CRM Project – HR Recruitment Process

Phase 1: Problem Understanding & Industry Analysis

Goal:
Identify why organizations require a Salesforce-based HR Recruitment CRM and analyze the recruitment challenges in modern enterprises.

Problem Statement:
Organizations face delays in filling critical job positions due to fragmented recruitment processes, reliance on emails and spreadsheets, and poor visibility into candidate pipelines. These manual methods increase the risk of miscommunication, missed follow-ups, and inconsistent candidate evaluations. High attrition rates or rapid business expansion amplify these challenges, leading to prolonged vacancies, higher recruitment costs, and reduced productivity.

Solution:
A Salesforce-based HR Recruitment CRM will centralize job requisitions, candidate profiles, and interview schedules. It automates workflows such as candidate status tracking, interview reminders, and offer approvals. Dashboards and reports will give HR managers and recruiters real-time insights to optimize hiring cycles, improve candidate experience, and ensure data security.

Stakeholders:

- ✓ HR Manager → Creates and manages job openings, monitors KPIs, and approves offers.
- ✓ Recruiters → Source candidates, update statuses, and coordinate interviews.
- ✓ Department Heads → Approve job requisitions and participate in interview panels.
- ✓ IT Admin → Maintains CRM configuration, permissions, and integrations.
- ✓ Candidates → Apply for positions and track their application status.

Business Process Flow:
Job Request Raised → HR Creates Job Opening → Recruiters Source Candidates

(LinkedIn, Referrals, Portals) → Screening and Shortlisting → Interview Scheduling &

Feedback → Department Head Approval → Offer Generation & Communication →

Candidate Onboarding → Job Opening Closed.

KPIs:

- Time-to-Hire (average days to fill a role)
- Offer Acceptance Rate
- Candidate Conversion Rate (screened to hired)
- Recruiter Productivity (applications processed per recruiter)
- Employee Retention Rate post-hire

Requirement Gathering Highlights:

- Functional Requirements: Job creation, candidate management, interview scheduling, approvals, and dashboards.
- Non-Functional Requirements: Secure access, scalability for high-volume hiring, and mobile-friendly UI.
- Pain Points Identified: Lack of automated notifications, inefficient approval workflows, and scattered candidate data.

Industry-Specific Use Case Analysis:

- Benchmarked against popular ATS tools (e.g., Workday, LinkedIn Recruiter).
- Identified trends like AI-powered resume screening and automated reminders.
- Ensured compliance with data protection regulations (e.g., GDPR for candidate data privacy).

AppExchange Exploration:

- Reviewed Salesforce AppExchange for HR solutions such as Applicant Tracking Systems and Resume Parsers.
- Found potential integrations like LinkedIn Connector for Salesforce and resume parsing utilities.
- Selected components that could complement our custom recruitment workflow.

Phase 2: Org Setup & Configuration

Goal:

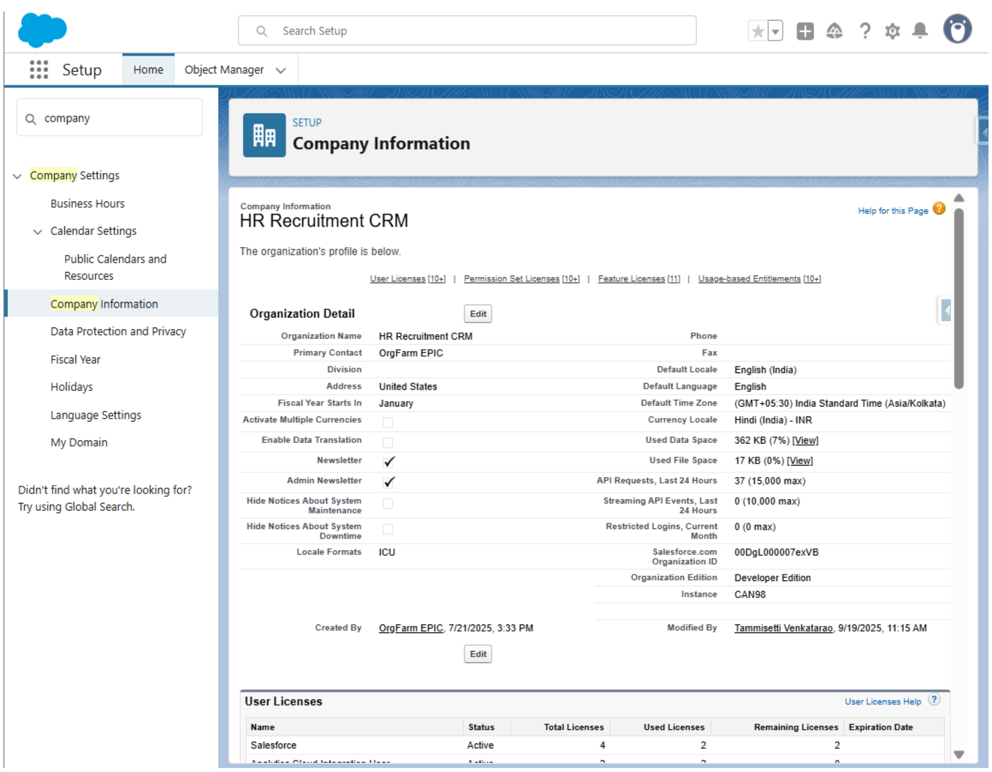
Prepare the Salesforce org for implementing the HR Recruitment Process CRM by configuring company settings, user roles, permissions, and foundational security measures.

Salesforce Edition:

- Developer Edition Org used for development and testing.
- Includes custom objects, Flows, Apex, Approval Processes, and Lightning components.
- Sandbox is used for testing automation and deployment before production.

Company Profile Setup:

- Navigated to Setup → Company Information.
- Updated organization name to "HR Recruitment CRM".
- Currency: INR, Locale: India, Time zone: IST.
- Ensured reports and currency fields reflect correct region settings.

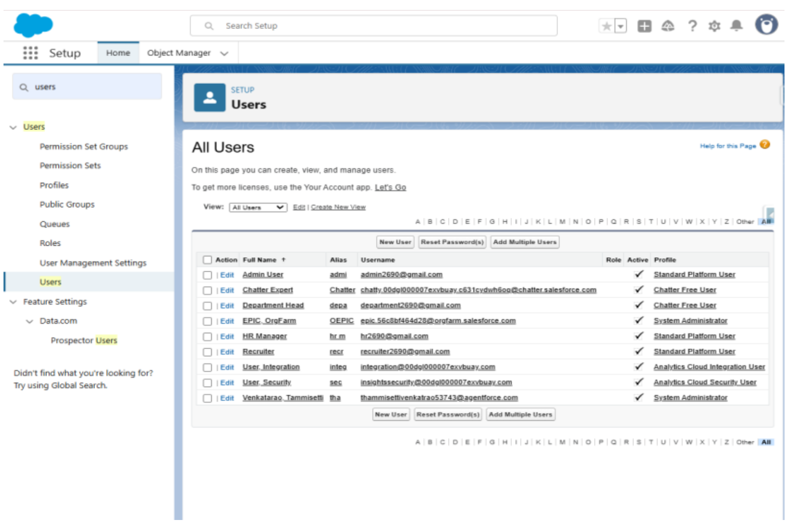


Fiscal Year Settings & Business Hours:

- Fiscal Year: Standard, starting from January.
- Defined Business Hours: Monday–Friday, 9 AM–6 PM.
- Added Key Holidays: Republic Day, Diwali, Independence Day.

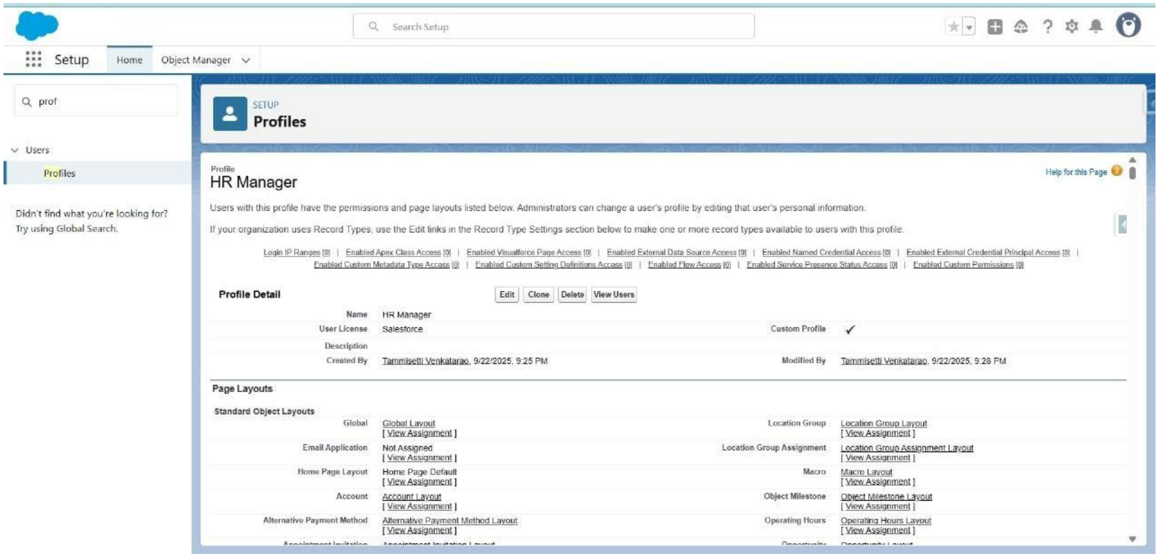
User Setup & Licenses:

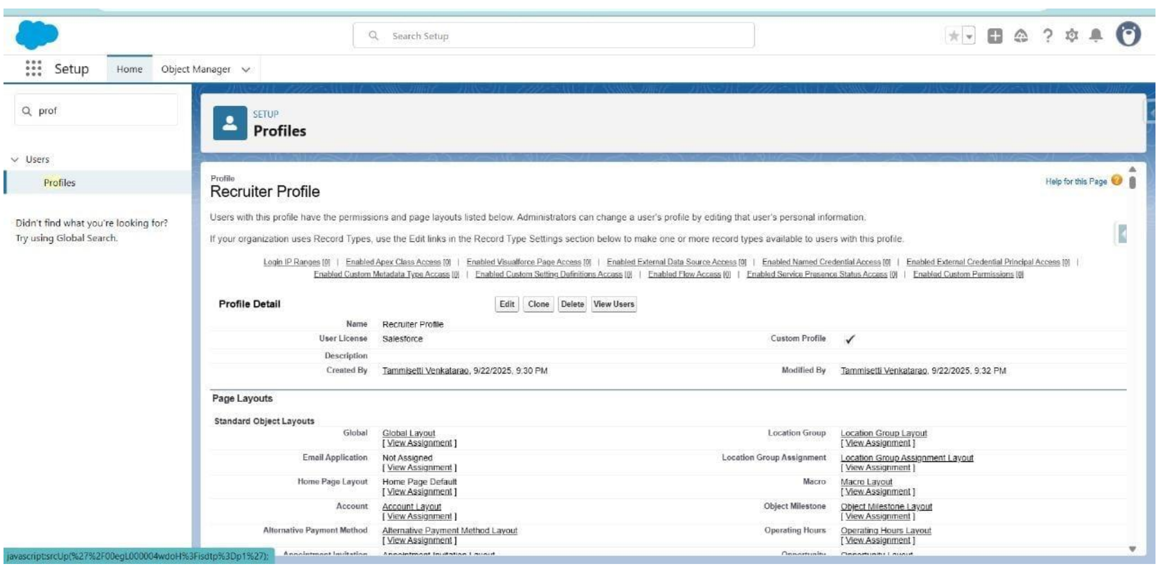
- Created test users for role-based testing:
    - Admin User – Full system access.
    - HR Manager – Approve offers and manage openings.
    - Recruiter – Manage candidate records and schedule interviews.
    - Department Head – View job openings and approve requisitions.
- Assigned Standard User licenses and mapped them to roles.



Profiles:

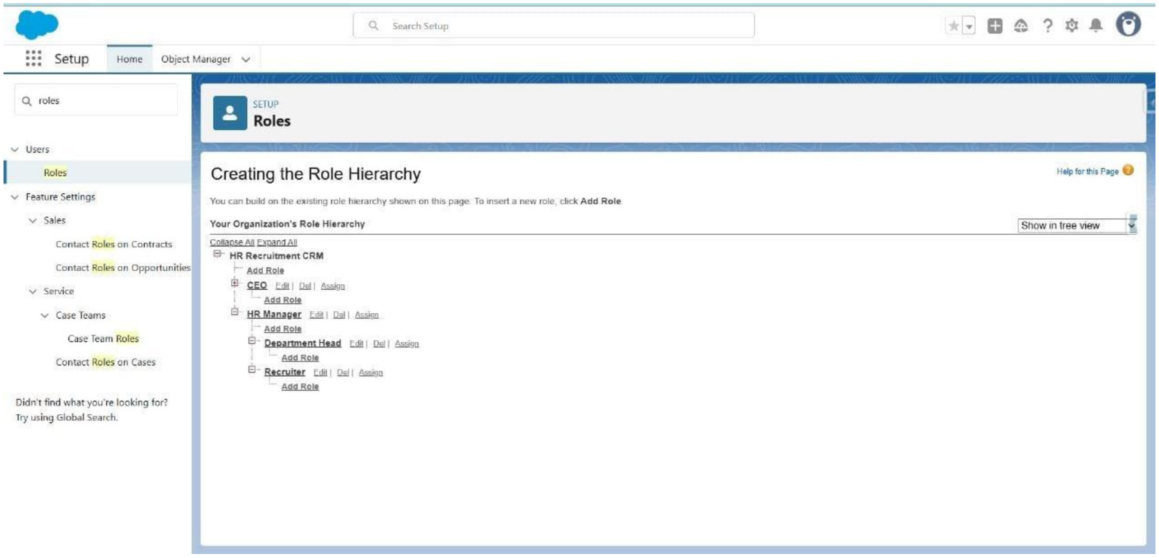
- Recruiter Profile → CRUD on Candidate & Interview objects, Read-only on Job Opening.
- HR Manager Profile → Full access on Job Openings, Candidates, and Interviews.
- Department Head Profile → Read/Edit on Job Openings and Interviews.
- Admin Profile → Full system access.





