

4.PROJECT DESIGN

TEAM ID	LTVIP2025TMID30940
PROJECT NAME	Asset Management Portal

Proposed Solution:

S.No	Parameter	Description
1	Problem Statement	Organizations often face challenges in tracking,managing,and maintaining their physical and digital assets,leading to asset loss,inefficiency and inaccurate records.
2	Idea/Solution	The proposed solution is an asset management portal ,a centralized and automated web based platform that streamlines the tracking.
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.
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	The asset management portal follows a software-as-a-service business model.
6	Scalability of the solution	The asset management portal is highly scalable,capable of handling increasing numbers of users.

Asset Management Portal

Asset Management:

Asset Management is a platform designed to help organizations track, manage, and optimize their IT assets throughout their lifecycle, from procurement to disposal. It provides a centralized system for managing hardware, software, and cloud assets, offering features like lifecycle management, cost tracking, compliance reporting, and automated workflows.

Importance:

1. Enhanced Visibility and Control.
2. Improved Asset Lifecycle Management.
3. Cost Optimization.
4. Enhanced Compliance and Security.
5. Improved Incident Management.

Milestone 1: Tables

Table is a structured container that holds records, similar to a database table. Each table represents a specific type of data, like users, incidents, or assets, and is composed of rows (records) and columns (fields).

Use

Tables are fundamental for organizing and storing data. They are essentially database tables that hold records, similar to how a spreadsheet organizes information into rows and columns.

Activity 1: Create Table

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**

Name : asset inventory

Home | ServiceNow Developers

asset inventory | Table | ServiceNow

dev344266.service-now.com/now/nav/ui/classic/params/target/sys_db_object.do%3Fsys_id%3D7c80e0bd931e62509e127ee32bba10ce%26sysparm_action=edit

servicenowAllFavoritesHistoryWorkspacesAdminTable - asset inventory

<≡Table
asset inventory

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 corresponds to a column on that table. Applications can use tables to store information about their users, assets, or other resources.

* Labelasset inventory

* Nameu_asset_inventory

ColumnsControlsApplication Access

≡🔍Table Columnsfor textSearch

Dictionary Entries

Column label	Type	Reference	Max length
Updated	Date/Time	(empty)	
Created by	String	(empty)	
Created	Date/Time	(empty)	
Updates	Integer	(empty)	
Updated by	String	(empty)	
Sys ID	Sys ID (GUID)	(empty)	
+ Insert a new row...			

DeleteUpdateDelete All Records

Fields are individual data points within a record (like an incident or a problem) that store specific information. They are analogous to columns in a database table.

Use

Fields are used to store specific pieces of information within a record of a table. They act as containers for data, allowing users to input, view, and manage information relevant to the record.

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

The screenshot shows the ServiceNow interface for configuring the 'asset inventory' table. The 'Columns' tab is selected, displaying a list of dictionary entries. The table has 13 columns in total. The visible entries are as follows:

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

Milestone 2: UI Action

UI actions are custom actions that allow users to interact with records on forms and lists, appearing as buttons, links, or context menu items.

Use

UI actions help streamline workflows, automate tasks, and improve user efficiency by providing quick access to common functions.

Activity 1: UI Action 1

Steps

1. Navigate to System Definition >> UI action
2. Click on New
3. Fill in the details ;

- a. Name : Mark As Lost
- b. Table : Asset Inventory
- c. Action name : mark_as_lost
- d. Condition : current.u_status != 'Lost'
- e. Script :

```
current.u_status = 'Lost';  
current.update();  
action.setRedirectURL(current);
```

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

The screenshot shows the ServiceNow 'UI Action - Mark As Lost' configuration page. The form is divided into several sections:

- General Information:**
 - Name: Mark As Lost
 - Table: asset_inventory[u_asset_inventory]
 - Order: 100
 - Action name: mark_as_lost
 - Active: ☒
 - Show insert: ☒
 - Show update: ☒
 - Client: ☐
 - Overrides: (empty search field)
- Application:** Global
- Form Options:**
 - Form button: ☒
 - Form context menu: ☐
 - Form link: ☐
 - Form style: -- None --
- List Options:**
 - List banner button: ☐
 - List bottom button: ☐
 - List context menu: ☐
 - List choice: ☐
 - List link: ☐
 - List style: -- None --
- Messages:** (empty text area)
- Comments:** (empty text area)
- Hint:** (empty text area)
- Condition:** current.u_status != 'lost'

At the bottom right, there are 'Update' and 'Delete' buttons. The bottom of the screen shows a Windows taskbar with the date '25-09-2023' and time '10:12'.

