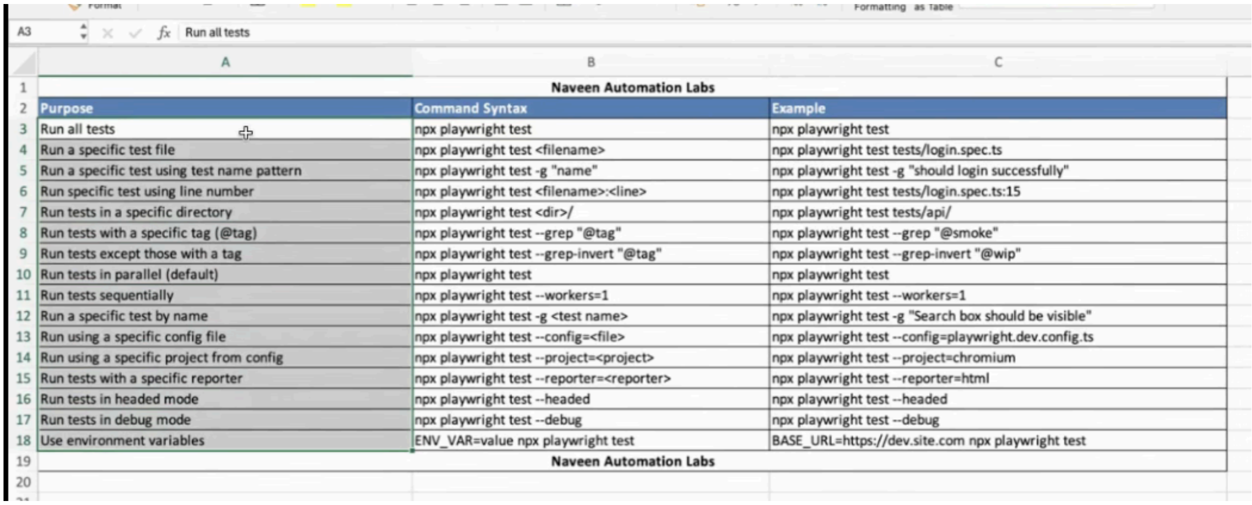
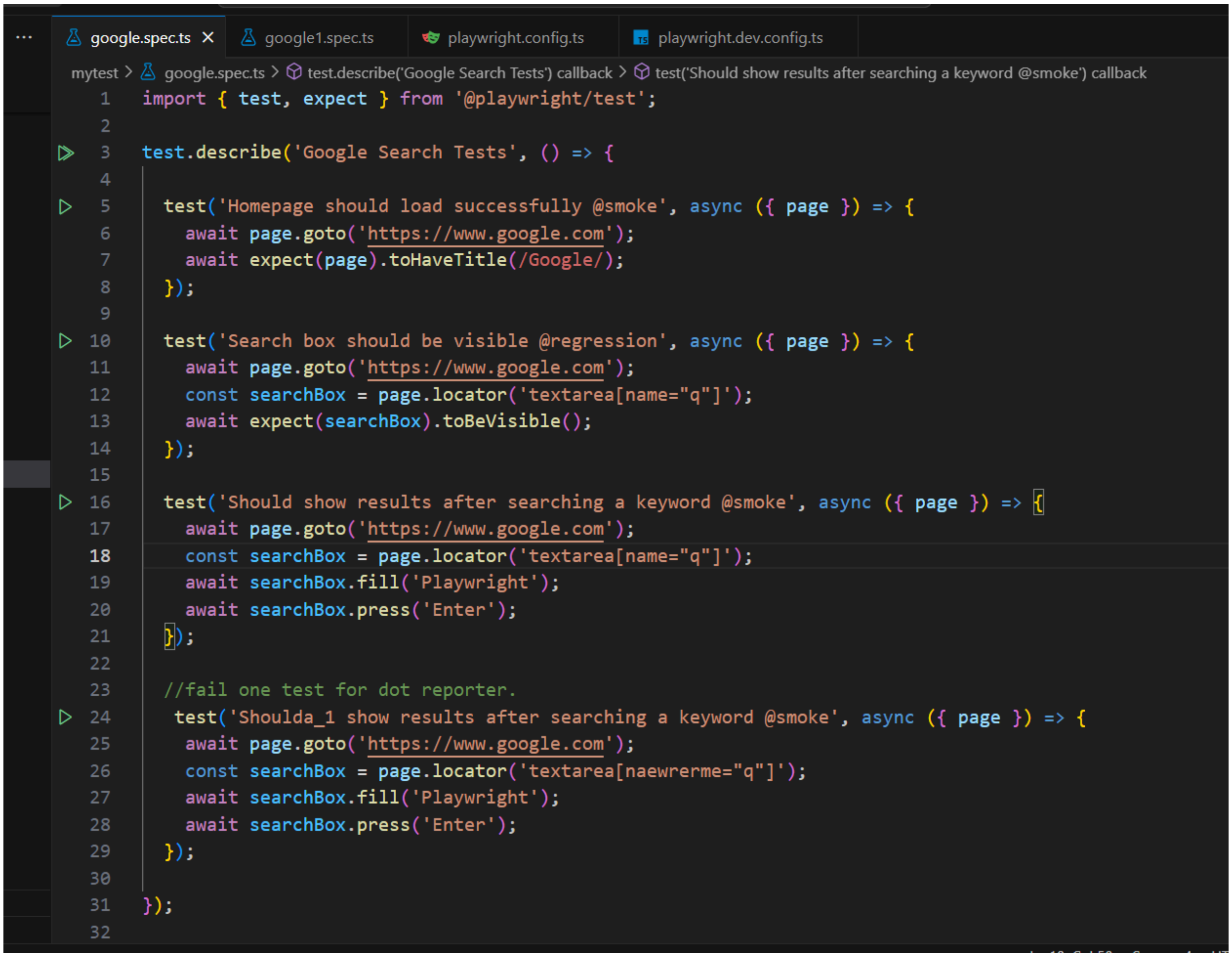
Cheat sheet –


Formatting as Table
A3
Run all tests
A
B
C

Naveen Automation Labs

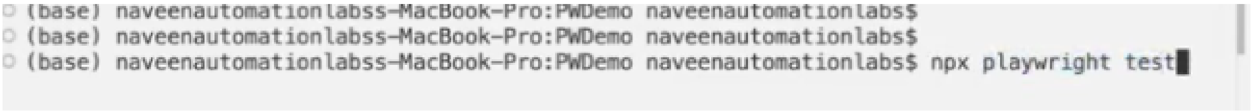
Purpose	Command Syntax	Example
Run all tests	npx playwright test	npx playwright test
Run a specific test file	npx playwright test <filename>	npx playwright test tests/login.spec.ts
Run a specific test using test name pattern	npx playwright test -g "name"	npx playwright test -g "should login successfully"
Run specific test using line number	npx playwright test <filename>:<line>	npx playwright test tests/login.spec.ts:15
Run tests in a specific directory	npx playwright test <dir>/	npx playwright test tests/api/
Run tests with a specific tag (@tag)	npx playwright test --grep "@tag"	npx playwright test --grep "@smoke"
Run tests except those with a tag	npx playwright test --grep-invert "@tag"	npx playwright test --grep-invert "@wip"
Run tests in parallel (default)	npx playwright test	npx playwright test
Run tests sequentially	npx playwright test --workers=1	npx playwright test --workers=1
Run a specific test by name	npx playwright test -g <test name>	npx playwright test -g "Search box should be visible"
Run using a specific config file	npx playwright test --config=<file>	npx playwright test --config=playwright.dev.config.ts
Run using a specific project from config	npx playwright test --project=<project>	npx playwright test --project=chromium
Run tests with a specific reporter	npx playwright test --reporter=<reporter>	npx playwright test --reporter=html
Run tests in headed mode	npx playwright test --headed	npx playwright test --headed
Run tests in debug mode	npx playwright test --debug	npx playwright test --debug
Use environment variables	ENV_VAR=value npx playwright test	BASE_URL=https://dev.site.com npx playwright test

Naveen Automation Labs


google.spec.ts
google1.spec.ts
playwright.config.ts
playwright.dev.config.ts
mytest > google.spec.ts > test.describe('Google Search Tests') callback > test('Should show results after searching a keyword @smoke') callback

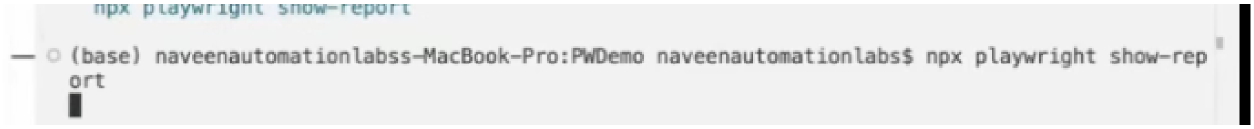
```
import { test, expect } from '@playwright/test';

test.describe('Google Search Tests', () => {

  test('Homepage should load successfully @smoke', async ({ page }) => {
    await page.goto('https://www.google.com');
    await expect(page).toHaveTitle(/Google/);
  });

  test('Search box should be visible @regression', async ({ page }) => {
    await page.goto('https://www.google.com');
    const searchBox = page.locator('textarea[name="q"]');
    await expect(searchBox).toBeVisible();
  });

  test('Should show results after searching a keyword @smoke', async ({ page }) => {
    await page.goto('https://www.google.com');
    const searchBox = page.locator('textarea[name="q"]');
    await searchBox.fill('Playwright');
    await searchBox.press('Enter');
  });

  //fail one test for dot reporter.
   test('Shoulda_1 show results after searching a keyword @smoke', async ({ page }) => {
    await page.goto('https://www.google.com');
    const searchBox = page.locator('textarea[naewrerme="q"]');
    await searchBox.fill('Playwright');
    await searchBox.press('Enter');
  });

});

