



MAR EPHRAEM COLLEGE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NAAN MUDHALVAN

NM1051- Service Now Administrator

A PROJECT WORK ON
Laptop Request and Catalog Item

SUBMITTED BY

BRIJISHA B S 961422104026

**MAR EPHRAEM COLLEGE OF ENGINEERING AND
TECHNOLOGY**

**MALANKARA HILLS, ELAVUVILAI, MARTHANDAM-629 171,
KANYAKUMARI DISRICT**



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING**

A PROJECT REPORT

2025-2026

This is to certify that this is the bonafide record of the work done by Mr./Mrs Register Number..... of the (sem/Year), BE/B.Tech Department of in the NM1050-Service Now Adminstrator.

Staff-in-charge

Head of Department

University Examination held on.....

Internal Examiner

External Examinar

Table of Contents

CHAPTER	CONTENT
1.	Problem Statement
2.	Abstract
3.	Introduction
4.	Objectives
5.	Methodology
6.	Architecture
7.	Workflow Explanation
8.	Benefits of Automation
9.	Results and Outcomes
10.	Future Scope
11.	Appendix
12.	Conclusion

LAPTOP REQUEST CATALOG ITEM

1. Problem Statement

Many organizations handle laptop requests manually—through emails or direct messages to IT. This process is **time-consuming, lacks transparency**, and often results in **delays or miscommunication**. There is a need for an **automated, standardized process** that allows employees to easily request laptops, track approval status, and receive updates in one place. From an IT administration perspective, this lack of a structured system makes it difficult to **monitor inventory**, manage stock levels, and maintain accurate asset records.

2. Abstract

In today's digital era, managing family expenses requires a systematic and transparent process. This project, titled "Calculating Family Expenses Using ServiceNow," focuses on leveraging

ServiceNow's platform to automate the process of tracking daily and family-based expenses. By using ServiceNow tables, forms, and relationships, the system enables users to enter daily expense details, associate them with specific family members, and automatically compute totals. This ensures accuracy, reduces manual work, and provides a centralized view of financial data.

3. Introduction

Families often struggle to maintain clear and accurate expense records. Manually maintaining these records through spreadsheets or notebooks is prone to errors, duplication, and inefficiency. ServiceNow provides an effective platform to automate such processes by using custom tables, business rules, and relationships. In this project, a Family Expenses application is developed within ServiceNow that automates expense entry, categorization, and calculation for each family member.

4. Objectives

- To automate the process of tracking and calculating family expenses.
- To provide an easy-to-use interface for recording daily expenses.
- To maintain relationships between family members and their daily expenses.
- To enhance visibility and reporting of financial data.
- To demonstrate how ServiceNow can be applied beyond ITSM for personal and household management.

5. Methodology

Step 1: Setting up ServiceNow Instance

- Sign up for a developer account at <https://developer.servicenow.com>.
- Navigate to **Personal Developer Instance** → **Request Instance** to create a new ServiceNow instance.
- Fill out the required information and submit the request.
- Once the instance is ready, log in using the provided credentials.
- The instance serves as the base platform for building and testing the Family Expense Management system.

Step 2: Creation of New Update Set

- Go to **All** → **Local Update Set** → **New**.
- Enter details:
 - **Name:** Family Expenses
- Click **Submit** and then **Make Current** to activate the update set for this project.

Step 3: Creation of Family Expenses Table

- Navigate to **All** → **Tables** → **New**.
- Enter details:
 - **Label:** Family Expenses
 - **Name:** Auto-Populated
 - **New Menu Name:** Family Expenditure
- Save the configuration by right-clicking the header and selecting **Save**.

Step 4: Creation of Columns (Fields)

- Add new fields by double-clicking **Insert a new row** near the Columns section.
 - **Column Label:** Number → **Type:** String
 - **Column Label:** Date → **Type:** Date
 - **Column Label:** Amount → **Type:** Integer
 - **Column Label:** Expense Details → **Type:** String → **Max length:** 800
- Right-click the header and click **Save** to store the column configurations.

Step 5: Making Number Field an Auto-Number

- Open the **Number** field and enable **Advanced View**.
- In **Default Value**, check **Use Dynamic Default** and set **Dynamic Default Value** to *Get Next Padded Number*.
- Navigate to **All** → **Number Maintenance** → **New**.

