

Educational Organization Using Servicenow

Team Id: NM2025TMID01655

Team Leader: ASLIN SONIA S

Team Members:

INDHU R

DEVIKA D

ESTINA SATHEESH

Organization: Ponjesly College of Engineering

Department: Computer Science and Engineering

1. Executive Summary:

Educational institutions handle a wide variety of administrative and operational requests — admissions, student records, faculty data, maintenance requests, exam queries, library requests, hostel support, transport, etc. These are often managed manually (spreadsheets, emails, paper forms), causing latency, data inconsistency, and lack of visibility.

This project builds an Educational Management & Support System on the ServiceNow platform to centralize requests, automate routing and approvals, enforce role-based access, and enable tracking and reporting. The solution uses ServiceNow tables, forms, Flow Designer, client scripts, and update sets to deliver a robust, low-code automation for campus operations.

2. Problem Statement:

Educational institutions process hundreds of ad-hoc requests from students, faculty and staff daily. Current problems include:

- **Manual and fragmented workflows:** Requests are submitted via emails, physical forms, or disparate systems that do not talk to each other.

- **Delays and misrouting:** Requests are manually reviewed and forwarded which causes delays and sometimes gets assigned to wrong departments.
- **Lack of auditability and tracking:** Students and staff cannot easily track status; administrators lack a unified view for KPIs.
- **Data inconsistency:** Multiple copies of the same data (spreadsheets, local databases) lead to versioning issues and errors.
- **Resource inefficiencies:** Repetitive manual work leads to wasted staff hours and slower resolution times.

3.Goal:

Design and implement a ServiceNow-based solution to centralize request intake, automate routing and assignment, provide status visibility to requesters and teams, and enable administrators to monitor KPIs and audit trails — improving turnaround time, accuracy, and transparency.

4. Project Objectives:

- **Specific:** Build a ServiceNow application that supports centralized request submission (students/staff), auto-assignment to departmental support groups, and status tracking.
- **Measurable:** Reduce average request resolution time by X% (baseline to be measured). Provide dashboards with request counts by status, department, and SLA breaches.
- **Achievable:** Use ServiceNow Developer Instance, Flow Designer, and standard tables / client scripts.
- **Relevant:** Addresses operational inefficiencies for campus support processes.
- **Time-bound:** Core functionality delivered and tested within 9 working days (example timeline below).

5. Scope & Boundaries

In-Scope

- Create custom tables: Students, Staff, Requests (Campus Support Requests), Courses (optional).
- Build forms for request submission, admission, faculty record updates.
- Automate routing: Map request categories to Support Groups (IT, Library, Examination, Hostel, Transport).
- Use Flow Designer to auto-assign and notify groups/users.
- Configure number maintenance for unique ticket IDs.
- Implement client-side validations (client scripts).
- Provide basic dashboards / list views and reporting.
- Produce documentation and a demo video.

Out-of-Scope

- Complex integrations with external systems (e.g., ERP, LMS) — can be planned later.
- Advanced ML-based classification for automatic categorization.
- Full production deployment and multi-tenant features.

6. Deliverables

- ServiceNow Developer Instance configuration (update set).
- Custom Tables and Forms (Students, Staff, Campus Support Requests).
- Flow Designer workflows for auto-assignment and notifications.
- Client Scripts for validation and client-side behavior.
- Number Maintenance rules for unique IDs.
- List views & simple dashboards for administrators.
- Test cases and test report.
- Project documentation (this document, README, phase-wise docs).
- Demo video hosted on Google Drive (public link) and GitHub repository with documents.

7. Task Initiation

- Set up a ServiceNow Developer Instance.
- Create an Update Set for tracking custom changes.
- Design and create tables for managing students, teachers, and admissions.

- Customize form layouts and design for data entry.
- Configure number maintenance for record management.
- Implement workflow automation using Process Flow and Flow Designer.
- Write Client Scripts for field validation and automation.
- Test the functionality to ensure efficiency and accuracy.
- Document all phases of the project and record a demo video for submission.

