

## PROJECT DESIGN

<b>TEAM ID</b>	LTVIP2025TMID30733
<b>PROJECT NAME</b>	Asset Management Portal

### Proposed solution:

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

S. No	Parameter	Description
1	Problemstatement (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 Asset Management Portal is a robust and centralized web application designed to optimize the management of both physical and digital assets within an organization. It enables automated asset tracking, real-time inventory updates, user-friendly self-service features, and smart alerts for maintenance and warranty. By reducing human error and manual workload, the portal enhances operational efficiency, promotes responsible asset usage, and ensures timely decision-making through insightful reporting dashboards.
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 it does?

- **Tracks Assets:** Monitors location, status, and ownership of items like laptops, tools, or software licenses.
- **Manages Requests:** Employees can request assets; managers can approve or reject them through workflows.
- **Schedules Maintenance:** Sends alerts for servicing, warranty expiry, or calibration.
- **Generates Reports:** Provides visual insights into asset usage, availability, and performance.
- **Ensures Compliance:** Maintains logs and audit trails for regulatory and internal reviews.

#### Types:

- **Tables**
  - To store asset-related data in structured rows and fields.
- **Fields**
  - Data points like asset type, status, purchase date, etc. used for tracking each record.
- **UI Actions**
  - Custom buttons like “Mark as Lost”, “Mark as Damaged”, “Mark as Repaired” for quick updates.
- **Scheduled Jobs**
  - Automated scripts that run at set intervals, like warranty expiry alerts.
- **Reports**
  - Visual data presentations such as pie charts comparing assigned vs available assets.
- **Testing Activities**
  - Verification steps for UI actions and scheduled jobs to ensure proper functionality.

### Asset Management Portal

#### MILESTONE 1: TABLE

## **Activity 1: create table**

### **PURPOSE:**

The purpose of creating a table in the Asset Management Portal is to store and organize asset-related data in a structured format. This table serves as the backbone of the portal's database, where each asset record (like ID, name, type, owner, status, location, etc.) is stored in rows and columns for easy access, tracking, and reporting.

### **USE:**

Central Data Storage: The table stores all asset information in a centralized and searchable format.

### **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

**Table - New Record**

ServiceNow recommends creating custom tables in scoped applications. To learn more about creating scoped applications, see [Scoped Applications](#).

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 in a table. Applications use tables and records to manage data and processes. [More Info](#)

\* Label:

\* Name:

Extends table:

Application:

Create module: ☒

Create mobile module: ☒

Add module to menu:

New menu name:

**Table Columns**

for text Search

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

Submit Cancel

Related Links

[Track in Update Sets](#)

## MILESTONE 1: TABLE

### Activity 2: create fields

#### PURPOSE:

To define specific data points (like asset name, type, status) that will be stored in each record of the table.

#### USE:

Fields allow the portal to capture detailed asset information (e.g., serial number, owner, location) and enable accurate tracking, searching, filtering, and reporting of assets. They ensure that each asset entry is complete and consistent.