Activity 2: UI Action 2

Steps

4. Navigate to System Definition >> UI action
5. Click on New
6. Fill in the details ;
 - a. Name : Mark As Repaired
 - b. Table : Asset Inventory
 - c. Action name : mark_as_Repaired
 - d. Condition : `current.u_status = 'Lost' || current.u_status == 'Damaged'`
 - e. Script :

```
current.u_status = 'Availablet';
current.update();
action.setRedirectURL(current);
```

6. Check the form button box
7. Click on save

The screenshot shows the ServiceNow 'UI Action - Mark As Repaired' configuration page. The form includes the following fields and settings:

- Name:** Mark As Repaired
- Table:** asset inventory [u_asset_inventory]
- Order:** 100
- Action name:** mark_as_repaired
- Active:** ☒
- Show insert:** ☒
- Show update:** ☒
- Client:** ☐
- Overrides:** (empty field with search icon)
- 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:** (empty text area)
- Comments:** (empty text area)
- Hint:** (empty text area)
- Condition:** current.u_status == 'Damaged' || current.u_status == 'Lost'

Activity 3: UI Action 3

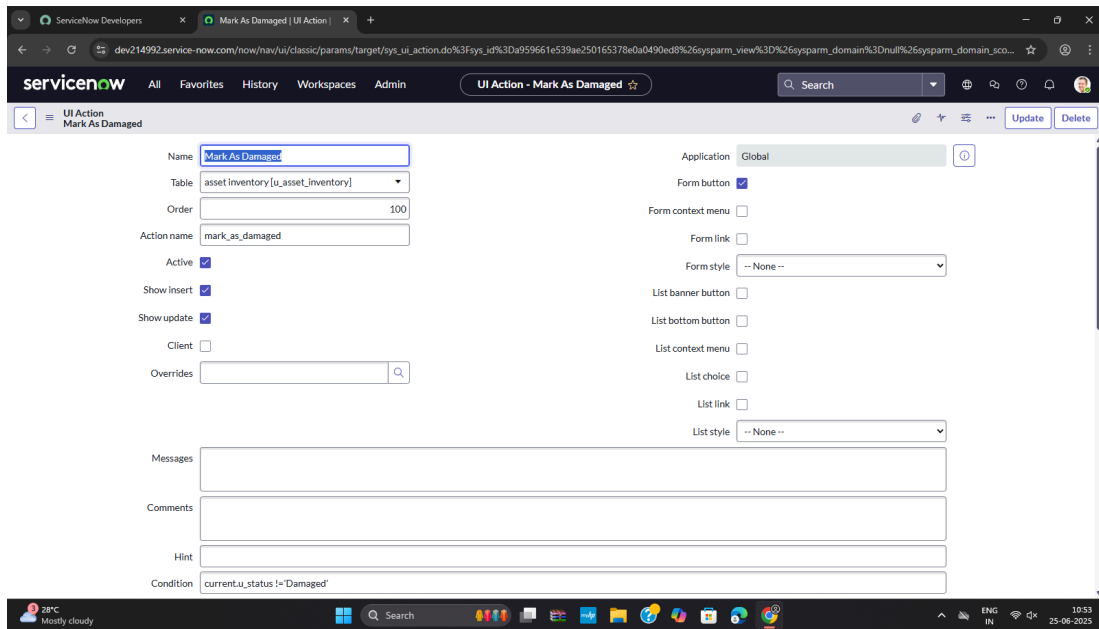
Steps

1. Navigate to System Definition >> UI action
2. Click on New
3. Fill in the details ;
 - a. Name : Mark As Damaged
 - b. Table : Asset Inventory
 - c. Action name : mark_as_damaged
 - d. Condition : current.u_status != 'Damaged'
 - e. Script :

```
current.u_status = 'Damaged';
current.update();

action.setRedirectURL(current);
```

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



Milestone 3: Scheduled Job

Scheduled job is a pre-defined task that executes automatically at a specified time or on a recurring schedule. These jobs automate various tasks, such as generating and distributing reports, creating and scheduling records, or running scripts.

