M.A.M COLLEGE OF ENGINEERING AND TECHNOLOGY

Siruganur, Tiruchirappalli.

Department of

Artificial Intelligence And Data Science

A Project on

LAPTOP REQUEST CATALOG ITEM

NM ID :

TEAM ID : NM2025TMIDO2812

TEAM SIZE : 05

GROUP MEMBER'S NAME WITH REG. NO :

GOKUL R – 812022243015 (Team Leader)

NISHANTH R – 812022243035 (Team Member)

SETHUPATHI T – 812022243052 (Team Member)

SIVASAKTHI S – 812022243055 (Team Member)

SAKTHEESHWARAN M – 812022243048 (Team Member)

MASTeR

M.A.M COLLEGE OF ENGINEERING AND TECHNOLOGY

TRICHY - CHENNAI TRUNK ROAD, SIRUGANUR, TIRUCHIRAPPALLI. PIN: 621 105

MAMCET

Department of Artificial Intelligence and Data Science

Bonafide Certificate

Certified that this is a Bonafide record work done by _____ Register no _____ of Second Year Computer Science in the **MN1051 – ServiceNow Administrator Laboratory** during the academic year **2025-2026**.

Place : Siruganur

Date :

Faculty in Charge **HOD**

Submitted for the University Practical Examination held on …….

INTERNAL EXAMINER **EXTERNAL EXAMINER**

TABLE OF CONTENTS

IDEATION PHASE

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.

Problem Definition

The current laptop request process within the organization is **manual and inefficient**, resulting in frequent delays, incomplete submissions, and miscommunication between employees and the IT department. Employees typically rely on static forms or email-based communication to request laptops, which often lack dynamic behavior to guide the user in providing accurate and complete information. This leads to repetitive clarifications, approval delays, and inconsistencies in data collection.

Additionally, the absence of form automation and governance controls poses challenges in tracking request history, managing configuration changes, and maintaining compliance during deployment. The IT department struggles to ensure that every modification to the process is properly documented and approved. Furthermore, users do not have the option to easily reset or modify their request forms without restarting the entire process, which adds to frustration and inefficiency.

In essence, the existing process fails to meet the growing demand for **speed, transparency, and accuracy** in IT service requests. There is a clear need for an **automated, dynamic, and user-friendly Service Catalog item** that addresses these challenges and enhances the overall employee experience.

Abstract

In many organizations, employees require laptops to perform their daily tasks effectively. However, traditional manual processes for requesting such assets often lead to inefficiencies, errors, and delays in fulfillment. This project aims to design and implement a **dynamic Service Catalog item** within the organization's IT Service Management (ITSM) platform to streamline laptop requests. The solution will replace the manual process with an automated, user-friendly interface that dynamically adjusts form fields based on user inputs, provides clear instructions, and enables users to reset or modify their requests with ease. Additionally, all form changes, submissions, and deployments will be tracked to ensure transparency, compliance, and effective governance. By implementing this solution, the

organization can enhance employee experience, improve request accuracy, and reduce turnaround time in laptop provisioning.

Empathy Map canvas

1. Says

- "I need a laptop quickly to start my project."
- "The current request process takes too long."
- "I'm not sure what details I need to fill in."
- "It would be easier if the form guided me automatically."

2. Thinks

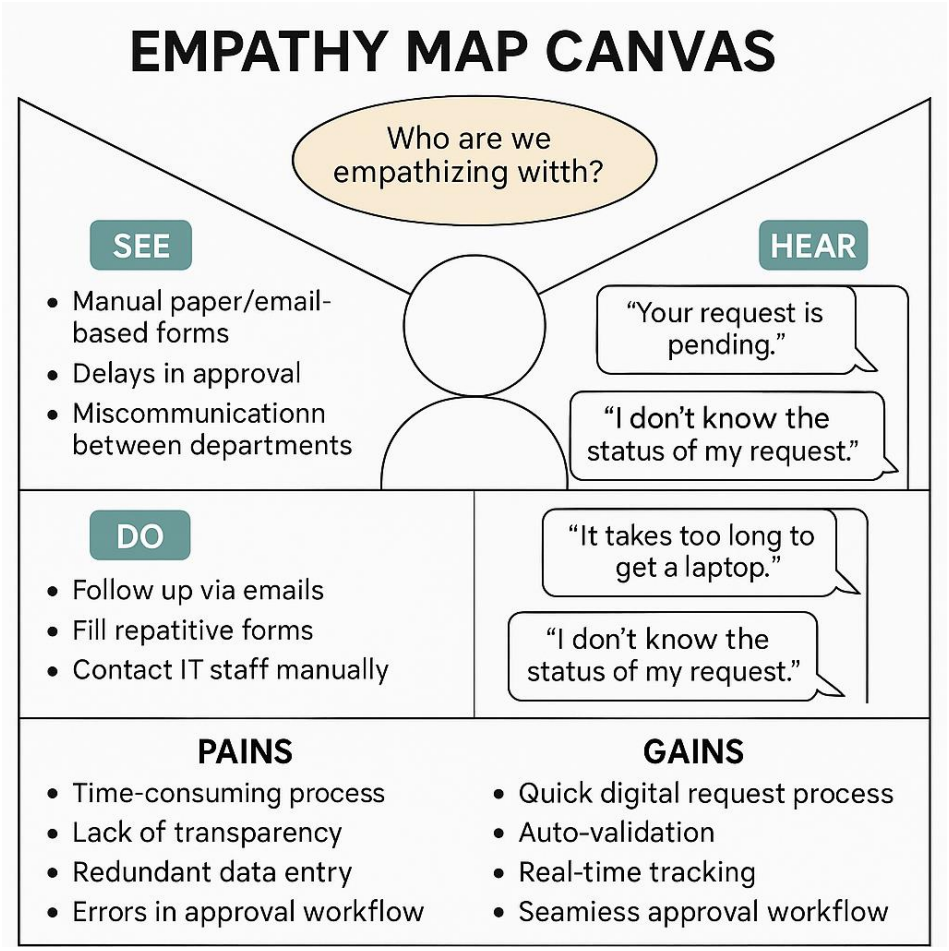
- "Why can't this process be automated?"
- "I hope my request doesn't get delayed again."
- "There should be a simpler way to track my request."
- "It would be nice if I could reset the form and start fresh."

3. Does

- Fills out manual or static forms to request laptops.
- Contacts the IT or HR department for updates.
- Submits repeated requests due to missing or incorrect data.
- Waits for approvals without knowing the request status.

4. Feels

- Frustrated with delays and repetitive manual steps.
- Confused about what information to provide.
- Anxious about approval timelines.
- Relieved when the process is automated and transparent.


