

## Candidate Information

- **Name:** Amit Raj
- **Technology:** Playwright (UI Automation)
- **Language:** TypeScript
- **Assignment:** UI Automation – Part 2

## Application Under Test

URL:

<https://claude.ai/public/artifacts/1e02a9a5-4f20-4f19-a7ba-6c3f16c6eab9>

## Assignment Objective

The objective of this assignment is to automate different UI scenarios such as:

- Delayed button enablement
- Lazy loaded list items
- Dynamic IDs handling
- Conditional UI rendering
- Nested modal handling

All tests are written without using hard waits and are stable across multiple runs.

## Tools & Technologies Used

- Playwright
- TypeScript
- Node.js
- HTML Reporter
- JUnit Reporter

## Project Structure

`playwright-ui-automation/`

```
|— tests/
|   |— delayed-button.spec.ts
|   |— lazy-list.spec.ts
|   |— dynamic-ids.spec.ts
|   |— conditional-render.spec.ts
|   └─ modal-flow.spec.ts
|— playwright.config.ts
|— package.json
└─ README.md
```

---

## Test Cases Overview

### Test 1: Delayed Button Flow

- Waits for a button to become enabled
- Clicks confirm action
- Verifies success message
- No hardcoded waits used

### Test 2: Load and Verify List Items

- Loads items using “Load More Items” button
  - Verifies total count is 15
  - Confirms presence of both “active” and “pending” status items
- 

### Test 3: Dynamic ID Handling

- Handles dynamically changing IDs
- Uses text-based selectors instead of ID attributes
- Selects and verifies “Beta” item

### Test 4: Conditional Login Flow

- Logs in as Admin user and verifies Admin Panel
- Logs out and logs in as Standard user
- Verifies correct panel visibility based on user type

## **Test 5: Modal Confirmation Flow**

- Handles modal and nested modal
- Ensures correct modal layer interaction
- Verifies confirmation result after modal close