|  |  |
| --- | --- |
| **AI2** | Dokumentacja projektu |
| **Autorzy** | Paweł Kolasa 117845, Przemysław Pisaniak 117845, Damian Stasiak 117835, Michał Woźniak 117845 |
| **Kierunek, rok** | Informatyka, IV rok, st. stacjonarne (3,5-l) |
| **Temat  projektu** | *Projekt i implementacja aplikacji do wyszukiwania opiekunów zwierząt.* |

# Wprowadzenie

Projekt ma na celu stworzenie aplikacji internetowej, która umożliwi łatwe wyszukiwanie opiekunów dla zwierząt. W ramach aplikacji użytkownicy będą mieli dostęp do zasobów publicznych, które obejmują przeglądanie profili opiekunów zwierząt z podstawowymi informacjami, możliwość sortowania oraz filtrowania wyników. Aplikacja Animals-Caretakers jest intuicyjną platformą, która ułatwi proces rekrutacji opiekunów zwierząt, oraz pomoże pracownikom w znalezieniu odpowiednich ofert zatrudnienia.

## Cel projektu

Celem projektu jest dostarczenie efektywnego narzędzia do łączenia opiekunów zwierząt z potencjalnymi pracodawcami, czyli rekruterami. Aplikacja ma ułatwić proces wyszukiwania, selekcji oraz nawiązywania kontaktów między opiekunami zwierząt a osobami poszukującymi ich usług.

## Zakres projektu

Zasoby Publiczne oferują możliwość przeglądania publicznych profili opiekunów zwierząt. Dodatkowo, funkcje sortowania, filtrowania i paginacji ułatwią wygodne przeglądanie wyników, dzięki czemu niezalogowany użytkownik lepszej interakcji z platformą.

Zasoby Publiczne:

* Umożliwienie przeglądania publicznych profili opiekunów zwierząt.
* Implementacja funkcji sortowania, filtrowania i paginacji wyników dla użytkowników niezalogowanych.

Rejestracja Potencjalnych Pracowników:

* Stworzenie mechanizmu rejestracji dla osób poszukujących pracy jako opiekuni zwierząt.
* Umożliwienie potencjalnym pracownikom uzupełniania informacji na temat swojego profilu.

Rejestracja Rekruterów:

* Implementacja procesu rejestracji dla rekruterów, czyli osób poszukujących opiekunów zwierząt.

Typy Zalogowanych Użytkowników:

* Wprowadzenie dwóch typów zalogowanych użytkowników: rekruter (recruiter) oraz pracownik (employee).

Migracje bazy danych obejmujące:

* Tworzenie tabeli użytkowników.
* Dodanie kolumny do śledzenia liczby odwiedzin profilu.
* Utworzenie tabeli zawierającej poufne informacje użytkowników.
* Zmiany w strukturze tabeli poufnych informacji użytkowników.
* Stworzenie tabeli przechowującej plany "Do zobaczenia".
* Dodanie tabeli dla zaproszeń.

# Opis aplikacji

## Perspektywa użytkownika niezalogowanego

Endpoint umożliwia użytkownikowi przeglądanie listy pracowników - opiekunów.

#### Parametry Zapytania

pageSize: Liczba wyników na stronę.

TotalPages: Całkowita liczba stron.

TotalRows: Całkowita liczba wyników.

PageIndex: Numer bieżącej strony.

Sort: Pole do sortowania wyników.

Order: Kierunek sortowania (ASC lub DESC).

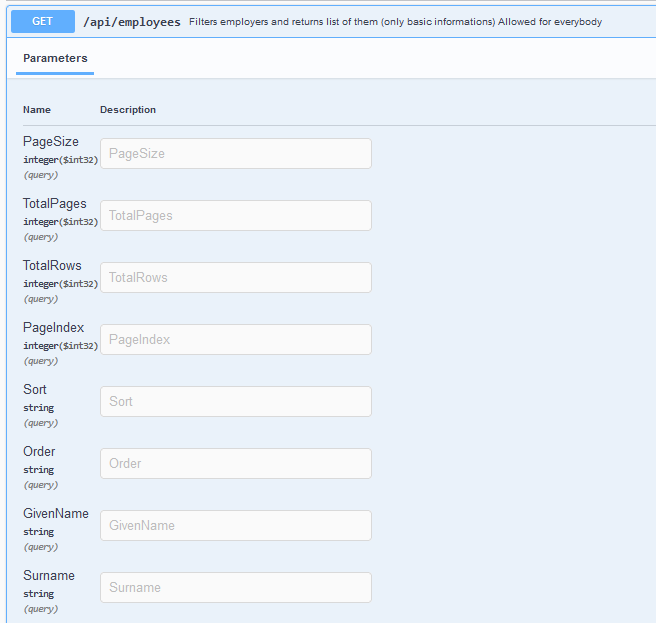
GivenName: Imię pracownika.

Surname: Nazwisko pracownika.

#### Odpowiedzi

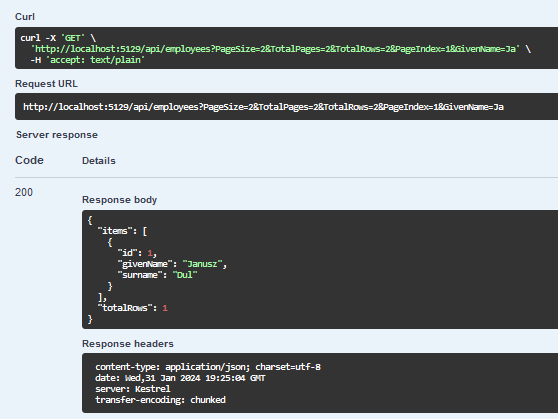
Kod 200 (Success): Lista pracowników została pomyślnie zwrócona.

