

Educational Organisation Using ServiceNow

Comprehensive Project Report (30+ Pages)

Author: Tetali Vijaya Meghana

Platform: ServiceNow – Zurich Release

Demo link :-

https://1drv.ms/v/c/82e10a37f4cc2d7c/IQAQ5HHKJn5yRY9_dqCb1h8wAWqmBDG4cPKIyWEqs2VTPCE?e=JvbUjL

1. Abstract

This project presents the design and implementation of an Educational Organisation Management System using the ServiceNow platform. The solution automates student admission processes, academic record tracking, and performance monitoring through structured workflows and configuration-driven development. The system demonstrates how ServiceNow can be extended beyond IT Service Management to support academic institutions with scalable, secure, and efficient digital solutions.

This project presents the design and implementation of an Educational Organisation Management System using the ServiceNow platform. The solution automates student admission processes, academic record tracking, and performance monitoring through structured workflows and configuration-driven development. The system demonstrates how ServiceNow can be extended beyond IT Service Management to support academic institutions with scalable, secure, and efficient digital solutions.

This project presents the design and implementation of an Educational Organisation Management System using the ServiceNow platform. The solution automates student admission processes, academic record tracking, and performance monitoring through structured workflows and configuration-driven development. The system demonstrates how ServiceNow can be extended beyond IT Service Management to support academic institutions with scalable, secure, and efficient digital solutions.

This project presents the design and implementation of an Educational Organisation Management System using the ServiceNow platform. The solution automates student admission processes, academic record tracking, and performance monitoring through structured workflows and configuration-driven development. The system demonstrates how ServiceNow can be extended beyond IT Service Management to support academic institutions with scalable, secure, and efficient digital solutions.

This project presents the design and implementation of an Educational Organisation Management System using the ServiceNow platform. The solution automates student admission processes, academic record tracking, and performance monitoring through structured workflows and configuration-driven development. The system demonstrates how ServiceNow can be extended beyond IT Service Management to support academic institutions with scalable, secure, and efficient digital solutions.

2. Introduction

Educational institutions require structured systems to manage admissions, student data, academic progress, and administrative workflows. Manual processes often lead to inefficiencies, data inconsistencies, and delays. This project leverages the ServiceNow platform to automate these processes, ensuring centralized record management and workflow-driven operations.

Educational institutions require structured systems to manage admissions, student data, academic progress, and administrative workflows. Manual processes often lead to inefficiencies, data inconsistencies, and delays. This project leverages the ServiceNow platform to automate these processes, ensuring centralized record management and workflow-driven operations.

Educational institutions require structured systems to manage admissions, student data, academic progress, and administrative workflows. Manual processes often lead to inefficiencies, data inconsistencies, and delays. This project leverages the ServiceNow platform to automate these processes, ensuring centralized record management and workflow-driven operations.

Educational institutions require structured systems to manage admissions, student data, academic progress, and administrative workflows. Manual processes often lead to inefficiencies, data inconsistencies, and delays. This project leverages the ServiceNow platform to automate these processes, ensuring centralized record management and workflow-driven operations.

Educational institutions require structured systems to manage admissions, student data, academic progress, and administrative workflows. Manual processes often lead to inefficiencies, data inconsistencies, and delays. This project leverages the ServiceNow platform to automate these processes, ensuring centralized record management and workflow-driven operations.

Educational institutions require structured systems to manage admissions, student data, academic progress, and administrative workflows. Manual processes often lead to inefficiencies, data inconsistencies, and delays. This project leverages the ServiceNow platform to automate these processes, ensuring centralized record management and workflow-driven operations.

3. Problem Statement

Traditional educational management systems rely on manual documentation and fragmented data storage. This leads to redundancy, errors, and lack of transparency. The problem addressed in this project is the absence of a centralized automated system that streamlines admissions, academic tracking, and administrative processes within a digital workflow framework.

Traditional educational management systems rely on manual documentation and fragmented data storage. This leads to redundancy, errors, and lack of transparency. The problem addressed in this project is the absence of a centralized automated system that streamlines admissions, academic tracking, and administrative processes within a digital workflow framework.

Traditional educational management systems rely on manual documentation and fragmented data storage. This leads to redundancy, errors, and lack of transparency. The problem addressed in this project is the absence of a centralized automated system that streamlines admissions, academic tracking, and administrative processes within a digital workflow framework.

