

Date	16 Feb 2026
Team ID	LTVIP2026TMIDS36006
Project Name	Asset Management Portal using Service Now
Mentor Name	Dr Shaik Salma Begum
Maximum Marks	10 Marks

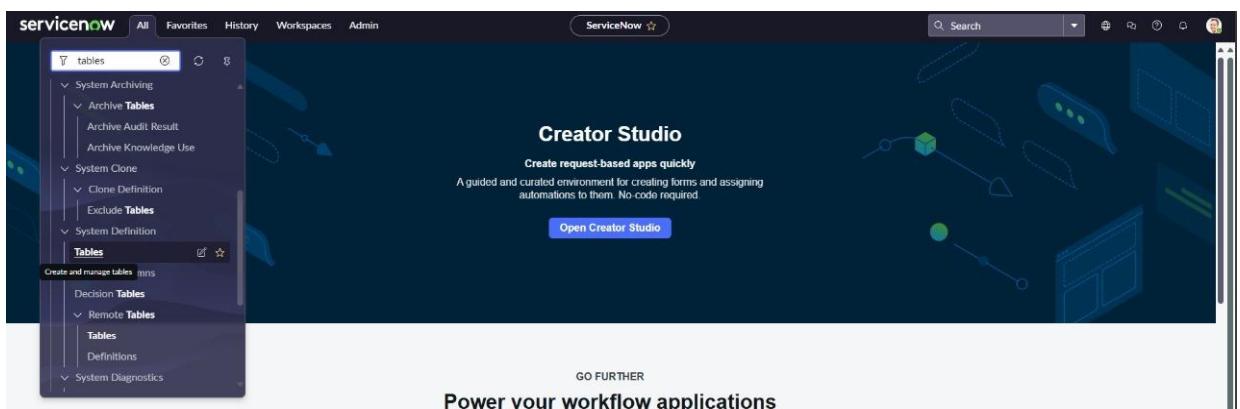
Asset Management Portal

Milestone 1: Setting Up ServiceNow Instance:

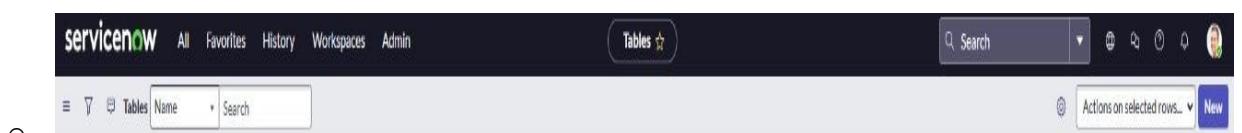
- Sign up on developer.servicenow.com and create a developer account.
- Navigate to the Personal Developer Instance section and request a new instance.
- Fill out the required information and submit the request.
- Instance details (URL, username, password) will be sent via email.
- Log in to the instance using the provided credentials.
- The instance is ready for development and customization.

Milestone 2: Creation of Table (Asset Inventory):

- Navigate to All in the application navigator.
- In the filter search bar, type Tables and select it from the results.

