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

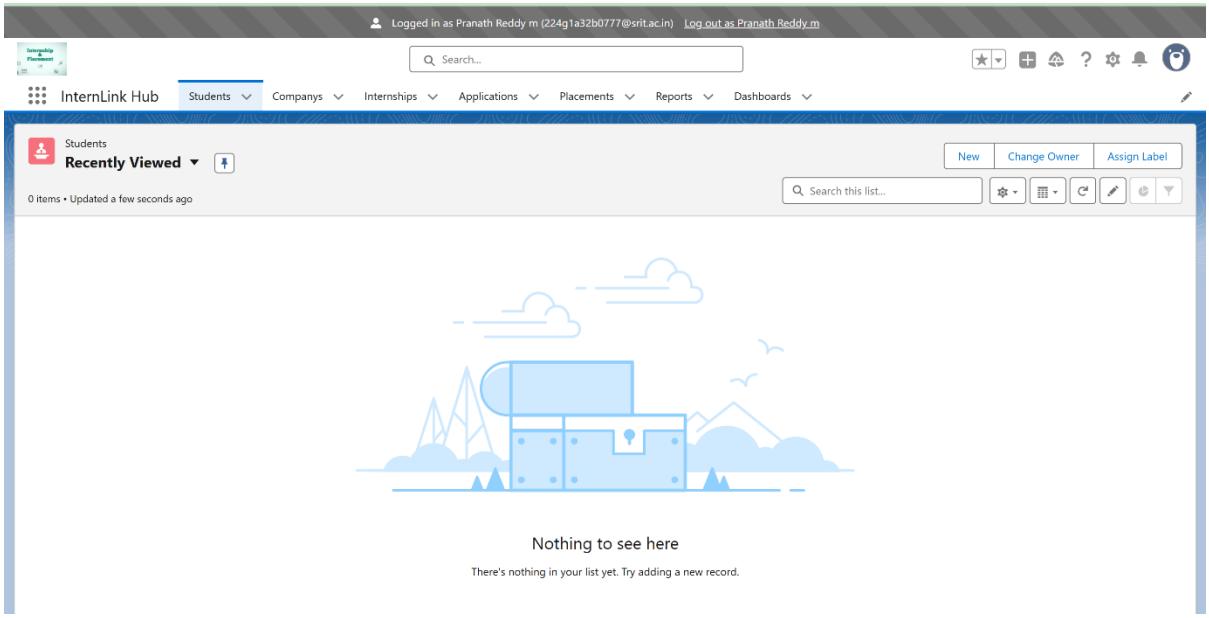
Phase 6: User Interface Development — InternLink Hub CRM

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)

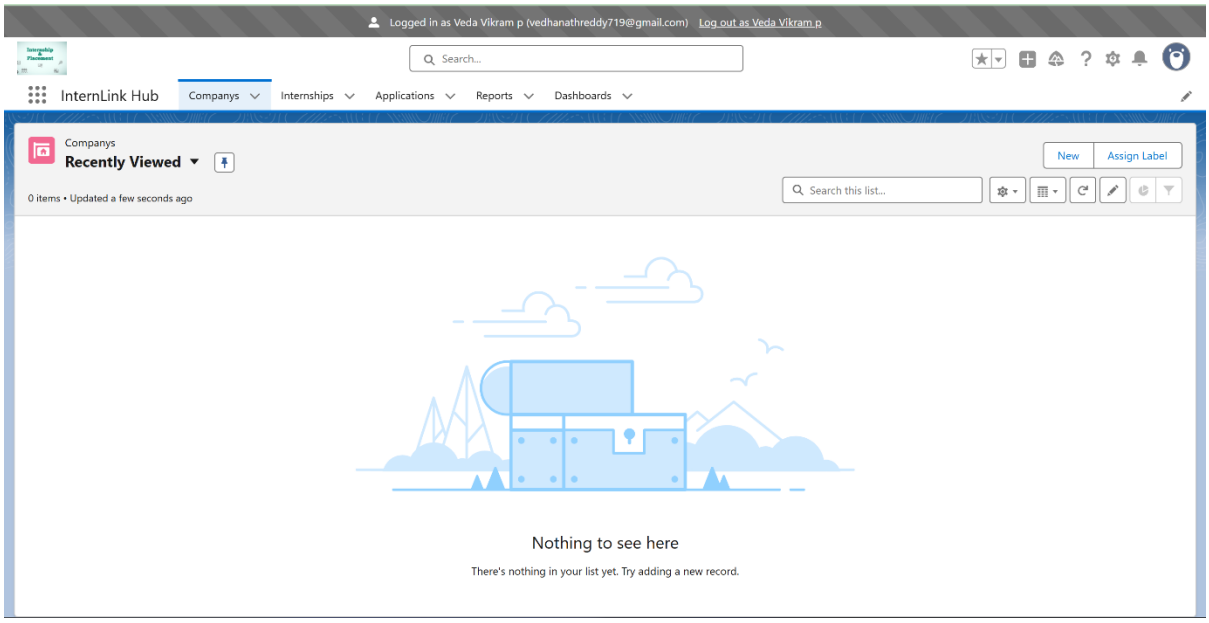


Fig 1.2 Recruiter view(can only access company,Internship,Application)

