InternLink Hub -“A Central Platform for Internships & Placements”

📄 Phase 1: Problem Understanding & Industry Analysis

---

1. Requirement Gathering

The requirements for InternLink Hub were collected by analyzing the needs of students, placement officers, and recruiters. The main requirements identified are:

- **Students** should be able to create profiles, upload resumes, and apply for internships.
- **Recruiters** should be able to post internships with details such as title, duration, stipend, and number of openings.
- **Placement Officers** should have the ability to monitor student applications, validate resumes, and track placement records.
- The system should maintain **applications** with statuses like *Applied, Under Review, Selected, Rejected*.
- Reports and dashboards should provide insights into applications, selections, and placement trends.

---

2. Stakeholder Analysis

The project involves three primary stakeholder groups:

- **Students** → Need a simple platform to find and apply for opportunities, and track status.
- **Recruiters** → Need to post internships and review applicants efficiently.
- **Placement Officers** → Need to manage both students and recruiters, and oversee the end-to-end placement process.
- **Administrators** → Manage system access, permissions, and configuration.

---

3. Business Process Mapping

The end-to-end workflow of InternLink is as follows:

1. Recruiters post internship opportunities.
2. Placement Officers verify postings and publish them.
3. Students search and apply for internships.
4. Applications are reviewed by Placement Officers and Recruiters.
5. The status of each application is updated (Applied → Under Review → Selected/Rejected).

6. Final placements are recorded with details like joining date and package.

---

4. Industry-Specific Use Case Analysis

Internship management is a critical need in the **education sector**. Many colleges still rely on manual methods (spreadsheets, emails), leading to inefficiencies.
Use Case Example:

- A student named *Ravi* applies for an internship at *Infosys*.
- The Placement Officer reviews and forwards Ravi's application.
- The recruiter evaluates and marks Ravi as *Selected*.
- The system updates the status and records the final placement.

---

5. AppExchange Exploration

A review of existing solutions on Salesforce AppExchange (like **TargetRecruit** and **JobScience**) showed that while these apps support recruitment, they are designed mainly for corporate hiring, not academic internships.
Hence, a **custom Salesforce solution** (InternLink) is required to meet the specific needs of students, recruiters, and placement officers in educational institutions.

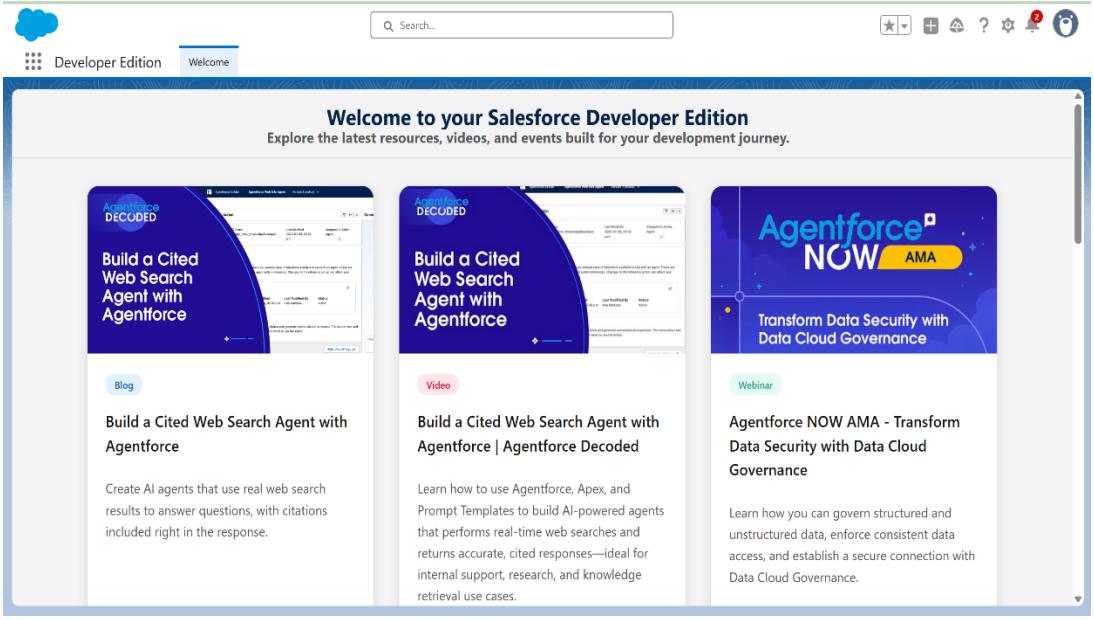
InternLink Hub -“A Central Platform for Internships & Placements”

Phase 2: Org Setup & Configuration (InternLink)

---

1. Salesforce Editions

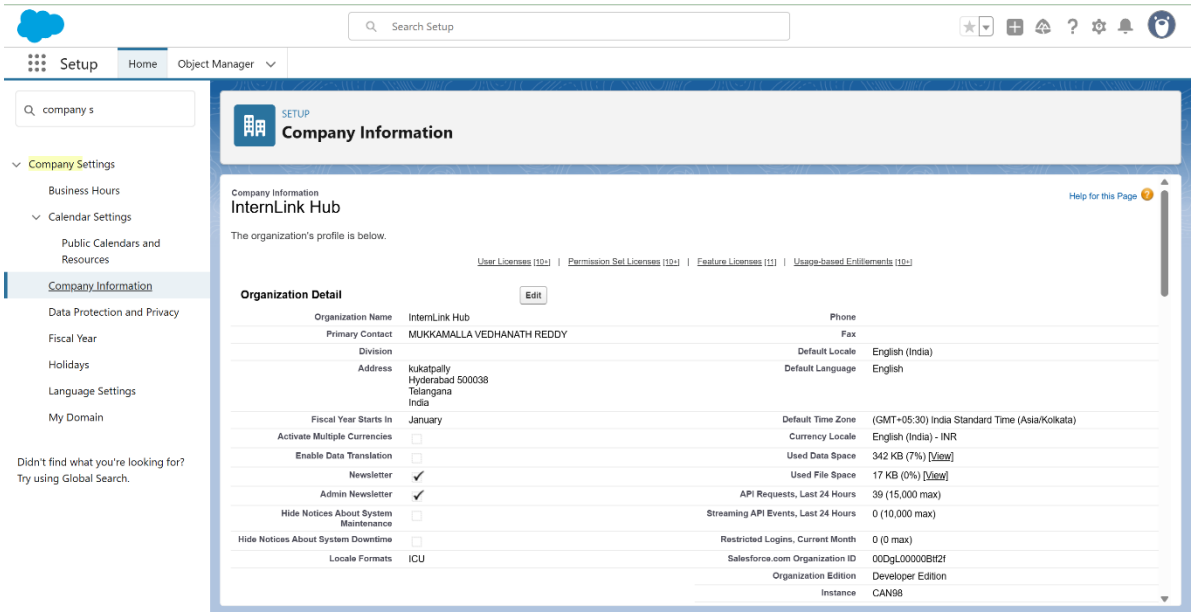
- Used **Salesforce Developer Edition (Free)** from developer.salesforce.com.
- Provides core features needed for the InternLink project (custom objects, automation, reports, etc.).



---

2. Company Profile Setup

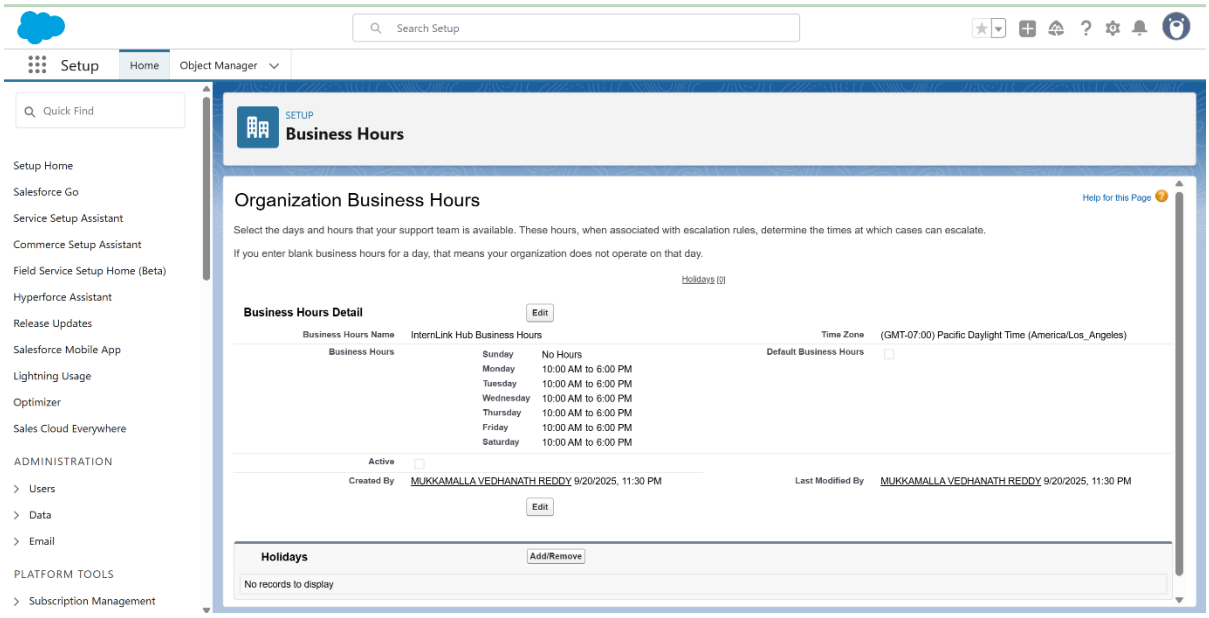
- Setup → **Company Settings → Company Information**
- Updated details:
    - Organization Name: *InternLink CRM*
    - Default Locale: English (India)
    - Currency: INR (₹)
    - Time Zone: IST



---

3. Business Hours & Holidays

- Defined **College Business Hours** → Mon–Fri, 10 AM – 6 PM
- Created holiday records (e.g., Republic Day, Independence Day)
- Linked holidays to the business hours schedule

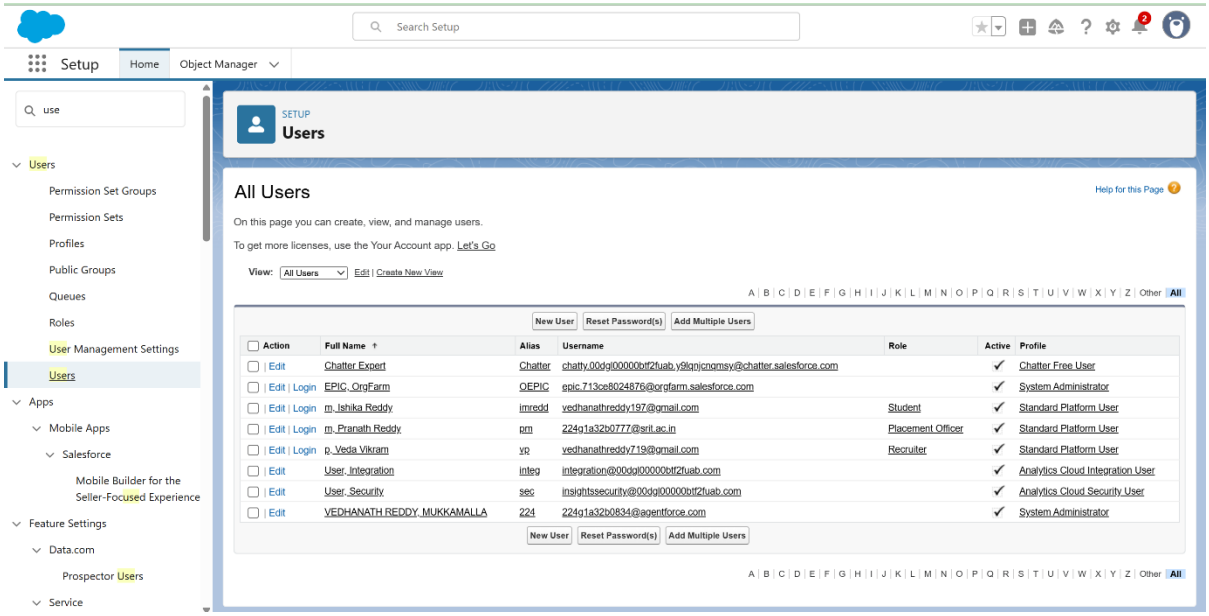


---

4. User Setup & Licenses

- Created three test users to represent project roles:
    1. **Placement Officer** → Full Salesforce license
    2. **Recruiter (Company HR)** → Salesforce license

3. **Student** → Salesforce license (later can be converted to Experience Cloud license)



5. Profiles

- Cloned **Standard User Profile** to create role-specific profiles:
  - **Placement Officer Profile** → Full access (CRED) to all objects
  - **Recruiter Profile** → Manage Internships + Applications, limited access to Placements
  - **Student Profile** → Limited to creating/viewing own Applications and reading Internships

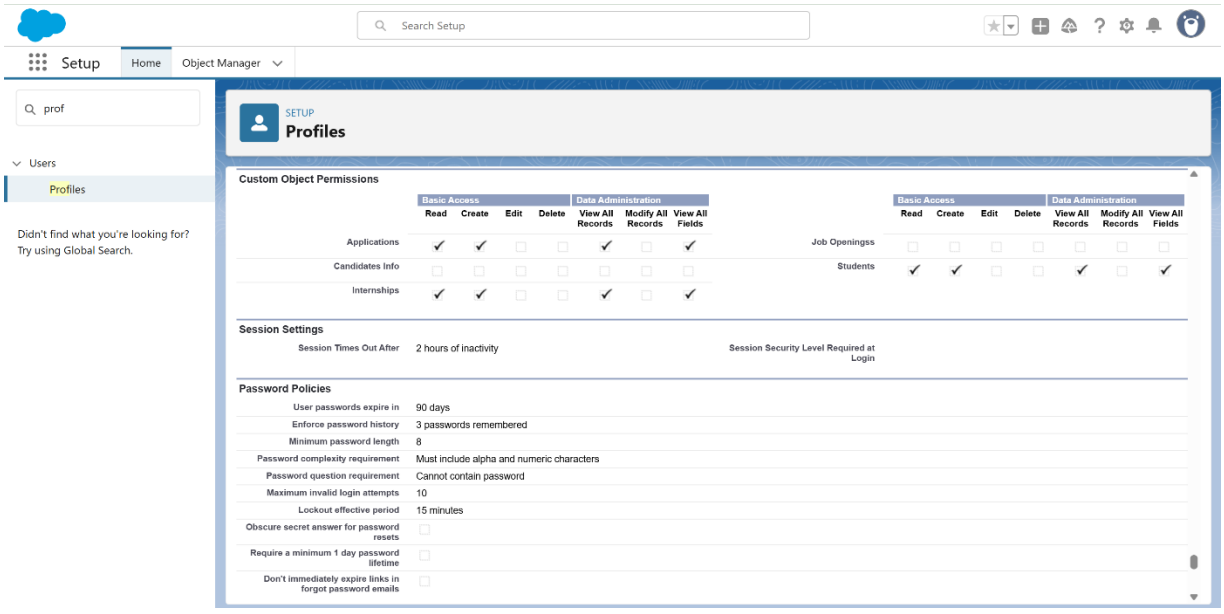


Fig:-5.1 Placement Officer Profile(showing Internship, Applications, Students permissions)

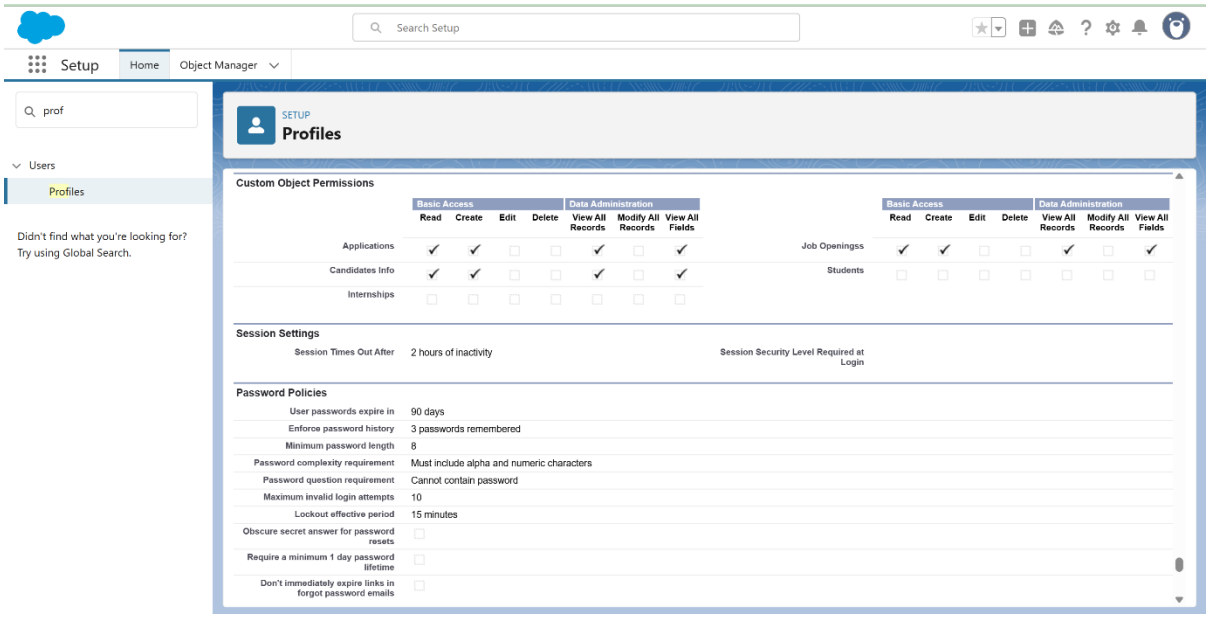


Fig:-5.2 Recruiter Profile(showing Job Openings, Applications, Candidate Info permissions)

---

6. Roles

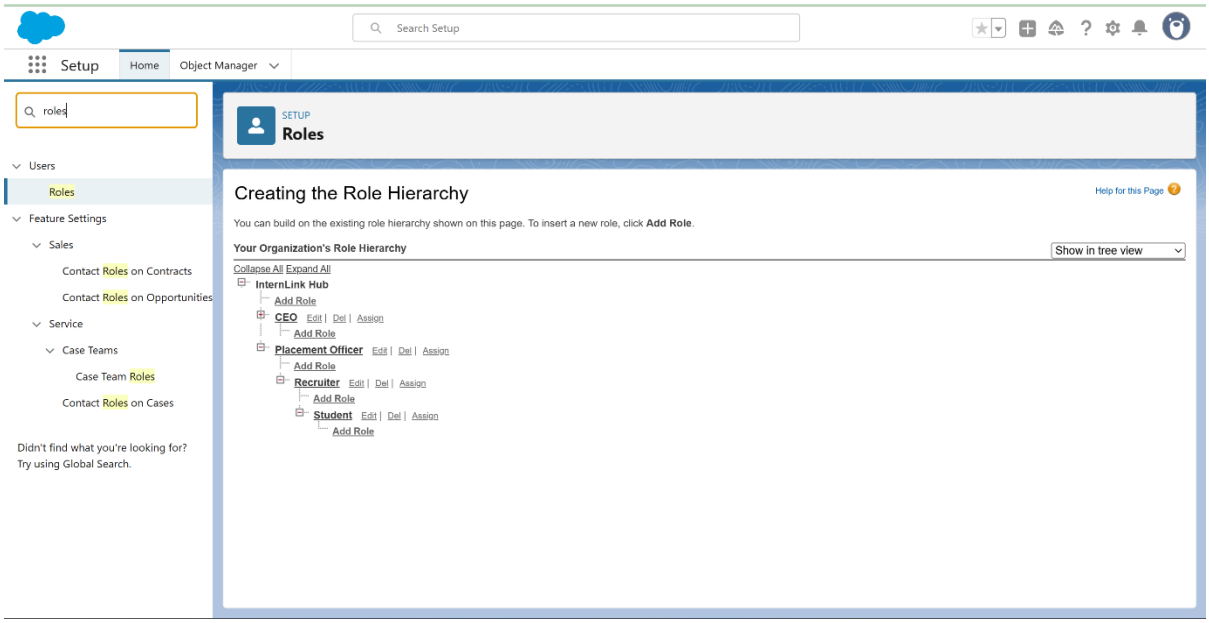
- Configured hierarchy for role-based visibility:

Placement Officer

├— Recruiter

└— Student

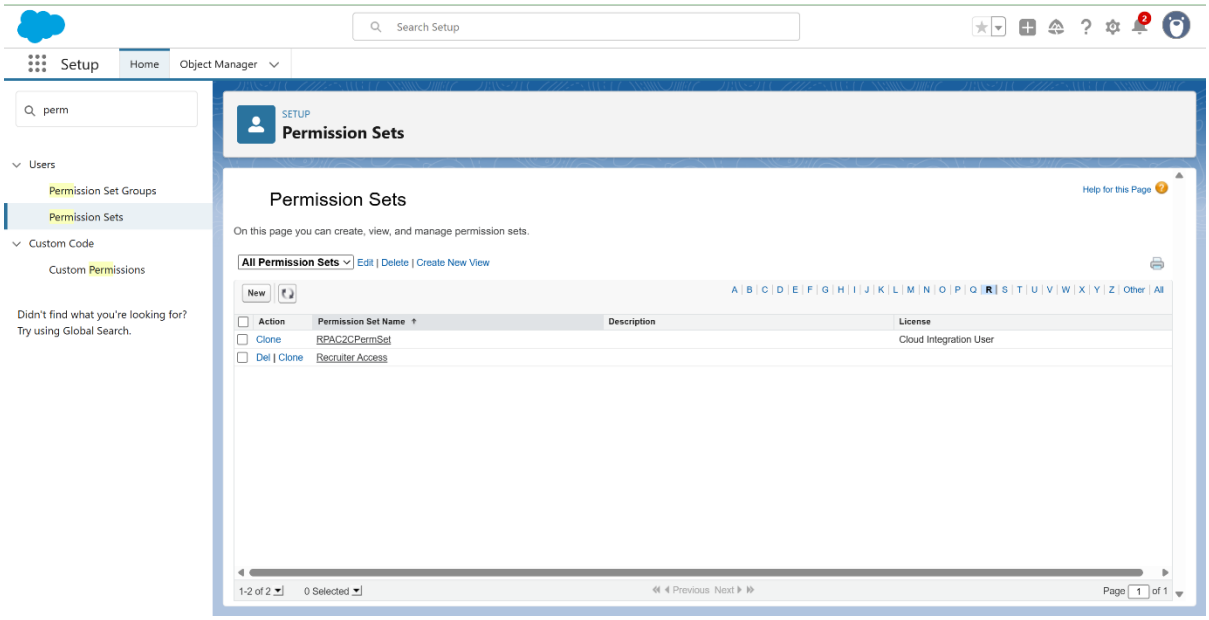
- Ensures proper upward visibility and record-level security



---

7. Permission Sets

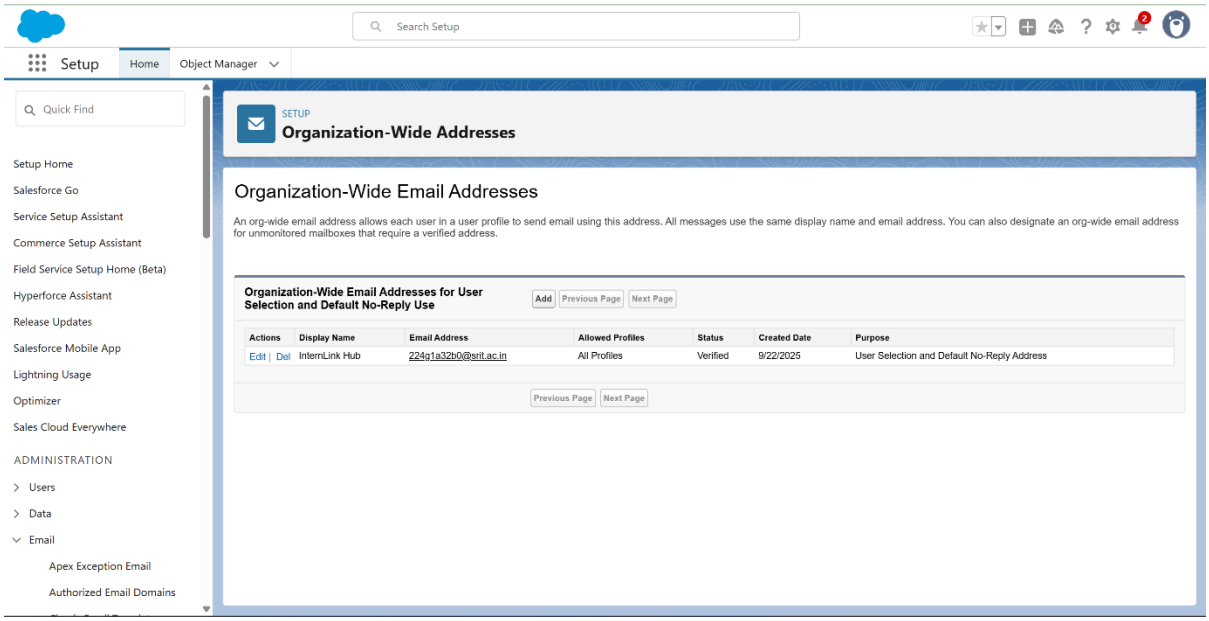
- Created **Recruiter Access** permission set
- Granted additional access to Company and Internship objects
- Assigned to Recruiter users for flexibility



---

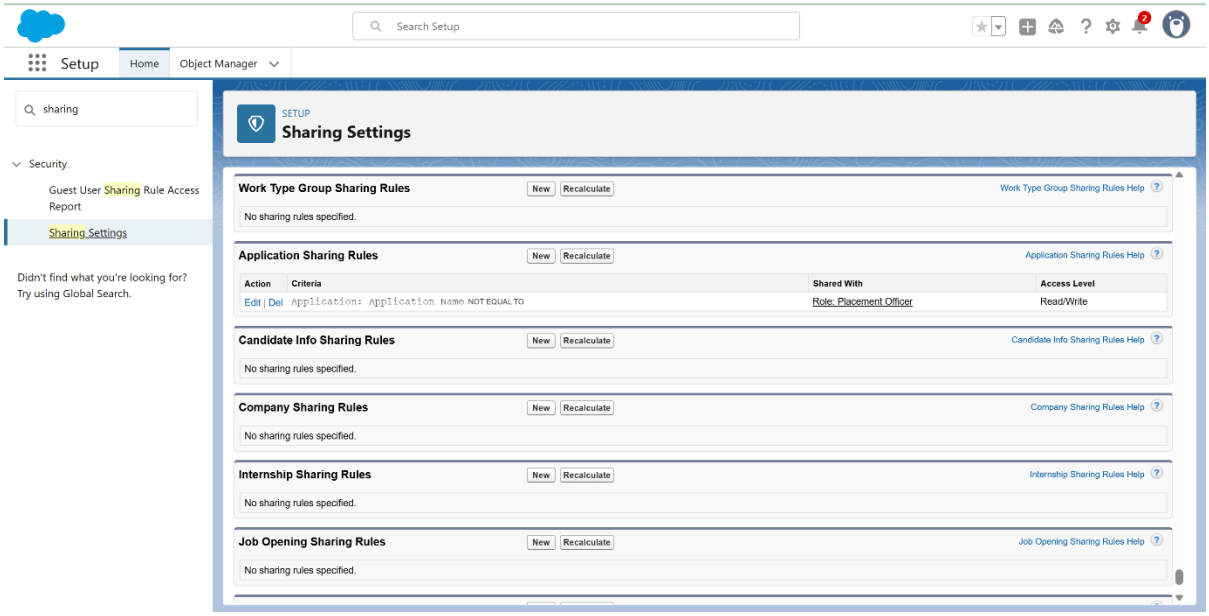
8. Org-Wide Defaults (OWD)

- Setup → **Sharing Settings**
- Security rules applied:
    - Student__c = Public Read Only
    - Company__c = Public Read Only
    - Internship__c = Public Read Only
    - Application__c = Private
    - Placement__c = Private



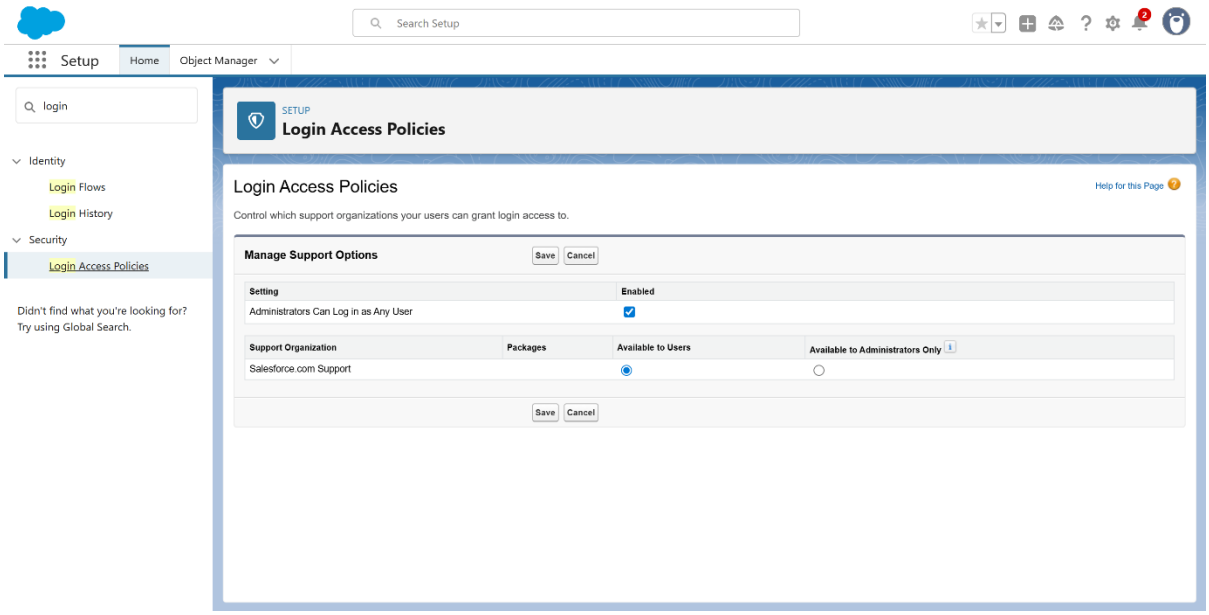
9. Sharing Rules

- Created a **Public Group** → Placement Officers
- Sharing Rule on **Application__c**:
  - Criteria: All records
  - Access: Read/Write
  - Shared with: Placement Officer Group



10. Login Access Policies

- Setup → **Login Access Policies**
- Enabled: *Administrators Can Log in as Any User*



---

11. Dev Org Setup

- Enabled **Lightning Experience** for modern UI
- Added custom tabs for: Student, Company, Internship, Application, Placement
- Created **InternLink CRM App** for centralized navigation

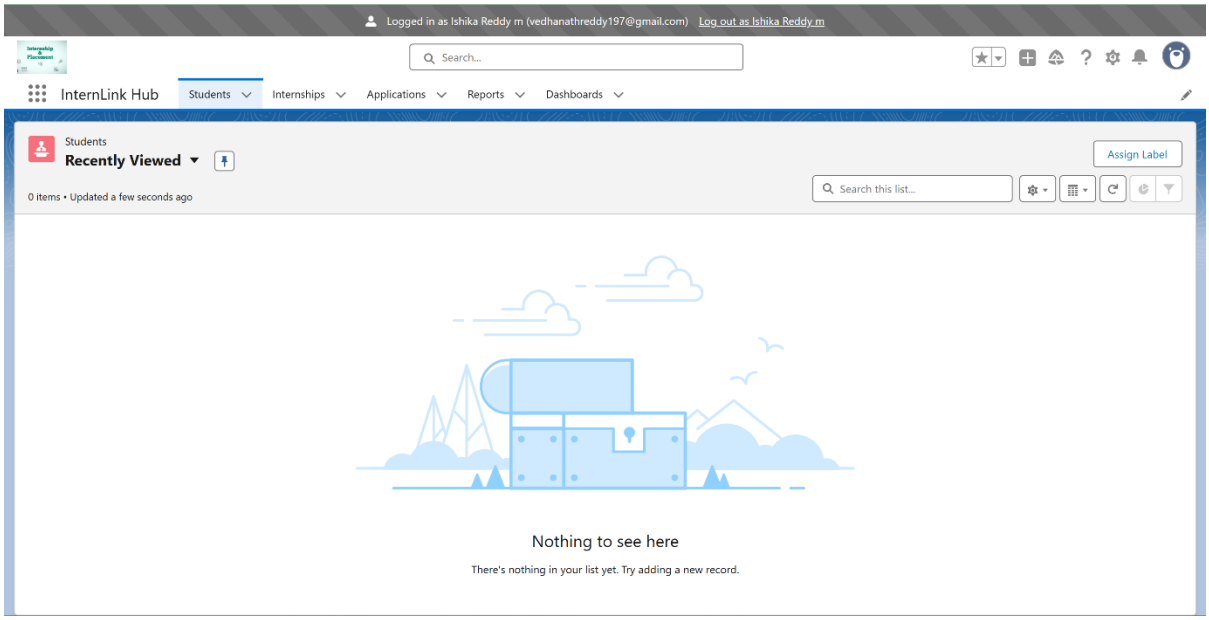


Fig:-11.1 Student View(can only access student,internship,application)

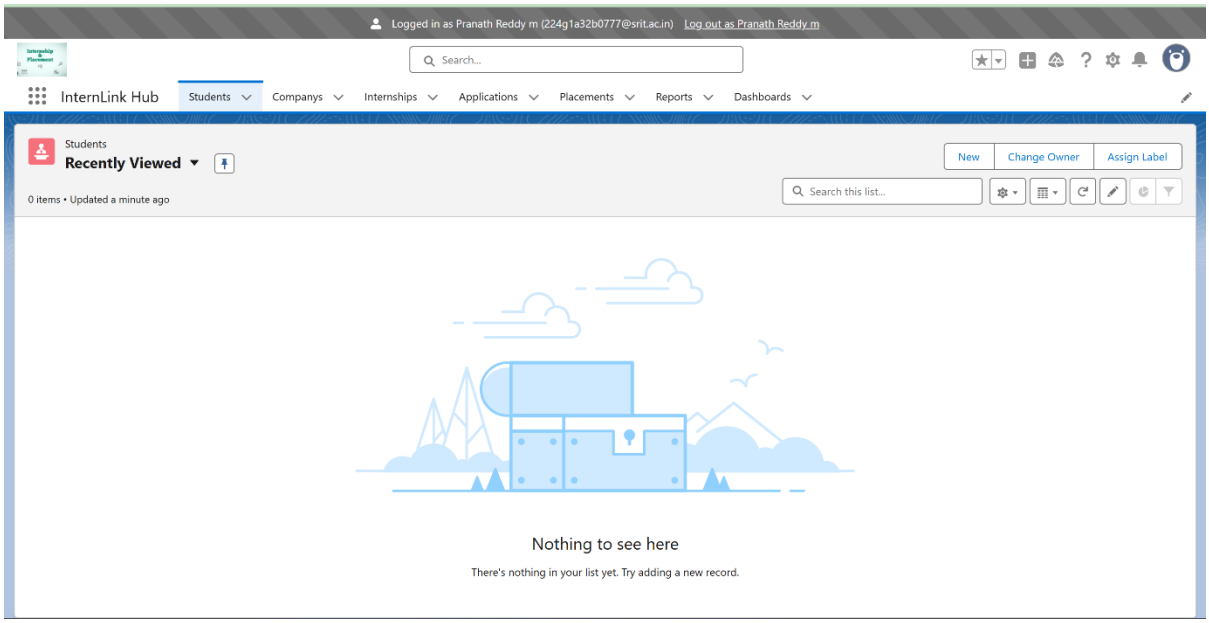


Fig:-11.2 Placement officer View(can access all the students info,companys,intership,application,placements)

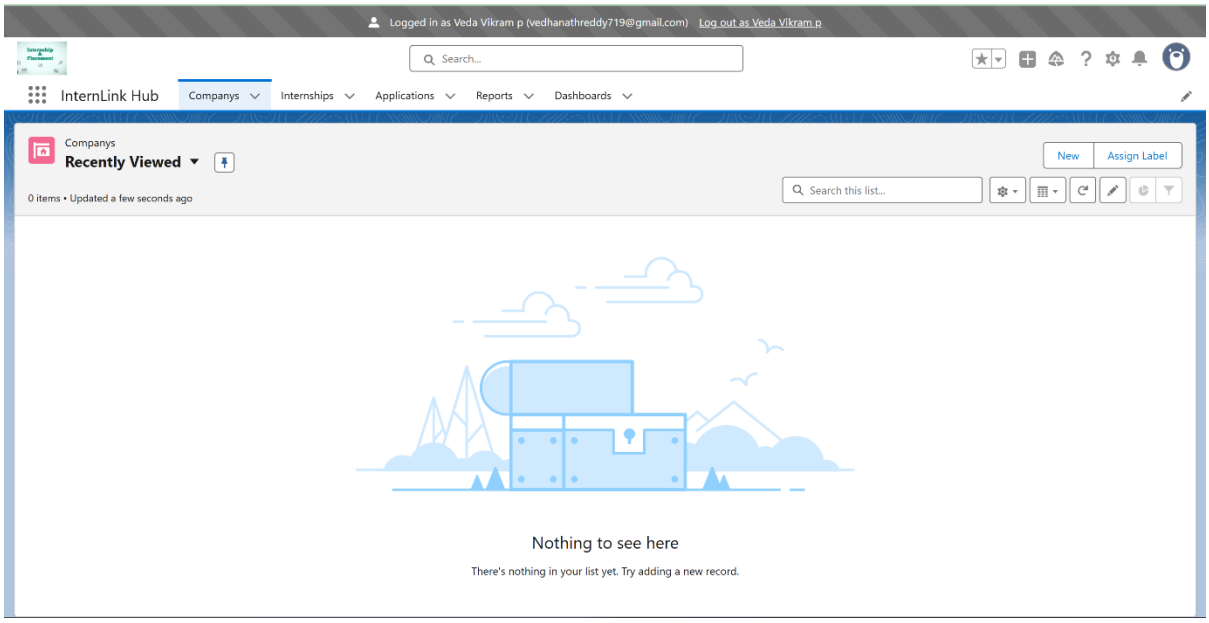


Fig:- 11.3 Recuritr view(can only access company,Internship,Application)

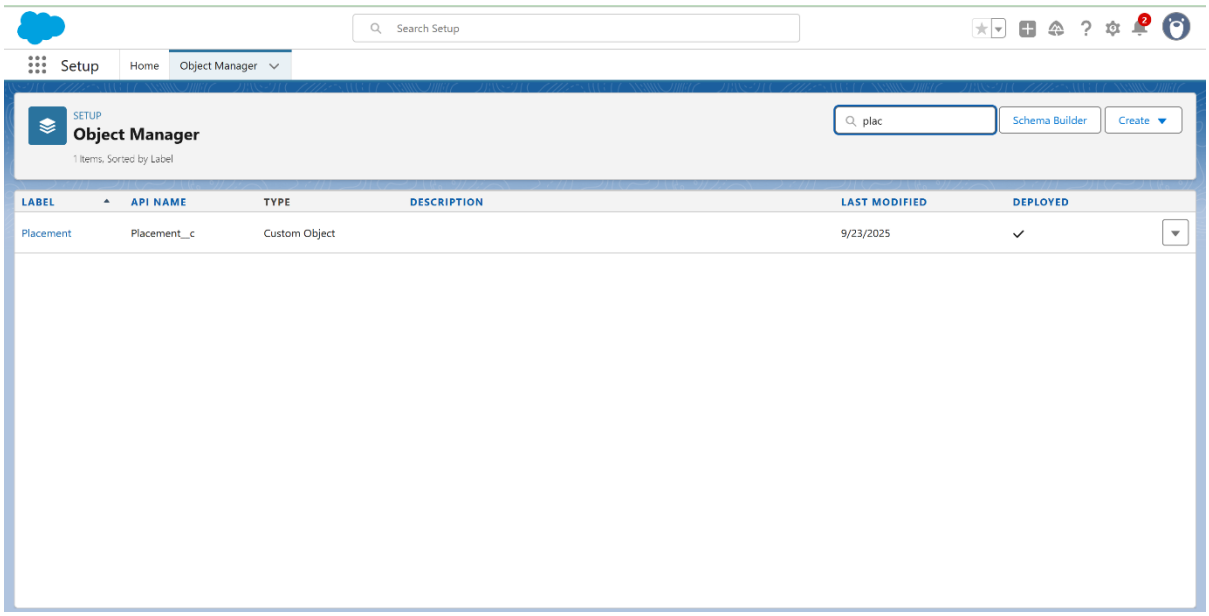
InternLink Hub -"A Central Platform for Internships & Placements"

📘 Phase 3: Data Modeling & Relationships

---

1. Standard & Custom Objects

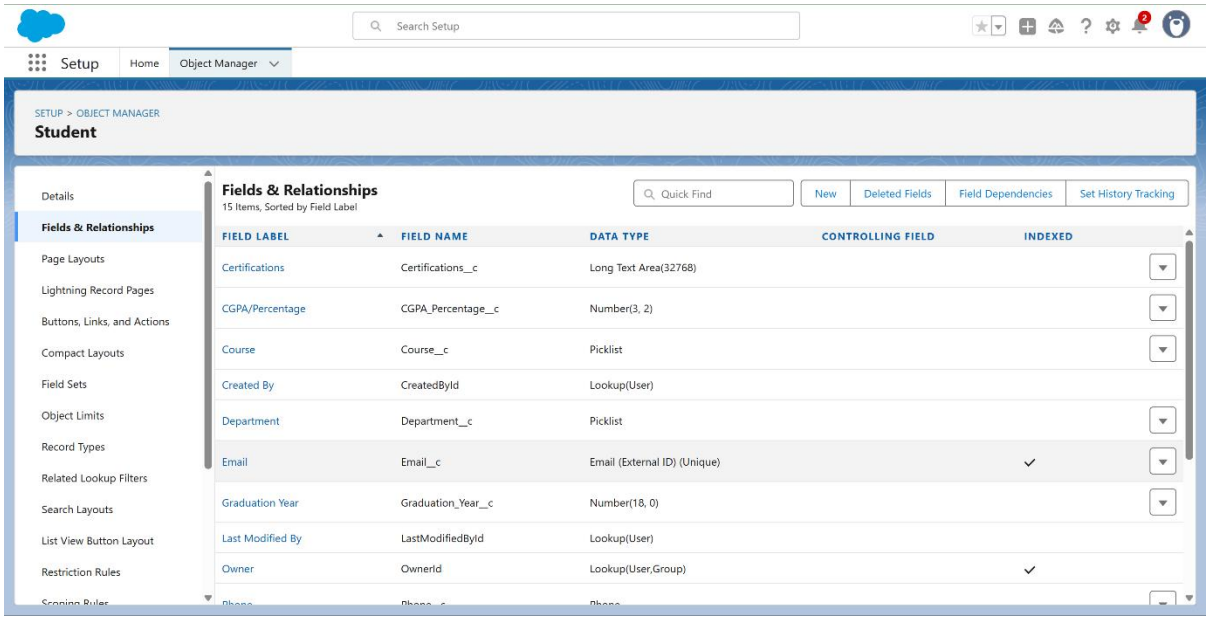
- **Standard Objects**:
  - *Account* → Stores company information.
  - *Contact* → Stores recruiter details.
- **Custom Objects**:
  - *Student__c* → Stores student details.
  - *Internship__c* → Stores internship postings and related info.
  - *Application__c* → Tracks applications submitted by students.
  - *Placement__c* → Stores final placement outcomes.



