**Objectives**

* Explain the need for Unit Testing in React
* Ensures that individual components work as expected.
* Helps in identifying bugs early in development.
* Makes code more maintainable and reliable.
* Facilitates refactoring without breaking functionality.
* Provides confidence while integrating new features.
* Improves code quality through test-driven development (TDD).
* Encourages modular, reusable component design.
* Working with Jest and Enzyme in React

**Jest**

* Default testing framework included with Create React App.
* Supports snapshot testing, mocking, and coverage reporting.
* Easy to configure and fast to execute tests.

**Enzyme**

* A testing utility developed by Airbnb for React.
* Allows shallow rendering, full DOM rendering, and static rendering.
* Provides APIs like shallow(), mount(), and render() to test components.
* List the types of Router Components

|  |  |
| --- | --- |
| Router Component | Description |
| BrowserRouter | Uses HTML5 history API; preferred for modern web apps |
| HashRouter | Uses URL hash (#) for routing; works better for legacy systems |
| MemoryRouter | Stores the history in memory; useful for testing and non-browser environments |
| StaticRouter | Used for server-side rendering (SSR) |
| NativeRouter | Used with React Native apps |

## **Notes**

Estimated time to complete this lab: **90 minutes.**

My Academy team at Cognizant want to create a dashboard containing the details of ongoing and completed cohorts. A react application is created which displays the detail of the cohorts using React component. You are assigned the task of unit testing the component to ensure it’s free of bugs.

Download the following attachment to find the application code.



1. Unzip the attachment and open it in visual studio code.
2. Open the terminal and execute the following command to restore the node modules



Figure : Packages Restore

1. Execute the following commands to install **Enzyme** and **Enzyme test adapter**

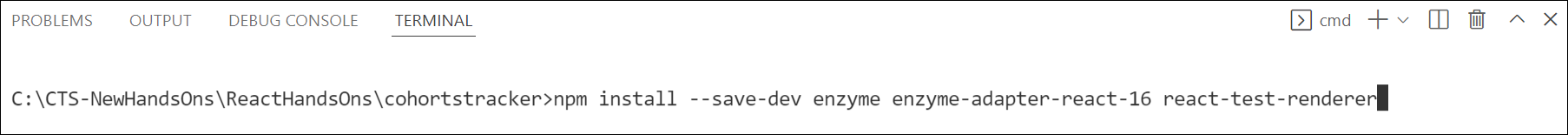


Figure : Installing Enzyme support

1. Go to file explorer and modify the **setupTests.js** file to make it look like the following

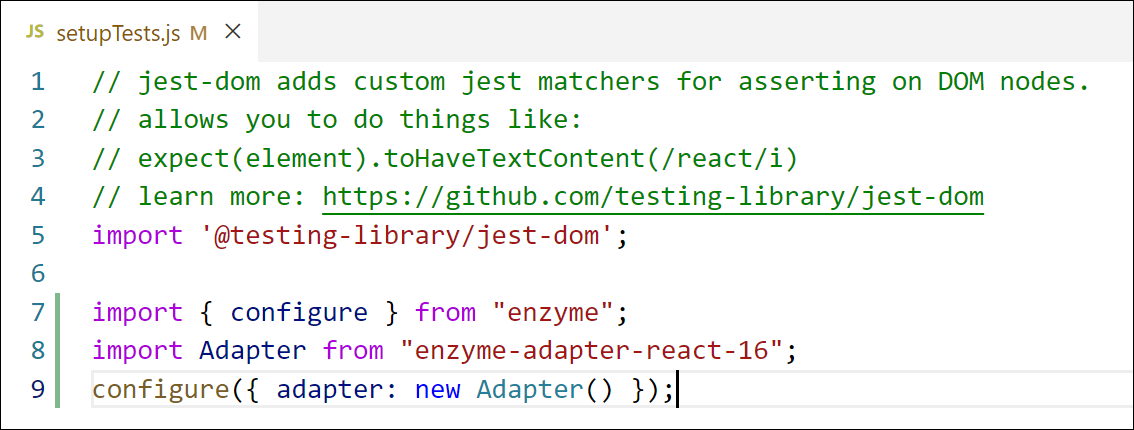


Figure : Configuring Enzyme support

1. Add a new unit test file with the name as **CohortDetails.test.js** to write the unit tests for the CohortDetails component defined inside **CohortDetails.js** file.
2. Import the following into the unit test
   1. mount, shallow from Enzyme library
   2. CohortDetails component
   3. CohortData array from Cohort.js file
3. Define a test suite using **describe()** and name it as “Cohort Details Component”
4. Define the following unit tests using **test()**

Table : Unit Test - 1

|  |  |
| --- | --- |
| Test - 1 | Name: should create the component |
| A unit test which will load the CohortDetails component in isolation | |

Table : Unit Test - 2

|  |  |
| --- | --- |
| Test - 2 | Name: should initialize the props |
| This test should mount the component and assign a cohort object to the props. Using matchers ensure that the props are assigned a given cohort object | |

Table : Unit Test - 3

|  |  |
| --- | --- |
| Test - 3 | Name: should display cohort code in h3 |
| This test should mount the component and search for an h3 element using find() and verify that it displays the proper cohort code of the object passed in props | |

Table : Unit Test - 4

|  |  |
| --- | --- |
| Test - 4 | Name: should always render same html |
| This test should be responsible for testing the snapshot of the component. | |

1. Build and run the tests using *npm test* command.

Handson:

