



Wydarzeniownik

Programowanie zaawansowane

Jakub Górnaś, Maksymilian Mieczkowski
K79

Spis treści

1. Opis działania projektu.....	2
Podstawowe funkcjonalności:	2
Rejestracja użytkowników:	2
Logowanie:	2
Tworzenie i edycja wydarzeń:.....	2
Rodzaje użytkowników:	2
Możliwości użytkowników:	3
Przeglądanie wydarzeń:.....	3
2. Instrukcja uruchomienia projektu	3
Pobranie projektu	3
Aktualizacja bazy danych	3
Rejestracja konta użytkownika	3
Dodanie nowego wydarzenia	3
Edycja lub usunięcie wydarzenia.....	4
Filtrowanie listy wydarzeń	4
Komentowanie wydarzeń	4
Polubienia.....	4
Deklaracja uczestnictwa w wydarzeniu	4
3. Struktura projektu	5
Wzorce projektowe:.....	5
MVC (Model – Widok – Kontroler).....	5
Obserwator	5
Strategia	5
Metoda szablonowa	5
Opis komponentów projektu:	6
Kontroler EventsController.cs	6
Model Event.cs.....	7
Model Comments.cs	7
Model UserLike.cs	8
Model EventRegistration.cs	8
Widoki Events.....	8
Widoki Shared	8

Opis systemu użytkowników:	9
Użytkownik niezalogowany	9
Użytkownik zalogowany	9
Wykorzystane technologie:	9

1. Opis działania projektu.

Wydarzeniownik to aplikacja typu **Social Media**, która służy dzieleniu się informacjami o najnowszych wydarzeniach w okolicy. Głównym celem aplikacji jest umożliwienie użytkownikom tworzenie, śledzenie i organizowanie wydarzeń które następnie umieszcza na stronie głównej, gdzie widoczne są dla innych użytkowników przeglądających portal. Poprzez zapostowane wydarzenia użytkownicy mogą wchodzić ze sobą w interakcje.

Podstawowe funkcjonalności:

Rejestracja użytkowników:

- Użytkownik może utworzyć konto, podając podstawowe dane (login, e-mail i hasło).
- Weryfikacja danych wejściowych (np. sprawdzenie unikalności adresu e-mail).
- Zabezpieczenie hasła za pomocą odpowiedniego algorytmu.

Logowanie:

- Zarejestrowany użytkownik może zalogować się na swoje konto, podając dane uwierzytelniające.
- Obsługa błędnych prób logowania.
- Sesja użytkownika która pozwala na zachowanie stanu zalogowania.

Tworzenie i edycja wydarzeń:

- Dodawanie nowego wydarzenia z tytułem, opisem, terminem, lokacją i uczestnikami.
- Edytowanie istniejących wydarzeń.
- Usuwanie wydarzeń z listy.

Rodzaje użytkowników:

- Gość posiada tylko możliwość rejestracji.
- Użytkownik może tworzyć zadania i zarządzać nimi oraz swoim kontem.

Możliwości użytkowników:

- Podgląd wszystkich wydarzeń.
- Komentowanie wydarzeń.
- Reakcja na wydarzenia.
- Deklaracje uczestnictwa w wydarzeniu.

Przeglądanie wydarzeń:

- Możliwość przeglądania wydarzeń na stronie głównej.
- Podgląd wszystkich swoich wydarzeń na stronie Moje Wydarzenia.

2. Instrukcja uruchomienia projektu

Pobranie projektu

- Projekt należy pobrać z repozytorium GitHub
<https://github.com/Maksymilian/Wydarzeniownik>
do wybranej lokalizacji.

Aktualizacja bazy danych

- Przed pierwszym uruchomieniem projektu należy zaktualizować bazę danych SQL komendą
UPDATE-DATABASE

Rejestracja konta użytkownika

- Po uruchomieniu projektu należy utworzyć konto wybierając opcję
Zarejestruj się
Następnie użytkownik podaje swój email i hasło.
Po utworzeniu konta należy je **potwierdzić** klikając w link na przekierowanej stronie.

Dodanie nowego wydarzenia

- Zalogowany użytkownik ma możliwość dodania nowego wydarzenia klikając w przycisk
Dodaj nowe wydarzenie na stronie głównej – **Wszystkie wydarzenia**
a następnie podając Tytuł, datę, dokładny czas, opis, lokalizację i zdjęcie i klikając
Utwórz
dodaje wydarzenie do listy.

Edycja lub usunięcie wydarzenia

- Zalogowany użytkownik ma możliwość edycji lub usunięcia wydarzeń dodanych przez siebie poprzez wejście w **Szczegóły** a następnie klikając w **Edytuj** lub **Usuń**.

Filtrowanie listy wydarzeń

- Zalogowany użytkownik ma możliwość przefiltrowania listy wydarzeń i wyświetlenia jedynie swoich wydarzeń wchodząc na stronę **Moje Wydarzenia** dostępną na pasku.
- Zalogowany użytkownik ma możliwość sortowania listy wydarzeń według daty:
 - od najnowszych
 - od najstarszych.

Komentowanie wydarzeń

- Zalogowany użytkownik ma możliwość dodania komentarza do istniejącego wydarzenia.

Polubienia

- Zalogowany użytkownik ma możliwość polubienia istniejącego wydarzenia.

Deklaracja uczestnictwa w wydarzeniu

- Zalogowany użytkownik może zadeklarować swoje uczestnictwo w istniejącym wydarzeniu dodając się do listy zadeklarowanych.

