** Uniwersytet Rzeszowski**

**Kolegium nauk przyrodniczych**

**Praca projektowa technologie internetowe:**

**Temat projektu :** Quiz Game – wiedza ogólna JavaScript

**Prowadzący zajęcia:**

mgr. inż. Wojciech Gałka

**Przedmiot:**

Technologie internetowe

**Wykonawca:**

Matykowski Patryk, 125143

**Kierunek:**

Informatyka, grupa lab 2

Rzeszów 2023/2024

Spis treści:

1. Wstęp
2. Zdefiniowanie problemu do realizacji
3. Propozycja rozwiązania problemu
4. Testowanie
5. Dokumentacja kodu źródłowego
6. Linki do źródeł
7. Raport SEO

**1. Wstęp**

Tytuł projektu: Quiz Game – wiedza ogólna JavaScript. Gra typu quiz z dziesięcioma pytaniami przy użyciu języka Javascript i elementów HTML i CSS.

Projekt składa się z jednej strony. Strona została zdefiniowana przy pomocy HTML

i wystylizowana z użyciem dedykowanej dla stron pliku CSS. Strona ta korzysta z JavaScriptu

w celu obsługi i poprawnego działania mechanik gry.

**Opis struktury projektu:**

* **Strona główna** (index.html) – storna ta posiada główna sekcje, która przechowuje trzy kontenery, które określają etap gry:
  + panel startowy, w który zawarty jest tytuł, pole do podania nazwy użytkownika oraz przycisk do rozpoczęcia gry.
  + quiz-panel odpowiedzialny za wyświetlanie pytań quizu oraz odpowiedzi,
  + panel wynikowy, w którym wyświetlane są rezultaty quizu oraz tabela wyników.
* St**ylizacja CSS** – dedykowany arkusz styli odpowiedni dla każdego panelu.
* **Skrypt JavaScript i funkcjonalności** – głównym plikiem jest scripts.js, który odpowiada za całą mechanikę, gry, obsługą bazy pytań w pliku JSON i magazynem sieciowy do tabeli wyników.

**2. Zdefiniowanie problemu do realizacji**

Celem projektu jest stworzenie interaktywnej aplikacji quizowej, która umożliwia użytkownikom odpowiedzi na różnorodne pytania i otrzymywanie wyników w zależności od ich odpowiedzi. Aplikacja ma mieć funkcjonalność ładowania pytań z pliku JSON, mieszania odpowiedzi, prezentacji wyników oraz zapisywania najlepszych wyników w lokalnym magazynie danych.

**Problem do realizacji:**

Problemem jest utworzenie poprawnie działającego skryptu, który:

* **Poprawnego pobierania pytań oraz odpowiedzi na nie:**
* Skrypt musi umożliwiać pobieranie pytań z bazy danych bez wprowadzania zmian w ich treści.
* Zbieranie danych z bazy powinno odbywać się bez błędów, zachowując oryginalną formę pytań wraz z odpowiedziami.
* **Poprawnego wyświetlania pytań oraz odpowiedzi:**
* Pytania powinny być prezentowane w sposób, który ułatwi zrozumienie ich treści, a odpowiedzi powinny być czytelne i łatwo dostępne dla gracza.
* **Mechanika gry:**
* Skrypt musi obsługiwać płynny przebieg gry, umożliwiając przechodzenie między pytaniami.
* Gracz powinien mieć możliwość udzielenia odpowiedzi na każde pytanie, a skrypt musi rejestrować poprawne i błędne odpowiedzi.
* Gra powinna być zakończona po udzieleniu odpowiedzi na wszystkie pytania.
* Gracz nie może się spodziewać tej samej kolejności pytań jak i odpowiedzi na nie.
* Kilku graczy korzystających z gry na jednym urządzeniu powinni rozróżnić swoje wyniki gry.
* **Nieprzewidziane ruchy gracza:**
* Skrypt powinien być odporny na nieprawidłowe działania gracza, które mogą prowadzić do błędów w odczycie danych.
* **Intuicyjność:**
* Gra powinna być przystosowana do każdego użytkownika nie zależnie od doświadczenia.
* Gracz powinien widzieć czy udzielił poprawnej lub błędnej odpowiedzi.
* **Dostępność:**
* Gra powinna być dostępna dla wszystkich użytkowników, nie zależnie od rodzaju urządzenie z jakiego korzystają.

