INTRODUCTION

Project: "Asset Management Portal"

The Asset Management Portal is a comprehensive platform designed to efficiently manage the lifecycle of both physical and digital assets within an organization. It provides a centralized system for tracking, allocating, and maintaining assets, enabling employees to request and receive assets through a user – friendly interface. The Asset Management Portal plays a crucial role in modern organizations by offering a centralized system for managing both physical and digital assets. One of its primary uses is to enable employees to request and receive assets efficiently through a user-friendly interface. This request system streamlines internal operations and ensures proper documentation of asset usage. The portal also automates the allocation and tracking of assets, maintaining records of which user is assigned which asset, along with time stamps, usage logs, and return status. It supports the entire asset lifecycle—starting from procurement, through allocation and maintenance, to final disposal or decommissioning. This full-spectrum lifecycle management allows organizations to control the usage, performance, and availability of all assets.

In addition, the portal is integrated with features such as preventive maintenance scheduling and automated alert systems. These features help monitor the health and performance of assets in real time, triggering alerts for upcoming maintenance, warranty expiry, or when a replacement is due. This reduces unplanned downtime and improves asset longevity. The system maintains a complete inventory of assets across departments and locations, offering complete visibility into asset availability, condition, and location. Furthermore, it generates real-time reports and analytics that help managers and decision-makers track utilization, plan procurement, and analyse trends for cost optimization.

The benefits of the Asset Management Portal are numerous and impactful. It enhances operational efficiency by automating routine tasks and reducing the workload on administrative staff. It also reduces costs by minimizing asset loss, avoiding duplicate purchases, and improving asset utilization. By maintaining a centralized digital record, the system ensures better compliance with audit and regulatory requirements. Additionally, it improves transparency and accountability in asset handling. Users benefit from a simplified process for accessing resources, while administrators gain the ability to make informed decisions using real-time data. Ultimately, the portal supports sustainable and responsible asset usage, contributing to long-term organizational success.

IDEATION PHASE

PROBLEM STATEMENT:

The Asset Management Portal will streamline the tracking, management, and allocation of both physical and digital assets across an organization. Employees will be able to request and receive assets through an intuitive portal, while administrators can manage the entire asset lifecycle, from procurement to disposal. The portal will also automate asset assignment, ensure accurate record-keeping, and generate real-time reports on asset utilization and condition. Alerts will be triggered for maintenance or replacement needs, ensuring optimal asset performance and reducing downtime. By centralizing asset management, the platform will improve operational efficiency, reduce asset loss, and support informed decision-making.

What are challenges faced without Asset management portal:

- Difficulty in tracking and locating assets accurately
- Increased chances of asset loss, theft, or duplication
- No automated alerts for maintenance or asset expiry
- Poor visibility into asset utilization and lifecycle
- Challenges in generating reports for audits or planning

OBJECTIVE:

Automate the tracking, allocation, and management of physical and digital assets, ensuring real-time updates on asset status, condition, and availability. Provide actionable insights for asset maintenance and replacement needs, notify administration about asset discrepancies or issues, and generate report to optimize asset utilization and decision-making.

REQUIREMENT ANALYSIS

SOLUTION REQUIREMENT:

TEAM ID	LTVIP2025TMID30624
PROJECT NAME	Asset Management Portal

Functional Requirements:

Following are the functional requirements.

FR NO	**Functional Requirement**	**Sub Requirement**
FR-1	TABLES	Create table (asset inventory), create fields in the table
FR-2	UI ACTION	Create 3 UI actions they are (mark as lost, mark as repaired, mark as damaged)
FR-3	SCHEDULED JOB	Create Scheduled job (warranty expire alerts) and give the script
FR-4	REPORT	Create reports in service now give the name, type (pia chart), and configure
FR-5	TESTING	Testing UI action and Scheduled job

Non-Functional Requirements:

Following are the functional requirements.

FR NO.	**Non-Functional Requirement**	**Description**
NFR-1	Usability	The Asset Management Portal provides a user-friendly and intuitive interface that allows employees and

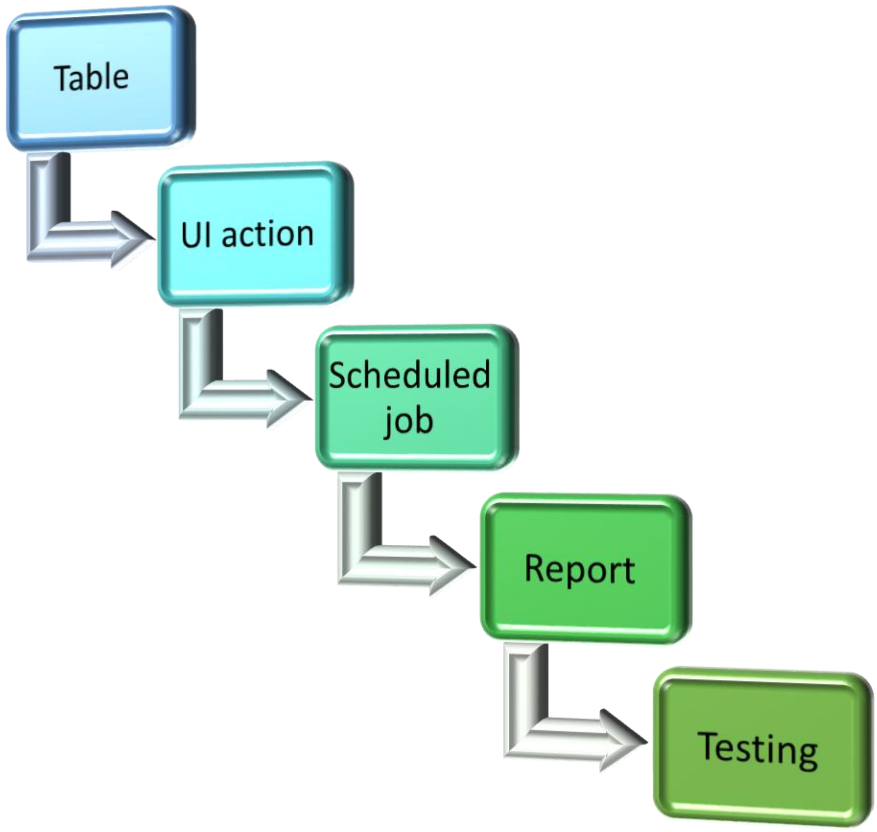
		administrators to easily manage, track, and request assets with minimal training or technical knowledge.
NFR-2	Security	The Asset Management Portal ensures data protection through user authentication, role-based access controls, and secure encryption to prevent unauthorized access and safeguard asset information.
NFR-3	Reliability	The Asset Management Portal consistently performs its functions without failure, ensuring accurate asset tracking, timely updates, and dependable system availability for all users.
NFR-4	Performance	The Asset Management Portal delivers fast response times, efficient processing of asset data, and smooth handling of multiple user requests without system lag or delays.
NFR-5	Availability	The Asset Management Portal is accessible at all times, ensuring users can manage and track assets anytime without downtime or service interruptions.
NFR-5	Scalability	The Asset Management Portal can efficiently handle growing numbers of users, assets, and data, making it adaptable to the expanding needs of any organization.

Data Flow Diagram:

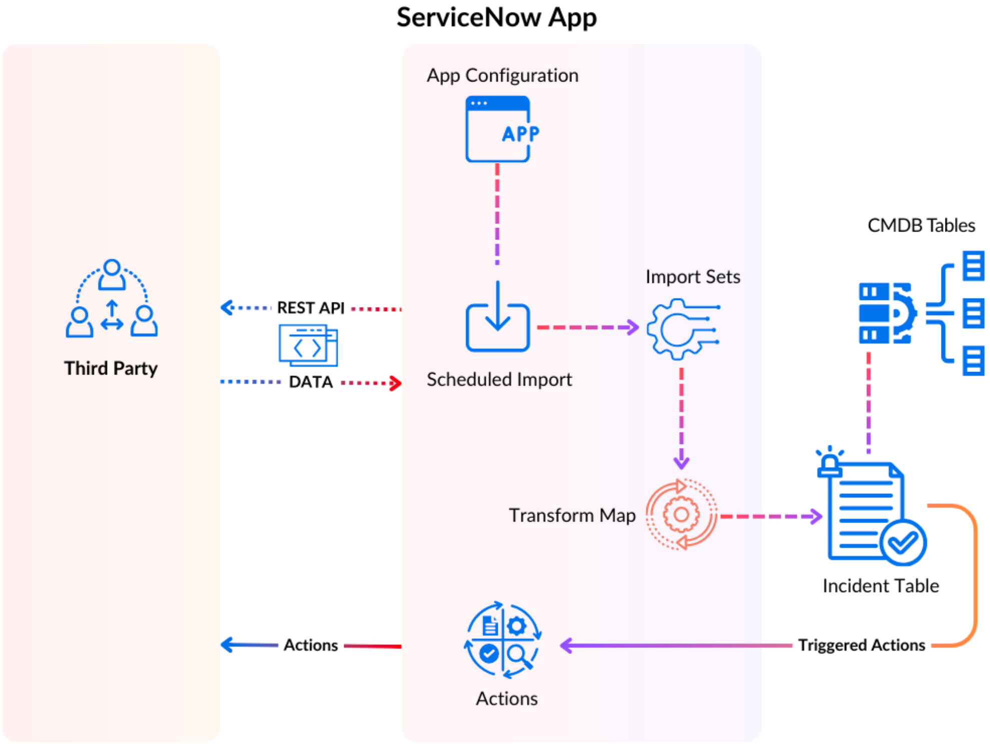
A Data Flow Diagram (DFD) is a simple graphical tool used to show how data moves through a system. It illustrates the input, processing, storage, and output of data using symbols like arrows, circles, and rectangles. In the Asset Management Portal, a DFD helps visualize how asset requests, approvals, assignments, and tracking are handled between users, the system, and databases.

