Name :- Nisarg Amlani
Roll : Ce001
Id : 22ceueg082
Lab : 10

=> Code of Student Model (student.cs)

```
namespace Lab10.Models;
using System.ComponentModel.DataAnnotations;


public class Student
{
    [Key]
    public int StuentId { get; set; }
    public string Name { get; set; }
    public DateTime AdmissionDate { get; set; }
    public string AdmissionBranch { get; set; }

}
```

=> Code Student Admission Context (StudentAdmission.cs)

```
using Microsoft.EntityFrameworkCore;

namespace Lab10.Models;

public class StudentAdmissionContext : DbContext
{
    public StudentAdmissionContext(DbContextOptions options) :
base(options)
    {
    }

    public DbSet<Student> Students { get; set; }

    protected override void OnModelCreating(ModelBuilder
modelBuilder)
    {
        base.OnModelCreating(modelBuilder);

        modelBuilder.Entity<Student>().HasData(
            new Student
            {
                StuentId = 1,
                Name = "nisarg",
                AdmissionDate = new DateTime(2025, 2, 20),
                AdmissionBranch = "Computer Engineering"
            }
            ,
            new Student
            {
                StuentId = 2,
                Name = "neha",
                AdmissionDate = new DateTime(2025, 2, 20),
                AdmissionBranch = "Computer Engineering"
            }
        );
    }
}
```

=> Code of Interface Student Admission (IStudentAdmissionRepository.cs)

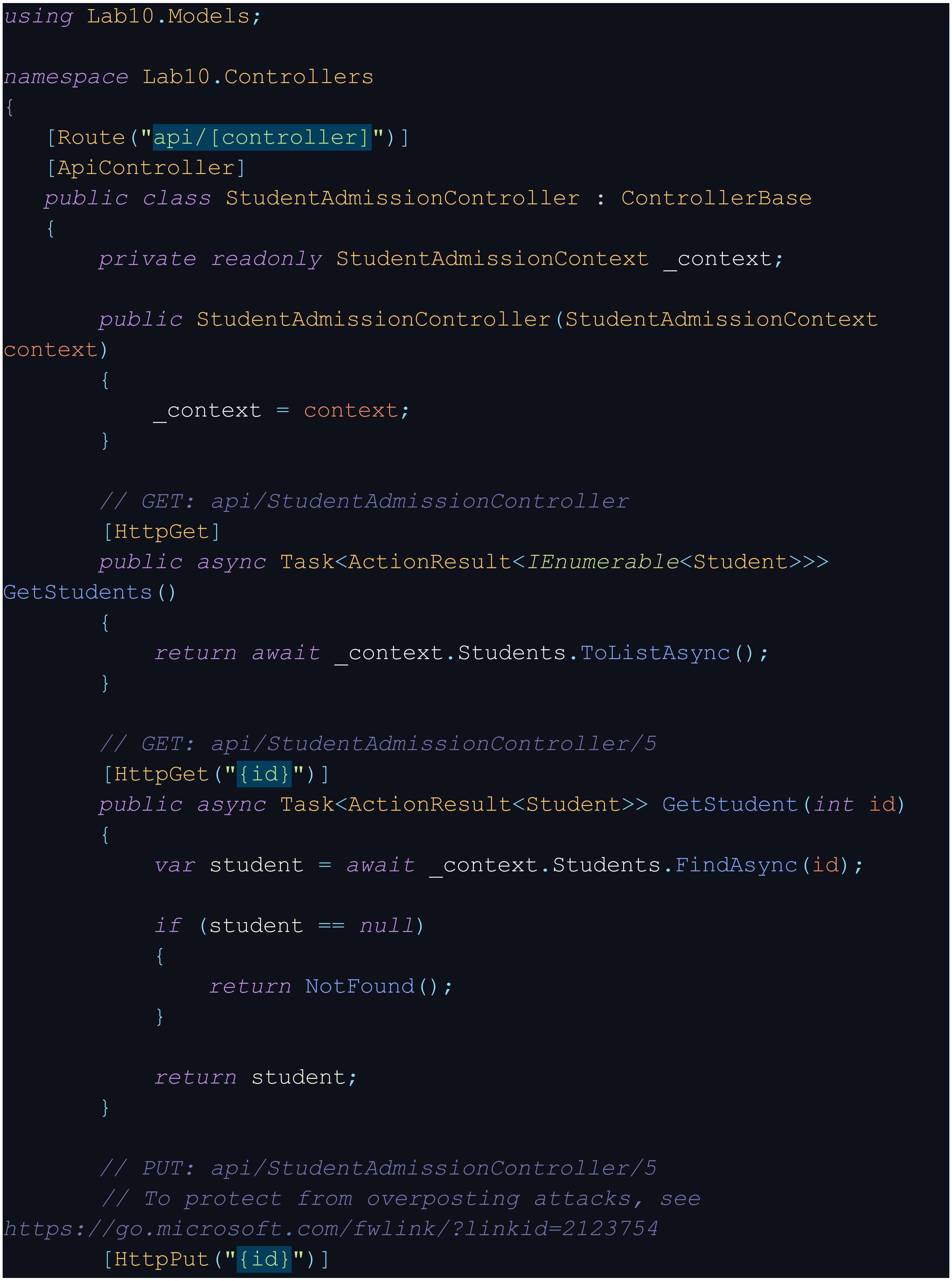
```
namespace Lab10.Repository;
using Lab10.Models;

public interface IStudentAdmissionRepository
{
    IEnumerable<Student> GetStudents();

    Student GetStudent(int id);
    void InsertStudent(Student student);

    void UpdateStudent(Student student);
    void DeleteStudent(int id);

    void Save();


}
```

=> Code of Implementation Student Admission (StudentAdmissionRepository.cs)

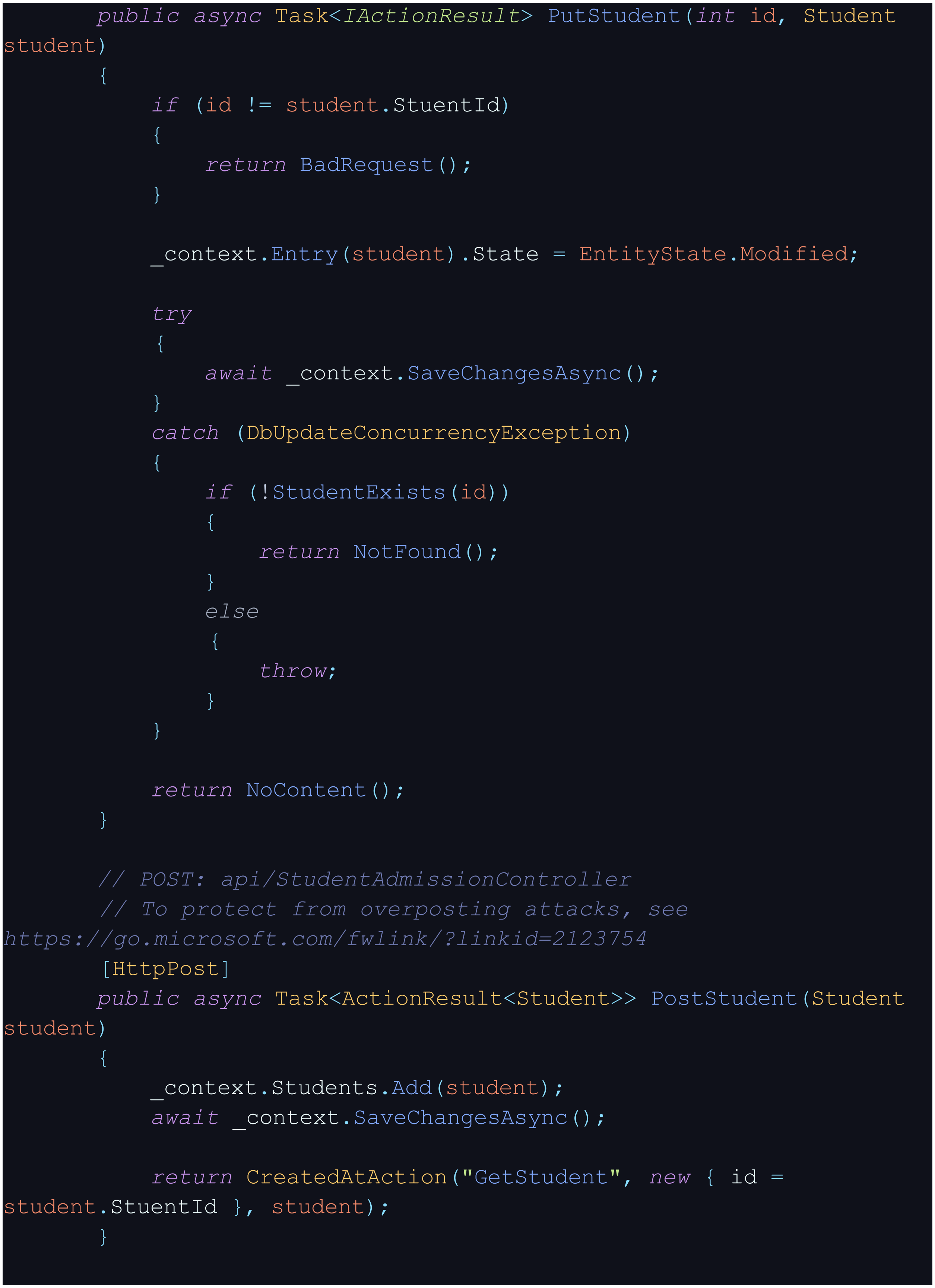
```
namespace Lab10.Repository;
using Lab10.Models;

public class StudentAdmissionRepository :
IStudentAdmissionRepository
{
    private readonly StudentAdmissionContext _context;
    public StudentAdmissionRepository(StudentAdmissionContext context)
{
        _context = context;
    }

    public void DeleteStudent(int id)
    {
        var student = _context.Students.Find(id);
        _context.Students.Remove(student);
        Save();
    }

    public Student GetStudent(int id)
    {
        return _context.Students.Find(id);
    }
    public IEnumerable<Student> GetStudents()
    {
        return _context.Students.ToList();
    }

    public void InsertStudent(Student student)
    {
        _context.Add(student);
        Save();
    }

    public void Save()
    {
        _context.SaveChanges();

    }

    public void UpdateStudent(Student student)
    {
        _context.Entry(student).State =
Microsoft.EntityFrameworkCore.EntityState.Modified;
        Save();

    }

}
```

