



## Laptop Request Catalog Item

**Team ID : NM2025TMID07340**

**Team Size : 4**

**Team Leader : SHALINI M**

**Team member : SANTHOSH R**

**Team member : SABHAREESAN G**

**Team member : SARAN R**

**Problem statement:** Employees in the organization need a quick and efficient way to request laptops for work. The current process is manual and prone to delays, with no dynamic form behavior to guide users or ensure accurate data collection. To address this, a Service Catalog item needs to be created, allowing users to easily request a laptop, with dynamic fields, clear instructions, and additional functionality like resetting the Form if needed. The solution should also ensure all changes are tracked for governance and deployment.

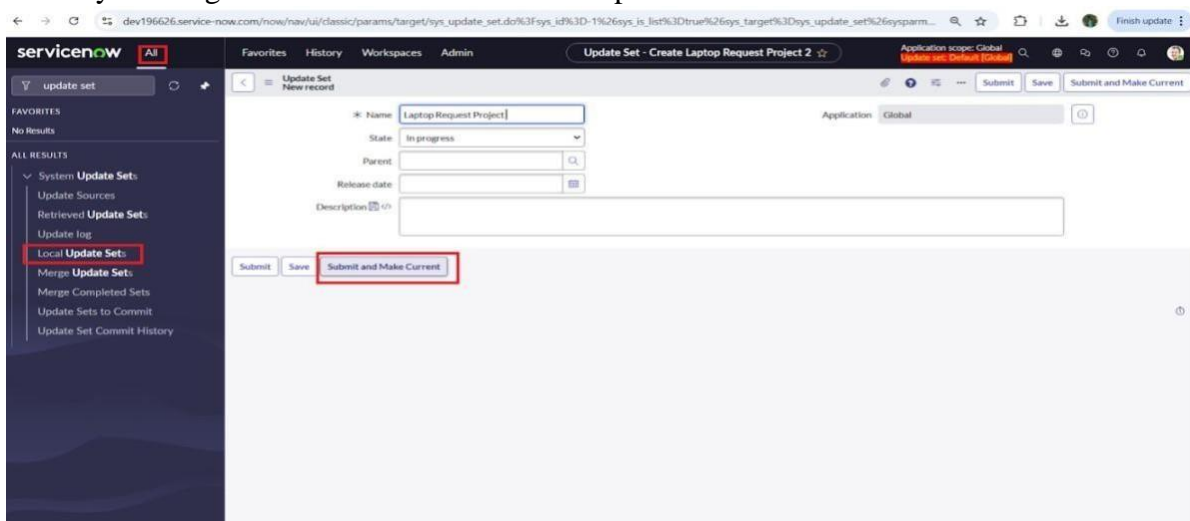
### TASK INITIATION

#### Milestone 1 : Update sets

Activity 1 : create Local Update Set

1. Open service now.
2. Click on All >> search for update sets
3. Select local update sets under system update sets
4. Click on new
5. Fill the following details to create a update set as: "Laptop Request"

