

HireMe - Job Recruitment Training CRM

Phase 1: Problem Understanding & Industry Analysis

👉 Goal: Understand what we're building and why.

1. Requirement Gathering

Talk to stakeholders (HR Manager, Training Coordinator, Hiring Manager, Candidate).

Example requirements:

- Track job recruitment training programs for new hires.
 - Allow candidates/employees to enroll in training sessions.
 - Provide automated reminders, attendance tracking, and completion certificates.
 - Link training results with job performance and recruitment success rate.
 - Generate reports on training effectiveness, candidate progress, and ROI.
-

2. Stakeholder Analysis

- **Admin (you):** Manage CRM setup, configure objects, automation, and reports.
 - **HR Manager:** Defines training curriculum, ensures compliance, and measures effectiveness.
 - **Training Coordinator:** Schedules sessions, tracks attendance, uploads materials.
 - **Hiring Manager:** Monitors candidate training progress and evaluates readiness for roles.
 - **Candidate/Employee:** End-user, enrolls in training, completes assessments, and gets certified.
-

3. Business Process Mapping

Draw a flow:

Job Application → Candidate Selected → Training Assigned → Candidate Enrolls → Session Conducted → Attendance + Assessments → Training Completed → Feedback + Certificate → HR Manager Reviews Performance.

4. Industry-Specific Use Case Analysis

- In the recruitment industry, **training is crucial** to bridge the gap between candidate selection and job readiness.
 - Many industries (IT, Banking, Retail, BPO, Healthcare) require **structured onboarding and skill-based training**.
 - Compliance-related training (data security, workplace ethics, safety) is often **mandatory before employment confirmation**.
 - Real-time progress tracking and **personalized learning paths** are essential to reduce dropouts.
-

5. AppExchange Exploration

- Search for "Learning Management System (LMS)" or "Recruitment Training" apps on AppExchange.
- Examples: **Cornerstone OnDemand, Docebo, Salesforce LMS apps**.
- But for learning purposes, we'll build a **custom Job Recruitment Training CRM** with Salesforce objects, automation, and dashboards.



Industry Analysis

1. Current Problems in Recruitment Training

- Training programs are managed manually in spreadsheets/emails.
 - No centralized view of candidate enrollment, progress, or completion.
 - Difficult to measure ROI of training on hiring success.
 - Lack of personalized training schedules for different roles.
 - No automated reminders → high dropout rates.
-

2. Current Market Solutions

- **Learning Management Systems (LMS)**: Platforms like Moodle, SAP SuccessFactors, Cornerstone.

- **HR Tech Tools:** Workday, Oracle HCM, Zoho People – offer onboarding + training features.
 - **Recruitment CRMs:** Focus on hiring but lack deep training integration.
-

3. Market Trends

- **Skill-Based Hiring:** Recruiters emphasize training for role-specific skills.
 - **Blended Learning:** Mix of online (self-paced) + classroom training.
 - **Gamification:** Quizzes, leaderboards, and badges increase candidate engagement.
 - **AI-driven Personalization:** Adaptive training paths for different candidates.
 - **Compliance Training:** Mandatory certifications in regulated industries (finance, healthcare, IT).
-

4. Gaps in Current Systems

- Existing tools are either **too generic (LMS)** or **too recruitment-focused (ATS)**.
 - Lack of **integration** between job recruitment and training modules.
 - Limited **analytics on training effectiveness vs job performance**.
 - High **cost** of enterprise LMS tools for small-medium businesses.
-

5. Opportunity for Job Recruitment Training CRM

The custom CRM can solve these gaps by:

1. **Centralized Platform:** Integrates recruitment and training in one system.
2. **Automated Enrollment:** Assign training sessions to candidates automatically after hiring stage.
3. **Session Management:** Schedule classes, upload materials, track attendance.
4. **Assessment & Certification:** Online tests + auto-generated certificates.
5. **Progress Tracking:** Dashboards for candidates, HR, and hiring managers.
6. **Performance Reports:** Measure training impact on candidate success in job roles.

👉 Phase 2: Org Setup & Configuration

◆ 1. Salesforce Edition & Org Setup

- Selected **Salesforce Developer Edition Org** for implementation.
- Connected Org with **VS Code (SFDX Project Setup)** for version control & deployment.
- Verified basic org information:
 - Company Profile: **HireMe Recruitment Pvt. Ltd.**
 - Default Locale: **English (India)**
 - Default Currency: **INR (₹)**

The screenshot shows the 'Company Information' page in the Salesforce setup interface. At the top, there's a blue header bar with the 'SETUP' button and the 'Company Information' section title. Below the header, the organization's name is displayed as 'HireMe Recruitment Pvt. Ltd.' with a small info icon. A note says 'The organization's profile is below.' Below this, there are tabs for 'User Licenses [10+]', 'Permission Set Licenses [10+]', 'Feature Licenses [11]', and 'Usage-based Entitlements [10+]'. The main content area is titled 'Organization Detail' and contains two columns of settings. The left column includes fields like Organization Name (HireMe Recruitment Pvt. Ltd.), Primary Contact (OrgFarm EPIC), Division, Address (United States), Fiscal Year Starts In (January), Activate Multiple Currencies (checked), Enable Data Translation (unchecked), Newsletter (checked), Admin Newsletter (checked), Hide Notices About System Maintenance (unchecked), Hide Notices About System Downtime (unchecked), and Locale Formats (ICU). The right column includes fields like Phone, Fax, Default Locale (Hindi (India)), Default Language (English), Default Time Zone (GMT+05:30) India Standard Time (Asia/Kolkata), Corporate Currency (Indian Rupee), Used Data Space (1.9 MB (39%) [View]), Used File Space (803 KB (4%) [View]), API Requests, Last 24 Hours (131 (15,000 max)), Streaming API Events, Last 24 Hours (0 (10,000 max)), Restricted Logins, Current Month (0 (0 max)), Salesforce.com Organization ID (00DNS00000Ggos2), Organization Edition (Developer Edition), and Instance (IND56). There are also 'Edit' and 'Currency Setup' buttons at the top of the detail section.

◆ 2. Company Profile Setup

- Updated Company Information → Company Name = **HireMe CRM**.
- Fiscal Year = **Standard (Jan–Dec)**.
- Configured Business Hours:
 - Default = **Mon–Sat, 09:00 AM – 7:00 PM IST**
 - Holidays = Added **National & Regional holidays** (placeholder list).

Business Hours Detail

Business Hours Name	Default	Time Zone
Business Hours	Sunday Monday Tuesday Wednesday Thursday Friday Saturday	24 Hours 9:00 AM to 7:00 PM 9:00 AM to 7:00 PM
Active <input checked="" type="checkbox"/>		Default Business Hours <input checked="" type="checkbox"/>
Created By	OrgFarm EPIC	Created On 3/24/2025, 8:39 PM
Last Modified By	Prabhav Kalage	Modified On 9/21/2025, 3:57 AM

Holidays

Holiday Name	Description	Date and Time
Independence Day		8/15/2026 All Day Edit
Republic Day		1/26/2026 All Day Edit

◆ 3. User Setup & Licenses

Created initial project users with **Standard Salesforce Licenses**:

- CEO (System Administrator)
- HR Head
- Recruitment Manager
- Trainer
- Recruiter

◆ 4. Role Hierarchy

Configured Role Hierarchy for controlled visibility:

```
graph TD
```

```
A[CEO] --> B[HR Head]
```

```
B --> C[Recruitment Manager]
```

```
C --> D[Trainer]
```

```
D --> E[Recruiter]
```

Creating the Role Hierarchy

You can build on the existing role hierarchy shown on this page. To insert a new role, click **Add Role**.

Your Organization's Role Hierarchy

HireMe Recruitment Pvt. Ltd.

- Add Role
- **CEO** Edit | Del | Assign
- Add Role
- **CFO** Edit | Del | Assign
- Add Role
- **COO** Edit | Del | Assign
- Add Role
- **HR Head** Edit | Del | Assign
- Add Role
- **Recruitment Manager** Edit | Del | Assign
- Add Role
- **Trainer** Edit | Del | Assign
- Add Role
- **Recruiter** Edit | Del | Assign
- Add Role

Show in tree view

- **CEO → Full visibility**
- **HR Head → Access to all Recruitment Managers' records**
- **Recruitment Manager → Access to Trainers & Recruiters' records**
- **Trainer → Access to Recruiters' assigned candidates**
- **Recruiter → Access only to their own candidate/job records**

◆ 5. Organization-Wide Defaults (OWD)

- **Position_c → Private (roles/visibility controlled)**
- **Job_Posting_c → Private (linked to Position, synced externally)**
- **Employment_Website_c → Controlled by Parent (linked with Job Posting)**
- **Job_Application_c → Private (central workflow node)**
- **Candidate_c → Private (submitted via Job Application)**
- **Review_c → Controlled by Parent (evaluations tied to Job Applications)**

👉 **Grant Access Using Hierarchies → Enabled**

Sharing Settings

Organization-Wide Sharing Defaults Edit

Edit your organization-wide sharing defaults below. Changing these defaults will cause all sharing rules to be recalculated. This could require significant system resources and time depending on the amount of data in your organization. Setting an object to Private makes records visible to record owners and those above them in the role hierarchy, and access can be extended using sharing rules.

