M.A.M COLLEGE OF ENGINEERING AND TECHNOLOGY

Siruganur, Tiruchirappalli.

Department of

Artificial Intelligence and Data Science

A Project on

STREAMLINING TICKET ASSIGNMENT FOR EFFICIENT SUPPORT OPERATIONS

NM ID :

TEAM ID : NM2025TMID02372

TEAM SIZE : 05

GROUP MEMBER'S NAME WITH REG. NO :

DHARANITHARAN S – 812022243012 (Team Leader)

ADHITHYA S P – 812022243301 (Team Member)

VENKATAKRISHNAN S – 812022243308 (Team Member)

RONALDO G– 812022243041 (Team Member)

ROSHAN S– 8120222433702 (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

The objective of this initiative is to implement an automated system for ticket routing at ABC Corporation, aimed at improving operational efficiency by accurately assigning support tickets to the appropriate teams. This solution aims to reduce delays in issue resolution, enhance customer satisfaction, and optimize resource utilization within the support department.

Problem Definition

The current manual ticket routing process at ABC Corporation presents several operational challenges. Due to the reliance on human intervention, tickets are often misclassified or delayed before reaching the appropriate support team. This inefficiency not only slows down issue resolution but also contributes to an uneven workload distribution and decreased productivity across teams. Furthermore, inconsistent ticket handling leads to lower customer satisfaction and an increased backlog of unresolved issues.

Therefore, there is a need for an **automated ticket routing system** capable of accurately classifying support tickets and assigning them to the appropriate teams in real-time. The system should minimize manual dependency, reduce routing errors, and enhance overall operational efficiency, enabling ABC Corporation to deliver faster and more reliable customer support services.

Abstract

The rapid growth of support requests in organizations often leads to inefficiencies in manual ticket management processes. At ABC Corporation, the current ticket routing method relies heavily on human intervention, resulting in delays, misclassification, and uneven workload distribution across support teams. This project aims to develop an **automated ticket routing system** that utilizes intelligent algorithms to categorize and assign support tickets to the appropriate teams with minimal human input. By automating the routing process, the solution seeks to improve operational efficiency, minimize response times, enhance customer satisfaction, and ensure optimal utilization of technical resources.

Empathy Map canvas

Section	Details
Says	“It takes too long for tickets to reach the right team.” “We often get tickets that don’t belong to our department.”
Thinks	“If ticket routing was automated, we could respond faster.” “Manual routing leads to confusion and missed deadlines.”
Does	Reassigns tickets manually, follows up on misrouted issues, tracks ticket queues frequently.
Feels	Frustrated by delays, stressed by workload, eager for a more efficient and automated process.

EMPATHY MAP CANVAS

SAYS

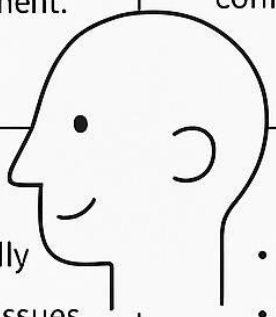
"It takes too long for tickets to reach the right team."

"We often get tickets that don't belong to our deprtment."

THINKS

"I'f ticket routing was automated, we could respond faster."

"Manual routing leads to confusion and missed deadines."



DOES

- Reassigns tickets manually
- Follows up on misrouted issues
- Tracks ticket queues frequently

FEELS

- Frustrated by delays
- Stressed by workload
- Eager for a more efficient and automated process

INSIGHT

Users need a smart, automated system that ensures tickets reach the right group or agent based on category, priority, and skiliset — minimizing human effort and errors.

Introduction

In modern enterprises, providing timely and effective customer support is a critical component of organizational success. However, as businesses scale, the volume of support requests increases exponentially, creating challenges in managing and resolving issues efficiently. ABC Corporation currently employs a manual ticket routing process in which support personnel review, categorize, and assign incoming tickets to relevant teams. This approach is not only time-consuming but also prone to human error and inconsistency, leading to delays in issue resolution and reduced customer satisfaction.

To address these challenges, this project proposes the implementation of an automated ticket routing system. The system will leverage rule-based logic or machine learning techniques to analyze ticket content, identify its nature, and automatically direct it to the most suitable team. By streamlining the routing process, ABC Corporation can significantly improve service delivery, reduce operational costs, and ensure faster resolution times, ultimately enhancing overall customer experience.

Objectives

The main objective of this project is to design and implement an **automated ticket routing system** that enhances the efficiency and accuracy of support operations at ABC Corporation. The specific objectives include:

1. **Automate Ticket Classification:** Develop a system capable of automatically analyzing and categorizing incoming support tickets based on their content, type, and priority.
2. **Intelligent Ticket Assignment:** Implement algorithms or predefined rules to automatically assign tickets to the most appropriate support team or personnel.
3. **Reduce Resolution Time:** Minimize delays caused by manual ticket triage, thereby improving overall response and resolution times.
4. **Enhance Customer Satisfaction:** Ensure that customer issues are handled quickly and accurately, leading to improved user experience and satisfaction.
5. **Optimize Resource Utilization:** Distribute workload evenly among support teams to maximize productivity and prevent bottlenecks.
6. **Enable Data-Driven Insights:** Generate analytical reports on ticket flow, team performance, and issue trends to support continuous improvement in the support process.

