Horse Race

**Dokumentacja projektu**

**Autor:** Paulina Woźnica   
**Repozytorium:** https://github.com/PaullWo/STUDIA-HorseRace

02.06.2025

**Wprowadzenie**

Projekt „HorseRace” to aplikacja webowa stworzona w technologii ASP.NET Core MVC, której głównym celem jest symulacja gry wyścigowej z udziałem koni. Celem gry jest rozwijanie i ściganie się końmi, zarządzanie nimi, a także wykonywanie zadań i interakcja z elementami gry. Admin może dodawać konie, zarządzać nimi, a także tworzyć i przeprowadzać wyścigi. Dane przechowywane są w relacyjnej bazie danych z użyciem Entity Framework Core.

**Podział prac**

Projekt został wykonany samodzielnie.

**Funkcjonalności**

1. **Logowanie i rejestracja użytkowników**

Użytkownik może się zarejestrować i zalogować do systemu. Jeden z użytkowników to specjalny admin o nazwie „Grand Master”, który posiada uprawnienia do zarządzania końmi i wyścigami.

1. **Panel administracyjny:**
   1. Dodawanie, edytowanie, usuwanie i przeglądanie koni
   2. Tworzenie wyścigów z wyborem 5 koni
   3. Przegląd listy wyścigów wraz ze szczegółami (koszt, nagroda, poziom trudności)
   4. Możliwość przeglądania szczegółów wyścigu wraz z listą uczestniczących koni
   5. **Dynamiczne ładowanie koni z zewnętrznych bibliotek (DLL)**
   6. Walidacja danych
   7. Obsługa enumów (Umaszczenie, PoziomTrudnosci)
   8. Wyświetlanie zdjęcia konia na podstawie umaszczenia
   9. Wyszukiwanie, sortowanie i stronicowanie listy koni
   10. Prosta statystyka wyścigów
2. **Panel użytkownika:**
   1. Wyścigi konne

Użytkownik może dołączyć do wyścigu, o ile posiada przynajmniej jednego konia oraz wystarczającą liczbę złotych podków.

**Wyścig:**

- Odbywa się automatycznie (symulacja co 2 sekundy),

- Konie poruszają się zależnie od swoich statystyk,

- Losowane są przeszkody,

- Gracz może użyć „benzyny” (przyspieszenie) oraz „smaru” (spowolnienie przeciwnika),

- Wyświetlane są komunikaty o wydarzeniach,

- Na końcu następuje przekierowanie do widoku z wynikami

* 1. Praca:

Użytkownik może wykonywać pracę co 60 minut (np. w stajni Grand Mastera). Po wykonaniu pracy losowany jest komunikat oraz nagroda w postaci złotych podków.

* 1. Komunikaty i interakcje

1. **Interfejs graficzny z użyciem CSS**

**Dynamiczne ładowanie koni z zewnętrznych bibliotek (DLL)**

Projekt umożliwia wczytywanie pakietów koni z wtyczek DLL znajdujących się w folderze `Plugins`. Mechanizm:

- Wspólna biblioteka `BibliotekaWspolna.dll` definiuje klasę `KonPodstawowy`, interfejs `IListaKoni`, atrybut `InfoAttribute` oraz enum `Umaszczenie

- Każdy plugin implementuje `IListaKoni` i zawiera listę koni o określonej tematyce (np. GKlasa, Lato)

