

Calculating Family Expenses using ServiceNow

Team ID: LTVIP2026TMIDS35551

Team Size: 4

Team Leader: Devagupthapu Srinu

Team Member: Nakka Mahalakshmi Surya Nandini

Team Member: Ramya Peluri

Team Member: Sabbavarapu Bhimesh Prasad

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.

Objective:

The main objective of this project is to design and develop an Asset Management Portal using ServiceNow that will:

1. Provide a centralized platform to manage all organizational assets.
2. Allow employees to request assets easily through a portal.
3. Enable administrators to track asset allocation and availability.
4. Automate asset assignment, maintenance alerts, and lifecycle tracking.
5. Generate reports and dashboards for asset utilization and performance.
6. Improve asset visibility, reduce asset loss, and increase operational efficiency.

Skills:

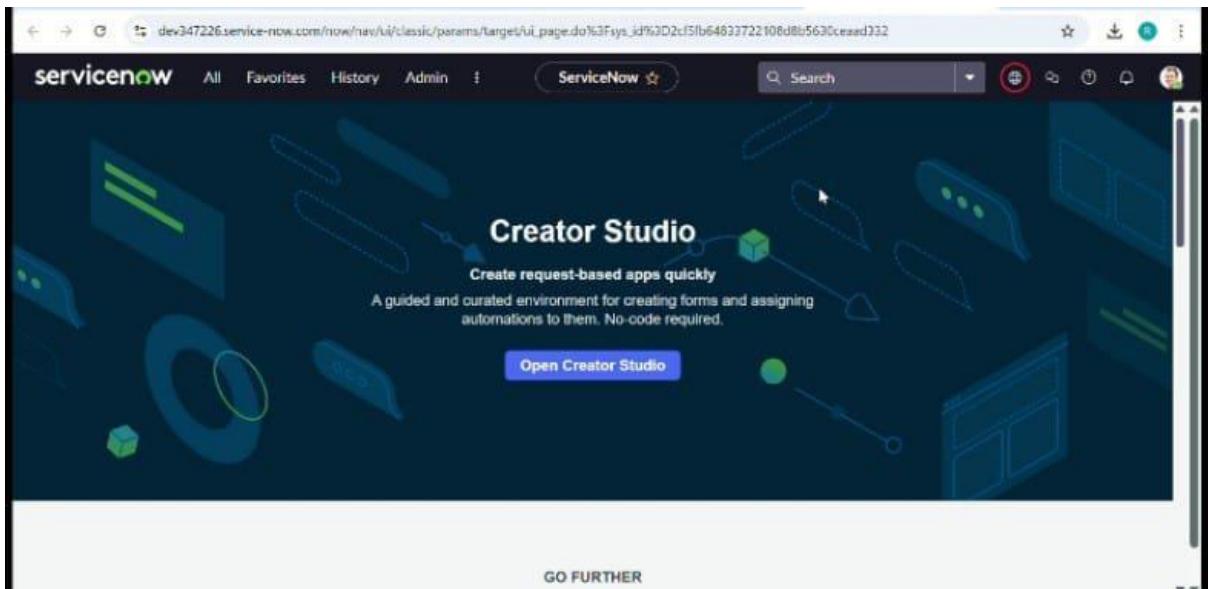
- ServiceNow App Engine Studio
- Table Creation & Data Modeling
- Form Design & UI Customization
- Flow Designer (Automation)
- Business Rules & Scripting
- Reporting & Dashboards
- Access Control Lists (ACLs)
- Service Portal Development

TASK INITIATION

Milestone 1 : Instance

Activity 1: Setting up ServiceNow instance

1. Sign up for a developer account on the ServiceNow Developer site "<https://developer.servicenow.com>".
2. Once logged in, navigate to the "Personal Developer Instance" section.
3. Click on "Request Instance" to create a new ServiceNow instance.
4. Fill out the required information and submit the request.
5. You'll receive an email with the instance details once it's ready.
6. Log in to your ServiceNow instance using the provided credentials.
7. Now you will navigate to the ServiceNow.



Milestone 2: Creation of Asset Request Table

Activity 1: Create Asset Request Table

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
6. Save the table

The screenshot shows the ServiceNow 'Tables' list page. On the left, there is a navigation sidebar with a search bar containing '7 tables'. The main content area displays a table with columns: Label, Name, Extends table, Extensible, and Updated. The table lists various system tables, such as 'adaptive_auth_event', 'agent_assist_recommendation', and 'aisa_rp_config'. A red box highlights the 'New' button at the top right of the table header.