Object	Default Internal Access	Default External Access	Grant Access Using Hierarchies
Lead	Public Read/Write/Transfer	Private	<input checked="" type="checkbox"/>
Account and Contract	Public Read/Write	Private	<input checked="" type="checkbox"/>
Order	Controlled by Parent	Controlled by Parent	<input checked="" type="checkbox"/>
Contact	Controlled by Parent	Controlled by Parent	<input checked="" type="checkbox"/>
Asset	Controlled by Parent	Controlled by Parent	<input checked="" type="checkbox"/>
Opportunity	Public Read/Write	Private	<input checked="" type="checkbox"/>
Case	Public Read/Write/Transfer	Private	<input checked="" type="checkbox"/>
Campaign	Public Full Access	Private	<input checked="" type="checkbox"/>
Campaign Member	Controlled by Campaign	Controlled by Campaign	<input checked="" type="checkbox"/>
User	Public Read Only	Private	<input checked="" type="checkbox"/>
Individual	Public Read/Write	Private	<input checked="" type="checkbox"/>
Voice Call	Private	Private	<input checked="" type="checkbox"/>
Activity	Private	Private	<input checked="" type="checkbox"/>
Calendar	Hide Details and Add Events	Hide Details and Add Events	<input checked="" type="checkbox"/>

◆ 6. Login & Security Policies

6.1 Trusted IP Ranges (Network Access)

Start IP End IP Description

47.11.15.0 47.11.15.255 Primary Office IP Range

Network Access

Trusted IP Range Edit

Enter the range of valid IP addresses from which user logins are trusted. Users logging in from trusted IP addresses are not asked to activate their computers and may use their user password instead of a security token to log in to the API or a desktop client such as Connect for Outlook, Connect Offline, Connect for Office, Connect for Lotus Notes, or the Data Loader.

Please specify IP range	
Start IP Address	47.11.15.0
Description	Primary Office IP Range
End IP Address	47.11.15.255

6.2 Session Settings

- Session Timeout: **30 minutes**
- Lock sessions to originating IP: **Disabled (kept flexible for remote users)**

Session Settings

Set the session security and session expiration timeout for your organization.

Session Timeout

Timeout Value: 30 minutes

Disable session timeout warning popup
 Force logout on session timeout

Session Settings

Lock sessions to the IP address from which they originated
 Lock sessions to the domain in which they were first used
 Terminate all of a user's sessions when an admin resets that user's password
 Force relogin after Login-As-User
 Require HttpOnly attribute
 Use POST requests for cross-domain sessions
 Enforce login IP ranges on every request
 When embedding a Lightning application in a third-party site, use a session token instead of a session cookie.

Extended use of IE11 with Lightning Experience

EXTENDED USE OF IE11 WITH LIGHTNING EXPERIENCE HAS NOW ENDED
AS OF DECEMBER 31, THE EXTENDED PERIOD HAS ENDED, AND USE OF INTERNET EXPLORER 11 (IE 11) WITH LIGHTNING EXPERIENCE IS NO LONGER SUPPORTED. ISSUES WITH PERFORMANCE OR FUNCTIONALITY THAT AFFECT ONLY IE 11 WILL NOT BE FIXED. PLEASE SWITCH TO A SUPPORTED BROWSER.

6.3 Login Hours (Profiles)

Profiles

Profile Overview > Login Hours

Login Hours

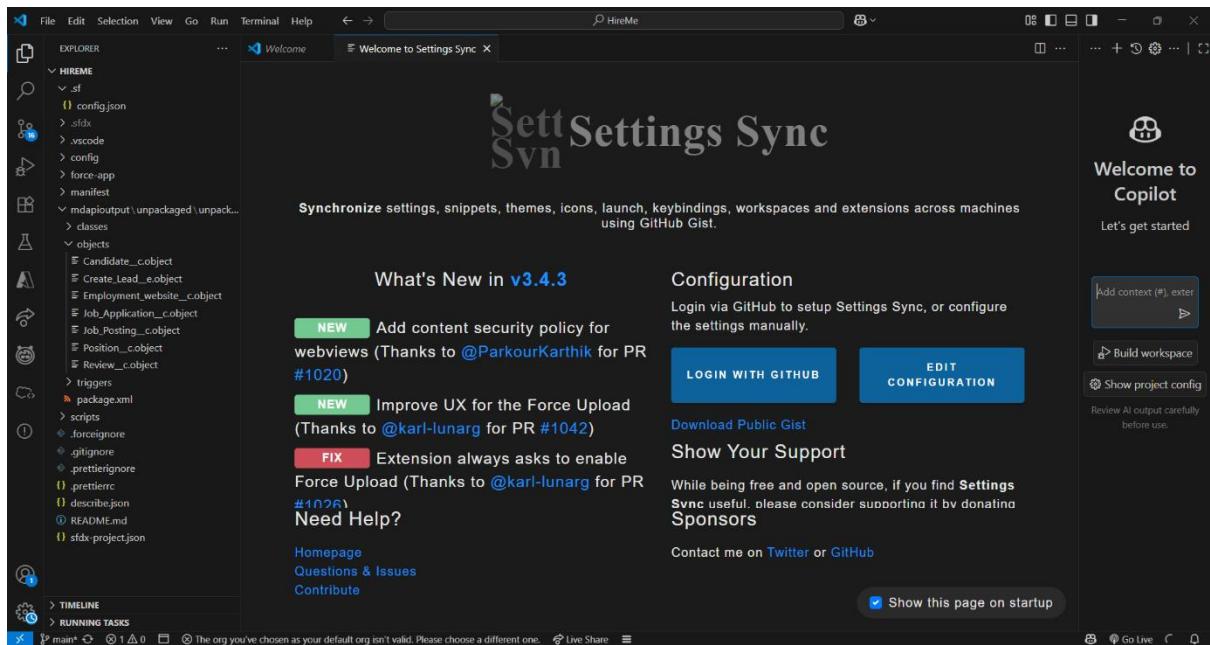
All times are in (GMT+0530) India Standard Time (Asia/Kolkata)

Day	Start Time	End Time	Action
Sunday	-None--	-None--	Clear times
Monday	9:00 AM	5:00 PM	Clear times
Tuesday	9:00 AM	5:00 PM	Clear times
Wednesday	9:00 AM	5:00 PM	Clear times
Thursday	9:00 AM	5:00 PM	Clear times
Friday	9:00 AM	5:00 PM	Clear times
Saturday	-None--	-None--	Clear times

[Clear all times](#)

◆ 7. Deliverables Produced

- Role Hierarchy Diagram
- OWD & Access Rules Documentation
- Login & Security Policy Document
- Org Setup Verification Report



📌 Phase 2 Status: Completed Successfully ✅

Next Step → Phase 3: Data Modeling & Object Configuration (Custom Objects, Fields, Relationships).

👉 Phase 3: Data Modeling & Relationships

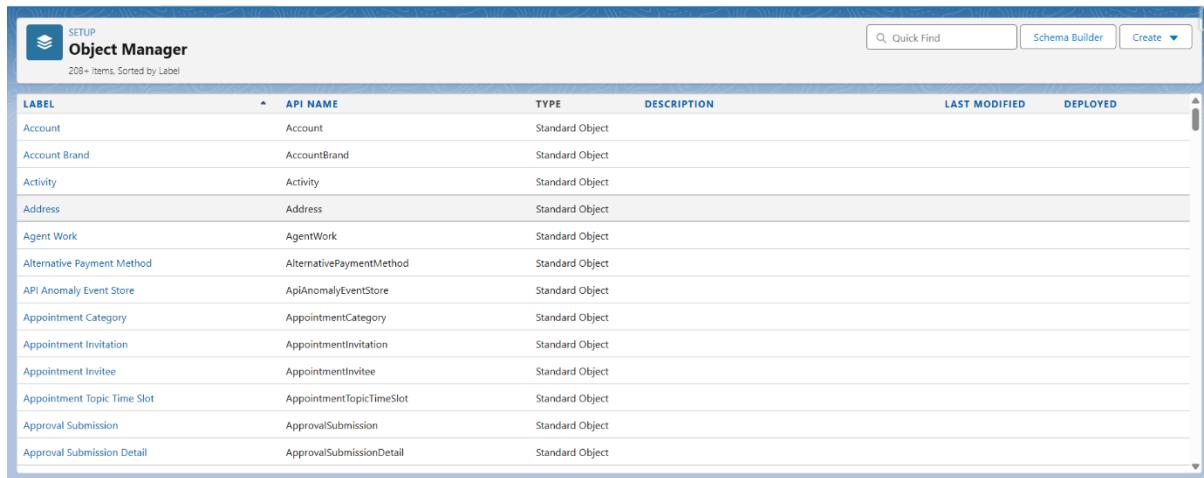
◆ 1. Standard & Custom Objects

• Standard Objects Used

- **User** (mapped to Recruiter, HR Head, Reviewer roles)
- **Contact** (candidate information reference, if needed)

• Custom Objects Created

1. **Position__c** → Represents roles within the organization
2. **Job_Posting__c** → Advertisements linked to positions
3. **Employment_Website__c** → External job boards/platforms
4. **Candidate__c** → Applicant records
5. **Job_Application__c** → Central application object (junction between Candidate & Position)
6. **Review__c** → Stores evaluations, decisions, and next steps



The screenshot shows the Salesforce Object Manager interface. At the top, there's a header with a 'SETUP' button, the 'Object Manager' title, and a '208+ Items. Sorted by Label' message. To the right are 'Quick Find', 'Schema Builder', and 'Create' buttons. Below the header is a table with the following columns: LABEL, API NAME, TYPE, DESCRIPTION, LAST MODIFIED, and DEPLOYED. The table lists various standard and custom objects:

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Account	Account	Standard Object			
Account Brand	AccountBrand	Standard Object			
Activity	Activity	Standard Object			
Address	Address	Standard Object			
Agent Work	AgentWork	Standard Object			
Alternative Payment Method	AlternativePaymentMethod	Standard Object			
API Anomaly Event Store	ApiAnomalyEventStore	Standard Object			
Appointment Category	AppointmentCategory	Standard Object			
Appointment Invitation	AppointmentInvitation	Standard Object			
Appointment Invitee	AppointmentInvitee	Standard Object			
Appointment Topic Time Slot	AppointmentTopicTimeSlot	Standard Object			
Approval Submission	ApprovalSubmission	Standard Object			
Approval Submission Detail	ApprovalSubmissionDetail	Standard Object			

◆ 2. Fields

Created **custom fields** on each object:

- **Position__c**
 - Position Title (Text)
 - Department (Picklist: Tech, HR, Marketing, Sales, etc.)
 - Status (Picklist: Open, Closed, On Hold)
- **Job_Posting__c**
 - Posting Title (Text)
 - Posting Date (Date)
 - Expiry Date (Date)
 - Status (Picklist: Active, Inactive)
- **Employment_Website__c**
 - Website Name (Text)
 - Website URL (URL)
 - Integration Status (Picklist: Connected, Not Connected)
- **Candidate__c**
 - Full Name (Text)
 - Email (Email)
 - Phone (Phone)
 - Resume (File)
- **Job_Application__c**
 - Application Date (Date)
 - Status (Picklist: Submitted, In Review, Shortlisted, Rejected)
 - Source (Picklist: Direct, Website, Referral)
- **Review__c**
 - Review Date (Date)
 - Reviewer (Lookup → User)

- Review Comments (Long Text)
- Result (Picklist: Selected, On Hold, Rejected)

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Candidate ID	Candidate_ID_c	Lookup(Candidate)		✓
Count Of review	Count_of_review_c	Roll-Up Summary (COUNT Review)		✗
Cover letter	Cover_letter_c	Long Text Area(32768)		✗
Created By	CreatedById	Lookup(User)		✗
Currency	CurrencyIsoCode	Picklist		✗
Job Application ID	Name	Auto Number		✓
Last Modified By	LastModifiedById	Lookup(User)		✗
Max Review Summary	Max_Review_Summary_c	Roll-Up Summary (MAX Review)		✗
Min Review Rating	Min_Review_Rating_c	Roll-Up Summary (MIN Review)		✗

◆ 3. Record Types

- **Position_c**
 - Tech Roles
 - Non-Tech Roles
- **Job_Application__c**
 - Fresher Application
 - Experienced Application

RECORD TYPE LABEL	DESCRIPTION	ACTIVE	MODIFIED BY
Non-Tech Roles		✓	Prabhav Kalage, 9/21/2025, 5:48 AM
Tech Roles		✓	Prabhav Kalage, 9/21/2025, 5:47 AM

◆ 4. Page Layouts

- Configured separate layouts for:

- **Recruiters** (Candidate intake-focused fields)
- **Hiring Reviewer** (Review-specific fields)
- **HR Head** (Summary view of Position, Job Posting, Applications)

The screenshot shows the Salesforce Setup interface for managing object layouts. The 'Job Application' object is selected. The 'Page Layouts' tab is active. On the left, a sidebar lists various layout configuration options like Details, Fields & Relationships, Page Layouts, and Compact Layouts. The main area shows a detailed view of the 'Page Layout Properties' for the 'Job Application' object. It includes sections for 'Information' (Header visible on edit only) and 'System Information' (Header visible on edit only). The 'Information' section contains fields such as Job Application ID (Value: GEN-2004-001234), Status (Value: Sample Text), Cover letter (Value: Sample Text), Position Name (Value: Sample Text), Candidate ID (Value: Sample Text), Total Number Of Review (Value: 14.446), Sum Of Review Rating (Value: 83.053), Min Review Rating (Value: 57.666), Max Review Summary (Value: 64.573), Count Of review (Value: 65.800), and Position & candidateID (Value: Sample Text). The 'System Information' section includes Created By (Value: Sample Text), Last Modified By (Value: Sample Text), and Owner (Value: Sample Text). At the bottom right, there is a 'Show desktop' button.

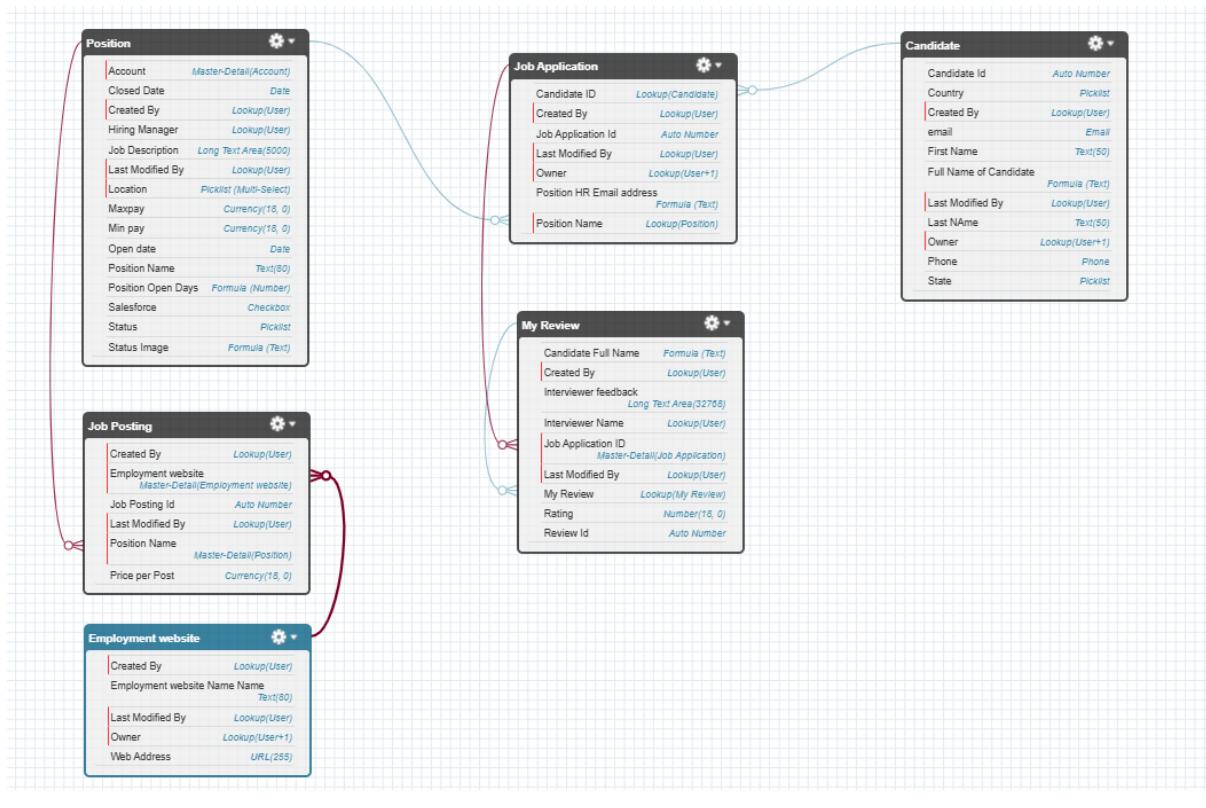
◆ 5. Compact Layouts

- Defined compact layouts for quick reference:
 - **Candidate__c** → Name, Email, Phone, Application Status
 - **Job_Application__c** → Candidate, Position, Status, Date

The screenshot shows the Candidate Record detail view for 'CA-00001'. The 'Details' tab is selected. The page layout includes sections for 'Personal Information Section' and 'Address Information Section'. In the 'Personal Information Section', fields include Candidate ID (Value: CA-00001), Country (Value: Sample Text), GState (Value: Sample Text), Full Name (Value: Shivani Kalage), Trailhead Id (Value: Sample Text), Link of trailhead (Value: Sample Text), First Name (Value: Shivani), Last Name (Value: Kalage), Email (Value: yogitadhu232male@gmail.com), and Phone (Value: (836) 728-8398). In the 'Address Information Section', fields include Street (Value: At Post Sasan Ramapur Ta. Daryapur), City (Value: Amraoti), State (Value: Maharashtra), and Postal Code (Value: 444706). At the bottom, there are navigation links for Omni-Channel, List View, Flow, and History.