#### STEPS:

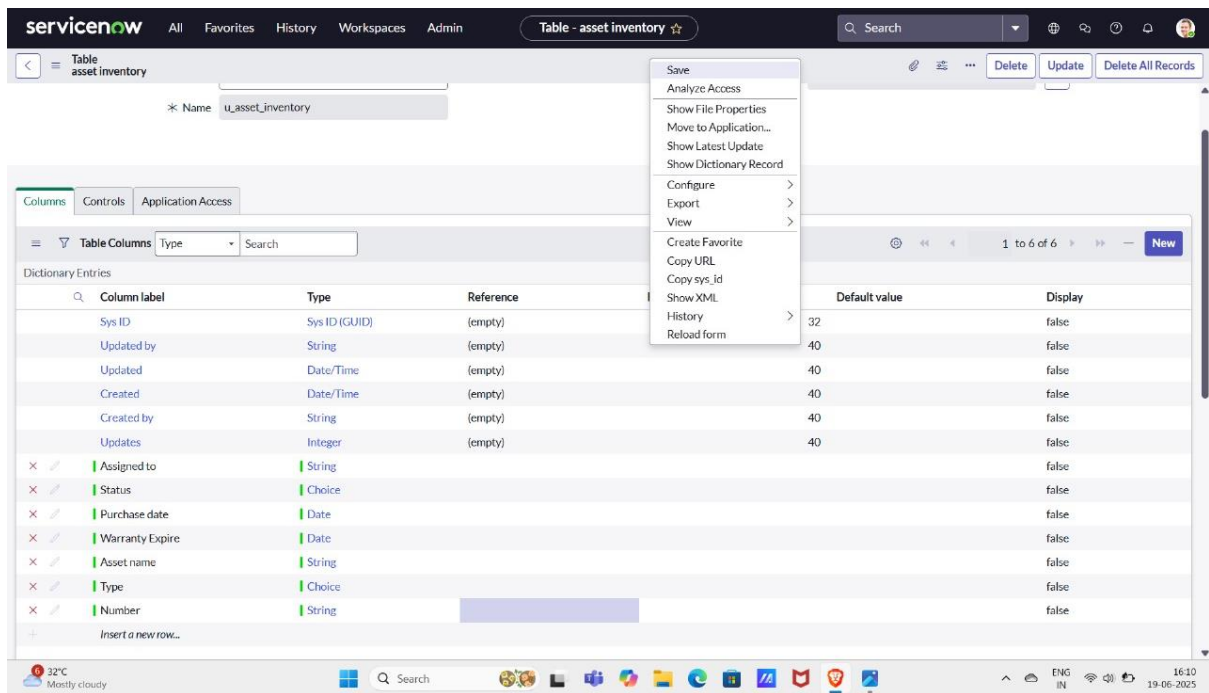
1) After saving the table scroll down

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



## MILESTONE 2: UI ACTION

### Activity 1: create UI action 1

#### PURPOSE:

To add a custom button or link on a form or list that performs a specific action when clicked.

#### USE:

UI Actions improve user interaction by allowing quick actions like "Assign Asset," "Return Asset," or "Request Approval" directly from the portal interface. This enhances usability and speeds up common tasks.

#### 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

The screenshot shows the ServiceNow 'UI Action - New Record' configuration page. The 'Name' field is 'Mark As Repaired', the 'Table' is 'asset inventory [u\_asset\_inventory]', and the 'Order' is '100'. The 'Action name' is 'mark\_as\_repaired'. The 'Active' checkbox is checked. The 'Show insert' and 'Show update' checkboxes are also checked. The 'Client' checkbox is unchecked. The 'Overrides' field is empty. The 'Messages' field is empty. The 'Comments' field is empty. The 'Hint' field is empty. The 'Condition' field contains the script 'current.u\_status == 'Damaged' || current.u\_status == 'Lost''. The 'Script' field is set to 'Turn on ECMAScript 2021 (ES12) mode'. The 'Form button' checkbox is checked. The 'Form context menu' checkbox is unchecked. The 'Form link' checkbox is unchecked. The 'Form style' dropdown is set to '-- None --'. The 'List banner button' checkbox is unchecked. The 'List bottom button' checkbox is unchecked. The 'List context menu' checkbox is unchecked. The 'List choice' checkbox is unchecked. The 'List link' checkbox is unchecked. The 'List style' dropdown is set to '-- None --'. A 'Save' dropdown menu is open, showing options: 'Save', 'Configure', 'Export', 'Create Favorite', 'Copy URL', 'Copy sys\_id', and 'Reload form'. The 'Submit' button is visible in the top right corner.

## MILESTONE 2: UI ACTION

### Activity 2: create UI action 2

#### 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

The screenshot shows the ServiceNow 'UI Action - New Record' form. The form is titled 'UI Action - New Record' and has a search bar at the top. The form fields are as follows:

- Name: Mark As Repaired
- Table: asset inventory [u\_asset\_inventory]
- Order: 100
- Action name: mark\_as\_repaired
- Active: ☒
- Show insert: ☒
- Show update: ☒
- Client: ☐
- Overrides:
- Messages:
- Comments:
- Hint:
- Condition: `current.u_status == 'Damaged' || current.u_status == 'Lost'`
- Script: ☒ Turn on ECMAScript 2021 (ES12) mode

A 'Save' dropdown menu is open, showing the following options:

- Save
- Configure
- Export
- Create Favorite
- Copy URL
- Copy sys\_id
- Reload form

The 'Form button' checkbox is checked, and the 'Form style' is set to '-- None --'.

