FIELD SERVICE WORKORDER OPTIMIZATION

Report

BY-

Bandi Veera Venkata Satyanarayana

venkatbandi9030@@gmail.com

Malla Reddy University

PROJECT ABSTRACT

The Field Service Workorder Optimization project is designed to improve the efficiency and effectiveness of managing field service operations. By implementing automated scheduling systems, work orders are prioritized and assigned based on technician availability, skill sets, and proximity, reducing both downtime and travel time. Real-time data integration and GPS tracking provide technicians with the most current information and directions, ensuring smooth communication with the back office. Technicians can easily access and update job details through a user-friendly interface, while managers gain insights from dashboards displaying key performance metrics. This optimization not only enhances operational efficiency and lowers costs but also boosts customer satisfaction by ensuring timely, high-quality service. By leveraging technology and data analytics, the project significantly streamlines field service management, leading to better resource utilization and improved service outcomes.

INTRODUCTION

Efficient field service management is essential for organizations striving to deliver prompt, high-quality customer service. The Field Service Workorder Optimization project tackles this challenge by utilizing advanced scheduling algorithms, real-time data integration, and smart resource allocation. In industries where minimizing downtime, reducing travel time, and boosting customer satisfaction are critical, this project aims to optimize the management and execution of work orders. By integrating technology and data-driven strategies, it seeks to modernize traditional field service operations, ensuring efficient resource use and improved service delivery.

CONTENTS

1. Salesforce -

- Creating Developer Account : Created a developer organization in salesforce platform by filling all the relevant details.

- Account Activation : Then I have got the mail, verified my account, and set up a password.

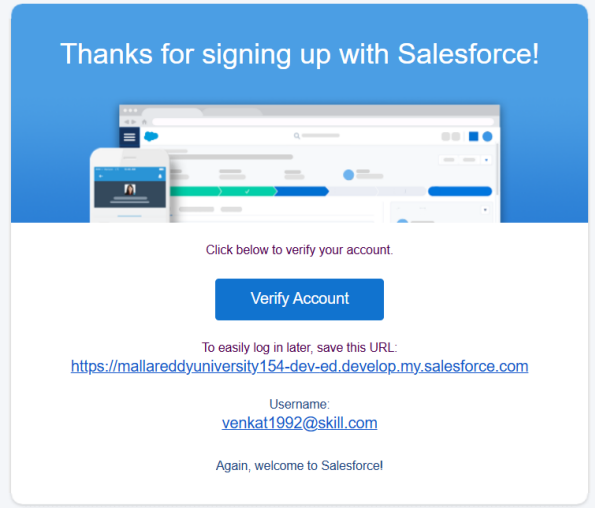


FIG 1.1: Verification Mail

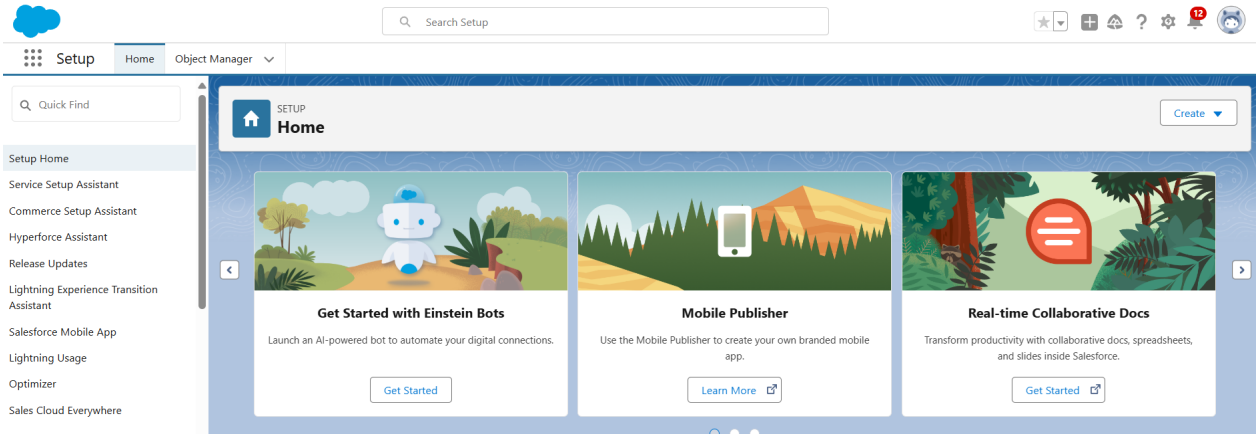


FIG 1.1: Home Page

2. Object

- Create Technician Object : After downloading the given Technician spreadsheet, create custom object from spreadsheet, upload the technician.csv file and import the data as shown.

- Create WorkOrder Object : Same steps to be followed for work order object, here we no need to import the data.

- Create Assignment Object : Creating a custom object "Assignment" and enter details, save.

