

Dokumentacja techniczna projektu

Tytuł projektu: TurboReserve

Repozytorium: <https://github.com/pzx113092/TurboReserve>

Autorzy projektu:

- Jakub Olejnik
- Dawid Majewski

Spis treści

Krótki opis projektu	3
Specyfikacja wykorzystanych technologii.....	3
Instrukcje pierwszego uruchomienia projektu	3
Najciekawsze funkcjonalności systemu	3
Role i uprawnienia	4
Przypisywanie ról użytkownikom	4
Uprawnienia użytkowników	4
Informacje powiązane z użytkownikami	4
Użytkownicy testowi.....	5
Ustugodawcy	5
Klienci	5
Opis struktury projektu.....	6
Modele	6
Model: ErrorViewModel	6
Model: Reservation	7
Model: ReservationViewModel	8
Model: Service	8
Model: ServiceViewModel.....	10
Model: ServiceProvider.....	11
Model: ServiceProviderViewModel	12
Model: ServiceScheduleViewModel	14
Model: ScheduleSlotViewModel.....	15
Model: ScheduleSlot	15
Modele Microsoft Identity	17
Kontrolery	18
Kontroler: AboutController.....	18
Kontroler: CustomerController.....	19
Kontroler: HomeController	20
Kontroler: ReservationsController	21
Kontroler: ScheduleController	24
Kontroler: ServiceProviderPanelController	27
Kontroler: ServicesController	29

Krótki opis projektu

TurboReserve to platforma umożliwiająca rezerwację usług takich jak mechanik, elektryk, hydraulik, wulkanizacja czy ślusarz. Klienci mogą zakładać konta, przeglądać dostępnych usługodawców i ich lokalizację, rezerwować usługi oraz monitorować status rezerwacji. Usługodawcy mogą zarządzać swoimi ofertami, dostępnością, cennikiem oraz statusami rezerwacji.

Specyfikacja wykorzystanych technologii

- Backend: C# .NET 8
- Frontend: HTML, CSS, JavaScript
- Baza danych: SQLite
- API mapy: openstreetmap.org

Instrukcje pierwszego uruchomienia projektu

1. Sklonuj repozytorium w Visual Studio
2. Odśwież pakiety nuget w Package Manager Console:

```
dotnet nuget remove source "Microsoft Visual Studio Offline Packages"
```

```
dotnet nuget add source https://api.nuget.org/v3/index.json --name nuget.org
```
3. Skompiluj i uruchom aplikację w przeglądarce

Najciekawsze funkcjonalności systemu

1. Przypisanie lokalizacji z mapą dla usługodawcy

Podczas rejestracji usługodawca może wskazać swoją lokalizację, korzystając z interaktywnej mapy (Leaflet). Po wybraniu punktu na mapie współrzędne geograficzne (szerokość i długość) są automatycznie zapisane w systemie i mapa jest wyświetlana w profilu usługodawcy.

2. Import grafiku z pliku Excel

Usługodawcy mogą wczytać plik Excel zawierający harmonogram dostępnych terminów. System waliduje wprowadzone dane (np. brak przeszłych terminów, poprawność przedziałów czasowych) i zapisuje je w bazie danych.

Role i uprawnienia

Przypisywanie ról użytkownikom

Role są przypisywane użytkownikom podczas rejestracji – osobny proces rejestracyjny dla klienta i usługodawcy.

Uprawnienia użytkowników

1. Goście (niezalogowani użytkownicy):

- Mogą:
 - Przeglądać stronę główną i informacje o systemie (np. "O nas").
- Nie mogą:
 - Przeglądać listy usług.
 - Rezerwować terminów.
 - Zarządzać usługami ani rezerwacjami.

2. Zalogowani użytkownicy (Customer):

- Mają dostęp do funkcji takich jak:
 - Lista usług i usługodawców.
 - Rezerwacja terminów.
 - Historia rezerwacji.

3. Zalogowani użytkownicy (ServiceProvider):

- Mogą:
 - Zarządzać swoimi usługami.
 - Dodawać i edytować grafik terminów.
 - Zarządzać rezerwacjami klientów.

Informacje powiązane z użytkownikami

1. Customer:

- Identyfikator użytkownika (IdentityUserId).
- Lista rezerwacji.

2. ServiceProvider:

- Identyfikator użytkownika (IdentityUserId).
- Dane firmy (nazwa, adres, lokalizacja).
- Lista usług.

Użytkownicy testowi

Na potrzeby demonstracyjne utworzono konta usługodawców i klientów. Aby utworzyć kolejne konta należy przejść do zakładki Rejestracja w aplikacji.

Usługodawcy

Email: elektrycznyeryk@gmail.com

Hasło: ElektrycznyEryk1!

Email: supermechanik@gmail.com

Hasło: SuperMechanik1!

Email: hydraulikhubert@gmail.com

Hasło: HydraulikHubert1!

Email: stolarzsebastian@gmail.com

Hasło: StolarzSebastian1!

Klienci

Email: jannowak@gmail.com

Hasło: JanNowak1!

Email: dawidmajewski@gmail.com

Hasło: DawidMajewski1!

Email: jakubolejnik@gmail.com

Hasło: JakubOlejnik1!

Opis struktury projektu

Backend:

Foldery projektu:

- Controllers – Zawiera wszystkie kontrolery obsługujące logikę API.
- Models – Definicje modeli używanych w aplikacji.
- Services - Usługi odpowiedzialne za logikę biznesową.
- Data - Pliki konfiguracji bazy danych oraz migracje.
- Areas/Identity/Pages – Elementy Microsoft Identity.

Frontend:

Foldery projektu:

- Views – Główne strony aplikacji.
- wwwroot/lib – Zmodyfikowany bootstrap

Modele

Model: `ErrorViewModel`

Opis: Model służy do przekazywania informacji o błędach w aplikacji, takich jak identyfikator żądania (RequestId), co pozwala na łatwiejsze śledzenie problemów.

