

PROJECT REPORT

Educational Organisation Using ServiceNow

Team ID: LTVIP2026MIDS36010

Team Members: 04

Team Leader: Kanna Priyadarshini

Team Member: Kondu Shesikala

Team Member: Ummadi Ramya Sree

Team Member: Dumpal Sindhu

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.

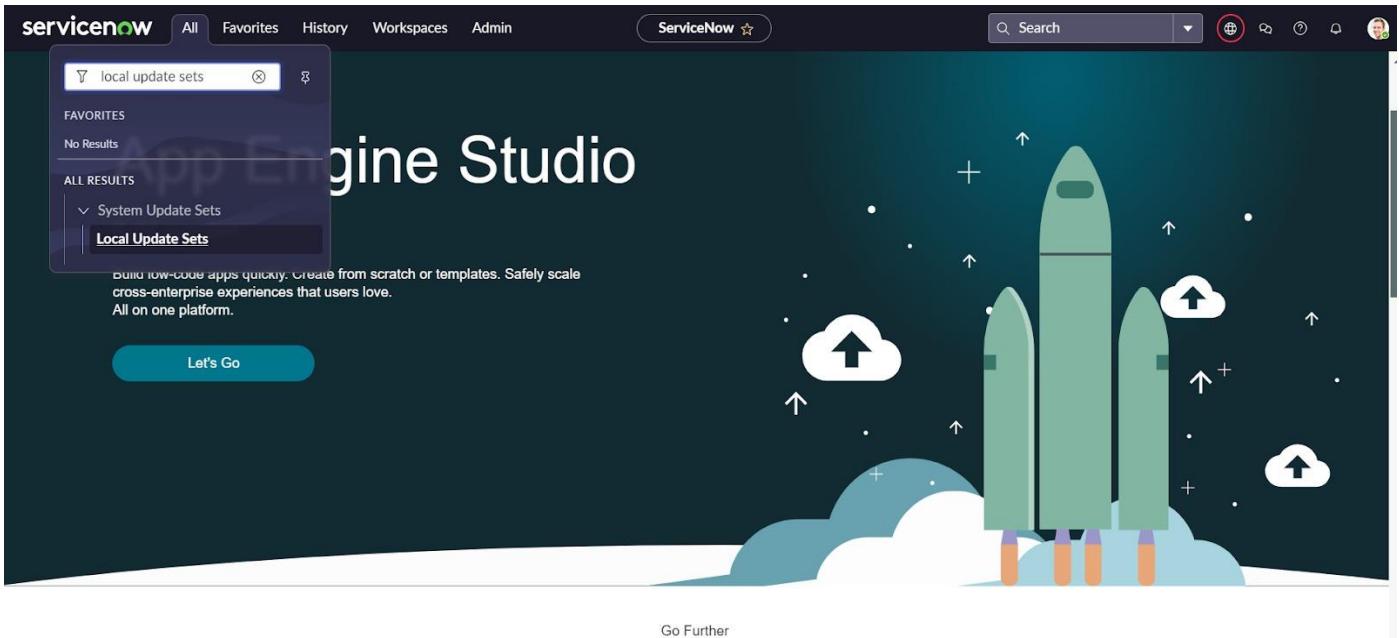
Objective: The project aims to develop a system for schools and colleges to manage admissions, student details, and academic progress, making the entire process quick, paperless, and user-friendly.

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.

This screenshot shows the "Update Sets" list view in ServiceNow. The top navigation bar includes "Update Sets" and a "New" button, which is highlighted with a red box and an arrow. The main area displays a table with columns: Name, Application, State, Installed from, Created, Created by, Parent, and Batch Base. Two rows are visible: one for "Default" (Application: App Engine Studio, State: In progress) and another for "Default" (Application: Global, State: In progress). The "Actions on selected rows..." button is also highlighted with a red box and an arrow.

Step 2: Click on New

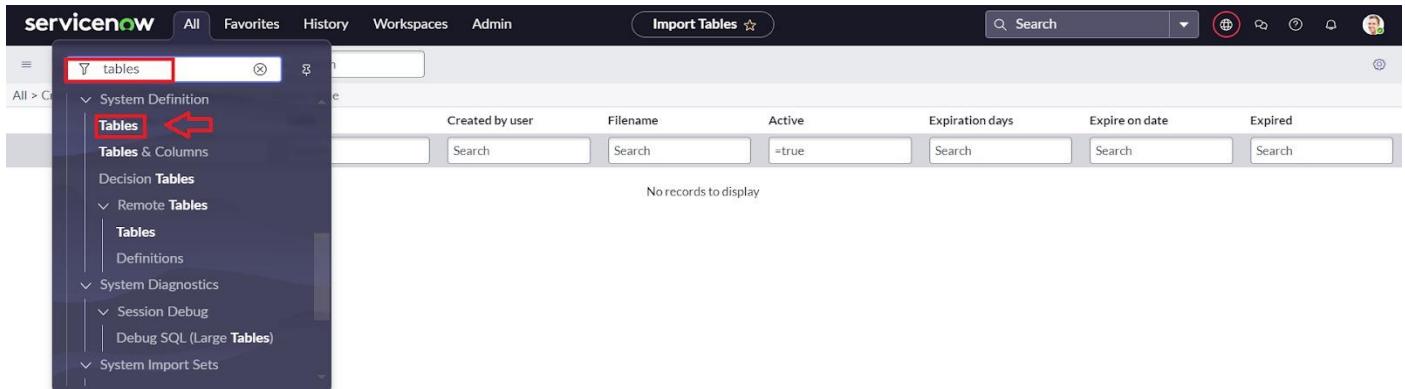
This screenshot shows the "Update Set - Create New Update Set" form. The top navigation bar includes "Update Set" and "New record". The form fields include: Name (set to "New Update Set"), State (set to "In progress"), Parent (empty), Release date (empty), and Description (empty). At the bottom, there are "Submit" and "Submit and Make Current" buttons.

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

▪ Module 2: Creating A Table

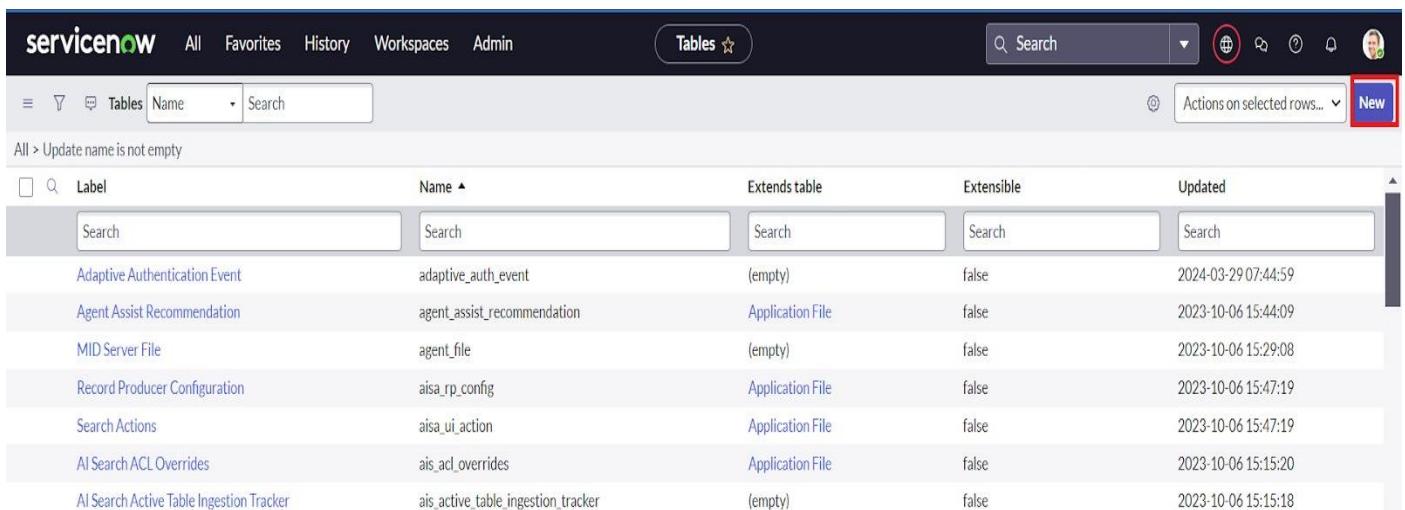
Sub Module 1: Creating Salesforce Table.

Step 1: All>>Tables



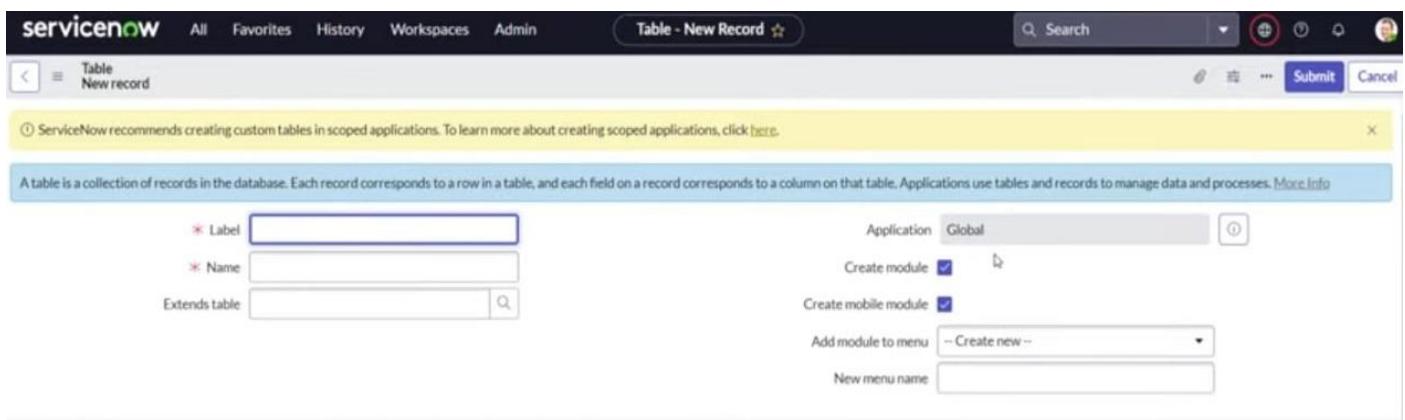
The screenshot shows the ServiceNow web interface. The top navigation bar includes links for 'All', 'Favorites', 'History', 'Workspaces', and 'Admin'. A search bar with the placeholder 'Search' is located on the right. Below the navigation is a sidebar with a tree view. The 'Tables' node under 'System Definition' is highlighted with a red box and an arrow pointing to it. The main content area displays a table with columns: 'Created by user', 'Filename', 'Active', 'Expiration days', 'Expire on date', and 'Expired'. A message at the bottom says 'No records to display'.

Step 2: Click On New



The screenshot shows the 'Tables' list page. The top navigation bar and search bar are visible. The main content area shows a table with columns: 'Label', 'Name', 'Extends table', 'Extensible', and 'Updated'. The 'Label' column contains entries like 'Adaptive Authentication Event', 'Agent Assist Recommendation', etc. The 'Name' column contains entries like 'adaptive_auth_event', 'agent_assist_recommendation', etc. The 'Extends table' column contains '(empty)'. The 'Extensible' column contains 'false'. The 'Updated' column contains dates like '2024-03-29 07:44:59'. A blue box highlights the 'New' button in the top right corner of the table header.

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



The screenshot shows the 'Table - New Record' form. The top navigation bar and search bar are present. The main form has fields for 'Label' (with a red asterisk), 'Name' (with a red asterisk), and 'Extends table'. To the right, there are options for 'Application' (set to 'Global'), 'Create module' (checkbox checked), 'Create mobile module' (checkbox checked), 'Add module to menu' (dropdown set to '- Create new -'), and 'New menu name' (text input field). A yellow banner at the top provides a recommendation about creating scoped applications.

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.

servicenow All Favorites History Workspaces Admin Table - New Record ⭐

Table New record

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

* Label

* Name

Extends table

Columns * Controls Application Access

Table Columns for text Search

Dictionary Entries

	Column label	Type	Reference	Max length
+	<input style="border: 2px solid red; width: 150px; height: 20px;" type="text"/>	<input checked="" type="checkbox"/> <input type="radio"/> <input type="button" value="X"/>		

Submit Cancel



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

Admin Date	Date	(empty)	40	false
Admin Number	String	(empty)	40	javascript:getNextObjNumberPadded(); true
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

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(); true
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

Columns Controls Application Access

Table Columns for text Search

Dictionary Entries

Column label	Type	Reference	Max length	De
Class	System Class Name	(empty)	80	jav true
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

Save Analyze Access

- Show File Properties
- Move to Application...
- Show Latest Update
- Create Child Table
- Show Dictionary Record
- Configure > Display false
- Export > false
- View > false
- Create Favorite
- Copy URL
- Copy sys_id
- Show XML
- History
- Reload form

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.

	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

Sub Module 2: Creating Admission Table

Step 1: Create an Admission Table with the 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' screen in ServiceNow. At the top, there are tabs for 'All', 'Favorites', 'History', 'Workspaces', and 'Admin'. A search bar says 'Search' with a dropdown arrow. Below the header, it says 'Table - New Record' with a star icon. On the left, there's a back arrow and a 'Table New record' button. On the right, there are 'Submit' and 'Cancel' buttons. A yellow message bar at the top says 'ServiceNow recommends creating custom tables in scoped applications. To learn more about creating scoped applications, click here.' A blue message bar below it says '(i) This form has annotations - click (i) to toggle them - (click here) to never show this again.' The main form area has fields for 'Label' (Admission), 'Name' (u_admission), 'Extends table' (Salesforce, highlighted with a red box), 'Application' (Global), 'Create module' (checked), 'Create mobile module' (checked), and 'Add module to menu' (Salesforce, highlighted with a red box). Below this, there are tabs for 'Columns', 'Controls', and 'Application Access'. Under 'Columns', there's a table with columns for 'Column label', 'Type', 'Reference', 'Max length', 'Default value', and 'Display'. The first row has a plus sign and a blank input field. At the bottom are 'Submit' and 'Cancel' buttons.

The screenshot shows the 'Table' screen for the 'Admission' table in ServiceNow. At the top, there are tabs for 'Columns', 'Controls', and 'Application Access'. A search bar says 'Search' with a dropdown arrow. On the right, there are buttons for 'Update', 'Delete', and 'Delete All Records'. Below the header, it says 'Table Admission'. The main area is a table of fields. The columns are 'Column label', 'Type', 'Reference', 'Max length', 'Default value', and 'Display'. The rows include: Sys ID (Sys ID (GUID)), Admin Status (Choice), Admission Number (Reference to Salesforce), Area (String), City (String), Comments (String (Full UTF-8)), District (String), Fee (Price), House No (String), Mandal (String), Pincode (Choice), Purpose of join (Choice), School (Choice, highlighted with a red box), School Area (Choice), and Class (System Class Name). The 'Display' column for the 'School' field is set to 'false'.

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.getTableName();	false

Step 4: Create choice for Admin Status as:

Dictionary Entry
Admin Status

Create Choice List Delete Column Update

Related Links
Show Table Run Point Scan Advanced view

Access Controls Choices (7) Attributes Labels (1) Dictionary Overrides

Label Search Actions on selected rows... New

Choices

<input type="checkbox"/>	Label	Value	Language	Sequence	Inactive	Updated
	New	New	en	1	false	2024-04-02 21:10:25
	Join in progress	In progress	en	2	false	2024-04-02 21:11:03
	Joined	Joined	en	3	false	2024-04-02 21:11:26
	Rejected	Rejected	en	4	false	2024-04-02 21:12:00
	Closed	Closed	en	5	false	2024-04-02 21:13:05
	Rejoined	Rejoined	en	6	false	2024-04-02 21:13:08
	Cancelled	Cancelled	en	7	false	2024-04-02 21:13:27

+ Insert a new row...

Step 5: Create choice for Pin code as:

Access Controls Choices (3) Attributes Labels (1) Dictionary Overrides

Label Search Actions on selected rows... New

Choices

<input type="checkbox"/>	Label	Value	Language	Sequence	Inactive	Updated
	509358	509358	en	1	false	2024-04-02 21:15:19
	500079	500079	en	2	false	2024-04-02 21:15:46
	500081	500081	en	3	false	2024-04-02 21:16:05

+ Insert a new row...

Step 6: Create choice for Purpose of Join as:

Access Controls Choices (3) Attributes Labels (1) Dictionary Overrides

Label Search Actions on selected rows... New

Choices

<input type="checkbox"/>	Label	Value	Language	Sequence	Inactive	Updated
	Tuition	Tuition	en	1	false	2024-04-02 21:17:09
	Coaching	Coaching	en	2	false	2024-04-02 21:17:31
	Teacher	Teacher	en	3	false	2024-04-02 21:17:53

+ Insert a new row...

Step 7: Create choice for School as:

Access Controls Choices (2) Attributes Labels (1) Dictionary Overrides

Label Search Actions on selected rows... New

Choices

<input type="checkbox"/>	Label	Value	Language	Sequence	Inactive	Updated
	Stanley	Stanley	en	1	false	2024-04-02 21:19:14
	Naresh It	Naresh It	en	2	false	2024-04-02 21:19:35

+ Insert a new row...

Step 8: Create choice for School Area as:

Access Controls Choices (2) Attributes Labels (1) Dictionary Overrides

Label Search Actions on selected rows... New

Choices

<input type="checkbox"/>	Label	Value	Language	Sequence	Inactive	Updated
	Near Market	Near Market	en	1	false	2024-04-02 21:20:53
	Near Bus Stand	Near Bus Stand	en	2	false	2024-04-02 21:21:24

+ Insert a new row...

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:

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

The screenshot shows the 'Table student progress' page. At the top, there is a table with two rows: 'Telugu' and 'Total'. Below the table, there is a button labeled 'Insert a new row...'. At the bottom of the page, there are three buttons: 'Update', 'Delete', and 'Delete All Records'. On the right side, under 'Related Links', the 'Layout Form' link is highlighted with a purple border. Other links include 'Design Form', 'Layout List', 'Show Form', 'Show List', 'Show Schema Map', 'Add to Service Catalog', 'Run Point Scan', and 'Explore REST API'.

Step 2: Click on Admission Number [+].

The screenshot shows the 'Configuring Table form' dialog box. It has two main sections: 'Available' and 'Selected'. The 'Available' section contains fields like 'Admission Number [+]', 'Created', 'Created by', 'Updated', 'Updated by', 'Updates', '|- begin_split -|', '|- split -|', '|- end_split -|', '* Annotation', '* Chart', 'Activities (filtered)', 'Contextual Search Results', 'Ratings', 'Attachments', and 'Goal relationships'. The 'Selected' section contains fields like '|- begin_split -|', 'Admission Number', 'Hindi', 'English', 'Telugu', 'Science', '|- split -|', 'Total', 'Average', 'Social', 'Maths', and '|- end_split -|'. At the bottom, there are 'Cancel' and 'Save' buttons. Below the dialog, the 'Form view and section' and 'Create new field' tabs are visible, along with a toolbar and system status icons at the bottom.

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

The screenshot shows the ServiceNow 'Configuring Table form' dialog. On the left, under 'Available', there is a list of various fields and operations such as 'Admission Number', 'Created', 'Updated', 'Activities (filtered)', and 'Annotations'. On the right, under 'Selected', there is a list of fields that have been moved from the available list, including 'Admission Number', 'Hindi', 'English', 'Telugu', 'Science', 'Maths', and several composite fields like 'Admission Number.Admin Date', 'Admission Number.Student Name', etc. At the bottom of the dialog are 'Cancel' and 'Save' buttons.

▪ 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.

The screenshot shows the ServiceNow 'Tables' list view. The 'Label' column header has a red arrow pointing to it, indicating the search term 'salesforce' which is typed into the search input field. The table lists several tables: 'Adaptive Authentication Event' (label: 'adaptive_auth_event'), 'Agent Assist Recommendation' (label: 'agent_assist_recommendation'), and others. Columns include 'Name', 'Extends table', 'Extensible', and 'Updated'.

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

The screenshot shows the ServiceNow 'Table - Salesforce' page. A context menu is open at the top right, with 'Configure' selected. Under 'Configure', 'Form Design' is highlighted with a red arrow. The main table view shows columns like 'Label', 'Type', 'Reference', 'Max length', and 'Default'. A sidebar on the left lists 'Table Columns' and 'Dictionary Entries'.

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

The screenshot shows the 'Form Design' editor. The left sidebar is titled 'Fields' and shows a list of fields. A red arrow points to the search bar in the sidebar where 'salesforce' is typed. The main area displays the 'Table [sys_db_object]' configuration, with the 'Salesforce [u_salesforce]' table selected. The configuration screen includes sections for 'Annotation', 'Columns', and 'Controls'.

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

The screenshot shows the 'Form Design' editor with the 'Fields' panel open on the left. A red arrow points to the 'Salesforce [u_salesforce]' table in the list. The main area shows a grid of fields being mapped to controls. The fields listed are Admin Number, Admin Date, Grade, Student Name, Father Name, Mother Name, Father Cell, and Mother Cell. Each field has a corresponding control and mapping options.

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 left sidebar contains sections for 'Fields' (with 'Admin Number' selected), 'Field Types', 'Filter', 'Formatter' (with 'Activities (filtered)' selected), 'Contextual Search Results', and 'Ratings'. The main area displays the form structure:

- Process Flow (Formatter):** Contains fields for 'Admission Number', 'Purpose of join', 'Student Name', 'Father Name', 'Mother Name', 'Comments', and 'School Details' (with 'School Area').
- Address:** Contains fields for 'Pincode', 'Mandal', and 'House No', which map to 'Area', 'City', and 'District' respectively.
- Admin Data:** Contains fields for 'Grade', 'Fee', 'Father Cell', 'Mother Cell', and 'Admin Status'.

Sub Module 3: Creating Form Design for Student progress Table

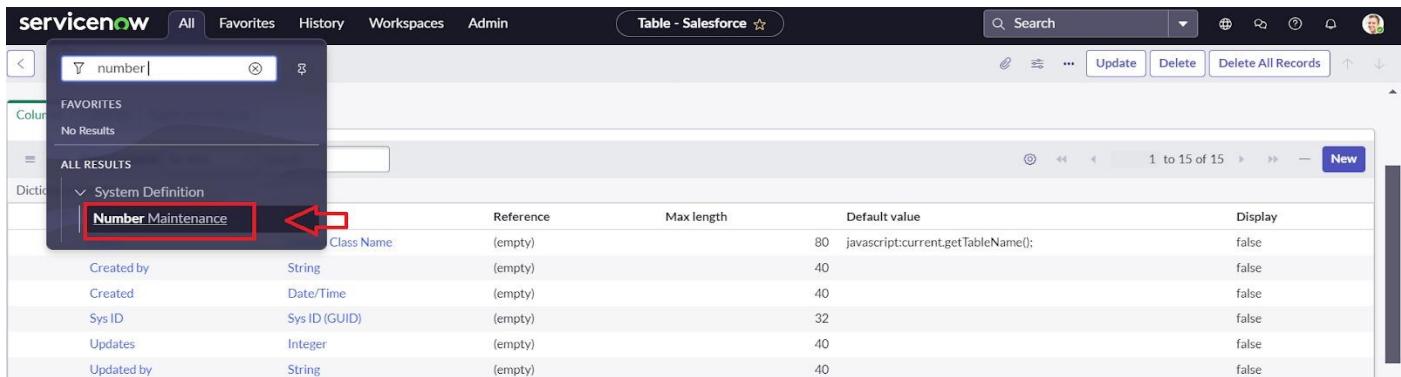
Step 1: Follow the same steps as Activity1, Configure the fields as below and save.

The screenshot shows the 'Form Design' interface for the 'Student Progress [u_stude]' table. The left sidebar contains sections for 'Fields' (with 'Class' selected), 'Field Types', 'Filter', 'Formatter' (with 'Activities (filtered)' selected), 'Contextual Search Results', and 'Ratings'. The main area displays the form structure:

- New Section:** Contains the field 'Admission Number'.
- Student Progress:** Contains fields for 'Telugu', 'Hindi', 'English', 'Maths', 'Science', 'Total', 'Percentage', and 'Result'.

▪ Module 5: Number Maintenance

Step 1: All >> Number Maintenance >> New



The screenshot shows a ServiceNow search interface for 'Number Maintenance'. A red box highlights the 'Number Maintenance' entry in the search results list. An arrow points from the text 'Number Maintenance' in the 'Step 1' section to this highlighted entry.

Class Name	Reference	Max length	Default value	Display
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

Step 2: Fill the details >> Submit.



The screenshot shows a ServiceNow form for creating a new 'Number' record. A red box highlights the 'Number' input field, which contains 'SAL'. An arrow points from the text 'Number SAL' in the 'Step 2' section to this highlighted field.

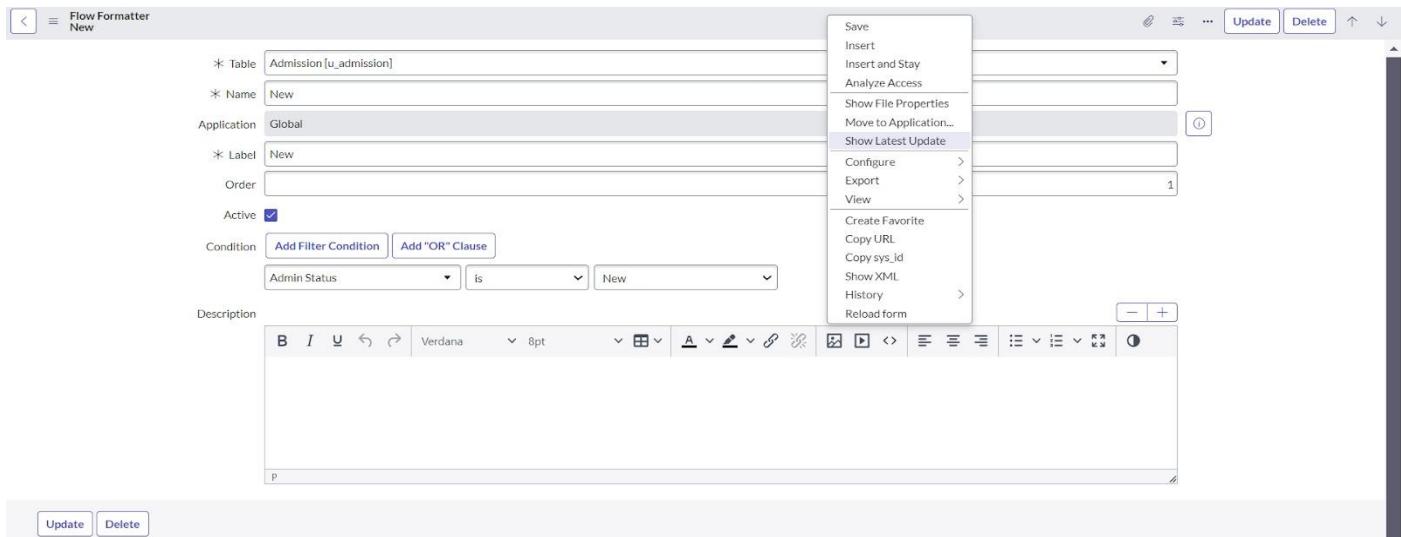
* Table	Salesforce
Prefix	SAL.
* Number	1.000
Application	Global
Number of digits	7

Buttons at the bottom: Update, Delete.

▪ Module 6: Process Flow

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

Step 2: Fill the Details as given Below



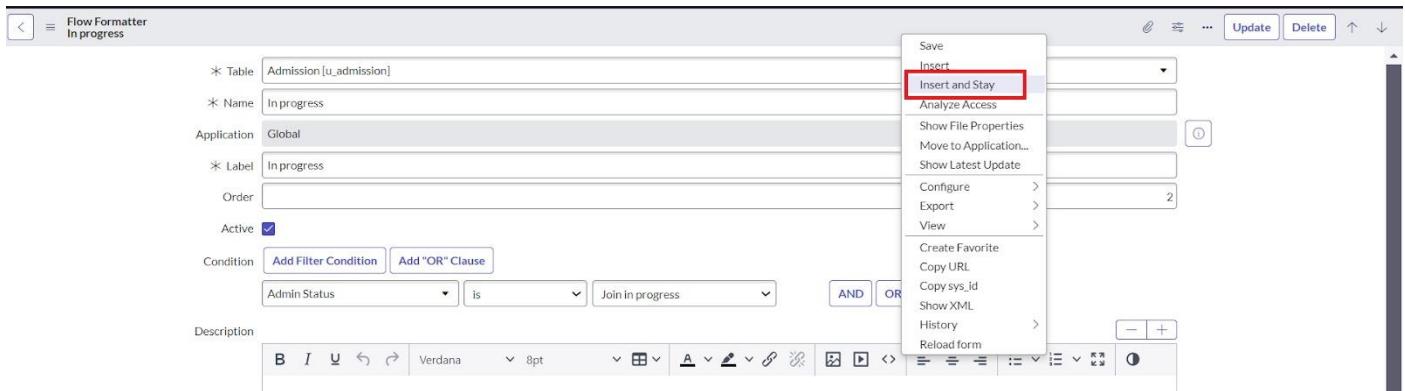
The screenshot shows a ServiceNow form for creating a new 'Flow Formatter' record. A red box highlights the 'Name' input field, which contains 'New'. An arrow points from the text 'New' in the 'Step 2' section to this highlighted field.

* Table	Admission [u_admission]
* Name	New
Application	Global
* Label	New
Order	
Active	<input checked="" type="checkbox"/>
Condition	Add Filter Condition Add "OR" Clause
Admin Status is New	
Description	<p>P</p>

Context menu open on the right side of the form, showing options like Save, Insert, Analyze Access, etc. Buttons at the bottom: Update, Delete.

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.

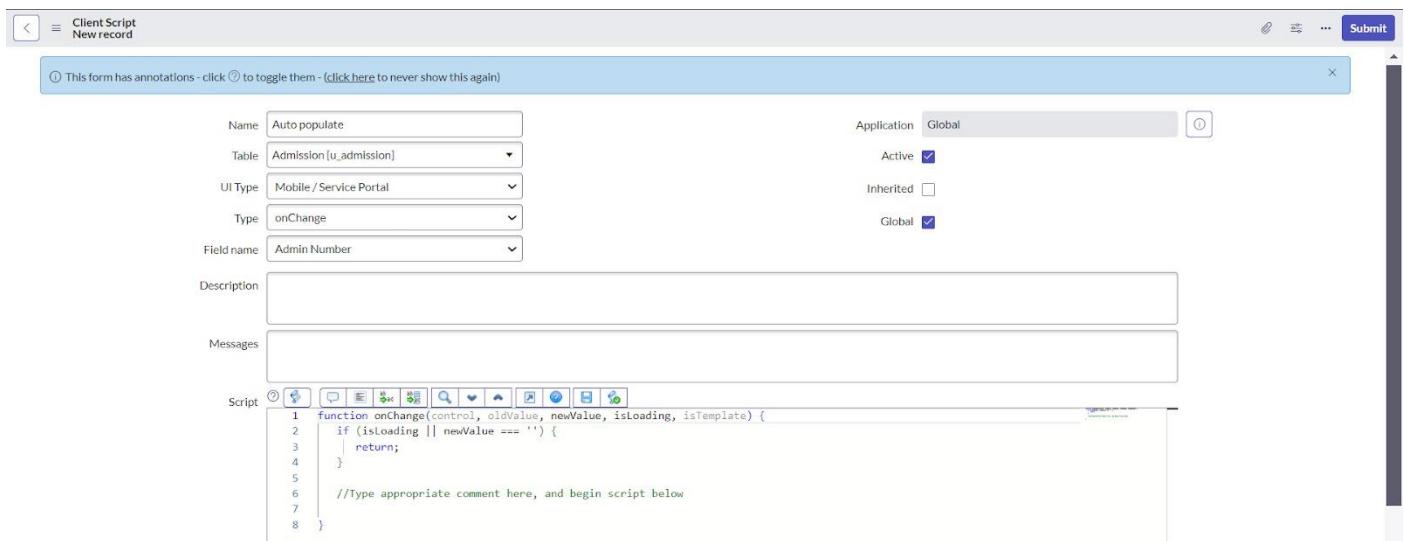
▪ Module 7: Client Script

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.

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

Name: Pincode Update
Table: Admission [u_admission]
UI Type: Desktop
Type: onChange
Field name: Pincode

Application: Global
Active:
Inherited:
Global:

Description:

Messages:

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');
}
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');
}

```

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');

}

```

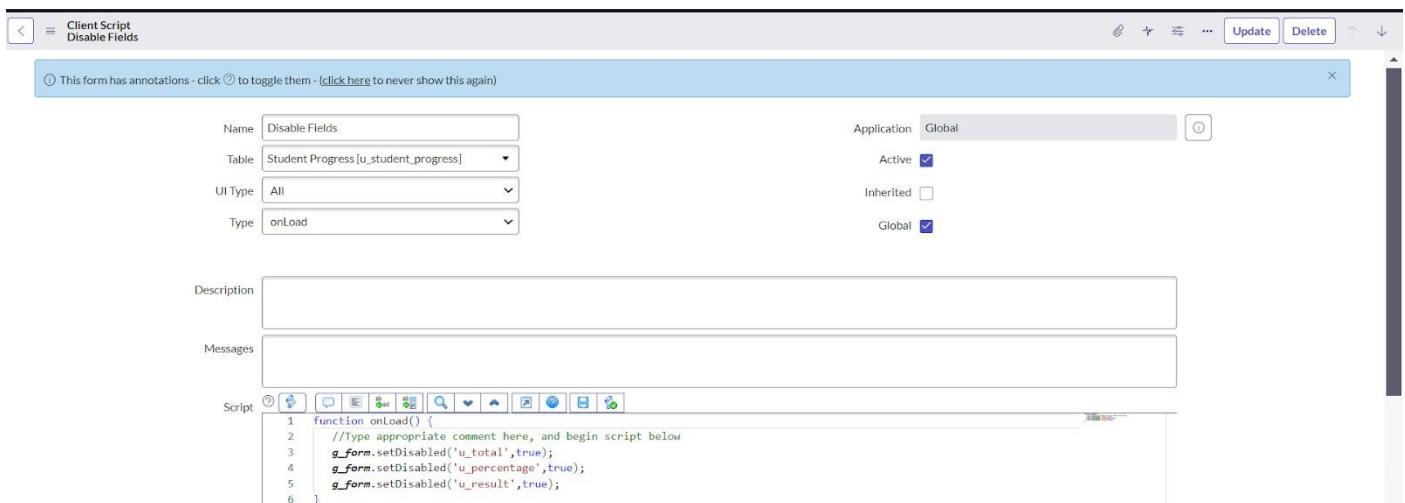
```

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.

The screenshot shows the configuration of a Client Script named "Total Update" for the "Student Progress [u_student_progress]" table. The script type is "onChange" and it is associated with the "Social" field. It is set to run at the "All" UI level and is a global script. The "Active" checkbox is checked, while "Inherited" and "Isolate script" are unchecked.

```

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);
    }
}

```

```

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 configuration of a client script named 'Result' for the 'Student Progress' table. The script type is 'onChange' for the 'Percentage' field. It contains logic to check if the value is between 0 and 59, setting 'u_result' to 'Fail'. If between 60 and 100, it sets 'u_result' to 'Pass'. Otherwise, it handles out-of-range values by adding an error message and clearing the result field.

```

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             g_form.addErrorMessage('Percentage should be between 0 and 100.');
15             g_form.clearValue('u_result');
16         }
17     }
18 }

```

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 'Client Script Percentage' configuration page in ServiceNow. The 'Name' field is set to 'Percentage'. The 'Table' is 'Student Progress [u_student_progress]'. The 'UI Type' is 'All'. The 'Type' is 'onChange'. The 'Field name' is 'Total'. The 'Application' is 'Global'. The 'Active' checkbox is checked. The 'Inherited' checkbox is unchecked. The 'Global' checkbox is checked. The 'Script' section contains the following code:

```
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+'%');
}
```

The 'Isolate script' checkbox is checked. 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+'%');
}
```

Output:

The screenshot shows the 'Salesforce - Create SAL0001005' page. The 'Admin Number' field is 'SAL0001005'. The 'Admin Date' field has a calendar icon. The 'Grade' dropdown is set to '-- None --'. The 'Student Name' field is empty. On the right side, there are fields for 'Father Name' and 'Mother Name', both of which are empty. Below these, there are fields for 'Father Cell' and 'Mother Cell', also empty. At the bottom left is a 'Submit' button.

Servicenow All Favorites History Workspaces ... Student Progress - Create SAL0001007 ★

New Section New record

Admission Number

Grade -- None --

Student Name

Father Name

Mother Name

Father Cell

Mother Cell

Student Progress

Telugu

Hindi

English

Maths

Science

Total

Percentage

Result

Submit

Advantages & Disadvantages:

Advantages of the Project:

- This project simplifies and enhances the efficient management of student and teacher data through centralized and automated workflows.
 - It simplifies the student admission process by automating and streamlining each step, from application to enrollment.
 - All student, faculty, and institutional data is stored accurately and securely.
 - Minimizes physical documentation and promotes a paperless environment.
 - It helps track student attendance and monitor their academic progress effectively.

Disadvantages of the Project:

- Entering records into the system requires a stable internet connection within the educational institution.
- Sensitive data could be at risk if access controls and encryption are not properly managed.
- Consistent system updates and maintenance are necessary to keep the platform running efficiently.
- Updating multiple student and faculty records can take considerable time, especially when dealing with large volumes of data.

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.

