



# **Aditya College of Engineering and Technology**

## **In association with SmartInternz & ServiceNow**



**Project Title:**  
**Educational Organization Management Using  
ServiceNow**

**Submitted by:**  
**Team ID: LTVIP2025TMID28591**

**Team Members:**

**Roopa Gayatri Devarapalli**  
**Naga Sandeep Saladi**  
**Nelli Chandini**  
**Satya Sai Devi Arumilli**

- 1. INTRODUCTION**
  - 1.1 Project Overview
  - 1.2 Purpose
- 2. IDEATION PHASE**
  - 2.1 Problem Statement
  - 2.2 Empathy Map Canvas
  - 2.3 Brainstorming
- 3. REQUIREMENT ANALYSIS**

- 3.1 Customer Journey map
- 3.2 Solution Requirement
- 3.3 Data Flow Diagram
- 3.4 Technology Stack

- 4. PROJECT DESIGN**

- 4.1 Problem Solution Fit
- 4.2 Proposed Solution
- 4.3 Solution Architecture

- 5. PROJECT PLANNING & SCHEDULING**

- 5.1 Project Planning

- 6. FUNCTIONAL AND PERFORMANCE TESTING**

- 6.1 Performance Testing

- 7. RESULTS**

- 7.1 Output Screenshots

- 8. ADVANTAGES & DISADVANTAGES**

- 9. CONCLUSION**

- 10. FUTURE SCOPE**

- 11. APPENDIX**

Source Code(if any)

Dataset Link

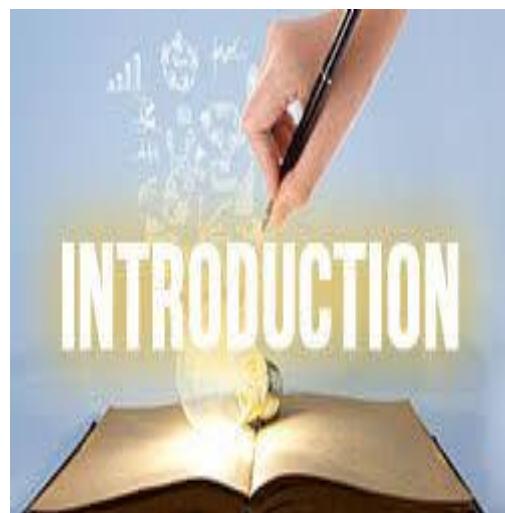
[GitHub & Project Demo Link](#)

---

## **1.INTRODUCTION**

### **1.1 Project overview**

### **1.2 Purpose**



# **1. INTRODUCTION**

## **1.1 Project Overview**

In today's fast-paced academic environment, educational institutions need digital solutions to manage their core functions such as admissions, student records, faculty data, and course scheduling.

Manual handling of these processes often leads to inefficiencies, errors, and delays. Our project titled Educational Organization Management System is a customized ServiceNow application designed to address these challenges.

The system provides an integrated, cloud-based solution to manage various academic operations in a centralized and efficient manner. This platform facilitates the automation of key processes such as student registration, record keeping, attendance tracking, and faculty management.

Leveraging the ServiceNow platform's capabilities—tables, forms, workflows, business rules, and UI customization—this application ensures real-time access, accuracy, and accountability in operations.

## 1.2 Purpose

The primary purpose of this project is to offer an efficient and scalable solution for managing educational organizations' day-to-day operations. By implementing this system, institutions can:

- Streamline administrative workflows through automation.
- Maintain an organized digital repository of students, faculty, and courses.
- Reduce the manual workload and eliminate paper-based systems.
- Enable better communication and transparency between departments.

This solution is especially beneficial for institutions seeking to adopt digital transformation in their administrative processes. It ensures that both staff and students experience improved service delivery and faster response times.

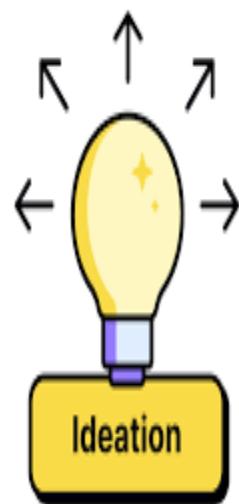


## **2. IDEATION PHASE**

### **2.1 Problem Statement**

### **2.2 Empathy Map Canvas**

### **2.3 Brainstroming**



## **2.1 Problem Statement:**

Educational institutions traditionally rely on manual processes and disconnected systems to handle student admissions, faculty data, and academic records. These systems are prone to errors, time-consuming, and lack transparency.

