PROJECT TITLE

Smart Recruit

(Salesforce-powered Job Application Tracking System)

Phase 1: Problem Understanding & Industry Analysis

Requirement Gathering

Current Challenge

Recruiters and HR managers manually track applications using emails and spreadsheets. This results in:

- Difficulty managing high volumes of applications.
- Missed follow-ups due to lack of automation.
- Poor coordination between recruiters and HR managers.
- No real-time visibility for leadership into recruitment metrics.

Functional Requirements (What the system must do)

1. **Job Posting Management**
    a. HR/Recruiters should be able to create job postings with fields like:
        i. Job Title
        ii. Department
        iii. Location

        iv. Required Skills
        v. Application Deadline
    b. Recruiters should be able to update job postings when positions are filled.
2. **Candidate Management**
    a. Capture candidate information:
        i. Name
        ii. Contact Details
        iii. Resume (File Upload or URL)
        iv. Experience, Skills
    b. Maintain a history of applications per candidate.
3. **Application Tracking**
    a. Each application should move through lifecycle stages:
        i. Applied
        ii. Shortlisted
        iii. Interview Scheduled
        iv. Offered
        v. Hired / Rejected
    b. System should automatically update status when recruiters take action.
4. **Workflow Automation**
    a. **Flows/Process Builder**:
        i. Auto-create Interview record when an application is shortlisted.
        ii. Auto-send email/SMS notification to candidates at key stages.
    b. **Approval Process**:
        i. Offer letter stage must be approved by HR Manager.
5. **Notifications & Communication**
    a. Email alerts to candidates (status updates).
    b. Email notifications to recruiters (new application received).
    c. Reminder notifications for interviews.
6. **Reports & Dashboards**
    a. HR Managers should have dashboards showing:
        i. Applications by Job Posting.
        ii. Applications by Status (Funnel: Applied → Hired).
        iii. Recruiter Performance (applications handled, conversions).
    b. Recruiters should be able to run reports on their own candidates.

Non-Functional Requirements (System qualities)

1. **Usability**

a. Simple UI for recruiters to add/manage applications.
b. Lightning App with Tabs for Job Postings, Candidates, Applications, Interviews.

2. **Scalability**
   a. Should handle many applications without performance issues.
3. **Security**
   a. OWD: Applications private by default.
   b. Recruiters can only see applications they own.
   c. HR Managers can see all applications.
   d. Field-level security: sensitive fields (salary expectations, offer details) restricted.
4. **Reliability**
   a. Automated workflows should ensure no missed updates or communications.
5. **Extensibility (Future Scope)**
   a. Later, system can integrate with external portals like LinkedIn or Naukri.
   b. AI-based candidate ranking could be added.

Stakeholder Analysis

Stakeholder	Role		Needs / Expectations
HR Manager	Oversees recruitment process		- Access to recruitment KPIs via dashboards - Approval workflows for job offers - Compliance tracking
Recruiter	Manages job postings & candidates		- Simple interface for job posting & application tracking - Automated status updates - Email notifications for new application
Candidate	Applies for jobs		- Timely updates on application status - Transparent & fair process - Smooth communication

Management / Leadership	Reviews hiring metrics		- High-level dashboards & reports - KPIs like time-to-hire, offer acceptance rate - Data-driven insights for strategic decisions

Business Process Mapping

Current Process (Manual System)

Recruitment is handled mostly through offline tools like emails, spreadsheets, and job portals. This results in inefficiencies, delays, and lack of visibility:

- **Job Posting Creation** → Recruiters prepare job postings in Excel or upload them to third-party job portals, with no centralized control
- **Candidate Application** → Applications arrive via personal or shared HR emails. Resumes are stored manually in folders, often leading to mismanagement.
- **Status Tracking** → Recruiters track candidate progress (Applied, Shortlisted, Interviewed, etc.) using spreadsheets. This is error-prone and not transparent.
- **Communication** → Recruiters send emails manually to update candidates, leading to inconsistent or delayed responses.
- **Leadership Monitoring** → Management has no real-time insights into recruitment KPIs like time-to-hire, offer acceptance, or pipeline health.

Limitations:

- Manual handling increases chances of errors.
- Delayed candidate communication impacts candidate experience.
- No centralized system → duplication of data.
- Lack of dashboards → leadership cannot take data-driven decisions.

Proposed Process (Salesforce-Powered ATS)

By leveraging Salesforce, the recruitment lifecycle becomes automated, transparent, and trackable in real time:

- **Job Posting Creation** → Recruiters create structured Job Posting records in Salesforce, specifying department, role, skills, and deadlines.

- **Candidate Application** → When a candidate applies, their details and resume are captured in a Candidate object, automatically linked to a Job Posting.

- **Application Lifecycle** → Applications move through defined stages (*Applied* → *Shortlisted* → *Interview* → *Offer* → *Hired/Rejected*).
    - When an application is **Shortlisted**, a **Flow** auto-creates an Interview record and sends an email notification to the candidate.
    - When status = **Offer**, an **Approval Process** routes the request to the HR Manager for final confirmation.

- **HR Manager Approval** → If approved, the application status updates to **Hired** and dashboards refresh automatically.

