**LAPTOP REQUEST CATALOG ITEM**

**Team Id:NM2025TMID15835**

**Team Members: 3**

**Team Leader :** PriyaDarshini K

Team Member 1:Keerthana v

Team Member 2:Prasanna M

Team Member 3:Perinba Prabakaran A

**PROBLEM STATEMENT**

Employees currently face difficulties when requesting laptops because the process is manual, slow, and lacks guidance. There is no dynamic form behavior to help users provide accurate information, which often leads to errors and delays. To improve this, a new Service Catalog item will be created that allows employees to easily submit laptop requests. The catalog form will include dynamic fields, clear instructions, and options like resetting the form for convenience. In addition, all updates will be properly tracked to maintain governance and support smooth deployment.

**OBJECTIVE**

* To automate and streamline the laptop request process within the organization using ServiceNow’s Service Catalog.
* To replace manual and error-prone procedures with a dynamic, user-friendly catalog item.
* To ensure accurate data collection through guided form fields and UI policies.
* To improve service delivery efficiency and enhance employee satisfaction by providing a modern request experience.

**SKILLS UTILIZED**

* **ServiceNow Development** – Creating catalog items, variables, and UI policies.
* **Service Catalog Management** – Designing user-friendly and dynamic request forms.
* **Scripting (JavaScript/Client Scripts)** – Implementing reset functionality and form validation.
* **Update Set Management** – Capturing, exporting, and importing changes across instances.
* **Process Automation** – Replacing manual workflows with automated catalog-driven processes.
* **Problem-Solving & Testing** – Ensuring dynamic behavior works as expected and fulfills requirements.

**LEARNING OUTCOMES**

* Gained hands-on experience in **ServiceNow Service Catalog development**, including creating catalog items, variables, and categories.
* Learned to design **dynamic request forms** using Catalog UI Policies and UI Policy Actions.
* Understood how to implement **form reset functionality** through UI Actions and client-side scripting.
* Developed skills in **Update Set Management**, including creating, exporting, importing, previewing, and committing update sets across different instances.
* Improved knowledge of **process automation** by replacing manual workflows with streamlined, user-friendly solutions.
* Strengthened problem-solving skills through **testing and troubleshooting** dynamic behaviors (e.g., showing mandatory fields based on conditions).
* Enhanced ability to document and communicate technical steps in a clear and structured way.

**ADVANTAGES**

* **Automation of Requests** – Eliminates manual processes, reducing delays and errors.
* **Dynamic Form Behavior** – Shows or hides fields based on user input, ensuring accuracy and ease of use.
* **Improved Efficiency** – Speeds up laptop request handling, saving time for both employees and IT teams.
* **Better Data Accuracy** – Mandatory fields and guided forms ensure correct information is captured.
* **User-Friendly Experience** – Provides employees with a modern, intuitive interface for submitting requests.
* **Governance & Tracking** – All updates are tracked through update sets, ensuring proper change management.
* **Reusability** – The same approach can be applied to other hardware or service requests.
* **Cross-Instance Deployment** – Update sets allow easy migration of changes between different ServiceNow environments.
* **Enhanced Employee Satisfaction** – Faster, smoother, and transparent request process improves the end-user experience.

**CHALLENGES FACED**

* **Understanding ServiceNow Navigation** – Initially, locating the correct modules (like Update Sets, Catalog Items, and UI Policies) required practice.
* **Dynamic Form Configuration** – Setting up UI Policies and ensuring fields became visible/mandatory only under the right conditions was challenging.
* **Scripting for Reset Action** – Writing and testing the client-side script for the Reset button required troubleshooting to avoid errors.
* **Update Set Management** – Ensuring all changes were captured correctly in the update set before exporting/importing between instances needed careful attention.
* **Cross-Instance Migration** – Importing update sets into another instance sometimes resulted in missing elements, requiring re-validation.
* **Testing Dynamic Behavior** – Validating that the “Accessories Details” field appeared only when the checkbox was selected and making it mandatory took multiple iterations.
* **Documentation Effort** – Recording every step clearly for governance and knowledge sharing was time-consuming but essential.

**FUTURE ENHANCEMENTS**

* **Approval Workflow** – Add a manager or IT approval step before the laptop request is processed.
* **Email Notifications** – Configure automated email alerts to notify employees and approvers about request status.
* **Integration with Asset Management** – Connect the catalog item with ServiceNow Asset Management to automatically update inventory when laptops are issued.
* **Predefined Laptop Models** – Replace free-text entry with a dropdown list of approved laptop models to standardize requests.
* **Cost Display** – Show estimated cost of laptops and accessories for better transparency.
* **Reporting & Analytics** – Build dashboards to track the number of laptop requests, processing time, and fulfillment status.
* **Mobile Access** – Optimize the catalog item for mobile users, allowing requests via the ServiceNow mobile app.
* **Multi-Language Support** – Enable localization so employees in different regions can use the form in their preferred language.

**IMPLEMENTATION IN REAL LIFE**

The **Laptop Request Catalog Item** can be practically implemented within an organization to replace the traditional manual request process. In a real business environment:

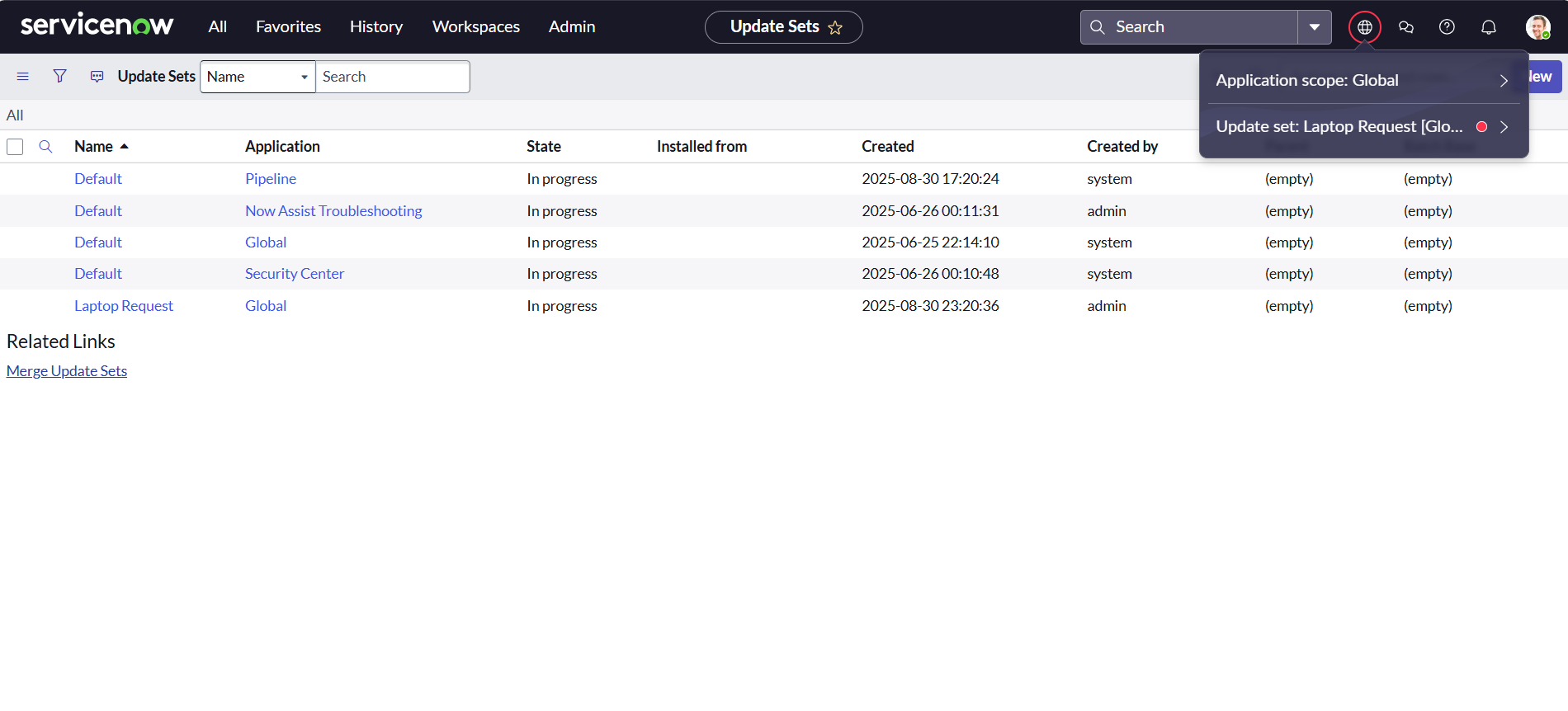
* **Employees** will log in to ServiceNow and submit a request through the **Service Catalog → Hardware → Laptop Request** item.
* The **dynamic form** guides them to enter only relevant details (e.g., Accessories Details shown only if “Additional Accessories” is checked), ensuring accurate data.
* Once submitted, the request can automatically trigger an **approval workflow** involving the manager and IT team.
* After approval, the request can be linked with **Asset Management** to check inventory and issue the laptop.
* **Notifications** (emails or ServiceNow alerts) can keep the requester updated at every stage.
* The IT team gains **tracking and reporting capabilities**, helping them analyze laptop demand, delivery timelines, and costs.

**TASK INITIATION**

**Milestone 1:UPDATE SET**

**Activity 1:Creating Local Update Set**

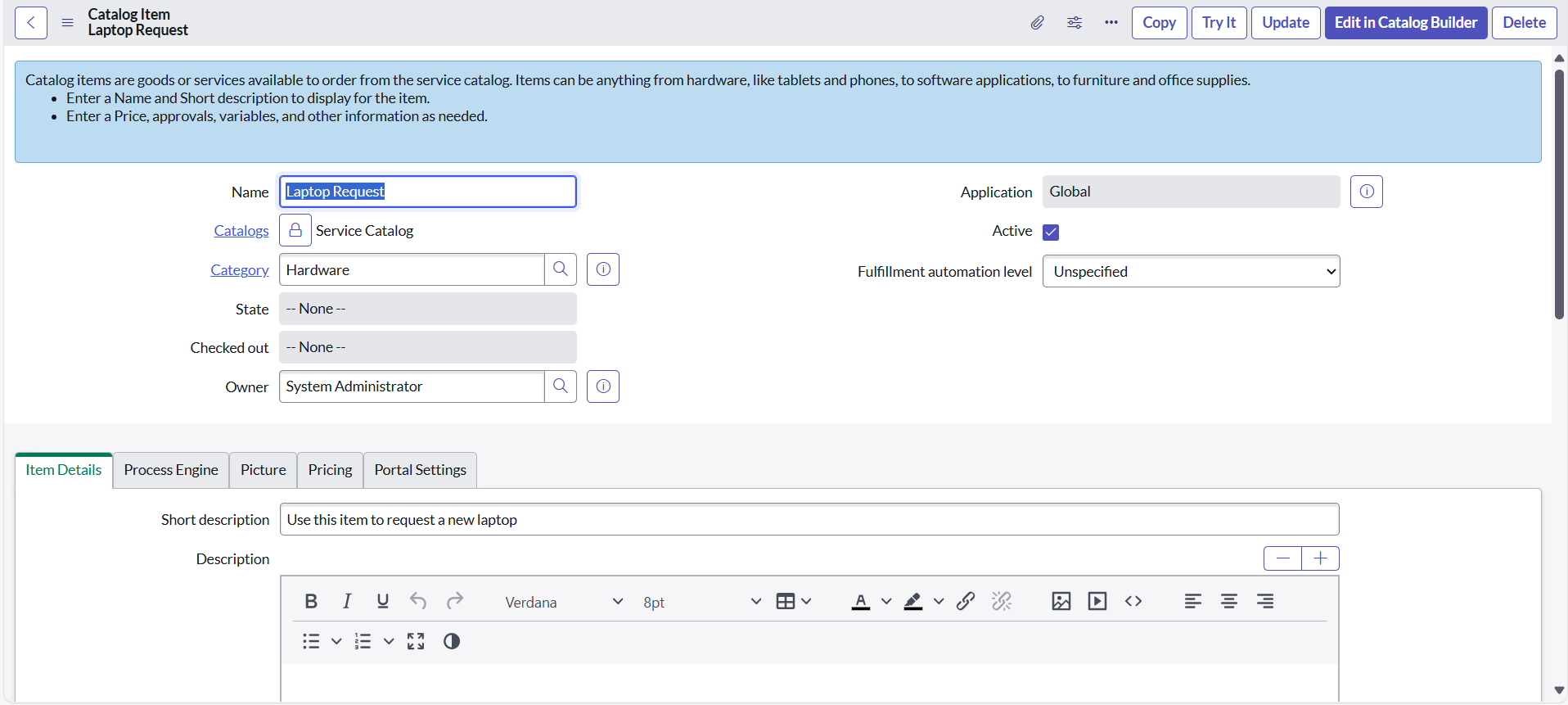
1. Log in to ServiceNow.
2. From the left-hand navigation, go to All and search for Update Sets.
3. Under System Update Sets, select Local Update Sets.
4. Click on New to create a new update set.
5. Enter the required details and name the update set “Laptop Request”.
6. Click Submit, then choose Make Current.
7. Once activated, this update set will capture all related changes.