```

Run all tests-



```
○ (base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$
○ (base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$
○ (base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$ npx playwright test
```

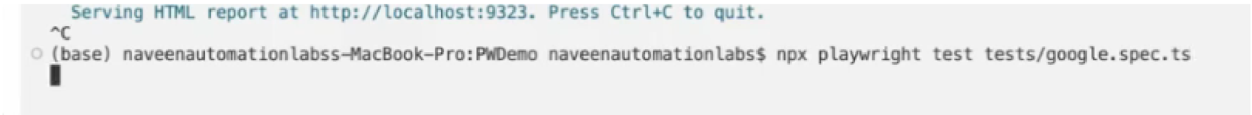
See report–



```
    npx playwright show-report

○ (base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$ npx playwright show-rep
ort
```

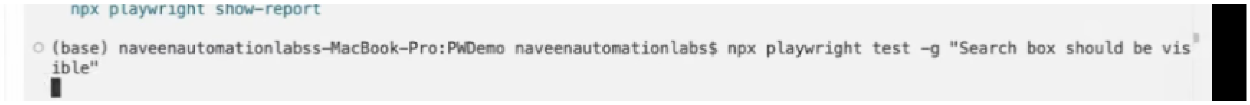
Default is headless and parallel.

Run specific file–



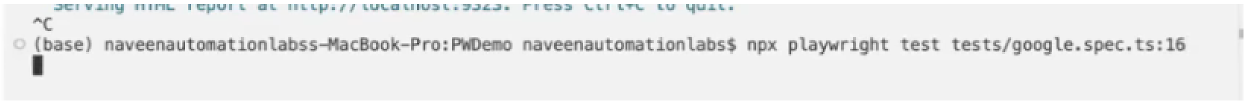
```
   Serving HTML report at http://localhost:9323. Press Ctrl+C to quit.
 ^C
○ (base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$ npx playwright test tests/google.spec.ts
```

Run specific test from file –



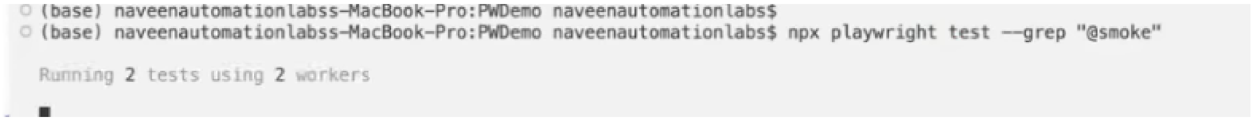
```
   npx playwright show-report

○ (base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$ npx playwright test -g "Search box should be vis
 ible"
```

Run test based on line number –



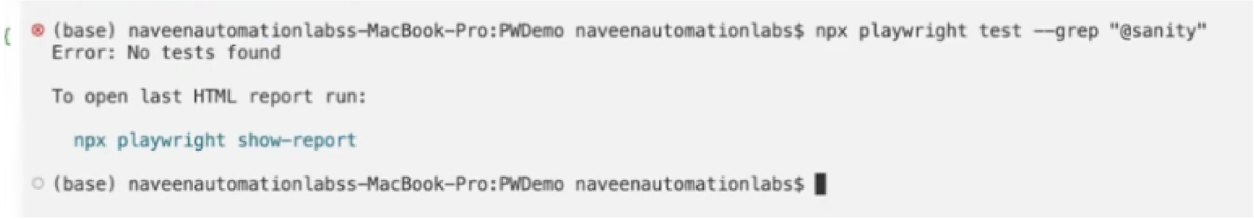
```
   Serving HTML report at http://localhost:9323. Press Ctrl+C to quit.
 ^C
○ (base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$ npx playwright test tests/google.spec.ts:16
```

Run the specific test with annotation–



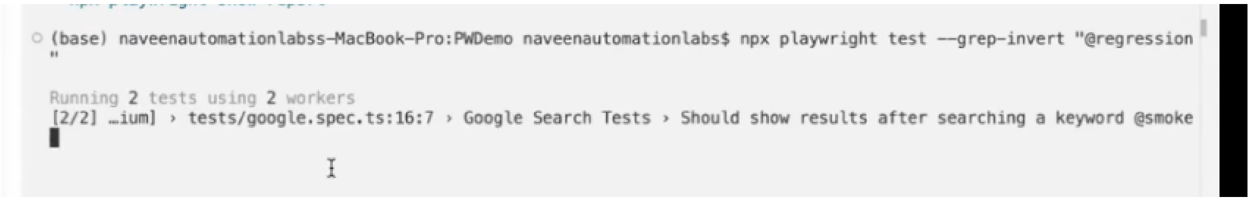
```
○ (base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$
○ (base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$ npx playwright test --grep "@smoke"

  Running 2 tests using 2 workers
```

When tag doesnt exist–



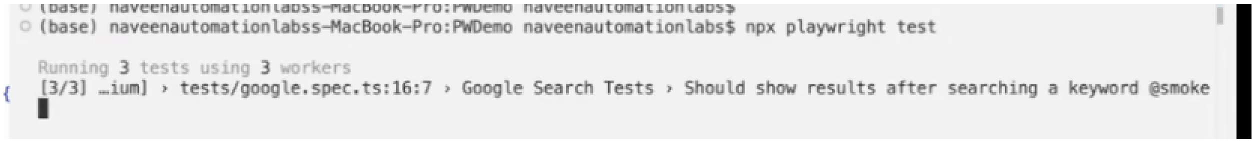
```
⊗ (base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$ npx playwright test --grep "@sanity"
  Error: No tests found

  To open last HTML report run:

    npx playwright show-report

○ (base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$
```

Run test other than tag mentioned–



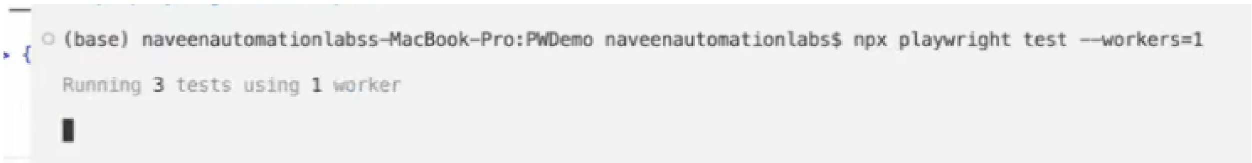
```
○ (base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$ npx playwright test --grep-invert "@regression
"

  Running 2 tests using 2 workers
  [2/2] …ium] › tests/google.spec.ts:16:7 › Google Search Tests › Should show results after searching a keyword @smoke
```

All test run in parallel and headless when running this –



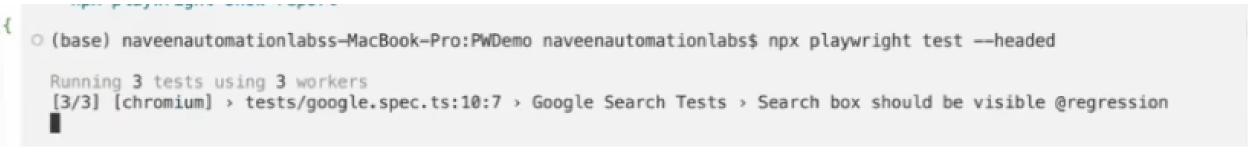
```
○ (base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$
○ (base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$ npx playwright test

  Running 3 tests using 3 workers
  [3/3] …ium] › tests/google.spec.ts:16:7 › Google Search Tests › Should show results after searching a keyword @smoke
```

Run all test using one worker-
Useful for sequential runs.



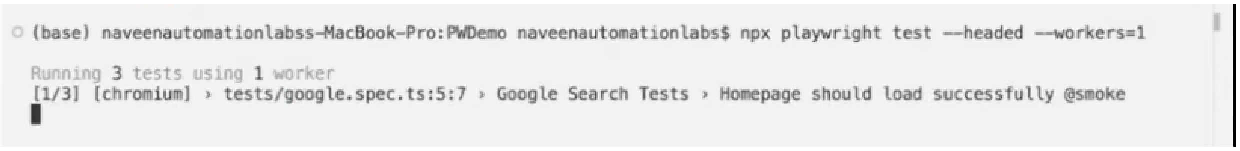
```
○ (base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$ npx playwright test --workers=1

  Running 3 tests using 1 worker
```

By default all test runs headless-
To run in headed.