The screenshot shows the 'Table - Asset Inventory' edit screen. The top navigation bar includes 'Favorites', 'History', 'Workspaces', 'Tables', 'Search', and a 'New' button. The main area shows the table configuration with fields for 'Label' (Asset Inventory) and 'Name' (u.asset_inventory). A context menu is open on the right side, with 'Save' highlighted. The menu also includes options like 'Analyze Access', 'Show File Properties', 'Move to Application...', 'Show Latest Update', 'Show Dictionary Record', 'Configure', 'Export', 'View', 'Create Favorite', 'Copy URL', 'Copy sys_id', 'Show XML', 'History', and 'Reload form'. The bottom part of the screen shows the table's dictionary entries with columns: Column label, Type, Reference, Max length, Default value, and Display.

Activity 2: Create Fields for Asset Request Table

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 managing asset inventory. On the left, the navigation bar has 'asset' selected. The main area displays a table of dictionary entries for the 'Asset Inventory' table. A context menu is open over the last column of the table, specifically over the 'Created' entry. The menu items include 'Save', 'Analyze Access', 'Show File Properties', 'Move to Application...', 'Show Latest Update', 'Show Dictionary Record', 'Configure', 'Export', 'View', 'Create Favorite', 'Copy URL', 'Copy sys_id', 'Show XML History', and 'Reload form'. The 'Save' option is highlighted with a red box.

Milestone 3: UI Action 1

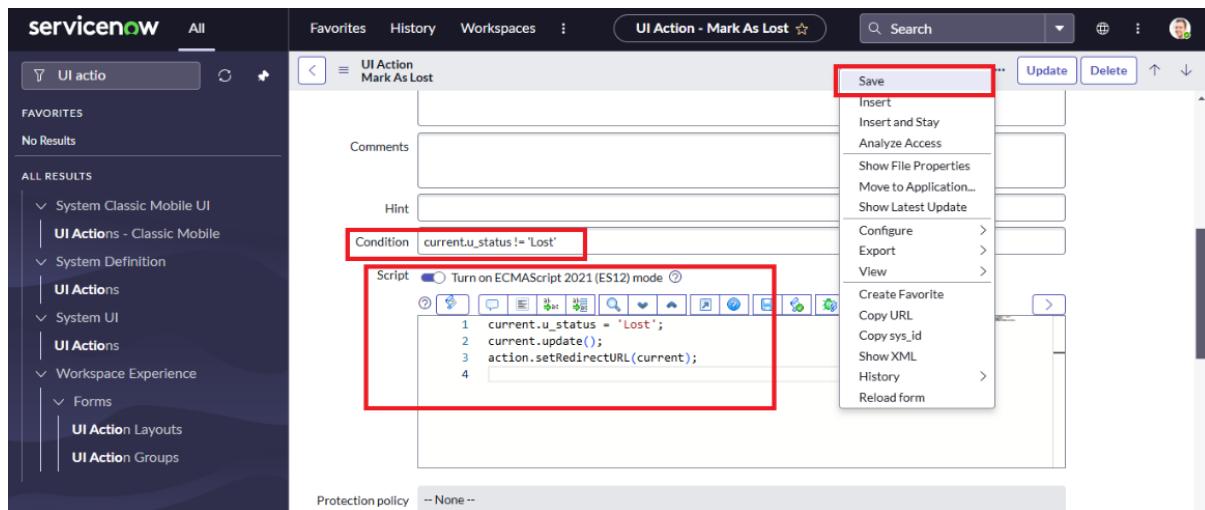
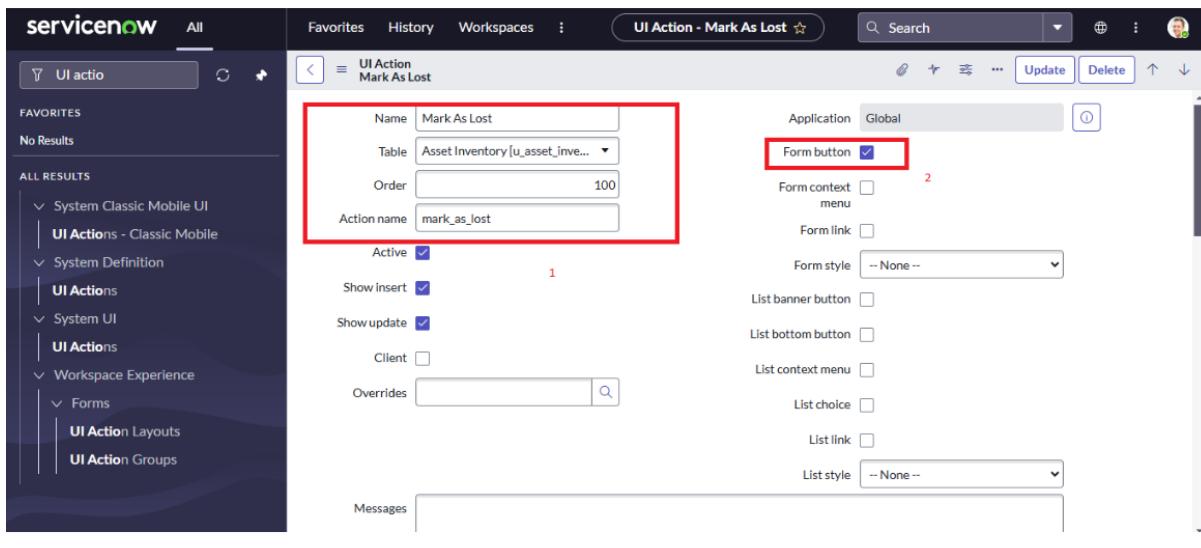
Activity 1: Creating UI Action-1

