Phase 4: Process Automation (Admin)

1. Validation Rules

When to use: Block invalid data entry (simple business rules).

- Prevent scheduling an Interview before Application date.

Steps

1. Setup → Object Manager → select the object (e.g., Interview).
2. Left menu → Validation Rules → New.
3. Fill:
    - Rule Name: Interview_Date_Check
    - Description: Prevent interview before application date
    - Error Condition Formula:

   Interview_Date__c < DATETIMEVALUE (Application__r.Application_Date__c)

      - Error Message: Interview date must be on or after the application date.
    - Error Location: Field → select Interview Date.
4. Click Save.
5. Test: Try to create an Interview with an earlier date — you must get the error.


Interview Validation Rule
Back to Interview

Validation Rule Detail — Edit | Clone

Rule Name	Interview_Date_Check	Active	✓
Error Condition Formula	Interview_Date__c < DATETIMEVALUE(Application__r.Application_Date__c)		
Error Message	Interview date must be on or after the application date.	Error Location	Interview Date
Description	Prevent interview before application date		
Created By	Santhosh Pathulothu, 9/25/2025, 6:38 AM	Modified By	Santhosh Pathulothu, 9/25/2025, 6:38 AM

2. Workflow Rules

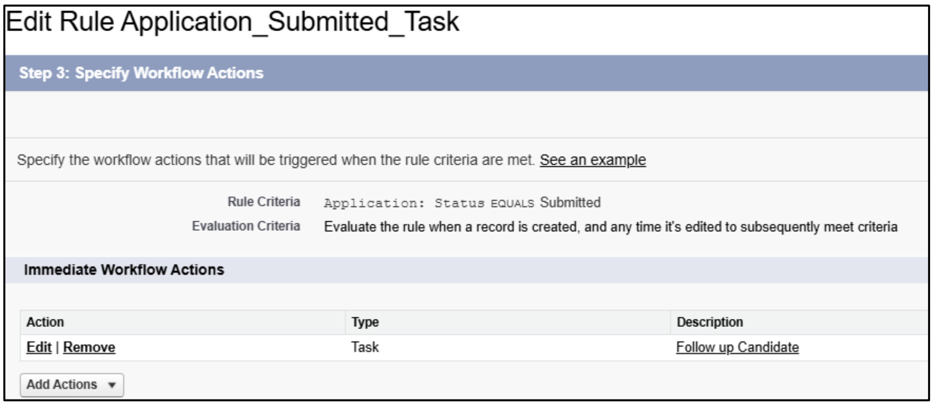
When to use: Simple email/field-updates/tasks that must run on a record create/edit.

- When Application.Status = Submitted, create a Task for Recruiter and send confirmation email.

Steps

1. Setup → in Quick Find type Workflow Rules → click Workflow Rules.
2. Click New Rule.
3. Choose Object → Application → Next.
4. Rule Name: Application_Submitted_Task
5. Evaluation Criteria: created, and every time it’s edited to meet the criteria
6. Rule Criteria: Status EQUALS Submitted (use picklist selector) → Save & Next.

7. Add Workflow Actions:
    - New Task → set Subject = Follow up Candidate, Assigned To = Owner, Due Date = TODAY () +2, Priority = Normal → Save.
8. Click Done → Activate the workflow rule.



3. Approval Process

When to use: Formal approvals (HR approves hiring).

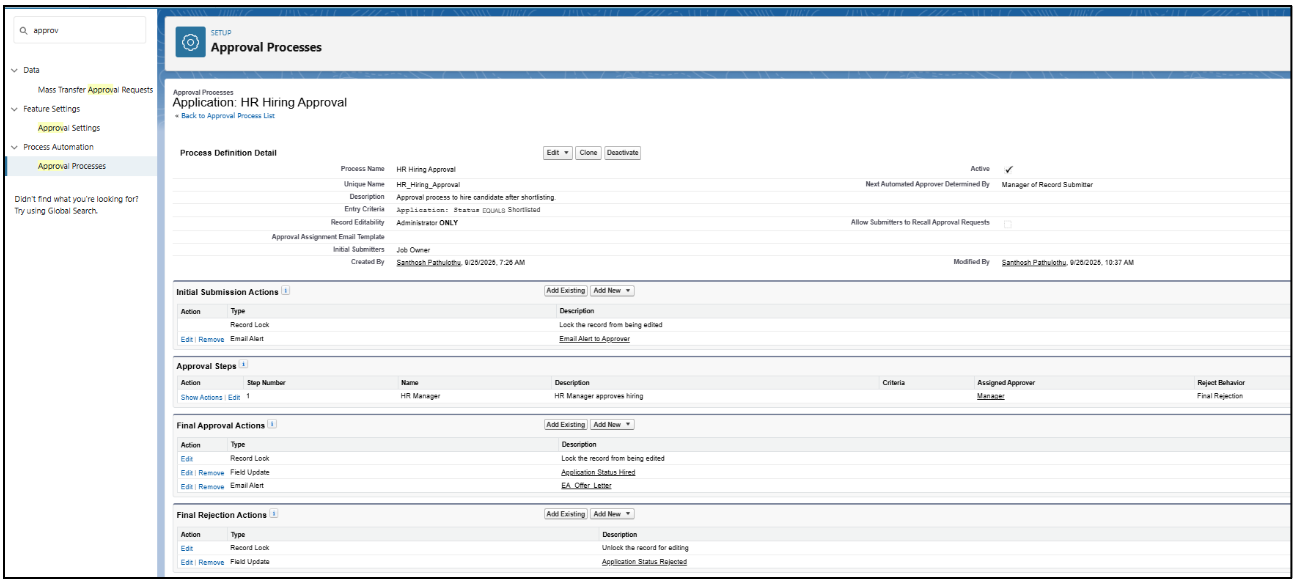
- Approve Application to move to Hired.

