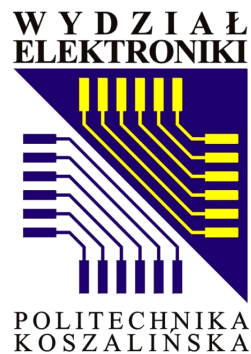


Zastosowanie Programowania Obiektowego
Informatyka III Semestr

Piłka nożna - Katalog (Aplikacja PHP)

Jakub Achtelik, Oliwier Budnik



Politechnika Koszalińska
Koszalin 2023

Spis Treści

1	Koncepcja i funkcjonalność aplikacji	2
1.1	Założenia przyjęte dla tworzenia aplikacji:	2
1.1.1	Środowisko uruchomieniowe:	2
1.2	Założenia przyjęte dla tworzenia aplikacji	3
1.2.1	Cechy	3
1.2.2	Ograniczenia:	3
1.3	Narzędzia programistyczne:	4
1.4	Wykaz funkcjonalności aplikacji	5
1.5	Prototyp	6
2	Diagram przypadków użycia	8
2.1	Rysunek (diagram)	8
2.2	Opis	9
3	Diagram klas	10
3.1	Rysunek (diagram)	10
3.2	Opis przeznaczenia klas	10
3.3	Models	11
3.3.1	Aplikacja	11
3.3.2	BazaDanych	11
3.3.3	ZapytaniaSql	11
3.3.4	FiltrowanieSql	11
3.3.5	OperacjePilkarzy	11
3.3.6	WyswietlaniePilkarzy	11
3.3.7	PobieraczObrazowWikipedia	11
3.4	Controllers	11
3.4.1	Autoryzacja	11
3.4.2	FiltrowanieKontroler	11
3.4.3	KontrolerDanych	11
3.4.4	PilkarzPost	11
3.4.5	Autoryzacja	11
3.4.6	ZarzadzaniePilkarzami	11
3.5	Views	11
3.5.1	StronaHtml	11
3.5.2	SzablonHtml	12
3.6	Helpers	12
3.6.1	BazaDanychHelper	12
3.6.2	FormularzHelper	12
3.7	Projekt	12
3.8	FileLoader	12
4	Projekt Bazy Danych	13
4.1	Model bazy danych - relacyjna SQL	13
4.2	Opis	13
4.2.1	Ustawienie uprawnień dla użytkownika	13
4.3	Rysunek	13

5	Opis działania i obsługi aplikacji	14
5.1	Przewodnik	14
5.1.1	Strona internetowa	14
5.2	Wyświetlanie oraz filtrowanie piłkarzy	14
5.3	Panel Logowania	14
5.4	Panel Administracyjny	15
5.5	Modyfikowanie danych na temat piłkarzy	16
5.5.1	Dodawanie piłkarza	16
5.5.2	Usuwanie piłkarza	16
5.5.3	Edycja piłkarza	16
6	Wnioski	18
6.1	Podsumowanie	18
6.2	Podział pracy	18
6.2.1	Zadania wykonane przez U-20019	18
6.2.2	Zadania wykonane przez U-20041	18
7	Bibliografia - źródła	19

1 Koncepcja i funkcjonalność aplikacji

1.1 Założenia przyjęte dla tworzenia aplikacji:

Rodzaj aplikacji: Aplikacja internetowa (architektura klient-serwer)
Aplikacja uruchamiana lokalnie na komputerach oraz zdalnie na serwerze VPS.
Aplikacja jest dostosowana również do urządzeń mobilnych.

Środowisko lokalne: Windows 10, Ubuntu 22.04 LTS

Środowisko produkcyjne (serwer): VPS Linux (Ubuntu 20.04 LTS):

1.1.1 Środowisko uruchomieniowe:

Dla Windows:

- XAMPP 8.2.4 (serwer HTTP Apache, Serwer bazy danych MariaDB, interpreter PHP)

Dla Ubuntu:

- PHP 8.1.2-1ubuntu2.14 Development Server (serwer HTTP + interpreter PHP)
- mysql-server ver 8.0.34-0ubuntu0.22.04.1 (serwer bazy danych MySQL)

Dla VPS:

