**BikeRental.xml**

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

<?xml-stylesheet type="text/xsl" href="transform.xsl"?>

<Data xmlns="http://www.bikerental.com"

      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

      xsi:schemaLocation="http://www.bikerental.com bikeRental.xsd">

    <bikeRental>

        <bike id="B1" type="mountain">

            <name>Mountain Bike</name>

            <rentalDate>2024-08-10</rentalDate>

            <returnDate>2024-08-11</returnDate>

            <rentalPrice>25.00</rentalPrice>

            <renterName>John Doe</renterName>

            <location>Downtown</location>

        </bike>

        <bike id="B2" type="road">

            <name>Road Bike</name>

            <rentalDate>2024-08-15</rentalDate>

            <returnDate>2024-08-16</returnDate>

            <rentalPrice>30.00</rentalPrice>

            <renterName>Jane Smith</renterName>

            <location>Uptown</location>

        </bike>

    </bikeRental>

    <Rents>

        <Bikes>

            <id>1</id>

            <name>CBR 120</name>

            <description>Advanced engine</description>

            <rent>33</rent>

            <stock>2</stock>

        </Bikes>

        <Bikes>

            <id>2</id>

            <name>SPLENDOR</name>

            <description>City-based use</description>

            <rent>22</rent>

            <stock>3</stock>

        </Bikes>

        <Bikes>

            <id>3</id>

            <name>GSX 320</name>

            <description>Sports bike</description>

            <rent>65</rent>

            <stock>1</stock>

        </Bikes>

    </Rents>

</Data>

**bikeRental.dtd**

<!ELEMENT bikeRental (rental+)>

<!ELEMENT rental (bikeModel, renterName, rentalDate, returnDate, rentalPrice, pickupLocation)>

<!ATTLIST rental id ID #REQUIRED>

<!ELEMENT bikeModel (#PCDATA)>

<!ELEMENT renterName (#PCDATA)>

<!ELEMENT rentalDate (#PCDATA)>

<!ELEMENT returnDate (#PCDATA)>

<!ELEMENT rentalPrice (#PCDATA)>

<!ELEMENT pickupLocation (#PCDATA)>

**bikeRental.xsd**

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

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"

           xmlns="http://www.bikerental.com"

           targetNamespace="http://www.bikerental.com"

           elementFormDefault="qualified">

    <xs:element name="Data">

        <xs:complexType>

            <xs:sequence>

                <xs:element name="bikeRental">

                    <xs:complexType>

                        <xs:sequence>

                            <xs:element name="bike" maxOccurs="unbounded">

                                <xs:complexType>

                                    <xs:sequence>

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

                                        <xs:element name="rentalDate" type="xs:date"/>

                                        <xs:element name="returnDate" type="xs:date"/>

                                        <xs:element name="rentalPrice">

                                            <xs:simpleType>

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

                                                    <xs:minInclusive value="0.00"/>

                                                    <xs:fractionDigits value="2"/>

                                                </xs:restriction>

                                            </xs:simpleType>

                                        </xs:element>

                                        <xs:element name="renterName">

                                            <xs:simpleType>

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

                                                    <xs:minLength value="3"/>

                                                </xs:restriction>

                                            </xs:simpleType>

                                        </xs:element>

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

                                    </xs:sequence>

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

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

                                </xs:complexType>

                            </xs:element>

                        </xs:sequence>

                    </xs:complexType>

                </xs:element>

                <xs:element name="Rents">

                    <xs:complexType>

                        <xs:sequence>

                            <xs:element name="Bikes" maxOccurs="unbounded">

                                <xs:complexType>

                                    <xs:sequence>

                                        <xs:element name="id" type="xs:int"/>

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

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

                                        <xs:element name="rent" type="xs:decimal"/>

                                        <xs:element name="stock" type="xs:int"/>

                                    </xs:sequence>

                                </xs:complexType>

                            </xs:element>

                        </xs:sequence>

                    </xs:complexType>

                </xs:element>

            </xs:sequence>

        </xs:complexType>

    </xs:element>

</xs:schema>

**transform.xsl**

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

<xsl:stylesheet version="1.0"