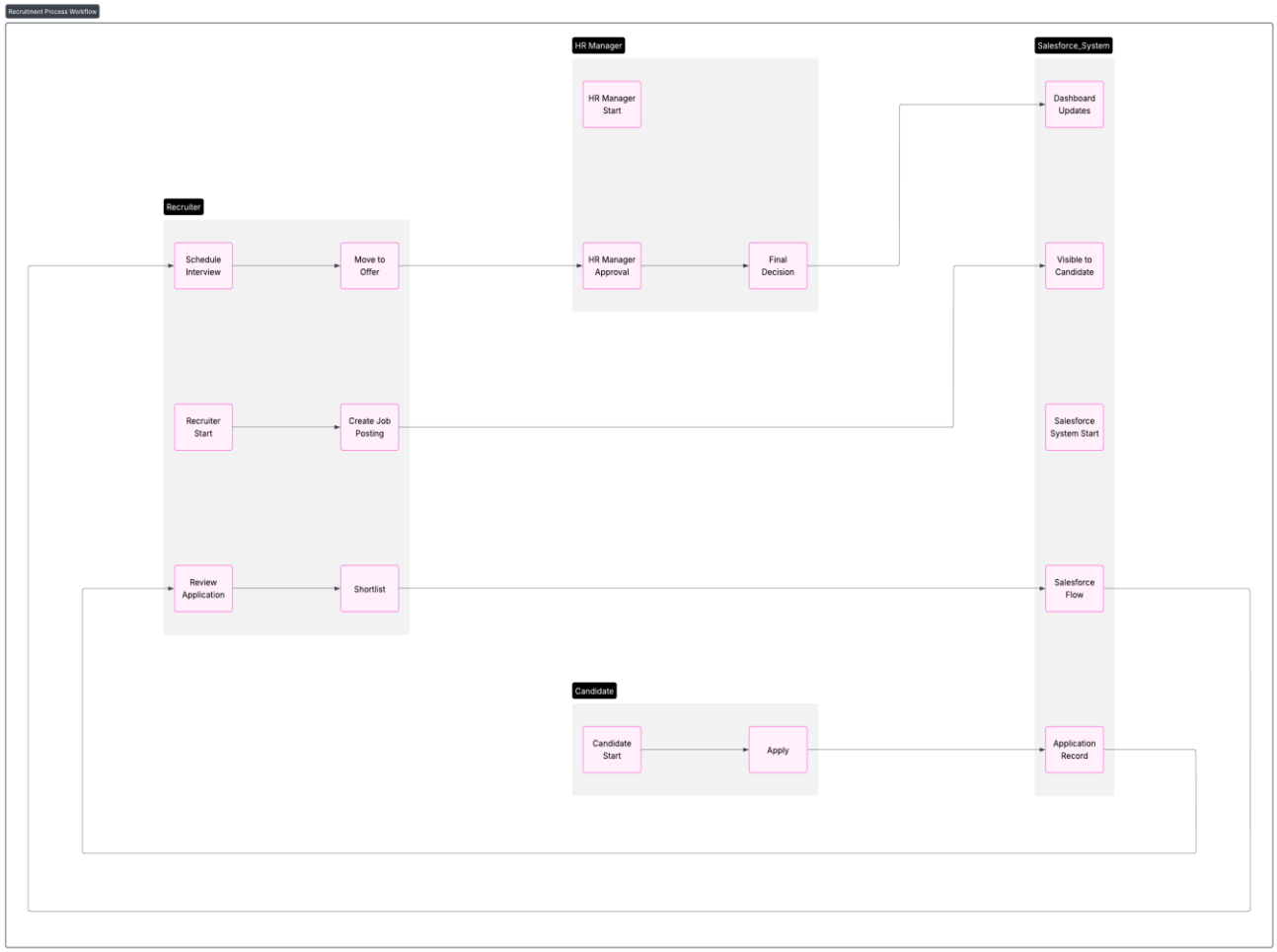
- **Automated Notifications** → Email alerts keep candidates informed at every stage.

- **Leadership Monitoring** → Dashboards and reports give real-time insights into hiring pipeline, recruiter performance, and bottlenecks.

Advantages:

- Centralized data storage → all records in Salesforce.
- Real-time communication with candidates → improves candidate experience.
- Automated workflows reduce manual effort.
- Dashboards provide actionable insights for HR and leadership.
- Scalable → system can handle hundreds of job applications simultaneously.

Workflow Diagram



Industry-Specific Use Case Analysis for Smart Recruit

1. IT Services & Consulting (TCS, Infosys, Wipro)

- **Challenge**: Thousands of applicants in campus drives, manual shortlisting, multiple interview rounds.

- **How Smart Recruit Helps**:
  - Auto-assign candidates to recruiters based on location/skill.

- Approval workflows for offer letters.
- Real-time dashboards of hiring funnel across multiple campuses.

2. Healthcare Industry (Hospitals & Pharma)

- **Challenge**: High demand for skilled nurses, doctors, pharmacists; manual hiring delays impact patient care.
- **How Smart Recruit Helps**:
  - Track applicant licenses & certifications as part of candidate records.
  - Automate scheduling of interviews with department heads.
  - Approval workflows for onboarding sensitive roles (like surgeons).

3. Retail & E-commerce (Amazon, Flipkart, Reliance Retail)

- **Challenge**: Seasonal hiring surges (festive seasons) → thousands of temporary staff applications.
- **How Smart Recruit Helps**:
  - Bulk import candidate applications from job portals.
  - Auto-screen candidates based on availability/shift preference.
  - Dashboards for HR to track store-wise hiring progress.

4. Banking & Financial Services (HDFC, ICICI, Deloitte)

- **Challenge**: Strict compliance; need to hire employees with verified backgrounds.
- **How Smart Recruit Helps**:
  - Track application → background verification → final approval.
  - Automate alerts for missing compliance documents (PAN, Aadhaar, KYC).
  - Dashboards for branch-wise recruitment stats.

5. Manufacturing & Logistics (Tata Motors, DHL, Mahindra)

- **Challenge**: Large blue-collar workforce recruitment, distributed across multiple plants/warehouses.
- **How Smart Recruit Helps**:
  - Region-wise recruiter assignment.
  - Candidate mobile app → easy application process for workers.
  - SMS/email alerts for interview scheduling.

6. Education (Universities & EdTech like Byju's, Coursera)

- **Challenge**: Hiring large teaching/administrative staff during academic sessions.
- **How Smart Recruit Helps**:
  - Manage separate pipelines for faculty vs administrative roles.
  - Approval workflows with academic deans for faculty selection.
  - Reports on hiring time per department.

Phase 2: Org Setup & Configuration – Smart Recruiter ATS

In this phase, the Salesforce environment for the Smart Recruiter Applicant Tracking System (ATS) was prepared and configured. The objective was to establish a secure, well-structured, and scalable foundation before implementing business processes. This setup ensures that organizational details, user management, and security controls are aligned with the recruitment workflow.