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 interface for managing UI actions. The left sidebar has 'UI Actions' selected under 'System Definition'. The main area shows a list of existing UI actions. A new UI action is being created, indicated by the 'New' button at the top right of the list table. The 'UI Actions' search bar in the top navigation is also highlighted with a red box.

Name	Table	Comments	Form action	List action	Active	Order
Deploy	SA RCA SMC Config [sa_rca_smc_config]		true	false	true	
View in Workspace	Project Definition Version [promin_model_def_version]		true	false	true	
Save	Template [sys_template]	Updates an existing record and redirects back to self (button version, advanced mode) in Agent Workspace	false	false	true	-10
Save	Template [sys_template]	Saves a new record and redirects back to self (button version, advanced mode) in Agent Workspace	false	false	true	-10
Discover Application	Application Suggestion [cmdb_process_groups]	Create Discovery for Suggested Application	true	false	true	-10
Delete	GlobalDelete	Deletes current record after confirmation for	true	false	true	-1



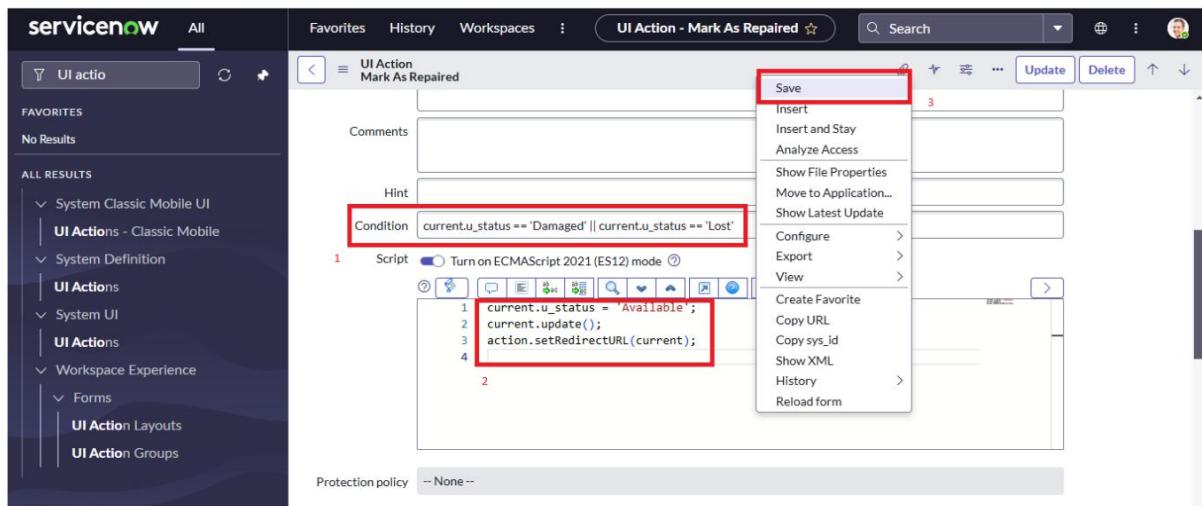
Column label	Type	Reference	Max length	Default value	Display
Updated	Date/Time	(empty)	40	false	
Created by	String	(empty)	40	false	
Sys ID	Sys ID (GUID)	(empty)	32	false	
Created	Date/Time	(empty)	40	false	
Updated by	String	(empty)	40	false	
Updates	Integer	(empty)	40	false	
Number	String			false	
Date	Date			false	
Amount	Integer			false	
Expense Details	String		800		

