Project

Educational Organisation Using ServiceNow

Team ID:NM2025TMID13584

Team Members: 04

Team Leader: DIVYADHARSHINI. K

Team Member: DHANUSHIYA .M

Team Member: SUNIL .A

Team Member: RAJASEKAR .J

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.**

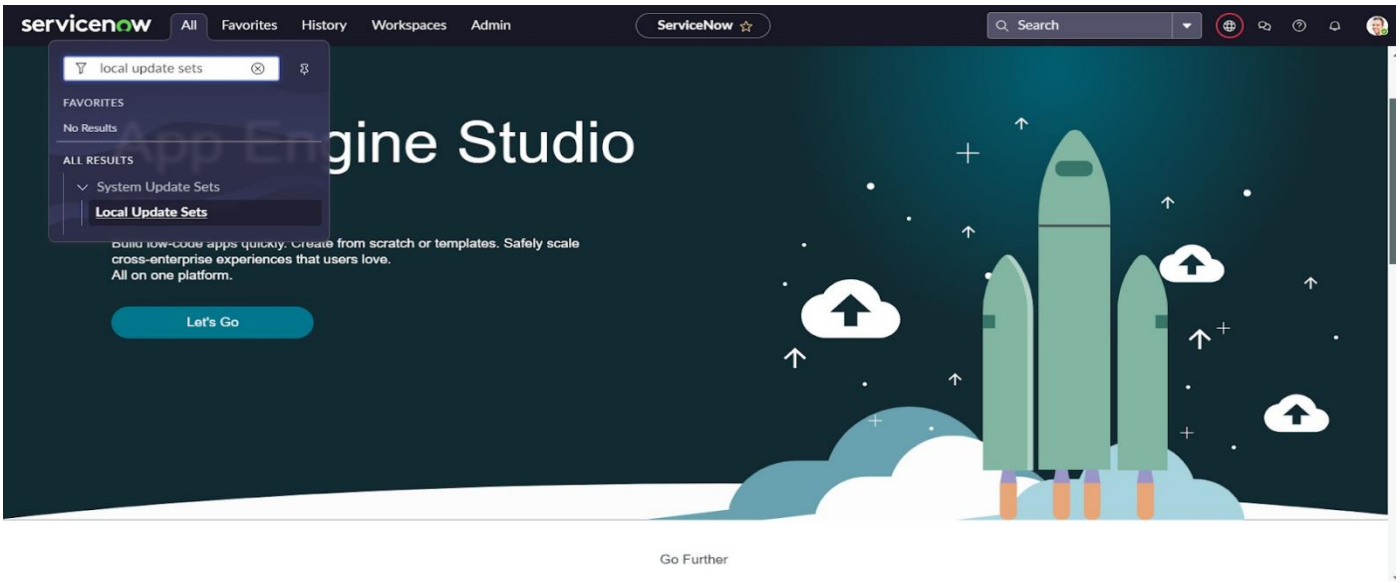
Objective: **The project aims to develop a system for schools and colleges to manage admissions, student details, and academic progress, making the entire process quick, paperless, and user-friendly.**

Skills:

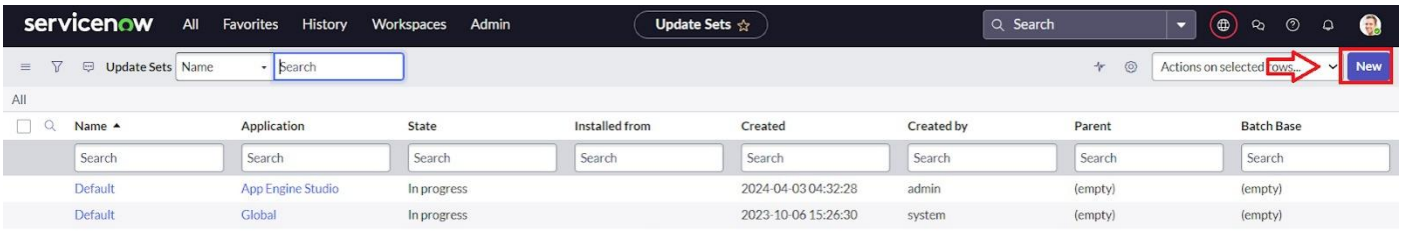
- **ServiceNow Platform Knowledge**
- **Scripting (JavaScript)**
- **Database Concepts (RDBMS)**
- **Web Technologies (HTML, CSS, PHP/Node.js)**
- **Workflow Automation**
- **Problem-Solving**
- **Team Collaboration**
- **Project Presentation**
- **Analytical Thinking**

TASK INITIATION

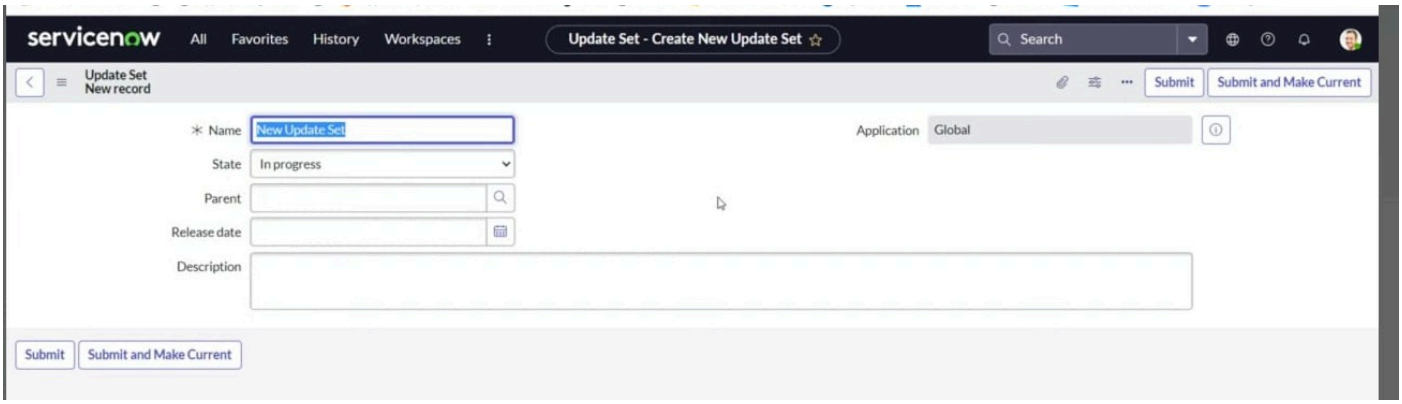
- **<u>Module 1</u>: Creating an Update Set**



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



Step 2: Click on New

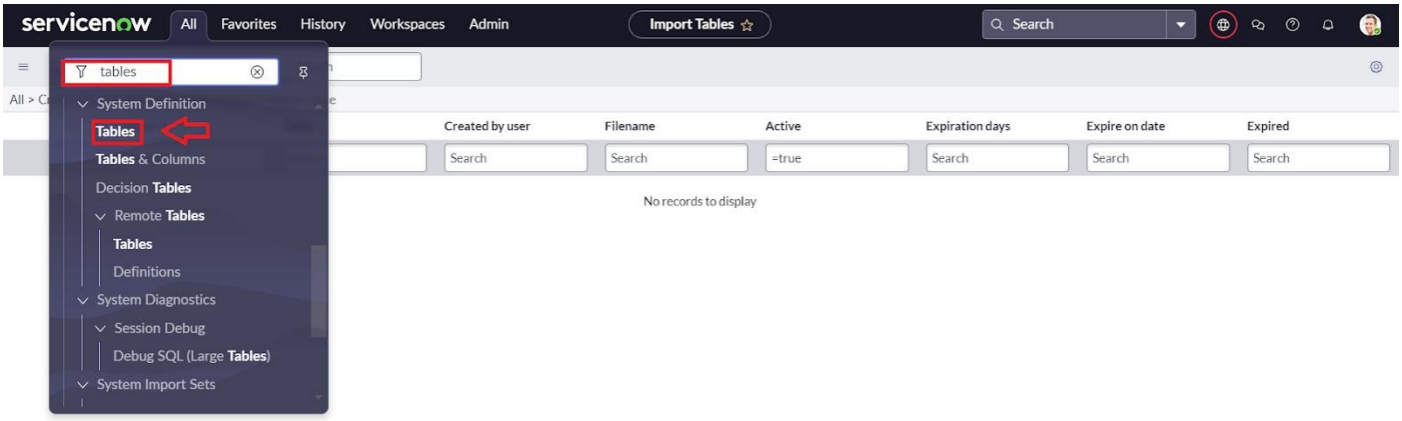


Step 3: Enter the Details Name: Educational Organisation >> Click on Submit and make Current.

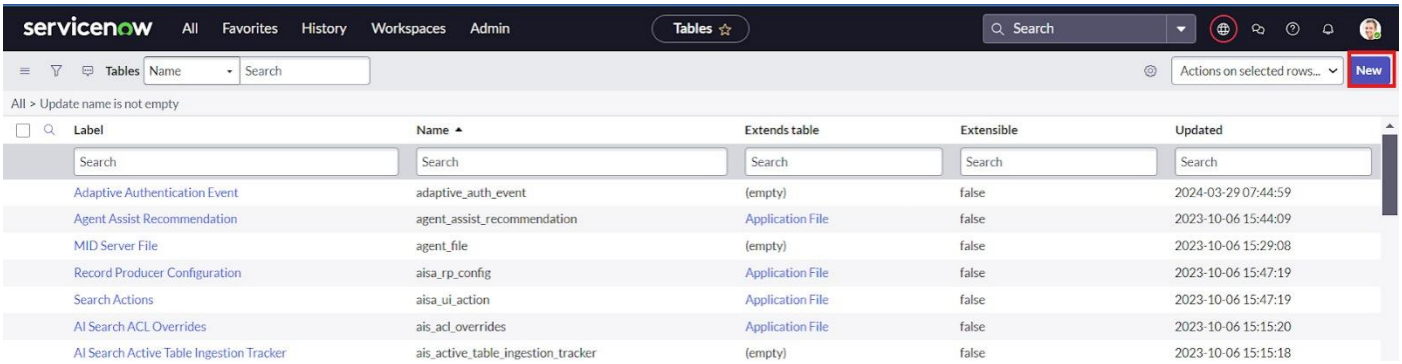
- **Module 2: Creating A Table**

Sub Module 1: Creating Salesforce Table.