Uses:

- Visualizes system processes and how data moves between them.
- Helps in understanding system structure clearly and quickly.
- Aids in system design and development by showing data input, processing, and output.



Technology Stack:



Architecture of ServiceNow

- Third-party system sends data via REST API to ServiceNow.
- Scheduled Import collects data into Import Sets.
- Transform Map processes and maps data to target tables.
- Data is stored in the Incident Table.
- Incident Table links to CMDB Tables for asset/configuration info.
- Triggered actions automate tasks like notifications or escalations.
- Responses/actions can be sent back to the third-party system.

Project Design:

Proposed Solution:

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

S. No	Parameter	Description
1	Problem statement (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 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, 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

What is ASSET?

An asset in the Asset Management Portal refers to any valuable item or resource owned and used by an organization that needs to be tracked, maintained, and managed throughout its lifecycle.

TYPES:

- Physical Assets – Laptops, desktops, printers, furniture, tools, etc.
- Digital Assets – Software licenses, cloud services, digital certificates, etc.
- IT Assets – Network devices, servers, storage units, etc.
- Consumables – Toners, batteries, cables (tracked as expendable items).

MILESTONE 1: TABLE

PURPOSE:

- To define a structured storage space for asset-related data.
- To organize and manage information like asset name, type, status, owner, etc.
- To serve as the foundation for forms, lists, workflows, and reports in the portal.

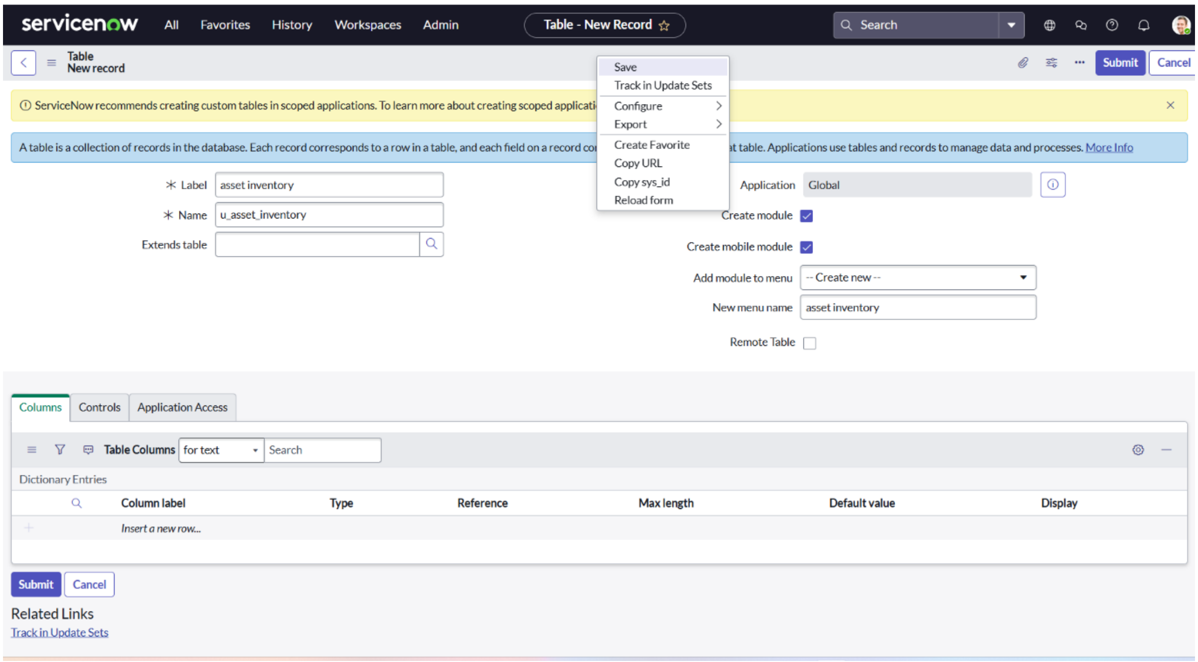
USE:

- Stores details of each asset in rows (records) and columns (fields).
- Enables easy tracking and retrieval of asset information.
- Supports workflows like asset assignment, return, maintenance, etc.
- Acts as the data source for dashboards, reports, and UI views.
- Helps maintain data consistency, accuracy, and accountability.

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
    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
Remote Table	□

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

MILESTONE 1: TABLE

PURPOSE:

- To define specific data attributes for each asset (e.g., name, type, status).
- To ensure organized and structured data entry in the table.
- To support form creation, filtering, and reporting within the portal.

USE:

- Captures key details like Asset ID, Category, Owner, Purchase Date, Location, etc
- Enables searching, sorting, and filtering of asset records.
- Supports automated workflows (e.g., assign based on status or location).
- Helps generate accurate reports and dashboards.
- Ensures data consistency and validation across records.

Activity 2: create fields

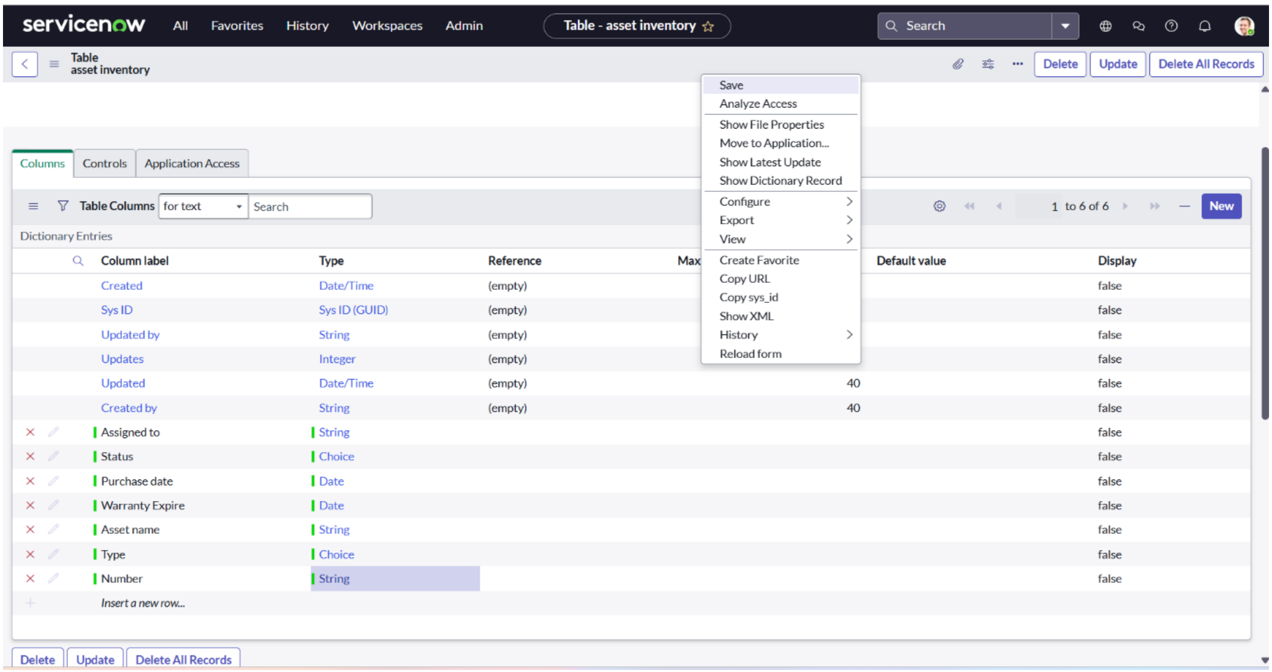
STEPS:

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

MILESTONE 2: UI ACTION



Column label	Type	Reference	Max	Default value	Display
Created	Date/Time	(empty)			false
Sys ID	Sys ID (GUID)	(empty)			false
Updated by	String	(empty)			false
Updates	Integer	(empty)			false
Updated	Date/Time	(empty)	40		false
Created by	String	(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...					

PURPOSE:

- To add a custom button or link on a form or list.
- To allow users to perform specific actions quickly (e.g., assign, return, approve).
- To enhance user interaction and improve workflow execution.

USE:

- Let’s users take actions like
  - Assign Asset
  - Return Asset
  - Send for Repair
  - Mark as lost
  - Mark as repaired
  - Mark as damaged
- Speeds up common tasks with one-click execution.
- Improves user experience by reducing navigation steps
- Supports custom logic or conditions using scripts.
- Ensures consistent and efficient asset handling across the system.

