PROJECT DESIGN

TEAM ID	LTVIP2025TMID30624
PROJECT NAME	Asset Management Portal

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:

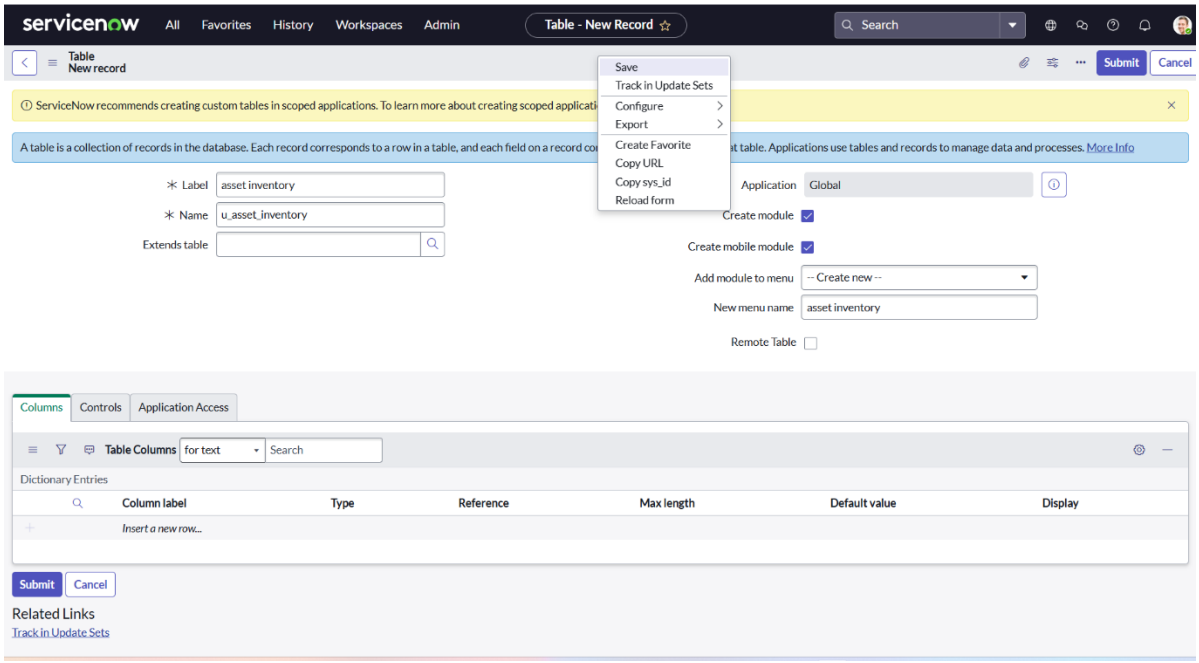
- Stores details of each asset in rows (records) and columns (fields).
- Enables easy tracking and retrieval of asset information.
- Supports workflows like asset assignment, return, maintenance, etc.
- Acts as the data source for dashboards, reports, and UI views.
- Helps maintain data consistency,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



