

# LAPTOP REQUEST CATALOG ITEM

**Team ID: NM2025TMID09128**

**Team size:4**

**Team Leader: AANANTHA SREE M**

**Team member : ABBIRAMI K S**

**Team member : ABISHA P S**

**Team member : ASWINI U R**

## Problem Statement

Employees need a fast, reliable way to request laptops. The existing process is manual, slow, and error-prone: forms aren't dynamic to guide users, required information is often missed, there's no simple reset option, and changes aren't tracked for controlled deployment.

## Objective

To streamline and automate the laptop request process by creating a dynamic, user-friendly Service Catalog item in ServiceNow.

To eliminate manual, error-prone methods by implementing guided forms, conditional logic (UI policies), and reset functionality.

To ensure all configuration changes are properly tracked and transferable between environments through update sets.

To improve efficiency, accuracy, and governance in laptop provisioning requests.

## Skills Demonstrated

### 1. ServiceNow Configuration

Creating and managing Service Catalog Items.

Designing and adding variables (form fields) for structured data collection.

### 2. UI Policies & Policy Actions

Implementing dynamic form behavior (show/hide fields, mandatory conditions).

### 3. UI Actions (Client-side Scripting)

Writing client-side scripts for enhanced usability (e.g., reset form button).

### 4. Update Set Management

Capturing, exporting, importing, and committing update sets for controlled deployment.

### 5. Testing & Validation

End-to-end testing of catalog items and workflows to ensure correct behavior and user experience.

### 6. Problem-Solving & Process Automation

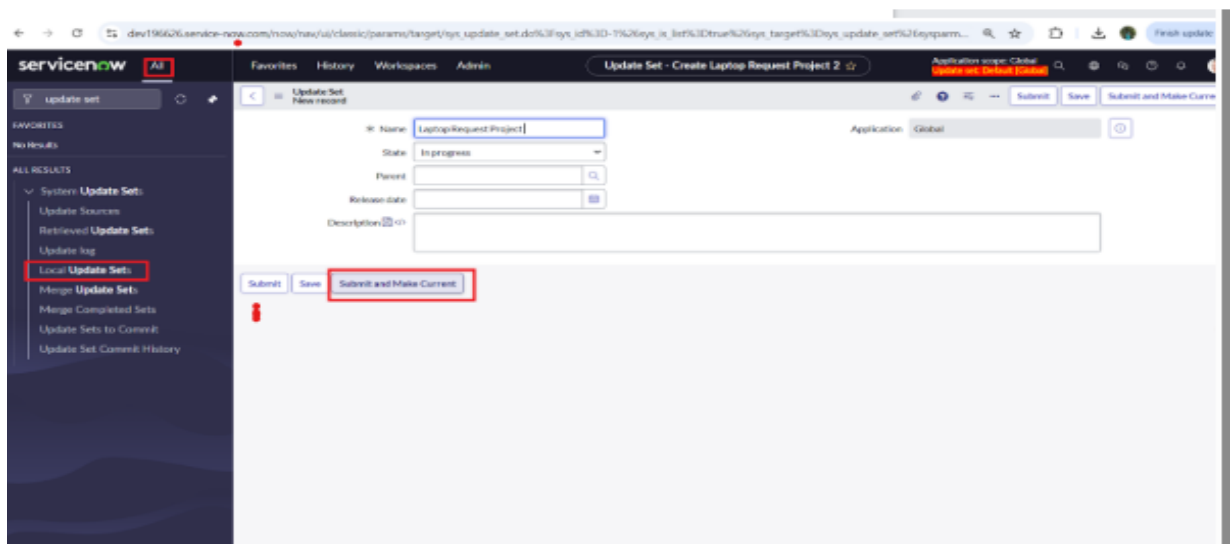
Analyzing manual processes and transforming them into efficient, automated solutions.

## Update set

### Step-by-step process

#### Create a local update set (capture all changes)

1. All → System Update Sets → Local Update Sets → New.
2. Name: Laptop Request (or Laptop Request Project). Add description if needed.



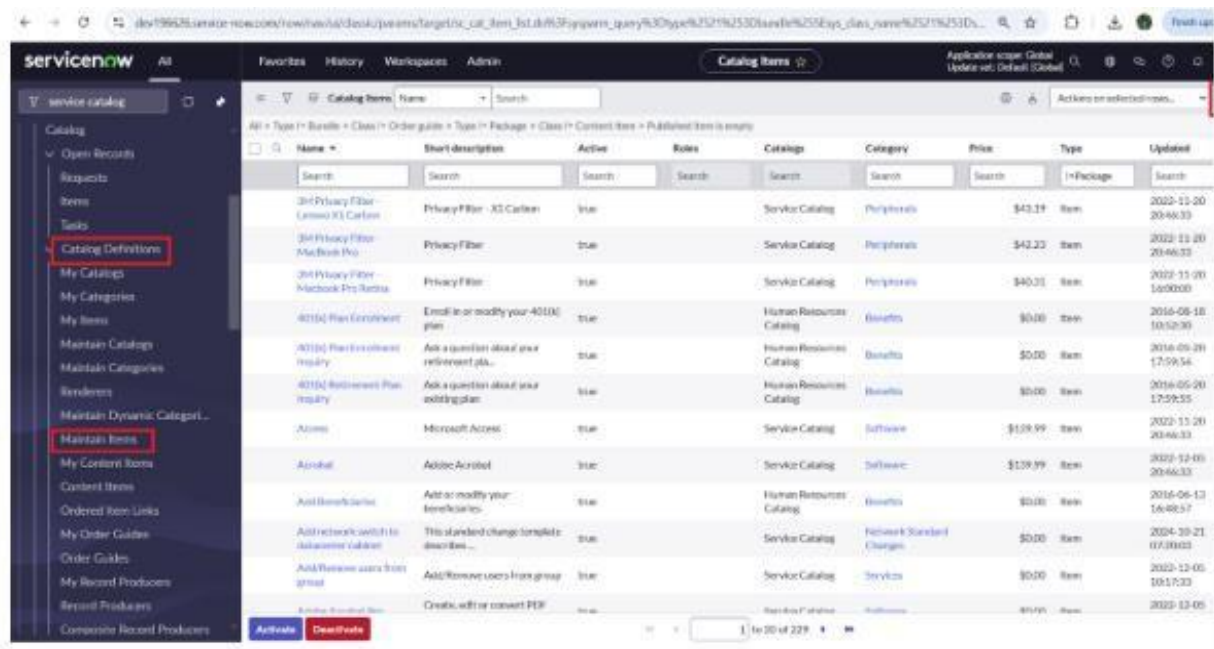
3. Click Submit, then click Make Current.

> Perform every subsequent action while this update set is current so all changes are recorded.

## Service Catalog Item

### Create the Service Catalog item

1. All → Service Catalog → Catalog Definitions → Maintain Items → New.



2. Fill in:

Name: Laptop Request Catalog:

Service Catalog Category:

Hardware

Short description: Use this item to request a new laptop

ServiceNow Developers | New Record | Catalog Item | Student

dev215933.service-now.com/now/nav/ui/classic/params/target/sc\_cat\_item.do%3Fsys\_id%3D-1%26sys\_is\_list%3Dtrue%26sys\_target%3Dsc\_cat\_item%26sysparm\_checke...

**service now** All

service cat

Catalog Definitions

- My Catalogs
- My Categories
- My Items
- Maintain Catalogs
- Maintain Categories
- Renderers
- Maintain Dynamic Categori...
- Maintain Items
- My Content Items
- Content Items
- Ordered Item Links
- My Order Guides
- Order Guides
- My Record Producers
- Record Producers
- Composite Record Producers
- User Criteria
- Maintain Cart Layouts
- Catalog Administration
- Service Catalog Overview**
- Service Fulfillment Steps R...
- Service Fulfillment Steps C...

Favorites History Workspaces Admin

Catalog Item - New Record

Search

Submit Try It

Enter a Price, approvals, variables, and other information as needed.

Name: Laptop Request

Application: Global

Active: ☒

Fulfillment automation level: Unspecified

Catalogs: Service Catalog

Select target record

Category: Hardware

State: -- None --

Checked out: -- None --

Owner: System Administrator

Item Details Process Engine Picture Pricing Portal Settings

Short description: Use this item to request a new laptop

Description

B I U ↶ ↷ Verdana 8pt

1:36 PM 9/2/2025

3. Click Save.

## Add variables (form fields)

On the saved catalog item, use the Variables related list → New for each variable:

### Variable 1

Label: Laptop Model

Type: Single line text

Name: laptop\_model

Order: 100 → Submit

ServiceNow Variable - New Record

Application: Global

Type: Single Line Text

Catalog Item: Laptop Request

Order: 100

Active: ☒

Mandatory: ☐

Read only: ☐

Hidden: ☐

Question

Specify the Question that explains the options available to the end user when ordering the item

\* Question: Laptop Model

\* Name: laptop\_model

Conversational label:

Tooltip:

Example Text:

Submit

## Variable 2

Label: Justification Type:

Multi-line text Name:

justification

Order: 200 → Submit

## Variable 3

Label: Additional Accessories

Type: Checkbox

Name: additional\_accessories

Order: 300 → Submit

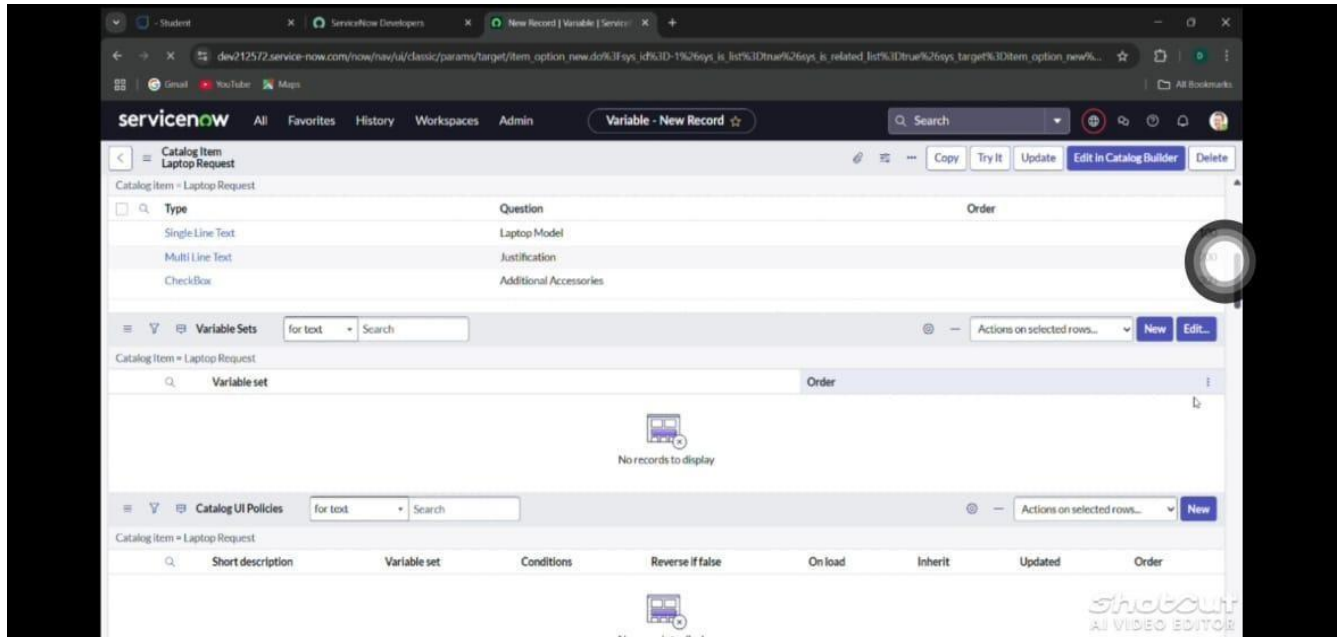
## Variable 4

Label: Accessories Details Type:

Multi-line text

Name: accessories\_details

Order: 400 → Submit

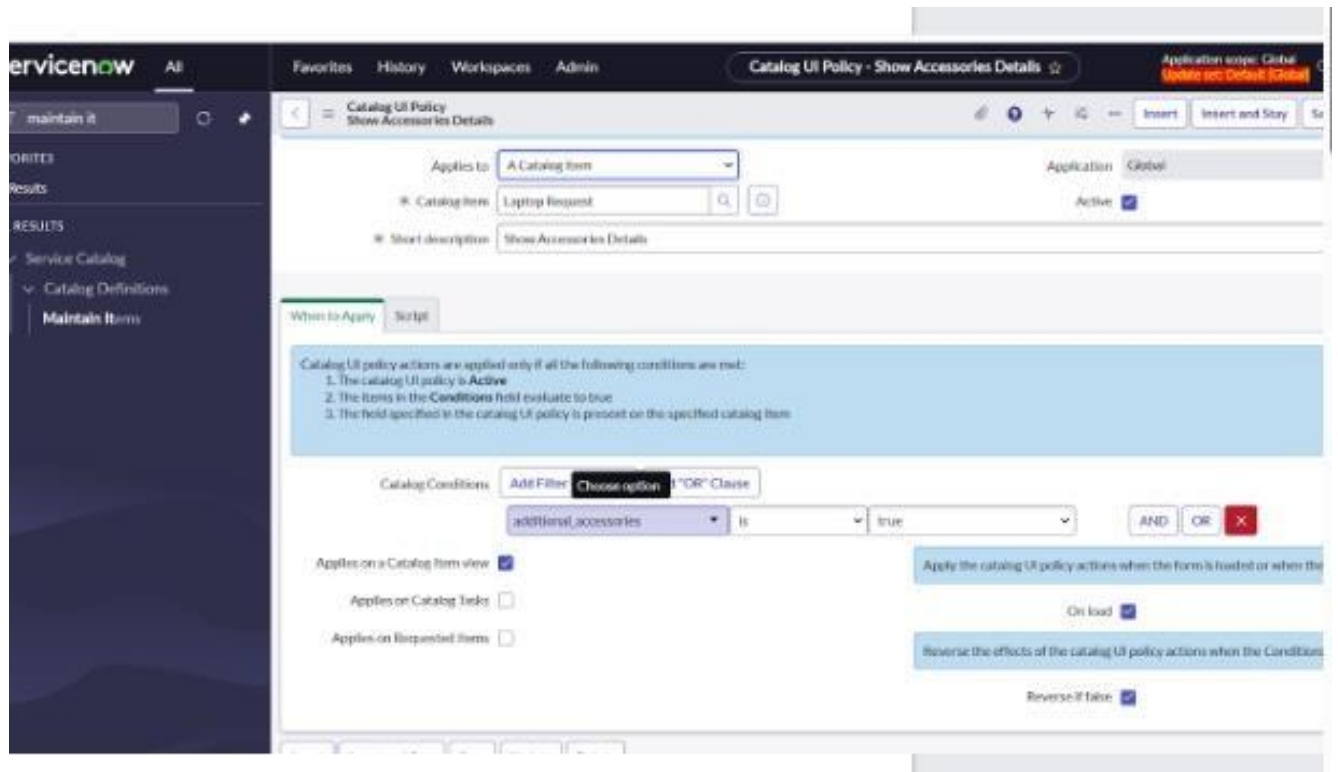


After adding variables, Save the catalog item form.

## UI Policy

### Create Catalog UI Policy (show + mandatory behavior)

1. On the catalog item form, scroll to Catalog UI Policies → New.
2. Short description: Show accessories details.
3. When to apply (Catalog Condition): set Field additional\_accessories is true.
4. Click Save (do not click Submit if your process prefers Save).



5. In the same policy, open Catalog UI Policy Actions → New and set:

Variable name: accessories\_details Order:

100

Mandatory: True

Visible: True → Save

6. Click Save again on the Catalog UI Policy record.

UI policy actions specify exactly what actions to take on a specified field. The conditions specified in the UI policy determine when these actions are triggered. [More Info](#)

Catalog Item	Laptop Request	Application	Global
Variable name	additional_accessories	Mandatory	True
Order	100	Visible	True
		Read only	Leave alone
		Value action	Leave alone
		Field message type	None

Submit

## Result:

when the user checks Additional Accessories, Accessories Details becomes visible and mandatory.

## UI Action

### Create UI Action (Reset form button)

1. All → System Definition → UI Actions → New.

2. Fill:

Table: sc\_cart (or sc\_cart\_item if you want it on the item) Order:

100

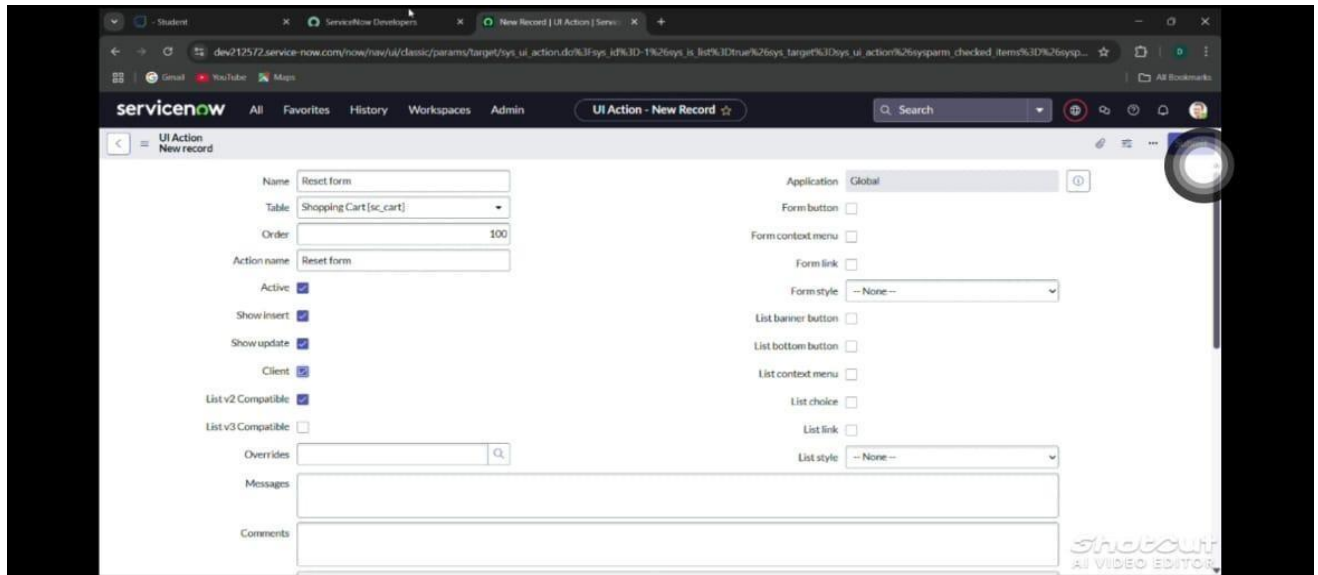
Action name / Label: Reset form Check

Client (so it runs client-side).

3. Script (client-side):

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



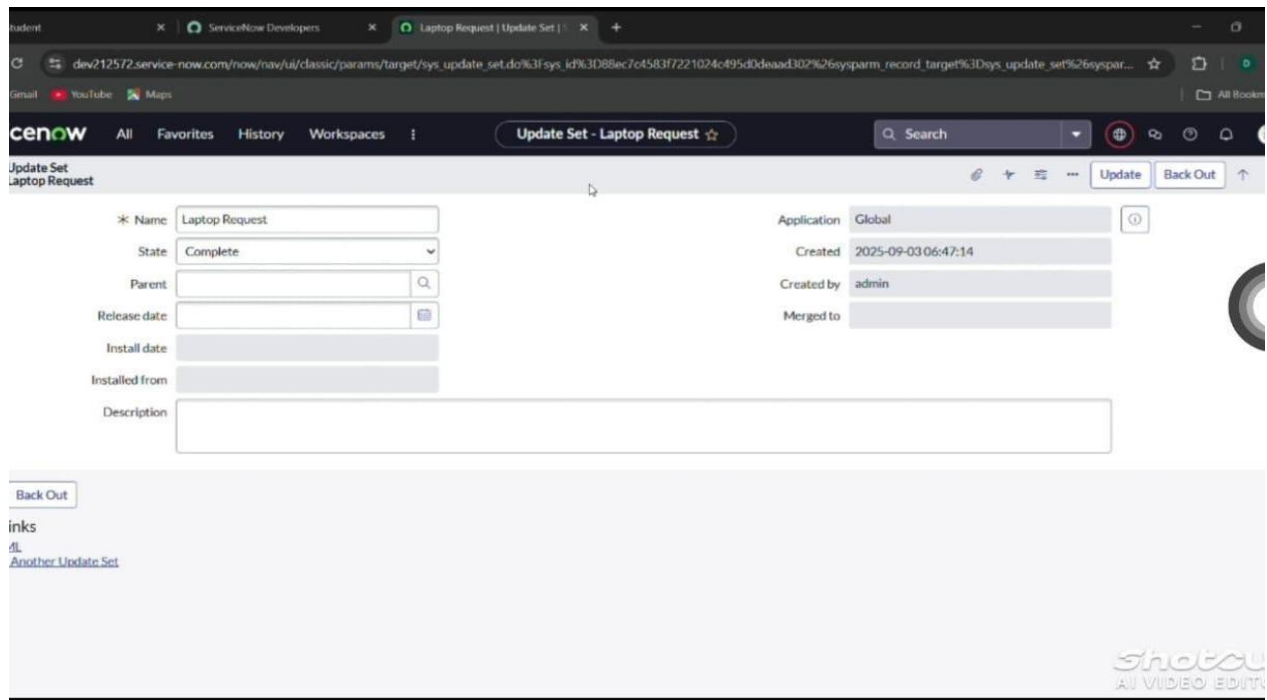


4. Click Save.

## Export Update Set

### Complete and export the update set

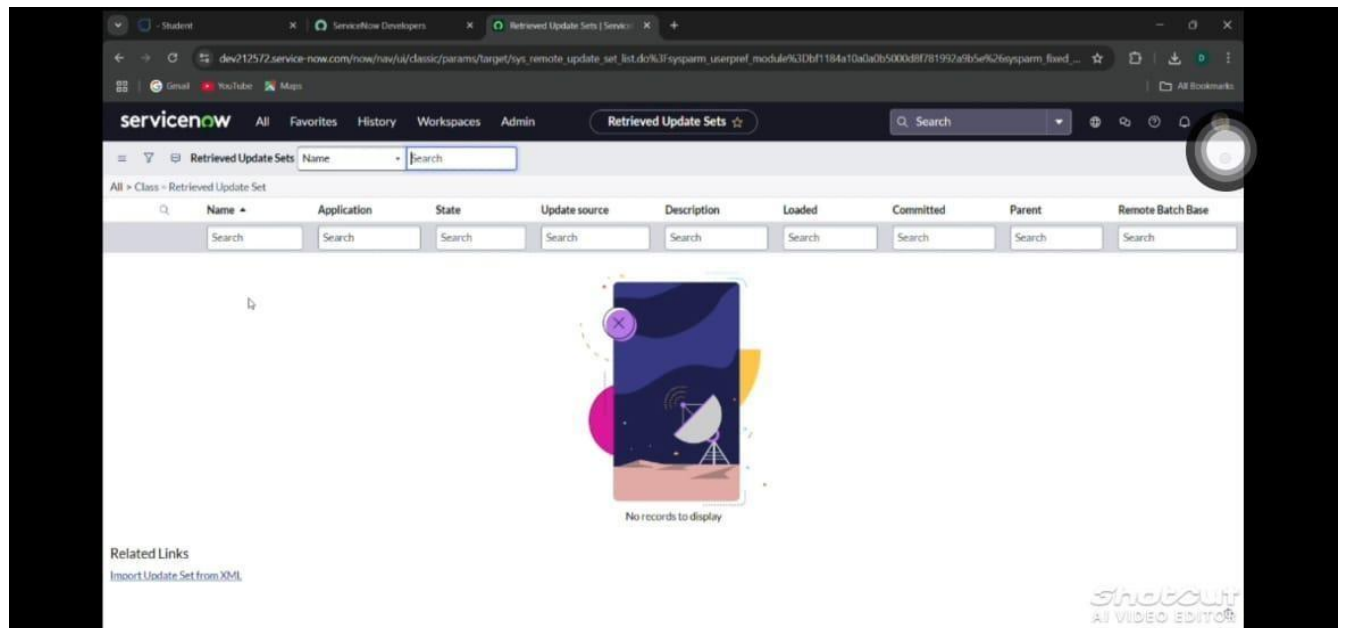
1. All → System Update Sets → Local Update Sets → open Laptop Request.
2. Change State to Complete.
3. Verify the Updates related list contains the catalog item, variables, UI policy, UI policy action, and UI action.
4. Click Export to XML and download the XML file.



## Login To Another Instance

### Import & commit update set in target instance

1. Log into target instance. All → System Update Sets → Retrieved Update Sets.
2. Click Import Update Set from XML, upload the XML file → Upload.



3. Open the retrieved update set → Preview Update Set (resolve any preview issues).



After committing this update set, ensure you map any unmapped custom tables to a subscription so that you can track your entitlements accurately in Subscription Management.

Name	Laptop Request	Committed	
Application	Global	Inserted	0
Update source		Updated	10
Parent		Deleted	0
State	Previewed	Collisions	0
Loaded	2025-09-03 07:33:16	Total	10
Description			
Application name	Global		

Update Delete Run Preview Again Commit Update Set

Related Links  
[Show All Preview Records](#)

Customer Updates (10) Child Update Sets

2. Verify fields: Laptop Model, Justification, Additional Accessories, Accessories Details (hidden initially).

3. Test behavior:

Leave Additional Accessories unchecked → accessories\_details stays hidden. Check Additional Accessories → accessories\_details appears and is mandatory.

4. Add the item to cart and verify Reset form works (clears fields) in the cart or item area where created.

Name	Application	State	Update source	Description	Loaded	Committed	Parent	Remote Batch Base
Laptop Request	Global	Previewed	(empty)		2025-09-03 07:33:16	(empty)	(empty)	(empty)

Related Links  
[Import Update Set from XML](#)

## Conclusion

Creating the Laptop Request ServiceNow catalog item replaces a slow, manual laptop request process with a tracked, dynamic, user-friendly form. Dynamic fields and UI policy logic ensure required details are captured only when relevant, the reset button improves usability, and using an update set guarantees changes are governed and portable between instances — resulting in faster requests, fewer errors, and better governance.