Day 5: Assertions & Validations - Playwright Test Assertions (expect)
===

1. Visibility Checks

```
await expect(page.locator('text=Login')).toBeVisible();
await expect(page.locator('#logout')).toBeHidden();
```

2. URL Checks

```
await expect(page).toHaveURL(/.*dashboard/);
```

3. Title Checks

```
await expect(page).toHaveTitle("My App - Dashboard");
```

4. Text Checks

```
await expect(page.locator('.welcome')).toHaveText("Welcome Vignesh!");
await expect(page.locator('.error')).toContainText("Invalid credentials");
```

Practice: Validate Successful Login

```
test('should login successfully', async ({ page }) => {
await page.goto('https://example.com/login');

await page.fill('#username', 'testuser');
await page.fill('#password', 'Password123');
await page.click('button[type="submit"]');

// Assertions
await expect(page).toHaveURL(/.*dashboard/);
await expect(page.locator('h1')).toHaveText("Welcome testuser");
await expect(page.locator('#logout')).toBeVisible();
});
```

Day 6: Handling Waits & Timing in Playwright
===

Auto-Waiting in Playwright

- Playwright automatically waits for elements to appear, become visible, and be enabled before performing actions.

Example:

```
await page.click('button#submit');
```

Explicit Waiting
----

1. Wait for Selector

```
await page.waitForSelector('#dashboard');
```

2. Locator Wait

```
const dashboard = page.locator('#dashboard');
await dashboard.waitFor();
```

Waiting for Navigation

```
await Promise.all([
page.waitForNavigation(),
page.click('text=Login')
]);
```

Waiting for Network Idle

```
await page.waitForLoadState('networkidle');
```

Avoiding Flaky Tests

- Do not use page.waitForTimeout(5000) (bad practice)
- Use auto-waits, expect conditions, waitForSelector, waitForLoadState

Stable Login Flow Example

```
test('stable login flow', async ({ page }) => {
await page.goto('https://example.com/login');

await page.fill('#username', 'testuser');
await page.fill('#password', 'Password123');
await Promise.all([
page.waitForNavigation({ waitUntil: 'networkidle' }),
page.click('button[type="submit"]')
]);

await expect(page).toHaveURL(/.*dashboard/);
});
```