

Educational Organisation Using ServiceNow

TEAM ID: NM2025TMID18283

TEAM MEMBERS:

Team Leader: Yamuna.A

Team Member: Rithika.B

Team Member: Saranya.R

Team Member: Pooja.K

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.

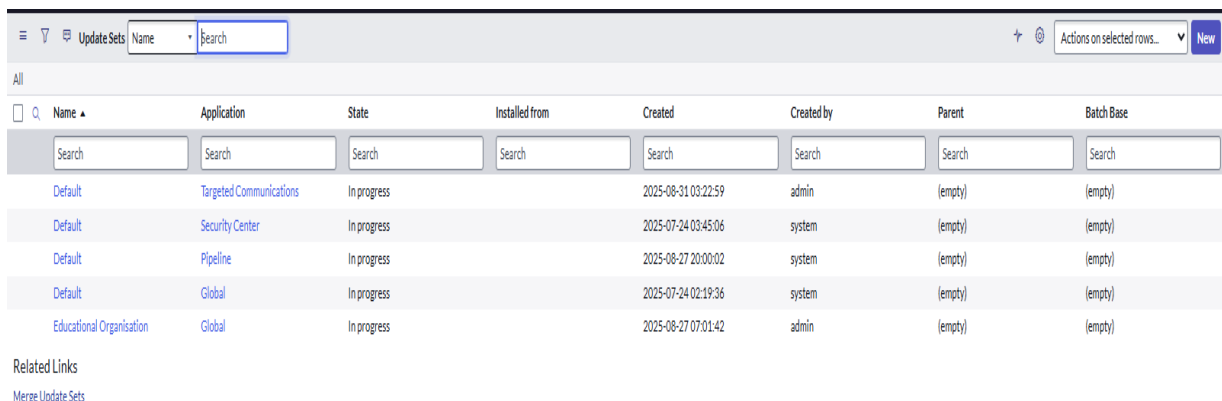
TASK INITIATION

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.



| Name | Application | State | Installed from | Created | Created by | Parent | Batch Base |
|--------------------------|-------------------------|-------------|----------------|---------------------|------------|---------|------------|
| 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
[Merge Update Sets](#)

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 Application: Global

* Name: u_salesforce

Columns Controls Application Access

Table Columns Column label Search 1 to 15 of 15 New

| Dictionary Entries | Column label | Type | Reference | Max length | Default value | Display |
|-------------------------------------|--------------|-------------------|-----------|------------|-------------------------------------|---------|
| <input checked="" type="checkbox"/> | Admin Date | Date | (empty) | 40 | | false |
| <input checked="" type="checkbox"/> | Admin Number | String | (empty) | 40 | javascript:getNextPadded() | true |
| <input checked="" type="checkbox"/> | Class | System Class Name | (empty) | 80 | javascript:current.getTable{Name()} | false |
| <input checked="" type="checkbox"/> | Created | DateTime | (empty) | 40 | | false |
| <input checked="" type="checkbox"/> | Created by | String | (empty) | 40 | | false |
| <input checked="" type="checkbox"/> | Father Call | String | (empty) | 40 | | false |
| <input checked="" type="checkbox"/> | Father Name | String | (empty) | 40 | | false |
| <input checked="" type="checkbox"/> | Grade | Choice | (empty) | 40 | | false |
| <input checked="" type="checkbox"/> | Mother Call | String | (empty) | 40 | | false |
| <input checked="" type="checkbox"/> | Mother Name | String | (empty) | 40 | | false |
| <input checked="" type="checkbox"/> | Student Name | String | (empty) | 40 | | false |
| <input checked="" type="checkbox"/> | Sys ID | Sys ID (UUID) | (empty) | 32 | | false |
| <input checked="" type="checkbox"/> | Updated | DateTime | (empty) | 40 | | false |
| <input checked="" type="checkbox"/> | Updated by | String | (empty) | 40 | | false |
| <input checked="" type="checkbox"/> | 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] Application: Global

