

QUIZ

MARCEL

ŻAK

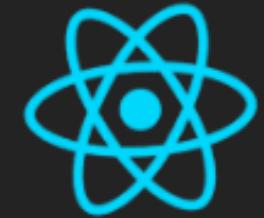
QUIZ W REACT

KROK 1 TWORZENIE

```
$ npm create vite@latest quiz -- --template react  
cd quiz  
npm install
```

KROK 2 START SERWERA

```
npm run dev
```

[HOME](#)[ABOUT](#)[MORE](#)

Vite + React

count is 0

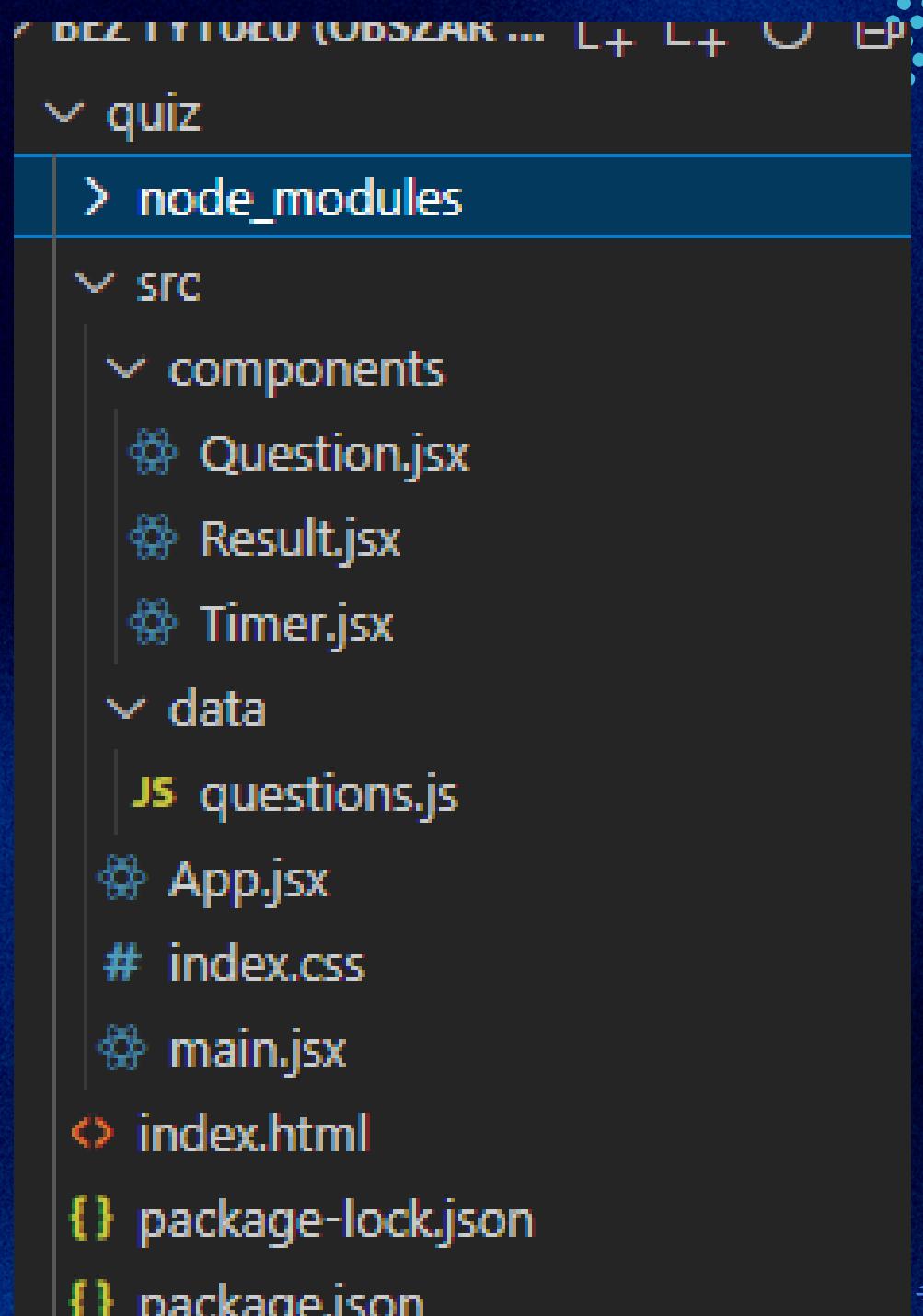
Edit src/App.tsx and save to test HMR

Click on the Vite and React logos to learn more

KROK 3

STRUKTURA

PLIKÓW



```
DESKTOP (ODSŁĄK) L+ L+ ⌂
└── quiz
    ├── node_modules
    └── src
        ├── components
        │   ├── Question.jsx
        │   ├── Result.jsx
        │   └── Timer.jsx
        ├── data
        │   ├── questions.js
        │   ├── App.jsx
        │   ├── index.css
        │   ├── main.jsx
        │   └── index.html
        └── package-lock.json
            └── package.json
```

PYTANIA

Stworzenie pytań

Aby nasza aplikacja mogła działać, potrzebujemy zestawu pytań i odpowiedzi. W tym kroku tworzymy plik questions.js w folderze data, który zawiera tablicę obiektów. Każdy obiekt reprezentuje jedno pytanie z czterema możliwymi odpowiedziami oraz indeksem poprawnej odpowiedzi. Ten plik będzie później importowany w głównym komponencie, dzięki czemu aplikacja będzie mogła dynamicznie przechodzić przez kolejne pytania.

Kod

```
export const questionsData = [
  {
    question: "Stolica Polski to:",
