



M.A.M.

College of Engineering

<http://www.mamce.org>

*DEPARTMENT OF COMPUTER SCIENCE
ENGINEERING*

TITLE: Educational Organisation Using ServiceNow

Team Ledeader: Nandhini R

Team member : Kalpana G

Kiruthiga P

Pavithra R

TABLE OF CONTENTS

1. Introduction
 - 1.1 Overview of Educational Administration
 - 1.2 Need for Automation
 - 1.3 Role of ServiceNow in Data Management
2. Existing System
 - 2.1 Manual Data Management
 - 2.2 Challenges in Current System
 - 2.3 Limitations Identified
3. Proposed System
 - 3.1 Objectives of the Proposed System
 - 3.2 Key Features
 - 3.3 Scope of Implementation
4. System Analysis
 - 4.1 Functional Requirements
 - 4.2 Non-Functional Requirements
 - 4.3 System Architecture
5. System Design
 - 5.1 System Architecture Diagram
 - 5.2 Use Case Diagram
 - 5.3 Entity Relationship Diagram (ERD)
 - 5.4 Data Flow Diagram (DFD)
 - 5.5 Table Design
 - 5.6 Design Summary
6. Module Description
 - 6.1 Admission Management Module
 - 6.2 Student Progress & Performance Module
 - 6.3 Process Flow Tracking Module
7. Methodology
 - 7.1 Table Creation

- 7.2 Form Design
- 7.3 Number Maintenance Setup
- 7.4 Process Flow Creation
- 7.5 Client Scripts Implementation
 - 7.5.1 Auto Populate Script
 - 7.5.2 Periodic Update Script
 - 7.5.3 Disable Fields Script
 - 7.5.4 Total Marks Calculation Script
 - 7.5.5 Percentage Calculation Script
 - 7.5.6 Result Evaluation Script
- 8. Results
- 9. Advantages
- 10. Disadvantage
- 11. Future Enhancement
- 12. Conclusion
- 13. References

EDUCATIONAL ORGANISATION USING SERVICE NOW

INTRODUCTION

1.1 Overview of Educational Administration

Educational administration plays a vital role in organizing and managing all aspects of an institution's activities. It encompasses student admissions, academic scheduling, faculty assignments, course management, examinations, and performance monitoring. In many traditional setups, these activities are handled manually, which increases administrative workload and reduces efficiency.

Modern educational institutions require a systematic and digital approach for better management. With advancements in technology, cloud-based platforms such as ServiceNow enable institutions to automate routine administrative tasks, store data securely, and ensure easy access for students and staff. Through such systems, educational organizations can achieve transparency, accountability, and faster decision-making.

1.2 Need for Automation

In most institutions, data related to students, faculty, and courses is maintained using spreadsheets or physical records. Managing this data manually results in several challenges such as data duplication, loss of information, and inefficiency in tracking academic progress.

Automation addresses these issues by digitizing all processes. It ensures that data is consistent, accurate, and available in real time. Automation not only reduces human effort but also provides better insights through reports and dashboards. For educational organizations, automation helps streamline admissions, attendance tracking, grading, and communication among departments.

Therefore, implementing an automated system through ServiceNow can save time, reduce operational costs, and improve the overall productivity of the institution.

1.3 Role of ServiceNow in Data Management

ServiceNow is a robust cloud-based platform widely used for enterprise automation. It offers a no-code/low-code environment to create custom applications, manage workflows, and handle large amounts of data efficiently.

In the context of educational institutions, ServiceNow can serve as a centralized platform to handle student admissions, manage course information, assign faculty, and monitor performance. The platform's in-built tools like Table Builder, Form Designer, Flow Designer, and Client Scripts make it ideal for developing automation workflows without traditional programming.

By using ServiceNow, educational institutions can achieve better data security, faster processing, and centralized management of all operations. It ensures that every activity—from admission to performance evaluation—is recorded and traceable.

2. EXISTING SYSTEM

2.1 Manual Data Management

The existing process of managing institutional data is manual. Student and faculty details are recorded in notebooks, spreadsheets, or isolated digital files. Each department maintains its own set of data, making it difficult to coordinate across departments.

This traditional method of handling data leads to inconsistent information, delayed updates, and errors during record retrieval.

Furthermore, tracking student progress manually consumes a large amount of time and effort.

2.2 Challenges in Current System

1. Data Duplication:

When different departments maintain the same student details separately, inconsistencies arise.

