Online Course Management CRM

# Phase 1: Problem Understanding & Industry Analysis

Abstract Regarding my project Idea: I am aiming to build a Salesforce project called Online Course Enrollment Manager CRM. The idea is to help small training institutes manage their students better and their processes easier ,like they can handle inquiries, enrollments, fees, and track how students are progressing in live classes.

In my project, whenever a student shows interest, I will consider it as a Lead. If the student joins, that Lead will be converted into an Enrollment, linked to a Course and a specific Class Session (batch). For every enrollment, Progress records will be created for tracking the student’s progress in each module. Trainers can update this progress, while Salesforce automation will send reminders for payments, classes, or if the student is falling behind.

* **Requirement Gathering**

1. Capture **student inquiries** (Leads).
2. Manage **Courses** (name, duration, fees, modules).
3. Manage **Trainer** Assignments to courses.
4. Run multiple **batches** for the same course.
5. Track **Enrollments** → each student joins a course session.
6. Monitor **Student Progress** → completion of modules.
7. Handle different student start times in the same course.
8. Automate **reminders** (fees due, missed classes, progress alerts).
9. Provide **dashboards** for enrollments, revenue, popular courses, dropout risk.

* **Stakeholder Analysis**

1. **Institute Owner/Admin** → needs a business overview: revenue, enrollments per course, dropout risks.
2. **Admission/Support Staff** → capture student leads, convert to enrollments, manage fees.
3. **Trainers/Faculty** → teach sessions, update student module progress, track attendance.
4. **Students** → inquire, enroll, attend sessions, receive automated reminders.

* **Business Process Mapping**

**Student Enquiry** (Lead Creation) 🡪**Qualify the Lead** (by checking on some parameters like did he attend the demo session etc.,)🡪**Lead Conversion** (when the student enrolls a particular course)🡪**Automate the Progress Records** (Generate the course modules progress records for the student)🡪**Trainer Updates the Progress of course** (upto which topic he had completed the course)🡪**Automate the Live session emails** 🡪**Generate dashboards and Reports for Admin Overview.**

* **Industry-specific Use Case Analysis**
* **Problem in Education Sector:** Training institutes find it hard to keep track of students joining at different times, leading to poor follow-ups, missed payments, and unnecessary student dropouts.
* Salesforce CRM Solution:
  1. Organize Courses → Run multiple Class Sessions/batches per Course.
  2. Capture & convert Leads into Enrollments.
  3. Track student-level Progress regardless of when they joined.
  4. Automate communication and reminders.
* Expected Benefits:
  1. No potential Lead will be missed.
  2. Clear view of active batches, students, and revenue.
  3. Improved student retention with timely reminders and personalized guidance based on performance tracking.
* **AppExchange Exploration**

1.Class Enrollment: Companies can use Class Enrollment for AppExchange to monitor student enrollment for the courses they offer and to track the internal — and external — classes their employees attend.

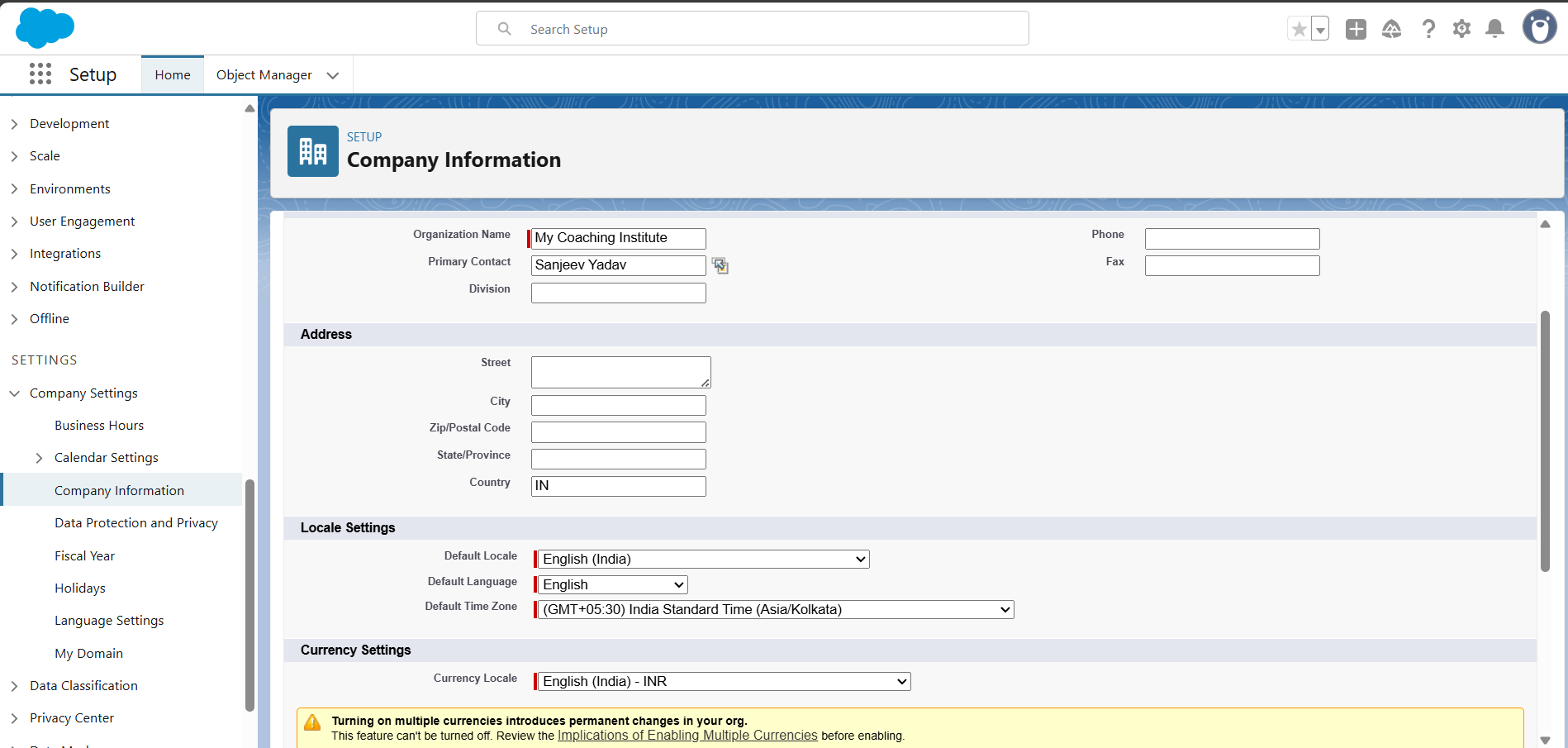
# Phase 2: Org Setup & Configuration

* **Salesforce Editions**

I am using the developer edition as it is freely available right now and is well suited for testing and developing.

