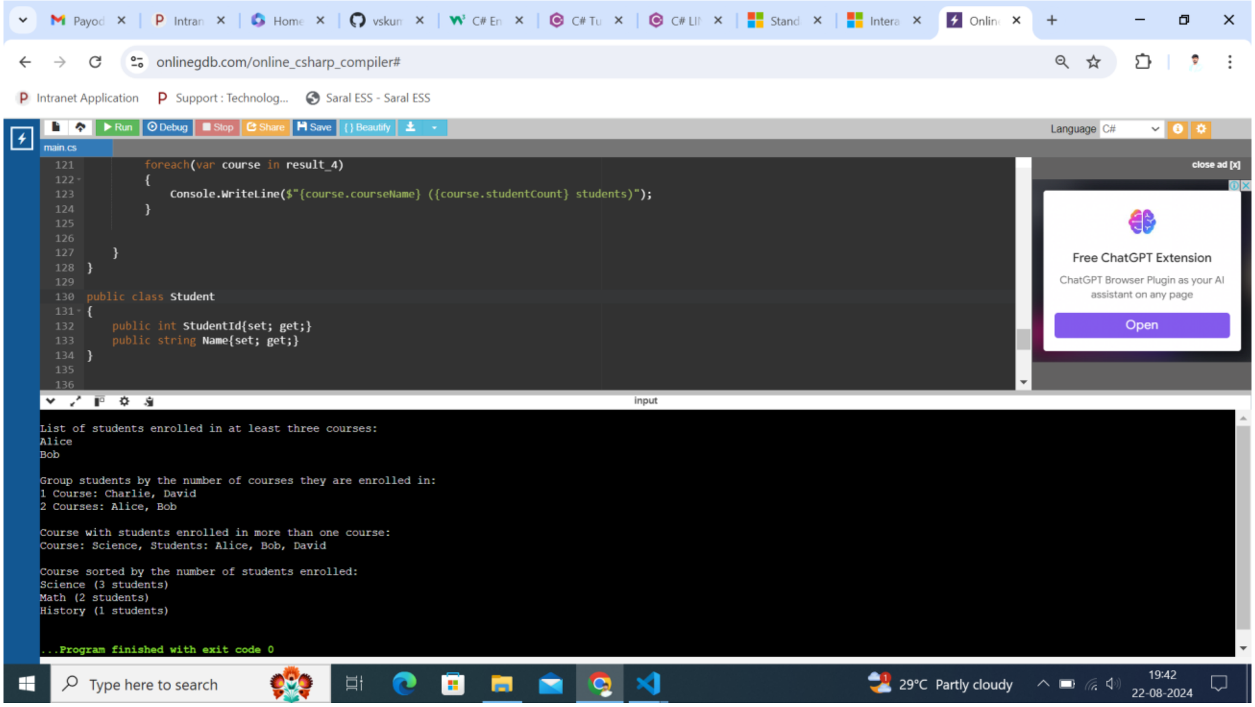
4 – DAY – TASK (22-08-2024)

1. **LINQ Program Code:**

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

public class Program
{


public static void Main()
{
var students = new List<Student>
{
new Student { StudentId = 1, Name = "Alice" },
new Student { StudentId = 2, Name = "Bob" },
new Student { StudentId = 3, Name = "Charlie" },
new Student { StudentId = 4, Name = "David" }
};

var courses = new List<Course>
{
new Course { CourseId = 1, Title = "Math" },
new Course { CourseId = 2, Title = "Science" },
new Course { CourseId = 3, Title = "History" }
};

var enrollments = new List<Enrollment>
{
new Enrollment { StudentId = 1, CourseId = 1 },
new Enrollment { StudentId = 1, CourseId = 2 },
new Enrollment { StudentId = 2, CourseId = 2 },
new Enrollment { StudentId = 2, CourseId = 3 },
new Enrollment { StudentId = 3, CourseId = 1 },
new Enrollment { StudentId = 4, CourseId = 2 }
};


var result_1 = enrollments
        .GroupBy(e => e.StudentId)
        .Where(g => g.Count() >= 2)
        .Select(g => students.FirstOrDefault(s => s.StudentId == g.Key).Name);

    Console.WriteLine();
Console.WriteLine("List of students enrolled in at least three courses:");
foreach (var s in result_1)
{
Console.WriteLine(s);
}


var result_2 = enrollments
        .GroupBy(e => e.StudentId)
        .Select(g => new {
          StudentName = students.FirstOrDefault(s => s.StudentId == g.Key).Name,
          CourseCount = g.Count()
        })
        .OrderBy(e => e.CourseCount)
        .GroupBy(e => e.CourseCount)
        .Select(g => new {
          CourseCount = g.Key,
          StudentNames = string.Join(", ", g.Select(x => x.StudentName))
        });

    Console.WriteLine();
Console.WriteLine("Group students by the number of courses they are enrolled
in:");
foreach (var group in result_2)
{
if(group.CourseCount == 1)
Console.WriteLine($"{group.CourseCount} Course: {group.StudentNames}");
else
```

```
Console.WriteLine($"{group.CourseCount} Courses: {group.StudentNames}");

}

var result_3 = enrollments
        .Join(students,
        e => e.StudentId,
        s => s.StudentId,
        (e, s) => new {
          Enrollment = e, Student = s
        })
        .Join(courses,
        e => e.Enrollment.CourseId,
        c => c.CourseId,
        (e, c) => new {
          StudentName = e.Student.Name,
          CourseName = c.Title
        })
        .GroupBy(e => e.CourseName)
        .Where(e => e.Count() > 2)
        .Select(g => new {
          CourseName = g.Key,
          StudentNames = string.Join(", ", g.Select(x => x.StudentName))
        });

Console.WriteLine();
Console.WriteLine("Course with students enrolled in more than one course:");

foreach (var group in result_3)
{
Console.WriteLine($"Course: {group.CourseName}, Students:
{group.StudentNames}");
}

var result_4 = enrollments
        .Join(courses,
        e => e.CourseId,
        c => c.CourseId,
            (e, c) => new {
              studentId = e.StudentId,
              courseName = c.Title
            })
            .GroupBy(e => e.courseName)
            .OrderByDescending(e => e.Count())
            .Select(g => new {
               courseName = g.Key,
               studentCount = g.Count()
            });

 Console.WriteLine();
 Console.WriteLine("Course sorted by the number of students enrolled:");
 foreach(var course in result_4)
 {
    Console.WriteLine($"{course.courseName} ({course.studentCount} students)");
 }



}
}

public class Student
{
public int StudentId{set; get;}
public string Name{set; get;}
}



public class Course
{
public int CourseId{set; get;}
public string Title{set; get;}
}

public class Enrollment
{
public int StudentId{set; get;}
public int CourseId{set; get;}
}
```

Output:



```
            foreach(var course in result_4)
            {
                Console.WriteLine($"{course.courseName} ({course.studentCount} students)");
            }


    }
}

public class Student
{
    public int StudentId{set; get;}
    public string Name{set; get;}
}
```

```
List of students enrolled in at least three courses:
Alice
Bob

Group students by the number of courses they are enrolled in:
1 Course: Charlie, David
2 Courses: Alice, Bob

Course with students enrolled in more than one course:
Course: Science, Students: Alice, Bob, David

Course sorted by the number of students enrolled:
Science (3 students)
Math (2 students)
History (1 students)


...Program finished with exit code 0
```