2. Lack of Real-Time Access:

Data updates are delayed because they depend on manual entries.

3. Poor Reporting:

Manual systems lack analytical tools to generate reports for performance tracking.

4. Time-Consuming:

Administrative staff spend significant time maintaining and verifying records.

5. Security Issues:

Manual files are prone to damage, loss, or unauthorized access.

2.3 Limitations Identified

The manual approach lacks integration and real-time collaboration. There is no centralized database where all information can be accessed securely by authorized personnel. In addition, generating customized reports and performing data analysis are extremely difficult without automated tools.

This creates an urgent need for a solution that centralizes data, provides automation, and allows access through role-based permissions.

3. PROPOSED SYSTEM

3.1 Objectives of the Proposed System

The primary goal of this project is to develop an automated educational management system using ServiceNow. The specific objectives include:

- To automate student admission and record management.

- To streamline faculty and course assignment processes.
- To generate student IDs automatically using padded numbers.
- To create approval workflows for various administrative tasks.
- To provide quick report generation and performance analytics.

3.2 Key Features

- Centralized Database: All student, course, and faculty details are stored in one place.
- Automated ID Generation: Student IDs are generated automatically using ServiceNow's "Get Next Padded Number" functionality.
- Flow Automation: Admission approvals, course assignments, and request tracking are automated.
- Custom UI Forms: User-friendly interfaces for easy data entry and updates.
- Role-Based Access Control: Different permissions for students, faculty, and administrators.
- Reporting & Dashboards: Real-time reports for performance and admission tracking.

3.3 Scope of Implementation

The system can be implemented in schools, colleges, or universities. It can handle multiple departments and thousands of students efficiently.

The scope can later be expanded by integrating other ServiceNow modules such as Human Resources, Finance, and Incident Management to form a complete educational ERP solution.

4. SYSTEM ANALYSIS

4.1 Functional Requirements

- Student record management (Add, Update, Delete, Search).
- Faculty assignment for courses.
- Admission approval workflow.
- Automatic calculation of total marks and percentage.
- Result generation and evaluation.
- Report generation and dashboard visualization.

4.2 Non-Functional Requirements

- Performance:

The system should respond quickly to user actions.

- Scalability:

It should handle large data volumes.

- Security:

Role-based access should protect sensitive data.

- Reliability:

System should ensure data consistency and backup.

- Usability:

Easy-to-use interfaces designed through Form Designer.

4.3 System Architecture

The architecture consists of three major layers:

1. Presentation Layer:

The ServiceNow user interface where users interact through forms and dashboards.

2. Application Layer:

Contains the logic for workflows, approvals, and scripts.

3. Database Layer:

Stores all institutional data in ServiceNow tables.

This multi-layered architecture ensures flexibility, maintainability, and security of the entire system.

5. System Design

System Design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specific requirements.

In this project, Educational Organisation Using ServiceNow, the system design focuses on building an efficient and automated platform for managing educational data such as admissions, student details, performance records, and staff interactions.

The goal of the design phase is to translate the requirements collected during system analysis into a blueprint that guides implementation.

5.1 System Architecture Design

The architecture of the Educational Organisation System follows a three-tier architecture model

1. User Interface Layer (Front-End)

- Provides access for users such as Admins, Teachers, and Students.

- ☐ Users interact through forms, lists, and dashboards created in ServiceNow
- ☐ The interface is designed to be user-friendly with validation to prevent data errors.

2. Application Layer (Logic Layer)

- ☐ This layer handles the business logic and workflows.
- ☐ Includes client scripts, flow designer processes, and business rules.
- ☐ Responsible for actions such as auto-calculation of marks, result evaluation, and updating status fields.

3. Database Layer (Back-End)

- ☐ Stores all system data securely within ServiceNow tables.
- ☐ Custom tables such as Student Details, Admission Records, and Performance Tracker are used.
- ☐ Relationships between tables ensure data consistency and integrity.

