

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.

output:-

The screenshot displays the 'Laptop Request' catalog item form in ServiceNow. The top navigation bar includes 'servicenow', 'All', 'Favorites', 'History', 'Workspaces', 'Admin', and a 'Laptop Request' button. A search bar is also present. The breadcrumb trail shows 'Service Catalog > Hardware > Laptop Request'. The form itself is titled 'Use this item to request a new laptop' and contains several input fields: 'Laptop Model', 'Justification', and 'Accessories Details' (which is expanded to show 'Additional Accessories'). On the right side, there is a summary box showing 'Order this Item' with a quantity of 1, a delivery time of 2 days, and buttons for 'Order Now' and 'Add to Cart'. Below this, a 'Shopping Cart' section shows 'Empty'.

1. Project Overview

The *Laptop Request* catalog item in ServiceNow is designed to streamline the process of requesting laptops for employees within an organization. Instead of relying on manual communication or emails, users can submit a structured request directly through the Service Catalog.

The catalog item provides employees with a simple and user-friendly interface where they can specify their laptop requirements such as model type, operating system, accessories, or additional software. Once submitted, the request is automatically routed through predefined workflows, including approvals and task assignments for the IT fulfillment team.

The purpose of this catalog item is to:

- **Improve efficiency** by reducing manual effort in handling laptop requests.
- **Enhance user experience** with a guided and transparent request submission process.
- **Ensure compliance** with organizational policies by enforcing approval flows.

- **Provide tracking and visibility** to both requesters and approvers throughout the request lifecycle.

Ultimately, this project supports automation and standardization of IT asset requests, ensuring that employees receive the right devices quickly while IT maintains control and governance

2. Business Requirement / Objective

Modern organizations rely heavily on laptops as essential tools for daily operations. Traditionally, employees requested laptops through emails, phone calls, or in-person communication with the IT team, which often led to delays, lack of tracking, and inconsistent approvals. This manual process also made it difficult for IT to manage inventory, ensure compliance, and maintain service quality.

The *Laptop Request* catalog item in ServiceNow addresses these challenges by providing a **centralized, automated, and standardized process** for requesting laptops.

Objectives of the project:

- **Streamline request handling** by enabling employees to raise laptop requests directly through the Service Catalog.
- **Standardize information collection** using predefined variables (such as laptop type, configuration, and accessories).
- **Automate approval workflows** so that requests are reviewed and authorized by managers before fulfillment.
- **Enhance visibility and transparency** by allowing both requesters and approvers to track the status of requests in real time.
- **Improve IT efficiency** by generating fulfillment tasks automatically for the IT team.
- **Ensure compliance** with organizational policies regarding asset allocation and cost management.

This project ensures that the organization delivers laptops to employees in a timely, consistent, and controlled manner while reducing manual effort and improving user satisfaction.

3. Scope

The scope of this project defines what is covered under the *Laptop Request* catalog item and what falls outside its boundaries.

In-Scope

- Creation of a **Service Catalog Item** for Laptop Requests in ServiceNow.
- Configuration of **variables** to capture requester details and laptop requirements (e.g., laptop model, operating system, accessories, justification).
- Use of **Flow Designer / Workflow** to automate the approval and fulfillment process.
- Implementation of **manager approval** before request fulfillment.
- Automatic generation of **fulfillment tasks** for the IT support team to process the request.
- **Notifications** for requester, approver, and fulfillment team (e.g., submission confirmation, approval/rejection, completion).
- **Access control** to ensure only eligible users can submit requests.
- **Request tracking** via ServiceNow portal for end users.

Out-of-Scope

- Procurement of new laptops from external vendors (only internal stock is considered).
- Integration with third-party asset management or procurement systems.
- Automation of hardware delivery logistics outside the ServiceNow platform.
- Management of non-laptop assets (desktops, mobile devices, peripherals).
- Handling of software license requests (separate catalog items handle software).

4. Design / Implementation Details

4.1 updateset

Step 1: Create an Update Set

Before starting the configuration of the *Laptop Request* catalog item, an **Update Set** is created to capture all the customizations. This ensures that the work can be moved across environments (e.g., from development to test or production).