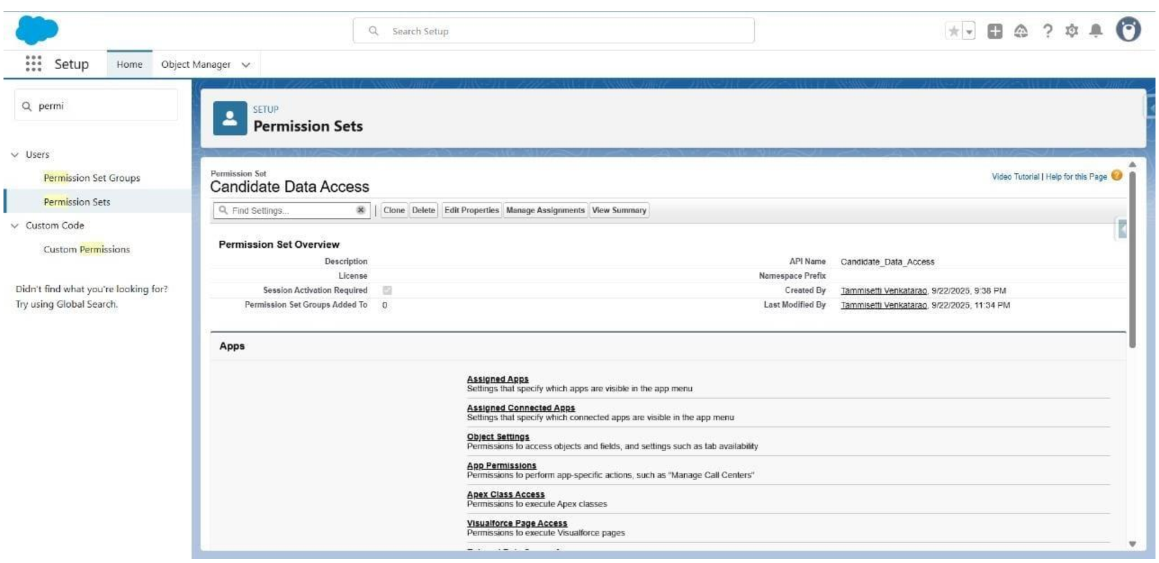
Roles & Role Hierarchy:

- Admin → HR Manager → Recruiter / Department Head.
- Ensures managers can view and approve records owned by subordinates.



Permission Sets:

- Created a "Candidate Data Access" permission set to grant temporary or additional permissions.
- Assigned permission sets for testing advanced access scenarios.

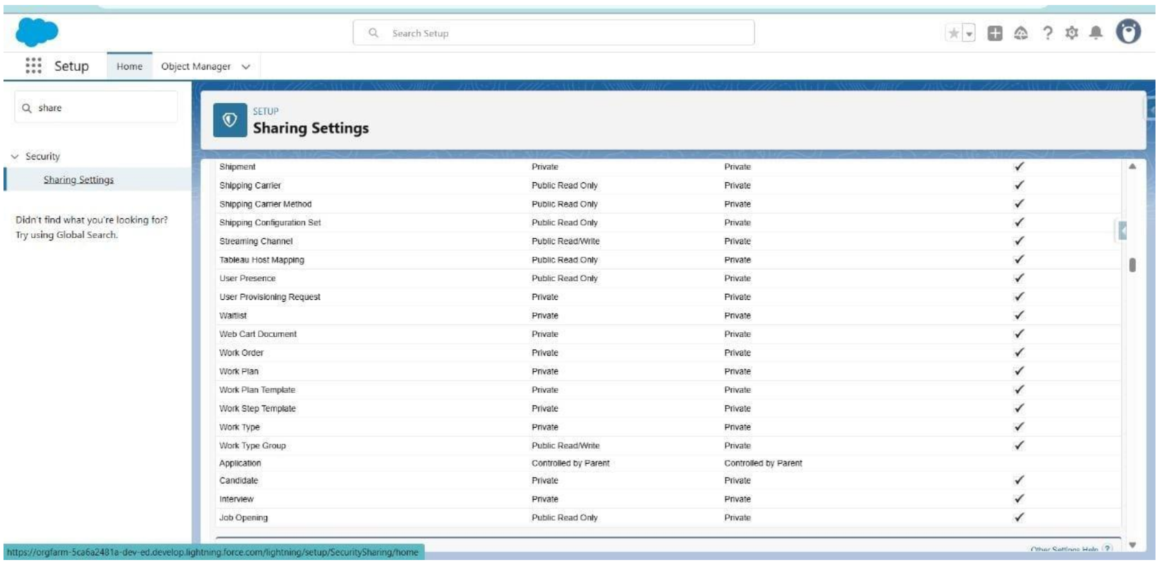


OWD (Organization-Wide Defaults) & Sharing Rules:

- Sharing Settings:
  - Job Openings = Private.
  - Candidates = Private.

- Interviews = Controlled by Parent (Candidate).

- Sharing Rules:
  - Share Candidate records owned by Recruiters with HR Managers (Read/Write).
  - Share Job Openings with Department Heads for visibility.



Login Access Policies:

- Default policies retained (Admins can log in as users to troubleshoot issues).
- Enabled IP restrictions for added security.

Sandbox Usage & Deployment Basics:

- Created a Developer Sandbox for testing automation and approval flows.
- Deployment Basics:
  - Used Change Sets to migrate configurations to production.
  - Validated deployments before applying to the live environment.

Deliverables:

- Configured Salesforce org with company profile, business hours, and fiscal year.
- Defined user roles, profiles, and permission sets.
- Established OWD, sharing rules, and login policies for secure data access.
- Sandbox created and deployment steps documented.

Phase 3: Data Modeling & Relationships

Goal:
Build a robust data model that represents jobs, candidates, applications, interviews, and related entities. Create relationships (master-detail, lookup, junction), record types, page layouts, compact layouts, and ensure the model supports reporting and automation.

1. Create Custom Objects (core objects)

Purpose: Create objects to store recruitment data.

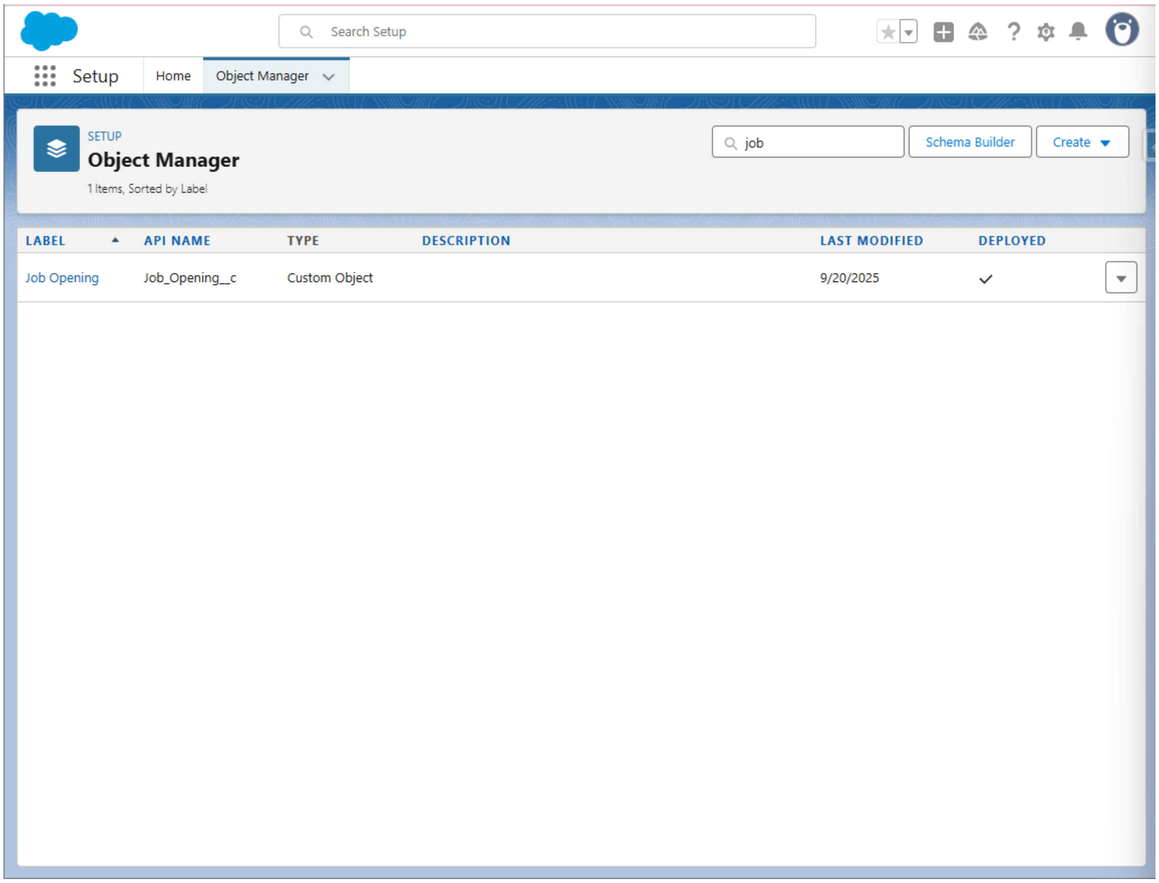
Objects to create (recommended):

- **Job_Opening__c** — stores job requisitions.
- **Candidate__c** — stores candidate profile & contact info.
- **Application__c** — *junction object* between Candidate and Job Opening (one candidate can apply to many jobs, one job can have many candidates).
- **Interview__c** — interview records linked to Application (or Candidate).
- **Resume__c** (optional) — file/reference to candidate resume or parsed data.

Steps (example for Job_Opening__c):

1. Click **Setup** → enter **Object Manager** in Quick Find → **Object Manager**.
2. Click **Create** → **Custom Object**.
3. For **Label** enter: Job Opening
   Plural Label: Job Openings
   Object Name (API): Job_Opening__c
   Record Name: Job Opening Name (Auto-Number or Text)
4. Check **Allow Reports**, **Allow Activities**, and **Track Field History** (as needed).
5. Click **Save**.

Repeat the same for **Candidate__c**, **Application__c**, **Interview__c**, etc.



2. Add Fields to Objects (detailed examples)

General: For each object, create fields with clear API names and types.

Job_Opening__c — suggested fields

- Job_Code__c — Auto Number JOB-{0000}
- Job_Title__c — Text (255)
- Department__c — Picklist (e.g., Sales, Engineering, HR, Finance)
- Hiring_Manager__c — Lookup(User)
- Positions_Open__c — Number (Integer)
- Status__c — Picklist (Draft, Open, Closed, On Hold)
- Location__c — Text or Picklist
- Salary_Range_Low__c & Salary_Range_High__c — Currency

Candidate__c — suggested fields

- FirstName__c, LastName__c — Text (or use Contact standard object)
- Email__c — Email (Required)
- Phone__c — Phone
- Resume_Link__c — URL or File (use Files)
- Source__c — Picklist (LinkedIn, Referral, Job Portal, Walk-in)
- Current_Status__c — Picklist (Applied, Screened, Interview, Offered, Hired, Rejected)
- Current_Score__c — Number (3,1) — optional scoring field.

Application__c (junction between Candidate & Job)

- Candidate__c — Master-Detail → Candidate__c
- Job_Opening__c — Master-Detail → Job_Opening__c
- Application_Date__c — Date
- Stage__c — Picklist (Applied, Screening, Interviewing, Offer, Hired, Rejected)
- Resume_Attached__c — Checkbox

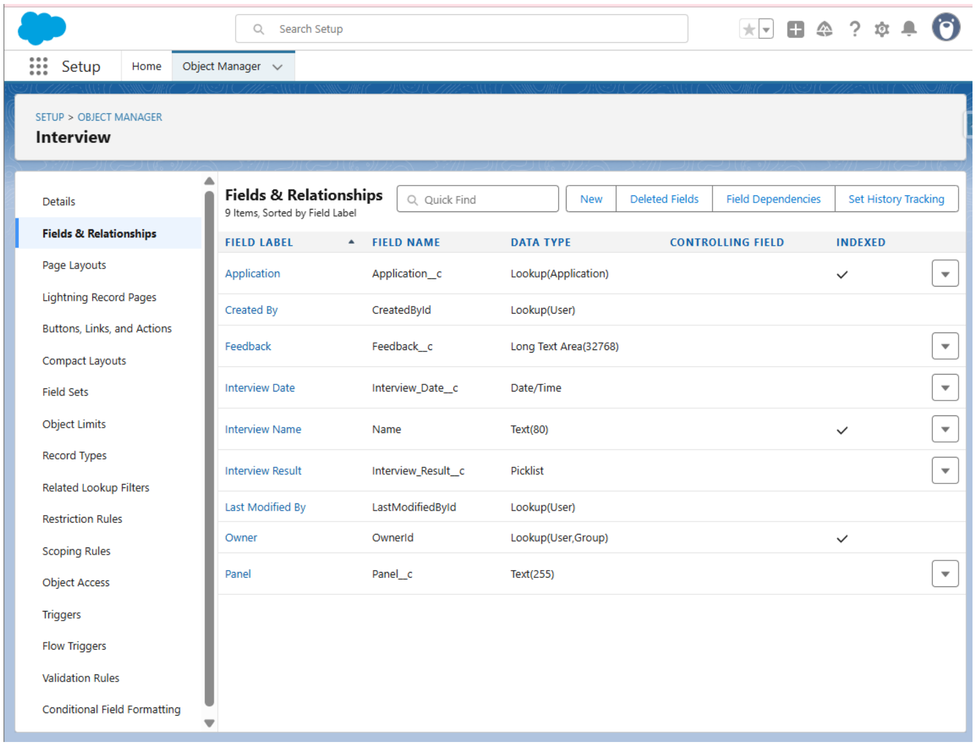
Interview__c

- Application__c — Lookup(Application__c) or Master-Detail to Application
- Interview_Date__c — Date/Time
- Panel__c — Text / Related Users (could be a multi-select picklist of panel members or create Interview_Panel__c child records)
- Feedback__c — Long Text Area
- Interview_Result__c — Picklist (Pass, Fail, Hold)

Steps to add a field (example Email on Candidate__c):

1. Setup → Object Manager → open **Candidate → Fields & Relationships → New**.
2. Choose **Email** type → Next.

Field Label: Email → Field Name: Email__c → Set **Required** if desired → Next → Set field-level security → Next → Add to Page Layouts → Save.