* **Company Profile Setup**

Setup the company profile according to the requirements,

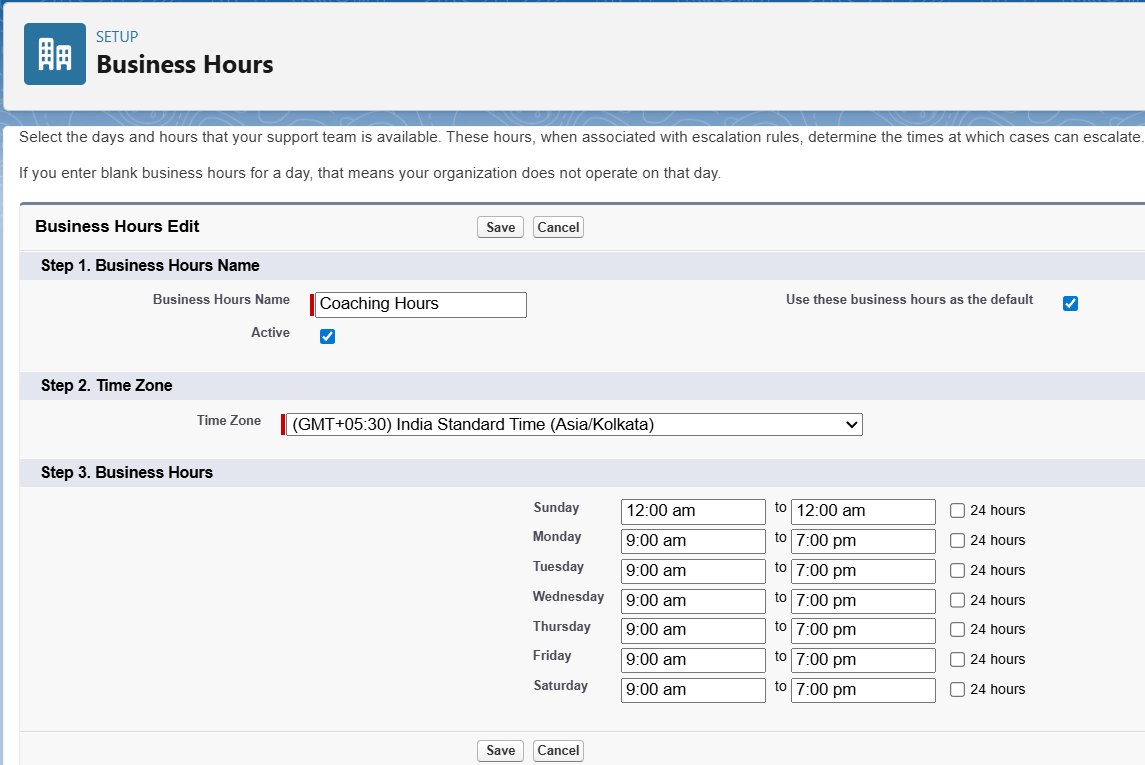


I have put the values of company info like this,

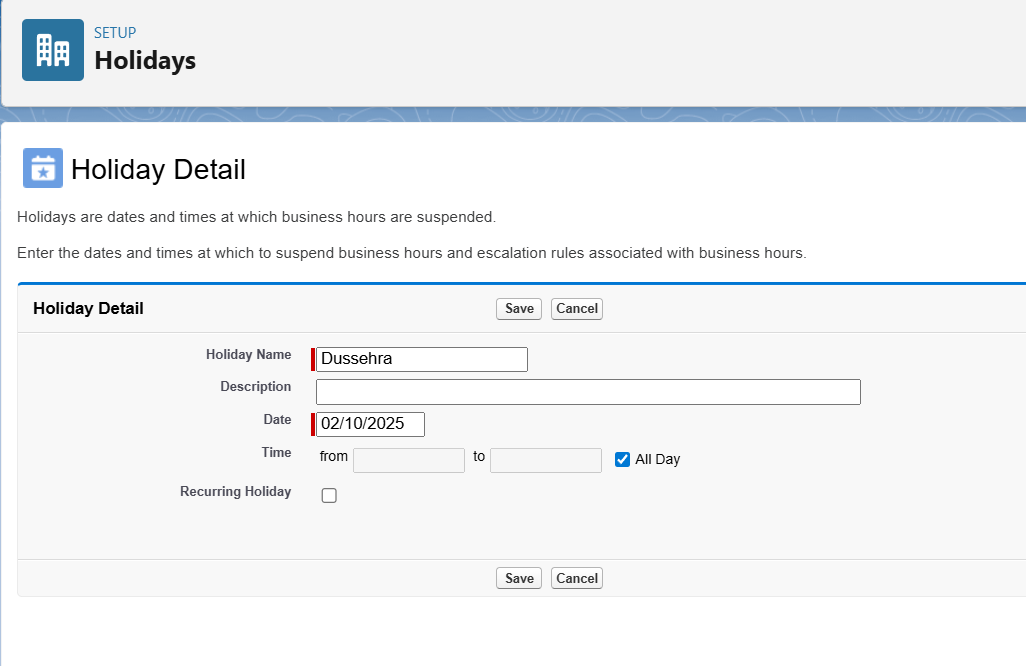
1. Company Name: *My coaching Institute*
2. Default Time Zone: IST (India Standard Time)
3. Default Currency: INR (₹)
4. Default Locale: English (India)

* **Business Hours & Holidays**

I have added some basic business hours like Monday-Saturday 9:00 am to 7:00pm.



I have added a holiday for Dussehra.



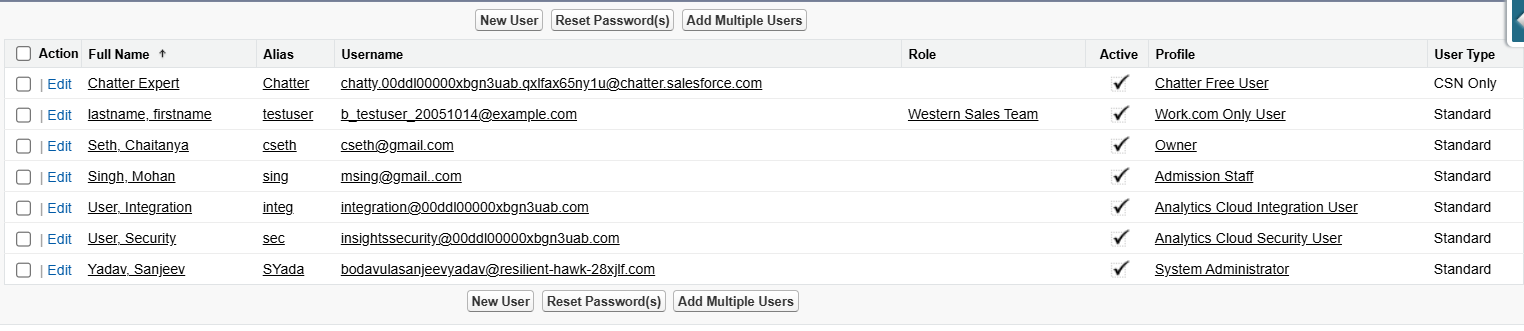
* **Fiscal Year Settings**

I used a standard fiscal year settings that is, January-December.

* **User Setup & Licenses**

I have Edited my user as System Admin.

I have added a new User Owner using the Owner Profile created Below.

I have added a new Admission Staff user using the Admission Staff profile I created.  
  
This is the users page after user creation using the profiles  


* **Profiles**

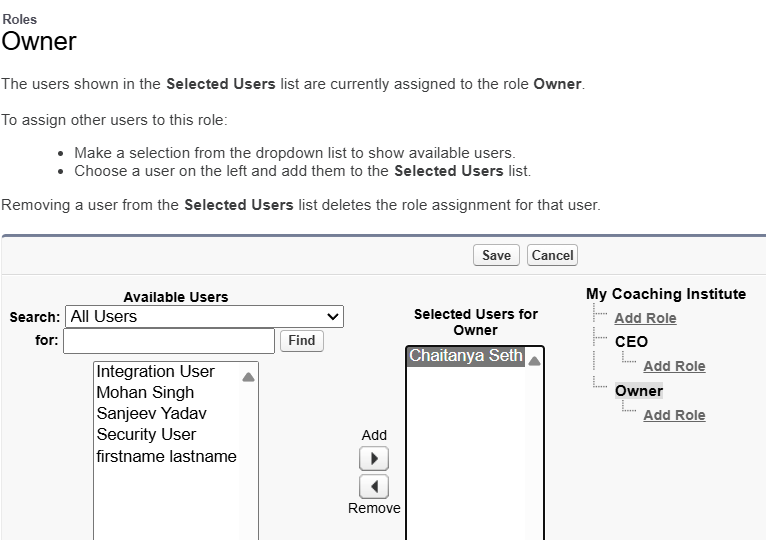
I have created 4 profiles according to my Org.

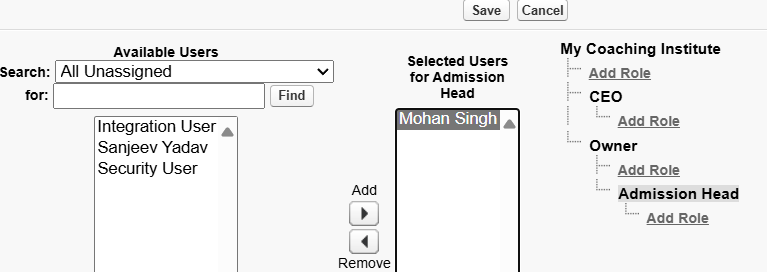
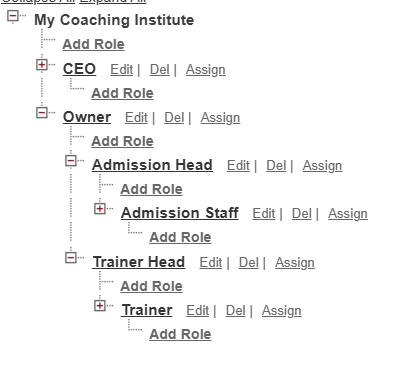
1.System Administrator:Has full access to all the objects.  
2.Owner: Access to all objects: Leads, Courses, Sessions, Enrollments, Progress,He can Run Reports & Dashboards but Cannot change setup.

3.Admission Staff:Has full access on Lead Objects , Has create ,read ,edit access on Enrollments,Has Read Access on courses ,batches,progress and Reports.

4.Trainer:Has Create,Read,Edit access on Progress,Has Read and Edit access on batches,Has Read Access on Enrollments and Leads.

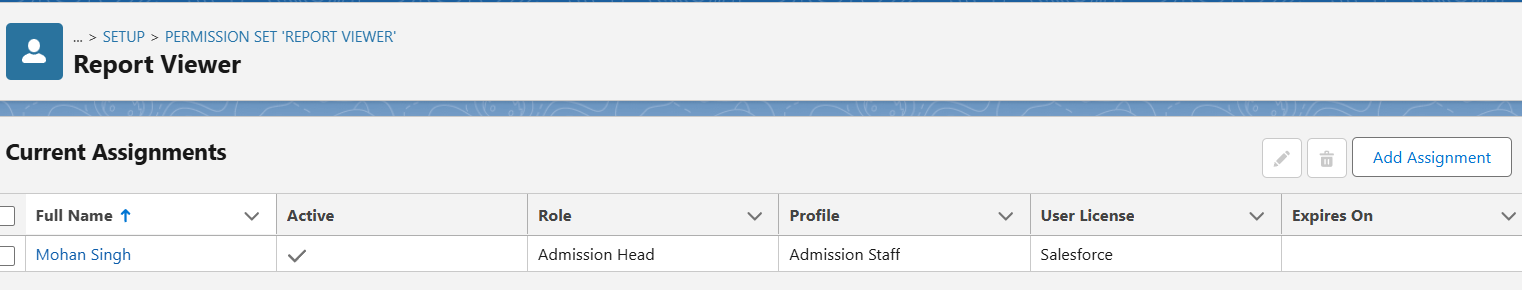
* **Roles**

🡪I have created the owner role and added the owner user to the role.  


