```
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 PhaseEnd1.Data;
using PhaseEnd1.Models;

namespace PhaseEnd1.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class DeptMastersController : ControllerBase
    {
        private readonly PhaseEnd1DBContext _context;

        public DeptMastersController(PhaseEnd1DBContext context)
        {
            _context = context;
        }

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

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

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

            return deptMaster;
        }

        // PUT: api/DeptMasters/5
        // To protect from overposting attacks, see
https://go.microsoft.com/fwlink/?linkid=2123754
        [HttpPut("{id}")]
        public async Task<IActionResult> PutDeptMaster(int id, DeptMaster
deptMaster)
```

```
        {
            if (id != deptMaster.DeptCode)
            {
                return BadRequest();
            }

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

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

            return NoContent();
        }

        // POST: api/DeptMasters
        // To protect from overposting attacks, see
https://go.microsoft.com/fwlink/?linkid=2123754
        [HttpPost]
        public async Task<ActionResult<DeptMaster>> PostDeptMaster(DeptMaster
deptMaster)
        {
          if (_context.DeptMaster == null)
          {
              return Problem("Entity set 'PhaseEnd1DBContext.DeptMaster'  is
null.");
          }
            _context.DeptMaster.Add(deptMaster);
            await _context.SaveChangesAsync();

            return CreatedAtAction("GetDeptMaster", new { id = deptMaster.DeptCode
}, deptMaster);
        }

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

```
            }

            _context.DeptMaster.Remove(deptMaster);
            await _context.SaveChangesAsync();

            return NoContent();
        }

        private bool DeptMasterExists(int id)
        {
            return (_context.DeptMaster?.Any(e => e.DeptCode ==
id)).GetValueOrDefault();
        }
    }
}


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 PhaseEnd1.Data;
using PhaseEnd1.Models;

namespace PhaseEnd1.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class EmpProfilesController : ControllerBase
    {
        private readonly PhaseEnd1DBContext _context;

        public EmpProfilesController(PhaseEnd1DBContext context)
        {
            _context = context;
        }

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

        // GET: api/EmpProfiles/5
        [HttpGet("{id}")]
        public async Task<ActionResult<EmpProfile>> GetEmpProfile(int id)
        {
          if (_context.EmpProfile == null)
          {
              return NotFound();
```

```
          }
            var empProfile = await _context.EmpProfile.FindAsync(id);

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

            return empProfile;
        }

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

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

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

            return NoContent();
        }

        // POST: api/EmpProfiles
        // To protect from overposting attacks, see
https://go.microsoft.com/fwlink/?linkid=2123754
        [HttpPost]
        public async Task<ActionResult<EmpProfile>> PostEmpProfile(EmpProfile
empProfile)
        {
          if (_context.EmpProfile == null)
          {
              return Problem("Entity set 'PhaseEnd1DBContext.EmpProfile'  is
null.");
          }
            _context.EmpProfile.Add(empProfile);
            await _context.SaveChangesAsync();
```

```
            return CreatedAtAction("GetEmpProfile", new { id = empProfile.EmpCode },
empProfile);
        }

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

            _context.EmpProfile.Remove(empProfile);
            await _context.SaveChangesAsync();

            return NoContent();
        }

        private bool EmpProfileExists(int id)
        {
            return (_context.EmpProfile?.Any(e => e.EmpCode ==
id)).GetValueOrDefault();
        }
    }
}
```

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

namespace PhaseEnd1.Models
{
    [Table("DeptMaster")]
    public class DeptMaster
    {
        [Key]
        public int DeptCode { get; set; }

        public string DeptName { get; set; }
        public virtual ICollection<EmpProfile> EmpProfile { get; set; }
    }
}
```

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

namespace PhaseEnd1.Models
{
    [Table("EmpProfile")]
    public class EmpProfile
```

```
    {
        [Key]
        public int EmpCode { get; set; }
        public DateTime DateOfBirth { get; set; }
        public string EmpName { get; set; }
        public int DeptCode { get; set; }
        public virtual ICollection<DeptMaster> DeptMaster { get; set; }
    }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.EntityFrameworkCore;
using PhaseEnd1.Models;

namespace PhaseEnd1.Data
{
    public class PhaseEnd1DBContext : DbContext
    {
        public PhaseEnd1DBContext (DbContextOptions<PhaseEnd1DBContext> options)
            : base(options)
        {
        }

        public DbSet<PhaseEnd1.Models.DeptMaster> DeptMaster { get; set; } =
default!;

        public DbSet<PhaseEnd1.Models.EmpProfile>? EmpProfile { get; set; }
    }
}
// <auto-generated />
using System;
using Microsoft.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore.Infrastructure;
using Microsoft.EntityFrameworkCore.Metadata;
using Microsoft.EntityFrameworkCore.Storage.ValueConversion;
using PhaseEnd1.Data;

#nullable disable

namespace PhaseEnd1.Migrations
{
    [DbContext(typeof(PhaseEnd1DBContext))]
    partial class PhaseEnd1DBContextModelSnapshot : ModelSnapshot
    {
        protected override void BuildModel(ModelBuilder modelBuilder)
        {
#pragma warning disable 612, 618
            modelBuilder
                .HasAnnotation("ProductVersion", "7.0.10")
                .HasAnnotation("Relational:MaxIdentifierLength", 128);

            SqlServerModelBuilderExtensions.UseIdentityColumns(modelBuilder);

            modelBuilder.Entity("DeptMasterEmpProfile", b =>
                {
                    b.Property<int>("DeptMasterDeptCode")
```

```
                        .HasColumnType("int");

                    b.Property<int>("EmpProfileEmpCode")
                        .HasColumnType("int");

                    b.HasKey("DeptMasterDeptCode", "EmpProfileEmpCode");

                    b.HasIndex("EmpProfileEmpCode");

                    b.ToTable("DeptMasterEmpProfile");
                });

            modelBuilder.Entity("PhaseEnd1.Models.DeptMaster", b =>
                {
                    b.Property<int>("DeptCode")
                        .ValueGeneratedOnAdd()
                        .HasColumnType("int");


SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("DeptCode"));

                    b.Property<string>("DeptName")
                        .IsRequired()
                        .HasColumnType("nvarchar(max)");

                    b.HasKey("DeptCode");

                    b.ToTable("DeptMaster");
                });

            modelBuilder.Entity("PhaseEnd1.Models.EmpProfile", b =>
                {
                    b.Property<int>("EmpCode")
                        .ValueGeneratedOnAdd()
                        .HasColumnType("int");


SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("EmpCode"));

                    b.Property<DateTime>("DateOfBirth")
                        .HasColumnType("datetime2");

                    b.Property<int>("DeptCode")
                        .HasColumnType("int");

                    b.Property<string>("EmpName")
                        .IsRequired()
                        .HasColumnType("nvarchar(max)");

                    b.HasKey("EmpCode");

                    b.ToTable("EmpProfile");
                });

            modelBuilder.Entity("DeptMasterEmpProfile", b =>
                {
                    b.HasOne("PhaseEnd1.Models.DeptMaster", null)
                        .WithMany()
```

```
                        .HasForeignKey("DeptMasterDeptCode")
                        .OnDelete(DeleteBehavior.Cascade)
                        .IsRequired();

                    b.HasOne("PhaseEnd1.Models.EmpProfile", null)
                        .WithMany()
                        .HasForeignKey("EmpProfileEmpCode")
                        .OnDelete(DeleteBehavior.Cascade)
                        .IsRequired();
                });
#pragma warning restore 612, 618
        }
    }
}
```