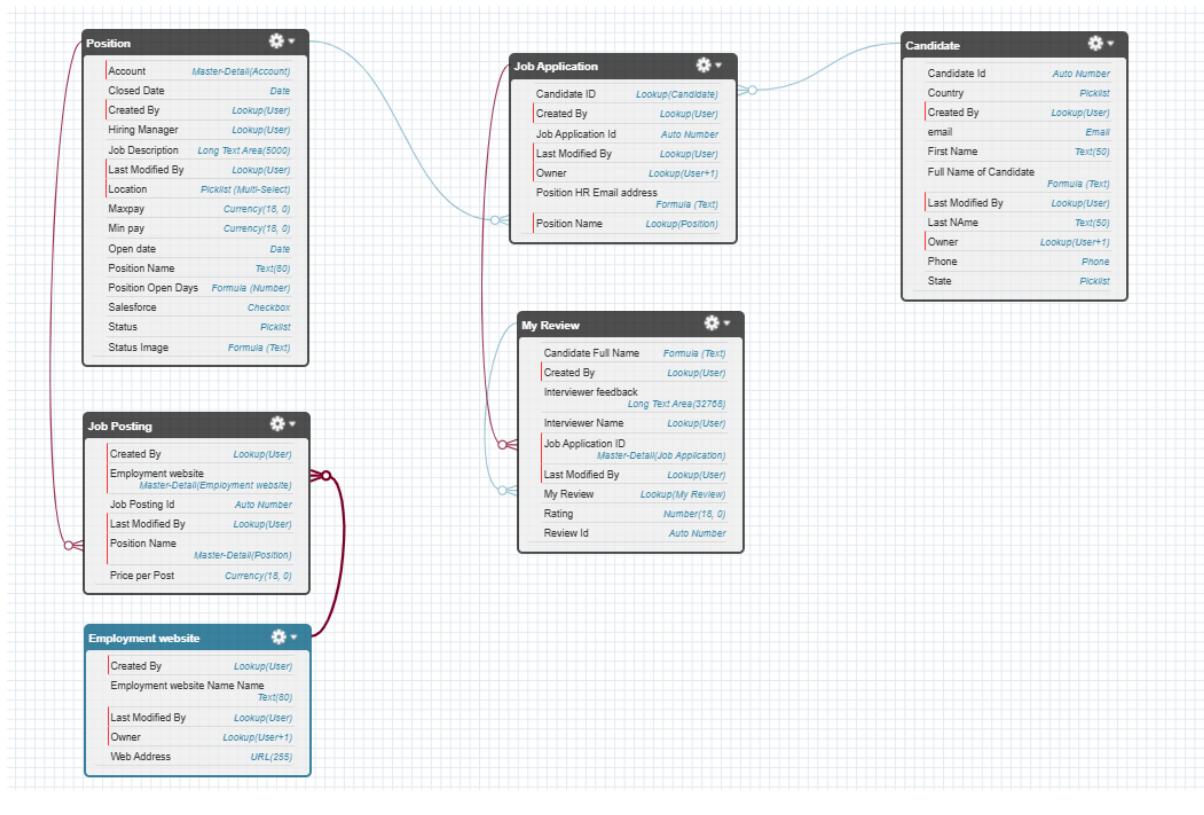
◆ 6. Schema Builder

- Visualized the **relationships** among all objects:



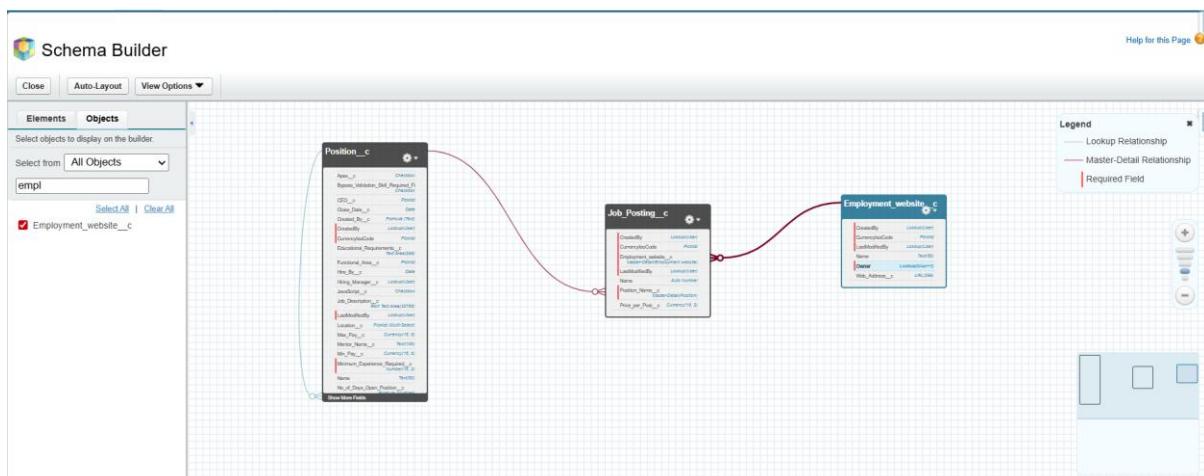
◆ 7. Lookup vs Master-Detail vs Hierarchical Relationships

- **Position → Job_Posting** → Master-Detail (Job Posting tied to Position)
- **Job_Posting → Employment_Website** → Lookup (external references, optional)
- **Candidate → Job_Application** → Lookup (candidate may apply multiple times)
- **Position → Job_Application** → Master-Detail (application always tied to a position)
- **Job_Application → Review** → Master-Detail (reviews belong to application)
- **Hierarchical** → Not used (only applicable for User object).



◆ 8. Junction Objects

- **Job_Application__c** functions as a **junction object**:
 - Connects **Candidate__c** ↔ **Position__c**
 - Enables a many-to-many relationship (one candidate can apply to many positions, and one position can have many candidates).



Phase 4: Process Automation

◆ 1. Validation Rules

Definition: Enforce data integrity by preventing invalid data from being saved.

- Example 1 (Position): $\text{MIN_Pay_c} > \text{MAX_Pay_c}$ → Error: “*Min Pay cannot exceed Max Pay.*”
- Example 2 (Candidate): $\text{NOT}(\text{REGEX}(\text{Phone_Number_c}, "[7-9]{1}[0-9]{9}"))$ → Error: “*Phone number must start with 7-9 and be 10 digits.*”

The screenshot shows a configuration interface for a validation rule named "Minpay_smaller_than_max_pay".

Rule Name: Minpay_smaller_than_max_pay

Active:

Description: Minpay_smaller_than_max_pay

Error Condition Formula:

Example: Discount_Percent_c>0.30 [More Examples...](#)

Display an error if Discount is more than 30%

If this formula expression is true, display the text defined in the Error Message area

Insert Field Insert Operator

Min_Pay_c > Max_Pay_c

Functions:

- ABS
- ACOS
- ADDMONTHS
- AND
- ASCII
- ASIN

[Insert Selected Function](#)

ABS(number)
Returns the absolute value of a number, a number without its sign

[Help on this function](#)

Error Message:

Example: Discount percent cannot exceed 30%

This message will appear when Error Condition formula is true

Error Message: Please Enter Min Pay Smaller than Max Pay

This error message can either appear at the top of the page or below a specific field on the page

Error Location: Top of Page Field [i](#)

◆ 4. Approval Process

Definition: Automates record submission for approval with multiple levels of approvers.

- Example (Position Approval Flow):
 1. Recruiter submits Position → Goes to Recruitment Manager.
 2. If approved → Goes to HR Head.
 3. If approved → Position Status = “Open Approved”.
 4. If rejected → Status = “Closed - Not Approved”.

SETUP Approval Processes

Position: Position_approval
[Help for this Page](#)

[Back to Approval Process List](#)

Process Definition Detail	
Process Name	Position_approval
Unique Name	Position_approval
Description	
Entry Criteria	(Position: Status EQUALS New Position) AND (Position: Minimum Experience Required GREATER OR EQUAL 3)
Record Editability	Administrator ONLY
Approval Assignment Email Template	
Initial Submitters	Position Owner
Created By	Prabhav Kalage, 6/3/2025, 9:00 AM
Modified By	Prabhav Kalage, 6/3/2025, 2:33 PM
Active <input checked="" type="checkbox"/>	
Next Automated Approver Determined By	
Allow Submitters to Recall Approval Requests <input checked="" type="checkbox"/>	

Initial Submission Actions [Add Existing | Add New]

Action	Type	Description
Record Lock		Lock the record from being edited
Edit Remove	Email Alert	pending_approval
Edit Remove	Field Update	new_status
Edit Remove	Task	Approval Request Pending
Edit Remove	Outbound Message	RequestBin

SETUP Approval Processes

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Show Actions Edit	1	manager approval		Position: Max Pay LESS OR EQUAL "USD 10,000", else Approve	Approval based on first response User: Akhilesh Toley, Kshitij Thotange, Ayush Korade	Final Rejection
Show Actions Edit	2	VP manager		(Position: Max Pay GREATER THAN "USD 10,000") AND (Position: Max Pay LESS OR EQUAL "USD 25,000")	Approval based on first response User: Ayush Korade, Ram Gawande	Final Rejection
Show Actions Edit	3	CEO Approver		Position: Max Pay GREATER THAN "USD 25,000"	User: Prabhav Kalage	Final Rejection

Final Approval Actions [Add Existing | Add New]

Action	Type	Description
Edit	Record Lock	Lock the record from being edited
Edit Remove	Field Update	field_update

Final Rejection Actions [Add Existing | Add New]

Action	Type	Description
Edit	Record Lock	Unlock the record for editing
Edit Remove	Field Update	update_status

Recall Actions [Add Existing | Add New]

Action	Type	Description
Edit	Record Lock	Unlock the record for editing
Edit Remove	Field Update	update_field

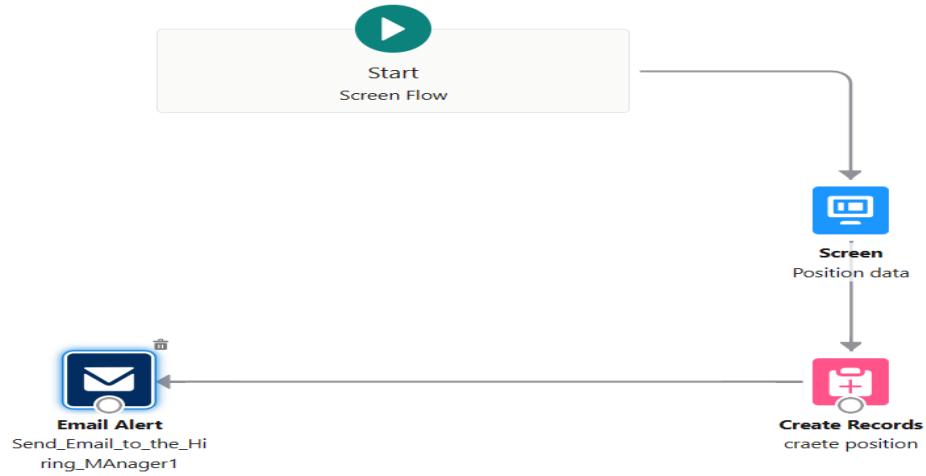
◆ 5. Flow Builder

Definition: Modern automation tool (recommended over Workflow & Process Builder).

