FIELD SERVICE WORKORDER OPTIMIZATION

Report

Prepared by

Swetha Gudivada

swethagudivada2004@gmail.com

MallaReddy University

PROJECT ABSTRACT

The Field Service Workorder Optimization project aims to enhance the efficiency and effectiveness of managing field service operations. By implementing automated scheduling systems, the project ensures work orders are prioritized and assigned based on technician availability, skills, and proximity, thereby reducing downtime and travel time. Real-time data integration and GPS tracking provide technicians with up-to-date information and directions, facilitating seamless communication with the back office. An intuitive user interface allows technicians to easily access and update job details, while managers benefit from dashboards displaying key performance metrics. This optimization not only increases operational efficiency and reduces costs but also improves customer satisfaction by ensuring timely and high-quality service delivery. Overall, the project leverages technology and data analytics to streamline field service management, resulting in significant improvements in resource utilization and service outcomes.

INTRODUCTION

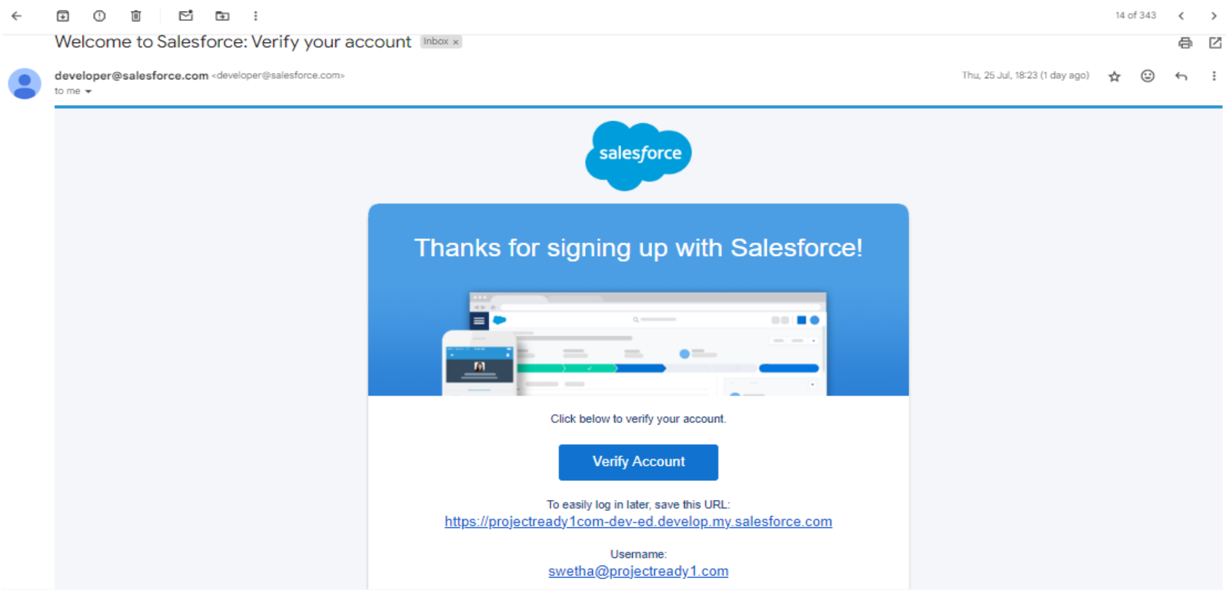
Efficient management of field service operations is crucial for organizations aiming to provide timely and high-quality service to their customers. The Field Service Workorder Optimization project addresses this need by leveraging advanced scheduling algorithms, real-time data integration, and intelligent resource allocation. In an industry where minimizing downtime, reducing travel time, and enhancing customer satisfaction are paramount, this project seeks to streamline the process of managing and executing work orders. By incorporating technology and data-driven strategies, the project aims to transform traditional field service management, ensuring optimal use of resources and improved service outcomes.

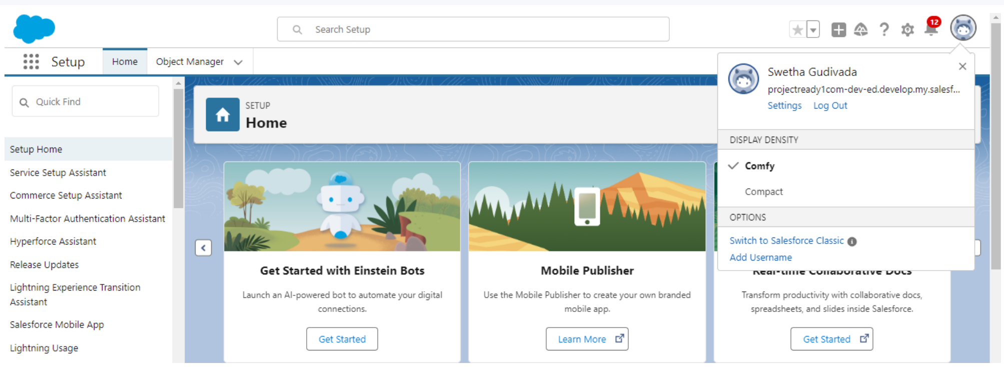
CONTENTS

S.NO	TOPIC	Pg.No
1	SalesForce - Creating Developer Account - Account Activation	4
2	Object - Create Technician Object - Create WorkOrder Object - Create Assignment Object	4-5
3	Tabs - Create a Custom Tab	6
4	The Lightning App - Create Lightning App	6-7
5	Fields & Relationships - Creating a lookup field in Assignment Object - Manage your Picklist Values - Creating Formula field in WorkOrder Object	8-9
6	Profiles - Technician Profile	10
7	Users - Create User	10
8	Apex Trigger - Create Apex Classes - Create Triggers	11-13
9	Reports & Dashboards - Create Reports - Create Dashboards	13-14

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.





