

1. ให้สรุปสิ่งที่ได้เรียนรู้จากการฟัง Guest speaker บรรยายเรื่อง Automated Testing ไม่เกิน 1 หน้า

จากการฟังวิทยากรบรรยายเรื่อง Automated Testing ผมได้เรียนรู้หลายประเด็นสำคัญที่เป็นประโยชน์ต่อการพัฒนาซอฟต์แวร์อย่างมาก

เริ่มต้นจากการทำความเข้าใจว่า Automated Testing คืออะไร และมีขั้นตอนอย่างไร ซึ่งเป็นการใช้เครื่องมือช่วยในการทดสอบซอฟต์แวร์แบบอัตโนมัติ ช่วยลดเวลาการทำงาน และเพิ่มความแม่นยำในการตรวจสอบระบบ

ได้ทราบถึง ข้อดีของ Automated Testing เช่น การประหยัดเวลาในการทดสอบซ้ำ ๆ เพิ่มความถูกต้อง ลดข้อผิดพลาดจากมนุษย์ และสามารถใช้ร่วมกับกระบวนการ CI/CD ได้อย่างมีประสิทธิภาพ

ในด้านของผู้ที่เหมาะสมกับการใช้ Automated Testing ได้แก่

- QA Engineers & Testers
- Software Engineers & Developers
- Product Owners & Project Managers
- DevOps & Site Reliability Engineers
- Software Architects & System Analysts

นอกจากนี้ยังได้เข้าใจความแตกต่างระหว่าง Manual Testing และ Automated Testing โดย Manual Testing ใช้คนในการทดสอบ ซึ่งเหมาะกับการทดสอบเชิงประสบการณ์หรือ UI ส่วน Automated Testing เหมาะกับการทดสอบซ้ำ ๆ หรือระบบที่มีความซับซ้อน

อย่างไรก็ตาม ก็ได้เรียนรู้ว่า Automated Testing ก็มีข้อเสีย เช่น ต้องใช้เวลาสร้างเทสในครั้งแรก ใช้ทรัพยากรสูง และไม่สามารถทดสอบทุกกรณีได้ โดยอัตโนมัติ เช่น UI หรือ UX

ได้รู้จักเครื่องมือยอดนิยม เช่น

- Cucumber
- Playwright
- Cypress
- Selenium

รวมถึงได้เข้าใจประเภทของ Testing Framework ที่ใช้ร่วมกับ Automated Testing ได้แก่

- Linear Testing Framework
- Data-Driven Framework
- Keyword-Driven Framework
- Modular-Based Testing Framework
- Hybrid Framework

โดยรวม การบรรยายในครั้งนี้ได้เปิดโลกทัศน์ใหม่และเสริมความรู้ในการเลือกใช้และพัฒนา Automated Testing อย่างมีประสิทธิภาพ เหมาะสำหรับผู้ที่เกี่ยวข้องกับกระบวนการพัฒนาซอฟต์แวร์ในทุกตำแหน่ง ซึ่งผมสามารถเอาไปใช้ในการทำงานโปรเจกต์ SE 2 รวมถึงงานในอนาคตได้อีกด้วย

2. Capture หน้าจอแสดงโค้ดส่วนที่ได้ลองแก้ไขพร้อมคำอธิบายว่าทำอะไรไปบ้าง และผลลัพธ์จากการ run test เช่น terminal ที่แสดงผลลัพธ์

File 1: inventory_modifying.steps.ts

เพิ่มสินค้าจำนวน itemCount ขึ้นแรกลงตะกร้าจากหน้า inventory โดยคลิกปุ่ม "Add to cart" ของแต่ละรายการ

```
// เพิ่มสินค้าเข้าไปในตะกร้าตามจำนวนที่กำหนด
When(
  "I add the first {int} items to the cart",
  async function (itemCount: number) {
    const page = customWorld.page;
    if (!page) throw new Error("Page is not available");

    const allItems = page.locator(".inventory_item");
    const total = await allItems.count();

    expect(total).toBeGreaterThanOrEqual(itemCount);

    for (let index = 0; index < itemCount; index++) {
      const product = allItems.nth(index);
      const addToCartBtn = product
        .locator("button")
        .filter({ hasText: "Add to cart" });
      await addToCartBtn.click();
    }
  }
);
```

ลบสินค้าจำนวน removeCount รายการโดยคลิกปุ่ม "Remove" ของรายการที่แสดงอยู่บนสุดที่ละรายการ

```
When(
  "I remove {int} items from the cart, starting with the first item displayed",
  async function (removeCount: number) {
    const page = customWorld.page;
    if (!page) throw new Error("Page is not available");

    await page.click(".shopping_cart_link");

    const cartItems = page.locator(".cart_item");
    const total = await cartItems.count();

    expect(total).toBeGreaterThanOrEqual(removeCount);

    for (let i = 0; i < removeCount; i++) {
      await cartItems.first().locator("button", { hasText: "Remove" }).click();
    }
  }
);
```

ตรวจสอบว่าตัวแสดงจำนวนสินค้าในตะกร้า (cart badge) แสดงจำนวนสินค้าตรงกับค่าที่คาดไว้ (expected) หรือไม่ หากเป็น 0 ต้องไม่มี badge ปรากฏเลย

```
Then("I should see {int} items in the cart", async function (expected: number) {
  const badge = customWorld.page?.locator('[data-test="shopping-cart-badge"]');
  if (!badge) throw new Error("Badge locator is undefined");

  if (expected === 0) {
    await expect(badge).toHaveCount(0);
  } else {
    await expect(badge).toHaveText(expected.toString());
  }
});
```

เชื่อว่าปุ่มที่มีข้อความ {label} ยังคงแสดงอยู่สำหรับสินค้าที่เหลืออยู่

```
Then(
  "I should see a {string} button for the remaining items",
  async function (label: string) {
    const page = customWorld.page;
    if (!page) throw new Error("Page is not available");

    const buttons = page.locator("button").filter({ hasText: label });
    const count = await buttons.count();
    expect(count).toBeGreaterThan(0);
  }
);
```

File 2: inventory_sorting.steps.ts

เลือกตัวเลือก "za" จาก dropdown แล้วตรวจสอบว่าแสดง error dialog ที่มีข้อความตรงกับ expectedMessage และกดยอมรับ alert นั้น

```

Then(
  "I should see an error dialog with error message {string}",
  async function (expectedMessage: string) {
    const page = customWorld.page;
    if (!page) throw new Error("Page not found");

    // Setup dialog listener
    const dialogPromise = new Promise<void>((resolve, reject) => {
      page.once("dialog", async (dialog) => {
        try {
          expect(dialog.message()).toMatch(expectedMessage);
          await dialog.accept();
          resolve();
        } catch (error) {
          reject(error);
        }
      });
    });

    // Trigger dropdown selection (to possibly raise dialog)
    const dropdown = page.locator('[data-test="product-sort-container"]');
    await expect(dropdown).toBeVisible();

    const zaOption = dropdown.locator('option[value="za"]');
    const hasOption = await zaOption.count();
    expect(hasOption).toBeGreaterThan(0);

    await dropdown.selectOption("za");

    // Wait for dialog to be handled
    await dialogPromise;
  }
);

```

Output

จาก Terminal จะเห็นว่า run ผ่านทุก case

```

• (base) film@Waranthon-MacBook-Air cucumber-playwright-workshop % npm run cucumber-ts
> cucumber-playwright-workshop@1.0.0 cucumber-ts
> cucumber-js test

5/20 steps [=====] Finished scenario: Validate the addition and removal of items in the cart
7/20 steps [=====] Finished scenario: Successfully sorting inventory products
10/20 steps [=====] Finished scenario: Failed to sort the inventory products
14/20 steps [=====] Finished scenario: Login with standard user
18/20 steps [=====] Finished scenario: Login with locked out user
=====
5 scenarios (5 passed)
20 steps (20 passed)
0m18.508s (executing steps: 0m12.219s)
❖ (base) film@Waranthon-MacBook-Air cucumber-playwright-workshop %

```