3. Create Relationships (lookup, master-detail, junction)

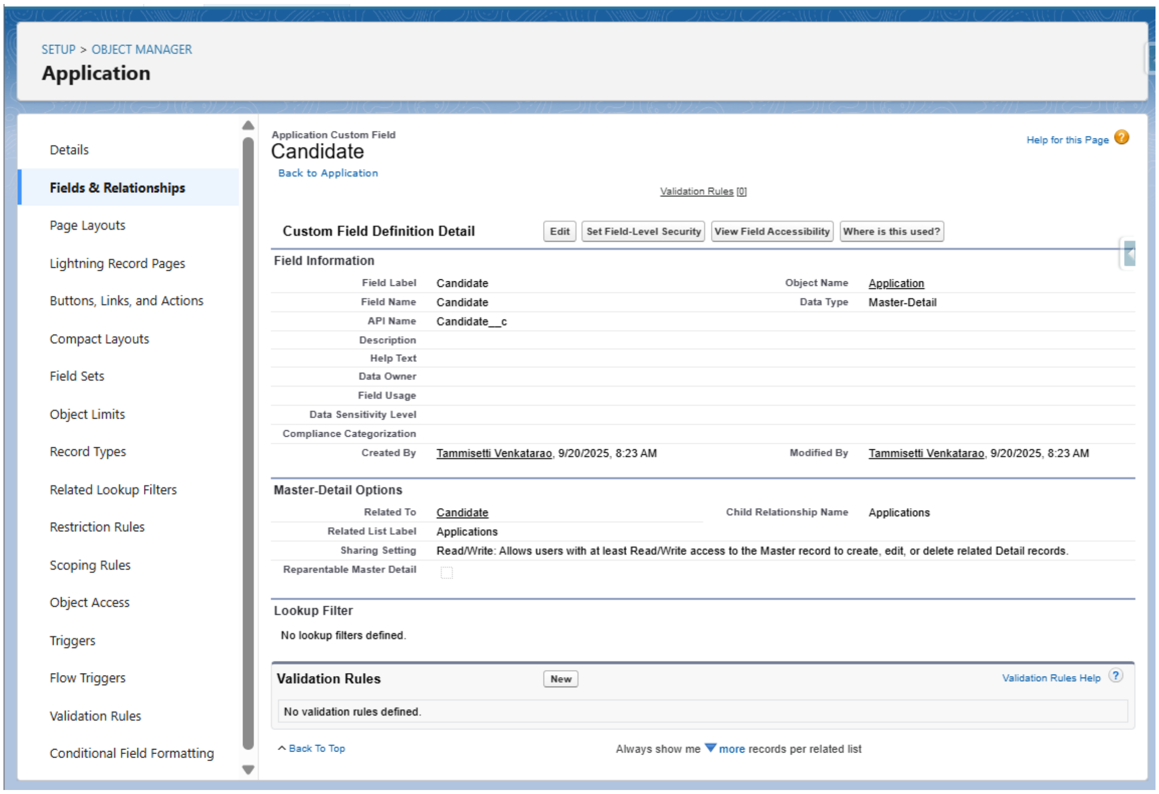
When to use which:

- **Master-Detail:** Use when child record should inherit parent sharing and be deleted when parent deleted (e.g., Application as master-detail both Candidate & Job).
- **Lookup:** Use when child should be independent (e.g., Interview may be a lookup to Application if you want interviews to survive application deletion).
- **Junction Object:** Use Application__c to model many-to-many between Job and Candidate.

Step to create a Master-Detail (Application → Candidate):

1. Setup → Object Manager → **Application** → **Fields & Relationships** → **New**.
2. Choose **Master-Detail Relationship** → Next.
3. Related To: **Candidate** → Next.
4. Field Label: Candidate → Field Name: Candidate__c → Next → Set sharing and behavior → Save.
5. Repeat for **Job_Opening** master-detail.

Result: Application now shows as related list on both Candidate and Job Opening pages.



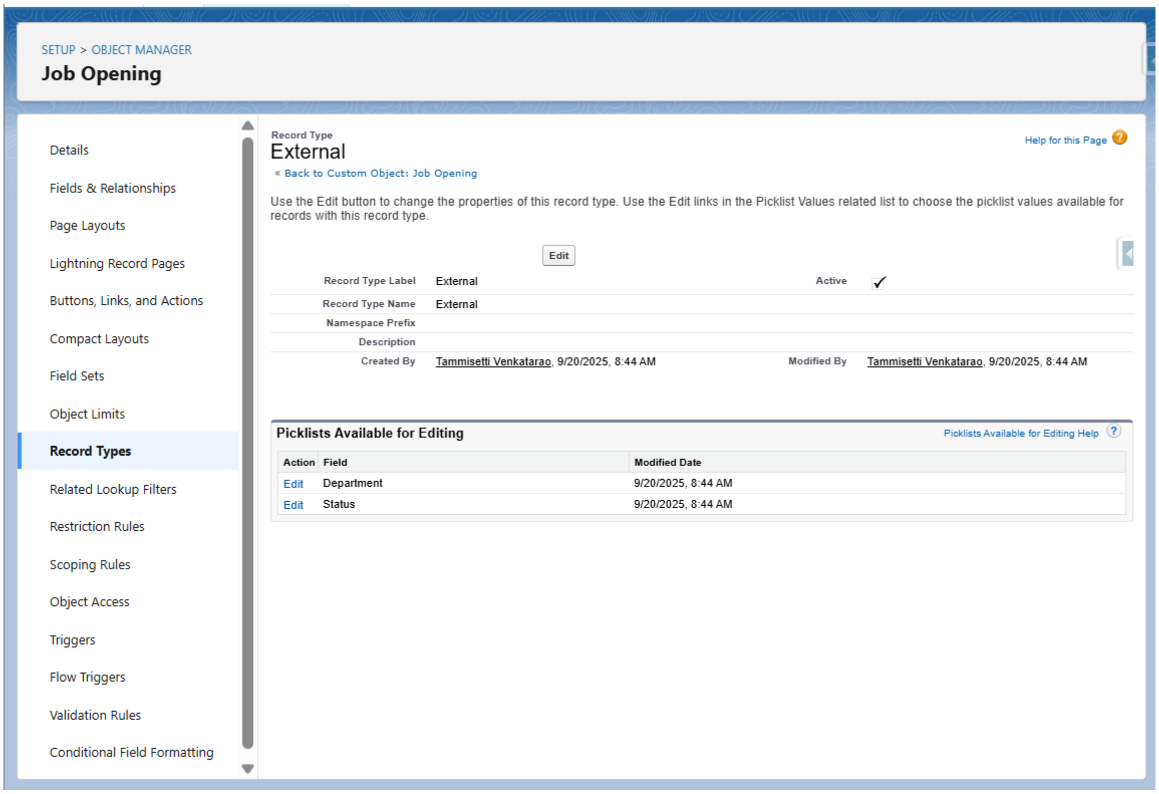
4. **Create Record Types & Picklist Variants**

Use case: Show different fields or page layouts for **Internal Hiring vs External Hiring or Referral vs Open Application.**

Steps:

1. Setup → Object Manager → **Job Opening** → **Record Types** → **New**.
2. Select existing profile defaults to clone.
3. Enter Record Type Label: Internal and API name: Internal.
4. Repeat to create External.
5. For each Record Type, assign Page Layouts and set picklist value availability.

Tip: Use record types when fields, picklist values, or required fields differ by hiring type.



5.Page Layouts & Compact Layouts

Page Layouts control detail page appearance; **Compact Layouts** control highlights on record cards & mobile.

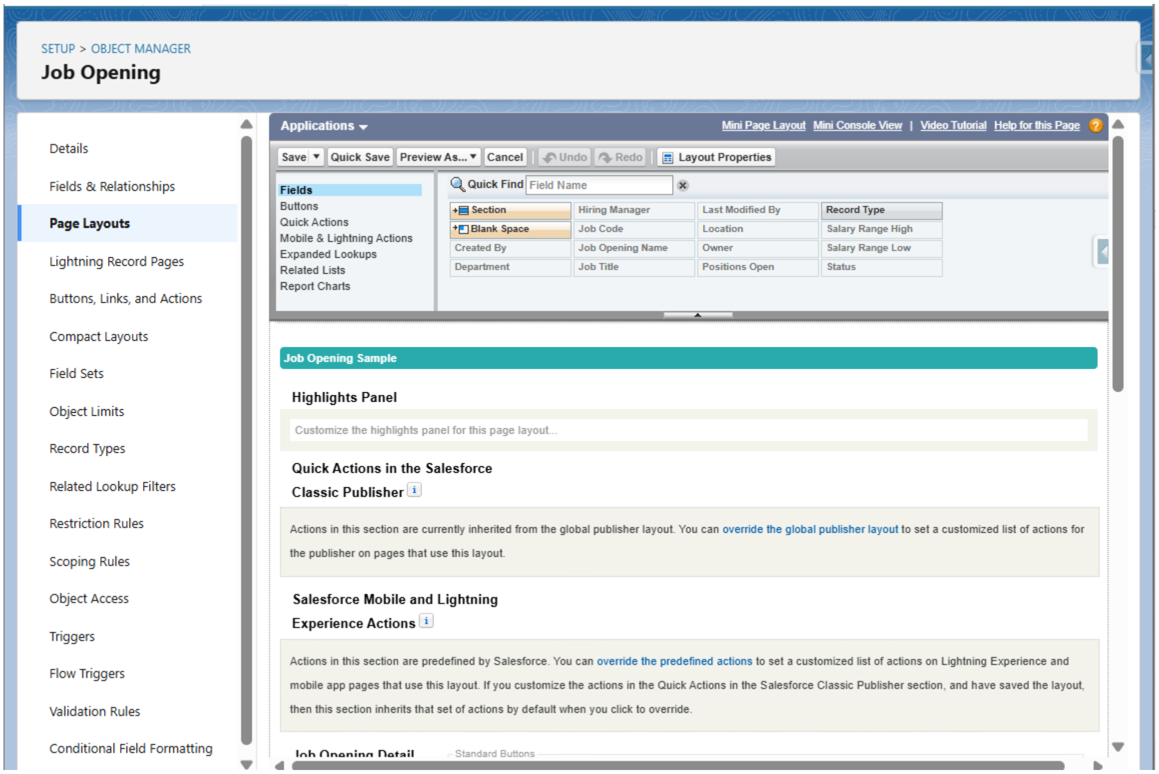
Create/Edit Page Layout:

1. Setup → Object Manager → Job Opening → **Page Layouts** → **New** (or edit existing).
2. Drag/Drop fields, Related Lists (Applications, Interviews), Buttons.
3. Save.
4. Assign page layout to profiles and record types (Page Layout Assignment).

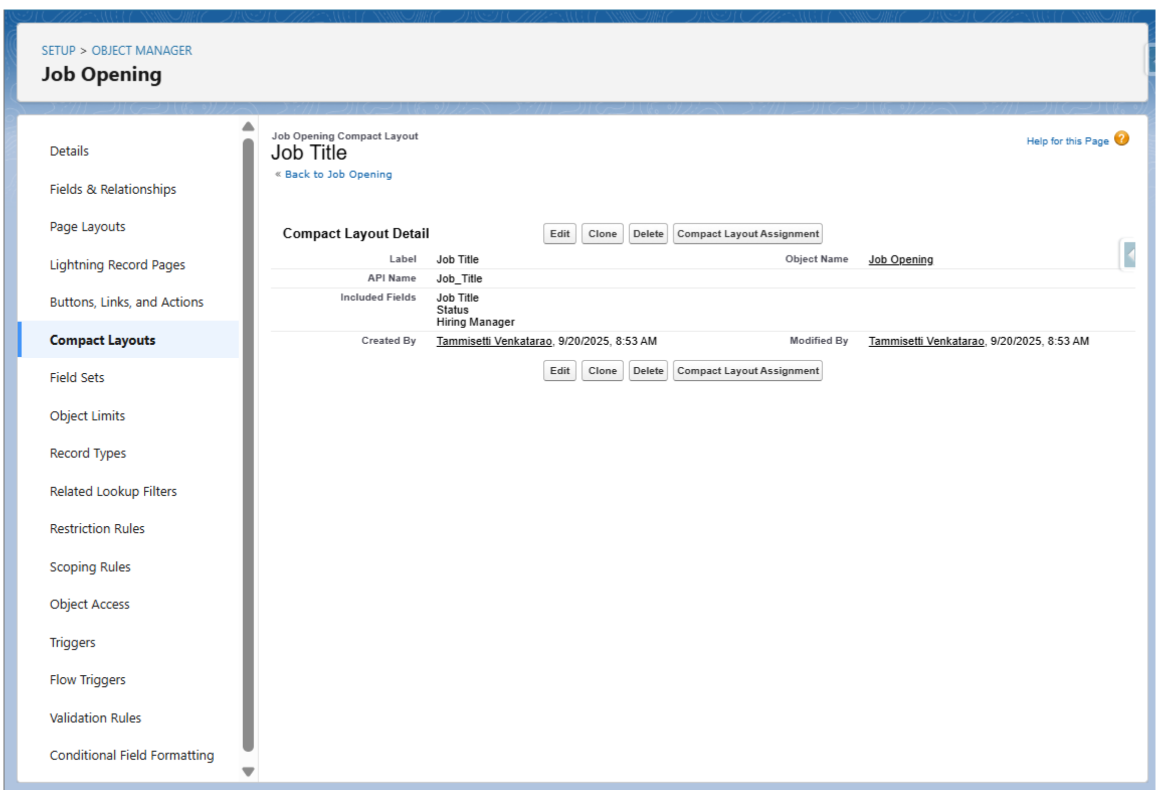
Create Compact Layout:

1. Setup → Object Manager → Job Opening → **Compact Layouts** → **New**.
2. Choose fields shown in the highlights panel (Job Title, Status, Hiring Manager).

Save → Set as the org default or assign by record type.



SETUP > OBJECT MANAGER
Job Opening
Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Restriction Rules
Scoping Rules
Object Access
Triggers
Flow Triggers
Validation Rules
Conditional Field Formatting
Applications
Mini Page Layout | Mini Console View | Video Tutorial | Help for this Page
Save | Quick Save | Preview As... | Cancel | Undo | Redo | Layout Properties
Fields
Buttons
Quick Actions
Mobile & Lightning Actions
Expanded Lookups
Related Lists
Report Charts
Quick Find Field Name
Section
Blank Space
Created By
Department
Hiring Manager
Job Code
Job Opening Name
Job Title
Last Modified By
Location
Owner
Positions Open
Record Type
Salary Range High
Salary Range Low
Status
Job Opening Sample
Highlights Panel
Customize the highlights panel for this page layout...
Quick Actions in the Salesforce Classic Publisher
Actions in this section are currently inherited from the global publisher layout. You can override the global publisher layout to set a customized list of actions for the publisher on pages that use this layout.
Salesforce Mobile and Lightning Experience Actions
Actions in this section are predefined by Salesforce. You can override the predefined actions to set a customized list of actions on Lightning Experience and mobile app pages that use this layout. If you customize the actions in the Quick Actions in the Salesforce Classic Publisher section, and have saved the layout, then this section inherits that set of actions by default when you click to override.
Standard Buttons



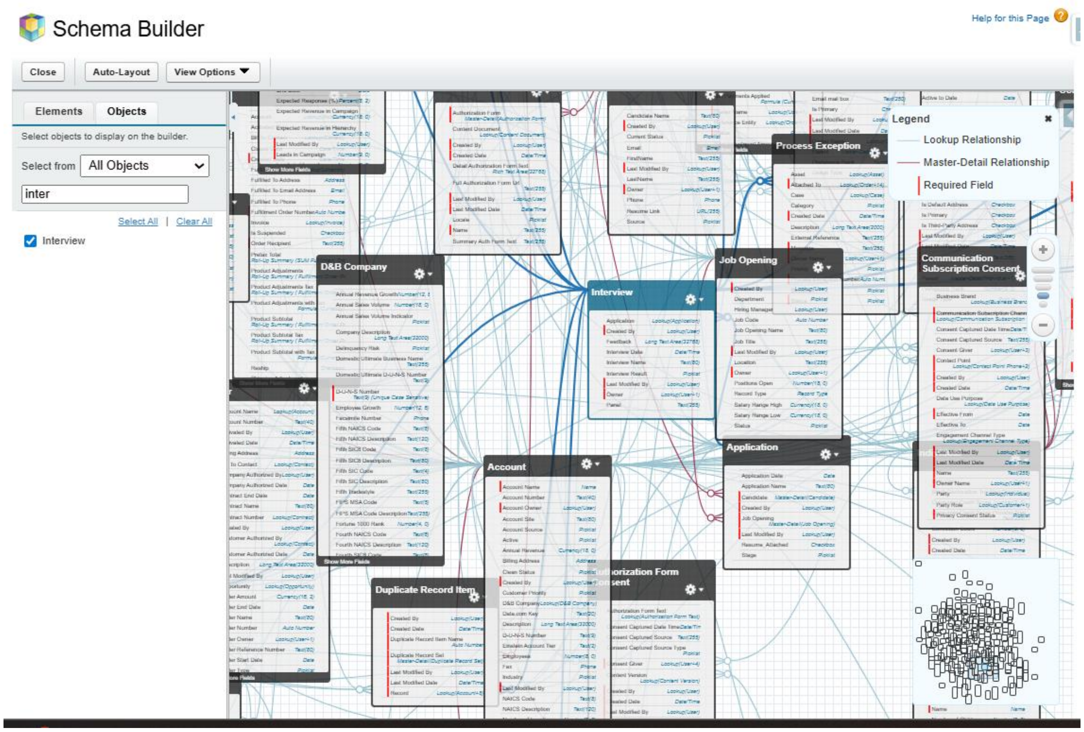
SETUP > OBJECT MANAGER
Job Opening
Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Restriction Rules
Scoping Rules
Object Access
Triggers
Flow Triggers
Validation Rules
Conditional Field Formatting
Job Opening Compact Layout
Job Title
« Back to Job Opening
Help for this Page
Compact Layout Detail
Edit | Clone | Delete | Compact Layout Assignment
Label Job Title
Object Name Job Opening
API Name Job_Title
Included Fields Job Title
Status
Hiring Manager
Created By Tammisetti Venkatarao, 9/20/2025, 8:53 AM
Modified By Tammisetti Venkatarao, 9/20/2025, 8:53 AM
Edit | Clone | Delete | Compact Layout Assignment

6. Schema Builder (visualize & adjust)

Steps:

1. Setup → enter **Schema Builder** in Quick Find → **Schema Builder**.
2. From the left pane, check the objects you created (Job_Opening, Candidate, Application, Interview).
3. Drag objects onto canvas to view relationships.
4. You can also create fields or relationships from Schema Builder (click on object → Add field).