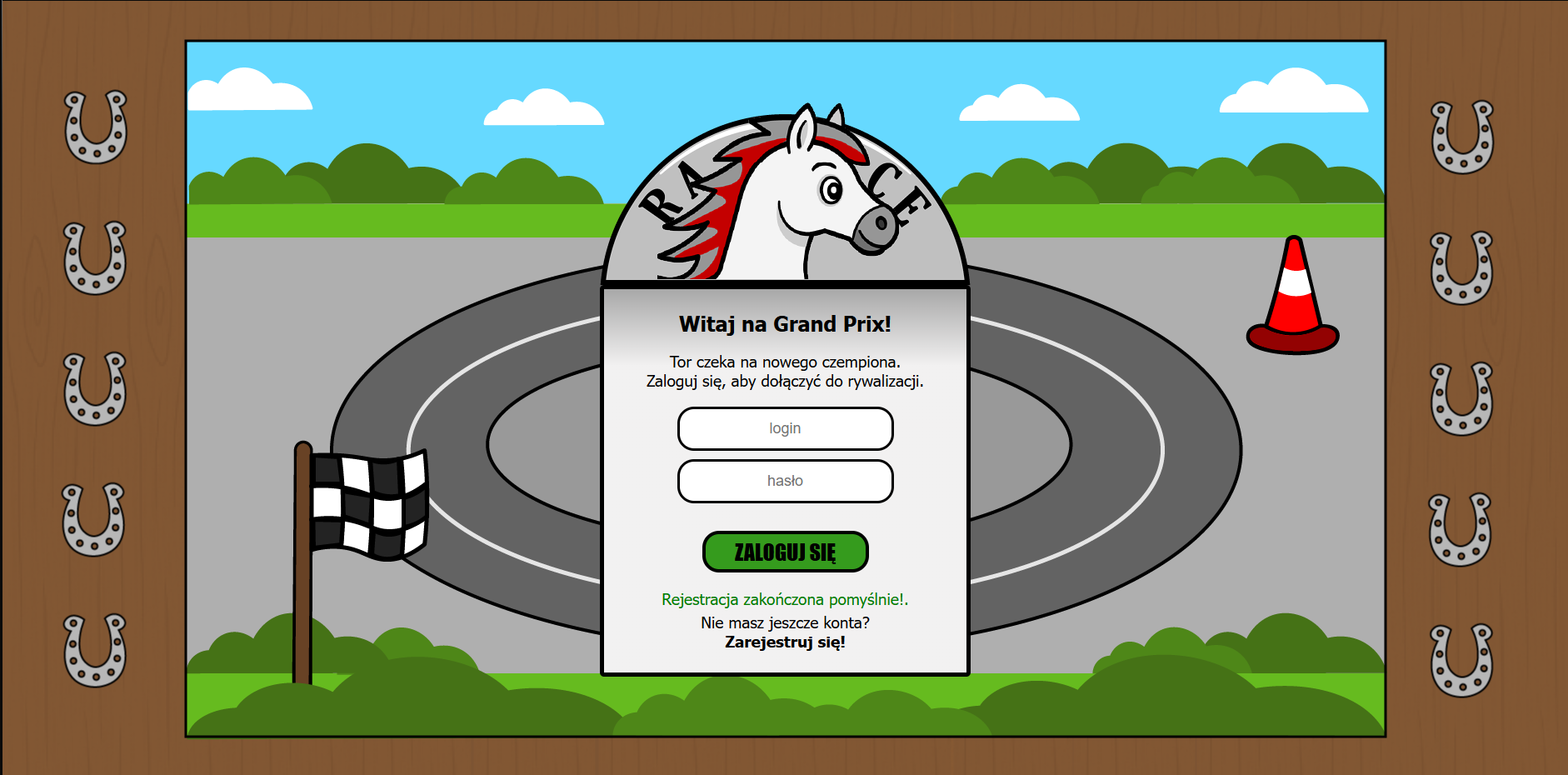
- Admin może wczytać konie z DLL – są one automatycznie dodawane do bazy danych (jeśli nie istnieją identyczne)

- W konsoli zapisywany jest komunikat zawierający nazwę pakietu i liczbę załadowanych koni