---

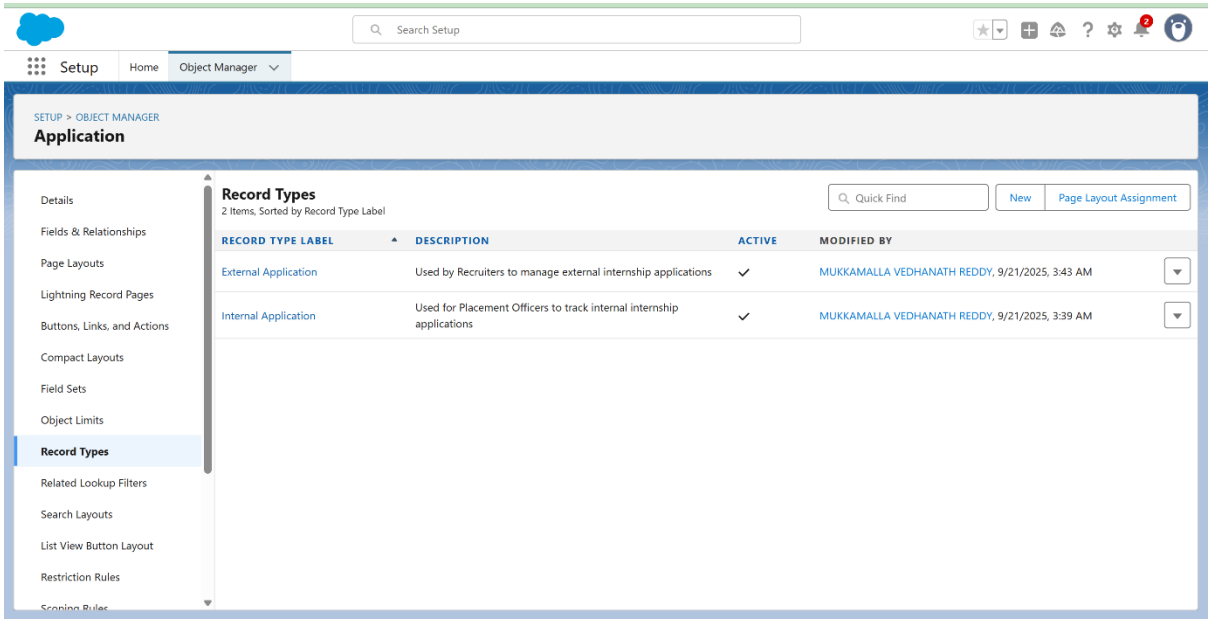
2. Fields

- Each object contains specific fields to store relevant information.
- Standard fields capture basic details such as Name, Email, and Phone.
- Custom fields are designed to store project-specific data like Course, Graduation Year, Stipend, Resume, and Application Status.
- Proper field design ensures accurate data entry and consistency.
- Well-structured fields also support effective reporting and automation processes.



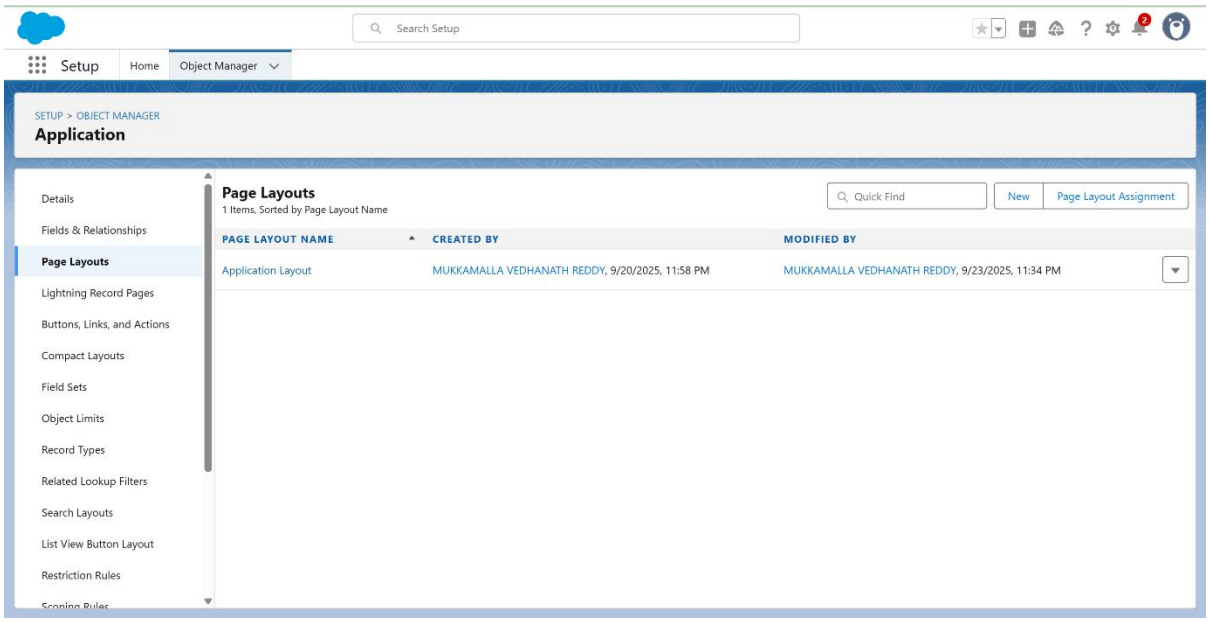
3. Record Types

- Record Types allow differentiation of workflows within the same object.
- In InternLink, they separate **internal and external applications**.
- They also help distinguish between **technical and non-technical internships**.
- Record Types enable customization of **page layouts, picklist values, and processes**.
- They ensure different user roles (e.g., Placement Officer vs Recruiter) see data relevant to them.



4. Page Layouts

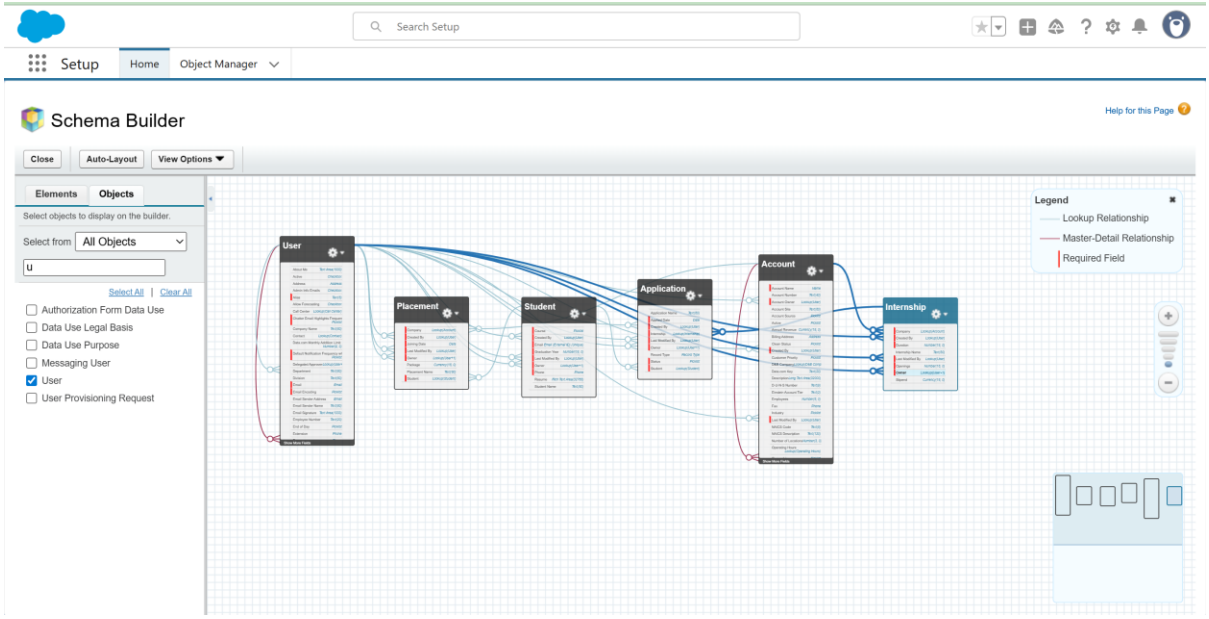
- Page Layouts define the **structure and order** of fields, sections, and related lists.
- In InternLink, each profile has its **own dedicated layout** (Student, Recruiter, Placement Officer).
- **Students** see academic details and resume information.
- **Recruiters** see internship-related fields (stipend, duration, status, etc.).
- **Placement Officers** see placement and application-related fields.



---

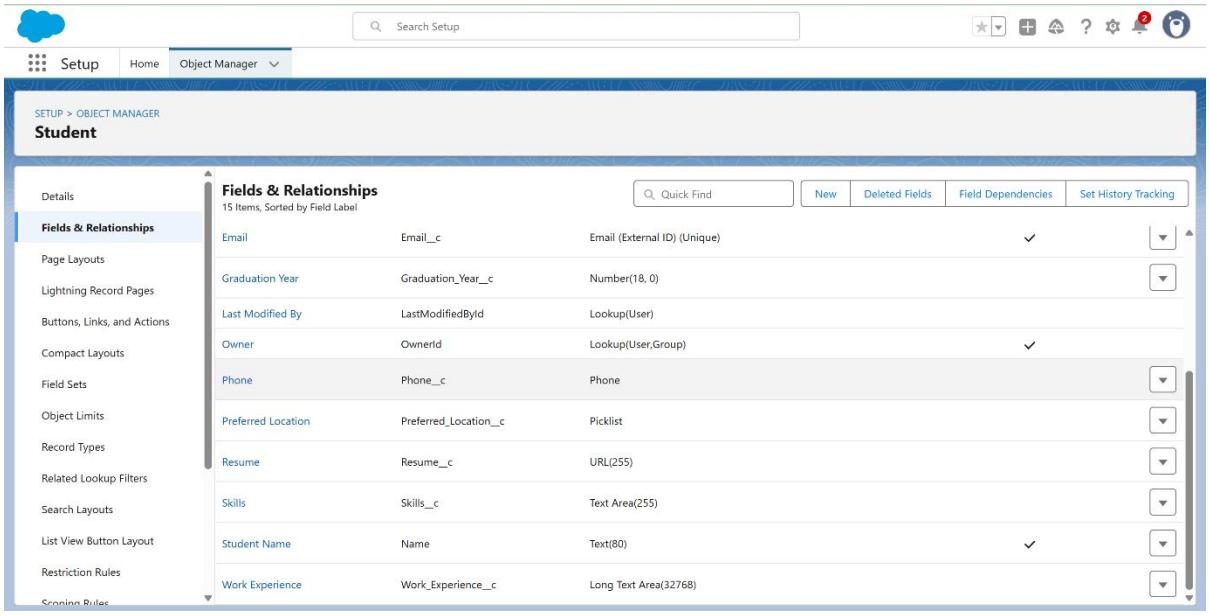
5. Schema Builder

- Schema Builder provides a **visual representation** of the data model.
- It shows **objects and their relationships** clearly.
- In InternLink, it helps admins and developers understand how **Student, Company, Internship, Application, and Placement** objects connect.
- Supports **quick creation of new fields** through drag-and-drop.
- Allows easy **validation of relationships** within the system.



6.**Lookup Relationship**

- Creates a link between two objects.
- Allows one object to reference another object's record.
- Can be **optional** or **required**.
- Does **not** affect ownership or security of the linked record.
- Example: Application__c → Student__c to link an application to a student.



InternLink Hub -“A Central Platform for Internships & Placements”

Phase 4: Process Automation (Admin) — InternLink Hub

1.Validation Rule: Phone Number Format

For this project, I have created a validation rule for the Phone field to ensure proper data quality. This rule checks that the phone number entered contains exactly 10 numeric digits and does not include any letters, special characters, or spaces. This ensures consistency and accuracy of contact information stored in the system, making it reliable for communication and operational purposes.

Validation Rule Name: phoneno

Object: Student

Field: Phone

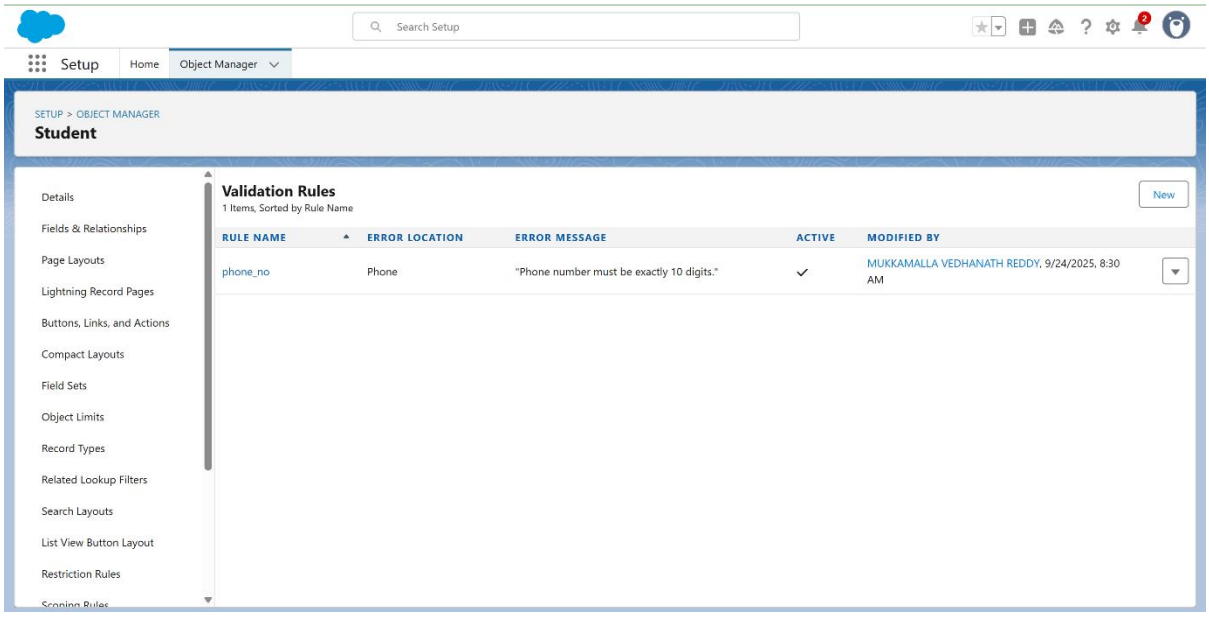
Purpose: Ensure that the phone number entered is exactly 10 digits and contains only numeric values.

Logic: Checks that the phone field does not contain letters, special characters, or spaces, and enforces a strict 10-digit format.

Benefit: Maintains data accuracy and consistency for reliable communication and reporting.

Description: Ensures proper data quality by validating that the phone number contains exactly 10 numeric digits without letters, special characters, or spaces. This provides consistent and accurate

contact information for communication and operational use.



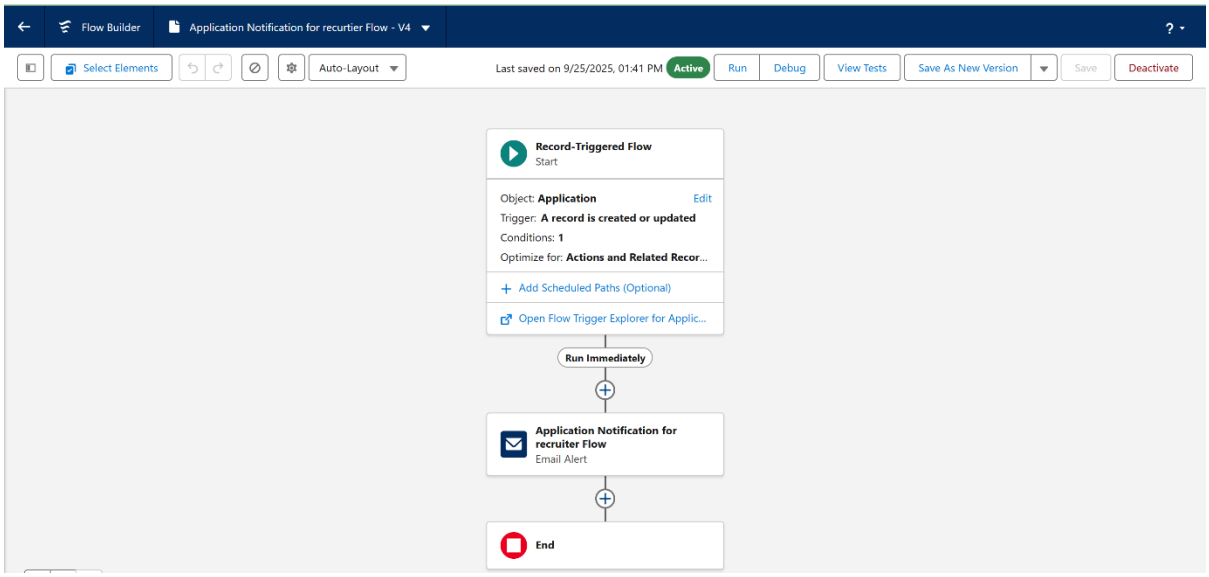
---

2. Flow Builder: Application Notification for Recruiter

For this project, I have created a flow named **Application_Notification_for_Recruiter** that

automatically sends an email to the recruiter when a student applies for an internship. This flow is triggered when a new Application record is created and the status is set to "Applied". The email contains important details such as the student's name, internship name, and application status so that the recruiter can quickly review and take action.

- **Flow Name:** Application_Notification_for_Recruiter
- **Object:** Application
- **Trigger:** Record-Triggered Flow (on Create of Application record when Status = "Applied")
- **Purpose:** To automate email notifications for recruiters when a student applies for an internship, enabling faster review and action.
- **Logic:** Triggered when a new Application record is created with status "Applied", then automatically sends an email to the recruiter containing student name, internship name, and application status.
- **Benefit:** Improves recruiter efficiency, reduces manual follow-ups, and ensures timely processing of internship applications.
- **Description:** This flow improves process automation by sending instant email notifications to recruiters whenever a new application is submitted with the "Applied" status. This ensures quick communication and streamlines the recruitment process.
- **Outcome:** Recruiters receive immediate alerts for new applications, allowing them to respond faster.



