

Dokumentacja projektu:  
"Aplikacja współpracująca z bazą danych: FitFit"

Zespół projektowy:  
Artur Sliepchenko 335900, Denys Moldovan 335967

11 maja 2025

# Spis treści

1	Wprowadzenie	2
2	Wykorzystane technologie	2
3	Opis funkcjonalności aplikacji	3
4	Perspektywy użytkownika	4
5	Operacje na bazie danych	10
6	Uwagi i podsumowanie	10

# 1 Wprowadzenie

Celem projektu jest stworzenie aplikacji współpracującej z wcześniej zaprojektowaną bazą danych. Aplikacja ma umożliwiać wykonywanie podstawowych operacji CRUD (create, read, update, delete) oraz zapewniać różnorodne funkcjonalności dla różnych grup użytkowników.

Celem praktycznym członków zespołu było poznanie nowych technologii, takich jak framework Java Spring, oraz upewnienie się, że główne tematy przedmiotu, takie jak projektowanie i zarządzanie bazami danych, są solidnie opanowane w praktyce.

## 2 Wykorzystane technologie

W ramach realizacji projektu wykorzystano następujące technologie:

- **Język programowania: Java**

Jeden z najpopularniejszych języków programowania, charakteryzujący się niezależnością od platformy dzięki zastosowaniu wirtualnej maszyny JVM (Java Virtual Machine). Ma bogatą ekosferę bibliotek i frameworków, które wspierają rozwój aplikacji wielowarstwowych. Java zapewnia bezpieczeństwo, stabilność i możliwość obsługi aplikacji o dużym stopniu złożoności.

- **Frameworki: Spring, Hibernate**

Spring to zaawansowany framework dla języka Java. W projekcie wykorzystano Spring do obsługi warstwy backendowej, w tym zarządzania logiką biznesową i integracji z bazą danych. Hibernate, z kolei, jest frameworkiem ORM (Object-Relational Mapping), który umożliwia mapowanie obiektów w Javie na tabele w bazie danych. Dzięki Hibernate możliwe było ograniczenie ilości zapytań SQL w kodzie oraz uproszczenie pracy z danymi.

- **Frontend: HTML, CSS, JavaScript**

HTML (HyperText Markup Language) został wykorzystany do tworzenia struktury stron internetowych. CSS (Cascading Style Sheets) odpowiadał za stylizację aplikacji, umożliwiając stworzenie nowoczesnego i estetycznego interfejsu użytkownika. JavaScript był używany do implementacji dynamicznych elementów, takich jak interaktywne menu czy walidacja danych po stronie klienta. Kombinacja tych technologii pozwoliła na stworzenie przejrzystego i intuicyjnego interfejsu aplikacji.

- **Backend: Spring Boot**

Spring Boot jest rozszerzeniem frameworka Spring, które upraszcza konfigurację aplikacji oraz przyspiesza jej rozwój dzięki mechanizmowi auto-konfiguracji. Umożliwia tworzenie samodzielnych aplikacji, które mogą być uruchamiane bez konieczności dodatkowych serwerów aplikacyjnych. W projekcie Spring Boot został wykorzystany do implementacji API, obsługi żądań użytkowników oraz integracji z bazą danych PostgreSQL.

- **Baza danych: PostgreSQL**

PostgreSQL to zaawansowany, otwartoźródłowy system zarządzania relacyjnymi bazami danych. Jest on znany ze swojej stabilności, wydajności oraz wsparcia dla rozbudowanych zapytań SQL. W projekcie PostgreSQL pełnił rolę głównej bazy

danych, przechowując dane aplikacji oraz wspierając transakcyjne operacje CRUD. Dzięki jego rozbudowanym funkcjom, takim jak indeksowanie i optymalizacja zapytań, możliwe było szybkie i efektywne przetwarzanie danych.

- **Bezpieczeństwo: Spring Security**

Spring Security to potężny framework zapewniający kompleksowe mechanizmy ochrony aplikacji. W projekcie wykorzystano go do implementacji bezpiecznego systemu logowania, autoryzacji użytkowników oraz zabezpieczenia zasobów przed nieautoryzowanym dostępem. Mechanizmy takie jak szyfrowanie haseł, kontrola ról użytkowników oraz ochrona przed atakami CSRF (Cross-Site Request Forgery) gwarantowały bezpieczeństwo danych przechowywanych w aplikacji.

