***LAPTOP REQUEST CATALOG ITEM***

**INTRODUCTION**

The Laptop Request Catalog Item is a digital solution developed as part of the Naan Mudhalvan skill development initiative, which aims to empower students and professionals across Tamil Nadu with industry-relevant capabilities. This project focuses on streamlining the laptop requisition process within an organization or institution by leveraging the ServiceNow platform.

**⚠️ Summary of the Current Challenges**

Despite the importance of laptop provisioning, the existing process is:

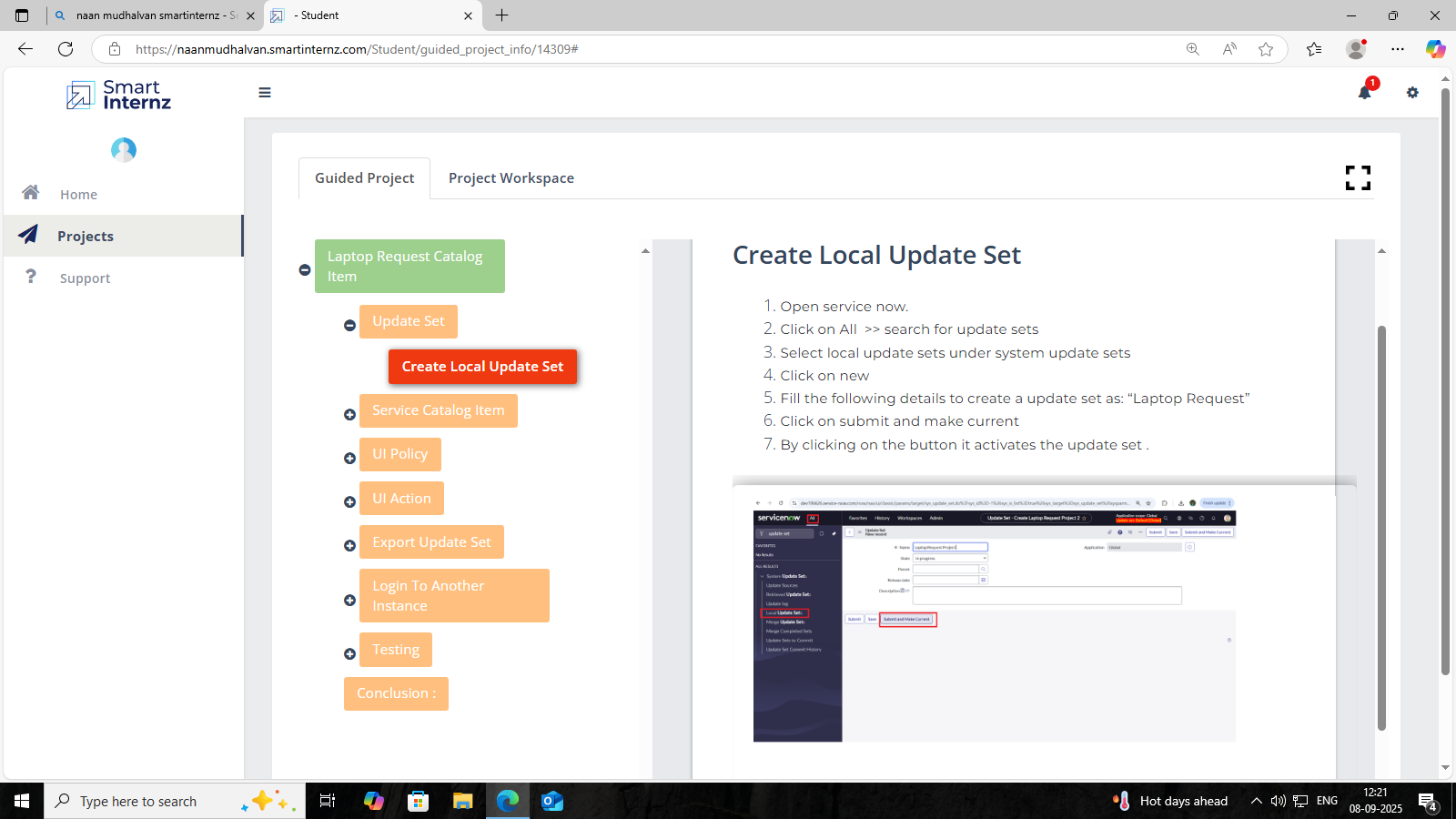
* Manual and time-consuming, often involving emails, paper forms, or verbal requests.
* Prone to delays and errors, due to lack of standardized input and validation.
* Lacking dynamic form behavior, which leads to confusion and incomplete submissions.
* Difficult to track, with no centralized dashboard or status updates for users or administrators.
* Governance-deficient, making it hard to audit changes or manage deployments across environments.

**🎯 Project Goals for Laptop Request Catalog**

* **Create a Service Catalog Item**
  + Design a user-friendly form for employees to request laptops.
  + Include essential fields like laptop type, justification, and urgency.
* **Implement Form Behavior**
  + Configure submission logic, approval workflows, and notification triggers.
  + Ensure the form behaves intuitively for both requesters and approvers.
* **Automate with Flow Designer**
  + Link the catalog item to a Flow Designer flow to automate request handling.
  + Include steps like approval routing, task creation, and fulfillment tracking.
* **Apply UI Policies**
  + Use UI policies to enhance usability (e.g., show/hide fields based on user input).
  + Ensure the form complies with organizational standards and reduces errors.
* **Track Changes with Update Set**
  + Bundle all configurations into an update set for version control and deployment.
  + Facilitate governance and easy migration across ServiceNow instances.
* **Enable Testing and Validation**
  + Test the catalog item thoroughly to ensure functionality and user experience.
  + Validate that automation and UI policies work as intended.

**🛠️ What Is a Local Update Set?**

An **Update Set** is like a container that tracks changes you make to configurations, customizations, or development work. It allows you to **package and move** those changes from one instance (like a development environment) to another (like testing or production).



**🧭 Step-by-Step Summary**

1. Open ServiceNow– Log into your ServiceNow instance.

2. Navigate to Update Sets – Use the application navigator to search for “Update Sets.”

3. Select Local Update Set – Choose the option under “System Update Sets.”

4. Click ‘New’ – This opens a form to create a new update set.

5. Fill Mandatory Fields – For example, name it something like "Laptop Request" to reflect the purpose.

6. Submit the Form – This saves your new update set.

7. Activate the Update Set– Click on the position field to make it the current active update set. This means all changes you make will be tracked under this set.

**📦 What Is a Service Catalog Item?**

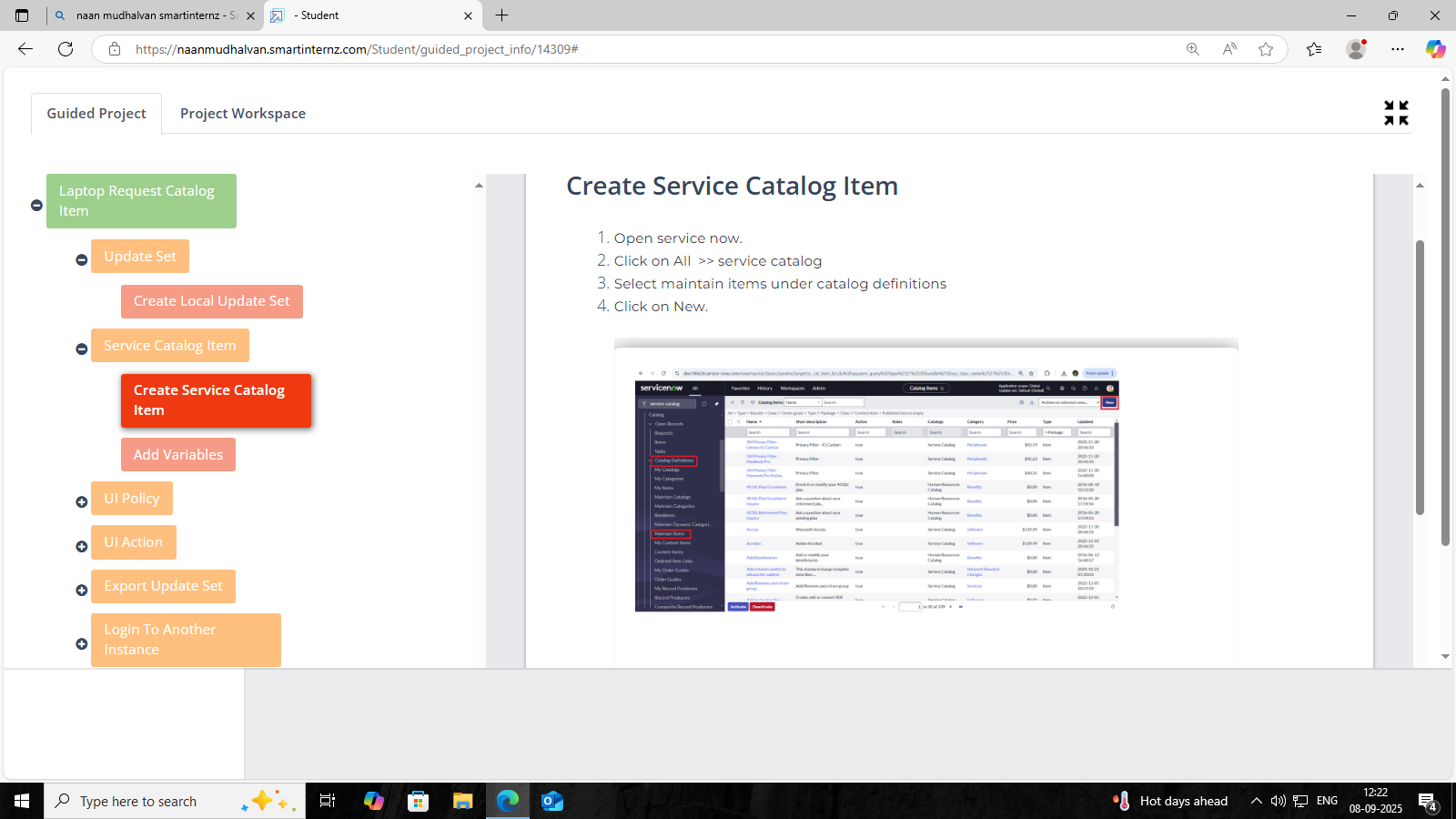
A Service Catalog Item is like a product or service offering that users can request—think of it like an online shopping item.