**3. Propozycja rozwiązań problemu**

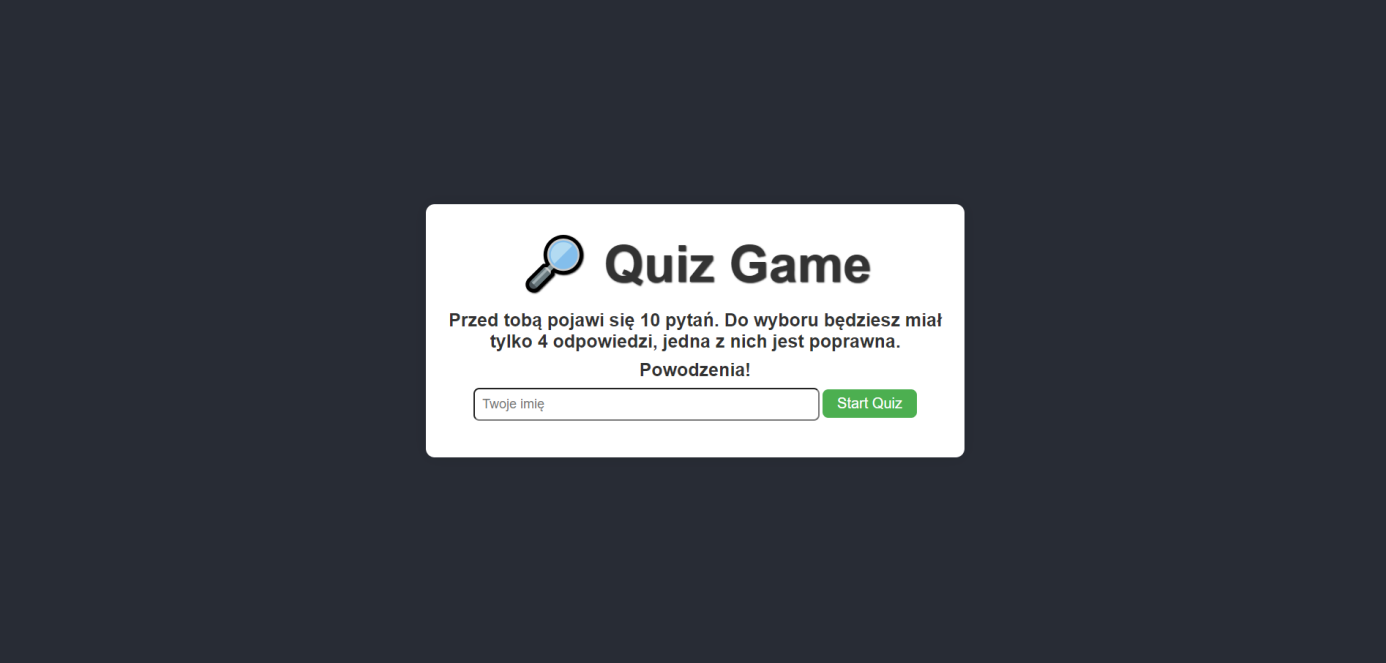
* **Struktura oraz organizacja gry:**
* Cel: Łatwe zarządzanie plikami gry, budową strony.
* Rozwiązanie:
  + Utworzenie osobnych kontenerów w pliku HTML, odpowiadających za dany etap w grze.
* Implementacja:
  + Kod jest zorganizowany względem wyżej wymienionych części, z przemyślanym nazewnictwem klas, identyfikatorów, funkcji oraz różnych zmiennych.
  + Wykorzystanie czytelnego formatowania aby ułatwić poruszanie się w kodzie.
* **Poprawność danych:**
  + Cel: Poprawne sczytanie danych do inicjacji gry.
  + Rozwiązanie:
    - Utworzenie pliku baz pytań JSON i funkcji w kodzie JavaScript łączącej się z bazą.
* Implementacja:
  + Utworzona funkcja łączy się z bazą pytań oraz sprawdza poprawność zainicjowanych pytań do kodu.
* **Zabezpieczenie kodu:**
  + Cel: Zabezpieczenie kodu przed niechcianymi ruchami gracza.
  + Rozwiązanie:
    - Ustawienie wielu funkcji sprawdzających ruchy gracza.
* Implementacja:
  + Zablokowanie ruchu gracza po udzieleniu odpowiedzi oraz chwilowe zatrzymaniu się na pytaniu.
* **Interaktywność** 
  + Cel: Stworzenie interaktywnej gry, która jest dostępna na wszystkie urządzenia oraz dostępna dla wielu graczy jeśli grają na jednym urządzeniu.
  + Rozwiązanie:
    - Za pomocą arkusza styli CSS, ustawienie wyglądu oraz animacji paneli. Wykorzystanie języka JavaScript w celu obsługi interaktywności oraz mechanik gry.
* Implementacja:
  + Interfejs jak i pytania zostały uatrakcyjnione. Gracz może dostrzec, czy wybrana przez niego odpowiedz jest poprawną odpowiedzą czy też nie. W panelu końcowy wyświetlany jest wynik końcowy całego quizu oraz tabela wyników, która zapamiętuje wyniki danej gry.