=> Code of Student Admission Controller (StudentAdmissionController.cs)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
```



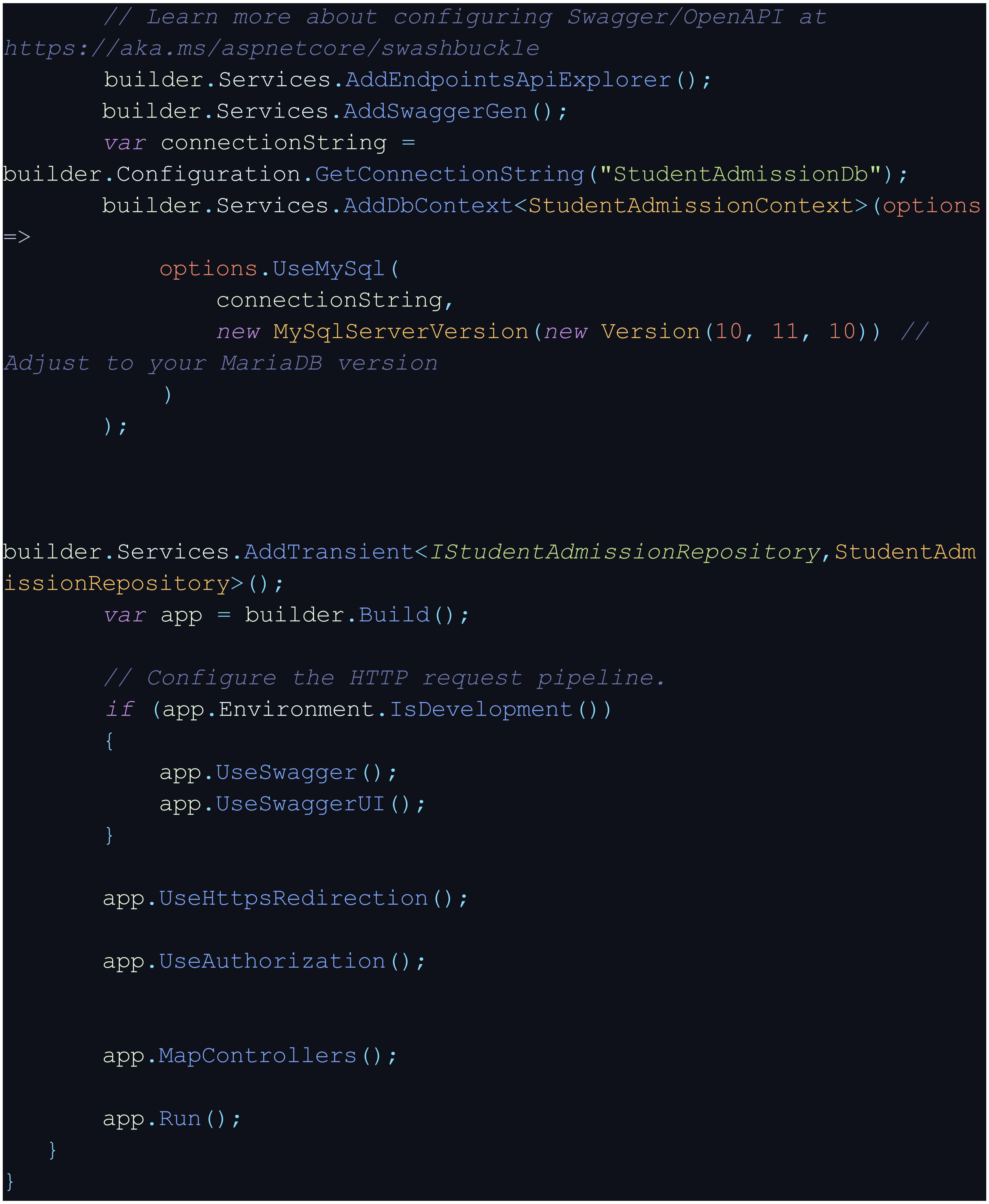
```
using Lab10.Models;

namespace Lab10.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class StudentAdmissionController : ControllerBase
    {
        private readonly StudentAdmissionContext _context;

        public StudentAdmissionController(StudentAdmissionContext
context)
        {
            _context = context;
        }

        // GET: api/StudentAdmissionController
        [HttpGet]
        public async Task<ActionResult<IEnumerable<Student>>>
GetStudents()
        {
            return await _context.Students.ToListAsync();
        }

        // GET: api/StudentAdmissionController/5
        [HttpGet("{id}")]
        public async Task<ActionResult<Student>> GetStudent(int id)
        {
            var student = await _context.Students.FindAsync(id);

            if (student == null)
            {
                return NotFound();
            }

            return student;
        }

        // PUT: api/StudentAdmissionController/5
        // To protect from overposting attacks, see
https://go.microsoft.com/fwlink/?linkid=2123754
        [HttpPut("{id}")]
```



```
        public async Task<IActionResult> PutStudent(int id, Student
student)
        {
            if (id != student.StuentId)
            {
                return BadRequest();
            }

            _context.Entry(student).State = EntityState.Modified;

            try
            {
                await _context.SaveChangesAsync();
            }
            catch (DbUpdateConcurrencyException)
            {
                if (!StudentExists(id))
                {
                    return NotFound();
                }
                else
                {
                    throw;
                }
            }

            return NoContent();
        }

        // POST: api/StudentAdmissionController
        // To protect from overposting attacks, see
https://go.microsoft.com/fwlink/?linkid=2123754
        [HttpPost]
        public async Task<ActionResult<Student>> PostStudent(Student
student)
        {
            _context.Students.Add(student);
            await _context.SaveChangesAsync();

            return CreatedAtAction("GetStudent", new { id =
student.StuentId }, student);
        }
        // DELETE: api/StudentAdmissionController/5
        [HttpDelete("{id}")]
        public async Task<IActionResult> DeleteStudent(int id)
        {
            var student = await _context.Students.FindAsync(id);
            if (student == null)
            {
                return NotFound();
            }