EMPATHY MAP CANVAS
Who are we empathizing witth?
SEE
Manual paper/email-based forms
Delays in approval
Miscommunicationn between departments
HEAR
"Your request is pending."
"I don't know the status of my request."
DO
Follow up via emails
Fill repatitive forms
Contact IT staff manually
"It takes too long to get a laptop."
"I don't know the status of my request."
PAINS
Time-consuming process
Lack of transparency
Redundant data entry
Errors in approval workflow
GAINS
Quick digital request process
Auto-validation
Real-time tracking
Seamiess approval workflow

4. **Form Reset and Modification Functionality**
   Enable users to reset or modify form entries before submission, ensuring flexibility and minimizing data entry mistakes.
5. **Governance and Change Tracking**
   Establish a mechanism to log and track all changes made to the catalog item, workflows, and configurations for compliance and auditing purposes.
6. **Seamless Integration and Deployment**
   Ensure that the catalog item integrates smoothly with existing ITSM workflows (e.g., approval, fulfillment, and asset management) and that changes are easily deployable across environments (development, testing, production).
7. **Improved Efficiency and Transparency**
   Reduce request turnaround time, eliminate manual intervention, and provide visibility into the request lifecycle for both employees and administrators.

Scope

The scope of this project encompasses the **design, development, implementation, testing, and deployment** of a dynamic Service Catalog item for laptop requests within the organization's IT Service Management (ITSM) system (e.g., ServiceNow or an equivalent platform).

The solution will focus on automating the end-to-end laptop request process — from submission to approval and fulfillment — ensuring improved accuracy, efficiency, and governance. The project will also include the creation of form logic, dynamic field behavior, and tracking mechanisms for change management.

In-Scope Activities

- Design and development of a dynamic catalog item for laptop requests.
- Implementation of client-side and server-side scripting for dynamic behavior.
- Integration with existing workflows (e.g., approvals, fulfillment tasks, notifications).
- Implementation of data validation, tooltips, and instructional text.
- Development of form reset and modification features.
- Configuration of audit trails and change tracking mechanisms.
- Deployment management through proper governance and version control.
- User training and documentation preparation.

Out of Scope

- Physical laptop procurement or inventory management processes beyond catalog integration.
- Development of unrelated IT assets or service catalog items.
- Integration with third-party procurement systems (unless specified in later phases).

PERFORMANCE & TESTING

Procedure / Implementation Steps:

Phase 1 : Create Local Update set

1. Open service now.
2. Click on All >> search for update sets
3. Select local update sets under system update sets
4. Click on new
5. Fill the following details to create a update set as: “Laptop Request”
6. Click on submit and make current
7. By clicking on the button it activates the update set.

Phase 2 : Create Service Catalog Item

1. Open service now.
2. Click on All >> service catalog
3. Select maintain items under catalog definitions
4. Click on New.
5. Fill 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
6. Click on ‘SAVE’

Phase 3 : Add variables

Step1:

- After saving the catalog item form scroll down and click on variable(related list)
- Click on new and enter the details as below

1. Variable 1:Laptop Model
   Type: Single line text
   Name: laptop_model
   Order:100

- Click on submit
- Again click on new and add Remaining variables in the above process

2. Variable 2:Justification
   Type: Multi line text

Name: justification
Order:200

3. Variable 3:Additional Accessories
   Type: Checkbox
   Name: additional_accessories
   Order:300
4. Variable 4: Accessories Details
   Type: Multi line text
   Name:accessories_details
   Order:400

Step2:

- After adding above variable which are added to newly created catalog item
- Then save the catalog item form

Phase 4 : Create Catalog Ui policies

1. Click on all>> search for service catalog
2. Select maintain item under catalog definition
3. Search for 'laptop request' which is created before
4. Select 'laptop request' and scroll down click on "Catalog Ui policies"
5. In the catalog ui policies related list tab click on new
6. Give short description as: show accessories details
7. Set the Catalog Condition in the related list tab 'when to apply'

   [field: additional_ accessories, operator: is, value: true]

8. Click on **save**.(do not click on submit)
9. Scroll down and select 'catalog ui action'
10. Then click on new button
11. Select variable name as: accessories_details

