PRACTICAL-9

Reading and Writing an XML file

Que.1: Create a Student table with following fields (S_id, F_Name, M_Name, S_Name, Address, City, gender, Branch, semester, Contact, Email_id).

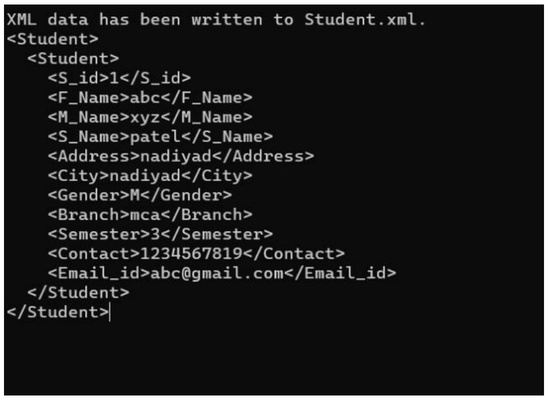
```
CREATE TABLE [dbo].[Student]
(
   [S_id] INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
   [F_Name] VARCHAR(255) NOT NULL,
   [M_Name] VARCHAR(255),
   [S_Name] VARCHAR(255) NOT NULL,
   [Address] VARCHAR(255) NOT NULL,
   [City] VARCHAR(100) NOT NULL,
   [Gender] VARCHAR(10) NOT NULL,
   [Branch] VARCHAR(255),
   [Semester] INT,
   [Contact] VARCHAR(20),
   [Email_id] VARCHAR(255) NOT NULL
);
```

Que.2: Read the records from the table and store in the xml file "student.xml". Also display the content of the file on the console. (Use XmlReader and XmlWriter to read and write)..

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Xml;

namespace q2
{
   class Program
   {
      static void Main(string[] args)
      {
         string connectionString = @"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\iamja\OneDrive\Documents\LAB8.mdf;Integrated
Security=True;Connect Timeout=30";

         try
         {
            using (SqlConnection connection = new SqlConnection(connectionString))
            {
               connection.Open();

               // SQL query to retrieve data from the "User" table and convert it to XML
               string query = "SELECT * FROM [Student] FOR XML AUTO, ELEMENTS, ROOT('Student')";

               using (SqlCommand command = new SqlCommand(query, connection))
               {
                  // Execute the query and get the result as a single XML string
                  string xmlResult = (string)command.ExecuteScalar();
```

```
                // Save the XML to a file (e.g., "users.xml")
                System.IO.File.WriteAllText("Student.xml", xmlResult);

                Console.WriteLine("XML data has been written to Student.xml.");
            }
        }

        // Read and display the XML content from the file
        string xmlFilePath = "Student.xml";
        using (XmlTextReader reader = new XmlTextReader(xmlFilePath))
        {
            reader.WhitespaceHandling = WhitespaceHandling.None;

            using (XmlTextWriter writer = new XmlTextWriter(Console.Out) { Formatting = Formatting.Indented })
            {
                while (reader.Read())
                {
                    writer.WriteNode(reader, true);
                }
            }
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine("An error occurred: " + ex.Message);
    }
    finally
    {
        Console.ReadKey();
    }
   }
  }
}
```

Output:



```
XML data has been written to Student.xml.
<Student>
  <Student>
    <S_id>1</S_id>
    <F_Name>abc</F_Name>
    <M_Name>xyz</M_Name>
    <S_Name>patel</S_Name>
    <Address>nadiyad</Address>
    <City>nadiyad</City>
    <Gender>M</Gender>
    <Branch>mca</Branch>
    <Semester>3</Semester>
    <Contact>1234567819</Contact>
    <Email_id>abc@gmail.com</Email_id>
  </Student>
</Student>
```

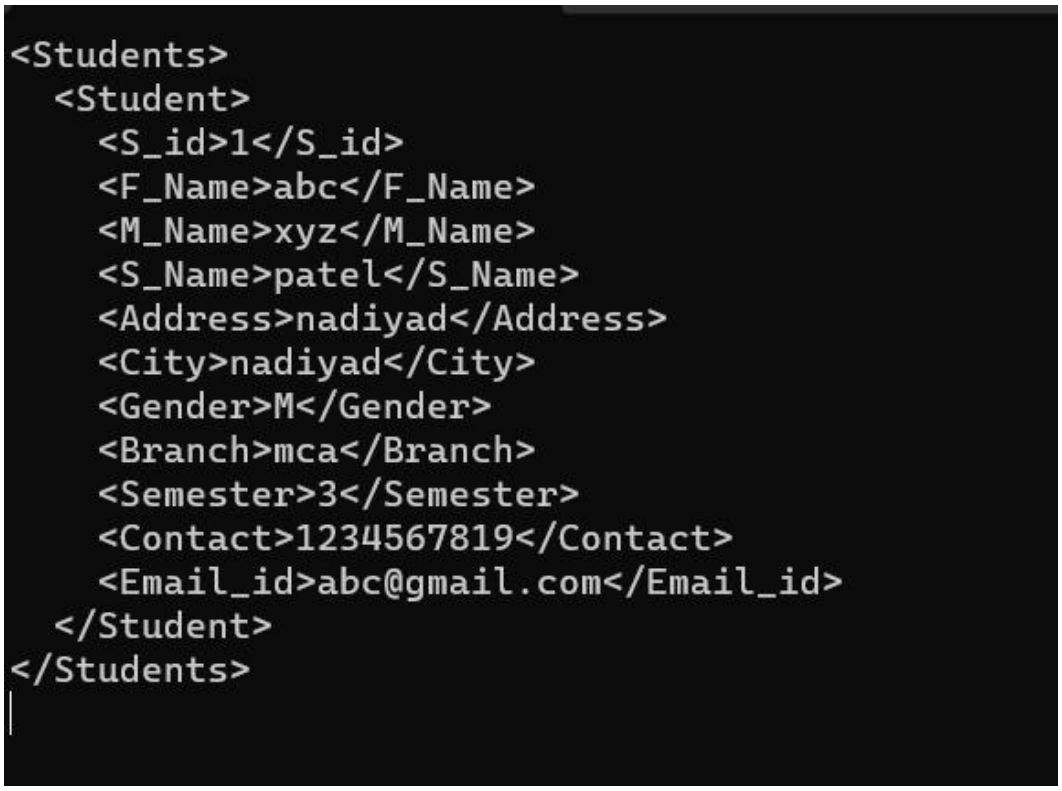
PRACTICAL-9

Que.3: Perform program 1 using XmlDocument class.

.

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Xml;

namespace q2
{
    class Program
    {
        static void Main()
        {
            string connectionString = @"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\iamja\OneDrive\Documents\LAB8.mdf;Integrated
Security=True;Connect Timeout=30";
            string tableName = "Student";

            // Create a connection to the database
            using (SqlConnection connection = new SqlConnection(connectionString))
            {
                connection.Open();

                // SQL query to retrieve records from the Student table
                string query = $"SELECT * FROM {tableName}";

                using (SqlCommand command = new SqlCommand(query, connection))
                {
                    using (SqlDataReader reader = command.ExecuteReader())
                    {
                        // Create a new XmlDocument
                        XmlDocument xmlDoc = new XmlDocument();
                        XmlElement rootElement = xmlDoc.CreateElement("Students");
                        xmlDoc.AppendChild(rootElement);

                        while (reader.Read())
                        {
                            // Create a new Student element for each record
                            XmlElement studentElement = xmlDoc.CreateElement("Student");

                            for (int i = 0; i < reader.FieldCount; i++)
                            {
                                // Create elements for each field and add them to the Student element
                                XmlElement fieldElement = xmlDoc.CreateElement(reader.GetName(i));
                                fieldElement.InnerText = reader[i].ToString();
                                studentElement.AppendChild(fieldElement);
                            }

                            // Add the Student element to the root element
                            rootElement.AppendChild(studentElement);
                        }
                        // Save the XmlDocument to a file named "student.xml"
                        xmlDoc.Save("student2.xml");
                    }
                }
            }

            // Display the content of the XML file on the console
            string xmlContent = System.IO.File.ReadAllText("student2.xml");
            Console.WriteLine(xmlContent);
            Console.ReadKey();
        }
    }
}
```

