

# FUNCTIONAL AND PERFORMANCE TESTING

DATE	
TEAM ID	LTVIP2025TMID30810
PROJECT NAME	Asset Management Portal

## MILESTONE 5: TESTING

### Activity 1: testing UI action

#### PURPOSE:

To ensure that the UI action (button or link) works correctly and performs the intended function without errors.

#### USE:

Testing UI actions like "Assign Asset" or "Return Asset" ensures the system responds correctly to user inputs, improves reliability, and provides a smooth user experience.

#### STEPS:

1. Go to Asset Inventory table
2. Click on New
3. Fill in the details
  - a) Asset name: Laptop
  - b) Type: laptop
  - c) Assigned to: Abel Tutor
  - d) Status: Available
  - e) select some purchase and expiry date
4. Click on submit
5. Open the record again
6. Click on mark as lost button and save
7. Check the status is changed to lost.

## MILESTONE 5: TESTING

### Activity 2: testing scheduled job

#### PURPOSE:

To verify that the scheduled job runs automatically at the set time and performs its task correctly.

#### USE:

Testing ensures that automated tasks like sending maintenance alerts or generating reports run as expected, helping maintain system accuracy and reducing manual effort.

#### STEPS:

1. Navigate to background scripts
2. Write the Scheduled job script in the background scripts
3. Click on Run Script
4. Check the result

The screenshot displays the ServiceNow interface. The top section shows an asset record for 'asset inventory - IST001101'. The record details include: Number (IST001101), Purchase date (2025-06-22), Status (Lost), Assigned to (Abel Tutor), Asset name (Laptop), Type (laptop), and Warranty Expire (2025-07-30). Below the record details are buttons for 'Update', 'Mark As Damaged', 'Mark As Repaired', and 'Delete'.

The bottom section shows a background script editor. The script is written in GlideScript and is designed to send a warranty expiry alert email. The script includes the following code:

```
1 var grAsset = new GlideRecord('u_asset_inventory'); // Replace with your table name
2
3 var today = new GlideDateTime();
4
5 var futureDate = new GlideDateTime();
6
7 futureDate.addDays(30); // Get date 30 days from now
8
9 grAsset.addQuery('u_warranty_expiry', '<=', futureDate); // Warranty expiring within the next 30 days
10
11 grAsset.addQuery('u_warranty_expiry', '>=', today); // Warranty expiring after today
12
13 grAsset.query();
14
15 while (grAsset.next()) {
16
17     var email = new GlideEmailOutbound();
18
19     email.setSubject("Warranty Expiry Alert: " + grAsset.getValue('u_assest_name')); // Use getValue for dynamic field access
20
21     email.setBody("The warranty for " + grAsset.getValue('u_assest_name') + " (Type: " + grAsset.getValue('u_asset_type') +
22         " | | ") is expiring soon on " + grAsset.getValue('u_warranty_expiry') + ". Please take action."); // Get values dynamically
23
24     email.setTo('it-support@company.com'); // Change to your IT support email
25
26     email.send();
27
28
29
30
31     gs.info("Email sent for assest: " + grAsset.getValue('u_assest_name')); // Log for confirmation
32
33 }
```

At the bottom of the script editor, there are options to 'Run Script' and 'Instance Scripts'. The 'Run Script' button is highlighted.