            _context.Students.Remove(student);
            await _context.SaveChangesAsync();

            return NoContent();
        }

        private bool StudentExists(int id)
        {
            return _context.Students.Any(e => e.StuentId == id);
        }
    }
}
```

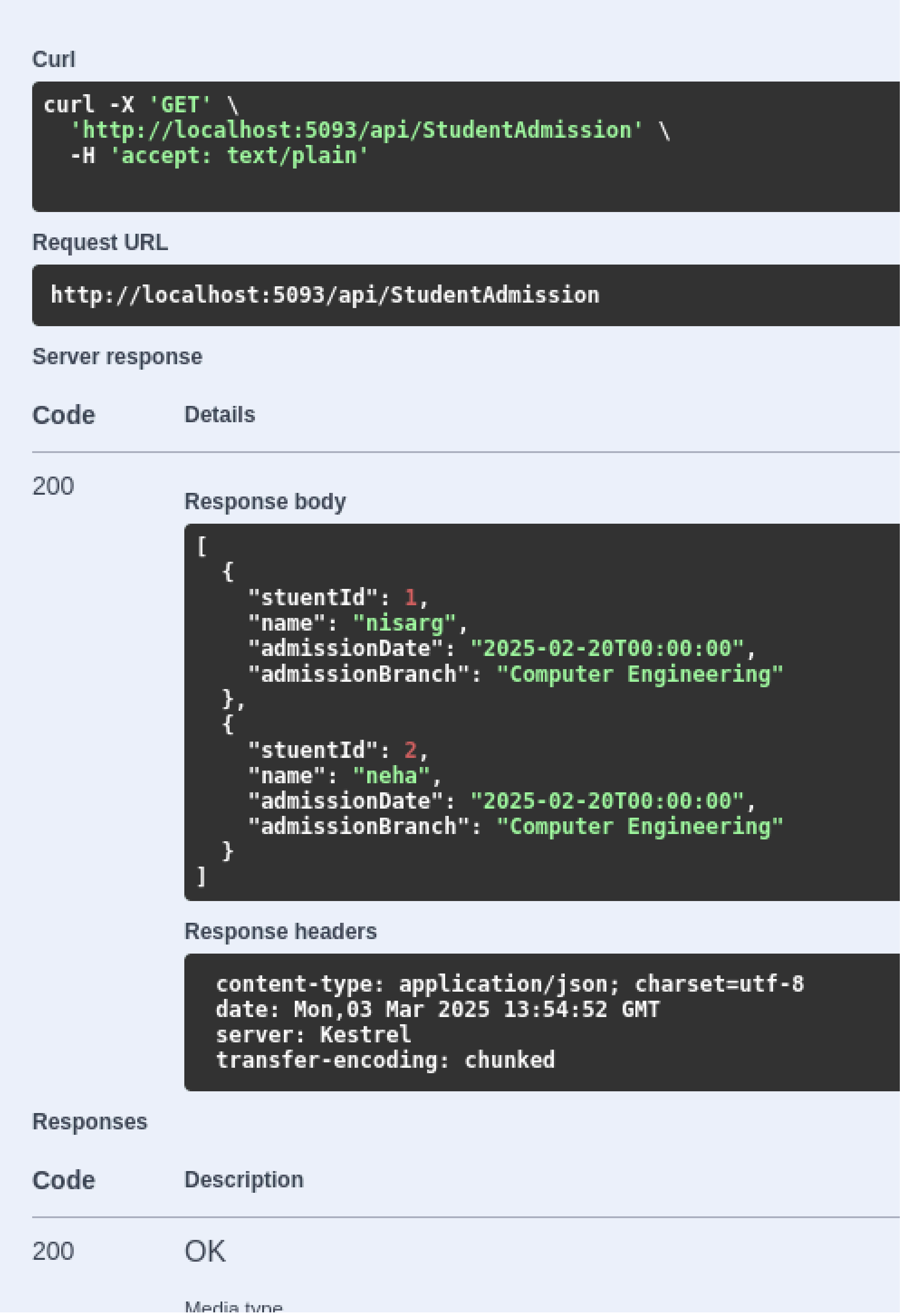
=> Code for Program.cs

```
using Microsoft.EntityFrameworkCore;

namespace Lab10;
using Lab10.Models;
using Lab10.Repository;

public class Program
{
    public static void Main(string[] args)
    {
        var builder = WebApplication.CreateBuilder(args);

        // Add services to the container.

        builder.Services.AddControllers();
```



```
        // Learn more about configuring Swagger/OpenAPI at
https://aka.ms/aspnetcore/swashbuckle
        builder.Services.AddEndpointsApiExplorer();
        builder.Services.AddSwaggerGen();
        var connectionString =
builder.Configuration.GetConnectionString("StudentAdmissionDb");
        builder.Services.AddDbContext<StudentAdmissionContext>(options
=>
            options.UseMySql(
                connectionString,
                new MySqlServerVersion(new Version(10, 11, 10)) //
Adjust to your MariaDB version
            )
        );



builder.Services.AddTransient<IStudentAdmissionRepository,StudentAdm
issionRepository>();
        var app = builder.Build();

        // Configure the HTTP request pipeline.
        if (app.Environment.IsDevelopment())
        {
            app.UseSwagger();
            app.UseSwaggerUI();
        }

        app.UseHttpsRedirection();

        app.UseAuthorization();


        app.MapControllers();

        app.Run();
    }
}
```

=> Output



Curl

```
curl -X 'GET' \
  'http://localhost:5093/api/StudentAdmission' \
  -H 'accept: text/plain'
```

Request URL

```
http://localhost:5093/api/StudentAdmission
```

Server response

Code	Details
200	Response body

```
[
  {
    "stuentId": 1,
    "name": "nisarg",
    "admissionDate": "2025-02-20T00:00:00",
    "admissionBranch": "Computer Engineering"
  },
  {
    "stuentId": 2,
    "name": "neha",
    "admissionDate": "2025-02-20T00:00:00",
    "admissionBranch": "Computer Engineering"
  }
]
```

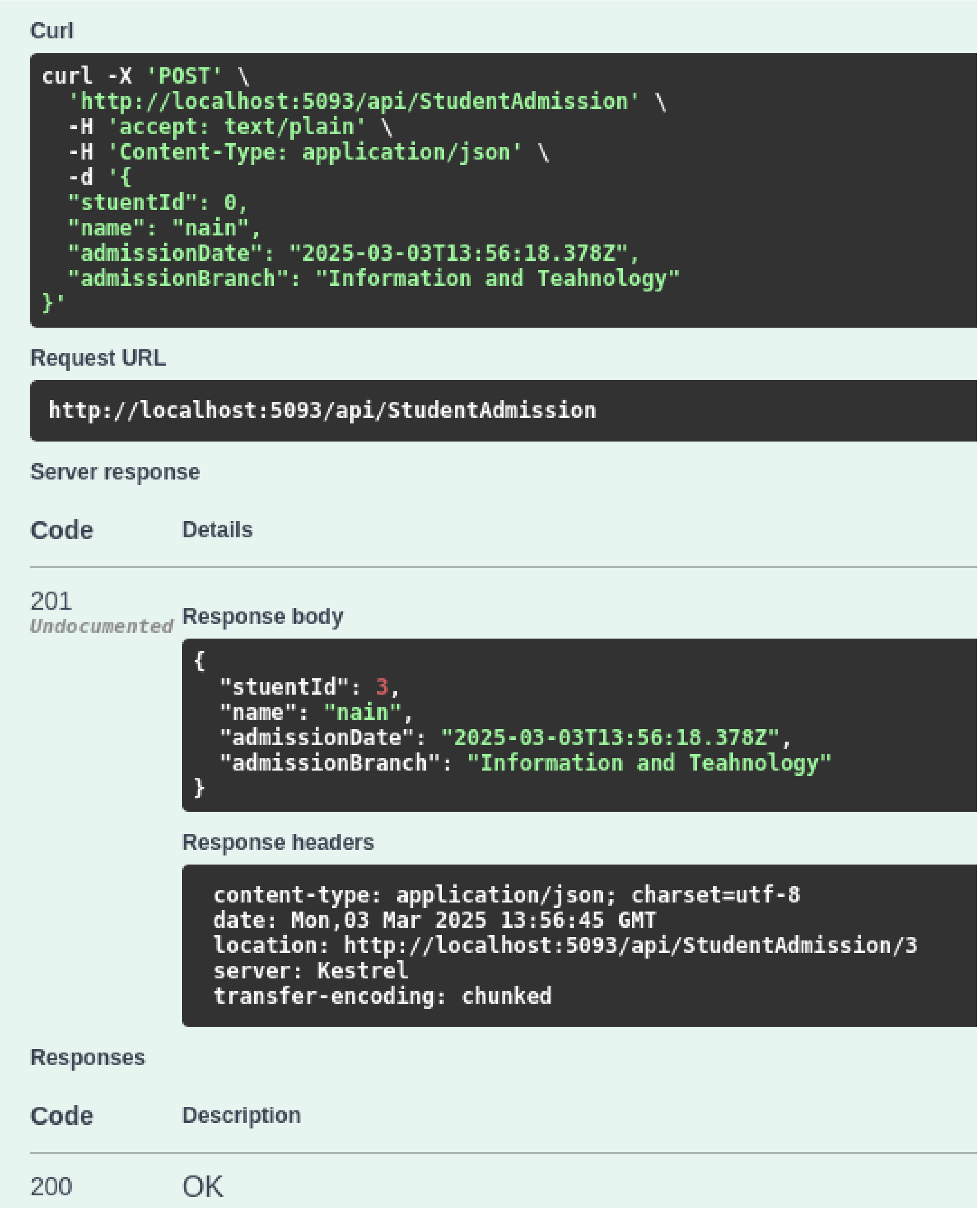
Response headers