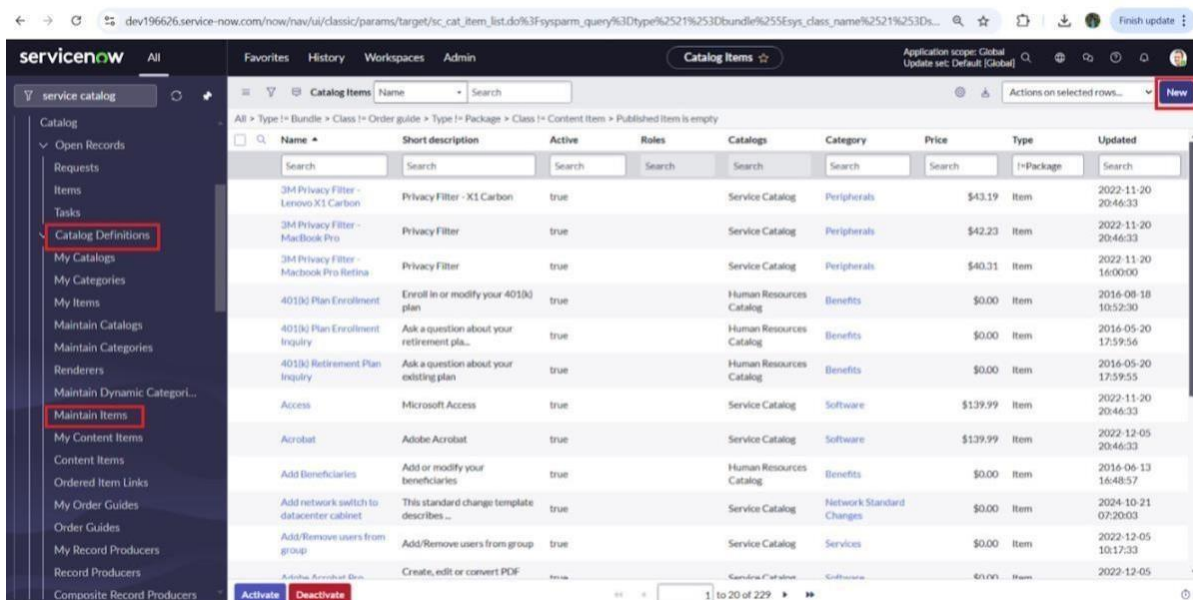
- Click on submit and make current
- By clicking on the button it activates the update set .



## Milestone 2: Service Catalog Item

### Activity: Create Service Catalog Item

- Open service now.
- Click on All >> service catalog
- Select maintain items under catalog definitions 4. Click on New.



- Fill the following details to create a new catalog item  
 Name: Laptop Request  
 Catalog: service Catalog  
 Category: Hardware  
 Short Description: Use this item to request a new laptop
- Click on „SAVE“

The screenshot shows the ServiceNow 'Catalog Item - Laptop Request' form. The left sidebar contains a navigation menu with 'Service Catalog' and 'Catalog Definitions' expanded. The main form area has a header with 'Catalog Item - Laptop Request' and a sub-header 'Laptop Request'. Below this, there's a description of catalog items. The form fields include:
 

- Name: Laptop Request (highlighted with a red box)
- Application: Global
- Category: Hardware (highlighted with a red box)
- State: --None--
- Checked out: --None--
- Owner: System Administrator
- Active: ☒
- Fulfillment automation level: Unspecified

 Below these fields are tabs for 'Item Details', 'Process Engine', 'Picture', 'Pricing', and 'Portal Settings'. The 'Item Details' tab is active, showing a 'Short description' field with the text 'Use this item to request a new laptop' (highlighted with a red box) and a 'Description' field with a rich text editor.

## Activity 2: Add Variable

### Step1:

- After saving the catalog item form scroll down and click on variable(related list)
- Click on new and enter the details as below
  1. Variable 1:Laptop Model
    - Type: Single line text
    - Name: laptop\_model
    - Order:100
- Click on submit
- Again click on new and add Remaining variables in the above process

The screenshot shows the ServiceNow 'Variable - New Record' form. The left sidebar contains a navigation menu with 'service catalog' and 'Catalog Definitions' expanded. The main form area has a header with 'Variable - New Record' and a sub-header 'New record'. Below this, there's a description of variables. The form fields include:
 

- Application: Global
- Type: Single Line Text (highlighted with a red box)
- Catalog Item: Laptop Request
- Order: 100
- Active: ☒
- Mandatory: ☐
- Read only: ☐
- Hidden: ☐

 Below these fields are tabs for 'Question', 'Annotation', 'Type Specifications', 'Default Value', 'Auto-populate', 'Permission', and 'Availability'. The 'Question' tab is active, showing a 'Specify the Question that explains the options available to the end user when ordering the item' section. This section contains:
 

- \* Question: Laptop Model (highlighted with a red box)
- \* Name: laptop\_model (highlighted with a red box)
- Tooltip: (empty field)
- Example Text: (empty field)

 At the bottom are 'Submit' and 'Save' buttons.

