NM1051 – SERVICENOW ADMINISTRATOR – SMART INTERNZ

EDUCATIONAL ORGANIZATION USING SERVICENOW

A PROJECT REPORT

Submitted by

SHALINI S	**(815422104049)**
PARANJOTHI S	**(815422104032)**
MENAGA S	**(815422104028)**
BAVANI B	**(815422104008)**

BACHELOR OF ENGINEERING

IN

SEVENTH SEMESTER

COMPUTER SCIENCE AND ENGINEERING

SRI RAMAKRISHNA COLLEGE OF ENGINEERING

ANNA UNIVERSITY : CHENNAI 600 025

NOV/DEC 2025

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"EDUCATIONAL ORGANIZATION USING SERVICENOW"** is the bonafide work of **"SHALINI S, PARANJOTHI S, MENAGA S, BAVANI B"** who carried out the project work under my supervision. No part of the dissertation has been submitted for any degree or any other academic award anywhere before.

SIGNATURE
Mrs. C.SURYA M.E.,
ASSISTANT PROFESSOR
Department of CSE
Sri Ramakrishna College of Engineering
Perambalur

SIGNATURE
Mr. R.DINESH RAJ M.E.,
HEAD OF THE DEPARTMENT
Department of CSE
Sri Ramakrishna College of Engineering
Perambalur

Submitted for the University Practical Examination held on…….

INTERNAL EXAMINER **EXTERNAL EXAMINER**

TABLE OF CONTENT

1. IDEATION PHASE

Introduction :

The ideation phase marks the beginning of the project development lifecycle, where creative ideas and innovative solutions are generated to solve existing problems. In this project, the focus is on using **ServiceNow** to enhance the functioning of an **educational organization** by automating its academic and administrative workflows.

Educational institutions today face several challenges such as manual record management, delayed responses to student queries, inefficient communication channels, and lack of centralized service systems. These problems reduce productivity and overall satisfaction among students and staff. Therefore, a cloud-based workflow automation platform like **ServiceNow** can offer a robust solution.

Problem Identification :

Educational institutions often struggle with:

- Manual handling of student and faculty requests.
- Slow communication between departments.
- Time-consuming approval processes.
- Lack of transparency in service request tracking.
- Inefficient IT resource management.

Idea Generation :

To address these challenges, ServiceNow can be introduced as a centralized platform where:

- Students and faculty can raise requests through a self-service portal.
- IT and administrative departments can track, manage, and resolve issues efficiently.
- Workflows can be automated to reduce human effort and errors.
- Management can monitor performance through dashboards and analytics.

Significance of the Solution :

The integration of ServiceNow into an educational environment:

- Promotes digital transformation in institutions.
- Ensures better accountability and faster issue resolution.
- Saves time and resources for both staff and students.
- Enhances institutional reputation through better service delivery.

Expected Outcomes :

By the end of this project, the institution will have a fully functional **ServiceNow portal** that allows:

- Students to submit service requests online.
- Staff to monitor and resolve those requests in real time.
- Administrators to view analytics and generate performance reports.

2. PROJECT PLANNING PHASE

Definition :

The project planning phase defines the roadmap for executing the project. It involves creating schedules, allocating resources, and defining milestones to ensure smooth development and implementation of ServiceNow in the educational environment.

Objectives of Planning :

- Identify resources and team roles.
- Define the project scope and deliverables.
- Create a timeline for each phase of the project.
- Estimate cost and determine risk mitigation strategies.

Project Scope :

The ServiceNow implementation will cover:

- Student request management system.
- Faculty support module.
- IT service automation.
- Administrative workflow automation.
- Asset and resource tracking.

Team and Roles :

- ServiceNow – Cloud-based platform used to automate workflows and manage institutional services.
- Incident – An unplanned disruption or issue that affects normal service operations.
- Request – A formal user submission seeking help, access, or information.
- Workflow – A sequence of automated actions that complete a process.
- Service Catalog – A digital list of services (like IT support, admin help, maintenance).
- Ticket – A record used to track the progress of a request or incident.
- Knowledge Base – A collection of FAQs and solutions for common problems.
- Dashboard – A visual display showing system performance, pending tasks, and analytics.
- Automation – Performing repetitive tasks automatically using system logic.
- SLA (Service Level Agreement) – Time-based standard defining how quickly an issue must be resolved.
- Performance Metrics – Indicators used to measure efficiency (e.g., uptime, response time).
- Self-Service Portal – An online platform for students and staff to raise and track requests.

3. PROJECT DESIGN PHASE

System Overview :

The design phase involves converting project requirements into a practical structure that guides implementation. It focuses on defining workflows, data flow, system interfaces, and user interaction design.

System Architecture :

The proposed architecture includes:

- **User Layer:** Students, staff, and administrators access the system through a web portal.
- **Application Layer:** ServiceNow manages workflows, service catalogs, and notifications.
- **Database Layer:** Stores user requests, tickets, approvals, and asset data.
- Required Date

Workflow Design :

- Student submits request via portal.
- Request routed to concerned department (IT/Admin).
- Assigned technician resolves issue.
- Status updated automatically.
- Student receives completion notification.

Data Flow :

Level 0:

Student → ServiceNow Portal → IT/Faculty/Admin → Response → Student

Level 1:

- Input: Request details.
- Process: Ticket generation and assignment.
- Output: Request resolution and feedback.

User Interface Design :

- Homepage: Login options for student, staff, and admin.
- Dashboard: Displays pending and resolved requests.
- Service Catalog: List of available support services.
- Reports Section: Generates service performance data.

4. REQUIREMENT ANALYSIS

Functional Requirements :

- **User Login Module:** Authentication for students, faculty, and admin.
- **Service Request Module:** Raise, track, and manage requests.
- **Notification System:** Alerts users on ticket updates.
- **Report Generation:** Admin dashboard for monitoring performance.
- **Knowledge Base:** Self-help articles for common issues.

Non-Functional Requirements :

- **Performance:** System should handle multiple requests concurrently.
- **Reliability:** 99% uptime for uninterrupted access.
- **Security:** Role-based access control and data encryption.
- **Usability:** Simple interface for users with minimal training.
- **Scalability:** Support for increasing number of users and modules.

Hardware Requirements :

- Server: Cloud-hosted (ServiceNow SaaS)
- Client Systems: Basic desktop/laptop with internet access
- Network: Stable broadband connection

Objectives of Requirement Analysis :

- To understand user needs (students, staff, admin).
- To identify necessary system functions and workflows.
- To define system performance and usability expectations.
- To ensure the platform supports scalability and security.
- To create a clear base for design and development phases.

Expected Outcome :

- A clear understanding of what the system must deliver.
- Well-defined requirements ensure accurate design and development.
- Helps prevent project delays and technical issues later.
- Guarantees that the ServiceNow platform meets institutional goals effectively.

5. PERFORMANCE TESTING

Types of Testing :

- **Load Testing:** Simulate multiple users submitting requests simultaneously.
- **Stress Testing:** Determine system behavior beyond normal capacity.
- **Scalability Testing:** Evaluate system's ability to handle growth in data.
- **Usability Testing:** Ensure user-friendly interface for all roles.
- **Security Testing:** Validate data privacy and access control.

Testing Tools :

- ServiceNow inbuilt testing module.
- Browser Developer Tools for response time analysis.
- Postman for API validation.