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 as shown in the steps.
- Create Assignment Object : Creating a custom object with the label name as Assignment and enter the details, then save.

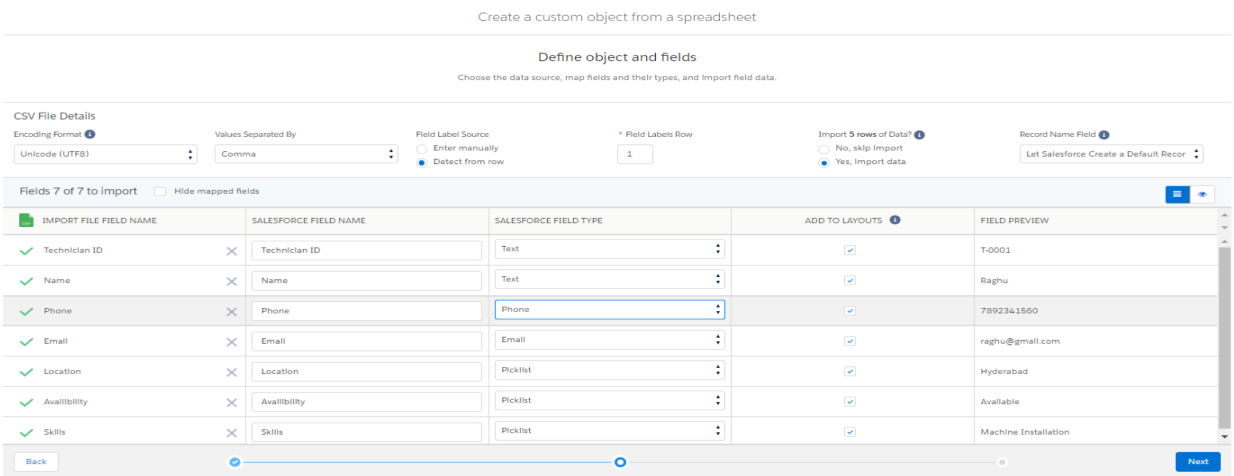


FIG 2.1: Create Technician Object

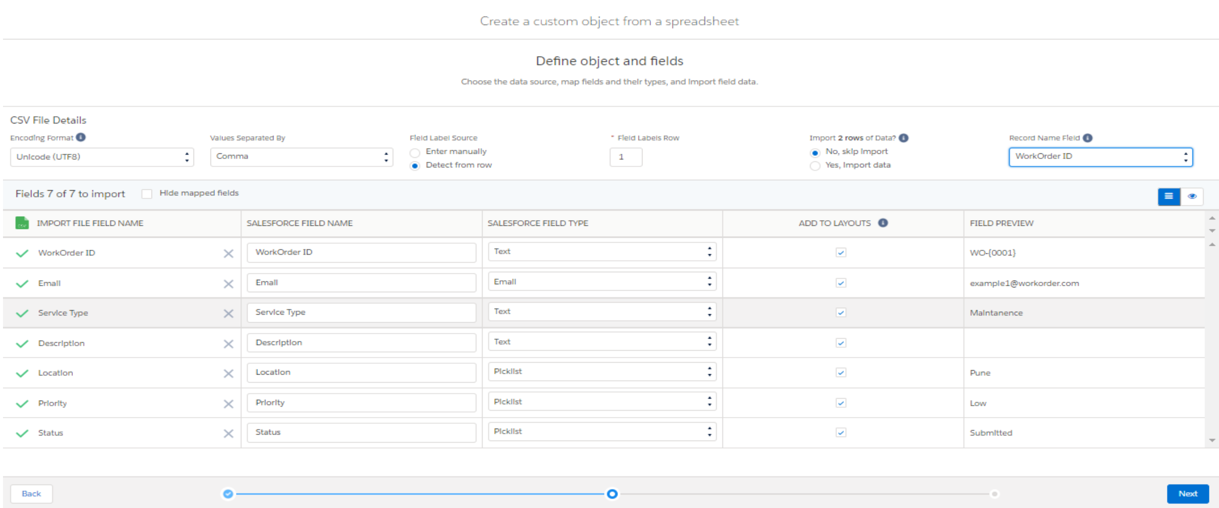


FIG 2.2 : Create WorkOrder Object

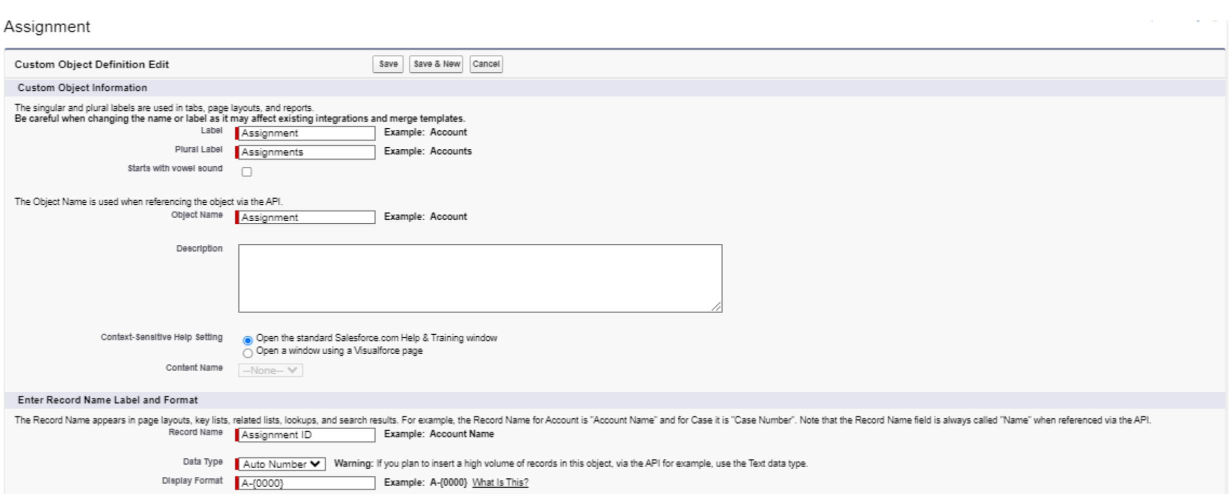