Steps (Wizard)

1. Setup → Approval Processes → Create New Approval Process → Use Standard Setup Wizard.
2. Choose Object = Application.
3. Name = HR_Hiring_Approval. Next.
4. Entry Criteria = Status EQUALS Shortlisted (or add Interview.Result = Selected). Next.
5. Specify Approver (e.g., Manually choose approver or use a field like HR_Manager__c).
6. Initial Submission Actions: (optional) Lock Record; Email Alert to approver.
7. Final Approval Actions:

    Field Update set Application.Status = Hired; Email Alert (Offer).
8. Final Rejection Actions:

    Field Update set Application.Status = Rejected; Email Alert (Rejection).
9. Save & Activate.



4. Flow Builder

Record-Triggered Flow

Step 1: Create a New Flow

Select Record-Triggered Flow → click Create.

Step 2: Define the Trigger

1. Object: Choose Application.
2. Trigger: When a record is Created.
3. Condition Requirements → All Conditions Are Met (AND).
    - Field: Status
    - Operator: Equals
    - Value: Submitted
4. Optimize the Flow for: Actions and Related Records.
5. Click Done.

Step 3: Add Email Alert (Confirmation to Candidate)

1. In the Flow canvas, click + → choose Action.
2. Search Email Alert.
3. If you've already created the Application Confirmation Email Alert (with template), select it.
    - Name it: Send_Application_Confirmation.
    - This will send to Candidate Email automatically.
4. Click Done.

Step 4: Add Task (For Recruiter)

1. Click + again → choose Create Records.

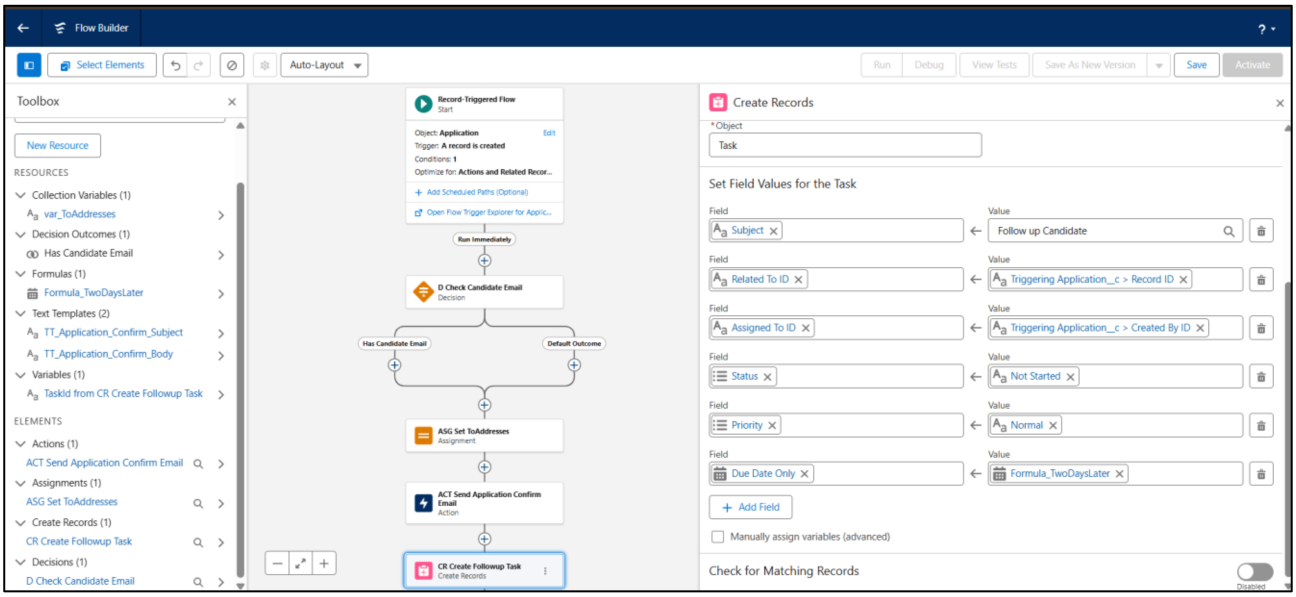
2. Label: Create Recruiter Task.
3. How Many Records: One.
4. How to Set Fields: Use separate resources and literal values.
5. Object: Task.
6. Set field values:
    - Subject: "Follow up with Candidate"
    - Assigned To: Application.OwnerId (Recruiter who owns the Application).
    - Due Date: TODAY () + 2
    - Priority: Normal. Click Done

