Uniwersytet Bielsko-Bialski

LABORATORIUM

Programowanie dla Internetu w technologii ASP.NET

Sprawozdanie nr 7

Uproszczona autoryzacja

GRUPA: 2B / SEMESTR: 6 / ROK: 3

Kacper Lizak / 59443

Cel ćwieczenia

Celem ćwiczenia było zapoznanie się z podstawami uwierzytelniania i autoryzacji użytkowników w aplikacjach tworzonych za pomocą ASPNET Core Identity.

Wprowadzenie

Uwierzytelnienie to proces polegający na potwierdzeniu zadeklarowanej tożsamości podmiotu biorącego udział w procesie komunikacji. W praktyce odbywa się to poprzez porównanie przedstawionych przez użytkownika dowodów tożsamości z danymi przechowywanymi w systemie. Celem uwierzytelniania jest uzyskanie określonego poziomu pewności, że dany podmiot jest w rzeczywistości tym, za którego się podaje.

Autoryzacja to proces nadawania podmiotowi dostępu do zasobu. Celem autoryzacji jest kontrola dostępu, która potwierdza, czy dany podmiot jest uprawniony do korzystania z żądanego zasobu. Autoryzacja następuje dopiero po potwierdzeniu tożsamości podmiotu za pomocą identyfikacji i uwierzytelnienia.

Microsoft.AspNetCore.Identity to interfejs API, który obsługuje funkcje logowania interfejsu użytkownika. Zarządza użytkownikami, hasłami, danymi profilu, rolami, oświadczeniami, tokenami, potwierdzeniem wiadomości e-mail i nie tylko. Użytkownicy mogą utworzyć konto przy użyciu informacji logowania przechowywanych w Identity lub mogą użyć zewnętrznego dostawcy logowania. ASP.NET Core Identity dodaje funkcje logowania interfejsu użytkownika do aplikacji internetowych platformy ASP.NET Core

Wykonanie ćwiczenia

W pierwszej kolejności należy zmienić dziedziczenie klasy TripContext z DbContext (EF Core) na IdentityDbContext (ASPNET Core Identity)

```
public class TripContext : IdentityDbContext<IdentityUser>
{
    public DbSet<Trip> Trips { get; set; }
    public DbSet<Client> Clients { get; set; }
    public DbSet<Reservation> Reservations { get; set; }
    public TripContext(DbContextOptions<TripContext> options) : base(options)
    {
    }
```

```
    protected override void OnModelCreating(ModelBuilder modelBuilder)
    {
        base.OnModelCreating(modelBuilder);
        modelBuilder.Entity<Trip>().ToTable("Trips");
        modelBuilder.Entity<Client>().ToTable("Clients");
        modelBuilder.Entity<Reservation>().ToTable("Reservations");
    }
}
```

Następnie za pomocą scaffoldingu wygenerować widoki dla następujących zachwań :

