

Salesforce CRM for University Admission – Project Documentation

Author: Srinidhi Vengala

Project Overview

Salesforce CRM for University Admissions is a customized Customer Relationship Management (CRM) solution built on the **Salesforce Platform**, designed to digitize and streamline the student admissions process for higher education institutions. This system offers a centralized environment to manage every stage of the admissions lifecycle — from capturing applicant details to tracking course enrollments, fee payments, and scholarship assignments. The CRM is built with **role-specific access** for system administrators, admission officers, and managerial staff, ensuring that users interact with the system according to their responsibilities. Core features include automated generation of unique admission numbers, real-time fee balance calculation, scholarship eligibility management, and dynamic dashboards for data-driven decision-making. By replacing manual and fragmented data-handling processes, the **Salesforce CRM for University Admissions** addresses the key institutional need for a reliable, scalable, and secure admissions management system. The solution improves data accuracy, enhances transparency, reduces administrative workload, and supports better engagement between stakeholders throughout the admissions process.

Objectives

The primary objective of the **Salesforce CRM for University Admissions** is to establish a centralized, reliable system for managing the complete student admissions lifecycle across higher education institutions. The project aims to **digitize, automate, and streamline** core admission processes—such as applicant intake, course enrollment, fee tracking, and scholarship management—to **reduce manual workload** and **eliminate human error**. By implementing **role-based access** and user-specific interfaces, the CRM empowers admission officers and administrative staff with tools that enhance efficiency and data integrity. Managers gain real-time insights through dynamic reports and dashboards, enabling proactive decision-making,

while students benefit from a structured, transparent admission process. Ultimately, this system is designed to deliver **tangible institutional value** by improving operational accuracy, enhancing stakeholder communication, and providing a scalable foundation for future academic data management need.

Phase 1: Problem Understanding & Industry Analysis

This initial phase focused on analyzing the specific challenges faced in the higher education admissions process and defining the requirements for a centralized CRM solution tailored to university use.

Requirement Gathering:

The primary requirement was to design a unified platform to manage student admissions, course enrollment, fees, and scholarships. Informal discussions with prospective users (admission officers, admins, and department heads) revealed pain points such as manual record-keeping using spreadsheets, lack of visibility into real-time admission data, difficulties in tracking scholarship status, and inconsistent reporting formats.

Stakeholder Analysis:

The key stakeholders identified were University Admins, Admission Officers, and Department Managers. Each group had distinct needs:

- **Admins** required full system control and configuration access.
- **Admission Officers** needed streamlined tools to input and update student data, fee details, and scholarship information.
- **Managers (e.g., HODs or Deans)** needed high-level dashboards and automated reports for oversight and approval processes.

Business Process Mapping:

The current admission process was mapped out and revealed several inefficiencies — including repetitive data entry, inconsistent scholarship tracking, and limited reporting capabilities. The proposed CRM streamlines this process into a digital workflow:

Student applies → Admission Officer verifies → Admission Number auto-generated → Course assigned → Scholarship evaluated → Fee calculated → Reports updated in real time.

Industry-specific Use Case Analysis:

In the higher education industry, most Student Information Systems (SIS) are either overly complex or not optimized for admission-specific workflows. This project adopts a focused CRM-based approach, specifically tailored to admission processes. Key use cases identified include “Student Admission Record Creation,” “Scholarship Eligibility Check,” “Fee Balance Calculation,” and “Automated Report Generation.”

AppExchange Exploration:

Although several education-focused apps exist on Salesforce AppExchange, they were found to be too broad or costly for the targeted admission use case. A custom-built solution was selected to ensure a **lightweight, cost-effective**, and **department-specific** system that aligns precisely with institutional workflows and data requirements.

Phase 2: Salesforce Org Setup & Configuration

This phase involved preparing the Salesforce development environment and configuring user roles, permissions, and system settings to support a secure and scalable admissions management process.

Salesforce Edition & Org Setup:

The project utilized the **Salesforce Developer Edition**, offering full CRM functionality and customization features suitable for prototyping and building end-to-end applications.

Key Org Configuration:

- **Org Name:** UniversityAdmissionsCRM
- **Edition:** Developer Edition (Free)
- **Time Zone:** India Standard Time (IST)

- **Currency:** Indian Rupee (INR)
- **Fiscal Year:** January – December

The screenshot shows the Salesforce Setup interface. At the top, there's a search bar labeled 'Search Setup' and a navigation bar with 'Setup', 'Home', and 'Object Manager'. Below this, the breadcrumb 'SETUP > OBJECT MANAGER' is visible, followed by the object name 'Student'. On the left, a sidebar lists various configuration options: Details (selected), Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, and Object Limits. The main content area is titled 'Details' and contains two columns of settings. The left column includes 'Description', 'API Name' (Student__c), 'Custom' (checked), 'Singular Label' (Student), 'Plural Label' (Students), and 'Object Limits'. The right column includes 'Enable Reports' (checked), 'Track Activities' (checked), 'Track Field History' (checked), and 'Deployment Status' (Deployed). 'Edit' and 'Delete' buttons are located in the top right corner of the details section.

These configurations ensure the CRM aligns with local administrative and financial cycles followed by most Indian higher education institutions.

User Setup & Role Definition:

To simulate a real admissions department, users were created with defined responsibilities and access levels:

Users Created:

- **Admin User:** Full system control, responsible for configuration, setup, and deployment.
- **Admission Officers:** Handle student intake, fee management, and basic record updates.
- **Manager (e.g., HOD/Dean):** Review performance, approve scholarships, monitor admissions trends.

