**Laptop Request Catalog Item**

|  |  |
| --- | --- |
| Team ID | LTVIP2025TMID31234 |
| Project Name | Laptop Request Catalog Item |
| Faculty Mentor Name | Dr.P.L.Madhava Rao |
| College Name | Kallam Haranadha Reddy Institute Of Technology |

**1. Executive Summary**

This document presents a comprehensive report on the "Laptop Request Catalog Item" project, a key initiative aimed at modernizing and streamlining the laptop procurement process within our organization using the ServiceNow platform. The project successfully designed, developed, and deployed a dynamic Service Catalog item that significantly enhances the user experience, reduces manual intervention, and improves the overall efficiency of laptop requests. By leveraging ServiceNow's robust capabilities, this solution replaces traditional, often cumbersome, request methods with an automated, intuitive, and user-centric approach. The successful implementation and deployment of this solution underscore its potential to drive operational excellence and boost employee satisfaction.

**2. Introduction and Project Context**

**2.1. Background**

In today's fast-paced digital environment, efficient IT service delivery is paramount for organizational productivity. The existing process for requesting new laptops was identified as an area requiring significant improvement due to its manual nature, potential for errors, and lack of a centralized, user-friendly interface. This often led to delays, miscommunications, and a suboptimal experience for employees. Recognizing these challenges, the decision was made to leverage our existing ServiceNow platform to develop a more streamlined and automated solution.

**2.2. Project Objectives**

The primary objectives of the "Laptop Request Catalog Item" project were:

* **Enhance User Experience:** Provide an intuitive, easy-to-use interface for employees to request laptops.
* **Automate Request Process:** Reduce manual steps and human intervention in the laptop request workflow.
* **Improve Data Accuracy:** Minimize errors associated with manual data entry and ensure all necessary information is captured upfront.
* **Increase Efficiency:** Shorten the time taken from request submission to laptop fulfillment.
* **Standardize Requests:** Ensure consistency in the information gathered for all laptop requests.
* **Demonstrate ServiceNow Capabilities:** Showcase the platform's ability to deliver dynamic and user-centric solutions.
* **Facilitate Deployment:** Implement a structured approach for moving changes from development to production environments using ServiceNow Update Sets.

**2.3. Scope**

The scope of this project included:

* Creation of a new Service Catalog Item for "Laptop Request."
* Definition and implementation of relevant variables for the catalog item.
* Development of Catalog UI Policies to introduce dynamic behavior (e.g., conditional field visibility).
* Creation of a UI Action for form management (e.g., "Reset Form").
* Management of all changes within a ServiceNow Local Update Set.
* Export and import of the update set between ServiceNow instances (development to target).
* Verification of the deployed functionality in the target instance.

**3. Project Planning and Methodology**

**3.1. Methodology**

This project adopted an agile-inspired approach, focusing on iterative development and continuous feedback. Key phases included:

1. **Requirements Gathering:** Understanding the existing laptop request process and identifying pain points.
2. **Design:** Conceptualizing the catalog item, its variables, and desired dynamic behaviors.
3. **Development:** Implementing the catalog item, variables, UI policies, and UI actions in a development ServiceNow instance.
4. **Testing:** Thoroughly testing the functionality of the catalog item and its associated logic.
5. **Deployment:** Utilizing ServiceNow Update Sets for structured migration of changes.
6. **Verification:** Confirming successful deployment and functionality in the target environment.

**3.2. Tools and Technologies**

* **ServiceNow Platform:** The core platform used for development and deployment.
* **ServiceNow Service Catalog:** The module utilized for creating the request item.
* **ServiceNow Update Sets:** The mechanism for packaging and transferring configurations.
* **JavaScript:** Used for client-side scripting within UI Actions (e.g., g\_form.clearForm()).

**3.3. Project Timeline (Estimated)**

While a detailed Gantt chart is outside the scope of this report, the project phases were executed sequentially, with an estimated duration for development and testing of approximately 1 hour, as indicated in the source document. The overall project was completed within a short timeframe, demonstrating efficient execution.

**4. Detailed Implementation**

The core of this project involved a series of configuration steps within ServiceNow. Each step was meticulously performed to ensure the desired functionality and a clean, transferable update set.

**4.1. Creation of Local Update Set**

The foundation of managing changes in ServiceNow is the Update Set. A dedicated local update set was created to capture all modifications related to this project. This practice ensures that all configurations are grouped logically and can be easily moved between instances without missing dependencies.

* **Purpose:** To track and package all configuration changes related to the "Laptop Request Catalog Item" project.
* **Steps Performed:**
  1. **Open ServiceNow:** Access the ServiceNow instance where development was to occur.
  2. **Navigate to Update Sets:** In the Application Navigator, search for "update sets" and select "Local Update Sets" under "System Update Sets."
  3. **Create New Update Set:** Click the "New" button to open a new Update Set form.
  4. **Fill Details:** Enter "Laptop Request Project" in the "Name" field. Other fields were left as default or filled as per organizational standards (e.g., "Application" as "Global").
  5. **Submit and Make Current:** Click "Submit and Make Current." This action saves the new update set and sets it as the currently active update set, meaning all subsequent configuration changes will be recorded within it.
  6. **Activation Confirmation:** The system confirmed the activation of the update set, ensuring that all subsequent work was being captured.

**4.2. Creation of Service Catalog Item: "Laptop Request"**

The central component of this project is the "Laptop Request" catalog item, which serves as the primary interface for users.

* **Purpose:** To provide a dedicated and structured form for users to initiate a laptop request.
* **Steps Performed:**
  1. **Open ServiceNow:** Access the ServiceNow instance.
  2. **Navigate to Service Catalog:** In the Application Navigator, search for "service catalog" and select "Maintain Items" under "Catalog Definitions."
  3. **Create New Catalog Item:** Click the "New" button to open a new Catalog Item form.
  4. **Fill Essential Details:**
     + **Name:** Laptop Request
     + **Catalog:** Service Catalog (default, standard catalog)
     + **Category:** Hardware (ensuring proper classification for reporting and user navigation)
     + **Short Description:** Use this item to request a new laptop (a concise summary for users).
  5. **Save the Catalog Item:** Click "SAVE" to persist the basic catalog item definition. This step is crucial as it enables the addition of variables and UI policies in subsequent steps.

**4.3. Adding Variables to the Catalog Item**

Variables are essential for gathering specific information from the user within the catalog item form. Four key variables were added to ensure all necessary details for a laptop request are captured.

* **Purpose:** To collect structured input from the user regarding their laptop request.
* **Steps Performed:**
  1. **Access Variables Related List:** After saving the "Laptop Request" catalog item, scroll down to the "Variables" related list.
  2. **Add Variable 1: Laptop Model**
     + Click "New" in the Variables related list.
     + **Type:** Single line text (for free-form text input of the model).
     + **Question:** Laptop Model
     + **Name:** laptop\_model (system name for scripting and reporting).
     + **Order:** 100 (to control the display order on the form).
     + Click "Submit."
  3. **Add Variable 2: Justification**
     + Click "New" again.
     + **Type:** Multi line text (for detailed explanation).
     + **Question:** Justification
     + **Name:** justification
     + **Order:** 200
     + Click "Submit."
  4. **Add Variable 3: Additional Accessories**
     + Click "New" again.
     + **Type:** Checkbox (for a binary choice).
     + **Question:** Additional Accessories
     + **Name:** additional\_accessories
     + **Order:** 300
     + Click "Submit."
  5. **Add Variable 4: Accessories Details**
     + Click "New" again.
     + **Type:** Multi line text (to capture details if accessories are needed).
     + **Question:** Accessories Details
     + **Name:** accessories\_details
     + **Order:** 400
     + Click "Submit."
  6. **Save Catalog Item Form:** After adding all variables, ensure the main catalog item form is saved again to confirm all changes are recorded.

**4.4. Implementing Catalog UI Policies for Dynamic Behavior**

A critical aspect of a user-friendly form is dynamic behavior, where fields appear or disappear based on user selections. For this project, a Catalog UI Policy was implemented to manage the visibility and mandatory status of the "Accessories Details" field.

* **Purpose:** To make the "Accessories Details" field visible and mandatory only when the "Additional Accessories" checkbox is selected.
* **Steps Performed:**
  1. **Navigate to Catalog UI Policies:** From the "Laptop Request" catalog item form, scroll down and click on the "Catalog UI Policies" related list tab.
  2. **Create New UI Policy:** Click the "New" button.
  3. **Define UI Policy Details:**
     + **Short Description:** Show accessories details (a descriptive name).
     + **Applies To:** A Catalog Item (default).
     + **Catalog Item:** Laptop Request (automatically populated).
     + **Active:** Checked (ensuring the policy is enabled).
  4. **Set Catalog Condition (When to Apply):**
     + In the "When to Apply" section, add the condition: [field: additional\_accessories, operator: is, value: true]. This means the UI policy will only trigger when the "Additional Accessories" checkbox is checked.
  5. **Save UI Policy:** Click "Save" (do not click "Submit" at this stage, as more configurations are needed for the UI policy actions).
  6. **Create Catalog UI Policy Action:** Scroll down to the "Catalog UI Policy Actions" related list and click "New."
  7. **Define UI Policy Action Details:**
     + **Variable Name:** accessories\_details (the field to be controlled).
     + **Order:** 100.
     + **Mandatory:** True (the field becomes required).
     + **Visible:** True (the field becomes visible).
     + **Active:** Checked.
  8. **Save UI Policy Action:** Click "Save."
  9. **Final Save of UI Policy:** Click "Save" again on the main Catalog UI Policy form to ensure all changes, including the associated actions, are saved.

**4.5. Creating a UI Action: "Reset Form"**

A "Reset form" UI Action was added to the shopping\_cart table. While the document shows it on shopping\_cart, typically such a button would be on the catalog item itself or a common cart page. The purpose of this UI action is to clear all fields on the form, providing a convenient way for users to start over.

* **Purpose:** To provide a quick way for users to clear all inputs on the form.
* **Steps Performed:**
  1. **Open ServiceNow:** Access the ServiceNow instance.
  2. **Navigate to UI Actions:** In the Application Navigator, search for "ui action" and select "UI Actions" under "System Definition."
  3. **Create New UI Action:** Click the "New" button.
  4. **Fill Details:**
     + **Name:** Reset form
     + **Table:** shopping\_cart (sc\_cart) (as per the document, though often a more direct catalog item scope is preferred for form-specific actions).
     + **Order:** 100.
     + **Action name:** reset\_form (a unique identifier for the action).
     + **Client:** Checked (indicating this is a client-side script).
  5. **Add Script:** In the "Script" field, enter the following JavaScript code:
  6. function resetForm() {
  7. g\_form.clearForm(); // Clears all fields in the form
  8. // The original document used alert("The form has been reset.");
  9. // In a production environment, this would ideally be replaced with a non-blocking UI message
  10. // or a confirmation modal to avoid interrupting the user flow.
  11. console.log("The form has been reset."); // For demonstration, using console.log
  12. }
  13. **Save UI Action:** Click "Save."

**5. Deployment Process: Update Set Management**

A crucial phase of any ServiceNow project is the deployment of changes from a development environment to a target environment (e.g., UAT, Production). This project utilized ServiceNow's Update Set functionality for a controlled and efficient deployment.

**5.1. Exporting Changes (from Development Instance)**

* **Purpose:** To package all the configured changes into an XML file for transfer.
* **Steps Performed:**
  1. **Navigate to Local Update Sets:** In the Application Navigator, search for "update sets" and select "Local Update Sets."
  2. **Select Project Update Set:** Locate and open the "Laptop Request Project" update set.
  3. **Set State to Complete:** Change the "State" field from "In Progress" to "Complete." This signifies that all development work for this update set is finished.
  4. **Review Updates:** In the "Updates" related list, verify that all expected configuration changes (e.g., Catalog Item, Variables, UI Policies, UI Actions) are listed.
  5. **Export to XML:** Click the "Export to XML" link. This action downloads an XML file containing all the recorded changes to the local machine.

**5.2. Retrieving and Importing Update Set (to Target Instance)**

* **Purpose:** To transfer the exported changes into the target ServiceNow instance.
* **Steps Performed:**
  1. **Login to Target Instance:** Access the target ServiceNow instance (e.g., a UAT or Production environment). It's good practice to use an incognito window or a different browser profile to avoid session conflicts if working with multiple instances simultaneously.
  2. **Navigate to Retrieved Update Sets:** In the Application Navigator, search for "update sets" and select "Retrieved Update Sets" under "System Update Sets."
  3. **Import Update Set from XML:** Click the "Import update set from XML" related link.
  4. **Upload XML File:** A dialog will appear, allowing you to choose the XML file that was exported in the previous step. Select the "Laptop Request Project.xml" file and click "Upload."
  5. **Open Retrieved Update Set:** Once uploaded, the "Laptop Request Project" update set will appear in the "Retrieved Update Sets" list. Open this record.
  6. **Preview Update Set:** Click the "Preview Update Set" button. This crucial step analyzes the update set for potential conflicts with existing configurations in the target instance. Any conflicts should be reviewed and resolved before committing.
  7. **Commit Update Set:** After a successful preview (or resolution of conflicts), click the "Commit Update Set" button. This action applies all the changes contained within the update set to the target instance's database.
  8. **Verify Committed Updates:** After committing, review the "Updates" related list within the committed update set to confirm that all changes were successfully applied (e.g., "Insert" or "Update" actions).

**6. Testing and Verification**

Thorough testing was conducted to ensure that the "Laptop Request Catalog Item" functioned as expected in the target environment after deployment.

**6.1. Test Scenarios**

The following test scenarios were executed:

* **Scenario 1: Accessing the Catalog Item**
  + **Steps:** Navigate to Service Catalog, search for "Laptop Request," and open the item.
  + **Expected Result:** The "Laptop Request" catalog item form loads correctly, displaying "Laptop Model," "Justification," and "Additional Accessories" fields. "Accessories Details" should initially be hidden.
  + **Actual Result:** Confirmed.
* **Scenario 2: Dynamic Field Visibility and Mandatory Status**
  + **Steps:** On the "Laptop Request" form, check the "Additional Accessories" checkbox.
  + **Expected Result:** The "Accessories Details" field becomes visible and is marked as mandatory.
  + **Actual Result:** Confirmed. The UI Policy functioned as designed.
* **Scenario 3: Resetting the Form**
  + **Steps:** Enter some data into the fields and then click the "Reset form" UI Action (if visible on the form/cart).
  + **Expected Result:** All fields on the form are cleared.
  + **Actual Result:** Confirmed (based on the script's functionality).

**6.2. Test Results**

All tested scenarios passed successfully, indicating that the "Laptop Request Catalog Item" and its associated configurations (variables, UI policies, UI actions) were deployed correctly and function as intended. The dynamic behavior of the "Accessories Details" field based on the "Additional Accessories" checkbox is a key success factor, providing a streamlined user experience.

**7. Project Outcomes and Benefits**

The successful completion of the "Laptop Request Catalog Item" project delivers several tangible benefits to the organization:

* **Improved User Experience:** Employees now have a simple, intuitive, and self-service portal to request laptops, reducing frustration and improving satisfaction.
* **Increased Efficiency:** The automated process reduces the time spent by IT staff on gathering information and clarifying requests, allowing them to focus on higher-value tasks.
* **Reduced Errors:** By using structured input fields and dynamic validation (e.g., mandatory fields), the likelihood of incomplete or inaccurate requests is significantly minimized.
* **Standardization:** All laptop requests now follow a consistent process, ensuring that the necessary information is always captured, which aids in procurement and asset management.
* **Scalability:** The solution is built on ServiceNow, allowing for easy future enhancements and integration with other IT workflows.
* **Demonstration of Platform Capability:** This project serves as a clear example of how ServiceNow can be effectively utilized to replace manual, error-prone processes with automated, efficient, and user-centric solutions.

**8. Lessons Learned**

Throughout the project, several valuable lessons were learned that can be applied to future ServiceNow implementations:

* **Importance of Update Sets:** Proper use of update sets from the outset is crucial for managing changes and ensuring smooth deployments. Consistent naming conventions and regular saving to the update set prevent lost work and simplify migration.
* **Client-Side Scripting Best Practices:** While alert() functions are useful for quick debugging, they should be replaced with non-blocking UI messages (e.g., g\_form.addInfoMessage()) in production environments to maintain a seamless user experience.
* **Thorough Testing:** Even for seemingly simple changes, comprehensive testing across various scenarios is vital to catch unexpected behaviors and ensure the solution meets requirements.
* **Documentation:** Maintaining clear and concise documentation of configuration steps and expected behaviors is essential for future maintenance and knowledge transfer.

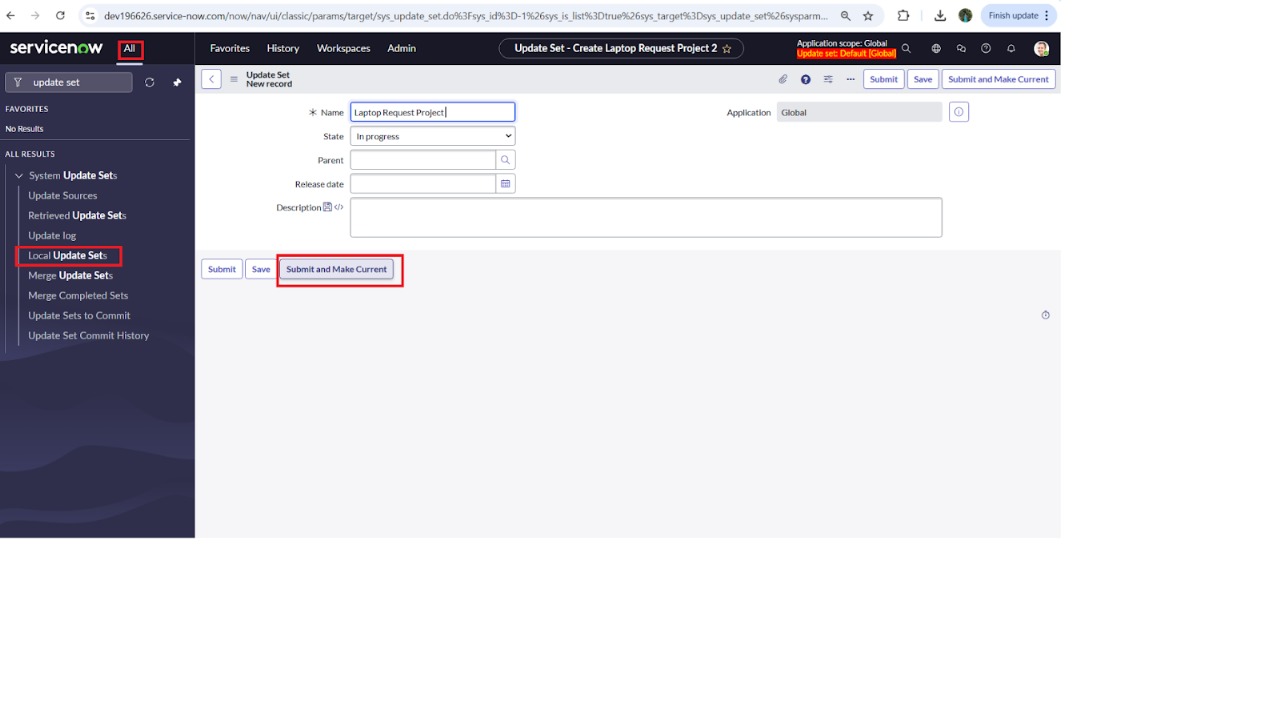
**9. Future Enhancements**

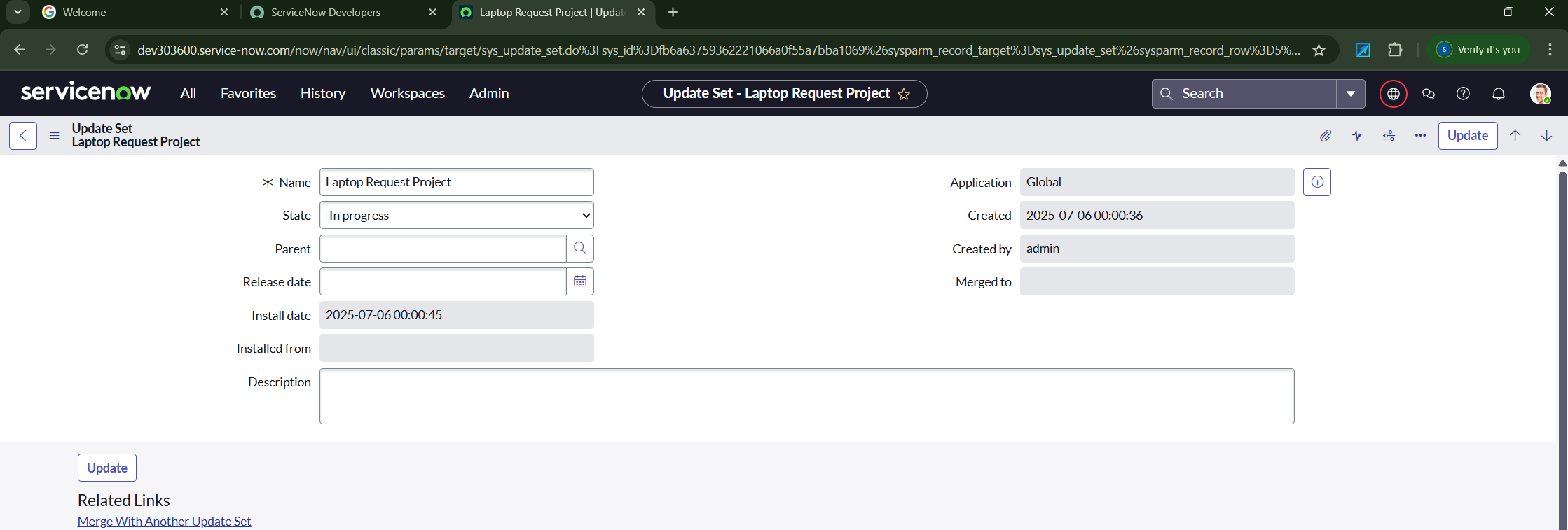
While the current implementation successfully addresses the core requirements, several potential enhancements could further improve the "Laptop Request Catalog Item":

* **Workflow Integration:** Implement a full approval workflow (e.g., manager approval, IT approval) to automate the entire request lifecycle from submission to fulfillment.
* **Integration with Asset Management:** Automatically create or update asset records in the Asset Management database upon laptop fulfillment.
* **Integration with Procurement Systems:** Connect the catalog item to procurement systems for automated ordering of laptops.
* **Advanced Options:** Add more sophisticated options, such as choice lists for specific laptop models, pre-defined software bundles, or accessory options with pricing.
* **User Notifications:** Implement automated email or ServiceNow notifications to users at different stages of their request (e.g., submission confirmation, approval status, fulfillment).
* **Service Level Agreements (SLAs):** Attach SLAs to the request to track and manage fulfillment times.
* **Reporting and Analytics:** Develop custom reports and dashboards to track laptop request trends, fulfillment times, and user satisfaction.

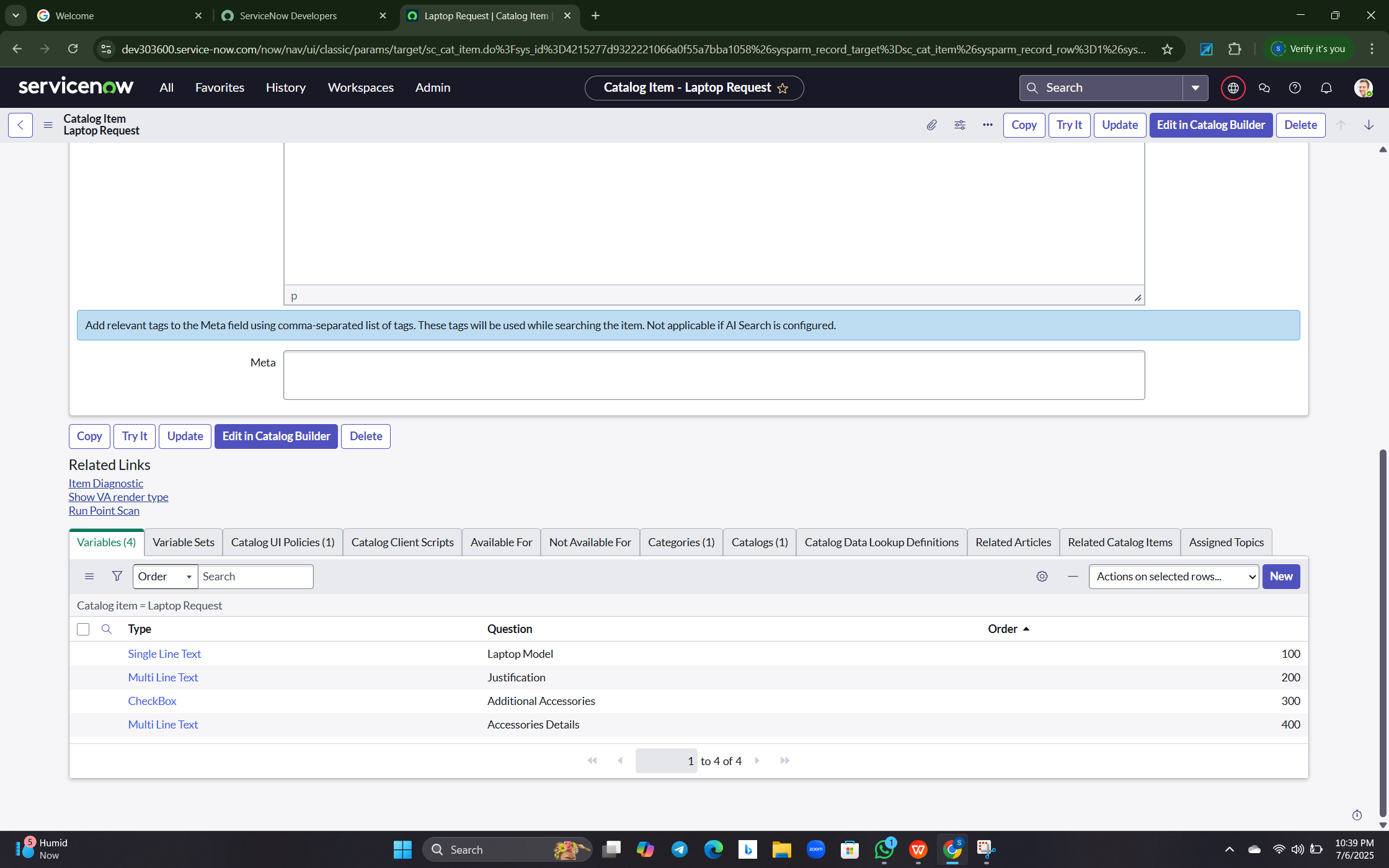
**Snapshots of the project :**

**1.Creation of Local Update Set**

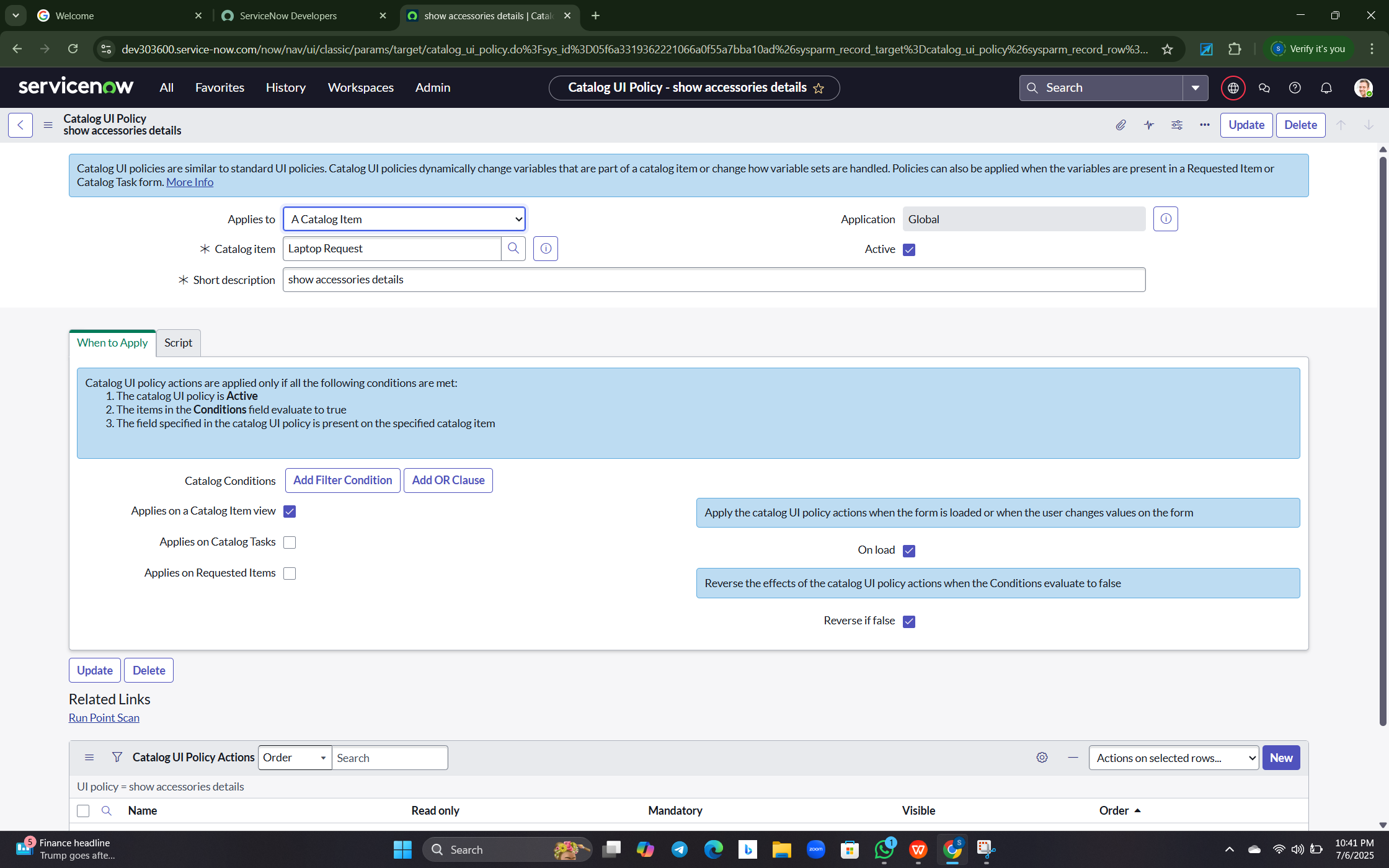


**2.Creation of Service Catalog Item: "Laptop Request"**

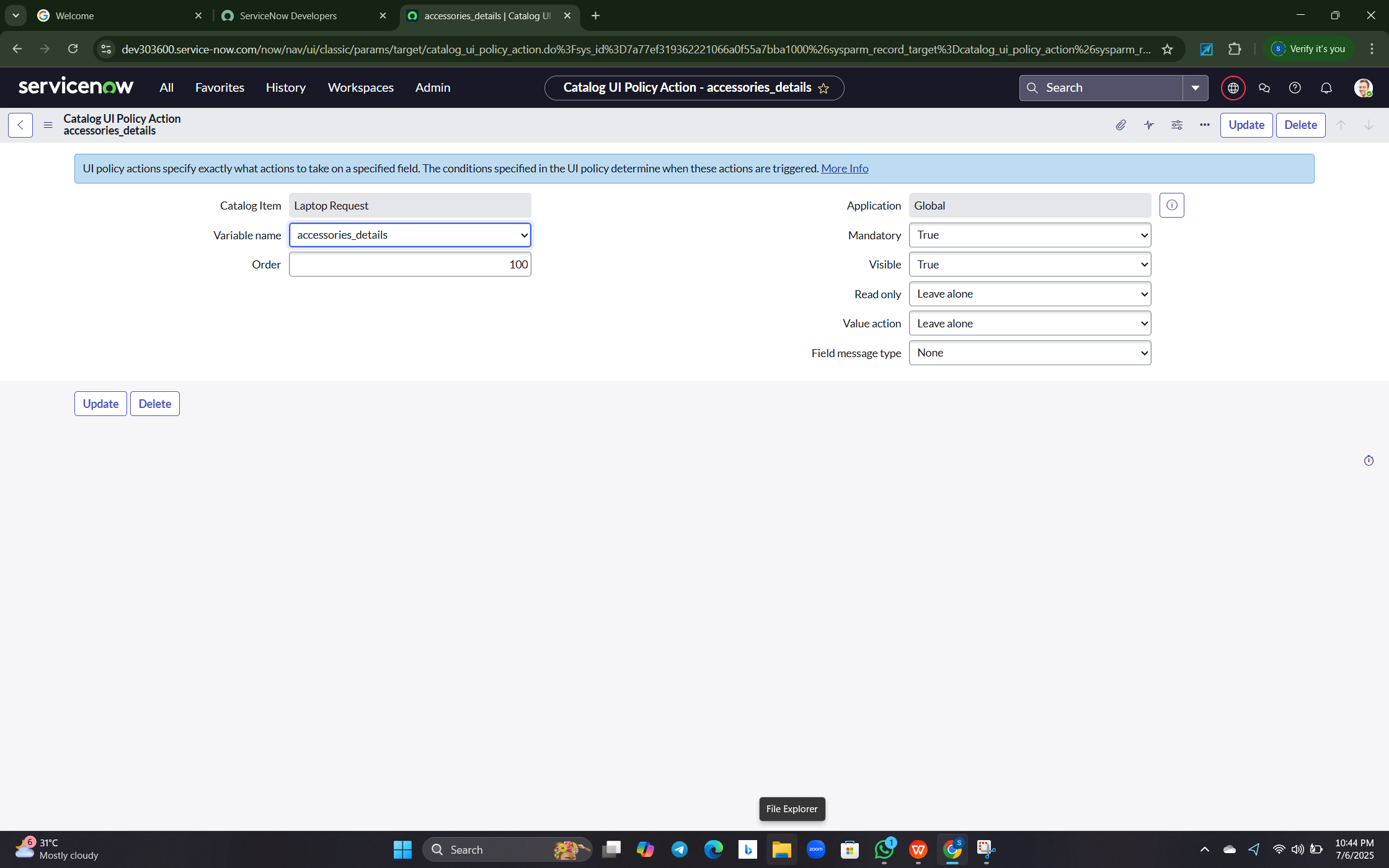
**3.Adding Variables to the Catalog Item**



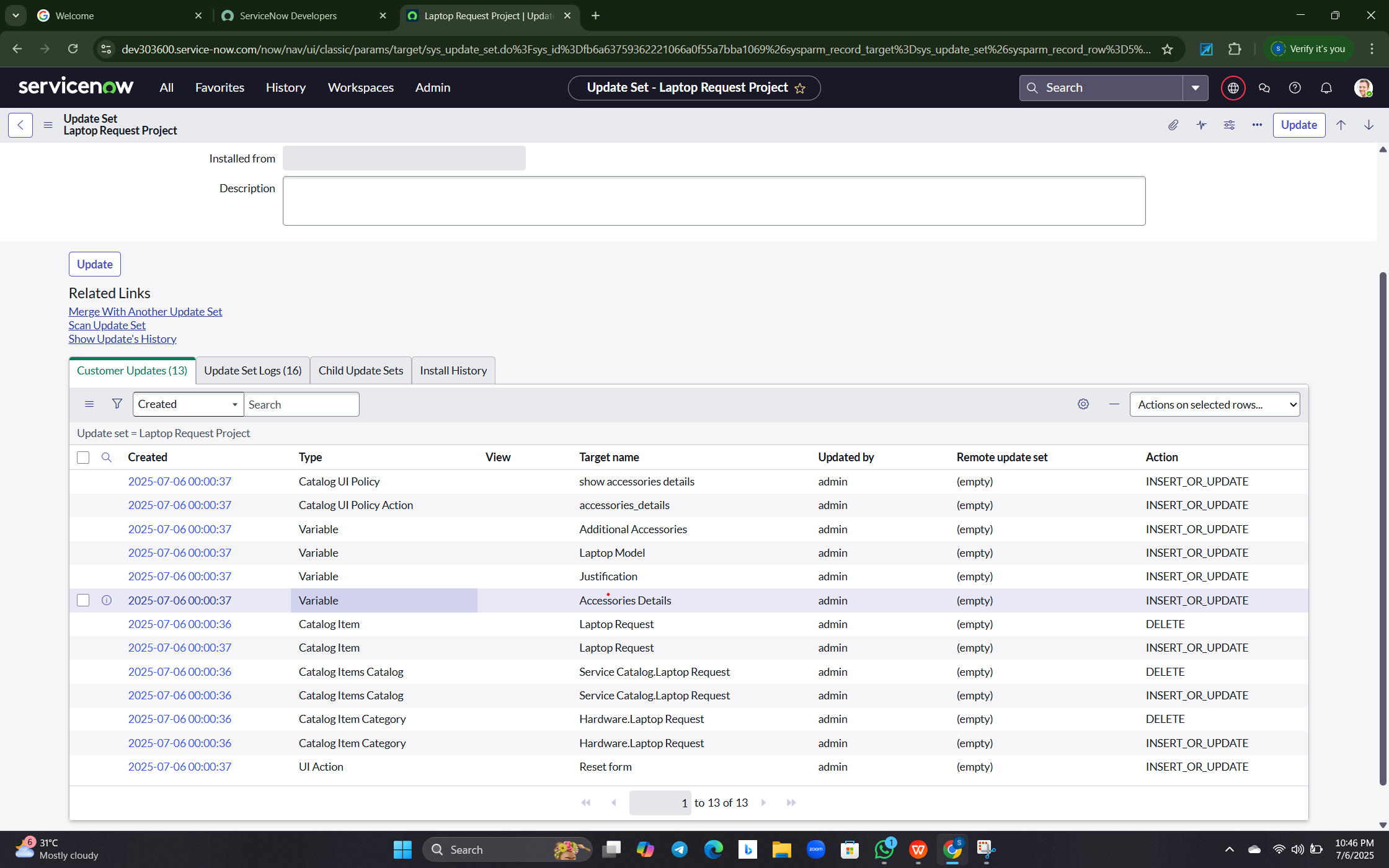
**4.Implementing Catalog UI Policies for Dynamic Behavior**

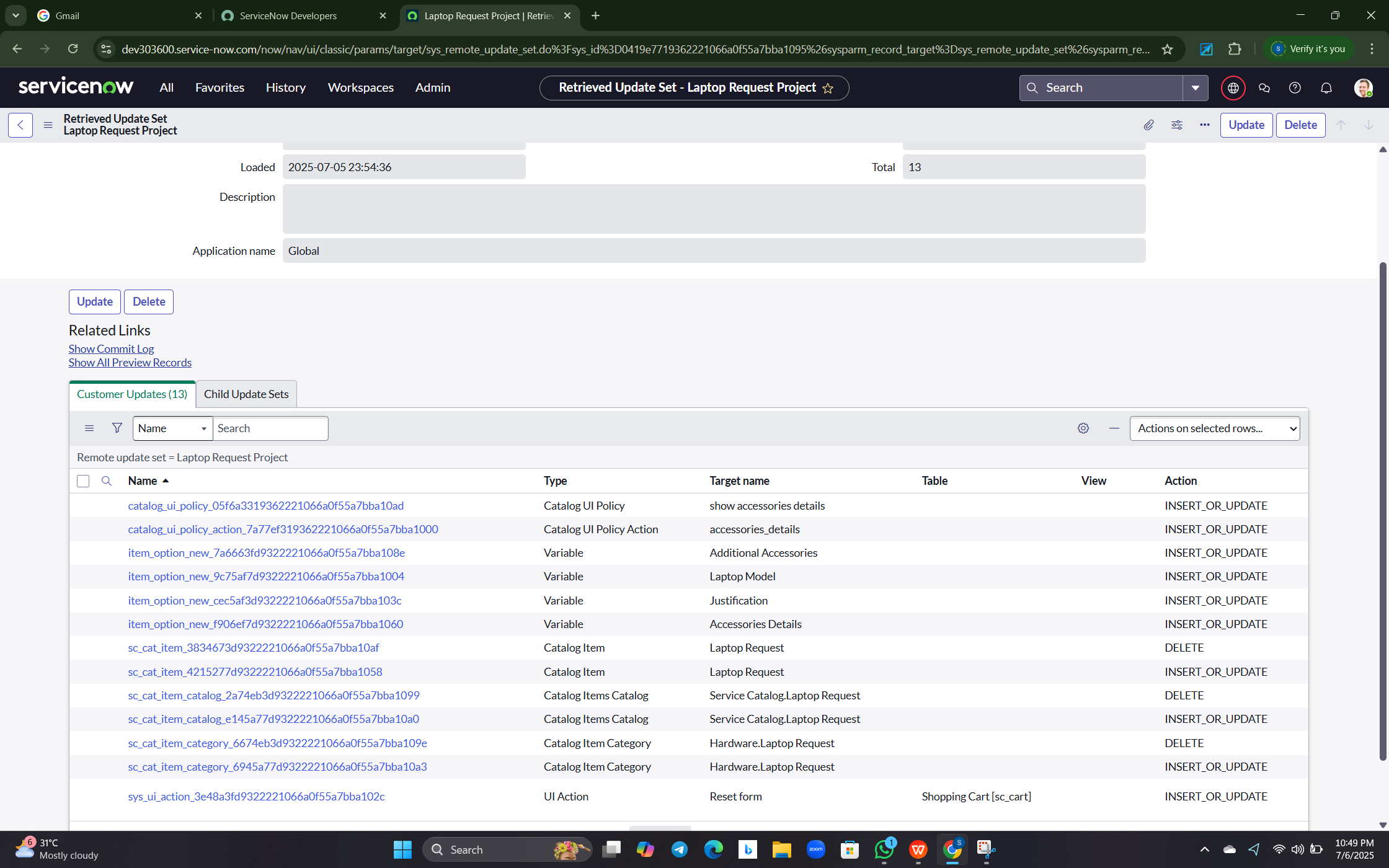


**5.Creating a UI Action: "Reset Form"**

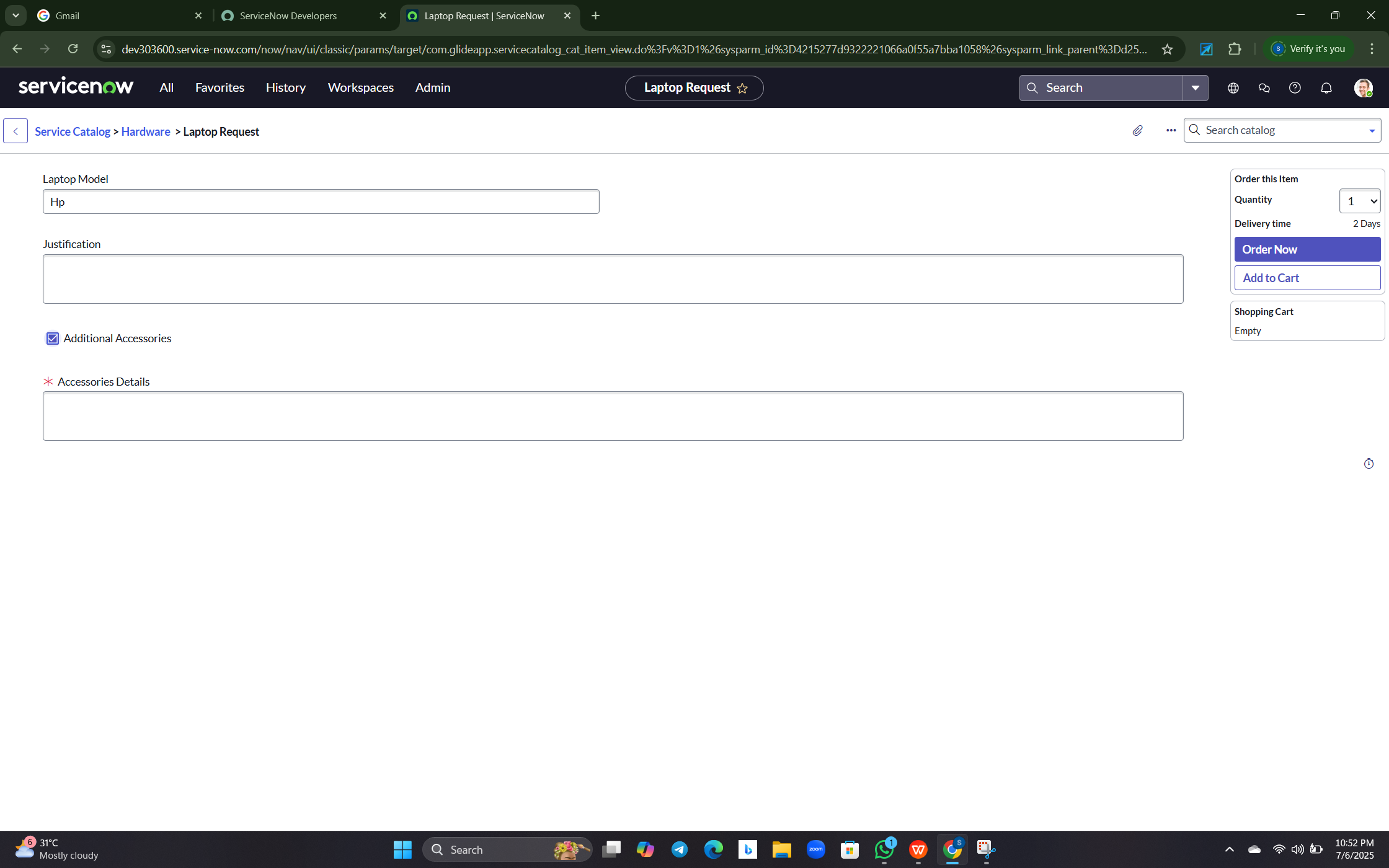


**6.Exporting Changes**



**7.Retrieving and Importing Update Set** 

**8.Testing and Verification**



**1. Appendix**

• Source Code: No external code; used in ServiceNow platform

• Dataset Link: Not applicable

• GitHub & Project Demo Link:

<https://github.com/Honson-max/Laptop-Request-Catalog-Item>

• Demo video: <https://1drv.ms/v/c/692726FD35D06D85/EYp5MnSfz5VJnKoSpQoEb08BwvksMF9kJstTzKtpxOJ07A?e=zDT2CO>

• Drive Link: <https://1drv.ms/v/c/692726FD35D06D85/EYp5MnSfz5VJnKoSpQoEb08BwvksMF9kJstTzKtpxOJ07A?e=zDT2CO>