ServiceNow Admin Certification

Module1: Understanding Architecture and UI Navigation

Overview Of UI of the instance

In ServiceNow, the User Interface (UI) is the visual part of the platform through which users interact with the system — whether to manage incidents, view dashboards, or automate workflows.

1. Key Components of ServiceNow UI

a) Application Navigator (Left Panel)

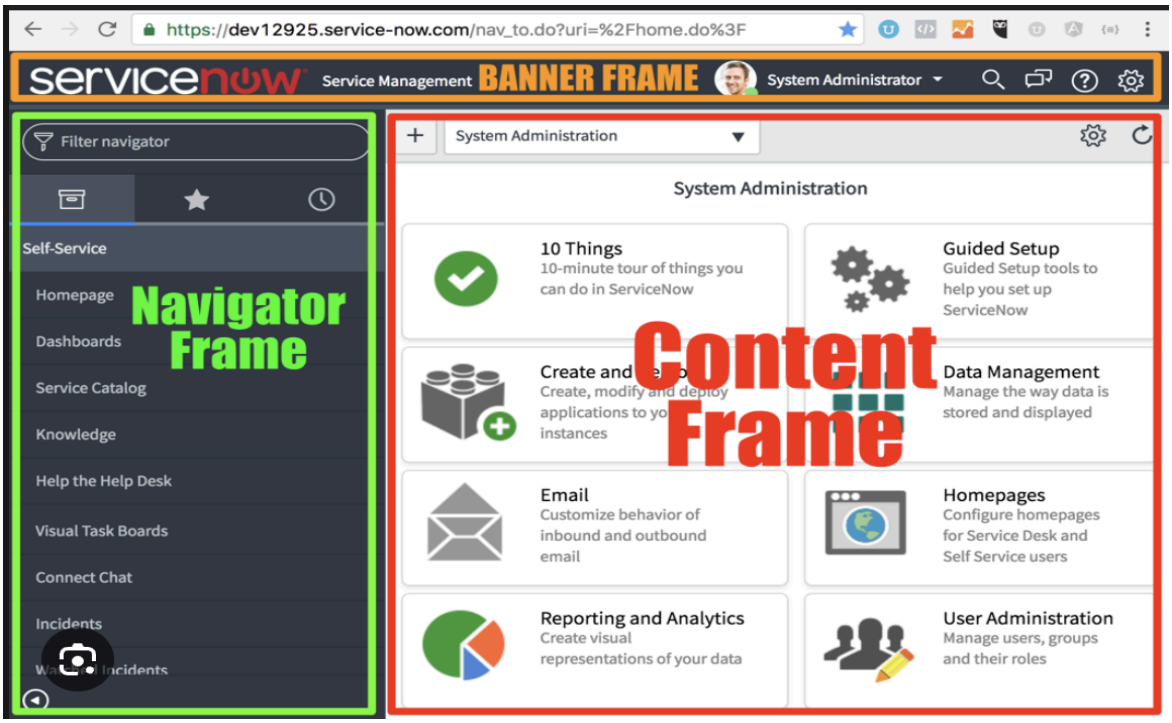
- It helps users **find applications and modules**.
- You can **filter** using keywords.
- Example: Typing “Incident” shows all related modules like “Create New,” “All,” “Open,” etc.

b) Content Frame (Center Area)

- The main working area where **forms, lists, and dashboards** appear.
- Example: When you open an incident record, the form loads here.

c) Banner Frame (Top Bar)

- Contains:
    - **User menu** (profile, preferences, logout)
    - **Settings icon** (theme, notification settings)
    - **Global search bar**
    -
    - **Help or support icon**



2. Common UI Elements

UI Element	Description
Forms	Used to create or edit records (e.g., Incident Form).
Lists	Display multiple records in table format (e.g., all open incidents).
Filter Navigator	Quickly access apps/modules.
Breadcrumbs	Show navigation path and let you filter lists.
Related Lists	Display linked records at the bottom of a form (e.g., related tasks).
UI Actions	Buttons, links, or context menu items that perform actions (e.g., Save, Update, Delete).
UI Policies	Dynamically change form behavior (make fields visible, mandatory, or read-only).
UI Pages	Custom web pages built with Jelly/HTML for unique interfaces.
UI Macros	Reusable UI components (like a search bar).

3. Customizing the UI

Developers or admins can customize:

- **Form layouts** (add/remove fields)
- **List layouts**
- **UI actions and policies**
- **Themes and branding**
- **Service Portal pages** (using widgets, CSS, JS)

Example Scenario

When a support agent logs in, they see the **Agent Workspace**, which has:

- Left side: Navigation for Incidents, Problems, etc.
- Center: Active ticket details
- Right: Related information or customer history
  This allows them to handle incidents efficiently within a single view.

