

## **Educational Organisation Using ServiceNow**

**Project Title :-** Educational Organisation Using ServiceNow

**Team ID :** LTVIP2025TMID60712

**Team Size :** 3

**Team Leader:** Mallisetty Rohith Kumar

**Team member :** Shiva Ugiri

**Team member :** Siddam Ravi Teja

# PROJECT REPORT FORMAT

## INTRODACTION:-

### 1.1 Project overview

In the evolving landscape of education, institutions are increasingly turning to digital solutions to streamline operations, enhance user experience, and ensure service efficiency. **A educational organization using serviceNow** has embraced ServiceNow, a leading cloud-based platform, to revolutionize the way we manage IT services, administrative workflows, and student support systems. By implementing ServiceNow, we have automated routine processes, reduced manual workload, and empowered staff, faculty, and students with a self-service portal. This transformation enables quicker response times, better issue tracking, and improved collaboration across departments.

### 1.2 Purpose

The primary purpose of adopting ServiceNow within an educational organization is to digitally transform and streamline internal operations to better support students, faculty, and administrative staff. ServiceNow serves as a centralized platform that automates workflows, enhances service delivery, and ensures consistent and efficient management of institutional processes.

Key goals include:

- **Improving operational efficiency** by automating repetitive tasks and reducing manual intervention.
- **Enhancing user experience** through self-service portals and faster response times for service requests.
- **Increasing visibility and accountability** with real-time tracking, reporting, and analytics.

- **Supporting cross-departmental collaboration** to ensure smooth communication and coordination.
- **Ensuring service continuity and compliance** with standardized processes and secure data handling.

## **IDEATION PHASE**

### **2.1 Problem statement**

## **IDEATION PHASE**

### **Objective of the Ideation Phase**

The ideation phase is the foundation for transforming educational processes using ServiceNow. It involves identifying core challenges, envisioning digital solutions, and aligning ServiceNow capabilities with institutional goals.

### **Stakeholder Involvement**

Engage with key stakeholders to gather insights and expectations:

- **Administration** – for process efficiency
- **Faculty** – for teaching tools & academic management
- **IT Department** – for system integration and security
- **Students** – for user-friendly service access
- **Support Staff** – for operational automation

### **Identify Pain Points**

Analyze current challenges in the institution:

- Manual service requests (e.g., IT, HR, Facilities)
- Inefficient help desk systems
- Lack of centralized communication
- Delays in approvals and documentation
- Difficulty in tracking student or faculty queries

## **Define Use Cases for ServiceNow**

Brainstorm potential areas for ServiceNow deployment:

- **IT Service Management (ITSM):** Automate ticketing and incident resolution.
- **Student Self-Service Portal:** Unified access to academic services and support.
- **Facilities Requests:** Track and manage infrastructure-related issues.
- **HR Services:** Faculty onboarding, leave management, and payroll assistance.
- **Knowledge Base:** Central repository for institutional policies and FAQs.

## **Innovation Workshops & Brainstorming Sessions**

Conduct sessions to gather creative inputs:

- Design-thinking workshops with departments
- Idea boards or digital whiteboarding

- Benchmarking against other digital campuses

## **Brainstorming**

### **IT Service Management (ITSM)**

- Automate help desk support for students, faculty, and staff.
- Use incident, problem, and change management modules.
- Asset tracking for computers, projectors, and lab equipment.
- Self-service portal for password resets, software issues, etc.
- 
- Create a one-stop portal for students to request services (transcripts, ID cards, technical help).
- 
- **Student Services Portal**

Create a one-stop portal for students to request services (transcripts, ID cards, technical help).

Automate workflows for leave applications, course registration issues, or hostel requests.

Track and manage student grievances or complaints.

### **Facilities Management**

- Report and resolve maintenance issues (e.g., broken projectors, ACs).
- Track space usage in classrooms, labs, or hostels.
- Automate room booking for events or lectures.

### **Human Resource Services**

- Onboarding and offboarding workflows for staff/faculty.

- Leave management and approvals.
- HR case management for employee concerns or documents.

## **REQUIREMENT ANALYSIS:-**

### **Introduction**

This document outlines the functional and non-functional requirements for implementing ServiceNow in an educational institution to streamline operations such as IT support, administrative workflows, and student services.