    answers: ["Kraków", "Warszawa", "Gdańsk", "Wrocław"],
    correct: 1,
  },
  {
    question: "Ile to  $5 + 3$ ?",
    answers: ["6", "7", "8", "9"],
    correct: 2,
  },
  {
    question: "Który pierwiastek chemiczny ma symbol O?",
    answers: ["Ołów", "Złoto", "Tlen", "Tytan"],
    correct: 2,
  },
  {
    question: "Które morze leży na północy Polski?",
    answers: ["Bałtyckie", "Czarne", "Śródziemne", "Północne"],
    correct: 0,
  },
  {
    question: "W którym roku rozpoczęła się II wojna światowa?",
    answers: ["1918", "1939", "1945", "1920"],
    correct: 1,
  },
  {
    question: "Jak nazywa się proces wytwarzania energii w Słońcu?",
    answers: ["Fotosynteza", "Rozszczepienie", "Synteza jądrowa", "Kondensacja"],
    correct: 2,
  },
  {
    question: "Kto napisał 'Pan Tadeusz'?",
    answers: ["Juliusz Słowacki", "Adam Mickiewicz", "Henryk Sienkiewicz", "Bolesław Prus"],
    correct: 1,
  },
  {
    question: "Który język programowania używany jest w React?",
    answers: ["Java", "Python", "JavaScript", "C++"],
    correct: 2,
  },
  {
    question: "Co to jest CPU w komputerze?",
    answers: ["Monitor", "Procesor", "Pamięć", "Dysk"],
    correct: 1,
  },
  {
    question: "Która planeta jest najbliższej Słońca?",
    answers: ["Wenus", "Ziemia", "Mars", "Mercury"],
    correct: 3,
  },
];
```