Pola:

1. RequestId (string?):

- **Opis:** Identyfikator żądania, który umożliwia odnalezienie konkretnego błędu w logach.
- **Ograniczenia:** Może być null.

2. ShowRequestId (bool):

- **Opis:** Wskazuje, czy RequestId jest ustawiony i powinien być wyświetlany.
- **Logika:** Wartość true, jeśli RequestId nie jest pusty ani null.

Relacje: Brak. Model jest niezależny.

Zastosowanie:

- Wyświetlanie szczegółów błędów w dedykowanej stronie błędów (Error.cshtml).

Model: Reservation

Opis:

Reprezentuje rezerwację usługi przez klienta, wiążąc użytkownika, usługę i usługodawcę z określonymi czasami rezerwacji.

Pola:

- 1. Id (int):**
 - **Opis:** Klucz główny rezerwacji.
- 2. CustomerId (string):**
 - **Opis:** Identyfikator klienta (powiązany z IdentityUser).
- 3. Customer (IdentityUser):**
 - **Opis:** Nawigacyjne pole do obiektu klienta.
- 4. ServiceId (int):**
 - **Opis:** Identyfikator wybranej usługi.
- 5. Service (Service):**
 - **Opis:** Nawigacyjne pole do szczegółów usługi.
- 6. ServiceProviderId (int):**
 - **Opis:** Identyfikator usługodawcy, który realizuje usługę.
- 7. ServiceProvider (ServiceProvider):**
 - **Opis:** Nawigacyjne pole do obiektu usługodawcy.
- 8. ReservationDate (DateTime):**
 - **Opis:** Data i czas złożenia rezerwacji.
- 9. StartTime (DateTime):**
 - **Opis:** Data i czas rozpoczęcia rezerwacji.
- 10. EndTime (DateTime):**
 - **Opis:** Data i czas zakończenia rezerwacji.
- 11. Status (ReservationStatus):**
 - **Opis:** Status rezerwacji (np. Pending, Confirmed, Completed, Canceled).

Relacje:

- **Reservation** ma powiązania:
 - Z klientem (Customer).
 - Z usługą (Service).
 - Z usługodawcą (ServiceProvider).

Zastosowanie:

- **Reservation:** Obsługa logiki biznesowej i bazy danych.

Model: ReservationViewModel

Opis:

Reprezentuje uproszczone dane rezerwacji do wyświetlania na stronie użytkownika.

Pola:

- 1. ReservationId (int):**
 - **Opis:** Klucz główny rezerwacji.
- 2. ServiceName (string):**
 - **Opis:** Nazwa usługi.
 - **Walidacja:** Wymagane pole.
- 3. ServiceProviderName (string):**
 - **Opis:** Nazwa usługodawcy.
 - **Walidacja:** Wymagane pole.
- 4. ReservationDate (DateTime):**
 - **Opis:** Data i czas złożenia rezerwacji.
 - **Walidacja:** Wymagane pole.
- 5. StartTime (DateTime):**
 - **Opis:** Data i czas rozpoczęcia rezerwacji.
 - **Walidacja:** Wymagane pole.
- 6. EndTime (DateTime):**
 - **Opis:** Data i czas zakończenia rezerwacji.
 - **Walidacja:** Wymagane pole.
- 7. Status (string):**
 - **Opis:** Status rezerwacji jako ciąg znaków.
 - **Walidacja:** Wymagane pole.
- 8. CustomerName (string):**
 - **Opis:** Imię i nazwisko klienta.
 - **Walidacja:** Wymagane pole.

Relacje:

ReservationViewModel jest używane do wyświetlania danych w uproszczonej formie, bez pełnych relacji.

Zastosowanie:

ReservationViewModel: Wyświetlanie danych w interfejsie użytkownika.

Model: Service

Opis:

Reprezentuje usługę oferowaną przez usługodawcę. Zawiera szczegóły, takie jak nazwa, opis, cena oraz powiązania z usługodawcą, rezerwacjami i grafikami.

Pola:

1. Id (int):

- **Opis:** Klucz główny identyfikujący usługę w bazie danych.

2. Name (string):

- **Opis:** Nazwa usługi.
- **Walidacja:**
 - Pole wymagane.
 - Maksymalna długość: 100 znaków.
 - **Komunikat błędu:** "Nazwa jest wymagana." / "Nazwa nie może przekraczać 100 znaków."

3. Description (string):

- **Opis:** Szczegółowy opis usługi.
- **Walidacja:**
 - Pole wymagane.
 - Maksymalna długość: 500 znaków.
 - **Komunikat błędu:** "Opis jest wymagany." / "Opis nie może przekraczać 500 znaków."

4. Price (string):

- **Opis:** Cena usługi w formacie tekstowym (np. "50 zł/h").
- **Walidacja:**
 - Pole wymagane.
 - Maksymalna długość: 50 znaków.
 - **Komunikat błędu:** "Cena jest wymagana." / "Cena nie może przekraczać 50 znaków."

5. ServiceProviderId (int):

- **Opis:** Klucz obcy odnoszący się do usługodawcy, który oferuje usługę.

6. ServiceProvider (ServiceProvider):

- **Opis:** Nawigacyjne pole umożliwiające dostęp do danych usługodawcy powiązanego z usługą.

7. ScheduleSlots (List<ScheduleSlot>):

- **Opis:** Lista grafików (dostępnych terminów) związanych z usługą.

8. Reservations (List<Reservation>):

- **Opis:** Lista rezerwacji związanych z usługą.

Relacje:

- **Jeden-do-wielu z usługodawcą:** Jedna usługa jest powiązana z jednym usługodawcą (ServiceProvider).
- **Jeden-do-wielu z rezerwacjami:** Jedna usługa może mieć wiele rezerwacji (Reservations).
- **Jeden-do-wielu z grafikami:** Jedna usługa może mieć wiele dostępnych terminów (ScheduleSlots).