- Enter details:
 - **Table:** Family Expenses
 - **Prefix:** MFE
- Click **Submit** to enable automatic numbering for new records.

Step 6: Configure the Form

- Navigate to **All** → **Family Expenses** → **Open Family Expenses** → **New**.
- Right-click the header → **Configure** → **Form Design**. Customize the layout as required.
- Make **Number** a *Read-Only Field*.
- Make **Date** and **Amount** *Mandatory Fields*.
- Click **Save** to apply the form configuration.

Step 7: Creation of Daily Expenses Table

- Navigate to **All** → **Tables** → **New**.
- Enter details:
 - **Label:** Daily Expenses
 - **Name:** Auto-Populated
 - **Add Module to Menu:** Family Expenditure
- Right-click the header and click **Save**.

Step 8: Creation of Columns (Fields) for Daily Expenses Table

- Add new fields by inserting rows under the Columns section:
 - **Column Label:** Number → **Type:** String
 - **Column Label:** Date → **Type:** Date
 - **Column Label:** Expense → **Type:** Integer
 - **Column Label:** Family Member Name → **Type:** Reference
 - **Column Label:** Comments → **Type:** String → **Max length:** 800
- Right-click the header and click **Save**.

Step 9: Configure the Form

- Navigate to **All** → **Daily Expenses** → **Open Daily Expenses** → **New**.
- Right-click the header → **Configure** → **Form Design**.
- Customize as needed.
- Make **Number** field *Read-Only*.
- Make **Date** and **Family Member Name** *Mandatory Fields*.
- Save the changes.

Step 10: Creation of Relationship between Family Expenses and Daily Expenses Tables

- Go to **All** → **Relationships** → **New**.
- Enter details:
 - **Name:** Daily Expenses
 - **Applies to Table:** Family Expenses
 - **Daily Expenses:** Daily Expenses
- Click **Save** to establish the relationship.

Step 11: Configuring Related List on Family Expenses

- Navigate to **All** → **Family Expenses** → **Open Family Expenses** → **New**.
- Right-click the header → **Configure** → **Related Lists**.
- Add **Daily Expenses** to the selected area.
- Click **Save** to finalize the related list configuration.

Step 12: Creation of Business Rules

- Navigate to **All** → **System Definition** → **Business Rules** → **New**.
- Enter details:
 - **Name:** Family Expenses BR
 - **Table:** Daily Expenses
- Check **Advanced**, and under *When to run*, check **Insert** and **Update**.
- Add the script in the **Advanced** section.
- Save the business rule to automate updating expenses in the *Family Expenses* table whenever a new *Daily Expense* record is added.

Step 13: Configure the Relationship

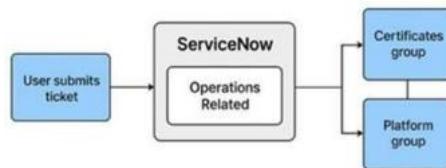
- Navigate to **All** → **Relationships** → **Open Daily Expenses Relationship**.
- Set **Applies to Table** as *Family Expenses*.
- In the **Query with** section, enter the script.
- Click **Update** to save the configuration.

6. Architecture Diagram

System Architecture

- The architecture consists of two main tables
 - **Family Expenses** and **Daily Expenses**
 - linked through a **reference relationship**.
- When a new daily expense is added, the system triggers a **Business Rule** that recalculates the total expenses for the related family member.

Streamlining Ticket Assignment for Efficient Support Operations



Flow Explanation

- User creates a **Daily Expense** record.
- The record is **linked to a Family Member**.
- The system **calculates and updates** the total automatically.

7. Workflow Explanation

Manual Process:

- Each family member records expenses manually.
- Summations are calculated periodically by hand.

Automated Process (ServiceNow Implementation):

- Daily expense entries are automatically linked to family members.
- Total expenses are calculated dynamically via business rules.
- Users can view detailed expense breakdowns per member in real-time.

8. Benefits of Automation

- **Accuracy:** Reduces human error in calculations.
- **Efficiency:** Eliminates manual summation tasks.
- **Transparency:** Provides clear visibility of expenses per family member.
- **Scalability:** Can be expanded for monthly or yearly tracking.
- **Integration:** Can integrate with dashboards for data visualization.

