Practice Project-4

Section 9 Source Code

## Marks.cs

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace WebAPI.Models
{
    [Table("Marks")]
    public class Marks
    {
        [Key]
        public int StudentId { get; set; }
        [Required]
        [StringLength(50)]
        public string StudentName { get; set;}
        [Required]
        [StringLength(50)]
        public string ClassName { get; set; }
        [Required]
        [StringLength(50)]
        public string SubjectName { get; set;}
        [Required]
        public int Mark { 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 WebAPI.Data;
using WebAPI.Models;

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

        public MarksController(WebAPIDbContext 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.StudentId)
            {
                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 'WebAPIDbContext.Marks'  is null.");
  }
    _context.Marks.Add(marks);
    await _context.SaveChangesAsync();

    return CreatedAtAction("GetMarks", new { id = marks.StudentId }, 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.StudentId == id)).GetValueOrDefault();
        }
    }
}
```

## WebAPIDbContext

```
using Microsoft.EntityFrameworkCore;

namespace WebAPI.Data
{
    public class WebAPIDbContext : DbContext
    {
        public WebAPIDbContext (DbContextOptions<WebAPIDbContext> options)
            : base(options)
        {
        }

        public DbSet<WebAPI.Models.Marks> Marks { get; set; } = default!;
    }
}
```