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

\* ITC5202 – Project1

\* I declare that this assignment is my 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.

\*

\* Name: Giovanny Avillaneda Student ID: N01604159  Date: Feb-26-2024

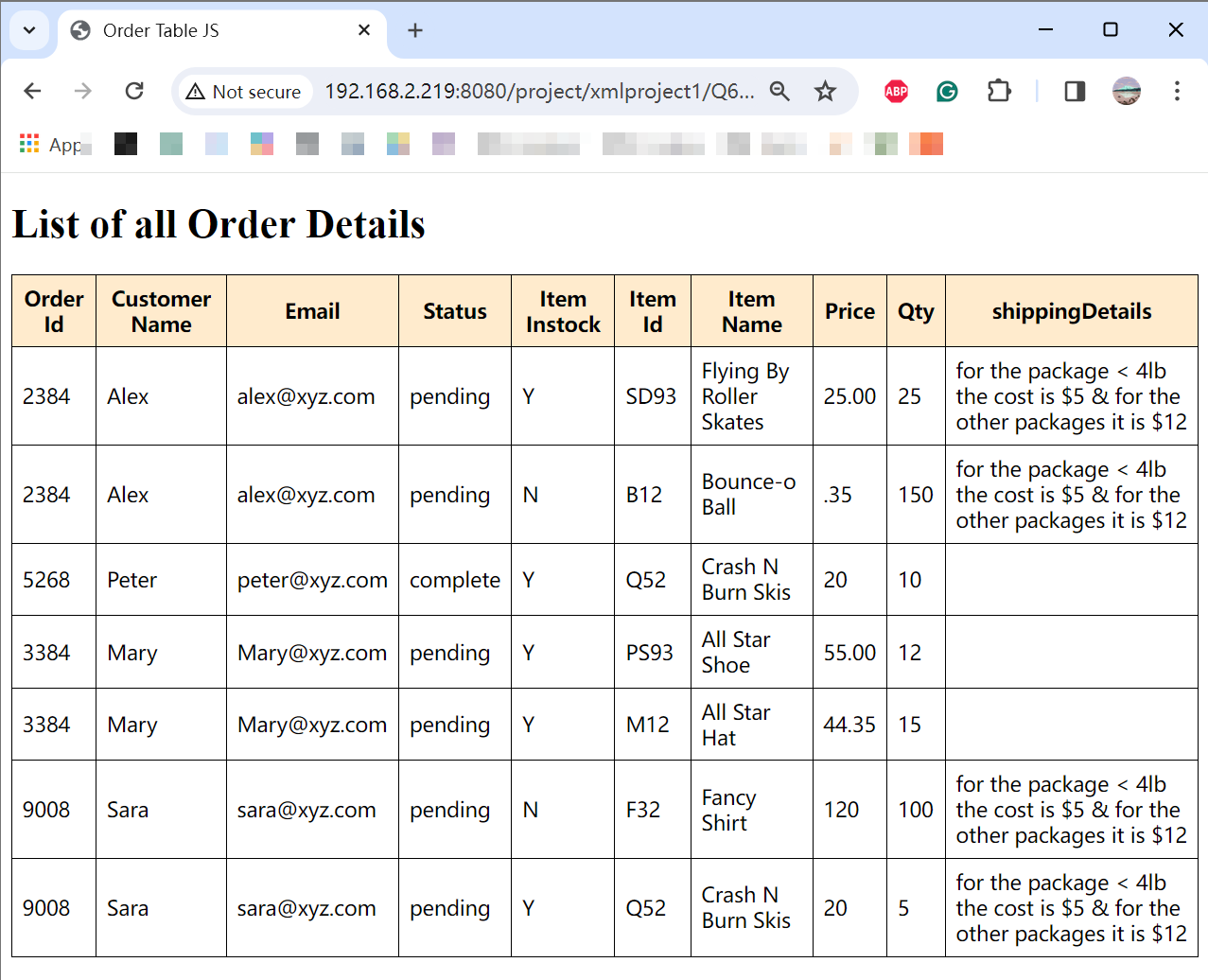
\* Name: Kexin Zhu           Student ID: N01621302  Date: Feb-26-2024