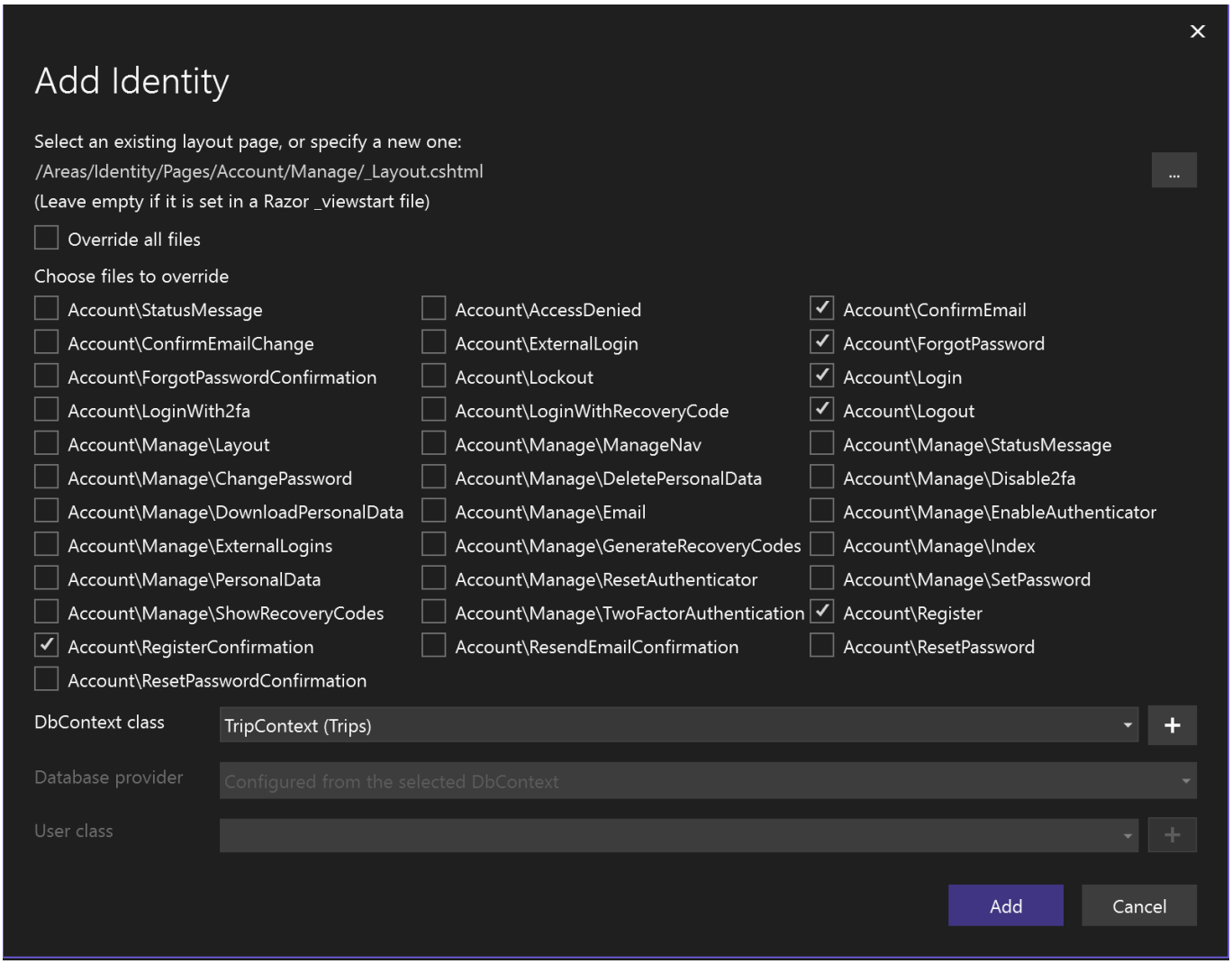
1.Rejestracja

2.Logowanie

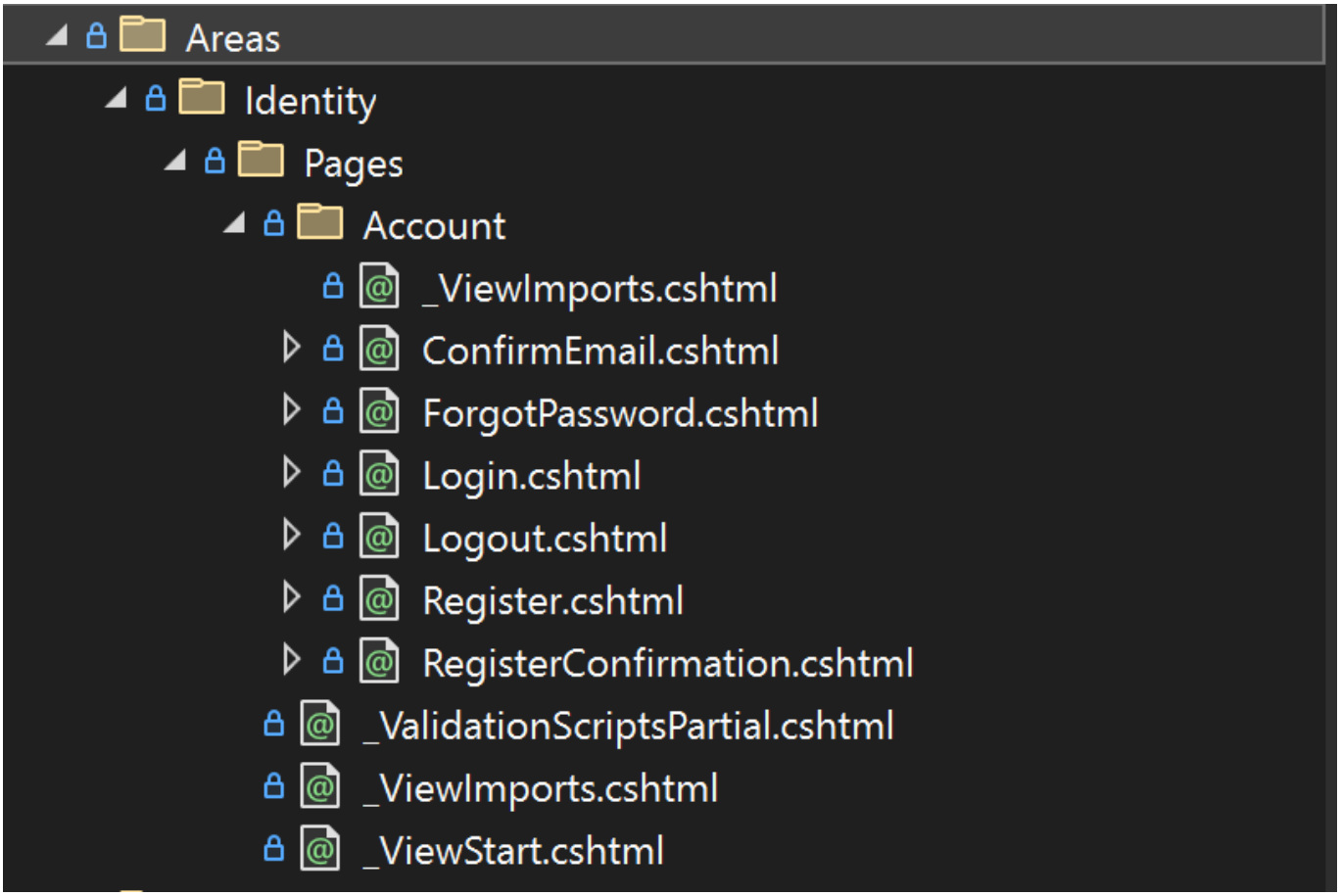
3.Wylogowywanie

4.Potwierdzenie rejestracji

5.Przypomnienie hasła



Co zostało wygenerowane :

