Phase 8: Data Management & Deployment

1. Data Import Wizard

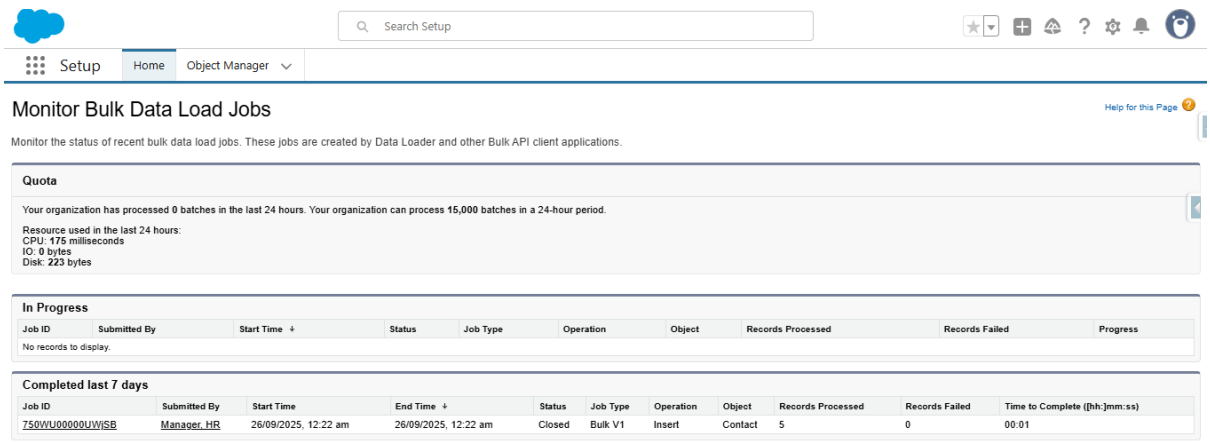
To populate sample data for testing and demonstration, the Salesforce **Data Import Wizard** was used for both standard and custom objects.

Steps Taken:

- Prepared CSV files for **Contacts** and **Job Postings** containing 5 records each.
- Uploaded CSVs via **Setup → Data Import Wizard → Launch Wizard**.
- Mapped CSV columns to Salesforce fields (e.g., FirstName → First Name, Location__c → Location).
- Imported successfully to create initial test data for the project.

Outcome:

- All sample Contacts and Job Postings are now available in Salesforce.
- Enabled creation of **Application** and **Interview** records for testing automation and flows.



2. Duplicate Rules

To prevent accidental duplicate records and ensure data integrity:

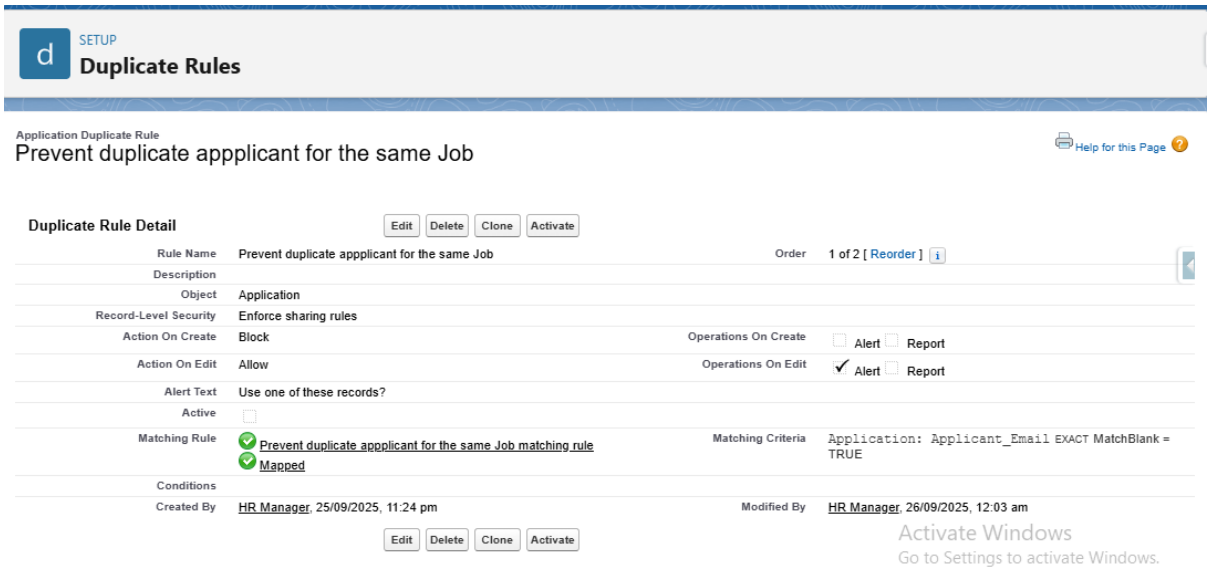
Implementation:

- Created **Matching Rules** on Contact object (Email equals).
- Created **Duplicate Rules** to **Alert** users if duplicate contacts are detected.
- Activated both rules.