9. Results and Outcomes

After implementation, the **Family Expense Management System** achieved:

- Complete automation of expense calculation.
- Improved coordination among users (family members).
- Real-time updates to total expenses upon adding daily transactions.
- Simplified data entry and management within ServiceNow.
- Enhanced reporting and visibility across family members' expenses.

This clearly demonstrates the **efficiency and reliability** of ServiceNow's automation features in real-world management scenarios.

10. Future Scope

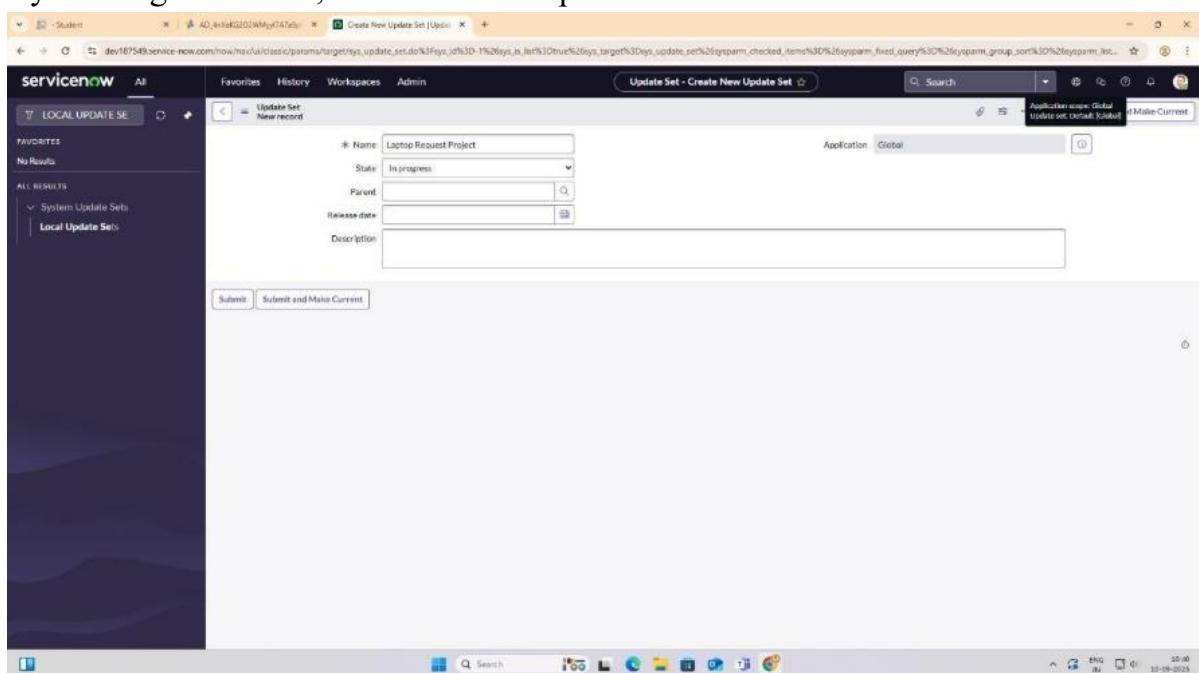
- Integration with **ServiceNow reports and dashboards** for analytics.
- Adding **budget limits and notifications** for overspending.
- Creating **charts and visual dashboards** for expense distribution.
- Extending the project for **income tracking and savings calculation**.
- Integration with **external payment or finance tracking APIs**.

11. TASK INITIATION

Milestone 1: Create Local Update Set

Activity 1: Create Update Set in ServiceNow

- Open the **ServiceNow instance**.
- Click on **All → Search for Update Sets**.
- Select **Local Update Sets** under *System Update Sets*.
- Click on **New**.
- Fill in the following details:
 - **Name:** Laptop Request
- Click **Submit** and then **Make it Current**.
- By clicking the button, it activates the update set.



Milestone 2: Create Service Catalog Item

Activity 1: Create New Service Catalog Item in ServiceNow

- Open **ServiceNow**.

- Click on All → Service Catalog.
- Select Maintain Items under Catalog Definitions.