Output:



```
<Students>
  <Student>
    <S_id>1</S_id>
    <F_Name>abc</F_Name>
    <M_Name>xyz</M_Name>
    <S_Name>patel</S_Name>
    <Address>nadiyad</Address>
    <City>nadiyad</City>
    <Gender>M</Gender>
    <Branch>mca</Branch>
    <Semester>3</Semester>
    <Contact>1234567819</Contact>
    <Email_id>abc@gmail.com</Email_id>
  </Student>
</Students>
```

Que.4: Read an existing XML file and find out total number of lines in an XML file as well as display total attributes, white spaces, name of element and line number for each line of xml.

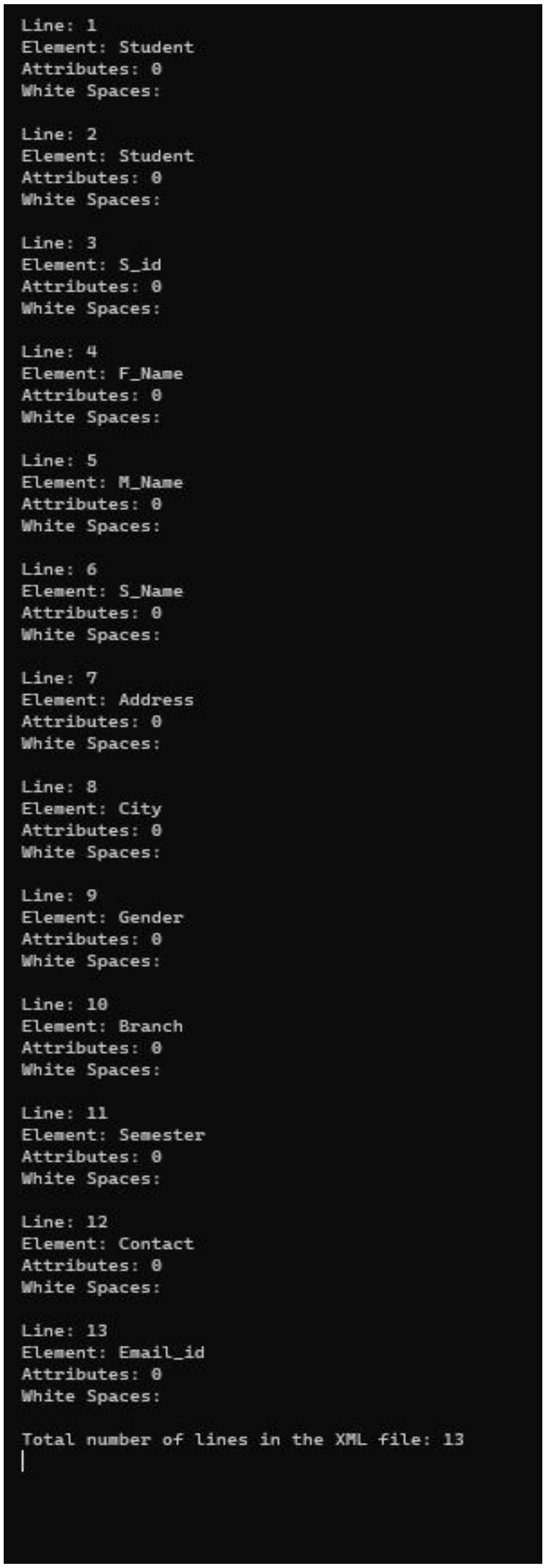
```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Xml;
using System.IO;

namespace q2
{
  class Program
  {
```