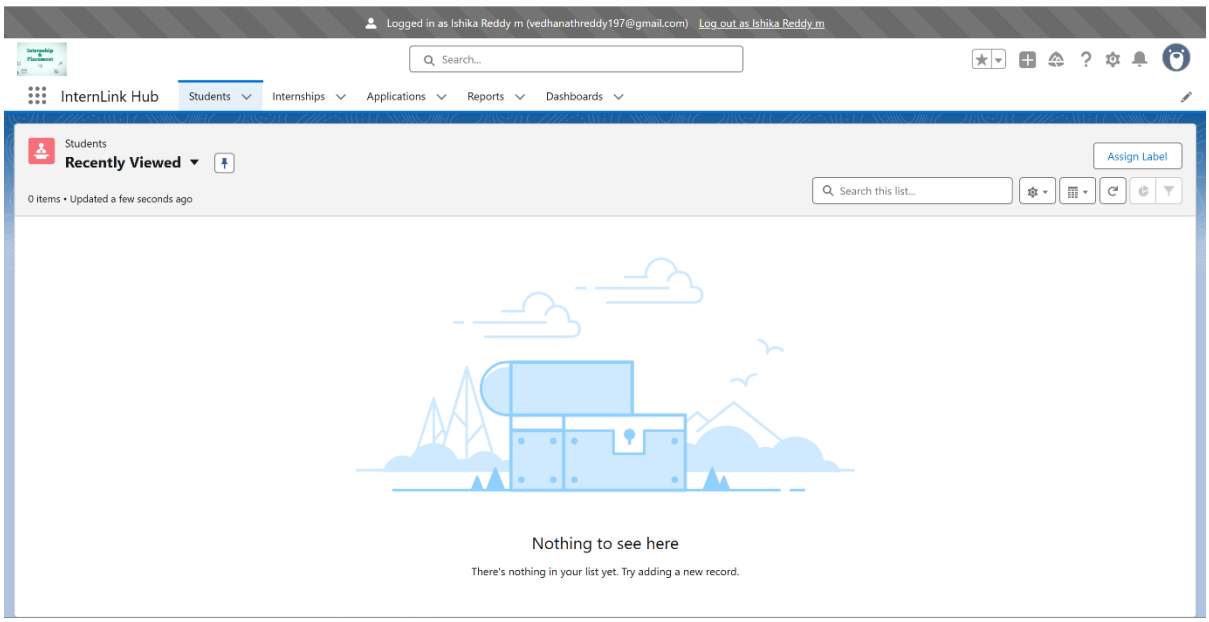


Fig 1.3 Student View(can only access student,internship,application)

