

Projektowanie Aplikacji Internetowych(PAINT)

Projekt - Aplikacja webowa klubu piłkarskiego

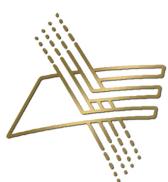
Autorzy projektu:

Kacper Karczmarek **325284**
Marcin Dudek **329066**
Mikołaj Górnny **325284**
Cezary Zięba **325284**
Mateusz Motrenko **325284**
Kacper Kuczyński **325284**

Opiekun projektu:

dr inż. Piotr Krzysztof Bobiński

10 czerwca 2024



**Wydział Elektroniki
i Technik Informacyjnych**

POLITECHNIKA WARSZAWSKA

Spis treści

1. Temat projektu	3
2. Zespół i opis ról w Projekcie	3
3. Wykorzystane technologie	3
3.1. Java Spring	3
3.2. CSS oraz Bootstrap	3
3.3. SQL Developer	4
4. Opis realizacji	4
4.1. Proces powstawania aplikacji	4
4.2. Schemat ER na poziomie konceptualnym	5
4.3. Schemat ER na poziomie modelu logicznego	6
5. Opis techniczny produktu	7
5.1. Java spring, a baza danych	7
5.1.1. Model: Pracownicy.java	7
5.1.2. Kontroler: PracownicyController.java	7
5.1.3. DAO: PracownicyDAO.java	7
5.1.4. Interfejsy między modułami	8
5.2. Główne kontrolery	8
5.2.1. LoginController	8
5.2.2. MyErrorController	8
5.2.3. SecurityConfiguration	9
6. Aplikacja - strona internetowa	10
6.1. Rozmieszczenie komponentów na stronie	10
6.2. Komunikaty ze strony i zabezpieczenia	11
6.3. Strony obsługujące błędy	12
7. Aplikacja - obsługa perspektyw	13
8. Aplikacja - operacje w ramach komunikacji z bazą danych	14
9. opis instalacji Produktu	17
9.1. Założenie osobistego konta w serwisie Oracle	17
9.2. Oprogramowanie bazy danych Oracle	17
9.3. Połączenie do lokalnej bazy danych Oracle	19
10.Uruchamianie strony internetowej i łączenie się z bazą danych	22
11.Podsumowanie i wnioski	25

1. Temat projektu

Tematem projektu było przygotowanie przez nasz zespół projektowy aplikacji współpracującej z wcześniej zaprojektowaną bazą danych SQL. Stworzyliśmy stronę internetową w języku HTML wykorzystującą arkusze stylów CSS. Umożliwiliśmy dostęp do baz danych na potrzeby aplikacji internetowej z wykorzystaniem framework'a JavaSpring. Nasza strona korzysta z Spring Security używanym do zapewnienia bezpieczeństwa serwisów internetowych. Nasza aplikacja jest stroną www klubu piłkarskiego, połączonego z bazą danych SQL.

2. Zespół i opis ról w Projekcie

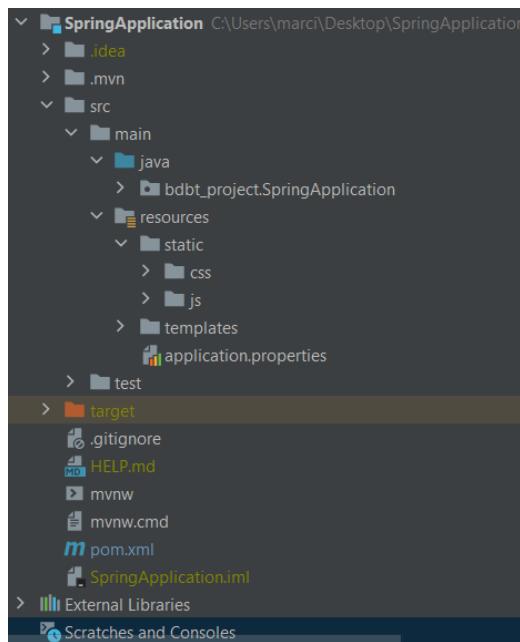
Zespół i opisanie ról członków:

- Mikołaj Górnny – kierownik projektu, tester rozwiązań
- Marcin Dudek – programista back-end
- Kacper Karczmarek – programista front-end
- Kacper Kuczyński – programista front-end
- Cezary Zięba – programista baz danych
- Mateusz Motrenko – programista back-end

3. Wykorzystane technologie

3.1. Java Spring

Do tworzenia strony internetowej wykorzystaliśmy framework Java Spring Boot wykorzystywany do tworzenia aplikacji w języku Java. Dzięki temu narzędziu proces tworzenia aplikacji był znacznie ułatwiony i szybszy, ponieważ Spring Boot dostarcza już gotowe rozwiązania i konwencje programistyczne.



Rysunek 1. Struktura utworzona z pomocą Spring Initializr

3.2. CSS oraz Bootstrap

Wykorzystaliśmy język CSS, który pomógł nam w projektowaniu naszej strony internetowej. CSS umożliwia separację struktury dokumentu(HTML) od jego prezentacji(stylów), a także definiowanie różne właściwości, takie jak kolor, czcionkę, odstępy, marginesy, czy układ elementów na stronie. Dzięki gotowym komponentom

frameworka Bootstrap, a także stylom, narzędziami oraz skryptom JavaScript, mogliśmy efektywnie zbudować estetyczną i funkcjonalną stronę internetową.

3.3. SQL Developer

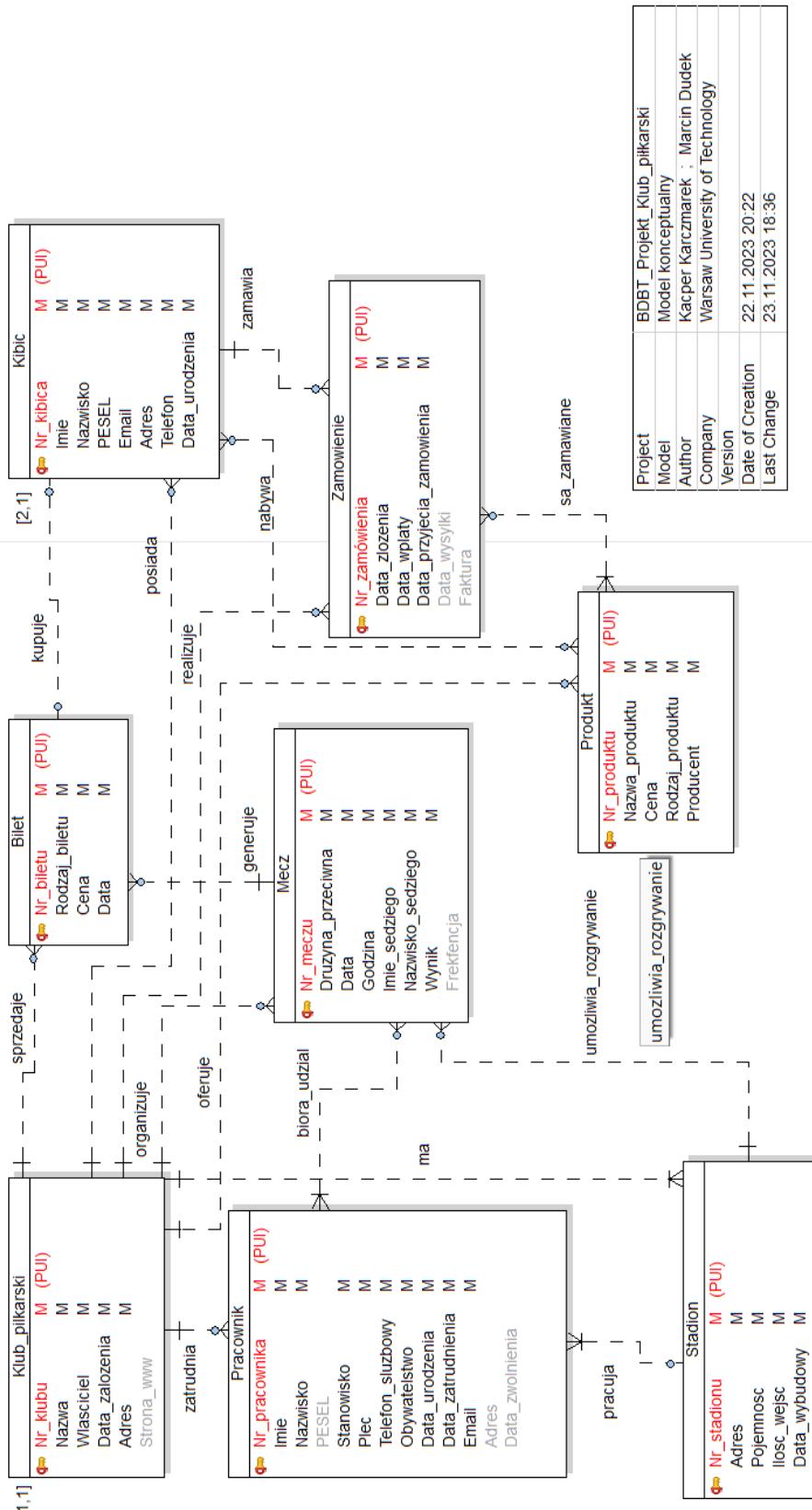
Do zarządzania bazami danych wykorzystaliśmy program SQL Developer, który umożliwia wykonywanie różnych operacji na bazach danych, takich jak tworzenie, modyfikacja i usuwanie tabel, indeksów, procedur składowanych czy widoków. Dzięki temu narzędziu mogliśmy efektywnie analizować dane, testować zapytania i monitorować wydajność baz danych.