- Fig:- 2.1 **:** Application Notification for Recruiter flow

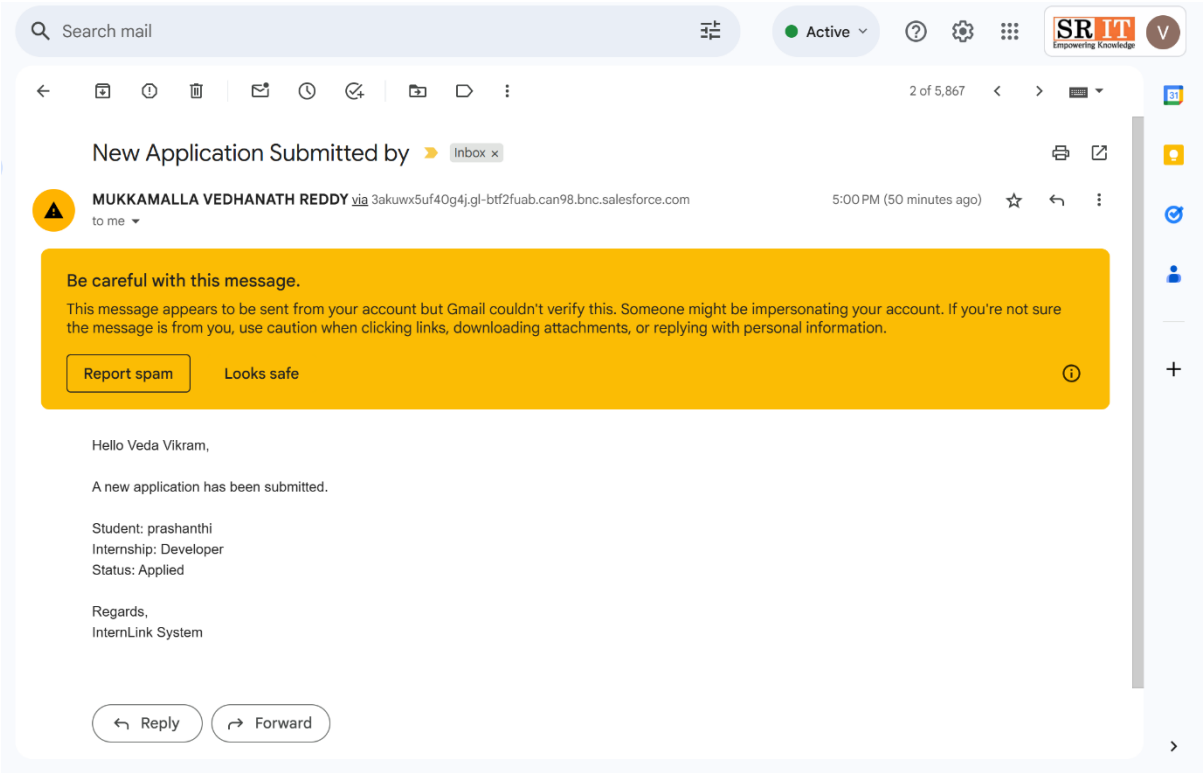


Fig:- 2.2 **:** Application Notification for Recruiter mail

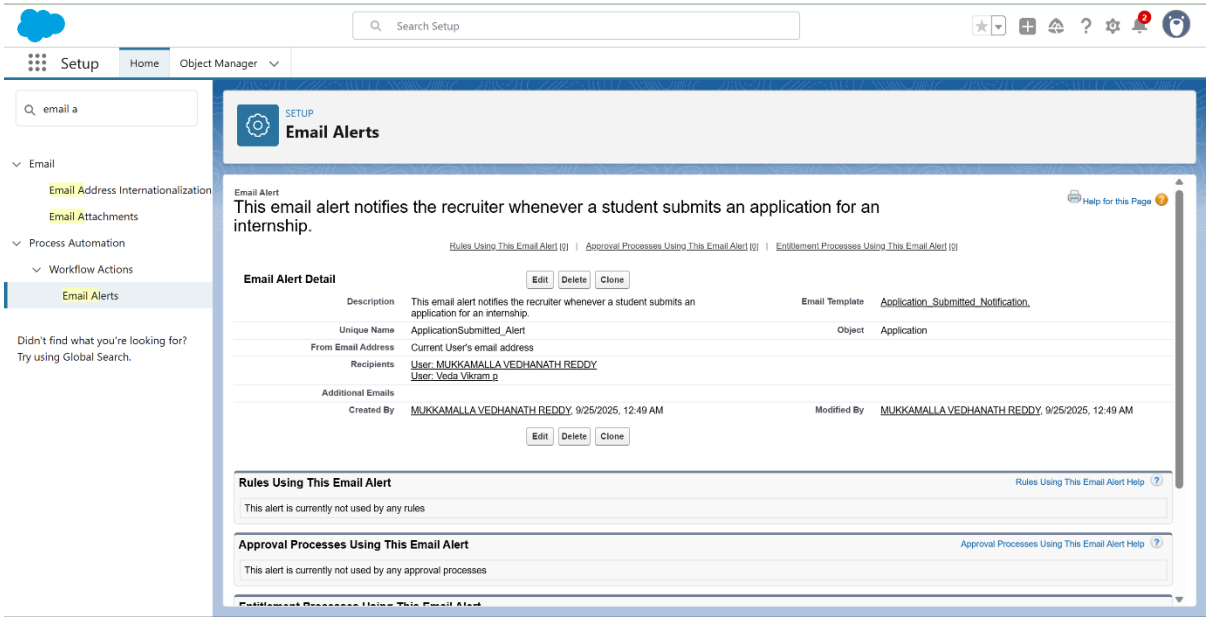
---

3. Email Alerts: Application Submitted Mail

For this project, I have created an email alert named **Application_Submitted_Mail** that works in conjunction with the **Application Notification for Recruiter** flow. This email alert automatically sends a notification to the recruiter when a student applies for an internship. The email contains key details such as the student's name, internship name, and application status so the recruiter can take prompt action.

- **Email Alert Name:** Application_Submitted_Mail
- **Object:** Application
- **Trigger:** Used within a Record-Triggered Flow (when a new Application record is created with Status = "Applied")
- **Purpose:** To notify recruiters instantly when a student applies for an internship, ensuring quick processing of applications.
- **Logic:** The email alert is triggered by the flow when conditions are met, sending a pre-defined email template to the recruiter containing relevant application details.

- **Benefit:** Improves recruiter response time, reduces manual communication, and ensures timely handling of internship applications.
- **Description:** This email alert works as part of process automation to ensure recruiters receive timely notifications when a student applies for an internship. It provides consistent, accurate, and fast communication between the system and recruiters.
- **Outcome:** Recruiters receive automatic email notifications for each new application with status “Applied”, helping streamline the recruitment process.



---

4. Field Updates: Placement Status Update

For this project, I have created a field update action within a flow to automatically update the placement status of a student when certain conditions are met. This ensures accurate tracking of the placement process without manual intervention.

- **Field Update Name:** Update_Placement_Status
- **Object:** Placement
- **Field:** Status
- **Trigger:** Used inside a Record-Triggered Flow (when a Placement record is created or updated).
- **Purpose:** To automatically update the placement status based on defined conditions, ensuring consistent and accurate data.
- **Logic:** The flow checks the Placement record when created or updated. If certain conditions are satisfied (such as application approval or completion), the Status field is updated automatically to values like "Placed", "In Progress", or "Completed".
- **Benefit:** Improves efficiency by automating status updates, eliminates manual errors, and keeps placement tracking consistent.

- **Description:** This field update action is part of the automated process that ensures placement statuses are kept up to date without manual effort. It provides accurate, real-time updates so recruiters, students, and admins have correct placement information.
- **Outcome:** The placement status is updated automatically, ensuring transparency, smooth workflow, and accurate reporting.

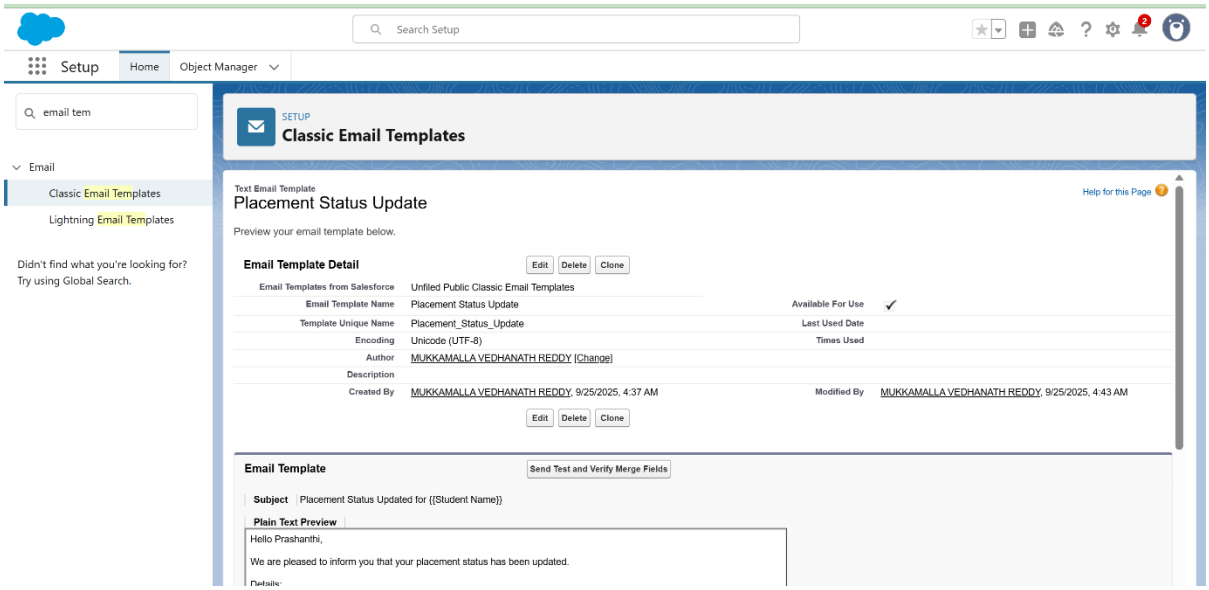


Fig4.1 Placement Email template

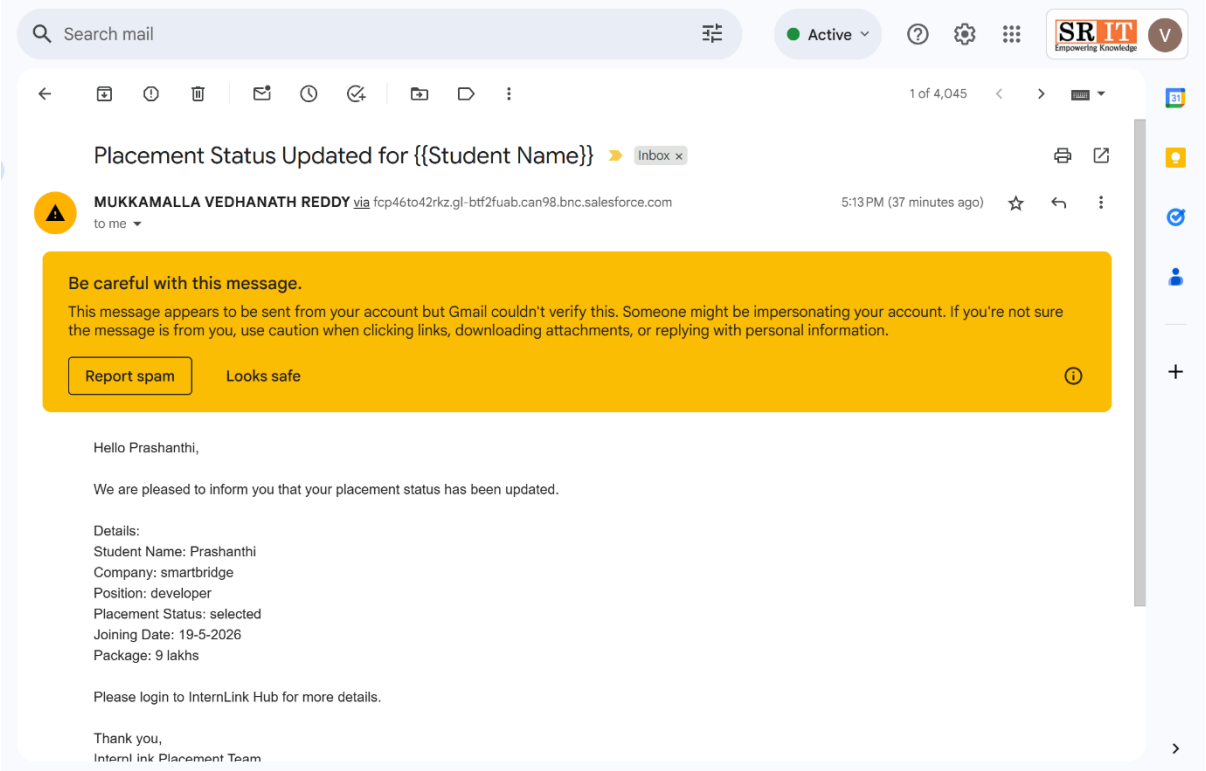


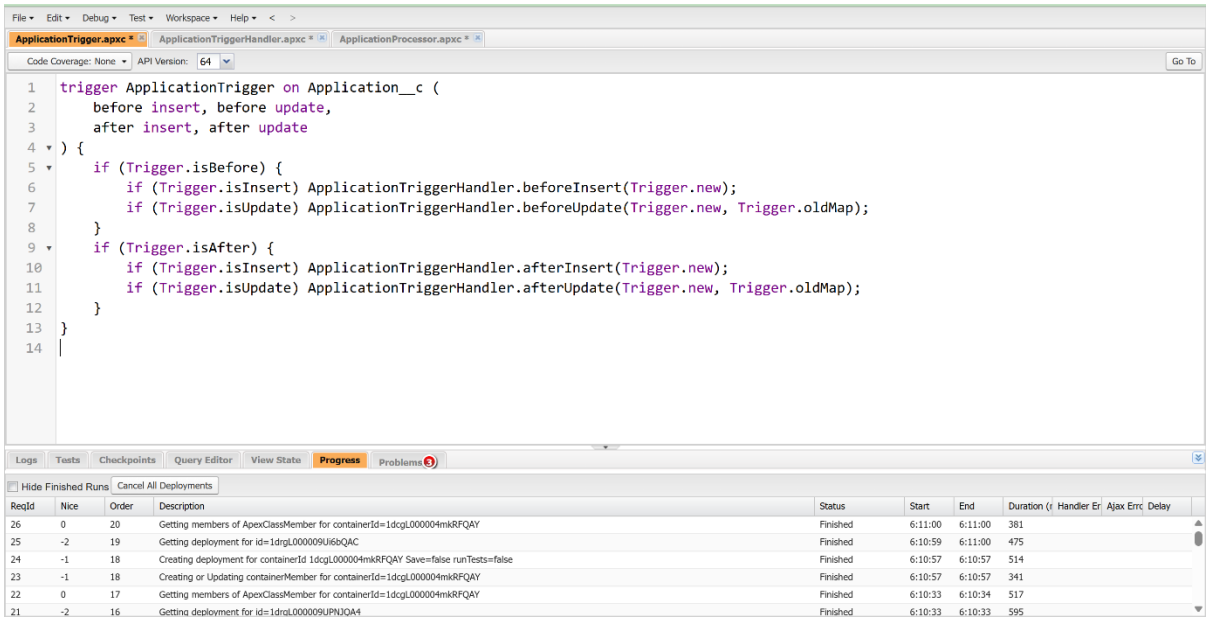
Fig 4.2 Placement Email

InternLink Hub -"A Central Platform for Internships & Placements"

Phase 5: Apex Programming (Developer) — Requirements for InternLink Hub

1.Classes & Objects

- Develop Apex classes to handle core project logic such as application processing, placement status updates, and recruiter notifications.
- Use Salesforce objects such as Student__c, Application__c, Placement__c, and Internship__c to store and manage relevant data efficiently.



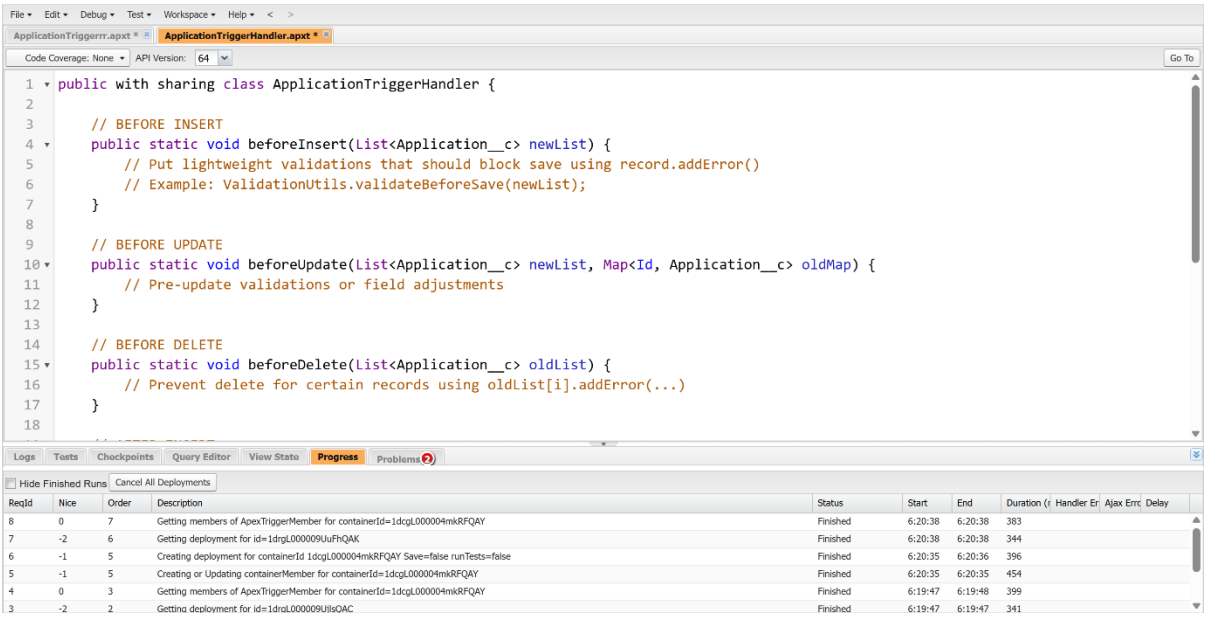
2. Apex Triggers

For this project, I have created Apex Triggers to automate actions on the Application__c object whenever records are inserted, updated, or deleted. Triggers are used to enforce complex business logic and ensure proper workflow execution beyond what declarative tools can achieve.

- **Trigger Name:** ApplicationTrigger
- **Object:** Application__c
- **Purpose:** To manage automation when a student applies for an internship, updates their application status, or when records are deleted. This ensures consistent behavior across the system.
- **Logic:**
  - **Before Insert/Update:** Perform data validations before saving records (e.g., ensuring required fields are filled correctly).
  - **After Insert:** Notify recruiters when a new application is created with status "Applied".
  - **After Update:** Update related records (such as Placement__c) or trigger notifications when the application status changes.

- **Before Delete/After Delete:** Control or track record deletions if needed, ensuring data integrity.

- **Benefit:** Helps enforce business rules, maintain data consistency, and automate recruiter notifications without manual intervention.
- **Description:** This trigger follows the **One Trigger per Object pattern** with a dedicated handler class. The handler manages different events (before insert, after insert, before update, after update, etc.), ensuring the logic is clean, reusable, and bulkified to handle multiple records efficiently.



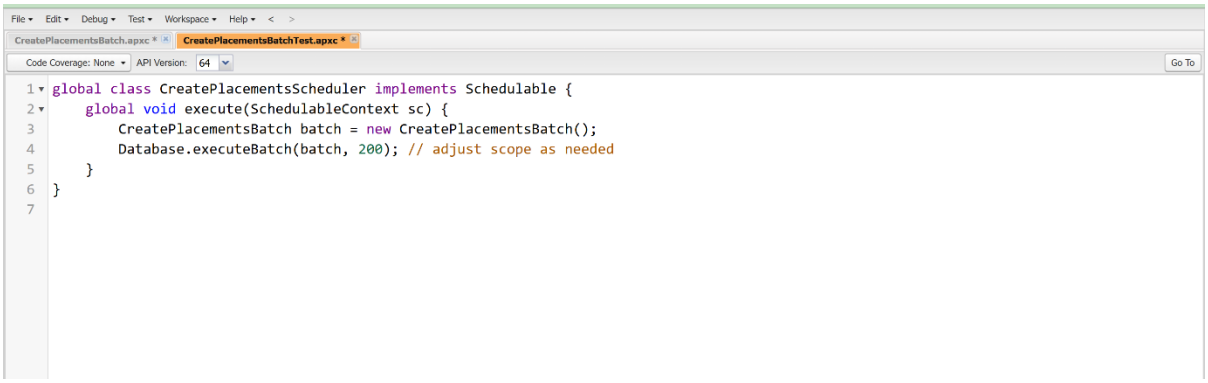
---

3. Batch Apex

For this project, I have created a Batch Apex job to handle large-scale data processing that cannot be executed synchronously. Batch Apex is used to process thousands of records efficiently in manageable chunks, ensuring automation and reporting for the InternLink Hub system.

- **Batch Class Name:** CreatePlacementsBatch
- **Object(s):** Application__c, Placement__c, Student__c, Internship__c
- **Purpose:** To automatically create Placement__c records for applications with status **"Applied"** where no placement exists, and to reconcile placement statuses in bulk. This ensures that placement data remains accurate and up-to-date across the system.
- **Logic:**
    - **Start:** Select all Application__c records with Status__c = 'Applied' and no related placement.
    - **Execute:** For each batch of records:
        - Collect Student and Internship details.

        - Create and insert missing Placement records.
        - Capture errors and log them for reporting.
    - **Finish:** Send a summary of the job (placements created, errors logged) and optionally chain follow-up processes such as notifications.
- **Benefit:** Improves efficiency by automating repetitive tasks, eliminates manual errors in placement creation, and ensures the system can process large volumes of records without hitting governor limits.
- **Description:** This Batch Apex job is designed following best practices (bulkification, error handling, and logging). It runs either on-demand by an admin or through a scheduled job (e.g., nightly), providing real-time accuracy in placement records and consistent reporting.

