Asset Management Portal – ServiceNow Administration Project

Team ID : LTVIP2025TMID30344

Team Size : 4

Team Leader : Vuyyuru Sri Dinesh

Team member : Padala Mercy Elizabeth

Team member : Pilla Durga Sandeep

Team member : Rachakonda Anirudh Saketh

1. INTRODUCTION

1.1 Project Overview

The **Asset Management Portal** is a **ServiceNow-based platform** designed to simplify the tracking, management, and allocation of physical and digital assets across an organization. It automates the asset lifecycle from procurement to disposal and offers administrators robust control through real-time dashboards and workflows.

1.2 Purpose

The portal aims to enhance **asset visibility**, reduce loss, improve efficiency, and facilitate informed decision-making through centralized data and automation tools.

2. IDEATION PHASE

2.1 Problem Statement

The **Asset Management Portal** will streamline the tracking, management, and allocation of both physical and digital assets across an organization. It will automate asset assignments, improve record-keeping, and generate real-time reports. Alerts for maintenance and replacements ensure optimal performance and reduced downtime. The system reduces asset loss and boosts decision-making through centralized management.

2.2 Empathy Map Canvas

Section	Details	Says	Thinks	Does	Feels
User	Employee/Administrator	"I don’t know	"This manual	Tracks in Excel, sends	Frustrated, uncertain

Section Details	Says	Thinks	Does	Feels
	where the asset is now"	tracking takes too much time"	reminder emails	about asset status

2.3 Brainstorming Highlights

- Automate asset status updates
- Scheduled warranty and maintenance alerts
- UI buttons for asset status change
- Real-time reports
- Email alerts before warranty expiry

---

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

The customer journey through the Asset Management Portal can be broken down into distinct phases:

- **Request**: An employee submits an asset request via the portal.
- **Approval**: An administrator reviews and approves the request, then assigns the asset.
- **Usage**: The employee uses the assigned asset.
- **Maintenance**: The administrator receives automated alerts for scheduled service or maintenance needs.
- **Disposal/Return**: The asset is either returned to inventory or retired from service.

3.2 Solution Requirements

Functional:

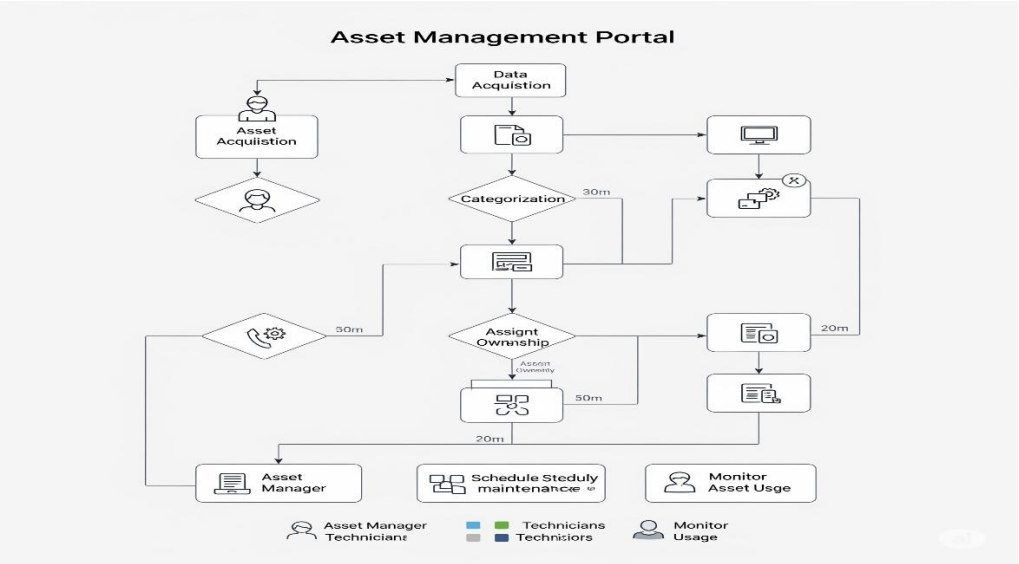
- Robust **request & approval workflow** for assets.
- Efficient **asset assignment** and tracking.
- **Real-time reporting** capabilities on asset status and utilization.
- **Scheduled jobs** for proactive alerts (e.g., warranty expiry).

- Intuitive **UI actions** for managing asset status changes.