The screenshot shows the ServiceNow interface for managing catalog items. The left sidebar is collapsed, and the main area displays a table of catalog items. The columns are: Name, Short description, Active, Roles, Catalogs, Category, Price, Type, and Updated. The table contains numerous entries, including various privacy filters, software like Acrobat, and hardware like iPhone models and MacBook accessories. A search bar at the top right allows filtering by name.

- Click on New and fill in the following details to create a new catalog item:
 - **Name:** Laptop Request
 - **Catalog:** Service Catalog
 - **Category:** Hardware
 - **Short Description:** Use this item to request a new laptop
- Click on Save.

The screenshot shows the 'Catalog Item - New Record' screen. The 'Name' field is populated with 'Laptop Request'. The 'Catalog' is set to 'Service Catalog', 'Category' is 'Hardware', and 'Active' is checked. The 'Description' field contains the text 'Use this item to request a new laptop'. The 'Short description' field also has the same text. The 'Owner' is listed as 'System Administrator'. The 'Submit' button is visible at the top right of the form.

Activity 2: Add Variables to the Catalog Item

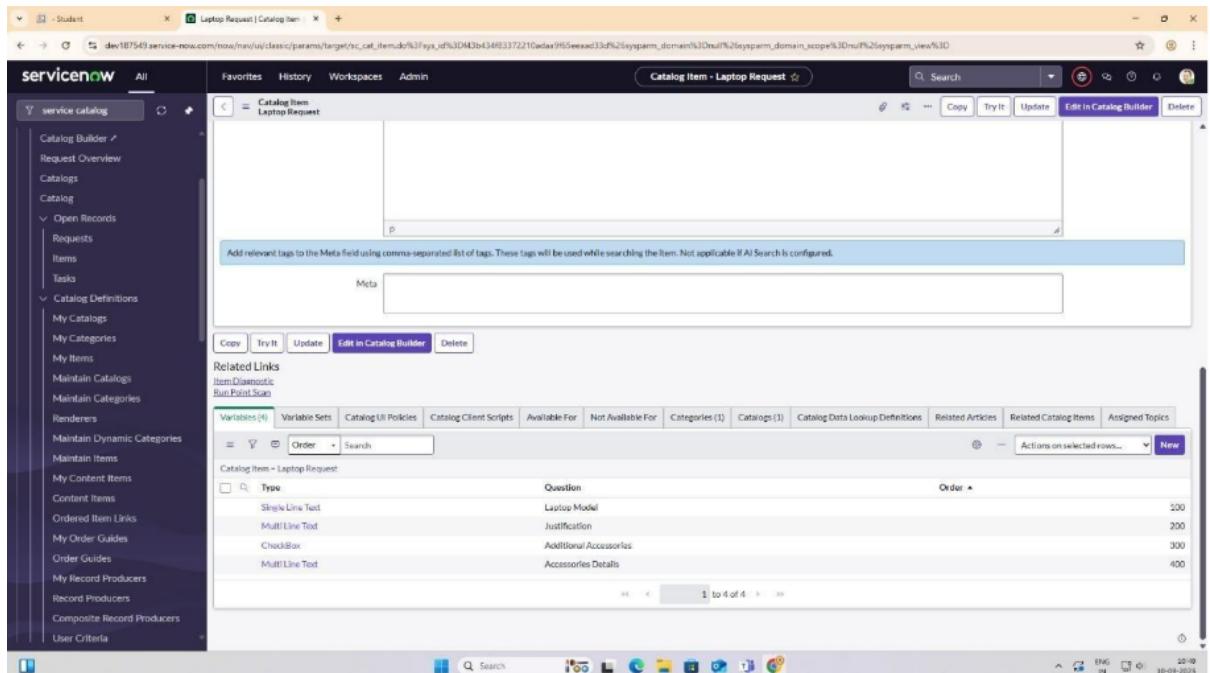
Step 1:

- After saving the catalog item form, scroll down and click on **Variables (Related List)**.
- Click on **New** and enter the details below:
 - **Variable 1: Laptop Model**
 - **Type:** Single Line Text
 - **Name:** laptop_model
 - **Order:** 100
 - Click on **Submit**.
- Again, click on **New** and add the remaining variables using the same process:

The screenshot shows the ServiceNow interface for creating a new variable. The left sidebar is titled 'service catalog' and lists various catalog-related options. The main window is titled 'Variable - New Record'. The 'Catalog Item' dropdown is set to 'Laptop Request'. The 'Order' field is set to 100. The 'Question' tab is active, showing the configuration for 'Variable 1: Laptop Model'. The 'Question' field contains 'Laptop Model', 'Name' is 'laptop_model', and 'Conversational label' is empty. The 'Submit' button is visible at the bottom.

