

# Educational Organisation Using ServiceNow

**TEAM ID: NM2025TMID18283**

## **TEAM MEMBERS:**

**Team Leader:** Samreen Taj N

**Team Member:** Hemalatha D

**Team Member:** Keerthana Bai S

**Team Member:** Poongavanam S

## **PROBLEM STATEMENT:**

In many educational institutions, student admissions, records, and academic progress are still managed manually. This traditional approach creates several issues such as delays in the admission process, errors in data handling, and difficulties in generating unique admission numbers. The absence of a centralized system leads to scattered student and parent information, making it hard to access and maintain. Furthermore, tracking student performance and progress becomes inefficient, while the lack of automation increases staff workload and reduces accuracy. To address these challenges, there is a need for an automated system that streamlines admissions, ensures data consistency, and simplifies overall management.

## **OBJECTIVE:**

The objectives of the project “Educational Organisation Using ServiceNow” are:

- To automate the **student admission process** and reduce manual errors.
- To generate and maintain **unique admission numbers** using number maintenance.
- To create a **centralized system** for storing student and parent details.
- To design **user-friendly forms and layouts** for easy data entry and management.
- To implement **process flows** for tracking admission stages effectively.
- To add **client scripts** (e.g., Pincode Update) for dynamic updates and accuracy.
- To provide a system that improves **efficiency, accuracy, and transparency** in educational management.

## SKILLS:

During the development of the project “**Educational Organisation Using ServiceNow**”, the following skills were applied and enhanced:

- **ServiceNow Development Skills** – Creating update sets, tables, forms, workflows, and client scripts.
- **Form Design & Layout Skills** – Customizing form structures for clear and user-friendly data entry.
- **Workflow and Process Automation** – Designing process flows for admissions and automating record management.
- **Scripting Skills** – Writing client scripts such as Pincode Update for dynamic form behavior.
- **Database Management** – Creating and managing custom tables for student, admission, and progress records.
- **Problem-Solving** – Identifying challenges in manual systems and designing automated solutions.
- **Team Collaboration** – Working as a team to design, configure, and test the system step by step.

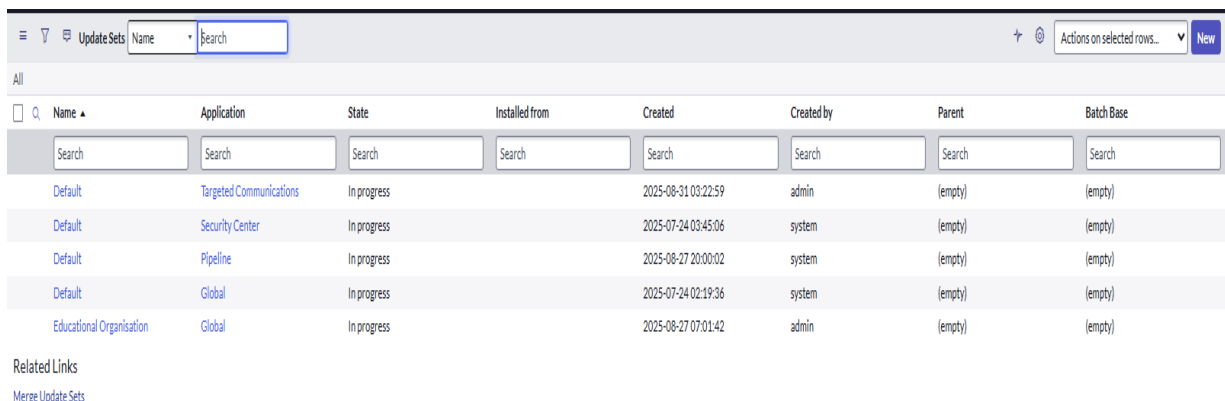
## TASK INITIATION

### Milestone 1: Update Set

#### Activity 1: Create Update Set

##### Steps:

1. Navigate to **All** → **Local Update Sets**.
2. Click on **New**.
3. Enter the details:
  - **Name:** Educational Organisation
4. Click **Submit** and then **Make Current** to set it as the active update set.



Update Sets								Search	Actions on selected rows...	New
All	Name	Application	State	Installed from	Created	Created by	Parent	Batch Base		
<input type="checkbox"/>	Search	Search	Search	Search	Search	Search	Search	Search		
	Default	Targeted Communications	In progress		2025-08-31 03:22:59	admin	(empty)	(empty)		
	Default	Security Center	In progress		2025-07-24 03:45:06	system	(empty)	(empty)		
	Default	Pipeline	In progress		2025-08-27 20:00:02	system	(empty)	(empty)		
	Default	Global	In progress		2025-07-24 02:19:36	system	(empty)	(empty)		
	Educational Organisation	Global	In progress		2025-08-27 07:01:42	admin	(empty)	(empty)		
Related Links										
<a href="#">Merge Update Sets</a>										

## Milestone 2: Creating a Table

### Activity 1: Creating Salesforce Table.

#### Steps:

1. Navigate to **All** → **Tables**.
2. Click on **New**.
3. Enter the table details:
  - **Label:** Salesforce (API Name will be auto-generated).
4. Add the following fields/columns to the table:

Table - Salesforce

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: Salesforce  
\* Name: u\_salesforce

Application: Global

Columns Controls Application Access

Table Columns Column label Search 1 to 15 of 15 New

Column label	Type	Reference	Max length	Default value	Display
Admin Date	Date	(empty)	40		false
Admin Number	String	(empty)	40	javascript:getNextObj/Number Padded()	true
Class	System Class Name	(empty)	80	javascript:current.getTable/Name()	false
Created	Date/Time	(empty)	40		false
Created by	String	(empty)	40		false
Father Call	String	(empty)	40		false
Father Name	String	(empty)	40		false
Grade	Choice	(empty)	40		false
Mother Call	String	(empty)	40		false
Mother Name	String	(empty)	40		false
Student Name	String	(empty)	40		false
Sys ID	Sys ID (GUID)	(empty)	32		false
Updated	Date/Time	(empty)	40		false
Updated by	String	(empty)	40		false
Updates	Integer	(empty)	40		false

Insert a new row...

Delete Update Delete All Records

#### 5. For Admin Number:

1. Open the field, go to **Advanced View** → **Default View**.
2. Enable **Use dynamic default**.
3. Select **Get Next Padded Number** in Dynamic Default Value.
4. Update the field.

Dictionary Entry AdminNumber View: Advanced\*

A dictionary entry manages how ServiceNow stores data in tables and fields (columns). For new dictionary entries, select a Table and the field Type of the new column. Also enter a column label, which becomes the field label, and the column name. If necessary, set a Max length for text String type fields, make the field Mandatory to save a record, and make the field a Display Value for reference fields so it appears on records that reference this table. [More Info](#)

\* Table: Salesforce [u\_salesforce]  
\* Type: String  
\* Column label: AdminNumber  
\* Column name: u\_admin\_number  
\* Max length: 40

Application: Global  
Active: ☒  
Function field: ☐  
Read only: ☐  
Mandatory: ☐  
Display: ☒

Attributes

Choice List Specification Calculated Value Default Value

The Default value specifies what value the field has when first displayed.

Use dynamic default: ☒  
Dynamic default value: Get Next Padded Number

Delete Column Update

## 6. For Grade:

1. Open the field, go to **Choices**.
2. Define **Label**, **Value**, and **Sequence** for grade options.

Label	Value	Language	Sequence	Inactive	Updated
III	3rd	en	6	false	2025-08-29 08:35:40
V	5th	en	8	false	2025-08-29 08:35:51
Nursery	Nursery	en	2	false	2025-08-29 08:29:57
VI	6th	en	9	false	2025-08-29 08:35:57
I	1st	en	4	false	2025-08-29 08:31:08
UKG	UKG	en	3	false	2025-08-29 08:44:11
IV	4th	en	7	false	2025-08-29 08:35:44
VIII	8th	en	11	false	2025-08-29 08:36:07
IX	9th	en	12	false	2025-08-29 08:36:11
Prept	Prept	en	1	false	2025-08-29 08:27:01
X	10th	en	13	false	2025-08-29 08:36:23
II	2nd	en	5	false	2025-08-29 08:31:33
VII	7th	en	10	false	2025-08-29 08:36:03

## Activity 2: Creating Admission Table

### Steps:

1. Navigate to **All** → **Tables**.
2. Click on **New**.
3. Enter the table details:
  - **Label:** Admission
  - **Extends Table:** Salesforce (to inherit existing student details).
  - **Add Module to Menu:** Salesforce (to make the table accessible in the application menu).
4. Add the required fields/columns for admission details.

Column label	Type	Reference	Max length	Default value	Display
Admin Date	Date	(empty)	40		false
Admin Status	Choice	(empty)	40		false
Admin Number	String	(empty)	40	javascript.getNextCb(NumberOfPadded)	true
Admission Number	Reference	Salesforce	32		false
Area	String	(empty)	40		false
City	String	(empty)	40		false
Class	System Class Name	(empty)	80	javascript.current.getTableName()	false
Comments	String (Full UTF-8)	(empty)	255		false
Created	Date/Time	(empty)	40		false
Created by	String	(empty)	40		false
District	String	(empty)	40		false
Father Cell	String	(empty)	40		false
Father Name	String	(empty)	40		false
Fee	Price	(empty)	20		false
Grade	Choice	(empty)	40		false
House No	String	(empty)	40		false
Mandal	String	(empty)	40		false
Mother Cell	String	(empty)	40		false
Mother Name	String	(empty)	40		false
Pincode	Choice	(empty)	40		false

5. **For Admin Status:** Create a **choice field** with predefined values to represent different admission statuses (e.g., Pending, Approved, Rejected).

Access Controls

Choices (7)

Attributes

Labels (1)

≡

▽

Label ▾

Search

⊙

—

Actions on selected rows... ▾

New

Choices

<input type="checkbox"/>	<div>🔍</div> Label	Value	Language	Sequence	Inactive	Updated
<input type="checkbox"/>	<div>⊙</div> Rejoined	Rejoined	en	5	false	2025-08-31 03:29:21
	Joined	Joined	en	3	false	2025-08-29 09:53:45
	Closed	Closed	en	6	false	2025-08-31 03:29:27
	Join in progress	In Progress	en	2	false	2025-08-29 09:53:38
	Rejected	Rejected	en	4	false	2025-08-29 09:53:47
	New	New	en	1	false	2025-08-29 09:49:49
	Cancelled	Cancelled	en	7	false	2025-08-29 09:53:58
<div>+</div>	Insert a new row...					

⏪

⏩

1 to 7 of 7

⏪

⏩

6. **For Pincode:** Create a **choice field** with valid pincode options to standardize address entry.

Access Controls

Choices (3)

Attributes

Labels (1)

≡

▼

Label

Search

⊙

—

Actions on selected rows...

New

Choices

☐

🔍

Label

Value

Language

Sequence

Inactive

Updated

500081

500081

en

3

false

2025-08-29 10:14:33

500079

500079

en

2

false

2025-08-29 10:14:30

509358

509358

en

1

false

2025-08-29 10:14:23

+

Insert a new row...

⏪

⏩

1 to 3 of 3

⏪

⏩

7. **For Purpose of Join:**

Access Controls

Choices (3)

Attributes

Labels (1)

≡

▼

Label

Search

⊙

—

Actions on selected rows...

New

Choices

☐

🔍

Label	Value	Language	Sequence	Inactive	Updated
Teacher	Teacher	en	3	false	2025-08-29 10:17:49
Coaching	Coaching	en	2	false	2025-08-29 10:17:46
Tuition	Tuition	en	1	false	2025-08-29 10:17:43

+

Insert a new row...

⏪

◀

1 to 3 of 3

▶

⏩

8. **For School:** Create a **choice field** with a list of schools to select from.

Access Controls

Choices (2)

Attributes

Labels (1)

≡

▼

Label

▼

Search

⊙

—

Actions on selected rows...

▼

New

Choices

☐

🔍

Label

Value

Language

Sequence

Inactive

Updated

[Naresh It](#)

Naresh It

en

2 false

2025-08-29 10:20:08

[Stanley](#)

Stanley

en

1 false

2025-08-29 10:19:42

+

Insert a new row...

⏪

⏩

1 to 2 of 2

⏪

⏩

9. **For School Area:** Create a **choice field** with area names for accurate mapping of school locations.

Access Controls

Choices (2)

Attributes

Labels (1)

≡

▼

Label

Search

⊙

—

Actions on selected rows...

New

Choices

<input type="checkbox"/>	<div><div>🔍</div><div>Label</div></div>	Value	Language	Sequence	Inactive	Updated
	<a href="#">Near Bus Stand</a>	Near Bus Stand	en		2 false	2025-08-29 10:22:10
	<a href="#">Near Market</a>	Near Market	en		1 false	2025-08-29 10:22:08
<div>+</div>	Insert a new row...					

⏪

⏴

1

to 2 of 2

⏵

⏩

## Activity 3: Creating Student Progress Table

### Steps:

1. Navigate to **All** → **Tables**.
2. Click on **New**.
3. Enter the table details:
  - **Label:** Student
4. **Add Module to Menu:** Salesforce (so the table appears in the Salesforce application menu).
5. Add the required fields/columns to track student progress.

Column label	Type	Reference	Max length	Default value	Display
Admission Number	Reference	Admission	32		false
Class	System Class Name	(empty)	80	javascript:current.getTableNamed();	false
Created	Date/Time	(empty)	40		false
Created by	String	(empty)	40		false
English	String	(empty)	40		false
Father Cell	String	(empty)	40		false
Father Name	String	(empty)	40		false
Grade	Choice	(empty)	40		false
Hindi	String	(empty)	40		false
Maths	String	(empty)	40		false
Mother Cell	String	(empty)	40		false
Mother Name	String	(empty)	40		false
Percentage	String	(empty)	40		false
Result	String	(empty)	40		false
Science	String	(empty)	40		false
Social	String	(empty)	40		false
Student Name	String	(empty)	40		false
Sys ID	Sys ID (GUID)	(empty)	32		false
Telugu	String	(empty)	40		false
Total	String	(empty)	40		false
+ Insert a new row...					

## Milestone 3: Form Layout

### Activity 1: Configuring Table form for Student Progress Table

#### Steps:

- Open the **Student Progress Table**.
- Click on **Layout Form** to configure the table form view.
- From the **Available Fields** list, locate **Admission Number** [+].
- Expand the field and select the required sub-fields of **Admission Number**.
- Move the selected fields to the **Selected Fields** side.
- Click **Save** to update the form layout

## Milestone 4: Form Design

### Activity 1: Creating Form Design for Salesforce Table

#### Steps:

1. Navigate to **All** → **System Definition** → **Tables**.
2. Search for the table **Salesforce** and open it.
3. Right-click on the top header (toggle) and select **Configure** → **Form Design**.
4. In the dropdown menu, choose **Salesforce(u\_salesforce)**.
5. Drag and drop the required fields into the form layout on the left side.
6. Click **Save** to apply the changes.

### Activity 2: Creating Form Design for Admission Table

#### Step:

1. Navigate to **All** → **System Definition** → **Tables**.
2. Search for the table **Admission** and open it.
3. Right-click on the top header (toggle) and select **Configure** → **Form Design**.
4. In the dropdown menu, choose **Admission(u\_admission)**
5. Drag and drop the required fields into the form layout on the left side.
6. Click **Save** to apply the changes.

### Activity 3: Creating Form Design for Student progress Table

#### Step:

1. Navigate to **All** → **System Definition** → **Tables**.
2. Search for the table **Students** and open it.
3. Right-click on the top header (toggle) and select **Configure** → **Form Design**.
4. In the dropdown menu, choose **Students(u\_students)**
5. Drag and drop the required fields into the form layout on the left side.
6. Click **Save** to apply the changes.

### Milestone 5: Number Maintenance



## Activity 1: Creating Number Maintenance for Admin Number

### Steps:

1. Navigate to **All** → **Number Maintenance** → **New**.
2. Fill in the required details for the **Admin Number** configuration.
3. Click **Submit** to save the record.

The screenshot shows a web form titled "Number Maintenance" for the "Admin Number" configuration. The form includes the following fields:

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

At the bottom of the form, there are "Update" and "Delete" buttons.

## Milestone 6: Process Flow

### Activity 1: Creating Process Flow for Admission Table

### Steps:

1. Navigate to **All** → **Process Flow** → **New**.
2. Fill in the required details for the admission process flow.
3. Right-click on the header (toggle) and click **Save**.
4. Replace the **Name** and **Label** fields as needed, then click **Insert and Stay**.
5. Continue updating the **Name** and **Label** for each process stage, clicking **Insert and Stay** after each step.

The screenshot shows a web form titled "Process Flow" for the "Admission Table" configuration. The form includes the following fields:

- \* Table:** Admission [u\_admission]
- \* Name:** new
- Application:** Global
- \* Label:** New
- Order:** 1
- Active:** ☒
- Condition:** Admin Status is New

At the bottom of the form, there are "Add Filter Condition" and "Add \*OR\* Clause" buttons. The condition is currently set to "Admin Status is New".

Flow Formatter

Join\_in\_Progress

Update

Delete

↑

↓

\* Table

Admission[u\_admission]

\* Name

Join\_in\_Progress

Application

Global

0

\* Label

Join in progress

Order

2

Active

☒

Condition

Add Filter Condition

Add "OR" Clause

Admin Status

Is

Join in progress

AND

OR

X

Flow Formatter

joined

Update

Delete

↑

↓

\* Table

Admission[u\_admission]

\* Name

joined

Application

Global

0

\* Label

Joined

Order

3

Active

☒

Condition

Add Filter Condition

Add "OR" Clause

Admin Status

Is

Joined

AND

OR

X

## Milestone 7: Client Script

### Activity 1: Auto Populate Script – Admission Table.

#### Steps:

1. Navigate to **All** → **Client Scripts** → **New**.
2. Fill in the required details, such as **Name**, **Table**, **UI Type**, and **Script**.
3. Click **Submit** to save the client script.

Client Script

Auto populate

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.black.client.global" to false.

Name

Auto populate

Table

Admission[u\_admission]

UI Type

Mobile / Service Portal

Type

onChange

Field name

AdmissionNumber

Application

Global

0

Active

☒

Inherited

☐

Global

☒

Description

Messages

Script

```

1 //Type appropriate comment here, and begin script below
2
3 var = $form.getReference("u_admission_number");
4
5 $form.setValue("u_admission_date",u_admission_date);
6
7 $form.setValue("u_grade",u_grade);
8
9 $form.setValue("u_student_name",u_student_name);
10
11 $form.setValue("u_father_name",u_father_name);
12
13 $form.setValue("u_mother_name",u_mother_name);
14
15 $form.setValue("u_father_cell",u_father_cell);
16
17 $form.setValue("u_mother_cell",u_mother_cell);
18
19 $form.setValue("u_admission_date",u_admission_date);
20
21

```

Isolate script

☒

Update

Delete

## Activity 2: Pincode Update Script – Admission Table.

### Steps:

1. Navigate to **All → Client Scripts → New**.
2. Fill in the required details, such as **Name**, **Table**, **UI Type**, and **Script**.
3. Click **Submit** to save the client script.

The screenshot shows the 'Client Script' configuration form for a script named 'Pincode Update'. The form is titled 'Pincode Update' and has a subtitle 'Pincode Update'. It includes a warning message: '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.' The form fields are: Name (Pincode Update), Table (Admission [u\_admission]), UI Type (Desktop), Type (onChange), Field name (Pincode), Application (Global), Active (checked), Inherited (unchecked), and Global (checked). There is a Description field and a Messages field. The Script field contains a JavaScript function that sets values for 'u\_name', 'u\_city', 'u\_district', 'u\_email', 'u\_password', 'u\_confirm\_password', 'u\_phone', 'u\_address', 'u\_state', 'u\_zip', 'u\_pincode', 'u\_gender', 'u\_age', 'u\_dob', 'u\_marital\_status', 'u\_education', 'u\_occupation', 'u\_income', 'u\_expense', 'u\_savings', 'u\_loans', 'u\_insurance', 'u\_health', 'u\_wellness', 'u\_mindfulness', 'u\_spirituality', 'u\_religion', 'u\_beliefs', 'u\_values', 'u\_traits', 'u\_talents', 'u\_skills', 'u\_knowledge', 'u\_wisdom', 'u\_experience', 'u\_learning', 'u\_growth', 'u\_development', 'u\_transformation', 'u\_evolution', 'u\_enlightenment', 'u\_nirvana', 'u\_moksha', 'u\_arhantship', 'u\_buddhahood', 'u\_enlightenment', 'u\_nirvana', 'u\_moksha', 'u\_arhantship', 'u\_buddhahood'. The 'isolate script' checkbox is checked. At the bottom are 'Update' and 'Delete' buttons.

## Activity 3: Disable Fields Script – Student Progress Table.

### Steps:

1. Navigate to **All → Client Scripts → New**.
2. Fill in the required details, such as **Name**, **Table**, **UI Type**, and **Script**.
3. Click **Submit** to save the client script.

The screenshot shows the 'Client Script' configuration form for a script named 'Disable Fields'. The form is titled 'Disable Fields' and has a subtitle 'Disable Fields'. It includes a warning message: '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.' The form fields are: Name (Disable Fields), Table (Student [u\_student]), UI Type (All), Type (onLoad), Application (Global), Active (checked), Inherited (unchecked), and Global (checked). There is a Description field and a Messages field. The Script field contains a JavaScript function that sets 'disabled' to true for 'u\_total', 'u\_percentage', and 'u\_result'. The 'isolate script' checkbox is checked. At the bottom are 'Update' and 'Delete' buttons.

## Activity 4: Total Update Script – Student Progress Table.

### Steps:

1. Navigate to **All** → **Client Scripts** → **New**.
2. Fill in the required details, such as **Name**, **Table**, **UI Type**, and **Script**.
3. Click **Submit** to save the client script.

The screenshot shows the 'Client Script' configuration window for a script named 'Total Update'. The configuration includes the following fields:

- Name:** Total Update
- Table:** Student [u\_student]
- UI Type:** All
- Type:** onChange
- Field name:** Social
- Application:** Global
- Active:** ☒
- Inherited:** ☐
- Global:** ☒

The **Script** field contains the following code:

```
1 function onChange(control, oldValue, newValue, isLoading, isTemplate) {
2
3     if (isLoading || newValue === '') {
4         return;
5     }
6
7 }
8
9 //Type appropriate comment here, and begin script below
10
11 if (newValue) {
12     var a = parseInt(g_form.getValue('u_setup'));
13     var b = parseInt(g_form.getValue('u_hindi'));
14     var c = parseInt(g_form.getValue('u_english'));
15     var d = parseInt(g_form.getValue('u_maths'));
16     var e = parseInt(g_form.getValue('u_science'));
17 }
```

## Activity 5: Result Script – Student Progress Table.

### Steps:

1. Navigate to **All** → **Client Scripts** → **New**.
2. Fill in the required details, such as **Name**, **Table**, **UI Type**, and **Script**.
3. Click **Submit** to save the client script.

The screenshot shows the 'Client Script' configuration window for a script named 'Result'. The configuration includes the following fields:

- Name:** Result
- Table:** Student [u\_student]
- UI Type:** All
- Type:** onChange
- Field name:** Percentage
- Application:** Global
- Active:** ☒
- Inherited:** ☐
- Global:** ☒

The **Script** field contains the following code:

```
1 function onChange(control, oldValue, newValue, isLoading, isTemplate) {
2
3     if (isLoading || newValue === '') {
4         return;
5     }
6
7 }
8
9 //Type appropriate comment here, and begin script below
10
11 if (newValue) {
12     var a = parseInt(g_form.getValue('u_percentage')); // Convert the value to an integer for comparison
13     if (a >= 0 && a <= 59) {
14         g_form.setValue('u_result', 'fail');
15     } else if (a >= 60 && a <= 100) {
16         g_form.setValue('u_result', 'Pass');
17     } else {
18     }
19 }
```

## Activity 6: Percentage Script – Student Progress Table.

### Steps:

1. Navigate to **All** → **Client Scripts** → **New**.
2. Fill in the required details, such as **Name**, **Table**, **UI Type**, and **Script**.
3. Click **Submit** to save the client script.

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 "global:script.block.client.global" to false.

Name: Percentage

Table: Student [u\_student]

UI Type: All

Type: onChange

Field name: Total

Application: Global

Active: ☒

Inherited: ☐

Global: ☒

Description:

Messages:

Script:

```

1 function onChange(control, oldValue, newValue, isLoading, isTemplate) {
2
3   if (isLoading || newValue === '') {
4     return;
5   }
6
7
8
9
10  //Type appropriate comment here, and begin script below
11  var Total = g_form.getValue('u_total');
12  var Percentage = (Total/100)*100;
13  g_form.setValue('u_percentage', Percentage+"%");
14
15
16
17
18
19
20

```

Isolate script: ☒

## Result

1. Salesforce Table created.

servicenow All Favorites Salesforce - Create SAL0001009 Search

Submit

AdminNumber: SAL0001009

Admin Date:

Grade: -- None --

Student Name:

Father Name:

Mother Name:

Father Cell:

Mother Cell:

Submit

2. Admission Table created.

### 3. Student Progress Table created

## Conclusion

The project “**Educational Organisation Using ServiceNow**” successfully automated the management of student admissions and records. By using custom tables, forms, workflows, and client scripts, the system improved efficiency, reduced manual work, and provided a structured way to handle educational data