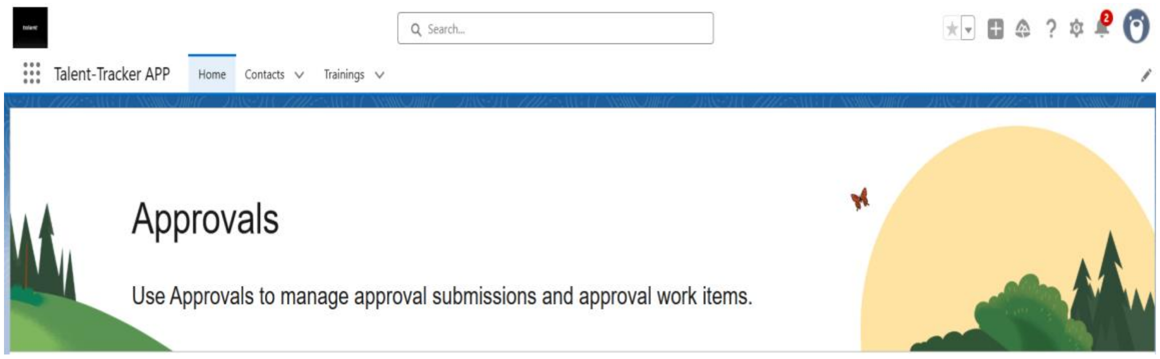
Talent Tracking App – Salesforce Project Report



Approvals

Use Approvals to manage approval submissions and approval work items.

Phase 1: Problem Understanding & Industry Analysis

- **Requirement Gathering**: Track employee training progress efficiently.
- **Stakeholder Analysis**: HR team, Trainers, Employees.
- **Business Process Mapping**:

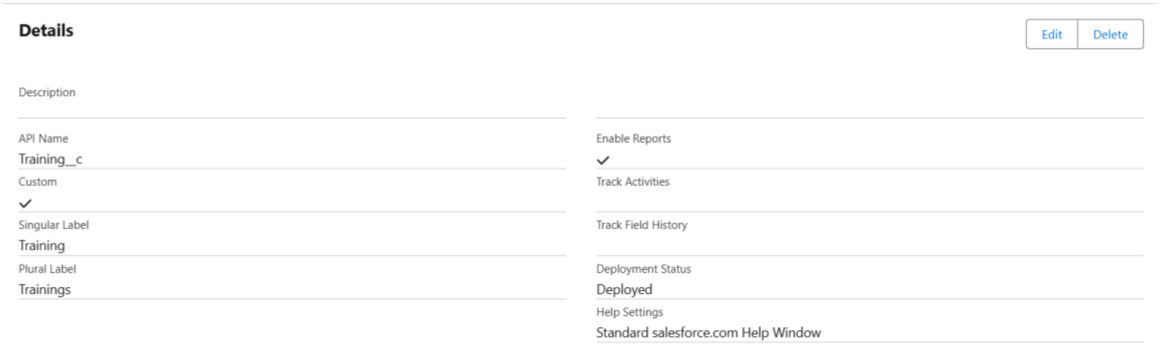
- HR assigns training.
- Employees complete training.
- HR monitors training status.

- **Industry-specific Use Case**: Corporate learning & development.
- **AppExchange Exploration**: Alternatives like Cornerstone, but opted for custom solution.

Phase 2: Org Setup & Configuration

- Salesforce **Developer Edition** org used.
- Profiles: Standard User, HR Manager.
- Roles: HR > Employee.
- Permission Sets: Access to Training object.

