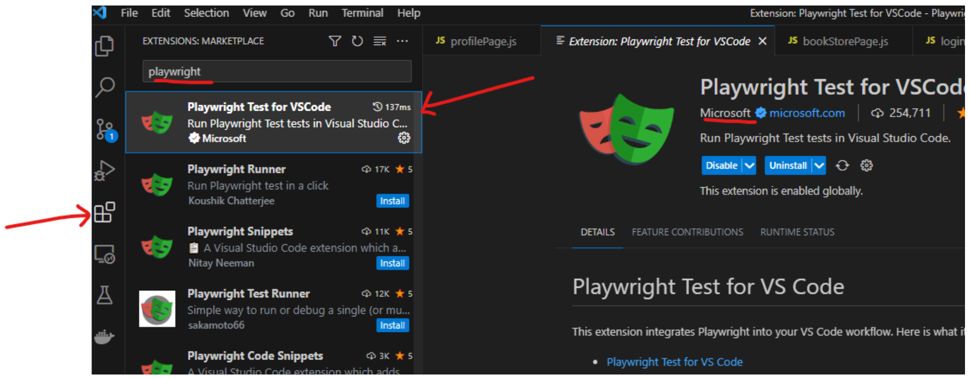
Installing Playwright with Node and Visual Studio Code:

1. **Install Node.js, *(if installed Skip to Step 2)*:**
   - Navigate to the Node.js website by clicking here.
   - Download the version appropriate for your operating system.
   - Run the installer and follow the installation instructions.
2. **Install Visual Studio Code (VS Code), *(if installed Skip to Step 3)*:**
   - Navigate to the Visual Studio Code website by clicking here.
   - Download the installer appropriate for your operating system.
   - Run the installer and follow the installation instructions.
3. **Open Visual Studio Code:**
   - After installation, open Visual Studio Code from your applications or desktop.
4. **Create/Open a New Project Folder in VS Code:**
   - Click on "File" in the menu bar.
   - Select "Open Folder" and choose or create a new folder for your project at your selected location.
5. **Install Playwright Extension:**
   - On VS Code, Click on Extensions icon on the left Pane(Activity Bar) or press "CTRL+Shift+X"
   - Search for "Playwright for VS Code" by Microsoft and hit Install


File Edit Selection View Go Run Terminal Help
Extension: Playwright Test for VSCode - Playwri
EXTENSIONS: MARKETPLACE
playwright
profilePage.js
Extension: Playwright Test for VSCode
bookStorePage.js
login
Playwright Test for VSCode
Run Playwright Test tests in Visual Studio C...
Microsoft
137ms
Playwright Runner
Run Playwright test in a click
Koushik Chatterjee
17K 5
Install
Playwright Snippets
A Visual Studio Code extension which a...
Nitay Neeman
11K 5
Install
Playwright Test Runner
Simple way to run or debug a single (or mu...
sakamoto66
12K 5
Install
Playwright Code Snippets
3K 5
A Visual Studio Code extension which adds
Playwright Test for VSCod
Microsoft microsoft.com 254,711
Run Playwright Test tests in Visual Studio Code.
Disable
Uninstall
This extension is enabled globally.
DETAILS
FEATURE CONTRIBUTIONS
RUNTIME STATUS
Playwright Test for VS Code
This extension integrates Playwright into your VS Code workflow. Here is what i
Playwright Test for VS Code

6. **Open Command Palette:**
   - In Visual Studio Code, click on "View" in the menu bar and select Command Palette
7. **Install Playwright files and Set Up the Project:**
   - In the Command Palette input field, type the following command and press Enter: Install Playwright
   - This will give you the option choose which browsers and language to install
   - Ensure to tick "***Chromium***" and "***Use JavaScript***" out of the available options
   - And Hit Ok to install
   - This command should install all the necessary Playwright files in the Project Folder that was created in Step 4.

With these steps completed, we are ready to copy the GitHub repository into the Project folder that we created.