Types of Flows:

- Screen Flow** → Candidate self-service application form.
- Record-Triggered Flow** → On Job Application creation → auto-send email to recruiter.
- Schedule-Triggered Flow** → Every Monday → Send weekly report of pending applications to HR.

4. Auto-launched Flow → When Candidate is marked “Selected” → auto-update Position status.



◆ 6. Email Alerts

Definition: Send standard/custom email templates when triggered by automation.

The screenshot shows the 'Classic Email Templates' page in Salesforce. A new template named 'RE Account Creation' has been created. The template details are as follows:

Email Template Detail	Value
Email Templates from Salesforce	MyEmailTemplate
Email Template Name	RE Account Creation
Template Unique Name	RE_Account_Creation
Encoding	Unicode (UTF-8)
Author	Velocity Institute [Change]
Description	RE Account <*>

The 'Available For Use' checkbox is checked. Below the template details, there is a preview section titled 'Plain Text Preview' which contains the following text:

```

Dear {[Account.Owner.fullName]}

Congratulation your Account {[Account.Name]} has been created Successfully with Annual Revenue
{[Account.AnnualRevenue]}

Thanks and Regards
{[Account.CreatedBy]}

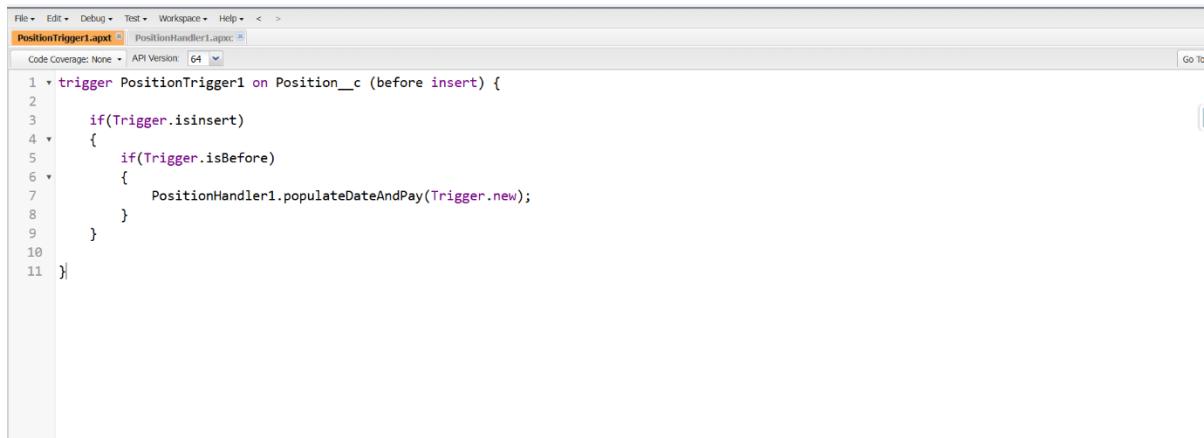
```

Phase 5: Apex Programming

1. Apex Triggers (Before/After Insert, Update, Delete)

In this phase, a **before insert trigger** was created on the Position__c object. The purpose of this trigger is to **auto-populate important fields** when a new Position record is created.

Example: PositionTrigger1 (Before Insert)

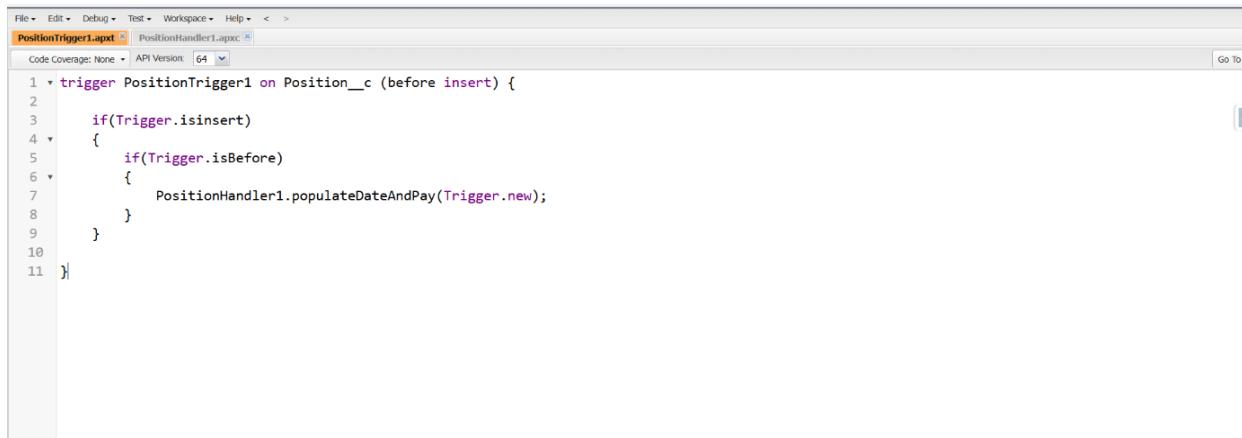


```
trigger PositionTrigger1 on Position__c (before insert) {
    if(Trigger.isinsert)
    {
        if(Trigger.isBefore)
        {
            PositionHandler1.populateDateAndPay(Trigger.new);
        }
    }
}
```

2. Trigger Design Pattern with Handler Class

The handler class PositionHandler1 contains the business logic. Its job is to **set default values** when a new Position record is created with Status = 'New Position'.

Example: PositionHandler1

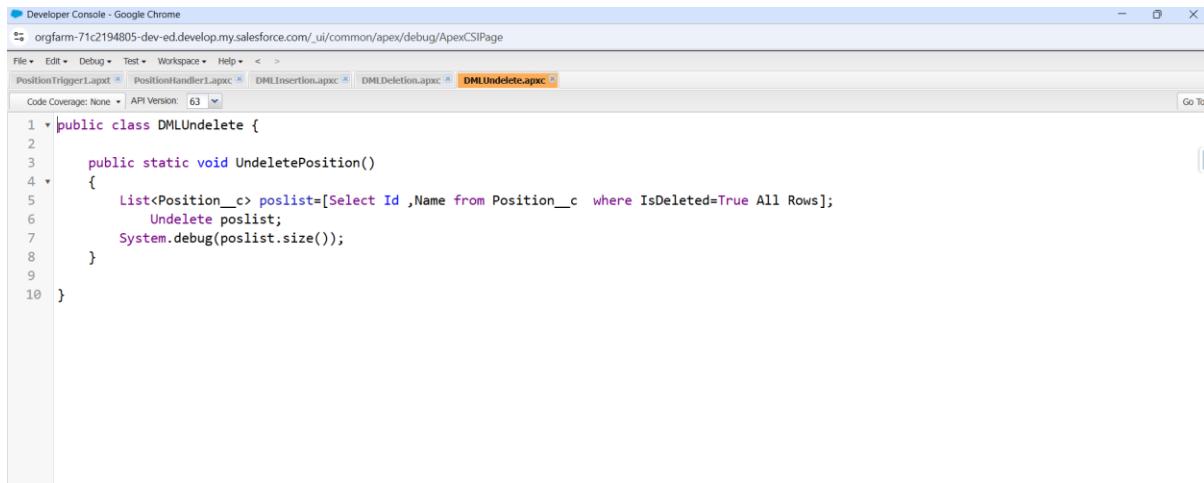


```
trigger PositionTrigger1 on Position__c (before insert) {
    if(Trigger.isinsert)
    {
        if(Trigger.isBefore)
        {
            PositionHandler1.populateDateAndPay(Trigger.new);
        }
    }
}
```

3. DML – Undelete Operation

Apex provides the ability to **restore records that were moved to the Recycle Bin** using the undelete keyword. This is helpful when users mistakenly delete records but the data needs to be recovered programmatically.

