

Educational Organisation Using ServiceNow

1. Project Overview

The Educational Management System (EMS) is a streamlined solution built on the ServiceNow platform to enhance administrative efficiency within educational institutions. It manages student and teacher data, simplifies the admission process, and provides tools for tracking academic progress. By implementing EMS in ServiceNow, institutions benefit from a user-friendly, customizable, and automated environment that supports better decision-making and operational management.

2. Setting Up the ServiceNow Instance

Sign Up for a Developer Account

- ⑩ Visit the ServiceNow Developer Portal at <https://developer.servicenow.com>.
- ⑩ Create a new developer account by providing the required information.

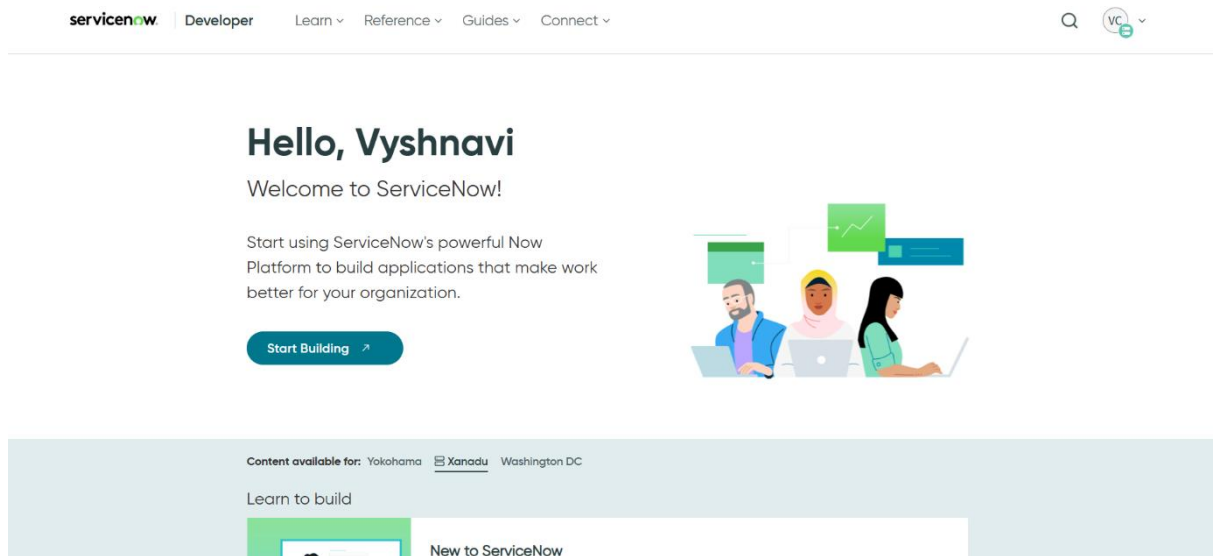
Request a Personal Developer Instance

- ⑩ Log in to your developer account.
- ⑩ Navigate to the “Manage > Instance” section.
- ⑩ Click “Request Instance” and choose the latest available release.
- ⑩ You will receive an email with the instance details (URL, username, and password).

Access Your Instance

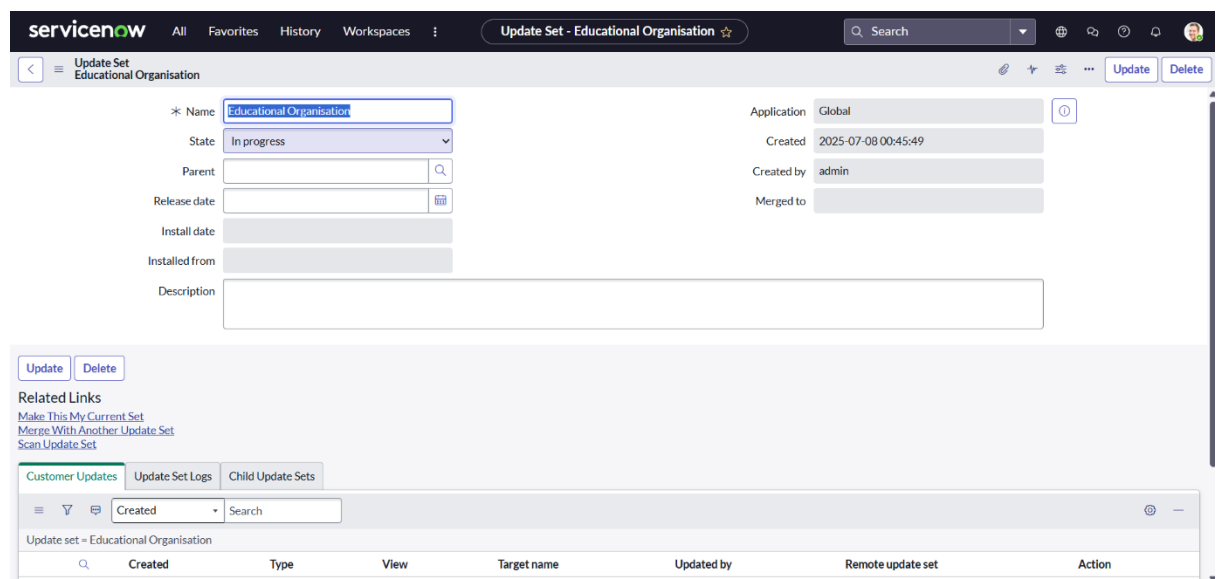
- ⑩ Open the instance URL received via email.

- 10 Log in using the provided credentials to access your personal ServiceNow instance.



3. Creating an Update Set

1. Navigate to All → Local Update Sets from the Application Navigator.
2. Click on New to create a new update set.
3. Enter the name as: Educational Organisation.
4. Click Submit and then Make Current to activate it.



4. Creating Tables

Creating Salesforce Table

1. Navigate to All → Tables.
2. Click on New to create a new table.
3. Enter the label: Salesforce. The API name will be generated automatically.
4. Add the required columns:
 - Admin Number (Display: True)
 - Grade (with choice values and sequence)
 - Other necessary fields as per requirement



The screenshot shows the configuration page for a table named 'Table Salesforce'. At the top, there is a warning message: 'This record is in the Global application, but Educational Organisation is the current application. To edit this record click here.' Below this is a table with 5 columns. The first column lists field names, the second lists data types, the third shows current values (all empty), the fourth shows lengths (all 40), and the fifth shows whether the field is displayed (true for Admin Number, false for others).

Field Name	Data Type	Value	Length	Display
Admin Date	Date	(empty)	40	false
Admin Number	String	(empty)	40	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

5. Enable Extensible from the Controls section.
6. For the Admin Number field:
 - Go to Advanced View in Related Links.
 - Enable Use Dynamic Default
 - Select Get Next Padded Number for default value
7. For the Grade field:
 - Set up choices with appropriate labels, values, and sequence.

The screenshot shows the ServiceNow 'Dictionary Entry - Grade' interface. It features a table with columns: Label, Value, Language, Sequence, Inactive, and Updated. The table lists grades from Prept to X, each with a corresponding value, language (en), sequence number, and an 'Inactive' status of false. The 'Updated' column shows timestamps for each entry.

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

Creating Admission Table

1. Create a new table and name it Admission.
2. Set Extends Table to Salesforce.
3. Check the box for Add module to menu and select Salesforce.
4. Add fields as required

The screenshot shows the ServiceNow 'Table Admission' interface. It displays a list of columns to be added to the table, with columns: Column label, Type, Reference, Max length, Default value, and Display. The list includes fields like Sys ID, Admin Status, Admission Number, Area, City, Comments, District, Fee, House No, Mandal, Pincode, Purpose of join, School, School Area, and Class.

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

5. Add necessary fields such as Admission Number, Grade, School, and Pincode.
6. Create choice fields for Admin Status, Purpose of Join, School, Pincode, and School Area.

Creating Student Progress Table

1. Create a table named Student Progress.
2. Enable Add module to menu and assign to Salesforce.
3. Add necessary fields.

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...				

5. Form Layout

Configuring Table Form for Student Progress Table

1. Navigation:

Open the Student Progress Table from the Application Navigator.

2. Open Form Layout:

On the table configuration page, click on Layout (found under the related links or in the top-right corner depending on UI version).

3. Expand Admission Number Field:

- Find the Admission Number field in the form layout.
- Click the [+] icon beside it to expand and view the related fields.

4. Move Fields to the Form:

- In the Available column, locate the relevant fields under Admission Number (e.g., Student Name, Grade, Admin Date, etc.).
- Move the required fields to the Selected column using the arrow button.
- These fields will now appear on the Student Progress form for reference.

5. Save the Layout:

Click Save to apply the new layout settings.

6. Creating Form Design

Creating Form Design for Salesforce Table

1. Navigation:

Go to All → System Definition → Tables.

2. Search and Open Table:

- In the Label field, search for Salesforce.
- Click to open the Salesforce Table (u_salesforce).

3. Open Form Designer:

- Right-click on the header toggle bar of the record.
- Select Configure → Form Design.

4. Select the Form Layout:

- In the form designer dropdown, ensure Salesforce (u_salesforce) is selected.

5. Design the Form:

- Drag and drop the required fields from the right-side panel to the form layout area on the left.
- Arrange the fields logically (e.g., Admin Number, Grade, Admission Date, etc.).

6. Save the Form:

Click Save to apply the design.

The screenshot shows the Salesforce Form Design interface. At the top, there's a header with 'Salesforce [u_salesforce]', 'Default view', and 'Form Design'. Below this, there's a sidebar with 'Fields' and 'Field Types' tabs. The 'Fields' tab is active, showing a filter and a list of fields: Class, Created, Created by, Updated, Updated by, and Updates. Below this, there's a 'Formatters' section with 'Activities (filtered)', 'Contextual Search Results', and 'Ratings'. The main area displays a table with 2 columns. The table has 6 rows of fields: Admin Number, Admin Date, Grade, Student Name, Father Name, Mother Name, Father Cell, and Mother Cell. Each field has a settings icon.

Creating Form Design for Admission Table

Follow the same steps as above (Salesforce Table form design) with the following adjustments:

- Search Table: Search for Admission in the System Definition → Tables.
- Form Fields: Drag fields relevant to admissions such as:
 - Student Name
 - Father Name
 - Mother Name
 - Grade
 - Admin Status
 - School & School Area
 - Purpose of Join
 - Pincode and Location
- Click Save to apply.

The screenshot shows the 'Form Design' interface for the 'Admission [u_admission]' form. The sidebar on the left lists field types: Fields, Field Types, and Filter. The main area displays the form structure with sections: Admission [u_admission], Process Flow (Formatted), School Details, and Address. The Admission section includes fields for Admission Number, Class, Created, Created by, Updated, Updated by, Updates, Admission Date, Grade, Fee, Father Cell, Mother Cell, and Admission Status. The School Details section includes School Area and School. The Address section includes Pincode, Area, Ward, City, and District.

Creating Form Design for Student Progress Table

Repeat the same steps once again for the Student Progress Table:

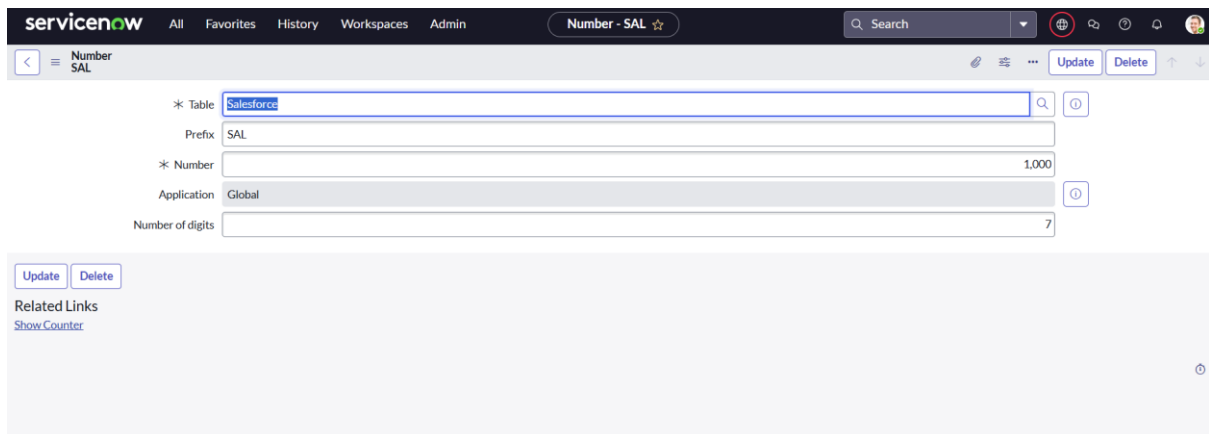
- Search Table: Search for Student Progress.
- Form Fields: Include:
 - Admission Number (reference field)
 - Subject Marks (Telugu, Hindi, English, Maths, Science, Social)
 - Total, Percentage, and Result
- Drag and drop fields in logical sequence.
- Click Save to finalize.

The screenshot shows the 'Form Design' interface for the 'Student Progress [u_student]' form. The sidebar on the left lists field types: Fields, Field Types, and Filter. The main area displays the form structure with sections: New Section, Student Progress, and Student Progress. The New Section includes the Admission Number field. The Student Progress section includes fields for Admission Number Grade, Admission Number Father Name, Admission Number Mother Name, Admission Number Father Cell, Admission Number Mother Cell, Telugu, Hindi, English, Maths, Science, Total, Percentage, and Result.

7. Number Maintenance

Admin Number Setup

1. Navigate to All → Number Maintenance.
2. Click on New.
3. Fill in the details as needed.
4. Click Submit.



The screenshot shows the ServiceNow Number Maintenance form. The top navigation bar includes 'servicenow', 'All', 'Favorites', 'History', 'Workspaces', 'Admin', and a search bar. The breadcrumb trail is 'Number - SAL'. The form fields are: '* Table' (Salesforce), 'Prefix' (SAL), '* Number' (1,000), 'Application' (Global), and 'Number of digits' (7). There are 'Update' and 'Delete' buttons at the top right and bottom left. Below the form, there is a 'Related Links' section with a link to 'Show Counter'.

8. Process Flow

Creating Process Flow for Admission Table

1. Go to All → Process Flow and click on New.
2. Fill in the details.
3. Right-click on the toggle bar and click Save.
4. Replace name and label as required.
5. Use Insert and Stay to add the following states:
 - New → In Progress → Joined → Rejected → Rejoined → Closed → Cancelled

9. Client Scripts

Creating “Auto Populate” Client Script for Admission Table

1. Navigation:

Go to **All → Client Scripts** in the Application Navigator.

2. Create a New Script:

Click on **New**.

3. Fill in the Script Details:

- **Name:** Auto Populate Admission Fields
- **Table:** Admission
- **UI Type:** All
- **Type:** onChange
- **Field name:** Admission Number (u_admission_number)
- **Active:** Checked
- **Isolate Script:** Checked

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
```

4. Script Logic:

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

```
// Fetch the admission record and populate related fields
```

```

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);

// Disable fields after auto-population
g_form.setDisabled('u_admin_date', true);
g_form.setDisabled('u_grade', true);
g_form.setDisabled('u_student_name', true);
g_form.setDisabled('u_father_name', true);
g_form.setDisabled('u_mother_name', true);
g_form.setDisabled('u_father_cell', true);
g_form.setDisabled('u_mother_cell', true);
}

```

5. **Save the Script:**

Click on **Submit** or **Update** to save the script.

Creating “Pincode Update” Client Script for Admission Table

Navigation:

Go to **All → Client Scripts** from the Application Navigator.

1. **Create a New Script:**

Click on **New**.

2. Fill in the Script Details:

- **Name:** Pincode Update
- **Table:** Admission
- **UI Type:** All
- **Type:** onChange
- **Field Name:** Pincode (u_pincode)
- **Active:** Checked
- **Isolate Script:** Checked

The screenshot shows the 'Client Script' configuration window for a script named 'Pincode Update'. The interface includes several input fields and checkboxes:

- Name:** Pincode Update
- Table:** Admission [u_admission]
- UI Type:** Desktop
- Type:** onChange
- Field name:** Pincode
- Application:** Global
- Active:** ☒
- Inherited:** ☐
- Global:** ☒

Below these fields are sections for 'Description' and 'Messages', both currently empty. At the bottom is a 'Script' editor with a toolbar and the following code:

```
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');  
10  }
```

3. Script Logic:

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

```
// Auto-fill mandal, city, and district based on pincode
```

```
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');
}
}
```

4. **Save the Script:**

Click on **Submit** or **Update** to save the script.

Creating “Disable Fields” Client Script for Student Progress Table

1. **Navigation:**

Go to **All → Client Scripts** from the Application Navigator.

2. **Create a New Script:**

Click on **New**.

3. **Fill in the Script Details:**

- **Name:** Disable Fields on Load
- **Table:** Student Progress
- **UI Type:** All

- **Type:** onLoad
- **Active:** Checked
- **Isolate Script:** Checked

The screenshot shows the 'Client Script' configuration interface. At the top, there's a header bar with a back arrow, a title 'Client Script Disable Fields', and buttons for 'Update' and 'Delete'. Below this is a blue notification bar stating 'This form has annotations - click ⓘ to toggle them - (click here to never show this again)'. The main form is divided into two columns. The left column contains: 'Name' (Disable Fields), 'Table' (Student Progress [u_student_progress]), 'UI Type' (All), and 'Type' (onLoad). The right column contains: 'Application' (Global), 'Active' (checked), 'Inherited' (unchecked), and 'Global' (checked). Below these are fields for 'Description' and 'Messages'. At the bottom is a 'Script' editor with a toolbar and a code area containing the following JavaScript code:

```

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 }

```

4. **Script Logic:**

```

function onLoad() {
// Disable calculated fields to prevent manual entry
g_form.setDisabled('u_total', true);
g_form.setDisabled('u_percentage', true);
g_form.setDisabled('u_result', true);
}

```

5. **Save the Script:**

Click **Submit** or **Update** to save the script.

Creating “Total Update” Client Script for Student Progress Table

1. **Navigation:**

Go to **All → Client Scripts** from the Application Navigator.

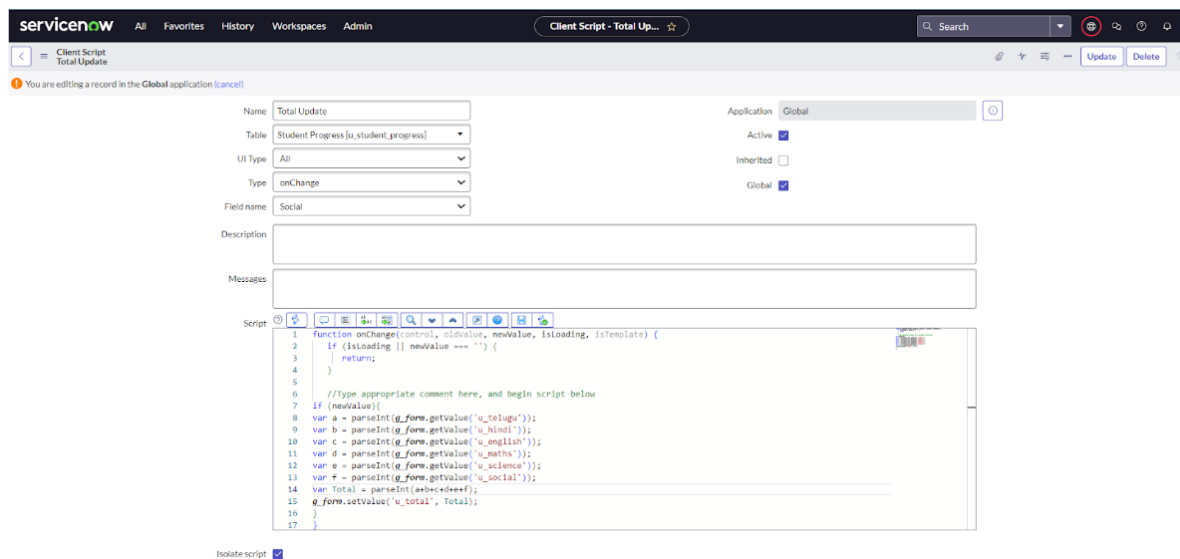
2. **Create a New Script:**

Click on **New**.

3. **Fill in the Script Details:**

- **Name:** Total Marks Calculation

- **Table:** Student Progress
- **UI Type:** All
- **Type:** onChange
- **Field Name:** Any one subject field (e.g., u_telugu) — repeat for each subject as needed
- **Active:** Checked
- **Isolate Script:** Checked



4. Script Logic:

```

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

```

```

// Calculate total marks from all subjects

```

```

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 = a + b + c + d + e + f;  
g_form.setValue('u_total', Total);  
}  
}
```

5. **Save the Script:**

Click on **Submit** or **Update** to save it.

Creating “Result” Client Script for Student Progress Table

1. **Navigation:**

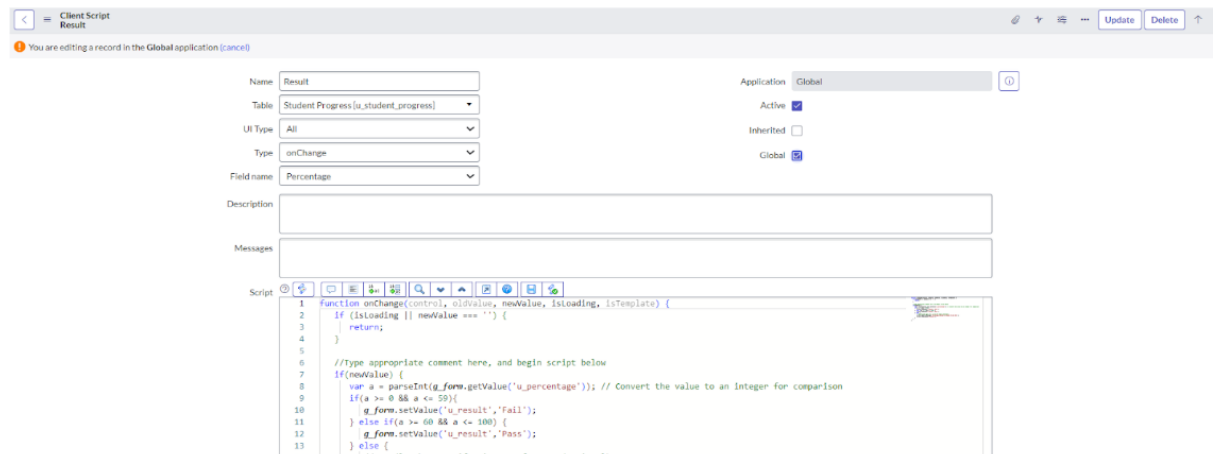
Go to **All → Client Scripts** from the Application Navigator.

2. **Create a New Script:**

Click on **New**.

3. **Fill in the Script Details:**

- **Name:** Result Based on Percentage
- **Table:** Student Progress
- **UI Type:** All
- **Type:** onChange
- **Field Name:** Percentage (u_percentage)
- **Active:** Checked
- **Isolate Script:** Checked



4. Script Logic:

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

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

```
    return;
```

```
}
```

```
// Set result based on percentage value
```

```
if (newValue) {
```

```
    var a = parseInt(g_form.getValue('u_percentage')); // Convert percentage to
    number
```

```
if (a >= 0 && a <= 59) {
```

```
    g_form.setValue('u_result', 'Fail');
```

```
} else if (a >= 60 && a <= 100) {
```

```
    g_form.setValue('u_result', 'Pass');
```

```
} else {
```

```
    // Invalid percentage range
```

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

```
    g_form.clearValue('u_result');
```

```
}  
  
}  
  
}
```

5. Save the Script:

Click **Submit** or **Update** to save it.

Creating “Percentage” Client Script for Student Progress Table

1. Navigation:

Go to **All → Client Scripts** from the Application Navigator.

2. Create a New Script:

Click on **New**.

3. Fill in the Script Details:

- **Name:** Percentage Calculation
- **Table:** Student Progress
- **UI Type:** All
- **Type:** onChange
- **Field Name:** Total (u_total)
- **Active:** Checked
- **Isolate Script:** Checked

The screenshot shows the Oracle APEX Client Script editor interface. At the top, there's a breadcrumb 'Client Script > Percentage' and buttons for 'Update' and 'Delete'. A warning message states: 'You are editing a record in the Global application (cancel)'. The 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.

4. **Script Logic:**

```
function onChange(control, oldValue, newValue, isLoading, isTemplate) {  
    if (isLoading || newValue === '') {  
        return;  
    }  
  
    // Calculate percentage based on total marks  
    var Total = g_form.getValue('u_total');  
    var Percentage = (Total / 600) * 100;  
    g_form.setValue('u_percentage', Percentage + '%');  
}
```

5. **Save the Script:**

Click on **Submit** or **Update** to save the script.

10. **Results**

The implemented Educational Management System on ServiceNow provides:

- ⑩ Centralized management of student and admission data.
- ⑩ Automated workflows for consistent and error-free operations.
- ⑩ Dynamic forms and scripts that enhance data entry and validation.
- ⑩ Seamless tracking of student progress and admission stages.

Screenshots should be included for:

- ⑩ Table and form configurations
- ⑩ Process flow
- ⑩ Script execution in forms

SALESFORCE RECORD

servicenow

AllFavoritesHistoryAdmin

Salesforce - Create SAL0001008

Search

<

≡

Salesforce

New record

Submit

Admin Number

SAL0001008

Father Name

Admin Date

Mother Name

Grade

-- None --

Mother Cell

Student Name

Father Cell

Submit

ADMISSION RECORD

Admission

New record

<

≡

Admission

New record

Submit

Admission Number

Admin Date

Purpose of join

-- None --

Grade

-- None --

Student Name

Fee

\$

0.00

Father Name

Father Cell

Mother Name

Mother Cell

Admin Status

-- None --

Comments

School Details

Address

School Area

Near Market

School

-- None --

Submit

PROGRESS RECORD

servicenow

All

Favorites

History

Admin

Progress - Create Created ☆

Search

<

New Section

New record

Submit

Admission Number

Admission Number.Grade

-- None --

Admission Number.Student Name

Admission Number.Father Name

Admission Number.Mother Name

Admission Number.Father Cell

Admission Number.Mother Cell

Student Progress

Telugu

Hindi

English

Maths

Science

Social

Total

Percentage

Result

Submit