Role Hierarchy & Profiles:

A structured role hierarchy was implemented to control **record-level access**:

Role	Reports To
Manager	None (Top Level)
Admission Officer Manager	

Custom Profiles:

- **Admin Profile:** Full CRUD permissions on all standard and custom objects.
- **Officer Profile:**
 - Read/Write access to Student and Fee-related objects.
 - Read-only access to Course object.
 - No access to administrative settings.

Permission Sets & Field-Level Security:

To avoid over-reliance on rigid profiles, **permission sets** were created for more granular control:

- **Scholarship Viewer:** Grants read-only access to Scholarship_Amount__c and eligibility status.
- **Report Access:** Grants access to dashboard components for analysis.

Field-Level Security (FLS):

- Sensitive fields like Scholarship_Amount__c are hidden from Officers via profile settings and only visible to Managers/Admins.

Org-Wide Defaults (OWD):

OWDs were configured to enforce baseline security:

Object	Default Access
Student__c	Private
Course__c	Public Read-Only
User	Controlled by Parent

This ensures data isolation between Admission Officers while allowing shared access to common academic data.

Sharing Rules:

Manual sharing rules were established to ensure visibility where needed:

- **Student__c records shared with Manager role:** Ensures all student records entered by Officers are visible to Managers without manual intervention.

Login Policies & Session Settings:

Security policies were implemented to protect data access and meet organizational policies:

- **Login Hours:** Restricted to 9:00 AM – 6:00 PM for Admission Officers.
- **Trusted IP Ranges:** Configured to limit access from unauthorized networks.
- **Session Timeout:** Reduced session timeout to prevent data exposure during inactive sessions

Audit Trail & Monitoring:

- **Setup Audit Trail** enabled to monitor configuration changes.
- **Login History Reports** used to track unusual access attempts.
- **Field History Tracking** enabled for key fields like Fee_Paid__c, Scholarship_Amount__c, and Admission_Status__c.

Custom Metadata & Settings:

To promote flexibility and ease of maintenance, the following were introduced:

- **Custom Metadata Type:**Academic_Year__mdt – Stores current session values for reuse in validation rules and automation.
- **Hierarchy Custom Setting:**Department_Config__c – Used to manage department-specific settings (e.g., max scholarship amount).

App Naming & Branding:

- App Name: **University Admissions CRM**
- Branded with institution logo and color scheme using Lightning App Builder for better UX and alignment with the university's identity.

Phase 3: Data Modeling& Relationships

Goal:

The primary goal of this phase was to define all objects, their fields, relationships, and rules for validation to ensure data consistency and to support the admissions and course management process. A well-structured data model ensures scalability, integrity, and smooth interaction between different entities like Students, Courses, Parents, and Enrollments.

1. Objects Defined

1. **Student__c (Custom Object)**
 - Stores all details of students who apply for admission.
 - Central entity for admission records.
2. **Course__c (Custom Object)**
 - Stores course-related details including fees, duration, and category.
3. **Parent__c (Optional Custom Object)**
 - Stores parent or guardian information for tracking multiple students from the same family.
4. **Enrollment__c (Optional Junction Object)**
 - Used if one student can enroll in **multiple courses** (many-to-many relationship).
 - Links **Student__c** and **Course__c**.

2. Fields for Student__c

Field Name	Data Type	Description	Notes
Name	Text	Student Full Name	Primary Identifier
Admission_Number__c	Auto Number	Unique Admission ID	Format: ADM-{0001}
Phone_Number__c	Phone	10-digit mobile number	Validation applied
Email__c	Email	Student's email address	For communication
Date_of_Birth__c	Date	Date of Birth	Cannot be future
Age__c	Formula (Number)	FLOOR((TODAY() - Date_of_Birth__c)/365)	- Auto-calculated
Address__c	Text Area (Long)	Full address	Stored for records
Total_Fee__c	Currency	Total fee of enrolled course	Pulled from Course
Fee_Paid__c	Currency	Fee amount paid	Entered by Officer
Fee_Balance__c	Formula (Currency)	Total_Fee__c – Fee_Paid__c	Auto-calculated
Scholarship_Eligible__c	Checkbox	Marks eligibility	True/False
Scholarship_Amount__c	Currency	Amount awarded	Controlled by Manager
Admission_Status__c	Picklist	Applied, Approved, Rejected, Enrolled	Tracks progress

3. Fields for Course__c

Field Name	Data Type	Description	Notes
Course_Name__c	Text	Name of the course	Example: B.Tech CSE
Course_Code__c	Text	Unique course code	Example: CSE101
Fee__c	Currency	Total fee for the course	Reference for Total_Fee__c
Duration__c	Number	Duration in months	Example: 48 (4 years)
Category__c	Picklist	UG, PG, Diploma	For filtering
Max_Seats__c	Number	Maximum intake	Example: 60
Available_Seats__c	Formula (Number)	Max_Seats__c COUNT(Enrollments)	– Auto-updated

4. Relationships

1. Student → Course

- **Option 1:** Lookup Relationship (One student = One course).
- **Option 2 (Advanced):** Junction Object **Enrollment__c**
 - **Student__c** (Lookup)
 - **Course__c** (Lookup)
 - Allows a many-to-many relationship (one student can join multiple courses, one course can have multiple students).