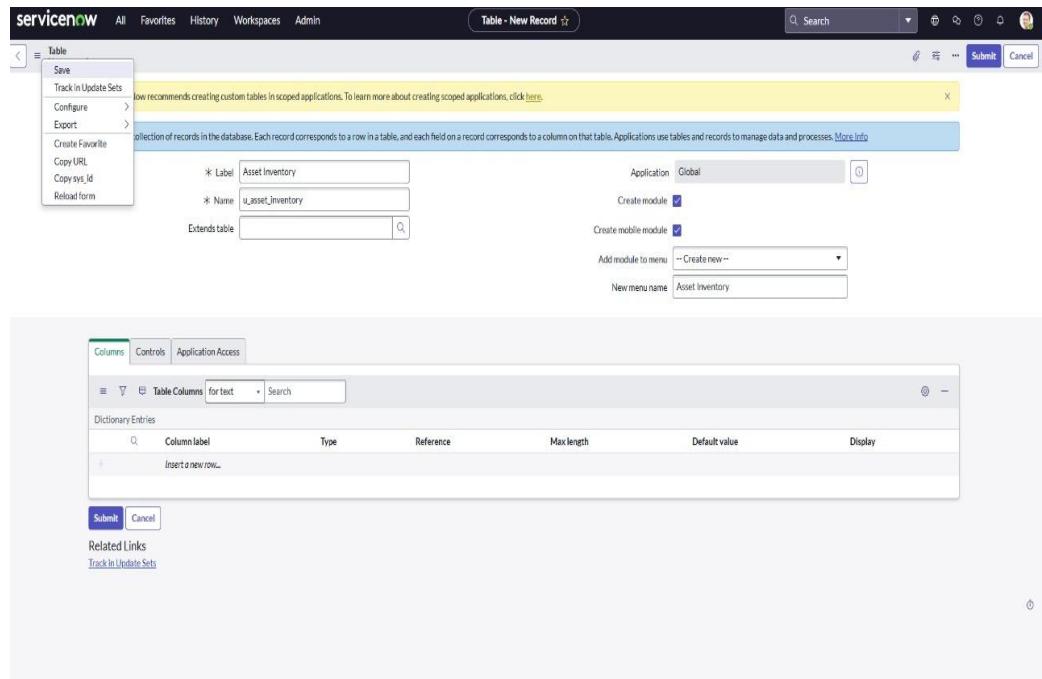


- Click on the New button to create a new table.



- In the form, provide the following details:

- Label: Asset Inventory
- Name: Auto-Populated (automatically generated based on the label)
- New menu name: Asset Management Portal



- Navigate to the header section of the form, right-click on the header, and select Save from the context menu.

Milestone 3: Creation of Fields:

1) After saving the table scroll down

2) Create the following fields

- Assigned to : string
- Status : choice
- Purchase date : date
- Warranty Expire : date
- Asset name : string
- Type : choice
- Number : String

3) Click on save

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

Milestone 4: Creation of UI Actions:

- Navigate to UI Actions under System Definition:

The screenshot shows the ServiceNow interface for managing UI Actions. The left sidebar has a search bar for 'UI action' and a tree view of categories like 'System Classic Mobile UI', 'System Definition', and 'System UI'. The main area is titled 'Table - Asset Inventory' and shows a list of UI actions for the 'Asset Inventory' table. The first UI action listed is named 'mark_as_lost'.

Type	Reference	Max length	Default value	Display
Choice	(empty)	40	false	false
Choice	(empty)	40	false	false
Text/Item	(empty)	40	false	false

- Click on new to create a new UI Action
- Fill in the UI Action Details ;
- Name: Mark as Lost
- Table: Asset Inventory
- Action name: mark_as_lost
- Condition: current.u_status !='Lost'
- Update the script as
 - current.u_status = 'Lost';

```
current.update();
action.setRedirectURL(current);
```
- Check the box Form Button to make it visible on the form layout.
- Click Save to apply the changes.

The screenshot shows the 'UI Action - New Record' configuration page. The 'Name' field is set to 'Mark As Lost', 'Table' is 'Asset Inventory [u_asset_inventory]', and 'Action name' is 'mark_as_lost'. The 'Condition' field contains the script: 'current.u_status != "Lost"'. The 'Script' section shows the following code:

```
Turn on ECMAScript 2021 (ES12) mode. ☐
1 current.u_status = "Lost";
2 current.update();
3 action.setRedirectURL(current);
```

- Navigate to Application Navigator
 - Click on UI Actions under System Definition.
 - Click on New(top-right corner) to create new UI Action
 - Fill in the following details:
 - Name: Mark as Repaired
 - Table: Asset Inventory
 - Action name: mark_as_repaired
 - Condition: current.u_status=='Damaged'
 - Add the Script as:
 - current.u_status = 'Available';
current.update();
action.setRedirectURL(current);
 - This Script updates the u_status field to Available saves the record, and then redirects the user back to the current record view.
 - Scroll down to the Form button section.
 - Check the box labeled Form button to ensure this action appears as a button on the form view.
 - Click on Submit to create the UI Action.

The screenshot shows the 'UI Action - New Record' configuration page in ServiceNow. The 'Name' field is set to 'Mark As Repaired', 'Table' is 'Asset Inventory [u_asset_inventory]', 'Action name' is 'mark_as_repaired', and 'Condition' is 'current.u_status == "Damaged" || current.u_status == "Lost"'. The 'Script' section contains the following ECMAScript 2021 (ES12) code:

```

1 current.u_status = "Available";
2 current.update();
3 action.setRedirectURL(current);

```

The 'Form style' dropdown is set to 'None'. The 'Protection policy' dropdown is set to 'None'.

Milestone 5: Creation of Scheduled Job:

- Navigate to All in the application navigator.
- In the filter search bar, type Scheduled Job and select it from the list.
- Click on the New button to create a new scheduled job.