- Serwer HTTP Apache2
- Serwer MySQL – MariaDB
- Interpreter PHP 8+

1.2 Założenia przyjęte dla tworzenia aplikacji

1.2.1 Cechy

1. Zastosowanie statycznego typowania (zmiennych, funkcji, metod, pól klasy), podobnie jak w językach C/C++, Java, C#. Jest to bardziej przewidywalne i pozwala narzucić określony typ np. zwracanej zmiennej, aby uniknąć wielu błędów. Domyślnie PHP nie wymaga statycznego typowania.
2. Podział projektu na wiele plików według struktury MVC Model-View-Controller (pol. Model-Widok-Kontroler):
 - Model jest pewną reprezentacją problemu bądź logiki aplikacji.
 - Widok opisuje, jak wyświetlić pewną część modelu w ramach interfejsu użytkownika.
 - Kontroler przyjmuje dane wejściowe od użytkownika i reaguje na jego poczynania
3. Logika aplikacji będzie zawarta w sposób obiektowy w klasach, każda klasa to osobny plik.

1.2.2 Ograniczenia:

- PHP jest podatny na pewne rodzaje ataków, takich jak na przykład wstrzykiwanie SQL, dlatego bezpieczeństwo aplikacji nie jest na najwyższym możliwym poziomie i szczegółowa konfiguracja zabezpieczeń nie jest łatwa do wdrożenia w krótkim czasie
- PHP jest językiem interpretowanym dlatego wydajność w stosunku do języków kompilowanych jest niższa
- PHP nie posiada wszystkich elementów obiektowych znanych z innych języków
- Ograniczony czas, przez co nie można zawrzeć wszystkich celów w wzorcowy sposób zgodny w 100% z dokumentacją
- Ograniczenie aktualnej wiedzy, przez co niektóre elementy projektu mogą stanowić wyzwanie
- W naszej aplikacji proces logowania ogranicza się do jednego super użytkownika, którego poświadczenia są statycznie zapisane w pliku **KonfiguracjaApp.php** jako tzw. **plain text** (niezahaszowane hasła przechowywane jako zwykły, łatwy do odczytania i przechwycenia tekst), jest to zła praktyka jednak chcieliśmy uprościć tą część aplikacji ponieważ oddala się ona dość znacząco od pierwotnego tematu, a wymaga dość dużo nadprogramowej pracy.

1.3 Narzędzia programistyczne:

Język: PHP 8+ OOP

Dodatkowe biblioteki: mysqli (łączenie się z bazą danych)

Dodatkowe technologie: HTML, CSS, JavaScript, MySQL, FontAwesone(ikonki)

Visual Studio Code + PHP Code Extensions: IDE (Zintegrowane środowisko programistyczne)

XAMPP: środowisko uruchomieniowe dla Windows

GIT – System Kontroli Wersji

phpMyAdmin - graficzna nakładka na serwer MySQL, ułatwiająca operacje na bazie danych

Brave, Google Chrome – przeglądarka internetowa posiadająca narzędzia Chrome DevTools

Pakiet **make** – automatyzacja poleceń w terminalu

FileZilla – klient FTP

LaTeX - skład tekstu do sprawozdania

PlantUML - narzędzie do tworzenia rysunków i schematów, z poziomu pików tekstowych

Trello – zarządzanie zadaniami w zespole

Figma – Prototypowanie wyglądu aplikacji

GIMP 2.10.34 – Prosta edycja oraz tworzenie grafiki rastrowej

1.4 Wykaz funkcjonalności aplikacji

Interfejs webowy, zarządzanie bazą danych z poziomu przeglądarki internetowej:

- edycje, usuwanie, dodawanie nowego piłkarza,
- sortowanie oraz wyświetlanie zdjęć,
- wyszukiwanie po nazwisku, imieniu itp.
- filtrowanie szczegółowe po np. kraju, pozycji itp.
- logowanie oraz autoryzacja użytkownika przeglądającego aplikację
- dodawanie/edycje, zdjęcia piłkarza

Użytkownik może za pomocą przeglądarki internetowej:

- połączyć się z serwerem na którym hostowana jest aplikacja
- zalogować się do panelu poprzez formularz logowania, uzyskać autoryzację
- Panel umożliwia przeglądanie katalogu piłkarzy w przystępnej formie oraz inne operacje (edycja, usuwaniem, filtrowanie itp.).
- Użytkownik końcowy (klient) nie musi posiadać znajomości obsługi relacyjnej bazy danych aby w intuicyjny sposób zarządzać aplikacją.

1.5 Prototyp

Prototyp graficzny interfejsu użytkownika. Wykonane w programie Figma.