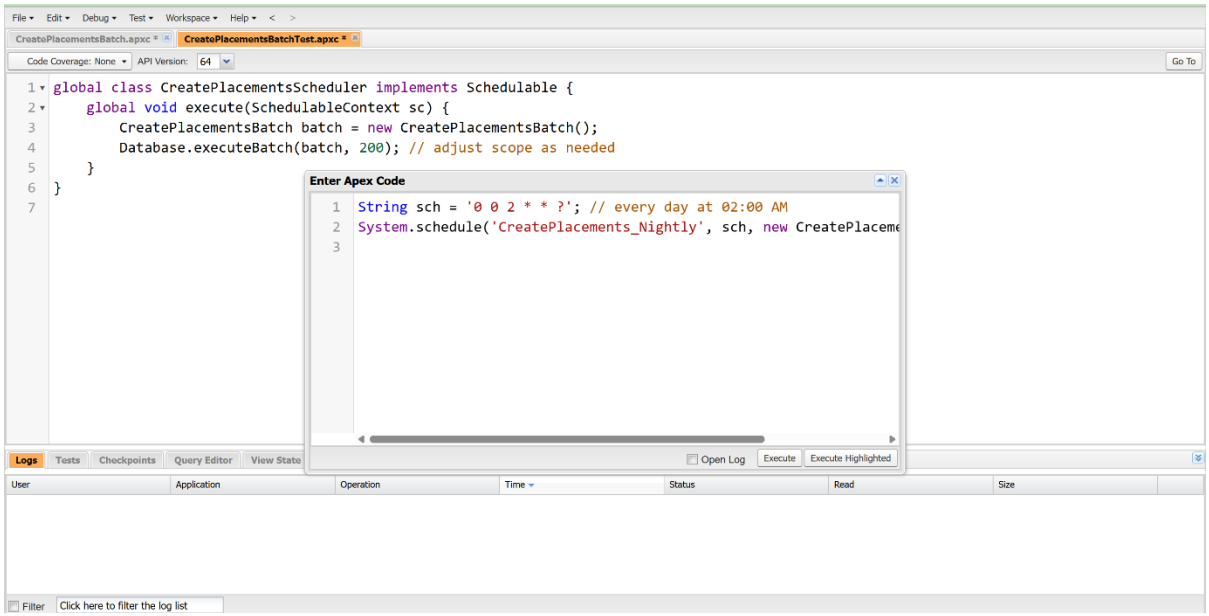


---

4. Asynchronous Processing

For this project, I have implemented Asynchronous Processing to handle time-consuming operations in the background without affecting user experience. This ensures efficiency, scalability, and compliance with Salesforce governor limits.

- **Techniques Used:** Batch Apex, Queueable Apex, Scheduled Apex, Future Methods.
- **Purpose:** To process large volumes of records, perform background tasks, and automate scheduled processes without blocking the main transaction.
- **Logic:**
    - **Batch Apex:** Large dataset processing in chunks.
    - **Queueable Apex:** Smaller, queued jobs for notifications or dependent tasks.
    - **Scheduled Apex:** Run jobs at specific times (e.g., nightly updates).
    - **Future Methods:** Perform asynchronous callouts or background updates.
- **Benefit:** Improves system performance, enables automation, reduces manual work, and ensures scalable processing for growing datasets.
- **Description:** Asynchronous processing in InternLink Hub enables reliable execution of background tasks, freeing resources for user operations while maintaining data accuracy and efficiency.



Here's the **documentation-style entry** for **Exception Handling** in the same format as your earlier sections:

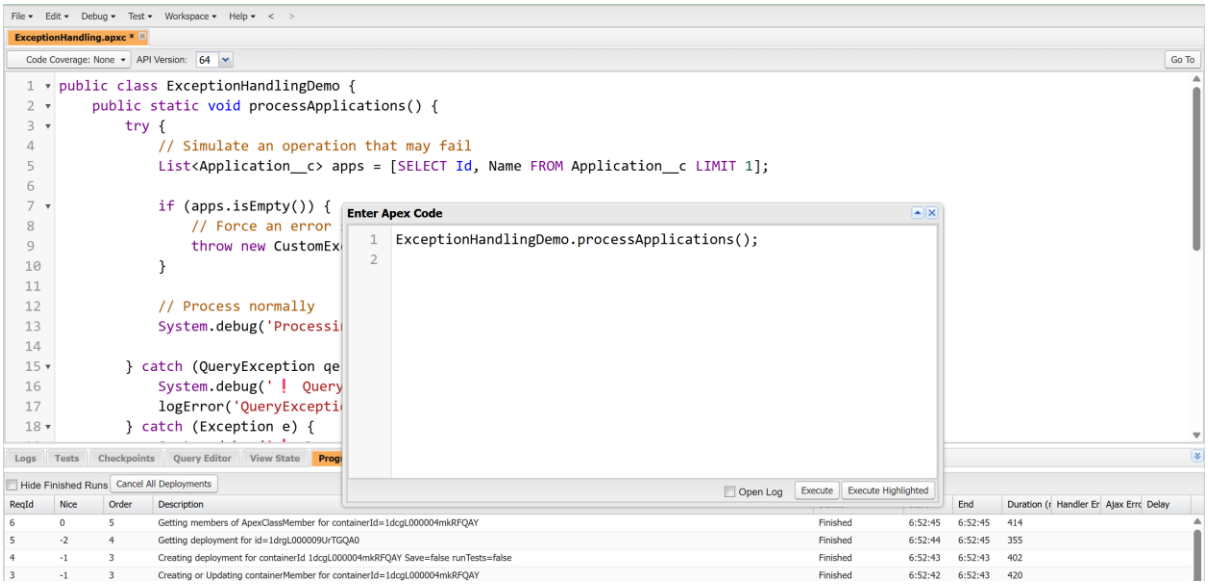
---

5. Exception Handling

For this project, I have implemented Exception Handling to ensure robust error management across all Apex classes, triggers, and asynchronous processes in the InternLink Hub system. Exception handling improves system stability, provides meaningful error feedback, and ensures smooth execution of automation even when unexpected errors occur.

- **Purpose:** To detect, manage, and log errors gracefully without disrupting the system processes or user experience.
- **Logic:**
    - Use try-catch blocks in Apex classes, triggers, and batch jobs to handle runtime exceptions.
    - Catch specific exceptions where possible (e.g., DmlException, QueryException) and handle accordingly.
    - Log errors with relevant context into a custom object (e.g., Error_Log__c) or send email alerts to admins.
    - Maintain user-friendly error messages without exposing technical details.
- **Benefit:**
    - Improves system reliability and stability.
    - Helps identify issues quickly through logs and alerts.
    - Prevents full process failure by handling errors gracefully.
- **Description:** Exception handling in InternLink Hub follows best practices for Salesforce development, ensuring that all automated processes handle errors effectively, log

important details for troubleshooting, and notify appropriate stakeholders without breaking the user experience.



.

InternLink Hub -"A Central Platform for Internships & Placements"

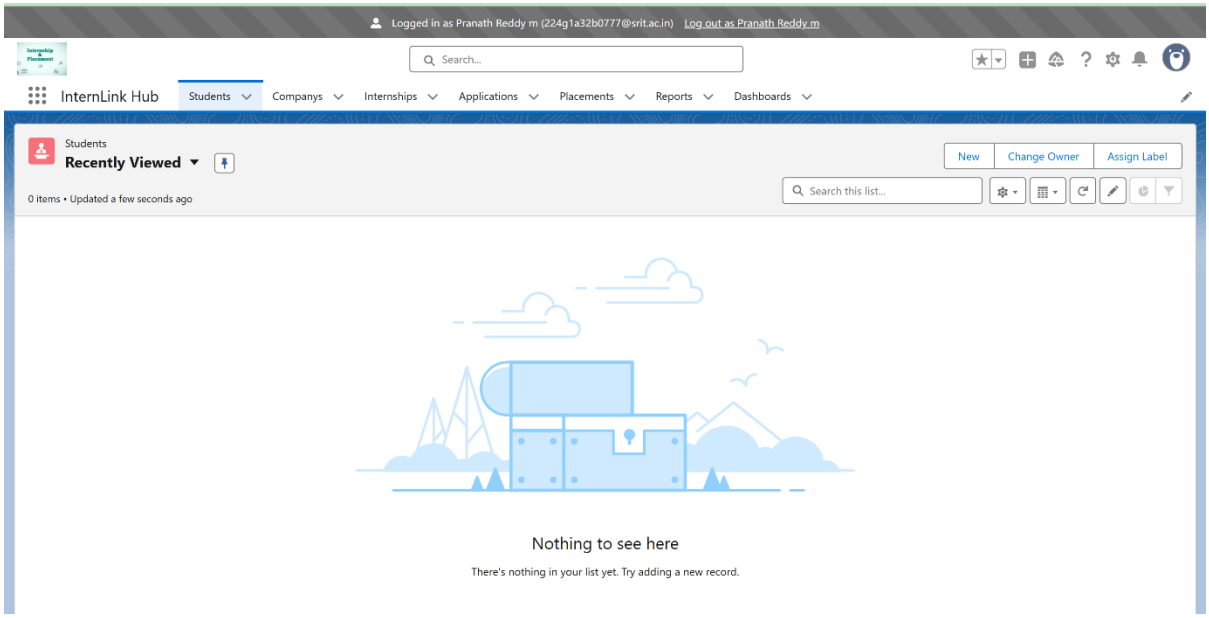
Phase 6: User Interface Development — InternLink Hub CRM

Phase 6 focuses on designing a clean, intuitive, and efficient user interface for the InternLink Hub CRM using Salesforce Lightning tools and Lightning Web Components (LWC). This phase ensures recruiters, students, and admins can easily access and interact with data, enhancing workflow and productivity.

---

1. Lightning App Builder

- **Purpose:** Create and customize pages for InternLink Hub objects to match CRM workflow.
- **Implementation:**
    - Designed custom Lightning Record Pages for Application__c, Student__c, Placement__c, and Internship__c.
    - Used Lightning App Builder's drag-and-drop interface to arrange fields, related lists, and LWCs.
- **Benefit:** Enables flexible page layouts tailored to InternLink Hub's needs without coding.



Fig;-1.1 Placement officer View(can access all the students info,companys,intership,application,placements)

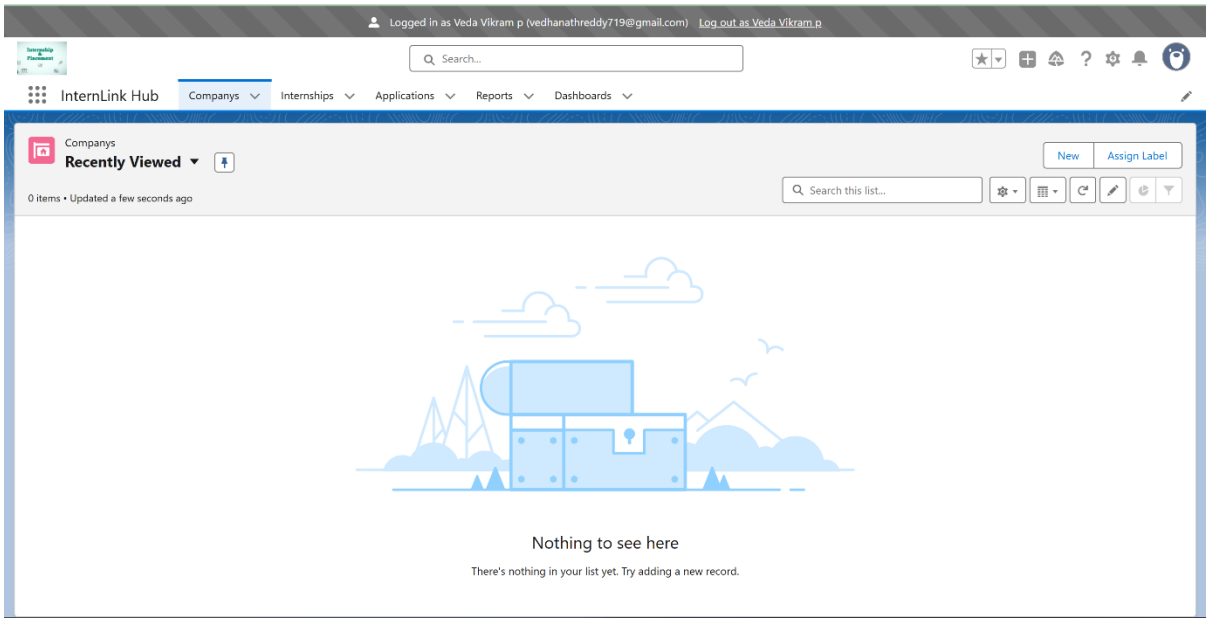


Fig 1.2 Recruiter view(can only access company,Internship,Application)

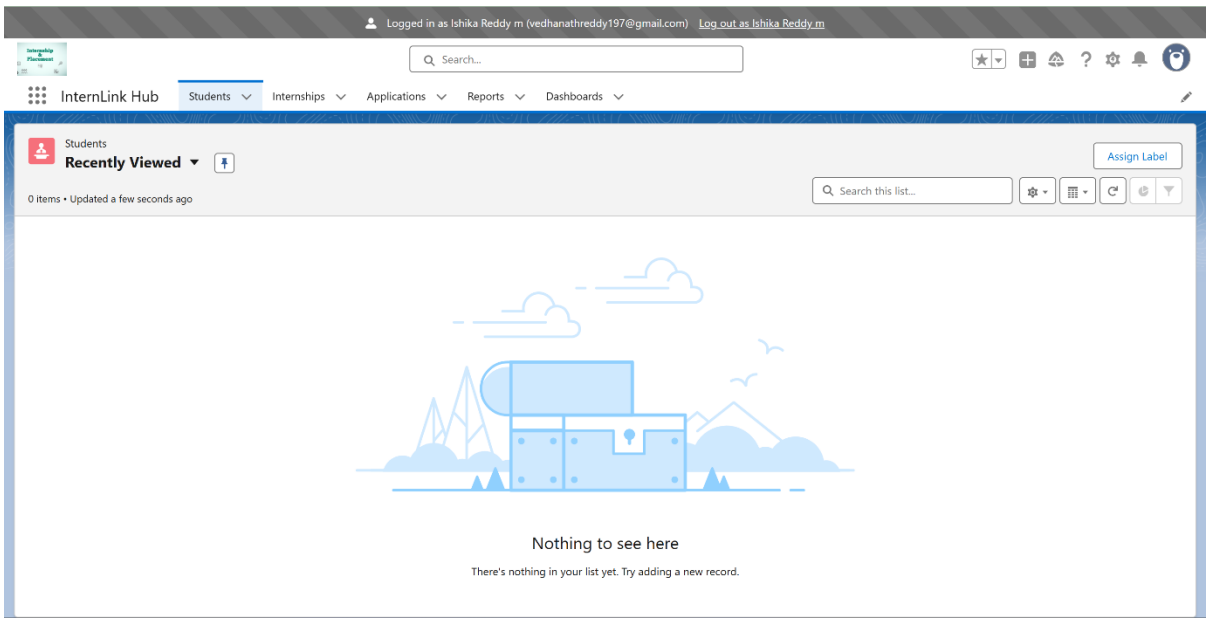


Fig 1.3 Student View(can only access student,internship,application)

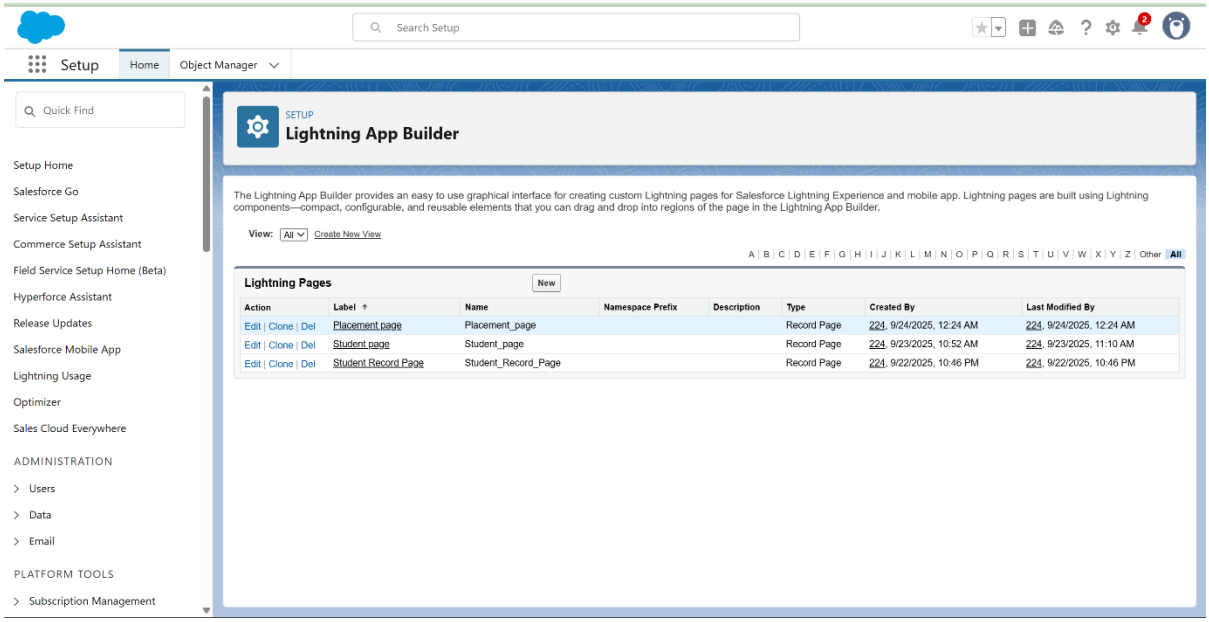
---

2. Record Pages

- **Purpose:** Provide user-specific views of records to improve clarity.
- **Implementation:**

- Customized record pages for each object so users can see essential information quickly.
- Added components for quick access to related data like internship details, student profiles, and application status.

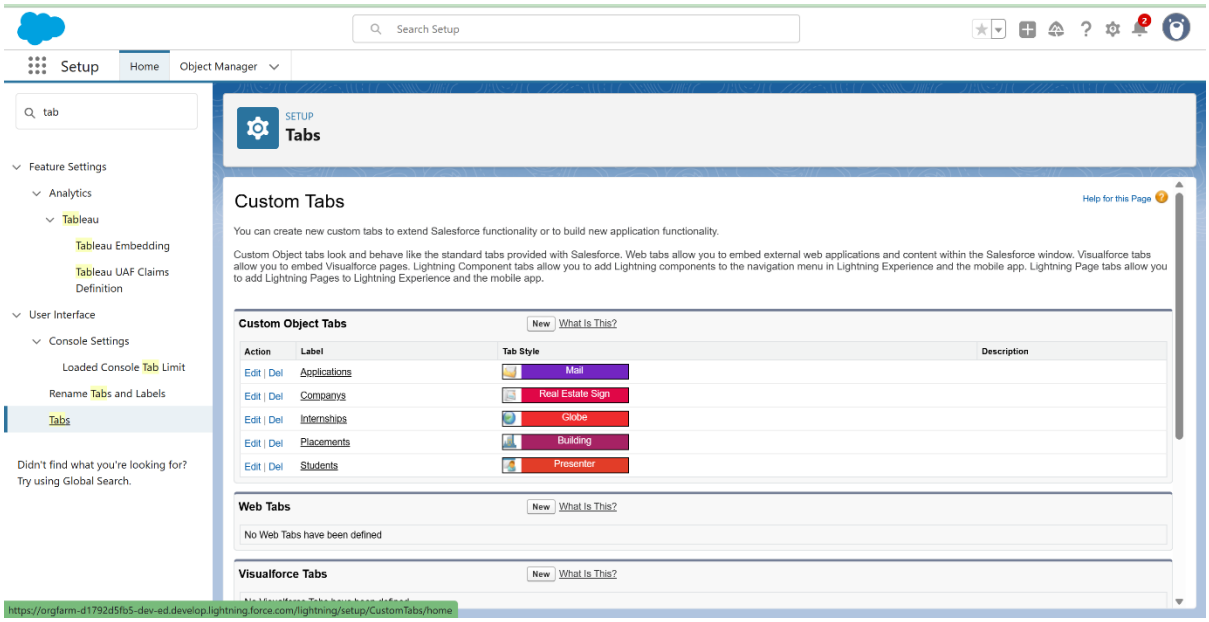
- **Benefit:** Improves efficiency for recruiters and admins by presenting relevant data clearly.