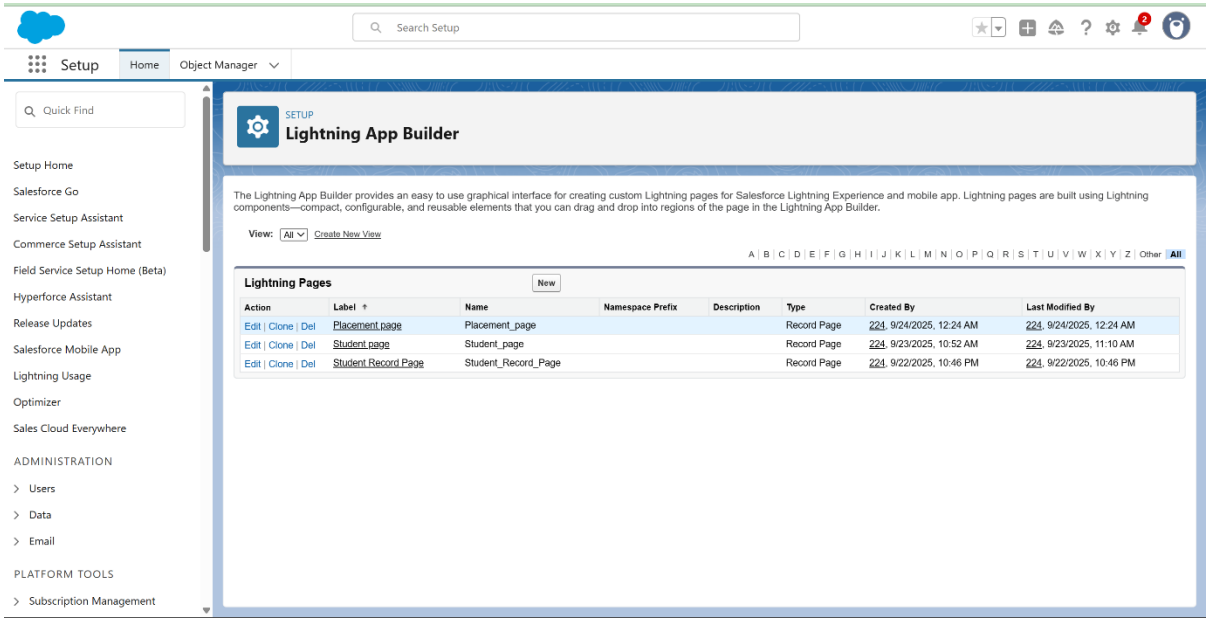
---

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.

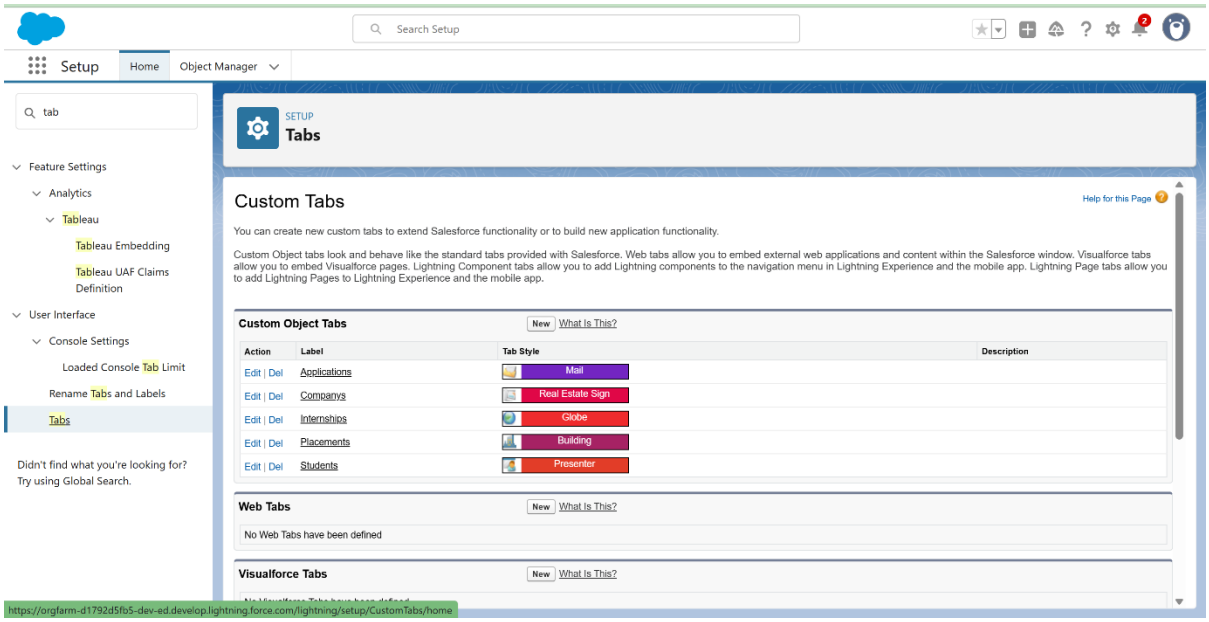
- **Benefit:** Improves efficiency for recruiters and admins by presenting relevant data clearly.