Rysunek 1: Widok strony głównej



Rysunek 2: Widok ekranu logowania

The screenshot shows the 'Dodawanie' form in the 'Piłka Nożna Katalog' application. The header bar is gold and contains the title 'Piłka Nożna Katalog'. Below the header is a navigation bar with four buttons: 'Strona Główna', 'Dodaj', 'Szukaj', and 'Wyloguj'. The main content area is light gray and contains the title 'Dodawanie' in bold. Below the title are six input fields with labels: 'imię', 'nazwisko', 'kraj', 'pozycja', 'wzrost', and 'dodaj'. The 'dodaj' button is highlighted in gold.

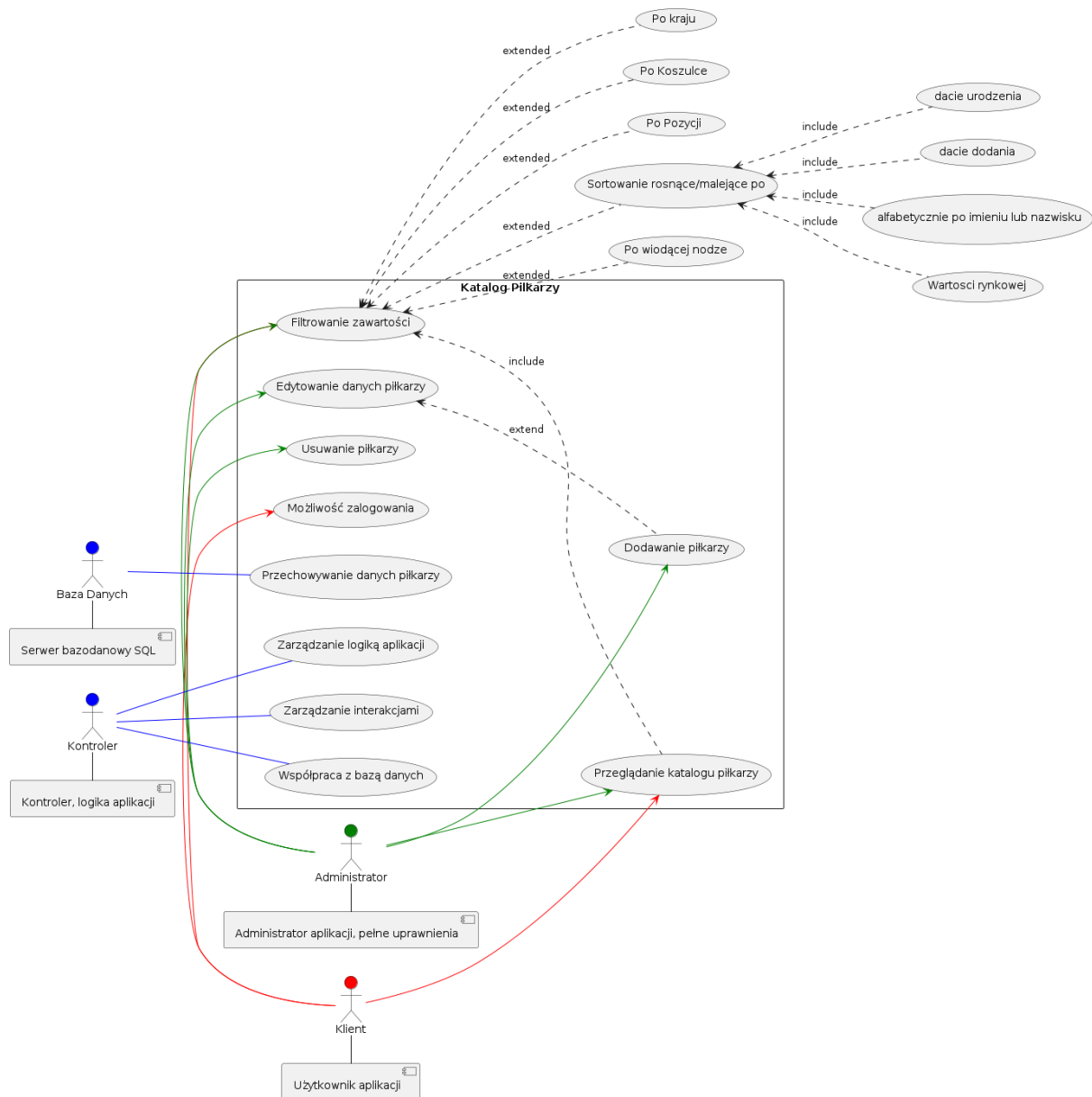
Rysunek 3: Widok strony głównej

The screenshot shows the 'Edycja' form in the 'Piłka Nożna Katalog' application. The header bar is gold and contains the title 'Piłka Nożna Katalog'. Below the header is a navigation bar with four buttons: 'Strona Główna', 'Dodaj', 'Szukaj', and 'Wyloguj'. The main content area is light gray and contains the title 'Edycja' in bold. Below the title are six input fields with labels: 'Lionel', 'Messi', 'Argentyna', 'napastnik', '1.62m', and 'edycja'. The 'edycja' button is highlighted in gold.