PRACTICAL-9

```
    static void Main()
    {
        string xmlFilePath = "student.xml"; // Replace with the path to your XML file

        if (File.Exists(xmlFilePath))
        {
            using (XmlTextReader reader = new XmlTextReader(xmlFilePath))
            {
                int lineCount = 0;

                while (reader.Read())
                {
                    switch (reader.NodeType)
                    {
                        case XmlNodeType.Element:
                            lineCount++;
                            string elementName = reader.Name;
                            int attributeCount = reader.AttributeCount;
                            string whitespace = reader.Value; // Get white spaces if needed

                            Console.WriteLine($"Line: {lineCount}");
                            Console.WriteLine($"Element: {elementName}");
                            Console.WriteLine($"Attributes: {attributeCount}");
                            Console.WriteLine($"White Spaces: {whitespace}");
                            Console.WriteLine();

                            break;
                        case XmlNodeType.EndElement:
                            break;
                        case XmlNodeType.Text:
                            // Handle text nodes if needed
                            break;
                            // Add cases for other node types if needed
                    }
                }

                Console.WriteLine($"Total number of lines in the XML file: {lineCount}");
                Console.ReadKey();
            }
        }
        else
        {
            Console.WriteLine("The XML file does not exist.");
        }
    }
  }
}
```

PRACTICAL-9

Output :



```
Line: 1
Element: Student
Attributes: 0
White Spaces:

Line: 2
Element: Student
Attributes: 0
White Spaces:

Line: 3
Element: S_id
Attributes: 0
White Spaces:

Line: 4
Element: F_Name
Attributes: 0
White Spaces:

Line: 5
Element: M_Name
Attributes: 0
White Spaces:

Line: 6
Element: S_Name
Attributes: 0
White Spaces:

Line: 7
Element: Address
Attributes: 0
White Spaces:

Line: 8
Element: City
Attributes: 0
White Spaces:

Line: 9
Element: Gender
Attributes: 0
White Spaces:

Line: 10
Element: Branch
Attributes: 0
White Spaces:

Line: 11
Element: Semester
Attributes: 0
White Spaces:

Line: 12
Element: Contact
Attributes: 0
White Spaces:

Line: 13
Element: Email_id
Attributes: 0
White Spaces:

Total number of lines in the XML file: 13
```

PRACTICAL-9

Que.5: Read the content of the xml file created in problem 1. Also create an xslt style-sheet to transform the xml file and display the student data sorted by F_name in a table in the output.

student.xslt

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/Students">
        <html>
                <head>
                        <style>
                                table {
                                border-collapse: collapse;
                                width: 100%;
                                }
                                th, td {
                                border: 1px solid black;
                                padding: 8px;
                                text-align: left;
                                }
                                th {
                                background-color: #f2f2f2;
                                }
                        </style>
                </head>
                <body>
                        <h1>Student Data Sorted by F_Name</h1>
                        <table>
                                <tr>
                                        <th>S_id</th>
                                        <th>F_Name</th>
                                        <th>M_Name</th>
                                        <th>S_Name</th>
                                        <th>Address</th>
                                        <th>City</th>
                                        <th>Gender</th>
                                        <th>Branch</th>
                                        <th>Semester</th>
                                        <th>Contact</th>
                                        <th>Email_id</th>
                                </tr>
                                <xsl:apply-templates select="Student">
                                        <xsl:sort select="F_Name"/>
                                </xsl:apply-templates>
                        </table>
                </body>
        </html>
</xsl:template>