**Fig1: Update Set Creation – The Laptop Request update set is listed under System Update Sets**

**Milestone 2 :SERVICE CATALOG ITEM**

**Activity 1: Creating service catalog item**

1. Log in to **ServiceNow**.
2. In the left navigation panel, click on **All** and go to **Service Catalog**.
3. Under **Catalog Definitions**, select **Maintain Items**.
4. Click on **New** to create a new catalog item.
5. Enter the details for the new catalog item as follows:
   1. **Name**: Laptop Request
   2. **Catalog**: Service Catalog
   3. **Category**: Hardware
   4. **Short Description**: Use this item to request a new laptop
   5. After filling in the fields, click **Save** to create the catalog item.

**Fig2: Catalog Item Setup – You created a new catalog item named Laptop Request under the Hardware category with descriptions**

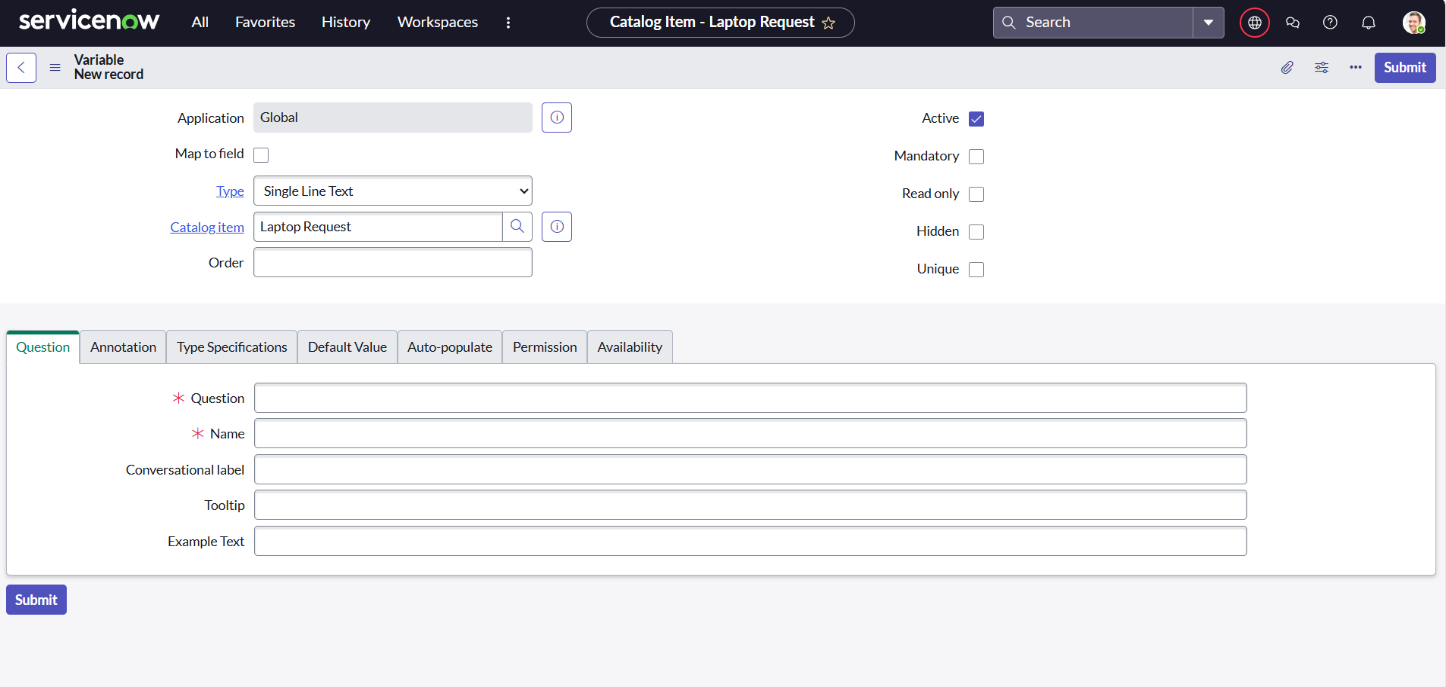
**Activity 2: Add variables**

**Step 1: Add Variables to the Catalog Item**

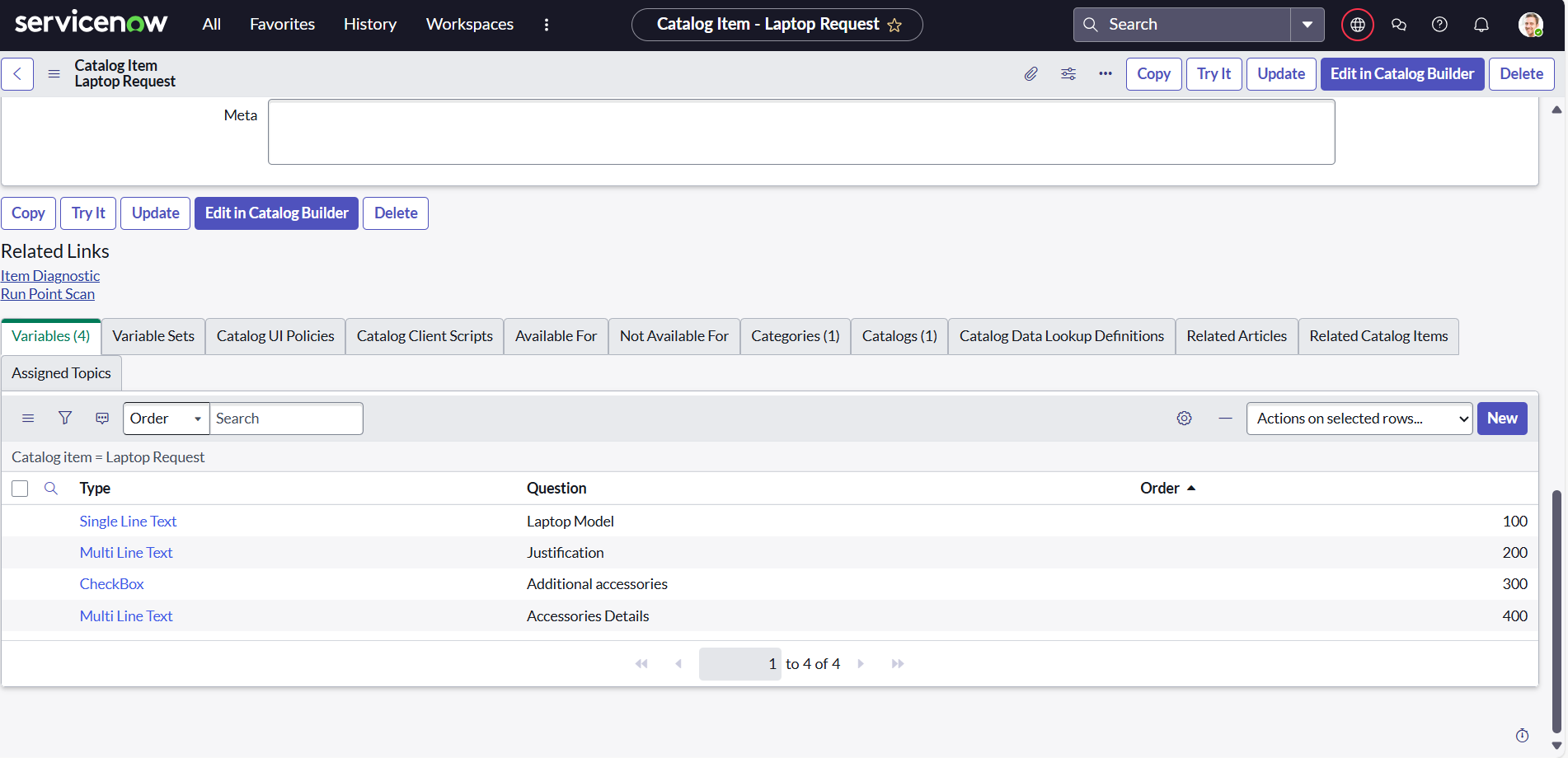
1. Once the catalog item form is saved, scroll down to the Variables (Related List) section.
2. Click New and fill in the following details:
   * Variable 1: Laptop Model
     + Type: Single Line Text
     + Name: laptop\_model
     + Order: 100
   * Click Submit.
3. Repeat the same process for the remaining variables:
   * Variable 2: Justification
     + Type: Multi Line Text
