

Kandydat: Jan Kowalski; **Poziom:** Junior; **Technologia:** JavaScript; **Subtechnologie:** React, TypeScript

Opinia MockIT: kandydat kwalifikuje się na pozycję Juniora

Doświadczenie i projekty (7/10)

Kandydat posiada kilka ambitnych prywatnych projektów m.in. projekt pobierający dane na temat cen nieruchomości i analizujący zmienność cen w roku oraz na przestrzeni kilku lat. Kandydat wykorzystał w nim React, Express, Docker i PostgreSQL. Potrafił wytłumaczyć jak działają funkcjonalności aplikacji. Brak stworzonej dokumentacji do projektu. Tylko podstawowa organizacja kodu i niejasne nazewnictwo plików.

Umiejętności techniczne (6/10)

Na pytania na poziomie juniorskim kandydat w większości odpowiadał wyczerpująco i poprawnie. Najczęściej brak odpowiedzi przy pytaniach rozszerzających pytania wyjściowe.

1. React i ogólny frontend

- Znajomość podstawowych metod renderowania strony WWW (SSR, CSR) (**mały plus**)
- Podstawowa wiedza o hookach w React (useState, useEffect, useContext) (**mały plus**)
- Zrozumienie zasady działania i wszystkich sposobów wykorzystania refów (**znaczny plus**)
- Nieznajomość pojęcia memoizacji w React (**znaczny problem**)
- Nieznajomość pojęcia customowych hooków (**mały problem**)

2. Craftsmanship

- Umiejętność pisania komponentów reużywalnych, znajomość dobrych praktyk w kodowaniu (DRY, KISS) (**znaczny plus**)
- Kandydat wie co zrobić gdy zablokuje go task (**mały plus**)

3. TypeScript

- Znajomość i zrozumienie podstaw TypeScripta (typowanie, interfejsy, różnice względem JavaScriptu) (**nieznaczny plus**)
- Kandydat nie potrafił wskazać antypatternów przy korzystaniu z TypeScriptu (**mały problem**)

4. JavaScript

- Bardzo dobra znajomość funkcji JSa (deep clone obiektów, konstrukcje wprowadzone w ES6, wskazanie niepoprawnie działających fragmentów kodu) (**znaczny plus**)
- Znajomość stanów, metod statycznych i obsługi wyników Promise'ów (**znaczny plus**)

Rozwiązywanie problemów (8/10)

Podczas rozmowy kandydat otrzymał zadanie polegające na przekształceniu tablicy: `arr = [1, 1, 2, 4, 9, 1, 3, 5, 3, 7]` w taki sposób, aby zawierała tylko unikalne wartości. W zadaniu została sprawdzona umiejętność używania wbudowanych funkcji JavaScript - usuwanie duplikatów i sortowanie tablicy. Kandydat rozwiązał zadanie bez dodatkowych wskazówek, a jego kod zadziałał poprawnie za pierwszym razem. Rozwiązanie nie było optymalne (złożoność - $O(n^2)$ zamiast $O(n)$).

Komunikacja (8/10)

1. Forma odpowiedzi

- Zwykle wyczerpująca i rozbudowana jeśli kandydat dobrze zna temat.

2. Współpraca z rekruterem

- Kandydat dopytuje o szczegóły postawionych przed nim problemów/zadań, przez co czasami był w stanie odpowiedzieć na mniej znane mu pytania. (**znaczny plus**)
- Ograniczona komunikacja swoich myśli podczas “live codingu” (**mały problem**)
- Otwartość i umiejętność słuchania podkreślają zdolność kandydata do efektywnej współpracy w zespole. (**znaczny plus**)

3. Angielski

- Komunikatywny i rozumiały angielski, kandydat używał branżowego słownictwa.
- Zadane pytania: What good coding practices do you know? What's new about ES6?