Importance of Defining Roles :

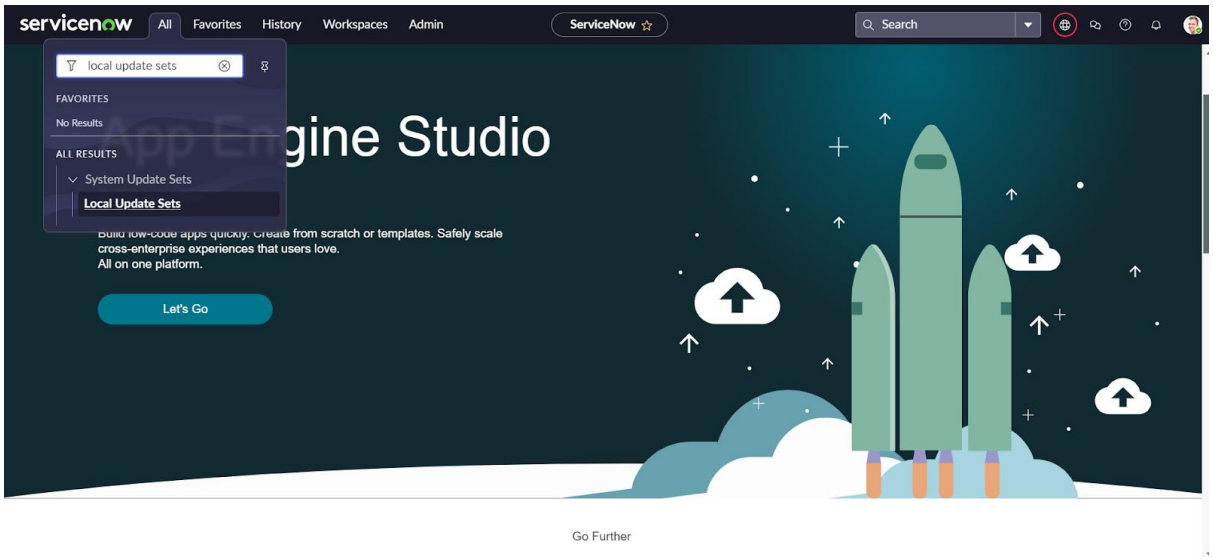
- Ensures accountability for every task.
- Promotes teamwork and smooth communication.
- Prevents overlapping duties and confusion.
- Helps achieve project goals efficiently.
- Improves monitoring and performance tracking.

Setting Up ServiceNow Instance :

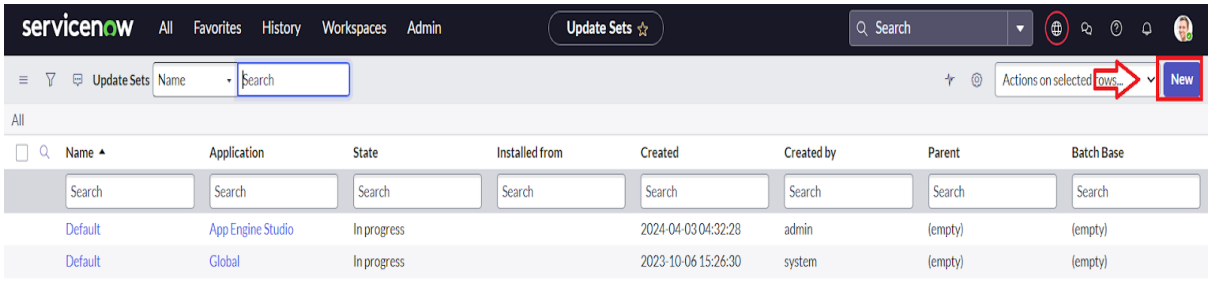
1. 1.Sign up for a developer account on the ServiceNow Developer site “https://developer.servicenow.com”.
2. 2.Once logged in, navigate to the "Personal Developer Instance" section.
3. 3.Click on "Request Instance" to create a new ServiceNow instance.
4. 4.Fill out the required information and submit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

Creating A Update Set :

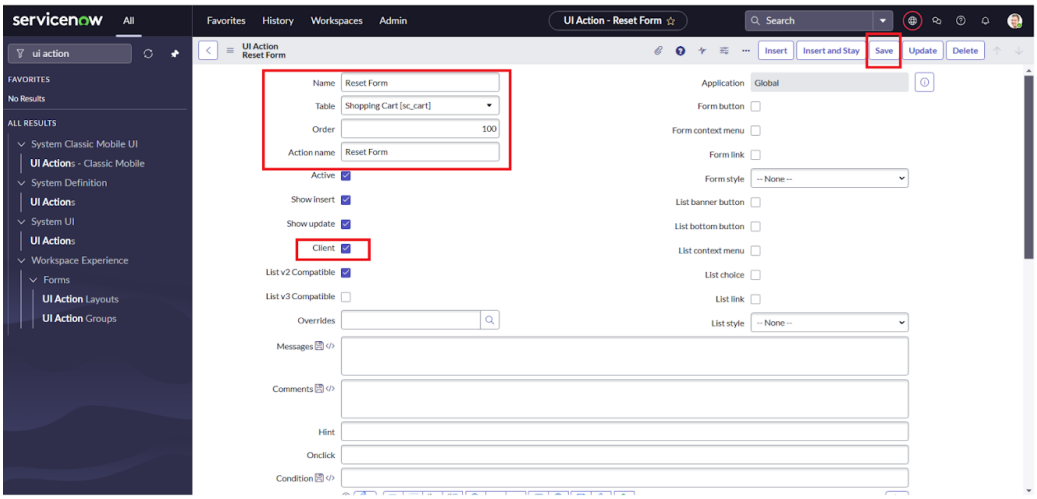
1. **Click on All >> Local update sets .**



2. Click on new



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



Creating Salesforce Table :

All >> Tables.

Click on new

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

Create columns as given below,Double Click on Column label and Enter the Column labels and click on the tick mark >> Give Type as given .

x

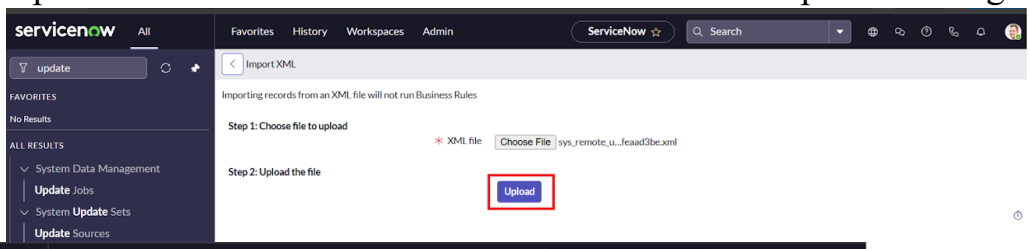
For “Admin Number” Give Display as True and right click on the toggle bar on top >> save.

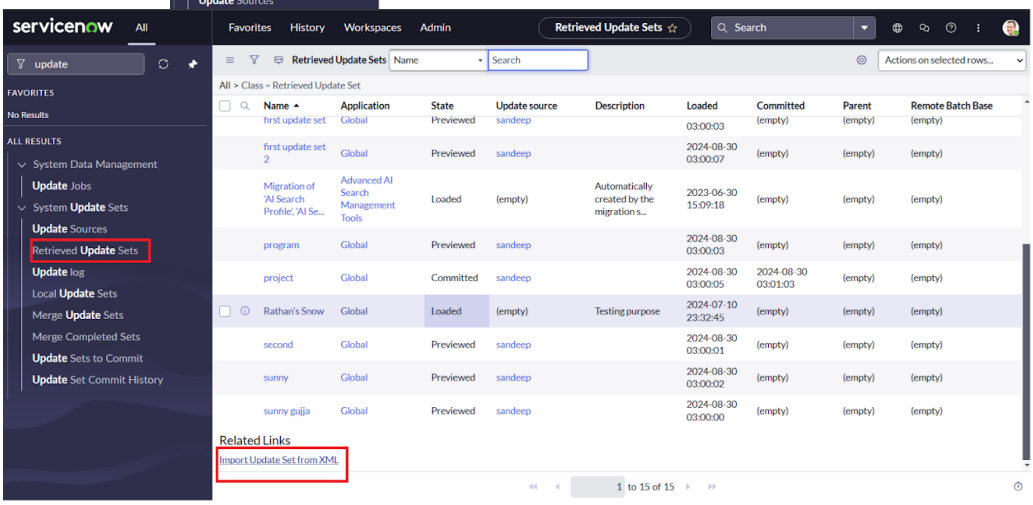
Click on controls >> Enable Extensible.