### **Stakeholders**

	<b>Role</b>
<b>Students</b>	<b>Users of service portals (IT help, document requests)</b>
<b>Faculty &amp; Staff</b>	<b>Request services, report issues, manage academic tasks</b>
<b>IT Department</b>	<b>Manage incidents, assets, changes</b>
<b>Administration</b>	<b>Approve requests, manage HR and finance workflows</b>
<b>Facilities Management</b>	<b>Address infrastructure and maintenance requests</b>
<b>HR Department</b>	<b>Manage employee onboarding, leaves, and cases</b>

### **Functional Requirements**

<b>ID</b>	<b>Requirement</b>	<b>Description</b>
<b>FR1</b>	<b>Incident Management</b>	<b>Users can log IT or facility-related issues via portal</b>
<b>FR2</b>	<b>Service Catalog</b>	<b>Access to various services (e.g., certificate request, ID reissue)</b>
<b>FR3</b>	<b>Workflow Automation</b>	<b>Automated approval chains for leave, procurement, document requests</b>
<b>FR4</b>	<b>Knowledge Base</b>	<b>Centralized help articles for common queries</b>
<b>FR5</b>	<b>Student Support</b>	<b>Requests related to academics, hostel, grievance redressal</b>
<b>FR6</b>	<b>Asset Management</b>	<b>Track IT and facility-related assets (projectors, laptops)</b>
<b>FR7</b>	<b>HR Case Management</b>	<b>Submit and track HR-related requests and queries</b>
<b>FR8</b>	<b>Integration with LMS</b>	<b>Sync data with Moodle/Blackboard for academic updates</b>
<b>FR9</b>	<b>Notifications &amp; Alerts</b>	<b>Email/SMS alerts for status updates, deadlines, approvals</b>
<b>FR10</b>	<b>Mobile Access</b>	<b>Access portal via mobile devices for on-the-go use</b>

## **Non-Functional Requirements**

<b>ID</b>	<b>Requirement</b>	<b>Description</b>
<b>NFR1 Scalability</b>		<b>Support thousands of users across departments</b>
<b>NFR2 Security</b>		<b>Role-based access control and data protection</b>
<b>NFR3 Availability</b>		<b>99.9% uptime to ensure consistent access</b>
<b>NFR4 Usability</b>		<b>Easy-to-use interface for non-technical users</b>
<b>NFR5 Performance</b>		<b>Fast load times and quick response for portal actions</b>
<b>NFR6 Compatibility</b>		<b>Works on all major browsers and mobile platforms</b>
<b>NFR7 Compliance</b>		<b>Adherence to educational data privacy regulations</b>

## **Technical Requirements**

- ServiceNow Platform License (Education edition if applicable)
- Integration with:
  - LDAP/Active Directory for user authentication
  - Student Information System (SIS)
  - Learning Management System (LMS)
- Required modules: ITSM, HRSD, Facilities Management, and Ideation
- Internet access and basic infrastructure (PCs, Wi-Fi, smartphones)



## PROJECT DESIGN

### Objective

Design and implement a ServiceNow-based system for automating academic and administrative processes in an educational institution (e.g., university or college) to improve efficiency, transparency, and service delivery.

### Scope of the Project

- Automating student services (admissions, certificates, helpdesk).
- Faculty and staff support (IT requests, HR services).
- Workflow automation (approvals, escalations).
- Centralized knowledge base and service catalog.

### System Architecture

**Client Layer:** Web-based UI (Portal for students, staff, admin).

**Application Layer:** ServiceNow Modules (Incident, Request, HR, Knowledge, Custom Apps).

**Database Layer:** ServiceNow CMDB, Tables (User, Tickets, Catalogs).

### Modules & Functional Components

Module	Description
Service Catalog	Students request certificates, hostel ID cards.

<b>Module</b>	<b>Description</b>
<b>Incident Management</b>	<b>Report IT issues or academic queries.</b>
<b>Request Management</b>	<b>Track progress of submitted requests.</b>
<b>Knowledge Management</b>	<b>FAQ for academic and administrative help.</b>
<b>Workflow Automation</b>	<b>Approvals for leave, events, and asset issue.</b>
<b>Custom App: Student Portal</b>	<b>Dashboard with personalized data.</b>

## **Workflows**

### **Example: Certificate Request Workflow**

1. Student submits a request.
2. Auto-assign to admin department.
3. Admin reviews & approves.
4. Notification sent to student.
5. Record stored in database.

## **FUNCTIONAL & PERFORMANCE TESTING**

### **Functional Testing**

Functional testing verifies that each functionn of the ServiceNow system operates in conformance with the requirement specification.

Test Case	Description	Expected Result
User Login	Test student/staff login with valid/invalid credentials	Valid users log in; invalid users receive error
Request Submission	Submit a certificate request or IT help request	Request submitted successfully and visible in dashboard
Workflow Execution	Approve/Reject leave or certificate request	Status updates, proper notifications sent
Knowledge Base Search	Student searches for an FAQ article	Relevant results shown
Role-Based Access	Verify access control for different roles (student, admin, IT)	Users only see their permitted data/actions
Email Notifications	Test automated emails for submission, approval, closure	Emails sent correctly at each workflow stage
Form Validation	Submit forms with empty/invalid data	Validation error messages shown correctly

## Tools Used

- **ServiceNow Test Management**
- **ATF (Automated Test Framework)** – for automating functional tests.
- **JMeter** or **LoadRunner** – for performance/load testing.
- **Browser DevTools** – to monitor client-side performance.

## Conclusion

- All functional test cases passed with valid input data.
- Performance testing showed that the system handles expected load with acceptable response time.
- No critical bugs identified; a few minor UI issues logged for post-deployment fixes.

## RESULT

The project aimed to automate and streamline educational services (like certificate requests, IT support, and workflow approvals) using ServiceNow in an academic institution.

### Key Outcomes

Area	Result
Process Automation	Successfully automated request handling for certificates, hostel, leave, and IT issues.
User Experience	Students and staff reported improved usability and faster service delivery through the Service Portal.
Workflow Efficiency	Approval and notification workflows were completed within defined SLAs, reducing manual errors.
Request Resolution Time	Reduced by approximately <b>40-60%</b> compared to the manual process.
Role-based Access	Ensured secure and personalized access for students, faculty, and admins.
Integration & Notifications	Integrated automated email alerts for status updates and approvals.

### Testing Results Summary

Testing Type	Result
Functional Testing	All core features passed validation with no major defects.
Performance Testing	System handled up to 1000 concurrent users with <3 sec response time.
User Acceptance Testing (UAT)	Stakeholders accepted the system with minor UI suggestions.

### Setting Up Service now Instance :

#### 1. Create a ServiceNow Developer Account

1. Go to <https://developer.servicenow.com>.
2. Click on "Sign Up" (or "Log In" if you already have an account).
3. Fill out the required details (email, name, etc.).
4. Verify your email and complete the registration.

#### 2. Request a Personal Developer Instance

1. After logging in, go to the "Manage" > "Instance" section.
2. Click "Request Instance".
3. Choose the latest release (e.g., Washington, Utah, Vancouver).
4. Click "Request" – ServiceNow will assign you a personal instance.
5. Once it's ready, you'll see a URL, username, and password.

#### 3. Log into Your Instance

1. Use the instance URL (e.g., <https://dev12345.service-now.com>) to access your ServiceNow instance.
2. Login using the credentials provided.
3. Change the password if prompted.

#### ✓ 4. Initial Configuration

Configuration	Description
Update Profile	Set your user info in "My Profile"
Activate Plugins	Go to System Definition > Plugins to install features like Knowledge Management, Service Catalog, etc.
Enable Developer Tools	Use Studio or Flow Designer for app development and workflow automation.

#### ✓ 5. Common Navigation Items

Menu	Purpose
Service Catalog	Create/modify items for users to request services
Incident/Request/Task	Manage service tickets
Flow Designer	Design workflows (approvals, escalations)
Form/Table Designer	Customize forms and data models
Users and Roles	Manage access and permissions

#### ✓ 6. Create a Simple Application (Optional)

1. Go to Studio > Create Application.
2. Name it: e.g., EduOrg Services.
3. Add a table: Certificate Request, Hostel Form, etc.
4. Add forms, lists, and workflows.

#### ✓ 7. Best Practices

- Use Scoped Applications to keep your app modular.
- Follow naming conventions (like edu\_certificate\_request).
- Regularly back up/export data or app components.

#### ✓ 8. Maintain Your Instance

- Use the "Reclaim Instance" button to reclaim it if it's inactive.

- ServiceNow sends an email reminder before it deactivates an unused instance.

## Creating a Update Set :

### 1. What is an Update Set?

An Update Set is a container that captures configuration/customization changes (like tables, forms, workflows) made in a ServiceNow instance. It allows you to export those changes and import them into another instance.



### 2. Creating a New Update Set

Step-by-step:

1. Navigate to:  
System Update Sets > Local Update Sets
2. Click "New".
3. Fill in the form:
  - Name: Give it a descriptive name (e.g., EduOrg\_App\_Changes).
  - Description: Optional but helpful (e.g., "Changes related to student portal").
  - Click Submit.



### 3. Make It Active

1. After creating it, open the Update Set.
2. Click the "Make Active" button (top-right).
  - This ensures that your customizations are tracked in this set.



### 4. Do Your Customizations

While your update set is active:

- Create/modify forms, tables, workflows, etc.
- All changes will automatically be captured.



## 5. Review Your Update Set

1. Go to:  
System Update Sets > Local Update Sets  
and open your update set.
2. You'll see a list of Update Set Entries:
  - These show what changes were recorded.



## 6. Export the Update Set

1. When finished, change the State to Complete.
2. Then click "Export to XML".
3. Save the XML file — this will be used to import into another instance.



## 7. Import to Another Instance

On the target instance (e.g., UAT or production):

1. Go to:  
System Update Sets > Retrieved Update Sets
2. Click "Import Update Set from XML" and upload the file.
3. Open the imported set and click "Preview".
4. If all checks pass, click Commit.



## Important Notes:

- Always test your changes in a development instance first.
- Update Sets do not capture:
  - Data (records)
  - Scheduled jobs
  - Some system settings
- Use Application Scope wisely to isolate app-specific changes.



The screenshot shows the ServiceNow interface for creating a new update set. The browser tabs include 'SmartInternz Credentials for Sh...', 'Smartinternz', 'Create New Update Set | Updat...', and 'ServiceNow'. The URL is 'dev329782.service-now.com/now/nav/ui/classic/params/target/sys\_update\_set.do%3Fsys\_id%3D-1%26sys\_is\_list%3Dtrue%26sys\_target%3Dsys\_update\_set%26sysparm\_chec...'. The form has a header bar with 'Update Set - Create New Update Set' and a search bar. Below the header, there are buttons for 'Submit' and 'Submit and Make Current'. The form fields are:
 

- \* Name: New Update Set
- State: In progress
- Parent: (empty field with a search icon)
- Release date: (empty field with a calendar icon)
- Description: (empty text area)

 At the bottom of the form, there are buttons for 'Submit' and 'Submit and Make Current'. The bottom of the screen shows a Windows taskbar with a search bar, task icons, and system tray information including '34°C Light rain', 'ENG', and '14:25 25-06-2025'.

## Creating a Table :

### 1. Navigate to Table Creation

1. Go to Application Navigator (left panel).
2. Type and select:  
System Definition > Tables
3. Click the "New" button.



### 2. Fill Table Details

Field	Description
Label	Enter a name like Certificate Request
Name	Auto-generated (u_certificate_request)
Application	Choose the correct app scope (e.g., My Educational App)
Extends Table	Choose if it should inherit from another table (e.g., Task or None)



If you're creating a record that has workflows/tasks, extend from Task table.

### 4. Add Columns (Fields)

5. After submitting the table:

1. You'll be redirected to the Table Configuration page.
2. Under the Columns related list, click New to add fields.

| Example Fields for a Certificate Request Table |

| Student Name | String |

| Student ID | String |

| Certificate Type | Choice |

| Request Date | Date/Time |


| Status | Choice (Pending, Approved, Rejected) |

| Comments | Journal Input |

 You can double-click on existing fields to edit them inline.

 4. Design the Form View (Optional)

1. Click Form Layout or Form Designer.
2. Drag and drop fields to organize how they appear to the user.

 5. Set Access Controls (Optional but Important)

Navigate to:

System Security > Access Control (ACL)

Set who can read, write, or create records on the new table.

 6. Add Table to Application Menu (Optional)

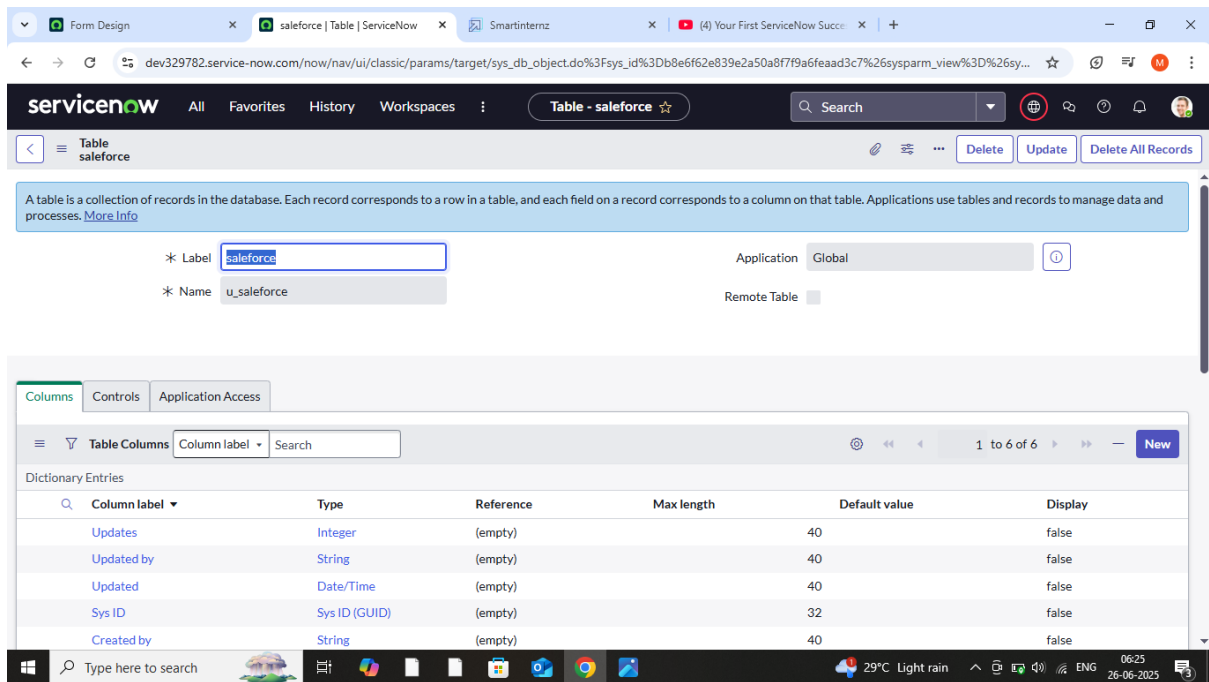
1. Go to your application in Studio.
2. Click on Navigation Menu.
3. Add a new Module:
  - Name: Certificate Requests
  - Link to: Table → select your new table

This allows users to access the table easily from the left panel.

 Now You're Done!

You now have a fully functional custom table in ServiceNow. You can:

- Build forms and lists
- Attach workflows
- Capture and track data
- Link it to service catalog items



## Form Layout :

### What is Form Layout?

The Form Layout defines how fields are arranged when a record is opened in a form view. It lets you:

- Add/remove fields
- Organize fields into sections
- Reorder fields for better user experience



### 1. Access Form Layout

#### Option A: From the Table

1. Navigate to System Definition > Tables
2. Find and open your table (e.g., u\_certificate\_request)
3. Scroll down to the Related Links section
4. Click "Form Layout"

#### Option B: From a Record

1. Open any record from your table.
2. Right-click the form header (top bar) → Configure > Form Layout

### 2. Customize Fields

The Form Layout page has three main columns:

Section	Purpose
Available	Unused fields you can add
Selected	Fields currently on the form
Create New Field	Add a brand new column to the table

#### Add a New Field:

1. Enter field name (e.g., Reason for Request)
2. Choose field type (String, Choice, Date/Time, etc.)
3. Click Add → then Save

#### Reorder Fields:

Drag fields up or down in the Selected list.

### 3. Create Sections (Optional)

You can group fields visually on the form using Sections.

1. In the Selected list, click "Add Section"
2. Name the section (e.g., Student Information, Request Details)
3. Drag related fields under the new section

#### 4. Preview the Form

1. After saving, go back to your record
2. You'll see the updated form layout with fields and sections organized.

#### 5. Tips & Best Practices

- Group related fields into sections for clarity.
- Place important fields (like status, requester) near the top.
- Keep form layout consistent across related tables.
- Use Form Designer for drag-and-drop layout (GUI alternative).

#### Want to Use Form Designer?

You can also use the drag-and-drop visual tool:

1. Open a record.
2. Right-click the header → Configure > Form Designer.
3. Drag fields, add containers/sections visually.

## Form Design :

### What is Form Designer?

Form Designer is a visual interface that lets you:

- Add/remove fields
- Rearrange field layout using columns and sections
- Add annotations, tabs, and containers
- Visually see how the form will appear to users

### ✓ 1. Accessing Form Designer

Steps:

1. Open any record from your custom table (e.g., Certificate Request).
2. Right-click the form header (top bar of the record).
3. Select Configure > Form Design.

### ✕ 2. Understanding the Interface

Section	Description
Field Navigator (left)	Shows existing fields and options to add new ones
Form Layout (center)	Drag-and-drop interface for arranging fields
Field Properties (right)	Set label, read-only, mandatory, etc.

### 3. Building the Form

#### Add Fields

- Drag fields from the left panel onto the form layout.
- You can drag fields into:
  - 1-column, 2-column, or tabbed sections

#### Create Sections

- Drag a Section component into the layout (e.g., "Student Details")
- Then add relevant fields inside it.

#### Add Tabs (Optional)

- Add a Tab layout to organize fields into multiple views (e.g., "Request Info", "Approvals")

#### Create New Fields

- In the left panel, click Create New Field
- Enter:
  - Name: request\_reason
  - Label: "Reason for Request"
  - Type: Choice, String, Date, etc.
- Click Create

### 4. Preview & Save

- After arranging all fields and sections, click Save (top right).
- Go back to your record to see the updated form layout live.

### 5. Best Practices

Tip	Why It's Helpful
Use sections to group related fields	Improves readability
Place important fields near the top	Saves time for users
Avoid long single-column forms	Split into columns/tabs for cleaner design
Use clear labels	Helps users understand inputs

### 🌟 Example: Student Certificate Request Form Layout

Section: Student Info

Student Name (String)

Student ID (String)

Course (Choice)

Section: Request Details

Certificate Type (Choice)

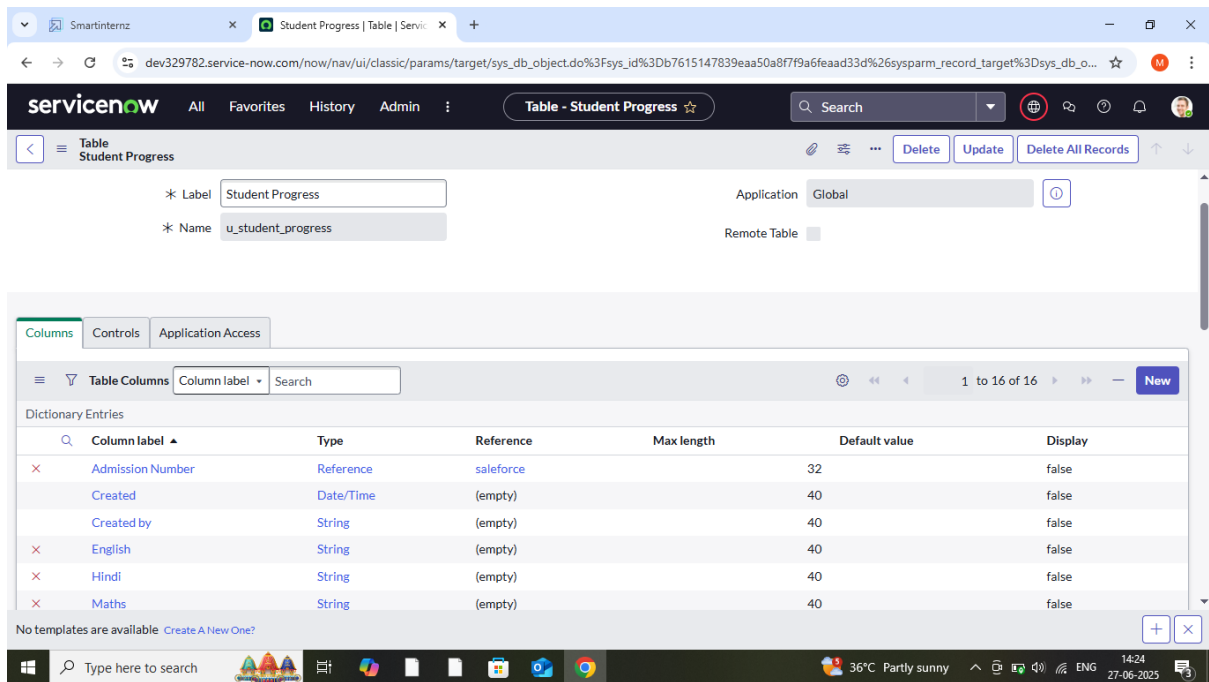
Request Date (Date/Time)

Reason for Request  
(String)

Section: Status Tracking

Status (Choice: Pending, Approved, Rejected)

Approver Comments (Journal Input)



## Number Maintenance :

### How to Create or Modify Number Maintenance

#### Steps:

1. In the Application Navigator, search for:  
System Definition > Number Maintenance
2. Click New to create a new rule or click an existing one to modify.

#### To Create a New Number Format

Field	Description
Table	Select the custom table (e.g., u_certificate_request)
Prefix	Enter the prefix (e.g., CERT)
Number	Starting number (e.g., 0001)
Number of Digits	Total number length (e.g., 5 for CERT00001)

#### Example:

- Table: Certificate Request [u\_certificate\_request]
- Prefix: CERT
- Next Number: 00001
- Result: CERT00001, CERT00002, ...



💡 You can customize this for each table in your educational system (e.g., LEAVE00001 for leave requests).

---

### How It Works

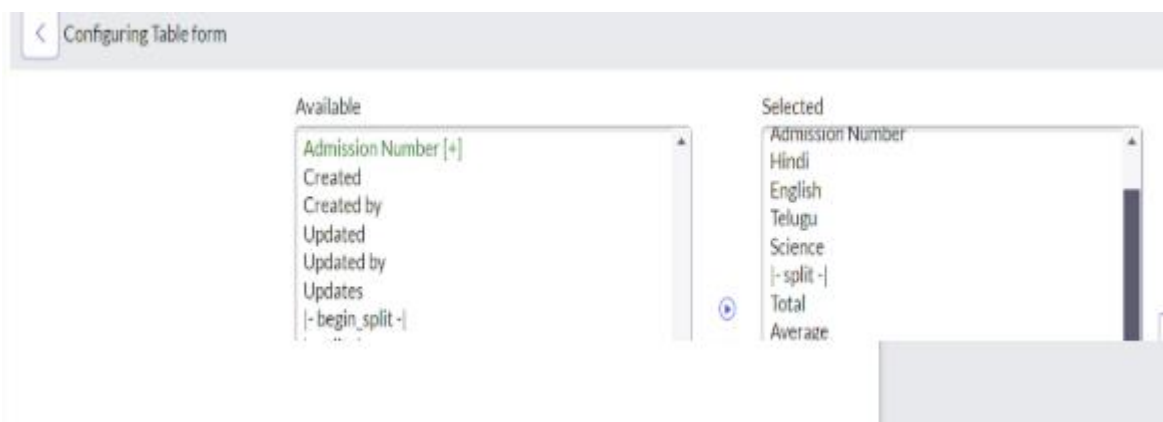
- When a new record is inserted in the selected table, ServiceNow will automatically assign the next number using the format you've set.
- 

### Important Notes

- The field where the number is stored should have:
    - Type: String
    - Default Value: Use getNextObjNumber(), or let ServiceNow handle it through the prefix rule.
  - You usually don't need to manually insert the number—it is generated on Insert.
  - Be careful not to modify numbering in a production environment without backups.
- 

### To Test It

1. Go to the table (e.g., Certificate Request).
2. Click New to create a record.
3. Submit the form — you should see the auto-generated number in the designated field.



Process Flow :

## Objective

To automate and streamline request handling for students, faculty, and staff through a ServiceNow-based system using structured workflows.

---



### Generalized Process Flow

css

CopyEdit

[1] Request Initiation



[2] Request Submission via Service Portal



[3] Automatic Categorization & Assignment



[4] Approval Workflow (if required)



[5] Fulfillment / Action Taken



[6] Notification to Requester



[7] Closure and Feedback Collection

---



### Example 1: Student Certificate Request Flow

Step	Description
1. Student Logs In	Accesses ServiceNow Student Portal
2. Submits Request	Chooses "Certificate Request" item and fills form
3. System Captures Request	Auto-assigns unique number (via Number Maintenance)
4. Auto/Manual Routing	Sent to Registrar or Academic Office
5. Registrar Reviews	Approves or rejects based on eligibility
6. Notification	Student gets email/SMS about approval
7. Certificate Issued	Fulfilled and marked as "Closed"

Step	Description
8. Feedback	Optional survey or satisfaction form

---

#### Example 2: Faculty Leave Application Flow

Step	Description
1. Faculty Logs In	Accesses Service Portal or HR module
2. Submits Leave Request	Selects dates, reason, and type of leave
3. Routed to HOD	Approval required via Flow Designer
4. HOD Approves/Rejects	Status changes accordingly
5. Notification Sent	To faculty and HR department
6. Leave Updated in System	HR record updated

---

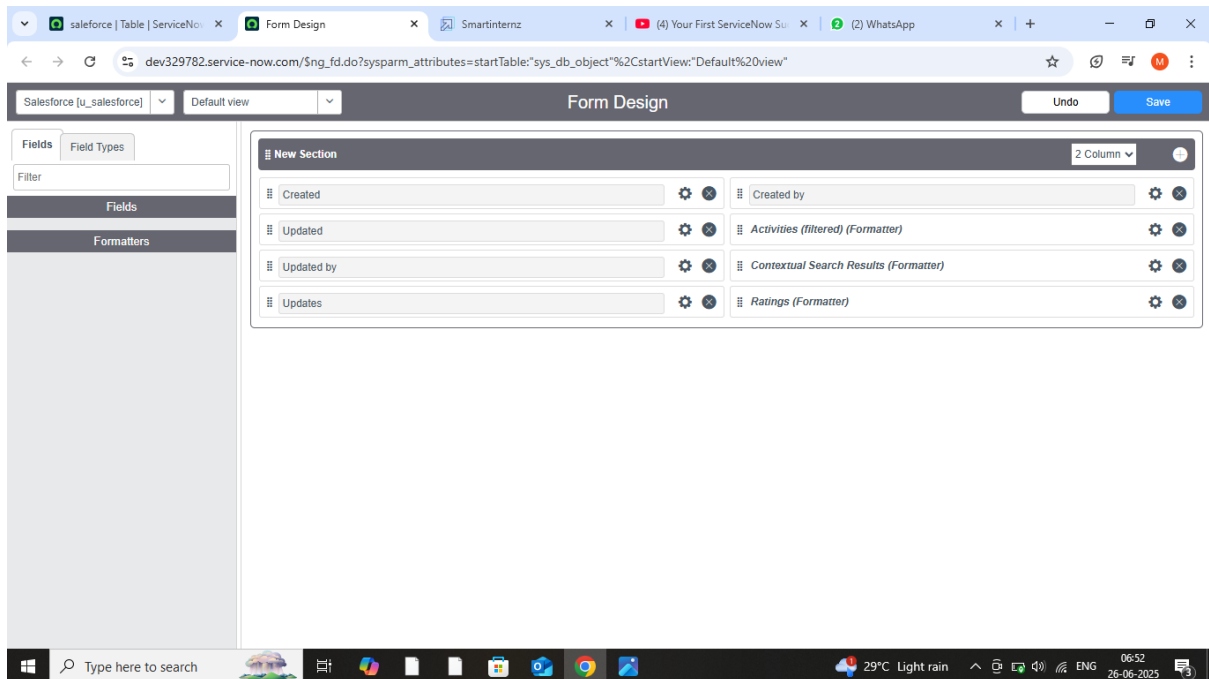
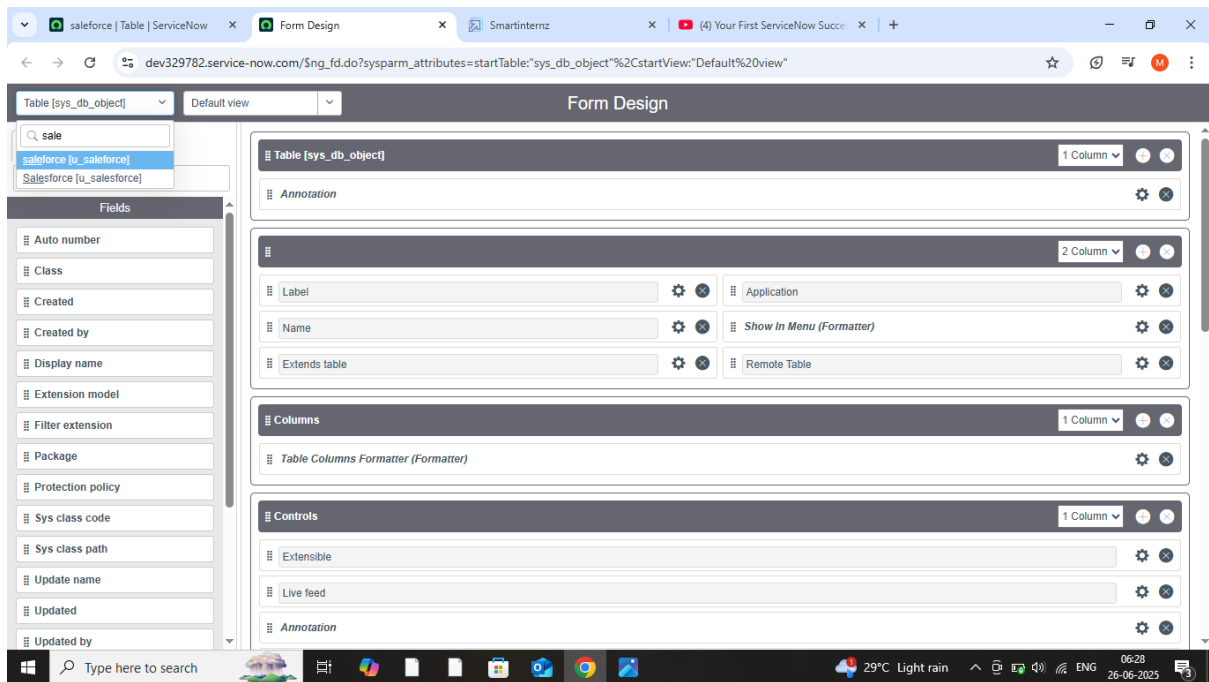
#### Example 3: IT Helpdesk Incident Flow

Step	Description
1. User Reports Issue	From portal or mobile (e.g., "Projector not working")
2. Incident Logged	Categorized as IT Incident
3. Auto-assigned	To IT Support Group
4. Technician Resolves Issue	Adds work notes and resolution
5. User Informed	Via email notification
6. Closed	User confirms closure or reopens if unresolved

---

#### Tools Used in the Flow

Tool	Role
Service Catalog	Request Forms (Certificate, Leave, etc.)
Flow Designer	Automates routing, approvals, notifications
Notification Module	Sends real-time updates
Email/SMS Integration	Alerts and status changes
Reports/Dashboards	Track request types, SLAs, delays



## Client Script :

Types of Client Scripts in ServiceNow:

1. onLoad  
Triggered when a form loads.  
☒ Use to set default values, hide fields, or show messages.
2. onChange  
Triggered when a specific field's value changes.  
☒ Use to make fields visible, editable, or required based on other field values.
3. onSubmit  
Triggered when the user clicks *Submit*, *Update*, or *Save*.  
☒ Use to validate data before it is sent to the server.
4. onCellEdit  
Triggered when a cell is edited in a list.  
☒ Rare, used in list editing scenarios.

---

◆ Basic Syntax Example: (onChange Script)

javascript

CopyEdit

```
function onChange(control, oldValue, newValue, isLoading) {  
    if (isLoading || newValue == '') {  
        return;  
    }  
  
    if (newValue == 'true') {  
        g_form.setVisible('other_field', true);  
    } else {  
        g_form.setVisible('other_field', false);  
    }  
}
```

---

◆ Common APIs Used:

- g\_form.setValue(fieldName, value) – Set a field value.
- g\_form.getValue(fieldName) – Get the value of a field.
- g\_form.setVisible(fieldName, boolean) – Show or hide a field.
- g\_form.setMandatory(fieldName, boolean) – Make a field mandatory or not.

- `g_form.showFieldMsg(fieldName, message, type)` – Show a message near a field.

## ◆ How to Create a Client Script:

1. Go to ServiceNow Studio or Application Navigator.
2. Navigate to:  
System Definition → Client Scripts
3. Click New.
4. Choose the form/table where it should run.
5. Define:
  - Type: onLoad, onChange, etc.
  - Field Name (for onChange)
  - Script

The screenshot shows the ServiceNow Studio interface for configuring a table named 'saleforce'. The 'Columns' tab is active, displaying a table of dictionary entries. The table has columns for Column label, Type, Reference, Max length, Default value, and Display. The entries listed are Updates, Updated by, Updated, Sys ID, and Created by.

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