     + Name: justification
     + Order: 200
   * Variable 3: Additional Accessories
     + Type: Checkbox
     + Name: additional\_accessories
     + Order: 300
   * Variable 4: Accessories Details
     + Type: Multi Line Text
     + Name: accessories\_details
     + Order: 400

**Step 2: Save the Catalog Item**

* After adding all the variables, they will now appear under the newly created catalog item.
* Finally, click **Save** to update and store the catalog item form.



**Fig 3**



**Fig 4**

**Fig3 & Fig4: Variable Creation – You added variables like Laptop Model, Justification, Additional Accessories, Accessories Details**

**MileStone 3:UI POLICY**

**Activity 1: Create Catalog UI Policies**

Step 1: Open the Catalog Item

1. In the left navigation panel, click on All and search for Service Catalog.
2. Under Catalog Definition, select Maintain Items.
3. Search for the previously created item “Laptop Request”.
4. Open the Laptop Request record and scroll down to the Catalog UI Policies related list.

Step 2: Create a Catalog UI Policy

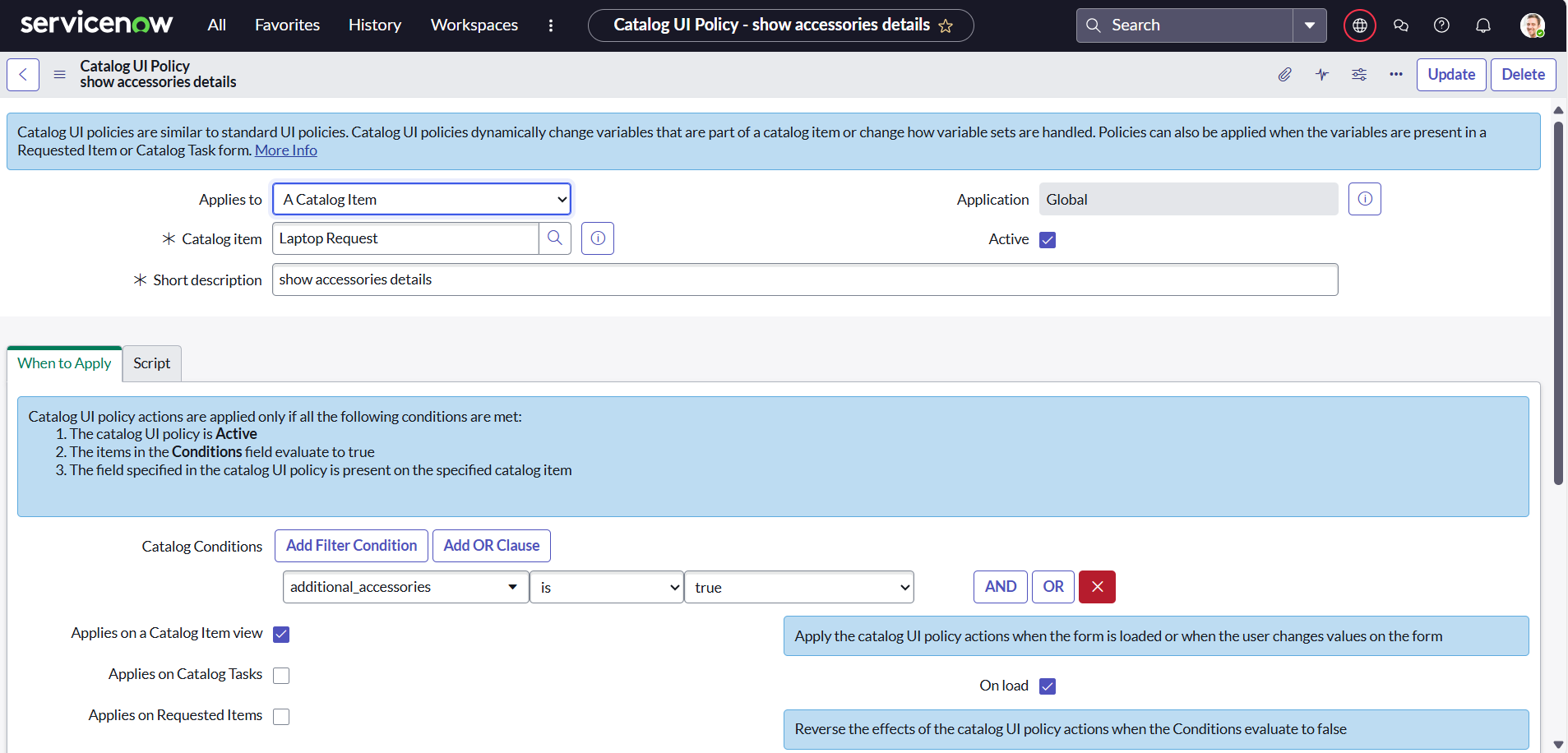
1. In the Catalog UI Policies tab, click New.
2. Enter the Short Description: *Show Accessories Details*.
3. Under the When to Apply section, set the following condition:
   * Field: additional\_accessories
   * Operator: is
   * Value: true
4. Click Save (do not click Submit yet).

