

Student.cs

```
using System.ComponentModel.DataAnnotations;

namespace RainbowSchool.Models
{
    public class Student
    {
        [Key]
        public int StudentId { get; set; }
        [Required]
        public string StudentName { get; set; }
        public string StudentEmail { get; set; }
    }
}
```

Subject.cs

```
using System.ComponentModel.DataAnnotations;

namespace RainbowSchool.Models
{
    public class Subject
    {
        [Key]
        public int SubjectId { get; set; }
        [Required]
        public string SubjectName { get; set; }
    }
}
```

Marks.cs

```
using System.ComponentModel.DataAnnotations;

namespace RainbowSchool.Models
{
    public class Marks
    {
        [Key]
        public int SubjectId { get; set; }
        [Required]
        public int StudentId { get; set; }
        [Required]
        public double SubjectMarks { get; set; }
    }
}
```

MarksController.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 RainbowSchool.Data;
using RainbowSchool.Models;

namespace RainbowSchool.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class MarksController : ControllerBase
    {
        private readonly RainbowSchoolContext _context;

        public MarksController(RainbowSchoolContext context)
        {
            _context = context;
        }

        // GET: api/Marks
        [HttpGet]
        public async Task<ActionResult<IEnumerable<Marks>>> GetMarks()
        {
            if (_context.Marks == null)
            {
                return NotFound();
            }
            return await _context.Marks.ToListAsync();
        }

        // GET: api/Marks/5
        [HttpGet("{id}")]
        public async Task<ActionResult<Marks>> GetMarks(int id)
        {
            if (_context.Marks == null)
            {
                return NotFound();
            }
            var marks = await _context.Marks.FindAsync(id);

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

            return marks;
        }

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

```

```

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

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

        return NoContent();
    }

    // POST: api/Marks
    // To protect from overposting attacks, see
https://go.microsoft.com/fwlink/?linkid=2123754
    [HttpPost]
    public async Task<ActionResult<Marks>> PostMarks(Marks marks)
    {
        if (_context.Marks == null)
        {
            return Problem("Entity set 'RainbowSchoolContext.Marks' is null.");
        }
        _context.Marks.Add(marks);
        await _context.SaveChangesAsync();

        return CreatedAtAction("GetMarks", new { id = marks.SubjectId }, marks);
    }

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

        _context.Marks.Remove(marks);
        await _context.SaveChangesAsync();

        return NoContent();
    }

    private bool MarksExists(int id)

```

```

        {
            return (_context.Marks?.Any(e => e.SubjectId ==
id)).GetValueOrDefault();
        }
    }
}

```

appSettings.json

```

{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "RainbowSchoolContext": "Server=DESKTOP-
CHNJ5UD;Database=RainbowSchoolAPI;TrustServerCertificate=true;Trusted_Connection=Tru
e;MultipleActiveResultSets=true"
  }
}

```