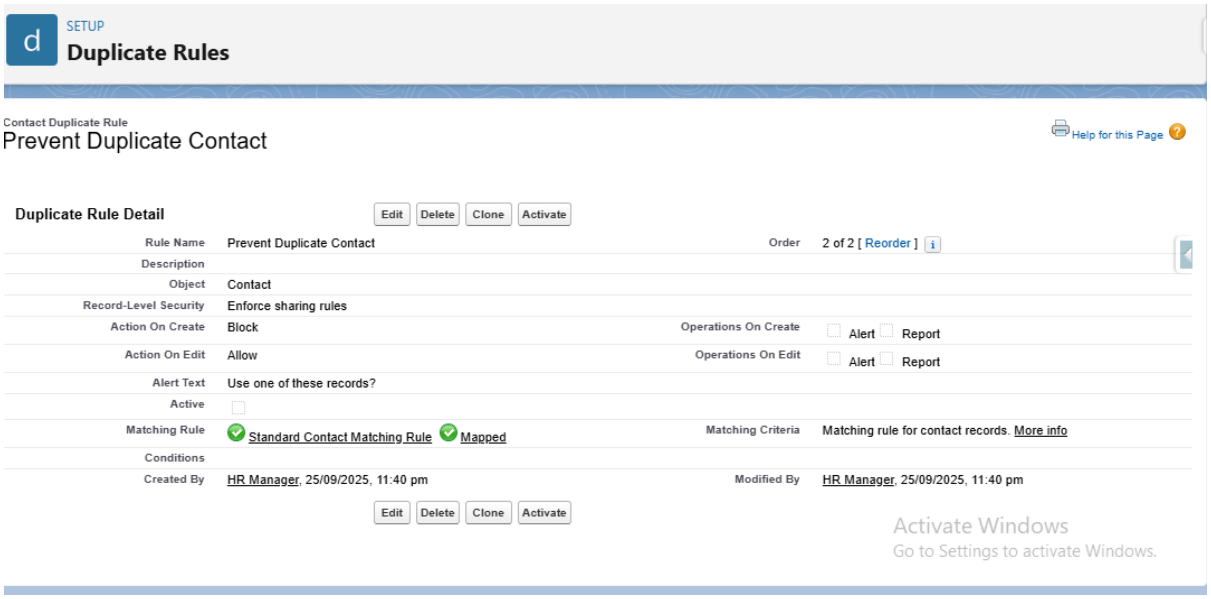
Outcome:

- Recruiters cannot create duplicate candidate records.
- Ensures clean and reliable test data for recruitment processes.

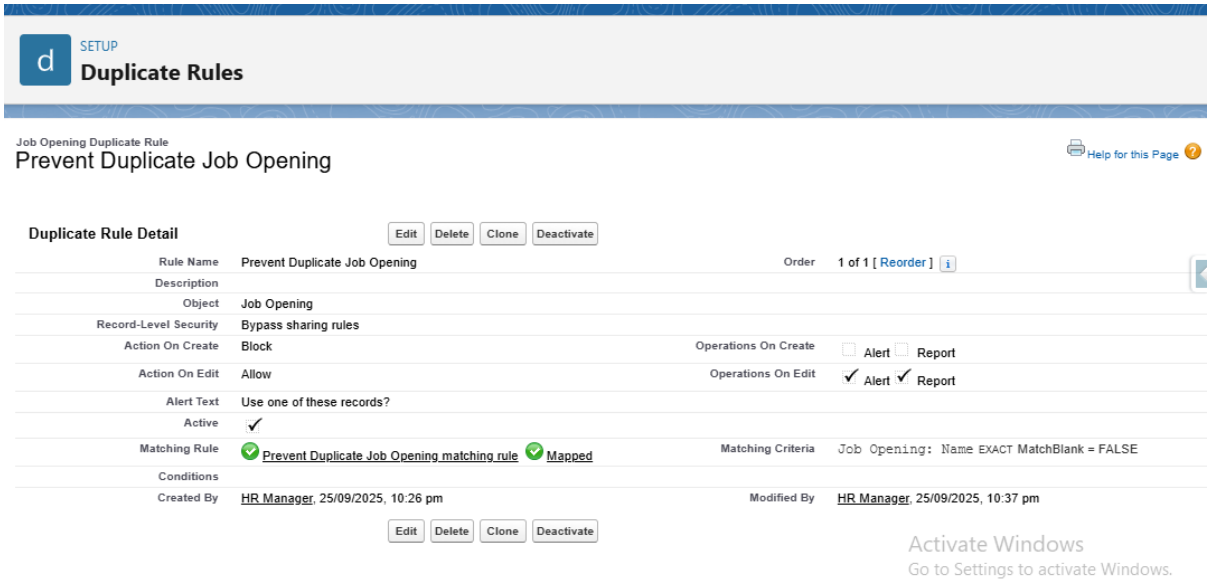
Prevent Duplicate Application For the Same Job



