integrations, positioning the "Smart Recruiter" project for continued success.

<u>Link the candidate Portal</u> -> <u>https://smartrecruiter-dev-ed.develop.my.site.com/careerconnect/s</u>

Phase 8: Data Management & Deployment

1. Data Import Wizard

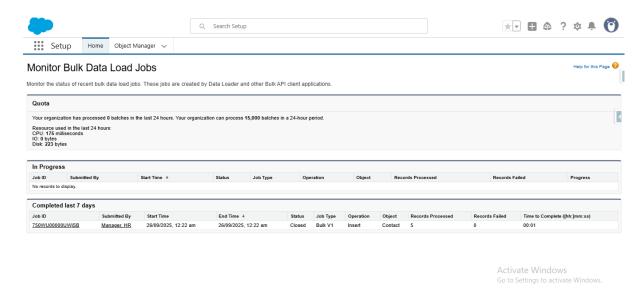
To populate sample data for testing and demonstration, the Salesforce **Data Import Wizard** was used for both standard and custom objects.

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:

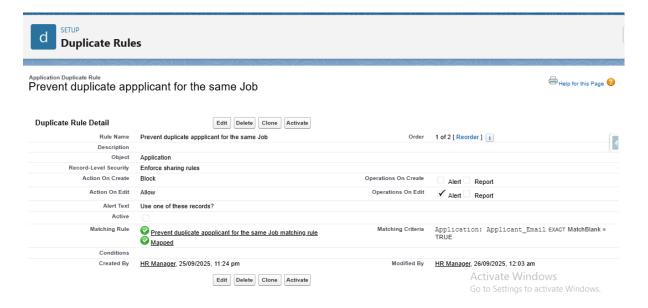
- Created Matching Rules on Contact object (Email equals).
- Created Duplicate Rules to Alert users if duplicate contacts are detected.
- Activated both rules.

Outcome:

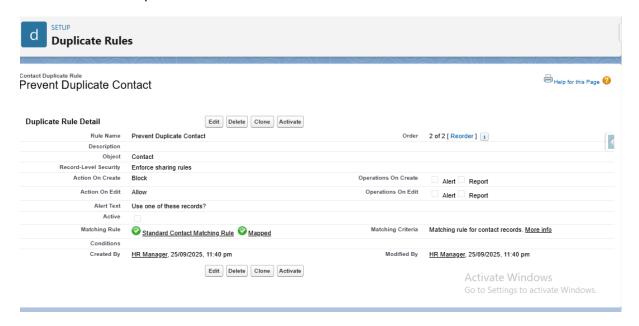
• Recruiters cannot create duplicate candidate records.

Ensures clean and reliable test data for recruitment processes.

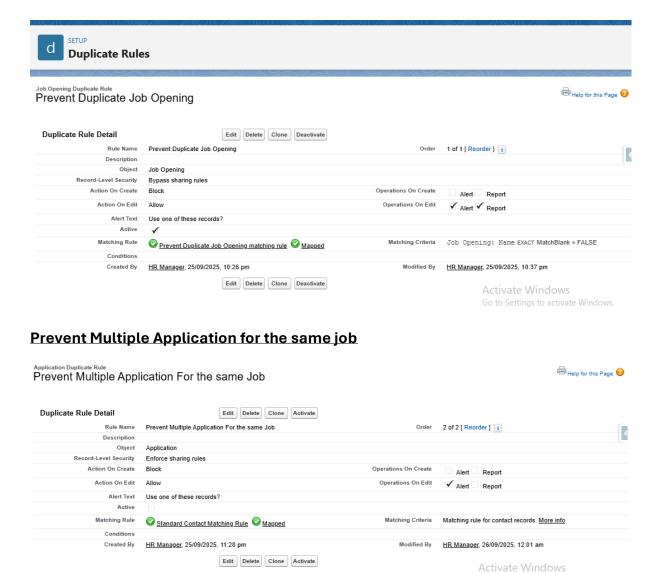
Prevent Duplicate Application For the Same Job



Prevent Duplicate Contact



Prevent Duplicate Job Opening



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.

Go to Settings to activate Windows.

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.