

Spis treści

1. Zakres i cel projektu	3
2. Zespół	3
3. Założenia funkcjonalności	3
4. Baza danych	3
5. Zastosowana technologia	5
5.1. Baza danych	5
5.2. Spring Boot	5
5.3. JPA, Hibernate	6
5.4. Bootstrap	6
5.5. Thymeleaf	6
5.6. Maven	6
5.7. Plik <i>pom.xml</i>	6
5.8. Klasy DAO	6
5.9. Kontrolery	6
5.10. Plik <i>SecurityConfiguration</i>	7
5.11. Plik <i>application.properties</i>	7
5.12. Testy JUnit	7
5.13. Folder <i>Templates</i>	7
5.14. Folder <i>Errors</i>	7
6. Opis realizacji	7
7. Opis instalacji produktu	7
8. Działanie aplikacji	8
8.1. Strona startowa	8
8.2. Perspektywa Zwykłego użytkownika	9
8.3. Perspektywa Księgowego	11
8.4. Perspektywa Pracownika	12
8.5. Perspektywa Administratora	13
8.6. Dodatkowe funkcje	14
8.7. Obsługa błędów	15
9. Podsumowanie	16



**Wydział Elektroniki
i Technik Informacyjnych**

POLITECHNIKA WARSZAWSKA

1. Zakres i cel projektu

Celem naszego projektu było stworzenie aplikacji współpracującej z zaprojektowaną bazą danych. Tematem projektu jest oceanarium. Aplikacja została zbudowana przy użyciu frameworka Spring Boot w języku Java. Do realizacji projektu wykorzystaliśmy następujące narzędzia:

- **IntelliJ IDEA** – zaawansowane środowisko programistyczne, które wspomagało nas podczas pisania kodu
- **Maven** – narzędzie do zarządzania zależnościami i automatyzacją budowania projektu
- **Oracle Database** – system zarządzania bazą danych, który przechowuje wszystkie dane dotyczące oceanarium
- **SQL Developer** – narzędzie do zarządzania i interakcją z bazą danych Oracle

2. Zespół

- **Igor Czunikin-Krasowicki** - Kierownik projektu / Baza danych
- **Jakub Strzelczyk** - Frontend
- **Jakub Grzechnik** - Frontend
- **Adam Januszewski** - Backend
- **Mateusz Gajdowicz** - Backend

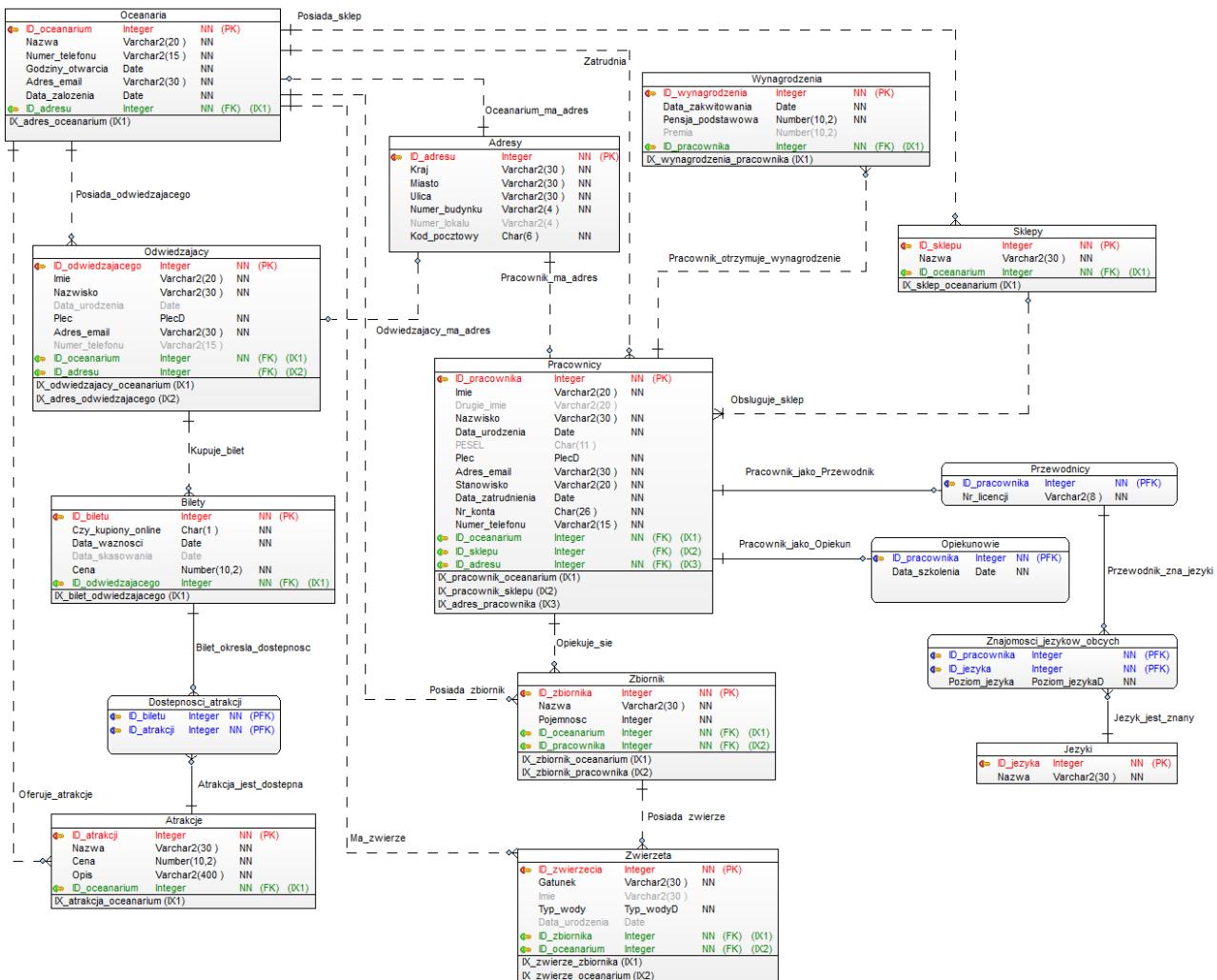
3. Założenia funkcjonalności

- Obsługa kont dla wielu użytkowników i kilku perspektyw (np. Klient, pracownik, szef)
- Automatyczne wylogowywanie przy nieaktywności
- Obsługa błędów
- Możliwość zakupienia oraz modyfikacji biletu do oceanarium
- Możliwość edytowania danych przechowywanych w bazie takie jak pracownicy lub ich wynagrodzenia

4. Baza danych

Stworzyliśmy relacyjną bazę danych dla oceanarium, opartą na technologii Oracle.

Poniżej przedstawiamy model logiczny utworzonej bazy danych.



Rysunek 1. Model logiczny

W bazie danych oceanarium gromadzone są dane dotyczące różnych aspektów działalności oceanarium, w tym danych o pracownikach, odwiedzających, biletach, atrakcjach, wynagrodzeniach oraz zwierzętach. Struktura bazy danych jest zorganizowana w następujący sposób:

- **Oceanarium:** Przechowuje podstawowe informacje o oceanarium, takie jak nazwa, numer telefonu, godziny otwarcia, adres e-mail, data założenia oraz adres.
- **Adresy:** Zawierają informacje o ulicy, mieście, numerze budynku, numerze lokalu oraz kodzie pocztowym. Adresy są powiązane z odwiedzającymi, pracownikami i oceanarium.
- **Odwiedzający:** Przechowują dane takie jak imię, nazwisko, data urodzenia, numer telefonu, adres e-mail, płeć oraz powiązanie z adresem.
- **Bilety:** Zawierają informacje o tym, czy bilet został kupiony online, dacie ważności, cenie, zniżkach oraz powiązanie z odwiedzającymi i atrakcjami.
- **Dostępność atrakcji:** Określa dostępność różnych atrakcji w zależności od biletów.
- **Atrakcje:** Przechowują informacje o dostępnych atrakcjach, takich jak nazwa, cena, opis oraz powiązanie z oceanarium.
- **Pracownicy:** Zawierają dane takie jak imię, nazwisko, data urodzenia, adres e-mail, numer telefonu, płeć, numer konta bankowego oraz powiązanie z oceanarium, adresami i sklepami.
- **Wynagrodzenia:** Przechowują informacje o dacie zatrudnienia, pensji podstawowej, premii oraz powiązanie z pracownikami.
- **Sklepy:** Zawierają dane o sklepach powiązanych z oceanarium, w tym nazwę i adres.
- **Przewodnicy:** Przechowują dane o numerze licencji oraz powiązanie z pracownikami i językami.
- **Języki:** Zawierają informacje o językach znanych przez przewodników.
- **Opiekunowie:** Przechowują dane o dacie szkolenia oraz powiązanie z pracownikami.
- **Zbiorniki:** Zawierają informacje o zbiornikach wodnych w oceanarium, takie jak nazwa, typ ekosystemu, rodzaj wody oraz powiązanie z oceanarium i pracownikami.
- **Zwierzęta:** Przechowują dane o gatunku, liczbie, nazwie łacińskiej, typie wody oraz powiązanie z oceanarium i zbiornikami.