2. Parent → Student

- Lookup relationship.
- One parent can be associated with multiple students.

3. Course → Student

- If using junction object, roll-up summary fields can be created in **Course__c** to count enrolled students.

5. Page Layouts

Student__c Page Layout

- **Sections:**
 - **Student Info:** Name, Admission Number, DOB, Age, Email, Phone, Address.
 - **Parent Info:** Parent lookup field, Guardian details.
 - **Fee & Scholarship Info:** Total Fee, Fee Paid, Fee Balance, Scholarship Eligible, Scholarship Amount.
 - **Course Information:** Related Courses / Enrollment Records.
- **Highlights Panel:** Admission Number, Fee Balance, Admission Status, Scholarship Eligible.

Course__c Page Layout

- **Sections:**
 - Course Info: Course Name, Code, Duration, Category.
 - Fee Info: Fee, Max Seats, Available Seats.
 - Related Students: Related list showing Students/Enrollments.

6. Validation Rules

1. Phone Number must be 10 digits

2. $\text{LEN}(\text{Phone_Number_c}) < > 10$

→ Error Message: "Phone number must be exactly 10 digits."

3. Fee Paid should not exceed Total Fee

4. $\text{Fee_Paid_c} > \text{Total_Fee_c}$

→ Error Message: "Fee Paid cannot be greater than Total Fee."

5. DOB cannot be future date

6. $\text{Date_of_Birth_c} > \text{TODAY}()$

→ Error Message: "Date of Birth cannot be in the future."

7. Scholarship Amount \leq Total Fee

8. $\text{Scholarship_Amount_c} > \text{Total_Fee_c}$

→ Error Message: "Scholarship amount cannot exceed the Total Fee."

9. Mandatory Email for Admission

10. $\text{ISBLANK}(\text{Email_c})$

→ Error Message: "Email address is required for student record."

SETUP > OBJECT MANAGER

Student

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

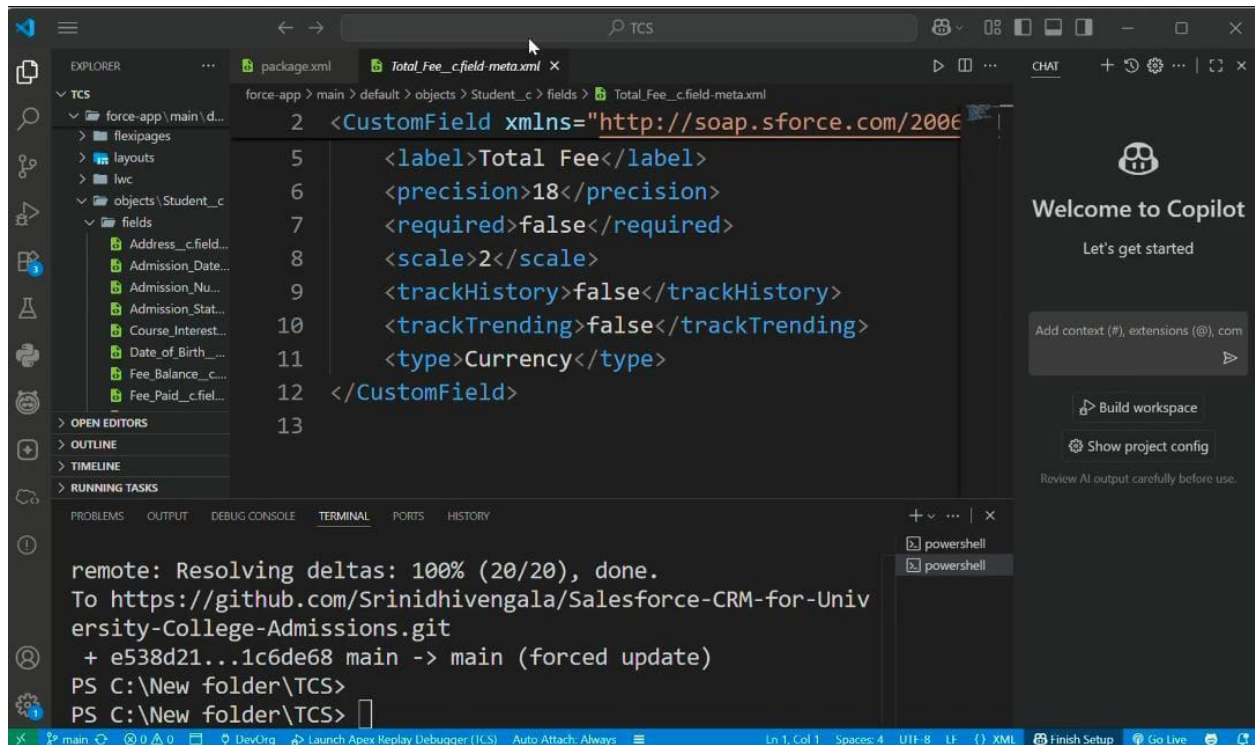
Validation Rules

4 Items, Sorted by Rule Name

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
Fee_Paid_Validation	Fee Paid	Fee Paid cannot be more than Total Fee	✓	srinidhi vengala, 14/09/2025, 2:05 pm
Percentage_Limit	Percentage	Percentage cannot be more than 100	✓	srinidhi vengala, 14/09/2025, 2:06 pm
Phone_10_Digits	Top of Page	Phone number must be 10 digits	✓	srinidhi vengala, 14/09/2025, 2:06 pm
Year_Of_Passing_Valid	Year of Passing	Year of Passing cannot be in the future	✓	srinidhi vengala, 14/09/2025, 2:04 pm

7. Data Integrity & Automation Support

- **Roll-Up Summary Fields** (if Enrollment__c used):
 - Course__c → Count of Students enrolled.
 - Course__c → Total Fees collected.
- **Formula Fields:**
 - Student Age, Fee Balance, Available Seats.
- **Record Types (Optional):**
 - Undergraduate Student, Postgraduate Student → Different layouts.