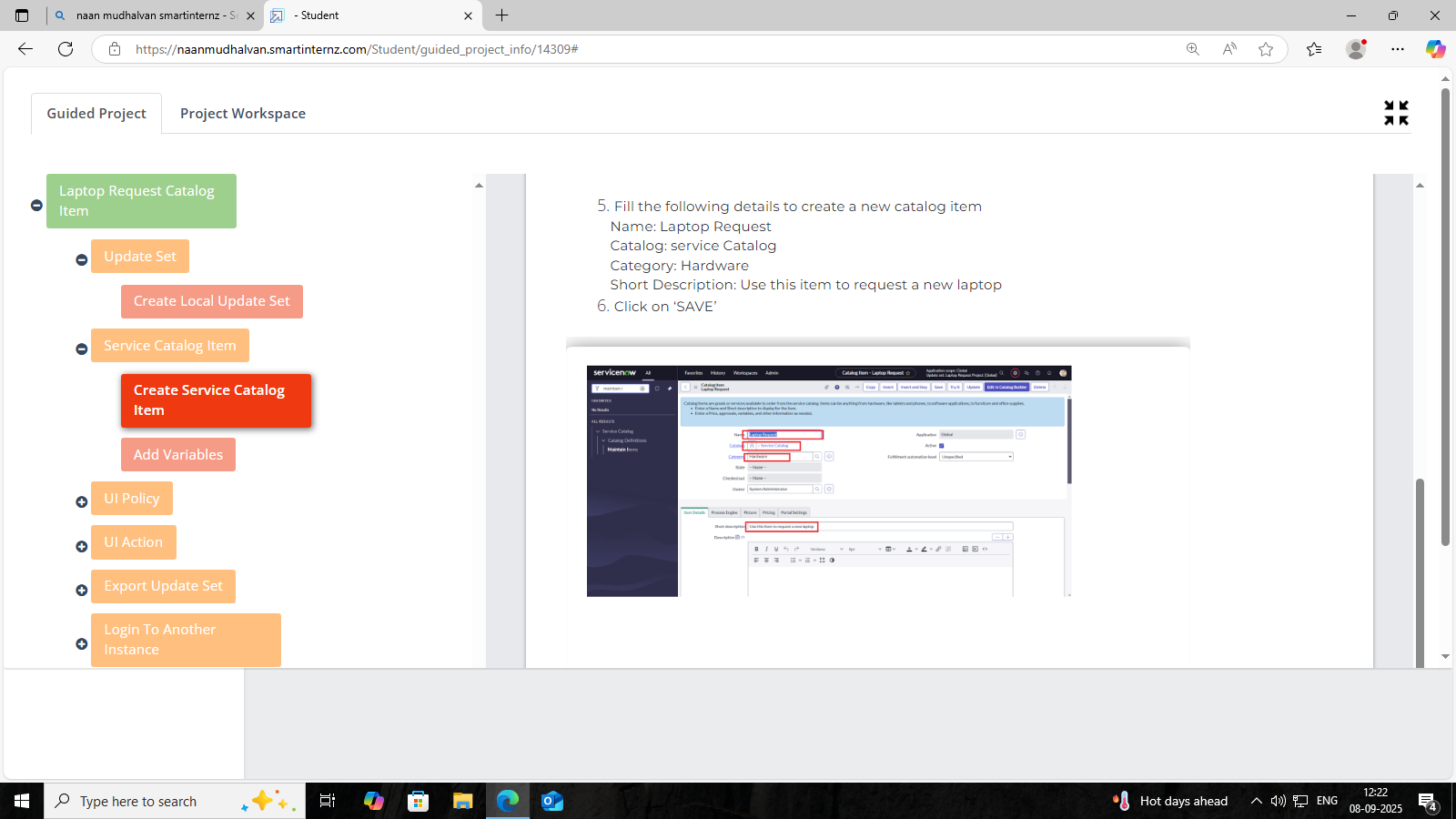
Examples include:

- Requesting a laptop

- Asking for software installation

- Submitting a leave application

These items are part of the Service Catalog, which helps streamline service delivery across IT and other departments.



**🛠️ Steps to Create a Service Catalog Item**

1. Open ServiceNow – Log into your instance.

2. Navigate to Service Catalog – Use the Application Navigator:

All → Service Catalog

3. Go to Maintain Items – Under Catalog Definitions, select Maintain Items. This is where all catalog items are listed and managed.

4. Click on ‘New’ – This opens a form to create a new catalog item.

5. Form Details for Creating a Catalog Item

Name: Laptop Request

This is the title of the item users will see when browsing the catalog.

- Catalog: Service Catalog

This defines which catalog the item belongs to. “Service Catalog” is the default and most commonly used one.