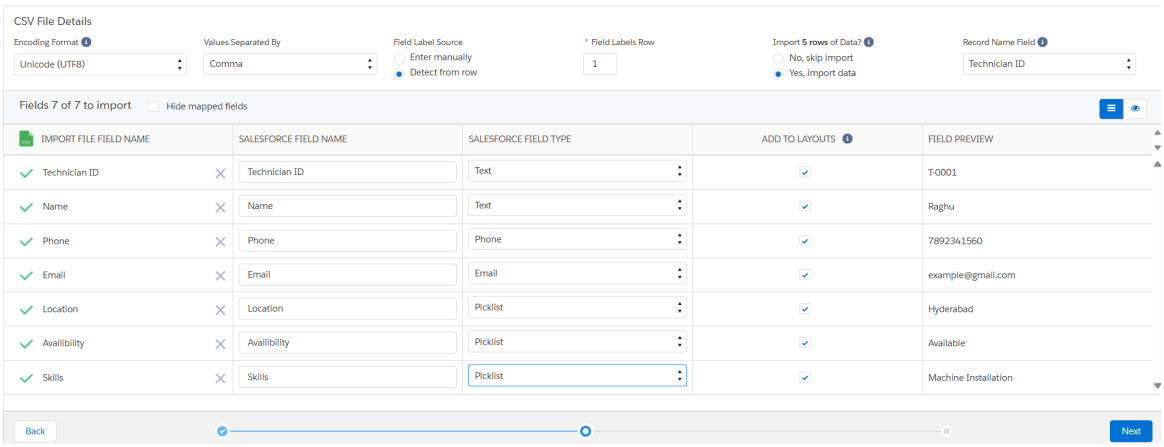


FIG 2.1: Create Technician Object

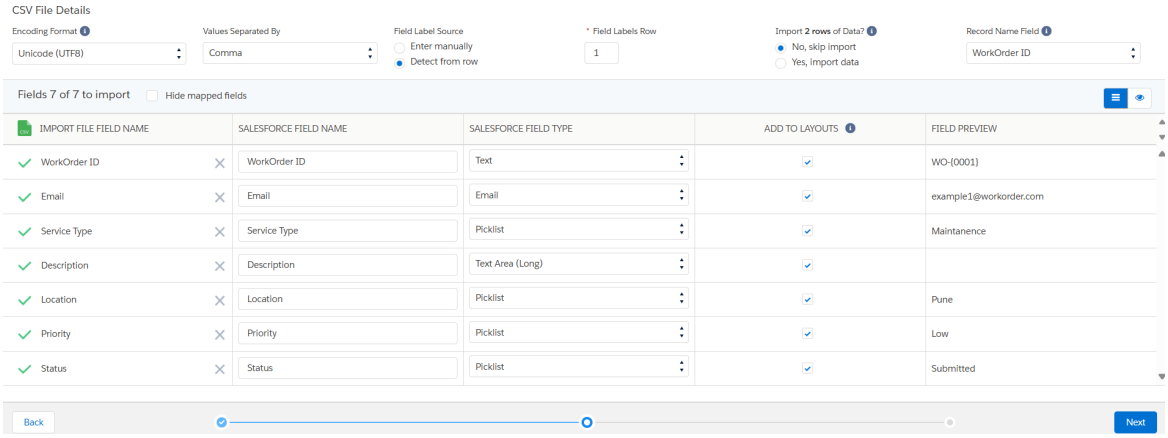


FIG 2.2 : Create WorkOrder Object

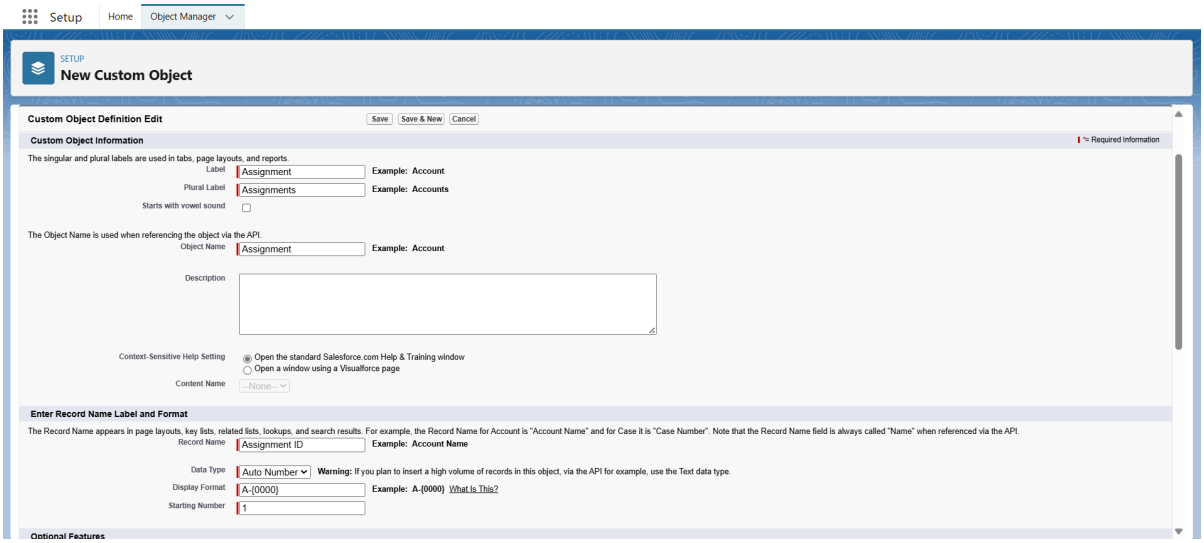


FIG 2.3 : Create Assignment Object

3. Tabs

- Creating a Custom Tab : Select the Tabs option in the Quick find box and create a new tab named Assignment.

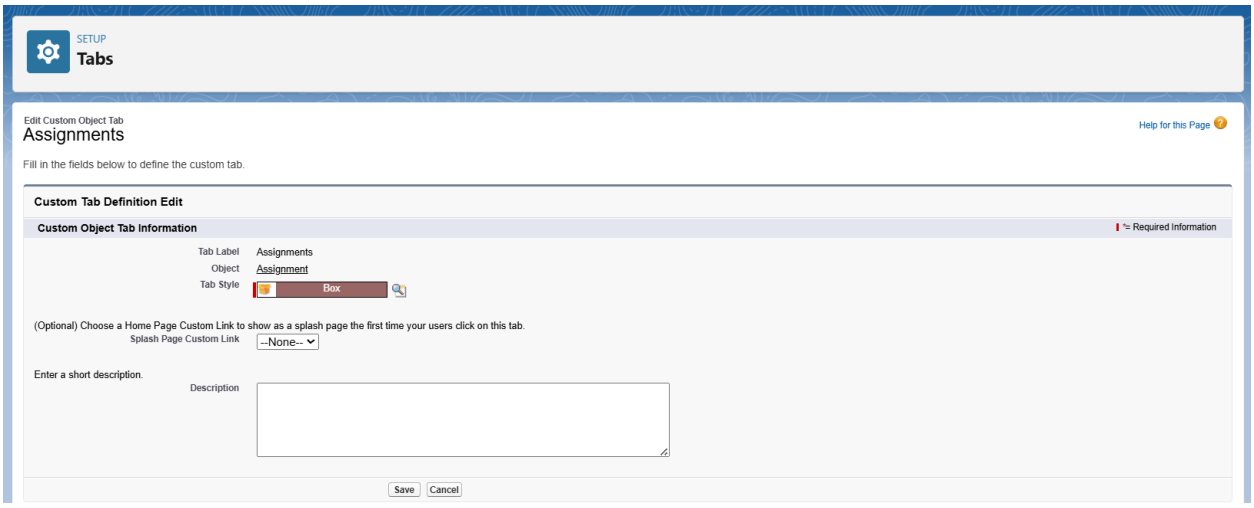


FIG 3.1 : Custom Tab

- By default, Technician and WorkOrder tabs will be created once the custom object is created.

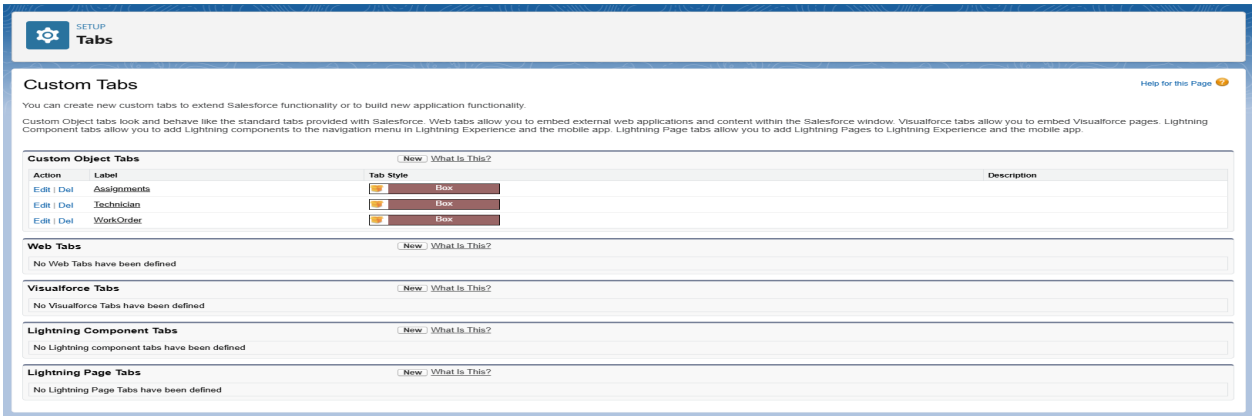


FIG 3.2: Custom tabs List

4. The Lightning App

- Create a Lightning App : Again, in quick find box, go to app manager and create new lightning app with the name of project and some further details.

- Add Navigation Items (i.e, Home, WorkOrder, Assignments, Technician, Reports & Dashboards)

- Add User Profile (i.e, System Administrator)

- Then click **save** and finish

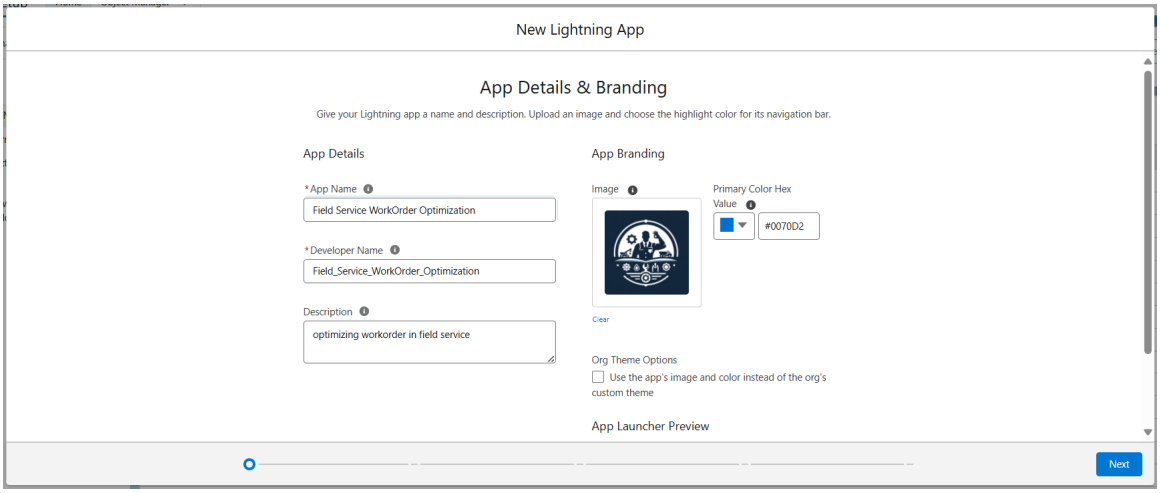


FIG 4.1: Create Lightning App

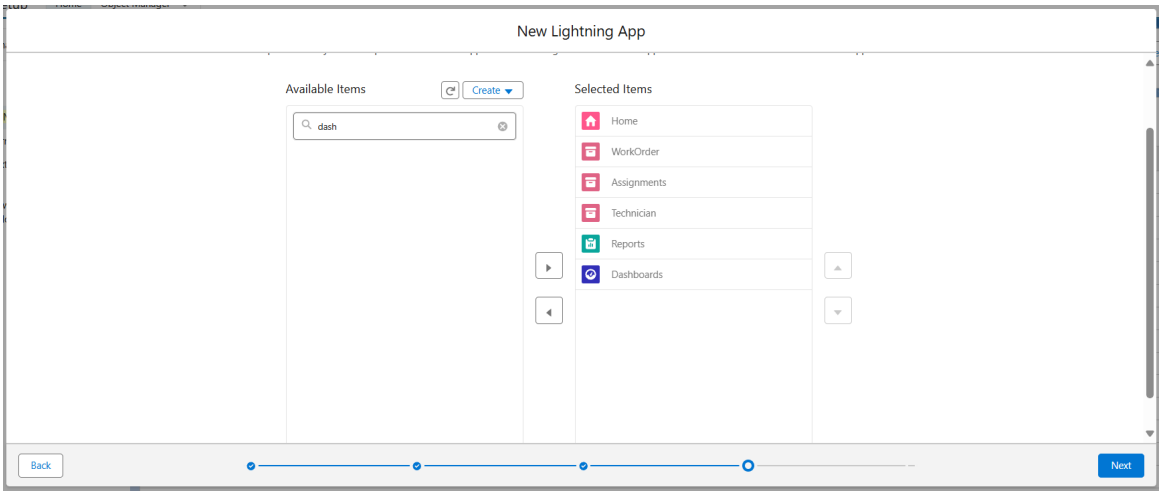


FIG 4.2 : Add Navigation Items

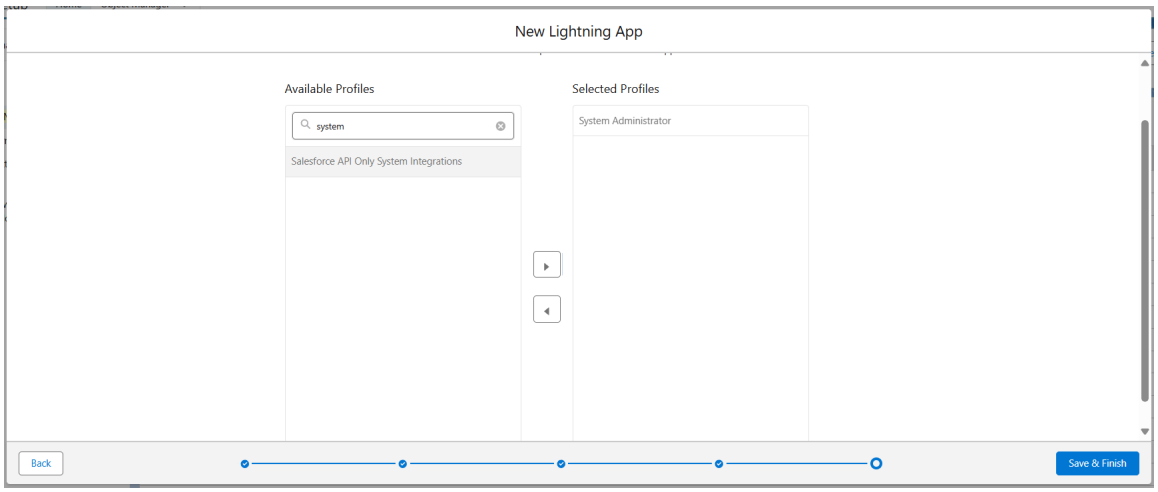


FIG 4.3 Add User Profile

5. Fields & Relationships

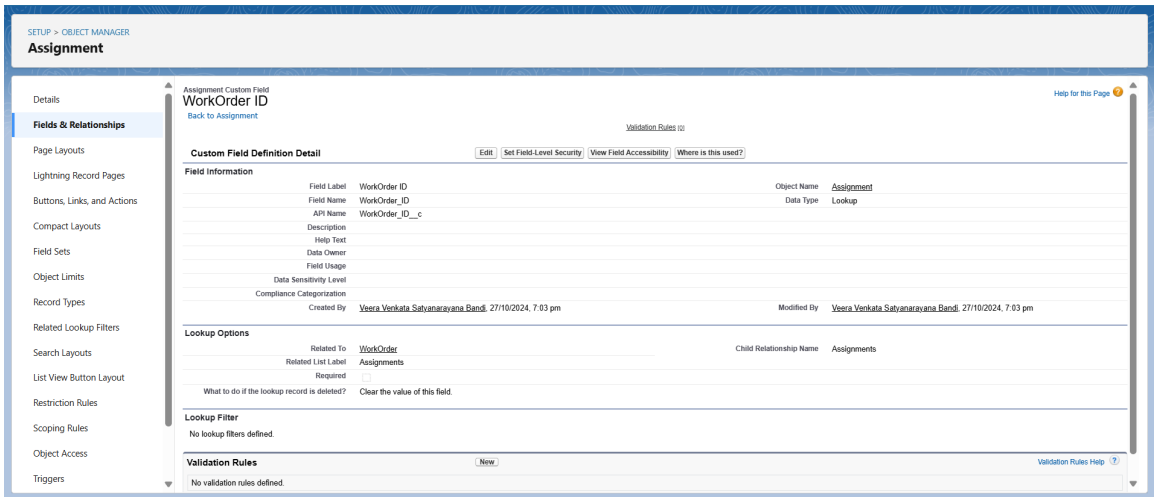


FIG 5.1: Lookup WorkOrder

- Creating Lookup Field In Assignment Object : In Assignment go to Fields & Relationships and create a new field labeled as WorkOrder and datatype as Lookup.

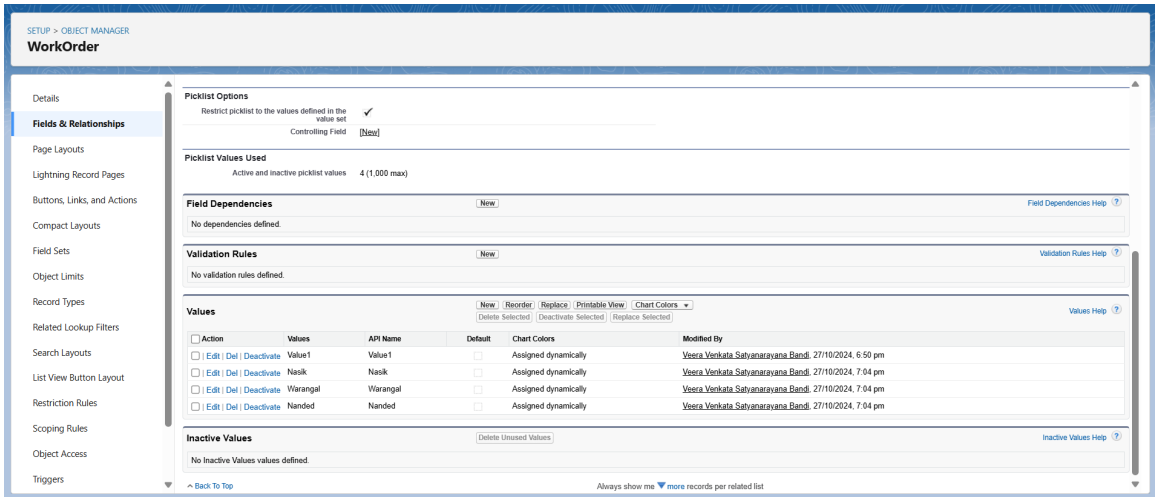


FIG 5.2: Update Location Picklist

- Manage your Picklist Values : In Object Manager select WorkOrder object go to fields & relationships. In that location field add new values (i.e, Nasik, Warangal, Nanded).

- Add more values in the fields of priority(i.e, High) and Service type(i.e, Hardware repair, Troubleshoot/Debugging, Lane-Management).

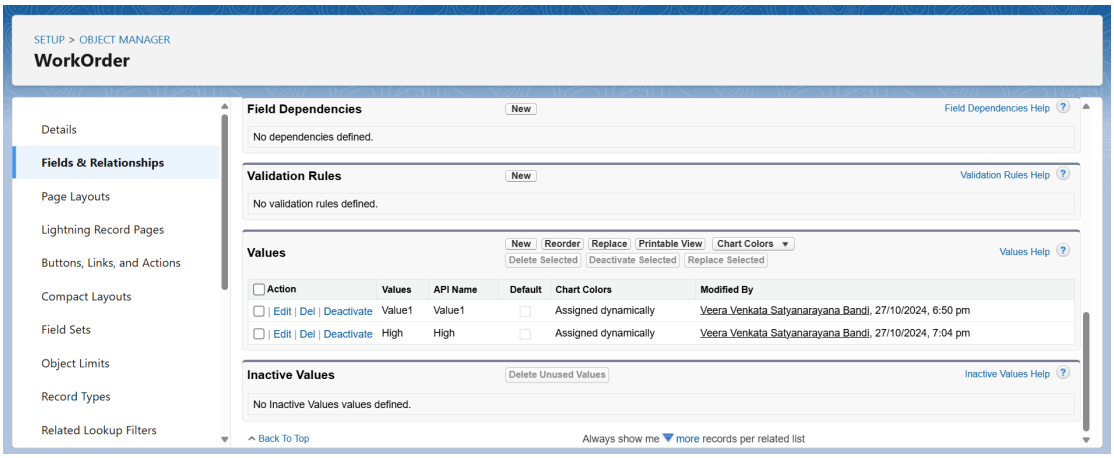


FIG 5.3 Priority Field

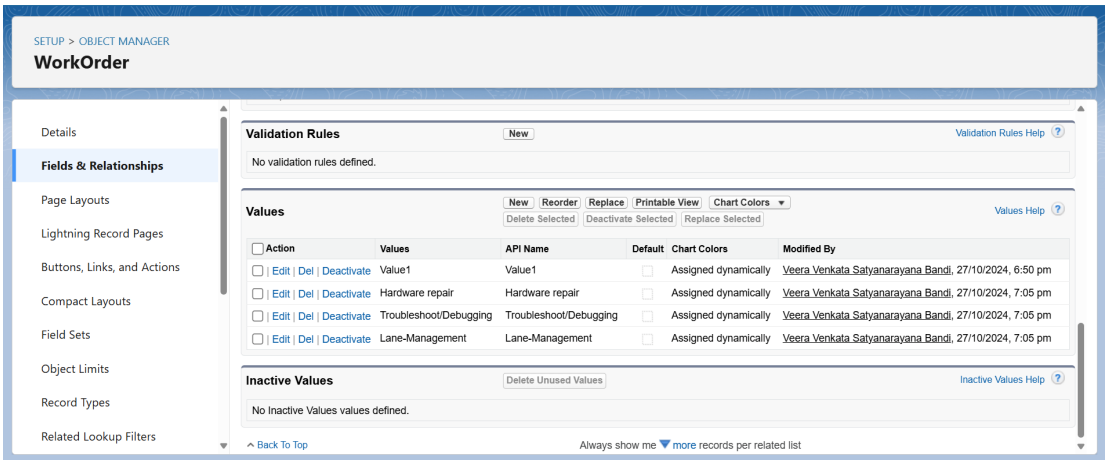


FIG 5.4 Service Type Field

- Creating Formula Field in WorkOrder Object: Now create a Formula Datatype and give the field label as "date". The formula is "CreateDate".