5. Zastosowana technologia

5.1. Baza danych

Baza danych to dostępny dla wielu użytkowników zbiór logicznie powiązanych danych. Jest tworzona w celu zaspokojenia potrzeb informacyjnych klientów. Dzięki systemowi zarządzania bazą danych (SZBD) istnieje możliwość zarządzania, definiowania, tworzenia i utrzymywania bazy danych.

Relacyjna baza danych to rodzaj bazy danych, która opiera się na modelu relacyjnym. Model ten zakłada, że dane są przechowywane w postaci dwuwymiarowych tabel, a relacje między tabelami są reprezentowane za pomocą kluczy obcych. Nasza aplikacja korzysta z relacyjnej bazy danych zaprojektowanej przez nas, a połączenie z nią jest skonfigurowane w pliku application.properties.

5.2. Spring Boot

Spring Boot to framework oparty na platformie Spring, który umożliwia szybkie i efektywne tworzenie zaawansowanych aplikacji Java. Dostarcza on wbudowany serwer aplikacyjny, co eliminuje konieczność manualnej konfiguracji serwera. Aplikacje napisane w Spring Boot zawierają wszystkie niezbędne zależności oraz komponenty, co umożliwia szybkie i proste uruchomienie aplikacji.

Spring Boot oferuje automatyczną konfigurację, co oznacza, że wiele ustawień i konfiguracji jest domyślnie obsługiwanych przez framework. Domyślne ustawienia pozwalają na uruchomienie zasadniczej aplikacji bez potrzeby ręcznej konfiguracji. Dodatkowo, istnieje możliwość dostosowania konfiguracji do indywidualnych potrzeb, co jest opcjonalne.

Dzięki prostocie i wydajności Spring Boot, proces developmentu staje się szybszy i bardziej efektywny. Framework ten dostarcza gotowych szablonów, ułatwiając skupienie się na innych aspektach, zamiast na konfiguracji i ustawieniach. Szybkość tworzenia aplikacji przekłada się na krótszy czas dostarczania nowych funkcji i zmniejsza koszty rozwoju.

5.3. JPA, Hibernate

JPA (Java Persistence API) to specyfikacja w języku Java umożliwiająca mapowanie obiektów Java na struktury baz danych relacyjnych. Hibernate jest popularnym frameworkiem implementującym JPA, który ułatwia zarządzanie danymi w bazach relacyjnych. JPA definiuje encje jako obiekty reprezentujące dane, a Hibernate dostarcza funkcji ORM (Object-Relational Mapping), obsługuje transakcje, oraz umożliwia korzystanie z JPQL (Java Persistence Query Language) do operacji na danych. Wspólnie, JPA i Hibernate stanowią efektywne narzędzie dla pracy z bazami danych w aplikacjach Java, eliminując konieczność bezpośredniego korzystania z zapytań SQL.

5.4. Bootstrap

Bootstrap to otwarta biblioteka projektowania stron internetowych (CSS, JavaScript), stworzona przez zespół Twittera. Zapewnia gotowe komponenty interfejsu użytkownika, takie jak przyciski, nawigacja, formularze, oraz wiele innych, co ułatwia projektowanie responsywnych i estetycznych stron internetowych. Bootstrap jest oparty na technologii CSS, a także wykorzystuje komponenty JavaScript, co pozwala na łatwe tworzenie interaktywnych elementów interfejsu.

5.5. Thymeleaf

Thymeleaf to silnik szablonów przeznaczony dla warstwy widoku w aplikacjach opartych na Java. Pozwala na łatwe integrowanie dynamicznych treści z kodem Java w plikach HTML. Thymeleaf umożliwia wstawianie danych, warunkowe renderowanie elementów.

5.6. Maven

Maven odegrał kluczową rolę w procesie zarządzania projektem. Dzięki niemu ustandaryzowaliśmy budowę projektu, zarządzanie zależnościami oraz automatyzację wielu procesów związanych z developmentem. Plik konfiguracyjny pom.xml stał się centralnym punktem zarządzania projektem, zawierając informacje o strukturze projektu, zależnościach, pluginach i innych kluczowych ustawieniach. W efekcie uzyskaliśmy jednolity i uporządkowany sposób zarządzania projektem w kontekście technologii Java.

5.7. Plik *pom.xml*

Plik pom.xml jest używany w projektach Maven dla aplikacji napisanych w języku Java. Plik ten zawiera informacje dotyczące konfiguracji, zależności, wtyczek oraz innych ustawień używanych w aplikacji. W naszym projekcie plik pom.xml definiuje wersję narzędzia Spring Boot jako 2.6.1 oraz zawiera zależności, takie jak **spring-boot-starter-thymeleaf** i **spring-boot-starter-web**. Plik pom.xml zawiera również informacje o wersji Javy (używamy wersji 17), framework'u Bootstrap oraz bibliotece Lombok.

5.8. Klasy DAO

Każda klasa odpowiadająca poszczególnym encjom z bazy danych posiada również klasę DAO (Data Access Object). Klasy te zawierają podstawowe funkcje CRUD (Create, Read, Update, Delete) oraz inne metody, które umożliwiają uzyskanie specyficznych danych z odpowiednich tabel.

5.9. Kontrolery

Kontrolery są kluczowym elementem aplikacji, ponieważ obsługują żądania przychodzące od klientów. Zarządzają akcjami, które mają być wykonane w odpowiedzi na te żądania. Kontrolery zawierają metody odpowiadające na żądania HTTP, takie jak **@RequestMapping**.

5.10. Plik *SecurityConfiguration*

Plik *SecurityConfiguration* zabezpiecza aplikację webową poprzez definiowanie zasad dostępu do danych. W naszej aplikacji zdefiniowano cztery poziomy dostępu, które różnią się uprawnieniami do określonych części aplikacji:

- **User** - może kupować i zarządzać kupionymi biletami,
- **Accountant** - ma dostęp do edytowania wynagrodzeń wszystkich pracowników,
- **Employee** - ma dostęp do swoich danych osobistych oraz wynagrodzenia,
- **Admin** - ma dostęp do wszystkich tabel.

5.11. Plik *application.properties*

Plik *application.properties* jest jednym z plików konfiguracyjnych używanych w projektach na framework Spring Boot. Służy do konfiguracji logowania oraz innych ustawień dotyczących bazy danych.

5.12. Testy JUnit

W naszej aplikacji wykorzystujemy JUnit do przeprowadzania testów jednostkowych, które sprawdzają poprawność działania poszczególnych funkcjonalności.

5.13. Folder *Templates*

Folder *Templates* w naszym projekcie zawiera pliki szablonów, które służą do reprezentacji rekordów i danych z bazy danych w czytelnej i estetycznej formie, ułatwiając użytkownikom korzystanie z aplikacji.

5.14. Folder *Errors*

Folder *Errors* w naszej aplikacji zarządza obsługą różnych błędów, takich jak 403, 404 czy 500, które mogą zakłócać działanie strony. Zabezpieczenie tych błędów jest kluczowe dla zapewnienia prawidłowego użytkowania aplikacji.