    Order:100
    Mandatory: True
    Visible : True

12. Click on save and again click save button of the catalog ui policy form

Phase 5 : Create ui action

1. Open service now.
2. Click on All >> search for ui action
3. Select ui actions under system definition

4. Click on new
5. Fill the following details to create ui action

   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.");
}
```

6. Click on save

Phase 6 : Exporting changes to another instances

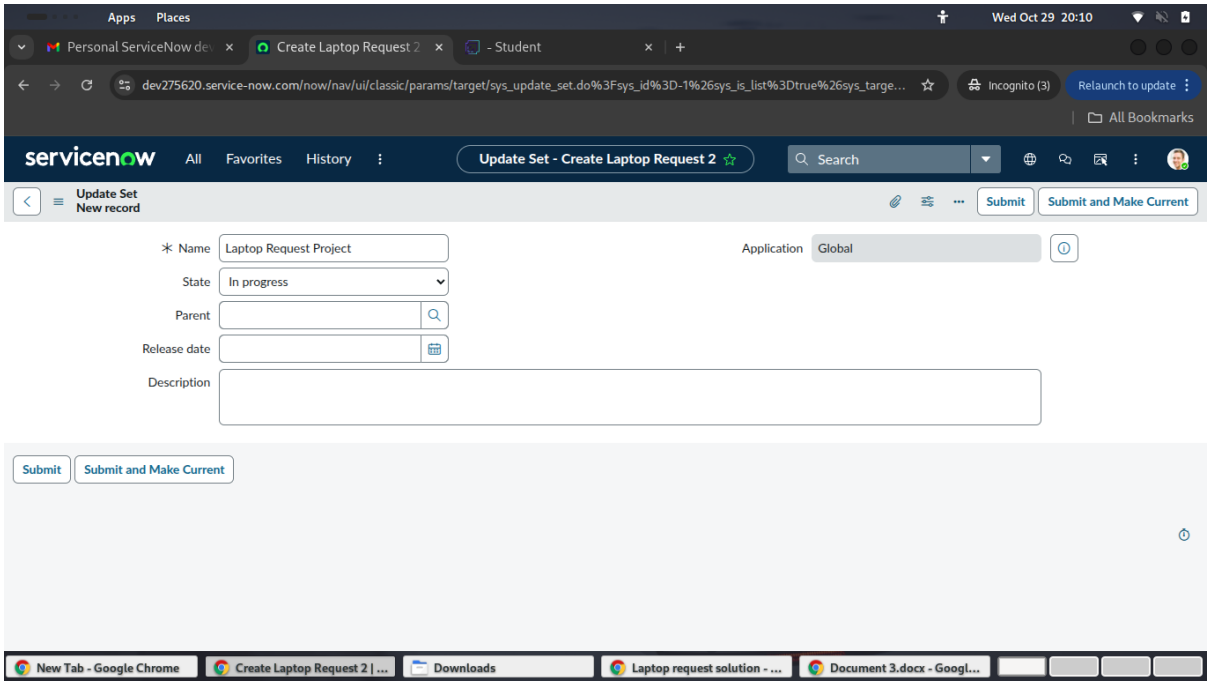
1. Click on All >> search for update sets
2. Select local update set
3. Select created update set i.e. 'Laptop Request Project'
4. Set the state to 'Complete'
5. In the related list Update tab, updates are visible which we perform under this update set.
6. Click on export to XML ,it download one file

Phase 7 : Retrieving the update set

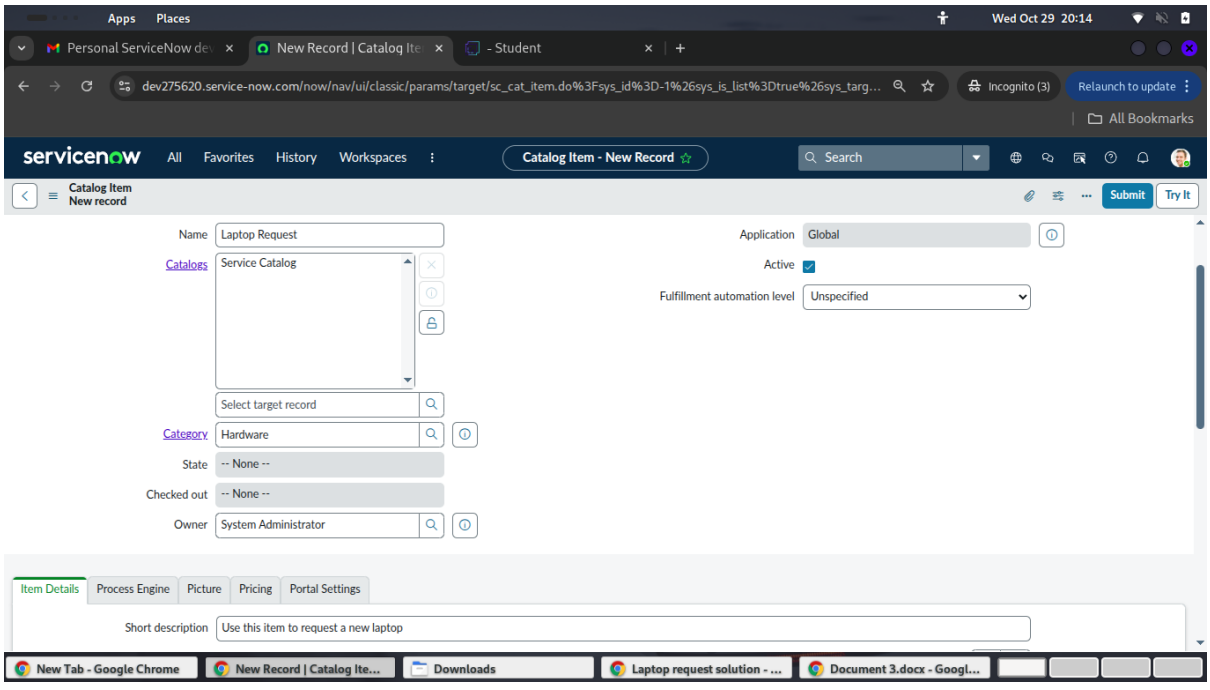
1. Open another instance in incognito window
2. Login with credentials
3. Click on all>> search for update sets
4. Select "Retrieved update set" under system update set
5. It open retrieved update set list and scroll down
6. Click on Import update set from XML
7. Upload the downloaded file in XML file
8. Click on Upload and it gets uploaded.
9. Open retrieved update set 'laptop request project'
10. Click on preview update set
11. And click on commit update set

Screenshots:

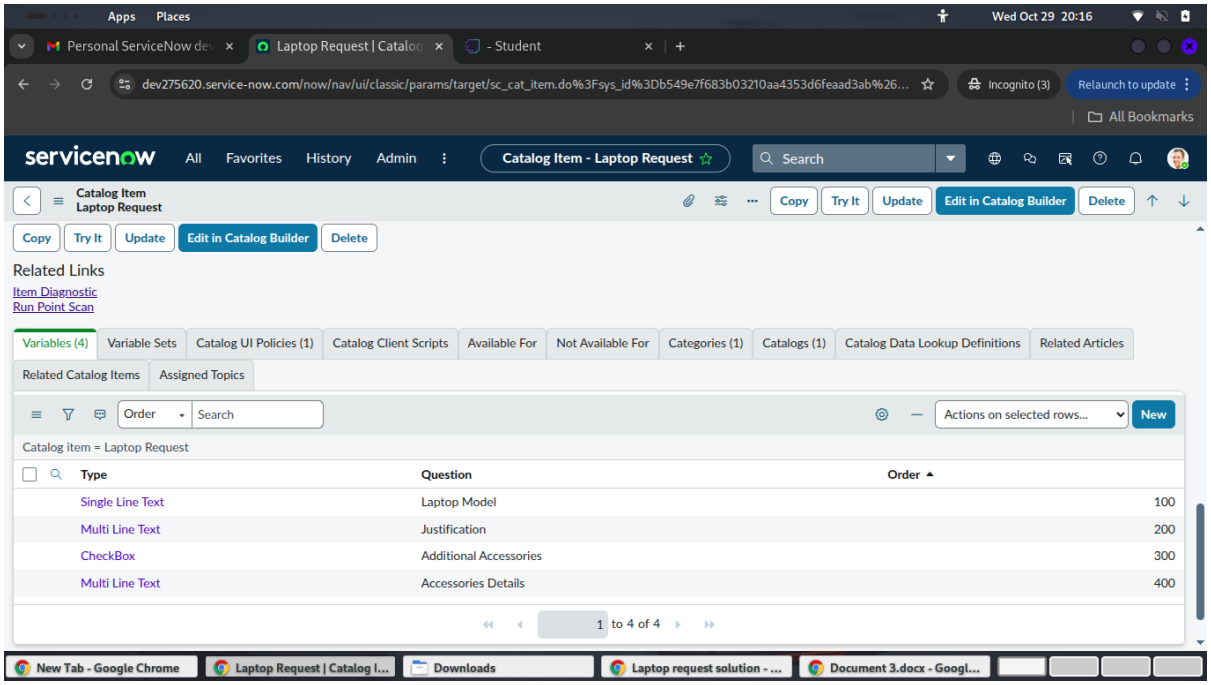
Phase 1 : Create Local Update set



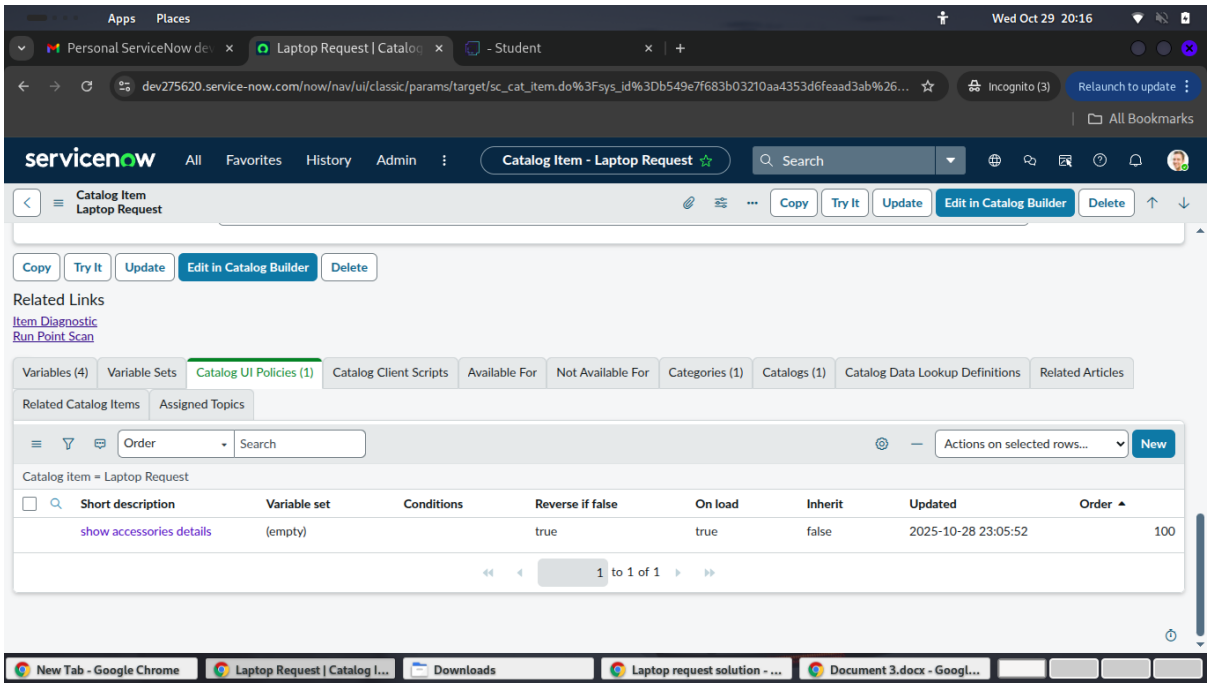
Phase 2 : Create Service Catalog Item



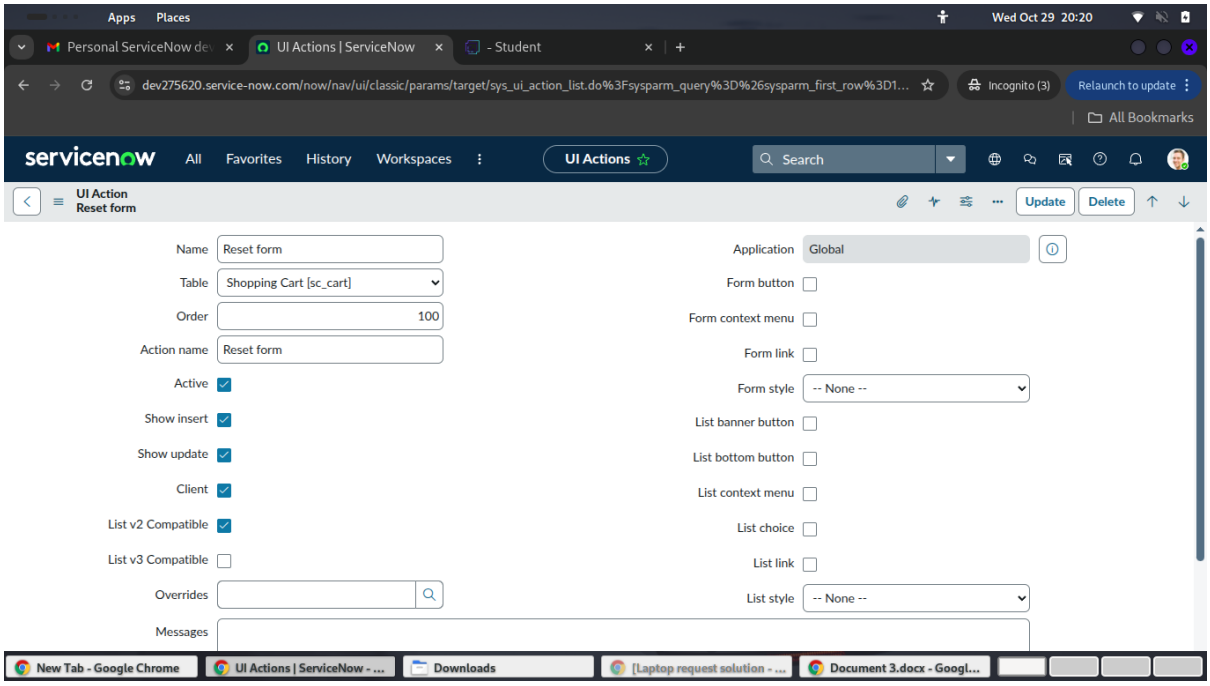
Phase 3 : Add variables



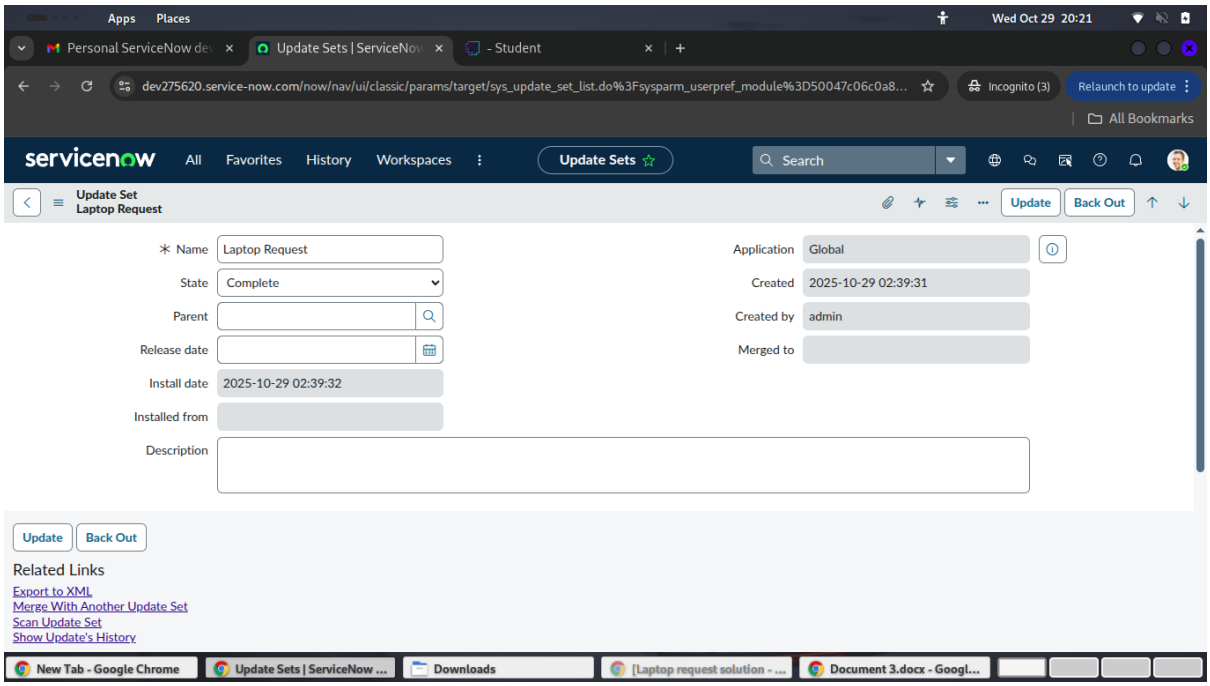
Phase 4 : Create Catalog Ui policies



Phase 5 : Create ui action



Phase 6 : Exporting changes to another instances



PROJECT DESIGN PHASE

Problem – Solution fit

Problem

The organization's current laptop request process is manual, time-consuming, and lacks transparency. Employees must rely on emails or paper forms, causing:

- Delays in approvals and provisioning.
- Missing or incorrect data due to unstructured forms.
- Repeated communication with IT staff.
- No clear visibility into request status or history.

This inefficiency reduces productivity and increases workload for both employees and IT teams.

Proposed Solution

To address these challenges, the project introduces an **automated Service Catalog Item** in the ServiceNow platform that:

- Allows employees to **submit laptop requests digitally** through a self-service portal.
- Includes **dynamic form fields** that adapt based on the employee's department, role, or laptop type.
- Offers a **reset button** to clear the form and start over easily.
- Integrates a **workflow-driven approval process**, automatically routing requests to managers and IT admins.
- Ensures **real-time tracking** and **email notifications** at each stage.
- Maintains **audit logs** to track all changes for compliance and governance.

Parameter	Description
Service Catalog Item Creation	Develop a dedicated catalog item in ServiceNow that allows employees to request laptops through a structured, user-friendly interface.
Dynamic Form Fields	Configure form fields that change dynamically based on user input, such as department, role, or device preference, ensuring accurate and relevant data collection.
UI Policy and Client Scripts	Implement UI Policies and scripts to guide users, enforce mandatory fields, and display context-specific instructions for better usability.
Reset Form Functionality	Add a reset button to allow users to clear all fields and restart the request process easily.
Approval Workflow	Design an automated approval workflow that routes requests to the appropriate manager or IT admin for quick action.
Notification System	Enable automated email notifications to update users about request submission, approval, and provisioning status.
Update Set and Migration	Use Update Sets to capture configuration changes, enabling easy deployment to other ServiceNow instances.
Audit and Tracking	Maintain complete visibility of all changes and requests for governance, compliance, and accountability.