Activity 2: Creating UI Action-2

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



Activity 3: Creating UI Action-3

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

The screenshot shows the ServiceNow interface for creating a UI Action named "Mark As Damaged". The left sidebar lists categories like System Classic Mobile UI, UI Actions - Classic Mobile, and UI Actions. The main area displays the UI Action configuration with a "Script" section containing the following code:

```

Condition: current.u_status != 'Damaged' 1
Script: Turn on ECMAScript 2021 (ES12) mode ②
1 current.u_status = 'Damaged';
2 current.update();
3 action.setRedirectURL(current);
4

```

A red box highlights the "Save" option in the context menu, which is also labeled with a red "3". Other options in the menu include Insert, Insert and Stay, Analyze Access, Show File Properties, Move to Application..., Show Latest Update, Configure, Export, View, Create Favorite, Copy URL, Copy sys_id, Show XML, History, and Reload form.

Milestone 4: Create Scheduled Job

Activity 1: Creating Scheduled Job

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

The screenshot shows the ServiceNow interface for creating a Scheduled Script Execution named "Warranty Expiry Alert". The left sidebar lists categories like Self-Service, Visual Task Boards, and My Connected Apps. The main area displays the script configuration with a "Run this script" section containing the following code:

```

Run this script Turn on ECMAScript 2021 (ES12) mode ②
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 grAsset.addQuery('u_warranty_expire', '<=', futureDate); // Warranty
expiring within the next 30 days
7 grAsset.addQuery('u_warranty_expire', '>', today); // Warranty
expiring after today
8 grAsset.query();
9
10 while (grAsset.next()) {
11   var email = new GlideEmailOutbound();
12   email.setSubject("Warranty Expiry Alert: " + grAsset.getValue
('u_asset_name')); // Use getValue for dynamic field access
13   email.setBody("The warranty for " + grAsset.getValue
('u_asset_name') + " (Type: " + grAsset.getValue('u_asset_type') +
" " + grAsset.getValue('u_warranty_expiry')) is expiring soon. " + grAsset.getValue
('u_warranty_expiry') + ". Please take action."); //

```

SCRIPT:

```
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
}
```

Milestone 5: Create Report

Activity 1: Creation of Report

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
 - Column label : Comments
 - Type : String
 - Max length : 800
11. Go to the Header and right click there>> click on Save.

The screenshot shows the ServiceNow 'Edit report' configuration page for the 'Available vs. Assigned Assets' report. The left sidebar shows the navigation menu under 'Reports'. The main area displays the report configuration steps:

- Data**: Set 'Report name' to 'Available vs. Assigned Assets' and 'Source type' to 'Table' (Asset Inventory [u_asset_inventory]).
- Type**: Set 'Group by' to 'Status' and 'Aggregation' to 'Count'.
- Configure**: Step 1 is completed.
- Style**: Step 2 is completed.

The right side shows a pie chart titled 'Available vs. Assigned Assets' with two segments: orange (representing Available assets) and blue (representing Assigned assets).

This screenshot is similar to the previous one but shows the configuration process further along:

- Data**: Set 'Report name' to 'Available vs. Assigned Assets' and 'Source type' to 'Table' (Asset Inventory [u_asset_inventory]).
- Type**: Set 'Group by' to 'Status' and 'Aggregation' to 'Count'.
- Configure**: Step 1 is completed.
- Style**: Step 2 is completed.

