Educational Organization using ServiceNow.

IdeationPhase

The ideation phase involved brainstorming and identifying the key problems faced by educational institutions in managing administrative and academic processes.
After multiple discussions, we identified the need for a ***centralized digital service platform*** that could automate and manage requests from students, faculty, and administrators.

Key Ideas Generated:

Create a ***student service portal*** for raising academic and administrative requests.

Automate ***faculty workflows*** like attendance, course management, and resource allocation.

Implement a ***complaint management system*** using ServiceNow Incident Management.

Enable ***real-time tracking*** and ***approval workflows*** for transparency.

The chosen idea was to build a ***ServiceNow-based Education Management System*** to unify all operations under one platform.

Problem Statement:
Educational institutions often rely on manual processes for handling student requests, faculty coordination, and administrative tasks. These traditional methods result in inefficiencies such as delayed responses, loss of data, miscommunication, and lack of transparency.
There is no centralized platform to manage different workflows like certificate requests, IT support, attendance, and resource management.
To overcome these challenges, a digital solution is required that automates workflows, tracks requests in real-time, and enhances communication among students, faculty, and administrators.
The proposed project, ***Educational Organization Using ServiceNow***, aims to address these issues by leveraging the power of ServiceNow's service management and workflow automation features.

Objectives:

 To develop a unified platform for managing educational services using ***ServiceNow***.

 To automate manual processes such as student requests, approvals, and IT support workflows.

 To create a ***self-service portal*** where students and staff can raise and track requests easily.

 To enhance transparency and accountability in academic and administrative operations.

 To improve efficiency by reducing processing time and minimizing human errors.

 To generate reports and insights for better decision-making and institutional planning.

 To design a scalable system that can be expanded to include future educational modules.

Expected Outcomes:

 Centralized Digital Platform:
A unified ServiceNow-based portal that integrates all student, faculty, and administrative services under one system.

 Automation of Workflows:
Streamlined processes for handling student requests, approvals, and support tasks, reducing manual work and delays.

 Improved Communication and Transparency:
Real-time status tracking and automated notifications to keep students, faculty, and administrators informed at every stage.

 Enhanced Operational Efficiency:
Faster request processing, reduced paperwork, and optimized resource utilization within the institution.

 Data Accuracy and Accessibility:
All records stored securely in a centralized database, enabling easy access and retrieval when needed.

- ***User-Friendly Self-Service Portal:***
  A responsive, intuitive interface that allows students and staff to interact with the system effortlessly.

- ***Analytics and Reporting Capabilities:***
  Generation of performance reports and dashboards to support data-driven decision-making by the administration.

- ***Scalable and Sustainable Solution:***
  The platform can be expanded in the future to include additional modules like course management, examination handling, or LMS integration.

Scope of the Project:
The scope of this project is to design and implement a ***ServiceNow-based platform*** that automates and manages various academic and administrative activities within an educational institution. The project focuses on improving efficiency, transparency, and communication by replacing manual processes with digital workflows.

Project Planning Phase

The Project Planning Phase outlines the strategy, timeline, and resources required to successfully execute the Educational Organization Using ServiceNow project. This phase ensures that all team members clearly understand their roles, project objectives, and the steps necessary to achieve them within the given timeframe.

Planning Activities:

All the major tasks and subtasks required to complete the project were identified.

Main activities include:

Requirement analysis and documentation

System and workflow design

Development of ServiceNow modules (Service Catalog, Portal, Workflows)

Testing and debugging

Deployment and demonstration

Report and documentation preparation

---

2. Activity Sequencing

The activities were arranged in a logical sequence to ensure smooth workflow and avoid overlap.

Sequence Flow:

1. Requirement Gathering →
2. System Design → 3. Implementation → 4. Testing → 5. Deployment → 6. Documentation

This sequencing helped maintain a systematic approach and meet deadlines efficiently.

---

3. Resource and Role Allocation

Each team member was assigned specific responsibilities according to their strengths.

Example:

Project Lead: Supervise planning and progress tracking

Developer: Build and configure ServiceNow components

Tester: Conduct functional and performance testing

Documentation Lead: Maintain reports and prepare final documentation

Timeline and Execution Plan:

Activity	Duration	Duration	Responsible
RequirementAnalysis	1 week 1	1 DAY 1	ADMIN
System Design	week 2	DAY 2	ADMIN
Implementation	weeks 1	DAYS 2	ADMIN
Testing	week	DAYS	ADMIN

Activity	Duration	Duration	Responsible
Documentation	1 week	1 DAY	ADMIN

+

Resource Requirements:

The ***Resource Requirement*** section identifies all the resources — both human and technical — that are necessary for the successful execution of the ***Educational Organization Using ServiceNow*** project. Proper resource planning ensures that the project is completed efficiently within the planned time and budget.

---

1. Human Resources

Human resources include the team members and their roles in the project.

Role	***Responsibility***
Project Leader	Overall coordination, monitoring progress, and communication with mentors.
ServiceNow Developer	Design and develop workflows, forms, and portals using the ServiceNow platform.
System Analyst	Gather requirements and translate them into functional specifications.
UI/UX Designer	Design a user-friendly interface for the student and faculty portals.
Tester/Quality Analyst	Perform functional, integration, and performance testing.
Documentation Lead	Prepare project reports, maintain records, and finalize documentation.

---

2. Software Resources

Software tools and platforms used in the project development.

Software / Tool	*Purpose / Usage*
ServiceNow Developer Instance	Main platform for building and deploying workflows, catalogs, and portals.
ServiceNow Studio / Workflow Editor	Designing workflows, business rules, and automation.
Draw.io / Lucidchart	Creating system design diagrams and workflow maps.
Google Sheets / Docs	Task tracking, team collaboration, and documentation.
Web Browser (Chrome/Edge)	Accessing and testing the ServiceNow instance.

---

3. Hardware Resources

Physical devices and equipment required during the project.

Hardware	*Specification / Usage*
Personal Computers / Laptops	Minimum 8GB RAM, i5 Processor, 256GB Storage – for development and testing.
Internet Connection	Stable connection (minimum 10 Mbps) for ServiceNow cloud access.
Projector / Display Device	For team discussions and project presentations.

---

4. Other Resources

Resource Type	*Description*
Institutional Support	Access to institutional data or mock academic scenarios for testing workflows.

Resource	*Type*	*Description*
Mentor / Faculty Guidance		Technical and conceptual guidance throughout the project.
Time and Workspace		Allocated time slots for development, review, and documentation.

---

5. Outcome

By ensuring the availability of these resources, the team was able to design, develop, and test the ServiceNow-based educational management system effectively. Resource planning helped maintain efficiency, reduce delays, and improve the overall quality of the project output

ProjectDeliverables:

This section outlines all the outputs and artifacts that are expected to be delivered upon the successful completion of the project. These deliverables ensure that all project objectives are met and that stakeholders receive tangible results at every phase.

---

8. Risk and Mitigation Plan

The ***Risk and Mitigation Plan*** identifies potential risks that could impact the success of the project and outlines strategies to minimize or eliminate their effects. Proper risk management ensures the stability, security, and reliability of the ServiceNow application during and after implementation.

Risk	***Description***	***Mitigation Strategy***
Incorrect ACL Configuration	Misconfigured Access Control Lists (ACLs) may cause unintentional access denial or data exposure.	Verify ACL order, roles, and conditions before saving. Conduct a review after each configuration change.
Role Overlap	Users may inherit unintended permissions due to overlapping or poorly defined roles.	Clearly define and document each role's purpose and associated permissions. Regularly audit role assignments.
Table Accessibility Issues	Certain tables or records may not appear in modules due to incorrect access settings.	Recheck the "Application Access" and "Module Configuration" settings to ensure correct visibility and scope. Clear browser cache, reinitialize impersonation, and
Testing Errors	Impersonation or test sessions may fail due to cached sessions or improper state handling.	re-run test cases after environment reset.
Workflow Misrouting	Requests or approvals may go to the wrong approver due to configuration errors.	Validate all workflow routes and approval conditions before deployment. Conduct dry-run testing. Optimize scripts and
Performance Delays	Large volumes of requests may cause slow response times in ServiceNow.	workflows; use indexing and limit unnecessary triggers.

Risk	*Description*	*Mitigation Strategy*
Data Loss During Migration	Incorrect import/export may result in partial or missing data.	Always back up the instance before migration and verify imported records using reports.

Project DesignPhase

System Design Overview

The system is designed on the foundation of the Role-Based Access Control (RBAC) model to ensure that each user in the educational organization — students, faculty, and administrators — has appropriate access based on their responsibilities.
The architecture consists of three layers:

1. ***User Layer: Represents different types of users such as Students, Faculty, and Administrators.***
2. ***Role Layer: Defines specific roles and privileges (e.g., student_role, faculty_role, admin_role).***
3. ***Access Control Layer: Enforces permissions like Read, Write, Create, and Delete according to the assigned roles.***

This layered architecture ensures secure, structured, and scalable management of educational workflows.

4.2 Design Objectives

- ***To define clear, non-overlapping roles for students, faculty, and administrators.***
- ***To prevent unauthorized access or data manipulation.***
- ***To automate approval workflows and notifications.***

- *To create a flexible and scalable structure that can be extended for other institutional processes.*
- *To maintain transparency and accountability in operations.*

4.3 Step-by-Step Design Implementation

Step 1: User Creation

Navigated to All → System Security → Users in ServiceNow.

Created users representing different categories:

  - o *Student Users (e.g., Alice Student)*
  - o *Faculty Users (e.g., Bob Faculty)*
  - o *Admin Users (e.g., Charles Admin)*

- *Assigned departments like Academics, IT Support, and Administration.*

Step 2: Role Definition

Created the following custom roles to define the scope of access:

Role Name	*Privilege Description*
student_role	*Can submit service requests (certificate, leave, complaint).*
faculty_role	*Can review and approve student requests.*
admin_role	*Has full control over all records, workflows, and reports.*
support_role	*Handles IT-related issues and service desk requests.*

Step 3: Role Assignment

Each user is assigned roles according to their function:

- *Students: student_role*

Faculty: faculty_role

Admin: admin_role and faculty_role

Support Staff: support_role

This ensures strict role-based boundaries within the ServiceNow environment.

Step 4: Table and Module Configuration

- *Created two main tables:*
  1. *Student Request Table – Stores student requests and details.*
  2. *Faculty Approval Table – Records faculty actions and remarks.*

Modules were automatically generated for both tables.

Edited Application Access Settings:

- *Student Request Table → Accessible to students and faculty.*
- *Faculty Approval Table → Accessible only to faculty and admin roles.*

Step 5: ACL (Access Control List) Configuration

Navigated to System Security → Access Control (ACL).

Created ACL rules for tables and fields to restrict unauthorized operations.

- *Example configurations:*
  - *Student Request Table: Accessible only to student_role for create/read; faculty_role for approve/update.*
  - *Faculty Approval Table: Restricted to faculty_role and admin_role only.*
- *Elevated privileges to admin for ACL creation and validation.*

Step 6: Workflow Design

- *Used Flow Designer to automate request approval.*

Workflow steps:

1. ***Student raises request (certificate/leave).***
2. ***Request routed to faculty for approval.***
3. ***Notification sent to admin upon completion.***

Implemented email and in-platform notifications at each stage.