### 3 Opis funkcjonalności aplikacji

Aplikacja została zaprojektowana z myślą o zapewnieniu prostego i intuicyjnego sposobu obsługi dla różnych grup użytkowników, takich jak nowi użytkownicy, trenerzy, zawodnicy oraz administratorzy. Główne funkcjonalności aplikacji to:

- **Rejestracja i logowanie użytkowników**

Użytkownicy mogą zakładać konta w systemie, podając wymagane dane, takie jak nazwa użytkownika i hasło. Mechanizm rejestracji został wzbogacony o walidację poprawności wprowadzonych danych oraz szyfrowanie haseł, co zapewnia bezpieczeństwo danych. Logowanie odbywa się za pomocą bezpiecznej strony, która weryfikuje uprawnienia użytkownika i umożliwia mu dostęp do funkcjonalności aplikacji odpowiednich dla jego roli.

- **Operacje na danych w kontekście perspektyw wąskiej i szerokiej**

Aplikacja wspiera różne role użytkowników:

- Zwykli użytkownicy (perspektywa wąska) mogą przeglądać i edytować swoje dane osobowe, takie jak adres zamieszkania czy numer telefonu. Mogą również poprosić o zmianę swojej roli na trenera, zawodnika lub pracownika biura.
- Administratorzy (perspektywa szeroka) mają dostęp do funkcji zarządzania systemem, takich jak dodawanie i usuwanie użytkowników, zatwierdzanie ról czy edycja globalnych danych, takich jak dodawanie klubów czy zespołów.
- Trenerzy (perspektywa wąska) mają uprawnienia do przeglądania swoich danych o zarobkach oraz możliwość zmiany składu zespołów. Mogą również tworzyć zespoły, przypisując do nich zawodników.
- Pracownicy biura (perspektywa wąska) mają uprawnienia do edycji danych przypisanych zawodnikom oraz trenerom. Odgrywają kluczową rolę w sprawach organizacyjnych, takich jak organizowanie wydarzeń sportowych.
- Zawodnicy (perspektywa wąska) to najliczniejsza grupa użytkowników o najmniejszych uprawnieniach. Mogą dołączać do nowych grup, uczestniczyć w zajęciach, przeglądać dane swoich trenerów oraz innych zawodników z grup zajęciowych, w których uczestniczą.

- **Zarządzanie dostępem i uprawnieniami**

Mechanizmy autoryzacji zapewniają, że użytkownicy mają dostęp wyłącznie do

funkcji odpowiednich dla ich roli. Na przykład zwykły użytkownik nie może uzyskać dostępu do panelu administracyjnego.

- **Główna strona**

Główna strona łączy wszystkich użytkowników i pozwala przeglądać wydarzenia sportowe, nawet gdy użytkownik nie jest zalogowany. Umożliwia również dostęp do swojego profilu, gdzie znajdują się różne kategorie danych. Możliwość dostępu do poszczególnych funkcji zależy od roli użytkownika i przypisanych mu uprawnień.