Steps to create an Update Set:

1. Log in to **ServiceNow** with the appropriate role (admin/developer).
2. In the left navigation pane, click on **All** and search for **Update Sets**.
3. Under **System Update Sets**, select **Local Update Sets**.
4. Click on **New** to create a new update set.
5. Enter the following details:
 - **Name:** Laptop Request
 - (Other fields such as Description can be filled as needed).
6. Click **Submit**.
7. Once created, open the update set record and click **Make Current**.

By making the update set current, all further configurations (catalog item, variables, workflows, etc.) will automatically be captured in this update set.

The screenshot shows the 'Update Set' configuration page for 'Laptop Request Project'. The form includes fields for Name, State (set to 'In progress'), Parent, Release date, Install date (2025-09-16 22:16:42), Installed from, and Description. On the right, there are fields for Application (Global), Created (2025-09-16 22:16:41), Created by (admin), and Merged to. Below the form are 'Update' and 'Back Out' buttons. A 'Related Links' section contains links for 'Export to XML', 'Merge With Another Update Set', 'Scan Update Set', and 'Show Update's History'. At the bottom, there are tabs for 'Customer Updates (10)', 'Update Set Logs (12)', 'Child Update Sets', and 'Install History'. A table header is visible with columns: Created, Type, View, Target name, Updated by, Remote update set, and Action.

4.2 Service Catalog Item

Step 1: Create Service Catalog Item

After setting up the update set, the next step is to create a new **Catalog Item** for the laptop request. This item will be available in the Service Catalog for end users to submit their requests.

Steps to create the Catalog Item:

1. Log in to **ServiceNow**.
2. In the left navigation pane, click on **All** and search for **Service Catalog**.
3. Under **Catalog Definitions**, select **Maintain Items**.
4. Click on **New** to create a new catalog item.
5. Fill in the following details:
 - **Name:** Laptop Request
 - **Catalog:** Service Catalog
 - **Category:** Hardware
 - **Short Description:** Use this item to request a new laptop
6. Click on **Save** to create the item.

servicenow All Favorites History Workspaces Admin Catalog Item - Laptop Request

Build and modify items faster with the improved Catalog Builder.

Catalog items are goods or services available to order from the service catalog. Items can be anything from hardware, like tablets and phones, to software applications, to furniture and office supplies.

- Enter a Name and Short description to display for the item.
- Enter a Price, approvals, variables, and other information as needed.

Name: Application:

Catalogs: Active: ☒

Category: Fulfillment automation level:

State:

Checked out:

Owner:

Item Details Process Engine Picture Pricing Portal Settings

Short description:

Description:

Step 3: Add Variables to Catalog Item

Once the catalog item (*Laptop Request*) is created, the next step is to define **variables**. Variables capture user input when submitting the request and help the IT team understand the requirements clearly.

Steps to Add Variables:

Step 2: Create Variables

1. Open the newly created **Laptop Request** catalog item.
2. Scroll down to the **Variables** related list.
3. Click **New** to create a new variable.
4. Enter the following details for each variable:
5. **Variable 1: Laptop Model**
 - Type: *Single Line Text*
 - Name: `laptop_model`
 - Order: 100
 - Click **Submit**.
6. **Variable 2: Justification**
 - Type: *Multi Line Text*
 - Name: `justification`
 - Order: 200
 - Click **Submit**.

7. Variable 3: Additional Accessories

- Type: *Checkbox*
- Name: additional_accessories
- Order: 300
- Click **Submit**.

8. Variable 4: Accessories Details

- Type: *Multi Line Text*
- Name: accessories_details
- Order: 400
- Click **Submit**.

Step 3: Save the Catalog Item

- After all variables have been added, they are automatically linked to the *Laptop Request* catalog item.
- Finally, click **Save** on the catalog item form to confirm the changes.

These variables ensure that the requester provides all the necessary details (laptop specifications, justification, and accessories requirements) while submitting a request.

Variable Laptop Model

Application: Global

Map to field: ☐

Type: Single Line Text

Catalog item: Laptop Request

Order: 100

Active: ☒

Mandatory: ☐

Read only: ☐

Hidden: ☐

Unique: ☐

Disable automatic slot fill based on user context: ☐

Question Annotation Type Specifications Default Value Auto-populate Permission Availability

* Question: Laptop Model

* Name: laptop_model

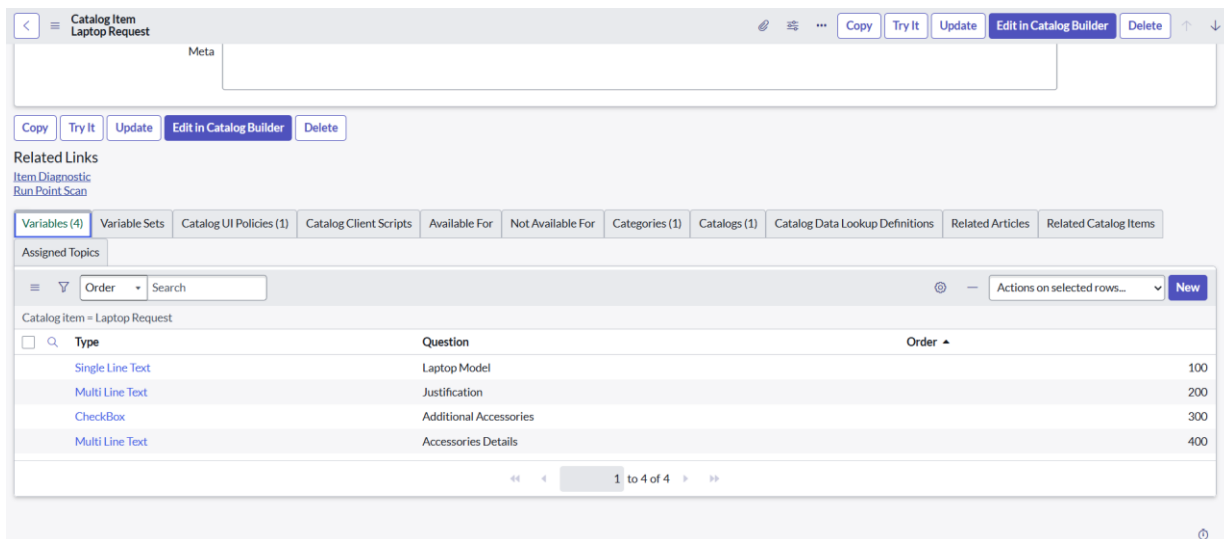
Conversational label:

Tooltip:

Example Text:

Copy Update Delete

Related links



3) UI Policy

Step 1: Create Catalog UI Policies

Catalog UI Policies are used to dynamically control the behavior of variables (e.g., making them visible, mandatory, or read-only) based on conditions. For the *Laptop Request* catalog item, a UI policy is created to ensure that when a user selects **Additional Accessories**, they must provide details in the **Accessories Details** field.

Steps to Create a Catalog UI Policy:

1. In the left navigation pane, click **All** and search for **Service Catalog**.
2. Under **Catalog Definitions**, select **Maintain Items**.
3. Search for the previously created item: **Laptop Request**.
4. Open the *Laptop Request* catalog item record.
5. Scroll down to the **Catalog UI Policies** related list and click **New**.
6. Fill in the details:
 - **Short Description:** Show Accessories Details
 - **Catalog Conditions** (When to Apply):
 - Field: additional_accessories
 - Operator: is
 - Value: true
7. Click **Save** (do not click Submit yet).

Step 2: Configure Catalog UI Policy Actions

1. Scroll down to the **Catalog UI Policy Actions** related list.
2. Click **New** to add an action.
3. Fill in the details:
 - **Variable Name:** accessories_details
 - **Order:** 100
 - **Mandatory:** True
 - **Visible:** True
4. Click **Save**.
5. Return to the Catalog UI Policy form and click **Save** again to confirm the policy.

Result:

When the requester selects the **Additional Accessories** checkbox, the **Accessories Details** field will automatically become visible and mandatory. If unchecked, the field will remain hidden.

The screenshot shows the ServiceNow interface for configuring a Catalog UI Policy. The page title is 'Catalog UI Policy - show accessories details'. The breadcrumb trail is 'Catalog UI Policy > show accessories details'. The page includes a search bar and navigation links for 'Update' and 'Delete'.

The main configuration area is divided into two tabs: 'When to Apply' and 'Script'. The 'When to Apply' tab is active.

When to Apply Configuration:

- Applies to:** A Catalog Item (dropdown)
- Application:** Global (dropdown)
- * Catalog item:** Laptop Request (text input)
- Active:** ☒
- * Short description:** show accessories details (text input)

Conditions:

- Catalog Conditions:** additional_accessories is true
- AND OR X:** (buttons for logical operators)

Applies on a Catalog Item view: ☒

Applies on Catalog Tasks: ☐

Applies on Requested Items: ☐

On load: ☒

Reverse the effects of the catalog UI policy actions when the Conditions evaluate to false: ☐

The screenshot shows the ServiceNow interface for configuring a Catalog UI Policy Action. The breadcrumb trail is 'Catalog UI Policy Action - accessories_details'. A blue banner at the top states: 'UI policy actions specify exactly what actions to take on a specified field. The conditions specified in the UI policy determine when these actions are triggered. [More Info](#)'. Below this, the configuration form is divided into two columns. The left column contains: 'Catalog Item' (Laptop Request), 'Variable name' (accessories_details), and 'Order' (100). The right column contains: 'Application' (Global), 'Mandatory' (True), 'Visible' (True), 'Read only' (Leave alone), 'Value action' (Leave alone), and 'Field message type' (None). At the bottom left of the form are 'Update' and 'Delete' buttons. At the bottom right is a small icon.

4) UI Action

Step 1: Create UI Action

UI Actions in ServiceNow are used to add custom buttons, links, or context menu items to forms and lists. For the *Laptop Request* catalog item, a UI Action is created to allow users to **reset the form**, clearing all entered values.

Steps to Create a UI Action:

1. In the left navigation pane, click **All** and search for **UI Actions**.
2. Under **System Definition**, select **UI Actions**.
3. Click on **New** to create a new UI Action.
4. Fill in the following details:
 - **Table:** Shopping Cart (sc_cart)
 - **Order:** 100
 - **Action Name:** Reset form
 - **Client:** Checked
5. In the **Script** field, enter the following client-side script:
6.


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

Result:

A **Reset Form** button will now be available on the Shopping Cart form. When clicked, it clears all

entered values and alerts the user that the form has been reset.

The screenshot shows the ServiceNow UI Action configuration interface. The top navigation bar includes 'servicenow', 'All', 'Favorites', 'History', 'Workspaces', 'Admin', and a search bar. The main header indicates the current action is 'UI Action - Reset form'. The configuration form is divided into two columns. The left column contains fields for 'Name' (Reset form), 'Table' (Shopping Cart [sc_cart]), 'Order' (100), 'Action name', and several checkboxes for 'Active', 'Show insert', 'Show update', 'Client', 'List v2 Compatible', and 'List v3 Compatible'. Below these are fields for 'Overrides', 'Messages', 'Comments', and 'Hint'. The right column contains a dropdown for 'Application' (Global) and a series of checkboxes for 'Form button', 'Form context menu', 'Form link', 'List banner button', 'List bottom button', 'List context menu', 'List choice', and 'List link'. At the bottom of this column are dropdowns for 'Form style' and 'List style', both set to '-- None --'. 'Update' and 'Delete' buttons are located at the top right of the configuration area.

5) Export Update set

Step 1: Exporting Changes to Another Instance

Once all configurations (catalog item, variables, UI policies, and UI actions) are completed in the development instance, the changes must be exported to another instance (e.g., Test or Production). This is done using **Update Sets**.

Steps to Export Update Set:

1. In the left navigation pane, click **All** and search for **Update Sets**.
2. Select **Local Update Sets**.
3. Open the update set created earlier: **Laptop Request Project**.
4. Change the **State** of the update set to **Complete**.
5. Scroll down to the **Updates** related list tab to verify all captured updates (catalog item, variables, UI policies, UI actions, etc.).
6. From the header, click **Export to XML**.
7. The system will generate and download an **XML file** containing all the changes made under this update set.

This XML file can then be imported into another ServiceNow instance to migrate the *Laptop Request* project configurations.

Update Set - Laptop Request Project

* Name:

State:

Parent:

Release date:

Install date:

Installed from:

Description:

Application:

Created:

Created by:

Merged to:

[Update](#) [Back Out](#)

Related Links

[Export to XML](#)

[Merge With Another Update Set](#)

[Scan Update Set](#)

Customer Updates (10) | **Update Set Logs** | **Child Update Sets** | **Install History**

Update set = Laptop Request Project

Created	Type	View	Target name	Updated by	Remote update set	Action

6) Login to another Instance

Step 1: Retrieving the Update Set

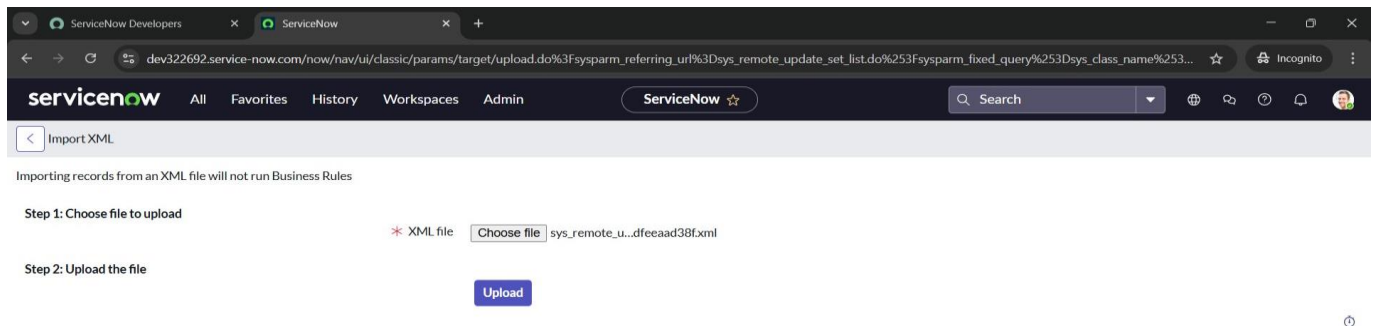
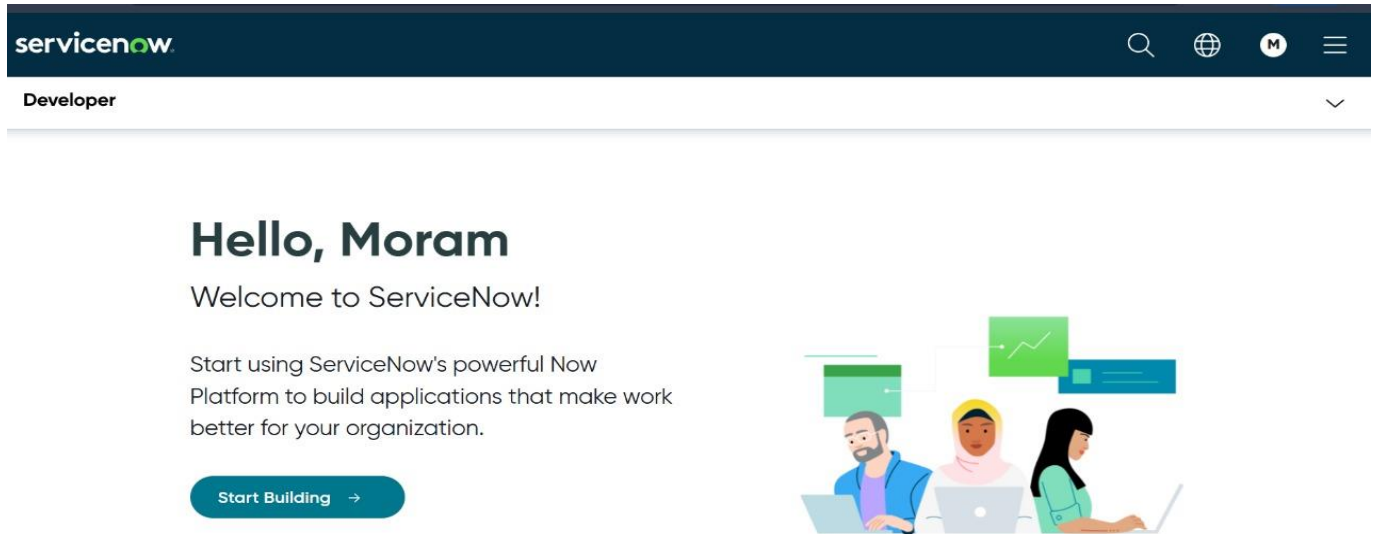
After exporting the update set from the development instance, the next step is to import it into another instance (e.g., Test, UAT, or Production). This ensures that all configurations are replicated in the target instance.

Steps to Retrieve and Commit the Update Set:

1. Open the target ServiceNow instance (recommended: use an incognito/private window).
2. Log in with the appropriate credentials.
3. In the left navigation pane, click **All** and search for **Update Sets**.
4. Under **System Update Sets**, select **Retrieved Update Sets**.
5. On the Retrieved Update Sets list view, scroll down and click **Import Update Set from XML**.
6. Upload the previously downloaded XML file of the update set (*Laptop Request Project*).
7. Click **Upload**. The update set is now available in the list of retrieved update sets.
8. Open the retrieved update set record (**Laptop Request Project**).
9. Click on **Preview Update Set** to validate changes and check for errors or collisions.
10. If the preview is successful, click **Commit Update Set** to apply the changes in this instance.
11. Verify the **Updates** related list tab to confirm all configurations (catalog item, variables, UI policies, UI actions, etc.) have been migrated.

Result:

After committing the update set, the *Laptop Request Project* catalog item and all related configurations are now available in the new instance, ensuring consistency across environments.



Retrieved Update Set
Laptop Request Project

Update

Delete

Name

Laptop Request Project

Application

Global

Update source

Parent

State

Committed

Loaded

2025-09-16 22:02:12

Description

Application name

Global

Committed

2025-09-16 22:16:41

Inserted

0

Updated

10

Deleted

0

Collisions

0

Total

10

Update

Delete

Related Links

Show Commit Log
Show All Preview Records

Customer Updates (10)

Child Update Sets

Name

Search

Actions on selected rows...

Remote update set = Laptop Request Project

Name	Type	Target name	Table	View	Action
catalog_ui_policy_732803598388b210c28ab5dfeead3e4	Catalog UI Policy	show accessories details			INSERT_OR_UPDATE
catalog_ui_policy_action_c5980b9d8388b210c28ab5dfeead358	Catalog UI Policy Action	additional_accessories			INSERT_OR_UPDATE
item_option_new_22c5c7598388b210c28ab5dfeead366	Variable	Justification			INSERT_OR_UPDATE
item_option_new_b426cb998388b210c28ab5dfeead342	Variable	Accessories Details			INSERT_OR_UPDATE
item_option_new_c7e543d98388b210c28ab5dfeead3de	Variable	Additional Accessories			INSERT_OR_UPDATE
item_option_new_cf65c7598388b210c28ab5dfeead360	Variable	Laptop Model			INSERT_OR_UPDATE
sc_cat_item_25d40b958388b210c28ab5dfeead309	Catalog Item	Laptop Request			INSERT_OR_UPDATE
sc_cat_item_catalog_053543998388b210c28ab5dfeead3a2	Catalog Items Catalog	Service Catalog, Laptop Request			INSERT_OR_UPDATE
sc_cat_item_category_0d3543998388b210c28ab5dfeead3a6	Catalog Item Category	Hardware, Laptop Request			INSERT_OR_UPDATE
sys_ui_action_cd2a0b1183c8b210c28ab5dfeead39c	UI Action	Reset form	Shopping Cart [sc_cart]		INSERT_OR_UPDATE

1 to 10 of 10

7) Testing

Step 1: Testing the Catalog Item

After migrating the update set to the target instance, the *Laptop Request* catalog item must be tested to ensure all configurations work as expected.

Steps to Test the Catalog Item:

1. In the target ServiceNow instance, search for **Service Catalog** in the application navigator.

2. Select **Catalog** under *Service Catalog*.
3. Open the **Hardware** category.
4. Locate and open the catalog item **Laptop Request**.
5. Verify that the form displays the defined variables:
 - **Laptop Model**
 - **Justification**
 - **Additional Accessories**
6. Test the **UI Policy functionality**:
 - By default, only three variables are visible.
 - When the **Additional Accessories** checkbox is selected, the field **Accessories Details** should appear.
 - The **Accessories Details** field should also be **mandatory** when visible.

Result:

The test confirms that the catalog item behaves as expected. The *Laptop Request* item captures the necessary information, dynamically displays the **Accessories Details** field when required, and enforces mandatory input.

The screenshot shows the ServiceNow interface for a 'Laptop Request' catalog item. The top navigation bar includes 'servicenow', 'All', 'Favorites', 'History', 'Workspaces', 'Admin', and a 'Laptop Request' button. Below the navigation bar, the breadcrumb trail reads '< Service Catalog > Hardware > Laptop Request'. The main form area is titled 'Use this item to request a new laptop'. It contains three input fields: 'Laptop Model' (a single-line text box), 'Justification' (a multi-line text box), and 'Additional Accessories' (a checkbox). To the right of the form, there is a summary section titled 'Order this Item'. This section displays 'Quantity' as 1, 'Delivery time' as 2 Days, and two buttons: 'Order Now' and 'Add to Cart'. Below the summary section is a 'Shopping Cart' section, which currently shows 'Empty'. A small icon is visible in the bottom right corner of the form area.

The screenshot displays the ServiceNow user interface for a 'Laptop Request' catalog item. The top navigation bar includes the ServiceNow logo, a menu with 'All', 'Favorites', 'History', 'Workspaces', and 'Admin', a 'Laptop Request' button with a star icon, a search bar, and user profile icons. Below the navigation bar, a breadcrumb trail shows 'Service Catalog > Hardware > Laptop Request'. The main form area is titled 'Use this item to request a new laptop' and contains three input fields: 'Laptop Model', 'Justification', and 'Additional Accessories' (which is checked). Below these is a section for 'Accessories Details' marked with a red asterisk. On the right side, a sidebar contains an 'Order this item' section with a 'Quantity' dropdown set to '1' and a 'Delivery time' of '2 Days'. Below this are 'Order Now' and 'Add to Cart' buttons. At the bottom of the sidebar, a 'Shopping Cart' section shows it is 'Empty'. A small help icon is located at the bottom right of the form area.

Conclusion

The *Laptop Request Catalog Item* project successfully streamlines the process of requesting laptops within the organization by leveraging ServiceNow's Service Catalog capabilities. By creating a dynamic and automated catalog item, the project provides employees with an intuitive, user-friendly interface that minimizes errors, reduces manual effort, and improves overall efficiency.

The implementation of variables, UI policies, UI actions, and automated workflows ensures that requests are captured accurately, routed for necessary approvals, and fulfilled in a timely manner. Additionally, the migration of configurations through update sets demonstrates how ServiceNow supports smooth transitions between development, test, and production environments.

This project clearly highlights how ServiceNow can replace manual, error-prone processes with standardized, automated, and user-centric solutions. It not only improves service delivery and IT governance but also enhances employee satisfaction by offering a modern and transparent request experience.

