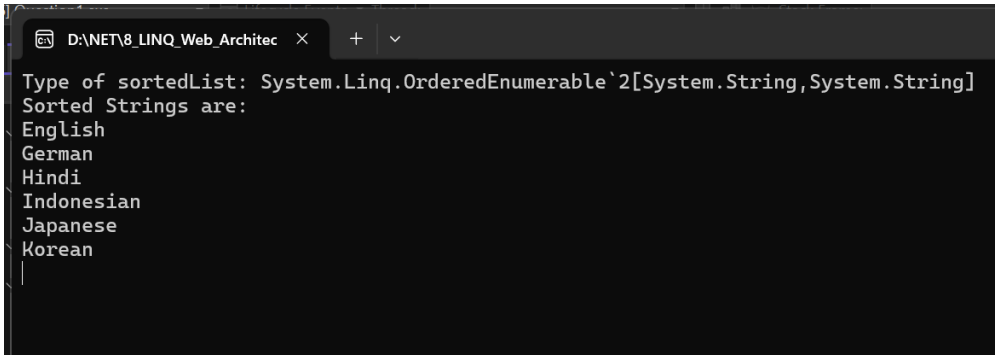
Assignment8

QUESTION 1 – LINQ to Array / List Sorting (Ascending)

```
using System;
using System.Collections.Generic;
using System.Linq;

namespace Assignment8_Q1
{
   class Program
   {
      static void Main(string[] args)
      {
         List<string> languages = new List<string>
         {
            "Indonesian", "Korean", "Japanese", "English", "German", "Hindi"
         };

         var sortedList = from lang in languages
                          orderby lang
                          select lang;

         // Print type of sortedList
         Console.WriteLine("Type of sortedList: " + sortedList.GetType());

         Console.WriteLine("Sorted Strings are:");
         foreach (var item in sortedList)
         {
            Console.WriteLine(item);
         }

         Console.ReadLine();
      }
   }
}
```

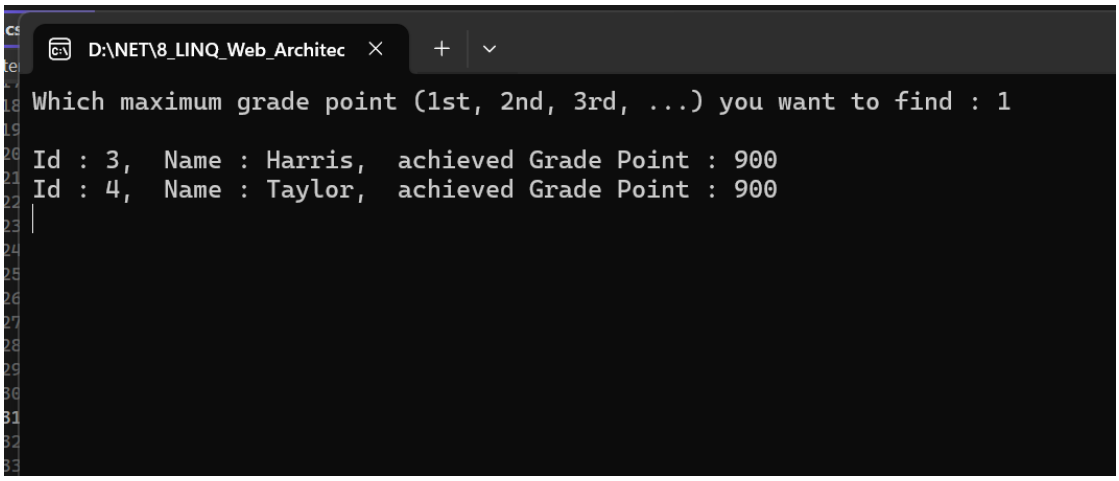


👉 **LINQ is a set of *extension methods***
👉 When we write using System.Linq;, we add methods like ***OrderBy, Where, Select*** to all types that implement **IEnumerable<T>**.

We unlock over **100 extension methods**, including: ***Where, Select, OrderBy, ThenBy, GroupBy, Count, etc.***

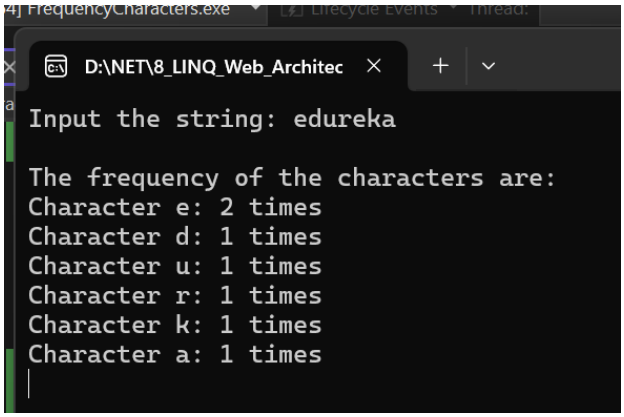
QUESTION 2 – LINQ Filter Students by Grade

```
using System;
using System.Collections.Generic;
using System.Linq;

namespace Assignment8_Q2
{
    class Student
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public int Grade { get; set; }
        public int GradePoint { get; set; }
    }

    class Program
    {
        static void Main(string[] args)
        {
            List<Student> students = new List<Student>()
            {
                new Student { Id = 1, Name = "John",   Grade = 3, GradePoint = 850 },
                new Student { Id = 2, Name = "Emily",  Grade = 2, GradePoint = 760 },
                new Student { Id = 3, Name = "Harris", Grade = 1, GradePoint = 900 },
                new Student { Id = 4, Name = "Taylor", Grade = 1, GradePoint = 900 },
                new Student { Id = 5, Name = "Chris",  Grade = 3, GradePoint = 780 }
            };

            Console.Write("Which maximum grade point (1st, 2nd, 3rd, ...) you want to find : ");
            int userGrade = int.Parse(Console.ReadLine());

            var result = from s in students
                         where s.Grade == userGrade
                         select s;

            Console.WriteLine();
            foreach (var s in result)
            {
                Console.WriteLine($"Id : {s.Id},  Name : {s.Name},  achieved Grade Point : {s.GradePoint}");
            }

            Console.ReadLine();
        }
    }
}
```