Staff members often struggle with maintaining large volumes of paper-based records, which also makes retrieving data difficult during audits or emergencies. The absence of a centralized, user-friendly platform significantly impacts decision-making and student service delivery.

Our project solves this problem by developing a cloud-based, modular, and easy-to-use application using the ServiceNow platform, aiming to automate repetitive tasks and unify data management for educational organizations.



## **2.2 Empathy Map Canvas:**

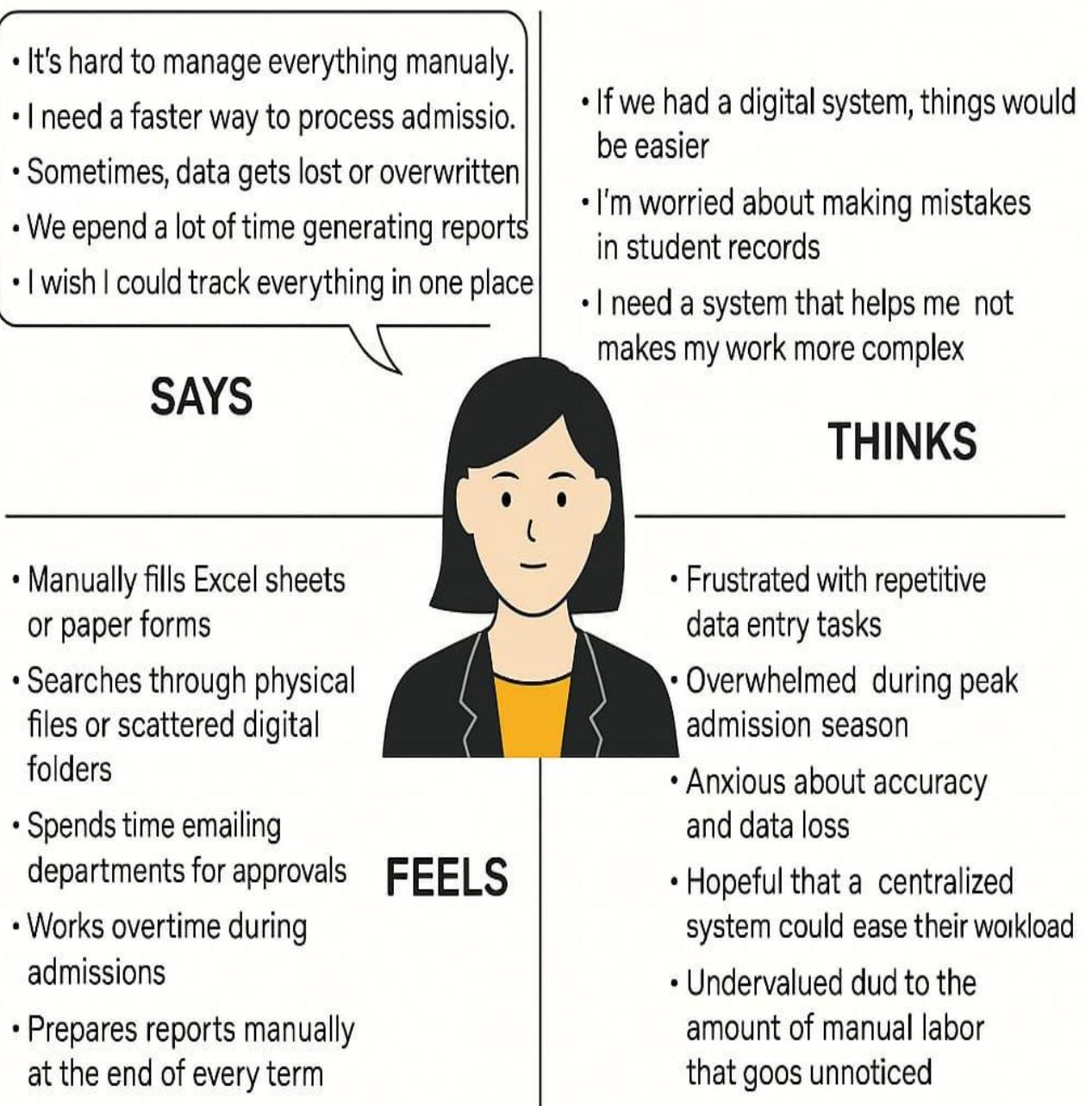
We created an empathy map after interacting with potential users—administrative staff and faculty members. Their thoughts and experiences helped us understand user expectations:

- Says: “We spend too much time on repetitive tasks.”
- Thinks: “If the process was automated, we could focus on teaching or planning.”
- Does: Enters records manually, follows up with multiple departments for approvals.
- Feels: Frustrated due to delays, errors, and lack of control.