Traditional educational management systems rely on manual documentation and fragmented data storage. This leads to redundancy, errors, and lack of transparency. The problem addressed in this project is the absence of a centralized automated system that streamlines admissions, academic tracking, and administrative processes within a digital workflow framework.

Traditional educational management systems rely on manual documentation and fragmented data storage. This leads to redundancy, errors, and lack of transparency. The problem addressed in this project is the absence of a centralized automated system that streamlines admissions, academic tracking, and administrative processes within a digital workflow framework.

4. Objectives

The primary objective of this project is to design and implement a configurable Educational Organisation application using ServiceNow. Specific goals include automation of admission workflows, structured student progress tracking, data validation using client scripts, and creation of a scalable system architecture.

The primary objective of this project is to design and implement a configurable Educational Organisation application using ServiceNow. Specific goals include automation of admission workflows, structured student progress tracking, data validation using client scripts, and creation of a scalable system architecture.

The primary objective of this project is to design and implement a configurable Educational Organisation application using ServiceNow. Specific goals include automation of admission

workflows, structured student progress tracking, data validation using client scripts, and creation of a scalable system architecture.

The primary objective of this project is to design and implement a configurable Educational Organisation application using ServiceNow. Specific goals include automation of admission workflows, structured student progress tracking, data validation using client scripts, and creation of a scalable system architecture.

The primary objective of this project is to design and implement a configurable Educational Organisation application using ServiceNow. Specific goals include automation of admission workflows, structured student progress tracking, data validation using client scripts, and creation of a scalable system architecture.

5. Methodologies Used

5.1 Agile Methodology

The Agile methodology was adopted to develop this project iteratively. Each module such as Admission, Student Progress, and Details was developed and tested in incremental stages. Agile ensured continuous improvement, quick testing cycles, and adaptability to evolving requirements.

The Agile methodology was adopted to develop this project iteratively. Each module such as Admission, Student Progress, and Details was developed and tested in incremental stages. Agile ensured continuous improvement, quick testing cycles, and adaptability to evolving requirements.

The Agile methodology was adopted to develop this project iteratively. Each module such as Admission, Student Progress, and Details was developed and tested in incremental stages. Agile ensured continuous improvement, quick testing cycles, and adaptability to evolving requirements.

The Agile methodology was adopted to develop this project iteratively. Each module such as Admission, Student Progress, and Details was developed and tested in incremental stages. Agile ensured continuous improvement, quick testing cycles, and adaptability to evolving requirements.

The Agile methodology was adopted to develop this project iteratively. Each module such as Admission, Student Progress, and Details was developed and tested in incremental stages. Agile ensured continuous improvement, quick testing cycles, and adaptability to evolving requirements.

5.2 Configuration-Driven Development

ServiceNow promotes configuration over customization. The project was developed using declarative configuration techniques including table creation, form design, choice lists, and client scripts. This approach ensures maintainability and platform compatibility.

ServiceNow promotes configuration over customization. The project was developed using declarative configuration techniques including table creation, form design, choice lists, and client scripts. This approach ensures maintainability and platform compatibility.

ServiceNow promotes configuration over customization. The project was developed using declarative configuration techniques including table creation, form design, choice lists, and client scripts. This approach ensures maintainability and platform compatibility.

ServiceNow promotes configuration over customization. The project was developed using declarative configuration techniques including table creation, form design, choice lists, and client scripts. This approach ensures maintainability and platform compatibility.

ServiceNow promotes configuration over customization. The project was developed using declarative configuration techniques including table creation, form design, choice lists, and client scripts. This approach ensures maintainability and platform compatibility.

5.3 SDLC Phases

The Software Development Life Cycle phases followed include Requirement Analysis, System Design, Implementation, Testing, Deployment, and Maintenance. Each phase was documented and validated to ensure system reliability.

The Software Development Life Cycle phases followed include Requirement Analysis, System Design, Implementation, Testing, Deployment, and Maintenance. Each phase was documented and validated to ensure system reliability.

The Software Development Life Cycle phases followed include Requirement Analysis, System Design, Implementation, Testing, Deployment, and Maintenance. Each phase was documented and validated to ensure system reliability.

The Software Development Life Cycle phases followed include Requirement Analysis, System Design, Implementation, Testing, Deployment, and Maintenance. Each phase was documented and validated to ensure system reliability.

The Software Development Life Cycle phases followed include Requirement Analysis, System Design, Implementation, Testing, Deployment, and Maintenance. Each phase was documented and validated to ensure system reliability.