Salesforce Edition

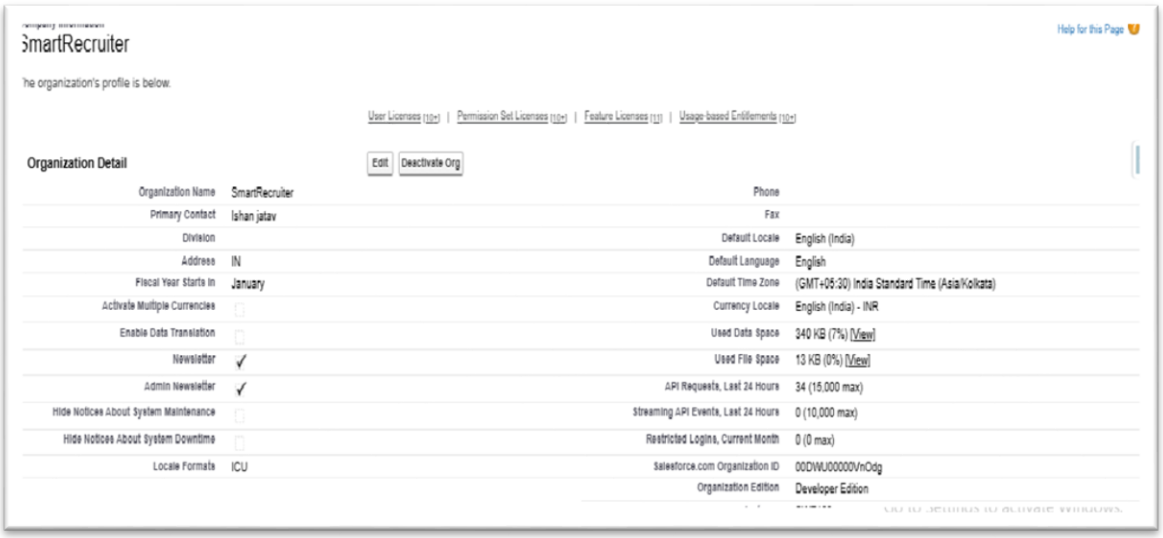
The project was developed using a **Salesforce Developer Edition Org**, which provides access to core features like Apex, automation tools, custom objects, and AppExchange apps. Although storage and user limits are restricted, this edition is suitable for proof-of-concept and academic projects.

Company Profile Setup

The company profile was configured with the following details:

- **Company Name**: Smart Recruiter
- **Default Locale**: English (India)
- **Currency**: INR
- **Time Zone**: Asia/Kolkata

This ensures that job postings, candidate data, and reports are aligned with the organization's region and currency standards.



Business Hours & Holidays

Business hours were defined as **Monday to Friday, 9:00 AM – 6:00 PM**, reflecting typical HR operations. Public holidays in India were added for demonstration purposes. These settings support time-based workflows, such as escalation rules for pending approvals.

Fiscal Year Settings

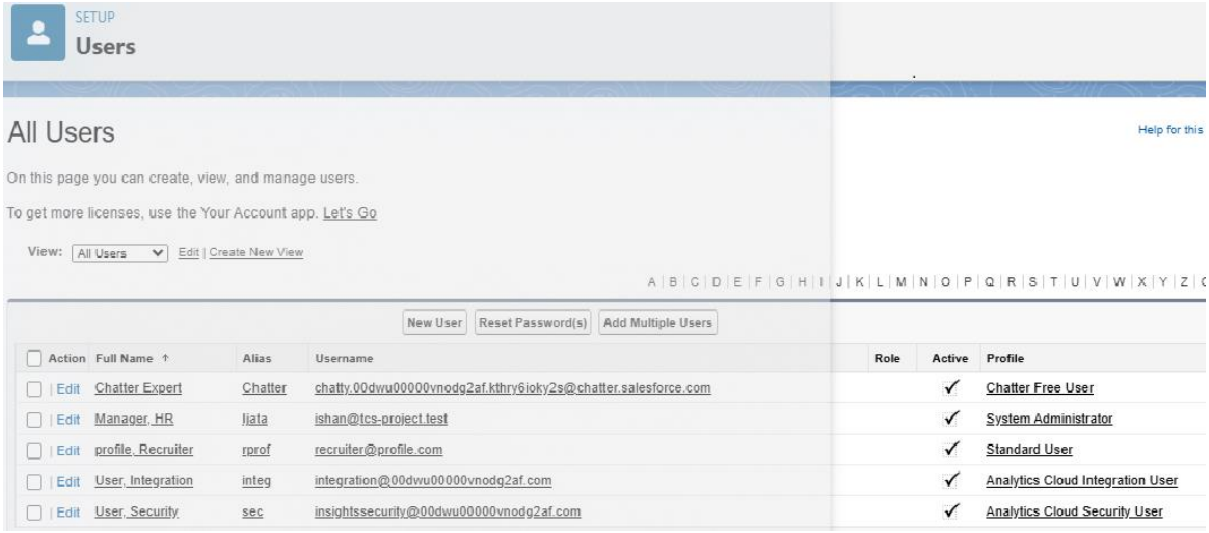
A **Standard Fiscal Year (April–March)** was enabled. This ensures that recruitment reports, such as hires per quarter or year, are synchronized with the organization's financial reporting cycle.

User Setup & Licenses

Sample users were created to represent real-world roles:

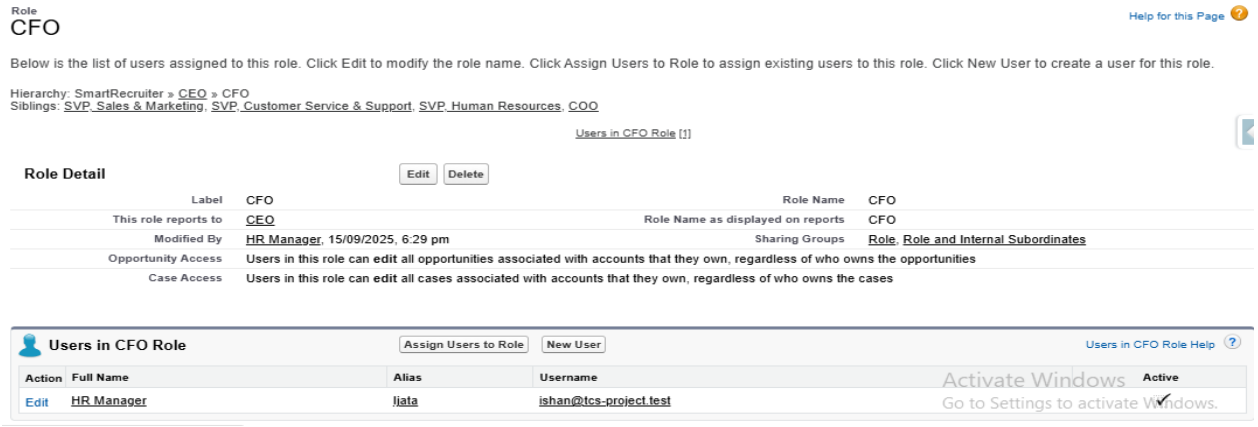
- **HR Manager** – manages approvals and oversees recruitment.
- **Recruiter** – manages job postings and candidate applications.
- **Candidate** – applies for jobs via portals or external submission.

