ANJALAI AMMAL MAHALINGAM ENGINEERING COLLEGE

DEPARTMENT OF INFORMATION TECHNOLOGY

NM-SERVICE NOW ADMINISTRATOR

LAPTOP REQUEST CATALOG ITEMS

Team Members:

NAME	REGISTER NUMBER
Varshini.A	820422205089
Pradheeba.G	820422205052
Suba Aishwarya.R	820422205080
Suweatha.E	802422205084

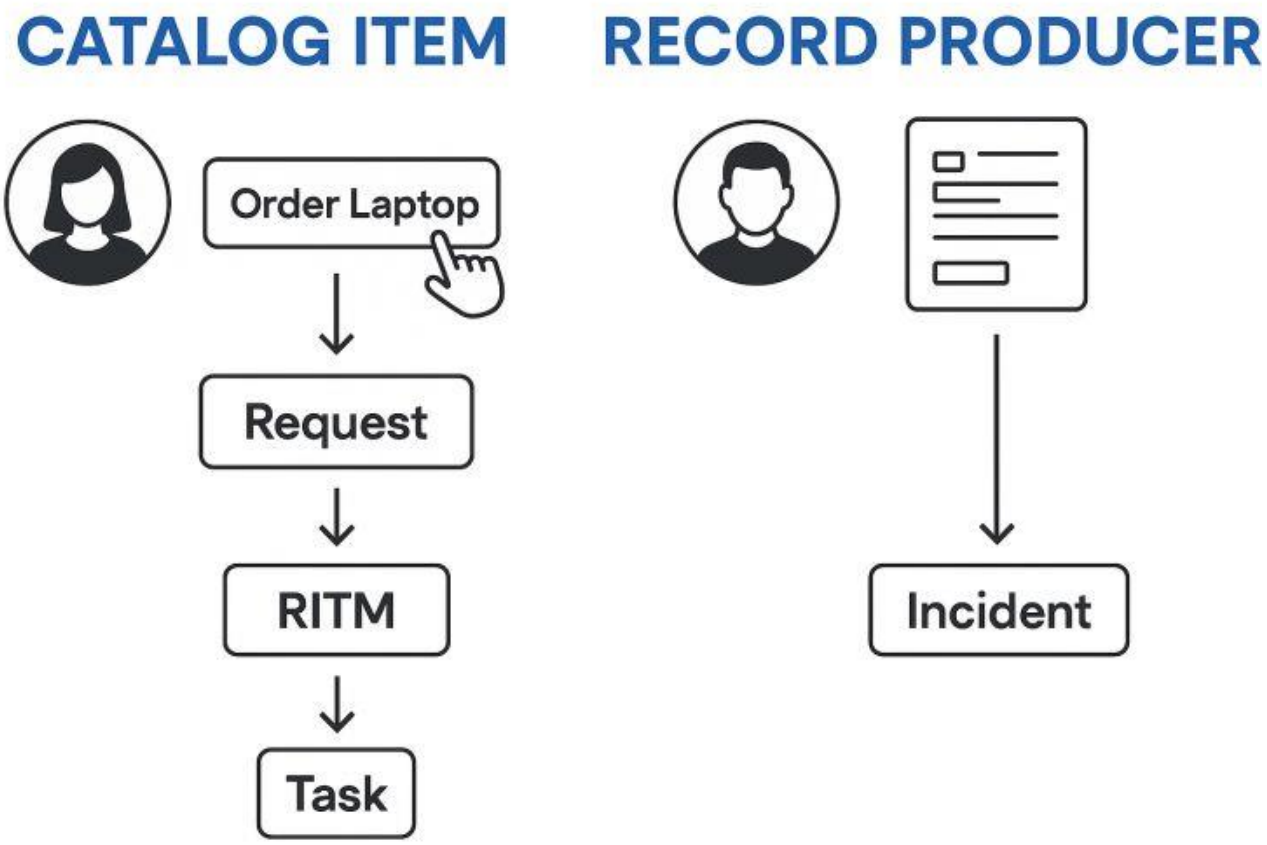
LAPTOP REQUEST CATALOG ITEMS

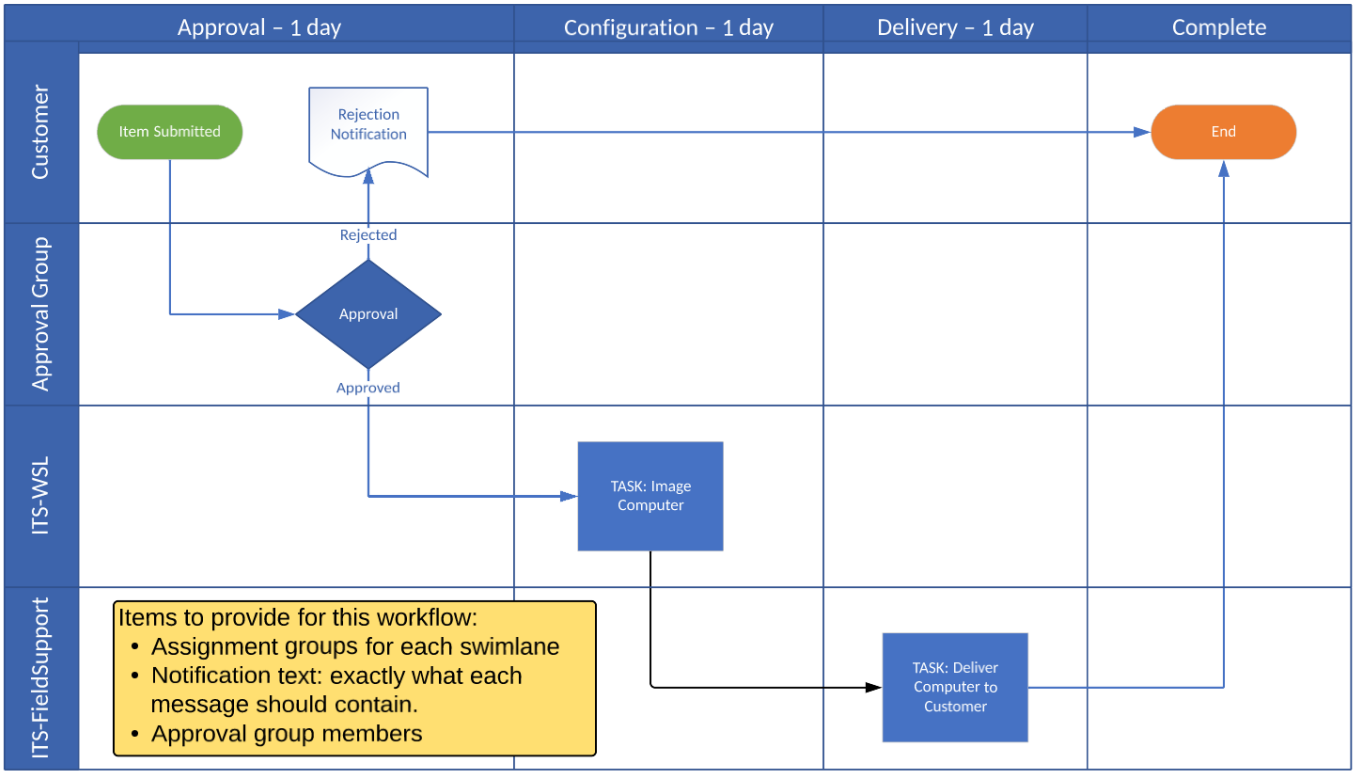
Abstract

In modern organizations, IT assets such as laptops are essential for employees to perform their daily work. However, many companies still depend on manual, paper-based, or email-driven request systems to provide laptops to employees. These manual processes are inefficient, prone to human error, and result in delays, data inaccuracies, and a lack of visibility in tracking requests.

This project aims to resolve these issues by developing a **Service Catalog item in ServiceNow** that enables employees to request laptops easily and efficiently. The system introduces **dynamic form behavior** to improve user guidance and ensure correct data input, along with features such as a **form reset option** and automated tracking for governance. The project follows a systematic design and implementation approach using the **ServiceNow IT Service Management (ITSM)** platform, leveraging key modules like **Service Catalog**, **Flow Designer**, and **UI Policies**.

The final solution simplifies the laptop request process, reduces manual intervention, accelerates fulfillment, and enhances transparency between employees and the IT department. Additionally, it sets a foundation for future automation in asset requests and lifecycle management.





1. Introduction

1.1 Background of the Project

In a technology-driven organization, providing the right hardware resources to employees on time is crucial for productivity. However, many companies still depend on **manual laptop request processes** where employees email the IT department to ask for new laptops or replacements. These requests often lack standardization, causing delays in approvals, miscommunication, and difficulty in tracking.

ServiceNow—a cloud-based IT Service Management (ITSM) platform—offers a comprehensive solution for automating such service requests. Using ServiceNow's **Service Catalog**, employees can request IT assets through a standardized digital form, while the IT department can manage, track, and fulfill requests efficiently.

1.2 Why ServiceNow Was Chosen

ServiceNow was selected for this project due to the following advantages:

1. **No-Code/Low-Code Development:** Enables quick form and workflow creation without deep programming knowledge.
2. **Automation Capabilities:** Seamless integration of approvals, notifications, and tracking using **Flow Designer**.
3. **Governance and Compliance:** Maintains detailed logs, version control, and auditing of changes.
4. **Scalability:** Can be easily extended to include other assets and IT services in the future.
5. **User-Friendly Interface:** Provides an easy-to-use Service Catalog that employees can access from the self-service portal.

Business Requirement

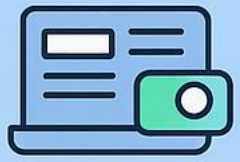
Efficient Laptop Requesting

Why ServiceNow?

Automation & Workflows

- Auto-assign tasks
- Approvals flow
- Reduce manual work

Dynamic Forms & UI Policies



- Show/hide fields dynamically
- Ensure accuracy

Centralized Request Mgmt

- One platform for requests
- Easy tracking

Benefits Achieved

- ✓ Faster fulfillment
- ✓ Accurate data capture
- ✓ Better user experience
- ✓ Reduced admin effort

1.3 Objectives

- To design and implement a **Service Catalog item** for requesting laptops.
- To ensure accurate and dynamic data collection using **UI Policies** and **Client Scripts**.
- To automate the **workflow** for request approvals and fulfillment.
- To maintain transparency and governance through notifications and audit trails.
- To enhance user experience with a reset feature and guided form instructions.

2. Problem Statement

The current laptop request process in the organization is entirely manual. Employees either email or verbally inform the IT department about their needs. This results in:

- **Delay** in request approvals and laptop delivery.
- **Miscommunication** between requesters, approvers, and IT teams.
- **Incomplete or incorrect data**, as users may forget to mention specific requirements.
- **Lack of visibility** into the current status of requests.
- **No tracking or audit trail** for governance or compliance purposes.

Therefore, there is a clear need for an **automated, efficient, and transparent** system that enables employees to request laptops through a **ServiceNow Service Catalog**. The system should be intelligent enough to guide users with dynamic fields, ensure accurate data entry, and provide complete tracking from request submission to fulfillment.

3. Methodology / System Design

3.1. Requirement Gathering

Purpose:

To fully understand the **existing manual laptop request process** and identify all the improvements needed through automation in ServiceNow.

Activities:

- Interview stakeholders (employees, managers, IT support staff).
- Analyze current pain points — delays, missing data, lack of approval visibility.
- Define the data fields required:
    - Employee Name, Department, Laptop Type, Delivery Date, etc.
- Specify **form behavior**:
    - Dynamic visibility (show/hide fields based on user input).
    - Mandatory fields and validation logic.
- Define **workflow logic**:
    - Who approves the request.
    - How IT receives and fulfills it.
    - When notifications should be sent.

Outcome:

A **clear set of functional and technical requirements** to be implemented in ServiceNow.

3.2. System Design

Purpose:

To create a **blueprint** for how the system will look, behave, and integrate across components.

Activities:

- Design the **Service Catalog Form Layout**:
  - Logical arrangement of variables for clarity.
  - Intuitive flow for the end-user.
- Design **Workflow Architecture**:
  - Define approval flow (Manager → IT Fulfillment).
  - Map notifications and automated tasks.
- Define **Data Flow Architecture**:
  - How information moves between tables (Request → Requested Item → Task).
  - How request data is stored and retrieved.

Outcome:

A detailed **system design document** and architecture diagram guiding the development phase.

3.3. Development

Purpose:

To build and configure all components in ServiceNow as per the design.

Activities:

- **Create Catalog Item:**
  - Name: *Laptop Request*
  - Description, Category, and Active status configured.
- **Add Variables:**
  - Employee Name, ID, Department, Laptop Type, Accessories, Delivery Date.
- **Configure UI Policies:**
  - Control field visibility and mandatory status dynamically.
- **Add Catalog Client Scripts:**
  - Add form reset and validation logic.
- **Build Workflow (Flow Designer):**
  - Automate approval, fulfillment, and notifications.
- **Implement Notifications:**
  - Inform requesters, approvers, and IT staff automatically.

Outcome:

A working, dynamic, and interactive **Laptop Request Catalog Item** ready for testing.

3.4. Testing

Purpose:

To ensure the catalog item functions correctly and meets all requirements before deployment.

Activities:

- **Functional Testing:**
  - Verify form fields behave dynamically.
  - Validate workflows trigger correctly.
- **User Acceptance Testing (UAT):**
  - Allow selected employees/managers to test in a sub-production instance.
  - Gather feedback for improvements.
- **Validation:**
  - Confirm that approvals, notifications, and fulfillment tasks run automatically and accurately.

Outcome:

A **stable and verified catalog item** that performs as expected in all test cases.

3.5. Deployment and Documentation

Purpose:

To move the final, tested solution into the production environment and maintain clear documentation.

Activities:

- **Deployment:**
  - Migrate the catalog item, workflow, and scripts to the production instance.
  - Ensure all dependencies (tables, flows, notifications) are active.
- **Documentation:**
  - Record all configurations, logic, and dependencies.
  - Include workflow diagrams, variable tables, and test results.
  - Store in a shared repository for future reference.

Outcome:

A **fully deployed, documented, and governed** catalog solution accessible to all employees.

4. System Architecture

4.1. Presentation Layer (User Interface)

This is the **front-end layer** where end-users (employees) directly interact with the system through the **Service Catalog**.

Purpose:

To provide an easy-to-use, dynamic interface that captures employee inputs accurately and guides them through the request process.

Main Components:

- **Service Catalog Form:**
  - The form where employees request laptops by providing details such as model, type, justification, and delivery date.
  - Fields are well-organized for clarity and simplicity.
- **Dynamic Field Visibility (UI Policies):**
  - Fields appear or hide based on the user's choices to ensure a clean and relevant form.
  - **Examples:**
    - If *"MacBook Pro"* is selected → Show a field for *"Business Justification."*
    - If *"Windows Laptop"* is selected → Hide the justification field.
- **Form Reset Button:**
  - Allows users to clear all entered data instantly and start over, improving the user experience.

Key Functions:

- Collects accurate input data from employees.
- Guides users through conditional, dynamic form behavior.
- Ensures a consistent, intuitive, and professional interface.
- Reduces user input errors and form clutter.

Benefit:

Improves **usability and data accuracy**, ensuring requests are submitted correctly the first time.