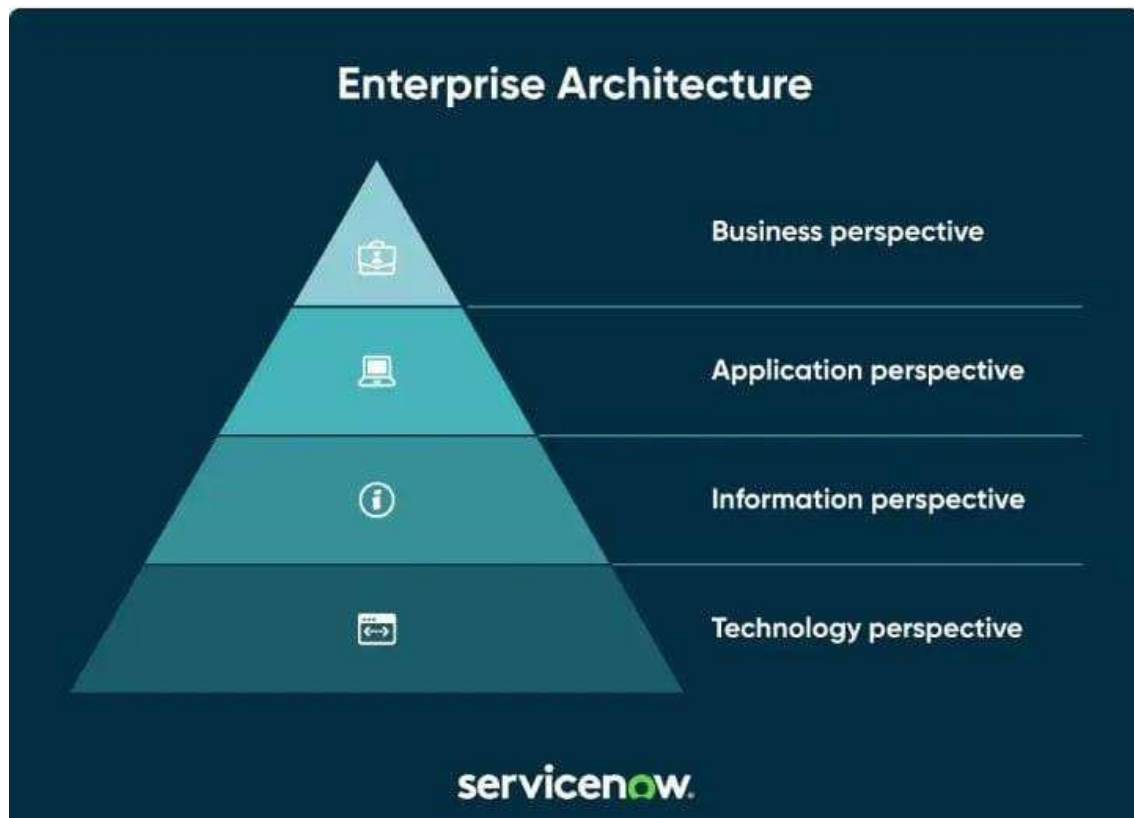
Architecture Explanation:

When a user enters data in the Admission Form, it first passes through the UI layer. The logic layer verifies and processes the input using defined scripts and workflows, then stores it in the database layer. Reports and dashboards retrieve this data back through the same layers for visualization and analysis.

5.2 System Architecture Diagram

(diagram showing User → Application Logic → Database → Output)

Figure



System Architecture of Educational Organisation Using ServiceNow

Description:

The above diagram illustrates how different users communicate with the ServiceNow platform. The Administrator can access all modules, while Teachers and Students have restricted access based on their roles. The system ensures secure communication and centralized data handling.

5.3 Use Case Diagram

The Use Case Diagram depicts the functional interactions between users (actors) and the system.

Actors:

- Administrator

- Teacher
- Student Use Cases:
 - Manage Admissions
 - Update Student Records
 - Evaluate Results
 - View Reports

Each actor interacts with the system through specific modules.

Explanation:

The Administrator manages admissions and results, the Teacher updates performance details, and the Student can view results. Each user role has distinct access levels defined through ServiceNow's role-based access control (RBAC).

5.4 Entity Relationship Diagram (ERD)

The ER Diagram defines the logical relationships between different entities in the system.

Entities:

- Student
- Admission
- Course
- Result

Relationships:

- Each Student has one Admission Record.
- Each Admission Record is linked to one Course.
- Each Course can have multiple Results related to students.

Explanation:

The ERD ensures that all tables are interconnected for efficient data retrieval. The primary keys like Student ID and Course ID uniquely identify records, maintaining referential integrity.

5.5 Data Flow Diagram (DFD)

The Data Flow Diagram represents how information moves through the system.

Level 0 DFD – Context Diagram

Shows the overall system as a single process interacting with external entities like Admin, Student, and Teacher.

Data flows include Admission Details, Performance Data, and Reports.