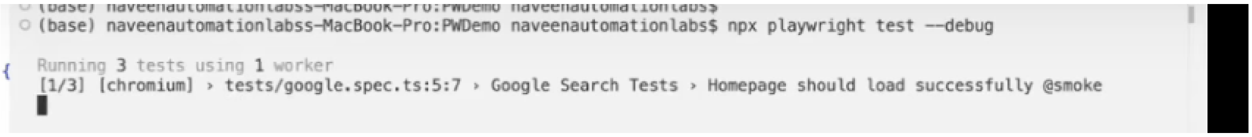
```
○ (base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$ npx playwright test --headed

  Running 3 tests using 3 workers
  [3/3] [chromium] › tests/google.spec.ts:10:7 › Google Search Tests › Search box should be visible @regression
```

Club multiple options–



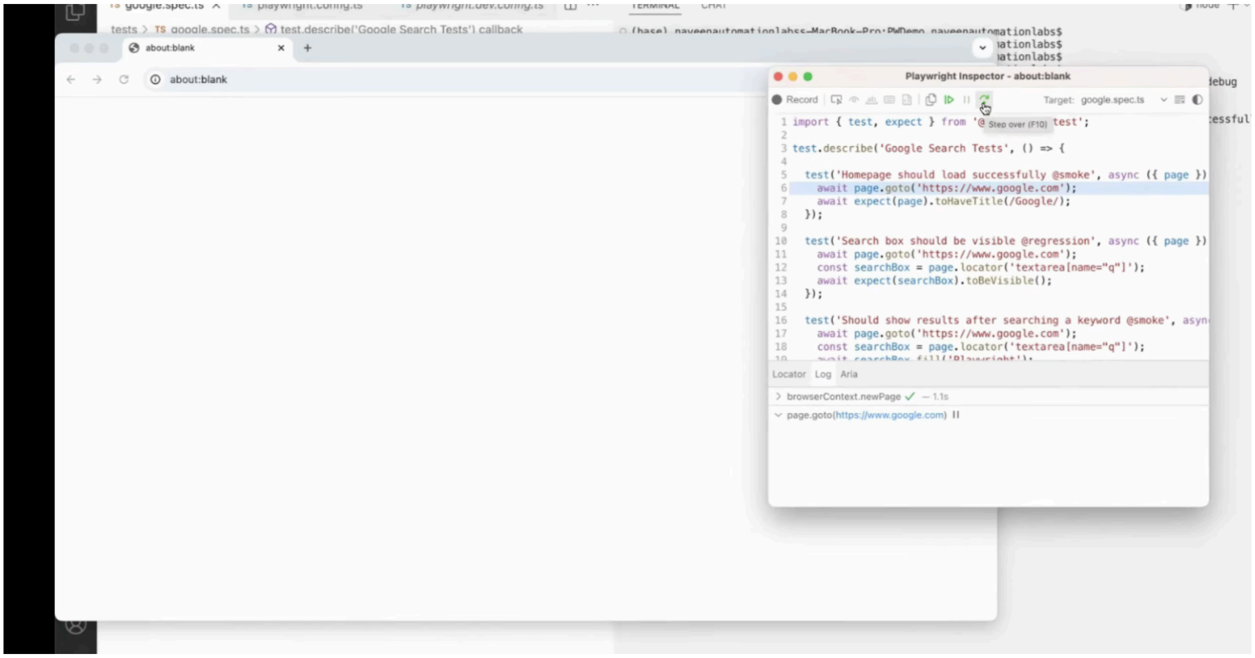
```
○ (base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$ npx playwright test --headed --workers=1

  Running 3 tests using 1 worker
  [1/3] [chromium] › tests/google.spec.ts:5:7 › Google Search Tests › Homepage should load successfully @smoke
```

Debug mode-



```
○ (base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$
○ (base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$ npx playwright test --debug

  Running 3 tests using 1 worker
  [1/3] [chromium] › tests/google.spec.ts:5:7 › Google Search Tests › Homepage should load successfully @smoke
```

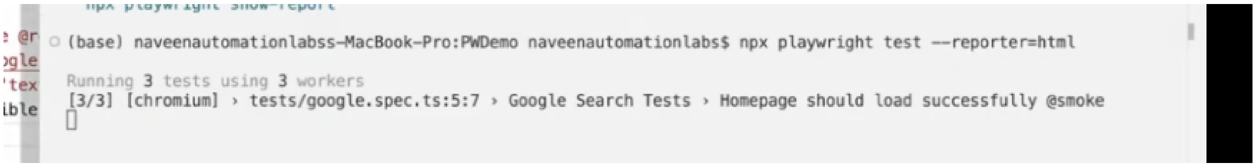
Browser plus code opened.



```
import { test, expect } from '@ Step over (F10) test';

test.describe('Google Search Tests', () => {

  test('Homepage should load successfully @smoke', async ({ page })
    await page.goto('https://www.google.com');
    await expect(page).toHaveTitle(/Google/);
  });

  test('Search box should be visible @regression', async ({ page })
    await page.goto('https://www.google.com');
    const searchBox = page.locator('textarea[name="q"]');
    await expect(searchBox).toBeVisible();
  });

  test('Should show results after searching a keyword @smoke', asyn
    await page.goto('https://www.google.com');
    const searchBox = page.locator('textarea[name="q"]');
```

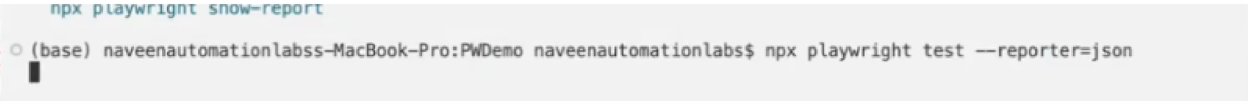
```
> browserContext.newPage ✓ — 1.1s
˅ page.goto(https://www.google.com) II
```

Generate html report –



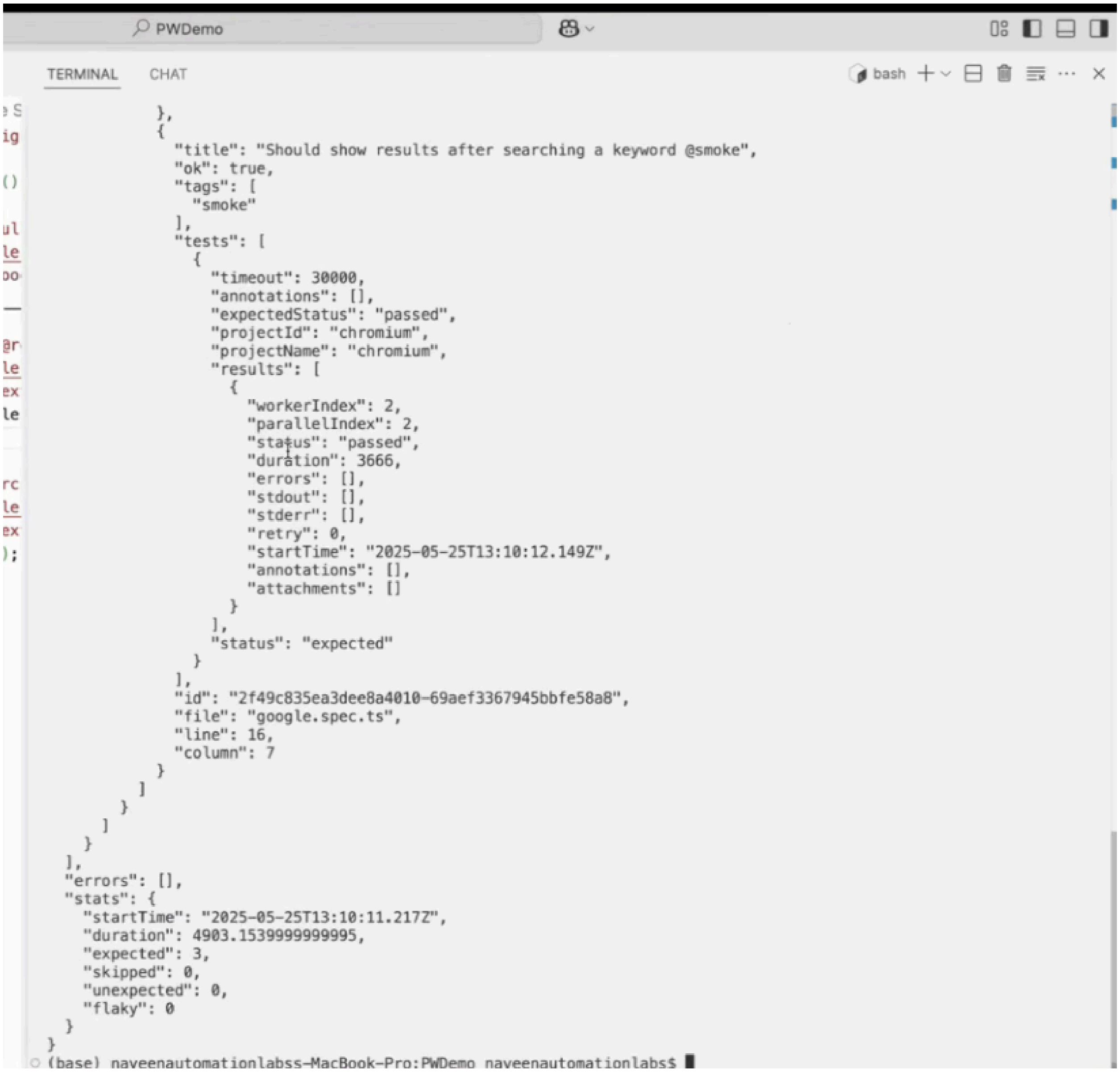
```
(base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$ npx playwright test --reporter=html

Running 3 tests using 3 workers
[3/3] [chromium] › tests/google.spec.ts:5:7 › Google Search Tests › Homepage should load successfully @smoke
```

Report in json format–



```
  npx playwright show-report

(base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$ npx playwright test --reporter=json
```

