

# Aplikacja webowa do porównywania wyników bukmacherskich

## Wstęp

Celem projektu było utworzenie aplikacji MVC w technologii .NET, która umożliwiałaby użytkownikom porównywanie wyników bukmacherskich

## Opis elementów projektów








### 1. Modele danych

W projekcie wykorzystuje dwa modele danych:

- Player odpowiadający za danego gracza
- Group odpowiadającą za grupę graczy

#### 1.1 Player

Zadaniem klasy Player jest przechwytywanie informacji o każdym z graczy

```
public class Player
{
     11 usages
    public int ID { get; set; }
     12 usages
    public string Name { get; set; }
     9 usages
    public int GroupID { get; set; }
     9 usages
    public Group? Group { get; set; }
     13 usages
    public int SuccessfulBets { get; set; }
     13 usages
    public int AllAttempts { get; set; }
     9 usages
    public float Score { get; set; }
}
```

Ta klasa posiada następujące pola:

- ID - unikalne Id dla każdego playera
- Name - Nazwę danego Playera
- GroupID - Id grupy do której należy
- Group - pomocnicze pole grupy które ułatwia dostęp do grupy danego Playera

- SuccessfulBets - liczba trafionych zakładów
- AllAttempts - liczba wszystkich zakładów
- Score - SuccessfulBets / AllAttempts

## 1.2 Group

Zadaniem klasy Group jest przechwytywanie informacji o każdej z grup graczy

```
public class Group
{
    16 usages
    public int ID { get; set; }
    13 usages
    public string Name { get; set; }

    public IList<Player>? Players { get; set; }
}
```

Ta klasa posiada następujące pola:

- ID - unikalne Id dla każdej grupy
- Name - Nazwę danej grupy
- Players - listę playerów

## 2. Widoki

W projekcie wykorzystywane są następujące widoki:

### 2.1 Home

Widok *Home* to podstawowy widok w którym można przeglądać rankingi dla poszczególnych grup

### 2.2 Player

Widok *Player* umożliwia wykorzystanie operacji CRUD na klasie Player

### 2.3 Group

Widok *Group* umożliwia wykorzystanie operacji CRUD na klasie Group

## Użytkowanie Aplikacji

[Home](#) [Privacy](#) [Player](#) [Group](#)

Na głównej stronie jest widoczna lista poszczególnych widoków, można między nimi nawigować poprzez klikanie w poszczególny widok. Oprócz tego na głównej stronie wyświetlany jest ranking dla wybranej grupy:

# Ranking

Group 1 Group 2

Name	Score
Player 1	0,07692308

Poprzez kliknięciu w odpowiednią grupę wyświetla nam się ranking dla danej grupy. W rankingu mamy nazwę danego gracza oraz jego score. Gracze są posortowani według score.

Po kliknięciu na pole Group w pasku na górze, zostajemy przekierowani do widoku danej grupy gdzie możemy przeprowadzić operację CRUD.

## Group CRUD

[Create New](#)

Name	
Group 1	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
Group 2	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>

Analogicznie wygląda sytuacja dla Playera.

## Player CRUD

[Create New](#)

Name	Group	SuccessfulBets	AllAttempts	
Player 1	Group 1	1	13	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
Player 4	Group 1	1	4	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>

W przypadku Playera jednak nie możemy ustawić pola Score, gdyż jest ono wyliczane na podstawie trafionych oraz wszystkich obstawionych zakładów. Ułatwione również zostało przypisywanie do poszczególnej grupy

# Edit

## Player

---

Name

Group Name

AllAttempts

[Back to List](#)

Nie trzeba wpisywać ID grupy, należy wybrać grupę z listy grup która posiada nazwy, a Group ID zostanie ustawione automatycznie.