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

Q5:

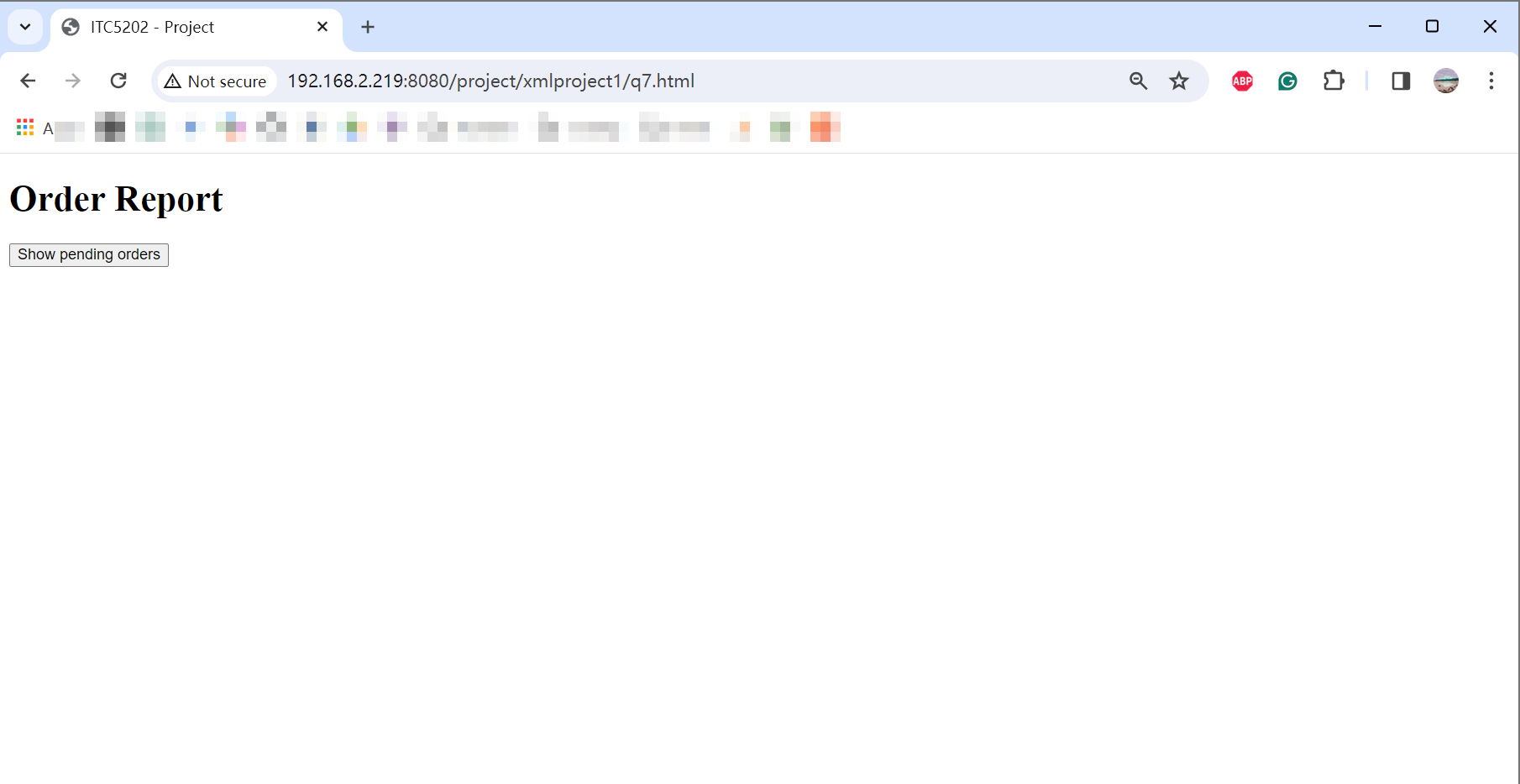
Q6:

Output

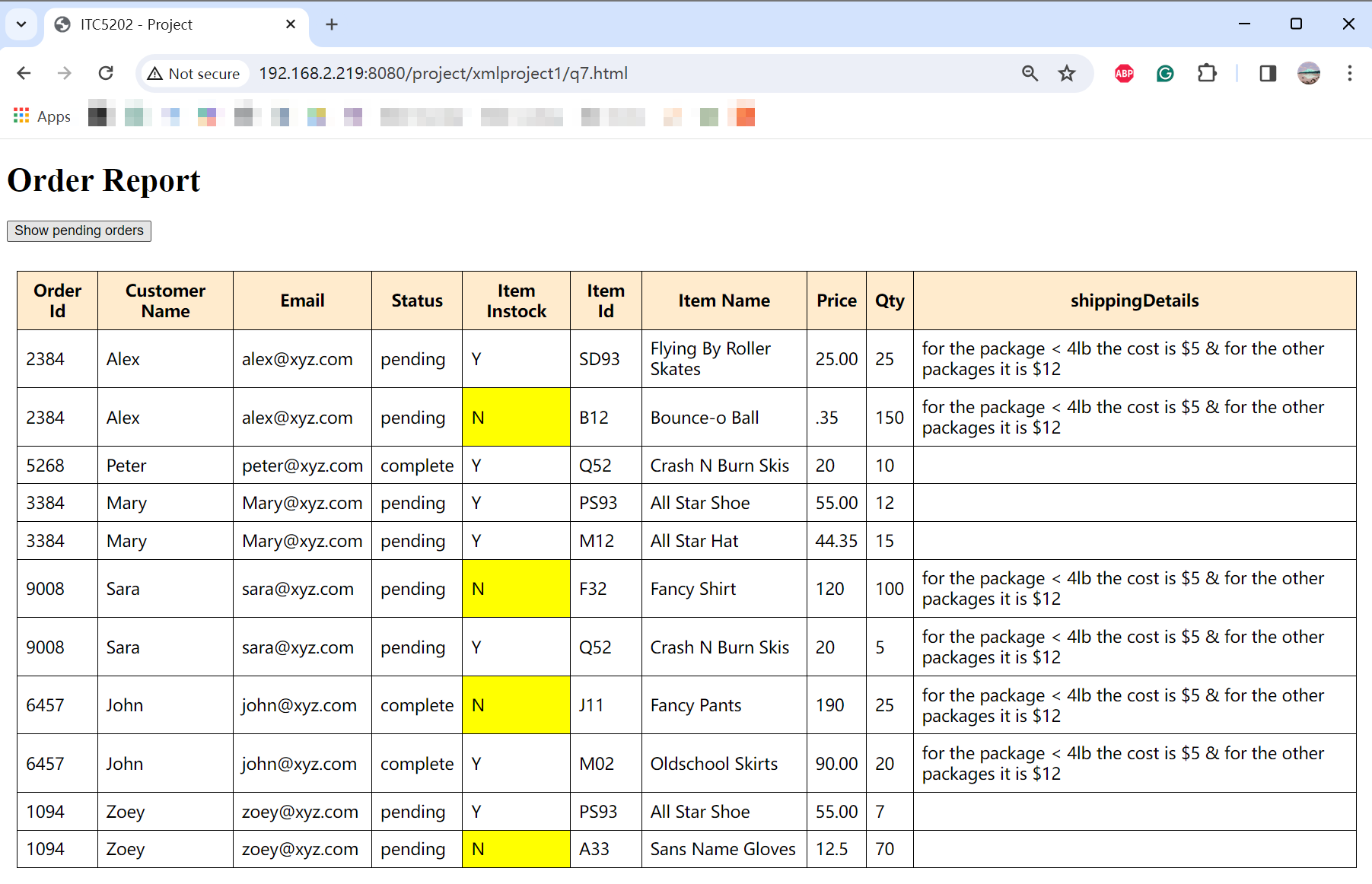


Q7:

Output



When click the button, highlighting the “N” value of “Item Instock”

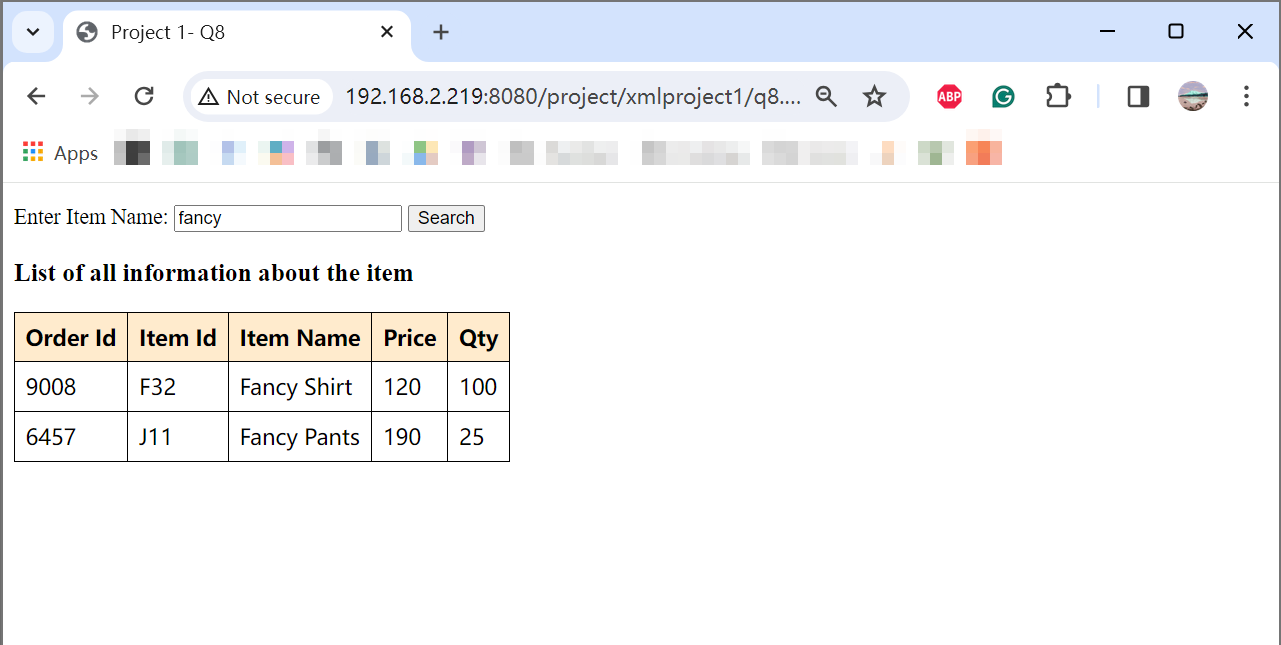


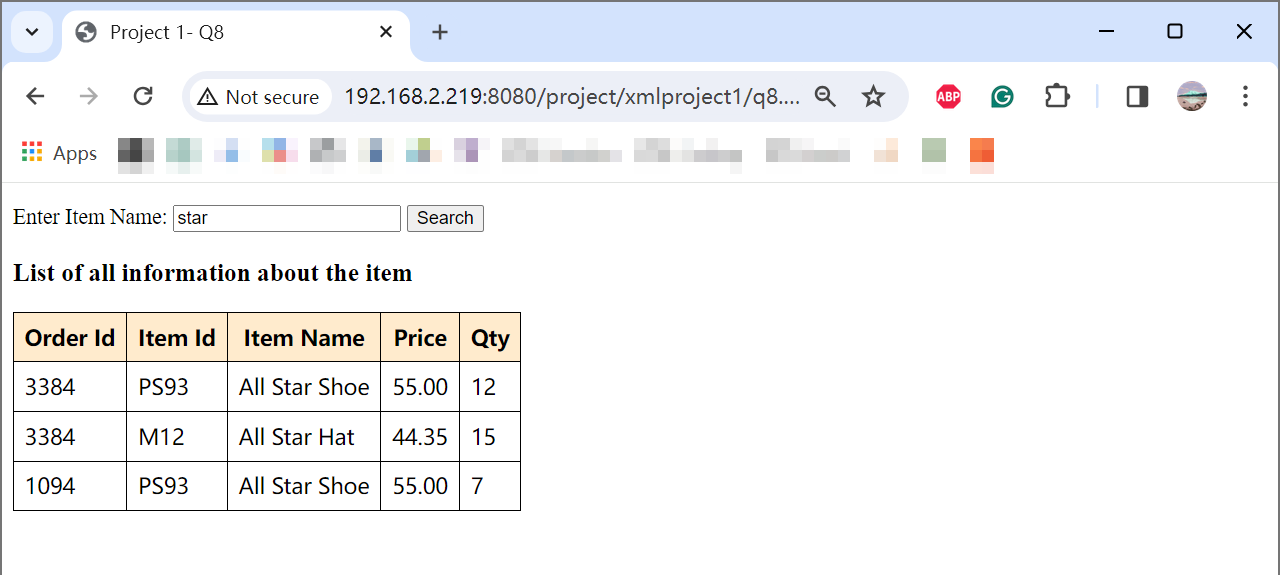
Q8:

Output

I chose 4 fields to display: item id, item name, price, and qty, as well as “oid” as recommended.