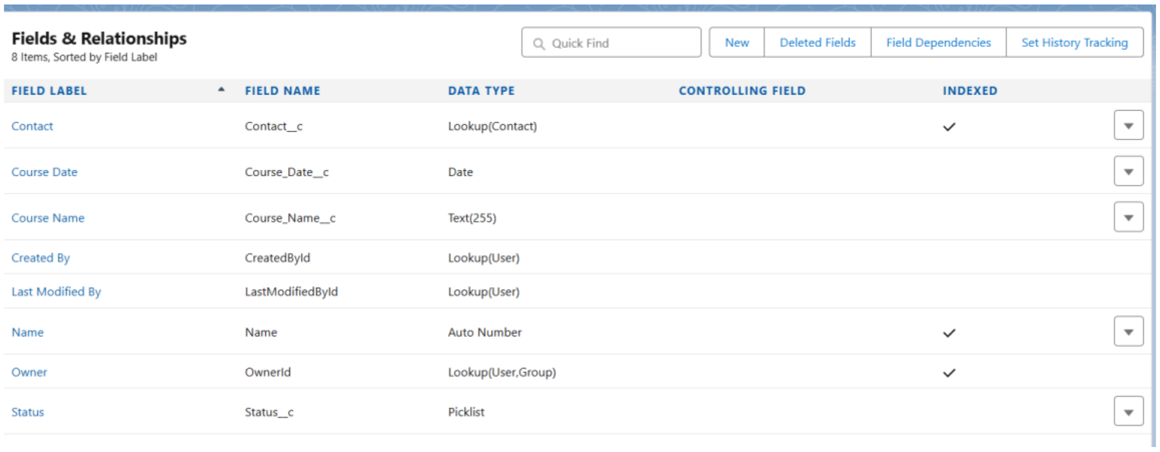
- OWD: Private for Training object, shared via lookup.



Details

Edit | Delete

Description	
API Name: Training__c	Enable Reports: ✓
Custom: ✓	Track Activities
Singular Label: Training	Track Field History
Plural Label: Trainings	Deployment Status: Deployed
	Help Settings: Standard salesforce.com Help Window



Fields & Relationships
8 Items, Sorted by Field Label

Quick Find | New | Deleted Fields | Field Dependencies | Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Contact	Contact__c	Lookup(Contact)		✓
Course Date	Course_Date__c	Date		
Course Name	Course_Name__c	Text(255)		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Name	Name	Auto Number		✓
Owner	OwnerId	Lookup(User,Group)		✓
Status	Status__c	Picklist		

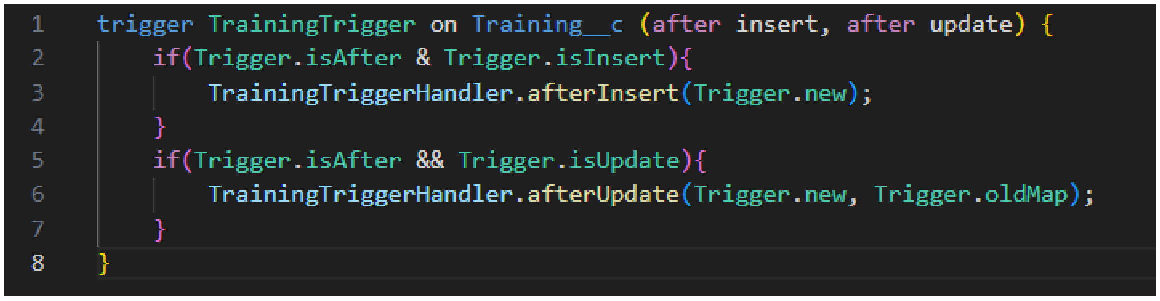
Phase 3: Data Modeling & Relationships

- **Custom Fields & Objects**:
    - On **Contact**: Training_Status__c (Picklist: Not Started, In Progress, Completed).
    - **Training__c** (Custom Object).
        - Fields:
            - Name (Auto Number)
            - Contact (Lookup → Contact)
            - Course_Date__c (Date)
            - Course_Name__c (Text)
            - Status__c (Picklist: Not Started, In Progress, Completed)

- Relationship: Contact ↔ Training (1-to-Many).

Phase 4: Process Automation (Admin)