**Podsumowanie:**

Projekt skoncentrował się na stworzeniu gry quizowej, która nie tylko dostarcza wiedzy, ale również zapewnia użytkownikom przyjemne doświadczenie. Zastosowanie odpowiednich technologii, zabezpieczeń oraz dbałość o strukturę kodu przyczyniły się do stworzenia stabilnego i atrakcyjnego produktu.

**4. Testowanie**

1. **Panel startowy:**

****

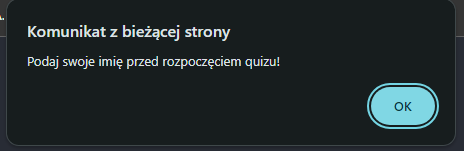
rys. 1.0

W panelu Starowym zawiera się:

* tytuł,
* paragraf informujący nas jakie są zasady gry,
* pole do podania swojego imienia,
* przycisk „Start Quiz”, który rozpoczyna grę.

Do rozpoczęcia gry wymagane jest podanie imienia gracza, które będzie użyte w tabeli wyników. Jeśli gracz nie uzupełni pole, program wyświetli komunikat informujący go

o podanie swojego imienia.



rys. 1.1

1. **Gra:**

****

rys. 2.0

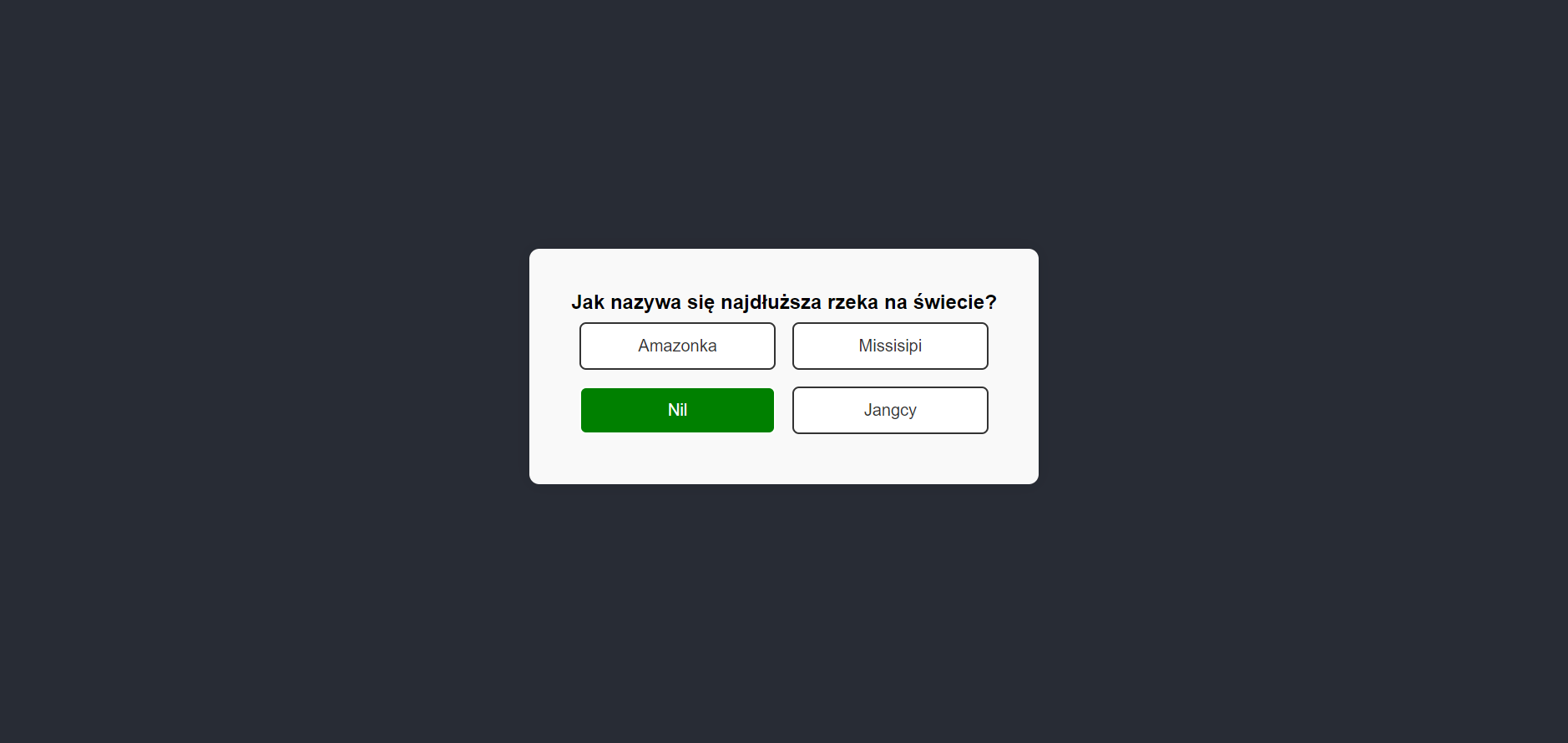
Po podaniu przez gracza imienia gracza zostają wyświetlone pytania quizowe. Przed graczem ukazuje się pytanie z czterema odpowiedziami, tylko jedna z nich jest poprawna.