Zastosowanie:

- Przechowywanie szczegółowych informacji o usługach oferowanych przez usługodawcę.
- Powiązanie usług z rezerwacjami i terminami dostępnymi w grafiku.

Model: ServiceViewModel

Opis:

Reprezentuje uproszczony model usługi używany do wyświetlania lub edycji danych usługi w interfejsie użytkownika.

Pola:

1. ServiceId (int):

- **Opis:** Identyfikator usługi.
- **Zastosowanie:** Używane do identyfikacji usługi w bazie danych lub w operacjach edycji i usuwania.

2. Name (string):

- **Opis:** Nazwa usługi.
- **Walidacja:**
 - Pole wymagane.
 - **Komunikat błędu:** Automatyczny komunikat ASP.NET Core dla pól wymaganych.

3. Description (string):

- **Opis:** Szczegółowy opis usługi.
- **Walidacja:**
 - Pole wymagane.
 - **Komunikat błędu:** Automatyczny komunikat ASP.NET Core dla pól wymaganych.

4. Price (string):

- **Opis:** Cena usługi w formacie tekstowym (np. "50 zł/h").
- **Walidacja:**
 - Pole wymagane.

- **Komunikat błędu:** Automatyczny komunikat ASP.NET Core dla pól wymaganych.

Relacje:

- Model nie ma bezpośrednich relacji z innymi modelami. Jest używany do uproszczenia wyświetlania i obsługi danych usługi w widokach.

Zastosowanie:

- **Wyświetlanie danych:** Prezentowanie informacji o usłudze w widokach (np. lista usług, szczegóły usługi).
- **Edycja danych:** Obsługa formularzy do aktualizacji danych usługi w widokach.
- **Uproszczenie widoków:** Zmniejszenie liczby pól w porównaniu do pełnego modelu Service.

Model: ServiceProvider

Opis:

Reprezentuje usługodawcę w systemie, przechowując informacje kontaktowe, lokalizację oraz powiązane usługi.

Pola:

1. **Id (int):**
 - **Opis:** Klucz główny identyfikujący usługodawcę w bazie danych.
2. **IdentityUserId (string):**
 - **Opis:** Identyfikator użytkownika (IdentityUser) powiązanego z usługodawcą.
 - **Walidacja:**
 - Pole wymagane.
3. **BusinessName (string):**
 - **Opis:** Nazwa firmy usługodawcy.
 - **Walidacja:**
 - Pole wymagane.
4. **Address (string):**
 - **Opis:** Adres usługodawcy (np. ulica i numer domu).
 - **Walidacja:**
 - Pole wymagane.
5. **City (string):**
 - **Opis:** Miasto, w którym działa usługodawca.
 - **Walidacja:**
 - Pole wymagane.
6. **ZipCode (string):**
 - **Opis:** Kod pocztowy usługodawcy.

- **Walidacja:**
 - Pole wymagane.
- 7. Latitude (double):**
 - **Opis:** Szerokość geograficzna lokalizacji usługodawcy.
- 8. Longitude (double):**
 - **Opis:** Długość geograficzna lokalizacji usługodawcy.
- 9. User (IdentityUser):**
 - **Opis:** Nawigacyjne pole do obiektu użytkownika powiązanego z usługodawcą.
- 10. Services (ICollection<Service>):**
 - **Opis:** Lista usług oferowanych przez usługodawcę.
 - **Inicjalizacja:** Domyślnie inicjalizowana jako pusta lista.

Relacje:

- **Jeden-do-jeden z IdentityUser:** Każdy usługodawca jest powiązany z jednym kontem użytkownika (User).
- **Jeden-do-wielu z Service:** Usługodawca może oferować wiele usług.

Zastosowanie:

- **Przechowywanie danych kontaktowych:** Informacje o firmie i lokalizacji usługodawcy.
- **Powiązanie z kontem użytkownika:** Łączenie usługodawcy z tożsamością w systemie logowania.
- **Obsługa usług:** Zarządzanie listą usług oferowanych przez usługodawcę.
- **Lokalizacja:** Wykorzystywane do wyświetlania lokalizacji usługodawców na mapie

Model: ServiceProviderViewModel

Opis:

Reprezentuje uproszczony model usługodawcy używany do wyświetlania danych w interfejsie użytkownika. Zawiera podstawowe informacje o usługodawcy oraz listę jego usług.

Pola:

- 1. ServiceProviderId (int):**
 - **Opis:** Identyfikator usługodawcy.
 - **Zastosowanie:** Służy do identyfikacji usługodawcy w operacjach wyświetlania lub edycji.
- 2. BusinessName (string):**
 - **Opis:** Nazwa firmy usługodawcy.

- **Walidacja:**
 - Pole wymagane.
 - **Komunikat błędu:** Automatyczny komunikat ASP.NET Core dla pól wymaganych.

3. Address (string):

- **Opis:** Adres usługodawcy (np. ulica i numer domu).
- **Walidacja:**
 - Pole wymagane.
 - **Komunikat błędu:** Automatyczny komunikat ASP.NET Core dla pól wymaganych.

4. City (string):

- **Opis:** Miasto, w którym działa usługodawca.
- **Walidacja:**
 - Pole wymagane.
 - **Komunikat błędu:** Automatyczny komunikat ASP.NET Core dla pól wymaganych.

5. ZipCode (string):

- **Opis:** Kod pocztowy usługodawcy.
- **Walidacja:**
 - Pole wymagane.
 - **Komunikat błędu:** Automatyczny komunikat ASP.NET Core dla pól wymaganych.

6. Latitude (double):

- **Opis:** Szerokość geograficzna lokalizacji usługodawcy.
- **Zastosowanie:** Wykorzystywane do prezentacji lokalizacji na mapach.

