LAPTOP REQUEST CATALOG ITEM

Objective:

The primary objective of this project is to automate and streamline the employee laptop request process within ServiceNow. This is achieved by creating a dynamic, user-friendly Service Catalog Item that ensures accurate data collection, provides a guided user experience, and tracks all development changes in a dedicated Update Set for seamless deployment.

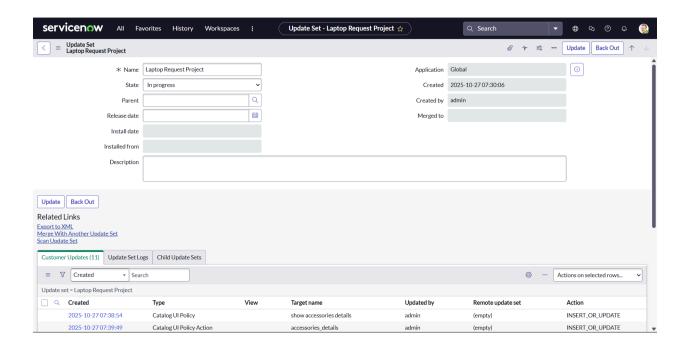
Introduction:

This project addresses the inefficiencies inherent in the organization's existing manual process for requesting laptops. That manual system is prone to delays and errors due to a lack of structure. The implemented solution, the Laptop Request Catalog Item, transforms this process into a governed, digital workflow. Key features include the use of variables to capture specific user needs (like laptop model and justification), the implementation of a Catalog UI Policy for conditional visibility of fields, and the creation of a UI Action to enhance user experience on the shopping cart. All work is captured in the Laptop Request Update Set to facilitate system migration and governance.

Implementation Steps:

Step 3.1. Create Local Update Set

- 1. Navigate to **System Update Sets > Local Update Sets**.
- 2. Create a new set named Laptop Request.
- 3. Select **Submit and Make Current** to ensure all development work is tracked.



Step 3.2. Create Service Catalog Item

- 1. Go to Service Catalog > Catalog Definitions > Maintain Items.
- 2. Click New and fill in the details:

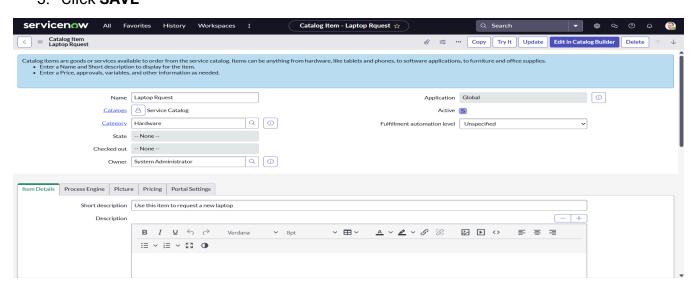
• Name: Laptop Request

Catalog: Service Catalog

Category : Hardware

• Short Description: Use this item to request a new laptop

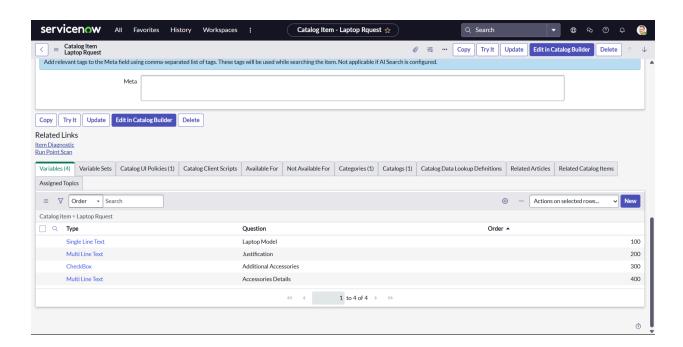
3. Click SAVE



Step 3.3. Add Form Variables

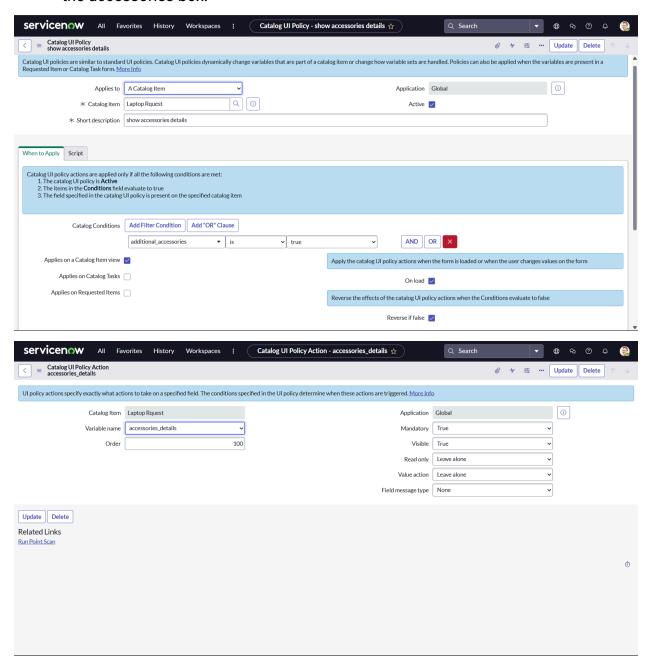
Add the following four variables in the **Variables** related list of the item:

Variable Name	Туре	Order	Purpose
laptop_model	Single line text	100	Capture the desired laptop model.
justification	Multi line text	200	Capture the reason for the request.
additional_accessories	Checkbox	300	Flag if extra accessories are needed.
accessories_details	Multi line text	400	Capture specific details about accessories.