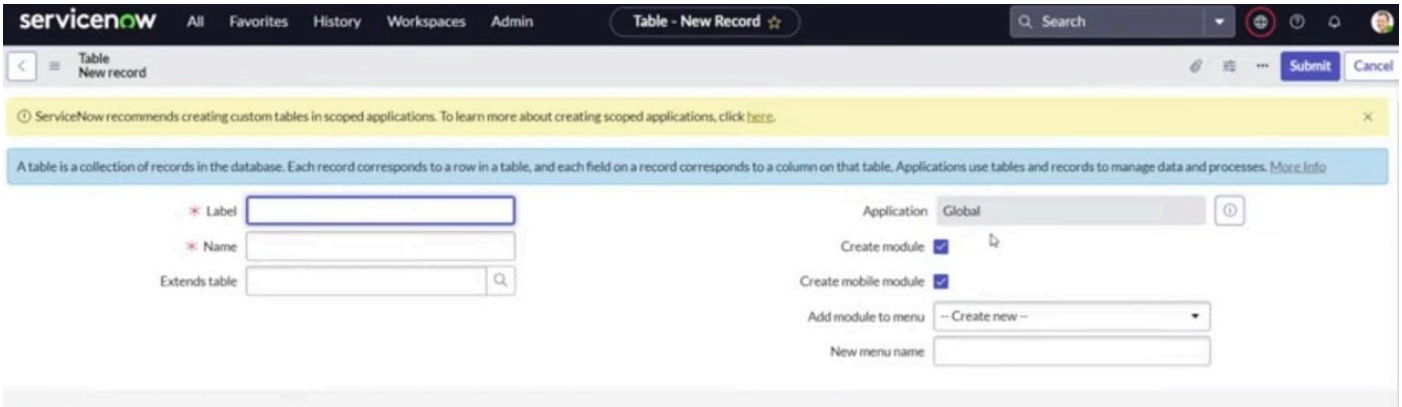
Step 1: All>>Tables



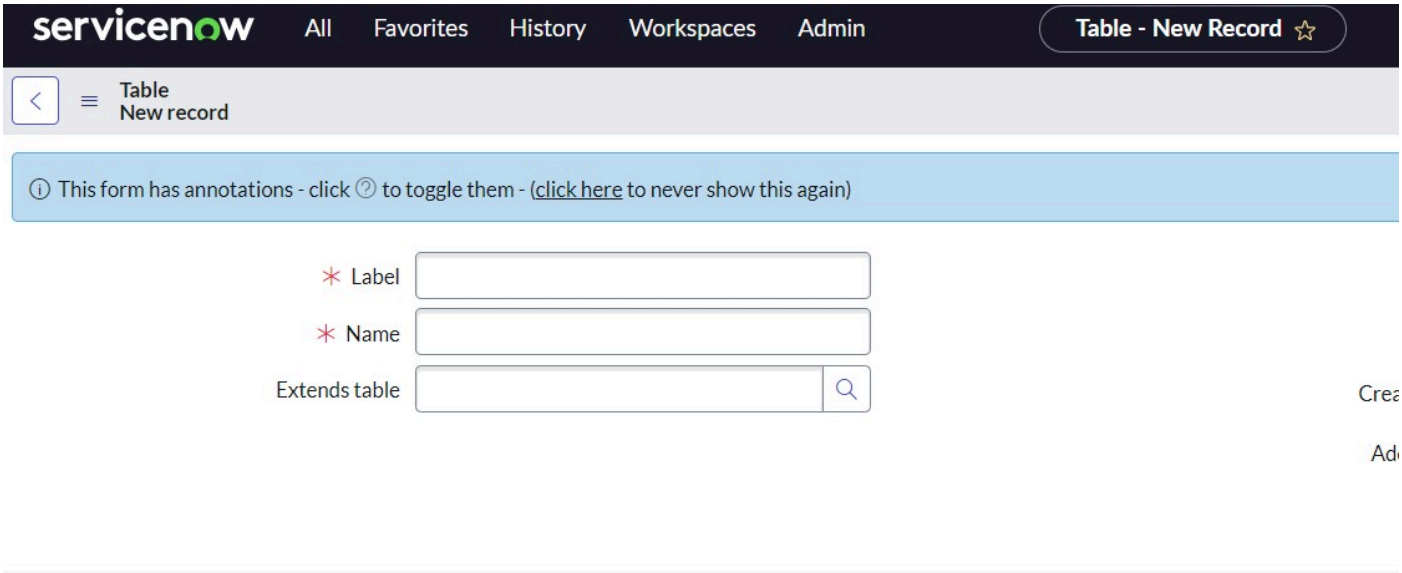
Step 2: Click On New

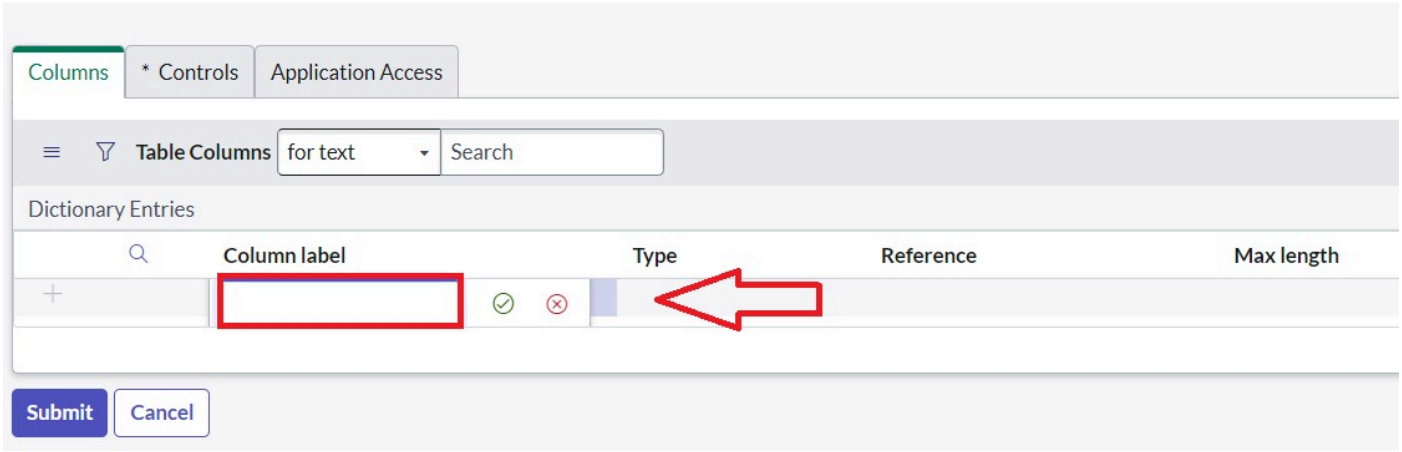


Step 3: Enter the Label (Anything you want): Salesforce >> Click on Name it will Automatically generate Api name.

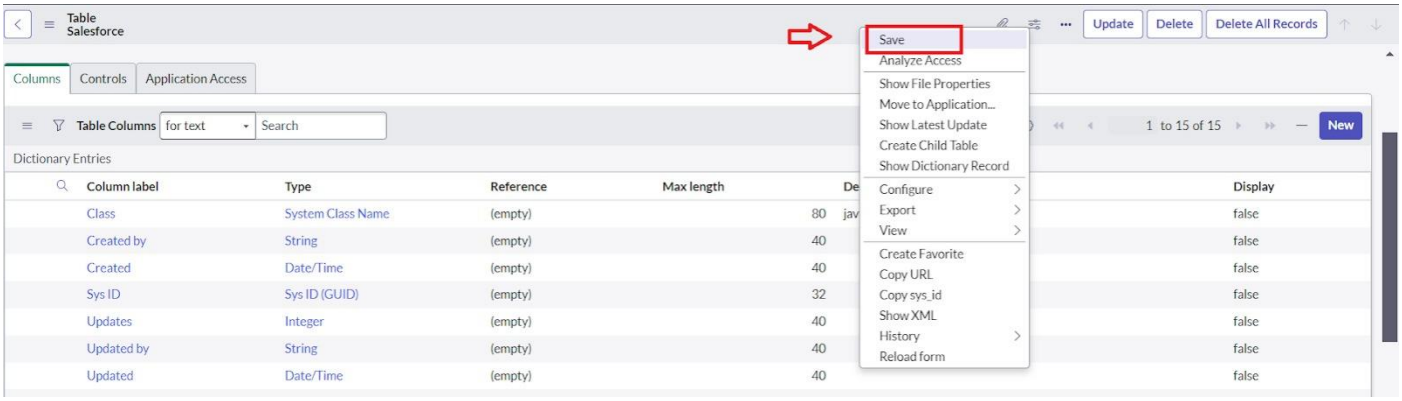


Step 4: Create columns as given below, Double Click on Column label and Enter the Column labels and click on the tick mark >> Give Type as given.

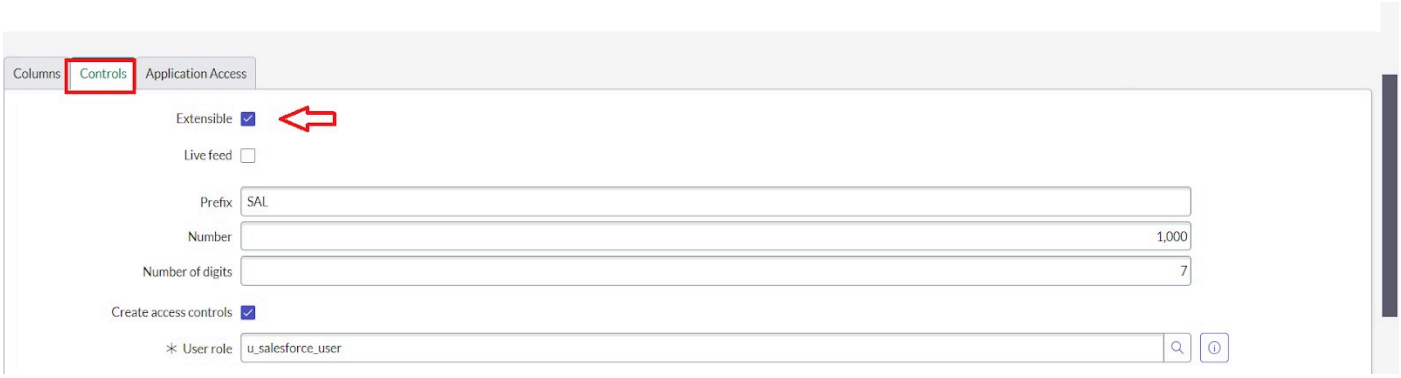




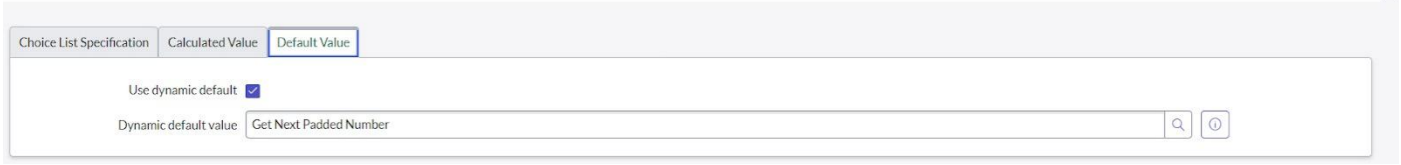
Step 5: For “Admin Number” Give Display as True and right click on the toggle bar on top >> save.



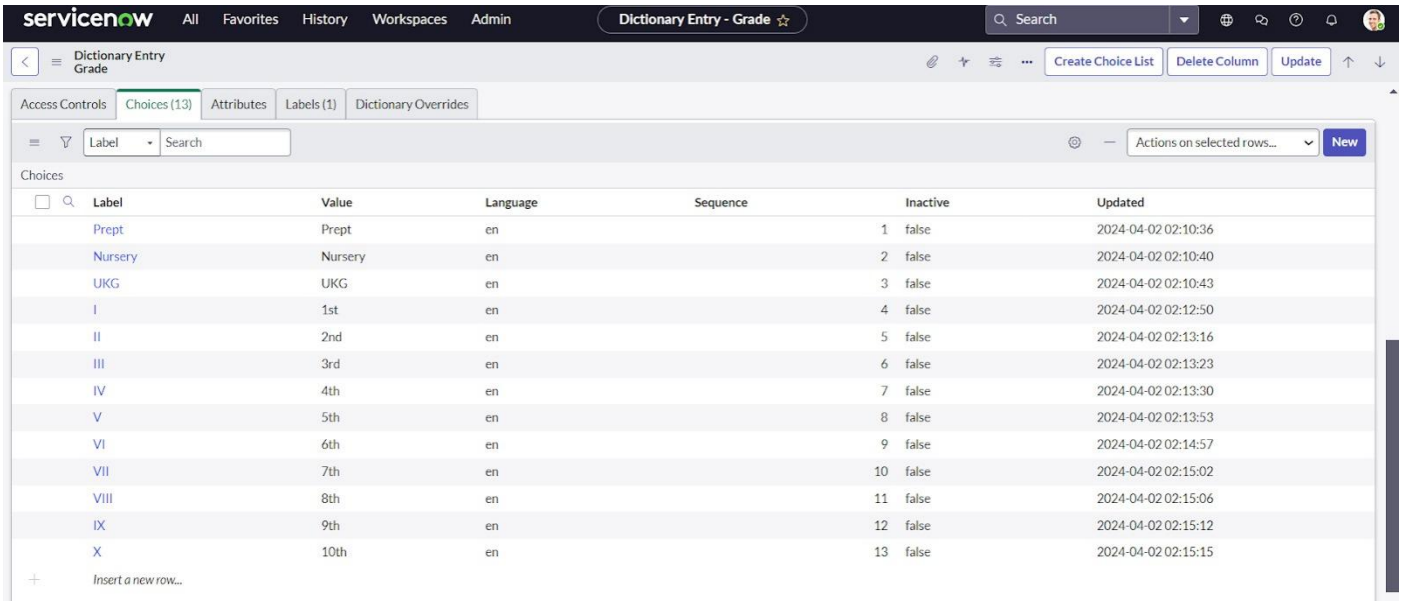
Step 6: Click on controls >> Enable Extensible.



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.



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

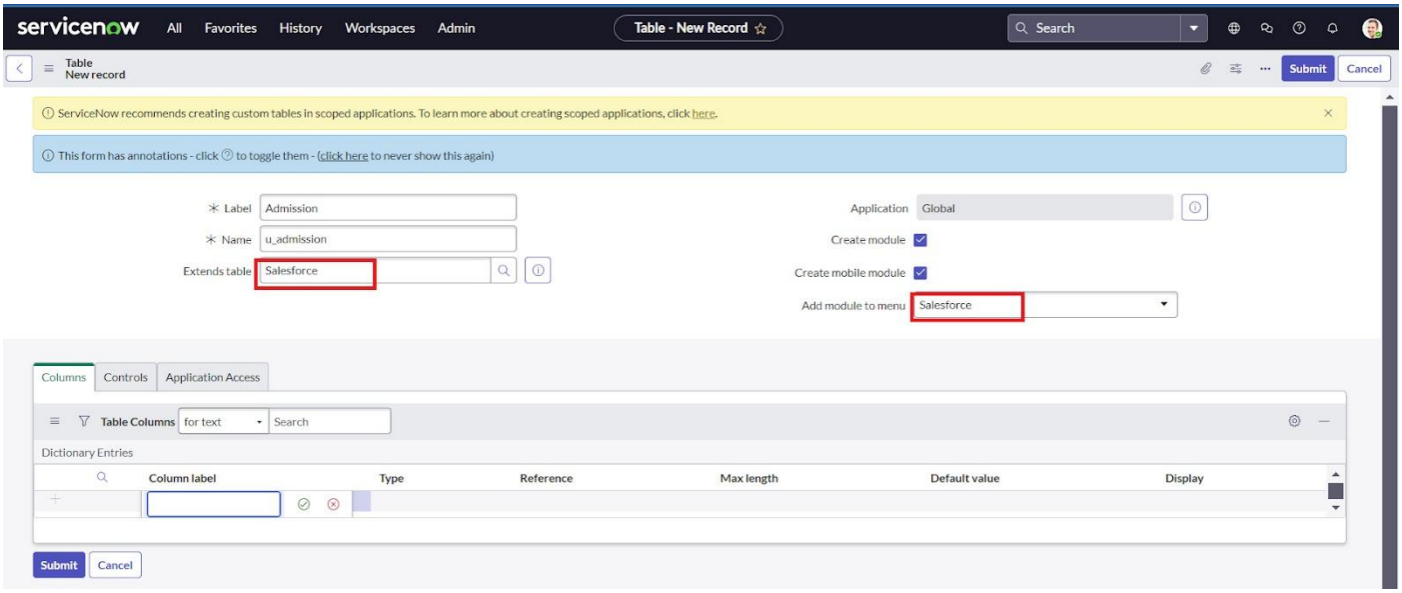


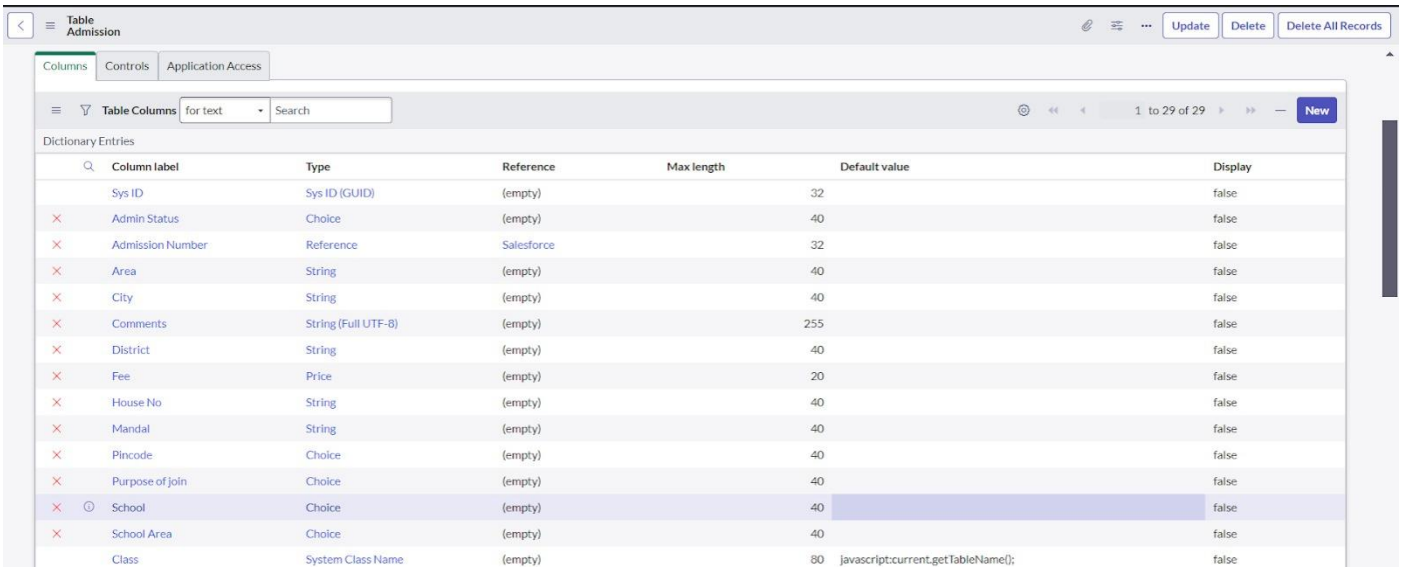
Sub Module 2: Creating Admission Table

Step 1: Create an Admission Table with Columns given.

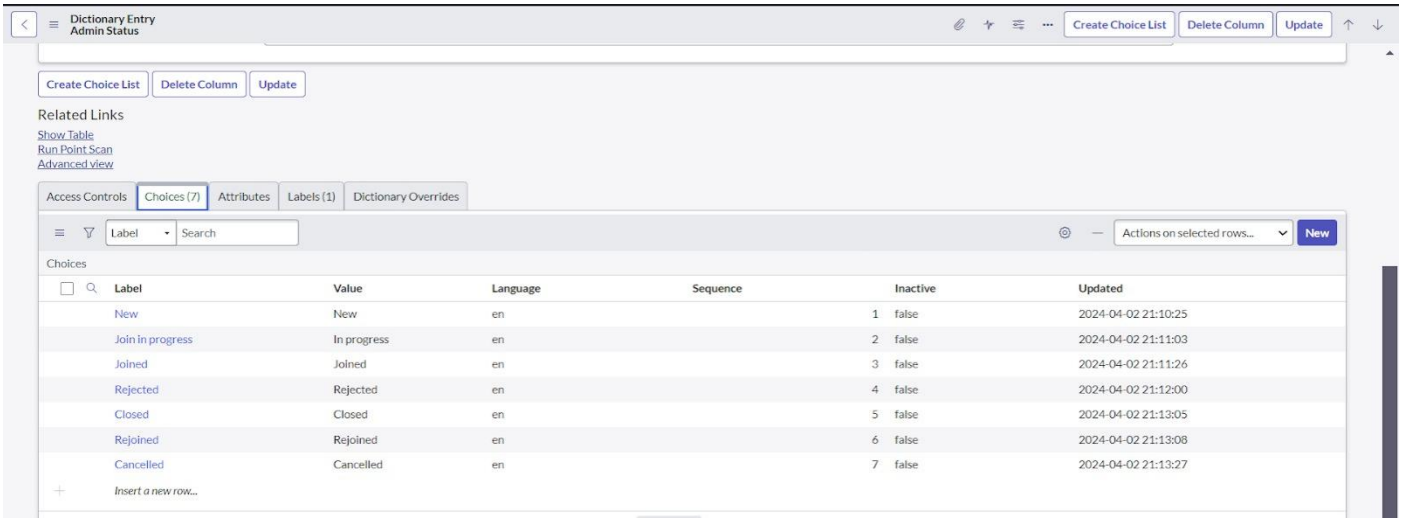
Step 2: Select Extends Table >> Salesforce and also Select Add module to menu >> Salesforce.

Step 3: Create Fields as shown.

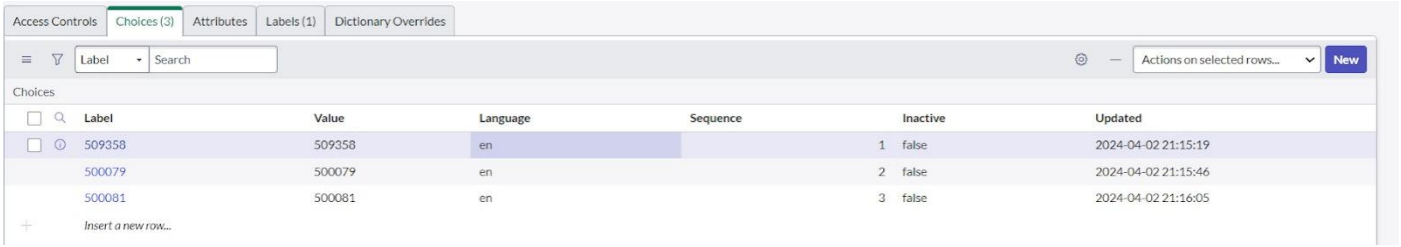




Step 4: Create choice for Admin Status as:

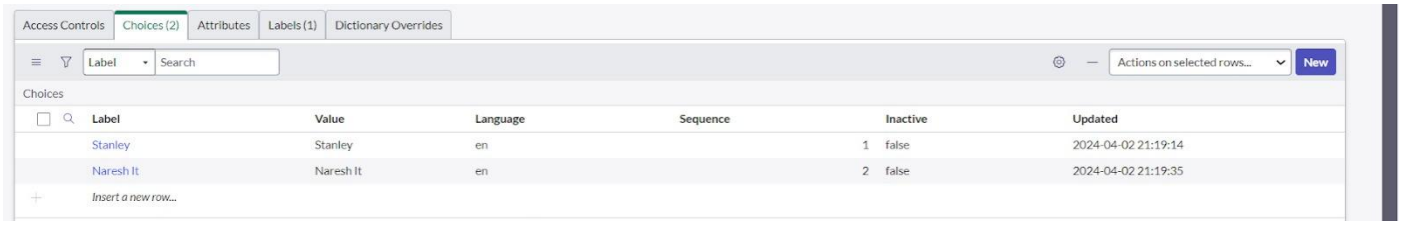


Step 5: Create choice for Pin code as:

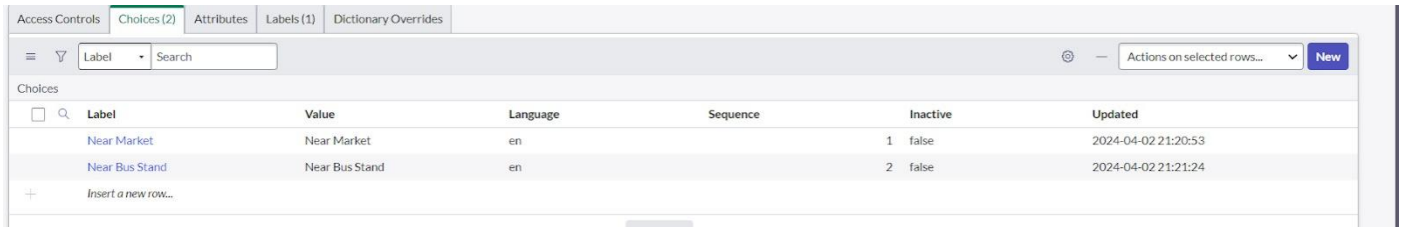


Step 6: Create choice for Purpose of Join as:

Step 7: Create choice for School as:



Step 8: Create choice for School Area as:



Sub Module 3: Creating Student Progress Table

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

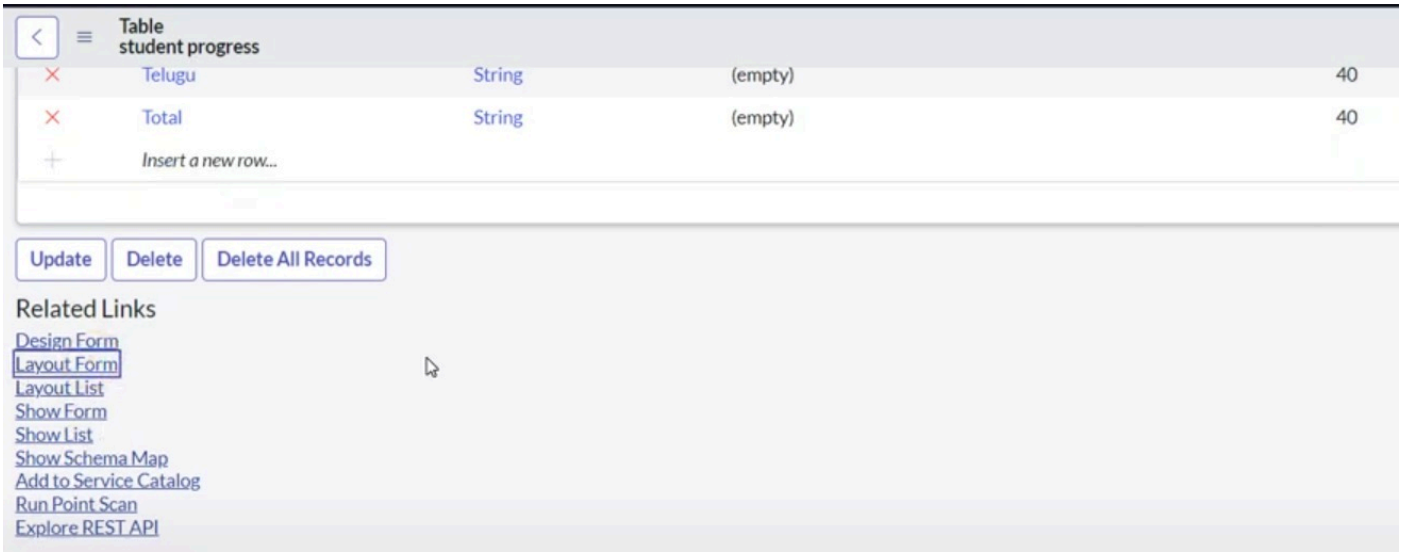
Step 2: Select Add module to menu >> Salesforce.

Step 3: Create Fields as shown:

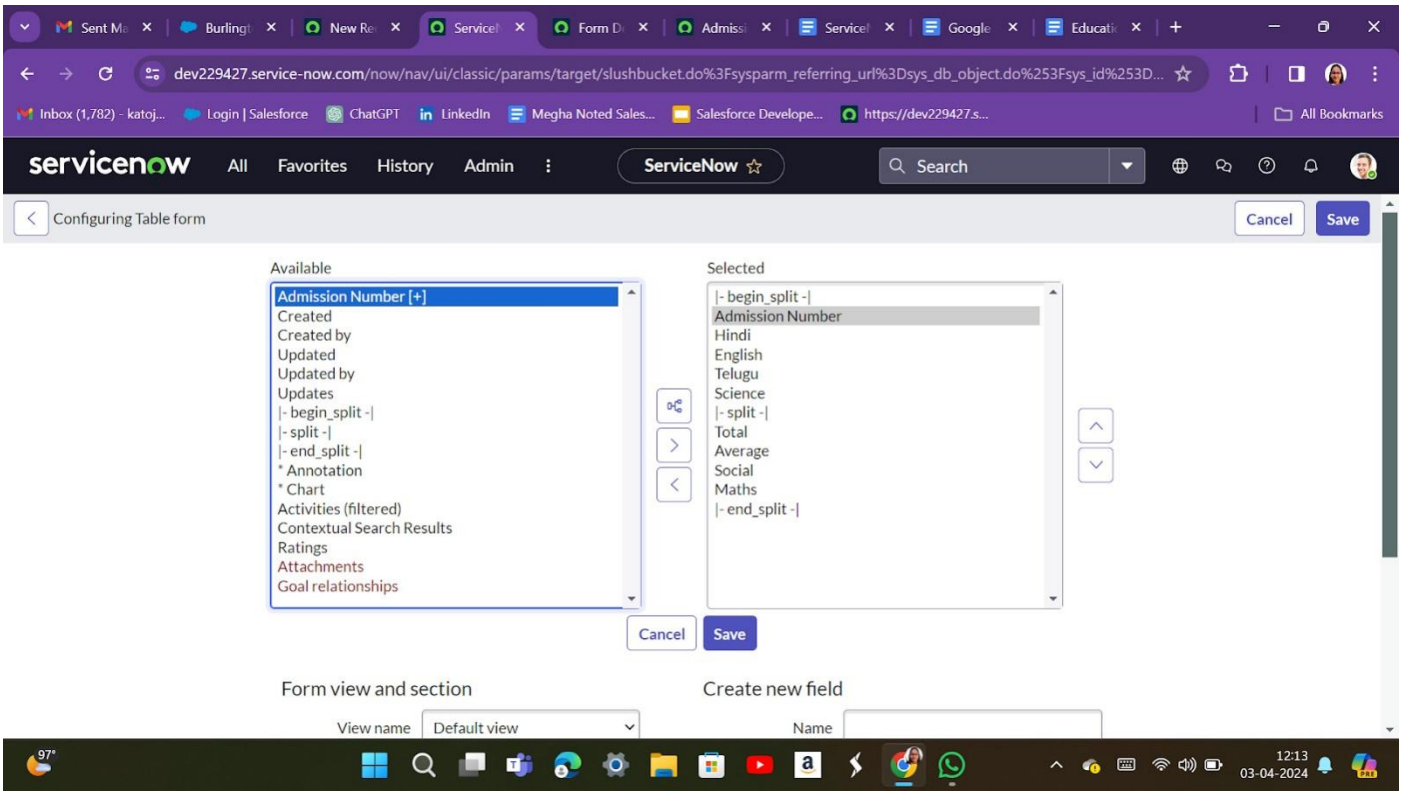
×	Admission Number	Reference	Salesforce	32	false
×	English	String	(empty)	40	false
×	Hindi	String	(empty)	40	false
×	Maths	String	(empty)	40	false
×	Percentage	String	(empty)	40	false
×	Result	String	(empty)	40	false
×	Science	String	(empty)	40	false
×	Social	String	(empty)	40	false
×	Telugu	String	(empty)	40	false
×	Total	String	(empty)	40	false
+	*Insert a new row...*				

- **Module 3: Form Layout**

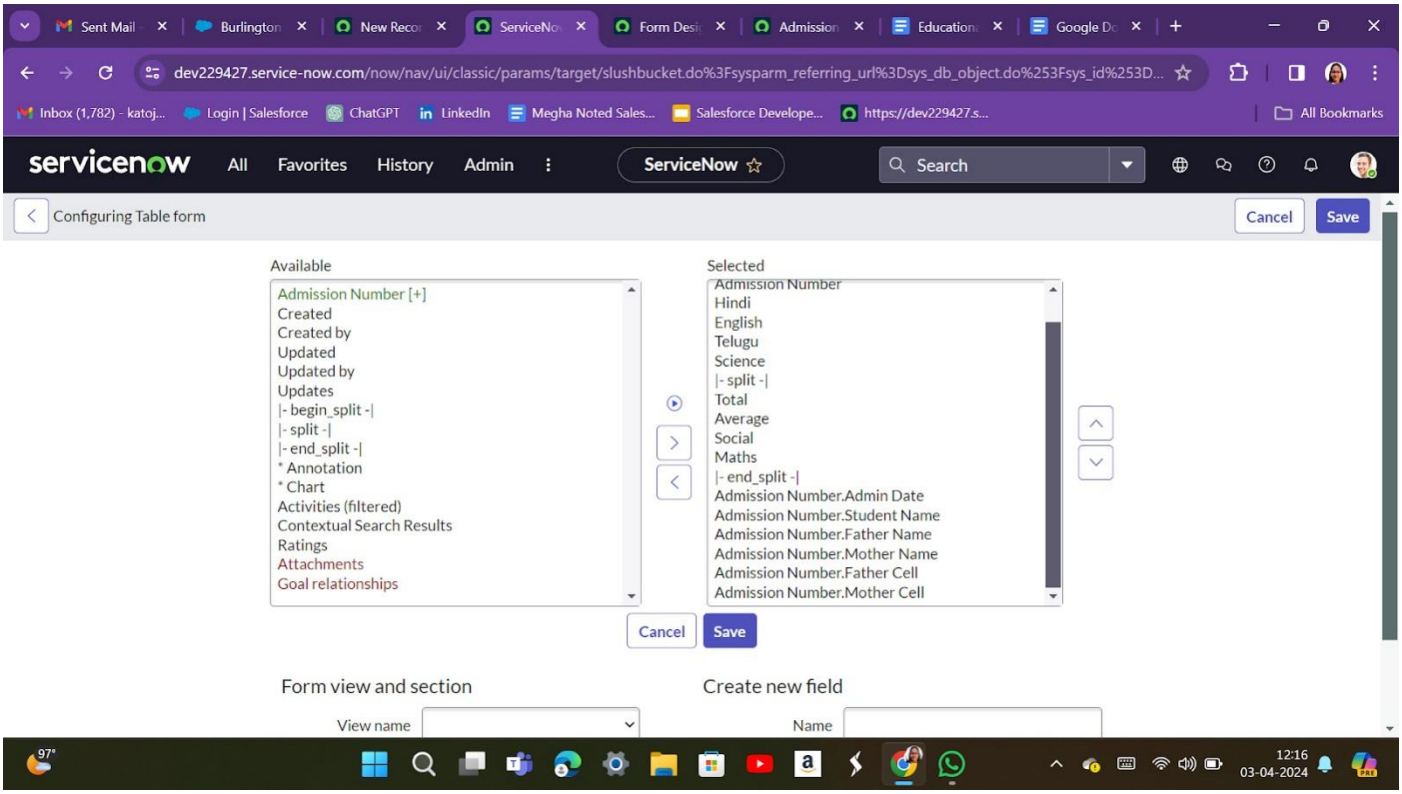
Step 1: In the Student Progress Table Page, Click on Layout form.



Step 2: Click on Admission Number [+].



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

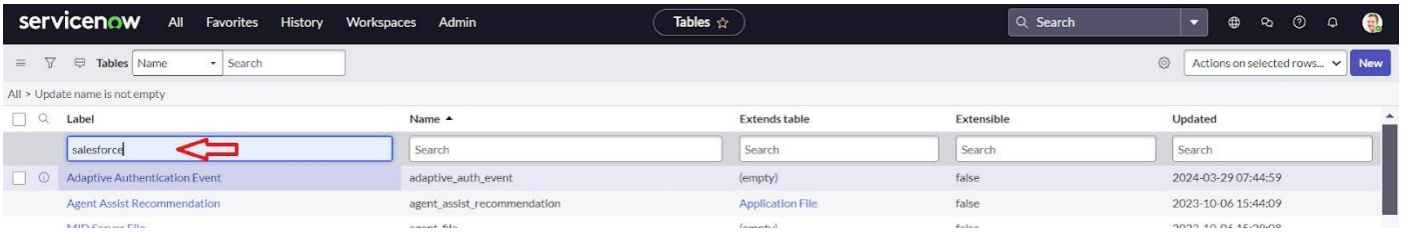


- **Module 4: Form Layout**

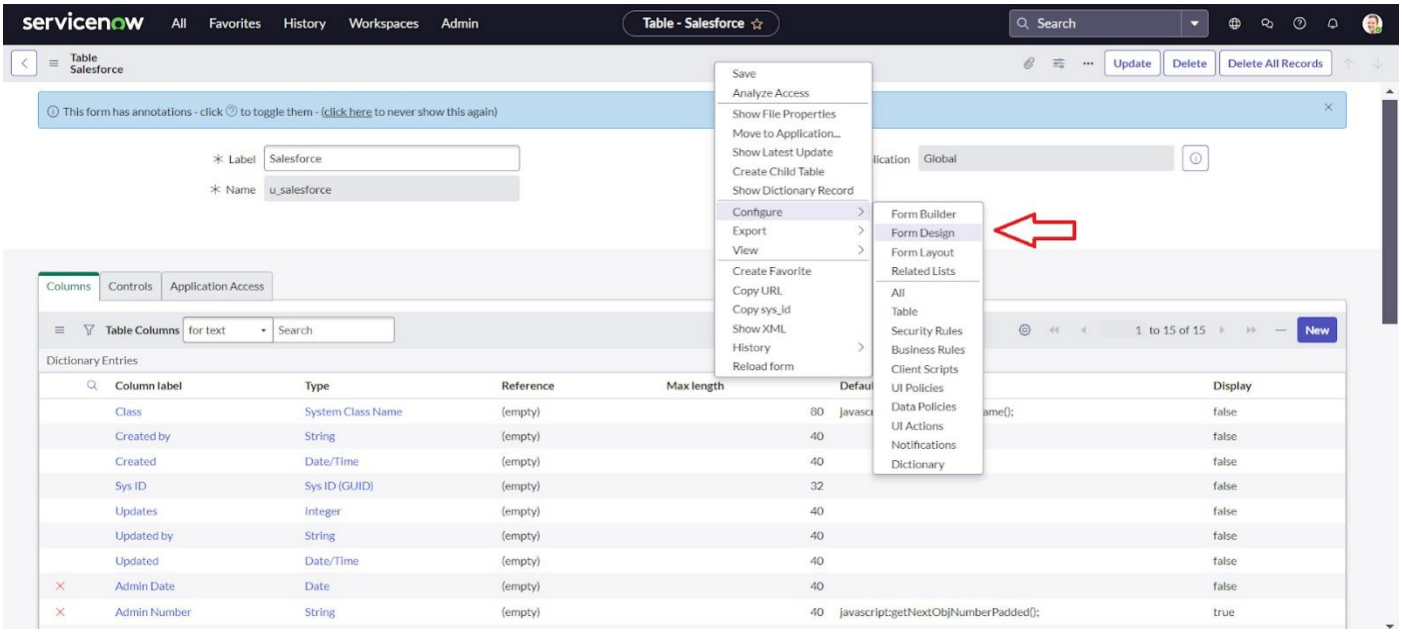
Sub Module 1: Creating Form Design for Salesforce Table

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

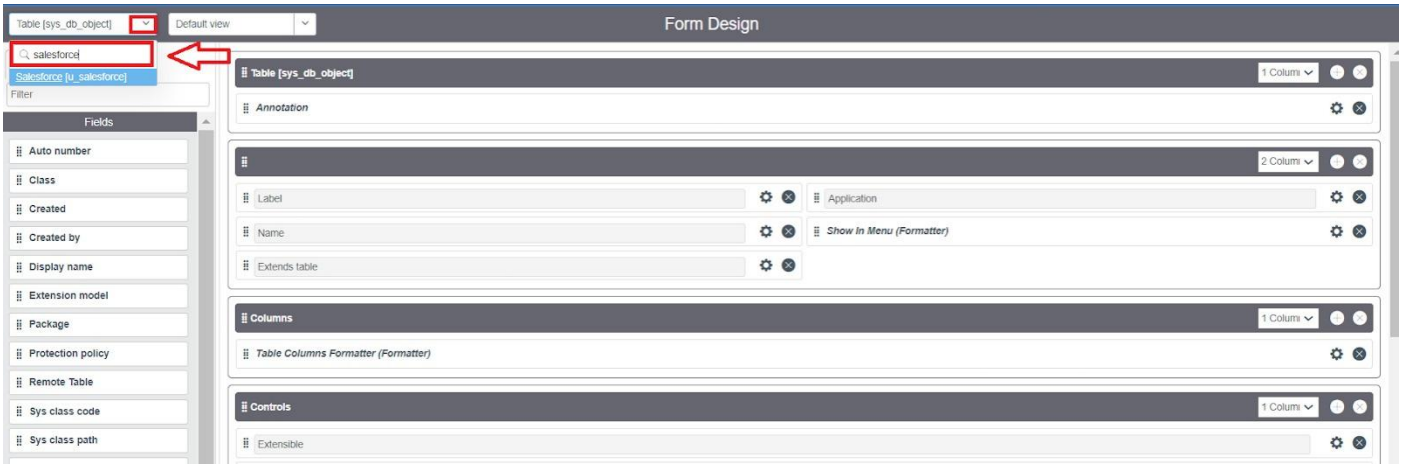
Step 2: In Label Search for Salesforce and open.



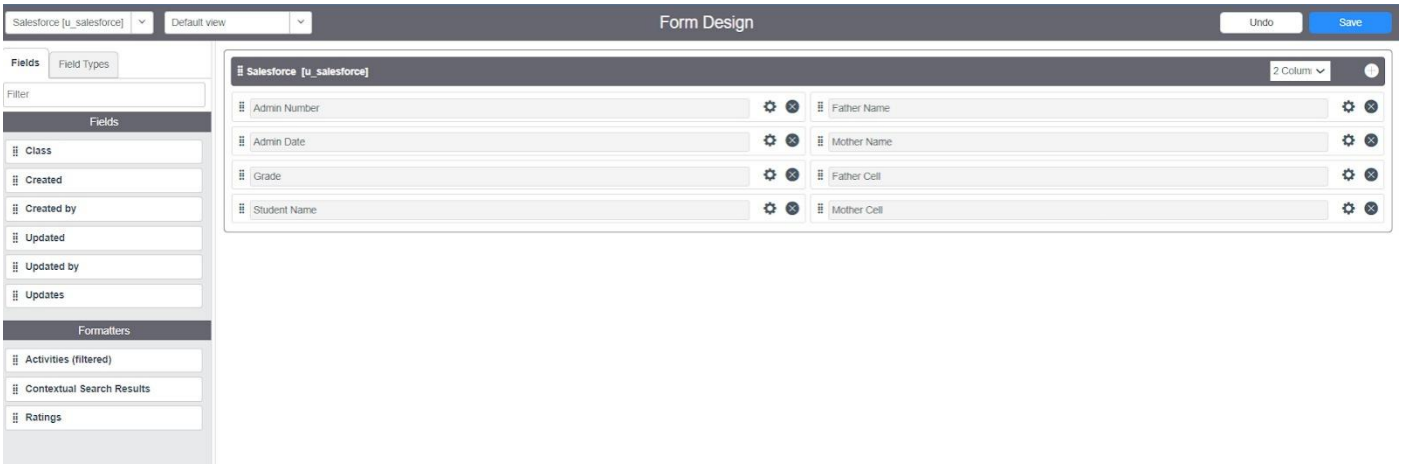
Step 3: Right Click on top Toggle >> Configure >> Form Design



Step 4: In drop down select Salesforce(u_salesforce).



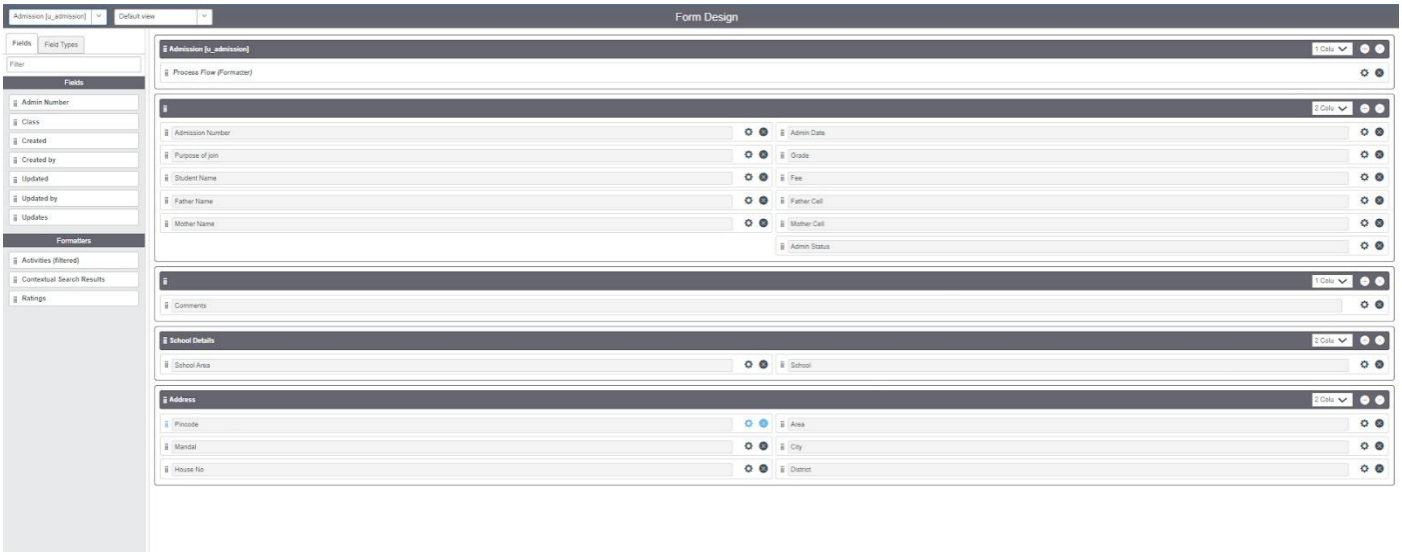
Step 5: Drag and drop the fields to the left side as below.



Step 6: Save

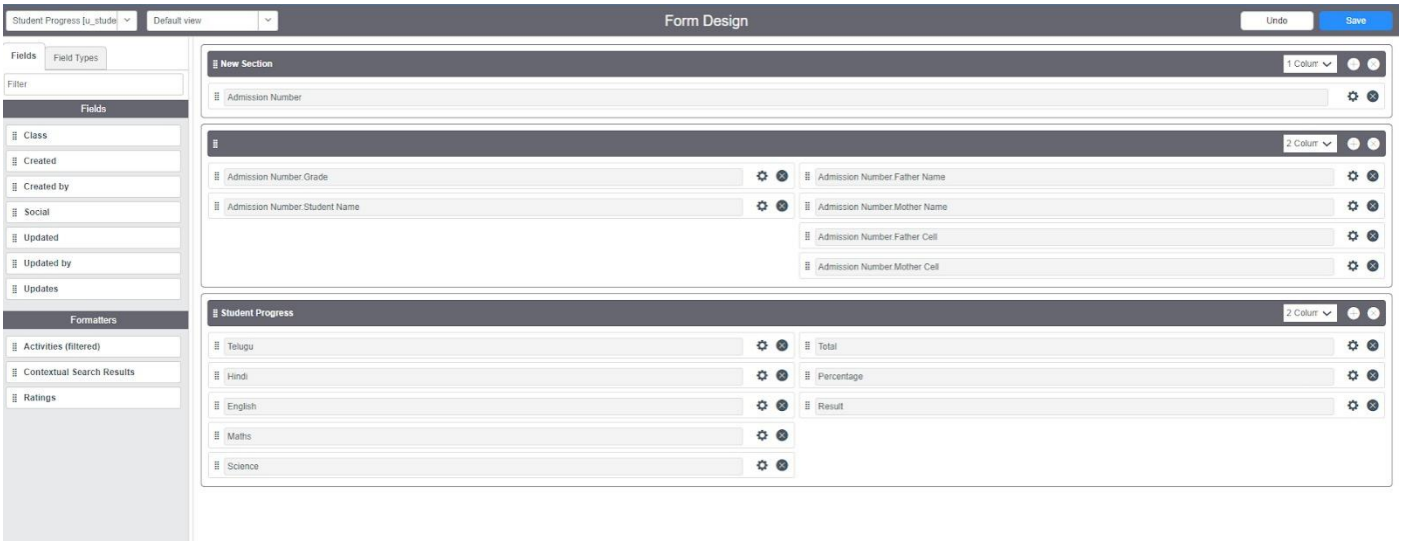
Sub Module 2: Creating Form Design for Admission Table

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



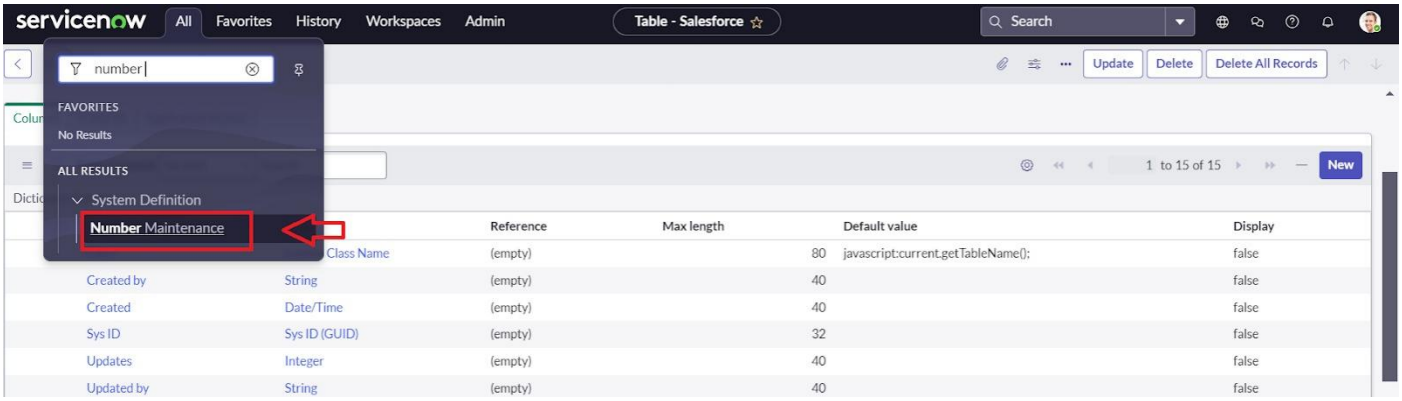
Sub Module 3: Creating Form Design for Student progress Table

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



- **Module 5: Number Maintenance**

Step 1: All >> Number Maintenance >> New

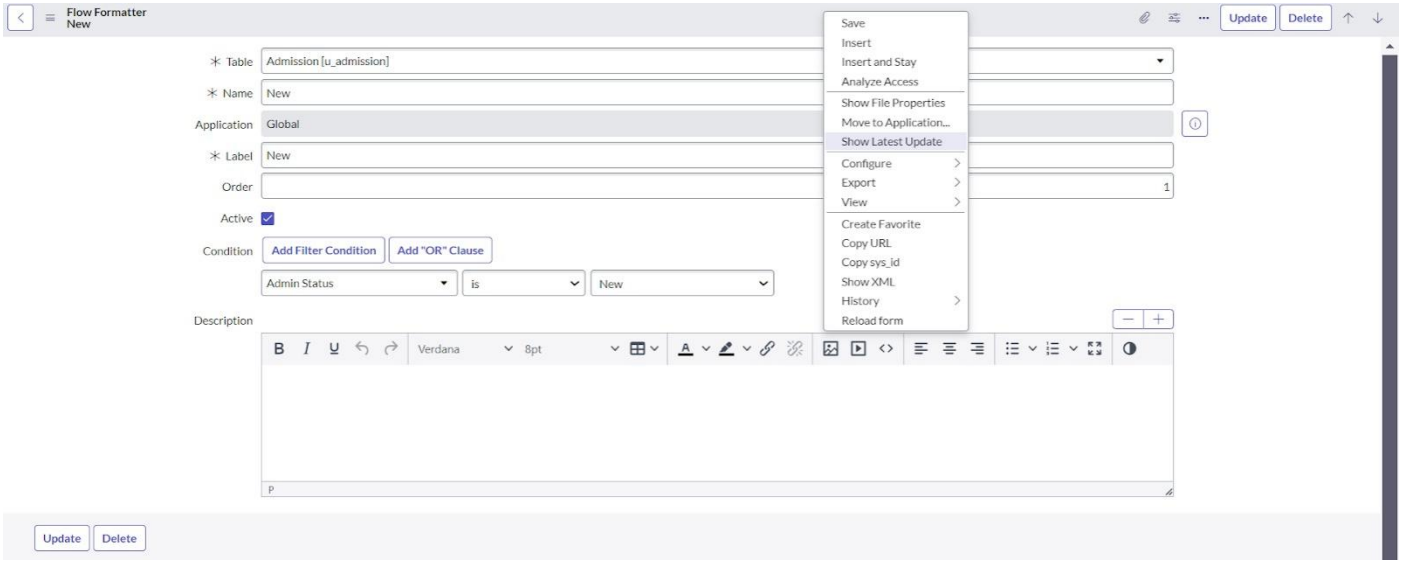


Step 2: Fill the details >> Submit.

- **Module 6: Process Flow**

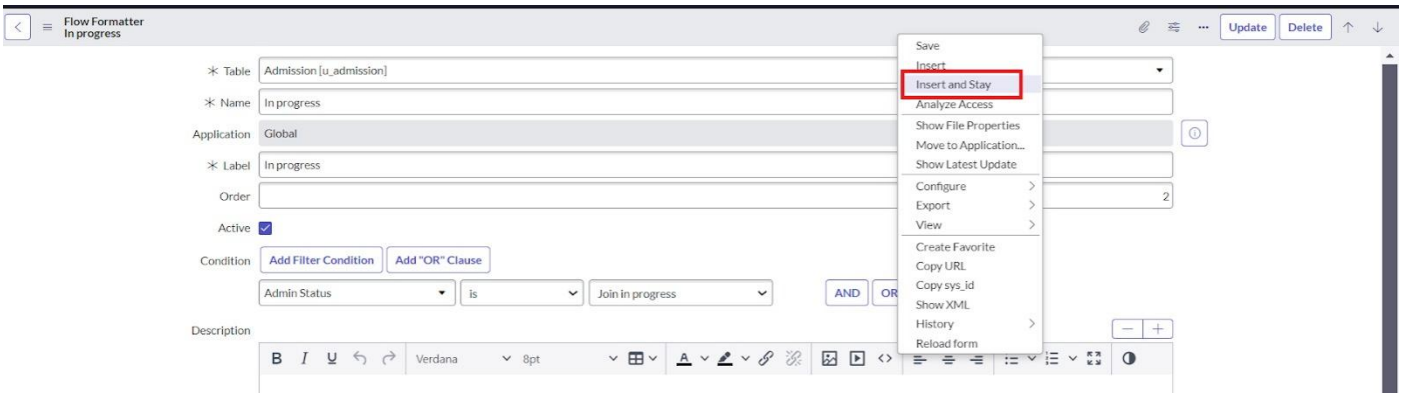
Step 1: All >> Process Flow>> New.

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.

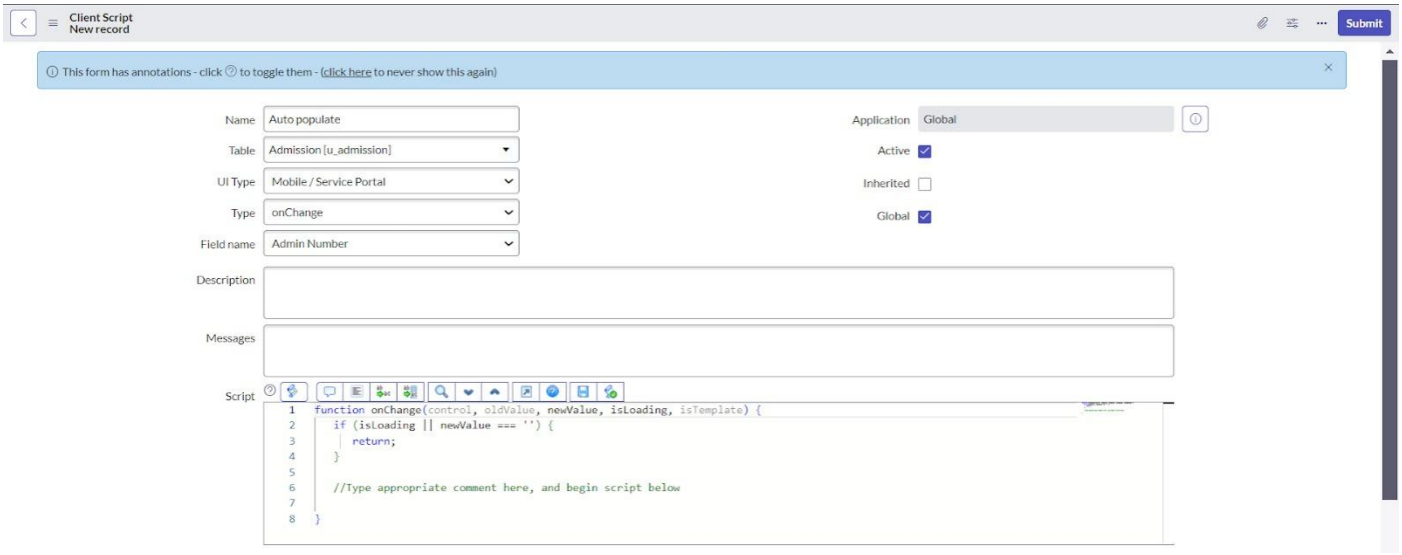
- **Module 7: Client Script**

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.




```
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);

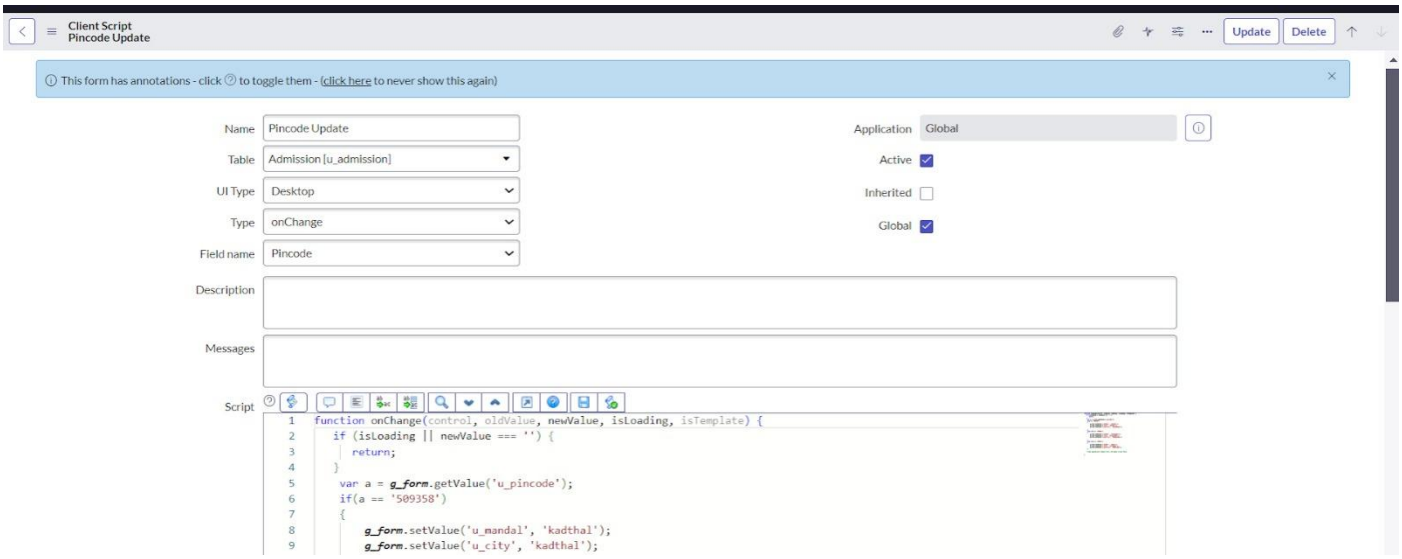
g_form.setDisabled('u_mother_cell',a.u_mother_cell);

}

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.



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');

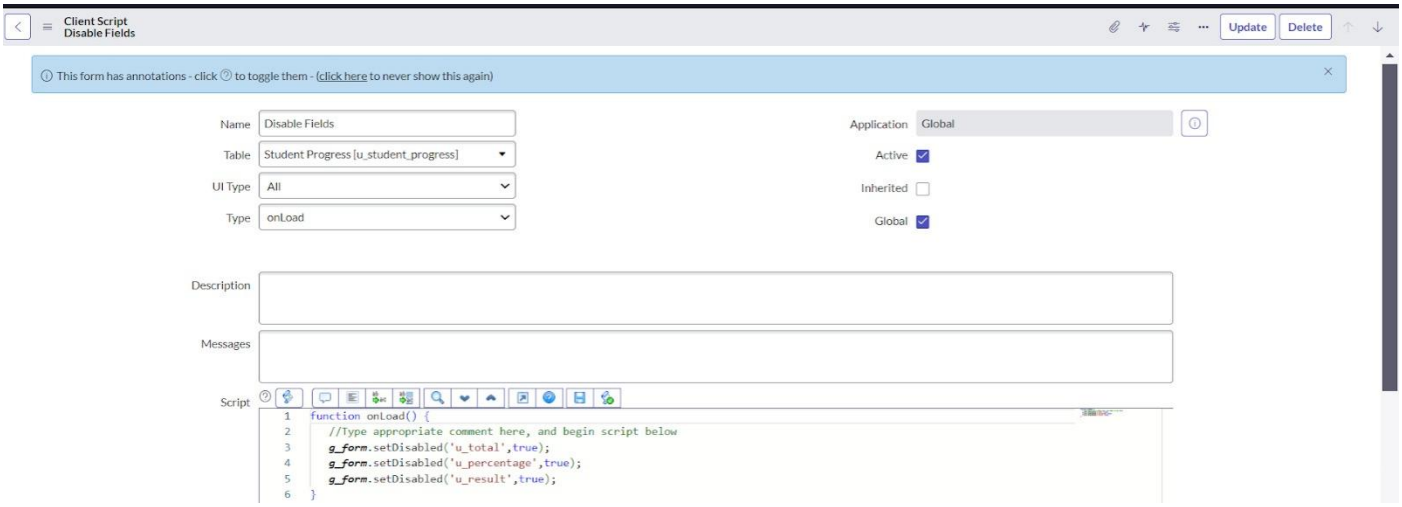
if(a == '509358')

{

```
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.



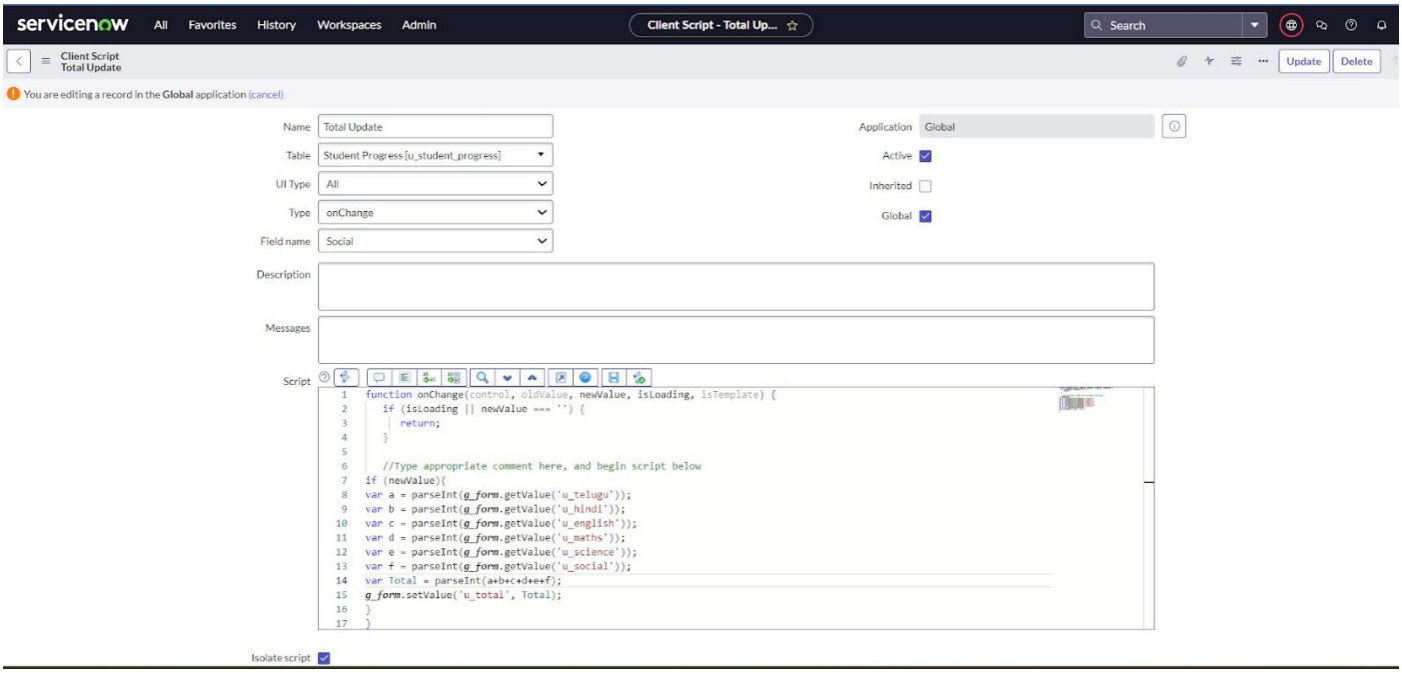
Step 2: Write the Code as below, Enable Isolate script and save.

```
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.

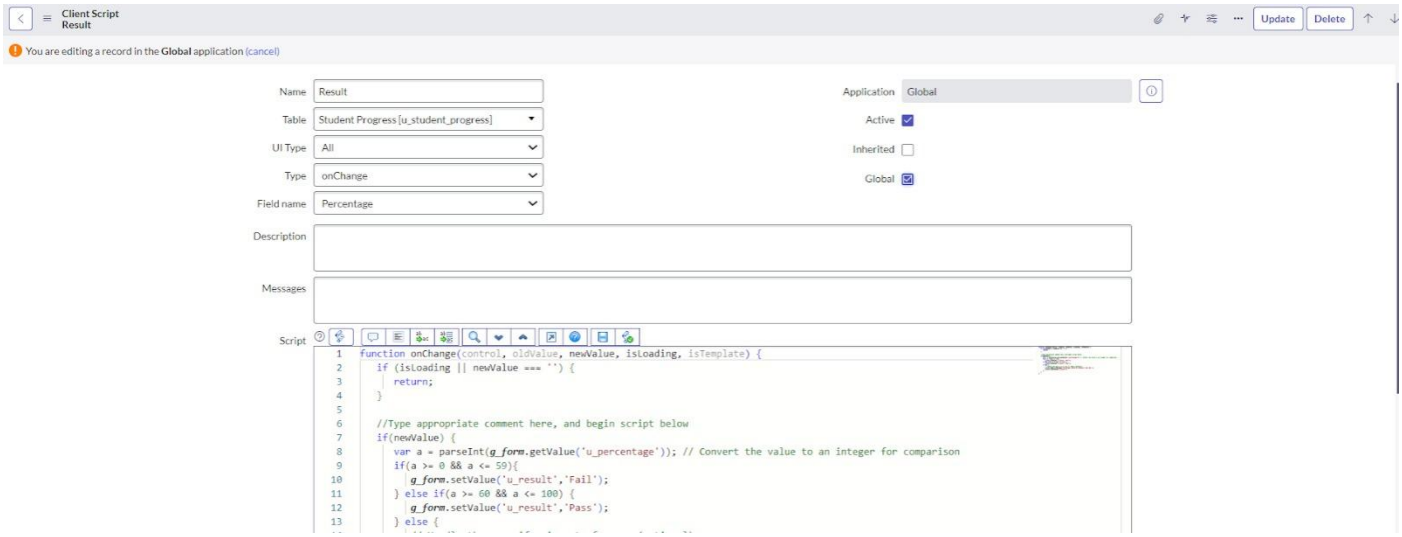
Step 2: Write the Code as below, Enable Isolate script and save.




```
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.



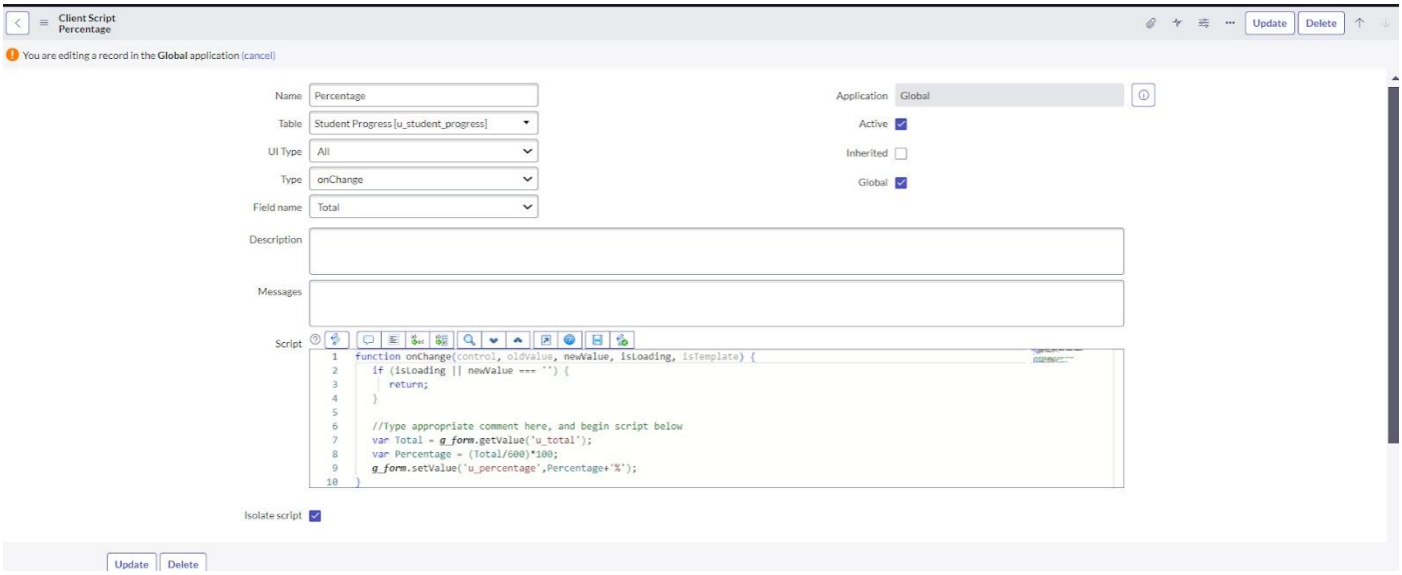
Step 2: Write the Code as below, Enable Isolate script and save.

```
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.

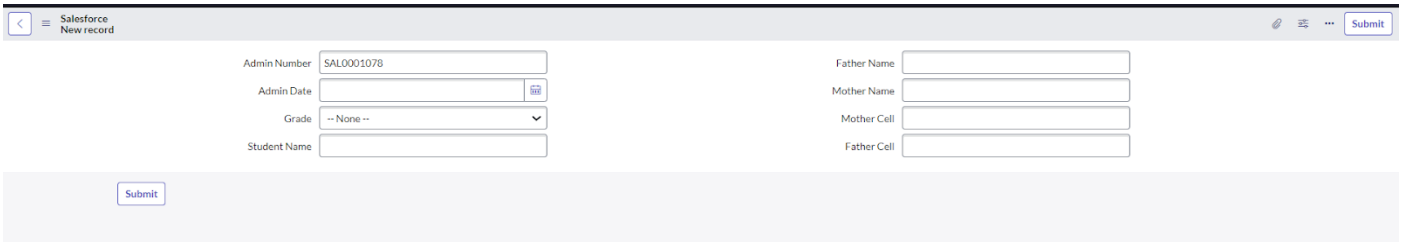


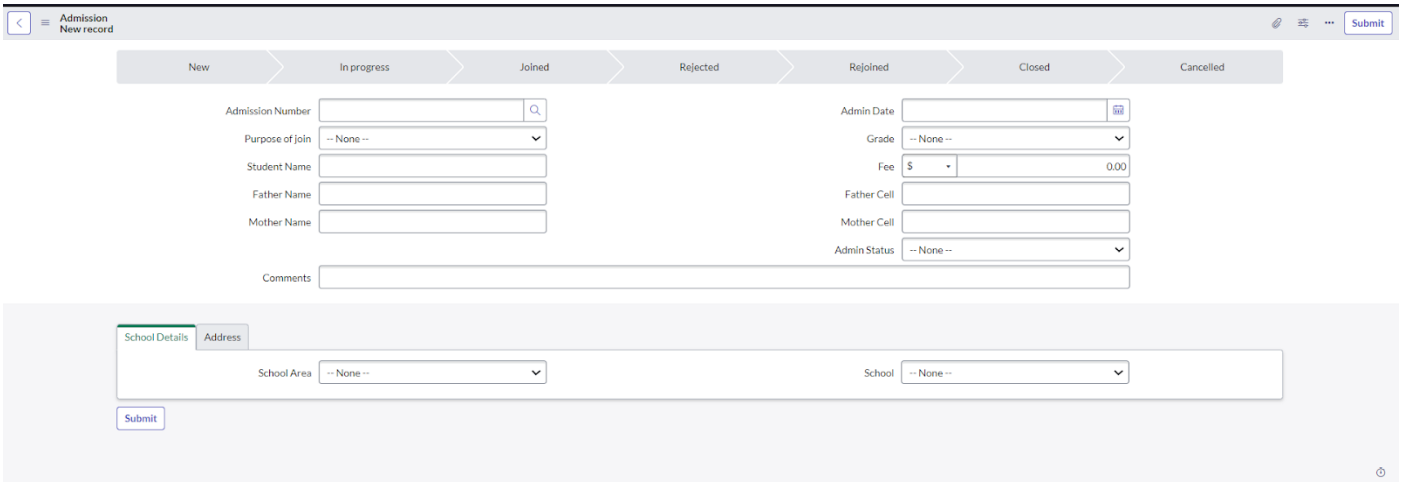
Step 2: Write the Code as below, Enable Isolate script and Save.

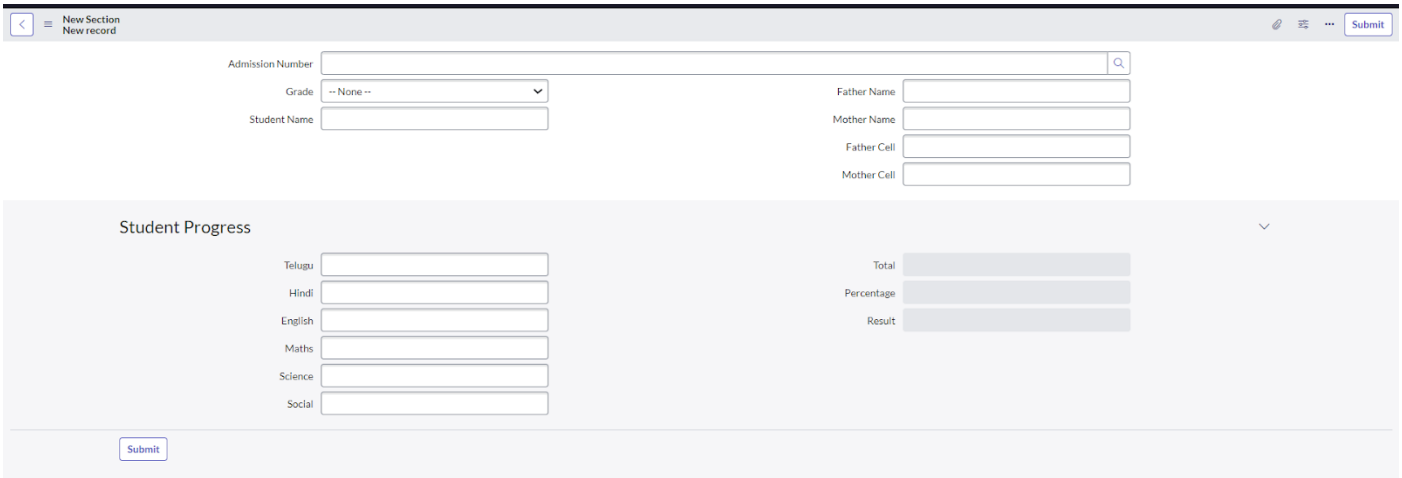
```
function onChange(control, oldValue, newValue, isLoading, isTemplate) {
  if (isLoading || newValue === '') {
    return;
```

```
} //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:







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.