**Steps to build a Sample Project**

==Install following Packages

* 1. Microsoft.AspNetCore.StaticFiles
  2. Microsoft.AspNetCore.Mvc
  3. Microsoft.AspNetCore.Razor
  4. Microsoft.EntityFrameworkCore.SqlServer
  5. Microsoft.EntityFrameworkCore.Tools
  6. Microsoft.Extensions.Configuration.Json

==Configure MVC Services: For that just write following code in startup.cs   
**services.AddMvc();**

==Now add Following line of code for middleware.

1. **loggerFactory.AddConsole();**
3. **if (env.IsDevelopment())**
4. **{**
5. **app.UseDeveloperExceptionPage();**
6. **}**
7. **app.UseStatusCodePages();//extaintion Method**
8. **app.UseStaticFiles();//extaintion Method**
9. **app.UseMvcWithDefaultRoute();//extaintion Method**

==Create 3 Folders Models, Views, Controllers

==Create Item Class

1. **public** **class** Item
2. {
3. **public** **int** ItemId { **get**; **set**; }
4. **public** **string** ItemName { **get**; **set**; }
5. **public** **string** ShortDescription { **get**; **set**; }
6. **public** **string** LongDescription { **get**; **set**; }
7. **public** **decimal** Price { **get**; **set**; }
8. **public** **string** ImageUrl { **get**; **set**; }
9. **public** **int** CategoryId { **get**; **set**; }
10. **public** **virtual** Category Category{ **get**; **set**; }
11. }

==Create Category Class

1. **public** **class** Category
2. {
3. **public** **int** CategoryId { **get**; **set**; }
4. **public** **string** CategoryName { **get**; **set**; }
5. **public** **string** Description { **get**; **set**; }
6. **public** List<Item> Items{ **get**; **set**; }
7. }

==Creating Repository

1. **interface** IItemRepository
2. {
3. IEnumerable<Item> Items { **get**; }
4. Item GetItemById(**int** itemId);
5. }

8. **public** **interface** ICategoryRepository
9. {
10. IEnumerable<Category> Categories { **get**; }
11. }

==Create AppDbContext File

1. **public** **class** AppDbContext : DbContext
2. {
3. **public** AppDbContext(DbContextOptions<AppDbContext> options) : **base**(options)
4. {
6. }
7. **public** DbSet<Category> Categories { **get**; **set**; }
8. **public** DbSet<Item> Items { **get**; **set**; }
10. }

==Adding appSetting.JSON File

Change DB Connection

==Create Constructor in Startup.cs

1. **public** Startup(IHostingEnvironment hostingEnvironment)
2. {
3. \_configurationRoot = **new** ConfigurationBuilder()
4. .SetBasePath(hostingEnvironment.ContentRootPath)
5. .AddJsonFile("appsettings.json")
6. .Build();
7. }

==Add ItemRepository

1. **public** **class** ItemRepository:IItemRepository
2. {
3. **private** **readonly** AppDbContext \_appDbContext;
5. **public** ItemRepository(AppDbContext appDbContext)
6. {
7. \_appDbContext = appDbContext;
8. }
9. **public** IEnumerable<Item> Items
10. {
11. **get**
12. {
13. **return** \_appDbContext.Items.Include(c => c.Category);
14. }
15. }
17. **public** Item GetItemById(**int** itemId)
18. {
19. **return** \_appDbContext.Items.FirstOrDefault(p => p.ItemId == itemId);
20. }
21. }

==Add CategoryRepository

1. **public** **class** CategoryRepository : ICategoryRepository
2. {
3. **private** **readonly** AppDbContext \_appDbContext;
5. **public** CategoryRepository(AppDbContext appDbContext)
6. {
7. \_appDbContext = appDbContext;
8. }
9. **public** IEnumerable<Category> Categories => \_appDbContext.Categories;
10. }

==Configure services Startup.css

1. services.AddDbContext<AppDbContext>(options=>
2. options.UseSqlServer(\_configurationRoot.GetConnectionString("DefaultConnection")));
3. services.AddTransient<ICategoryRepository, CategoryRepository>();
4. services.AddTransient<IItemRepository, ItemRepository>();

==Update Database using following commands

Add-Migration Initial

Update-database

==Insert data in database

1. INSERT INTO dbo.Categories
2. ( CategoryName, Description )
3. values('Soup','Soup'),
4. ('South Indian','South Indian'),
5. ('North Indian','North Indian'),
6. ('Chinese','Noodle'),
7. ('Snakes','Snakes')
9. GO
11. INSERT INTO dbo.Items
12. (ItemName,
13. ShortDescription,
14. LongDescription,
15. ImageUrl,
16. CategoryId,
17. Price)
18. VALUES
19. ('Tomato Soup','Tomato Soup','Tomato Soup','/Images/ts.jpg',1,60.00)
20. ,('Paper Dosa','Paper Dosa','Paper Dosa','/Images/pd.jpg',2,70.00)
21. ,('Matar Paneer','Matar Paneer','Matar Paneer','/Images/mp.jpg',3,150.00)
22. ,('Hakka Noodles','Hakka Noodles','Hakka Noodles','/Images/hn.jpg',4,200.00)
23. ,('Manchurian','Manchurian','Manchurian','/Images/m.jpg',4,60.00)
24. ,('Chilli Potato','Chilli Potato','Chilli Potato','/Images/cp.jpg',5,200.00)
25. ,('Chilli Paneer','Chilli Paneer','Chilli Paneer','/Images/pc.jpg',5,60.00)

==Create Item Controller

1. public class ItemController : Controller
2. {
3. private readonly IItemRepository \_itemRepository;
4. private readonly ICategoryRepository \_categoryRepository;
5. public ItemController(ICategoryRepository categoryRepository, IItemRepository itemRepository)
6. {
7. \_itemRepository = itemRepository;
8. \_categoryRepository = categoryRepository;
9. }
10. public ActionResult List()
11. {
12. return View(\_itemRepository.Items);
13. }
14. }

==Now add following dotnet lib in csproj to generate scaffolding template

<DotNetCliToolReference Include="Microsoft.VisualStudio.Web.CodeGeneration.Tools" Version="1.0.0-msbuild3-final" />

==Finally Generate View using Scaffolding

Done ☺