PERFORMANCE & TESTING

Procedure or Implementation steps

Phase 1 : Create Users

1. Open service now.
2. Click on All >> search for users
3. Select Users under system security
4. Click on new
5. Fill the following details to create a new user
6. Click on submit Cre-
   ate one more user:
7. Create another user with the following details
8. Click on submit

Phase 2 : Create Groups

1. Open service now.
2. Click on All >> search for groups
3. Select groups under system security
4. Click on new
5. Fill the following details to create a new group
6. Click on submit

Create one more group:

1. Create another group with the following details
2. Click on submit

Phase 3 : Create Roles

1. Create one more role:
2. Create another role with the following details
3. Click on submit

Phase 4 : Create Table

1. Open service now.
2. Click on All >> search for tables
3. Select tables under system definition
4. Click on new
5. Fill the following details to create a new table

Label : Operations related

Check the boxes Create module & Create mobile module

6. Under new menu name : Operations related
7. Under table columns give the columns

Phase 5 : Assign roles & users to certificate group

1. Open service now.
2. Click on All >> search for tables
3. Select tables under system definition
4. Select the certificates group
5. Under group members
6. Click on edit
7. Select Katherine Pierce and save
8. Click on roles
9. Select Certification_role and save

Phase 6 : Assign roles & users to platform group

1. Open service now.
2. Click on All >> search for tables
3. Select tables under system definition
4. Select the platform group
5. Under group members
6. Click on edit
7. Select Manne Niranjan and save
8. Click on roles
9. Select Platform_role and save

Phase 7 : Assign role to table

1. Open service now.
2. Click on All >> search for tables
3. Select operations related table
4. Click on the Application Access
5. Click on u_operations_related read operation
6. Click on the profile on top right side
7. Click on elevate role
8. Click on security admin and click on update
9. Under Requires role

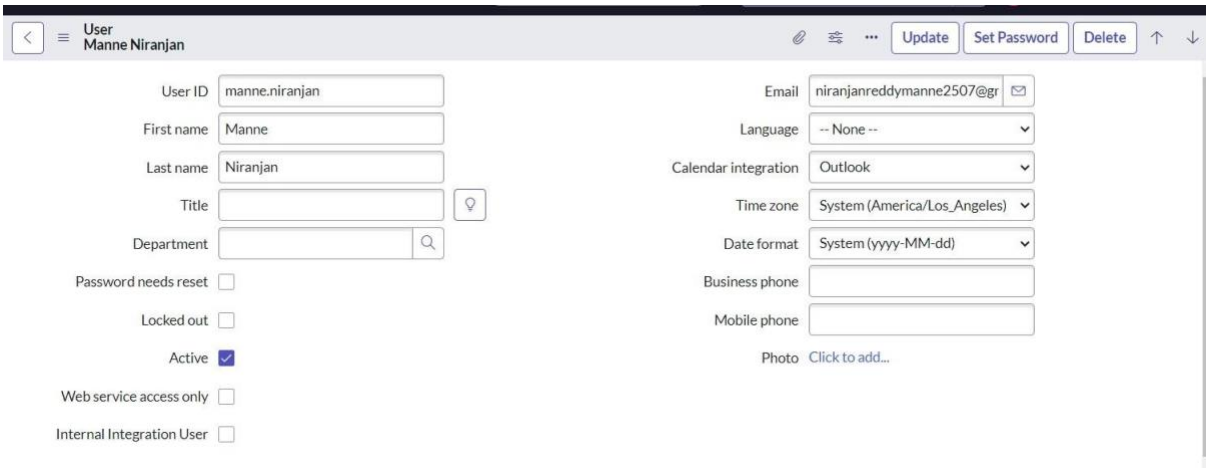
Operator : is

Value : Regrading User expired

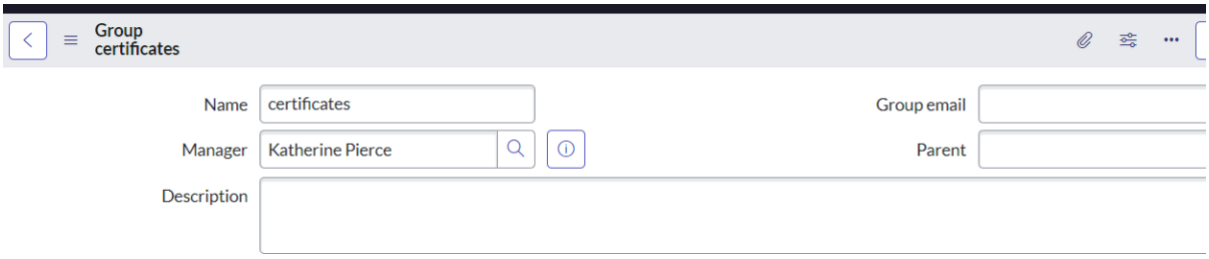
7. After that click on Done.
8. Now under Actions.
9. Click on Add an action.
10. Select action in that search for “ Update Record ”.
11. In Record field drag the fields from the data navigation from left side
12. Table will be auto assigned after that
13. Give the field as “ Assigned to group ”.
14. Give value as “ Platform ”.
15. Click on Done.
16. Click on Save to save the Flow.
17. Click on Activate.