Step 5: Save & Activate

1. Click Save → give it a Name:

   FL Application OnCreate SendConfirm CreateTask .

2. Click Activate.



Build the Screen Flow

1) Open Flow Builder

1. Setup → Quick Find → Flows → Flows.
2. Click New Flow.
3. Choose Screen Flow → Click Create.

2) Create *recordId* input variable

1. In Flow Builder, open the Manager tab (left side).
2. Click New Resource.
    - Resource Type: Variable
    - API Name: recordId

- Data Type: Text
- Check Available for input → Done (Save resource).

This is required so Salesforce passes the current Interview record Id to the Flow when launched from the record page.

3) Add the Screen (interviewer form)

1. Click the + below Start → choose Screen.
2. Label the screen: Interviewer Feedback.
3. Add components (drag from left palette):

a. Radio Buttons (Result)

- Component: Radio Buttons
- Label: Result
- API Name: result
- Click New Choice → add choices:
    - Label Pending, Value Pending → Save choice
    - Label Selected, Value Selected → Save choice
    - Label Rejected, Value Rejected → Save choice
- Under Store Output Value In, click New Resource → create a Variable:
    - Resource Type: Variable
    - API Name: varResult
    - Data Type: Text
    - Leave Available for input unchecked → Create.
- (Optional) Check Required.

b. Long Text Area (Feedback)

- Component: Text Area (or Long Text Area)
- Label: Feedback
- API Name: feedback

Click Done to save the Screen.

4) Get the Interview record (so we can update it and use fields)

1. Click + after the Screen → Get Records.
2. Label: Get_Interview_Record.
3. Object: Interview.
4. Filter: Id | Operator Equals | Value {!recordId} .

5. How Many Records to Store: Only the first record.
6. Store: All fields (so you can reference Interview.Interviewer__c etc.).
7. Click Done.

Alternative (shortcut): you could skip Get Records and use Update Records with condition Id = {!recordId}. But Get Records makes it easy to reference interviewer Id.

5) Update the Interview record with screen inputs

1. Click + → Update Records.
2. Label: Update_Interview_With_Feedback.
3. Choose Use the Interview record that was found earlier (select the Get_Interview_Record resource).
4. Under Set Field Values for the Interview Record set:
   - Result__c = {!varResult}
   - Feedback__c = {!varFeedback}
   - Next_Steps__c = {!varNextSteps}
5. Click Done.

6) Create a Task for follow-up

1. Click + → Create Records.
2. Label: Create_Interview_Task.
3. How Many Records to Create: One.
4. Record Choice: Use separate resources, and literal values.
5. Object: Task.
6. Set Field Values (example mapping):
   - Subject = Interview follow-up
   - OwnerId = {!Get_Interview_Record.Interviewer__c} ← assigns to the Interviewer user (if Interviewer__c is a User lookup)
   - WhatId = {!Get_Interview_Record.Id} ← links task to the Interview record
   - ActivityDate = {!varNextSteps} (date/time value; if null, do not set)
   - Status = Not Started
   - Priority = Normal
7. Click Done.

Note: If your Interviewer field is not a User lookup, set OwnerId to another appropriate user or the record owner: {!Get_Interview_Record.OwnerId}.