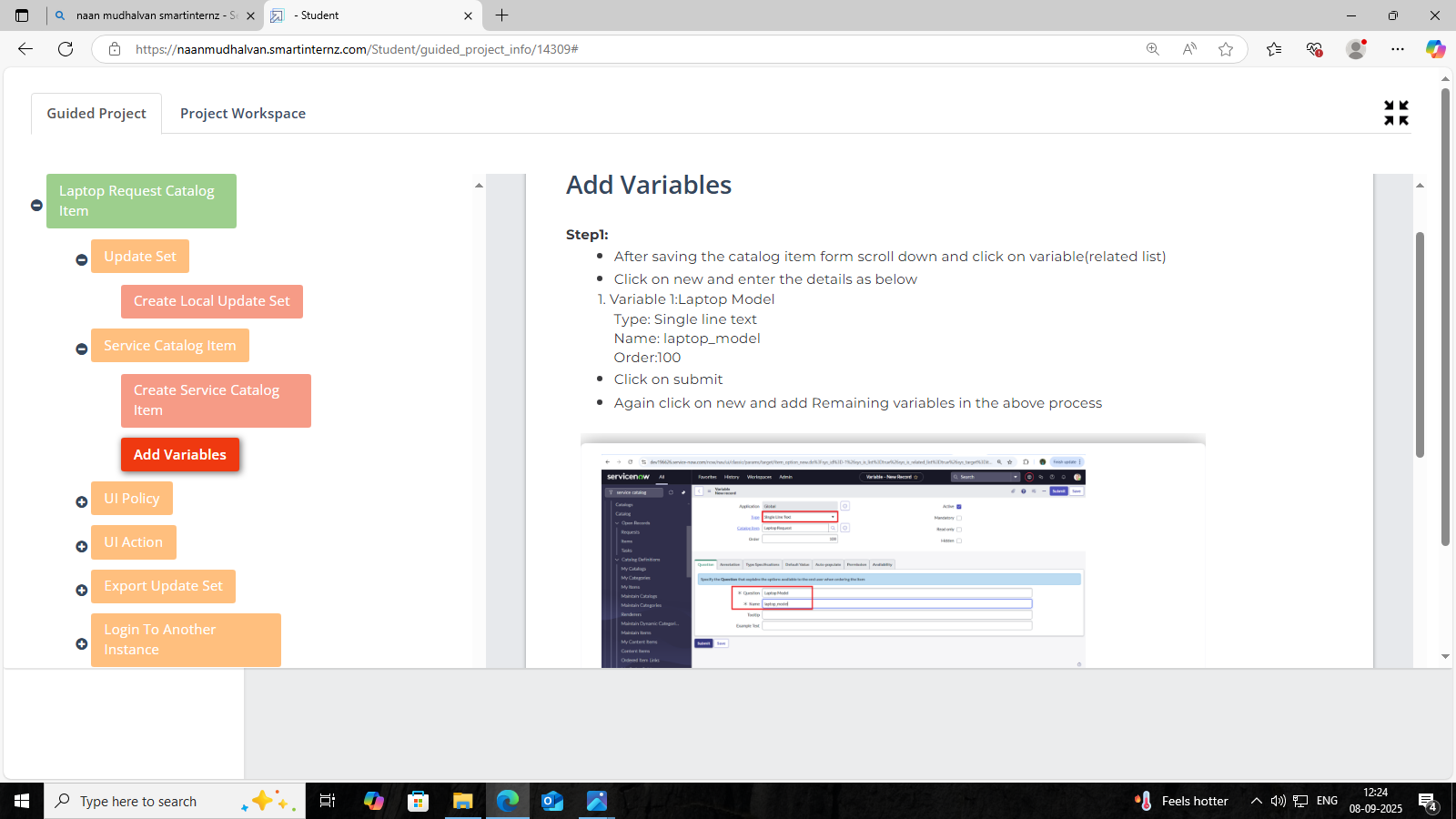
- Category: Hardware

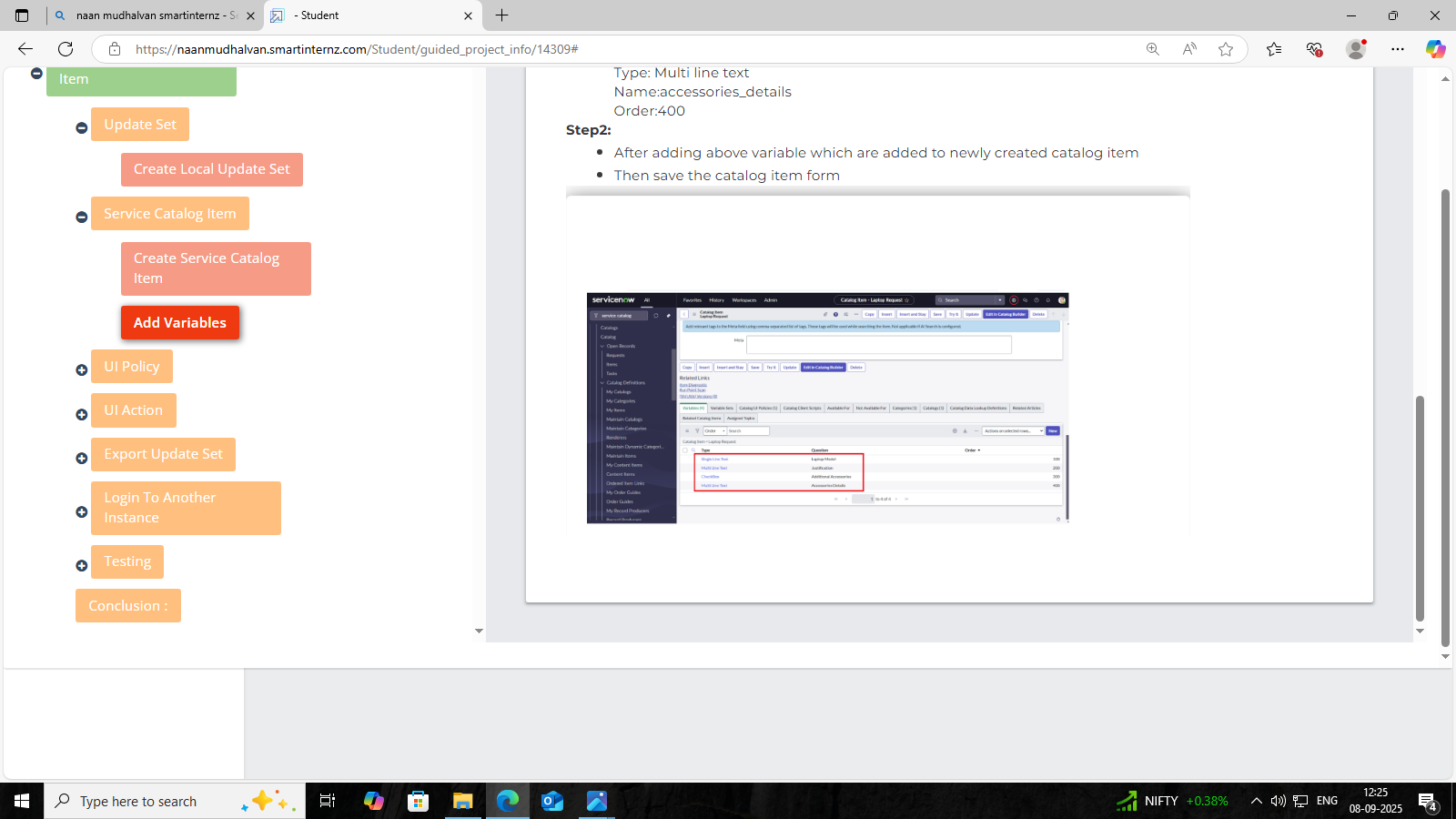
This helps organize items into logical groups. “Hardware” makes sense for a laptop request.

6. Final Step: Click “SAVE”.

**🧮 What Are Variables?**

Variables are \*form fields\* that users fill out when submitting a catalog item request. They help gather specific information needed to fulfill the request. Think of them like questions on a form—text boxes, dropdowns, checkboxes, etc.





**🛠️ Steps**

Scroll Down to the Variables Section

After saving your catalog item (e.g., “Laptop Request”), scroll to the related list section and click on Variables.

1️⃣ Click ‘New’ to Add a Variable

You’re creating a new input field for the form.

Fill Out the Variable Details

- Type: Single Line Text

- Name: laptop\_model (used internally)

- Question: Model (this is what users will see)

- Order: 10 (controls the display sequence)

Click ‘Submit’

This saves the variable.

Repeat for Additional Variables

You can add more fields like RAM size, preferred brand, etc., using the same process.

2️⃣ Variable: Justification

- Type: Multi Line Text

- Name: justification

- Order: 200

- Purpose: Lets the user explain why they need the laptop—useful for approvals or prioritization.

3️⃣ Variable: Additional Accessories

- Type: Checkboxes

- Name: additional\_accessories

- Order: 300

- Purpose: Allows users to select extra items like mouse, keyboard, or docking station.

4️⃣ Variable: Accessories Details

- Type: Multi Line Text

- Name: accessories\_details

- Order: 400

- Purpose: Lets users specify which accessories they want or any special requirements.

**💾 Final Step**

Once these variables are added to the catalog item, the instruction says to save the catalog item form.