Non-Functional:

- **Responsive interface** accessible from various devices.
- **Secure access** with role-based permissions.
- **Minimal training** needed for users and administrators.

3.3 Data Flow Diagram



A[Employee] --> B(Request Asset);

B --> C(ServiceNow Portal);

C --> D(Admin Approves/Assigns);

D --> E(Asset Inventory Table);

E --> F(ServiceNow Alerts);

F --> G(IT/Support Team);

E --> H(ServiceNow Reports/Dashboards);

3.4 Technology Stack

- **Platform**: ServiceNow
- **Scripting**: JavaScript (utilizing GlideRecord and GlideDateTime APIs)
- **Database**: ServiceNow Tables (configured with a CMDB-like structure for assets)
- **Reporting**: ServiceNow's native reporting tools, including Pie Charts and Tables

---

4. PROJECT DESIGN

4.1 Problem-Solution Fit

Manual asset tracking is inherently inefficient, prone to errors, and time-consuming. ServiceNow's robust workflows and powerful scripting capabilities provide a comprehensive solution by enabling **real-time status updates**, **proactive alerts**, and **dynamic reporting**, effectively addressing the challenges of traditional asset management.

4.2 Proposed Solution

The proposed solution is an **admin-driven portal** designed to manage the entire asset lifecycle, incorporating several key features:

- **UI buttons**: "Mark as Lost," "Mark as Damaged," and "Mark as Repaired" for quick status updates.
- **Email alerts**: Automated notifications for impending warranty expiry dates.
- **Daily scheduled checks**: Background processes to monitor asset conditions and trigger alerts.
- **Pie chart**: Visual representation of asset status distribution (e.g., available, assigned, lost).

4.3 Solution Architecture

The architecture is built entirely within the ServiceNow platform:

- **Frontend**: Utilizes the **ServiceNow portal** for user interaction, including custom forms and UI actions.
- **Backend**: Powered by **ServiceNow scripts** and workflows that handle business logic and data processing.
- **Database**: A custom ServiceNow table named asset_inventory serves as the central data repository.
- **Automation**: Implemented using **Glide scripts** for sending emails and automating asset status changes.

---

5. PROJECT PLANNING & SCHEDULING

5.1 Planning Timeline

Phase	Duration	Tasks
Requirement Gathering	Week 1	Define portal functionalities, design asset table, create necessary fields.
Development	Week 2	Implement UI actions, develop reporting dashboards.
Automation & Testing	Week 3	Configure scheduled jobs, perform comprehensive UI testing.
Finalization	Week 4	Complete project documentation, conduct final report review.

---

6. FUNCTIONAL & PERFORMANCE TESTING

6.1 Performance Testing

Thorough testing was conducted to ensure the portal's functionality and performance:

- **UI actions** were rigorously tested on various asset records to confirm their speed and accuracy in updating asset statuses.
- **Email alerts** were verified by simulating asset records with near-expiry warranty dates, ensuring timely notification delivery.
- The **scheduled job** was tested by executing its script in the background to confirm its proper execution and data processing.