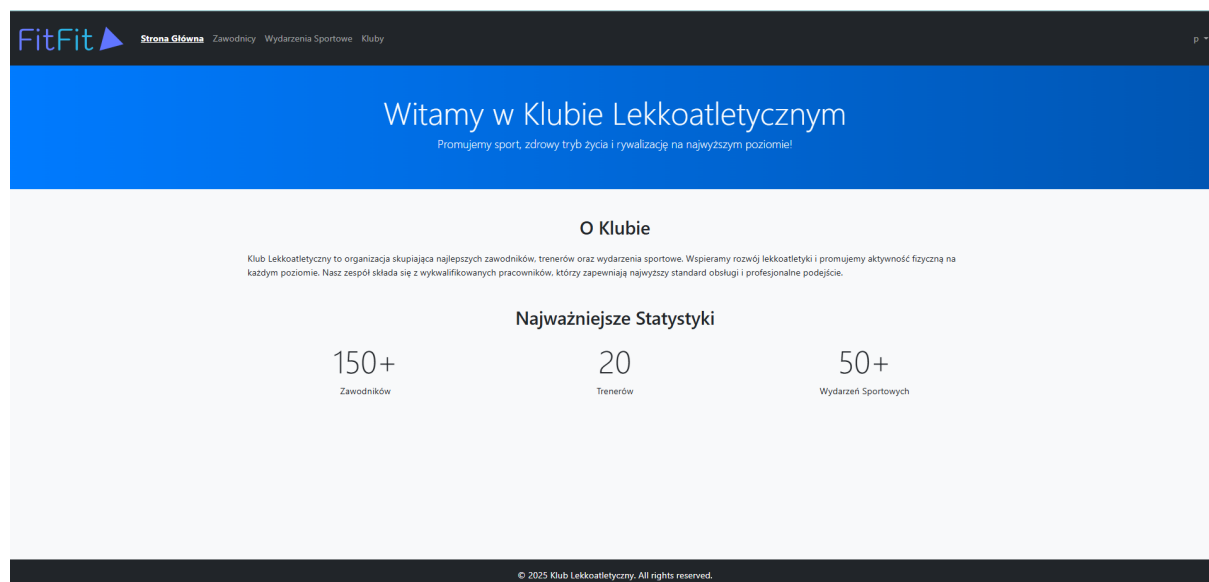
## 4 Perspektywy użytkownika

### Główna strona

Główna strona aplikacji jest centralnym punktem dostępu dla wszystkich użytkowników, zarówno zalogowanych, jak i gości. Oferuje przyjazny interfejs, który umożliwia przeglądanie najważniejszych informacji o klubie, takich jak statystyki (liczba zawodników, trenerów, wydarzeń sportowych) oraz krótki opis działalności klubu. Strona została zaprojektowana z myślą o intuicyjności i estetyce, łącząc proste elementy nawigacyjne z nowoczesnym stylem wizualnym.

W sekcji nawigacyjnej znajdują się odnośniki do głównych funkcjonalności aplikacji, takich jak zarządzanie grupami, przegląd wydarzeń sportowych oraz podgląd klubów. Na samej górze strony widoczny jest slogan motywujący do aktywności fizycznej i zdrowego trybu życia.

Dodatkowo, strona zawiera interaktywne elementy umożliwiające szybki dostęp do szczegółowych informacji o wydarzeniach sportowych oraz klubach. Ilustracja tej strony została przedstawiona na obrazie poniżej (Rysunek 1).



Rysunek 1: Widok głównej strony aplikacji

## Wydarzenia sportowe

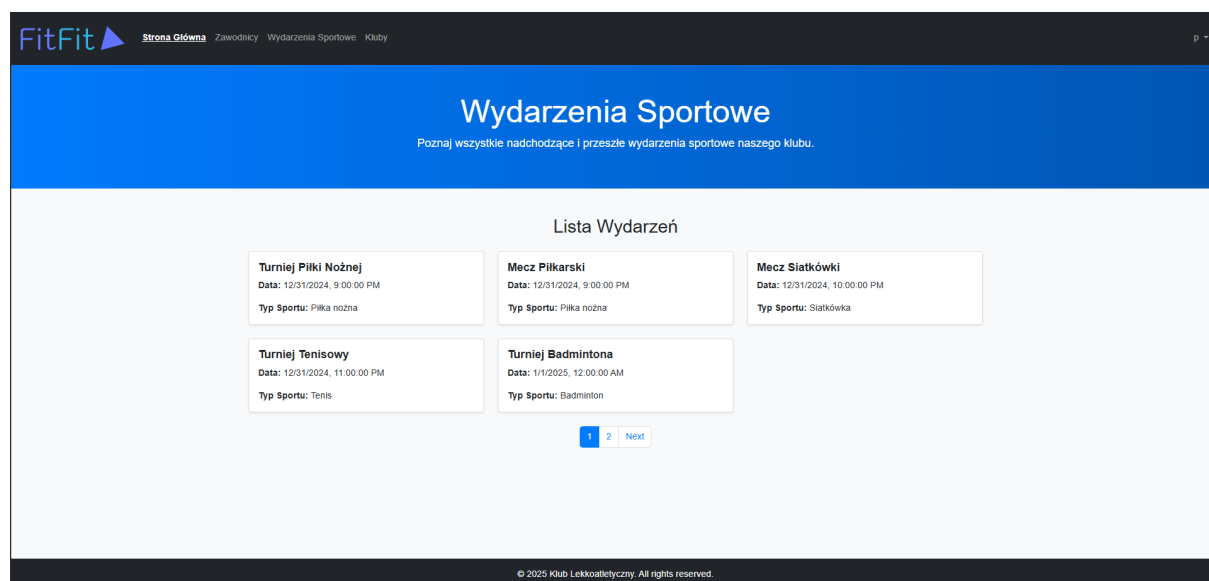
Sekcja „Wydarzenia sportowe” umożliwia użytkownikom przeglądanie wszystkich nadchodzących i przeszłych wydarzeń organizowanych przez kluby "FitFit". Dane dotyczące wydarzeń są dynamicznie pobierane z bazy danych, co zapewnia aktualność informacji.