Activity 1: create UI action 1 (Mark as lost)

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:

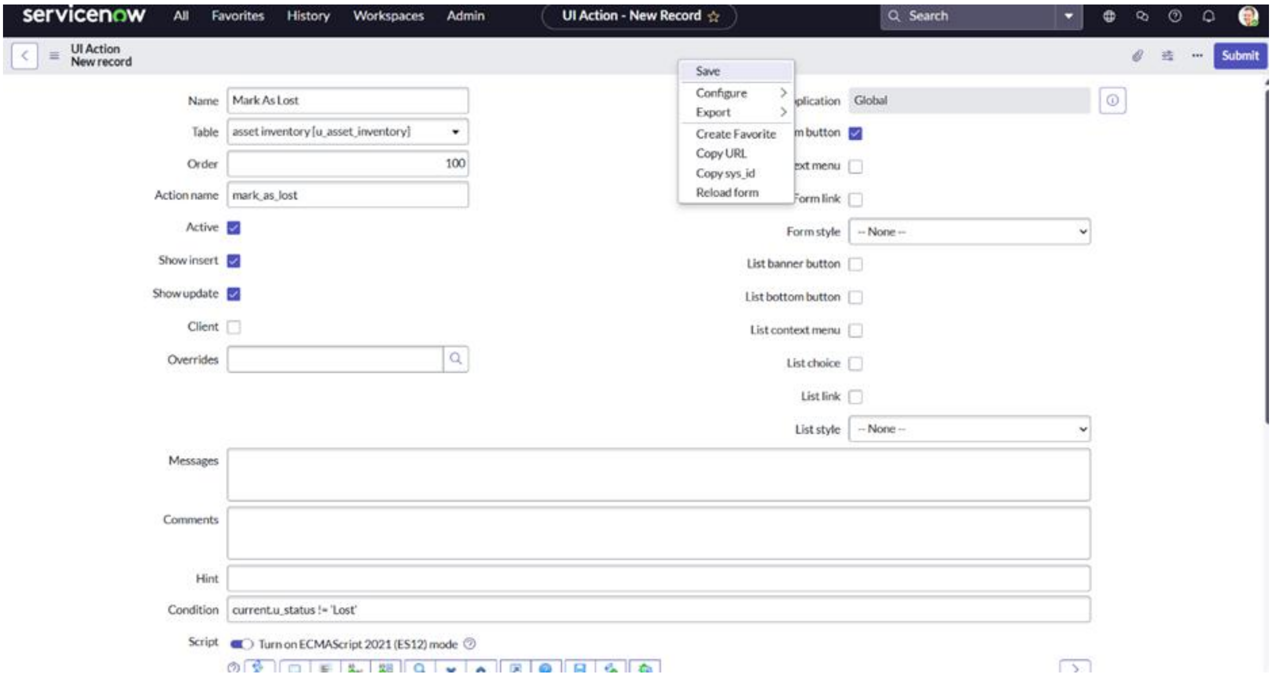
```
current.u_status = 'Lost';
```

current.update();

action.setRedirectURL(current);

4. Check the form button box

5. Click on save



Name: Mark As Lost
Table: asset inventory [u_asset_inventory]
Order: 100
Action name: mark_as_lost
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 != 'Lost'
Script: Turn on ECMAScript 2021 (ES12) mode

MILESTONE 2: UI ACTION

Activity 2: create UI action 2 (Mark as repaired)

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

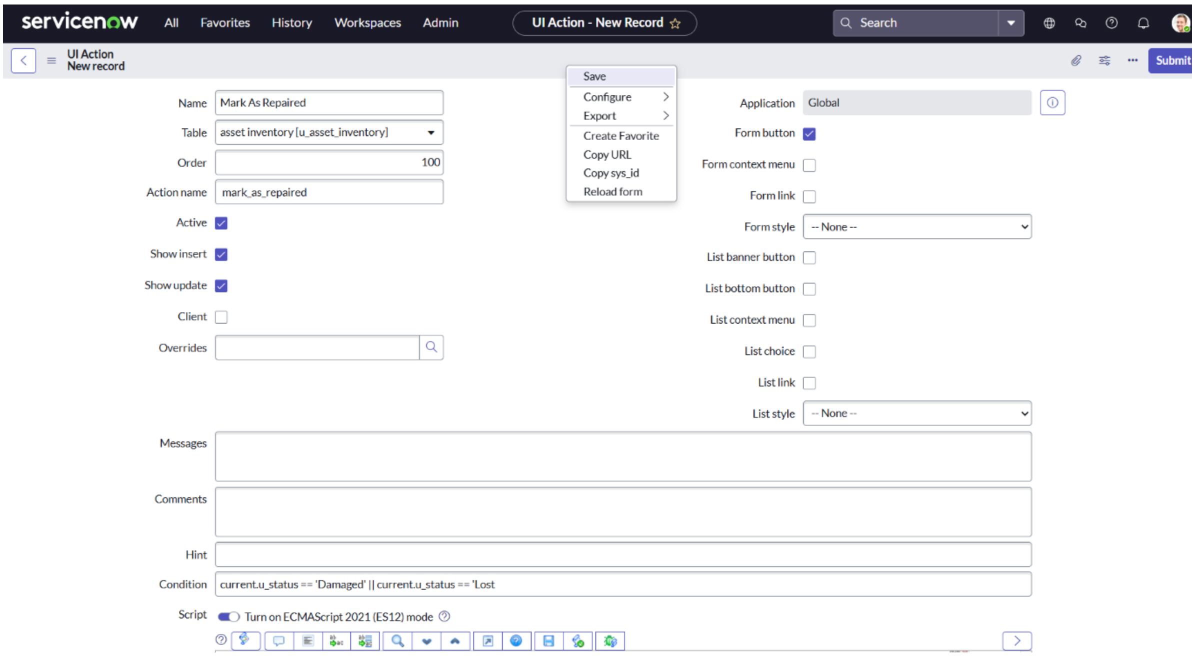
4. Condition: current.u_status == 'Damaged' || current.u_status == 'Lost’

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(Mark as damaged)

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

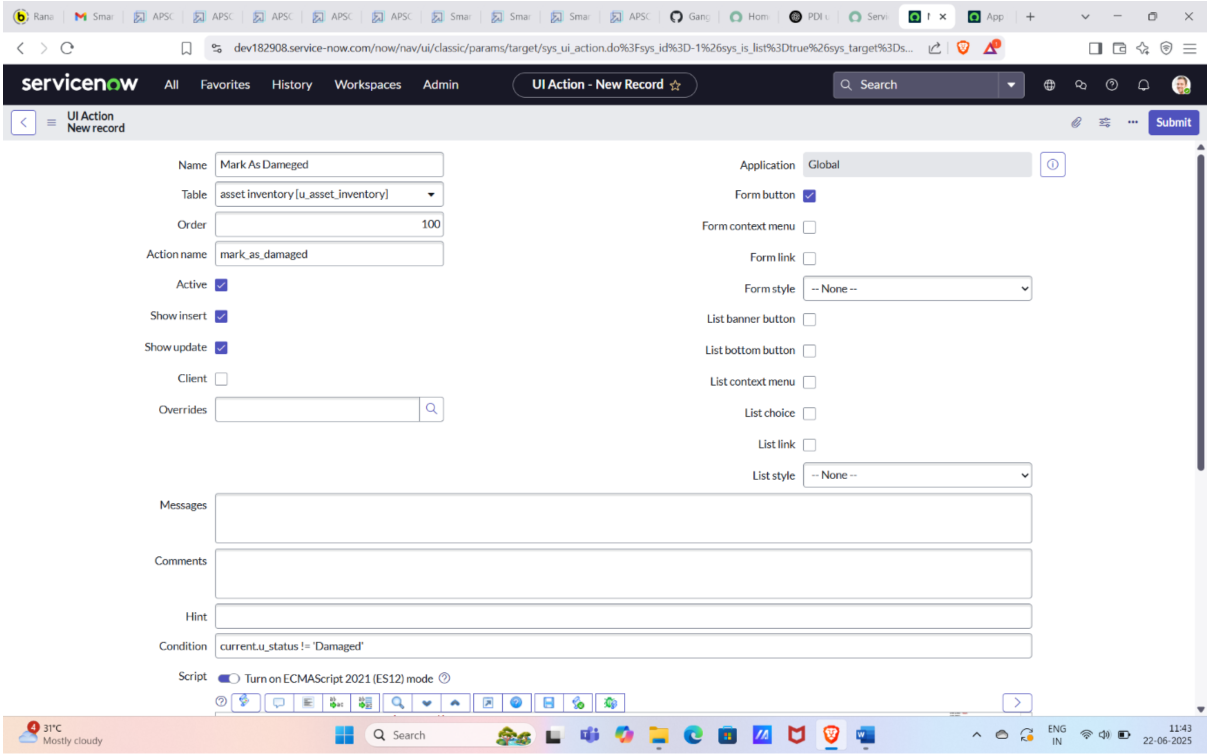
Condition: current.u_status != 'Damaged'

