using Microsoft.EntityFrameworkCore;

using RepositoryPatternDemo.Context;

namespace RepositoryPatternDemo.Models

{

public class Role

{

public int RoleID { get; set; }

public string RoleName { get; set; }

}

}

Add DbInitialize class

internal static class DbInitializerExtension

{

public static IApplicationBuilder UseItToSeedSqlServer(this IApplicationBuilder app)

{

ArgumentNullException.ThrowIfNull(app, nameof(app));

using var scope = app.ApplicationServices.CreateScope();

var services = scope.ServiceProvider;

try

{

var context = services.GetRequiredService<CourseDbContext>();

DbInitializer.Initialize(context);

}

catch (Exception ex)

{

}

return app;

}

}

internal class DbInitializer

{

internal static void Initialize(CourseDbContext dbContext)

{

ArgumentNullException.ThrowIfNull(dbContext, nameof(dbContext));

dbContext.Database.EnsureCreated();

if (dbContext.Roles.Any()) return;

var roles = new Role[]

{

new Role{ RoleName = "Admin" },

new Role{ RoleName = "HRAdmin" }

//add other users

};

foreach (var user in roles)

dbContext.Roles.Add(user);

dbContext.SaveChanges();

}

}

}

Program.cs file

builder.Services.AddDbContext<CourseDbContext>(x => x.UseSqlServer(builder.Configuration["ConnectionStrings:MyCon"]));

builder.Services.AddScoped<DbInitializer>();

var app = builder.Build();

app.UseItToSeedSqlServer();