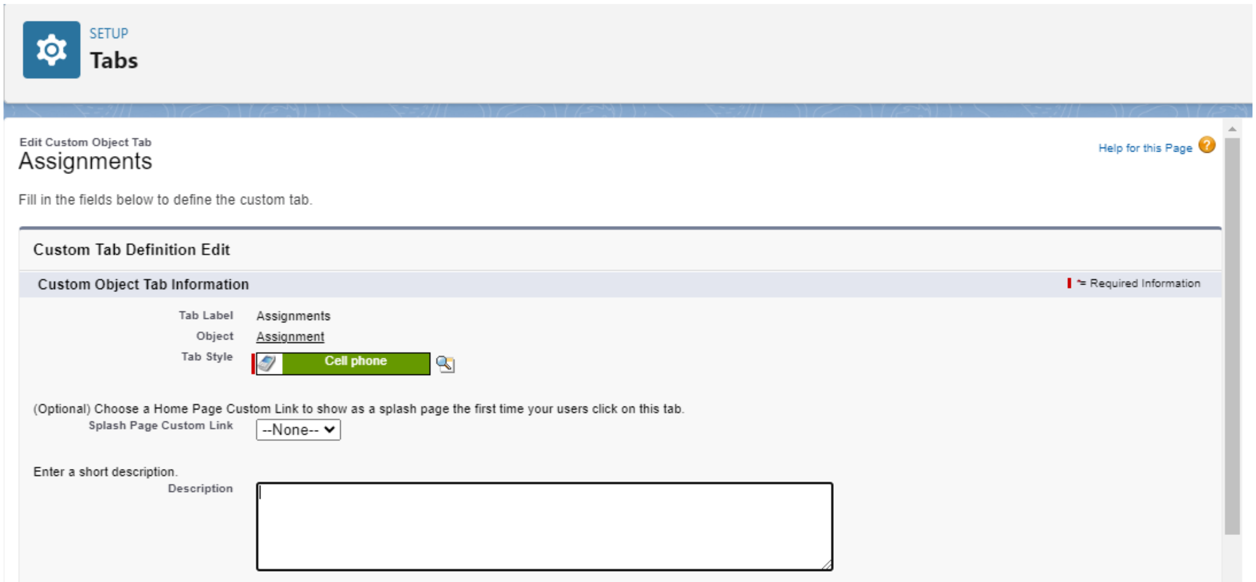


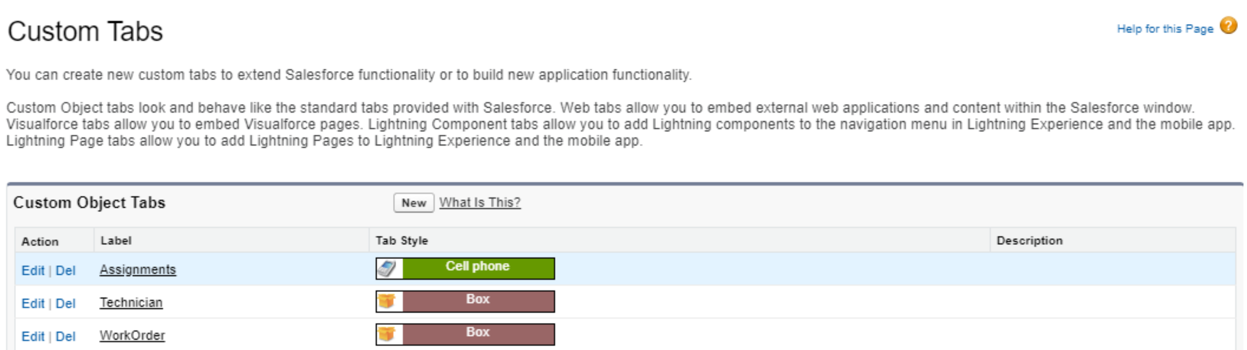
FIG 2.3 : Create Assignment Object

3. Tabs

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



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



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 and Dashboards)
- Add User Profile (i.e, System Administrator)
- Then click save and finish