Rysunek 4: Widok strony głównej

2 Diagram przypadków użycia

2.1 Rysunek (diagram)



Rysunek 5: Diagram przypadków użycia

2.2 Opis

Aktorami osobowymi są: Administrator oraz Klient, są to odbiorcy aplikacji

Aktorami bezosobowymi, abstrakcyjnymi są:

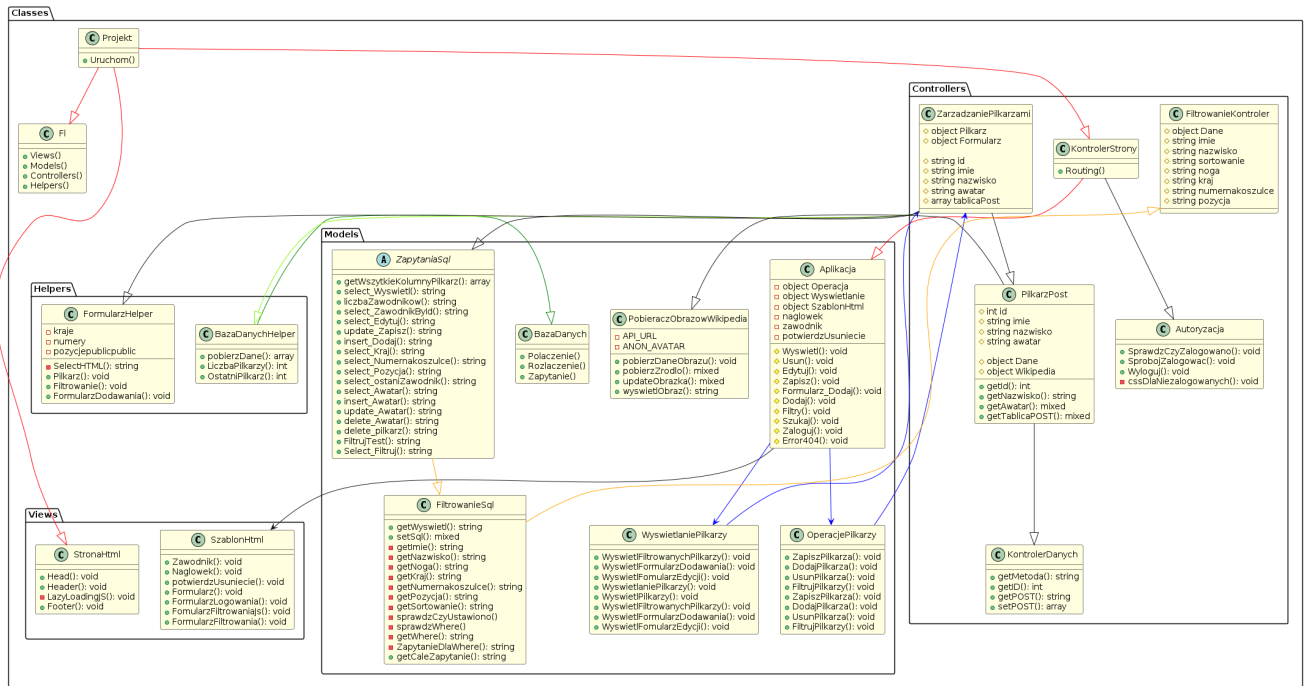
- Kontroler – reprezentuje logikę aplikacji
- BazaDanych – reprezentuje miejsce, gdzie informacje są przechowywane

Podstawowe operacje aktorów:

- Klient -> (Przeglądanie katalogu piłkarzy)
- Klient -> (Filtrowanie zawartości)
- Klient -> (Możliwość zalogowania)
- Administrator -> (Przeglądanie katalogu piłkarzy)
- Administrator -> (Filtrowanie zawartości)
- Administrator -> (Dodawanie piłkarzy)
- Administrator -> (Edytowanie danych piłkarzy)
- Administrator -> (Usuwanie piłkarzy)
- BazaDanych – (Przechowywanie danych piłkarzy)
- Kontroler – (Zarządzanie interakcjami)
- Kontroler – (Współpraca z bazą danych)
- Kontroler – (Zarządzanie logiką aplikacji)

3 Diagram klas

3.1 Rysunek (diagram)



Rysunek 6: Diagram klas

3.2 Opis przeznaczenia klas

Klasy są segregowane według modelu **MVC** i pomocniczych klas opisanych jako **Helpers**

3.3 Models

3.3.1 Aplikacja

3.3.2 BazaDanych

3.3.3 ZapytaniaSql

3.3.4 FiltrowanieSql

3.3.5 OperacjePilkarzy

3.3.6 WyświetlaniePilkarzy

3.3.7 PobieraczObrazowWikipedia

Klasa za pomocą otwartego API Wikipedia, pobiera adres URL do głównego zdjęcia piłkarza z artykułu na Wikipedii. Zdjęcie jest pobierane podczas dodawania nowego piłkarza bądź jego edycji i zapisywana w bazie danych w tabeli "awatar" w postaci linku.

3.4 Controllers

3.4.1 Autoryzacja

3.4.2 FiltrowanieKontroler

3.4.3 KontrolerDanych

Ta klasa służy do odbierania danych od użytkownika z poziomu aplikacji, używając metod GET (parametry z linku) oraz POST (niejawnych, przesyłanych z formularzy.)

3.4.4 PilkarzPost

3.4.5 Autoryzacja

3.4.6 ZarządzaniePilkarzami

3.5 Views

3.5.1 StronaHtml

Zawiera szkielet strony HTML, taki jak sekcja <head> <body> <footer> czy <header>. Dzięki temu rozwiązaniu można ustawić Podstawowe dane takie jak tytuł strony, autorów, w kilku miejscach na stronie, wczytywana z pliku konfiguracyjnego **KonfiguracjaApp.php**

3.5.2 SzablonHtml

3.6 Helpers

3.6.1 BazaDanychHelper

3.6.2 FormularzHelper

3.7 Projekt

```
1 <?php
2 // klasa do uruchamiania projektu
3
4 namespace Pilkanozna;
5
6 use Pilkanozna\Controller\KontrolerStrony;
7 use Pilkanozna\Views\StronaHtml;
8
9
10 final class Projekt
11 {
12     public static function Uruchom(): void
13     {
14         $Aplikacja = new KontrolerStrony;
15         $Strona = new StronaHtml;
16
17
18         $Strona->Head();
19         $Strona->Header();
20         $Aplikacja->Routing();
21         $Strona->Footer();
22     }
23
24
25 }
```

3.8 FileLoader

4 Projekt Bazy Danych

4.1 Model bazy danych - relacyjna SQL

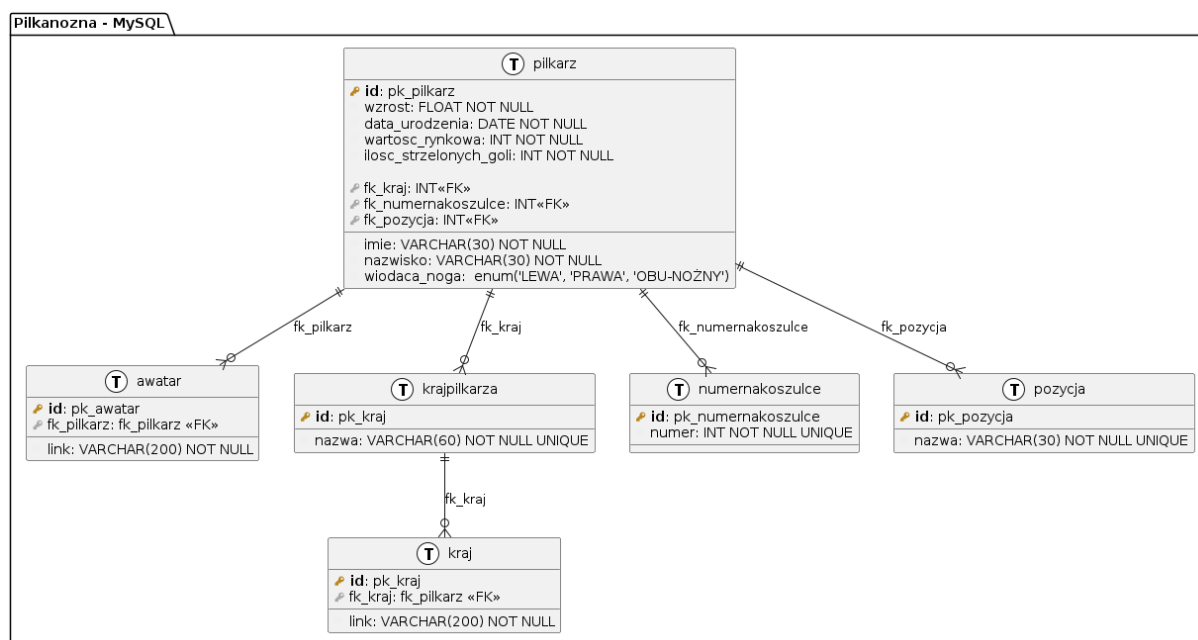
Baza danych jest relacyjna oraz obsługiwana jest na serwerze SQL.