Example: DMLUndelete Class

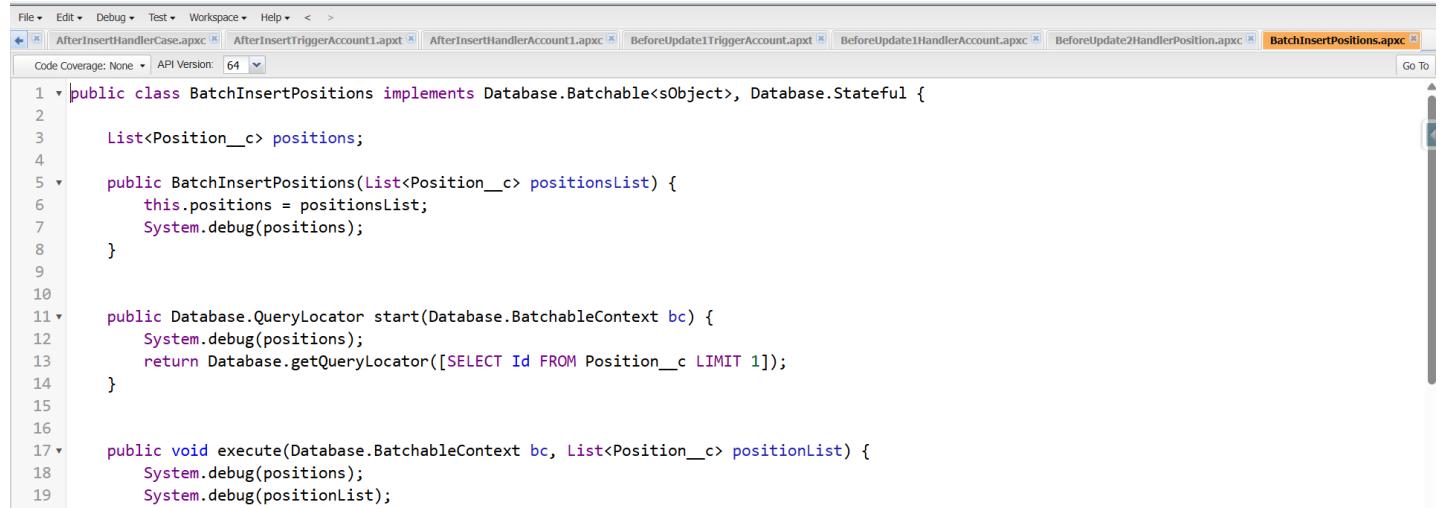


```
Developer Console - Google Chrome
orgfarm-71c2194805-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
PositionTrigger.apxt PositionHandler1.apxc DMLInsertion.apxc DMLDeletion.apxc DMLUndelete.apxc
Code Coverage: None API Version 63 ▾ Go To
1 public class DMLUndelete {
2
3     public static void UndeletePosition()
4     {
5         List<Position__c> poslist=[Select Id ,Name from Position__c where IsDeleted=True All Rows];
6         Undelete poslist;
7         System.debug(poslist.size());
8     }
9
10 }
```

4. Batch Apex

Batch Apex was implemented to handle **large-scale record insertion**. This is particularly useful when data comes from external systems or REST APIs.

Example: BatchInsertPositions



```
File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
AfterInsertHandlerCase.apxc AfterInsertTriggerAccount1.apxt AfterInsertHandlerAccount1.apxc BeforeUpdate1TriggerAccount.apxt BeforeUpdate1HandlerAccount.apxc BeforeUpdate2HandlerPosition.apxc BatchInsertPositions.apxc
Code Coverage: None API Version 64 ▾ Go To
1 public class BatchInsertPositions implements Database.Batchable<sObject>, Database.Stateful {
2
3     List<Position__c> positions;
4
5     public BatchInsertPositions(List<Position__c> positionsList) {
6         this.positions = positionsList;
7         System.debug(positions);
8     }
9
10
11     public Database.QueryLocator start(Database.BatchableContext bc) {
12         System.debug(positions);
13         return Database.getQueryLocator([SELECT Id FROM Position__c LIMIT 1]);
14     }
15
16
17     public void execute(Database.BatchableContext bc, List<Position__c> positionList) {
18         System.debug(positions);
19         System.debug(positionList);
```

5. Test Class

This test class validates the functionality of the **BatchInsertPositions** class by simulating different scenarios. It ensures that the batch job inserts records correctly and handles empty input lists without errors.

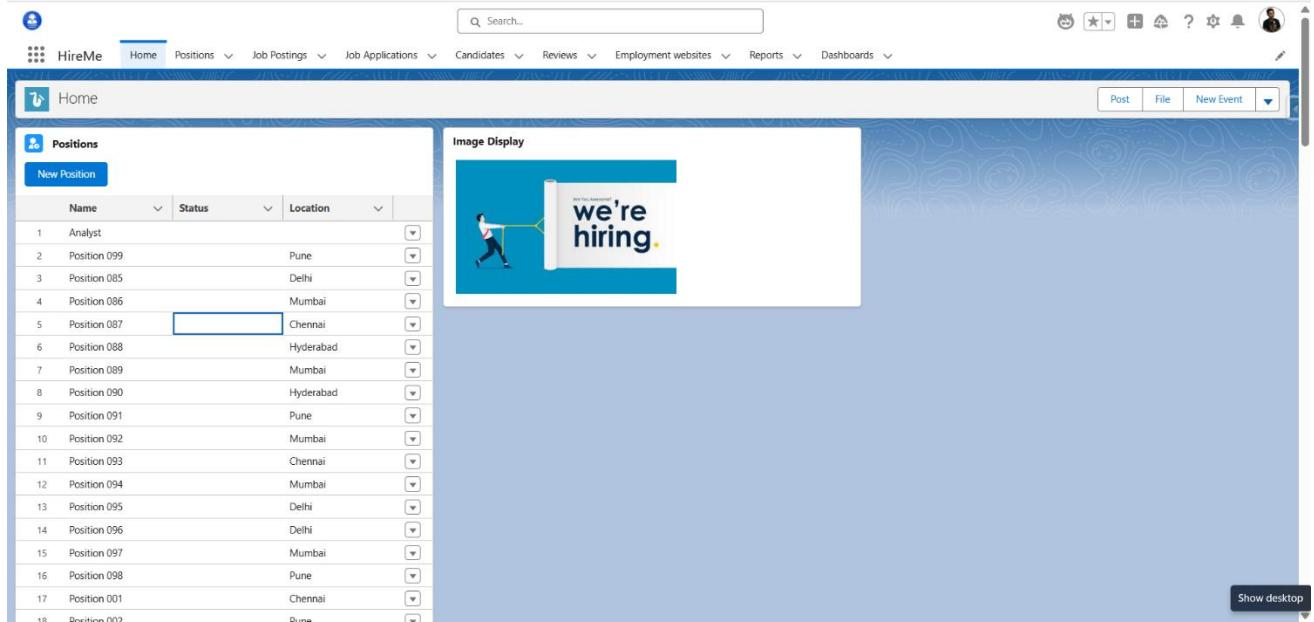


The screenshot shows a Salesforce IDE interface with the following details:

- File Bar:** File, Edit, Debug, Test, Workspace, Help.
- Tab Bar:** AfterInsertHandlerCase.apxc, AfterInsertTriggerAccount1.apxt, AfterInsertHandlerAccount.Lapxc, BeforeUpdate1TriggerAccount.apxc, BeforeUpdate1HandlerAccount.apxc, BeforeUpdate2HandlerPosition.apxc, **BatchInsertPositionsTest.apxc**.
- Code Coverage:** None.
- API Version:** 64.
- Code Preview:** The code is displayed in a monospaced font, showing a test class with setup data and a test method.

```
1  @isTest
2  public class BatchInsertPositionsTest {
3
4      @testSetup
5      static void setupData() {
6
7          List<Position__c> testPositions = new List<Position__c>();
8          for(Integer i = 0; i < 5; i++) {
9              testPositions.add(new Position__c(
10                  Name = 'Test Position ' + i,
11                  Status__c = 'New Position',
12                  Min_Pay__c = 5000,
13                  Max_Pay__c = 10000
14              ));
15          }
16          insert testPositions;
17      }
18
19      @isTest
20      static void testBatchInsertPositions() {
```

Phase 6: User Interface Development

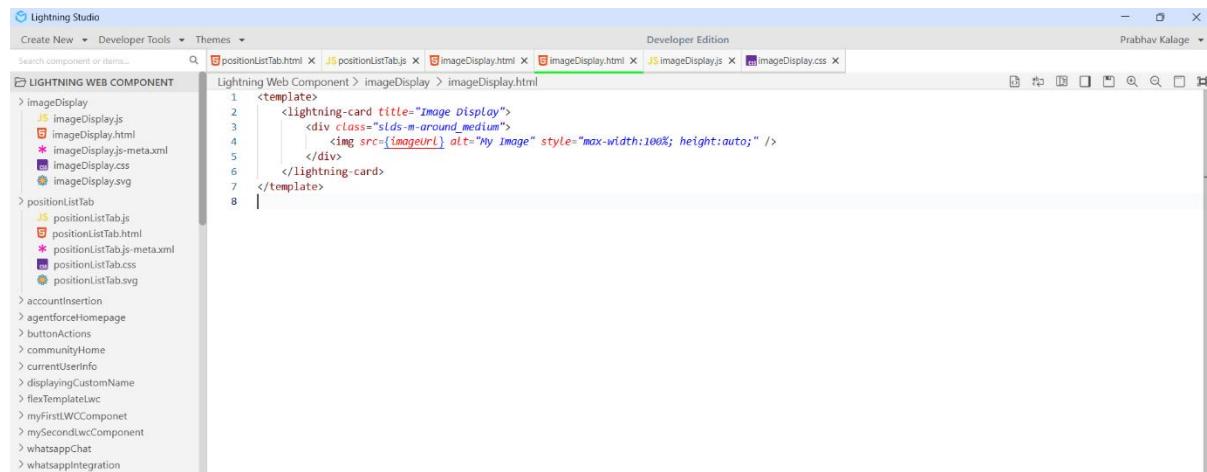


1) Image Display Component (imageDisplay)

Displays a static image from Salesforce static resources with fixed dimensions (288x175 px) in a compact, centered layout.