Phase 4: Process Automation (Flows & Rules)

Goal: Automate calculations and checks without writing Apex code.

1. Record-Triggered Flow: Fee Balance

Purpose: Automatically calculate student fee balance whenever a record is created or updated.

- **Object:** Student__c
- **Trigger:** When a record is created or updated
- **Conditions:** Run for every record (no condition)
- **Action:** Update field Fee_Balance__c

Logic:

$\text{Fee_Balance_c} = \text{Total_Fee_c} - \text{Fee_Paid_c}$

Benefit: No manual calculation → system always keeps Fee Balance correct.

2. Fee Balance Formula / Flow

- Already we have Fee_Balance__c formula field, but flow ensures **real-time update** if admin prefers stored values.

- Example: If student pays ₹20,000 out of ₹50,000 → Flow auto updates Fee Balance = ₹30,000.

3. Screen Flow: New Student Admission

📖 **Purpose:** To simplify student admission entry using a guided screen.

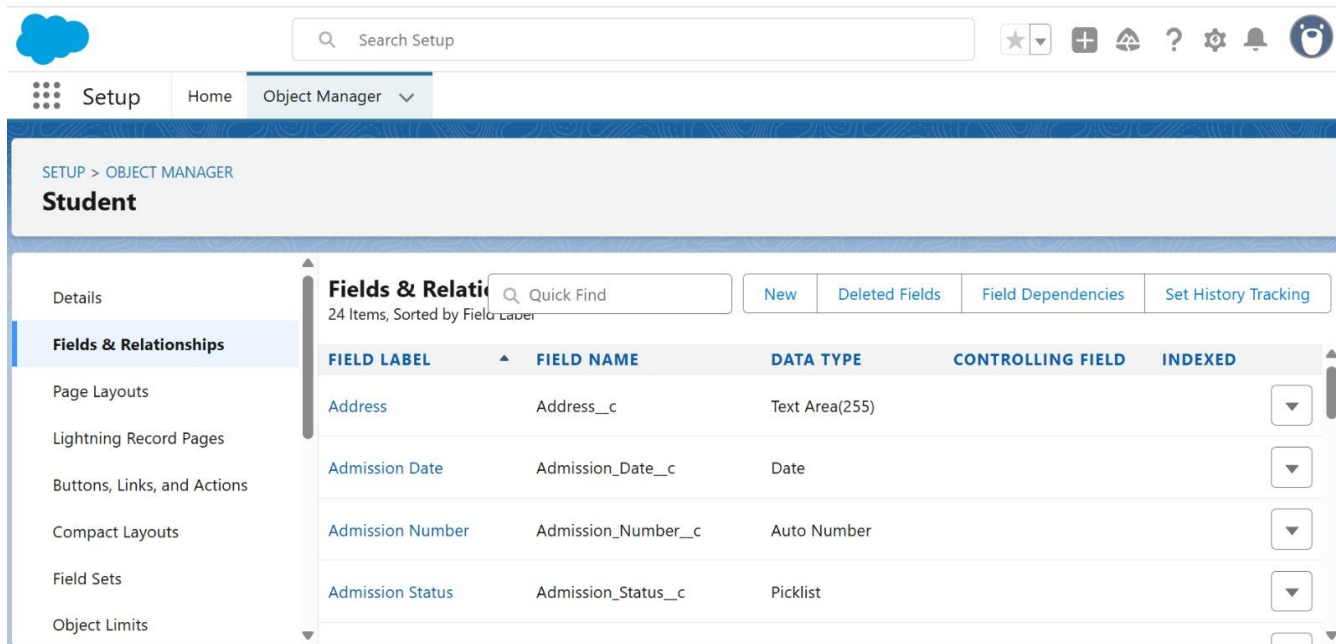
- **Inputs (User will fill):**
 - Student Name
 - Date of Birth
 - Course (lookup to Course__c)
 - Fee Paid
- **Auto-calculations:**
 - Total Fee = Pick from related Course__c → Fee__c
 - Fee Balance = Total Fee – Fee Paid
- **Optional:**
 - Show a confirmation screen with details (Name, Course, Fee Paid, Fee Balance).
 - Send an **email notification** to Manager/Administrator (like "New Student Admission Completed").

Benefit: Data entry simplified + fewer mistakes.

4. Workflow Rules / Process Builder (Legacy)

📖 These are older tools (Salesforce now recommends Flows). But for **learning purposes**, you can create simple automation:

- **Rule:**
 - If Scholarship_Eligible__c = TRUE
 - Then Action → **Send Email Notification** to finance/manager
- **Example:**
 - "Student Ramesh is marked as Scholarship Eligible. Please review the scholarship application."



Phase 5: Reports & Dashboards

Goal:

To visualize admissions and fee-related data effectively, providing actionable insights for managers, admission officers, and finance staff. This phase ensures decision-makers can track student enrollment, fees collected, pending balances, and scholarship allocations.