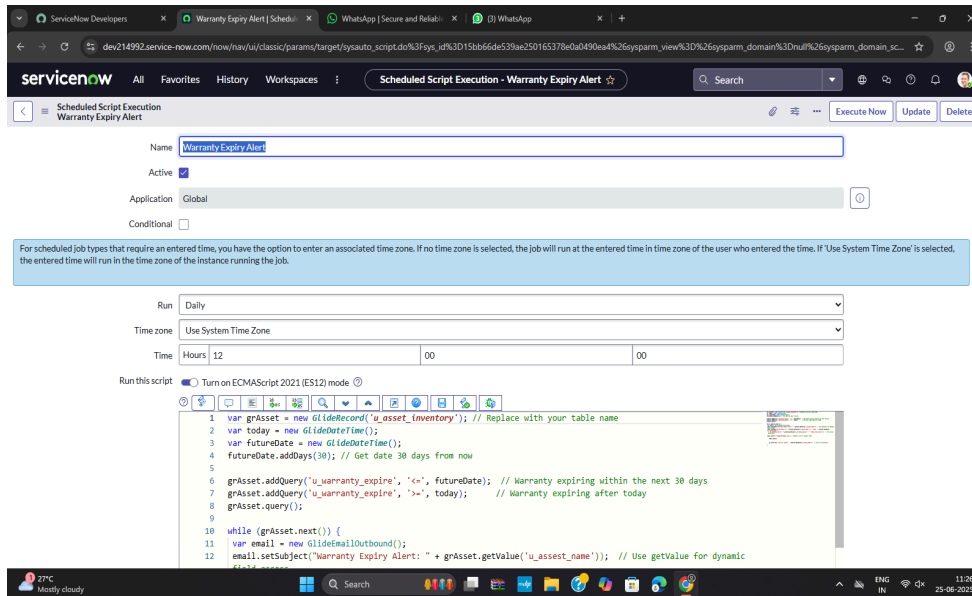
Use

Scheduled jobs are automated tasks that execute at a specified time or on a recurring schedule. They are used to perform various operations, including running scripts, generating reports, and creating records.

Step

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



Milestone 4: Report

Reports allow users to analyze and visualize data from their instance. They can be used to identify trends, monitor performance, track progress, and more. Reports can be created, viewed, and run through the "Reports" module in the application navigator.

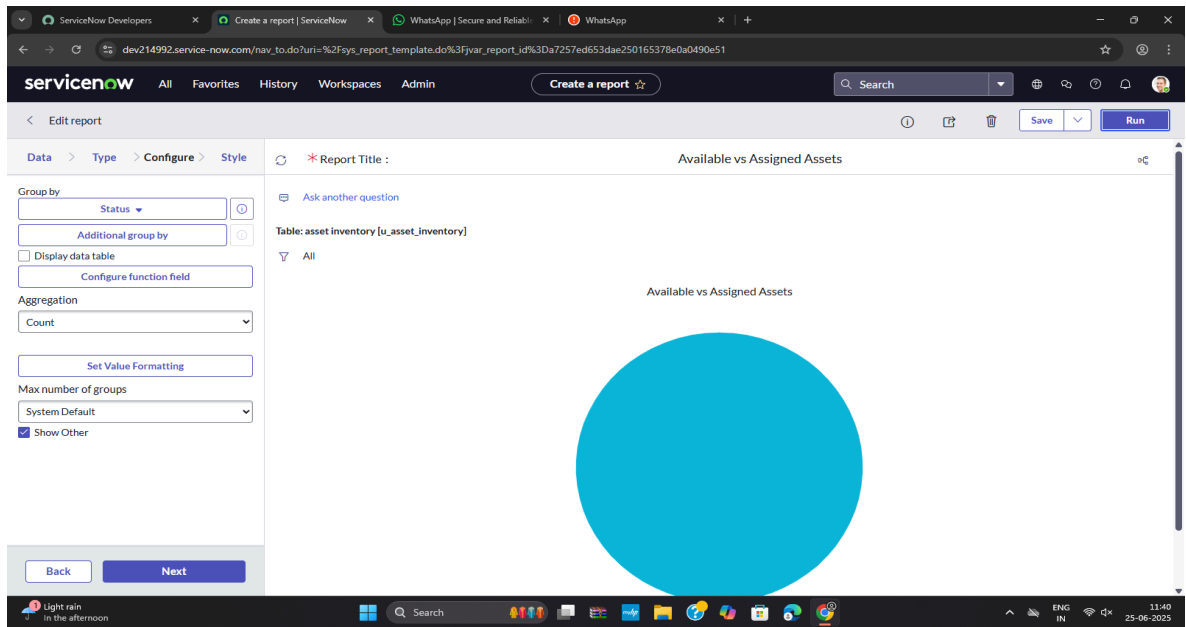
Use

Reports are used to extract, visualize, and analyze data from your instance. They allow users to gain insights, monitor performance, track trends, and make data-driven decisions.