7. Longitude (double):

- **Opis:** Długość geograficzna lokalizacji usługodawcy.
- **Zastosowanie:** Wykorzystywane do prezentacji lokalizacji na mapach.

8. Services (List<ServiceViewModel>):

- **Opis:** Lista usług oferowanych przez usługodawcę.
- **Zastosowanie:** Wyświetlanie szczegółowych informacji o usługach w interfejsie użytkownika.

Relacje:

- **Zawiera listę ServiceViewModel:** Uproszczony model usług powiązanych z usługodawcą.

Zastosowanie:

- **Wyświetlanie usługodawców:** Prezentacja danych usługodawcy na stronie klienta.

- **Wyświetlanie usług:** Prezentacja listy usług przypisanych do konkretnego usługodawcy.
- **Integracja z mapami:** Umożliwia lokalizowanie usługodawców na mapach za pomocą współrzędnych geograficznych (Latitude, Longitude).

Model: ServiceScheduleViewModel

Opis:

Reprezentuje widok harmonogramu (grafiku) dla konkretnej usługi, zawierający podstawowe informacje o usłudze oraz listę dostępnych terminów (ScheduleSlots).

Pola:

- ServiceId (int):**
 - **Opis:** Identyfikator usługi.
 - **Zastosowanie:** Używany do identyfikacji usługi, której dotyczy grafik.
- ServiceName (string):**
 - **Opis:** Nazwa usługi.
 - **Walidacja:**
 - Pole wymagane.
 - **Komunikat błędu:** Automatyczny komunikat ASP.NET Core dla pól wymaganych.
- ScheduleSlots (List<ScheduleSlotViewModel>):**
 - **Opis:** Lista terminów dostępnych dla danej usługi.
 - **Walidacja:**
 - Pole wymagane.
 - **Komunikat błędu:** Automatyczny komunikat ASP.NET Core dla pól wymaganych.

Relacje:

- **Zawiera listę ScheduleSlotViewModel:**
Każdy element reprezentuje pojedynczy termin dostępny dla tej usługi.

Zastosowanie:

- **Wyświetlanie grafiku:** Prezentacja listy dostępnych terminów dla konkretnej usługi na stronie klienta lub usługodawcy.
- **Interakcja z klientem:** Umożliwia klientom wybór terminu do rezerwacji.
- **Łatwe zarządzanie:** Służy jako model do zarządzania harmonogramami usługi przez usługodawcę.

Model: ScheduleSlotViewModel

Opis:

Reprezentuje widok pojedynczego dostępnego terminu w grafiku dla danej usługi, zawierający szczegóły dotyczące czasu rozpoczęcia i zakończenia oraz informacji, czy termin jest już zarezerwowany.

Pola:

1. SlotId (int):

- **Opis:** Identyfikator terminu.
- **Zastosowanie:** Używany do identyfikacji terminu w bazie danych lub podczas interakcji z użytkownikiem (np. rezerwacji).

2. StartTime (DateTime):

- **Opis:** Data i czas rozpoczęcia terminu.
- **Zastosowanie:** Określa, kiedy termin się rozpoczyna.

3. EndTime (DateTime):

- **Opis:** Data i czas zakończenia terminu.
- **Zastosowanie:** Określa, kiedy termin się kończy.

4. IsBooked (bool):

- **Opis:** Informacja o tym, czy termin jest już zarezerwowany.
- **Domyślna wartość:** false (domyślnie termin jest dostępny).
- **Zastosowanie:** Wskazuje, czy termin jest dostępny do rezerwacji.

Relacje:

- Model jest częścią listy w **ServiceScheduleViewModel**, co umożliwia zarządzanie wieloma terminami w kontekście jednej usługi.

Zastosowanie:

- **Wyświetlanie dostępnych terminów:** Prezentacja czasu rozpoczęcia i zakończenia terminów w harmonogramie usługi.
- **Rezerwacja terminu:** Informacja o dostępności (IsBooked) pozwala określić, czy termin może być wybrany przez klienta.
- **Zarządzanie grafikami:** Usługodawca może dodawać, usuwać lub edytować terminy w harmonogramie.

Model: ScheduleSlot

Opis:

Reprezentuje pojedynczy termin w harmonogramie usługodawcy dla konkretnej usługi. Zawiera informacje o czasie rozpoczęcia i zakończenia, statusie rezerwacji oraz powiązania z usługą i usługodawcą.

Pola:

1. Id (int):

- **Opis:** Klucz główny identyfikujący termin w bazie danych.

2. ServiceProviderId (int):

- **Opis:** Identyfikator usługodawcy, który oferuje ten termin.
- **Zastosowanie:** Służy do powiązania terminu z konkretnym usługodawcą.

3. ServiceId (int):

- **Opis:** Identyfikator usługi, której dotyczy ten termin.
- **Zastosowanie:** Powiązanie terminu z usługą.

4. Service (Service):

- **Opis:** Nawigacyjne pole do obiektu usługi powiązanej z terminem.
- **Zastosowanie:** Umożliwia dostęp do szczegółowych informacji o usłudze.

5. StartTime (DateTime):

- **Opis:** Data i czas rozpoczęcia terminu.
- **Walidacja:** Pole wymagane.
- **Zastosowanie:** Określa, kiedy termin zaczyna być dostępny.

6. EndTime (DateTime):

- **Opis:** Data i czas zakończenia terminu.
- **Walidacja:**
 - Pole wymagane.
 - Atrybut niestandardowy EndTimeAfterStartTime zapewnia, że EndTime jest późniejsze niż StartTime.
 - **Komunikat błędu:** "EndTime must be later than StartTime."

7. IsBooked (bool):

- **Opis:** Informacja, czy termin został zarezerwowany.
- **Domyślna wartość:** false (domyślnie termin jest dostępny).
- **Zastosowanie:** Określa, czy termin może być rezerwowany.