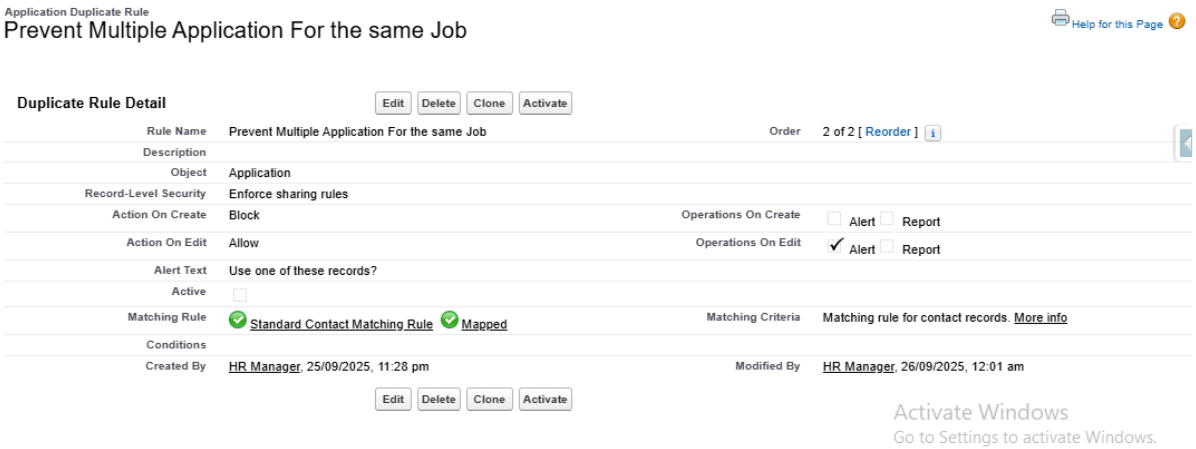
- Prevent Duplicate Contact



Prevent Duplicate Job Opening



Prevent Multiple Application for the same job



3. Data Loader

- Although the **Data Import Wizard** was sufficient for small datasets, **Data Loader** can be used for bulk insert, update, or export of records.
- In this project, small datasets were handled manually or via the Import Wizard due to time constraints.

4. Data Export & Backup

- Regular backups are critical for data safety.
- Salesforce **Data Export** feature can be used to export all records and metadata.
- For the project, manual export of Contacts, Job Postings, Applications, and Interviews was demonstrated.

- I Have Exported All the data in this Zip File.
- WE_00DWU00000VnOdg2AF_1.ZIP

6. VS Code & SFDX

- **VS Code** with Salesforce Extensions was used for development and deployment of:
    - Apex Classes & Triggers
    - Lightning Web Components

    - Metadata retrieval
- **SFDX Commands** help in pushing/pulling changes between local and org.

Outcome:

- Demonstrates modern Salesforce development practices.
- Enables easy tracking, version control, and deployment of all project components.
- I have imported all the data of salesforce in the vs code of connecting vs code to my org and pushed it in the project GitHub repository. This was Done by me in the previous phase only .

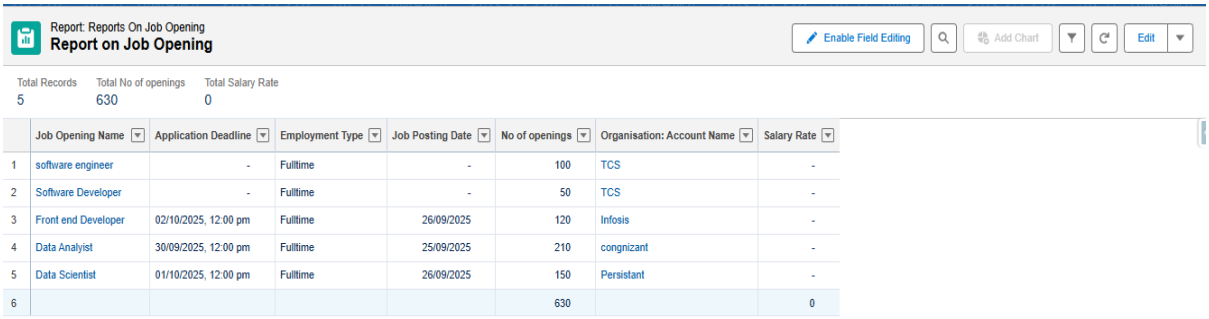
Phase 9: Reporting, Dashboards & Security Review