Performance Optimization	Ensure the catalog item loads quickly, uses minimal resources, and delivers a seamless experience for users across devices.

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 be used to 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.

PROJECT PLANNING PHASE

Project Planning Phase

1. Product Backlog

ID	Feature / Task	Description	Priority
PB1	Laptop Request Form	Create a dynamic form to capture laptop request details	High
PB2	Form Reset Option	Implement reset functionality for user convenience	Medium
PB3	UI Policy & Validation	Ensure correct field visibility and data validation	High
PB4	Approval Workflow	Design automated approval routing	High
PB5	Notification System	Send confirmation and approval emails	Medium
PB6	Audit & Tracking Module	Track and log user actions for governance	High
PB7	Deployment Management	Export update sets and migrate configuration	Medium

2. Sprint Planning

Sprint	Duration	Major Goals
Sprint 1	Week 1–2	Requirement gathering and catalog form design
Sprint 2	Week 3–4	Add UI policies, validations, and client scripts
Sprint 3	Week 5–6	Implement workflows and notifications
Sprint 4	Week 7–8	Conduct testing, fix bugs, and prepare deployment

3. User Stories

Story ID	As a...	I want to...	So that I can...
US1	Employee	Request a laptop easily	Get the device without delays
US2	Manager	Approve or reject requests quickly	Maintain workflow efficiency
US3	Admin	Track and audit form usage	Ensure compliance and accuracy
US4	Developer	Reset form and manage validations	Improve usability and data quality

4. Story Points

Story ID	Complexity	Story Points
US1	Medium	5
US2	Medium	4
US3	High	6
US4	Low	3

REQUIREMENT ANALYSIS

1. Solution Requirements

Type	Requirement	Description

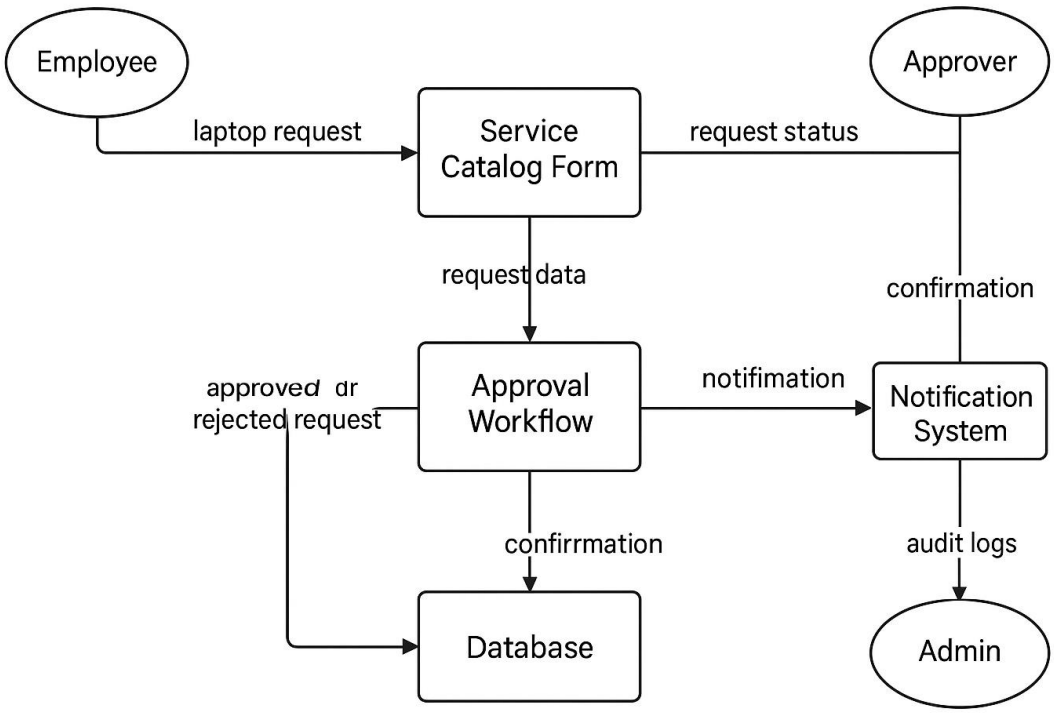
Type	Requirement	Description
Functional	Dynamic Laptop Request Form	Enable users to submit laptop requests with dependent fields (e.g., department → model type).
Functional	Form Reset Function	Allow users to reset all form fields easily.
Functional	Approval Workflow	Route requests automatically to the appropriate approver.
Functional	Notification Alerts	Send automated emails for submission, approval, and rejection.
Functional	Audit Logging	Record all activities for governance and transparency.
Non-Functional	Performance Efficiency	Ensure form submission happens under 2 seconds.
Non-Functional	Data Security	Protect employee and asset data with access control.
Non-Functional	Scalability	Support increasing number of users without performance drop.
Non-Functional	Maintainability	Allow easy configuration updates through update sets.

2. Data Flow Diagram (DFD – Level 1)

Description:
The DFD represents the data movement between the *User*, *System Catalog Module*, *Approval Workflow*, *Database*, and *Notification System*.

Flow Steps:

1. **Employee** submits laptop request via the **Service Catalog Form**.
2. Request data is validated and sent to the **Workflow Engine**.
3. The **Approver** reviews and updates the request status.
4. Approved or rejected data is stored in the **Database**.
5. The **Notification System** sends confirmation emails to the employee.
6. **Admin** monitors through the audit and tracking module.



3. Technology Stack

GENERATIVE AI IN ACTION

Artificial intelligence has evolved through numerous phases, yet it remains both intriguing and inspiring to witness machines become increasingly capable of crafting poetry, humor, and responses that uncannily mimic human creativity. You'll learn about the history of AI, how deep learning plays a pivotal role in generative AI ("gen AI"), and how gen-AI works and is applied to various industries. Throughout this course, you will also learn how to create algorithms, and gain hands-on experience writing code using popular programming languages.

Completed the following required modules to earn an industry-recognized IBM SkillsBuild digital credential called **Generative AI in Action:**

1. Introduction to Generative AI
2. Crafting Precision Prompts with Generative AI