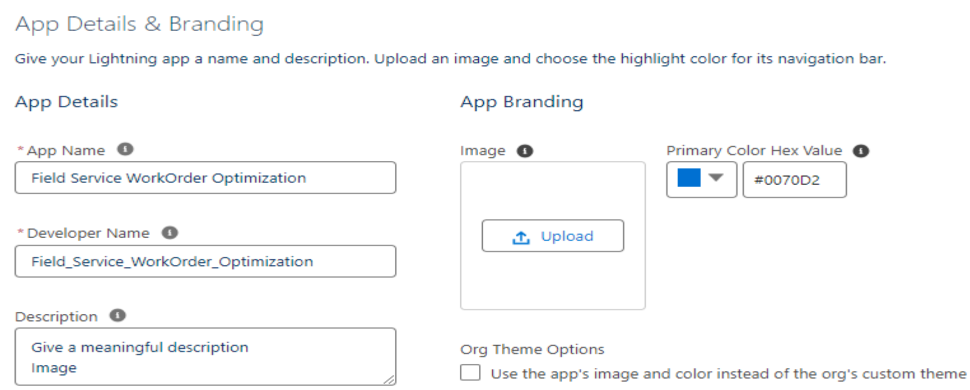


FIG 4.1: Create Lightning App

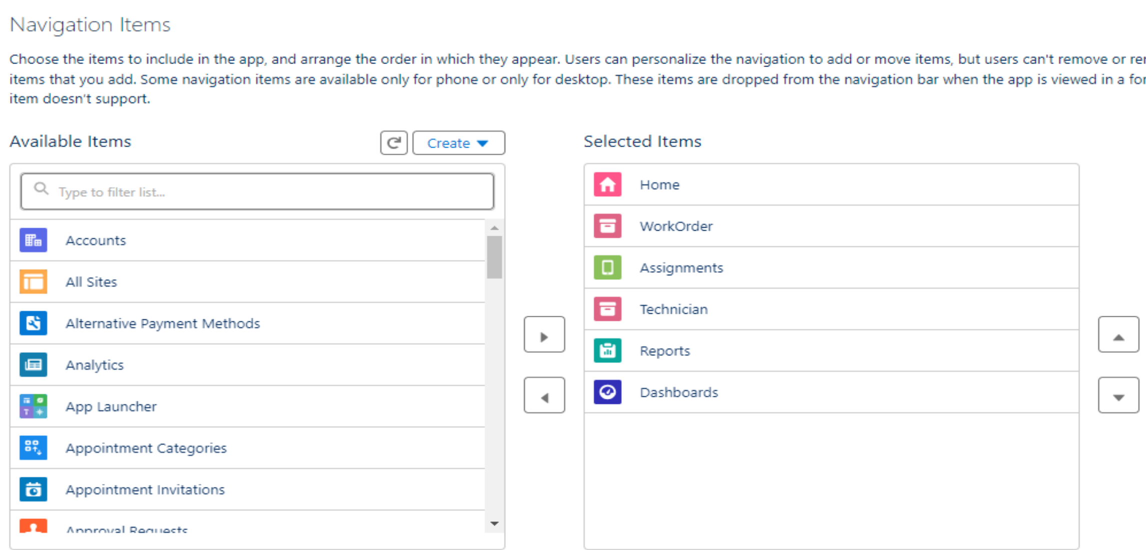


FIG 4.2 : Add Navigation Items

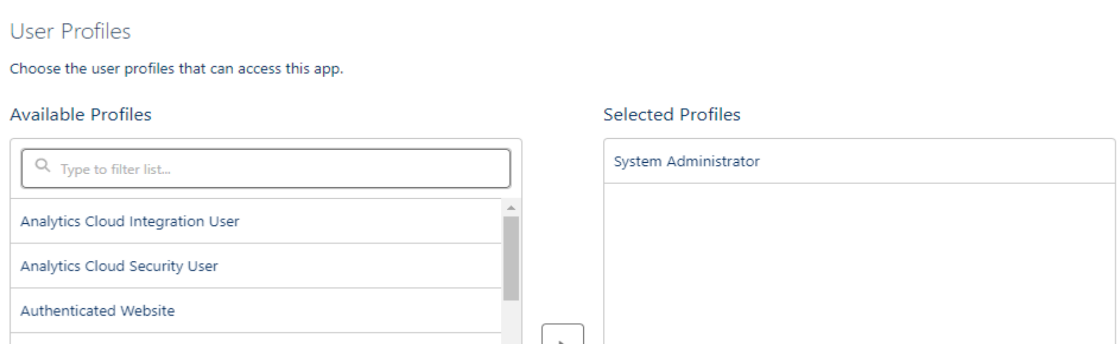
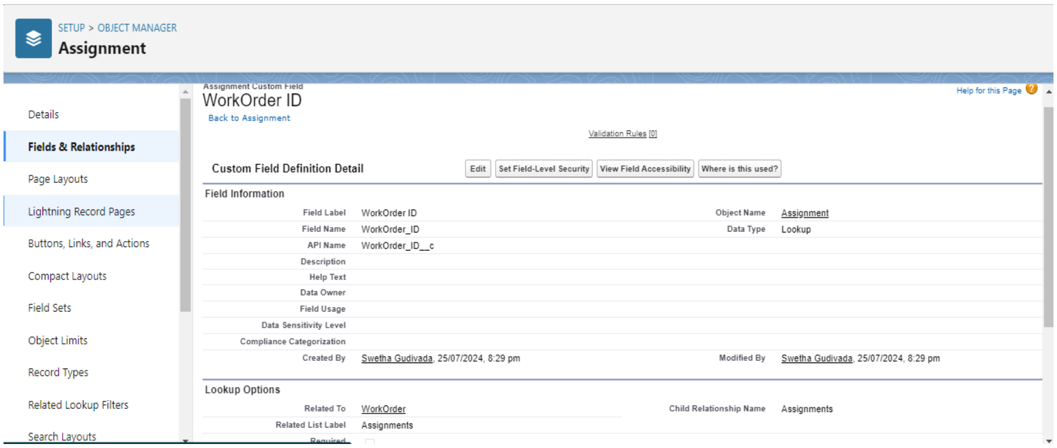