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

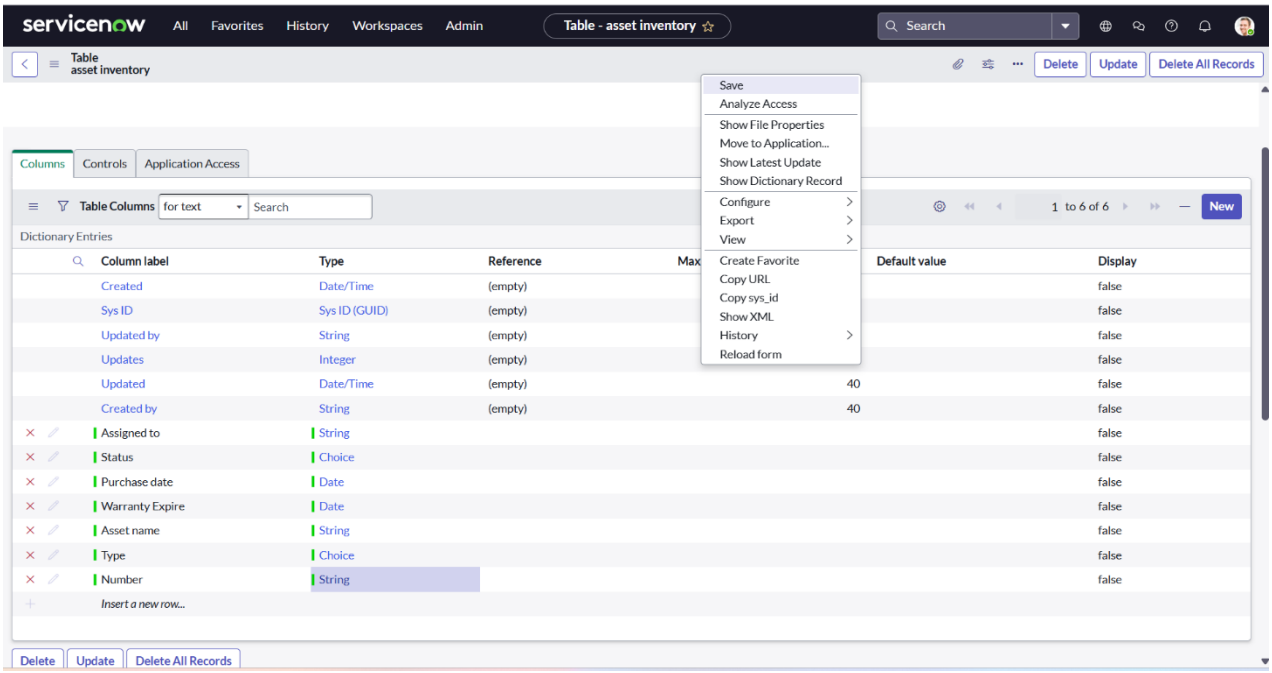
2)Create fields

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

MILESTONE 2: UI ACTION

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:

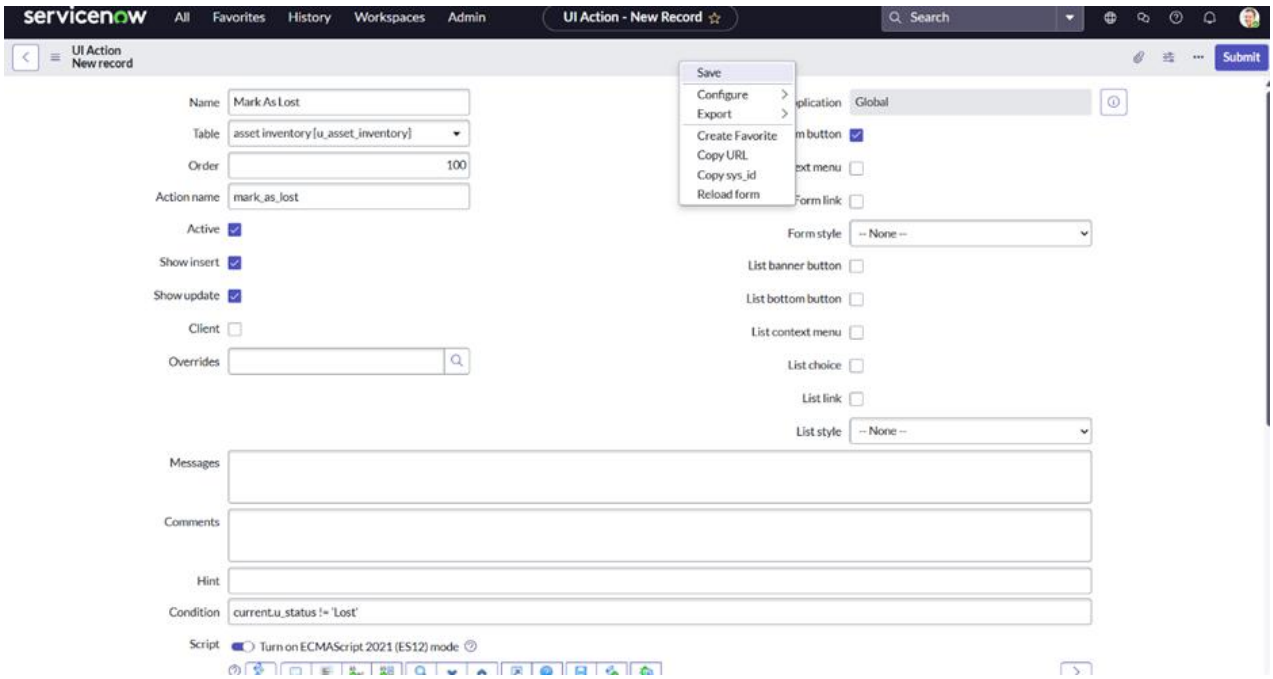
```
current.u_status = 'Lost';

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

4. Check the form button box

5. Click on save

MILESTONE 2: UI ACTION



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

STEPS:

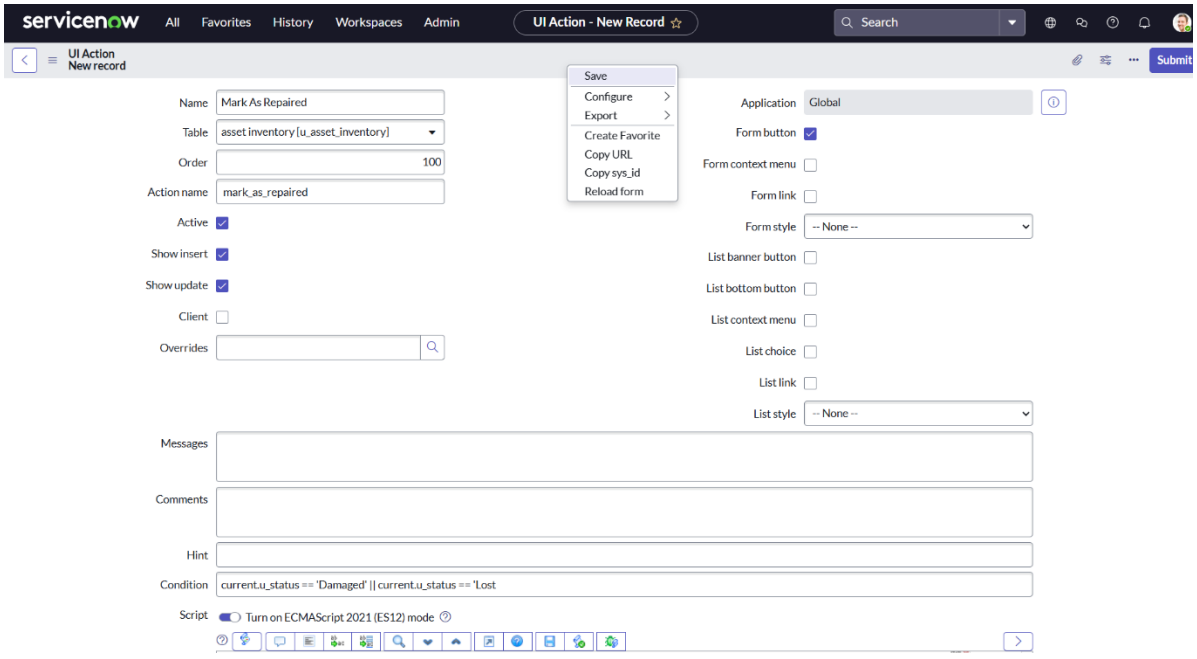
1. Navigate to System Definition >> UI action
2. Click on New
3. Fill in the details;
    1. Name: Mark As Repaired
    2. Table: Asset Inventory
    3. Action name: mark_as_repaired
    4. Condition: current.u_status == 'Damaged' || current.u_status == 'Lost’
    5. Script:

current.u_status = 'Available';

current.update();

action.setRedirectURL(current);

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



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

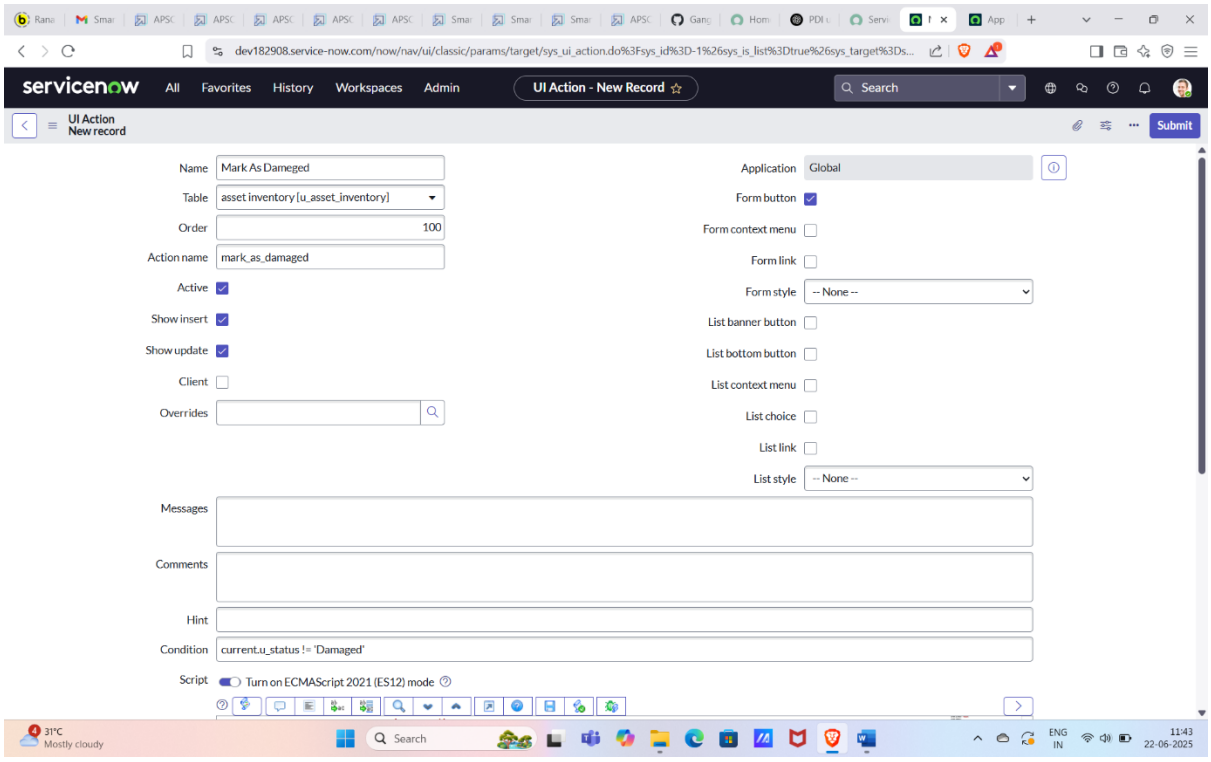
Condition: current.u_status != 'Damaged'

Script:

```
current.u_status = 'Damaged';

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

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



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.

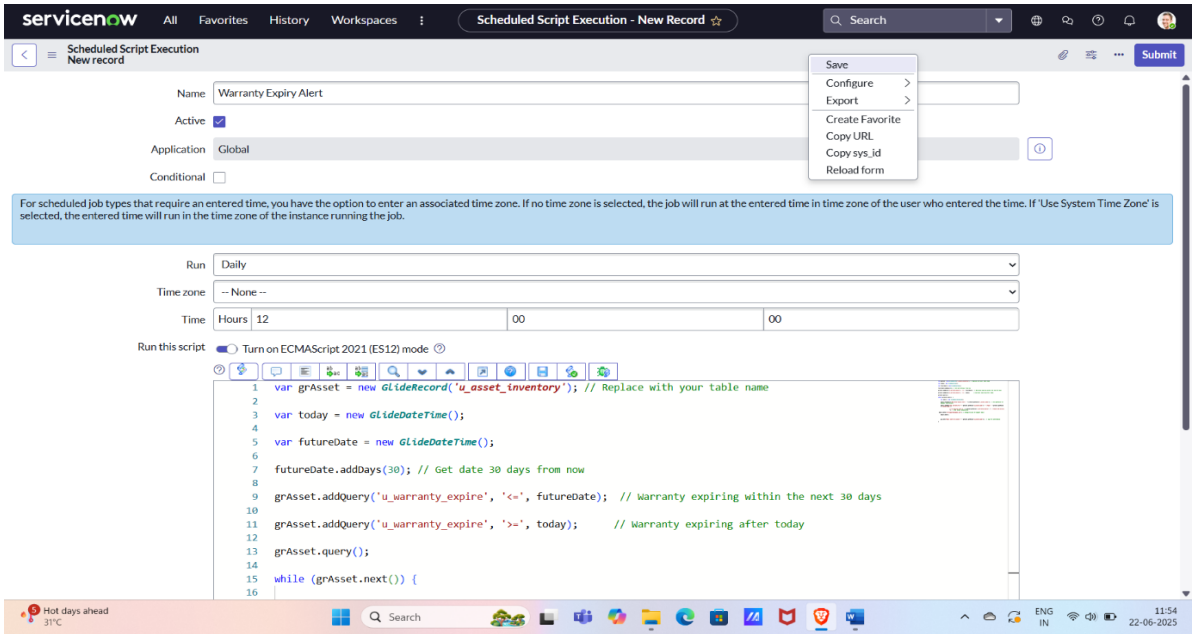
USE:

- 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



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.