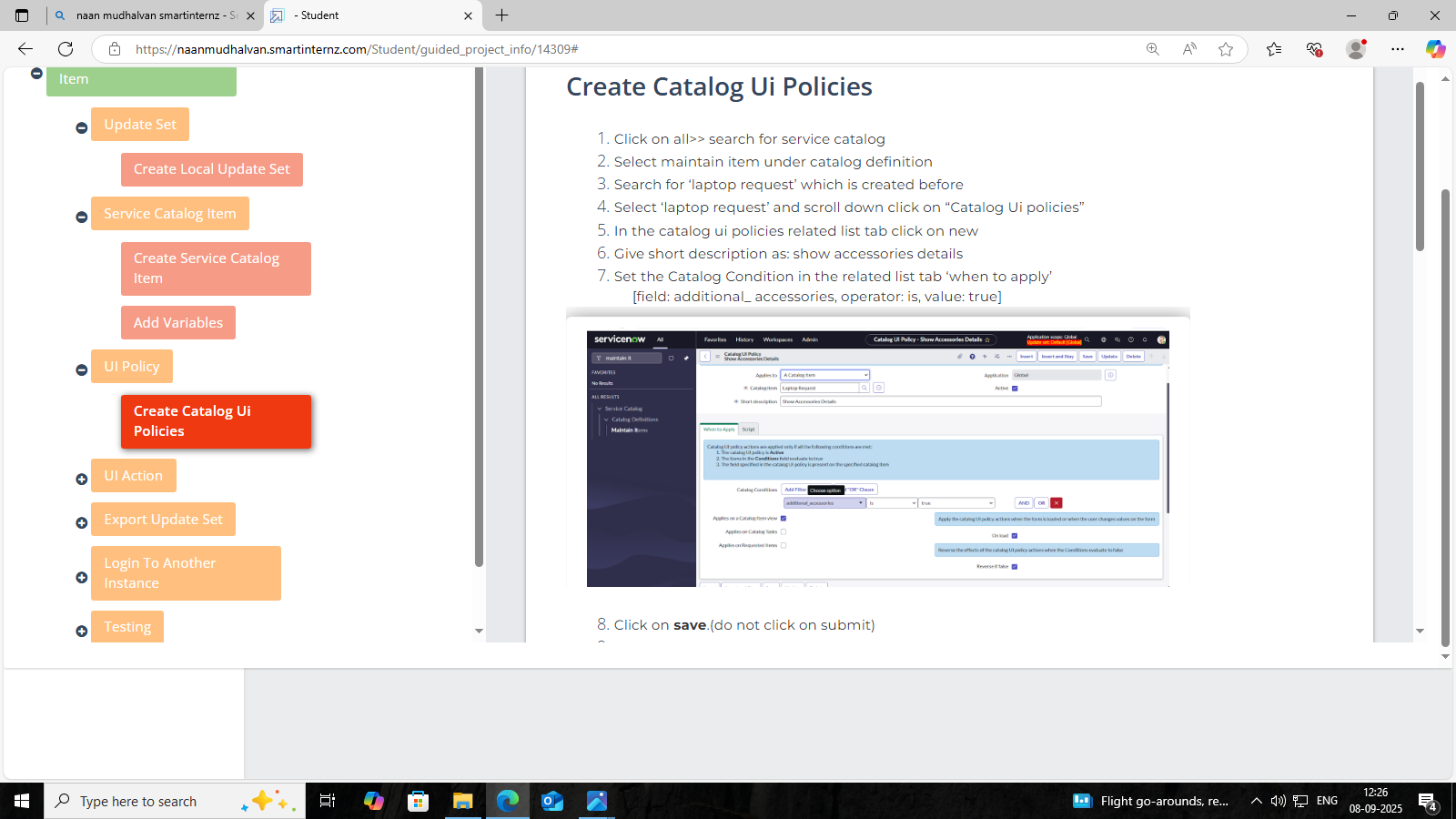
**🎯 What Is a Catalog UI Policy?**

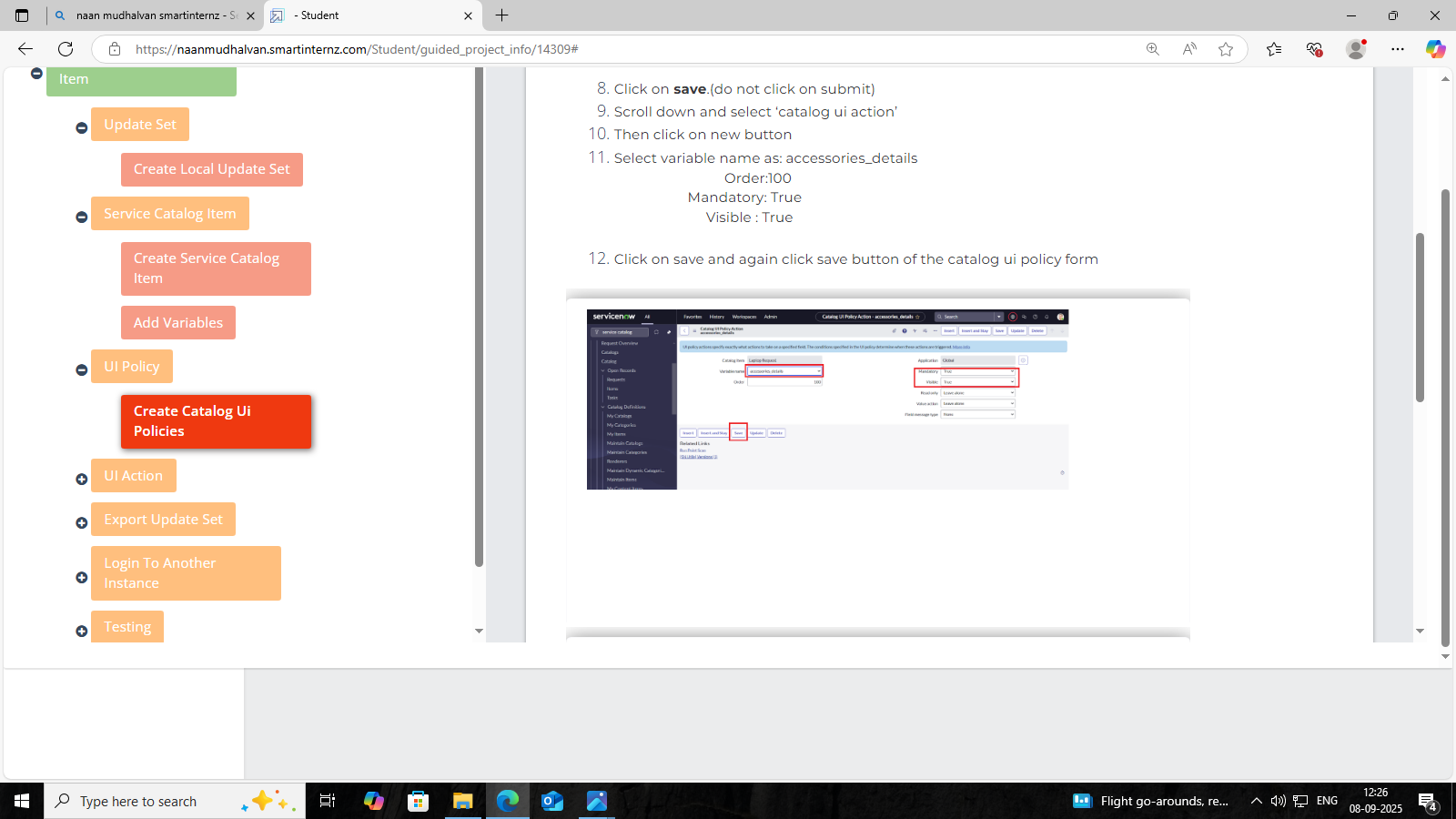
A Catalog UI Policy lets you dynamically control how fields behave on the catalog item form based on conditions. For example:

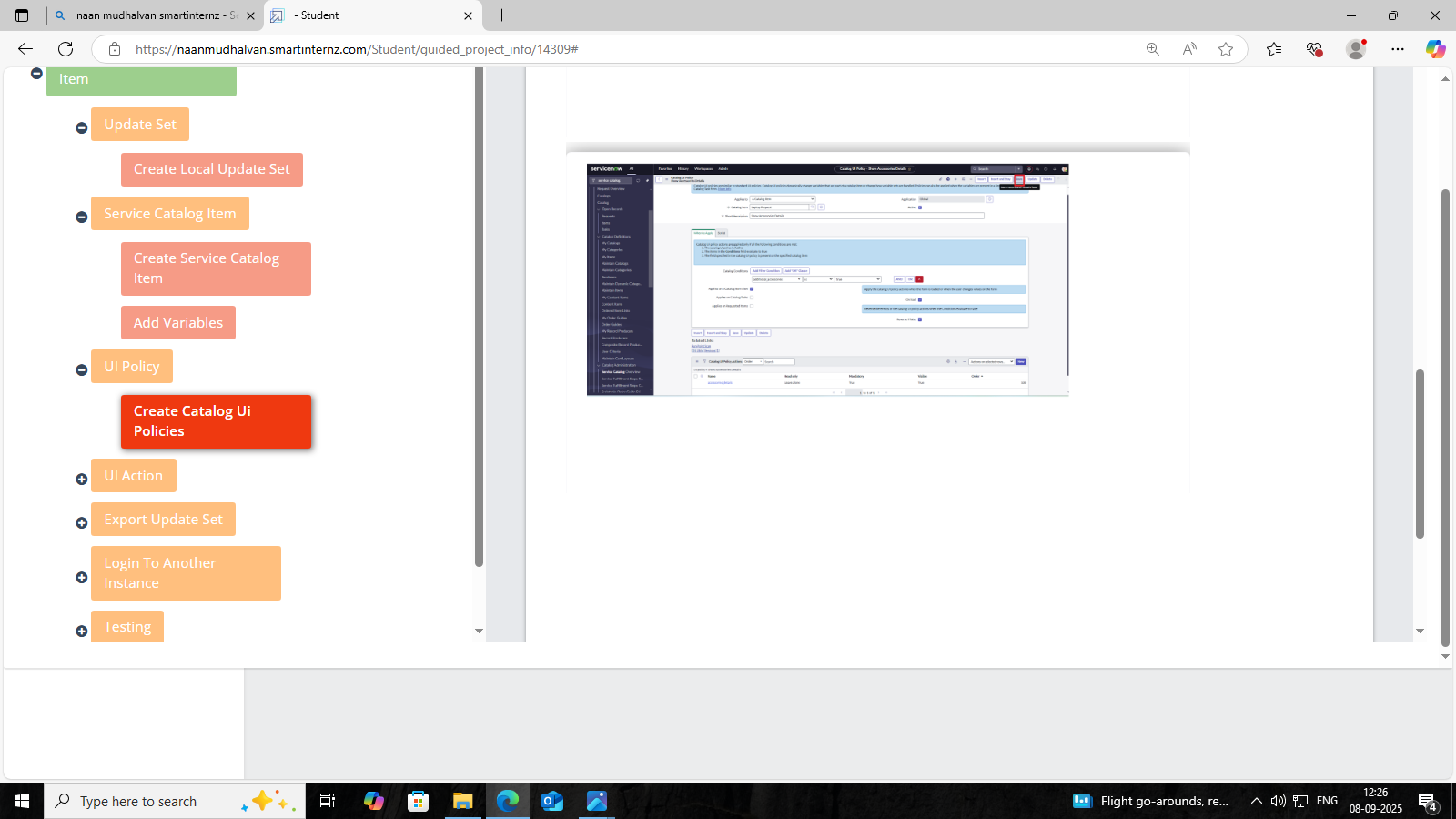
- Show or hide fields

- Make fields mandatory

- Set fields to read-only



****

****

**🛠️ Instructions**

1. Open Service Catalog

Use Alt + cn to search for “Service Catalog” in the navigator.

2. Search for the Catalog Item

Go to:

Service Catalog → Maintain Items → Search for "Laptop"

3. Open the Catalog Item

Click on the item you created earlier (e.g., “Laptop Request”).

4. Go to the UI Policies Related List

Scroll down to the Catalog UI Policies tab and click New.

5. Create a New UI Policy

- Name: Visible Condition

- Catalog Item: Select “Laptop”

- Condition: Define when the policy should apply (e.g., when a certain variable is set to true)

6. Select the Catalog Item as "Laptop Request"

This step links the UI Policy to a specific catalog item—in this case, Laptop Request.

7. Define the UI Policy Condition

- When to Apply: Always

This means the policy will run every time the form loads or changes, regardless of user input.

- Run Scripts: True

This enables the policy to execute client-side scripts if needed. It’s useful when you want to add custom logic beyond basic field actions.

Why it matters:

This setup ensures the policy is always active and ready to run any additional logic you define—like showing the “Accessories Details” field only when “Additional Accessories” is selected.

8. Click Save (not Submit)

This keeps the form open so you can continue editing without finalizing it yet.

9. Scroll Down and Select Catalog as Travel

This might be a misstep unless you're working on a travel-related item. If your item is “Laptop Request,” you’d typically keep the catalog as “Service Catalog” or “Hardware.”

10. Click Save Again

This confirms the catalog selection.

11. Configure the Variable Action

- Variable Name: accessories\_details

- Mandatory: True → User must fill this field

- Hidden: True → Field is hidden unless condition is met

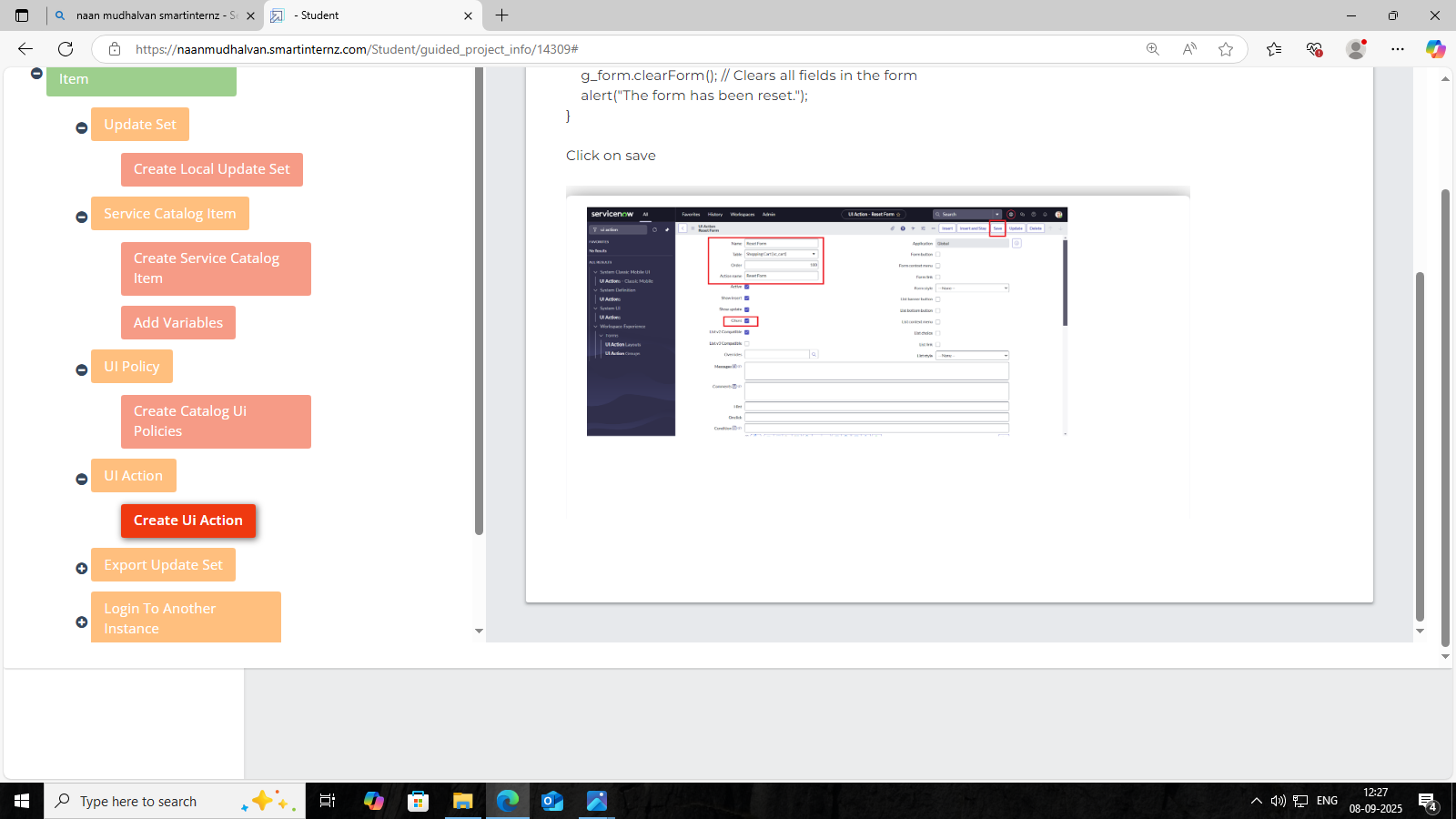
- Master Table: True → Applies across multiple catalog items if needed

12. Click Save Again

This finalizes the UI policy action setup.

**⚙️ What This UI Action Does?**

You're creating a button called "Reset form" on the Shopping Cart (sc\_cart) table. When clicked, it clears all fields and shows a confirmation alert.