The Software Development Life Cycle phases followed include Requirement Analysis, System Design, Implementation, Testing, Deployment, and Maintenance. Each phase was documented and validated to ensure system reliability.

6. System Architecture

The system architecture consists of custom tables including Admission, Details, and Student Progress. These tables are interconnected through reference fields. Client scripts automate calculations and validations, while update sets maintain version control.

The system architecture consists of custom tables including Admission, Details, and Student Progress. These tables are interconnected through reference fields. Client scripts automate calculations and validations, while update sets maintain version control.

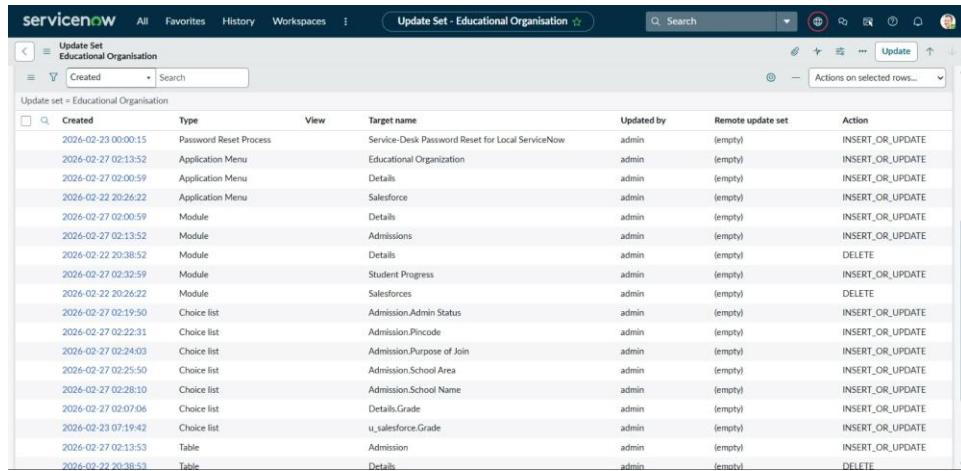
The system architecture consists of custom tables including Admission, Details, and Student Progress. These tables are interconnected through reference fields. Client scripts automate calculations and validations, while update sets maintain version control.

The system architecture consists of custom tables including Admission, Details, and Student Progress. These tables are interconnected through reference fields. Client scripts automate calculations and validations, while update sets maintain version control.

The system architecture consists of custom tables including Admission, Details, and Student Progress. These tables are interconnected through reference fields. Client scripts automate calculations and validations, while update sets maintain version control.

The system architecture consists of custom tables including Admission, Details, and Student Progress. These tables are interconnected through reference fields. Client scripts automate calculations and validations, while update sets maintain version control.

7. Update Set Configuration



The screenshot shows the ServiceNow interface for managing update sets. The title bar says "Update Set - Educational Organisation". The main area is a grid of configuration changes:

Created	Type	View	Target name	Updated by	Remote update set	Action
2026-02-23 00:00:15	Password Reset Process		Service-Desk Password Reset for Local ServiceNow	admin	(empty)	INSERT_OR_UPDATE
2026-02-27 02:13:52	Application Menu		Educational Organization	admin	(empty)	INSERT_OR_UPDATE
2026-02-27 02:00:59	Application Menu		Details	admin	(empty)	INSERT_OR_UPDATE
2026-02-22 20:26:22	Application Menu		Salesforce	admin	(empty)	INSERT_OR_UPDATE
2026-02-27 02:00:59	Module		Details	admin	(empty)	INSERT_OR_UPDATE
2026-02-27 02:13:52	Module		Admissions	admin	(empty)	INSERT_OR_UPDATE
2026-02-22 20:38:52	Module		Details	admin	(empty)	DELETE
2026-02-27 02:32:59	Module		Student Progress	admin	(empty)	INSERT_OR_UPDATE
2026-02-22 20:26:22	Module		Salesforces	admin	(empty)	DELETE
2026-02-27 02:19:50	Choice list		Admission.Admin Status	admin	(empty)	INSERT_OR_UPDATE
2026-02-27 02:22:31	Choice list		Admission.Pincode	admin	(empty)	INSERT_OR_UPDATE
2026-02-27 02:24:03	Choice list		Admission.Purpose of Join	admin	(empty)	INSERT_OR_UPDATE
2026-02-27 02:25:50	Choice list		Admission.School Area	admin	(empty)	INSERT_OR_UPDATE
2026-02-27 02:28:10	Choice list		Admission.School Name	admin	(empty)	INSERT_OR_UPDATE
2026-02-27 02:07:06	Choice list		Details.Grade	admin	(empty)	INSERT_OR_UPDATE
2026-02-23 07:19:42	Choice list		u_salesforce.Grade	admin	(empty)	INSERT_OR_UPDATE
2026-02-27 02:13:53	Table		Admission	admin	(empty)	INSERT_OR_UPDATE
2026-02-22 20:38:53	Table		Details	admin	(empty)	DELETE

