

Dokumentacja techniczna do projektu

**„Aplikacja do rezerwacji  
biletów na wydarzenia”**

Autorzy:

Bartosz Szuba

Nikodem Tyranowski

## Spis treści

|    |  |    |
|----|--|----|
| 1. | Krótki opis.....                                   | 3  |
| 2. | Specyfikacja użytych technologii.....              | 3  |
| 3. | Instrukcja pierwszego uruchomienia projektu .....  | 4  |
|    | Pobranie projektu .....                            | 4  |
|    | Przygotowanie środowiska lokalnego .....           | 4  |
|    | Uruchomienie aplikacji.....                        | 4  |
|    | Zarządzanie bazą danych.....                       | 4  |
| 4. | Opis struktury projektu.....                       | 4  |
|    | Modele (Models).....                               | 4  |
|    | Widoki (Views).....                                | 5  |
|    | Kontrolery (Controllers).....                      | 5  |
|    | Baza danych (Data) .....                           | 5  |
|    | Warstwa autoryzacji i uwierzytelniania.....        | 6  |
|    | Zasoby statyczne (wwwroot).....                    | 6  |
|    | Pliki konfiguracji.....                            | 6  |
| 5. | Opis modeli .....                                  | 7  |
|    | <b>Model: Event</b> .....                          | 7  |
|    | <b>Model: EventCreate</b> .....                    | 8  |
|    | <b>Model: Reservation</b> .....                    | 9  |
| 6. | Opis kontrolerów .....                             | 10 |
|    | <b>HomeController</b> .....                        | 10 |
|    | <b>EventsController</b> .....                      | 10 |
| 7. | Opis systemu użytkowników .....                    | 13 |
|    | <b>Role w systemie</b> .....                       | 13 |
|    | Mechanizmy autoryzacji.....                        | 14 |
|    | Powiązanie danych użytkownika z rezerwacjami ..... | 14 |
|    | Zarządzanie użytkownikami i rolami .....           | 14 |
| 8. | Najciekawsze funkcjonalności .....                 | 15 |
|    | Interaktywny system rezerwacji miejsc.....         | 15 |
|    | Filtry i wyszukiwanie wydarzeń.....                | 15 |
|    | System rezerwacji z obsługą miejsc VIP.....        | 15 |
|    | Zakładka "Moje rezerwacje" .....                   | 15 |
|    | Zarządzanie obrazami wydarzeń .....                | 15 |
|    | Obsługa kategorii wydarzeń .....                   | 15 |

# 1. Krótki opis

Aplikacja umożliwia użytkownikom przeglądanie wydarzeń (np. koncertów, spektakli) według kategorii, dokonywanie rezerwacji biletów, zarządzanie miejscami oraz oferuje panel administracyjny dla organizatorów do zarządzania wydarzeniami. System obsługuje bazę danych, która przechowuje informacje o użytkownikach, wydarzeniach i dokonanych rezerwacjach.

## Główne funkcjonalności:

1. **Przeglądanie wydarzeń:** Filtracja wydarzeń według kategorii.
2. **Rezerwacja biletów:** Możliwość wyboru wydarzenia i rezerwacji biletów.
3. **Zarządzanie miejscami:** Wybór konkretnych miejsc na wydarzeniu.
4. **Panel administracyjny dla organizatorów:** Dodawanie i edytowanie wydarzeń.
5. **Baza danych:** Tabele przechowujące informacje o użytkownikach, wydarzeniach i rezerwacjach.

# 2. Specyfikacja użytych technologii

- **Framework backendowy:** ASP.NET 8 (Model-View-Controller)
- **Baza danych:** PostgreSQL (zarządzana przez pgAdmin)
- **ORM:** Entity Framework Core
- **Język programowania:** C#
- **Narzędzie do zarządzania bazą danych:** pgAdmin 4
- **Hosting:** Serwer PostgreSQL na apaturehost.pl
- **Frontend:** Razor Pages
- **Obsługa autentykacji:** ASP.NET Identity

### 3. Instrukcja pierwszego uruchomienia projektu

#### Pobranie projektu

Przejdź na stronę <https://github.com/NTyranowski/aplikacja-zaliczeniowa>, pobierz repozytorium na swój komputer i je rozpakuj.