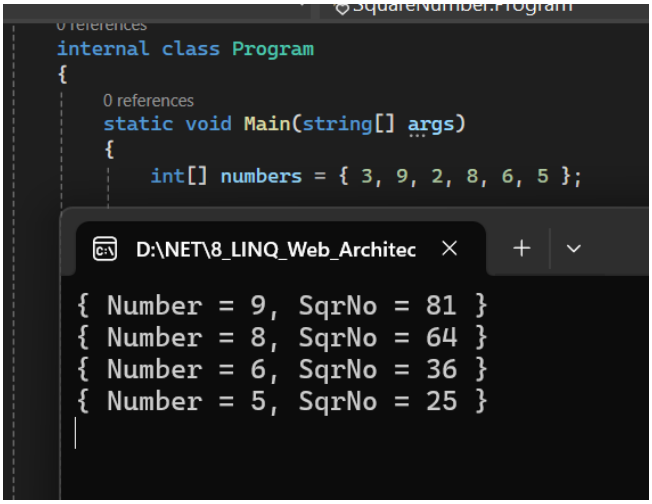
QUESTION 3 — Frequency of Characters (Solved in C# with LINQ)

```
using System;
using System.Linq;

namespace Assignment8_Q3
{
   class Program
   {
      static void Main(string[] args)
      {
         Console.Write("Input the string: ");
         string input = Console.ReadLine();

         Console.WriteLine("\nThe frequency of the characters are:");

         var result =
            from c in input
            group c by c into g
            select new
            {
               Character = g.Key,
               Count = g.Count()
            };

         foreach (var item in result)
         {
            Console.WriteLine($"Character {item.Character}: {item.Count} times");
         }

         Console.ReadLine();
      }
   }
}
```



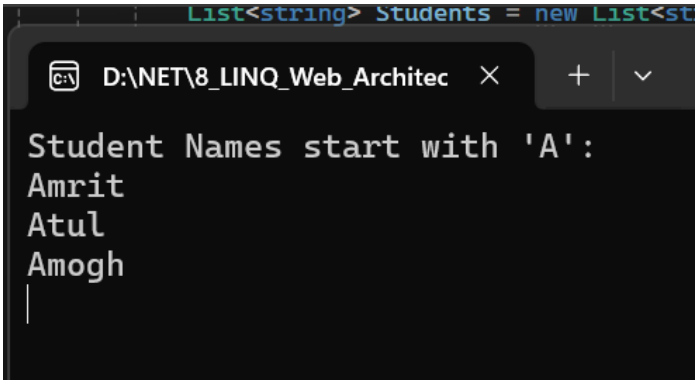
QUESTION 4 — Square of Numbers > 20 (Classic LINQ)

```
using System;
using System.Linq;

namespace Assignment8_Q4
{
   class Program
   {
      static void Main(string[] args)
      {
         int[] numbers = { 3, 9, 2, 8, 6, 5 };

         var result =
            from n in numbers
            let sq = n * n
            where sq > 20
            select new
            {
               Number = n,
               SqrNo = sq
            };

         foreach (var item in result)
         {
            Console.WriteLine($"{{ Number = {item.Number}, SqrNo = {item.SqrNo} }}");
         }

         Console.ReadLine();
      }
   }
}
```



QUESTION 5 – Searching List Elements Starting With 'A'

```
using System;
using System.Collections.Generic;
using System.Linq;

namespace Assignment8_Q5
{
   class Program
   {
      static void Main(string[] args)
      {
         List<string> Students = new List<string>();
         Students.Add("Amrit");
         Students.Add("Sumit");
         Students.Add("Atul");
         Students.Add("Shaurya");
         Students.Add("Amogh");

         // Classic LINQ syntax
         var result =
            from s in Students
            where s.StartsWith("A", StringComparison.OrdinalIgnoreCase)
            select s;

         Console.WriteLine("Student Names start with 'A':");
         foreach (var name in result)
         {
            Console.WriteLine(name);
         }

         Console.ReadLine();
      }
   }
}
```



Question6 CRUD with LINQ to XML

products.xml | Program.cs* | Output | Error List

```xml
<?xml version="1.0" encoding="utf-8"?>
<Products>
    <Product id="01">
        <name>Prouct_a</name>
        <price currency="INR">1200</price>
    </Product>
    <Product id="02">
        <name>Product2</name>
        <price currency="INR">1500</price>
    </Product>
    <Product id="03">
        <name>Product3</name>
        <price currency="INR">2000</price>
    </Product>
    <Product id="5">
        <name>Product5</name>
        <price currency="INR">3000</price>
    </Product>
</Products>
```

