Playwright Automation Framework - By Daljeet

A modern, production-ready Playwright test automation framework for web applications. This setup supports secure credential handling, page object model, Allure reporting, TypeScript, and GitHub SSH integration.

🛠️ Tools, Technologies & Languages Used

Category	Tools / Technologies
Language	TypeScript
Automation Tool	Playwright
Test Runner	@playwright/test
Reporting	Allure Reporter
Package Manager	npm
Code Editor	Visual Studio Code (VS Code)
Version Control	Git, GitHub (SSH based integration)
Environment Config	dotenv
OS Compatibility	macOS, Linux, Windows

📁 Folder Structure

```
Playwright_Framework_By_Daljeet/
├── tests/                         # All test cases
├── pages/                         # Page Object files
├── utils/                         # Utility functions (optional)
├── reports/                       # Allure & HTML reports (optional)
├── allure-results/               # Generated allure results
├── allure-report/                # Final Allure report output
├── .env                           # Secure credentials (ignored in git)
├── .gitignore
├── README.md
├── package.json
├── playwright.config.ts
```

⚙️ Prerequisites

Make sure the following tools are installed:

- Node.js v18+ or v20+ ✅

- Git ✅
- VS Code (Recommended IDE) ✅
- Playwright Test Library ✅
- Allure CLI ✅
- GitHub SSH Setup ✅

To install Allure CLI (globally):

```
brew install allure
```

📦 Installation & Setup

1. Clone the Repository via SSH:

```
git clone git@github.com:Diljeet-Ranaut/Playwright_Framework_By_Daljeet.git
cd Playwright_Framework_By_Daljeet
```

2. Install Dependencies

```
npm install
```

3. Install Browsers

```
npx playwright install
```

4. Create `.env` File for Credentials

```
# .env file
SAUCE_USERNAME=standard_user
SAUCE_PASSWORD=secret_sauce
```

✅ Running the Tests

Run all tests:

```
npx playwright test
```

Run specific test file:

```
npx playwright test tests/checkout.spec.ts
```

📄 Writing a Test

Example `tests/login.spec.ts`:

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

test('Login with valid credentials', async ({ page }) => {
  const login = new LoginPage(page);
  await login.navigate();
  await login.login(process.env.SAUCE_USERNAME!,
process.env.SAUCE_PASSWORD!);


  ✅ Assertions
  await expect(page).toHaveURL(/.*inventory/);
  await expect(page.locator('.inventory_list')).toBeVisible();
  await expect(page.locator('.app_logo')).toHaveText('Swag Labs');
});
```

📚 Page Object Example

Example `pages/LoginPage.ts`:

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

export class LoginPage {
  constructor(private page: Page) {}

  async navigate() {
    await this.page.goto('https://www.saucedemo.com');
  }

  async login(username: string, password: string) {
    await this.page.fill('[data-test="username"]', username);
    await this.page.fill('[data-test="password"]', password);
    await this.page.click('[data-test="login-button"]');
  }
}
```

📊 Allure Reporting

1. Generate Report

After running tests:

```
allure generate ./allure-results --clean -o ./allure-report
```

2. Open Report

```
allure open ./allure-report
```

Ensure `allure-playwright` is installed:

```
npm install --save-dev allure-playwright
```

And in `playwright.config.ts`:

```
reporter: [
  ['html', { open: 'never' }],
  ['allure-playwright']
]
```

🔒 Git SSH Setup (For Pushing Code)

```
git remote set-url origin git@github.com:Diljeet-
Ranaut/Playwright_Framework_By_Daljeet.git
git push -u origin main
```

✅ Additional Tips

- Use `.gitignore` to avoid committing sensitive or heavy folders:

```
node_modules/
.env
allure-results/
allure-report/
.vscode/
```

- Recommended plugins for VS Code:
  - Playwright Test for VS Code
  - dotenv extension
  - GitLens

🙆 Author

Daljeet Singh Ranot
Automation Test Lead at Alinma Bank | Playwright Automation Framework