7) Add a Success Screen

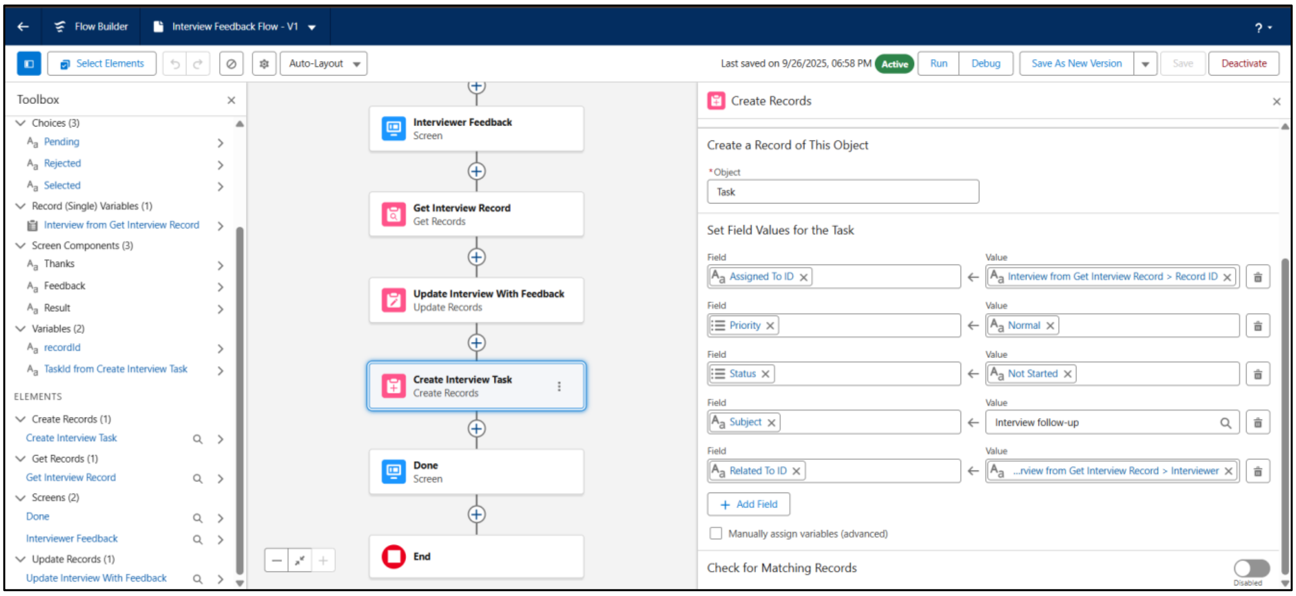
1. Click + → Screen.
2. Label: Done
3. Add a Display Text component: “Thanks — feedback saved and task created.”
4. Click Done.

8) Connect the elements

- Ensure flow path is: Start → Interviewer Feedback (Screen) → Get_Interview_Record → Update_Interview_With_Feedback → Create_Interview_Task → Done (or End).

9) Save & Activate

1. Click Save.
   - Flow Label: **Interview_Feedback_Flow**
   - API Name auto fills.
   - Description: Screen flow for interviewer to submit feedback; updates Interview and creates Task.
2. Click Activate.



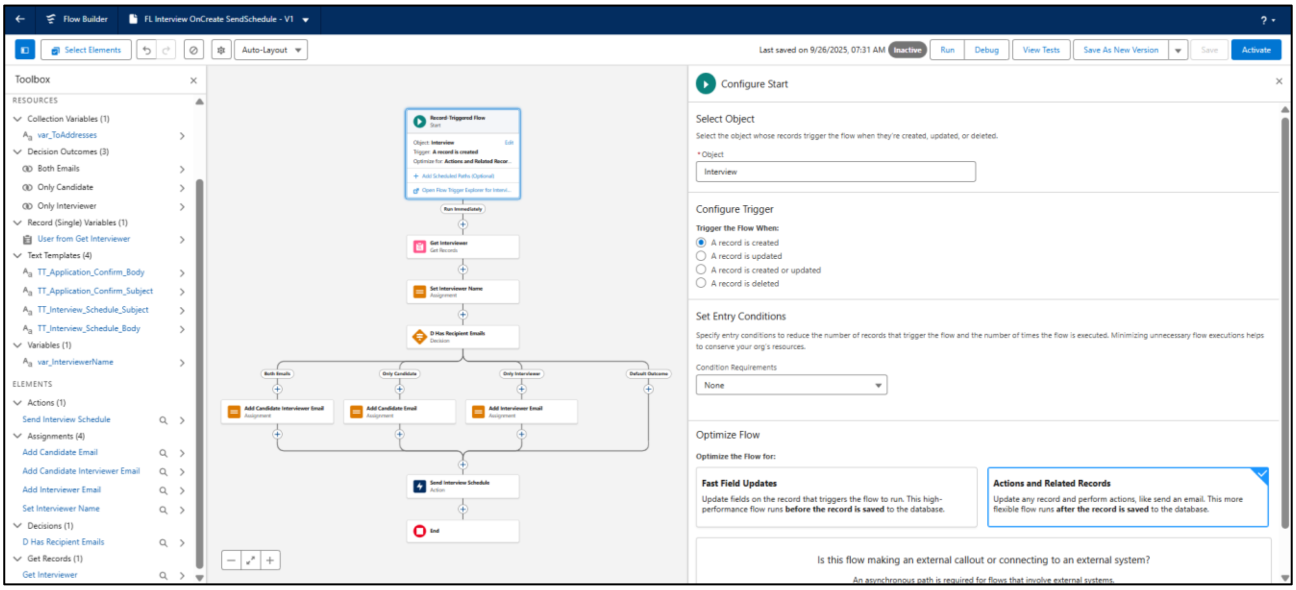
Build Record-Triggered Flow: Application Created → Send Confirmation Email + Create Task

Use Record-Triggered Flow (After Save).