4.2 Opis

4.2.1 Ustawienie uprawnień dla użytkownika

```
1 CREATE USER 'projekt'@'localhost' IDENTIFIED BY 'Pracownia107!';
2 GRANT ALL PRIVILEGES ON pilkanozna.* TO 'projekt'@'localhost';
```

4.3 Rysunek



Rysunek 7: Diagram tabel bazy danych pilkanozna

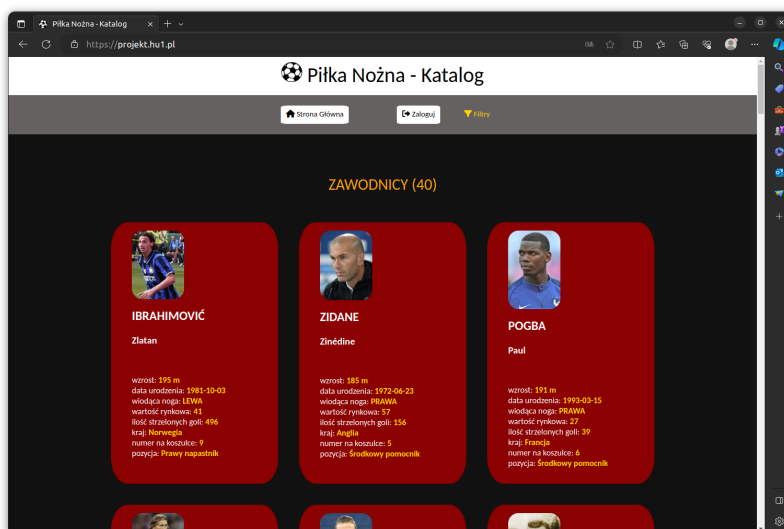
5 Opis działania i obsługi aplikacji

5.1 Przewodnik

5.1.1 Strona internetowa

Aplikacja znajduje się na zewnętrznym serwerze VPS i można ją przetestować bez uruchamiania jej na swoim komputerze, dostępna jest pod adresem:

- <https://projekt.hu1.pl/>



Rysunek 8: Strona Główna

5.2 Wyświetlanie oraz filtrowanie piłkarzy

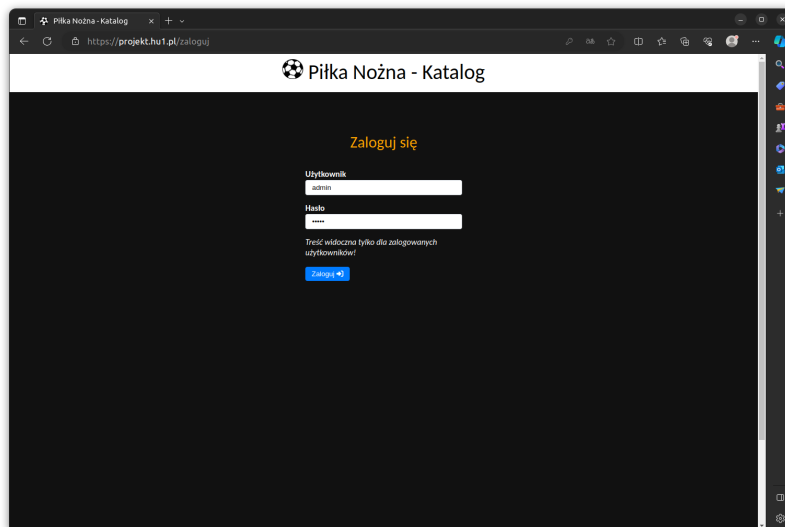
5.3 Panel Logowania

Panel logowania dostępny jest pod adresem:

- <https://projekt.hu1.pl/zaloguj>

Dane do logowania:

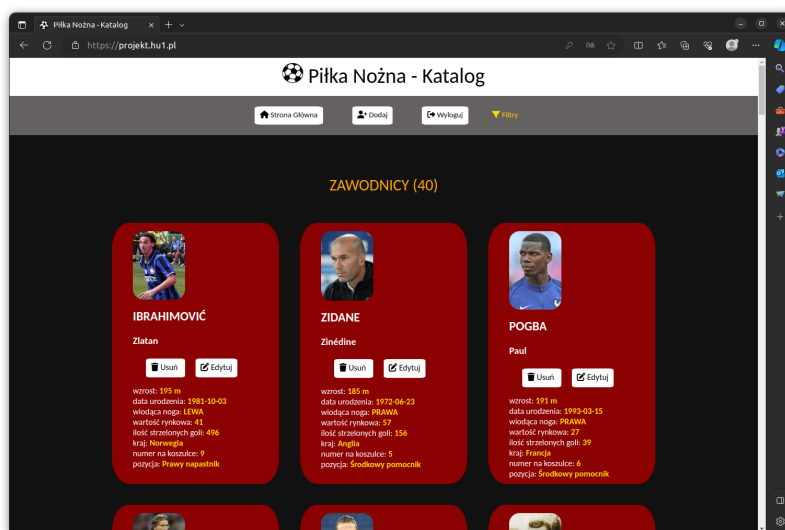
- Użytkownik: **admin**
- Hasło: **admin**



Rysunek 9: Panel logowania

5.4 Panel Administracyjny

Jako administrator użytkownik ma podniesione uprawnienia i dodatkową zakładkę *Dodaj* oraz przycisk *Edytuj* lub *Usuń* nad zdjęciem piłkarza.

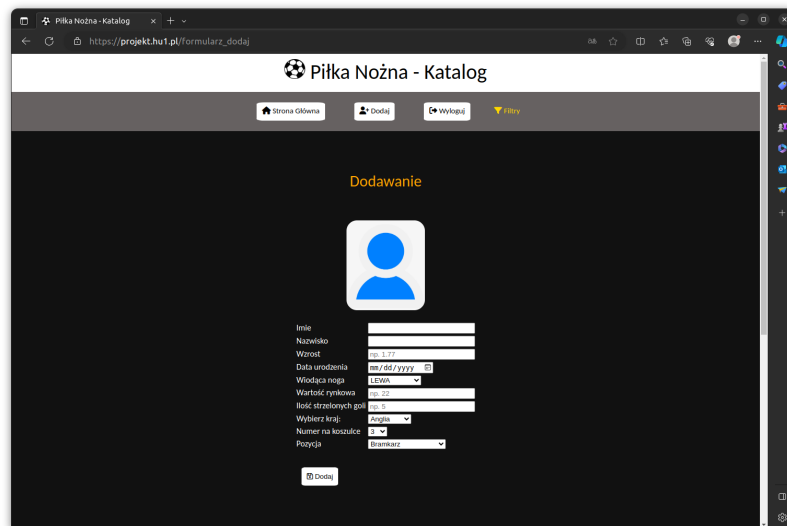


Rysunek 10: Panel Administracyjny

5.5 Modyfikowanie danych na temat piłkarzy

5.5.1 Dodawanie piłkarza

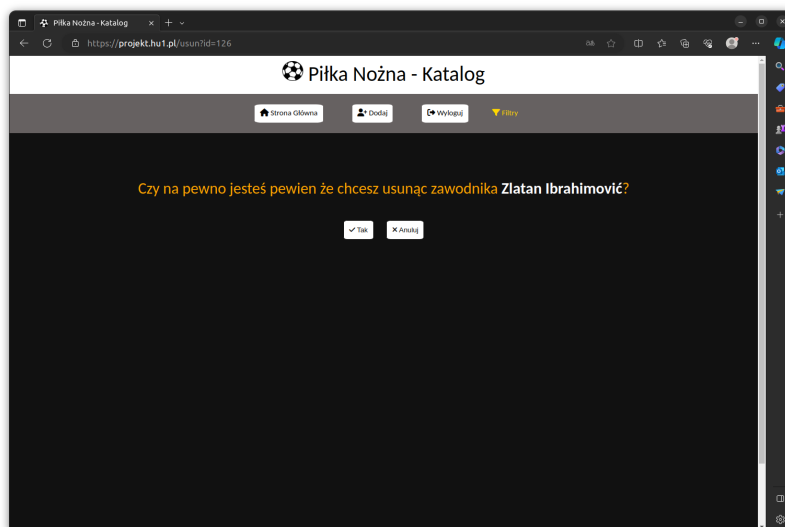
Na podstawie **Imienia** oraz **Nazwiska** zdjęcie piłkarza jest pobierane z serwisu Wikipedia, jeżeli piłkarz ma tam swój artykuł wraz ze zdjęciem (w 90 % przypadkach posiada), w innym wypadku ustawia domyślne zdjęcie anonimowego użytkownika.



