FIELD SERVICE WORKORDER OPTIMIZATION

By

Kancharla Lakshmi Sravani

Kancharla.lakshmi.sravani6@gmail.com

ABSTRACT

Objective: The Field Service Work Order Optimization System streamlines operations for acompany providing installations and repairs. Utilizing a robust database, the system efficiently matches work orders with skilled technicians based on technicians' location, availability, and

skills. The system employs a prioritization algorithm, focusing on assigning tasks to technicians.Automated communication keeps technicians informed, while analytics offer insights for continuous improvement. Overall, this solution maximizes efficiency, reduces operational costs,and improves customer satisfaction in the dynamic realm of field service operations

Key Technologies:

1. **Salesforce Field Service:** Salesforce's comprehensive field service management solution will be the core platform, providing tools for scheduling, dispatching, and real-time communication.
2. **Artificial Intelligence & Machine Learning:** Integrated with Salesforce, AI and ML algorithms will predict service demands, optimize scheduling, and match the right technician to the right job based on skill set, location, and availability.
3. **Predictive Analytics:** Leveraging Salesforce's analytics capabilities, predictive models will forecast service needs and preemptively address potential issues by analyzing historical data and current conditions.
4. **Internet of Things (IoT):** IoT devices will provide real-time data from field equipment, which will be integrated into Salesforce for proactive maintenance and swift response to issues.

Implementation Phases:

1. **Salesforce Field Service:** Salesforce's comprehensive field service management solution will be the core platform, providing tools for scheduling, dispatching, and real-time communication.
2. **Artificial Intelligence & Machine Learning:** Integrated with Salesforce, AI and ML algorithms will predict service demands, optimize scheduling, and match the right technician to the right job based on skill set, location, and availability.
3. **Predictive Analytics:** Leveraging Salesforce's analytics capabilities, predictive models will forecast service needs and preemptively address potential issues by analyzing historical data and current conditions.
4. **Internet of Things (IoT):** IoT devices will provide real-time data from field equipment, which will be integrated into Salesforce for proactive maintenance and swift response to issues.

Potential Challenges:

1. **Data Integration:** Ensuring seamless integration of various data sources and legacy systems.
2. **Change Management:** Managing the transition and ensuring buy-in from all stakeholders.
3. **Scalability:** Ensuring the solution can scale to accommodate growth and increased demand.
4. **Security and Privacy:** Protecting sensitive customer and operational data from breaches.

Measurable Outcomes:

1. Efficiency Metrics
2. Customer Satisfaction
3. Operational Excellence

Functional Requirements:

1. Work Order Management
2. Scheduling and Dispatching
3. Resource Management
4. Mobile Access
5. Customer Communication
6. Analytics and Reporting
7. Integration
8. User Management and Security
9. Maintenance and Support

By fulfilling these functional requirements, the Salesforce Field Service Work Order Optimization project will enhance the efficiency of field operations, improve customer satisfaction, and achieve overall business objectives

INDEX PAGE

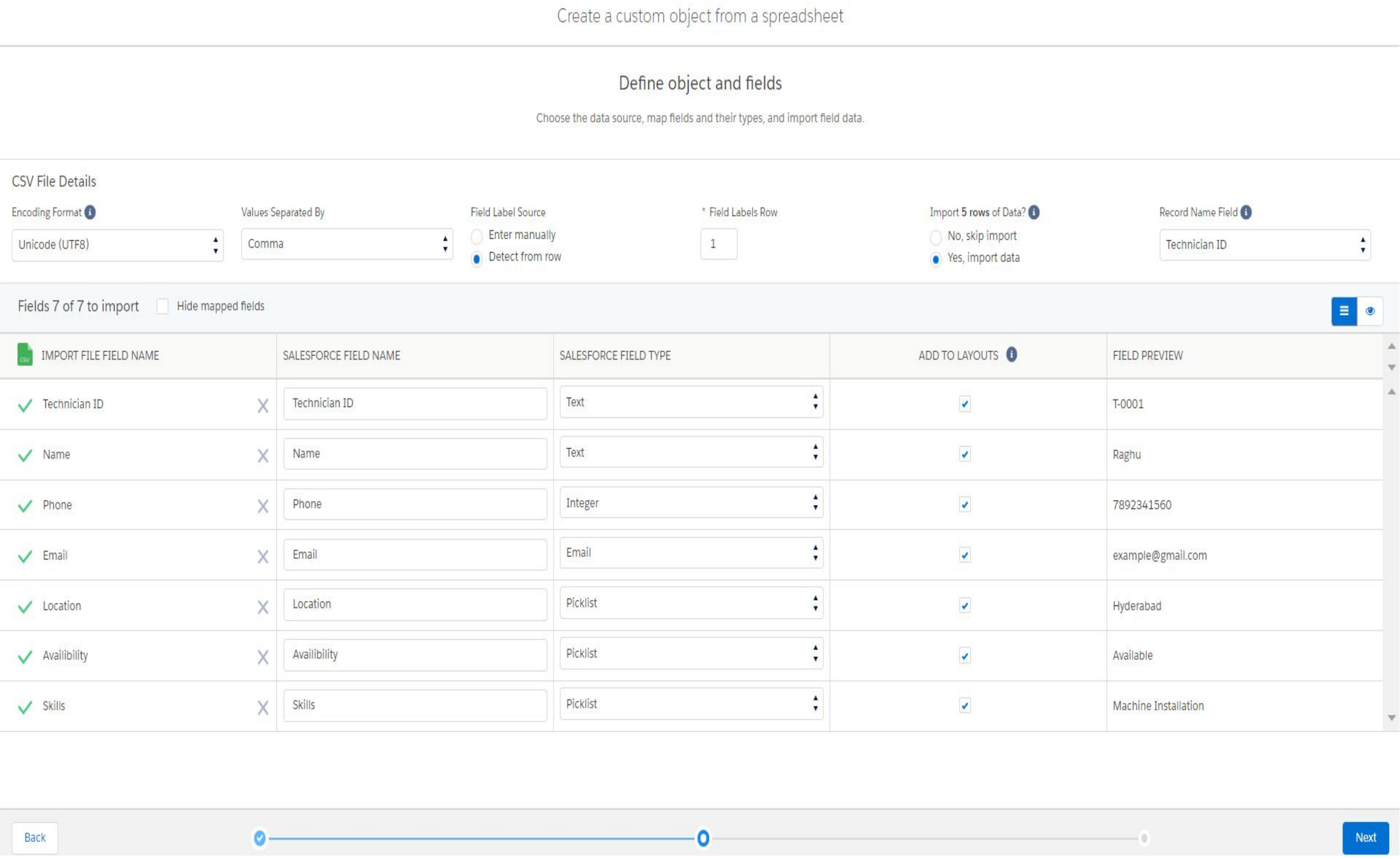
INTRODUCTION

The Field Service Work Order Optimization System streamlines operations for a company providing installations and repairs. Utilizing a robust database, the system efficiently matches work orders with skilled technicians based on technicians' location, availability, and skills. The system employs a prioritization algorithm, focusing on assigning tasks to technicians. Automated communication keeps technicians informed, while analytics offer insights for continuous improvement. Overall, this solution maximizes efficiency, reduces operational costs, and improves customer satisfaction in the dynamic realm of field service operations.

Task 1:

1.1 Create Technician Object:

An entity representing field technicians, capturing details like skills, name, location, availability, and contact information for optimized service dispatch.



Create a custom object from a spreadsheet

Define object and fields

Choose the data source, map fields and their types, and import field data.

CSV File Details

Encoding Format: Unicode (UTF8)

Values Separated By: Comma

Field Label Source: ○ Enter manually ● Detect from row

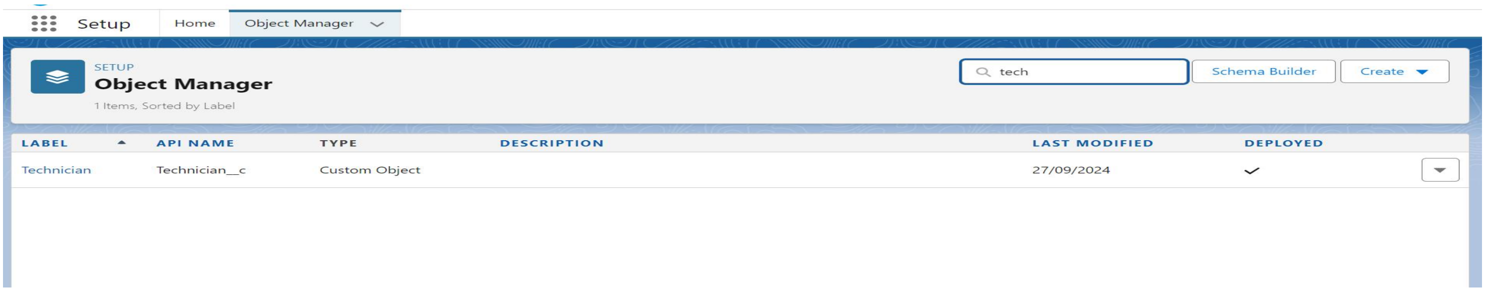
* Field Labels Row: 1

Import 5 rows of Data? ○ No, skip import ● Yes, import data

Record Name Field: Technician ID

Fields 7 of 7 to import □ Hide mapped fields

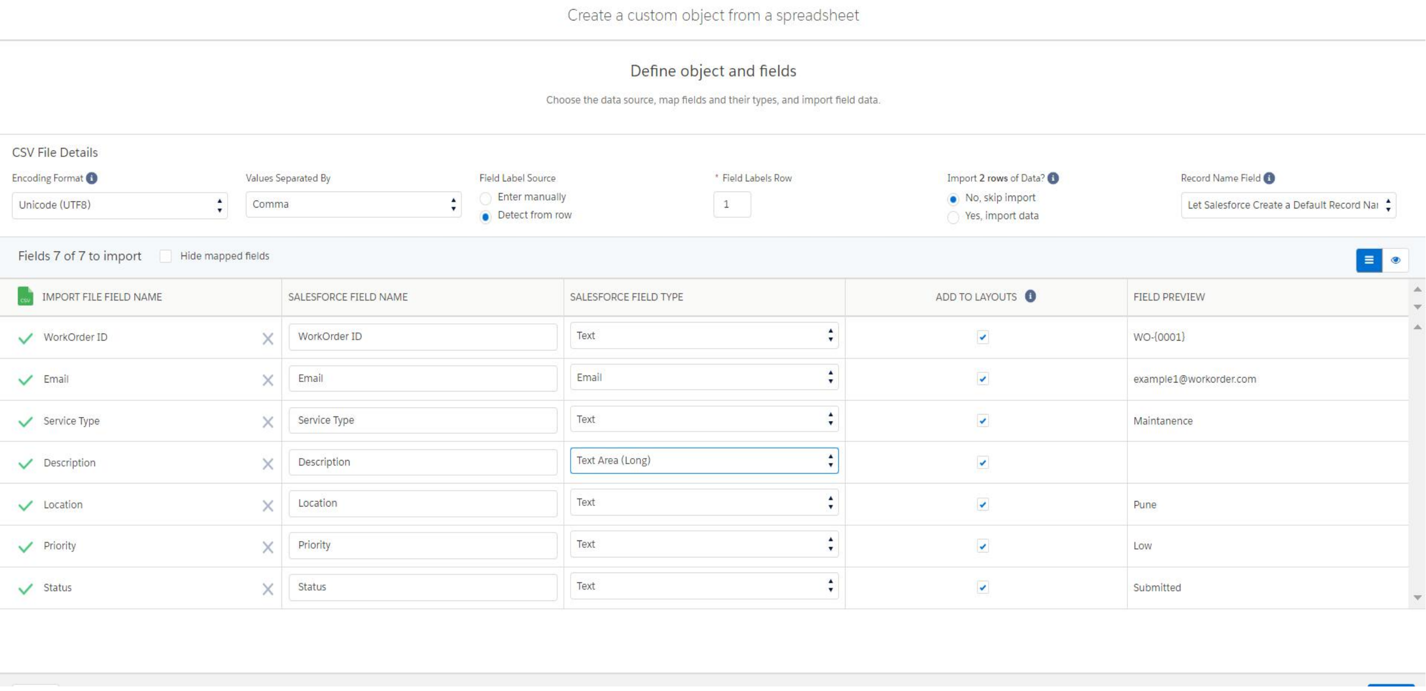
IMPORT FILE FIELD NAME	SALESFORCE FIELD NAME	SALESFORCE FIELD TYPE	ADD TO LAYOUTS	FIELD PREVIEW
✓ Technician ID	Technician ID	Text	☑	T-0001
✓ Name	Name	Text	☑	Raghu
✓ Phone	Phone	Integer	☑	7892341560
✓ Email	Email	Email	☑	example@gmail.com
✓ Location	Location	Picklist	☑	Hyderabad
✓ Availibility	Availibility	Picklist	☑	Available
✓ Skills	Skills	Picklist	☑	Machine Installation



LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Technician	Technician__c	Custom Object		27/09/2024	✓

1.2 Create WorkOrder Object:

An entity tracking service tasks, detailing job requirements, status, assigned technician, and customer information for efficient field operations.


Create a custom object from a spreadsheet

Define object and fields

Choose the data source, map fields and their types, and import field data.

CSV File Details

Encoding Format: Unicode (UTF8)

Values Separated By: Comma

Field Label Source: ○ Enter manually ● Detect from row

* Field Labels Row: 1

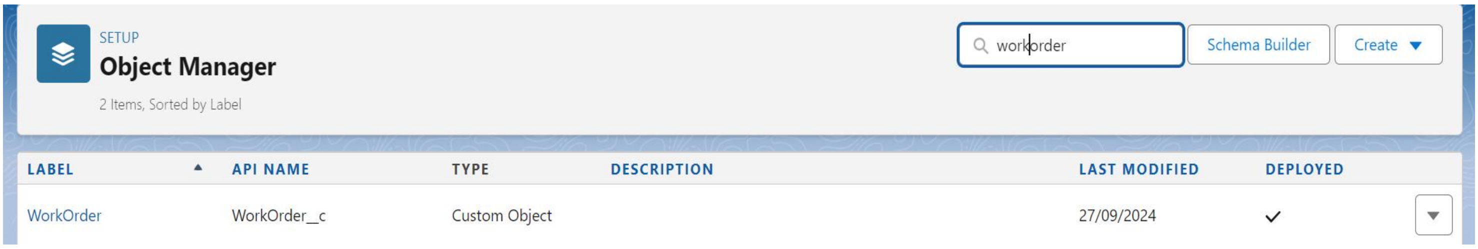
Import 2 rows of Data?: ● No, skip import ○ Yes, import data

Record Name Field: Let Salesforce Create a Default Record Nai

Fields 7 of 7 to import □ Hide mapped fields

IMPORT FILE FIELD NAME	SALESFORCE FIELD NAME	SALESFORCE FIELD TYPE	ADD TO LAYOUTS	FIELD PREVIEW
✓ WorkOrder ID	WorkOrder ID	Text	☑	WO-{0001}
✓ Email	Email	Email	☑	example1@workorder.com
✓ Service Type	Service Type	Text	☑	Maintanence
✓ Description	Description	Text Area (Long)	☑	
✓ Location	Location	Text	☑	Pune
✓ Priority	Priority	Text	☑	Low
✓ Status	Status	Text	☑	Submitted

After creating the WorkOrder Custom object it looks like the below

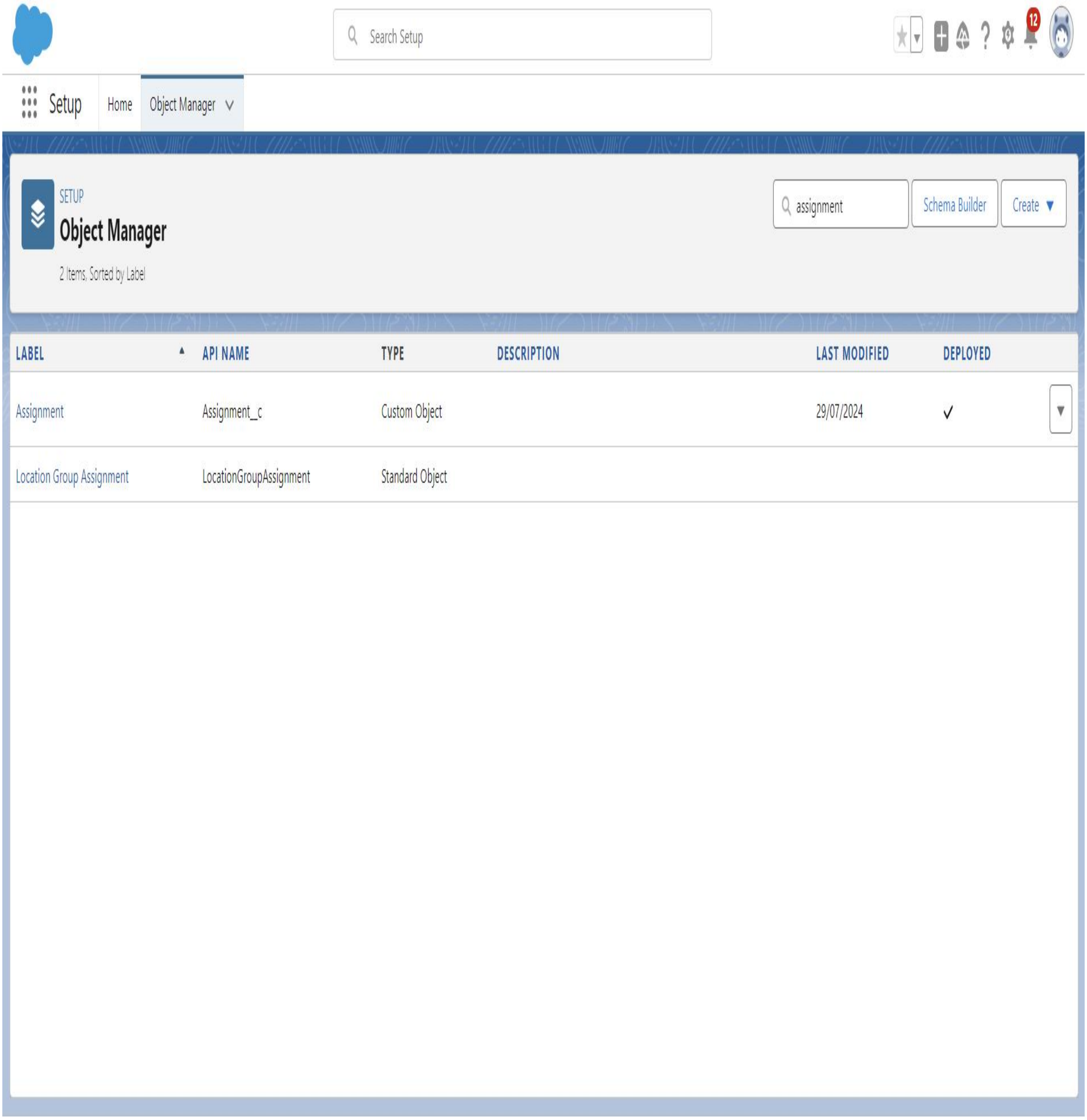


LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
WorkOrder	WorkOrder__c	Custom Object		27/09/2024	✓

1.3 Create Assignment Object :

An entity linking technicians to work orders, detailing assignment dates, priority, status, and specific tasks for optimized field service.

After creating the Assignment custom object, the object manager bar looks the below



2 Items, Sorted by Label

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Assignment	Assignment_c	Custom Object		29/07/2024	✓
Location Group Assignment	LocationGroupAssignment	Standard Object			

Task 2:

Creating a Custom Tab

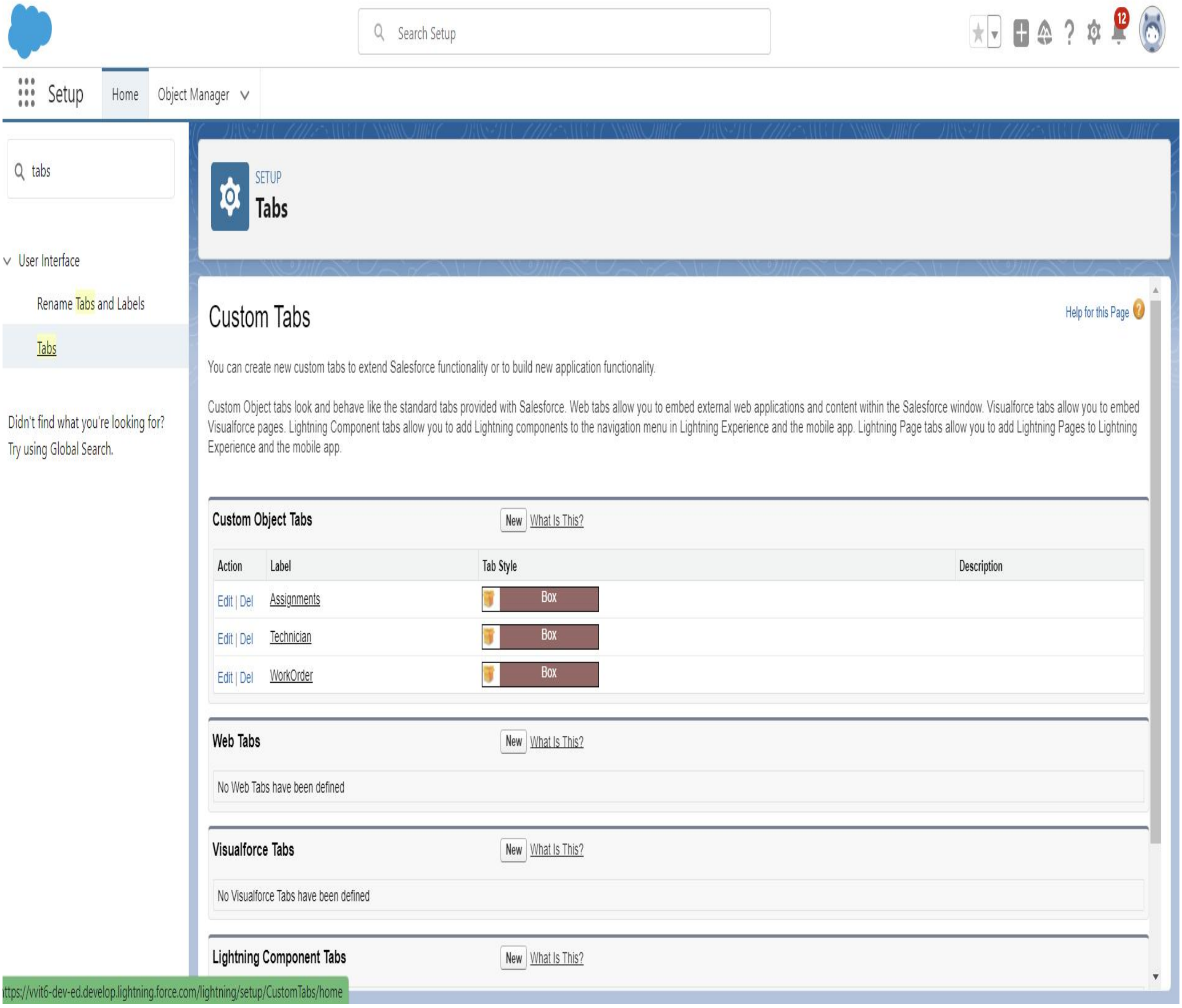
A user interface element in Salesforce that provides access to custom objects, records, or web content, enhancing navigation and organization of data within the Salesforce environment.

To create a Tab:(Assignment)

1. Go to the setup page --> type Tabs in the Quick Find bar --> click on tabs --> New (under the custom object tab)
2. Select Object(Assignment) --> Select any tab style --> Next (Add to profiles page) keep it as default --> Next (Add to Custom App) keep it as default --> Save.

Note: Tabs for WorkOrder & Technician objects do get created automatically. We do not need to create tabs for those objects.

After following the above steps, the output looks like this:



Custom Tabs

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	Assignments	Box	
Edit | Del	Technician	Box	
Edit | Del	WorkOrder	Box	

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?

ttps://wit6-dev-ed.develop.lightning.force.com/lightning/setup/CustomTabs/home

Task 3 :

Create a Lightning App

To create a lightning app page:

1. Go to the setup page --> search "app manager" in quick find --> select "app manager" --> click on New lightning App.
2. Fill the app name in app details and branding as follow

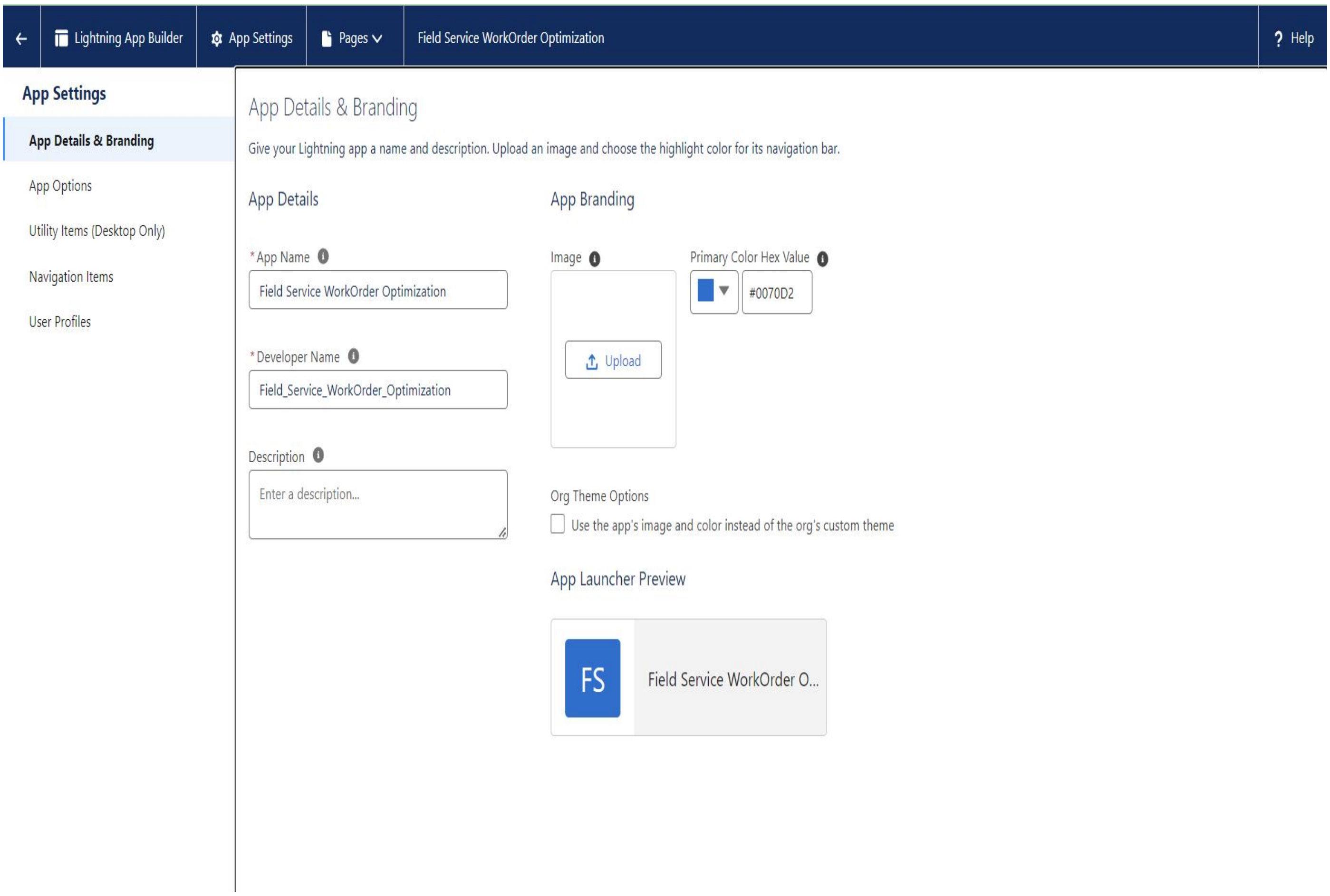
App Name : Field Service WorkOrder Optimization

Developer Name : this will be auto populated

Description : Give a meaningful description

Image : optional (if you want to give any image you can, otherwise not mandatory)

Primary color hex value : keep this default



App Details & Branding

Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.

App Details

*App Name: Field Service WorkOrder Optimization

*Developer Name: Field_Service_WorkOrder_Optimization

Description: Enter a description...

App Branding

Image

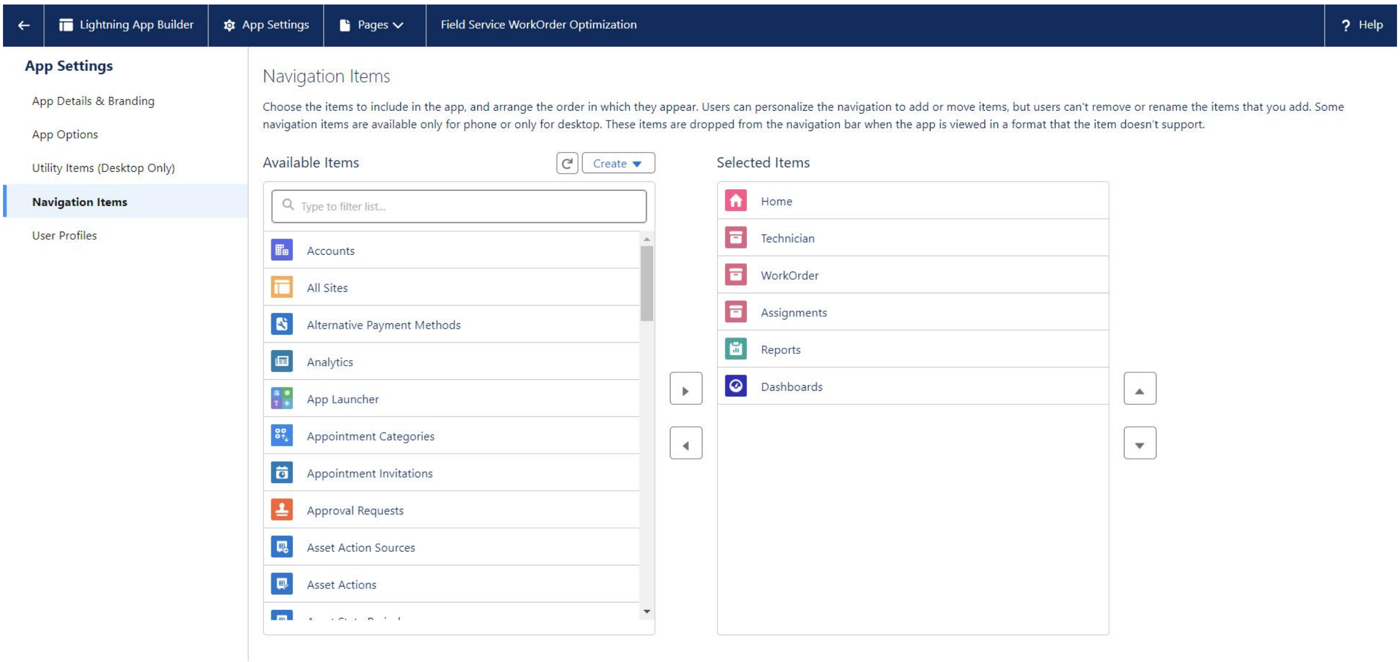
Primary Color Hex Value: #0070D2

Org Theme Options

□ Use the app's image and color instead of the org's custom theme

App Launcher Preview

3. Then click Next --> (App option page) keep it as default --> Next --> (Utility Items) keep it as default --> Next
4. To Add Navigation Items:



Navigation Items

Choose the items to include in the app, and arrange the order in which they appear. Users can personalize the navigation to add or move items, but users can't remove or rename the items that you add. Some navigation items are available only for phone or only for desktop. These items are dropped from the navigation bar when the app is viewed in a format that the item doesn't support.

Available Items

- Accounts
- All Sites
- Alternative Payment Methods
- Analytics
- App Launcher
- Appointment Categories
- Appointment Invitations
- Approval Requests
- Asset Action Sources
- Asset Actions

Selected Items

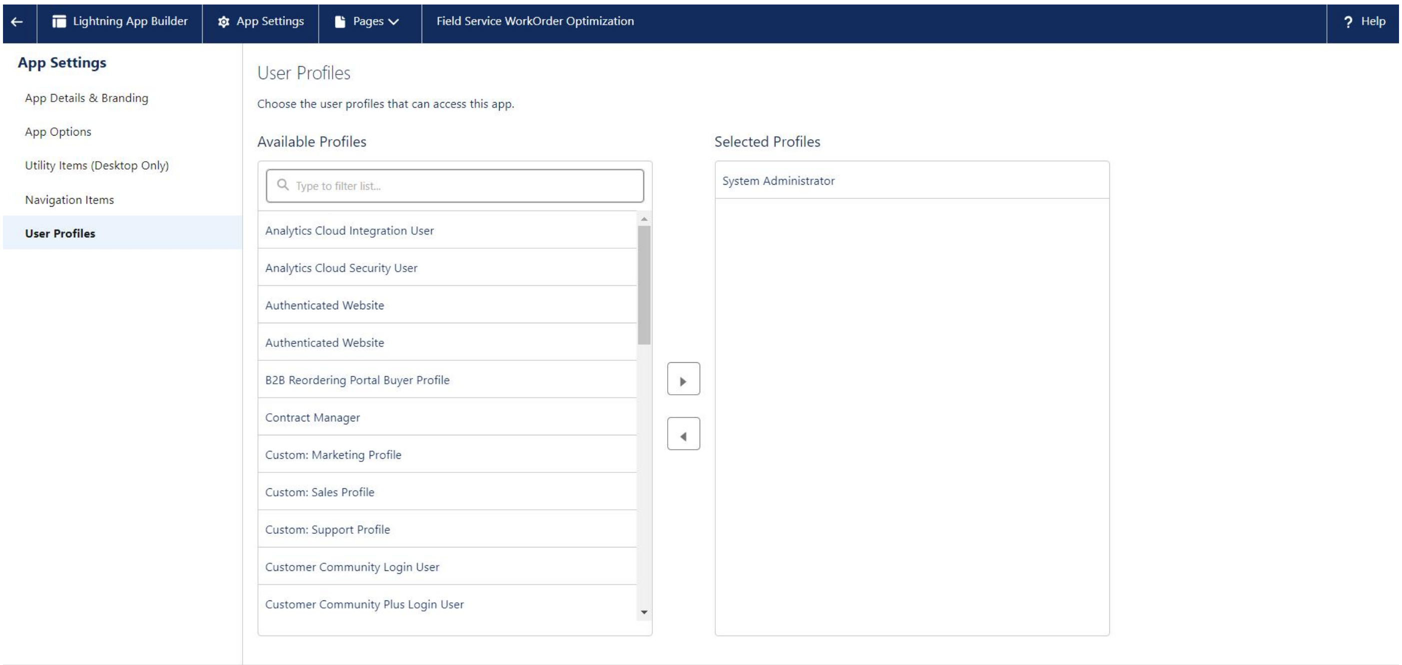
- Home
- Technician
- WorkOrder
- Assignments
- Reports
- Dashboards

Search the items in the search bar(Home, WorkOrder, Technician, Assignment, Reports, Dashboard) from the search bar and move it using the arrow button ? Next.

Note: select asset the custom object which we have created in the previous activity.

5. To Add User Profiles:

Search profiles (System administrator) in the search bar --> click on the arrow button --> save & finish



User Profiles

Choose the user profiles that can access this app.

Available Profiles

- Analytics Cloud Integration User
- Analytics Cloud Security User
- Authenticated Website
- Authenticated Website
- B2B Reordering Portal Buyer Profile
- Contract Manager
- Custom: Marketing Profile
- Custom: Sales Profile
- Custom: Support Profile
- Customer Community Login User
- Customer Community Plus Login User

Selected Profiles

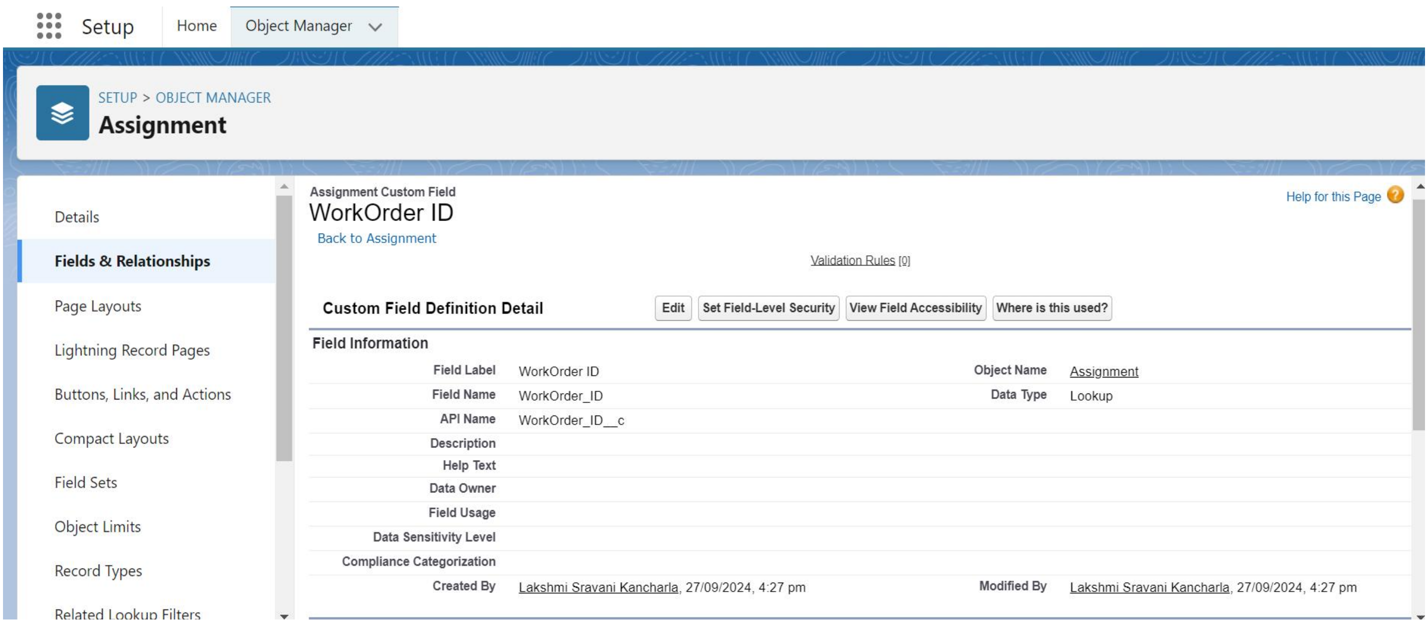
- System Administrator

This is the output after completion of following the above procedure.

Task 4 :

4.1 Creating Lookup Field in Assignment Object

A lookup field in the Assignment Object establishes a relationship with another object, such as Technicians or Work Orders, enabling users to link and reference related records for improved data organization and relational tracking.



Assignment Custom Field

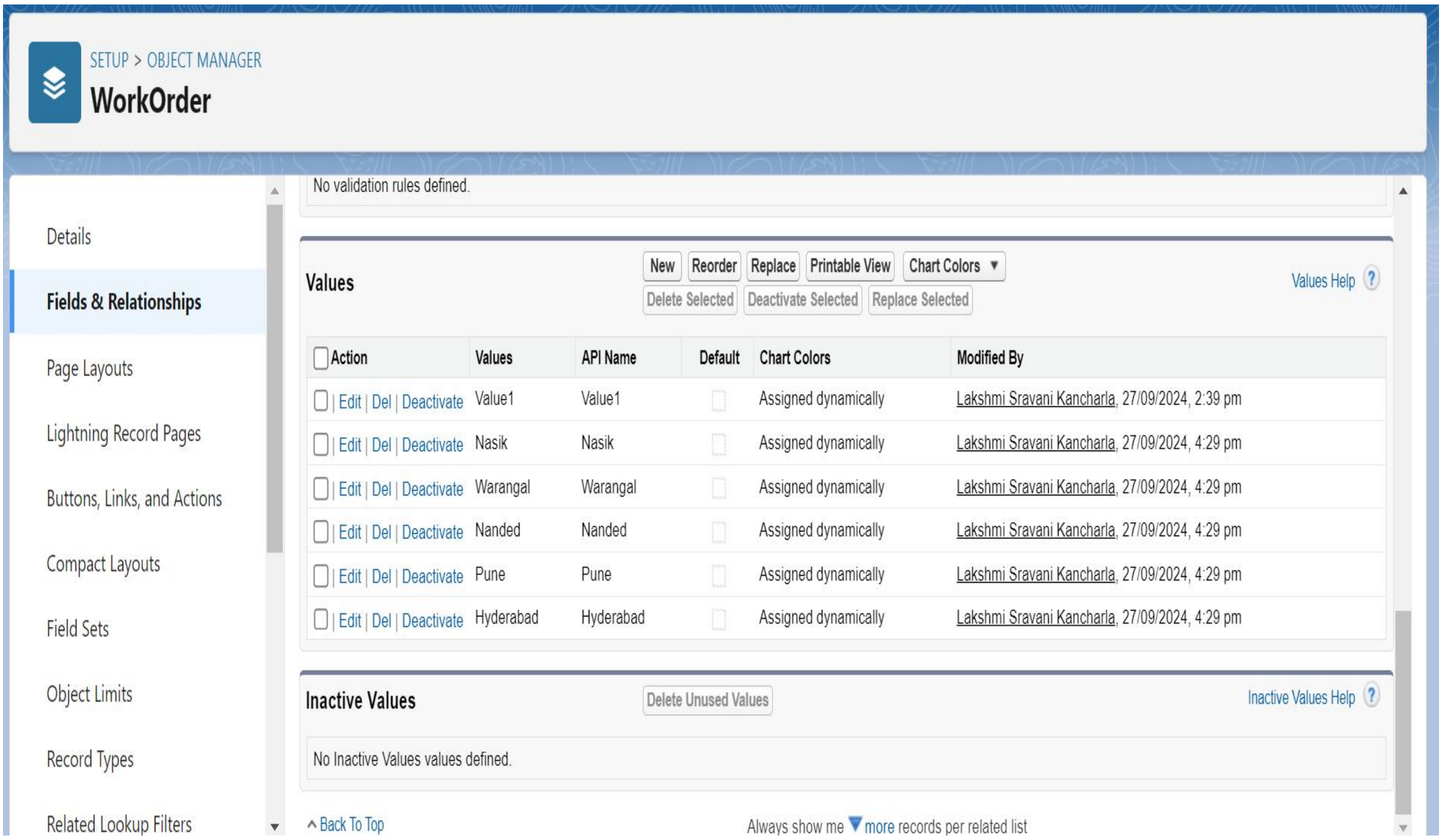
WorkOrder ID

Custom Field Definition Detail

Field Information

Field Label	WorkOrder ID	Object Name	Assignment
Field Name	WorkOrder_ID	Data Type	Lookup
API Name	WorkOrder_ID__c		
Description			
Help Text			
Data Owner			
Field Usage			
Data Sensitivity Level			
Compliance Categorization			
Created By	Lakshmi Sravani Kancharla, 27/09/2024, 4:27 pm	Modified By	Lakshmi Sravani Kancharla, 27/09/2024, 4:27 pm

4.2 Manage your picklist values



No validation rules defined.

Values

□ Action	Values	API Name	Default	Chart Colors	Modified By
□ | Edit | Del | Deactivate	Value1	Value1	□	Assigned dynamically	Lakshmi Sravani Kancharla, 27/09/2024, 2:39 pm
□ | Edit | Del | Deactivate	Nasik	Nasik	□	Assigned dynamically	Lakshmi Sravani Kancharla, 27/09/2024, 4:29 pm
□ | Edit | Del | Deactivate	Warangal	Warangal	□	Assigned dynamically	Lakshmi Sravani Kancharla, 27/09/2024, 4:29 pm
□ | Edit | Del | Deactivate	Nanded	Nanded	□	Assigned dynamically	Lakshmi Sravani Kancharla, 27/09/2024, 4:29 pm
□ | Edit | Del | Deactivate	Pune	Pune	□	Assigned dynamically	Lakshmi Sravani Kancharla, 27/09/2024, 4:29 pm
□ | Edit | Del | Deactivate	Hyderabad	Hyderabad	□	Assigned dynamically	Lakshmi Sravani Kancharla, 27/09/2024, 4:29 pm

Inactive Values

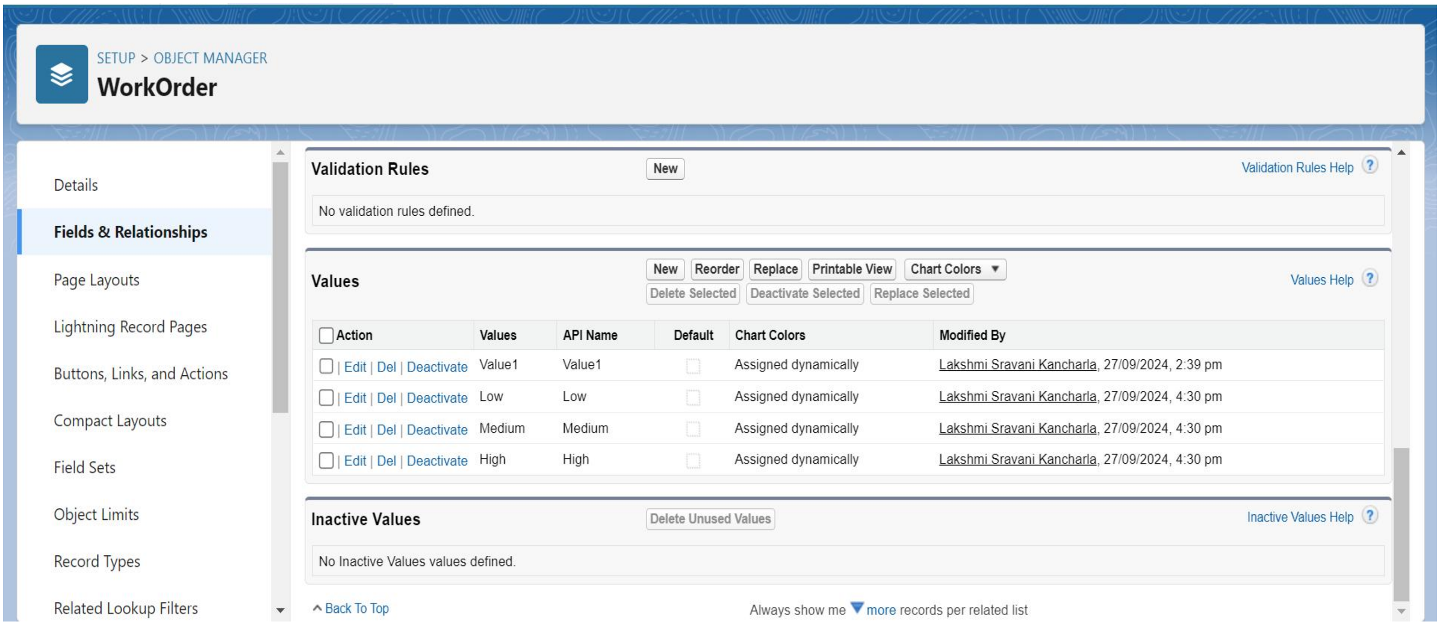
No Inactive Values values defined.

Always show me more records per related list

4.3 Manage your picklist values :

Add following values to the respective fields in WorkOrder object:

Field	Values
Priority	High
Service Type	Hardware repair Troubleshoot/Debugging Lane-Management



Validation Rules

No validation rules defined.

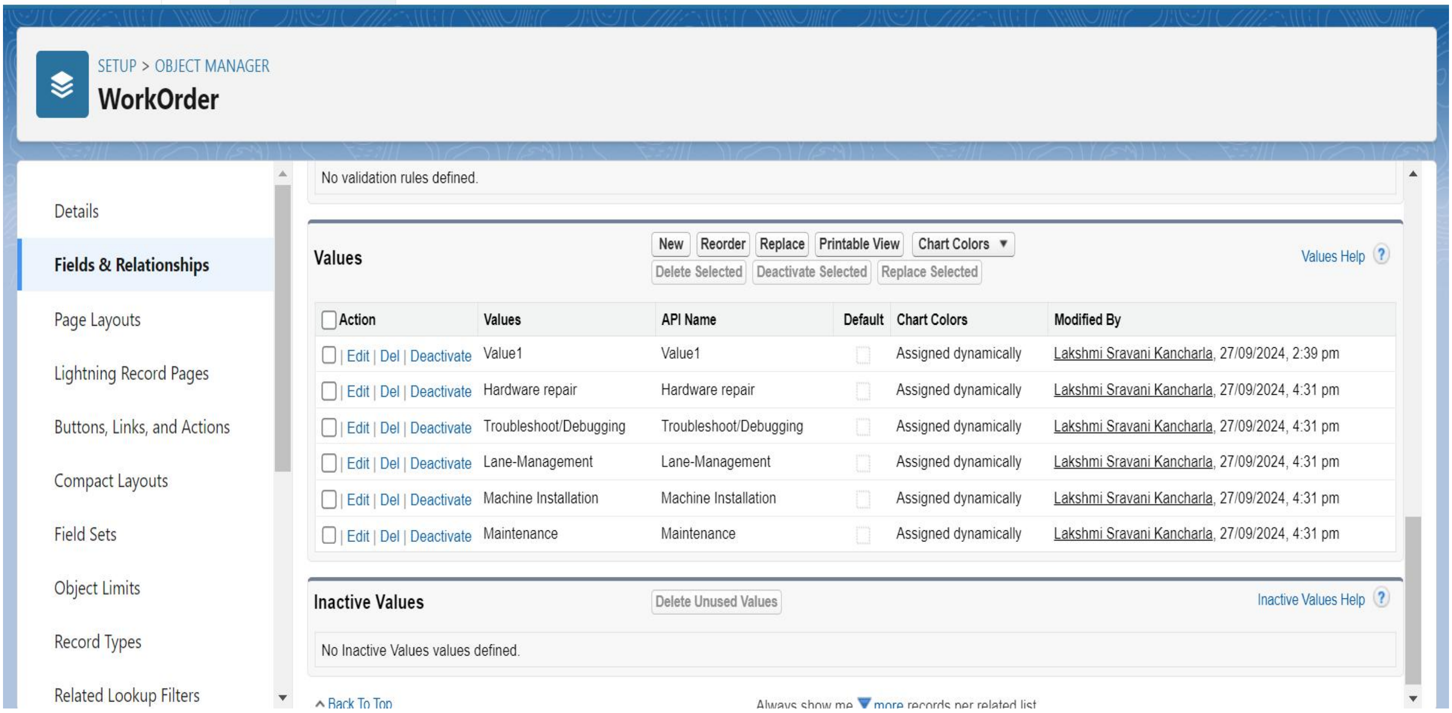
Values

Action	Values	API Name	Default	Chart Colors	Modified By
□ | Edit | Del | Deactivate	Value1	Value1	□	Assigned dynamically	Lakshmi Sravani Kancharla, 27/09/2024, 2:39 pm
□ | Edit | Del | Deactivate	Low	Low	□	Assigned dynamically	Lakshmi Sravani Kancharla, 27/09/2024, 4:30 pm
□ | Edit | Del | Deactivate	Medium	Medium	□	Assigned dynamically	Lakshmi Sravani Kancharla, 27/09/2024, 4:30 pm
□ | Edit | Del | Deactivate	High	High	□	Assigned dynamically	Lakshmi Sravani Kancharla, 27/09/2024, 4:30 pm

Inactive Values

No Inactive Values values defined.

Always show me more records per related list



No validation rules defined.

Values

Action	Values	API Name	Default	Chart Colors	Modified By
□ | Edit | Del | Deactivate	Value1	Value1	□	Assigned dynamically	Lakshmi Sravani Kancharla, 27/09/2024, 2:39 pm
□ | Edit | Del | Deactivate	Hardware repair	Hardware repair	□	Assigned dynamically	Lakshmi Sravani Kancharla, 27/09/2024, 4:31 pm
□ | Edit | Del | Deactivate	Troubleshoot/Debugging	Troubleshoot/Debugging	□	Assigned dynamically	Lakshmi Sravani Kancharla, 27/09/2024, 4:31 pm
□ | Edit | Del | Deactivate	Lane-Management	Lane-Management	□	Assigned dynamically	Lakshmi Sravani Kancharla, 27/09/2024, 4:31 pm
□ | Edit | Del | Deactivate	Machine Installation	Machine Installation	□	Assigned dynamically	Lakshmi Sravani Kancharla, 27/09/2024, 4:31 pm
□ | Edit | Del | Deactivate	Maintenance	Maintenance	□	Assigned dynamically	Lakshmi Sravani Kancharla, 27/09/2024, 4:31 pm

Inactive Values

No Inactive Values values defined.

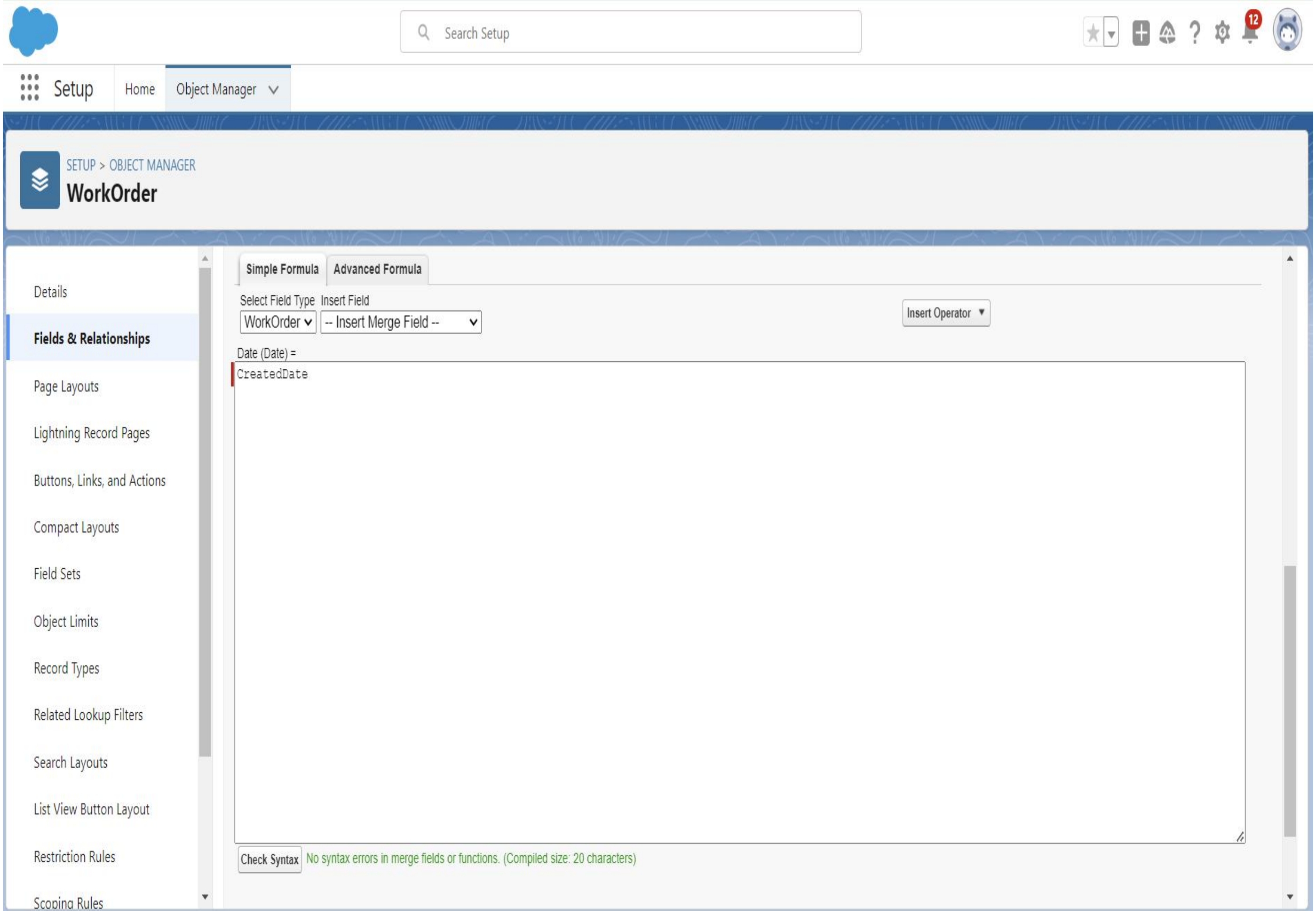
Always show me more records per related list

4.4 Creating Formula Field in WorkOrder Object

A formula field in the Work Order Object automatically calculates and displays data based on other fields or custom logic. This feature streamlines data entry, ensures consistency, and provides real-time insights without manual updates.

1. Repeat steps 1 and 2 mentioned in activity 1
2. Select Data type as "Formula" and click Next.
3. Give Field Label and Field Name as "Date" and select formula return type as "Date" and click next.
4. Under Advanced Formula, write the formula and click "Check Syntax"

   Formula: CreatedDate
5. Next--> Next--> Save.



Date (Date) =

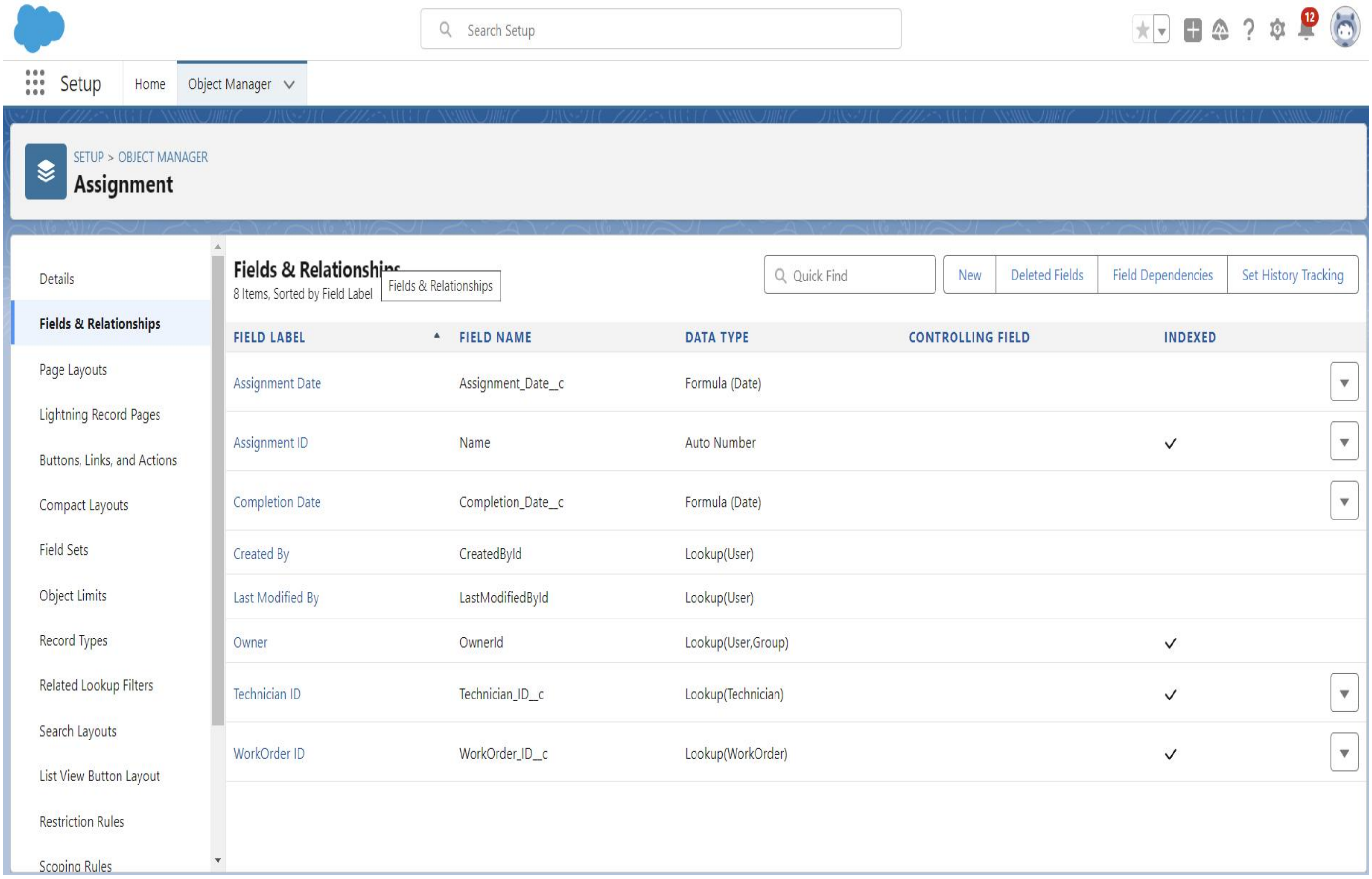
```
CreatedDate
```

No syntax errors in merge fields or functions. (Compiled size: 20 characters)

4.5 Creating Remaining fields for the respective objects

Now create the remaining fields using the data types mentioned in the table.

<table>
<tr><th>Sl No</th><th>Object Name</th><th>Field</th></tr>
<tr><td>1</td><td>Assignment</td><td>
<table>
<tr><th>Field Name</th><th>Datatype</th></tr>
<tr><td>● Technician ID</td><td>Lookup(Technician)</td></tr>
<tr><td>● Assignment Date</td><td>Formula: return type : Date
(WorkOrder_ID__r.Date__c)</td></tr>
<tr><td>● Completion Date</td><td>Formula: return type : Date
IF(ISPICKVAL(WorkOrder_ID__r.Status__c , 'Resolved'), WorkOrder_ID__r.LastModifiedDate , NULL)</td></tr>
</table>
</td></tr>
</table>



Fields & Relationships

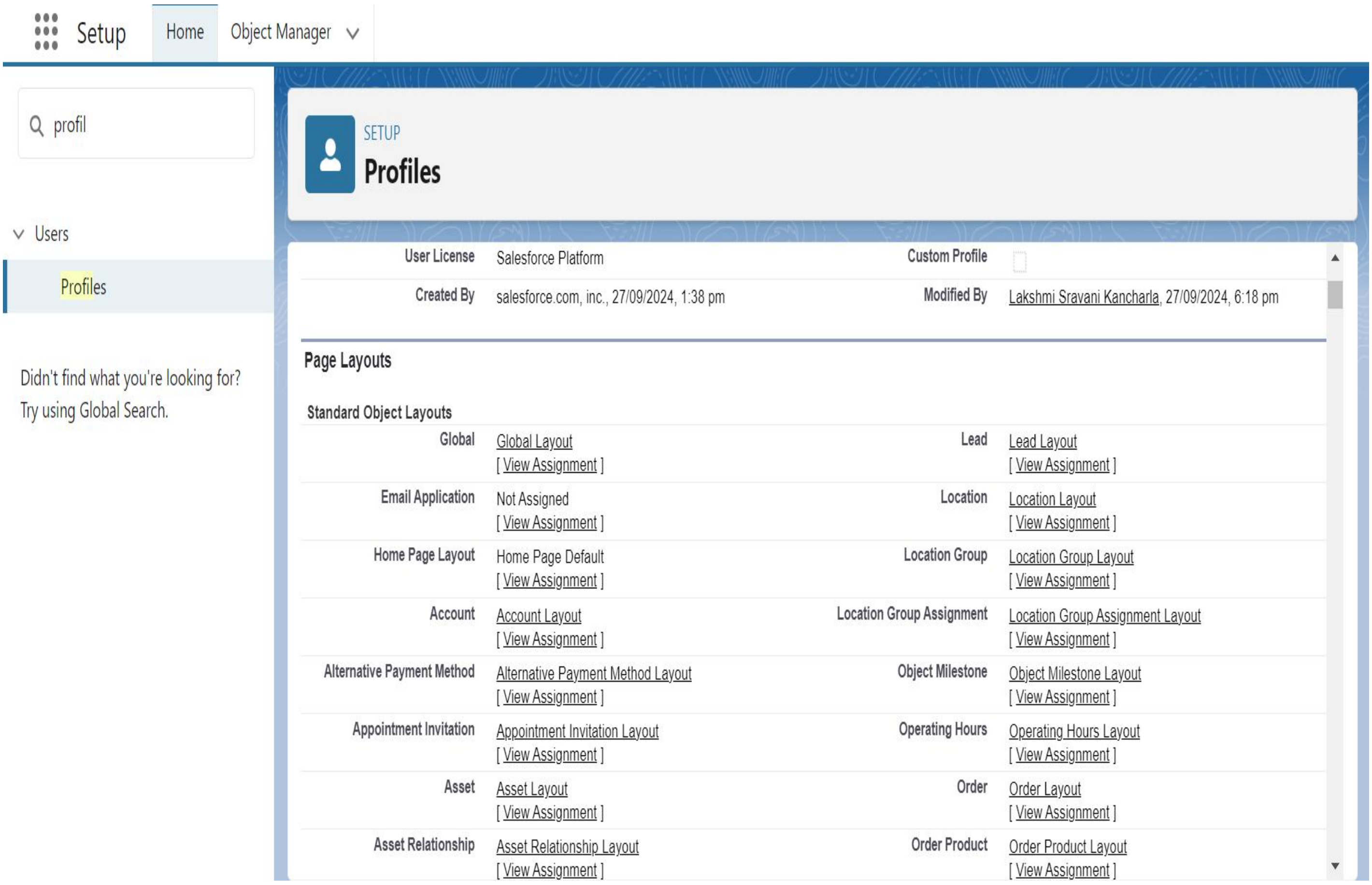
8 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Assignment Date	Assignment_Date__c	Formula (Date)		
Assignment ID	Name	Auto Number		✓
Completion Date	Completion_Date__c	Formula (Date)		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Technician ID	Technician_ID__c	Lookup(Technician)		✓
WorkOrder ID	WorkOrder_ID__c	Lookup(WorkOrder)		✓

Task 5 :

Technician Profile

1. Go to setup --> type profiles in the quick find box --> click on profiles --> click on new profile.
2. Select 'Standard Platform User' for existing profile and give 'Technician' for Profile Name and click on Save.
3. While still on the profile page, then click Edit.
4. While still on the profile page, then click Edit.
5. Scroll down and Click on Save.
6. Now from the profile detail page scroll down to custom field level security click on view next to WorkOrder object.
7. Click on Edit, enable the check box for the status field.
8. Click on Save.



Didn't find what you're looking for? Try using Global Search.

Profiles

User License	Salesforce Platform	Custom Profile	□
Created By	salesforce.com, inc., 27/09/2024, 1:38 pm	Modified By	Lakshmi Sravani Kancharla, 27/09/2024, 6:18 pm

Page Layouts

Standard Object Layouts

Global	Global Layout [View Assignment]	Lead	Lead Layout [View Assignment]
Email Application	Not Assigned [View Assignment]	Location	Location Layout [View Assignment]
Home Page Layout	Home Page Default [View Assignment]	Location Group	Location Group Layout [View Assignment]
Account	Account Layout [View Assignment]	Location Group Assignment	Location Group Assignment Layout [View Assignment]
Alternative Payment Method	Alternative Payment Method Layout [View Assignment]	Object Milestone	Object Milestone Layout [View Assignment]
Appointment Invitation	Appointment Invitation Layout [View Assignment]	Operating Hours	Operating Hours Layout [View Assignment]
Asset	Asset Layout [View Assignment]	Order	Order Layout [View Assignment]
Asset Relationship	Asset Relationship Layout [View Assignment]	Order Product	Order Product Layout [View Assignment]

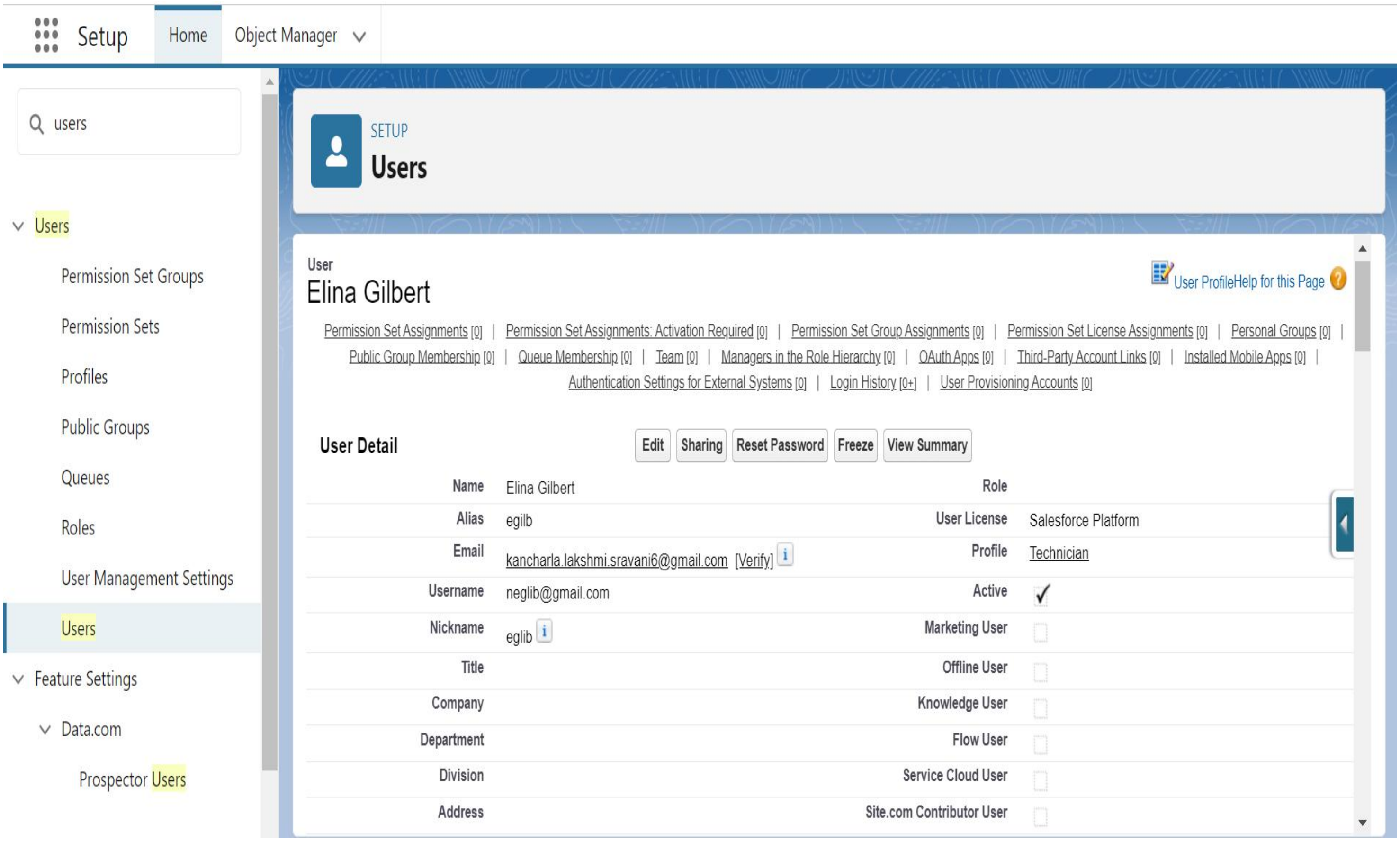
Task 6 :

Create User

User is engaged in the Field Service Workforce Optimization Project, utilizing Salesforce to optimize field operations, improve resource management, and enhance customer service through
efficient scheduling, real-time tracking, and comprehensive analytics.

1. Go to setup --> type users in the quick find box --> select users --> click New user.
2. Fill in the fields

1. First Name : Elina
2. Last Name : Gilbert
3. Alias : Give an Alias Name
4. Email id : Give your Personal Email id
5. Username : Username should be in this form: text@text.text
6. Nick Name : Give a Nickname
7. Role :
8. User license : Salesforce Platform
9. Profiles : Technician


Setup
Home
Object Manager
users
Users
Permission Set Groups
Permission Sets
Profiles
Public Groups
Queues
Roles
User Management Settings
Users
Feature Settings
Data.com
Prospector Users
SETUP
Users
User
User ProfileHelp for this Page
Permission Set Assignments [0] | Permission Set Assignments: Activation Required [0] | Permission Set Group Assignments [0] | Permission Set License Assignments [0] | Personal Groups [0] |
Public Group Membership [0] | Queue Membership [0] | Team [0] | Managers in the Role Hierarchy [0] | OAuth Apps [0] | Third-Party Account Links [0] | Installed Mobile Apps [0] |
Authentication Settings for External Systems [0] | Login History [0+] | User Provisioning Accounts [0]
Edit
Sharing
Reset Password
Freeze
View Summary

Elina Gilbert

User Detail

Name	Elina Gilbert	Role	
Alias	egilb	User License	Salesforce Platform
Email	kancharla.lakshmi.sravani6@gmail.com [Verify]	Profile	Technician
Username	neglib@gmail.com	Active	✓
Nickname	eglib	Marketing User	□
Title		Offline User	□
Company		Knowledge User	□
Department		Flow User	□
Division		Service Cloud User	□
Address		Site.com Contributor User	□

Task 7 :

7.1 Create an Apex Class

1. Go to Setup --> Click on the gear icon --> Select Developer Console.
2. Then we can see the Developer console. Click on the developer console and you will navigate to a new console window.
3. To create a new Apex Class follow the below steps:

Click on the file --> New --> Apex Class.

4. Give the Apex Class name as "WorkOrderClass" .
5. Click ok.
6. Now write the code logic here
7. **Source Code:**

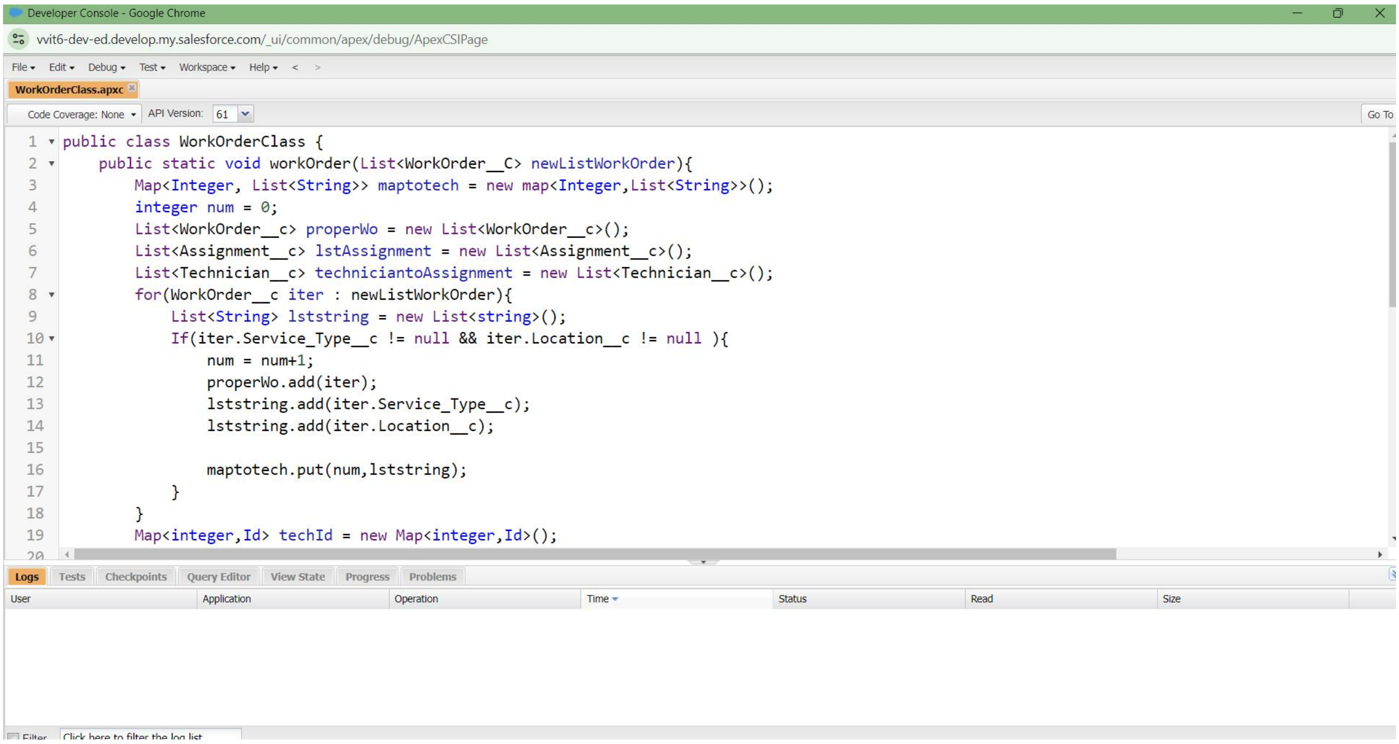
```
public class WorkOrderClass {
    public static void workOrder(List<WorkOrder__C> newListWorkOrder){
        Map<Integer, List<String>> maptotech = new map<Integer,List<String>>();
        integer num = 0;
        List<WorkOrder__c> properWo = new List<WorkOrder__c>();
        List<Assignment__c>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

        }
          }

        }
        integer num3 = 0;
        For(WorkOrder__c W : properWo){
          num3 = num3 + 1;
          Assignment__c A = new Assignment__c();
          A.WorkOrder_ID__c = W.Id;
          A.Technician_ID__c = techid.get(num3);
          lstAssignment.add(A);
        }
        If(!lstAssignment.IsEmpty()){
          insert lstAssignment;
        }
      }
    }
```

8. Save the code.(click on file --> Save)



```
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>();
```

7.2 Create an Apex Trigger

1. To create a new Apex Class follow the below steps:

Click on the file --> New --> Apex Class.

2. Give the Apex Trigger name as “WorkOrderTrigger” , and select “WorkOrder__c” from the

dropdown for sObject.

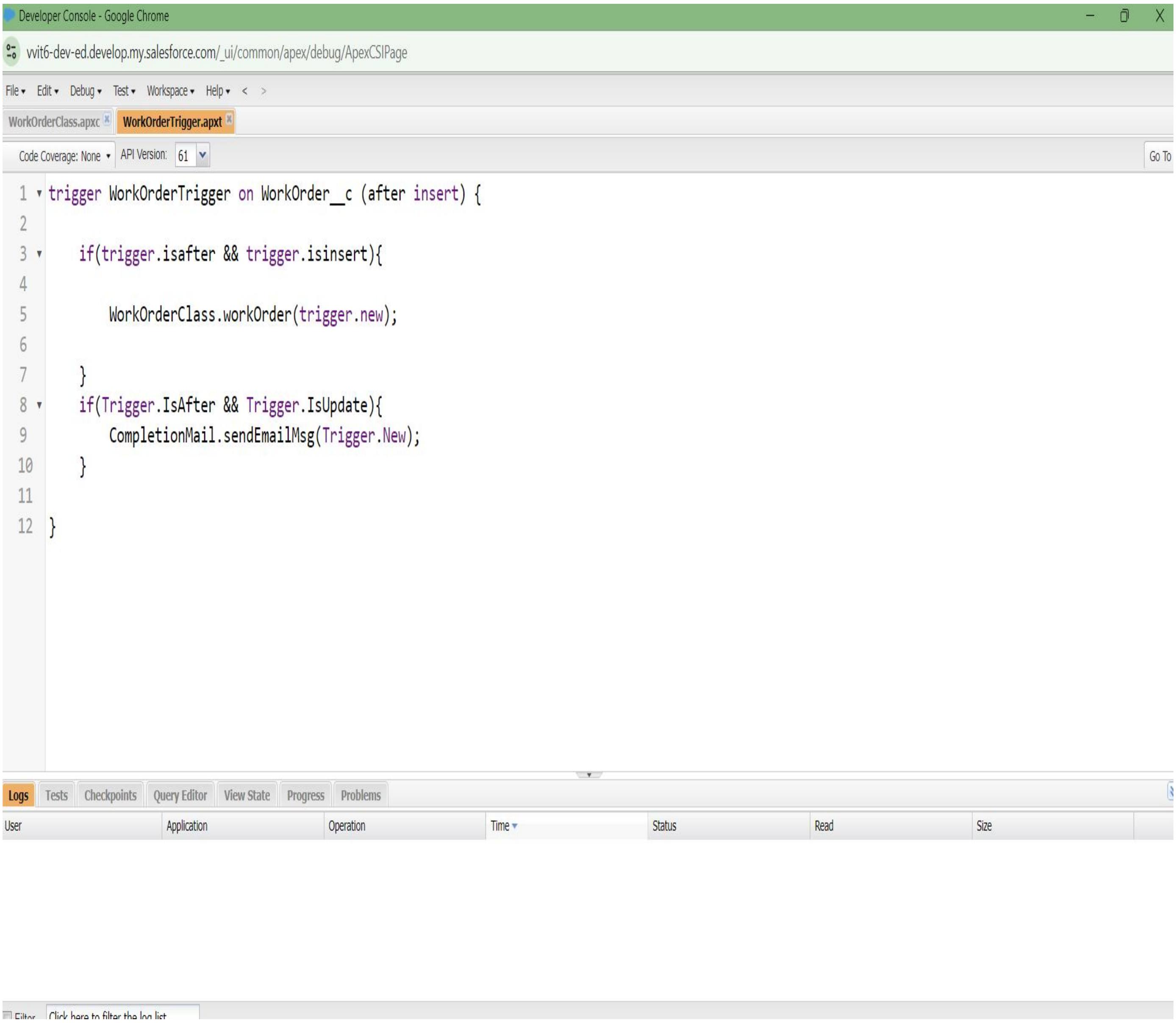
3. Click Submit.

4. Now write the code logic here

Source Code:

```
trigger WorkOrderTrigger on WorkOrder__c (after insert) {
    if(trigger.isafter && trigger.isinsert){
        WorkOrderClass.workOrder(trigger.new);
    }
}
```

5. Save the code.(click on file --> Save)


Developer Console - Google Chrome
vvit6-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
WorkOrderClass.apxc
WorkOrderTrigger.apxt
Code Coverage: None ▾
API Version: 61
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

```
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);
    }

}
```

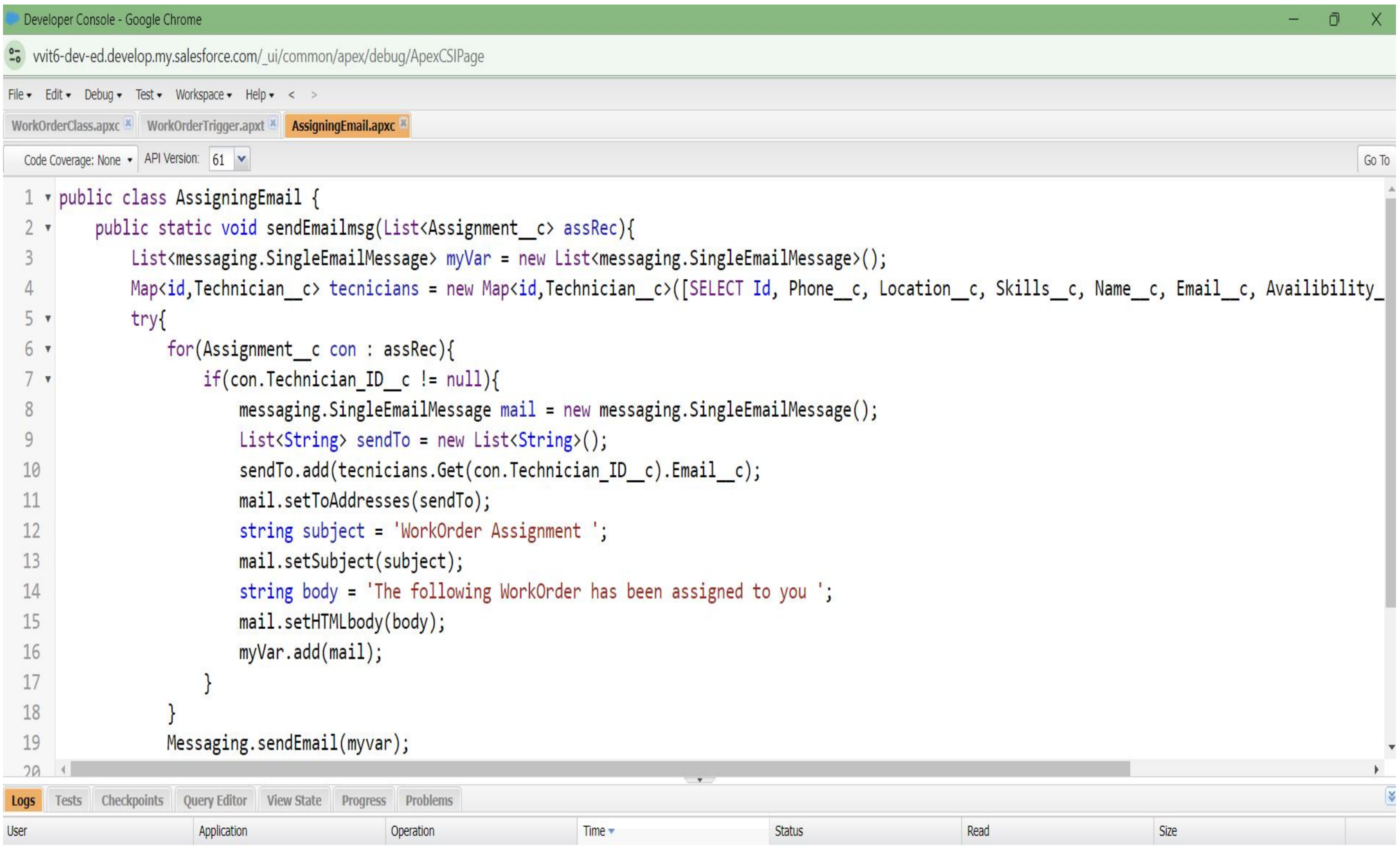
7.3 Create an Apex Class

1. Go to Setup --> Click on the gear icon --> Select Developer Console.
2. Then we can see the Developer console. Click on the developer console and you will navigate to a new console window.
3. To create a new Apex Class follow the below steps:
Click on the file --> New --> Apex Class.
4. Give the Apex Class name as “AssigningEmail”.
5. Click ok.
6. Now write the code logic here
7. **Source Code:**

```
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());
        }
    }
```

}

8. Save the code.(click on file --> Save)



```
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_
        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);
```

7.4 Create an Apex Trigger

To create a new Apex Class follow the below steps:

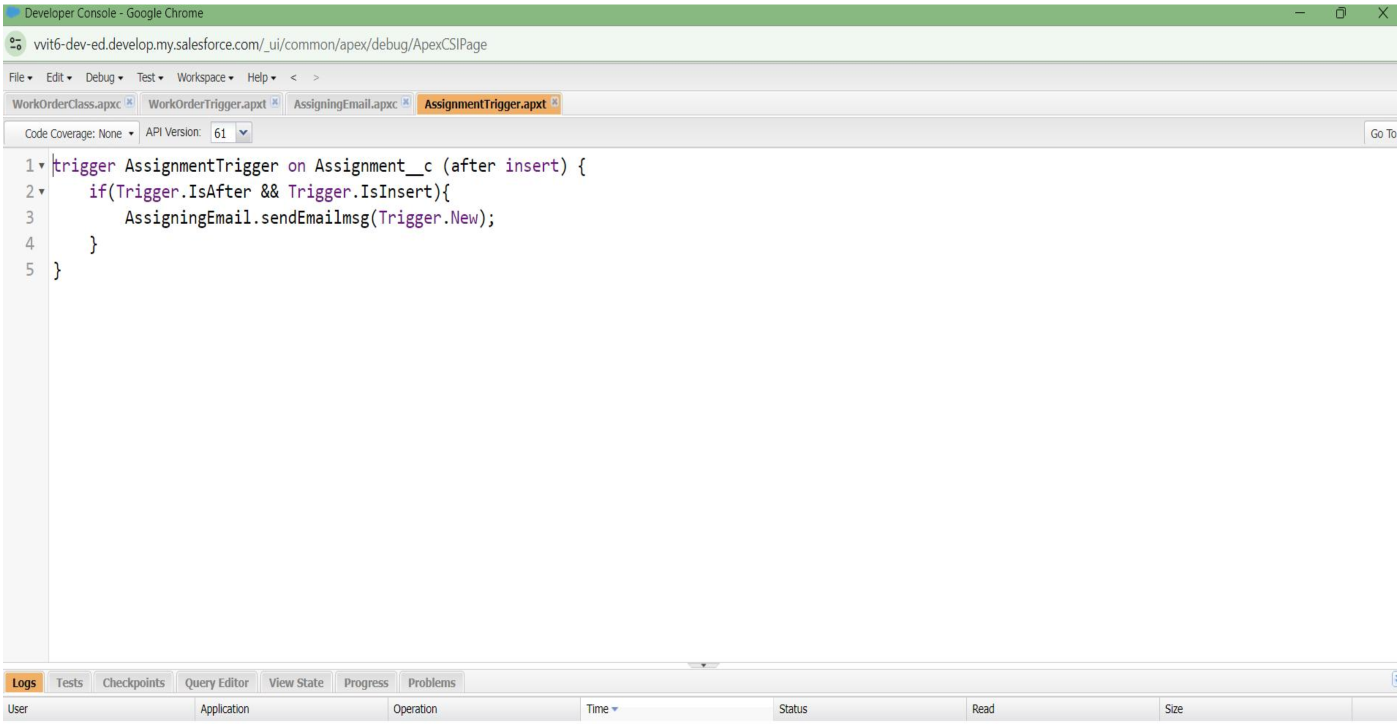
1. Click on the file --> New --> Apex Class.
2. Give the Apex Trigger name as “AssignmentTrigger” , and select “Assignment__c” from the

dropdown for sObject.

3. Click Submit.
4. Now write the code logic here
5. **Source Code:**

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

6. Save the code.(click on file --> Save)



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

7.5 Create an Apex Class

1. Go to Setup --> Click on the gear icon --> Select Developer Console.

2. Then we can see the Developer console. Click on the developer console and you will navigate to a new console window.

3. To create a new Apex Class follow the below steps:

Click on the file --> New --> Apex Class.

4. Give the Apex Class name as “CompletionMail”.

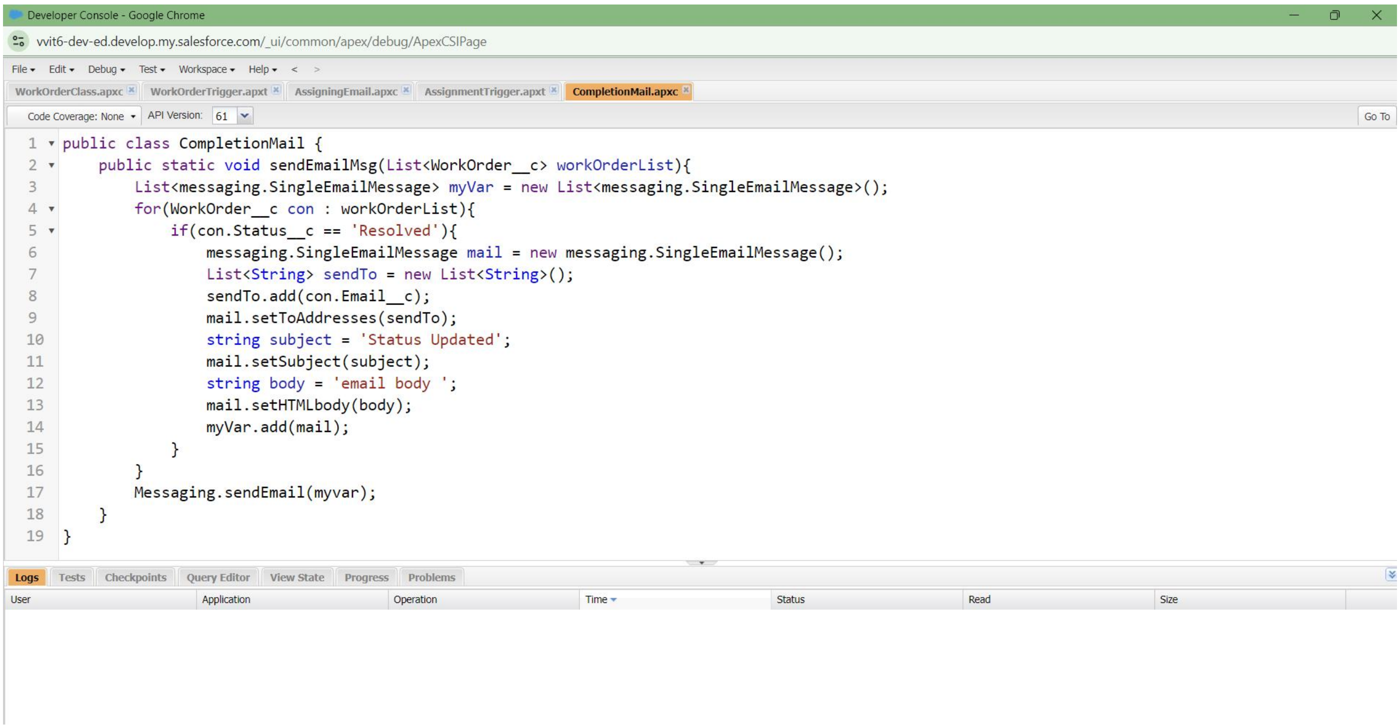
5. Click ok.

6. Now write the code logic here

7. **Source Code:**

```
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);
    }
}
```

8. Save the code.(click on file --> Save)



Code Coverage: None API Version: 61

```
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);
    }
}
```

7.6 Create an Apex Trigger

1. Click on the file --> Open.

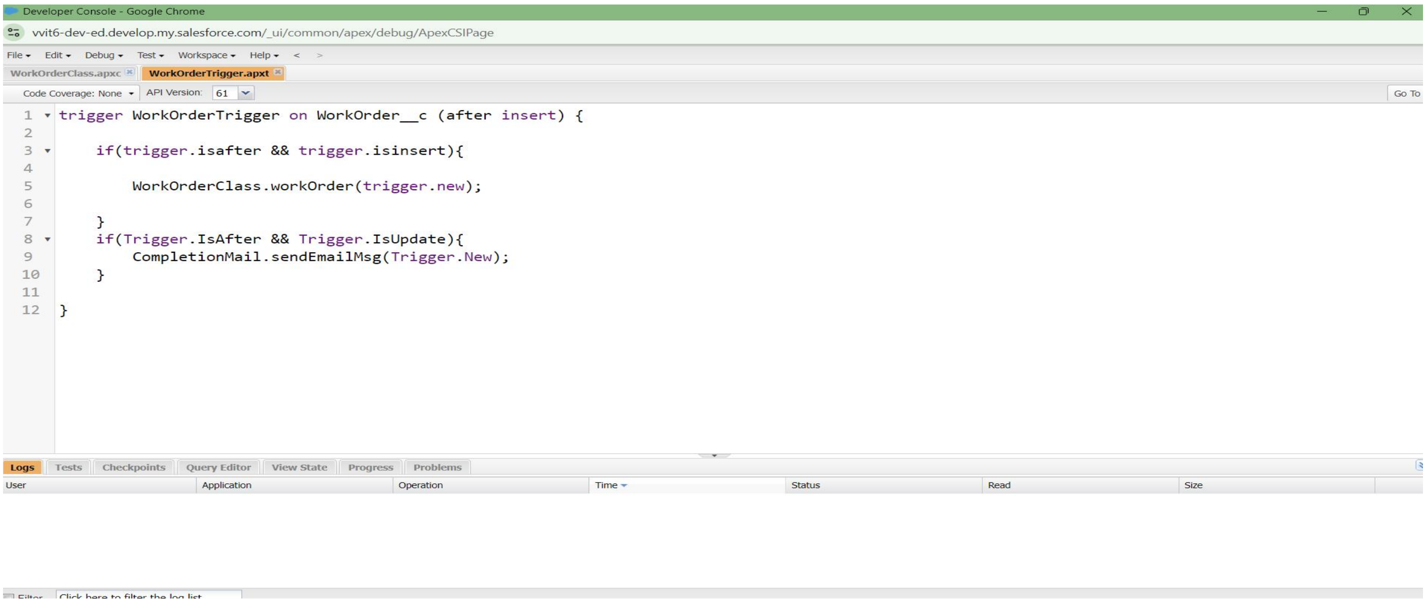
2. A pop up window opens click on Triggers, then select “WorkOrderTrigger” and click on
“Open”

3. Now write the code logic here.

4. WorkOrderClass.workOrder(trigger.new);

```
    }
    if(Trigger.IsAfter && Trigger.IsUpdate){
        CompletionMail.sendEmailMsg(Trigger.New);
    }
}
```

5. Save the code.(click on file --> Save)


Developer Console - Google Chrome
vvit6-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
WorkOrderClass.apxc
WorkOrderTrigger.apxt
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

Code Coverage: None API Version: 61

```
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);
    }

}
```

7.7 Create an Asynchronous Apex Class

Create an Apex Class to Delete all the WorkOrder records which meets the following criteriaL

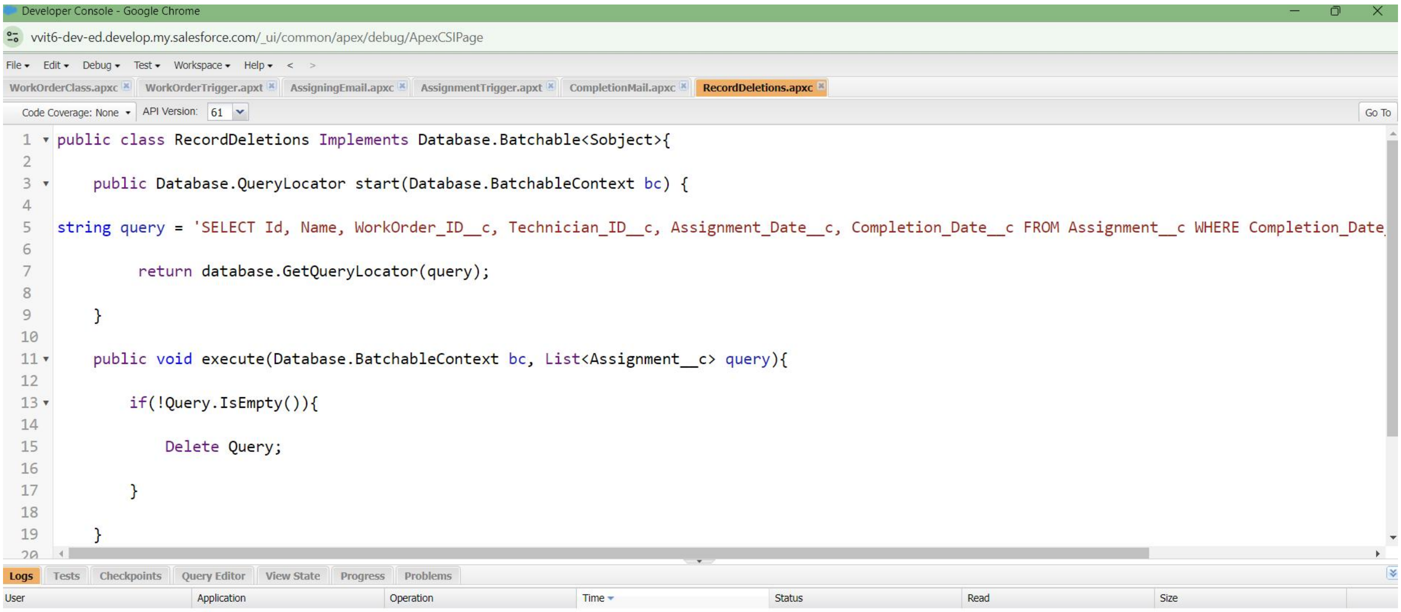
1. Completed date should be more than 30 days.
2. Status should be ‘Resolved’.

Create an Apex Class

1. Go to Setup --> Click on the gear icon --> Select Developer Console.
2. Then we can see the Developer console. Click on the developer console and you will navigate to a new console window.
3. To create a new Apex Class follow the below steps:
Click on the file --> New --> Apex Class.
4. Give the Apex Class name as “RecordDeletion”.
5. Click ok.
6. Now write the code logic here

```
 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){
    }
      }
```

7. Save the code.(click on file --> Save)



```
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

         return database.GetQueryLocator(query);

    }

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

        if(!Query.IsEmpty()){

            Delete Query;

        }

    }
```

7.8 Create an Apex Schedule Class

1. Go to Setup --> Click on the gear icon --> Select Developer Console.

2. Then we can see the Developer console. Click on the developer console and you will navigate to a new console window.

3. To create a new Apex Class follow the below steps:

Click on the file --> New --> Apex Class.

4. Give the Apex Class name as "ScheduleClass" .

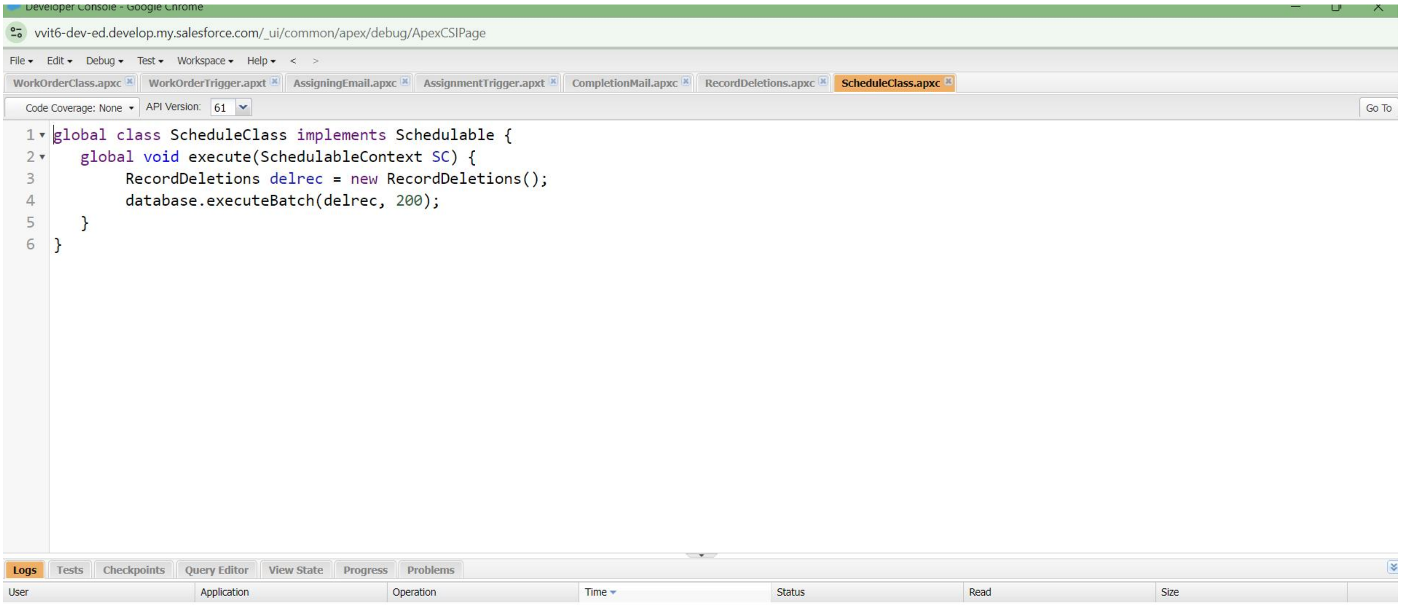
5. Click ok.

6. Now write the code logic here

Source Code:

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

7. Save the code.(click on file ? Save)

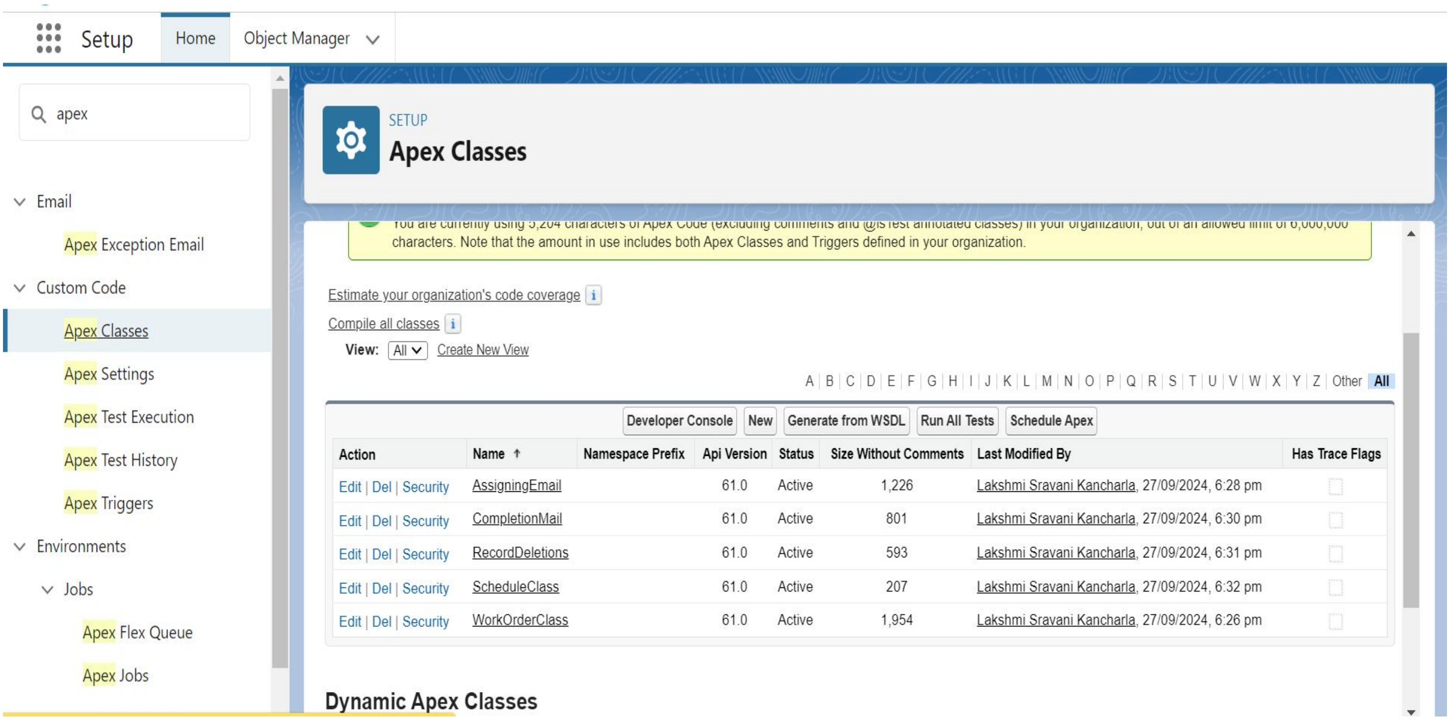


```
global class ScheduleClass implements Schedulable {
   global void execute(SchedulableContext SC) {
        RecordDeletions delrec = new RecordDeletions();
        database.executeBatch(delrec, 200);
   }
}
```

7.9 Create a Schedule Apex

Schedule the Apex class:

1. From the Setup page search for “Apex Classes” in quick search.
2. Click on “Schedule Apex” as shown below.
3. Click on Schedule Apex and enter the Job name.
4. Job Name : DeleteAssignmentSchedule
5. Apex Class : ScheduleClass (from clicking on lookup icon)
6. Frequency : Monthly
7. Preferred Start Time : Select any time
8. Click Save.



You are currently using 5,204 characters of Apex Code (excluding comments and @isTest annotated classes) in your organization, out of an allowed limit of 6,000,000 characters. Note that the amount in use includes both Apex Classes and Triggers defined in your organization.

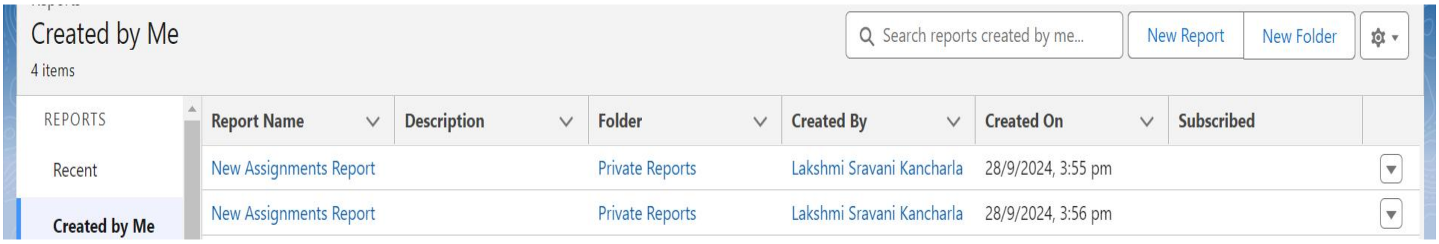
Action	Name ↑	Namespace Prefix	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
Edit | Del | Security	AssigningEmail		61.0	Active	1,226	Lakshmi Sravani Kancharla, 27/09/2024, 6:28 pm	□
Edit | Del | Security	CompletionMail		61.0	Active	801	Lakshmi Sravani Kancharla, 27/09/2024, 6:30 pm	□
Edit | Del | Security	RecordDeletions		61.0	Active	593	Lakshmi Sravani Kancharla, 27/09/2024, 6:31 pm	□
Edit | Del | Security	ScheduleClass		61.0	Active	207	Lakshmi Sravani Kancharla, 27/09/2024, 6:32 pm	□
Edit | Del | Security	WorkOrderClass		61.0	Active	1,954	Lakshmi Sravani Kancharla, 27/09/2024, 6:26 pm	□

Task 8 :

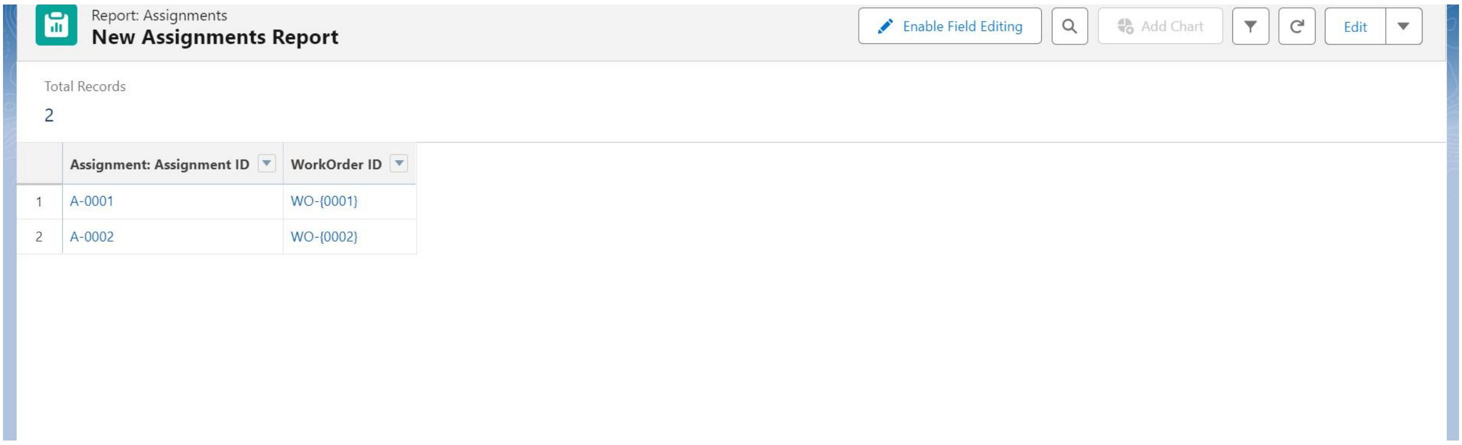
8.1 Report

1. Go to the app --> click on the reports tab
2. Click New Report.
3. Select report type from category or from report type panel or from search panel --> click on start report.
4. Customize your report
5. Add fields from left pane as shown below
6. Grouped by workorder ID
7. Save or run it.

Note: Reports may get varied from the above pictures as the data might be different

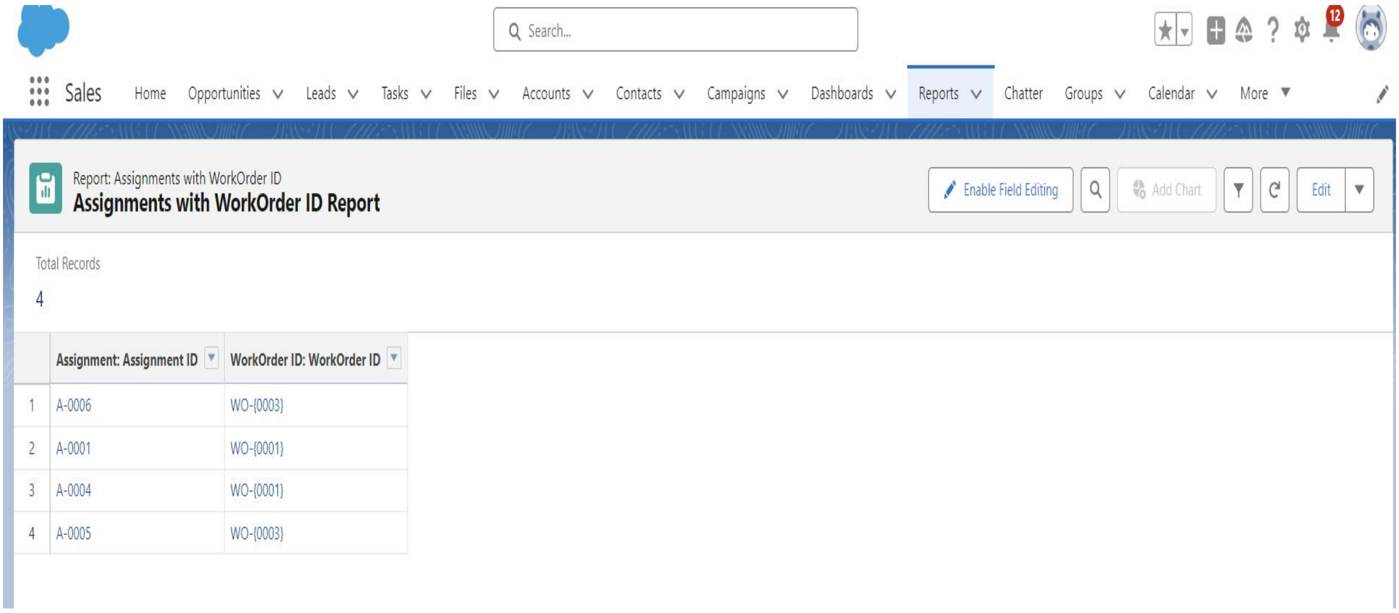


Report Name	Description	Folder	Created By	Created On	Subscribed
New Assignments Report		Private Reports	Lakshmi Sravani Kancharla	28/9/2024, 3:55 pm	
New Assignments Report		Private Reports	Lakshmi Sravani Kancharla	28/9/2024, 3:56 pm	



Total Records
2

	Assignment: Assignment ID	WorkOrder ID
1	A-0001	WO-{0001}
2	A-0002	WO-{0002}

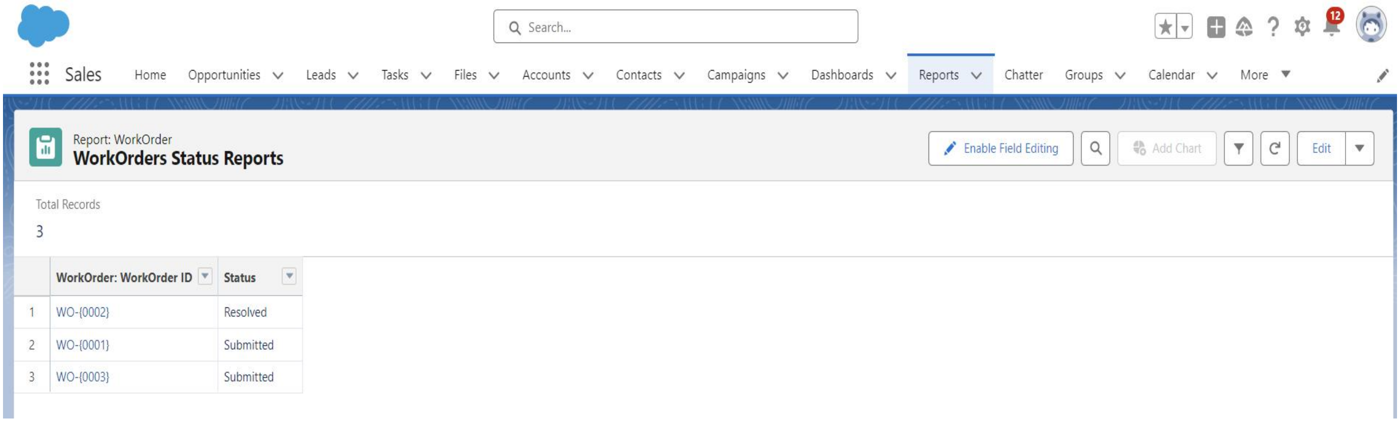


Total Records
4

	Assignment: Assignment ID	WorkOrder ID: WorkOrder ID
1	A-0006	WO-{0003}
2	A-0001	WO-{0001}
3	A-0004	WO-{0001}
4	A-0005	WO-{0003}

8.2 Create Reports

1. Create a report with report type: “WorkOrders Status Reports” .

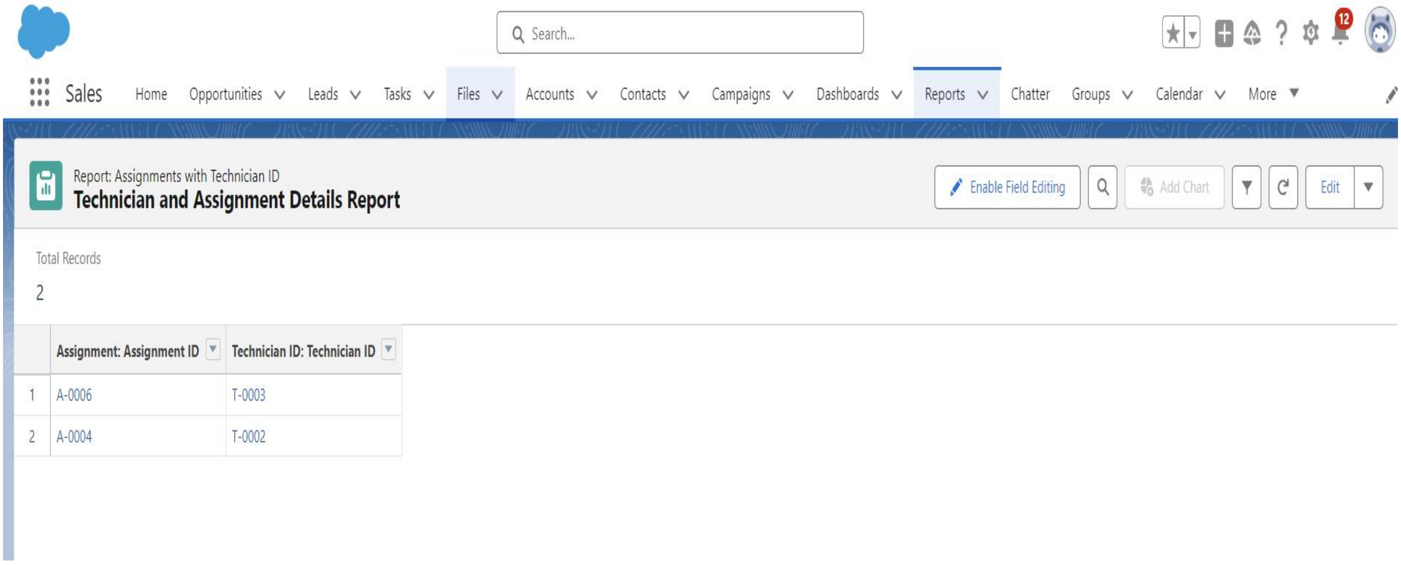


Report: WorkOrder
WorkOrders Status Reports

Total Records
3

	WorkOrder: WorkOrder ID	Status
1	WO-{0002}	Resolved
2	WO-{0001}	Submitted
3	WO-{0003}	Submitted

2. Create a report with report type: “Technician and Assignment Details Reports”



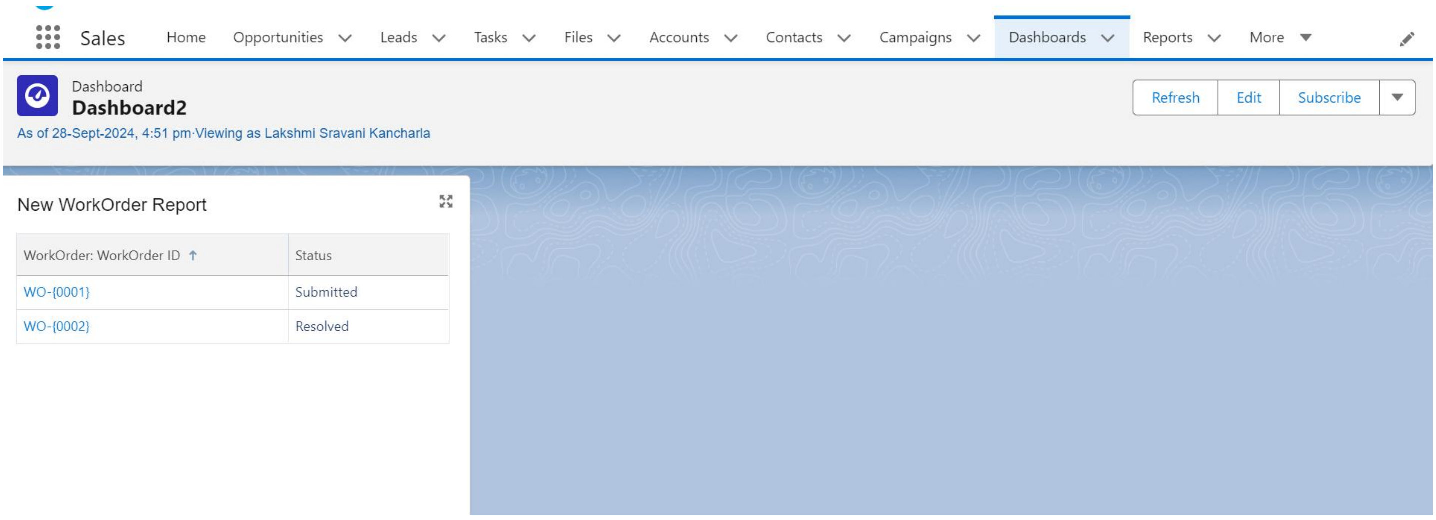
Report: Assignments with Technician ID
Technician and Assignment Details Report

Total Records
2

	Assignment: Assignment ID	Technician ID: Technician ID
1	A-0006	T-0003
2	A-0004	T-0002

8.3 Dashboard

1. Go to the app --> click on the Dashboards tabs.
2. Give a Name and click on Create.
3. Select add component.
4. Select a Report which we have created in the previous activities and click on select.
5. Click Add then click on Save and then click on Done.



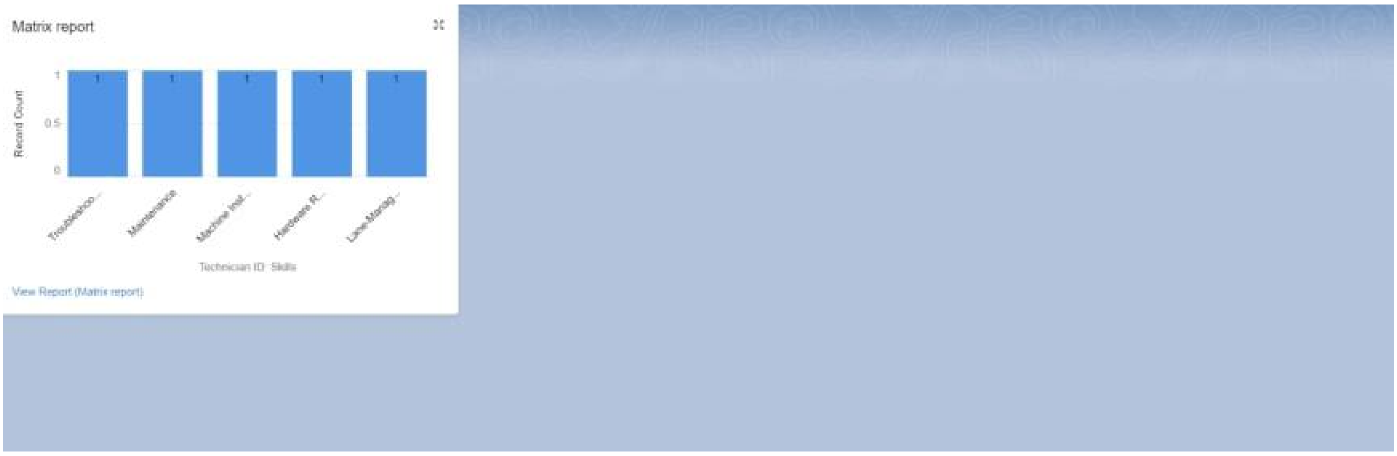
Dashboard
Dashboard2
As of 28-Sept-2024, 4:51 pm·Viewing as Lakshmi Sravani Kancharla

New WorkOrder Report

WorkOrder: WorkOrder ID ↑	Status
WO-{0001}	Submitted
WO-{0002}	Resolved

8.4 Create Dashboards

Create another Dashboard as we discussed in activity 3 which shows the details of completed Work order status in a vertical bar graph.



Thank
you