* Type: String Active: ☒

* Column label: AdminNumber Function field: ☐

* Column name: u_admin_number Read only: ☐

* Max length: 40 Mandatory: ☐

Display: ☒

Attributes

Choice List Specification Calculated Value Default Value

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

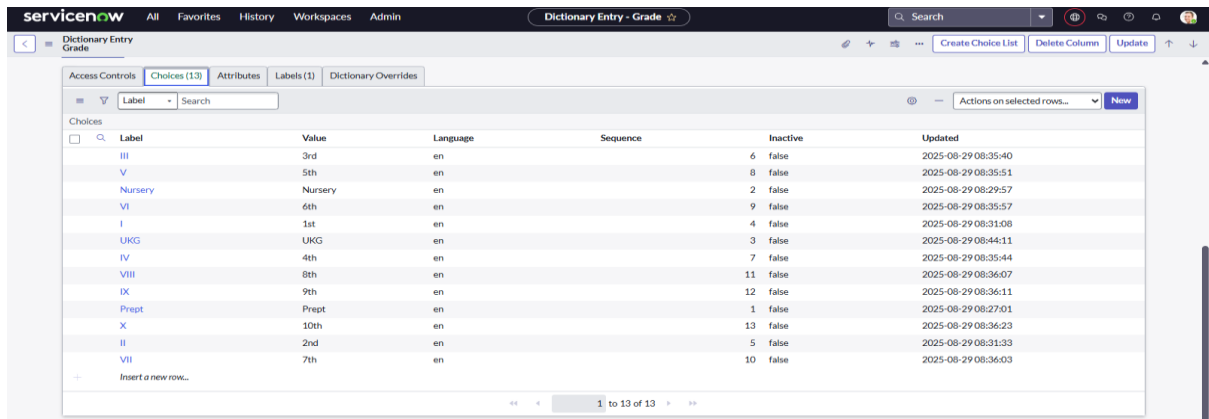
Use dynamic default: ☒

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.



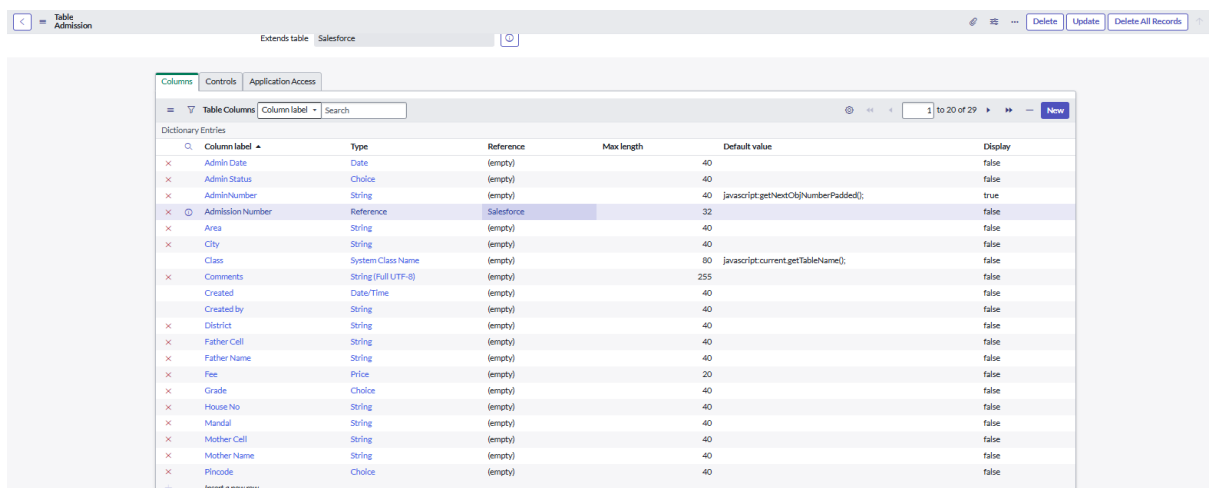
The screenshot shows the ServiceNow Dictionary Entry - Grade interface. The 'Choices' tab is selected, displaying a table of grade options. The table has columns for Label, Value, Language, Sequence, Inactive, and Updated. The data is as follows:

| 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.



The screenshot shows the ServiceNow Table Admission interface. The 'Columns' tab is selected, displaying a table of columns for the Admission table. The table has columns for Column label, Type, Reference, Max length, Default value, and Display. The data is as follows:

| 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:getNextCsb(Number Padded); | 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 |
| Phcode | Choice | (empty) | 40 | | false |

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

| Label | Value | Language | Sequence | Inactive | Updated |
|----------------|----------------|----------|----------|----------|---------------------|
| Near Bus Stand | Near Bus Stand | en | 2 | false | 2025-08-29 10:22:10 |
| Near Market | Near Market | en | 1 | false | 2025-08-29 10:22:08 |

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 | Salesforce | 32 | | false |
| Class | System Class Name | (empty) | 80 | javascript:current.getTableNames(); | 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 |

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.

The screenshot shows the 'Form Design' interface for the 'Admission (u_admission)' table. The left sidebar contains a 'Fields' section with a filter and a list of fields: AdmissionNumber, Class, Created, Created by, Updated, Updated by, and Updates. Below this is a 'Formatters' section with 'Activities (Filtered)', 'Contextual Search Results', and 'Ratings'. The main form area is divided into several sections: 'Admission (u_admission)' with fields for Admission Number, Class, Created, Created by, Updated, Updated by, and Updates; 'Process Flow (Formatter)' with fields for Admission Date, Grade, Fee, Father Cell, Mother Cell, and Admin Status; 'School Details' with fields for School Area and School; and 'Address' with fields for Pincode, Area, City, and District. Each field has a small icon and a '2 Column' dropdown menu.

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.

The screenshot shows the 'Form Design' interface for the 'Student (u_student)' table. The left sidebar contains a 'Fields' section with a filter and a list of fields: Class, Created, Created by, Updated, Updated by, and Updates. Below this is a 'Formatters' section with 'Activities (Filtered)', 'Contextual Search Results', and 'Ratings'. The main form area is divided into two sections: 'Student (u_student)' with fields for Admission Number, Grade, Student Name, Father Name, Mother Name, Father Cell, and Mother Cell; and 'Student Progress' with fields for Telugu, Hindi, English, Maths, Science, Social, Total, Percentage, and Result. Each field has a small icon and a '2 Column' dropdown menu.

