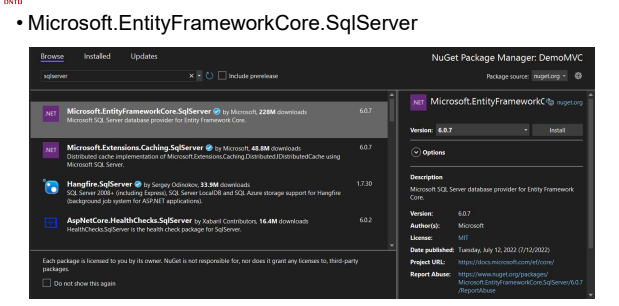
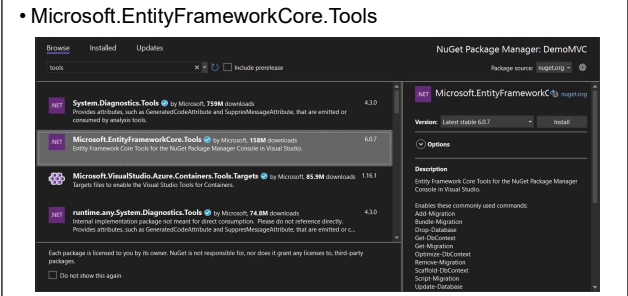
**1.Cài đặt các gói cần thiết**

Trước tiên, bạn cần cài đặt các gói NuGet cần thiết cho dự án của mình. Mở NuGet Package Manager và cài đặt các gói sau:

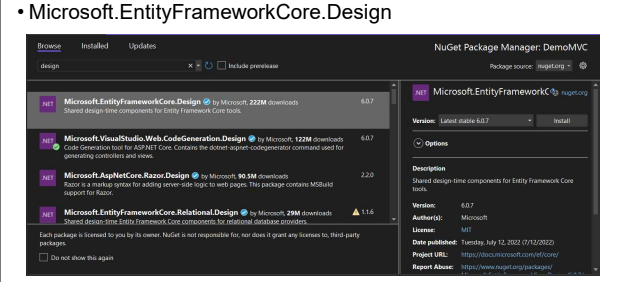
* **Microsoft.EntityFrameworkCore.SqlServer:** Gói này cung cấp các nhà cung cấp SQL Server cho Entity Framework Core.



* **Microsoft.EntityFrameworkCore.Tools:** Gói này cung cấp các công cụ cho Entity Framework Core.



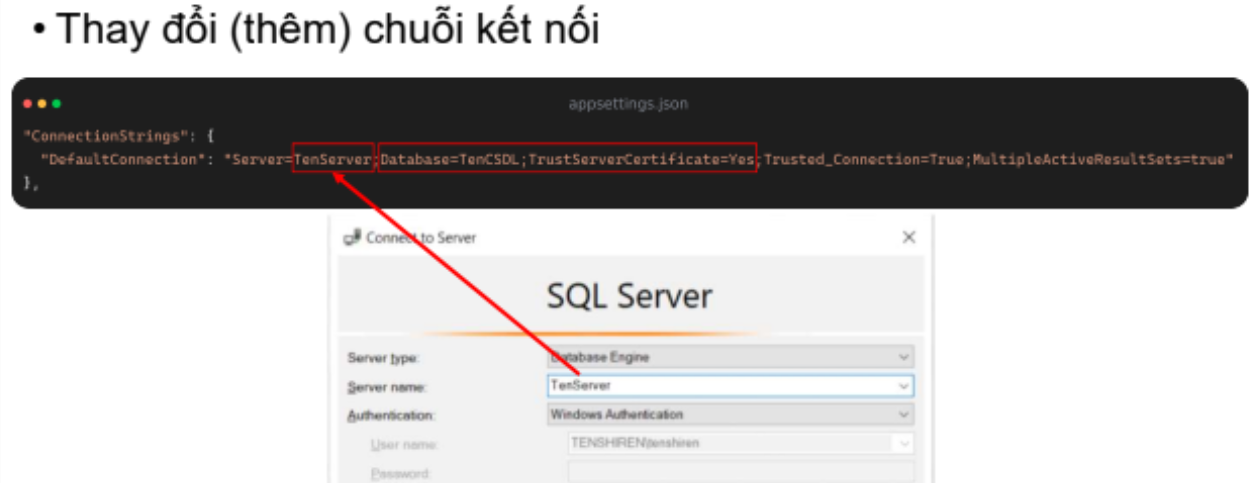
* **Microsoft.EntityFrameworkCore.Design**



