

Educational Organisation Using ServiceNow

1. Objective

- The objective of this project is to develop an **Educational Management System** using **ServiceNow** to automate and simplify the key administrative processes of educational institutions. The system aims to enhance productivity by managing student and teacher data, streamlining admissions, and monitoring academic progress through a centralized and user-friendly interface.
-

2. Introduction

- Educational institutions often face challenges in managing large volumes of student and staff information, handling admissions, and tracking performance. Manual systems can lead to errors, data redundancy, and inefficiency.
 - This project introduces an **Educational Management System** built on the **ServiceNow platform**, leveraging its powerful automation and workflow management capabilities. The system is designed to ensure seamless data integration, real-time updates, and secure access to educational records. Additionally, the integration of **TensorFlow** enables data-driven insights and predictive analytics to support better decision-making and performance analysis.
-

3. Project Scope

The scope of this project includes the development and deployment of a digital management system for educational organizations with the following modules:

- **Student Information Management:** Securely store, update, and access student details.
- **Teacher Management:** Maintain records of faculty members, schedules, and teaching activities.
- **Admission Management:** Simplify and automate the admission and enrollment process.
- **Academic Progress Monitoring:** Track student grades, attendance, and performance trends.
- **Notifications & Reporting:** Automate alerts and generate performance reports.
- **Analytics Module (TensorFlow):** Implement AI-based analysis for predicting student outcomes and optimizing learning strategies.

The project will be implemented within the ServiceNow environment, ensuring scalability, reliability, and role-based access control.

4. Tools and Technologies Used

Category	Tools / Technologies
Platform	ServiceNow
ServiceNow	ServiceNow Cloud platform for workflow automation
Update Sets	To capture and move customizations
Tables & Fields	Used to store expense records
Forms & Lists	For user interaction and record display
Reports Module	For data visualization and analytics
Other Tools	ServiceNow Flow Designer, Service Catalog, Workflow Editor

5. System Requirements

Hardware Requirements

- Processor: Intel Core i5 or above
- RAM: Minimum 8 GB
- Hard Disk: 500 GB or higher
- Network: Stable Internet connection

Software Requirements

- Operating System: Windows 10 / Linux / macOS
 - Web Browser: Google Chrome / Mozilla Firefox
 - ServiceNow Developer Instance
 - ServiceNow Account & Studio
-

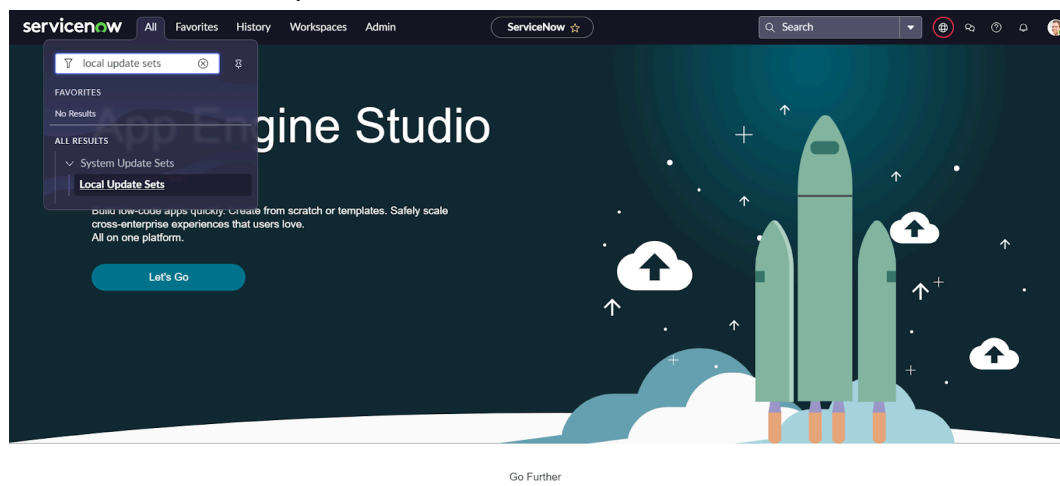
6. Project Implementation Steps

Step 1: Setting up ServiceNow Instance

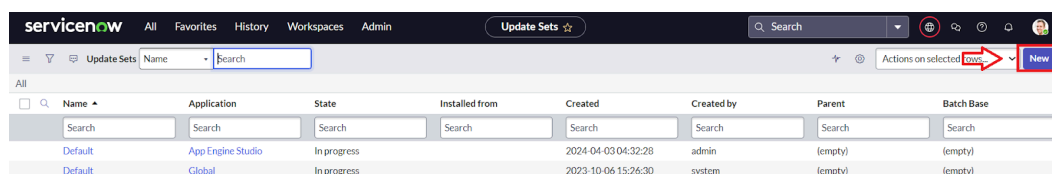
1. Sign up for a developer account on the ServiceNow Developer site ["https://developer.servicenow.com"](https://developer.servicenow.com).
2. Once logged in, navigate to the "Personal Developer Instance" section.
3. Click on "Request Instance" to create a new ServiceNow instance.
4. Fill out the required information and submit the request.
5. .You'll receive an email with the instance details once it's ready.
6. Log in to your ServiceNow instance using the provided credentials.
7. Now you will navigate to the ServiceNow.

Step 2: Creating a Update Set

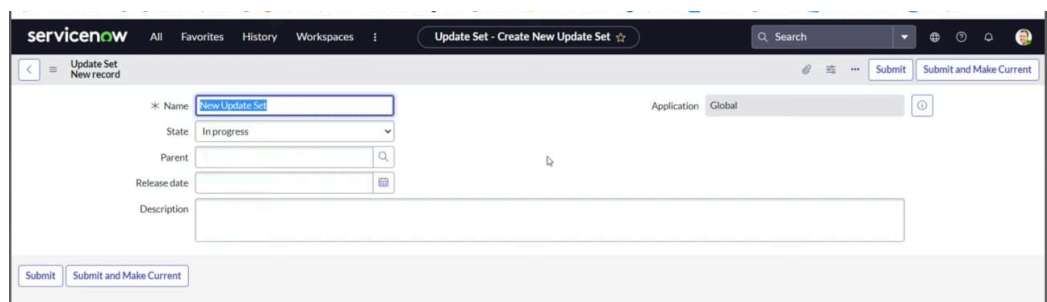
1. Click on All >> Local update sets



2. Click on new



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



Step 3: Creating Salesforce Table

1. All >> Tables.
2. Click on new
3. Enter the Label(Anything you want): Salesforce >> Click on Name it will Automatically generate Api name.
4. Create columns as given below, Double Click on Column label and Enter the Column labels and click on the tick mark >> Give Type as given .
5. For “Admin Number” Give Display as True and right click on the toggle bar on top >> save.
6. Click on controls >> Enable Extensible.
7. Click on “Admin Number” column, In Related Links Click on Advanced View >> Default View (Enable Use dynamic default) >> select Get Next Padded Number in Dynamic default value >> Update .
8. Click on “Grade” Column >> Click on Choices and give Label, Value and Sequence as given below.

Step 4: Creating Admission Table

- 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 'Table - New Record' form in ServiceNow. The 'Label' is 'Admission' and the 'Name' is 'u_admission'. The 'Extends table' is set to 'Salesforce'. The 'Application' is 'Global'. The 'Create module' checkbox is checked. The 'Add module to menu' dropdown is set to 'Salesforce'. Below this, the 'Columns' tab is active, showing a table of columns to be created.

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

- Create choice for Admin Status as:

Dictionary Entry Admin Status

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

Choices

Label	Value	Language	Sequence	Inactive	Updated
New	New	en	1	false	2024-04-02 21:10:25
Join in progress	In progress	en	2	false	2024-04-02 21:11:03
Joined	Joined	en	3	false	2024-04-02 21:11:26
Rejected	Rejected	en	4	false	2024-04-02 21:12:00
Closed	Closed	en	5	false	2024-04-02 21:13:09
Rejoined	Rejoined	en	6	false	2024-04-02 21:13:08
Cancelled	Cancelled	en	7	false	2024-04-02 21:13:27

- Create choice for Pincode as:

Dictionary Entry Pincode

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

Choices

Label	Value	Language	Sequence	Inactive	Updated
509358	509358	en	1	false	2024-04-02 21:15:19
500079	500079	en	2	false	2024-04-02 21:15:46
500081	500081	en	3	false	2024-04-02 21:16:05

- Create choice for Purpose of Join as:

Dictionary Entry Purpose of Join

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

Choices

Label	Value	Language	Sequence	Inactive	Updated
Tuition	Tuition	en	1	false	2024-04-02 21:17:09
Coaching	Coaching	en	2	false	2024-04-02 21:17:31
Teacher	Teacher	en	3	false	2024-04-02 21:17:53

- Create choice for School as:

Dictionary Entry School

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

Choices

Label	Value	Language	Sequence	Inactive	Updated
Stanley	Stanley	en	1	false	2024-04-02 21:19:14
Nareesh It	Nareesh It	en	2	false	2024-04-02 21:19:35

- Create choice for School Area as:

Dictionary Entry School Area

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

Choices

Label	Value	Language	Sequence	Inactive	Updated
Near Market	Near Market	en	1	false	2024-04-02 21:20:53
Near Bus Stand	Near Bus Stand	en	2	false	2024-04-02 21:21:24

Step 5: Creating Student Progress Table

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

Field	Type	Language	Sequence	Inactive	Updated
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	

Step 6: Configuring Table form for Student Progress Table

- In the Student Progress Table Page , Click on Layout form .

	Type	Value	
Telugu	String	(empty)	40
Total	String	(empty)	40
Insert a new row...			

Update Delete Delete All Records

Related Links
[Design Form](#)
[Layout Form](#)
[Layout List](#)
[Show Form](#)
[Show List](#)
[Show Schema Map](#)
[Add to Service Catalog](#)
[Run Point Scan](#)
[Explore REST API](#)

- Click on Admission Number [+] .

Configuring Table form

Available

- Admission Number (+)
- Created
- Created by
- Updated
- Updated by
- Updates
- [- begin_split -]
- [- split -]
- [- end_split -]
- * Annotation
- * Chart
- Activities (filtered)
- Contextual Search Results
- Ratings
- Attachments
- Goal relationships

Selected

- [- begin_split -]
- Admission Number
- Hindi
- English
- Telugu
- Science
- Total
- Average
- Social
- Maths
- [- end_split -]

Form view and section: View name: Default view

Create new field: Name:

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

Configuring Table form

Available

- Admission Number (+)
- Created
- Created by
- Updated
- Updated by
- Updates
- [- begin_split -]
- [- split -]
- [- end_split -]
- * Annotation
- * Chart
- Activities (filtered)
- Contextual Search Results
- Ratings
- Attachments
- Goal relationships

Selected

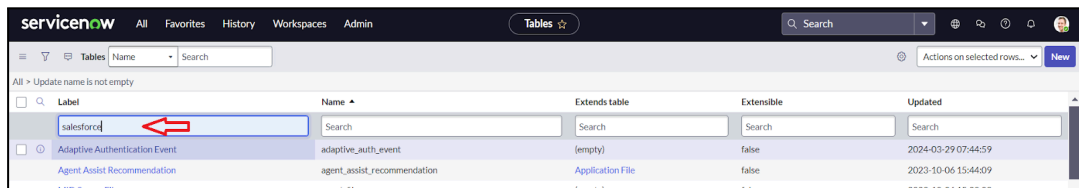
- Admission Number
- Hindi
- English
- Telugu
- Science
- [- split -]
- Total
- Average
- Social
- Maths
- [- end_split -]
- Admission Number.Admin Date
- Admission Number.Student Name
- Admission Number.Father Name
- Admission Number.Mother Name
- Admission Number.Father Cell
- Admission Number.Mother Cell

Form view and section: View name: Default view

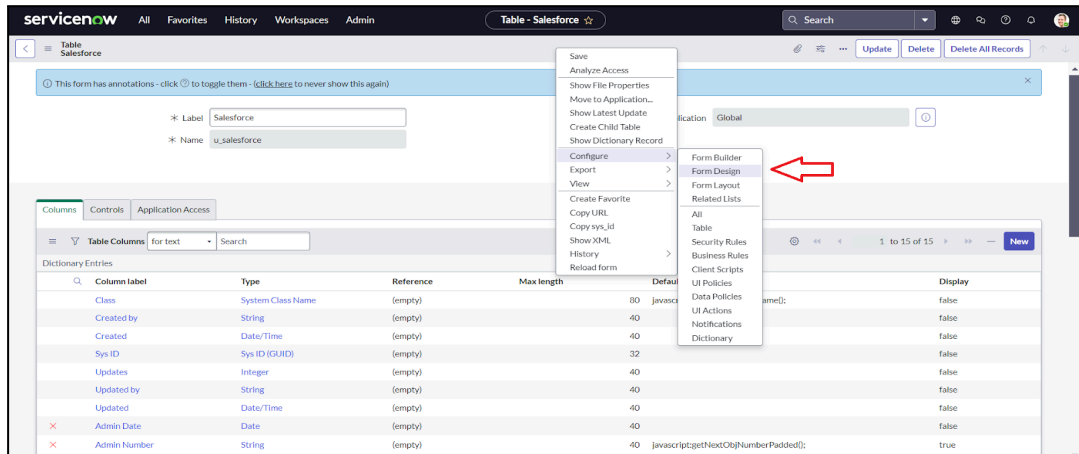
Create new field: Name:

Step 7: Creating Form Design for Salesforce Table

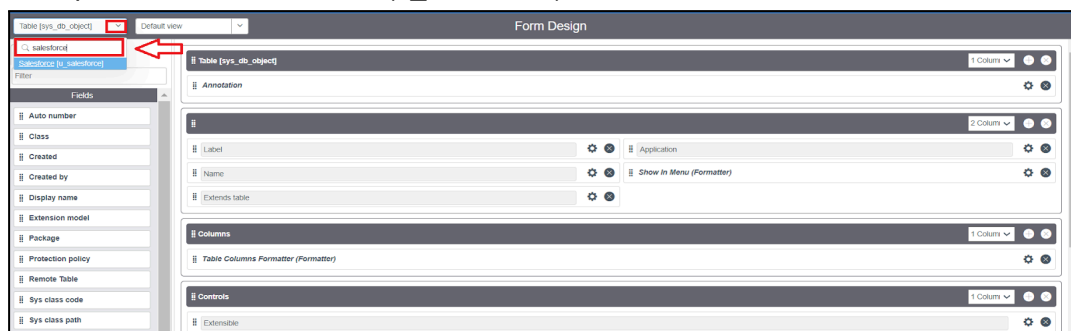
1. All >> System Definition >> Tables .
2. In Label Search for Salesforce and open .



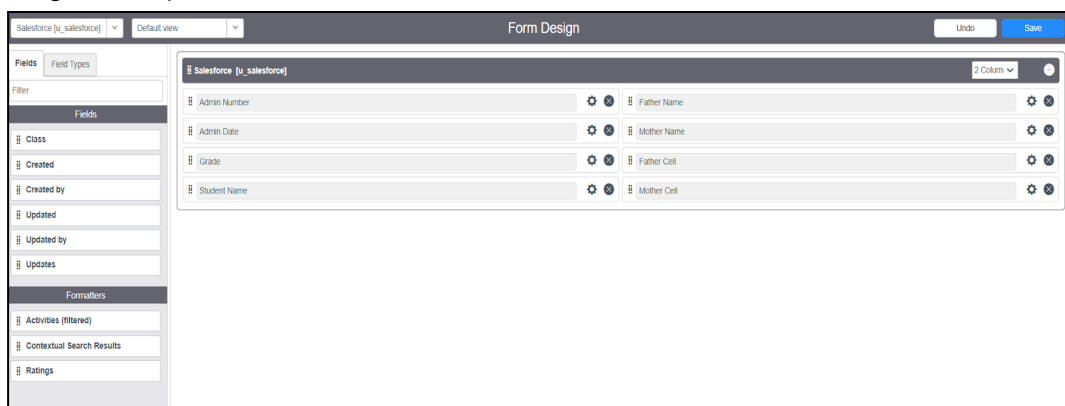
3. Right Click on top Toggle >> Configure >> Form Design.



4. In drop down select Salesforce(u_salesforce).



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



6. Save

Step 8: Creating Form Design for Admission Table

- Follow the same Step s as Activity1,Configure the fields as below and Save.

The screenshot shows the 'Form Design' interface for the 'Admission' table. The sidebar on the left contains 'Fields' and 'Formatters' sections. The main area displays the form structure with sections for 'Admission Details', 'School Details', and 'Address'. Fields include Admission Number, Class, Created, Created by, Updated, Updated by, Updates, Admission Date, Grade, Fee, Father Cell, Mother Cell, Admission Status, Comments, School Area, School, Phone, Area, City, and District.

Step 9: Creating Form Design for Student progress Table

- Follow the same steps as Activity1,Configure the fields as below and Save.

The screenshot shows the 'Form Design' interface for the 'Student Progress' table. The sidebar on the left contains 'Fields' and 'Formatters' sections. The main area displays the form structure with sections for 'New Section' and 'Student Progress'. Fields include Admission Number, Admission Number Grade, Admission Number Father Name, Admission Number Mother Name, Admission Number Student Name, Admission Number Father Cell, Admission Number Mother Cell, Telugu, Hindi, English, Maths, Science, Total, Percentage, and Result.

Step 10: Creating Number Maintenance for Admin Number

- All >> Number Maintenance >> New

Class Name	Reference	Max length	Default value	Display
Created by	String	(empty)	80	false
Created	Date/Time	(empty)	40	false
Sys ID	Sys ID (GUID)	(empty)	32	false
Updates	Integer	(empty)	40	false
Updated by	String	(empty)	40	false

- Fill the details >> Submit.

Table: Salesforce
Prefix: SAL
Number: 1,000
Application: Global
Number of digits: 7

Update Delete

Step 11: Creating Process Flow for Admission Table

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

Table: Admission [u_admission]
Name: New
Application: Global
Label: New
Order: 1
Active: ☒
Condition: Admin Status is New
Description:

Update Delete

- Right Click on 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.
- Order should be New >> InProgress >> Joined >> Rejected >> Rejoined >> Closed >> Cancelled.

Step 12: Creating “Auto populate” Client Scripts for Admission Table

- All >> Client Scripts >> New.
- 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 a = g_form.getReference('u_admission_number');
g_form.setValue('u_admin_date',a.u_admin_date);
g_form.setValue('u_grade',a.u_grade);
g_form.setValue('u_student_name',a.u_student_name);
g_form.setValue('u_father_name',a.u_father_name);
```

```

g_form.setValue('u_mother_name',a.u_mother_name);
g_form.setValue('u_father_cell',a.u_father_cell);
g_form.setValue('u_mother_cell',a.u_mother_cell);

g_form.setDisabled('u_admin_date',a.u_admin_date);
g_form.setDisabled('u_grade',a.u_grade);
g_form.setDisabled('u_student_name',a.u_student_name);
g_form.setDisabled('u_father_name',a.u_father_name);
g_form.setDisabled('u_mother_name',a.u_mother_name);
g_form.setDisabled('u_father_cell',a.u_father_cell);
g_form.setDisabled('u_mother_cell',a.u_mother_cell);
}

```

Note: Make sure the Field names should be the same as you created .

Step 13: Creating “Pincode Update” Client Scripts for Admission Table

- Fill the Details as given.

The screenshot shows the SAP Client Script configuration window for a script named 'Pincode Update'. The configuration details are as follows:

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

The Description and Messages fields are empty. The Script editor contains 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');

```

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

```

function onChange(control, oldValue, newValue, isLoading, isTemplate) {
    if (isLoading || newValue === '') {
        return;
    }
    var a = g_form.getValue('u_pincode');
    if(a == '509358')
    {
        g_form.setValue('u_mandal', 'kadthal');
        g_form.setValue('u_city', 'kadthal');
        g_form.setValue('u_district', 'RangaReddy');
    }
}

```

```

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

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

//Type appropriate comment here, and begin script below

}

```

Step 14: Creating “Disable Fields” Client Scripts for Student progress Table

- 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);
}

```

Step 15: Creating “Total Update” Client Scripts for Student progress Table

- Fill the Details as given.

The screenshot shows the ServiceNow interface for configuring a Client Script. The script is named 'Total Update' and is associated with the 'Student Progress [u_student_progress]' table. It is set to trigger on the 'Social' field when it changes. The script is active and is a global client script. The script code is as follows:

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

    //Type appropriate comment here, and begin script below
    if (newValue){
        var a = parseInt(g_form.getValue('u_telugu'));
        var b = parseInt(g_form.getValue('u_hindi'));
        var c = parseInt(g_form.getValue('u_english'));
        var d = parseInt(g_form.getValue('u_maths'));
        var e = parseInt(g_form.getValue('u_science'));
        var f = parseInt(g_form.getValue('u_social'));
        var Total = parseInt(a+b+c+d+e+f);
        g_form.setValue('u_total', Total);
    }
}
```

- 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_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);
}
}
```

Step 16: Creating “Result” Client Scripts for Student progress Table

- Fill the Details as given.

The screenshot shows the 'Client Script' editor for the 'Result' field. The configuration details are as follows:

- Name: Result
- Table: Student Progress [u_student_progress]
- UI Type: All
- Type: onChange
- Field name: Percentage
- Application: Global
- Active: ☒
- Inherited: ☐
- Global: ☒

The script editor contains the following JavaScript code:

```

1 function onChange(control, oldValue, newValue, isLoading, isTemplate) {
2   if (isLoading || newValue === '') {
3     return;
4   }
5
6   //Type appropriate comment here, and begin script below
7   if(newValue) {
8     var a = parseInt(g_form.getValue('u_percentage')); // Convert the value to an integer for comparison
9     if(a >= 0 && a <= 59){
10      g_form.setValue('u_result','Fail');
11    } else if(a >= 60 && a <= 100) {
12      g_form.setValue('u_result','Pass');
13    } else {

```

- 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');
```

```

    }
  }
}
```

Step 17: Creating “Percentage” Client Scripts for Student progress Table

- Fill the Details as given.

The screenshot shows the ServiceNow Client Script editor interface. The 'Name' field is set to 'Percentage'. The 'Table' dropdown is set to 'Student Progress [u_student_progress]'. The 'UI Type' dropdown is set to 'All'. The 'Type' dropdown is set to 'onChange'. The 'Field name' dropdown is set to 'Total'. The 'Application' dropdown is set to 'Global'. The 'Active' checkbox is checked. The 'Inherited' checkbox is unchecked. The 'Global' checkbox is checked. The 'Description' field is empty. The 'Messages' field is empty. The 'Script' field contains the following code:

```
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   var Total = g_form.getValue('u_total');
7   var Percentage = (Total/600)*100;
8   g_form.setValue('u_percentage', Percentage+'%');
9 }
10
```

The 'Isolate script' checkbox is checked. The 'Update' and 'Delete' buttons are visible at the bottom.

- 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+'%');
}
```

7. Conclusion:

- The **Educational Organisation Using ServiceNow** project effectively automates and streamlines key administrative tasks in educational institutions. By integrating **ServiceNow** for workflow management and **TensorFlow** for data analysis, the system ensures efficient handling of student and teacher data, smooth admissions, and improved decision-making. Overall, it provides a reliable and scalable solution for modern educational management.

8. Result:

The image displays three sequential screenshots of the ServiceNow user interface, illustrating the process of creating new records in different modules.

Screenshot 1: Educational Organisation Form

The top navigation bar shows the 'Update Set - Create...' button. The breadcrumb trail indicates 'Update Set' > 'New record'. The form fields include:

- Name: Educational Organisation
- State: In progress
- Parent: (empty)
- Release date: (empty)
- Description: (empty)
- Application: Global

Buttons at the bottom include 'Submit' and 'Submit and Make Current'.

Screenshot 2: Salesforce Form

The top navigation bar shows the 'Salesforce - Create ...' button. The breadcrumb trail indicates 'Salesforce' > 'New record'. The form fields include:

- Admin Number: SAL0001002
- Admin Date: (empty)
- Grade: -- None --
- Student Name: (empty)
- Father Name: (empty)
- Mother Name: (empty)
- Father Cell: (empty)
- Mother Cell: (empty)

A 'Submit' button is located at the bottom.

Screenshot 3: Student Progress Form

The top navigation bar shows the 'Student Progress - ...' button. The breadcrumb trail indicates 'Student Progress' > 'New record'. The form fields include:

- Maths: (empty)
- Science: (empty)
- English: (empty)
- Result: (empty)
- Percentage: (empty)
- Hindi: (empty)
- Social: (empty)
- Total: (empty)
- Telugu: (empty)
- Admission Number: (empty)

A 'Submit' button is located at the bottom.