Script:

```
current.u_status = 'Damaged';

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

4. Check the form button box

5. Click on save


dev182908.service-now.com/now/nav/ui/classic/params/target/sys_ui_action.do%3Fsys_id%3D-1%26sys_is_list%3Dtrue%26sys_target%3Ds...
servicenow
All
Favorites
History
Workspaces
Admin
UI Action - New Record
Search
UI Action
New record
Submit

Name	Mark As Dameged	Application	Global
Table	asset inventory [u_asset_inventory]	Form button	☑
Order	100	Form context menu	□
Action name	mark_as_damaged	Form link	□
Active	☑	Form style	-- None --
Show insert	☑	List banner button	□
Show update	☑	List bottom button	□
Client	□	List context menu	□
Overrides		List choice	□
		List link	□
		List style	-- None --

Messages

Comments

Hint

Condition current.u_status != 'Damaged'

Script ● Turn on ECMAScript 2021 (ES12) mode

MILESTONE 3: SCHEDULED JOB

PURPOSE:

- To automate tasks at specific times or intervals.

- To ensure routine operations run without manual input.
- To improve efficiency and reliability of time-based processes.

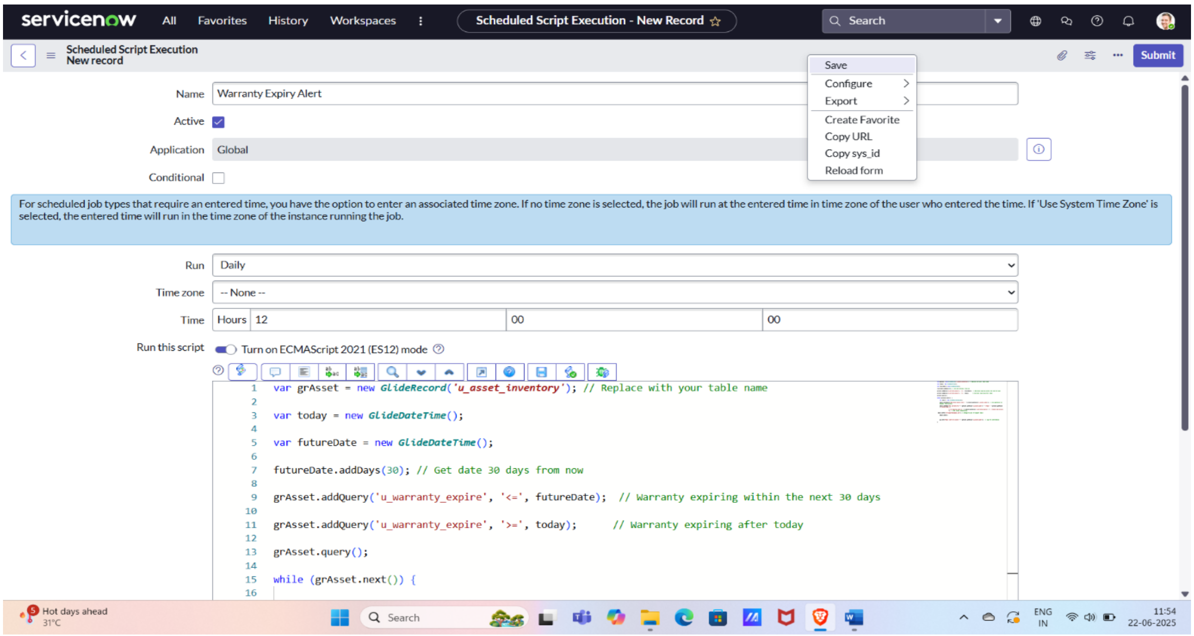
USE:

- Sends automatic maintenance reminders or alerts.
- Generates daily, weekly, or monthly asset reports.
- Updates asset statuses based on conditions (e.g., warranty expiry).
- Reduces manual effort by handling background tasks.
- Ensures timely actions are taken to maintain asset performance.

Activity 1: create scheduled job (Warranty expire alerts)

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

PURPOSE:

- To visually present data for better understanding and analysis.
- To help in monitoring asset performance, usage, and status
- To support data-driven decisions with clear insights.

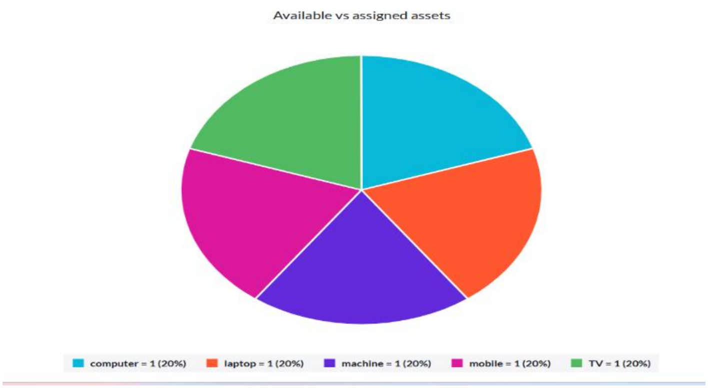
USE:

- Tracks asset availability, assignments, and maintenance history
- Identifies underutilized or idle assets.
- Helps in budget planning and forecasting.
- Supports audit and compliance reporting.
- Improves management decisions through real-time data visibility.

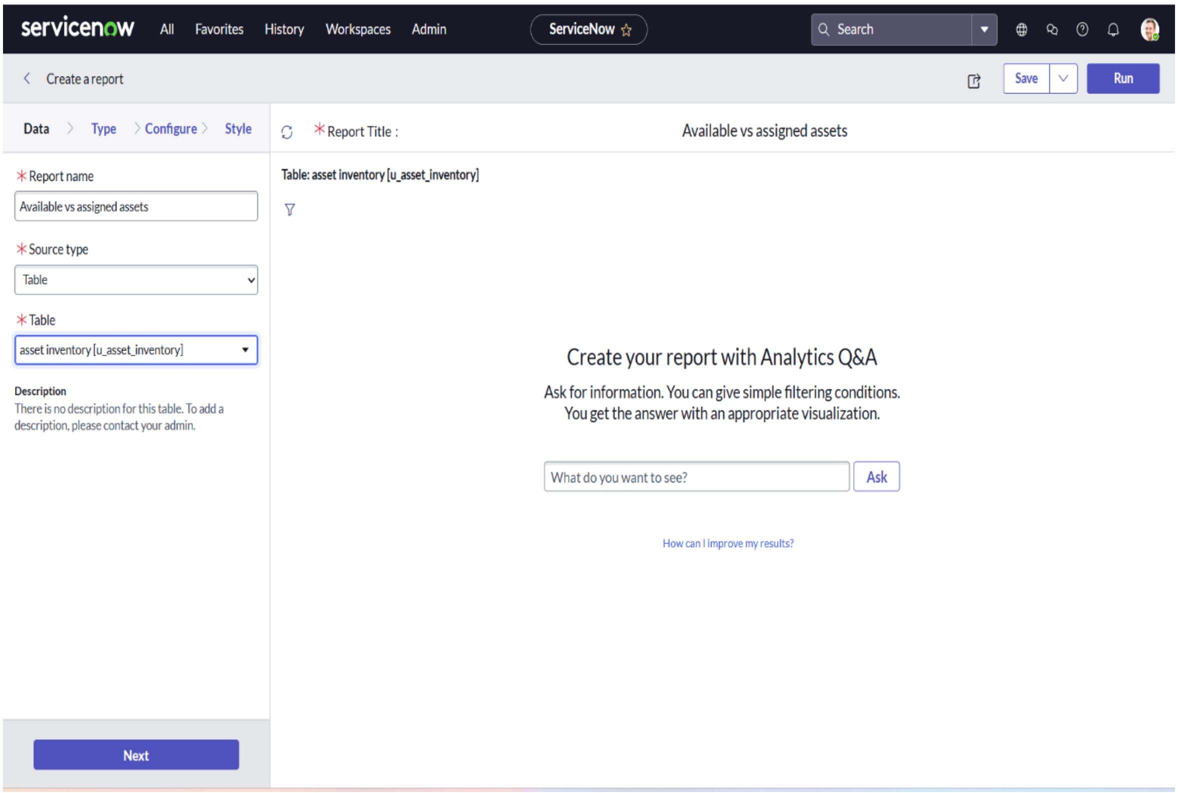
Activity 1: create report

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 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

PURPOSE:

- To verify that the created UI action (button or link) works as expected.
- To ensure the action triggers the correct function without errors.
- To identify and fix bugs before going live.

USE:

- Confirms that actions like "Assign Asset" or "Return Asset" perform correctly.
- Ensures a smooth user experience by preventing broken or faulty buttons.
- Validates that scripts, conditions, and permissions tied to the UI action function properly.
- Improves system reliability and reduces user confusion or errors.

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.

MILESTONE 5: TESTING

PURPOSE:

- To ensure the scheduled job runs automatically at the defined time.
- To verify the job performs the intended task correctly (e.g., send alert, update record).
- To detect and fix errors in automation before production use.

USE:

- Confirms that alerts or reports are generated on time.
- Ensures asset maintenance reminders are sent without failure.
- Validates data updates or status changes happen as scheduled.
- Improves automation reliability and reduces manual oversight.
- Helps maintain consistent and accurate system performance.

Activity 2: testing scheduled job

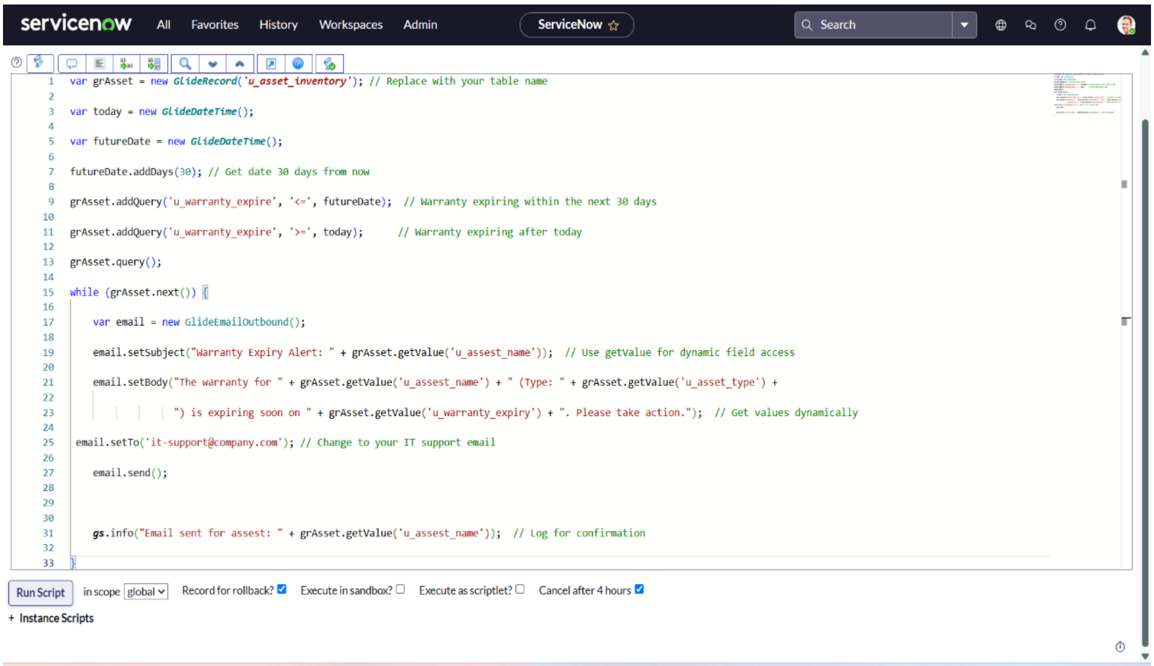
STEPS:

1. Navigate to background scripts

2. Write the Scheduled job script in the background scripts

3. Click on Run Script

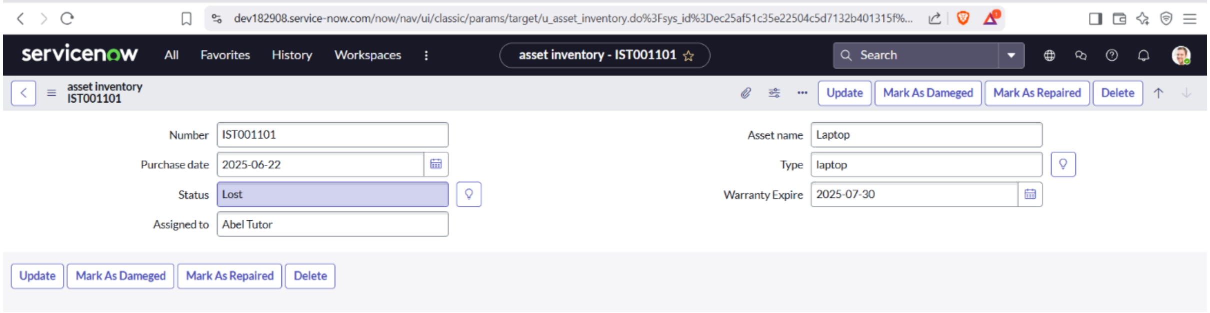
4. Check the result