Relacje:

- **Jeden-do-jeden z Service:** Każdy termin jest przypisany do jednej usługi.
- **Jeden-do-wielu z ServiceProvider:** Każdy termin jest przypisany do jednego usługodawcy.

Zastosowanie:

- **Wyświetlanie grafiku:** Usługodawca może zarządzać swoimi dostępnymi terminami (dodawać, usuwać, edytować).
- **Rezerwacja:** Klient może rezerwować terminy, które nie są oznaczone jako IsBooked = true.
- **Walidacja czasowa:** Dzięki atrybutowi EndTimeAfterStartTime zapewniona jest spójność danych dotyczących czasu.

Modele Microsoft Identity

Główne modele, które wykorzystano:

1. IdentityUser

- **Opis:**
 - Reprezentuje użytkownika w systemie.
- **Kluczowe pola:**
 - Id – unikalny identyfikator użytkownika.
 - UserName – nazwa użytkownika (login).
 - Email – adres e-mail użytkownika.
 - PasswordHash – zaszyfrowane hasło użytkownika.
 - PhoneNumber – numer telefonu.

2. IdentityRole

- **Opis:**
 - Reprezentuje rolę (np. "Admin", "Customer", "ServiceProvider").
- **Kluczowe pola:**
 - Id – unikalny identyfikator roli.
 - Name – nazwa roli.

3. IdentityUserRole<TUser>

- **Opis:**
 - Łączy użytkowników z rolami w relacji wiele-do-wielu.
- **Kluczowe pola:**
 - UserId – identyfikator użytkownika.
 - RoleId – identyfikator roli.

4. IdentityUserClaim<TUser>

- **Opis:**
 - Reprezentuje przypisane do użytkownika roszczenia (claims) dla dodatkowych danych, np. uprawnień.
- **Kluczowe pola:**
 - UserId – identyfikator użytkownika.
 - ClaimType i ClaimValue – typ i wartość roszczenia.

5. IdentityRoleClaim<TRole>

- **Opis:**
 - Reprezentuje roszczenia przypisane do ról.
- **Kluczowe pola:**

- RoleId – identyfikator roli.
- ClaimType i ClaimValue – typ i wartość roszczenia.

6. IdentityUserLogin<TUser>

- **Opis:**
 - Reprezentuje dane logowania zewnętrznego.
- **Kluczowe pola:**
 - LoginProvider – dostawca logowania.
 - ProviderKey – identyfikator użytkownika dostawcy.
 - UserId – identyfikator użytkownika.

7. IdentityUserToken<TUser>

- **Opis:**
 - Reprezentuje tokeny użytkownika (np. tokeny uwierzytelnienia dla API).
- **Kluczowe pola:**
 - UserId – identyfikator użytkownika.
 - LoginProvider – dostawca tokena
 - Name i Value – nazwa i wartość tokena.

Zastosowanie:

- IdentityUser: Reprezentacja użytkownika w bazie danych.
- IdentityRole: Zarządzanie rolami i uprawnieniami.

Kontrolery

Kontroler: AboutController

Opis:

Odpowiada za obsługę strony "O nas" aplikacji.

Metody:

1. Index

- **Metoda HTTP:** GET
- **Parametry:** Brak
- **Opis:**
 - Wyświetla stronę informacyjną "O nas" z zawartością statyczną.
- **Zwracane dane:**
 - Widok Index.

Kontroler: CustomerController

Opis:

Obsługuje akcje związane z klientami, takie jak przeglądanie usług, harmonogramów, rezerwowanie terminów oraz przeglądanie historii rezerwacji.

Metody:

1. ServiceSchedule

- **Metoda HTTP:** GET
- **Parametry:**
 - id (int) – Identyfikator usługi.
- **Opis:**

Wyświetla grafik dostępnych terminów (ScheduleSlots) dla wybranej usługi.
- **Zwracane dane:**

Widok z modelem ServiceScheduleViewModel, zawierający terminy dostępne do rezerwacji.

2. ServiceProviderDetails

- **Metoda HTTP:** GET
- **Parametry:**
 - id (int) – Identyfikator usługodawcy.
- **Opis:**

Wyświetla szczegóły wybranego usługodawcy, w tym jego usługi oraz lokalizację.
- **Zwracane dane:**

Widok z modelem ServiceProviderViewModel, zawierający dane usługodawcy i jego usługi.

3. ServiceProviders

- **Metoda HTTP:** GET
- **Parametry:**
 - searchQuery (string) – Fraza wyszukiwania (opcjonalnie).
 - sortOrder (string) – Kryterium sortowania (opcjonalnie, np. "name_asc", "city").
- **Opis:**

Wyświetla listę usługodawców z opcjami wyszukiwania i sortowania.

- **Zwracane dane:**
Widok z listą usługodawców (List<ServiceProviderViewModel>).

4. BookService

- **Metoda HTTP:** POST
- **Atrybuty:** [Authorize]
- **Parametry:**
 - id (int) – Identyfikator wybranego terminu (ScheduleSlot).
- **Opis:**
Tworzy rezerwację dla zalogowanego klienta na wybrany termin.
- **Zwracane dane:**
 - Przekierowanie do widoku harmonogramu usługi (ServiceSchedule).
 - Komunikat sukcesu lub błędu w zależności od wyniku operacji.

5. MyReservations

- **Metoda HTTP:** GET
- **Parametry:** Brak.
- **Opis:**
Wyświetla listę rezerwacji zalogowanego klienta.
- **Zwracane dane:**
Widok z modelem List<ReservationViewModel> zawierającym szczegóły rezerwacji.

Kontroler: HomeController

Opis:

Obsługuje podstawowe strony aplikacji, w tym stronę główną dostosowaną do ról użytkownika oraz stronę błędów.