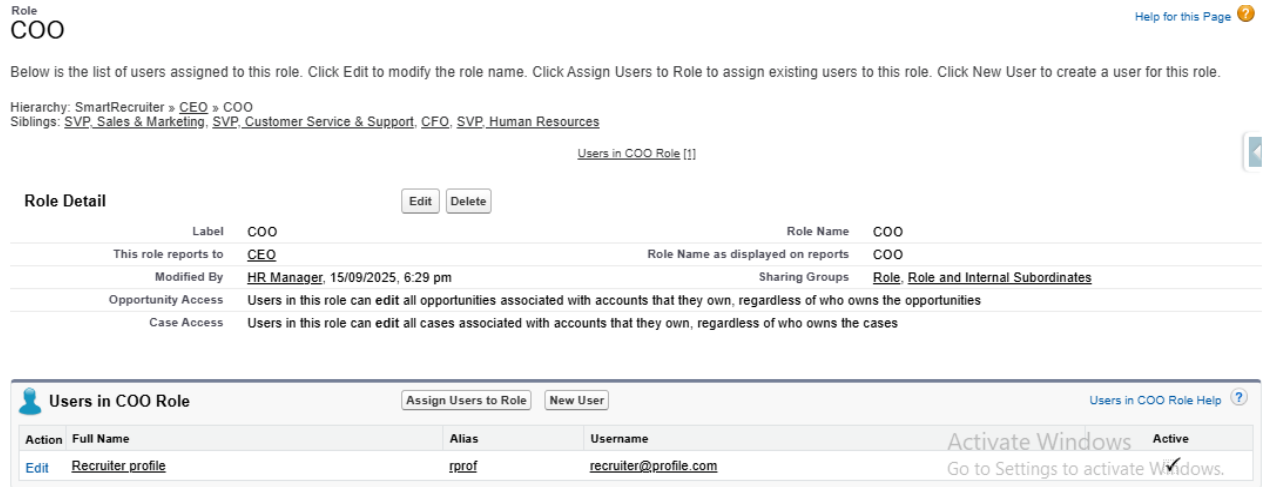
Each user was assigned appropriate licenses and profiles to simulate practical scenarios.



Profiles, Roles, and Permission Sets

Profiles were customized to control access to objects and fields. Roles were defined hierarchically: **HR Director → HR Manager → Recruiter**. Permission Sets were created for granting additional privileges, such as access to reporting features. This setup ensures a balance between security and operational flexibility.





Organization-Wide Defaults (OWD) and Sharing Rules

- **OWD** was set as follows:
  - Job Applications and Candidate records → Private
  - Job Postings → Public Read/Write
- **Sharing Rules** were implemented to allow recruiters from specific departments to collaborate on relevant applications.

This prevents unauthorized access to sensitive candidate data while enabling teamwork among HR staff.

Note -> I will be completing the OWD setup after creating my custom objects. And will establish sharing rules as per requirements.

Login Access Policies

Login restrictions were applied by IP ranges for administrators, while recruiters were granted trusted access for remote work. These measures strengthen system security.

Developer Org Setup & Sandbox Usage

The project was built on a **Developer Org**. For enterprise-level deployment, a sandbox strategy is recommended:

- **Developer Sandbox** → for building features.
- **UAT Sandbox** → for testing by HR staff.
- **Production Org** → for live usage.

Deployment Basics

Metadata and configurations were deployed using **Change Sets** and **Salesforce DX (SFDX) with VS Code**. A GitHub repository was also used for version control and collaboration, ensuring that project changes are tracked effectively.

Phase 3: Data Modelling & Relationships Report

1. Objective

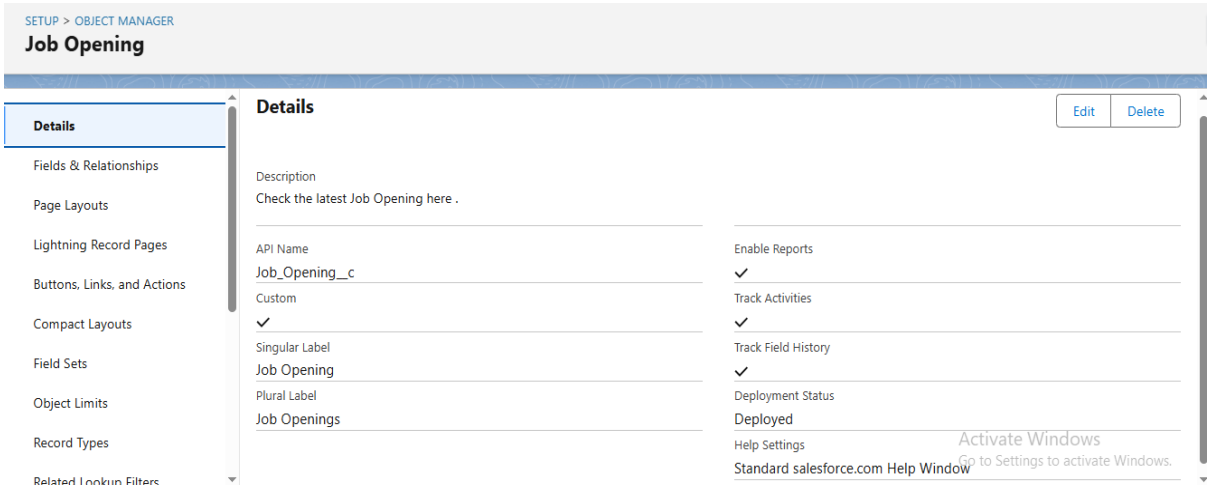
- To design the **data model** for Smart Recruiter that supports job openings, applications, interviews, and applicant tracking.