In this phase, the focus was on building insights, monitoring business processes, and ensuring data security within the Salesforce environment. The following tasks were implemented:

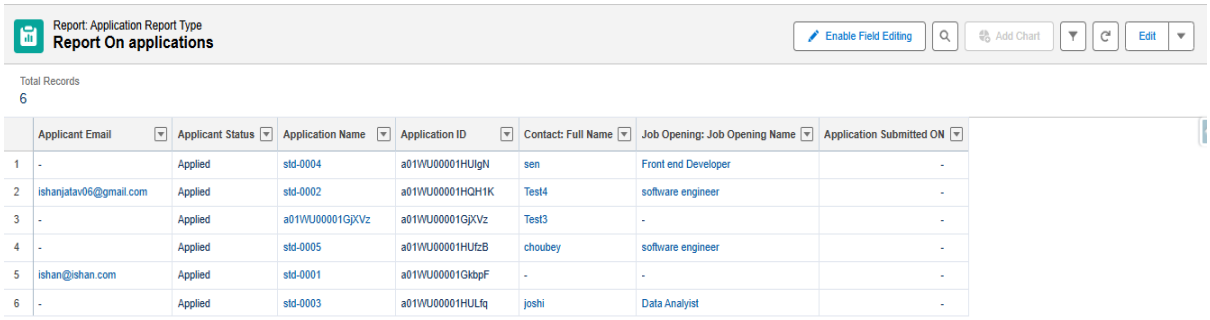
1. Reports

Different types of reports were created to analyse recruitment data and provide meaningful insights:

- **Tabular Reports:**
  Simple record listings, such as a list of all Candidates or Job Applications.
- Simpler report on the active job opening



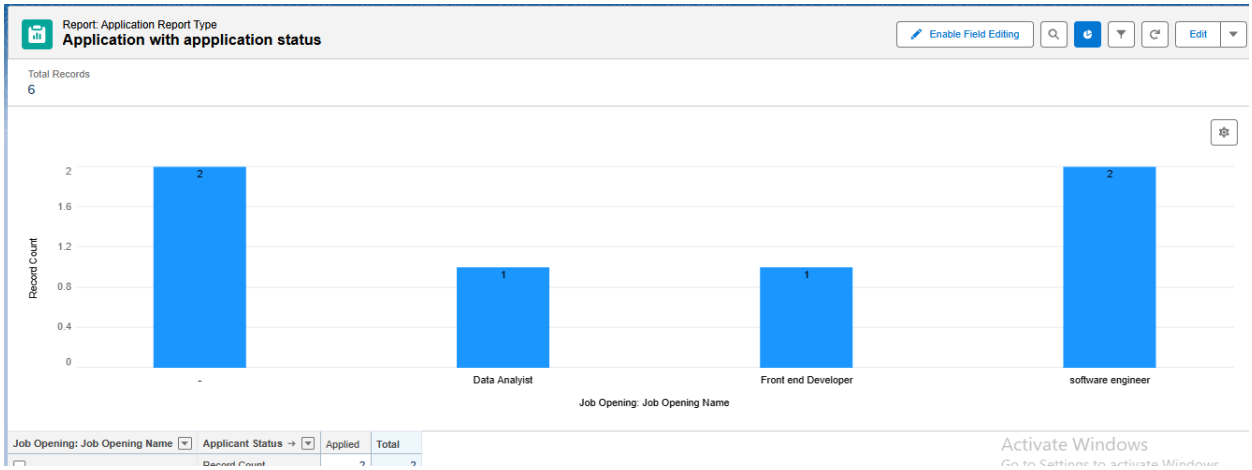
	Job Opening Name	Application Deadline	Employment Type	Job Posting Date	No of openings	Organisation: Account Name	Salary Rate
1	software engineer	-	Fulltime	-	100	TCS	-
2	Software Developer	-	Fulltime	-	50	TCS	-
3	Front end Developer	02/10/2025, 12:00 pm	Fulltime	26/09/2025	120	Infosis	-
4	Data Analyist	30/09/2025, 12:00 pm	Fulltime	25/09/2025	210	congnizant	-
5	Data Scientist	01/10/2025, 12:00 pm	Fulltime	26/09/2025	150	Persistant	-
6					630		0