4. Opis realizacji

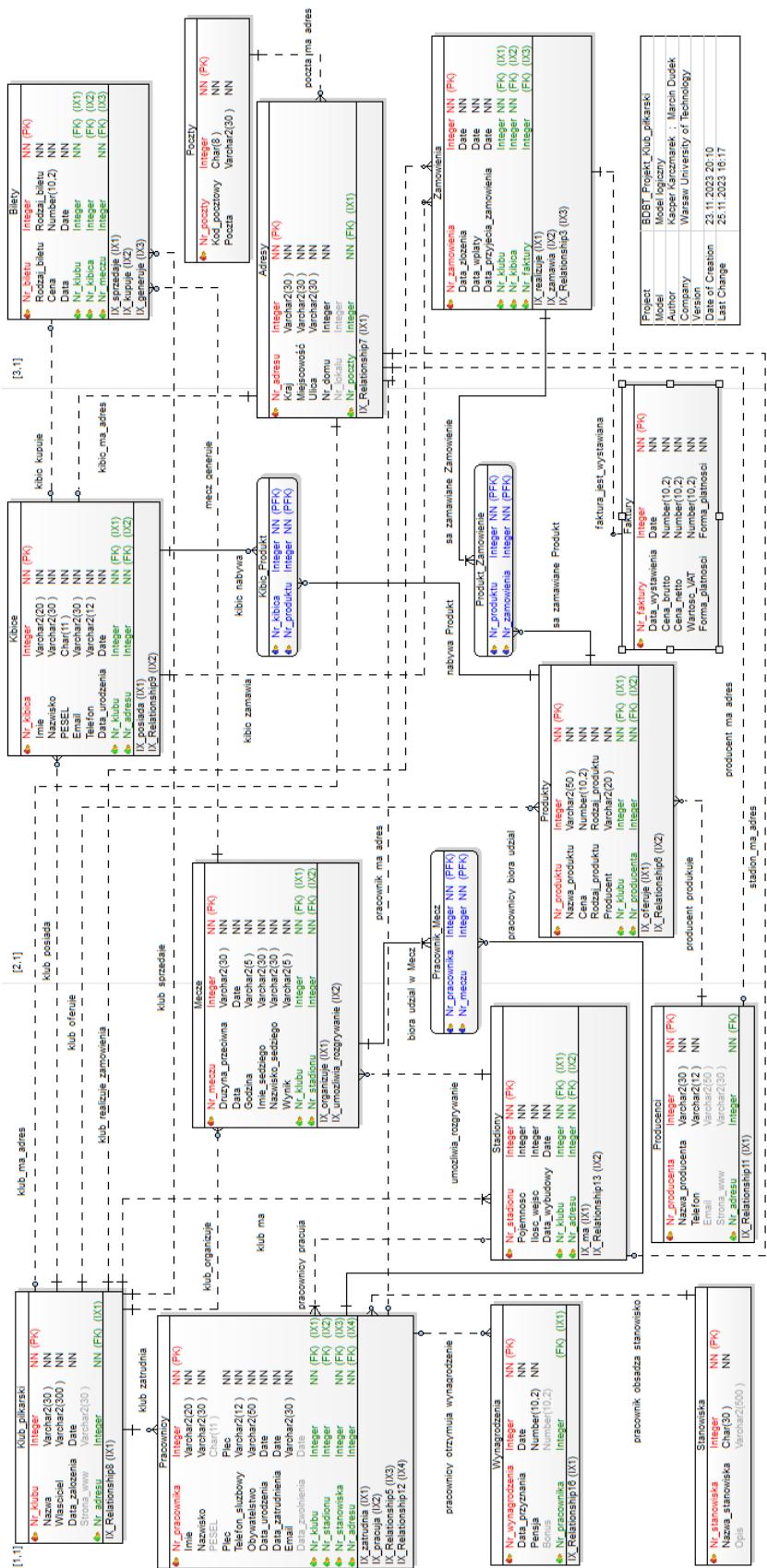
4.1. Proces powstawania aplikacji

Impusem do wymyślenia tematu projektu oraz funkcjonalności strony internetowej była wspólna pasja do piłki nożnej. Przeglądając strony internetowe największych klubów piłkarskich na świecie doszliśmy do wniosku, że brakuje im intuicyjnego interfejsu graficznego oraz prostoty w swoim wykonaniu. W ten sposób rozpoczęła się nasza praca nad projektem. Wyeksponowaliśmy mocne strony każdego z członków naszego zespołu oraz podzieliliśmy się zadaniami. Kluczowym etapem było stworzenie modelu koncepcualnego oraz logicznego w Toad Data Modeler. Na tej podstawie powstała baza danych w Oracle SQL Developer. Dużym wyzwaniem było określenie odpowiedniego typu danych, wybranie wymagań dostępności oraz następne przekonwertowanie modelu do bazy danych Oracle. Kolejnym etapem było zaprojektowanie samej strony internetowej oraz umożliwienie jej komunikacji z bazą danych. Zespół programistów front-end po komunikacji z resztą zespołu oraz kierownikiem projektu, starannie dobrą grafiki dostępne na stronie oraz zaplanował odpowiednie rozmieszczenie danych elementów każdego z panelów. Równolegle funkcjonował zespół programistów back-end, który zadbał o konkretne funkcjonalności strony internetowej, opisane w kolejnych rozdziałach. Finalnie otrzymaliśmy końcowy projekt.

4.2. Schemat ER na poziomie konceptualnym



4.3. Schemat ER na poziomie modelu logicznego



5. Opis techniczny produktu

5.1. Java spring, a baza danych

Ideeę oraz funkcjonalność wyjaśnimy na podstawie systemu zarządzania pracownikami. Każdy z systemów składa się z trzech głównych komponentów:

- Model: Klasa Pracownicy
- Kontroler: Klasa PracownicyController
- DAO (Data Access Object): Klasa PracownicyDAO

Każdy z tych komponentów pełni specyficzne funkcje i współpracuje z pozostałymi, aby zapewnić pełną funkcjonalność zarządzania pracownikami w aplikacji Spring Boot.

5.1.1. Model: Pracownicy.java

Struktura i atrybuty

Klasa Pracownicy reprezentuje tabelę Pracownicy w bazie danych. Zawiera ona następujące pola:

- Nr_pracownika (int)
- Imie (String)
- Nazwisko (String)
- PESEL (int)
- Płeć (String)
- Telefon_sluzbowy (int)
- Obywatelstwo (String)
- Data_urodzenia (Date)
- Data_zatrudnienia (Date)
- Email (String)
- Data_zwolnienia (Date)
- Nr_klubu (int)
- Nr_stadionu (int)
- Nr_stanowiska (int)
- Nr_adresu (int)

Konstruktor:

- Konstruktor z wszystkimi argumentami do inicjalizacji obiektu Pracownicy.
- Konstruktor domyślny bez argumentów.

Metody:

- Metody akcesorów i mutatorów (getters i setters): Pozwalają na dostęp i modyfikację prywatnych pól klasy.
- Metody formatowania dat: getFormattedDataUrodzenia(), getFormattedDataZatrudnienia(), getFormattedDataZwolnienia() - formatują daty na yyyy/MM/dd.

5.1.2. Kontroler: PracownicyController.java

Klasa PracownicyController jest kontrolerem Spring MVC, który obsługuje żądania HTTP dotyczące operacji na pracownikach.

Metody:

- viewHomePage(Model model): Wyświetla stronę główną z listą pracowników.
- showNowyPracownik(Model model): Wyświetla formularz dodawania nowego pracownika.
- save(@ModelAttribute("pracownicy") Pracownicy pracownik): Zapisuje nowego pracownika w bazie danych.
- showEditForm(@PathVariable(name = "Nr_pracownika") int Nr_pracownika, Model model): Wyświetla formularz edycji pracownika.
- update(@ModelAttribute("pracownicy") Pracownicy pracownik): Aktualizuje dane pracownika w bazie danych.
- delete(@PathVariable(name = "Nr_pracownika") int Nr_pracownika): Usuwa pracownika z bazy danych.

5.1.3. DAO: PracownicyDAO.java

Klasa PracownicyDAO obsługuje bezpośrednie operacje na bazie danych dla tabeli Pracownicy za pomocą JdbcTemplate.

Metody:

- list(): Zwraca listę wszystkich pracowników.
- save(Pracownicy pracownicy): Wstawia nowego pracownika do bazy danych.
- get(int Nr_pracownika): Zwraca pracownika na podstawie numeru Nr_pracownika.
- update(Pracownicy pracownicy): Aktualizuje dane pracownika w bazie danych.
- delete(int Nr_pracownika): Usuwa pracownika na podstawie numeru Nr_pracownika.

Szczegóły implementacyjne

- JdbcTemplate: Używane do wykonywania operacji na bazie danych.
- SimpleJdbcInsert: Używane do wstawiania nowych rekordów.
- BeanPropertyRowMapper: Używane do mapowania wyników zapytań na obiekty Pracownicy.

5.1.4. Interfejsy między modułami

PracownicyController i PracownicyDAO

- Kontroler PracownicyController używa PracownicyDAO do wykonywania operacji CRUD (Create, Read, Update, Delete) na bazie danych.
- Za pomocą adnotacji @Autowired, instancja PracownicyDAO jest wstrzykiwana do kontrolera PracownicyController.

PracownicyDAO i JdbcTemplate

- PracownicyDAO korzysta z JdbcTemplate do interakcji z bazą danych, co upraszcza wykonywanie zapytań SQL i operacji na bazie danych.

5.2. Główne kontrolery

5.2.1. LoginController

Konstrukcja: LoginController pełni rolę konfiguratora widoków oraz kontrolera, który obsługuje logikę przekierowania użytkowników po zalogowaniu. Jest to klasa oznaczona adnotacją @Configuration, co oznacza, że jest używana do definiowania konfiguracji Springa. Wewnątrz tej klasy zagnieżdżona jest również klasa DashboardController, która zawiera logikę przekierowania użytkowników na odpowiednie strony w zależności od ich roli.

Interfejs do innych modułów:

- ViewControllerRegistry: Służy do rejestrowania widoków, co upraszcza mapowanie adresów URL na odpowiednie widoki bez konieczności pisania osobnych metod kontrolera.
- HttpServletRequest: Wykorzystywane do sprawdzania roli użytkownika w metodzie defaultAfterLogin.