D:\NET\8_LINQ_Web_Architec

```
Product names based on id_____________
Prouct_a
Product2
Product3
Product5

Single product with id = 03:
Id = 03, Name = Product3, Price = 2000 INR

Final product list:
Prouct_a
Product2_New
Product3
Product6

Press Enter to exit...
```

products.xml | Output | Error List | Package Manag

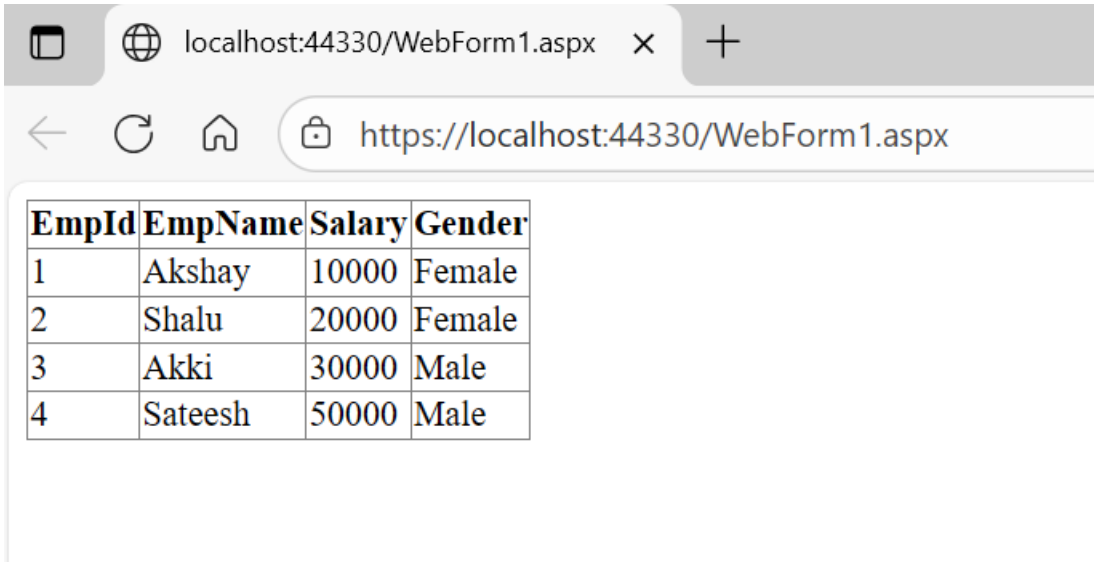
```xml
<?xml version="1.0" encoding="utf-8"?>
<Products>
  <Product id="01">
    <name>Prouct_a</name>
    <price currency="INR">1200</price>
  </Product>
  <Product id="02">
    <name>Product2_New</name>
    <price currency="INR">1800</price>
  </Product>
  <Product id="03">
    <name>Product3</name>
    <price currency="INR">2000</price>
  </Product>
  <Product id="06">
    <name>Product6</name>
    <price currency="INR">4000</price>
  </Product>
</Products>
```

```
using System;
using System.Linq;
using System.Xml.Linq;

namespace Assignment8_Q6
{
   class Program
   {
      // Change this path if needed
      private const string XmlPath = "products.xml";

      static void Main(string[] args)
      {
         // Load the XML file
         XDocument doc = XDocument.Load(XmlPath);

         Console.WriteLine("Product names based on id_____________");

         // READ ALL
         PrintAllProductNames(doc);

         // --- EXAMPLES OF CRUD OPERATIONS ---

         // ADD a new product
         AddProduct(doc, "06", "Product6", 4000, "INR");

         // UPDATE product name & price by id
         UpdateProduct(doc, "02", "Product2_New", 1800);

         // DELETE product by id
         DeleteProduct(doc, "5");

         // READ one product by id
         Console.WriteLine("\nSingle product with id = 03:");
         ReadOneProductById(doc, "03");

         // Save back to the file (optional)
         doc.Save(XmlPath);

         Console.WriteLine("\nFinal product list:");
         PrintAllProductNames(doc);

         Console.WriteLine("\nPress Enter to exit...");
         Console.ReadLine();
      }

      // READ ALL: show names (like sample output)
      private static void PrintAllProductNames(XDocument doc)
      {
         var products =
            from p in doc.Descendants("Product")
            select new
            {
               Id = (string)p.Attribute("id"),
               Name = (string)p.Element("name")
            };

         foreach (var p in products)
         {
            Console.WriteLine(p.Name);
         }
      }

      // ADD: add new <Product> element
      private static void AddProduct(XDocument doc, string id, string name, decimal price, string currency)
      {
```

