

# ASSIGNMENT: END-TO-END AUTOMATION USING PLAYWRIGHT POM

(SESSION 02)



## PLAYWRIGHT AUTOMATION

# ASSIGNMENT OVERVIEW

## Goal

- Automate the full 'Create New Claim' workflow on DummyClaim using Playwright with TypeScript.
- Apply Page Object Model (POM) to structure reusable and clean automation code.

## Steps to Automate

- Navigate to DummyClaim and click Login
- Login using email and password
- Verify 'Total Claim' on dashboard is \$1,240
- Click 'Create New Claim'
- Fill in Name, Amount, and Description
- Verify the entered values in the Review section
- Click Submit and assert alert message
- Accept alert and verify redirection to dashboard

## Important

- Use realistic dummy data for form. Structure project using POM with one class per page. Use async/await syntax and TypeScript typing

# EXPLORE THE APPLICATION VIA BROWSER

## Steps to Inspect UI

- Open <https://sharikmp.github.io/dummyclaim/> in browser
- Use DevTools to identify selectors for buttons, inputs, values
- Verify page transitions via URL inspection

## Page-wise Breakdown

- **\*\*HomePage:\*\*** Click Login button
- **\*\*LoginPage:\*\*** Fill Email & Password and click Login
- **\*\*DashboardPage:\*\*** Verify URL, locate Total Claim text, click Create New Claim
- **\*\*NewClaimPage:\*\*** Fill form, review values, click Submit, handle alert

# PROJECT FOLDER STRUCTURE

## Recommended Directory Layout

```
/pages
  base-page.ts
  homepage.ts
  login-page.ts
  dashboard-page.ts
  new-claim-page.ts
/tests
  dummy-claim-e2e.spec.ts
playwright.config.ts
tsconfig.json
```

# POM CLASS OUTLINES

## base-page.ts

```
async waitForLoad(): Promise  
async getAlertTextAndAccept(): Promise
```

## homepage.ts

```
async navigate(): Promise  
async clickLogin(): Promise
```

## login-page.ts

```
async login(email: string, password: string): Promise
```

## dashboard-page.ts

```
async verifyTotalClaim(expected: string): Promise  
async goToNewClaim(): Promise
```

## new-claim-page.ts

```
async fillForm(data: { name: string; amount: string; desc: string }): Promise  
async verifyReview(data: { name: string; amount: string; desc: string }): Promise  
async submit(): Promise
```