

## Wprowadzenie

Aplikacja to interaktywny quiz, który pozwala użytkownikom na udział w teście składającym się z 10 pytań.

## Opis działania aplikacji

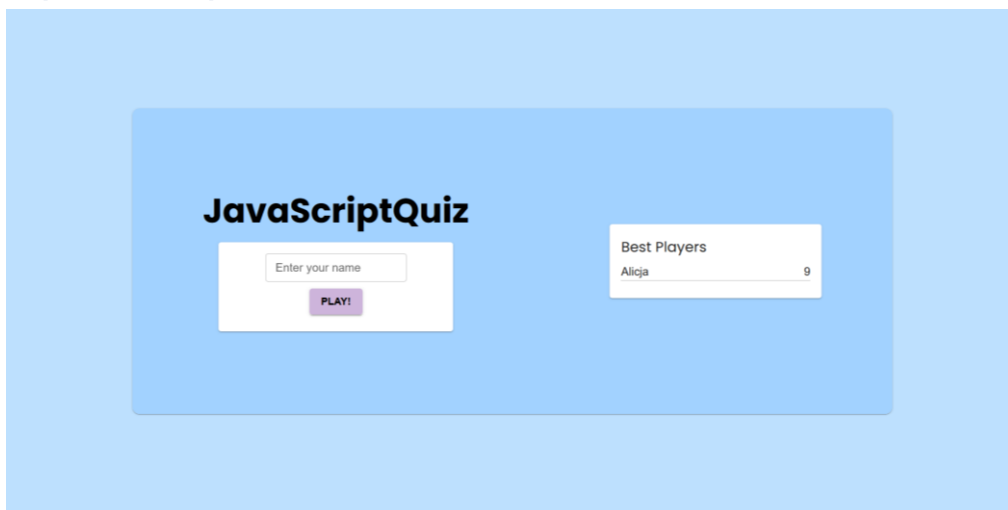
1. Po wejściu na stronę główną, użytkownik zostaje poproszony o podanie swojego imienia.
2. Po wprowadzeniu imienia i kliknięciu przycisku "PLAY!", użytkownik zostaje przeniesiony do strony z quizem.
3. Na stronie z quizem wyświetlane są pytania pochodzące z API. Pytania są filtrowane, aby wyświetlać tylko te z pojedynczą poprawną odpowiedzią.
4. Użytkownik odpowiada na pytania, klikając na odpowiedni przycisk, który reprezentuje jedną z odpowiedzi.
5. Po udzieleniu odpowiedzi, aplikacja sprawdza, czy udzielona odpowiedź jest poprawna. Jeśli tak, zwiększa liczbę punktów użytkownika o 1.
6. Aplikacja przechodzi do następnego pytania, wyświetlając kolejne pytanie.
7. Po udzieleniu odpowiedzi na 10 pytanie, użytkownik zostaje przekierowany na stronę główną.
8. Na stronie głównej wyświetlane są również informacje o najlepszych graczach, pobrane z bazy danych za pomocą API.
9. Po zakończeniu quizu, informacje o gracz (imię i liczba zdobytych punktów) są wysyłane do API i zapisywane w bazie danych.
10. Aplikacja umożliwia wielokrotne rozpoczęcie quizu przez różnych użytkowników i wyświetla ranking najlepszych graczy.

## Użyte technologie

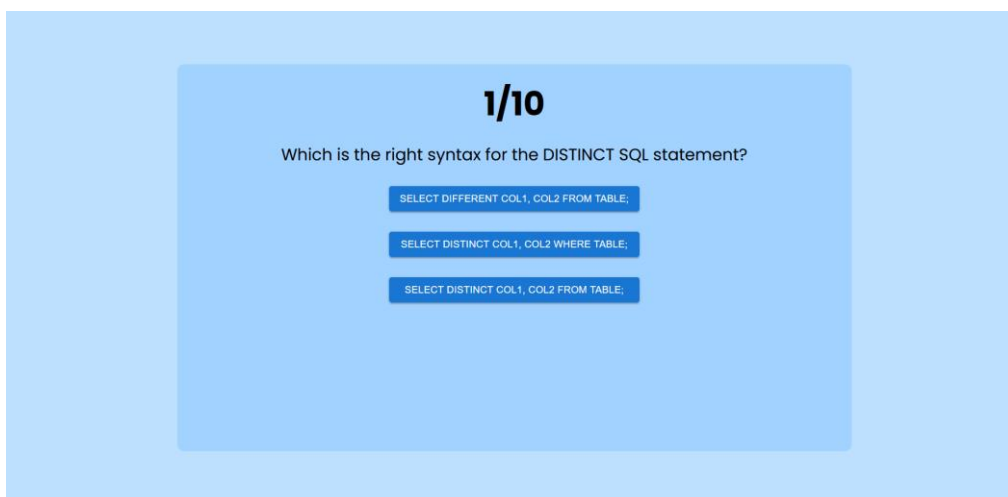
- React: Aplikacja jest napisana w języku JavaScript przy użyciu biblioteki React, która umożliwia tworzenie interaktywnych interfejsów użytkownika.
- React Router: Do obsługi routingu w aplikacji wykorzystuje się bibliotekę React Router. Pozwala to na nawigację między różnymi widokami (stronami) aplikacji.
- Material-UI: Do stylizacji interfejsu użytkownika wykorzystuje się bibliotekę Material-UI, która dostarcza gotowe komponenty UI zgodne z zasadami Material Design.
- React Hook Form: Biblioteka React Hook Form jest używana do obsługi formularzy w aplikacji. Umożliwia łatwe zarządzanie stanem formularza i walidację wprowadzonych danych.

- REST API: Aplikacja komunikuje się z API przy użyciu protokołu HTTP i wykorzystuje metody takie jak GET i POST do pobierania i wysyłania danych.
- MongoDB: Baza danych MongoDB jest wykorzystywana do przechowywania informacji o graczach i ich wynikach. Aplikacja korzysta z biblioteki mongoose do interakcji z bazą danych.
- Express: Aplikacja serwerowa została zaimplementowana przy użyciu frameworka Express, który umożliwia budowanie aplikacji sieciowych w Node.js.

## Opis komponentów



Strona główna aplikacji użytkownik zostaje poproszony o podanie swojego imienia. Po wprowadzeniu imienia i kliknięciu przycisku "PLAY!", użytkownik zostaje przeniesiony do strony z quizem.



Drugim oknem aplikacji jest właściwa strona quizu, dane o nazwie oraz o poszczególnych odpowiedziach pobierane są z API. Po kliknięciu w przycisk czynność jest powtarzana do momentu kliknięcia w ostatnie pytanie, wtedy użytkownik zostaje przekierowany na stronę główną.

## Podsumowanie

Aplikacja JavaScript Quiz wykorzystuje React do budowy interfejsu użytkownika, React Router do obsługi routingu, Material-UI do stylizacji UI, React Hook Form do zarządzania formularzami, REST API do komunikacji z serwerem, MongoDB do przechowywania danych, oraz Express jako framework serwerowy.