3. Coding Simplified with Generative AI

After completing Generative AI in Action, I was able to:

- Explain how generative artificial intelligence works
- Define what foundation models are and their role in machine learning
- Understand how transformers models are used to solve various language-related tasks
- Describe how prompt engineering improves generative AI models
- Identify common prompt elements
- Explore code generation using a natural language prompt
- Describe prompt engineering techniques
- Perform common programming tasks using Python's built-in functions and libraries
- Create scripts and code for solving real-world problems and automating routine tasks

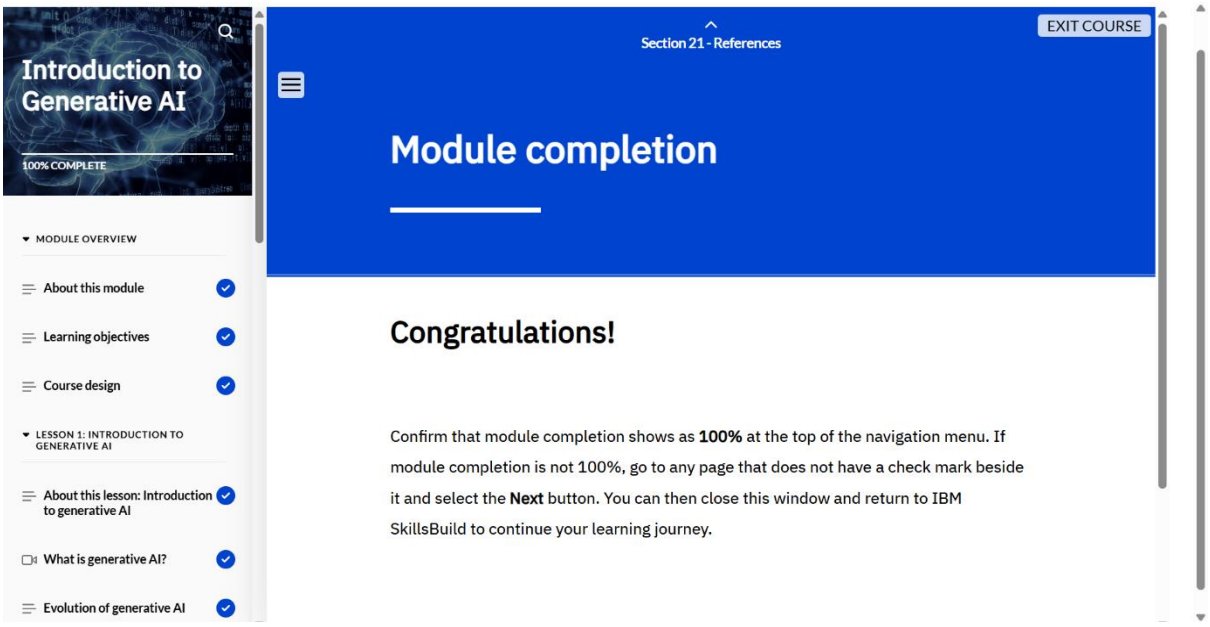
MODULE 1 - Introduction to Generative AI

In this course, I have learned about the history of AI, how deep learning plays a pivotal role in generative AI (gen-AI), and how gen-AI works and is applied to various industries. I also learned how to create algorithms, and gain hands-on experience writing code using popular programming languages.

After completing this module, I was able to:

- Explain how generative artificial intelligence (gen-AI) works
- Define what foundation models are and their role in machine learning

- Understand how transformers models are used to solve various language-related tasks
- Describe how prompt engineering improve generative AI models
- Perform common programming tasks using Python's built-in functions and libraries
- Create scripts and code for solving real-world problems and automating routine tasks



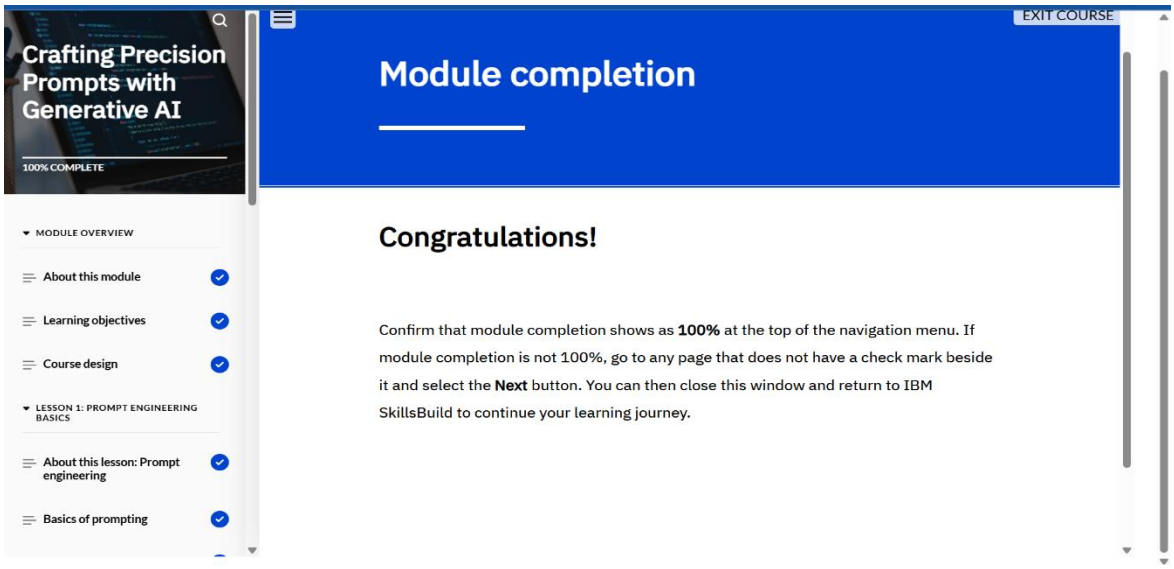
MODULE 2- Crafting Precision Prompts with Generative AI

This was an activity-based course. I learned about AI language models and the rules to follow when giving instructions, or prompting, an AI language model. I walked-through a guided activity that demonstrates how to write effective prompts for an AI language model to help plan a travel itinerary. Finally, I participated in an activity to apply what I learned to effectively write prompts for an AI language model to create my own custom music playlist.

After completing this course, I was able to:

- Describe an AI language model
- Explain how an AI language model understands and responds to humans
- Identify the rules to follow to write effective prompts to generate focused and accurate results from an AI language model
- List the steps to sign up for a ChatGPT account
- Follow the steps to effectively write and refine a series of prompts for ChatGPT for a travel itinerary scenario

- Demonstrate the steps to effectively write and refine a series of prompts for ChatGPT to create a custom music playlist

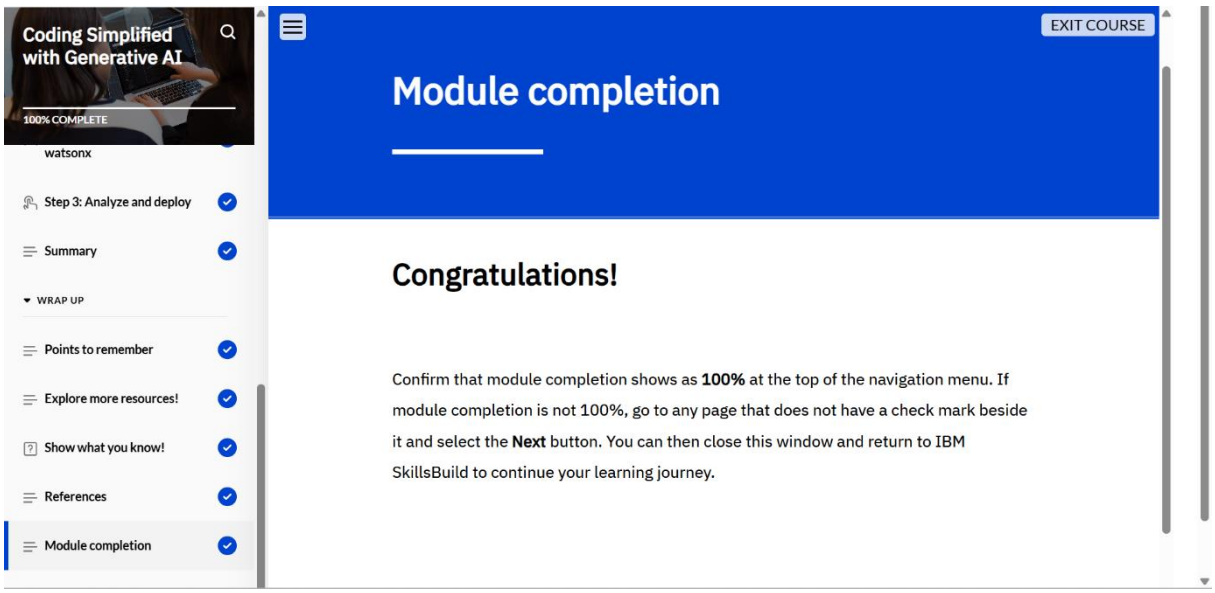


MODULE 3- Coding Simplified with Generative AI

In this course, I had learned the basics of scripting, understand its distinctions from traditional programming, and explore how generative AI models are used to simplify and streamline code generation. Through hands-on labs, I also learned how to create algorithms and apply my skills using widely used programming languages.

After completing this module, I was able to:

- Define scripting and how it is works
- Explain the differences between scripting and traditional programming and when each approach is used
- Describe how Python is used to perform various tasks
- Create a working Python application using IBM watsonx Code Generation



CONCLUSION

The Generative AI in Action program provided a solid foundation in understanding the core principles and practical applications of generative artificial intelligence. Through the three modules—Introduction to Generative AI, Crafting Precision Prompts with Generative AI, and Coding Simplified with Generative AI—I developed a deep understanding of how AI models such as foundation and transformer models function, and how prompt engineering enhances their effectiveness. The hands-on activities improved my ability to craft accurate prompts, automate coding tasks, and generate creative as well as technical outputs using Python and AI-assisted tools. Overall, this program strengthened my knowledge of AI-driven innovation, improved my coding proficiency, and equipped me with essential skills to apply generative AI techniques in real-world scenarios across multiple industries.