Metody:

1. Index

- **Metoda HTTP:** GET
- **Atrybuty:** [Authorize]
- **Parametry:** Brak.
- **Opis:**
Wyświetla stronę główną aplikacji dostosowaną do zalogowanego użytkownika:
 - Jeśli użytkownik ma rolę **Customer**, renderuje widok CustomerIndex.

- Jeśli użytkownik ma rolę **ServiceProvider**, renderuje widok ServiceProviderIndex.
- Jeśli użytkownik nie ma przypisanej roli, renderuje domyślny widok Index.
- **Zwracane dane:**
Odpowiedni widok (CustomerIndex, ServiceProviderIndex, lub Index).

2. Error

- **Metoda HTTP:** GET
- **Atrybuty:** [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]
- **Parametry:** Brak.
- **Opis:**
Wyświetla stronę błędów aplikacji. Informacje o błędzie, takie jak RequestId, są przekazywane w modelu ErrorViewModel.
- **Zwracane dane:**
Widok Error z modelem ErrorViewModel.

Kontroler: ReservationsController

Opis:

Obsługuje akcje związane z zarządzaniem rezerwacjami przez klientów. Klient może przeglądać, tworzyć, edytować oraz usuwać swoje rezerwacje.

Atrybuty kontrolera:

- [Authorize(Roles = "Customer")]: Ogranicza dostęp do kontrolera wyłącznie dla użytkowników z rolą **Customer**.

Metody:

1. Index

- **Metoda HTTP:** GET
- **Parametry:** Brak.
- **Opis:**
Wyświetla listę wszystkich rezerwacji należących do zalogowanego klienta.
- **Zwracane dane:**
Widok z modelem zawierającym listę rezerwacji klienta.

2. Details

- **Metoda HTTP:** GET
- **Parametry:**
 - id (int): Identyfikator rezerwacji.
- **Opis:**
Wyświetla szczegóły wybranej rezerwacji.
Zabezpiecza przed dostępem do danych innych klientów.
- **Zwracane dane:**
Widok z modelem rezerwacji lub kod 403 Forbid, jeśli klient nie ma uprawnień.

3. Create

- **Metoda HTTP:** GET
- **Parametry:** Brak.
- **Opis:**
Wyświetla formularz do tworzenia nowej rezerwacji.
- **Zwracane dane:**
Widok formularza Create.

4. Create

- **Metoda HTTP:** POST
- **Atrybuty:** [ValidateAntiForgeryToken]
- **Parametry:**
 - reservation (Reservation): Obiekt nowej rezerwacji przestany z formularza.
- **Opis:**
Tworzy nową rezerwację dla zalogowanego klienta z domyślnym statusem Pending.
- **Zwracane dane:**
 - Przekierowanie do Index w przypadku sukcesu.
 - Widok z błędami walidacji w przypadku niepowodzenia.

5. Edit

- **Metoda HTTP:** GET
- **Parametry:**
 - id (int): Identyfikator rezerwacji.
- **Opis:**
Wyświetla formularz do edycji wybranej rezerwacji.
Zabezpiecza przed dostępem do rezerwacji innych klientów.
- **Zwracane dane:**
Widok formularza Edit lub kod 403 Forbid, jeśli klient nie ma uprawnień.

6. Edit

- **Metoda HTTP:** POST
- **Atrybuty:** [ValidateAntiForgeryToken]
- **Parametry:**
 - id (int): Identyfikator rezerwacji.
 - reservation (Reservation): Zaktualizowany obiekt rezerwacji przestany z formularza.
- **Opis:**

Aktualizuje dane istniejącej rezerwacji, jeśli walidacja zakończy się sukcesem.
- **Zwracane dane:**
 - Przekierowanie do Index w przypadku sukcesu.
 - Widok z błędami walidacji w przypadku niepowodzenia.

7. Delete

- **Metoda HTTP:** GET
- **Parametry:**
 - id (int): Identyfikator rezerwacji.
- **Opis:**

Wyświetla stronę potwierdzenia usunięcia wybranej rezerwacji.
Zabezpiecza przed dostępem do rezerwacji innych klientów.
- **Zwracane dane:**

Widok Delete lub kod 403 Forbid, jeśli klient nie ma uprawnień.

8. DeleteConfirmed

- **Metoda HTTP:** POST
- **Atrybuty:** [ValidateAntiForgeryToken], [ActionName("Delete")]
- **Parametry:**
 - id (int): Identyfikator rezerwacji.
- **Opis:**

Usuwa rezerwację po potwierdzeniu operacji przez klienta.
- **Zwracane dane:**
 - Przekierowanie do Index w przypadku sukcesu.
 - Kod 403 Forbid, jeśli klient nie ma uprawnień.

Podsumowanie:

- **Index:** Lista wszystkich rezerwacji klienta.
- **Details:** Szczegóły wybranej rezerwacji.
- **Create:** Tworzenie nowej rezerwacji.

- **Edit:** Edycja istniejącej rezerwacji.
- **Delete:** Wyświetlenie potwierdzenia usunięcia rezerwacji.
- **DeleteConfirmed:** Usunięcie wybranej rezerwacji.

Kontroler: ScheduleController

Opis:

Obsługuje zarządzanie grafikami usługodawcy, w tym dodawanie, edycję, usuwanie terminów oraz importowanie i generowanie szablonów plików Excel.

Atrybuty kontrolera:

- [Authorize(Roles = "ServiceProvider")] – Dostęp ograniczony wyłącznie do użytkowników z rolą **ServiceProvider**.

Metody:

1. Create (GET)

- **Metoda HTTP:** GET
- **Parametry:** Brak.
- **Opis:**
Wyświetla formularz do dodawania nowego terminu.
- **Zwracane dane:**
Widok Create z listą usług usługodawcy (ViewBag.Services).

2. Create (POST)

- **Metoda HTTP:** POST
- **Atrybuty:** [ValidateAntiForgeryToken]
- **Parametry:**
 - model (ScheduleSlot) – Obiekt nowego terminu przesyłany z formularza.
- **Opis:**
Dodaje nowy termin do grafiku po walidacji danych.
- **Zwracane dane:**
 - Przekierowanie do Index po sukcesie.
 - Komunikat błędu w przypadku niepowodzenia walidacji.