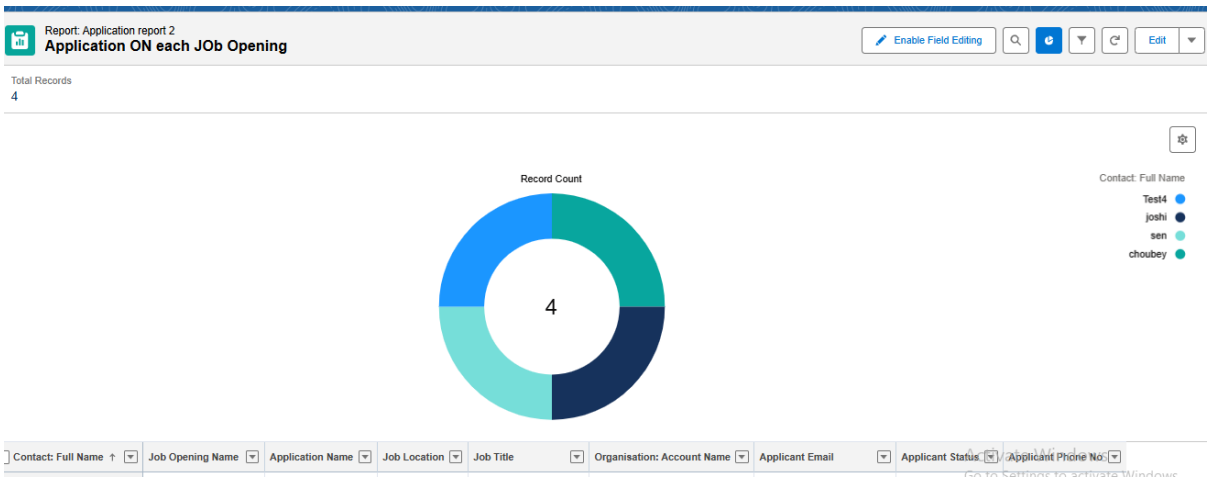


	Applicant Email	Applicant Status	Application Name	Application ID	Contact: Full Name	Job Opening: Job Opening Name	Application Submitted ON
1	-	Applied	std-0004	a01WU00001HUlgN	sen	Front end Developer	-
2	ishanjatav06@gmail.com	Applied	std-0002	a01WU00001HQH1K	Test4	software engineer	-
3	-	Applied	a01WU00001GjXVz	a01WU00001GjXVz	Test3	-	-
4	-	Applied	std-0005	a01WU00001HUfzB	choubey	software engineer	-
5	ishan@ishan.com	Applied	std-0001	a01WU00001GkbpF	-	-	-
6	-	Applied	std-0003	a01WU00001HULfq	joshi	Data Analyist	-

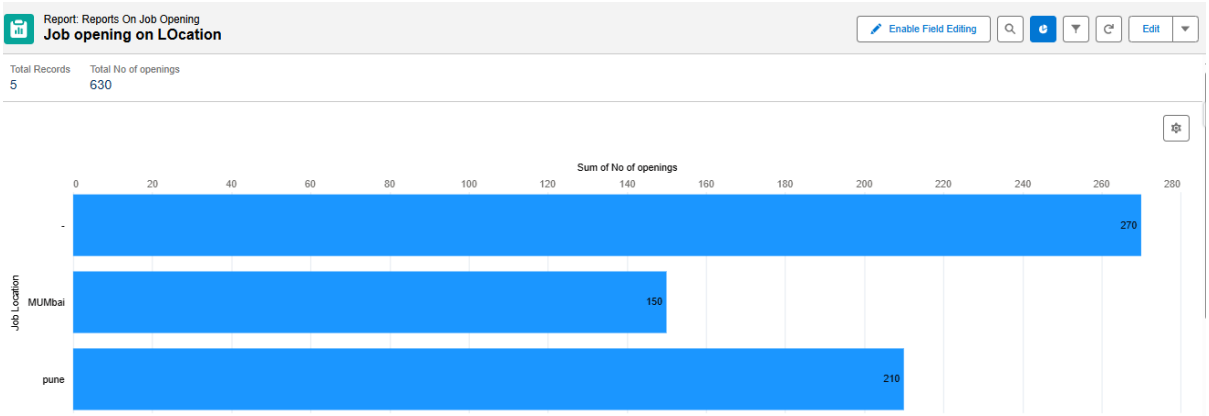
- **Summary Reports:**
  Applications grouped by **Job Posting** or **Status** to quickly visualize recruitment progress.

