

SERVICE NOW GLOBAL CERTIFICATION PROGRAM

SmartInternz

LAPTOP REQUEST CATALOG ITEM

NAME: SAI JOSHITHA PALAVALASA

E-MAIL: 322103382046@gvpce.ac.in

BRANCH: CSE-AIML(CSM)

**COLLEGE NAME: GAYATRI VIDYA PARISHAD COLLEGE OF ENGINEERING,
AUTONOMOUS**

PROJECT TITLE: LAPTOP REQUEST CATALOG ITEM

1. Project Overview

The **Laptop Request Catalog Item** project aims to simplify and automate the laptop request process within an organization using **ServiceNow**. Previously, the process of requesting laptops was **manual, time-consuming, and error-prone**, lacking real-time validation or automation features.

By creating a **Service Catalog Item** in ServiceNow, employees can easily submit laptop requests with guided form inputs, dynamic behavior (showing or hiding fields based on selections), and instant feedback.

The project also implements **UI Policies, UI Actions, and Local Update Sets** to ensure efficient management, reusability, and easy deployment across instances.

This solution provides:

- A **user-friendly form** to request laptops.
- **Dynamic visibility** of form fields (e.g., accessory details shown only if selected).
- A “**Reset Form**” button to clear inputs instantly.
- **Seamless migration** of configuration changes via Update Sets.

2. Objectives

Business Goals

The *Laptop Request Catalog Item* project was initiated to simplify and digitize the laptop request process within the organization, which was earlier managed manually and often led to delays, errors, and lack of visibility. The main business goal is to provide a **streamlined, automated, and user-friendly platform** that allows employees to request laptops effortlessly through the ServiceNow Service Catalog.

One of the key objectives is to **enhance operational efficiency** by minimizing manual intervention and ensuring that requests flow seamlessly from initiation to fulfillment. The project also aims to **ensure data accuracy** by utilizing dynamic form behaviors, where only relevant fields appear based on user inputs, reducing the chances of

incomplete or incorrect submissions.

Another important goal is to **improve the overall employee experience** by offering a clear and guided interface that helps users raise requests confidently. The system's ability to reset the form instantly through a dedicated UI Action adds further convenience for users.

From a governance standpoint, the project is designed to **enable traceability and standardization** by leveraging ServiceNow's **Local Update Sets**, ensuring that all configuration changes are properly tracked, versioned, and can be easily deployed to multiple environments. Overall, the project promotes a **self-service culture** within the organization, reducing the IT team's workload and encouraging automation-driven service delivery.

Specific Outcomes

As a result of this implementation, several significant outcomes were achieved. The laptop request process, which earlier depended on emails and manual approvals, is now completely **automated through ServiceNow's Service Catalog**. Employees can submit requests digitally, ensuring faster processing, accurate data capture, and improved tracking of each request.

The introduction of **dynamic form behavior**—such as displaying the “Accessories Details” field only when the “Additional Accessories” checkbox is selected—enhanced data precision and made the form more intelligent and user-responsive. The addition of the **“Reset Form”** button through a UI Action improved usability, allowing users to clear all inputs and start afresh without manually deleting data.

The use of **Catalog UI Policies** ensured that mandatory fields were enforced appropriately, maintaining the integrity of data collected through the catalog item. Moreover, the project leveraged **Local Update Sets** to manage and migrate configurations across ServiceNow instances. This provided a robust way to maintain **governance, version control, and audit readiness**, ensuring every change could be tracked and replicated safely in other environments.

The solution also led to a **reduction in IT workload** since employees could now handle routine laptop requests themselves through the catalog, allowing IT staff to focus on more critical issues. Additionally, the project significantly **enhanced employee satisfaction** by offering a faster, more transparent, and error-free experience when

requesting laptops.

In essence, the *Laptop Request Catalog Item* project not only modernized the process but also demonstrated how ServiceNow's capabilities—when implemented effectively—can create real business value by boosting productivity, transparency, and user satisfaction across the organization.

