

Project

Educational Organisation Using ServiceNow

Team ID : LTVIP2026TMIDS24889

Team Size : 4

Team Leader : Kanaparthi.Aruna Sri

Team member : Ganapathi G

Team member : Bhavya Tatineni

Team member : Divvela Jagadeesh

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.

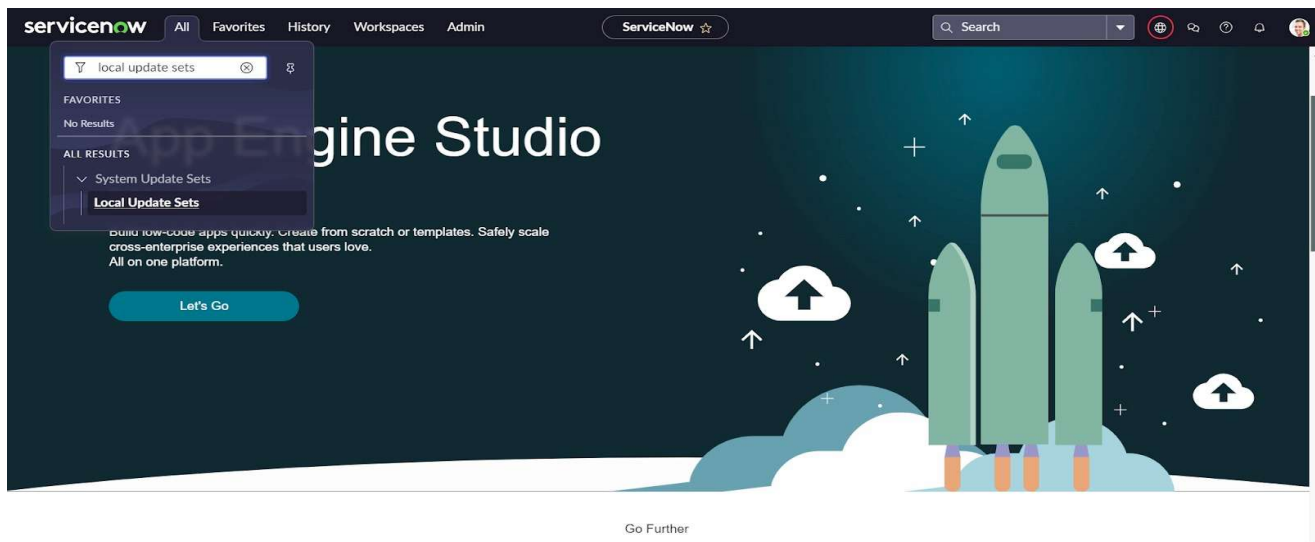
Objective: The project aims to develop a system for schools and colleges to manage admissions, student details, and academic progress, making the entire process quick, paperless, and user-friendly.

Skills:

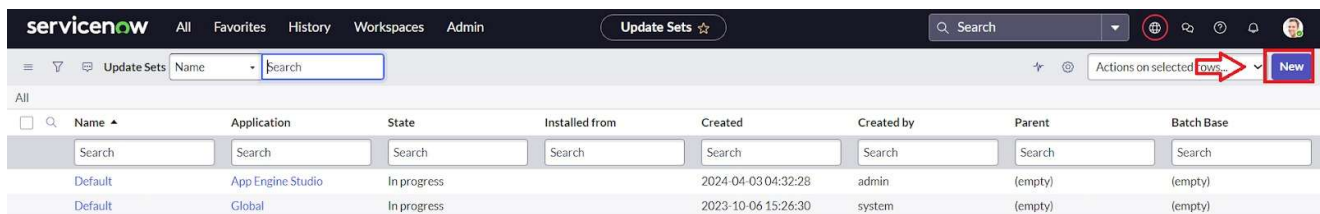
- ❖ ServiceNow Platform Knowledge
- ❖ Scripting (JavaScript)
- ❖ Database Concepts (RDBMS)
- ❖ Web Technologies (HTML, CSS, PHP/Node.js)
- ❖ Workflow Automation
- ❖ Problem-Solving
- ❖ Team Collaboration
- ❖ Project Presentation
- ❖ Analytical Thinking