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 the 'Number Maintenance' form in Salesforce. The form is titled 'Number Maintenance' and has a 'SAL' prefix. The fields are: 'Table' (Salesforce), 'Prefix' (SAL), 'Number' (1,000), 'Application' (Global), and 'Number of digits' (7). There are 'Update' and 'Delete' buttons at the bottom.

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 the 'Process Flow' form in Salesforce. The form is titled 'Flow Formatter' and has a 'new' prefix. The fields are: 'Table' (Admission [u_admission]), 'Name' (new), 'Application' (Global), 'Label' (New), 'Order' (1), and 'Active' (checked). There are 'Add Filter Condition' and 'Add "OR" Clause' buttons. The condition is 'Admin Status is New'. There are 'AND', 'OR', and 'X' buttons at the bottom.

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 a = g_form.getReference("u_admission_number");
4
5  g_form.setValue("u_admission_date", a.u_admission_date);
6
7  g_form.setValue("u_grade", a.u_grade);
8
9  g_form.setValue("u_student_name", a.u_student_name);
10
11 g_form.setValue("u_father_name", a.u_father_name);
12
13 g_form.setValue("u_mother_name", a.u_mother_name);
14
15 g_form.setValue("u_father_email", a.u_father_email);
16
17 g_form.setValue("u_mother_email", a.u_mother_email);
18
19 g_form.setValue("u_admission_date", a.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 'Pincode Update' client script configuration form. The form includes the following fields and values:

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

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

```
13 {  
14   g_form.setValue("u_mandals", "kaththal");  
15   g_form.setValue("u_city", "kaththal");  
16   g_form.setValue("u_district", "Rangabuddy");  
17 }  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35 }
```

The **Isolate script** checkbox is checked.

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 'Disable Fields' client script configuration form. The form includes the following fields and values:

- Name:** Disable Fields
- Table:** Student [u_student]
- UI Type:** All
- Type:** onLoad
- Application:** Global
- Active:** ☒
- Inherited:** ☐
- Global:** ☒

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

```
1 function onLoad()  
2 {  
3   //Type appropriate comment here, and begin script below  
4   g_form.setDisabled("u_total", true);  
5   g_form.setDisabled("u_percentage", true);  
6   g_form.setDisabled("u_result", true);  
7 }  
8  
9  
10  
11  
12  
13 }
```

The **Isolate script** checkbox is checked.

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 page for a script named 'Total Update'. The form includes the following fields and values:

- 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_telugu'));
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 page for a script named 'Result'. The form includes the following fields and values:

- 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 }
```

At the bottom of the page, there is a checkbox labeled 'Isolate script' which is currently unchecked.

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.

The screenshot shows the Salesforce Client Script configuration interface. At the top, there's a warning banner about strict mode. Below it, the configuration fields are as follows:

- Name:** Percentage
- Table:** Student [u_student]
- UI Type:** All
- Type:** onChange
- Field name:** Total
- Application:** Global
- Active:** ☒
- Inherited:** ☐
- Global:** ☒

The **Description** and **Messages** fields are empty. 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
10  //Type appropriate comment here, and begin script below
11
12  var Total = g_form.getValue('u_total');
13  var Percentage = (Total/600)*100;
14
15  g_form.setValue('u_percentage',Percentage+'%');
16
17
18
19
20 }
```

At the bottom, there is an **Isolate script** checkbox which is currently unchecked.

Result

1. Salesforce Table created.

The screenshot shows the Salesforce record page for a record with ID SAL0001009. The page header includes the ServiceNow logo, navigation links (All, Favorites), and a search bar. The record details are as follows:

- AdminNumber:** SAL0001009
- Admin Date:** [Calendar icon]
- Grade:** -- None --
- Student Name:** [Empty text field]
- Father Name:** [Empty text field]
- Mother Name:** [Empty text field]
- Father Cell:** [Empty text field]
- Mother Cell:** [Empty text field]

At the bottom of the form, there is a **Submit** button.

2. Admission Table created.

servicenow All Favorites History Workspaces Admission - Create SAL0001010 Search

Admission New record

New Join in progress Joined Rejected Rejoined Closed Cancelled

Admission Number Search

Purpose of join -- None --

Student Name

Father Name

Mother Name

Comments

Admin Date

Grade -- None --

Fee \$ 0.00

Father Cell

Mother Cell

Admin Status -- None --

School Details Address

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

Submit

3. Student Progress Table created

servicenow All Favorites History Workspaces Student - Create Created Search

Student New record

Admission Number Search

Grade -- None --

Student Name

Father Name

Mother Name

Father Cell

Mother Cell

Student Progress

Telugu

Hindi

English

Maths

Science

Social

Total

Percentage

Result

Submit

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