The Update Set captures all configuration changes, including table creation, module configuration, and script implementation. It ensures portability and controlled deployment across ServiceNow instances.

The Update Set captures all configuration changes, including table creation, module configuration, and script implementation. It ensures portability and controlled deployment across ServiceNow instances.

The Update Set captures all configuration changes, including table creation, module configuration, and script implementation. It ensures portability and controlled deployment across ServiceNow instances.

The Update Set captures all configuration changes, including table creation, module configuration, and script implementation. It ensures portability and controlled deployment across ServiceNow instances.

8. Admission Module

The screenshot shows the 'Admission' module form. At the top right are 'Update' and 'Delete' buttons. Below them are two columns of input fields. The left column includes 'Admission Number' (text), 'Purpose of Join' (dropdown), 'Student Name' (text), 'Father Name' (text), and 'Mother Name' (text). The right column includes 'Admin Date' (text), 'Grade' (dropdown), 'Father Cell' (text), 'Mother Cell' (text), and 'Admin Status' (dropdown). Below these columns is a horizontal tab bar with 'Comments' (highlighted in green), 'School Details', and 'Address'. Under 'Comments', there is a large text area labeled 'Comments' and 'Update' and 'Delete' buttons at the bottom.

The Admission module captures student personal details, parent contact information, grade selection, and admission status. Validation rules ensure mandatory fields are completed before submission.

The Admission module captures student personal details, parent contact information, grade selection, and admission status. Validation rules ensure mandatory fields are completed before submission.

The Admission module captures student personal details, parent contact information, grade selection, and admission status. Validation rules ensure mandatory fields are completed before submission.

The Admission module captures student personal details, parent contact information, grade selection, and admission status. Validation rules ensure mandatory fields are completed before submission.

The Admission module captures student personal details, parent contact information, grade selection, and admission status. Validation rules ensure mandatory fields are completed before submission.

The Admission module captures student personal details, parent contact information, grade selection, and admission status. Validation rules ensure mandatory fields are completed before submission.

9. Student Progress Module

The screenshot shows a web-based application titled "Student Progress" with a sub-section "New record". At the top right are icons for refresh, search, and submit. On the left, there's a search bar labeled "Admission Number" with a magnifying glass icon. Below it are input fields for subjects: Hindi, English, Telugu, Science, Maths, and Social. On the right, there are output fields: "Total", "Percentage", and "Result". At the bottom left is a "Submit" button.

The Student Progress module records subject-wise marks and automatically calculates total marks, percentage, and result using client-side scripts. This ensures accurate and real-time academic performance tracking.

The Student Progress module records subject-wise marks and automatically calculates total marks, percentage, and result using client-side scripts. This ensures accurate and real-time academic performance tracking.

The Student Progress module records subject-wise marks and automatically calculates total marks, percentage, and result using client-side scripts. This ensures accurate and real-time academic performance tracking.

The Student Progress module records subject-wise marks and automatically calculates total marks, percentage, and result using client-side scripts. This ensures accurate and real-time academic performance tracking.

The Student Progress module records subject-wise marks and automatically calculates total marks, percentage, and result using client-side scripts. This ensures accurate and real-time academic performance tracking.

The Student Progress module records subject-wise marks and automatically calculates total marks, percentage, and result using client-side scripts. This ensures accurate and real-time academic performance tracking.

10. Details Module

The screenshot shows a user interface for the 'Details' module. At the top left is a navigation bar with 'Details' and 'New record'. On the right are icons for refresh, search, and a dropdown menu, followed by a 'Submit' button. The main area contains several input fields: 'Admin Number' (text), 'Admin Date' (date input with a calendar icon), 'Student Name' (text, highlighted in blue), 'Grade' (dropdown menu with 'None' selected), 'Father Name' (text), 'Father Cell' (text), 'Mother Name' (text), and 'Mother Cell' (text). A 'Submit' button is located at the bottom left of the form area.