🡪I have Added the Role **Admission Head** and added the Admission Staff user to it.  
  
  
🡪Similarly I have created **Admission staff,Trainer Head and Trainer Roles** also.  


* **Permission Sets**

🡪In my project, I created a **Report Viewer Permission Set** because I wanted my trainer user to be able to see the **Reports** and **Dashboards** tabs. The trainer profile I made didn’t have access to these tabs by default, so instead of changing the whole profile, I just added this extra permission set. After creating it, I assigned it to the trainer, and now the trainer can open and view reports and dashboards while still keeping their main profile limited to Progress updates.



* **OWD**

🡪In my project, I configured **Organization-Wide Defaults (OWD)** to control the baseline access to records across different users in the institute. Since I am working in a Salesforce Developer Edition with only internal users, I updated the **Default Internal Access** settings and left the external access settings unchanged.

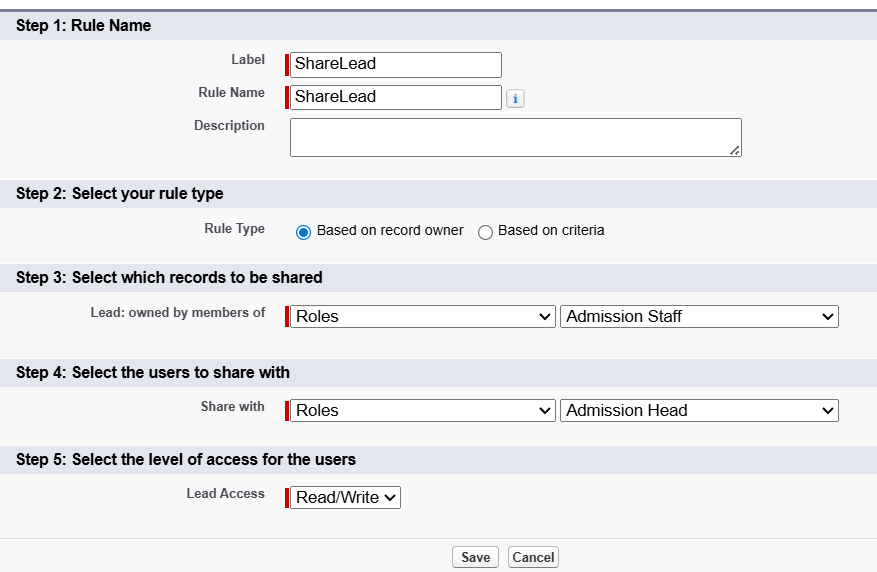
I set **Leads** to **Private**, so each admission staff member can only see their own leads, while managers and the owner can see them through the role hierarchy. I set **Courses** and **Class Sessions (Batches)** to **Public Read Only**, so that all users can view these records but only the owner or admin can edit them. For **Enrollments**, I chose **Private**, because each admission staff should only manage their own enrollments, while the trainer can see their assigned students through sharing and roles.

For the **Progress** object, I used a **Master-Detail relationship with Enrollment**, so it automatically inherits visibility from its parent Enrollment. Because of this setup, Progress did not appear in the OWD list, which is expected behavior in Salesforce.

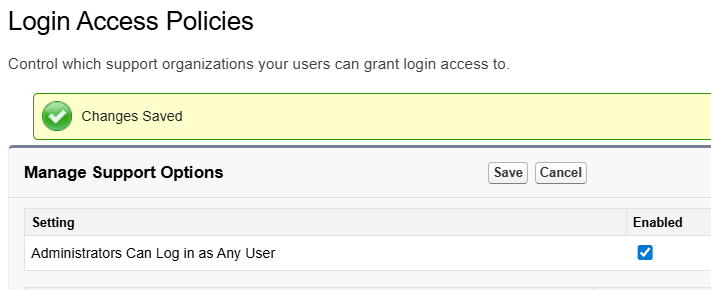
This configuration ensures that inquiry and student data remain secure and role-specific, while common information like courses and sessions stays visible to all.

* **Sharing Rules**

🡪I created a sharing rule for lead for Admission Head to be Able to access the leads of all the admission Staff.



* **Login Access Policies**

🡪I have enabled the Admin can login as any user option ,so that I can login and check different user perspectives of my org.  


* Dev Org Setup
* Sandbox Usage
* Deployment Basics

# Phase 3: Data Modeling & Relationships

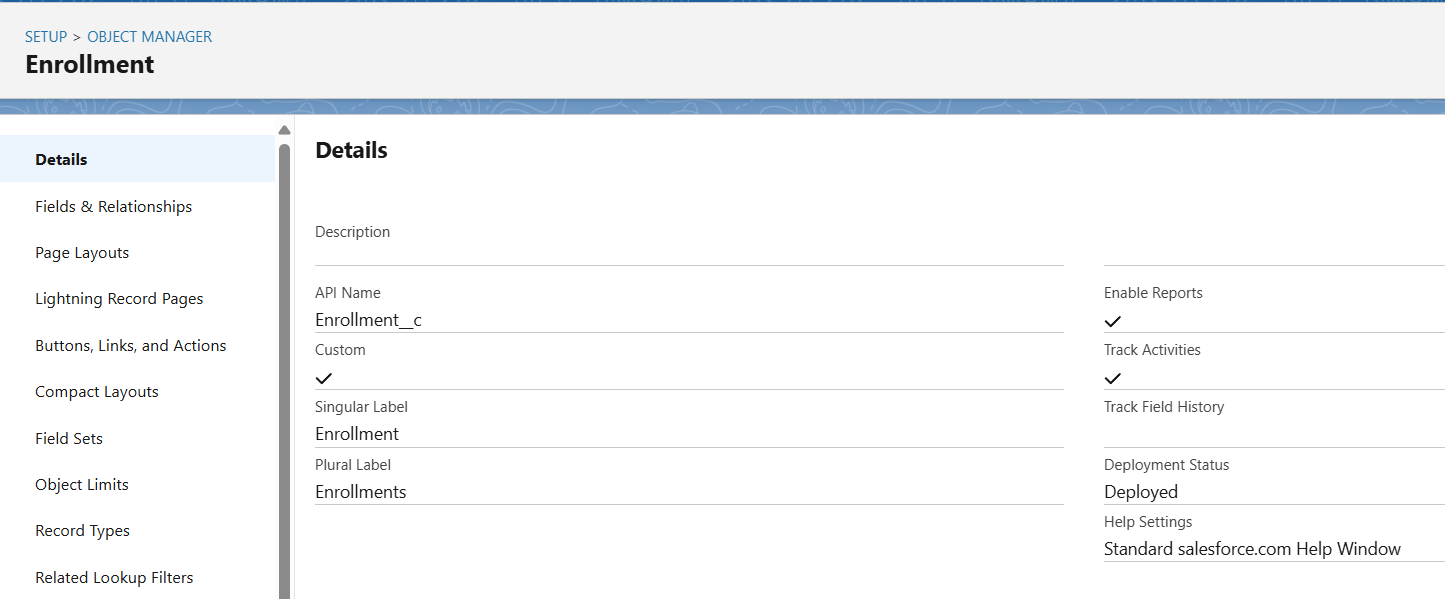
* **Standard & Custom Objects**

🡪The **Standard Objects** Used for the Org are:

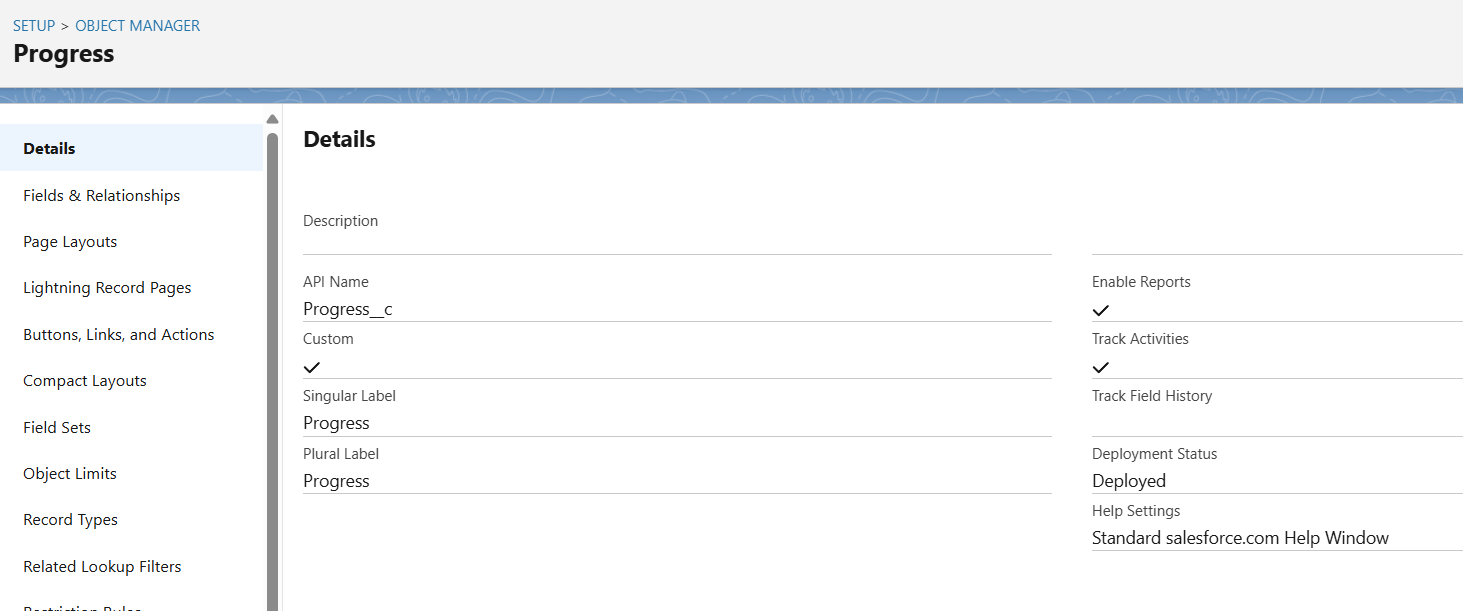
1. Lead: To capture the users who enquire regarding the courses.
2. Contact: To store details of Leads who are qualified.
3. User: To maintain all the Users of the institution.