Downloading GitHub Repository:

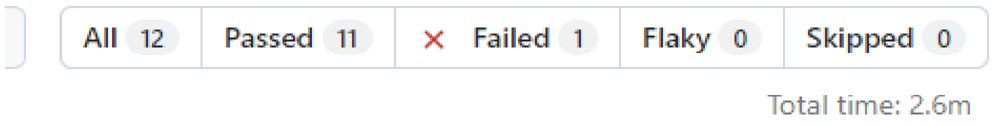
1. **Open the GitHub Repository:**
    - Navigate to GitHub: https://github.com/Sachin25sd/DEMO-QA-testing---Playwright.git
    - Search for the repository you want to download.
    - Download the Repository:
    - On the repository page, click on the green "Code" button.
    - Under HTTPS, select "Download ZIP" to download the repository as a ZIP file.
    - Extract the Repository from the Zip file.
    - Locate the downloaded ZIP file on your computer and extract its contents to a folder.
2. **Copying Downloaded Repository to the Project folder**
    - Open the Project Folder (where Playwright was installed) as well as the extracted repository folder created on Step 1.
    - Copy the folders **DemoQA – pages**, **DemoQA – TestData** and **tests** from the extracted repository to the Project Folder
    - Delete the default test case file from the ***tests*** folder named ***test-1.spec.js***

Running Tests and Generating Reports:

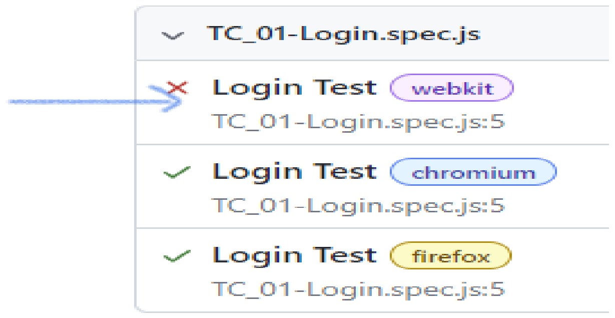
1. **Running Tests and generating reports**
    - Open New Terminal from the Terminal Menu bar
    - Finally, to run the report, please copy and paste the command in the terminal and press enter: **npx playwright test --headed --project=chromium**
    - Please use the below commands as required:
        - Headless mode (without showing the browser): ***npx playwright test***
        - Headed mode: ***npx playwright test --headed***
        - Generate Report: ***npx playwright show-report***
        - Run across all browsers: **npx playwright test**

Please use the Generate Report command if the report does not automatically display after execution.