This exercise highlighted the need for a centralized platform to reduce workload, improve accessibility, and enhance institutional transparency

# EDUCATIONAL ORGANIZATION MANAGEMENT SYSTEM

⌚ Target User: Administrative Staff/College Office Clerk



## **2.2 BrainStroming:**

Our team brainstormed and listed potential features based on user pain points:

- A dedicated admission form with automated field validations.
- Role-based dashboard access for faculty and administrators.
- Reports for student attendance, marks, and enrollment.
- Notification system for form approval, updates, or rejections.
- A simplified form layout to make user interaction intuitive.

By evaluating these ideas using prioritization matrices, we selected features that balanced feasibility, impact, and urgency.

# EDUCATIONAL ORGANIZATION MANAGEMENT

## Pain Points

- repetitive data entry
- manual verification
- lack of transparency
- time-consuming approvals

## Key Ideas

- centralized portal
- automation of admissions
- dashboards for reports
- role-based access

## How might we...

automate workflows  
for faster processing

## Features

- admission form
- role-based dashboards
- reports & analytics
- notification system

## **3. Requirement Analysis**

3.1 Customer Journey Map

3.2 Solution Requirement

3.3 Data Flow Diagram

3.4 Technology Stack

### **3.1 Customer Journey Map:**

Mapping the customer journey allowed us to identify friction points and design effective solutions. For example:

- A student applies for admission using the online form.
- Admin receives a notification and reviews the application.
- Upon approval, the student data is automatically added to the student table.
- Email confirmation is sent, and status is updated in real time.

This journey emphasizes the need for status tracking, auto-notifications, and form validations, all of which we implemented using ServiceNow workflows and policies.

# CUSTOMER JOURNEY MAP

Administrative staff/college office clerk managing admissions, student & faculty records

APPLICATION | TRACKING | PROCESSING | REO

Manual data entry into records

Searching for status of applications

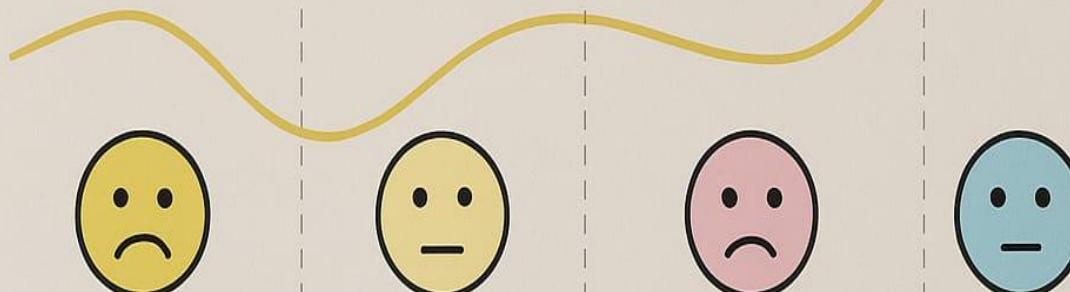
Coordinating with faculty for approvals



Applying data entry into records



Preparing reports for management



## **3.2 Solution Requirement:**

Key requirements include:

- Functional:**

- Admission Form with fields like Name, Email, Course, DOB, and Gender.
- Faculty Module for managing departments and subjects.
- Report generation for student records.
- Role-based access to protect data privacy.

- Non-Functional:**

- User-friendly interface.
- Fast form response time.
- Integration with existing ServiceNow modules.