🡪The **Custom Objects** Created for the Org are:

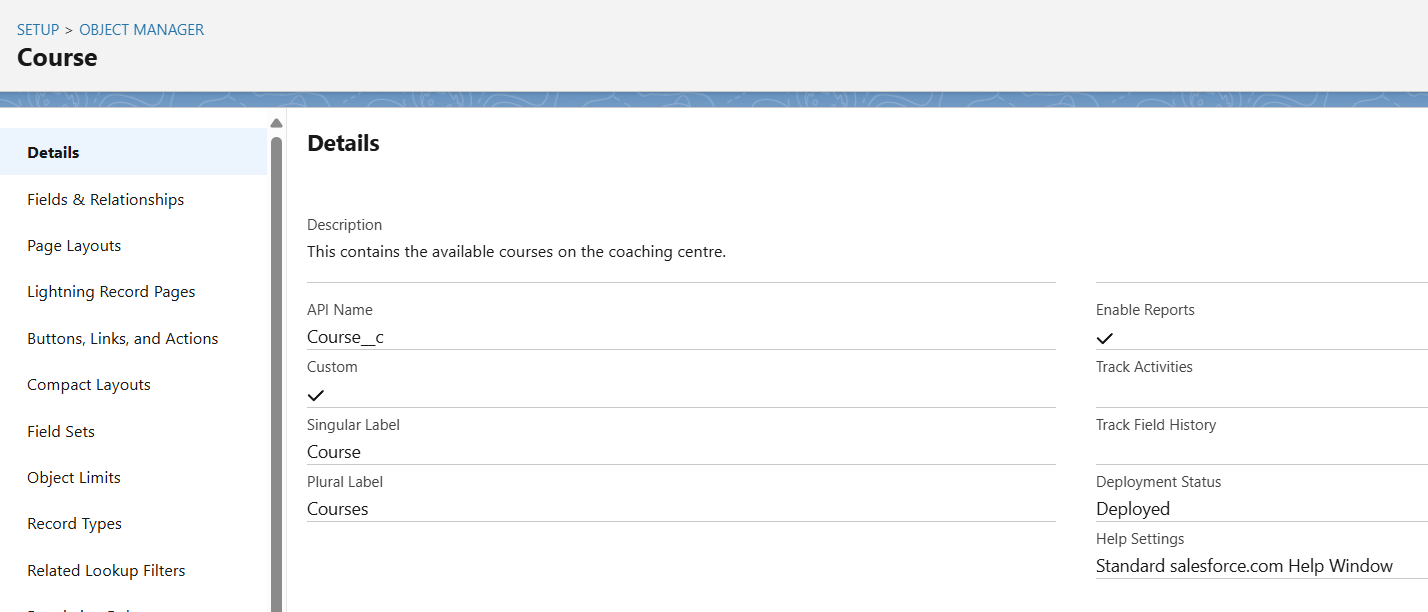
1. Enrollment: Tracks the Students Who are enrolled in any course given by the Institution.

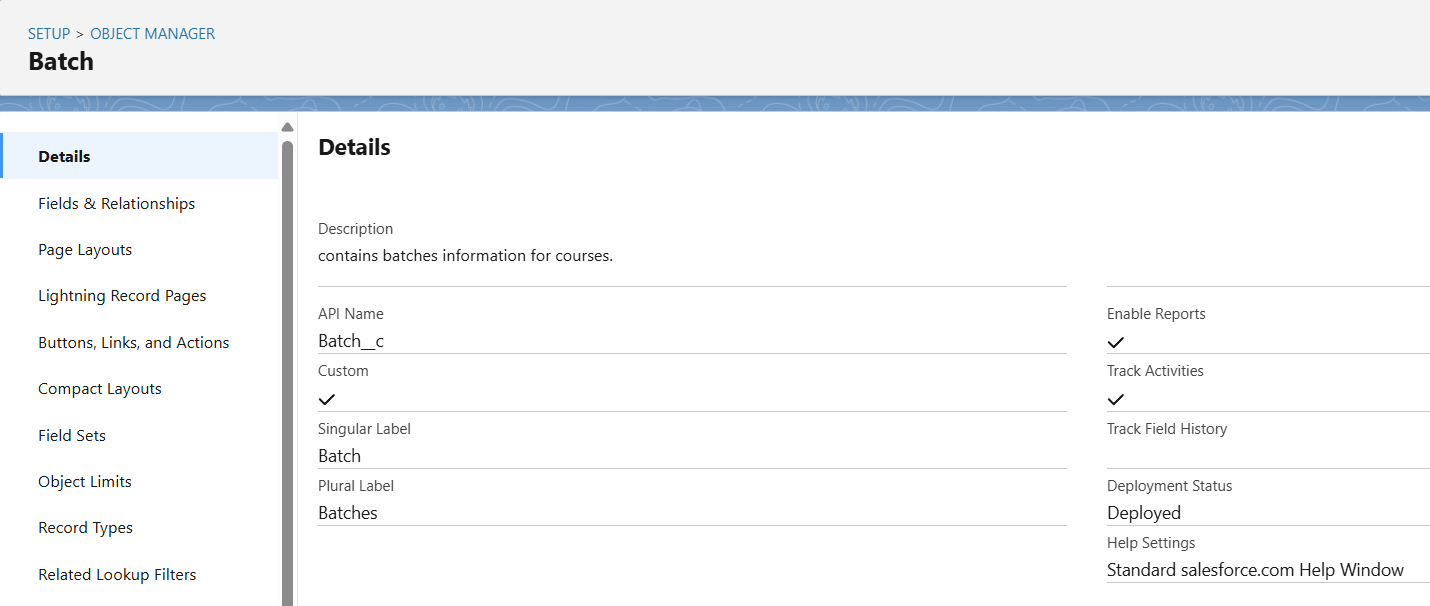


1. Progress: Tracks the Completion Progress of the course enrolled the student , It is maintained by the trainer teaching the particular Course.



1. Course: Contains the information regarding the courses available in the Institution.



1. Batch: Contains the information regarding the ongoing Course Batches at the institution.  
   

* **Fields**

🡪These are the fields that I make use of for the Standard Objects:

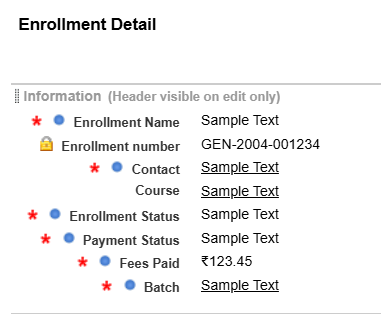
1. Lead
2. First Name (Standard)
3. Last Name (Standard)
4. Email (Standard)
5. Phone (Standard)
6. Lead Source (Picklist – Website, Walk-in, Referral, Social Media, Advertisement, Other)
7. Status (Picklist – New, Contacted, Interested, Qualified, Converted, Dropped)
8. Contact
9. First Name (Standard)
10. Last Name (Standard)
11. Email (Standard)
12. Phone (Standard)

🡪These are the fields that I used for the Custom Obejcts that I created:

1. Course
2. Course Name (Standard Name field – Text)
3. Category (Picklist – Tech, Language, Music, Other)
4. Duration (weeks) (Number)
5. Fees (Currency)
6. Description (Long Text Area)
7. Status (Picklist – Active, Inactive)
8. Batch
9. Session Name (Standard Name field – Text or Auto Number)
10. Course (Lookup → Course)
11. Trainer (Lookup → User)
12. Start Date (Date)
13. End Date (Formula: Start\_Date\_\_c + (Course\_\_r.Duration\_\_c \* 7) - 1)
14. Status (Picklist – Scheduled, Ongoing, Completed, Cancelled)
15. Enrollment
16. Enrollment Number (Standard Name field – Auto Number: ENR-{0000})
17. Student (Lookup → Contact)
18. Course (Lookup → Course)
19. Class Session (Lookup → Class Session)
20. Start Date (Date)
21. End Date (Date or Formula)
22. Payment Status (Picklist – Pending, Partial, Paid)
23. Enrollment Status (Picklist – Active, Completed, Dropped)
24. Fees Paid (Currency)
25. Progress
26. Progress Name (Standard Name field – Auto Number or Text)
27. Enrollment (Master-Detail → Enrollment)
28. Module Name (Text)
29. Completion Status (Picklist – Not Started, In Progress, Completed)
30. Completion % (Percent or Number)