## MILESTONE 2: UI ACTION

### Activity 3: create UI action 3

## STEPS:

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

Name: Mark As Damedged

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

The screenshot shows the ServiceNow 'UI Action - New Record' form. The form is divided into several sections:

- Header:** ServiceNow logo, navigation tabs (All, Favorites, History, Workspaces, Admin), and a search bar.
- Form Fields:**
  - Name:** Mark As Damedged
  - Table:** asset inventory [u\_asset\_inventory]
  - Order:** 100
  - Action name:** mark\_as\_damaged
  - 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'
- Script:** ☒ Turn on ECMAScript 2021 (ES12) mode

The bottom of the screen shows a Windows taskbar with the date 22-06-2023 and time 11:43.



## MILESTONE 3: SCHEDULED JOB

### Activity 1: create scheduled job

#### PURPOSE:

To automate tasks that need to run at specific times or intervals without manual intervention.

#### USE:

Scheduled jobs are used to automatically check asset status, send maintenance alerts, or generate daily/weekly reports, ensuring timely actions and reducing manual workload.

#### 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

**servicenow** All Favorites History Workspaces Scheduled Script Execution - New Record

Scheduled Script Execution  
New record

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

```
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
11 grAsset.addquery('u_warranty_expire', '>=', today); // Warranty expiring after today
12
13 grAsset.query();
14
15 while (grAsset.next()) {
16
```

## **MILESTONE 4: REPORT**

### **Activity 1: create report**

#### **PURPOSE:**

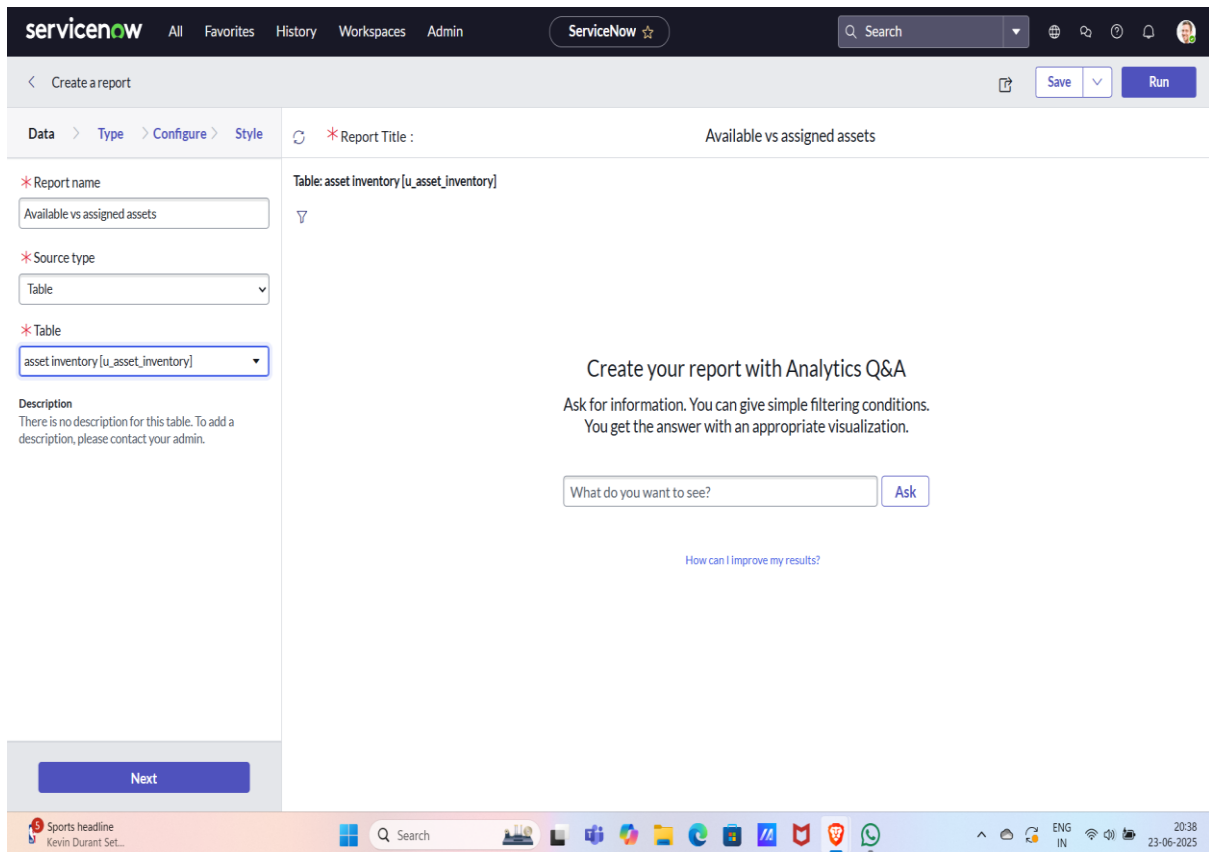
To visually display and analyse data stored in the system for better decision-making.