6. Opis realizacji

Relizację zadania projektowego podzieliliśmy na kilka najbardziej istotnych części, dzieląc się zadaniami do wykonania po równo między wszystkich członków zespołu. Realizację projektu możemy podzielić na poszczególne etapy:

1. **Utworzenie bazy danych** - na samym początku za pomocą aplikacji ToadModeer utworzyliśmy bazę danych zawierającą informacje na temat atrakcji, pracowników i podobnych elementów naszego oceanarium
2. **Utworzenie strony internetowej** - następnie przeszliśmy do stworzenia strony internetowej obsługującą naszą bazę danych. Do wykonania tej części zadania wykorzystaliśmy takie narzędzia jak Spring Boot, Maven i Bootstrap.

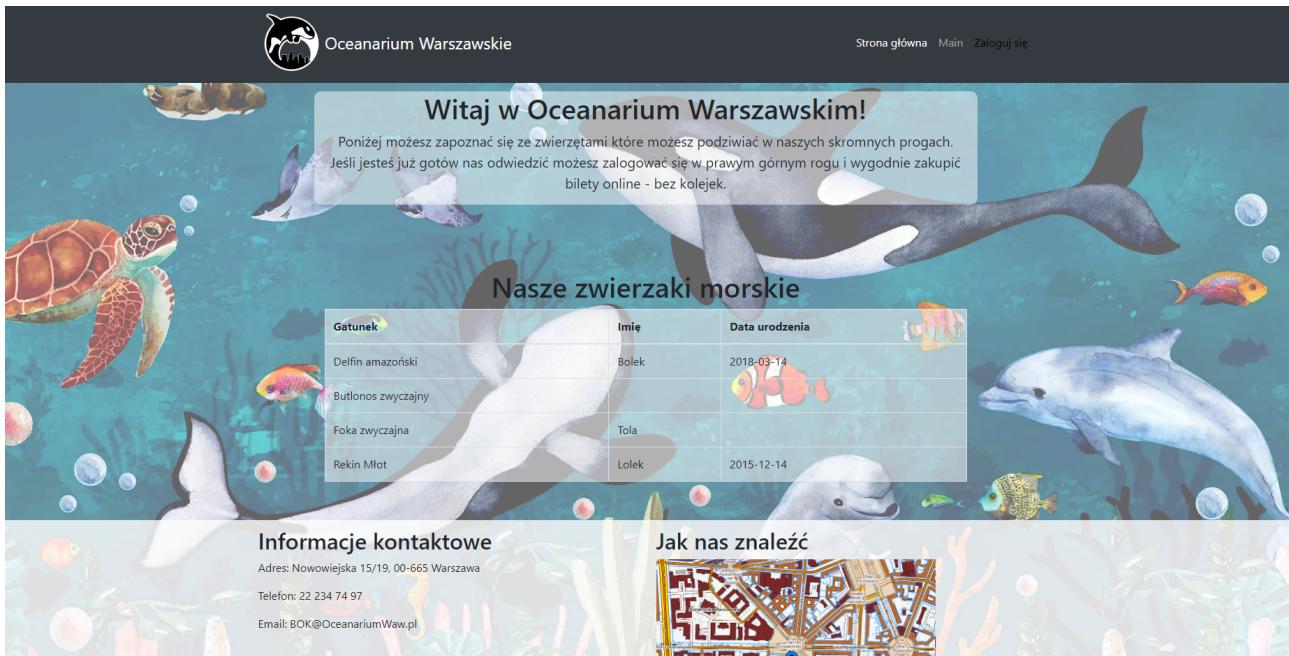
Utworzyliśmy klasy obsługujące poszczególne encje z naszego oceanarium dla różnych użytkowników, zaprogramowaliśmy wizualnie naszą stronę internetową oraz dodaliśmy dodatkowe funkcje takie jak mapa oraz rejestracja. Działanie naszej aplikacji zaprezentujemy w punktach poniżej.

7. Opis instalacji produktu

Aby zainstalować produkt, wystarczy sklonować repozytorium z platformy uczelnianej GitLab. Odpalenie produktu wymaga załadowanie projektu Maven, a następnie poprawnego połączenia z bazą danych. Podczas tworzenia naszej aplikacji wykorzystaliśmy z bazy danych hostowanej na komputerze kierownika projektu, z którą każdy mógł się połączyć dzięki dołączeniu do wspólnej sieci wirtualnej. Uruchomienie projektu przez Przewodzącego będzie możliwe poprzez dołączenie do tej samej sieci wirtualnej, z którą połączony będzie komputer, na którym znajduje się baza danych.

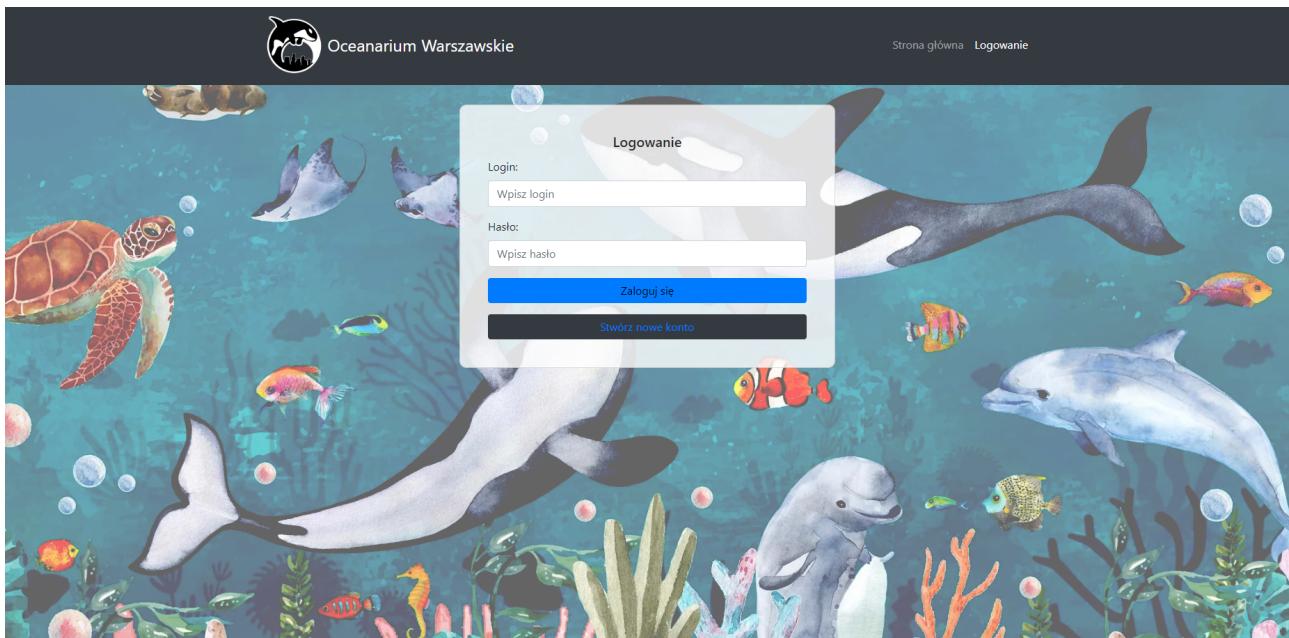
8. Działanie aplikacji

8.1. Strona startowa



Rysunek 2. Widok strony startowej

Na stronie startowej, pokazanej na rysunku 2. widzimy powitanie osoby odwiedzającej stronę, a także ukazuje się nam tabelka ze zwierzętami dostępnymi w naszym oceanarium. W tle widzimy zdjęcie nawiązujące do tematu strony. Na dole strony są dane kontaktowe do oceanarium, a także mała mapa wskazująca położenie naszego obiektu. W prawym górnym rogu jest dostępna jedyna funkcja z tej strony, którą jest możliwość logowania po wciśnięciu przycisku **Zaloguj się** w prawym górnym rogu strony. Reszta funkcji jest osiągalna po zalogowaniu.