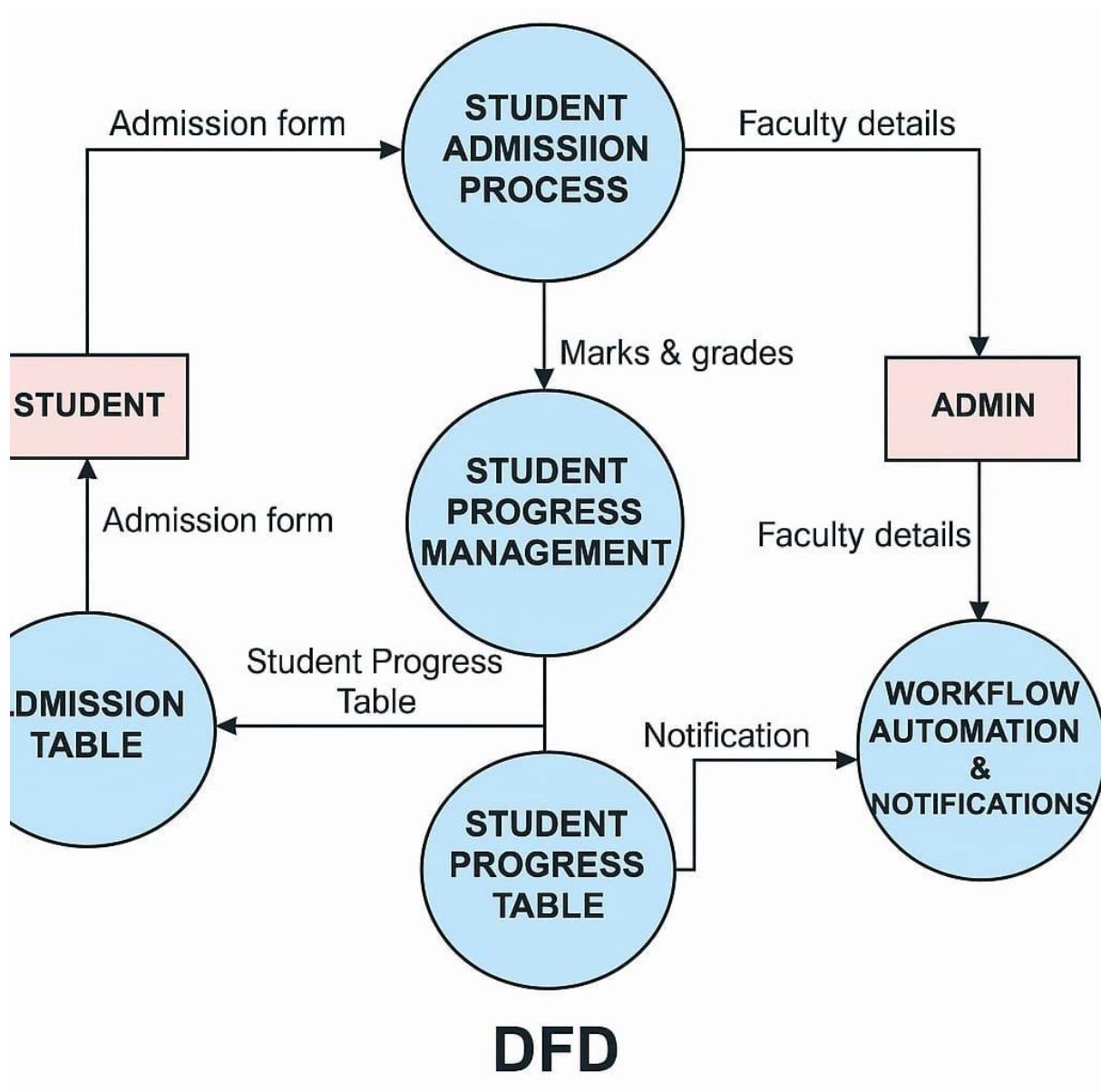
We also required customization options such as choice lists (gender, course), date validation, and email format validation all of which are supported natively in ServiceNow

### **3.3 Data Flow Diagram:**

The system's data flow follows a simple but efficient pattern:

- Input: Student/Faculty fills the form.
- Processing: Data is stored in respective tables, workflows triggered.
- Output: Status update, report generation, email notification.

This flow ensures transparency and accuracy, as each step is logged and traceable within ServiceNow.



### **3.4 Technology Stack:**

We used:

- **Platform:** ServiceNow
- **Front-end:** Forms, UI Actions, Choice Lists, Client Scripts
- **Back-end:** Tables, Business Rules, Server-side Scripts
- **Tools:** Update Sets, Flow Designer, Report Designer

The decision to use ServiceNow was influenced by its robust workflow automation, cloud-native infrastructure, and low-code development features.

## **4. Project Design**

4.1 Problem Solution Fit

4.2 Proposed Solution

4.3 Solution Architecture

## 4.1 Problem Solution Fit:

Our solution directly addresses the identified problems—manual work, errors, and data fragmentation. The form-based interface automates admission and course processes. Admins no longer need to manually update spreadsheets or maintain physical records. With status tracking and workflows, every operation is transparent and accountable.

