/\* Question1 \*/

/\* Task1 \*/

app.Map(("/Currenttime"),Index);

static void Index(IApplicationBuilder application)

{

application.Run(async (context) =>

{

await context.Response.WriteAsync(DateTime.Now.ToString("dddd, dd MMMM yyyy HH:mm:ss"));

});

}

app.Run(async (context) =>

{

await context.Response.WriteAsync("Hello ASP.NET Core MVC Class!!!.");

});

/\* Task2 \*/

app.Map("/Currenttime", (IApplicationBuilder application) =>

{

application.Use(async (context, next) =>

{

await next();

await context.Response.WriteAsync("\n" + DateTime.Now.ToString("dddd, dd MMMM yyyy HH:mm:ss"));

});

application.Use(async (context, next) =>

{

await next();

});

application.Run((context) =>

{

return context.Response.WriteAsync(DateTime.Now.ToString("dddd, dd MMMM yyyy HH:mm:ss"));

});

});

/\* Question2 \*/

var x = 2;

app.Map("/varX", (IApplicationBuilder application) =>

{

application.Use(async (context, next) =>

{

x += 2;

await next();

await context.Response.WriteAsync($"{x}");

});

application.Use(async (context, next) =>

{

x += 2;

await next();

});

application.Use(async (context, next) =>

{

x += 2;

await next();

});

});

/\* Question3 \*/

DatabaseContext.cs

public DbSet<Department> Departments { get; set; }

public DatabaseContext(DbContextOptions<DatabaseContext> options): base(options)

{

Database.EnsureCreated();

}

Department.cs

public class Department

{

public int Id { get; set; }

public string Name { get; set; }

}

Program.cs

string connection =

builder.Configuration.GetConnectionString("DefaultConnection");

builder.Services.AddDbContext<DatabaseContext>(options => options.UseSqlServer(connection));

app.MapGet("/get", (DatabaseContext context) => context.Departments);

app.MapPost("/post", (DatabaseContext context, Department department) =>

{

context.Departments.Add(department);

context.SaveChanges();

});

app.MapPut("/put", (DatabaseContext context, Department department) =>

{

var dept = context.Departments.FirstOrDefault(x => x.Id == department.Id);

if (dept != null)

{

dept.Name = department.Name;

context.Departments.Update(dept);

context.SaveChanges();

}

});

app.MapDelete("/{id}", (DatabaseContext context, int id) =>

{

var dept = context.Departments.FirstOrDefault(x => x.Id == id);

if (dept != null)

{

context.Departments.Remove(dept);

context.SaveChanges();

}

});

OR

DatabaseContext.cs

public DbSet<Employee> Employees { get; set; }

public DatabaseContext(DbContextOptions<DatabaseContext> options): base(options)

{

Database.EnsureCreated();

}

Employee.cs

public class Employee

{

public int Id { get; set; }

public string Name { get; set; }

public int Salary { get; set; }

}

Program.cs

List<Employee> employees = new List<Employee>

{

new() { Id = 1, Name = "qwerty", Salary =1250},

new() { Id = 2, Name = "qwerty101", Salary =3200 },

new() { Id = 3, Name = " qwerty102", Salary =4000}

};

string connection =

builder.Configuration.GetConnectionString("DefaultConnection");

builder.Services.AddDbContext<DatabaseContext>(options => options.UseSqlServer(connection));

app.MapGet("/get", (DatabaseContext context) => context.Employees);

app.MapPost(("/post"), (DatabaseContext context) =>

{

context.Employees.Add(employee);

context.SaveChanges();

});

app.MapPut(("/put"), (DatabaseContext context) =>

{

var emp = context.Employees.FirstOrDefault(x => x.Id == employee.Id);

if (emp != null)

{

emp.Salary = employee.Salary;

emp.Name = employee.Name;

context.Employees.Update(emp);

context.SaveChanges();

}

});

app.MapDelete(("/{id}"), (DatabaseContext context, int id) =>

{

var emp = context.Employees.FirstOrDefault(x => x.Id == id);

if (emp != null)

{

context.Employees.Remove(emp);

context.SaveChanges();

}

});