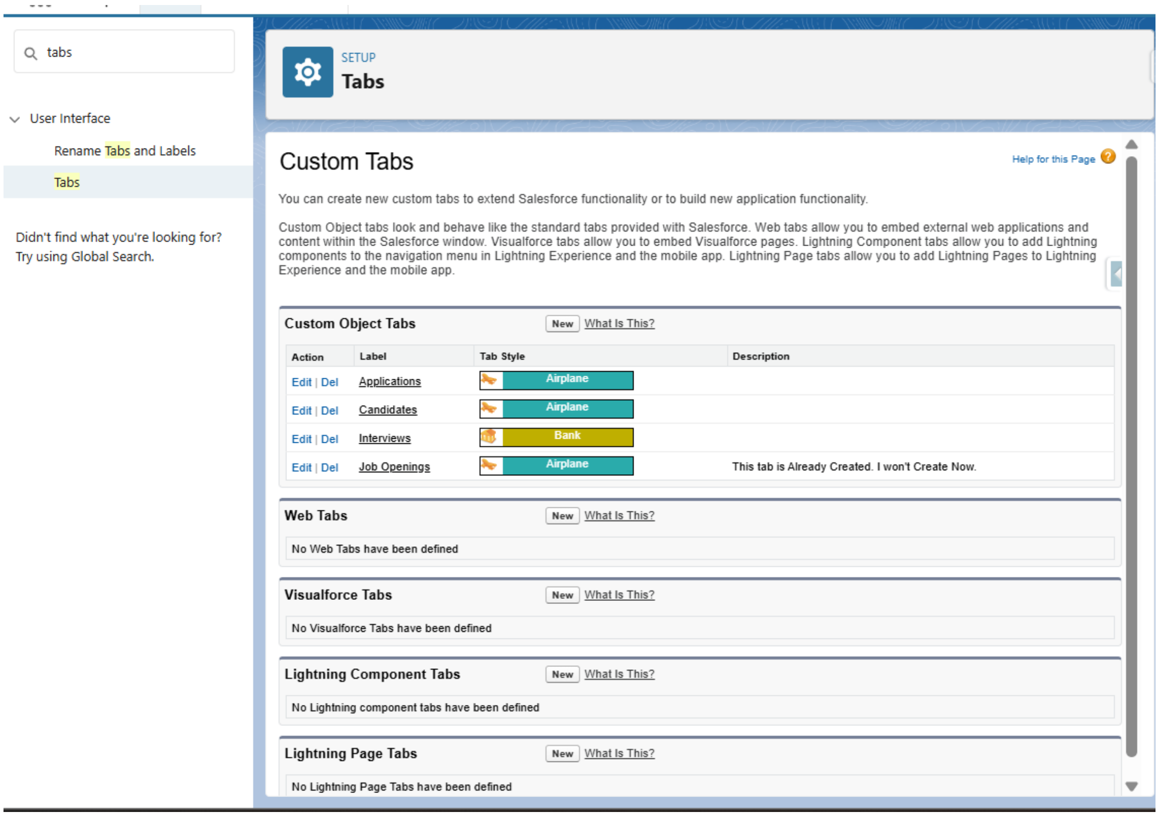
Use it to verify your ERD and to export/interpret model when writing documentation.



7. Create Tabs & Add to App

Steps:

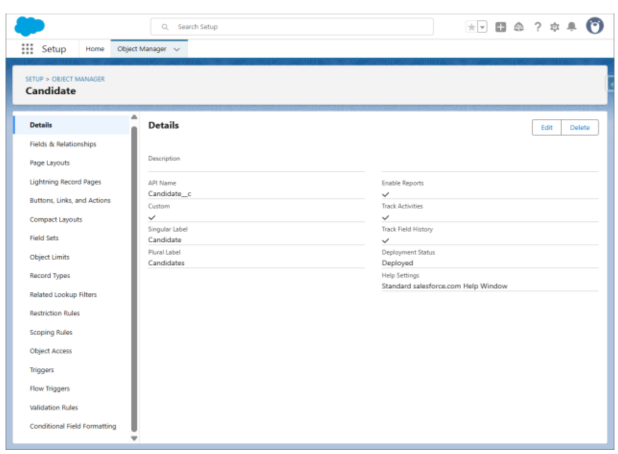
1. Setup → Quick Find → **Tabs** → **New** (Custom Object Tabs).
2. Select **Job Opening** → Choose tab style → Next → Add to desired Apps (Recruitment App) → Save.
3. Repeat for Candidate and Application.



8. Search Layouts & List Views

- Setup → Object Manager → Candidate → **Search Layouts for Salesforce Classic** / **Search Layouts for Lightning Experience** → Configure fields that show in search results.

Create List Views for hiring managers: e.g., Open Applications, Interviews Today, Offers Pending.



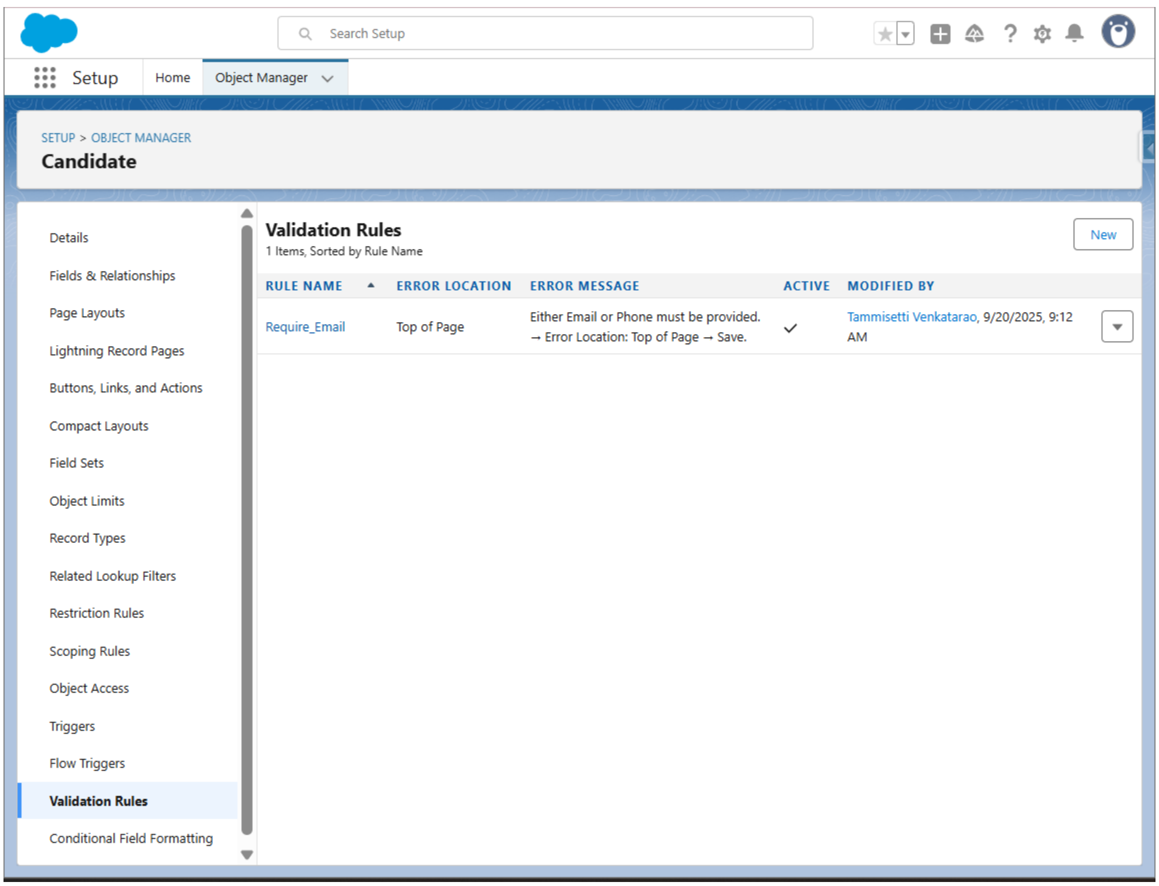
9.Validation Rules (examples)

Example 1 — Require Email or Phone for Candidate:

- Setup → Object Manager → Candidate → **Validation Rules** → **New**.
- Rule Name: Require_Email_or_Phone
- Formula:

AND(ISBLANK(Email__c), ISBLANK(Phone__c))

Error Message: Either Email or Phone must be provided. → Error Location: Top of Page → Save.



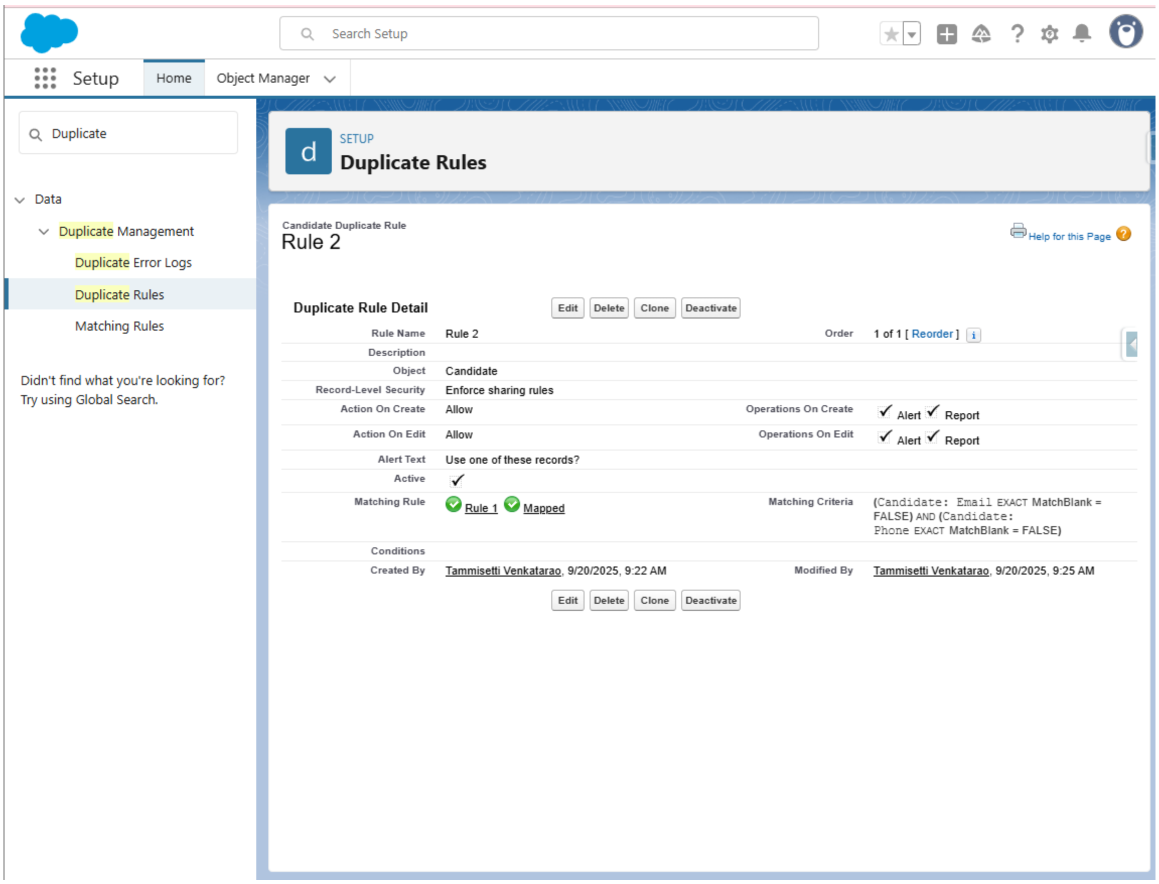
10.Duplicate Management (Matching & Duplicate Rules)

Create Matching Rule (Candidate by Email & Phone):

1. Setup → Quick Find: **Matching Rules** → **New Rule**.
2. Object: Candidate. Matching Criteria: Email (Exact), Phone (Exact).
3. Save → Activate.

