

# **Project**

## **Educational Organisation Using ServiceNow**

**Team ID: LTVIP2026TMIDS24894**

**Team Members: 04**

**Team Leader: Katta Sirisha**

**Team Member: Gangadhar Dakarmi**

**Team Member: J. Narasimha Swamy**

**Team Member: Gandu Kavya**

**Problem Statement:** In many educational organisations, daily operations such as admissions, student progress tracking, staff management, and communication are often handled manually or through different disconnected systems. This creates several challenges like delays in processes, lack of transparency for students and staff, and difficulties in managing large amounts of data. Without proper automation, tasks such as approvals, notifications, and report generation consume more time and effort. Moreover, the absence of a single digital platform leads to communication gaps between students, teachers, and administrators. These issues highlight the need for a centralized and automated system to make educational management more efficient and reliable.

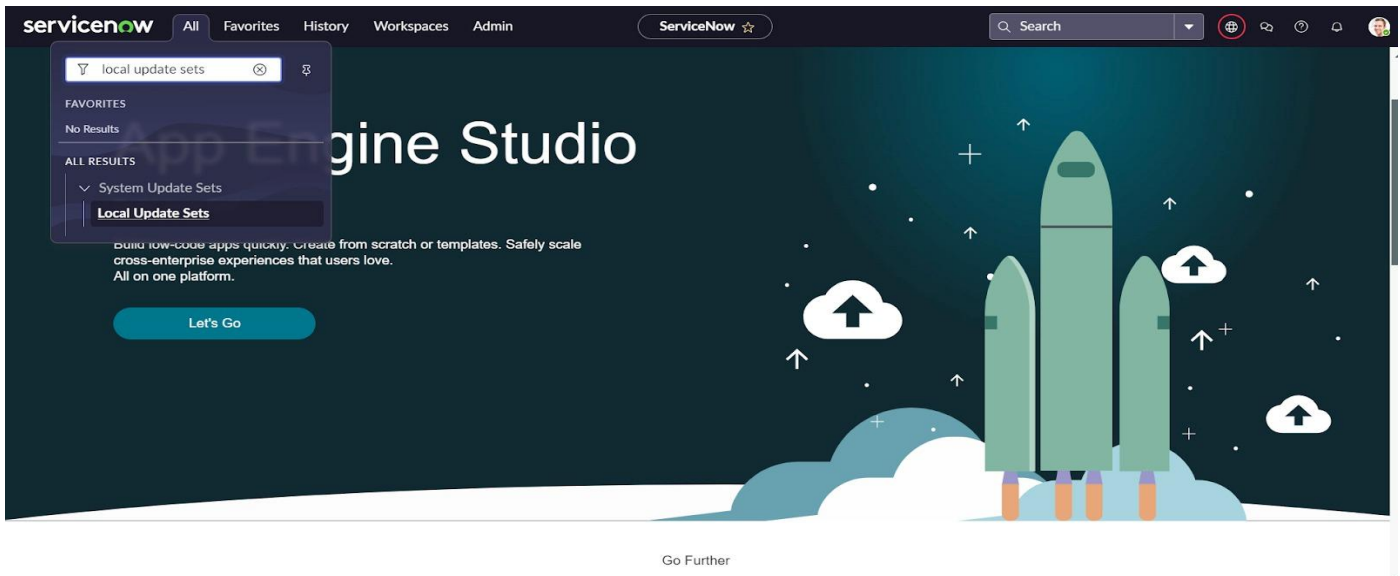
**Objective:** The project aims to develop a system for schools and colleges to manage admissions, student details, and academic progress, making the entire process quick, paperless, and user-friendly.

### **Skills:**

- ❖ ServiceNow Platform Knowledge
- ❖ Scripting (JavaScript)
- ❖ Database Concepts (RDBMS)
- ❖ Web Technologies (HTML, CSS, PHP/Node.js)
- ❖ Workflow Automation
- ❖ Problem-Solving
- ❖ Team Collaboration
- ❖ Project Presentation
- ❖ Analytical Thinking

# TASK INITIATION

## ■ Module 1: Creating an Update Set



**Step 1: Click on All >> Local update sets.**

The screenshot shows the ServiceNow Update Sets list. The 'New' button is highlighted in the top right corner. The table below lists the update sets.

	Name	Application	State	Installed from	Created	Created by	Parent	Batch Base
	Search	Search	Search	Search	Search	Search	Search	Search
	Default	App Engine Studio	In progress		2024-04-03 04:32:28	admin	(empty)	(empty)
	Default	Global	In progress		2023-10-06 15:26:30	system	(empty)	(empty)

**Step 2: Click on New**

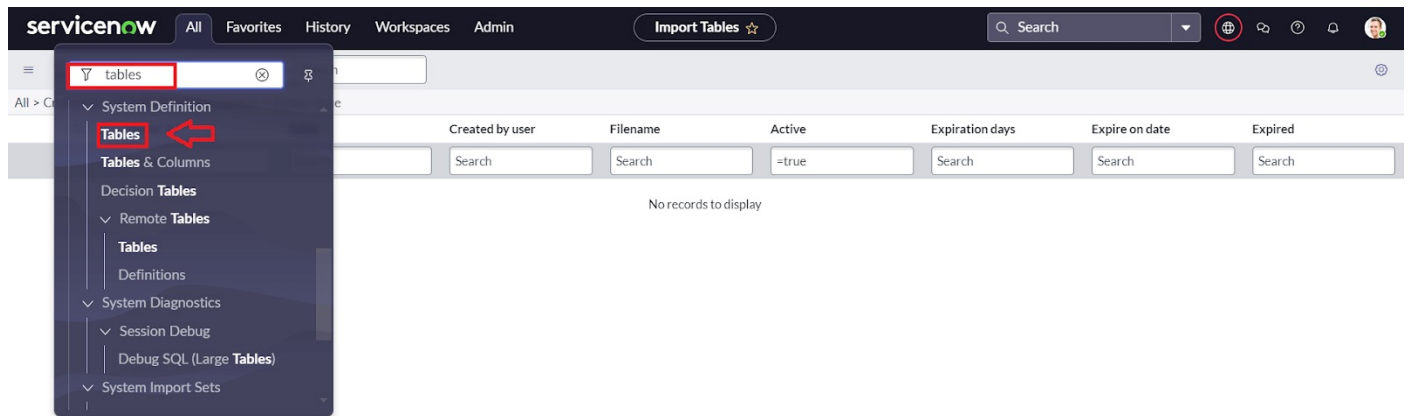
The screenshot shows the 'Create New Update Set' form. The 'Name' field is filled with 'New Update Set'. The 'Application' is set to 'Global'. The 'State' is 'In progress'. The 'Parent' field is empty. The 'Release date' field is empty. The 'Description' field is empty. The 'Submit' and 'Submit and Make Current' buttons are at the bottom.