Application grouped by application Status





Applicaitons On each Job Opening

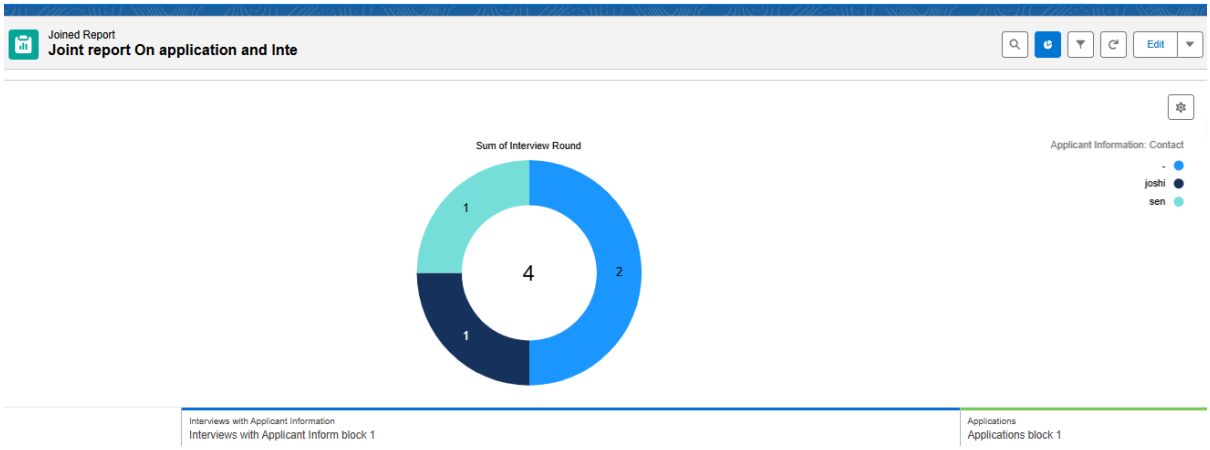


Job Opening Grouped by Location

- **Matrix Reports:**
  Applications summarized by **Recruiter vs. Status**, giving a two-dimensional view of workload distribution.

- **Joined Reports:**
  Applications and Interviews displayed together, helping track how many interviews were scheduled per application.

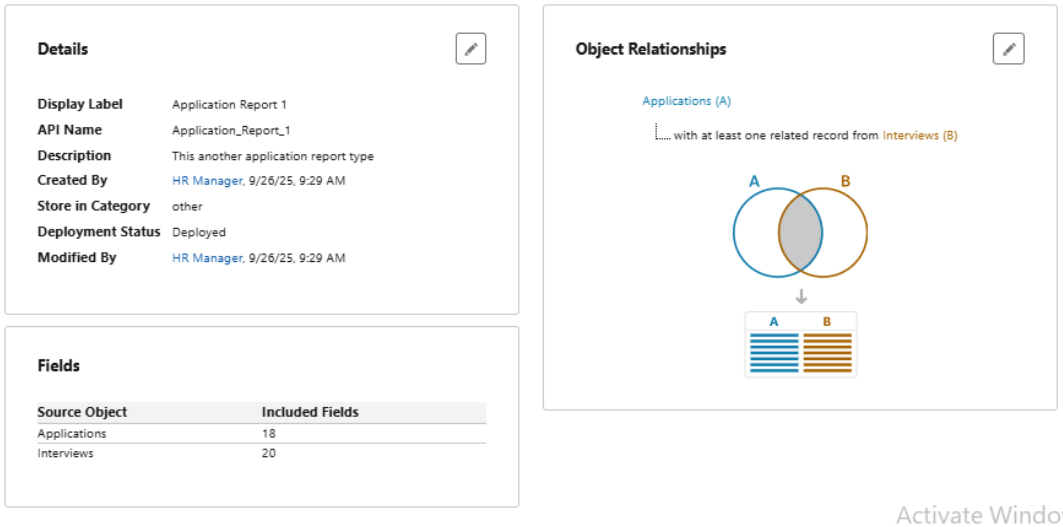
Joint Report on Application and Interview