Funkcjonalności:

- Dodawanie widoków do rejestru widoków (ViewControllerRegistry).
- Przekierowanie użytkowników na odpowiednie strony po zalogowaniu (main_admin lub main_user).
- Obsługa podstawowych stron widoków dla administratora i użytkownika.

5.2.2. MyErrorController

Konstrukcja: MyErrorController implementuje interfejs ErrorController i jest oznaczony adnotacją @Controller, co wskazuje, że jest kontrolerem Springa. Klasa ta obsługuje błędy HTTP i zapewnia odpowiednie przekierowanie w zależności od kodu statusu błędu.

Interfejs do innych modułów:

- HttpServletRequest: Używane do pobierania kodu statusu błędu oraz parametrów zapytania.
- Model: Używany do dodawania atrybutów do modelu widoku.
- SecurityContextHolder: Używane do pobierania informacji o aktualnie zalogowanym użytkowniku i jego rolach.

Funkcjonalności:

- Obsługa błędów HTTP i zwracanie odpowiednich stron błędów (np. 403, 404, 500, 504).
- Przekierowanie użytkowników na odpowiednie strony w zależności od ich roli po wystąpieniu błędu.
- Obsługa wylogowania i przekierowanie na stronę logowania z komunikatem o wygaśnięciu sesji.

5.2.3. SecurityConfiguration

Konstrukcja: SecurityConfiguration to klasa konfiguracyjna oznaczona adnotacją @Configuration oraz @EnableWebSecurity, co aktywuje zabezpieczenia sieciowe w aplikacji Spring. Klasa ta dziedziczy po WebSecurityConfigurerAdapter i nadpisuje metody konfiguracyjne, aby dostosować autoryzację i uwierzytelnianie użytkowników.

Interfejs do innych modułów:

- AuthenticationManagerBuilder: Używany do konfigurowania uwierzytelniania użytkowników (w tym przypadku użytkowników w pamięci).
- HttpSecurity: Używany do konfigurowania zasad zabezpieczeń HTTP, takich jak autoryzacja dostępu do poszczególnych zasobów i konfiguracja formularza logowania.
- PasswordEncoder: Interfejs używany do enkodowania haseł.

Funkcjonalności:

- Konfiguracja użytkowników w pamięci z rolami USER i ADMIN.
- Ustawienie encoder'a haseł (w tym przypadku używany jest NoOpPasswordEncoder do prostego przechowywania haseł w postaci zwykłego tekstu).
- Konfiguracja zasad dostępu do zasobów na podstawie ról użytkowników.
- Konfiguracja formularza logowania oraz stron przekierowania po udanym logowaniu i wylogowaniu.

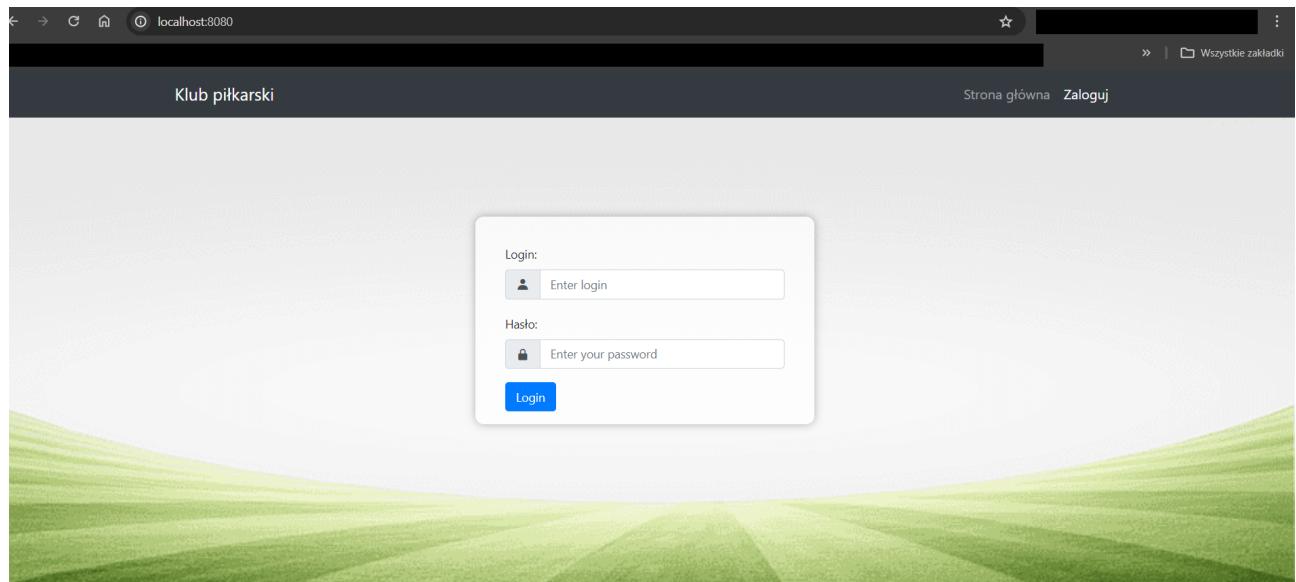
6. Aplikacja - strona internetowa

6.1. Rozmieszczenie komponentów na stronie

Stworzona przez nas strona internetowa jest wykonana w sposób intuicyjny, aby użytkownik mógł łatwo zlokalizować odpowiednie przyciski na stronie oraz z łatwością mógł się po niej poruszać.



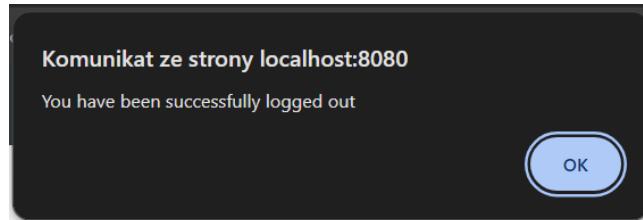
Rysunek 2. Strona startowa



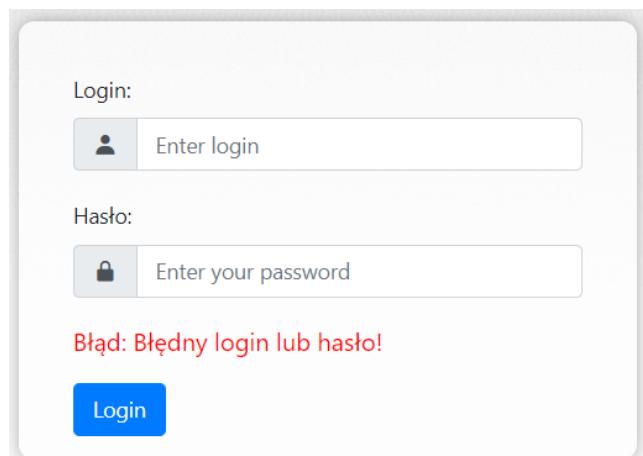
Rysunek 3. Strona logowania

6.2. Komunikaty ze strony i zabezpieczenia

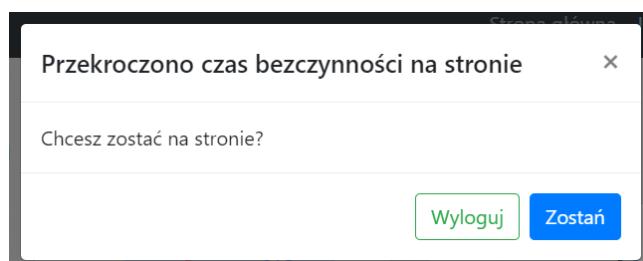
Nasza strona internetowa informuje użytkownika o poprawnym wylogowaniu (Rys.4), błędnych danych logowania (Rys.5), zbyt długim czasie bezczynności (Rys.6), a także w przypadku zostawienia pustego pola podczas edycji rekordu, użytkownik otrzyma informacje, że to pole nie może być puste (Rys.7). Nasza strona zabezpieczona przed wejściem na panel administratora z poziomu konta o ograniczonych uprawnieniach, poprzez wyświetlenie strony z błędem (Rys.9). W ramach zabezpieczenia naszej strony ukryliśmy widoczność całej ścieżki URL do danej strony, przykładowo na stronie logowania widać tylko adres URL: localhost:8080, a w rzeczywistości użytkownik znajduje się na stronie localhost:8080/login co widać na Rys.3.



Rysunek 4. Komunikat o poprawnym wylogowaniu



Rysunek 5. Komunikat o błędnych danych logowania



Rysunek 6. Komunikat o przekroczeniu czasu bezczynności

Edycja biletu

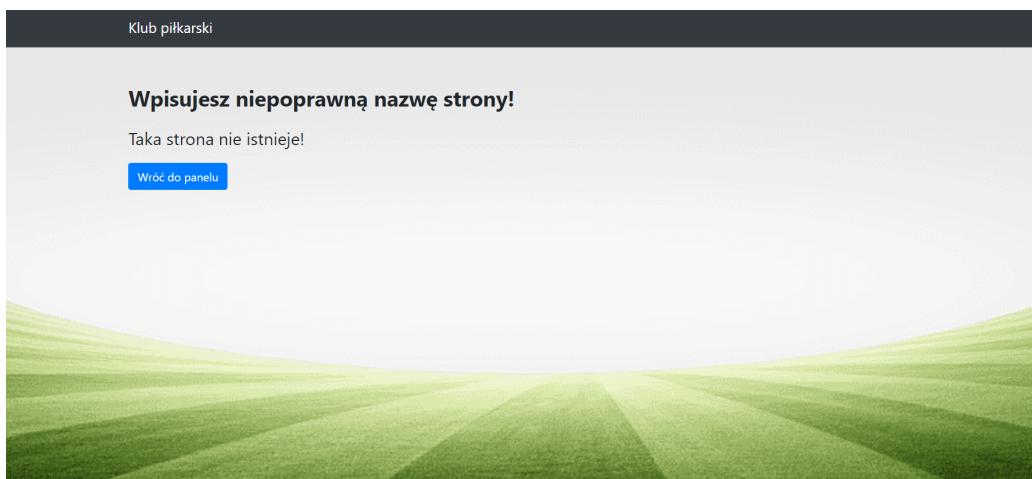
Numer biletu	102
Rodzaj biletu	Normalny
Cena	
Data	<div style="border: 1px solid #ccc; padding: 2px;">dd</div> ! Wypełnij to pole.
Numer klubu	1
Numer kibica	2
Numer meczu	2