* dotnet add package Microsoft.EntityFrameworkCore.SqlServer
* dotnet add package Microsoft.EntityFrameworkCore.Tools
* dotnet add package **Microsoft.EntityFrameworkCore.Design**

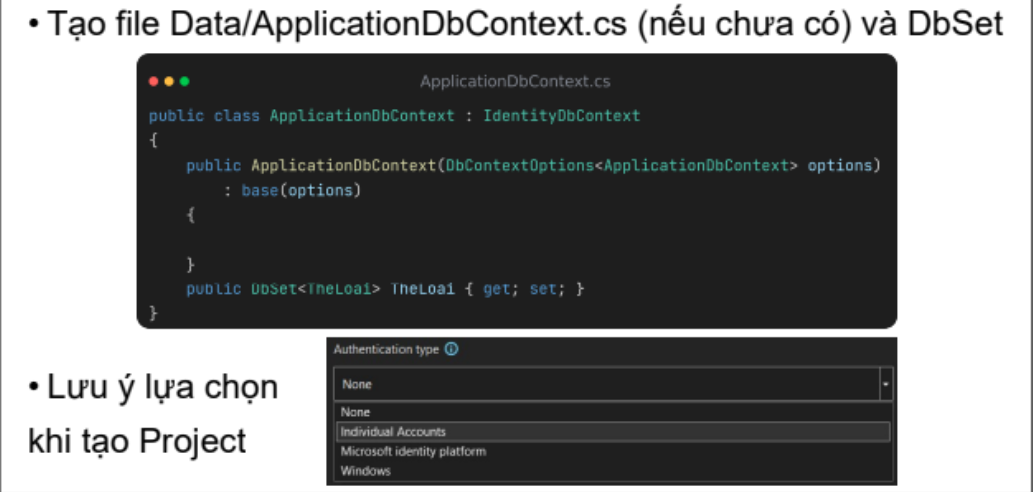
**2.Cấu hình chuỗi kết nối**.

Thêm chuỗi kết nối vào tệp appsettings.json của bạn. Chuỗi kết nối này chứa thông tin cần thiết để kết nối với cơ sở dữ liệu SQL Server.



**3.Tạo lớp DbContext**

-Tạo một lớp DbContext để quản lý kết nối và truy vấn cơ sở dữ liệu. Lớp này sẽ kế thừa từ Microsoft.EntityFrameworkCore.DbContext.



**4.Đăng ký DbContext trong DI Container**

-Trong tệp Startup.cs hoặc Program.cs (tùy thuộc vào phiên bản ASP.NET Core bạn đang sử dụng), đăng ký DbContext với Dependency Injection (DI) container.

**C#**

public void ConfigureServices(IServiceCollection services)

{

services.AddDbContext<ApplicationDbContext>(options =>

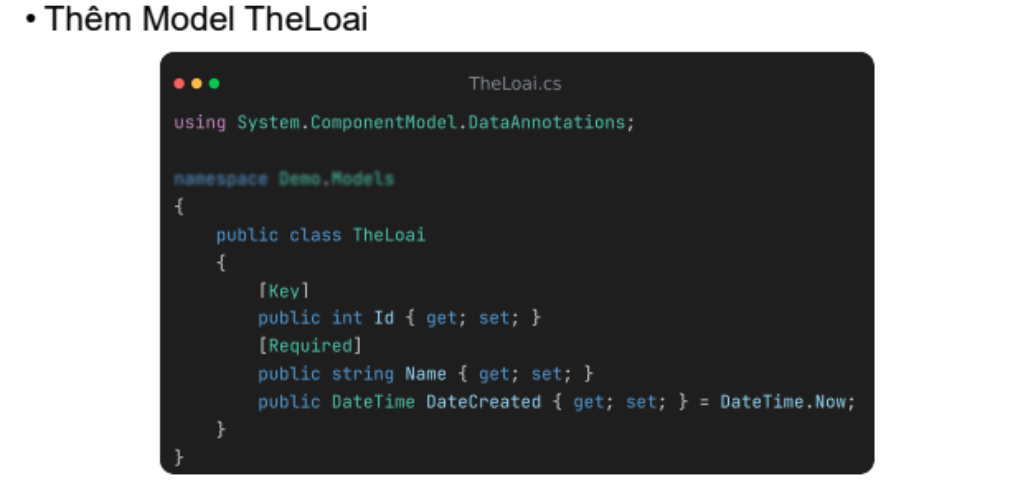
options.UseSqlServer(Configuration.GetConnectionString("DefaultConnection")));