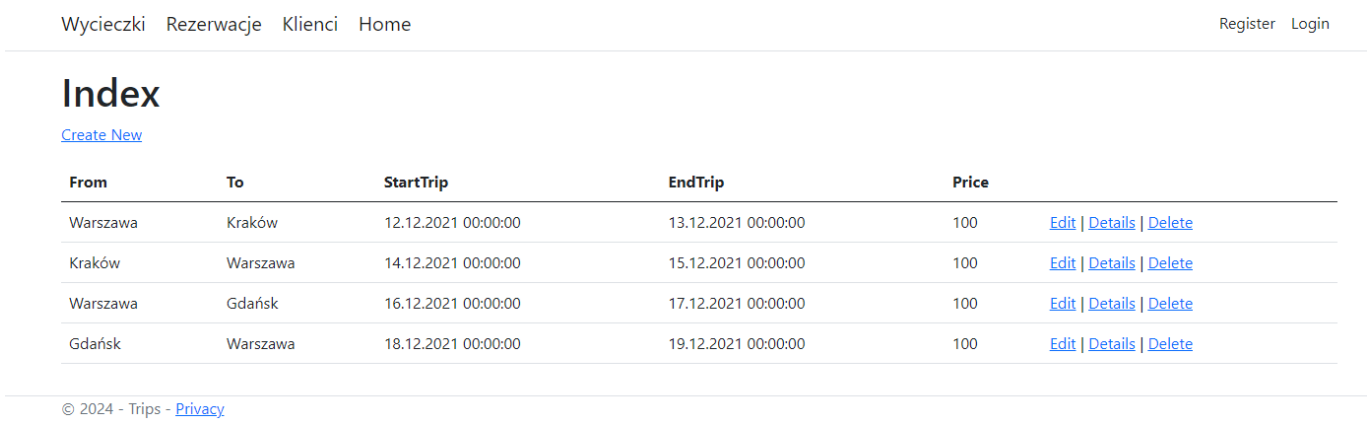


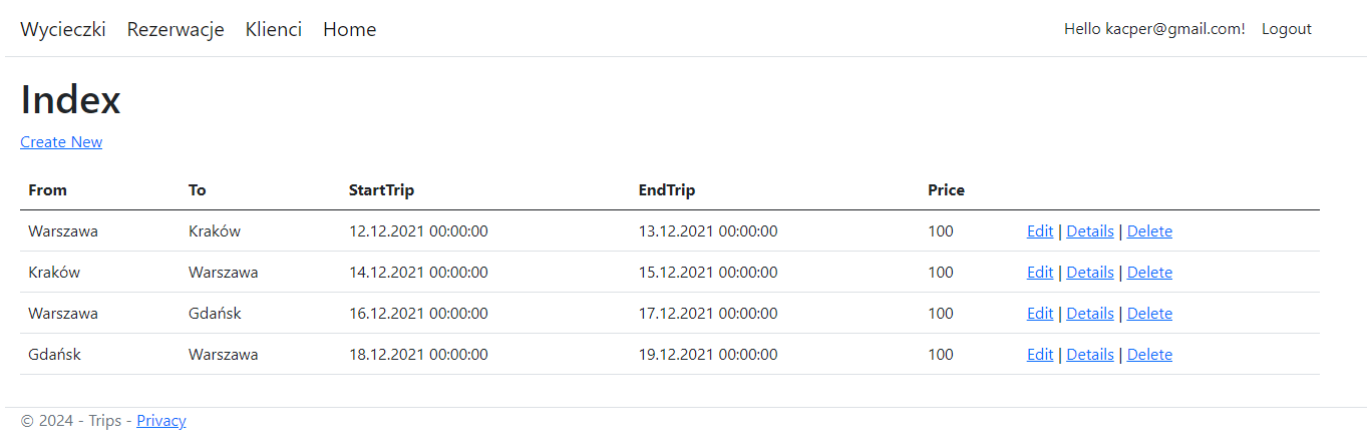
Mapowanie mechanizmuu uwierzytelniania, autoryzacji oraz mapowania stron blazor.

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

Przykład użycia udostępniania dostępu do danego kontrolera osobom nieuwierzytelnionym

```
    [AllowAnonymous]
    public class TripsController : Controller
```