Screenshots:

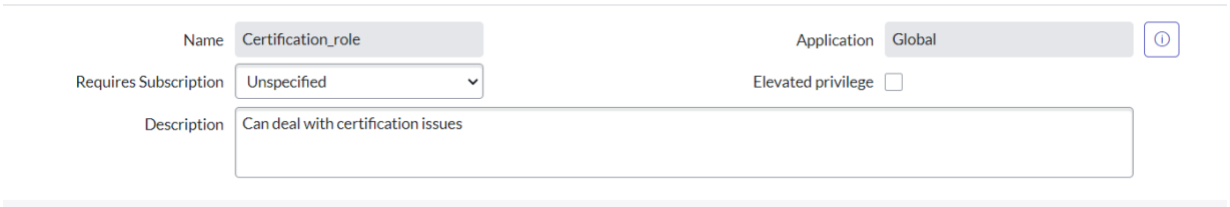
Phase 1 : Create users



Phase 2 : Create Groups



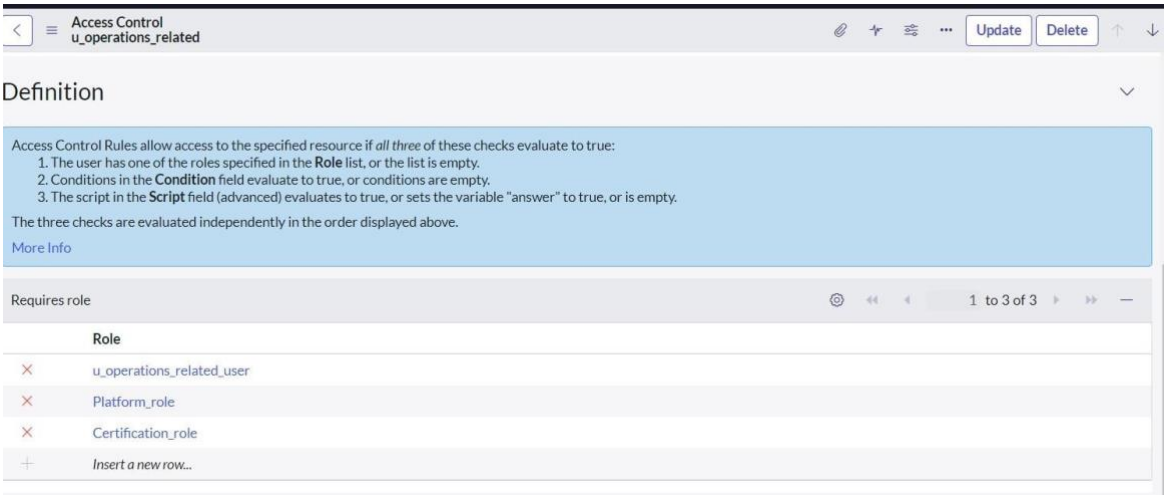
Phase 3 : Create roles



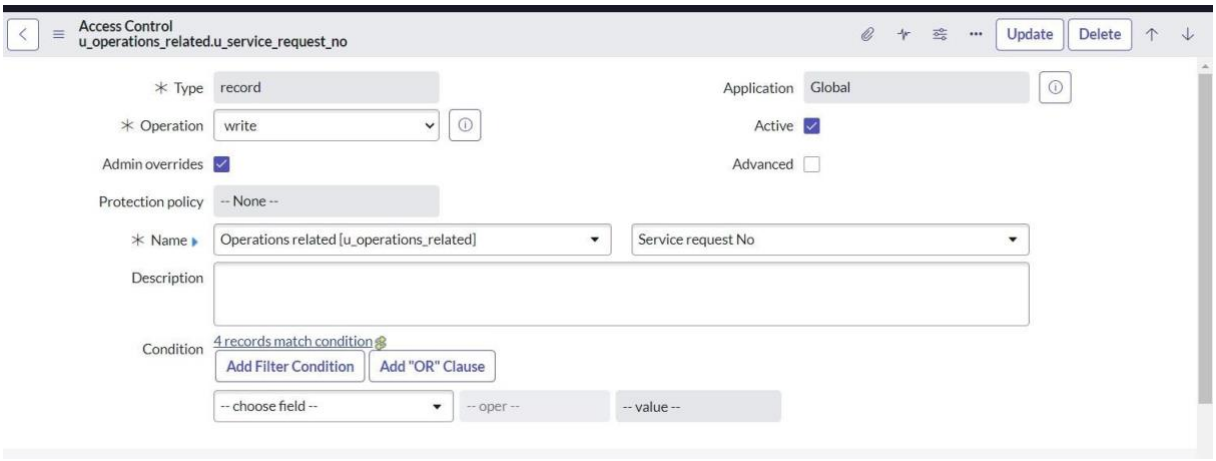
Phase 4 : Create tables

Column label	Type	Reference	Max length	Default value	Display
Created by	String	(empty)	40		false
Created	Date/Time	(empty)	40		false
Sys ID	Sys ID (GUID)	(empty)	32		false
Updates	Integer	(empty)	40		false
Updated by	String	(empty)	40		false
Updated	Date/Time	(empty)	40		false
Assigned to group	Reference	Group	40		false
Assigned to user	Reference	User	32		false
Comment	String	(empty)	40		false
Issue	String	(empty)	40		false
Name	String	(empty)	40		false
Priority	String	(empty)	40		false
Service request No	String	(empty)	40	javascript:getNextObjNumberPadded();	false
Ticket raised Date	Date/Time	(empty)	40		false
Insert a new row...					