Milestone 1: Users

Activity 1: Create Users

Creating a Update Set :

1. Click on All >> Local update sets .



2. Click on new.

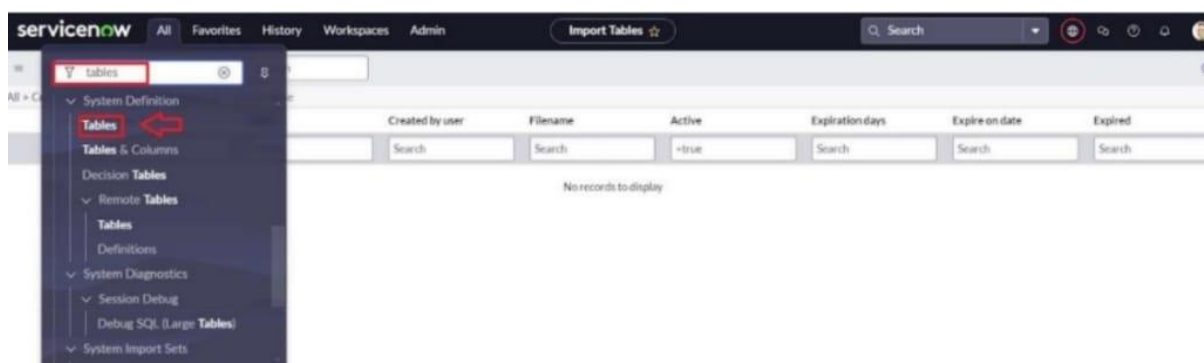
Name	Application	State	Installed from	Created	Created by	Parent	Batch Size
Default	App Engine Studio	In progress		2024-04-03 09:02:09	admin	(empty)	(empty)
Default	Global	In progress		2023-10-08 15:26:30	system	(empty)	(empty)

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

The screenshot shows the 'Update Set - Create New Update Set' form in ServiceNow. The 'Name' field contains 'New Update Set'. The 'State' is set to 'In progress'. The 'Application' is 'Global'. There are 'Submit' and 'Submit and Make Current' buttons at the bottom.

Creating Salesforce Table :

- All >> Tables.



- Click on new .

Label	Name	Extends table	Extensible	Updated
Adaptive Authentication Event	adaptive_auth_event	(empty)	false	2024-03-29 07:44:59
Agent Assist Recommendation	agent_assist_recommendation	Application File	false	2023-10-06 15:44:09
MID Server File	agent_file	(empty)	false	2023-10-06 15:29:08
Record Producer Configuration	aiua_rp_config	Application File	false	2023-10-06 15:47:19
Search Actions	aiua_sl_action	Application File	false	2023-10-06 15:47:19
AI Search ACL Overrides	ais_acl_overrides	Application File	false	2023-10-06 15:15:20
AI Search Active Table Ingestion Tracker	ais_active_table_ingestion_tracker	(empty)	false	2023-10-06 15:15:18

- Enter the Label(Anything you want): Salesforce >> Click on Name it will Automatically generate Api name.

ServiceNow recommends creating custom tables in scoped applications. To learn more about creating scoped applications, click [here](#).

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

* Name

Extends table

Application: Global

Create module ☒

Create mobile module ☒

Add module to menu: -- Create new --

New menu name:

- 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 .

ServiceNow

All Favorites History Workspaces Admin

Table - New Record

This form has annotations - click [?](#) to toggle them - ([click here](#) to never show this again)

* Label

* Name

Extends table

Columns * Controls Application Access

Table Columns for text Search

Dictionary Entries

Column label	Type	Reference	Max length
<input type="text"/>	<input type="text"/>		

Submit Cancel

Table Salesforce

This record is in the Global application, but Educational Organisation is the current application. To edit this record click [here](#).

Column label	Type	Reference	Max length
Admin Date	Date		40
Admin Number	String	javascript:getNextOtj(Number Padded);	40
Father Cell	String		40
Father Name	String		40
Grade	Choice		40
Mother Cell	String		40
Mother Name	String		40
Student Name	String		40

- For “Admin Number” Give Display as True and right click on the toggle bar on top >> save.