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.

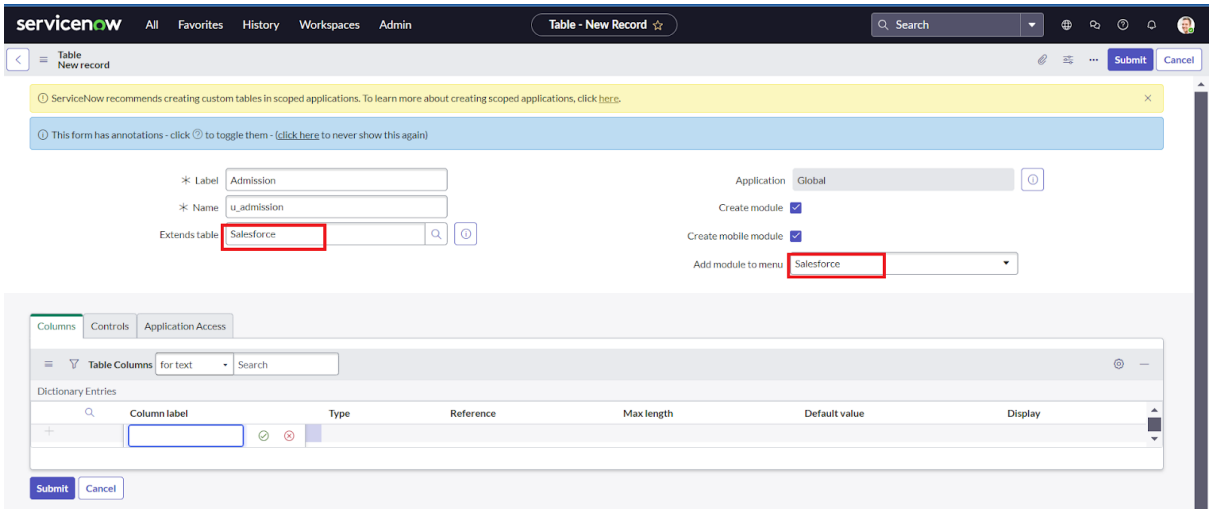
1.
2. Click on all>> search for update sets
3. Select “Retrieved update set” under system update set
4. It open retrieved update set list and scroll down
5. Click on Import update set from XML
6. Upload the downloaded file in XML file8.Click on Upload and it gets

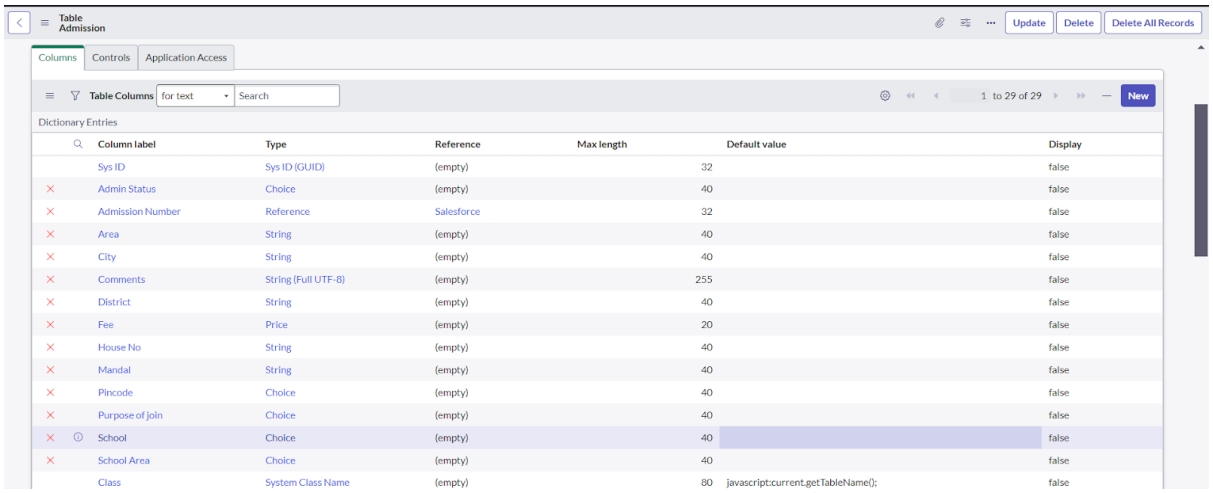




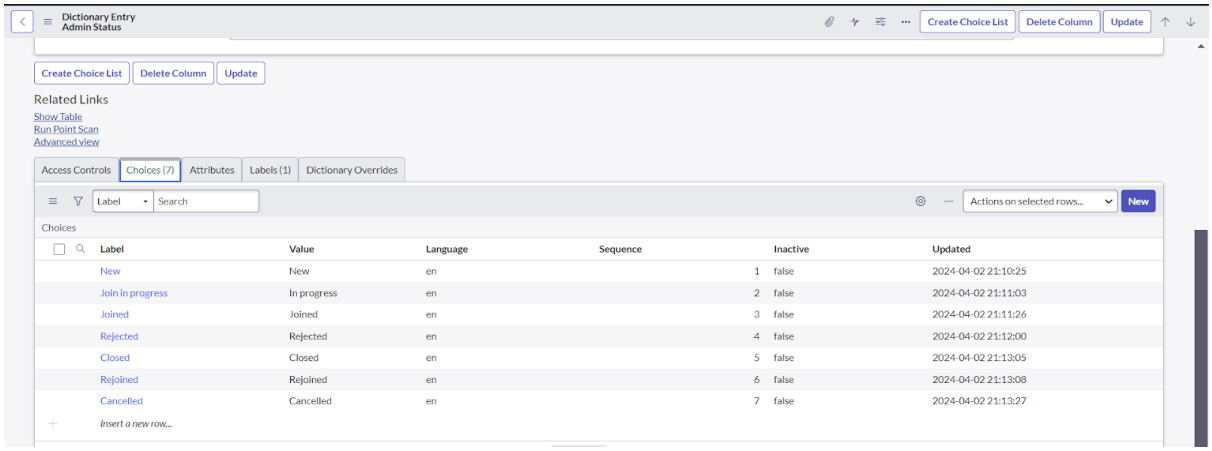
Creating Admission Table :

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

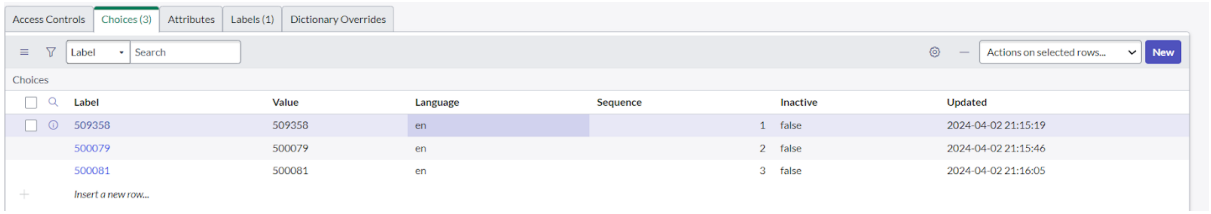




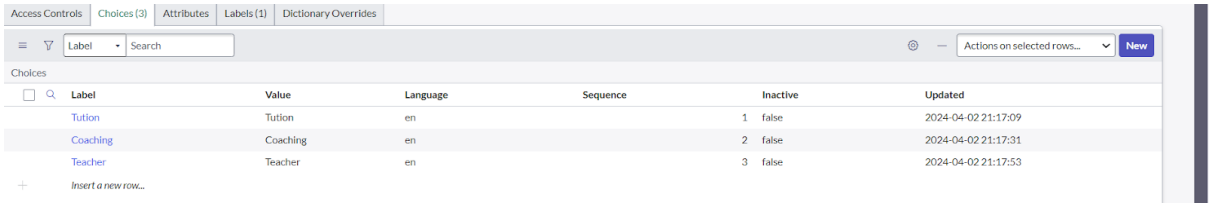
- **Create choice for Admin Status as:**