Step 3.4. Create Catalog UI Policy (Dynamic Logic)

- 1. Create a new Catalog UI Policy named show accessories details.
- 2. Set the **Condition** to: additional_accessories is true.
- Create a Catalog UI Policy Action for the accessories_details variable, setting Mandatory and Visible to True. This hides the details field unless the user checks the accessories box.



Step 3.5. Create UI Action (Reset Form)

- 1. Navigate to **System Definition > UI Actions**.
- 2. Create a new action with the following:

• **Table**: Shopping Cart [sc_cart]

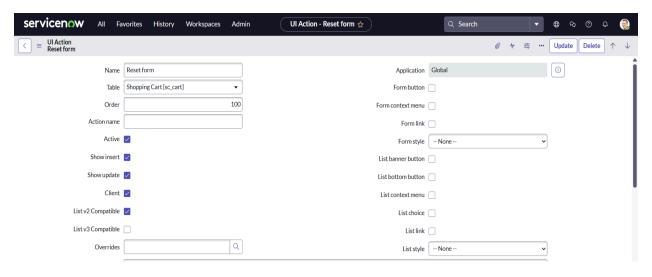
• Action name: Reset form

• Client: Check the box.

Script:

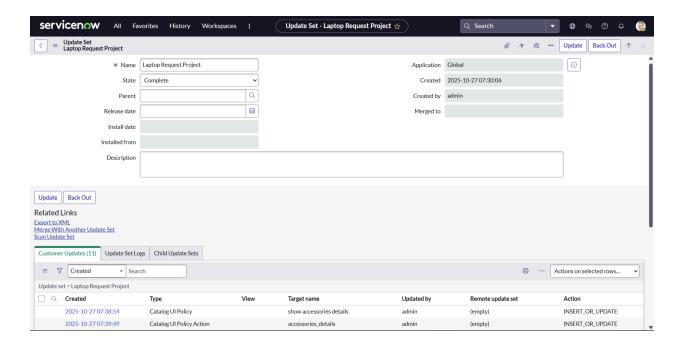
```
function resetForm() {
    // Clears all form fields
    g_form.clearForm();
    // Notifies the user
    alert('The form has been reset.');
}
```

3. Click Save.



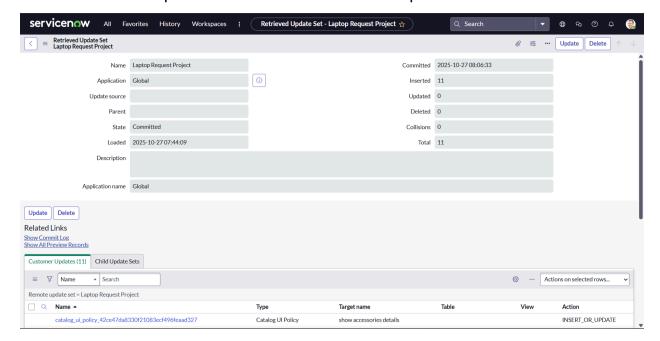
Step 3.6. Export Update Set

- 1. Navigate to **System Update Sets > Local Update Sets**.
- 2. Open Laptop Request and set the state to Complete.
- 3. Download the file by clicking on **Export to XML.**



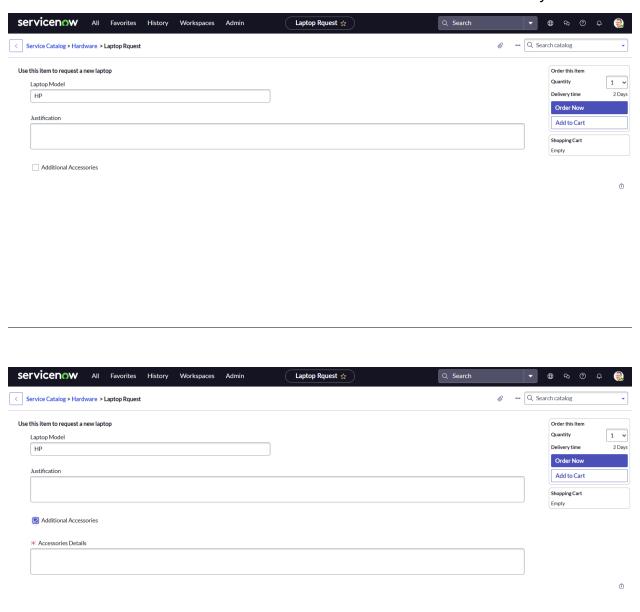
Step 3.7. Retriving the Update Set

- Login into another instance and navigate to System Update Sets > Retrived
 Update Sets.
- 2. Upload the downloaded XML file using **Import Update Set from XML** from the related links tab.
- Commit the update set and see the related tab updates.



Step 3.8. Testing

- 1. Navigate to **Service Catalog** > **Catalog** in the target instance.
- 2. Open Hardware
- 3. As per our scenario, when we click on **Additional Accessories** checkbox then **Accessories Details** field will be visible and that should be mandatory.



Conclusion:

The Laptop Request Catalog Item project successfully delivered a complete, governed, and user-friendly solution for managing employee laptop requests. By capturing all changes in the designated Update Set, the project ensures adherence to deployment best practices. The dynamic logic provided by the UI Policy streamlines the form experience, and the added Reset form UI Action enhances overall user satisfaction. The solution is now ready for formal testing and subsequent migration to higher environments.