```
 content-type: application/json; charset=utf-8
 date: Mon,03 Mar 2025 13:54:52 GMT
 server: Kestrel
 transfer-encoding: chunked
```

Responses

Code	Description
200	OK

Media type



Curl

```
curl -X 'POST' \
  'http://localhost:5093/api/StudentAdmission' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
  "stuentId": 0,
  "name": "nain",
  "admissionDate": "2025-03-03T13:56:18.378Z",
  "admissionBranch": "Information and Teahnology"
}'
```

Request URL

```
http://localhost:5093/api/StudentAdmission
```

Server response

Code	Details
201 *Undocumented*	Response body

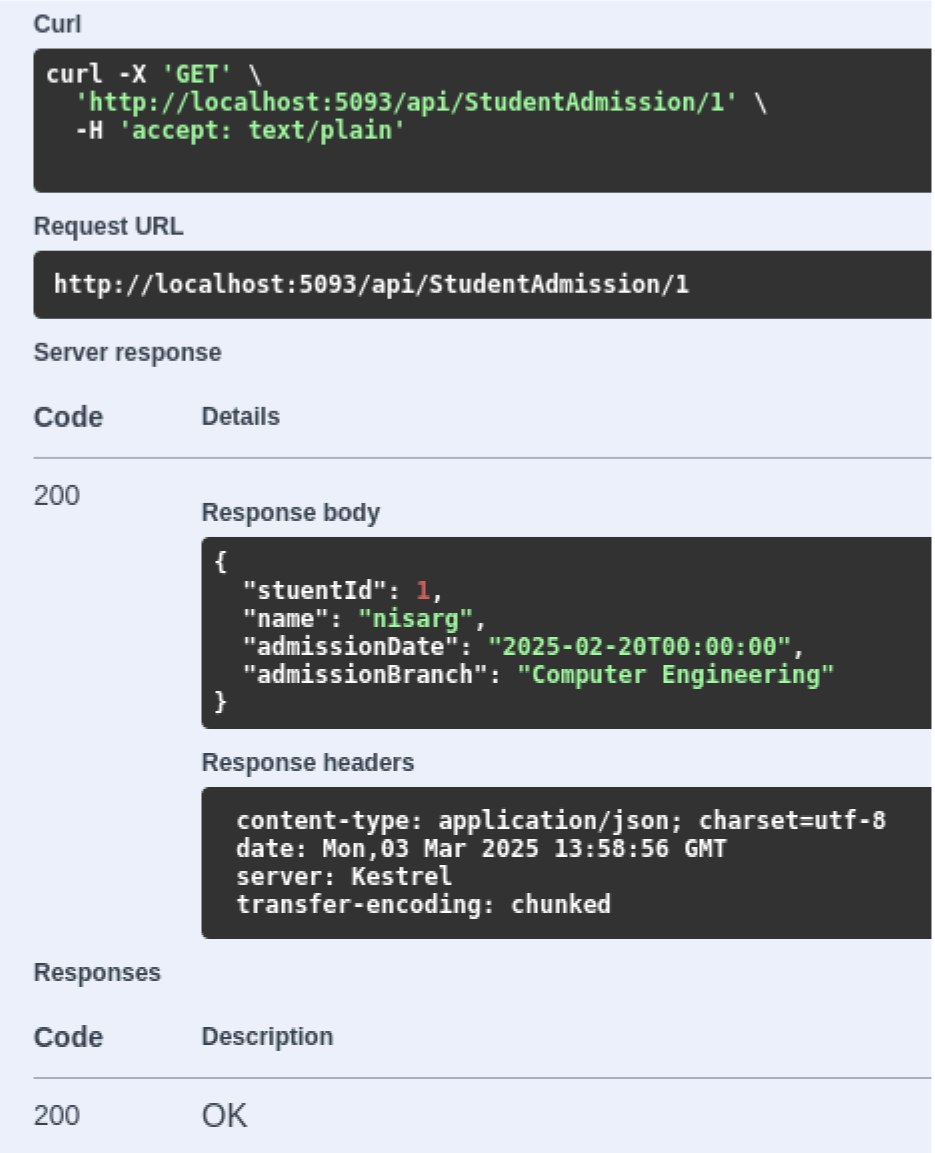
```
{
  "stuentId": 3,
  "name": "nain",
  "admissionDate": "2025-03-03T13:56:18.378Z",
  "admissionBranch": "Information and Teahnology"
}
```

Response headers

```
 content-type: application/json; charset=utf-8
 date: Mon,03 Mar 2025 13:56:45 GMT
 location: http://localhost:5093/api/StudentAdmission/3
 server: Kestrel
 transfer-encoding: chunked
```

Responses

Code	Description
200	OK



Curl

```
curl -X 'GET' \
  'http://localhost:5093/api/StudentAdmission/1' \
  -H 'accept: text/plain'
```

Request URL

```
http://localhost:5093/api/StudentAdmission/1
```

Server response

Code	Details
200	Response body

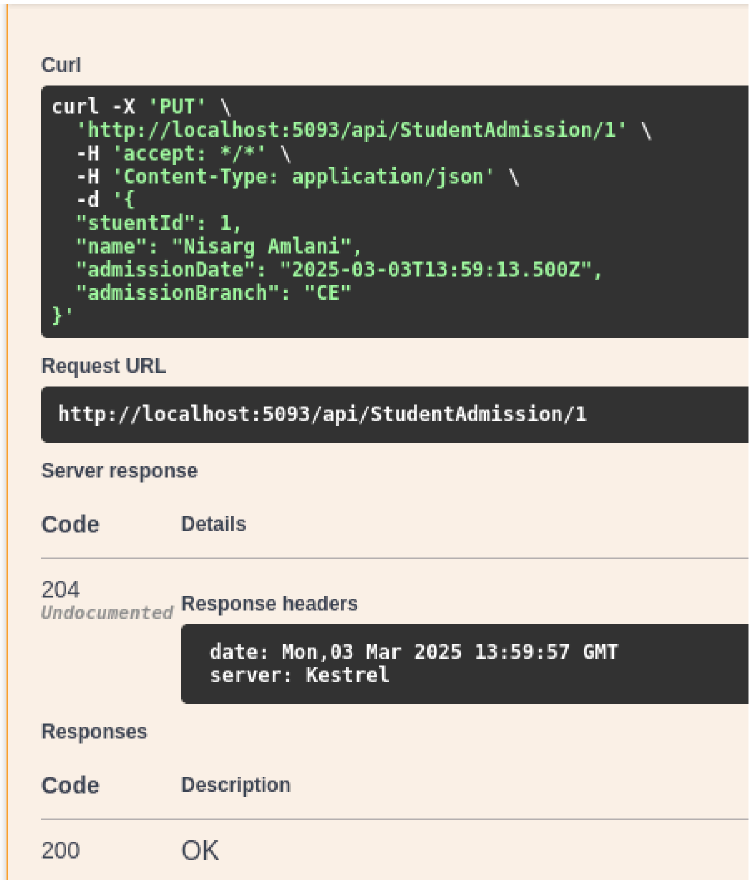
```
{
  "stuentId": 1,
  "name": "nisarg",
  "admissionDate": "2025-02-20T00:00:00",
  "admissionBranch": "Computer Engineering"
}
```

Response headers

```
 content-type: application/json; charset=utf-8
 date: Mon,03 Mar 2025 13:58:56 GMT
 server: Kestrel
 transfer-encoding: chunked
```

Responses

Code	Description
200	OK



Curl

```
curl -X 'PUT' \
  'http://localhost:5093/api/StudentAdmission/1' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
  "stuentId": 1,
  "name": "Nisarg Amlani",
  "admissionDate": "2025-03-03T13:59:13.500Z",
  "admissionBranch": "CE"
}'
```

Request URL

```
http://localhost:5093/api/StudentAdmission/1
```

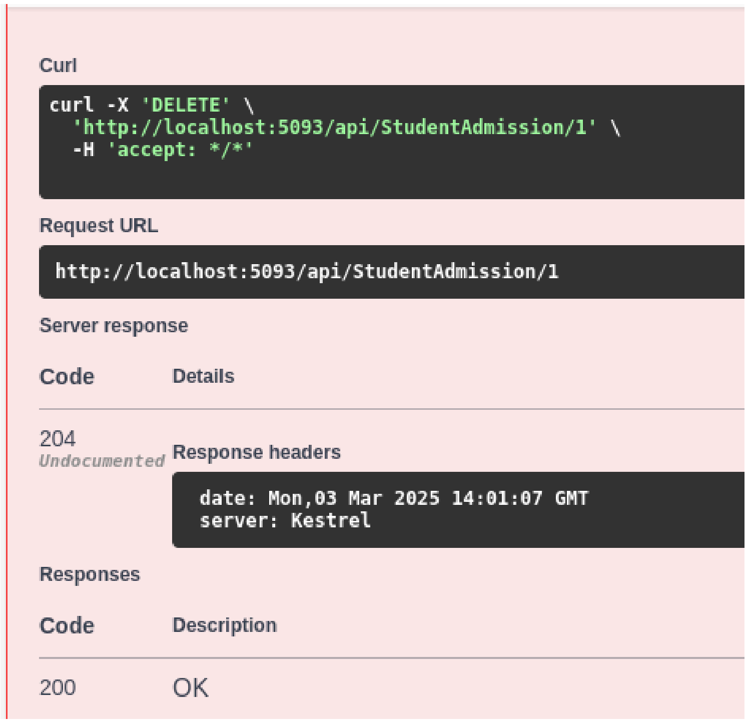
Server response

Code	Details

204
Undocumented
	Response headers
	`date: Mon,03 Mar 2025 13:59:57 GMT`
	`server: Kestrel`

Responses

Code	Description
200	OK



Curl