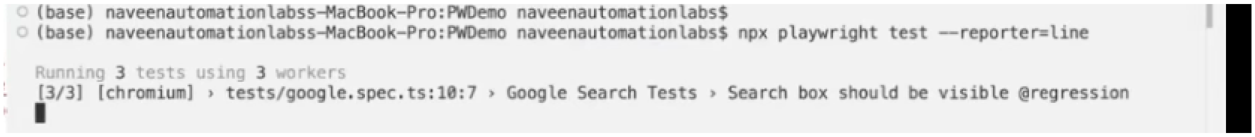
Report comes in terminal only.



```
            },
            {
              "title": "Should show results after searching a keyword @smoke",
              "ok": true,
              "tags": [
                "smoke"
              ],
              "tests": [
                {
                  "timeout": 30000,
                  "annotations": [],
                  "expectedStatus": "passed",
                  "projectId": "chromium",
                  "projectName": "chromium",
                  "results": [
                    {
                      "workerIndex": 2,
                      "parallelIndex": 2,
                      "status": "passed",
                      "duration": 3666,
                      "errors": [],
                      "stdout": [],
                      "stderr": [],
                      "retry": 0,
                      "startTime": "2025-05-25T13:10:12.149Z",
                      "annotations": [],
                      "attachments": []
                    }
                  ],
                  "status": "expected"
                }
              ],
              "id": "2f49c835ea3dee8a4010-69aef3367945bbfe58a8",
              "file": "google.spec.ts",
              "line": 16,
              "column": 7
            }
          ]
        }
      ]
    }
  ],
  "errors": [],
  "stats": {
    "startTime": "2025-05-25T13:10:11.217Z",
    "duration": 4903.1539999999995,
    "expected": 3,
    "skipped": 0,
    "unexpected": 0,
    "flaky": 0
  }
}
```

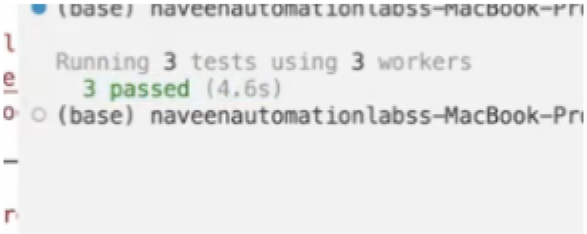
```
○ (base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$ █
```

Line reporter–



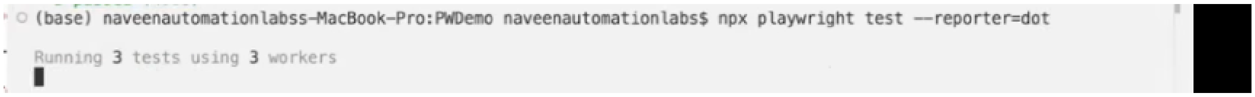
```
○ (base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$
○ (base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$ npx playwright test --reporter=line

  Running 3 tests using 3 workers
  [3/3] [chromium] › tests/google.spec.ts:10:7 › Google Search Tests › Search box should be visible @regression
  █
```

Terminal after run–

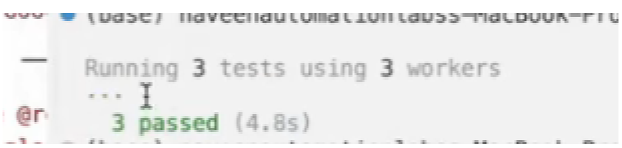


```
(base) naveenautomationlabss-MacBook-Pr

  Running 3 tests using 3 workers
    3 passed (4.6s)
(base) naveenautomationlabss-MacBook-Pr
```

Dot reporter–

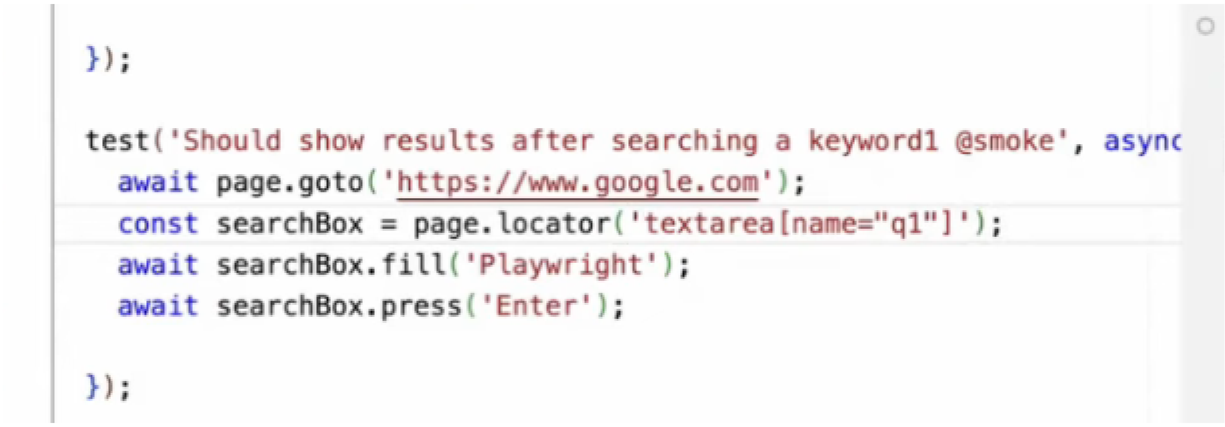


```
(base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$ npx playwright test --reporter=dot

Running 3 tests using 3 workers
```

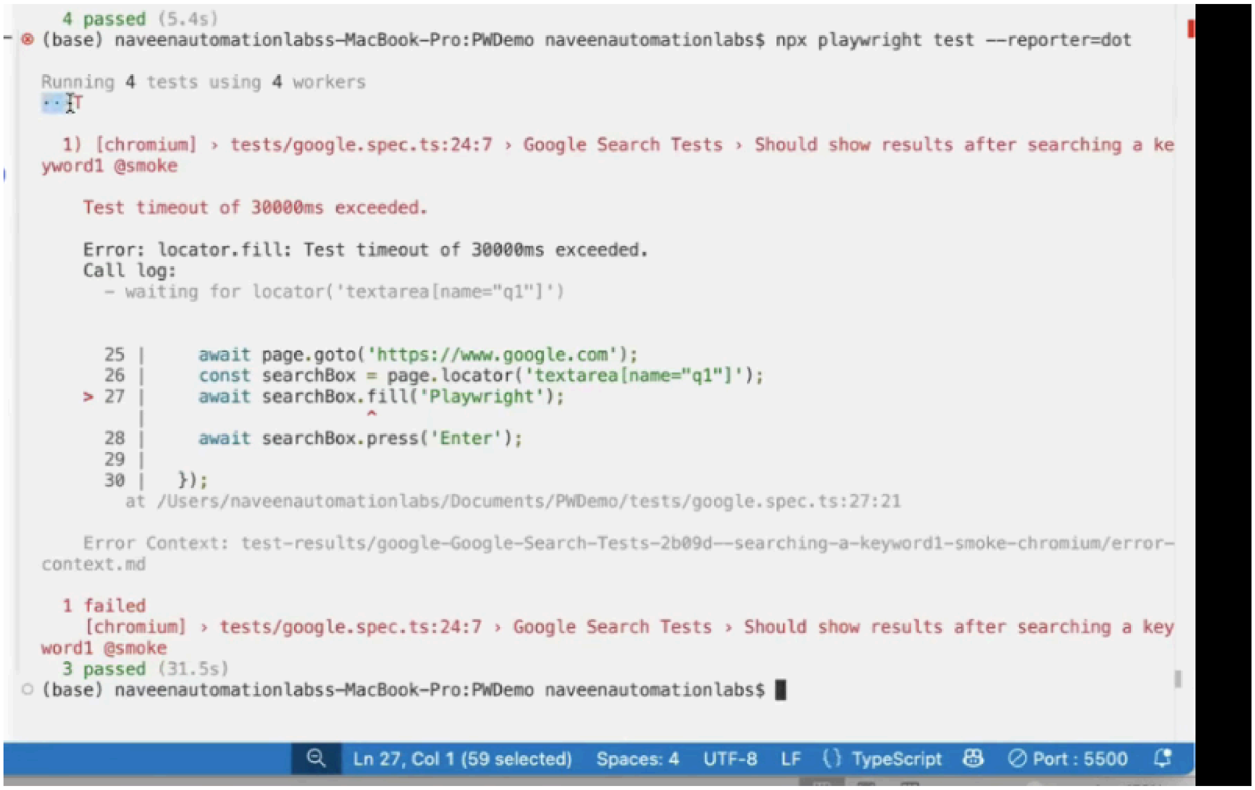


```
(base) naveenautomationlabss-MacBook-Pro

  Running 3 tests using 3 workers
  ···
    3 passed (4.8s)
```

Fail test-
It shows dots for passed and for failed shows log. T means timeout exceeded.