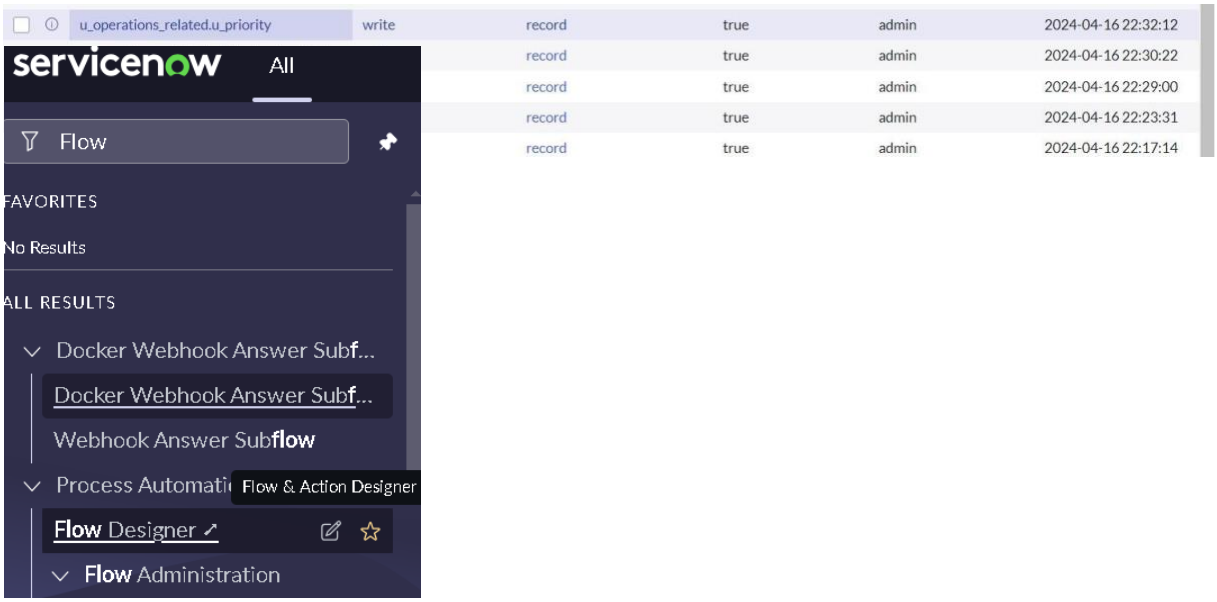
Phase 7 : Assign role to table



Phase 8 : Create ACL



Phase 9 : Create a Flow to Assign operations ticket to group



PROJECT DESIGN PHASE

Problem – Solution fit

The existing ticket assignment process at ABC Corporation relies heavily on manual routing, resulting in frequent delays, misassignments, and uneven workload distribution among support teams. This not only increases response and resolution times but also impacts customer satisfaction and overall team efficiency. The need for an automated and intelligent system to handle ticket classification and assignment is essential to minimize human intervention and errors. The proposed solution directly addresses these pain points by implementing a rule-based and machine learning–driven approach to ticket routing, ensuring that every support request is instantly categorized and assigned to the most suitable team or agent. This alignment between the problem and the solution ensures improved operational efficiency, faster resolution times, and a better customer experience.

Proposed Solution

The proposed solution involves developing an Automated Ticket Routing System within the ServiceNow platform, integrated with AI-based classification mechanisms. The system will analyze ticket details such as issue type, priority, and department using predefined business rules and, where applicable, intelligent models trained on historical data. Once analyzed, the ticket will be automatically assigned to the most relevant team or agent, ensuring accuracy and balance in workload distribution. The platform will also include dashboards for monitoring performance, analytics for route optimization, and feedback loops for continuous improvement. Additionally, audit tracking and notifications will be incorporated to ensure transparency, accountability, and seamless communication between users and support staff**.**

Parameter	Description
Objective	To automate ticket routing and assignment to the appropriate support teams, reducing manual effort and improving efficiency.
Platform	Implemented within the **ServiceNow** platform using its workflow automation and integration capabilities.
Core Functionality	Automatically classifies incoming tickets based on issue category, urgency, and department using predefined rules or AI models.
Automation Engine	Utilizes **business rules**, **flow designer**, and **machine learning models** to route tickets intelligently and accurately.
User Interface	A user-friendly portal for employees to raise support tickets, track status, and receive notifications.
Approval & Escalation	Includes automated approval flows and escalation paths to ensure quick response and issue resolution.
Monitoring & Analytics	Provides real-time dashboards to track ticket distribution, agent performance, and service-level compliance.
Audit & Governance	Maintains logs of all ticket assignments and routing changes for compliance and traceability.
Notifications	Sends automated updates to users and support teams for ticket creation, assignment, and resolution.
Expected Outcome	Enhanced operational efficiency, reduced ticket handling time, improved customer satisfaction, and optimized team workload.

Conclusion

The implementation of the automated ticket routing system at ABC Corporation has been a significant success. By leveraging the capabilities of ServiceNow, we have streamlined the process of assigning support tickets to the appropriate teams, addressing the challenges of manual routing, and ensuring timely resolution of issue.

Solution Architecture

1. Goals of the Architecture

- To automate ticket classification and assignment to appropriate support teams.
- To ensure faster ticket resolution and improved customer satisfaction.
- To provide real-time tracking, reporting, and SLA management.
- To enhance transparency, accountability, and governance in ticket management.
- To ensure scalability, security, and maintainability within the organization's IT infrastructure.

2. Key Components

- Service Portal: Interface for users to create and track support tickets.
- Ticket Management Module: Manages ticket data including priority, category, and status.
- Routing Engine: Core system component that automatically assigns tickets using rules or AI models.
- Flow Designer / Business Rules: Automates workflows and ticket assignment logic.
- Notification Engine: Sends alerts for ticket creation, assignment, and status updates.
- SLA Management System: Monitors and enforces resolution timelines.
- Reports & Dashboards: Displays metrics on ticket handling and team performance.
- Audit & Governance Layer: Logs routing and workflow activities for compliance.
- Database Layer: Stores user, ticket, and configuration data securely.