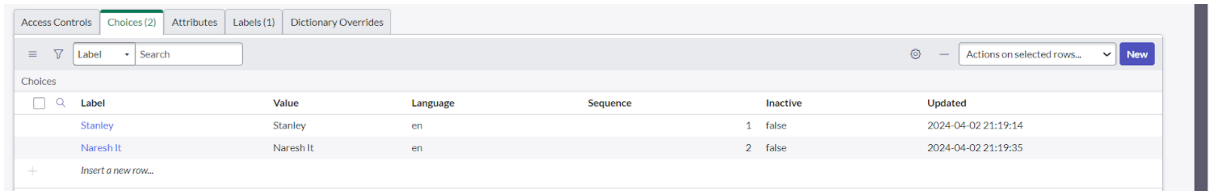
- **Create choice for Pincode as:**



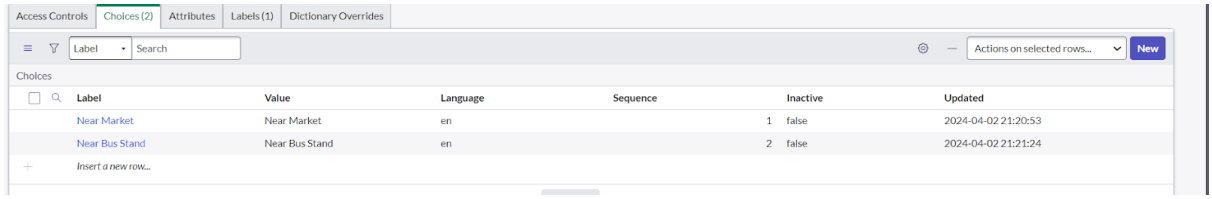
- **Create choice for Purpose of Join as:**



- **Create choice for School as:**

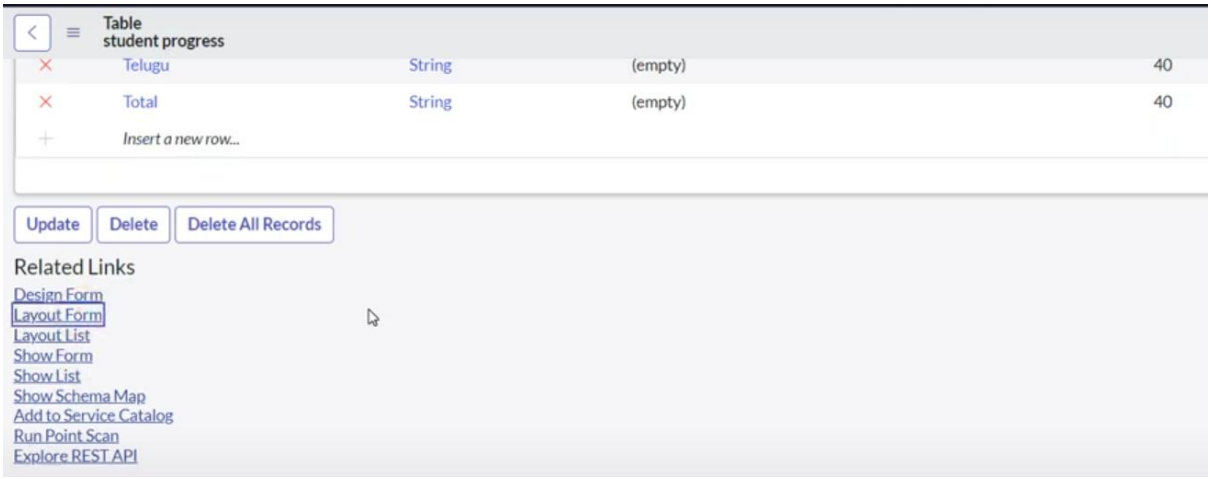


- **Create choice for School Area as:**

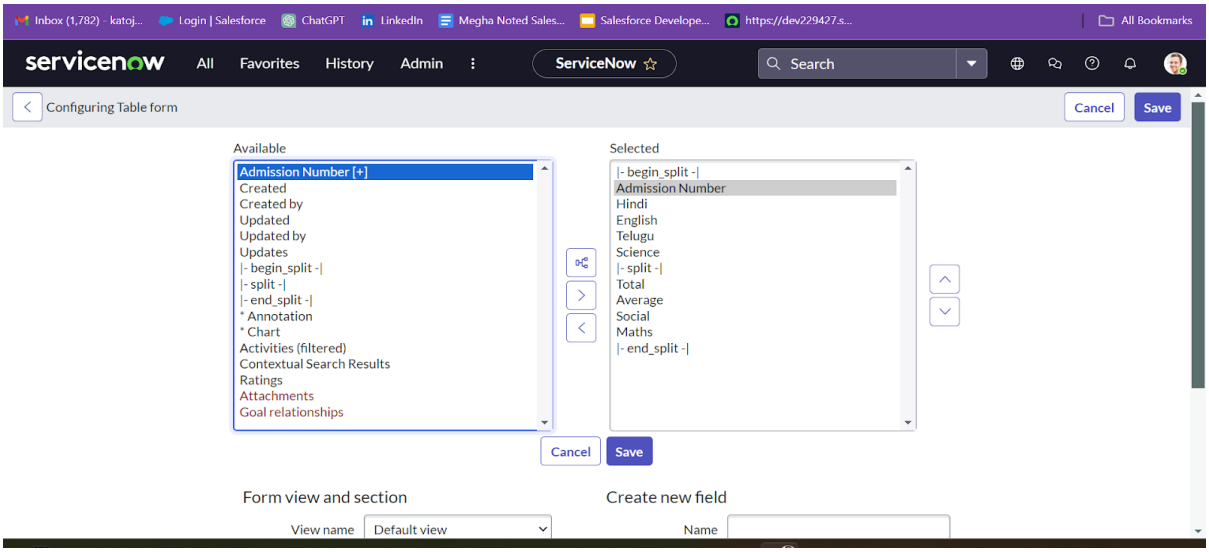


Configuring Table form for Student Progress Table :

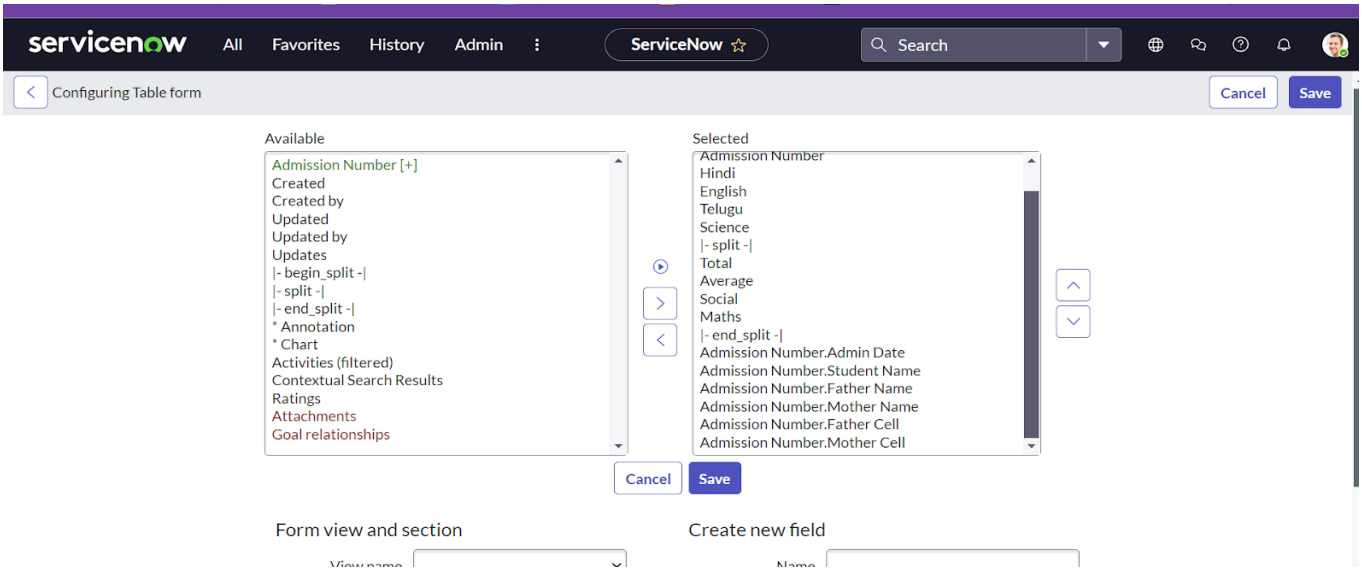
- In the Student Progress Table Page , Click on Layout form .