#### Widok endpointu w Swagger UI.

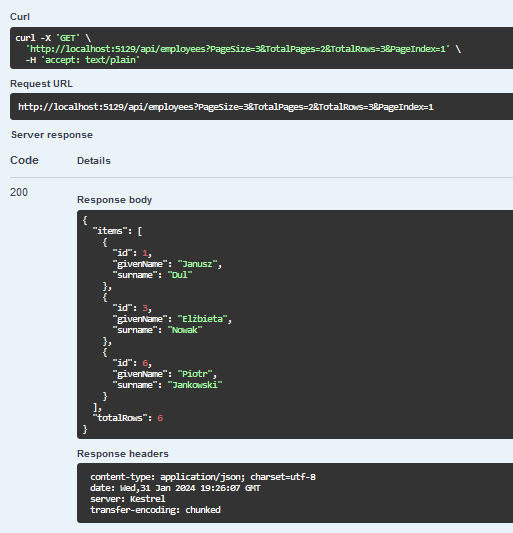


#### Przykładowe Zapytania

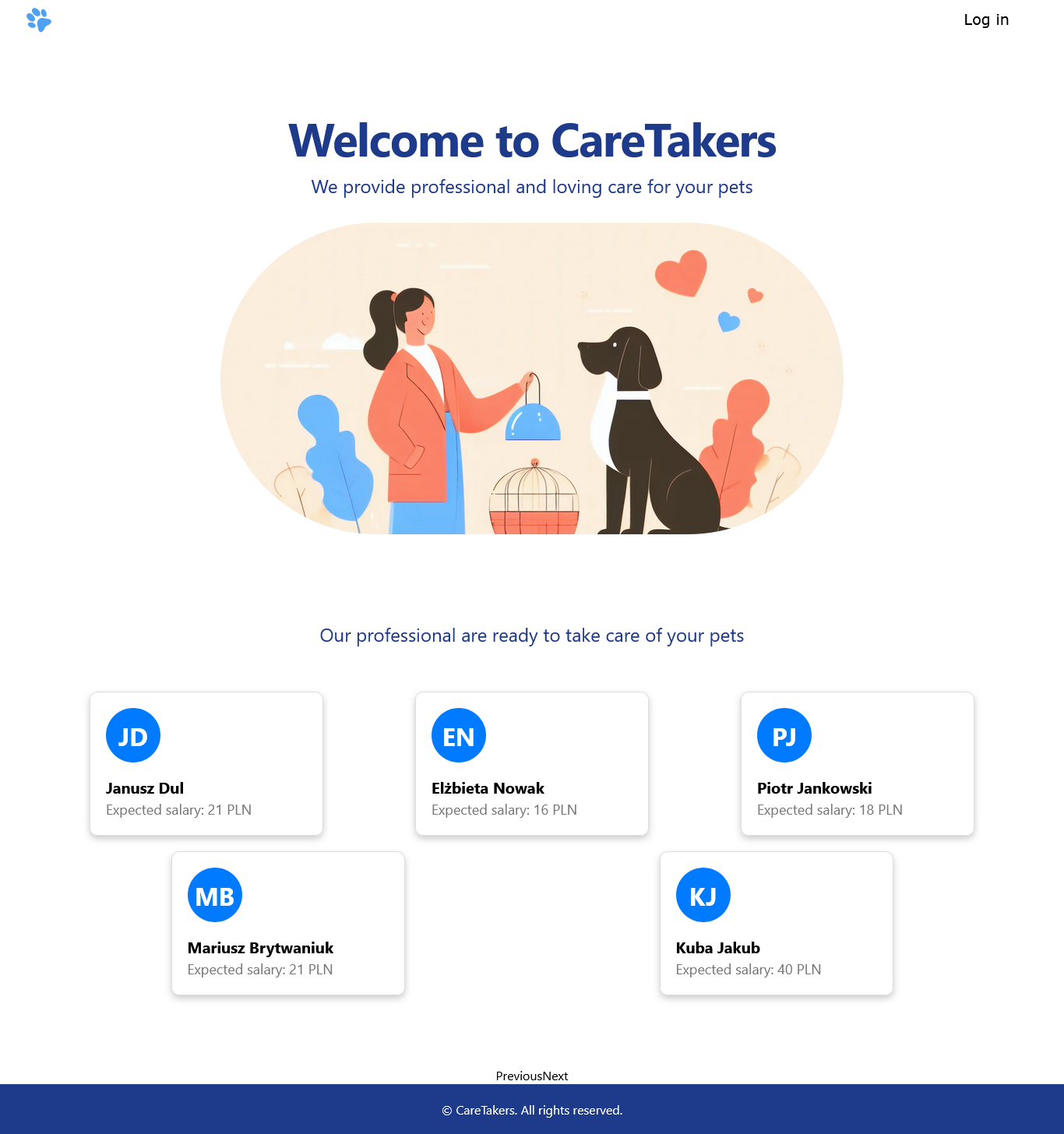
GivenName=Ja



PageSize=3



Widok niezalogowanego użytkownika.



## Rejestracja użytkownika

Endpoint umożliwia użytkownikowi rejestrowanie nowego konta w systemie Animal Caretakers.

#### Parametry Zapytania

givenName: Imię użytkownika.

surname: Nazwisko użytkownika.

username: Nazwa użytkownika (login).

password: Hasło użytkownika.

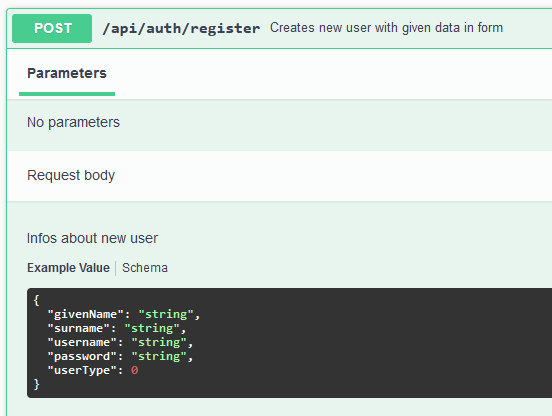
userType: Typ użytkownika (0 - właściciel, 1 - opiekun).

#### Odpowiedzi

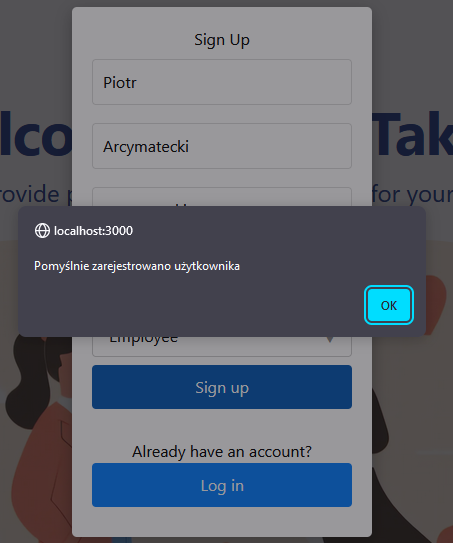
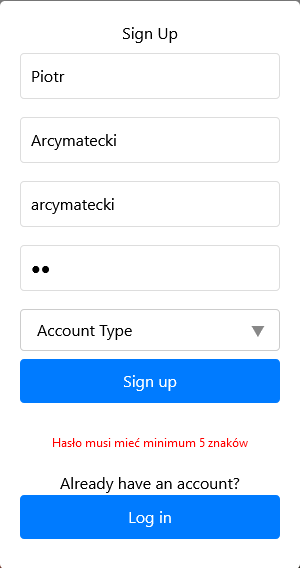
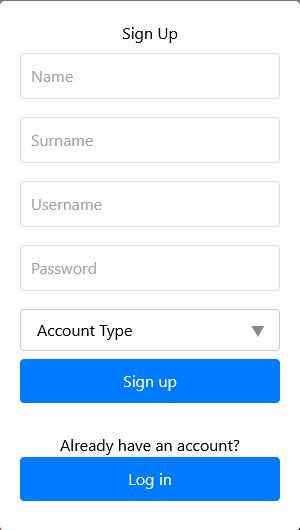
Kod 200 (Success): Konto zostało pomyślnie zarejestrowane.