2. Variable 2:Justification Type:
  - Multi line text Name:
  - justification
  - Order:200

### 3. Variable 3:Additional Accessories

Type: Checkbox

Name: additional\_accessories

Order:300

### 4. Variable 4: Accessories Details

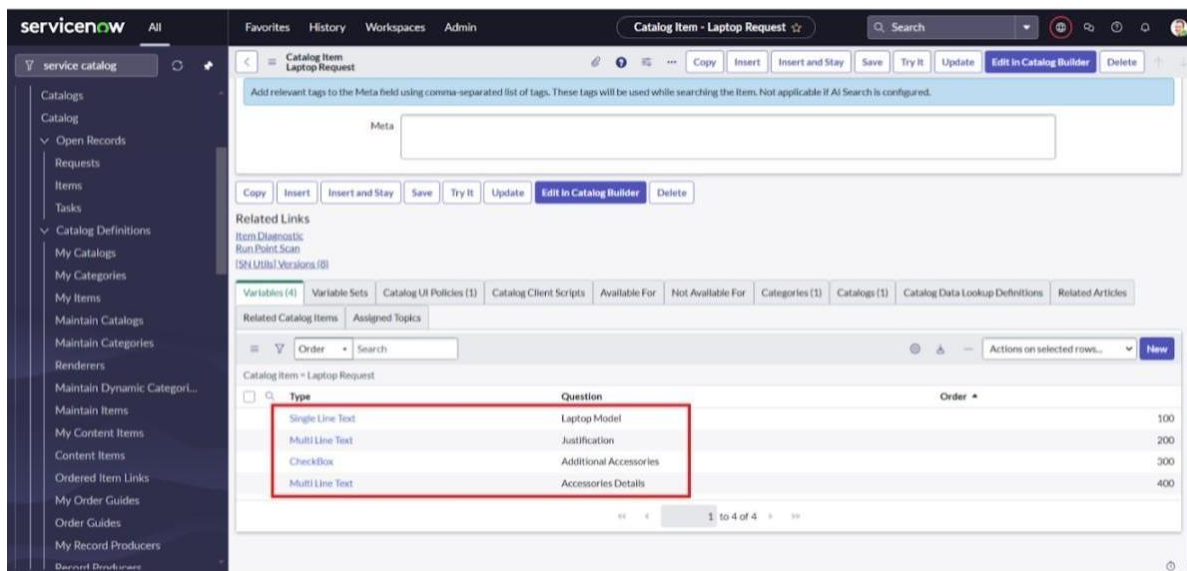
Type: Multi line text

Name:accessories\_details

Order:400

#### Step2:

- After adding above variable which are added to newly created catalog item
- Then save the catalog item form



## Milestone 3:UI POLICY

### Activity: Create Catalog Ui policies

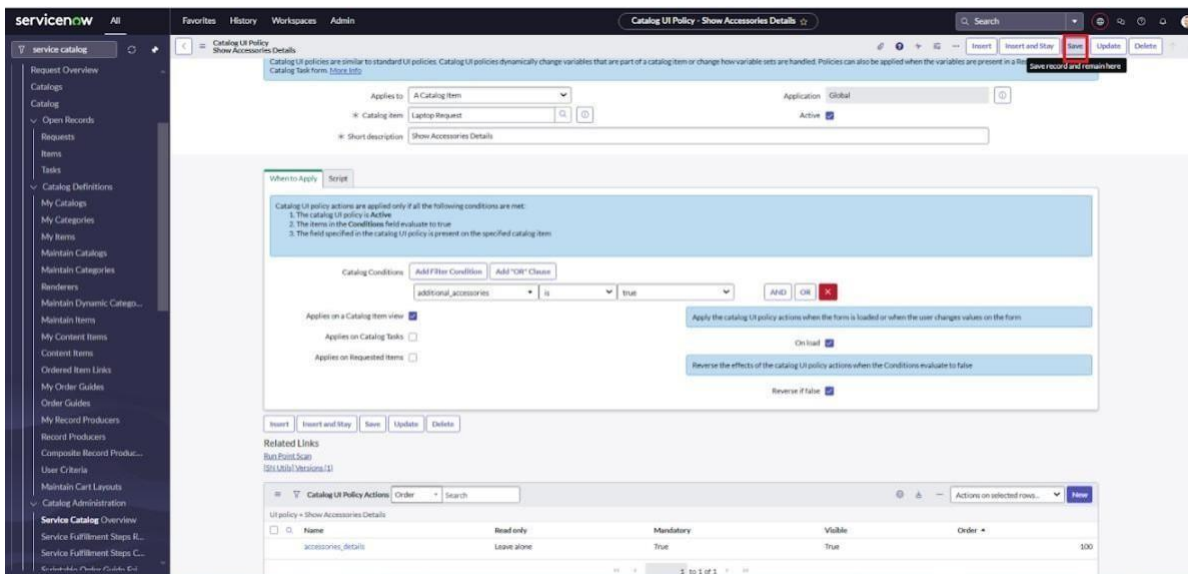
1. Click on all>> search for service catalog
2. Select maintain item under catalog definition
3. Search for „laptop request“ which is created before
4. Select „laptop request“ and scroll down click on “Catalog Ui policies”
5. In the catalog ui policies related list tab click on new
6. Give short description as: show accessories details

7. Set the Catalog Condition in the related list tab „when to apply“  
[field: additional\_ accessories, operator: is, value: true]

The screenshot shows the 'Catalog UI Policy - Show Accessories Details' configuration page. The 'When to Apply' tab is selected, showing a 'Script' section with instructions: 'Catalog UI policy actions are applied only if all the following conditions are met: 1. The catalog UI policy is Active, 2. The items in the Conditions field evaluate to true, 3. The field specified in the catalog UI policy is present on the specified catalog item'. Below this, the 'Catalog Conditions' section shows a filter for 'additional\_accessories' with the operator 'is' and value 'true'. The 'Applies on' section has checkboxes for 'Applies on a Catalog Item view' (checked), 'Applies on Catalog Tasks' (unchecked), and 'Applies on Requested Items' (unchecked). The 'Apply the catalog UI policy actions when the form is loaded or when the user changes values on the form' checkbox is checked. The 'Reverse the effects of the catalog UI policy actions when the Conditions evaluate to false' checkbox is also checked.

8. Click on **save**.(do not click on submit)
9. Scroll down and select „catalog ui action“
10. Then click on new button
11. Select variable name as: accessories\_details Order:100  
Mandatory: True  
Visible : True
12. Click on save and again click save button of the catalog ui policy form

The screenshot shows the 'Catalog UI Policy Action - accessories\_details' configuration page. The 'Catalog Item' is 'Laptop Request'. The 'Variable name' is 'accessories\_details' and the 'Order' is '100'. The 'Application' is 'Global'. The 'Mandatory' checkbox is checked, and the 'Visible' checkbox is also checked. The 'Read only' checkbox is unchecked, and the 'Value action' is 'Leave alone'. The 'Field message type' is 'None'. The 'Save' button is highlighted with a red box. The 'Related Links' section shows 'Run Point Scan' and 'SN Utils Versions (1)'.



