PROJECT DESIGN

TEAM ID	LTVIP2025TMID30733
PROJECT NAME	Asset Management Portal

Proposed solution:

Project team shall fill the following information in the proposed solution template

S. No	Parameter	Description
1	Problemstatement (problem to be solved)	Organizations often face challenges in tracking, managing, and maintaining their physical and digital assets, leading to asset loss, inefficiency, and inaccurate records. The lack of a centralized system results in poor visibility, delayed maintenance, and difficulty in asset allocation.
2	Idea / Solution description	The Asset Management Portal is a robust and centralized web application designed to optimize the management of both physical and digital assets within an organization. It enables automated asset tracking, real-time inventory updates, user-friendly self-service features, and smart alerts for maintenance and warranty. By reducing human error and manual workload, the portal enhances operational efficiency, promotes responsible asset usage, and ensures timely decision-making through insightful reporting dashboards.
3	Novelty/Uniqueness	The Asset Management Portal stands out with its automation of the entire asset lifecycle, including real-time tracking, self-service asset requests, and intelligent maintenance alerts.
4	Social Impact/Customer satisfaction	The Asset Management Portal improves organizational transparency and accountability, reducing asset misuse and promoting responsible resource utilization
5	Business model (Revenue Model)	The Asset Management Portal follows a Software-as-a-Service (SaaS) business model
6	Scalability of the Solution	The Asset Management Portal is highly scalable, capable of handling increasing numbers of users, assets, and organizational

		data without compromising performance.

Asset Management Portal

MILESTONE 1: TABLE

Activity 1: create table

PURPOSE:

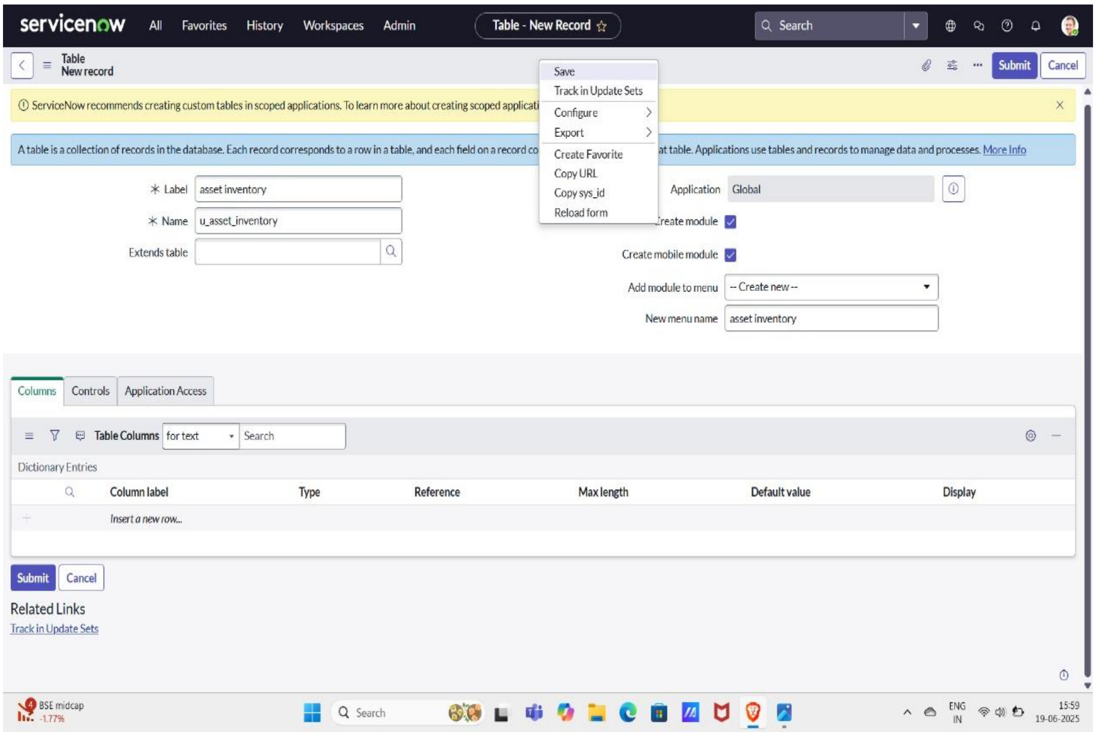
The purpose of creating a table in the Asset Management Portal is to store and organize asset-related data in a structured format. This table serves as the backbone of the portal's database, where each asset record (like ID, name, type, owner, status, location, etc.) is stored in rows and columns for easy access, tracking, and reporting.