Level 1 DFD – Detailed View

Breaks the system into processes:

1. Admission Management
2. Student Performance Tracking
3. Result Evaluation
4. Report Generation

Explanation:

When a new admission is entered, data flows to the Admission Table. Teachers then update student performance, which triggers automated calculations through scripts. The system finally produces the result report that can be viewed by the admin and student.

Figure 5.4:

5.6 Table Design

Student Performance Table:

Field Name	Data Type	Description
Student ID	Auto Number	Unique identifier for each student
Name	String	Student full name
Department	Choice	Select department (e.g., CSE, ECE, IT)
Total Marks	Integer	Sum of all subject marks
Percentage	Decimal	Calculated percentage

Result Status	String	Displays Pass/Fail automatically

Explanation:

Each field is carefully chosen to ensure smooth automation. For example, the Percentage and Result Status fields are automatically updated using client scripts, ensuring real-time accuracy and efficiency.

5.7 Design Summary The system design ensures:

- Scalability:

Easy to add more modules in the future (e.g., Attendance Tracking).

- Security:

Only authorized users can access or modify records.

- Automation:

Scripts handle repetitive calculations and validations.

- Usability:

Simple interface for non-technical users.

This design provides a strong foundation for implementation in the ServiceNow platform, ensuring that all administrative activities within the educational organisation are streamlined, automated, and secure.

6. MODULE DESCRIPTION

6.1 Admission Management Module

This module manages the entire student admission process. When a new student record is created, the system automatically generates a unique Student ID. Approval workflows can be configured to validate each admission.

6.2 Student Progress & Performance Module

This module tracks marks, attendance, and other performance parameters. Client scripts are used to calculate total marks, percentage, and results automatically. Reports can be generated for individual or batch-level performance.

6.3 Process Flow Tracking Module

The Flow Designer tool manages approvals and notifications for new admissions, student updates, and other processes. It ensures that every change is logged and reviewed by the concerned authorities.

7. METHODOLOGY

7.1 Table Creation

Custom tables are created using the Table Builder.

- u_student_details
- u_faculty_details
- u_course_details

Relationships between tables are defined using reference fields.

7.2 Form Design

Forms are created using Form Designer. These allow users to enter and view data efficiently. UI Policies are used to make fields mandatory or read-only depending on user roles.

7.3 Number Maintenance Setup

The “Number Maintenance” setup is configured to generate a unique, padded Student ID whenever a new record is created. This ensures systematic and error-free record creation.

7.4 Process Flow Creation

The Flow Designer automates workflows such as admissions, updates, and approval processes. Each flow includes triggers, actions, and conditions to handle various tasks.

7.5 Client Scripts Implementation

Scripts are written to enhance form behavior dynamically.

7.5.1 Auto Populate Script

Automatically fills in department or email when a student name is selected.

7.5.2 Periodic Update Script

Schedules automatic updates to reflect attendance or performance.

7.5.3 Disable Fields Script

Prevents editing of certain fields after record submission.

7.5.4 Total Marks Calculation Script

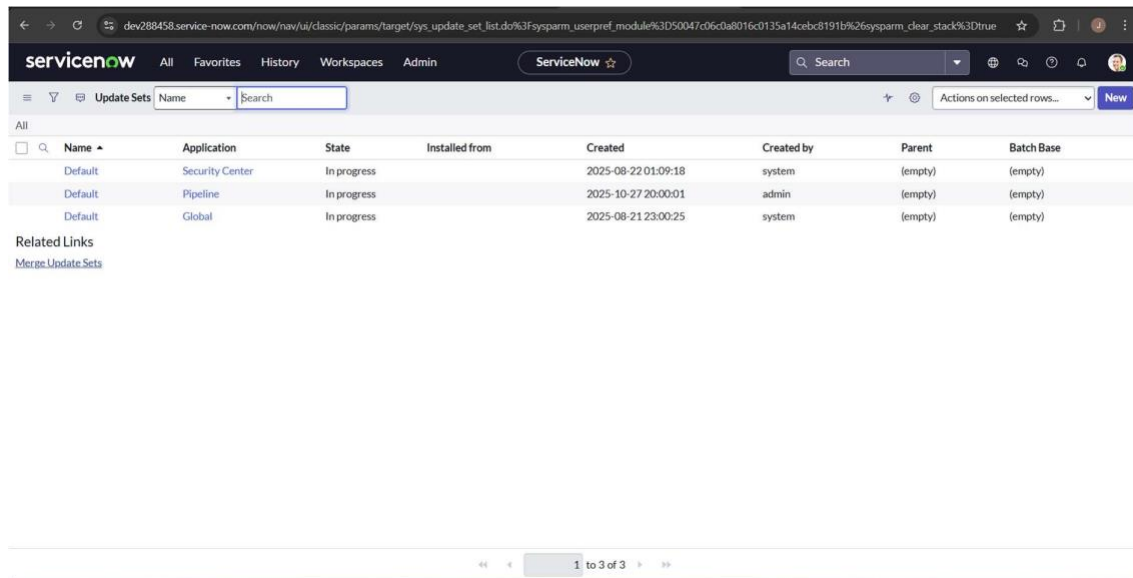
Calculates total marks from multiple subjects.