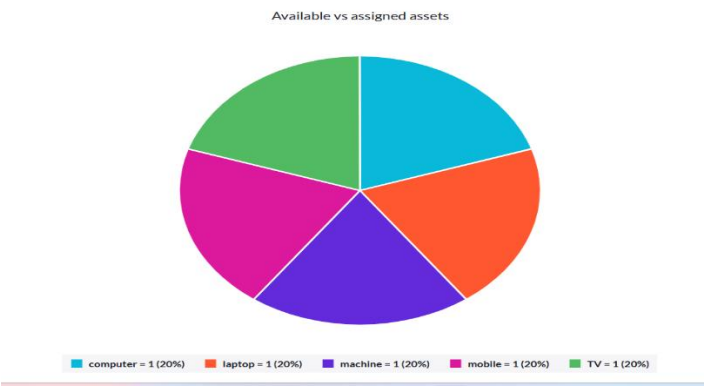
USE:

- 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.

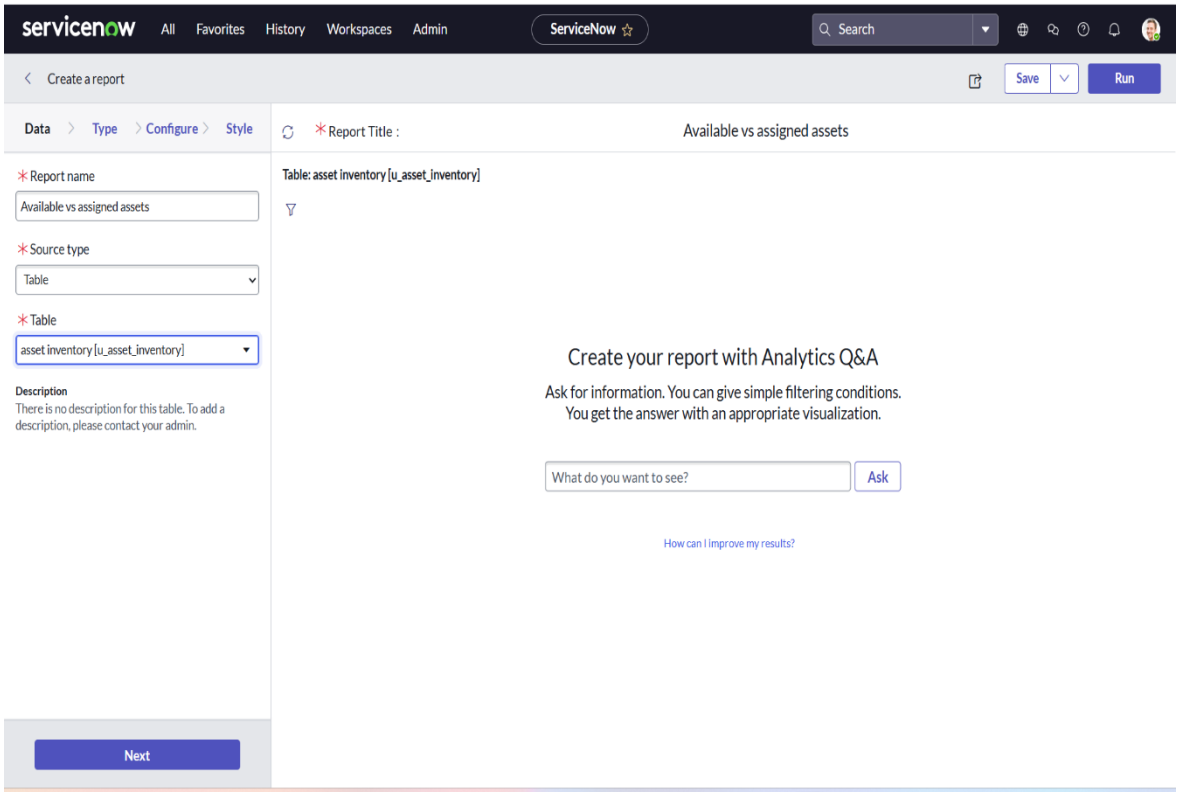
Activity 1: create report

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



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