Step 3: Define Catalog UI Policy Actions

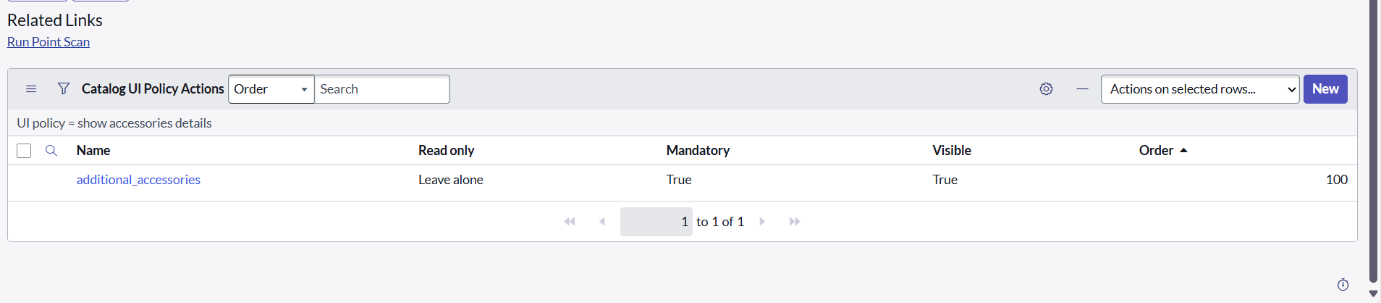
1. Scroll down to the Catalog UI Policy Actions related list.
2. Click New and fill in the details:
   * Variable Name: accessories\_details
   * Order: 100
   * Mandatory: True
   * Visible: True
3. Click Save.

Step 4: Save the UI Policy

* Once the policy action is saved, click Save again on the Catalog UI Policy form to complete the setup.



**Fig5: Catalog UI Policy – You configured a UI policy to show *Accessories Details* only when *Additional Accessories* is checked**



**Fig6: UI Policy**

* A UI Policy named “show accessories details” is applied.
* It makes the field additional\_accessories → Mandatory = True and Visible = True.
* This ensures users cannot submit a Laptop Request without filling in accessories info.

**Milestone4:UI ACTIONS**

**Activity 1: Create ui action**

Steps to Create a UI Action for Reset Form

1. Log in to ServiceNow.
2. In the left navigation panel, click on All and search for UI Actions.
3. Under System Definition, select UI Actions.
4. Click on New to create a new UI Action.
5. Fill in the details as follows:
   * Table: Shopping Cart (sc\_cart)
   * Order: 100
   * Action Name: Reset form
   * Client: Checked
6. In the Script field, enter the following code:

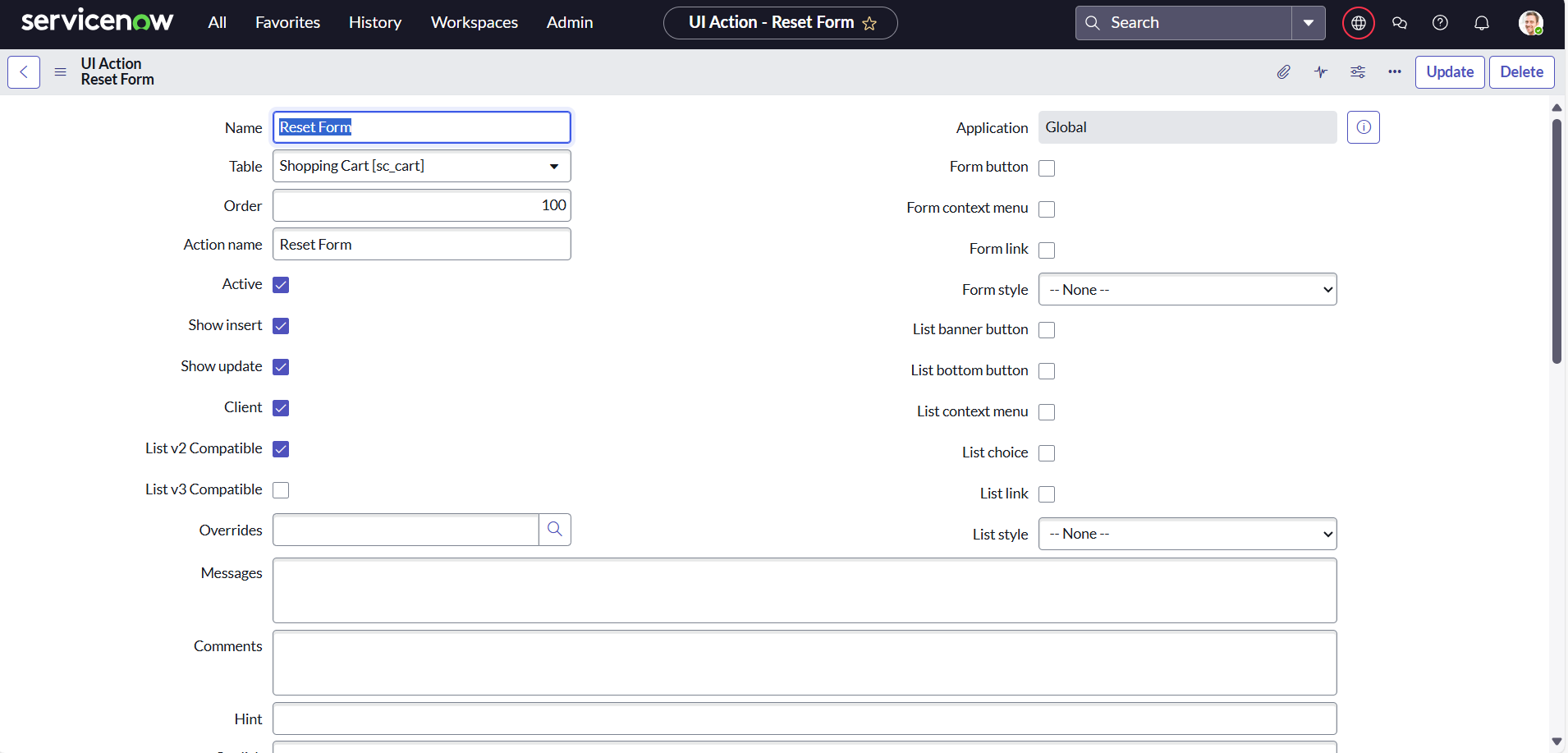
function resetForm() {

g\_form.clearForm(); // Clears all fields in the form

alert("The form has been reset.");

}

1. Click Save to create the UI Action.

****

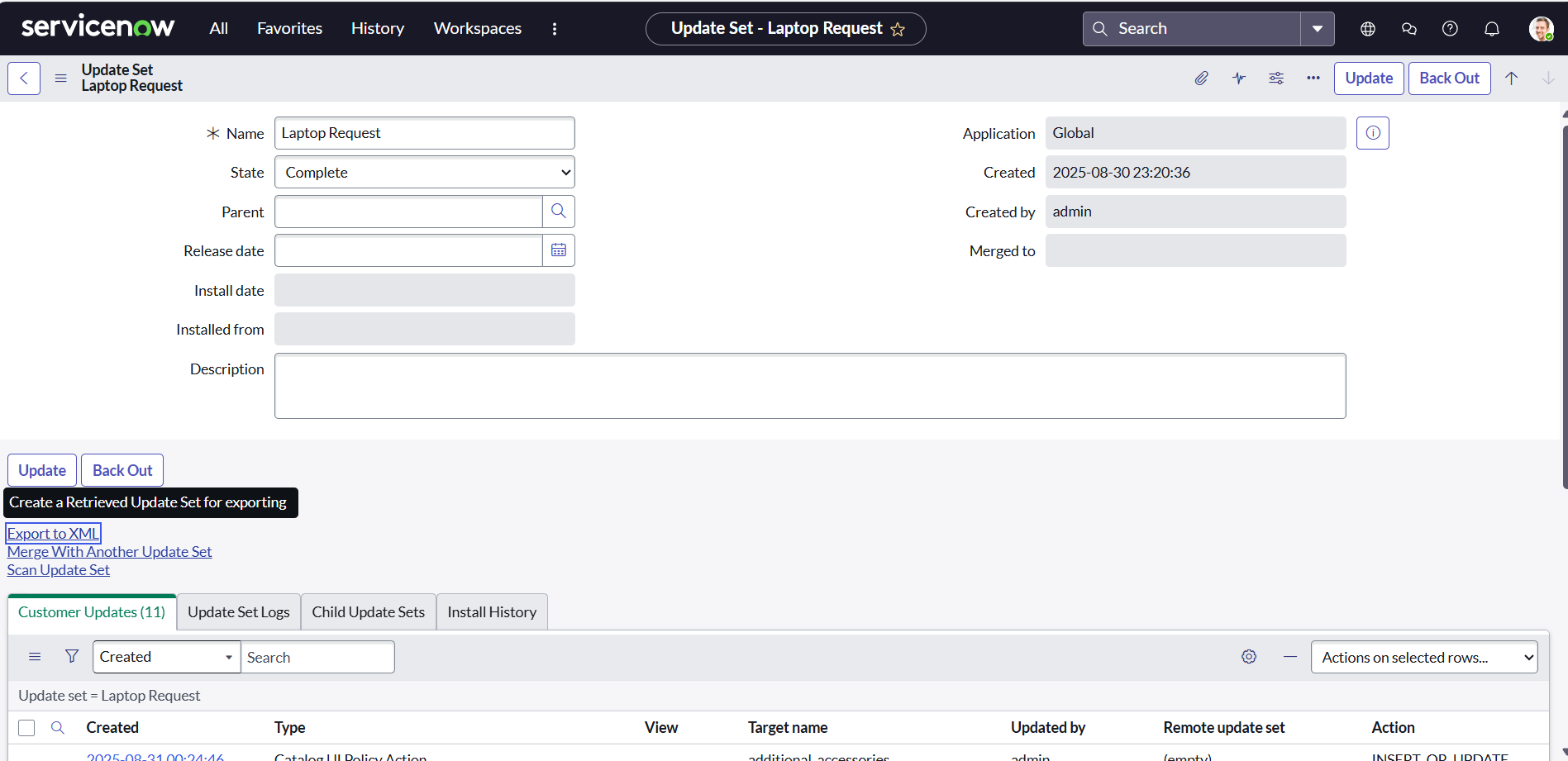
**Fig 7: UI Action**

* A UI Action called “Reset Form” is created for the Shopping Cart [sc\_cart] table.
* It can be used to clear/reset fields when a user clicks it, improving the user experience.

**Milestone5: EXPORT UPDATE SET**

**Activity: Exporting changes to another instances**

1. In the left navigation panel, click on **All** and search for **Update Sets**.
2. Under **System Update Sets**, select **Local Update Sets**.
3. Open the update set you created earlier, **“Laptop Request Project”**.
4. Change the **State** of the update set to **Complete**.
5. Scroll down to the **Updates** related list to view all the changes captured in this update set.
6. Click on **Export to XML**.
   * An XML file will be downloaded containing the update set details.



**Fig 8: Import XML**

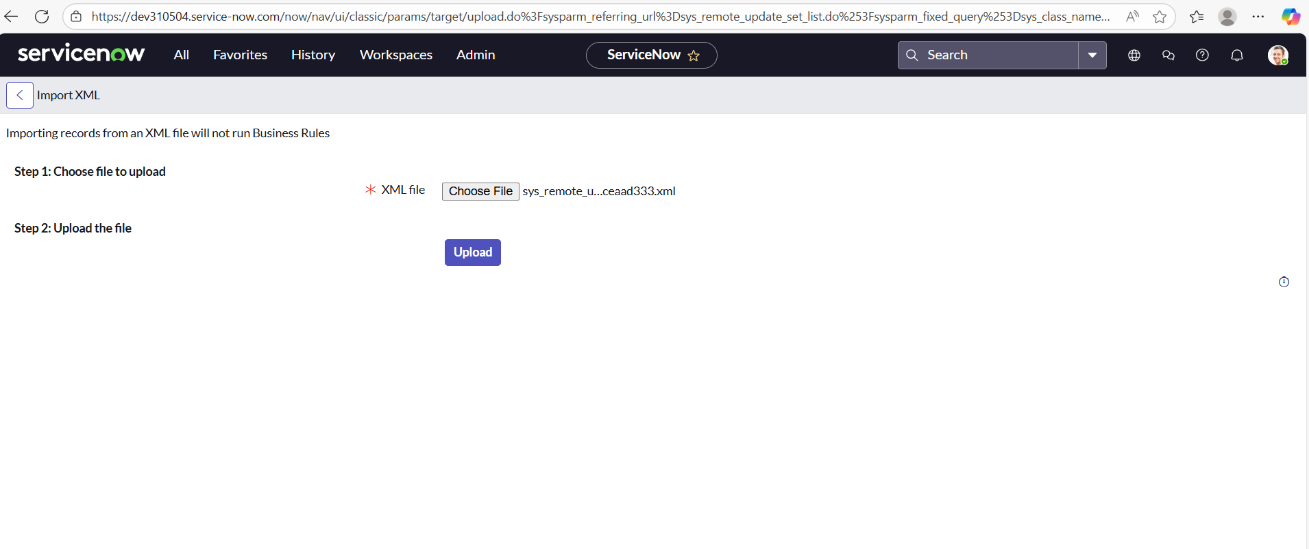
* An Update Set XML file is being imported.
* This is how you move changes (catalog items, policies, UI actions, etc.) from one ServiceNow instance (dev/test) to another (prod).

**Milestone 6: LOGIN TO ANOTHER INSTANCE**

**Activity 1: Retrieving the update set**

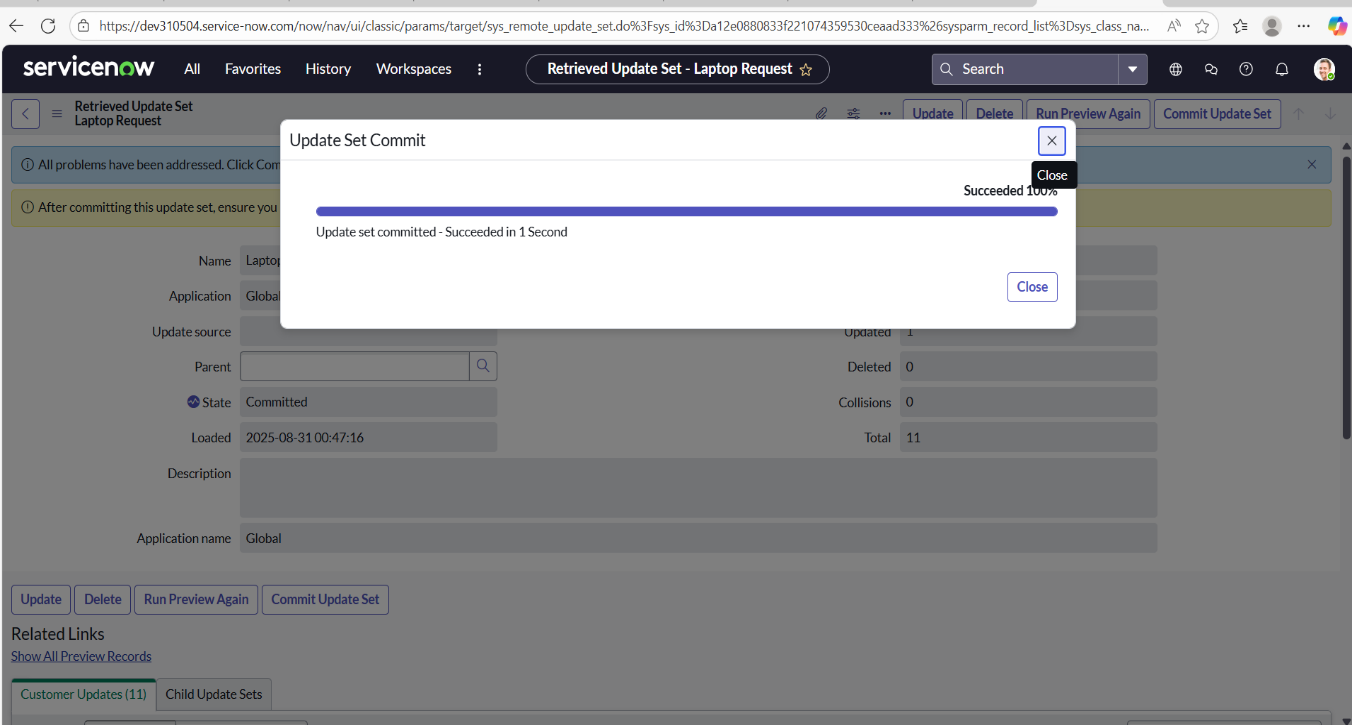
Steps to Import and Commit the Update Set in Another Instance

1. Open a new ServiceNow instance in an incognito window.
2. Log in using your credentials.
3. In the left navigation panel, click on All and search for Update Sets.
4. Under System Update Sets, select Retrieved Update Sets.
5. The retrieved update set list will open. Scroll down and click on Import Update Set from XML.
6. Upload the previously downloaded XML file.
7. Click Upload – the update set will now appear in the list.
8. Open the retrieved update set “Laptop Request Project”.
9. Click on Preview Update Set.
10. After previewing, click on Commit Update Set.
11. Check the Updates related list tab to review all the changes included.



**Fig9: Commit Update Set**

* The **Laptop Request update set** has been committed successfully.
* Now all customizations (form fields, UI actions, UI policies, etc.) are available in the target instance.

****

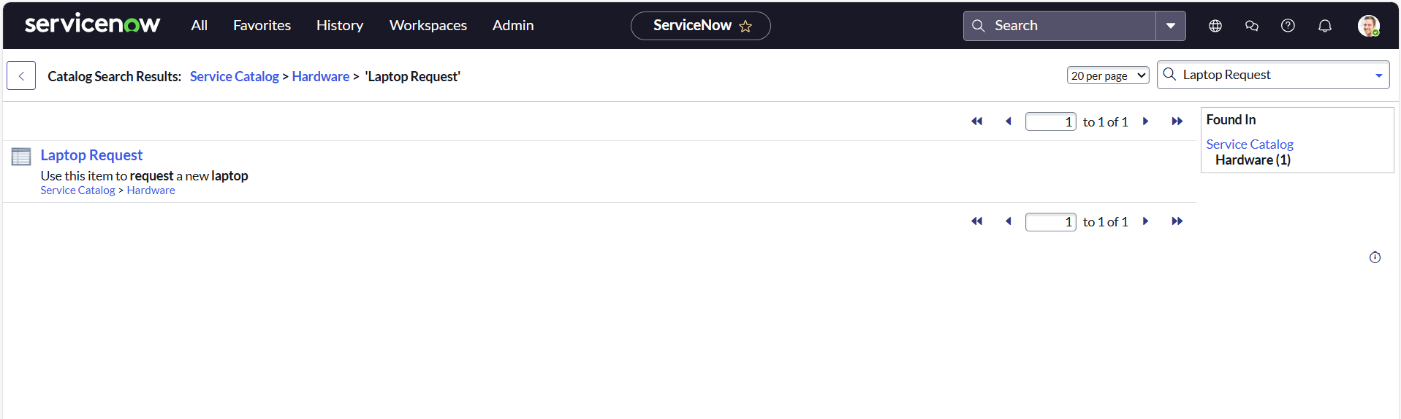
**Fig 10:Update and Commit**

**Milestone 7:TESTING**

**Activity:Testing Catalog Item**

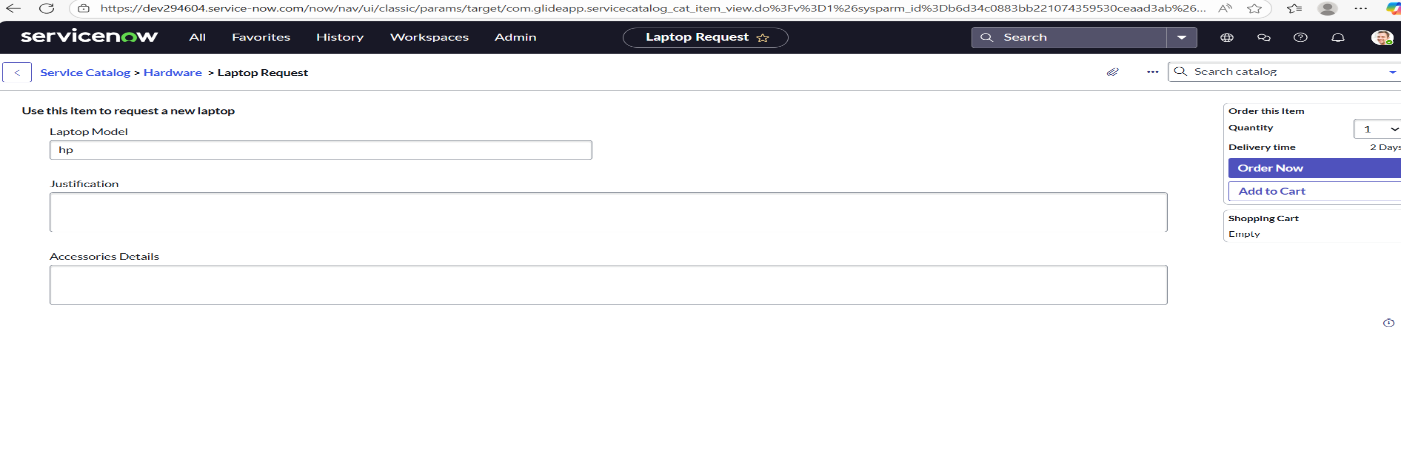
Steps to Verify the Laptop Request Item in Target Instance

1. In the target ServiceNow instance, search for Service Catalog in the application navigator.
2. Under Service Catalog, select Catalog.
3. Open the Hardware category and search for the “Laptop Request” item.
4. Click on the Laptop Request catalog item to open it.
5. You will see that only three variables are displayed**.**
6. As per our scenario, when the user selects the **“Additional Accessories”** checkbox, the **“Accessories Details”** field becomes visible. This field is also set to **mandatory**, ensuring that users must provide the required details.
7. On testing, the behavior works as expected — the form dynamically displays the field only when needed, fulfilling the project requirements.

****

**Fig 11: Laptop Request Form**

* + The final catalog item form for requesting a laptop is shown.
  + It contains:
    - Laptop Model field
    - Justification field
    - Accessories Details (made mandatory by the UI Policy)
  + Buttons: Order Now / Add to Cart

****

**Fig 12**

**Catalog**.

* **Form Fields Visible**:
  1. **Laptop Model** – A text field where the employee specifies the required laptop model.
  2. **Justification** – A multi-line field for the employee to explain why the laptop is needed (e.g., new hire, replacement, project work).
  3. **Accessories Details** – A mandatory field (enforced by the **UI Policy**) that appears only if the **“Additional Accessories”** checkbox is selected.
* **Buttons at the Bottom**:
  1. **Order Now** → Directly submits the request.
  2. **Add to Cart** → Allows the user to add this request to a cart along with other items before submitting.
* **Dynamic Behavior**:
  1. The **Accessories Details** field is not always visible.
  2. It only appears **when the checkbox for “Additional Accessories” is checked**, ensuring users provide details only when relevant.
  3. This reduces clutter and improves accuracy.

**CONCLUSION**

The Laptop Request Catalog Item Project has successfully streamlined the laptop request process within the organization by utilizing ServiceNow’s Service Catalog capabilities. With the introduction of a dynamic catalog item, the solution provides users with an intuitive and user-friendly interface that minimizes errors and improves overall efficiency.

This project highlights how ServiceNow can transform manual, error-prone procedures into automated, reliable, and user-centric solutions. Beyond improving service delivery, it also enhances employee satisfaction by offering a modern, guided, and seamless request experience.