The search is partial as well as case insensitive.





The oid is added to the output by the line below, since it’s an attribute of order (item’s parent node), it needs the getAttribute() statement to get the value.

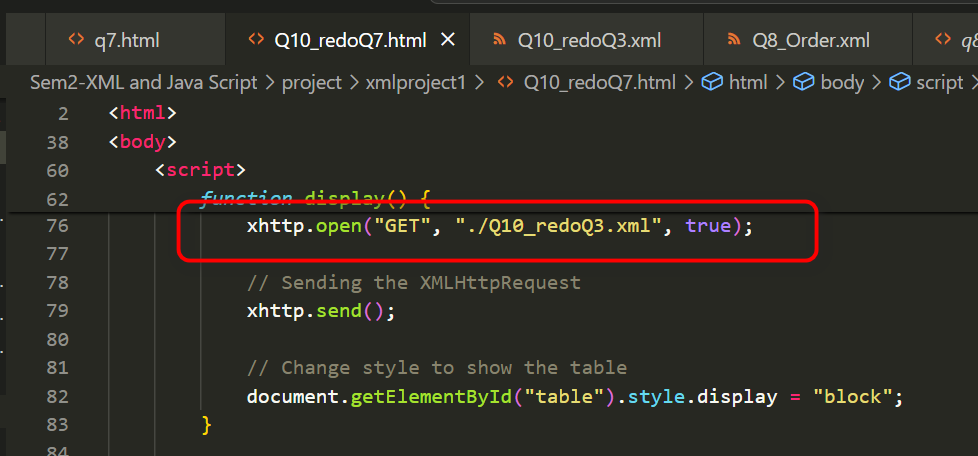
*const* oid = *item*.parentNode.getAttribute("oid");

Q9:

Q10:

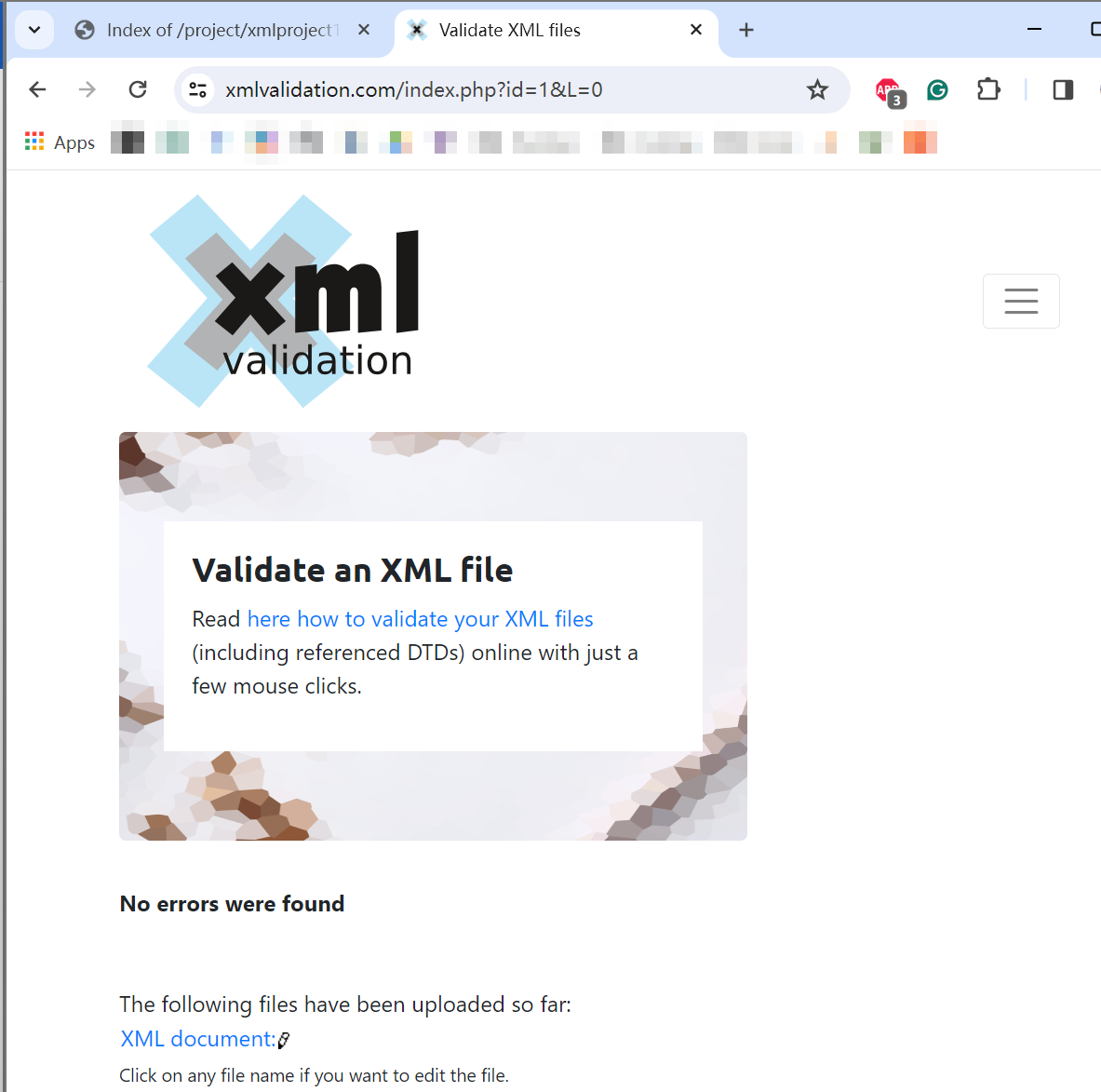
**Note:**

Since Q3 doesn’t link to a schema file, I didn’t design schema for redo Q3 as well. And since the redo Q3 XML file contains DTD, I used this file as the XML for redo Q7 and Q8.