- To define relationships between objects, fields, page layouts, record types, and file handling.

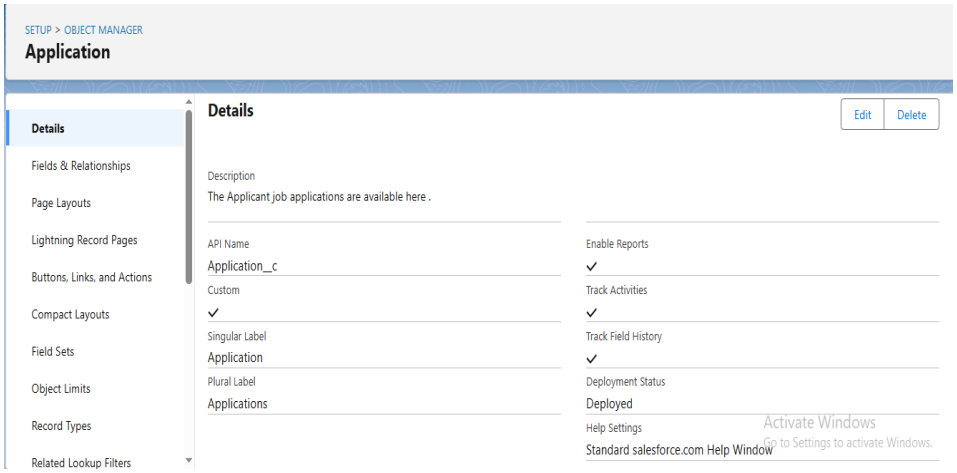
2. Objects Created

List all the objects in my project the Smart Recruiter with description of each .

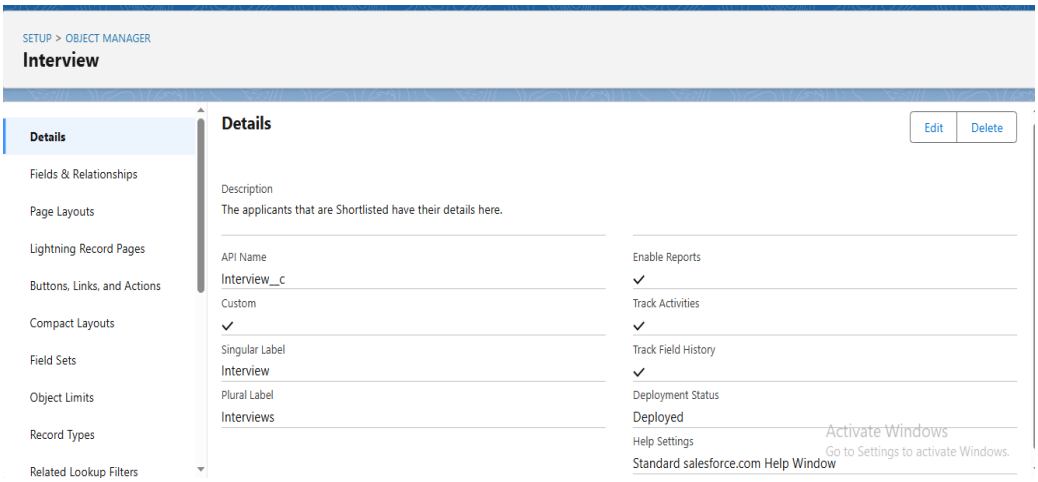
- **Job Opening** _c -> It is a custom object created for storing for the Job opening details from the different organisations.



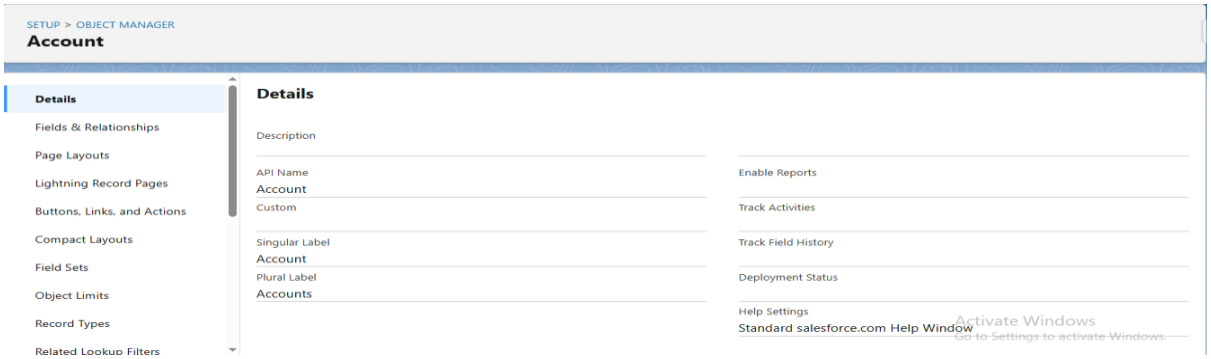
- **Application _c ->** this is also a custom object that is used to store the applicant records that applied for the job opening.