---

3. Tabs

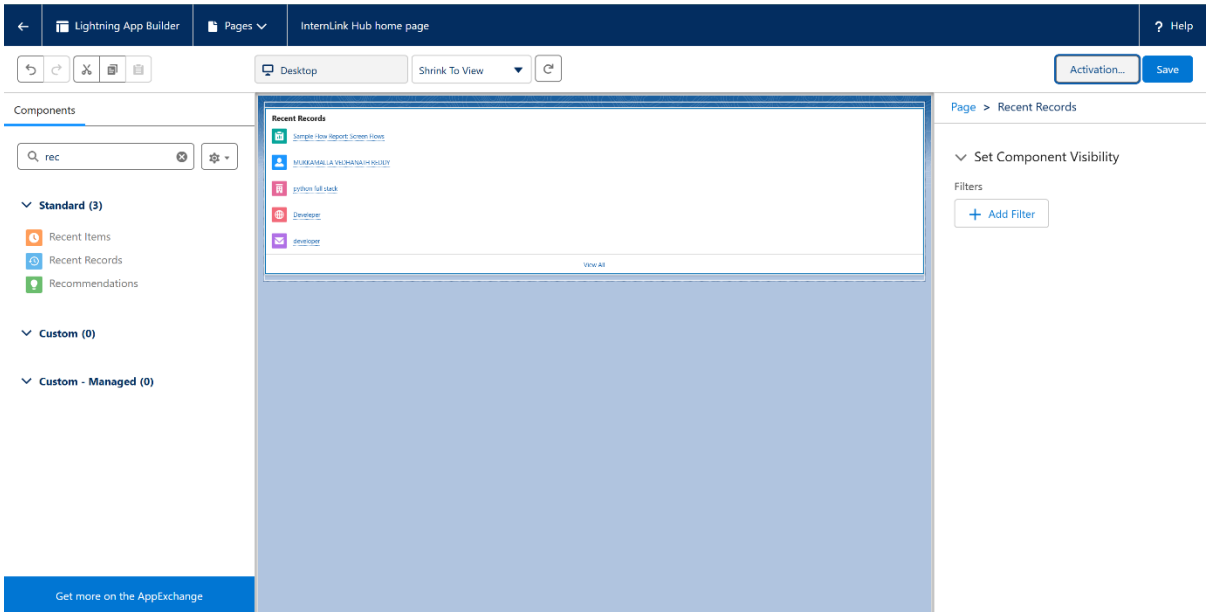
- **Purpose:** Organize CRM data access logically.
- **Implementation:**
  - Created custom tabs for Students, Applications, Placements, and Internships.
  - Configured tab visibility for different profiles (Recruiter, Admin, Student).
- **Benefit:** Simplifies navigation and aligns with user workflows.



---

4. Home Page Layouts

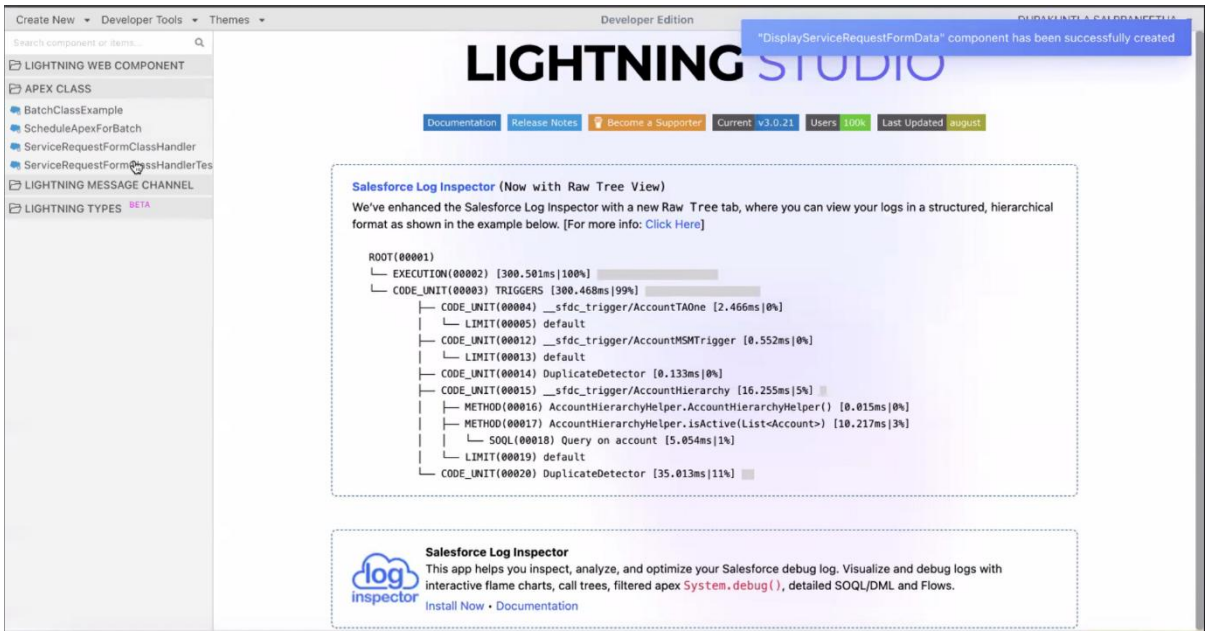
- **Purpose:** Provide a central dashboard for CRM users.
- **Implementation:**
  - Built a custom Home Page for InternLink Hub using Lightning App Builder.
  - Added components such as Recent Applications, Placement Status Charts, and Notifications.
- **Benefit:** Enhances productivity by displaying important information at a glance.



---

5. Lightning Web Components (LWC)

- **Purpose: Create dynamic, reusable UI components to improve the interactivity and usability of InternLink Hub CRM.**
- **Implementation:**
    - **Developed custom LWCs for key CRM functionalities, such as:**
        - **Application Submission Form — allows recruiters/students to submit internship applications easily.**
        - **Placement Status Updates — displays and updates placement status in real time.**
        - **Recruiter Notifications — shows alerts for new applications or status changes.**
        - **Placement & Application Dashboards — provides interactive summaries of applications and placements.**
    - **Used Salesforce Lightning Design System (SLDS) for consistent styling across components.**
    - **Followed a modular design pattern so components can be reused across different record pages and apps.**
- **Benefit: Enhances user experience with responsive, fast, and reusable components that streamline CRM processes.**



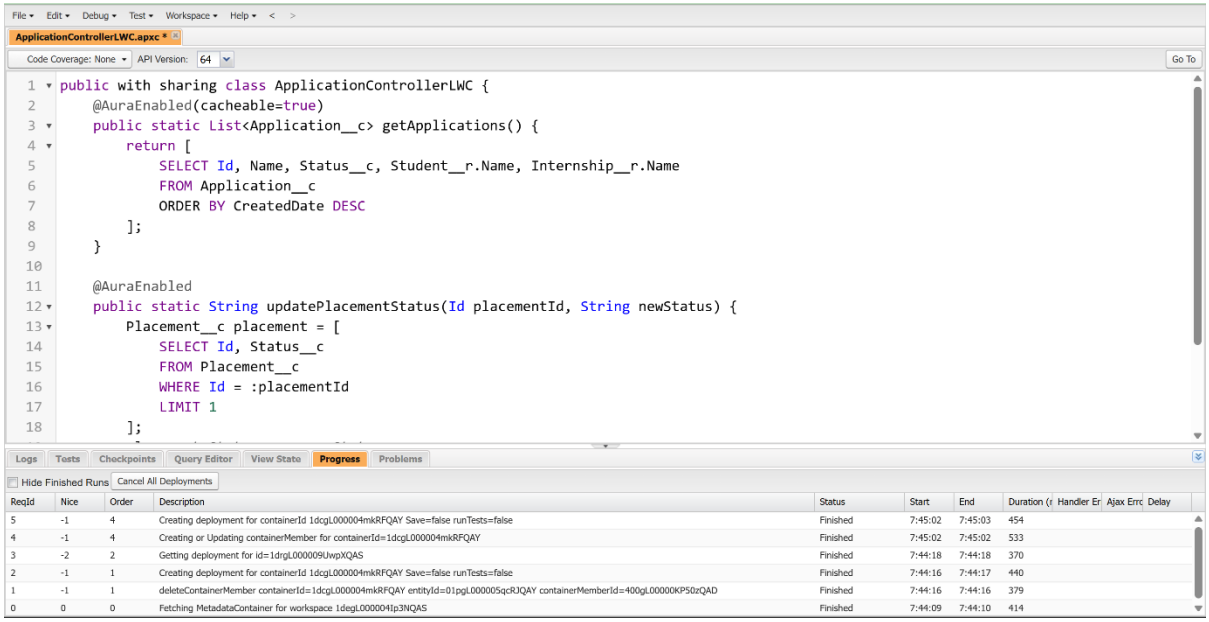
---

6. Apex with LWC

- **Purpose: Enable Lightning Web Components to communicate with backend Apex logic for advanced CRM functionality.**
- **Implementation:**

- **Created Apex classes annotated with @AuraEnabled to allow LWCs to access Salesforce data and execute business logic.**
- **Used Apex methods to handle operations such as:**
  - **Retrieving application, student, placement, and internship records.**
  - **Updating placement statuses based on recruiter actions.**
  - **Sending notifications to recruiters.**
- **Ensured Apex methods follow best practices for bulk processing and governor limits compliance.**
- **Integrated these Apex methods into LWCs using imperative calls and wire adapters.**

- **Benefit: Extends the capability of LWCs by allowing complex operations and business rules to be executed efficiently in the backend.**

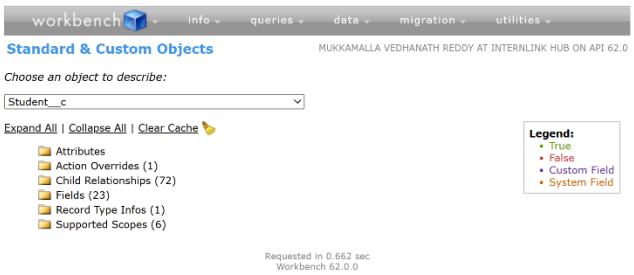


InternLink Hub -“A Central Platform for Internships & Placements”

Phase 7: Integration & External Access — InternLink Hub CRM

1. External Services

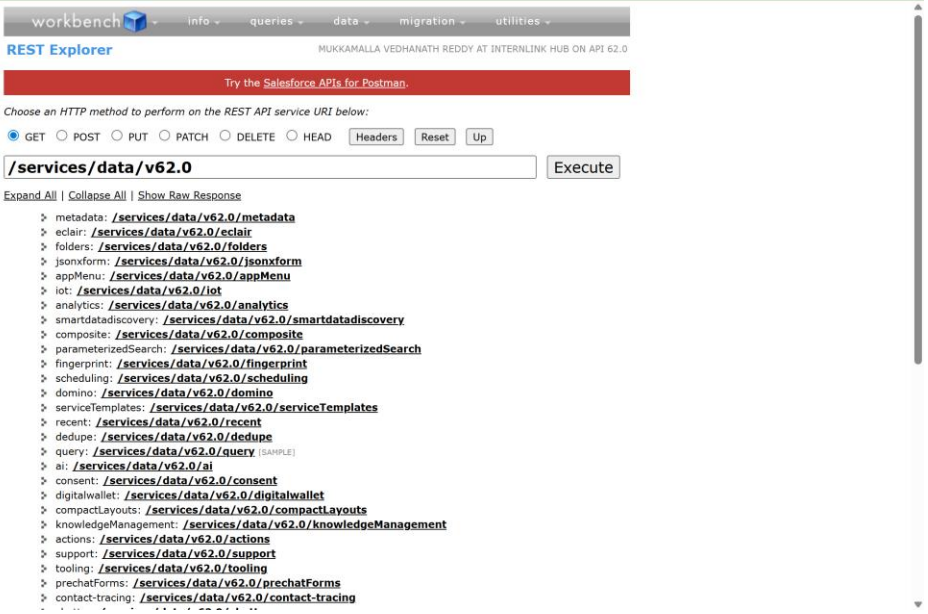
- **Purpose:** Connect Salesforce to external APIs in a declarative way without heavy coding.
- **Implementation:**
    - Used **Workbench** to register and test external APIs.
    - Imported API specifications (e.g., OpenAPI/Swagger) into Salesforce External Services.
    - Configured Named Credentials to securely authenticate API access.
- **Benefit:** Allows seamless integration with external services while reducing development complexity, enabling quick access to external internship-related data.



2. Web Services (REST)

- **Purpose:** Exchange data between Salesforce and external systems using REST APIs to enhance CRM capabilities.
- **Implementation:**
    - Built **Apex classes** to perform REST callouts for retrieving and updating internship-related data.
    - Configured **Named Credentials** or **Remote Site Settings** to securely access external endpoints.
    - Tested the REST callouts using **Salesforce Workbench** to ensure accuracy and proper data handling.

- **Benefit:** Enables dynamic interaction with external systems, allowing InternLink Hub CRM to fetch, update, or synchronize data in real time without manual entry.

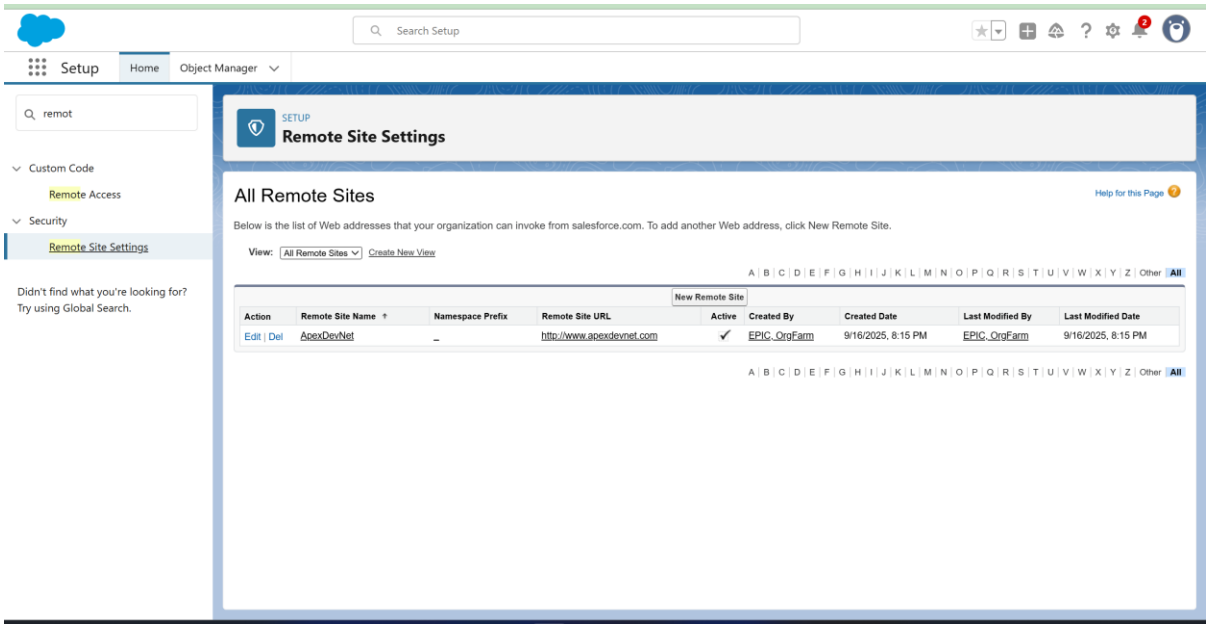


3.OAuth & Authentication

- **Purpose:** Securely authenticate integrations with external systems or applications, ensuring that data exchange is safe and compliant with security standards.
- **Description:** OAuth is an industry-standard protocol for authorization that enables secure access to external APIs without exposing user credentials. In InternLink Hub CRM, OAuth is configured using Salesforce Authentication Providers, allowing integrations with services such as Google APIs or LinkedIn APIs for internship data. This ensures that all data interactions are authorized and secure.

4.Remote Site Settings

- **Purpose:** Whitelist external domains so Salesforce can safely perform callouts to them.
- **Description:** Salesforce requires explicit permission to connect to external web services. Remote Site Settings store the URLs of trusted external APIs, enabling secure callouts. For InternLink Hub CRM, Remote Site Settings are configured for external internship portals or notification services so that Apex callouts and integrations work without security errors.



5.Named Credentials

- **Purpose:** Securely store authentication details for external APIs and simplify callouts.
- **Description:** Named Credentials store the URL, authentication type, and credentials for external services. This avoids hardcoding sensitive information in Apex code and makes integrations easier to maintain. In InternLink Hub CRM, Named Credentials are used to connect to third-party internship services and notification APIs, ensuring secure and reliable integrations while keeping credentials managed centrally.

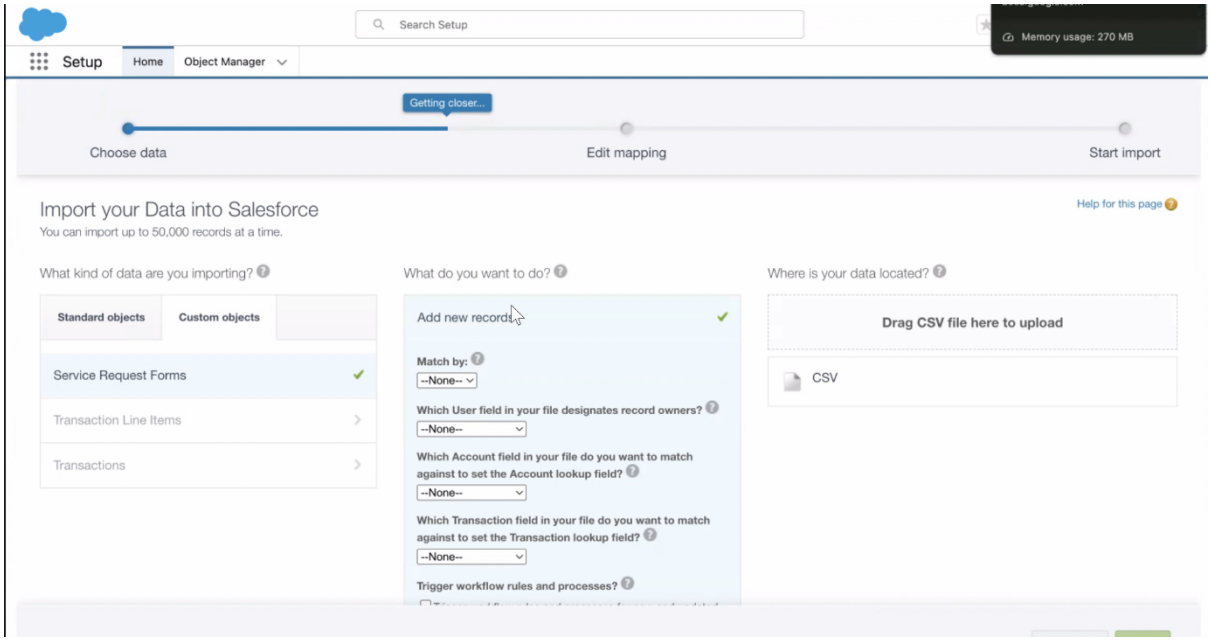
InternLink Hub -“A Central Platform for Internships & Placements”

Phase 8: Data Management & Deployment — InternLink Hub CRM

---

1. Data Import Wizard

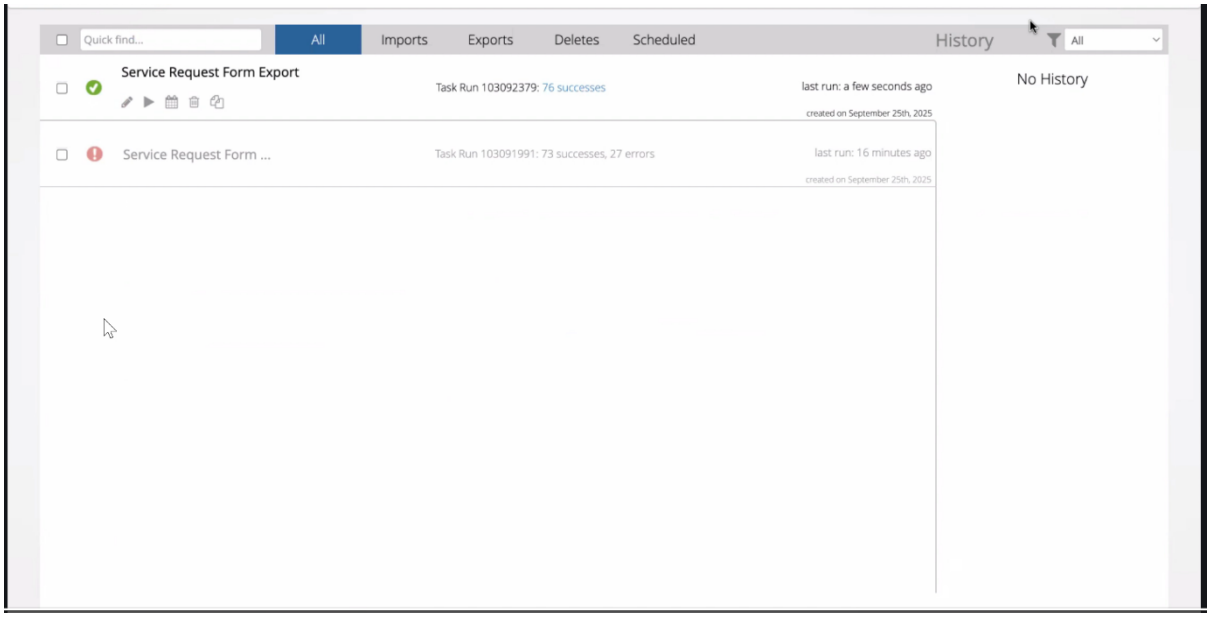
- **Purpose:** Import small to medium-sized datasets (up to 50,000 records).
- **Implementation:** Used to upload Student, Internship, and Placement data in bulk via CSV.
- **Benefit:** Quick, easy, and user-friendly tool for admins without requiring advanced technical skills.