TASK INITIATION

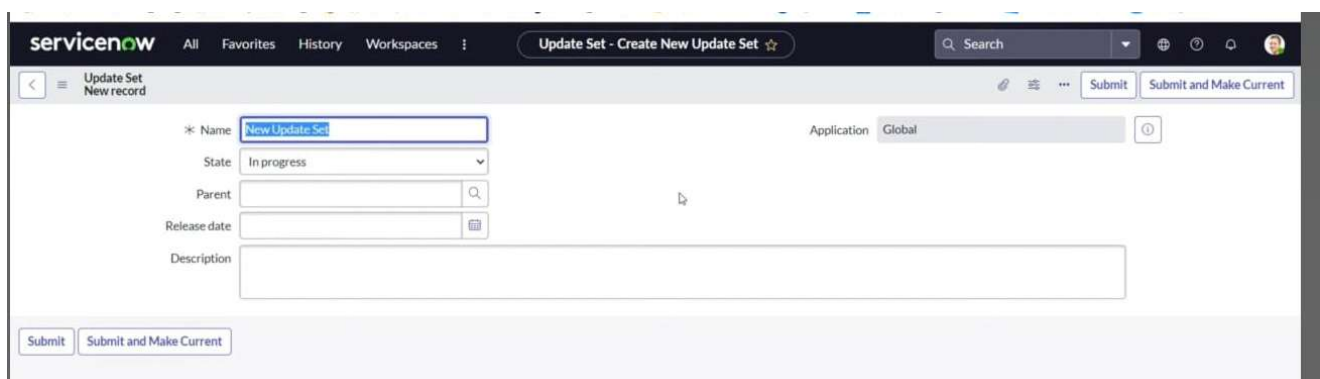
▪ Module 1: Creating an Update Set



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



Step 2: Click on New

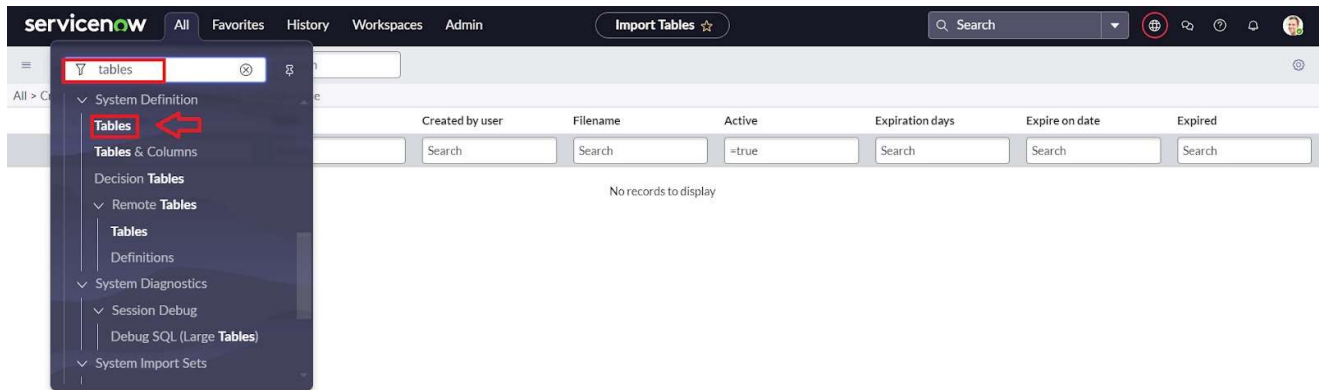


Step 3: Enter the Details Name: Educational Organisation >> Click on Submit and make Current.