- Write the following script:
- Enter the following details:
 - Name: Warranty Expiry Alert
 - Run : Daily
 - Time : 12:00

Write the following 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
}

```

- Click Save to store the scheduled job.

The screenshot shows the 'Scheduled Script Execution - New Record' page in ServiceNow. The script is titled 'Warranty Expiry Alert'. It is set to run daily at 12:00. The script code is as follows:

```

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_expire', '<', futuredate); // Warranty expiring within the next 30 days
10 grasset.addQuery('u_warranty_expire', '>', today); // Warranty expiring after today
11
12 grasset.query();
13
14 while (grasset.next()) {
15
16     var email = new GlideEmailOutbound();
17
18     email.setSubject("Warranty Expiry Alert: " + grasset.getValue('u_asset_name')); // Use getValue for
19     dynamic field access
20
21     email.setBody("The warranty for " + grasset.getValue('u_asset_name') + " (" + Type: " + grasset.getValue(
22         'u_asset_type') + "

```

Milestone 6: Creation of Report :

- Navigate to All in the main menu.
- In the filter/search bar, type Reports, then select and open the Reports.
- Click on the New button to create a new Report.
- Enter the following details:
 - Report Name: Available vs assigned assets
 - Source Type: Table
 - Table: Asset Inventory
 - Type: Pie chart
 - Group By: Status
 - Aggregation: Count.

The screenshot shows the 'Create a report' page in ServiceNow. The report is titled 'Available vs assigned assets'. The source table is 'Asset Inventory [u_asset_inventory]'. The report type is 'Table' and the aggregation is 'Count'. The 'Report Title' field is empty. There is a section for 'Create your report with Analytics Q&A' with a text input field 'What do you want to see?' and a button 'Ask'.

servicenow All Favorites History Workspaces Admin Create a report Save Run

*** 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

Number	Asset name	Assigned to	Purchase date	Status	Type	Warranty Expire
1	Laptop	Abel Tutor	2025-06-01	Damaged		2025-07-01
2	Mobile	Abraham Lincoln	2025-06-00			2025-06-00
3	Camera	Adela	2025-07-03			2025-12-26

Available vs assigned assets

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

Time series
Visualize data over time.

Back Next

servicenow All Favorites History Workspaces Admin Create a report Save Run

*** 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

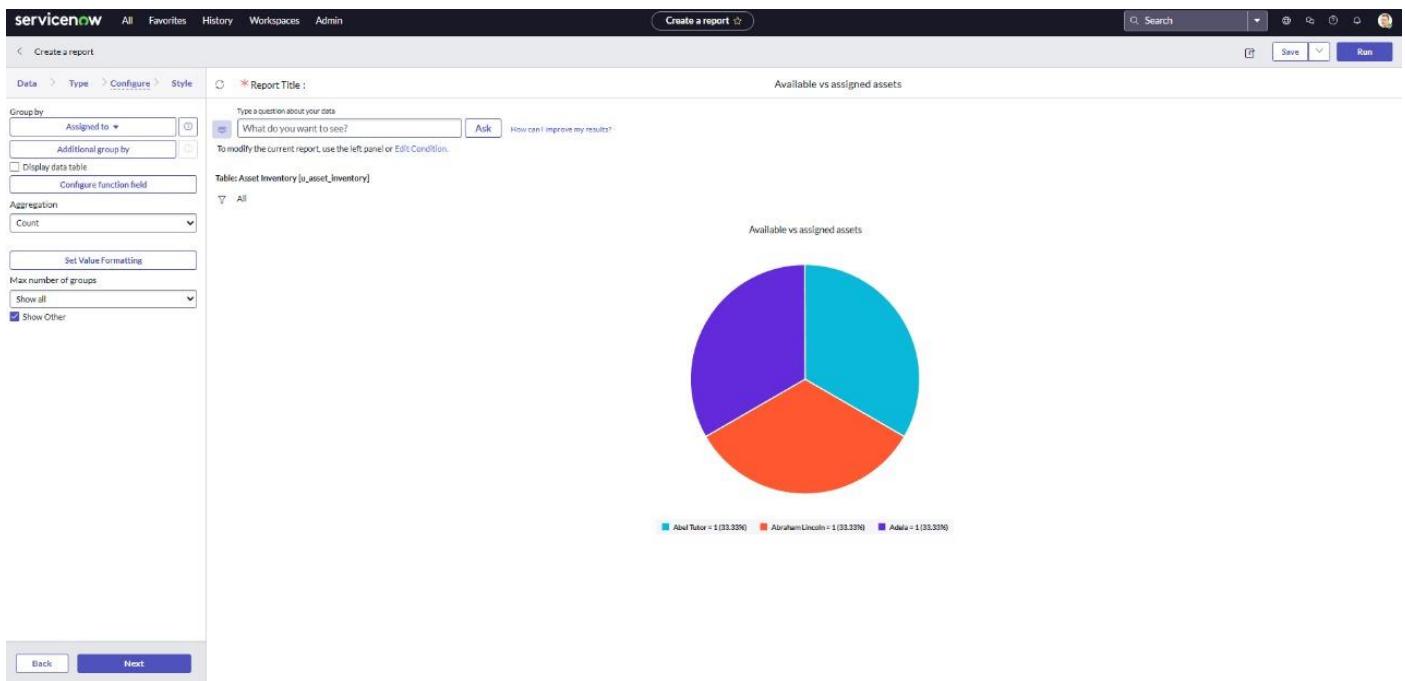
(empty) = 2 (66.67%) Damaged = 1 (33.33%)

Group by Status Additional group by Display data table Configure function field

Aggregation Count Set Value Formatting Max number of groups Show all Show Other

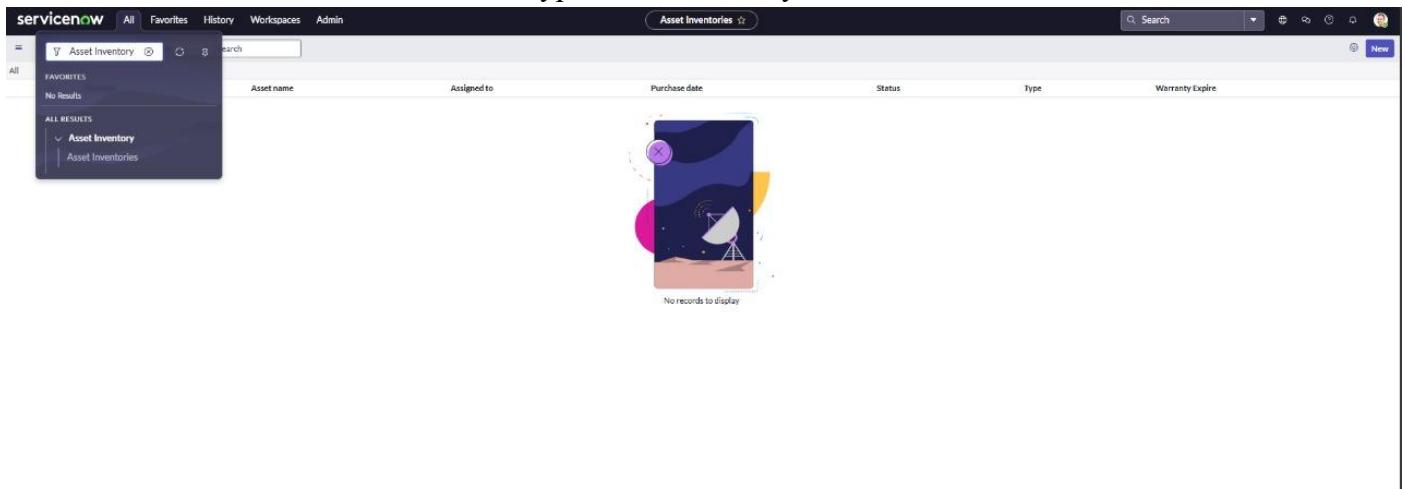
Back Next

- Click Save or Done (depending on the interface) to apply the changes.
- Next click on run.



Milestone 7: Testing of UI Action:

- From the main navigation, go to All.
- In the search or filter bar, type Asset Inventory table.



- Click on the New button to test UI Action.
- Fill in the Asset inventory table record Details:
 - Asset Name:Laptop
 - Type: laptop
 - Assigned to: Abel Tutor
 - Status: Available
 - Purchase date: 2025-06-01
 - Expiry date: 2025-07-01

- Click on Submit.
- Open the record once again and click on the mark as lost button .
- Save the record.
- Check the status is changed to lost.

	Number	Asset name	Assigned to	Purchase date	Status	Type	Warranty Expire
1	Laptop	Abel Tutor	2025-06-01	Damaged	2025-07-01		
2	Mobile	Abraham Lincoln	2025-06-03	2025-08-03			
3	Camera	Adela	2025-07-03	2025-12-26			

Milestone 8: Testing of Scheduled Job:

- From the main navigation, go to All.
- In the search bar, type background scripts, then open the Scheduled Job script in the background scripts.

- Click on run script button to run the code.

```

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 GlideEmail();
    email.setSubject("Warranty expiry alert: " + grasset.getValue('u_asset_name')); // Use getvalue for dynamic field access
    email.setBody("The warranty for " + grasset.getValue('u_asset_name') + " (" + 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 asset: " + grasset.getValue('u_asset_name')) // Log for confirmation

```

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

- After running the script, check the result.

[0 00:00:368] Script completed in scope global: aascript

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

*** Script: (email sent for asset): null