- Click on controls >> Enable Extensible.



- 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 .



- Click on “Grade” Column >> Click on Choices and give Label, Value and Sequence as given below.

servicenow All Favorites History Workspaces Admin Dictionary Entry - Grade

Dictionary Entry Grade

Access Controls Choices (13) Attributes Labels (1) Dictionary Overrides

Label Search

Actions on selected rows... New

Label	Value	Language	Sequence	Inactive	Updated
Prept	Prept	en	1	false	2024-04-02 02:10:36
Nursery	Nursery	en	2	false	2024-04-02 02:10:40
URG	URG	en	3	false	2024-04-02 02:10:43
I	1st	en	4	false	2024-04-02 02:12:50
II	2nd	en	5	false	2024-04-02 02:13:16
III	3rd	en	6	false	2024-04-02 02:13:23
IV	4th	en	7	false	2024-04-02 02:13:30
V	5th	en	8	false	2024-04-02 02:13:53
VI	6th	en	9	false	2024-04-02 02:14:57
VII	7th	en	10	false	2024-04-02 02:15:02
VIII	8th	en	11	false	2024-04-02 02:15:06
IX	9th	en	12	false	2024-04-02 02:15:12
X	10th	en	13	false	2024-04-02 02:15:15

Insert a new row...

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

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 Fee	String	(empty)	40		false
Mandate	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:

Label	Value	Language	Sequence	Inactive	Updated
New	New	en	1	false	2024-04-02 21:10:25
In progress	In progress	en	2	false	2024-04-02 21:11:03
Admitted	Admitted	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 21:13:05
Rejected	Rejected	en	6	false	2024-04-02 21:13:06
Cancelled	Cancelled	en	7	false	2024-04-02 21:13:27

- Create choice for Pincode as:

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:

Label	Value	Language	Sequence	Inactive	Updated
500079	500079	en	1	false	2024-04-02 21:15:46
500081	500081	en	2	false	2024-04-02 21:16:05
500082	500082	en	3	false	2024-04-02 21:16:05

- Create choice for School as:

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

- Create choice for School Area as:

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

Creating Student Progress Table :

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

Field Name	Field Type	Field Length	Field Format	Field Inactive
Admission Number	Reference	32		false
English	String	40	(empty)	false
Hindi	String	40	(empty)	false
Maths	String	40	(empty)	false
Percentage	String	40	(empty)	false
Result	String	40	(empty)	false
Science	String	40	(empty)	false
Social	String	40	(empty)	false
Telugu	String	40	(empty)	false
Total	String	40	(empty)	false

Configuring Table form for Student Progress Table :

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

Table student progress			
X	Telugu	String	(empty) 40
X	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