WELCOME TO SERVICE NOW

Practice Scenarios for ServiceNow Admin

1.**Create a new user for a contractor, assign them to an "IT Support" group, and ensure they can only access the *Incident* application.**
Solution:

- **Create the Contractor User**
- Navigate to **Users** → *User Administration > Users*.
- Click **New**.
- Fill in details:
    - **User ID**: contractor1
    - **First name / Last name**: Contractor User
    - **Email**: contractor1@gmail.com
    - **Active**: Checked.
- Save.
- **Assign the User to the "IT Support" Group**
- On the user record, scroll to **Groups** (related list).
- Click **Edit**.
- Add to the **IT Support** group.
- Save.
- **Restrict Access to Only the Incident Application**

Now we need to make sure this contractor can only work with **Incident**.

Option A: Role-Based Control (Mostly Preferred)

- By default, Incident application requires **itil** role.
- Instead of giving full **itil** access (which gives too much), do the following:
    - Create a **new custom role**, ex: **incident_contractor**.
    - Assign this role only to permissions needed for Incident (using ACLs).
    - Assign the new role to your contractor user.
    - Do **not** give **itil** or other broad roles.

Option B: Application Menu Restriction

- Go to **System Definition > Application Menus**.
- Open the **Incident** application menu.
- In the **Roles** field, add your custom role (**incident_contractor).**
    - This ensures only users with this role can see the Incident.
- **Verify Access**
- **Impersonate** the contractor user.
- Check:
    - They should only see the **Incident application** in the left nav.
    - They can open/create/edit incidents (based on the ACLs you configured).
    - They cannot access other apps (like Change, Problem, etc.).

2. **Assign a role to a new group so members can read *Knowledge Articles* but cannot create or edit them.**

- ***Create a New Group***
- Navigate to User Administration > Groups.
- Click New.
- Enter a Name for the group (e.g., Knowledge Readers).
- Optionally, add a Description.
- Click Submit.
- ***Assign the Appropriate Role***

To allow read-only access to Knowledge Base articles, assign the **knowledge** role:

- Open the newly created group.
- Scroll to the Roles related list.
- Click Edit.
- Add the role: knowledge
    - This role allows users to view published articles.
- Click Save.

******Do NOT assign roles like **knowledge_admin or knowledge_manager,** which grant create/edit permissions.

- ***Add Users to the Group***
- In the group record, scroll to the Group Members related list.
- Click Edit.
- Select users you want to add.
- Click Save.
- ***Verify Access***
- Log in as one of the group members.
- Navigate to Knowledge > Articles.
- Confirm they can view articles.
- Try creating or editing an article — they should not have access.

3. **Configure a UI Policy that hides the "Work Notes" field unless the state is "In Progress"**. **Solution:**

- **Navigate to UI Policies**
- Go to Application Navigator → type UI Policies → click System UI > UI Policies.
- Create a New UI Policy
- Click New.
- Select the Table → e.g., *Incident* (or whichever table you're working on).
- Provide a Name (e.g., *Hide Work Notes unless In Progress*).
- In the Conditions section, set:
    - Field = *State*
    - Operator = *is*
    - Value = *In Progress*.

- Check the box Active.
- Save the record.
- **Add a UI Policy Action**
- In the same UI Policy record, scroll to UI Policy Actions (Related List).
- Click New.
- Configure the action:
    - Field name = *Work notes*
    - Visible = *True* (since you want it visible only when the condition is met).
- Submit the action

4. **Configure a UI Policy to hide Notes section in incident, when state is In Progress.**

Solution:

- **Navigate to UI Policies**