Steps:

1. Setup → Flow → New Flow → Record-Triggered Flow → Create.
2. Object: Application. Trigger: A record is created.
3. Condition Requirements: Status__c EQUALS Submitted.
4. Run the Flow: After Save (so email can access related record data). Click Done.

5. On canvas, click + → choose Action → search Send Email (core action) OR select Email Alerts action (if you created Email Alert).
   - If using Send Email:
     - To Addresses: set to $Record.Candidate__r.Email__c (click the field chooser to insert this).
     - Subject: create a Text Template resource or type Application Received: {!$Record.Job__r.Job_Title__c}.
     - Body: create a Text Template resource and reference fields.
   - If using Email Alert: pick the Email Alert EA_Application_Confirmation.
6. After the send email action, click + → Create Records → create a Task:
   - Label: Create Task - Follow up Candidate
   - How Many Records to Create: One
   - Object: Task
   - Set fields:
     - Subject = Follow up Candidate
     - WhatId = $Record.Id (or link to Application record)
     - OwnerId = $Record.OwnerId (assign to app owner)
     - ActivityDate = Formula TODAY() + 2 (or a Date resource)
     - Status = Not Started
   - Save element.
7. Save the Flow → give name **FL_Application_OnCreate_SendConfirm_CreateTask** → Activate.



Record-Triggered Flow: sends the Interview Schedule email to both the Candidate and the Interviewer

Create Text Templates

Subject Text Template

- Label: TT_Interview_Schedule_Subject
- API Name: TT_Interview_Schedule_Subject

Interview Scheduled: {!$Record.Application__r.Job__r.Name}

Body Text Template

- Label: TT_Interview_Schedule_Body
- API Name: TT_Interview_Schedule_Body

Create the Text Collection Variable for recipients

Steps:

1. Open Flow → Flows → New Flow
2. In Flow Builder left panel → New Resource:
   - Resource Type: Variable
   - API Name: var_ToAddresses
   - Label: To Address Collection
   - Data Type: Text
   - Allow multiple values (collection): Checked
   - Leave Available for Input/Output unchecked.
   - Save.

Create the Record-Triggered Flow (Start)

1. Setup → Quick Find → Flow → Flows → New Flow.
2. Choose Record-Triggered Flow → Create.
3. Configure the Start:
   - Object: Interview (select Interview__c)
   - Trigger: A record is created
   - Condition Requirements: None
   - Run the Flow: After the record is saved
   - Click Done.

Add Decision element — decide which emails to send (both/only candidate/only interviewer)

1. Click + under Start → Decision.
   - Label: D_Has_Recipient_Emails

- API Name: D_Has_Recipient_Emails

Create outcomes:

- Outcome 1:
  - Label: Both_Emails
  - API Name: Both_Emails
  - Conditions (All conditions are true):

    $Record.Application__r.Candidate__r.Email__c Is Null False

    $Record.Interviewer__r.Email Is Null False
- Outcome 2:
  - Label: Only_Candidate
  - API Name: Only_Candidate
  - Conditions:

    $Record.Application__r.Candidate__r.Email__c Is Null False

    $Record.Interviewer__r.Email Is Null True
- Outcome 3:
  - Label: Only_Interviewer
  - API Name: Only_Interviewer
  - Conditions:
    - $Record.Application__r.Candidate__r.Email__c Is Null True
    - $Record.Interviewer__r.Email Is Null False

Add Assignment(s) — populate the var_ToAddresses collection

Path: Both_Emails

1. Click + on the Both_Emails path → Assignment.
   - Label: ASG_Add_Candidate_Email (first assignment)
   - API Name: ASG_Add_Candidate_Email
   - Set Variable: var_ToAddresses Operator = Add Value = $Record.Application__r.Candidate__r.Email__c
2. Add another Assignment (or in same assignment add another line):
   - Label: ASG_Add_Interviewer_Email
   - API Name: ASG_Add_Interviewer_Email
   - Set Variable: var_ToAddresses Operator = Add Value = $Record.Interviewer__r.Email

Path: Only_Candidate

- Add Assignment: add Candidate email to var_ToAddresses.