```
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

}
```

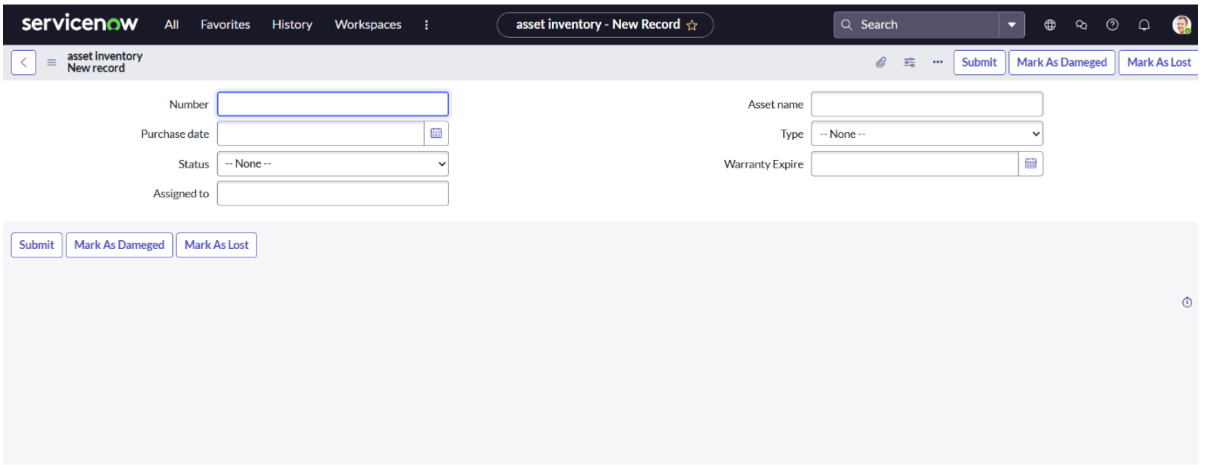
Run Script in scope global Record for rollback? ☑ Execute in sandbox? □ Execute as scriptlet? □ Cancel after 4 hours ☑

+ Instance Scripts


dev182908.service-now.com/now/nav/ui/classic/params/target/u_asset_inventory.do%3Fsys_id%3Dec25af51c35e22504c5d7132b401315f%...
servicenow
All
Favorites
History
Workspaces
asset inventory - IST001101 ☆
Search
asset inventory
IST001101
Update
Mark As Dameged
Mark As Repaired
Delete

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

Update
Mark As Dameged
Mark As Repaired
Delete

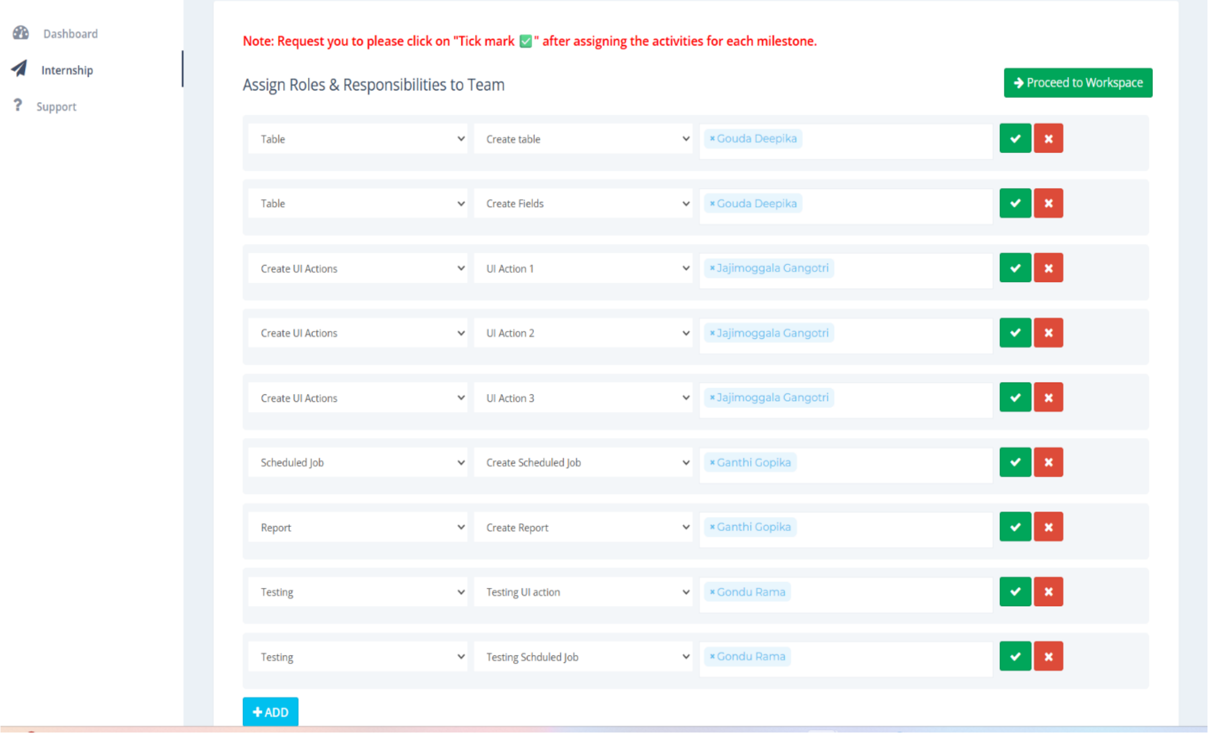


Number		Asset name	
Purchase date		Type	-- None --
Status	-- None --	Warranty Expire	
Assigned to			

Project planning & Scheduling:

Assigned Task to the Group members as shown in below.

Functional requirement	**User story**	**No of Activity**	**Team Members**
TABLE	As an administrator, I want to create and manage a table to store asset information so that I can easily track, update, and organize all assets in the system	2	G Deepika
UI ACTION	As a user, I want a button (UI action) to quickly assign or return an asset so that I can perform actions efficiently without navigating through multiple steps.	3	J Gangotri
SCHEDULED JOB	As an admin, I want scheduled jobs to run automatically at set times so that tasks like sending maintenance alerts.	1	G Gopika
REPORT	As a manager, I want to generate reports on asset usage and status so that I can make informed decisions and track performance.	1	G Gopika
TESTING	As a developer, I want to test all features like UI actions and scheduled jobs to ensure they work correctly.	2	G Rama



Note: Request you to please click on "Tick mark ✅" after assigning the activities for each milestone.

Assign Roles & Responsibilities to Team

Table	Create table	×Gouda Deepika	✓ ✖
Table	Create Fields	×Gouda Deepika	✓ ✖
Create UI Actions	UI Action 1	×Jajimoggala Gangotri	✓ ✖
Create UI Actions	UI Action 2	×Jajimoggala Gangotri	✓ ✖
Create UI Actions	UI Action 3	×Jajimoggala Gangotri	✓ ✖
Scheduled Job	Create Scheduled Job	×Ganthi Gopika	✓ ✖
Report	Create Report	×Ganthi Gopika	✓ ✖
Testing	Testing UI action	×Gondu Rama	✓ ✖
Testing	Testing Schduled Job	×Gondu Rama	✓ ✖

Functional and Performance Testing:

MILESTONE 5: TESTING

PURPOSE:

- To verify that the created UI action (button or link) works as expected.
- To ensure the action triggers the correct function without errors.
- To identify and fix bugs before going live.

USE:

- Confirms that actions like "Assign Asset" or "Return Asset" perform correctly.
- Ensures a smooth user experience by preventing broken or faulty buttons.
- Validates that scripts, conditions, and permissions tied to the UI action function properly.
- Improves system reliability and reduces user confusion or errors.

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.

MILESTONE 5: TESTING

PURPOSE:

- To ensure the scheduled job runs automatically at the defined time.
- To verify the job performs the intended task correctly (e.g., send alert, update record).
- To detect and fix errors in automation before production use.

USE:

- Confirms that alerts or reports are generated on time.
- Ensures asset maintenance reminders are sent without failure.
- Validates data updates or status changes happen as scheduled.
- Improves automation reliability and reduces manual oversight.
- Helps maintain consistent and accurate system performance.

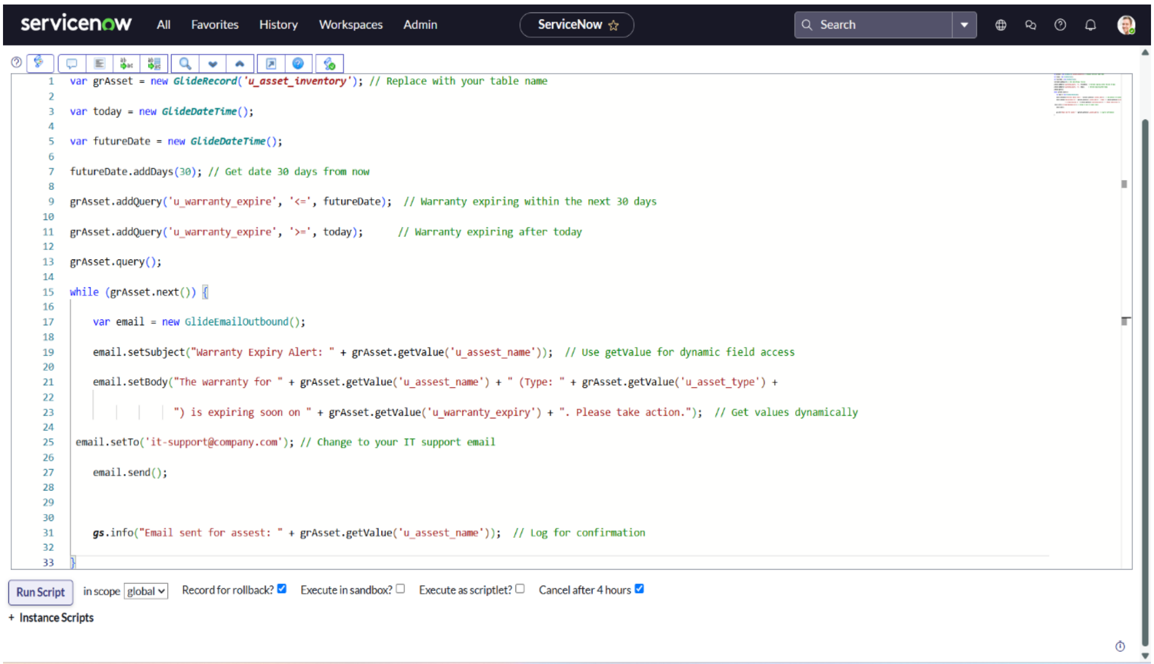
Activity 2: testing scheduled job

STEPS:

5. Navigate to background scripts

6. Write the Scheduled job script in the background scripts

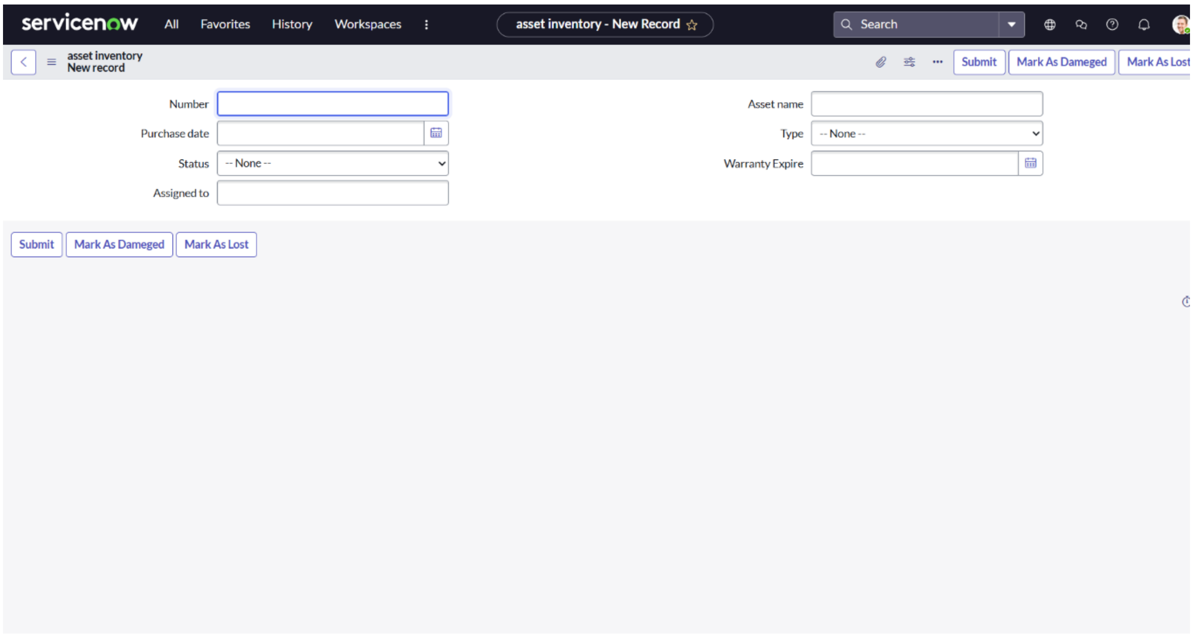
7. Click on Run Script, Check the result



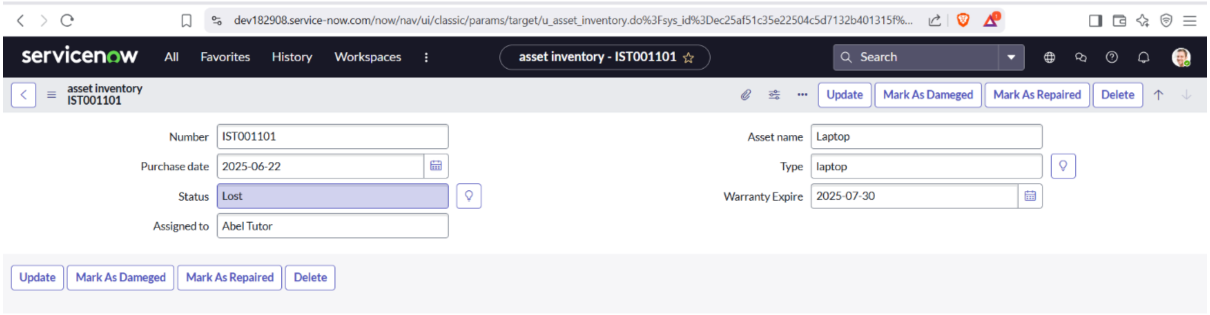
```
var grAsset = new GlideRecord('u_asset_inventory'); // Replace with your table name