---

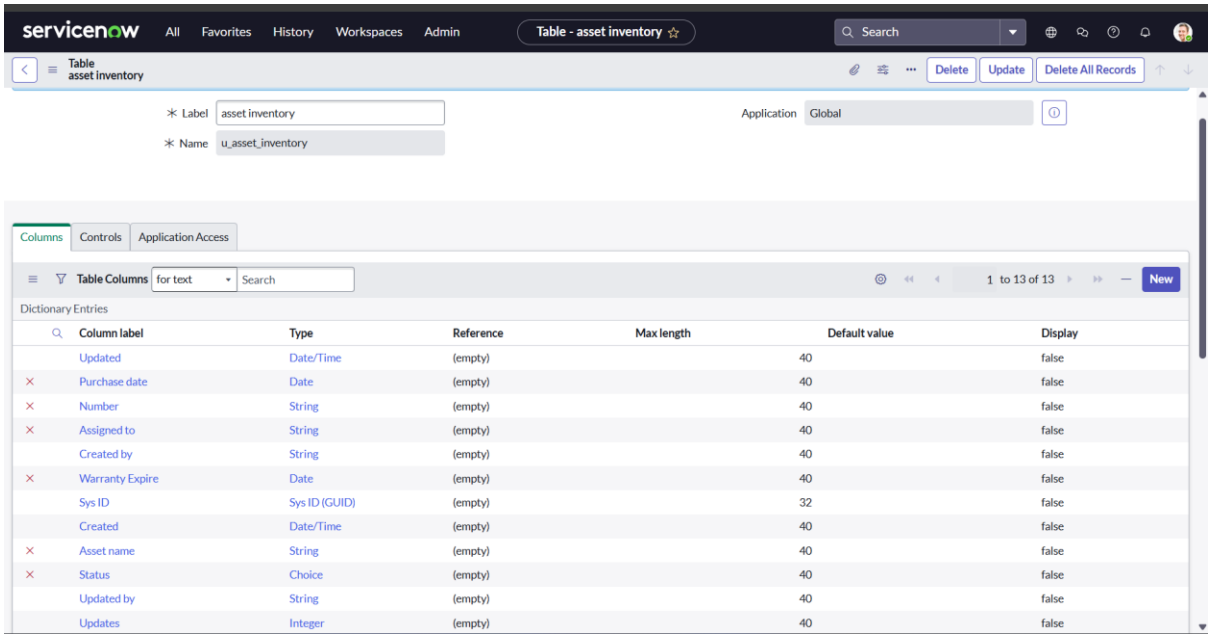
7. RESULTS

7.1 Output Screenshots

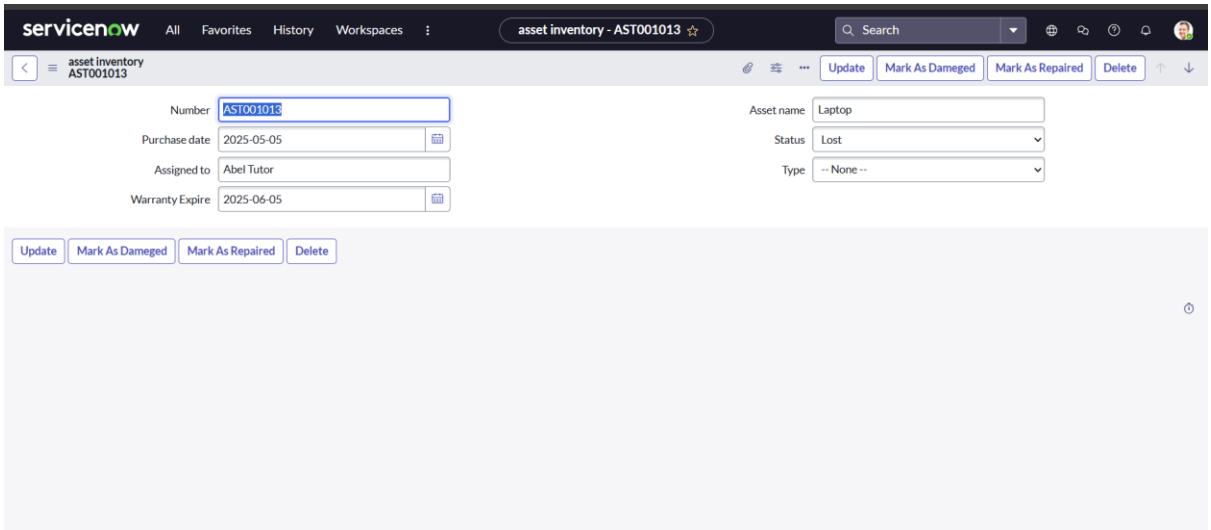
Asset Inventory Table :

Field Name	Type	Description
Assigned to	String	Name of the employee or department to whom the asset is currently assigned.
Status	Choice	Current status of the asset (e.g., Available, Assigned, Damaged, Lost).
Purchase date	Date	The date when the asset was purchased.