Implementation Details:

- Uses @salesforce/resourceUrl to import the image.
- HTML wraps the image in a card and container <div>.
- CSS sets width, height, centering, and optional hover effects.



```

Lightning Studio
Create New ▾ Developer Tools ▾ Themes ▾
Search component or items... positionListTab.html JS positionListTab.js JS imageDisplay.html JS imageDisplay.js JS imageDisplay.css
Developer Edition Prabhav Kalage ▾
LIGHTNING WEB COMPONENT
> imageDisplay
  JS imageDisplay.js
  ImageDisplay.html
  * imageDisplay.js-meta.xml
  imageDisplay.css
  ImageDisplay.svg
> positionListTab
  JS positionListTab.js
  positionListTab.html
  * positionListTab.js-meta.xml
  positionListTab.css
  positionListTab.svg
> accountInsertion
> agentforceHomepage
> buttonActions
> communityHome
> currentUserInfo
> displayingCustomName
> flexTemplateLwc
> myFirstLWCComponent
> mySecondLwcComponent
> whatsappChat
> whatsappIntegration

```

The screenshot shows the Lightning Studio interface with the 'Developer Edition' tab selected. The top navigation bar includes 'Create New', 'Developer Tools', 'Themes', and a user dropdown for 'Prabhav Kalage'. A search bar is present above the component tree. The component tree on the left lists various components like 'imageDisplay' and 'positionListTab'. The main editor area displays the 'imageDisplay.js' file content:

```

1 import { LightningElement } from 'lwc';
2 import myImage from '@salesforce/resourceUrl/image';
3
4 export default class ImageDisplay extends LightningElement {
5   imageUrl = myImage;
6 }
7

```

```

Lightning Studio
Create New ▾ Developer Tools ▾ Themes ▾
Search component or items... positionListTab.html JS positionListTab.js JS imageDisplay.html JS imageDisplay.js JS imageDisplay.css JS imageDisplay.js-meta.xml
Developer Edition Prabhav Kalage ▾
LIGHTNING WEB COMPONENT
> imageDisplay
  JS imageDisplay.js
  ImageDisplay.html
  * imageDisplay.js-meta.xml
  imageDisplay.css
  ImageDisplay.svg
> positionListTab
  JS positionListTab.js
  positionListTab.html
  * positionListTab.js-meta.xml
  positionListTab.css
  positionListTab.svg
> accountInsertion
> agentforceHomepage
> buttonActions
> communityHome
> currentUserInfo
> displayingCustomName
> flexTemplateLwc
> myFirstLWCComponent
> mySecondLwcComponent
> whatsappChat
> whatsappIntegration

```

The screenshot shows the Lightning Studio interface with the 'Developer Edition' tab selected. The top navigation bar includes 'Create New', 'Developer Tools', 'Themes', and a user dropdown for 'Prabhav Kalage'. A search bar is present above the component tree. The component tree on the left lists various components like 'imageDisplay' and 'positionListTab'. The main editor area displays the 'imageDisplay.js-meta.xml' file content:

```

1 <xml version="1.0">
2 <LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
3   <apiVersion>63.0</apiVersion>
4   <isExposed>true</isExposed>
5   <targets>
6     <target>lightning__AppPage</target>
7     <target>lightning__HomePage</target>
8     <target>lightning__RecordPage</target>
9     <target>lightning__Tab</target>
10   </targets>
11 </LightningComponentBundle>

```

2) Position List Component (positionList)

Displays a datatable of Position__c records with Edit, Delete, and New actions, allowing easy record management.

Implementation Details:

- Uses @wire to fetch records from getPositions Apex method.
- NavigationMixin handles navigation to record edit/new pages.
- Toast messages show success/error notifications.
- CSS can be used to adjust layout and spacing.

Lightning Studio

Create New ▾ Developer Tools ▾ Themes ▾

Search component or items... positionListTab.html JS positionListTab.js imageDisplay.html JS imageDisplay.js imageDisplay.css * imageDisplay.js-meta.xml

Developer Edition Prabhav Kalage

LIGHTNING WEB COMPONENT

```

<template>
  <lightning-card title="Positions" icon-name="standard:employee_job_position">
    <div class="slds-m-around_small">
      <lightning-button
        label="New Position"
        variant="brand"
        onclick={handleNew}>
      </lightning-button>
    </div>

    <lightning-datatable
      key-field="Id"
      data={positions}
      columns={columns}
      hide-checkbox-column="true"
      onrowaction={handleRowAction}
      show-row-number-column="true">
    </lightning-datatable>
  </lightning-card>
</template>

```

Lightning Studio

Create New ▾ Developer Tools ▾ Themes ▾

Search component or items... positionListTab.html JS positionListTab.js imageDisplay.html JS imageDisplay.js imageDisplay.css * imageDisplay.js-meta.xml

Developer Edition Prabhav Kalage

LIGHTNING WEB COMPONENT

```

import { LightningElement, wire, track } from 'lwc';
import getPositions from '@salesforce/apex/PositionController.getPositions';
import deletePositions from '@salesforce/apex/PositionController.deletePositions';
import { ShowToastEvent } from 'lightning/platformShowToastEvent';
import { refreshApex } from '@salesforce/apex';
import { NavigationMixin } from 'lightning/navigation';

const actions = [
  { label: 'Edit', name: 'edit' },
  { label: 'Delete', name: 'delete' }
];

const columns = [
  { label: 'Name', fieldName: 'Name' },
  { label: 'Status', fieldName: 'Status_c' },
  { label: 'Location', fieldName: 'Location_c' },
  { type: 'action', typeAttributes: { rowActions: actions } }
];

export default class PositionList extends NavigationMixin(LightningElement) {
  @track positions;
  columns = columns;
  wiredPositionsResult;

  @wire(getPositions)
  wiredPositions(result) {
    this.wiredPositionsResult = result;
    if (result.data) {
      this.positions = result.data;
    } else if (result.error) {
      this.showToast('Error', result.error.body.message, 'error');
    }
  }

  // Row action handler (Edit/Delete)
  handleRowAction(event) {
    const rowAction = event.detail.name;
    const row = event.detail.row;
    const position = this.positions.find(p => p.Id === row.Id);
    if (rowAction === 'edit') {
      this.editPosition(position);
    } else if (rowAction === 'delete') {
      this.deletePosition(position);
    }
  }
}

```

APEX CLASS

- AbstractionExample
- AbstractionSubtract
- AbstractionSum
- AccountActionsController
- AccountController
- AccountHandler
- AccountHandler1
- AccountSummaryPrompt

Lightning Studio

Create New ▾ Developer Tools ▾ Themes ▾

Search component or items... positionListTab.html JS positionListTab.js imageDisplay.html JS imageDisplay.js imageDisplay.css * imageDisplay.js-meta.xml * positionListTab.js-meta.xml

Developer Edition Prabhav Kalage

LIGHTNING WEB COMPONENT

```

<?xml version="1.0"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
  <apiVersion>63.0</apiVersion>
  <isExposed>true</isExposed>
  <targets>
    <target>lightning__AppPage</target>
    <target>lightning__HomePage</target>
    <target>lightning__RecordPage</target>
    <target>lightning__Tab</target>
  </targets>
</LightningComponentBundle>

```

APEX CLASS

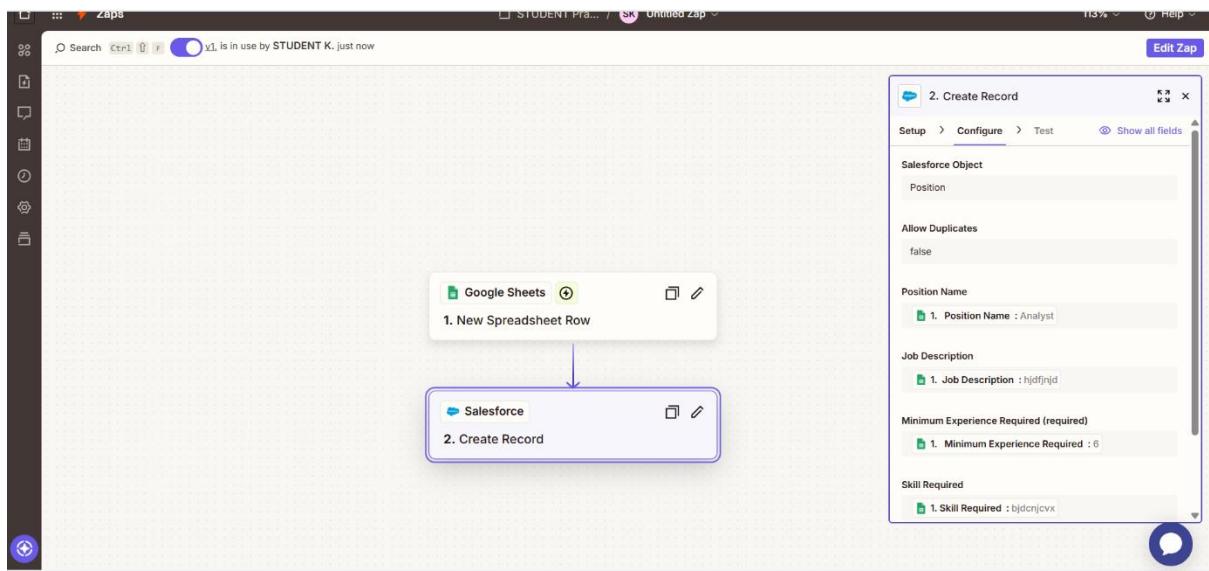
- AbstractionExample
- AbstractionSubtract
- AbstractionSum
- AccountActionsController
- AccountController
- AccountHandler
- AccountHandler1
- AccountSummaryPrompt

Phase 7: Integration & External Access

This phase focused on integrating Salesforce with external systems and enabling secure, automated data exchange.