Create Duplicate Rule:

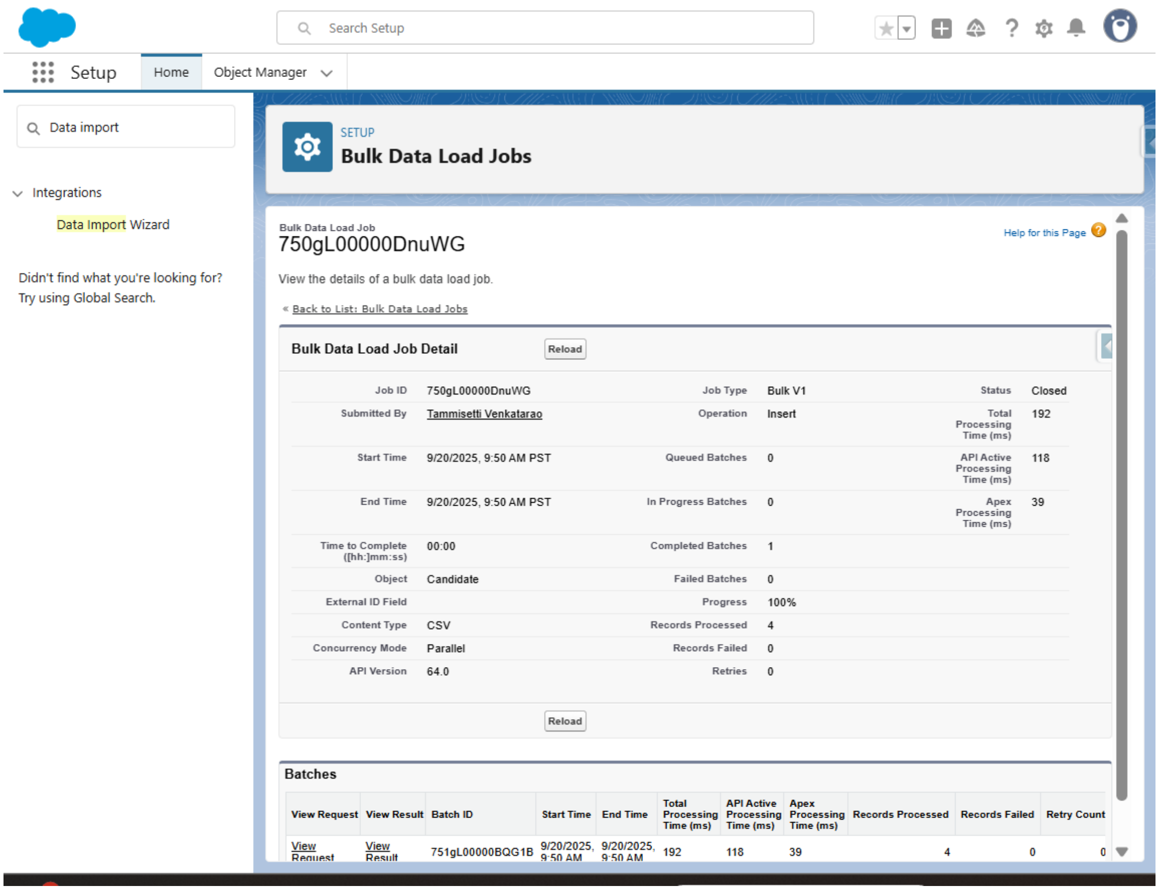
1. Setup → Quick Find: **Duplicate Rules** → **New Rule**.
2. Object: Candidate. Rule Type: Use the matching rule just created.
3. Action on Create/Edit: Block or Allow and Report (choose Allow and Report during testing).
4. Save → Activate.



11.Import Sample Data

Small import (Data Import Wizard):

1. Setup → Quick Find → **Data Import Wizard** → Launch Wizard.
2. Choose object: Candidates (or Accounts/Contacts if using Contact).
3. Upload CSV file with mapped headers (FirstName, LastName, Email, Phone, Source).
4. Map CSV columns to Salesforce fields → Start Import.
5. Review import results.



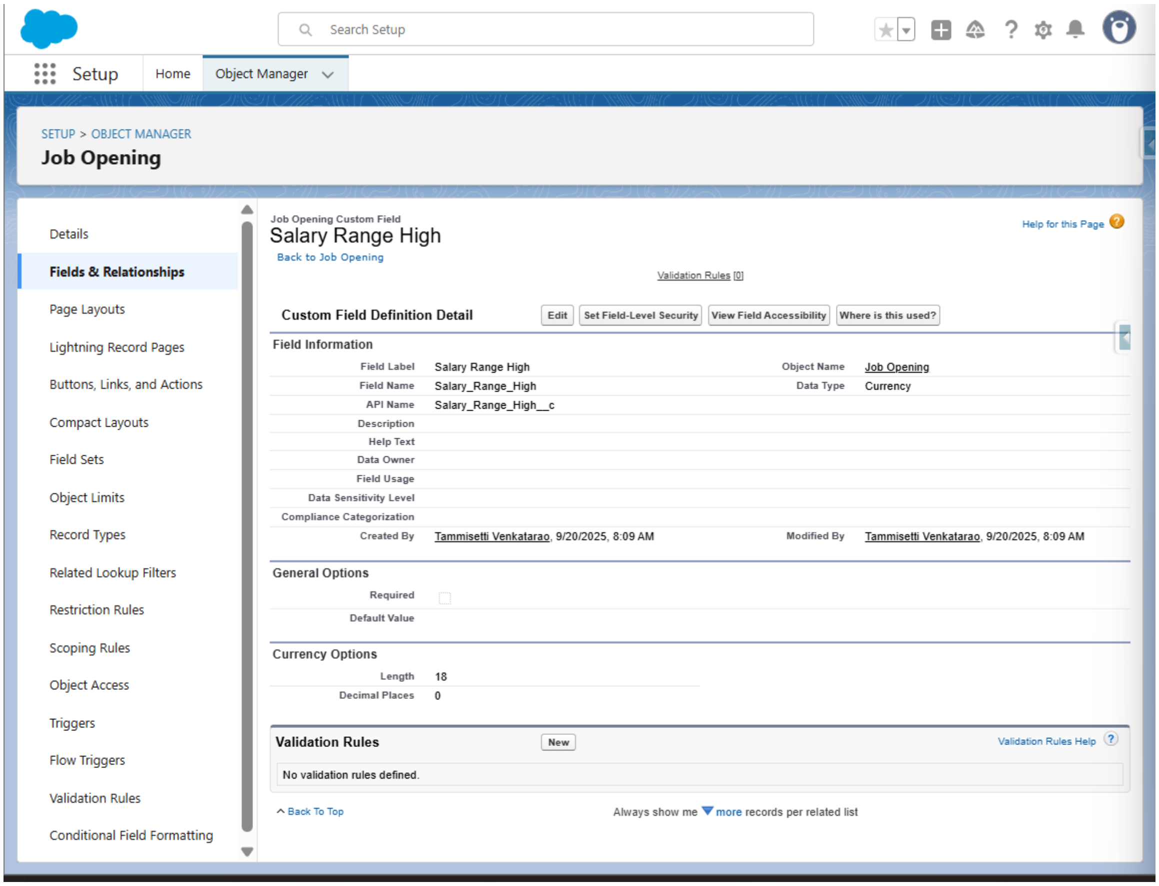
Large imports (Data Loader):

- Use Data Loader to insert Application records or historical data; map lookup fields using external IDs or Salesforce IDs.

Tip: For Master-Detail relationships during import, either import parent records first (Candidate, Job) and use their Salesforce IDs in child records, or use External ID fields to match.

12.Security for Objects & Fields

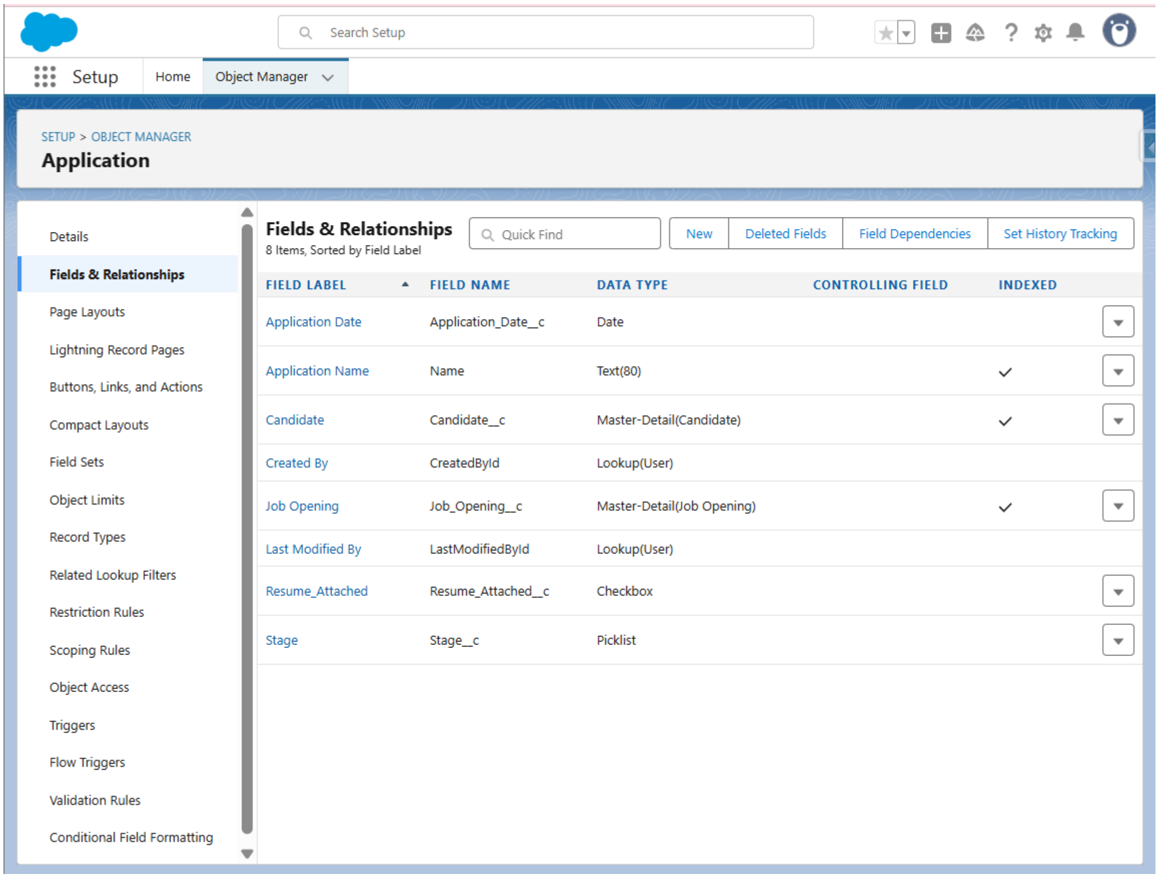
- For each sensitive field (e.g., Salary_Range), set **Field-Level Security**: Setup → Object Manager → Field → **Set Field-Level Security** — hide for recruiter if needed.
- Use **Permission Sets** to grant special access to certain users.



13.Use a Junction Object for Many-to-Many (Application example)

Why Application__c: A Candidate can apply to multiple Job Openings and each Job Opening can have many Candidates.

- Create Application__c → two master-detail fields: Candidate__c and Job_Opening__c.
- This makes Applications appear in related lists on both Candidate and Job Opening pages and supports per-application stages.



Documenting the Model & Deliverables

Deliverables for Phase 3:

- ERD (Entity Relationship Diagram) — can export from Schema Builder or draw in Visio/Miro.
- Object & field dictionary (table with Field Label, API Name, Type, Description, Required?).
- Record Type list & page layout assignment matrix (which profile sees which layout).
- Validation rules list with formulas and purpose.
- Duplicate/matching rules configuration.
- Sample CSV files used to import test data and import logs.
- Screenshots of key configurations (Object Manager, schema builder, page layout).

Phase 4: Process Automation (Admin)

Goal:

Automate recruitment workflows using declarative Salesforce tools (Validation Rules, Flows, Approval Processes, Email Alerts, Tasks, and Scheduled Automations). Ensure automations are maintainable, tested in Sandbox, and documented.

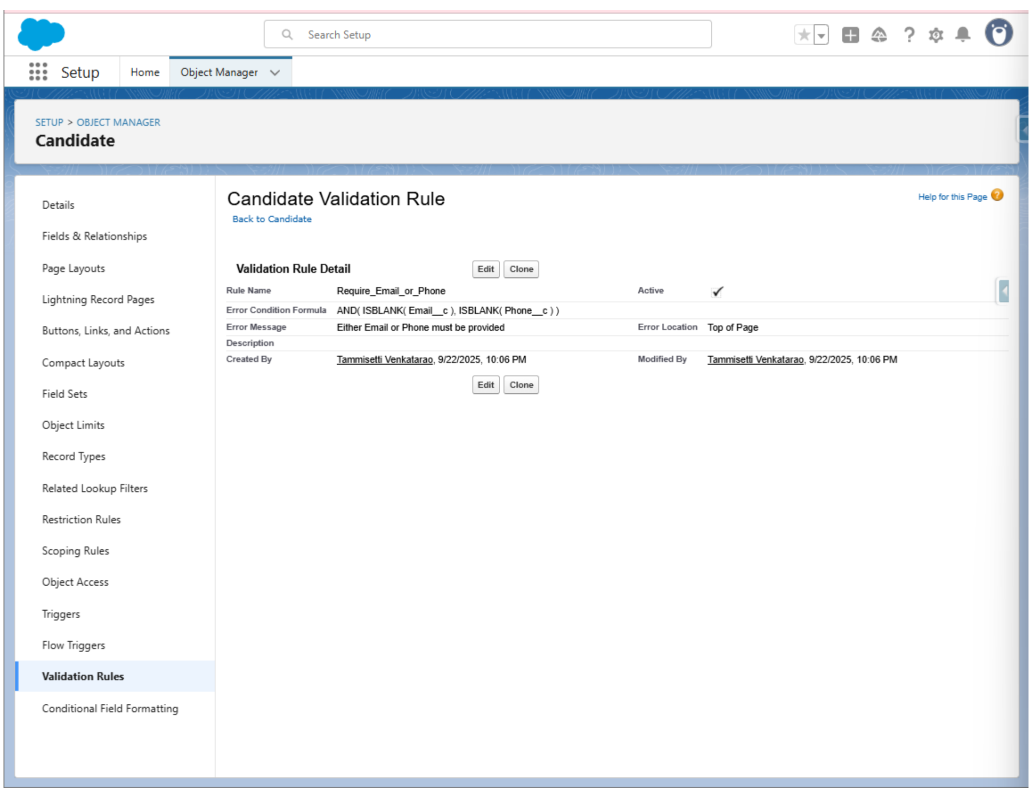
1) Validation Rules (examples)

Example A — Require Email OR Phone on Candidate

1. Click **Setup → Object Manager → Candidate → Validation Rules → New**.
2. Rule Name: Require_Email_or_Phone
3. Error Condition Formula:

    AND(ISBLANK(Email__c), ISBLANK(Phone__c))

4. Error Message: Either Email or Phone must be provided.
5. Error Location: **Top of Page → Save**.



2) Create Email Templates & Alerts (prepare first)

Create a Lightning Email Template (Congratulate Hired)

1. App Launcher → **Email Templates** → **New Email Template**.
2. Name: Candidate_Hired_Congrats
3. Related Entity Type: **Application** (or Candidate)
4. Subject: Congratulations — Offer Accepted for {!Application__c.Job_Opening__r.Job_Title__c}
5. Body (example):