Strona umożliwia wyświetlanie wydarzeń w postaci listy, gdzie na każdej stronie widocznych jest maksymalnie 5 wydarzeń. Dzięki zastosowaniu paginacji użytkownicy mogą wygodnie przemieszczać się między stronami, przeglądając kolejne wydarzenia. Każdy element listy zawiera kluczowe informacje o wydarzeniu, takie jak:

- **Nazwa wydarzenia** – przedstawia nazwę danego wydarzenia, np. „Turniej Piłki Nożnej”.
- **Data i godzina** – określa termin i godzinę rozpoczęcia wydarzenia, np. „31.12.2024, 21:00”.
- **Typ sportu** – wskazuje dyscyplinę sportową, np. „Piłka nożna” lub „Tenis”.

Projekt interfejsu łączy prostotę z czytelnością. Na górze sekcji znajduje się nagłówek zachęcający do poznania wszystkich wydarzeń sportowych, co wzmacnia pozytywne wrażenia użytkownika.

Poniższa strona (Rysunek 2) prezentuje przykład interfejsu z danymi, który umożliwia szybki i intuicyjny dostęp do szczegółowych informacji o wydarzeniach.

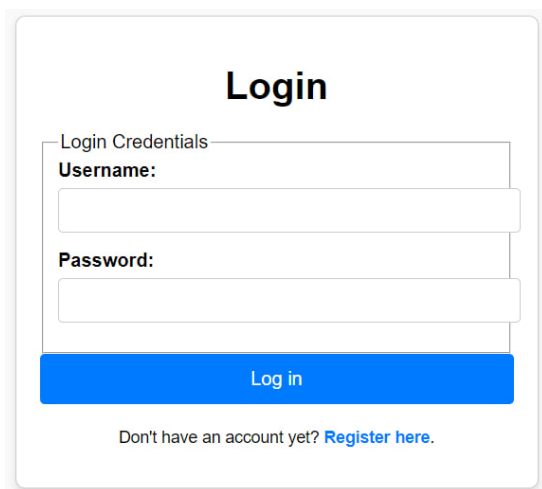


Rysunek 2: Widok sekcji „Wydarzenia sportowe” z dynamicznie generowaną listą wydarzeń

## Formularz logowania

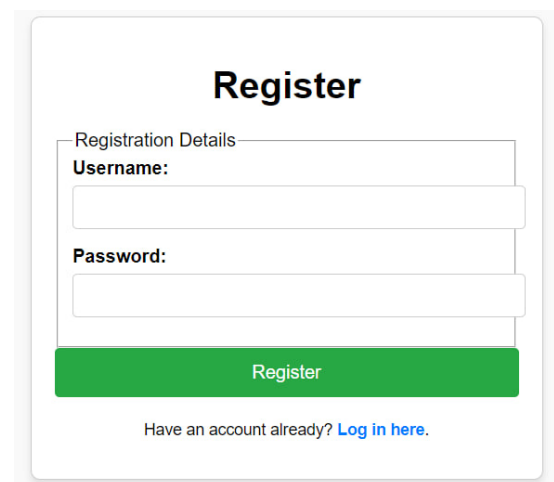
- **Tytuł:** Login
- **Pola formularza:**
  - Username (pole tekstowe)

- Password (pole tekstowe)
- **Przycisk:** Log in (kolor niebieski)
- **Link rejestracyjny:** Register here (kolor niebieski, podkreślony)
- **Proces logowania:**
  - Po wprowadzeniu danych użytkownika następuje ich weryfikacja w bazie danych.
  - Logowanie odbywa się w sposób bezpieczny z użyciem szyfrowania danych.
  - W przypadku błędnych danych wyświetlany jest komunikat o błędzie.