The platform also ensures that sensitive information is only accessible to authorized users through role-based permissions.

The screenshot shows a ServiceNow Project Workspace interface. At the top, there are two tabs: "Guided Project" (which is selected) and "Project Workspace". On the right side of the header, there is a "More" icon represented by three vertical dots. The main content area is titled "Educational Organisation Using ServiceNow". To the left of this title is a green button labeled "Educational Organisation Using ServiceNow". Below this are several orange buttons, each with a plus sign icon and a label: "Setting Up ServiceNow Instance", "Creating A Update Set", "Creating A Table", "Form Layout", "Form Design", "Number Maintenance", "Process Flow", "Client Script", and "Result". The "Educational Organisation Using ServiceNow" title and its description are contained within a light gray box with rounded corners. The description text reads: "The Educational Management System is a comprehensive platform designed to streamline administrative tasks within educational institutions. It facilitates efficient management of student and teacher data, simplifies the admission process, and provides tools for monitoring student progress."

## **4.2 Proposed Solution:**

To address the challenges faced by educational institutions in managing student and administrative data, we proposed a centralized application on the ServiceNow platform, focused on three core functional tables. These tables form the foundation of the entire solution and are interconnected to ensure seamless data flow and workflow execution.

### Key Tables Used:

#### 1. Admission Table

This table captures all the essential data for new student admissions. It includes fields such as:

- Student Name
- Date of Birth
- Gender
- Email ID
- Course Applied
- Admission Status

Using this table, administrative staff can easily create, edit, and track the admission status of students. We implemented field validations like email format checks, date validations, and choice lists for fields like gender and course selection to ensure data integrity.

## **2. Student Progress Table**

This table maintains academic performance and progress details of each student post-admission. Fields include:

- Student ID (reference field from Admission Table)
- Course Enrolled
- Attendance %
- Internal Marks
- Final Grade
- Faculty Remarks

This table supports reporting and helps faculty members update student performance. It can also be extended to auto-generate grade reports or trigger alerts for low-performing students.

-  3. Salesforce Table (*used for demo/data integration purposes*)

This table is designed to simulate integration or mimic external CRM-like data handling, primarily for institutional outreach, parent communication, or lead management during admissions. It includes:

- Lead Name
- Contact Number
- Follow-up Date
- Status (Hot/Warm/Cold)
- Notes

This was optionally added to represent how the system could be extended to manage inquiries or integrate with Salesforce-like systems in real use cases.

### How These Tables Work Together:

- When a student fills out the admission form, the Admission Table stores the data.
- Once admitted, the student's ID is used in the Student Progress Table to track performance.
- For external inquiry tracking or follow-up, the Salesforce Table is optionally populated.

### Other Design Elements:

- Reference Fields: Used to link student data across tables.
- Workflows: Triggered on form submission for approval and status update.
- UI Policies & Scripts: Used to dynamically control form behavior and improve user experience.

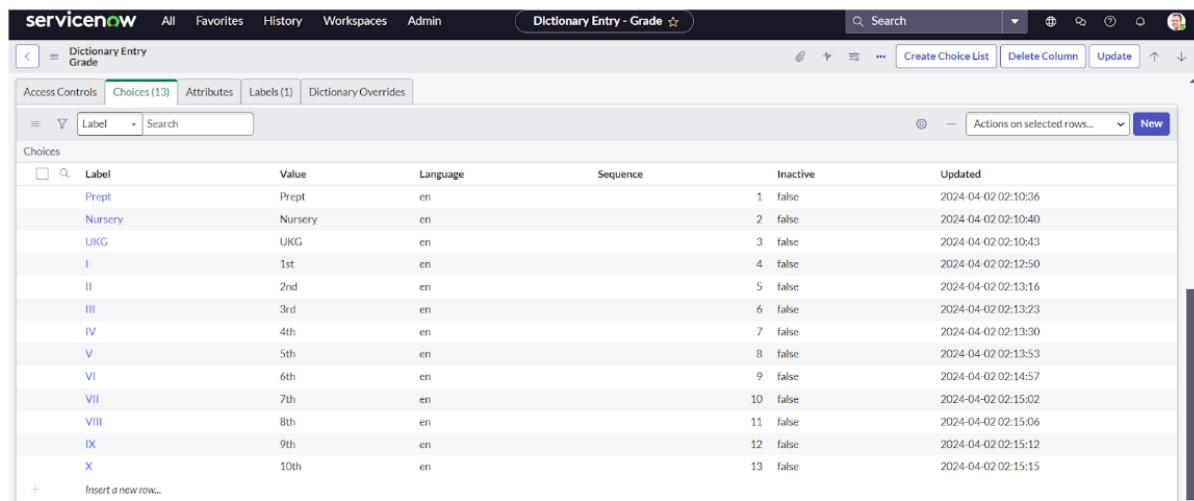
This modular approach allows the system to be extended or scaled easily while keeping the interface clean and focused on core academic operations.

