cars.dtd:

```
<!ELEMENT cars (car+)>
<!ELEMENT car (id, company_name, model, engine, mileage)>
<!ELEMENT id (#PCDATA)>
<!ELEMENT company_name (#PCDATA)>
<!ELEMENT model (#PCDATA)>
<!ELEMENT engine (#PCDATA)>
<!ELEMENT mileage (#PCDATA)>
```

cars.xsd:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="cars">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="car" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="id" type="xs:int"/>
                            <xs:element name="company_name" type="xs:string"/>
                            <xs:element name="model" type="xs:string"/>
                            <xs:element name="engine" type="xs:string"/>
                            <xs:element name="mileage" type="xs:int"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

cars.xsl:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
    <xsl:output method="html" indent="yes"/>
    <xsl:template match="/cars">
        <html>
```

```
        <head>
            <title>Car Information</title>
            <style>
                table { border-collapse: collapse; width: 100%; }
                table, th, td { border: 1px solid black; }
                th, td { padding: 8px; text-align: left; }
                th { background-color: #f2f2f2; }
            </style>
        </head>
        <body>
            <h1>Car Information</h1>
            <table>
                <tr>
                    <th>ID</th>
                    <th>Company Name</th>
                    <th>Model</th>
                    <th>Engine</th>
                    <th>Mileage</th>
                </tr>
                <xsl:for-each select="car">
                    <tr>
                        <td><xsl:value-of select="id"/></td>
                        <td><xsl:value-of select="company_name"/></td>
                        <td><xsl:value-of select="model"/></td>
                        <td><xsl:value-of select="engine"/></td>
                        <td><xsl:value-of select="mileage"/></td>
                    </tr>
                </xsl:for-each>
            </table>
        </body>
    </html>
    </xsl:template>
</xsl:stylesheet>


cars.xml:
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="cars.xsl" type="text/xsl"?>
```

```
<!DOCTYPE cars SYSTEM "cars.dtd">

<cars>

    <car>

        <id>1</id>

        <company_name>Tesla</company_name>

        <model>Model S</model>

        <engine>Electric</engine>

        <mileage>35000</mileage>

    </car>

    <car>

        <id>2</id>

        <company_name>Ford</company_name>

        <model>Mustang</model>

        <engine>V8</engine>

        <mileage>50000</mileage>

    </car>

    <car>

        <id>3</id>

        <company_name>BMW</company_name>

        <model>Series 3</model>

        <engine>Inline 6</engine>

        <mileage>25000</mileage>

    </car>

</cars>
```

OUTPUT:

Car Information

ID	Company Name	Model	Engine	Mileage
1	Tesla	Model S	Electric	35000
2	Ford	Mustang	V8	50000
3	BMW	Series 3	Inline 6	25000

CONCLUSION: The program to implement the xml document using dtd and xml schema and to display it using xsl was implemented successfully.