```
curl -X 'DELETE' \
  'http://localhost:5093/api/StudentAdmission/1' \
  -H 'accept: */*'
```

Request URL

```
http://localhost:5093/api/StudentAdmission/1
```

Server response

Code	Details

204
Undocumented
	Response headers
	`date: Mon,03 Mar 2025 14:01:07 GMT`
	`server: Kestrel`

Responses

Code	Description
200	OK

=> Code of Student Model (student.cs)

```
namespace Lab10.Models;
using System.ComponentModel.DataAnnotations;


public class Student
{
   [Key]
   public int SteuntId { get; set; }
   public int Semester { get; set; }
   public double Attendance_Percentage { get; set; }
}
```

=> Code Student Attandence Context (StudentAdttendanceContext.cs)

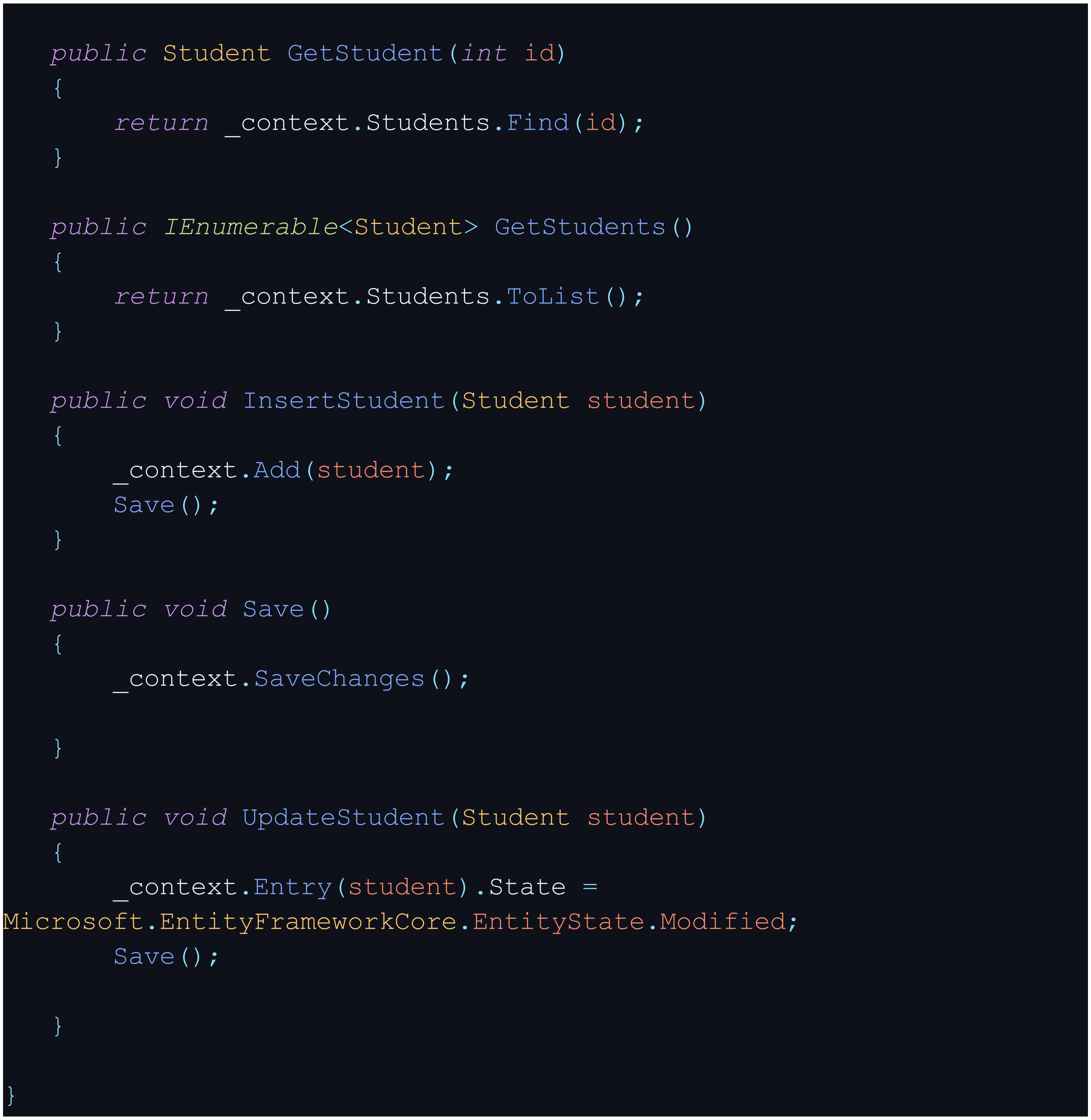
```
using Microsoft.EntityFrameworkCore;

namespace Lab10.Models;

public class StudentAdttendanceContext : DbContext
{
   public StudentAdttendanceContext(DbContextOptions options) :
base(options)
   {
   }

   public DbSet<Student> Students { get; set; }

   protected override void OnModelCreating(ModelBuilder
modelBuilder)
   {
       base.OnModelCreating(modelBuilder);

       modelBuilder.Entity<Student>().HasData(
           new Student
           {
               StuentId = 1,
               Semester = 6,
               Attendance_Percentage = 86.5
           }
           ,
           new Student
           {
               StuentId = 2,
               Semester = 6,
               Attendance_Percentage = 88.5
           }
       );
   }
}
```

=> Code of Interface Student Attandence (IStudentAttandenceRepository.cs)

```
using Microsoft.EntityFrameworkCore;

namespace Lab10.Models;

public class StudentAdttendanceContext : DbContext
{
   public StudentAdttendanceContext(DbContextOptions options) :
base(options)
   {
   }

   public DbSet<Student> Students { get; set; }

   protected override void OnModelCreating(ModelBuilder
modelBuilder)
   {
       base.OnModelCreating(modelBuilder);