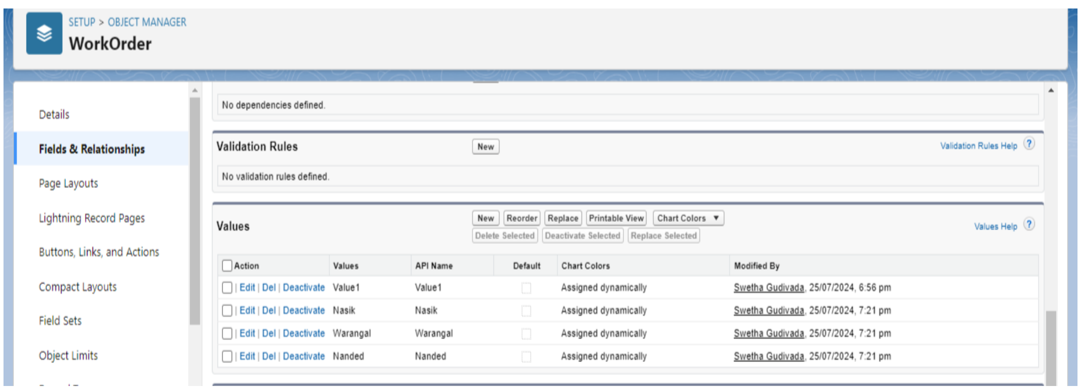
FIG 4.3 Add User Profile

5) <u>Fields & Relationships</u>

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



- 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).

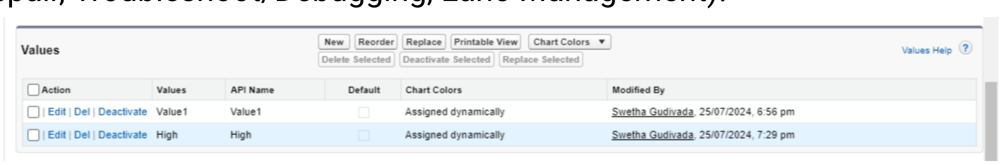
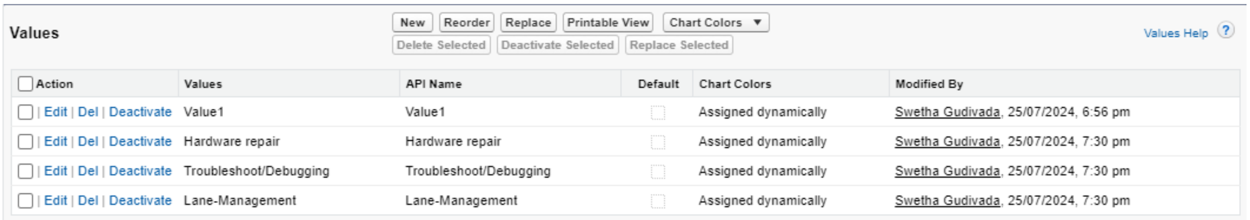


FIG 5.1 Priority Field



Values

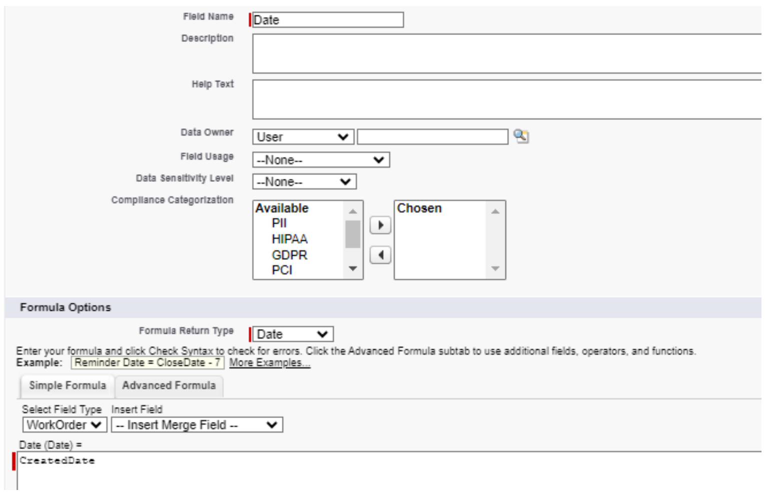
New | Reorder | Replace | Printable View | Chart Colors
Delete Selected | Deactivate Selected | Replace Selected

Values Help

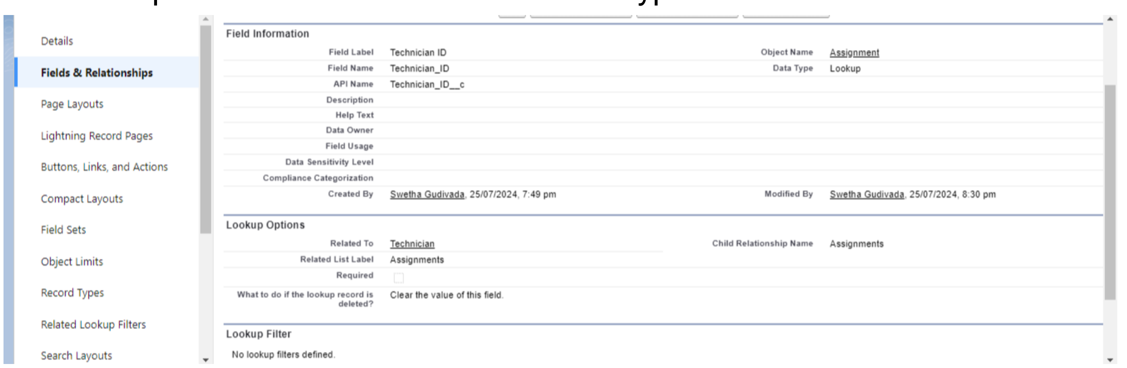
Action	Values	API Name	Default	Chart Colors	Modified By
Edit | Del | Deactivate	Value1	Value1		Assigned dynamically	Swetha Gudivada, 25/07/2024, 6:56 pm
Edit | Del | Deactivate	Hardware repair	Hardware repair		Assigned dynamically	Swetha Gudivada, 25/07/2024, 7:30 pm
Edit | Del | Deactivate	Troubleshoot/Debugging	Troubleshoot/Debugging		Assigned dynamically	Swetha Gudivada, 25/07/2024, 7:30 pm
Edit | Del | Deactivate	Lane-Management	Lane-Management		Assigned dynamically	Swetha Gudivada, 25/07/2024, 7:30 pm

FIG 5.2 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".



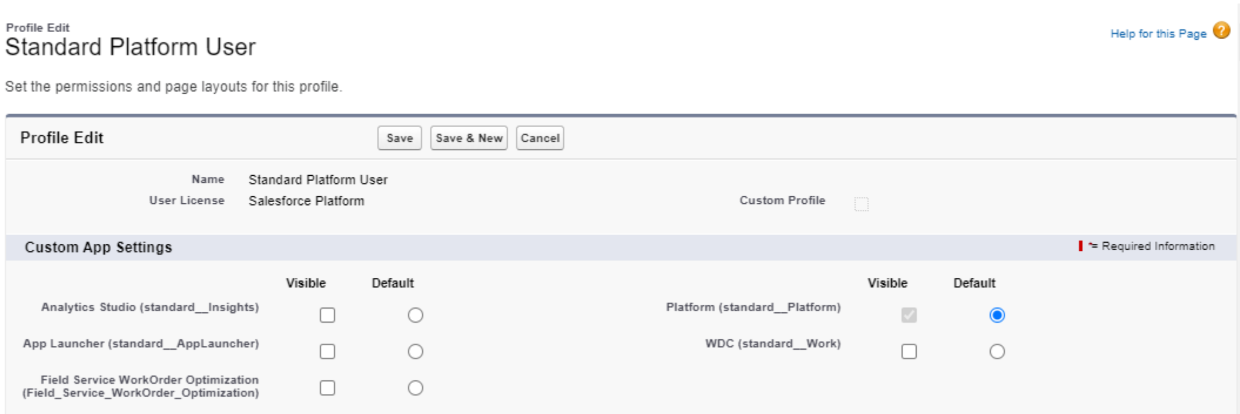
- Now, in the Assignment object, create a Formula Datatype in Fields & relationships. Add Technician ID with return type Date.



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

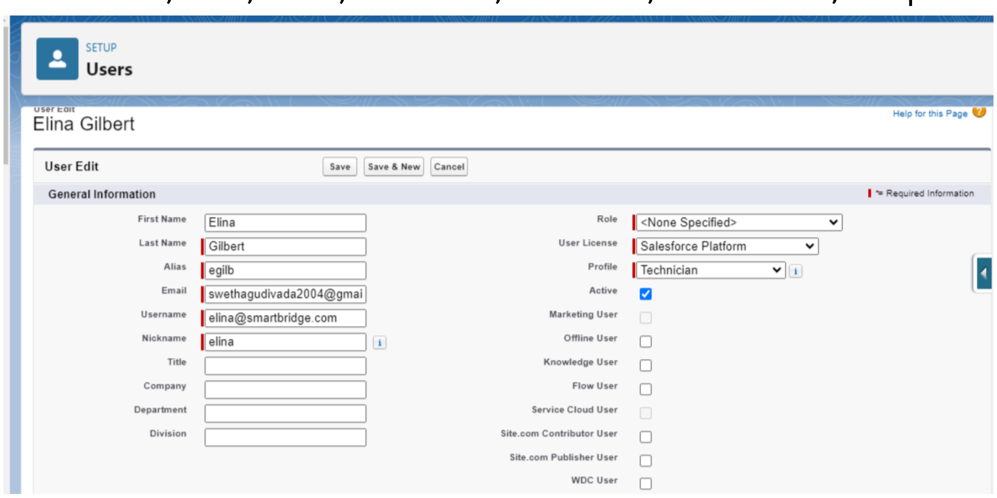
6. Profiles

- 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 Read-only 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."