3. Development Phase

- Requirement Analysis: Identify routing rules, user roles, and automation needs.
- System Design: Plan architecture, workflows, and data flow structure.
- Development: Configure modules, scripts, and automated flows in ServiceNow.
- Testing & Validation: Verify routing accuracy, notifications, and SLA tracking.
- Deployment: Move configurations to the production environment.
- Monitoring & Optimization: Continuously analyze performance and refine rules.

Solution Architecture Description

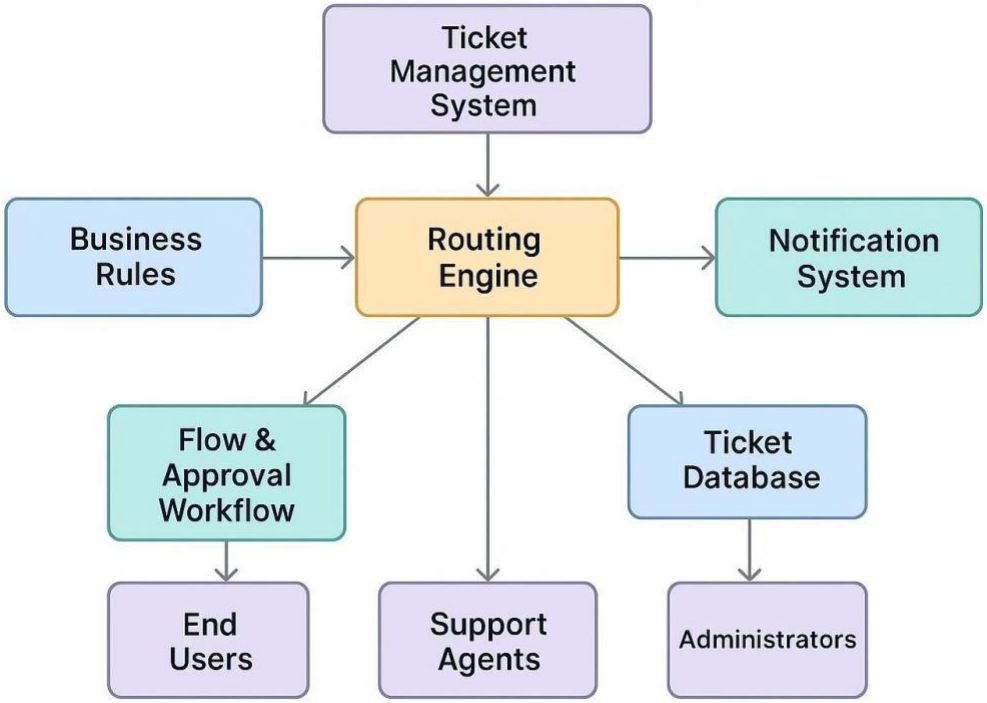
The **Solution Architecture** for the Automated Ticket Routing System is designed to streamline the support process by integrating user interaction, workflow automation, and backend data management into a single cohesive platform. When a user submits a support ticket through the **Service Portal**, the request is immediately processed by the **Routing Engine**, which applies predefined business rules or AI-based logic to determine the most suitable support team or agent.

The **Flow Designer** automates the entire workflow—from ticket creation and categorization to assignment and notification—ensuring a seamless and error-free routing process. The **Notification Engine** sends instant alerts to both users and support teams, keeping all stakeholders informed about ticket status and progress. Meanwhile, the **SLA Management System** continuously monitors response and resolution times to ensure adherence to organizational service standards.

All ticket data, user information, and system configurations are securely stored in the **Database Layer**, ensuring data integrity and accessibility. The **Reports and Dashboards** component provides real-time insights into ticket metrics, team performance, and SLA compliance, helping managers make data-driven decisions. Additionally, the **Audit and Governance Layer** ensures every routing action and configuration change is logged for transparency and compliance.

This architecture provides a scalable, efficient, and automated environment that reduces manual workload, improves ticket resolution speed, enhances user satisfaction, and ensures accountability across the support process.

Solution Architecture Diagram


Ticket Management System
Business Rules
Routing Engine
Notification System
Flow & Approval Workflow
Ticket Database
End Users
Support Agents
Administrators

PROJECT PLANNING PHASE

Project Planning Phase

1. Product Backlog

ID	Feature	Description	Priority
PB1	User & Role Setup	Create and configure users, groups, and roles	High
PB2	Table Configuration	Build ticket table with necessary fields	High
PB3	Automated Routing Flow	Design flow for ticket assignment	High
PB4	SLA Tracking	Integrate SLA for timely response	Medium
PB5	Notification Rules	Configure alerts for status updates	Medium
PB6	Reporting Dashboard	Create visual performance dashboard	Low

2. Sprint Planning

Sprint	Duration	Major Goals
Sprint 1	Week 1–2	Define roles, users, and groups
Sprint 2	Week 3–4	Configure tables and ACLs
Sprint 3	Week 5–6	Build flow and automate ticket routing
Sprint 4	Week 7–8	Add SLA, notifications, and dashboard

3. User Stories

Story ID	As a...	I want to...	So that I can...
US1	End User	Submit a ticket easily	Get faster issue resolution
US2	Support Agent	Receive tickets relevant to my expertise	Respond quickly and effectively
US3	Manager	Track SLA and performance	Ensure timely resolution
US4	Admin	Automate ticket assignment	Reduce manual errors and workload