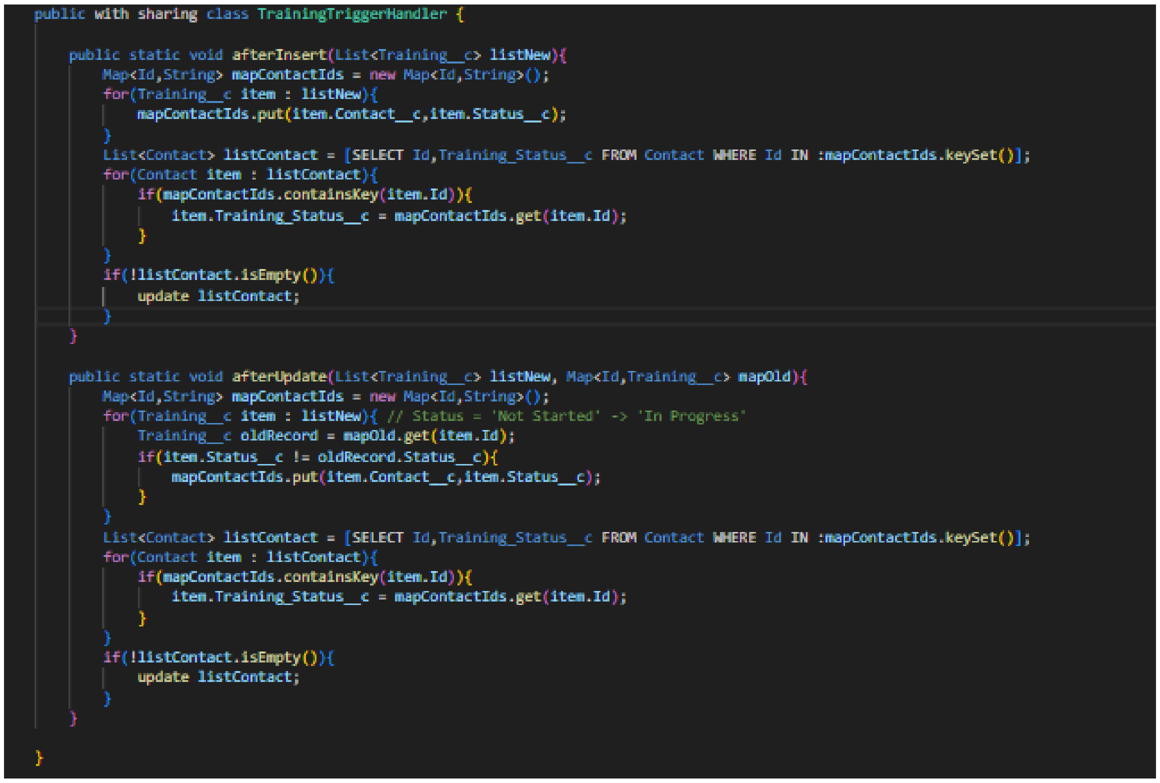
- Validation Rule: Ensure Course Date is not in the past when creating a training.
- Flows could be used but here we chose **Apex Trigger** for business logic.



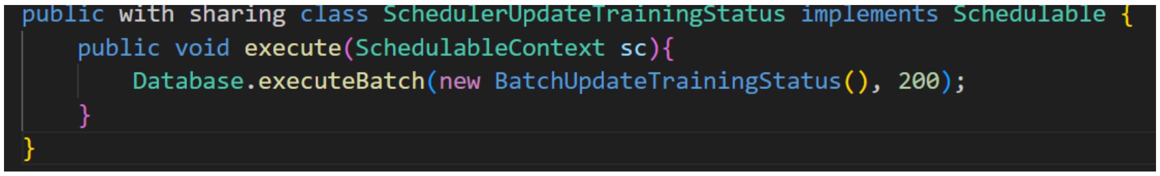
```
trigger TrainingTrigger on Training__c (after insert, after update) {
    if(Trigger.isAfter & Trigger.isInsert){
        TrainingTriggerHandler.afterInsert(Trigger.new);
    }
    if(Trigger.isAfter && Trigger.isUpdate){
        TrainingTriggerHandler.afterUpdate(Trigger.new, Trigger.oldMap);
    }
}
```

Phase 5: Apex Programming (Developer)

