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

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

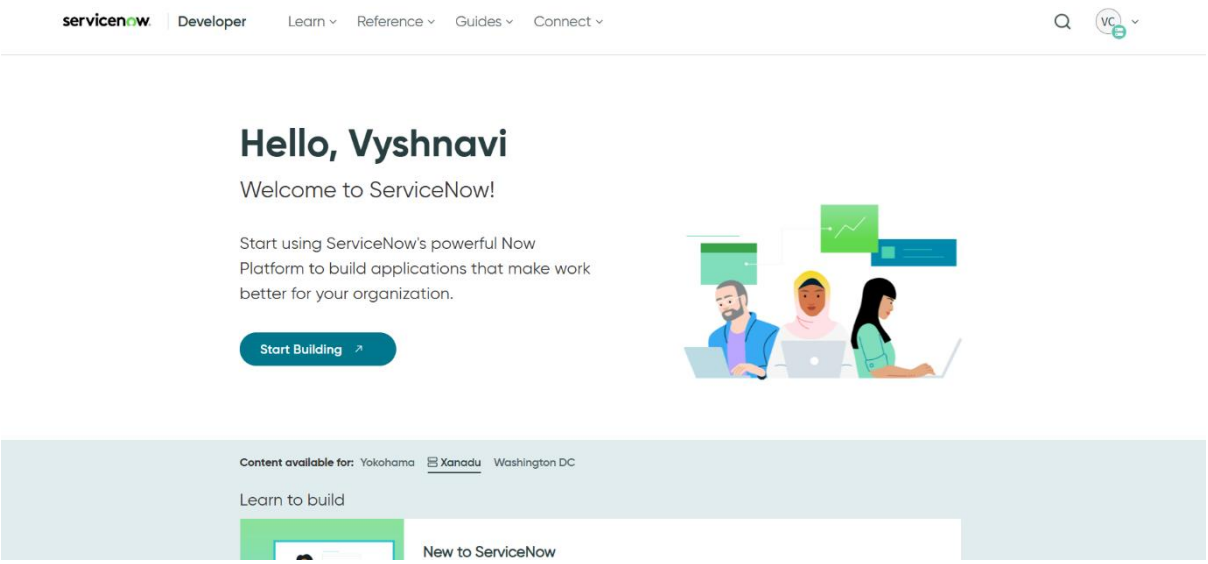
Request a Personal Developer Instance

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

Access Your Instance

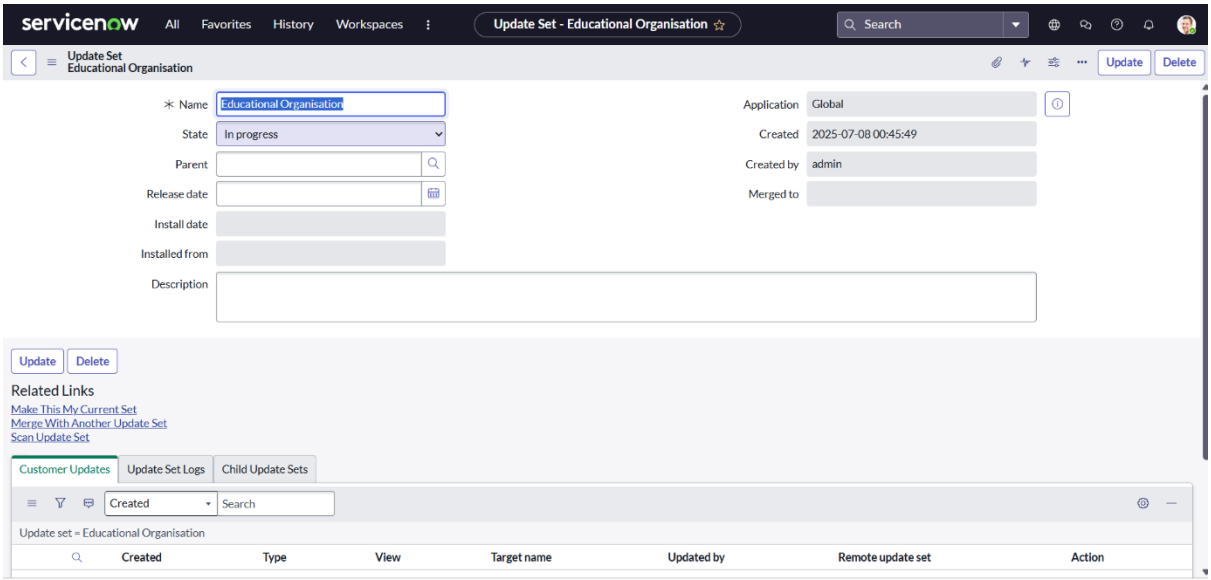
- Open the instance URL received via email.

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.

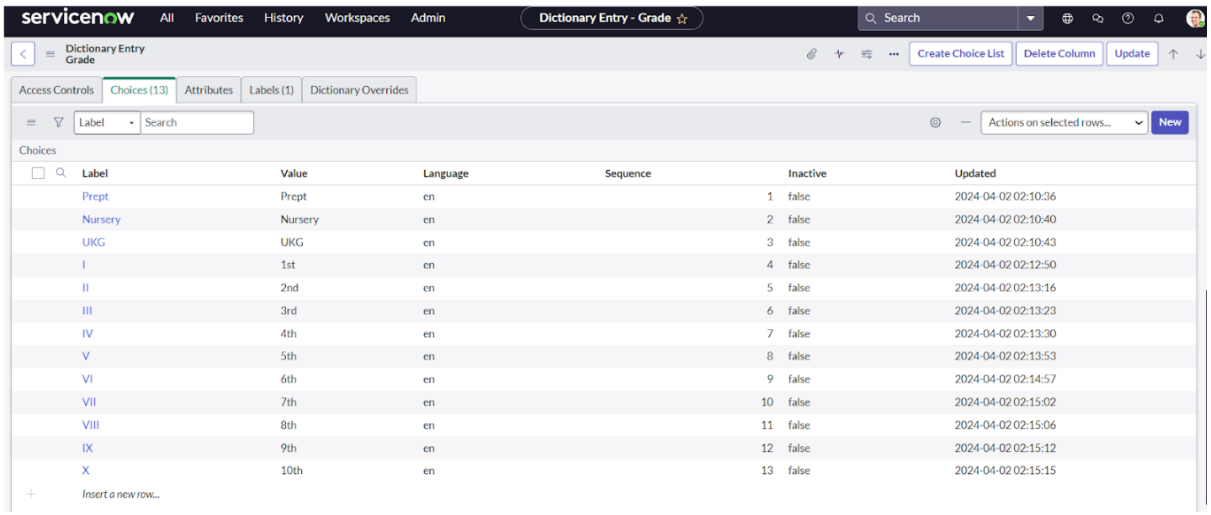


4. Creating Tables

Creating Salesforce Table

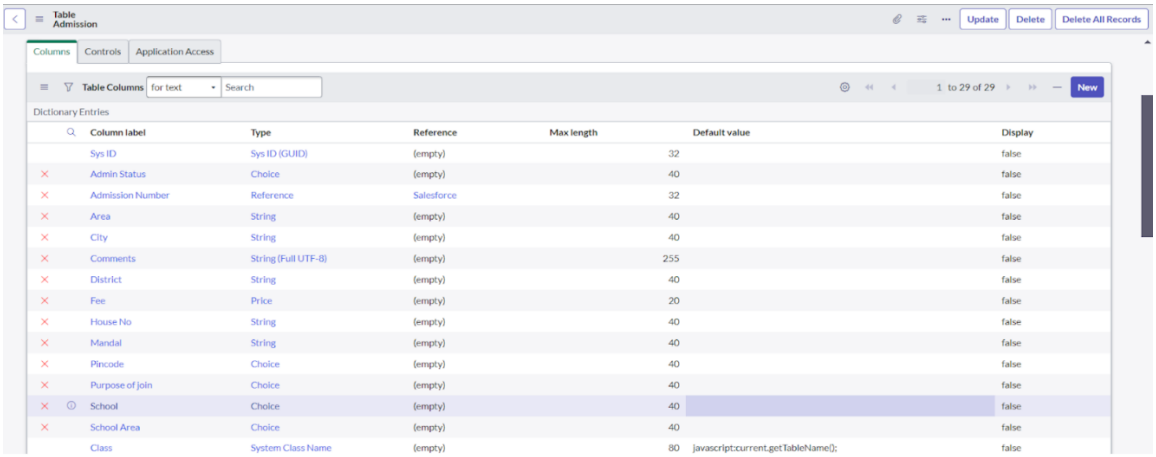
1. Navigate to All ➔ Tables.
2. Click on New to create a new table.
3. Enter the label: Salesforce. The API name will be generated automatically.
4. Add the required columns:
    - Admin Number (Display: True)
    - Grade (with choice values and sequence)
    - Other necessary fields as per requirement

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.



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



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.

×	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...*				

5. Form Layout

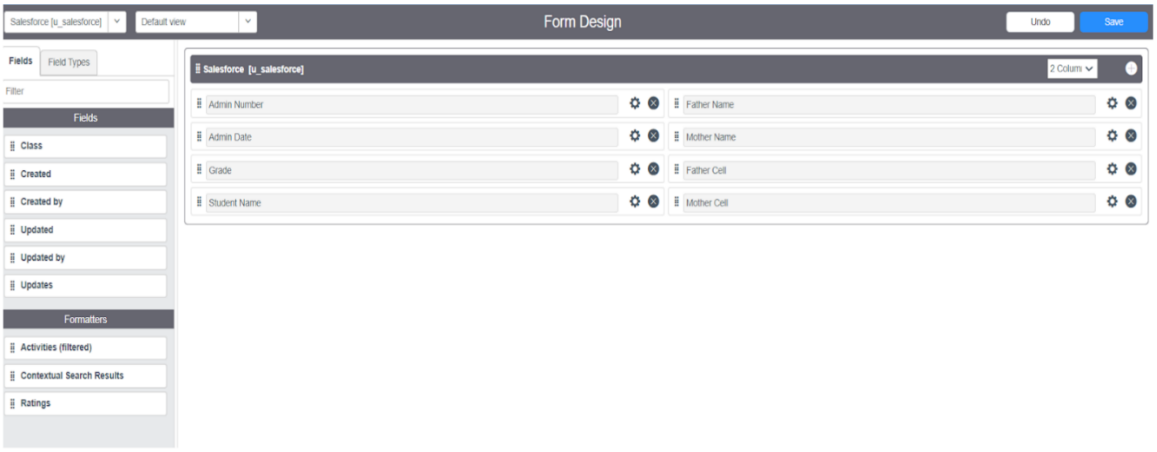
Configuring Table Form for Student Progress Table

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

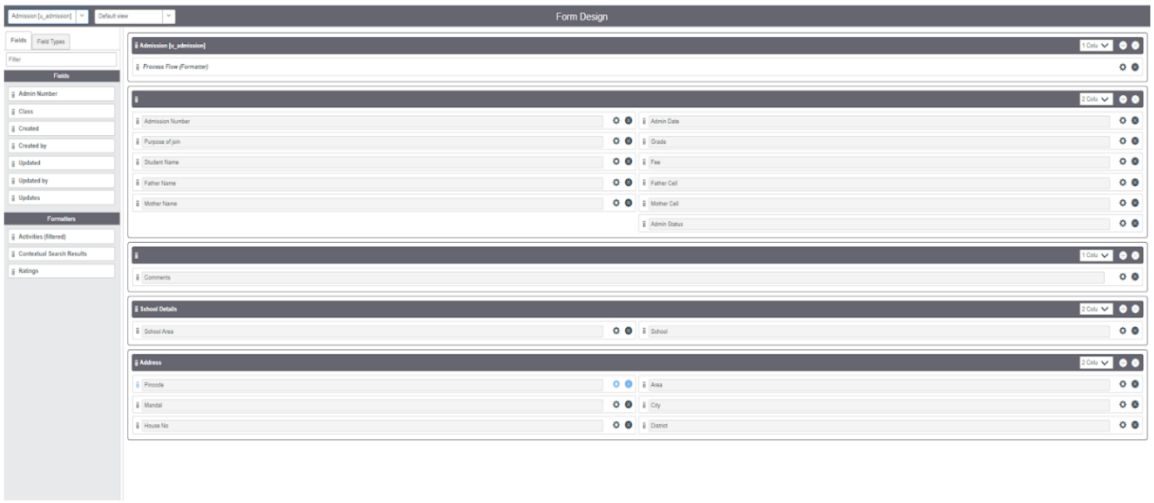
1. **Navigation**:
   Go to All → System Definition → Tables.
2. **Seach and Open Table**:
   - In the Label field, search for Salesforce.
   - Click to open the Salesforce Table (u_salesforce).
3. **Open Form Designer**:
   - Right-click on the header toggle bar of the record.
   - Select Configure → 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.



Creating Form Design for Admission Table

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

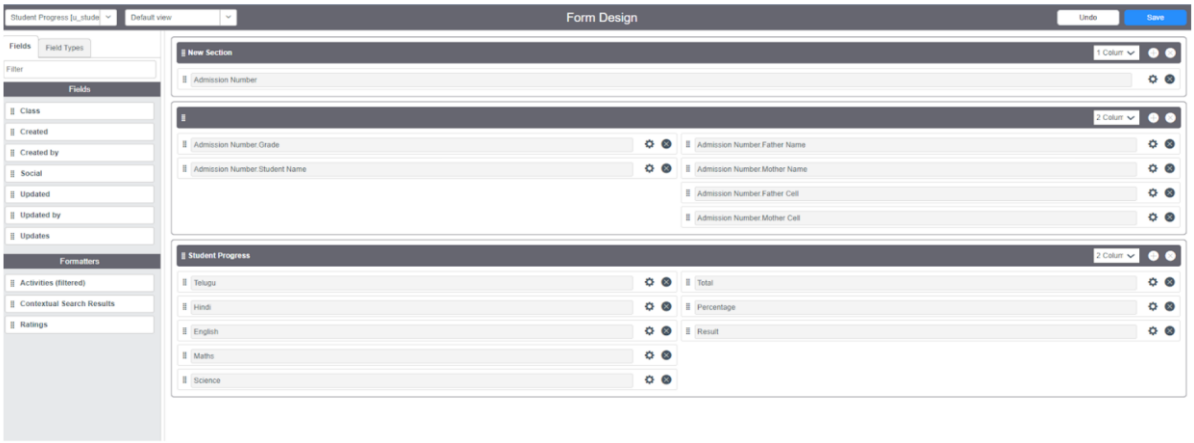
- Search Table: Search for Admission in the System Definition → Tables.
- Form Fields: Drag fields relevant to admissions such as:
    - Student Name
    - Father Name
    - Mother Name
    - Grade
    - Admin Status
    - School & School Area
    - Purpose of Join
    - Pincode and Location
- Click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



Creating Form Design for Student Progress Table

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

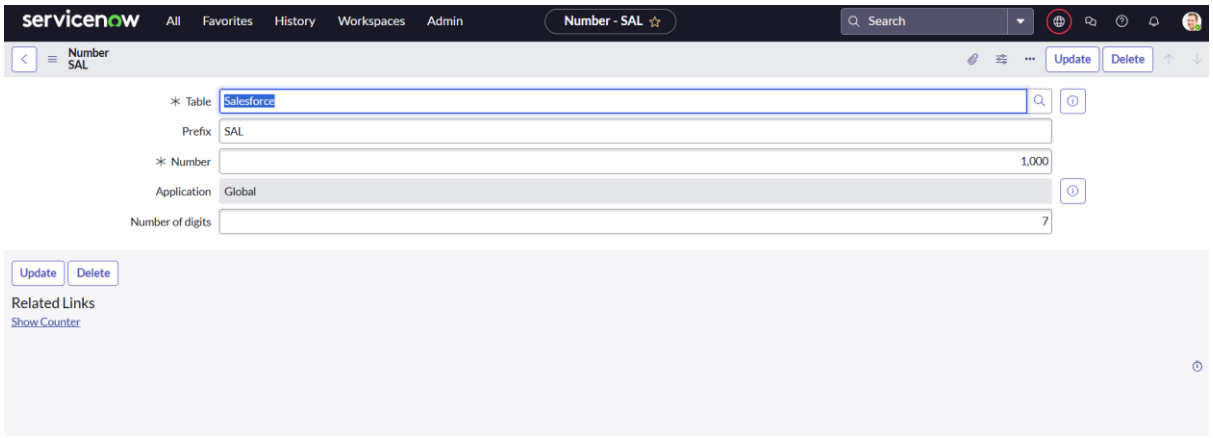
- Search Table: Search for Student Progress.
- Form Fields: Include:
    - Admission Number (reference field)
    - Subject Marks (Telugu, Hindi, English, Maths, Science, Social)
    - Total, Percentage, and Result
- Drag and drop fields in logical sequence.
- Click Save to finalize.



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.



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**:

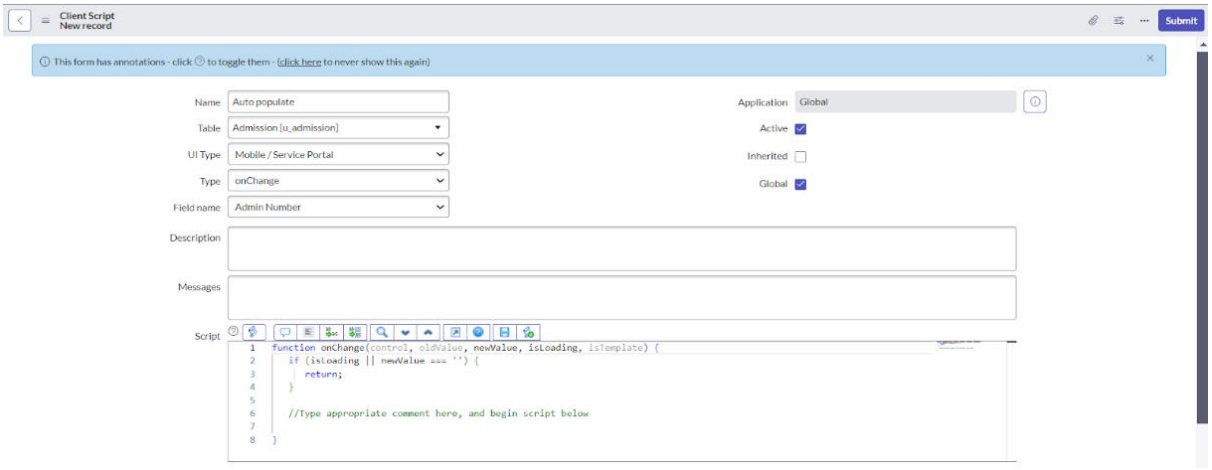
   Go to **All → Client Scripts** in the Application Navigator.

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



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**:

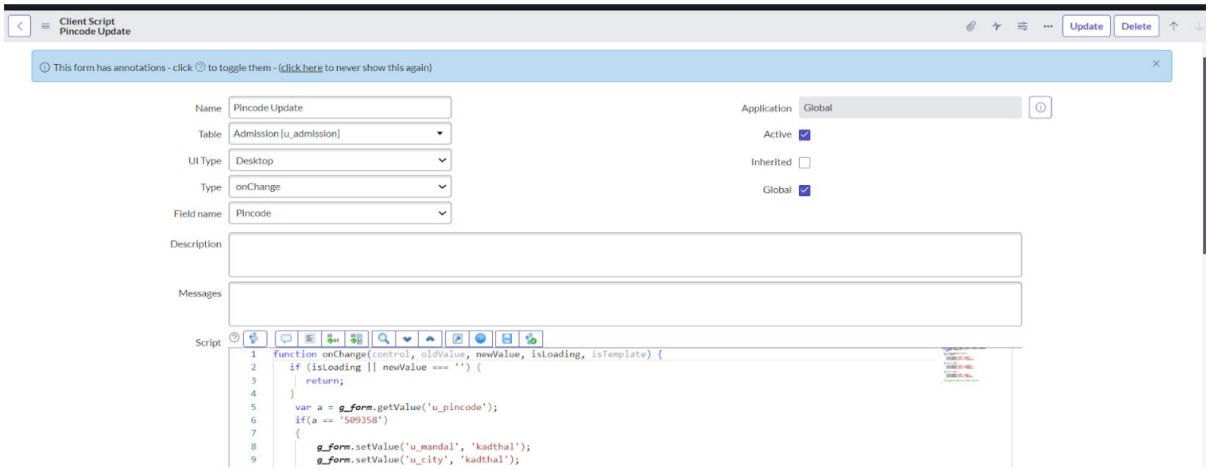
   Click on **Submit** or **Update** to save the script.

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



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');

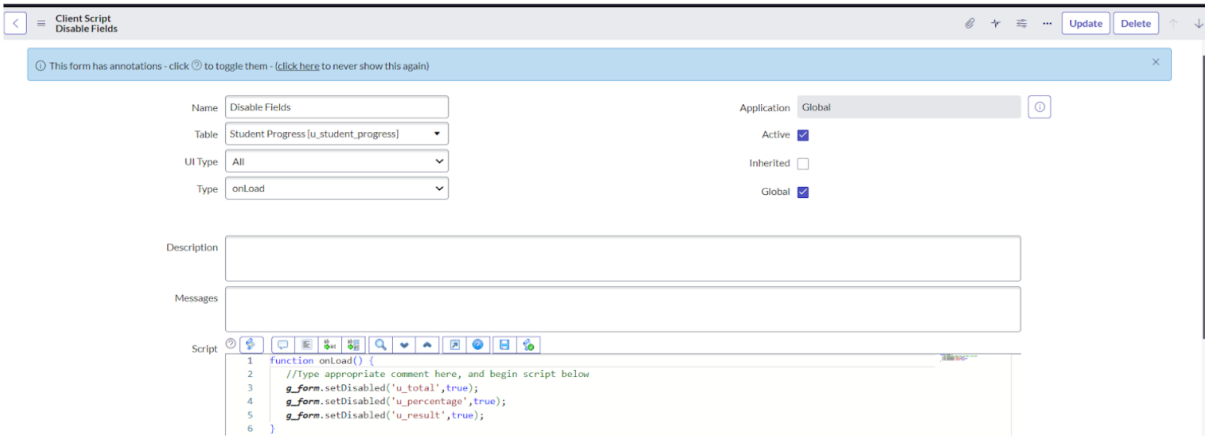
```
  g_form.setValue('u_city', 'kadthal');
  g_form.setValue('u_district', 'RangaReddy');
 }
 else if (a == '500081') {
  g_form.setValue('u_mandal', 'karmanghat');
  g_form.setValue('u_city', 'karmanghat');
  g_form.setValue('u_district', 'RangaReddy');
 }
 else if (a == '500079') {
  g_form.setValue('u_mandal', '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



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);

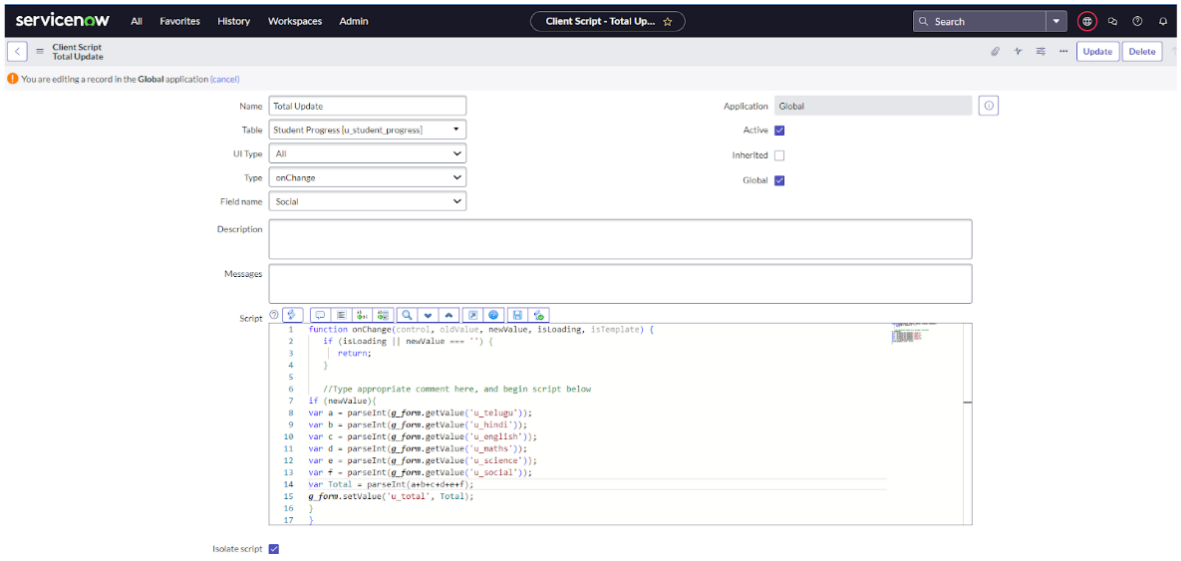
}

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'));
```

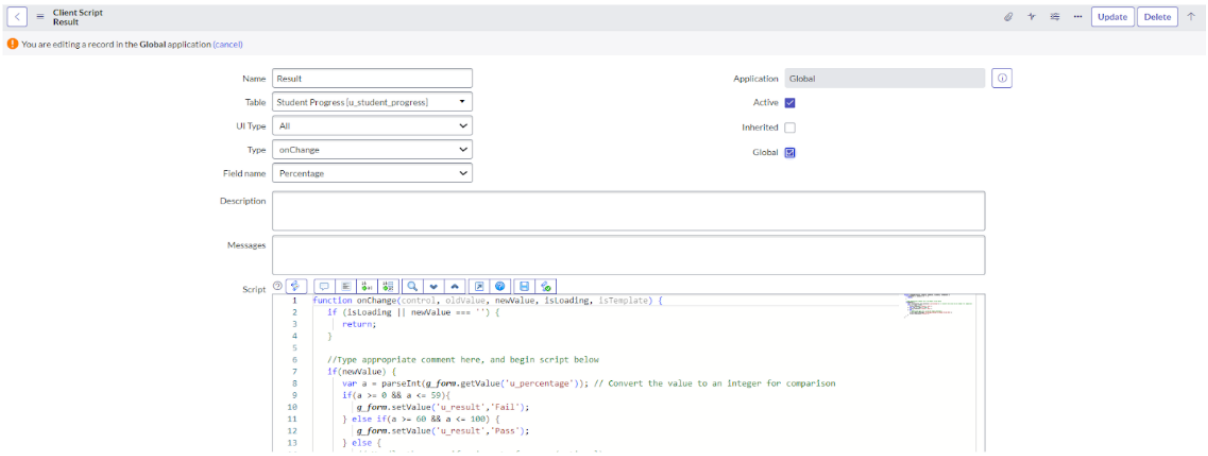
```
    var d = parseInt(g_form.getValue('u_maths'));
    var e = parseInt(g_form.getValue('u_science'));
    var f = parseInt(g_form.getValue('u_social'));

    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

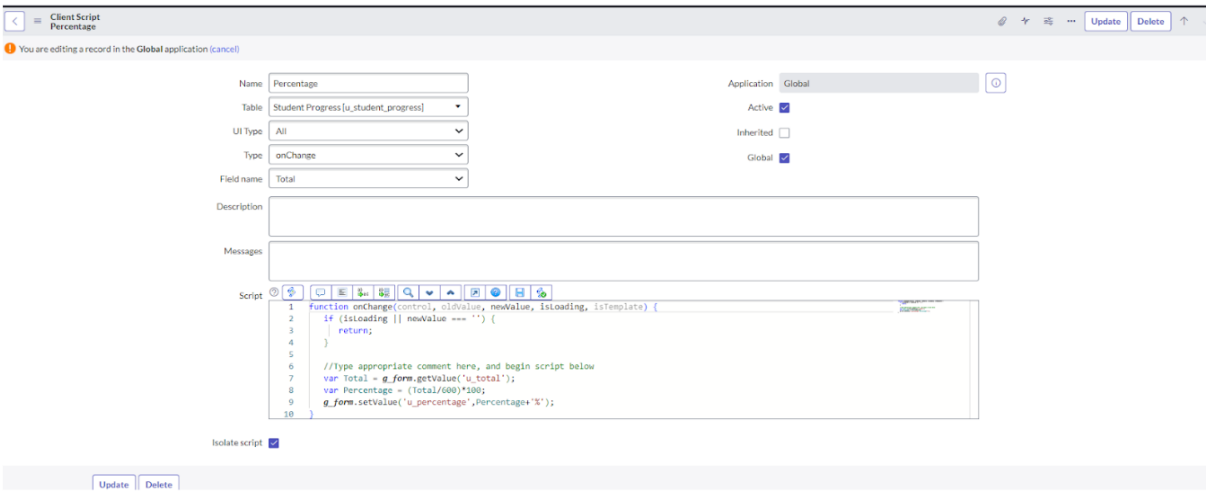
1. **Navigation**:

   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



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

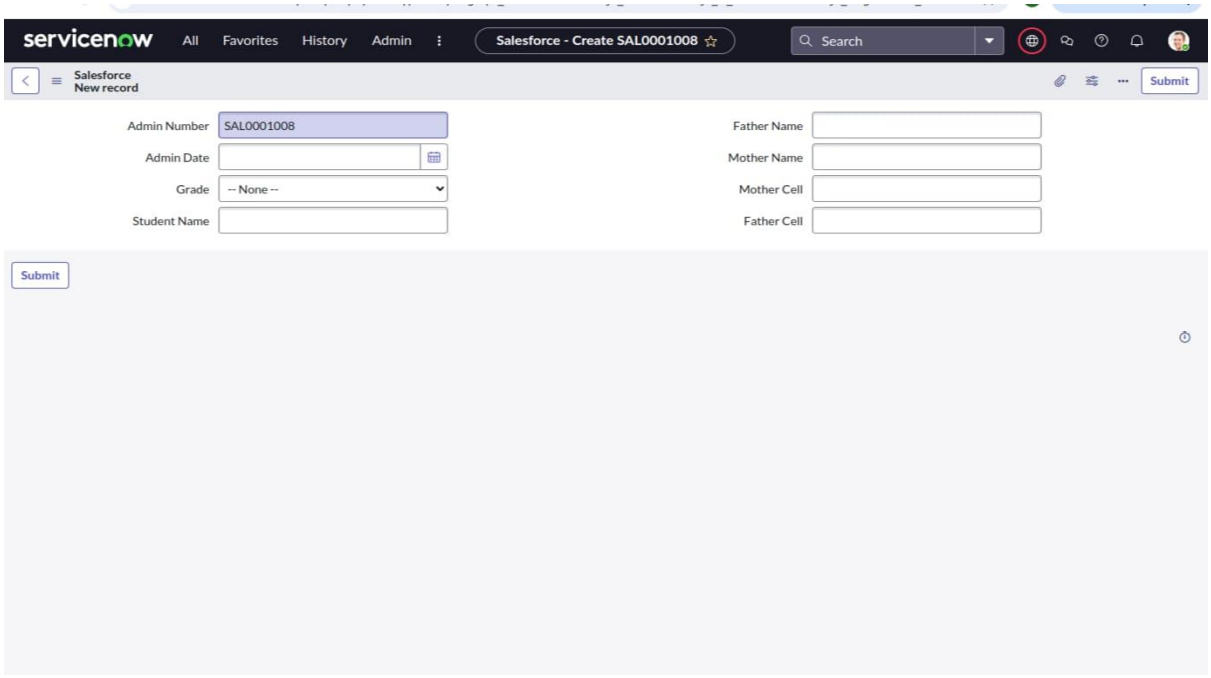
The implemented Educational Management System on ServiceNow provides:

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

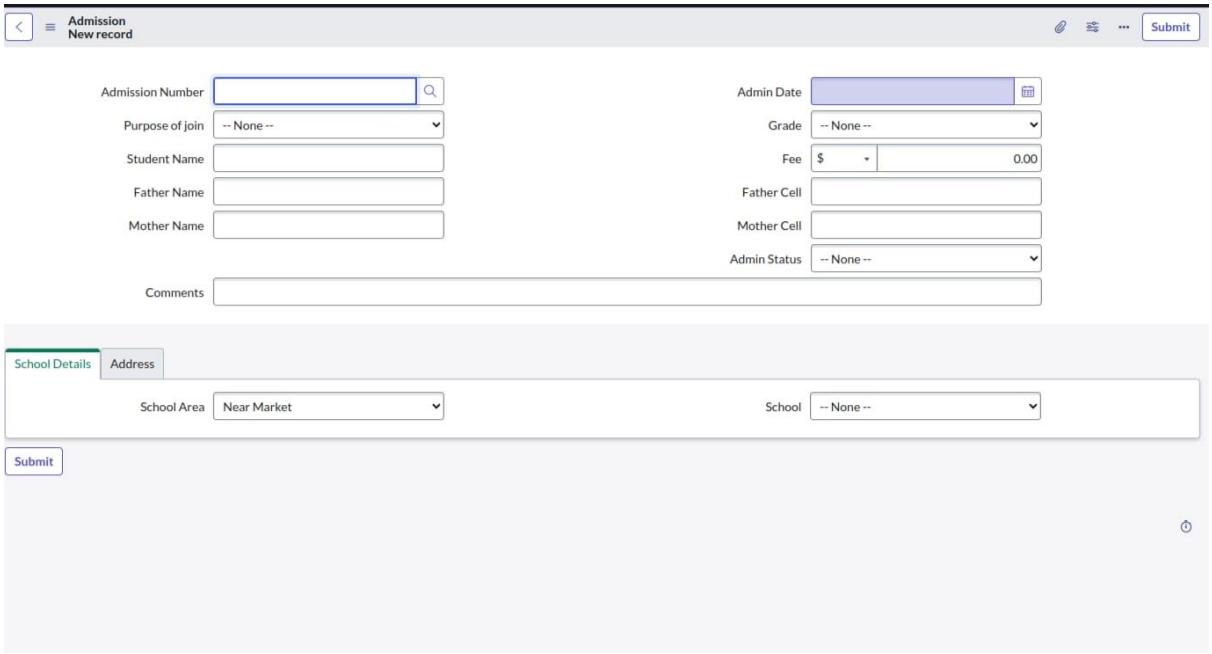
Screenshots should be included for:

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

SALESFORCE RECORD



ADMISSION RECORD



PROGRESS RECORD