* Record Types
* **Page Layouts**

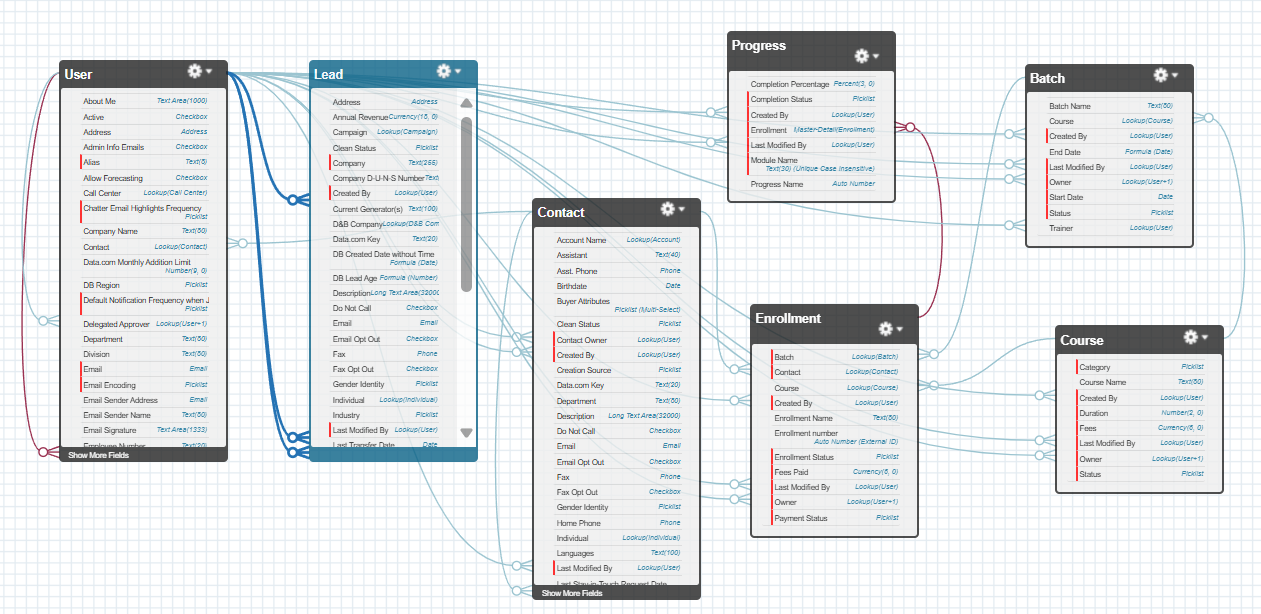
🡪 I worked with **Page Layouts** to arrange the fields properly so that users (like Admission Staff and Trainers) see only the details relevant to their work. For example, on the **Enrollment**, I included student, course, session, fees, and status fields in one section so it’s easier for staff to manage enrollments.



* Compact Layouts
* **Schema Builder**

I explored the **Schema Builder** to visualize the objects and their relationships. It gave me a bird’s-eye view of how **Course, Class Session, Enrollment, Progress, Lead, and Contact** are all connected. This helped me confirm if my relationships (like Lookup and Master-Detail) were correctly set up.

🡪This is how the schema looks,



* Lookup vs Master-Detail vs Hierarchical Relationships

In my project, I used both **Lookup** and **Master-Detail** relationships:

* **Lookup** → for flexible links, like *Enrollment → Contact (Student)* and *Class Session → Course*. These can exist independently.
* **Master-Detail** → for strict dependency, like *Progress → Enrollment*. If an enrollment is deleted, its progress records are deleted too.
* Junction Objects
* External Objects

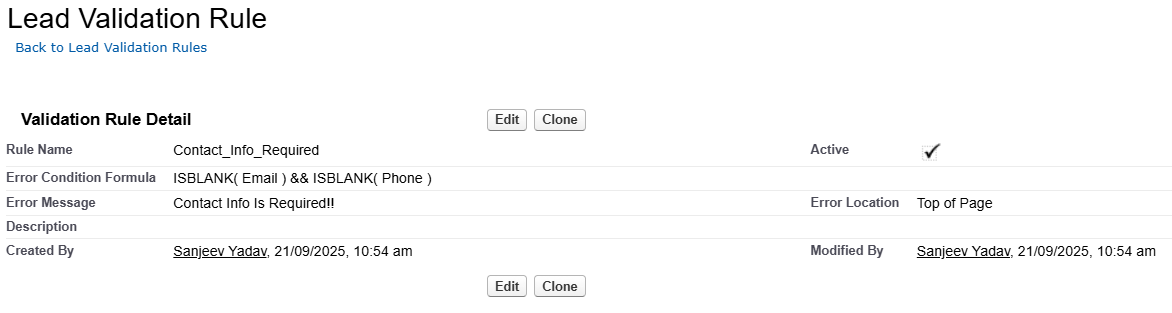
# Phase 4: Process Automation (Admin)

* **Validation Rules**

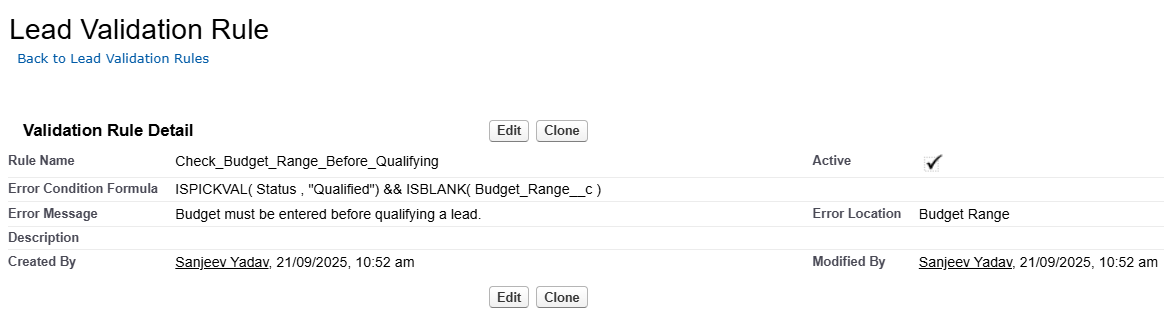
🡪 In my project, I added validation rules mainly to ensure that key fields are not left blank, dates are consistent, and numerical values (like fees and completion percentages) are within valid ranges. These rules help admission staff and trainers enter **clean and correct data**, which improves the **reliability** of the entire system.

