Assignment 13 - Time for the Test

Q1 What are different types for testing ?

⇒ Software testing is the process of finding errors in a product, whether it be a mobile or web application. Errors include bugs in the code, missing requirements, glitches, and more. Software testing can also determine whether the outcome when engaging with the application differs from the expectation.

→ Manual Testing:
It is done in person, by clicking through the application or interacting with the software and APIs with the appropriate tooling. This is very expensive since it requires someone to setup an environment and execute the tests themselves, and it can be prone to human error as the tester might make typos or omit steps in the test script.

→ Automated Testing:
It is performed by a machine that executes a test script that was written in advance. These tests can vary in complexity, from checking a single method in a class to making sure that performing a sequence of complex actions in the UI leads to the same results. It's much more robust and reliable than manual tests - but the quality of your automated tests depends on how well your tests scripts have been written.

There are different types of tests, some popular types are described below:

1) Unit tests:
They consist in testing individual methods and functions of the classes, components or modules used by your software. Unit tests are generally quite cheap to automate and can run very quickly by a continuous integration server.

2) Integration tests:
It verify that different modules or services used by your application work well together. For example, it can be testing the interaction with the database or making sure that microservices work together as expected. These types of tests are more expensive to run as they require multiple parts of the application to be up and running.

3) End-to-end tests:
It replicates a user behavior with the software in a complete application environment. It verifies that various user flows work as expected and can be as simple as loading a web page or logging in or much more complex scenarios verifying email notification, online payments, etc.

End-to-end tests are very useful, but they're expensive to perform and can be hard to maintain when they're automated.

Q2 What is Enzyme?

⇒ Enzyme is a JavaScript testing utility that was created by Airbnb. It is designed to make it easier to test React components by providing a set of tools for asserting, manipulating, and traversing React components output. Enzyme simplifies the testing process by allowing developers to interact with components as if they were in the browser.

Enzyme provides three different rendering methods: shallow rendering, full DOM rendering, and static rendering. Shallow rendering allows developers to render only the component they are testing, while full DOM rendering renders the entire component tree. Static rendering is used to render components without a DOM.

Q3 Enzym vs React Testing Library

⇒

	React Testing Library	Enzyme
1)	Developed in 2017 by Kent C. Dodds and the React community to encourage testing in a way that reflects how users interact with the application	Developed by Airbnb in 2015 to provide a comprehensive testing utility for React components.
2)	Uses a lightweight DOM testing library	Offers both shallow rendering and full rendering capabilities
3)	RTL Uses queries like getBy, queryBy, findBy for more user-centric testing	Provides utility functions for DOM manipulation and querying, Enabling precise control
4)	RTL Focuses on testing components in a more isolated manner	Allows shallow rendering to isolate the tested component. It can also perform full rendering.
5)	Limited direct access to component state and props	Enzyme allows direct access to component state and props
6)	Encourages making assertions based on user interaction and visible changes in the DOM.	Supports various assertions, including snapshot testing for both shallow and deep tests
7)	Gained popularity for its simplicity and user-centric approach. Widely used in React community	Popular for its longer presence, extensive community and comprehensive documentation
8)	No direct integration with Jest; can be used independently or with other test runners.	Officially integrated with Jest, making it seamless for projects using Jest as the test runner.

Q4 What is Jest and why do we use it?

⇒ Jest is a JavaScript Testing Framework by Facebook. It is used mostly commonly for Unit Testing. It is when you provide input to a unit of code (usually, a function) and match the output with the expected output.

Jest is a Node environment attempting to mock out a real DOM via JsDOM. It's essentially a hybrid framework built on JavaScript, designed majorly to work with React and React Native-based web applications. It can validate almost everything around JavaScript, especially the browser rendering of web applications.