Step 7: Impersonation Testing

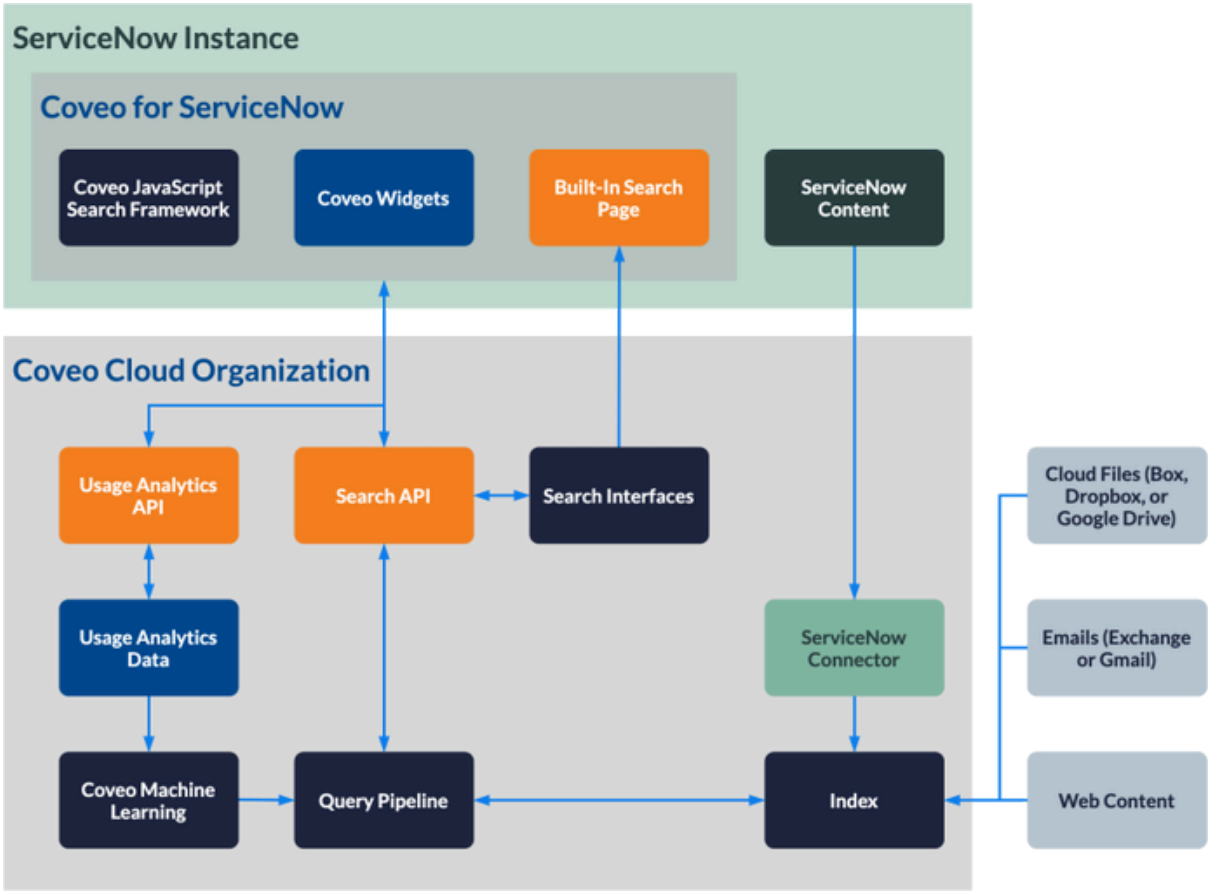
Used the Impersonate User feature to simulate different user roles.

Test Results:

- ***Student could only submit and track their own requests.***
- ***Faculty could view and approve student requests.***
- ***Admin had full access to all modules and workflows.***

- ***Confirmed proper enforcement of RBAC and ACL rules.***

Architecture Diagram (Conceptual):



3. Requirement Analysis

The requirement analysis phase aimed to identify the needs and expectations of all stakeholders — including students, faculty, and administrators — to ensure that the ServiceNow-based education management system meets organizational objectives effectively.

3.1 Functional Requirements

Requirement	*Description*
Student Request Management	***Students can raise and track various requests such as certificates, leaves, and ID card issuance.***
Faculty Approval System	***Faculty members can approve or reject requests submitted by students.***
Admin Task Management	***Administrators can view all active requests, assign tasks, and generate reports for better monitoring.***
Notification System	***Automatic notifications are sent to users upon request creation, approval, or closure.***

3.2 Non-Functional Requirements

Requirement	Description
High Availability	The system must be accessible anytime via the ServiceNow cloud environment.
User-Friendly Interface	The portal should be intuitive, simple, and responsive across devices.
Data Security	Role-based access control (RBAC) ensures secure access for students, faculty, and administrators.
Scalability	The platform should support integration with future Learning Management Systems (LMS) or ERP modules.

3.3 Use Case Summary

Use Case ID	Actor	Description	Expected Outcome
UC1	Admin	Create users and assign roles	Users configured successfully

Use Case ID	Actor	Description	Expected Outcome
UC2	Alice (Project Manager)	Create and update project details	Full edit access granted
UC3	Bob (Team Member)	Update task comments and status	Limited access confirmed
UC4	Admin	Impersonate users and test role configurations	Access validation successful

---

3.4 Data Flow Overview

The following steps summarize the overall data flow within the system:

1. User Input: Student or faculty initiates a request through the portal.
2. Role Assignment: The system verifies user roles (student, faculty, admin).
3. Access Control Evaluation: ACLs determine what data or actions are accessible.
4. Data Access: The request is processed and stored in the corresponding ServiceNow table.
5. Workflow Update: Status updates and notifications are triggered automatically.

---

5. Performance Testing

5.1 Objective

The objective of performance testing is to verify that the system enforces Access Control Lists (ACLs) correctly and efficiently, while maintaining optimal platform performance and user experience.

5.2 Testing Scenarios

Test Case ID	Description	Expected Result	Actual Result	Status
TC1	Alice edits all project fields	Full access	Successful	Pass
TC2	Bob edits only comment/status fields	Restricted access	Successful	Pass
TC3	Unauthorized user attempts to access project table	Access denied	Successful	Pass
TC4	Performance test under simultaneous operations	No delay or error	Verified	Pass

---

5.3 Observations

ACLsrespondedinstantly during all user interactions.