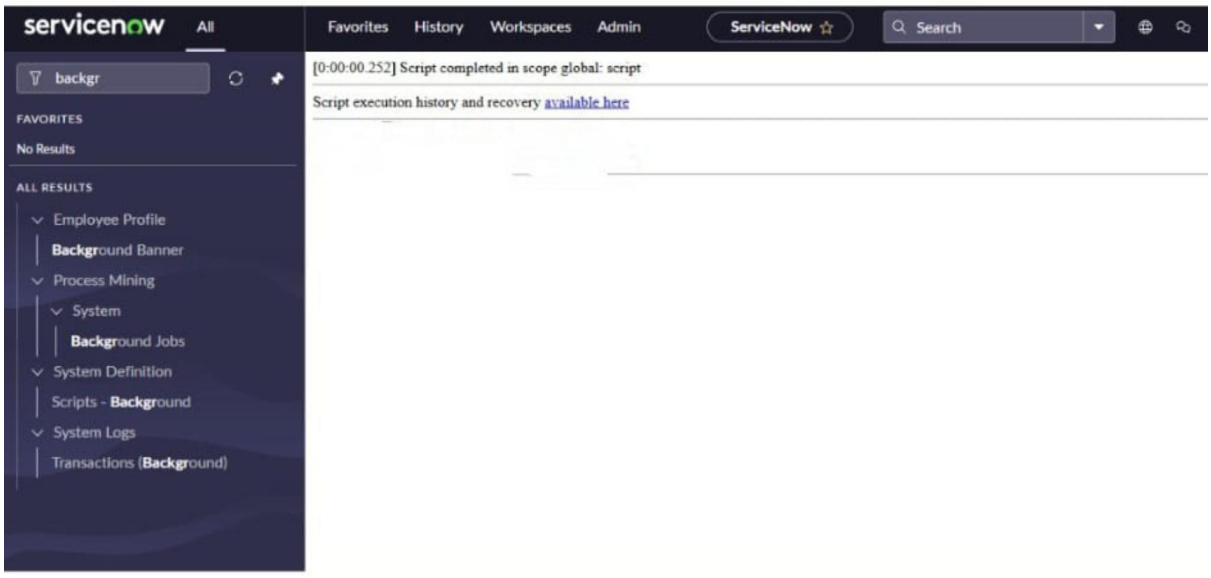
Field Name	Type	Description
Warranty Expire	Date	The expiration date of the asset's warranty.
Asset name	String	The name or identifier of the asset (e.g., Laptop, Projector).
Type	Choice	Type or category of the asset (e.g., Laptop, Desktop, Mobile).
Number	String	Serial number, asset tag, or identification number of the asset.



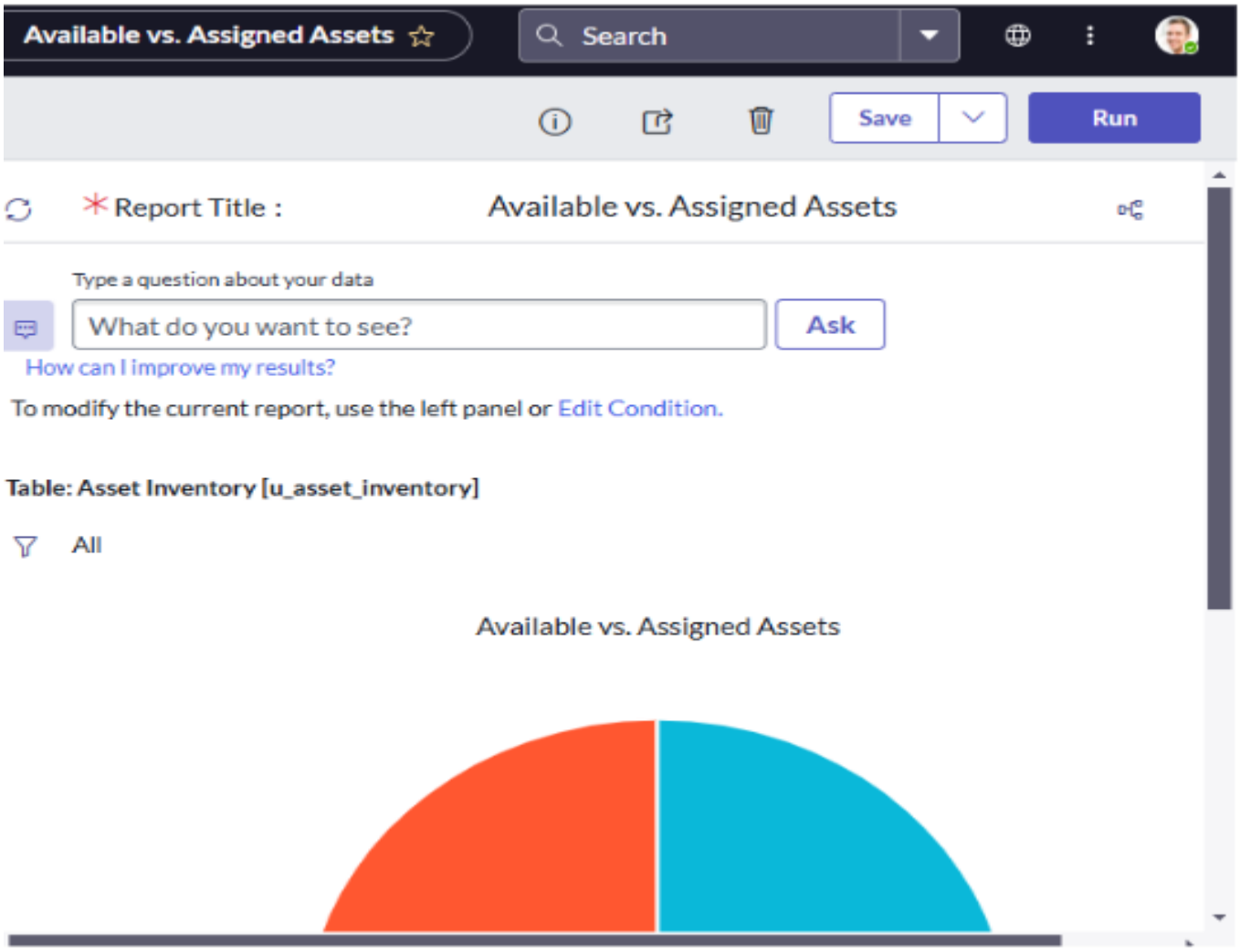
“Mark As Lost” UI Action in Form :