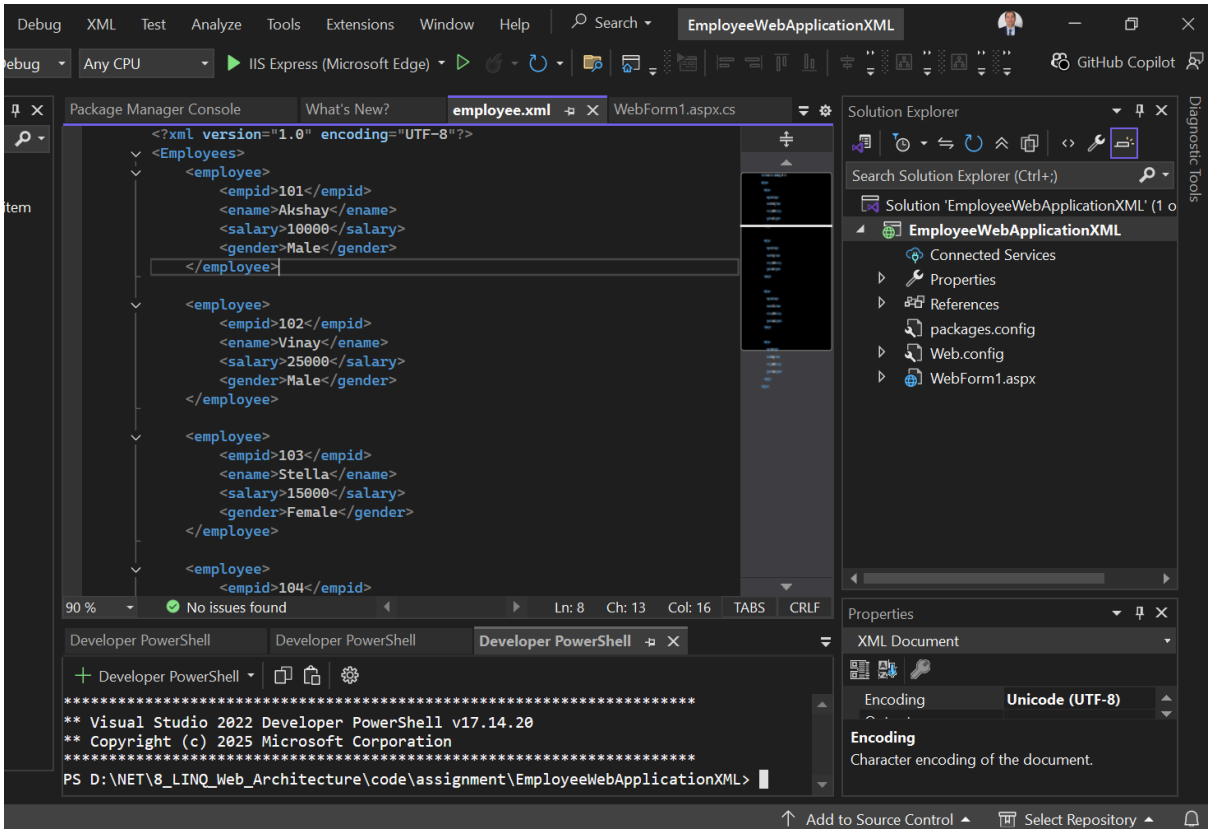
```
        XElement newProduct =
            new XElement("Product",
                new XAttribute("id", id),
                new XElement("name", name),
                new XElement("price",
                    new XAttribute("currency", currency),
                    price.ToString())
            );

        doc.Root.Add(newProduct);
    }

    // DELETE: remove product with matching id
    private static void DeleteProduct(XDocument doc, string id)
    {
        var product =
            (from p in doc.Descendants("Product")
             where (string)p.Attribute("id") == id
             select p).FirstOrDefault();

        if (product != null)
        {
            product.Remove();
        }
    }

    // UPDATE: change name and price for product with given id
    private static void UpdateProduct(XDocument doc, string id, string newName, decimal newPrice)
    {
        var product =
            (from p in doc.Descendants("Product")
             where (string)p.Attribute("id") == id
             select p).FirstOrDefault();

        if (product != null)
        {
            product.Element("name")!.Value = newName;
            product.Element("price")!.Value = newPrice.ToString();
        }
    }

    // READ ONE: print a single product by id
    private static void ReadOneProductById(XDocument doc, string id)
    {
        var product =
            (from p in doc.Descendants("Product")
             where (string)p.Attribute("id") == id
             select p).FirstOrDefault();

        if (product != null)
        {
            string name = (string)product.Element("name");
            string price = (string)product.Element("price");
            string currency = (string)product.Element("price")?.Attribute("currency");

            Console.WriteLine($"Id = {id}, Name = {name}, Price = {price} {currency}");
        }
        else
        {
            Console.WriteLine($"Product with id = {id} not found.");
        }
    }
  }
}
```

QUESTION 7 : Employee record Web application with XML File

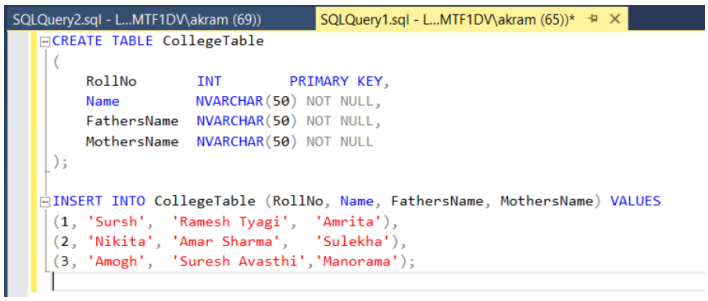
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Xml.Linq;

namespace EmployeeWebApplicationXML
{
   public partial class WebForm1 : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
         if (!Page.IsPostBack)
         {
            // Load the XML file
            XElement doc = XElement.Load(Server.MapPath("Employees.xml"));

            var data =
               from emp in doc.Descendants("employee")
               select new
               {
                  EmpId = (int)emp.Element("empid"),
                  EmpName = (string)emp.Element("empname"),
                  Salary = (int)emp.Element("salary"),
                  Gender = (string)emp.Element("gender")
               };

            GridView1.DataSource = data.ToList();
            GridView1.DataBind();
         }
      }
   }
}
```

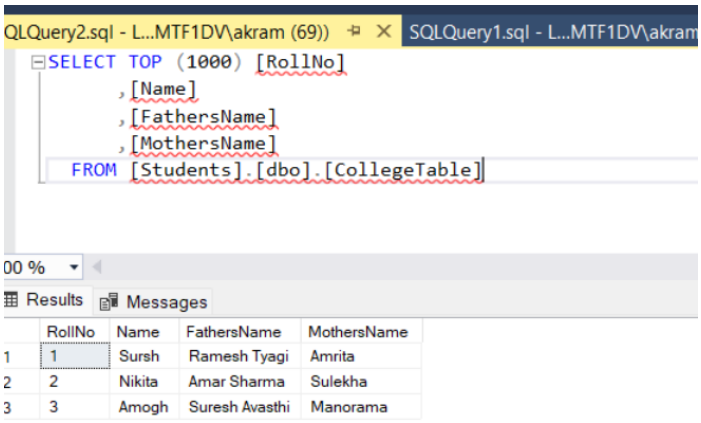


EmpId	EmpName	Salary	Gender
1	Akshay	10000	Female
2	Shalu	20000	Female
3	Akki	30000	Male
4	Sateesh	50000	Male

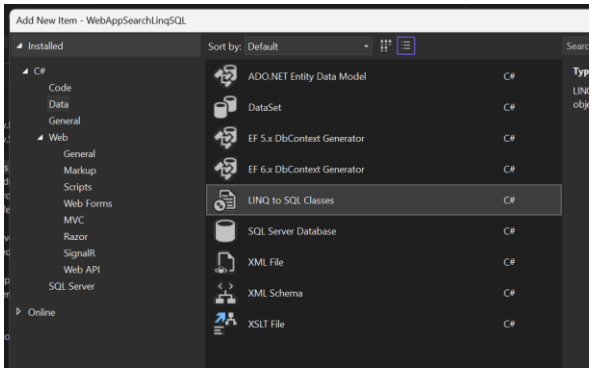

Debug XML Test Analyze Tools Extensions Window Help Search EmployeeWebApplicationXML
Any CPU IIS Express (Microsoft Edge) GitHub Copilot
Package Manager Console What's New? employee.xml WebForm1.aspx.cs
<?xml version="1.0" encoding="UTF-8"?>
<Employees>
<employee>
<empid>101</empid>
<ename>Akshay</ename>
<salary>10000</salary>
<gender>Male</gender>
</employee>
<employee>
<empid>102</empid>
<ename>Vinay</ename>
<salary>25000</salary>
<gender>Male</gender>
</employee>
<employee>
<empid>103</empid>
<ename>Stella</ename>
<salary>15000</salary>
<gender>Female</gender>
</employee>
<employee>
<empid>104</empid>
90 % No issues found Ln: 8 Ch: 13 Col: 16 TABS CRLF
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'EmployeeWebApplicationXML' (1 o
EmployeeWebApplicationXML
Connected Services
Properties
References
packages.config
Web.config
WebForm1.aspx
Properties
XML Document
Encoding Unicode (UTF-8)
Encoding
Character encoding of the document.
Developer PowerShell Developer PowerShell Developer PowerShell
Developer PowerShell
** Visual Studio 2022 Developer PowerShell v17.14.20
** Copyright (c) 2025 Microsoft Corporation
PS D:\NET\8_LINQ_Web_Architecture\code\assignment\EmployeeWebApplicationXML>
Add to Source Control Select Repository
Diagnostic Tools

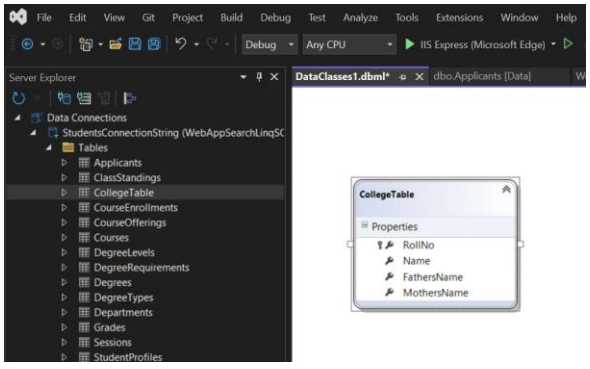
QUESTION 8 : LINQ to SQL

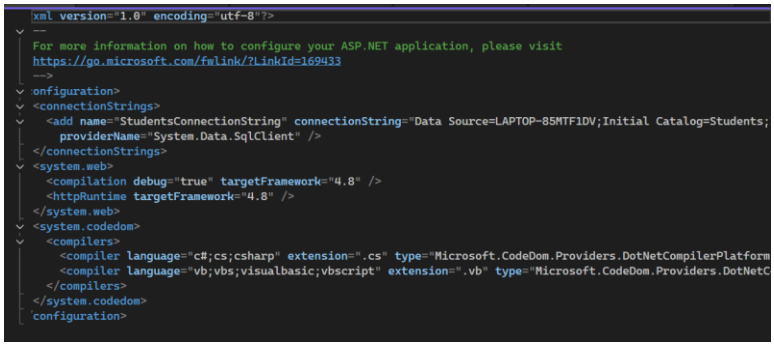


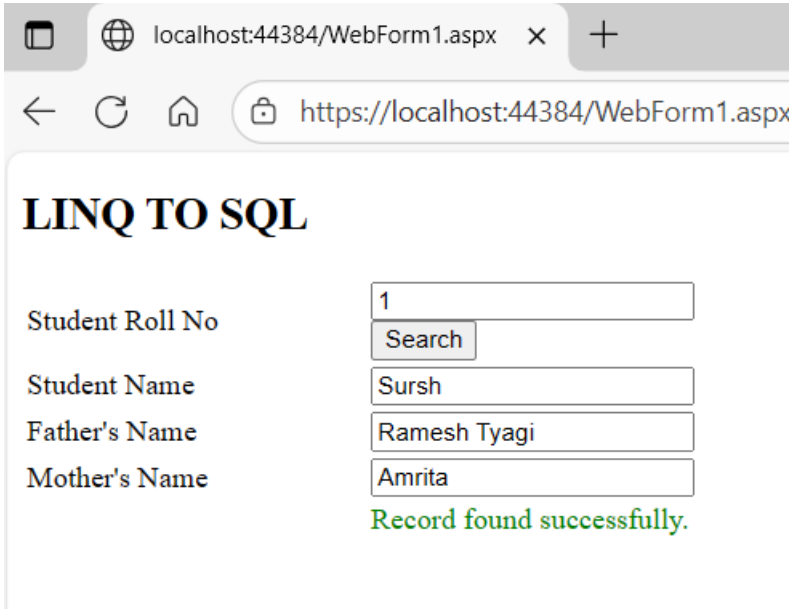


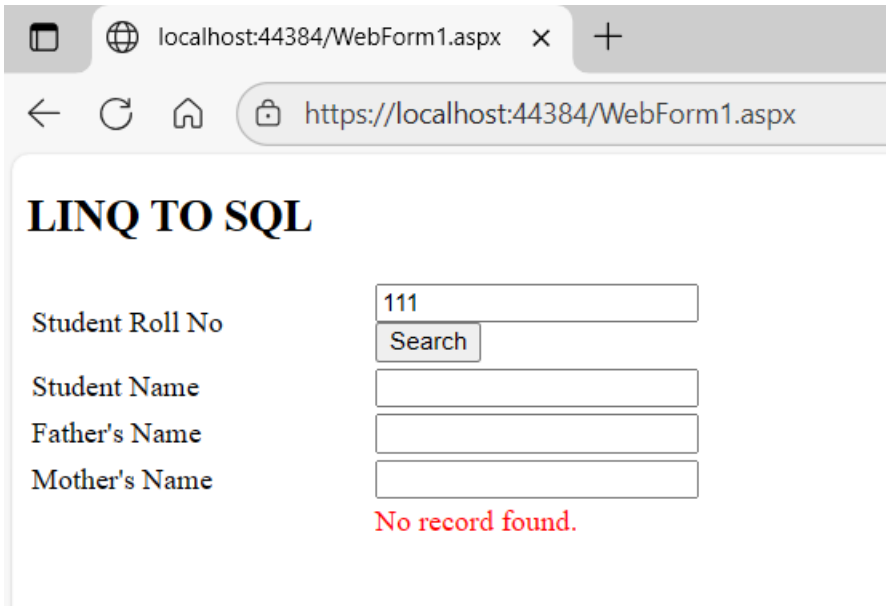
	RollNo	Name	FathersName	MothersName
1	1	Sursh	Ramesh Tyagi	Amrita
2	2	Nikita	Amar Sharma	Sulekha
3	3	Amogh	Suresh Avasthi	Manorama

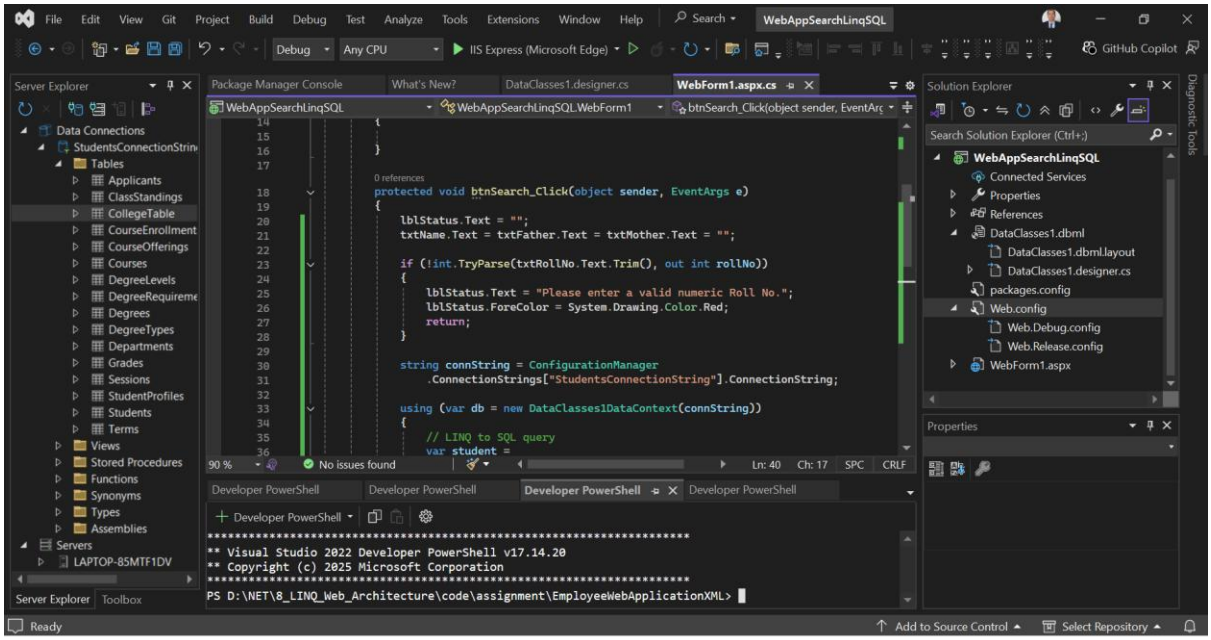





<?xml version="1.0" encoding="utf-8"?>
<!--
For more information on how to configure your ASP.NET application, please visit
https://go.microsoft.com/fwlink/?LinkId=169433
-->
<configuration>
<connectionStrings>
<add name="StudentsConnectionString" connectionString="Data Source=LAPTOP-85MTF1DV;Initial Catalog=Students;
providerName="System.Data.SqlClient" />
</connectionStrings>
<system.web>
<compilation debug="true" targetFramework="4.8" />
<httpRuntime targetFramework="4.8" />
</system.web>
<system.codedom>
<compilers>
<compiler language="c#;cs;csharp" extension=".cs" type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform
<compiler language="vb;vbs;visualbasic;vbscript" extension=".vb" type="Microsoft.CodeDom.Providers.DotNetC
</compilers>
</system.codedom>
</configuration>


localhost:44384/WebForm1.aspx
https://localhost:44384/WebForm1.aspx
LINQ TO SQL
Student Roll No
1
Search
Student Name
Sursh
Father's Name
Ramesh Tyagi
Mother's Name
Amrita
Record found successfully.

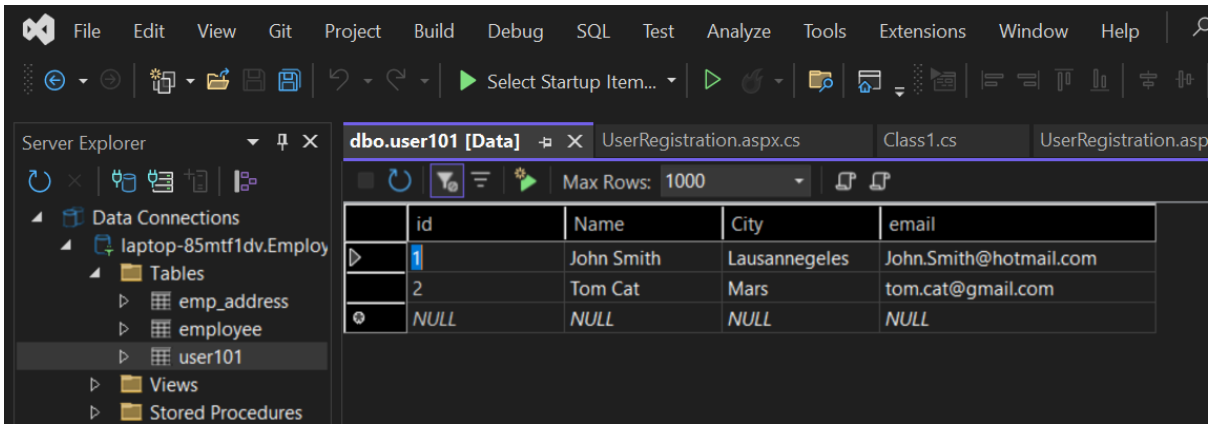

localhost:44384/WebForm1.aspx
https://localhost:44384/WebForm1.aspx
LINQ TO SQL
Student Roll No
111
Search
Student Name
Father's Name
Mother's Name
No record found.

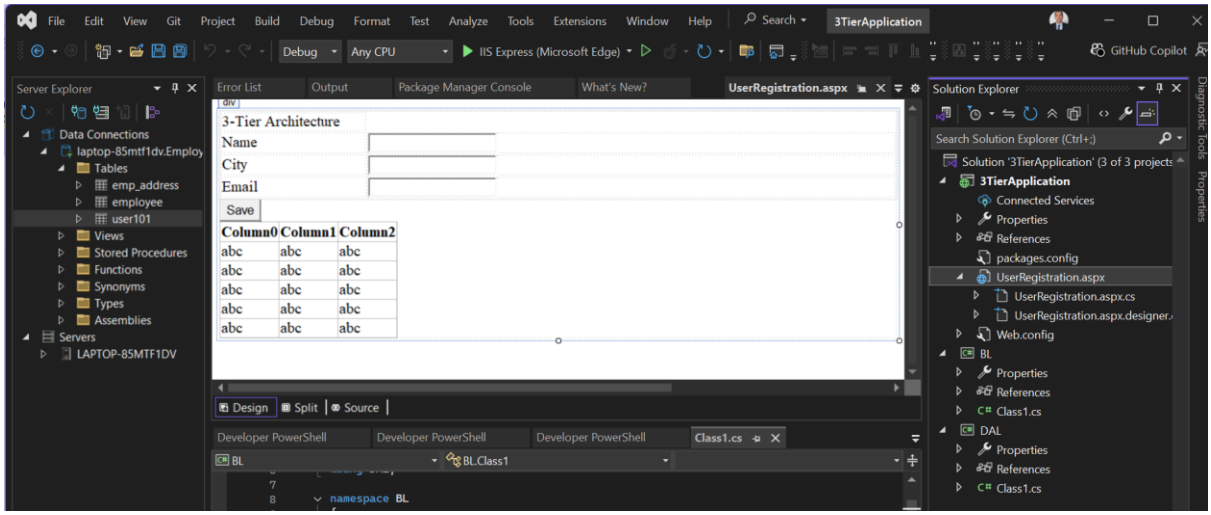


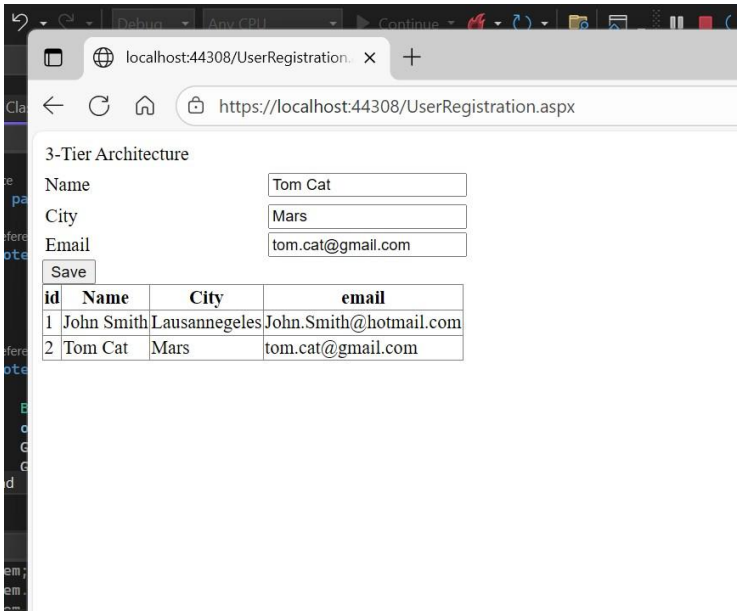

```
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace WebAppSearchLinqSQL
{
   public partial class WebForm1 : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {

      }

      protected void btnSearch_Click(object sender, EventArgs e)
      {
         lblStatus.Text = "";
         txtName.Text = txtFather.Text = txtMother.Text = "";

         if (!int.TryParse(txtRollNo.Text.Trim(), out int rollNo))
         {
            lblStatus.Text = "Please enter a valid numeric Roll No.";
            lblStatus.ForeColor = System.Drawing.Color.Red;
            return;
         }

         string connString = ConfigurationManager
            .ConnectionStrings["StudentsConnectionString"].ConnectionString;

         using (var db = new DataClasses1DataContext(connString))
         {
            // LINQ to SQL query
            var student =
               (from s in db.CollegeTables
                where s.RollNo == rollNo
```

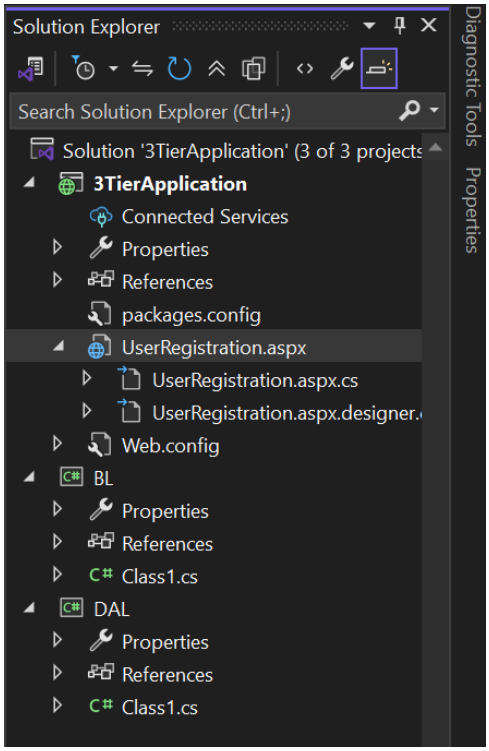
```
                select s).SingleOrDefault();

            if (student == null)
            {
                lblStatus.Text = "No record found.";
                lblStatus.ForeColor = System.Drawing.Color.Red;
            }
            else
            {
                txtName.Text = student.Name;
                txtFather.Text = student.FathersName;
                txtMother.Text = student.MothersName;

                lblStatus.Text = "Record found successfully.";
                lblStatus.ForeColor = System.Drawing.Color.Green;
            }
        }
    }
  }
}
```

QUESTION 9 : 3-tier web application









BL Business Logic

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using DAL;

namespace BL
{
   public class Class1
   {
   }

      public class BLL {

          DL ob1 = new DL();
          public void Inputuser(string name, string city, string email)
          {
              ob1.insertData(name, city, email);
          }

          public object selectUser()
          {
              return ob1.selectData();
          }


      }
}
```

DAL Data Access Layer

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Xml.Linq;


namespace DAL
{
   public class Class1
   {
   }

   public class DL
   {
      SqlConnection con = new SqlConnection(
   @"Data Source=LAPTOP-85MTF1DV;
    Initial Catalog=Employees;
    Integrated Security=True;
    TrustServerCertificate=True");


      public void insertData(string name, string city, string email)
      {
         SqlDataAdapter sda = new SqlDataAdapter("insert into user101 values('"+name+"','"+city+"','"+email+"')", con);
         DataTable dt = new DataTable();
         sda.Fill(dt);

      }

      public object selectData()
      {
         SqlDataAdapter sda = new SqlDataAdapter("select * from user101", con);
         DataTable dt = new DataTable();
         sda.Fill(dt);
         return dt;


      }

   }
}
```

View Layer

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using BL;

namespace _3TierApplication
{
   public partial class WebForm1 : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
```

```
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            BLL ob2 = new BLL();
            ob2.Inputuser(TextBox1.Text, TextBox2.Text, TextBox3.Text);
            GridView1.DataSource = ob2.selectUser();
            GridView1.DataBind();
        }
    }
}
```