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



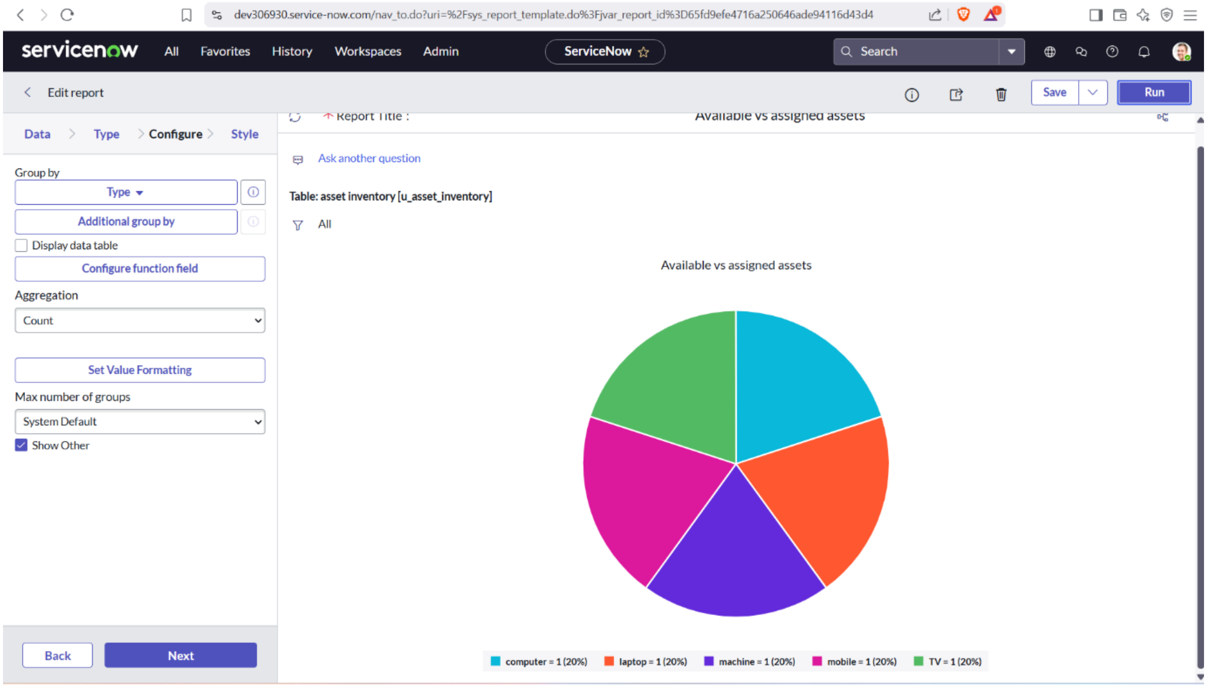
Number
Purchase date
Status -- None --
Assigned to
Asset name
Type -- None --
Warranty Expire


dev182908.service-now.com/now/nav/ui/classic/params/target/u_asset_inventory.do%3Fsys_id%3Dec25af51c35e22504c5d7132b401315f%...
servicenow
All
Favorites
History
Workspaces
asset inventory - IST001101 ☆
Search
asset inventory
IST001101
Update
Mark As Dameged
Mark As Repaired
Delete

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

Update
Mark As Dameged
Mark As Repaired
Delete

RESULTS:

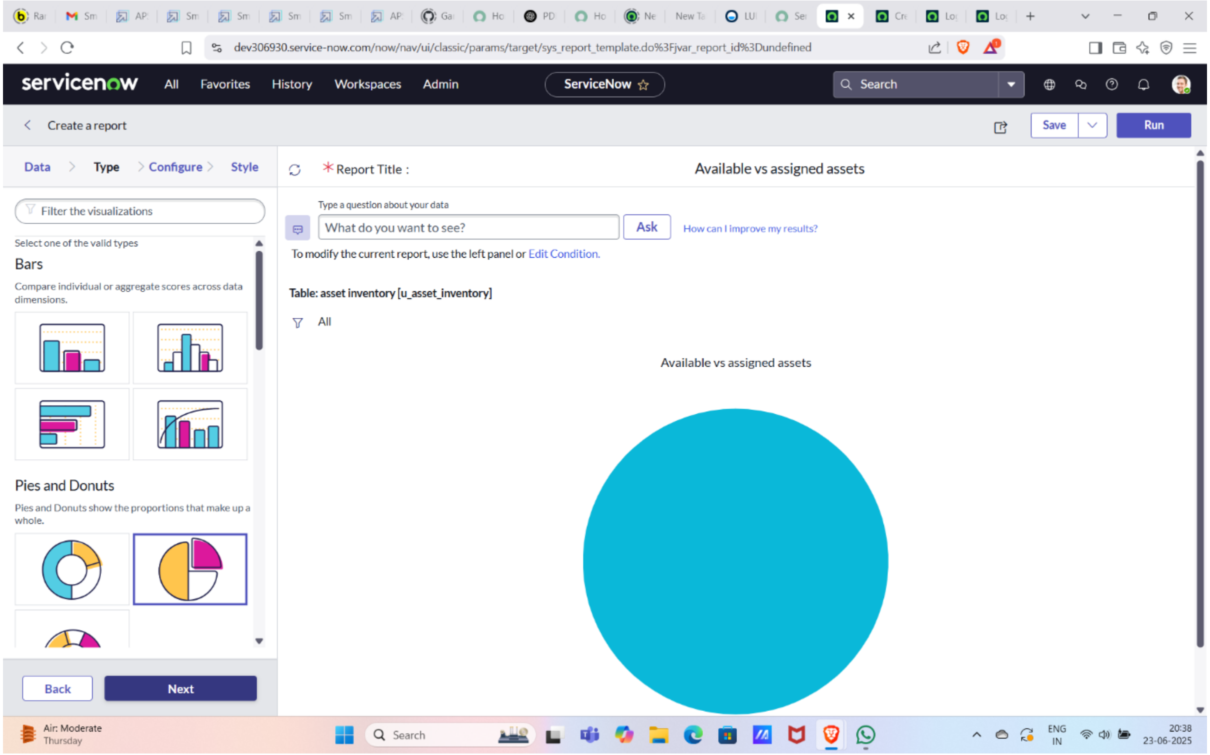

dev306930.service-now.com/nav_to.do?uri=%2Fsys_report_template.do%3Fjvar_report_id%3D65fd9efe4716a250646ade94116d43d4
servicenow
All
Favorites
History
Workspaces
Admin
ServiceNow ☆
Search
Edit report
Save
Run
Data
Type
Configure
Style
Report Title :
Available vs assigned assets
Ask another question

Group by: Type
Additional group by
☐ Display data table
Configure function field
Aggregation: Count
Set Value Formatting
Max number of groups: System Default
☑ Show Other

Table: asset inventory [u_asset_inventory]

All

Available vs assigned assets



Select one of the valid types

Bars

Compare individual or aggregate scores across data dimensions.

Pies and Donuts

Pies and Donuts show the proportions that make up a whole.

Report Title : Available vs assigned assets

Type a question about your data

What do you want to see? Ask

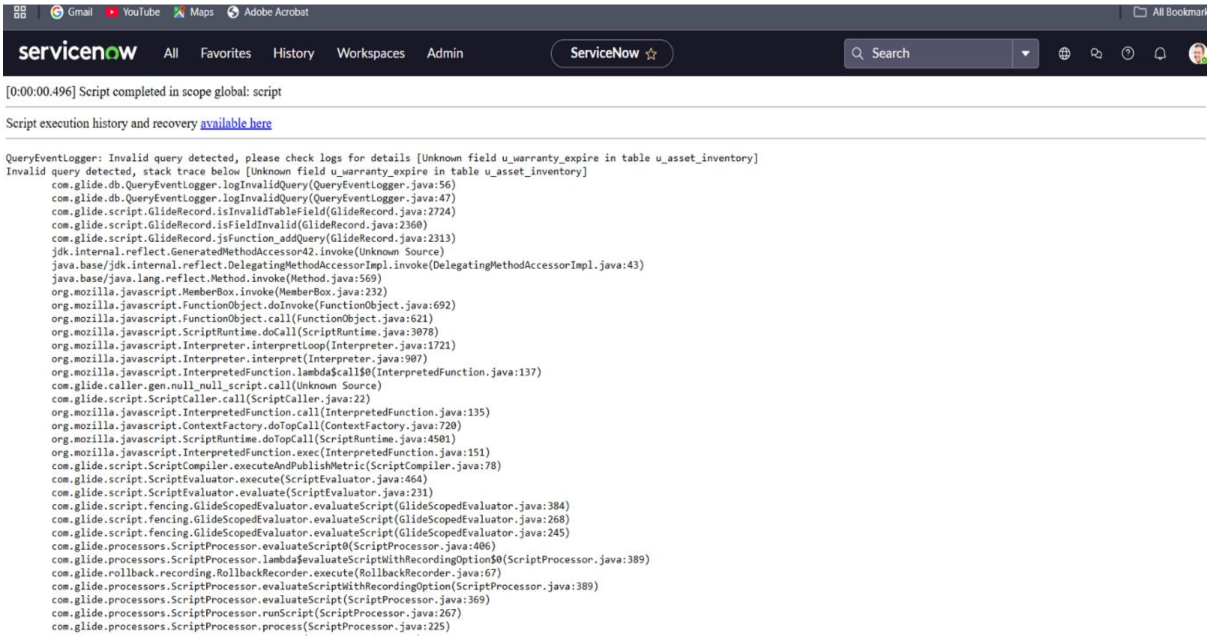
How can I improve my results?

To modify the current report, use the left panel or Edit Condition.

Table: asset inventory [u_asset_inventory]