LOGIKA PYTAÑ

Question.jsx to komponent odpowiedzialny za wyświetlanie konkretnego pytania oraz możliwych odpowiedzi. Użytkownik może kliknąć jedną z opcji, a komponent zaznaczy ją jako wybraną i na chwilę pokaże, czy odpowiedź była poprawna (zielona) czy błędna (czerwona). Następnie po dwóch sekundach informuje główny komponent (App) o wyniku tej odpowiedzi. Dodatkowo, jeśli użytkownik nie udzieli odpowiedzi w ciągu pięciu sekund, automatycznie uruchamia się funkcja, która traktuje pytanie jako błędnie udzielone i pokazuje poprawną odpowiedź. Dzięki temu komponent potrafi działać zarówno interaktywnie (kliknięcie), jak i reagować na upływ czasu.

```
import React, { useState, useEffect } from 'react'; //Import React
import Timer from './Timer'; //Import Timera

export default function Question({ question, onAnswer }) {

  if (!question) return <div>Ładowanie pytania...</div>;// Jesli brak pytania, wyswietlamy komunikat o ladowaniu
  const [selected, setSelected] = useState(null); //indeks odpowiedzi uzytkownika
  const [answered, setAnswered] = useState(false); //zmienna sprawdzajaca czy jest odpowiedz na pytanie

  useEffect(() => {
    setSelected(null);
    setAnswered(false);
  }, [question]);

  // Funkcja wywoływaną po kliknięciu odpowiedzi
  const handleAnswer = (index) => {
    if (answered) return; // Jesli uzytkownik juz odpowiedzial, ignorujemy kliknięcie
    setSelected(index); //indeks wybranej odpowiedzi
    setAnswered(true); //odpowiedziano

    // Po 2 sekundach wywołujemy funkcję onAnswer z informacją, czy odpowiedź była poprawna
    setTimeout(() => {
      onAnswer(index === question.correct);
    }, 2000);
  };

  // Funkcja wywoływaną, gdy czas minie i użytkownik nie odpowie
  const handleTimeUp = () => {
    if (!answered) {
      setAnswered(true); // Zaznaczamy jako odpowiedziane
      setSelected(null); // Brak wybranej odpowiedzi

      // Po 2 sekundach pokazuje, że nie było odpowiedzi
      setTimeout(() => {
        onAnswer(false);
      }, 2000);
    }
  };
}
```

```
return (
  <>
  {/*jesli brak odpowiedzi wyświetla timer */}
  {!answered && <Timer duration={5} onTimeUp={handleTimeUp} />}

  {/* div z pytaniem i propsem question */}
  <div className="question">{question.question}</div>

  {/* Wyswietlenie listy odpowiedzi */}
  <div className="answers">
    {question.answers.map((ans, idx) => {
      const isCorrect = idx === question.correct;//czy poprawna odpowiedz
      const isSelected = idx === selected;//czy poprawna jest wybrana
      //przypisanie CSS w zaleznosci od odpowiedzi
      let className = '';
      if (answered) {
        if (isCorrect) className = 'correct';
        else if (isSelected && !isCorrect) className = 'incorrect';
      }
    })
  }

  return (
    <button
      key={idx}// unikalny klucz
      onClick={() => handleAnswer(idx)}// klikniecie = wybor odpowiedzi
      className={className}// CSS
      disabled={answered}// blokujemy kliknienia po odpowiedzi
      type="button"//ustawienie typu button
    >
      {ans} /* Tekst odpowiedzi */
    </button>
  );
})}
</div>
</>
);
```

ODLICZANIE

Timer.jsx odpowiada za odliczanie czasu dla każdego pytania. Wizualnie przedstawia to jako pasek postępu nad pytaniem, który stopniowo się skraca. Komponent ten jest bardzo dynamiczny – działa przez pięć sekund i wywołuje określoną funkcję w Question, jeśli czas się skończy. Działa bardzo płynnie, aktualizując pasek co 50 milisekund. Gdy użytkownik udzieli odpowiedzi, timer zniką lub przestaje działać, co zapobiega dalszemu odliczaniu.

[HOME](#)[ABOUT](#)[MORE](#)

```
import React, { useEffect, useState } from "react";

export default function Timer({ duration, onTimeUp }) {
  const [widthPercent, setWidthPercent] = useState(100); // ustawiamy timer na 100

  useEffect(() => {
    setWidthPercent(100);

    const start = Date.now(); // czas rozpoczęcia
    const interval = setInterval(() => {

      const elapsed = Date.now() - start; // ile czasu zostało
      const percent = Math.max(100 - (elapsed / (duration * 1000)) * 100, 0); // Obliczamy procent pozostałego czasu
      setWidthPercent(percent); // aktualizacja timera
      if (percent <= 0) {
        clearInterval(interval); // zatrzymanie czasu jeśli się skończył
        onTimeUp(); // wywołanie przekazanej funkcji jako props
      }
    }, 50); // interwał co 50 ms

    return () => clearInterval(interval); // usuwamy interwał przy odmontowaniu komponentu
  }, [duration, onTimeUp]);

  return (
    <div className="timer">
      <div
        className="timer-progress"
        style={{ width: `${widthPercent}%` }} // szerokość oparta o stan
      />
    </div>
  );
}
```