Kod 400 (Bad Request): Błąd w żądaniu.

Widok endpointu w Swagger UI.



Przykład logowania nowego użytkownika w aplikacji.



## Okno logowania użytkownika

Umożliwia zalogowanie się do aplikacji Animal Caretakers.

#### Parametry Zapytania

username: Nazwa użytkownika (login).

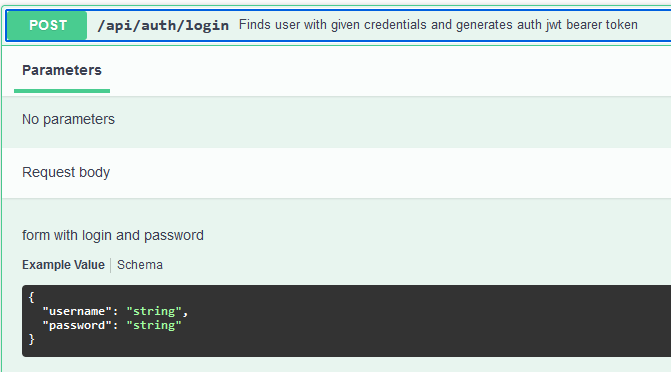
password: Hasło użytkownika.

#### Odpowiedzi

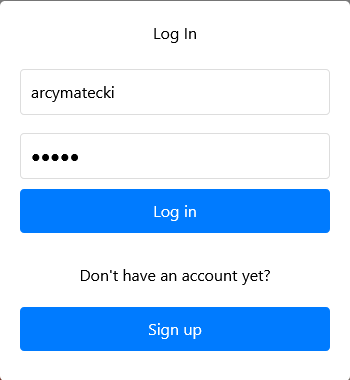
Kod 200 (Success): Zalogowanie pomyślne.

Kod 400 (Bad Request): Błąd w żądaniu, np. nieprawidłowe dane logowania.

#### Widok endpointu, w Swagger UI.

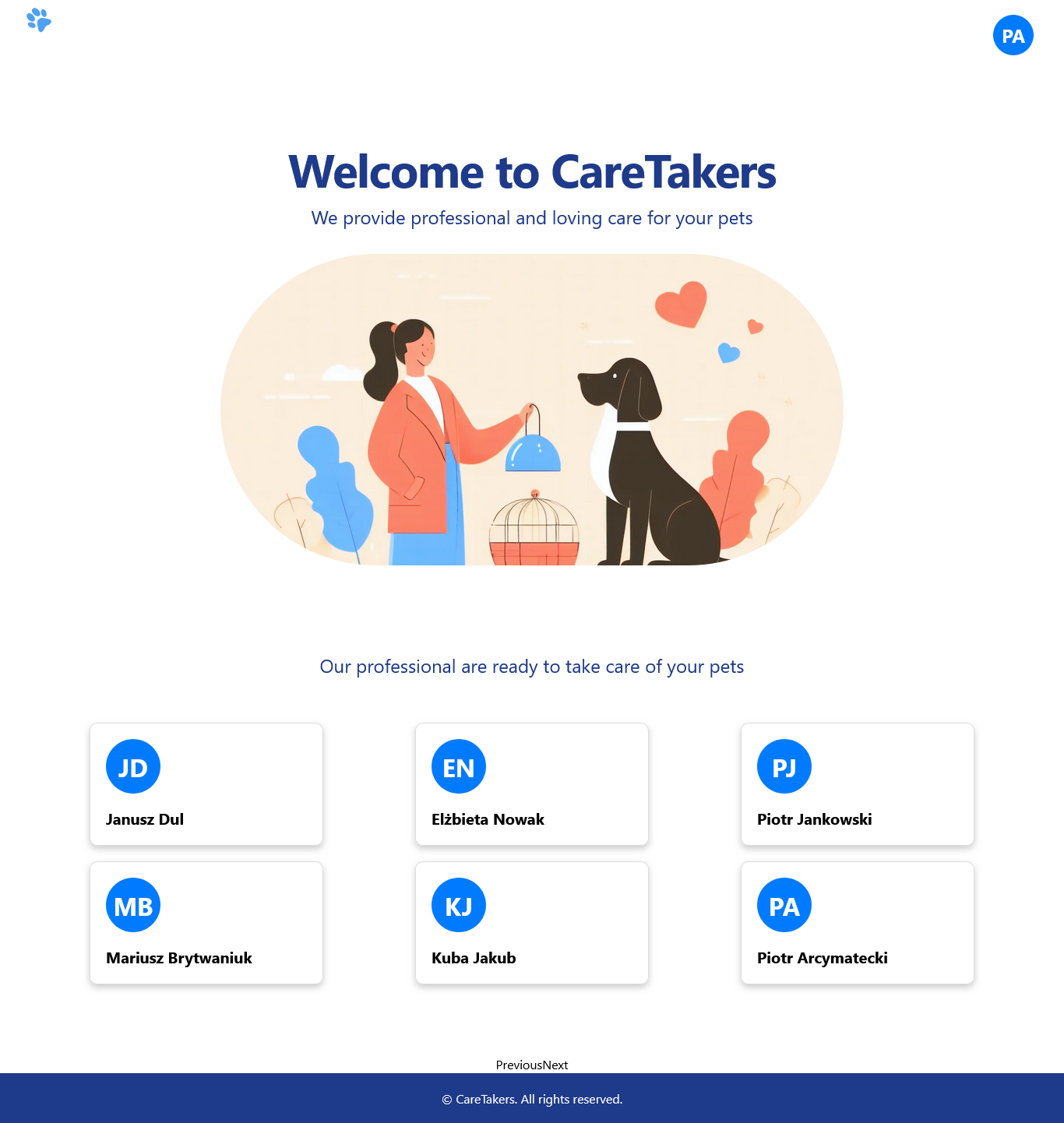


Przykład logowania użytkownika.



## Perspektywa opiekuna

Opiekunowie mają dostęp do strony, która jest identyczna jak dla użytkowników niezalogowanych, jednak z dodatkową możliwością tworzenia spotkań.



## Zarządzanie profilem opiekuna

Opiekunowie mają możliwość zarządzania swoim profilem, dostosowując informacje dotyczące umiejętności, portfolio, sukcesów oraz oczekiwanej pensji.

#### Parametry Zapytania

givenName: Imię opiekuna.

surname: Nazwisko opiekuna.

skills: Umiejętności opiekuna (np. "Znawca psów") oraz flaga isPublic (true/false).

portfolio: Portfolio opiekuna oraz flaga isPublic (true/false).

successes: Sukcesy opiekuna oraz flaga isPublic (true/false).