Google Form Integration with Salesforce using Zapier:

1. **Create Google Form:** Collect responses (e.g., Job Applications).
2. **Setup Zapier:** Create a new Zap to automate data transfer.
 - o **Trigger:** Google Form submission.
 - o **Action:** Salesforce “Create Record” in the Position object.
3. **Configure Salesforce in Zapier:** Authenticate via OAuth.
4. **Map Fields:** Map Google Form fields (Name, Location, Experience, etc.) to Salesforce Position fields.
5. **Test & Activate:** Each new form submission automatically creates a Salesforce record.



Phase 8: Data Management & Deployment

1. Data Import Wizard

Using Salesforce Data Import Wizard, I successfully inserted 100 records for the Position object. The import included fields such as Position Name, Owner, Type, Currency, Location, Status, Experience, Open/Close Dates, Role & Responsibilities, Skills, Educational Requirements, Pay Range, Travel Requirement, and Other Information.

All records were validated for accuracy, and the import process completed without errors, ensuring the data is ready for use in Salesforce workflows, reporting, and assignment processes.

The screenshot shows the 'Choose data' step of the Data Import Wizard. It includes sections for selecting the type of data (Standard objects or Custom objects), defining field mappings (Match by, User field for record owner, Hiring Manager lookup field, Sub Position lookup field), and specifying the data source (CSV file upload). A progress bar at the top indicates 'Getting closer.'

The screenshot shows the 'Edit Field Mapping' step of the Data Import Wizard. It displays a table of mapped fields, showing how CSV headers are mapped to Salesforce objects like Position Name, Location, and Responsibilities. A progress bar at the top indicates 'Almost done.'

Edit	Mapped Salesforce Object	CSV Header	Example	Example	Example
Change	Position Name	Position Name	Position 001	Position 002	Position 003
Change	Location	Location	Chennai	Pune	Pune
Change	Minimum Experience Required	Min Experience (Years)	5	2	3
Change	Responsibilities	Role and Responsibilities	Integrate APIs	Assist data analyse	Develop Apex classes
Change	Skill Required	Skill Required	Apex, JavaScript, T	Salesforce Configu	Apex, JavaScript, Communication
Change	Educational Requirements	Educational Requirements	B.Tech + MBA	B.Tech + MBA	MBA
Change	Min Pay	Min Pay	34931	39575	20973
Change	Max Pay	Max Pay	119949	114799	97210

Bulk Data Load Jobs

[« Back to List: Bulk Data Load Jobs](#)

Bulk Data Load Job Detail		Reload	Status	Closed
Job ID	750NS0000WfhG	Job Type	Bulk V1	
Submitted By	Prabhav Kalage	Operation	Insert	Total Processing Time (ms)
Start Time	9/23/2025, 1:03 PM CAT	Queued Batches	0	262
End Time	9/23/2025, 1:03 PM CAT	In Progress Batches	0	API Active Processing Time (ms)
Time to Complete ([hh:]mm:ss)	00:01	Completed Batches	1	211
Object	Position	Failed Batches	0	Apex Processing Time (ms)
External ID Field		Progress	100%	80
Content Type	CSV	Records Processed	100	
Concurrency Mode	Parallel	Records Failed	100	
API Version	64.0	Retries	0	

[Reload](#)

Batches													
View Request	View Result	Batch ID	Start Time	End Time	Total Processing Time (ms)	API Active Processing Time (ms)	Apex Processing Time (ms)	Records Processed	Records Failed	Retry Count	State Message	Status	
View Request	View Result	751NS00000SbUGj	9/23/2025, 1:03 PM	9/23/2025, 1:03 PM	282	211	80	100	100	0	Completed		

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
Id	Success	Created	Error														
2 a00NS00001yeC7ZYAU	TRUE	TRUE															
3 a00NS00001yeC7YAE	TRUE	TRUE															
4 a00NS00001yeC7oYAE	TRUE	TRUE															
5 a00NS00001yeC7oYAE	TRUE	TRUE															
6 a00NS00001yeC7oYAE	TRUE	TRUE															
7 a00NS00001yeC7oYAE	TRUE	TRUE															
8 a00NS00001yeC7tYAE	TRUE	TRUE															
9 a00NS00001yeC7tYAE	TRUE	TRUE															
10 a00NS00001yeC7hYAE	TRUE	TRUE															
11 a00NS00001yeC7tYAE	TRUE	TRUE															
12 a00NS00001yeC7tYAE	TRUE	TRUE															
13 a00NS00001yeC7tYAE	TRUE	TRUE															
14 a00NS00001yeC7tYAE	TRUE	TRUE															
15 a00NS00001yeC7mYAE	TRUE	TRUE															
16 a00NS00001yeC7nYAE	TRUE	TRUE															
17 a00NS00001yeC7oYAE	TRUE	TRUE															
18 a00NS00001yeC7oYAE	TRUE	TRUE															
19 a00NS00001yeC7oYAE	TRUE	TRUE															
20 a00NS00001yeC7oYAE	TRUE	TRUE															
21 a00NS00001yeC7oYAE	TRUE	TRUE															
22 a00NS00001yeC7tYAE	TRUE	TRUE															
23 a00NS00001yeC7uYAE	TRUE	TRUE															
24 a00NS00001yeC7vYAE	TRUE	TRUE															
25 a00NS00001yeC7wYAE	TRUE	TRUE															
26 a00NS00001yeC7xYAE	TRUE	TRUE															
27 a00NS00001yeC7yYAE	TRUE	TRUE															
28 a00NS00001yeC7zYAE	TRUE	TRUE															
29 a00NS00001yeC80YAE	TRUE	TRUE															
30 a00NS00001yeC81YAE	TRUE	TRUE															

Duplicate Rule :

Duplicate Rule named “Duplicate Position” has been configured for the Position object in Salesforce to prevent duplicate entries.

- Purpose: Ensure data integrity by blocking duplicate Position records.
- Rule Behavior:
 - On Create: Block duplicate records
 - On Edit: Allow, with alert and report generation
- Alert Message: *This is Duplicate Record*

- Matching Logic: Uses Position Matching rule to compare Position Name for exact matches.
- Field Mapping: Confirmation mapping applied.
- Conditions: No additional filters; applies to all Position records.

This setup helps maintain clean and accurate Position data while notifying users of duplicates and allowing controlled updates.

The screenshot shows the Salesforce Duplicate Rules page. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. A search bar is present on the left. The main content area has a title 'd SETUP Duplicate Rules' and a sub-section 'Duplicate Position'. Below this, a table titled 'Duplicate Rule Detail' displays a single rule:

	Rule Name	Duplicate Position	Action	Operations On Create	Operations On Edit
Record Level Security	Description	Position	Block	<input type="checkbox"/> Alert	<input type="checkbox"/> Report
Action On Create	Object	Bypass sharing rules	Allow	<input checked="" type="checkbox"/> Alert	<input checked="" type="checkbox"/> Report
Action On Edit	Alert Text	This is Duplicate Record			
Active		✓			
Matching Rule	Conditions	Position Matching Mapped		Matching Criteria	Position: Name EXACT MatchBlank = FALSE
Created By	Created By	Prabhav Kalege, 6/4/2025, 12:54 PM		Modified By	Prabhav Kalege, 6/4/2025, 12:54 PM

At the bottom of the table are 'Edit', 'Delete', 'Clone', and 'Deactivate' buttons.

Phase 9: Reporting, Dashboards & Security Review

In this phase, the focus was on generating insights, visualizing data, and ensuring secure access to reports and dashboards. Salesforce's robust reporting features enabled effective tracking of business metrics, while dashboards provided real-time visibility into operations.

The screenshot shows a Salesforce report titled "Report: Positions New Positions Report". The report displays 128 total records. The data is organized into categories such as Admin Assistant, Analyst, Business Analyst, Data Analyst, DevOps Engineer, Dr. Sujoy Pal, and Frontend Developer. Each category includes sub-totals for various job descriptions like "Assist admin & HR activities", "Analyze requirements, document flows", etc. The report includes columns for Job Description, Educational Requirements, Max Pay, Skill Required, Min Pay, and Status of Min Pay. The status of min pay is categorized as Low budget, Medium budget, or High budget.

The screenshot shows a Salesforce dashboard titled "Dashboard Position Report". The dashboard was last refreshed 114 days ago. It includes a summary message: "Last refreshed 114 days ago. Refresh this dashboard to see the latest data." and "As of Jun 3, 2025, 5:22 AM Viewing as Prabhav Kalage Change". The main area contains two charts: a bar chart titled "New Positions Report" showing the sum of max pay for various positions like Admin Assistant, Business Analyst, Data Analyst, etc., and a treemap chart titled "New Positions Report" showing the same data in a hierarchical structure. A sidebar on the left shows a pie chart for "Sum of Min Pay (USD)" across different position categories. At the bottom, there are links to "View Report (New Positions Report)" and "As of Jun 3, 2025, 5:22 AM".