#### Przygotowanie środowiska lokalnego

Upewnij się, że masz na komputerze zainstalowany program Visual Studio.

#### Uruchomienie aplikacji

Przejdź do folderu \aplikacja-zaliczeniowa-main\Strona do rezerwacji biletów i uruchom plik [Strona do rezerwacji biletów.sln](#)

#### Zarządzanie bazą danych

Baza danych znajduje się na stronie [apaturehost.pl](https://pgadmin4.apaturehost.pl/).

Link: <https://pgadmin4.apaturehost.pl/pgadmin4/browser/>

### 4. Opis struktury projektu

Projekt aplikacji internetowej oparty jest na architekturze Model-View-Controller (MVC), co zapewnia czytelny podział odpowiedzialności pomiędzy warstwy aplikacji. Struktura projektu wygląda następująco:

#### Modele (Models)

- Modele reprezentują dane oraz logikę aplikacji.
- Klasy modelu takie jak **Event** i **Reservation** odpowiadają za strukturę danych, przechowywanie informacji o wydarzeniach oraz rezerwacjach.

- Modele definiują także relacje między danymi, takie jak powiązanie rezerwacji z wydarzeniami oraz użytkownikami.

## Widoki (Views)

- Widoki zawierają interfejs użytkownika, zdefiniowany w plikach Razor (**.cshtml**).
- Dla każdego kontrolera dostępne są odpowiednie widoki, np. widok rezerwacji miejsc (**Reserve.cshtml**), widok szczegółów wydarzenia (**Details.cshtml**) czy widok zarządzania wydarzeniami.
- Widoki wspierają dynamiczne renderowanie danych przesyłanych z kontrolerów i umożliwiają wizualizację miejsc w formie interaktywnej szachownicy.

## Kontrolery (Controllers)

- Kontrolery pośredniczą między widokami a modelami, zarządzając logiką biznesową i przetwarzaniem danych:
  - **EventsController**: Obsługuje operacje związane z przeglądaniem wydarzeń, rezerwacją miejsc, filtrowaniem wydarzeń oraz zarządzaniem wydarzeniami (funkcje dla administratorów).
  - **HomeController**: Zarządza stroną główną i obsługuje standardowe widoki, takie jak polityka prywatności.

## Baza danych (Data)

- Aplikacja korzysta z bazy danych PostgreSQL, zarządzanej za pomocą **pgAdmin**.
- Dane przechowywane w bazie to m.in. informacje o wydarzeniach, rezerwacjach, użytkownikach i miejscach.
- **ApplicationDbContext** implementuje wzorzec Repository, integrując się z Entity Framework Core, co ułatwia wykonywanie operacji CRUD oraz zarządzanie migracjami.

## Warstwa autoryzacji i uwierzytelniania

- Użytkownicy muszą się zarejestrować i zalogować, aby rezerwować miejsca i przeglądać swoje rezerwacje.
- Administratorzy mają dostęp do dedykowanych funkcji, takich jak dodawanie nowych wydarzeń, edytowanie ich i usuwanie, dzięki autoryzacji opartej na rolach.

## Zasoby statyczne (wwwroot)

- Przechowywane są tutaj pliki statyczne, takie jak obrazy wydarzeń, arkusze stylów CSS czy skrypty JavaScript.
- Obrazy wydarzeń są automatycznie zapisywane w dedykowanym katalogu, co pozwala na łatwą organizację.

## Pliki konfiguracji

- Plik **appsettings.json** zawiera konfigurację połączenia z bazą danych PostgreSQL oraz inne ustawienia aplikacji.
- Mechanizmy logowania i autoryzacji są konfigurowane w pliku **Program.cs**.