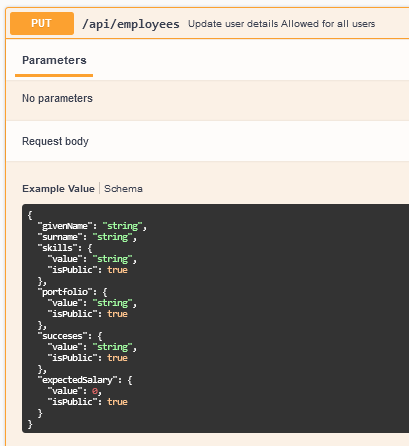
expectedSalary: Oczekiwana pensja opiekuna oraz flaga isPublic (true/false).

#### Odpowiedzi

Kod 200 (Sukces): Profil opiekuna został pomyślnie zaktualizowany.

Kod 404 (NotFound): Nie znaleziono profilu opiekuna.

#### Widok endpointu, w Swagger UI.



#### Przykładowe zapytanie

givenName=Jan

surname=Kowalski

skills={"value":"Znawca psów","isPublic":true}

portfolio={"value":"https://portfolio.com/jankowalski","isPublic":true}

successes={"value":"Wychowanie agresywnego psa","isPublic":true}

expectedSalary={"value":50,"isPublic":true}

Endpoint DELETE umożliwia usuwanie pracowników z systemu Animal Caretakers.

#### Parametry Zapytania

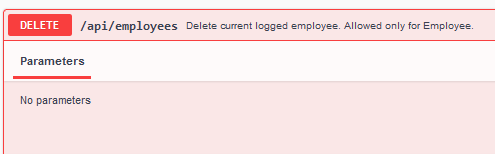
employeeId: Identyfikator pracownika do usunięcia.

#### Odpowiedzi

Kod 200 (Success): Pracownik został pomyślnie usunięty.

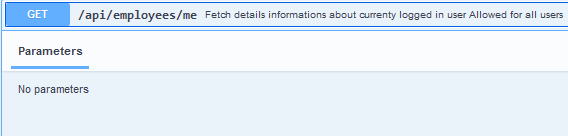
Kod 404 (NotFound): Nie znaleziono pracownika o podanym identyfikatorze.

#### Widok endpointu, w Swagger UI.



Endpoint /api/employees/me jest dostępny dla wszystkich użytkowników i umożliwia pobranie szczegółowych informacji o obecnie zalogowanym użytkowniku.

#### Widok endpointu, w Swagger UI.



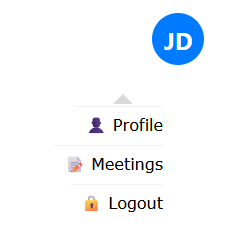
#### Odpowiedzi

Kod 200 (Sukces): Pomyślnie pobrano szczegółowe informacje o zalogowanym użytkowniku.

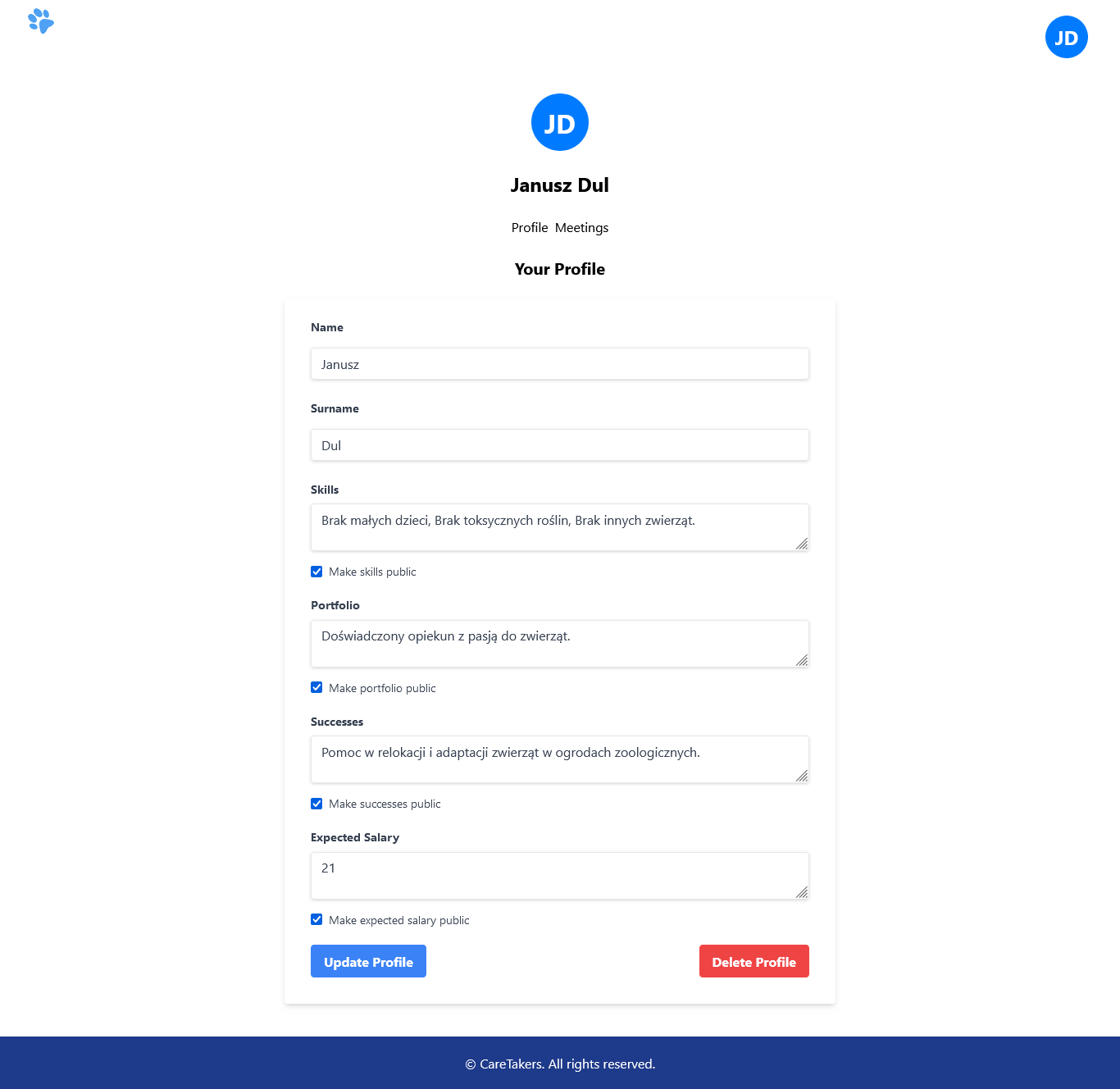
Kod 404 (NotFound): Nie znaleziono informacji o zalogowanym użytkowniku.