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.



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.

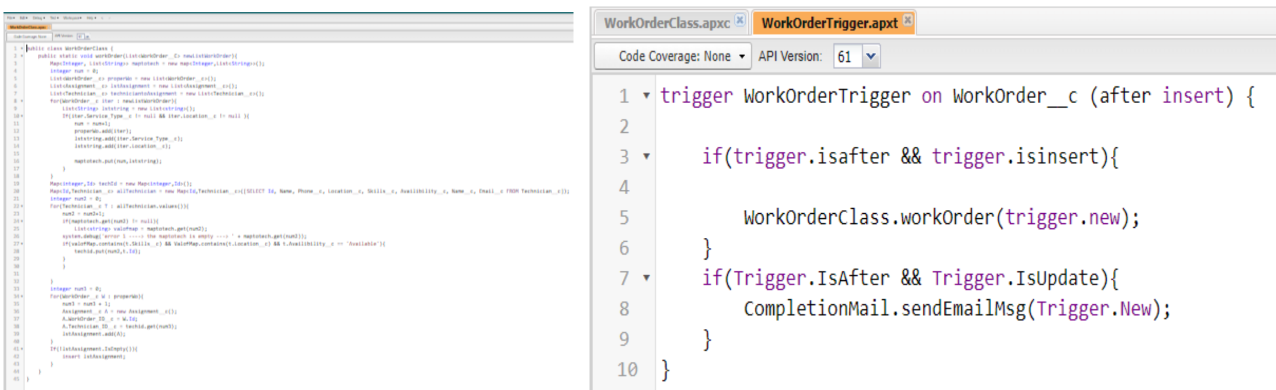
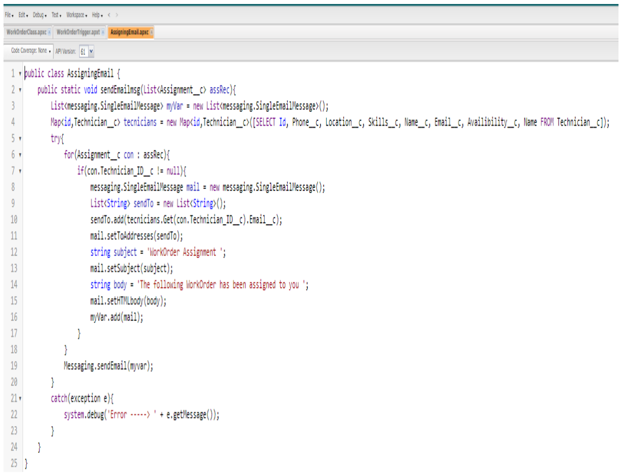


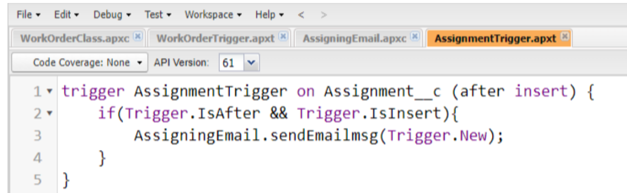
FIG 8.1 WorkOrder Apex Class and Trigger Code

- 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.



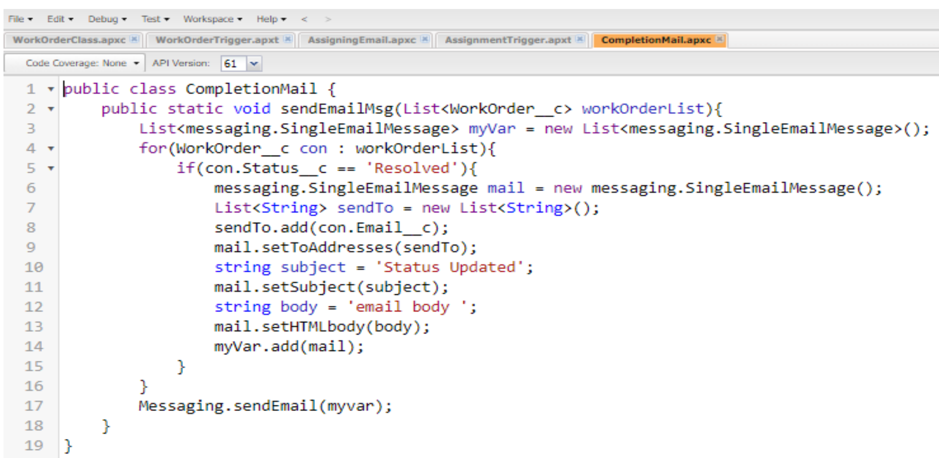
```
public class AssigningEmail {
    public static void sendEmailmsg(List<Assignment__c> assRec){
        List<messaging.SingleEmailMessage> myVar = new List<messaging.SingleEmailMessage>();
        Map<id,Technician__c> tecnicians = new Map<id,Technician__c>([SELECT Id, Phone__c, Location__c, Skills__c, Name__c, Email__c, Availibility__c, Name FROM Technician__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."



```
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);
    }
}
```

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."

```
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){
    }
}
```

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."

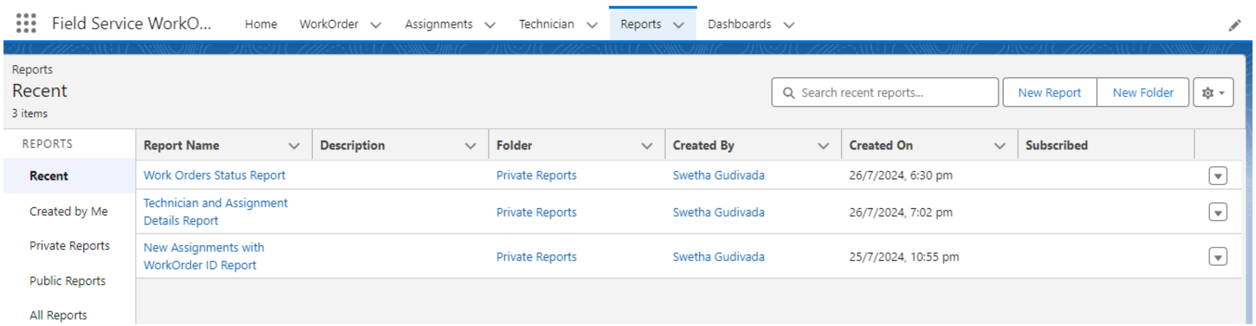
```
global class ScheduleClass implements Schedulable {
   global void execute(SchedulableContext SC) {
        RecordDeletions delrec = new RecordDeletions();
        database.executeBatch(delrec, 200);
   }
}
```

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. <u>Reports & Dashboards</u>

- 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".

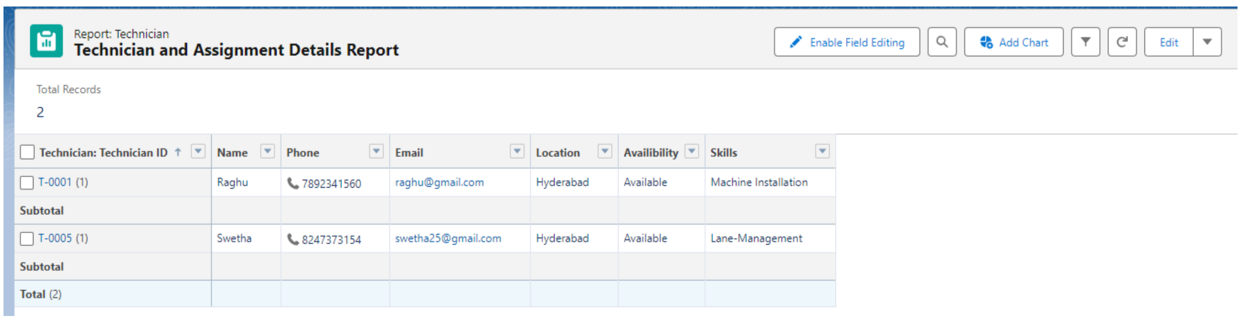
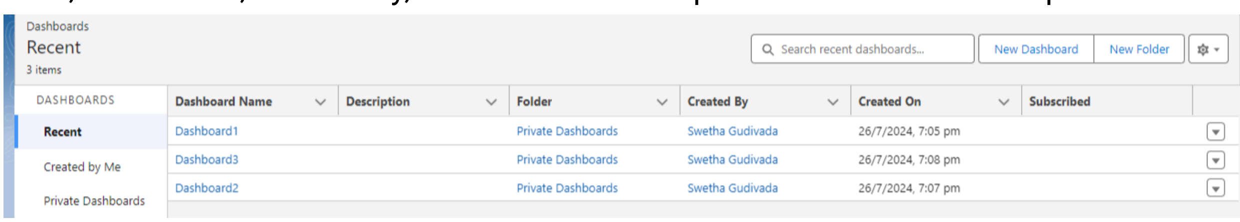


FIG 9.1 Technician and Assignment Details Report

- 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.

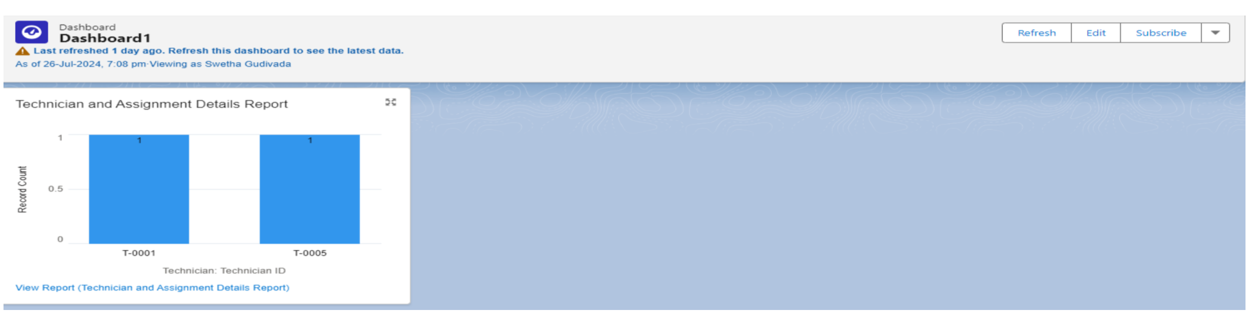


FIG 9.2 Technician and Assignment Details Report Dashboard