All

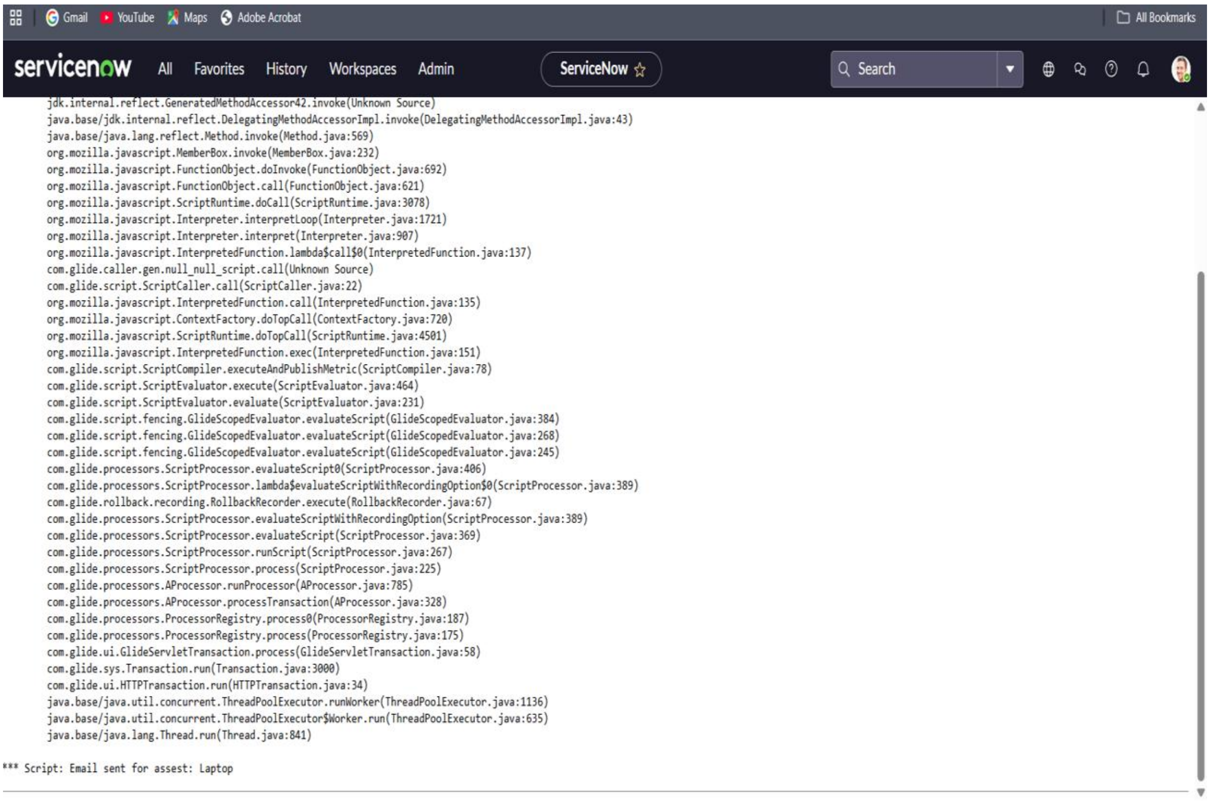
Available vs assigned assets



[0:00:00.496] Script completed in scope global: script

Script execution history and recovery available here

```
QueryEventLogger: Invalid query detected, please check logs for details [Unknown field u_warranty_expire in table u_asset_inventory]
Invalid query detected, stack trace below [Unknown field u_warranty_expire in table u_asset_inventory]
        com.glide.db.QueryEventLogger.logInvalidQuery(QueryEventLogger.java:56)
        com.glide.db.QueryEventLogger.logInvalidQuery(QueryEventLogger.java:47)
        com.glide.script.GlideRecord.isInvalidTableField(GlideRecord.java:2724)
        com.glide.script.GlideRecord.isFieldInvalid(GlideRecord.java:2360)
        com.glide.script.GlideRecord.jsFunction_addQuery(GlideRecord.java:2313)
        jdk.internal.reflect.GeneratedMethodAccessor42.invoke(Unknown Source)
        java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        java.base/java.lang.reflect.Method.invoke(Method.java:569)
        org.mozilla.javascript.MemberBox.invoke(MemberBox.java:232)
        org.mozilla.javascript.FunctionObject.doInvoke(FunctionObject.java:692)
        org.mozilla.javascript.FunctionObject.call(FunctionObject.java:621)
        org.mozilla.javascript.ScriptRuntime.doCall(ScriptRuntime.java:3078)
        org.mozilla.javascript.Interpreter.interpretLoop(Interpreter.java:1721)
        org.mozilla.javascript.Interpreter.interpret(Interpreter.java:907)
        org.mozilla.javascript.InterpretedFunction.lambda$call$0(InterpretedFunction.java:137)
        com.glide.caller.gen.null_null_script.call(Unknown Source)
        com.glide.script.ScriptCaller.call(ScriptCaller.java:22)
        org.mozilla.javascript.InterpretedFunction.call(InterpretedFunction.java:135)
        org.mozilla.javascript.ContextFactory.doTopCall(ContextFactory.java:720)
        org.mozilla.javascript.ScriptRuntime.doTopCall(ScriptRuntime.java:4501)
        org.mozilla.javascript.InterpretedFunction.exec(InterpretedFunction.java:151)
        com.glide.script.ScriptCompiler.executeAndPublishMetric(ScriptCompiler.java:78)
        com.glide.script.ScriptEvaluator.execute(ScriptEvaluator.java:464)
        com.glide.script.ScriptEvaluator.evaluate(ScriptEvaluator.java:231)
        com.glide.script.fencing.GlideScopedEvaluator.evaluateScript(GlideScopedEvaluator.java:384)
        com.glide.script.fencing.GlideScopedEvaluator.evaluateScript(GlideScopedEvaluator.java:268)
        com.glide.script.fencing.GlideScopedEvaluator.evaluateScript(GlideScopedEvaluator.java:245)
        com.glide.processors.ScriptProcessor.evaluateScript0(ScriptProcessor.java:406)
        com.glide.processors.ScriptProcessor.lambda$evaluateScriptWithRecordingOption$0(ScriptProcessor.java:389)
        com.glide.rollback.recording.RollbackRecorder.execute(RollbackRecorder.java:67)
        com.glide.processors.ScriptProcessor.evaluateScriptWithRecordingOption(ScriptProcessor.java:389)
        com.glide.processors.ScriptProcessor.evaluateScript(ScriptProcessor.java:369)
        com.glide.processors.ScriptProcessor.runScript(ScriptProcessor.java:267)
        com.glide.processors.ScriptProcessor.process(ScriptProcessor.java:225)
```



```
        jdk.internal.reflect.GeneratedMethodAccessor42.invoke(Unknown Source)
        java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        java.base/java.lang.reflect.Method.invoke(Method.java:569)
        org.mozilla.javascript.MemberBox.invoke(MemberBox.java:232)
        org.mozilla.javascript.FunctionObject.doInvoke(FunctionObject.java:692)
        org.mozilla.javascript.FunctionObject.call(FunctionObject.java:621)
        org.mozilla.javascript.ScriptRuntime.doCall(ScriptRuntime.java:3078)
        org.mozilla.javascript.Interpreter.interpretLoop(Interpreter.java:1721)
        org.mozilla.javascript.Interpreter.interpret(Interpreter.java:907)
        org.mozilla.javascript.InterpretedFunction.lambda$call$0(InterpretedFunction.java:137)
        com.glide.caller.gen.null_null_script.call(Unknown Source)
        com.glide.script.ScriptCaller.call(ScriptCaller.java:22)
        org.mozilla.javascript.InterpretedFunction.call(InterpretedFunction.java:135)
        org.mozilla.javascript.ContextFactory.doTopCall(ContextFactory.java:720)
        org.mozilla.javascript.ScriptRuntime.doTopCall(ScriptRuntime.java:4501)
        org.mozilla.javascript.InterpretedFunction.exec(InterpretedFunction.java:151)
        com.glide.script.ScriptCompiler.executeAndPublishMetric(ScriptCompiler.java:78)
        com.glide.script.ScriptEvaluator.execute(ScriptEvaluator.java:464)
        com.glide.script.ScriptEvaluator.evaluate(ScriptEvaluator.java:231)
        com.glide.script.fencing.GlideScopedEvaluator.evaluateScript(GlideScopedEvaluator.java:384)
        com.glide.script.fencing.GlideScopedEvaluator.evaluateScript(GlideScopedEvaluator.java:268)
        com.glide.script.fencing.GlideScopedEvaluator.evaluateScript(GlideScopedEvaluator.java:245)
        com.glide.processors.ScriptProcessor.evaluateScript0(ScriptProcessor.java:406)
        com.glide.processors.ScriptProcessor.lambda$evaluateScriptWithRecordingOption$0(ScriptProcessor.java:389)
        com.glide.rollback.recording.RollbackRecorder.execute(RollbackRecorder.java:67)
        com.glide.processors.ScriptProcessor.evaluateScriptWithRecordingOption(ScriptProcessor.java:389)
        com.glide.processors.ScriptProcessor.evaluateScript(ScriptProcessor.java:369)
        com.glide.processors.ScriptProcessor.runScript(ScriptProcessor.java:267)
        com.glide.processors.ScriptProcessor.process(ScriptProcessor.java:225)
        com.glide.processors.AProcessor.runProcessor(AProcessor.java:785)
        com.glide.processors.AProcessor.processTransaction(AProcessor.java:328)
        com.glide.processors.ProcessorRegistry.process0(ProcessorRegistry.java:187)
        com.glide.processors.ProcessorRegistry.process(ProcessorRegistry.java:175)
        com.glide.ui.GlideServletTransaction.process(GlideServletTransaction.java:58)
        com.glide.sys.Transaction.run(Transaction.java:3000)
        com.glide.ui.HTTPTransaction.run(HTTPTransaction.java:34)
        java.base/java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1136)
        java.base/java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:635)
        java.base/java.lang.Thread.run(Thread.java:841)

*** Script: Email sent for assest: Laptop
```

Advantages and Disadvantages:

ADVANTAGES:

- Centralized Asset Management: ServiceNow provides a unified platform to manage all physical and digital assets in one place.
- Automation & Workflow Integration: Automates asset requests, approvals, assignments, and maintenance using built-in workflows.
- Real-Time Visibility: Tracks asset status, location, and usage in real time, improving accountability and planning
- Integration with Other Modules: Easily integrates with ITSM, CMDB, Procurement, and other ServiceNow modules for seamless operations.
- User-Friendly Interface: Offers a customizable and intuitive UI for both end-users and administrators.

- Scalable & Cloud-Based: Easily scales with organization size and needs, with cloud accessibility ensuring global reach.
- Security & Role-Based Access: Strong access controls and data security features protect sensitive asset data.

DISADVANTAGES:

- High Licensing Costs: ServiceNow is a premium platform, and asset management modules may involve additional licensing fees.
- Complex Setup & Customization: Initial configuration and customization require technical expertise and time.
- Training Required: Users and admins may need training to fully utilize features and workflows.
- Performance Dependency on Configuration: Poorly designed workflows or excessive customization may affect system performance.
- Ongoing Maintenance: Requires regular updates, testing, and support to keep workflows and integrations running smoothly.

Conclusion:

The Asset Management Portal provides a comprehensive solution for tracking, managing, and optimizing physical and digital assets throughout their lifecycle. By leveraging automation and real-time data updates, the platform ensures efficient asset allocation, minimizes discrepancies, and enhances operational visibility. Automated workflows for asset tracking, maintenance alerts, and reporting enable organizations to make data-driven decisions, reduce asset downtime, and optimize resource utilization. This project demonstrates the power of ServiceNow's capabilities in integrating asset tracking, automation, and reporting tools to create a streamlined asset management system. By improving asset accountability and operational efficiency, the platform helps organizations maximize asset value, reduce costs, and enhance overall productivity.

The Asset Management Portal project in ServiceNow offers a powerful, automated, and centralized solution for managing an organization's physical and digital assets. By leveraging ServiceNow's robust platform, the portal enhances visibility, streamlines asset allocation, automates lifecycle management, and reduces manual errors. Its seamless integration with other ServiceNow modules like ITSM and CMDB adds tremendous value, making operations more efficient and data-driven.

The project not only simplifies asset tracking and maintenance but also supports scalability, security, and user-friendly interaction across departments. Despite initial setup costs and the need for training, the long-term benefits—including better resource planning, cost savings, audit readiness, and improved user satisfaction—make this solution a strategic asset for any organization. In conclusion, implementing the Asset Management Portal on ServiceNow empowers organizations to modernize asset operations and drive digital transformation effectively.