WSD Lab 05

XML Styling and Validating Documentation

Domain:

Restaurant Website

XML content:

The xml file contains data related to dishes sold by restaurant, and each dish is part of the menu of the restaurant. The restaurant contains data like id, name, hours and reviews as well. Dish contains the name of the dish, description and price.

XSD validation:

An XML schema file has been created to validate the data entry. With the help of this file, the name of entities and the values provided for entities are cross-checked and errors are highlighted.

The element name has been defined as “restaurant-name” in the XSD file, but since it has been misspelled, it has raised an error.

The price has been limited between $0 and $7, so a price outside the range has been flagged as an invalid value.

The data type for price has been set as decimal, so when a value that is a string has been given, it is marked as an invalid data type, and it is also prompted for decimal data type.

For hours in the restaurant, a condition has been given to check if it has values from Mon to Sun, but since there is a spelling mistake, an error has been raised.

XSL styling:

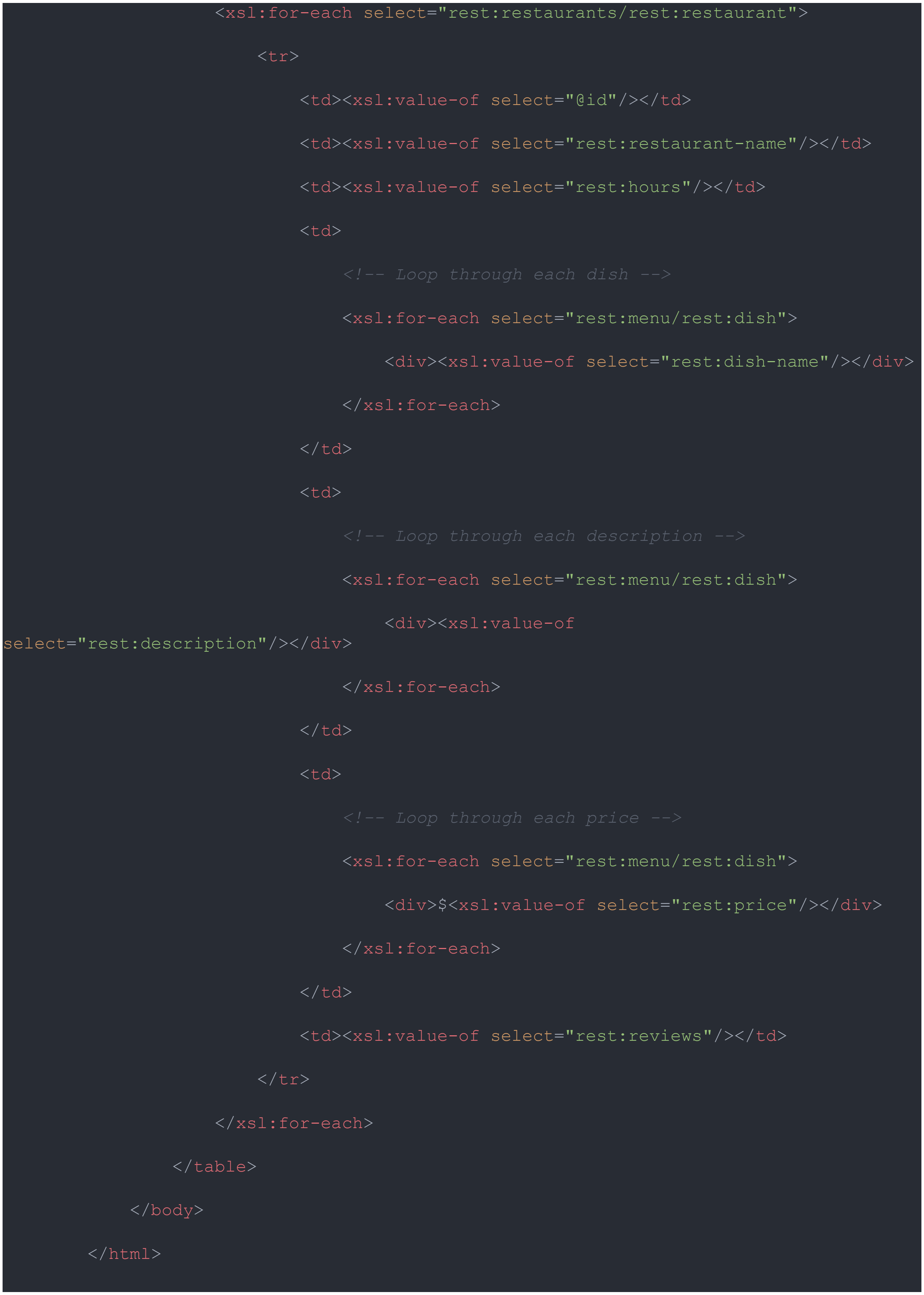
The given sheet is the html styling received after using “restaurant.xsl” to give a styling reference to the XML file. As seen, it is only a styling sheet, and does not validate data. This is why data that has been marked as errors in the editor still can be seen in the HTML page. The sytling sheet provides a user-friendly representation of the XML data and makes it easy to view.

restaurant.xsd

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="www.abc.com/restaurant" xmlns:rest="www.abc.com/restaurant"
elementFormDefault="qualified">


    <xs:element name="restaurants" type="rest:restaurantsType"></xs:element>

    <xs:complexType name="restaurantsType">

        <xs:sequence>

            <xs:element name="restaurant" type="rest:restaurantType"
maxOccurs="unbounded">

            </xs:element>

        </xs:sequence>

    </xs:complexType>


    <xs:complexType name="restaurantType">

        <xs:sequence>

            <xs:element name="restaurant-id" type="xs:integer"></xs:element>

            <xs:element name="restaurant-name" type="xs:string"/>
            <xs:element name="hours" type="rest:hoursType"/>

            <xs:element name="menu" type="rest:menuType"></xs:element>

            <xs:element name="reviews" type="xs:string"></xs:element>

        </xs:sequence>

        <xs:attribute name="id" type="xs:string" use="required"></xs:attribute>

    </xs:complexType>



    <xs:simpleType name="hoursType">

        <xs:restriction base="xs:string">

                <xs:pattern
value="(Mon|Tue|Wed|Thu|Fri|Sat|Sun)-(Mon|Tue|Wed|Thu|Fri|Sat|Sun):
[0-9]{1,2}(am|pm)-[0-9]{1,2}(am|pm)"/>

        </xs:restriction>

    </xs:simpleType>



    <xs:complexType name="menuType">

        <xs:sequence>

            <xs:element name="dish" maxOccurs="unbounded"
type="rest:dishType"></xs:element>

        </xs:sequence>

    </xs:complexType>



    <xs:complexType name="dishType">

        <xs:sequence>

            <xs:element name="dish-name" type="xs:string"/>

            <xs:element name="description" type="xs:string"/>

            <xs:element name="price" type="rest:priceType"/>
        </xs:sequence>
        <xs:attribute name="id" type="xs:string" use="required"></xs:attribute>
    </xs:complexType>

    <xs:simpleType name="priceType">
        <xs:restriction base="xs:decimal">
            <xs:minInclusive value="0.00"/>
            <xs:maxInclusive value="7.00"/>
            <xs:fractionDigits value="2"/>
        </xs:restriction>
    </xs:simpleType>

</xs:schema>
```

restaurant.xsl

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0"
    xmlns:rest="www.abc.com/restaurant"
    exclude-result-prefixes="rest">
    <xsl:output method="html" encoding="UTF-8" indent="yes"/>

    <!-- Root template -->
    <xsl:template match="/">
        <html>
            <head>
                <title>Restaurants</title>
                <style>
                    table {
                        width: 100%;
                        border-collapse: collapse;
                    }
                    table, th, td {
                        border: 1px solid black;
                    }
                    th, td {
                        padding: 8px;
                        text-align: left;
                    }
                    th {
                        background-color: #f2f2f2;
                    }
                </style>
            </head>
            <body>
                <h2>Restaurant Menu</h2>
                <table>
                    <tr>
                        <th>ID</th>
                        <th>Name</th>
                        <th>Hours</th>
                        <th>Dish</th>
                        <th>Description</th>
                        <th>Price</th>
                        <th>Reviews</th>
                    </tr>
                    <!-- Loop through each restaurant -->
```



```
                    <xsl:for-each select="rest:restaurants/rest:restaurant">
                        <tr>
                            <td><xsl:value-of select="@id"/></td>
                            <td><xsl:value-of select="rest:restaurant-name"/></td>
                            <td><xsl:value-of select="rest:hours"/></td>
                            <td>
                                <!-- Loop through each dish -->
                                <xsl:for-each select="rest:menu/rest:dish">
                                    <div><xsl:value-of select="rest:dish-name"/></div>
                                </xsl:for-each>
                            </td>
                            <td>
                                <!-- Loop through each description -->
                                <xsl:for-each select="rest:menu/rest:dish">
                                    <div><xsl:value-of
select="rest:description"/></div>
                                </xsl:for-each>
                            </td>
                            <td>
                                <!-- Loop through each price -->
                                <xsl:for-each select="rest:menu/rest:dish">
                                    <div>$<xsl:value-of select="rest:price"/></div>
                                </xsl:for-each>
                            </td>
                            <td><xsl:value-of select="rest:reviews"/></td>
                        </tr>
                    </xsl:for-each>
                </table>
            </body>
        </html>
```

```
    </xsl:template>

</xsl:stylesheet>
```