## Milestone 4: UI ACTION

### Activity 1: Create Ui Action

1. Open service now.
2. Click on All >> search for ui action
3. Select ui actions under system definition
4. Click on new
5. Fill the following details to create ui action

Table: shopping cart(sc\_cart)

Order:100

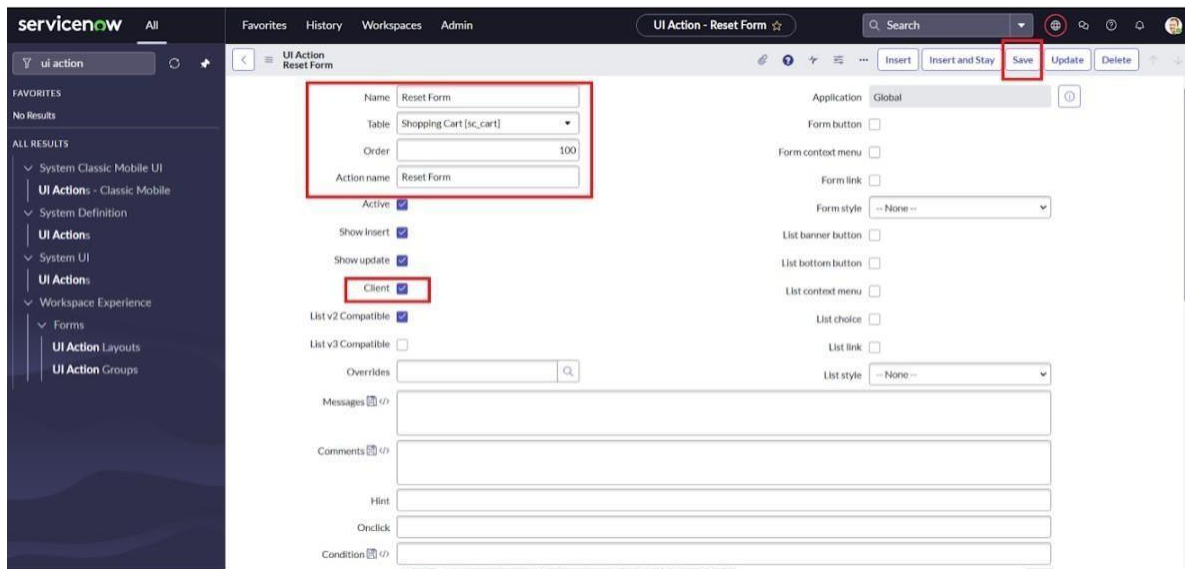
Action name: Reset form

Client : checked

Script:

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

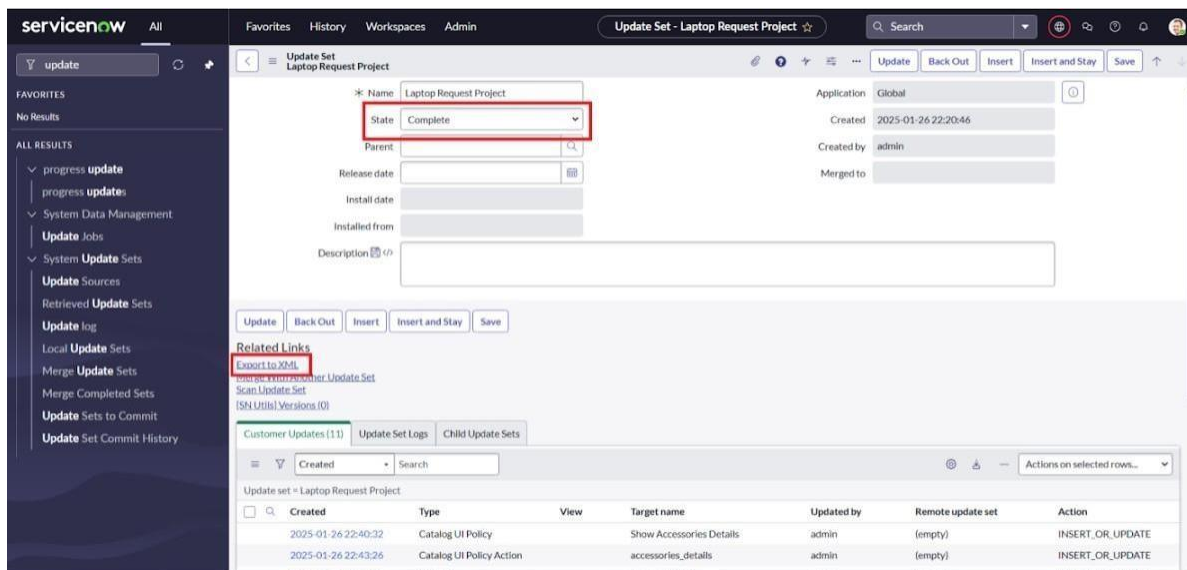
Click on save



## Milestone 5: Export Update set

### Activity: Exporting changes to another instances

1. Click on All >> search for update sets
2. Select local update set
3. Select created update set i.e. 'Laptop Request Project'
4. Set the state to 'Complete'
5. In the related list Update tab, updates are visible which we perform under this update set.
6. Click on export to XML ,it download one file

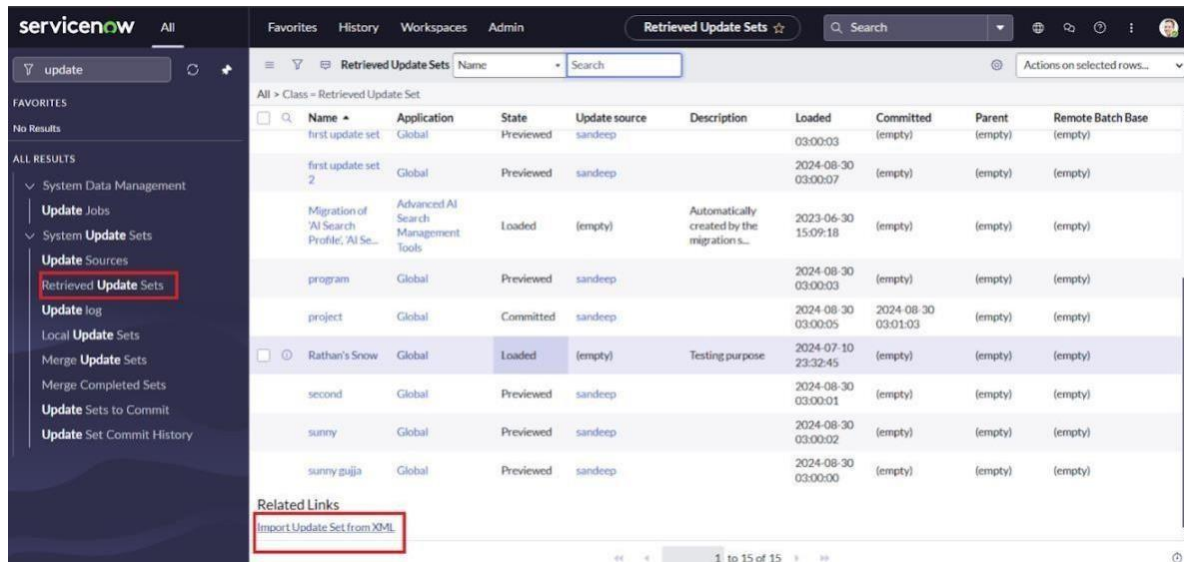


## Milestone 6: Login to another Instance

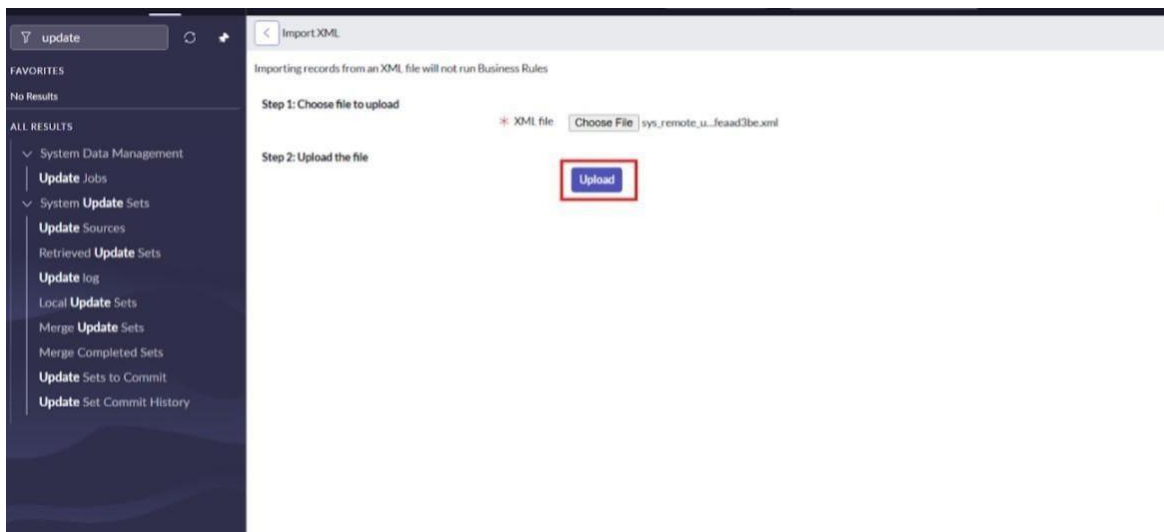
### Activity 1: Retrieving the update set



1. Open another instance in incognito window
2. Login with credentials
3. Click on all>> search for update sets
4. Select “Retrieved update set” under system update set
5. It open retrieved update set list and scroll down
6. Click on Import update set from XML

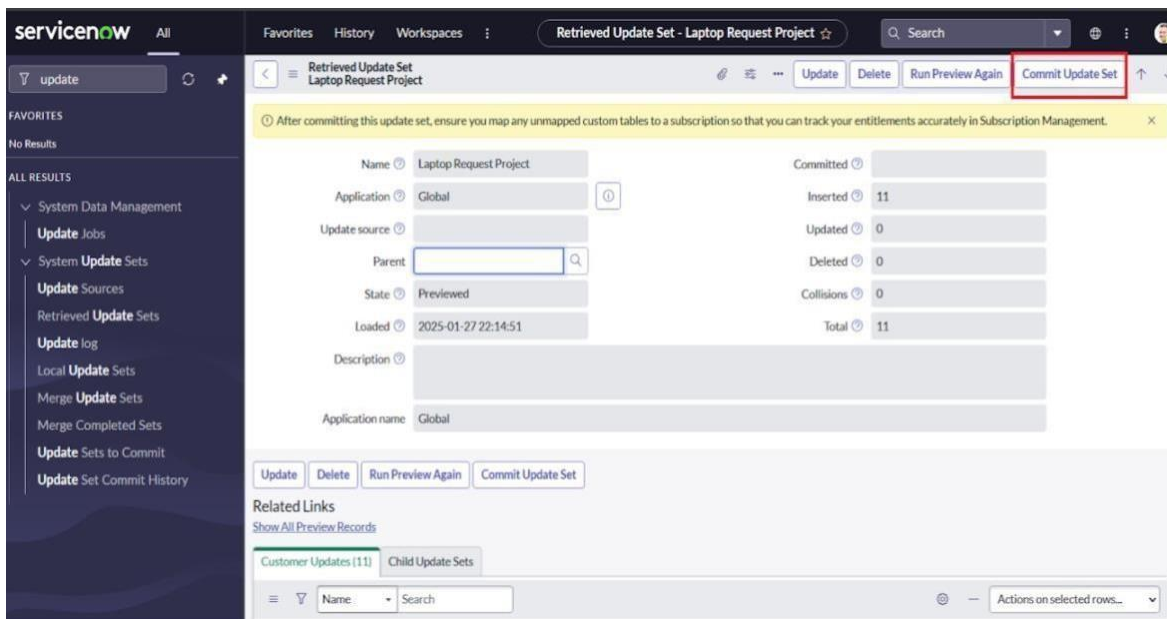


7. Upload the downloaded file in XML file
8. Click on Upload and it gets uploaded.



9. Open retrieved update set 'laptop request project
10. Click on preview update set
11. And click on commit update set
12. And also see the related tab updates
13. After committing **update** set in this instance we get all updates which are done in the previous instance

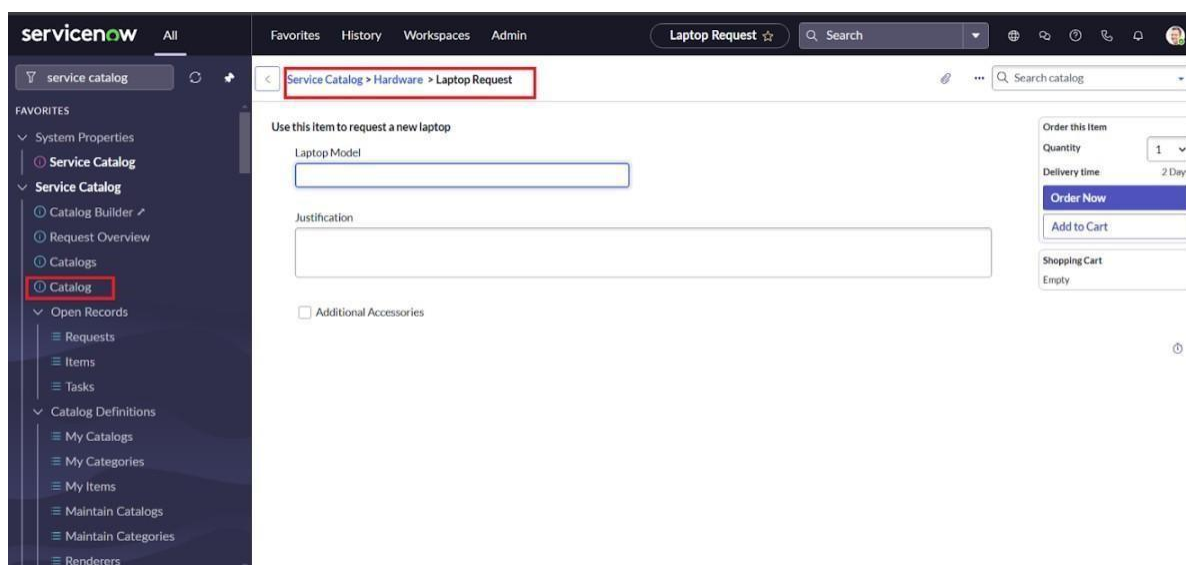




## Milestone 7: Testing

### Activity 1: Test Catalog Item

1. Search for service catalog in application navigator in target instance
2. Select catalog under service catalog
3. Select hardware category and search for 'laptop request' item



4. Select laptop request item and open it
5. It shows three variables only
6. As per our scenario, when we click on additional accessories checkbox then accessories details fields is visible and that should be mandatory
7. Now see the results, it fulfills our requirements.

The screenshot displays the ServiceNow interface for a 'Laptop Request' catalog item. The left sidebar shows the navigation menu with 'Service Catalog' expanded. The main form area is titled 'Use this item to request a new laptop'. It contains a 'Laptop Model' field with 'hp' entered, a 'Justification' text area, and a checkbox for 'Additional Accessories' which is checked. Below this is an 'Accessories Details' section, also highlighted with a red box. The right sidebar shows the 'Order this Item' section with a 'Quantity' dropdown set to 1, a 'Delivery time' of 2 Days, and buttons for 'Order Now' and 'Add to Cart'. The 'Shopping Cart' section at the bottom right indicates it is 'Empty'.

## Conclusion :

The Laptop Request Catalog Item project successfully streamlines the process of requesting laptops within the organization by leveraging ServiceNow's Service Catalog capabilities. Through the implementation of a dynamic catalog item, the project ensures that users have an intuitive and user-friendly interface, reducing errors and improving efficiency. This project demonstrates how ServiceNow can be used to replace manual, error-prone processes with automated, efficient, and user-centric solutions. It not only improves service delivery but also enhances employee satisfaction by providing a modern and streamlined request experience.