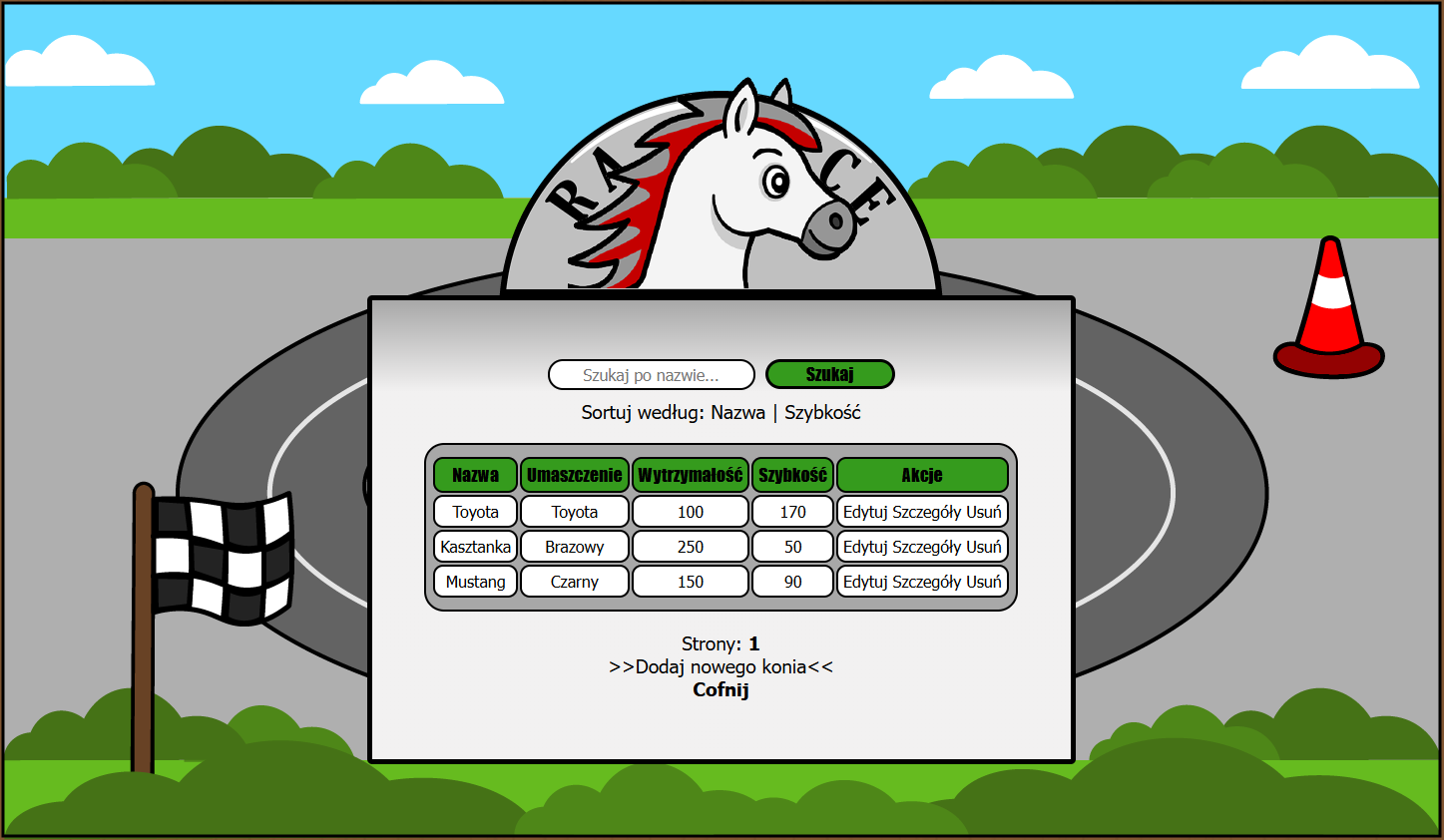
**Obrazki i zwięzłe opisy najciekawszych elementów aplikacji**

1. Panel logowania

Widok umożliwiający logowanie i rejestracje użytkowników.