       modelBuilder.Entity<Student>().HasData(
           new Student
           {
               StuentId = 1,
               Semester = 6,
               Attendance_Percentage = 86.5
           }
           ,
           new Student
           {
               StuentId = 2,
               Semester = 6,
               Attendance_Percentage = 88.5
           }
       );
   }
}
```

=> Code of Implementation Student Attendance (StudentAttendanceRepository.cs)

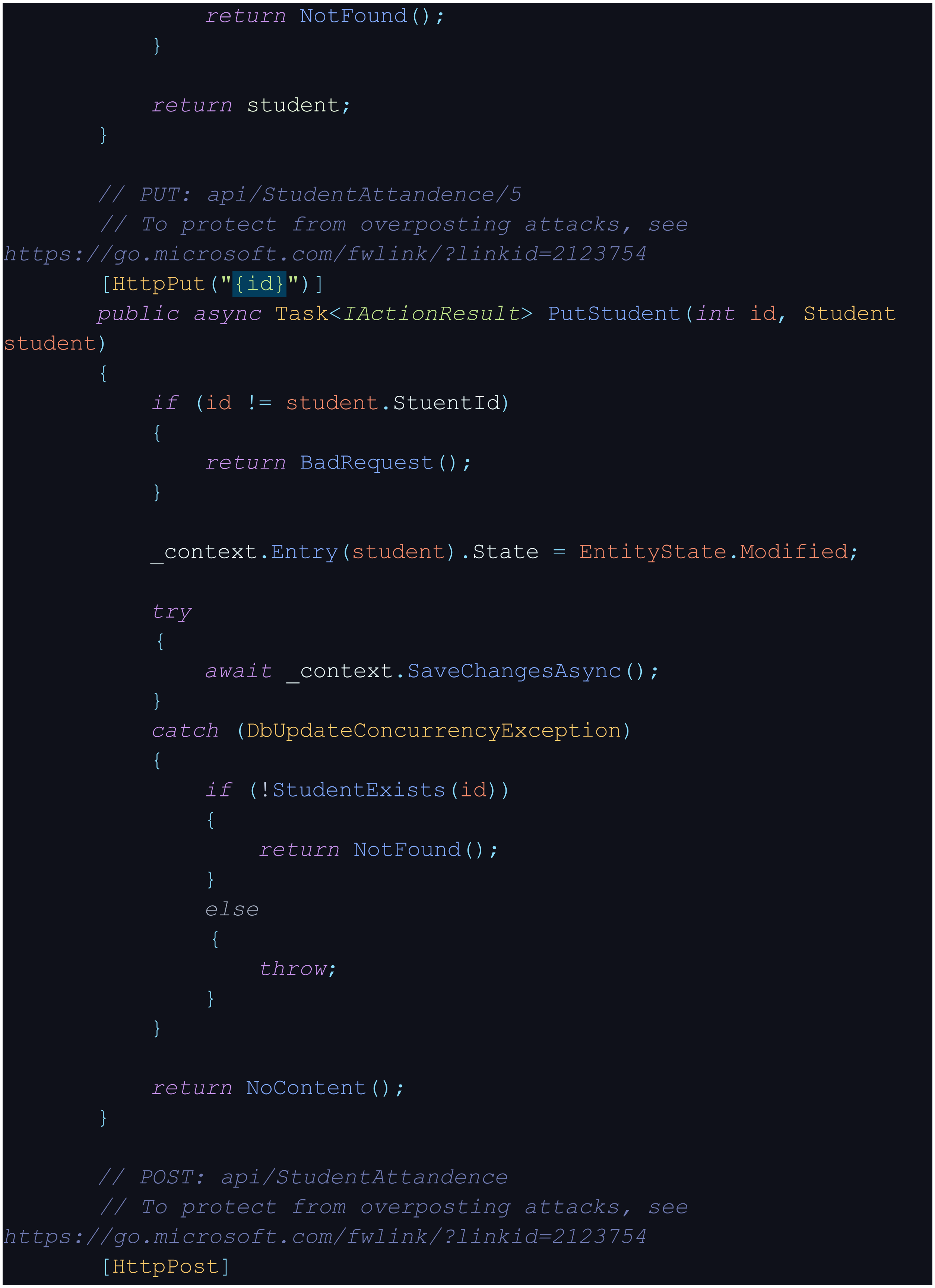
```
namespace Lab10.Repository;
using Lab10.Models;

public class StudentAttandenceRepository :
IStudentAttandenceRepository
{
   private readonly StudentAdttendanceContext _context;
   public StudentAttandenceRepository(StudentAdttendanceContext
context) {
       _context = context;
   }

   public void DeleteStudent(int id)
   {
       var student = _context.Students.Find(id);
       _context.Students.Remove(student);
       Save();
   }
```



```
    public Student GetStudent(int id)
    {
        return _context.Students.Find(id);
    }

    public IEnumerable<Student> GetStudents()
    {
        return _context.Students.ToList();
    }

    public void InsertStudent(Student student)
    {
        _context.Add(student);
        Save();
    }

    public void Save()
    {
        _context.SaveChanges();

    }

    public void UpdateStudent(Student student)
    {
        _context.Entry(student).State =
Microsoft.EntityFrameworkCore.EntityState.Modified;
        Save();

    }

}
```

=> Code of Student Attendance Controller (StudentAttandenceController.cs)

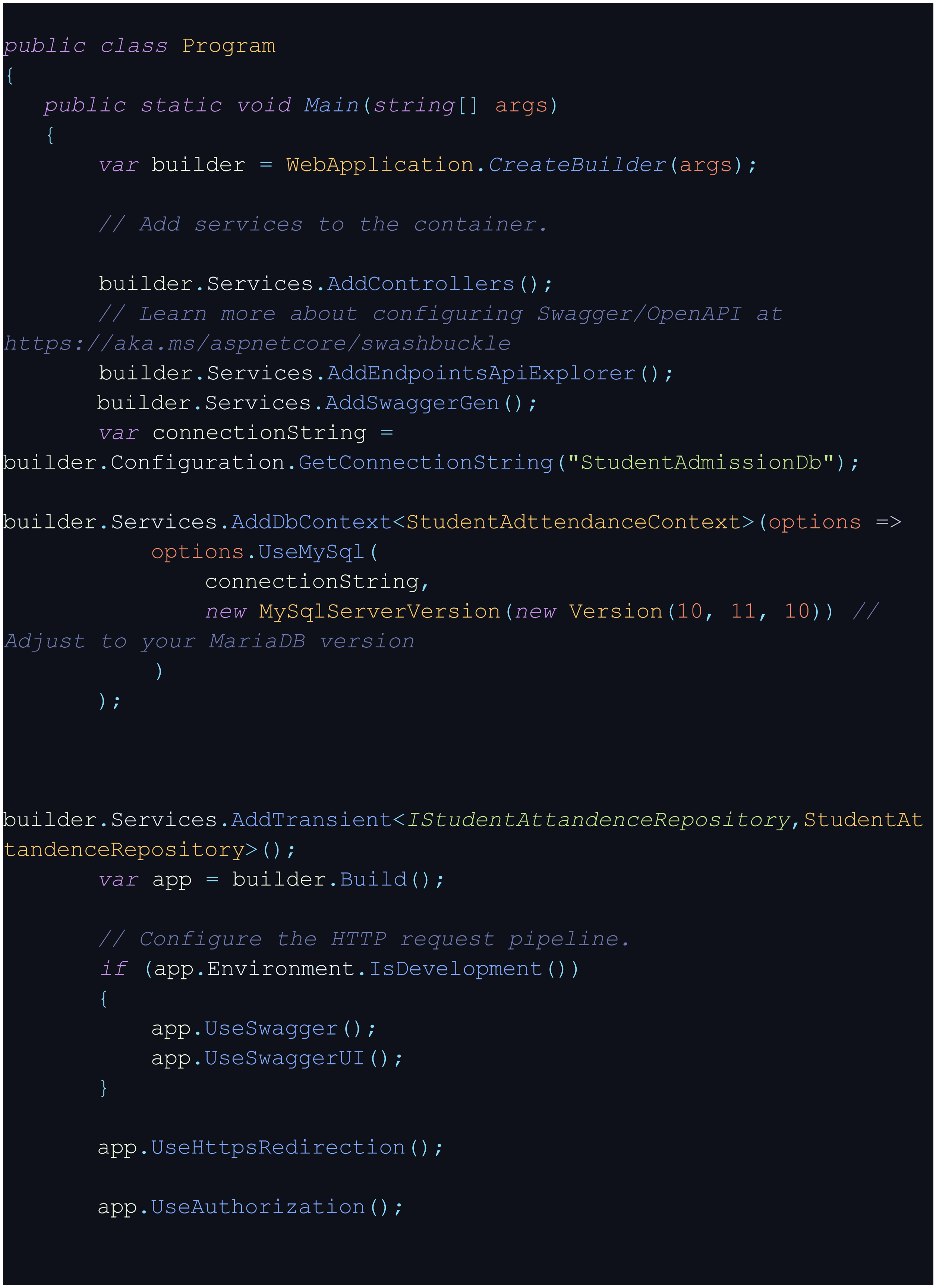
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using Lab10.Models;

namespace Lab10.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class StudentAttandenceController : ControllerBase
    {
        private readonly StudentAdttendanceContext _context;

        public StudentAttandenceController(StudentAdttendanceContext
context)
        {
            _context = context;
        }

        // GET: api/StudentAttandence
        [HttpGet]
        public async Task<ActionResult<IEnumerable<Student>>>
GetStudents()
        {
            return await _context.Students.ToListAsync();
        }

        // GET: api/StudentAttandence/5
        [HttpGet("{id}")]
        public async Task<ActionResult<Student>> GetStudent(int id)
        {
            var student = await _context.Students.FindAsync(id);

            if (student == null)
            {
```



```
                return NotFound();
            }

            return student;
        }

        // PUT: api/StudentAttandence/5
        // To protect from overposting attacks, see
https://go.microsoft.com/fwlink/?linkid=2123754
        [HttpPut("{id}")]
        public async Task<IActionResult> PutStudent(int id, Student
student)
        {
            if (id != student.StuentId)
            {
                return BadRequest();
            }

            _context.Entry(student).State = EntityState.Modified;

            try
            {
                await _context.SaveChangesAsync();
            }
            catch (DbUpdateConcurrencyException)
            {
                if (!StudentExists(id))
                {
                    return NotFound();
                }
                else
                {
                    throw;
                }
            }

            return NoContent();
        }

        // POST: api/StudentAttandence
        // To protect from overposting attacks, see
https://go.microsoft.com/fwlink/?linkid=2123754
        [HttpPost]
        public async Task<ActionResult<Student>> PostStudent(Student student)
        {
            _context.Students.Add(student);
            await _context.SaveChangesAsync();

            return CreatedAtAction("GetStudent", new { id = student.StuentId }, student);
        }

        // DELETE: api/StudentAttandence/5
        [HttpDelete("{id}")]
        public async Task<IActionResult> DeleteStudent(int id)
        {
            var student = await _context.Students.FindAsync(id);
            if (student == null)
            {
                return NotFound();
            }

            _context.Students.Remove(student);
            await _context.SaveChangesAsync();

            return NoContent();
        }

        private bool StudentExists(int id)
        {
            return _context.Students.Any(e => e.StuentId == id);
        }
    }
}
```

