

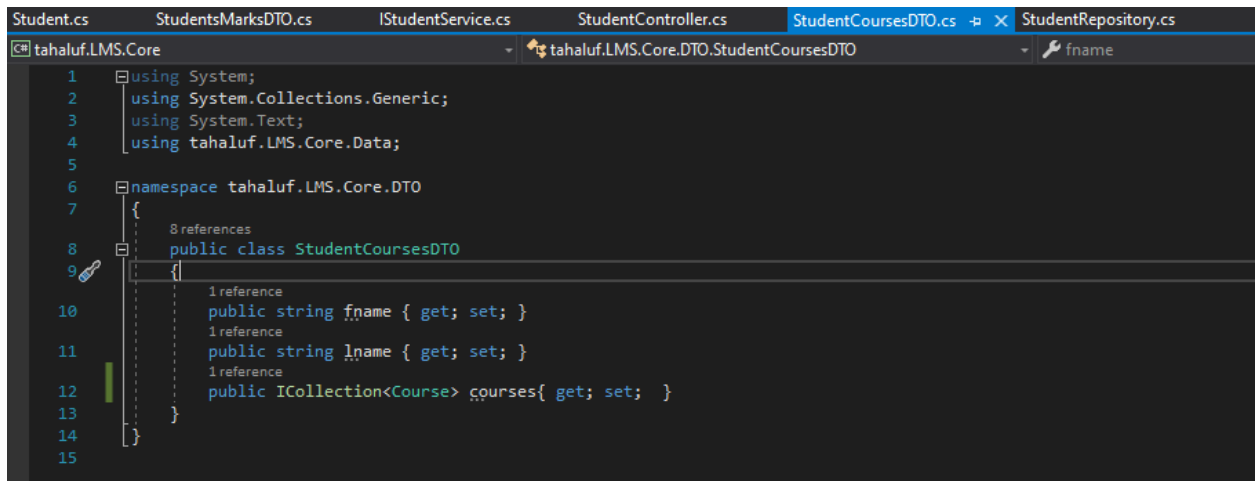
*SQL Code

Worksheet	Query Builder
<pre>create or replace PACKAGE Student_Package AS PROCEDURE Find (StudentId1 NUMBER); PROCEDURE Marks; PROCEDURE GetCoursesList(StudentId1 NUMBER); END Student_Package; create or replace PACKAGE Body Student_Package AS PROCEDURE Find (StudentId1 NUMBER) AS c_all SYS_REFCURSOR; BEGIN OPEN c_all FOR SELECT * FROM student WHERE StudentId = StudentId1; DBMS_SQL.RETURN_RESULT(c_all); END; PROCEDURE Marks AS c_all SYS_REFCURSOR; BEGIN OPEN c_all FOR SELECT MAX(Mark) "Maximum", MIN(Mark) "Minimum", COUNT(Mark) "Count", Round(AVG(Mark), 2) "Average" FROM student ; DBMS_SQL.RETURN_RESULT(c_all); END Marks; PROCEDURE GetCoursesList(StudentId1 Number) AS c_all SYS_REFCURSOR; BEGIN OPEN c_all FOR SELECT Course.CourseId ,CourseName,Price,StartDate ,EndDate,ImageName FROM Course JOIN StudentCourse ON Course.CourseId = StudentCourse.CourseId WHERE StudentCourse.StudentId = StudentId1; DBMS_SQL.RETURN_RESULT(c_all); END GetCoursesList; END Student_Package;</pre>	

2.2.Create an API to retrieve student name and his courses.