1. Lista koni z wyszukiwarką i sortowaniem  
   Widok listy koni umożliwia wyszukiwanie po nazwie oraz sortowanie według szybkości lub nazw.



1. Formularz dodawania wyścigu  
   Formularz pozwala wprowadzić dane o wyścigu oraz wybrać dokładnie 5 koni biorących udział.



1. Szczegóły wyścigu z kontekstową nawigacją  
   Widok zawiera dane o wyścigu i umożliwia szybki podgląd szczegółów koni.



1. Statystyka – średnia szybkość koni  
   Prosty ekran prezentujący statystykę wyścigów w systemie.



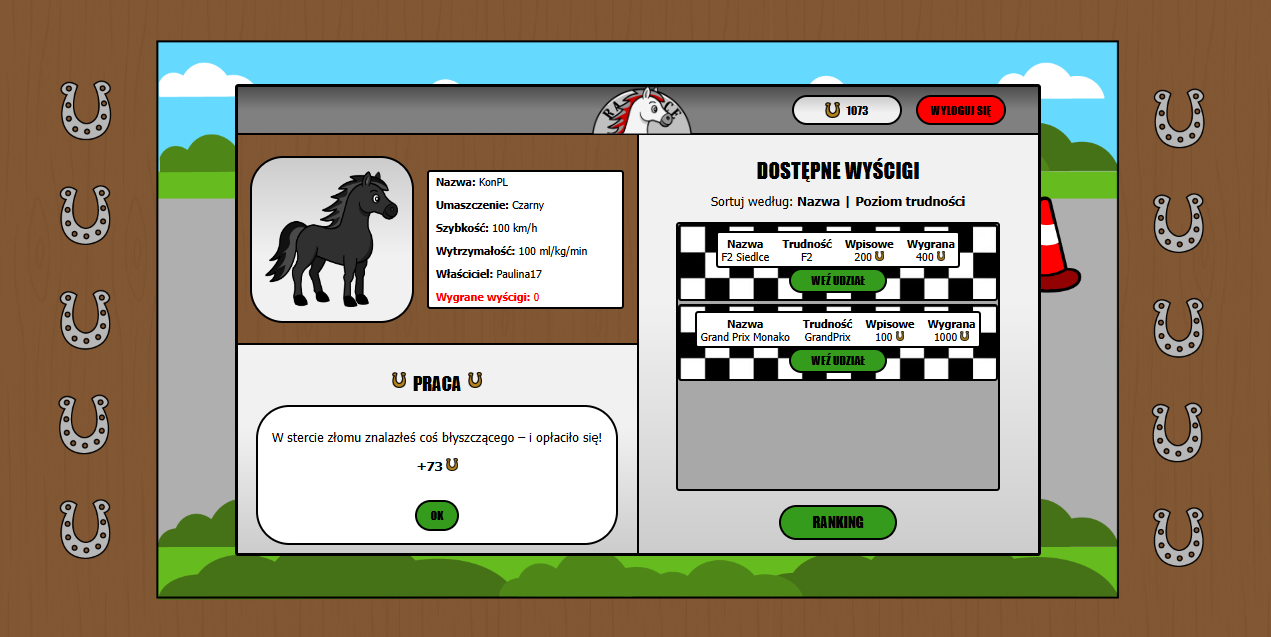
1. Wybór konia

Nowy użytkownik ma możliwość wyboru konia.



1. Okno główne użytkownika

Gracz ma możliwość poglądu statystyk konia, wykonani prostych prac, oraz przegląd dostępnych wyścigów.



1. Wyścig

Podczas wyścigu gracz musi przeskakiwać losowo pojawiające się przeszkody, ma również dostępne dwa boosty. Gra aktualizowana jest automatycznie co 2 sek.

Obraz zawierający zrzut ekranu, tekst, Oprogramowanie multimedialne, oprogramowanie

Zawartość wygenerowana przez AI może być niepoprawna.

1. Ranking

Kolory koni adekwatne do umaszczenia.



**Diagram klas**