3. ServiceNow Key Features and Concepts Utilized

The *Laptop Request Catalog Item* project makes extensive use of several **key ServiceNow features and concepts** to deliver a complete and dynamic solution.

The foundation of the implementation lies in the **Service Catalog**, which serves as a centralized platform for employees to request IT and non-IT services. Within this catalog, a dedicated **Catalog Item** called “Laptop Request” was created under the *Hardware* category. This item acts as the main interface through which users can submit laptop requests, providing a structured and easy-to-use form that replaces traditional manual methods.

To collect user input, various **Variables** were added to the catalog item, including fields such as *Laptop Model*, *Justification*, and *Additional Accessories*. These variables are the building blocks of the form, capturing all necessary details from the requester. Additionally, a field named *Accessories Details* was configured to appear dynamically only when the user selects the “Additional Accessories” checkbox, thereby ensuring that only relevant fields are displayed. This dynamic control is achieved through the use of **Catalog UI Policies**.

Catalog UI Policies and their corresponding **UI Policy Actions** were implemented to determine when certain fields should be visible, mandatory, or read-only. For instance, when “Additional Accessories” is selected, the “Accessories Details” field becomes visible and mandatory, guiding the user to provide complete information. This approach ensures that the form remains simple, interactive, and free of unnecessary clutter.

Another key concept used in this project is the **UI Action** feature. A custom UI Action named “Reset Form” was created and linked to the *Shopping Cart (sc_cart)* table. This action allows users to instantly clear all inputs on the form by clicking a button, providing an intuitive and user-friendly way to start over without refreshing the entire

page. The functionality was implemented using a short client-side script, demonstrating how client scripts can be used effectively to enhance the end-user experience.

To ensure proper tracking, governance, and migration of all configurations made during the project, a **Local Update Set** was created. The update set acts as a container that records every change performed in ServiceNow, such as creating catalog items, variables, UI policies, and UI actions. Once all changes were finalized, the update set was marked as complete and exported as an XML file. This file could then be imported into another ServiceNow instance, ensuring consistency and traceability across environments.

Overall, this project utilized ServiceNow's **core configuration management principles**—including catalog design, dynamic scripting, client-side interactivity, and update set tracking—to create a cohesive, maintainable, and scalable solution. Each of these ServiceNow features contributed to building a well-structured, automated, and efficient workflow for handling laptop requests within the organization.

4. Detailed Steps to Solution Design

Step 1: Create Local Update Set

1. Navigate to: **All → System Update Sets → Local Update Sets**
2. Click **New** → Fill in details:
 - Name: **Laptop Request**
3. Click **Submit** and then **Make Current** to activate it.

The screenshot shows a ServiceNow web interface titled "Update Set - Create New Update Set". The URL in the address bar is https://dev199906.service-now.com/nav/ui/classic/params/target/sys_update_set.do%3Fsys_id%3D-1%26sys_is_list%3Dtrue%26sys_target%3Dsys_update_set%26sysparm_checked_items%. The page displays a form with the following fields:

- Name: Laptop Request
- State: In progress
- Parent: (dropdown menu)
- Release date: (date picker)
- Description: (text area)

At the bottom of the form are two buttons: "Submit" and "Submit and Make Current". The browser's header includes "servicenow" and "All Bookmarks". The taskbar at the bottom shows various application icons and the date/time: 04-11-2025, 12:33.

Step 2: Create Service Catalog Item

1. Go to: All → Service Catalog → Maintain Items
2. Click New and fill in the following:
 - **Name:** Laptop Request
 - **Catalog:** Service Catalog
 - **Category:** Hardware
 - **Short Description:** Use this item to request a new laptop
3. Click Save

Step 3: Add Variables

After saving the Catalog Item, scroll to the **Variables Related List**, and add the following:

Variable Name	Type	Name (system)	Order
Laptop Model	Single Line Text	laptop_model	100
Justification	Multi Line Text	justification	200
Additional Accessories	Checkbox	additional_accessories	300
Accessories Details	Multi Line Text	accessories_details	400

Click **Submit** after each variable.