7.5.5 Percentage Calculation Script

Computes percentage based on total marks.

7.5.6 Result Evaluation Script

Determines pass/fail status based on percentage thresholds.



Name	Application	State	Installed from	Created	Created by	Parent	Batch Base
Default	Security Center	In progress		2025-08-22 01:09:18	system	(empty)	(empty)
Default	Pipeline	In progress		2025-10-27 20:00:01	admin	(empty)	(empty)
Default	Global	In progress		2025-08-21 23:00:25	system	(empty)	(empty)

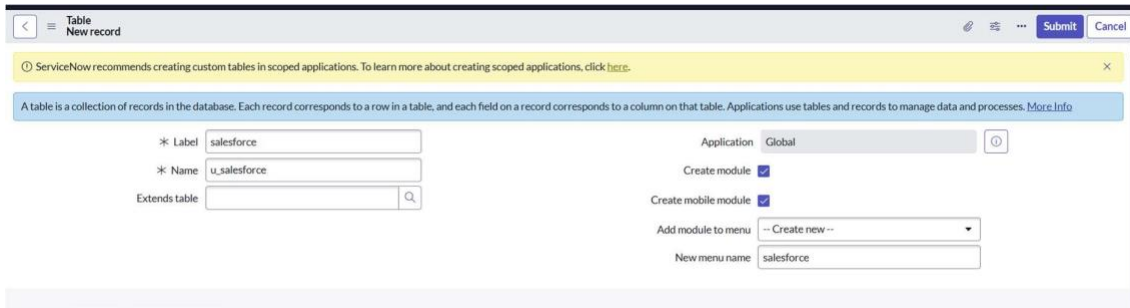


Table New record

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:

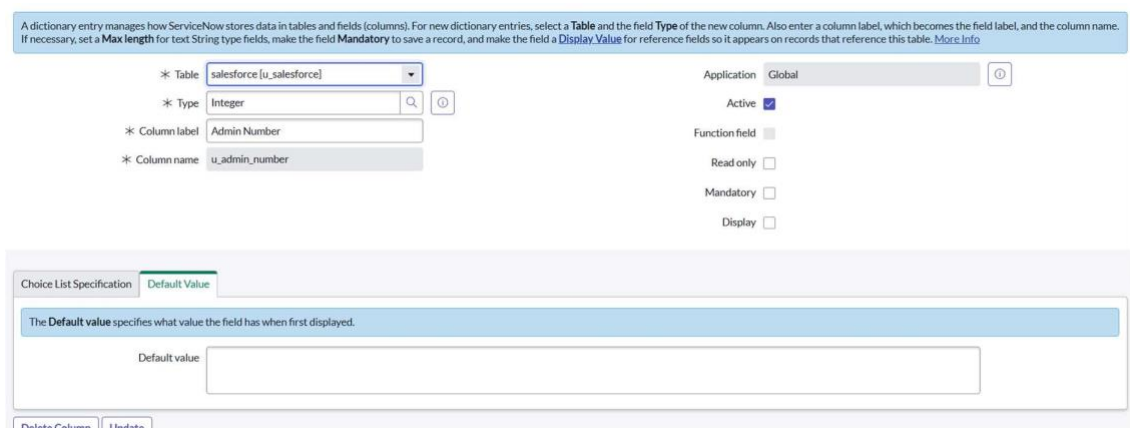
Application:

Create module: ☒

Create mobile module: ☒

Add module to menu:

New menu name:



A dictionary entry manages how ServiceNow stores data in tables and fields (columns). For new dictionary entries, select a **Table** and the field **Type** of the new column. Also enter a column label, which becomes the field label, and the column name. If necessary, set a **Max length** for text String type fields, make the field **Mandatory** to save a record, and make the field a **Display Value** for reference fields so it appears on records that reference this table. [More Info](#)