- Click on Admission Number [+] .

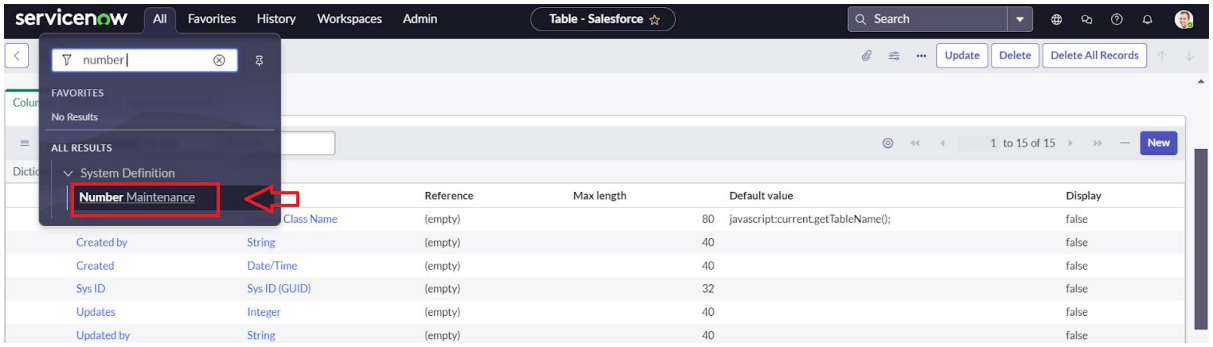


Select below Admission Number fields in Available side and send it to selected side as below >> save.

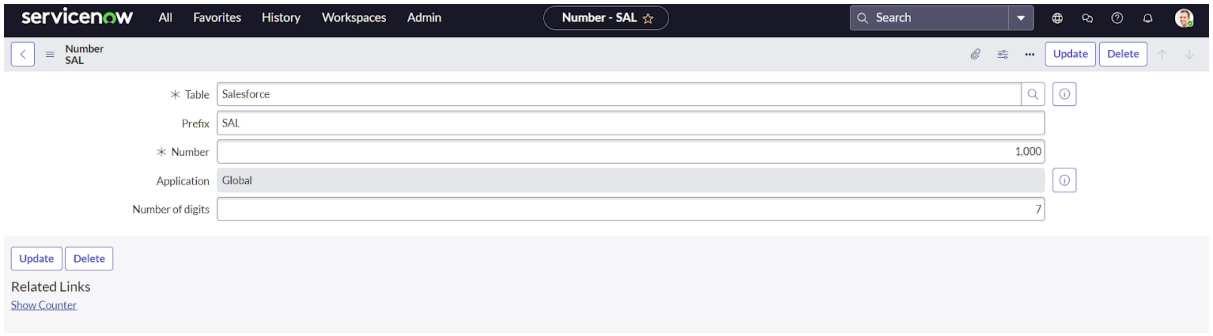


Creating Number Maintenance for Admin Number

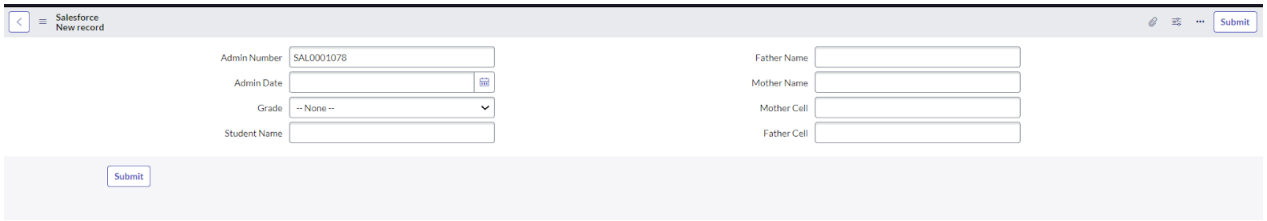
- **All >> Number Maintenance >> New**

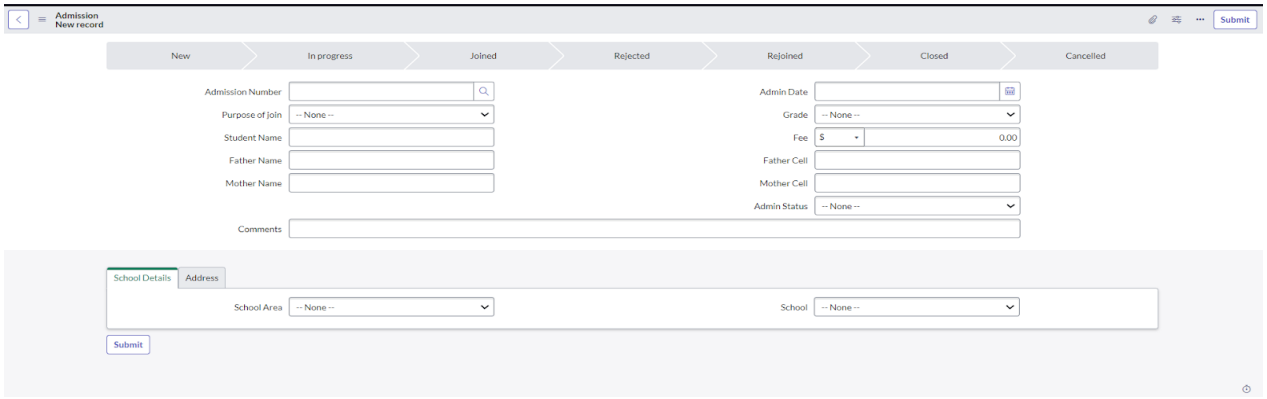


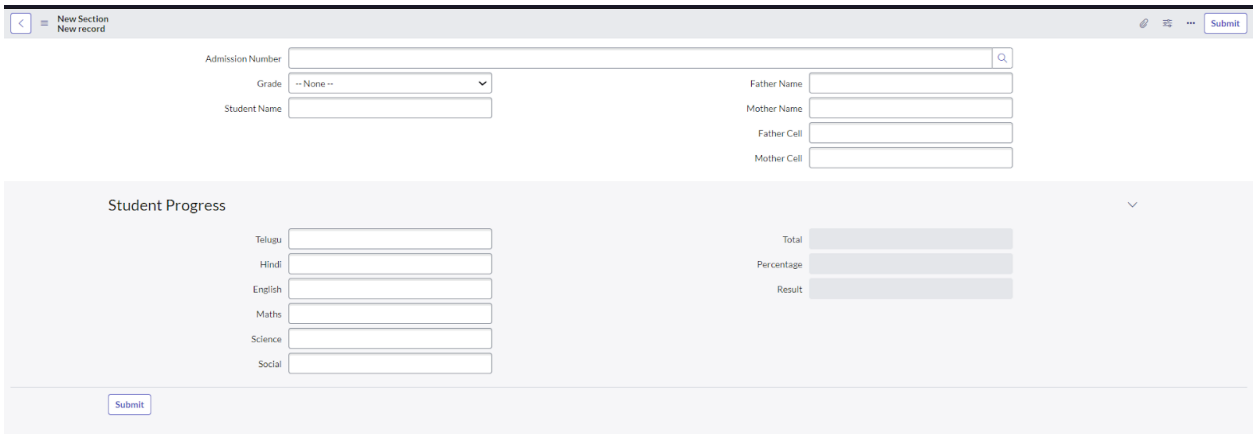
- **Fill the details >> Submit.**



Result :