[Zapisz](#) [Anuluj](#)

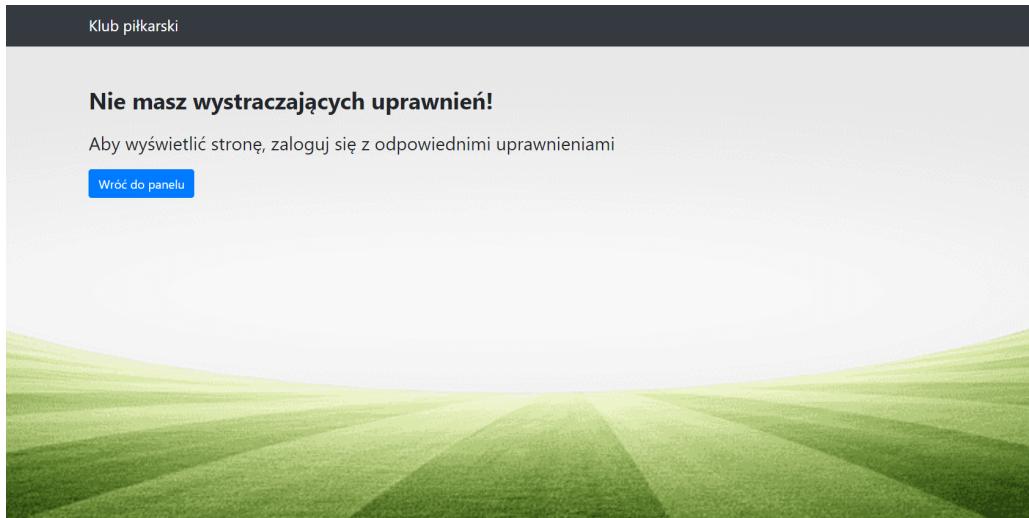
Rysunek 7. Komunikat o pozostawieniu pustego pola podczas edycji rekordu

6.3. Strony obsługujące błędy

Kiedy użytkownik podejmie próbę usunięcia powiązanego rekordu, zalogowania na nieistniejącą stronę (Rys.8), wejścia na stronę do której nie ma uprawnień (Rys.9) będzie chciał podjąć inne niedozwolone działanie, zostanie wyświetlona strona informująca użytkownika o niepoprawnej operacji.



Rysunek 8. Strona z błędem o wpisaniu niepoprawnej nazwy strony



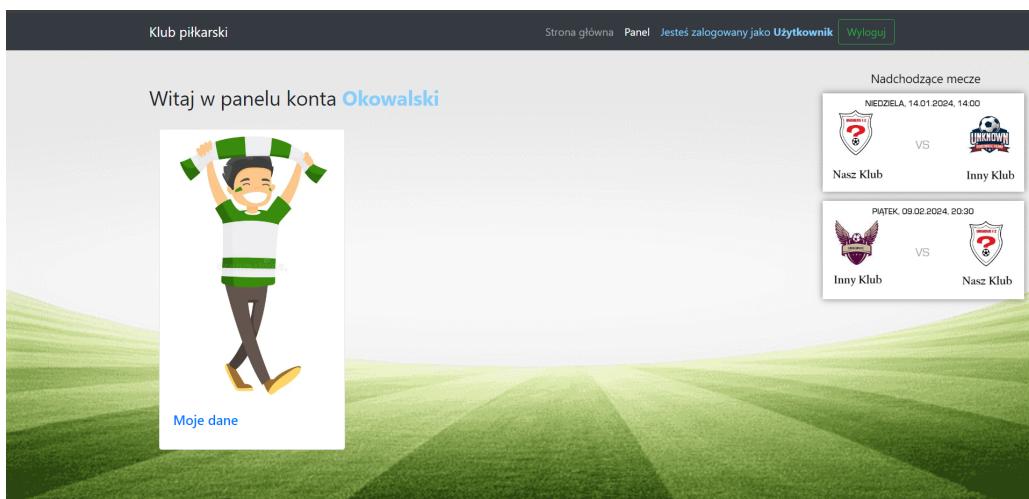
Rysunek 9. Strona z błędem o braku uprawnień do strony

7. Aplikacja - obsługa perspektyw

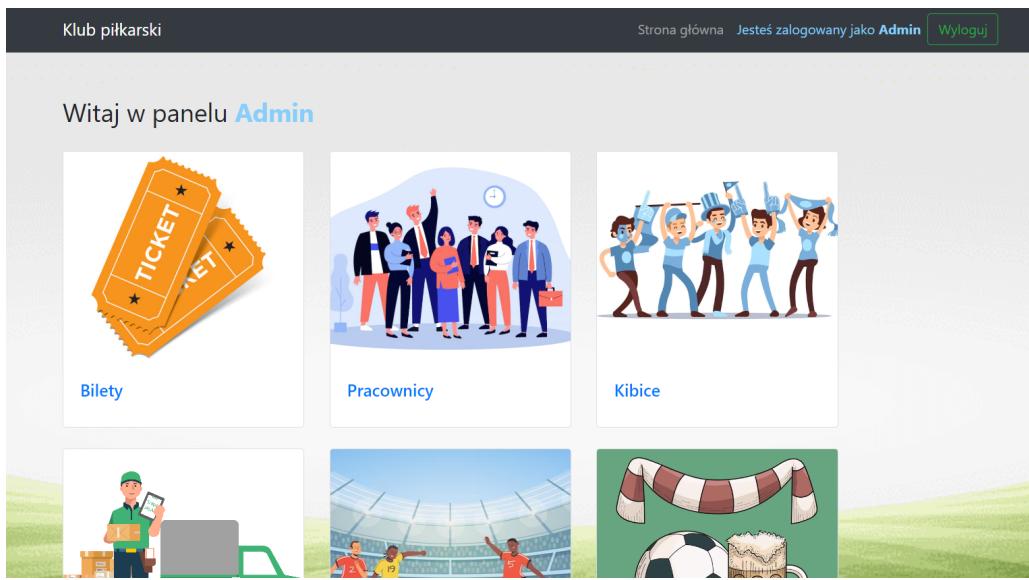
Nasza aplikacja zapewnia prezentacje danych w odniesieniu do dwóch perspektyw użytkowników z niewielkim poziomem uprawnień:

- perspektywa „wąska” - perspektywa kibica, który po zalogowaniu w ramach swojej perspektywy ma jedynie dostęp do swoich danych. Kibic może jedynie przeglądać oraz edytować swoje dane. Widok perspektywy widać na Rys.10. Aby zalogować się na konto kibica należy podać **username: okowalski** oraz **hasło: okowalski**.

-perspektywa „szeroka” - perspektywa administratora strony ma możliwość wyświetlania wprowadzania, usuwania danych oraz modyfikacji już istniejących informacji o kibicach, ale ma także dostęp do innych tabel, przykładowo mecze, zamówienia czy pracownicy. Widok perspektywy widać na Rys.11. Aby zalogować się na konto admina należy podać **username: admin** oraz **hasło: admin**.



Rysunek 10. Perspektywa ”wąska” - perspektywa szefa kibiców



Rysunek 11. Perspektywa "szeroka" - perspektywa administratora

Na rysunkach Rys. 12 oraz Rys.13 przedstawiono różnice w widoku i uprawnieniach do tablicy Kibice między perspektywą wąską oraz szeroką:

Moje dane									
Numer kibica	Imię	Nazwisko	PESEL	Email	Telefon	Data urodzenia	Numer klubu	Numer adresu	Akcje
1	Olgierd	Kowalski	2147483647	okowal@wp.pl	111111112	1999/04/30	1	10	<button>Edytuj</button>
Wróć									

Rysunek 12. Widok tabeli Kibice dla perspektywy wąskiej

8. Aplikacja - operacje w ramach komunikacji z bazą danych

Aplikacja wspiera podstawowe operacje w ramach komunikacji z bazą danych, daje możliwość wykonania czterech podstawowych operacji w kontekście komunikacji z bazą danych, tj.: dodawanie danych, usuwanie danych, modyfikacja danych oraz ich przeglądanie. Dla perspektywy szerokiej możliwe są wszystkie operacje, natomiast dla perspektywy wąskiej poza przeglądaniem danych dodatkowo możemy wykonać operację modyfikacji danych jak przedstawiono na rysunku. Na rysunku Rys.14 widać zmodyfikowany rekord w tabeli Kibice, widać różnicę między tabelą, w której nie modyfikowano danych Rys.13.

Wykonaliśmy operacje na tabeli Produkty, aby przedstawić możliwe działania jakie można wykonać na tej tabeli oraz widok tabli po modyfikacjach. Tabela Produkty bez żadnych modyfikacji została przedstawiona na rysunku Rys.15. Dodaliśmy do tabeli nowy rekord (Rys.16), po czym tabela Produkty wyglądała tak jak na rysunku Rys.17. Wykonaliśmy także operację usuwania rekordów, po której z tabeli Produkty usunęliśmy nowo dodany rekord oraz jeden rekord, który już operował w tabeli wcześniej (Rys.18).