Po naciśnięciu inicjałów zalogowanego użytkownika na tle koła w prawym górnym rogu strony,

pojawia się panel z zakładkami: "Profile", "Meetings" i "Logout". Te zakładki umożliwiają użytkownikowi dostęp do wybranych endpointów. Dodatkowo, wykorzystanie emotikonów ułatwia intuicyjną interakcję z interfejsem użytkownika, nadając mu przyjazny charakter.



Po kliknięciu w Profile użytkownik przechodzi do widoku, w którym ma pełną kontrolę nad swoimi danymi. Może edytować podstawowe informacje, takie jak imię, nazwisko, umiejętności, portfolio, sukcesy oraz oczekiwaną pensję. Ponadto, użytkownik ma możliwość zdecydowania, czy chce udostępnić te informacje publicznie. Istnieje także opcja usunięcia konta, z aplikacji Animal Caretakers.



## Spotkanie

Endpoint umożliwiający filtrowanie, paginację oraz sortowanie spotkań na podstawie określonych parametrów. Dostępny dla zalogowanych użytkowników każdego typu.

#### Parametry Zapytania

pagesize: Liczba wyników na stronę.

totalpages: Całkowita liczba stron.

totalrows: Całkowita liczba wyników.

pageindex: Numer bieżącej strony.

sort: Pole do sortowania wyników.

order: Kierunek sortowania (ASC lub DESC).

status: Status spotkania (0 - nowe, 1 - anulowane, 2 - zakończone).

meetingdatefrom: Data początkowa zakresu spotkań.

meetingdateto: Data końcowa zakresu spotkań.

#### Odpowiedzi

Kod 200 (Sukces): Spotkania zostały pomyślnie pobrane.

#### Widok endpointu, w Swagger UI.

#### 

## Tworzenie nowego spotkania

Endpoint umożliwiający rekruterowi (opiekunowi zwierząt) utworzenie nowego spotkania z pracownikiem. Dostępny wyłącznie dla rekrutera.

#### Parametry Zapytania

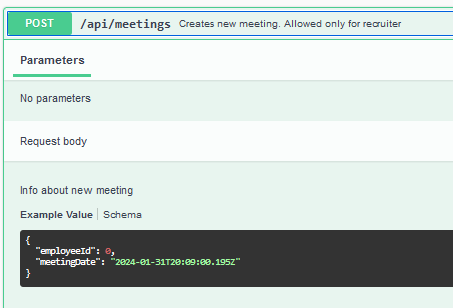
EmployeeId: Identyfikator pracownika, z którym ma zostać zorganizowane spotkanie.

meetingDate: Data spotkania.

#### Odpowiedzi

Kod 200 (Sukces): Spotkanie zostało pomyślnie utworzone.

#### Widok endpointu, w Swagger UI.



Endpointy PUT umożliwiają aktualizację statusu spotkania na "anulowane" lub zakończone. Dostępne wyłącznie dla rekrutera - opiekuna.

#### Parametry Zapytania

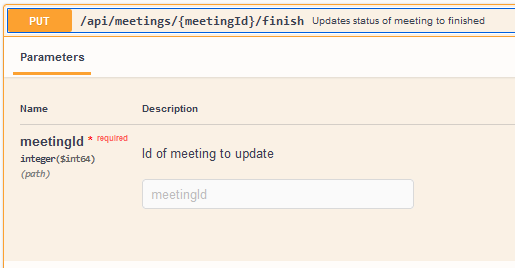
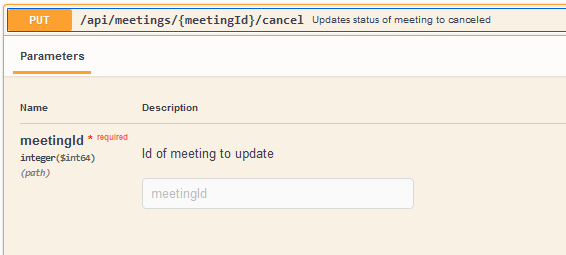
meetingId: Identyfikator spotkania do anulowania.

#### Odpowiedzi

Kod 200 (Sukces): Status spotkania został pomyślnie zaktualizowany na "anulowane".

Kod 404 (Not Found): Nie znaleziono spotkania o podanym identyfikatorze.