*Visual Studio

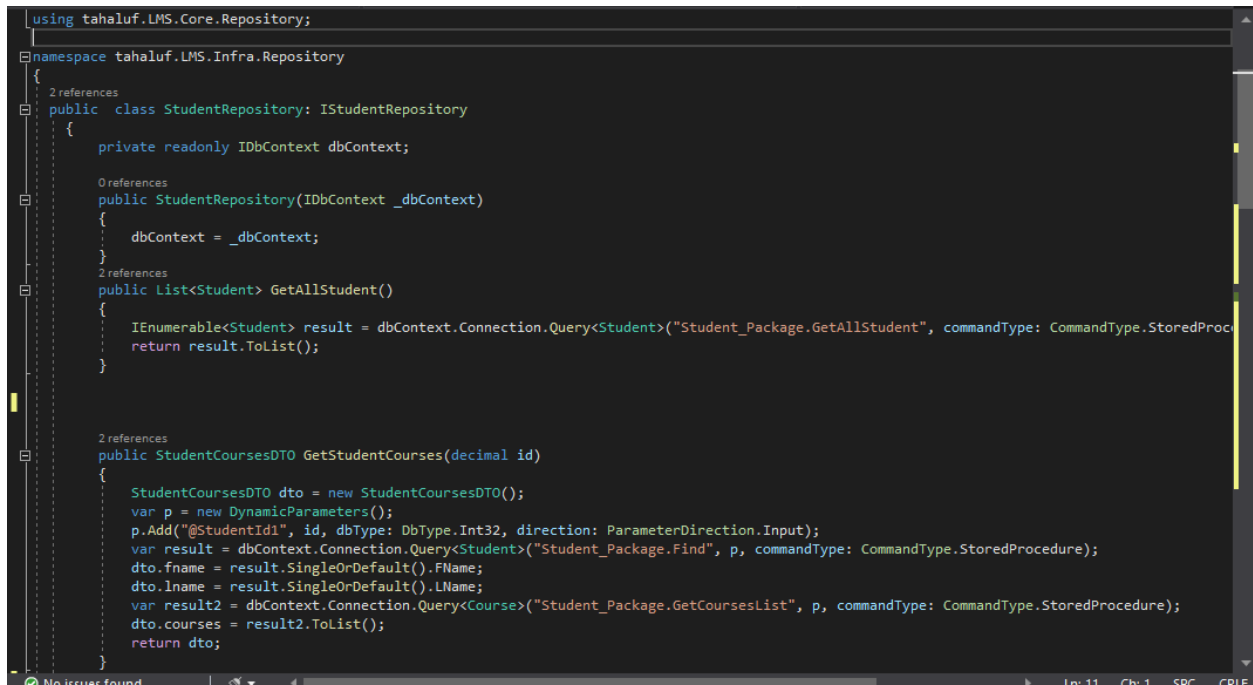
A: DTO



The screenshot shows the Visual Studio IDE with the 'StudentCoursesDTO.cs' file open. The code defines a public class 'StudentCoursesDTO' within the 'tahaluf.LMS.Core.DTO' namespace. It includes several using statements at the top: 'using System;', 'using System.Collections.Generic;', 'using System.Text;', and 'using tahaluf.LMS.Core.Data;'. The class has three public properties: 'fname' of type 'string', 'lname' of type 'string', and 'courses' of type 'ICollection<Course>'. Each property has a getter and a setter. The 'courses' property is annotated with a '1 reference' label. The file explorer on the left shows the project structure, including 'tahaluf.LMS.Core' and 'tahaluf.LMS.Core.DTO'.

```
1 using System;
2 using System.Collections.Generic;
3 using System.Text;
4 using tahaluf.LMS.Core.Data;
5
6 namespace tahaluf.LMS.Core.DTO
7 {
8     public class StudentCoursesDTO
9     {
10         public string fname { get; set; }
11         public string lname { get; set; }
12         public ICollection<Course> courses { get; set; }
13     }
14 }
15
```