The top right toolbar has 'Save' and 'Run' buttons highlighted with red boxes. The right side shows the same pie chart.

Milestone 6: Testing UI action

Activity 1: Testing UI action

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 Create screen. A new record is being created for an asset named 'Laptop'. The asset number is 'AST001013'. The status is currently set to 'Available'. The 'Submit' button at the bottom left is highlighted with a red box. Other buttons like 'Mark As Damaged' and 'Mark As Lost' are also visible.

Milestone 7: Testing Scheduled Job

Activity 1: Testing Scheduled Job

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 shows the ServiceNow Scripts - Background screen. A scheduled job script is displayed, titled 'Run script (JavaScript executed on server)'. The script code is as follows:

```

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_assest_name')); // Use getValue
15     email.setBody("The warranty for " + grAsset.getValue('u_assest_name') + " (Type: " + grAsset.
16         getValue('u_asset_type') +
17             ") is expiring soon on " + grAsset.getValue('u_warranty_expiry') + ". Please take
18             action."); // Get values dynamically
19

```

servicenow All

Favorites History Workspaces Admin ServiceNow ☆ Search

FAVORITES No Results

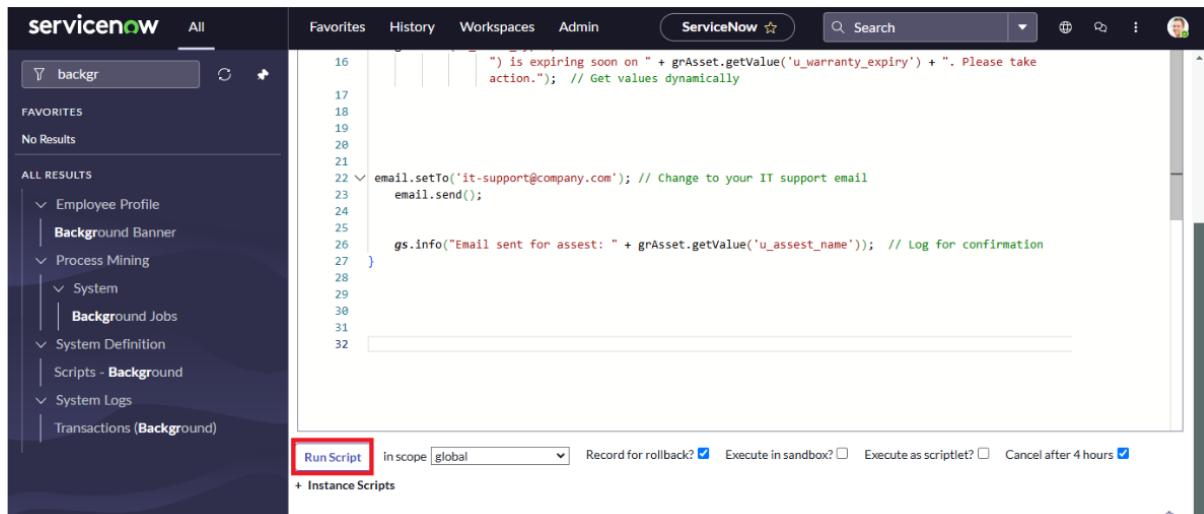
ALL RESULTS

- Employee Profile
 - Background Banner
- Process Mining
 - System
 - Background Jobs
- System Definition
 - Scripts - Background
- System Logs
 - Transactions (Background)

16 " is expiring soon on " + grAsset.getValue('u_warranty_expiry') + ". Please take
17 action."); // Get values dynamically
18
19
20
21
22 email.setTo('it-support@company.com'); // Change to your IT support email
23 email.send();
24
25
26 gs.info("Email sent for asset: " + grAsset.getValue('u_assest_name')) // Log for confirmation
27 }
28
29
30
31
32

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

+ Instance Scripts



servicenow All

Favorites History Workspaces Admin ServiceNow ☆ Search

FAVORITES No Results

ALL RESULTS

- Employee Profile
 - Background Banner
- Process Mining
 - System
 - Background Jobs
- System Definition
 - Scripts - Background
- System Logs
 - Transactions (Background)

[0:00:00.252] Script completed in scope global: script
Script execution history and recovery [available here](#)

*** Script: Email sent for asset: Mobile
*** Script: Email sent for asset: Laptop 1