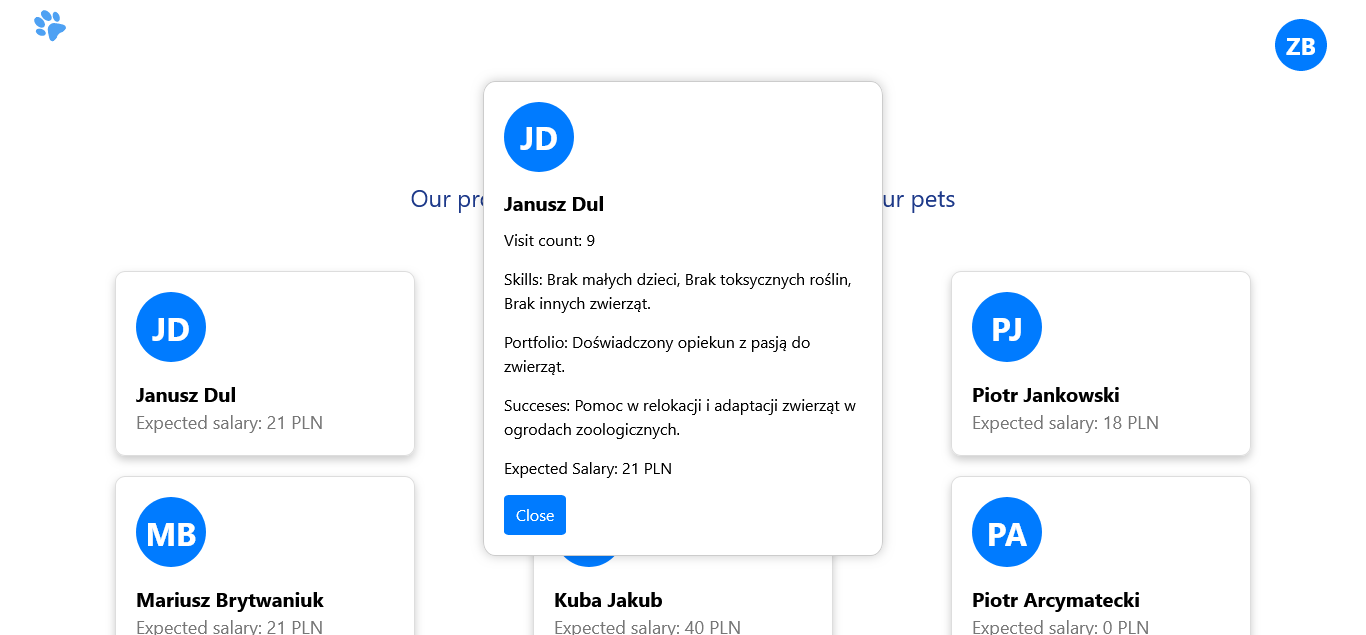
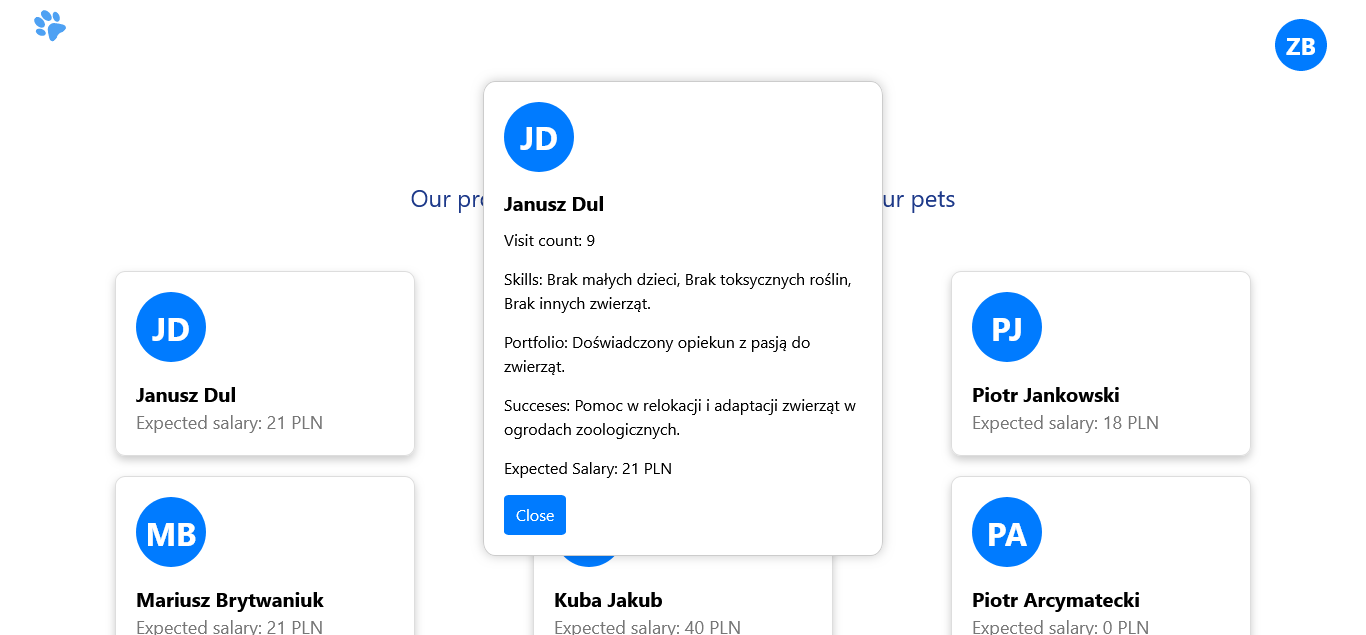
#### Widok endpointów, w Swagger UI.



## Perspektywa rekrutera

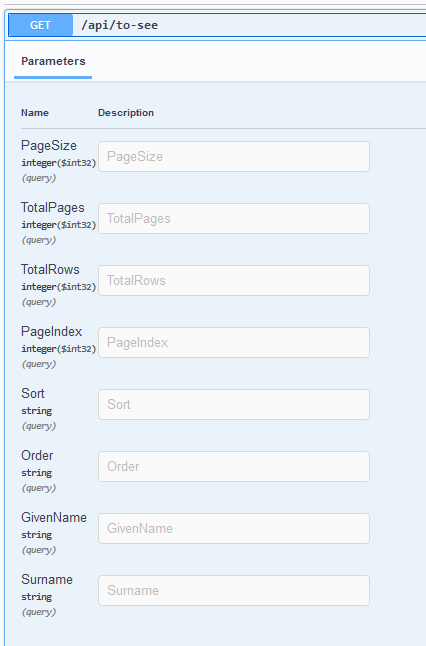
Frontend dla rekrutera (właściciela) umożliwia przeglądanie informacji publicznych zawartych w profilach pracowników. Po kliknięciu na kartę danego pracownika, widoczne są jego podstawowe informacje, a także dodatkowo wyświetlany jest VisitCount - czyli licznik odwiedzin tego profilu przez innych użytkowników w aplikacji. Ta funkcja dostarcza rekruterowi przydatnych statystyk, umożliwiając śledzenie zainteresowania danym pracownikiem w społeczności Animal Caretakers.

#### Przykłady wyświetlenia informacji o pracowniku.



## Zapisane profile

Endpoint umożliwiający pobranie zapisanych profili pracowników przez rekrutera. Dostępny wyłącznie dla rekrutera.



#### Parametry Zapytania

pagesize (opcjonalny): Liczba wyników na stronę.

totalpages (opcjonalny): Całkowita liczba stron.

totalrows (opcjonalny): Całkowita liczba wyników.

pageindex (opcjonalny): Numer bieżącej strony.

sort (opcjonalny): Pole do sortowania wyników.

order (opcjonalny): Kierunek sortowania (ASC lub DESC).

#### Odpowiedzi

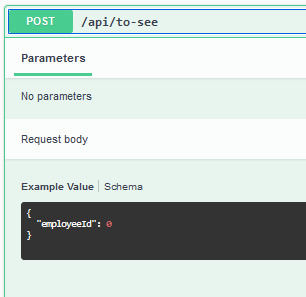
Kod 200 (Sukces): Zapisane profile zostały pomyślnie pobrane.

Endpoint POST /api/to-see umożliwia rekruterowi zapisanie profilu pracownika do obejrzenia.

#### Odpowiedzi

Kod 200 (Sukces): Profil pracownika został pomyślnie zapisany.

Kod 400 (Bad Request): Nieprawidłowe żądanie.

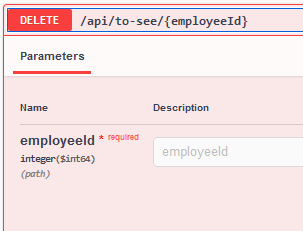


Endpoint umożliwia rekruterowi usunięcie zapisanego profilu pracownika do obejrzenia.

#### Odpowiedzi

Kod 200 (Sukces): Profil pracownika został pomyślnie usunięty z listy.

Kod 404 (Not Found): Nie znaleziono zapisanego profilu o podanym employeeId.