  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"

  xmlns:br="http://www.bikerental.com"

  exclude-result-prefixes="br">

  <xsl:output method="html" indent="yes"/>

  <!-- Template to match the root element -->

  <xsl:template match="/">

    <html>

      <head>

        <title>Bike Rentals</title>

      </head>

      <body>

        <h1>Current Bike Rentals</h1>

        <table border="1">

          <thead>

            <tr>

              <th>Bike ID</th>

              <th>Type</th>

              <th>Name</th>

              <th>Rental Date</th>

              <th>Return Date</th>

              <th>Rental Price</th>

              <th>Renter Name</th>

              <th>Location</th>

            </tr>

          </thead>

          <tbody>

            <!-- Use proper namespace prefix to select elements -->

            <xsl:for-each select="//br:bikeRental/br:bike">

              <tr>

                <td><xsl:value-of select="@id"/></td>

                <td><xsl:value-of select="@type"/></td>

                <td><xsl:value-of select="br:name"/></td>

                <td><xsl:value-of select="br:rentalDate"/></td>

                <td><xsl:value-of select="br:returnDate"/></td>

                <td><xsl:value-of select="br:rentalPrice"/></td>

                <td><xsl:value-of select="br:renterName"/></td>

                <td><xsl:value-of select="br:location"/></td>

              </tr>

            </xsl:for-each>

          </tbody>

        </table>

        <h1>Bike Rent Details</h1>

        <table border="1">

          <thead>

            <tr>

              <th>Bike ID</th>

              <th>Name</th>

              <th>Description</th>

              <th>Rent</th>

              <th>Stock</th>

            </tr>

          </thead>

          <tbody>

            <!-- Use default namespace to select elements -->

            <xsl:for-each select="//br:Rents/br:Bikes">

              <tr>

                <td><xsl:value-of select="br:id"/></td>

                <td><xsl:value-of select="br:name"/></td>

                <td><xsl:value-of select="br:description"/></td>

                <td><xsl:value-of select="br:rent"/></td>

                <td><xsl:value-of select="br:stock"/></td>

              </tr>

            </xsl:for-each>

          </tbody>

        </table>

      </body>

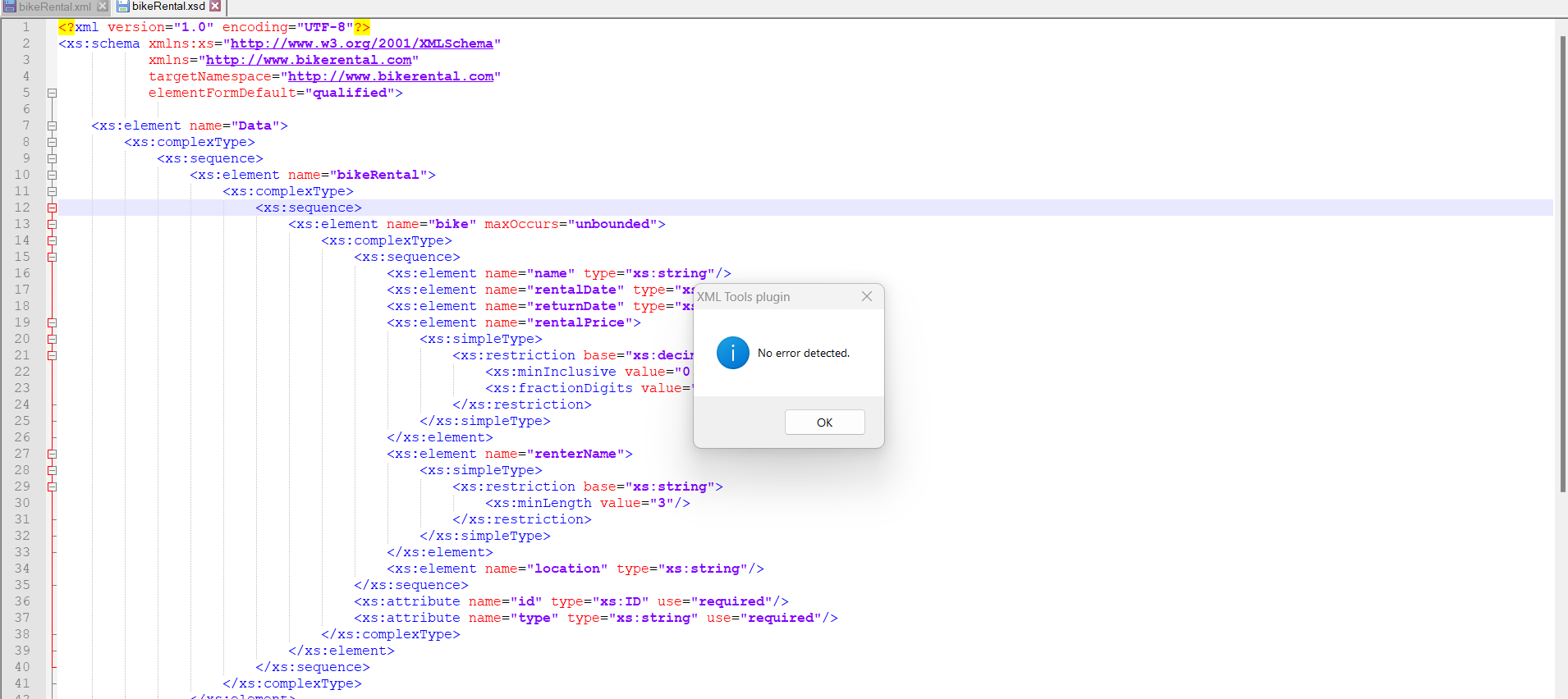
    </html>

  </xsl:template>

</xsl:stylesheet>

**Validation process**

in the validation process we are validating the following xml file with the xsd file and found out that there is no error in the xsd file. In this process we have used the notepad++ to validate the xml file against the xsd file