## 4.3 Solution Architecture:

The architecture is structured as:

- **Presentation Layer:** Forms, views, navigation modules.
- **Application Layer:** Workflows, UI Policies, Client Scripts.
- **Data Layer:** Tables, fields, relationships.

We also implemented reference fields to create relationships between tables (e.g., Course table linked to Student table).



A screenshot of the ServiceNow interface showing the 'Dictionary Entry - Grade' page. The top navigation bar includes 'servicenow', 'All', 'Favorites', 'History', 'Workspaces', and 'Admin'. The title bar says 'Dictionary Entry - Grade'. The main content area shows a table titled 'Choices' with the following data:

Label	Value	Language	Sequence	Inactive	Updated
Prept	Prept	en	1	false	2024-04-02 02:10:36
Nursery	Nursery	en	2	false	2024-04-02 02:10:40
UKG	UKG	en	3	false	2024-04-02 02:10:43
I	1st	en	4	false	2024-04-02 02:12:50
II	2nd	en	5	false	2024-04-02 02:13:16
III	3rd	en	6	false	2024-04-02 02:13:23
IV	4th	en	7	false	2024-04-02 02:13:30
V	5th	en	8	false	2024-04-02 02:13:53
VI	6th	en	9	false	2024-04-02 02:14:57
VII	7th	en	10	false	2024-04-02 02:15:02
VIII	8th	en	11	false	2024-04-02 02:15:06
IX	9th	en	12	false	2024-04-02 02:15:12
X	10th	en	13	false	2024-04-02 02:15:15

## **5.Project Planning And Scheduling**

### **5.1 Project Planning**

## **5.1 Project Planning:**

Given the limited duration of the internship program, we adopted a **condensed and focused project timeline** to ensure that all essential features of the Educational Organization Management System were completed efficiently. Our team followed a task-based planning approach instead of a long-term schedule.

### **Project Timeline Overview**

We completed the project in **5 working days**, with each day dedicated to a specific module or milestone.

#### **Day** **Planned Tasks**

- Project requirement analysis
- Day 1** - Creation of tables: Admission Table, Student Progress Table, Salesforce Table  
- Initial form setup
- Designing form layouts
- Day 2** - Adding appropriate fields with validations (choice list, email, date)  
- Establishing reference relationships between tables
- Day 3** - Workflow and flow designer implementation  
- Automating status changes and approval processes  
- Creating UI Policies for dynamic field behavior
- Day 4** - Testing of forms and workflows  
- Data entry and record submission trials  
- Adjustments based on performance checks

## Day Planned Tasks

	-	Final	screenshots	capture
Day	-	Project documentation		preparation
5	-	Review of the complete application in ServiceNow instance		

## Planning Tools Used

- **Task Distribution:** Done among team members via Google Sheets
- **Daily Progress Tracking:** Manual check-ins with the team leader
- **Milestone Verification:** By verifying step completion inside the ServiceNow instance

## Efficiency Strategy

To stay on track, we followed these strategies:

- Prioritized **essential features only** (e.g., form setup, field validation, workflows).
- Avoided over-complication by focusing on **core functionality**, not extensive customization.
- Completed testing alongside development to save time and ensure quality.

## **6. FUNCTIONAL AND PERFORMANCE TESTING**

### **6.1 Performance Testing**



## 6.1 Performance Testing

Performance testing ensured that the system performs reliably under normal usage conditions. Since this is a lightweight educational application, we focused on **form load time**, **workflow execution speed**, and **UI responsiveness**.



### Parameters Evaluated:

#### 1. Form Load Time

- Admission and Student Progress Forms consistently loaded in **under 2 seconds**.
- No noticeable delays when switching between records.

#### 2. Workflow Execution Time

- Workflow for admission approval executed in **less than 3 seconds**.
- Real-time status updates were reflected immediately upon form submission.