Changing Logo Of Servicenow

In **ServiceNow**, the **System Properties → Basic Configuration** module allows you to **customize the look, feel, and branding** of your instance — things like the company logo, colors, name, and banner.

Key Settings in Basic Configuration

Setting	Description
Company Name	Sets the organization name shown in the banner.
Banner Image	Lets you upload a custom logo or image shown at the top of the UI.
Banner Background Color	Choose or set a hex color code (e.g., `#2C3E50`).
Text Color	Adjusts the banner text color for readability.

Form Header Background Color	Changes the header color on forms.
List Title Background Color	Sets color for list view headers.
System Date/Time Format	Defines the default format shown in all records.
System Time Zone	Sets the default time zone for users.
Welcome Message	Message displayed to users when they log in.
Default Theme	Selects the color scheme (Light, Dark, or Custom).

Example: Customizing Branding

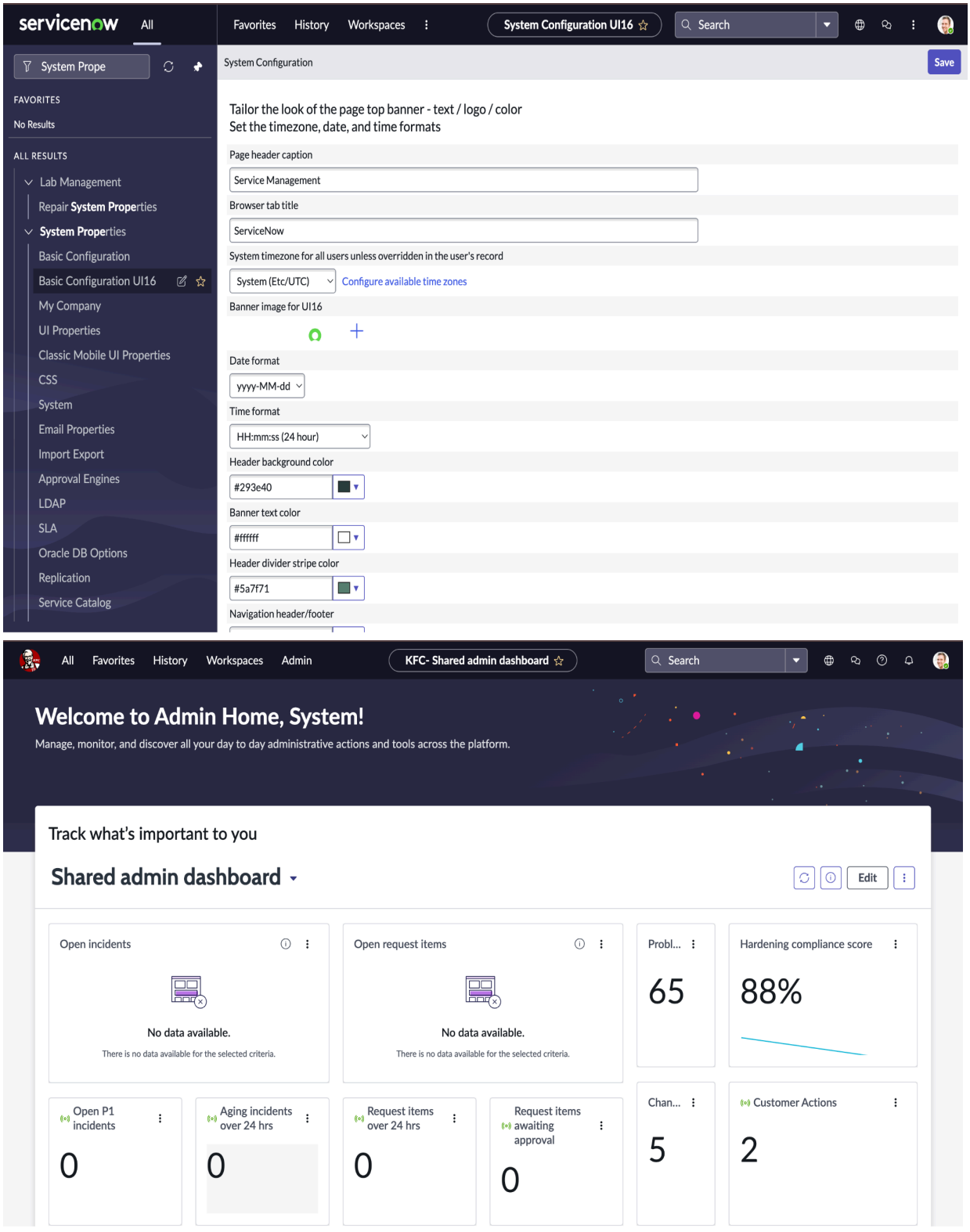
Suppose your company is *TechSphere Pvt. Ltd.*, and you want to:

- Add your **logo**
- Set **banner color** to blue
- Change **text color** to white

Steps:

1. Navigate to **System Properties** → **Basic Configuration (UI16)**
2. Upload your logo file.
3. Set banner background = `#007BFF`
4. Set text color = `#FFFFFF`
5. Click **Save** ✅

Now your instance will have a personalized theme and logo.