## 5. Opis modeli

**Model: Event** - Przechowuje informacje o wydarzeniu.

| Pole                        | Typ danych               | Walidacja                      | Opis  |
|-----------------------------|--------------------------|--------------------------------|---|
| <b>Id</b>                   | int                      | Klucz główny                   | Identyfikator wydarzenia.                       |
| <b>Title</b>                | string                   | [Required],<br>MaxLength (100) | Nazwa wydarzenia.                               |
| <b>Date</b>                 | DateTime                 | [Required]                     | Data wydarzenia.                                |
| <b>Description</b>          | string                   | [Required],<br>MaxLength (100) | Opis wydarzenia                                 |
| <b>Category</b>             | string                   | [Required],<br>MaxLength (100) | Kategoria wydarzenia (np. "Sport", "Muzyka")    |
| <b>AvailableNormalSeats</b> | int                      | [Range (1, 132)]               | Liczba dostępnych normalnych miejsc.            |
| <b>AvailableVIPSeats</b>    | int                      | [Range (1, 12)]                | Liczba dostępnych miejsc VIP.                   |
| <b>ImagePath</b>            | string                   | [Required],<br>MaxLength (100) | Lokalizacja pliku z grafiką.                    |
| <b>CreatorId</b>            | string                   | [Required]                     | Identyfikator użytkownika tworzącego wydarzenie |
| <b>Reservations</b>         | ICollection<Reservation> | Brak walidacji, jest relacją   | Kolekcja rezerwacji powiązanych z wydarzeniem   |

### Model: EventCreate – Tworzy nowe wydarzenie.

| Pole                        | Typ danych | Walidacja                      | Opis  |
|-----------------------------|------------|--------------------------------|---|
| <b>Id</b>                   | int        | Klucz główny                   | Identyfikator wydarzenia.                               |
| <b>Title</b>                | string     | [Required],<br>MaxLength (100) | Nazwa wydarzenia.                                       |
| <b>Date</b>                 | DateTime   | [Required]                     | Data wydarzenia.  |
| <b>Description</b>          | string     | [Required],<br>MaxLength (100) | Opis wydarzenia   |
| <b>Category</b>             | string     | [Required],<br>MaxLength (100) | Kategoria wydarzenia (np. "Sport", "Muzyka")            |
| <b>AvailableNormalSeats</b> | int        | [Range (1, 132)]               | Liczba dostępnych normalnych miejsc.                    |
| <b>AvailableVIPSeats</b>    | int        | [Range (1, 12)]                | Liczba dostępnych miejsc VIP.                           |
| <b>Image</b>                | IFormFile  | [Required],<br>MaxLength (100) | Obrazek reprezentujący wydarzenie (plik do załadowania) |



**Model: Reservation** - Reprezentuje rezerwację biletów na wydarzenie.

| Pole          | Typ danych | Walidacja                       | Opis  |
|---------------|------------|---------------------------------|---|
| Id            | int        | Klucz główny                    | Identyfikator rezerwacji.   |
| UserId        | string     | [Required],<br>MaxLength (100)  | Identyfikator użytkownika.  |
| EventId       | int        | [Required]                      | Identyfikator wydarzenia  |
| SeatsReserved | int        | [Required],<br>Min(1)           | Ilość zarezerwowanych miejsc.   |
| Event         | virtual    | Brak walidacji,<br>jest relacją | Relacja do obiektu Event, reprezentująca wydarzenie, z którym związana jest rezerwacja. |
| SeatIds       | string     | [Range (1, 144)]                | Identyfikatory zarezerwowanych biletów.   |

## 6. Opis kontrolerów

**HomeController** - Odpowiada za ogólne strony aplikacji:

- Przekierowanie na stronę główną (lista wydarzeń).
- Wyświetlanie strony prywatności.
- Obsługa błędów.

| Metoda         | HTTP | Parametry | Opis  | Zwracane dane                     |
|----------------|------|-----------|---|-----------------------------------|
| <b>Index</b>   | GET  | Brak      | Przekierowuje użytkownika do widoku listy wydarzeń.                                       | Przekierowanie do listy wydarzeń. |
| <b>Privacy</b> | GET  | Brak      | Wyświetla widok strony prywatności.   | Widok strony prywatności.         |
| <b>Error</b>   | GET  | Brak      | Wyświetla widok błędu, zwraca szczegóły błędu na podstawie ID żądania lub kontekstu HTTP. | Widok błędu z danymi debugowania. |