Rysunek 3. Widok strony logowania

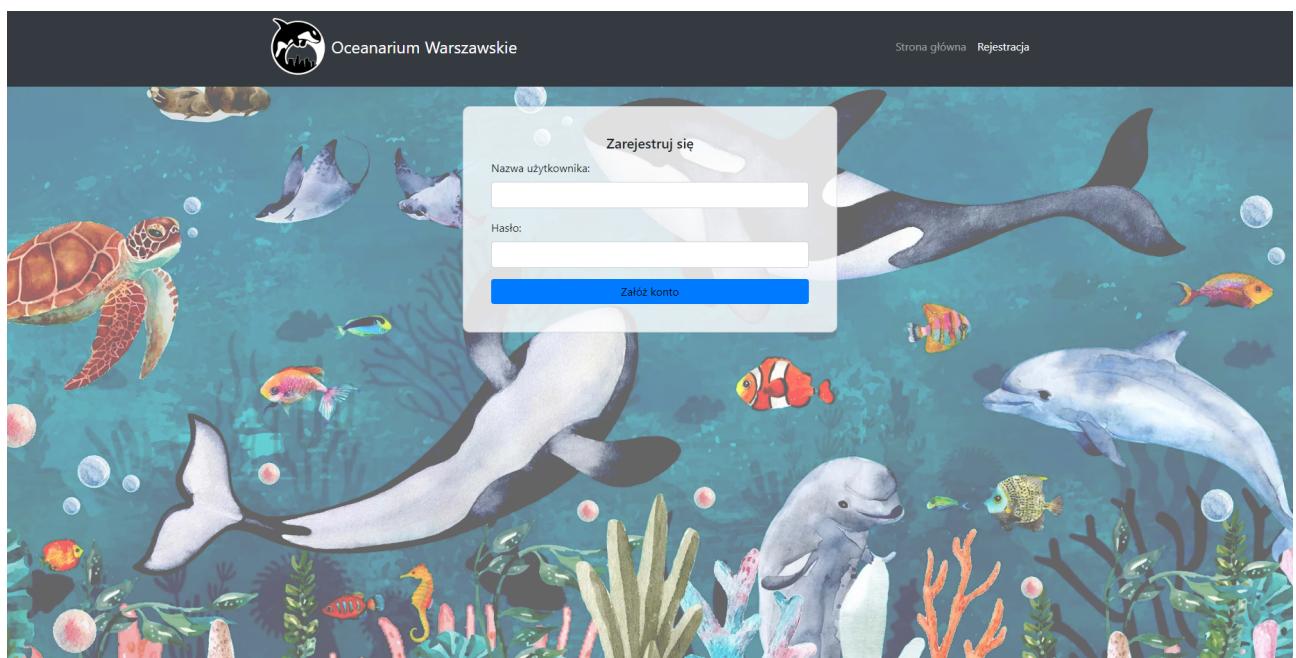
Po przekierowaniu na stronę logowania, możemy zalogować się jako użytkownik, który posiada jedną z wymienionych poniżej ról:

- **Zwykły użytkownik,**
- **Księgowy,**
- **Pracownik,**
- **Administrator.**

W zależności od danej roli użytkownicy posiadają różne uprawnienia. Zwykły użytkownik może kupić bilet i wyświetlić bilety, które posiada. Księgowy ma możliwość edytowania wynagrodzeń pracowników i może stworzyć nowe. Pracownik widzi swoje wynagrodzenia i może zobaczyć swoje dane. Natomiast administrator posiada wszystkie uprawnienia - może nawet zmieniać role użytkowników.

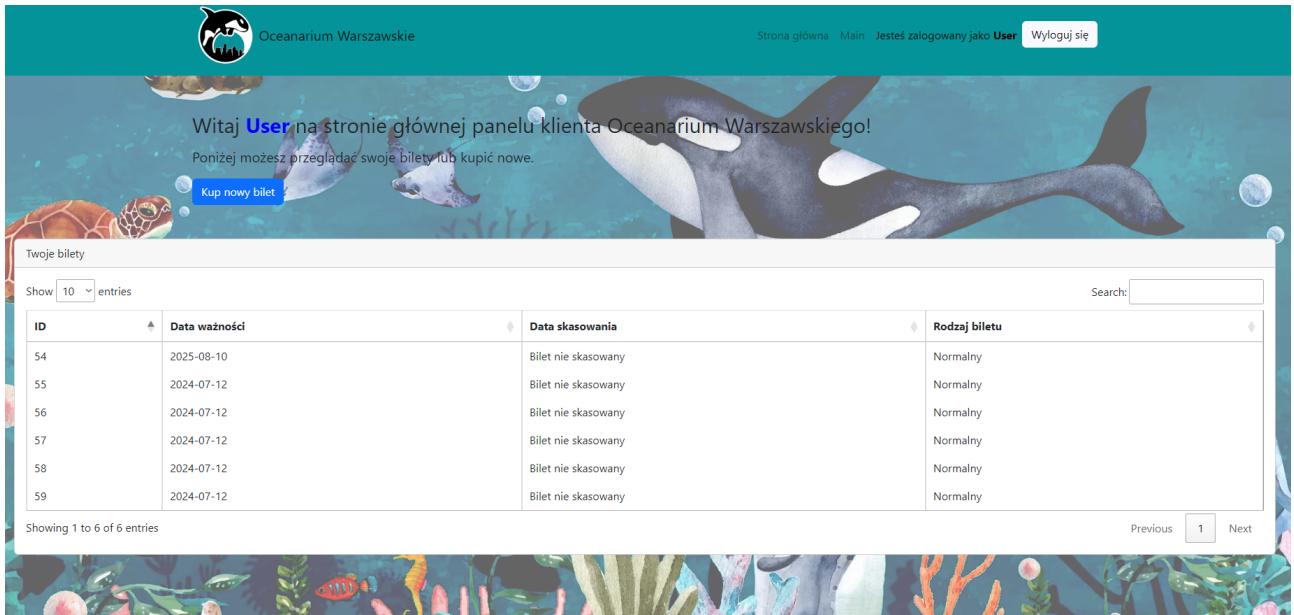
8.2. Perspektywa Zwykłego użytkownika

Jednym z kluczowych elementów naszej aplikacji jest możliwość tworzenia nowych kont użytkowników. Funkcjonalność ta pozwala nowym odwiedzającym zarejestrować się w systemie, co umożliwia im pełne korzystanie z dostępnych funkcji, takich jak zakup biletów czy zarządzanie swoimi rezerwacjami. Rejestracja nowego użytkownika odbywa się za pośrednictwem intuicyjnego formularza, który wymaga tylko podania nazwy oraz hasła którego będziemy używać. Na poniższym rysunku możemy zobaczyć widok formularza do rejestracji konta.



Rysunek 4. Widok strony rejestracji

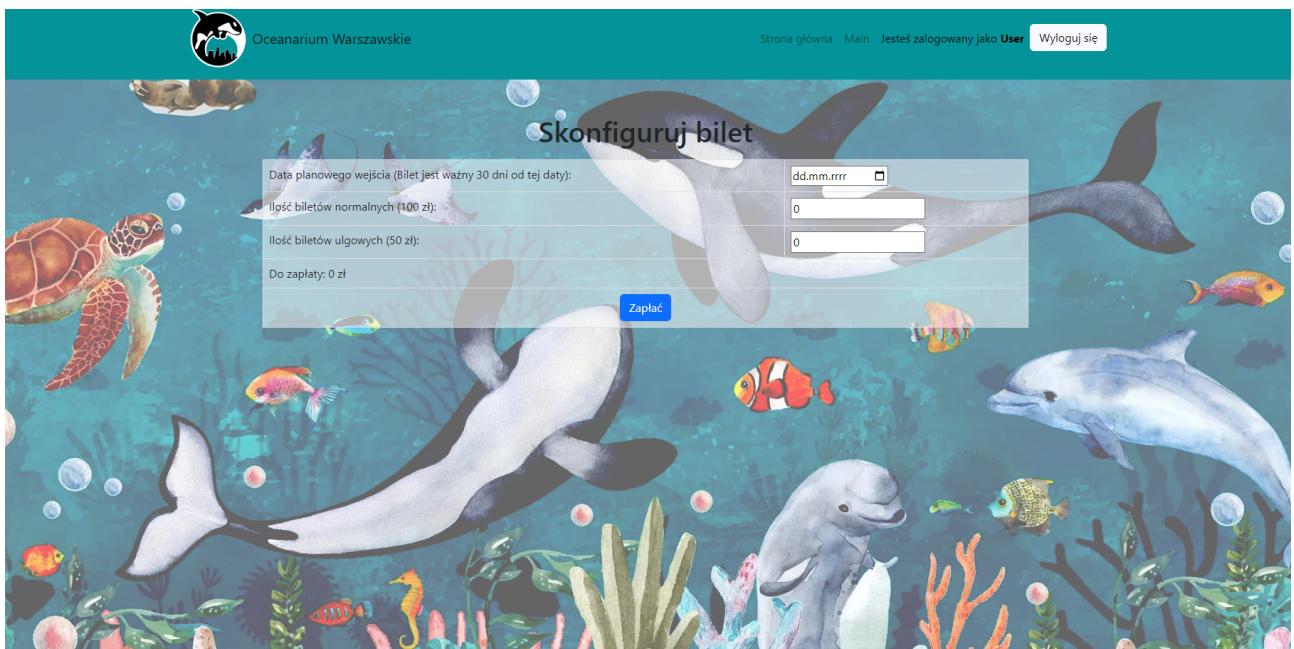
Widok po zalogowaniu przedstawiliśmy na rysunku 5.