4. Story Points

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

REQUIREMENT ANALYSIS

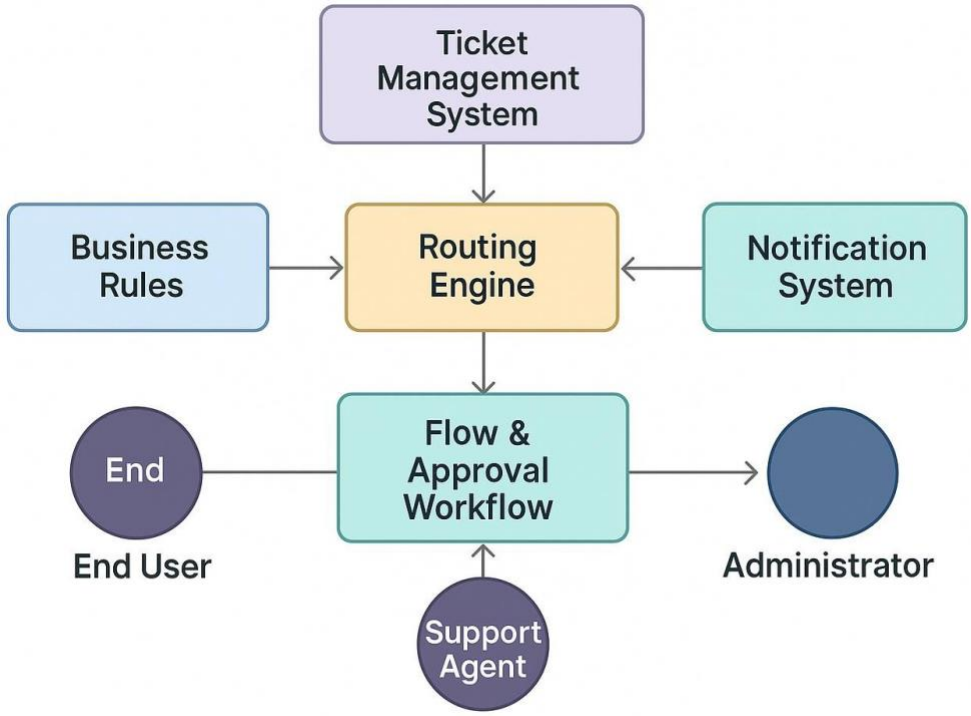
1. Solution Requirements

Type	Requirement	Description
Functional	Automated Ticket Routing	Automatically assign tickets to correct teams based on category or keywords.
Functional	User & Role Management	Define users, roles, and groups for ticket access and assignment.
Functional	SLA Tracking	Monitor ticket response and resolution time.
Functional	Notifications	Send alerts when a ticket is assigned, escalated, or breached.
Functional	Dashboard & Reports	Visualize ticket distribution and team performance.
Non-Functional	Performance	Routing should happen within 2 seconds after ticket creation.
Non-Functional	Security	Restrict ticket visibility based on roles and permissions.
Non-Functional	Scalability	Support growing number of tickets and users.
Non-Functional	Reliability	Ensure consistent routing accuracy and audit tracking.

2. Data Flow Diagram (DFD – Level 1)

Flow Description:

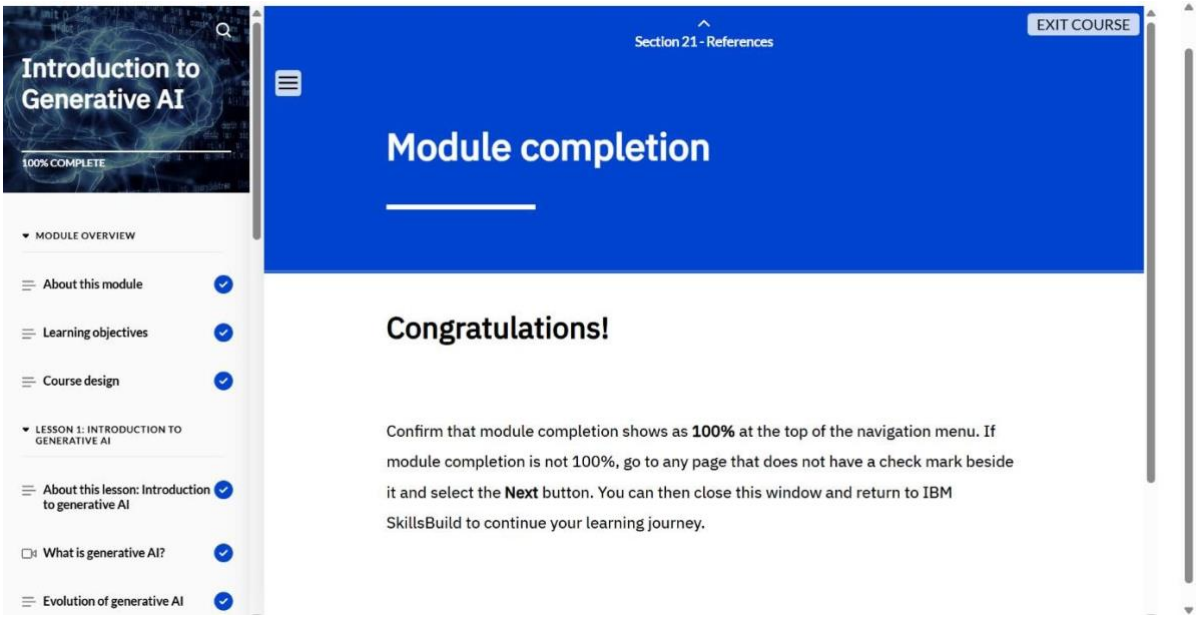
1. **User** submits or logs a ticket via the Service Portal.
2. The **System** captures the details (category, priority, issue type).
3. The **Routing Engine** processes rules and assigns the ticket to the correct **Group** or **Agent**.
4. The assigned **Team** receives a notification and updates the ticket status.
5. Ticket information is stored in the **Database** and reflected in **Reports & Dashboards** for tracking.
6. **Admin** oversees performance and ensures SLA compliance.