**EventsController** - Odpowiada za zarządzanie wydarzeniami i rezerwacjami:

- Wyświetlanie listy wydarzeń (z wyszukiwaniem i filtrowaniem).
- Szczegóły wydarzeń.
- Obsługa rezerwacji miejsc (dodawanie, anulowanie).
- Wyświetlanie rezerwacji użytkownika.
- Dodawanie nowych wydarzeń (dostępne tylko dla administratora).

| Metoda                   | HTTP | Parametry                                  | Opis   | Zwracane dane   |
|--------------------------|------|--|--|---|
| <b>Index</b>             | GET  | category (string),<br>searchQuery (string) | Wyświetla listę wydarzeń z możliwością filtrowania po kategorii i wyszukiwania.                          | Widok listy wydarzeń z kategoriami i filtrem.               |
| <b>Details</b>           | GET  | id (int)                                   | Wyświetla szczegóły wybranego wydarzenia.  | Widok szczegółów wydarzenia.                                |
| <b>Reserve</b>           | GET  | id (int)                                   | Wyświetla formularz rezerwacji miejsc na wybrane wydarzenie oraz listę zarezerwowanych miejsc.           | Widok rezerwacji miejsc na wydarzenie.                      |
| <b>Reserve</b>           | POST | eventId (int),<br>selectedSeats (string)   | Przetwarza rezerwację miejsc, aktualizuje liczbę dostępnych miejsc i zapisuje rezerwację w bazie danych. | Widok szczegółów wydarzenia zaktualizowanego po rezerwacji. |
| <b>MyReservations</b>    | GET  | Brak                                       | Wyświetla listę rezerwacji użytkownika oraz liczbę wszystkich zarezerwowanych miejsc.                    | Widok listy rezerwacji użytkownika.                         |
| <b>CancelReservation</b> | POST | id (int)                                   | Usuwa rezerwację z bazy danych, przywraca dostępność miejsc w wydarzeniu.                                | Przekierowanie do widoku rezerwacji użytkownika.            |
| <b>Add</b>               | GET  | Brak                                       | Wyświetla formularz dodawania nowego wydarzenia (tylko dla roli "Admin").                                | Widok formularza dodawania wydarzenia.                      |

|                        |      |                        |  |   |
|------------------------|------|------------------------|--|---|
| <b>Add</b>             | POST | model<br>(EventCreate) | Przetwarza formularz dodawania wydarzenia, zapisuje dane w bazie i przesyła plik graficzny do folderu "images/events". | Przekierowanie do listy wydarzeń.                                 |
| <b>Delete</b>          | GET  | id (int)               | Wyświetla formularz potwierdzenia usunięcia wydarzenia o podanym id.   | Formularz potwierdzenia usunięcia wydarzenia                      |
| <b>DeleteConfirmed</b> | POST | id (int)               | Usuwa wydarzenie oraz powiązane rezerwacje z bazy danych.  | Przekierowanie do listy wydarzeń                                  |
| <b>Edit</b>            | GET  | id (int)               | Wyświetla formularz edycji wydarzenia o podanym id.  | Formularz edycji wydarzenia                                       |
| <b>Edit</b>            | POST | model<br>(EventCreate) | Aktualizuje dane wydarzenia w bazie danych, zapisując zmiany, w tym ewentualny nowy plik obrazka.                      | Przekierowanie do listy wydarzeń lub widoku szczegółów wydarzenia |

## 7. Opis systemu użytkowników

### Role w systemie

#### 1. Gość (Niezalogowany użytkownik):

- Może przeglądać listę wydarzeń.
- Może filtrować i wyszukiwać wydarzenia według kategorii i tytułów.
- Może przeglądać szczegóły poszczególnych wydarzeń.
- Brak możliwości rezerwacji biletów i dostępu do panelu administracyjnego.

#### 2. Zalogowany użytkownik:

- Ma wszystkie uprawnienia gościa.
- Może rezerwować miejsca na wydarzenia.
- Ma dostęp do zakładki "Moje rezerwacje", gdzie widzi listę swoich rezerwacji, liczbę zajętych miejsc, numery miejsc oraz szczegóły wydarzeń.
- Może anulować swoje rezerwacje.
- Może wybrać miejsca VIP (jeśli dostępne) i standardowe w ramach rezerwacji.

#### 3. Administrator:

- Ma wszystkie uprawnienia zalogowanego użytkownika.
- Dodatkowo może:
  - Dodawać nowe wydarzenia z pełną specyfikacją (tytuł, opis, liczba miejsc, kategoria, data, obraz).
  - Edytować istniejące wydarzenia (opis, liczba dostępnych miejsc itd.).
  - Zarządzać wydarzeniami, co obejmuje również usuwanie ich z systemu.

## Mechanizmy autoryzacji:

- **[Authorize]**
  - Domyślnie wymaga zalogowania do uzyskania dostępu do metod kontrolerów.
  - Dla wybranych akcji (np. wyświetlanie wydarzeń) ustawiono atrybut **[AllowAnonymous]**, aby umożliwić dostęp gościom.
- **[Authorize (Roles = "Admin")]**
  - Wykorzystywany do ochrony metod dostępnych wyłącznie dla administratorów, takich jak dodawanie, edytowanie i usuwanie wydarzeń.

## Powiązanie danych użytkownika z rezerwacjami:

- Każda rezerwacja jest powiązana z identyfikatorem użytkownika (**UserId**), uzyskiwanym z systemu autoryzacji (**ClaimTypes.NameIdentifier**).
- Rezerwacje zalogowanego użytkownika są filtrowane w bazie danych, aby umożliwić wyświetlanie wyłącznie jego rezerwacji w zakładce "Moje rezerwacje".

## Zarządzanie użytkownikami i rolami:

- Role użytkowników są przechowywane w systemie autoryzacji.
- Aby nadać użytkownikowi rolę administratora należy podczas tworzenia konta zaznaczyć odpowiednie pole i otrzymać zatwierdzenie tej operacji przez głównego Administratora.

## 8. Najciekawsze funkcjonalności

### Interaktywny system rezerwacji miejsc

- Użytkownik widzi wizualną reprezentację miejsc w formie szachownicy 12x10.
- Kliknięcie na miejsce zmienia jego kolor na żółty (wskazuje wybór), a po zatwierdzeniu rezerwacji miejsce staje się niedostępne dla innych użytkowników.
- System rozróżnia miejsca VIP i standardowe, z automatycznym uwzględnieniem ich dostępności podczas rezerwacji.

### Filtry i wyszukiwanie wydarzeń

- Użytkownicy mogą filtrować wydarzenia według kategorii oraz wyszukiwać je na podstawie tytułu lub opisu.
- Wyniki są dynamicznie sortowane według dat, co ułatwia znalezienie nadchodzących wydarzeń.

### System rezerwacji z obsługą miejsc VIP

- Rezerwacje uwzględniają różne typy miejsc (VIP i standardowe).
- Aplikacja automatycznie aktualizuje liczbę dostępnych miejsc VIP i standardowych w wydarzeniach na podstawie wybranych miejsc.

### Zakładka "Moje rezerwacje"

- Użytkownicy mają dostęp do swojej historii rezerwacji z informacjami o liczbie zarezerwowanych miejsc i szczegółach wydarzeń.
- Możliwość anulowania rezerwacji z automatycznym zwrotem miejsc do puli dostępnych.

### Zarządzanie obrazami wydarzeń

- Administratorzy mogą przysyłać obrazy dla wydarzeń, które są automatycznie zapisywane w katalogu aplikacji.
- Obrazy są unikalnie nazwane za pomocą GUID, co zapobiega kolizjom plików.

### Obsługa kategorii wydarzeń

- System automatycznie generuje listę unikalnych kategorii na podstawie dostępnych wydarzeń w bazie danych.
- Użytkownicy mogą szybko przeglądać wydarzenia z interesujących ich kategorii.