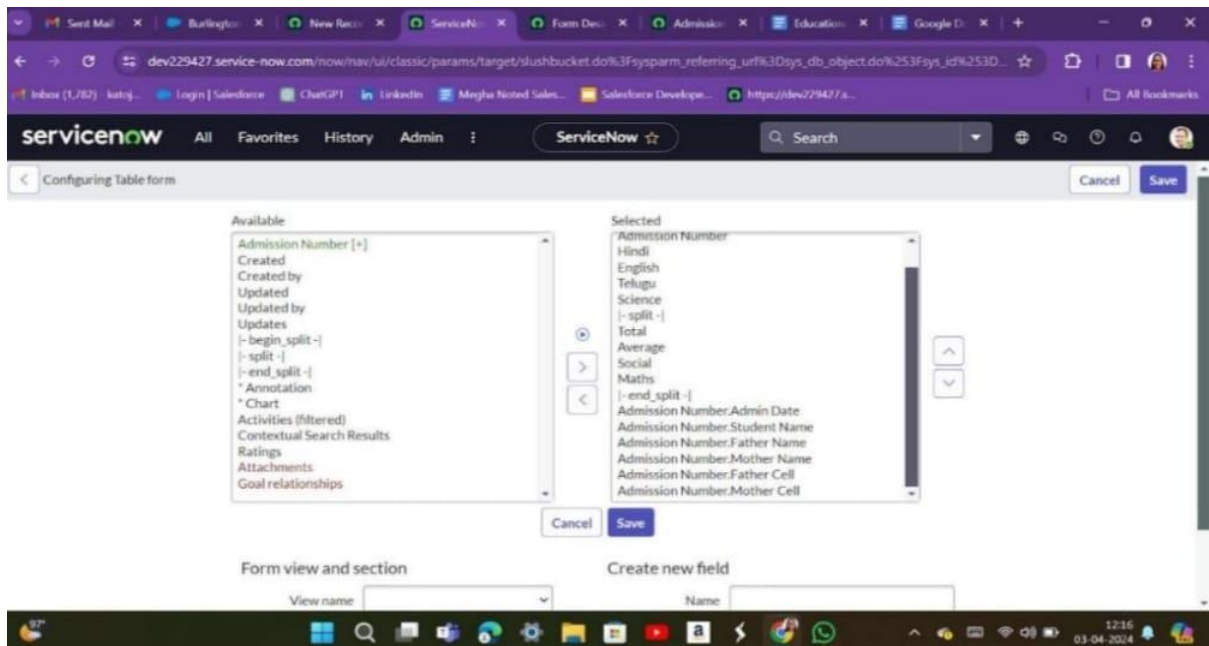
- begin_split -
- Admission Number
- Hindi
- English
- Telugu
- Science
- split -
- Total
- Average
- Social
- Maths
- end_split -