Salesforce
New record
Submit
Admin Number
SAL0001078
Admin Date
Grade
-- None --
Student Name
Father Name
Mother Name
Mother Cell
Father Cell
Submit


Admission
New record
Submit
New
In progress
Joined
Rejected
Rejoined
Closed
Cancelled
Admission Number
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


New Section
New record
Submit
Admission Number
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 Organization Using ServiceNow”** successfully demonstrates how a cloud-based platform can streamline and automate academic and administrative workflows.It addresses the key challenges faced by institutions such as delayed communication, manual data handling, and lack of centralized service management.By implementing **ServiceNow**, educational institutions can provide faster, more transparent, and efficient services to both students and staff.The system ensures **automation of requests, incident management, and service tracking**, improving overall operational efficiency.

PRACTICE SCENARIOS FOR SERVICENOW ADMIN

1. **Create a new user for a contractor, assign them to an "IT Support" group, and ensure they can only access the *Incident* application.**
   Solution:

   - **Create the Contractor User**

     - Navigate to **Users** → *User Administration* > *Users*.
     - Click **New**.
     - Fill in details:
       - **User ID**: contractor1
       - **First name / Last name**: Contractor User
       - **Email**: contractor1@gmail.com
       - **Active**: Checked.
     - Save.

   - **Assign the User to the "IT Support" Group**

     - On the user record, scroll to **Groups** (related list).
     - Click **Edit**.
     - Add to the **IT Support** group.
     - Save.

   - **Restrict Access to Only the Incident Application**

     Now we need to make sure this contractor can only work with **Incident**.

     Option A: Role-Based Control (Mostly Preferred)

     - By default, Incident application requires **itil** role.
     - Instead of giving full **itil** access (which gives too much), do the following:
       - Create a **new custom role**, ex: **incident_contractor**.
       - Assign this role only to permissions needed for Incident (using ACLs).
       - Assign the new role to your contractor user.
       - Do **not** give **itil** or other broad roles.

Option B: Application Menu Restriction

- Go to **System Definition > Application Menus**.
- Open the **Incident** application menu.
- In the **Roles** field, add your custom role (**incident_contractor).**
    - This ensures only users with this role can see the Incident.
    - Verify Access
- **Impersonate** the contractor user.
- Check:
    - They should only see the **Incident application** in the left nav.
    - They can open/create/edit incidents (based on the ACLs you configured).
    - They cannot access other apps (like Change, Problem, etc.).

2. Assign a role to a new group so members can read *Knowledge Articles* but cannot create or edit them.

- **Create a New Group**

  - Navigate to User Administration > Groups.
  - Click New.
  - Enter a Name for the group (e.g., Knowledge Readers).
  - Optionally, add a Description.
  - Click Submit.

- **Assign the Appropriate Role**

To allow read-only access to Knowledge Base articles, assign the **knowledge** role:

  - Open the newly created group.
  - Scroll to the Roles related list.
  - Click Edit.
  - Add the role: knowledge
    - This role allows users to view published articles.
  - Click Save.

****Do NOT assign roles like **knowledge_admin or knowledge_manager,** which grant create/edit permissions.

- **Add Users to the Group**

  - In the group record, scroll to the Group Members related list.
  - Click Edit.
  - Select users you want to add.
  - Click Save.

- **Verify Access**

  - Log in as one of the group members.
  - Navigate to Knowledge > Articles.
  - Confirm they can view articles.
  - Try creating or editing an article — they should not have access.

3. **Configure a UI Policy that hides the "Work Notes" field unless the state is "In Progress"**.
Solution:

- **Navigate to UI Policies**
- Go to Application Navigator → type UI Policies → click System UI > UI Policies.
- Create a New UI Policy
- Click New.
- Select the Table → e.g., *Incident* (or whichever table you're working on).
- Provide a Name (e.g., *Hide Work Notes unless In Progress*).
- In the Conditions section, set:
    - Field = *State*
    - Operator = *is*
    - Value = *In Progress*.
- Check the box Active.
- Save the record.
- **Add a UI Policy Action**
- In the same UI Policy record, scroll to UI Policy Actions (Related List).
- Click New.
- Configure the action:
    - Field name = *Work notes*
    - Visible = *True* (since you want it visible only when the condition is met).
- Submit the action

4. **Configure a UI Policy to hide Notes section in incident, when state is In Progress.**

Solution:

- **Navigate to UI Policies**
- Go to Application Navigator → type UI Policies → click System UI > UI Policies.
- Create a New UI Policy
- Click New.
- Select the Table → e.g., *Incident* (or whichever table you're working on).
- Provide a Name (e.g., *Hide Work Notes unless In Progress*).
- In the Conditions section, set:
    - Field = *State*
    - Operator = *is*
    - Value = *In Progress*.
- Check the box Active.
- Save the record.
- **Make Run Script box True**
- Just write one line of code:
- g_form.setSectionDisplay('notes',false);
- Submit the action.

5. **Configure a response SLA, the SLA should pause, when the incident state is in On Hold vice versa**.

Create or Modify an SLA Definition

- Navigate to **Service Level Management > SLA Definitions**.
- Click **New** or open an existing SLA (e.g., "Response SLA").
- Fill in the basic details:
    - **Name**: Response SLA
    - **Table**: **Incident**
    - **Type**: Response
    - **Duration**: Set your desired time (e.g., 1 hour)
    - **Set SLA Conditions**
- Under the **Start Condition**:
    - Example: **State is New**
- Under the **Stop Condition**:
    - Example: State is Resolved or Closed
- Under the **Pause Condition**:
    - Add: **State is On Hold**

This ensures the SLA timer **pauses** when the incident is moved to **On Hold**, and **resumes** when it returns to another **New** state

- **Test the SLA Behavior**
- Create a test incident.
- Confirm SLA starts when an incident is created.
- Change state to **On Hold** — SLA should pause.
- Change back to **Active** — SLA should resume.
- Resolve the incident — SLA should stop.

6.**Configure an email notification that alerts the assigned group whenever a new *Change Request* is created.**

Solution:

- **Navigate to Notifications**
- In the **Application Navigator**, type **Notifications**.
- Go to **System Notification > Email > Notifications**.
- **Create a New Notification**

1. Click **New**.
2. Fill in the basic details:
    a. **Name**: *New Change Request Assigned Group Alert*
    b. **Table**: *Change Request [change_request]*
    c. **Active**: Checked

- **Define When to Send**

1. Under **When to send**, configure:
    a. **When to send**: *Insert* (since you want this when a new record is created).

- **Define Who Will Receive**

1. In the **Recipients** tab:
    a. Under **Users/Groups in fields**, choose **Assigned to group** (or the field name for assigned group).
    b. This ensures the entire assigned group gets the email.

- **Define What Will Contain**
- In the **What it will contain** tab:

Please review and take necessary action.

- **Save & Test**
- Save the Notification.
- Create a new **Change Request** record, assign it to a group.
- Verify that the email goes out to all members of the Assigned Group.