#### **USE:**

Reports help track asset usage, availability, maintenance status, and inventory trends. They support data-driven decisions, improve transparency, and assist in audits or reviews.

#### **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

### Activity 1: testing UI action

#### PURPOSE:

To ensure that the UI action (button or link) works correctly and performs the intended function without errors.

#### USE:

Testing UI actions like "Assign Asset" or "Return Asset" ensures the system responds correctly to user inputs, improves reliability, and provides a smooth user experience.

#### 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 Tutors
- 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

### Activity 2: testing scheduled job

#### PURPOSE:

To verify that the scheduled job runs automatically at the set time and performs its task correctly.

#### USE:

Testing ensures that automated tasks like sending maintenance alerts or generating reports run as expected, helping maintain system accuracy and reducing manual effort.

#### STEPS:

- Navigate to background scripts
- Write the Scheduled job script in the background scripts
- Click on Run Script
- Check the result

The screenshot shows the ServiceNow interface for an Asset Inventory record. The top navigation bar includes 'servicenow', 'All', 'Favorites', 'History', 'Workspaces', and a search bar. The record title is 'asset inventory - IST001101'. Below the title, there are tabs for 'Update', 'Mark As Damaged', 'Mark As Repaired', and 'Delete'. The form fields are organized into two columns. The left column contains: 'Number' (IST001101), 'Purchase date' (2025-06-22), 'Status' (Lost), and 'Assigned to' (Abel Tutor). The right column contains: 'Asset name' (Laptop), 'Type' (laptop), and 'Warranty Expire' (2025-07-30). At the bottom of the form, there are buttons for 'Update', 'Mark As Damaged', 'Mark As Repaired', and 'Delete'.

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

servicenow

AllFavoritesHistoryWorkspacesAdmin

ServiceNow

Search

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\_expiry', '<=', futureDate); // Warranty expiring within the next 30 days

10

11

grAsset.addQuery('u\_warranty\_expiry', '>=', today); // Warranty expiring after today

12

13

grAsset.query();

14

15

while (grAsset.next()) {

16

17

var email = new GlideEmailOutbound();

18

19

email.setSubject("Warranty Expiry Alert: " + grAsset.getValue('u\_assest\_name')); // Use getValue for dynamic field access

20

21

email.setBody("The warranty for " + grAsset.getValue('u\_assest\_name') + " (Type: " + grAsset.getValue('u\_asset\_type') +

22

23

" is expiring soon on " + grAsset.getValue('u\_warranty\_expiry') + ". Please take action."); // Get values dynamically

24

25

email.setTo('it-support@company.com'); // Change to your IT support email

26

27

email.send();

28

29

30

31

gs.info("Email sent for assest: " + grAsset.getValue('u\_assest\_name')); // Log for confirmation

32

33

}

Run Script

in scope global

Record for rollback?

Execute in sandbox?

Execute as scriptlet?

Cancel after 4 hours

+ Instance Scripts

Feels hotter Now

Search

ENG IN

12:42 22-06-2025