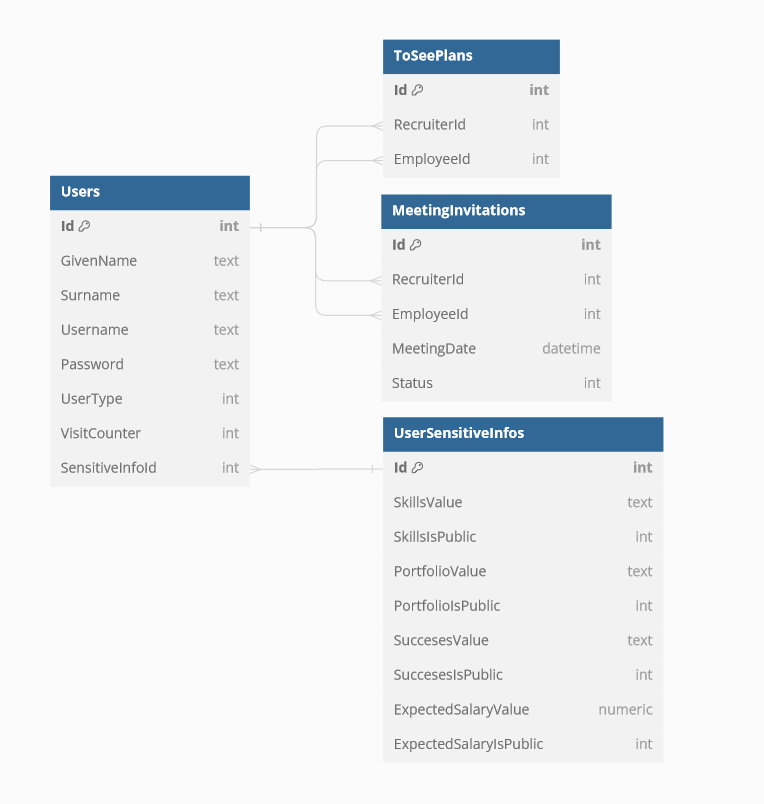
#### Przykładowe Zapytanie

DELETE /api/to-see/123

# Realizacja projektu:

## Widok endpointów w Swagger UI

## Diagram bazy danych



**User (Użytkownik)** - Tabela przechowuje informacje o użytkownikach aplikacji AnimalCaretakers. Każdy użytkownik może pełnić rolę opiekuna zwierząt lub właściciela zwierzęcia poszukującego opiekuna. Składa się z następujących atrybutów: identyfikator użytkownika (UserID), nazwa użytkownika (UserName), adres e-mail (UserEmail), typ użytkownika (UserType), a także licznik wyświetleń profilu użytkownika w aplikacji (VisitCounter). Klucz zewnętrzny SensitiveInfoID wskazuje na ID w tabeli UserSensitiveInfo, co pozwala na bezpieczne zarządzanie wrażliwymi informacjami o użytkowniku.

**ToSeePlan (Plan Wizyty)** - Tabela przechowuje informacje dotyczące planów wizyt, które opiekunowie zwierząt i pracodawcy mogą tworzyć. Posiada atrybuty takie jak: identyfikator planu (ToSeePlanID), ID użytkownika wysyłającego zaproszenie (SenderUserID), ID użytkownika odbierającego zaproszenie (ReceiverUserID), data wizyty (Date), oraz status planu (Status), określający czy wizyta jest zaplanowana, zrealizowana czy anulowana.

**MeetingInvitations (Zaproszenie na Spotkanie)** – Tabela zawiera informacje dotyczące zaproszeń na spotkania skierowanych do opiekunów zwierząt. Posiada atrybuty takie jak identyfikator zaproszenia (MeetingInvitationID), opis pracy (JobDescription), oraz oferta wynagrodzenia (SalaryOffered). Klucz obcy do tabeli MeetingInvitation wskazuje na ID w tabeli User, umożliwiając powiązanie zaproszenia z konkretnym użytkownikiem.

**UserSensitiveInfo (Wrażliwe Informacje o Użytkowniku)** - Tabela przechowuje poufne informacje o użytkowniku, takie jak dane kontaktowe, adres zamieszkania itp. Składa się z następujących atrybutów: identyfikator informacji (SensitiveInfoID), klucz obcy do tabeli User (UserID), numer kontaktowy (ContactNumber), oraz adres zamieszkania (Address).

## Database seed - przykładowe konta

Podczas inicjalizacji aplikacji tworzone są przykładowe konta. Hasło do wszystkich przedstawionych poniżej kont to "12345".

### Loginy kont opiekunów (employees) zwierząt:

* janusz
* ela
* piotr
* mariusz
* kuba

### Loginy kont właścicieli (recruiters) zwierząt:

* zofia
* mateusz
* ola
* janka
* marysia

## Abstrakcja walidacji

W pierwszym etapie projektu skoncentrowano się na integralności danych poprzez wprowadzenie walidacji dla wszystkich modeli. Abstrakcja walidacji zapewnia, że informacje zapisywane w bazie danych są poprawne.

## Rejestracja i logowanie użytkowników

Wprowadzono abstrakcję obsługującą rejestrację użytkowników, oraz logowanie. Etap obejmuje procesy tworzenia konta, uwierzytelniania, oraz dostępu do zasobów zależnych od roli (pracownik, rekruter).

## CRUD dla pracodawców

Na tym etapie dołożono starań aby zaimplementować operacje CRUD (Create, Read, Update, Delete) dla pracodawców. Funkcja umożliwia skuteczne zarządzanie danymi w systemie.

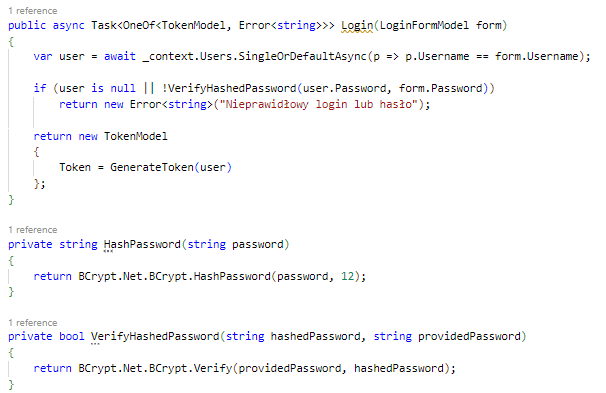
## Komponenty karty i nawigacji

Kolejnym krokiem była implementacja odmiennego stylu nawigacji oraz dodanie komponentu karty (card), który wyświetla informacje w sposób zorganizowany i atrakcyjny wizualnie.

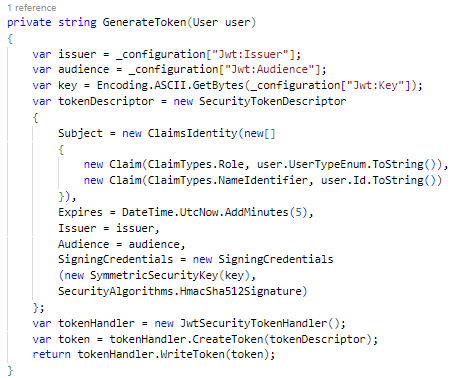
## Pobieranie użytkowników z filtrami i paginacją

Wprowadzono nowy endpoint API umożliwiający pobieranie danych użytkowników z zastosowaniem filtrów i paginacji. Funkcjonalność jest istotna dla skutecznego przeglądania użytkowników.

