ShopSmart

Summary: First, an XSLT stylesheet ("transform.xsl") was developed to dictate how XML information gets converted into HTML format.An XML Schema Definition file ‘product_schema.xsd’ was later created to define permissible elements, attributes, data types, and conditions for the product data.A Python script was built to automate the sequence of transformation and validation using the ’lxml’ library. The converted HTML content is stored as an "output.html" file. The validation procedure confirms whether the XML file corresponds to the structure and constraints defined in the schema.

1.products.xml : This xml file contains tags that define how the e-commerce web page is to be structured and the actual product data is stored in the file as well.
2.product_schema.xsd : It defines validation rules for products.xml file and explains the xml form
3.transform.xsl : Stylesheet used to transform products.xml into output.html and to visually format the output
4.transform.py : This python script applies transform.xsl stylesheet to the data from products.xml to generate output.html file as output.
5.validate.py : This python script cross-checks whether the xml file adheres to the schema definition and restrictions defined in product_schema.xsd
6.output.html : This is the html file generated from transform.py script

Validation Test :

1. Price had been wrongly assigned a constraint as ‘xs:string’ for the purpose of displaying the dollar symbol along with the price and also had the following pattern restriction for the string : '\$[0-9]+'

Received the following errors from validation.py:

```
     XML is not valid according to the XSD.
     Error: Element 'price': '$499' is not a valid value of the atomic
type 'priceType'.
     Error: Element 'price': '$449' is not a valid value of the atomic
type 'priceType'.
     Error: Element 'price': '$420' is not a valid value of the atomic
type 'priceType'.
     Error: Element 'price': '$475' is not a valid value of the atomic
type 'priceType'.
     Error: Element 'price': '$450' is not a valid value of the atomic
type 'priceType'.
```

```
Error: Element 'price': [facet 'pattern'] The value '499' is not accepted
by the pattern '\$[0-9]+'.
Error: Element 'price': [facet 'pattern'] The value '449' is not accepted
by the pattern '\$[0-9]+'.
```

```
Error: Element 'price': [facet 'pattern'] The value '420' is not accepted
by the pattern '\$[0-9]+'.
Error: Element 'price': [facet 'pattern'] The value '475' is not accepted
by the pattern '\$[0-9]+'.
Error: Element 'price': [facet 'pattern'] The value '450' is not accepted
by the pattern '\$[0-9]+'.
```

Rectified the error changing the restriction from xs:string to xs:decimal and removing the pattern constraint

2. Triggered a missing ‘required’ attribute error

Received the following error from validation.py:

```
XML is not valid according to the XSD.
Error: Element 'product': The attribute 'id' is required but missing.
```

3.Triggered a mismatch between the actual attributes of the element and the defined attributes of the element in the schema definition

Received the following error from validation.py:

```
Error: Element 'price': This element is not expected. Expected is ( name
).
```

For Error Checking :

```
for error in xsd_schema.error_log:
        print("Error:", error.message)
        if "is not a valid value" in error.message:
            print("Element structure is not valid:", error.message)
            print("Refer XSD and check whether the element definition
corresponds to the element in XML")
        elif "is required but missing" in error.message:
            print("Required element is missing:", error.message)
            print("Check if all the elements with required constraint are
present")
        elif "This element is not expected" in error.message:
            print("Invalid attribute:", error.message)
            print("Check if the attribute matches the allowed attributes
defined in the XSD.")
```