Rysunek 5. Widok strony użytkownika

Widzimy na nim tabelę z kupionymi biletami, które posiadają datę ważności oraz informację jakiego typu jest to bilet (normalny czy ulgowy) oraz czy został skasowany. Użytkownik posiada również opcję zakupu biletów. Ten użytkownik posiada uprawnienia 'zwykłego użytkownika'.

Opcje kupowania biletu przedstawiliśmy na rysunku 6.



Rysunek 6. Widok strony, umożliwiającej zakup biletu

W tej opcji Zwykły użytkownik kupuje bilety, może ustalić datę planowanego wejścia, a także ilość kupionych biletów normalnych i ulogowych. Po wyborze wyświetli się odpowiednia cena do zapłaty. Numer ID zostanie automatycznie przypisany do każdego z zakupionych biletów.

8.3. Perspektywa Księgowego

Po zalogowaniu na perspektywę Księgowego (rysunek 7), ukaże się nam tabelka z wszystkimi wynagrodzeniami pracowników oceanarium. Księgowy ma możliwość edytowania i usuwania już wcześniej powstałych wynagrodzeń, a także może dodawać nowe.

The screenshot shows a dashboard for the Warsaw Aquarium's accounting system. At the top, there's a logo for 'Oceanarium Warszawskie' and a navigation bar with links for 'Strona główna', 'Main', 'Jesteś zalogowany jako Adminigor', and 'Wyloguj się'. A prominent message box says 'Witaj Adminigor na stronie głównej księgowości Oceanarium Warszawskiego!' and 'Poniżej możesz przeglądać, edytować, dodawać lub usuwać wynagrodzenia naszych pracowników.' Below this is a button 'Dodaj nowe wynagrodzenie'. The main content area features a table titled 'Wynagrodzenia' showing four entries:

ID	Data zakwitowania	Pensja	Premia	ID Pracownika	Action
9	2024-01-01	2222	1111	1	Edytuj Usuń
24	2023-09-06	5555	333	2	Edytuj Usuń
25	2023-11-08	1123	111	1	Edytuj Usuń
26	2023-11-15	200	500	1	Edytuj Usuń

At the bottom of the table, it says 'Showing 1 to 4 of 4 entries' and has navigation buttons for 'Previous', '1', and 'Next'.

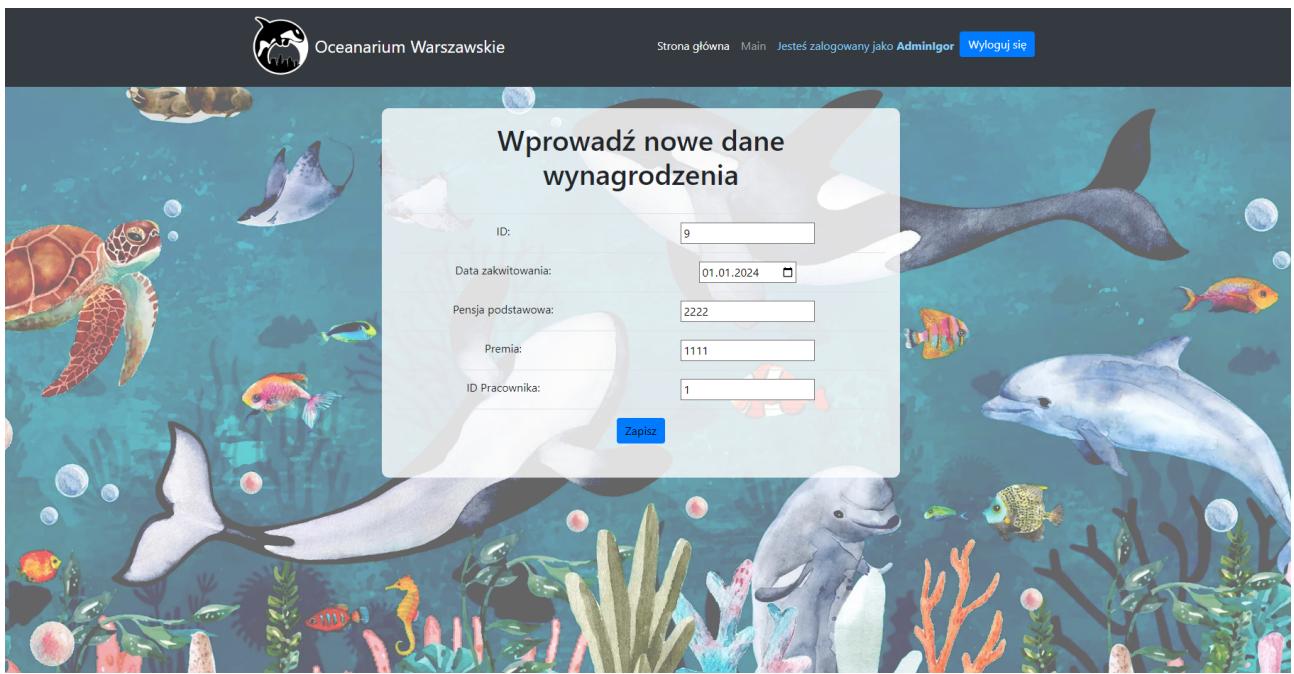
Rysunek 7. Widok strony księgowego

W opcji dodania nowego wynagrodzenia (rysunek 8), należy wybrać datę zakwitowania, ID pracownika dla którego chcemy przypisać to wynagrodzenie, czyli pensję podstawową i ewentualną premię. ID wynagrodzenia wygeneruje się automatycznie i zostanie przypisane do danego wynagrodzenia.

The screenshot shows a form for adding a new award ('Wprowadź dane nowego wynagrodzenia'). It includes fields for 'Data zakwitowania:' (date), 'Pensja podstawowa:' (base salary), 'Premia:' (bonus), and 'ID Pracownika:' (employee ID). There is also a 'Zapisz' (Save) button. The background features a vibrant underwater scene with various marine life like dolphins, turtles, and coral reefs.

Rysunek 8. Widok strony, umożliwiające dodanie nowego wynagrodzenia

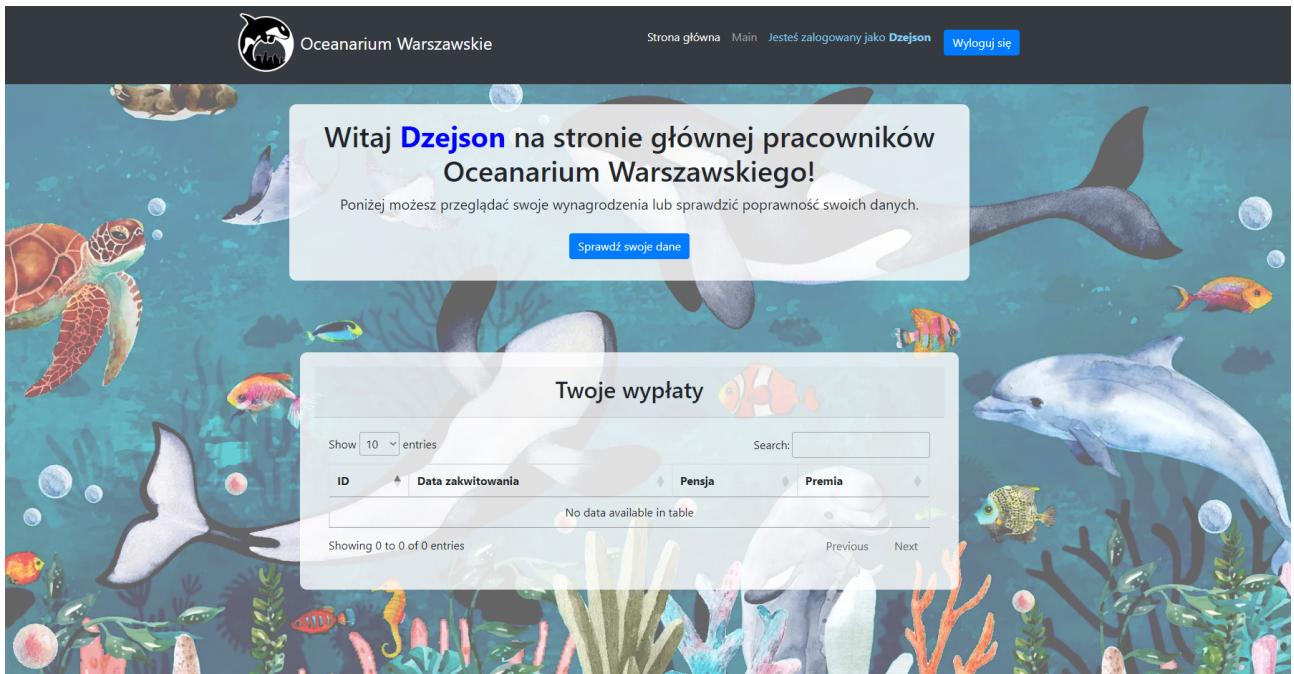
Mozemy również edytować już istniejące wygnagrodzenie co przedstawiliśmy na rysunku 9.