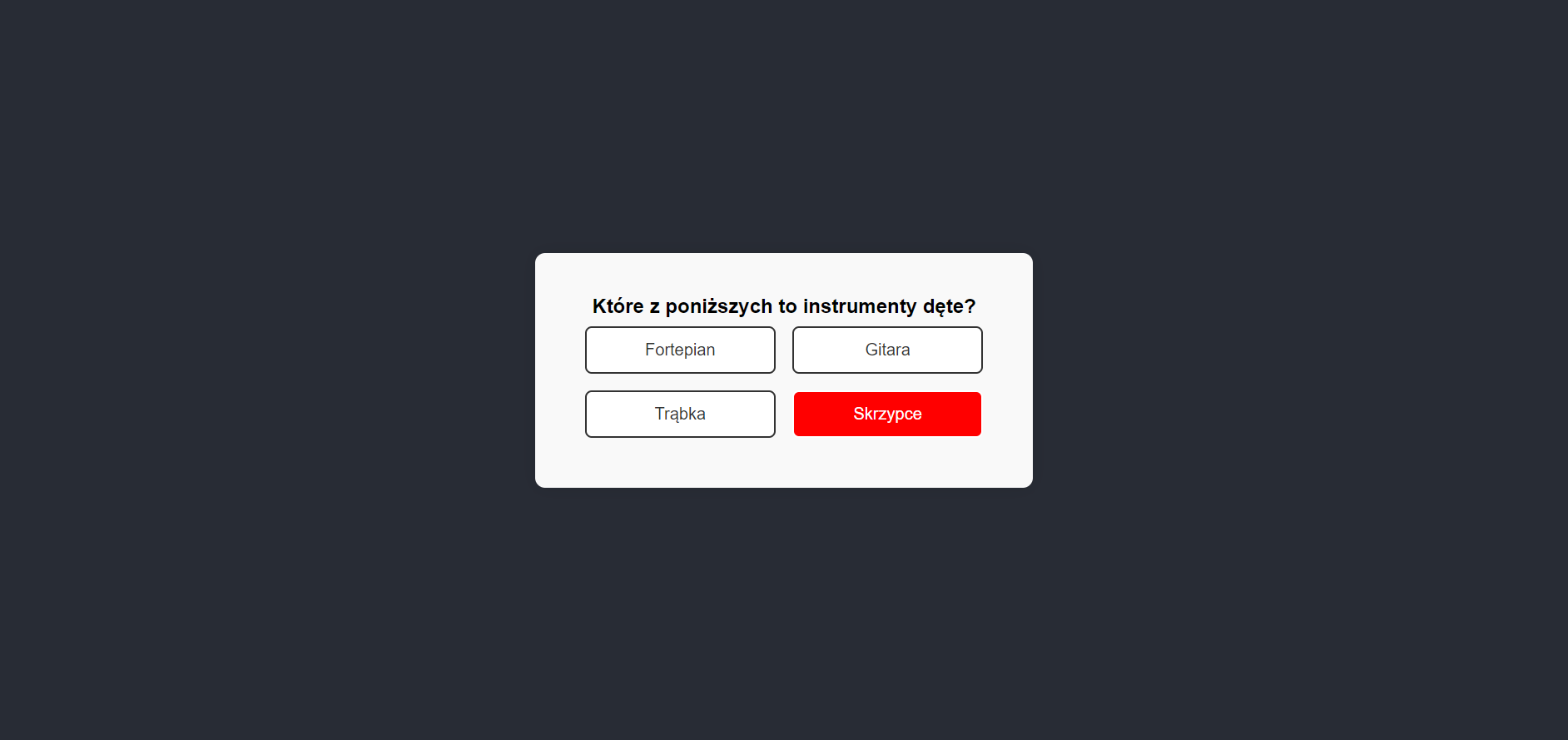
rys. 2.1

Po najechaniu przez użytkownika kursorem na jedną z odpowiedzi, zostaje ona wyróżniona. Zmienia się jest tło oraz kolor teksu.

