Summary report

During Validation

```xml
SmartShop.xsd > xs:schema
1    <?xml version="1.0" encoding="utf-8"?>
2
3    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
4        <xs:element name="products">
5            <xs:complexType>
6                <xs:sequence>
7                    <xs:element name="product" maxOccurs="unbounded">
8                        <xs:complexType>
9                            <xs:sequence>
10                               <xs:element name="name" type="stringType" />
11                               <xs:element name="price" type="priceType" />
12                               <xs:element name="description" type="xs:string" />
13                           </xs:sequence>
14                           <xs:attribute name="id" type="xs:string" use="required" />
15                       </xs:complexType>
16                   </xs:element>
17               </xs:sequence>
18           </xs:complexType>
19       </xs:element>
20
21       <xs:simpleType name="priceType">
22           <xs:restriction base="xs:decimal">
23               <xs:minInclusive value="1" />
24               <xs:maxInclusive value="99999999999" />
25           </xs:restriction>
26       </xs:simpleType>
27
28       <xs:simpleType name="stringType">
29           <xs:restriction base="xs:string">
30               <xs:minLength value="1"></xs:minLength>
31           </xs:restriction>
32       </xs:simpleType>
33   </xs:schema>
```

```xml
SmartShop.xml > products
1   <?xml version="1.0" encoding="UTF-8"?>
2   <?xml-model href="SmartShop.xsd" type="application/xml" schematypens="http://www.w3.org/2001/XMLSchema"?>
3   <products>
4       <product id="P1">
5           <name>Electronics</name>
6           <price>99990</price>
7           <description>Samsung Galaxy Z-Fold</description>
8       </product>
9       <product id="P2">
10          <name>Book</name>
11          <price>30</price>
12          <description>To Kill a Mockingbird</description>
13      </product>
14      <product id="P3">
15          <name>Clothing</name>
16          <price>150</price>
17          <description>Blazer - Black</description>
18      </product>
19      <product id="P4">
20          <name>Clothing</name>
21          <price>50</price>
22          <description>Chickankari Kurti - Blue</description>
23      </product>
24      <product id="P4">
25          <name></name>
26          <price>50</price>
27          <description>Chickankari Kurti - Blue</description>
28      </product>
29  </products>
```

Error encountered:

```
XML file is not valid against the XSD schema.
Traceback (most recent call last):
  File "c:/Users/Pearl/Desktop/HTML/XML, XSD and XLS/validator.py", line 10, in <module>
    print(validator.validate(xml_file))
  File "C:\Users\Pearl\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.7_qbz5n2kfra8p0\LocalCache\local-packages\Python37\site-packages\xmlschema\validators\schemas.py", line
1719, in validate
    raise error
  File "C:\Users\Pearl\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.7_qbz5n2kfra8p0\LocalCache\local-packages\Python37\site-packages\xmlschema\validators\simple_types.py",
line 1406, in iter_decode
    validator(result)
  File "C:\Users\Pearl\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.7_qbz5n2kfra8p0\LocalCache\local-packages\Python37\site-packages\xmlschema\validators\facets.py", line 6
2, in __call__
    self._validator(value)
  File "C:\Users\Pearl\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.7_qbz5n2kfra8p0\LocalCache\local-packages\Python37\site-packages\xmlschema\validators\facets.py", line 2
21, in _min_length_validator
    raise XMLSchemaValidationError(self, value, reason)
xmlschema.validators.exceptions.XMLSchemaValidationError: failed validating '' with XsdMinLengthFacet(value=1, fixed=False):

Reason: value length cannot be lesser than 1

Schema:

  <xs:minLength xmlns:xs="http://www.w3.org/2001/XMLSchema" value="1" />

Instance:

  <name />

Path: /products/product[5]/name
```

In above case,

The XML file was not validated because of the restriction in the schema file for minimum length = 1.

During Transformation

```xml
SmartShop.xml > products
1   <?xml version="1.0" encoding="UTF-8"?>
2   <?xml-model href="SmartShop.xsd" type="application/xml" schematypens="http://www.w3.org/2001/XMLSchema"?>
3   <products>
4       <product id="P1">
5           <name>Electronics</name>
6           <price>99990</price>
7           <description>Samsung Galaxy Z-Fold</description>
8       </product>
9       <product id="P2">
10          <name>Book</name>
11          <price>30</price>
12          <description>To Kill a Mockingbird</description>
13      </product>
14      <product id="P3">
15          <name>Clothing</name>
16          <price>150</price>
17          <description>Blazer - Black</description>
18      </product>
19      <product id="P4">
20          <name>Clothing</name>
21          <price>50</price>
22          <description>Chickankari Kurti - Blue</description>
23      </product>
24      <product id="P4">
25          <name></name>
26          <price>50</price>
27          <description>Chickankari Kurti - Blue</description>
28      </product>
29  </products>
```

```
SmartShop.xsl > ☒ xsl:stylesheet
3    <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
4        <xsl:template match="/products">
5            <html>
6                <body>
7                    <h1>Shop Smart</h1>
8                    <h4>Welcome to shop, here are the products available</h4>
9                    <table border="1">
10                       <tr>
11                           <th>Name</th>
12                           <th>Price</th>
13                           <th>Description</th>
14                       </tr>
15                       <xsl:for-each select="product">
16                           <tr>
17                               <td>
18                                   <xsl:value-of select="namo" />
19                               </td>
20                               <td>
21                                   <xsl:value-of select="price" />
22                               </td>
23                               <td>
24                                   <xsl:value-of select="description" />
25                               </td>
26                           </tr>
27                       </xsl:for-each>
28                   </table>
29               </body>
30           </html>
31       </xsl:template>
32   </xsl:stylesheet>
```

Shop Smart

Welcome to shop, here are the products available

Name	Price	Description
	99990	Samsung Galaxy Z-Fold
	30	To Kill a Mockingbird
	150	Blazer - Black
	50	Chickankari Kurti - Blue
	50	Chickankari Kurti - Blue

In the above case,

The value, name was mislabeled as 'namo'. Because of this, the output does not give any output in that section.

Purpose of the XSL stylesheet.

1. XSL is a language for expressing style sheets that describes how to display an XML document of a given type.
2. An XSL style sheet is a file that describes the way to display an XML document.
3. XSL shares the functionality and is compatible with CSS2, although it uses a different syntax.
4. It also adds a transformation language for XML documents called XSLT.

Purpose of the XSD schema.

1. An XML Schema is used to describe the structure of an XML document.
2. It defines the rules for all the attributes and elements in an XML document. The purpose of an XML Schema is to define the legal building blocks of an XML document: the elements and attributes that can appear in a document, the number of (and order of) child elements, data types for elements and attributes, default and fixed values for elements and attributes.
3. It is used to verify each piece of item content in a document, to assure it adheres to the description of the element it is placed in.

Purpose of the script or program you used for transformation and validation.

- XML transformation scripts are used to convert an XML document from one format to another.
- The purpose of these scripts is to transform the XML document into a format that can be used by other applications.
- The transformation process can include mapping from one XML schema to another or mapping from XML to some other format such as HTML or WML.
- XML validation scripts are used to validate an XML document against an XML schema.
- Besides needing to know whether a particular XML document is well-formed, you often need to know whether it conforms to a given XML schema, that is, whether it is valid with respect to that XML schema.