

## What are the different types of testing in UI?

In UI development, especially with frameworks like React, testing is typically divided into the following types:

### 1. Unit Testing (Isolated Testing)

This involves testing individual components in isolation to ensure they render correctly and behave as expected with given inputs (props, state, etc.). Tools like Jest and React Testing Library are commonly used for this.

### 2. Integration Testing

This tests how multiple components work together in a flow. It checks whether user interactions (like button clicks or form submissions) lead to the expected changes in the UI or data. It ensures that the integrated units function correctly as a group.

### 3. End-to-End (E2E) Testing

E2E testing simulates real user behavior by interacting with the entire application in a browser. Tools like **Cypress**, **Playwright**, or **Selenium** are used to test complete workflows from start to finish (e.g., logging in, filling forms, navigation). These tests validate the system as a whole.

Type	Scope	Tools	Purpose
Unit Testing	Individual component	Jest, RTL	Test logic/rendering in isolation
Integration Testing	Multiple components	Jest, RTL, MSW	Test interaction and data flow
End-to-End Testing	Full app in browser	Cypress, Playwright, Selenium	Test user journeys end-to-end

## What is Enzyme?

**Enzyme** is a JavaScript testing library for React developed by Airbnb. It provides utilities to render components (shallow, mount) and inspect their output, state, and props. It allows testing component internals and simulating events like clicks or form submissions. While powerful, Enzyme is now outdated and lacks full support for React 17+ and functional components with hooks. For modern React projects, **React Testing Library** is preferred as it promotes testing components from the user's perspective.

### Enzyme vs React Testing Library

**Enzyme** and **React Testing Library (RTL)** are both used for testing React components, but they follow different philosophies. Enzyme is **implementation-focused**, allowing access to

component internals like state, props, and lifecycle methods. It supports shallow rendering, which is useful for unit testing in isolation. In contrast, **React Testing Library** is **user-centric** — it encourages testing components the way users interact with them (e.g., using `getByText`, `getByRole`), improving test reliability. RTL works better with modern React (hooks, context, functional components) and has stronger community support. For new projects, **RTL is the recommended choice**, while Enzyme may still be used in legacy codebases.

#### ✅ What is Jest and why do we use it?

**Jest** is a JavaScript testing framework developed by Facebook, widely used for testing **React applications** and JavaScript code in general. It comes bundled with powerful features like **test runners**, **assertion libraries**, **mocking**, and **code coverage**, all without needing extra configuration. Jest is fast, supports parallel test execution, and works seamlessly with tools like **React Testing Library**. It's used to write **unit**, **integration**, and even **snapshot tests**, ensuring that components behave as expected and changes don't break existing functionality.