Cancel Save

Form view and section View name Default view

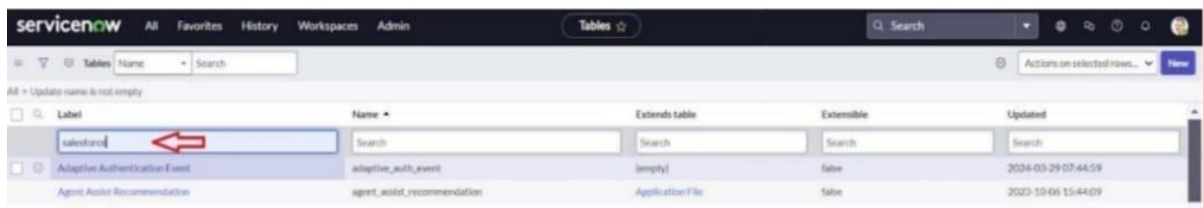
Create new field Name

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

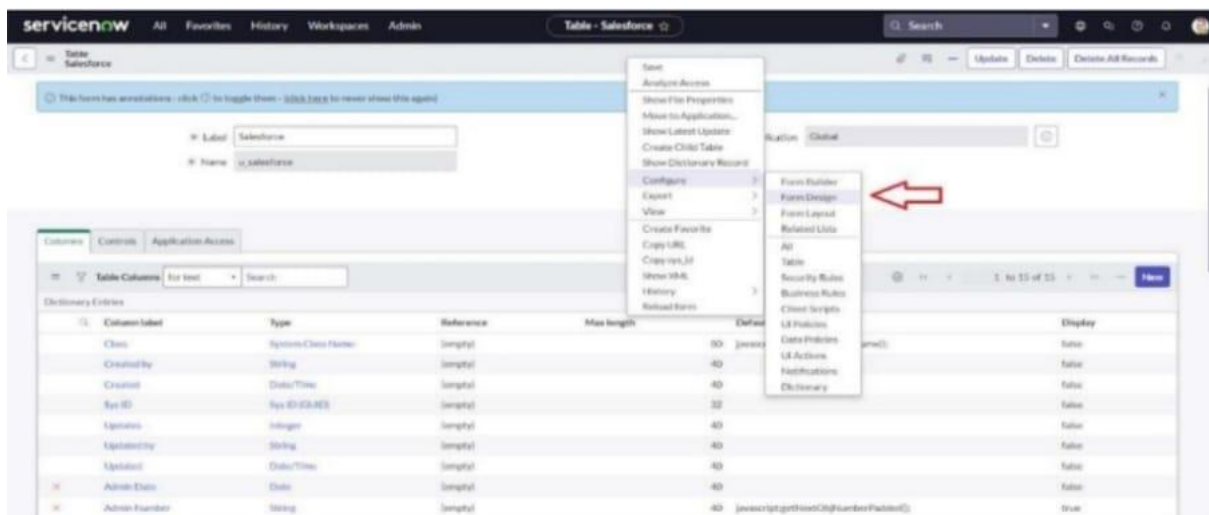


Creating Form Design for Salesforce Table :

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.

Creating Form Design for Admission Table :

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



Creating Form Design for Admission Table :

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

Creating Number Maintenance for Admin Number :

- All >> Number Maintenance >> New

Class Name	Reference	Max length	Default value	Display
Created by	String	(empty)	40	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.

Creating Process Flow for Admission Table :

All >> Process Flow>> New.

- Fill the Details as given Below

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

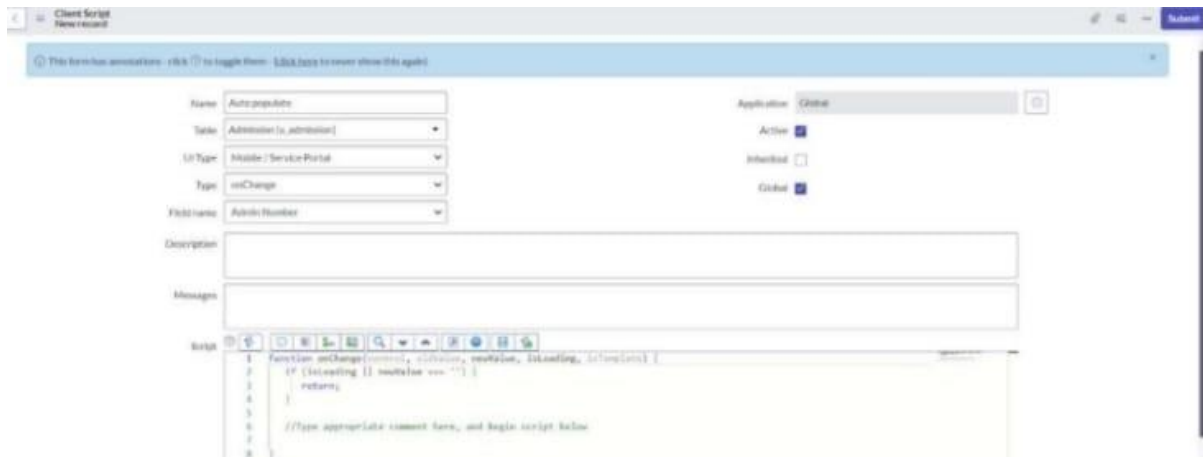
- Replace the Name and Label in order and click on Insert on stay.

Joined >> Rejected >> Rejoined >> Closed >> Cancelled.

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

Creating “Auto populate” Client Scripts for Admission Table

1. All >> Client Scripts >> New.
2. Fill the Details as given.



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

4. function onChange(control, oldValue, newValue, isLoading, isTemplate) {

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

6. return;

7. }

8. //Type appropriate comment here, and begin script below

9. var a =

10. g_form.getReference('u_admission_number');

11. g_form.setValue('u_admin_date',a.u_admin_date);

12. g_form.setValue('u_grade',a.u_grade);

13. g_form.setValue('u_student_name',a.u_student_name);

14. g_form.setValue('u_father_name',a.u_father_name);

15. g_form.setValue('u_mother_name',a.u_mother_name);

16. g_form.setValue('u_father_cell',a.u_father_cell);

17. g_form.setValue('u_mother_cell',a.u_mother_cell);

18. g_form.setDisabled('u_admin_date',a.u_admin_date);

