

Platformy programistyczne - sprawozdanie

Kalendarz z planerem wydarzeń

Jakub Piekarek 264202

Marcin Ilski 264272

15 czerwca 2024

Spis treści

1	Opis projektu	2
2	Opis funkcjonalności	2
2.1	Kalendarz	2
2.2	Obsługa zdarzeń i lokalizacji	3
2.3	Obsługa użytkowników	3

1 Opis projektu

Zrealizowaną aplikacją jest kalendarz z planerem wydarzeń i autoryzacją użytkowników. Aplikację wykonaliśmy w ASP.NET Core w architekturze MVC. Funkcje aplikacji:

- Rejestracja nowego użytkownika i logowanie się jako istniejący użytkownik.
- Autoryzacja użytkowników (tylko użytkownik, który stworzył Event może nim zarządzać).
- Dodawanie, edycja, usuwanie eventów i lokalizacji.
- Wyświetlanie eventów w kalendarzu.

2 Opis funkcjonalności

2.1 Kalendarz

Jako podstawę aplikacji wykorzystaliśmy bibliotekę JavaScript FullCalendar.io, która pozwala na tworzenie interaktywnych kalendarzy na stronach internetowych. Biblioteka ta jest łatwo integrowalna z różnymi frameworkami oraz wydajna, dlatego idealnie wpasowała się w wymagania projektowe.



Rysunek 1: FullCalendar.

2.2 Obsługa zdarzeń i lokalizacji

Operacje zarządzające zdarzeniami i lokalizacjami zdefiniowaliśmy w interfejsie IDAL oraz zaimplementowaliśmy w klasie DAL (Data Access Layer). Metody te umożliwiają pobieranie, tworzenie, aktualizowanie i usuwanie obiektów.

Opis Metod

- **GetEvents:** Zwraca listę wszystkich wydarzeń z bazy danych.
- **GetMyEvents:** Zwraca listę wydarzeń przypisanych do konkretnego użytkownika na podstawie jego *userid*.
- **GetEvent:** Zwraca pojedyncze wydarzenie na podstawie jego *id*.
- **CreateEvent:** Tworzy nowe wydarzenie na podstawie danych z formularza (*IFormCollection*), pobiera lokalizację i użytkownika, tworzy nowe wydarzenie i zapisuje je w bazie danych.
- **UpdateEvent:** Aktualizuje istniejące wydarzenie na podstawie danych z formularza, pobiera aktualne dane lokalizacji i użytkownika, a następnie zapisuje zmiany w bazie danych.
- **DeleteEvent:** Usuwa wydarzenie na podstawie jego *id*.
- **GetLocations:** Zwraca listę wszystkich lokalizacji z bazy danych.
- **GetLocation:** Zwraca pojedynczą lokalizację na podstawie jej *id*.
- **CreateLocation:** Tworzy nową lokalizację i zapisuje ją w bazie danych.

Kontrolery

Kontrolery w aplikacji ASP.NET Core MVC są odpowiedzialne za obsługę żądań HTTP, przetwarzanie danych i zwracanie odpowiednich odpowiedzi. W aplikacji utworzyliśmy 3 kontrolery:

- **EventController.cs:** Obsługuje operacje CRUD (Create, Read, Update, Delete) dla wydarzeń.
- **HomeController.cs:** Zarządza stronami ogólnymi i podstawowymi funkcjami aplikacji.
- **LocationController.cs:** Obsługuje operacje CRUD dla lokalizacji.

2.3 Obsługa użytkowników

ApplicationDbContext dziedziczy po specjalnej klasie IdentityDbContext, która zawiera wszystkie niezbędne konfiguracje i zestawy danych do obsługi tożsamości i zarządzania użytkownikami.

Dodatkowo utworzyliśmy ActionFilter **UserAccessOnly**, który sprawdza czy w żądaniu HTTP zawiera id eventu. Jeżeli tak, sprawdza czy aktualnie zalogowany użytkownik jest tym samym użytkownikiem, który utworzył dany event i na tej podstawie przyznaje dostęp lub nie. ActionFilter wykorzystywany jest w EventControllerze.