USE:

Central Data Storage: The table stores all asset information in a centralized and searchable format.

STEPS:

1. Open service now.
2. Click on All >> search for tables
3. Open System definition >> tables
4. Click on new
5. Fill in the details as
    a. Name: asset inventory
6. Save the table



ⓘ ServiceNow recommends creating custom tables in scoped applications. To learn more about creating scoped applicati

A table is a collection of records in the database. Each record corresponds to a row in a table, and each field on a record co at table. Applications use tables and records to manage data and processes. More Info

✱ Label	asset inventory
✱ Name	u_asset_inventory
Extends table	
Application	Global
Create module	☑
Create mobile module	☑
Add module to menu	-- Create new --
New menu name	asset inventory

Column label	Type	Reference	Max length	Default value	Display
Insert a new row...					

MILESTONE 1: TABLE

Activity 2: create fields

PURPOSE:

To define specific data points (like asset name, type, status) that will be stored in each record of the table.

USE:

Fields allow the portal to capture detailed asset information (e.g., serial number, owner, location) and enable accurate tracking, searching, filtering, and reporting of assets. They ensure that each asset entry is complete and consistent.

STEPS:

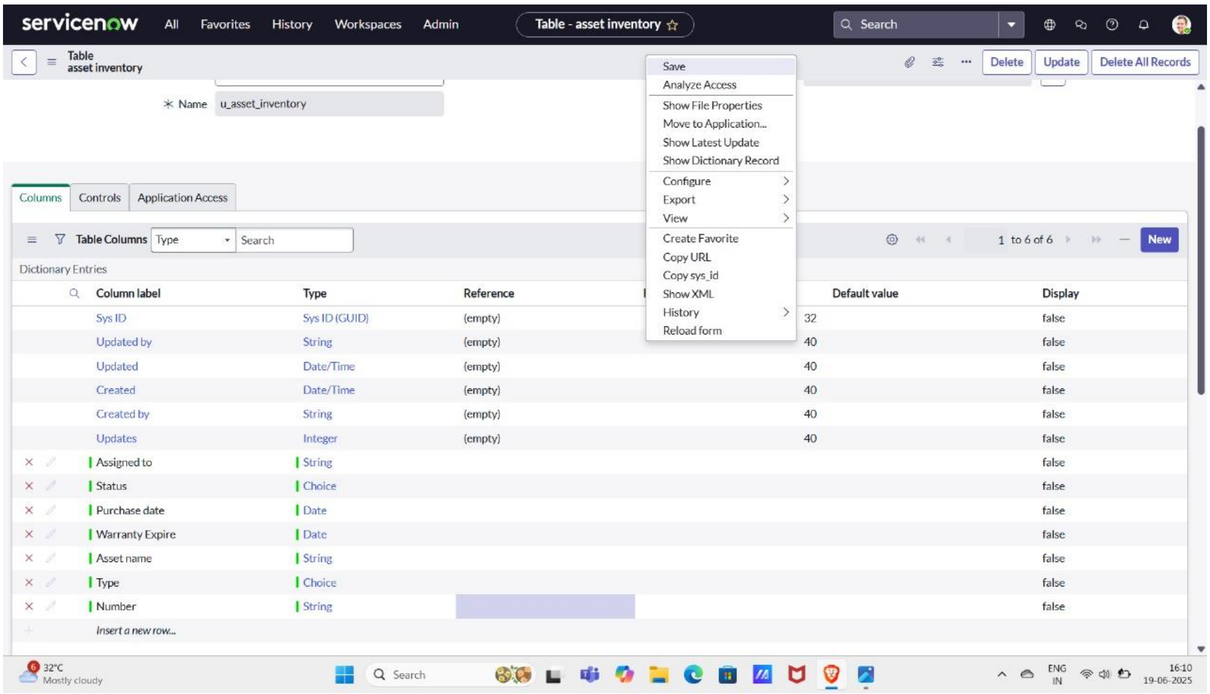
1)After saving the table scroll down

2)Create fields

- Assigned to: string
- Status: choice
- Purchase date: date

- Warranty Expire: date
- Asset name: string
- Type: choice
- Number: String

3) Click on save



Name: u_asset_inventory

Column label	Type	Reference		Default value	Display
Sys ID	Sys ID (GUID)	(empty)	32		false
Updated by	String	(empty)	40		false
Updated	Date/Time	(empty)	40		false
Created	Date/Time	(empty)	40		false
Created by	String	(empty)	40		false
Updates	Integer	(empty)	40		false
Assigned to	String				false
Status	Choice				false
Purchase date	Date				false
Warranty Expire	Date				false
Asset name	String				false
Type	Choice				false
Number	String				false
Insert a new row...					