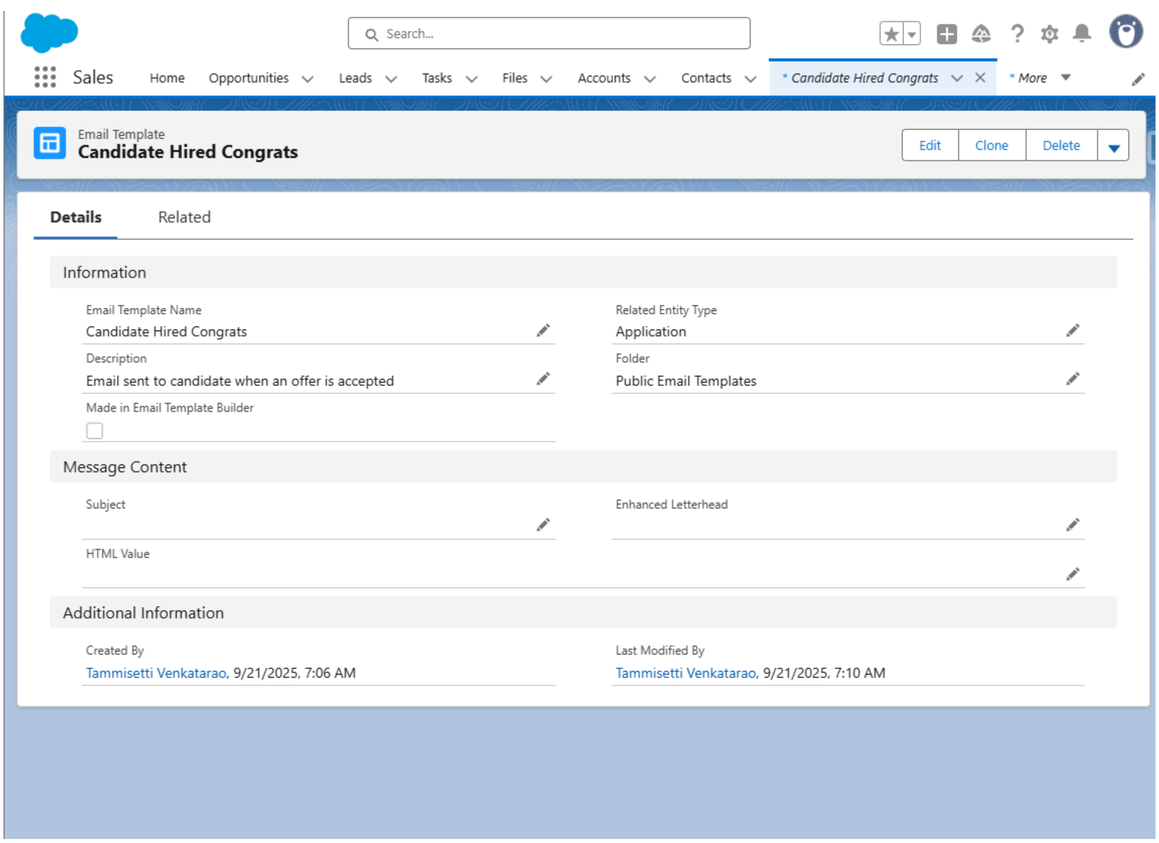
    Hi {!Application__c.Candidate__r.FirstName},

    Congratulations! Your application for {!Application__c.Job_Opening__r.Job_Title__c} has been successful.

    Your expected joining date: {!Application__c.Joining_Date__c}.

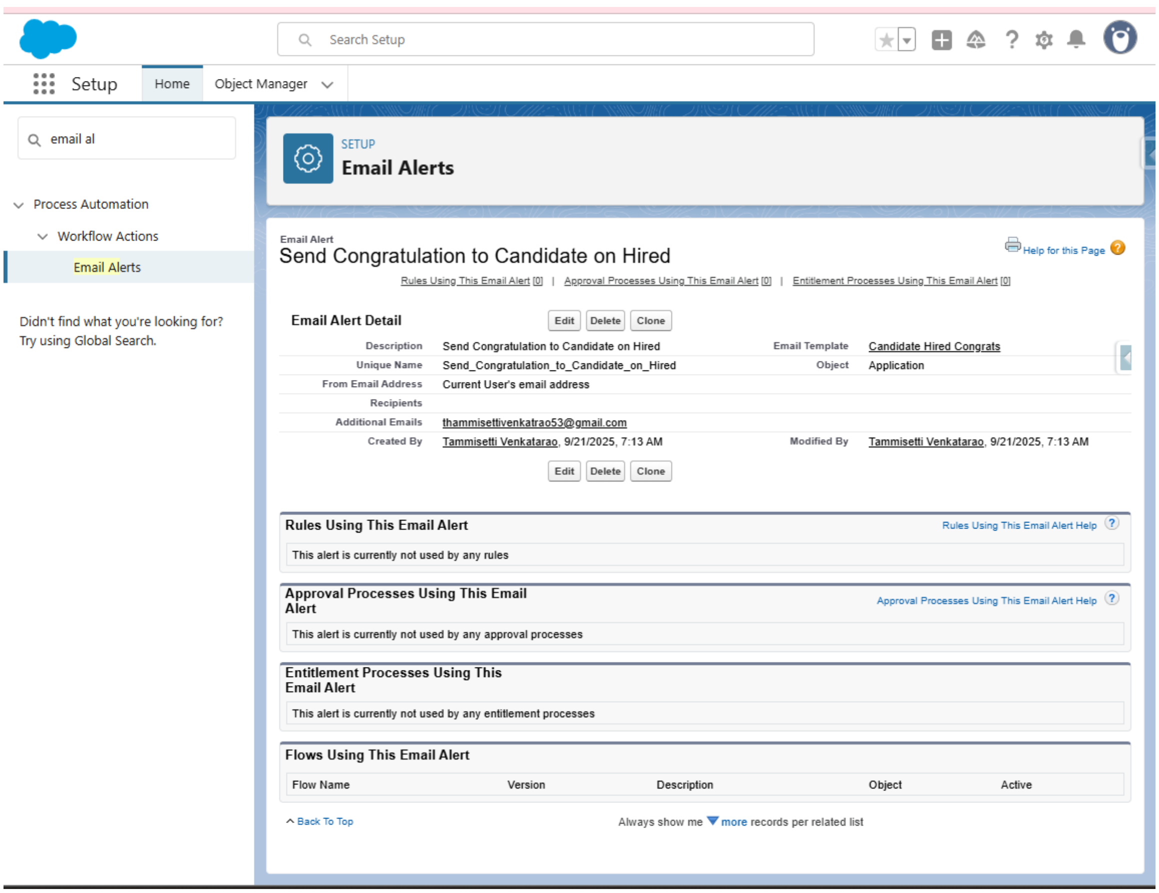
    HR Manager: {!User.FirstName} {!User.LastName}

6. Save.



Create an Email Alert to use in Flows

1. Setup → **Email Alerts** → **New Email Alert**.
2. Description: Send Congratulation to Candidate on Hired
3. Object: **Application**
4. Email Template: Candidate_Hired_Congrats
5. Recipient Type: Related Contact / Email Field → choose Candidate Email (or related Contact)
6. Save.

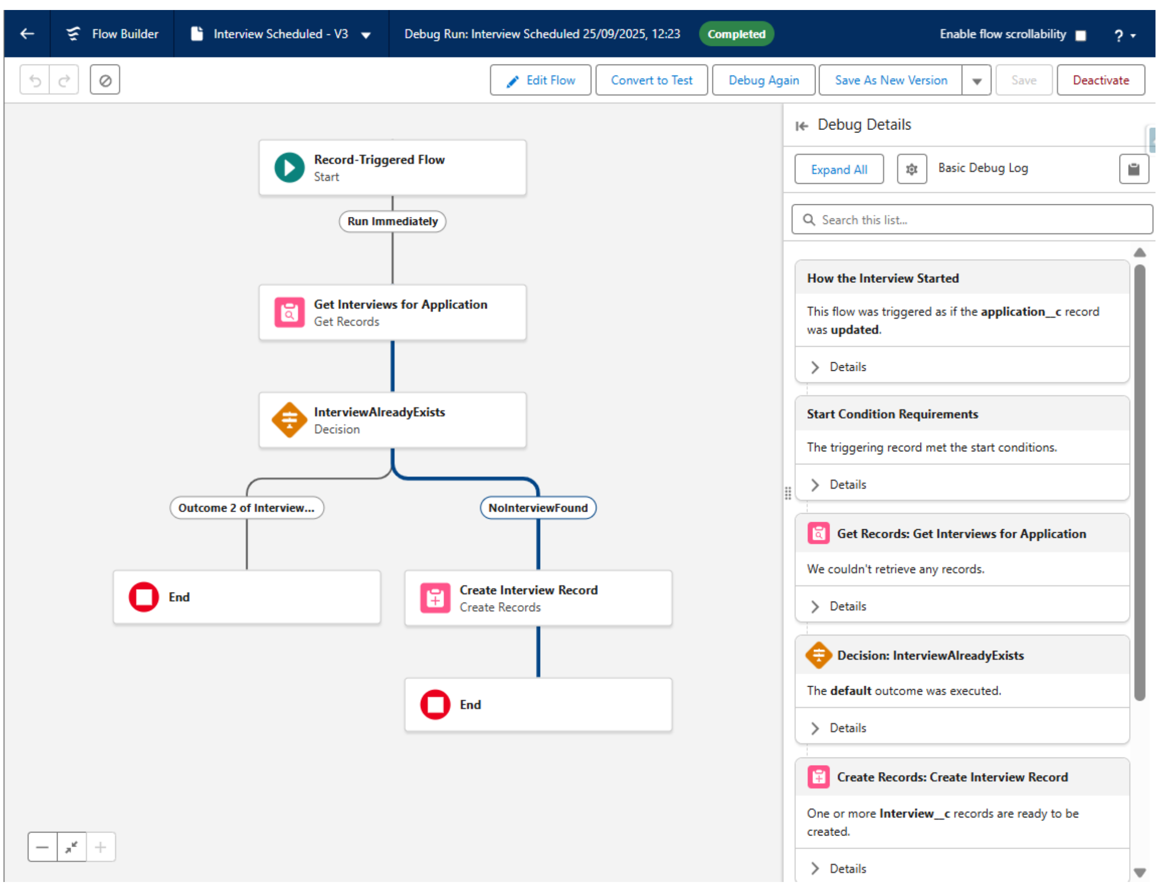


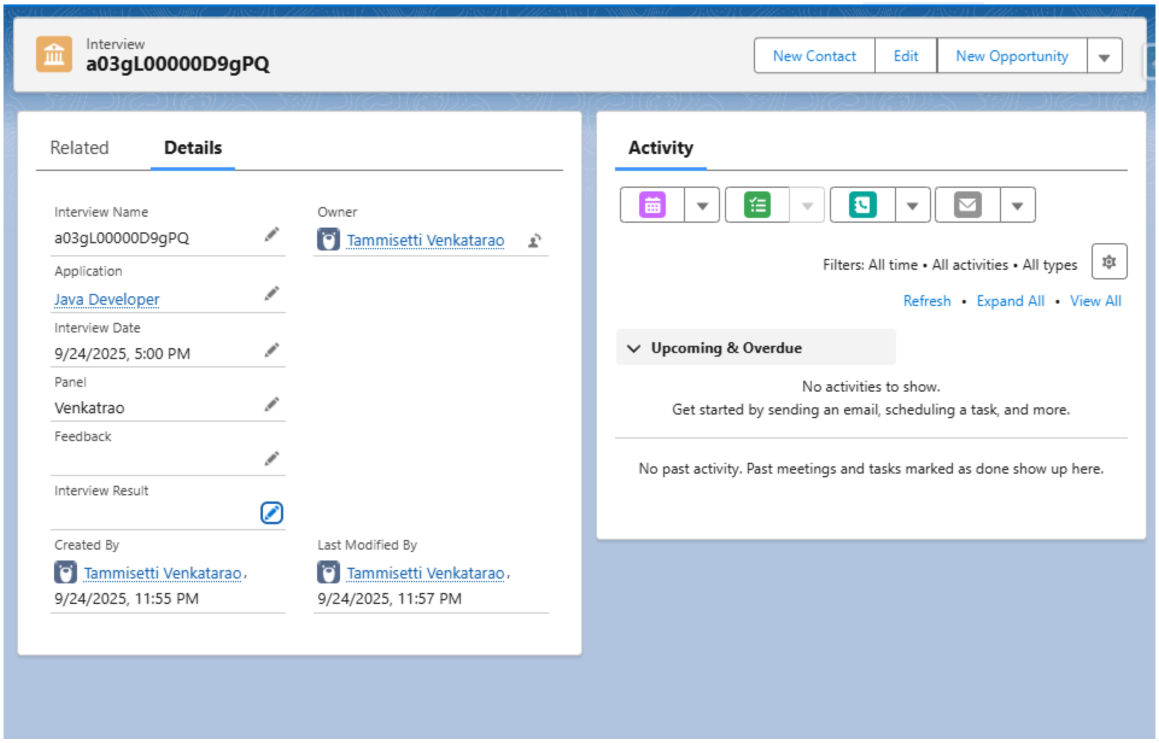
3) Record-Triggered Flow: Auto-create Interview when Stage = Interview Scheduled

1. Setup → **Flows** → **New Flow** → **Record-Triggered Flow**.
2. Object: **Application__c**. Trigger: **A record is updated**.

3. Entry Condition: Stage__c Equals Interview Scheduled. Optimize for: **Actions and Related Records**.
4. (Optional) Add **Decision**: InterviewAlreadyExists? to prevent duplicates.
5. Add **Create Records**: Object = **Interview__c**. Map fields:
    - Application__c = $Record.Id
    - Interview_Date__c = $Record.Interview_Date__c
    - Interview_Result__c = Scheduled
    - Panel__c = $Record.Panel__c
6. (Optional) Add **Send Email Alert** or **Send Custom Notification** to notify panel/recruiter.
7. Save → Name: AutoCreateInterview_On_InterviewScheduled → **Activate**.

Test: In Sandbox, update an Application to Stage = Interview Scheduled; verify Interview appears in related list and notification/email is received.