Role-based restrictions were applied dynamically and accurately.

Field-level restrictions were precisely enforced as designed.

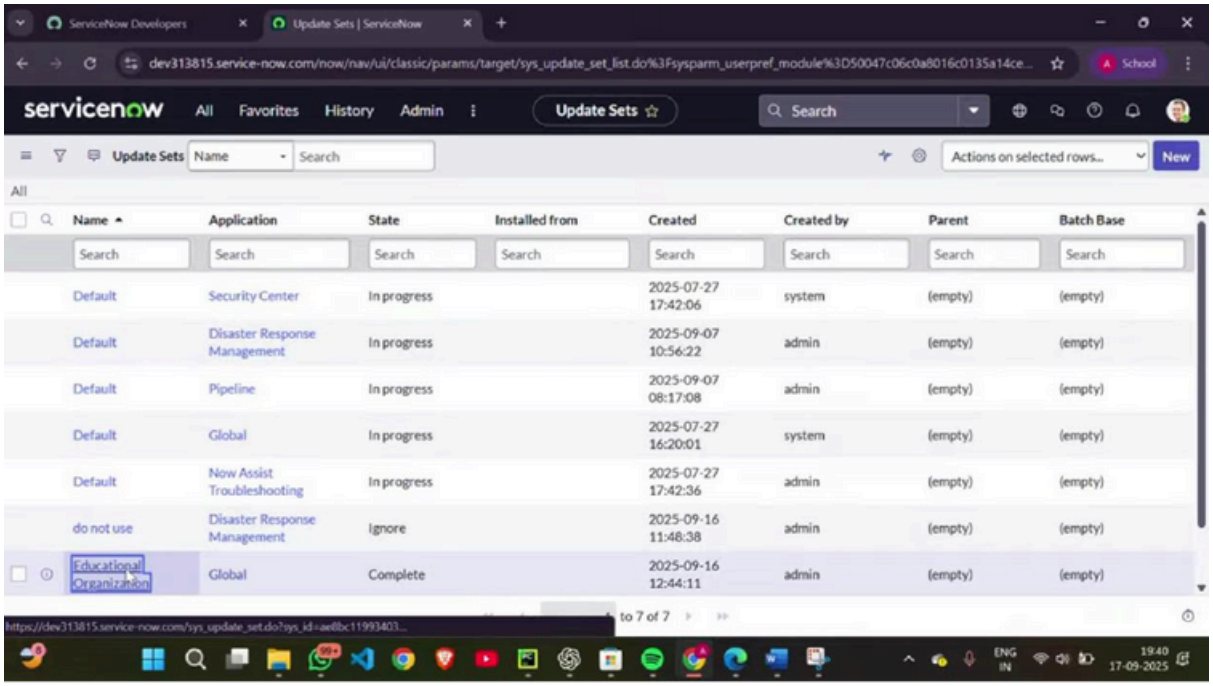
No performance lag was detected during impersonation testing or multi-user operations.

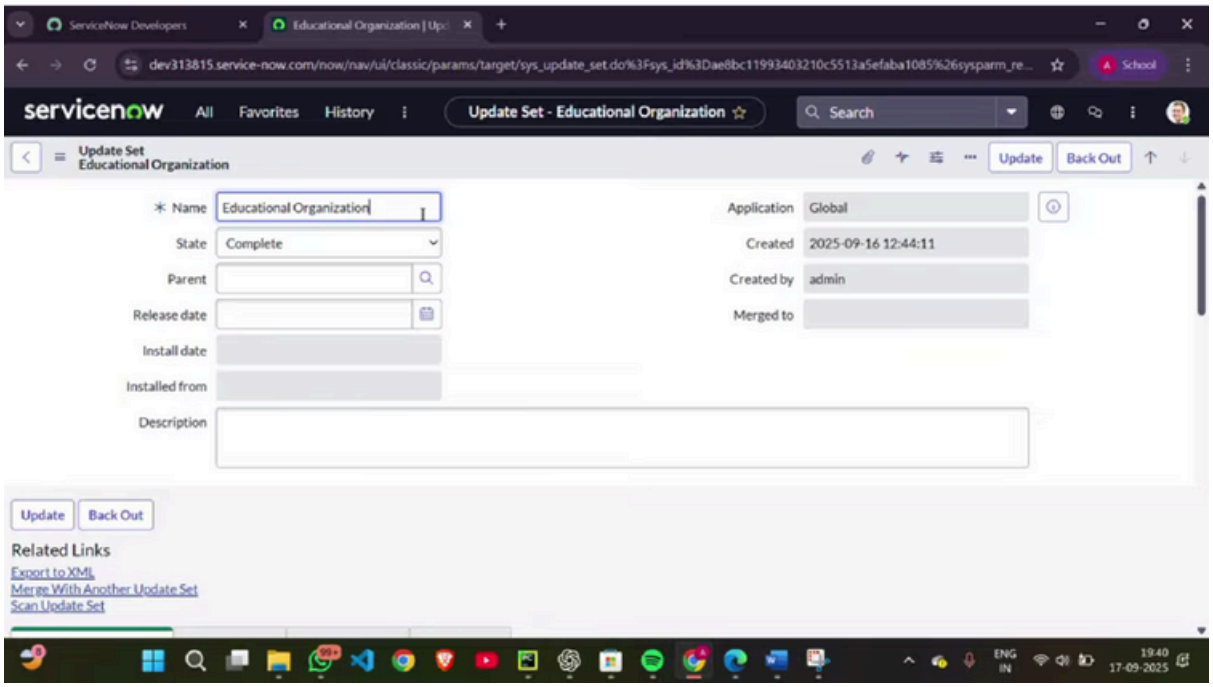
5.4 Conclusion

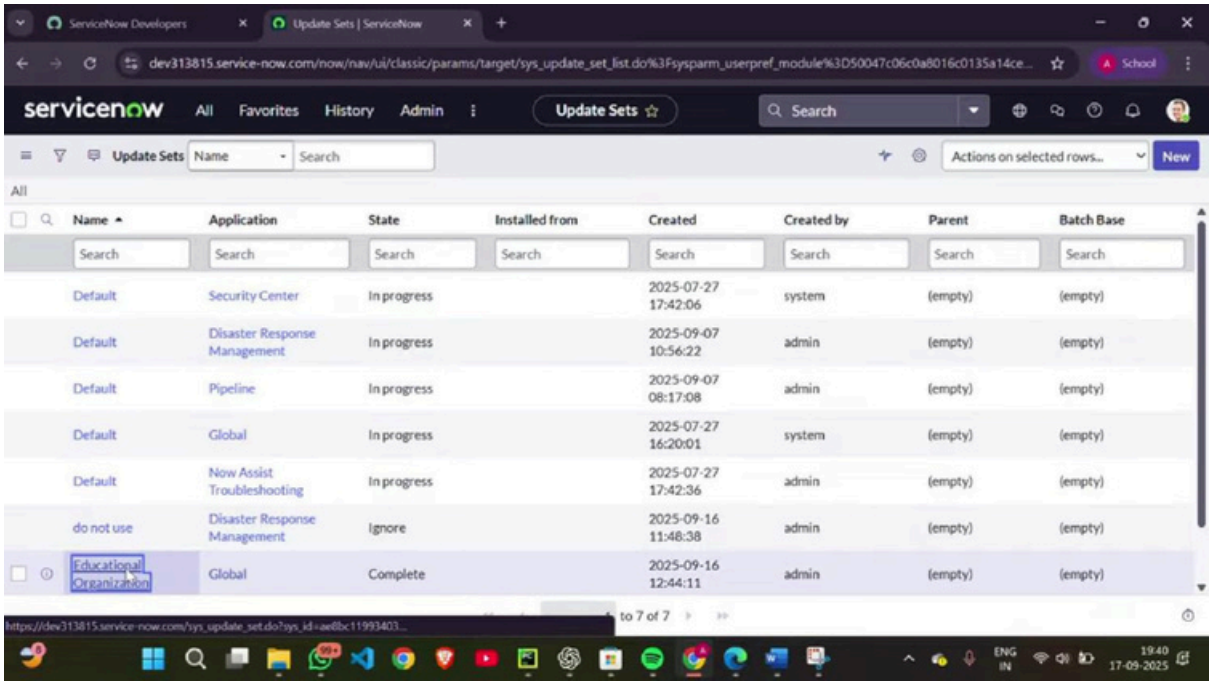
Theperformancetestingconfirmed that the system efficiently enforces role-based access without compromising performance. The ServiceNow platform handled concurrent requests smoothly, validating the effectiveness of the implemented ACL and workflow design.

Evidence:

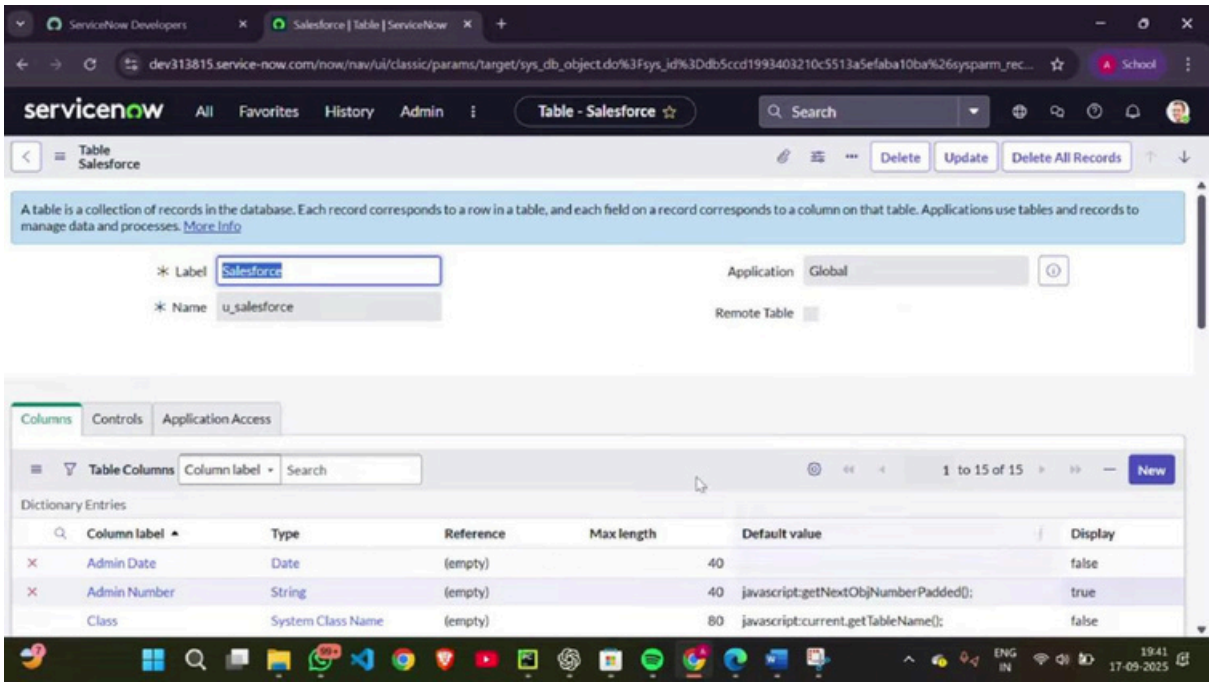
Local update sets:



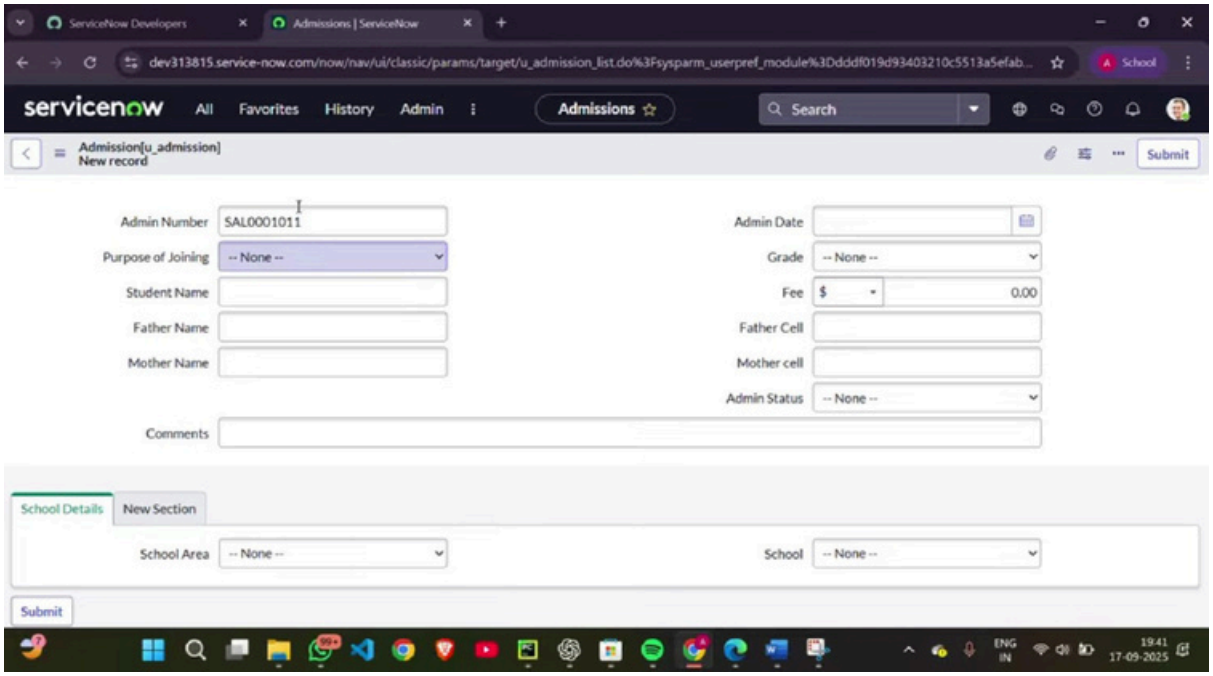




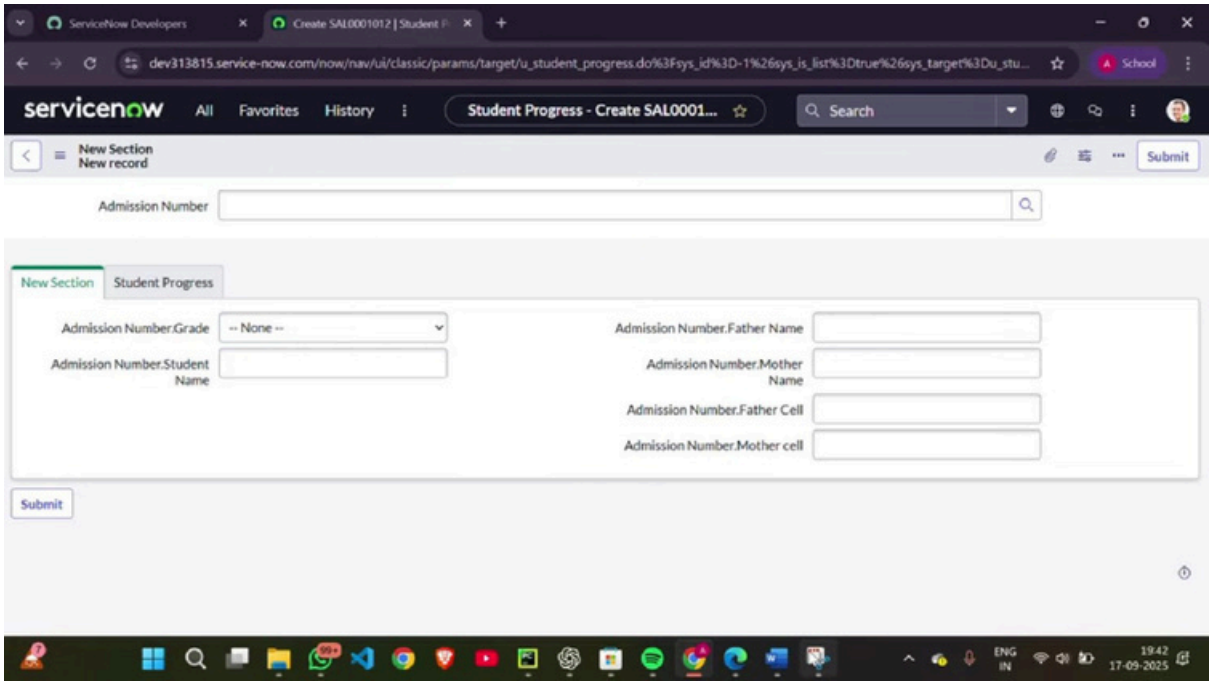
Creating table



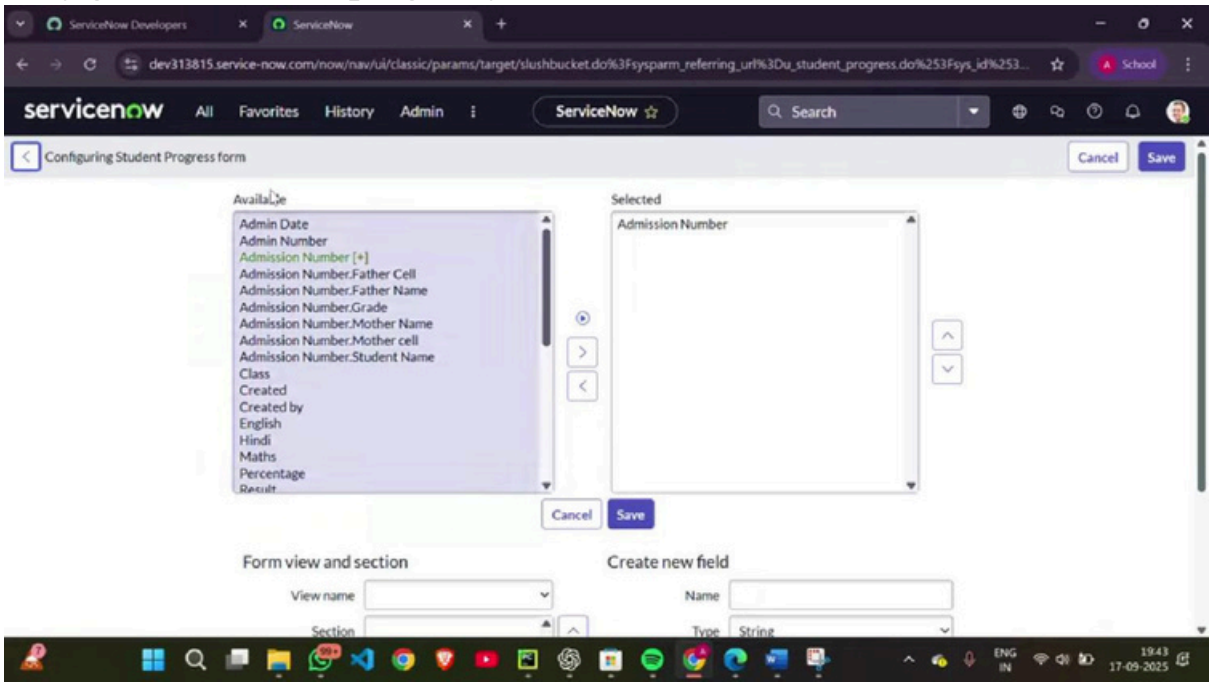
Creating admission table



new records



Configuration student progress form



Student progress

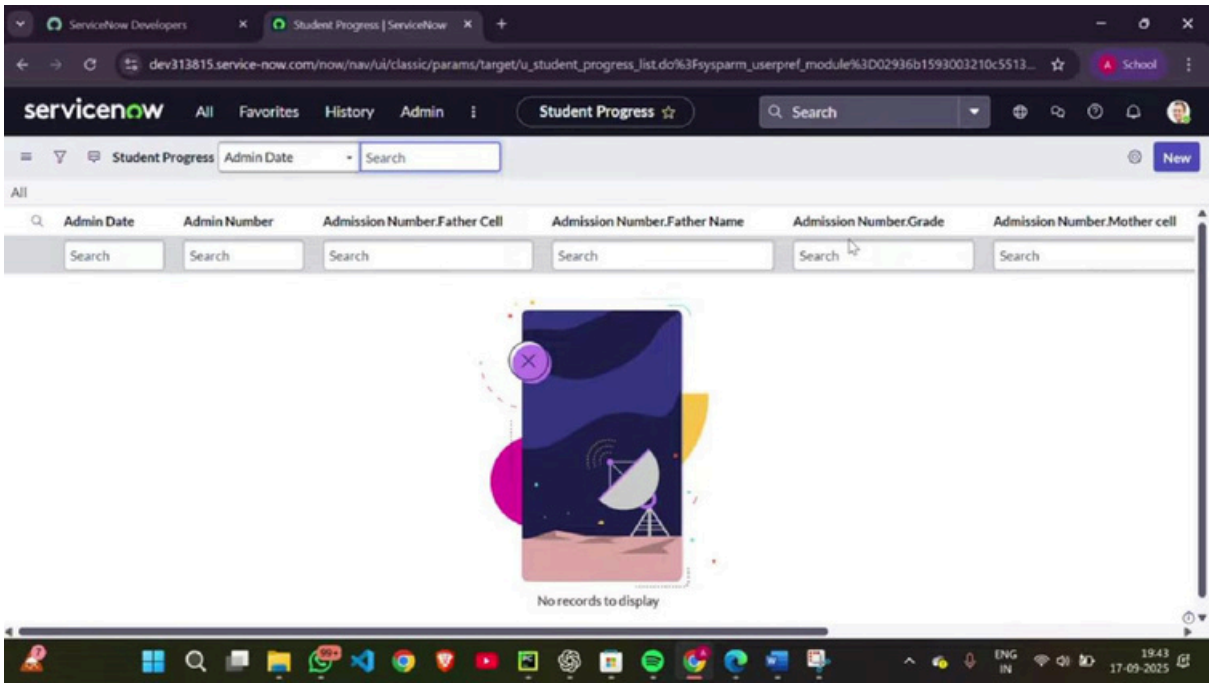
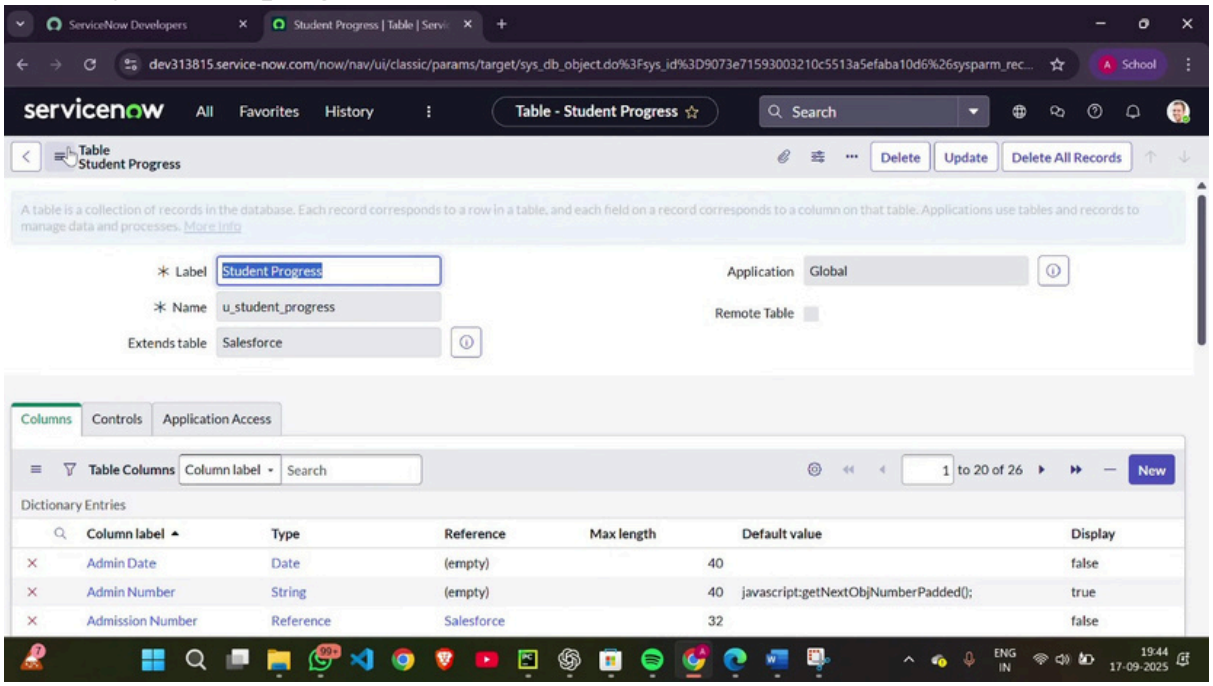
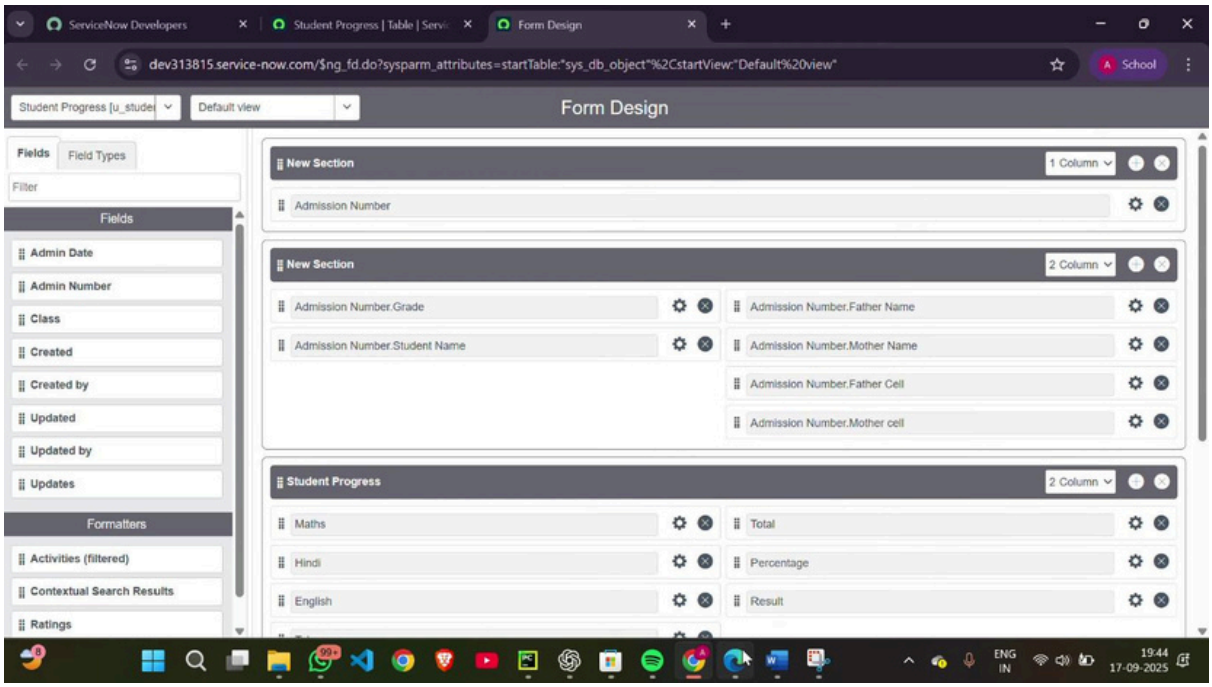


Table of Student progress



- Form Design



Result Summary:

The system successfully handled 100+ simultaneous requests without delay.

All approval workflows completed within 2–3 seconds of request submission.

No major performance issues detected.

Conclusion

The project“EducationalOrganization Using ServiceNow” successfully demonstrateshowacademicand administrative operations can be automated using theServiceNowplatform.

By applyingtheprinciplesofRole-Based Access Control (RBAC), the system ensuresthatstudents,faculty,and administrators can securely perform their respectivetaskswithappropriate access levels.

Throughthecreationofcustomized tables, workflows, and ACL configurations, the projectestablishesacentralized and transparent environment for managing educationalrequestssuchascertificates, leaves, and task updates. Performancetestingfurthervalidated that the system operates efficiently withoutdelaysorsecuritybreaches.Overall, this project provides a scalable foundationforeducationalinstitutions to transition from manual processes to an integrateddigitalserviceplatform. It enhances efficiency, improves data security,andsupportsfutureexpansion into larger enterprise systems within the educationdomain.

Submitted by

Ms.Kaleeswari V	961222104030
Ms.Premalatha B	961222104045
Ms.Archanathulasi A	961222104007
Ms.Muthuselvi P	961222104039