The Login form is titled "Login" in bold black text. Below the title is a section labeled "Login Credentials" in a smaller font. It contains two labels, "Username:" and "Password:", each followed by a white text input field with a light gray border. At the bottom of the form is a prominent blue button with the text "Log in" in white. Below the button, there is a link that says "Don't have an account yet? [Register here.](#)" where the link text is blue and underlined.

(a) Login



The Register form is titled "Register" in bold black text. Below the title is a section labeled "Registration Details" in a smaller font. It contains two labels, "Username:" and "Password:", each followed by a white text input field with a light gray border. At the bottom of the form is a prominent green button with the text "Register" in white. Below the button, there is a link that says "Have an account already? [Log in here.](#)" where the link text is blue and underlined.

(b) Register

Rysunek 3: Widoki stron Login i Register.

# Strona Użytkownika

Strona użytkownika /profile umożliwia każdemu zalogowanemu przegląd swojego profilu oraz wykonywanie akcji zależnych od roli przypisanej do konta. Jej struktura składa się z dwóch głównych elementów:

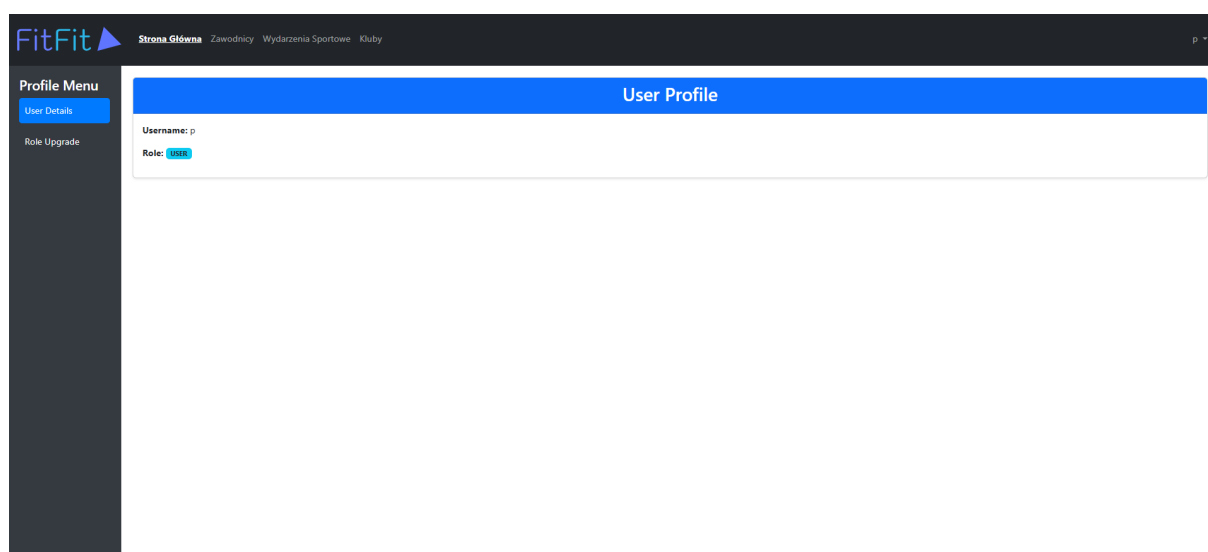
## 1. Główna część profilu

- Na środku ekranu wyświetlane są kluczowe informacje o użytkowniku:
  - **Username** – login użytkownika.
  - **Rola** – aktualna rola przypisana do konta (np. USER, TRENER, ZAWODNIK, PRACOWNIK BIURA, ADMIN).
- Sekcja ta pozwala szybko zapoznać się z aktualnym statusem użytkownika.

## 2. Menu boczne (Sidebar)

- Menu boczne znajduje się po lewej stronie ekranu i zawiera przyciski zależne od roli użytkownika:
  - **Role Upgrade** – przycisk dostępny dla użytkowników z rolami innymi niż ADMIN. Pozwala zgłosić chęć zmiany roli (np. z USER na TRENER).
  - **Admin Panel** – widoczny tylko dla użytkowników z rolą ADMIN.
- Menu dostosowuje się dynamicznie do przypisanej roli, co zapewnia intuicyjność i łatwość użytkowania.

