Phase 6 – User Interface Development

To provide a user-friendly interface for:

- Admins/trainers to manage trainings
- Managers to view training progress
- Contacts

Step 1: Standard UI Components Setup

Object Record Pages

For each object:

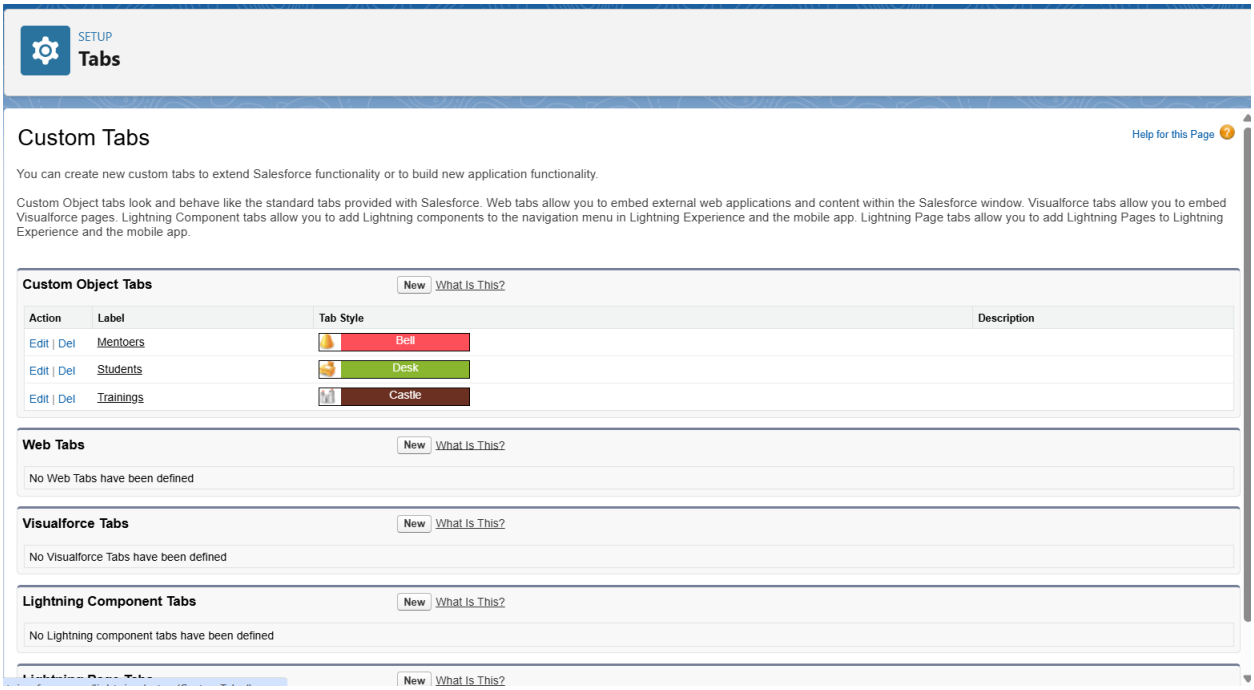
Contact

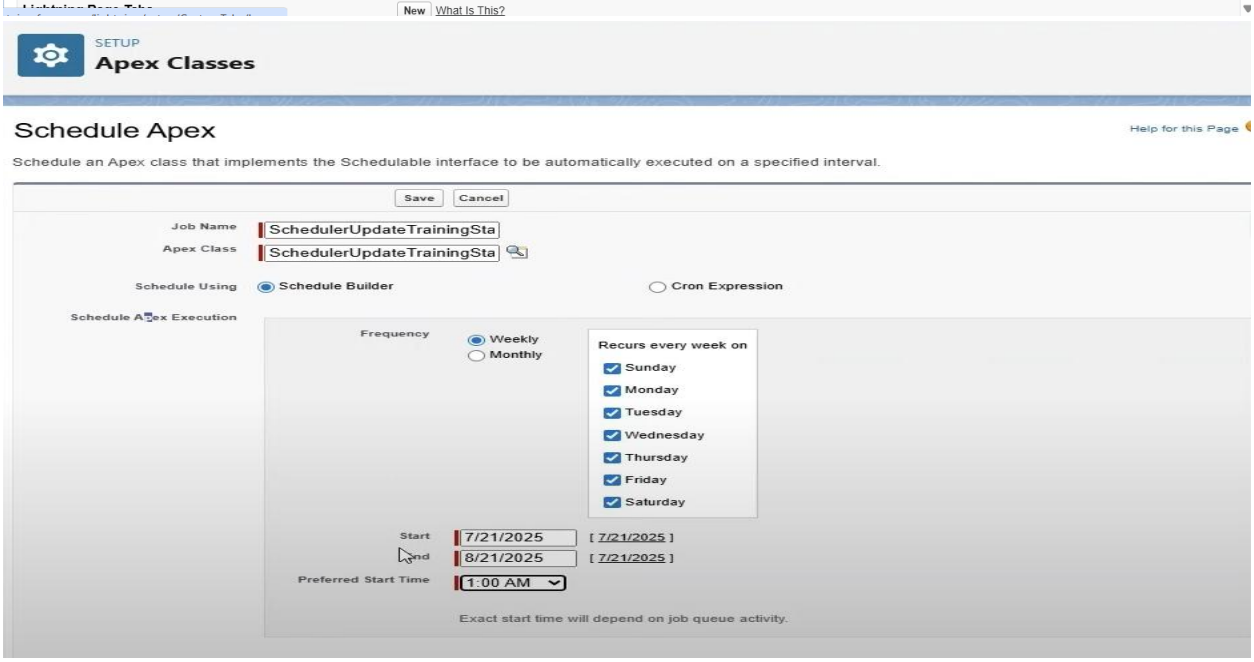
Add Related List: **Contact Trainings**

- **Training Session**: Add Related List: **Contact Trainings**
- **Contact Training**: Show fields like Score, Status, Certificate

🔧 **Steps:**

1. Go to **Object Manager**
2. Select Object → **Lightning Record Pages**
3. Click **Edit Page**
4. Drag and drop:
    - **Tabs**, **Field Sections**
    - **Related List – Single** or **Related List – Full**
    - **Reports/Dashboards** if needed


SETUP
Tabs
Custom Tabs
Help for this Page
You can create new custom tabs to extend Salesforce functionality or to build new application functionality.
Custom Object tabs look and behave like the standard tabs provided with Salesforce. Web tabs allow you to embed external web applications and content within the Salesforce window. Visualforce tabs allow you to embed Visualforce pages. Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app. Lightning Page tabs allow you to add Lightning Pages to Lightning Experience and the mobile app.
Custom Object Tabs
New
What Is This?
Action
Label
Tab Style
Description
Edit | Del
Mentoers
Bell
Edit | Del
Students
Desk
Edit | Del
Trainings
Castle
Web Tabs
New
What Is This?
No Web Tabs have been defined
Visualforce Tabs
New
What Is This?
No Visualforce Tabs have been defined
Lightning Component Tabs
New
What Is This?
No Lightning component tabs have been defined
New
What Is This?


SETUP
Apex Classes
Schedule Apex
Help for this Page
Schedule an Apex class that implements the Schedulable interface to be automatically executed on a specified interval.
Save
Cancel
Job Name
SchedulerUpdateTrainingSta
Apex Class
SchedulerUpdateTrainingSta
Schedule Using
Schedule Builder
Cron Expression
Schedule Apex Execution
Frequency
Weekly
Monthly
Recurs every week on
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Start
7/21/2025
[7/21/2025]
End
8/21/2025
[7/21/2025]
Preferred Start Time
1:00 AM
Exact start time will depend on job queue activity.

Step 2: Lightning Web Components (LWC)

Use LWC when:

- You need interactive dashboards
- You want to filter/display data dynamically
- You want to build custom forms

Training Summary for Contact

```
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 :mapC
        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 :mapC
        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 BatchUpdateTrainingStatus implements Database.Batchable<SObject> {

    public Database.QueryLocator start(Database.BatchableContext BC) {

        Date pre7Days = Date.today().addDays(-7);

        return Database.getQueryLocator([
            SELECT Id FROM Training__c
            WHERE Status__c = 'In Progress'
            AND Course_Date__c < :pre7Days
        ]);

    }

    public void execute(Database.BatchableContext BC, List<Training__c> scope) {
        for(Training__c item : scope){
            item.Status__c = 'Completed';
        }
        if(!scope.isEmpty()){
            update scope;
        }
    }

    public void finish(Database.BatchableContext BC) {}
}
```