Klub piłkarski

Tabela z kibicami

[Dodaj nowego kibica](#)

Numer kibica	Imię	Nazwisko	PESEL	Email	Telefon	Data urodzenia	Numer klubu	Numer adresu	Akcje
1	Olgierd	Kowal	2147483647	okowal@wp.pl	111111111	2024/01/17	1	10	Edytuj Usuń
2	Aleksandra	Januszczuk	2147483647	ajanuszczuk@wp.pl	222222222	2000/03/08	1	11	Edytuj Usuń
3	Andrzej	Foka	2147483647	afoka@gmail.com	777777777	1994/10/13	1	12	Edytuj Usuń
4	Cezary	Niezugoda	2147483647	cniezugoda@wp.pl	999999999	1979/04/01	1	13	Edytuj Usuń

[Wróć](#)

Rysunek 13. Widok tabeli Kibice dla perspektywy szerokiej

Tabela z kibicami

[Dodaj nowego kibica](#)

Numer kibica	Imię	Nazwisko	PESEL	Email	Telefon	Data urodzenia	Numer klubu	Numer adresu	Akcje
1	Olgierd	<u>Kowalski</u>	2147483647	okowal@wp.pl	111111111	<u>2000/06/19</u>	1	10	Edytuj Usuń
2	Aleksandra	Januszczuk	2147483647	ajanuszczuk@wp.pl	222222222	2000/03/08	1	11	Edytuj Usuń
3	Andrzej	Foka	2147483647	afoka@gmail.com	777777777	1994/10/13	1	12	Edytuj Usuń
4	Cezary	Niezugoda	2147483647	cniezugoda@wp.pl	999999999	1979/04/01	1	13	Edytuj Usuń

[Wróć](#)

Rysunek 14. Widok tabeli Kibice dla perspektywy szerokiej po modyfikacji rekordu

Klub piłkarski

Tabela z produktami

[Dodaj nowy produkt](#)

Numer produktu	Nazwa produktu	Cena	Rodzaj produktu	Producent	Numer klubu	Numer producenta	Akcje
1	Koszulka Dudek XL	220.0	Koszulka	R-gol	1	9	Edytuj Usuń
2	Czapka zimowa	80.0	Czapka	DomKibica	1	12	Edytuj Usuń
3	Koszulka Karczmarek M	200.0	Koszulka	DomKibica	1	12	Edytuj Usuń
4	Szalik klubowy	50.0	Szalik	DomKibica	1	12	Edytuj Usuń

[Wróć](#)

Rysunek 15. Widok tabeli Produkty bez żadnych modyfikacji

Dodaj nowy produkt

Numer produktu:	6
Nazwa produktu:	Koszulka wyjazdowa
Cena:	180
Rodzaj produktu:	Koszulka
Nazwa producenta:	R-gol
Numer klubu:	1
Numer producenta:	9

Zapisz Anuluj

Rysunek 16. Widok strony z dodawaniem nowego rekordu do tabeli Produkty

Klub piłkarski

Tabela z produktami

Dodaj nowy produkt

Numer produktu	Nazwa produktu	Cena	Rodzaj produktu	Producent	Numer klubu	Numer producenta	Akcje
1	Koszulka Dudek XL	220.0	Koszulka	R-gol	1	9	<button>Edytuj</button> <button>Usuń</button>
2	Czapka zimowa	80.0	Czapka	DomKibica	1	12	<button>Edytuj</button> <button>Usuń</button>
3	Koszulka Karczmarek M	200.0	Koszulka	DomKibica	1	12	<button>Edytuj</button> <button>Usuń</button>
4	Szalik klubowy	50.0	Szalik	DomKibica	1	12	<button>Edytuj</button> <button>Usuń</button>
64	Koszulka wyjazdowa	180.0	Koszulka	R-gol	1	9	<button>Edytuj</button> <button>Usuń</button>

Wróć

Rysunek 17. Widok tabeli Produkty po dodaniu nowego rekordu

Klub piłkarski

Tabela z produktami

Dodaj nowy produkt

Numer produktu	Nazwa produktu	Cena	Rodzaj produktu	Producent	Numer klubu	Numer producenta	Akcje
1	Koszulka Dudek XL	220.0	Koszulka	R-gol	1	9	<button>Edytuj</button> <button>Usuń</button>
2	Czapka zimowa	80.0	Czapka	DomKibica	1	12	<button>Edytuj</button> <button>Usuń</button>
3	Koszulka Karczmarek M	200.0	Koszulka	DomKibica	1	12	<button>Edytuj</button> <button>Usuń</button>

Wróć

Rysunek 18. Widok tabeli Produkty po usunięciu rekordów

9. opis instalacji Produktu

9.1. Założenie osobistego konta w serwisie Oracle

Najpierw należy założyć osobiste konto w serwisie Oracle(chyba, że już posiadamy konto, wtedy pomijamy ten krok). W tym celu należy udać się na stronę: <https://profile.oracle.com/myprofile/account/create-account.jspx>.

The screenshot shows the 'Utwórz konto Oracle' (Create Oracle Account) page. It includes fields for: Adres e-mail*, Hasło*, Wpisz ponownie hasło*, Kraj (Poland selected), Imię i nazwisko*, Telefon służbowy*, Nazwa firmy*, Adres*, and Miasto*. There are also notes about password complexity and a green shield icon indicating security.

Na stronie tej wypełniamy formularz i klikamy „Utwórz konto”

internetowej Oracle podlega **Warunkom użytkowania** wi
zbierania i wykorzystywania przez firmę Oracle Twoich d
przechowywania, korygowania, usuwania, zabezpieczeń
Zasadach ochrony prywatności firmy Oracle.

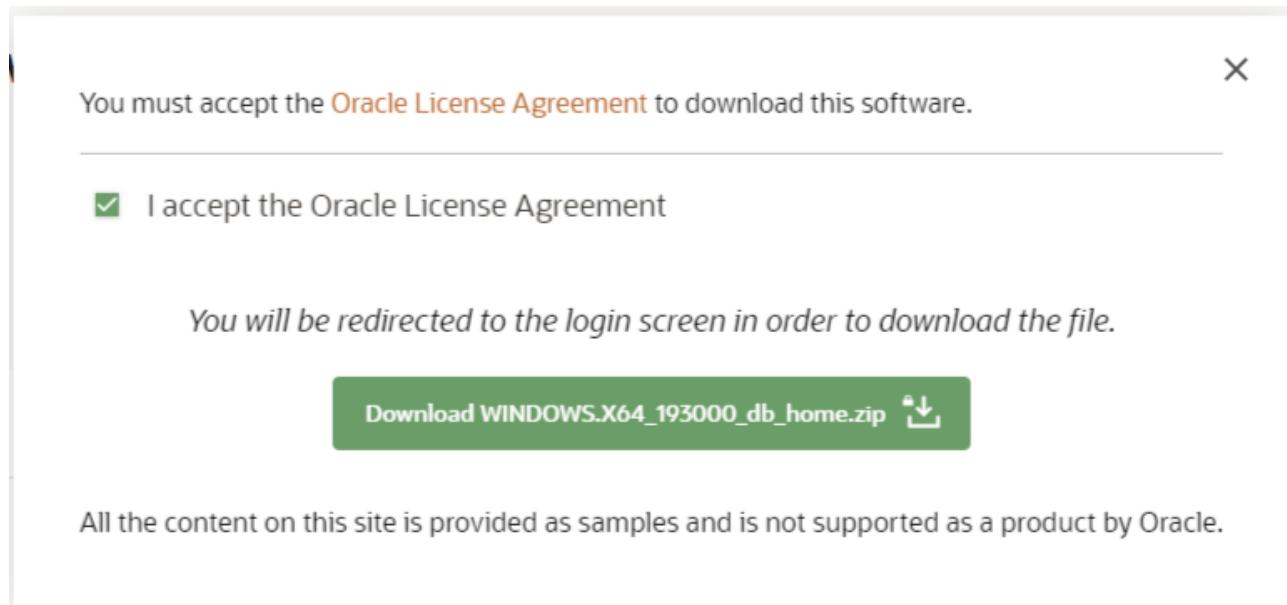
Utwórz konto

Gdy dysponujemy już swoim kontem
nie ma już żadnych formalnych
przeszkód abyśmy zaczęli pobierać
oprogramowanie ze strony Oracle.

Rysunek 19. Widok zakładania konta w serwisie Oracle

9.2. Oprogramowanie bazy danych Oracle

W celu pobrania bazy danych Oracle wchodzimy na stronę Oracle. Wybieramy wersję bazy danych, którą jesteśmy zainteresowani. W naszym przypadku była to Oracle Database 21c Express Edition for Windows x64. Następnie podczas instalacji Akceptujemy warunki korzystania z oprogramowania.



All the content on this site is provided as samples and is not supported as a product by Oracle.

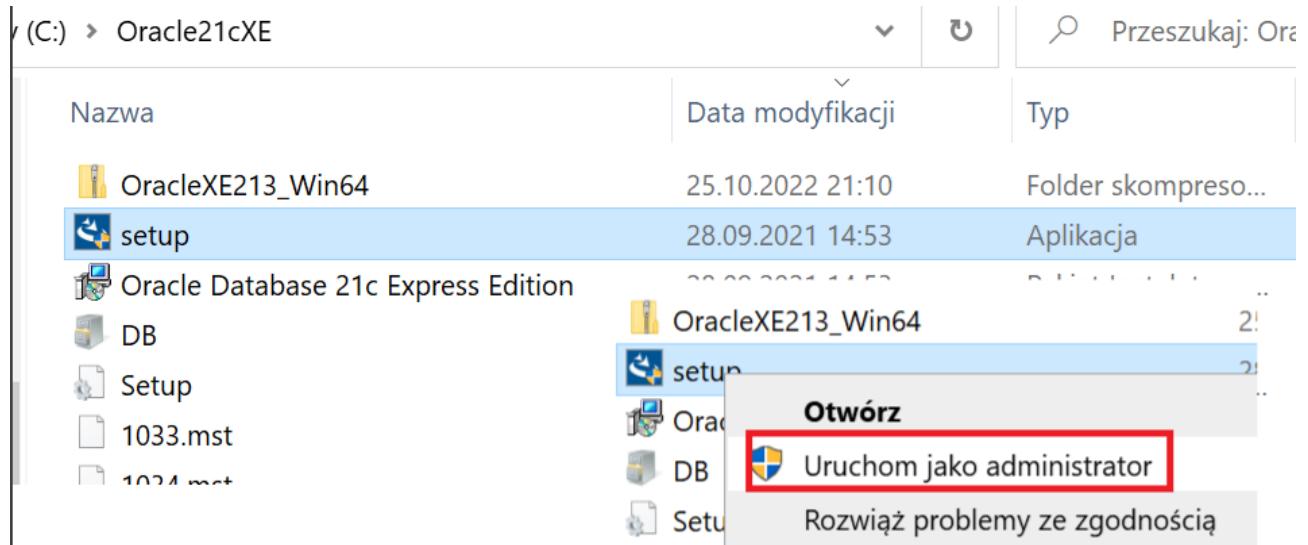
Rysunek 20. Widok z akceptowania warunków korzystania z oprogramowania

Zanim rozpocznie się pobieranie oprogramowania musimy się zautoryzować się. W tym celu wykorzystujemy wcześniejszą założone konto Oracle(lub te które już mieliśmy).