7. **Create a report showing the number of incidents opened by each department in the last 30 days.**

- **Navigate to Reports**
- Go to Reports > Open Reports Modules.
- Click Create a Report.
- **Define Report Source**
- Name: **Incidents by Department - Last 30 Days**
- Source Table: **Incident**
- **Set Conditions**
- **Under Filter, add:**
    - Opened At → on or after → Today - 30 days
    - Department → is not empty ***(optional, to exclude unassigned)***
- **Choose Report Type**
- Select Type: **Bar Chart** or **Pie Chart** (or **List** if you prefer tabular view)
- **Configure Grouping**
- Under Group By, select: **Department**
- Under Aggregation, choose: **Count**
- **Save and Run**
- Click Save.
- Click Run to view the report.

8. **Build a dashboard for Service Desk Managers showing KPIs like incidents by priority, created within a week, state wise also.**

Step 1: Create Individual Reports

You'll need to create three separate reports first:

- **Incidents by Priority**
- Go to: Reports > Create New
- Name: Incidents by Priority
- Type: Bar Chart or Pie Chart
- Group By: Priority
- Filter: Opened At → on or after → Today - 30 days
- **Incidents Created Within a Week**
- Name: Incidents Created - Last 7 Days
- Source Table: Incident
- Type: Time Series or Bar Chart
- Filter: Opened At → on or after → Today - 7 days

Step 2: Create a Dashboard

- Go to Self-Service > Dashboards.
- Click Create New Dashboard.
- Name: **Service Desk Manager KPIs**
- Add a Proper Description
- Click Submit.

Step 3: Add Reports to the Dashboard

1. Open the newly created dashboard.
2. Click Edit Content.
3. Use Add Reports to include:
    - **Incidents by Priority**
    - **Incidents Created - Last 7 Days**
    - **Incidents by State**

9. **Restrict the ability to delete records in the *Change Request* table so only users with the "admin" role can do so.**

- **Navigate to Access Control (ACLs)**
- In the **Application Navigator**, type **Access Control**.
- Go to **System Security > Access Control (ACL)**.
- **Create a New ACL Rule**
- Click **New**.
- Fill in details:
    - **Type**: *record*
    - **Operation**: *delete*
    - **Table**: *Change Request [change_request]*
    - **Name**: *(auto-populates when you pick table + operation)*
- **Define the Condition / Role**

In the **Requires role** field, add: **admin**

    - This ensures only users with the **admin** role can delete records.

- **Save & Test**
- Save the ACL.
- Test with a non-admin user → they should **not** see the delete option (or get a permission error if they try via URL).
- Test with an admin user → delete should work normally.

10. **Create a custom table and create two reference fields (ex: assignment group and assigned to). Display the users based on selection of assignment group.**

- **Create a Custom Table**

1. In the Application Navigator, type **Tables**.
2. Go to **System Definition > Tables**.
3. Click **New**.
    - Name: *u_custom_task*
    - Label: *Custom Task*.
    - Save.

- **Add Fields**

1. Open your table and go to the **Columns** tab.
2. Add two reference fields:
    - **Assignment Group** → Type = *Reference*, Table = *sys_user_group*.
    - **Assigned To** → Type = *Reference*, Table = *sys_user*.

- **Configure Reference Qualifier on "Assigned To"**
- We need to filter "Assigned To" users based on the selected Assignment Group.

Using Reference Qualifier

- Right click on the **Assigned To** field, click on **Configure Dictionary.**
- Go to **Dependent** Section, give the name of the Assignment Group(ex: u_ass_group)
- Update and Test the functionality.

11.**How to auto assign incidents when user selects a category as network, the same incident be assigned to Network group.**

Solution:

1. Go to Flow Designer → Designer.
2. Click New Flow.
    - Name: Assign Incident by Category
    - Trigger: Created or Updated → Table = Incident
3. Add a If action (Condition) with expression:
    - Select Trigger Record Category is Network
4. Under the If branch, add Action → Update Record:
    - Record: Trigger → Incident(Trigger Record)
    - Set field Assignment group → Network
5. Save and Activate the flow.
6. Test the Flow.

12. HR Groups members are only able to see HR Related Records in servicenow?

Solution:

Step 1: Create a Role for HR Access

Navigate to:

User Administration → Roles → New

1. Enter:
    - Name: hr_access
    - Description: Role to allow access to HR Cases
2. Click Submit.

Step 2: Assign the Role to HR Group

1. Navigate to:
   User Administration → Groups
2. Open your HR group record.
3. In the Roles tab → click Edit.
4. Move hr_access from Available → Selected.
5. Click Save.

Now all members of the HR group have the hr_access role.

Step 3: Create Access Control (ACL) for Viewing HR Cases

1. Navigate to:
   System Security → Access Control (ACL)
2. Click New.

Fill in:

Field	Value
Type	record
Operation	read
Table	Your HR Case table
Active	True

Step 4: Define Access Condition (No Script)

Scroll down to the Requires role section:

- Add the Role hr_access.

This means only users with the hr_access role can read/view HR Case records.

Step 5: Save and Test

1. Click Submit or Update to save the ACL.
2. Impersonate a non-HR user:
    - Go to your profile → click Impersonate User → choose a user *not in the HR group*.
    - Try opening an HR Case record → You should see a "Security constraints prevent access to requested page" message.
3. Now impersonate an HR group member:

They should be able to open HR Cases normally

13. When the Incident state changes to In Progress, Child incident related list should be hidden.

Solution:

1. Navigate to System UI → UI Policies → New.
2. Fill the header:
    - Name: Hide related lists when State is In Progress
    - Table: Incident
    - Active: checked
    - Global: checked
3. Condition: **State is In Progress**

(Use the exact label used in your instance for the In Progress state.)

4. Submit the UI Policy record.
5. In the UI Policy record click **New** under **UI Policy Actions**.

Set:

- **Field name:** select the related list–Child incident
- **Visible:** false
- **Read only:** optional
- Save and Test the UI Policy Action.

14.How to Display Incident number while loading the incident form

Solution:

1. Navigate to System UI → Client Scripts → New.
2. Fill the header:
    - Name: Show Incident Number on Load
    - Table: Incident
    - Type: onLoad
    - Active: True
3. Add this script:

```
function onLoad() {
  // Get the Incident number field value
  var incNum = g_form.getValue('number'); // 'number' is the field name
  alert('Incident Number: ' + incNum);
}
```

15. When the Incident state changes to In Progress, description should be hidden and short description should be mandatory.

Solution:

Step 1 — Navigate to Client Scripts

1. Go to:

   System UI → Client Scripts → New
2. Fill the header:
   - Name: Hide Description and Make Short Description Mandatory
   - Table: Incident
   - Type: onChange
   - Field name: state
   - Active: checked

Step 2 — Add the Client Script Code

```
function onChange(control, oldValue, newValue, isLoading) {
  if (isLoading) return;
  if (newValue === '2') {
     g_form.setDisplay('description', false);
     g_form.setMandatory('short_description', true);
  } else {
      g_form.setDisplay('description', true);
     g_form.setMandatory('short_description', false);
  }
```

16. Users can not change the state field values in the incident list.

Solution:

Step 1 — Navigate to Client Scripts

3. Go to:

   System UI → Client Scripts → New

4. Fill the header:
   - Name: Prevent State Inline Edit
   - Table: Incident
   - Type: onCellEdit
   - Field name: state
   - Active: checked

Step 2 — Add the Client Script Code

```
if(newValue==2){

alert('You can not edit this value');

saveAndClose==false;

}else{

saveAndClose==true;

}
```

17. How to set the Caller to Logged in user automatically in the incident table.

Solution:

1. Navigate: System Definition → Business Rules → New
2. Fill the details:
    - Name: Set Caller on Incident Create
    - Table: Incident
    - When: before
    - Insert/update: checked
    - Advanced: true
3. **Script**:

```
current.caller_id = gs.getUserID();
```

18.When a user updates an incident record, priority should change to Critical automatically.