Poniżej przedstawiono widok profilu użytkownika (Rysunek 4), który prezentuje interfejs umożliwiający przegląd kluczowych informacji oraz dostęp do funkcji zależnych od roli użytkownika.



Rysunek 4: Widok profilu użytkownika



## Zakładka "Role Upgrade"

Zakładka **Role Upgrade** umożliwia użytkownikowi zgłoszenie prośby o zmianę swojej aktualnej roli w systemie. Widok zawiera formularz, w którym użytkownik może wprowadzić niezbędne dane do weryfikacji. Na Rysunku 5 przedstawiono interfejs zakładki „Role Upgrade”.

### Elementy interfejsu

- **Nagłówek "Request Role Upgrade":**

- Wyświetlany na niebieskim tle, informuje o funkcji zakładki.
- Zawiera rozwijane menu, które pozwala użytkownikowi wybrać nową rolę, na przykład:
  - \* Trener,
  - \* Zawodnik,
  - \* Pracownik biura.

- **Podstawowe informacje użytkownika:**

- Pola do uzupełnienia:
  - \* *Imię*,
  - \* *Nazwisko*,
  - \* *Numer telefonu*,
  - \* *Email*,
  - \* *Data urodzenia* (z formatem mm/dd/yyyy).

- **Sekcja "Additional Information for Trener":**

- Wyświetlana dynamicznie w zależności od wybranej roli (w przykładzie dla roli Trener).
- Zawiera dodatkowe pola wymagane dla danej roli:
  - \* *Telefon*,
  - \* *PESEL*,
  - \* *Data zatrudnienia*,
  - \* *Typ sportu*,
  - \* *Doświadczenie* (pole tekstowe),
  - \* *Numer licencji*.
- Formularz w tej sekcji zawiera wiele pól, na podstawie których tworzone są wpisy w bazie danych. Dane wprowadzane w poszczególnych polach są przypisywane do różnych encji w bazie:
  - \* **Adres** – przechowywany jako osobna encja odpowiadająca za dane lokalizacyjne,
  - \* **Pracownik** – encja zawierająca szczegóły dotyczące zatrudnienia w organizacji,
  - \* **Trener** – encja przechowująca dane specyficzne dla osób pełniących rolę trenera, takie jak typ sportu czy numer licencji.