**🛠️ Key Configuration Details**

- Table: sc\_cart (Shopping Cart)

- Order: 100 (controls button placement)

- Action Name: Reset form

- Client: Checked (runs in the browser, not on the server)

**🧠 Script Function**

javascript

function resetForm() {

g\_form.clearForm(); // Clears all fields in the form

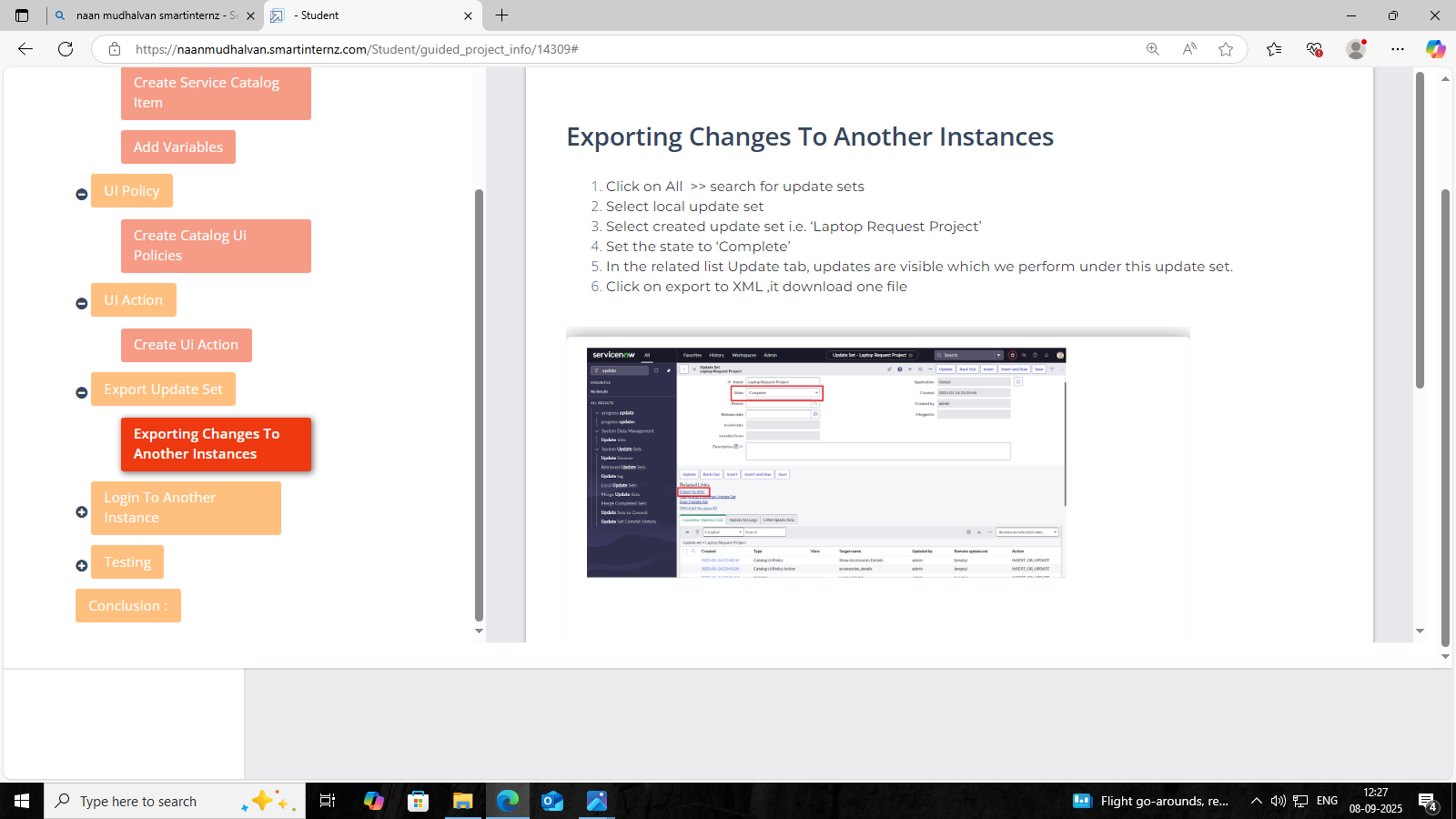
alert("The form has been reset.");

}

This script uses g\_form.clearForm() to wipe all user inputs and then displays a message.

**🚚 What Is Being Exported?**

You're exporting the "Laptop Request Project" update set, which contains all the changes you've made—catalog items, variables, UI policies, etc.—so they can be imported into another ServiceNow instance.



**🛠️ Steps Shown in the Image**

1. Search for Update Sets in the Application Navigator.

2. Select the Update Set you created (e.g., “Laptop Request Project”).

3. Open the Update Set Record.

4. View the Updates Tab to see all the changes captured.

5. Click “Export to XML” to download the update set as an XML file.

**🔄 Steps to Retrieve an Update Set**

1-2. Open Another Instance

Use an incognito window to access a different ServiceNow instance (e.g., production or test).

3. Search for Update Sets

In the Application Navigator, type “Update Sets.”

4. Select ‘Retrieved Update Sets’

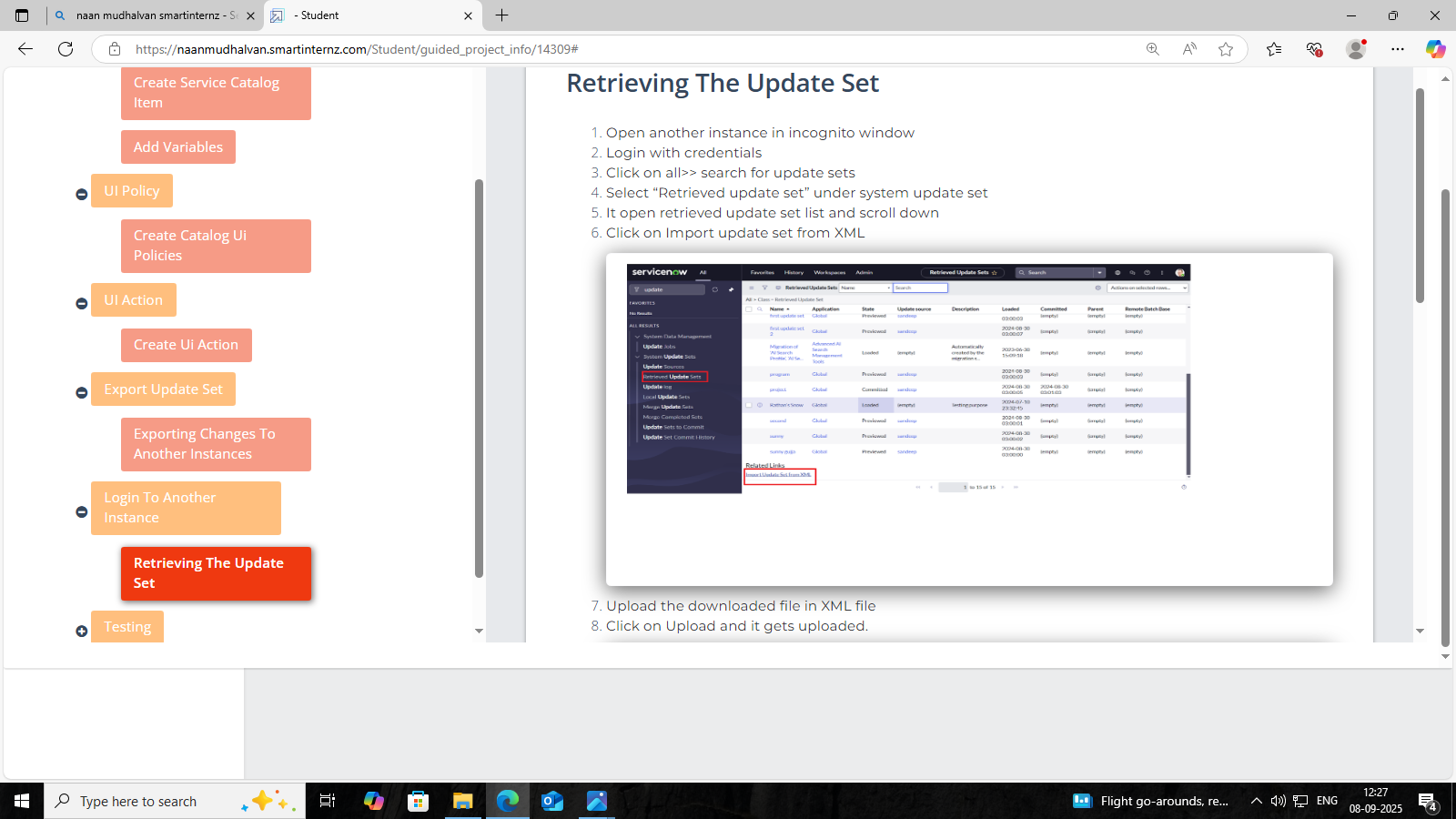
This shows all update sets that have been imported but not yet committed.

5. View the Retrieved List

Scroll through the list to find the one you imported.