Po udzieleniu odpowiedzi na pytanie, przycisk zmienia kolor w zależności od odpowiedzi. Jeśli użytkownik odpowiedział prawidłowo, przycisk zmieni swój kolor na zielony (rys. 2.3). Natomiast, jeśli gracz zaznaczył nieprawidłową odpowiedz, przycisk zmieni się na czerwony (rys 2.4).



rys. 2.3



rys. 2.4

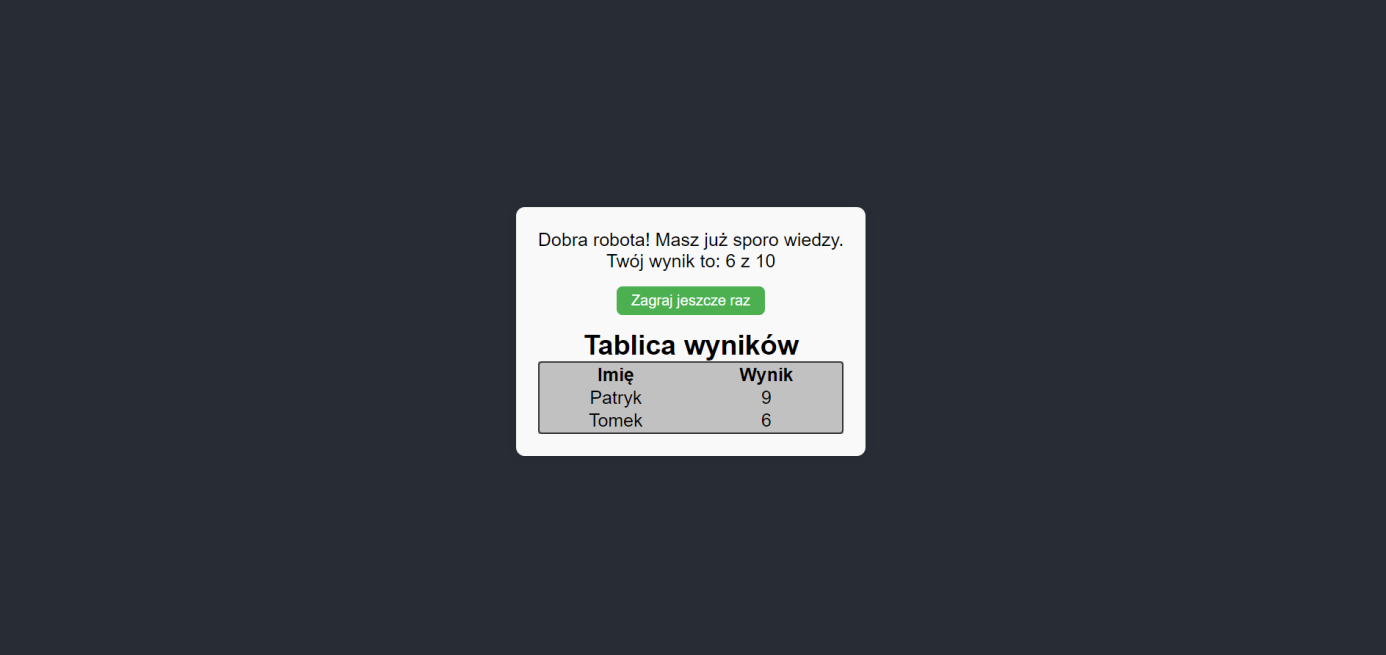
1. **Panel wyników.**

Po udzieleniu odpowiedzi na wszystkie pytania, graczowi wyświetla się podsumowanie gry. Zostają wyświetlone punkt gracza na ilość możliwych do zdobycia oraz hasło podsumowujące grę gracza.



rys. 3.0

Gracz zostaje również wpisany do tabeli wyników, gdzie można porównać wyniki z innymi graczami po rozegranej grze.