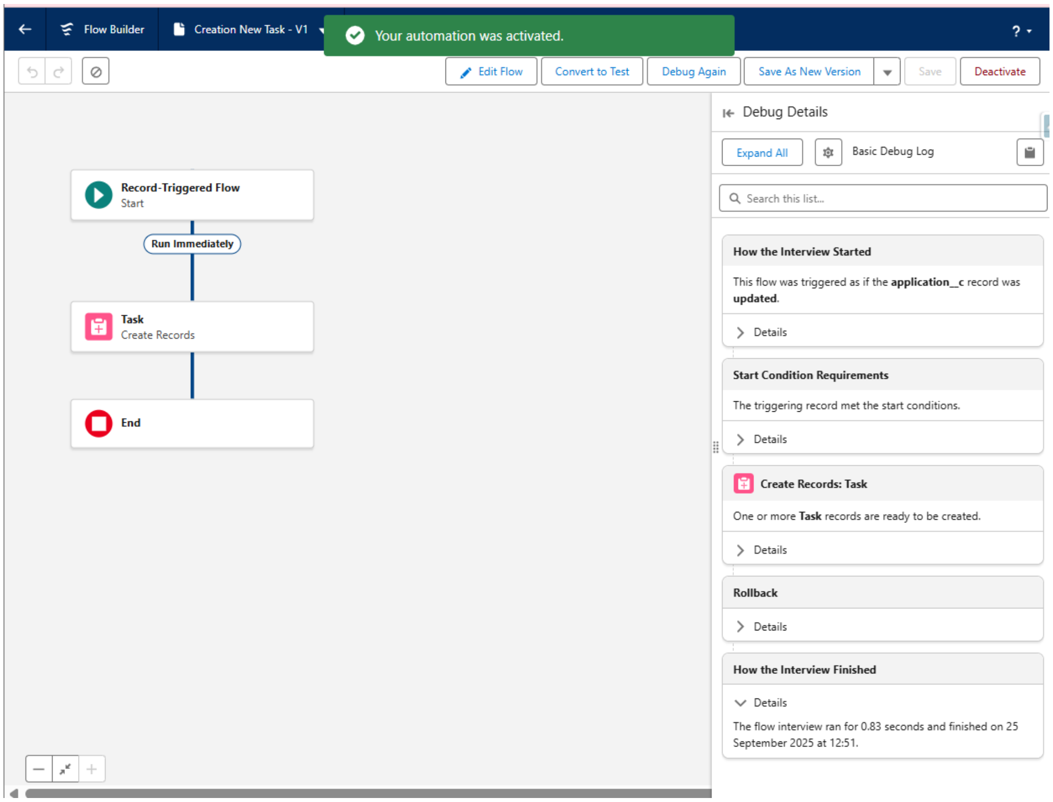


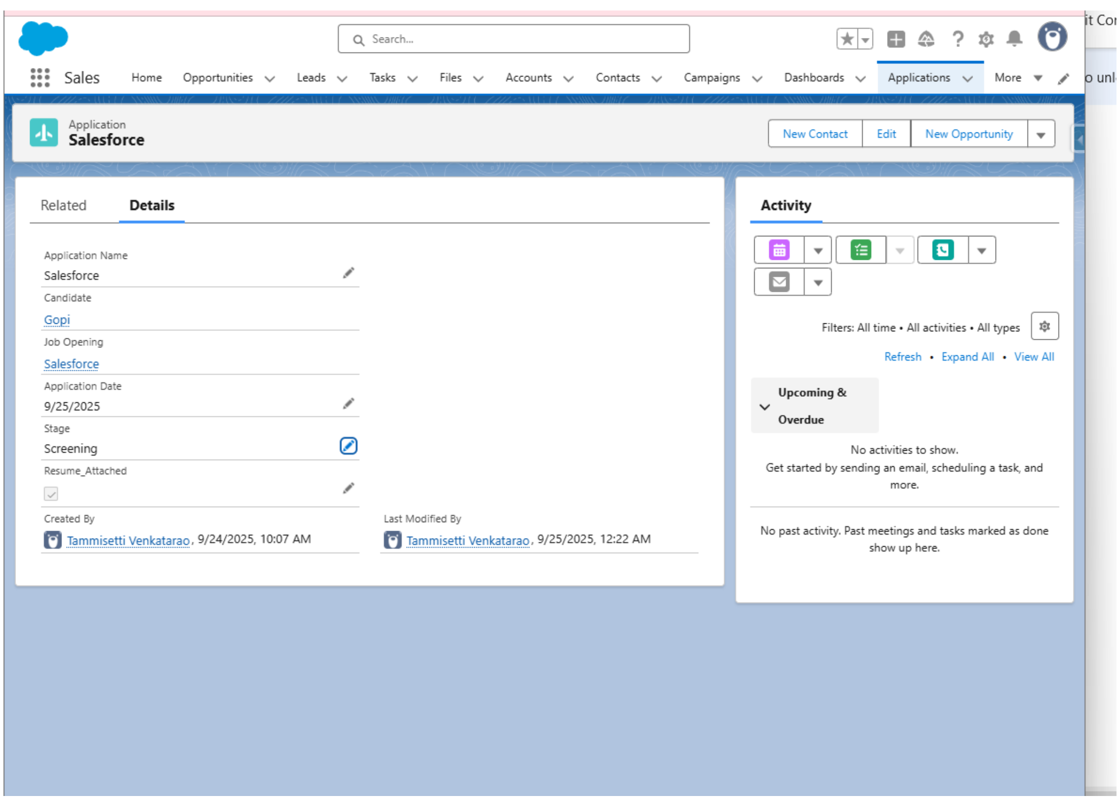


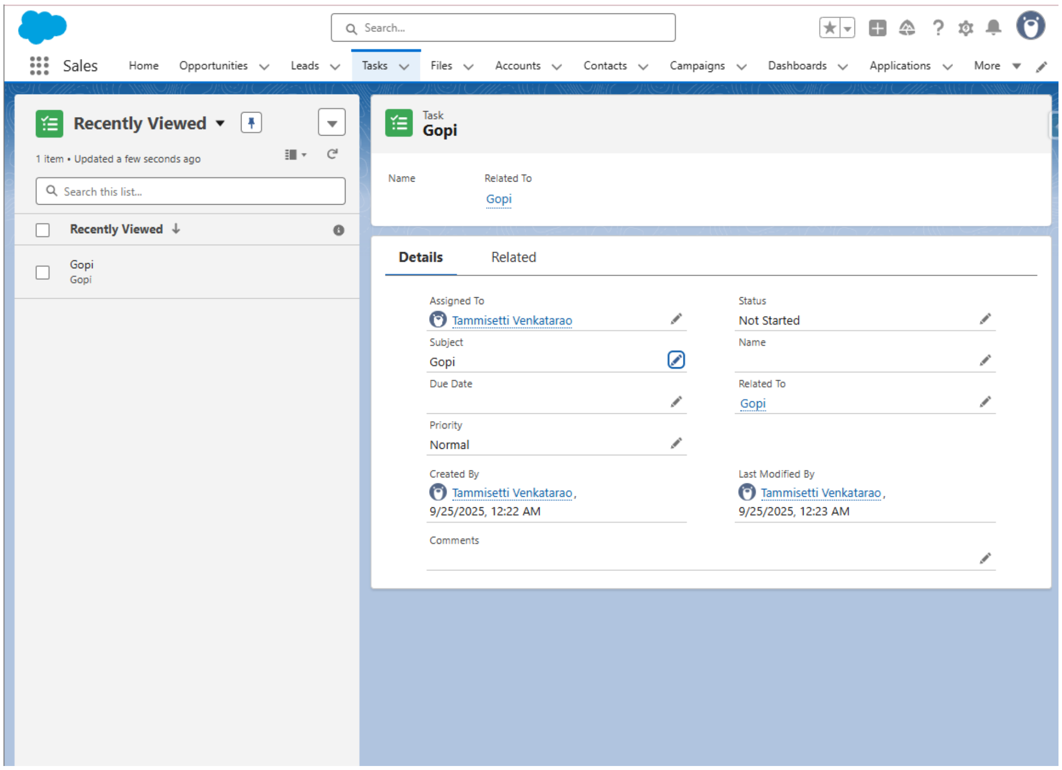
4) Record-Triggered Flow: Auto-create Task for Recruiter on Stage = Screening

1. Setup → Flows → New → Record-Triggered Flow.
2. Object: **Application__c** → Trigger when record is created or updated.
3. Entry Condition: Stage__c Equals Screening.
4. Add **Create Records** → Object: **Task**. Fields:
    - Subject: Screen Candidate - {!$Record.Candidate__r.Name}
    - WhoId / WhatId: set to Candidate or Application as appropriate
    - OwnerId: $Record.OwnerId (or specific recruiter)
    - Status: Not Started
    - Priority: Normal
5. Save & Activate.

Test: Change Stage to Screening → confirm Task appears under related Tasks.


Flow Builder
Creation New Task - V1
Your automation was activated.
Edit Flow
Convert to Test
Debug Again
Save As New Version
Save
Deactivate
Record-Triggered Flow
Start
Run Immediately
Task
Create Records
End
Debug Details
Expand All
Basic Debug Log
Search this list...
How the Interview Started
This flow was triggered as if the application__c record was updated.
Details
Start Condition Requirements
The triggering record met the start conditions.
Details
Create Records: Task
One or more Task records are ready to be created.
Details
Rollback
Details
How the Interview Finished
Details
The flow interview ran for 0.83 seconds and finished on 25 September 2025 at 12:51.

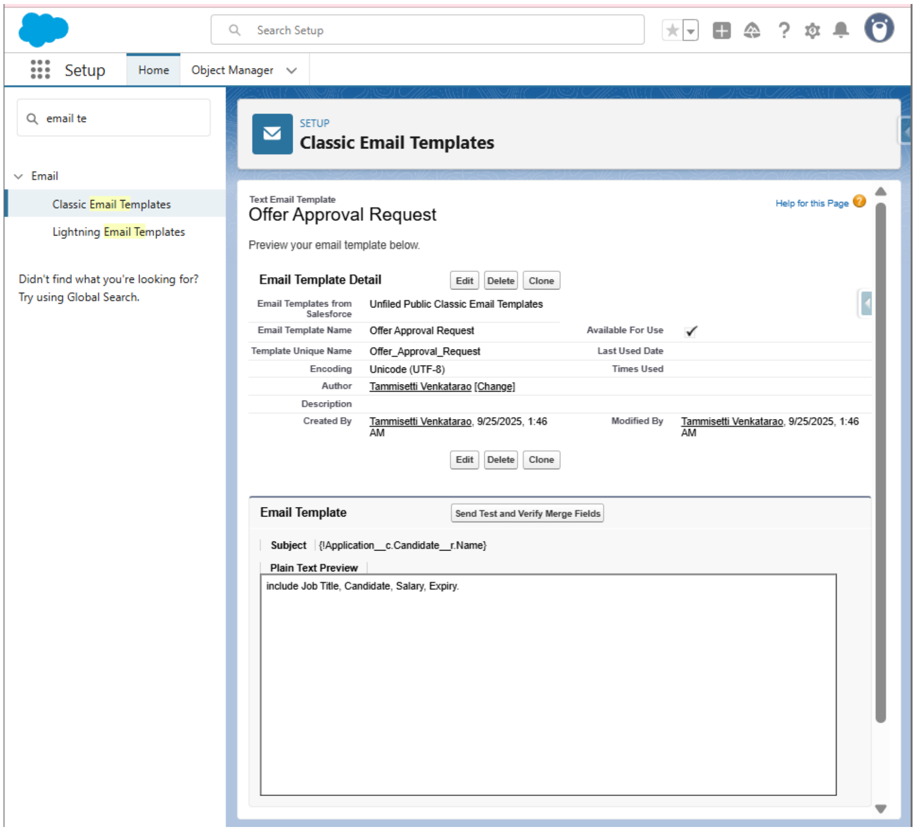

Search...
Sales
Home
Opportunities
Leads
Tasks
Files
Accounts
Contacts
Campaigns
Dashboards
Applications
More
Application
Salesforce
New Contact
Edit
New Opportunity
Related
Details
Application Name
Salesforce
Candidate
Gopi
Job Opening
Salesforce
Application Date
9/25/2025
Stage
Screening
Resume_Attached
Created By
Tammisetti Venkatarao, 9/24/2025, 10:07 AM
Last Modified By
Tammisetti Venkatarao, 9/25/2025, 12:22 AM
Activity
Filters: All time • All activities • All types
Refresh • Expand All • View All
Upcoming & Overdue
No activities to show.
Get started by sending an email, scheduling a task, and more.
No past activity. Past meetings and tasks marked as done show up here.



5) Approval Process: Offer Approval (Department Head)

Create Email Template (Offer Approval Request)

1. Setup → Email Templates → New Email Template.
   - Name: Offer_Approval_Request
   - Subject: Approval Required: Offer for {!Application__c.Candidate__r.Name}
   - Body: include Job Title, Candidate, Salary, Expiry.
2. Save.

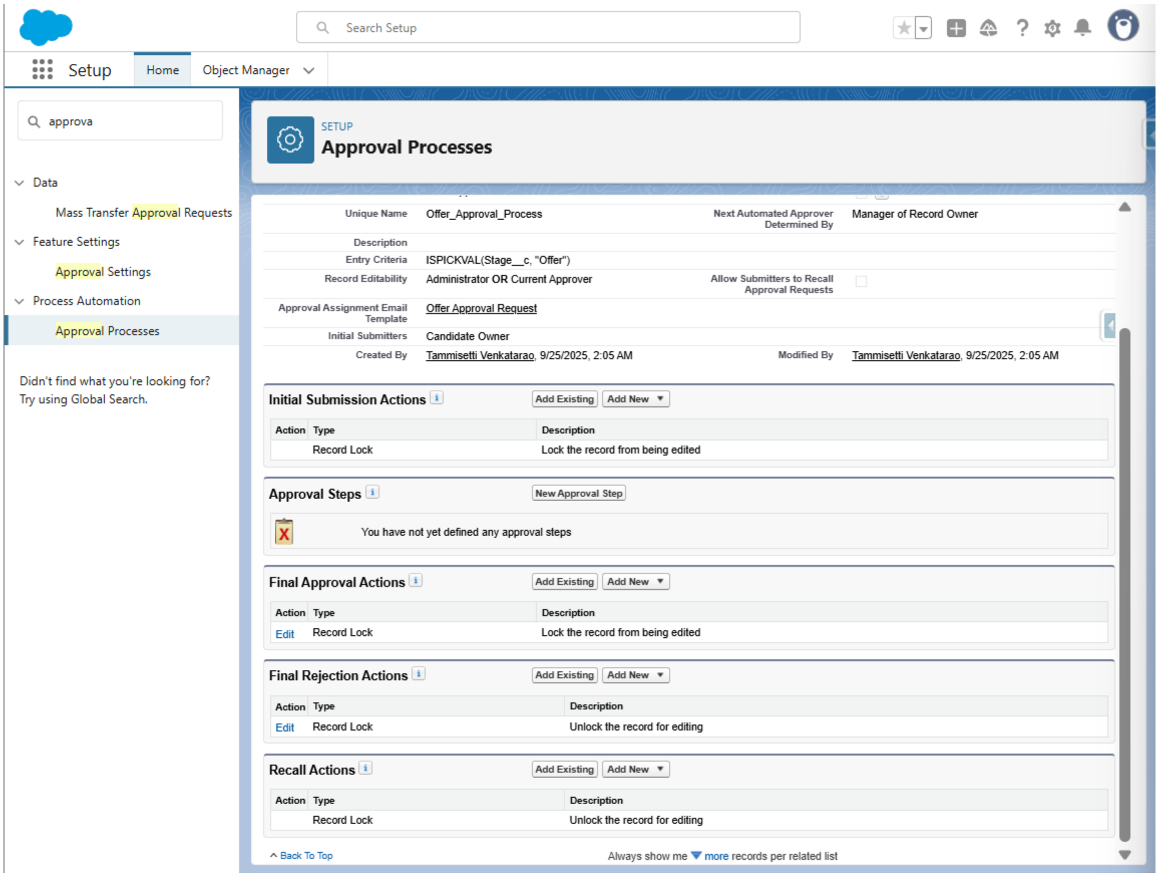


Create Approval Process

1. Setup → **Approval Processes** → Select object **Application** → **Create New Approval Process** → Standard Wizard.
2. Name: Offer Approval Process.
3. Entry Criteria: ISPICKVAL(Stage__c, "Offer") (or add threshold: AND(ISPICKVAL(Stage__c, "Offer"), Salary_Offered__c > 50000)).
4. Approver: Record Owner's Manager or a specified Department Head.
5. Initial Submitters: Application Owner.
6. Initial Submission Actions: Field Update → Approval_Status__c = Pending.
7. Final Approval Actions: Field Update → Offer_Status__c = Approved; Send Email Alert to Candidate.
8. Final Rejection Actions: Field Update → Offer_Status__c = Rejected; Send Email Alert.
9. Activate Approval Process.

Optional: Create a Record-Triggered Flow that submits the Application for approval automatically when Stage changes to Offer.

Test: Move Application Stage to Offer → approver receives email and can approve or reject; check field updates.

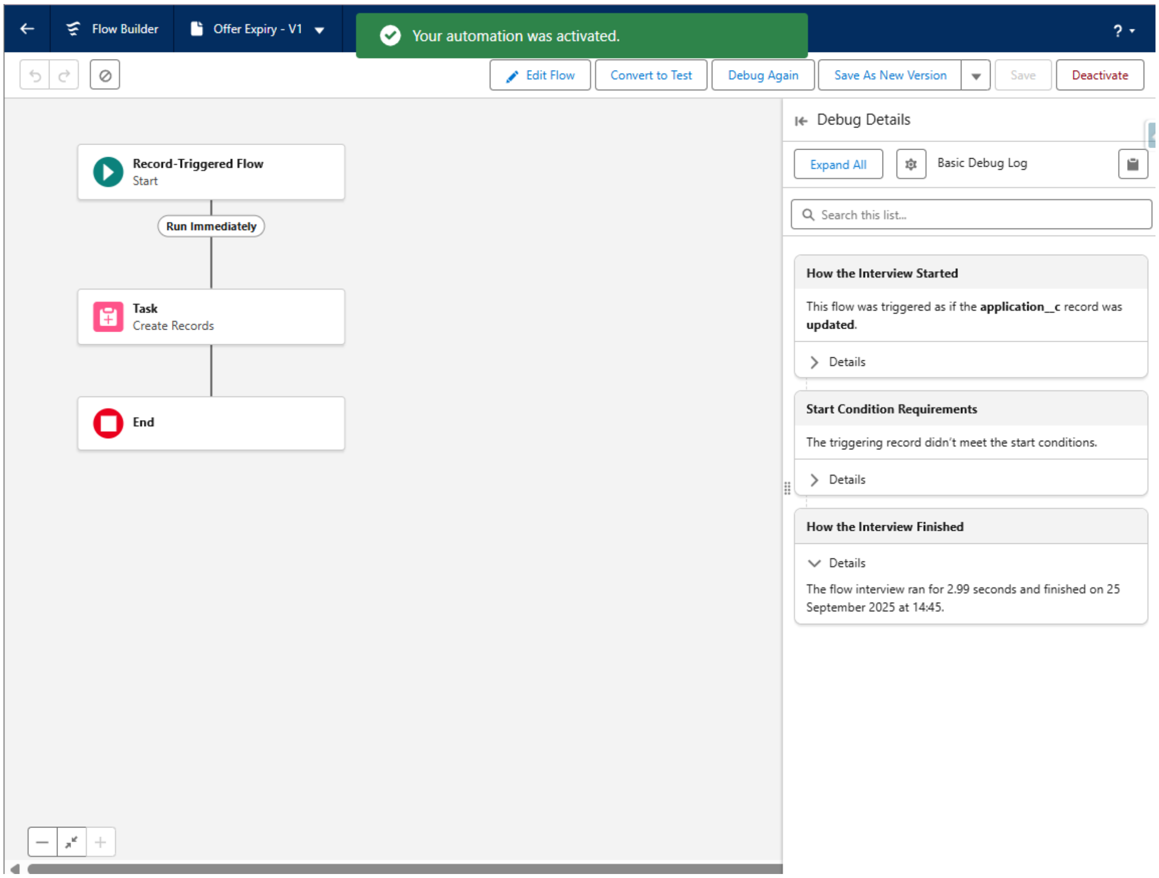


6) Scheduled Path in Record-Triggered Flow (Reminders / SLA)

Use case: Remind recruiter/candidate 3 days before Offer expiry or remind if no response.

1. Create Record-Triggered Flow on **Application__c** (trigger on create or update).
2. Add **Scheduled Path**: Run after Offer_Expiry_Date__c - 3 Days.
3. In scheduled path add **Decision**: If Stage__c = Offer AND Offer_Accepted__c = false, then **Create Task** or **Send Email Alert** to candidate & recruiter.
4. Save & Activate.

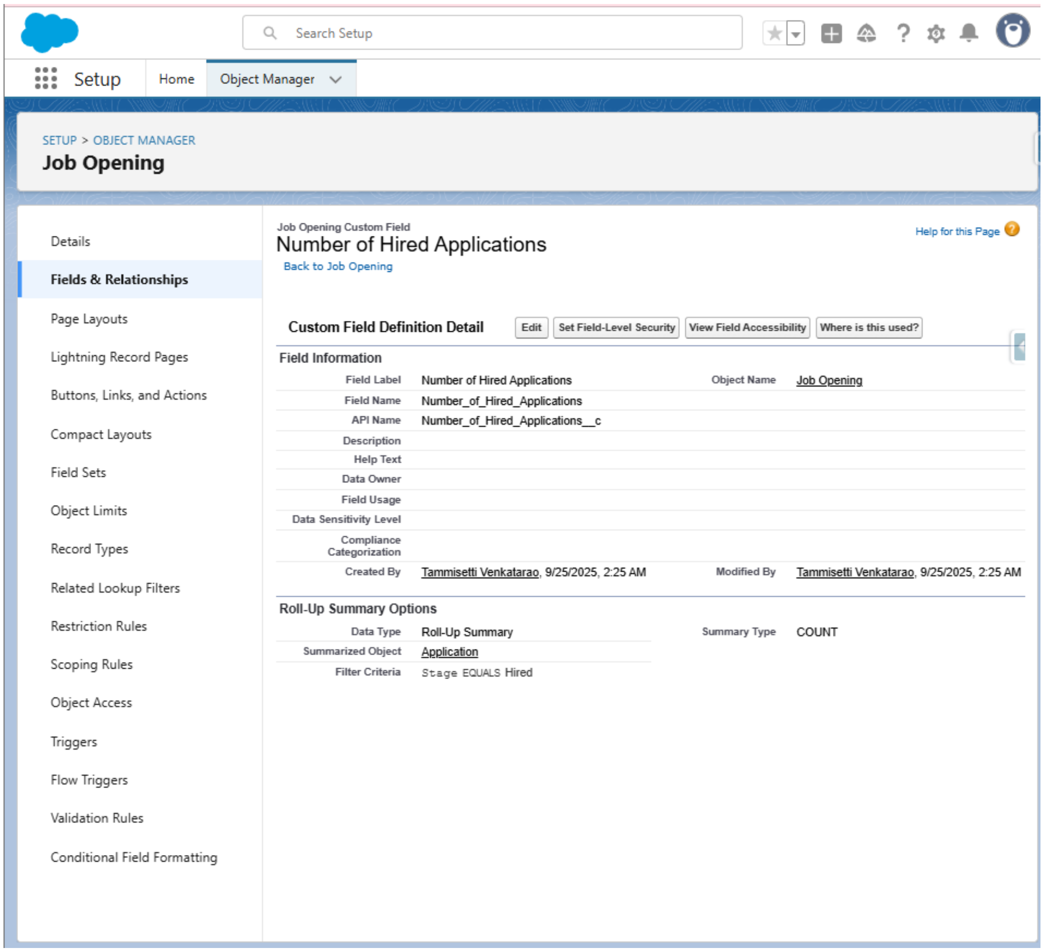
Test: Create an Application with Offer Expiry in near future and debug scheduled path or wait for execution.



7) Auto-update Job Opening Positions (Roll-up)

Option A — Roll-Up Summary (if Master-Detail exists)

1. If Application__c is master-detail to Job_Opening__c, create **Roll-Up Summary** on Job_Opening__c:
   - Summarized Object: Application__c
   - Roll-up Type: COUNT where Stage__c = Hired.



Test: Mark Application as Hired → Job Opening counters update.

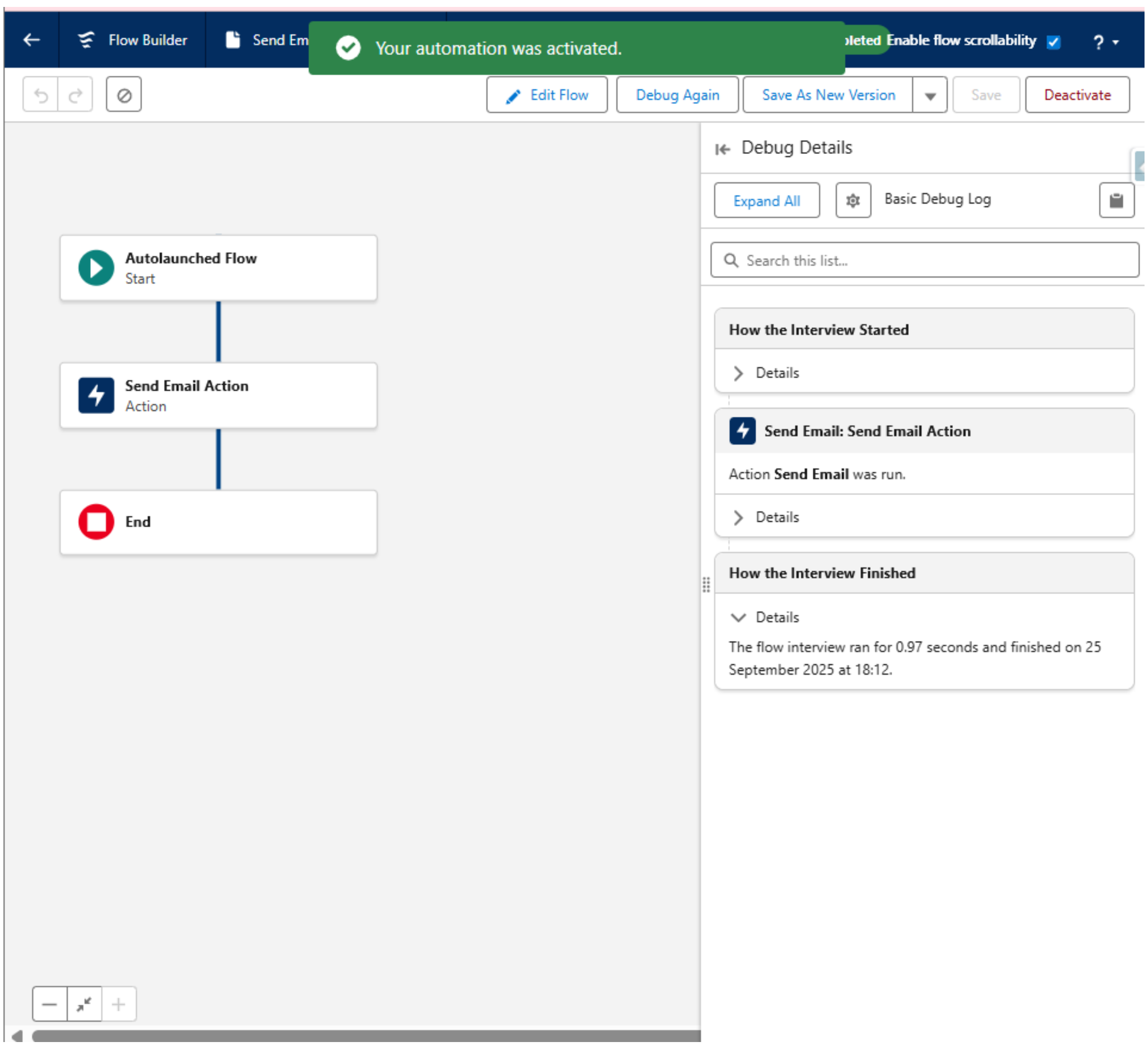
8) Auto-convert Leads to Candidates (incoming from LinkedIn / Site)

1. Create Record-Triggered Flow on **Lead** when Status = Closed - Converted or a custom checkbox Convert_to_Candidate__c = true.
2. Add **Create Records** → Object: **Candidate__c**. Map fields from Lead: FirstName, LastName, Email, Phone, Source = LinkedIn.
3. Optionally create **Application__c** to link Candidate → Job_Opening if Lead included interested job.
4. Update Lead to mark converted. Save & Activate.

Note: For robust lead conversion (account/contact/opportunity creation) consider Apex Database.LeadConvert or an invocable Apex class.

9) Subflows (Reusable) & Best Practices

- Create reusable subflows for repeated actions (e.g., Notify_Recruiter_Subflow to send email & custom notification).
- Name flows clearly and add descriptions (purpose, owner, inputs/outputs).
- Add **Fault Paths** on actions to log errors to a custom object (e.g., Flow_Error__c) for admin review.
- Avoid heavy synchronous processing; for bulk operations use scheduled flows or Batch Apex.
- Test flows with bulk updates to ensure governor limits are respected.



10) Send Custom Notifications (In-app)

1. Setup → **Notification Builder** → **Notification Types** → **New**.
   - Label: Recruitment_Update → Save.
2. In Flow add **Action** → **Send Custom Notification**, select Recruitment_Update, set title/body and recipient (e.g., recruiter or panel member).
3. Test: Notification shows in Salesforce bell icon and mobile app.

11) Test & Debug Flows

- Use **Debug** in Flow Builder to step through flow execution with sample input.
- Monitor **Paused and Failed Flow Interviews** (Setup → Paused & Failed Flow Interviews).
- Use **Debug Logs** (Setup → Debug Logs) especially when flows invoke Apex or platform events.
- Run bulk tests (update multiple records) to validate performance and bulk-safety.

12) Deployment of Flows & Automation

- In Sandbox: Create an **Outbound Change Set** → add Flows, Email Templates, Email Alerts, Approval Processes, Notification Types.
- Upload Change Set to Production. In Production, **Validate** and **Deploy**.
- Alternatively: use **SFDX / CI-CD** for version-controlled deployments.
- Post-deployment: run smoke tests (create sample records and exercise each automation path).

13) Monitoring & Maintenance

- Build an **Admin Dashboard** with Flow error counts, paused flows, and pending approvals.
- Document flow owners, inputs/outputs, and purpose in project documentation.
- Schedule periodic reviews to ensure automations reflect business needs.

- Keep a change log for all flow/config updates for audit/tracking.

14) Sample Deliverables for Phase 4

- List of Validation Rules (names, formulas, purpose).
- Flow names, diagrams (Visio/Miro) and descriptions.
- Email Templates and Email Alerts list with bodies.
- Approval Process specification and email templates.
- Custom Notification types and usage doc.
- Test cases and sandbox testing logs.
- Deployment Change Set or SFDX manifest.

15) Quick Checklist (perform in Sandbox first)

- Create and test Validation Rules.
- Create Email Templates and Email Alerts.
- Build and activate Record-Triggered Flows (Interview creation, Task creation, Submit for Approval).
- Build Scheduled Paths for reminders.
- Create Approval Process for Offers and test approval/rejection.
- Implement Roll-up/Flow to update Job Opening counts.
- Create Notification Types and test in-app/mobile notifications.
- Create Change Set and validate deployment.