

Types of Tests

Unit	
Description	Test one unit of code (for example, a function or React component) in isolation
Notes	Mocks can be used to keep tests isolated from other functions
Example (React / web)	Test that an input box turns red when the input is invalid
Common Tools (React / Web)	Jest, Enzyme, Testing Library

Integration	
Description	Test how multiple units work together
Notes	Tests the interface between two or more units
Example (React / web)	Test that a parent component passes correct props to a child component
Common Tools (React / Web)	Jest, Enzyme, Testing Library

Functional	
Description	Test a user flow
Notes	These tests are not tightly associated with code, and instead try to simulate user experience. They could be unit tests or integration tests, depending on how many components are involved.
Example (React / web)	Enter a number into an input box for an option, and test that the app updates the total displayed on the page
Common Tools (React / Web)	Jest, Enzyme, Testing Library

Acceptance / End-to-End (E2E)	
Description	Use actual browser and server to test user experience
Notes	These are usually functional tests, though most tools allow checking code, such as rendered HTML and attributes
Example (React / web)	Open the page, log in, and submit a new order to the server. Check that the order shows in the "my orders" list
Common Tools (React / Web)	Cypress, Selenium