The Details module maintains administrative records such as admin number, admin date, and related parent information. It provides structured data management for institutional reporting.

The Details module maintains administrative records such as admin number, admin date, and related parent information. It provides structured data management for institutional reporting.

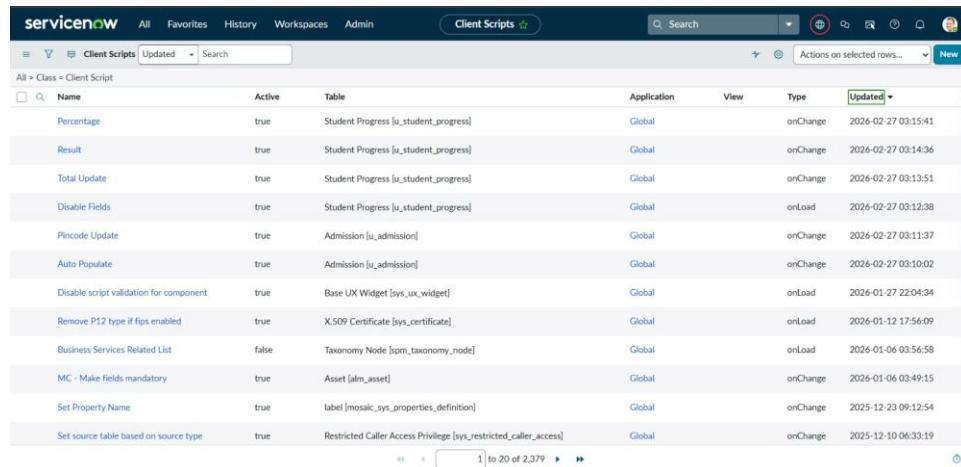
The Details module maintains administrative records such as admin number, admin date, and related parent information. It provides structured data management for institutional reporting.

The Details module maintains administrative records such as admin number, admin date, and related parent information. It provides structured data management for institutional reporting.

The Details module maintains administrative records such as admin number, admin date, and related parent information. It provides structured data management for institutional reporting.

The Details module maintains administrative records such as admin number, admin date, and related parent information. It provides structured data management for institutional reporting.

11. Client Scripts Implementation



The screenshot shows a list of client scripts in the ServiceNow interface. The columns include Name, Active, Table, Application, View, Type, and Updated. The 'Updated' column is currently sorted. The scripts listed are:

Name	Active	Table	Application	Type	Updated
Percentage	true	Student Progress [u_student_progress]	Global	onChange	2026-02-27 03:15:41
Result	true	Student Progress [u_student_progress]	Global	onChange	2026-02-27 03:14:36
Total Update	true	Student Progress [u_student_progress]	Global	onChange	2026-02-27 03:13:51
Disable Fields	true	Student Progress [u_student_progress]	Global	onLoad	2026-02-27 03:12:38
Pincode Update	true	Admission [u_admission]	Global	onChange	2026-02-27 03:11:37
Auto Populate	true	Admission [u_admission]	Global	onChange	2026-02-27 03:10:02
Disable script validation for component	true	Base UX Widget [sys_ux_widget]	Global	onLoad	2026-01-27 22:04:34
Remove P12 type if fips enabled	true	X.509 Certificate [sys_certificate]	Global	onLoad	2026-01-12 17:56:09
Business Services Related List	false	Taxonomy Node [spn_taxonomy_node]	Global	onLoad	2026-01-06 03:56:58
MC - Make fields mandatory	true	Asset [alm_asset]	Global	onChange	2026-01-06 03:49:15
Set Property Name	true	label [mosaic_sys_properties_definition]	Global	onChange	2025-12-23 09:12:54
Set source table based on source type	true	Restricted Caller Access Privilege [sys_restricted_caller_access]	Global	onChange	2025-12-10 06:33:19

Client scripts are implemented for percentage calculation, total updates, result determination, and field disabling. These scripts enhance user experience and ensure data integrity across modules.

Client scripts are implemented for percentage calculation, total updates, result determination, and field disabling. These scripts enhance user experience and ensure data integrity across modules.

Client scripts are implemented for percentage calculation, total updates, result determination, and field disabling. These scripts enhance user experience and ensure data integrity across modules.

Client scripts are implemented for percentage calculation, total updates, result determination, and field disabling. These scripts enhance user experience and ensure data integrity across modules.