Rysunek 21. Widok z logowania

Po prawidłowym zautoryzowaniu proces pobierania oprogramowania rozpoczyna się automatycznie. W tym celu przechodzimy do katalogu do którego ściągnęliśmy plik, a następnie rozpakowujemy go. Po rozpakowaniu uruchamiany instalator, tj. program „setup.exe”. Należy uruchomić z użytkownika Administrator.

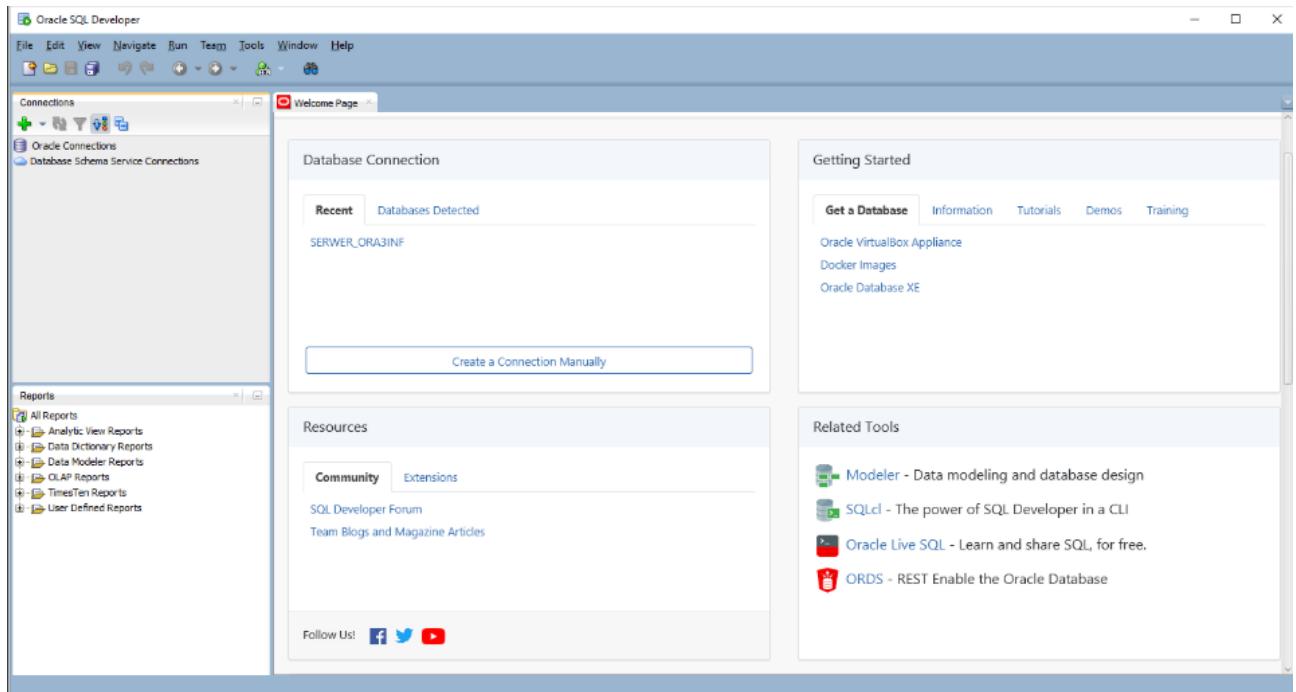


Rysunek 22. Uruchamianie instalatora

Musimy współpracować z kreatorem instalacji bazy i 'przeklikować' kolejne kroki. Współpraca ta wymaga m.in. udzielania odpowiedzi na proste pytania, lub podania haseł niezbędnych dla prawidłowego skonfigurowania bazy danych. Następnie proces instalacji i konfiguracji może zająć dłuższą chwilę, w zależności od możliwości sprzętowych.

9.3. Połaczanie do lokalnej bazy danych Oracle

W tym kroku należyściągnąć i zainstalować narzędzie graficzne SQL Developer(np: ze strony <https://www.oracle.com/database/>). Prawidłowa praca narzędzia SQL Developer wymaga dostępu do JDK Java. W związku z tym jeśli nie ustawiśmy prawidłowo ścieżki dla JDK Java podczas pracy instalatora bazy, musimy to zrobić teraz. Podanie prawidłowej ścieżki do JDK Java, powinno spowodować, że zostanie uruchomione narzędzie SQL Developer.



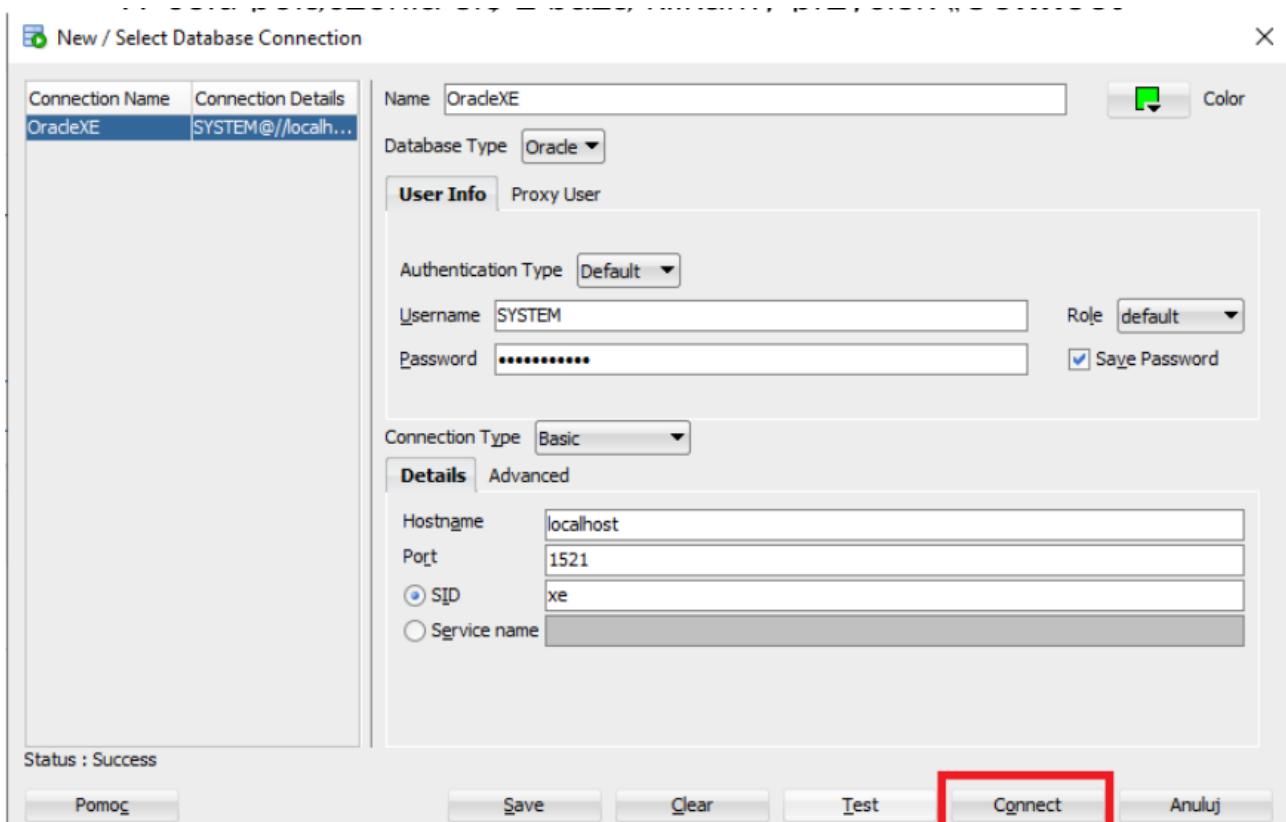
Rysunek 23. Widok z programu SQL Developer

Z poziomu zakładki „New / Select Database Connection” mamy możliwość tworzenia nowych definicji połączeń lub wykorzystywać wcześniejsz zdefiniowane. Ponieważ nie mamy jeszcze żadnego zdefiniowanego połączenia tworzymy go wypełniając pola:

- Connection Name – wartość dowolna
- Username – SYSTEM
- Password – hasło użyte przy instalacji bazy
- Hostname – localhost
- Port – 1521 lub inny jeżeli zmieniliśmy podczas instalacji
- SID – XE

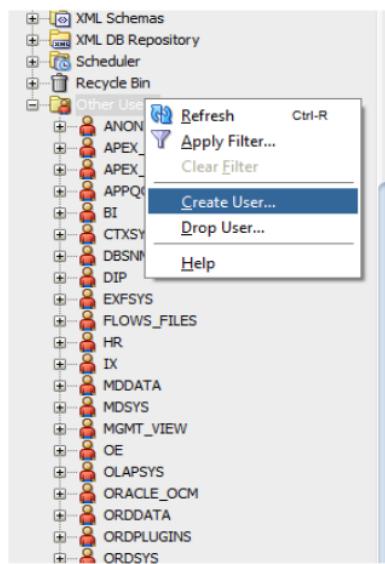
Klikamy przycisk „Test”, jeśli wszystko działa prawidłowo powinniśmy zobaczyć komunikat: Success.

W celu połączenia się z bazą klikamy przycisk „Connect”.



