# PROJECTFILES

| | |
|---|---|
| Date | 25June2025 |
| TeamID | LTVIP2025TMID31104 |
| ProjectName | Field Service Work Order Optimization |

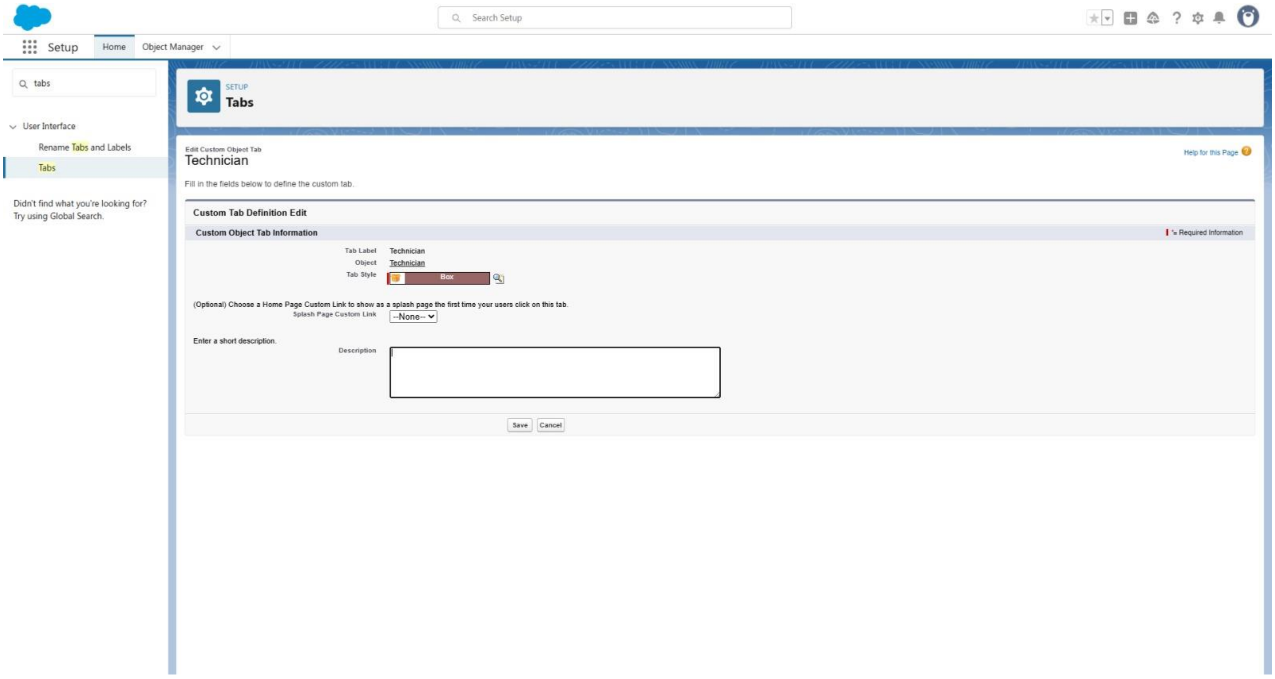
## Custom Salesforce Objects:

- Technician
- WorkOrder
- Assignment

## Custom Tabs and Lightning App:

- Tabs created for: Technician, WorkOrder, Assignment

## Technician Tab:



Edit Custom Object Tab

Technician

Fill in the fields below to define the custom tab.

Custom Tab Definition Edit

Custom Object Tab Information

| = Required Information

Tab Label: Technician

Object: Technician

Tab Style: Box

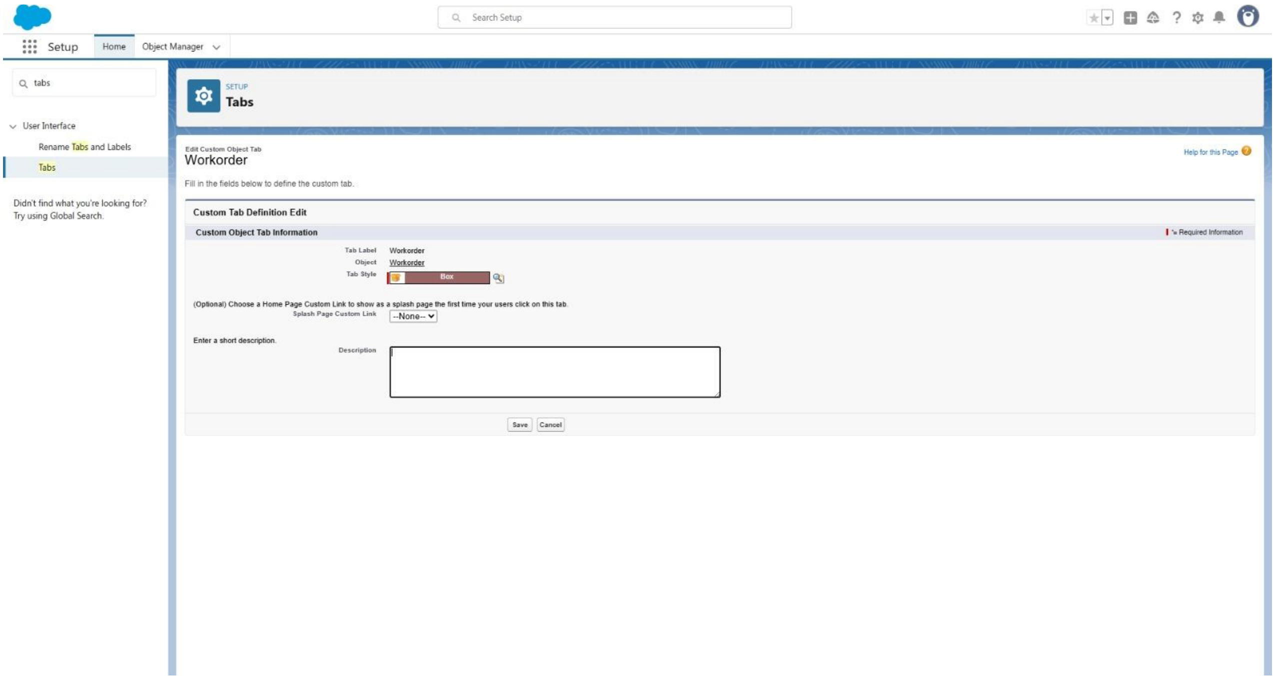
(Optional) Choose a Home Page Custom Link to show as a splash page the first time your users click on this tab.

Splash Page Custom Link: --None--

Enter a short description.

Description:

## WorkOrder Tab:



Edit Custom Object Tab

Workorder

Fill in the fields below to define the custom tab.

Custom Tab Definition Edit

Custom Object Tab Information

= Required Information

Tab Label: Workorder

Object: Workorder

Tab Style: Box

(Optional) Choose a Home Page Custom Link to show as a splash page the first time your users click on this tab.

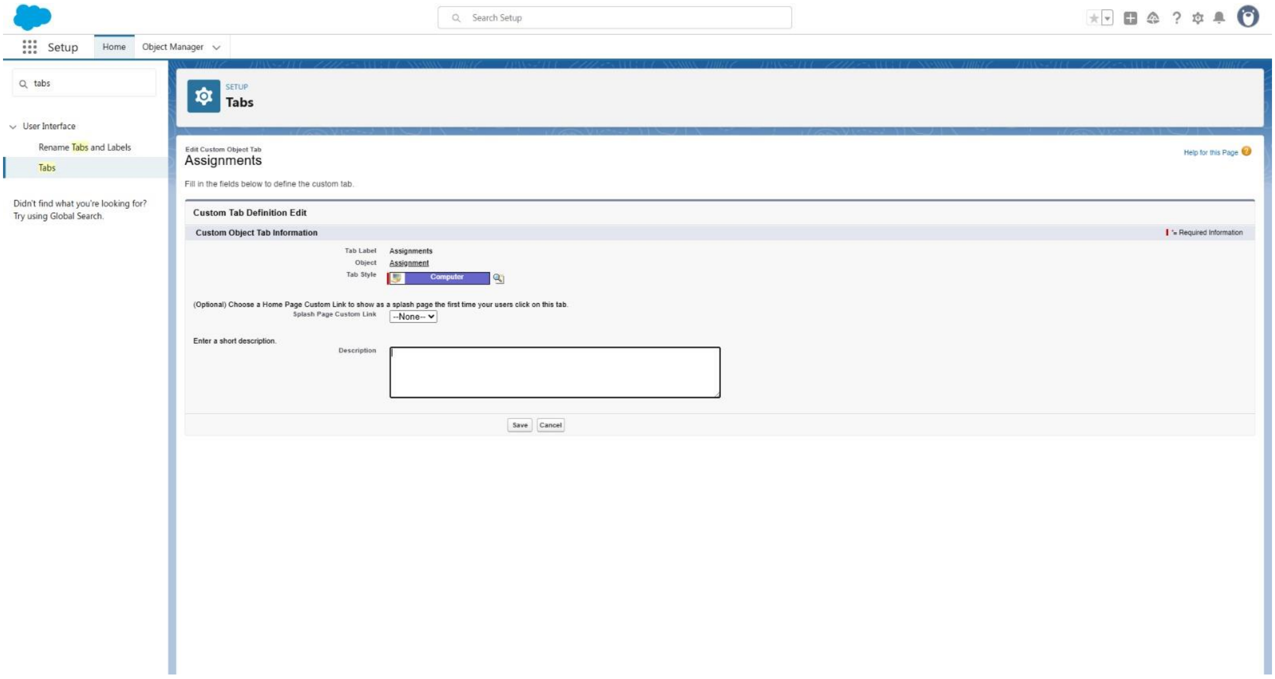
Splash Page Custom Link: --None--

Enter a short description.

Description:

Save Cancel

## Assignment Tab:



Edit Custom Object Tab

Assignments

Fill in the fields below to define the custom tab.

Custom Tab Definition Edit

Custom Object Tab Information

= Required Information

Tab Label: Assignments

Object: Assignment

Tab Style: Computer

(Optional) Choose a Home Page Custom Link to show as a splash page the first time your users click on this tab.

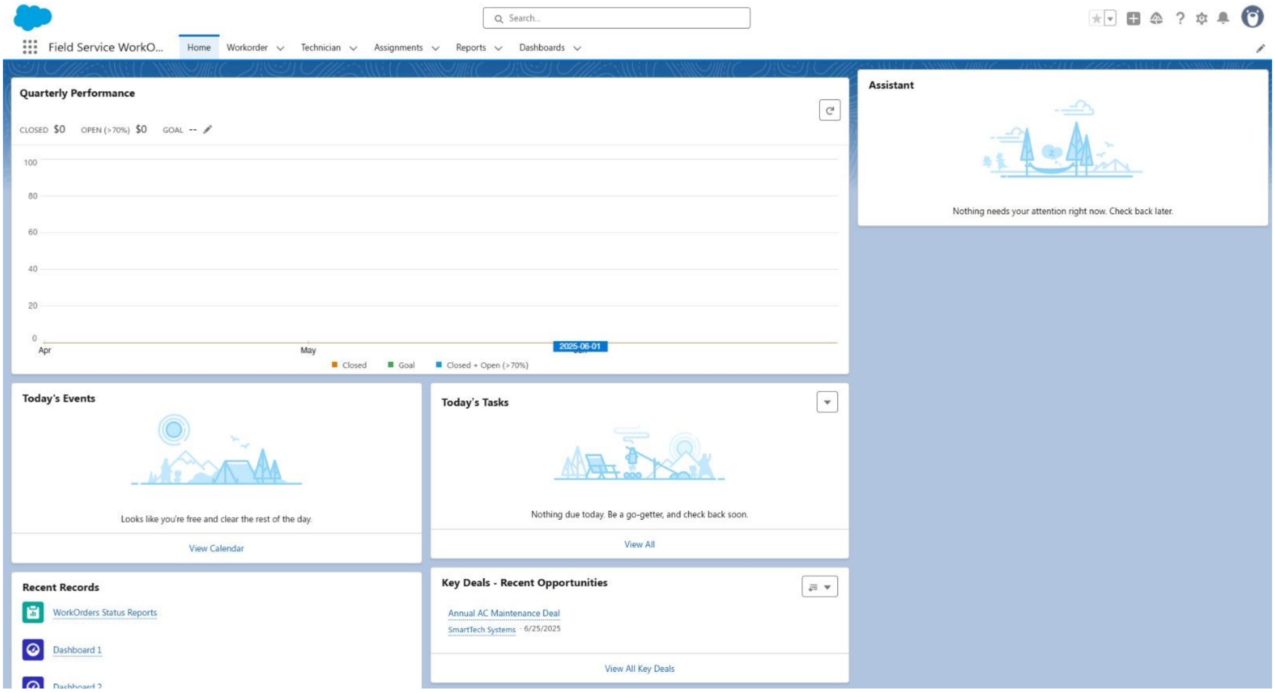
Splash Page Custom Link: --None--

Enter a short description.

Description:

Save Cancel

- Lightning App: **Field Service WorkOrder Optimization**
    - Navigation items: Home, WorkOrder, Technician, Assignment, Reports, Dashboard



Nothing needs your attention right now. Check back later.

Looks like you're free and clear the rest of the day.

Nothing due today. Be a go-getter, and check back soon.

WorkOrders Status Reports

Dashboard 1

Dashboard 2

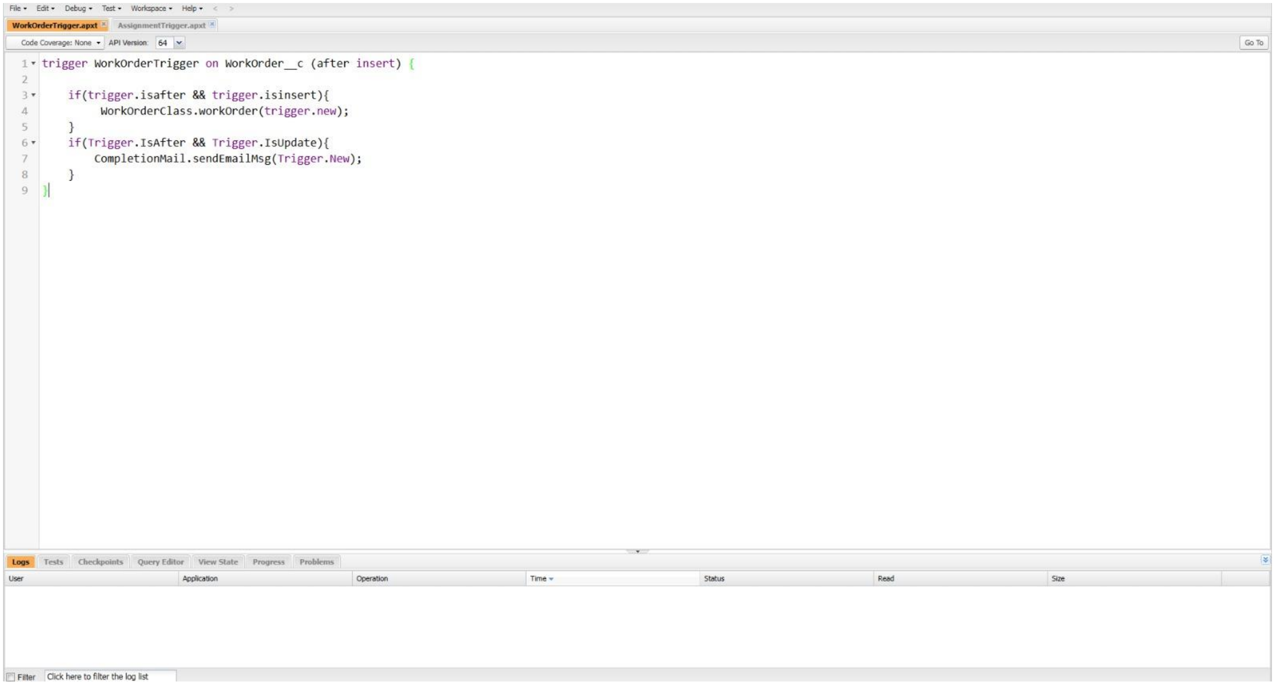
Annual AC Maintenance Deal

SmartTech Systems · 6/25/2025

## Apex Code:

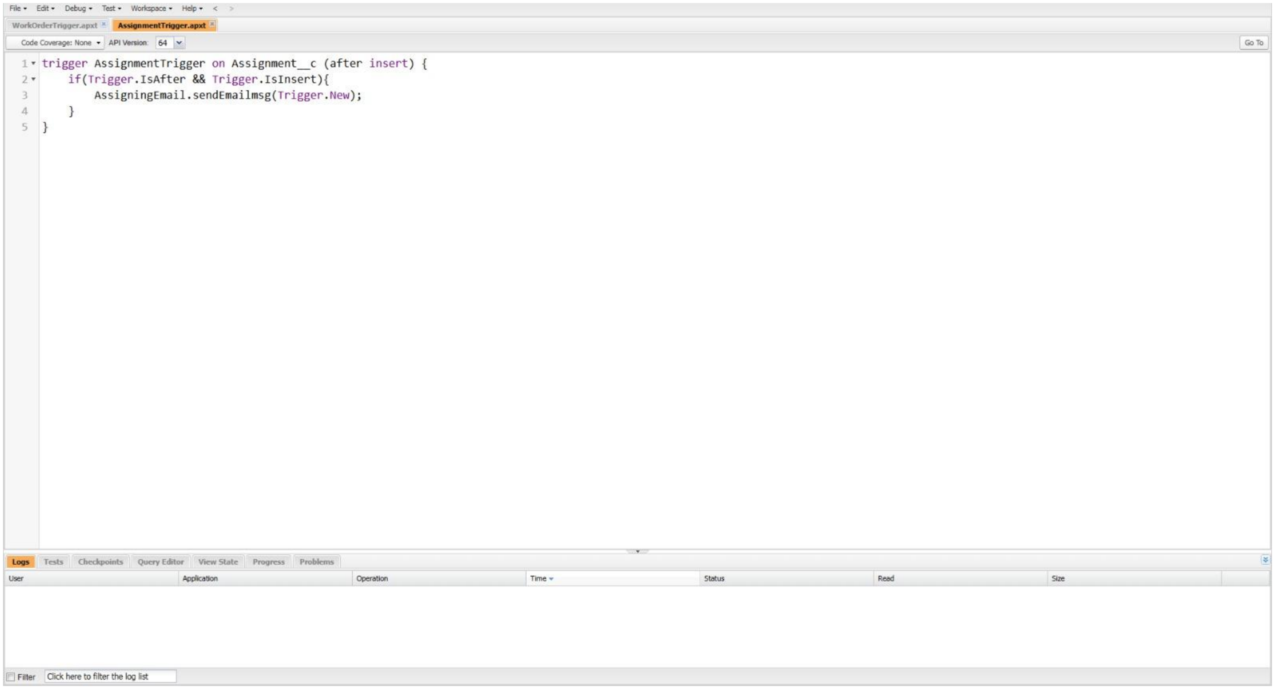
## Apex Triggers:

- WorkOrderTrigger
    - After Insert: Assign technician
    - After Update: Send completion email



```
trigger WorkOrderTrigger on WorkOrder__c (after insert) {

    if(trigger.isafter && trigger.isinsert){
         WorkOrderClass.workOrder(trigger.new);
    }
    if(Trigger.IsAfter && Trigger.IsUpdate){
        CompletionMail.sendEmailMsg(Trigger.New);
    }
}
```

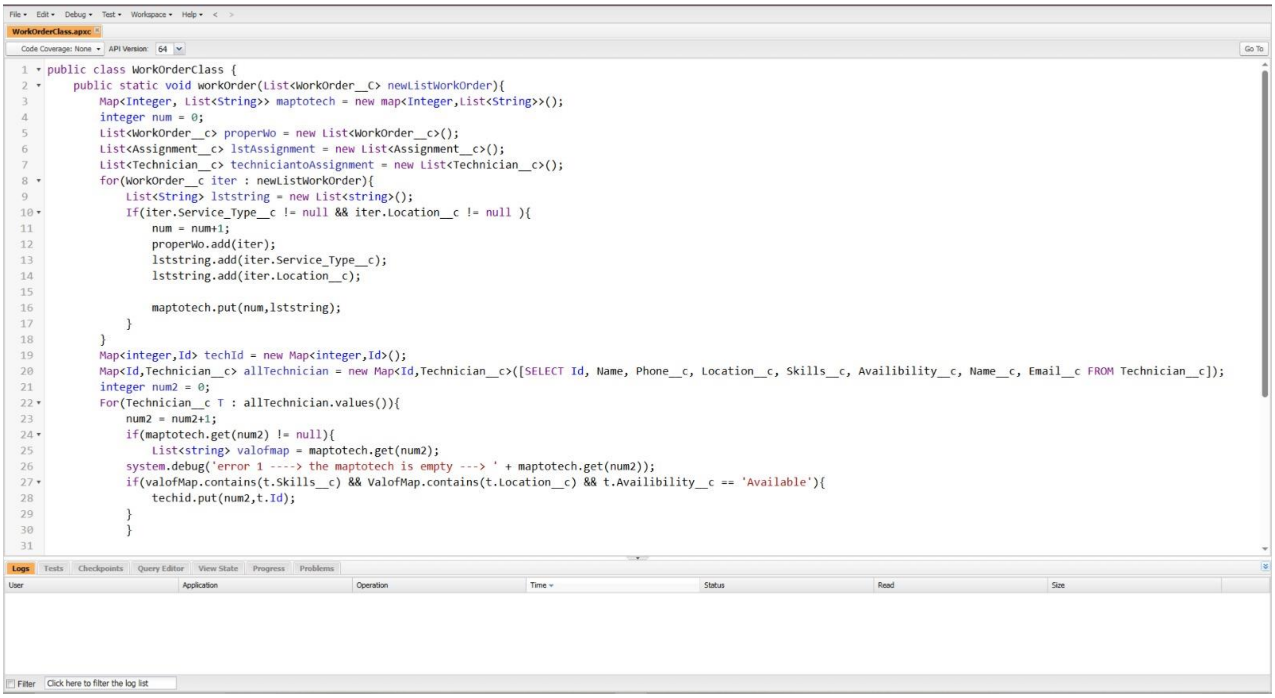
- AssignmentTrigger
  - After Insert: Send assignment email to technician



```
trigger AssignmentTrigger on Assignment__c (after insert) {
    if(Trigger.IsAfter && Trigger.IsInsert){
        AssigningEmail.sendEmailmsg(Trigger.New);
    }
}
```

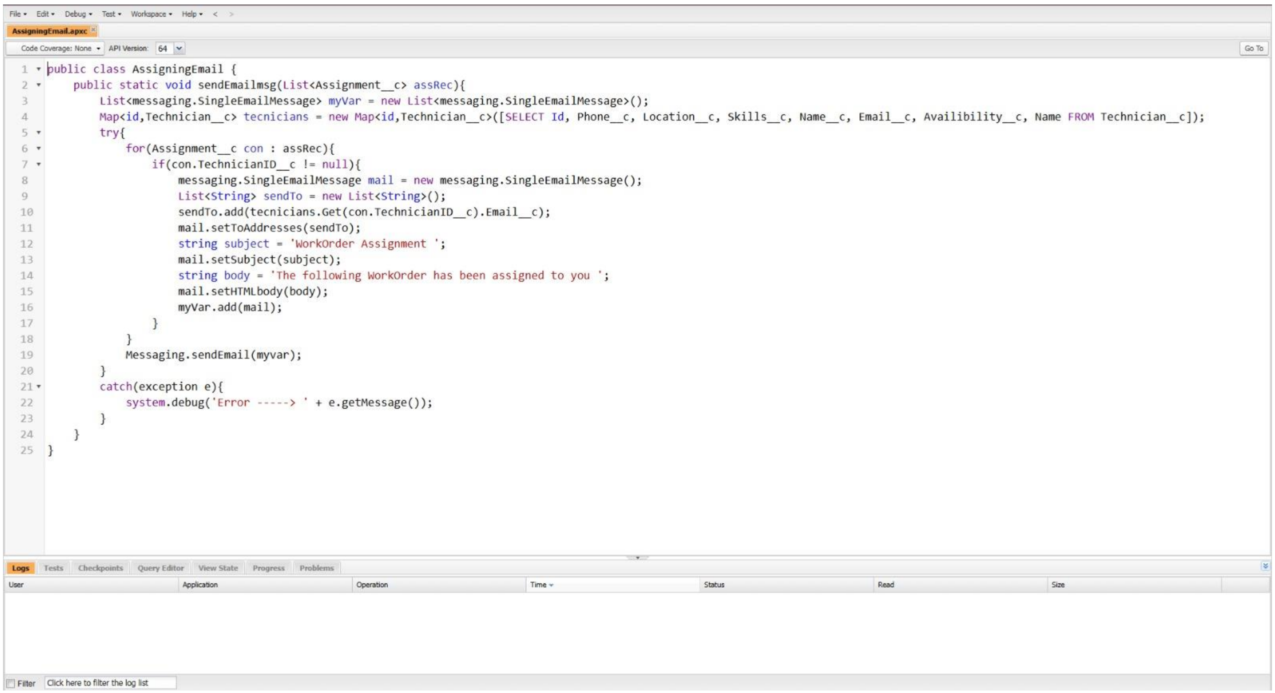
**Apex Classes:**

- WorkOrderClass: Handles technician assignment logic


File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
WorkOrderClass.apxc
Code Coverage: None ▾ API Version: 64
Go To
Logs Tests Checkpoints Query Editor View State Progress Problems
User Application Operation Time ▾ Status Read Size
Filter Click here to filter the log list

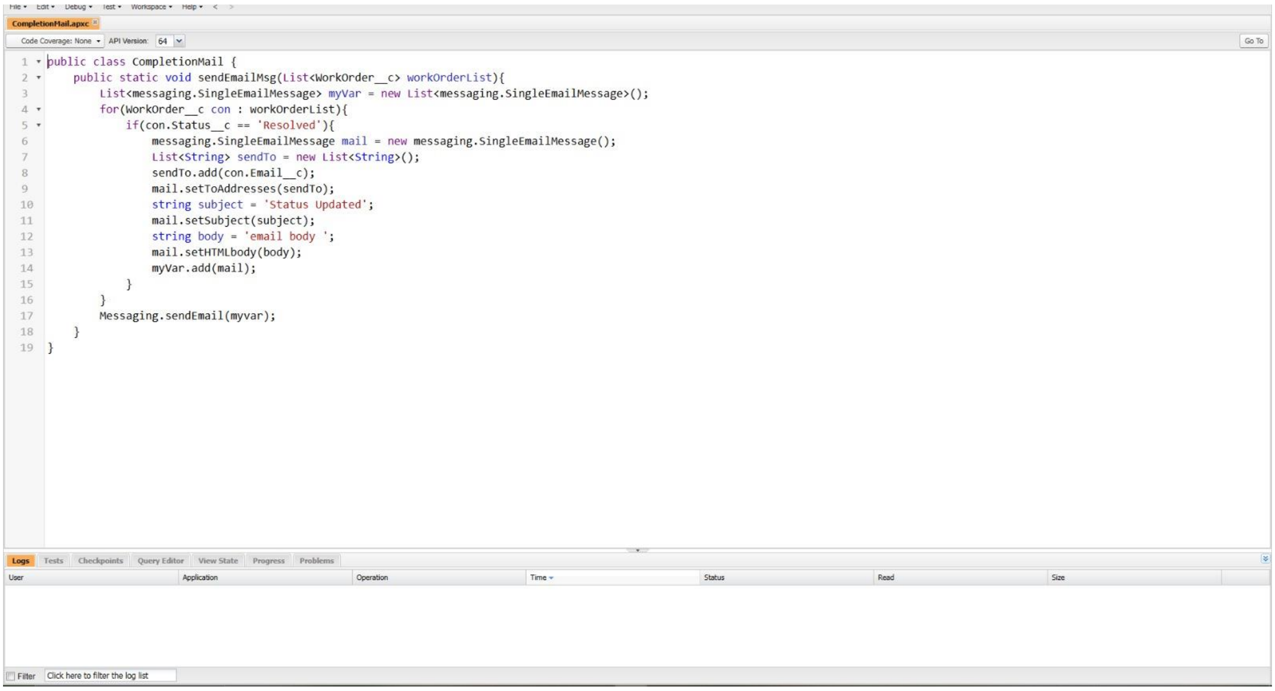
```
public class WorkOrderClass {
    public static void workOrder(List<WorkOrder__C> newListWorkOrder){
        Map<Integer, List<String>> maptotech = new map<Integer,List<String>>();
        integer num = 0;
        List<WorkOrder__c> properWo = new List<WorkOrder__c>();
        List<Assignment__c> lstAssignment = new List<Assignment__c>();
        List<Technician__c> techniciantoAssignment = new List<Technician__c>();
        for(WorkOrder__c iter : newListWorkOrder){
            List<String> lststring = new List<string>();
            If(iter.Service_Type__c != null && iter.Location__c != null ){
                num = num+1;
                properWo.add(iter);
                lststring.add(iter.Service_Type__c);
                lststring.add(iter.Location__c);

                maptotech.put(num,lststring);
            }
        }
        Map<integer,Id> techId = new Map<integer,Id>();
        Map<Id,Technician__c> allTechnician = new Map<Id,Technician__c>([SELECT Id, Name, Phone__c, Location__c, Skills__c, Availibility__c, Name__c, Email__c FROM Technician__c]);
        integer num2 = 0;
        For(Technician__c T : allTechnician.values()){
            num2 = num2+1;
            if(maptotech.get(num2) != null){
                List<string> valofmap = maptotech.get(num2);
            system.debug('error 1 ----> the maptotech is empty ---> ' + maptotech.get(num2));
            if(valofMap.contains(t.Skills__c) && ValofMap.contains(t.Location__c) && t.Availibility__c == 'Available'){
                techid.put(num2,t.Id);
            }
            }

```

- AssigningEmail: Sends technician assignment email


File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
AssigningEmail.apxc
Code Coverage: None ▾ API Version: 64
Go To
Logs Tests Checkpoints Query Editor View State Progress Problems
User Application Operation Time ▾ Status Read Size
Filter Click here to filter the log list

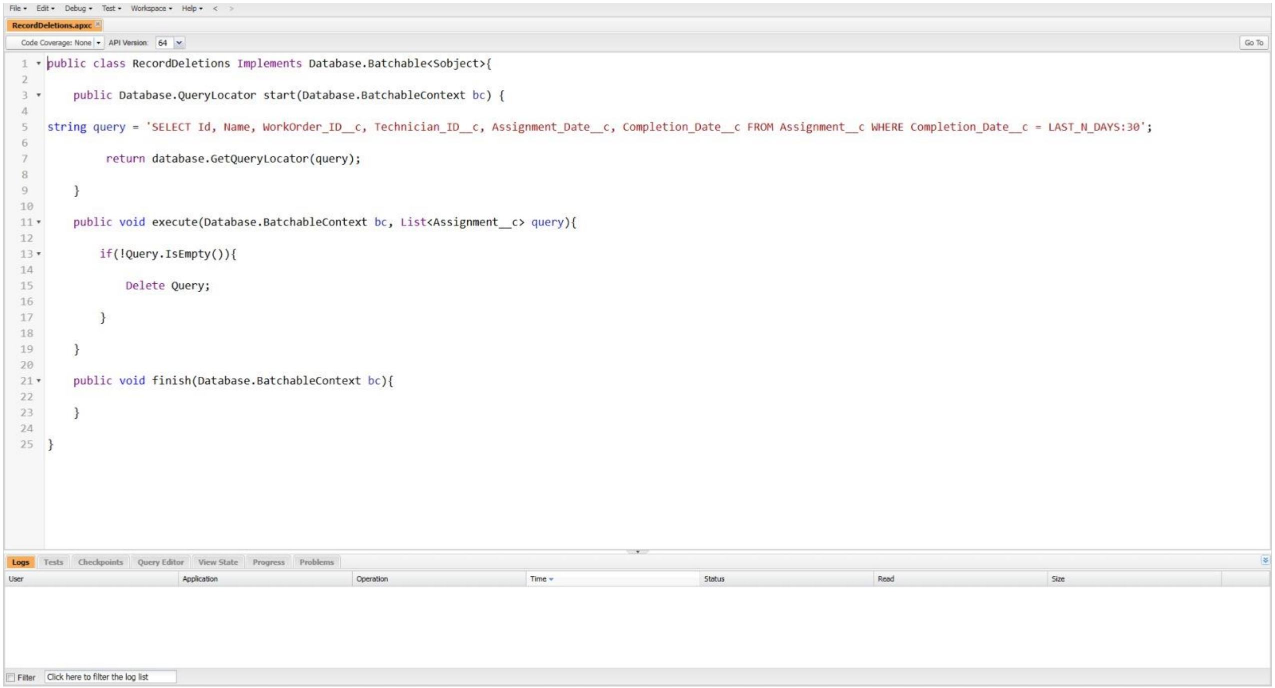
```
public class AssigningEmail {
    public static void sendEmailmsg(List<Assignment__c> assRec){
        List<messaging.SingleEmailMessage> myVar = new List<messaging.SingleEmailMessage>();
        Map<id,Technician__c> tecnicians = new Map<id,Technician__c>([SELECT Id, Phone__c, Location__c, Skills__c, Name__c, Email__c, Availibility__c, Name FROM Technician__c]);
        try{
            for(Assignment__c con : assRec){
                if(con.TechnicianID__c != null){
                    messaging.SingleEmailMessage mail = new messaging.SingleEmailMessage();
                    List<String> sendTo = new List<String>();
                    sendTo.add(tecnicians.Get(con.TechnicianID__c).Email__c);
                    mail.setToAddresses(sendTo);
                    string subject = 'WorkOrder Assignment ';
                    mail.setSubject(subject);
                    string body = 'The following WorkOrder has been assigned to you ';
                    mail.setHTMLbody(body);
                    myVar.add(mail);
                }
            }
            Messaging.sendEmail(myvar);
        }
        catch(exception e){
            system.debug('Error -----> ' + e.getMessage());
        }
    }
}
```

- CompletionMail: Sends work completion email to customer



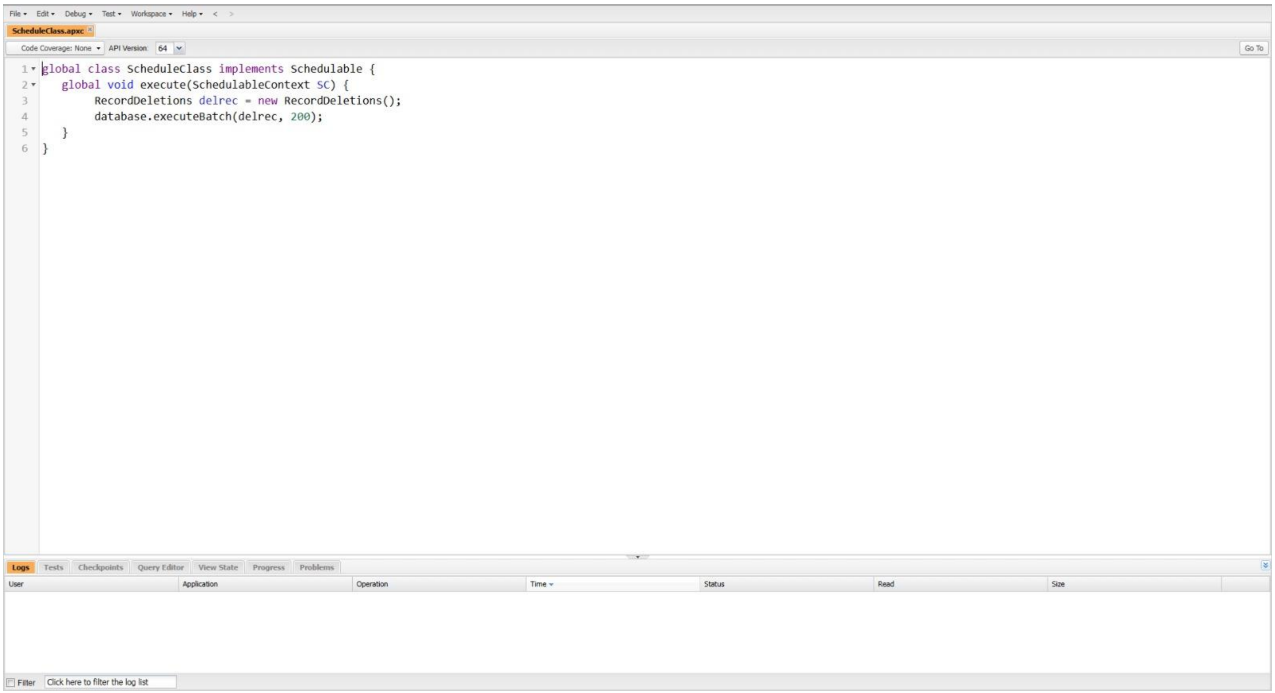
```
public class CompletionMail {
    public static void sendEmailMsg(List<WorkOrder__c> workOrderList){
        List<messaging.SingleEmailMessage> myVar = new List<messaging.SingleEmailMessage>();
        for(WorkOrder__c con : workOrderList){
            if(con.Status__c == 'Resolved'){
                messaging.SingleEmailMessage mail = new messaging.SingleEmailMessage();
                List<String> sendTo = new List<String>();
                sendTo.add(con.Email__c);
                mail.setToAddresses(sendTo);
                string subject = 'Status Updated';
                mail.setSubject(subject);
                string body = 'email body ';
                mail.setHTMLbody(body);
                myVar.add(mail);
            }
        }
        Messaging.sendEmail(myvar);
    }
}
```

- RecordDeletions: Batch class to delete old resolved assignments



```
public class RecordDeletions Implements Database.Batchable<Sobject>{

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

string query = 'SELECT Id, Name, WorkOrder_ID__c, Technician_ID__c, Assignment_Date__c, Completion_Date__c FROM Assignment__c WHERE Completion_Date__c = LAST_N_DAYS:30';

         return database.GetQueryLocator(query);

    }

    public void execute(Database.BatchableContext bc, List<Assignment__c> query){

        if(!Query.IsEmpty()){

            Delete Query;

        }

    }

    public void finish(Database.BatchableContext bc){

    }

}
```

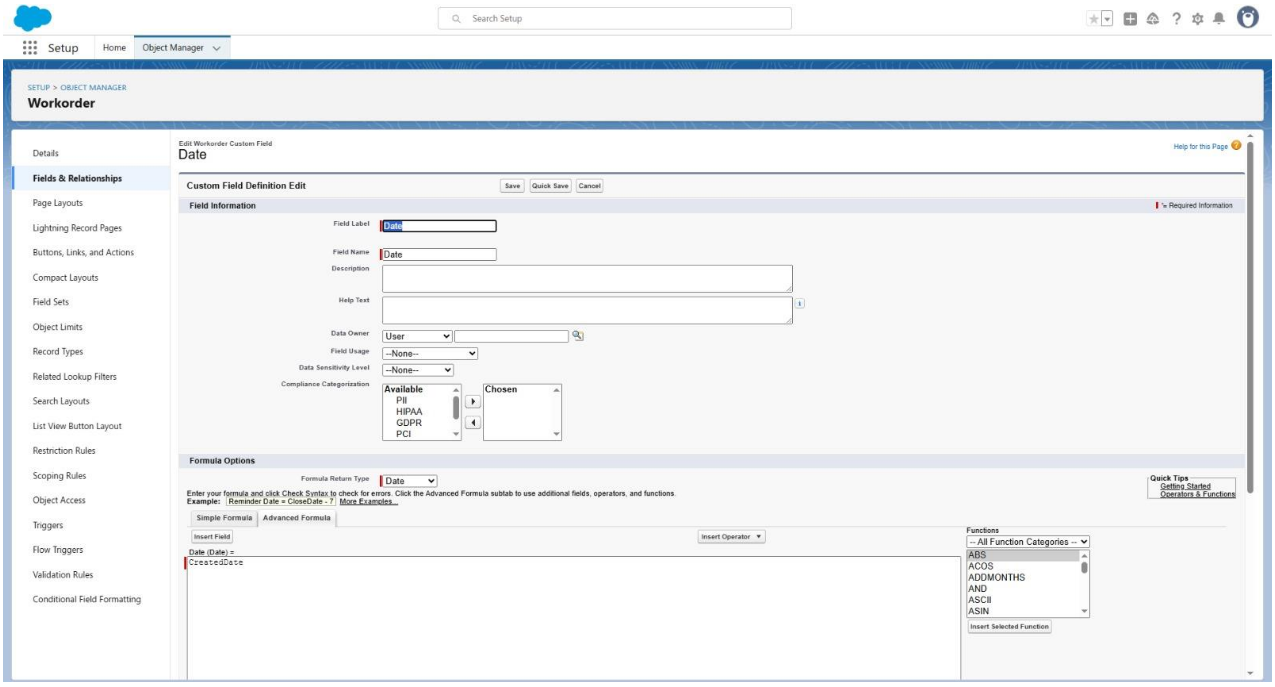
- ScheduleClass: Schedules the batch deletion monthly



```
global class ScheduleClass implements Schedulable {
    global void execute(SchedulableContext SC) {
         RecordDeletions delrec = new RecordDeletions();
         database.executeBatch(delrec, 200);
    }
}
```

## Formula Fields:

- WorkOrder.Date: Formula field (return type: Date) = CreatedDate



Edit Workorder Custom Field
Date

Custom Field Definition Edit

Field Information

Field Label: Date

Field Name: Date

Description:

Help Text:

Data Owner: User

Field Usage: --None--

Data Sensitivity Level: --None--

Compliance Categorization: Available: PII, HIPAA, GDPR, PCI; Chosen:

Formula Options

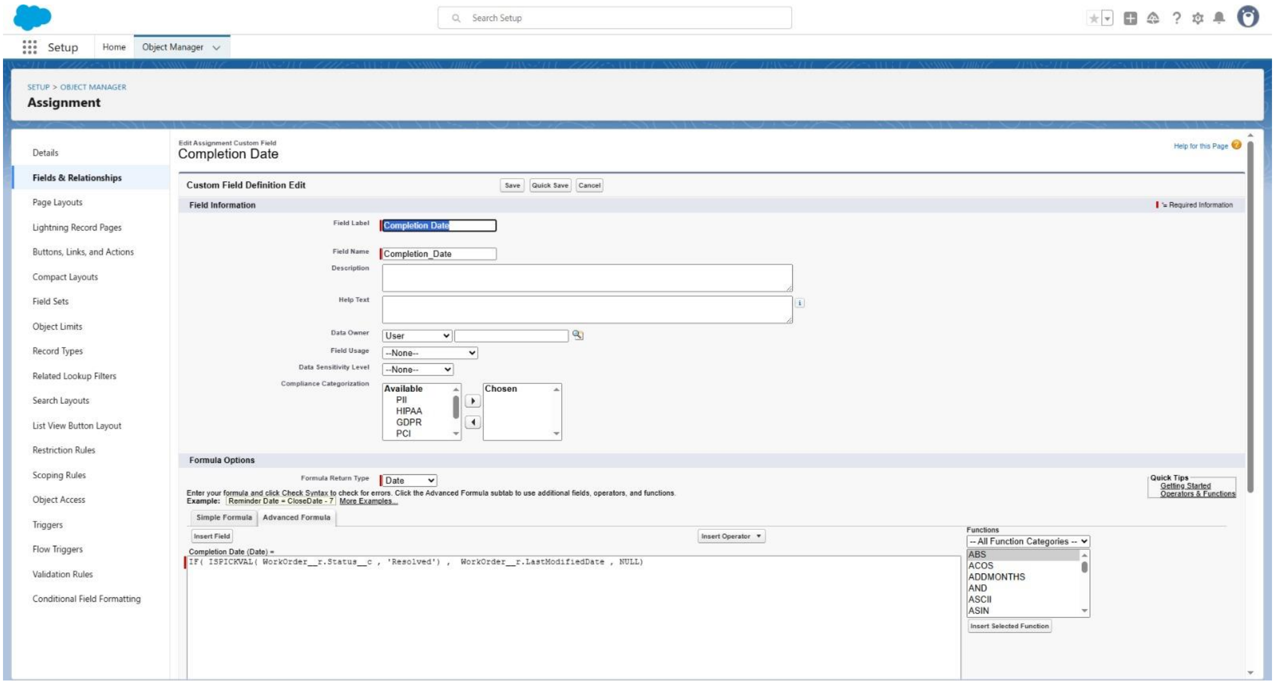
Formula Return Type: Date

Enter your formula and click Check Syntax to check for errors. Click the Advanced Formula subtab to use additional fields, operators, and functions.
Example: Reminder Date = CloseDate - 7 More Examples...

Date (Date) =

```
CreatedDate
```

- Assignment.Completion_Date: Conditional formula based on WorkOrder status



Assignment

Edit Assignment Custom Field

Completion Date

Custom Field Definition Edit

Field Information

| = Required Information

Field Label: Completion Date

Field Name: Completion_Date

Description:

Help Text:

Data Owner: User

Field Usage: --None--

Data Sensitivity Level: --None--

Compliance Categorization: Available: PII, HIPAA, GDPR, PCI; Chosen:

Formula Options

Formula Return Type: Date

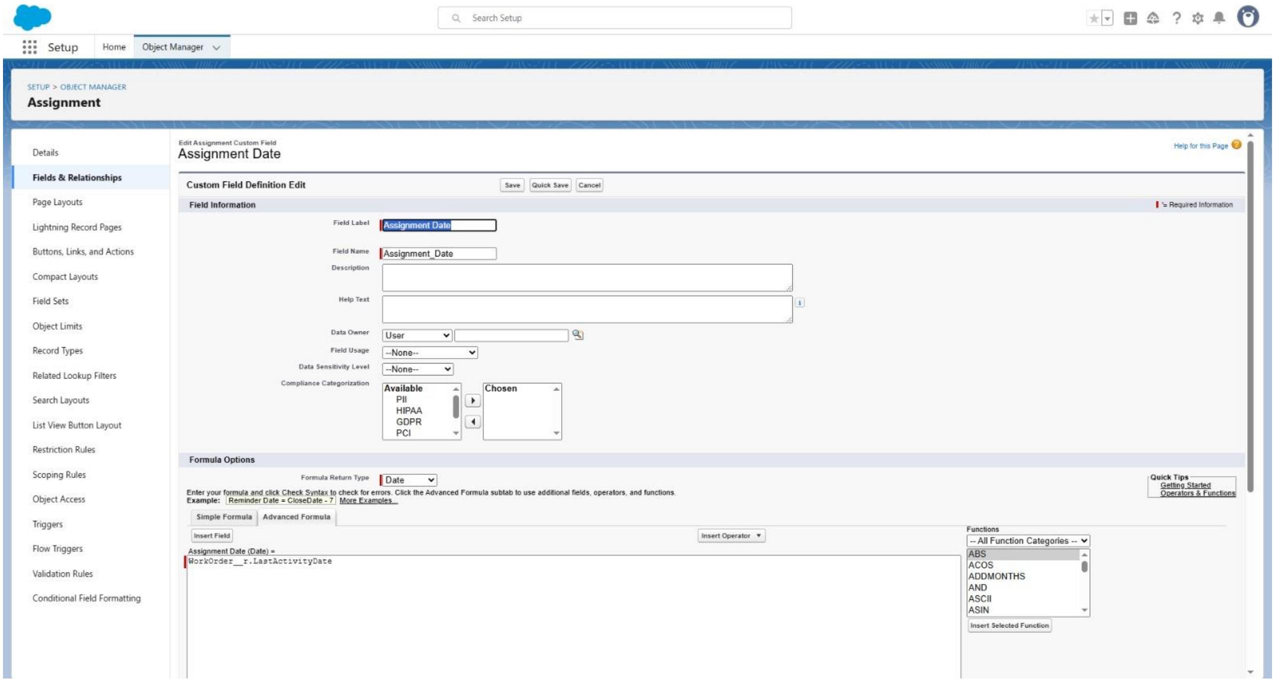
Enter your formula and click Check Syntax to check for errors. Click the Advanced Formula subtab to use additional fields, operators, and functions.
**Example:** Reminder Date = CloseDate - 7

Functions: -- All Function Categories --
ABS
ACOS
ADDMONTHS
AND
ASCII
ASIN

Completion Date (Date) =

```
IF( ISPICKVAL( WorkOrder__r.Status__c , 'Resolved') ,  WorkOrder__r.LastModifiedDate , NULL)
```

- Assignment.Assignment_Date: Formula from WorkOrder's date



Assignment

Edit Assignment Custom Field

Assignment Date

Custom Field Definition Edit

Field Information

| = Required Information

Field Label: Assignment Date

Field Name: Assignment_Date

Description:

Help Text:

Data Owner: User

Field Usage: --None--

Data Sensitivity Level: --None--

Compliance Categorization: Available: PII, HIPAA, GDPR, PCI; Chosen:

Formula Options

Formula Return Type: Date

Enter your formula and click Check Syntax to check for errors. Click the Advanced Formula subtab to use additional fields, operators, and functions.
**Example:** Reminder Date = CloseDate - 7

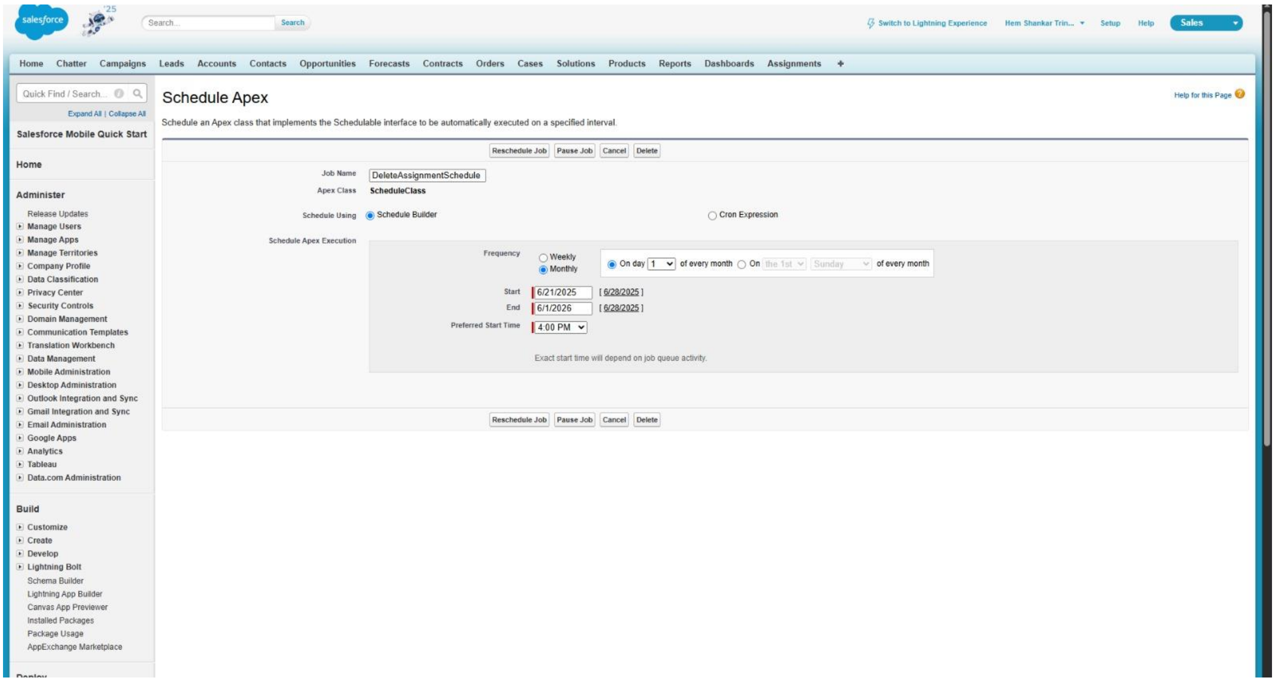
Functions: -- All Function Categories --
ABS
ACOS
ADDMONTHS
AND
ASCII
ASIN

Assignment Date (Date) =

```
WorkOrder__r.LastActivityDate
```

## Scheduler:

**Scheduled Job**: Runs monthly using `ScheduleClass`



Schedule Apex

Schedule an Apex class that implements the Schedulable interface to be automatically executed on a specified interval.

Job Name: DeleteAssignmentSchedule

Apex Class: ScheduleClass

Schedule Using: ● Schedule Builder ○ Cron Expression

Schedule Apex Execution

Frequency: ○ Weekly ● Monthly

● On day 1 of every month ○ On the 1st Sunday of every month

Start: 6/21/2025 [ 6/28/2025 ]

End: 6/1/2026 [ 6/28/2025 ]

Preferred Start Time: 4:00 PM

Exact start time will depend on job queue activity.