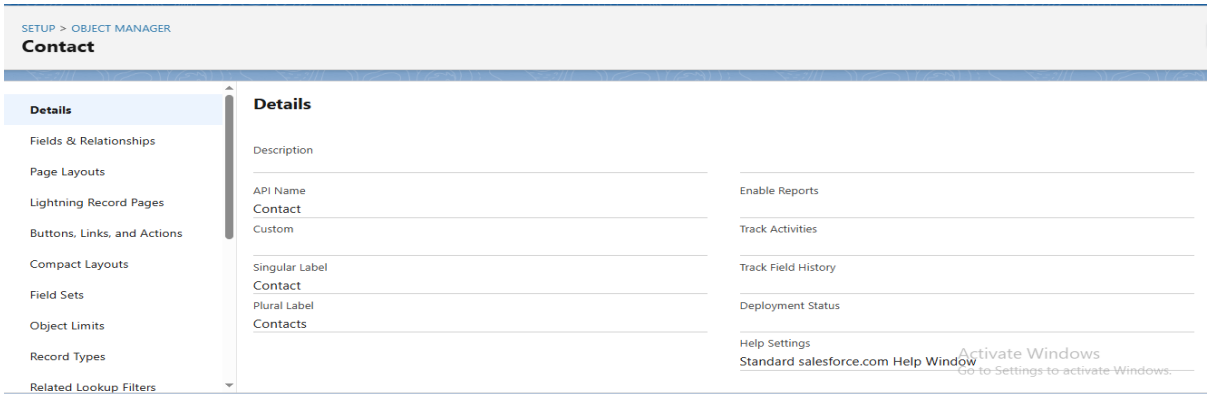
- Interview -> This is a custom object that is used for storing scheduled interview details and track details for applicant shortlisted for interview.



- Account -> This is a standard object that is used to store details of the organisations that has provided a job opening.

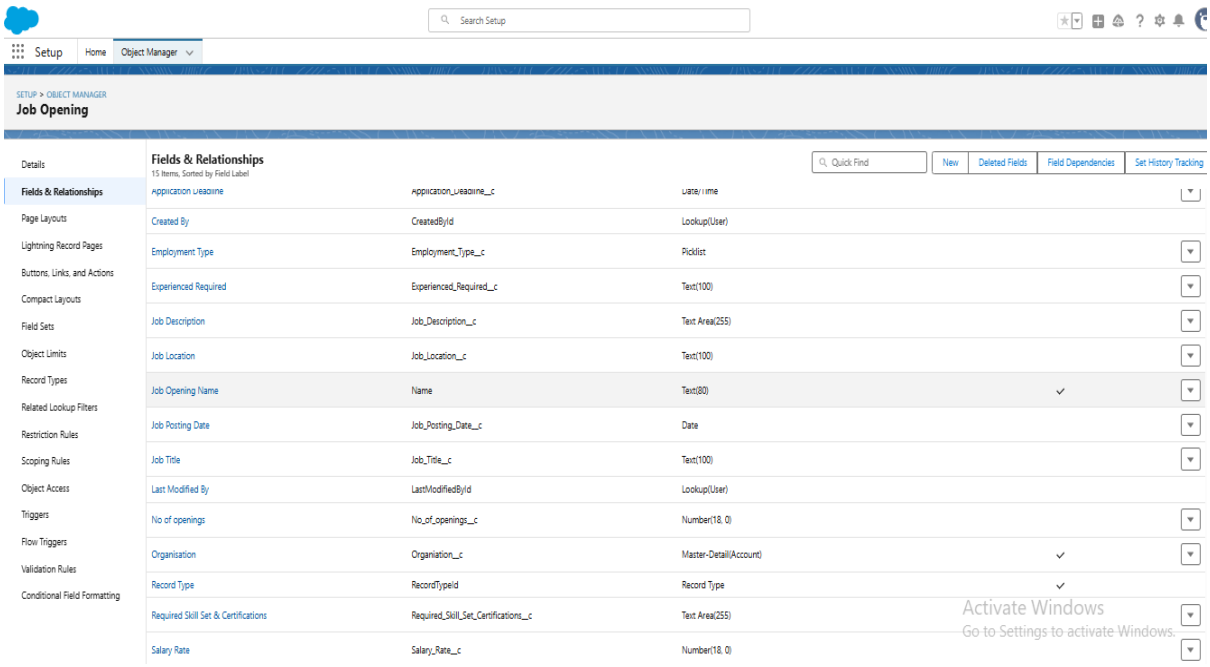


- **Contact ->** this is a standard object that is used for storing application contact details.

