Opgave: håndtering af brugere og login

Baggrund: Opgave om Running app (løbeklub-'Løb For Livet'), se tidligere opgaver:

- Start model klasse
- Unit Test
- Razorpages
- Koblet til database
- Sortering

Du skal tage udgangspunkt i de tidligere opgaver for running App – Du kan fortsat arbejde som par-programmering : se evt. en løsning her https://github.com/rf23da2b1-1b/RunningApp

Brugere og login

På 1.semester havde vi om, hvorledes vi kunne lave en login side, samt styre det (se projektet https://github.com/rf23da2b1-1b/ClassDemoTestLogin)

Der var bare en den ulempe at den 'kun' virkede for én bruger, så hvis vi har flere brugere til systemet samtidigt må vi ty til noget andet.

Der findes flere løsninger – her bliver du ført igennem, hvorledes du kan anvende Sessions, dette kan du så også benytte til fx indkøbskurve eller lignende.

Baggrund for sessions se

- Wikipedia: https://en.wikipedia.org/wiki/HTTP#HTTP_session
- Learn Razor Pages: https://www.learnrazorpages.com/razor-pages/session-state

Opgave 1: Opsæt Razor Page til brug af sessions

Du skal i program.cs understøtte at din applikation skal benytte Session.

Under builder tilføj:

```
builder.Services.AddSession();
```

og under app tilføj

```
app.UseSession();
```

I mappen Services lav en klasse SessionHelper, der her tre metoder
Du skal lave metoderne generiske, så de kan benyttes til at gemme objekter af forskellige klasser.

```
public static class SessionHelper

    // henter et objekt af typen T fra session
    public static T Get<T>(HttpContext context)

    // gemmer et objekt af typen T i session
    public static void Set<T>(T t, HttpContext context)

    // fjerner et objekt af type T fra session
    public static void Clear<T>(HttpContext context)
```

I Get-metoden skal du udfra HttpContext – session hente en string og json deserialisere den til et 'T'-objekt

```
String sessionName = typeof(T).Name;
String? s = context.Session.GetString(sessionName);
if (string.IsNullOrWhiteSpace(s))
{
    throw new ArgumentException($"No session value for {sessionName}");
}
return JsonSerializer.Deserialize<T>(s);
```

I Set metoden skal du serialisere 'T'-objektet og gemme det i session objektet

```
String sessionName = typeof(T).Name;
String s = JsonSerializer.Serialize(t);
context.Session.SetString(sessionName, s);
```

Mens Clear metoden fjerner et objekt af type T fra session

```
context.Session.Remove(typeof(T).Name);
```

Opgave 2: Lav en User og et UserRepository

Du tager udgangspunkt i Member og MemberReporitory og tilføjer password til en member.

Eller du skal lave en klasse User (Bruger), der indeholder (properties) UserName, der er unikt, Fuld Navn og Kodeord, samt Rolle f.eks. IsAdmin. (der er også konstruktører og ToString)

Du skal lave et UserRepository, der kan indeholde en liste af brugere (User), samt indeholder metoderne:

- void Add(User user)
- User Delete(String UserName)
- Bool CheckUser(String userName, String password)

Indsæt nogle brugere som testdata (mock data)

I konfigureringsfilen program.cs, tilføj din UserRepository (Du kan evt. lave et interface til UserRepository) som en singleton.

Opgave 3: Understøt login og logout

Du skal lave en ny mappe under pages fx logins, hvor du skal lave to Razor pages Login hhv. Logout

På **Login-siden** skal du lave to felter (UserName og Password) og en knap til at logge ind.

Lav de sædvanlige tjek af felterne og tjek mod UserRepository om det er en kendt bruger.
Hvis brugeren er godkendt husk at gemme brugeren i session.

```
SessionHelper.Set(user, HttpContext)
```

På **Logout-siden** skal du 'bare' lave noget kode i OnGet(), hvor du skal fjerne brugeren fra session

```
SessionHelper.Clear<User>(HttpContext);
```

Hvorefter brugeren dirigeres tilbage til index-siden.

Opgave 4: Tjek om brugeren er logget ind

Prøv på en side, som brugeren ikke kan tilgå uden at være logget ind, at lave et tjek og dirigere brugeren til login-siden, hvis vedkommende ikke er logget ind.

Prøv fx på 'members/index' at lave et tjek:
I filen Index.cshtml.cs i OnGet() som det første :

```
try
{
   User = SessionHelper.Get<User>(HttpContext);
   return Page();
}
catch (ArgumentException ne)
{
   return RedirectToPage("/Logins/Login");
}
```

Så om det virker, ved at prøve at komme ind på siden uden at logge ind.

Sørg for at dette tjek ligger på alle sider, hvor brugeren skal være logget ind for at kunne benytte siden.