Project Documentation

Educational Organisation Using ServiceNow

1.INTRODUCTION

Project Overview

The Educational Management System is a unified digital platform built on ServiceNow, aimed at optimizing core administrative tasks within educational institutions. It simplifies the management of student information, course enrollments, admission procedures, and academic progress monitoring through automated and structured workflows.

Purpose

The goal of this project is to eliminate inefficient, manual administrative tasks by introducing digital workflows powered by ServiceNow. It focuses on improving transparency, streamlining approval procedures, and enabling prompt communication between students and academic staff.

2.IDEATION PHASE

Problem Statement

Educational institutions often struggle with fragmented and manual handling of admissions, student data, and academic processes. Delays and miscommunication affect both students and staff.

User Insight Map

- **Who are the users?**
  Students, faculty members, and academic administration staff.
- **What do they think or feel?**
  Students often feel uncertain about their application progress, while staff struggle with handling excessive paperwork.
- **What do they see?**
  Disorganized records, long queues, and outdated paper-driven processes.
- **What do they say or do?**
  Frequently ask for updates and rely on submitting or managing physical documents.
- **What do they hear?**
  Complaints and dissatisfaction regarding delays and lack of proper communication.e

Brainstorming

Considered:

After evaluating options like Excel-based logs and manual registers, ServiceNow was selected for its structured workflow engine and scalability.

3. REQUIREMENT ANALYSIS

Student Journey Map

Student logs in → Applies for admission/course → System triggers approval → Faculty and HOD approve → Notification sent → Admission or enrollment confirmed

Solution Requirements

To support a streamlined and fully automated educational service request system, the following elements were implemented:

- **Custom Service Catalog**: Titled *"Educational Services"* to group all student-related requests.
- **Categorization**: Services were organized into logical groups—*Admissions*, *Academics*, and *Student Support*.
- **Catalog Items**: Included key student services such as *Admission Application*, *ID Card Request*, and *Grade Report Access*.
- **Role-Based Permissions**: Defined access levels for students, faculty members, and administrators.
- **Custom Data Tables**: Created dedicated tables to manage admission records and track academic progress.
- **Multi-Level Approval Workflows**: Enabled systematic review and approval at different stages (e.g., faculty and HOD).
- **Automated Notifications**: Configured email alerts for both approvals and rejections to keep users informed in real time.

Data Flow Diagram

User→Service Portal→Catalog→Request Form→Workflow→Task Table

↓

Notification → Mail → User/Group

Technology Stack

- **Platform:** ServiceNow
- **Scripts:** JavaScript (Glide APIs), HTML
- **Modules:** Service Catalog, Workflow Editor, Notifications, Tables, Access Control

4. PROJECT DESIGN

Problem-Solution Fit

Traditional academic workflows often suffer from delays, lack of visibility, and inefficiencies due to their manual nature. This solution addresses these challenges by introducing categorized service catalog items, automated approval processes, and real-time notification systems — significantly enhancing operational efficiency and the overall student experience.

Proposed Solution

To resolve the identified issues, the project incorporated the following key components:

- **Catalog:** A dedicated catalog named *Educational Services* was created.
- **Categories:** Organized into *Admissions*, *Academics*, and *Student Support* to simplify access and navigation.
- **Catalog Items:** Included essential services such as *Admission Requests*, *ID Card Applications*, and *Grade Reports*.
- **Custom Role:** Custom roles like *student_role* and *faculty_role* were defined to assign specific access privileges.
- **Groups:** An *Academic Office* group was established to manage and process requests.
- **Custom Tables:** Custom tables (*u_admission* and *u_student_progress*) were created to store and manage student-related data.
- **Workflows:** Approval processes were structured into multiple levels with final confirmation tasks to ensure proper review.
- **Notifications:** Email alerts were configured to keep users updated about request statuses throughout the workflow.

Solution Architecture

- **Frontend Interface:** Utilized the *Service Portal* for user interactions and service requests.
- **Logic Layer:** Implemented business logic using *Workflow Editor* and *UI Policies* for conditional form behavior.
- **Backend Structure:** Included custom data tables and dictionary entries to support data storage and management.
- **Notification Framework:** Dynamic HTML email templates were used to notify users of approvals or rejections.
- **Access Control Mechanism:** Role-based permissions ensured that users could only access features relevant to their roles.

5. PROJECT PLANNING & SCHEDULING

During this stage, the overall project was divided into smaller, well-defined phases to ensure modular development, systematic validation, and smooth deployment. The major tasks performed included:

- **Catalog Initialization:** A dedicated catalog titled *"Educational Services"* was established as the centralized hub for all student and staff service requests.
- **Category Design:** Categories such as Admissions, Academics, and Support were added for logical grouping of services.
- **Item Configuration:** Each catalog item (Admission, ID Card Request, Grade Report) was added with detailed descriptions, associated forms, and images where required.
- **User and Role Configuration:** Roles including *student_role* and *faculty_role* were created and linked to specific users. Additionally, the *Academic Office* group was assigned responsibilities for backend processing.
- **Table Design:** Custom tables (u_admission, u_student_progress) were created and structured with fields for capturing relevant data like student grades, contact information, application status, etc.
- **Workflow Development:** Approval workflows were set up involving faculty and HOD, followed by backend task handling by academic office members.
- **Portal and Interface Setup:** The Service Portal was configured for visibility of catalog items, form interactivity, and access control based on user roles.

- **Notification Integration:** Email notifications were designed using HTML templates to inform users about approval/rejection status with dynamic content.
- **Manual Testing:** Each functionality was tested step-by-step to ensure consistency, correctness, and complete process coverage.

6. IMPLEMENTATION WORKFLOW

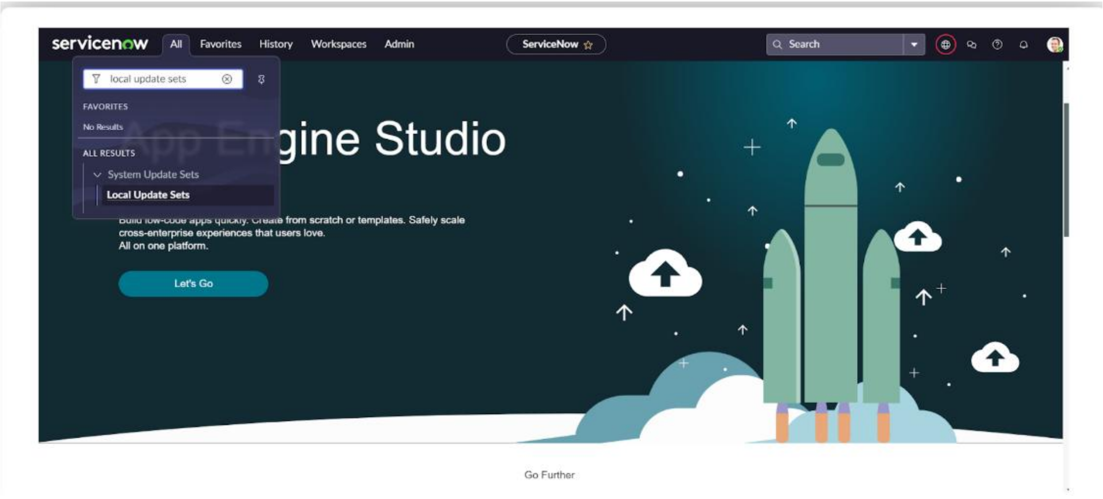
This phase outlines the sequence of steps performed in implementing the **Educational Management System**:

Step 1: Setting Up ServiceNow Instance

1. Sign up for a developer account on the ServiceNow Developer site “https://developer.servicenow.com”.
2. Once logged in, navigate to the "Personal Developer Instance" section.
3. Click on "Request Instance" to create a new ServiceNow instance.
4. Fill out the required information and submit the request.
5. You'll receive an email with the instance details once it's ready.
6. Log in to your ServiceNow instance using the provided credentials.
7. Now you will navigate to the ServiceNow

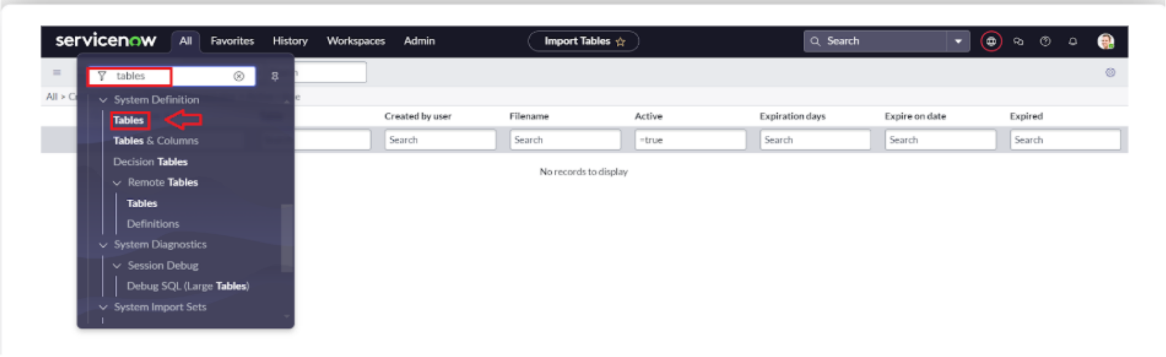
Step 2: Create a Update Set

- Click on All >> Local update sets.
- Click on new
- Enter the Details Name: Educational Organisation >> Click on Submit and make Current

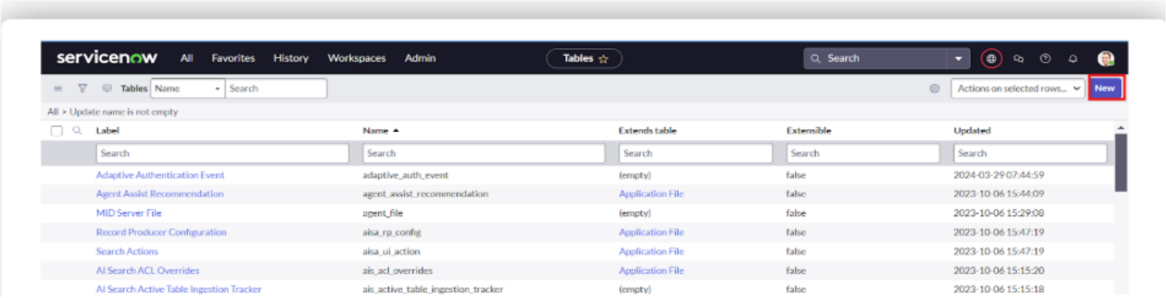


Step 3: Create a Table

- All >> Tables.
- Click **New**
- Enter the Label (Anything you want): Salesforce >> Click on Name it will Automatically generate Api name



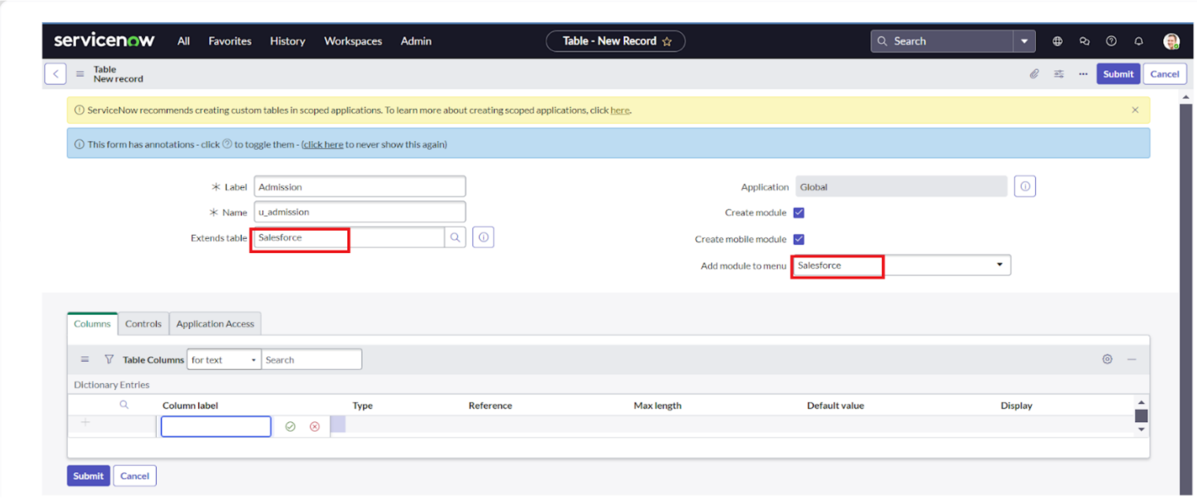
- Click on new



Label	Name ▲	Extends table	Extensible	Updated
Adaptive Authentication Event	adaptive_auth_event	(empty)	false	2024-03-29 07:44:59
Agent Assist Recommendation	agent_assist_recommendation	Application File	false	2023-10-06 15:44:09
MID Server File	agent_file	(empty)	false	2023-10-06 15:29:08
Record Producer Configuration	aisa_rp_config	Application File	false	2023-10-06 15:47:19
Search Actions	aisa_ui_action	Application File	false	2023-10-06 15:47:19
AI Search ACL Overrides	ais_acl_overrides	Application File	false	2023-10-06 15:15:20
AI Search Active Table Ingestion Tracker	ais_active_table_ingestion_tracker	(empty)	false	2023-10-06 15:15:18

Step 4: Create Admission Table

- Create an Admission Table with Columns given.
- Select Extends Table >> Salesforce and also Select Add module to menu >> Salesforce.
- Create Fields as shown



ⓘ ServiceNow recommends creating custom tables in scoped applications. To learn more about creating scoped applications, click here.

ⓘ This form has annotations - click ⓘ to toggle them - (click here to never show this again)

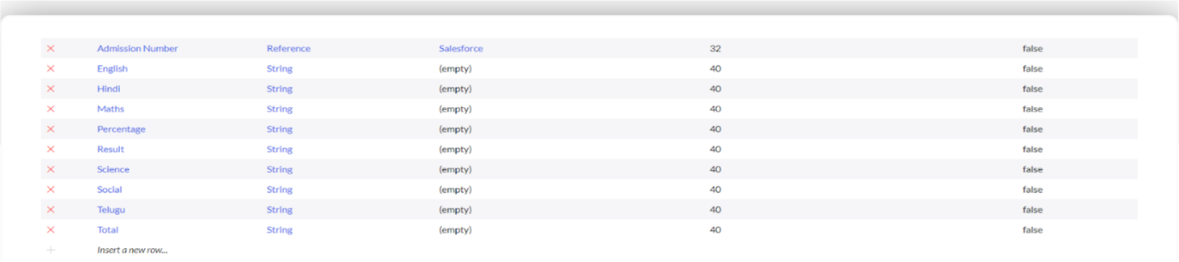
* Label	Admission	Application	Global
* Name	u_admission	Create module	☑
Extends table	Salesforce	Create mobile module	☑
		Add module to menu	Salesforce

Dictionary Entries

Column label	Type	Reference	Max length	Default value	Display

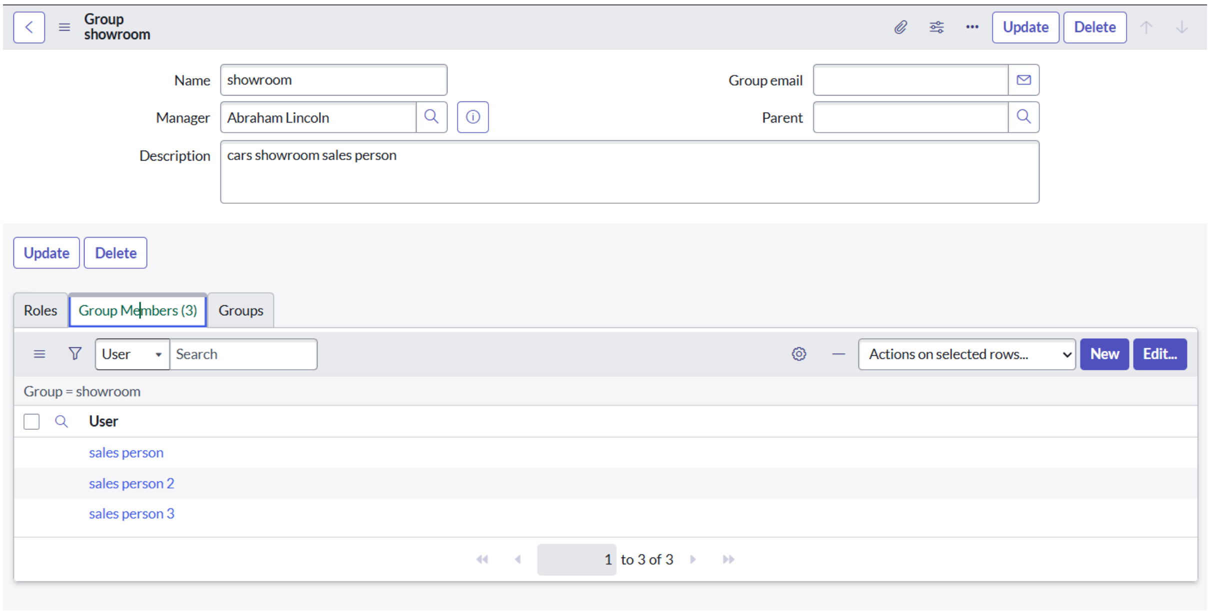
Step 5: Create Student Progress Table

- Create a Student Progress Table with Columns given.
- Select Add module to menu >> Salesforce.
- Create Fields as shown:



Admission Number	Reference	Salesforce	32	false
English	String	(empty)	40	false
Hindi	String	(empty)	40	false
Maths	String	(empty)	40	false
Percentage	String	(empty)	40	false
Result	String	(empty)	40	false
Science	String	(empty)	40	false
Social	String	(empty)	40	false
Telugu	String	(empty)	40	false
Total	String	(empty)	40	false

Insert a new row...



Group
showroom

Name	showroom	Group email	
Manager	Abraham Lincoln	Parent	
Description	cars showroom sales person		

Group = showroom

User
sales person
sales person 2
sales person 3

1 to 3 of 3

Step 6: Configuring Table form for Student Progress Table

- In the Student Progress Table Page, Click on Layout form.

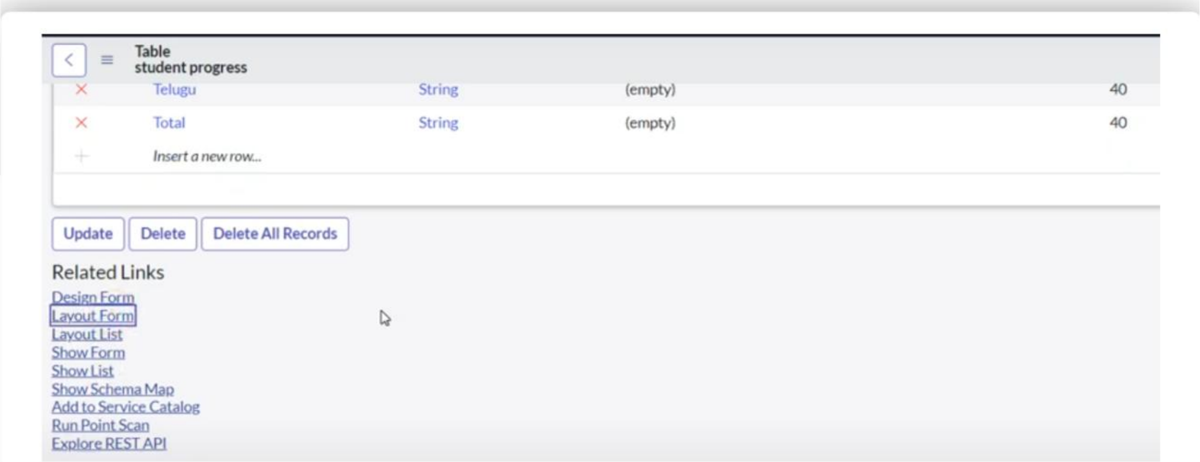
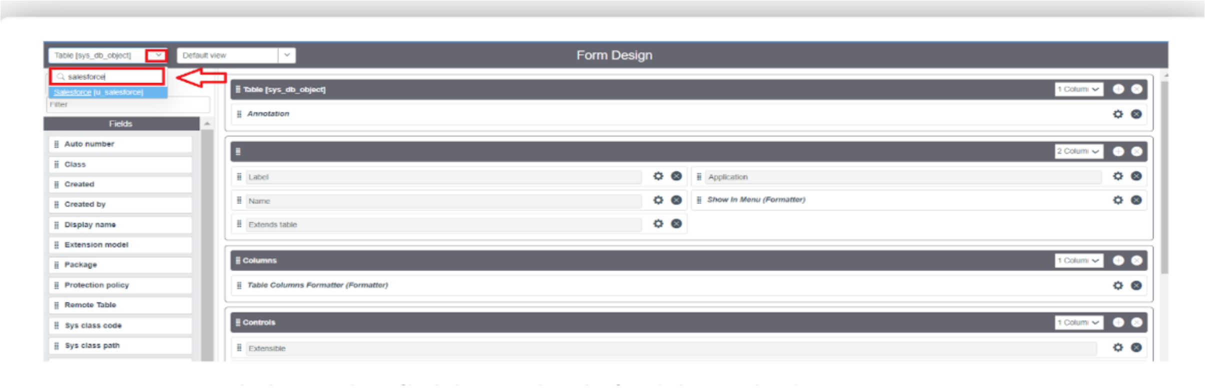


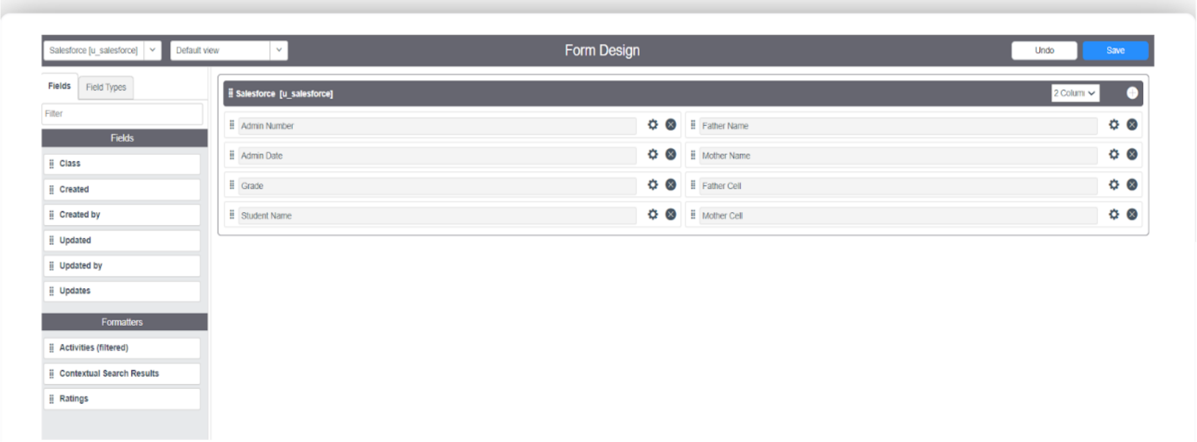
Table
student progress

Telugu	String	(empty)	40
Total	String	(empty)	40
Insert a new row...			

Step 7: Creating Form Design For Sales Force Table

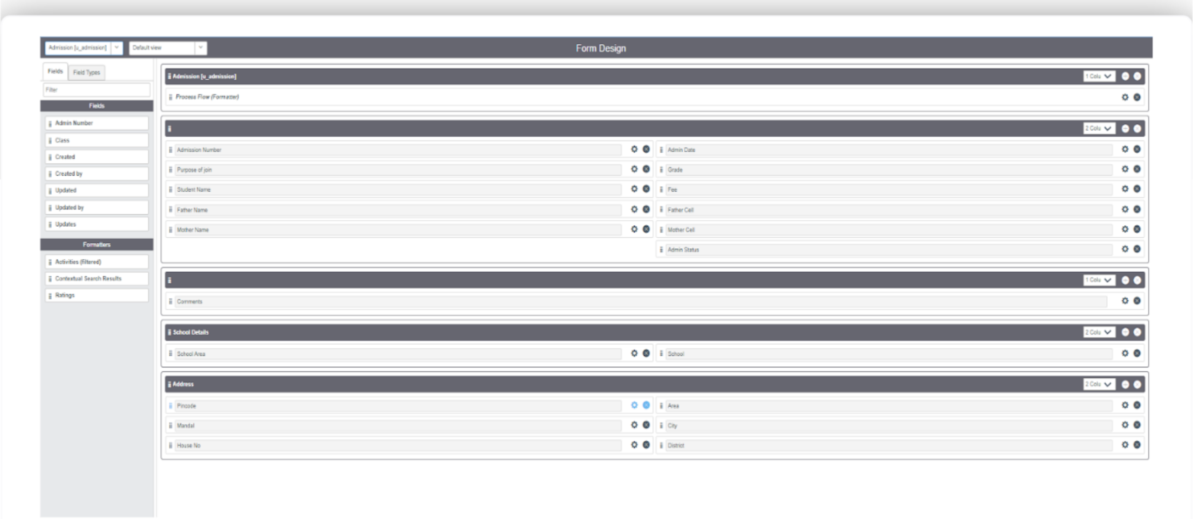
1. All >> System Definition >> Tables.
2. In Label Search for Salesforce and open.



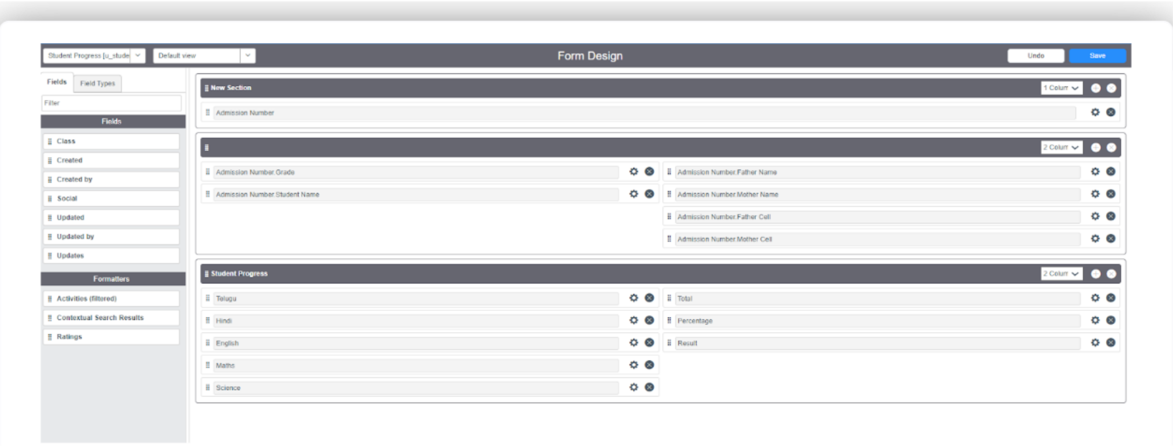


Task: Creating Form Design For Admission Table

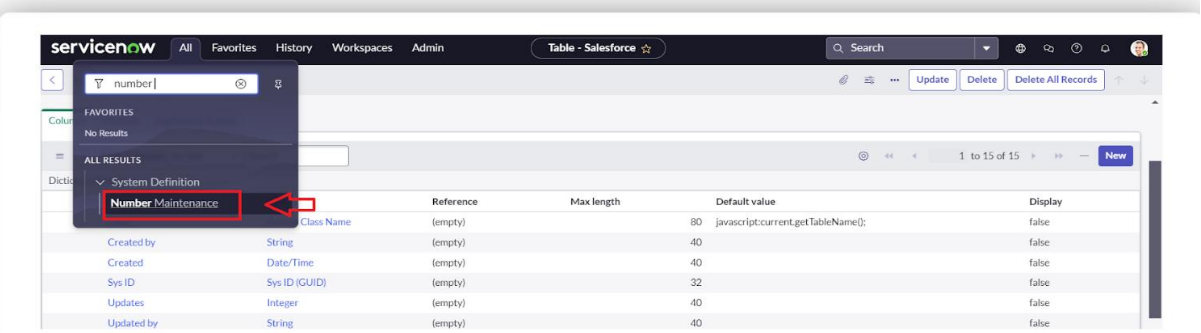
Follow the same steps as Activity1, Configure the fields as below and Save.



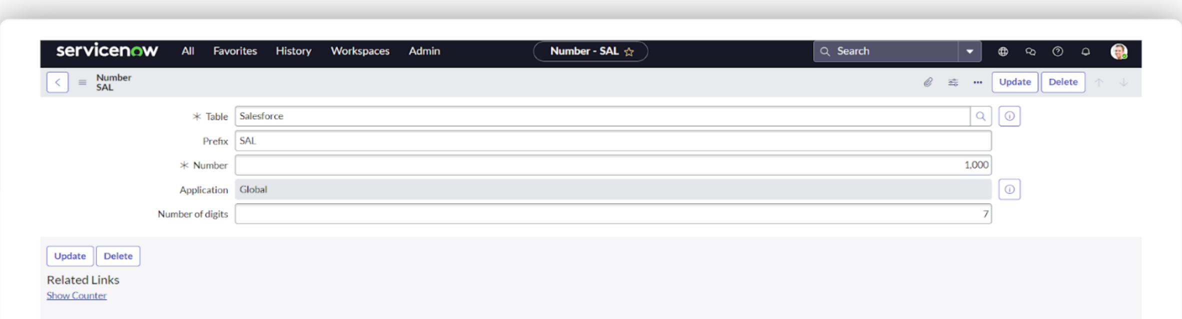
Task: Creating Form Design For Student Progress Table



- **Task: Creating Number Maintenance For Admin Number**
- All >> Number Maintenance >> New
- Fill the details >> Submit.

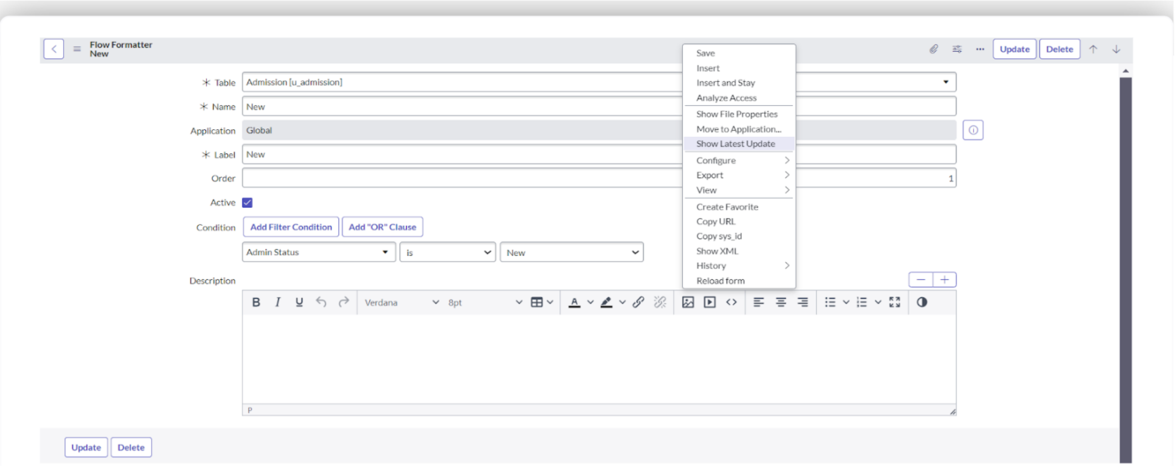


		Reference	Max length	Default value	Display
	Class Name	(empty)	80	javascript:current.getTableName();	false
Created by	String	(empty)	40		false
Created	Date/Time	(empty)	40		false
Sys ID	Sys ID (GUID)	(empty)	32		false
Updates	Integer	(empty)	40		false
Updated by	String	(empty)	40		false



* Table	Salesforce
Prefix	SAL
* Number	1,000
Application	Global
Number of digits	7

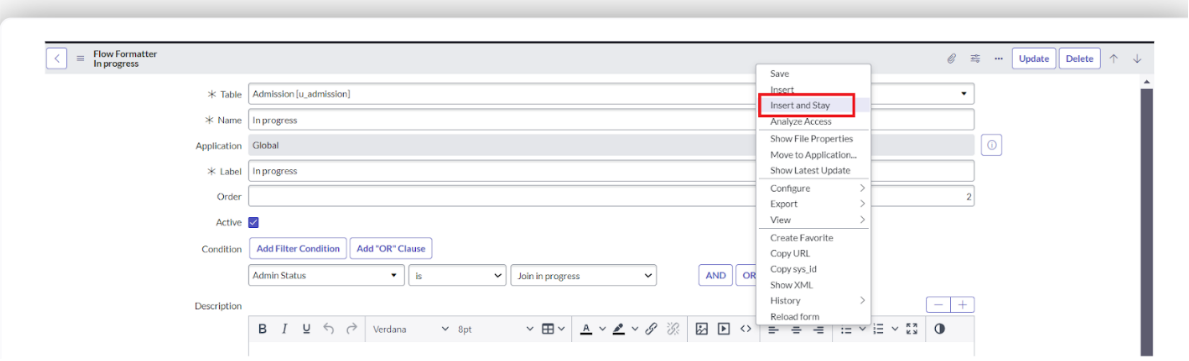
- **Notification: Creating Process Flow For Admission Table**
- All >> Process Flow>> New.
- Fill the Details as given Below



Flow Formatter
New

- Table: Admission [u_admission]
- Name: New
- Application: Global
- Label: New
- Order: 1
- Active: ☑
- Condition: Admin Status is New
- Description

- Right Click on toggle and click on the save .
- Replace the Name and Label as below and click on Insert on stay.



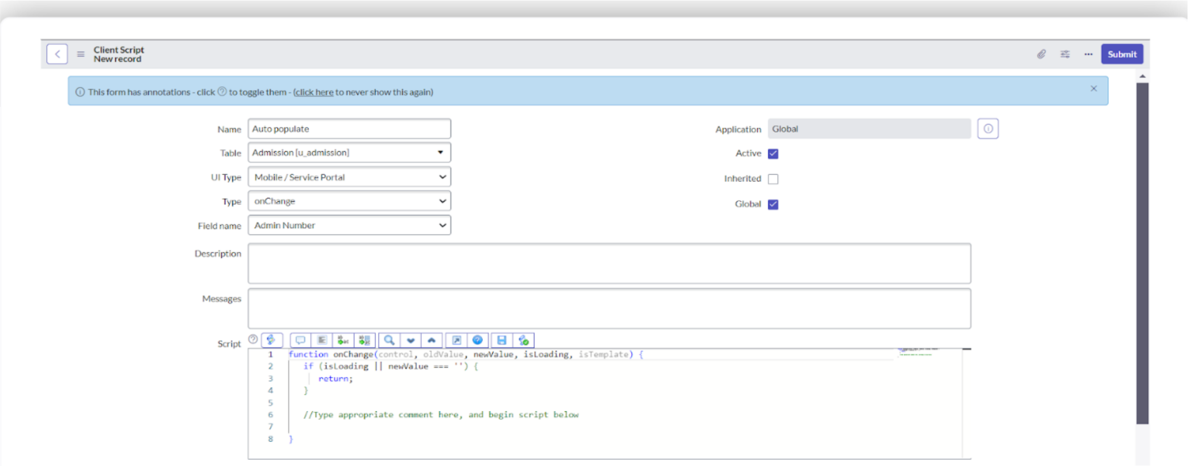
Flow Formatter
In progress

- Table: Admission [u_admission]
- Name: In progress
- Application: Global
- Label: In progress
- Order: 2
- Active: ☑
- Condition: Admin Status is Join in progress
- Description

- Replace the Name and Label in order and click on Insert on stay. Joined >> Rejected >> Rejoined >> Closed >> Cancelled.

- Order should be New >> InProgress >> Joined >> Rejected >> Rejoined >> Closed >> Cancelled.

- **Task: Client Script**
- All >> Client Scripts >> New.
- Fill the Details as given.



Client Script
New record

ⓘ This form has annotations - click ⓘ to toggle them - (click here to never show this again)

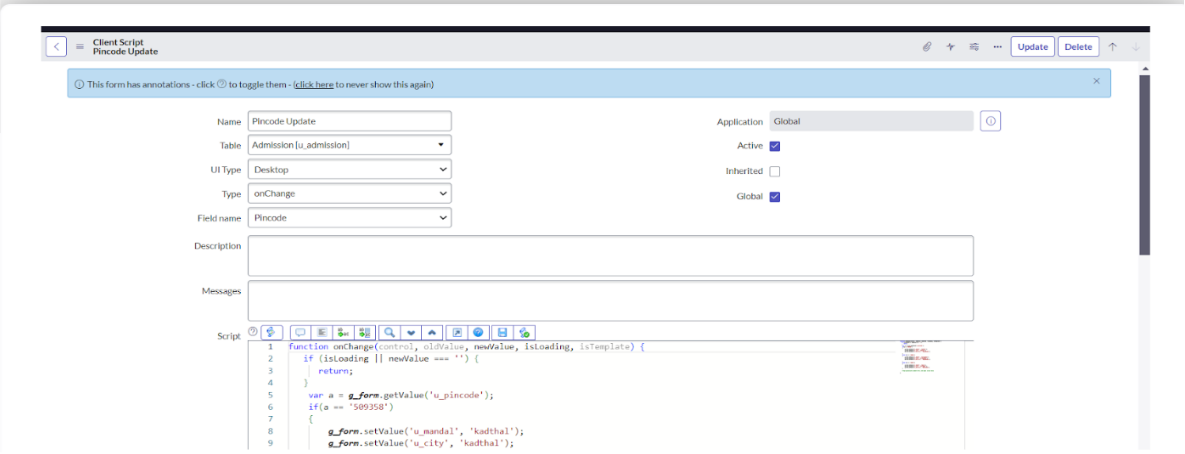
Name	Auto populate	Application	Global
Table	Admission [u_admission]	Active	☑
UI Type	Mobile / Service Portal	Inherited	□
Type	onChange	Global	☑
Field name	Admin Number		
Description			
Messages			

Script

```
function onChange(control, oldValue, newValue, isLoading, isTemplate) {
  if (isLoading || newValue === '') {
    return;
  }

  //Type appropriate comment here, and begin script below

}
```



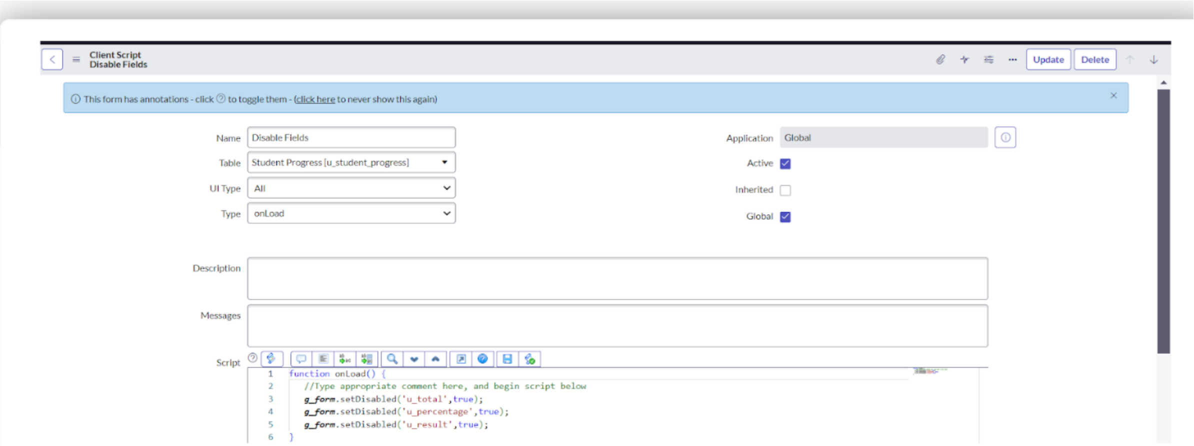
Client Script
Pincode Update

ⓘ This form has annotations - click ⓘ to toggle them - (click here to never show this again)

Name	Pincode Update	Application	Global
Table	Admission [u_admission]	Active	☑
UI Type	Desktop	Inherited	□
Type	onChange	Global	☑
Field name	Pincode		
Description			
Messages			

Script

```
function onChange(control, oldValue, newValue, isLoading, isTemplate) {
  if (isLoading || newValue === '') {
    return;
  }
   var a = g_form.getValue('u_pincode');
   if(a == '509358')
   {
      g_form.setValue('u_mandal', 'kadthal');
      g_form.setValue('u_city', 'kadthal');
```



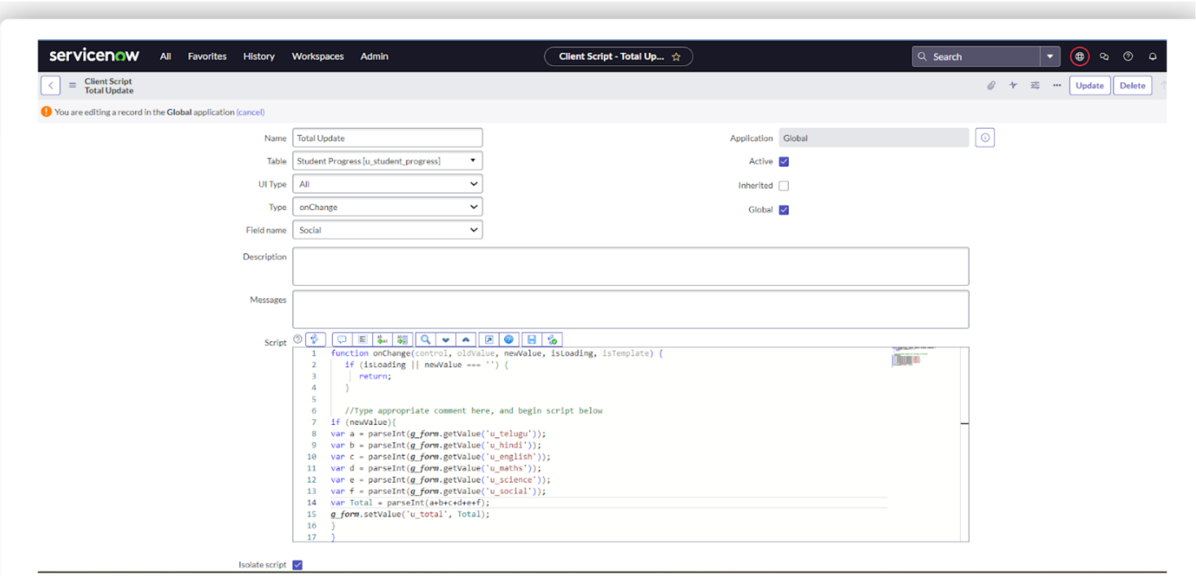
Client Script
Disable Fields

ⓘ This form has annotations - click ⓘ to toggle them - (click here to never show this again)

Name	Disable Fields	Application	Global
Table	Student Progress [u_student_progress]	Active	☑
UI Type	All	Inherited	□
Type	onLoad	Global	☑
Description			
Messages			

Script

```
function onLoad() {
   //Type appropriate comment here, and begin script below
   g_form.setDisabled('u_total',true);
   g_form.setDisabled('u_percentage',true);
   g_form.setDisabled('u_result',true);
}
```



Client Script
Total Update

You are editing a record in the Global application (cancel)

Name	Total Update	Application	Global
Table	Student Progress [u_student_progress]	Active	☑
UI Type	All	Inherited	□
Type	onChange	Global	☑
Field name	Social		

Description

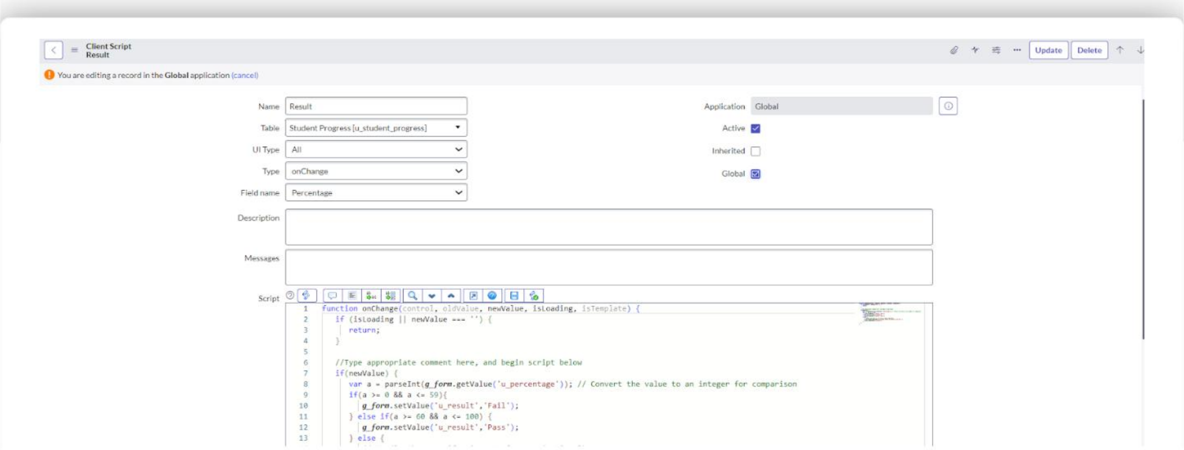
Messages

Script

```
function onChange(control, oldValue, newValue, isLoading, isTemplate) {
  if (isLoading || newValue === '') {
    return;
  }

  //Type appropriate comment here, and begin script below
if (newValue){
var a = parseInt(g_form.getValue('u_telugu'));
var b = parseInt(g_form.getValue('u_hindi'));
var c = parseInt(g_form.getValue('u_english'));
var d = parseInt(g_form.getValue('u_maths'));
var e = parseInt(g_form.getValue('u_science'));
var f = parseInt(g_form.getValue('u_social'));
var Total = parseInt(a+b+c+d+e+f);
g_form.setValue('u_total', Total);
}
}
```

Isolate script ☑



Client Script
Result

You are editing a record in the Global application (cancel)

Name	Result	Application	Global
Table	Student Progress [u_student_progress]	Active	☑
UI Type	All	Inherited	□
Type	onChange	Global	☑
Field name	Percentage		

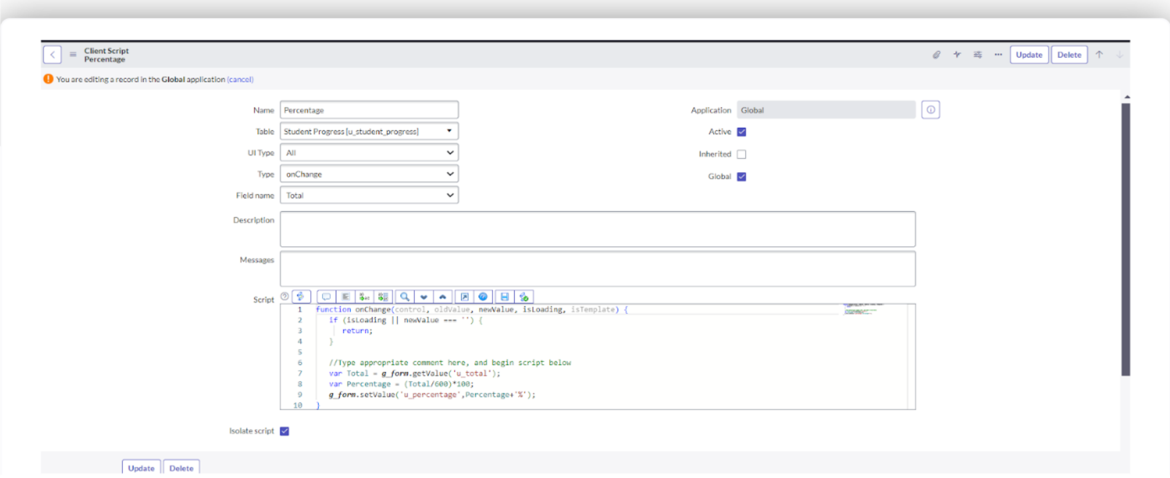
Description

Messages

Script

```
function onChange(control, oldValue, newValue, isLoading, isTemplate) {
  if (isLoading || newValue === '') {
    return;
  }

  //Type appropriate comment here, and begin script below
  if(newValue) {
    var a = parseInt(g_form.getValue('u_percentage')); // Convert the value to an integer for comparison
    if(a >= 0 && a <= 59){
      g_form.setValue('u_result','Fail');
    } else if(a >= 60 && a <= 100) {
      g_form.setValue('u_result','Pass');
    } else {
```



Client Script
Percentage

You are editing a record in the Global application (cancel)

Name	Percentage	Application	Global
Table	Student Progress [u_student_progress]	Active	☑
UI Type	All	Inherited	□
Type	onChange	Global	☑
Field name	Total		

Description

Messages

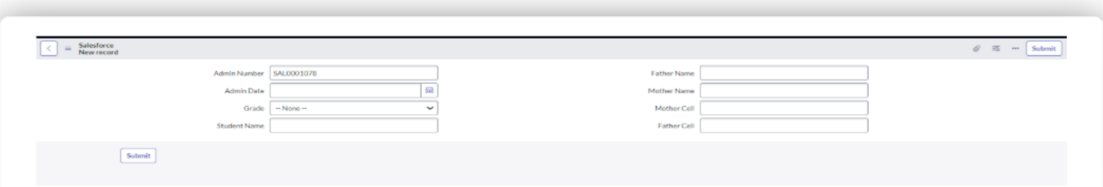
Script

```
function onChange(control, oldValue, newValue, isLoading, isTemplate) {
  if (isLoading || newValue === '') {
    return;
  }

  //Type appropriate comment here, and begin script below
  var Total = g_form.getValue('u_total');
  var Percentage = (Total/600)*100;
  g_form.setValue('u_percentage',Percentage+'%');
}
```

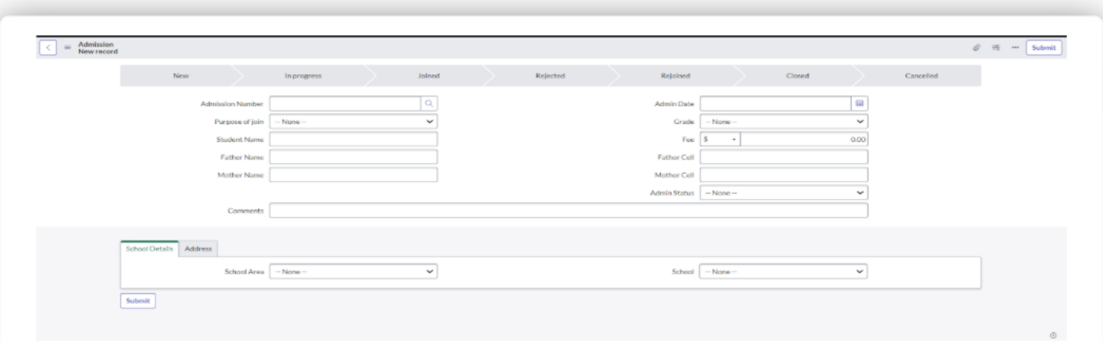
Isolate script ☑

Result:



Admin Number SAL0001078

Grade -- None --



Purpose of join -- None --

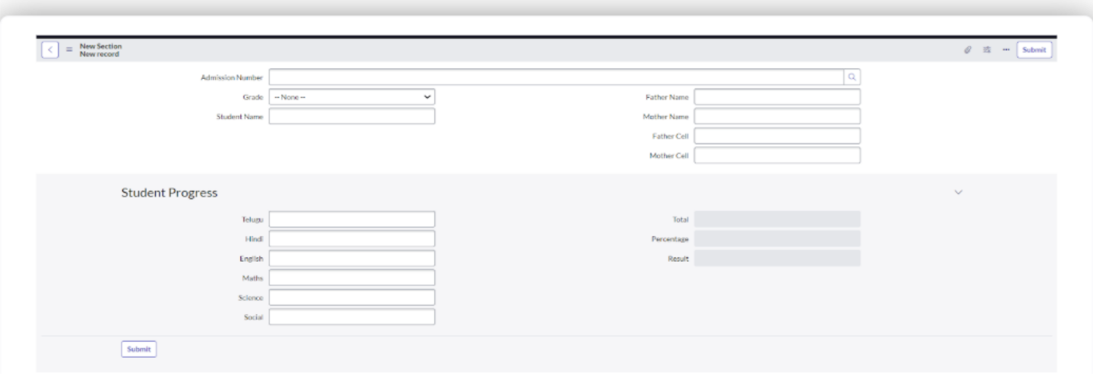
Grade -- None --

Fee $ 0.00

Admin Status -- None --

School Area -- None --

School -- None --



Grade -- None --

7. FUNCTIONAL AND PERFORMANCE TESTING

Performance Testing

Performance and Functional Testing involved checking every module and step in the workflow. Key validations included:

- **Catalog Display:** All items and categories visible under "Educational Services".
- **Form Accuracy:** Fields loaded correctly and accepted valid inputs.
- **Workflow Execution:** Requests routed to appropriate approvers with state tracking.
- **Email Delivery:** Notifications triggered instantly with correct content.

- **Data Consistency:** Entries correctly updated in u_admission and u_student_progress tables.

8. ADVANTAGEDISADVANTAGES

Advantages

- **Improved Efficiency:** Reduces manual work and speeds up request handling.
- **Error Reduction:** Validations prevent incorrect data entries.
- **Real-Time Notifications:** Keeps stakeholders updated at each stage.
- **Scalability:** Future academic services can be added easily without structural overhaul.

Limitations

- **Dependency on ServiceNow Expertise:** Requires administrators familiar with platform scripting and configuration.
- **Initial Complexity:** Workflow setup and portal tuning may take considerable time during the first phase.

9. CONCLUSION

The Educational Management System developed using ServiceNow provides an integrated, scalable, and efficient solution for managing key academic workflows such as admissions, progress tracking, and student support. It ensures structured communication, automated approvals, and transparent operations, enhancing the overall student and faculty experience.

10. FUTURE SCOPE

To make the system more robust and comprehensive, the following enhancements are proposed:

- **Feedback Module:** Allow students to rate and provide feedback on services or academic support.
- **Role-Based Dashboards:** Interactive dashboards for admin, teachers, and students with metrics and analytics.

- **Messaging Integration:** Notify students via WhatsApp or SMS in addition to emails.
- **Bulk Operations:** Enable mass course enrollment, attendance tracking, and batch promotions.
- **Payment Gateway Integration:** Allow students to pay application or tuition fees through the portal securely.