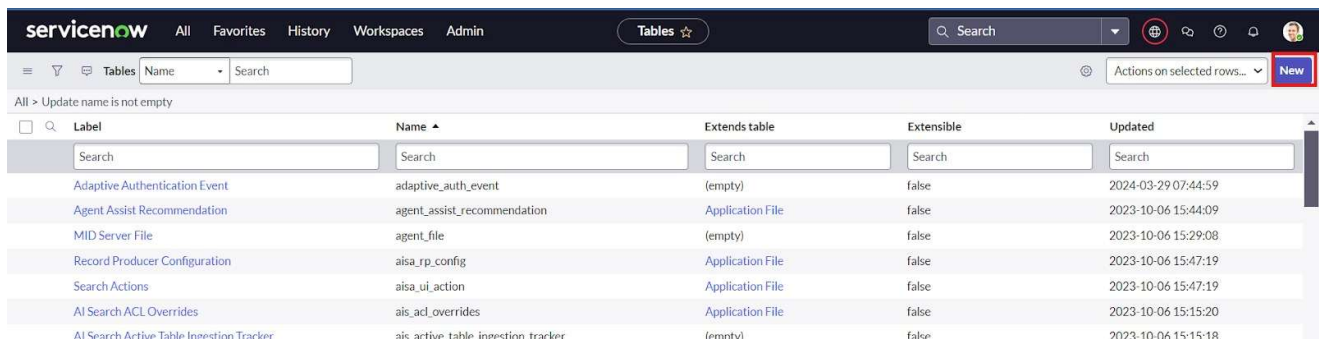
■ Module 2: Creating A Table

Sub Module 1: Creating Salesforce Table.

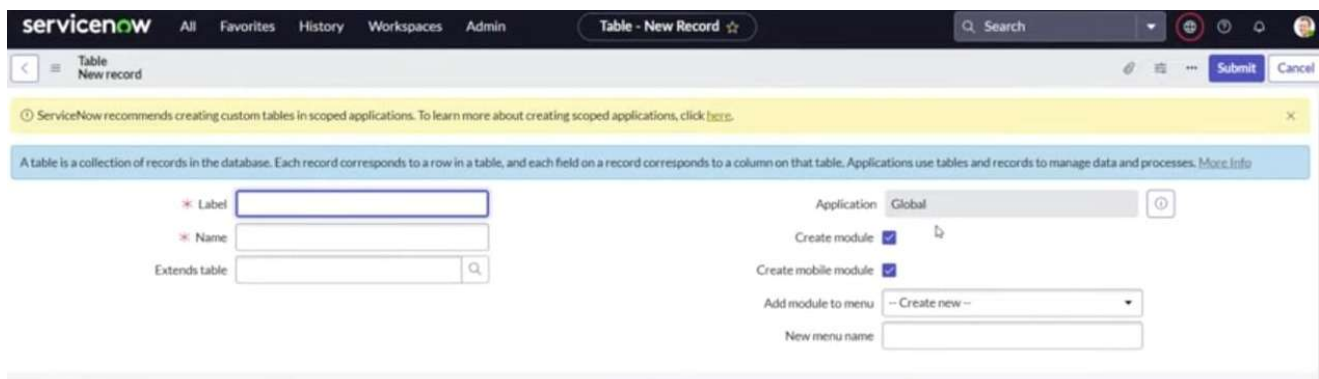
Step 1: All>>Tables



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.

* Label

* Name

Extends table

🔍

Ad

Columns

* Controls

Application Access

≡

Table Columns

for text

Search

Dictionary Entries

| | Column label | Type | Reference | Max length |
|---|----------------------|-------------------------------------|-----------|------------|
| + | <input type="text"/> | <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 'Table Designer' application with the 'Table Columns' tab selected. A red arrow points to the 'Save' button in the top right corner. The 'Table Columns' list is visible, showing columns: Column label, Type, Reference, Max length, and Display. The list contains 10 entries: Class, Created by, Created, Sys ID, Updates, Updated by, Updated, and three empty rows. The 'Save' button is highlighted with a red box and a red arrow.

Step 6: Click on controls >> Enable Extensible.



Columns Controls Application Access

Extensible ☒ 

Live feed ☐

Prefix SAL

Number 1,000

Number of digits 7

Create access controls ☒

* User role u_salesforce_user

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.

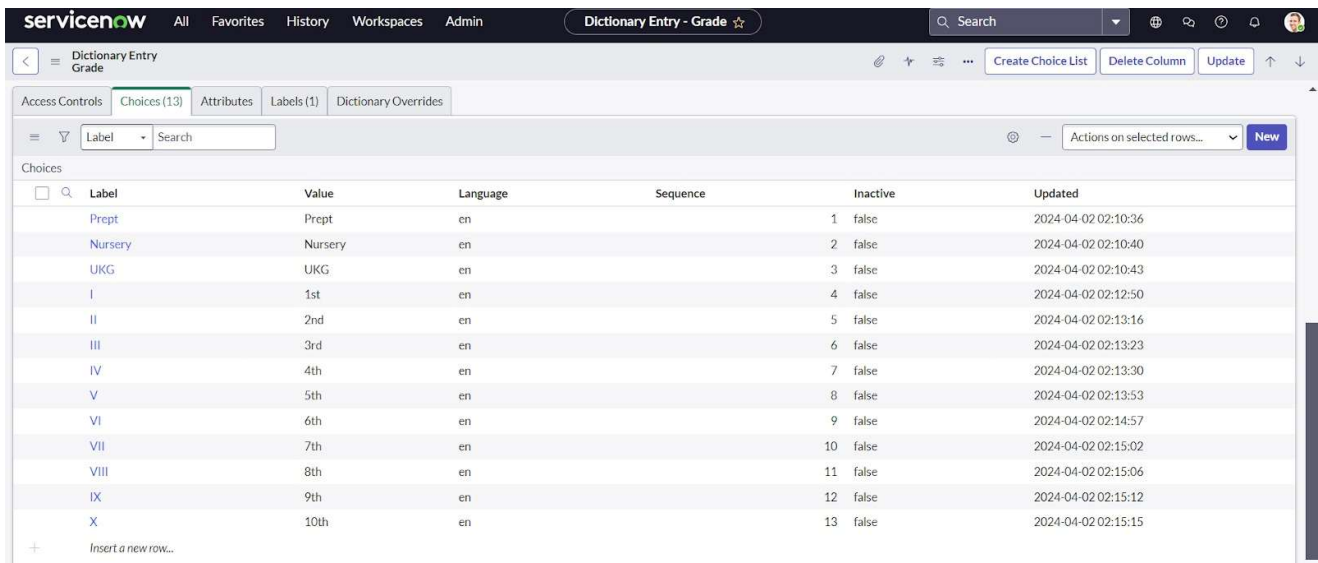


Choice List Specification Calculated Value Default Value

Use dynamic default ☒

Dynamic default value Get Next Padded Number

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



servicenow All Favorites History Workspaces Admin Dictionary Entry - Grade

Dictionary Entry Grade

Access Controls Choices (13) Attributes Labels (1) Dictionary Overrides

Label Search

Actions on selected rows... New

| 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.

The screenshot shows the 'Table - New Record' form in ServiceNow. The 'Label' is 'Admission' and the 'Name' is 'u_admission'. The 'Extends table' is set to 'Salesforce'. The 'Application' is 'Global'. The 'Create module' and 'Create mobile module' checkboxes are checked. The 'Add module to menu' dropdown is set to 'Salesforce'. Below the form, the 'Columns' tab is active, showing a table of dictionary entries.

| 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.getTableNm(); | 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 -]

Cancel Save

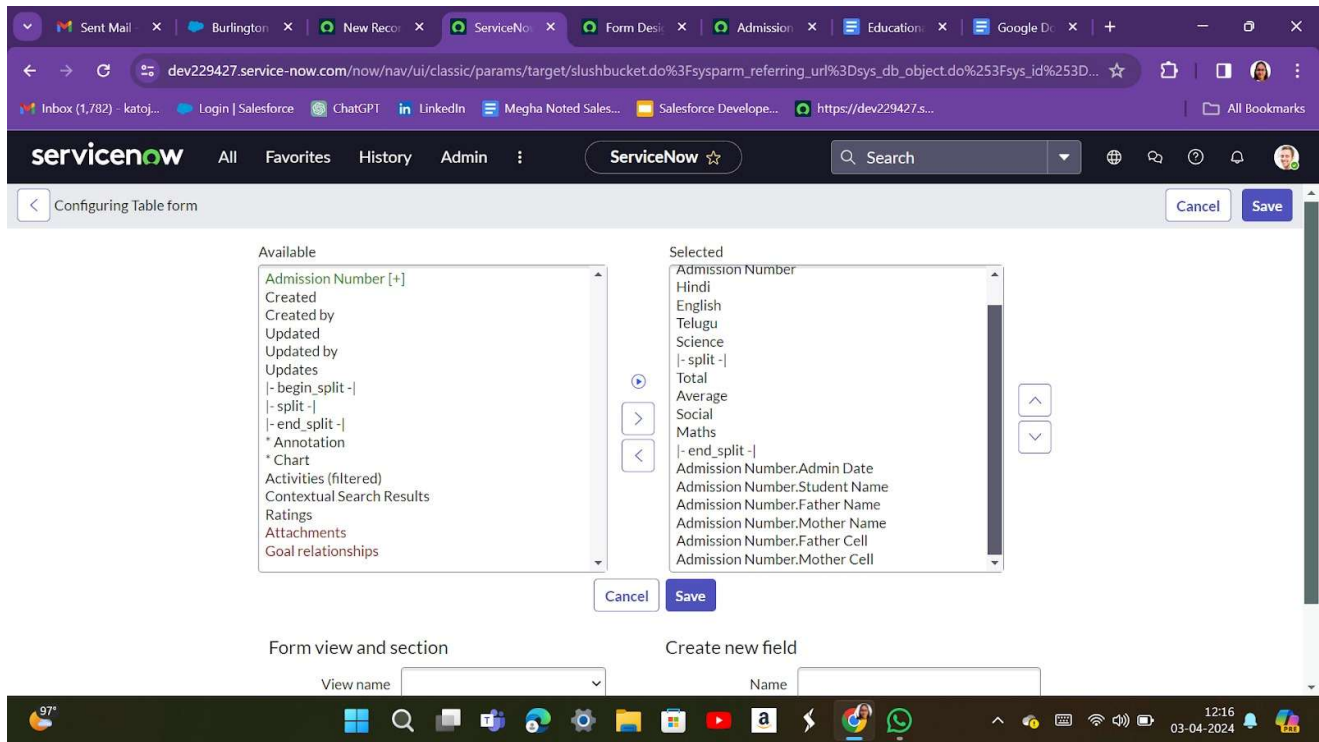
Form view and section

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.

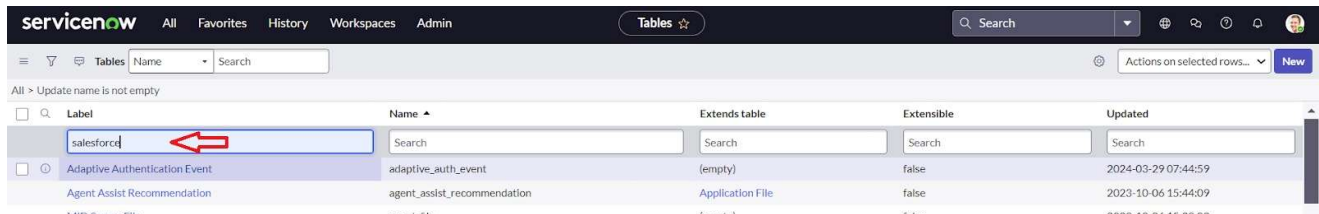


▪ Module 4: Form Layout

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 the 'Table - Salesforce' configuration. A right-click context menu is open, and the 'Configure' option is selected, leading to a sub-menu where 'Form Design' is highlighted. A red arrow points to the 'Form Design' option.

| Column label | Type | Reference | Max length | Default | Display |
|--------------|-------------------|-----------|------------|--------------------------------------|---------|
| Class | System Class Name | (empty) | 80 | javasc | 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:getNextObjNumberPadded(); | true |

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

The screenshot shows the 'Form Design' page in ServiceNow. The 'Table [sys_db_object]' dropdown is open, and 'Salesforce (u_salesforce)' is selected. A red arrow points to this selection. The main form design area shows various fields and controls.

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

The screenshot shows the 'Form Design' page in ServiceNow. The 'Salesforce (u_salesforce)' dropdown is selected. The main form design area shows various fields and controls, including 'Admin Number', 'Admin Date', 'Grade', 'Student Name', 'Father Name', 'Mother Name', 'Father Cell', and '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' table. The left sidebar contains a 'Fields' list with options like 'Admin Number', 'Class', 'Created', 'Created by', 'Updated', 'Updated by', and 'Updates'. Below this is a 'Formatters' section with 'Activities (Filtered)', 'Contextual Search Results', and 'Ratings'. The main area displays the form structure with sections: 'Process Flow (Formatter)', 'Admission Details' (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', 'School Details' (School Area, School), and 'Address' (Pincode, Area, Mandal, City, District, House No).

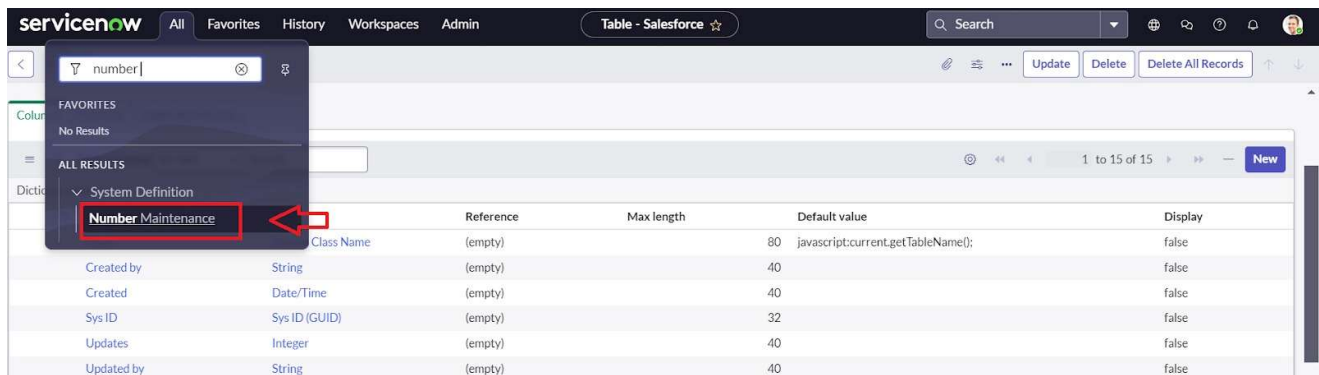
Sub Module 3: Creating Form Design for Student progress Table

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' table. The left sidebar is identical to the previous one. The main area displays the form structure with sections: 'New Section' (Admission Number), 'Admission Details' (Admission Number Grade, Admission Number Student Name, Admission Number Father Name, Admission Number Mother Name, Admission Number Father Cell, Admission Number Mother Cell), and 'Student Progress' (Telugu, Hindi, English, Maths, Science, Total, Percentage, Result).

▪ Module 5: Number Maintenance

Step 1: All >> Number Maintenance >> New



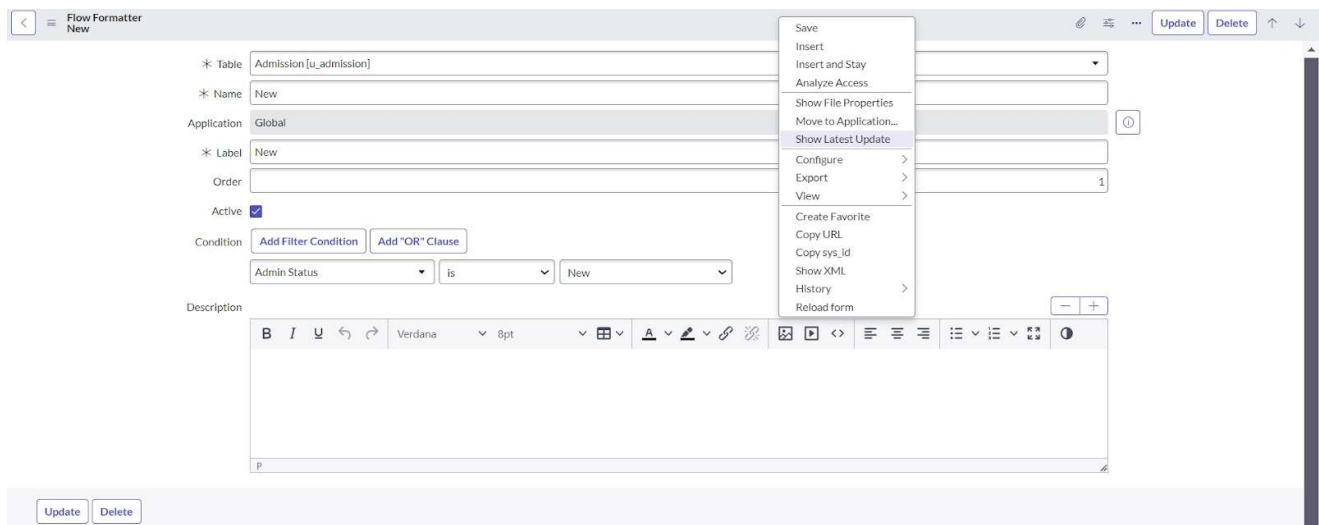
Step 2: Fill the details >> Submit.



▪ Module 6: Process Flow

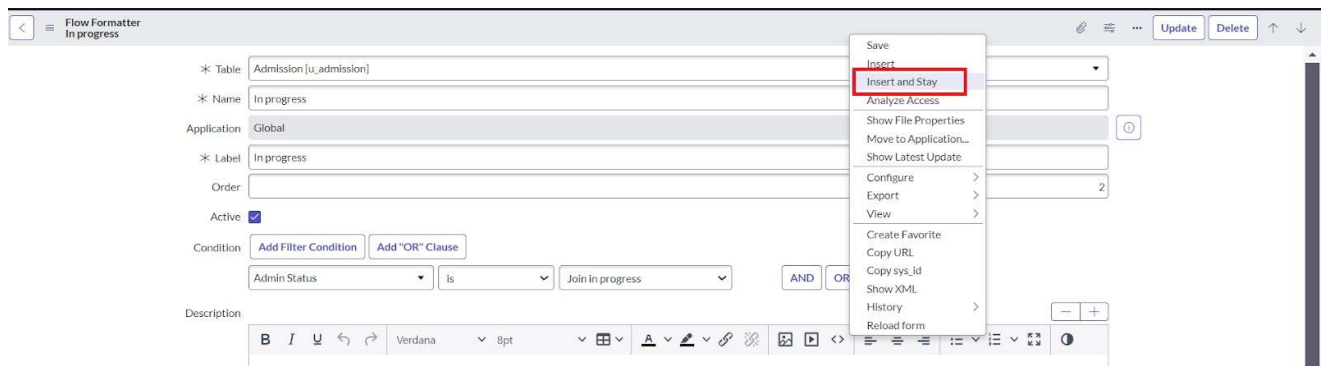
Step 1: All >> Process Flow>> New.

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: Auto populate

Table: Admission [u_admission]

UI Type: Mobile / Service Portal

Type: onChange

Field name: Admin Number

Application: Global ⓘ

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 field named 'Pincode Update'. The configuration includes the following details:

- 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.

This form has annotations - click ⓘ 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: ☒

UI Type: All Inherited: ☐

Type: onChange Global: ☒

Field name: Social

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   if (newValue){
7     var a = parseInt(g_form.getValue('u_telugu'));
8     var b = parseInt(g_form.getValue('u_hindi'));
9     var c = parseInt(g_form.getValue('u_english'));
10    var d = parseInt(g_form.getValue('u_maths'));
11    var e = parseInt(g_form.getValue('u_science'));
12    var f = parseInt(g_form.getValue('u_social'));
13    var Total = parseInt(a+b+c+d+e+f);
14    g_form.setValue('u_total', Total);
15  }
16 }
17

```

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 a record named 'Result'. The 'Table' is set to 'Student Progress[u_student_progress]', 'UI Type' is 'All', 'Type' is 'onChange', and 'Field name' is 'Percentage'. The 'Application' is 'Global', 'Active' is checked, 'Inherited' is unchecked, and 'Global' is checked. The 'Script' section 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 configuration fields are as follows:

- Name:** Percentage
- Table:** Student Progress[u_student_progress]
- UI Type:** All
- Type:** onChange
- Field name:** Total
- Application:** Global
- Active:** ☒
- Inherited:** ☐
- Global:** ☒

The **Script** section 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   var total = g_form.getValue('u_total');
8   var Percentage = (total/600)*100;
9   g_form.setValue('u_percentage',Percentage+'%');
10 }

```

The **Isolate script** checkbox is checked.

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

Submit

Admin Number

Father Name

Admin Date

Mother Name

Grade

Mother Cell

Student Name

Father Cell

Submit

< Admission
New record

Submit

New In progress Joined Rejected Rejoined Closed Cancelled

Admission Number

Admin Date

Purpose of Join

Grade

Student Name

Fee

Father Name

Father Cell

Mother Name

Mother Cell

Admin Status

Comments

School Details Address

School Area

School

Submit

< New Section
New record

Submit

Admission Number

Father Name

Grade

Mother Name

Student Name

Father Cell

Mother Cell

Student Progress

Telugu

Hindi

English

Maths

Science

Social

Total

Percentage

Result

Submit

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.