---

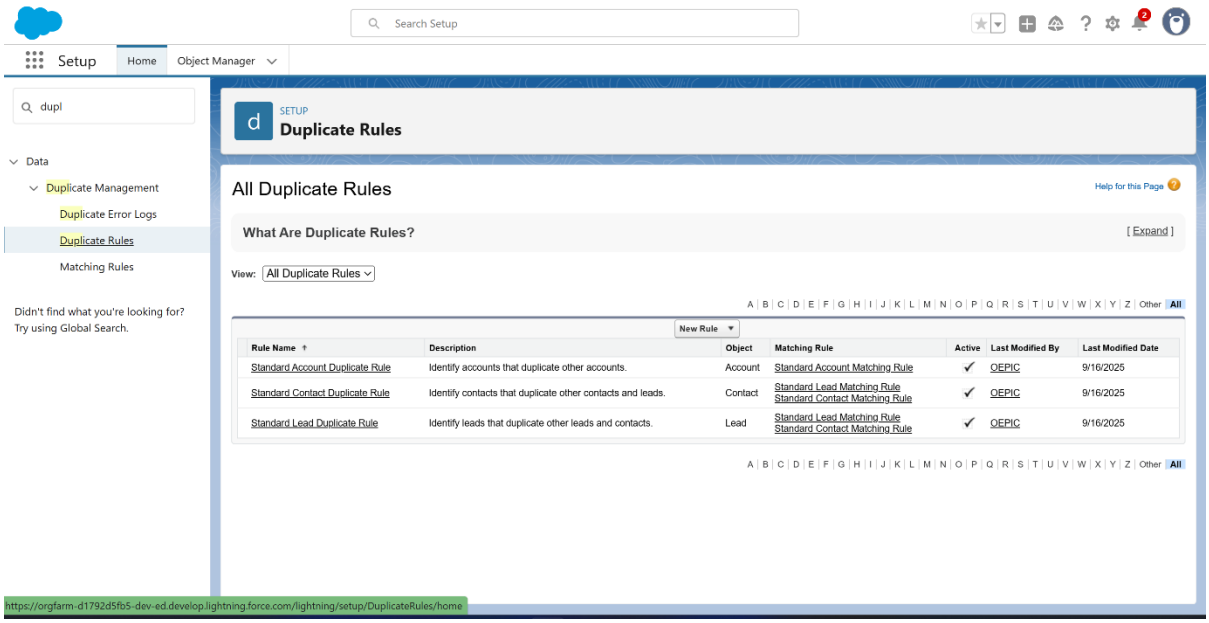
2. Data Loader

- **Purpose:** Handle large data volumes (up to millions of records) for import, export, update, and delete.
- **Implementation:** Used for bulk uploading Applications and updating Placement records.
- **Benefit:** Ensures efficiency for large datasets with advanced options like scheduled imports/exports.
- I have made the application as service request form



3. Duplicate Rules

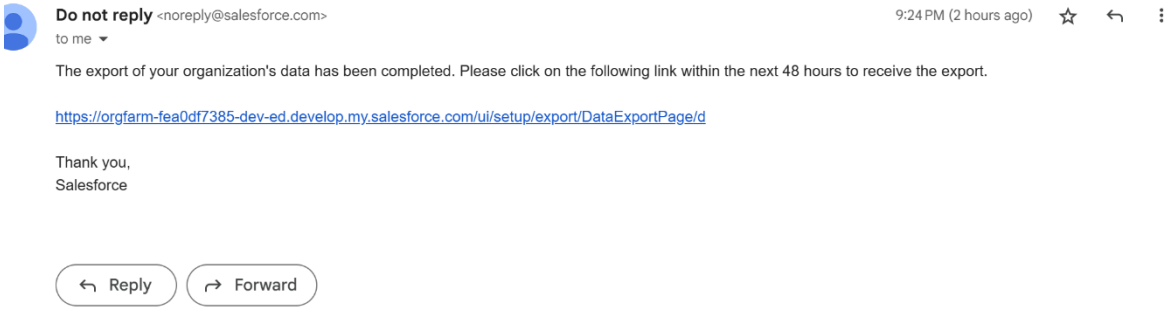
- **Purpose:** Prevent duplicate records for Students, Recruiters, and Internships.
- **Implementation:** Configured Duplicate Rules and Matching Rules to block or alert users when duplicates are detected.
- **Benefit:** Maintains clean, accurate data across the CRM.



4. Data Export & Backup

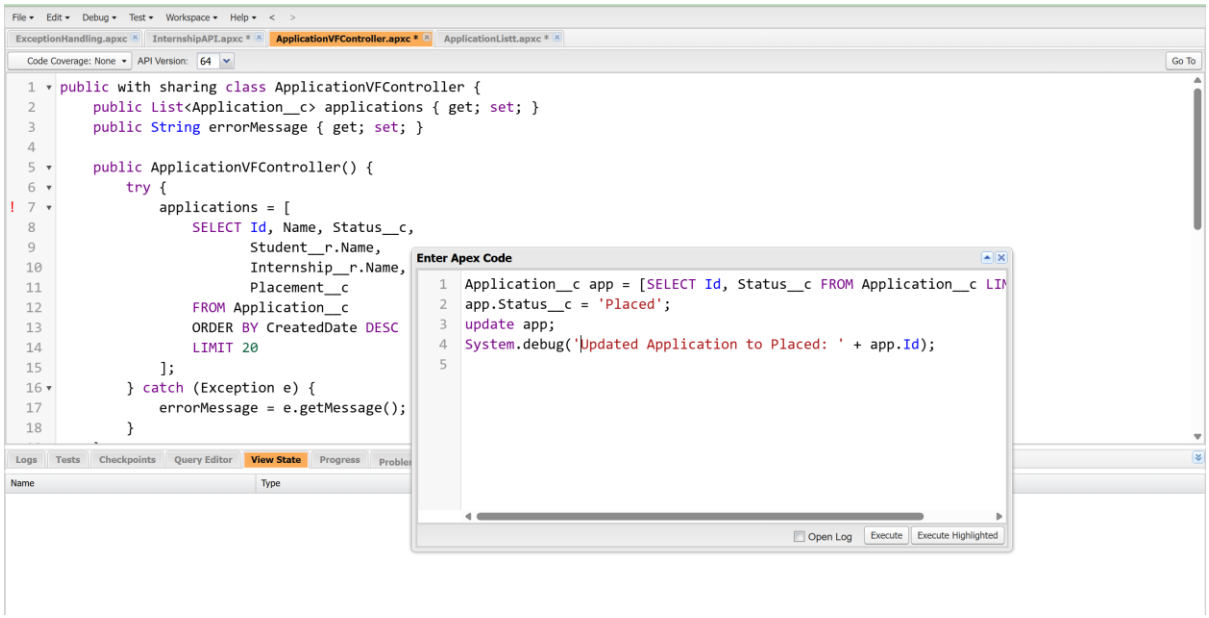
- **Purpose:** Securely back up CRM data to prevent loss and ensure recovery when needed.
- **Implementation:** Configured **weekly scheduled data exports** in Salesforce to back up **Student__c, Application__c, and Placement__c** records. Exports were generated in **.zip files containing .csv data** for each object. These files were stored securely for auditing and recovery purposes.

- **Benefit:** Protects against accidental deletions, corruption, or integration errors. Ensures that critical data for students, applications, and placements can be restored if issues occur.
- **Usage in InternLink Hub:** Provides a reliable backup strategy to maintain trust and data integrity for recruiters, students, and admins.
- We can export the dat which we want and download if we needed .
- After getting export we get mail like below.



5.Developer console:

For the InternLink Hub project, I have created custom Apex classes to handle automation and core business logic. These classes manage workflows such as application status updates, placement processing, and recruiter notifications. They interact with key custom objects like Student__c, Application__c, Placement__c, and Internship__c. This ensures efficient data handling and seamless process automation in the system.


File Edit Debug Test Workspace Help
ExceptionHandling.apxc
InternshipAPI.apxc *
ApplicationVFController.apxc *
ApplicationListt.apxc *
Code Coverage: None
API Version: 64
Go To
public with sharing class ApplicationVFController {
public List<Application__c> applications { get; set; }
public String errorMessage { get; set; }
public ApplicationVFController() {
try {
applications = [
SELECT Id, Name, Status__c,
Student__r.Name,
Internship__r.Name,
Placement__c
FROM Application__c
ORDER BY CreatedDate DESC
LIMIT 20
];
} catch (Exception e) {
errorMessage = e.getMessage();
}
Logs
Tests
Checkpoints
Query Editor
View State
Progress
Name
Type
Enter Apex Code
Application__c app = [SELECT Id, Status__c FROM Application__c LIM
app.Status__c = 'Placed';
update app;
System.debug('Updated Application to Placed: ' + app.Id);
Open Log
Execute
Execute Highlighted

InternLink Hub -“A Central Platform for Internships & Placements”

Phase 9: Reporting, Dashboards & Security Review

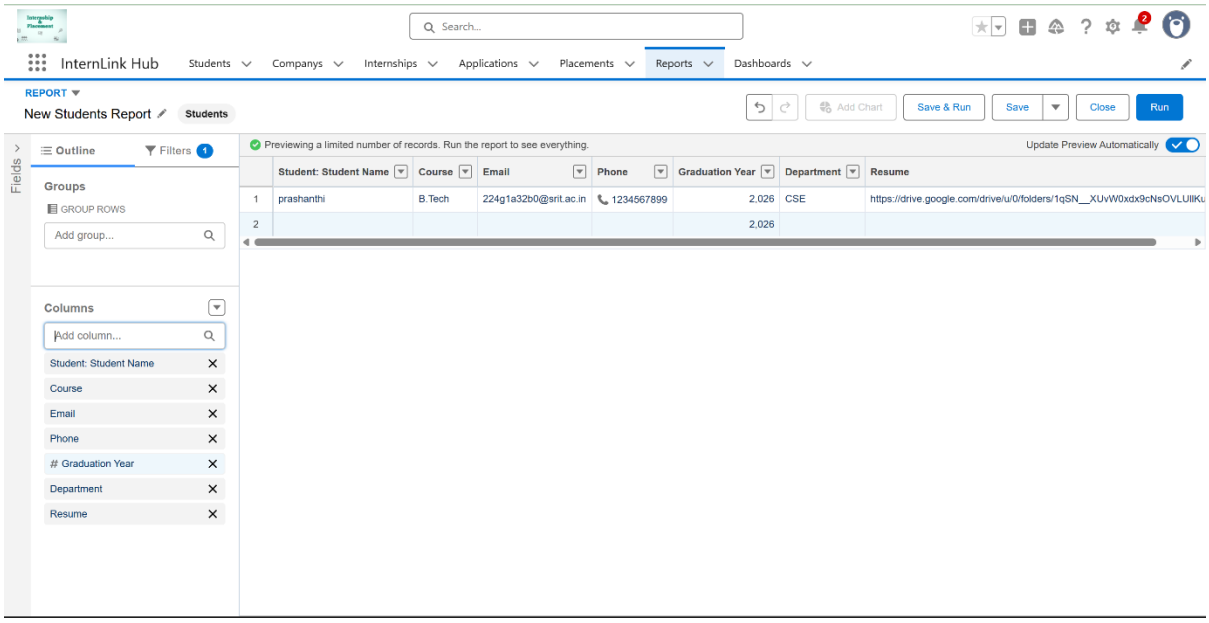
---

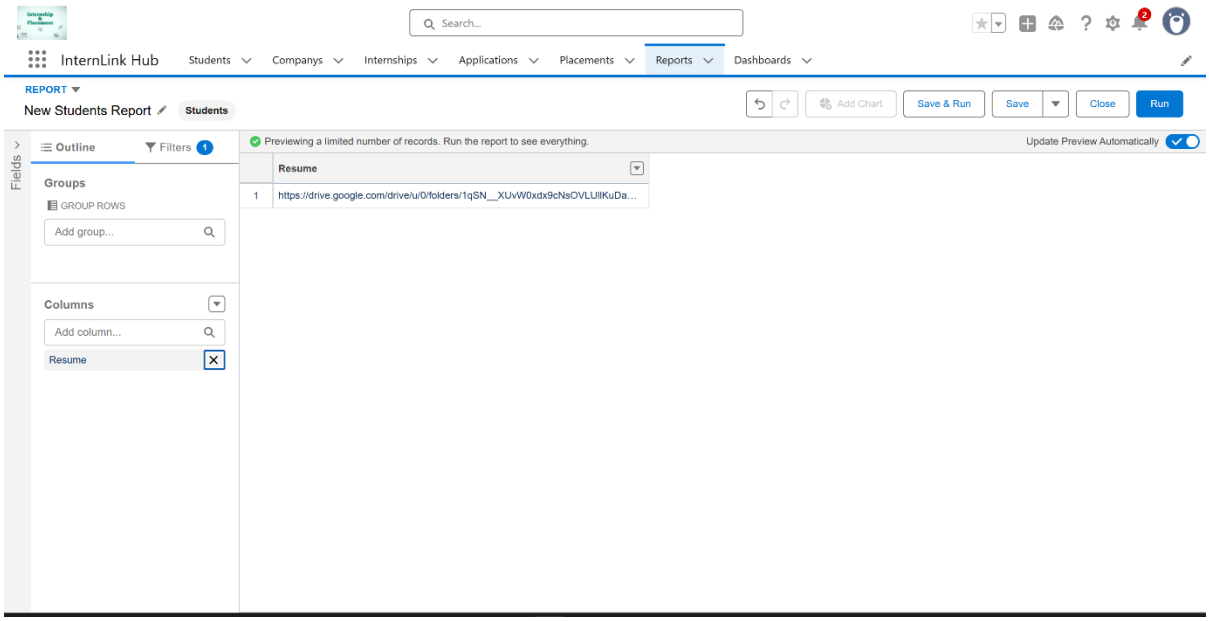
1. Reports

Reports in InternLink CRM help track and analyze data for informed decision-making. They are built using Salesforce’s reporting features to show information from custom and standard objects.

Types of Reports Created in InternLink:

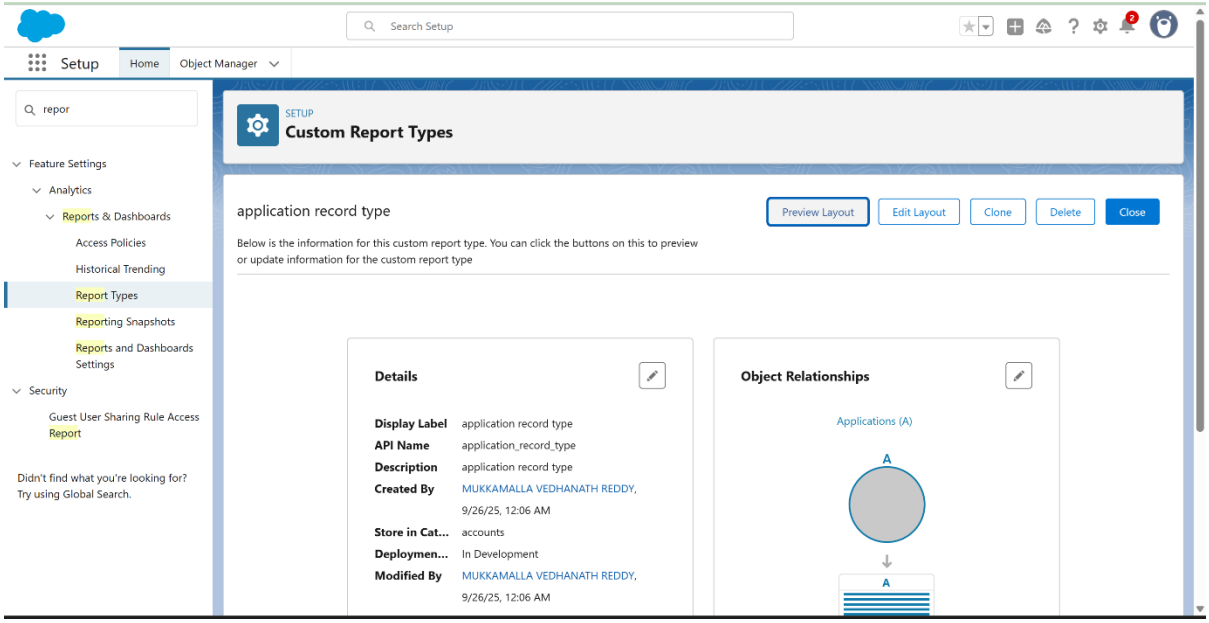
- **Tabular Report** → Simple list of records.
  Example: List of all Students with Name, Email, Course, Graduation Year.
- **Summary Report** → Groups data by a specific field.
  Example: Internship applications grouped by Company to show application count per company.
- **Matrix Report** → Groups data by both rows and columns for deeper analysis.





2.Record Type:

I have created a custom report type in InternLink CRM for Internship Reports, which includes key objects like *Internship__c*, *Company__c*, and *Application__c*. This report type allows viewing and analyzing internship details such as company name, internship title, stipend, duration, openings, and application status, providing a comprehensive view of all active and completed internships for better decision-making and tracking.



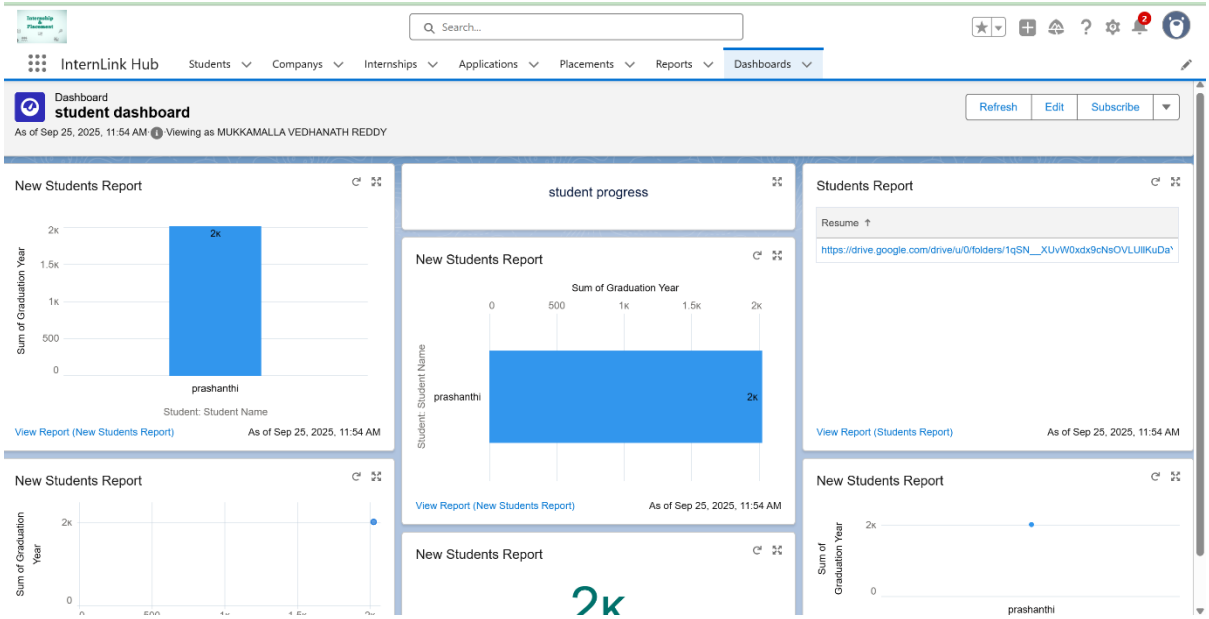
3. Dashboards in InternLink CRM

I have created dashboards in InternLink CRM to provide a clear and interactive view of internship and placement data. These dashboards help students and other stakeholders easily track applications, opportunities, and outcomes.

The dashboards include:

- **Bar charts** – to compare internship opportunities.
- **Pie charts –** to represent application status distribution**.**
- **Number charts –** to highlight key metrics such as total applications or confirmed placements.
- **Tables** – to show detailed placement records.

By combining these visual elements, the dashboards ensure quick understanding and better decision-making.

