Phase 10-Demo Video

(For Salesforce CRM Project — Event & Feedback Management)

Demo Video Link-

https://drive.google.com/file/d/1Ek4teHSDePe7nCg4lObj3Vpin09N_Lth/view?usp=sharing

1) Objective of QA Testing

To test all features of the project (Custom Objects, Workflows, Reports, Dashboards, Security Settings) to ensure that everything works as expected and there are no bugs/issues.

2) Test Environment

– Salesforce Developer Org (used for development and testing)
– Custom Objects: Event__c, ,Attendee__c, Feedback__c
– Features: Validation Rules, Workflow Rules, Reports, Dashboards, Security Settings, Apex Trigger

3) Test Scope

– Event Creation & Update
– Feedback Submission & Validation
– Workflow Email Alerts
– Reports & Dashboards Data Accuracy
– Security & Field Level Security
– Apex Trigger Execution

```
trigger EventFeedbackTrigger on Feedback__c (after insert, after update, after delete, after undelete) {

  Set<Id> eventIds = new Set<Id>();



  if(Trigger.isInsert || Trigger.isUndelete){

    for(Feedback__c f : Trigger.new) if(f.Event__c != null) eventIds.add(f.Event__c);

  }
```

```
if(Trigger.isDelete){
  for(Feedback__c f : Trigger.old) if(f.Event__c != null) eventIds.add(f.Event__c);
}

if(Trigger.isUpdate){
  // If Event lookup changed, include both old and new Event Ids
  for(Feedback__c fNew : Trigger.new){
    Feedback__c fOld = Trigger.oldMap.get(fNew.Id);
    if(fNew.Event__c != null) eventIds.add(fNew.Event__c);
    if(fOld.Event__c != null) eventIds.add(fOld.Event__c);
  }
}

if(eventIds.isEmpty()) return;

// Use aggregate query to get counts per Event (more efficient than subquery size)
Map<Id, Integer> eventToCount = new Map<Id, Integer>();
for(AggregateResult ar : [
  SELECT Event__c e, COUNT(Id) cnt
  FROM Feedback__c
  WHERE Event__c IN :eventIds
  GROUP BY Event__c
]) {
  eventToCount.put( (Id) ar.get('e'), Integer.valueOf( String.valueOf(ar.get('cnt')) ) );
}
```

```
  // Prepare Event records to update (set 0 if not present in map)

  List<Event__c> eventsToUpdate = new List<Event__c>();

  for(Event__c ev : [SELECT Id, Feedback_Count__c FROM Event__c WHERE Id IN :eventIds])
{

    Integer cnt = eventToCount.containsKey(ev.Id) ? eventToCount.get(ev.Id) : 0;

    ev.Feedback_Count__c = cnt;

    eventsToUpdate.add(ev);

  }


  if(!eventsToUpdate.isEmpty()) update eventsToUpdate;

}
```

4) Test Cases

Test Case ID	Test Scenario	Steps	Expected Result	Status
TC-01	Event Record Creation	Create a new Event record (fill all mandatory fields)	Record is successfully created	Pass/Fail
TC-02	Feedback Validation Rule	Try creating Feedback without an Attendee/Linked Event	Error message “Feedback must be associated with an Attendee” appears	Pass/Fail

TC-03	Dashboard/Report Data	Open Dashboard and verify if data is accurate	Correct Event & Feedback data is displayed	Pass/Fail
TC-04	Security Settings	Login with a different profile and check FLS & Sharing Settings	Restricted fields are hidden; only permitted data is accessible	Pass/Fail
TC-05	Apex Trigger Test	Run Test Classes from Developer Console	100% Code Coverage achieved & all assertions pass	Pass/Fail

5) Bug Tracking & Fixes

– Maintain a “Bug Log” during QA testing (Issue description, Steps to reproduce, Fix applied, Date fixed).
– Re-test the fixed version to ensure the issue is resolved.

6) QA Summary

– All critical workflows tested.
– Validation Rules working as expected.
– Email alerts & reports verified.
– Security & Field Level Security tested.
– Apex Test Classes executed (100% code coverage achieved).