Solution:

1. Navigate: System Definition → Business Rules → New

2. Settings:
    - Name: Set Priority field
    - Table: Incident
    - When: before
    - Update:checked
3. Script:

```
current.impact = 1;

current.urgency = 1;
```

19.Create a button on the Incident form that allows users to mark an Incident as Resolved with a single click.

Solution:

1. Navigate: System UI → UI Actions → New
2. Settings:
    - Name: Resolve Incident
    - Table: Incident
    - Action type: Form button
    - Active: checked
3. Script:
    - current.state = 6;
    - current.update();
    - action.setRedirectURL(current);

20.Create a button on the incident table that copies the Short Description value into the Description field.

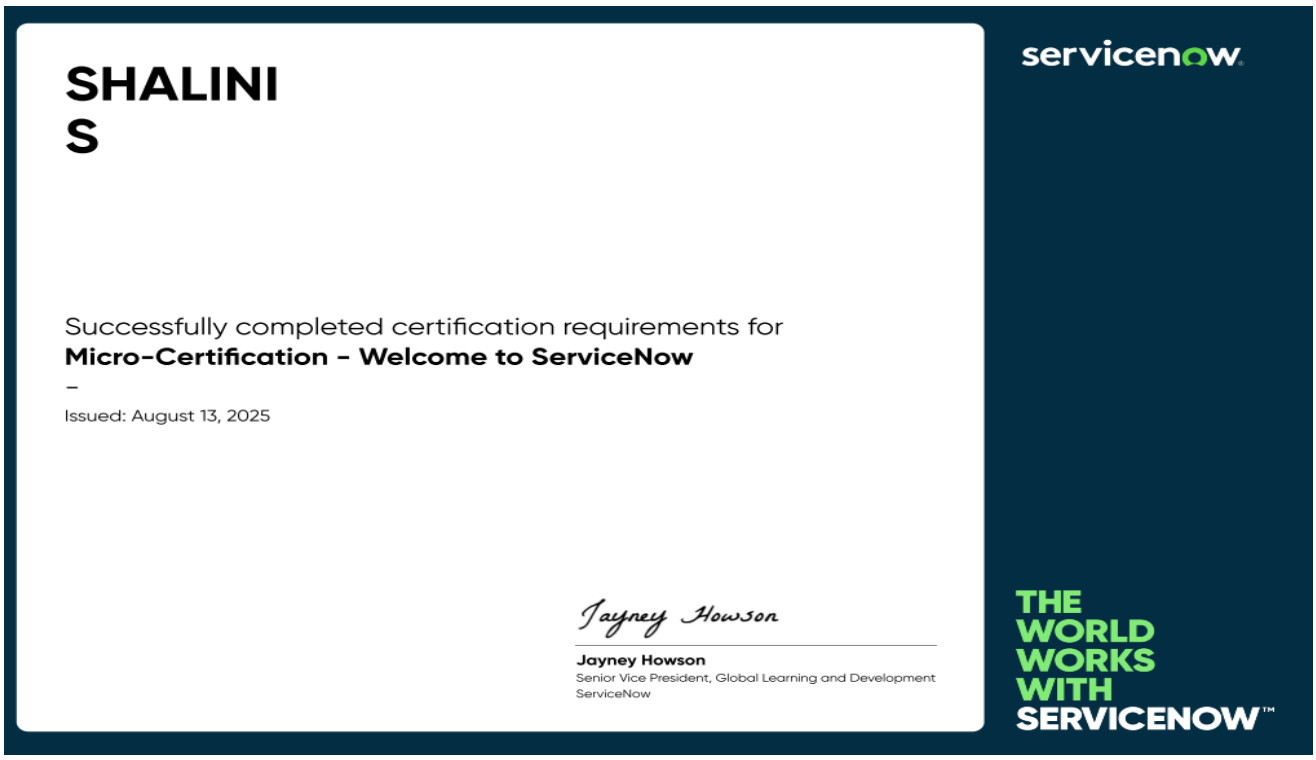
Solution:

1. Navigate: System UI → UI Actions → New
2. Settings:
    - Name: Copy Short Description
    - Table: Incident
    - Action type: Form button
    - Active: checked
3. Script:
    - current.description = current.short_description;
    - current.update();
    - action.setRedirectURL(current);

CERTIFICATES

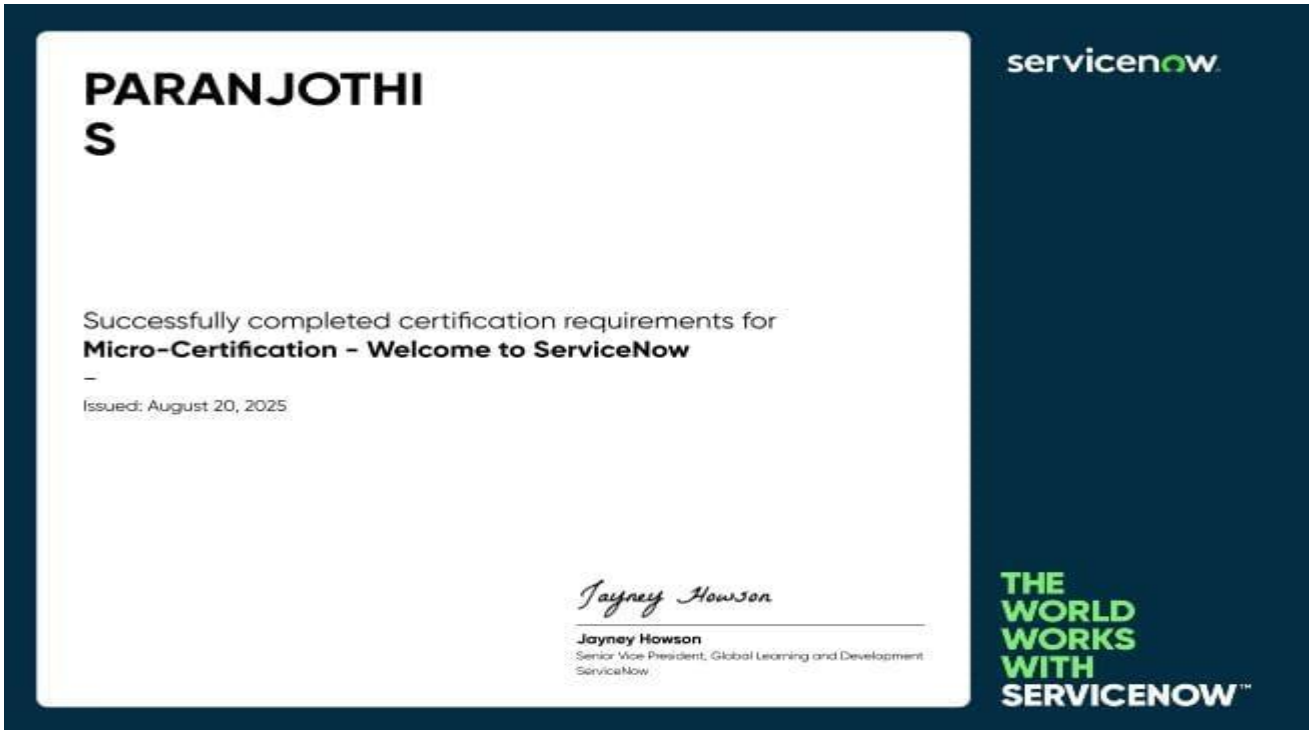
1) Generative AI in Action :

2) Welcome to Service Now – Micro Certification :



1)Generative AI in Action :

2) Welcome to Service Now – Micro Certification :



1)Generative AI in Action :

2) Welcome to Service Now – Micro Certification :

1)Generative AI in Action :

2) Welcome to Service Now – Micro Certification :