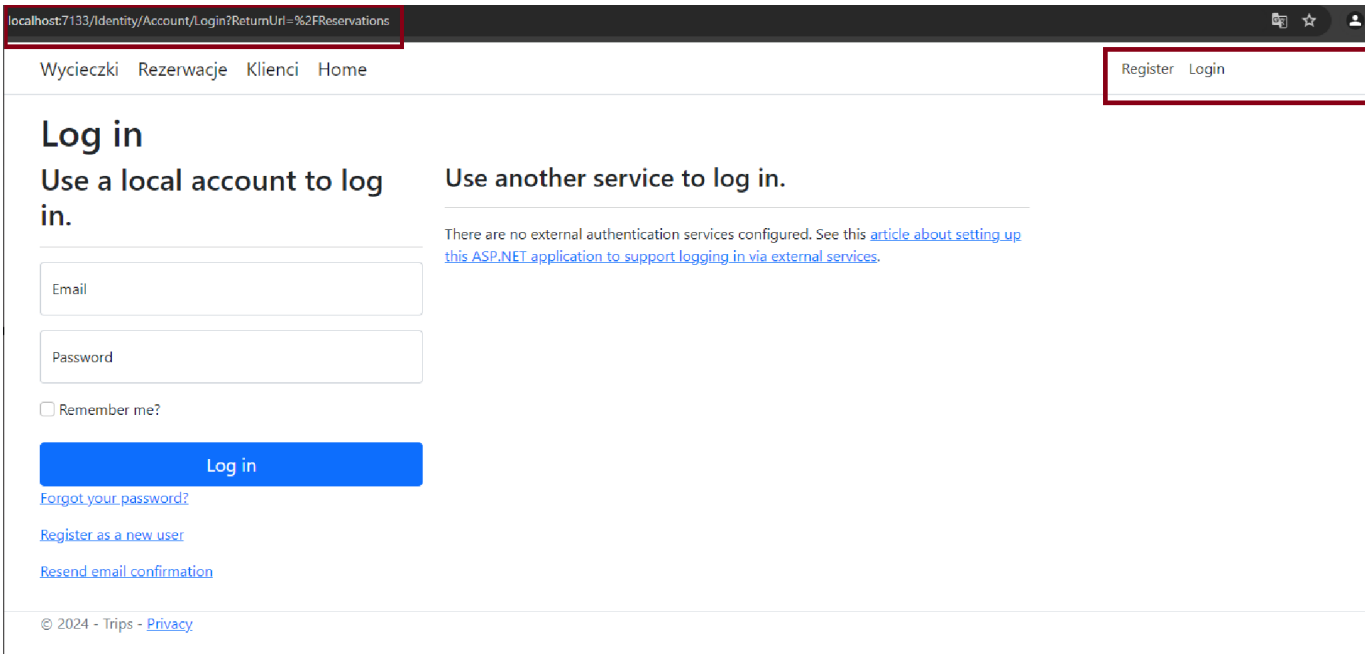


Jak widać użytkownik niezależnie od tego czy jest zalogowany czy nie widzi dokładnie to samo.

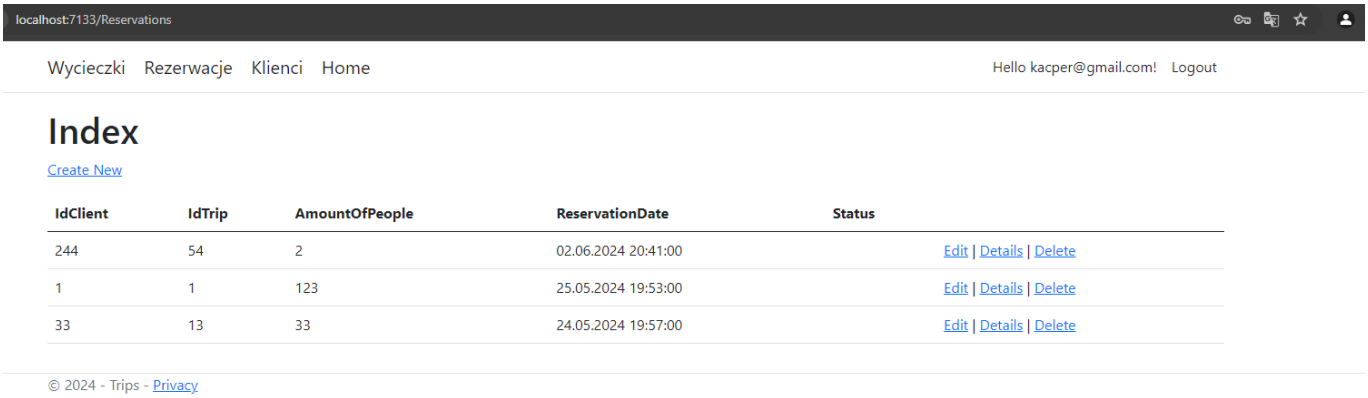
Przykład który niezalogowanemu użytkownikowi nie pozwoli zobaczyć rezerwacji:

```
[Authorize]
public class ReservationsController : Controller
```

Widok dla niezalogowanego użytkownika :



Widok dla zalogowanego użytkownika :



Wnioski

Ćwiczenie to pozwoliło na zrozumienie i praktyczne zastosowanie uwierzytelniania i autoryzacji w aplikacjach ASP.NET Core za pomocą ASPNET Core Identity. Zabezpieczenie dostępu do stron aplikacji poprzez implementację uwierzytelniania i autoryzacji jest kluczowym elementem w tworzeniu bezpiecznych aplikacji internetowych.