# 🧪 React Unit Testing Mock Test (Vitest + React Testing Library)

\*\*Instructions:\*\* Answer the following 20 questions based on your understanding of unit testing with Vitest and React Testing Library based on this article: https://www.robinwieruch.de/react-testing-library/

1. Which test runner does Robin recommend using with Vite?

2. What environment must be specified in Vite’s config to enable DOM testing?

3. Name two libraries you need to install alongside Vitest for React component testing.

4. In Vitest (or Jest), what’s the purpose of `describe(...)` and `it(...)` blocks?

5. What Vitest function is used to check if a value meets certain conditions (e.g. equality)?

6. Which React Testing Library function is used to render a component in tests?

7. How can you inspect the rendered DOM output during a test for debugging?

8. Which screen method do you use to select an element by its visible text?

9. Which method selects elements by their accessibility role (e.g., “button”, “textbox”)?

10. Name two other query methods besides `getByText` and `getByRole`.

11. What’s the key difference between `getBy`, `queryBy`, and `findBy` variants?

12. Which variant would you use to assert that \*no\* element is present?

13. Which variant is appropriate for elements that appear asynchronously?

14. After selecting an element, what assertion should you use to check it’s in the DOM?

15. What’s the recommended React Testing Library API for simulating typing or clicks?

16. Which lower-level API does `userEvent` wrap or improve upon?

17. How do you verify a callback handler (e.g. `onChange`) was called using Vitest?

18. If you type ten characters with `userEvent.type`, how many times does `onChange` fire?

19. When mocking `axios.get` to return data, which method lets you wait for DOM updates?

20. How do you test an error state if the API request fails (e.g. show error message)?

# ✅ Answer Key

1. Vitest

2. jsdom

3. @testing-library/react and @testing-library/jest-dom

4. Grouping and organizing tests and individual test cases

5. expect(...)

6. render(...)

7. screen.debug()

8. getByText(...)

9. getByRole(...)

10. getByLabelText, getByPlaceholderText

11. getBy throws an error if not found, queryBy returns null, findBy waits for async

12. queryBy

13. findBy

14. toBeInTheDocument()

15. userEvent

16. fireEvent

17. Use vi.fn() and assert it was called

18. 10 times

19. findBy or waitFor

20. Mock an error and assert error message is shown in the DOM