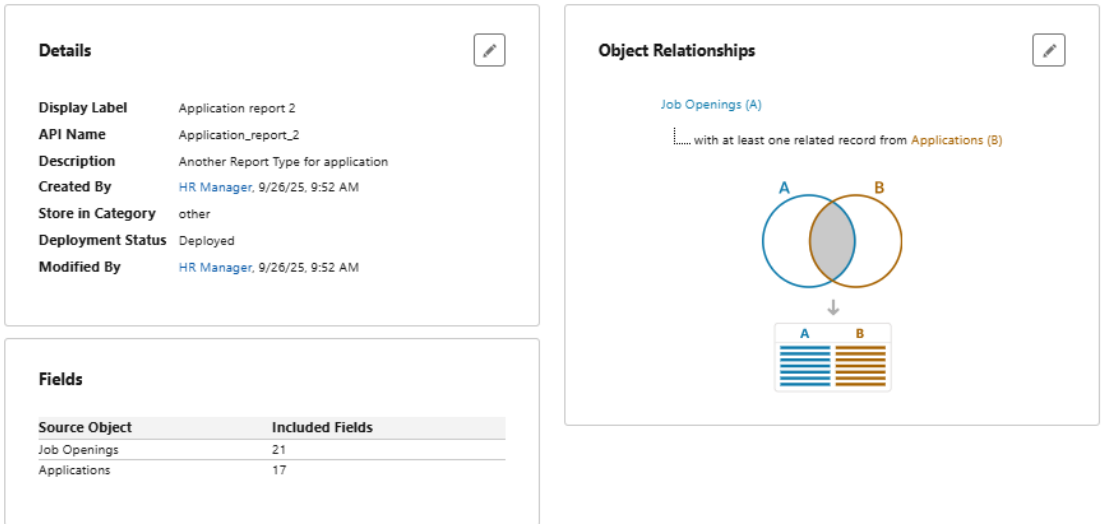
2. Report Types

Custom Report Types were created to combine related objects such as:

- **Applications with Job Postings**



- **Applications with Interviews**



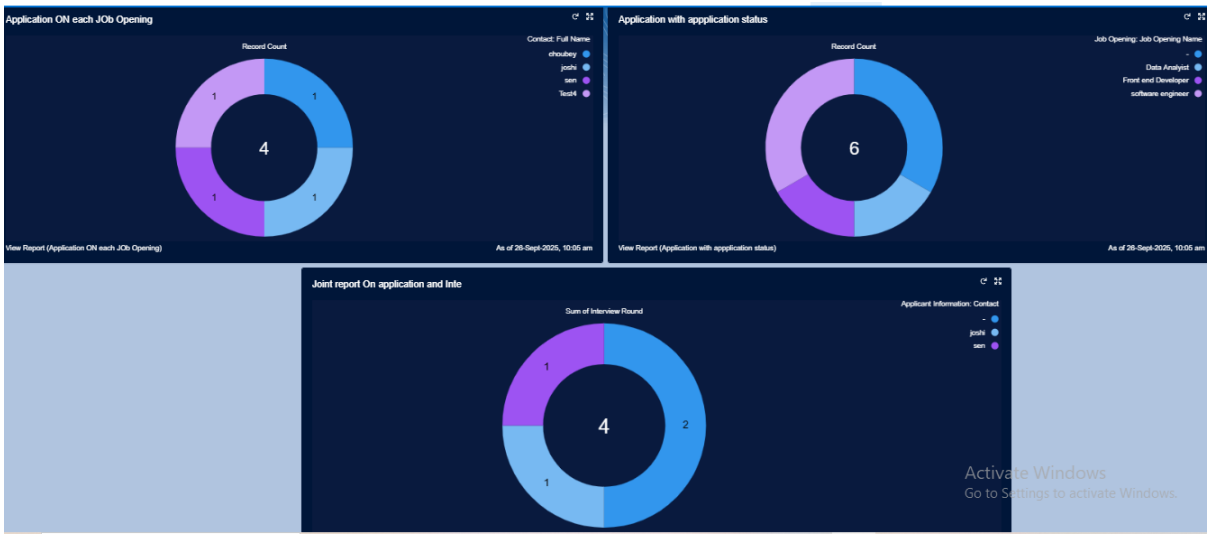
This enabled cross-object reporting to evaluate candidate progress throughout the hiring pipeline.

3. Dashboards

Dashboards were built to visually represent key recruitment KPIs:

- **Applications by Status** (Open, Approved, Rejected)
- **Applications per Job Posting**
- **Applications Selected for the interviews**

Each dashboard component provided recruiters with real-time insights into the hiring process.



