GUIDE DOTNET CORE 5

PACKAGES NUGETS

EntityFrameworkCore (et non EntityFramework)
EntityFrameworkCoreTools
Microsoft.EntityFrameworkCore.SqlServer
Microsoft.EntityFrameworkCore.Sqlite
AspNetCore.Diagnostics.EntityFrameworkCore

DATA CONTEXT

1 - Créer votre modèle (dans le répertoire Models)

```
public class Pizza
{
    public int PizzaID { get; set; }
    ....
}
```

2 - Créer un répertoire Data, puis nouvelle classe DataContext

```
public class DataContext : DbContext
{
    public DataContext(DbContextOptions<DataContext> options) : base(options)
    {
    }
    public DbSet<Pizza> Pizzas { get; set; }
}
```

3 - Dans configureServices :

```
services.AddDbContext<DataContext>(options =>
            options.UseSqlite(Configuration.GetConnectionString("DefaultConnection")));
```

4 - Dans appsettings.json:

```
  "ConnectionStrings": {
    "DefaultConnection": "Data Source=PizzaMama.db"
  },
```

INTIALISATION DE LA DB

Dans program.cs :

```
public static void Main(string[] args)
{
   //CreateHostBuilder(args).Build().Run();
   var host = CreateHostBuilder(args).Build();
   CreateDbIfNotExists(host);
   host.Run();
}

private static void CreateDbIfNotExists(IHost host)
{
   using (var scope = host.Services.CreateScope())
   {
      var services = scope.ServiceProvider;
      try
      {
         var context = services.GetRequiredService<DataContext>();
         context.Database.EnsureCreated();
      }
      catch (Exception ex)
      {
         var logger = services.GetRequiredService<ILogger<Program>>();
         logger.LogError(ex, "An error occurred creating the DB.");
      }
   }
}
```

EXEMPLE D'UTILISATION DE LA DB

```
var pizza = new Pizza() { nom = "Pizza test", prix = 5 };
dataContext.Pizzas.Add(pizza);
dataContext.SaveChanges();
```

OUTIL D'ANALYSE DB SQLITE

https://sqlitebrowser.org

AUTHENTIFICATION COOKIE

1 - Formulaire (dans /Admin/Index)

```
<form method="post">
    <input type="text" name="username" placeholder="Login">
    <input type="password" name="password" placeholder="Mot de passe">
    <button>Se connecter</button>
</form>
```

2 - ConfigureServices

Avant AddRazorPages :

```
services.AddAuthentication(CookieAuthenticationDefaults.AuthenticationScheme)
.AddCookie(options =>
{
    options.LoginPath = "/Admin";
});
```

3 - Configure

```
app.UseAuthentication();
app.UseAuthorization();
```

4 - Authorize

Rajouter [Authorize] dans vos PageModel

5 - OnPost

```
public async Task<IActionResult> OnPost(string username, string password, string ReturnUrl)
{
    if (username == "admin")
    {
        var claims = new List<Claim>
        {
            new Claim(ClaimTypes.Name, username)
        };
        var claimsIdentity = new ClaimsIdentity(claims, "Login");
        await HttpContext.SignInAsync(CookieAuthenticationDefaults.AuthenticationScheme, new
ClaimsPrincipal(claimsIdentity));
        return Redirect(ReturnUrl == null ? "/Admin/Pizzas" : ReturnUrl);
    }
    return Page();
}
```

6 - OnGet

```
public IActionResult OnGet()
{
    Console.WriteLine(HttpContext.User);
    if (HttpContext.User.Identity.IsAuthenticated)
    {
        return Redirect("/Admin/Pizzas");
    }
    return Page();
}
```

7 - Logout

```
<a class="nav-link text-dark" asp-area="" asp-page-handler="Logout" asp-page="/Admin/Index">Se déconnecter</a>
```

```
public async Task<IActionResult> OnGetLogout()
{
    await HttpContext.SignOutAsync();
    return Redirect("/Admin");
}
```

8 - IConfiguration

AppSettings.json

```
"Auth": {
"AdminLogin": "admin",
"AdminPassword":  "1234"
},
```

```
var credentials = configuration.GetSection("Auth");
if (username == credentials["AdminLogin"] && password == credentials["AdminPassword"])
```

CULTURE

Dans configure() :

```
var cultureInfo = new CultureInfo("fr-FR");
cultureInfo.NumberFormat.NumberDecimalSeparator = ".";
CultureInfo.DefaultThreadCurrentCulture = cultureInfo;
CultureInfo.DefaultThreadCurrentUICulture = cultureInfo;
```