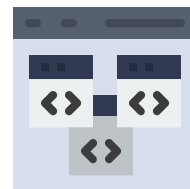


TESTOWANIE WE FRONTENDZIE



Kategoria	Czas trwania	Termin	Cena
Testowanie	21h	ustalamy indywidualnie	ustalamy indywidualnie

Program szkolenia:

Poniżej przedstawiamy przykładowy program szkolenia, który może zostać zmodyfikowany zgodnie z oczekiwaniami oraz poziomem grupy szkoleniowej. Przed przygotowaniem docelowego programu szkolenia, przeprowadzamy rozmowę techniczną, w której bierze udział trener oraz osoba techniczna lub cały zespół developerów reprezentujący klienta, w celu ustalenia szczegółów szkolenia.

■ Typy testów - jak je dobrać do przypadku i co właściwie oznaczają?

- Unit tests
- Components tests
- Integration tests

■ Testować czy nie testować?

- Co to znaczy że kod jest "testowalny"?
- Kompozycja kodu a "testowalność"
- Szczegóły implementacyjne vs kontrakt vs side effects
- Co zrobić z third-party podczas pisania testów?
- Przypadki brzegowe
- Jak uniknąć "zabetonowania" implementacji?

■ Struktura dobrego testu

- DRY vs DAMP vs WET
- Granice
- After* i Before* - jak uniknąć duplikacji i komplikacji
- Organizacja: test suites i test cases

■ Dobrze dobrane assercje

- Loose vs tighter assertions
- Asymmetric matchers
- Custom matchers
- Circular assertions

■ Mockowanie oraz dublerzy w testach - mocks, stubs i spies

- Kiedy wykorzystać te techniki
- Co mockować?
- Wsparcie bibliotek
- Record & Replay pattern
- Pułapki mockowania

■ Testy integracyjne i MSW

- Gdzie są granice?
- Czym są komponenty które integrujemy?
- Jak zapewnić dane w testach?

■ Dane w testach

- Fixtures – czy to zawsze dobra droga?
- Jak generować dane
- snapshots
- test data sets (data providers) – napisz jeden test dla wielu przypadków

■ Techniki pisania testów

- TDD vs Code First vs Test First

■ Jak wyciągnąć więcej z react-testing-library

- Selektory w RTL – byRole etc.
- RTL queries – get vs find vs query
- Asynchroniczność w testach komponentów waitFor, act oraz find*
- Testowanie hooków w RTL
- Interakcje w RTL – user-event
- Snapshots testing – yay or nay

■ Test Utils - jak rozszerzać własne użycie do testów aby nie narobić bałaganu

- Custom assertions
- Jest matchers
- Utils/Helpers i budowanie abstrakcji

■ Code coverage

- Typy pokrycia testami
- Kiedy code coverage pomaga a kiedy daje fałszywe poczucie bezpieczeństwa?

■ Wprowadzenie do e2e tests

- Wprowadzenie do tematu testów e2e
- Tooling i setup
- Wyzwania testów e2e

■ State management a testy

- Co i jak testować jeśli wykorzystujemy Redux'a lub podobne rozwiązanie?
- Testowanie kodu wykorzystującego React Context