Result.jsx to bardzo prosty komponent, którego zadaniem jest wyświetlenie informacji o zakończeniu quizu. Pokazuje użytkownikowi komunikat, że to już koniec, oraz jego końcowy wynik, czyli ile pytań zostało udzielonych poprawnie spośród wszystkich.

```
import React from "react";

/**
 * @param {Object} props - Przykładowe dane do komponentu
 * @param {Number} props.score - Wynik użytkownika
 * @param {Number} props.total - Liczba pytań
 */
export default function Result({ score, total }) {
  return (
    <div style={{
      textAlign: "center", // wyśrodkowanie tekstu
      marginTop: 40 // odstęp od góry (w pikselach)
    }>
      <h2>Koniec quizu!</h2> /* Nagłówek informujący o zakończeniu */
      <p>Twój wynik: {score} / {total}</p> /* Wyświetlenie wyniku */
    </div>
  );
}
```

```

import React, { useState } from 'react';

import Question from './components/Question';
import Result from './components/Result';
import { questionsData } from './data/questions';

export default function App() {
  const [currentQuestionIndex, setCurrentQuestionIndex] = useState(0); // indeks aktualnego pytania
  const [score, setScore] = useState(0); // poprawne odpowiedzi
  const [showResult, setShowResult] = useState(false); // sprawdzenie czy to ostatnie pytanie

  const handleAnswerClick = (isCorrect) => {
    if (isCorrect) setScore(score + 1); //poprawna odpowiedz score+=1

    const nextIndex = currentQuestionIndex + 1; //zwiększamy indeks pytania

    if (nextIndex < questionsData.length) { //jesli indeks mniejszy niz dlugosc tablicy questionData
      setCurrentQuestionIndex(nextIndex); //nastepne pytanie
    } else {
      setShowResult(true); //pokaz wynik
    }
  };

  return (
    <div className="app-container">
      {showResult ? (
        <Result score={score} total={questionsData.length} /> //przekazanie propsu wyniku i ilosci pytan
      ) : (
        <Question
          question={questionsData[currentQuestionIndex]} //przekazanie propsu indeksu pytania
          onAnswer={handleAnswerClick} // przekazanie funkcji
        />
      )}
    </div>
  );
}

```

PLIK APP.JSX TO NAJWAŻNIEJSZY ELEMENT APLIKACJI QUIZOWEJ, BO TO ON STERUJE CAŁYM PRZEBIEGIEM QUIZU – OD PIERWSZEGO PYTANIA AŻ DO POKAZANIA WYNIKU NA KOŃCU.

W PLIKU APP.JSX ZAPISYWANE SĄ TRZY NAJWAŻNIEJSZE INFORMACJE: KTÓRE PYTANIE JEST TERAZ WYSWIETLANE (CURRENTQUESTIONINDEX), ILE PUNKTÓW ZDOBYŁ UŻYTKOWNIK (SCORE), CZY QUIZ JUŻ SIĘ ZAKOŃCZYŁ (SHOWRESULT). ZA SPRAWDZANIE ODPowiedzi i PRZEHODZENIE DALEJ ODPOWIADA FUNKCJA HANDLEANSWERCLICK. GDY UŻYTKOWNIK KLIKNIĘ ODPOWIEDŹ: JEŚLI JEST POPRAWNA, WYNIK SIĘ ZWIĘKSZA,

- POTEM APLIKACJA PRZEHODZI DO KOLEJNEGO PYTANIA ALBO KOŃCZY QUIZ, JEŚLI TO BYŁO OSTATNIE.

W ZALEŻNOŚCI OD TEGO, CZY QUIZ TRWA, CZY JUŻ SIĘ SKOŃCZYŁ, APP.JSX POKAZUJE: PYTANIE (KOMPONENT QUESTION), GDY QUIZ JESZCZE TRWA, WYNIK (KOMPONENT RESULT), GDY QUIZ SIĘ ZAKOŃCZYŁ.

Stolica Polski to:

Kraków

Warszawa

Gdańsk

Wrocław

Stolica Polski to:

Kraków

Warszawa

Gdańsk

Wrocław

Stolica Polski to:

Kraków

Warszawa

Gdańsk

Wrocław

Koniec quizu!

Twój wynik 0 / 10

HOME

ABOUT

MORE

KONIEC

Marcel Źak 4p