****

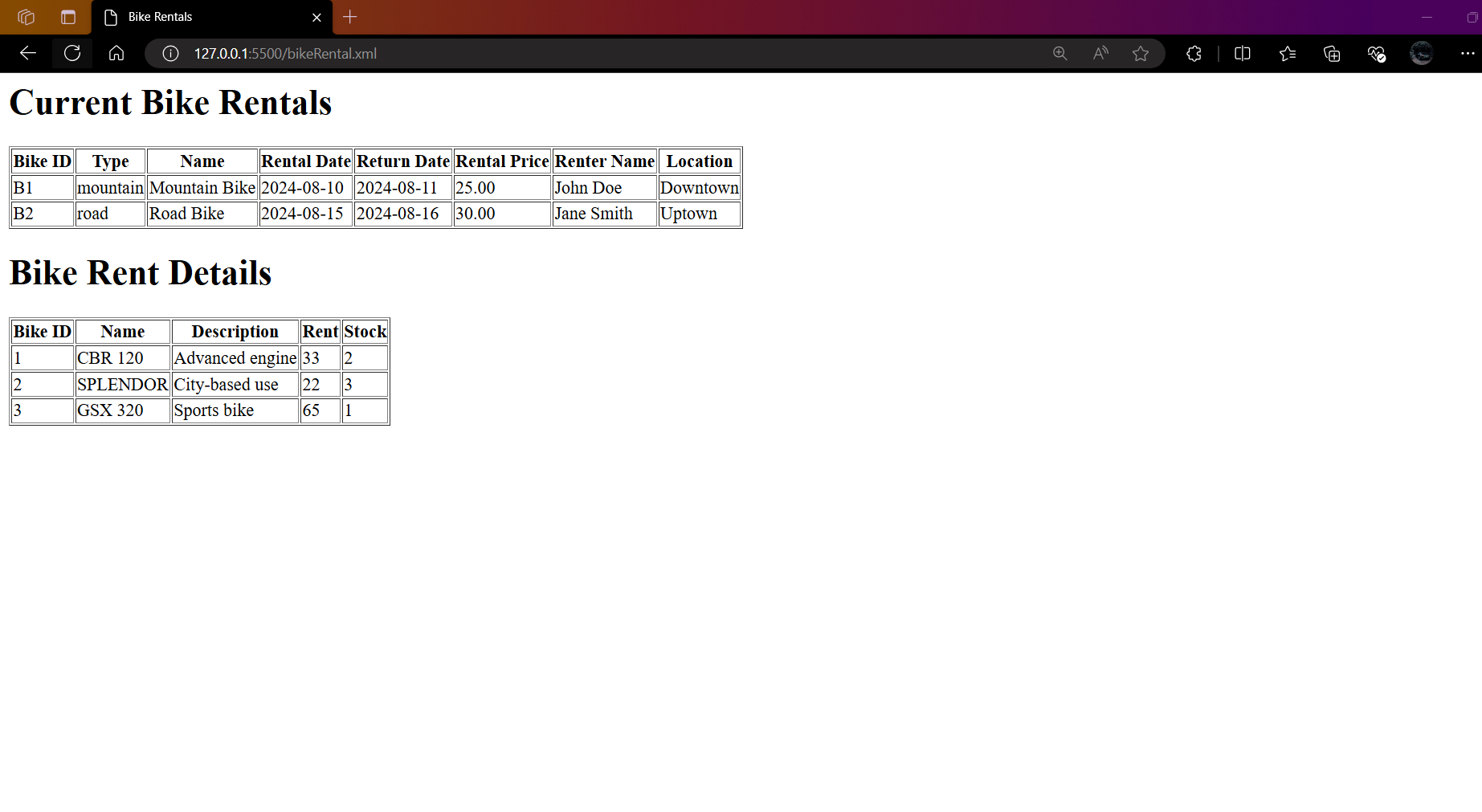
**Testing process**

In the testing process we give a wrong type of data in the xml file and try to validate it against the xsd file but because of the wrong type of data defined in the xml file there arises an error and it is being shown in the testing process when we try to validate the file

****

**OUTPUT IN THE FORM OF HTML**

For this process we use a file called transformr.xsl file to transform the following xml file to the html format in the form of rows and columns and we display them in the html page.



**Summary**

**1. XML Structure:**

* The XML file (BikeRental.xml) contains information about bike rentals and bike rent details.
* It uses namespaces, with elements organized under two main categories: bikeRental and Rents.
* The XML includes a reference to an XSL stylesheet (transform.xsl) for transforming the XML data into an HTML table format.

**2. DTD and XSD Validation:**

* **DTD (bikeRental.dtd):** Defines the structure for the bikeRental section, including the required elements like bikeModel, renterName, rentalDate, returnDate, rentalPrice, and pickupLocation.
* **XSD (bikeRental.xsd):** Provides a more comprehensive schema for validating the entire XML file. It defines complex types for the bikeRental and Rents sections, enforcing constraints such as minimum length for strings, minimum values for decimal types, and required attributes for elements like bike.

**3. Validation Process:**

* **Validation with Notepad++:** The XML file was validated against the XSD schema using Notepad++, and no errors were found.
* **Testing:** The process involved deliberately introducing errors, such as incorrect data types in the XML file, to ensure the schema would correctly identify validation issues. The errors were detected as expected, confirming the robustness of the XSD schema.

**4. XSL Transformation:**

* **Transforming XML to HTML:** The XSL file (transform.xsl) was used to transform the XML data into an HTML format. The transformation outputs two tables: one for current bike rentals and another for bike rent details. The tables are generated by iterating through the XML elements and displaying the relevant data in a structured format.

**5. Issues Encountered:**

* **Validation Tool Selection:** Initially, there was uncertainty about which application to use for validating the XML file against the XSD. This was resolved by using Notepad++.
* **HTML Display Issues:** Challenges were faced when displaying the XML data as an HTML table. These issues were eventually resolved, allowing the XML data to be successfully transformed and displayed in the desired format.

**6. Final Output:**

* The project successfully validated the XML file against the XSD, tested error handling in the schema, and transformed the XML data into an HTML format using XSLT. The final output was a clean and well-structured HTML page displaying the bike rental information in tables.