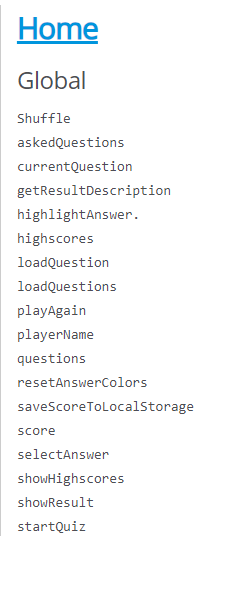
rys. 3.1

**5. Dokumentacja**

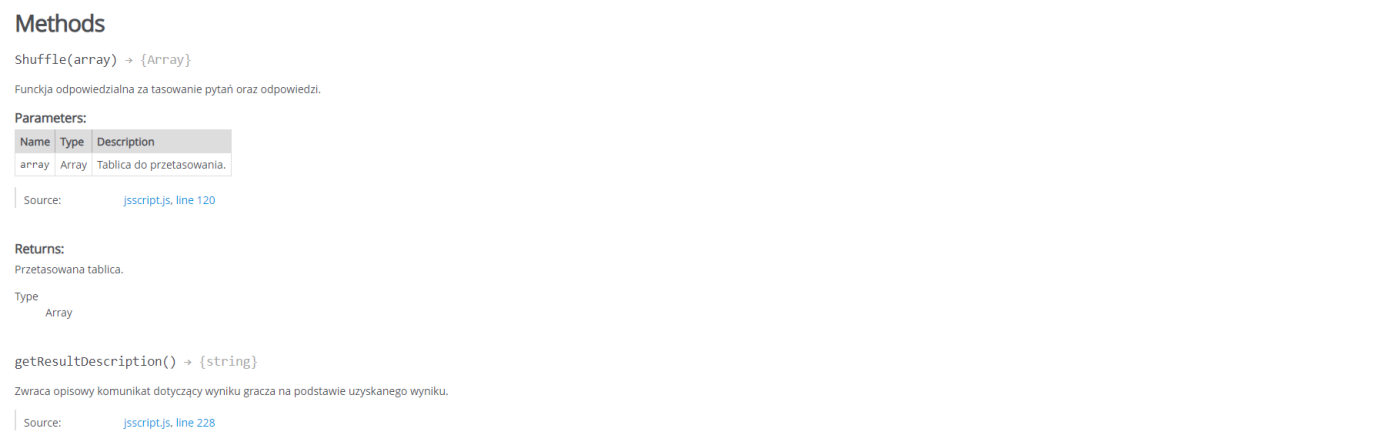
**JSDoc:**

JavaScript wykorzystany w projekcie udokumentowany jest za pomocą JSDoc, standardu dla komentarzy w kodzie JavaScript, który pozwala na utworzenie atrakcyjnego i funkcjonalnej dokumentacji. Obejmuje:

* opis funkcji i metod: jasne wyjaśnienia dotyczące tego, co wykonuje dana funkcja lub metoda, jakie argumenty przyjmuje i co zwraca.



rys. 4.0



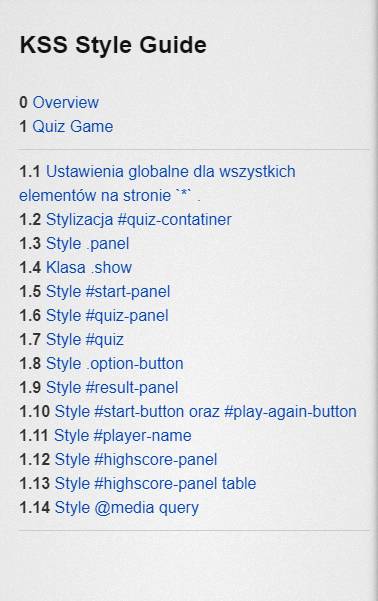
rys. 4.1

Całość dokumentacji znajduje się : [JSDoc: Home (matpat125143.github.io)](https://matpat125143.github.io/Projekt_JS/jsdoc/index.html)

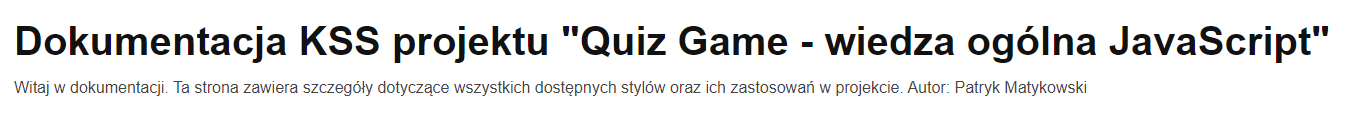
Oraz dokumentacja znajduje się w folderz **jsdoc** w pliku **index.html** .