MILESTONE 2: UI ACTION

Activity 1: create UI action 1

PURPOSE:

To add a custom button or link on a form or list that performs a specific action when clicked.

USE:

UI Actions improve user interaction by allowing quick actions like "Assign Asset," "Return Asset," or "Request Approval" directly from the portal interface. This enhances usability and speeds up common tasks.

STEPS:

1. Navigate to System Definition >> UI action

2. Click on New

3. Fill in the details;

Name: Mark As Lost

Table: Asset Inventory

Action name: mark_as_lost

Condition: current.u_status != 'Lost'

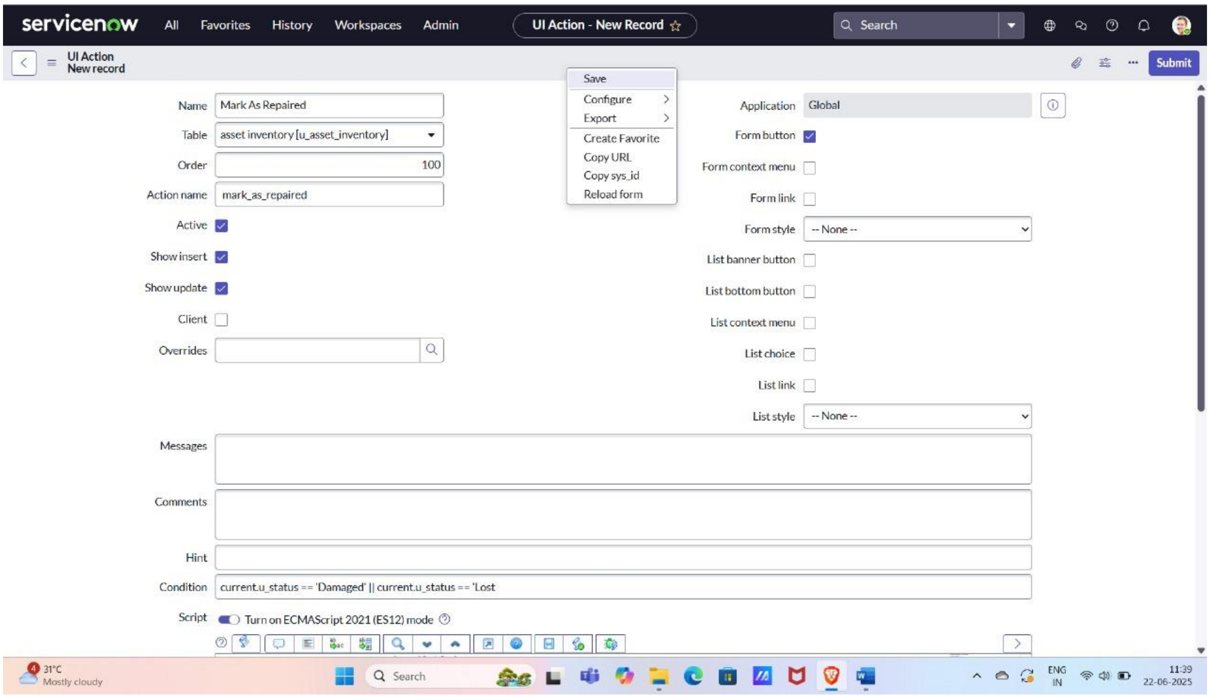
Script:

current.u_status = 'Lost';

current.update();
action.setRedirectURL(current);

4. Check the form button box

5. Click on save



Name: Mark As Repaired

Table: asset inventory [u_asset_inventory]

Order: 100

Action name: mark_as_repaired

Active ☑

Show insert ☑

Show update ☑

Client □

Overrides:

Application: Global

Form button ☑

Form context menu □

Form link □

Form style: -- None --

List banner button □

List bottom button □

List context menu □

List choice □

List link □

List style: -- None --

Messages:

Comments:

Hint:

Condition: current.u_status == 'Damaged' || current.u_status == 'Lost'

Script: Turn on ECMAScript 2021 (ES12) mode

MILESTONE 2: UI ACTION

Activity 2: create UI action 2

STEPS:

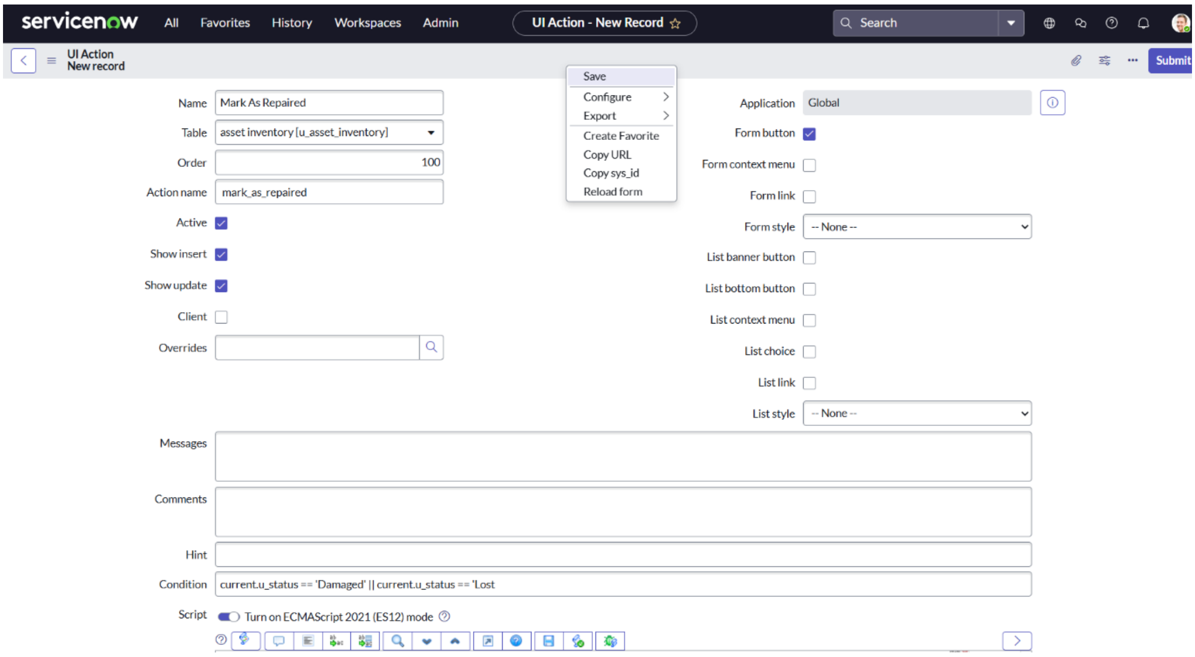
1. Navigate to System Definition >> UI action
2. Click on New
3. Fill in the details;
    1. Name: Mark As Repaired
    2. Table: Asset Inventory
    3. Action name: mark_as_repaired
    4. Condition: current.u_status == 'Damaged' || current.u_status == 'Lost’
    5. Script:

current.u_status = 'Available';

current.update();

action.setRedirectURL(current);

4. Check the form button box
5. Click on save



Name: Mark As Repaired
Table: asset inventory [u_asset_inventory]
Order: 100
Action name: mark_as_repaired
Active ☑
Show insert ☑
Show update ☑
Client □
Overrides:
Application: Global
Form button ☑
Form context menu □
Form link □
Form style: -- None --
List banner button □
List bottom button □
List context menu □
List choice □
List link □
List style: -- None --
Messages:
Comments:
Hint:
Condition: current.u_status == 'Damaged' || current.u_status == 'Lost
Script: Turn on ECMAScript 2021 (ES12) mode

MILESTONE 2: UI ACTION

Activity 3: create UI action 3

STEPS:

1. Navigate to System Definition >> UI action

2. Click on New

3. Fill in the details;

Name: Mark As Dameged

Table: Asset Inventory

Action name: mark_as_damaged

Condition: current.u_status != 'Damaged'

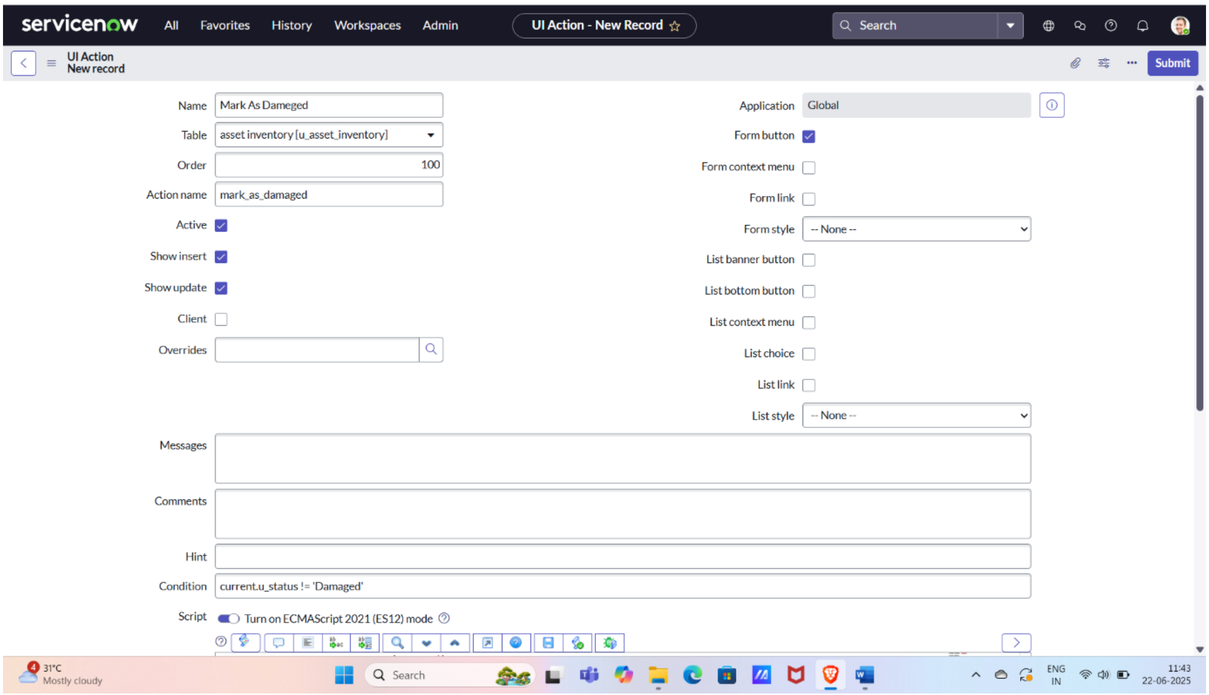
Script:

current.u_status = 'Damaged';

current.update();
action.setRedirectURL(current);

4. Check the form button box

5. Click on save



Name: Mark As Dameged
Table: asset inventory [u_asset_inventory]
Order: 100
Action name: mark_as_damaged
Active ☑
Show insert ☑
Show update ☑
Client □
Overrides:
Messages:
Comments:
Hint:
Condition: current.u_status != 'Damaged'
Script: Turn on ECMAScript 2021 (ES12) mode

Application: Global
Form button ☑
Form context menu □
Form link □
Form style: -- None --
List banner button □
List bottom button □
List context menu □
List choice □
List link □
List style: -- None --

MILESTONE 3: SCHEDULED JOB

Activity 1: create scheduled job

PURPOSE:

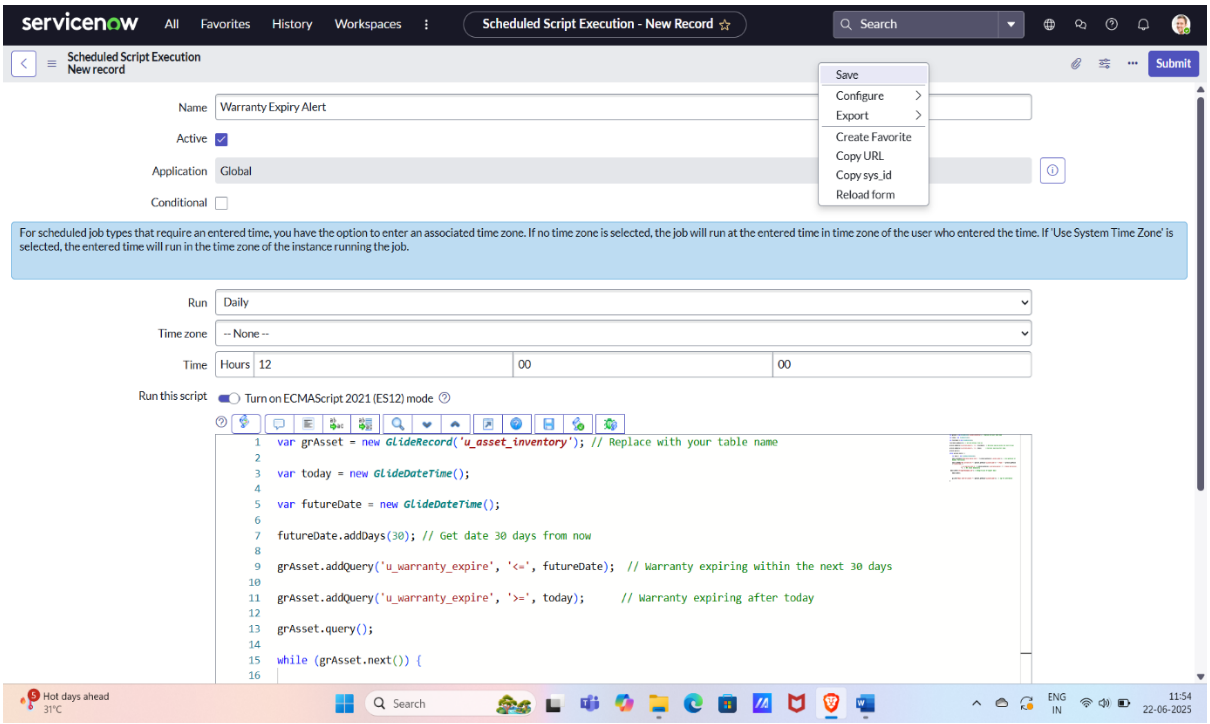
To automate tasks that need to run at specific times or intervals without manual intervention.

USE:

Scheduled jobs are used to automatically check asset status, send maintenance alerts, or generate daily/weekly reports, ensuring timely actions and reducing manual workload.

STEPS:

1. Navigate to System Definition >> Scheduled Job
2. Click on New
3. Name: Warranty Expiry Alert,
4. Run: Daily
5. Time: 12:00
6. Write the script
7. And click on save



Name: Warranty Expiry Alert

Active: ☑

Application: Global

Conditional: □

For scheduled job types that require an entered time, you have the option to enter an associated time zone. If no time zone is selected, the job will run at the entered time in time zone of the user who entered the time. If 'Use System Time Zone' is selected, the entered time will run in the time zone of the instance running the job.

Run: Daily

Time zone: -- None --

Time: Hours 12 | 00 | 00

Run this script: Turn on ECMAScript 2021 (ES12) mode

```
var grAsset = new GlideRecord('u_asset_inventory'); // Replace with your table name

var today = new GlideDateTime();

var futureDate = new GlideDateTime();

futureDate.addDays(30); // Get date 30 days from now

grAsset.addQuery('u_warranty_expire', '<=', futureDate);  // Warranty expiring within the next 30 days

grAsset.addQuery('u_warranty_expire', '>=', today);      // Warranty expiring after today

grAsset.query();

while (grAsset.next()) {

```

MILESTONE 4: REPORT

Activity 1: create report

PURPOSE:

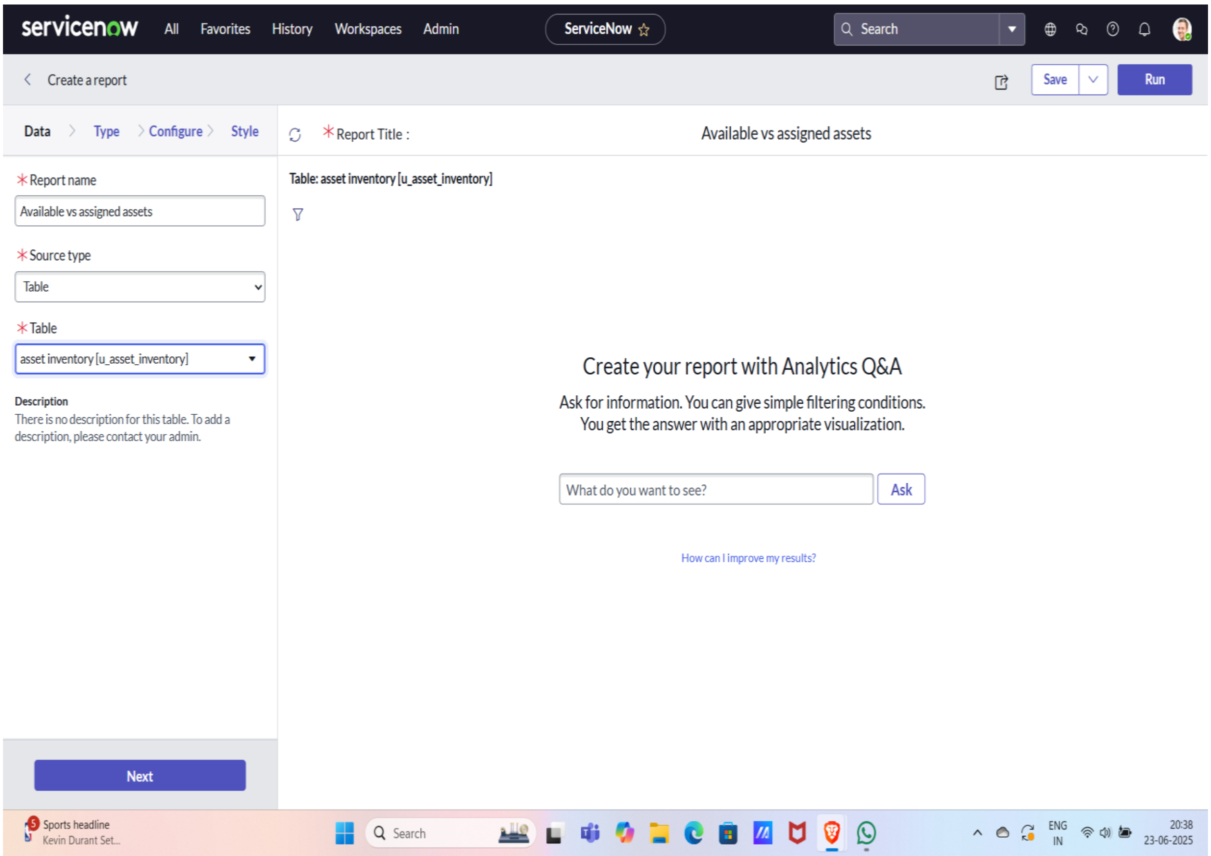
To visually display and analyse data stored in the system for better decision-making.

USE:

Reports help track asset usage, availability, maintenance status, and inventory trends. They support data-driven decisions, improve transparency, and assist in audits or reviews.

STEPS:

1. Navigate To Reports
2. Click on Create New
3. Report Name: Available vs assigned assets, Source Type: Table, Table: Asset Inventory
4. Type: Pie Chart
5. Group By: Status, Aggregation: Count
6. Click on save
7. And then click on Run



*Report name: Available vs assigned assets

*Source type: Table

*Table: asset inventory [u_asset_inventory]

Description
There is no description for this table. To add a description, please contact your admin.

*Report Title : Available vs assigned assets

Table: asset inventory [u_asset_inventory]

Create your report with Analytics Q&A

Ask for information. You can give simple filtering conditions.
You get the answer with an appropriate visualization.

What do you want to see?

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 Tutors

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:

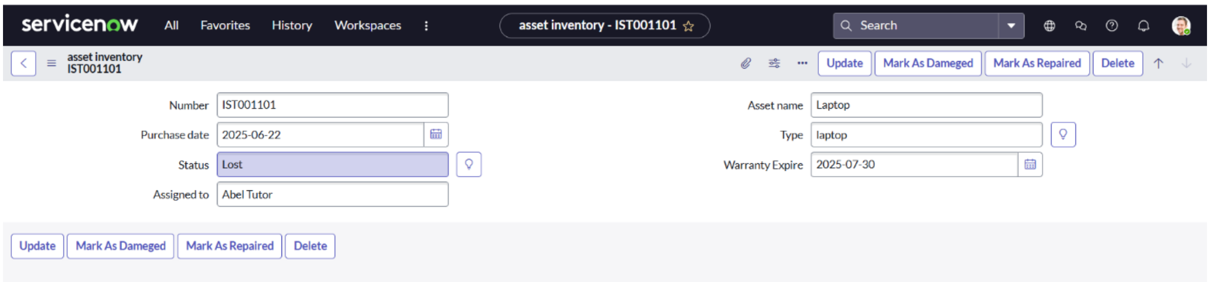
To verify that the scheduled job runs automatically at the set time and performs its task correctly.

USE:

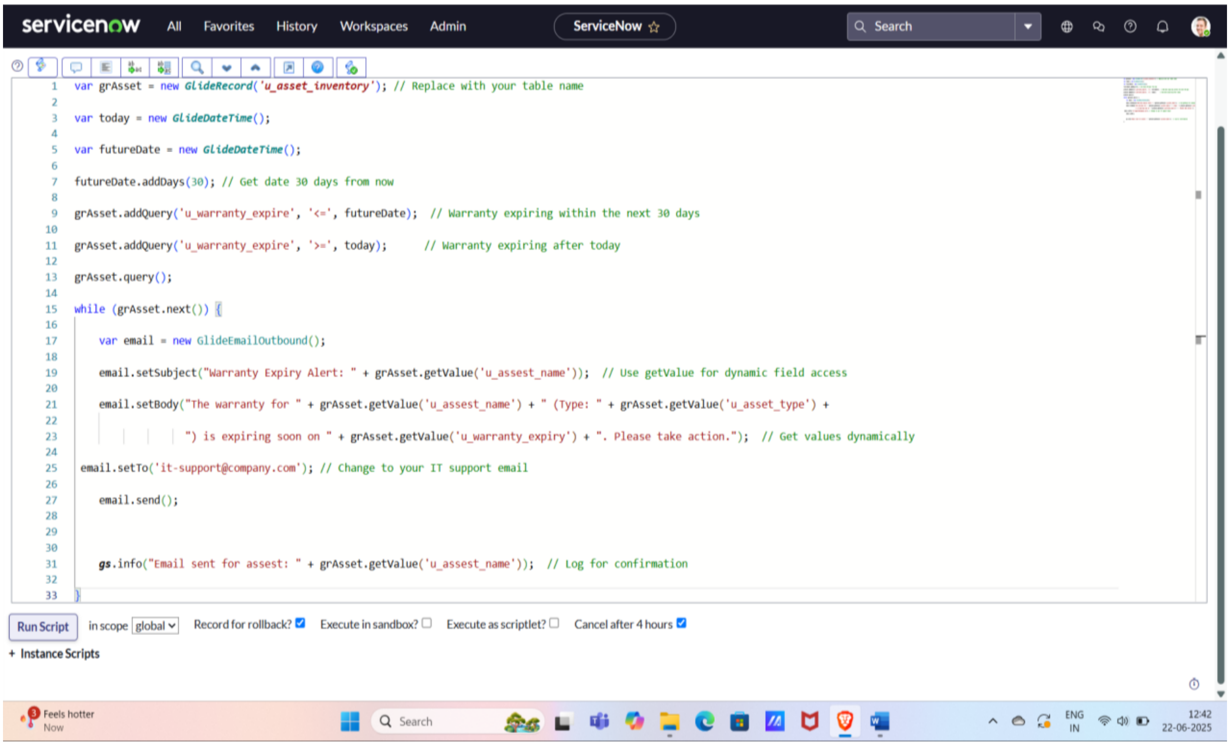
Testing ensures that automated tasks like sending maintenance alerts or generating reports run as expected, helping maintain system accuracy and reducing manual effort.

STEPS:

- Navigate to background scripts
- Write the Scheduled job script in the background scripts
- Click on Run Script
- Check the result



Number	IST001101	Asset name	Laptop
Purchase date	2025-06-22	Type	laptop
Status	Lost	Warranty Expire	2025-07-30
Assigned to	Abel Tutor		



```
var grAsset = new GlideRecord('u_asset_inventory'); // Replace with your table name

var today = new GlideDateTime();

var futureDate = new GlideDateTime();

futureDate.addDays(30); // Get date 30 days from now

grAsset.addQuery('u_warranty_expire', '<=', futureDate);  // Warranty expiring within the next 30 days

grAsset.addQuery('u_warranty_expire', '>=', today);      // Warranty expiring after today

grAsset.query();

while (grAsset.next()) {

    var email = new GlideEmailOutbound();

    email.setSubject("Warranty Expiry Alert: " + grAsset.getValue('u_assest_name'));  // Use getValue for dynamic field access

    email.setBody("The warranty for " + grAsset.getValue('u_assest_name') + " (Type: " + grAsset.getValue('u_asset_type') +

                  ") is expiring soon on " + grAsset.getValue('u_warranty_expiry') + ". Please take action.");  // Get values dynamically

 email.setTo('it-support@company.com'); // Change to your IT support email

    email.send();



    gs.info("Email sent for assest: " + grAsset.getValue('u_assest_name'));  // Log for confirmation

}
```

Run Script in scope global Record for rollback? ☑ Execute in sandbox? □ Execute as scriptlet? □ Cancel after 4 hours ☑

+ Instance Scripts