Rysunek 9. Widok strony, umożliwiające edycje wynagrodzenia

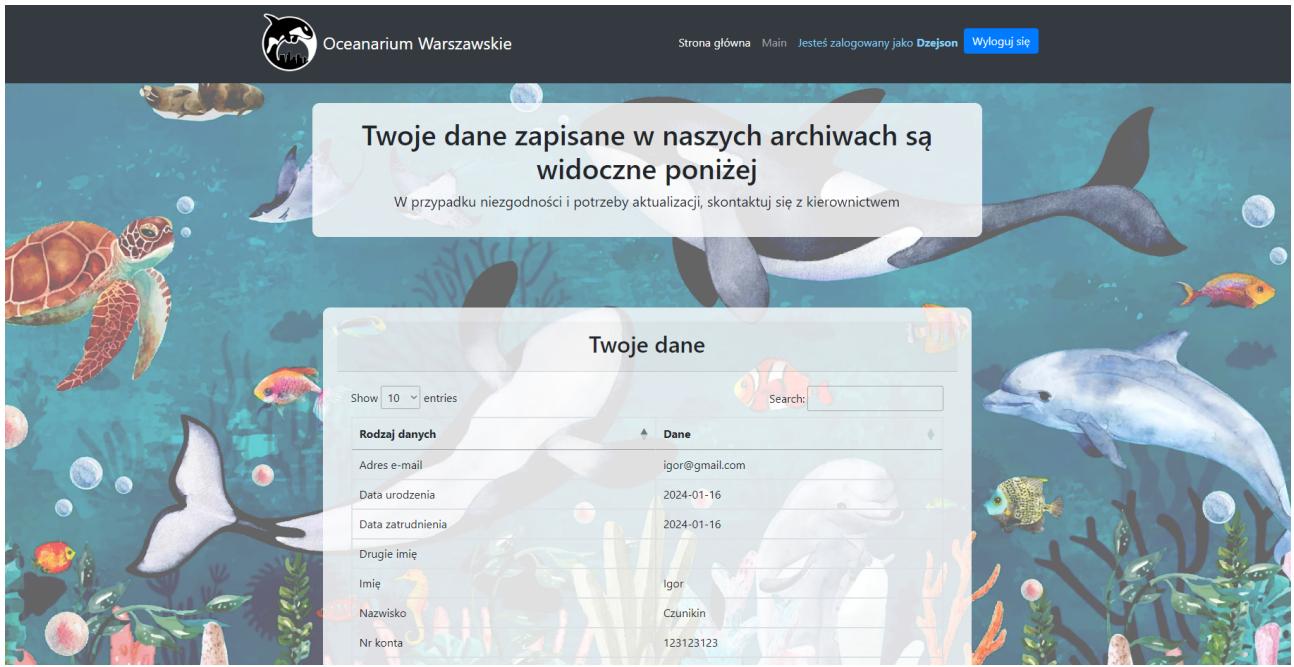
8.4. Perspektywa Pracownika

Po zalogowaniu na konto pracownika (rysunek 10), na stronie startowej pojawia się tabelka z wynagrodzeniami, a także przycisk umożliwiający sprawdzenie, czy dane osobiste pracownika są poprawne.



Rysunek 10. Widok strony Pracownika

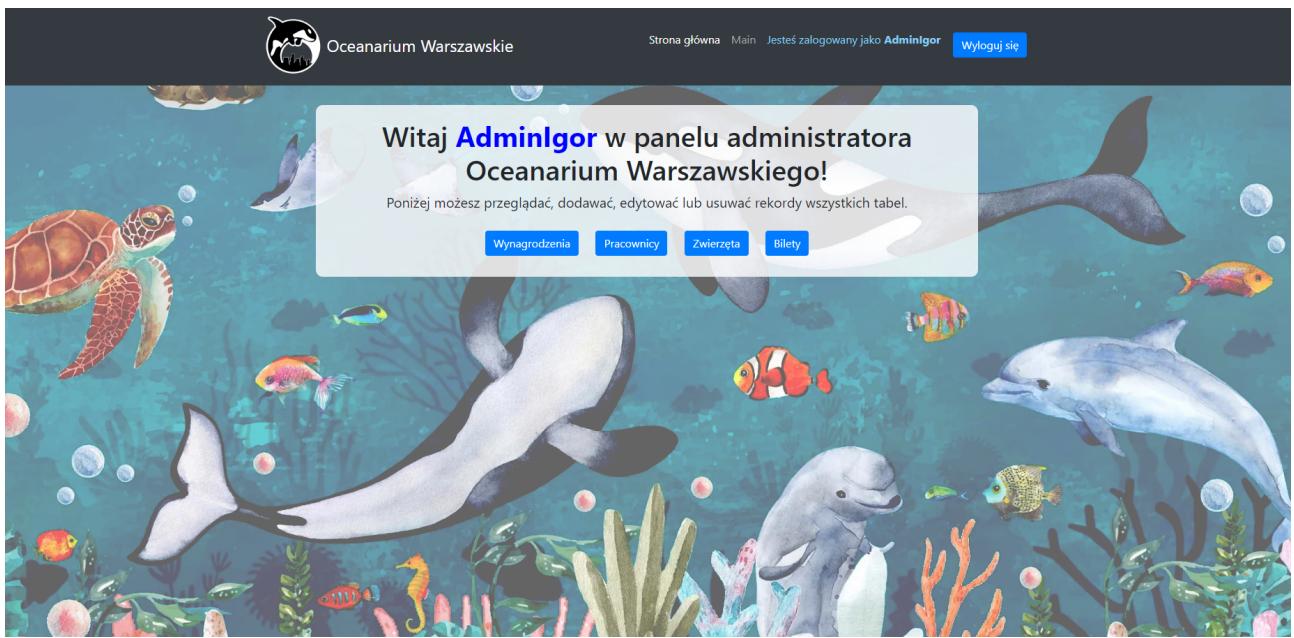
Na rysunku 11 przedstawiamy zapisane dane pracownika.