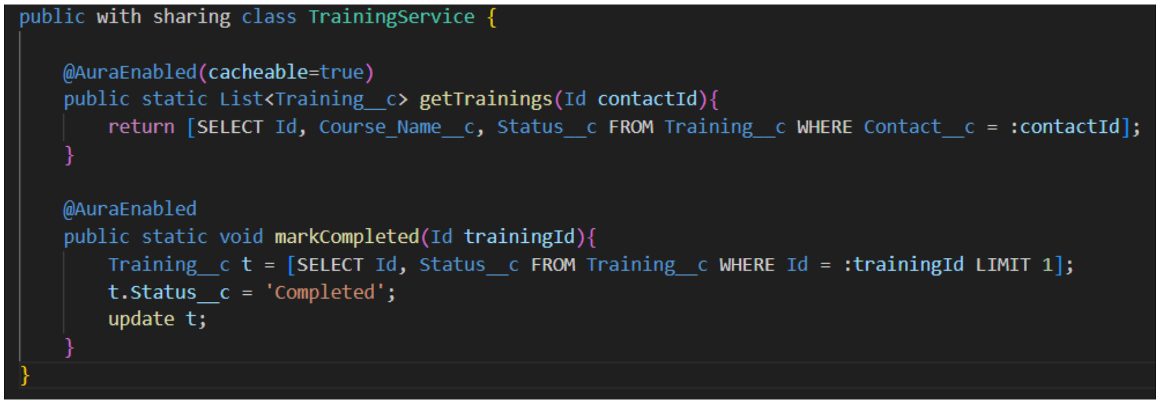
- **Trigger** on Training:
- After Insert: Update Contact’s Training_Status__c.
- After Update: Sync Contact’s Training_Status__c with Training record’s Status__c.
- **Batch Apex**:
- BatchUpdateTrainingStatus
- Finds Training records with Status = 'In Progress' and Course_Date < TODAY()-7.
- Updates them to Completed.
- **Scheduled Apex**:
- Job runs daily at **1 AM**.
- Calls BatchUpdateTrainingStatus.



```
public with sharing class TrainingTriggerHandler {

    public static void afterInsert(List<Training__c> listNew){
        Map<Id,String> mapContactIds = new Map<Id,String>();
        for(Training__c item : listNew){
            mapContactIds.put(item.Contact__c,item.Status__c);
        }
        List<Contact> listContact = [SELECT Id,Training_Status__c FROM Contact WHERE Id IN :mapContactIds.keySet()];
        for(Contact item : listContact){
            if(mapContactIds.containsKey(item.Id)){
                item.Training_Status__c = mapContactIds.get(item.Id);
            }
        }
        if(!listContact.isEmpty()){
            update listContact;
        }
    }

    public static void afterUpdate(List<Training__c> listNew, Map<Id,Training__c> mapOld){
        Map<Id,String> mapContactIds = new Map<Id,String>();
        for(Training__c item : listNew){ // Status = 'Not Started' -> 'In Progress'
            Training__c oldRecord = mapOld.get(item.Id);
            if(item.Status__c != oldRecord.Status__c){
                mapContactIds.put(item.Contact__c,item.Status__c);
            }
        }
        List<Contact> listContact = [SELECT Id,Training_Status__c FROM Contact WHERE Id IN :mapContactIds.keySet()];
        for(Contact item : listContact){
            if(mapContactIds.containsKey(item.Id)){
                item.Training_Status__c = mapContactIds.get(item.Id);
            }
        }
        if(!listContact.isEmpty()){
            update listContact;
        }
    }

}
```



```
public with sharing class SchedulerUpdateTrainingStatus implements Schedulable {
    public void execute(SchedulableContext sc){
        Database.executeBatch(new BatchUpdateTrainingStatus(), 200);
    }
}
```



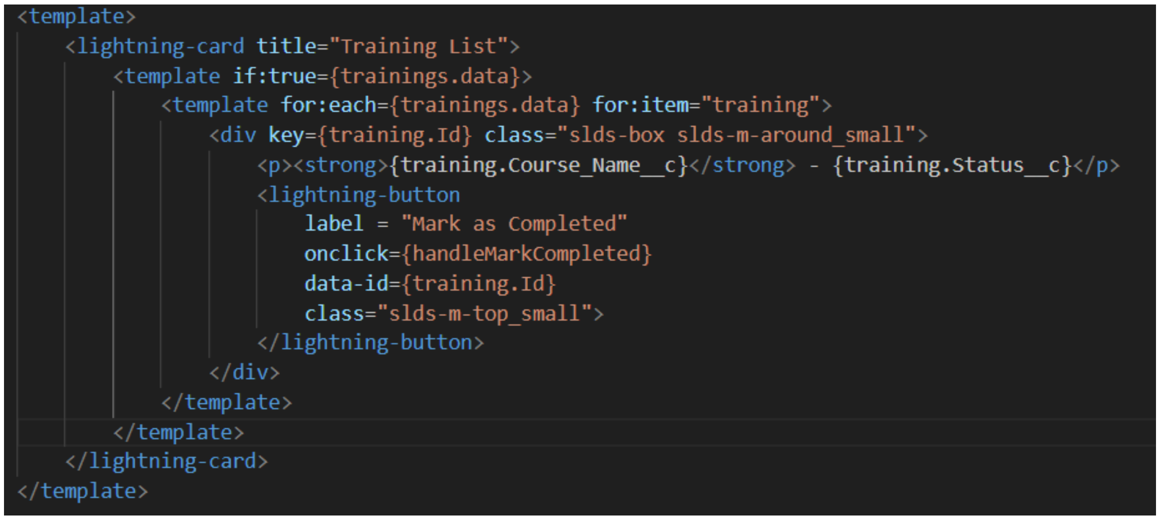
```
public with sharing class TrainingService {

    @AuraEnabled(cacheable=true)
    public static List<Training__c> getTrainings(Id contactId){
        return [SELECT Id, Course_Name__c, Status__c FROM Training__c WHERE Contact__c = :contactId];
    }

    @AuraEnabled
    public static void markCompleted(Id trainingId){
        Training__c t = [SELECT Id, Status__c FROM Training__c WHERE Id = :trainingId LIMIT 1];
        t.Status__c = 'Completed';
        update t;
    }
}
```

Phase 6: User Interface Development

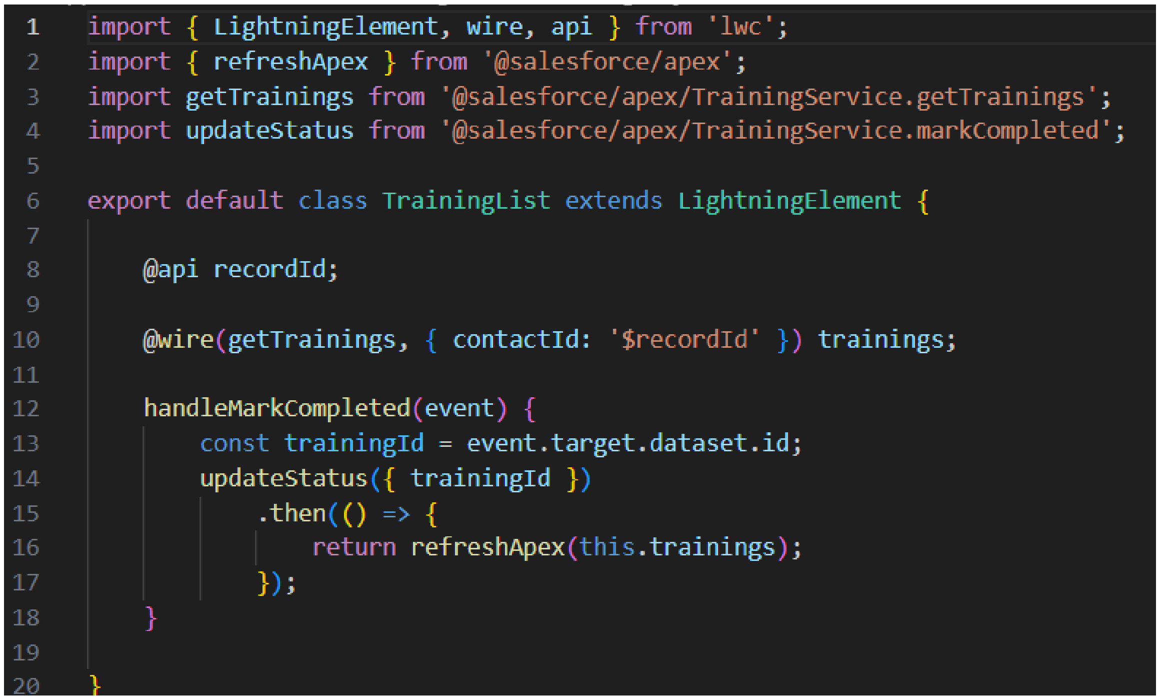
- **LWC: trainingList**

- Accepts recordId (Contact Id).

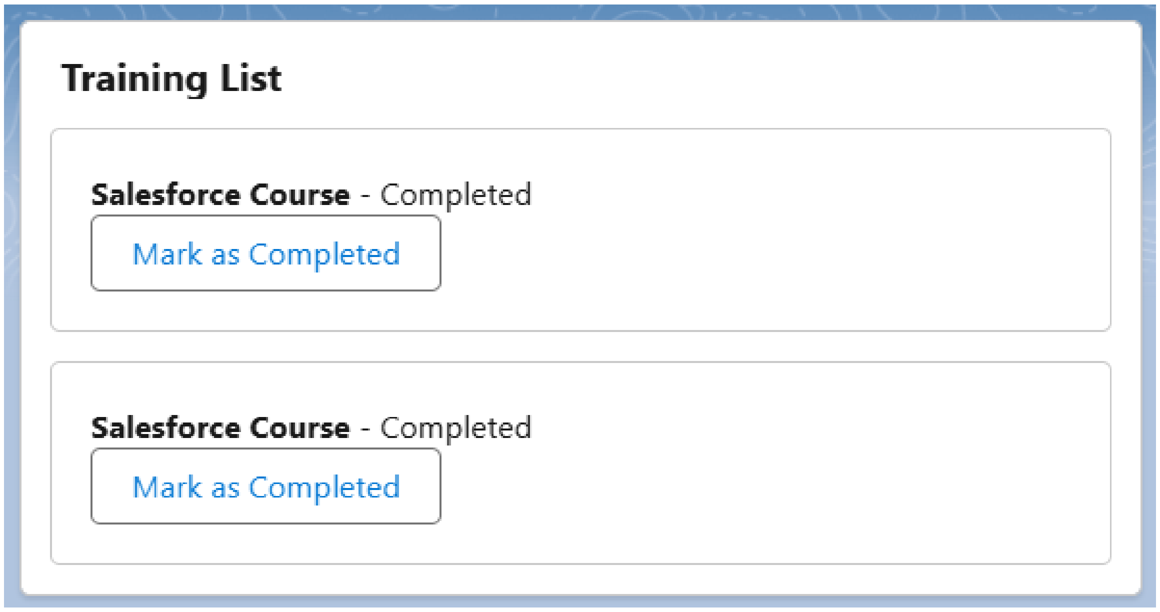
- Displays list of Training records for that Contact.
- Provides button → Mark training as Completed.
- Used @wire with Apex method to fetch trainings.
- Imperative Apex call on button click to update Status.



```html
<template>
    <lightning-card title="Training List">
        <template if:true={trainings.data}>
            <template for:each={trainings.data} for:item="training">
                <div key={training.Id} class="slds-box slds-m-around_small">
                    <p><strong>{training.Course_Name__c}</strong> - {training.Status__c}</p>
                    <lightning-button
                        label = "Mark as Completed"
                        onclick={handleMarkCompleted}
                        data-id={training.Id}
                        class="slds-m-top_small">
                    </lightning-button>
                </div>
            </template>
        </template>
    </lightning-card>
</template>
```



```javascript
import { LightningElement, wire, api } from 'lwc';
import { refreshApex } from '@salesforce/apex';
import getTrainings from '@salesforce/apex/TrainingService.getTrainings';
import updateStatus from '@salesforce/apex/TrainingService.markCompleted';

export default class TrainingList extends LightningElement {

    @api recordId;

    @wire(getTrainings, { contactId: '$recordId' }) trainings;

    handleMarkCompleted(event) {
        const trainingId = event.target.dataset.id;
        updateStatus({ trainingId })
            .then(() => {
                return refreshApex(this.trainings);
            });
    }

}
```