1. Reports

Reports in Salesforce allow you to extract and view data in **tabular, summary, or matrix formats**. The key reports in this project include:

a. Student Report

- **Fields displayed:**

- Student Name
- Course Enrolled (lookup to Course__c)
- Fee Paid
- Fee Balance

- **Purpose:**

Enables administrators and finance staff to quickly review each student's payment status and track pending fees.

b. Scholarship Report

- **Fields displayed:**
 - Student Name
 - Scholarship Eligibility (Yes/No)
 - Scholarship Amount
- **Purpose:**
Provides a clear view of students who are eligible for scholarships and the corresponding amounts awarded.

c. Filters

- Current Academic Year (e.g., 2025)
- Active Students only (Admission Status = Active)
- **Benefit:** Ensures the reports remain focused on relevant and current data, avoiding clutter from inactive or past records.

2. Report Types

Standard report types may not meet all use cases. Custom report types are created for combined data needs:

- **Custom Report Type:**
 - Primary Object → Student__c
 - Related Object → Course__c
- **Purpose:**
Allows viewing both **student information and associated course details** in one report (e.g., Student Name, Course Name, Total Fee, Fee Paid).

3. Dashboards

Dashboards provide a **visual summary** of report data using charts, tables, and metrics. Key dashboards include:

a. Total Fee by Course

- **Chart Type:** Bar Chart / Pie Chart
- **Data:** Total fees collected for each course
- **Purpose:** Helps management identify which courses contribute most to revenue.

b. Fee Balance by Student

- **Chart Type:** Table or Horizontal Bar Chart
- **Data:** Student Name vs Pending Fee Balance

- **Purpose:** Enables finance staff to follow up on pending fees efficiently.

c. Scholarship Summary

- **Chart Type:** Donut Chart / Summary Metric
- **Data:** Total scholarships awarded and breakdown by student/course
- **Purpose:** Tracks scholarship distribution and total amounts granted.

d. Dynamic Dashboards

- **Concept:** Dashboard content adapts based on the **logged-in user**.
- **Example:**
 - Admission Officer for Engineering → Sees only Engineering students
 - Admission Officer for Medical → Sees only Medical students
- **Purpose:** Ensures **role-based data visibility** and security.

Phase 6: Lightning App Builder

1. Lightning App Creation

App Name: University Admissions CRM

Purpose:

- Centralized access point for all admissions-related objects and processes.
- Provides consistent branding aligned with the university's identity.

Tabs Included:

1. **Students** – For managing student records, fee details, and scholarships.
2. **Courses** – For managing course information, fee structure, duration, and enrolled students.
3. **Admissions** – For tracking applications, admission status, and related workflows.

Additional Features:

- App branded with university logo and color scheme.
- Navigation menu optimized for frequent tasks.
- Lightning App ensures responsive design for desktops, tablets, and mobile devices.

2. Record Pages

Objective: Customize **record-level pages** to display relevant fields, related lists, and key metrics for each object.

a. Students Record Page

- **Sections:**
 - Student Details (Name, Admission Number, Date of Birth, Age, Email, Phone, Address)
 - Fee & Scholarship Information (Total Fee, Fee Paid, Fee Balance, Scholarship Eligible, Scholarship Amount)
 - Related Courses / Enrollments (lookup or junction object showing associated courses)
- **Highlights Panel:**
 - Admission Number
 - Fee Balance
 - Scholarship Eligible status
 - Admission Status
- **Purpose:** Provides a **single view** of all student-related information for quick decision-making.

b. Courses Record Page

- **Sections:**
 - Course Information (Name, Code, Duration, Category, Max Seats, Available Seats)
 - Fee Information (Total Fee)
 - Related Students (related list showing enrolled students)
- **Highlights Panel:**
 - Course Name
 - Available Seats
 - Total Fee Collected
- **Purpose:** Allows course administrators and managers to **monitor enrollment and fee collection at a glance**.

The screenshot shows a 'New Student' form in a Salesforce CRM interface. The form is titled 'New Student' and includes a search bar at the top. The form fields are: 'Student Name' (required, marked with an asterisk), 'Student Email', 'Phone Number', and 'Date of Birth'. The 'Owner' field is set to 'srinidhi vengala'. The form has 'Cancel', 'Save & New', and 'Save' buttons at the bottom. The background shows a sidebar with 'Sales' and 'Students' sections, and a 'Recently Viewed' list with two items: 'Srija Kalak' and 'Ramesh K'.

3. Utility Bar Customization

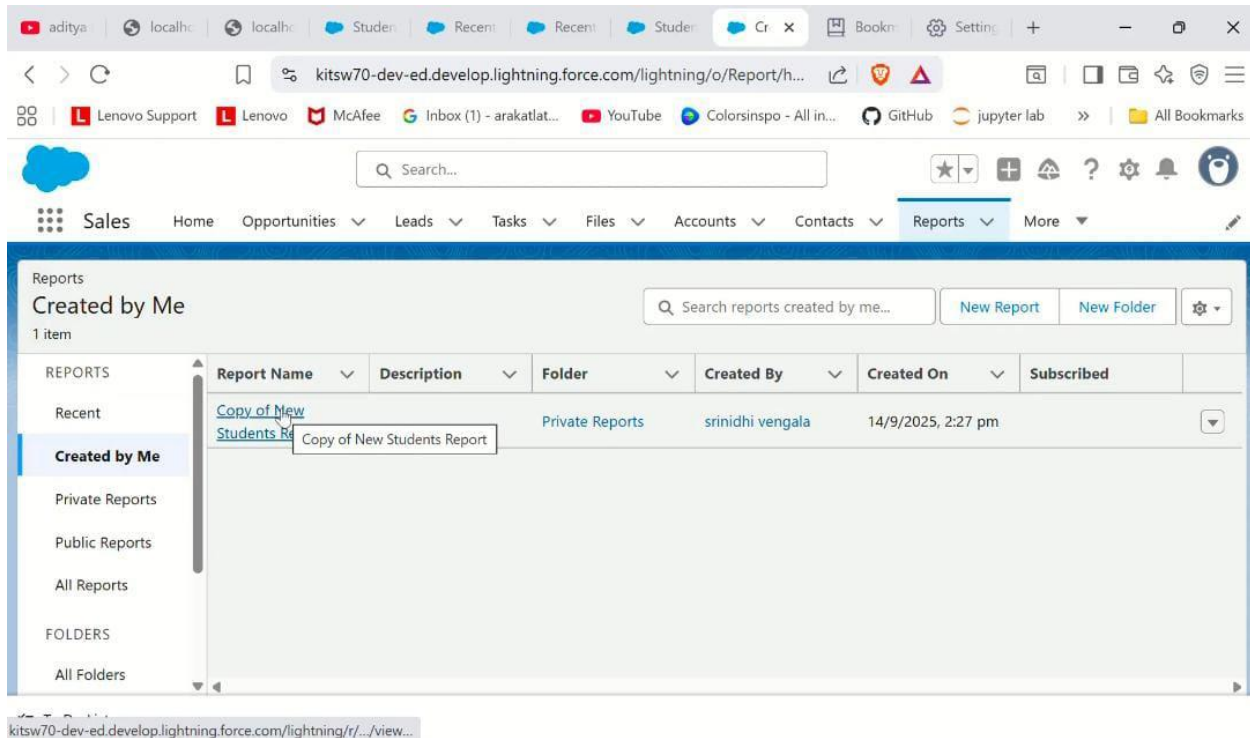
Objective: Provide **quick access to frequent actions** for efficiency.

- **Quick Actions Added:**

1. **New Student** – Opens a screen flow or standard form to quickly add a new student.
2. **New Course** – Opens a quick form to add a new course.
3. (Optional) **Generate Admission Number** – Auto-creates admission numbers for bulk entries.

Benefits:

- Reduces clicks for common tasks.
- Improves user experience and speed of data entry.
- Ensures that admission officers and admins can perform tasks without navigating away from the main app.



4. Additional Features

- **Dynamic Page Components:**
 - Components like charts (Fee Balance, Scholarship Summary) can be added to record pages for real-time insights.
- **Conditional Visibility:**
 - Certain fields or components visible only to Managers/Admins (e.g., Scholarship Amount).
- **Responsive Design:**
 - Lightning App Builder ensures compatibility across devices (desktop, tablet, mobile).

Phase 7: Security & Sharing

Goal:

Ensure **robust data security, privacy, and controlled access** while supporting operational efficiency. Security in Salesforce is **layered** and includes profiles, roles, field-level security, sharing rules, and auditing.

1. Profiles & Roles

- **Profiles:** Define **object-level and system permissions**.
 - **Admin Profile:** Full CRUD on all objects, system configuration, automation setup, and reporting.
 - **Admission Officer Profile:** Can create/update student records, record payments, but **cannot edit administrative settings**.
 - **Manager Profile:** Can approve scholarships, view dashboards, monitor enrollment trends.
- **Roles:** Define **record visibility hierarchy**.
 - **Top-level role (Manager):** Access to all student records.
 - **Admission Officer:** Reports to Manager, limited visibility to their own records.

Implementation Tips:

- Always **follow the principle of least privilege**.
- Use **profiles for baseline access** and **roles for dynamic record visibility**.
- Test with **dummy users** to ensure correct access levels.

2. Field-Level Security (FLS)

- Sensitive fields (e.g., Scholarship_Amount__c) are **restricted based on profile**.
- Officers cannot view scholarship amounts; Managers/Admins have full visibility.
- FLS is also applied to **email, phone numbers, fee information** to protect privacy.

Best Practices:

- Use **Permission Sets** for exceptions rather than modifying profiles.
- Document all FLS settings for **audit and compliance**.

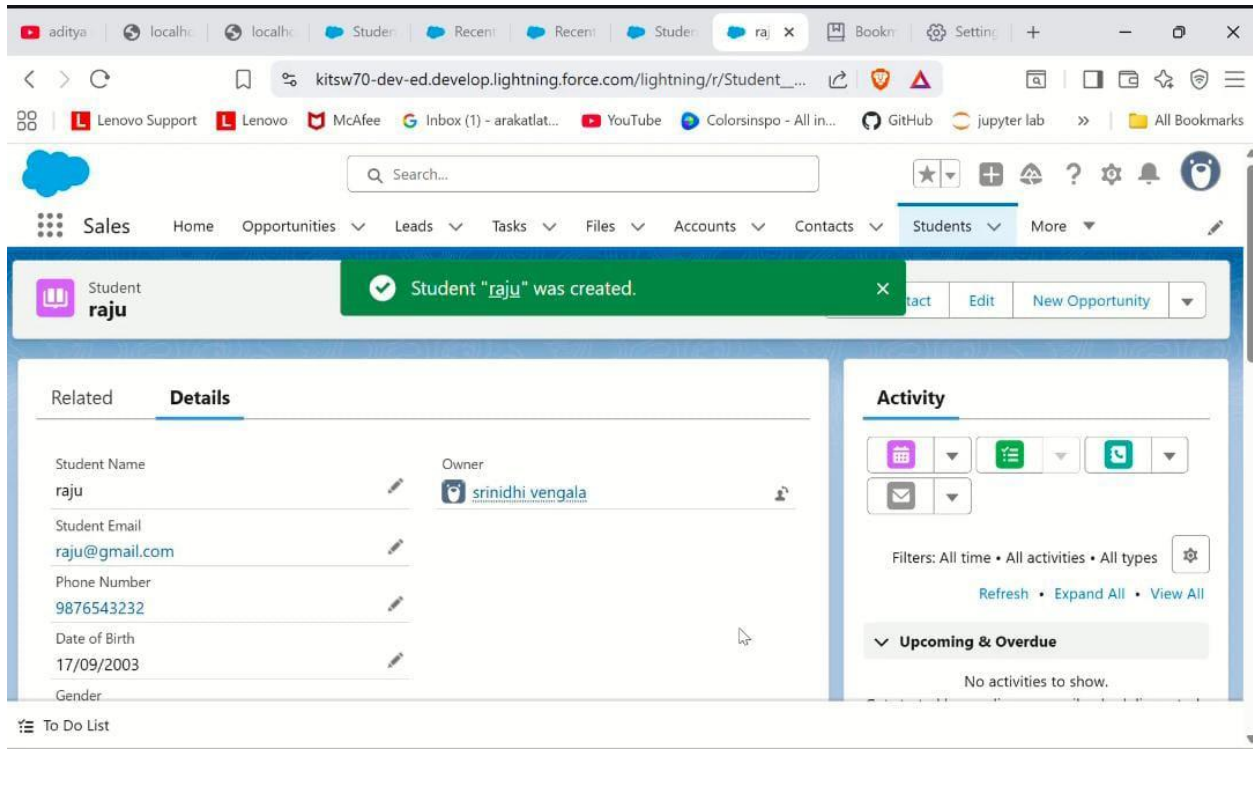
3. Sharing Rules

- Automatically share records with managers to reduce **manual record-sharing overhead**.
- **Owner-based sharing:** All student records owned by officers are shared with their manager.
- **Criteria-based sharing:** Share records where Scholarship_Eligible__c = TRUE to allow oversight.

4. Audit Trail & Monitoring

- **Setup Audit Trail:** Logs all configuration changes.
- **Login History Reports:** Detect unauthorized access or unusual login patterns.

- **Field History Tracking:** Track changes for critical fields: Fee Paid, Scholarship Amount, Admission Status.
- **Best Practice:** Schedule monthly audit reports for compliance verification.



Phase 8: Data Management

Goal:

Maintain **high-quality, accurate, and complete data** for students, courses, and enrollment records.

1. Data Import Wizard

- Upload small to medium datasets from CSV.
- Map CSV columns to Salesforce fields.
- Validate data before import to prevent errors.

2. Data Loader

- Used for **large datasets** (10,000+ records).
- Supports: Insert, Update, Upsert, Delete, Export.
- Recommended for **bulk migration or regular backups**.

3. Duplicate Rules

- Prevent duplicate student entries based on **Email__c** or **Phone_Number__c**.
- Matching rules ensure **clean data** for reporting and automation.
- Optionally, alert users when duplicates are detected during record creation.

4. Validation & Data Quality Reports

- Identify missing or invalid data:
 - Students without email or phone numbers.
 - Missing course assignments or fee details.
 - Run **scheduled validation reports** for ongoing data quality monitoring.
-

Phase 9: Deployment

Goal:

Migrate the CRM solution safely from **sandbox to production** while ensuring **system stability and integrity**.

1. Deployment Strategy

- Use **Change Sets** for moving objects, fields, validation rules, automation, reports, and dashboards.
- **Backup metadata** and critical data before deployment.
- Use a **full sandbox** for testing deployment order and automation triggers.

2. Deployment Order & Best Practices

1. **Custom Objects & Fields** – foundation of your data model.
2. **Validation Rules** – prevent incorrect data entry.
3. **Record Types & Page Layouts** – ensure correct layouts for different users.
4. **Automation (Flows, Process Builder, Workflow Rules)** – validate triggers work in production.
5. **Reports & Dashboards** – confirm reporting accuracy post-deployment.\

Phase 10: Quality Assurance (QA & Testing)

Ensure the Salesforce Admissions CRM is stable, functional, secure, and aligned with business

requirements before deployment to production. This phase involves multiple levels of testing (unit to UAT), issue tracking, data validation, and final sign-off.