**Step 3: Enter the Details Name: Educational Organisation >> Click on Submit and make Current.**

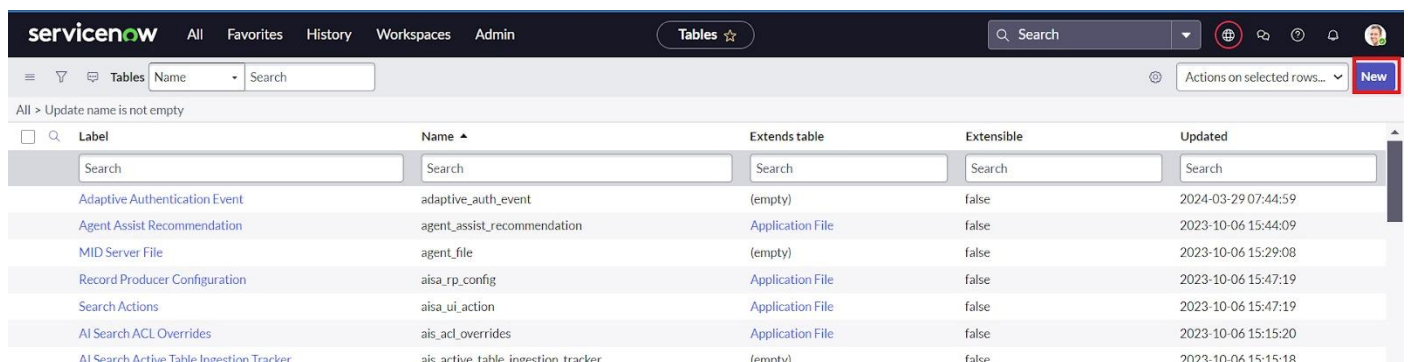
## ■ Module 2: Creating A Table

### Sub Module 1: Creating Salesforce Table.

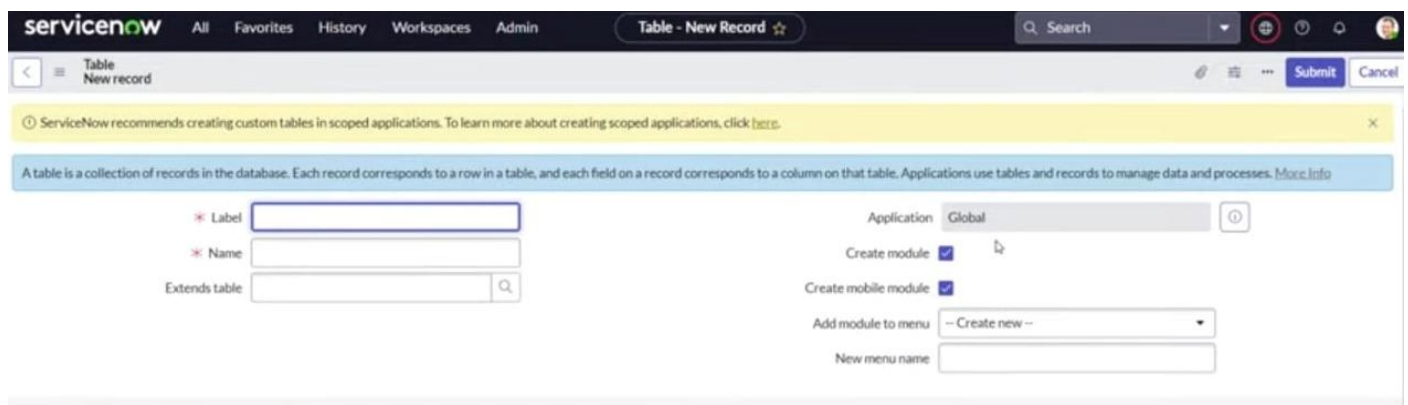
#### Step 1: All>>Tables



#### Step 2: Click On New



#### Step 3: Enter the Label (Anything you want): Salesforce >> Click on Name it will Automatically generate Api name.



#### Step 4: Create columns as given below, Double Click on Column label and Enter the Column labels and click on the tick mark >> Give Type as given.

① This form has annotations - click (?) to toggle them - ([click here](#) to never show this again)

\* Label

\* Name

Extends table

Q

Creaz

Ad

Columns

\* Controls

Application Access

≡

⌵

Table Columns

for text

Search

Dictionary Entries

	Column label	Type	Reference	Max length
<div>+</div>	<div></div>	<div><div>✓</div><div>✗</div></div>		

Submit

Cancel

**Step 5: For “Admin Number” Give Display as True and right click on the toggle bar on top >> save.**

Table  
Salesforce

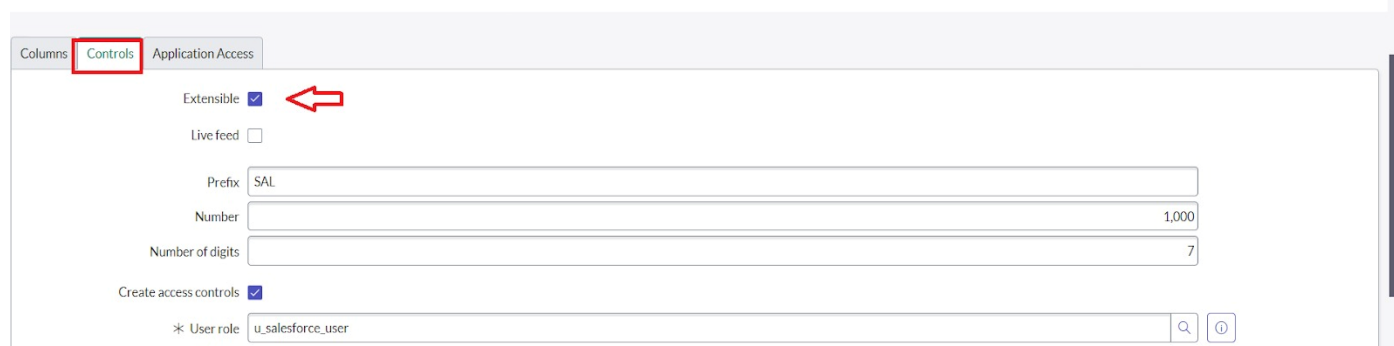
This record is in the **Global** application, but **Educational Organisation** is the current application. To edit this record click [here](#).