Rysunek 11. Strona z danymi pracownika

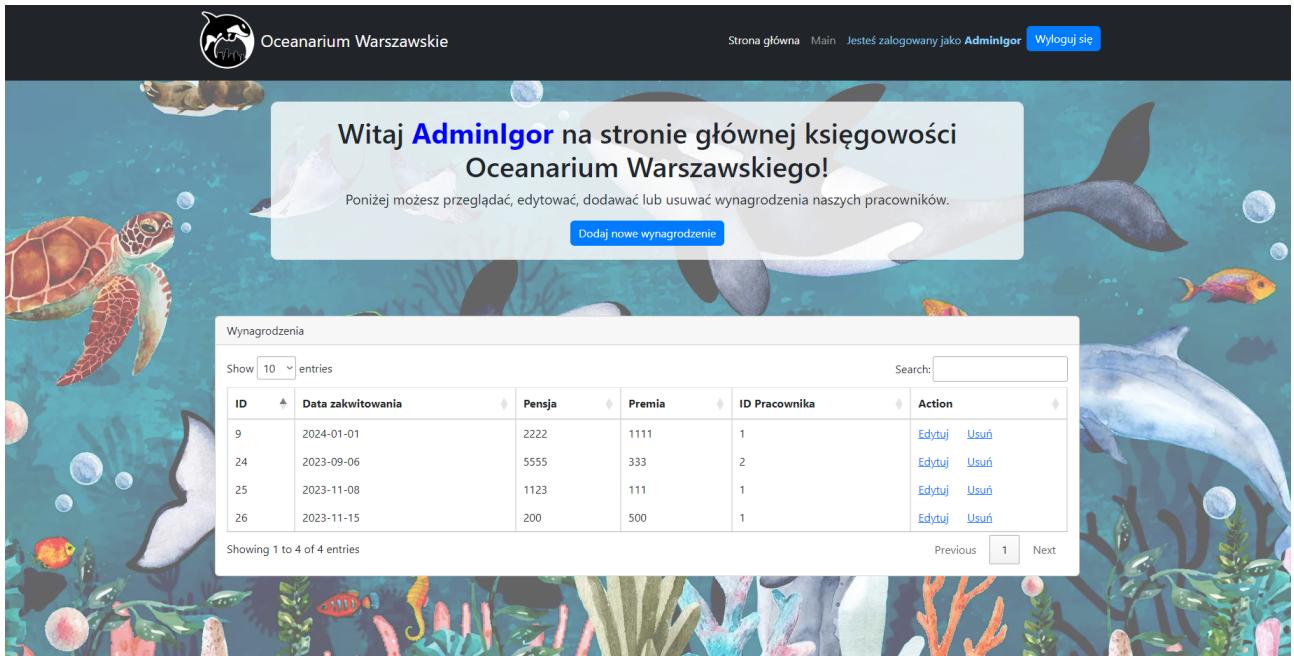
8.5. Perspektywa Administratora

Administrator posiada wszystkie wcześniej wymienione funkcje, może on dodawać edytować lub usuwać rekordy z tabeli Wynagrodzenia, Pracownicy, Zwierzęta i Bilety. Widok administratora przedstawiliśmy na rysunku 12.



Rysunek 12. Widok strony Administratora

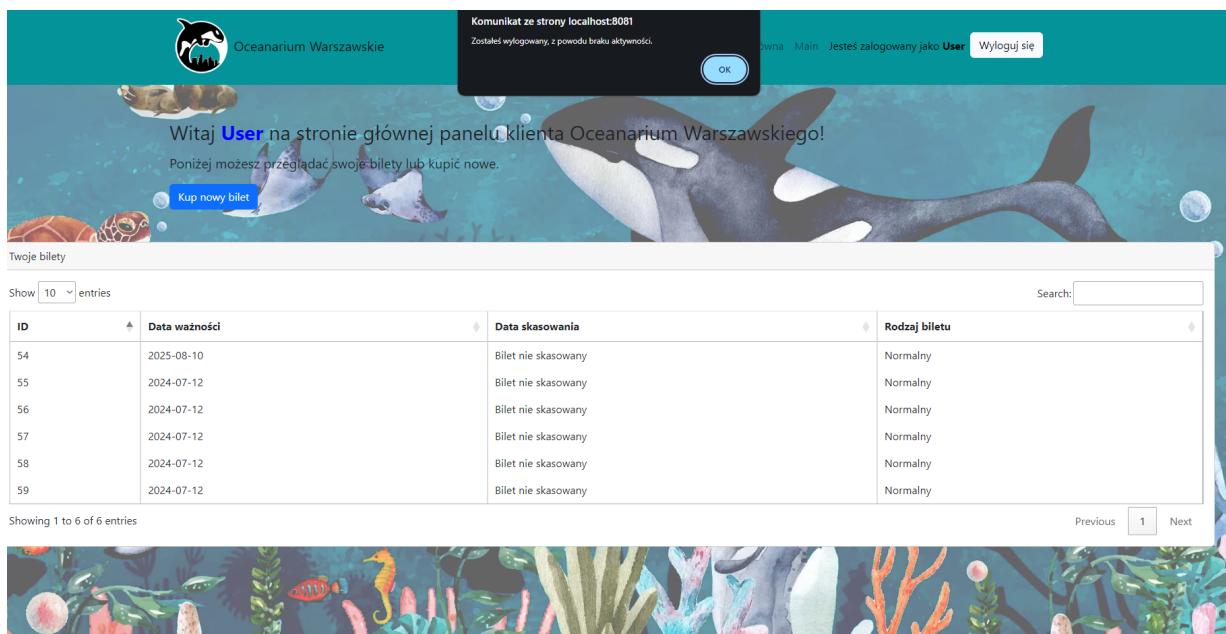
Przykładowa encja, którą obsługuje nasza aplikacja zawiera dane o wynagrodzeniach przydzielanych pracownikom oceanarium. Zawiera ona takie informacje jak ID pracownika, data zakwaterowania, pensja oraz premia.



Rysunek 13. Strona umożliwiająca podgląd danych o wynagrodzeniach

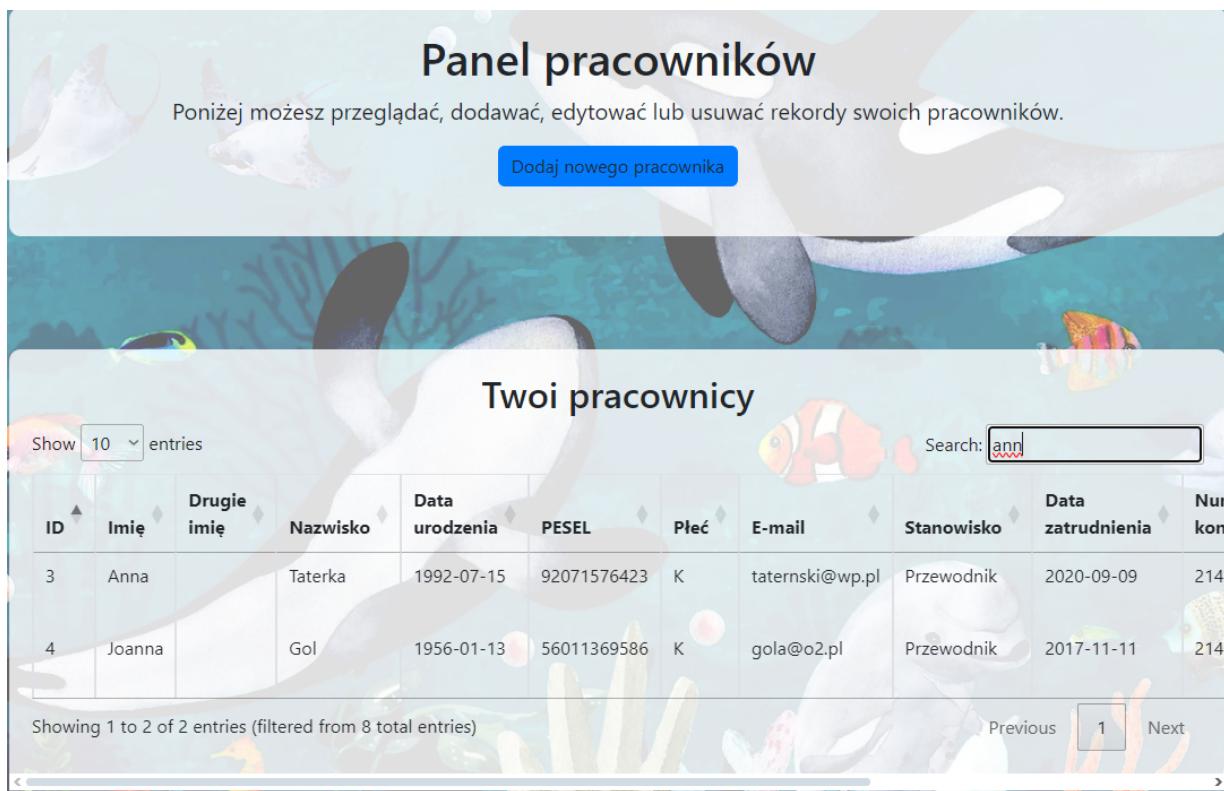
8.6. Dodatkowe funkcje

Nasza aplikacja posiada również kilka dodatkowych funkcji. Jedną z nich jest automatyczne wylogowywanie po 30s bez żadnej aktywności ze strony użytkownika (rysunek 14).