Training List

Salesforce Course - Completed

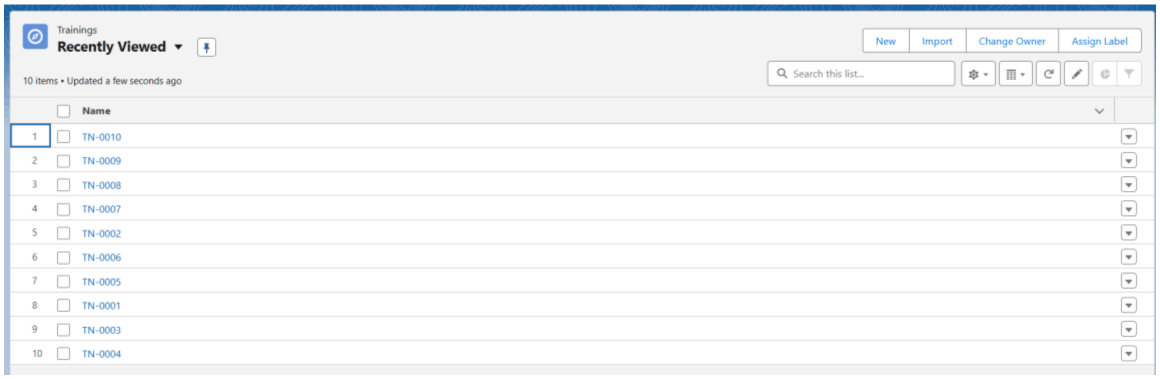
Salesforce Course - Completed

Phase 7: Integration & External Access

- No external integrations required in MVP.
- (Future enhancement: integrate with e-learning platforms via REST API).

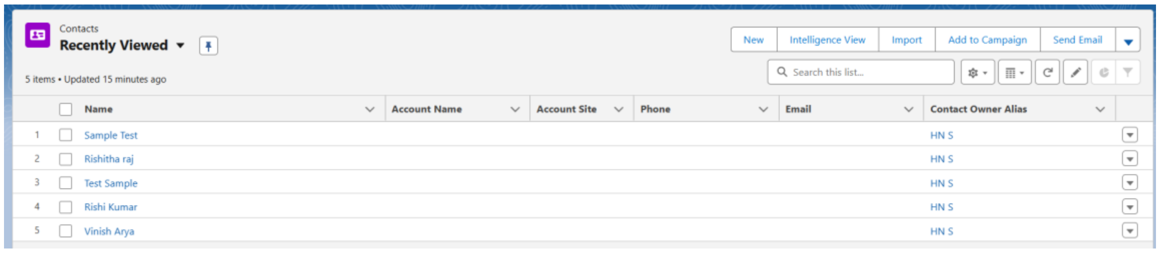
Phase 8: Data Management & Deployment

- Test Data: Loaded using Data Import Wizard.
- Metadata Deployment: SFDX + Change Sets.
- Backup: Data Export scheduled weekly.



10 items • Updated a few seconds ago

	Name
1	TN-0010
2	TN-0009
3	TN-0008
4	TN-0007
5	TN-0002
6	TN-0006
7	TN-0005
8	TN-0001
9	TN-0003
10	TN-0004


Contacts

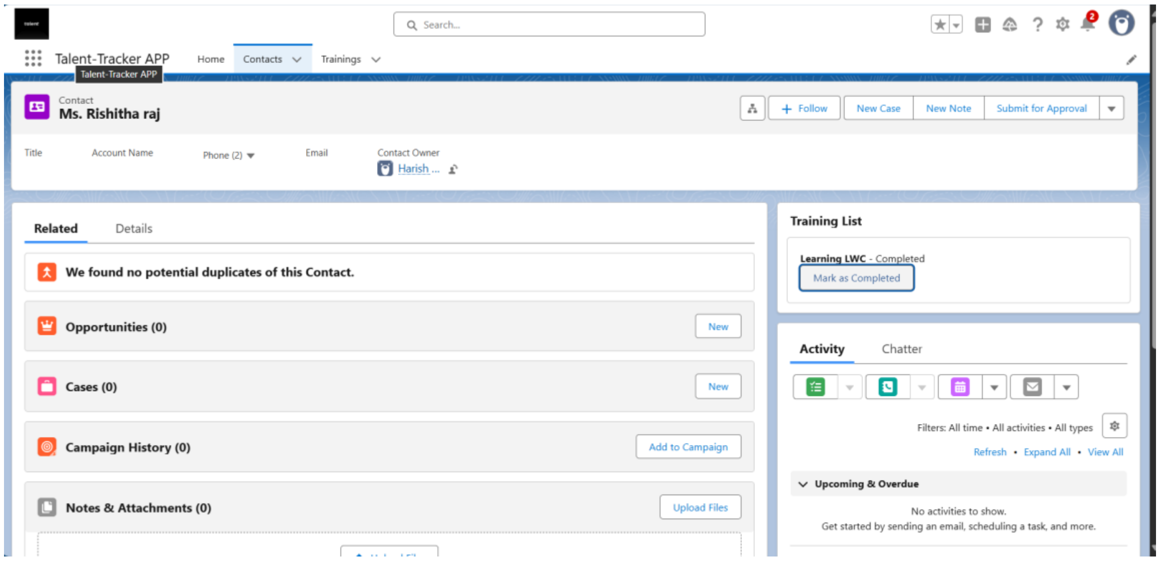
Recently Viewed

5 items • Updated 15 minutes ago

	Name	Account Name	Account Site	Phone	Email	Contact Owner Alias
1	Sample Test					HN S
2	Rishitha raj					HN S
3	Test Sample					HN S
4	Rishi Kumar					HN S
5	Vinish Arya					HN S

Phase 9: Reporting, Dashboards & Security Review

- Reports: Training by Status, Training by Contact.
- Dashboards: HR Dashboard → % Completed, In Progress, Overdue.
- Security:
- Field-Level Security for Training Status.
- Sharing rules for HR role.


Talent-Tracker APP

Contact

Ms. Rishitha raj

Title | Account Name | Phone (2) | Email | Contact Owner: Harish ...

Related | Details

We found no potential duplicates of this Contact.

Opportunities (0)

Cases (0)

Campaign History (0)

Notes & Attachments (0)

Training List

Learning LWC - Completed

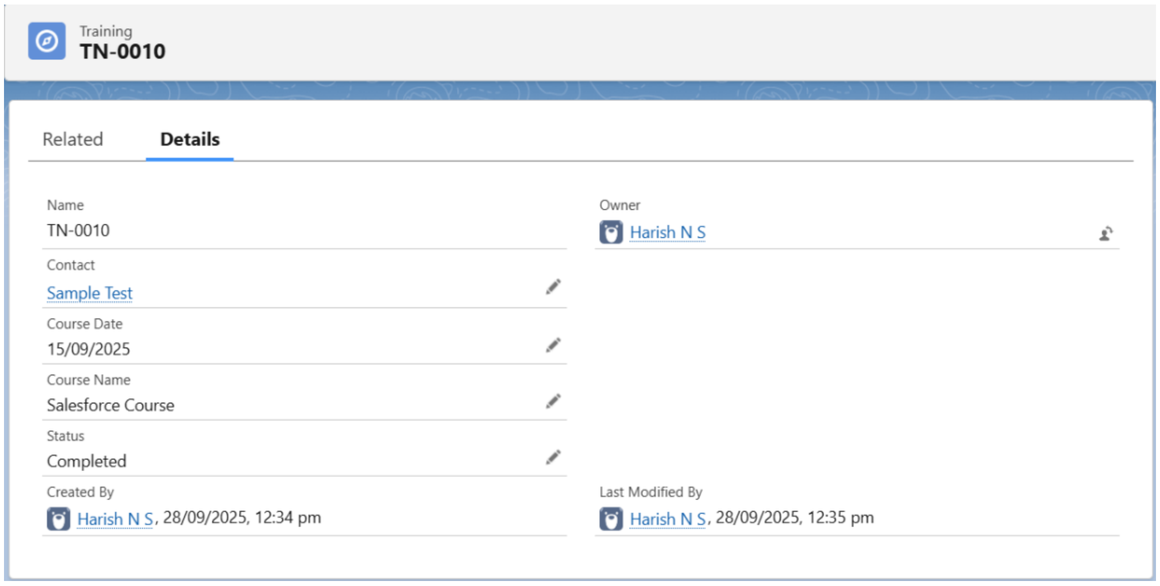
Mark as Completed

Activity | Chatter

Filters: All time • All activities • All types

Upcoming & Overdue

No activities to show.
Get started by sending an email, scheduling a task, and more.


Training
TN-0010

Related | Details

Name TN-0010	Owner Harish N S
Contact Sample Test	
Course Date 15/09/2025	
Course Name Salesforce Course	
Status Completed	
Created By Harish N S, 28/09/2025, 12:34 pm	Last Modified By Harish N S, 28/09/2025, 12:35 pm

Phase 10: Final Presentation & Demo Day

- **Demo Walkthrough**:
- Create Contact.
- Assign Trainings.
- Update Status manually or via batch job.
- Show Contact’s Training Status roll-up.
- **Client Q&A**: Cover automation & reporting.
- **Documentation**: Admin guide + developer code notes.
- **Portfolio Showcase**: Position project as L&D solution.