***Day 2***

1. **To execut a particular test**

npx playwright test -g "has title"

1. **To Rerun a failed test**

npx playwright test --last-failed

1. **To run the project automation on chromium and enable trace on as well:**

npx playwright test --project chromium --trace on

1. **Playwright's defineConfig**

To set up how my tests should run. This config gives me flexibility across browsers, environment control, retries, and even geolocation emulation.

1. **General Settings:**

|  |
| --- |
| **testDir**: Test files are located in the **./tests** directory.  **fullyParallel**: Enables full parallelism across test files.  **forbidOnly**: Ensures **.only is not left** accidentally in source (enabled on CI).  **retries**: Retries failed tests up to **2 times on CI**, none locally.  **workers**: Runs with **a single worker on CI** to avoid parallel execution issues.  **reporter**: Uses the **'html' reporter** for test results. |

1. **use block (Shared Settings for All Projects):**

use: {

    /\* Base URL to use in actions like `await page.goto('/')`. \*/

    // baseURL: 'http://127.0.0.1:3000',

    baseURL: 'https://playwright.dev/',

    /\* Collect trace when retrying the failed test. See

https://playwright.dev/docs/trace-viewer \*/

    trace: 'on-first-retry',

    screenshot: 'only-on-failure',

    video: 'on-first-retry'

    // Network configs

    acceptDownloads: false,

    extraHTTPHeaders: {

      'X-My-Header': 'value',

    },

    httpCredentials: {

      username: 'user',

      password: 'pass'

    },

    ignoreHTTPSErrors: true,

    offline: true,

    proxy: {

      server: '',

      bypass: 'localhost'

    }

    // Emulation options

    colorScheme: 'dark',

    geolocation: { longitude: 12.343535, latitude: 45.56575 },

    locale: 'en-GB',

    permissions: ['geolocation'],

    timezoneId: '',

    viewport: { width: 1280, height: 720 }

  },

1. **Browser Projects**

/\* Configure projects for major browsers \*/

  projects: [

    {

      name: 'chromium',

      use: { ...devices['Desktop Chrome'] },

    },

    {

      name: 'firefox',

      use: { ...devices['Desktop Firefox'] },

    },

    {

      name: 'webkit',

      use: { ...devices['Desktop Safari'] },

    },

    /\* Test against mobile viewports. \*/

    {

      name: 'Mobile Chrome',

      use: { ...devices['Pixel 5'] },

    },

    {

      name: 'Mobile Safari',

      use: { ...devices['iPhone 12'] },

    },

    /\* Test against branded browsers. \*/

    {

      name: 'Microsoft Edge',

      use: { ...devices['Desktop Edge'], channel: 'msedge' },

    },

    {

      name: 'Google Chrome',

      use: { ...devices['Desktop Chrome'], channel: 'chrome' },

    },

1. **Dev Server (Commented-out):**

Option to run a dev server (npm run start) before running tests.

Can be reused locally but starts fresh on CI.

/\* Run your local dev server before starting the tests \*/

  webServer: {

    command: 'npm run start',

    url: 'http://127.0.0.1:3000',

    reuseExistingServer: !process.env.CI,

  },

1. **Special Test States:**

* **test.only:** Focuses the test runner only on this test (focus on this test case).
* **test.skip**: Skips the skipped test case.
* **Conditional skip**:
  + Skips a test for chromium only using test.skip(browserName === 'chromium').
* **test.beforeAll**
  + **Pre-test setup logic before all tests run.**
* **Handles**:
  + Waiting for and interacting with a popup (waitForEvent('popup')).
  + Inside the popup, finds a close button and clicks it.
  + Listens for browser dialogs like alert(), confirm(), prompt() and accepts them automatically.