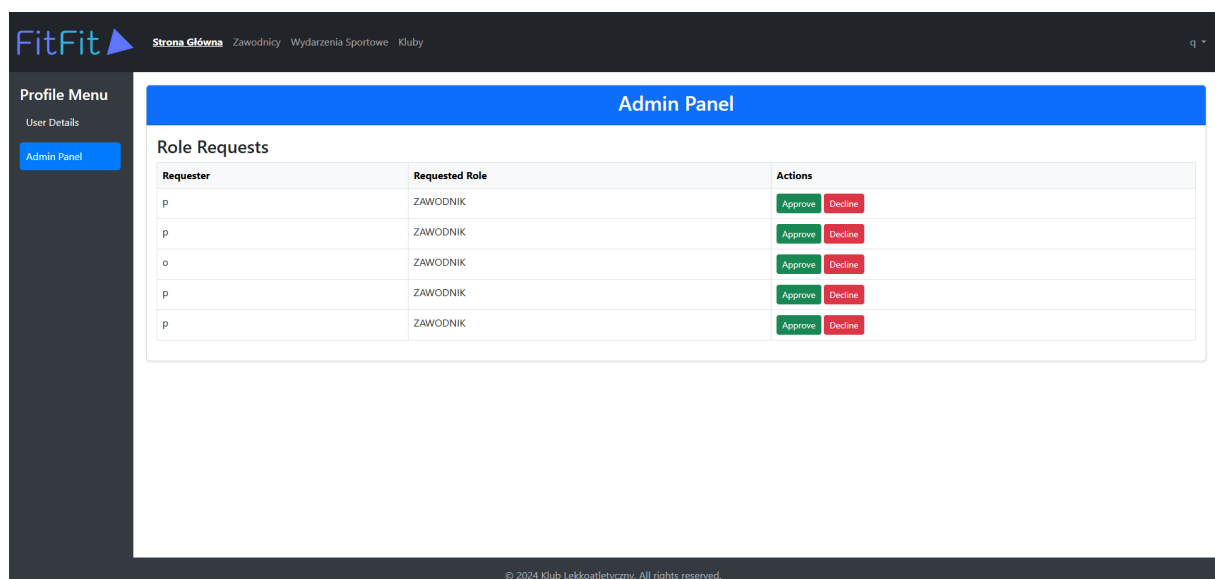


## Oddziaływanie z bazą danych

Działania podejmowane przez administratora mają bezpośredni wpływ na dane w systemie:

- Naciśnięcie przycisku **Approve** powoduje zapisanie zmienionej roli użytkownika w bazie danych. Odpowiedni wpis w tabeli użytkowników jest aktualizowany, przypisując nową rolę do rekordu użytkownika.
- Naciśnięcie przycisku **Decline** skutkuje odrzuceniem wniosku. W bazie danych zostaje zarejestrowana informacja o odrzuceniu zgłoszenia, poprzez zapisanie statusu w oddzielnej kolumnie.

Zakładka ta jest kluczowa dla zapewnienia kontroli i weryfikacji poprawności zmian ról, co zwiększa bezpieczeństwo oraz porządek w systemie.



Rysunek 6: Widok Admin Panel

## 5 Operacje na bazie danych

- Dodawanie danych - rejestracja użytkownika, zgłoszenie zmiany roli.
- Modyfikacja danych - zmiana stanu wniosku o zmianę roli.
- Przeglądanie danych - wydarzenia sportowe.

## 6 Uwagi i podsumowanie

W projekcie przyjęliśmy duże oczekiwania i mieliśmy zamiar je zrealizować, ale faktycznie poznanie technologii zajęło dużo więcej czasu i napotykalismy problemy, które czasami trzeba było rozwiązywać przez parę godzin. Dużo zyskalismy robiąc ten projekt. Między innymi urochomiliśmy silnik bazy danych na odrębnym serwerze łącząc się z nim przez sieć. Cały czas mieliśmy problemy z konwertacją danych, sekwencjami, java script-em. Więc zdecydowanie projekt był ciekawy i użyteczny, mimo że czasochłonny.

- **Zdobyte umiejętności:**

- Praktyczne opanowanie technologii takich jak Java Spring Boot, Hibernate, oraz PostgreSQL, co pozwoliło stworzyć funkcjonalną aplikację współpracującą z bazą danych.
- Poznanie mechanizmów pracy z serwerami baz danych uruchamianymi na oddzielnych maszynach oraz konfiguracji połączeń przez sieć.
- Rozwinięcie umiejętności debugowania i rozwiązywania problemów, takich jak obsługa sekwencji w bazach danych czy dynamiczne przetwarzanie danych w JavaScript.

- **Wyzwania i ich rozwiązania:**

- Problemy z konwersją danych i zarządzaniem sekwencjami zostały rozwiązane poprzez dokładną analizę dokumentacji oraz iteracyjne testy.
- JavaScript, choć początkowo problematyczny, pozwolił finalnie na stworzenie dynamicznych i intuicyjnych elementów frontendowych.

- **Osiągnięcia:**

- Stworzenie w pełni funkcjonalnej aplikacji, która umożliwia wykonywanie operacji CRUD i zarządzanie danymi użytkowników oraz grup.
- Integracja aplikacji z systemem ról użytkowników, zapewniająca dostęp do funkcjonalności zgodnie z ich uprawnieniami.

- **Czasochłonność projektu:**

- Proces nauki nowych technologii, takich jak Java Spring czy Spring Security, wymagał dużego nakładu pracy i czasu, jednak przyniósł wymierne korzyści w postaci lepszego zrozumienia zaawansowanych narzędzi.

- **Znaczenie projektu:**

- Realizacja tego projektu pozwoliła zdobyć praktyczne doświadczenie w projektowaniu i zarządzaniu aplikacjami współpracującymi z bazami danych, co znacząco wzbogaciło wiedzę zespołu i umiejętności praktyczne.