Laptop Request Catalog Item

INTRODUCTION :

In modern organizations, providing employees with the right tools and equipment is essential for maintaining productivity and efficiency. However, manual processes for requesting resources—such as laptops—often lead to delays, incomplete information, and communication gaps between employees and IT support teams. To overcome these challenges, a ***Service Catalog item*** is proposed in ***ServiceNow*** that will allow employees to easily and accurately request laptops through an automated, user-friendly interface.

This solution will feature ***dynamic form behavior***, ensuring that only relevant fields are displayed based on user selections, thereby improving data accuracy and user experience. Additionally, functionalities such as ***form reset***, ***clear instructions***, and ***change tracking*** will be implemented to enhance usability, maintain governance, and support deployment control. By automating and streamlining the laptop request process, this ServiceNow catalog item will reduce manual effort, minimize errors, and ensure faster fulfillment of requests, contributing to a more efficient and transparent IT service management process.

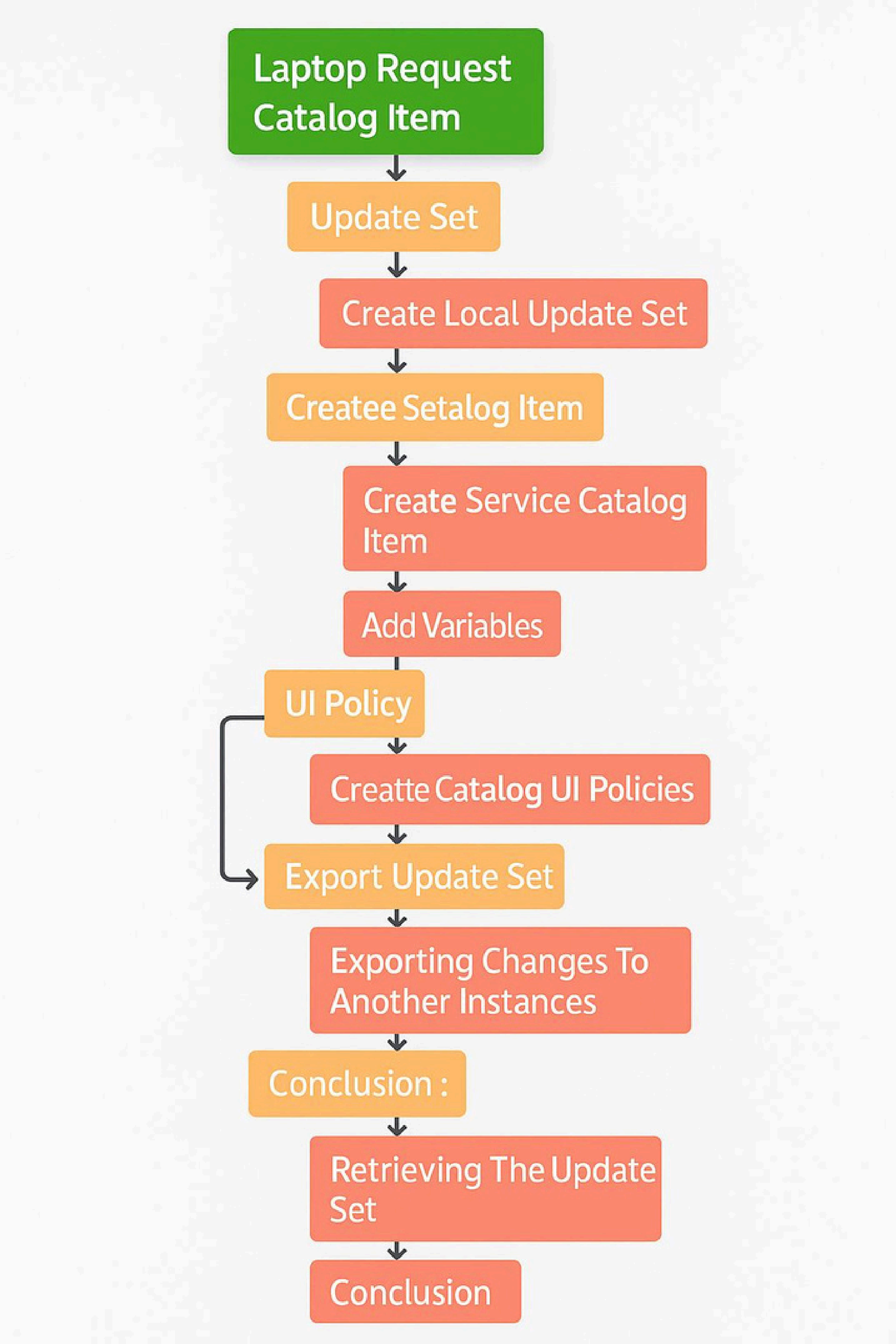
PROBLEM STATEMENT :

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.

ABSTRACT:

This project focuses on developing a ***ServiceNow Catalog Item*** that streamlines the laptop request process within an organization. The existing manual system causes delays and inaccuracies due to the lack of automation and dynamic guidance. The proposed solution introduces a user-friendly, interactive Service Catalog form that allows employees to quickly and accurately request laptops. Key features include ***dynamic field behavior***, ***form reset functionality***, and ***clear instructions*** to enhance user experience. Additionally, the solution incorporates ***change tracking and governance mechanisms*** to ensure transparency, compliance, and ease of deployment across environments. This implementation significantly improves operational efficiency and data accuracy in IT asset requests.

FLOW CHART:


Laptop Request Catalog Item
Update Set
Create Local Update Set
Createe Setalog Item
Create Service Catalog Item
Add Variables
UI Policy
Creatte Catalog UI Policies
Export Update Set
Exporting Changes To Another Instances
Conclusion :
Retrieving The Update Set
Conclusion

Tools and Platform :

- ***Platform:*** ServiceNow
- ***Modules Used:***
    - o Service Catalog
    - o Update Sets
    - o UI Policy
    - o UI Action
- ***Scripting Language:*** JavaScript (for client scripts)
- ***TestingInstance:***ServiceNowPersonal Developer Instance

EXPLAINATION:

Step 1: Create Update Set

In ServiceNow, an ***Update Set*** is used to capture and track configuration changes made during development. A ***Local Update Set*** was created to record all the configurations related to the ***Laptop Request Catalog Item***.

This ensures that all items such as variables, UI policies, and scripts are included and can later be moved or deployed to another ServiceNow instance.

By using update sets, administrators can maintain version control and governanceover all changes.

Step 2: Create Service Catalog Item

A new ***Service Catalog Item*** named ***"Laptop Request"*** was created under the ***Service Catalog*** module.This catalog item acts as a self-service form for employees to request laptops.The form includes fields like ***Short Description***, ***Category***, ***Request Item Name***, and ***Description***, giving users clear guidance when submitting their requests.

The catalog item is designed with a user-friendly interface that ensures consistent and accurate data collection.

Step 3: Add Variables

To make the catalog item interactive and data-driven, several ***variables*** were added.

These variables collect essential user input and customize the form based on employee needs.

Added variables:

- ***Employee Name*** – to identify the requester.
- ***Department*** – to categorize requests by team or division.
- ***Laptop Type*** – allows selection from available laptop models.
- ***RAM and Storage Options*** – lets users choose performance specifications.
- ***Purpose of Request*** – captures business justification.
- ***Required Date*** – specifies when the laptop is needed.

These fields make the form dynamic and help in automating downstream approvals and workflows.

Step 4: Apply UI Policies

Catalog UI Policies were created to control the visibility and requirement state of form fields dynamically.

This improves user experience and ensures accurate data entry.

Example:

If the ***Laptop Type*** selected is ***High-End***, then the ***Graphics Card*** field becomes visible and mandatory.

This dynamic behavior ensures that users only see fields relevant to their selection, minimizing confusion and data errors.

Step 5: Add UI Action (Reset Form Button)

To enhance usability, a ***UI Action*** button was added to reset the form fields.

This button triggers a ***client script*** that clears all entered values, allowing users to start over if needed.

Example Client Script:

```
    function resetForm() {
      var vars = ['employee_name', 'department', 'laptop_type', 'ram',
    'storage', 'justification'];
      for (var i = 0; i < vars.length; i++) {
        g_form.setValue(vars[i], '');
      }
      g_form.showFieldMsg('employee_name', 'Form has been reset
    successfully.', 'info');
}
```

This simple script loops through variable names and resets them, displaying a confirmation message afterward.

Step 7: Import and Retrieve Update Set in Target Instance

In the target instance, the exported XML file was ***imported*** using the ***Retrieved Update Sets*** module.

The imported update set was then ***previewed*** to identify conflicts or missing dependencies before being ***committed***.

This step ensures smooth migration and avoids overwriting any existing configurations.

Step 8: Testing

After deployment, ***testing*** was carried out to ensure everything functioned correctly.

Both ***functional testing*** and ***user acceptance testing (UAT)*** were performed to verify:

- Dynamic fields respond correctly based on user selection.
- The ***Reset Form*** button works properly.
- Data is stored correctly and visible in the ***Requested Items*** list.
- No errors occur during submission.

Results and Discussion

The implemented ***Laptop Request Catalog Item*** successfully automated the laptop request process in ServiceNow.
Users now experience a guided, error-free, and interactive form submission process.

Benefits observed:

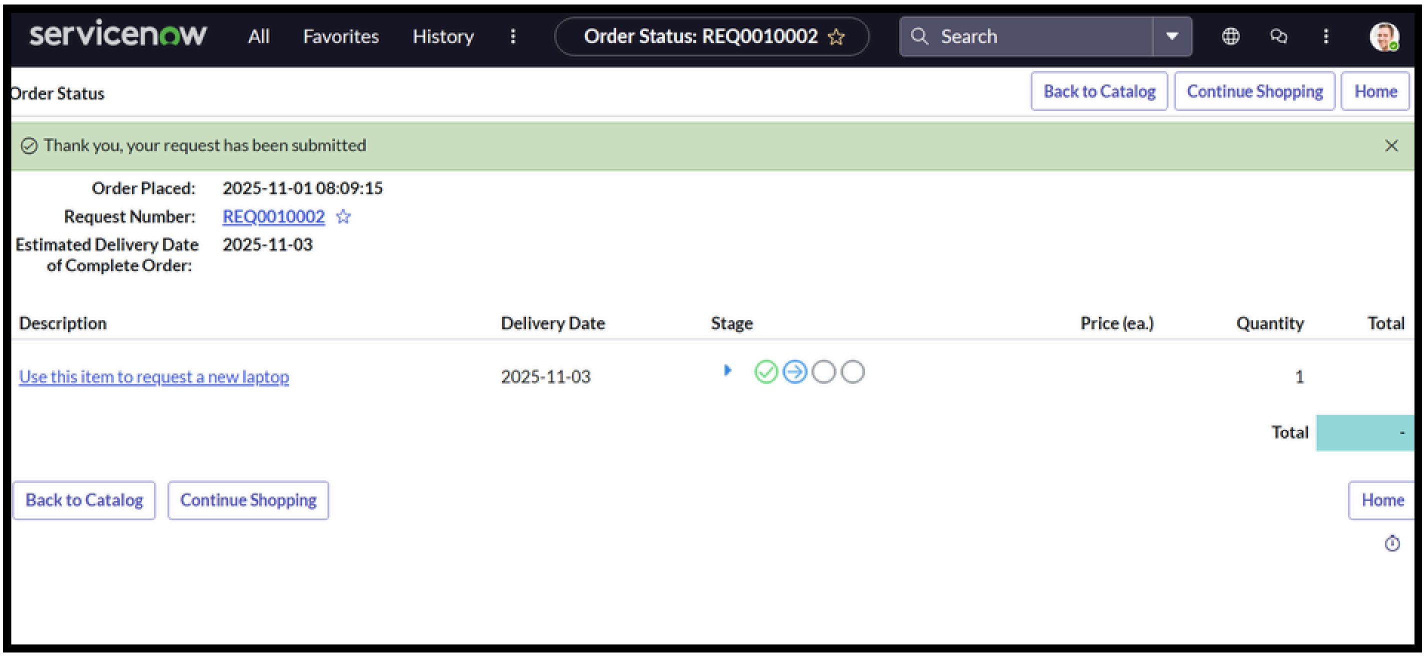
- Faster and more accurate request submissions.
- Reduced manual effort by IT and HR departments.
- Dynamic visibility of form fields enhances user experience.
- Easy reset functionality helps users correct mistakes quickly.
- All configurations are trackable through update sets for audit and deployment control.

Overall, this project improved efficiency, governance, and the overall user experience in the organization’s IT service request system

XML FILE CODE :

"C:\Users\Arjun\Downloads\sys_remote_update_set_e3355e67937c32901437f7718bba10ee.xml"

OUTPUT:



Order Status

Thank you, your request has been submitted

Order Placed:	2025-11-01 08:09:15
Request Number:	REQ0010002 ☆
Estimated Delivery Date of Complete Order:	2025-11-03

Description	Delivery Date	Stage	Price (ea.)	Quantity	Total
Use this item to request a new laptop	2025-11-03	▸ ✓ → ○ ○		1	
				Total	-

Demo video link:

https://drive.google.com/file/d/16DtiT50C-msj4VGsBRl_PC0kutw-Gkqt/view?usp=drive_link

CONCLUSION:

The ***Laptop Request Catalog Item*** project in ServiceNow successfully automated the process of requesting laptops within the organization. By using update sets, catalog items, variables, UI policies, and UI actions, the system ensures efficient and accurate data collection while enhancing the user experience.

The implementation of dynamic form behaviors, such as field visibility and reset functionality, made the form more interactive and user-friendly. Administrators can easily track all configuration changes through update sets, ensuring controlled and error-free deployment across different instances.

Overall, the project achieved its main objectives of improving request accuracy, reducing manual workload, and accelerating approval processes. It demonstrates the effectiveness of ServiceNow's Service Catalog features in streamlining IT service management and improving organizational productivity.