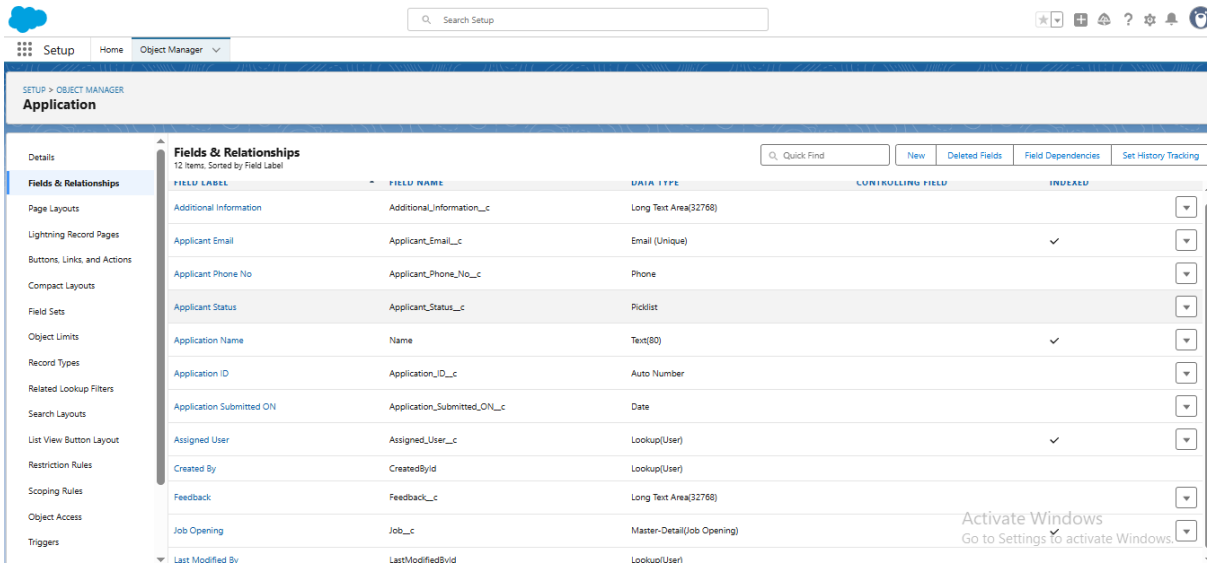


3. Fields

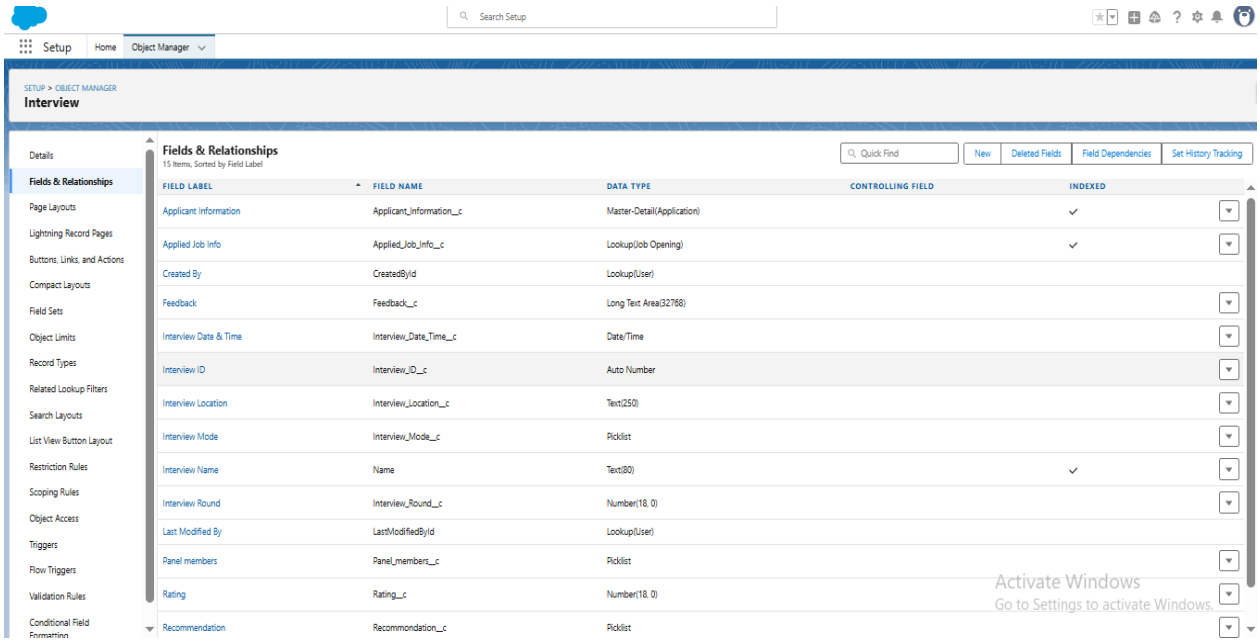
- Job Opening Objects fields are shown in the image below.



- Application Object Fields are shown in the image below.



- Interview Objects Fields are shown in the image below.

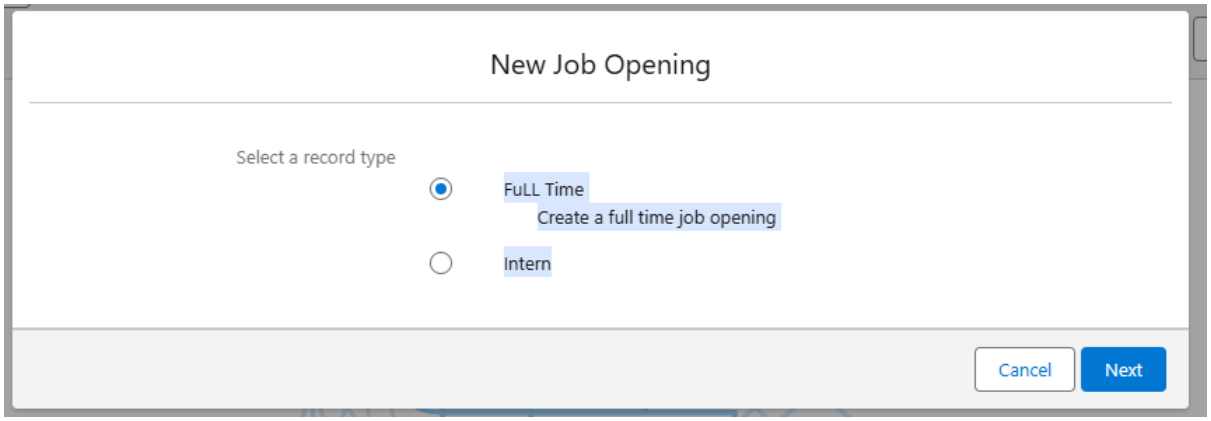


- Account and contacts are standard objects therefore their fields are present by default in the salesforce.

4. Record Types

- **Job Opening**:
    - Full-Time – A record type for storing full-time job opening details.

- Internship-A record type for storing Internship job opening details
- Each record type has its **own page layout** (Full-Time Layout, Internship Layout).