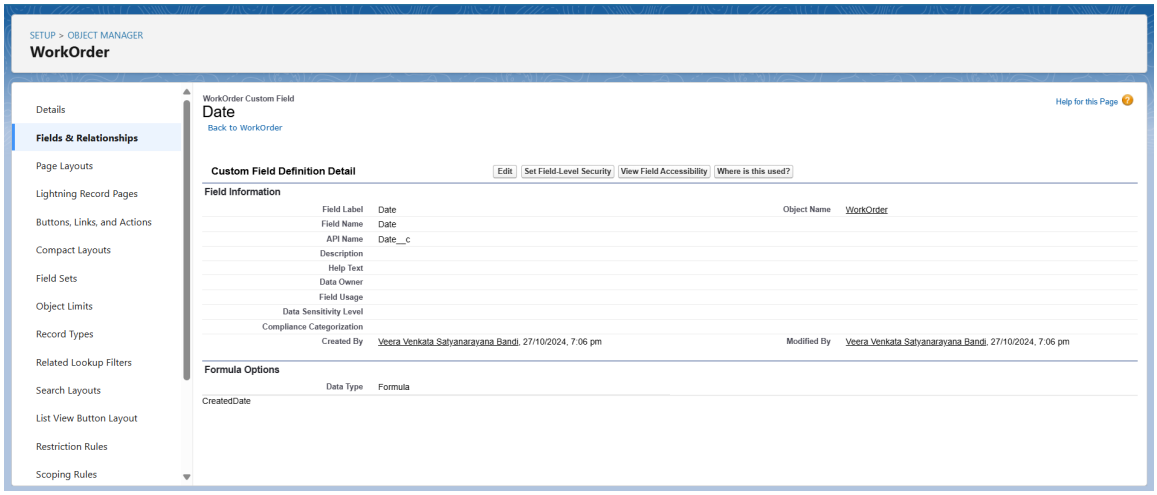


FIG 5.5 Date Formula Field

- Now, in the Assignment object, create a Formula Datatype in Fields & relationships. Add Technician ID with return type Date.

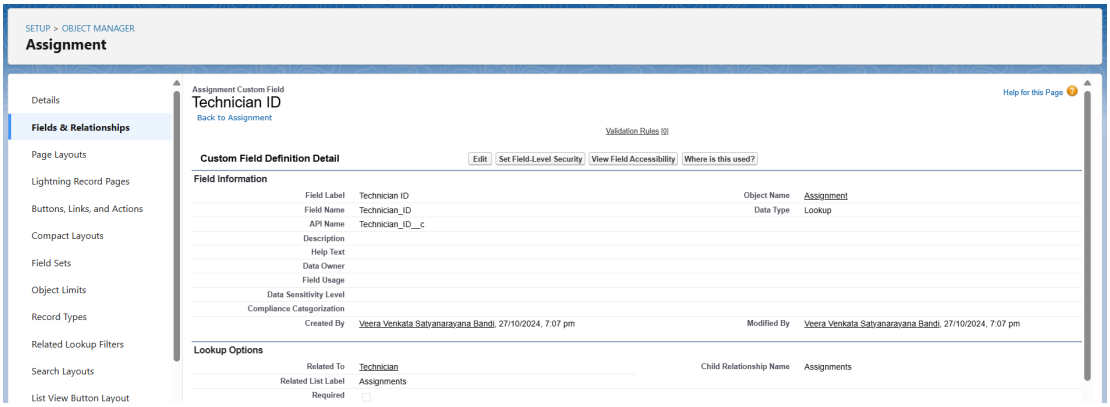


FIG 5.6 Lookup Technician ID

- Add Assignment Date with Formula with return type date (WorkOrder_ID__r.Date__c).

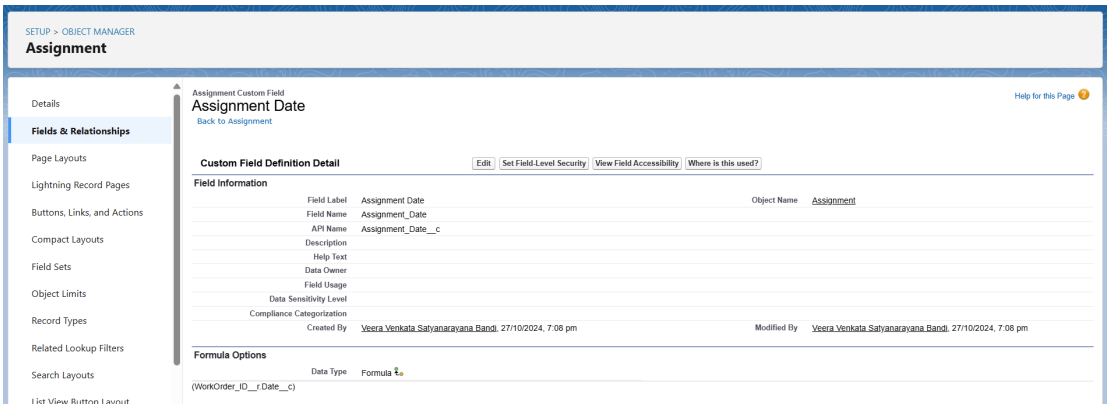


FIG 5.7 Formula Assignment Date

- Add Completion Date with Formula with return type date " IF(ISPICKVAL(WorkOrder_ID__r.Status__c , 'Resolved'), WorkOrder_ID__r.LastModifiedDate , NULL)"

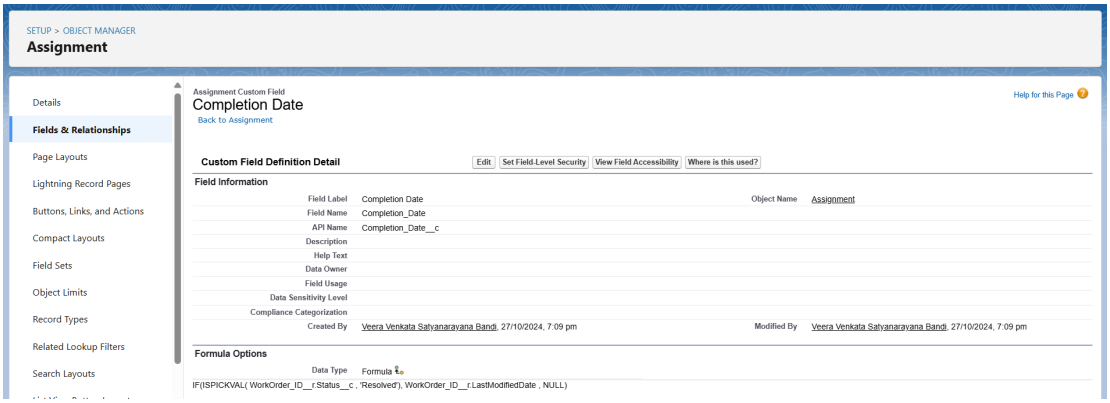


FIG 5.8 Formula Completion Date

6. Profiles -

Technician Profile : To create a new profile in Salesforce, navigate to Setup, type "Profiles" in the Quick Find box, and select "Profiles." Click "New Profile," choose "Standard Platform User" as the existing profile, name it "Technician," and click "Save." On the profile page, click "Edit." Scroll to Custom Object Permissions and grant Readonly access for Technician, WorkOrder, and Assignment objects, then click "Save." On the profile detail page, scroll to Custom Field-Level Security, click "View" next to the WorkOrder object, then "Edit." Enable the checkbox for the Status field and click "Save."

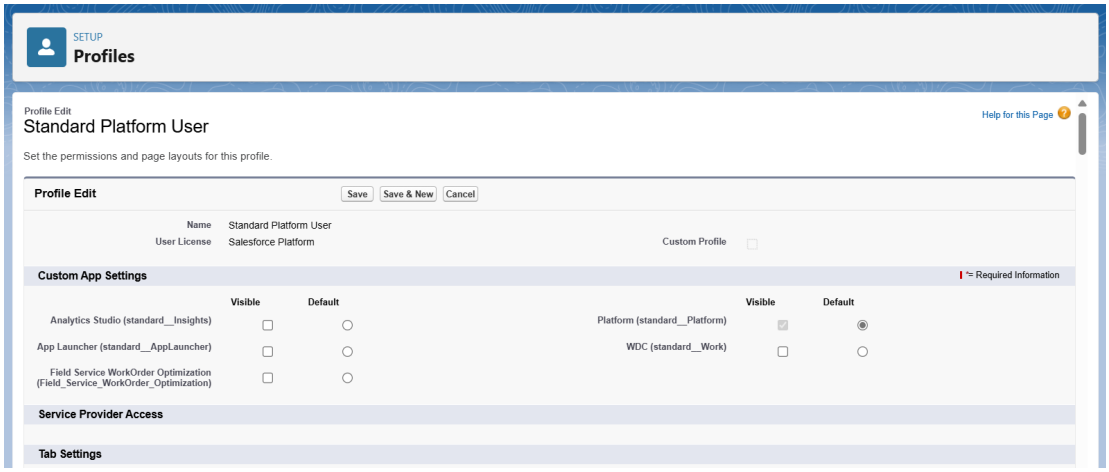


FIG 6.1 Technician Profile

7. Users

- Create User : Go to User and create a new one with new name and other details such as last-name, alias, email, username, nickname, user license, and profiles.

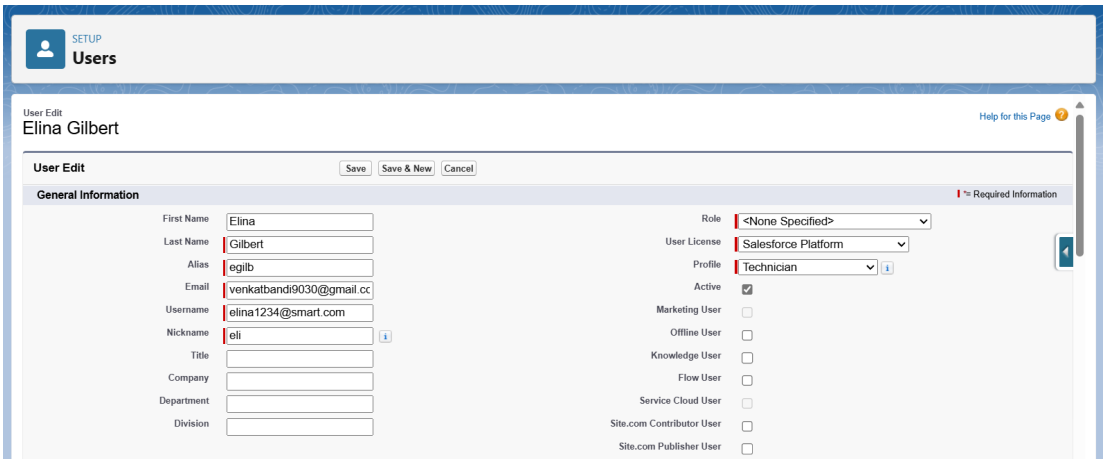
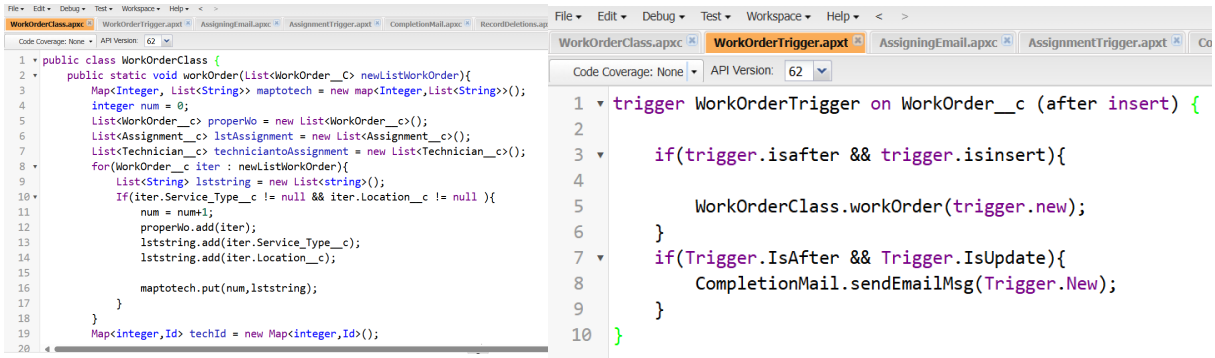


FIG 7.1 User

8. Apex Trigger

- Create Apex Class : To create a new Apex Class in Salesforce, go to Setup, click the gear icon, and select "Developer Console." The Developer Console will open in a new window. Click on "File," then "New," and select "Apex Class." Name the class "WorkOrderClass" and click "OK."

- Create Apex Trigger : To create a new Apex Trigger in Salesforce, open the Developer Console, click on "File," then "New," and select "Apex Trigger." Name the trigger "WorkOrderTrigger" and select "WorkOrder__c" from the sObject dropdown. Click "Submit" to create the trigger.



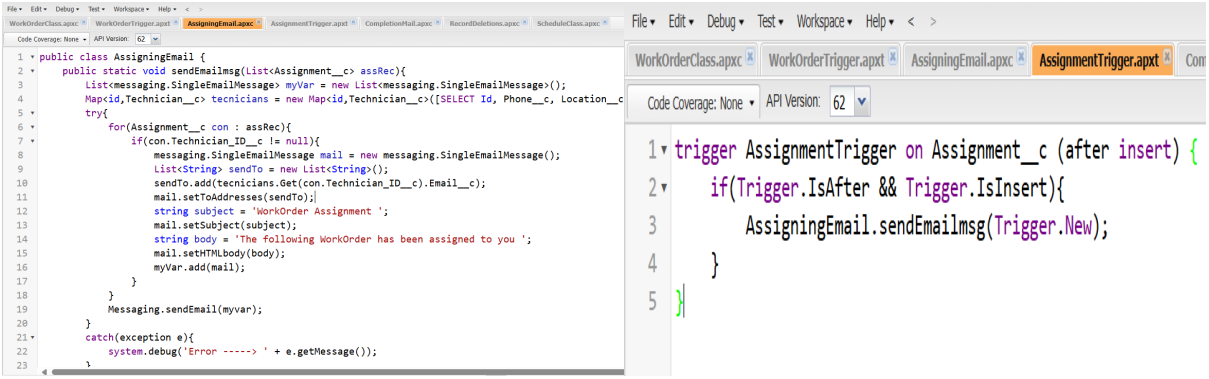
```
public class WorkOrderClass {
    public static void workOrder(List<WorkOrder__C> newListWorkOrder){
        Map<Integer, List<String>> maptotech = new map<Integer,List<String>>();
        integer num = 0;
        List<WorkOrder__c> properWo = new List<WorkOrder__c>();
        List<Assignment__c> 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>();
```

```
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);
    }
}
```

FIG 8.1 WorkOrder Apex Class and Trigger Code

- Create Apex Class : To create a new Apex Class in Salesforce, go to Setup, click the gear icon, and select "Developer Console." The Developer Console will open in a new window. Click on "File," then "New," and select "Apex Class." Name the class "AssigningEmail" and click "OK."

- Create Apex Trigger : To create a new Apex Trigger in Salesforce, open the Developer Console, click on "File," then "New," and select "Apex Trigger." Name the trigger "AssignmentTrigger" and select "WorkOrder__c" from the sObject dropdown. Click "Submit" to create the trigger.



```
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
        try{
            for(Assignment__c con : assRec){
                if(con.Technician_ID__c != null){
                    messaging.SingleEmailMessage mail = new messaging.SingleEmailMessage();
                    List<String> sendTo = new List<String>();
                    sendTo.add(tecnicians.Get(con.Technician_ID__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());
        }
```

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

FIG 8.2 Assignment Apex Class and Trigger Code

- Create Apex Class : To create a new Apex Class in Salesforce, go to Setup, click the gear icon, and select "Developer Console." The Developer Console will open in a new window. Click on "File," then "New," and select "Apex Class." Name the class "CompletionMail" and click "OK." FIG

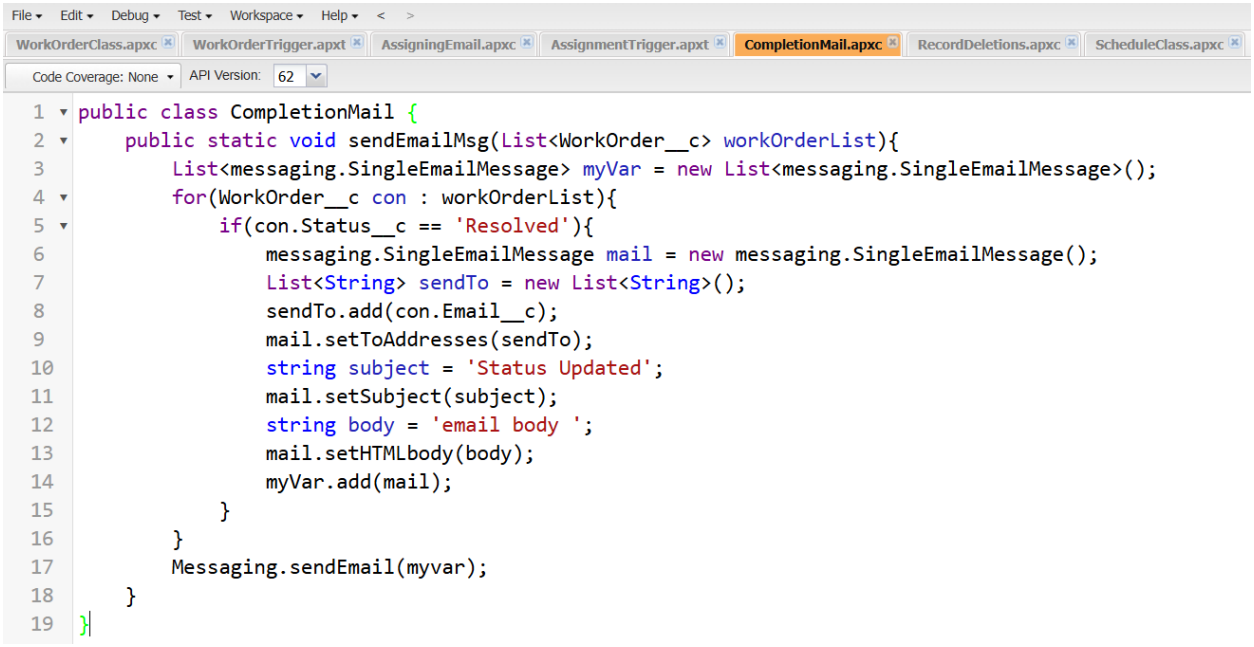


FIG 8.3 Completion Apex Class Code

- Create an Asynchronous Apex Class : To create a new Apex Class in Salesforce, go to Setup, click the gear icon, and select "Developer Console." The Developer Console will open in a new window. Click on "File," then "New," and select "Apex Class." Name the class "Record Deletion" and click "OK."

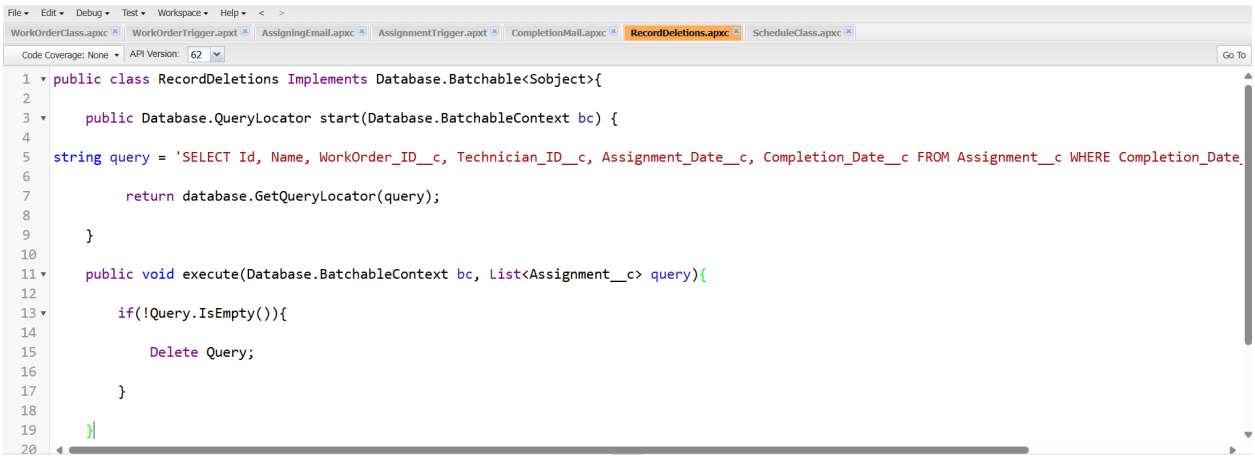


FIG 8.4 RecordDeletion Apex Class Code

- Create an Apex Schedule Class : To create a new Apex Class in Salesforce, go to Setup, click the gear icon, and select "Developer Console." The Developer Console will open in a new window. Click on "File," then "New," and select "Apex Class." Name the class "Schedule" and click "OK."