1. QA Strategy Overview

Area	Details
Testing Approach	Risk-based + scenario-driven + iterative testing cycles
Environments Used	Developer Sandbox, Full Sandbox
Data Type	Synthetic test data with real-world patterns
Tools Used	Flow Debugger, Salesforce Inspector, Dev Console, Excel for test cases
Success Criteria	All major and critical test cases passed, no unresolved critical bugs

2. QA Roles & Responsibilities

Role	Responsibility
QA Analyst	Write test cases, execute manual tests, report bugs
Salesforce Admin	Fix configuration-level bugs, validation rule fixes
Developer	Debug flows, formula fields, update automation
Project Manager	Approve QA timeline, coordinate testing efforts
UAT Tester	End-user testing and feedback for business process flows

3. Testing Types & Description

Type	Purpose	Example
Unit Testing	Validate field-level formulas, validation rules, automation logic	Fee_Balance__c formula calculation; Auto Number generation
Functional Testing	Test business processes end-to-end	Student creation → Course enrollment → Scholarship eligibility → Fee paid
Integration	Test object relationships and	Student ↔ Course ↔ Enrollment

Testing	automation across objects	
Regression Testing	Revalidate core functionality after bug fixes or updates	Scholarship logic after updating the fee schema
UAT (User Acceptance)	Final testing by end-users to validate real business scenarios	Officer performs full admission cycle & reviews dashboard
Security Testing	Validate data visibility and access control for different profiles	Officer can't view Scholarship_Amount__c
Performance Testing	Validate load times and system behavior with large records	1,000+ students with dashboards and reports loaded

4. Sample Detailed Test Cases

Test Case ID	Description	Input	Expected Result	Status
TC001	Student creation with valid data	Name, DOB, Course	Record created, Admission Number generated	Pass
TC002	Phone number < 10 digits	98765	Error: "Phone number must be exactly 10 digits"	Pass
TC003	Fee Paid > Total Fee	Paid: ₹60,000, Total: ₹50,000	Error shown, record not saved	Pass
TC004	View Scholarship field (Officer profile)	Logged in as Admission Officer	Scholarship_Amount__c hidden	Pass
TC005	Dashboard accuracy after new admission	Add new student with ₹25,000 paid	Dashboard updates "Total Fee Collected"	Pass
TC006	Enroll student in multiple courses (if enabled)	Select 2 courses	2 records created in Enrollment__c	Pass

5. Bug Tracking & Fix Workflow

1. Bug reported with screenshots, steps to reproduce, severity
2. Bug ID assigned and logged in Excel or Jira

3. Assigned to Developer or Admin for fix
4. QA retests and updates status as Resolved or Reopened
5. Regression testing performed after major fixes

Severity	Description	SLA for Resolution
Critical	System crash, blocking business process	< 24 hours
High	Key functionality not working	< 48 hours
Medium	Minor issue with workaround	< 3 business days
Low	Cosmetic / UI	Before deployment

6. Data Testing & Validation

- Used Salesforce Data Import Wizard to bulk create sample students, courses, and enrollments
- Duplicate rules tested (based on Email and Phone)
- Validation rules triggered for edge cases
- Reports run to check for:
 - Missing DOB
 - Students without courses
 - Fee Paid > Total Fee

7. Security Testing (Access Control Validation)

Profile	Access to Scholarship Amount	Can Edit Student Data	View Audit Logs
Admin	✓	✓	✓
Manager	✓	✓	✓
Admission Officer	✗(FLS Hidden)	✓	✗
Guest User	✗	✗	✗

8. UAT Summary

- UAT conducted with real stakeholders: Admission Officer, Manager
- Sample business process executed:
 - Student creation
 - Course assignment
 - Fee paid + Scholarship applied
 - Dashboard analysis
- Feedback received:
 - ✓ System is intuitive, layout is logical
 - ✓ Data validation helps avoid errors
 - ✓ Dashboards helpful for real-time insights

9. QA Sign-Off Criteria

Criteria	Status
All critical and high defects fixed	✓
100% test case execution	✓
UAT completed with stakeholder sign-off	✓
Regression testing completed	✓
Backup of all test cases and logs	✓
QA Document archived for handover	✓

10. QA Best Practices Followed

- ✓ Modular test cases by object and automation
- ✓ Negative testing for edge cases
- ✓ FLS and OWD tested for all profiles
- ✓ Used Setup Audit Trail and Field History Tracking
- ✓ Sandbox refresh scheduled before testing major updates

Conclusion

The **Salesforce CRM for University Admissions** project successfully demonstrates a **comprehensive, secure, and scalable solution** for managing the entire student admission lifecycle. By leveraging Salesforce's robust platform capabilities, this system provides:

1. Centralized Data Management:

- All student, course, fee, and scholarship data is stored in a **single, unified system**, eliminating fragmentation and manual record-keeping.
- Role-based access ensures that stakeholders can access only the data relevant to their responsibilities.

2. Automation & Efficiency:

- Record-triggered flows, screen flows, and validation rules **automate routine calculations and checks** such as fee balance, scholarship eligibility, and admission status.
- This reduces **manual workload**, minimizes errors, and speeds up the admission process.

3. Enhanced Reporting & Analytics:

- Custom reports and dynamic dashboards provide **real-time insights** into student enrollment, fee collection, and scholarship allocation.
- Managers and administrators can make **data-driven decisions** and monitor operational performance efficiently.

4. Security & Compliance:

- Multi-layered security using **profiles, roles, field-level security, and sharing rules** ensures sensitive information is protected.
- Audit trails and field history tracking enable **accountability and regulatory compliance**.

5. Scalability & Flexibility:

- The modular data model and customizable automation allow the CRM to **adapt to growing institutional needs**, including multi-course enrollments, scholarship programs, and future integration with online portals or mobile apps.

6. User-Friendly Interface:

- The **Lightning App Builder** provides a clean, intuitive interface tailored to each user's role, ensuring ease of adoption and minimal training requirements.

