

# Dokumentacja Techniczna Aplikacji "GuestList"

## Tytuł Projektu

GuestList

## Krótki Opis Działania Projektu

Aplikacja "GuestList" to internetowa aplikacja służąca do zarządzania listą gości przez zarejestrowanych użytkowników. Użytkownicy mogą tworzyć, edytować i usuwać gości.

## Autor Projektu

Sebastian Walczak, nr. Indexu: 119619

## Specyfikacja Wykorzystanych Technologii

- .NET Core 6.0
- ASP.NET Core MVC
- Entity Framework Core
- SQL Server
- Visual Studio 2022

## Instrukcje Pierwszego Uruchomienia Projektu

Dane przechowywane są w lokalnej bazie danych SQL, aby połączyć się z lokalną bazą należy w pierwszej kolejności umieścić nazwę lokalnej bazy danych w pliku appsettings.json:

```
11. "GuestList": "Server=NAZWA_SERWERA;
```

Następnie uruchomić *Konsola menedżera pakietów* i wpisać następującą komendę: „*Update-Database*”, co spowoduje dodanie tablic z zawartymi w nich polami do bazy danych.

## Opis Struktury Projektu

### 1. Controllers

Katalog zawierający kontrolery odpowiedzialne za obsługę żądań HTTP i zarządzanie logiką aplikacji.

- HomeController.cs: Kontroler odpowiedzialny za obsługę rejestracji, logowania i wylogowania użytkowników.
- GuestController.cs: Kontroler odpowiedzialny za obsługę głównych funkcji aplikacji, takich jak wyświetlanie, dodawanie, edytowanie i usuwanie gości.

## 2. Data

Katalog zawierający kontekst bazy danych.

- `ApplicationDbContext.cs`: Klasa reprezentująca kontekst bazy danych, używana do zarządzania połączeniami z bazą danych oraz wykonywania operacji CRUD na tabelach.

## 3. Migrations

Katalog zawierający pliki z migracją do bazy danych.

## 4. Models/Entities

Katalog zawierający modele danych.

- `Guest.cs`: Model reprezentujący gościa w systemie.
- `User.cs`: Model reprezentujący użytkownika systemu.

## 5. Views

Katalog zawierający widoki (pliki Razor), które są renderowane jako strony HTML.

- a) Home
  - `Login.cshtml`: Widok formularza logowania.
  - `Register.cshtml`: Widok formularza rejestracji.
- b) Guests
  - `List.cshtml`: Widok wyświetlający listę gości.
  - `Add.cshtml`: Widok formularza dodawania nowego gościa.
  - `Edit.cshtml`: Widok formularza edycji istniejącego gościa.
- c) Shared
  - `_Layout.cshtml`: Zawiera podstawowe właściwości strony, wyświetlane w pozostałych widokach.
  - `Error.cshtml`: Widok zawierający komunikaty błędów.

## 6. Konfiguracja Aplikacji

- `appsettings.json`: Plik konfiguracyjny aplikacji zawierający ustawienia takie jak parametry połączenia z bazą danych.
- `Program.cs`: Główny punkt wejścia aplikacji, konfiguruje i uruchamia aplikację.
- `Startup.cs`: Klasa konfigująca usługi i middleware używane przez aplikację.

## Opis Modeli

### User

Model reprezentujący użytkownika systemu.

- Id: int (klucz główny, automatycznie generowany)
- Login: string (login użytkownika)
- Password: string (hasło użytkownika, długość między 5 a 13 znaków)

### Guest

Model reprezentujący gościa przypisanego do użytkownika.

- Id: int (klucz główny, automatycznie generowany)
- UserId: int (klucz obcy, odnosi się do użytkownika tworzącego gościa)
- Name: string (imię i nazwisko gościa)
- PhoneNumber: string (numer telefonu gościa)
- Confirmed: bool (czy gość potwierdził wizytę)

## Opis Kontrolerów

### AccountController

#### 1. Login

##### a) GET

Opis: Wyświetla formularz logowania.

##### b) POST

Parametry: `User` (obiekt zawierający dane logowania)

Opis: Przetwarza dane logowania, weryfikuje użytkownika i ustawia sesję.

Zwraca: Widok logowania z komunikatem o błędzie lub przekierowanie do widoku listy gości.

#### 2. Register

##### a) GET

Opis: Wyświetla formularz rejestracji.

##### b) POST

Parametry: `User` (obiekt zawierający dane rejestracyjne)

Opis: Przetwarza dane rejestracyjne, tworzy nowego użytkownika i przekierowuje do strony logowania.

Zwraca: Widok rejestracji z komunikatem o błędzie lub przekierowanie do widoku logowania z komunikatem "Konto zostało utworzone".

#### 3. Logout

##### a) POST

Opis: Wylogowuje użytkownika, usuwając jego sesję.

Zwraca: Przekierowanie do widoku logowania.

## HomeController

### 1. List

#### a) GET

Opis: Wyświetla listę gości zalogowanego użytkownika.

### 2. Add

#### a) GET

Opis: Wyświetla formularz dodawania nowego gościa.

#### b) POST

Parametry: `Guest` (obiekt zawierający dane nowego gościa)

Opis: Przetwarza dane nowego gościa i zapisuje je w bazie danych.

Zwraca: Przekierowanie do widoku listy gości.

### 3. Edit

#### a) GET

Parametry: `id` (ID gościa do edycji)

Opis: Wyświetla formularz edycji gościa.

#### b) POST

Parametry: `Guest` (obiekt zawierający zmienione dane gościa)

Opis: Przetwarza zmiany i aktualizuje dane gościa w bazie danych.

Zwraca: Przekierowanie do widoku listy gości.

### 4. Delete

#### a) POST

Parametry: `id` (ID gościa do usunięcia)

Opis: Usuwa gościa z bazy danych po potwierdzeniu przez użytkownika.

Zwraca: Przekierowanie do widoku listy gości.

## Opis Systemu Użytkowników

Aplikacja nie korzysta z zaawansowanego systemu ról. Wszyscy zarejestrowani użytkownicy mają te same uprawnienia.

Możliwości Użytkowników Zalogowanych:

- Dodawanie, edytowanie i usuwanie gości.
- Możliwość sortowania listy gości według różnych kryteriów.

Informacje Powiązane z Użytkownikiem:

- Każdy użytkownik ma swoje własne konto, do którego przypisane są jego dane logowania.
- Lista gości jest powiązana bezpośrednio z użytkownikiem, który je stworzył.

## **Krótką Charakterystyką Najciekawszych Funkcjonalności**

- Sortowanie Listy Gości

Lista gości może być sortowana według imienia, numeru telefonu oraz statusu potwierdzenia, co ułatwia zarządzanie dużymi listami gości.

- Formularz Edycji Gościa

Formularz edycji gościa pozwala na łatwe zarządzanie danymi gościa, w tym zmianę imienia, numeru telefonu oraz statusu potwierdzenia wizyty.

- Potwierdzenie Usunięcia Gościa

Przed usunięciem gościa, użytkownik musi potwierdzić swoje działanie, co zapobiega przypadkowemu usunięciu danych.

- Obsługa Sesji Użytkownika

Po zalogowaniu, sesja użytkownika jest przechowywana, co umożliwia dostęp do listy gości oraz innych funkcji aplikacji. Użytkownik może się wylogować w dowolnym momencie, co usunie jego sesję i zabezpieczy dostęp do jego danych. Dodatkowo sesja zostanie automatycznie usunięta po zamknięciu przeglądarki.