#### 3. Field Responsiveness

- Choice lists, date fields, and reference fields responded instantly.
- No lag while switching between visible/invisible fields via UI Policies.

#### 4. Browser Compatibility

- Tested on Chrome and Microsoft Edge. All forms rendered correctly without layout breaks.

## 5. Record Submission Stress Test

- Submitted 20+ records in quick succession to ensure no data loss or crash.
  - The system handled the load efficiently.
- 



### Testing Methodology:

- **Manual Testing** using the ServiceNow instance.
  - **Exploratory Testing** to uncover edge cases and UI issues.
  - **Validation Testing** for fields like Email, Date of Birth, and Grades.
  - **End-to-End Workflow Testing** to ensure logical flow from form fill to approval to record update.
- 



### Outcome

- **All major functionalities passed** both positive and negative test scenarios.
- No workflow or validation issues were found during multiple test iterations.
- The application is stable and ready for deployment and demonstration.

## **7. Results**

### 7.1 Output Screenshots

## 7.1 Output Screenshots

Our results include:

- Screenshot of the filled **Admission Form**
- **Table view** showing all student records
- **Flow designer** view showing approval path
- **Reports** showing enrolled students by course

Each screenshot proves the correctness of design and functionality.

The image displays three vertically stacked screenshots of a Salesforce application interface, likely a custom-built system for managing student admissions. Each screenshot shows a different page of the process.

- Salesforce New record:** This screen shows the initial data entry stage. It includes fields for Admin Number (SAL0001078), Admin Date, Grade (set to None), Student Name, Father Name, Mother Name, Mother Cell, Father Cell, and a prominent blue "Submit" button.
- Admission New record:** This screen shows the progression of the admission status through various stages: New, In progress, Joined, Rejected, Rejoined, Closed, and Canceled. The "In progress" stage is currently active. It contains fields for Admission Number, Purpose of Join (set to None), Student Name, Father Name, Mother Name, Admin Status (set to None), Fee (\$ 0.00), Father Cell, Mother Cell, and a Comments section. Below this, there are tabs for "School Details" and "Address", with "School Area" and "School" dropdown menus. A "Submit" button is also present.
- New Section New record:** This screen shows the final data entry stage for a new section. It includes fields for Admin Number, Grade (set to None), Student Name, Father Name, Mother Name, Father Cell, and Mother Cell. Below this, a "Student Progress" section is expanded, showing fields for Telugu, Hindi, English, Maths, Science, and Social, along with Total, Percentage, and Result fields. A "Submit" button is located at the bottom.

## **8. Advantages & Disadvantages:**

Every software system has its set of strengths and limitations based on the platform, time, and scope. During the development of our project, we identified the following advantages and disadvantages based on our experience, user feedback, and system testing.

### **Advantages:**

#### **1. Centralized Data Management**

All student admission records, academic progress details, and inquiry data are stored in one place. This eliminates the need for physical records or multiple spreadsheets, making data management easy, quick, and organized.

#### **2. Workflow Automation**

ServiceNow's powerful workflow engine allowed us to automate admission status updates and data handling. This reduced manual intervention and saved time for administrative staff.

#### **3. User-Friendly Interface**

With form-based data entry, dropdowns, and clean layouts, users can easily interact with the system even without technical training. Field validations ensure accurate data submission.

#### **4. Real-Time Updates and Notifications**

Status changes and form submissions trigger real-time updates within the platform. This ensures all users stay informed about their actions and reduces delays.

## **5. Low-Code Development**

ServiceNow supports rapid development with minimal coding. This made it easy for our team to build and customize the application even with limited programming experience.

## **6. Highly Scalable and Extensible**

The system can be expanded to include new tables, forms, dashboards, or integrate with external tools (e.g., email, payment gateway, or analytics), making it future-proof for institutional use.

### **Disadvantages:**

#### **1. Limited Offline Access**

As ServiceNow is a cloud-based platform, users require internet access to interact with the system. Offline access or mobile use in areas with low connectivity becomes a limitation.

#### **2. Initial Learning Curve**

New users, especially non-technical staff, may need time to understand how to navigate modules, update records, or use workflows effectively. Some training is required at the start.

#### **3. Customization Complexity**

While basic features are low-code, advanced customization (e.g., scripted policies, external integration) may require knowledge of JavaScript or Glide scripting.

## **4. Platform Dependency**

The application is built entirely on ServiceNow, which means institutions need to maintain an active ServiceNow instance and subscription to use or scale the system.