3. ServiceSchedule

- **Metoda HTTP:** GET
- **Parametry:**
 - id (int) – Identyfikator usługi.
 - from (DateTime?) – Data początkowa (opcjonalnie).

- to (DateTime?) – Data końcowa (opcjonalnie).
- **Opis:**
Wyświetla grafik terminów dla konkretnej usługi w zadanym zakresie dat.
- **Zwracane dane:**
Widok ServiceSchedule z modelem ServiceScheduleViewModel.

4. Index

- **Metoda HTTP:** GET
- **Parametry:**
 - from (DateTime?) – Data początkowa (opcjonalnie).
 - to (DateTime?) – Data końcowa (opcjonalnie).
- **Opis:**
Wyświetla listę wszystkich terminów usługodawcy w zadanym zakresie dat.
- **Zwracane dane:**
Widok Index z listą terminów (List<ScheduleSlot>).

5. Delete (GET)

- **Metoda HTTP:** GET
- **Parametry:**
 - id (int) – Identyfikator terminu.
- **Opis:**
Wyświetla stronę potwierdzenia usunięcia terminu.
Zabezpiecza przed usuwaniem terminów z rezerwacjami.
- **Zwracane dane:**
Widok Delete.

6. DeleteConfirmed (POST)

- **Metoda HTTP:** POST
- **Atrybuty:** [ValidateAntiForgeryToken]
- **Parametry:**
 - id (int) – Identyfikator terminu.
- **Opis:**
Usuwa termin, jeśli nie jest zarezerwowany.
- **Zwracane dane:**
 - Przekierowanie do Index po sukcesie.
 - Komunikat błędu, jeśli termin ma przypisaną rezerwację.

7. ImportExcel (GET)

- **Metoda HTTP:** GET
- **Parametry:** Brak.
- **Opis:**
Wyświetla formularz do importowania plików Excel z terminami.
- **Zwracane dane:**
Widok ImportExcel.

8. ImportExcel (POST)

- **Metoda HTTP:** POST
- **Atrybuty:** [ValidateAntiForgeryToken]
- **Parametry:**
 - excelFile (IFormFile) – Przesłany plik Excel.
- **Opis:**
Importuje terminy z przesłanego pliku Excel, walidując wprowadzone dane (np. brak dat w przeszłości).
- **Zwracane dane:**
 - Przekierowanie do Index z komunikatem sukcesu lub błędu.

9. GenerateTemplate

- **Metoda HTTP:** GET
- **Parametry:** Brak.
- **Opis:**
Generuje szablon pliku Excel z przykładowymi terminami oraz listą dostępnych usług dla usługodawcy.
- **Zwracane dane:**
Plik Excel do pobrania.

Podsumowanie:

- **Zarządzanie grafikami:**
Usługodawcy mogą dodawać, edytować, usuwać i importować terminy.
- **Eksport i import Excel:**
Obsługa generowania szablonów i wczytywania terminów z plików.
- **Bezpieczeństwo:**
Dostęp ograniczony do usługodawców z rolą ServiceProvider.

Kontroler: ServiceProviderPanelController

Opis:

Obsługuje panel zarządzania usługami i rezerwacjami przez usługodawcę. Umożliwia przeglądanie, tworzenie, edytowanie i usuwanie usług, zarządzanie grafikami i rezerwacjami.

Atrybuty kontrolera:

- [Authorize(Roles = "ServiceProvider")] – Dostęp ograniczony do użytkowników z rolą **ServiceProvider**.

Metody:

Zarządzanie usługami:

1. Index

- **Metoda HTTP:** GET
- **Parametry:** Brak.
- **Opis:**
Wyświetla listę usług dodanych przez usługodawcę.
- **Zwracane dane:**
Widok z listą usług.

2. Create (GET)

- **Metoda HTTP:** GET
- **Parametry:** Brak.
- **Opis:**
Wyświetla formularz do dodawania nowej usługi.
- **Zwracane dane:**
Widok Create.

3. Create (POST)

- **Metoda HTTP:** POST
- **Atrybuty:** [ValidateAntiForgeryToken]
- **Parametry:**
 - model (Service) – Obiekt nowej usługi.
- **Opis:**
Dodaje nową usługę, ograniczając ich liczbę do 20 na usługodawcę.
- **Zwracane dane:**
Przekierowanie do Index.

4. EditService (GET)

- **Metoda HTTP:** GET
- **Parametry:**
 - id (int) – Identyfikator usługi.

- **Opis:**
Wyświetla formularz do edycji istniejącej usługi.
- **Zwracane dane:**
Widok EditService.

5. EditService (POST)

- **Metoda HTTP:** POST
- **Atrybuty:** [ValidateAntiForgeryToken]
- **Parametry:**
 - model (Service) – Zaktualizowana usługa.
- **Opis:**
Aktualizuje dane usługi.
- **Zwracane dane:**
Przekierowanie do Index.

6. Delete

- **Metoda HTTP:** GET
- **Parametry:**
 - id (int) – Identyfikator usługi.
- **Opis:**
Wyświetla formularz potwierdzenia usunięcia usługi.
- **Zwracane dane:**
Widok Delete.

Zarządzanie grafikami:

7. Schedule

- **Metoda HTTP:** GET
- **Parametry:**
 - from (DateTime?) – Data początkowa (opcjonalnie).
 - to (DateTime?) – Data końcowa (opcjonalnie).
- **Opis:**
Wyświetla grafik dostępnych terminów w wybranym zakresie dat.
- **Zwracane dane:**
Widok Schedule z listą terminów.

Zarządzanie rezerwacjami:

8. ManageReservations

- **Metoda HTTP:** GET
- **Parametry:**
 - from (DateTime?) – Data początkowa (opcjonalnie).

