Educational Organisation Management System Using ServiceNow

1. Introduction

This report presents a comprehensive overview of the development of an Educational Organisation Management System using the ServiceNow platform. The goal of the project is to design and implement a solution that helps institutions manage their student records, admissions, and administrative processes efficiently and with minimal manual intervention. By leveraging the power of ServiceNow's low-code development environment, we have built a fully functional application with automated workflows, structured data tables, custom UI components, and interactive forms.

Project Objectives

This section discusses project objectives in detail. It includes definitions, examples, and step-by-step procedures as followed during the development of the Educational Organisation Management System. Each step was carefully planned and executed to ensure consistency and reliability across all modules. The implementation approach not only follows ServiceNow best practices but also aligns with user requirements and business needs. Documentation, testing, and iterative improvement were part of every stage of the project lifecycle.

ServiceNow Overview

This section discusses servicenow overview in detail. It includes definitions, examples, and step-by-step procedures as followed during the development of the Educational Organisation Management System. Each step was carefully planned and executed to ensure consistency and reliability across all modules. The implementation approach not only follows ServiceNow best practices but also aligns with user requirements and business needs. Documentation, testing, and iterative improvement were part of every stage of the project lifecycle.

Instance Setup and Navigation

This section discusses instance setup and navigation in detail. It includes definitions, examples, and step-by-step procedures as followed during the development of the Educational Organisation Management System. Each step was carefully planned and executed to ensure consistency and reliability across all modules. The implementation approach not only follows ServiceNow best practices but also aligns with user requirements and business needs. Documentation, testing, and iterative improvement were part of every stage of the project lifecycle.

o begin working with ServiceNow, a Personal Developer Instance (PDI) must be set up from the ServiceNow Developer Portal. The following steps were followed to configure the development environment:

Step 1: Register for a Developer Account

Visit the official website: https://developer.servicenow.com

Sign up or log in with your credentials.

Accept the Developer Agreement.

Step 2: **Request a Personal Developer Instance**

After logging in, go to the "Manage" tab.

Click Request Instance.

Select the latest ServiceNow release (e.g., Washington DC, Utah).

Choose the "Start Instance" button.

Wait for the instance URL and credentials (e.g., https://dev12345.service-now.com).

Step 3: Access and Log Into the Instance

Open the instance URL in your browser.

Use the provided username and password to log in (usually admin / auto-generated password).

Change your password upon first login.

Step 4: Navigate the ServiceNow Interface

Use the Application Navigator (left sidebar) to access:

Studio

Tables

Update Sets

Form Designer

Flow DesignerUse the filter box to quickly search and access modules (e.g., type "Tables" to access System Definition > Tables).

Step 5: Enable Developer Options

Go to System Settings (gear icon top-right).

Enable Show Application Picker, Form Design, and Update Set Picker for better control.

Creating Update Sets

This section discusses creating update sets in detail. It includes definitions, examples, and step-by-step procedures as followed during the development of the Educational Organisation Management System. Each step was carefully planned and executed to ensure consistency and reliability across all modules. The implementation approach not only follows ServiceNow best practices but also aligns with user requirements and business needs. Documentation, testing, and iterative improvement were part of every stage of the project lifecycle.

Step 1: Navigate to Update Sets

Go to the Application Navigator

Type: Local Update Sets

Path: System Update Sets > Local Update Sets

Step 2: Create a New Update Set

Click on New

Fill in the following details:

Name: Educational Organisation Update Set

Application: Choose your custom application (e.g., Educational Organisation)

State: Set as In Progress

Description: "Tracks all configuration changes for student and admission management system"

Step 3: Mark Update Set as Current

Open the newly created update set

Click Make This My Current Set (top-right)

This ensures all your changes are recorded under this update set

Step 4: Perform Your Development Tasks

All changes made now (e.g., creating tables, forms, scripts) will be captured automatically in this update set.

Step 5: Review and Close the Update Set

Once development is complete:

Change State to Complete

Click Export to XML to download the update set

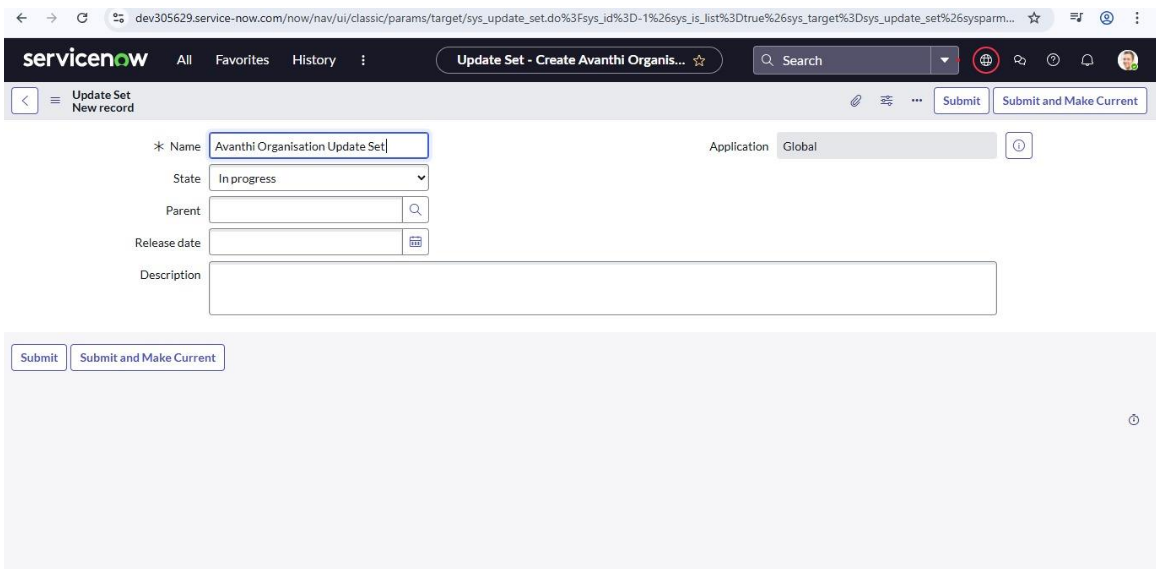
Step 6: Import to Another Instance

On another instance, go to:

System Update Sets > Retrieved Update Sets

Upload the XML file

Preview and Commit it


dev305629.service-now.com/now/nav/ui/classic/params/target/sys_update_set.do%3Fsys_id%3D-1%26sys_is_list%3Dtrue%26sys_target%3Dsys_update_set%26sysparm...
servicenow
All
Favorites
History
Update Set - Create Avanthi Organis...
Search
Update Set
New record
Submit
Submit and Make Current

Name	Avanthi Organisation Update Set
State	In progress
Parent	
Release date	
Description	
Application	Global

Submit
Submit and Make Current

Table Creation and Custom Fields

This section discusses table creation and custom fields in detail. It includes definitions, examples, and step-by-step procedures as followed during the development of the Educational Organisation Management System. Each step was carefully planned and executed to ensure consistency and reliability across all modules. The implementation approach not only follows ServiceNow best practices but also aligns with user requirements and business needs. Documentation, testing, and iterative improvement were part of every stage of the project lifecycle.

Steps to Create a Table and Add Fields:

Step 1: Navigate to Table Creation

Go to Studio from the Application Navigator.

Select your custom application (e.g., Educational Organisation).

Click on Create Application File > Data Model > Table.

Step 2: Define Table Properties

Label: e.g., Student Progress

Name: auto-generated as u_student_progress

Check the boxes to:

Automatically create a module

Add application scope prefix (u_)

Create access controls

Step 3: Add Custom Fields

Click Add Field.

Define:

Field Label (e.g., "Student Name")

Field Type (e.g., String, Date, Choice, Reference)

Add optional Default Value, Choices, or Reference Table as needed.

Step 4: Save and Verify

Save the table.

Navigate to the newly created module under the application to view and add sample records.

Tables Created:

1.Avanthi Education (u_avanthi_education):

Example: Student Name

Admission Number

Admin Date etc.

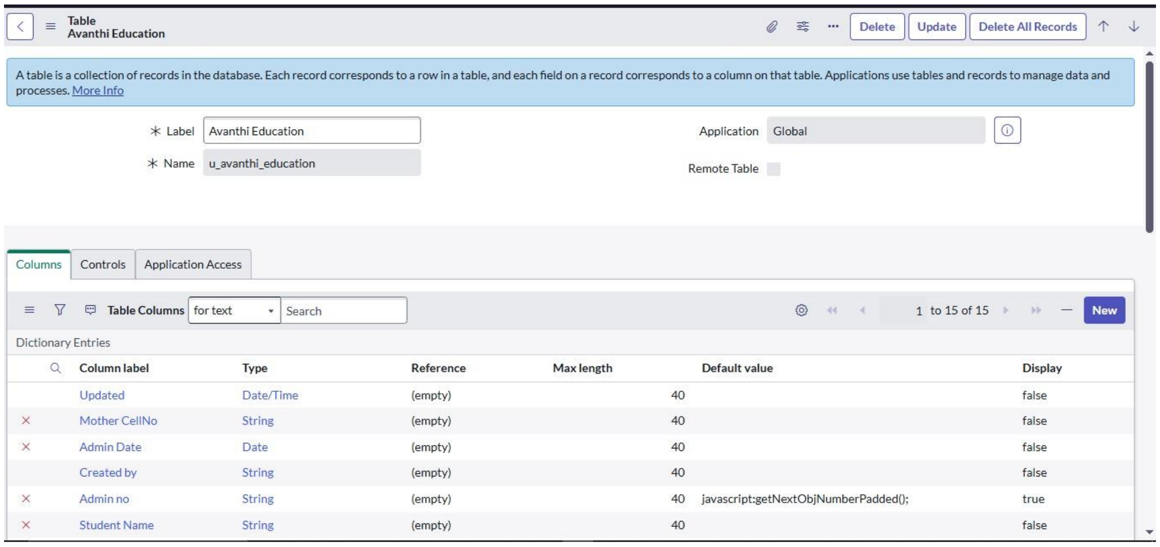


Table
Avanthi Education

A table is a collection of records in the database. Each record corresponds to a row in a table, and each field on a record corresponds to a column on that table. Applications use tables and records to manage data and processes. More Info

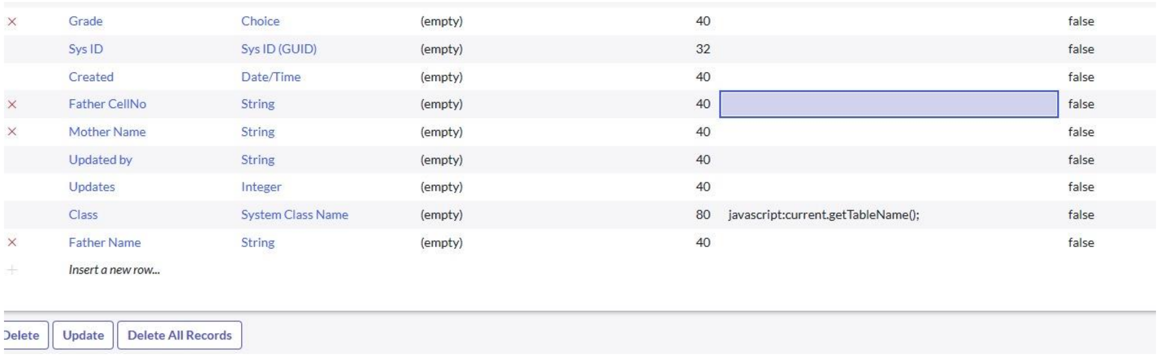
* Label	Avanthi Education	Application	Global
* Name	u_avanthi_education	Remote Table	

Columns | Controls | Application Access

Table Columns for text Search 1 to 15 of 15 New

Dictionary Entries

Column label	Type	Reference	Max length	Default value	Display
Updated	Date/Time	(empty)	40		false
Mother CellNo	String	(empty)	40		false
Admin Date	Date	(empty)	40		false
Created by	String	(empty)	40		false
Admin no	String	(empty)	40	javascript:getNextObjNumberPadded();	true
Student Name	String	(empty)	40		false



Column label	Type	Reference	Max length	Default value	Display
Grade	Choice	(empty)	40		false
Sys ID	Sys ID (GUID)	(empty)	32		false
Created	Date/Time	(empty)	40		false
Father CellNo	String	(empty)	40		false
Mother Name	String	(empty)	40		false
Updated by	String	(empty)	40		false
Updates	Integer	(empty)	40		false
Class	System Class Name	(empty)	80	javascript:current.getTableName();	false
Father Name	String	(empty)	40		false

Insert a new row...

2.Admission (u_admission):

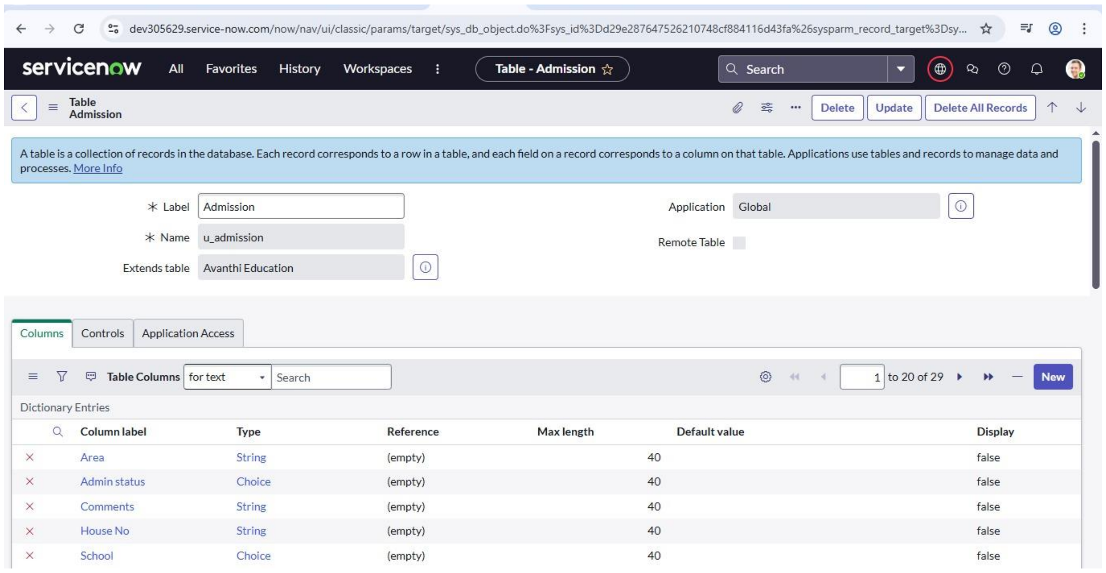

dev305629.service-now.com/now/nav/ui/classic/params/target/sys_db_object.do%3Fsys_id%3Dd29e287647526210748cf884116d43fa%26sysparm_record_target%3Dsy...

Table - Admission

Table
Admission

A table is a collection of records in the database. Each record corresponds to a row in a table, and each field on a record corresponds to a column on that table. Applications use tables and records to manage data and processes. More Info

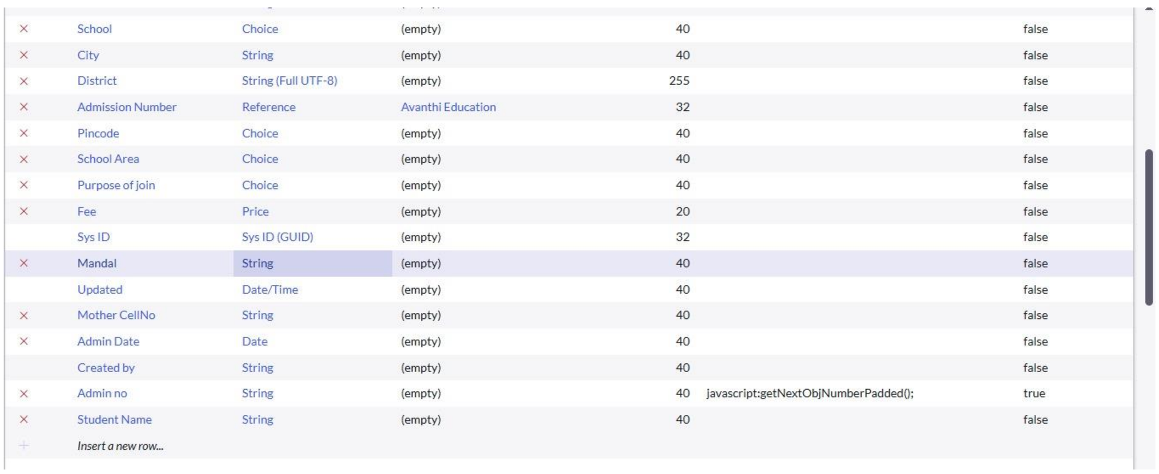
* Label	Admission	Application	Global
* Name	u_admission	Remote Table	
Extends table	Avanthi Education		

Columns | Controls | Application Access

Table Columns for text Search 1 to 20 of 29 New

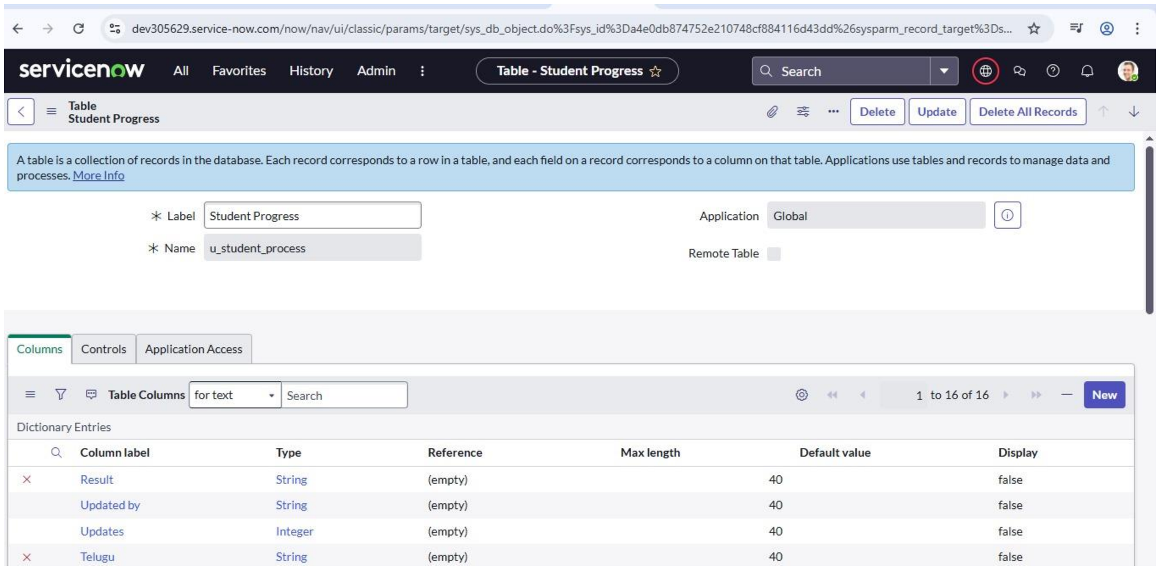
Dictionary Entries

Column label	Type	Reference	Max length	Default value	Display
Area	String	(empty)	40		false
Admin status	Choice	(empty)	40		false
Comments	String	(empty)	40		false
House No	String	(empty)	40		false
School	Choice	(empty)	40		false



×	School	Choice	(empty)	40		false
×	City	String	(empty)	40		false
×	District	String (Full UTF-8)	(empty)	255		false
×	Admission Number	Reference	Avanthi Education	32		false
×	Pincode	Choice	(empty)	40		false
×	School Area	Choice	(empty)	40		false
×	Purpose of join	Choice	(empty)	40		false
×	Fee	Price	(empty)	20		false
	Sys ID	Sys ID (GUID)	(empty)	32		false
×	Mandal	String	(empty)	40		false
	Updated	Date/Time	(empty)	40		false
×	Mother CellNo	String	(empty)	40		false
×	Admin Date	Date	(empty)	40		false
	Created by	String	(empty)	40		false
×	Admin no	String	(empty)	40	javascript:getNextObjNumberPadded();	true
×	Student Name	String	(empty)	40		false
+	*Insert a new row...*					

3.Student Progress (u_student_progress):



dev305629.service-now.com/now/nav/ui/classic/params/target/sys_db_object.do%3Fsys_id%3Da4e0db874752e210748cf884116d43dd%26sysparm_record_target%3Ds...

Table - Student Progress

Table
Student Progress

A table is a collection of records in the database. Each record corresponds to a row in a table, and each field on a record corresponds to a column on that table. Applications use tables and records to manage data and processes. More Info

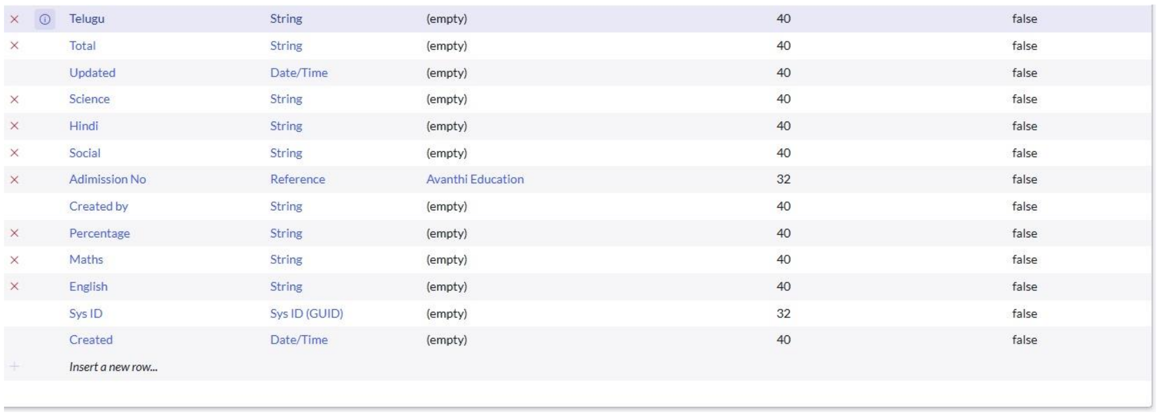
* Label	Student Progress
* Name	u_student_process
Application	Global
Remote Table	☐

Columns | Controls | Application Access

Table Columns — for text — Search — 1 to 16 of 16

Dictionary Entries

	Column label	Type	Reference	Max length	Default value	Display
×	Result	String	(empty)	40		false
	Updated by	String	(empty)	40		false
	Updates	Integer	(empty)	40		false
×	Telugu	String	(empty)	40		false



×	Telugu	String	(empty)	40		false
×	Total	String	(empty)	40		false
	Updated	Date/Time	(empty)	40		false
×	Science	String	(empty)	40		false
×	Hindi	String	(empty)	40		false
×	Social	String	(empty)	40		false
×	Adimission No	Reference	Avanthi Education	32		false
	Created by	String	(empty)	40		false
×	Percentage	String	(empty)	40		false
×	Maths	String	(empty)	40		false
×	English	String	(empty)	40		false
	Sys ID	Sys ID (GUID)	(empty)	32		false
	Created	Date/Time	(empty)	40		false
+	*Insert a new row...*					

Form Design and Layout Customization

This section discusses form design and layout customization in detail. It includes definitions, examples, and step-by-step procedures as followed during the development of the Educational Organisation Management System. Each step was carefully planned and executed to ensure consistency and reliability across all modules. The implementation approach not only follows ServiceNow best practices but also aligns with user requirements and business needs. Documentation, testing, and iterative improvement were part of every stage of the project lifecycle.

Purpose of Form Design:

Enhance user interface and readability

Group related fields using sections or tabs

Simplify data entry and avoid clutter

Improve workflow visibility (e.g., using Flow Formatter)

Steps for Customizing Form Layout:

Step 1: Navigate to Form Layout

Go to Application Navigator

Search for and open the desired table (e.g., Student)

Open any record, then right-click the header > Configure > Form Layout

Step 2: Rearranging Fields

Use the Form Layout interface to:

Move fields between "Selected" and "Available"

Change field order using drag-and-drop

Create sections for grouped fields (e.g., Student Details, Contact Info, Parent Details)

Step 3: Add New Sections

Click “Add Section” to insert a new logical group of fields

Name sections clearly (e.g., “School Details”, “Admission Info”)

Assign related fields under each section

Using Form Designer for Visual Customization:

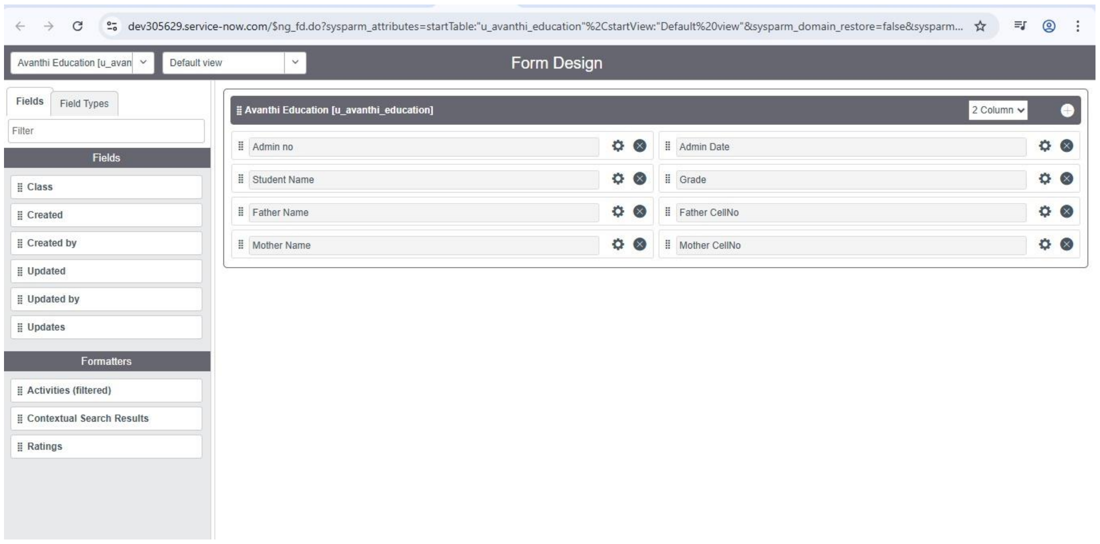
Step 1: Open Form Designer: Navigate to the table → Open any record → Right-click header → Configure > Form Design

Step 2: Arrange Fields Visually

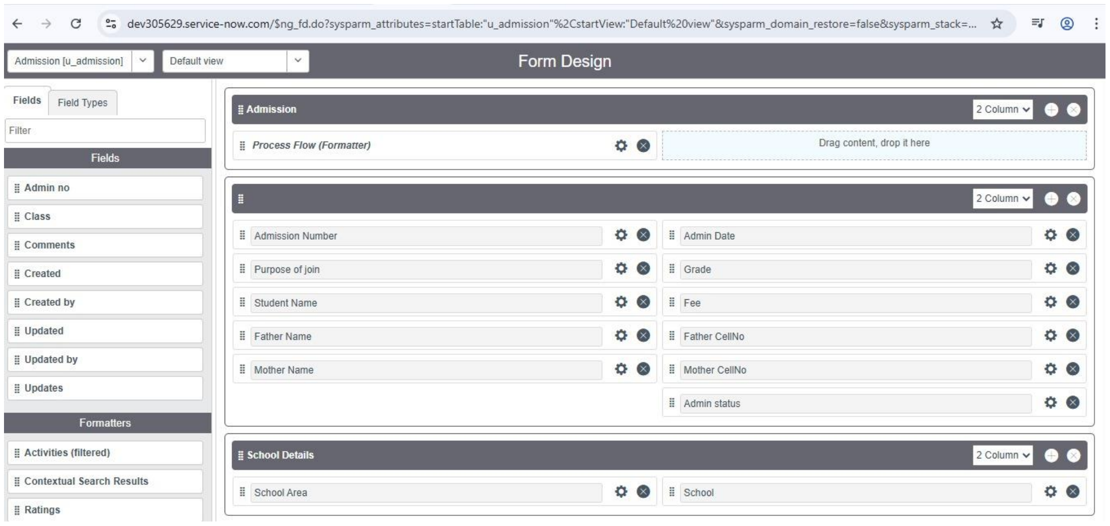
Drag and drop fields from the left panel onto the canvas,Create multiple columns, tabs, or split views for better structure,Drag sections and reorder them for improved flow

Step 3: Add Widgets (Optional)

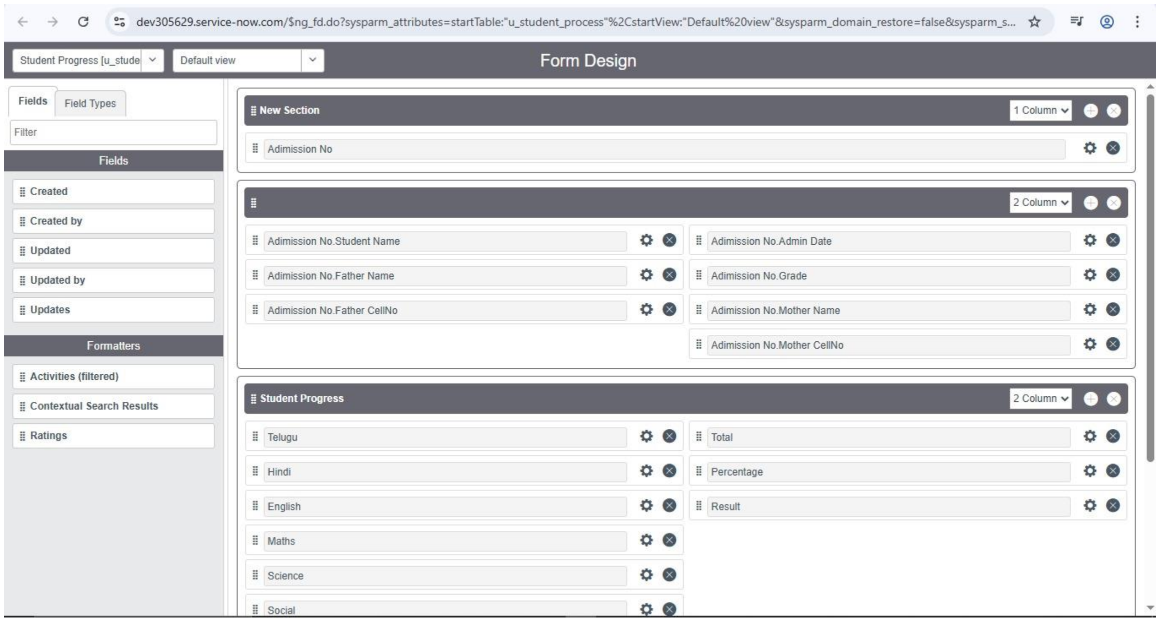
Example: Form Layout for Avanthi Education


dev305629.service-now.com/$ng_fd.do?sysparm_attributes=startTable:"u_avanthi_education"%2CstartView:"Default%20view"&sysparm_domain_restore=false&sysparm...
Avanthi Education [u_avan
Default view
Form Design
Fields
Field Types
Filter
Fields
Class
Created
Created by
Updated
Updated by
Updates
Formatters
Activities (filtered)
Contextual Search Results
Ratings
Avanthi Education [u_avanthi_education]
2 Column
Admin no
Admin Date
Student Name
Grade
Father Name
Father CellNo
Mother Name
Mother CellNo

Form layout for Admission:



Form layout for Student Progress:



dev305629.service-now.com/$ng_fd.do?sysparm_attributes=startTable:"u_student_process"%2CstartView:"Default%20view"&sysparm_domain_restore=false&sysparm_s...

Student Progress [u_stude | Default view | Form Design

Fields: Created, Created by, Updated, Updated by, Updates

Formatters: Activities (filtered), Contextual Search Results, Ratings

New Section (1 Column): Admission No

(2 Column): Admission No.Student Name, Admission No.Father Name, Admission No.Father CellNo | Admission No.Admin Date, Admission No.Grade, Admission No.Mother Name, Admission No.Mother CellNo

Student Progress (2 Column): Telugu, Hindi, English, Maths, Science, Social | Total, Percentage, Result

Admission Status Flow Using Flow Formatter

This section discusses admission status flow using flow formatter in detail. It includes definitions, examples, and step-by-step procedures as followed during the development of the Educational Organisation Management System. Each step was carefully planned and executed to ensure consistency and reliability across all modules. The implementation approach not only follows ServiceNow best practices but also aligns with user requirements and business needs. Documentation, testing, and iterative improvement were part of every stage of the project lifecycle.

Purpose of the Flow Formatter:

Visually show progress through stages like: New → In Progress → Joined

Improve form usability and user understanding

Reduce the chance of incorrect or skipped status updates

Support clear process visualization for administrators

Statuses Represented in the Flow:

The following choice values were configured in the Admission Status field (Choice field in u_admission table):

1. New – Application has been received

2. In Progress – Under review or document verification ongoing

3. Joined – Student has joined successfully

4. Rejoined – Re-admission for the same student

5. Rejected – Application not accepted

6. Cancelled – Admission process cancelled

7. Closed – Case closed (after joining or rejection)

These values were arranged in logical order and used in the flow formatter for tracking.

Steps to Add and Configure Flow Formatter:

Step 1: Add Flow Formatter to the Form

Navigate to Form Layout or Form Design for the Admission table

In the Form Layout, add the Flow Formatter field

In Form Design, drag and drop Flow Formatter into the top of the form

Step 2: Set up Flow Formatter Options:

Right-click the form header → Configure > Dictionary on the Flow Formatter field

Set the following:

Type: Flow Formatter

Formatter: Formatter_flow_formatter

Flow type field: Leave blank

Status field: Select the field Admission Status

Step 3: Save and Test

Save changes and return to the form

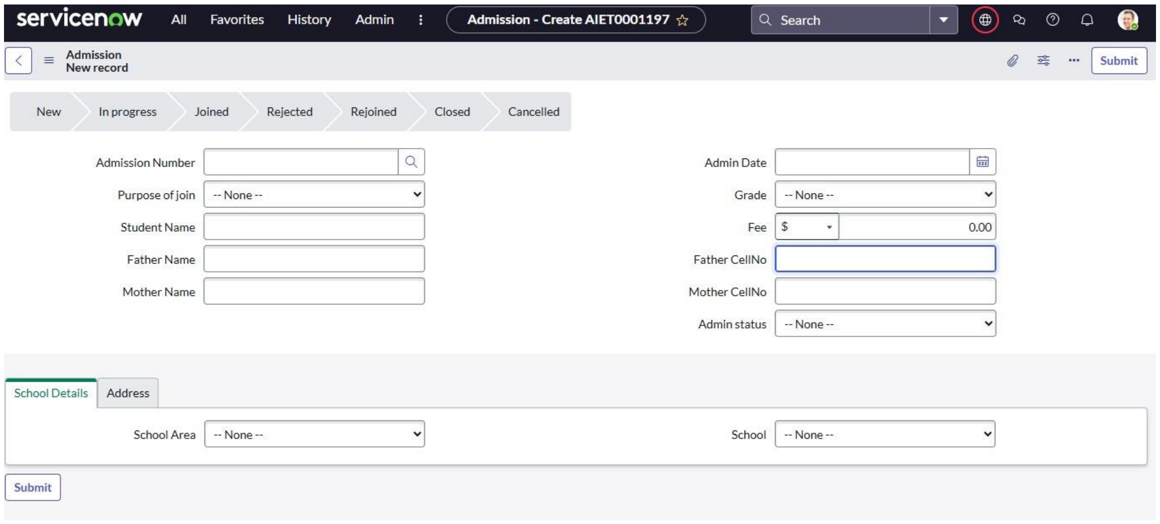
Add or edit a record to verify if the visual flow appears and updates as per status changes

Example Flow (Displayed at the Top of the Form):

New → In Progress → Joined → Closed

↓ ↓ ↓

Rejected Rejoined Cancelled



Admission Number		Admin Date	
Purpose of join	-- None --	Grade	-- None --
Student Name		Fee	$ 0.00
Father Name		Father CellNo	
Mother Name		Mother CellNo	
		Admin status	-- None --
School Area	-- None --	School	-- None --

Using Choice Fields for Status

This section discusses using choice fields for status in detail. It includes definitions, examples, and step-by-step procedures as followed during the development of the Educational Organisation Management System. Each step was carefully planned and executed to ensure consistency and reliability across all modules. The implementation approach not only follows ServiceNow best practices but also aligns with user requirements and business needs. Documentation, testing, and iterative improvement were part of every stage of the project lifecycle.

Purpose of Using a Choice Field:

Restricts input to defined values (avoids spelling mistakes and inconsistencies)

Helps automate workflows based on selected values

Supports conditional logic through UI Policies and Flow Designer

Makes form data easy to filter and report on

Steps to Create a Choice Field in ServiceNow

Step 1: Open the Admission Table

Navigate to System Definition > Tables

Search for and open the table u_admission

Step 2: Add a New Field

Scroll to the Columns section

Click New

Enter the following:

Column Label: Admission Status

Column Name: admission_status

Type: Choice

Step 3: Define Choice Values

In the Choice field settings, add each of the statuses:

New

In Progress

Joined

Rejoined

Rejected

Cancelled

Closed

Set one as the default value (e.g., New)

Example of Status Flow (Choice Values):

$$\begin{tikzcd}
\text{New} \arrow[r] \arrow[d] & \text{In Progress} \arrow[r] \arrow[d] & \text{Joined} \arrow[r] \arrow[d] & \text{Closed} \\
\text{Rejected} & \text{Rejoined} & \text{Cancelled} &
\end{tikzcd}$$

Client Scripts and UI Policies

This section discusses client scripts and ui policies in detail. It includes definitions, examples, and step-by-step procedures as followed during the development of the Educational Organisation Management System. Each step was carefully planned and executed to ensure consistency and reliability across all modules. The implementation approach not only follows ServiceNow best practices but also aligns with user requirements and business needs. Documentation, testing, and iterative improvement were part of every stage of the project lifecycle.

Purpose of Client Scripts and UI policies:

Improve the user experience by dynamically changing form behaviour

Enforce field-level validations

Reduce manual calculations and human error

Simplify data entry process

Client Scripts:

A script that runs in the browser when the form is loaded, changed, or submitted.

Example Used in This Project:

Script Name: Get Student name with Admission Number

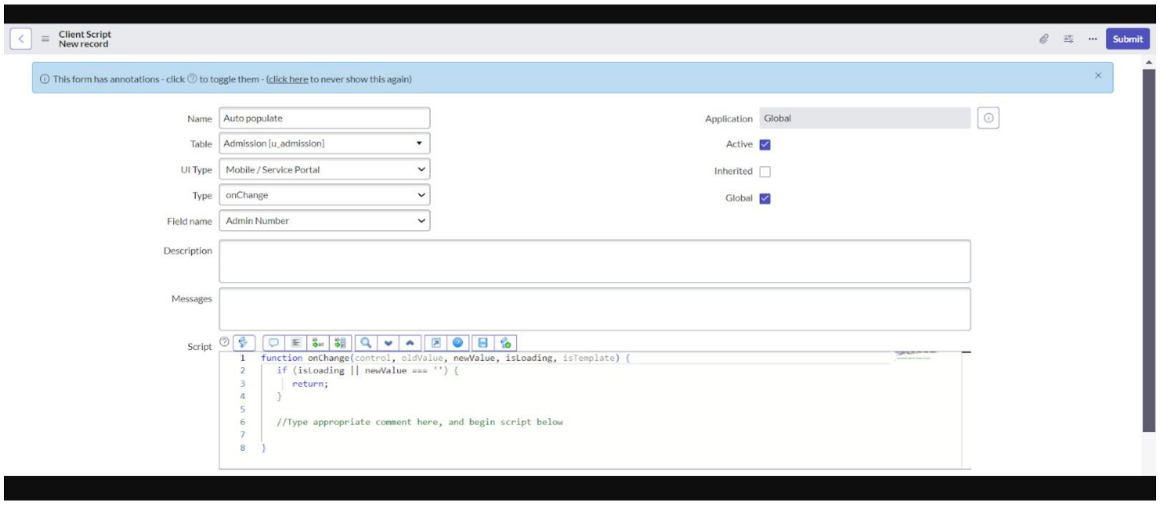
Table: Admission(u_admission)

Type: onChange

Field: Admission No

Examples of Client scripts with codes:

1.Admission:



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

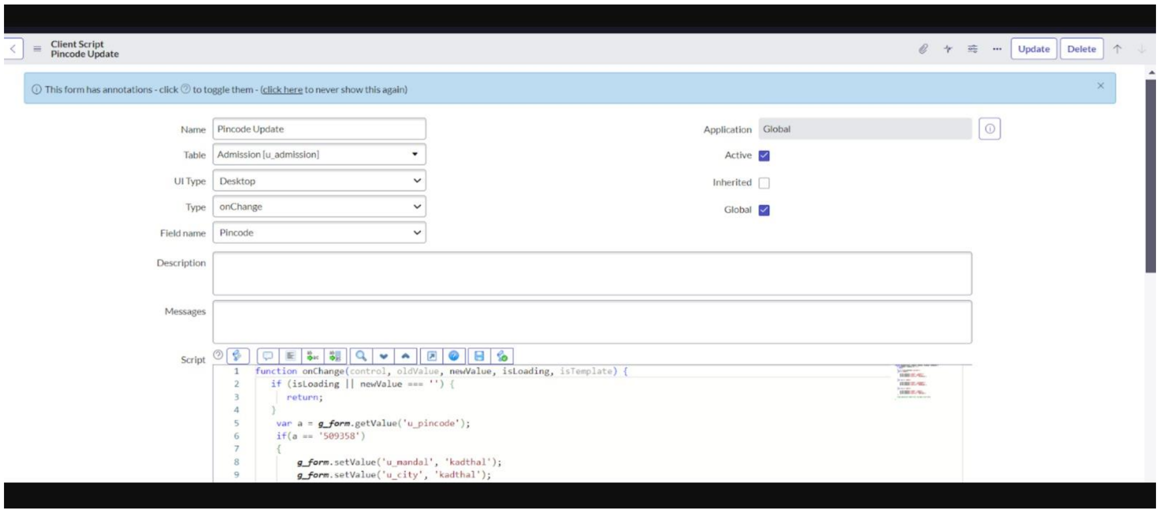
Name	Auto populate
Table	Admission [u_admission]
UI Type	Mobile / Service Portal
Type	onChange
Field name	Admin Number
Application	Global
Active	✓
Inherited	
Global	✓
Description	
Messages	

Script

```
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_cellNo);

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

}
```

2.Pincode:


Client Script
Pincode Update

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

Name	Pincode Update	Application	Global
Table	Admission [u_admission]	Active	✓
UI Type	Desktop	Inherited	☐
Type	onChange	Global	✓
Field name	Pincode		
Description			
Messages			

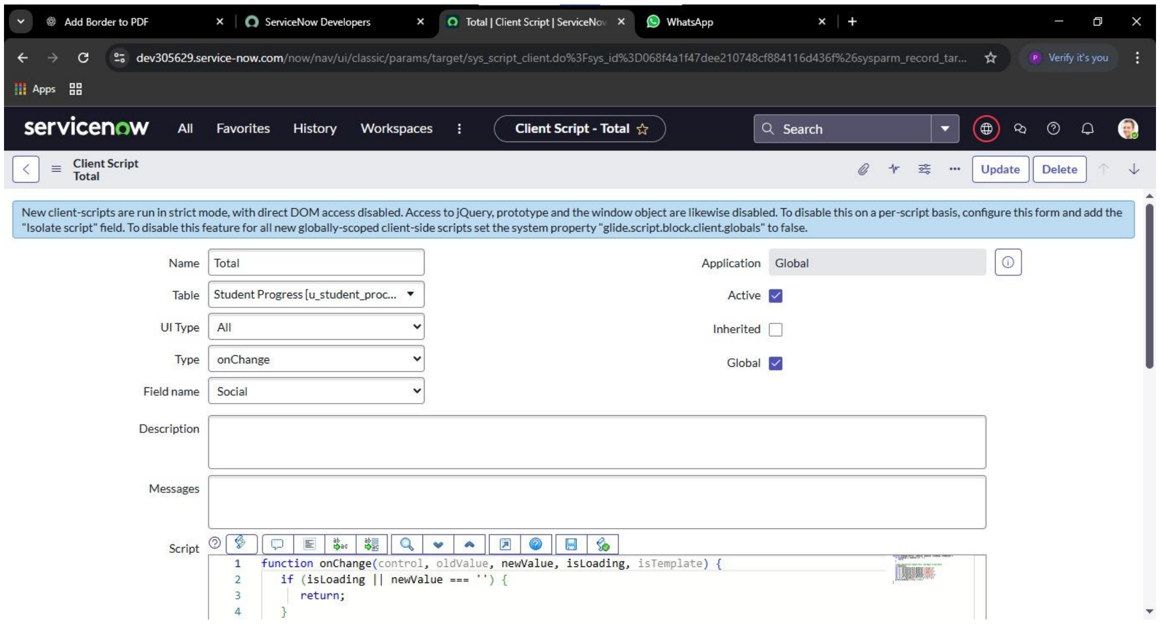
Script

```
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');
```

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

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

    return;
  }
   var a = g_form.getValue('u_pincode');
  if(x == '535547')
{
g_form.setValue('u_village', 'Tallavalasa');
g_form.setValue('u_mandal', 'Thagarapuvalasa');
g_form.setValue('u_city', 'Visakhapatanam');
g_form.setValue('u_district', ' Visakhapatanam ');
}
else if(x == '530027')
{
g_form.setValue('u_mandal', 'Bobbili');
g_form.setValue('u_city', 'Parvathipuram');
g_form.setValue('u_district', 'Vizianagaram');
}
else if(x == '535527')
{
g_form.setValue('u_mandal', ‘Kancherapalem’);
g_form.setValue('u_city', 'Narsipatnam');
g_form.setValue('u_district', 'East Godavari');
}
```

3.Total:

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

var t = parseInt(g_form.getValue('u_telugu'));

var h = parseInt(g_form.getValue('u_hindi'));

var  = 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);

}}
```


Add Border to PDF
ServiceNow Developers
Total | Client Script | ServiceNow
WhatsApp
dev305629.service-now.com/now/nav/ui/classic/params/target/sys_script_client.do%3Fsys_id%3D068f4a1f47dee210748cf884116d436f%26sysparm_record_tar...
Verify it's you
Apps
servicenow
All
Favorites
History
Workspaces
Client Script - Total
Search
Client Script
Total
Update
Delete

New client-scripts are run in strict mode, with direct DOM access disabled. Access to jQuery, prototype and the window object are likewise disabled. To disable this on a per-script basis, configure this form and add the "Isolate script" field. To disable this feature for all new globally-scoped client-side scripts set the system property "glide.script.block.client.globals" to false.

Name	Total	Application	Global
Table	Student Progress [u_student_proc...	Active	✓
UI Type	All	Inherited	
Type	onChange	Global	✓
Field name	Social		

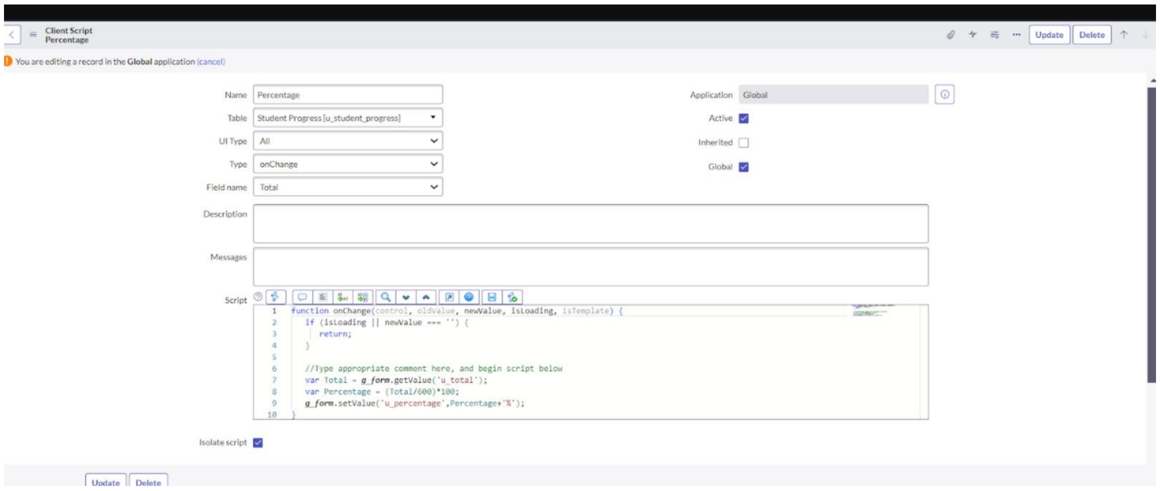
Description

Messages

Script

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

4.Percentage:



Client Script
Percentage

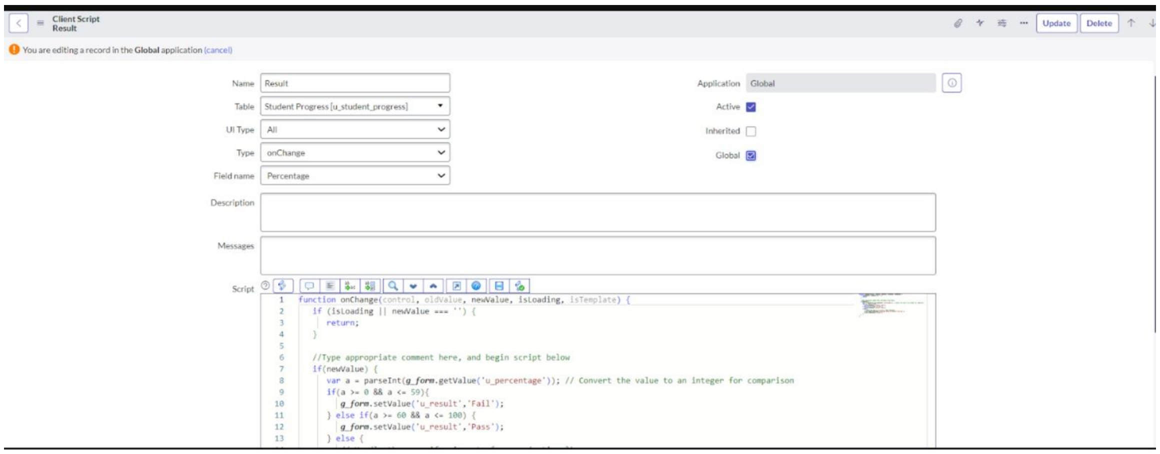
You are editing a record in the Global application (cancel)

Name	Percentage
Table	Student Progress [u_student_progress]
UI Type	All
Type	onChange
Field name	Total
Description	
Messages	
Application	Global
Active	✓
Inherited	
Global	✓
Isolate script	✓

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

```
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+'%');

}
```

5.Result:



Client Script
Result

You are editing a record in the Global application (cancel)

Name	Result
Table	Student Progress [u_student_progress]
UI Type	All
Type	onChange
Field name	Percentage
Description	
Messages	
Application	Global
Active	✓
Inherited	
Global	✓

```
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 {
```

```
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');
    }
  }
}
```

UI Policies:

A no-code rule to show/hide, make mandatory, or read-only a field based on conditions.

Example Used in This Project:

Policy Name: Show "Reason for Rejection" if Admission Status = Rejected

Table: Admission (u_admission)

Condition: Admission Status is Rejected

Actions:

Show the field Reason for Rejection

Make it mandatory

Flow Designer and Automation

This section discusses flow designer and automation in detail. It includes definitions, examples, and step-by-step procedures as followed during the development of the Educational Organisation Management System. Each step was carefully planned and executed to ensure consistency and reliability across all modules. The implementation approach not only follows ServiceNow best practices but also aligns with user requirements and business needs. Documentation, testing, and iterative improvement were part of every stage of the project lifecycle.

Purpose of Flow Designer:

Automate repetitive tasks and reduce manual errors

Trigger actions based on form submissions or field changes

Improve workflow efficiency and response time

Provide clear process logic using visual steps

Key Flows Implemented in the Project:

Flow 1: Notify Admin on Admission Status Change

Trigger: When a record in the Admission table is updated and the Admission Status field changes

Actions:

Look up the student's name from the reference field

Send an email to the admin with details like:

Student Name

Admission Number

Updated Status

Date of change

Flow 2: Set Admission Date Automatically

Trigger: When a new admission record is created

Actions:

If the Admission Status is Joined, auto-populate Admission Date with the current date

Flow 3: Audit Logging

Trigger: On record update

Actions:

Log the status change into an audit table (custom or system)Helps in tracking when and who changed the status

Steps to Create a Flow in ServiceNow:

1. Go to Flow Designer via Application Navigator

2. Click New > Flow

3. Enter flow name (e.g., "Admission Status Change Notification")

4. Choose a Trigger, such as "Record Updated" on u_admission

5. Add Actions:

Log Activity

Send Email

Update Record

6. Click Save and Activate the flow

Testing and Debugging

This section discusses testing and debugging in detail. It includes definitions, examples, and step-by-step procedures as followed during the development of the Educational Organisation Management System. Each step was carefully planned and executed to ensure consistency and reliability across all modules. The implementation approach not only follows ServiceNow best practices but also aligns with user requirements and business needs. Documentation, testing, and iterative improvement were part of every stage of the project lifecycle.

Objectives of Testing:

Validate data accuracy in custom tables (Student and Admission)

Ensure automation flows (e.g., email notifications, field updates) execute correctly

Verify form behaviors (e.g., UI policies, client scripts) function as designed

Detect and fix logical or configuration errors

Testing Performed:

1. Form Field Functionality

Created test records for both Student and Admission tables\

Verified field validations (mandatory, read-only)

Checked calculated fields like Age from Date of Birth

2. Client Script and UI Policy Testing

Changed values to see if fields appeared, became mandatory, or were automatically calculated

For example, when Admission Status = Rejected, the Reason for Rejection field appeared and became required

3. Flow Testing via Flow Designer

Modified admission status to see if email notifications were sent

Tested if Admission Date auto-filled when status was Joined

Verified conditions and actions through execution logs

Debugging Tools and Techniques:

1. System Logs

Navigated to System Logs > All to review:

Info logs

Warnings

Errors

2. Script Debugger

Enabled Debug Client Script or Debug Business Rule from browser console or developer tools

Added gs.info() statements to print variable values and flow steps

3. Flow Execution Details

Opened Flow Designer → Clicked on the flow → Viewed Execution History

Traced each step to ensure correct inputs and outputs

Fixes and Iterations

Fixed date format issues in DOB-to-Age script

Adjusted UI Policy conditions to handle empty values

Modified flow trigger conditions to avoid false positive

Exporting and Importing Update Sets

This section discusses exporting and importing update sets in detail. It includes definitions, examples, and step-by-step procedures as followed during the development of the Educational Organisation Management System. Each step was carefully planned and executed to ensure consistency and reliability across all modules. The implementation approach not only follows ServiceNow best practices but also aligns with user requirements and business needs. Documentation, testing, and iterative improvement were part of every stage of the project lifecycle.

Purpose of Update Set Migration:

Package and transport changes across instances

Maintain consistency in development, test, and production environments

Enable version control of customizations

Avoid manual re-creation of application elements

Steps to Export an Update Set:

Step 1: Complete the Update Set

Navigate to System Update Sets > Local Update Sets

Open the update set used for development (e.g., Educational Organisation Setup)

Ensure State = Complete

Click Close if the update set is not already closed

Step 2: Export to XML

Click Export to XML

A .xml file is downloaded containing all captured customizations

Steps to Import an Update Set:

Step 1: Navigate to Retrieved Update Sets

On the target instance, go to System Update Sets > Retrieved Update Sets

Click Import Update Set from XML

Upload the previously exported XML file

Step 2: Preview and Commit

After import, open the update set

Click Preview Update Set to check for conflicts or errors

If no errors are found, click Commit Update Set to apply the changes

Important Notes:

Only customizations (not data) are transferred through update sets

Always test on a staging instance before importing into production

Use naming conventions for update sets (e.g., edu_org_initial_build)

Avoid capturing changes from multiple applications in a single update set

Features Implemented and Technologies Used:

This section summarizes the major functionalities developed as part of the Educational Organisation Management System project, and the technologies and platform features used to implement them. The system was developed using the ServiceNow platform, taking full advantage of its low-code capabilities, automation tools, and integration features.

Features Implemented

1. Student Management

Capture and store personal, academic, and contact details of students

Auto-calculation of student age based on date of birth

2. Admission Tracking

Admission record creation and updates

Use of status field to track lifecycle stages (New, In Progress, Joined, etc.)

Flow Formatter for visualizing admission progress

3. Status Flow Automation

Admission process visualized using Flow Formatter

Automatic updates and email notifications via Flow Designer

4. Dynamic Form Behavior

Client Scripts to automate field values (e.g., Age)

UI Policies to dynamically show/hide and validate fields (e.g., Rejection Reason)

5. Custom Table Creation

Two custom scoped tables (u_student and u_admission)

Reference field used for relationship between student and admission

6. Update Set Migration

Changes tracked through update sets

Exported from development and imported to target instance

Technologies and Tools Used:

Technology / Tool	**Purpose**
ServiceNow Platform	Core development environment
Tables & Fields	Data structure definition
Form Designer	Designing and arranging form layouts
Flow Designer	Automating workflows and sending notifications
UI Policies	Dynamic form behavior (visibility, mandatory fields)
Client Scripts	Field value calculations and validation logic
Flow Formatter	Displaying admission process visually
Update Sets	Packaging and transporting changes across instances
System Logs	Debugging and tracing errors
Studio	Unified development workspace for app components

Benefits of Using ServiceNow for This Project:

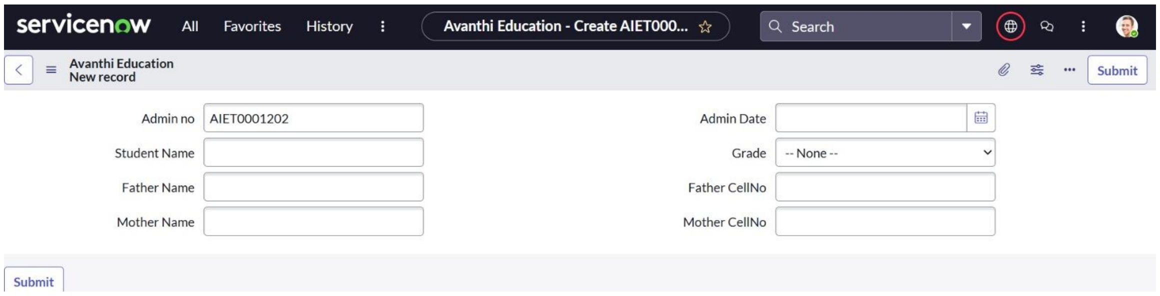
Rapid development with minimal coding

Easy form design and configuration

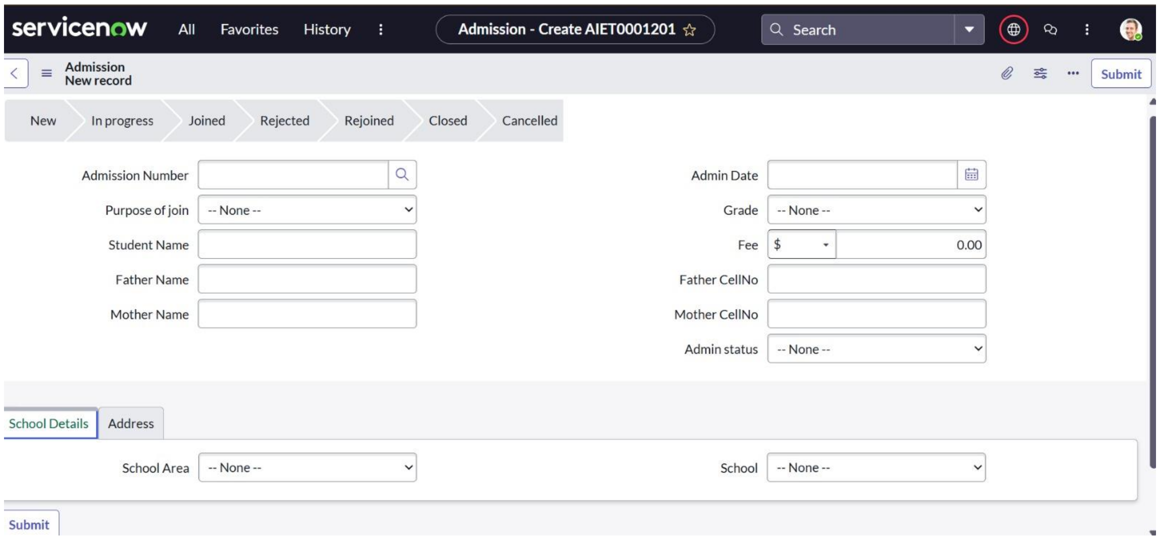
Strong integration and automation capabilities

Scalable and maintainable structure

Final Output:

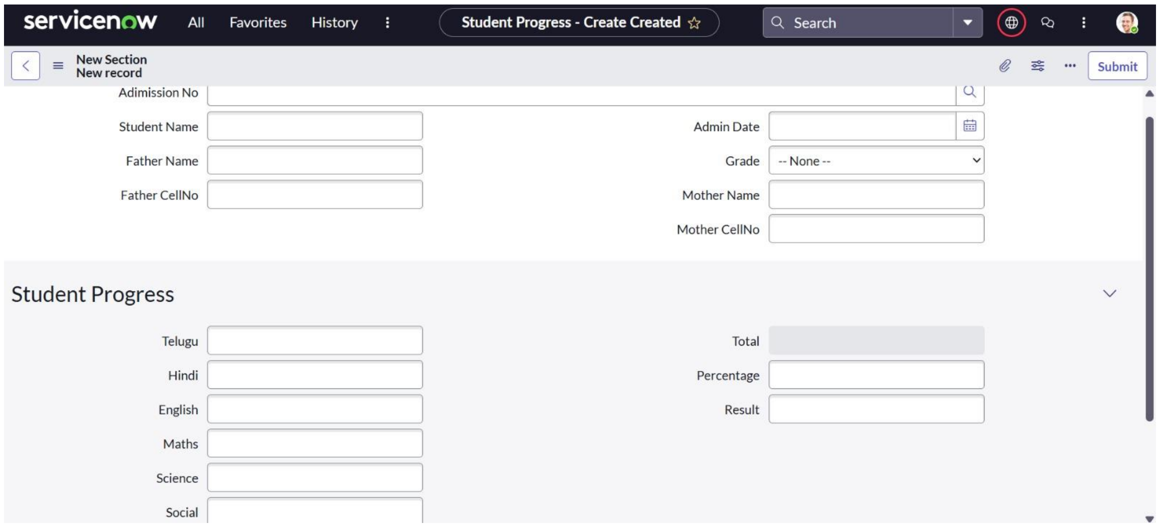


Admin no	AIET0001202	Admin Date	
Student Name		Grade	-- None --
Father Name		Father CellNo	
Mother Name		Mother CellNo	



Admission Number		Admin Date	
Purpose of join	-- None --	Grade	-- None --
Student Name		Fee	$ 0.00
Father Name		Father CellNo	
Mother Name		Mother CellNo	
		Admin status	-- None --

School Area	-- None --	School	-- None --



Admission No			
Student Name		Admin Date	
Father Name		Grade	-- None --
Father CellNo		Mother Name	
		Mother CellNo	

Student Progress

Telugu		Total	
Hindi		Percentage	
English		Result	
Maths			
Science			
Social			

Conclusion

The "Educational Organisation Using ServiceNow" project successfully demonstrates how the ServiceNow platform can be utilized to design and manage key administrative functions within an educational institution. By creating custom tables, forms, workflows, and client scripts, we developed a streamlined system capable of handling student admissions, teacher data, and student performance tracking.

This project highlights the practical benefits of low-code/no-code development, showing how powerful enterprise solutions can be built rapidly using ServiceNow's suite of tools. The automation of admissions and student management not only reduces manual effort but also increases accuracy, traceability, and transparency in institutional processes.

Through this project, we gained valuable hands-on experience with ServiceNow’s application development environment, including its update sets, flow designer, form customization, and scripting capabilities. This foundational knowledge sets the stage for further expansion of the system to include advanced features such as notifications, reporting, and portal integration in the future.

Ultimately, this project serves as a scalable and efficient model for modernizing educational management systems using ServiceNow.

Project Name: Educational Organisation

Team ID : LTVIP2025TMID28678

Team Size : 5

Team Leader : Koneti Sairam Avinash

Team member : Siraparapu Lakshmi Prasanna

Team member : Veerapindi Saikumari

Team member : Chandra Shaker

Team member : Maripi Simhadri