Rysunek 14. Okno z informacją o wylogowaniu użytkownika

Kolejnym, ułatwiającym obsługę aplikacji narzędziem, jest możliwość sortowania danych rekordów np. po numerze ID bądź wysokością wynagrodzenia, dzięki małym strzałkom, a także wyszukiwania danych w opcji Search (rysunek 15).



Panel pracowników

Poniżej możesz przeglądać, dodawać, edytować lub usuwać rekordy swoich pracowników.

[Dodaj nowego pracownika](#)

Twoi pracownicy

Show 10 entries

Search: ann

ID	Imię	Drugie imię	Nazwisko	Data urodzenia	PESEL	Płeć	E-mail	Stanowisko	Data zatrudnienia	Nr kon
3	Anna		Taterka	1992-07-15	92071576423	K	taternski@wp.pl	Przewodnik	2020-09-09	214
4	Joanna		Gol	1956-01-13	56011369586	K	gola@o2.pl	Przewodnik	2017-11-11	214

Showing 1 to 2 of 2 entries (filtered from 8 total entries)

Previous 1 Next

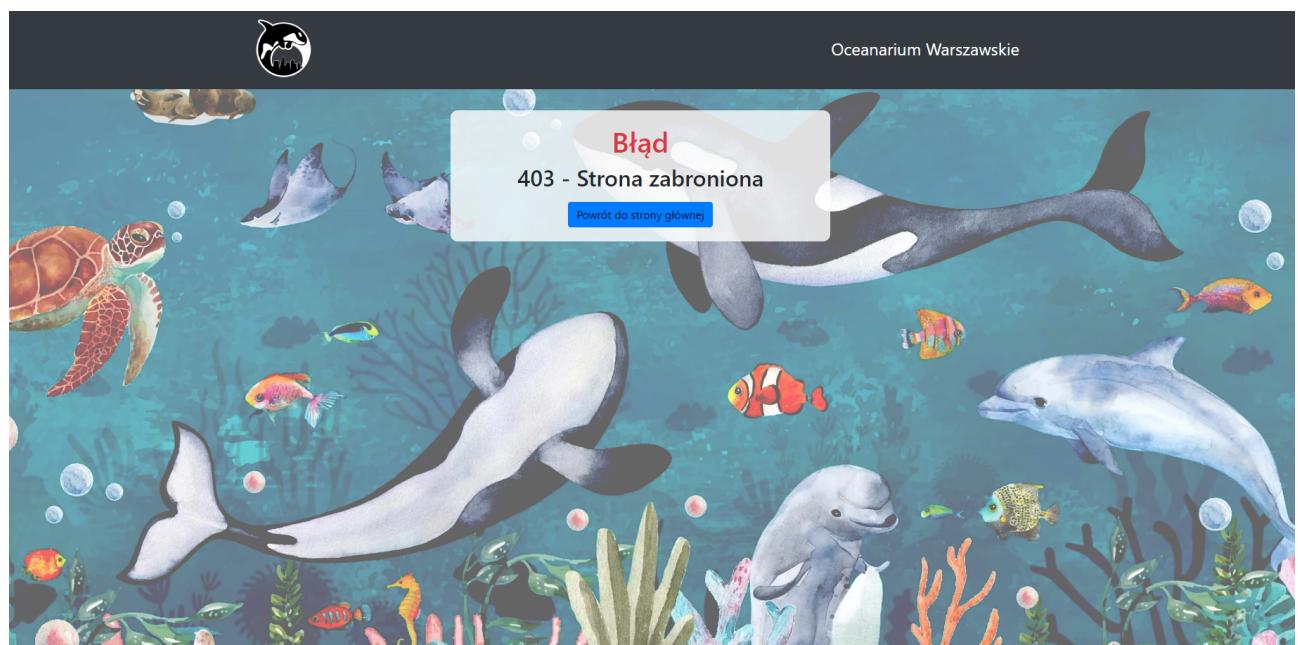
Rysunek 15. Opcja wyszukiwania oraz sortowania danych

8.7. Obsługa błędów

Nasza aplikacja zapewnia również obsługę błędów. W zależności od tego jaki błąd wystąpi, dostaniemy inne powiadomienie zawierające informację o niepowodzeniu.

Błąd 403

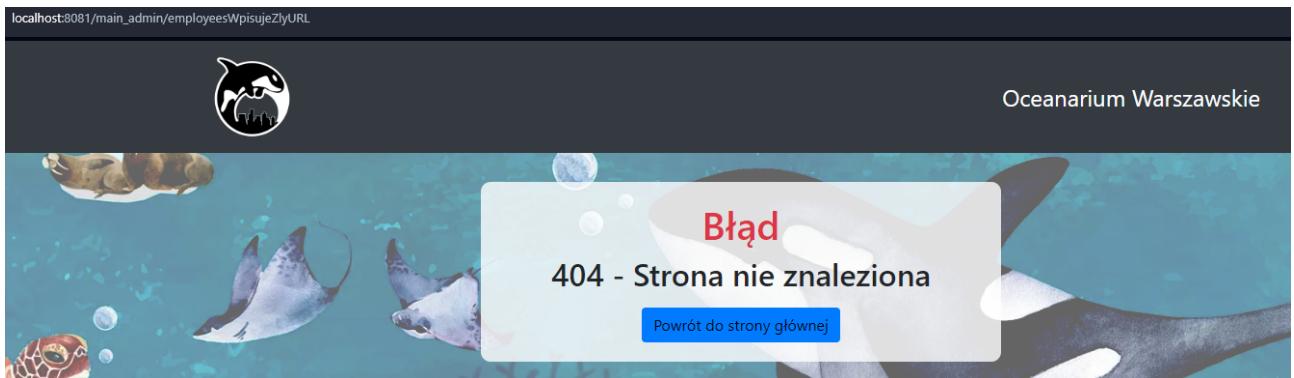
W przypadku próby przejścia do sekcji, która wykracza poza uprawnienia, dostaniemy komunikat 403 o zabronionym dostępie do strony. Komunikat ten możemy zobaczyć na rysunku 16.



Rysunek 16. Komunikat o zabronieniu dostępu do strony

Błąd 404

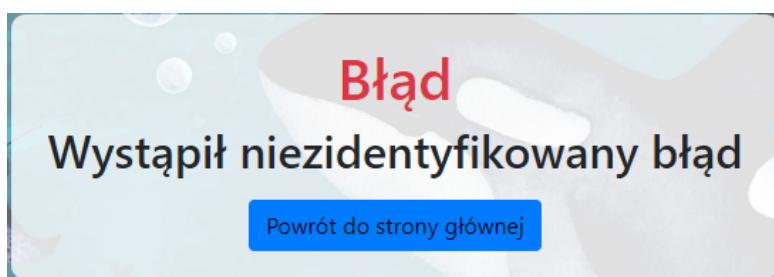
W przypadku podania niewłaściwego adresu URL dostaniemy błąd 404 komunikujący nas o nie znalezieniu strony. Komunikat możemy zobaczyć na rysunku 17.



Rysunek 17. Komunikat o nie znalezieniu strony

Błąd niezidentifikowany

W przypadku innego rodzaju błędu dostaniemy powiadomienie ogólne o wystąpieniu błędu. Komunikat ten możemy zobaczyć na rysunku 18.



Rysunek 18. Komunikat o niezidentyfikowanym błędzie

Błąd 500

Błąd 500 występuje przy błędzie wewnętrznym serwera.

Błąd 504

Błąd 504 występuje, gdy zostanie przekroczony limit czasu serwera.

9. Podsumowanie

System zarządza kontami użytkowników, pozwala na zakup biletów, administrowanie wynagrodzeniami pracowników oraz oferuje dodatkowe funkcje, takie jak automatyczne wylogowywanie. Baza danych, zaprojektowana w technologii Oracle, przechowuje szczegółowe informacje dotyczące oceanarium, jego pracowników, zwierząt, biletów oraz atrakcji.



**Wydział Elektroniki
i Technik Informacyjnych**

POLITECHNIKA WARSZAWSKA