<xsl:template match="Student">
        <tr>
                <td>
```

```
                        <xsl:value-of select="S_id"/>
                </td>
                <td>
                        <xsl:value-of select="F_Name"/>
                </td>
                <td>
                        <xsl:value-of select="M_Name"/>
                </td>
                <td>
                        <xsl:value-of select="S_Name"/>
                </td>
                <td>
                        <xsl:value-of select="Address"/>
                </td>
                <td>
                        <xsl:value-of select="City"/>
                </td>
                <td>
                        <xsl:value-of select="Gender"/>
                </td>
                <td>
                        <xsl:value-of select="Branch"/>
                </td>
                <td>
                        <xsl:value-of select="Semester"/>
                </td>
                <td>
                        <xsl:value-of select="Contact"/>
                </td>
                <td>
                        <xsl:value-of select="Email_id"/>
                </td>
        </tr>
</xsl:template>
</xsl:stylesheet>
```

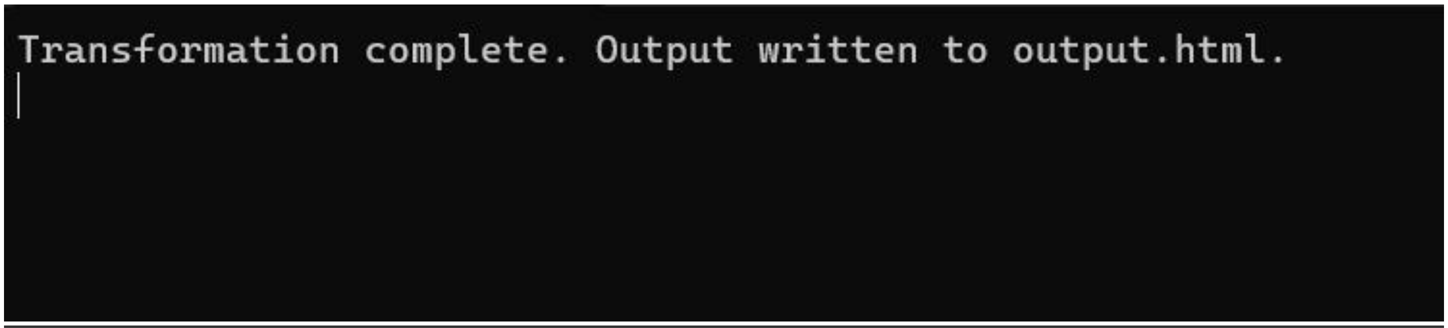
Que5.cs

```
using System.Xml.Xsl;

class Program
{
    static void Main()
    {
        string xmlFilePath = "Student.xml";
        string xsltFilePath = @"D:\MCA\SEM 3\ASP.net\LAB\q2\q2\student.xslt";
        string outputFilePath = @"D:\MCA\SEM 3\ASP.net\LAB\q2\q2\output.html";

        XslCompiledTransform xslt = new XslCompiledTransform();
        xslt.Load(xsltFilePath);
        xslt.Transform(xmlFilePath, outputFilePath);
```

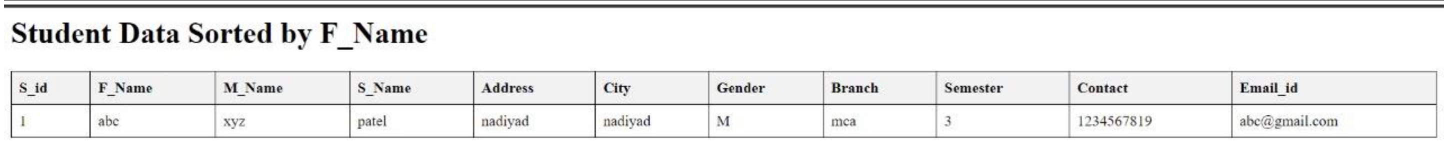
PRACTICAL-9

```
        System.Console.WriteLine("Transformation complete. Output written to output.html.");
        System.Console.ReadKey();
    }
}
```

Output:



```
Transformation complete. Output written to output.html.
```



Student Data Sorted by F_Name

S_id	F_Name	M_Name	S_Name	Address	City	Gender	Branch	Semester	Contact	Email_id
1	abc	xyz	patel	nadiyad	nadiyad	M	mca	3	1234567819	abc@gmail.com