B: Repository



The screenshot shows the Visual Studio IDE with the 'StudentRepository.cs' file open. The code defines a public class 'StudentRepository' within the 'tahaluf.LMS.Infra.Repository' namespace. It implements the 'IStudentRepository' interface. The class has a private readonly 'DbContext' property 'dbContext'. It has three public methods: 'StudentRepository(DbContext _dbContext)', 'List<Student> GetAllStudent()', and 'StudentCoursesDTO GetStudentCourses(decimal id)'. The 'GetAllStudent()' method uses 'dbContext.Connection.Query<Student>' to execute a stored procedure 'Student_Package.GetAllStudent'. The 'GetStudentCourses()' method uses 'dbContext.Connection.Query<Student>' and 'dbContext.Connection.Query<Course>' to execute stored procedures 'Student_Package.Find' and 'Student_Package.GetCoursesList' respectively. The file explorer on the left shows the project structure, including 'tahaluf.LMS.Infra.Repository'.

```
using tahaluf.LMS.Core.Repository;

namespace tahaluf.LMS.Infra.Repository
{
    public class StudentRepository: IStudentRepository
    {
        private readonly DbContext dbContext;

        public StudentRepository(DbContext _dbContext)
        {
            dbContext = _dbContext;
        }

        public List<Student> GetAllStudent()
        {
            IEnumerable<Student> result = dbContext.Connection.Query<Student>("Student_Package.GetAllStudent", commandType: CommandType.StoredProcedure);
            return result.ToList();
        }

        public StudentCoursesDTO GetStudentCourses(decimal id)
        {
            StudentCoursesDTO dto = new StudentCoursesDTO();
            var p = new DynamicParameters();
            p.Add("@StudentId1", id, DbType.Int32, direction: ParameterDirection.Input);
            var result = dbContext.Connection.Query<Student>("Student_Package.Find", p, commandType: CommandType.StoredProcedure);
            dto.fname = result.SingleOrDefault().FName;
            dto.lname = result.SingleOrDefault().LName;
            var result2 = dbContext.Connection.Query<Course>("Student_Package.GetCoursesList", p, commandType: CommandType.StoredProcedure);
            dto.courses = result2.ToList();
            return dto;
        }
    }
}
```

C: Controller

```
namespace tahaluf.LMS.API.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    1 reference
    public class StudentController : Controller
    {
        private readonly IStudentService studentService;
        0 references
        public StudentController(IStudentService _studentService)
        {
            studentService = _studentService;
        }

        [HttpGet("GetStudentCourses/{id}")]
        [ProducesResponseType(typeof(StudentCoursesDTO), StatusCodes.Status200OK)]
        [ProducesResponseType(StatusCodes.Status400BadRequest)]
        0 references
        public StudentCoursesDTO GetStudentCourses(decimal id)
        {
            return studentService.GetStudentCourses(id);
        }
    }
}
```

*Postman

The image displays two screenshots of the Postman API client interface, showing a GET request to `https://localhost:44336/api/Student/GetStudentCourses/1` and its corresponding JSON response.

Top Screenshot:

- Request:** GET `https://localhost:44336/api/Student/GetStudentCourses/1`
- Response Status:** 200 OK, Time: 4.21 s, Size: 799 B
- Response Body (JSON):**

```
{  "fname": "Amal",  "lname": "Hayajneh",  "courses": [    {      "courseId": 41,      "courseName": "DataBase",      "price": 250,      "startDate": "2022-01-28T18:53:16",      "endDate": "2022-01-08T18:56:00",      "imageName": "DB",      "books": null,      "studentCourses": null,      "teacherCourses": null    },    {      "courseId": 42,      "courseName": "MVC",      "price": 150,      "startDate": "2022-01-12T18:56:40",      "endDate": "2022-04-30T18:56:47",      "imageName": "MVC",      "books": null,      "studentCourses": null,      "teacherCourses": null    },    {      "courseId": 44,      "courseName": "HTML",      "price": 100,      "startDate": "2021-01-13T18:57:53",      "endDate": "2022-01-29T18:58:06"    }  ]}
```

Bottom Screenshot:

- Request:** GET `https://localhost:44336/api/Student/GetStudentCourses/1`
- Response Status:** 200 OK, Time: 4.21 s, Size: 799 B
- Response Body (JSON):** (This screenshot shows the same JSON response as the top one, but with line numbers 16-32 visible on the left side of the editor.)