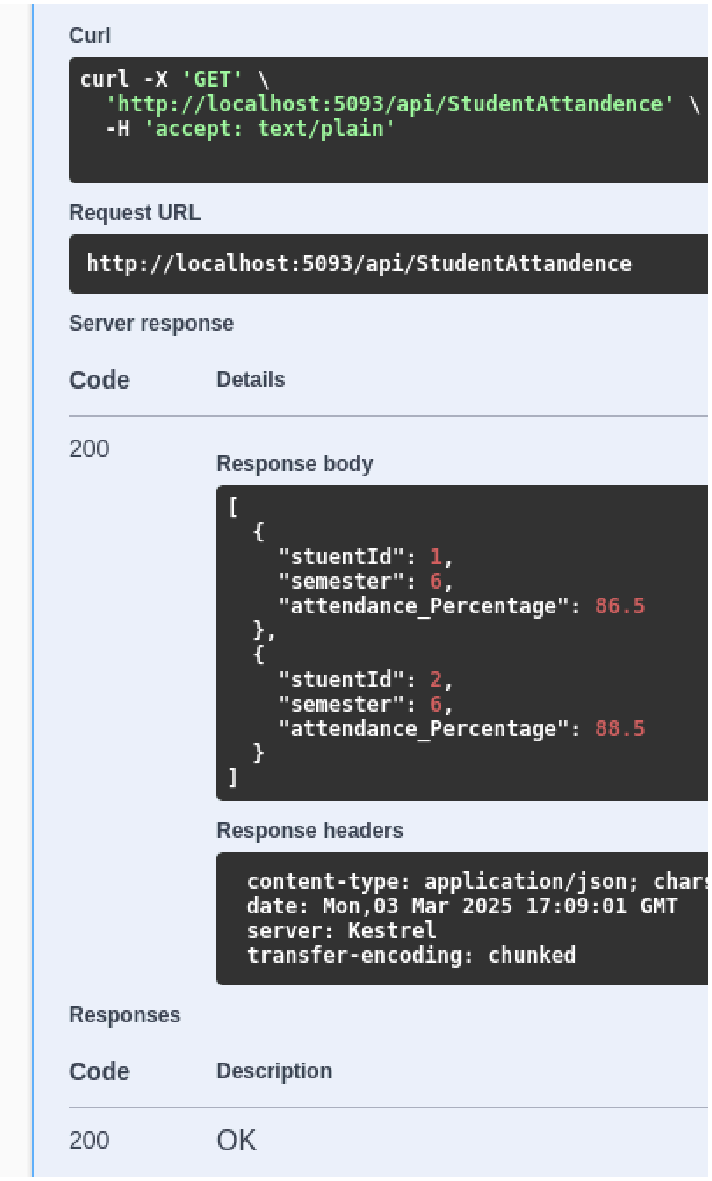
=> Code for Program.cs

```
using Microsoft.EntityFrameworkCore;

namespace Lab10;
using Lab10.Models;
using Lab10.Repository;
```



```
public class Program
{
    public static void Main(string[] args)
    {
        var builder = WebApplication.CreateBuilder(args);

        // Add services to the container.

        builder.Services.AddControllers();
        // Learn more about configuring Swagger/OpenAPI at
https://aka.ms/aspnetcore/swashbuckle
        builder.Services.AddEndpointsApiExplorer();
        builder.Services.AddSwaggerGen();
        var connectionString =
builder.Configuration.GetConnectionString("StudentAdmissionDb");

builder.Services.AddDbContext<StudentAdttendanceContext>(options =>
            options.UseMySql(
                connectionString,
                new MySqlServerVersion(new Version(10, 11, 10)) //
Adjust to your MariaDB version
            )
        );



builder.Services.AddTransient<IStudentAttandenceRepository,StudentAt
tandenceRepository>();
        var app = builder.Build();

        // Configure the HTTP request pipeline.
        if (app.Environment.IsDevelopment())
        {
            app.UseSwagger();
            app.UseSwaggerUI();
        }

        app.UseHttpsRedirection();

        app.UseAuthorization();
        app.MapControllers();

        app.Run();
    }
}
```

=> Output



Curl

```
curl -X 'GET' \
  'http://localhost:5093/api/StudentAttandence' \
  -H 'accept: text/plain'
```

Request URL

```
http://localhost:5093/api/StudentAttandence
```

Server response

Code Details

200

Response body

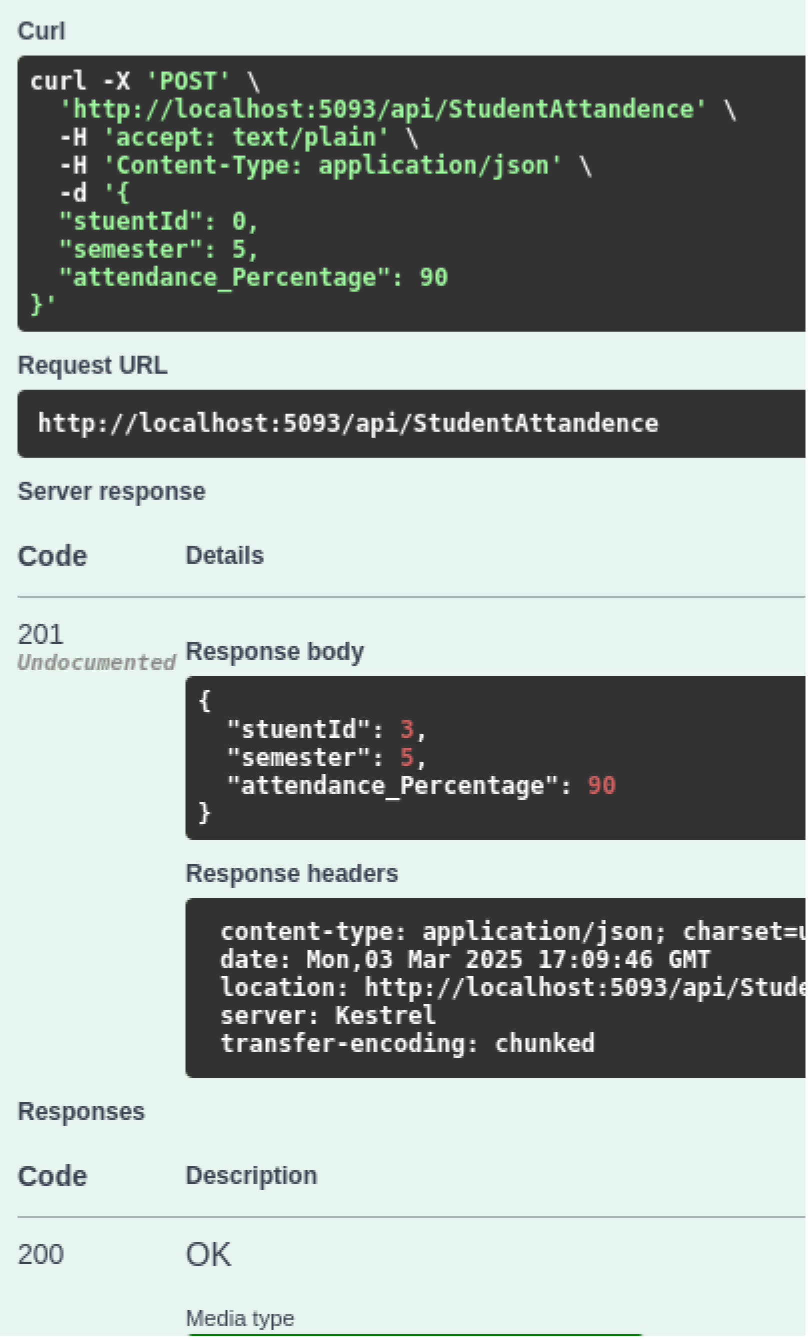
```
[
  {
    "stuentId": 1,
    "semester": 6,
    "attendance_Percentage": 86.5
  },
  {
    "stuentId": 2,
    "semester": 6,
    "attendance_Percentage": 88.5
  }
]
```

Response headers

```
 content-type: application/json; chars
 date: Mon,03 Mar 2025 17:09:01 GMT
 server: Kestrel
 transfer-encoding: chunked
```