Type	Question	Order
Single Line Text	Laptop Model	100
Multi Line Text	Justification	200
CheckBox	Additional Accessories	300
Multi Line Text	Accessories Details	400

Step 4: Create Catalog UI Policy

1. Open **Laptop Request** Catalog Item → Scroll to **Catalog UI Policies** → Click **New**
2. Fill:
 - **Short Description:** Show Accessories Details
 - **Condition:**
 - Field: additional_accessories
 - Operator: is
 - Value: true
3. Click **Save** (do not submit yet).
4. Scroll to **Catalog UI Policy Actions** → Click **New**
 - Variable Name: accessories_details
 - Mandatory: True
 - Visible: True
 - Order: 100
5. Click **Save**

This screenshot shows the 'Catalog UI Policy - show accessories details' page in ServiceNow. The top navigation bar includes 'All', 'Favorites', 'History', 'Workspaces', and a search bar. The main content area displays policy details:

- Applies to:** A Catalog Item
- Catalog item:** Laptop Request
- Application:** Global
- Active:** checked
- Short description:** show accessories details

The 'When to Apply' section contains a 'Script' tab and a 'Catalog Conditions' section with a dropdown menu showing 'additional_accessories' is true. It also includes checkboxes for 'Applies on a Catalog Item view' (checked), 'Applies on Catalog Tasks' (unchecked), 'Applies on Requested Items' (unchecked), and 'On load' (checked). There is also a 'Reverse if false' checkbox.

At the bottom are 'Update' and 'Delete' buttons, and a 'Related Links' section with 'Run Point Scan'.

This screenshot shows the 'Catalog UI Policy Action - accessories_details' page in ServiceNow. The top navigation bar includes 'All', 'Favorites', 'History', 'Workspaces', and a search bar. The main content area displays action details:

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

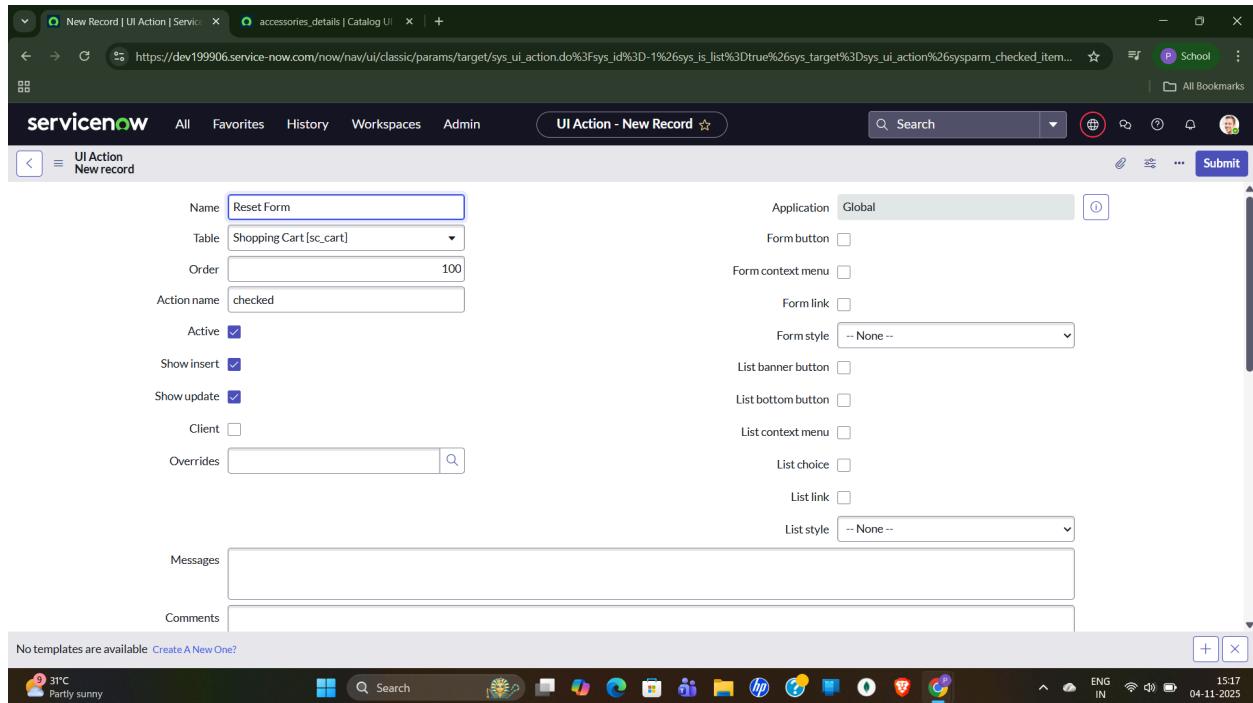
The 'When to Apply' section contains a 'Script' tab and a 'Catalog Conditions' section with a dropdown menu showing 'additional_accessories' is true. It also includes checkboxes for 'Applies on a Catalog Item view' (checked), 'Applies on Catalog Tasks' (unchecked), 'Applies on Requested Items' (unchecked), and 'On load' (checked). There is also a 'Reverse if false' checkbox.