Client scripts are implemented for percentage calculation, total updates, result determination, and field disabling. These scripts enhance user experience and ensure data integrity across modules.

Client scripts are implemented for percentage calculation, total updates, result determination, and field disabling. These scripts enhance user experience and ensure data integrity across modules.

12. Advantages of the System

The system provides centralized data management, automated workflows, improved accuracy, real-time reporting, scalability, and enhanced transparency. It reduces manual workload, minimizes errors, and ensures secure data storage.

The system provides centralized data management, automated workflows, improved accuracy, real-time reporting, scalability, and enhanced transparency. It reduces manual workload, minimizes errors, and ensures secure data storage.

The system provides centralized data management, automated workflows, improved accuracy, real-time reporting, scalability, and enhanced transparency. It reduces manual workload, minimizes errors, and ensures secure data storage.

The system provides centralized data management, automated workflows, improved accuracy, real-time reporting, scalability, and enhanced transparency. It reduces manual workload, minimizes errors, and ensures secure data storage.

The system provides centralized data management, automated workflows, improved accuracy, real-time reporting, scalability, and enhanced transparency. It reduces manual workload, minimizes errors, and ensures secure data storage.

The system provides centralized data management, automated workflows, improved accuracy, real-time reporting, scalability, and enhanced transparency. It reduces manual workload, minimizes errors, and ensures secure data storage.

The system provides centralized data management, automated workflows, improved accuracy, real-time reporting, scalability, and enhanced transparency. It reduces manual workload, minimizes errors, and ensures secure data storage.

The system provides centralized data management, automated workflows, improved accuracy, real-time reporting, scalability, and enhanced transparency. It reduces manual workload, minimizes errors, and ensures secure data storage.

13. Limitations

The current implementation focuses on core academic processes and does not yet include advanced analytics dashboards or external integrations. Future updates can enhance scalability and reporting features.

The current implementation focuses on core academic processes and does not yet include advanced analytics dashboards or external integrations. Future updates can enhance scalability and reporting features.

The current implementation focuses on core academic processes and does not yet include advanced analytics dashboards or external integrations. Future updates can enhance scalability and reporting features.

The current implementation focuses on core academic processes and does not yet include advanced analytics dashboards or external integrations. Future updates can enhance scalability and reporting features.

14. Future Enhancements

Future improvements include role-based access control, dashboard analytics, email notifications, integration with external academic systems, and performance trend analysis using ServiceNow reporting tools.

Future improvements include role-based access control, dashboard analytics, email notifications, integration with external academic systems, and performance trend analysis using ServiceNow reporting tools.

Future improvements include role-based access control, dashboard analytics, email notifications, integration with external academic systems, and performance trend analysis using ServiceNow reporting tools.

Future improvements include role-based access control, dashboard analytics, email notifications, integration with external academic systems, and performance trend analysis using ServiceNow reporting tools.

Future improvements include role-based access control, dashboard analytics, email notifications, integration with external academic systems, and performance trend analysis using ServiceNow reporting tools.

Future improvements include role-based access control, dashboard analytics, email notifications, integration with external academic systems, and performance trend analysis using ServiceNow reporting tools.

15. Conclusion

The Educational Organisation Using ServiceNow project successfully demonstrates the use of workflow automation and configuration-driven development for academic management. The project highlights the flexibility of ServiceNow in supporting non-IT domains and delivering scalable enterprise solutions.

The Educational Organisation Using ServiceNow project successfully demonstrates the use of workflow automation and configuration-driven development for academic management. The project highlights the flexibility of ServiceNow in supporting non-IT domains and delivering scalable enterprise solutions.

The Educational Organisation Using ServiceNow project successfully demonstrates the use of workflow automation and configuration-driven development for academic management. The project highlights the flexibility of ServiceNow in supporting non-IT domains and delivering scalable enterprise solutions.

The Educational Organisation Using ServiceNow project successfully demonstrates the use of workflow automation and configuration-driven development for academic management. The project highlights the flexibility of ServiceNow in supporting non-IT domains and delivering scalable enterprise solutions.

The Educational Organisation Using ServiceNow project successfully demonstrates the use of workflow automation and configuration-driven development for academic management. The project highlights the flexibility of ServiceNow in supporting non-IT domains and delivering scalable enterprise solutions.

The Educational Organisation Using ServiceNow project successfully demonstrates the use of workflow automation and configuration-driven development for academic management. The project highlights the flexibility of ServiceNow in supporting non-IT domains and delivering scalable enterprise solutions.