

Kalendarz z funkcją prowadzenia współdzielonych planów dnia

Autorzy

Jakub Sell

Wojciech Stala

Opis działania aplikacji

Aplikacja ma być podstawowym planerem, który umożliwi organizację dla jednej i wielu osób. Będzie zawierać kalendarz osobisty oraz grupowy. Aplikacja w widoku dla niezalogowanych użytkowników będzie posiadać filtry z możliwością nie pokazywania treści tworzonych przez użytkowników. Po zalogowaniu będzie możliwe przeglądanie kalendarzy oraz modyfikowanie ich.

Specyfikacja wykorzystanych technologii

- *Backend*: ASP.NET Core MVC
- *Baza danych*: Entity Framework Core z Microsoft SQL Server
- *Frontend*: Razor Pages, HTML5, CSS3, Bootstrap – dla estetycznej i responsywnej warstwy prezentacji
- *Autentykacja i autoryzacja*: ASP.NET Core Identity
- *Środowisko*: .NET 8.0 lub nowszy

Instrukcje pierwszego uruchomienia projektu

1. Sklonuj repozytorium lub pobierz paczkę z kodem źródłowym.
2. Upewnij się, że posiadasz zainstalowane:
 - .NET 8.0 lub nowszy
 - SQL Server (lokalnie lub w chmurze)
3. Otwórz projekt w Visual Studio.
4. Wykonaj w konsoli zarządzania pakietami Nuget polecenie :
`update-database`
5. Uruchom aplikację (F5 w Visual Studio lub `dotnet run` w terminalu).

Opis struktury projektu

- **Controllers** – kontrolery obsługujące żądania HTTP, logikę biznesową i przepływ danych między modelem a widokami.
- **Models** – modele danych (klasy encyjne dla EF Core oraz ViewModele do prezentacji w widokach).
- **Views** – widoki Razor (.cshtml), zawierające kod HTML/CSS oraz fragmenty Razor do dynamicznego generowania treści.
- **Data** – konfiguracja dostępu do bazy danych, klasa DbContext i migracje EF.
- **Program.cs** – konfiguracja usług, routingu oraz middleware aplikacji.

Wymienione wszystkie modele

- `CalendarElement` (reprezentuje pojedyncze wydarzenie: Id, Name, Description, StartDate, EndDate, UserId, GroupId)
- `CalendarGroup` (reprezentuje grupę z wydarzeniami: Id, Name, FounderId, kolekcje użytkowników)
- `CalendarDayViewModel` (ViewModel do wyświetlania dni w kalendarzu: Date, lista EventNames)
- `ErrorViewModel`: Model pomocniczy do wyświetlania informacji o błędach. Przechowuje np. RequestId, co ułatwia diagnozowanie problemów.
- `HomePage`: Model reprezentujący zawartość strony głównej aplikacji. Może obejmować elementy wprowadzenia, skrót funkcjonalności czy linki do głównych sekcji.

Wymienione kontrolery wraz z metodami

- **HomeController:**
 - `Index(int? month, int? year)` – wyświetla kalendarz dla danego (lub bieżącego) miesiąca i roku
 - `Calendar()` – pokazuje kalendarz dla zalogowanego użytkownika
 - `Privacy()` – wyświetla politykę prywatności
 - `Error()` – strona błędów
- **CalendarElementController:**
 - `Index()` – lista wszystkich widocznych dla użytkownika wydarzeń
 - `Details(int id)` – szczegóły konkretnego wydarzenia
 - `Create()` – formularz tworzenia nowego wydarzenia
 - `Delete(int id)` – usuwanie wydarzenia
- **CalendarGroupController:**
 - `Index()` – lista grup
 - `Details(int id)` – szczegóły wybranej grupy
 - `Create()` – tworzenie nowej grupy
 - `Edit(int id)` – edycja grupy
 - `Delete(int id)` – usuwanie grupy

Opis systemu użytkowników

System korzysta z ASP.NET Core Identity, umożliwiając użytkownikom rejestrację i logowanie. Każdy zalogowany użytkownik może należeć do jednej lub wielu grup oraz posiada własny kalendarz. Grupy kalendarzowe pozwalają na współdzielenie wydarzeń pomiędzy członkami. Uprawnienia mogą być ograniczane do członków grup lub administratorów.

Krótką charakterystyka najciekawszych funkcjonalności

- *Współdzielone kalendarze grupowe:* Pozwalają na planowanie wydarzeń w zespołach, rodzinach czy grupach projektowych.
- *Różne poziomy widoczności:* Niezalogowani widzą jedynie pusty kalendarz, zaś zalogowani mogą korzystać z pełnej funkcjonalności, w tym prywatnych i grupowych kalendarzy.
- *Nawigacja między miesiącami:* Za pomocą przycisków (strzałek) w interfejsie można przełączać się pomiędzy poszczególnymi miesiącami, co ułatwia przeglądanie planów.