4.2. Application Logic Layer (Workflow)

This is the **brain of the system** — where automation, rules, and processes are executed behind the scenes.

Purpose:

To handle business logic, approvals, task creation, and notification automation.

Main Components:

- **Flow Designer:**
  - Core tool that automates workflows and processes.
  - **Example Workflow:**
    1. User submits a laptop request.
    2. The request goes to the manager for approval.
    3. Once approved, IT fulfillment tasks are automatically created.
    4. The employee receives a notification about the request's status.
- **Business Rules:**
  - Enforce logic before or after record updates (e.g., auto-assigning a fulfillment group, checking field values).

    - Maintain consistency and control across records.
- **Notifications:**
    - Automatically send messages to users, managers, and IT teams during each key stage — request submission, approval, and fulfillment.

Key Functions:

- Automates the entire request lifecycle.
- Ensures approvals and task assignments happen instantly.
- Reduces manual administrative work.
- Maintains a consistent process flow and accountability.

Benefit:

Delivers **end-to-end automation**, improving efficiency and eliminating bottlenecks in the approval or fulfillment process.

4.3. Data Layer (Tables)

This layer manages **data storage and organization** within ServiceNow.
It is the **foundation** that supports all catalog items, workflows, and reporting.

Purpose:

To securely store and manage all data related to laptop requests and their processing.

Main Components:

- **Core Tables Used:**
    - `sc_request` → Stores overall request metadata (who requested, when, status).
    - `sc_req_item` → Stores item-specific details (e.g., laptop type, justification, delivery date).
    - `task` → Contains fulfillment and approval tasks created by the workflow.
- **Data Stored Includes:**
    - Employee name and department.
    - Laptop type requested.
    - Business justification and delivery date.
    - Request creation date, approval status, and assigned technician.

Key Functions:

- Centralized storage for all catalog request data.
- Supports **reporting and dashboards** for tracking request status and performance.
- Enables **trend analysis** (e.g., most requested laptop models, department-wise demand).
- Provides a data foundation for integration and automation.

Benefit:

Ensures **data reliability, traceability, and analytical insight**, supporting both operational and strategic decision-making.

4.4. Integration Layer (Optional)

This layer extends the ServiceNow system by connecting it to **external enterprise systems** for end-to-end process automation.

Purpose:

To integrate ServiceNow with other business tools like Asset Management, Procurement, or Communication systems for improved coordination and data sharing.

Possible Integrations:

- **Asset Management System:**
  - Automatically checks hardware availability in inventory.
  - Updates asset records once laptops are issued to employees.
- **Procurement System (e.g., SAP, Oracle):**
  - If no stock is available, automatically generates a purchase request in the procurement system.
- **Email or Chat Integrations:**
  - Send automatic email updates about approval and delivery.
  - Allow employees to track request status through chatbots or Outlook integration.

Key Functions:

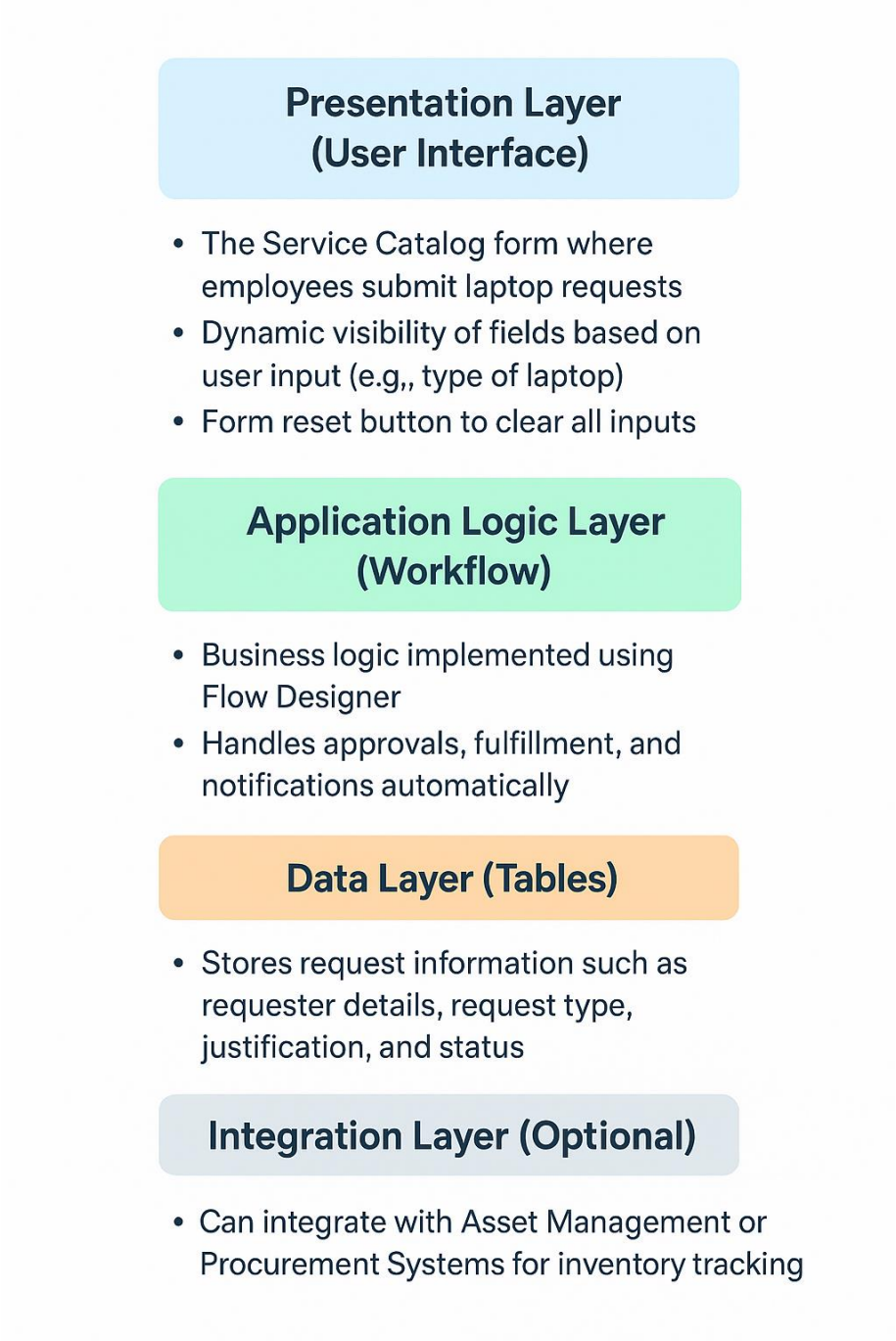
- Automates asset tracking and purchase processes.
- Prevents duplicate data entry across systems.
- Keeps inventory and procurement systems synchronized with ServiceNow.
- Enables real-time status updates and transparency.

Benefit:

Provides **seamless data flow and end-to-end lifecycle management**, enhancing efficiency and minimizing manual coordination.

Layer	Focus Area	Main Tools / Components	Key Outcomes
1. Presentation Layer	User Interface	Service Catalog Form, UI Policies, Client Scripts	Easy, dynamic, and accurate data entry
2. Application Logic Layer	Workflow & Automation	Flow Designer, Business Rules, Notifications	Automated approval and fulfillment
3. Data Layer	Storage & Reporting	`sc_request`, `sc_req_item`, `task` tables	Reliable data storage and analytics

Layer	Focus Area	Main Tools / Components	Key Outcomes
4. Integration Layer	External Connectivity	Asset Mgmt, Procurement, Email/Chat Integration	End-to-end lifecycle and synchronization



5.User Interface (UI) and User Experience (UX)

The **Laptop Request Form** was designed with simplicity and clarity as primary goals:

- **Fields Included:**

    - Employee Name
    - Department
    - Employee ID
    - Laptop Type (Standard, Developer, Executive, etc.)
    - Configuration Options (RAM, Storage, OS)
    - Accessories (Mouse, Keyboard, Bag)
    - Business Justification
    - Preferred Delivery Date
- **Dynamic Field Behavior:**
    - Certain fields like "Accessories" or "Configuration" appear only if the user selects specific laptop types.
- **Form Reset Option:**
    - Implemented using a **Catalog Client Script**, allowing users to clear all inputs with one click.
- **Instructional Text:**
    - Clear tooltips and labels guide users at each step, reducing input errors.
- **Validation:**
    - Mandatory fields ensure that no request is submitted with incomplete data.

6. Implementation Details

6.1 Platform Setup

1. **Instance Configuration:**
    - Created a dedicated **ServiceNow Developer Instance**.
2. **User Roles:**
    - **Requester:** Employees who submit laptop requests.
    - **Approver:** Managers who review and approve requests.
    - **IT Fulfillment Team:** Responsible for assigning laptops.
3. **Groups:**
    - Created "IT Asset Management" and "Request Approvers" groups for workflow routing.

6.2 Development and Customization

1. **Service Catalog Item Creation:**
    - Created a new item named **"Laptop Request Form"** under **IT Services Catalog**.
2. **Variables:**
    - Added variables for text input, dropdowns, checkboxes, and reference fields.
    - Configured variable sets for easy reuse in future asset request forms.
3. **UI Policies and Client Scripts:**
    - Implemented conditional logic to display or hide fields dynamically.
    - Added a reset button functionality using `g_form.clearValue()` in Catalog Client Script.
4. **Record Producer / Script Include:**
    - Configured to create records automatically in the **Requested Item (sc_req_item)** table.
    - Added server-side validation to ensure correct data entry.

6.3 Workflow Implementaion

1. **Flow Designer Process:**
    - Step 1: Request submission triggers flow.
    - Step 2: System routes request to manager for approval.
    - Step 3: Upon approval, IT fulfillment task is generated.
    - Step 4: Notifications sent to all stakeholders.
    - Step 5: Request marked as "Closed Complete" once fulfilled.
2. **Notifications:**
    - **On Submission:** Email to requester confirming receipt.
    - **On Approval/Rejection:** Notification to requester and IT team.
    - **On Fulfillment:** Notification to requester confirming completion.
3. **Audit and Governance:**
    - All changes and approvals recorded in **System Logs** and **Flow Execution History** for transparency.

7. Screenshots:

7.1 Update set:

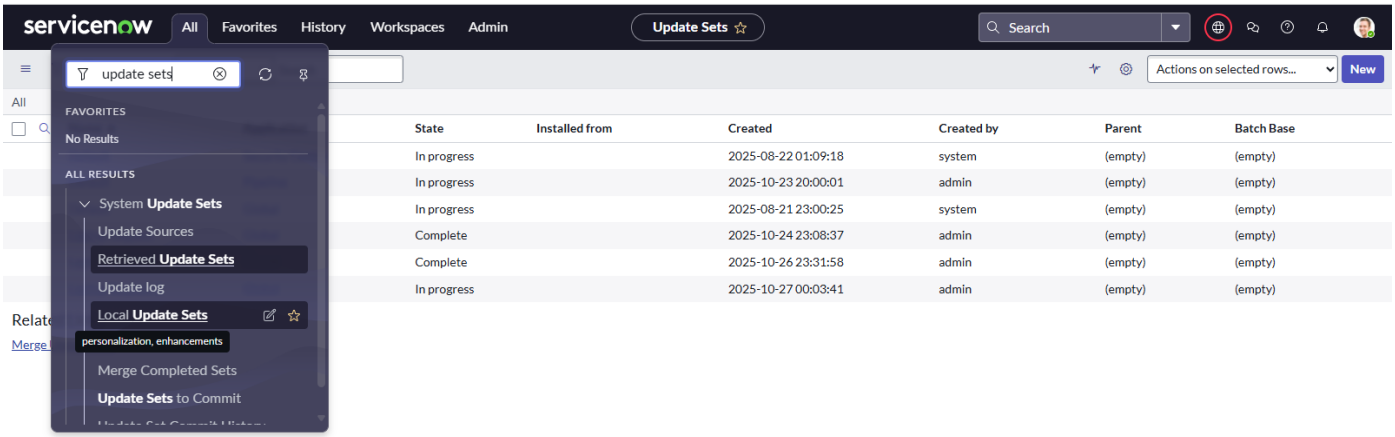
Create Local Update set

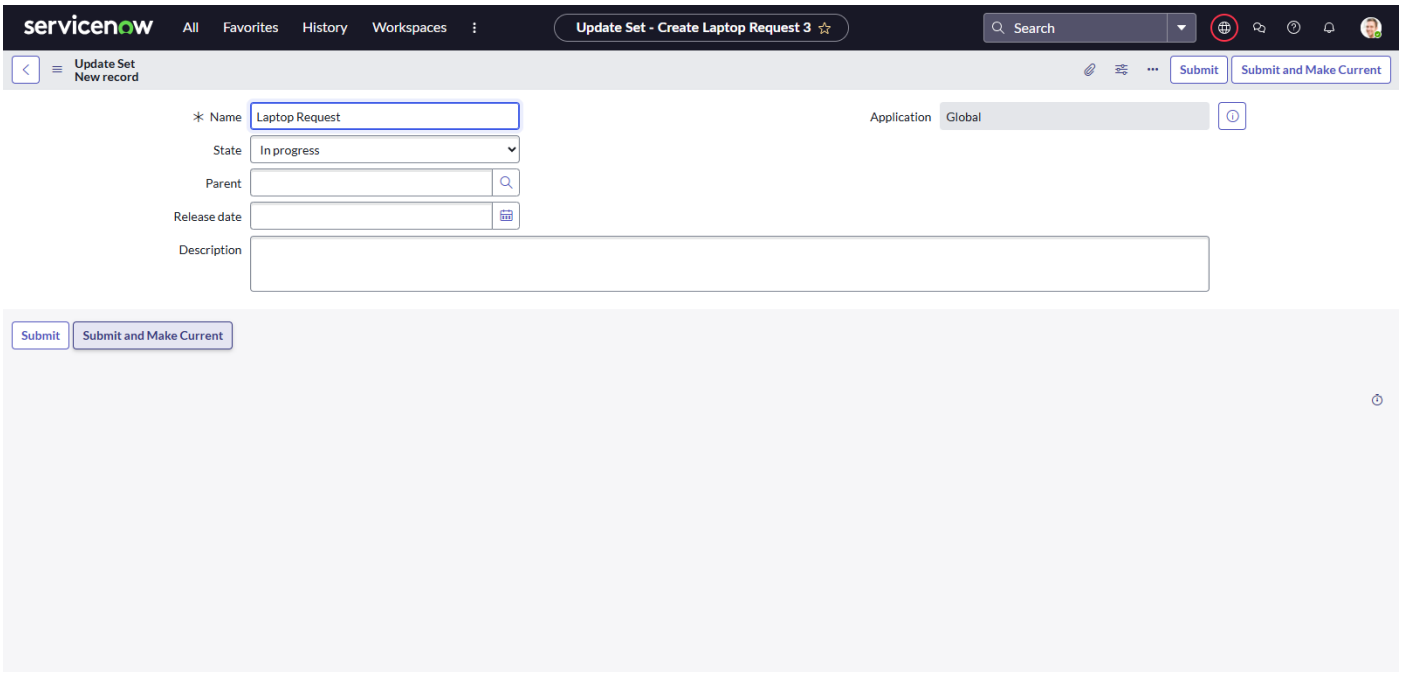
Update Sets are essential for capturing configuration and customization changes made within an instance. These update sets allow administrators and developers to **package and migrate** changes from one instance (such as *development*) to another (like *testing* or *production*) safely and efficiently.