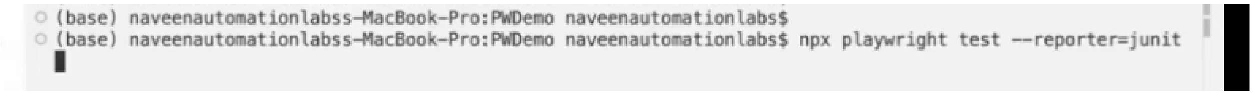


```
});

test('Should show results after searching a keyword1 @smoke', async
  await page.goto('https://www.google.com');
  const searchBox = page.locator('textarea[name="q1"]');
  await searchBox.fill('Playwright');
  await searchBox.press('Enter');

});
```



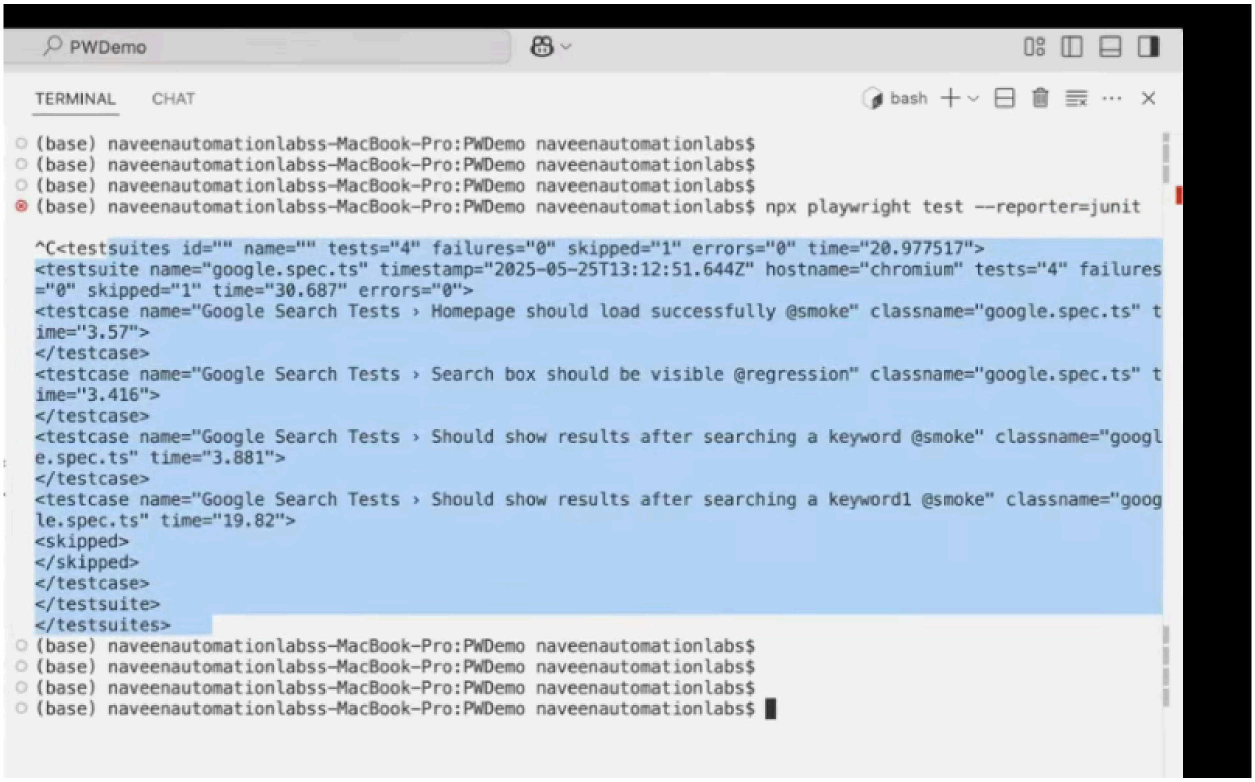
```
  4 passed (5.4s)
(base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$ npx playwright test --reporter=dot

Running 4 tests using 4 workers
··T

  1) [chromium] › tests/google.spec.ts:24:7 › Google Search Tests › Should show results after searching a ke
yword1 @smoke

    Test timeout of 30000ms exceeded.

    Error: locator.fill: Test timeout of 30000ms exceeded.
    Call log:
      - waiting for locator('textarea[name="q1"]')


      25 |     await page.goto('https://www.google.com');
      26 |     const searchBox = page.locator('textarea[name="q1"]');
    > 27 |     await searchBox.fill('Playwright');
         |                     ^
      28 |     await searchBox.press('Enter');
      29 |
      30 |   });
        at /Users/naveenautomationlabs/Documents/PWDemo/tests/google.spec.ts:27:21

    Error Context: test-results/google-Google-Search-Tests-2b09d--searching-a-keyword1-smoke-chromium/error-
context.md

  1 failed
    [chromium] › tests/google.spec.ts:24:7 › Google Search Tests › Should show results after searching a key
word1 @smoke
  3 passed (31.5s)
(base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$
```

Junit report –



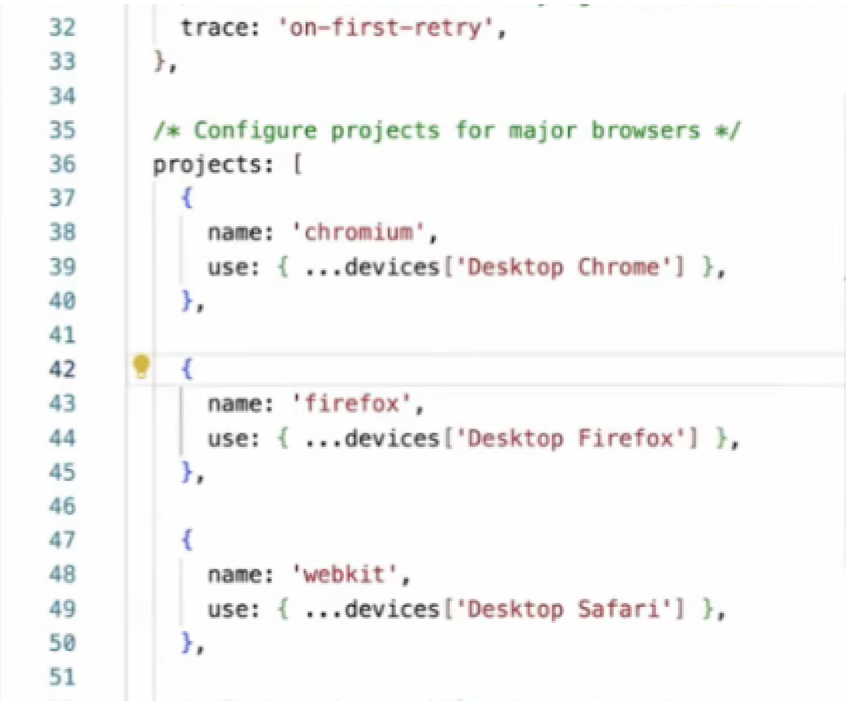
```
(base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$
(base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$ npx playwright test --reporter=junit
```

In terminal itself xml format comes.



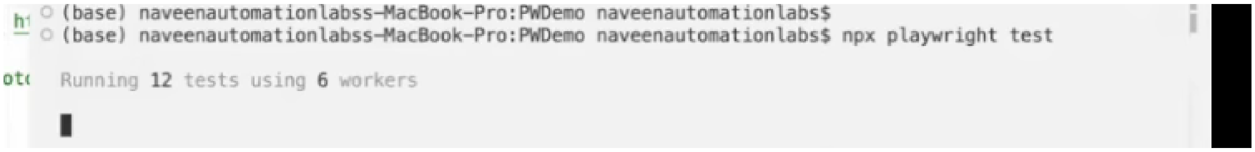
```
(base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$
(base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$
(base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$
(base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$ npx playwright test --reporter=junit

^C<testsuites id="" name="" tests="4" failures="0" skipped="1" errors="0" time="20.977517">
<testsuite name="google.spec.ts" timestamp="2025-05-25T13:12:51.644Z" hostname="chromium" tests="4" failures
="0" skipped="1" time="30.687" errors="0">
<testcase name="Google Search Tests › Homepage should load successfully @smoke" classname="google.spec.ts" t
ime="3.57">
</testcase>
<testcase name="Google Search Tests › Search box should be visible @regression" classname="google.spec.ts" t
ime="3.416">
</testcase>
<testcase name="Google Search Tests › Should show results after searching a keyword @smoke" classname="googl
e.spec.ts" time="3.881">
</testcase>
<testcase name="Google Search Tests › Should show results after searching a keyword1 @smoke" classname="goog
le.spec.ts" time="19.82">
<skipped>
</skipped>
</testcase>
</testsuite>
</testsuites>
(base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$
(base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$
(base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$
(base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$
```

Enable all projects in pw config file –



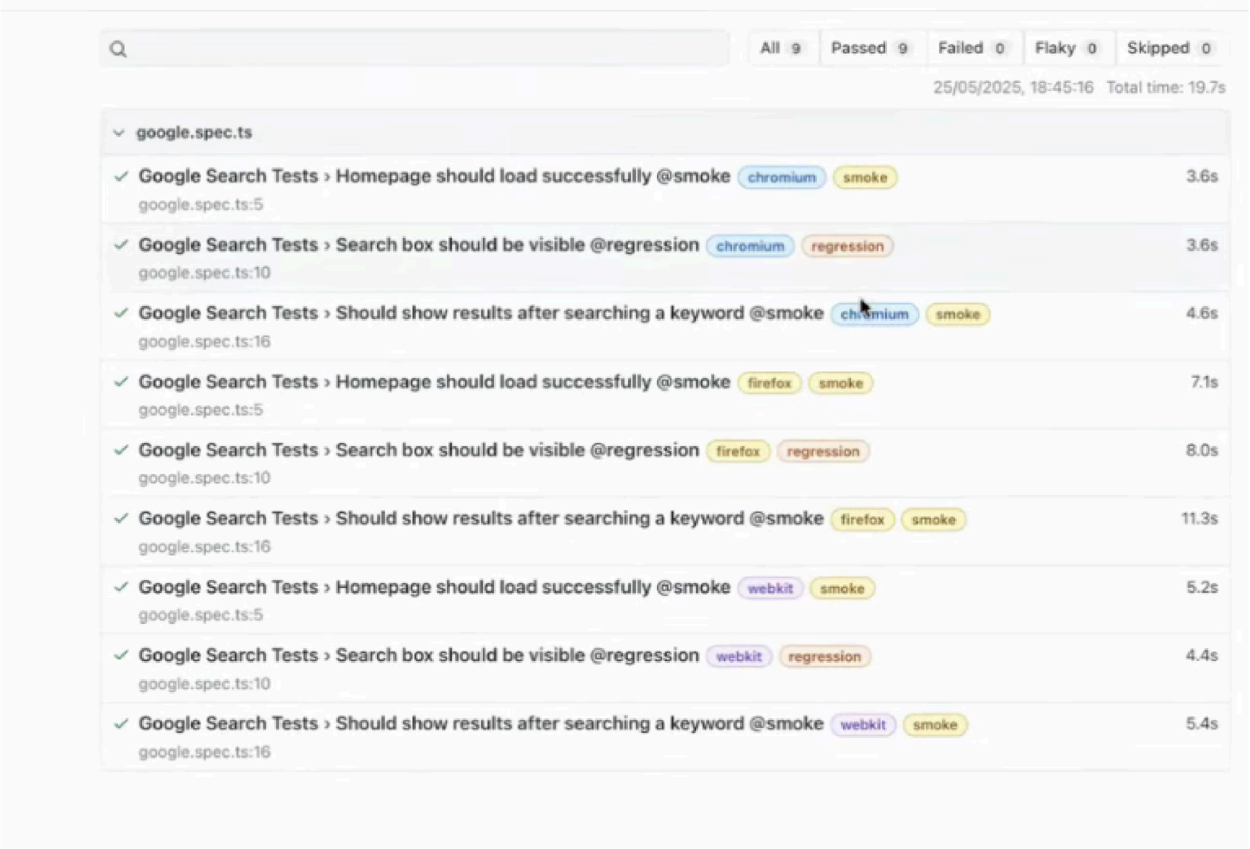
```
    trace: 'on-first-retry',
  },

  /* Configure projects for major browsers */
  projects: [
    {
      name: 'chromium',
      use: { ...devices['Desktop Chrome'] },
    },

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

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

```

Runs on all projects when not specified –



```
○ (base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$
○ (base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$ npx playwright test

  Running 12 tests using 6 workers

  █
```

Tags also come in output –



25/05/2025, 18:45:16 Total time: 19.7s

google.spec.ts

✓ Google Search Tests › Homepage should load successfully @smoke chromium smoke 3.6s
google.spec.ts:5

✓ Google Search Tests › Search box should be visible @regression chromium regression 3.6s
google.spec.ts:10

✓ Google Search Tests › Should show results after searching a keyword @smoke chromium smoke 4.6s
google.spec.ts:16

✓ Google Search Tests › Homepage should load successfully @smoke firefox smoke 7.1s
google.spec.ts:5

✓ Google Search Tests › Search box should be visible @regression firefox regression 8.0s
google.spec.ts:10

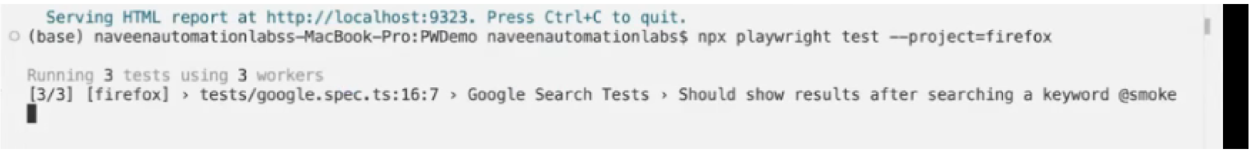
✓ Google Search Tests › Should show results after searching a keyword @smoke firefox smoke 11.3s
google.spec.ts:16

✓ Google Search Tests › Homepage should load successfully @smoke webkit smoke 5.2s
google.spec.ts:5

✓ Google Search Tests › Search box should be visible @regression webkit regression 4.4s
google.spec.ts:10

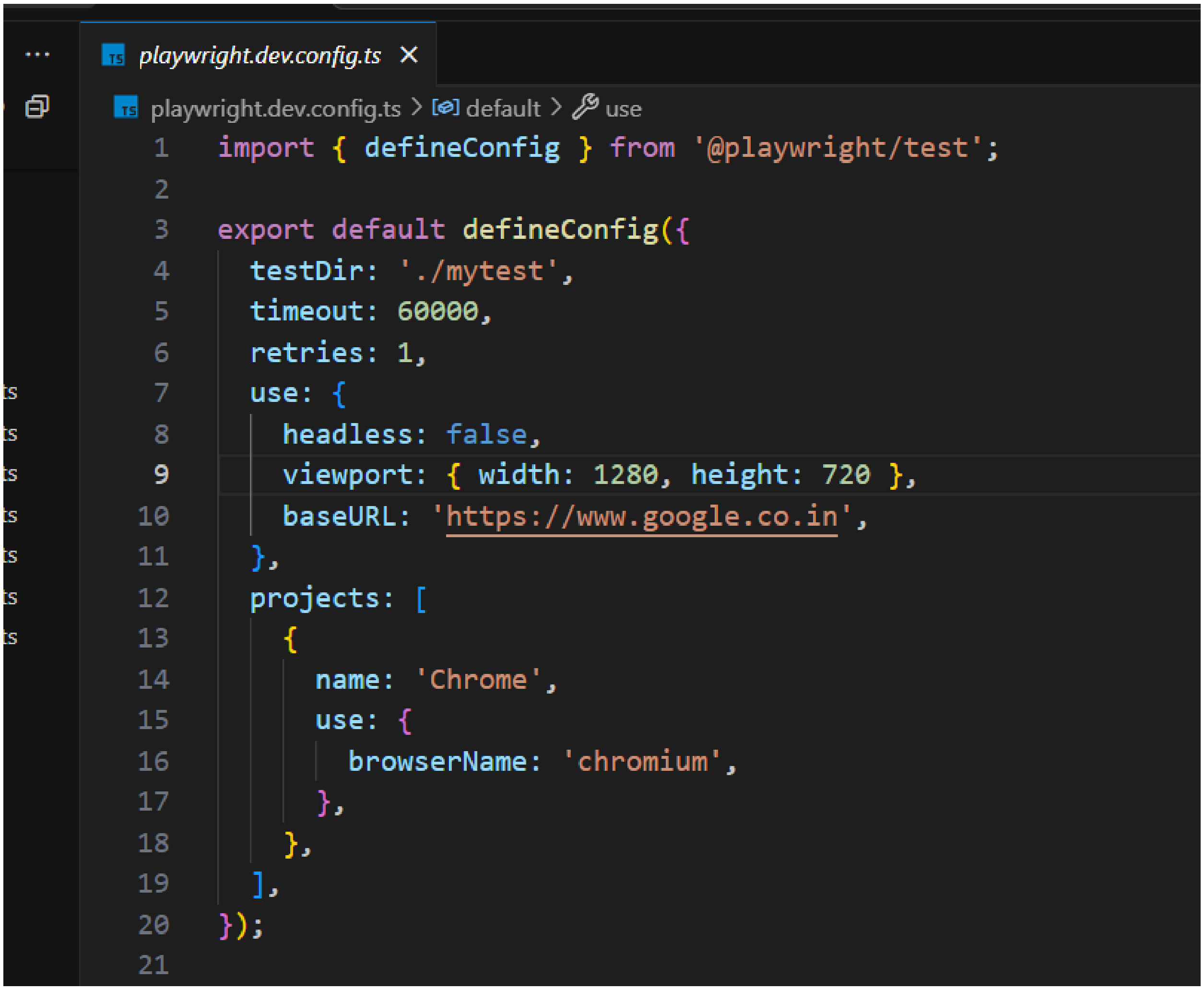
✓ Google Search Tests › Should show results after searching a keyword @smoke webkit smoke 5.4s
google.spec.ts:16