Path: Only_Interviewer

- Add Assignment: add Interviewer email to var_ToAddresses.

Add the Send Email Action

After each Assignment (for each path), click + → Action.

In Search Actions & Resources, type Send Email → choose Salesforce core Send Email action.

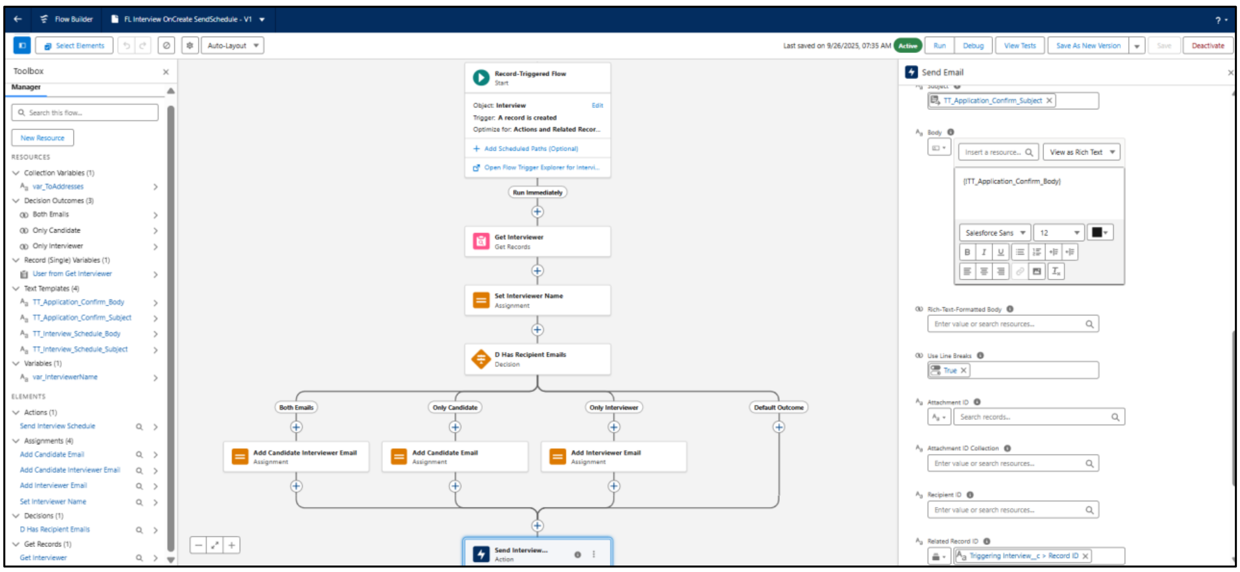
Configure the action:

- Label: ACT_Send_Interview_Schedule_Email
- API Name: ACT_Send_Interview_Schedule_Email
- Subject: select TT_Interview_Schedule_Subject (Text Template)
- Body: select TT_Interview_Schedule_Body (Text Template)
- Recipient Address Collection: select var_ToAddresses
- Sender Type: Default Workflow User (or Org-Wide Email Address if you set one)
- Use Line Breaks: True
- Rich-Text-Formatted Body: unchecked if using plain text templates
- Related Record ID: $Record.Id (links email to the Interview record)

Click Done.

Connect elements & Save Flow

- Connect Start → Decision → respective Assignments → Send Email action → End.
- Save the Flow:
    - Label: **FL_Interview_OnCreate_SendSchedule**
    - API Name: FL_Interview_OnCreate_SendSchedule
    - Description: Send interview schedule email to candidate and interviewer when Interview record is created.
- Click Activate.

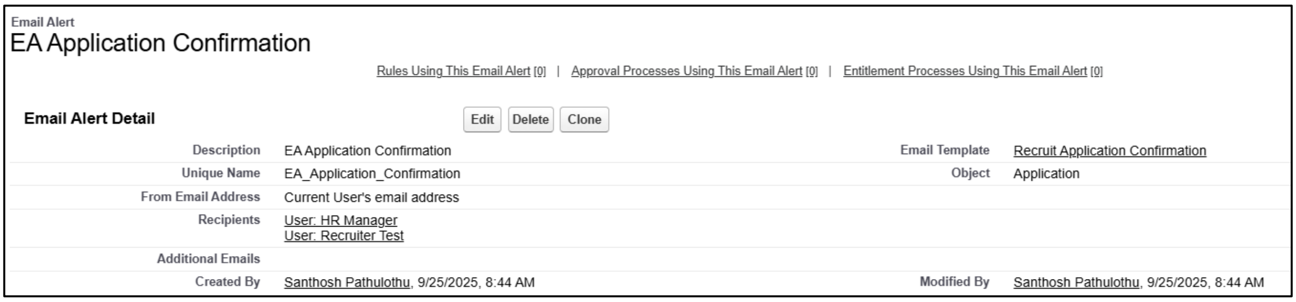


4. Email Alerts

1. EA Application Confirmation

1. Setup → Search Email Alerts → New Email Alert.
2. Gave it the label: *EA Application Confirmation*.
3. Chose the object: Application__c.
4. Selected the email template: *Recruit Application Confirmation*.
5. Set recipient type = Email Field → Candidate Email.

