# Complete Capstone Project Document

# Problem Statement: Student Admission & Enrollment Management System

Educational institutions often face challenges in handling large numbers of student inquiries, applications, and enrollments. The process is usually scattered across emails, spreadsheets, and manual record-keeping, leading to inefficiency and loss of potential students.  
  
The lack of a centralized system for managing admissions causes delays, duplicate efforts, and poor communication with students. There is a need for an automated, transparent, and efficient admission and enrollment management system.  
  
This project proposes to develop a Salesforce-based Student Admission & Enrollment Management System to manage the end-to-end process. The system will track inquiries, automate admission follow-ups, monitor application progress, and record final enrollment decisions.

# Phase 1: Problem Understanding & Industry Analysis

👉 Goal: Understand what we’re building and why.

- Requirement Gathering: Talk to stakeholders (admission officers, students, faculty).

- Stakeholder Analysis: Identify roles like Admission Officer, Student, Administrator.

- Business Process Mapping: Student inquiry → Application → Review → Admission → Enrollment.

- Industry Analysis: Admission processes often delayed by manual handling and poor tracking.

- AppExchange Exploration: Look for education CRM apps but design a custom lightweight version.

# Phase 2: Org Setup & Configuration

👉 Goal: Prepare Salesforce environment.

- Use Salesforce Developer Edition for building.

- Set up institution profile, local time zone, currency.

- Define business hours (e.g., 9am–6pm).

- Configure roles: Admin > Admission Officer > Staff.

- Set Org-Wide Defaults: Applications private, Students public.

- Create users with specific profiles and permissions.

- Set up sandbox and deployment basics.

# Phase 3: Data Modeling & Relationships

👉 Goal: Design data structure for the system.

- Objects: Student, Application, Enrollment.

- Fields: Student (Name, Email, Contact No), Application (Status, Program, Date), Enrollment (Course, Fees, Start Date).

- Relationships: Student ↔ Application (Lookup), Application ↔ Enrollment (Master-Detail).

- Page Layouts: Student page shows related Applications, Application page shows Enrollment.

- Use Schema Builder for visualization.

# Phase 4: Process Automation (Admin)

👉 Goal: Automate repetitive admission tasks.

- Validation Rules: Ensure required fields are filled (e.g., Application must have a Program).

- Flows: Auto-update Application Status when Admission Approved.

- Approval Process: High-value scholarship applications need Admin approval.

- Email Alerts: Notify students on application status changes.

- Tasks: Assign tasks to Admission Officers for pending reviews.

- Notifications: In-app alerts for new inquiries.

# Phase 5: Apex Programming (Developer)

👉 Goal: Add advanced logic with Apex.

- Create Apex classes for application logic.

- Triggers: Prevent duplicate applications for same student and program.

- SOQL Queries: Fetch list of pending applications.

- Batch Apex: Nightly job to mark overdue applications.

- Queueable Apex: Handle bulk email notifications.

- Test Classes: Ensure triggers and logic work correctly.

# Phase 6: User Interface Development

👉 Goal: Make the system user-friendly.

- Build custom Lightning App: 'Student Admission System'.

- Design Record Pages for Student, Application, Enrollment.

- Add Tabs for Applications and Enrollments.

- Dashboard on Home Page showing inquiries and admissions stats.

- LWC: Component to search applications by status.

- Navigation: Direct users to Application record after submission.

# Phase 7: Integration & External Access

👉 Goal: Connect system with external apps.

- Integration with Email/SMS gateway for student notifications.

- Named Credentials for secure API storage.

- External Services: Connect with student verification APIs.

- REST API callouts for checking student data externally.

- Salesforce Connect for external student records if needed.

# Phase 8: Data Management & Deployment

👉 Goal: Manage and deploy system data.

- Data Import Wizard: Import student records.

- Data Loader: Bulk upload applications.

- Duplicate Rules: Prevent duplicate students.

- Data Export & Backup for safety.

- Deployment via Change Sets or SFDX.

# Phase 9: Reporting, Dashboards & Security Review

👉 Goal: Monitor system usage and ensure security.

- Reports: Applications by Status, Enrollments per Program.

- Dashboards: Admission conversion rates.

- Dynamic Dashboards: Each officer sees only their data.

- Field Level Security: Hide sensitive student data.

- Audit Trail for monitoring changes.

# Phase 10: Final Presentation & Demo Day

👉 Goal: Deliver final project with documentation and demo.

- Pitch: Problem → Solution → Benefits.

- Demo: Walkthrough inquiry to enrollment.

- Documentation: Share user guide and system design.

- Portfolio: Showcase project on LinkedIn/GitHub.