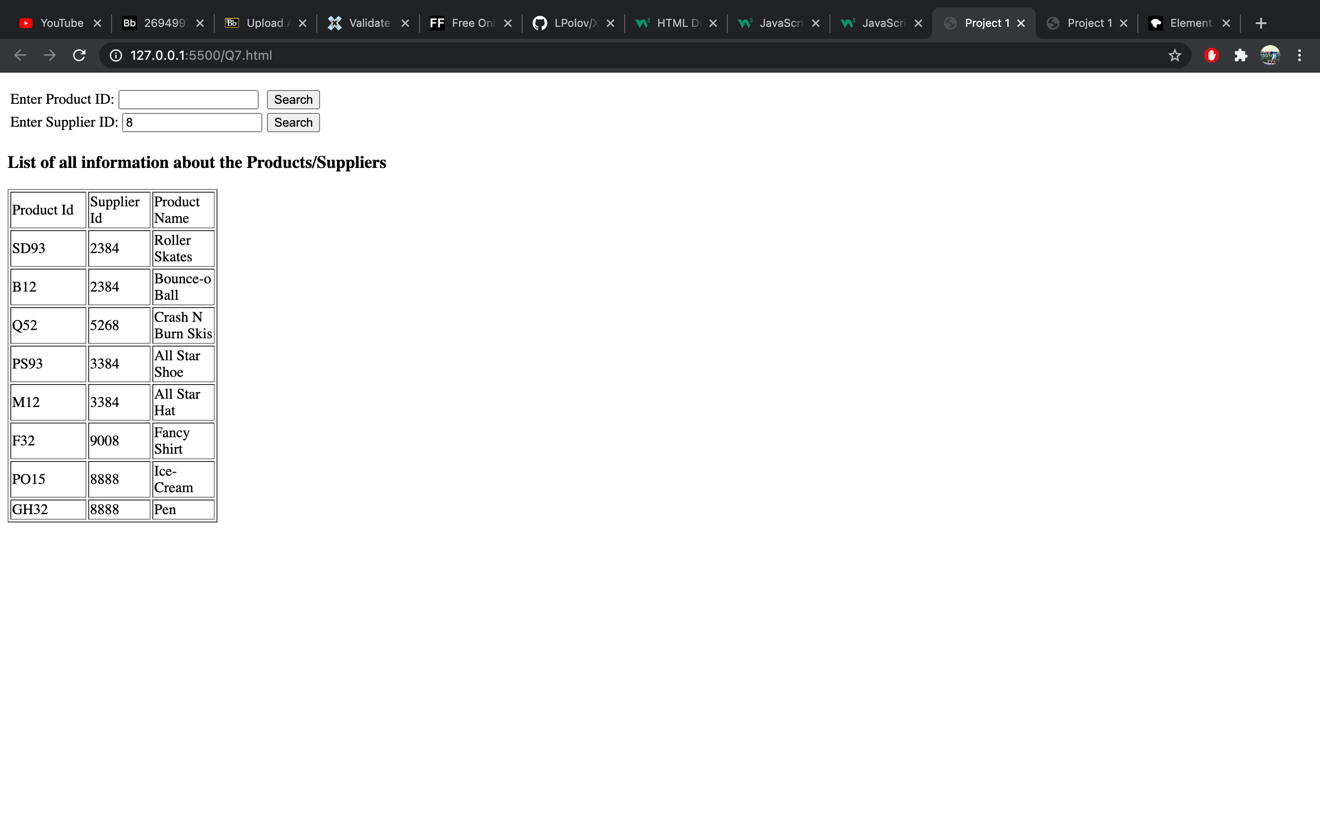
To implement searching by supplier id I have created the second function which works following the same principle. But in this case, it checks supplier id instead of product id.

Output:









# Question 9

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

# Bonus question

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

# Summary

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