   When a candidate applies for a job (new Application record is created), this email is automatically sent confirming their application submission.



2. EA Interview Schedule

1. Setup → **Email Alerts** → New.
2. Label: *EA Interview Schedule*.
3. Object: **Interview__c**.
4. Email Template: *Recruit Interview Schedule*.
5. Recipients: Candidate Email + Interviewer (User lookup).

   When an interview is scheduled (Interview__c created), this sends an email with the interview date, time, mode, and candidate details to both Candidate & Interviewer.

3. EA Offer Letter

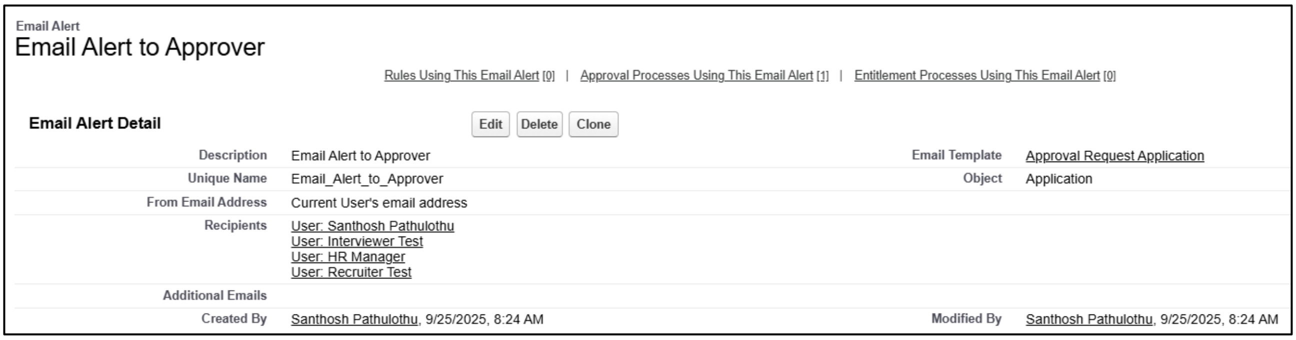
1. Setup → Email Alerts → New.
2. Label: *EA Offer Letter*.
3. Object: Application__c.
4. Email Template: *Recruit Offer Letter*.
5. Recipient: Candidate Email.

   When HR approves a shortlisted candidate's application, the system sends an Offer Letter email automatically.

4. Email Alert to Approver

1. Setup → Email Alerts → New.
2. Label: *Email Alert to Approver*.
3. Object: Application__c.
4. Email Template: *Approval Request Application*.
5. Recipient Type: Related User → Manager/HR approver.

When an Application record enters the Approval Process, this email notifies the assigned approver (HR Manager) to review and take action (approve/reject).



Email Alert

Email Alert to Approver

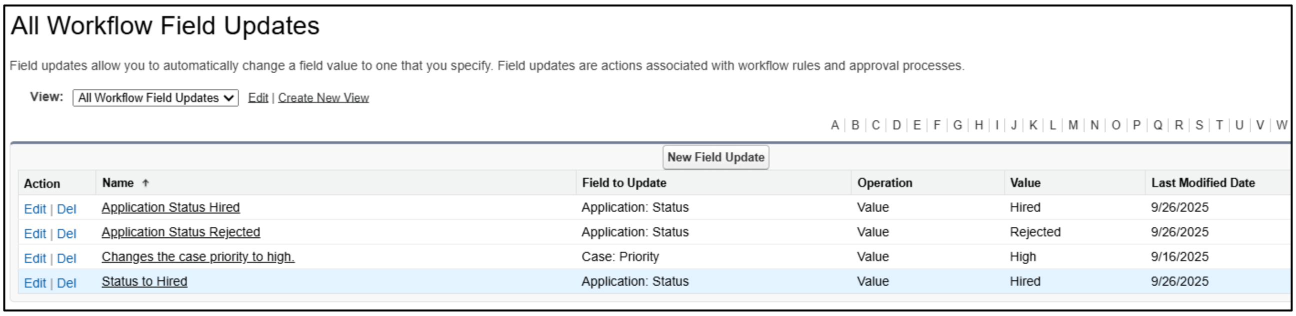
Rules Using This Email Alert [0] | Approval Processes Using This Email Alert [1] | Entitlement Processes Using This Email Alert [0]

Email Alert Detail Edit Delete Clone

Description	Email Alert to Approver	Email Template	Approval Request Application
Unique Name	Email_Alert_to_Approver	Object	Application
From Email Address	Current User's email address		
Recipients	User: Santhosh Pathulothu User: Interviewer Test User: HR Manager User: Recruiter Test		
Additional Emails			
Created By	Santhosh Pathulothu, 9/25/2025, 8:24 AM	Modified By	Santhosh Pathulothu, 9/25/2025, 8:24 AM

5. Field Updates

When to use: Change field values automatically (via Workflow, Process Builder, Flow, Approval Process).



