**PlayWright (Day 1)**

**Installing Playwright -** *npm init playwright@latest*

**Runs the end-to-end tests -** *npx playwright test*

**Starts the interactive UI mode -** *npx playwright test --ui*

**Runs the tests only on Desktop Chrome** - *npx playwright test --project=chromium*

**Runs the tests in a specific file** - *npx playwright test example*

**Runs the tests in debug mode** - *npx playwright test –debug*

**Runs the particular test** - *npx playwright test -g EndToEnd*

**Runs to show the report** *- npx playwright show-report*

--------------------------------------------------------------------------------------------------------------

**node\_modules/**:  
The node\_modules folder contains all the installed dependencies and packages for your project, including Playwright, other libraries, and their dependencies, managed via npm or yarn.

**playwright-report/**:  
This directory typically holds Playwright's generated reports, such as HTML reports, after running tests, helping you visualize test results and debug failures.

**test-results/**:  
The test-results folder stores the raw test output data (such as JSON files) generated during test runs, which can be useful for integration with CI/CD pipelines and further analysis.

**tests/**:  
This folder contains the actual test scripts written for the application. It includes the Playwright test files that automate and validate the application’s behavior.

**tests-examples/**:  
The tests-examples folder usually contains example or template test files that demonstrate how to write tests with Playwright, often used for learning purposes or as starting points for real tests.

**.gitignore**:  
The .gitignore file specifies which files and directories (such as node\_modules/, build artifacts, or log files) should not be tracked by Git, keeping the repository clean and efficient.

**package-lock.json**:  
This file locks down the exact versions of dependencies (including nested ones) to ensure that the project has consistent and reproducible installations across different environments.

**playwright.config.ts**:  
The playwright.config.ts file is used to configure Playwright’s settings for the test suite, including test timeouts, retries, base URL, and other options that affect how Playwright runs the tests.

--------------------------------------------------------------------------------------------------------------

**test()**

* **Description**: Defines a test case in Playwright, taking a test name and an async function with test actions.
* **Example**: test('Page Title', async ({ page }) => { ... })

**async**

* **Description**: Declares an asynchronous function that allows the use of await within the function.
* **Example**: const someFunction = async () => { await page.goto('https://example.com'); }

**page**

* **Description**: Represents the browser page being tested in Playwright, providing methods for interacting with the page.
* **Example**: await page.goto('https://www.typescriptlang.org/');

**await**

* **Description**: Pauses execution until the promise resolves, used with async functions to wait for actions like page navigation or element interactions.
* **Example**: await page.click('button#submit');

**locator**

* **Description**: A Playwright method to locate elements on the page, used for interacting with and asserting on UI elements.
* **Example**: const button = page.locator('button#submit'); await button.click();