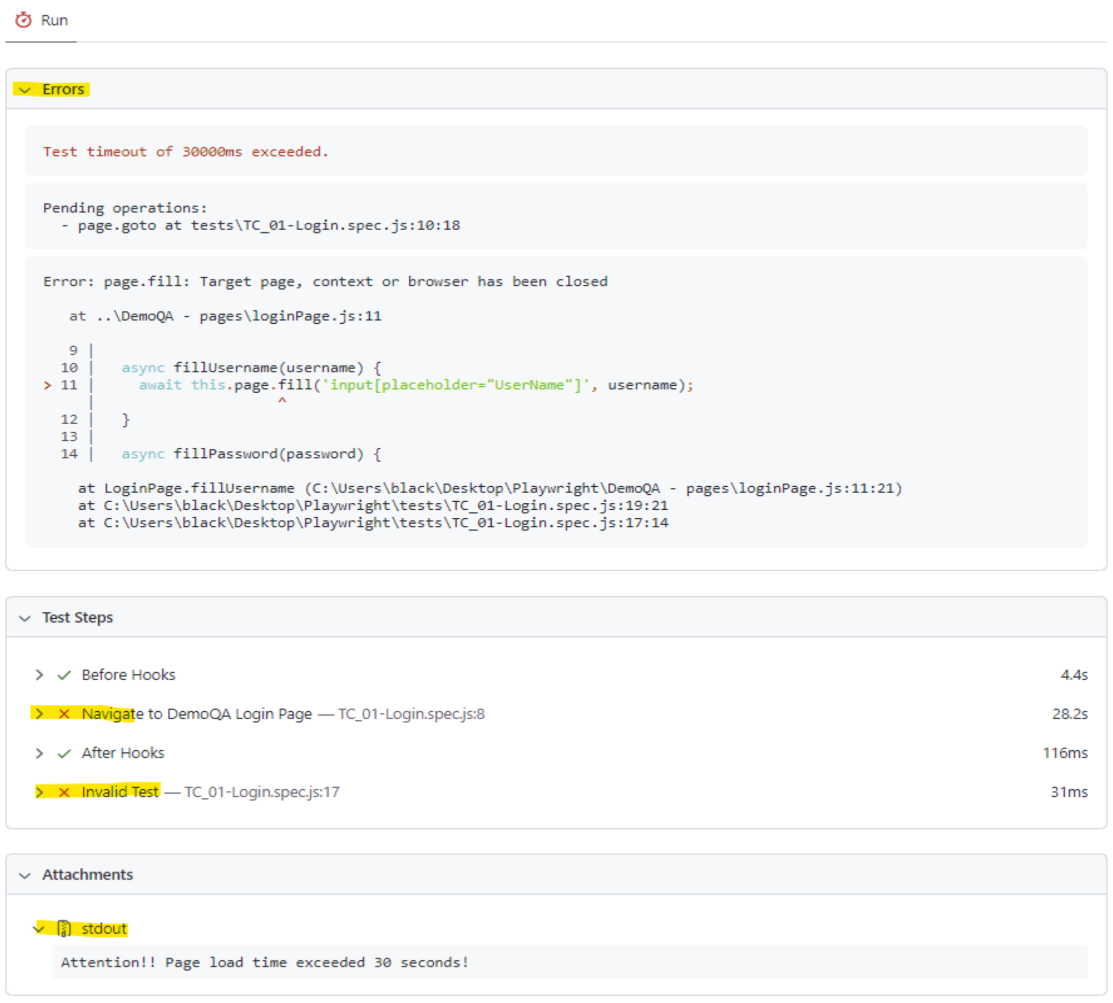
2. **Navigating through the report**
    - On the top right corner, you can find execution metrics and click on them for navigation



    - Each Test Case consists of multiple Test Steps and browsers they were executed on



- To navigate within the Test Steps, simply click on the Test Case Title (as marked on the above picture)
- The Test Case will show the following information:
    - **Error** that occurred during test run
    - **Test Steps** executed with Tick marks as Pass while Cross as Fail
    - And additionally, **Attachments** will capture Timeout errors or Alerts as highlighted in the below picture:



```
Test timeout of 30000ms exceeded.
```

```
Pending operations:
  - page.goto at tests\TC_01-Login.spec.js:10:18
```

```
Error: page.fill: Target page, context or browser has been closed

   at ..\DemoQA - pages\loginPage.js:11

   9 |
  10 |     async fillUsername(username) {
> 11 |       await this.page.fill('input[placeholder="UserName"]', username);
     |                      ^
  12 |     }
  13 |
  14 |     async fillPassword(password) {

    at LoginPage.fillUsername (C:\Users\black\Desktop\Playwright\DemoQA - pages\loginPage.js:11:21)
    at C:\Users\black\Desktop\Playwright\tests\TC_01-Login.spec.js:19:21
    at C:\Users\black\Desktop\Playwright\tests\TC_01-Login.spec.js:17:14
```

Test Steps
Before Hooks 4.4s
Navigate to DemoQA Login Page — TC_01-Login.spec.js:8 28.2s
After Hooks 116ms
Invalid Test — TC_01-Login.spec.js:17 31ms
Attachments
stdout

```
Attention!! Page load time exceeded 30 seconds!
```