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



Milestone 5: Testing

Testing involves ensuring the proper functionality and performance of the ServiceNow platform, including its applications, configurations, integrations and workflows.

Use

Testing ensures the quality, reliability, and security of applications built on the platform. It involves validating custom applications, configurations, integrations, and workflows before they are released to production.

Activity 1: Testing UI Action

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.

The screenshot shows the ServiceNow 'asset inventory - New Record' form. The form is titled 'asset inventory - New Record' and has a search bar. The form contains the following fields and values:

Field	Value
Number	AST001011
Purchase date	2025-05-03
Assigned to	Abel Tutor
* Status	available
Asset name	Laptop
Warranty Expire	2025-06-15
* Type	laptop

Buttons at the bottom: Submit, Mark As Damaged, Mark As Lost.

ServiceNow Developers | AST001015 | asset inventory | dev214992.service-now.com | WhatsApp | Secure and Reliable | (4) WhatsApp

dev214992.service-now.com/now/nav/ui/classic/params/target/u_asset_inventory.do%3Fsys_id%3D54c8f61e53dae250165378e0a0490ec0%26sysparm_view%3D%26sysparm_domain%3Dnull%26sysparm_domain...

servicenow All Favorites History Workspaces Admin asset inventory - AST001015 Search

asset inventory AST001015 Update Mark As Damaged Mark As Lost Mark As Repaired Delete

Number: AST001015
 Purchase date: 2025-05-03
 Assigned to: Abel Tutor
 * Status: Lost
 Asset name: Laptop
 Warranty Expire: 2025-06-15
 * Type: laptop

Update Mark As Damaged Mark As Lost Mark As Repaired Delete

ServiceNow Developers | ServiceNow | dev214992.service-now.com | WhatsApp | Secure and Reliable | (4) WhatsApp

dev214992.service-now.com/now/nav/ui/classic/params/target/u_asset_inventory_list.do%3Fsysparm_first_row%3D1%26sysparm_query%3D%26sysparm_view%3D...

servicenow All Favorites History Workspaces Admin asset inventories Search Actions on selected rows... New

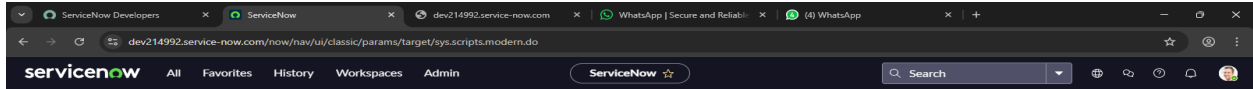
All	Number	Asset name	Assigned to	Purchase date	Status	Type	Warranty Expire
<input type="checkbox"/>	AST001015	Laptop	Abel Tutor	2025-05-03	Lost	laptop	2025-06-15

1 to 1 of 1

Activity 2: Testing Scheduled Job

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



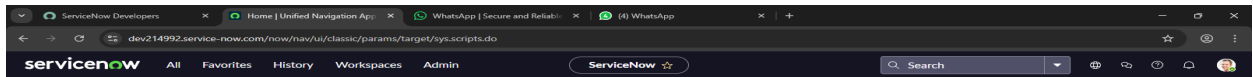
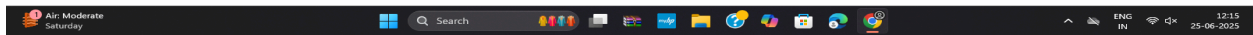
Running freeform script can cause system disruption or loss of data.

Run script (JavaScript executed on server)

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

In scope Record for rollback? ☒ Execute in sandbox? ☐ Execute as scriptlet? ☐ Cancel after 4 hours ☒

+ Instance Scripts



[0:00:00.184] Script completed in scope global: script

Script execution history and recovery [available here](#)