- to (DateTime?) – Data końcowa (opcjonalnie).
- **Opis:**
Wyświetla listę rezerwacji usługodawcy w wybranym zakresie dat.
- **Zwracane dane:**
Widok ManageReservations.

9. UpdateReservationStatus

- **Metoda HTTP:** POST
- **Parametry:**
 - id (int) – Identyfikator rezerwacji.
 - status (string) – Nowy status rezerwacji.
- **Opis:**
Aktualizuje status rezerwacji.
- **Zwracane dane:**
Przekierowanie do ManageReservations.

Pomocnicze:

10. GetServiceProviderIdByUserId

- **Metoda HTTP:** Brak (metoda pomocnicza).
- **Parametry:**
 - userId (string) – Identyfikator użytkownika.
- **Opis:**
Pobiera identyfikator usługodawcy na podstawie użytkownika.
- **Zwracane dane:**
Identyfikator usługodawcy (int).

Podsumowanie:

- **Zarządzanie usługami:** Dodawanie, edycja i usuwanie usług.
- **Zarządzanie grafikami:** Wyświetlanie terminów.
- **Zarządzanie rezerwacjami:** Przegląd i aktualizacja rezerwacji.
- **Bezpieczeństwo:** Wszystkie metody wymagają roli **ServiceProvider**.

Kontroler: ServicesController

Opis:

Obsługuje zarządzanie usługami przez usługodawcę, takie jak wyświetlanie, tworzenie, edytowanie i usuwanie usług.

Atrybuty kontrolera:

- [Authorize(Roles = "ServiceProvider")] – Dostęp ograniczony do użytkowników z rolą **ServiceProvider**.

Metody:

1. Index

- **Metoda HTTP:** GET
- **Parametry:** Brak.
- **Opis:**
Wyświetla listę wszystkich usług dodanych przez usługodawcę.
- **Zwracane dane:**
Widok z listą usług.

2. Details

- **Metoda HTTP:** GET
- **Parametry:**
 - id (int) – Identyfikator usługi.
- **Opis:**
Wyświetla szczegóły konkretnej usługi.
- **Zwracane dane:**
Widok z modelem usługi lub 404 NotFound, jeśli usługa nie istnieje.

3. Create (GET)

- **Metoda HTTP:** GET
- **Parametry:** Brak.
- **Opis:**
Wyświetla formularz do dodania nowej usługi.
- **Zwracane dane:**
Widok Create.

4. Create (POST)

- **Metoda HTTP:** POST
- **Atrybuty:** [ValidateAntiForgeryToken]
- **Parametry:**
 - service (Service) – Obiekt nowej usługi.
- **Opis:**
Dodaje nową usługę, jeśli dane są poprawne.
- **Zwracane dane:**
 - Przekierowanie do Index po sukcesie.

- Widok z błędami walidacji w przypadku niepowodzenia.

5. Edit (GET)

- **Metoda HTTP:** GET
- **Parametry:**
 - id (int) – Identyfikator usługi.
- **Opis:**

Wyświetla formularz edycji dla wybranej usługi.
- **Zwracane dane:**

Widok Edit lub 404 NotFound, jeśli usługa nie istnieje.

6. Edit (POST)

- **Metoda HTTP:** POST
- **Atrybuty:** [ValidateAntiForgeryToken]
- **Parametry:**
 - id (int) – Identyfikator usługi.
 - service (Service) – Zaktualizowana usługa.
- **Opis:**

Aktualizuje dane usługi po walidacji i sprawdzeniu istnienia.
- **Zwracane dane:**
 - Przekierowanie do Index po sukcesie.
 - Widok z błędami walidacji w przypadku niepowodzenia.

7. Delete (GET)

- **Metoda HTTP:** GET
- **Parametry:**
 - id (int) – Identyfikator usługi.
- **Opis:**

Wyświetla formularz potwierdzenia usunięcia usługi.
- **Zwracane dane:**

Widok Delete lub 404 NotFound, jeśli usługa nie istnieje.

8. DeleteConfirmed

- **Metoda HTTP:** POST
- **Atrybuty:** [ValidateAntiForgeryToken], [ActionName("Delete")]
- **Parametry:**
 - id (int) – Identyfikator usługi.

- **Opis:**
Usuwa wybraną usługę po potwierdzeniu.
- **Zwracane dane:**
 - Przekierowanie do Index po sukcesie.
 - 404 NotFound, jeśli usługa nie istnieje.

Podsumowanie:

- **Index:** Lista wszystkich usług usługodawcy.
- **Details:** Szczegóły wybranej usługi.
- **Create:** Dodawanie nowej usługi.
- **Edit:** Edytowanie istniejącej usługi.
- **Delete:** Usuwanie usługi z potwierdzeniem.

Kontroler zapewnia kompleksowe zarządzanie usługami dla usługodawców. Wszystkie operacje są bezpieczne i ograniczone do zalogowanych użytkowników z odpowiednią rolą.

Opis systemu użytkowników

Role w systemie

1. Customer (Klient):

- **Przeznaczenie:** Reprezentuje użytkownika, który może przeglądać dostępnych usługodawców i ich usługi, rezerwować terminy.
- **Uprawnienia:**
 - Przeglądanie usług i usługodawców.
 - Rezerwacja terminów.
 - Przegląd swoich rezerwacji.

2. ServiceProvider (Usługodawca):

- **Przeznaczenie:** Reprezentuje użytkownika, który oferuje usługi i zarządza swoim grafikiem oraz rezerwacjami.
- **Uprawnienia:**
 - Zarządzanie własnymi usługami (dodawanie, edycja).
 - Zarządzanie grafikami (dodawanie terminów, import z pliku Excel).
 - Zarządzanie rezerwacjami (potwierdzanie, odrzucanie, zmiana statusu).