**KSS:**

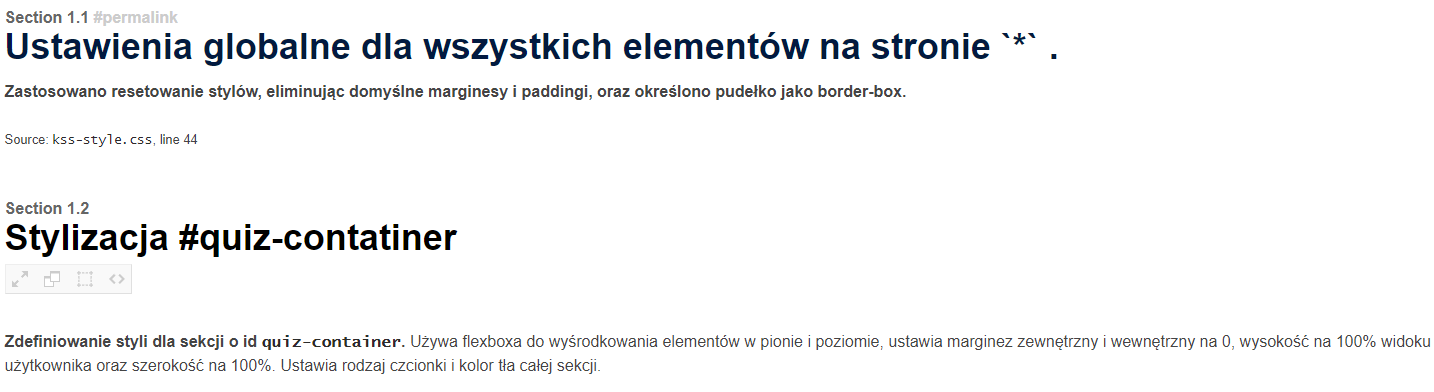
Komentarze są używane, aby opisać, do czego służy każdy blok kodu oraz jakie są jego cechy i funkcje. Pomaga zrozumieć strukturę i działanie kodu oraz ułatwia utrzymanie i ewentualne modyfikacje w przyszłości.



rys. 5.0



rys. 5.1



rys. 5.2

Całość dokumentacji znajduje się: [KSS Style Guide (matpat125143.github.io)](https://matpat125143.github.io/Projekt_JS/kss/styleguide/index.html)

Oraz dokumentacja znajduje się również w folderze **kss/styleguide** w pliku **index.html** .

**6. Linki do źródeł**

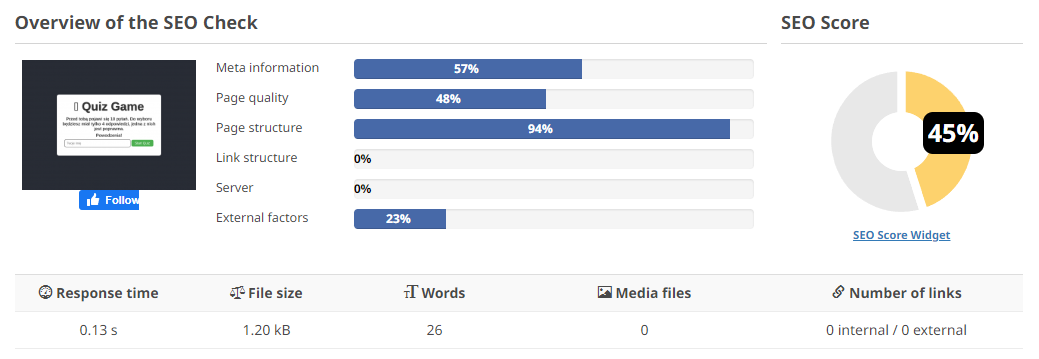
Inspiracje:

* Dokumentacja W3Schools: [W3Schools Online Web Tutorials](https://www.w3schools.com/)
* [How to make a simple JavaScript quiz | Simple Steps Code](https://simplestepscode.com/javascript-quiz-tutorial/)

Całość projektu dostępna na stronie: [Quiz Game (matpat125143.github.io)](https://matpat125143.github.io/Projekt_JS/)

**7. Raport SEO**

1. [SEO Checker | Test your website for free with Seobility](https://www.seobility.net/en/seocheck/)

****

rys. 6.0

**Ocena SEO**: Ogólna ocean SEO wynosi 45%, co jest przedstawione na diagramie kołowy. Ta ocena jest podsumowaniem wydajności strony we wszystkich mierzonych aspektach.

**Meta information**: Ocena wyniosła 57%, świadczy o tym, że tagi meta, takie jak tytuł storny, opis strony i słowa kluczowe są za krótkie.

**Page quality**: Ocena z jakość strony wyniosła 48%, co sugeruje, że są rzeczy do poprawy

jakości strony. Takimi obszarami do poprawy są m.in. treści strony, czytelność lub wydajność strony.