19. g_form.setDisabled('u_grade',a.u_grade);

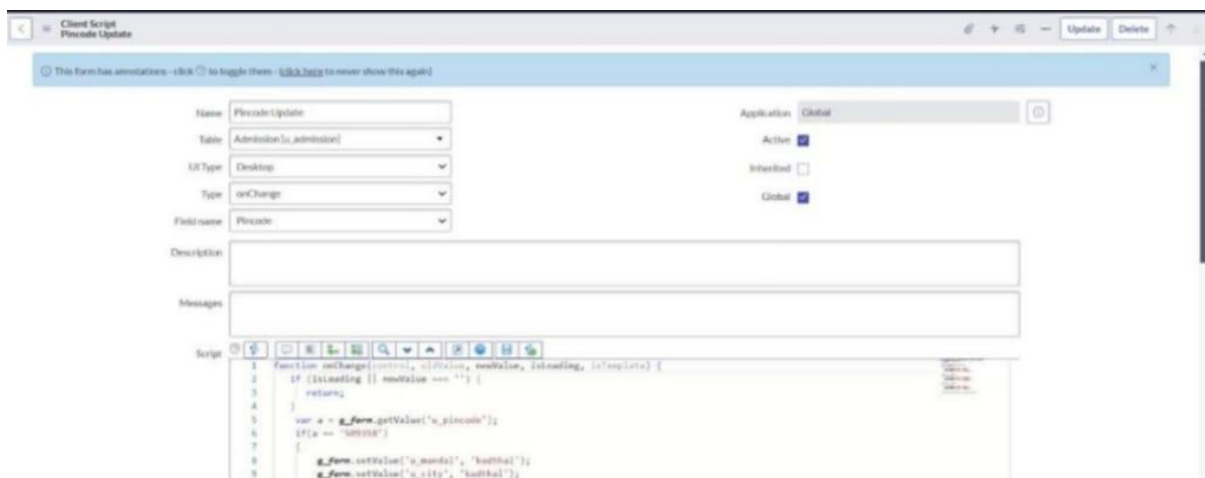
20. g_form.setDisabled('u_student_name',a.u_student_name);

21. g_form.setDisabled('u_father_name',a.u_father_name);

22. `g_form.setDisabled('u_mother_name',a.u_mother_name);`
23. `g_form.setDisabled('u_father_cell',a.u_father_cell`
24. `g_form.setDisabled('u_mother_cell',a.u_mother_c ell);`
25. `}`
26. Note: Make sure the Field names should be the same as you created.

Creating “Pincode Update” Client Scripts for Admission 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;
    }

    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
}

```

Creating “Disable Fields” Client Scripts for Student progress Table

- Fill the Details as given.

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

Name:

Table:

UI Type:

Type:

Application:

Active: ☒

Inherited: ☐

Global: ☒

Description:

Messages:

Script:

```

1 function onLoad() {
2     //Type appropriate comment here, and begin script below
3     g_form.setDisabled('u_total',true);
4     g_form.setDisabled('u_percentage',true);
5     g_form.setDisabled('u_result',true);
6 }

```

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

{