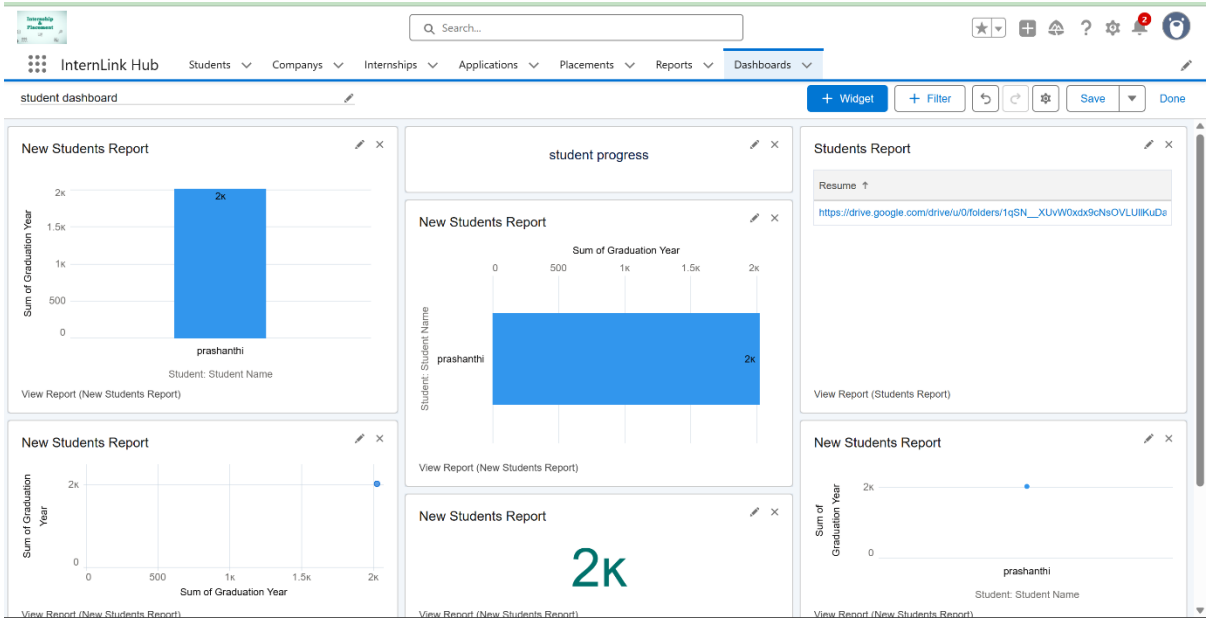


4.Dynamic Dashboards

Dynamic dashboards have been implemented so each user sees personalized, role-specific data without the need for multiple dashboard versions. This improves efficiency and ensures relevant insights for different profiles.

They were configured with the option to display data as the viewing user, which ensures proper data visibility and access control according to each user's role.

By setting up dynamic dashboards, InternLink CRM provides a single, unified dashboard experience while maintaining security and relevance for every user.



5. **Sharing Settings in InternLink CRM**

To ensure data security and proper access control, Sharing Settings were configured in InternLink CRM. These settings define who can view or edit records based on their role.

- **Organization-Wide Defaults (OWD):** Set to restrict access by default (e.g., Private for Applications, Controlled by Parent for related records).
- **Role Hierarchy:** Placement Officers, Recruiters, and Students have different levels of access according to their responsibilities.
- **Sharing Rules:** Configured to grant additional access where collaboration is needed

This ensures each user only sees the data relevant to their role, while maintaining security and privacy across the system.

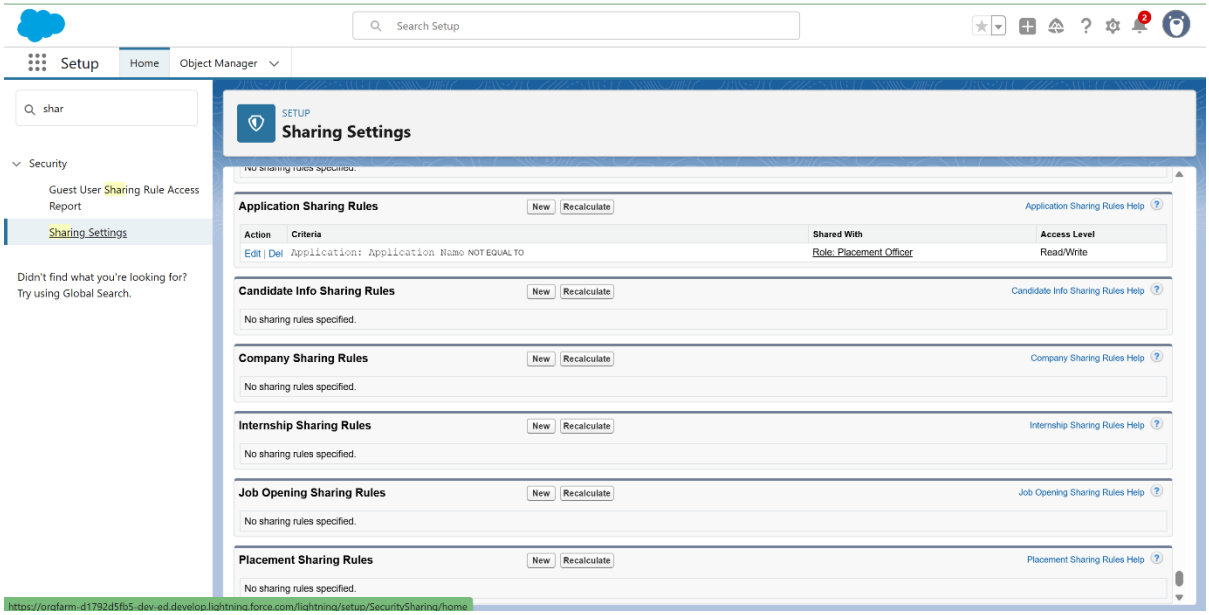


Fig 5.1 Sharing rules

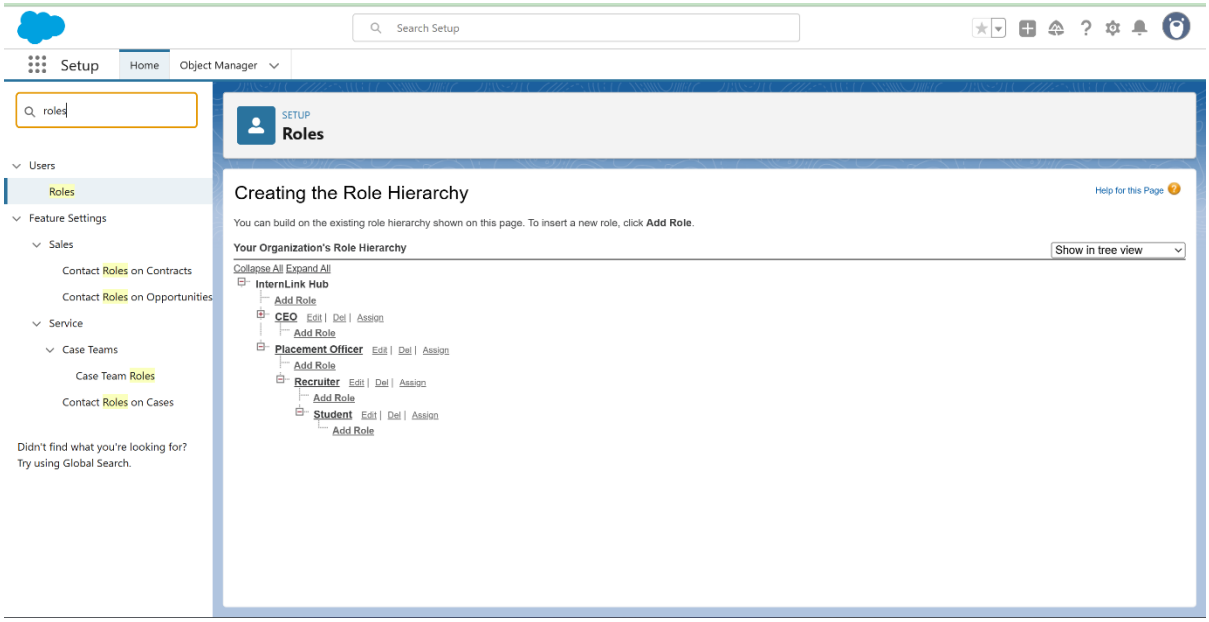


Fig:-5.2 Roles

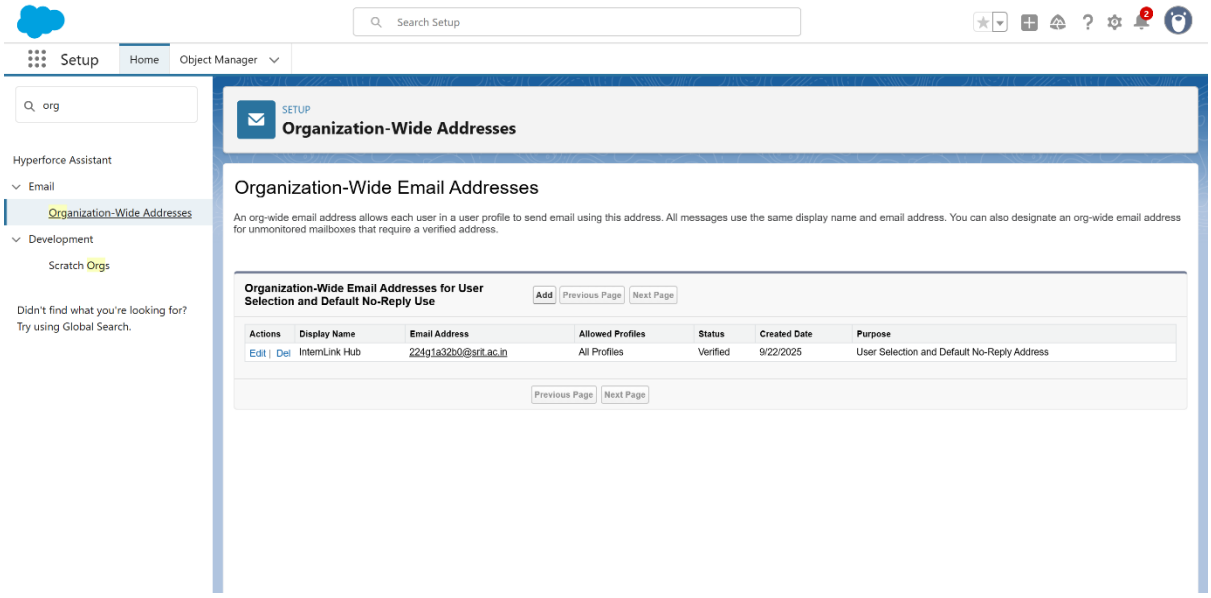


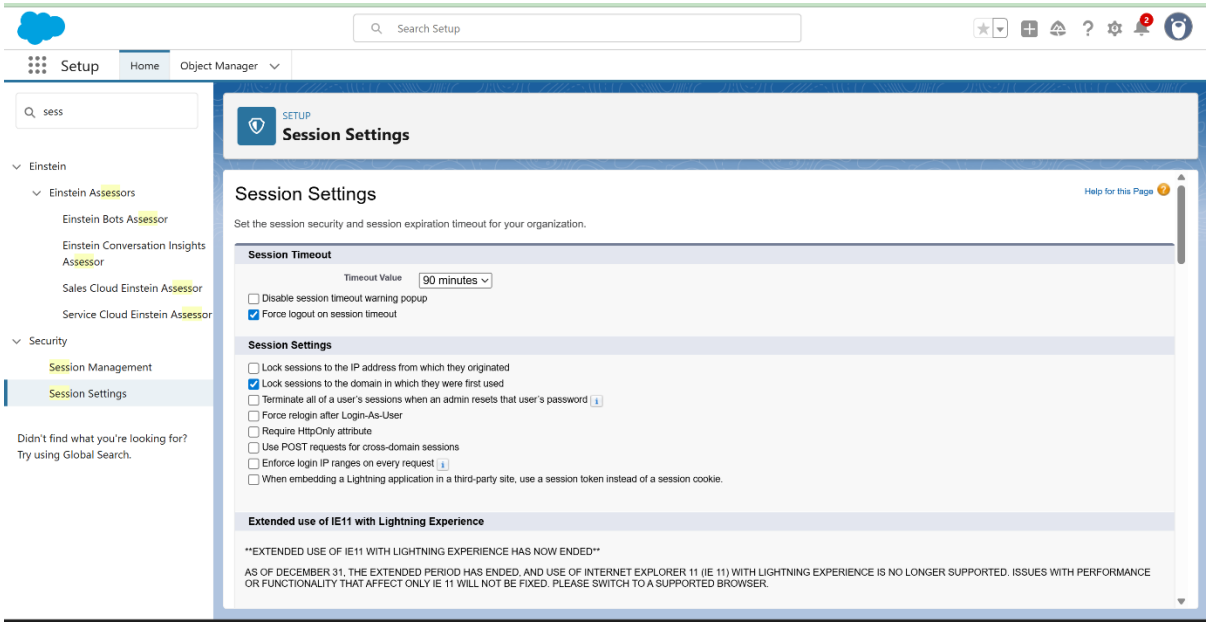
Fig5.3 **Organization-Wide Defaults (OWD)**

6.Session Settings in InternLink CRM

Session Settings were configured to improve **security** and **system control** for users in InternLink CRM. These settings define how user sessions behave, expire, and are secured.

- **Session Timeout:** Automatically logs out inactive users after a set duration (e.g., 30 minutes).
- **Login Security:** Prevents simultaneous logins from different locations or browsers.
- **HTTPS Enforcement:** Ensures all sessions are secured through HTTPS.
- **Session Locking:** Binds sessions to a specific IP address to prevent hijacking.
- **Reauthentication:** Users must re-enter credentials before performing sensitive actions such as editing placements or student data.
- **Force Relogin after Login-As-User:** Ensures admins who use "Login-As-User" must reauthenticate after exiting, maintaining account security.

These measures balance **data protection** and **user experience**, keeping sensitive internship and placement data secure.

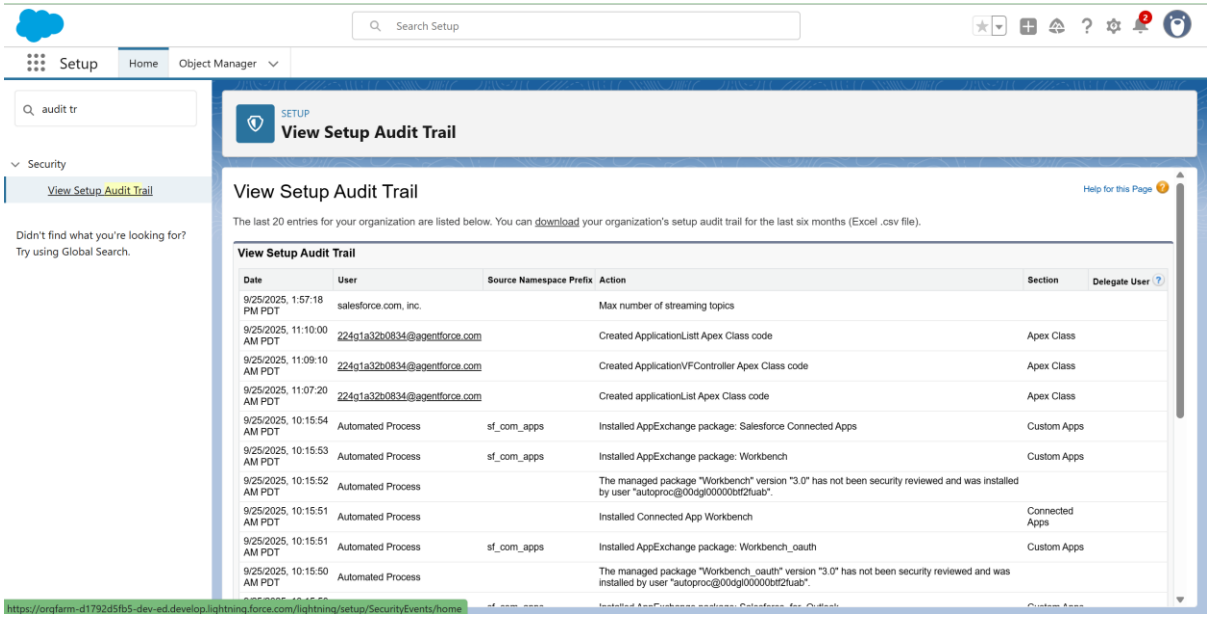


---

7. Audit Trail

- **Purpose**: Audit Trail in Salesforce tracks all administrative setup changes to maintain accountability and security.
- **Features**: Captures information such as who made the change, when it was made, and what was changed.
- **Usage in InternLink Hub**: Helps track modifications to objects like *Student, Application, Internship, Placement*, and ensures transparency in recruitment/placement workflows.
- **Access**:
    - Go to **Setup → View Setup Audit Trail**.
    - Download logs (up to 180 days history).

- **Benefit**: Provides visibility into configuration changes, ensuring data integrity and compliance.



InternLink Hub -"A Central Platform for Internships & Placements"
Phase: 10 – Final Documentation & Presentation

---

1. Project Overview

InternLink Hub is a Salesforce-based CRM platform designed to streamline internship and placement management for **students, recruiters, and placement officers**. The system provides a centralized hub where opportunities can be posted, applications tracked, and placements managed with transparency.

Through 10 structured phases, the project addressed real-world challenges such as **internship postings, application management, placement tracking, recruiter–student communication, and role-based access control**.

The solution was built on Salesforce with scalable features including **custom objects, flows, Apex triggers, validation rules, reports, and dashboards**.

The project delivered an end-to-end system where:

- **Students** can view internships, apply, and track their application/placement status.
- **Recruiters** can post opportunities, review applicants, and update statuses.
- **Placement Officers** can oversee the entire lifecycle, generate reports, and approve placements.

---

2. Problem Statement

Managing internships and placements in colleges and organizations is often manual, fragmented, and lacks transparency. The challenges include:

- Difficulty in managing student applications across multiple internships.
- Lack of real-time visibility into application and placement status.
- Communication gaps between recruiters, students, and placement officers.
- No centralized dashboards for data-driven decision-making.
- Absence of role-based access, leading to cluttered views for different users.

As a result, the process becomes time-consuming and inefficient.
InternLink Hub addresses this gap by providing a **centralized, automated Salesforce CRM solution** for internship and placement management.

---

3. Solution Overview

The InternLink Hub project was implemented in Salesforce with a focus on **role-based views and automation**. Key features delivered include:

- **Custom Objects** for Students, Companies, Internships, Applications, and Placements.

- **Tabs** configured for easy access based on roles.
- **Flows** for automated placement approval and application submission.
- **Apex Classes/Triggers** for handling backend logic and automation.
- **Validation Rules** for data consistency (e.g., phone numbers with +91 format).
- **Reports & Dashboards** for real-time tracking of applications and placements.
- **Lightning App (InternLink Hub)** combining all functionalities into a single workspace.

---

4. Project Phases & Deliverables

Phase 1: Problem Understanding & Industry Analysis

- Identified issues in manual internship and placement tracking.
- Defined need for automation, transparency, and dashboards.
- Success metrics defined: role-based access, reduced manual work, improved reporting.

Phase 2: Org Setup & Configuration

- Salesforce Developer Org created.
- Defined Roles: Placement Officer, Recruiter, Student.
- Created Profiles with specific object and tab permissions.
- Security Model configured (OWD, FLS, Permission Sets).

Phase 3: Data Modeling & Relationships

- Created custom objects:
    - **Student__c** → Personal details, skills, resume.
    - **Company__c** → Company info and contact details.
    - **Internship__c** → Internship details and requirements.
    - **Application__c** → Links student to internship with status.
    - **Placement__c** → Final placement record with package.
- Relationships defined (Student ↔ Application ↔ Internship ↔ Company).
- Page Layouts, Record Types, and Compact Layouts configured.

Phase 4: Process Automation

- Flows built for:
    - **Placement Officer Flow** → when a student is selected.
    - **Recruiter Flow** → when an application is submitted.

- Validation Rule: Phone number must include **+91** and 12 digits.

Phase 5: Apex Development

- Apex Classes for custom logic supporting flows.
- Triggers for automated status updates and notifications.
- Test Classes developed (>75% code coverage).

Phase 6: Lightning Web Components (LWCs)

- Not implemented. Used **standard Lightning components** for layouts and dashboards.

Phase 7: Integration

- Not implemented. Future scope to integrate with **Job Portals or HR Systems**.

Phase 8: Data Management & Deployment

- Created sample student, company, internship, and application records.
- Used Data Loader for bulk import/export.
- Simple deployment in Developer Org.

Phase 9: Reporting & Dashboards

- Reports created for:
    - Student details.
    - Internship postings by company.
    - Application statuses.
    - Placement statistics.
- Dashboards created (e.g., **Student Progress Dashboard**) to visualize end-to-end lifecycle.
- Dynamic Dashboards enabled for role-based visibility.

Phase 10: Final Presentation & Wrap-Up

- Consolidated documentation and demo video created.
- Showcased views for System Admin, Placement Officer, Recruiter, and Student.
- Demonstrated reports, dashboards, flows, and automation.

---

5. Security & Compliance

- Data Security: Roles, Profiles, OWD, and FLS enforced.
- GDPR compliance maintained (data usage limited to demo data).
- Audit Trail enabled for activity tracking.

---

6. Project Outcomes

- Delivered **InternLink Hub Lightning App** with 5 custom objects and role-based tabs.
- Automated key processes with **flows and Apex triggers**.
- Improved **data accuracy** with validation rules.
- Built **reports and dashboards** for transparency in applications and placements.
- Established a complete system for **students, recruiters, and placement officers** to collaborate.

---

7. Future Enhancements

- **Einstein Analytics** for predictive placement trends.
- **Student Portal LWC** for a more interactive interface.
- **Integration with external job portals** for real-world internship data.
- **Mobile App extension** for students and recruiters.

---

8. Conclusion

The InternLink Hub project successfully streamlined the internship and placement process using Salesforce. By combining **custom objects, automation, Apex, validation, reports, dashboards, and role-based access**, the system improved transparency, reduced manual work, and enhanced decision-making.

Final Status: Project Completed (Phase 1–10)
Deliverables: Documentation, Apex Code, Validation Rules, Reports, Dashboards, Lightning App, and Demo Presentation