**LAPTOP REQUEST CATALOG ITEM**

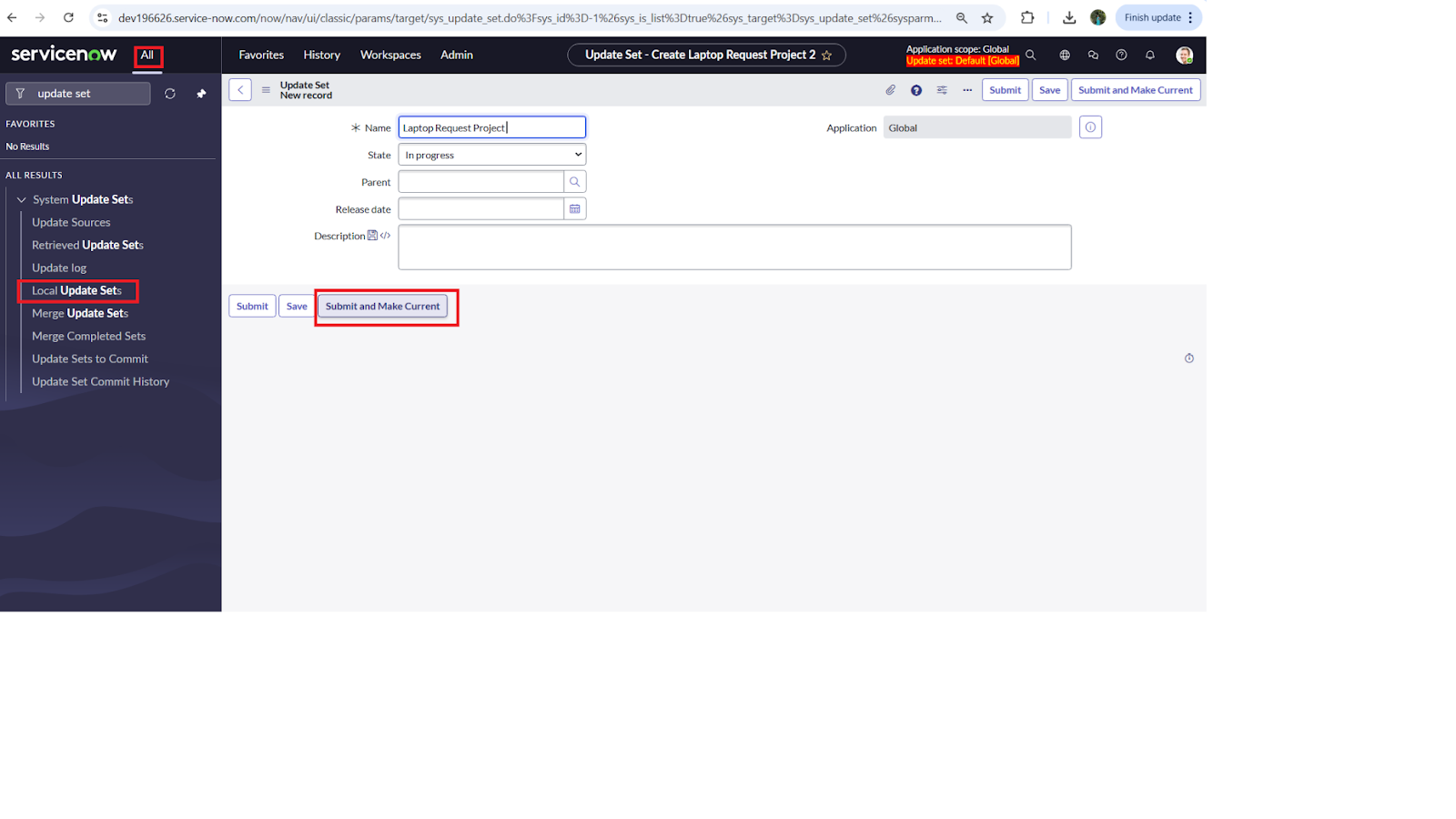
**Problem Statement:**

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.

### Update Set

**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”
6. Click on submit and make current
7. By clicking on the button it activates the update set .

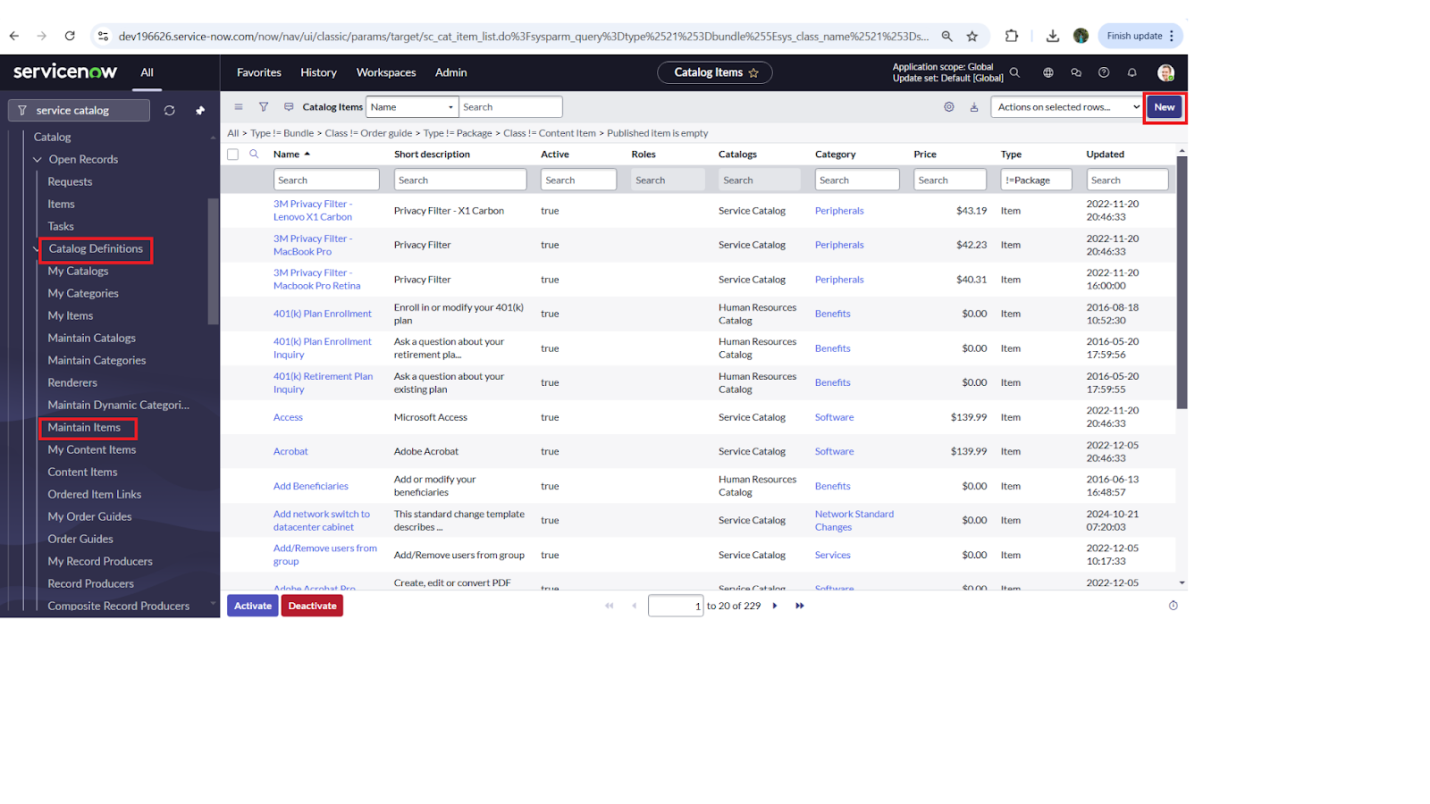


**NOTE: Perform all actions under this newly created update set only.**

### Service Catalog Item

**Create Service Catalog Item**

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



1. Fill the following details to create a new catalog item

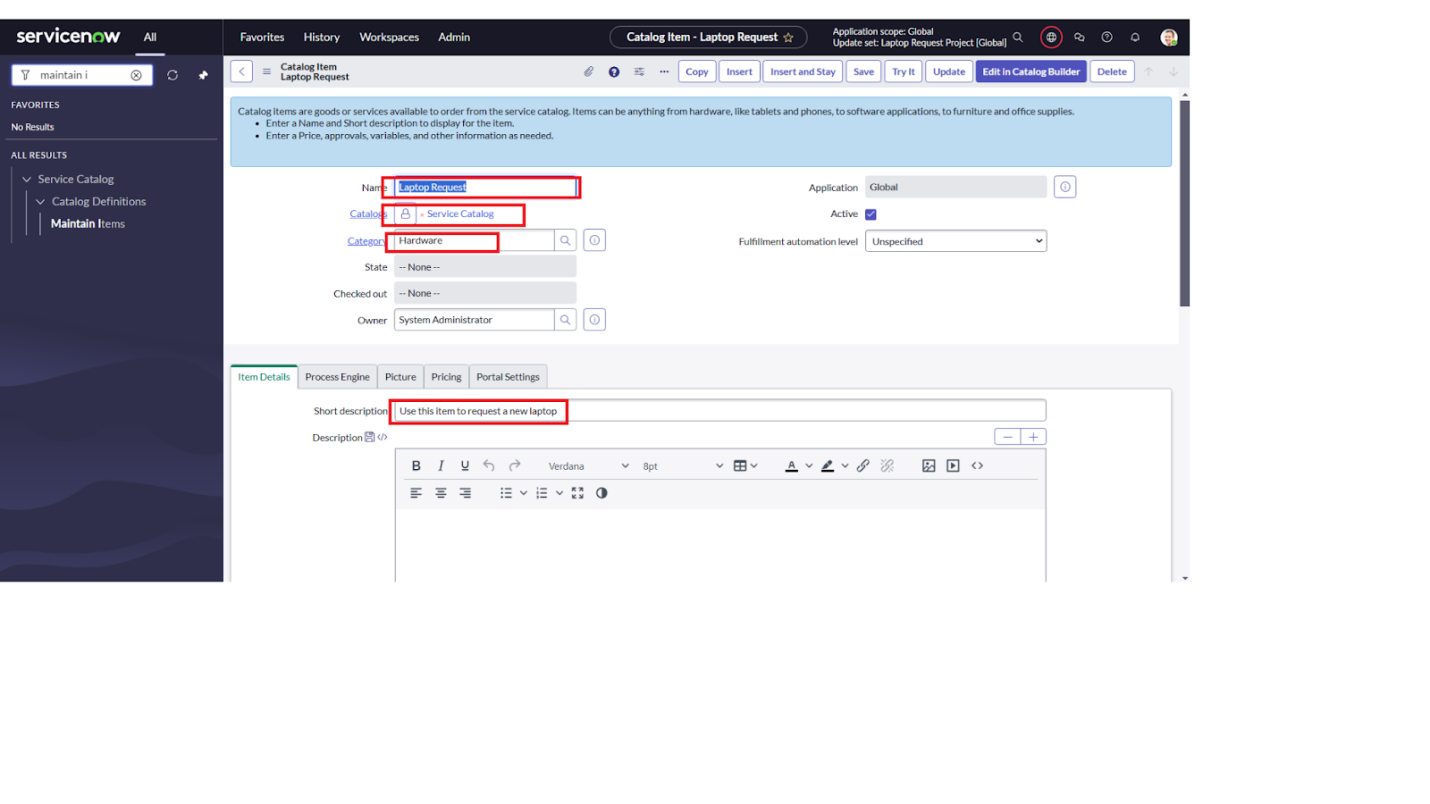
            Name: Laptop Request

            Catalog: service Catalog

            Category: Hardware

            Short Description: Use this item to request a new laptop

1. Click on ‘SAVE’



**Add Variables**

**Step1:**

* After saving the catalog item form scroll down and click on variable(related list)
* Click on new and enter the details as below

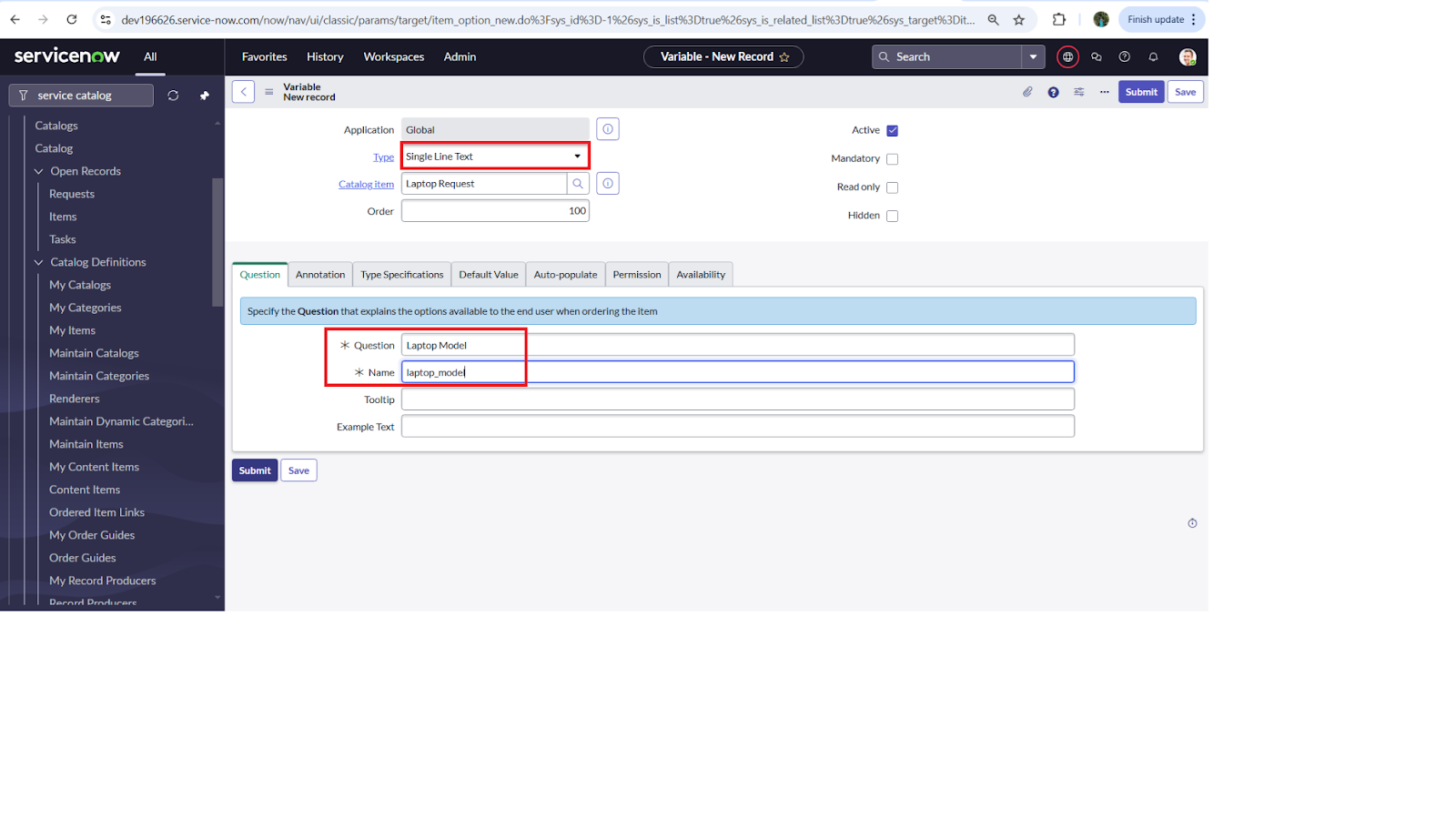
1. Variable 1:Laptop Model

             Type: Single line text

             Name: laptop\_model

             Order:100

* Click on submit
* Again click on new and add Remaining variables in the above process



     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

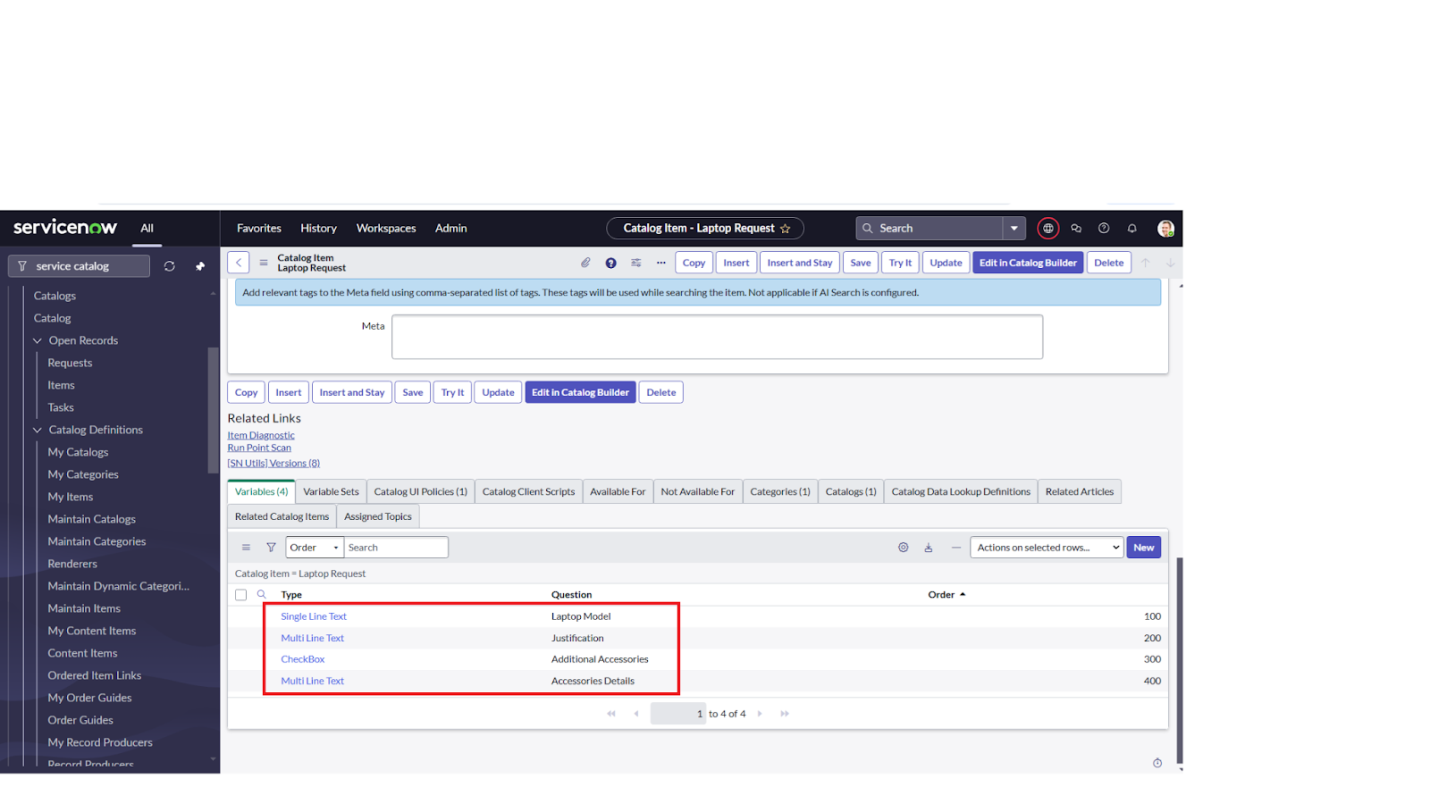
             Type: Multi line text

             Name:accessories\_details

             Order:400

**Step2:**

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

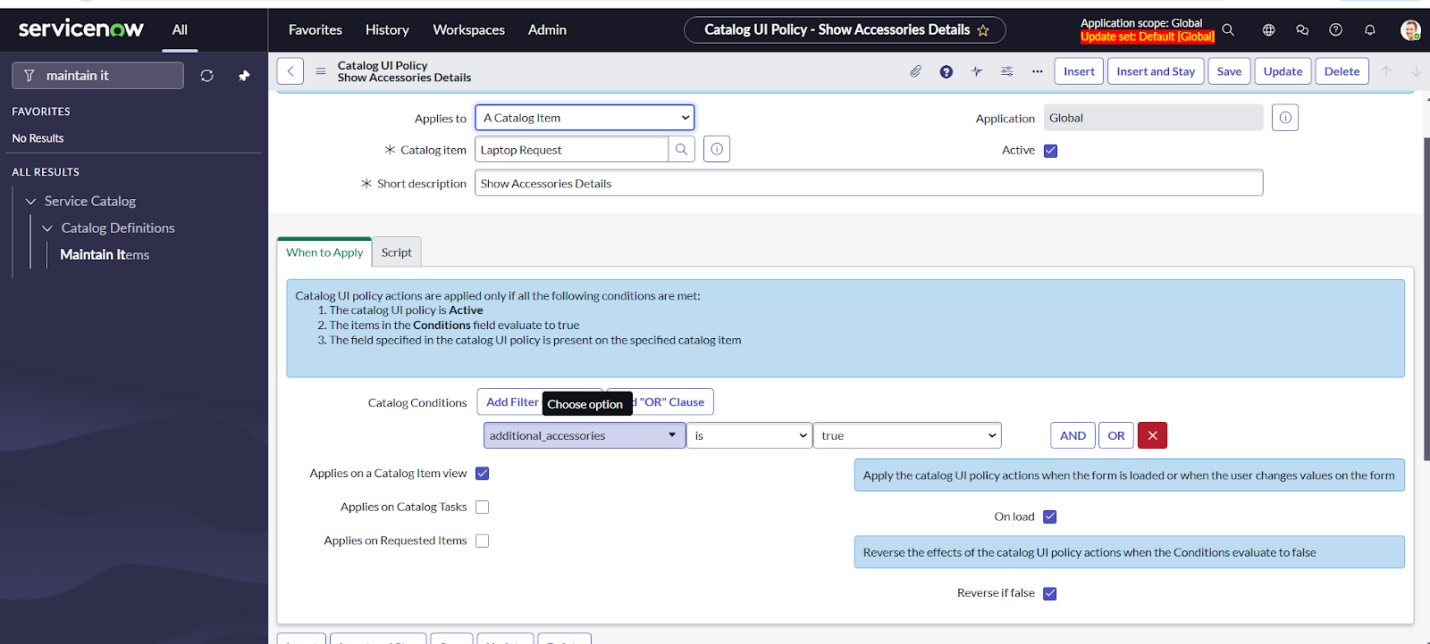


### UI Policy

**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]



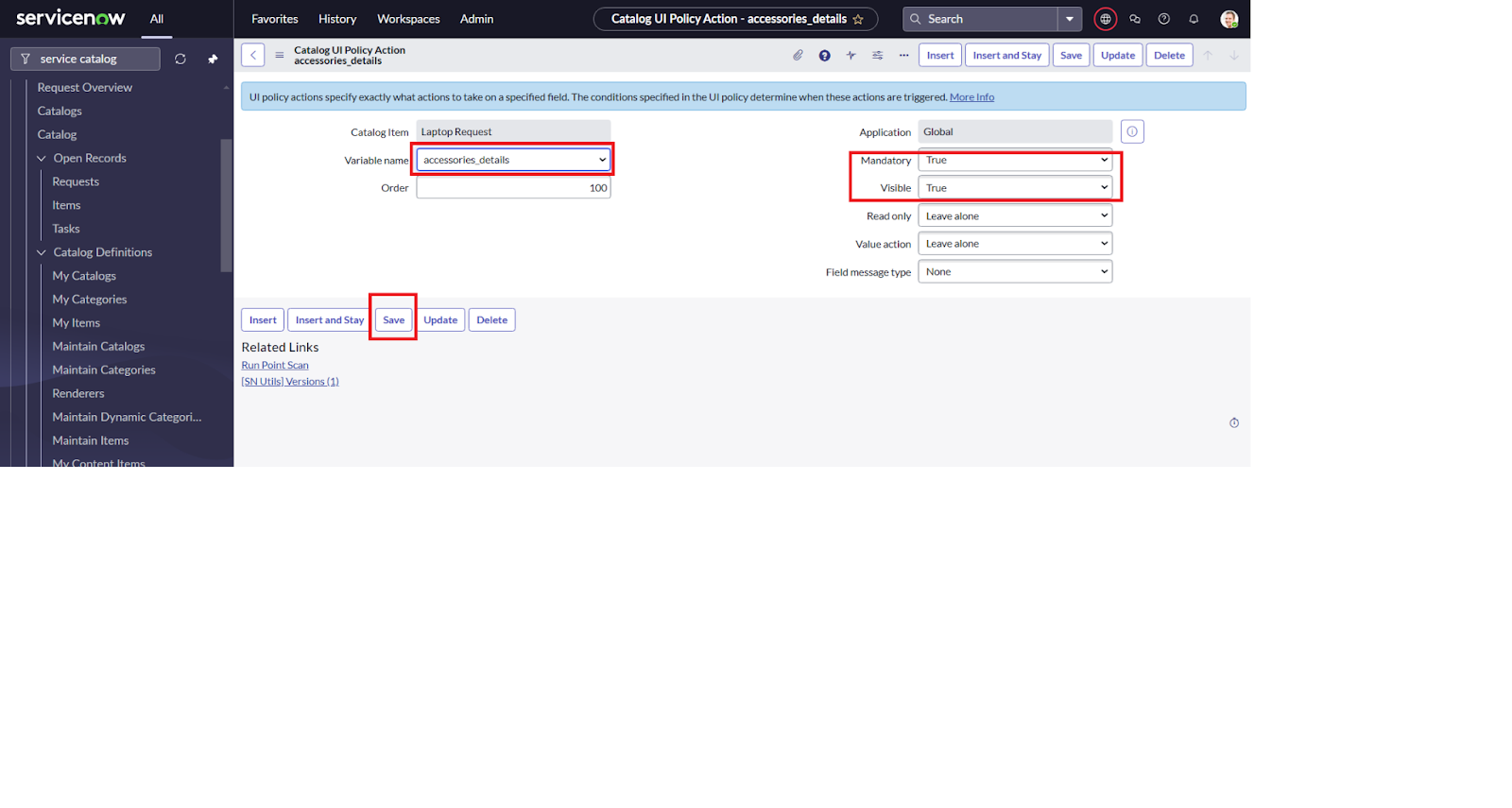
1. Click on **save**.(do not click on submit)
2. Scroll down and select ‘catalog ui action’
3. Then click on new button
4. Select variable name as: accessories\_details

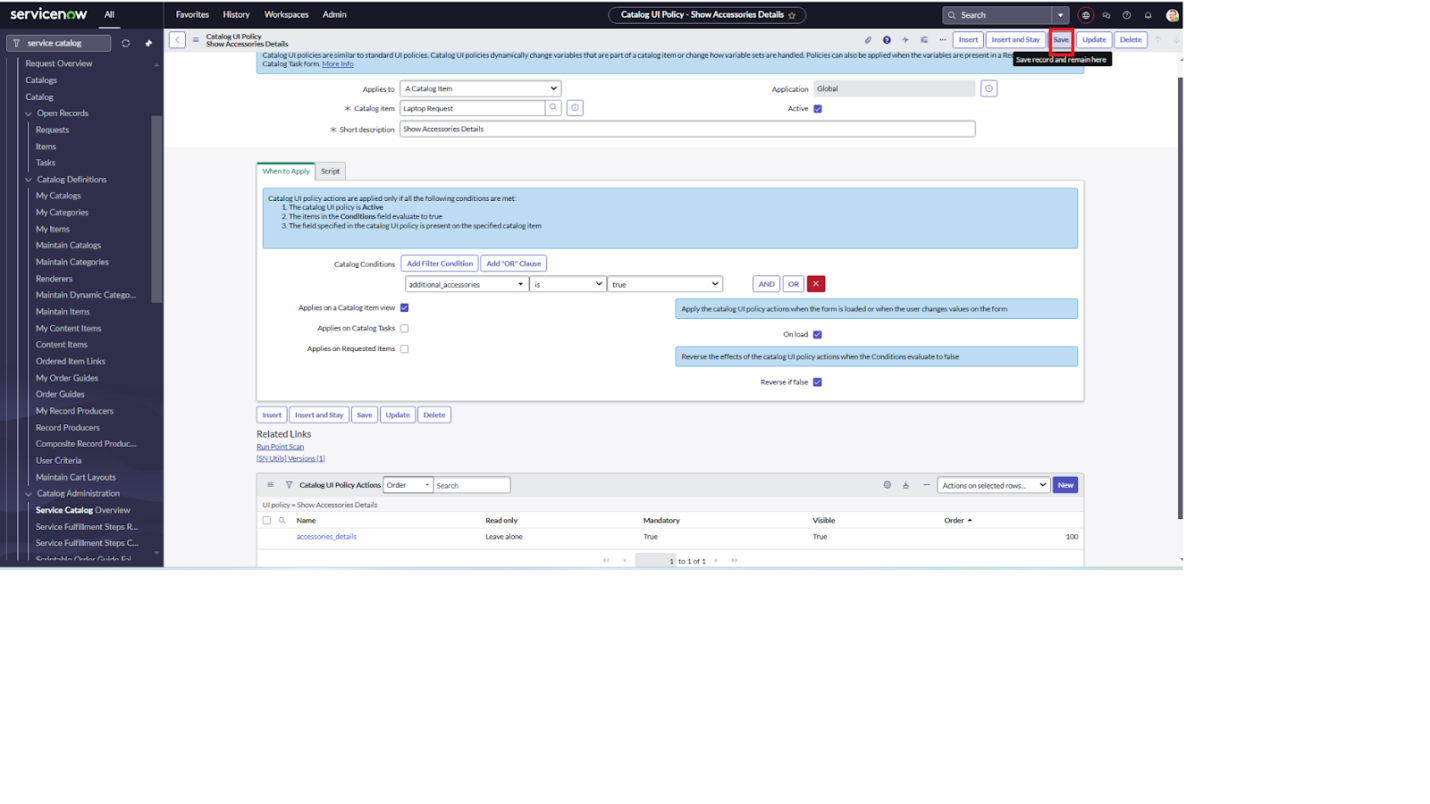
                              Order:100

                    Mandatory: True

                         Visible : True

1. Click on save and again click save button of the catalog ui policy form



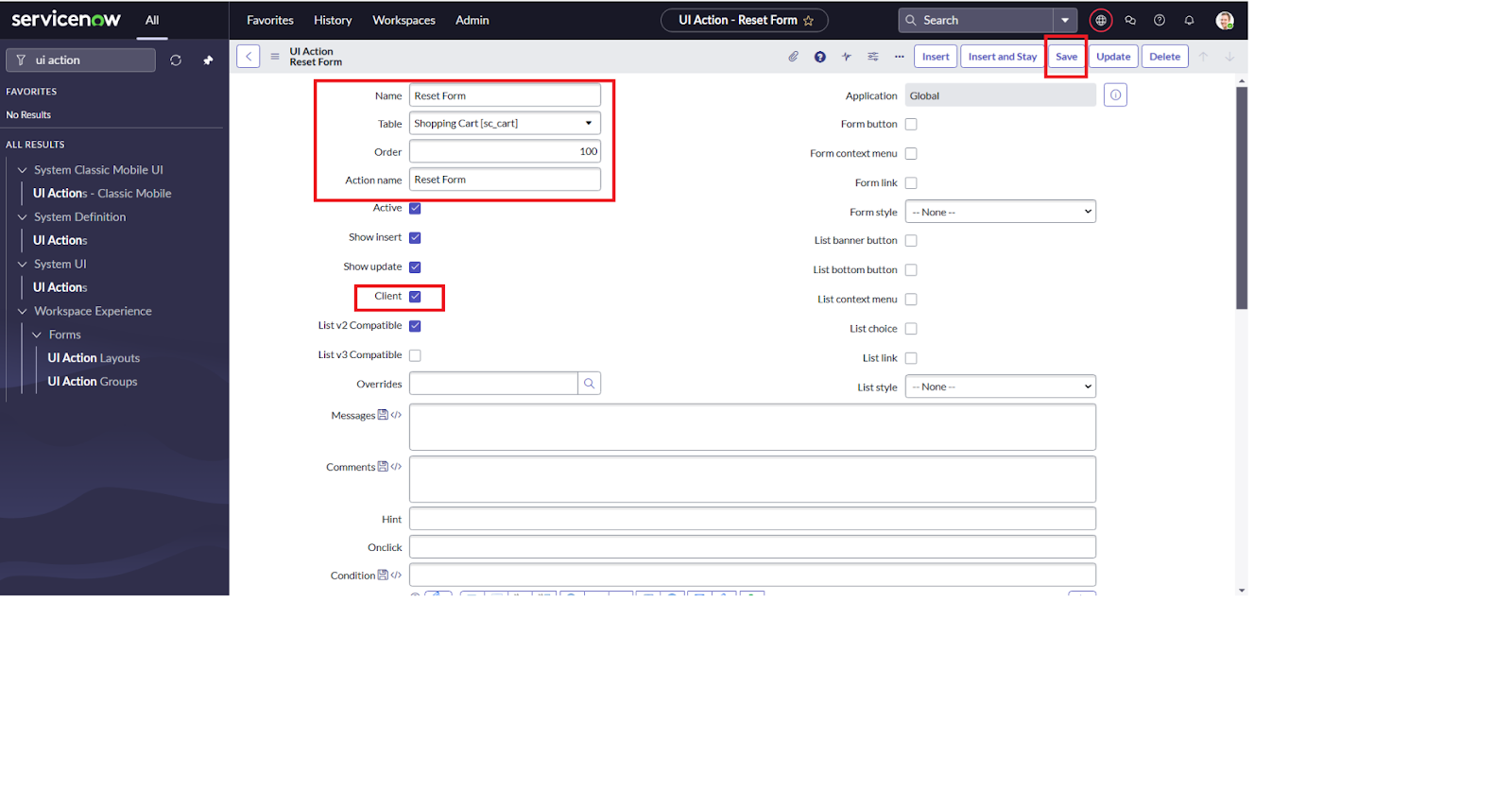


### UI Action

**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

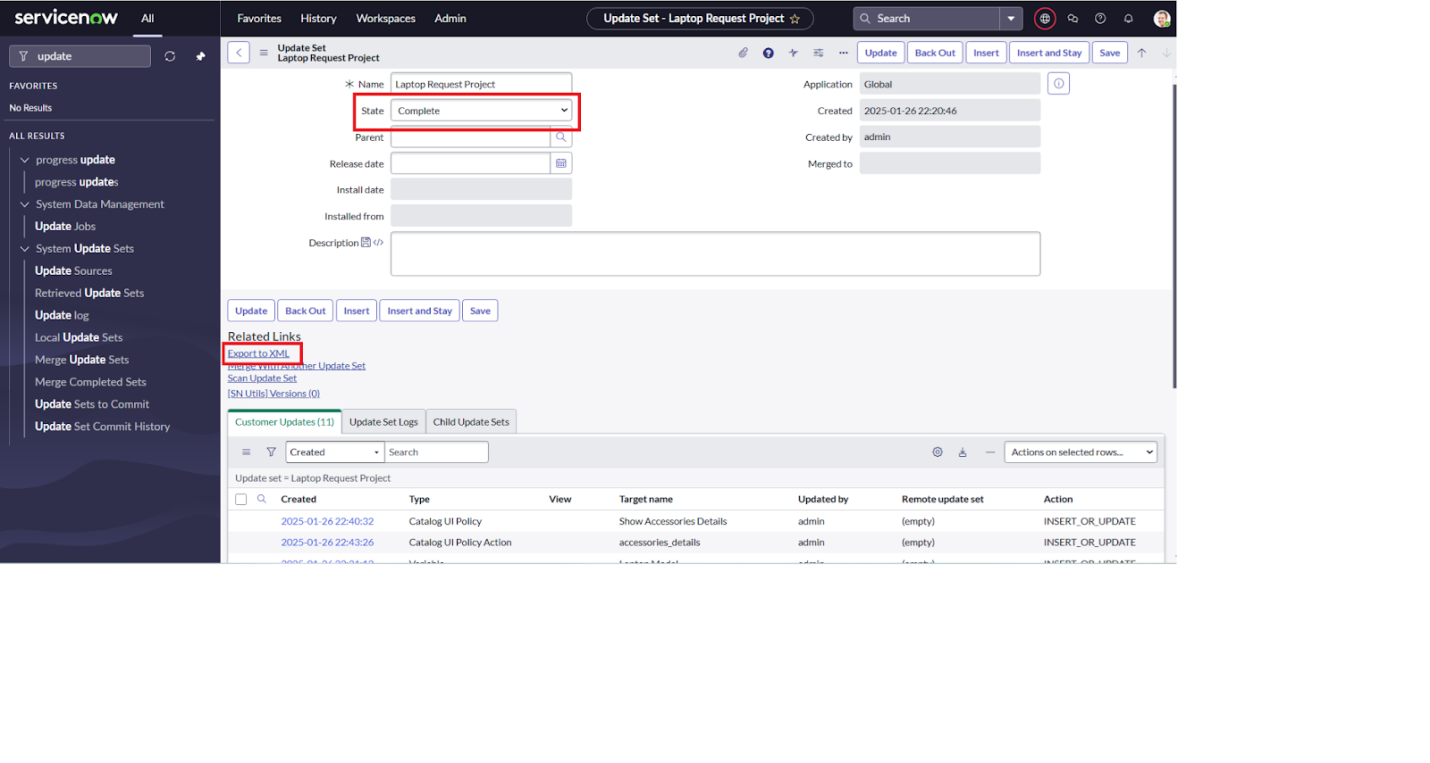
              Table: shopping cart(sc\_cart)  
              Order:100  
              Action name:  Reset form  
               Client : checked  
Script:  
    function resetForm() {  
    g\_form.clearForm(); // Clears all fields in the form  
    alert("The form has been reset.");  
}  
  
Click on save



### Export Update Set

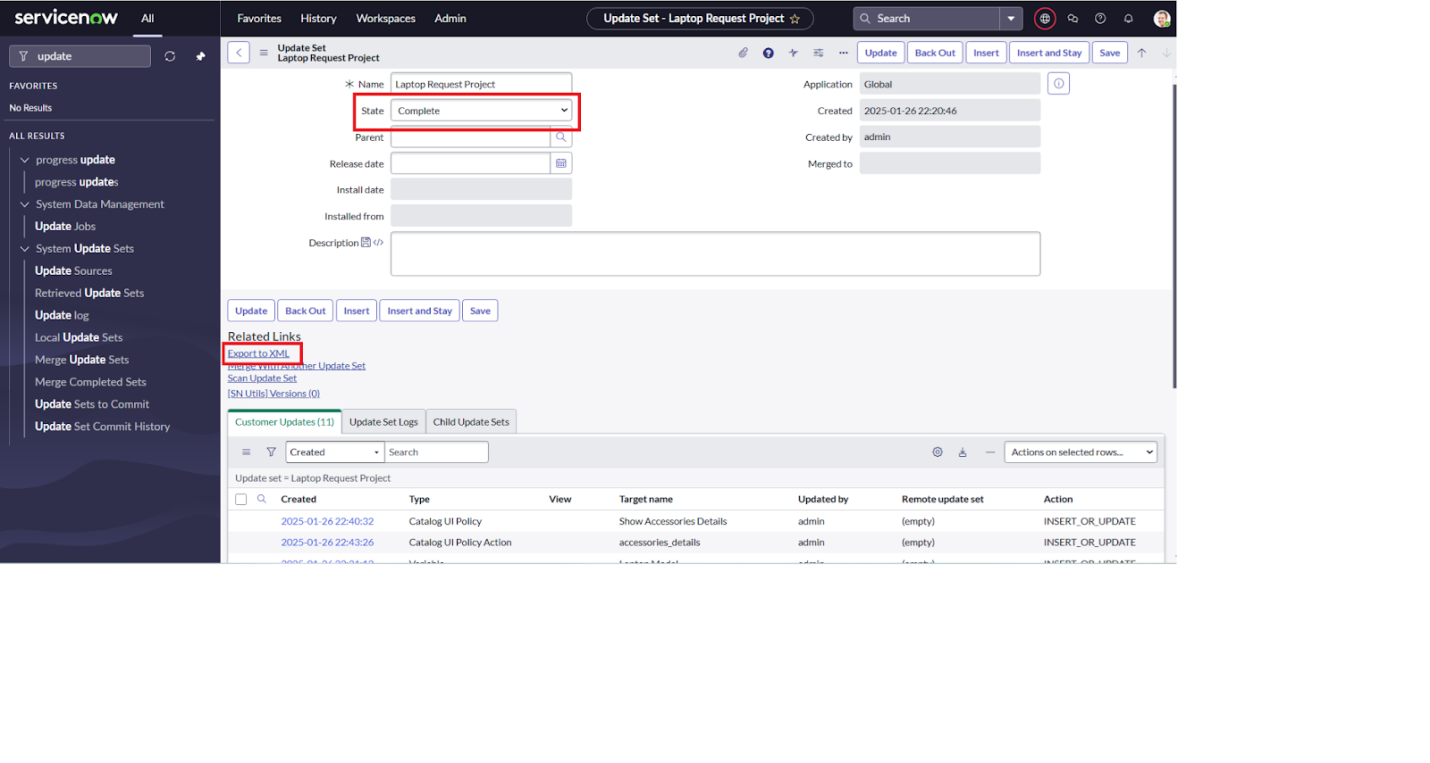
**Exporting Changes To Another Instances**

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



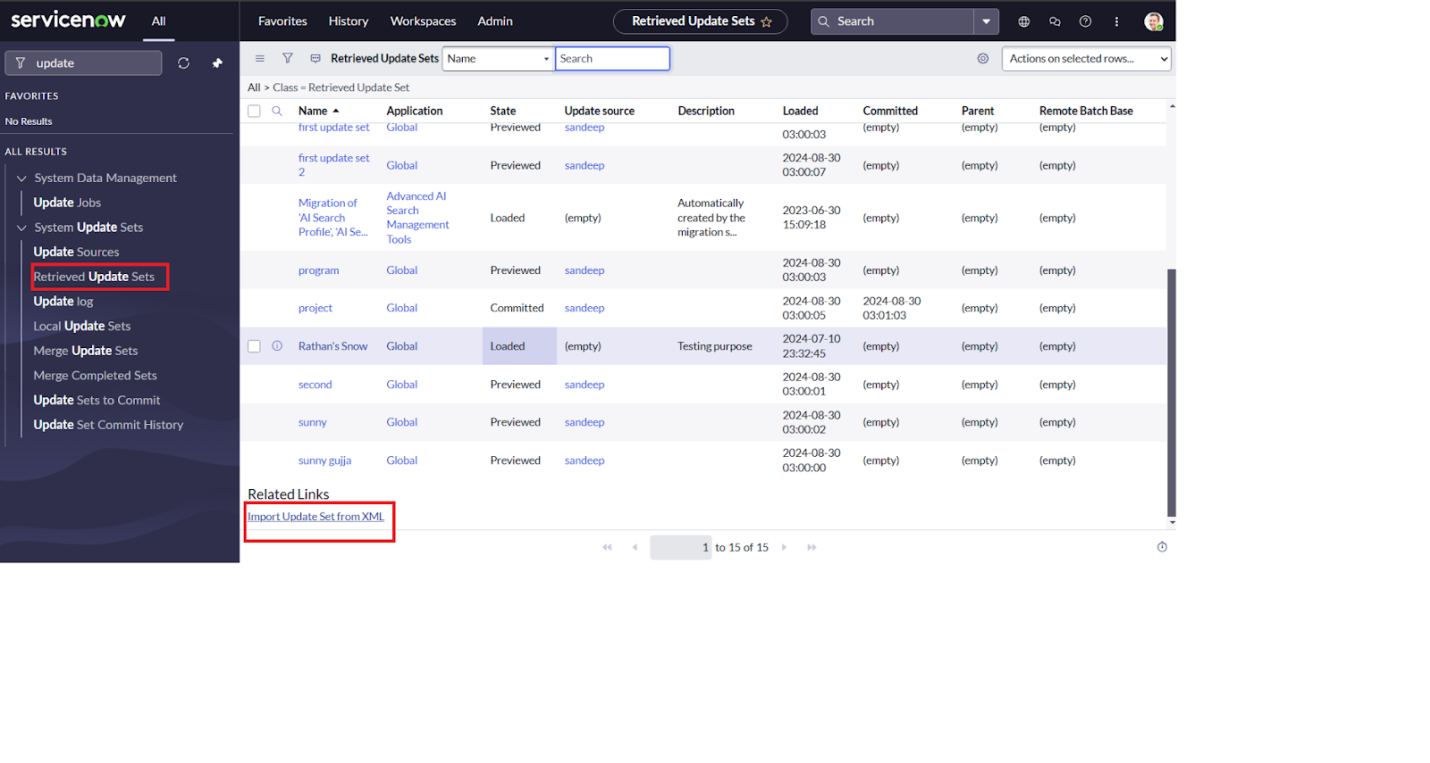
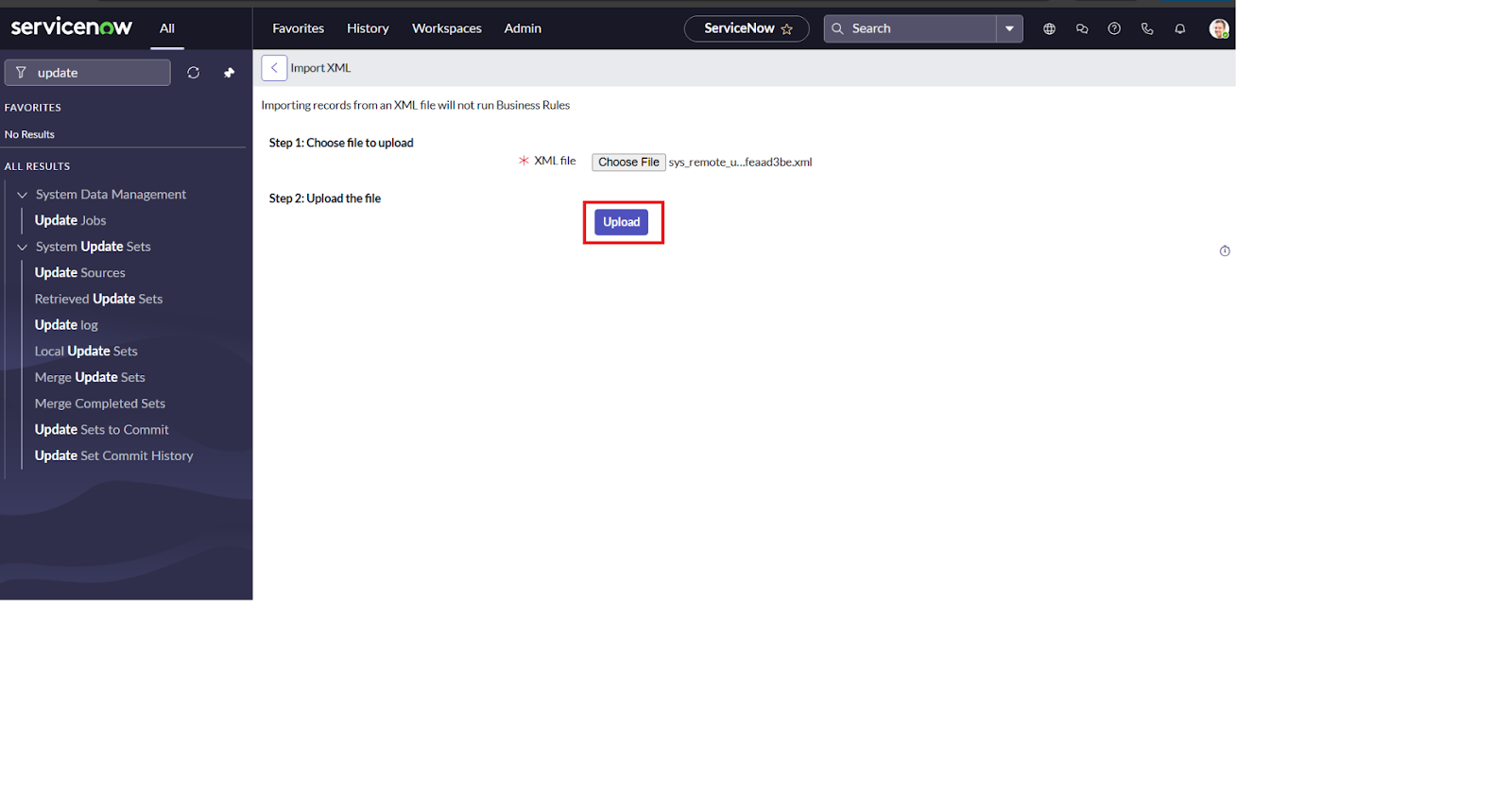
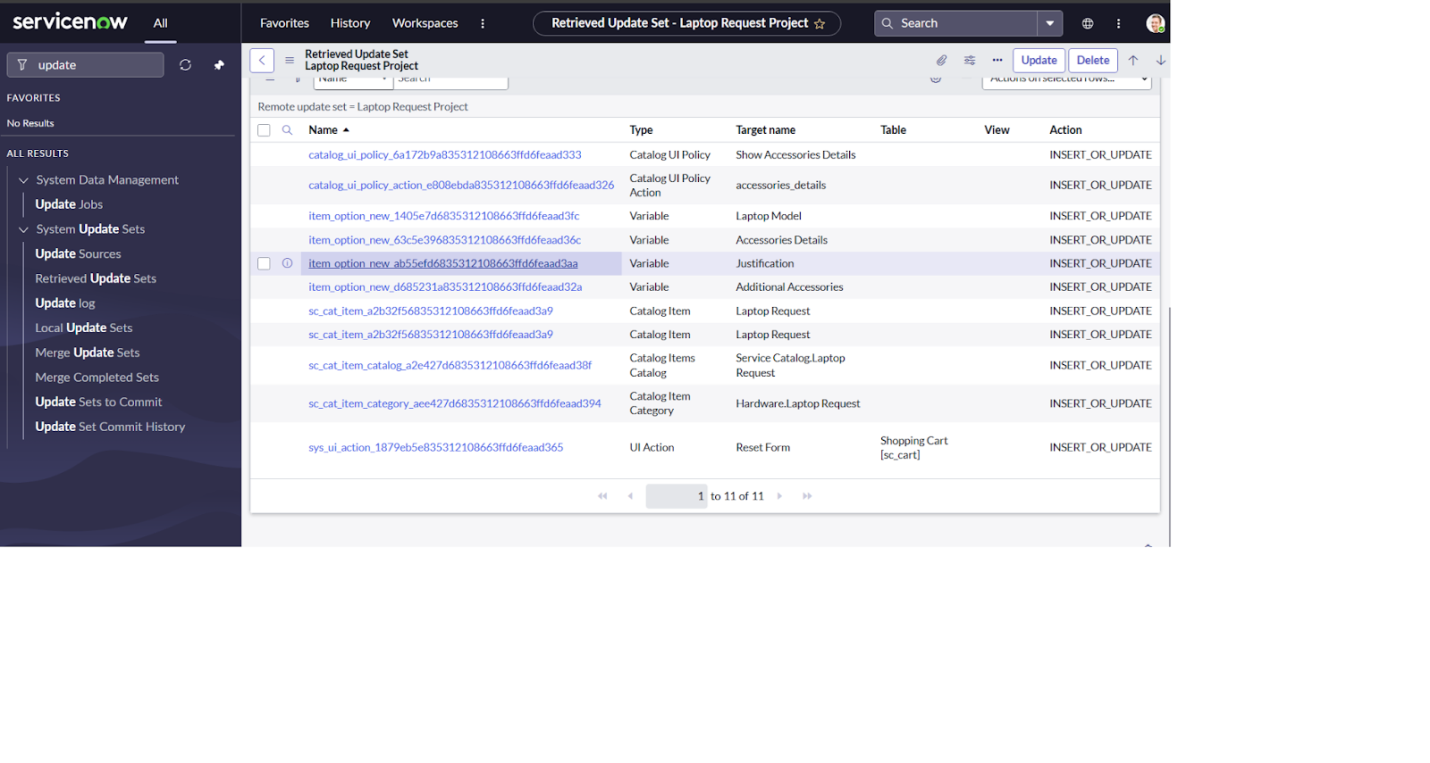
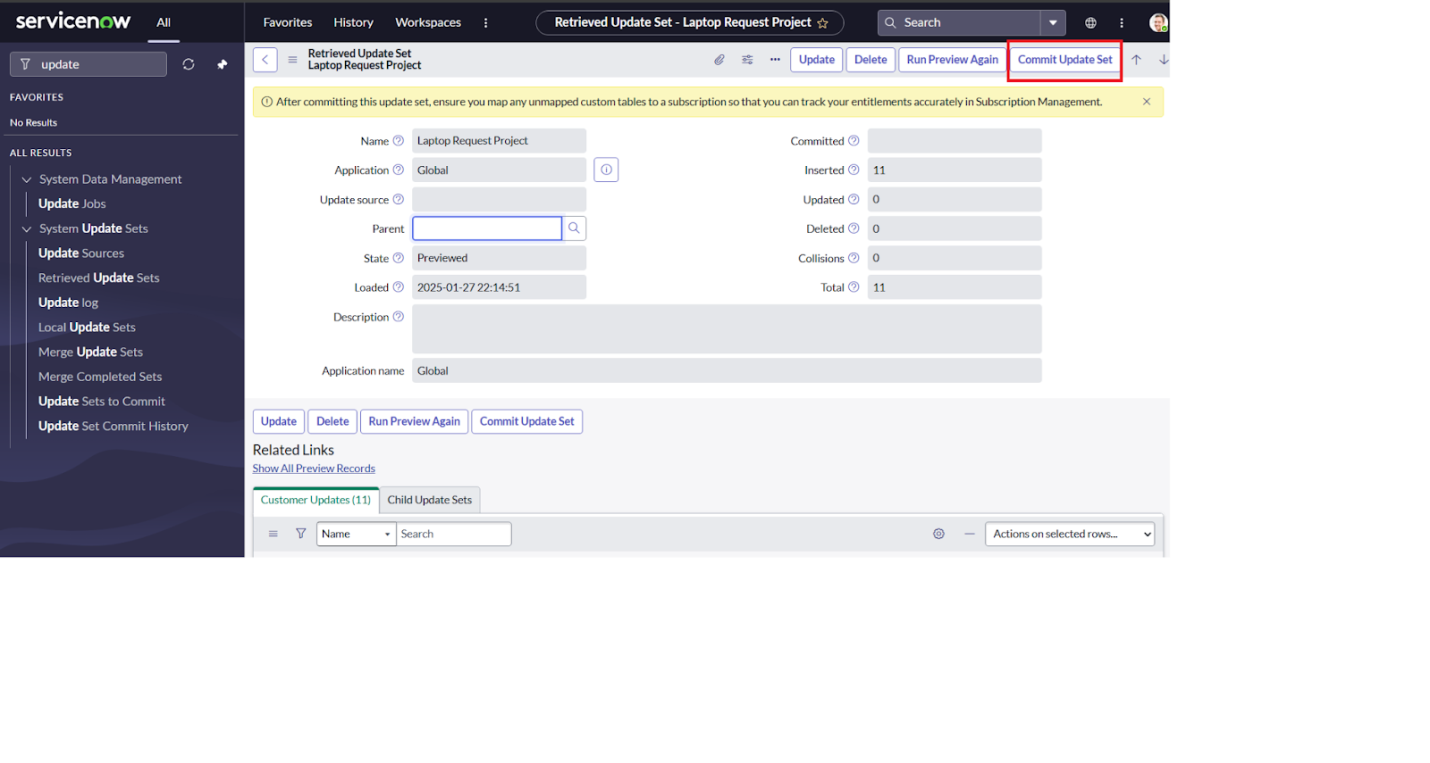
**Exporting Changes To Another Instances**

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



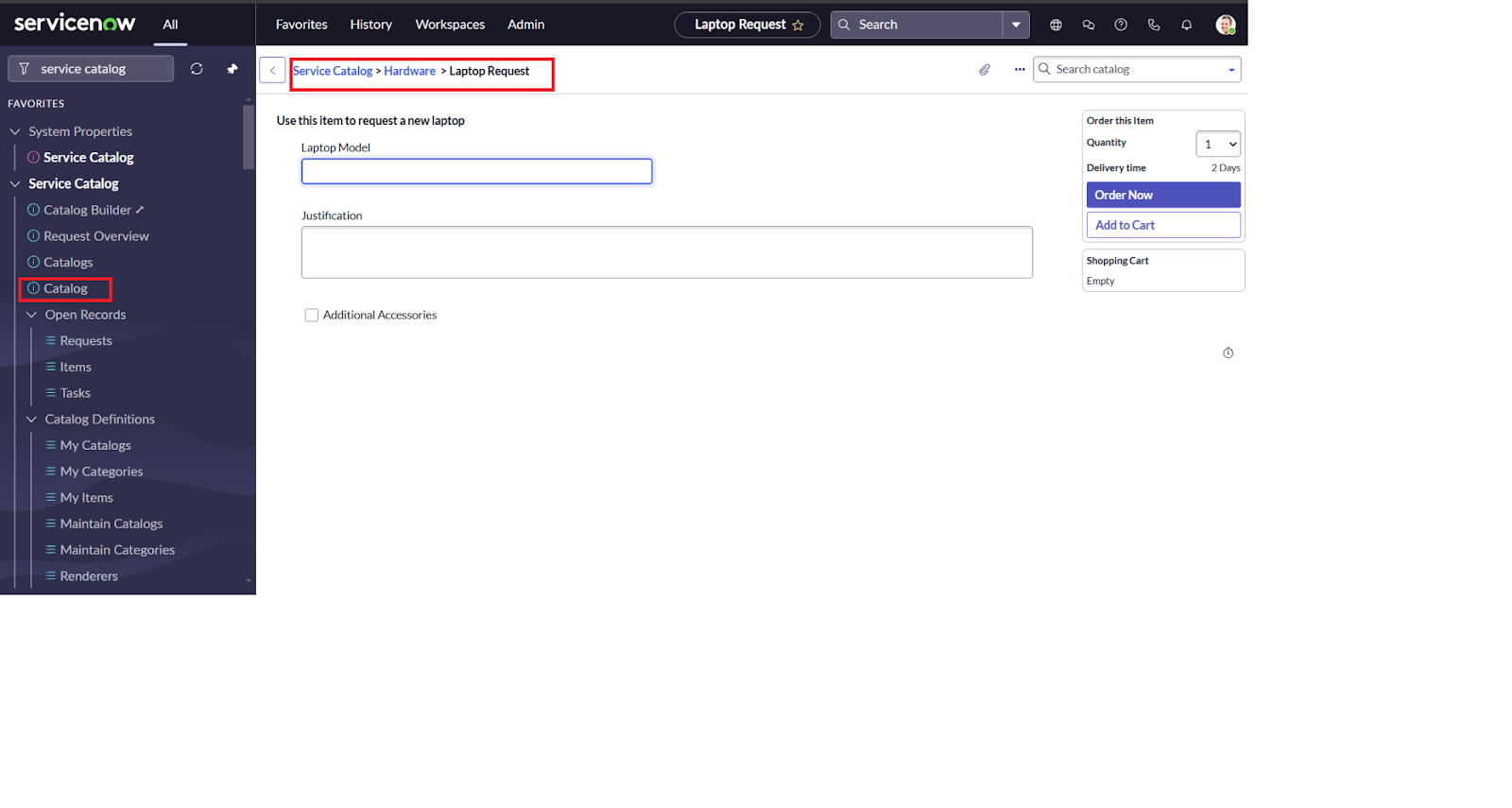
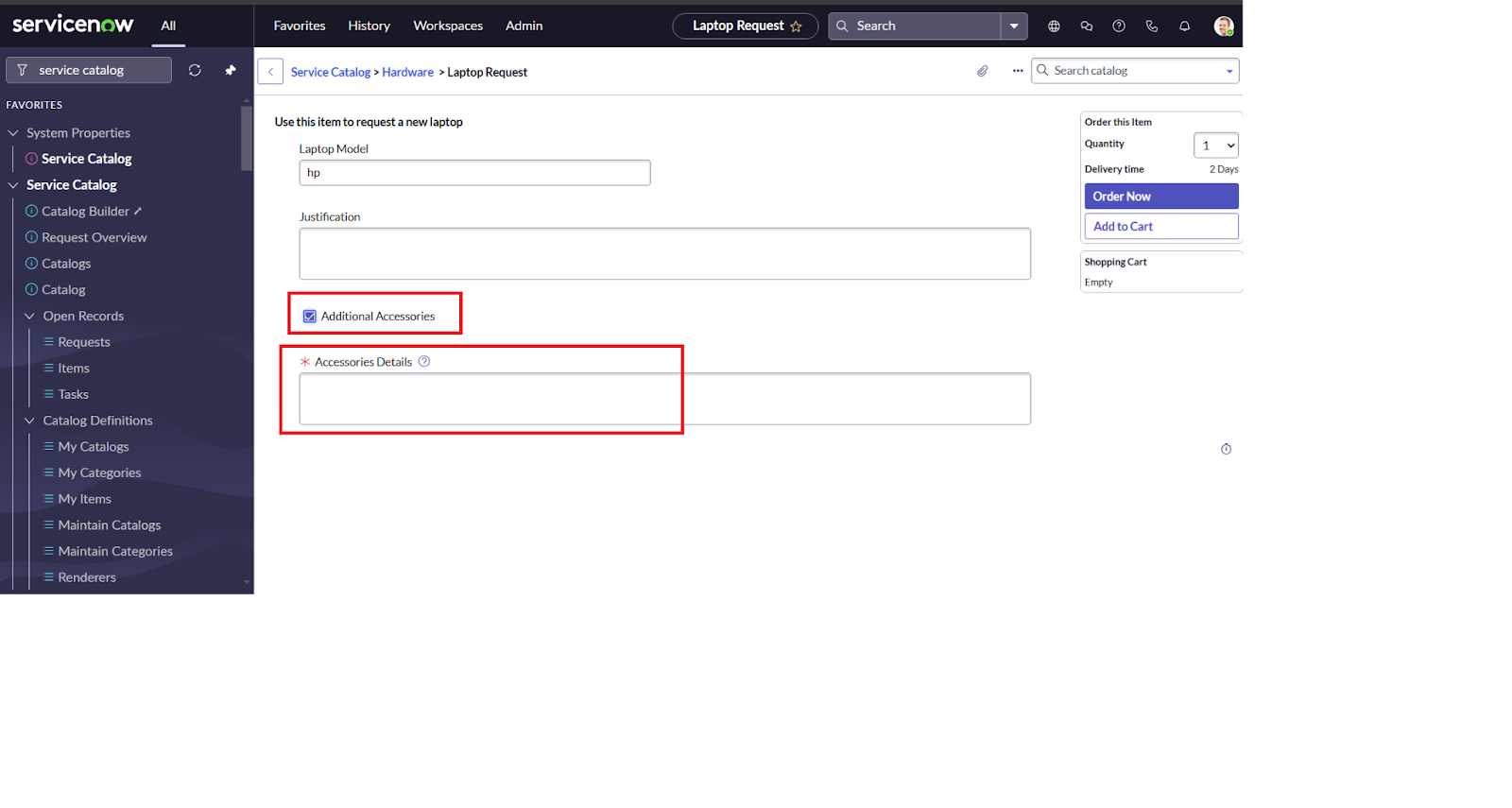
### Login To Another Instance

**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 commiting update set in this instance we get all updates which are done in the previous instance

### Testing

**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.**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.