Email Alert Triggered for Warranty :



Pie Chart of Available vs Assigned Assets:



8. ADVANTAGES & DISADVANTAGES

Advantages

- **Real-time asset tracking**: Provides immediate visibility into asset locations and statuses.
- **Scheduled alerts reduce downtime**: Proactive notifications for maintenance and replacements minimize operational interruptions.
- **Minimal training required**: Intuitive interface ensures quick user adoption.
- **Customizable and scalable**: Built on ServiceNow, allowing for easy adaptation and expansion.

Disadvantages

- **ServiceNow licensing cost**: A significant consideration for organizations.
- **Dependency on correct script configuration**: Errors in scripts can impact system functionality.
- **Requires familiarity with ServiceNow scripting**: Maintenance and further development may require specialized skills.

---

9. CONCLUSION

The **Asset Management Portal** built using **ServiceNow** offers a streamlined, automated, and centralized solution to asset lifecycle management. With intuitive UI actions, real-time reporting, and proactive alert systems, the project demonstrates how ServiceNow can effectively replace manual systems with scalable and efficient workflows. By improving asset accountability and operational efficiency, this platform helps organizations maximize asset value, reduce costs, and enhance overall productivity.

---

10. FUTURE SCOPE

The project has significant potential for future enhancements, including:

- **QR code/barcode scanning**: For faster and more accurate asset check-in and check-out.
- **Integration with procurement systems**: To automate the asset acquisition process from purchase order to deployment.
- **Mobile version of the portal**: To enable on-the-go asset management.

- **Asset depreciation calculator**: To track the financial value of assets over their lifecycle.

---

11. APPENDIX

Source Code: UI Actions

Mark As Lost

JavaScript

```
current.u_status = 'Lost';
current.update();
action.setRedirectURL(current);
```

Mark As Repaired

JavaScript

```
current.u_status = 'Available';
current.update();
action.setRedirectURL(current);
```

Mark As Damaged

JavaScript

```
current.u_status = 'Damaged';
current.update();
action.setRedirectURL(current);
```

Scheduled Job Script

JavaScript

```
var grAsset = new GlideRecord('u_asset_inventory');
var today = new GlideDateTime();
var futureDate = new GlideDateTime();

futureDate.addDays(30);
```

```
grAsset.addQuery('u_warranty_expire', '<=', futureDate);
grAsset.addQuery('u_warranty_expire', '>=', today);
grAsset.query();

while (grAsset.next()) {
  var email = new GlideEmailOutbound();
  email.setSubject("Warranty Expiry Alert: " + grAsset.getValue('u_assest_name'));
  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.");
  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'));
}
```

GitHub & Project Demo Link

- **GitHub Repo**:
- **Demo Video**: