



servicenow



Smart
Internz



KSK COLLEGE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

EDUCATIONAL ORGANIZATIONAL USING SERVICENOW

Team ID : NM2025TMID05478

Team Size : 4

Team Leader : Nithish C

Team member : Kumaresan G

Team member : Vasanth R

Team member : Vinith R

PROBLEM STATEMENT:

Educational institutions often struggle with managing large volumes of student and staff information through manual or disconnected systems. Tasks such as admissions, student record management, and progress tracking become time-consuming and prone to errors. The lack of a unified digital platform leads to inefficiency, communication gaps, and delays in decision-making.

This project addresses these challenges by developing an **Educational Management System** on the **ServiceNow platform** to automate administrative workflows, centralize data, and improve institutional efficiency.

OBJECTIVES:

To design and implement a centralized system for managing student and teacher information using ServiceNow.

1. To automate key processes such as admissions, attendance, and progress monitoring.
2. To enable real-time access and updates of academic and administrative data.
3. To improve coordination among departments and reduce manual workloads.
4. To generate reports and analytics for better academic and administrative decision-making.
5. To enhance transparency and user experience for students, faculty, and administrators.

Skills and Technologies Used:

Platform: ServiceNow

Skills:

- ServiceNow Administration and Configuration
- Form and Table Design
- Workflow and Flow Designer
- Reporting and Dashboard Creation

Task Initiation:

The **Educational Organization Using ServiceNow** project begins with the initiation phase, where the project scope, goals, and resources are clearly defined. The purpose of this phase is to establish a clear understanding of what the project aims to achieve and to prepare the foundation for successful implementation.

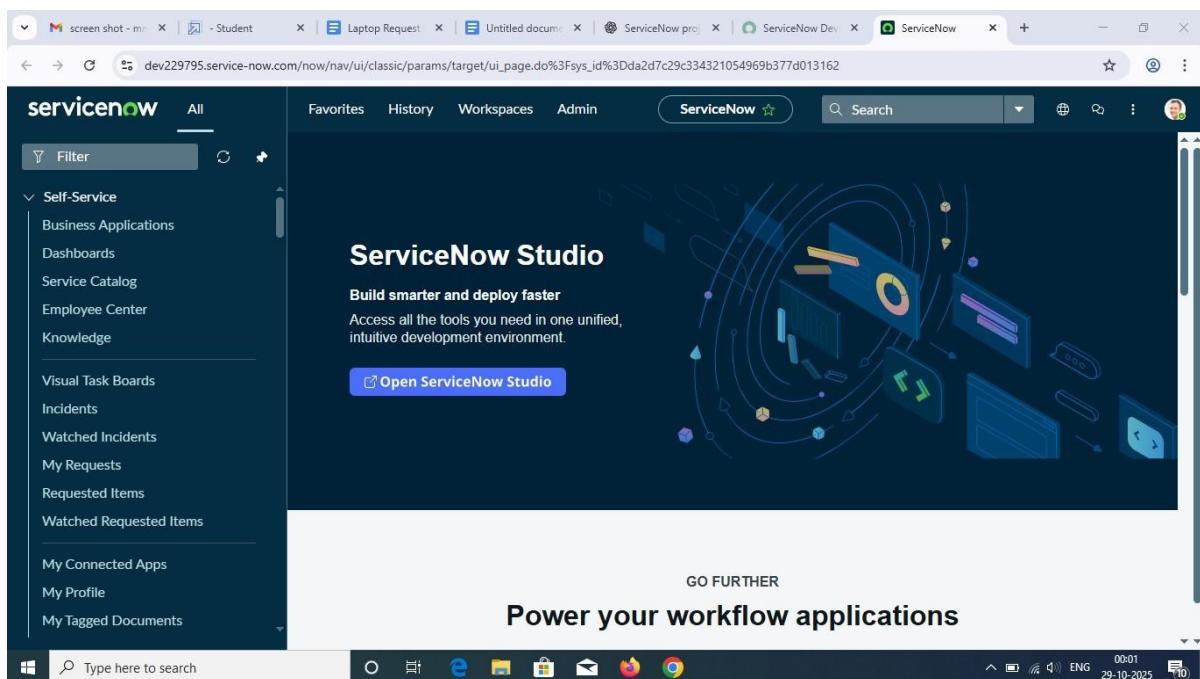
Features:

1. Centralized management of student and teacher information on a single ServiceNow platform.
2. Automated workflows for admissions, attendance, and progress tracking.
3. Real-time data access with reporting and dashboard analytics.
4. User-friendly interface for administrators, faculty, and students to improve efficiency.

STEPS:

STEP1:SettingupServiceNowInstance

- Sign up for a developer account on the ServiceNow Developer site "<https://developer.servicenow.com>".
- Once logged in, navigate to the "Personal Developer Instance" section.
- Click on "Request Instance" to create a new ServiceNow instance.
- Fill out the required information and submit the request.
- You'll receive an email with the instance details once it's ready.
- Log into your ServiceNow instance using the provided credentials.
- Now you will navigate to the ServiceNow



STEP2:CreatingAnUpdateSet

- Click on All >> Local update sets.
- Click on new.
- Enter the Details Name: Educational Organisation >> Click on Submit and make Current.

The screenshot shows the ServiceNow home page with a dark blue background. At the top, there is a search bar containing the text "local update". Below the search bar is a navigation menu with options like "All", "Favorites", "History", and "Workspaces". To the right of the menu is a "ServiceNow" logo and a user icon. A sidebar on the left displays search results for "local update", showing "No Results" under "FAVORITES" and "ALL RESULTS" showing "System Update Sets" and "Local Update Sets". At the bottom of the page, there is a promotional banner with the text "GO FURTHER" and "Power your workflow applications".

The screenshot shows the "Update Set - Educational Organisation" form. The title bar says "Update Set - Educational Organisation". The form contains the following fields:

- * Name: Educational Organisation
- State: Complete
- Parent: (empty)
- Release date: (empty)
- Install date: (empty)
- Installed from: (empty)
- Description: (empty)

On the right side of the form, there are additional details:

- Application: Global
- Created: 2025-10-24 22:14:27
- Created by: admin
- Merged to: (empty)

At the bottom of the form, there are "Update" and "Back Out" buttons. Below the form, there is a "Related Links" section with links to "Export to XML", "Merge With Another Update Set", and "Scan Update Set". At the very bottom, there is a footer with a search bar, a toolbar with icons for various applications, and system status information.

STEP 3: Creating A Table

Salesforce:

- All>>Tables.
- Clickonnew.
- Enter the Label (Anything you want): Salesforce >>
Click on Name it will
Automatically generate Api
name.
- Create columns as given
below, Double Click on
Column label and Enter the
Column labels and click on
thetickmark>>GiveTypeas
given.
- For “Admin Number” Give
Display as True and right
click on the toggle bar on top
>>save.
- Click on controls >> Enable
Extensible.
- Clickon“AdminNumber”

column, In Related Links

Click on Advanced View >>

Default View (Enable Use

dynamic default) >> select

Get Next Padded Number in

Dynamic default value

>>Update.

- Click on “Grade” Column >>

Click on Choices and give

Label, Value and Sequence as

given below.

Column label	Type	Reference	Max length	Default value	Display
Updates	Integer	(empty)	40		false
Updated by	String	(empty)	40		false
Updated	Date/Time	(empty)	40		false
Sys ID	Sys ID (GUID)	(empty)	32		false
Student Name	String	(empty)	40		false
Mother Name	String	(empty)	40		false
○ Mother cell	String	(empty)	40		false
Grade	Choice	(empty)	40		false
Father Name	String	(empty)	40		false
Father cell	String	(empty)	40		false
Created by	String	(empty)	40		false
Created	Date/Time	(empty)	40		false
Class	System Class Name	(empty)	80	javascript:current.getTableName();	false
Admin Number	String	(empty)	40	javascript:getNextObjNumberPadded();	true
Admin Date	Date	(empty)	40		false
Insert a new row...					

Admission:

- Create an Admission Table with Columns given.
- Select Extends Table >> Salesforce and also Select Add module to menu >> Salesforce.
- Create Fields as shown:

The screenshot shows the ServiceNow interface for creating a new table named 'Admission'. The left sidebar is open, showing categories like System Archiving, System Clone, System Definition, and Tables. Under Tables, 'Tables & Columns' is selected. The main panel displays the 'Table - Admission' configuration. It shows the table name 'Admission', label 'u_admission', application 'Global', and extends table 'Salesforce'. The 'Columns' tab is selected, showing a list of columns with their labels, types, and default values. The table has 29 columns, with the first few labeled: Admin Number (Integer), Admin Status (Choice), Area (String), City (String), and Class (System Class Name). The 'Display' column indicates whether the field is visible or not.

Column label	Type	Reference	Max length	Default value	Display
Admin Number	Integer	(empty)	40		true
Admin Status	Choice	(empty)	40	40	false
Area	String	(empty)	40	40	false
City	String	(empty)	40	40	false
Class	System Class Name	(empty)	80	javascript:current.getTableName();	false

- Create choice for Admin Status as:

The screenshot shows the 'Dictionary Entry' screen for 'Admin Status'. The top navigation bar includes 'Create Choice List', 'Delete Column', and 'Update'. Below this is a 'Related Links' section with links to 'Main Table', 'Run Portlet View', and 'Administration'. The main area is titled 'Choices' and contains a table with columns: Label, Value, Language, Sequence, Active, and Updated. The table lists seven entries: 'New', 'In progress', 'Assigned', 'Rejected', 'Closed', 'Reopened', and 'Cancelling'. Each entry has a timestamp in the 'Updated' column, ranging from 2024-01-02 21:20:25 to 2024-01-02 21:55:09.

Label	Value	Language	Sequence	Active	Updated
New	New	en	1	True	2024-01-02 21:20:25
In progress	In progress	en	2	True	2024-01-02 21:55:09
Assigned	Assigned	en	3	True	2024-01-02 21:55:06
Rejected	Rejected	en	4	True	2024-01-02 21:55:00
Closed	Closed	en	5	True	2024-01-02 21:55:05
Reopened	Reopened	en	6	True	2024-01-02 21:55:00
Cancelling	Cancelling	en	7	True	2024-01-02 21:55:09

- Create choice for Pincode as:

The screenshot shows the 'Dictionary Entry' interface for a field named 'Pincode'. The top section displays basic metadata: Column label ('Pincode'), Column name ('u_pincode'), and various checkboxes for Function field, Read only, Mandatory, and Display. Below this is a 'Choice List Specification' panel with tabs for 'Choice List Specification' and 'Default Value'. It contains a help text box and a dropdown menu labeled 'Choice' set to 'Dropdown with -- None --'. At the bottom of the panel are buttons for 'Create Choice List', 'Delete Column', and 'Update'. Underneath the specification panel is a 'Related Links' section with links to 'Show Table', 'Run Point Scan', and 'Advanced view'. The main content area is titled 'Choices' and contains a table with columns: Label, Value, Language, Sequence, Inactive, and Updated. The table lists three entries: 500079, 509358, and 500081, all in English (en) and marked as inactive. A 'New' button is located at the top right of the choices table.

Label	Value	Language	Sequence	Inactive	Updated
500079	500079	en	2	false	2025-10-27 08:32:22
509358	509358	en	1	false	2025-10-27 08:31:58
500081	500081	en	3	false	2025-10-27 08:32:46

- Create choice for Purpose of Join as:

The screenshot shows the 'Dictionary Entry' interface for a field named 'Purpose of join'. The top section displays basic metadata: Column label ('Purpose of join'), Column name ('u_purpose_of_join'), and various checkboxes for Function field, Read only, Mandatory, and Display. Below this is a 'Choice List Specification' panel with tabs for 'Choice List Specification' and 'Default Value'. It contains a help text box and a dropdown menu labeled 'Choice' set to 'Dropdown with -- None --'. At the bottom of the panel are buttons for 'Create Choice List', 'Delete Column', and 'Update'. Underneath the specification panel is a 'Related Links' section with links to 'Show Table', 'Run Point Scan', and 'Advanced view'. The main content area is titled 'Choices' and contains a table with columns: Label, Value, Language, Sequence, Inactive, and Updated. The table lists three entries: Coaching, Tution, and Teacher, all in English (en) and marked as inactive. A 'New' button is located at the top right of the choices table.

Label	Value	Language	Sequence	Inactive	Updated
Coaching	Coaching	en	2	false	2025-10-27 08:34:16
Tution	Tution	en	1	false	2025-10-27 08:33:52
Teacher	Teacher	en	3	false	2025-10-27 08:34:36

- Create choice for School as:

The screenshot shows the ServiceNow Dictionary Entry - School screen. At the top, there are fields for Column label (School) and Column name (u_school). Below these are checkboxes for Function field, Read only, Mandatory, and Display. A tab labeled "Choice List Specification" is selected, showing a dropdown menu set to "Dropdown with -- None --". Below this are buttons for Create Choice List, Delete Column, and Update. Under "Related Links", there are Show Table, Run Point Scan, and Advanced view options. At the bottom, there is a table titled "Choices" with columns: Label, Value, Language, Sequence, Inactive, and Updated. Two entries are listed: "Stanley" (Value: Stanley, Language: en, Sequence: 1, Inactive: false, Updated: 2025-10-27 08:35:51) and "Naresh It" (Value: Naresh It, Language: en, Sequence: 2, Inactive: false, Updated: 2025-10-27 08:36:27). A "New" button is visible at the top right of the choices table.

- Create choice for School Area as:

The screenshot shows the ServiceNow Dictionary Entry - School Area screen. At the top, there are fields for Column label (School Area) and Column name (u_school_area). Below these are checkboxes for Function field, Read only, Mandatory, and Display. A tab labeled "Choice List Specification" is selected, showing a dropdown menu set to "Dropdown with -- None --". Below this are buttons for Create Choice List, Delete Column, and Update. Under "Related Links", there are Show Table, Run Point Scan, and Advanced view options. At the bottom, there is a table titled "Choices" with columns: Label, Value, Language, Sequence, Inactive, and Updated. Two entries are listed: "Near Market" (Value: Near Market, Language: en, Sequence: 1, Inactive: false, Updated: 2025-10-27 08:37:57) and "Near Bus Stand" (Value: Near Bus Stand, Language: en, Sequence: 2, Inactive: false, Updated: 2025-10-27 08:38:23). A "New" button is visible at the top right of the choices table.

StudentProgress:

- Create a Student Progress Table with Columns given.
- Select Add module to menu >> Salesforce.
- Create Fields as shown:

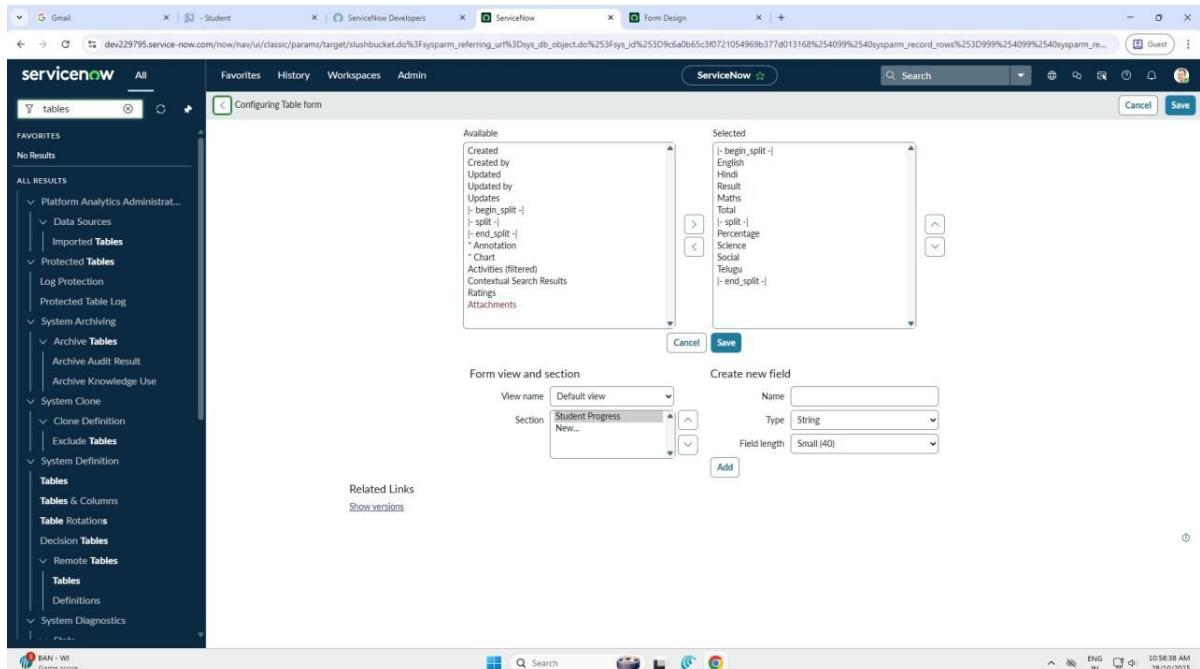
The screenshot shows the ServiceNow Table - Student Progress interface. The top navigation bar includes links for All, Favorites, History, Workspaces, and Admin. The title bar displays "Table - Student Progress". The main content area is titled "Table Columns" with a search bar. A table lists various fields with their properties:

Column label	Type	Reference	Max length	Default value	Display
Updates	Integer	(empty)	40	false	
Updated by	String	(empty)	40	false	
Updated	Date/Time	(empty)	40	false	
Total	String	(empty)	40	false	
Telugu	String	(empty)	40	false	
Sys ID	Sys ID (GUID)	(empty)	32	false	
Social	String	(empty)	40	false	
Science	String	(empty)	40	false	
Result	String	(empty)	40	false	
Percentage	String	(empty)	40	false	
Maths	String	(empty)	40	false	
Hindi	String	(empty)	40	false	
English	String	(empty)	40	false	
Created by	String	(empty)	40	false	
Created	Date/Time	(empty)	40	false	
Admission Number	Reference	Salesforce	32	false	

At the bottom, there are buttons for Delete, Update, and Delete All Records.

STEP4:FormLayout

- In the Student Progress Table Page, Click on Layout form.
- Click on Admission Number [+].
- Select below Admission Number fields in Available side and send it to selected side as below >> save.



STEP5:FormDesign

ForSalesforceTable:

- All>>SystemDefinition>>Tables.
- InLabelSearchforSalesforceandopen.
- RightClickontopToggle>>Configure>>FormDesign.
- Indropdownselect Salesforce(u_salesforce).
- Draganddropthefieldstotheleftsideasbelow.
- Save.

The screenshot shows the 'Form Design' interface for the 'Salesforce (u_salesforce)' table. On the left, there's a sidebar with tabs for 'Fields' (selected) and 'Field Types'. Under 'Fields', there are sections for 'Filter', 'Class', 'Created', 'Created by', 'Updated', 'Updated by', 'Updates', 'Formatters', 'Activities (Shared)', 'Contextual Search Results', and 'Ratings'. The main area is titled 'Salesforce (u_salesforce)' and contains a grid of fields. The fields are arranged in pairs: Admin Number and Father Name, Admin Date and Mother Name, Grade and Father Cell, and Student Name and Mother Cell. Each field has a small circular icon with a dot and a 'Format' button to its right. A '3 Columns' button is located at the top right of the grid area.

ForAdmissionTable:

- FollowthesamestepsasActivity1,Configurethefieldsasbelowandsave.

The screenshot shows the 'Form Design' interface for the 'Admission (u_admission)' table. On the left, there's a sidebar with tabs for 'Fields' (selected), 'Filter', 'Formatters', 'Activities (Shared)', 'Contextual Search Results', and 'Ratings'. The main area is titled 'Admission (u_admission)' and contains three distinct sections. The first section is 'Admission Details' and includes fields: Admission Number, Present of year, Student Name, Father Name, Mother Name, and Gender. The second section is 'Fathers' and includes fields: Present, Work, and House No. The third section is 'Others' and also includes fields: Present, Work, and House No. Each field has a small circular icon with a dot and a 'Format' button to its right. A '3 Columns' button is located at the top right of each section area.

For StudentprogressTable:

- Follow the same steps as Activity 1, Configure the fields as below and save.

Form Design

StudentProgress [x] [edit] Detail view [x]

Fields Field Type

File

Fields

- Address
- Class
- Created
- CreatedBy
- Deleted
- DeletedBy
- Updated

Formations

- Activities (internal)
- Candidate Search Results
- Referrals

Form Design

New Student

Address Number:

1

Address Number - Grade: Address Number Father Name:
Address Number Student Name: Address Number Mother Name:
Address Number Father DOB: Address Number Mother DOB:

Student Progress

Days: Test:
Hours: Percentage:
Hours: Result:
Math:
Science:

STEP6:NumberMaintenance

Creating Number Maintenance for Admin Number

The screenshot shows the ServiceNow interface with the search bar containing 'number'. The results table has a header row with columns: Reference, Max length, Default value, and Display. Below the header, there are six rows of data. The first row is highlighted with a red box and a red arrow points to the 'Number Maintenance' link in the 'Reference' column. The other five rows represent system fields: Created by, Created, Sys ID, Updates, and Updated by.

Reference	Max length	Default value	Display
Class Name	(empty)	80 javascript:current.getTableName();	false
Created by	String	(empty)	false
Created	Date/Time	(empty)	false
Sys ID	Sys ID (GUID)	(empty)	false
Updates	Integer	(empty)	false
Updated by	String	(empty)	false

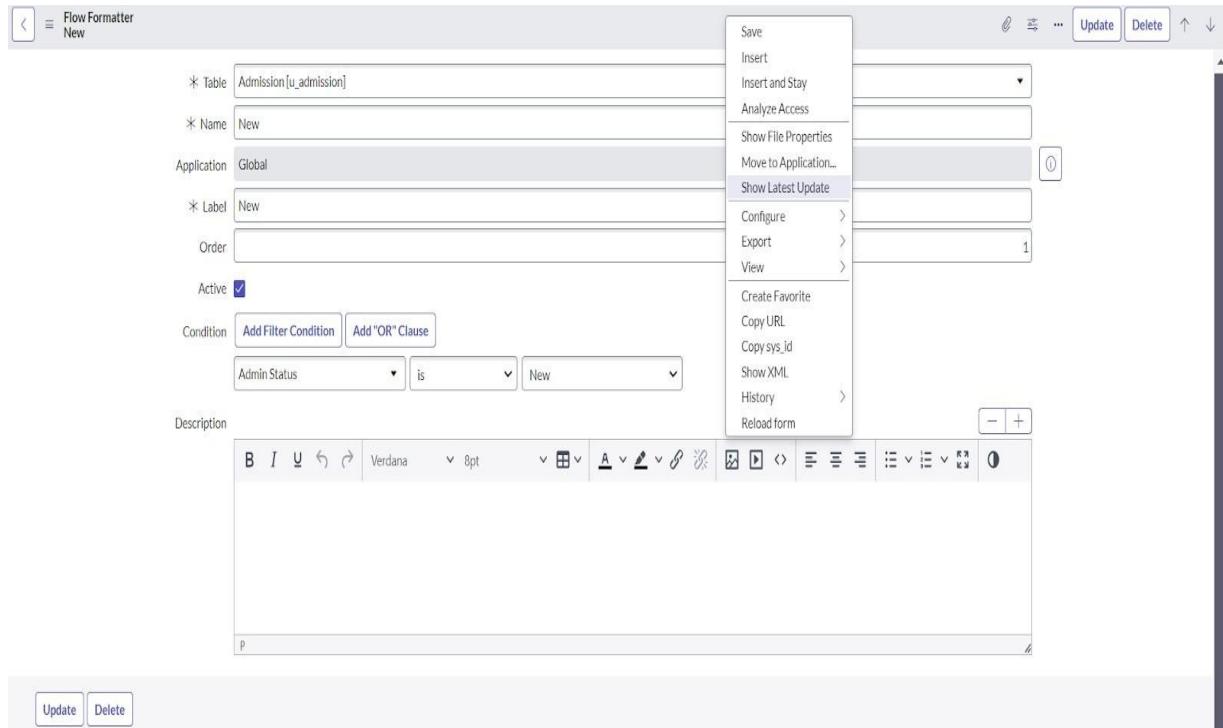
The screenshot shows the 'Number SAL' record creation form. The fields are: * Table (Salesforce), Prefix (SAL), * Number (1100), Application (Global), and Number of digits (3). At the bottom, there are 'Update' and 'Delete' buttons, and a 'Related Links' section with a 'Show Counter' link.

- Fill the details >> Submit.

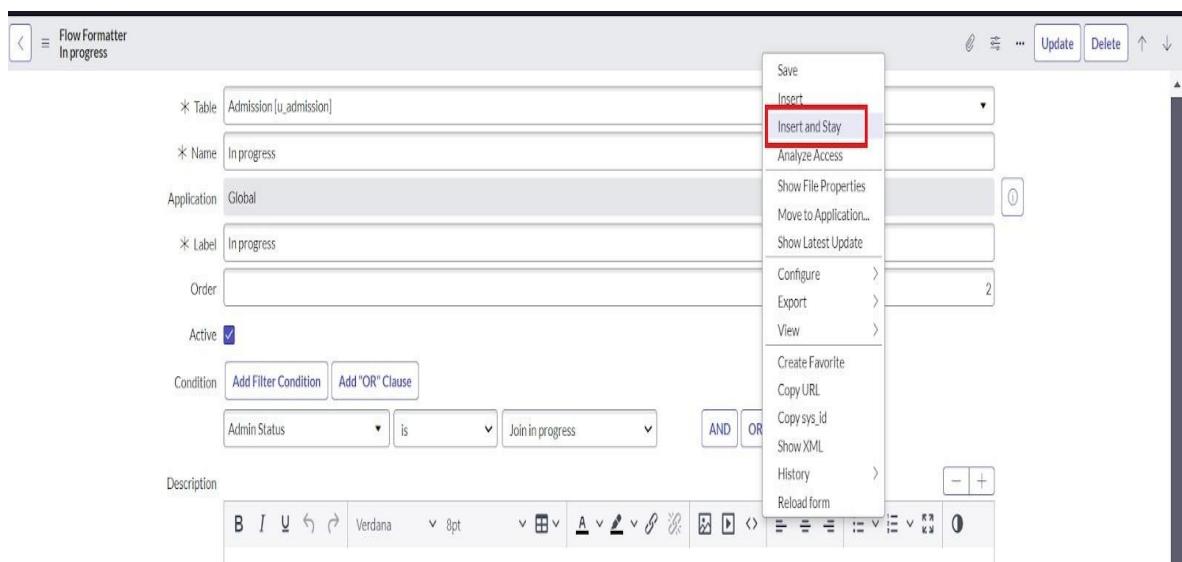
STEP7:ProcessFlow

1. Creating Process Flow for Admission Table

- All >> Process Flow >> New.
- Fill the Details as given Below



- Right Click to toggle and click on the save.
- Replace the Name and Label as below and click on Insert on stay.

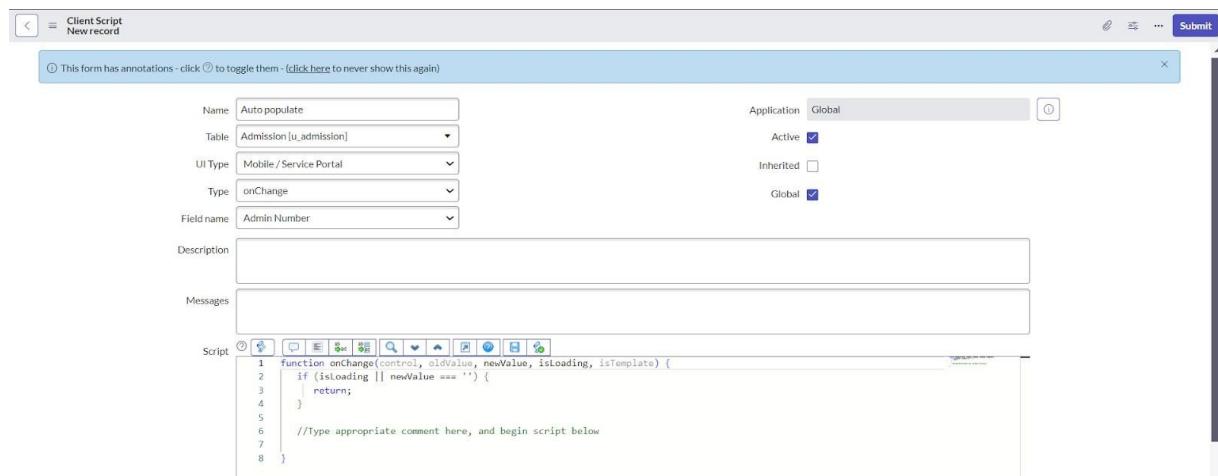


- Replace the Name and Label in order and click on Insert on stay.Joined>> Rejected>> Rejoined>> Closed >> Cancelled.
- Ordershould beNew>> InProgress >> Joined >> Rejected >> Rejoined >>Closed >>Cancelled.

STEP8:ClientScript

1. Creating“Autopopulate”ClientScriptsforAdmissionTable

- All>>ClientScripts>>New.
- FilltheDetailsasgiven.



- Write the Code as below, Enable Isolate script and Save.

```

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

```

2. Creating “PincodeUpdate” Client Scripts for Admission Table

- Fill the Details as given.

The screenshot shows a software interface for managing client scripts. At the top, there's a header bar with buttons for back, forward, search, and other application functions, along with 'Update' and 'Delete' buttons. Below the header is a message bar stating, "This form has annotations - click ⓘ to toggle them - (click here to never show this again)". The main area contains several configuration fields:

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

Below these fields are two large text input boxes: "Description" and "Messages". Underneath the "Description" box is a "Script" section containing 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');
}
```

WritetheCode asbelow,Enable IsolatescriptandSave.

```
functiononChange(control,oldValue,newValue,isLoading,isTemplate){if
(isLoading || newValue === "") {
    return;
}
vara=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');

}
elseif(a=='500081')
{
g_form.setValue('u_mandal', 'karmanghat');
g_form.setValue('u_city', 'karmanghat');
g_form.setValue('u_district', 'RangaReddy');

}
elseif(a=='500079')
{
g_form.setValue('u_mandal', 'Abids');
g_form.setValue('u_city', 'AsifNagar');
g_form.setValue('u_district', 'Hyderabad');

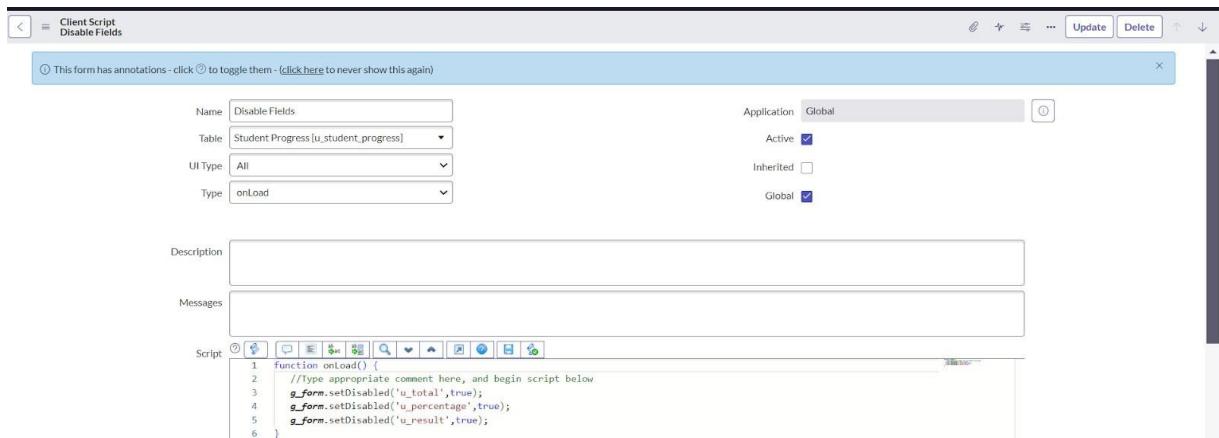
}

//Typeappropriatecommenthere, andbeginscriptbelow

}
```

3. Creating “DisableFields” ClientScripts for StudentprogressTable

- Fill the Details as given.



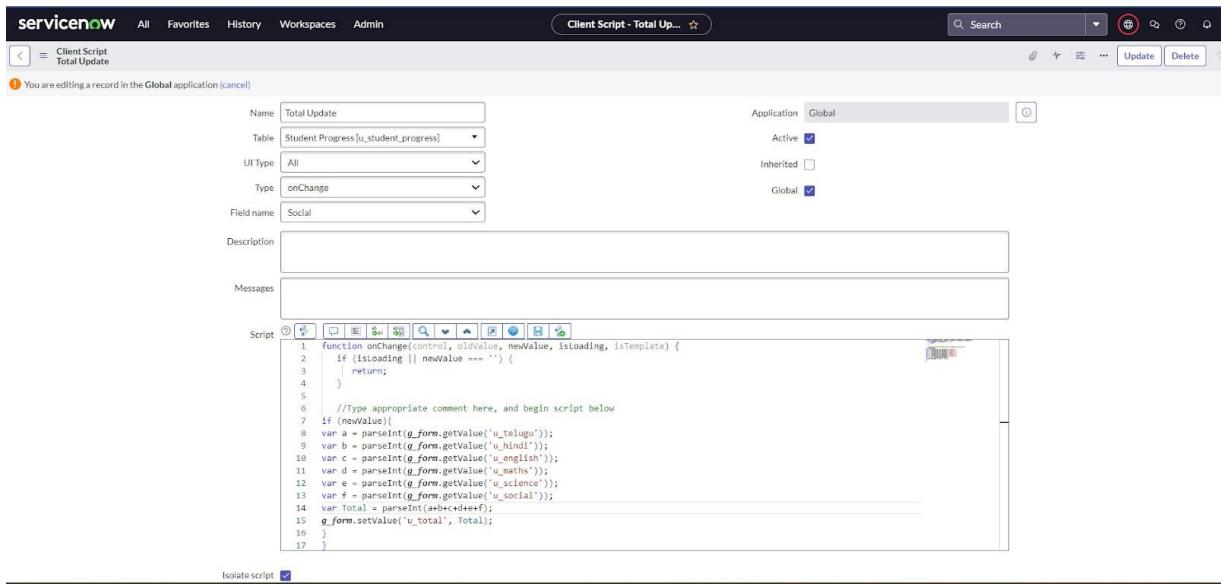
- Write the Code as below, Enable Isolate script and Save. `function onLoad() {`

```
//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);

}
```

4. Creating “TotalUpdate” ClientScripts for StudentprogressTable

- Fill the Details as given.



- Write the Code as below, Enable Isolate script and Save.

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

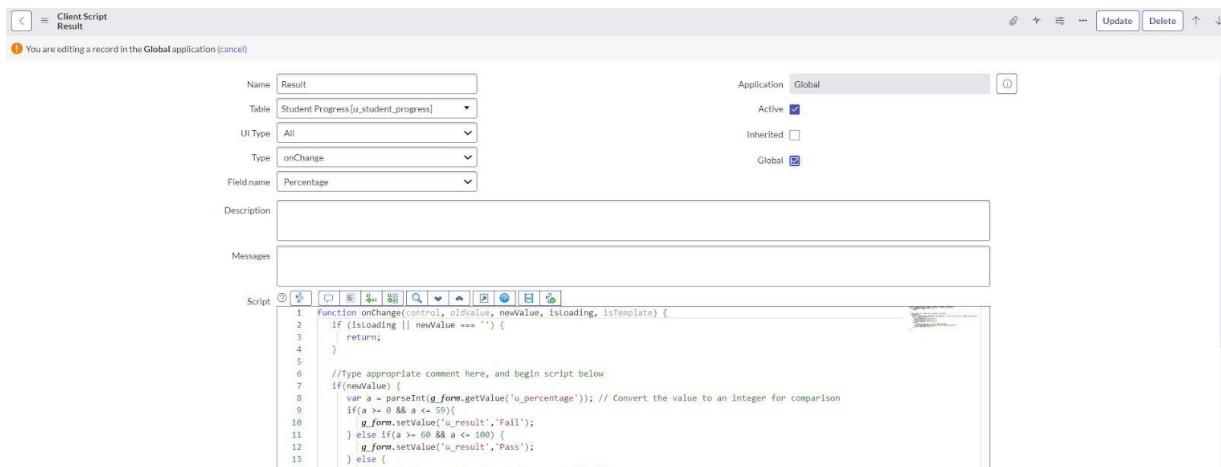
```
(isLoading || newValue === "") {  
    return;  
}
```

```
//Type appropriate comment here, and beginscript 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);  
}
```

Creating “Result” ClientScripts for StudentprogressTable

- Fill the Details as given.



- 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');

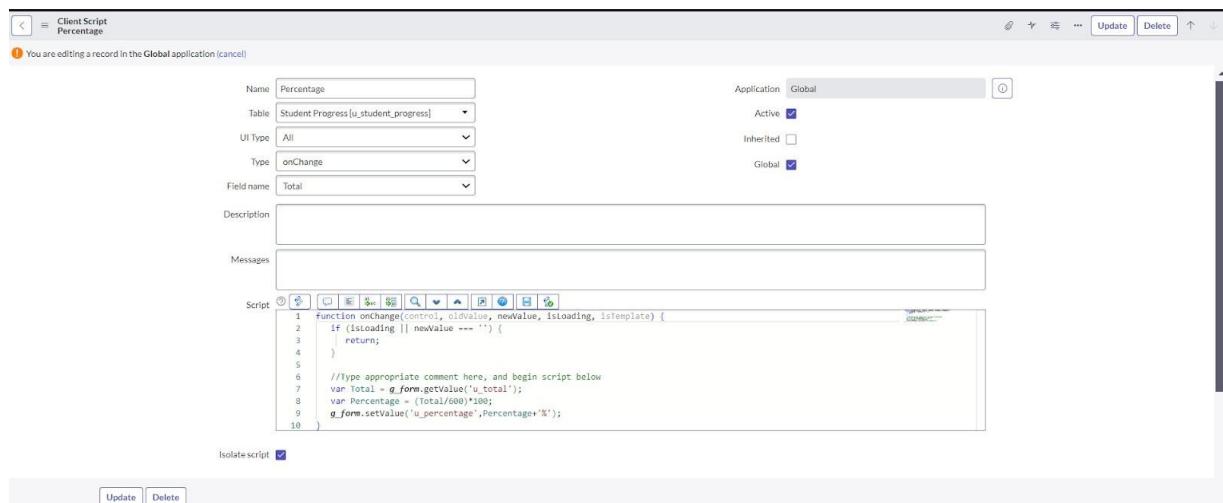
}else{
    // Handle the case if a is out of range (optional)
    g_form.addErrorMessage('Percentage
should be between 0 and 100.');
    g_form.clearValue('u_result');

}
}

}
```

5. Creating “Percentage” Client Scripts for Student progress Table

- Fill the Details as given.



- 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+'%');

}
```

Result:

The ServiceNow-based Educational Management System successfully demonstrates the automation of student admission, academic progress, and information management. Data is centralized, workflows are simplified, and administrators can easily generate reports and monitor academic progress.

The image displays three screenshots of the ServiceNow-based Educational Management System interface, illustrating the automation of student admission, academic progress, and information management.

Screenshot 1: Student Admission Form

This screenshot shows the "New record" screen for student admission. It includes fields for Admin Number (SAL0001078), Admin Date, Grade (dropdown: -- None --), Student Name, Father Name, Mother Name, Mother Cell, Father Cell, and a "Submit" button.

Screenshot 2: Student Progress Form

This screenshot shows the "New Section" screen for student progress. It includes fields for Admission Number, Grade (dropdown: -- None --), Student Name, Father Name, Mother Name, Father Cell, Mother Cell, and a "Submit" button. Below this is a "Student Progress" section with fields for Telugu, Hindi, English, Maths, Science, Social, Total, Percentage, and Result.

Screenshot 3: Admission Workflow Form

This screenshot shows the "Admission" screen with a workflow status bar at the top: New, In progress, Joined, Rejected, Rejoined, Closed, and Cancelled. It includes fields for Admission Number, Purpose of join (dropdown: -- None --), Student Name, Father Name, Mother Name, Admin Date, Grade, Fee (\$ dropdown: 0.00), Father Cell, Mother Cell, Admin Status (dropdown: -- None --), Comments, School Details (School Area dropdown: -- None --), Address, School (dropdown: -- None --), and a "Submit" button.

CONCLUSION:

The project “**Educational Organization Using ServiceNow**” effectively addresses the common challenges faced by educational institutions in handling administrative and academic data. Through automation and centralization, it enhances efficiency, transparency, and collaboration across departments.

By leveraging ServiceNow’s powerful workflow and reporting tools, the system reduces human error, saves time, and supports data-driven decisions.

This project stands as a practical example of how modern cloud platforms like **ServiceNow** can revolutionize educational management systems, paving the way for a smarter and more connected academic environment.