services.AddControllersWithViews();

}

**5.Tạo các mô hình (Models)**

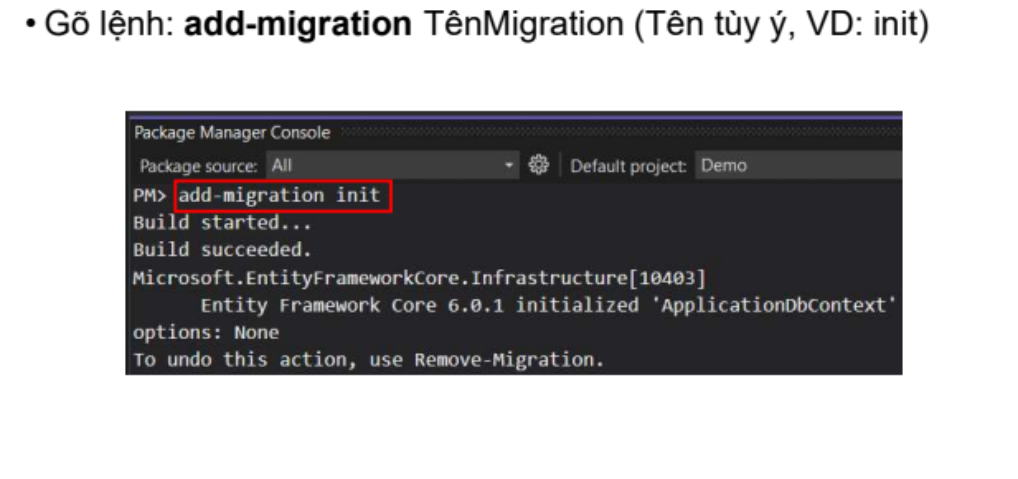
- Tạo các lớp mô hình tương ứng với các bảng trong cơ sở dữ liệu của bạn. Các lớp này sẽ được sử dụng bởi Entity Framework để ánh xạ dữ liệu giữa cơ sở dữ liệu và ứng dụng.

****

**6. Tạo và áp dụng migrations**

- Sử dụng Entity Framework Core để tạo và áp dụng migrations, giúp cập nhật cơ sở dữ liệu theo các thay đổi trong mô hình dữ liệu của bạn.

dotnet ef migrations add InitialCreate



dotnet ef database update



**7. Sử dụng DbContext trong Controllers**

- Trong các controller của bạn, sử dụng DbContext để truy vấn và thao tác dữ liệu.

**C#**

public class YourController : Controller

{

private readonly ApplicationDbContext \_context;

public YourController(ApplicationDbContext context)

{

\_context = context;

}

public async Task<IActionResult> Index()

{

var entities = await \_context.YourEntities.ToListAsync();

return View(entities);

}

}

**Vai trò của từng thành phần:**

•**NuGet Packages**: Cung cấp các thư viện cần thiết để làm việc với SQL Server và Entity Framework Core.

•**Connection String:** Chứa thông tin kết nối đến cơ sở dữ liệu.

•**DbContext**: Quản lý kết nối và truy vấn cơ sở dữ liệu.

•**Dependency Injection**: Cung cấp DbContext cho các thành phần khác trong ứng dụng.

•**Models**: Đại diện cho các bảng trong cơ sở dữ liệu.

•**Migrations:** Quản lý các thay đổi trong mô hình dữ liệu và cập nhật cơ sở dữ liệu.