```

```

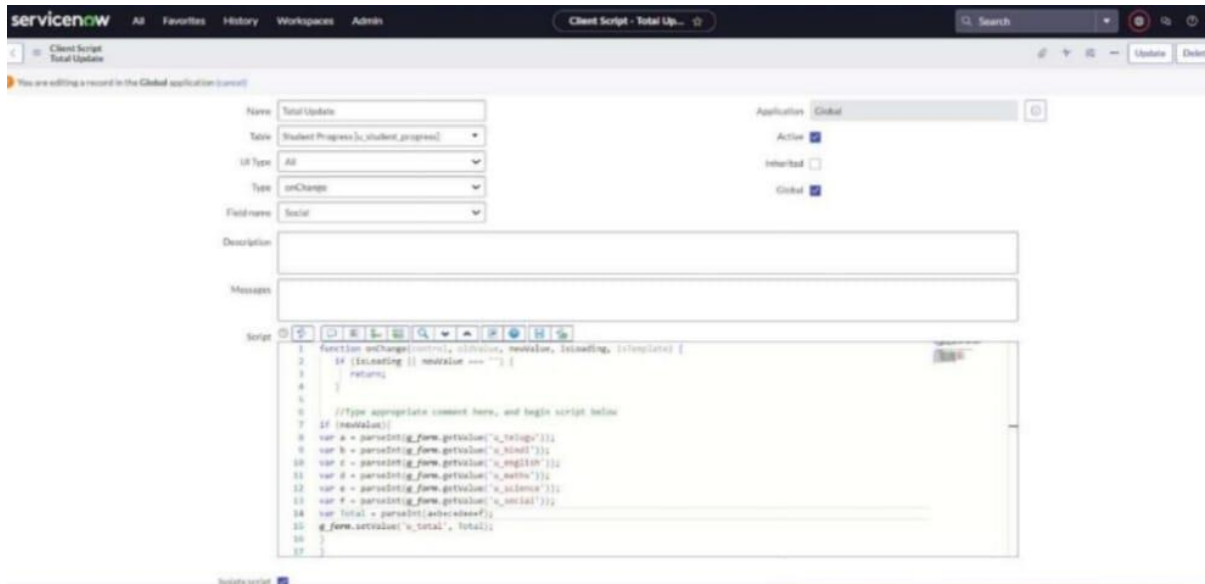
 //Type appropriate comment here, and begin script below
 g_form.setDisabled('u_total',true);
 g_form.setDisabled('u_percentage',true);

```

```
g_form.setDisabled('u_result',true);
}
```

## Creating “Total Update” 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
```

```
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);

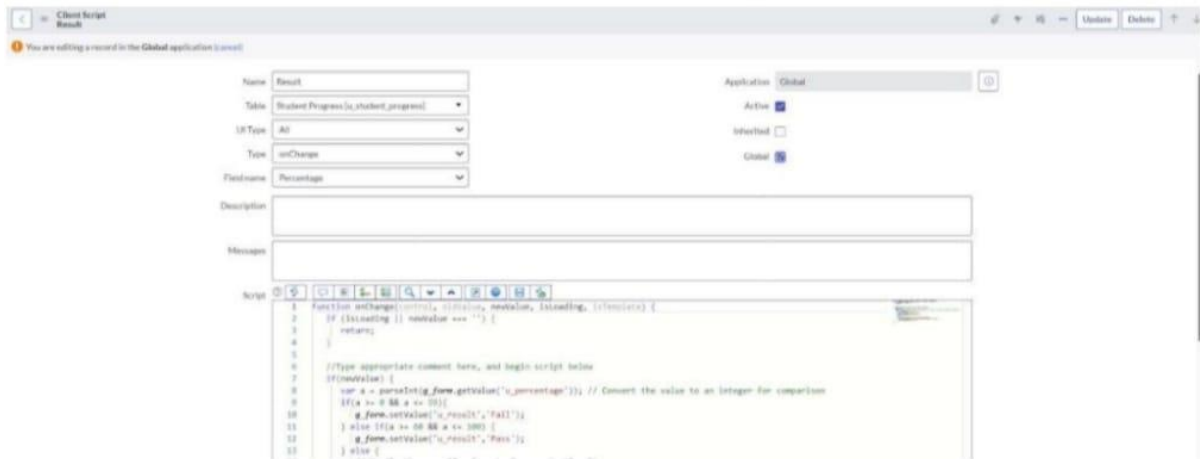
}

}

```

## Creating “Result” 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

```

```

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');
}
}
}
}

```

### Creating “Percentage” Client Scripts for Student progress Table

- Fill the Details as given.