3. Technology Stack

Layer	Technology / Tool	Purpose
Frontend	ServiceNow Service Portal	User interface for submitting and viewing tickets
Backend	ServiceNow Platform (Flow Designer, Script Includes)	Automates ticket assignment logic
Database	ServiceNow Tables	Stores ticket, user, and assignment data
Automation Tools	Flow Designer / Business Rules	Manages routing workflows
Scripting	JavaScript (Server-side & Client Scripts)	Implements routing logic and data handling
Notifications	ServiceNow Email Engine	Sends alerts and status updates
Analytics	ServiceNow Reports & Dashboards	Tracks ticket metrics and SLA compliance
Deployment	Update Sets	Moves configurations between environments

- 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



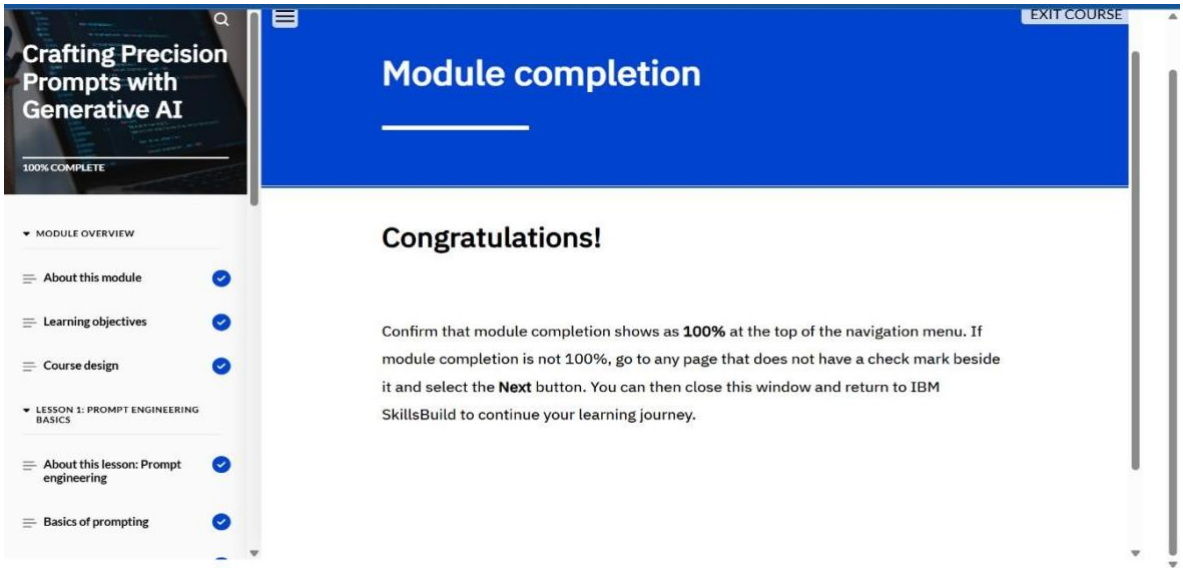
MODULE 2- Crafting Precision Prompts with Generative AI

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

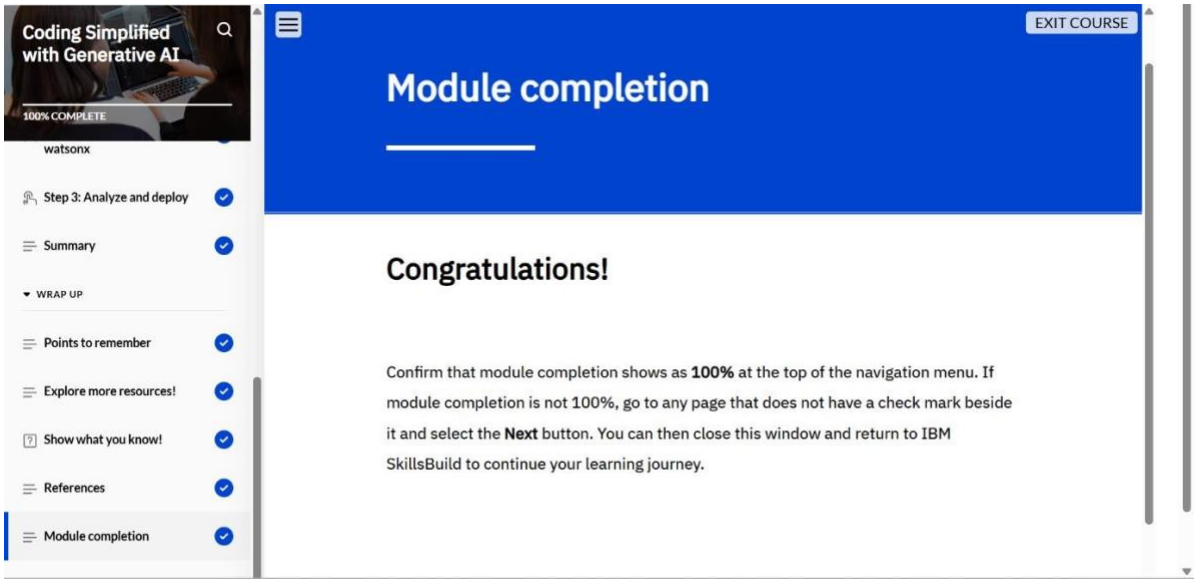


MODULE 3- Coding Simplified with Generative AI

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**:
    - Example: State is Resolved or Closed
- Under the **Pause Condition**:
    - Add: **State is On Hold**

This ensures the SLA timer **pauses** when the incident is moved to **On Hold**, and **resumes** when it returns to another **New** state

- ***Test the SLA Behavior***
- Create a test incident.
- Confirm SLA starts when an incident is created.
- Change state to **On Hold** — SLA should pause.
- Change back to **Active** — SLA should resume.
- Resolve the incident — SLA should stop.

