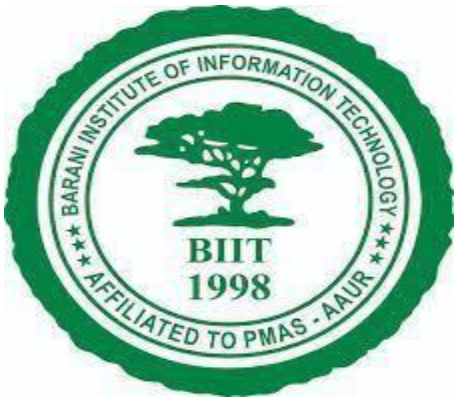

Barani Institute of Information Technology

NAME : Tayyaba Kousar

CLASS : BSCS-4A

REG NO : 2024-Arid-0200

SUBMITTED TO : Sir Aftab

Assignment : 3

```
using System.ComponentModel.DataAnnotations;

public class StudentData
{
   [Key]
   public int ID { get; set; }

   public string ARID_Number { get; set; }

   public string Email { get; set; }
}

using System.Data.Entity;

public class StudentContext : DbContext
{
   public DbSet<StudentData> StudentDatas { get;
set; }
}
using System;
using System.IO;
using System.Text.RegularExpressions;
using System.Linq;

class Program
{
   // Delegates for validation
   delegate bool Validator(string input);
```

```
    static void Main()
    {
        string inputFile = "student_data.txt";
        string invalidFile = "invalid_entries.txt";

        // Lambda expressions for validation
        Validator aridValidator = arid =>
            Regex.IsMatch(arid, @"^\d{4}-arid-\d{4}$");

        Validator emailValidator = email =>
            Regex.IsMatch(email,
@"^[^@\s]+@[^@\s]+\.[^@\s]+$");

        using (var context = new StudentContext())
        using (var invalidWriter = new
StreamWriter(invalidFile))
        {
            foreach (string line in File.ReadLines(inputFile))
            {
                string[] parts = line.Split('|');

                if (parts.Length != 2)
                {
                    invalidWriter.WriteLine(line);
                    continue;
                }

                string arid = parts[0];
                string email = parts[1];
```

```
            if (aridValidator(arid) &&
emailValidator(email))
            {
                context.StudentDatas.Add(new
StudentData
                {
                    ARID_Number = arid,
                    Email = email
                });
            }
            else
            {
                invalidWriter.WriteLine(line);
            }
        }
        context.SaveChanges();
    }

    Console.WriteLine("Data Processing
Completed.");
    RunLinqQueries();
  }

static void RunLinqQueries()
  {
    using (var context = new StudentContext())
    {
      // Task 1: Year-wise Filtering
```

```
        var students2021 = context.StudentDatas
            .Where(s =>
s.ARID_Number.StartsWith("2021"))
            .ToList();

        var students2022 = context.StudentDatas
            .Where(s =>
s.ARID_Number.StartsWith("2022"))
            .ToList();

        // Task 2: Search by Partial ARID or Email
        string searchText = "gmail";
        var searchResults = context.StudentDatas
            .Where(s =>
s.ARID_Number.Contains(searchText)
                    || s.Email.Contains(searchText))
            .ToList();

        // Task 3: Sorting
        var sortedByArid = context.StudentDatas
            .OrderBy(s => s.ARID_Number)
            .ToList();

        var sortedByEmailDesc =
context.StudentDatas
            .OrderByDescending(s => s.Email)
            .ToList();

        // Task 4: Count-Based Queries
```

```
        var studentsPerYear = context.StudentDatas
            .GroupBy(s => s.ARID_Number.Substring(0,
4))
            .Select(g => new
            {
                Year = g.Key,
                Count = g.Count()
            }).ToList();

        int gmailCount = context.StudentDatas
            .Count(s =>
s.Email.EndsWith("@gmail.com"));

        int nonGmailCount = context.StudentDatas
            .Count(s
=> !s.Email.EndsWith("@gmail.com"));

        // Task 5: Email Domain Analysis
        var emailDomains = context.StudentDatas
            .Select(s => s.Email.Split('@')[1])
            .Distinct()
            .ToList();

        var studentsPerDomain =
context.StudentDatas
            .GroupBy(s => s.Email.Split('@')[1])
            .Select(g => new
            {
                Domain = g.Key,
```

```
                Count = g.Count()
            }).ToList();

        Console.WriteLine("LINQ Queries Executed
Successfully.");
    }
  }
}
```