- Go to Application Navigator → type UI Policies → click System UI > UI Policies.
- Create a New UI Policy
- Click New.
- Select the Table → e.g., *Incident* (or whichever table you're working on).
- Provide a Name (e.g., *Hide Work Notes unless In Progress*).
- In the Conditions section, set:
    - Field = *State*
    - Operator = *is*
    - Value = *In Progress*.
- Check the box Active.
- Save the record.
- **Make Run Script box True**
- Just write one line of code:
- g_form.setSectionDisplay('notes',false);
- Submit the action

5. **Configure a response SLA, the SLA should pause, when the incident state is in On Hold vice versa**.

- ***Create or Modify an SLA Definition***
- Navigate to **Service Level Management > SLA Definitions**.
- Click **New** or open an existing SLA (e.g., "Response SLA").
- Fill in the basic details:
    - **Name**: Response SLA
    - **Table**: **Incident**
    - **Type**: Response
    - **Duration**: Set your desired time (e.g., 1 hour)
- ***Set SLA Conditions***
- Under the **Start Condition**:
    - Example: **State is New**
- Under the **Stop Condition**:

- Filter: Opened At → on or after → Today - 30 days
- ***Incidents Created Within a Week***
- Name: Incidents Created - Last 7 Days
- Source Table: Incident
- Type: Time Series or Bar Chart
- Filter: Opened At → on or after → Today - 7 days
- Group By: Opened At (Daily)
- Aggregation: Count
- ***Incidents by State***
- Name: Incidents by State
- Source Table: Incident
- Type: Bar Chart or Pie Chart
- Group By: State
- Aggregation: Count
- Filter: Opened At → on or after → Today - 30 days

Step 2: Create a Dashboard

- Go to Self-Service > Dashboards.
- Click Create New Dashboard.
- Name: **Service Desk Manager KPIs**
- Add a Proper Description
- Click Submit.

Step 3: Add Reports to the Dashboard

1. Open the newly created dashboard.
2. Click Edit Content.
3. Use Add Reports to include:
    - **Incidents by Priority**
    - **Incidents Created - Last 7 Days**
    - **Incidents by State**
4. Arrange the widgets as needed for clarity.

9. **Restrict the ability to delete records in the *Change Request* table so only users with the "admin" role can do so.**

- **Navigate to Access Control (ACLs)**
- In the **Application Navigator**, type **Access Control**.
- Go to **System Security > Access Control (ACL)**.
- **Create a New ACL Rule**
- Click **New**.
- Fill in details:
    - **Type**: *record*
    - **Operation**: *delete*
    - **Table**: *Change Request [change_request]*

Operation	read
Table	Your HR Case table
Active	True

Step 4: Define Access Condition (No Script)

Scroll down to the Requires role section:

- Add the Role hr_access.

This means only users with the hr_access role can read/view HR Case records.

Step 5: Save and Test

1. Click Submit or Update to save the ACL.
2. Impersonate a non-HR user:
    - Go to your profile → click Impersonate User → choose a user *not in the HR group*.
    - Try opening an HR Case record → You should see a "Security constraints prevent access to requested page" message.
3. Now impersonate an HR group member:
    - They should be able to open HR Cases normally

13. When the Incident state changes to In Progress, Child incident related list should be hidden.

Solution:

1. Navigate to System UI → UI Policies → New.
2. Fill the header:
    - Name: Hide related lists when State is In Progress
    - Table: Incident
    - Active: checked
    - Global: checked
3. Condition: **State is In Progress**
   (Use the exact label used in your instance for the In Progress state.)
4. Submit the UI Policy record.
5. In the UI Policy record click **New** under **UI Policy Actions**.

Set:

- **Field name:** select the related list–Child incident
- **Visible:** false
- **Read only:** optional
- Save and Test the UI Policy Action.

14.How to Display Incident number while loading the incident form

Solution:

1. Navigate to System UI → Client Scripts → New.
2. Fill the header:
    - Name: Show Incident Number on Load
    - Table: Incident
    - Type: onLoad
    - Active: True
3. Add this script:

```
function onLoad() {

  // Get the Incident number field value

  var incNum = g_form.getValue('number'); // 'number' is the field name

  alert('Incident Number: ' + incNum);

}
```

15. When the Incident state changes to In Progress, description should be hidden and short description should be mandatory.

Solution:

Step 1 — Navigate to Client Scripts

1. Go to:
   System UI → Client Scripts → New
2. Fill the header:
    - Name: Hide Description and Make Short Description Mandatory
    - Table: Incident
    - Type: onChange
    - Field name: state
    - Active: checked

Step 2 — Add the Client Script Code

```
function onChange(control, oldValue, newValue, isLoading) {

    if (isLoading) return;
```

```
    if (newValue === '2') {
        g_form.setDisplay('description', false);
        g_form.setMandatory('short_description', true);
    } else {
         g_form.setDisplay('description', true);
        g_form.setMandatory('short_description', false);
    }
}
```

- Click **Submit** or **Update** to save.

15. If the description field is empty in the incident table, prevent the form submission.

Solution:

Step 1 — Navigate to Client Scripts

1. Go to:
   System UI → Client Scripts → New
2. Fill the header:
   - Name: Prevent Submit if Description Empty
   - Table: Incident
   - Type: onSubmit
   - Active: checked

Step 2 — Add the Client Script Code

```
function onSubmit() {
 var description = g_form.getValue('description');
    if (description == '') {
   g_form.addErrorMessage('Description cannot be empty');
        return false;
    } else {
    return true;
}
}
```

16. Users can not change the state field values in the incident list.

Solution:

Step 1 — Navigate to Client Scripts

3. Go to:
   System UI → Client Scripts → New
4. Fill the header:
   - Name: Prevent State Inline Edit
   - Table: Incident
   - Type: onCellEdit
   - Field name: state
   - Active: checked

Step 2 — Add the Client Script Code

```
if(newValue==2){

alert('You can not edit this value');

saveAndClose==false;

}else{

saveAndClose==true;

}
```

17. How to set the Caller to Logged in user automatically in the incident table.

Solution:

1. Navigate: System Definition → Business Rules → New
2. Fill the details:
   - Name: Set Caller on Incident Create
   - Table: Incident
   - When: before
   - Insert/update: checked
   - Advanced: true
3. **Script**:

```
current.caller_id = gs.getUserID();
```

18.When a user updates an incident record, priority should change to Critical automatically.

Solution:

1. Navigate: System Definition → Business Rules → New

2. Settings:
    - Name: Set Priority field
    - Table: Incident
    - When: before
    - Update:checked
3. Script:

```
current.impact = 1;

current.urgency = 1;
```

19.Create a button on the Incident form that allows users to mark an Incident as Resolved with a single click.

Solution:

1. Navigate: System UI → UI Actions → New
2. Settings:
    - Name: Resolve Incident
    - Table: Incident
    - Action type: Form button
    - Active: checked
3. Script:
    - current.state = 6;
    - current.update();
    - action.setRedirectURL(current);

20.Create a button on the incident table that copies the Short Description value into the Description field.

Solution:

1. Navigate: System UI → UI Actions → New
2. Settings:
    - Name: Copy Short Description
    - Table: Incident
    - Action type: Form button
    - Active: checked

3. Script:
    - current.description = current.short_description;
    - current.update();
      action.setRedirectURL(current);