**Page structure**: Ocena za budowę strony wyniosła 94%. Jest to bardzo dobry wynik i świadczy o bardzo dobrym zoptymalizowaniu strony, jednakże istnieją drobne błędny która zaważyły o ocenie.

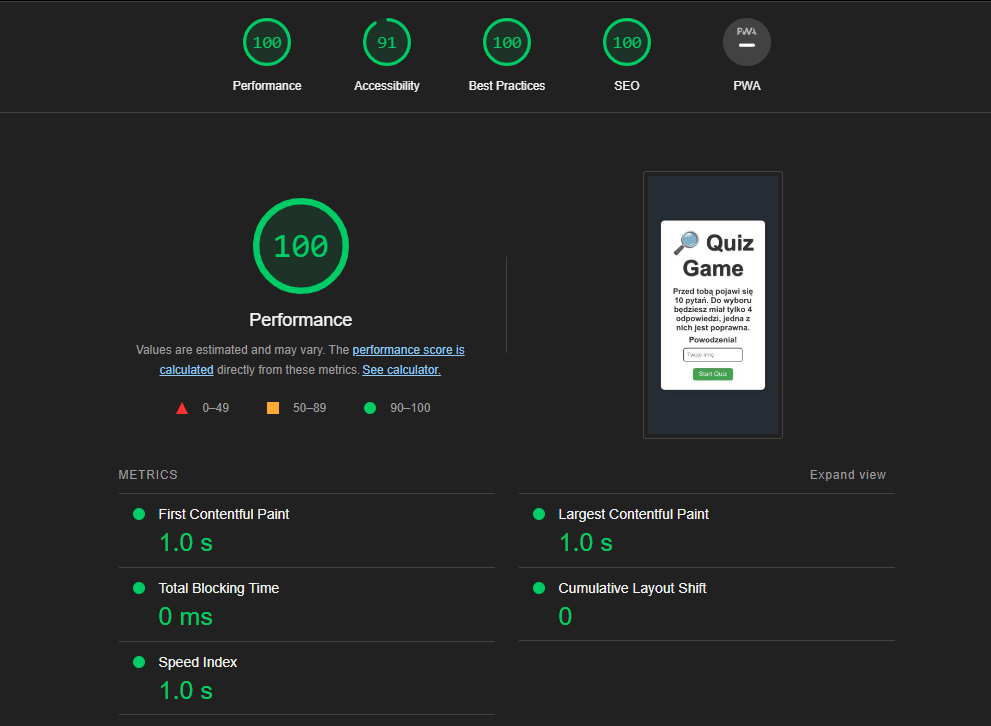
**External factors**: Wynik dla czynników zewnętrzny wyniósł 23%, co jest niską oceną. Świadczy o tym, że brakuje takich rzeczy jaki linków zwrotnych, obecność w mediach społecznościowych i inne czynniki zewnętrze poza stroną.

**Czas reakcji**: Czas odpowiedzi serwera jest bardzo dobry, wynosi 0,13 sekundy.

**Rozmiar pliku**: Całkowity rozmiar pliku strony wyniósł 1,20kB co jest bardzo korzystne dla czasów ładowania strony.

**Słowa**: Ilość użytych słów na stronie wyniosła 26. W zależności od przeznaczenia strony, można to uznać za bardzo małą ilość pod kątem SEO, ponieważ strony z dłuższymi treściami mogą być wyżej pozycjonowane od tym z mniejszą ilością.

1. <https://developer.chrome.com/docs/lighthouse/>



rys. 6.1

**Wydajność (Performance)**: 100/100 – Wynik idealny. Oznacza to, że strona jest bardzo dobrze zoptymalizowana pod kątem szybkości i wydajności. Wysoka ocena w tej kategorii sugeruje krótki czas ładowania, efektywne wykorzystanie zasobów i dobrą reakcję na interakcję użytkownika.

**Dostępność (Accessibility):** 91/100 - Wynik wskazuje, że strona w ogólnym rozrachunku jest dostępna, ale mogą występować pewne problemy, które można by poprawić, aby strona była bardziej dostępna dla osób z niepełnosprawnościami, takie jak potencjalne problemy z kontrastem, etykietami formularzy lub strukturą nagłówków.

**Najlepsze Praktyki (Best Practices):** 100/100 - Ten wynik wskazuje na stosowanie najlepszych praktyk w zakresie tworzenia stron internetowych.

**SEO:** 100/100 - Ocena SEO jest bardzo dobra, wskazuje to, że strona jest dobrze zoptymalizowana pod kątem wyszukiwarek internetowych.