All Workflow Field Updates

Field updates allow you to automatically change a field value to one that you specify. Field updates are actions associated with workflow rules and approval processes.

View: All Workflow Field Updates Edit | Create New View

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W

New Field Update

Action	Name ↑	Field to Update	Operation	Value	Last Modified Date
Edit | Del	Application Status Hired	Application: Status	Value	Hired	9/26/2025
Edit | Del	Application Status Rejected	Application: Status	Value	Rejected	9/26/2025
Edit | Del	Changes the case priority to high.	Case: Priority	Value	High	9/16/2025
Edit | Del	Status to Hired	Application: Status	Value	Hired	9/26/2025

Application Status Hired

Setup → Workflow Actions → Field Updates → New Field Update.

1. Action Name: *Application Status Hired*.
2. Object: Application__c.
3. Field to Update: Status__c.
4. Operation: Set Value = Hired.
   Whenever a workflow/flow triggers this action (e.g., after final HR approval), the Application's status automatically changes to "Hired".

Application Status Rejected

1. Setup → Field Updates → New.
2. Action Name: *Application Status Rejected*.

3. Object: Application__c.
4. Field: Status__c.
5. Operation: Set Value = Rejected.
   If a candidate fails interviews or gets rejected, automation updates the Application’s status to “Rejected”.

Changes the Case Priority to High

This is a sample field update that exists in Salesforce Cases object.

- Object: Case.
- Field: Priority.
- Value: High.

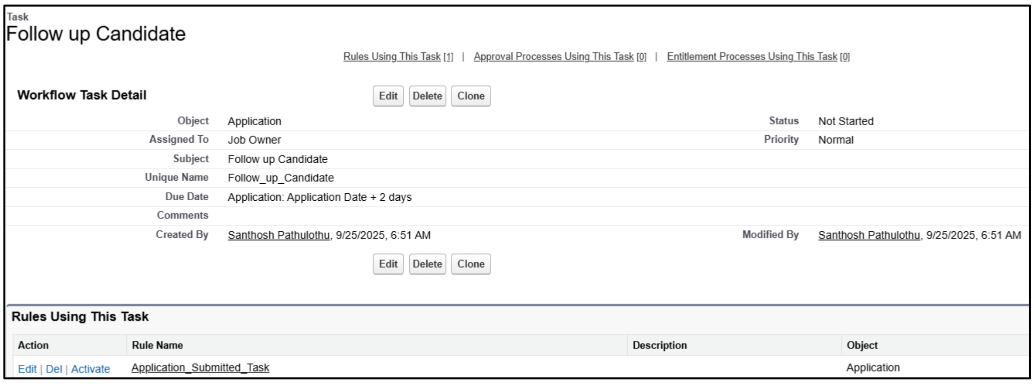
Used for Service/Support use cases. Not directly part of your Job Application project, but it automatically changes a Case’s Priority to High based on conditions (e.g., escalation workflow).

6. Tasks

Create follow-up reminders for recruiters/HR.

Steps (Flow or Workflow)

1. Add Create Records element in Flow or New Task in Workflow/Process Builder.
2. Fields to set:
   - Subject: Interview follow-up
   - WhoId: candidate contact (if using Contact), otherwise leave null
   - WhatId: related job/application id (use recordId)
   - OwnerId: assign to user (e.g., Recruiter)
   - Priority: Normal/High
   - Status: Not Started
   - ActivityDate: date due
3. Save → Activate.



7. Custom Notifications

Send in-app push/desktop notifications to users (better UX than email for timely alerts)

Steps

1. Setup → Notification Builder → Custom Notifications → New.
   - Name: Interview_Assigned
   - Supported Channels: Desktop, Mobile (select both). Save.
2. Add a Flow Action to Send Notification:
   - In Flow Builder add Action → Choose Send Custom Notification (system action).
   - Notification Type: pick Interview_Assigned.
   - Title/Body: use Text Template with merge fields.
   - Recipient(s): set a User or User Id collection (e.g., Interviewer user lookup).
3. Save & Activate Flow.