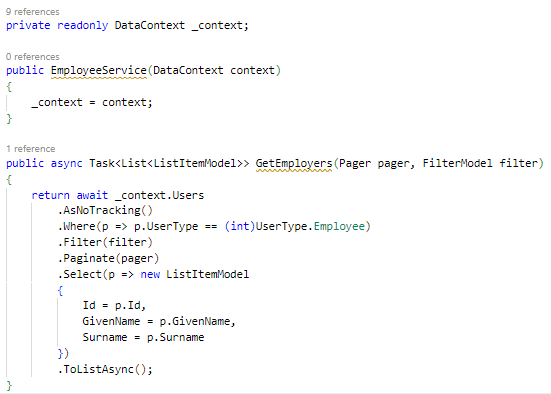
## Fragmenty kodu źródłowego



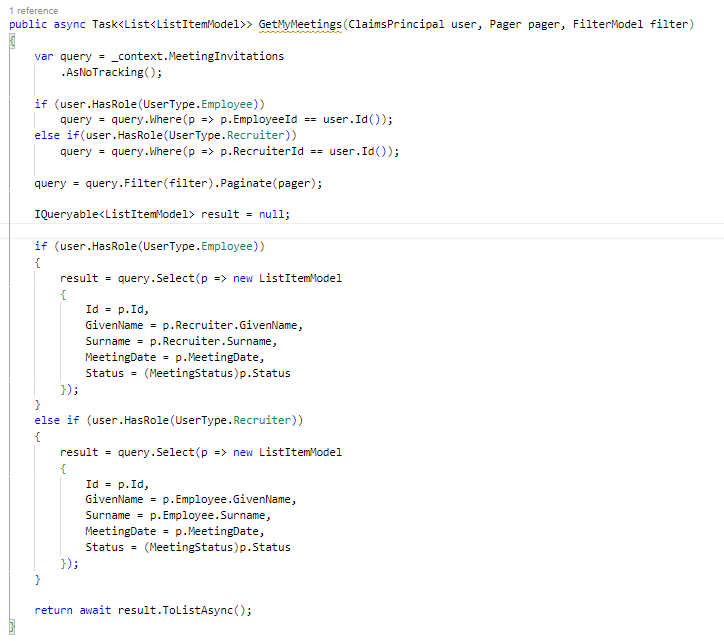
Powyższy fragment kodu przedstawia metodę Login w klasie AuthService odpowiada za uwierzytelnianie użytkownika na podstawie podanych danych logowania. Sprawdza istnienie użytkownika o podanym loginie, a następnie porównuje hasło za pomocą funkcji VerifyHashedPassword. W przypadku nieprawidłowych danych, zwraca błąd. Metoda HashPassword służy do weryfikacji hasła przy logowaniu. Wykorzystuje bibliotekę BCrypt.Net do porównania dostarczonego hasła z już zahashowanym w bazie danych.



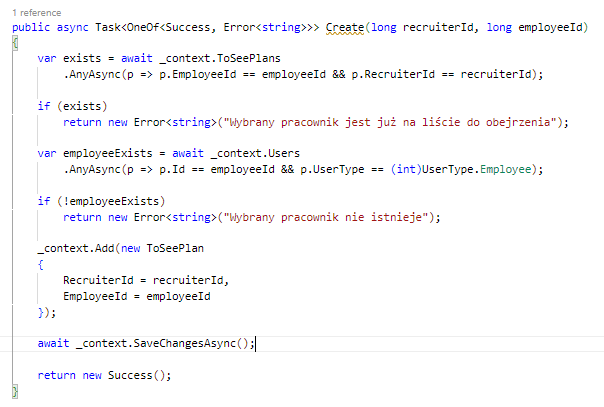
Kolejna metoda GenerateToken w klasie AuthService generuje token JWT (JSON Web Token) na podstawie konfiguracji, zawierający klucz kodujący zdefiniowany w pliku konfiguracyjnym. Token jest tworzony za pomocą obiektu SecurityTokenDescriptor, a następnie zamieniany na ciąg znaków.



Metoda GetEmployers w klasie EmployeeService pobiera listę pracowników z bazy danych, uwzględniając filtrowanie i paginację. Wykorzystuje metodę Filter do uwzględnienia warunków filtrowania oraz Paginate do obsługi paginacji. Następnie konstruuje obiekty ListItemModel na podstawie wyników zapytania.



Metoda GetMyMeetings w klasie MeetingsService pobiera spotkania z bazy danych dla zalogowanego użytkownika (zarówno pracownika, jak i rekrutera). W zależności od roli użytkownika, stosuje odpowiednie warunki do zapytania, a następnie filtruje i paginuje wyniki. Konstruuje obiekty ListItemModel reprezentujące spotkania.



Metoda Create w klasie ToSeePlanService służy do dodawania pracownika do listy do obejrzenia przez rekrutera. Sprawdza, czy taki plan już istnieje, czy wybrany pracownik istnieje, a następnie dodaje nowy plan do bazy danych. Zwraca odpowiednią informację o sukcesie lub błędzie.

# Podsumowanie i perspektywy rozwoju

## Potencjalne możliwości rozwoju projektu

* Rozszerzenie profili pracowników - dodanie dodatkowych informacji i sekcji do profili pracowników, takich jak opinie klientów czy zdjęcia z opieką nad zwierzętami.
* System oceny i recenzji - implementacja systemu ocen i recenzji, umożliwiającego klientom wystawianie opinii na temat opiekunów zwierząt.
* Powiadomienia i kalendarz - dodanie powiadomień dla opiekunów o nowych zaproszeniach na spotkanie. Integracja kalendarza, ułatwiającego śledzenie planów opiekunów.
* Rozwinięcie funkcjonalności rekruterów - dodanie narzędzi wspomagających rekruterów w procesie selekcji, takich jak lista ulubionych profili czy notatki.

# Wnioski i spostrzeżenia

Projekt AnimalCaretakers spełnia założenia aplikacji do łączenia opiekunów zwierząt z potencjalnymi pracodawcami.

Projekt dostarcza solidne narzędzie do skutecznego zarządzania informacjami w branży opieki nad zwierzętami. Dane związane z opieką nad zwierzętami mogą przynieść korzyści w zrozumieniu preferencji klientów, identyfikowaniu trendów w opiece nad zwierzętami, oraz ułatwić proces rekrutacji.

Aplikacja jest ułatwieniem również dla osób prywatnych oferujących usługi opieki nad zwierzętami, a także dla klientów poszukujących zaufanych opiekunów.

Biblioteka haszowania haseł: <https://github.com/BcryptNet/bcrypt.net>

Repozytorium projektu: <https://github.com/pprzemek56/animals_caretakers>