6. **Configure an email notification that alerts the assigned group whenever a new *Change Request* is created.**

Solution:

- **Navigate to Notifications**
- In the **Application Navigator**, type **Notifications**.
- Go to **System Notification > Email > Notifications**.
- **Create a New Notification**

1. Click **New**.
2. Fill in the basic details:
    a. **Name**: *New Change Request Assigned Group Alert*
    b. **Table**: *Change Request [change_request]*
    c. **Active**: Checked

- **Define When to Send**

1. Under **When to send**, configure:
    a. **When to send**: *Insert* (since you want this when a new record is created).

- **Define Who Will Receive**

1. In the **Recipients** tab:
    a. Under **Users/Groups in fields**, choose **Assigned to group** (or the field name for assigned group).
    b. This ensures the entire assigned group gets the email.

- **Define What Will Contain**
- In the **What it will contain** tab:
    - **Subject**: New Change Request Created - ${number}

      Message HTML (sample):

      A new Change Request has been created.

- Number: ${number}
- Short Description: ${short_description}
- Requested By: ${requested_by}
- Assignment Group: ${assignment_group}
- State: ${state}

Please review and take necessary action.

- **Save & Test**
- Save the Notification.
- Create a new **Change Request** record, assign it to a group.
- Verify that the email goes out to all members of the Assigned Group.

7. **Create a report showing the number of incidents opened by each department in the last 30 days.**

- ***Navigate to Reports***
- Go to Reports > Open Reports Modules.
- Click Create a Report.
- ***Define Report Source***
- Name: **Incidents by Department - Last 30 Days**
- Source Table: **Incident**
- ***Set Conditions***
- **Under Filter, add:**
  - Opened At → on or after → Today - 30 days
  - Department → is not empty ***(optional, to exclude unassigned)***
- ***Choose Report Type***
- Select Type: **Bar Chart** or **Pie Chart** (or **List** if you prefer tabular view)
- ***Configure Grouping***
- Under Group By, select: **Department**
- Under Aggregation, choose: **Count**
- ***Save and Run***
- Click Save.
- Click Run to view the report.

8. **Build a dashboard for Service Desk Managers showing KPIs like incidents by priority, created within a week, state wise also.**

Step 1: Create Individual Reports

You'll need to create three separate reports first:

- ***Incidents by Priority***
- Go to: Reports > Create New
- Name: Incidents by Priority
- Source Table: Incident

- Type: Bar Chart or Pie Chart
- Group By: Priority
- Aggregation: Count
- 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]*
- **Name**: *(auto-populates when you pick table + operation)*

- **Define the Condition / Role**

  In the **Requires role** field, add: **admin**

  - This ensures only users with the **admin** role can delete records.
- **Save & Test**
- Save the ACL.
- Test with a non-admin user → they should **not** see the delete option (or get a permission error if they try via URL).
- Test with an admin user → delete should work normally.

10. **Create a custom table and create two reference fields (ex: assignment group and assigned to). Display the users based on selection of assignment group.**

- **Create a Custom Table**

1. In the Application Navigator, type **Tables**.
2. Go to **System Definition > Tables**.
3. Click **New**.
   - Name: *u_custom_task*
   - Label: *Custom Task.*
   - Save.

- **Add Fields**

1. Open your table and go to the **Columns** tab.
2. Add two reference fields:
   - **Assignment Group** → Type = *Reference*, Table = *sys_user_group*.
   - **Assigned To** → Type = *Reference*, Table = *sys_user*.

- **Configure Reference Qualifier on "Assigned To"**
- We need to filter "Assigned To" users based on the selected Assignment Group.

Using Reference Qualifier

- Right click on the **Assigned To** field, click on **Configure Dictionary.**
- Go to **Dependent** Section, give the name of the Assignment Group(ex: u_ass_group)
- Update and Test the functionality.

11. **How to auto assign incidents when user selects a category as network, the same incident be assigned to Network group.**

Solution:

1. Go to Flow Designer → Designer.
2. Click New Flow.

- Name: Assign Incident by Category
- Trigger: Created or Updated → Table = Incident

3. Add a If action (Condition) with expression:
   - Select Trigger Record Category is Network
4. Under the If branch, add Action → Update Record:
   - Record: Trigger → Incident(Trigger Record)
   - Set field Assignment group → Network
5. Save and Activate the flow.
6. Test the Flow.

12. HR Groups members are only able to see HR Related Records in servicenow?

Solution:

Step 1: Create a Role for HR Access

Navigate to:
User Administration → Roles → New

1. Enter:
   - Name: hr_access
   - Description: Role to allow access to HR Cases
2. Click Submit.

Step 2: Assign the Role to HR Group

1. Navigate to:
   User Administration → Groups
2. Open your HR group record.
3. In the Roles tab → click Edit.
4. Move hr_access from Available → Selected.
5. Click Save.

Now all members of the HR group have the hr_access role.

Step 3: Create Access Control (ACL) for Viewing HR Cases

1. Navigate to:
   System Security → Access Control (ACL)
2. Click New.

Fill in:

Field	**Value**

Type	record
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.

}

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);