**Redo Q3:**

Validated. It’s well-formed and valid.

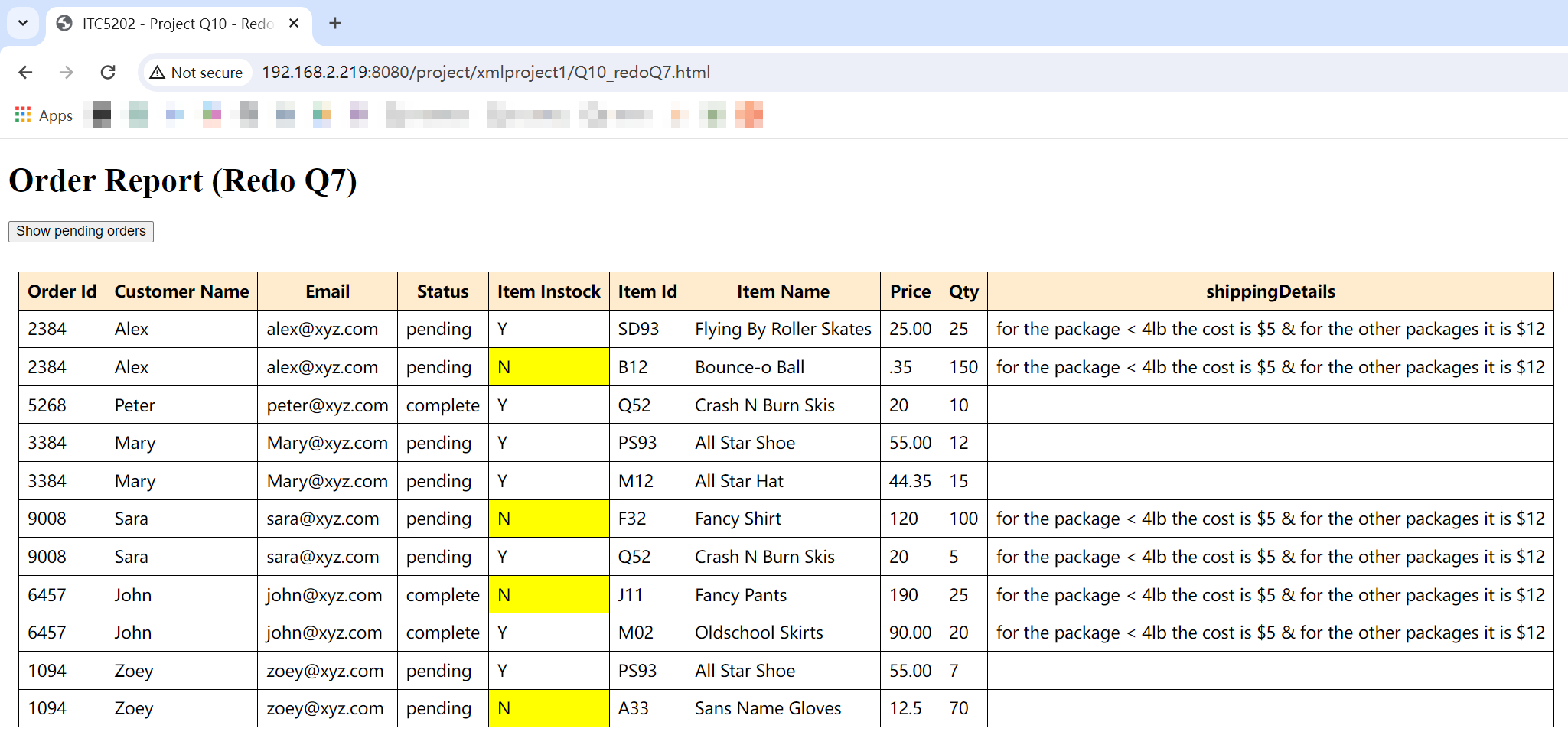


**Redo Q7:**

Modified the get status expression in the parseItem function:

*const* status = *item*.parentNode.getAttribute("status");

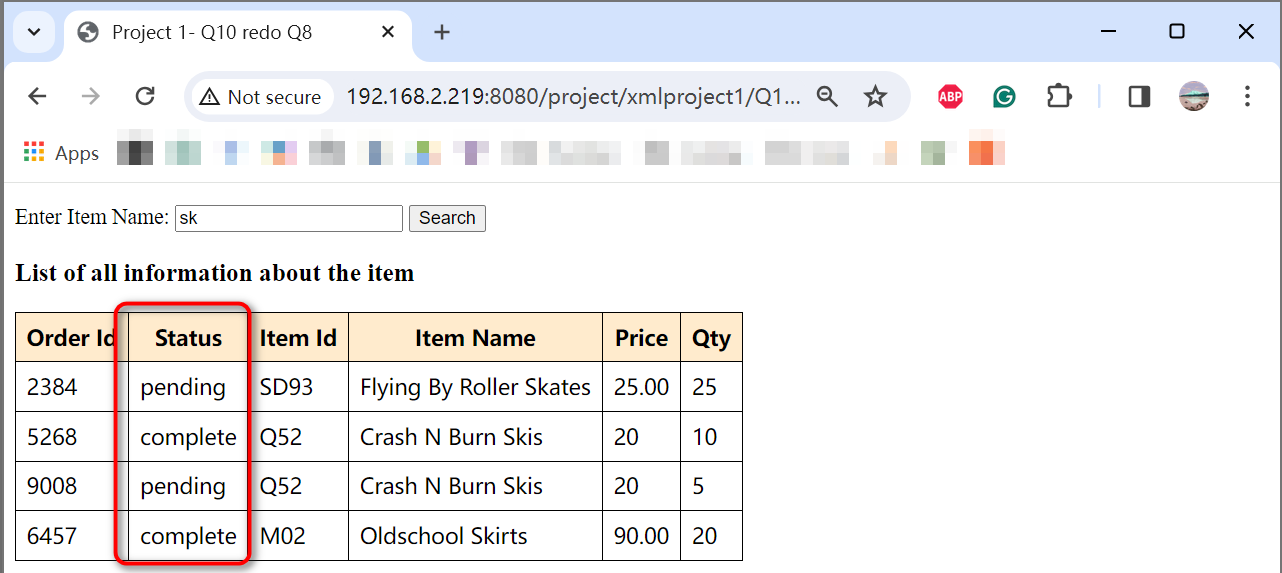
Output:



**Redo Q8:**

Note: based on the output of the original Q8, the modification of <status> element could not affect the output because Q8 is not related to “status”. To indicate the change, another column is added in the output in the redo Q8 and fetched the value of the status attribute.

Output



**From your understanding as developer, explain your idea about coding/processing XML**

**element vs attribute.**

Element: Elements represent hierarchical data structures in the XML document, making it easier to understand and navigate. An element can be simple or complex, depending on whether it contains only text or not.

Attribute: Attribute is additional values of elements, an element could have many attributes. Attributes are not suitable for structures but are useful for marking elements with identifiers.

**Which one is easier to develop with JS or XSLT, process *status* as**

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

Developing with JS to process an element or attribute is much more complex than that with XSLT. In this project, to get the *status* value with JS needs to declare the constant, and then convert it to a DOM node, and then pass it to a row element of the table. While only a <value-of> expression in XSLT.

Besides, processing attribute data is easier than processing element data. Since when the *status* is an attribute, it doesn’t need to worry about the node structure but only has to use a getAttribute() method in JS or @ mark in XSLT. When it’s an element, it’s necessary to apply the correct path or node in my code to get the proper value.