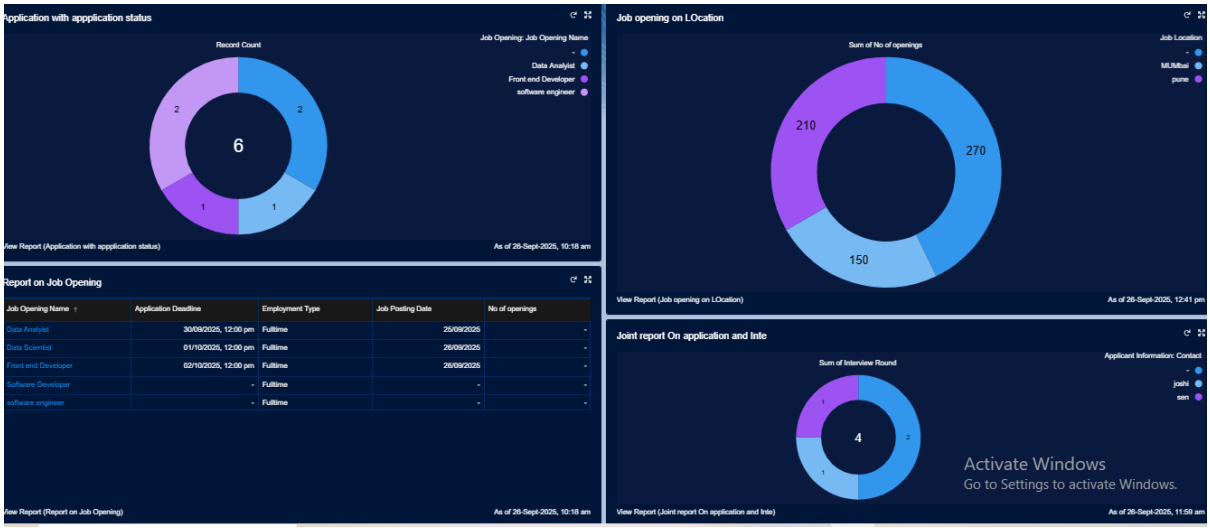
4. Dynamic Dashboards

Dynamic Dashboards were configured to ensure that users only see the data relevant to them. For example, if multiple recruiters are present, each recruiter would only see their own job applications and interviews. This was achieved by setting the dashboard

to run as the **logged-in user**, ensuring personalized views without the need for duplicate dashboards.

Dynamic Dashboard Includes

- **Application with application status**
- **Job opening with location**
- **Active job opening**
- **Applications selected for the interview**



5. Security Review

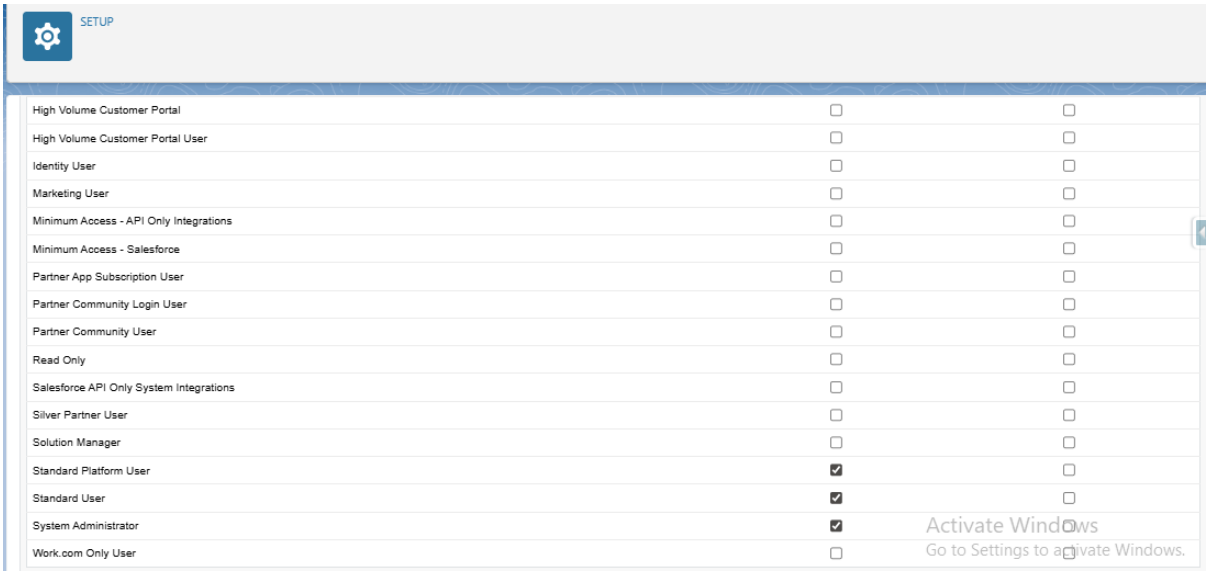
To ensure data integrity and compliance, several security measures were reviewed and implemented:

- **Sharing Settings:** Configured object-level and record-level access so that sensitive candidate and job data was only available to authorized users.
- I made application, interviews object private for the record level security and the job opening as public read only so the candidate can view the job opening.

Application	Private	Private	✓
Interview	Private	Private	✓
Job Opening	Controlled by Parent	Controlled by Parent	

- **Field-Level Security:** Restricted sensitive fields such as candidate contact details, ensuring only specific profiles could view or edit them. Hidden salary information from the external users.

Job salary field level security



- **Session Settings:** Enforced stricter login and session timeouts for security.
- **Login IP Ranges:** Configured IP restrictions for admin-level access, enhancing system security.
- **Audit Trail:** Enabled to keep track of all configuration changes, ensuring accountability and traceability.