* Table:

* Type:

* Column label:

* Column name:

Application:

Active: ☒

Function field: ☐

Read only: ☐

Mandatory: ☐

Display: ☐

Choice List Specification:

Default Value

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

Default value:

Delete Column: Update:

* Name u_student_progress

Application: u_student_progress

Table Columns						
Dictionary Entries						
Column label	Type	Reference	Max length	Default value	Display	
×	Social	String	(empty)	40 40	false	
×	Created by	String	(empty)	40	false	
×	Science	String	(empty)	40 40	false	
×	Created	Date/Time	(empty)	40	false	
×	Telugu	String	(empty)	40 40	false	
×	Hindi	String	(empty)	40 40	false	
×	Sys ID	Sys ID (GUID)	(empty)	32	false	
×	Maths	String	(empty)	40 40	false	
×	Updates	Integer	(empty)	40	false	
×	Updated by	String	(empty)	40	false	

Form Builder
Design Form
Layout Form
Layout List
Show Form
Show List
Show Schema Map
Add to Service Catalog
Run Point Scan
Explore REST API

Access Controls (4)						
Access Controls						
Name	Decision Type	Operation	Type	Active	Updated by	Updated
u_student_progress	Allow If	create	record	true	admin	2025-10-28 02:10:44
u_student_progress	Allow If	delete	record	true	admin	2025-10-28 02:10:44
u_student_progress	Allow If	write	record	true	admin	2025-10-28 02:10:44
u_student_progress	Allow If	read	record	true	admin	2025-10-28 02:10:44

Flow Formatters						
All						
Name	Active	Condition	Description	Label	Order	Table
In Progress	true	u_admin_status=In Progress*EQ	In Progress			Admission [u_admission]
author	true	stage=author*EQ	Author		1	Publication [sn_publications_publication]
review	true	stage=review*EQ	Review		3	Publication [sn_publications_publication]
ready_to_publish	true	stage=ready_to_publish*EQ	Ready To Publish		4	Publication [sn_publications_publication]
published	true	stage=published*EQ	Published		5	Publication [sn_publications_publication]
expired	true	stage=expired*EQ	Expired		6	Publication [sn_publications_publication]
cancelled	true	stage=cancelled*EQ	Cancelled		7	Publication [sn_publications_publication]
Draft	true	life_cycle_stage=Ideation*EQ	Draft		10	Business Process [cmdb_ci_business_process]
Review	true	life_cycle_stage=Deploy*EQ	Review		15	Business Process [cmdb_ci_business_process]
Published	true	life_cycle_stage=Operational*EQ	Published		20	Business Process [cmdb_ci_business_process]
Expired	true	life_cycle_stage=End of Life*EQ	Expired		30	Business Process [cmdb_ci_business_process]

8. RESULT

The proposed system successfully replaces the manual data handling process with an automated ServiceNow-based platform. It simplifies record management, reduces human errors, and enhances productivity.

Administrators can now track admissions, faculty assignments, and performance in real-time.

The use of automation has significantly improved transparency and efficiency within the educational organization

9. Advantages

- Centralized data management for educational institutions.
- Real-time tracking of requests and academic data.
- Automation reduces workload and human errors.
- Improved transparency between students, staff, and administration.
- Cloud-based system — accessible from anywhere.
- Secure and role-based access control.

10. Disadvantages

- Requires a stable internet connection.
- Some technical knowledge is needed to configure advanced workflows.
- Limited customization options in the free ServiceNow developer instance.
- Initial setup and configuration can be time-consuming.

11. Future Enhancement

- Integration with external systems like Google Workspace or Moodle for seamless learning management.
- Adding AI-based chatbots for student queries.
- Mobile app development for easier access.
- Advanced analytics and dashboards for performance monitoring.
- Automated report generation and data export features.

12. Conclusion

The project “Educational Organisation Using ServiceNow” successfully demonstrates how cloud-based platforms can be leveraged to manage educational activities efficiently. Through automation, digital workflows, and structured data handling, this system reduces manual work and improves overall productivity. It provides a solid foundation for future digital transformation in academic environments.

13. References

1. ServiceNow Official Documentation — <https://docs.servicenow.com>
2. ServiceNow Developer Portal — <https://developer.servicenow.com>
3. Educational Case Studies using ServiceNow, 2024 Edition
4. Research Paper on ITSM Automation in Education (IEEE, 2023)