Ex2

https://localhost:44336/api/Student/GetStudentCourses/2

GET https://localhost:44336/api/Student/GetStudentCourses/2

Status: 200 OK Time: 828 ms Size: 607 B

```
2  "fname": "Laith",
3  "lname": "Hayajneh",
4  "courses": [
5    {
6      "courseId": 43,
7      "courseName": "Angular",
8      "price": 400,
9      "startDate": "2022-01-07T18:57:29",
10     "endDate": "2022-01-08T18:57:34",
11     "imageName": "Angular",
12     "books": null,
13     "studentCourses": null,
14     "teacherCourses": null
15   },
16   {
17     "courseId": 42,
18     "courseName": "MVC",
19     "price": 150,
20     "startDate": "2022-01-12T18:56:40",
21     "endDate": "2022-04-30T18:56:47",
22     "imageName": "MVC",
23     "books": null,
24     "studentCourses": null,
25     "teacherCourses": null
26   }
27 ]
```

https://localhost:44336/api/Student/GetStudentCourses/2

GET https://localhost:44336/api/Student/GetStudentCourses/2

Status: 200 OK Time: 828 ms Size: 607 B

```
12     "books": null,
13     "studentCourses": null,
14     "teacherCourses": null
15   },
16   {
17     "courseId": 42,
18     "courseName": "MVC",
19     "price": 150,
20     "startDate": "2022-01-12T18:56:40",
21     "endDate": "2022-04-30T18:56:47",
22     "imageName": "MVC",
23     "books": null,
24     "studentCourses": null,
25     "teacherCourses": null
26   }
27 ]
```

2.Create an API to calculate the maximum, minimum, count and average of marks (student table).

A:DTO

```
StudentController.cs StudentCoursesDTO.cs Student.cs StudentService.cs StudentRepository.cs StudentsMarksDTO.cs X IStudentService.cs
tahaluf.LMS.Core
1 using System;
2 using System.Collections.Generic;
3 using System.Text;
4
5 namespace tahaluf.LMS.Core.DTO
6 {
7     8 references
8     public class StudentsMarksDTO
9     {
10         0 references
11         public decimal Maximum { get; set; }
12         0 references
13         public decimal Minimum { get; set; }
14         0 references
15         public decimal Count { get; set; }
16         0 references
17         public double Average { get; set; }
18     }
19 }
```

B: Repository

```
tahaluf.LMS.Infra
8 using tahaluf.LMS.Core.Data;
9 using tahaluf.LMS.Core.DTO;
10 using tahaluf.LMS.Core.Repository;
11
12 namespace tahaluf.LMS.Infra.Repository
13 {
14     2 references
15     public class StudentRepository: IStudentRepository
16     {
17         private readonly IDbContext dbContext;
18
19         0 references
20         public StudentRepository(IDbContext _dbContext)
21         {
22             dbContext = _dbContext;
23         }
24
25         2 references
26         public List<Student> GetAllStudent()
27         {
28             IEnumerable<Student> result = dbContext.Connection.Query<Student>("Student_Package.GetAllStudent", commandType: CommandType.Stor
29             return result.ToList();
30         }
31
32         2 references
33         public List<StudentsMarksDTO> GetMarks()
34         {
35             IEnumerable<StudentsMarksDTO> result = dbContext.Connection.Query<StudentsMarksDTO>
36             ("Student_Package.Marks", commandType: CommandType.StoredProcedure).ToList();
37
38             return result.ToList();
39         }
40     }
41 }
```

C: Controller

```
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using tahaluf.LMS.Core.DTO;
using tahaluf.LMS.Core.Service;

namespace tahaluf.LMS.API.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class StudentController : Controller
    {
        private readonly IStudentService studentService;

        public StudentController(IStudentService _studentService)
        {
            studentService = _studentService;
        }

        [HttpGet("GetMarks")]
        [ProducesResponseType(typeof(StudentsMarksDTO), StatusCodes.Status200OK)]
        [ProducesResponseType(StatusCodes.Status400BadRequest)]
        public List<StudentsMarksDTO> GetMarks()
        {
            return studentService.GetMarks();
        }
    }
}
```

*Postman

https://localhost:44336/api/Student/GetMarks

GET https://localhost:44336/api/Student/GetMarks

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 17.60 s Size: 242 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "maximum": 99.9,
4     "minimum": 70.9,
5     "count": 11,
6     "average": 90.31
7   }
8 }
```

Runner Trash

9:23 PM