The screenshot shows the Salesforce Client Script configuration window. The 'Name' field is 'Percentage'. The 'Table' is 'Student Progress (s\_student\_progress)'. The 'UI Type' is 'All'. The 'Type' is 'onChange'. The 'Field name' is 'Total'. The 'Application' is 'Global'. The 'Active' checkbox is checked. The 'Isolated' 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
6 //Type appropriate comment here, and begin script below
7 var total = g_form.getValue('u_total');
8 var Percentage = (total/1000)*100;
9 g_form.setValue('u_percentage', Percentage + '%');
10 }

```

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

- 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+'%');

}
```

## 8. Requirement Analysis

### Functional Requirements

- **Submit Request:** Authorized users (students/staff) must be able to submit support requests using a form with category, description, and attachments.
- **Auto Assignment:** The system should auto-assign incoming requests to pre-defined support groups based on category.
- **Notifications:** Send email notifications to the assigned group and to the requester at each milestone (created, assigned, in-progress, resolved).
- **Tracking & Audit:** Every request should preserve a history of actions (audit log) and allow status tracking.
- **Role-based Access:** Admins can update records; support users can view assigned requests only.
- **Reporting:** Provide list views and simple reports/dashboards for admin to monitor request status, SLA breaches, and volume.

### Non-functional Requirements

- **Performance:** Each create/update action should complete within acceptable response time (e.g., < 3 seconds under test load).
- **Scalability:** Designed so additional categories and groups can be added without rework.
- **Security:** Role-based access control, data visibility limited by user role.
- **Usability:** Forms should be mobile-friendly and intuitive.

## 9. Test Plan

### Test Types

- **Unit Tests:** Validate client scripts and business rules logic.
- **Integration Tests:** Flow Designer actions & email notifications.
- **System Tests:** End-to-end scenario testing across modules.

- Acceptance Tests: Final validation with mentor / stakeholder.

#### Example Test Cases

| ID    | TITLE                       | STEPS                                                              | EXPECTED RESULT                                                     |
|-------|-----------------------------|--------------------------------------------------------------------|---------------------------------------------------------------------|
| TC-01 | Create Student Record       | Create student with all mandatory fields                           | Student saved with unique student_id                                |
| TC-02 | Submit Support Request (IT) | Login as student; submit request category = IT                     | Request created; assigned_group = IT Support; notification sent     |
| TC-03 | Auto-assignment Logic       | Submit multiple requests for same category                         | Requests assigned to group per assignment rule (round robin / lead) |
| TC-04 | Client Validation           | Submit form leaving mandatory field blank                          | Client validation prevents submission and shows message             |
| TC-05 | Role-based Access           | Login as non-admin user and attempt to edit another user's request | Permission denied / read-only                                       |
| TC-06 | Notification Content        | Trigger assignment                                                 | Email contains request_id, short description, link to record        |

#### 10. Acceptance Criteria

- All critical test cases (TC-01 to TC-06) must pass.
- No critical defects outstanding.
- Demo video successfully demonstrates E2E flow and notifications.

- Documentation uploaded to GitHub and Update Set exported for reviewer.

## **11. Security & Privacy Considerations**

- Avoid storing sensitive personal data unnecessarily.
- Use ServiceNow roles to restrict access to student contact details.
- If real personal data is used for demo, anonymize or use sample records.

## **12. Maintenance & Future Enhancements**

- Integrate with student information system (SIS) for auto-populating student data.
- Add SLA timers and escalations.
- Add analytics dashboards and KPI tracking.
- Implement a knowledge base / FAQ for self-service.
- Add role-based mobile app views.

## **13. Expected Outcome**

- A fully functional ServiceNow-based Educational Management System.
- Automated workflows for admissions and student management.
- Improved data accuracy and simplified administrative operations.

## **14. Conclusion**

The Educational Organisation Using ServiceNow project successfully automated key academic and administrative processes within an institution. By using ServiceNow tools like tables, forms, and Flow Designer, the system improved efficiency, reduced manual work, and ensured accurate data management.

This project proved that cloud-based workflow automation can simplify institutional operations and enhance productivity. It provides a strong base for future upgrades like analytics dashboards and AI-based automation, promoting a smarter and more efficient educational environment.