- - **Variable 2: Justification**
 - **Type:** Multi Line Text
 - **Name:** justification
 - **Order:** 200
 - **Variable 3: Additional Accessories**
 - **Type:** Checkbox
 - **Name:** additional_accessories
 - **Order:** 300
 - **Variable 4: Accessories Details**
 - **Type:** Multi Line Text
 - **Name:** accessories_details
 - **Order:** 400
- **Step 2:**

- After adding the above variables, they are automatically linked to the newly created catalog item.
- Save the **Catalog Item Form**.



Milestone 3: Create Catalog UI Policies

Activity 1: Configure Catalog UI Policy for Dynamic Behavior

- Click on All → Search for Service Catalog.
- Select **Maintain Item** under *Catalog Definitions*.
- Search for the previously created **Laptop Request** item.
- Select **Laptop Request** and scroll down to click on **Catalog UI Policies**.
- In the *Catalog UI Policies* related list, click on **New**.
- Fill in the details:
 - **Short Description:** Show Accessories Details
 - **When to Apply:**
 - **Field:** additional_accessories
 - **Operator:** is
 - **Value:** true

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

* Catalog Item: Laptop Request

* Short description: show accessories details

Application: Global

Active:

When to Apply: Script

Catalog Conditions: additional_accessories is true

Applies on a Catalog Item view:

Applies on Catalog Tasks:

Applies on Requested Items:

On load:

Reverse if false:

[Submit](#)

- Click **Save** (do not click **Submit**).
- Scroll down and select **Catalog UI Actions**.
- Click on **New** and enter:
 - Variable Name:** accessories_details
 - Order:** 100
 - Mandatory:** True
 - Visible:** True
- Click **Save**, then again **Save** the *Catalog UI Policy Form*.

Catalog UI Policy Action - New Record

Catalog Item: Laptop Request

Variable name: accessories_details

Order: 100

Application: Global

Mandatory:

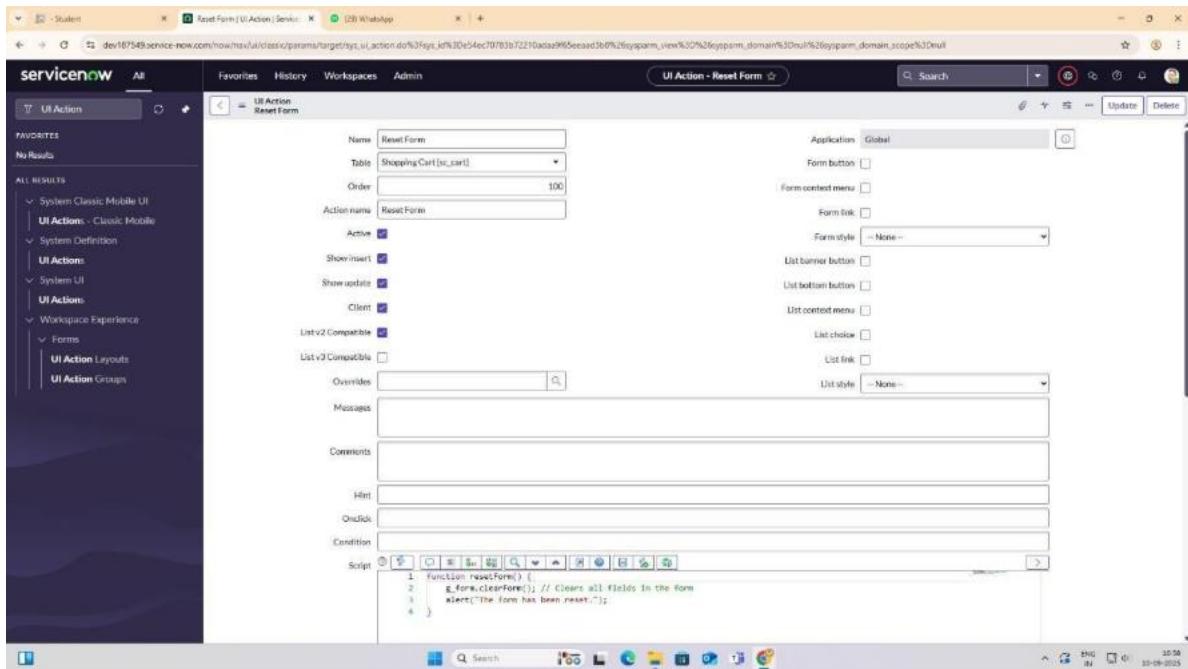
Visible:

Read only: Leave alone

Value action: Leave alone

Field message type: None

[Submit](#)



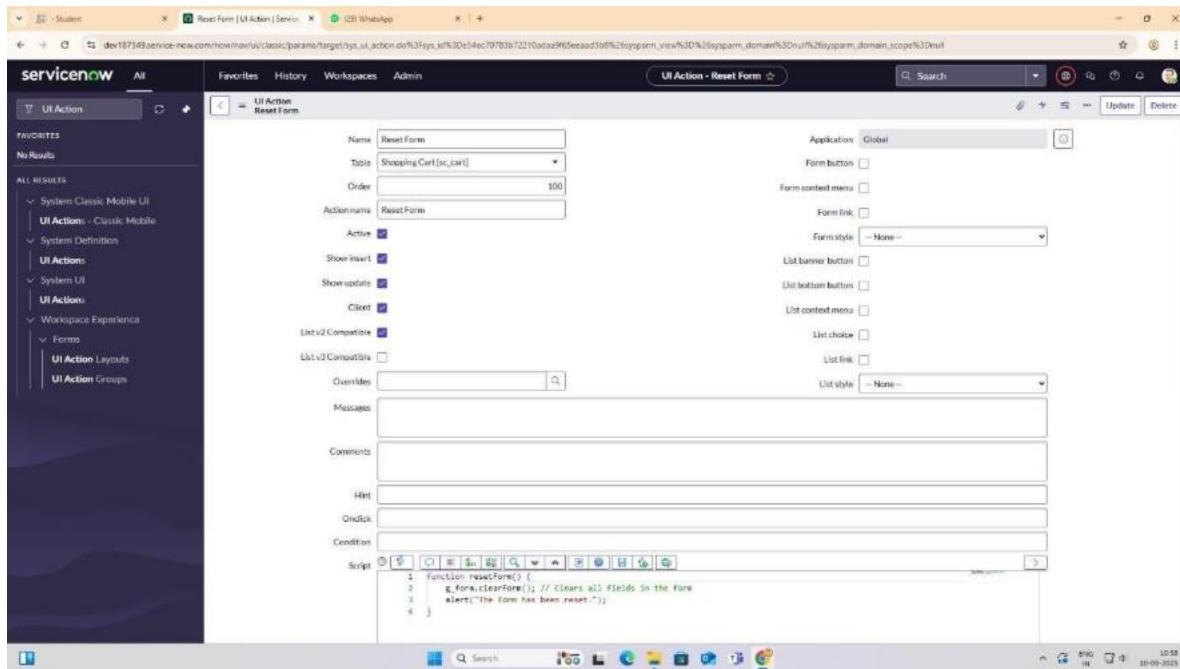
Milestone 4: Create UI Action

Activity 1: Create Reset Form UI Action

- Open ServiceNow.
- Click on All → Search for UI Action.
- Select UI Actions under System Definition.
- Click on New.
- Fill in the following details:
 - Table: shopping_cart (sc_cart)
 - Order: 100
 - Action Name: Reset Form
 - Client: Checked
 - Script:

```
function resetForm() {
  g_form.clearForm(); // Clears all fields in the form
  alert("The form has been reset.");
}
```