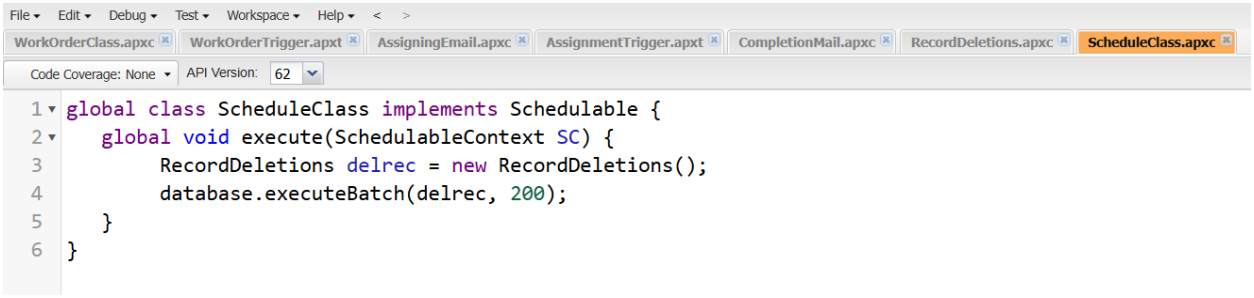


FIG 8.5 Schedule Apex Class Code

- Create A Schedule Apex : To schedule an Apex class in Salesforce, go to Setup and search for “Apex Classes” in the Quick Find box. Click on “Schedule Apex.” Enter the Job Name as "DeleteAssignmentSchedule," select "ScheduleClass" using the lookup icon for the Apex Class, set the Frequency to "Monthly," and choose a Preferred Start Time. Save the schedule to automate the Apex class execution.

9. Reports & Dashboards

- Create Reports : To create a new report in Salesforce, go to the app and click on the Reports tab. Click "New Report." Select the report type from the category, report type panel, or search panel, then click "Start Report." Customize your report by adding fields from the left pane. Group the report by Work Order ID for better organization. Save and run the report to view the results.

- Created a report type on “WorkOrders Status Reports”, “Technician and Assignment Details Reports”, "New Assignment with WorkOrder ID Report".

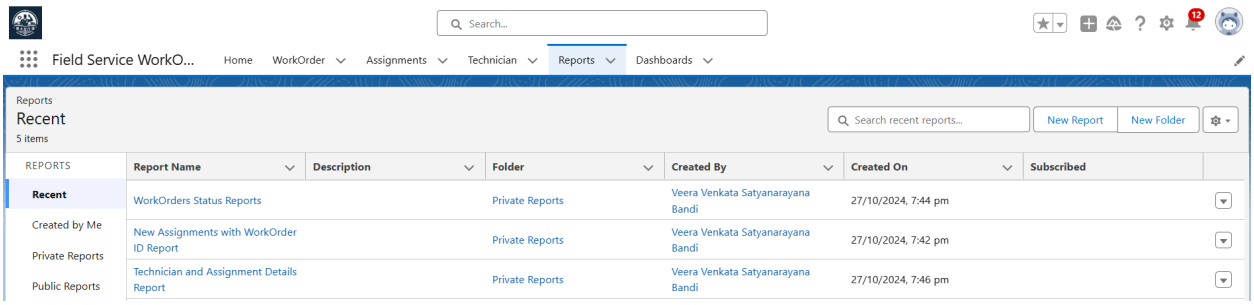


FIG 9.1 Report List

- Create Dashboards : To create a dashboard in Salesforce, go to the app and click on the Dashboards tab. Click "New Dashboard," give it a name, and click "Create." Select "Add Component," choose the report you created previously, and click "Select." Click "Add," then "Save," and finally, click "Done" to complete the dashboard setup.

-Created all three Dashboards on the above Reports.

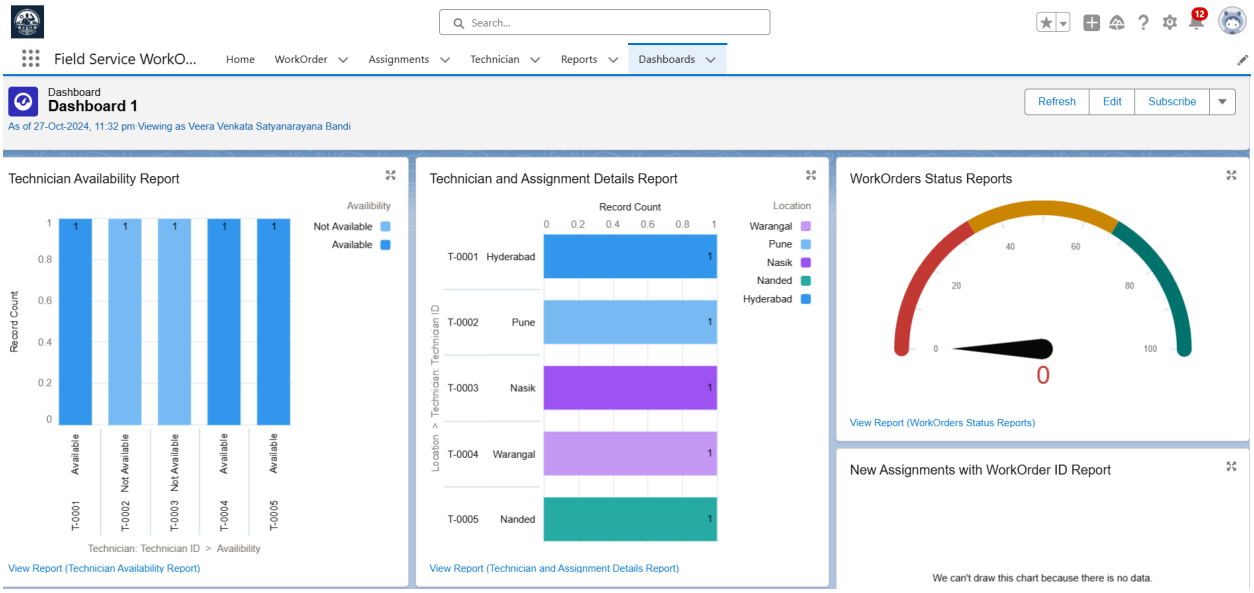


FIG 9.2 Technician and Assignment Details Report Dashboard

Thank You