Run test on firefox–

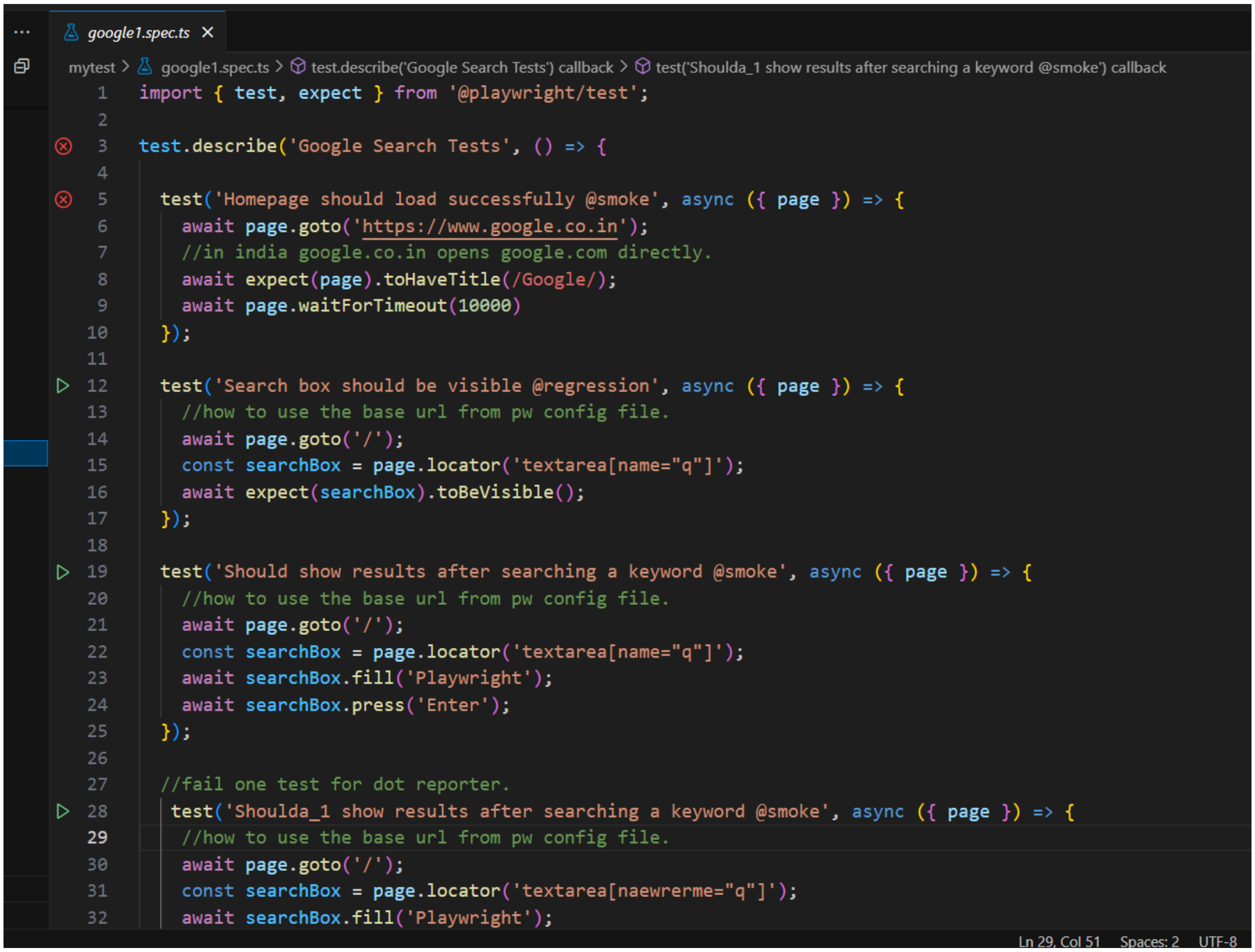


```
  Serving HTML report at http://localhost:9323. Press Ctrl+C to quit.
○ (base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$ npx playwright test --project=firefox

Running 3 tests using 3 workers
[3/3] [firefox] › tests/google.spec.ts:16:7 › Google Search Tests › Should show results after searching a keyword @smoke
█
```

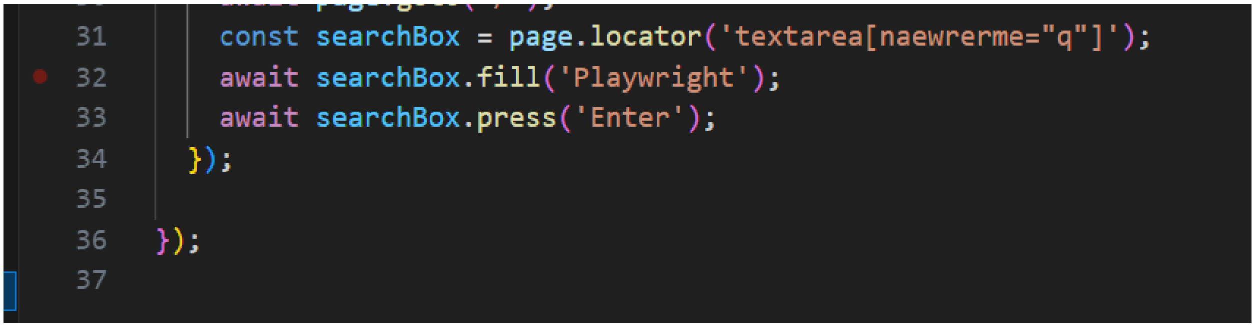
Create own config file –



```
import { defineConfig } from '@playwright/test';

export default defineConfig({
  testDir: './mytest',
  timeout: 60000,
  retries: 1,
  use: {
    headless: false,
    viewport: { width: 1280, height: 720 },
    baseURL: 'https://www.google.co.in',
  },
  projects: [
    {
      name: 'Chrome',
      use: {
        browserName: 'chromium',
      },
    },
  ],
});
```


google1.spec.ts
mytest > google1.spec.ts > test.describe('Google Search Tests') callback > test('Shoulda_1 show results after searching a keyword @smoke') callback
Ln 29, Col 51 Spaces: 2 UTF-8

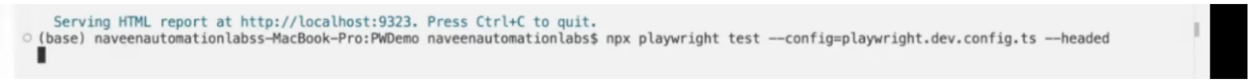
```
import { test, expect } from '@playwright/test';

test.describe('Google Search Tests', () => {

  test('Homepage should load successfully @smoke', async ({ page }) => {
    await page.goto('https://www.google.co.in');
    //in india google.co.in opens google.com directly.
    await expect(page).toHaveTitle(/Google/);
    await page.waitForTimeout(10000)
  });

  test('Search box should be visible @regression', async ({ page }) => {
    //how to use the base url from pw config file.
    await page.goto('/');
    const searchBox = page.locator('textarea[name="q"]');
    await expect(searchBox).toBeVisible();
  });

  test('Should show results after searching a keyword @smoke', async ({ page }) => {
    //how to use the base url from pw config file.
    await page.goto('/');
    const searchBox = page.locator('textarea[name="q"]');
    await searchBox.fill('Playwright');
    await searchBox.press('Enter');
  });

  //fail one test for dot reporter.
   test('Shoulda_1 show results after searching a keyword @smoke', async ({ page }) => {
    //how to use the base url from pw config file.
    await page.goto('/');
    const searchBox = page.locator('textarea[naewrerme="q"]');
    await searchBox.fill('Playwright');
```



```
    const searchBox = page.locator('textarea[naewrerme="q"]');
    await searchBox.fill('Playwright');
```

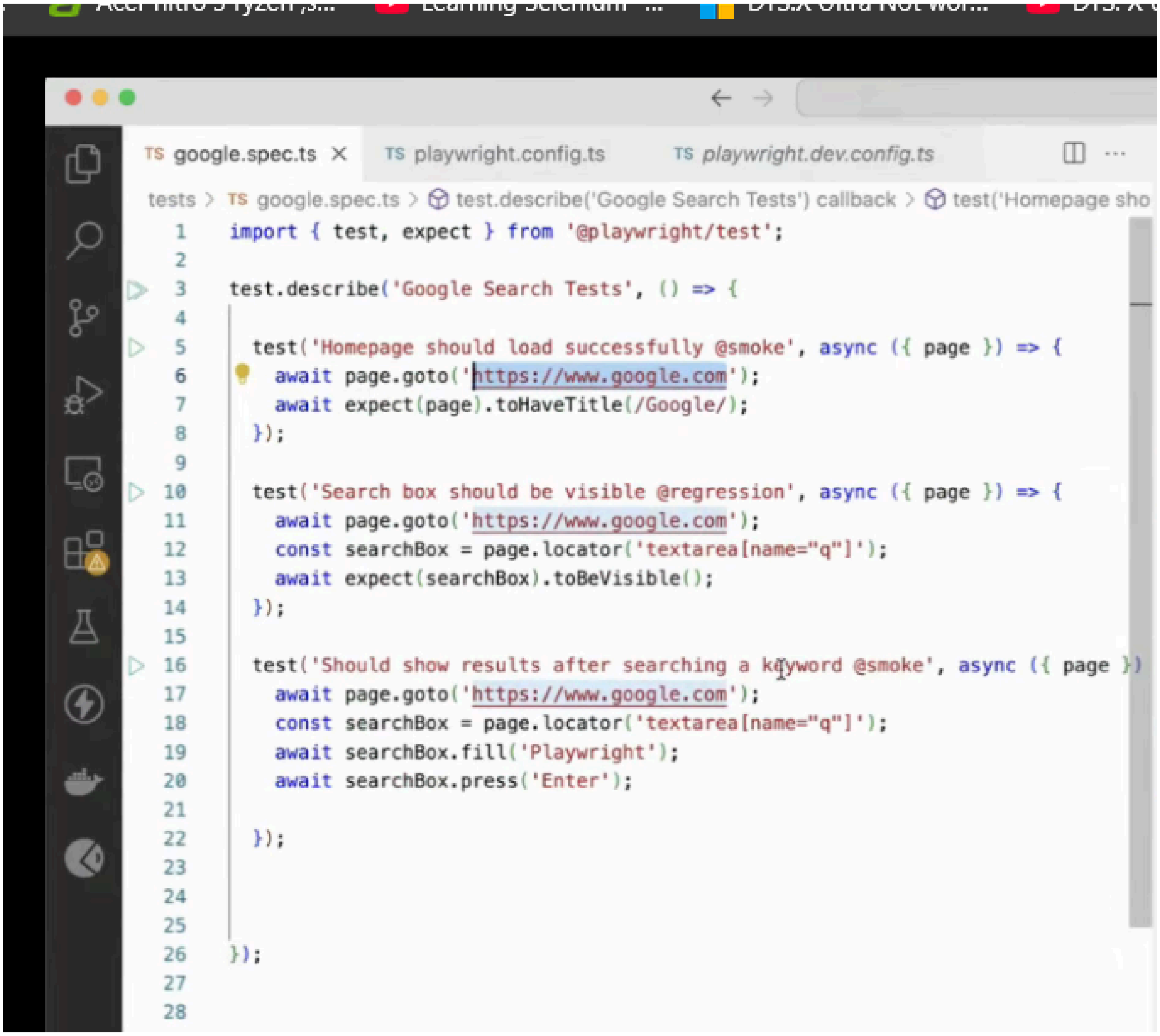
```
    await searchBox.press('Enter');
  });

});
```

Run the config file –
All files under the test directory folder which is mentioned in the pw config file will run.



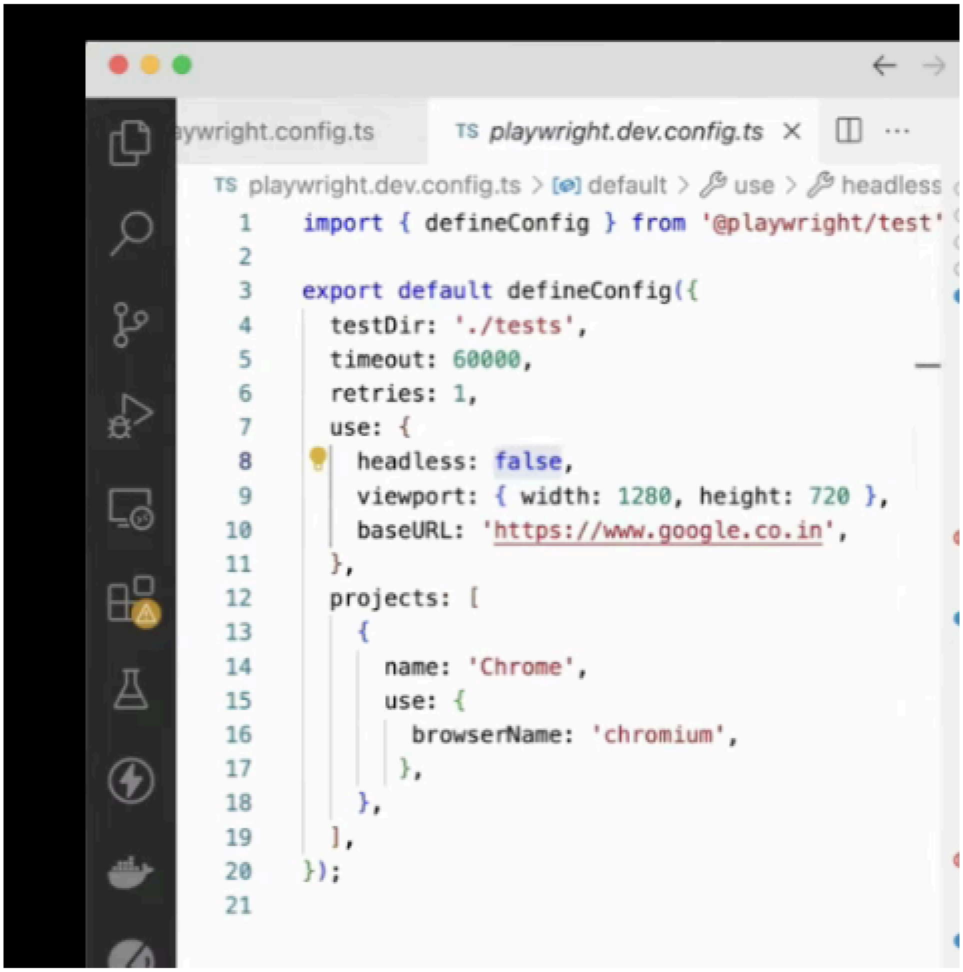
```
  Serving HTML report at http://localhost:9323. Press Ctrl+C to quit.
○ (base) naveenautomationlabss-MacBook-Pro:PWDemo naveenautomationlabs$ npx playwright test --config=playwright.dev.config.ts --headed
```

When script has a url -


google.spec.ts
playwright.config.ts
playwright.dev.config.ts
tests > google.spec.ts > test.describe('Google Search Tests') callback > test('Homepage sho

```
import { test, expect } from '@playwright/test';

test.describe('Google Search Tests', () => {

  test('Homepage should load successfully @smoke', async ({ page }) => {
    await page.goto('https://www.google.com');
    await expect(page).toHaveTitle(/Google/);
  });

  test('Search box should be visible @regression', async ({ page }) => {
    await page.goto('https://www.google.com');
    const searchBox = page.locator('textarea[name="q"]');
    await expect(searchBox).toBeVisible();
  });

  test('Should show results after searching a keyword @smoke', async ({ page })
    await page.goto('https://www.google.com');
    const searchBox = page.locator('textarea[name="q"]');
    await searchBox.fill('Playwright');
    await searchBox.press('Enter');

  });



});


```

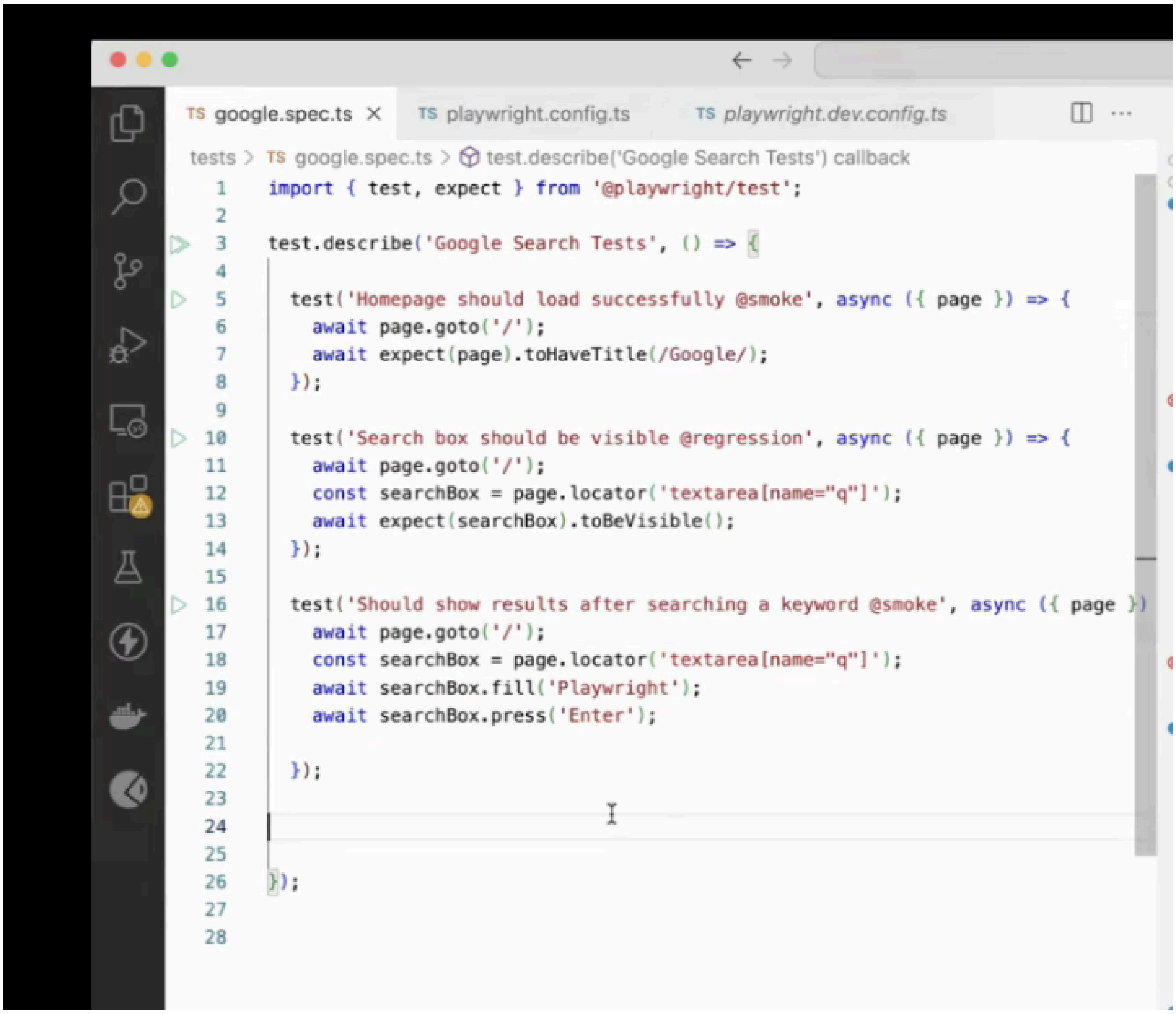
Config has another url -


ywright.config.ts
TS playwright.dev.config.ts
TS playwright.dev.config.ts > default > use > headless

```
import { defineConfig } from '@playwright/test'

export default defineConfig({
  testDir: './tests',
  timeout: 60000,
  retries: 1,
  use: {
    headless: false,
    viewport: { width: 1280, height: 720 },
    baseURL: 'https://www.google.co.in',
  },
  projects: [
    {
      name: 'Chrome',
      use: {
        browserName: 'chromium',
      },
    },
  ],
});
```

So preference given to the test file not config url.

How to use base url in test from config -


TS google.spec.ts
TS playwright.config.ts
TS playwright.dev.config.ts
tests > TS google.spec.ts > test.describe('Google Search Tests') callback

```
import { test, expect } from '@playwright/test';

test.describe('Google Search Tests', () => {

  test('Homepage should load successfully @smoke', async ({ page }) => {
    await page.goto('/');
    await expect(page).toHaveTitle(/Google/);
  });

  test('Search box should be visible @regression', async ({ page }) => {
    await page.goto('/');
    const searchBox = page.locator('textarea[name="q"]');
    await expect(searchBox).toBeVisible();
  });

  test('Should show results after searching a keyword @smoke', async ({ page })
    await page.goto('/');
    const searchBox = page.locator('textarea[name="q"]');
    await searchBox.fill('Playwright');
    await searchBox.press('Enter');

  });



});
```

Run the file and now base url will be correctly picked up.