See above image changing logo of the organization

Application & Module

An **Application** in ServiceNow is a **collection of related modules, tables, and functionalities** that serve a specific purpose.

A **Module** is a **subsection inside an Application**.

Example:-
Application: Incident
- Create New
- Open
- Assigned to Me
- All

How to Create a Custom Application

Steps:

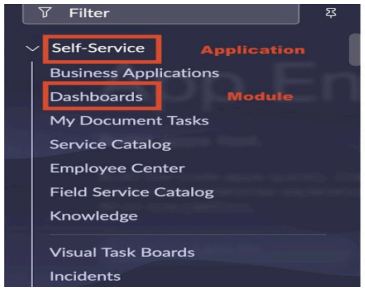
1. Navigate to: **System Definition → Application Menu**
2. Enter **Name**, **Scope**, and **Description**
3. Save it → ServiceNow automatically creates:
    - A **custom table**
    - A **menu (application)**
    - A **module**

You can then add more modules manually as needed.

How to Create a Module

Steps:

1. Go to: **System Definition → Modules**
2. Click **New**
3. Fill in:
    - **Title:** Name of the module
    - **Application:** Select the parent application
    - **Link type:** Choose what it opens (Form, List, URL, Script)
    - **Table:** Specify which table the module shows
4. Click **Submit**



Global Search:- The Global Search bar allows users to search across the entire ServiceNow instance — including records, tables, users, and knowledge articles — from one place.

Favorites:- The Favorites feature lets users bookmark frequently used modules or records for quick access.

Impersonate User:-The Impersonate User feature allows an admin to log in as another user without needing their password.
It's used to test permissions, roles, and UI visibility for that specific user.

Elevate Role:- Elevate Role allows a logged-in system administrator to temporarily gain higher privileges — specifically roles like security_admin — to perform sensitive configuration or security-related tasks.

Module2: List & Filter

List

A List in ServiceNow displays multiple records from a particular table in a tabular format (rows and columns).

It's like an Excel sheet inside ServiceNow — where each row = record and each column = field.

Example:

If you open **Incident** → **All**, you'll see:

Number	Short Description	Caller	State	Assigned To
INC001001	Email not working	John	New	—
INC001002	Laptop issue	Emma	In Progress	David

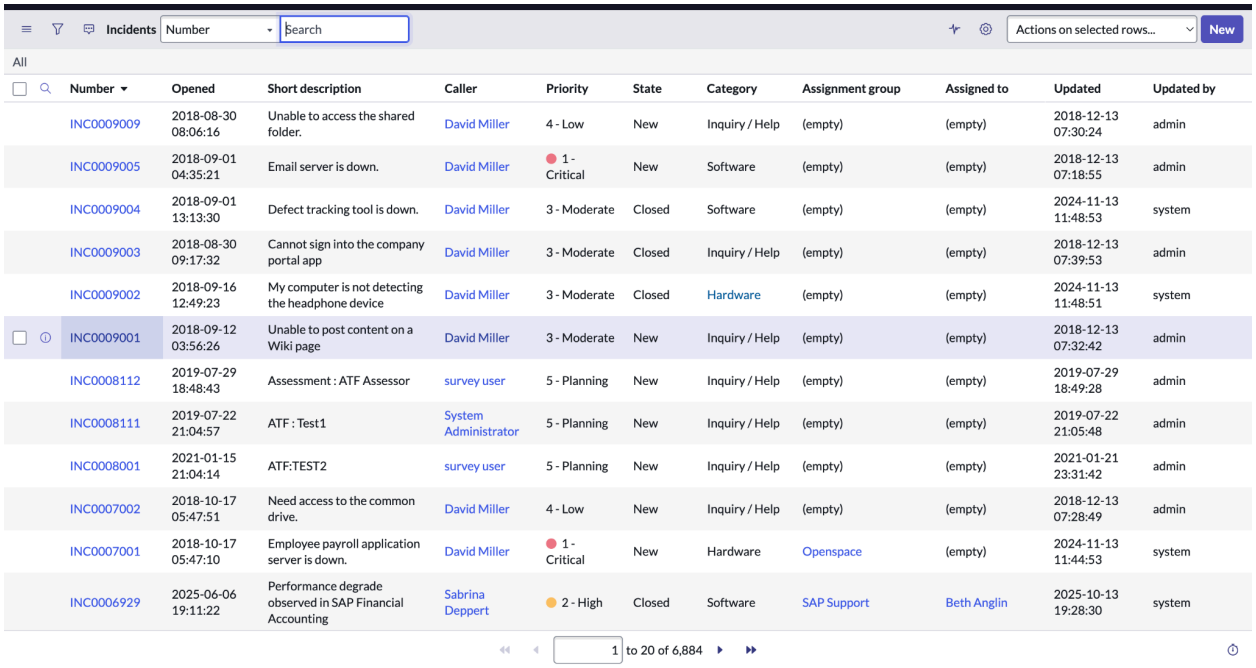
That's an **Incident List View** showing all incident records.