## **5. License Constraints**

For real-world institutional use, ServiceNow licensing costs may be a barrier for smaller organizations. While the educational version is free during the internship, future use could incur costs.

## **9. CONCLUSION:**

The **Educational Organization Management System**, developed using the **ServiceNow platform**, has successfully demonstrated how low-code platforms can simplify and streamline core administrative tasks in educational institutions. Through this project, we addressed real-time challenges faced by administrative staff such as managing student admissions, tracking academic progress, and maintaining structured digital records.

By leveraging key ServiceNow components like **tables**, **forms**, **choice lists**, **workflows**, and **UI policies**, we created a functional and user-friendly solution that reduces manual workload, minimizes data entry errors, and enhances overall efficiency. The project has provided a practical understanding of how automation and digital transformation can improve transparency, responsiveness, and data accessibility within an academic setup.

Throughout this development process, our team gained valuable exposure to platform-based development, testing methodologies, and documentation standards. The hands-on experience with ServiceNow also helped us understand its capabilities in delivering real-world business solutions with minimal code. In summary, this project is not just a working prototype—it is a scalable foundation upon which more modules (like fee management, feedback system, attendance monitoring, etc.) can be added in the future to create a comprehensive ERP for educational institutions.

## 10. FUTURE SCOPE

While the current version of the **Educational Organization Management System** meets the basic administrative needs of an educational institution, there is significant potential to extend its features and usability. This section outlines the enhancements that can be implemented in future phases to make the system more comprehensive, intelligent, and user-centric.

---



### 1. Role-Based Dashboards

Implementing customized dashboards for different user roles (students, faculty, and admin) would greatly improve user experience. These dashboards could include:

- Real-time admission stats for admins
- Class performance tracking for faculty
- Enrollment status, attendance, and grades for students

Each user would only see the information relevant to their responsibilities or needs.

---



### 2. Mobile Access and Responsive UI

Developing a responsive interface or integrating a **mobile application** would make the platform accessible from smartphones and tablets. This would be particularly useful for:

- Faculty updating marks on the go

- Students checking their admission status or grades
  - Admins monitoring records without being restricted to desktops
- 

### 3. Fee Management System Integration

A future enhancement could include a **fee collection and tracking module**. This could:

- Allow students to pay fees online
  - Generate automatic receipts
  - Notify defaulters and alert the finance team  
This would bring financial operations under the same digital umbrella.
- 

### 4. AI & Analytics for Student Performance

Integrating **AI-based analytics** tools would help in identifying patterns in student performance, attendance trends, and academic gaps. Predictive insights could assist faculty in taking corrective actions early.

---

### 5. Email and Notification Integration

Though ServiceNow supports internal notifications, integrating email or SMS alerts would ensure:

- Students receive real-time updates

- Admins are alerted on pending actions
- Faculty are notified of performance updates

This feature can improve response time and engagement.

---

## 6. Third-Party System Integration

Institutions may already use systems like Learning Management Systems (LMS), library databases, or payment gateways. In future phases, this platform can be integrated with:

- Google Classroom or Moodle
  - Razorpay, Paytm, or UPI for payments
  - DigiLocker or Aadhaar-based ID validation
- 

## 7. Chatbot Support

A simple chatbot (within ServiceNow or externally integrated) could help answer frequently asked questions like:

- "What is my admission status?"
- "How can I update my profile?"
- "Where can I see my marks?"

This would reduce repetitive queries to the admin team and increase system engagement.

## **11.Appendix:**

The Appendix serves as a supplementary section that provides supporting materials, technical resources, and additional documentation used during the development of the project. While the main content explains what the system does and how it was built, the appendix helps verify, reference, and understand implementation details for further use or review.

---

### **11.1 Source Code and Configuration**

Since the project was developed entirely on the ServiceNow platform, most of the development work was done through configurations rather than traditional coding. However, certain parts such as UI policies, business rules, and field validations involved logic and scripts.

- **Update Set File:** All configurations (tables, forms, workflows) were saved in an Update Set within ServiceNow for export/import.
- **Client Scripts:** Used for email validation and form field behaviors.
- **UI Policies:** Applied to show/hide specific fields based on conditions (e.g., show remarks when Grade is ‘F’).
- **Flow Designer:** Configured to automate admission status changes and record updates.

 *Note:* These items remain within the ServiceNow instance and are not uploaded to an external repository unless explicitly exported.

---

## **11.2 External Links (If applicable)**

Github & Project Demo Link: