Step 1: Config

builder.Services.AddMediatR(Assembly.GetExecutingAssembly());

Step 2:

public interface IStudentRepository

{

public Task<List<StudentDetails>> GetStudentListAsync();

public Task<StudentDetails> GetStudentByIdAsync(int Id);

public Task<StudentDetails> AddStudentAsync(StudentDetails studentDetails);

public Task<int> UpdateStudentAsync(StudentDetails studentDetails);

public Task<int> DeleteStudentAsync(int Id);

}

Step 3:

public class StudentRepository : IStudentRepository

{

private readonly DbContextClass \_dbContext;

public StudentRepository(DbContextClass dbContext)

{

\_dbContext = dbContext;

}

public async Task<StudentDetails> AddStudentAsync(StudentDetails studentDetails)

{

var result = \_dbContext.Students.Add(studentDetails);

await \_dbContext.SaveChangesAsync();

return result.Entity;

}

public async Task<int> DeleteStudentAsync(int Id)

{

var filteredData = \_dbContext.Students.Where(x => x.Id == Id).FirstOrDefault();

\_dbContext.Students.Remove(filteredData);

return await \_dbContext.SaveChangesAsync();

}

public async Task<StudentDetails> GetStudentByIdAsync(int Id)

{

return await \_dbContext.Students.Where(x => x.Id == Id).FirstOrDefaultAsync();

}

public async Task<List<StudentDetails>> GetStudentListAsync()

{

return await \_dbContext.Students.ToListAsync();

}

public async Task<int> UpdateStudentAsync(StudentDetails studentDetails)

{

\_dbContext.Students.Update(studentDetails);

return await \_dbContext.SaveChangesAsync();

}

// thực hiện các chức năng

}