PAra usar o scaffolding é necessário realizar a instalação do code generator global/ não me pergunte porque diabos isso... .net

dotnet tool install --global dotnet-aspnet-codegenerator

instalar o dotnet core

abrir o powershell

dotnet new mvc -n <NOmeDOProjeto>

cd <NOmeDOProjeto>

code .

Instalar as extensões c# no vs code

criar as models nas pastas models

Exemplo de model para crud simples:

namespace <NOmeProjeto>.Models

{

public class LiveShow

{

public int Id { get; set; }

public string Artist { get; set; }

public string Venue { get; set; }

public DateTime ShowDate { get; set; }

public string EnteredBy { get; set; }

}

}

Exemplo de relacionamento:

namespace <NOmeProjeto>.Models

{

public class Cliente

{

[Key]

public int Id { get;set; }

[Required]

public string Nome { get;set; }

public int Idade { get;set; }

public string Email { get;set; }

public DateTime DataEntrada { get;set; }

public DateTime DataSaida { get;set; }

public int CursoId { get; set;}

public Curso Curso { get; set; }

}

}

namespace <NOmeProjeto>.Models

{

public class Curso

{

[Key]

public int Id { get;set; }

[Required]

public string Nome { get;set; }

public DateTime DataInicial { get;set; }

public DateTime DataFInal { get;set; }

[ForeignKey("CursoId")]

public virtual IList<Cliente> Clientes { get; set; }

}

}

COmando para acrescentar a referencia ao SQLite

dotnet add package Microsoft.EntityFrameworkCore.SQLite -v 2.2.3

opçãosqlserver

dotnet add package Microsoft.EntityFrameworkCore.SQLSERVER

Criar uma pasta data para a classe dbCOntext:

using <NOmeProjeto>.Models;

using Microsoft.EntityFrameworkCore;

namespace LiveMusicFinder.Data

{

public class ApplicationDbContext : DbContext

{

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

{

}

//referenciar aqui as models

public DbSet<LiveShow> LiveShows { get; set; }

}

}

Acrescentar isso ao startup.cs logo depois de services.mvc

services.AddDbContext<ApplicationDbContext>(options => options.UseSqlite("Data Source=<NOmeProjeto>.db"));

acrescentar os bibliotecas ao startup.cs

using LiveMusicFinder.Data;

using Microsoft.EntityFrameworkCore;

no terminal acrescentar a referencia ao code generator

dotnet add package Microsoft.VisualStudio.Web.CodeGeneration.Design

Criar o scaffolding mudando aqui os models e nomes dos controllers

dotnet aspnet-codegenerator controller -name LiveShowsController -async -m <NOmeProjeto>.Models.LiveShow -dc <NOmeProjeto>.Data.ApplicationDbContext -namespace Controllers -outDir Controllers -udl

Isso cria todos os controllers e views

Entrar na página views shared layout e criar os links para os novas urls

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="LiveShows" asp-action="Index">Live Shows</a>

</li>

Acrescentar isso ao startup.cs para criar os bancos de dados ao iniciar o app

public void Configure(IApplicationBuilder app, IHostingEnvironment env, ApplicationDbContext dbContext)

{

dbContext.Database.EnsureCreated();

// ...the rest of the method

é necessário criar uma mod na classe cursos controller para mostrar os clientes na tela.