3. Struktura projektu

Wzorce projektowe:

MVC (Model – Widok – Kontroler)

- Służy do organizowania struktury aplikacji posiadających graficzne interfejsy użytkownika.
- Oddziela warstwę logiki (Model), interfejsu użytkownika (View) i kontrolera (Controller).
- **Model:** Klasa Event oraz modele Comments, UserLikes i EvenRegistration przechowują dane i reguły walidacji.
- **Widok:** Widoki MyEventsViewer, EventsViewer, Details oraz widoki współdzielone renderują dane dostarczone przez kontroler.
- **Kontroler:** Klasa EventsController zarządza logiką aplikacji, obsługując żądania HTTP, manipulując modelami i przekazując dane do widoków.

Obserwator

- Wzorzec, w którym obiekty reagują na zmiany w innych obiektach.
- Operacje zwiększanie licznika polubień, dodawanie komentarza i deklaracji uczestnictwa w wydarzeniu powodują aktualizację danych i ich przeniesienie do bazy.

Strategia

- Wzorzec pozwalający na dynamiczne wybieranie algorytmu w czasie wykonywania.
- Weryfikacja typu pliku obrazu (Image) podczas dodawania lub edycji wydarzenia opiera się na dynamicznej strategii sprawdzania rozszerzenia pliku.

Metoda szablonowa

- Wzorzec, który definiuje ogólny szkielet algorytmu, pozostawiając szczegóły do zaimplementowania w klasach podrzędnych.
- Metoda szablonowa w Entity Framework definiuje ogólny proces operacji na danych (CRUD).

Opis komponentów projektu:

Kontroler EventsController.cs

Kontroler EventsController zarządza logiką aplikacji związaną z wydarzeniami, obsługując żądania związane z ich wyświetlaniem, edytowaniem, usuwaniem oraz interakcjami użytkowników, takimi jak polubienia i komentarze.

- Wyświetlanie wydarzeń:
 - ❖ EventsViewer – wyświetla wszystkie dostępne wydarzenia
 - ❖ MyEventsViewer – Wyświetla listę wydarzeń utworzonych przez zalogowanego użytkownika.
- Szczegóły wydarzenia:
 - ❖ Edit (GET): Wyświetla formularz do edycji istniejącego wydarzenia.
 - ❖ Edit (POST): Obsługuje aktualizację danych wydarzenia, w tym zmianę obrazu.
- Usuwanie wydarzeń:
 - ❖ DeleteConfirmed: Usuwa wydarzenie na podstawie jego identyfikatora.
- Interakcje użytkowników:
 - ❖ Like: Umożliwia polubienie wydarzenia przez użytkownika, z uwzględnieniem, czy już je polubił.
 - ❖ AddComment: Dodaje komentarz do wydarzenia, z walidacją treści i użytkownika.
 - ❖ Register: Zapisuje użytkownika do wydarzenia.

Model Event.cs

Model Event reprezentuje pojedyncze wydarzenie w aplikacji. Jest on używany do zarządzania i przechowywania danych o wydarzeniach.

Lista właściwości modelu:

- Id (Int)
- UserId (String?)
- User (IdentityUser?)
- Title (String) - wymagany
- Date (DateTime) - wymagany
- Description (String) – wymagany, limit 1000 znaków
- Location (String) - wymagany
- LikeCount (Int)
- ShareCount(Int)
- Comments (ICollection<Comments>)
- ImagePath (String)
- UserEmail (String)
- EventRegistrations (ICollection<EventRegistration>)

Model Comments.cs

Model Comments reprezentuje strukturę komentarza w systemie. Służy do przechowywania informacji o treści komentarza, autorze oraz czasie jego dodania, a także powiązania z konkretnym wydarzeniem.

- Id (int)
- EventId (int) – Wymagany.
- UserName (string) – Wymagany, limit 100 znaków.
- Content (string) – Wymagany, limit 500 znaków.
- CreatedAt (DateTime)

Model UserLike.cs

Model UserLikes reprezentuje strukturę przechowującą informację o użytkownikach, którzy polubili dane wydarzenie. Służy do rejestrowania interakcji użytkowników z wydarzeniami poprzez polubienia.

- Id (int)
- EventId (int) – Wymagany.
- UserId (string) – Wymagany.

Model EventRegistration.cs

Model EventRegistration obsługuje zapisy na wydarzenia.

- Id (int)
- EventId (int)
- UserId (string)
- RegisteredAt (DateTime)
- Event (Event)
- User (IdentityUser)

Widoki Events

Widoki te odpowiadają za interakcję wydarzeń użytkowników, z możliwością tworzenia, usuwania, edycji, sortowania, przeglądania szczegółów oraz interakcji.

- Create
- Details
- Edit
- EventsViewer
- MyEventsViewer

Widoki Shared

Widoki współdzielone odpowiadają za wyświetlanie części wspólnych strony (strona główna, pasek zakładek, interfejs użytkownika).

- _Layout
- _LoginPartial

Opis systemu użytkowników:

Użytkownik niezalogowany

- Brak dostępu do systemu: zablokowany podgląd, dodawanie i edycja wydarzeń.
- Dostęp do strony głównej po rejestracji i zalogowaniu.

Użytkownik zalogowany

- Pełny dostęp do funkcjonalności systemu.
- Wymagane uwierzytelnienie po rejestracji.
- Działania takie jak edycja, usuwanie czy dodawanie wydarzeń są ograniczone do użytkownika który je utworzył.

Wykorzystane technologie:

- ASP.NET 8.0
- Entity Framework
- SQL Server