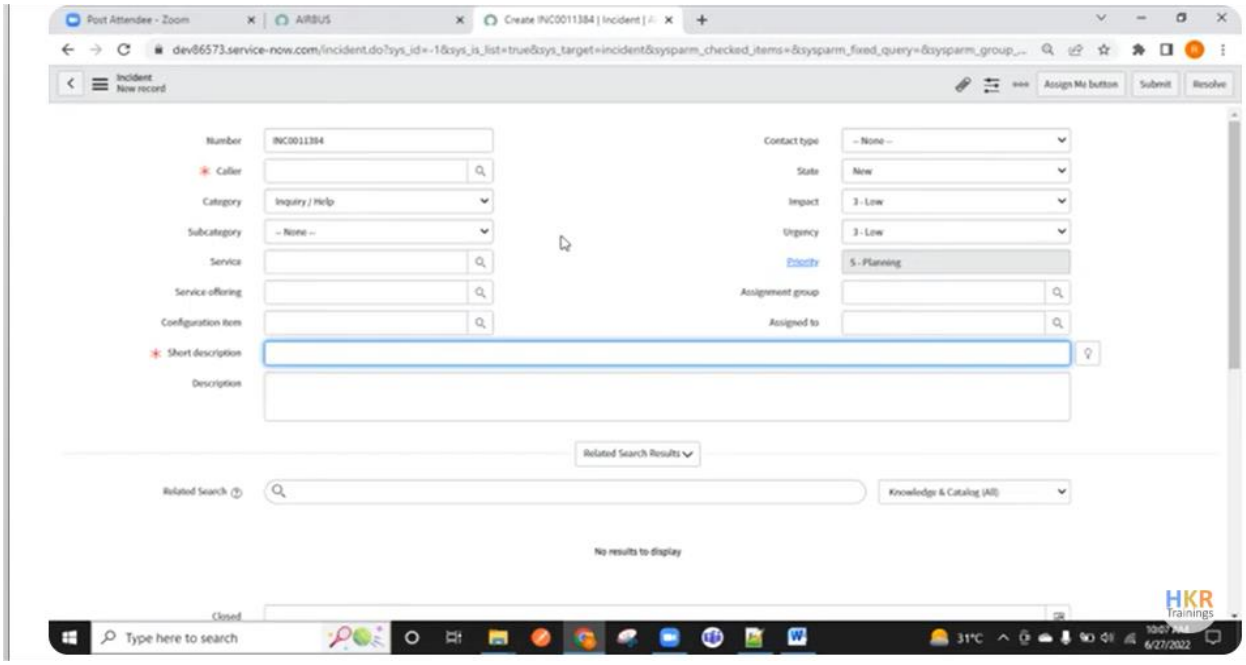
COGNIZANT WEEK 4

MODULE 4

SERVICENOW SCRIPTING

Scripting in ServiceNow involves writing code to customize the platform, automate tasks, and enhance functionality. It mainly uses JavaScript, and ServiceNow provides several scripting environments to support different customization needs.



CLEINT-SIDE SCRIPTING

Client-side scripting in **ServiceNow** involves writing JavaScript that executes in the user's browser, primarily for managing and manipulating the user interface (UI). These scripts are used to handle dynamic behaviors on forms, lists, and UI elements.

Client-side scripting refers to code that runs in a user's web browser, allowing dynamic interaction with web applications without needing server communication. In **ServiceNow**, client-side scripting is essential for managing user interactions, validating forms, and manipulating UI elements dynamically.

Types of Client Scripts

There are four main types of client scripts, each triggered by different events on forms or lists:

- **onLoad()**: Runs when a form is loaded. It's useful for manipulating form fields or setting initial field values.
- **onChange()**: Triggers when a field's value is changed by the user. Commonly used for validating input or updating other fields based on the change.
- **onSubmit()**: Executes when the user submits a form. Used for validating the entire form before submission or preventing it based on certain conditions.
- **onCellEdit()**: Runs when a cell in a list is edited. It's useful for validating data entered into list fields.

SERVER- SIDE SCRIPTING

Server-side scripting in **ServiceNow** is essential for controlling and automating backend processes. Unlike client-side scripts that run in the user's browser, server-side scripts execute on the server, making them ideal for data manipulation, processing logic, integrations, and enforcing business rules.

What is Server-Side Scripting?

Server-side scripting involves code that runs on the server and can:

- Interact with the database (query, insert, update, or delete records).
- Automate tasks like notifications, approvals, or workflow processes.
- Execute complex logic and handle integrations with other systems.
- Implement security measures like role-based access controls.

ServiceNow uses **JavaScript** for server-side scripting. Server-side scripts execute in the backend and are more powerful than client-side scripts since they can directly interact with databases and perform operations that aren't dependent on the user's browser or device.

2. Types of Server-Side Scripts

Business Rules

Business Rules are one of the most common types of server-side scripts. They are executed when a record is inserted, updated, deleted, or queried. They allow automation of tasks based on record changes and can be used to enforce rules.

- **Before**: Runs before a record is saved to the database.
- **After**: Runs after a record has been saved.
- **Async**: Runs after a record has been saved, but in the background without delaying form submission.
- **Display**: Executes when a record is retrieved and loaded into a form, allowing data to be pre-processed before being displayed.

INTEGRATION OF SCRIPTING

Integration scripting in ServiceNow refers to the process of writing scripts that facilitate communication and data exchange between ServiceNow and external systems or applications. These integrations can be used to automate processes, synchronize data, and enhance ServiceNow's functionality by pulling in or pushing out data.

Integration in ServiceNow typically involves server-side scripting and uses various technologies like **REST APIs**, **SOAP APIs**, **Outbound and Inbound Web Services**, **Scripted Web Services**, and **Middleware** tools. Scripting for integration enables custom business logic, transformations, and error handling during the exchange of data between systems.

FIX SCRIPT

A **Fix Script** in ServiceNow is a type of server-side script used to correct data issues, perform one-time data updates, or apply a specific change across records. Fix scripts are typically written to run once and are used in controlled scenarios, often during system updates, upgrades, or when there’s a need to bulk-update or clean up data in the instance. These scripts are manually executed by administrators and are not intended for recurring or automated use.

Key Characteristics of Fix Scripts:

1. **One-time Execution**
2. **Server-side**
3. **Designed for Data Cleanup and Corrections**
4. **Part of Update Sets**

Important Methods for Fix Scripts:

- **GlideRecord**: To query, update, insert, or delete records from the database.
  - gr.query(): Retrieves records based on the query conditions.
  - gr.update(): Saves changes to the current record.
  - gr.deleteRecord(): Deletes the current record from the table.
- **GlideSystem (gs)**: Provides system-level functionality such as logging and message creation.
  - gs.info(): Logs informational messages.
  - gs.error(): Logs error messages.

Executing a Fix Script:

1. Navigate to **System Definition > Fix Scripts** in the ServiceNow application navigator.
2. Create a new Fix Script and write the code needed for your update.
3. Test the Fix Script in a **sub-production environment** before running it in production.
4. Run the Fix Script manually from the form by clicking "Run Fix Script."

Fix Scripts in ServiceNow are powerful tools for administrators to address specific data-related issues and perform one-time updates or cleanups. Due to their ability to directly manipulate data, they must be handled with care, thoroughly tested, and executed in controlled environments to ensure system integrity.