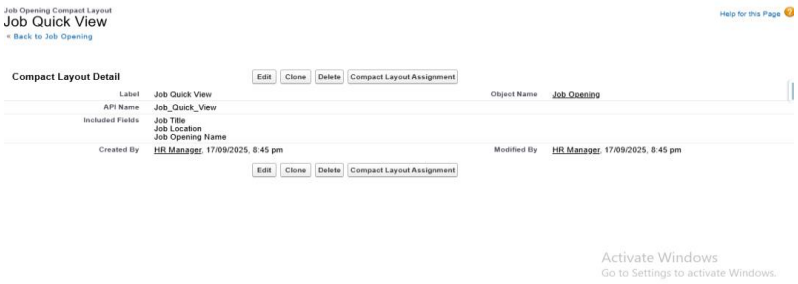


5. Page Layouts

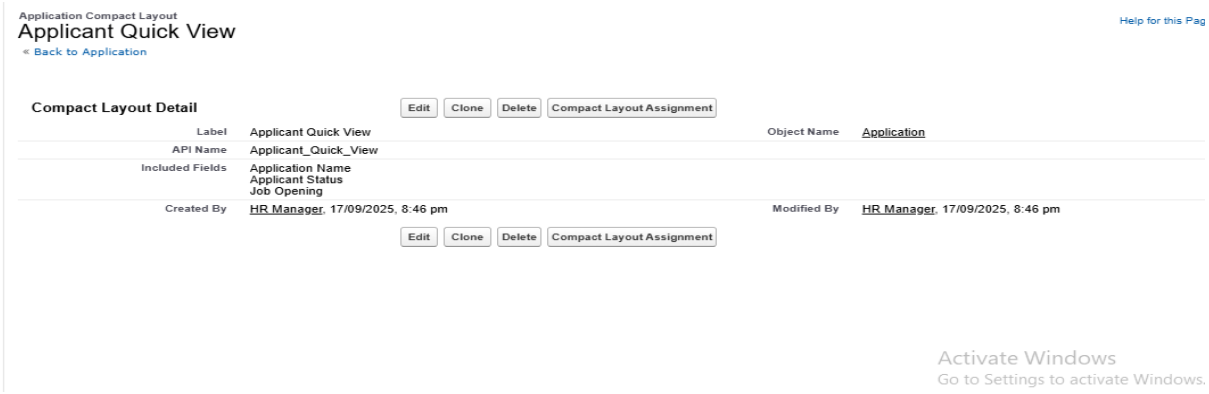
- **Job Opening**: Single layout showing details required for the Job Opening.
- **Application**: Single layout showing applicant info, resume, job applied, and status.
- **Interview** Layout shows interview details essential for scheduling an interview.

6. Compact Layouts

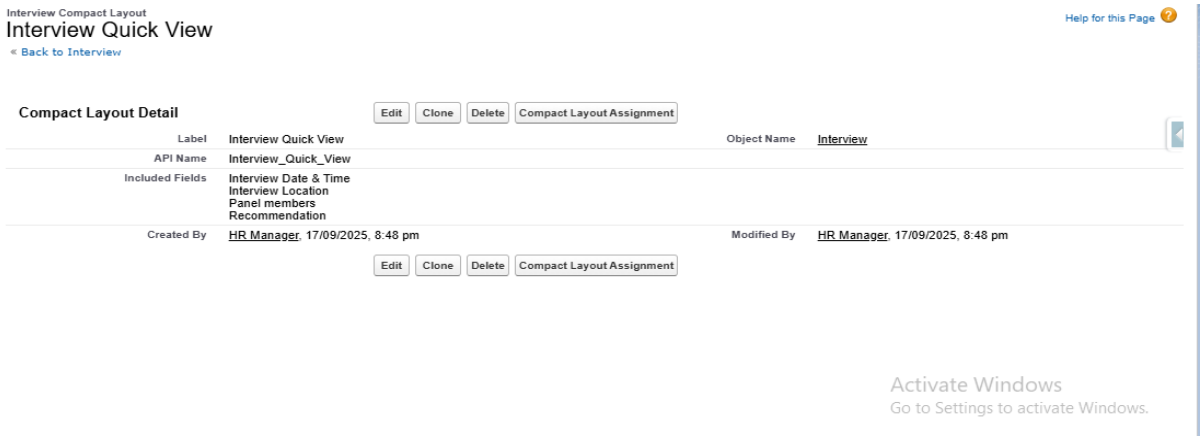
- **Job Opening Compact Layout**: Job Title, Location, Status.



- **Application Compact layout**: Applicaiton name, Application status etc.



- **Quick Interview Compact layout**: Interview Date & Time, Recommendation Interview Location



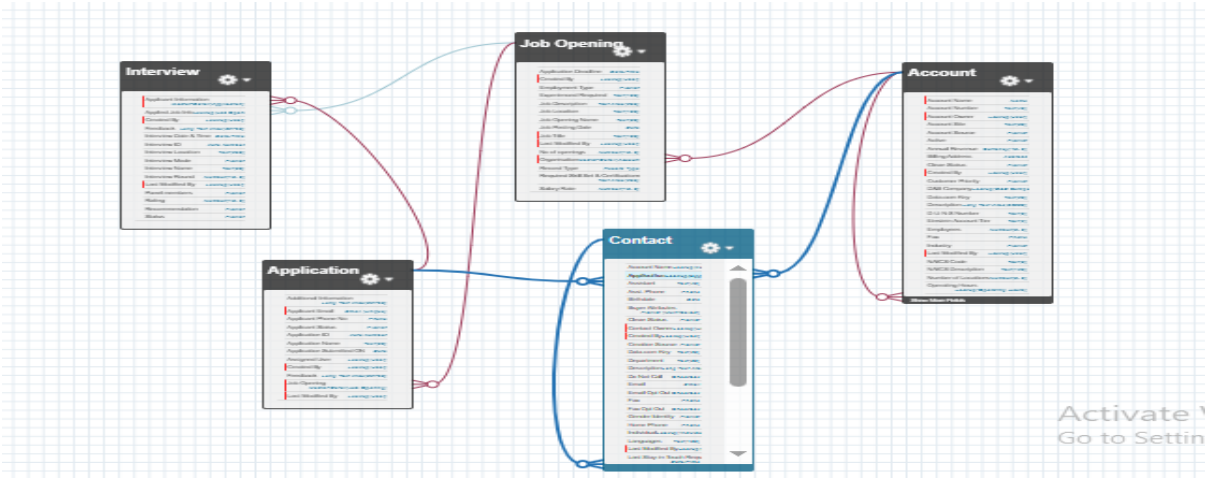
7. Object Relationships

Include **all relationships between all the objects used by me till now .**

Parent Object	Child Object	Relationship Type	Notes
Account	JobOpening_c	Master-Detail	Deleting Account deletes all Jobs
JobOpening_c	Application_c	Master-Detail	Applications tied to Job
Application_c	Interview_c	Master-Detail	Interviews tied to Application
Application_c	Contact	Lookup	Applicant can apply to multiple jobs

6. Schema Builder

- Use Schema Builder to **visualize relationships** between objects.
- Example: One **Job Opening _c** has many **Application__c** records.



8. Junction Objects

- Junction object is an object that is used for creating many to many relationships between two objects.
- I Have not used the Junction Object in my Project I may be showing it while working in the future scope.

9. External Objects

- Suppose Smart Recruiter wants to pull **job data from a third-party system (like LinkedIn/Indeed). I** Have Planned this idea as my future scope.