At the bottom are 'Update' and 'Delete' buttons, and a 'Related Links' section with 'Run Point Scan'.

Step 5: Create UI Action

1. Go to: **All → System Definition → UI Actions**
2. Click **New** and fill in:
 - **Table:** Shopping Cart (sc_cart)
 - **Order:** 100
 - **Action Name:** Reset Form
 - **Client:** Checked
3. Add the following script:
4.

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



Step 6: Exporting Update Set

1. Navigate to: **All → System Update Sets → Local Update Sets**
2. Select the created update set (**Laptop Request Project**)
3. Change **State** to *Complete*

4. In the **Updates** related list, verify all configurations are captured
 5. Click **Export to XML** – it downloads the update set as an XML file

The screenshot shows the ServiceNow interface for updating a system request. The title bar reads "Update Set - Laptop Request". The main form contains fields for Name (Laptop Request), State (Complete), Parent (empty), Release date (empty), Install date (empty), Installed from (empty), and Description (empty). To the right, application details are shown: Application (Global), Created (2025-11-03 23:03:10), Created by (admin), and Merged to (empty). At the bottom left are "Update" and "Back Out" buttons. Below the form, a section titled "Create a Retrieved Update Set for exporting" offers options like "Export to XML", "Merge With Another Update Set", and "Scan Update Set". A navigation bar at the bottom includes "Customer Updates (15)", "Update Set Logs", and "Child Update Sets". The status bar at the bottom right shows the date as 04-11-2020.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