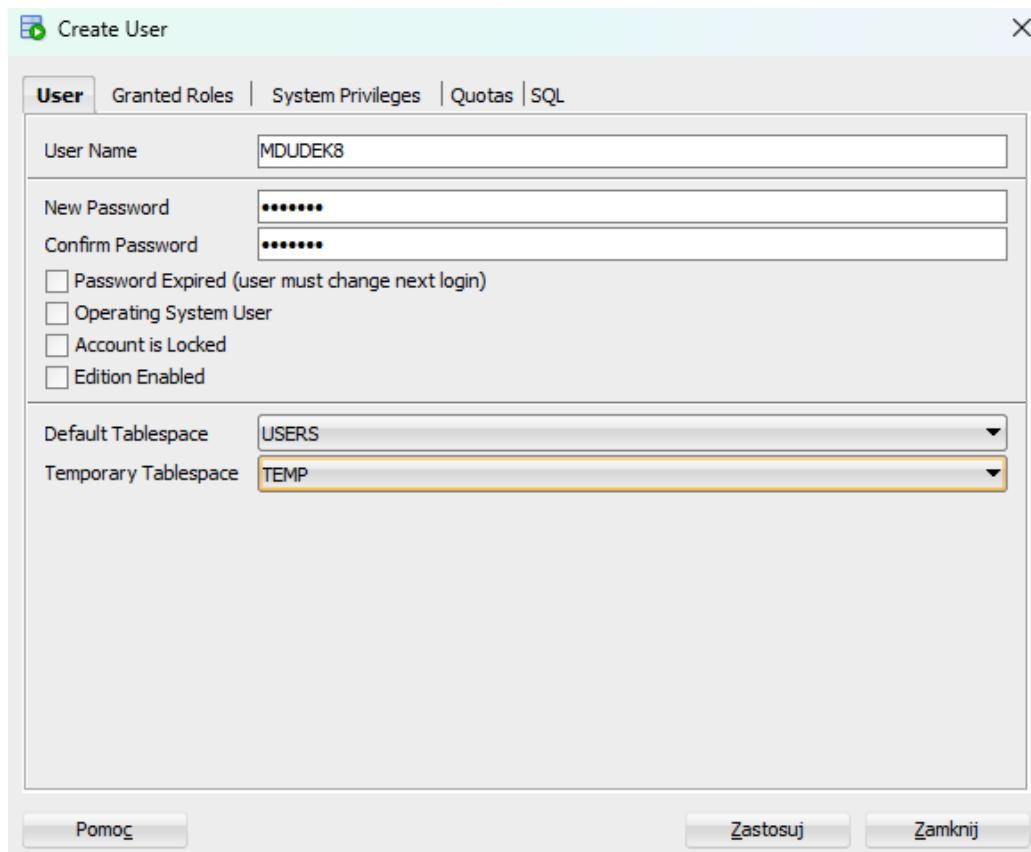
Rysunek 24. Widok z tworzenia połączenia

Ponieważ schemat SYSTEM pełni wybitnie znaczącą rolę w ramach w bazie Oracle, koniecznym jest utworzenie nowego dedykowanego schematu bazy (użytkownika), w którym będziemy tworzyć obiekty. W tym celu klikamy prawym przyciskiem na liść „Other Users” w liście obiektów i wybieramy opcje „Create User”.



Rysunek 25. Widok z tworzenia usera

Wypełniamy pole „User Name”, „New Password”, „Confirm Password” w zakładce „Users”. W naszym przypadku login i hasło są takie same.

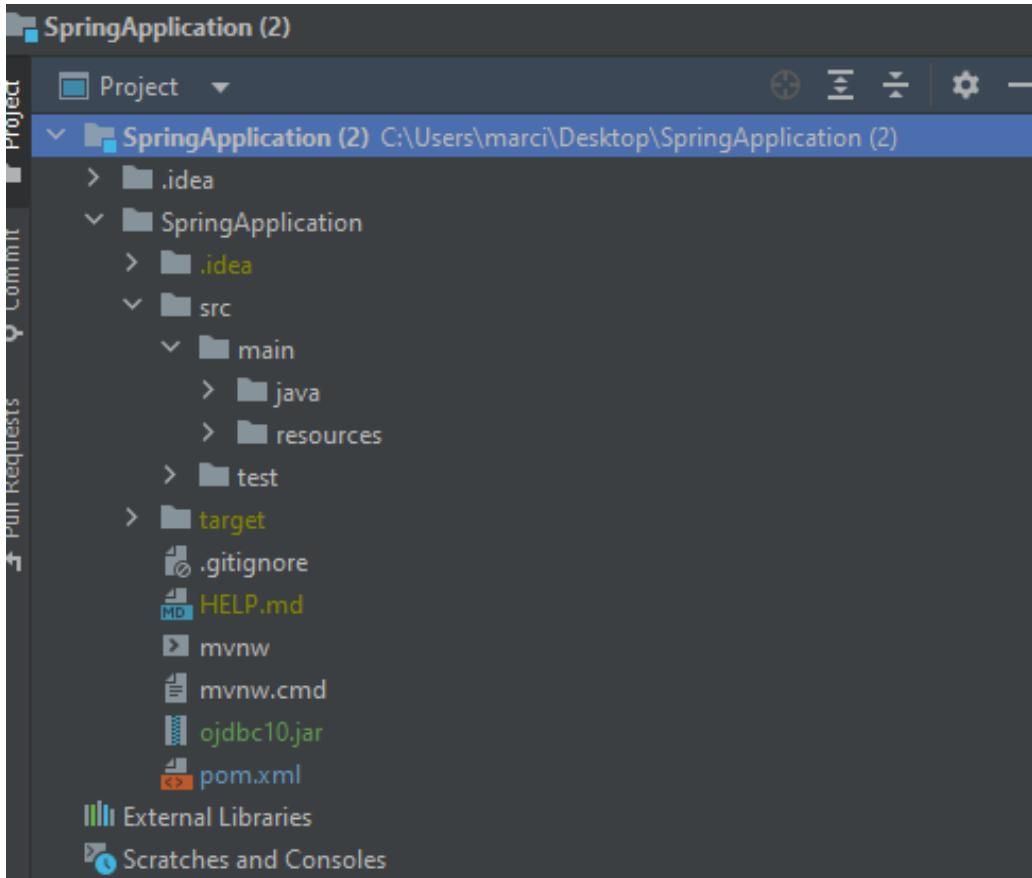


Rysunek 26. Widok zwypełniania pól usera

Następnie pobieramy bazę danych z naszego repozytorium i logujemy się na nią utworzonym kontem. W taki sposób otrzymaliśmy dostęp do bazy danych z poziomu SQL Developer.

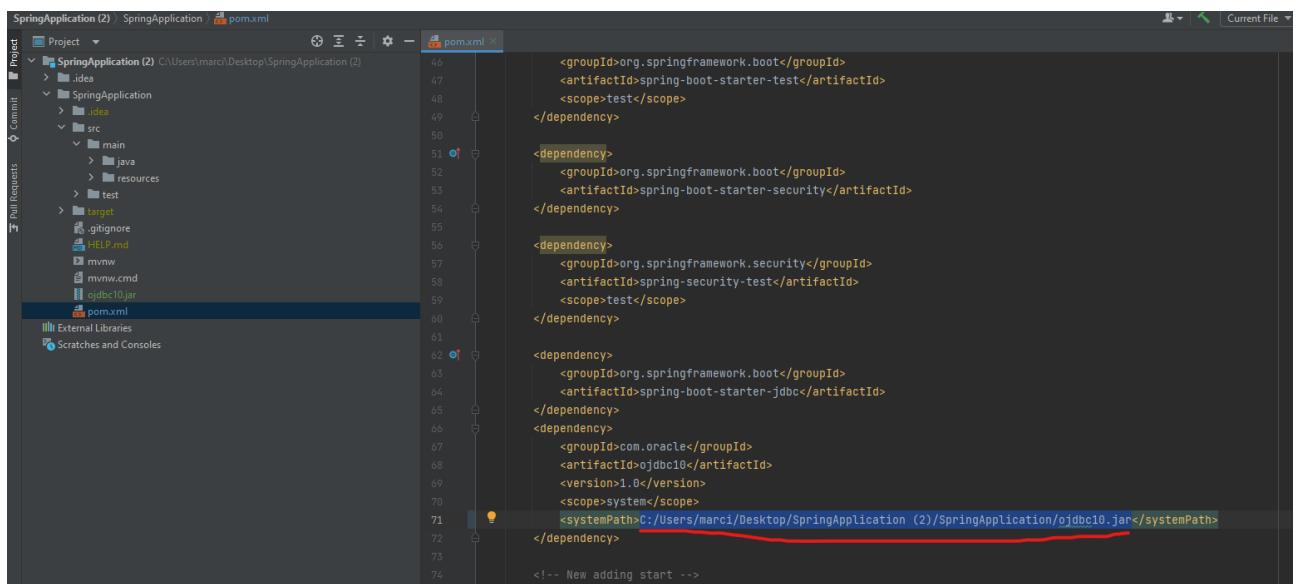
10. Uruchamianie strony internetowej i łączenie się z bazą danych

Następnie należy pobrać folder SpringApplication (2) z naszego repozytorium. Folder należy otworzyć w np: Intelij. Poniżej przedstawiamy jak powinien wyglądać struktura plików w Intelij.



Rysunek 27. Struktura plików w Intelij

Następnie należy w pliku pom.xml zmienić ścieżkę systemPath na ścieżkę do ojdbc(w naszym przypadku ojdbc, które wykorzystujemy jest wewnątrz pobranego folderu).



Rysunek 28. Zmiana ścieżki systemPath

W pliku application.properties należy ustawić odpowiedni login i hasło do bazy danych(w zależności jakie konto użytkownika używamy), tak jak jest to na poniższym rysunku:

The screenshot shows the IntelliJ IDEA interface. On the left, the project structure for 'SpringApplication (2)' is visible, containing .idea, SpringApplication, src (with main and resources), and static/templates/application.properties. The right side shows the contents of application.properties:

```

server.error.whitelabel.enabled=false
spring.thymeleaf.enabled=true
spring.thymeleaf.prefix=classpath:/templates/
spring.thymeleaf.suffix=.html

spring.datasource.url=jdbc:oracle:thin:@localhost:1521:xe
spring.datasource.username=M DUDEK8
spring.datasource.password=M DUDEK8
spring.datasource.driver-class-name=oracle.jdbc.OracleDriver
logging.level.root=INFO

```

Rysunek 29. Widok pliku application.properties

Następnie należy 'zbudować' plik pom.xml i powinniśmy otrzymać komunikat o sukcesie w następujący sposób:

The screenshot shows the IntelliJ IDEA interface during the build process of pom.xml. The left sidebar shows the project structure with .idea, SpringApplication, src (containing main, java, resources, test, target, .gitignore, HELP.md, mvnw, mvnw.cmd,ojdbc10.jar, pom.xml), External Libraries, and Scratches and Consoles. The right pane shows the XML code of pom.xml with some lines highlighted in blue. At the bottom, the Build Output window shows the build progress:

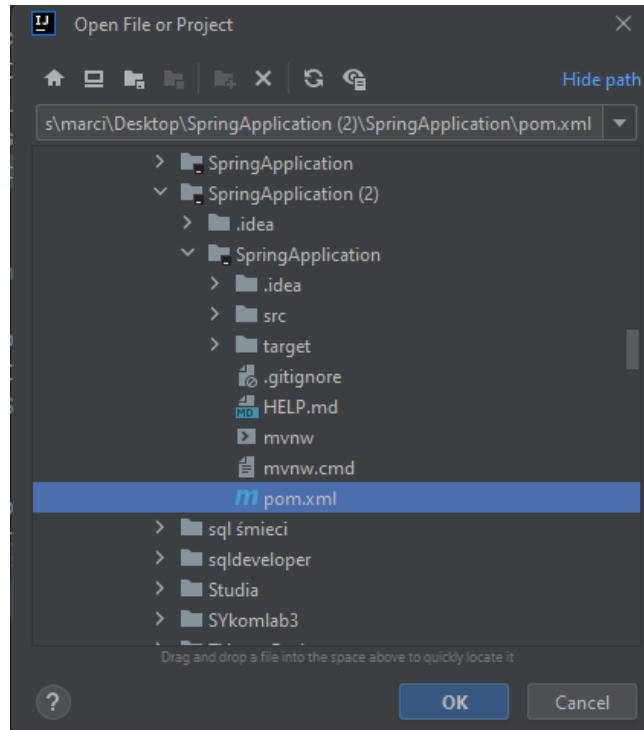
```

Build: Build Output X
SpringApplication (2): build finished UP-TO-DATE at 10.06.2024 22:56
1 sec, 666 ms Executing pre-compile tasks...
Running 'before' tasks
Checking sources
Running 'after' tasks
Finished, saving caches.
Executing post-compile tasks...
Finished, saving caches...

```

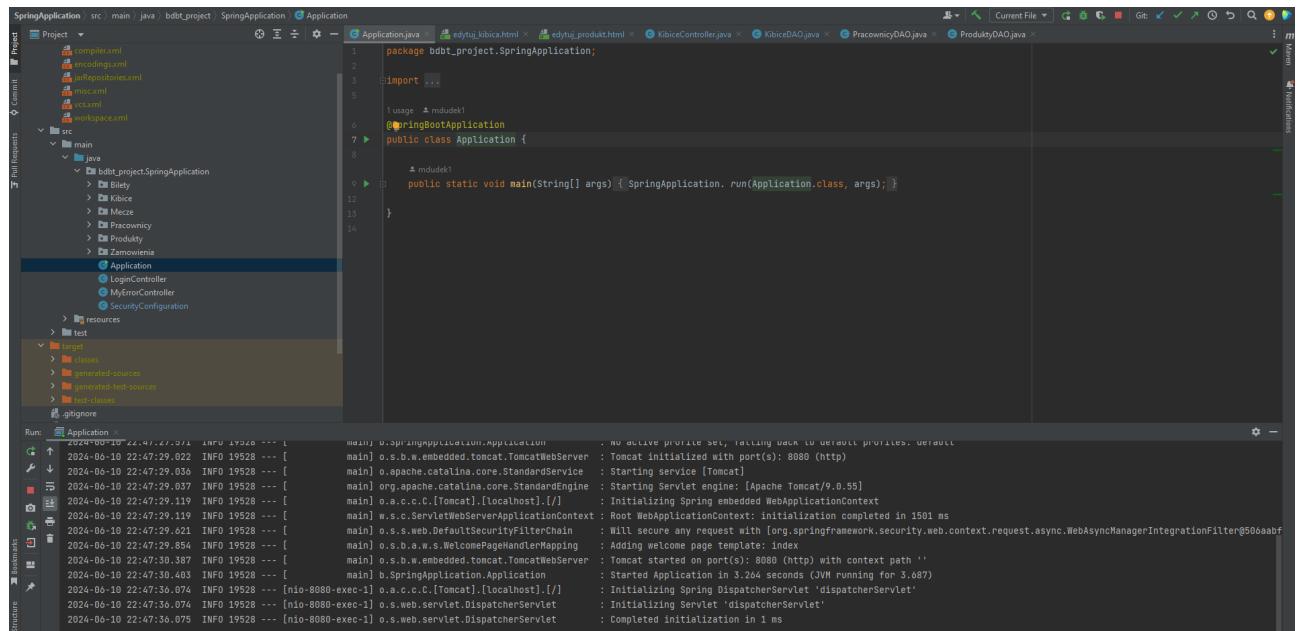
Rysunek 30. Budowa pliku pom.xml

Następnie w lewym górnym rogu Intelij naciskamy 'File' następnie 'Open' i otwieramy plik pom.xml z naszego folderu i naciskamy 'Ok'.



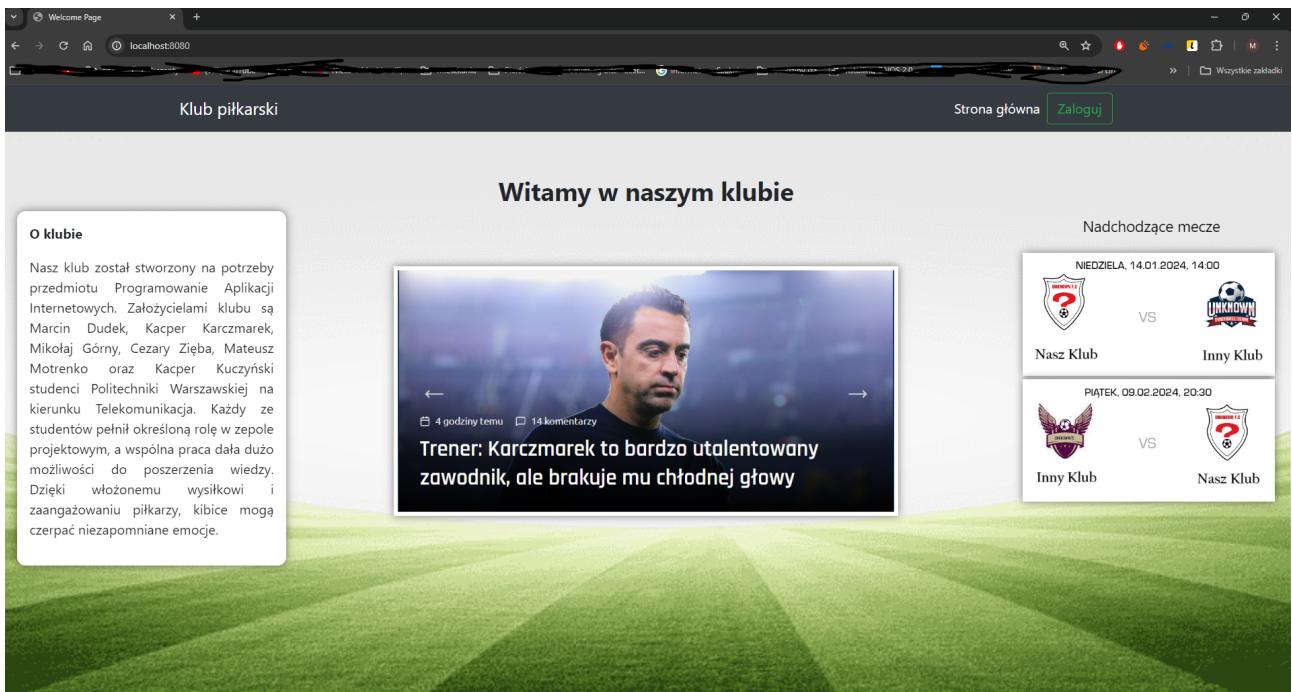
Rysunek 31. Otwieranie projektu

Następnie naciskamy przycisk, potwierdzając, że uufamy projektowi i otwieramy plik w tym samym oknie. Na końcu uruchamiamy naszą aplikację uruchamiając naszą klasę 'Application'.



Rysunek 32. Uruchomiona strona internetowa

Następnie w przeglądarce internetowej wpisujemy 'localhost:8080' i naszym oczom powinna ukazać się strona internetowa klubu piłkarskiego.



Rysunek 33. Uruchomiona strona internetowa

11. Podsumowanie i wnioski

Stworzona przez nas strona spełnia wymagania projektowe. Jest intuicyjna, przejrzysta, wykonana w estetyczny sposób, aby użytkownik mógł poruszać się po niej swobodnie. Aplikacja zapewnia prezentacje danych w odniesieniu do co najmniej dwóch perspektyw użytkowników z niej korzystających (wąskiej oraz szerokiej). Strona wspiera podstawowe operacje w ramach komunikacji z bazą danych, rozumianych jako możliwość wykonania czterech podstawowych operacji w kontekście komunikacji z bazą danych. Realizacja całego projektu dała nam dużą satysfakcję oraz wzbiogaciła o fundamentalną wiedzę z zakresu programowania aplikacji internetowych. Praca nad bazą danych oraz stroną internetową sprawiła nam przyjemność i zachęciła do pogłębiania tego tematu.