Responses

Code	Description
200	OK



Curl

```
curl -X 'POST' \
  'http://localhost:5093/api/StudentAttandence' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
  "stuentId": 0,
  "semester": 5,
  "attendance_Percentage": 90
}'
```

Request URL

```
http://localhost:5093/api/StudentAttandence
```

Server response

Code	Details
201 *Undocumented*	Response body

```
{
  "stuentId": 3,
  "semester": 5,
  "attendance_Percentage": 90
}
```

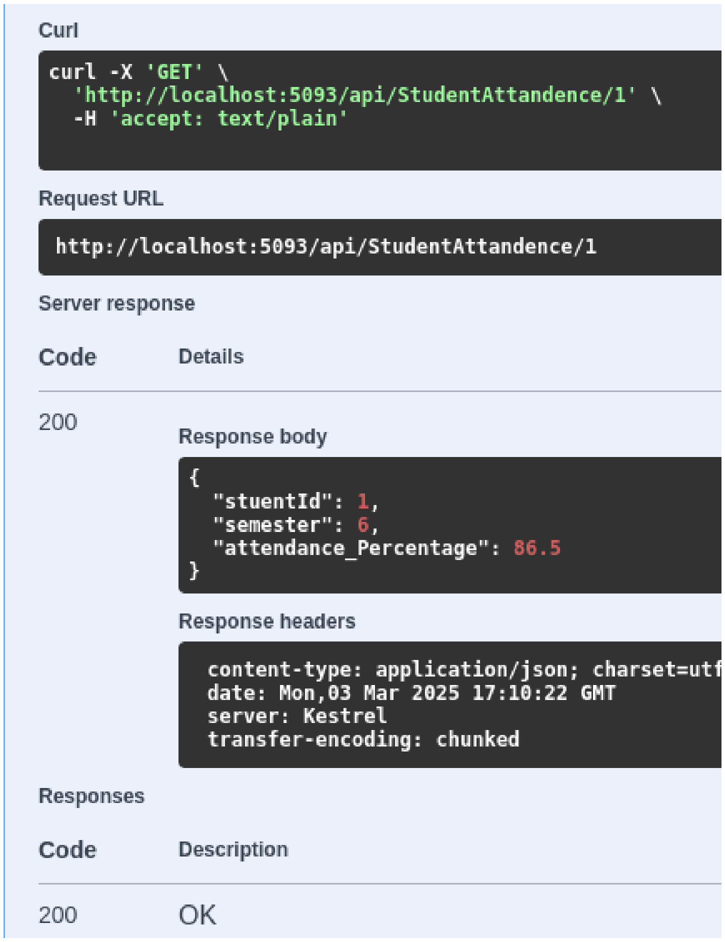
Response headers

```
 content-type: application/json; charset=u
 date: Mon,03 Mar 2025 17:09:46 GMT
 location: http://localhost:5093/api/Stude
 server: Kestrel
 transfer-encoding: chunked
```

Responses

Code	Description
200	OK

Media type



Curl

```
curl -X 'GET' \
  'http://localhost:5093/api/StudentAttandence/1' \
  -H 'accept: text/plain'
```

Request URL

```
http://localhost:5093/api/StudentAttandence/1
```

Server response

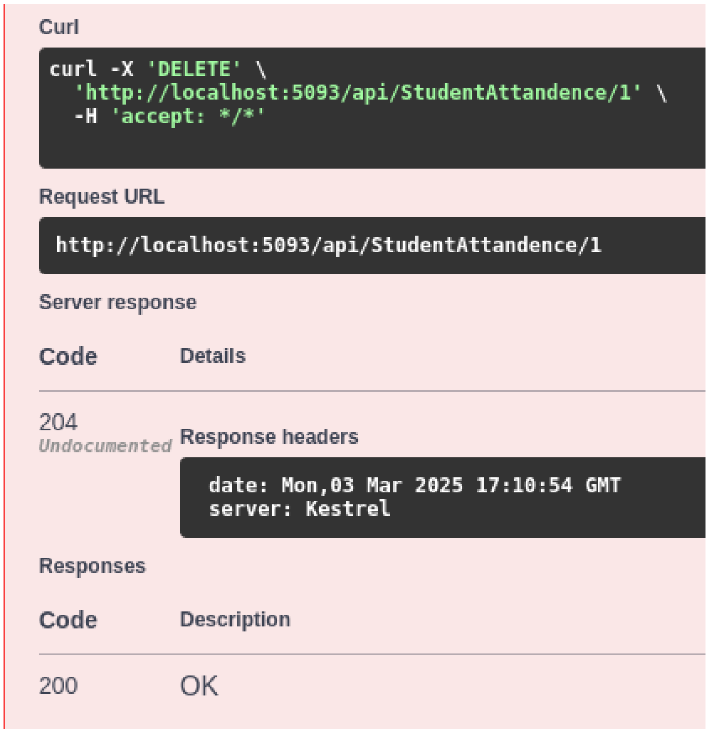
Code	Details
200	Response body `{ "stuentId": 1, "semester": 6, "attendance_Percentage": 86.5 }` Response headers

```
{
  "stuentId": 1,
  "semester": 6,
  "attendance_Percentage": 86.5
}
```

```
 content-type: application/json; charset=utf
 date: Mon,03 Mar 2025 17:10:22 GMT
 server: Kestrel
 transfer-encoding: chunked
```

Responses

Code	Description
200	OK



Curl

```
curl -X 'DELETE' \
  'http://localhost:5093/api/StudentAttandence/1' \
  -H 'accept: */*'
```

Request URL

```
http://localhost:5093/api/StudentAttandence/1
```

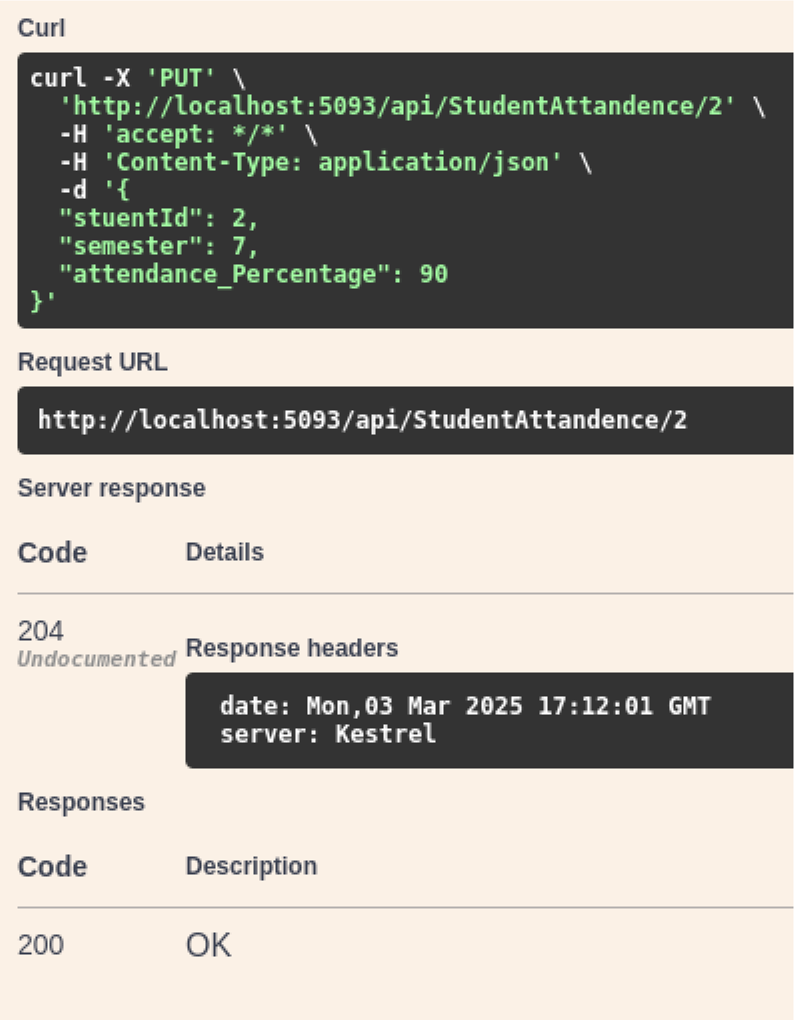
Server response

Code	Details
204 *Undocumented*	Response headers

```
 date: Mon,03 Mar 2025 17:10:54 GMT
 server: Kestrel
```

Responses

Code	Description
200	OK



Curl

```
curl -X 'PUT' \
  'http://localhost:5093/api/StudentAttandence/2' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
  "stuentId": 2,
  "semester": 7,
  "attendance_Percentage": 90
}'
```

Request URL

```
http://localhost:5093/api/StudentAttandence/2
```

Server response

Code	Details
204 *Undocumented*	Response headers

```
date: Mon,03 Mar 2025 17:12:01 GMT
server: Kestrel
```

Responses

Code	Description
200	OK