

# **Playwright Cucumber BDD Framework with Page Object Model**

A comprehensive test automation framework using Playwright, Cucumber BDD, and Page Object Model (POM) pattern written in JavaScript.

## **Installation:**

### **Prerequisites:**

- Node.js (v16 or higher)
- npm or yarn

## **Setup Steps:**

1. Install dependencies: npm install
2. Install Playwright browsers: npx playwright install
3. Create environment file: cp .env.example .env

## **Configuration**

Environment Variables (.env):

BASE\_URL=https://sample-project-dusky.vercel.app

BROWSER=chromium

HEADLESS=true

TIMEOUT=60000

VIEWPORT\_WIDTH=1280

VIEWPORT\_HEIGHT=720

## **Running Tests**

Run all tests: npm test

Run tests in specific browser:

npm run test:chrome, firefox, webkit

Run tests in parallel: npm run test:parallel

Run tagged tests: npm run test:tags "@Section1"

Run headed mode: HEADLESS=false npx cucumber-js

Run specific section: npx cucumber-js --name "Section 1"

Generate HTML report: npm run report

Open HTML report: start reports\html-report\index.html

Open Cucumber report: start reports\cucumber-report.html

## **Best Practices:**

- Use Page Object Model
- Write descriptive scenarios
- Use tags
- Reusable steps
- Proper waits
- Screenshots on failure
- Use environment variables
- Dynamic locators
- Data tables
- CI/CD

## **CI/CD Integration:**

### **GitHub Actions**

The project includes a GitHub Actions workflow that automatically runs tests on Chrome when code is pushed to the repository.

**Workflow file:** `.github/workflows/npm-test-chrome.yml`

### **Triggers:**

- Push to `main` or `master` branches
- Pull requests to `main` or `master` branches
- Manual workflow dispatch

### **Workflow Steps:**

1. Set up Node.js 20
2. Install project dependencies
3. Install Chromium browser
4. Run `npm test` with Chrome configuration
5. Upload test reports as artifacts

### **Viewing Results:**

- Navigate to the **Actions** tab in your GitHub repository
- Select the latest workflow run
- Download artifacts to view the test reports

## **Reports**

Built-in Cucumber Reports (automatic when running tests)

- HTML Report
- JSON Report
- JUnit Report
- Screenshots
- Videos