**1**.**Validation Rules For Lead**:

**a**.**Contact Info Required**:This rule ensures that the contact info is entered.

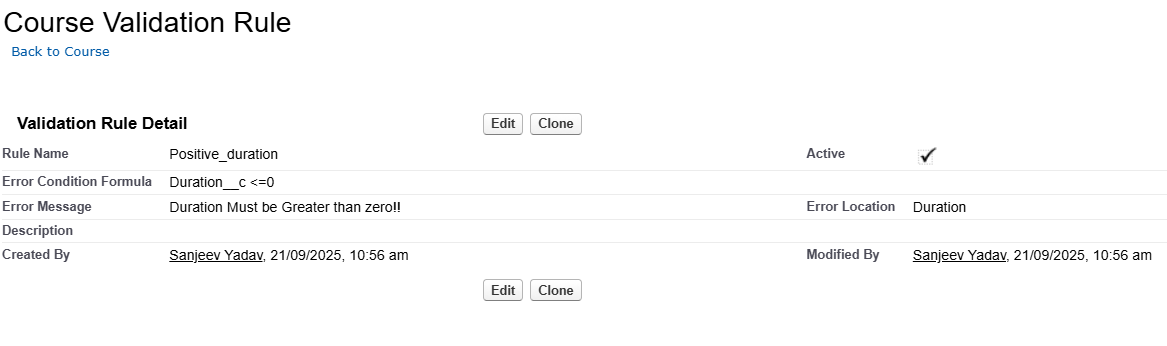


**b**.**Check budget before qualifying**:This rule ensures that budget range is entered before qualifying the lead.

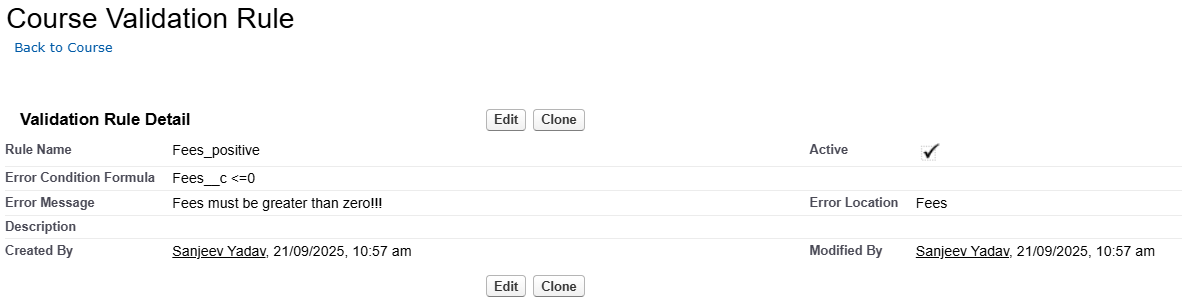


**2**.**Validation Rules for Course:**

**a. Duration must be Positive:** This rule ensures that the course duration entered is positive.

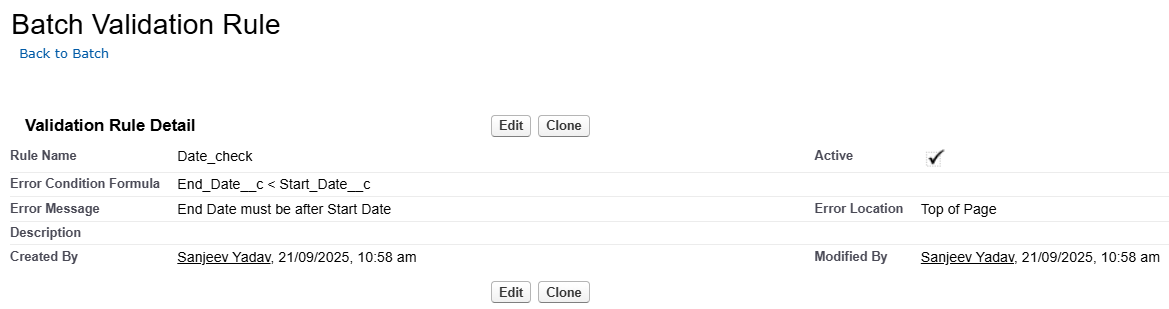


**b. Fees Must be positive:** This Rule ensures that the fees entered is positive.



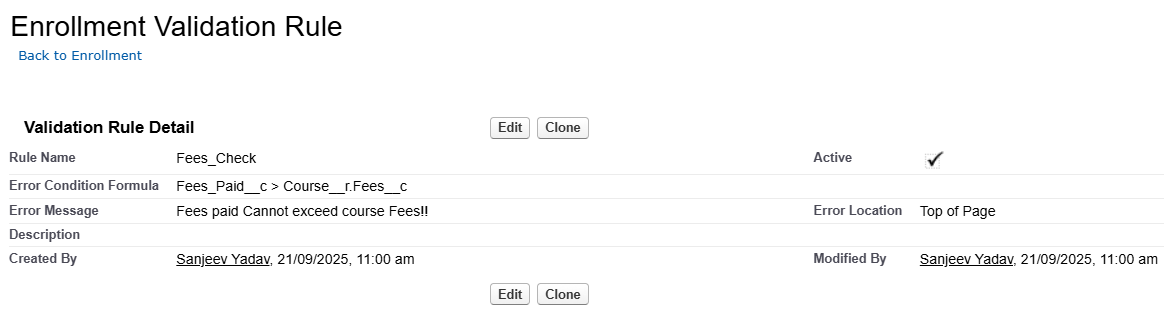
**3.Validation Rules For Batch:**

**a. Start Date Must be before end Date:** This rule ensures that the start date is entered before the end date.

****

**4.Validation Rules for Enrollment:**

**a. Fee paid :** This rule ensures that the fee paid is less than or equal to course fee.

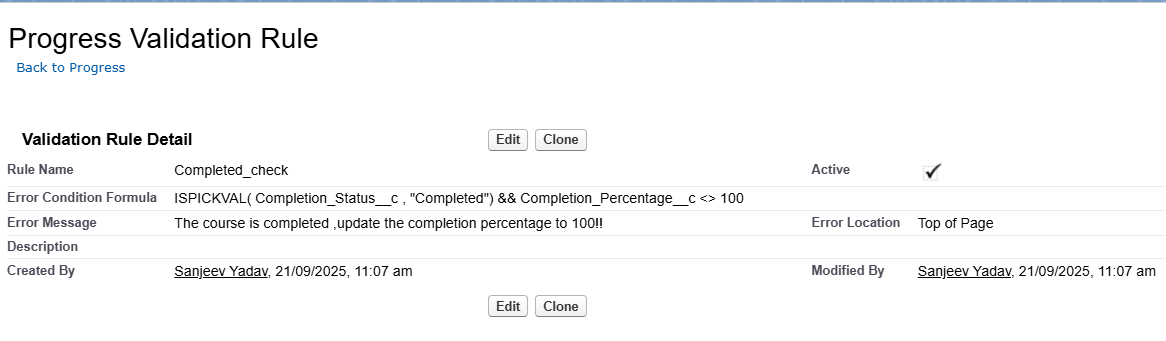
****

**b.Payment Status Check:** If payment status is paid ,fee must be greater than zero.

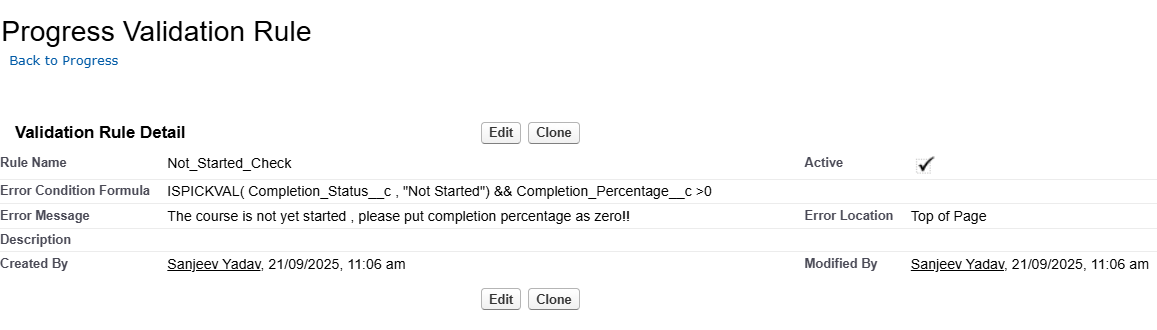


**5.Validation Rules for Progress:**

**a.Completion Status Check:** If course status is completed ,then course percentage must be 100.

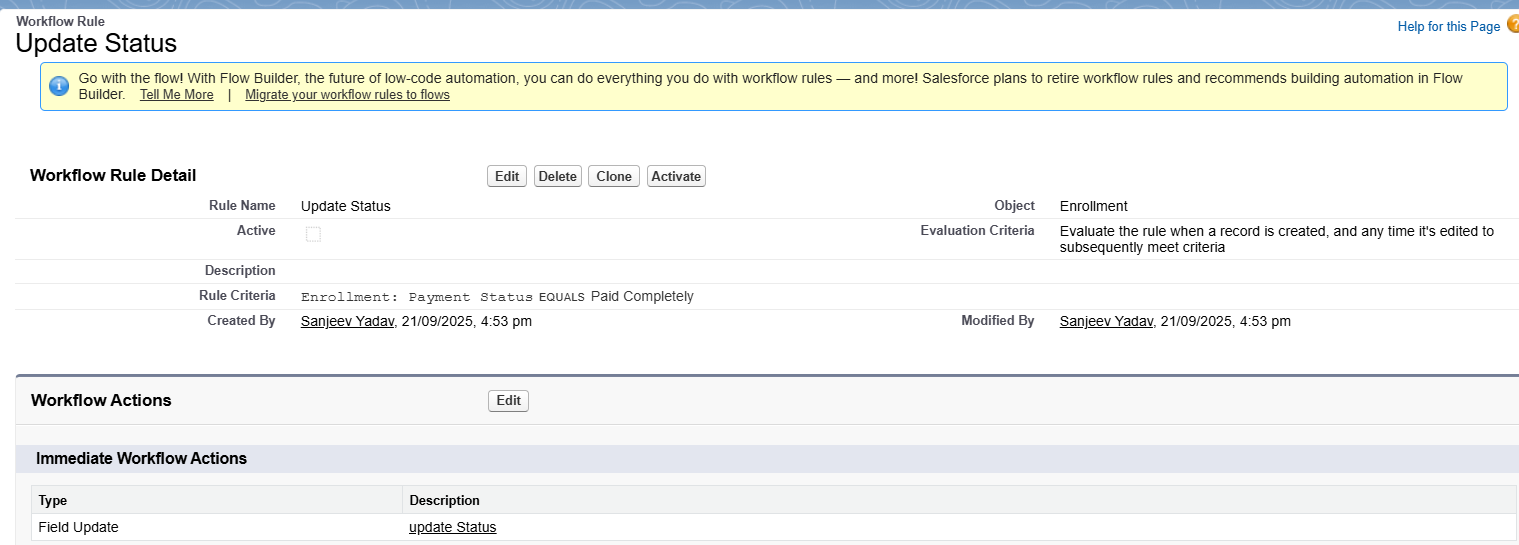


**b.Not Started Check:** If course status is not started, then course completion percentage be 0.

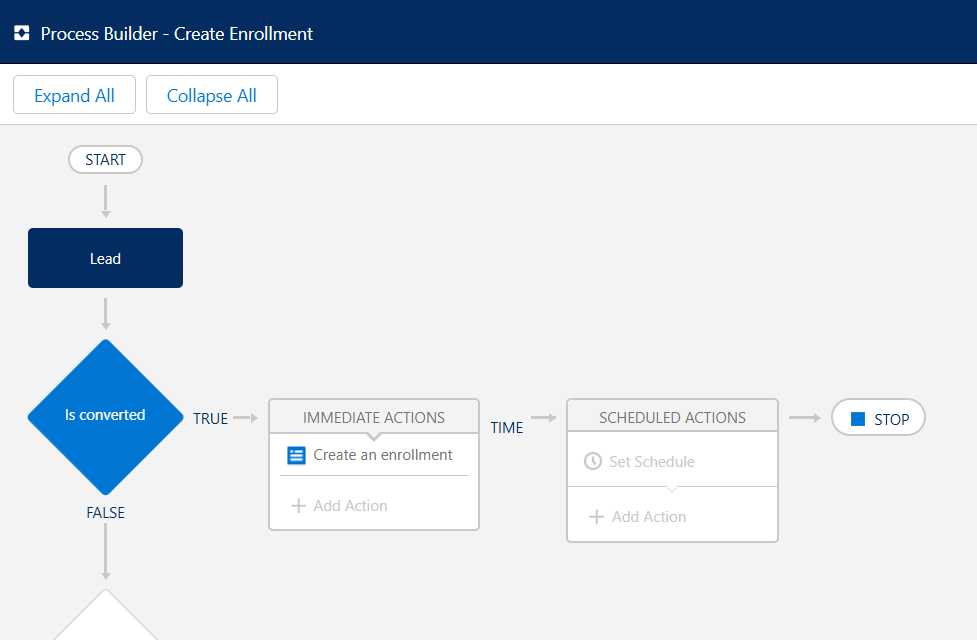


* **Workflow Rules**

🡪I created a workflow rule for Enrollment , if the payment status us paid completely then, update the Enrollment status to active .

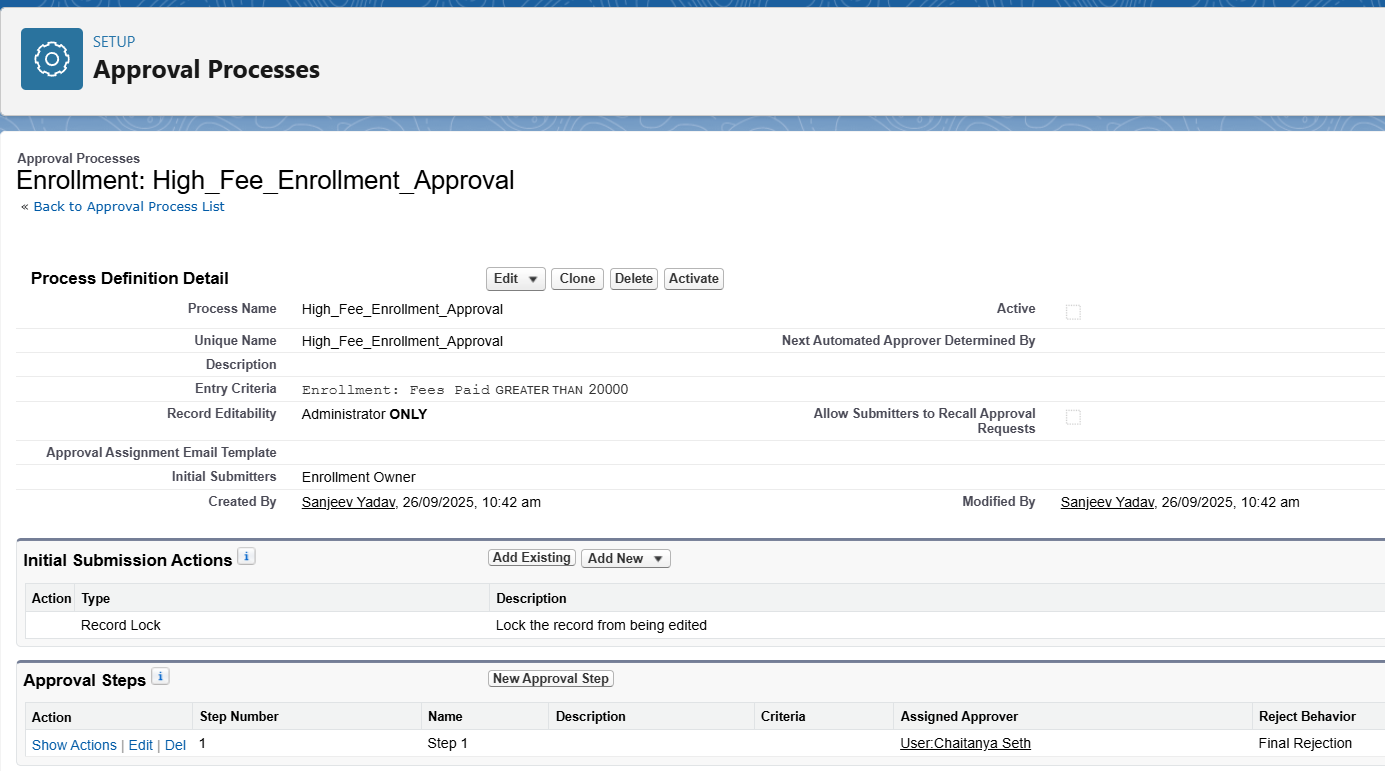
****

* **Process Builder**

🡪Added a process for creating an enrollment object automatically if a lead is converted.  


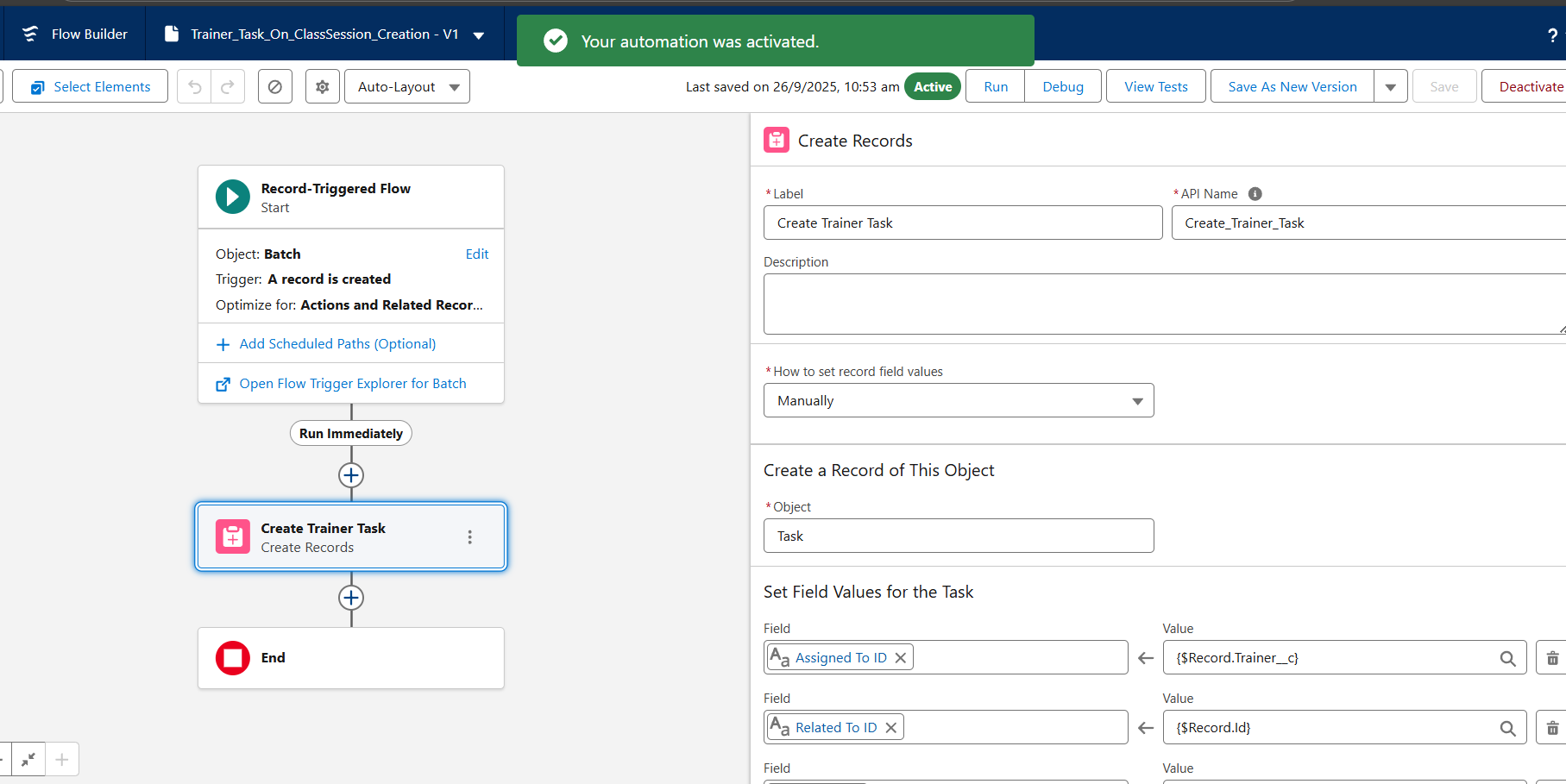
* **Approval Process**

🡪Added an approval process to owner to approve the enrollment ,if the fee paid by the student is more than 20,000.

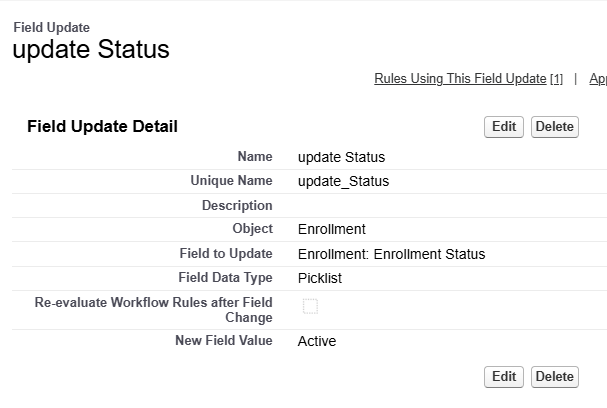


* **Flow Builder (Screen, Record-Triggered, Scheduled, Auto-launched)**

🡪Added a flow to create a task to the trainer to add a task to the trainer to create the session material ,If a new session is created and assigned to him.

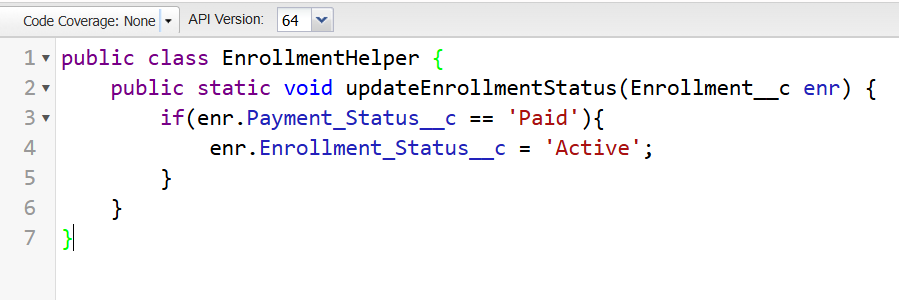


* **Field Updates**
* 🡪I added a workflow action for this field update of enrollment status to active .

****

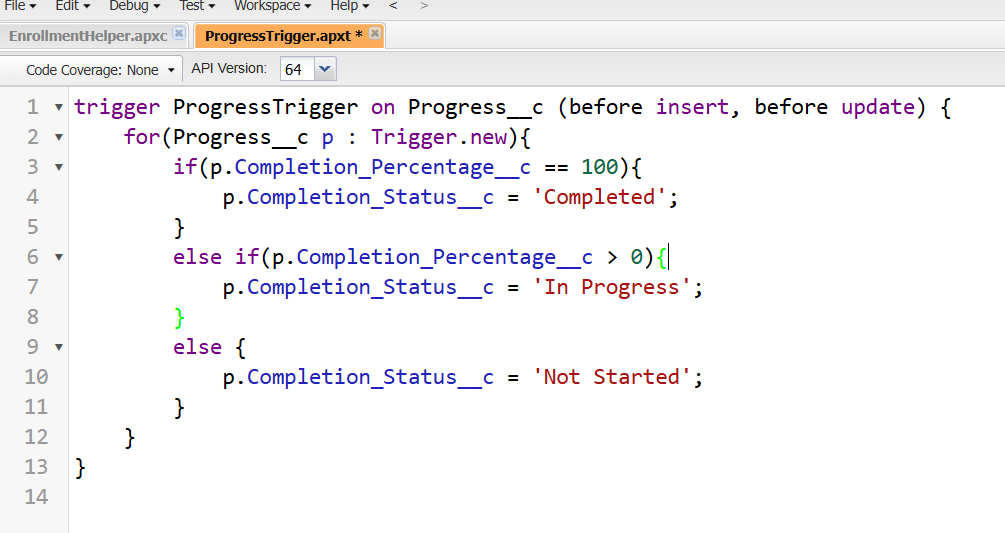
# Phase 5: Apex Programming (Developer)

* **Classes & Objects**

🡪I used a class to update the status of course to active if the payment status is paid completely,  


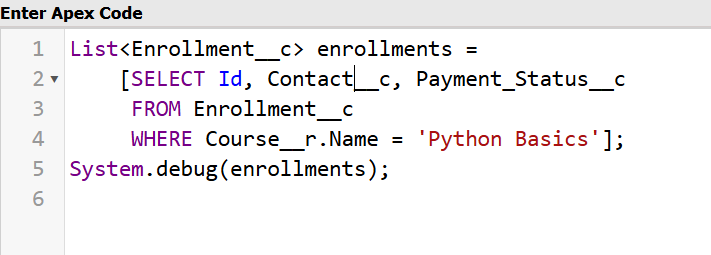
* **Apex Triggers (before/after insert/update/delete)**

🡪I used a trigger to update the calues of the completion status of progress based on completion percentage of the record,



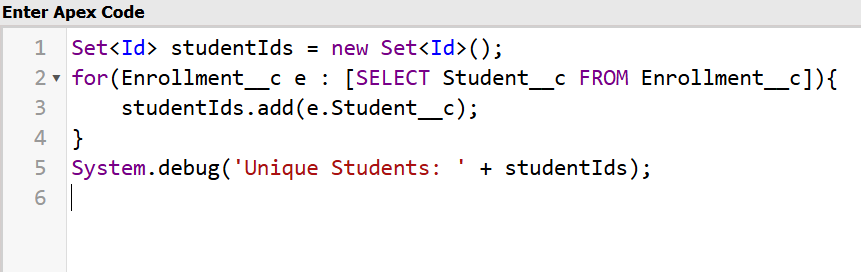
* **SOQL**

🡪We can use SOQL to fetch any records from the database based on our requirement, for example I have used an SOQL statement to retrieve all the enrollment details for the ‘Python Basics’ Course.



* **Collections: List, Set, Map**

**🡪**Collections can be used to store multiple values effective ,here is a sample usage to use a collection in apex,

****

* **Control Statements**

🡪If/Else, Loops are called as control statements ,we use them to check the certain conditions are met or not.

* **Batch Apex**

🡪 Batch Apex is used when you need to process **large volumes of records** that can’t be handled in a single transaction due to governor limits. It breaks the work into smaller chunks (batches) and processes them one by one. For example, cleaning up thousands of old enrollments or recalculating student progress in bulk.

* **Queueable Apex**

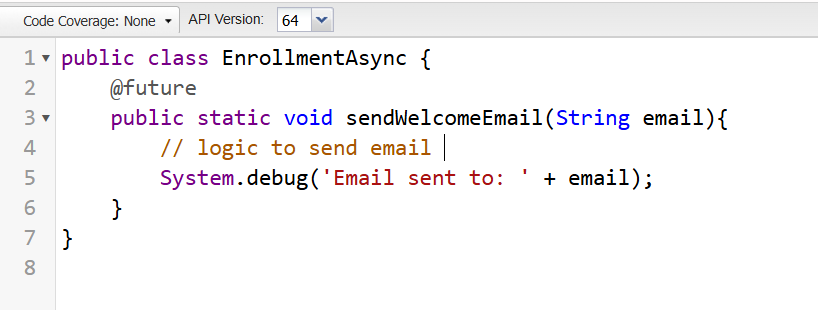
**🡪** Queueable Apex is used for background jobs that need more complex logic than future methods. You can chain multiple jobs together and pass parameters. For example, sending notifications to all students in a batch after enrollment is confirmed.

* **Scheduled Apex**

🡪 Scheduled Apex lets you run Apex code at **specific times or intervals**. For example, sending weekly reminders to trainers about their upcoming sessions every Monday morning.

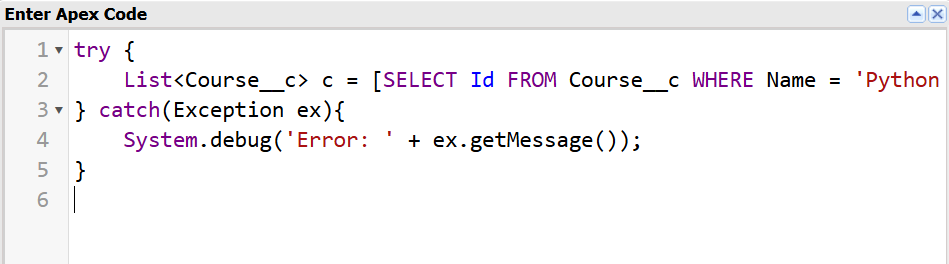
* **Future Methods**

🡪I used a sample future method that tries to send a mail asynchronously when a student is enrolled.



* **Exception Handling**

🡪Exceptions are handled using the try& catch blocks,



* **Test Classes**

🡪 Salesforce requires all Apex code to be covered by **unit tests** before deployment. Test Classes check that your triggers, classes, and flows work correctly and don’t break when data changes. For example, testing that enrollment status updates to *Active* when payment is marked as *Paid*.

* **Asynchronous Processing**

🡪 Asynchronous processing means running tasks **in the background without blocking the user**. Salesforce provides Future Methods, Batch Apex, Queueable Apex, and Scheduled Apex for this. It’s useful for long-running jobs like sending emails, making external calls, or processing thousands of records while the user continues working.