For this project — *Automation of Laptop Request Process* — a new local update set named **"Laptop Request"** was created. This update set captures all the Service Catalog configurations, client scripts, UI policies, Flow Designer workflows, and notifications that form part of the laptop request automation solution.

Steps to Create the Local Update Set

1. **Navigate to the Update Set Module:**
    - In the **Application Navigator**, search for:
    - `System Update Sets → Local Update Sets`
2. **Create a New Update Set:**
    - Click **New** to create a new record.
    - Fill in the following details:
        - **Name:** `Laptop Request`
        - **Description:** `This update set contains all configurations, scripts, and workflows developed for the Laptop Request Service Catalog item.`
        - **State:** `In Progress`
3. **Save the Update Set:**
    - Click **Submit** or **Save** to record your new update set.
    - Once saved, it will appear in the list of *Local Update Sets*.
4. **Make the Update Set Current:**
    - Open the newly created **Laptop Request** update set.
    - Click the **Make Current** button.
    - This ensures that any changes or configurations performed after this step are captured automatically under the *Laptop Request* update set.





7.2 Service Catalog Item:

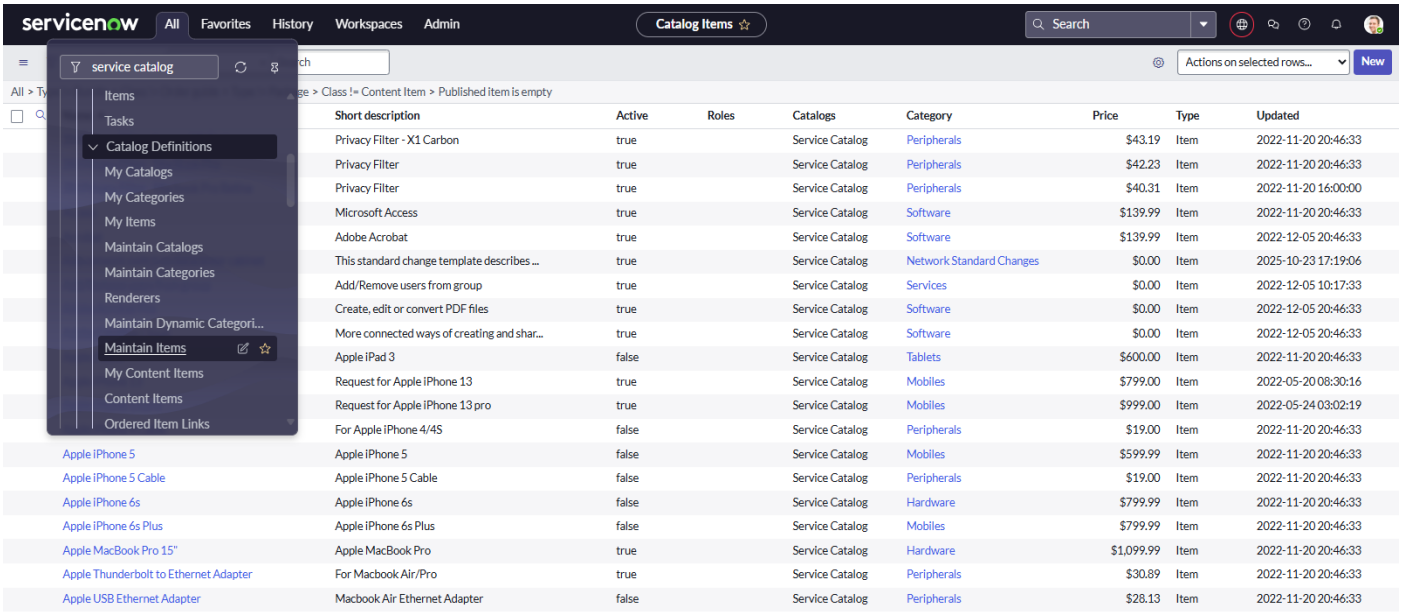
Create Service Catalog Item

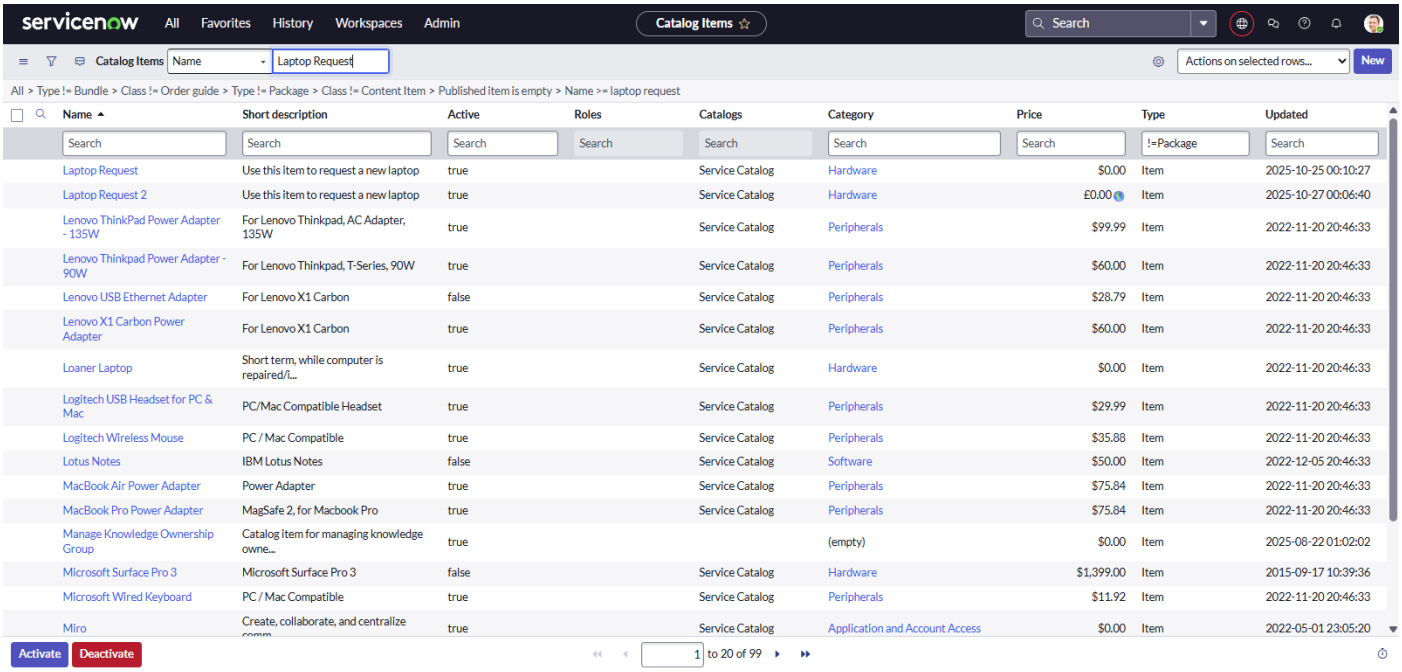
The **Laptop Request** catalog item in ServiceNow allows employees to quickly and efficiently request laptops through an online form instead of using a manual process. It ensures faster approvals, accurate data collection, and an improved user experience with dynamic fields and validations.

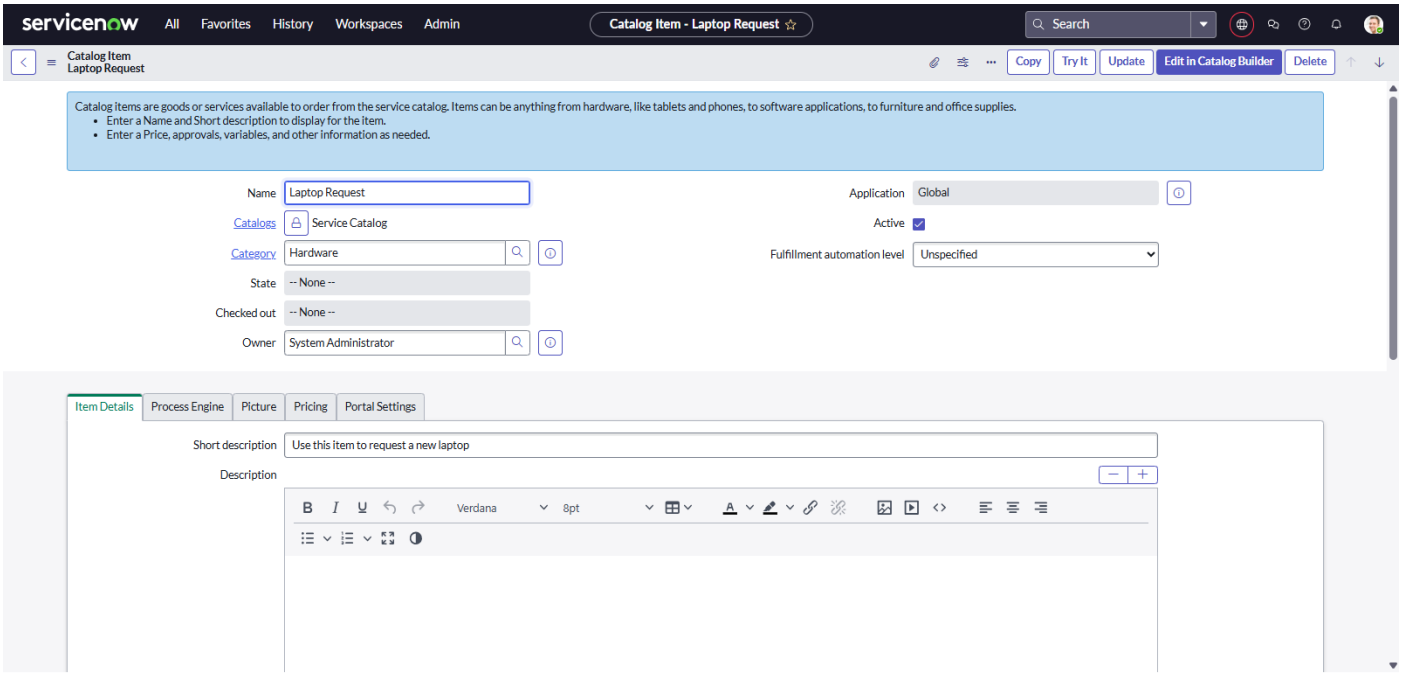
Steps to Create the Catalog Item

1. **Navigate to Service Catalog → Maintain Items → New**
    - Open a new catalog item form to begin creation.

2. **Enter Basic Information**
    - Name: *Laptop Request*
    - Add Short Description and Detailed Description.
    - Select Category: *IT Services*
    - Mark the item as **Active**.
3. **Add Variables (Form Fields)**
    - Employee Name
    - Employee ID
    - Department
    - Laptop Type
    - Accessories
    - Delivery Date
4. **Configure UI Policies**
    - Control visibility and mandatory fields.
    - Example: Show extra fields only when *Laptop Type = Developer*.
5. **Add Catalog Client Scripts**
    - Add a **Form Reset** button to clear inputs.
    - Add **Validation Scripts** (e.g., future date check).
6. **Link the Item to a Workflow**
    - Use **Flow Designer or Workflow Editor**.
    - Include steps for approval, task assignment, and notifications.
7. **Test the Item in Service Portal**
    - Check form visibility changes dynamically.
    - Verify workflow, approvals, and notifications.
8. **Publish for End Users**
    - Make it available under *IT Services* category for employees to request laptops.







7.3 Add variables:

Variables are used in the *Laptop Request* catalog item to collect all required details from employees. Each variable corresponds to a **form field** that captures specific input.

Steps to Add Variables

1. **Open the Laptop Request Catalog Item**
   - Navigate to **Service Catalog → Maintain Items**.
   - Open the **Laptop Request** catalog item.
2. **Access the Variables Related List**
   - Scroll down to the **Variables** tab.
   - Click **New** to create a new variable.
3. **Enter Variable Details**

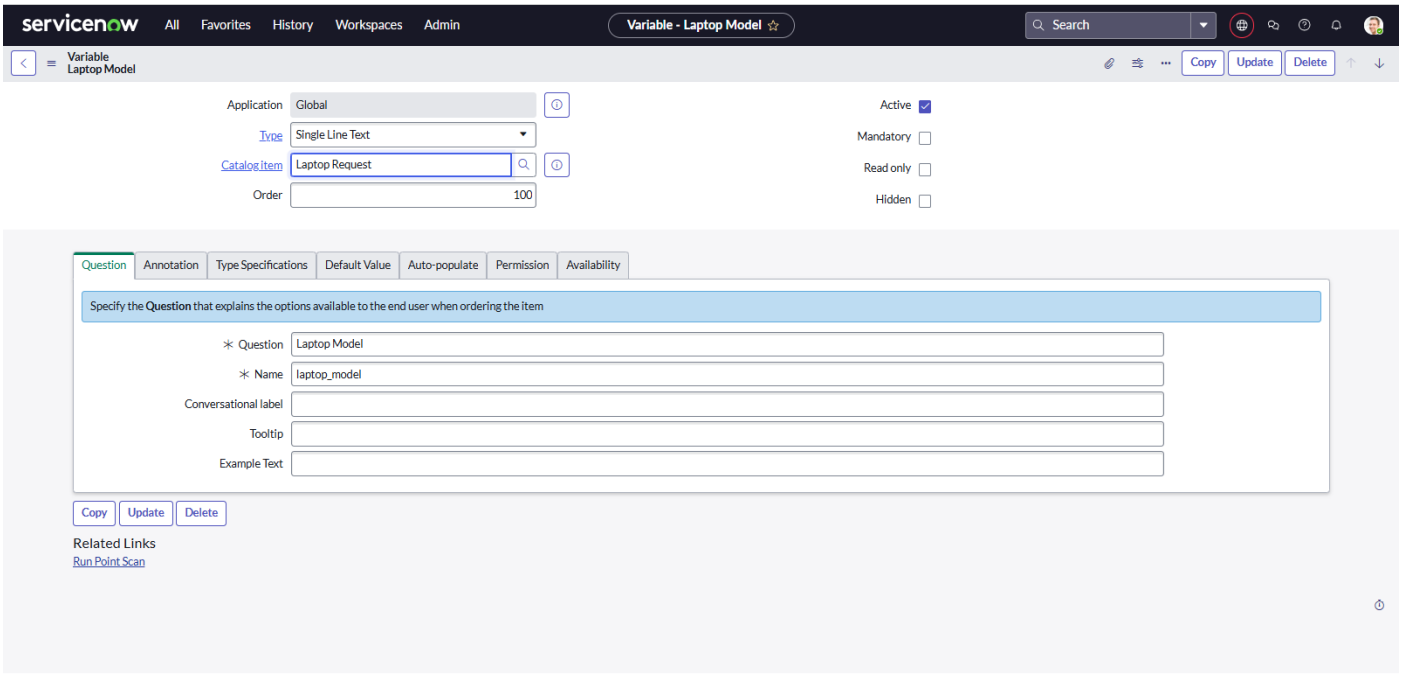
- Provide **Variable Name** (system name).
- Enter **Question Text** (label shown to user).
- Select **Type** (e.g., Single Line Text, Choice, Multiple Choice, Date).

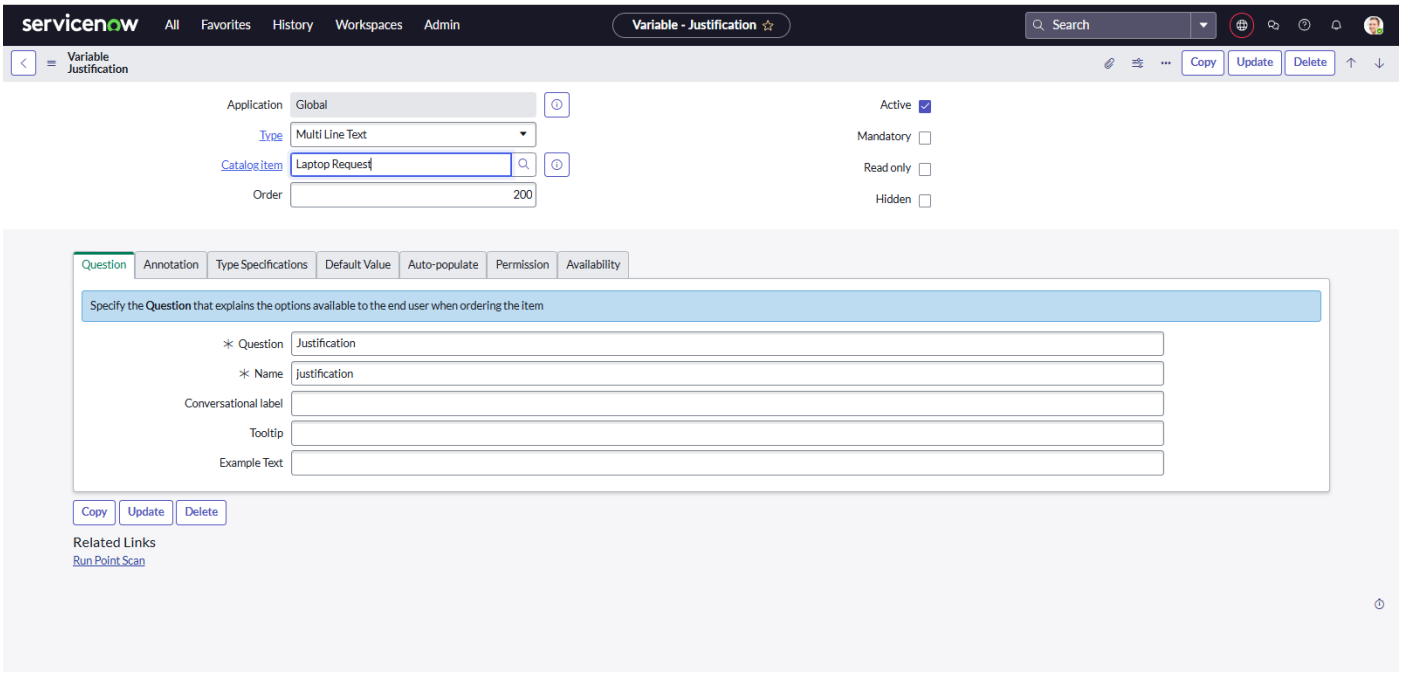
4. **Set Mandatory Fields**
   - Enable **Mandatory = True** for required inputs (e.g., Employee Name, Laptop Type).
5. **Add Help Text or Instructions**
   - Include short descriptions or tooltips to guide users while filling the form.
6. **Repeat for All Required Variables**
   - Create variables such as:
     - Employee Name
     - Employee ID
     - Department
     - Laptop Type
     - Accessories
     - Delivery Date

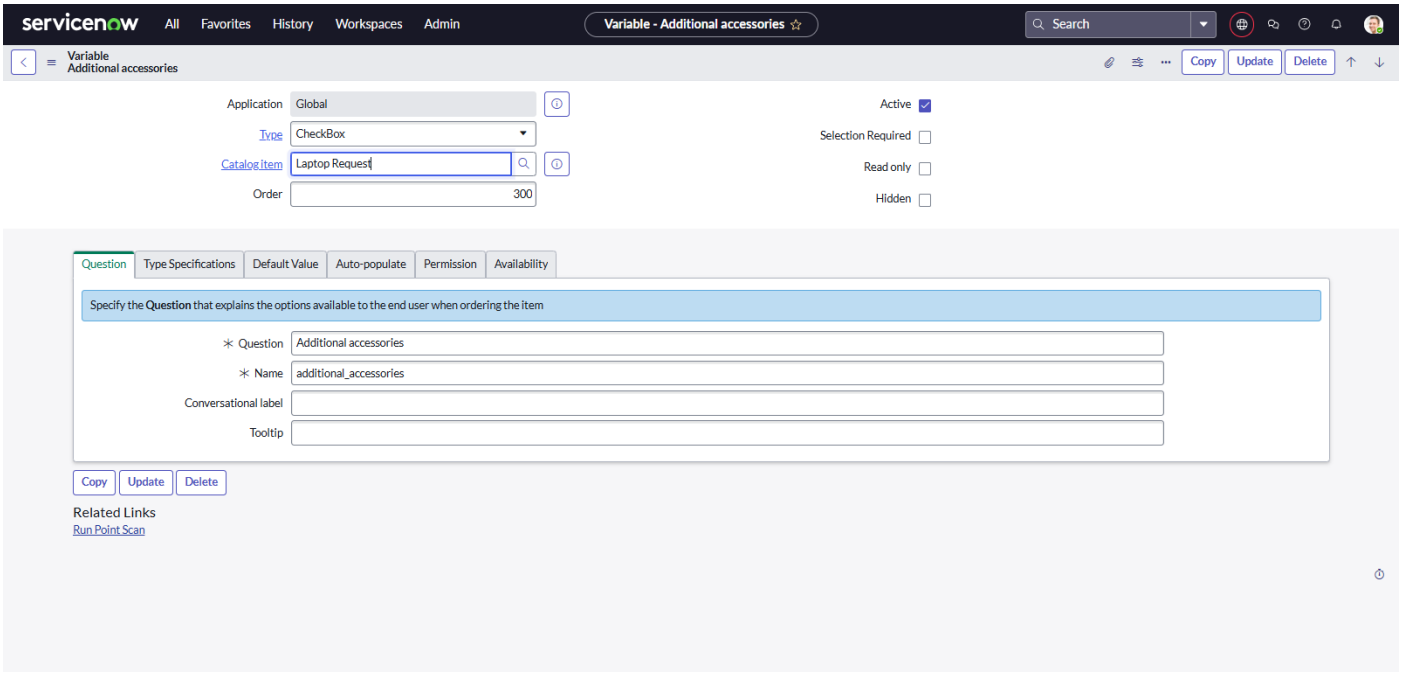
Variables Created

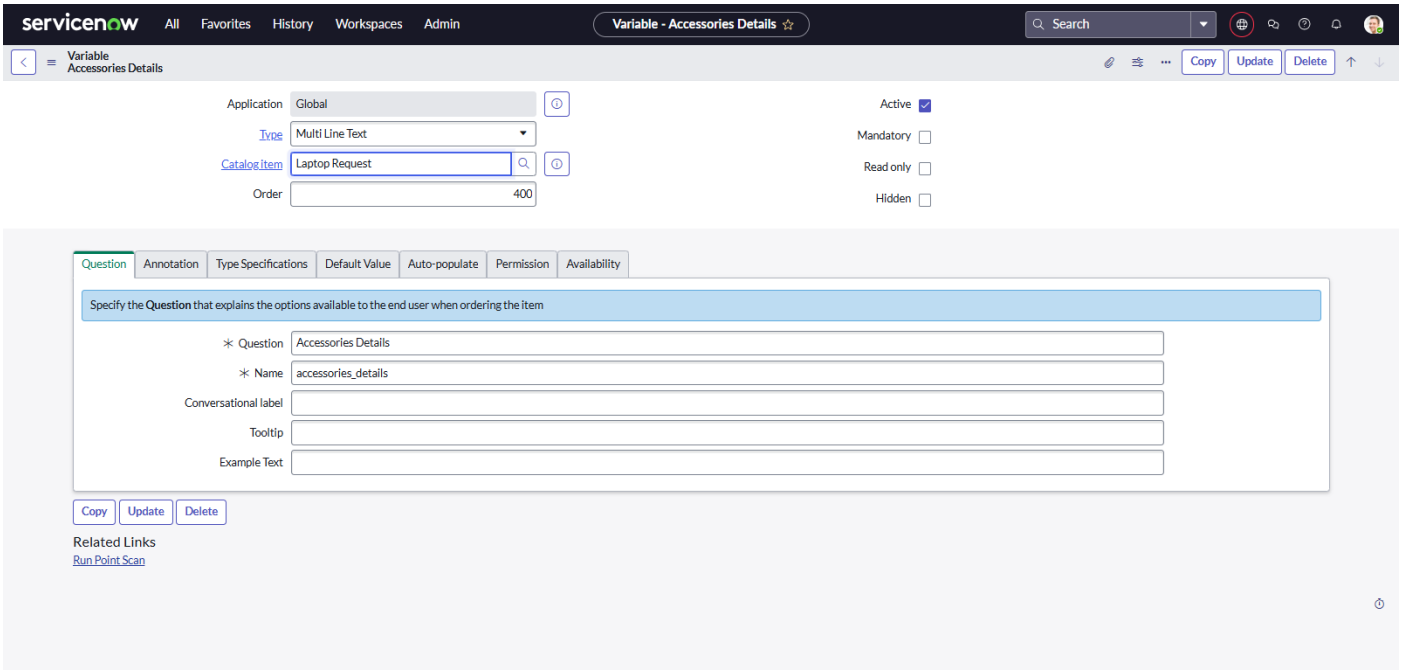
- **Employee Name** – *Single Line Text* (auto-filled or user-entered).
- **Employee ID** – *Single Line Text* (mandatory).
- **Laptop Type** – *Single Line Text* (user specifies laptop type).
- **Accessories Required** – *Check Box* (for selecting items like mouse, keyboard, etc.).
- **Business Justification** – *Multi Line Text* (for entering reason for the request).
- **Additional Notes** – *Multi Line Text* (optional for extra comments).

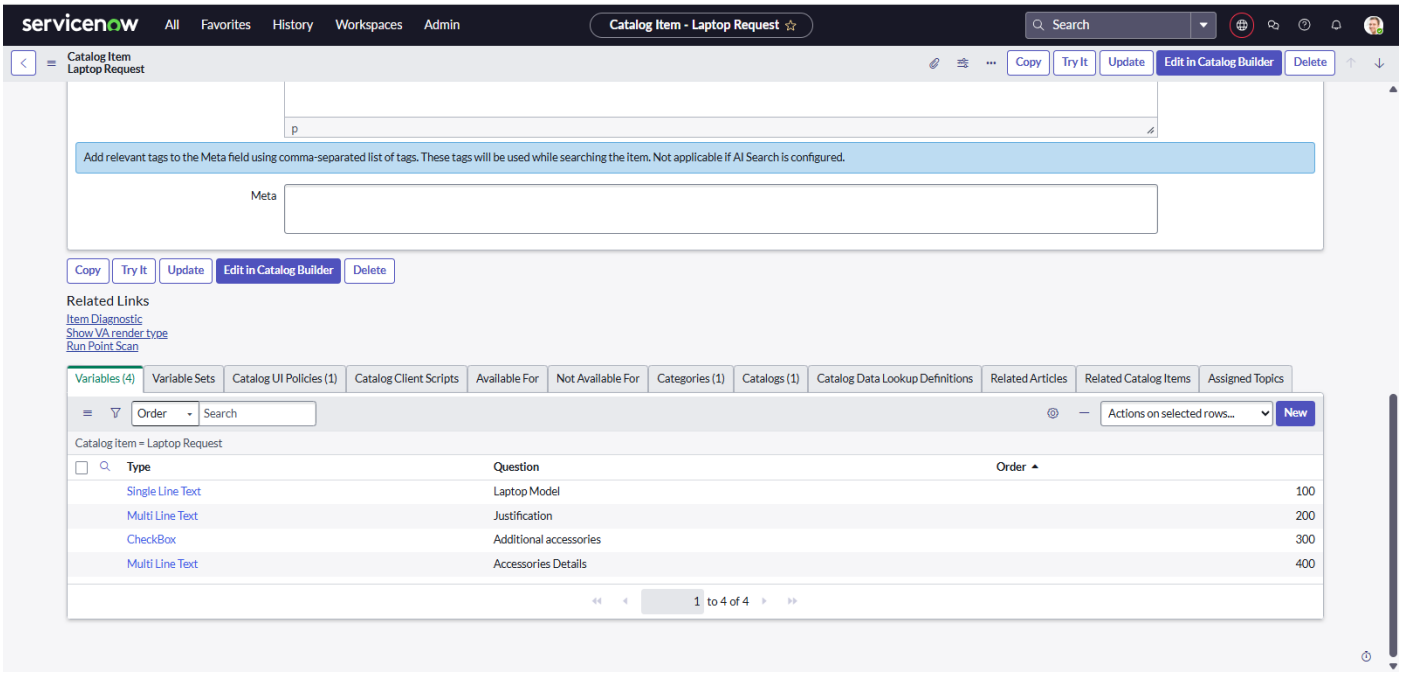
These variables make the **Laptop Request** form clear, simple, and interactive — ensuring accurate data collection and a smooth request submission process.




servicenow
All
Favorites
History
Workspaces
Admin
Variable - Justification
Search
Variable
Justification
Copy
Update
Delete
Application
Global
Type
Multi Line Text
Catalog item
Laptop Request
Order
200
Active
Mandatory
Read only
Hidden
Question
Annotation
Type Specifications
Default Value
Auto-populate
Permission
Availability
Specify the Question that explains the options available to the end user when ordering the item
Question
Justification
Name
justification
Conversational label
Tooltip
Example Text
Copy
Update
Delete
Related Links
Run Point Scan


servicenow
All
Favorites
History
Workspaces
Admin
Variable - Additional accessories
Search
Variable
Additional accessories
Copy
Update
Delete
Application
Global
Type
CheckBox
Catalog item
Laptop Request
Order
300
Active
Selection Required
Read only
Hidden
Question
Type Specifications
Default Value
Auto-populate
Permission
Availability
Specify the Question that explains the options available to the end user when ordering the item
Question
Additional accessories
Name
additional_accessories
Conversational label
Tooltip
Copy
Update
Delete
Related Links
Run Point Scan


servicenow
All
Favorites
History
Workspaces
Admin
Variable - Accessories Details
Search
Variable
Accessories Details
Copy
Update
Delete
Application
Global
Type
Multi Line Text
Catalog Item
Laptop Request
Order
400
Active
Mandatory
Read only
Hidden
Question
Annotation
Type Specifications
Default Value
Auto-populate
Permission
Availability
Specify the Question that explains the options available to the end user when ordering the item
Question
Accessories Details
Name
accessories_details
Conversational label
Tooltip
Example Text
Copy
Update
Delete
Related Links
Run Point Scan


servicenow
All
Favorites
History
Workspaces
Admin
Catalog Item - Laptop Request
Search
Catalog Item
Laptop Request
Copy
Try It
Update
Edit in Catalog Builder
Delete
p
Add relevant tags to the Meta field using comma-separated list of tags. These tags will be used while searching the item. Not applicable if AI Search is configured.
Meta
Copy
Try It
Update
Edit in Catalog Builder
Delete
Related Links
Item Diagnostic
Show VA render type
Run Point Scan
Variables (4)
Variable Sets
Catalog UI Policies (1)
Catalog Client Scripts
Available For
Not Available For
Categories (1)
Catalogs (1)
Catalog Data Lookup Definitions
Related Articles
Related Catalog Items
Assigned Topics
Order
Search
Actions on selected rows...
New
Catalog item = Laptop Request
Type
Question
Order
Single Line Text
Laptop Model
100
Multi Line Text
Justification
200
CheckBox
Additional accessories
300
Multi Line Text
Accessories Details
400
1 to 4 of 4

7.4 UI Policy:

Create Catalog Ui policies

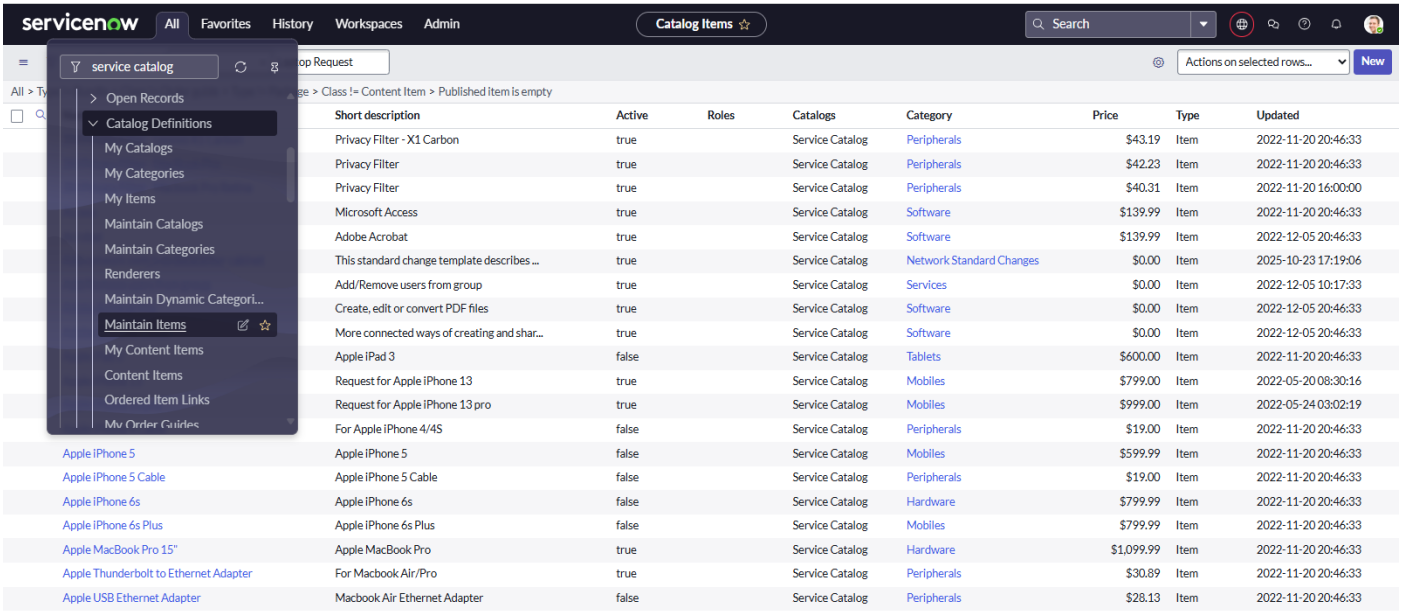
The **UI Policy** in ServiceNow is used to make catalog forms more interactive and user-friendly. It controls how fields behave — such as becoming visible, hidden, read-only, or mandatory — based on user inputs. For the **Laptop Request** catalog item, UI Policies were created to make the form dynamic. For example, when a user selects "Developer Laptop," additional configuration fields appear automatically, while for "Standard Laptop," they remain hidden. This improves accuracy and ensures users only fill in relevant details.

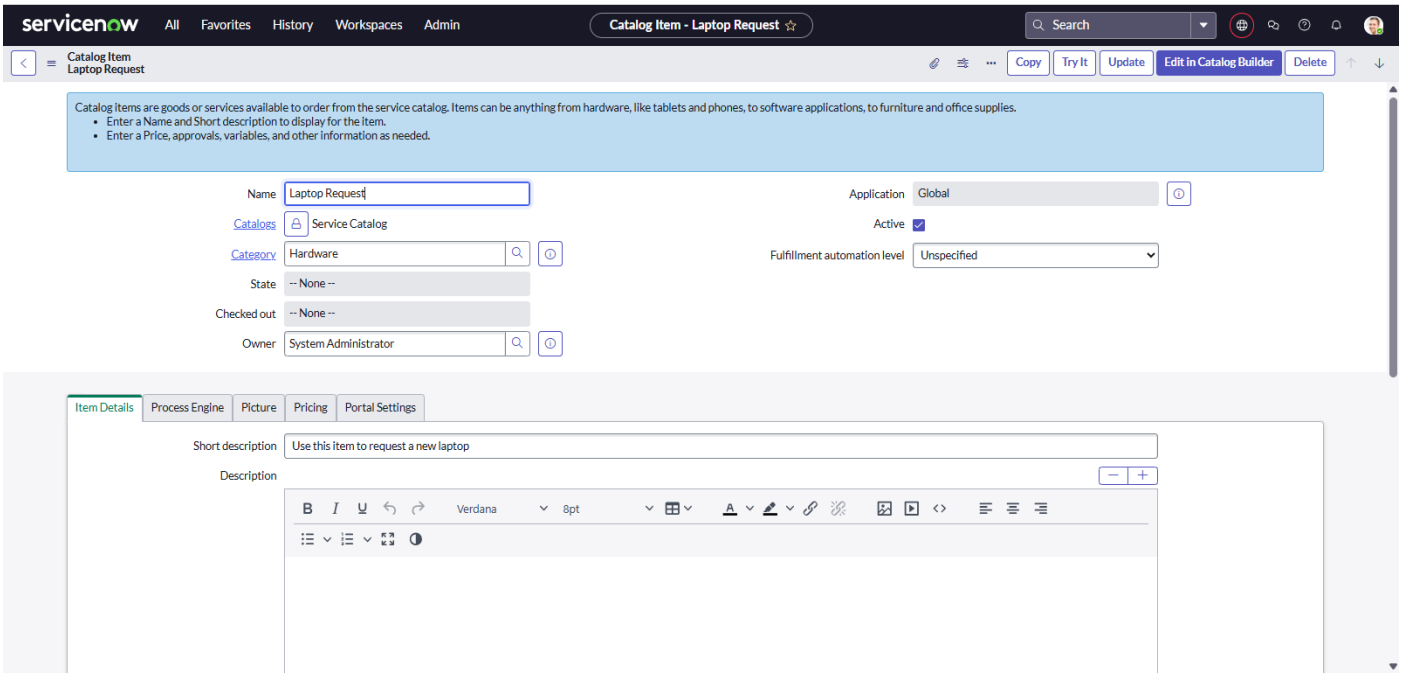
Steps to Create Catalog UI Policies

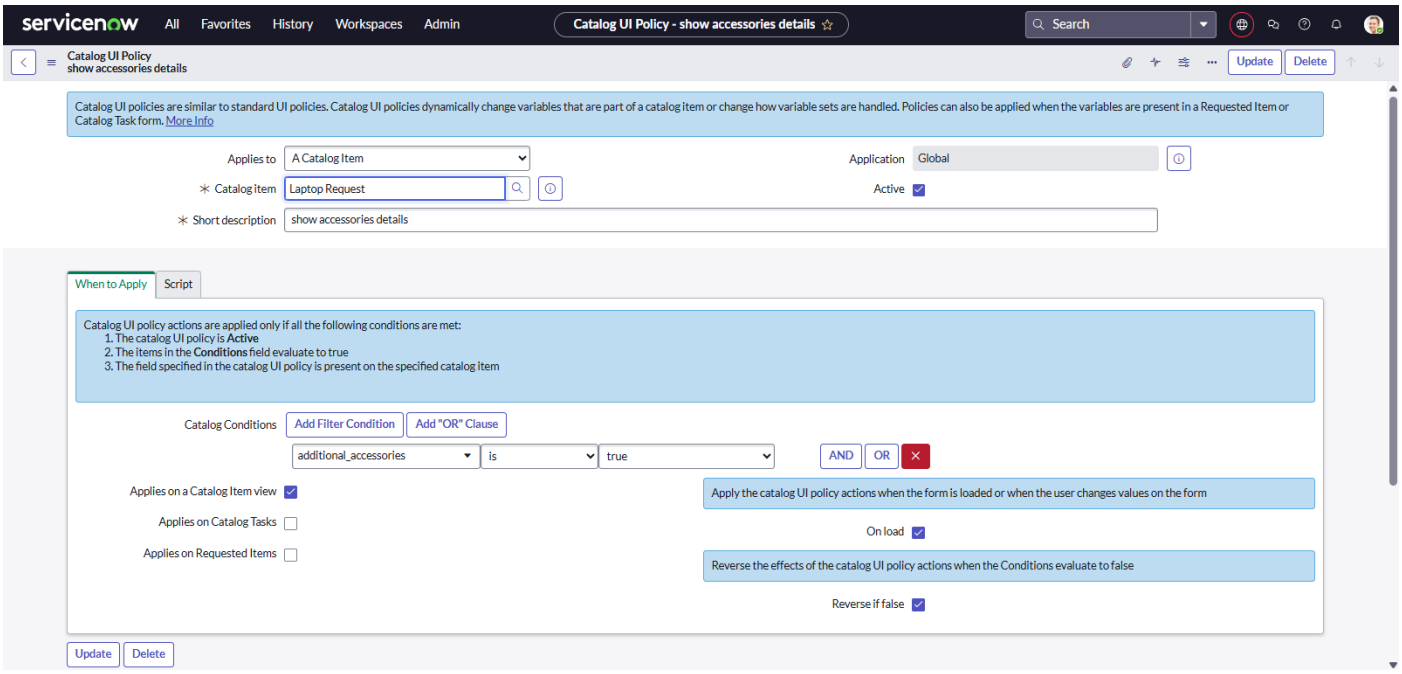
- **Navigate to the Catalog Item**
  - Go to **Service Catalog → Catalog Definitions → Maintain Items**.
  - Search and open the **Laptop Request** catalog item.
- **Create a New UI Policy**
  - Scroll down to the **Catalog UI Policies** related list.
  - Click **New** to create a new policy.
- **Enter Policy Details**
  - In the **Name** field, enter **Laptop Request – Dynamic Fields**.
  - Set **Applies to** as *Catalog Item*.
  - Ensure the **Active** checkbox is selected so that the policy is applied.
  - Optionally, check **Global** if the same policy should apply across multiple catalog items.
- **Define the Condition**
  - In the **Condition** field, specify when the policy should trigger.
  - Example:
    - **Condition:** *Laptop Type* is *Developer*.
    - This means the policy activates when the user selects "Developer" as the laptop type.
- **Save the Policy**
  - Click **Submit** or **Update** to save the new UI Policy record.
- **Add UI Policy Actions**
  - After saving, scroll down to the **Catalog UI Policy Actions** related list.
  - Click **New** to add an action for a specific variable.
  - In the **Variable Name**, select the field you want to control (e.g., *Configuration Details*).
  - Choose the desired behavior:
    - **Mandatory:** True (forces the user to fill in the field).
    - **Visible:** True (shows the field).

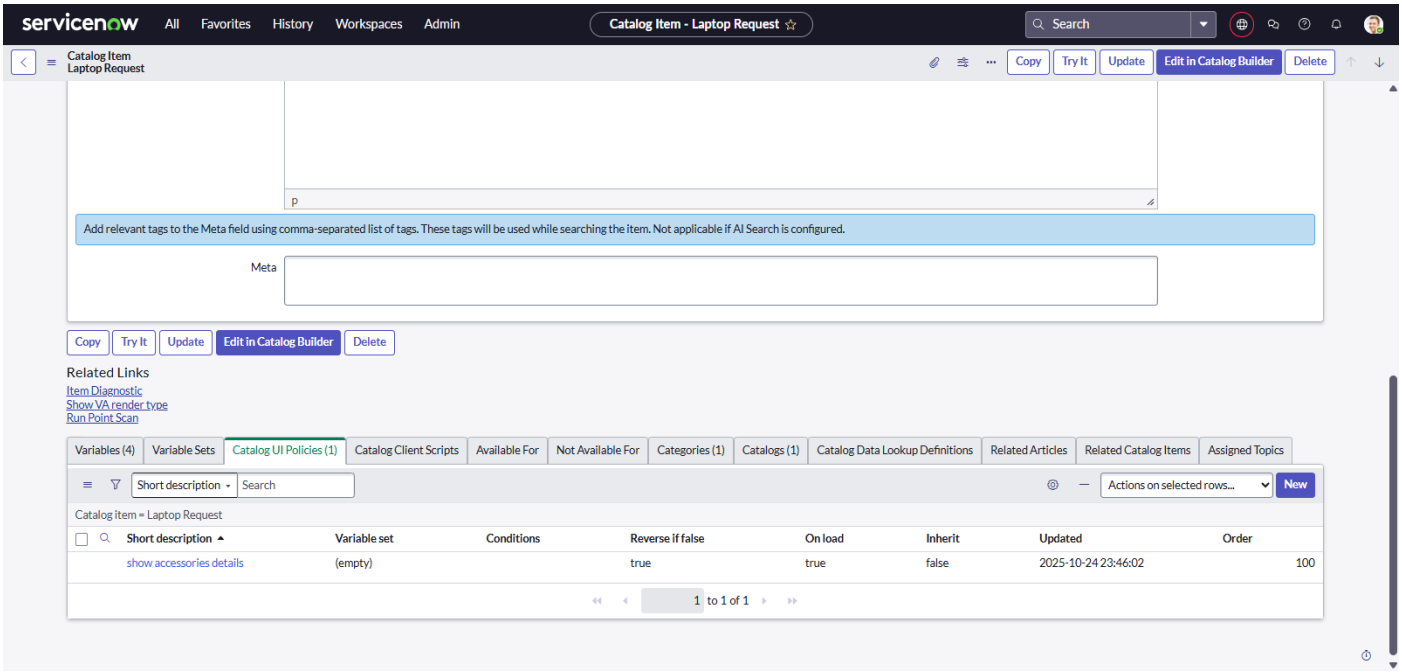
- **Read Only:** False (allows input).
- Save the action.

- **Test the Policy**
  - Open the **Laptop Request** form in the **Service Portal**.
  - Select different laptop types (Standard, Developer, Executive) and verify that fields appear or hide dynamically based on the policy.






servicenow
All
Favorites
History
Workspaces
Admin
Catalog UI Policy - show accessories details
Search
Catalog UI Policy
show accessories details
Update
Delete
Catalog UI policies are similar to standard UI policies. Catalog UI policies dynamically change variables that are part of a catalog item or change how variable sets are handled. Policies can also be applied when the variables are present in a Requested Item or Catalog Task form. More Info
Applies to
A Catalog Item
Application
Global
Catalog item
Laptop Request
Active
Short description
show accessories details
When to Apply
Script
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
Catalog Conditions
Add Filter Condition
Add "OR" Clause
additional_accessories
is
true
AND
OR
Applies on a Catalog Item view
Applies on Catalog Tasks
Applies on Requested Items
Apply the catalog UI policy actions when the form is loaded or when the user changes values on the form
On load
Reverse the effects of the catalog UI policy actions when the Conditions evaluate to false
Reverse if false
Update
Delete


servicenow
All
Favorites
History
Workspaces
Admin
Catalog Item - Laptop Request
Search
Catalog Item
Laptop Request
Copy
Try It
Update
Edit in Catalog Builder
Delete
Add relevant tags to the Meta field using comma-separated list of tags. These tags will be used while searching the item. Not applicable if AI Search is configured.
Meta
Copy
Try It
Update
Edit in Catalog Builder
Delete
Related Links
Item Diagnostic
Show VA render type
Run Point Scan
Variables (4)
Variable Sets
Catalog UI Policies (1)
Catalog Client Scripts
Available For
Not Available For
Categories (1)
Catalogs (1)
Catalog Data Lookup Definitions
Related Articles
Related Catalog Items
Assigned Topics
Short description
Search
Actions on selected rows...
New
Catalog item = Laptop Request
Short description
Variable set
Conditions
Reverse if false
On load
Inherit
Updated
Order
show accessories details
(empty)
true
true
false
2025-10-24 23:46:02
100
1 to 1 of 1

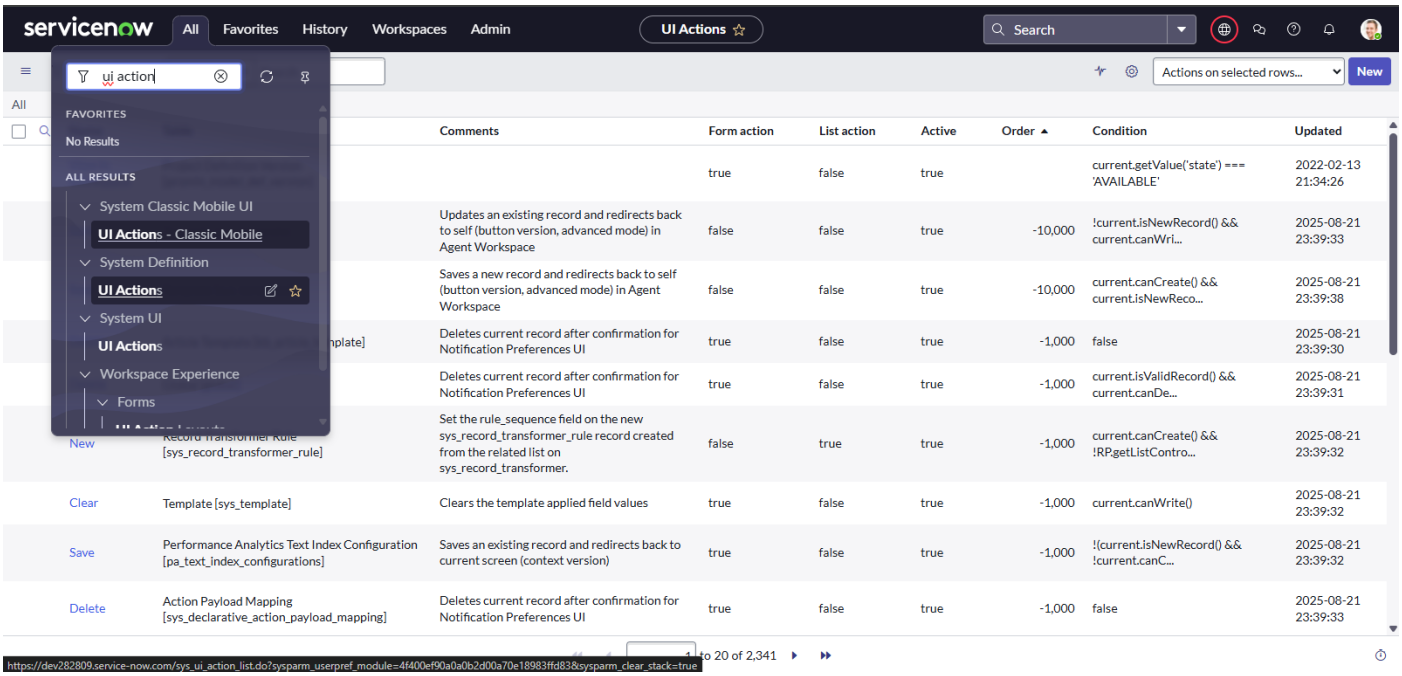
7.5 UI Action:

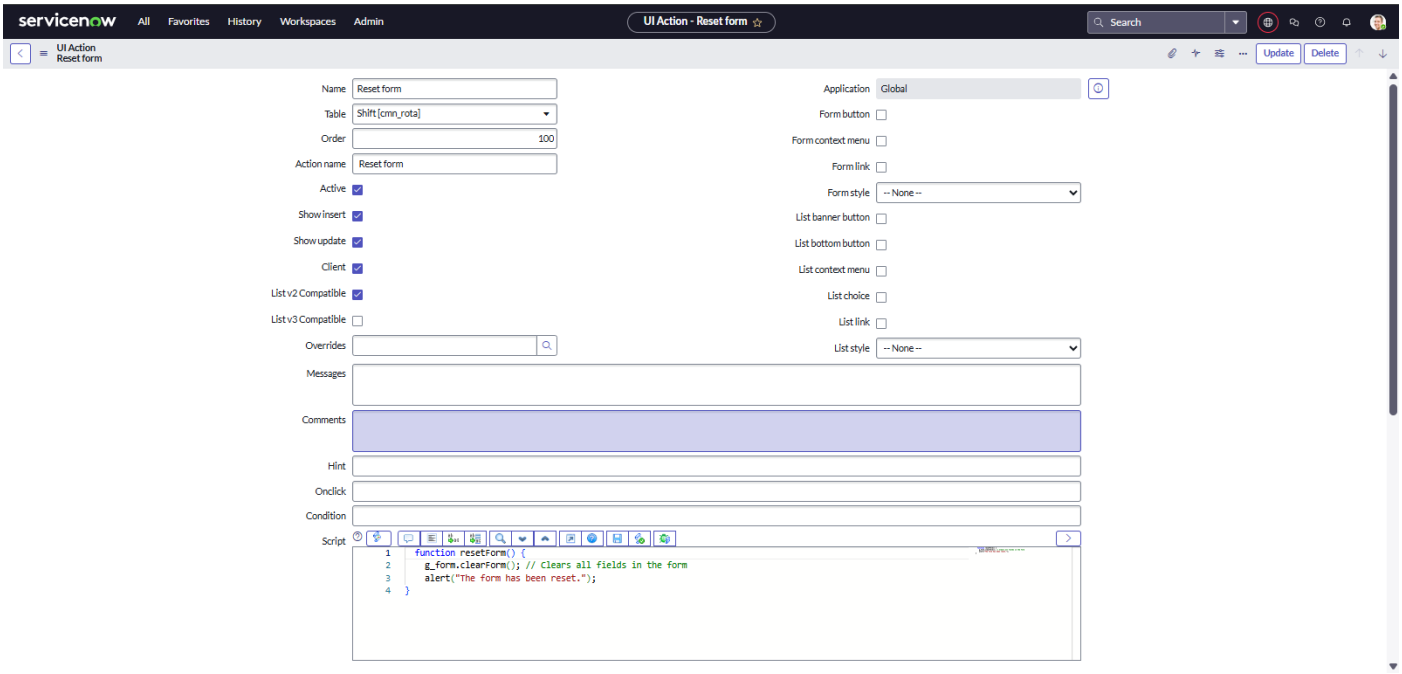
Create Ui action:

UI Actions are used to add buttons, links, or menu options that allow users to perform specific functions directly from a form or list.
For the **Laptop Request** catalog item, a **UI Action** was created to add a **"Reset Form"** button. This feature allows users to quickly clear all the entered data on the form and start over, ensuring that incorrect information can be easily removed without having to manually delete each field value. This enhances the overall usability and improves the user experience by saving time and reducing input errors.

Detailed Steps to Create a UI Action

1. **Navigate to the Catalog Item**
   - Go to **Service Catalog → Catalog Definitions → Maintain Items**.
   - Search for and open the catalog item named **Laptop Request**.
2. **Open the UI Actions Related List**
   - Scroll down to the **UI Actions** section at the bottom of the form.
   - Click **New** to create a new UI Action.
3. **Enter Basic Information**
   - In the **Name** field, enter **Laptop Request – Reset Form**.
   - Set the **Action Name** as `reset_form`.
   - Enter a **Label** as *Reset Form* — this will appear as the button text on the catalog form.
   - Choose **Form button** as the **UI Action Type**, ensuring the button appears directly on the catalog form.
   - Make sure the **Active** checkbox is selected to enable this action.
4. **Add the Client Script**
   - In the **Script** field, write a **client-side script** to clear all form fields when the button is clicked.
5. **Configure Display Options (Optional)**
   - You can specify conditions to display the button only in specific states or to certain users.
   - For example, you can make the button visible only when the form is being filled (not after submission).
6. **Save the UI Action**
   - Once all configurations are complete, click **Submit** or **Update** to save the UI Action.
7. **Test the UI Action**
   - Navigate to the **Service Portal** and open the **Laptop Request** form.
   - Fill in a few fields and click the **Reset Form** button.
   - Verify that all entered data is cleared instantly and that the form resets to its default state.


servicenow
All
Favorites
History
Workspaces
Admin
UI Actions
Search
ui action
FAVORITES
No Results
ALL RESULTS
System Classic Mobile UI
UI Actions - Classic Mobile
System Definition
UI Actions
System UI
UI Actions
Workspace Experience
Forms
Actions on selected rows...
New
All
Comments
Form action
List action
Active
Order
Condition
Updated
true
false
true
current.getValue('state') === 'AVAILABLE'
2022-02-13 21:34:26
Updates an existing record and redirects back to self (button version, advanced mode) in Agent Workspace
false
false
true
-10,000
!current.isNewRecord() && current.canWri...
2025-08-21 23:39:33
Saves a new record and redirects back to self (button version, advanced mode) in Agent Workspace
false
false
true
-10,000
current.canCreate() && current.isNewReco...
2025-08-21 23:39:38
Deletes current record after confirmation for Notification Preferences UI
true
false
true
-1,000
false
2025-08-21 23:39:30
Deletes current record after confirmation for Notification Preferences UI
true
false
true
-1,000
current.isValidRecord() && current.canDe...
2025-08-21 23:39:31
New
[sys_record_transformer_rule]
Set the rule_sequence field on the new sys_record_transformer_rule record created from the related list on sys_record_transformer.
false
true
true
-1,000
current.canCreate() && !RP.getListContro...
2025-08-21 23:39:32
Clear
Template [sys_template]
Clears the template applied field values
true
false
true
-1,000
current.canWrite()
2025-08-21 23:39:32
Save
Performance Analytics Text Index Configuration [pa_text_index_configurations]
Saves an existing record and redirects back to current screen (context version)
true
false
true
-1,000
!(current.isNewRecord() && !current.canC...
2025-08-21 23:39:32
Delete
Action Payload Mapping [sys_declarative_action_payload_mapping]
Deletes current record after confirmation for Notification Preferences UI
true
false
true
-1,000
false
2025-08-21 23:39:33
to 20 of 2,341
https://dev282809.service-now.com/sys_ui_action_list.do?sysparm_userpref_module=4f400ef90a0a0b2d00a70e18983ffd83&sysparm_clear_stack=true


servicenow
All
Favorites
History
Workspaces
Admin
UI Action - Reset form
Search
UI Action
Reset form
Update
Delete
Name
Reset form
Table
Shift [cmn_rota]
Order
100
Action name
Reset form
Active
Show insert
Show update
Client
List v2 Compatible
List v3 Compatible
Overrides
Application
Global
Form button
Form context menu
Form link
Form style
-- None --
List banner button
List bottom button
List context menu
List choice
List link
List style
-- None --
Messages
Comments
Hint
Onclick
Condition
Script
function resetForm() {
g_form.clearForm(); // Clears all fields in the form
alert("The form has been reset.");
}

7.6 Export Update set:

Exporting changes to another instances:

Update Sets are used to capture and move customizations—such as catalog items, UI policies, scripts, and workflows—from one instance to another (for example, from development to testing or production).
For the **Laptop Request** catalog item, an update set was created to include all related configurations so they can be easily exported and imported into another ServiceNow instance. This ensures consistency, version control, and proper governance across environments.

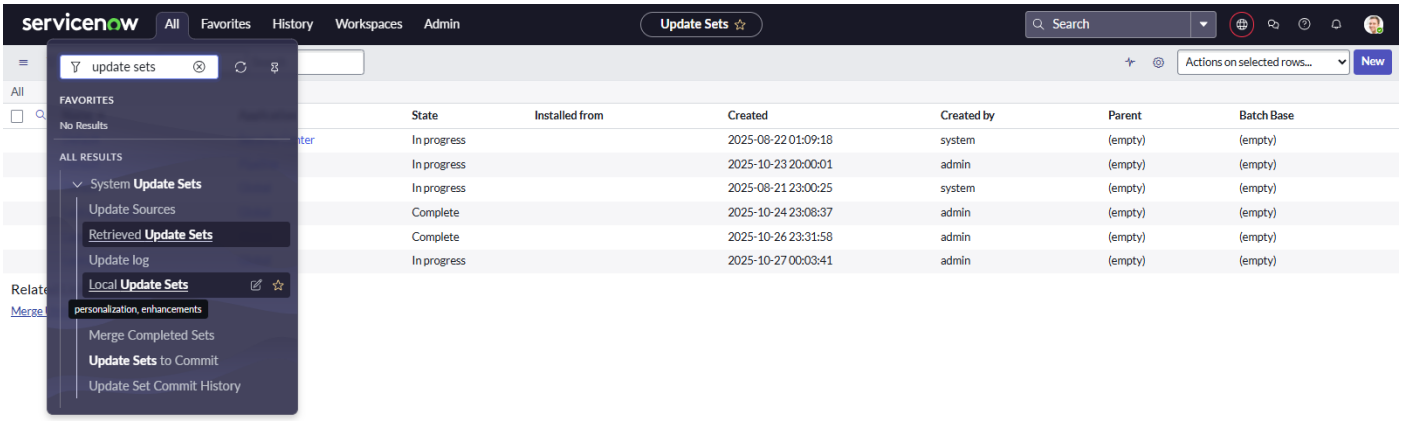
Detailed Steps to Export Update Set

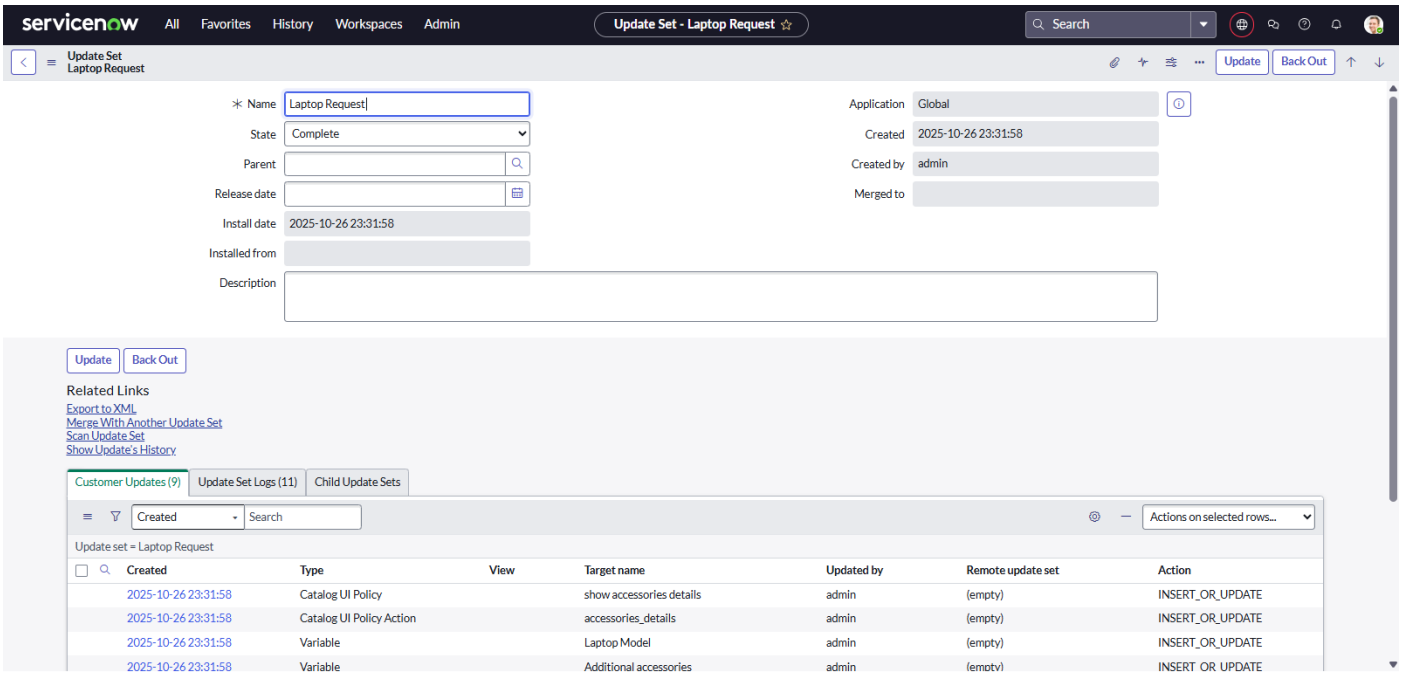
1. **Navigate to Update Sets**
    - Go to **System Update Sets → Local Update Sets** in the Application Navigator.
    - This section lists all update sets created within the current instance.
2. **Create a New Update Set**
    - Click **New** to create a new update set record.
    - Enter the **Name** as **Laptop Request**.
    - Add a short **Description** such as "Contains configurations for Laptop Request catalog item."
    - Click **Submit** to save the update set.
3. **Select the Update Set**
    - After creation, click on the **Laptop Request** update set record.
    - Click **Make Current** to ensure all changes made in the instance will be captured under this update set.
4. **Perform Customizations**
    - While the *Laptop Request* update set is active, create or modify all necessary configurations related to the project, including:
        - The **Service Catalog Item (Laptop Request)**
        - **Variables**
        - **UI Policies**
        - **UI Actions**
        - **Client Scripts** or **Workflows**
    - All these changes will automatically be recorded in the selected update set.
5. **Review Captured Changes**
    - Once all configurations are complete, navigate again to **System Update Sets → Local Update Sets**.
    - Open the **Laptop Request** update set.
    - Under the **Customer Updates** related list, verify that all modifications (Catalog Item, Scripts, etc.) are included.
6. **Export the Update Set to XML**
    - In the *Laptop Request* update set record, click the **Export to XML** button on the top-right corner of the form.
    - The system generates an XML file containing all recorded changes.
    - Save the **XML file** to your local computer for use in another instance.
7. **Import the Update Set into Another Instance**
    - Log in to the target instance (for example, a testing or production instance).
    - Navigate to **System Update Sets → Retrieved Update Sets**.
    - Click **Import Update Set from XML**.
    - Upload the previously exported XML file.
8. **Preview the Update Set**
    - After import, open the *Laptop Request* update set record.

- Click **Preview Update Set** to ensure there are no errors or conflicts.
- Resolve any warnings or missing dependencies if shown.

9. **Commit the Update Set**
   - Once the preview is successful, click **Commit Update Set** to apply all changes to the new instance.
   - The *Laptop Request* catalog item and all related components will now be available and functional in the new environment.
10. **Verify the Deployment**

- Open the **Service Catalog** in the target instance.
- Search for the **Laptop Request** item and test its form, UI Policies, and Reset functionality to ensure everything was migrated correctly.





7.7 Login to another Instance:

Retrieving the update set:

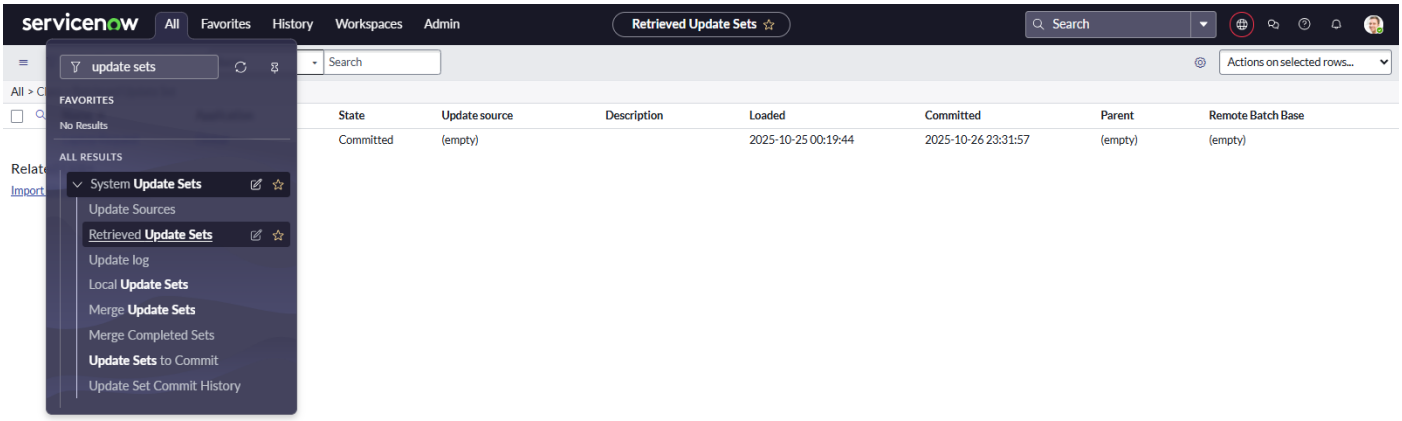
After exporting the **Laptop Request** update set from the development instance, it needs to be imported and retrieved into another instance such as **testing** or **production**.
This process is essential for maintaining consistency, governance, and version control across multiple environments. It ensures that all customizations, including the **Service Catalog item**, **variables**, **UI policies**, **UI actions**, and any associated **scripts or workflows**, are accurately transferred without manually recreating them.

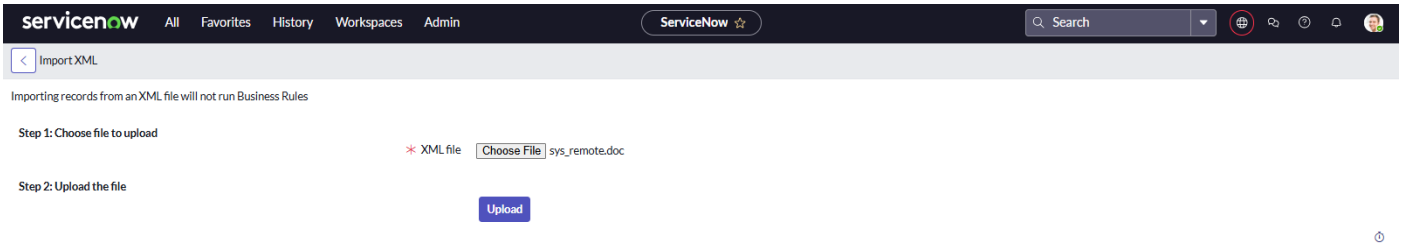
Retrieving the update set into another instance allows teams to **test functionalities**, **validate workflows**, and ensure that all configurations perform as expected before final deployment.
It also helps maintain a structured **change management process**, where every modification is tracked and approved through controlled migration.
By following this approach, organizations can avoid data loss, configuration mismatches, and manual errors during deployment.

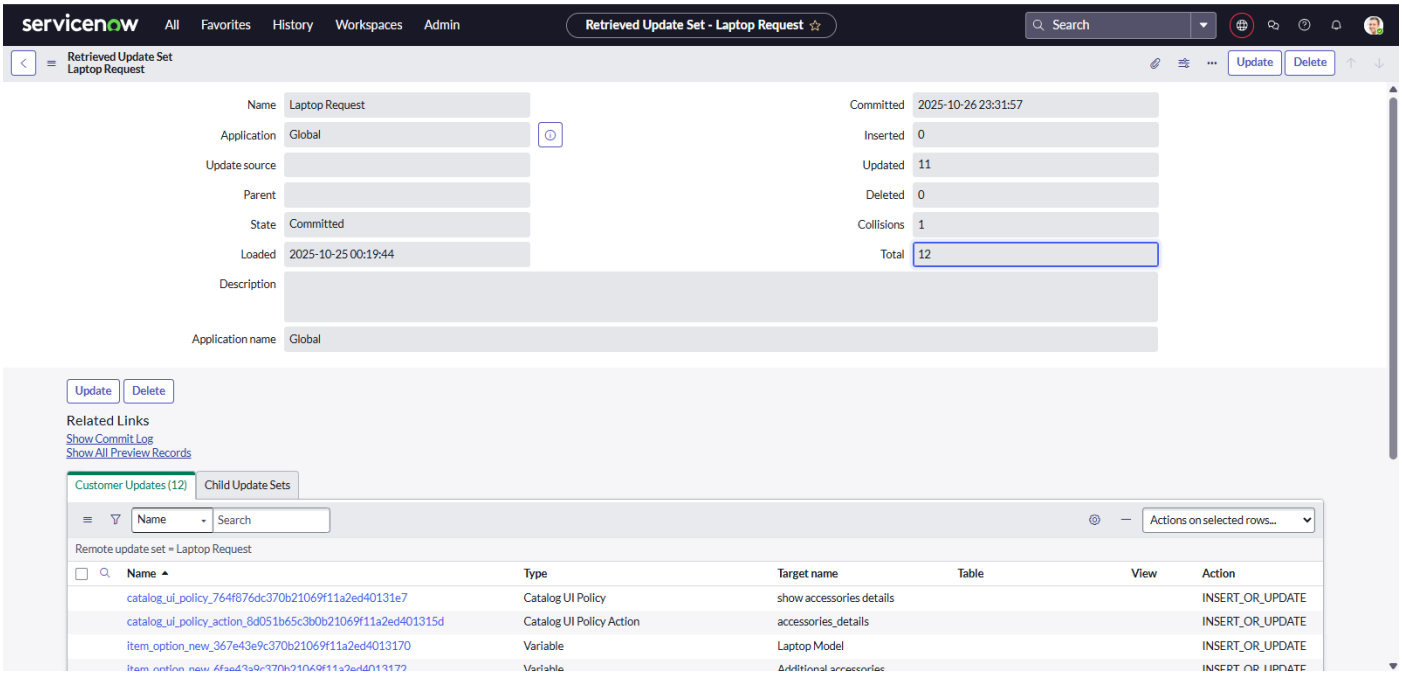
Furthermore, once the update set is retrieved, administrators can **preview**, **validate**, and **commit** the changes, making sure no dependencies are missing and the system remains stable.
This systematic import and retrieval process provides a reliable method to promote changes between environments while ensuring a secure, standardized deployment workflow.

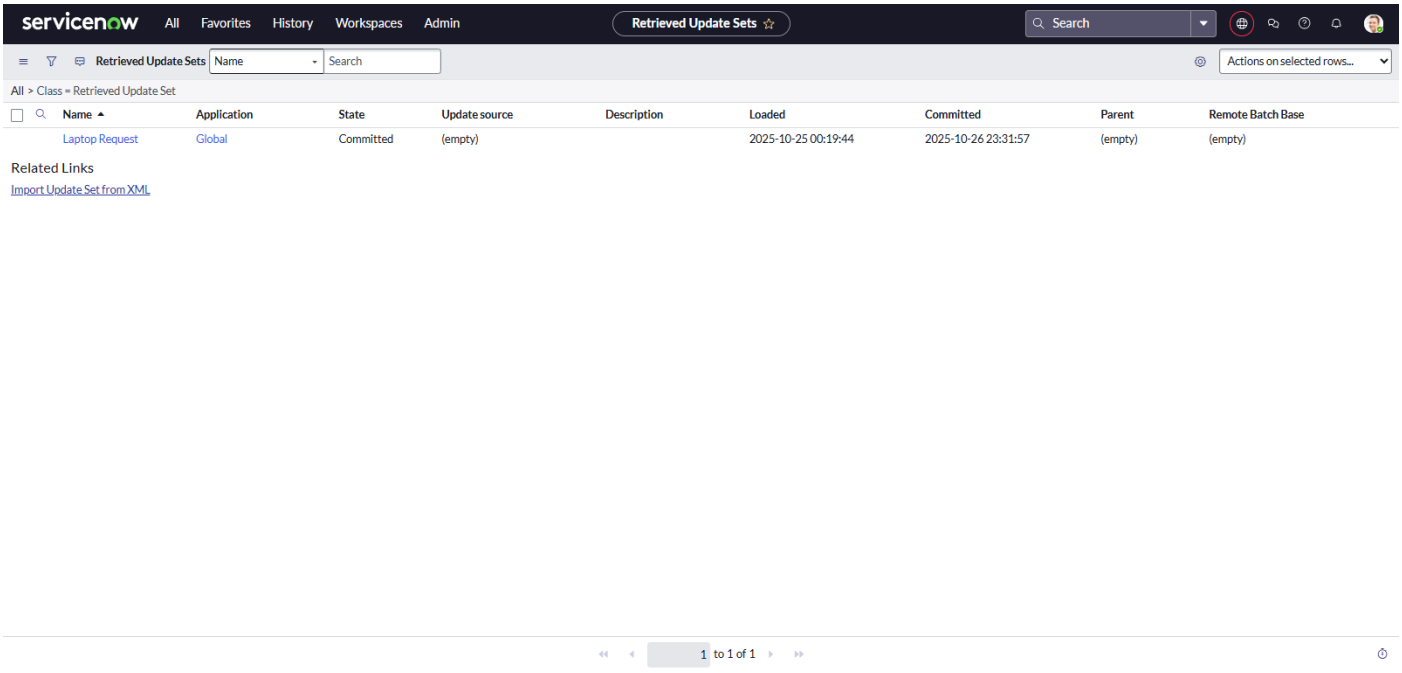
Detailed Steps to Retrieve the Update Set

1. **Login to the Target Instance**
    - Open your target ServiceNow instance (for example, *testing* or *production*).
    - Log in using **admin credentials** and verify your access permissions.
2. **Navigate to Retrieved Update Sets**
    - Go to the Application Navigator → **System Update Sets → Retrieved Update Sets**.
    - This section stores all update sets imported from other instances.
3. **Import the XML File**
    - Click **Import Update Set from XML** at the top of the list view.
    - Browse your computer for the exported file (**Laptop Request.xml**) and click **Upload**.
    - Once uploaded, it appears as a new entry in the list.
4. **Open and Preview the Update Set**
    - Click the imported record **Laptop Request**.
    - Click **Preview Update Set** to check for any errors or conflicts.
    - Review the preview results carefully.
5. **Resolve Any Issues**
    - If any missing dependencies or conflicts appear, fix them before proceeding.
    - Repeat the preview until no errors are found.
6. **Commit the Update Set**
    - Click **Commit Update Set** to apply all captured configurations (catalog item, variables, UI policies, etc.) to the new instance.
    - Wait for the process to complete successfully.
7. **Verify Deployment**
    - Navigate to **Service Catalog → Maintain Items**.
    - Open **Laptop Request** and verify that all configurations (fields, dynamic behaviors, reset actions) are working properly.


servicenow
All
Favorites
History
Workspaces
Admin
Retrieved Update Sets
Search
update sets
FAVORITES
No Results
ALL RESULTS
System Update Sets
Update Sources
Retrieved Update Sets
Update log
Local Update Sets
Merge Update Sets
Merge Completed Sets
Update Sets to Commit
Update Set Commit History
Search
Actions on selected rows...
State
Update source
Description
Loaded
Committed
Parent
Remote Batch Base
Committed
(empty)
2025-10-25 00:19:44
2025-10-26 23:31:57
(empty)
(empty)


servicenow
All
Favorites
History
Workspaces
Admin
ServiceNow
Search
Import XML
Importing records from an XML file will not run Business Rules
Step 1: Choose file to upload
XML file
Choose File
sys_remote.doc
Step 2: Upload the file
Upload





7.8 **Testing:**

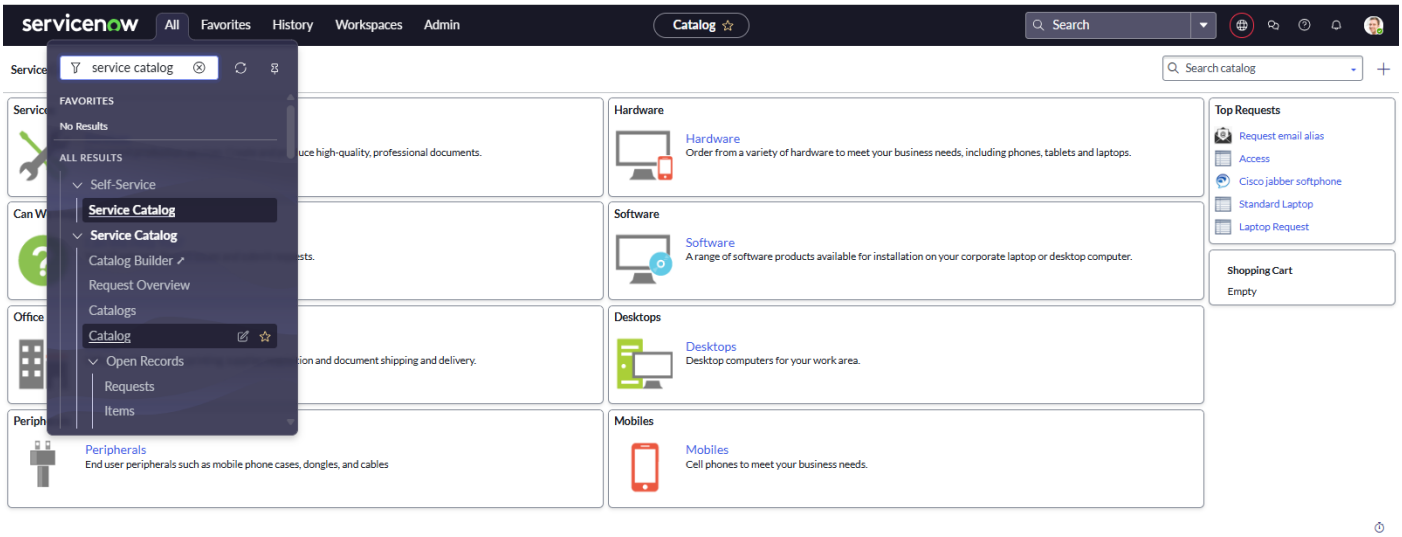
Test Catalog Item

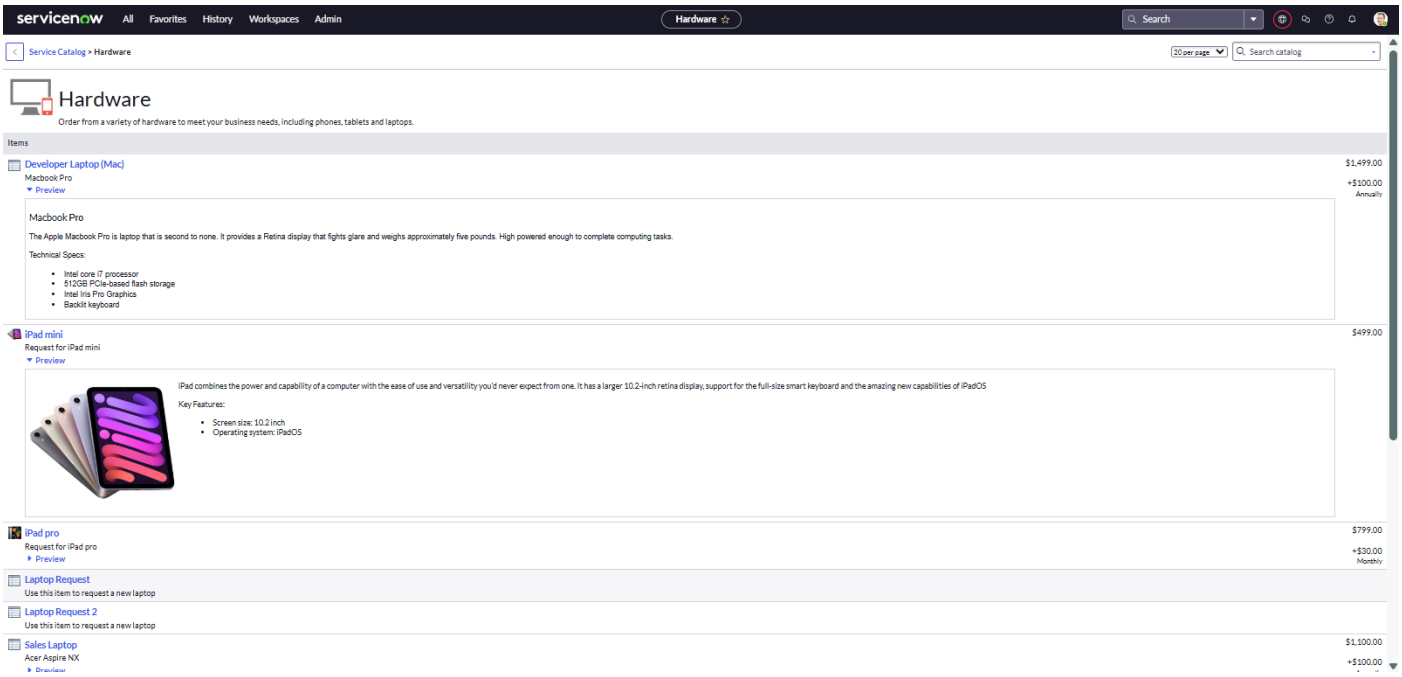
After creating and deploying the **Laptop Request** catalog item, it is essential to perform testing to ensure that the form, workflows, and UI behaviors work correctly.
Testing helps verify that users can submit laptop requests smoothly, all variables capture accurate data, dynamic form behaviors function as intended, and the overall process meets business requirements without errors.
Proper testing also ensures that the catalog item behaves consistently across different users, roles, and environments.

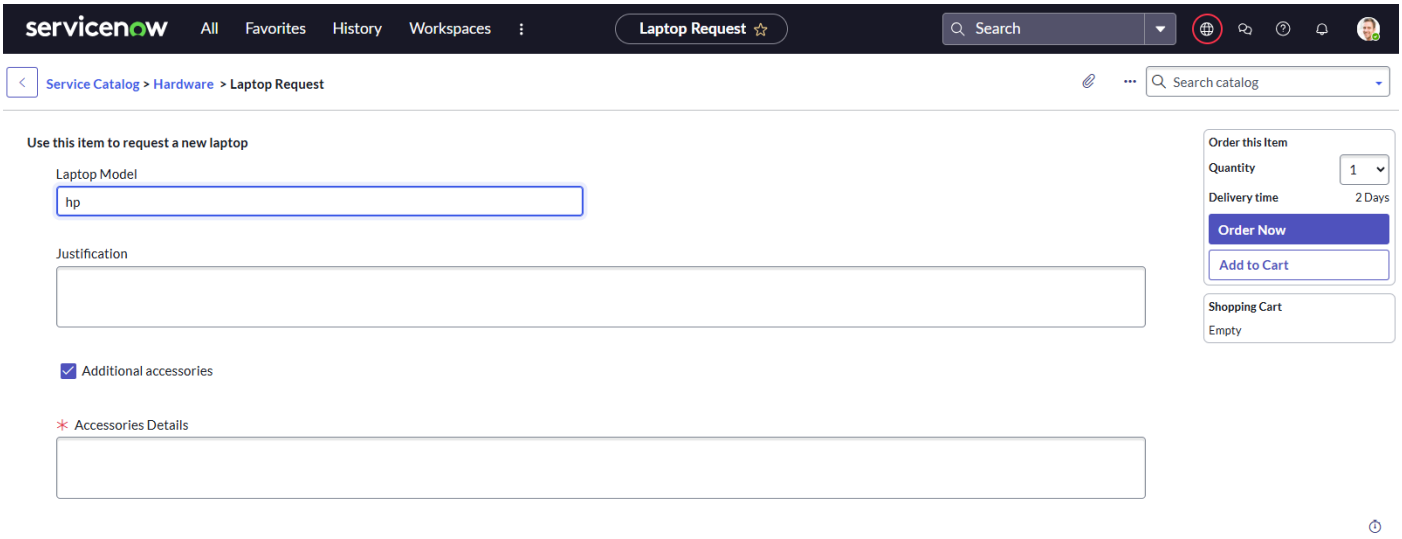
Steps to Test the Laptop Request Catalog Item

1. **Navigate to the Service Catalog**
    - Go to **Service Catalog → Catalog Definitions → Maintain Items**.
    - Open the **Laptop Request** catalog item that you created.
2. **Open the Item in Service Portal**
    - Click the **Try It** button or open the Service Portal link to view the *Laptop Request* form as an end user.
    - This allows testing of the real user experience.
3. **Test Form Fields (Variables)**
    - Check all variables — such as **single-line text**, **multi-line text**, and **check box** — to ensure they appear correctly.
    - Enter sample data to verify that all fields accept input properly.
    - Ensure that mandatory fields are enforced and optional fields behave as expected.
4. **Verify Dynamic Form Behavior**
    - Test the **UI Policies** applied to the form.
    - For example, confirm that certain fields show or hide based on user selections, and that read-only fields behave correctly.
    - Validate that the dynamic behavior makes the form user-friendly and intuitive.
5. **Check the Reset Button (UI Action)**
    - Click the **Reset Form** button created through the UI Action.
    - Verify that all entered data is cleared instantly and the form resets to its default state.
    - Ensure there are no errors in the browser console after clicking the button.
6. **Submit the Request**
    - Fill in the required details again and click **Submit**.
    - Confirm that the request is successfully created and routed to the appropriate workflow.
7. **Validate Workflow Execution**
    - Navigate to the **Requested Items (RITM)** record generated after submission.
    - Verify that the workflow runs correctly and moves through all expected stages.
    - Check if any notifications, approvals, or catalog tasks are triggered properly.
8. **Review Request Details**
    - Ensure that all data entered on the form is correctly captured in the request record.
    - Validate that timestamps, requestor details, and variable values appear as expected.
9. **Test with Different Users or Roles**
    - Log in with another user account to confirm access permissions.
    - Verify that only authorized users can submit requests.
10. **Document Test Results**

- Record all findings, screenshots, and any issues found during testing.
- Make corrections if needed, and retest to ensure all errors are resolved.


servicenow
All
Favorites
History
Workspaces
Admin
Catalog
Search
service catalog
FAVORITES
No Results
ALL RESULTS
Self-Service
Service Catalog
Service Catalog
Catalog Builder
Request Overview
Catalogs
Catalog
Open Records
Requests
Items
Search catalog
Hardware
Hardware
Order from a variety of hardware to meet your business needs, including phones, tablets and laptops.
Software
Software
A range of software products available for installation on your corporate laptop or desktop computer.
Desktops
Desktops
Desktop computers for your work area.
Mobiles
Mobiles
Cell phones to meet your business needs.
Peripherals
End user peripherals such as mobile phone cases, dongles, and cables
Top Requests
Request email alias
Access
Cisco jabber softphone
Standard Laptop
Laptop Request
Shopping Cart
Empty


servicenow
All
Favorites
History
Workspaces
Admin
Hardware
Search
Service Catalog > Hardware
20 per page
Search catalog
Hardware
Order from a variety of hardware to meet your business needs, including phones, tablets and laptops.
Items
Developer Laptop (Mac)
Macbook Pro
Preview
$1,499.00
+$100.00
Annually
Macbook Pro
The Apple Macbook Pro is laptop that is second to none. It provides a Retina display that fights glare and weighs approximately five pounds. High powered enough to complete computing tasks.
Technical Specs:
Intel core i7 processor
512GB PCIe-based flash storage
Intel Iris Pro Graphics
Backlit keyboard
iPad mini
Request for iPad mini
Preview
$499.00
iPad combines the power and capability of a computer with the ease of use and versatility you'd never expect from one. It has a larger 10.2-inch retina display, support for the full-size smart keyboard and the amazing new capabilities of iPadOS
Key Features:
Screen size: 10.2 inch
Operating system: iPadOS
iPad pro
Request for iPad pro
Preview
$799.00
+$30.00
Monthly
Laptop Request
Use this item to request a new laptop
Laptop Request 2
Use this item to request a new laptop
Sales Laptop
Acer Aspire NX
$1,100.00
+$100.00


servicenow
All
Favorites
History
Workspaces
Laptop Request
Search
Service Catalog > Hardware > Laptop Request
Search catalog
Use this item to request a new laptop
Laptop Model
hp
Justification
Additional accessories
Accessories Details
Order this Item
Quantity
1
Delivery time
2 Days
Order Now
Add to Cart
Shopping Cart
Empty

8.Conclusion:

The implementation of the **Laptop Request Service Catalog** in ServiceNow has transformed a previously manual, time-consuming, and error-prone process into a **fully automated, efficient, and transparent workflow**. By leveraging ServiceNow's robust platform capabilities — including **Service Catalog, Flow Designer, UI Policies, and Notifications** — the organization has streamlined IT service delivery and enhanced the end-user experience.