Admin Date	Date	(empty)	40	false
Admin Number	String	(empty)	40	javascript:getNextObjNumberPadded();
Father Cell	String	(empty)	40	false
Father Name	String	(empty)	40	false
Grade	Choice	(empty)	40	false
Mother Cell	String	(empty)	40	false
Mother Name	String	(empty)	40	false
Student Name	String	(empty)	40	false


The screenshot shows the Salesforce Table Editor interface. The 'Table Columns' tab is selected. A red arrow points to the 'Save' button in the top right corner. A dropdown menu is open from the 'Save' button, showing options like 'Analyze Access', 'Show File Properties', 'Move to Application...', 'Show Latest Update', 'Create Child Table', 'Show Dictionary Record', 'Configure', 'Export', 'View', 'Create Favorite', 'Copy URL', 'Copy sys\_id', 'Show XML', 'History', and 'Reload form'. The table below shows columns for 'Class', 'Created by', 'Created', 'Sys ID', 'Updates', 'Updated by', and 'Updated'.

Column label	Type	Reference	Max length	Display
Class	System Class Name	(empty)	80	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
Updated	Date/Time	(empty)	40	false

## Step 6: Click on controls >> Enable Extensible.



Columns: **Controls** Application Access

Extensible ☒ 

Live feed ☐

Prefix

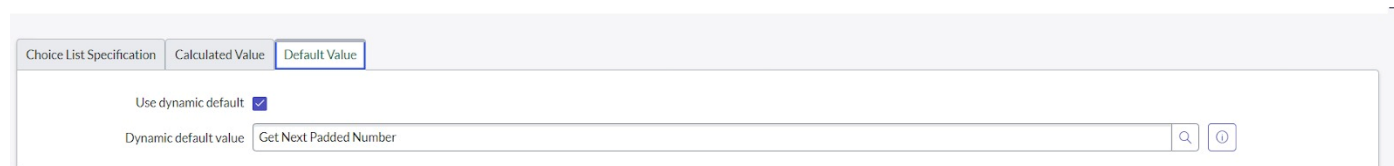
Number

Number of digits

Create access controls ☒

\* User role

## Step 7: Click on “Admin Number” column, In Related Links Click on Advanced View >> Default View (Enable Use dynamic default) >> select Get Next Padded Number in Dynamic default value >> Update.

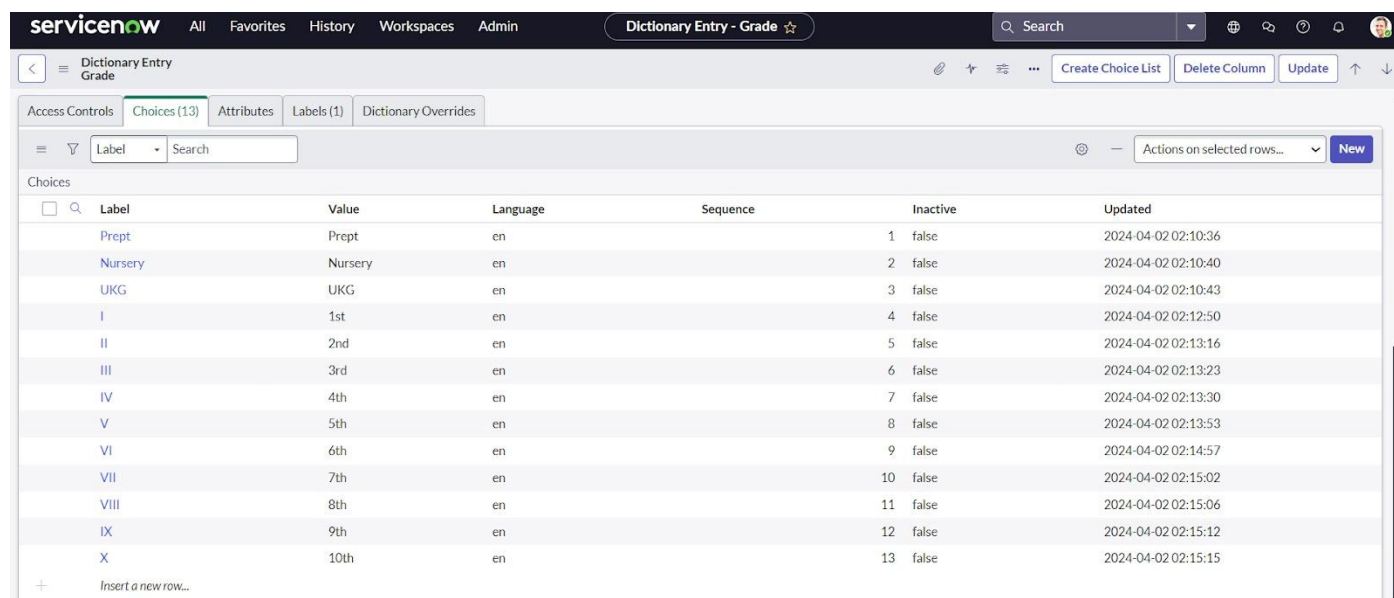


Choice List Specification Calculated Value **Default Value**

Use dynamic default ☒

Dynamic default value

## Step 8: Click on “Grade” Column >> Click on Choices and give Label, Value and Sequence as given below.



	Label	Value	Language	Sequence	Inactive	Updated
	Prept	Prept	en	1	false	2024-04-02 02:10:36
	Nursery	Nursery	en	2	false	2024-04-02 02:10:40
	UKG	UKG	en	3	false	2024-04-02 02:10:43
	I	1st	en	4	false	2024-04-02 02:12:50
	II	2nd	en	5	false	2024-04-02 02:13:16
	III	3rd	en	6	false	2024-04-02 02:13:23
	IV	4th	en	7	false	2024-04-02 02:13:30
	V	5th	en	8	false	2024-04-02 02:13:53
	VI	6th	en	9	false	2024-04-02 02:14:57
	VII	7th	en	10	false	2024-04-02 02:15:02
	VIII	8th	en	11	false	2024-04-02 02:15:06
	IX	9th	en	12	false	2024-04-02 02:15:12
	X	10th	en	13	false	2024-04-02 02:15:15
+	Insert a new row...					

## Sub Module 2: Creating Admission Table

**Step 1: Create an Admission Table with Columns given.**

**Step 2: Select Extends Table >> Salesforce and also Select Add module to menu >> Salesforce.**

**Step 3: 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  
\* Name: u\_admission  
Extends table: Salesforce  
Application: Global  
Create module: ☒  
Create mobile module: ☒  
Add module to menu: Salesforce

Columns | Controls | Application Access

Table Columns for text Search

Column label	Type	Reference	Max length	Default value	Display
--------------	------	-----------	------------	---------------	---------

Submit Cancel

Table Admission

Columns | Controls | Application Access

Table Columns for text Search

Column label	Type	Reference	Max length	Default value	Display
Sys ID	Sys ID (GUID)	(empty)	32		false
Admin Status	Choice	(empty)	40		false
Admission Number	Reference	Salesforce	32		false
Area	String	(empty)	40		false
City	String	(empty)	40		false
Comments	String (Full UTF-8)	(empty)	255		false
District	String	(empty)	40		false
Fee	Price	(empty)	20		false
House No	String	(empty)	40		false
Mandal	String	(empty)	40		false
Pincode	Choice	(empty)	40		false
Purpose of join	Choice	(empty)	40		false
School	Choice	(empty)	40		false
School Area	Choice	(empty)	40		false
Class	System Class Name	(empty)	80	javascript:current.getTable();	false

**Step 4: Create choice for Admin Status as:**





## Sub Module 3: Creating Student Progress Table

**Step 1: Create a Student Progress Table with Columns given.**

**Step 2: Select Add module to menu >> Salesforce.**

**Step 3: 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...				



## ■ Module 3: Form Layout

**Step 1: 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...			

Update Delete Delete All Records

Related Links

- [Design Form](#)
- [Layout Form](#)
- [Layout List](#)
- [Show Form](#)
- [Show List](#)
- [Show Schema Map](#)
- [Add to Service Catalog](#)
- [Run Point Scan](#)
- [Explore REST API](#)

**Step 2: Click on Admission Number [+].**

Configuring Table form

Available

- Admission Number [+]
- Created
- Created by
- Updated
- Updated by
- Updates
- [- begin\_split -]
- [- split -]
- [- end\_split -]
- \* Annotation
- \* Chart
- Activities (filtered)
- Contextual Search Results
- Ratings
- Attachments
- Goal relationships

Selected

- [- begin\_split -]
- Admission Number
- Hindi
- English
- Telugu
- Science
- [- split -]
- Total
- Average
- Social
- Maths
- [- end\_split -]

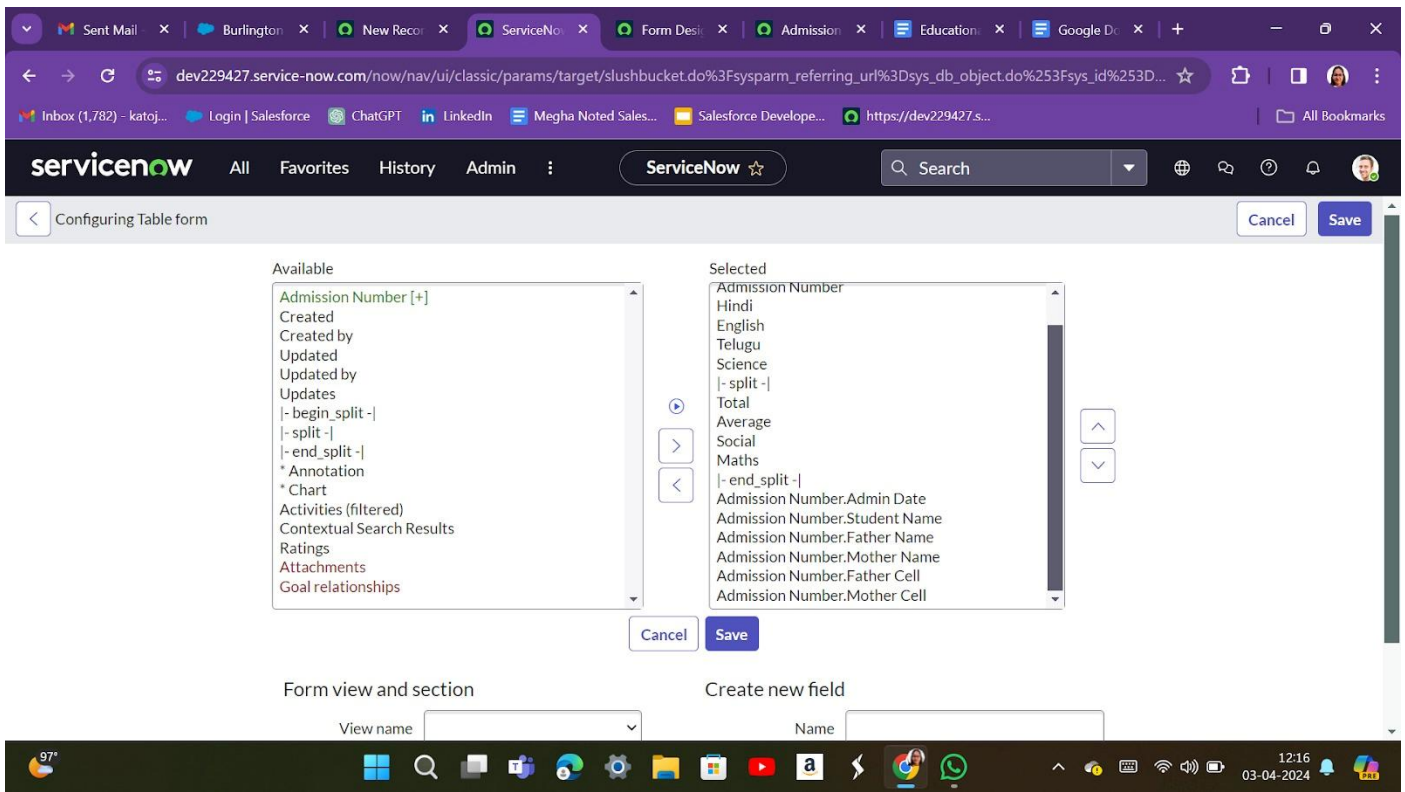
Form view and section

View name: Default view

Create new field

Name:

**Step 3: Select below Admission Number fields in Available side and send it to selected side as below >> save.**

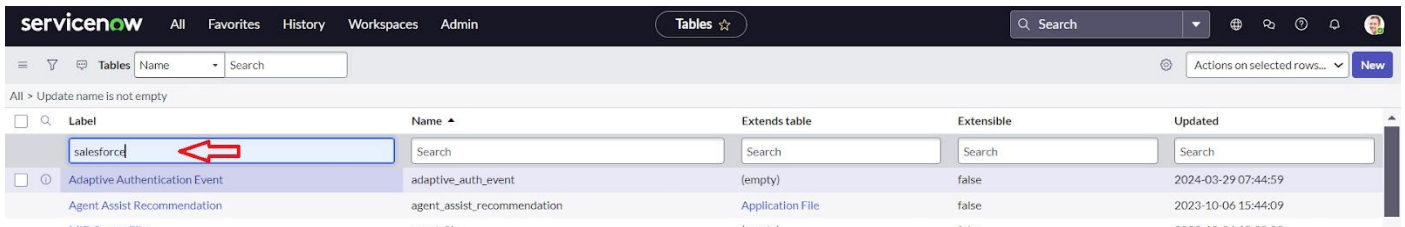


## - Module 4: Form Layout**

### **Sub Module 1: Creating Form Design for Salesforce Table**

**Step 1: All >> System Definition >> Tables.**

**Step 2: In Label Search for Salesforce and open.**



## Step 3: Right Click on top Toggle >> Configure >> Form Design

The screenshot shows the ServiceNow interface for configuring a table named 'Salesforce'. The top navigation bar includes 'servicenow', 'All', 'Favorites', 'History', 'Workspaces', and 'Admin'. The main header shows 'Table - Salesforce' with a search bar and buttons for 'Update', 'Delete', and 'Delete All Records'. A right-click context menu is open over the top toggle, with 'Configure' selected, leading to a sub-menu where 'Form Design' is highlighted. A red arrow points to 'Form Design'.

Column label	Type	Reference	Max length	Default	Display
Class	System Class Name	(empty)	80	javascript:nextObjNumberPadded();	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
Updated	Date/Time	(empty)	40		false
Admin Date	Date	(empty)	40		false
Admin Number	String	(empty)	40	javascript:nextObjNumberPadded();	true

## Step 4: In drop down select Salesforce(u\_salesforce).

The screenshot shows the 'Form Design' interface. The 'Table [sys\_db\_object]' dropdown is open, showing 'Salesforce [u\_salesforce]' selected. A red arrow points to the selected item.

Fields:

- Auto number
- Class
- Created
- Created by
- Display name
- Extension model
- Package
- Protection policy
- Remote Table
- Sys class code
- Sys class path

Form Design:

- Table [sys\_db\_object]
- Annotation
- Label
- Name
- Extends table
- Application
- Show In Menu (Formatter)
- Columns
- Table Columns Formatter (Formatter)
- Controls
- Extensible

## Step 5: Drag and drop the fields to the left side as below.

The screenshot shows the 'Form Design' interface for the 'Salesforce [u\_salesforce]' table. The 'Fields' list on the left contains 'Admin Number', 'Admin Date', 'Grade', and 'Student Name'. The 'Form Design' area on the right shows these fields being dragged into the form layout.

Fields:

- Admin Number
- Admin Date
- Grade
- Student Name

Form Design:

- Admin Number
- Admin Date
- Grade
- Student Name
- Father Name
- Mother Name
- Father Cell
- Mother Cell

## Step 6: Save

### Sub Module 2: Creating Form Design for Admission Table

**Step 1: Follow the same steps as Activity 1, Configure the fields as below and save.**

The screenshot shows the 'Form Design' interface for the 'Admission [u\_admission]' table. The interface is divided into a left sidebar and a main design area. The sidebar contains a 'Fields' tab with a list of fields: Admin Number, Class, Created, Created by, Updated, Updated by, and Updates. Below this is a 'Formatters' section with options like Activities (filtered), Contextual Search Results, and Ratings. The main design area shows a form layout with sections: 'Admission [u\_admission]' (1 Column), 'Process Flow (Formatter)' (1 Column), 'Admin Details' (2 Columns) containing fields like Admission Number, Purpose of join, Student Name, Father Name, Mother Name, Admin Date, Grade, Fee, Father Cell, Mother Cell, and Admin Status, 'Comments' (1 Column), 'School Details' (2 Columns) with School Area and School, and 'Address' (2 Columns) with Pincode, Mandal, City, and District.

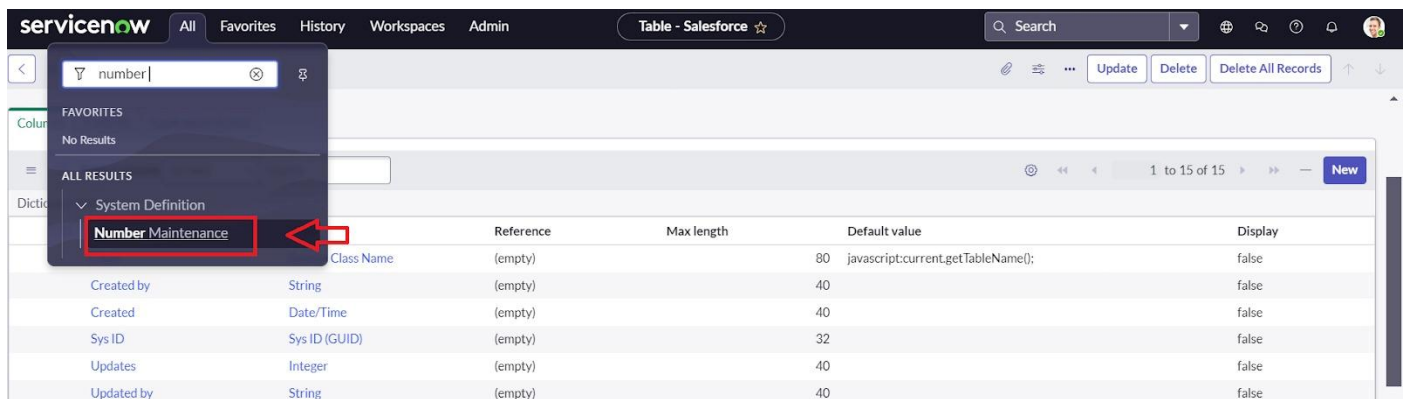
### Sub Module 3: Creating Form Design for Student progress Table

**Step 1: Follow the same steps as Activity1, Configure the fields as below and save.**

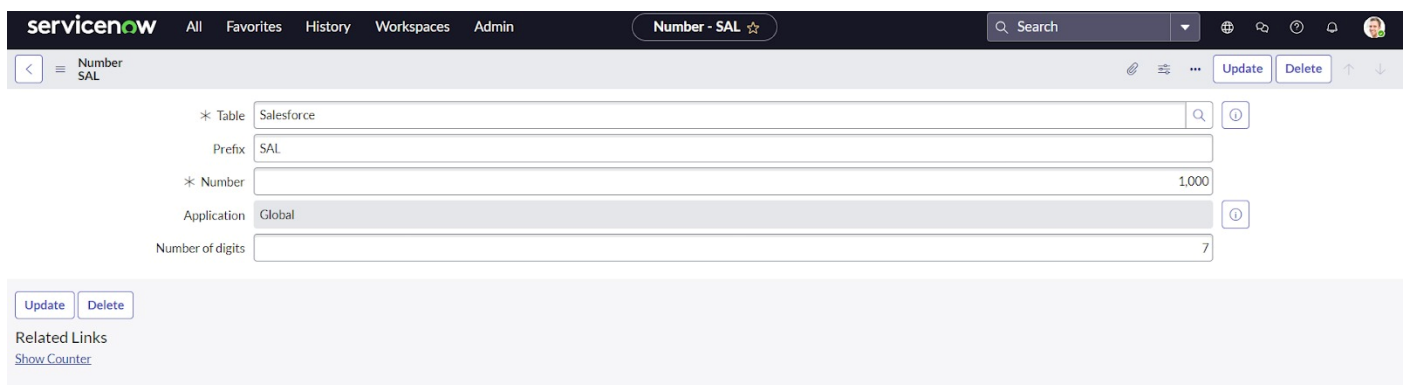
The screenshot shows the 'Form Design' interface for the 'Student Progress [u\_stude]' table. The interface is divided into a left sidebar and a main design area. The sidebar contains a 'Fields' tab with a list of fields: Class, Created, Created by, Social, Updated, Updated by, and Updates. Below this is a 'Formatters' section with options like Activities (filtered), Contextual Search Results, and Ratings. The main design area shows a form layout with sections: 'New Section' (1 Column) containing Admission Number, 'Student Progress' (2 Columns) containing fields like Admission Number Grade, Admission Number Father Name, Admission Number Mother Name, Admission Number Father Cell, Admission Number Mother Cell, Tolugu, Hindi, English, Maths, Science, Total, Percentage, and Result.

## ■ Module 5: Number Maintenance

### Step 1: All >> Number Maintenance >> New



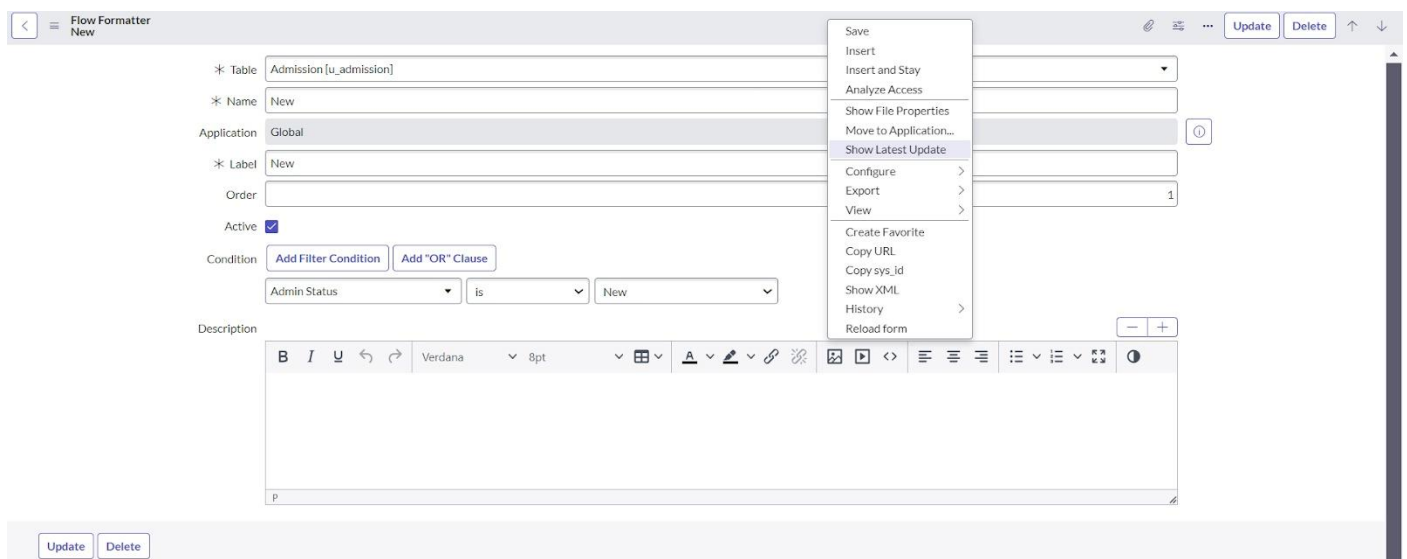
### Step 2: Fill the details >> Submit.



## ■ Module 6: Process Flow

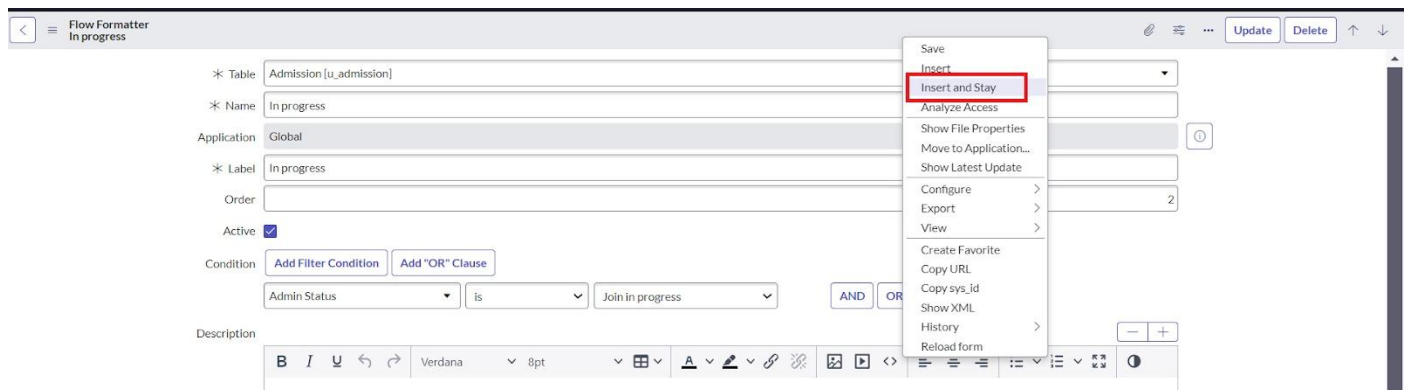
### Step 1: All >> Process Flow>> New.

### Step 2: Fill the Details as given Below



**Step 3: Right Click on toggle and click on the save.**

**Step 4: Replace the Name and Label as below and click on Insert on stay.**



**Step 5: Replace the Name and Label in order and click on Insert on stay.**

**(Joined >> Rejected >> Rejoined >> Closed >> Cancelled.)**

**Step 6: Order should be New >> InProgress >> Joined >> Rejected >> Rejoined >> Closed >> Cancelled.**

## ▪ **Module 7: Client Script**

### **Sub Module 1: Creating “Auto populate” Client Scripts for Admission Table**

**Step 1: All >> Client Scripts >> New.**

**Step 2: Fill the Details as given.**

**Step 3: Write the Code as below, Enable Isolate script and save.**

Client Script  
New record

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

Name:

Table:

UI Type:

Type:

Field name:

Application:

Active: ☒

Inherited: ☐

Global: ☒

Description:

Messages:

Script

```

1 function onChange(control, oldValue, newValue, isLoading, isTemplate) {
2   if (isLoading || newValue === '') {
3     return;
4   }
5   //Type appropriate comment here, and begin script below
6
7
8 }

```

**function onChange(control, oldValue, newValue, isLoading, isTemplate) {**

**if (isLoading || newValue === '') {**

**return;**

**}//Type appropriate comment here, and begin script below**

**var a = g\_form.getReference('u\_admission\_number');**

**g\_form.setValue('u\_admin\_date',a.u\_admin\_date);**

**g\_form.setValue('u\_grade',a.u\_grade);**

**g\_form.setValue('u\_student\_name',a.u\_student\_name);**

**g\_form.setValue('u\_father\_name',a.u\_father\_name);**

**g\_form.setValue('u\_mother\_name',a.u\_mother\_name);**

**g\_form.setValue('u\_father\_cell',a.u\_father\_cell);**

**g\_form.setValue('u\_mother\_cell',a.u\_mother\_cell);**

**g\_form.setDisabled('u\_admin\_date',a.u\_admin\_date);**

**g\_form.setDisabled('u\_grade',a.u\_grade);**

**g\_form.setDisabled('u\_student\_name',a.u\_student\_name);**



```

g_form.setDisabled('u_father_name',a.u_father_name);
g_form.setDisabled('u_mother_name',a.u_mother_name);
g_form.setDisabled('u_father_cell',a.u_father_cell);
g_form.setDisabled('u_mother_cell',a.u_mother_cell);
}

```

**Note: Make sure the Field names should be the same as you created.**

## Sub Module 2: Creating “Pin code Update” Client Scripts for Admission Table

**Step 1: Fill the Details as given.**

The screenshot shows the 'Client Script' configuration window for a script named 'Pincode Update'. The configuration includes the following fields:

- Name:** Pincode Update
- Table:** Admission [u\_admission]
- UI Type:** Desktop
- Type:** onChange
- Field name:** Pincode
- Application:** Global
- Active:** ☒
- Inherited:** ☐
- Global:** ☒
- Description:** (Empty text area)
- Messages:** (Empty text area)
- Script:**

```

1 function onChange(control, oldValue, newValue, isLoading, isTemplate) {
2   if (isLoading || newValue === '') {
3     return;
4   }
5   var a = g_form.getValue('u_pincode');
6   if(a == '509358')
7   {
8     g_form.setValue('u_mandal', 'kadthal');
9     g_form.setValue('u_city', 'kadthal');

```

**Step 2: Write the Code as below, Enable Isolate script and save.**

```

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');  
g_form.setValue('u_district', 'RangaReddy');  
}  
else if(a == '500081')  
{  
g_form.setValue('u_mandal', 'karmanghat');  
g_form.setValue('u_city', 'karmanghat');  
g_form.setValue('u_district', 'RangaReddy');  
}  
else if(a == '500079')  
{  
g_form.setValue('u_mandal', 'Abids');  
g_form.setValue('u_city', 'AsifNagar');  
g_form.setValue('u_district', 'Hyderabad');  
} //Type appropriate comment here, and begin script below  
}
```

### **Sub Module 3: Creating “Disable Fields” Client Scripts for Student progress Table**

**Step 1: Fill the Details as given.**

Client Script  
Disable Fields

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

Name: Disable Fields

Table: Student Progress [u\_student\_progress]

UI Type: All

Type: onLoad

Application: Global

Active: ☒

Inherited: ☐

Global: ☒

Description:

Messages:

Script

```
1 function onload() {  
2     //Type appropriate comment here, and begin script below  
3     g_form.setDisabled('u_total',true);  
4     g_form.setDisabled('u_percentage',true);  
5     g_form.setDisabled('u_result',true);  
6 }
```

**Step 2: Write the Code as below, Enable Isolate script and save.**  
**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);**

**}**

## **Sub Module 4: Creating “Total Update” Client Scripts for Student progress Table**

**Step 1: Fill the Details as given.**

**Step 2: Write the Code as below, Enable Isolate script and save.**

**servicenow** All Favorites History Workspaces Admin Client Script - Total Up...

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	<input checked="" type="checkbox"/>
UI Type	All	Inherited	<input type="checkbox"/>
Type	onChange	Global	<input checked="" type="checkbox"/>
Field name	Social		
Description			
Messages			
Script	<pre>1 function onChange(control, oldValue, newValue, isLoading, isTemplate) { 2   if (isLoading    newValue === '') { 3     return; 4   } 5 6   //Type appropriate comment here, and begin script below 7   if (newValue){ 8     var a = parseInt(g_form.getValue('u_telugu')); 9     var b = parseInt(g_form.getValue('u_hindi')); 10    var c = parseInt(g_form.getValue('u_english')); 11    var d = parseInt(g_form.getValue('u_maths')); 12    var e = parseInt(g_form.getValue('u_science')); 13    var f = parseInt(g_form.getValue('u_social')); 14    var Total = parseInt(a+b+c+d+e+f); 15    g_form.setValue('u_total', Total); 16  } 17 }</pre>		

Isolate 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);  
    }  
}
```

## Sub Module 5: Creating “Result” Client Scripts for Student progress Table

### Step 1: Fill the Details as given.

The screenshot shows the 'Client Script' configuration window for the 'Result' field. The 'Name' is 'Result', 'Table' is 'Student Progress [u\_student\_progress]', 'UI Type' is 'All', 'Type' is 'onChange', and 'Field name' is 'Percentage'. The 'Application' is set to 'Global', 'Active' is checked, 'Inherited' is unchecked, and 'Global' is checked. The 'Description' and 'Messages' fields are empty. The 'Script' area contains the following code:

```
1 function onChange(control, oldValue, newValue, isLoading, isTemplate) {  
2   if (isLoading || newValue === '') {  
3     return;  
4   }  
5  
6   //Type appropriate comment here, and begin script below  
7   if(newValue) {  
8     var a = parseInt(g_form.getValue('u_percentage')); // Convert the value to an integer for comparison  
9     if(a >= 0 && a <= 59){  
10      g_form.setValue('u_result','Fail');  
11    } else if(a >= 60 && a <= 100) {  
12      g_form.setValue('u_result','Pass');  
13    } else {  
14      // ...  
15    }  
16  }  
17 }
```

### Step 2: Write the Code as below, Enable Isolate script and save.

**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 {

    // Handle the case if a is out of range (optional)

    g_form.addErrorMessage('Percentage should be between 0
and 100.');
```

```

    g_form.clearValue('u_result');

}

}

}

```

## Sub Module 6: Creating “Percentage” Client Scripts for Student progress Table

### Step 1: Fill the Details as given.

The screenshot shows the Salesforce Client Script configuration interface. The top bar indicates 'Client Script Percentage' and 'You are editing a record in the Global application (cancel)'. The main form contains the following fields:

- Name:** Percentage
- Table:** Student Progress [u\_student\_progress]
- UI Type:** All
- Type:** onChange
- Field name:** Total
- Application:** Global
- Active:** ☒
- Inherited:** ☐
- Global:** ☒
- Description:** (Empty text area)
- Messages:** (Empty text area)
- Script:**

```

1 function onChange(control, oldValue, newValue, isLoading, isTemplate) {
2   if (isLoading || newValue === '') {
3     return;
4   }
5
6   //Type appropriate comment here, and begin script below
7   var Total = g_form.getValue('u_total');
8   var Percentage = (Total/600)*100;
9   g_form.setValue('u_percentage',Percentage+'%');
10 }

```
- Isolate script:** ☒

At the bottom, there are 'Update' and 'Delete' buttons.

### Step 2: Write the Code as below, Enable Isolate script and Save.

```

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+'%');

}

```

## Outcome:

Salesforce New record

Admin Number

Admin Date

Grade

Student Name

Father Name

Mother Name

Mother Cell

Father Cell

Admission New record

New In progress Joined Rejected Rejoined Closed Cancelled

Admission Number

Purpose of join

Student Name

Father Name

Mother Name

Comments

Admin Date

Grade

Fee \$

Father Cell

Mother Cell

Admin Status

School Details Address

School Area

School

New Section New record

Admission Number

Grade

Student Name

Father Name

Mother Name

Father Cell

Mother Cell

Student Progress

Telugu

Hindi

English

Maths

Science

Social

Total

Percentage

Result



## **Conclusion:**

**The project “Educational Organisation Using ServiceNow” successfully demonstrates how digital automation can simplify and improve the management of educational institutions. By using ServiceNow, we were able to centralize student data, automate workflows, and create an efficient platform for admissions, student progress tracking, and communication between staff and students. This solution reduces manual effort, saves time, and ensures accuracy in handling institutional processes. Overall, the project highlights the importance of using modern IT service management tools to build a smarter and more reliable educational system.**