Components of a List

Component	Description
List Header	Displays column names (like Number, State, Priority).
Column Context Menu (⚙️)	Options like Sort, Group, Filter, Personalize.
List Title Bar	Shows the table name and total records.
Breadcrumbs	Show active filters applied on the list.
List Controls	Icons to export, refresh, or configure the list.

Common Actions in Lists

- **Sort:** Click a column header (A→Z or Z→A).
- **Group By:** Right-click a column → *Group by this field.*
- **Personalize List:** Add/remove columns for your view.
- **Export:** Export list data to Excel, PDF, or CSV.



Number	Opened	Short description	Caller	Priority	State	Category	Assignment group	Assigned to	Updated	Updated by
INC0009009	2018-08-30 08:06:16	Unable to access the shared folder.	David Miller	4 - Low	New	Inquiry / Help	(empty)	(empty)	2018-12-13 07:30:24	admin
INC0009005	2018-09-01 04:35:21	Email server is down.	David Miller	1 - Critical	New	Software	(empty)	(empty)	2018-12-13 07:18:55	admin
INC0009004	2018-09-01 13:13:30	Defect tracking tool is down.	David Miller	3 - Moderate	Closed	Software	(empty)	(empty)	2024-11-13 11:48:53	system
INC0009003	2018-08-30 09:17:32	Cannot sign into the company portal app	David Miller	3 - Moderate	Closed	Inquiry / Help	(empty)	(empty)	2018-12-13 07:39:53	admin
INC0009002	2018-09-16 12:49:23	My computer is not detecting the headphone device	David Miller	3 - Moderate	Closed	Hardware	(empty)	(empty)	2024-11-13 11:48:51	system
INC0009001	2018-09-12 03:56:26	Unable to post content on a Wiki page	David Miller	3 - Moderate	New	Inquiry / Help	(empty)	(empty)	2018-12-13 07:32:42	admin
INC0008112	2019-07-29 18:48:43	Assessment : ATF Assessor	survey user	5 - Planning	New	Inquiry / Help	(empty)	(empty)	2019-07-29 18:49:28	admin
INC0008111	2019-07-22 21:04:57	ATF : Test1	System Administrator	5 - Planning	New	Inquiry / Help	(empty)	(empty)	2019-07-22 21:05:48	admin
INC0008001	2021-01-15 21:04:14	ATF:TEST2	survey user	5 - Planning	New	Inquiry / Help	(empty)	(empty)	2021-01-21 23:31:42	admin
INC0007002	2018-10-17 05:47:51	Need access to the common drive.	David Miller	4 - Low	New	Inquiry / Help	(empty)	(empty)	2018-12-13 07:28:49	admin
INC0007001	2018-10-17 05:47:10	Employee payroll application server is down.	David Miller	1 - Critical	New	Hardware	Openspace	(empty)	2024-11-13 11:44:53	system
INC0006929	2025-06-06 19:11:22	Performance degrade observed in SAP Financial Accounting	Sabrina Deppert	2 - High	Closed	Software	SAP Support	Beth Anglin	2025-10-13 19:28:30	system

Filter

A **Filter** allows you to **limit which records** are shown in a list by setting specific conditions.

It helps you find only the records you need.

Example:

Show all **Incidents** where

- **State = In Progress**
- **Priority = 1 - Critical**

👉 You'll see only those matching records.

Breadcrumbs in Filters

Breadcrumbs appear at the top of the list showing active filters.

You can click on any part of a breadcrumb to **remove** or **change** a filter.

Personalize List

It's a feature that lets users change the appearance and structure of a list view — including which columns appear, their order, and sorting preferences.

Module3: Forms

A Form is the interface used to view, create, or modify a record in a ServiceNow table. Each form corresponds to a table and displays its fields.

Form Layout

In ServiceNow, Form Layout defines how fields are arranged on a form for a specific table—allowing admins to control visibility, grouping, and user experience.

Types of Forms

1. **Standard Form**
    - Default form used for viewing or updating a record.
2. **Catalog Item Form**
    - For Service Catalog items, where users submit requests.

3. **Task Form**
    - Specific to task tables like Incident, Problem, Change.
4. **Customized Form**
    - You can create your own forms with:
        - Form Designer
        - Form Layout
        - Form Sections
        - UI Policies and Actions

Form Customization

a) Form Designer

- Drag-and-drop interface for customizing form layout.
- Add, remove, or reorder fields and sections.

b) Form Layout

- Classic layout editor.
- Divide fields into available and selected fields.

c) Form Sections

- Group related fields.
- Can be collapsed or expanded.

d) Field Types & Properties

- Each field has properties:
    - Mandatory → Must be filled before saving
    - Read-only → Cannot be edited
    - Visible → Can show/hide field dynamically
    - Default value → Pre-populate field

e) UI Policies

- Apply dynamic behavior to form fields.

- Example:
    - If `Category` is `Hardware`, show `Hardware Model` field.
    - Can make fields mandatory, read-only, or visible/invisible.

f) Client Scripts

- JavaScript scripts running in the browser for dynamic form behavior.
- Types:
    - onLoad → When form loads
    - onChange → When a field value changes
    - onSubmit → Before saving form
    - onCellEdit → In list editing

Form Actions

- Insert / Update / Delete
- Workflows can trigger on form submit.
- Approvals can be integrated into forms.
- Notifications can trigger based on form field values.

Form Views

A table can have multiple views:

- Default View → Standard form
- Custom Views → Different forms for different roles
- Example: `Incident View for Support`, `Incident View for Management`

Views allow you to:

- Show only relevant fields to certain users
- Simplify the UI for specific roles

Form Personalization

- **Users can personalize forms (if allowed):**
    - Hide unnecessary fields
    - Reorder fields
    - Save a personal view of the form

Module4: Tables and Fields

Table

A table in ServiceNow is a database table that stores records (rows). Every record represents an instance of a table.

Key points:

- Tables have **fields** (columns) that store data.
- Tables can **extend other tables** (inherit fields).
- Two types of tables:
    1. **System Tables** – Pre-built, e.g., `incident`, `task`, `change_request`.
    2. **Custom Tables** – Created by users, e.g., `u_project`, `u_asset`.

Table Hierarchy

ServiceNow tables follow a hierarchy:

1. Parent Table – Base table that provides common fields.
2. Child Table – Extends a parent table and inherits all its fields.
3. Task Table – A common system table for work items (Incident, Problem, Change).

Fields in ServiceNow

A field is a column in a table storing specific information about a record.

Components of a field:

- Name → Internal name (database name)
- Type → String, Choice, Reference, Boolean, etc.
- Properties → Mandatory, Read-only, Default value, Unique, Visible

Table Relationships

Tables often relate to each other:

1. Reference Fields
    - A field points to another table.
    - Example: `Incident → Assigned to → User table`
2. Extensions
    - A table can extend another table to inherit fields.
    - Example: `Problem` extends `Task`
3. Many-to-Many (M2M)
    - Special table storing relationships between two tables.
    - Example: `sc_cat_item_category` links catalog items to categories
4. Database Views
    - Combines multiple tables for reporting.
    - Example: Combine `Incident` + `User` + `Assignment` for reports

Creating Tables

1. Navigate: System Definition → Tables → New
2. Fill in:
    - Label → Display name
    - Name → Auto-generated database name
    - Extends Table → Choose parent table if needed

    - Create Module → Optional, adds navigation module

3. Add fields manually → save → table is ready

Forms and Fields Connection

- Forms display table fields.
- UI Policies or Client Scripts can dynamically make fields mandatory, read-only, or hidden.
- Reference fields create relationships between forms.
- Related lists in forms show child table records.

Views

- Tables can have multiple **views** to show different sets of fields.
- Example: `Incident` table:
    - `Default View` → All fields
    - `Support View` → Only necessary fields
    - `Manager View` → Only summary fields

Benefits:

- Simplifies UI for roles
- Hides unnecessary fields
- Controls access

Security and Access Control

- Fields and tables are controlled using **Access Control Rules (ACLs)**.
- ACLs define:
    - Who can **read**, **write**, or **create** a record/field.
- Role-based visibility ensures sensitive data is protected.

Schema Map

A Schema Map is a visual representation of how tables in ServiceNow relate to one another.
It helps you see parent and child relationships, reference fields, and extensions between tables.

🔹 Purpose

- To understand data structure and hierarchy
- To visualize table inheritance
- To identify reference relationships (foreign keys)
- To help design new tables correctly before creation

How to Open a Schema Map

You can open it in two ways:

✅ Method 1:

1. Navigate to System Definition → Tables
2. Search and open any table (e.g., `incident`)
3. Click the Schema Map related link at the bottom of the form

```
Tables
  ├─ System Tables (Incident, Task, User)
  ├─ Custom Tables (u_project, u_asset)
  │
Fields
  ├─ String, Text, Choice, Reference, Boolean, Date, Journal
  │
Table Relationships
  ├─ Extensions (child → parent)
  ├─ Reference Fields
  ├─ Many-to-Many
  └─ Database Views
  │
Forms
  ├─ Display fields
  ├─ UI Policies / Client Scripts
  ├─ Related Lists
  │
Schema Map
  ├─ Visual relationships
  ├─ Plan table creation
  └─ Avoid duplicate fields
```

Module5: Access Control

Access Control in ServiceNow is a security mechanism used to restrict user access to:

- Tables
- Fields
- Records
- Operations (CRUD) → Create, Read, Write, Delete

These rules determine who can do what in the system.

Access Control Rule

Each access control is a **rule** that defines:

- Which **object** (table or field)
- Which **operation** (read/write/create/delete)
- **Under what conditions** a user can access that data.

Example:

```
Table: Incident
Operation: read
Condition: Assigned to is (current user)
Script: return gs.hasRole('itil');
```

✅ Meaning: Only users with the “itil” role can read incident records assigned to them.

Types of Access Controls

(a) Table-level ACL

Controls access to the entire table (e.g., `incident.*`).

(b) Field-level ACL

Controls access to a specific field (e.g., `incident.short_description`).

(c) Record-level ACL

Uses conditions or scripts to allow access only to specific records.

Operations in ACL

Operation	Description
`create`	Can user insert records?
`read`	Can user view records/fields?
`write`	Can user update records/fields?
`delete`	Can user remove records?

How Access Control Works — Evaluation Order

When a user tries to access a record:

1. **System checks table ACLs**
2. **Checks parent tables** (if table is extended)
3. **Checks field ACLs** (if field is being accessed)
4. All relevant ACLs must return **true** for access to be granted.

⚠️ If **any one** ACL returns `false`, **access is denied.**

Components of an Access Control Rule

Each ACL rule consists of three main parts:

(a) Role

- Restricts access to users having specific roles.
- Example: `admin`, `itil`, `hr_user`

(b) Condition

- Uses condition builder (like filters) to define logic.
- Example: `Assigned to is (Current user)`

(c) Script

- Scripted logic in JavaScript that returns true or false.

Example:

```
(function() {
  return gs.hasRole('admin') || current.assigned_to == gs.getUserID();
})();
```

-

➡️ All 3 must be true for access to be granted.

ACL Inheritance

If a table **extends another table**, it **inherits the ACLs** from its parent.

Example:

- `incident` extends `task`
- If `task.read` ACL is false → `incident` cannot be read even if `incident.read` is true.

Field vs Table ACLs

Type	Applies To	Example	Priority
Table ACL	Entire record	`incident.read`	Checked first
Field ACL	Specific field	`incident.short_description.read`	Checked after table ACL

👉 **Both must return true** for access.

Creating an Access Control Rule

Navigation:
`System Security → Access Control (ACL) → New`

Steps:

1. Choose Table (e.g., incident)
2. Choose Type (record/field)
3. Choose Operation (read/write/create/delete)
4. Add Roles, Conditions, and Script
5. Save and test

Module6: Flow Designer

Flow Designer in ServiceNow is a **no-code/low-code automation tool** used to **build, automate, and manage workflows** across the ServiceNow platform.

It helps you:

- Automate business processes
- Eliminate manual work
- Connect ServiceNow apps with external systems

💡 **In short:**
It automates “what happens when something happens.”

Flow Designer Key Concepts

Term	Description
Flow Designer	Application to create, edit, and manage flows
Flow	A sequence of actions triggered by an event
Trigger	Defines when a flow should start
Action	A single operation in a flow (like update record, send email)
Data Pill	Reusable data token passed from one step to another
Subflow	A reusable flow that can be called inside another flow
Flow Logic	Branches, loops, if/else for decision-making
IntegrationHub	Used to connect Flow Designer to external systems (Slack, Gmail, etc.)

Components of Flow Designer

a) Trigger

- Defines **when** the flow starts.
- Examples:
    - When a record is created or updated
    - When a user submits a catalog item
    - On schedule (time-based)
    - Manually triggered by another process

b) Actions

- Define **what the flow does.**
- Examples:
    - Create a record
    - Update a field
    - Send email
    - Run a script
    - Call a subflow
    - Log message

There are two types:

1. **Core Actions** (built-in ServiceNow)
2. **Custom Actions** (you can create your own)

c) Flow Logic

Used for **decision-making** or repeating steps.

Examples:

- **If / Else** condition
- **For Each** loop
- **Wait for condition**

d) Data Pill

A **data reference** used to pass output from one step to another.

Example:
If the trigger captures an “Incident record,”
you can use `Trigger → Record → Short Description` as a data pill in later steps.

How Flow Designer Works (Flow Execution Order)

1. **Trigger activates** (e.g., new incident created)
2. **Flow runs** (it executes actions in order)
3. Each **action** produces **data pills** for next steps
4. **Flow logic** checks conditions and decides the next step
5. Ends when all actions complete successfully (or error occurs)

Creating a Flow — Step-by-Step

Steps:

1. **Go to Flow Designer**
    - Navigation: `All → Process Automation → Flow Designer`
2. **Click “New” → Flow**
3. **Add Flow Details**
    - Name: `Auto Assign Incident`
    - Table: `Incident`
4. **Add Trigger**
    - Example: `When record is created`
5. **Add Actions**
    - Step 1: Add action → “Update Record”

        - Field: Assigned to = “IT Support”
    - Step 2: Add action → “Send Email”
        - To: `current.opened_by.email`
        - Subject: “Your incident is being handled”
6. **Click Save & Activate**
7. **Test the Flow**
    - Create a new Incident → Flow auto-triggers.

Real-Time Use Cases

1. **Automatic Email Notification** – when ticket created/updated
2. **Auto Assignment** – assign incidents based on category
3. **Approval Workflow** – approve/reject requests automatically
4. **Service Catalog** – handle item requests using flow
5. **Integration** – send data to Slack, Teams, or external APIs

---

Module7: Notification

A **Notification** in ServiceNow is a **message automatically sent** (usually via email, SMS, or push) to users when a specific **event** or **condition** occurs.

🧠 Example:

- When an **incident is assigned** → send email to assignee
- When a **request is approved** → send notification to requester
- When a **task is completed** → notify manager

Purpose of Notifications

Objective	Description
📨 Communication	Keep users informed automatically
⚡ Automation	No need for manual emails
🧭 Tracking	Records of important actions
🎯 Consistency	Standard email formatting via templates

Notification Types in ServiceNow

Type	Description
Email Notification	Sent through email (most common)
SMS Notification	Sent through text messages
Push Notification	Sent through ServiceNow mobile app
Event Notification	Triggered via custom events
Flow Designer Notification	Sent via "Send Notification" or "Send Email" action

Where Notifications Are Configured

System Notification → Email → Notifications

Components of a Notification

Every notification has the following components 👇

Component	Description
Name	Identifies the notification
Table	Which table (like Incident, Request, Task) triggers it
When to send	Defines the condition/event
Who will receive	Recipients (users, groups, roles)
What it will contain	Email subject, body, and template
Email Template (optional)	Predefined structure used to format email body

How Notification Works

1. A record event occurs (like update, insert, delete).
2. System evaluates conditions in notification.
3. If matched → notification is triggered.
4. Template merges dynamic data (fields, user info).
5. Email/SMS/Push sent to recipients.

Steps to Create an Email Notification

🧭 Navigation:

`System Notification → Email → Notifications`

🧩 Step-by-Step:

1. Click New
2. Fill in:
    - Name: “Incident Assigned Notification”
    - Table: Incident
3. When to send:
    - Trigger: Record Insert or Update
    - Condition: `Assigned to changes`
4. Who will receive:
    - Add → `Assigned to`
    - Or specific role/group (like `itil`)
5. What it will contain:
    - Subject: `Incident ${number} has been assigned to you`

Message HTML:

```
 Hello ${assigned_to.name},
<br><br>
You have been assigned a new Incident: ${number}.
<br>
Short Description: ${short_description}
<br>
Priority: ${priority}
```

```
<br>
<a href="${URI_REF}">Click here</a> to view the record.
```

-
6. Save
7. Test by assigning a new incident.

✅ Result: Email sent automatically when incident is assigned.

Using Templates in Notifications

Templates are reusable **HTML email structures** that define the design and layout of the message.

🧭 Navigation:

```
System Notification → Email → Templates
```

🧱 Template Structure

A template has:

- **Name**
- **Table**
- **Subject**
- **Message HTML (Body)**

🧠 Example:
Template Name: Incident Update
Table: Incident
Subject: Update on ${number}
Message HTML:

```
<h3>Incident Update: ${number}</h3>

<p><b>Short Description:</b> ${short_description}</p>

<p><b>Priority:</b> ${priority}</p>
```

```
<p><b>Updated by:</b> ${sys_updated_by}</p>

<p><a href="${URI_REF}">View Incident</a></p>
```

You can use this template in multiple notifications.

Dynamic Content (Email Variables)

Inside the template or notification body, you can use variables like:

Variable	Description
`${number}`	Record number
`${short_description}`	Short description of record
`${assigned_to.name}`	Name of assigned user
`${priority}`	Priority value
`${URI_REF}`	Link to the record
`${current.field_name}`	Current record field
`${event.parm1}` / `${event.parm2}`	Parameters from event triggers

10. Example — "Incident Resolved" Notification

Field	Value
Name	Incident Resolved Notification
Table	Incident
Condition	State changes to "Resolved"
Recipients	Opened by
Subject	Incident ${number} Resolved
Body	

```html
Hello ${opened_by.name},

Your Incident ${number} has been resolved.

Short Description: ${short_description}
Resolution Notes: ${close_notes}

<a href="${URI_REF}">Click here</a> to view details.
```

12. Notifications in Flow Designer

You can also send notifications using **Flow Designer** instead of classic notifications.

Steps:

1. Create Flow → Add Action → "Send Notification" or "Send Email"
2. Choose:
   - **Recipients** (user, group, or email)
   - **Subject**
   - **Body**
3. Use **Data Pills** to insert dynamic record data.

🧠 Example:

Flow triggers when Incident created → sends email to "Assigned to" with incident details.

---

13. Email Logs & Debugging

Feature	Description	Path
Sent Emails	Shows all sent messages	`System Mailboxes → Outbound → Sent`
Email Logs	Detailed status logs	`System Logs → Emails`
System Logs	General logs (business rules, etc.)	`System Logs → All`
Test Email	Send test notifications	"Send Test Notification" button

Checking Mail:- search **Outbox** in search TAB;

Access Control for Notifications

Role	Access
`admin`	Full access
`notification_admin`	Can create/edit notifications
`email_admin`	Manage email configurations
`flow_designer`	Can send notifications from flows

Module8:Client Scripts & UI Policies

🧩 UI POLICIES (Client-side Logic)

What is a UI Policy?

A UI Policy is used to dynamically change the behavior or appearance of form fields on the client side (browser).
It does not require scripting — it's a no-code way to control form behavior.

🔷 Key Uses

- Make fields mandatory, read-only, or hidden based on conditions.
- Simplify user experience — show only relevant fields.
- Ensure users fill the form correctly before submission.

🔷 How It Works

A UI Policy has:

1. Condition → When should the rule run (e.g., Category = Hardware).
2. Actions → What should happen (e.g., Make Subcategory mandatory).
3. Reverse if False → Undo the actions if the condition is no longer true.

🔷 Example

Scenario:

If *Category* = *Hardware*, make *Subcategory* visible and mandatory.

Step	Field
Condition	Category == Hardware
Actions	Subcategory → Mandatory = True, Visible = True
Reverse if False	Yes (to hide when Category ≠ Hardware)

🔷 UI Policy Scripts (Optional)

You can write small client-side scripts inside a UI Policy if you need custom logic.

Example:

```
function onCondition() {
  g_form.setMandatory('subcategory', true);
  g_form.setVisible('subcategory', true);
}
```

🔷 Advantages

✅ No scripting needed.

✅ Easier maintenance.

✅ Fast performance (runs on client side).

🔷 Best Practice

- Prefer UI Policies over Client Scripts when possible for form field visibility or mandatory control.
- Keep conditions simple.

Module9:Reports and Dashboard:-

A **Report** in ServiceNow is a way to **visualize and analyze data** stored in tables — for example, to show:

- Number of incidents by priority,
- Tasks assigned to each group,
- Monthly change requests trend, etc.

⚙️ Types of Reports

ServiceNow supports many types of reports, like:

Report Type	Example
List	Displays tabular data like a table
Bar / Column Chart	Incidents by priority
Pie Chart	Tasks by state
Pivot Table	Summarized data like Excel pivot
Line Chart	Monthly trends (like number of incidents created)
Calendar / Timeline	Events or tasks over time
Map Report	Data by location
Multilevel (Drilldown)	Click one chart section to see detailed data

Use of Reports

- Reports can be created by users to provide insights into business processes, track performance, identify trends, and support data-driven decision-making.
- They can be accessed from various locations in the ServiceNow platform, including dashboards, homepages, and list views, and can be exported in a variety of formats, such as CSV, Excel, PDF, and HTML.

Use Cases:-

1. **Create a Report in Pie Chart showing list of Incident ticket in High Priority**
2. **Create a dashboard and add it over there as well as Publish it**
3. **Schedule it for every Monday @11am GMT time.**