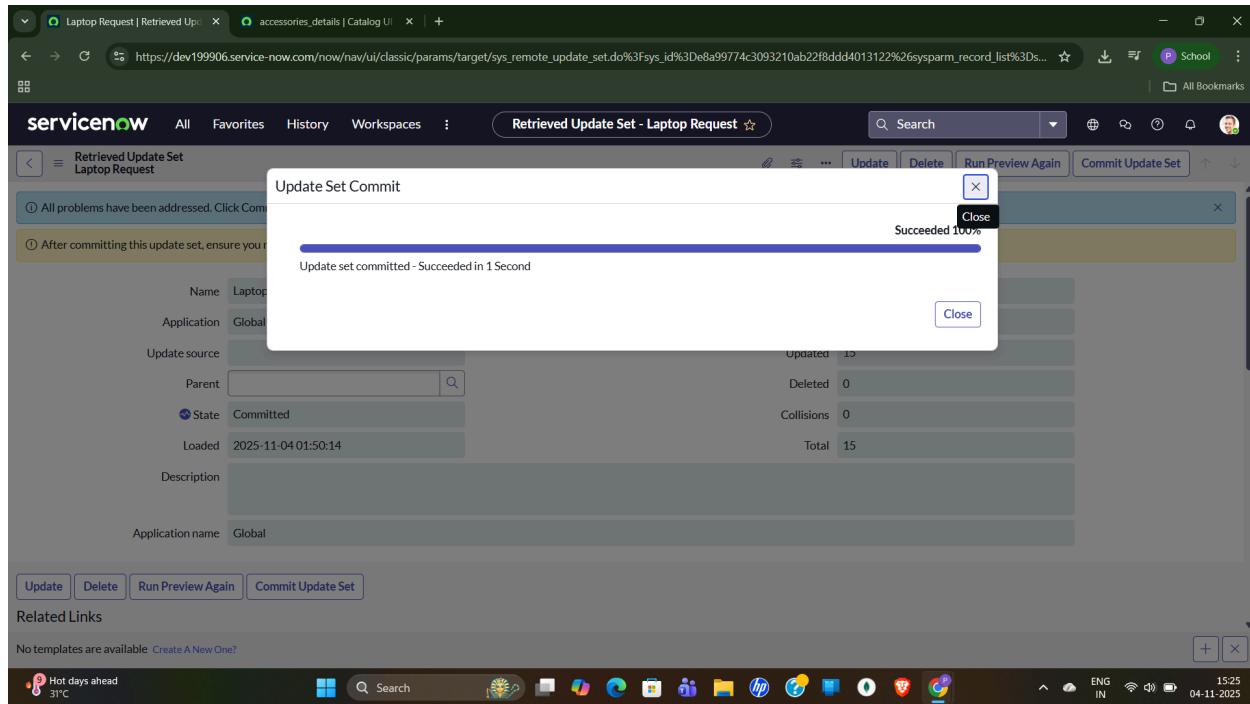
```
<?xml version="1.0" encoding="UTF-8"?><record_update table="item_option_new"><item_option_new action="INSERT_OR_UPDATE"><application display_value="Global"/><application_name>Global</application_name><application_scope>global</application_scope><application_version></application_version><collisions/><commit_date/><deleted/><description/><inserted/><name>Laptop Request</name><parent_id/><parent display_value="" /><release_date/><remote_base_update_set display_value="" /><remote_parent_id/><remote_sys_id>b9333238c3453210ab22f8ddd40131db</remote_sys_id><state>loaded</state><updated/><sys_class_name>sys_remote_update_set</sys_class_name><sys_created_by>admin</sys_created_by><sys_created_on>2025-11-04 09:58:14</sys_created_on><sys_id>e8a99774c3093210ab22f8ddd4013122</sys_id><sys_mod_count>0</sys_mod_count><sys_updated_by>admin</sys_updated_by><sys_updated_on>2025-11-04 09:58:14</sys_updated_on><update_set display_value="" /><update_set display_value="" /><updated/></sys_remote_update_set><?sys_update_xml action="INSERT_OR_UPDATE"?><action>INSERT_OR_UPDATE</action><application display_value="Global">global</application><category>customer</category><comments/><name>item_option_new_93165f70c3093210ab22f8ddd401310a</name><payloads><?xml version="1.0" encoding="UTF-8"?><record_update table="item_option_new"><item_option_new action="INSERT_OR_UPDATE"><active>true</active><attributes><cat_item display_value="Laptop Request">b0e37678c3453210ab22f8ddd40131dc</cat_item><choice_direction>down</choice_direction><choice_field><choice_table><conversational_label><create_roles><default_html_value><default_val_value><delete_roles><delivery_plan><description><disallow_title>false</disallow_title><do not select first>false</do not select first>
```

Step 7: Retrieve Update Set in Another Instance

1. Open another instance in **Incognito Mode**
2. Go to: **All → System Update Sets → Retrieved Update Sets**
3. Click **Import Update Set from XML**
4. Upload the previously downloaded XML file
5. Once imported, preview and commit the update set

The screenshot shows a ServiceNow web interface with the following details:

- URL:** https://dev199906.service-now.com/nav/uiclassic/params/target/sys_remote_update_set_list.do?sysparm_fixed_query%3Dsys_class_name%3Dsys_remote_update_set
- Page Title:** Retrieved Update Sets
- Search Bar:** Name (dropdown), Search
- Table Headers:** Name, Application, State, Update source, Description, Loaded, Committed, Parent, Remote Batch Base
- Table Data:** One row with the following values:
 - Name: Laptop Request
 - Application: Global
 - State: Loaded
 - Update source: (empty)
 - Description: Import update set from an XML file exported from another instance
 - Loaded: 2025-11-04 01:50:14
 - Committed: (empty)
 - Parent: (empty)
 - Remote Batch Base: (empty)
- Tooltip:** Import update set from an XML file exported from another instance
- Bottom Status Bar:** Shows weather (31°C, Partly sunny), search bar, system icons, and system status (ENG IN, 15:22, 04-11-2025).



5. Testing and Validation

After deployment to the target instance:

1. Navigate to **Service Catalog → Hardware Category**
2. Search for **Laptop Request**
3. Open the item → verify the following:
 - Fields *Laptop Model*, *Justification*, and *Additional Accessories* appear
 - *Accessories Details* appears only when the checkbox is selected
 - “Reset Form” button clears all inputs
 - The request form submits successfully

The screenshot shows a ServiceNow Service Catalog page titled "Laptop Request". The main content area contains fields for "Laptop Model" and "Justification", both represented by text input boxes. Below these is a checked checkbox for "Additional Accessories". A required field "Accessories Details" is indicated by a red asterisk. To the right of the form is a sidebar with an "Order Now" section containing "Quantity" (set to 1) and "Delivery time" (set to 2 Days). A blue "Order Now" button is present. Below this is a "Shopping Cart" section showing "Empty". The browser address bar shows the URL: https://dev199906.service-now.com/nav/uiclassic/params/target/com.glideapp.servicatalog_cat_item_view.do?3Fv%3D1%26sysparm_id%3D0be37678c3453210ab22f8ddd4... .

Result: The catalog item works as expected, with all dynamic behaviors and UI actions functioning correctly.

6. Key Scenarios Addressed by the ServiceNow Implementation

The **Laptop Request Catalog Item** project effectively addresses several real-world challenges that existed in the traditional laptop request process.

Firstly, the manual and delayed nature of requests has been resolved through the **automation of the entire workflow** using the ServiceNow Service Catalog. Employees can now raise requests quickly through a structured and guided form, eliminating unnecessary paperwork and follow-ups.

Secondly, the issue of **inaccurate or incomplete data entry** has been handled by incorporating **dynamic UI Policies** that make fields mandatory or visible only when needed. This ensures that users provide all required information without confusion.

Thirdly, the repetitive effort involved in clearing or re-entering data is minimized through the implementation of a "**Reset Form**" **UI Action**, allowing users to instantly clear all

input fields and start over if needed.

Next, to address configuration migration and governance, the use of **Local Update Sets** provides a controlled mechanism for capturing and exporting all changes, ensuring smooth transfer of configurations between different ServiceNow instances.

Finally, the problem of **lack of user guidance** has been solved by designing an **intuitive and interactive form interface**, where conditional visibility and descriptive text guide the user step by step.

Through these improvements, the project not only enhances the efficiency of laptop requests but also demonstrates how ServiceNow can be leveraged to build dynamic, user-friendly, and enterprise-ready solutions.

7. Summary / Conclusion

The **Laptop Request Catalog Item Project** demonstrates how **ServiceNow** can replace traditional manual processes with automated, dynamic, and efficient workflows.

Through the use of **Service Catalog, UI Policies, UI Actions, and Update Sets**, the project successfully:

- Streamlined the laptop request process
- Improved data accuracy and user experience
- Enabled portability of configurations between instances
- Showcased ServiceNow's ability to enhance organizational service delivery

Overall, this implementation promotes **automation, efficiency, and governance** in IT service management while providing a modern, interactive interface for employees.