---

3. Tabs

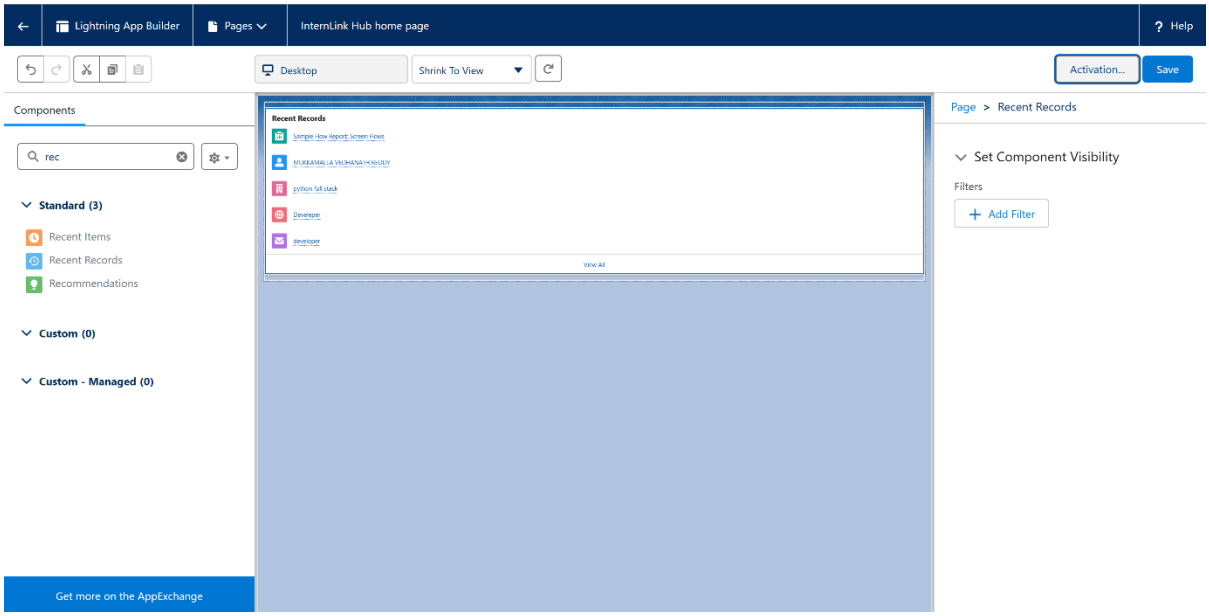
- **Purpose:** Organize CRM data access logically.
- **Implementation:**
  - Created custom tabs for Students, Applications, Placements, and Internships.
  - Configured tab visibility for different profiles (Recruiter, Admin, Student).
- **Benefit:** Simplifies navigation and aligns with user workflows.



---

4. Home Page Layouts

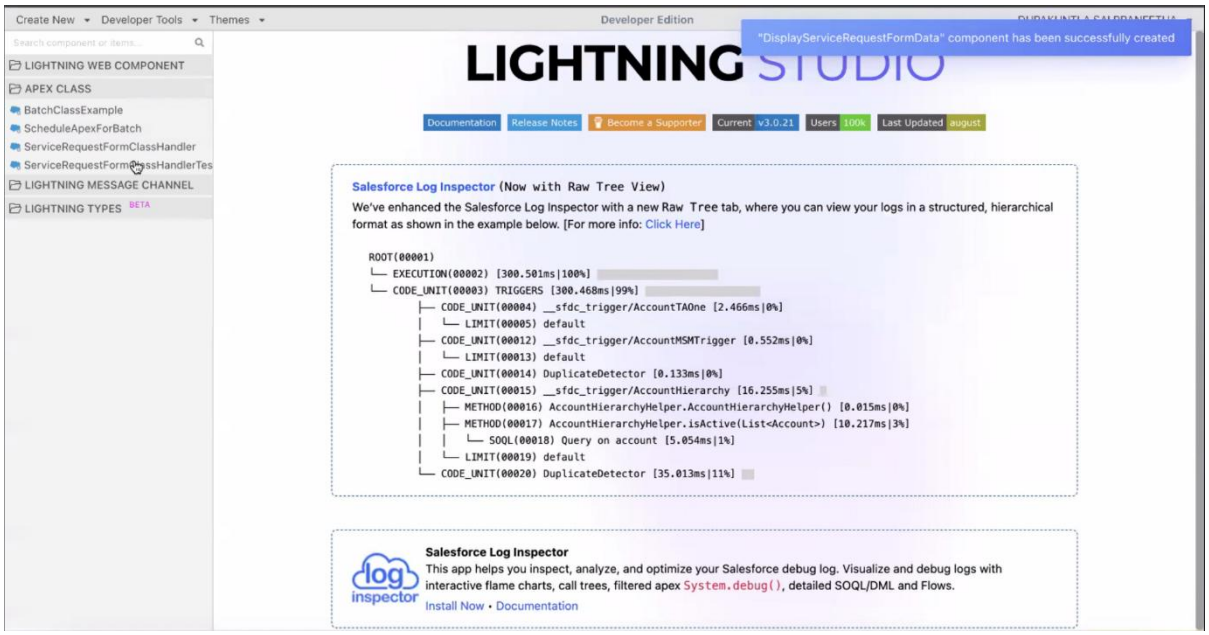
- **Purpose:** Provide a central dashboard for CRM users.
- **Implementation:**
  - Built a custom Home Page for InternLink Hub using Lightning App Builder.
  - Added components such as Recent Applications, Placement Status Charts, and Notifications.
- **Benefit:** Enhances productivity by displaying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