6. Click on the Imported Update Set from XML

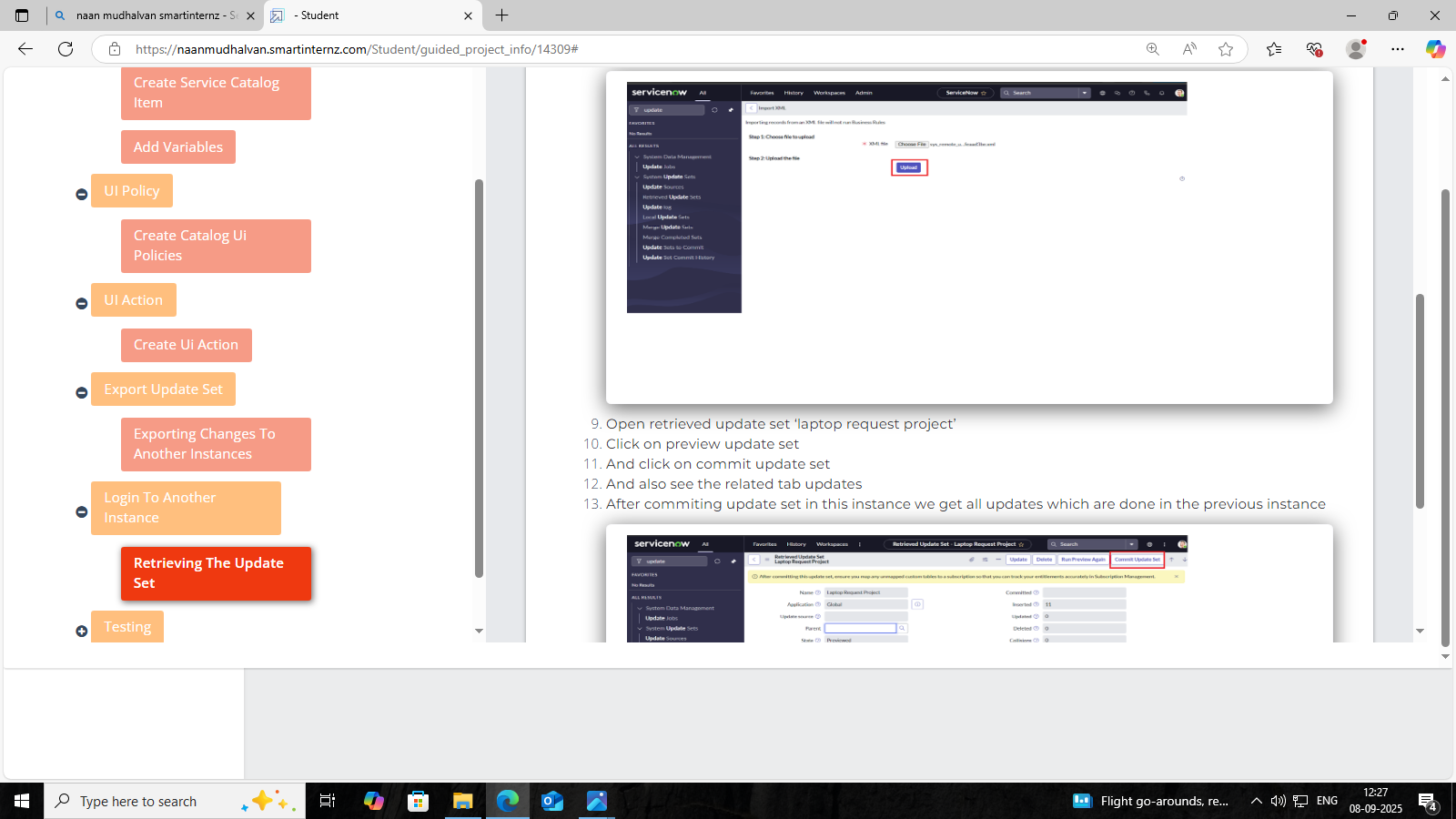
This opens the update set you downloaded earlier (e.g., “Laptop Request Project”).



7–8. Upload the XML File

- Use the “Retrieved Update Sets” section to upload the XML file you downloaded.

- Click Upload to bring it into the new instance.



9–10. Open and Preview the Update Set

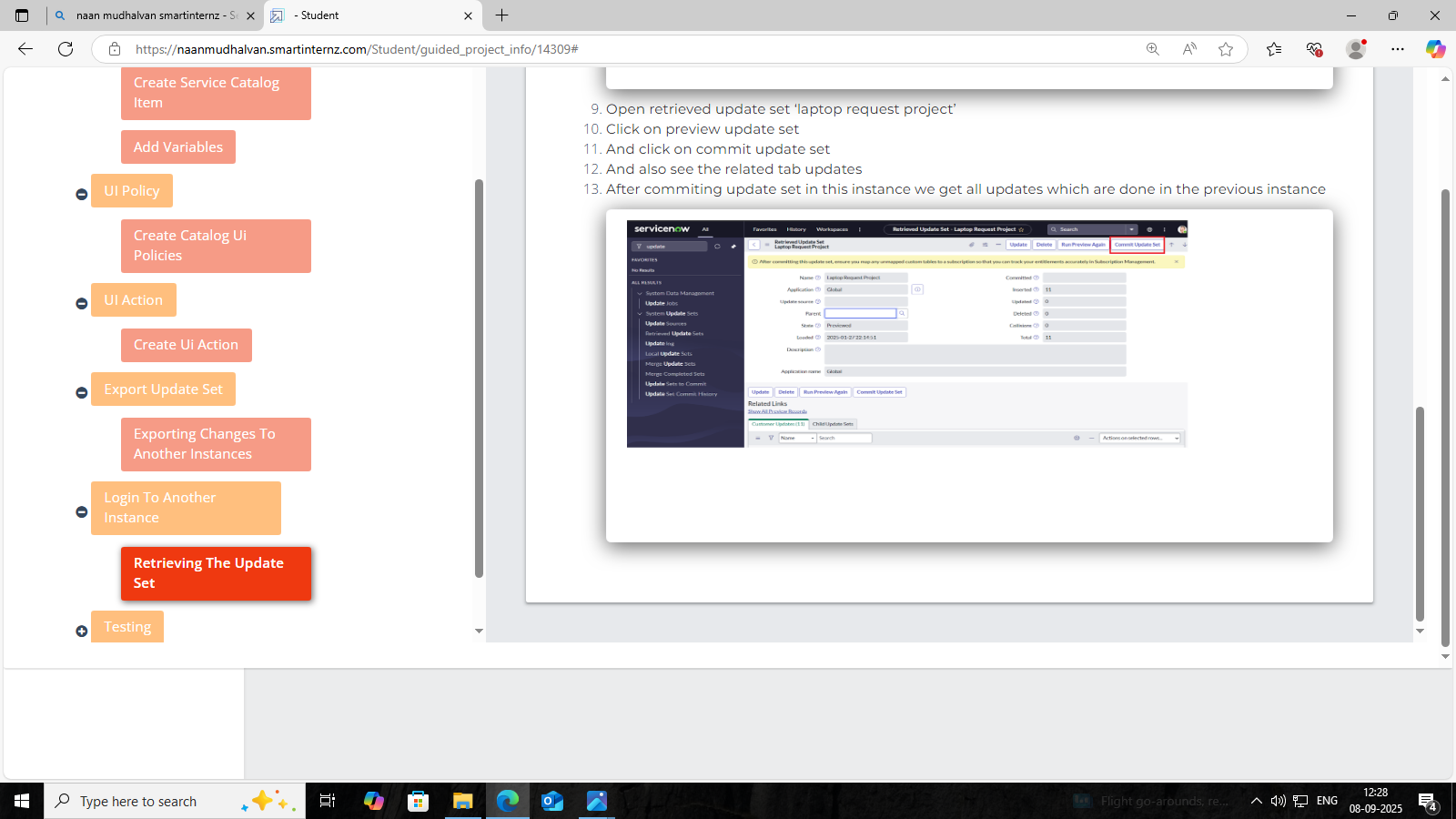
- Find the update set (e.g., “Laptop Request Project”) in the retrieved list.

- Click Preview Update Set to review all included changes and check for issues.

11–12. Retrieve Late Updates & Commit

- If there are any late updates (changes made outside the update set), you can retrieve them.

13. Finally, Commit the update set to apply all changes—your catalog item, variables, UI policies, and scripts will now be live in this instance.