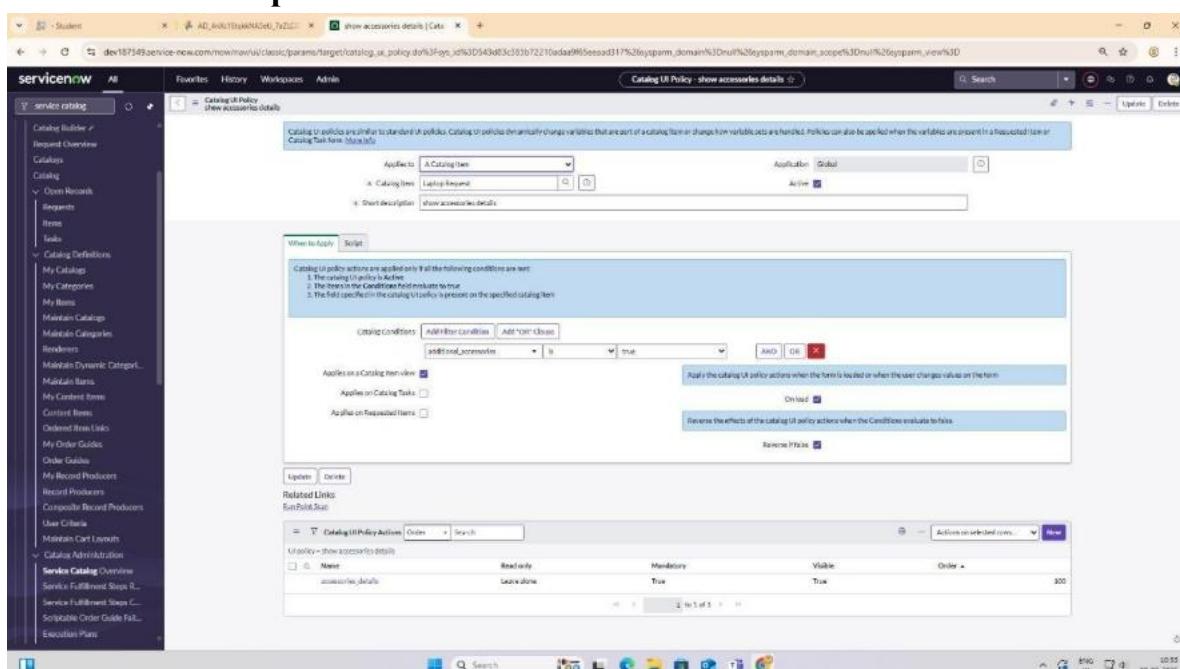
- Click Submit to save the UI Action.



Milestone 5: Exporting Changes to Another Instance

Activity 1: Export Update Set as XML

- Click on All → Search for Update Sets.
- Select Local Update Sets.
- Select the created update set — *Laptop Request Project*.
- Set the State to *Complete*.
- In the *Related List* → *Updates* tab, verify all updates performed under this update set.
- Click on **Export to XML** — it downloads one XML file.



Milestone 6: Retrieving the Update Set

Activity 1: Import Update Set from XML into Another Instance

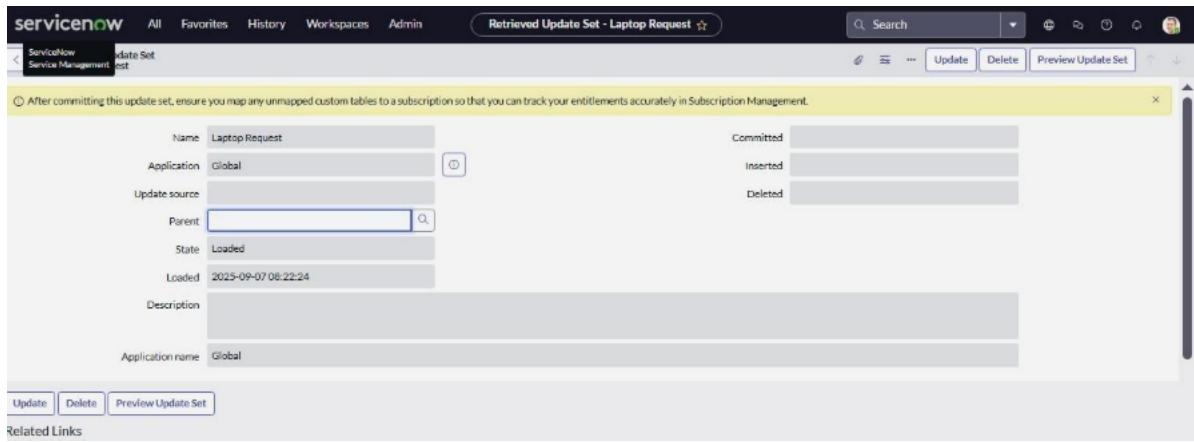
- Open another ServiceNow instance in an *Incognito Window*.
- Log in using valid credentials.
- Click on All → Search for Update Sets.
- Select Retrieved Update Sets under System Update Sets.
- Scroll down and click on Import Update Set from XML.
- Upload the downloaded XML file.