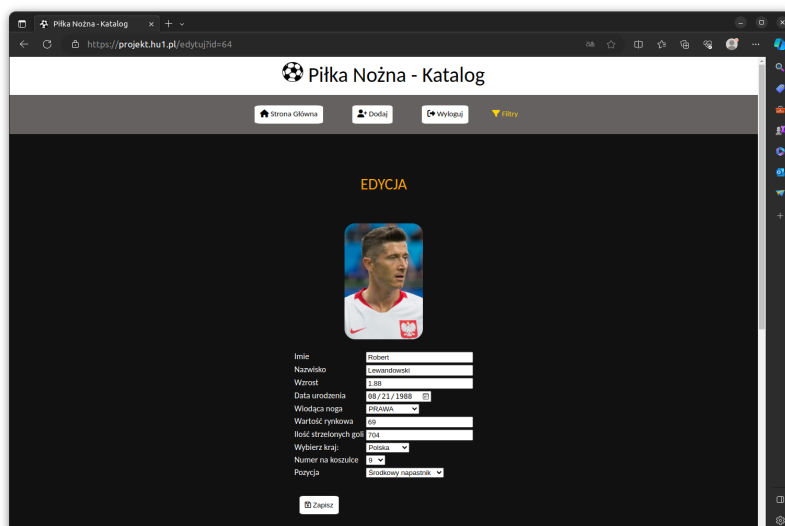
Rysunek 11: Formularz, który umożliwia dodanie nowego Piłkarza

5.5.2 Usuwanie piłkarza

5.5.3 Edycja piłkarza



Rysunek 12: Potwierdzenie usunięcia Piłkarza



Rysunek 13: Panel służący do edycji informacji na temat piłkarza

6 Wnioski

6.1 Podsumowanie

6.2 Podział pracy

U-20019 - Jakub Achtelik

U-20041 - Oliwier Budnik

6.2.1 Zadania wykonane przez U-20019

- Zaprojektowanie oraz wykonanie bazy danych
- Konstrukcja zapytań oraz klas, które służą do łączenia z bazą danych
- Konfiguracja usług na serwerze VPS
- Skład tekstu sprawozdania w LaTeX
- Implementacja połączenia API Wikipedia z naszą aplikacją
- Proste grafiki w programie GIMP
-

6.2.2 Zadania wykonane przez U-20041

- Prototyp wyglądu aplikacji
- Implementacja autoryzacji oraz formularz logowania
- Formularz edycji, dodawania i przekazanie tego do jednej tablicy
- Wyświetlanie zwróconych wyników z bazy danych w formie Szablonów HTML
- Szablon strony HTML oraz dostosowanie wyglądu w CSS
- Routing strony, czyli ustawienie klasy, która na podstawie /linku wykonuje określoną operację

7 Bibliografia - źródła

Literatura

- [1] php.net - Oficjalna Dokumentacja PHP
<https://www.php.net/docs.php>
- [2] Medium - Informacje na temat MVC w PHP
<https://medium.com/@iamjoestack/how-to-build-a-custom-php-mvc-framework-e5a23da8f73d>
- [3] Konfiguracja Serwera SQL - Ubuntu VPS
<https://ubuntu.com/server/docs/databases-mysql>
- [4] Konfiguracja Serwera HTTP - Apache 2
<https://httpd.apache.org/docs/2.4/>
- [5] Tworzenie diagramów i schematów - PlantUML
<https://plantuml.com/>
- [6] Pobieranie obrazów z API Wikipedia
<https://stackoverflow.com/questions/8363531/accessing-main-picture-of-wikipedia-page-by-api>