**🧪 Steps to Test the Catalog Item**

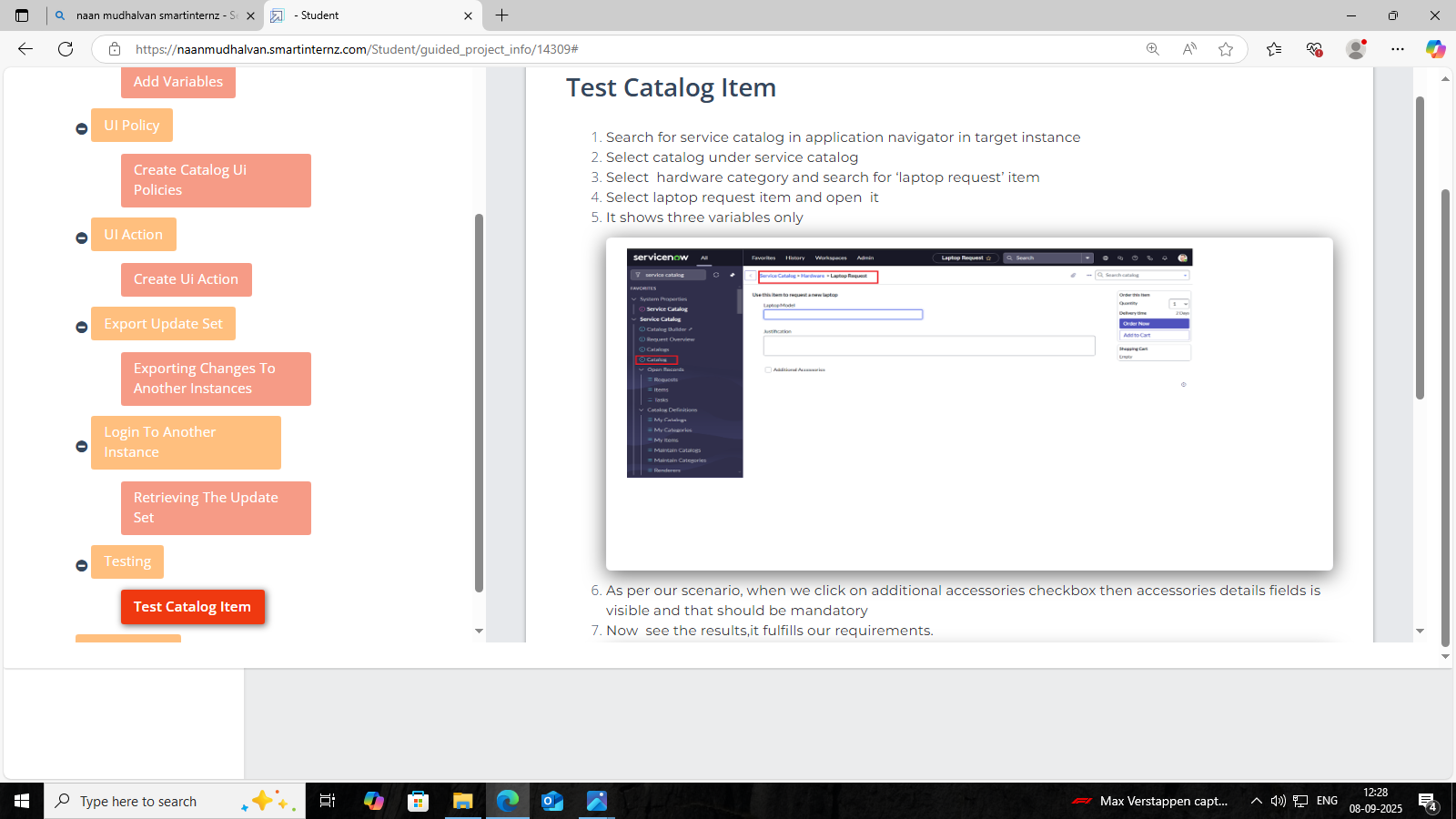
1. Search for “Service Catalog” in the Application Navigator.

2. Select the Catalog section under Service Catalog.

3. Find the Item you created earlier—“Laptop Request.”

4. Open the Item to view its form.

5. Fill in Mandatory Fields like model, justification, accessories, etc.

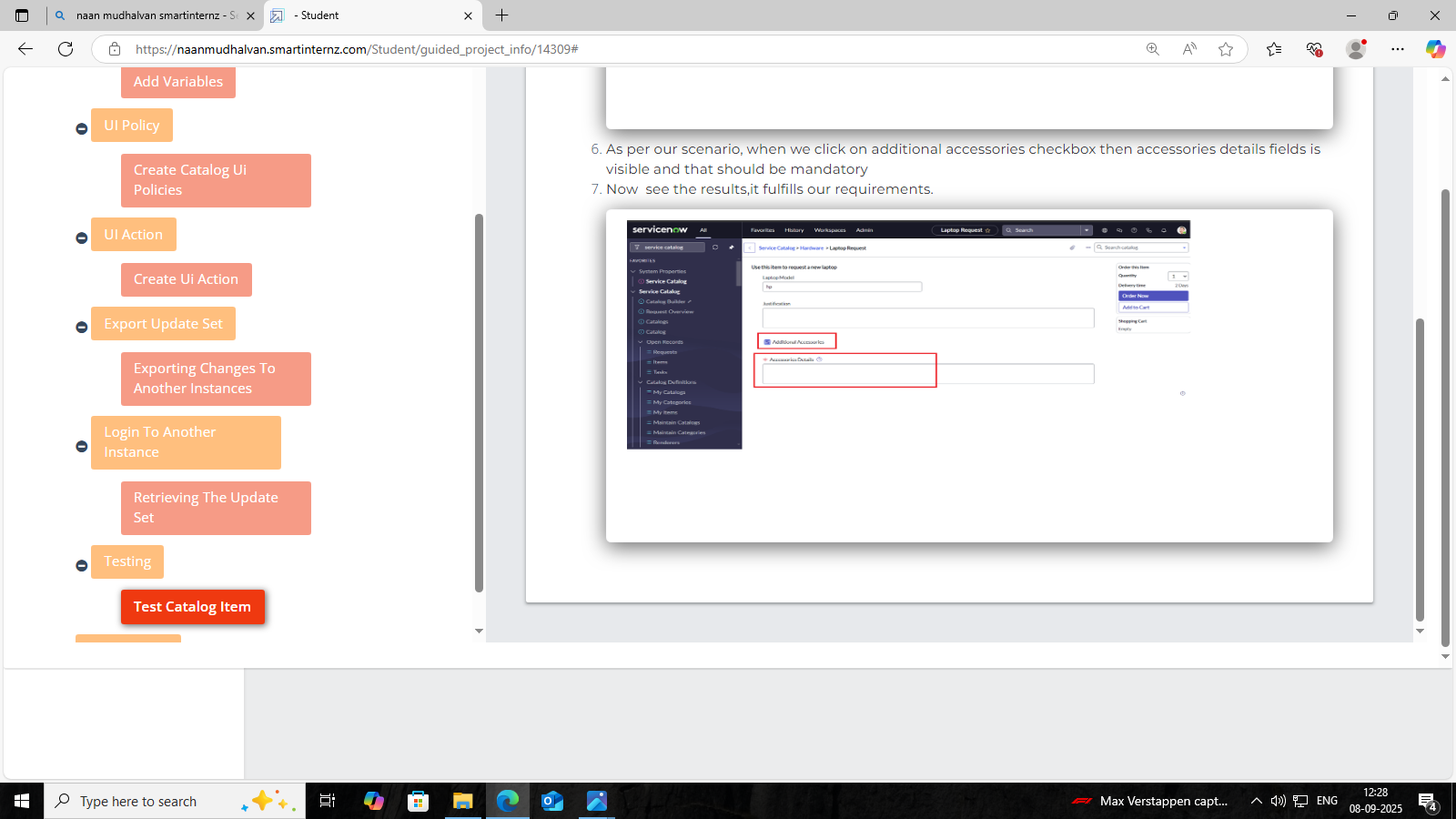


6. Checkbox Triggers Field Visibility and Mandatory Status

When the user selects the “Additional Accessories” checkbox, the “Accessories Details” field becomes both visible and mandatory—exactly as you configured using UI Policy and UI Policy Actions.

7. Validation of Expected Behavior

This step confirms that the form behaves as intended, fulfilling the scenario requirements. Your dynamic logic is working perfectly, enhancing both usability and data accuracy.



**CONCLUSION**

**✅ Streamlined Laptop Requests:**

The catalog item simplifies how employees request laptops, offering a clean and intuitive interface.

**⚙️ UI Actions & Workflows:**

You’ve implemented dynamic form behavior (like showing accessories fields conditionally), reset actions, and automated approval flows.

**🧪 Testing & Validation:**

The catalog item was thoroughly tested to ensure it works as expected, with mandatory fields and conditional logic functioning properly.

**🚀 Efficiency & Productivity:**

By automating the request and fulfillment process, the project reduces manual effort and speeds up service delivery.

**💡 ServiceNow Best Practices:**

Your work demonstrates how to leverage ServiceNow’s capabilities to build scalable, user-friendly IT service management solutions.

**TEAM ID: NM2025TMID16186**

**TEAM :**

**NAVEEN RAJ P. K. (LEADER)**

**PRAVEEN KUMAR V.**

**RAJESH K.**

**MUTHU E.**