The screenshot shows the ServiceNow web interface with the title bar "servicenow All Favorites History Workspaces Admin". A search bar at the top right contains the text "Retrieved Update Sets". Below the search bar is a navigation menu with "All > C" and a search input field. The main content area displays a table titled "Retrieved Update Sets" with columns: State, Update source, Description, Loaded, Committed, Parent, and Remote Batch Base. The table has several search input fields above it. A large graphic of a satellite dish is centered on the page. At the bottom, a message says "No records to display". On the left sidebar, under "FAVORITES", there is a section for "System Update Sets - Local Up..." with options like "Update Sources", "Retrieved Update Sets", and "Merge Update Sets". A tooltip on the "Retrieved Update Sets" link indicates "Update sets that have been pulled from another ServiceNow instance". Below the sidebar, "Related Links" include "Import Update Set from XML". In the bottom right corner, a clipboard icon shows "24 of 24 - Clipboard" and "Item not Collected: Delete items".

- Click on **Upload**, and it gets imported.
- Open the retrieved update set **Laptop Request Project**.

The screenshot shows the "Import XML" wizard. The title bar says "servicenow All Favorites History Workspaces Admin" and "ServiceNow". The main content area has a heading "Importing records from an XML file will not run Business Rules". It shows "Step 1: Choose file to upload" with a "Choose File" button labeled "sys_remote_u_c5e45aeaxml". Below it is "Step 2: Upload the file" with a blue "Upload" button. A progress bar at the bottom indicates "0%".

- Click on **Preview Update Set**, then click on **Commit Update Set**.
- Review the **Related List Updates**.
- After committing, all updates from the previous instance are now available in this instance.



Milestone 7: Test Catalog Item

Activity 1: Verify Functionality of the Laptop Request Catalog Item

- Search for **Service Catalog** in the *Application Navigator* in the target instance.
- Select **Catalog** under *Service Catalog*.
- Select the **Hardware Category** and search for *Laptop Request*.

Item	Description	Price	Category
iPad pro	Request for iPad pro	₹70,508.3175	Hardware
Laptop Model	Use this item to request a new laptop	+₹2,647.3711 Monthly	Hardware
Laptop request	Use this item to request a new laptop		Hardware
Sales Laptop	Acer Aspire NX	₹97,070.2744	Hardware
Standard Laptop	Lenovo - Carbon x1	+₹8,824.5704 Annually	Hardware
Apple Watch	Apple Watch - Their most personal device ever	₹97,070.2744	Hardware
		+₹8,824.5704 Annually	Hardware
		₹30,683.114	Hardware

- Open the *Laptop Request* item — it shows **three variables** initially.

- As per the scenario, when you click on the **Additional Accessories** checkbox, the **Accessories Details** field becomes visible and mandatory.
- Review the results — the functionality works as expected and fulfills the requirements.

The screenshot shows the ServiceNow Service Catalog interface for a 'Laptop request'. The main form has fields for 'Laptop Model' and 'Justification', with a note above stating 'Use this item to request a new laptop'. Below these is a checked checkbox for 'Additional Accessories', which triggers the visibility of the 'Accessories Details' field. To the right, a sidebar displays an 'Order this Item' section with quantity set to 1, delivery time set to 2 Days, and a large blue 'Order Now' button. There are also links for 'Add to Cart' and 'Shopping Cart'.

Conclusion

The **Laptop Request Catalog Item** project successfully streamlines the process of requesting laptops within the organization by leveraging **ServiceNow's Service Catalog** capabilities. Through the implementation of a dynamic catalog item, the project ensures that users have an **intuitive and user-friendly interface**, reducing errors and improving efficiency.

This project demonstrates how **ServiceNow** can effectively replace **manual, error-prone processes** with **automated, efficient, and user-centric solutions**. It not only improves **service delivery** but also enhances **employee satisfaction** by providing a **modern and streamlined request experience**.

