FIELD SERVICE WORKORDER OPTIMIZATION

By

MUVVA VIDYA

21BQ1A05F3@vvit.net

ABSTRACT

Objective: The Field Service Work Order Optimization System helps a company manage installations and repair tasks more efficiently. This system uses a powerful database to find the right technician for each job. It considers factors like location, availability, and skill set. The main goal is to make sure that every technician gets assigned to the most suitable work order. The system uses an algorithm to prioritize tasks and assigns them accordingly. It helps save time, reduces costs, and improves customer satisfaction.

The system also has automated communication features. It sends real-time updates to technicians, so they are always informed about their assignments. This reduces confusion and ensures smoother operations. There are built-in analytics tools that offer insights into performance, which helps in making better decisions for future tasks. This leads to continuous improvement of services and overall efficiency.

Key Technologies:

1.**Salesforce Field Service:** The core platform for managing field services. It offers tools for scheduling, dispatching, and real-time communication.

2.**Artificial Intelligence (AI) & Machine Learning (ML):** AI and ML algorithms are integrated into Salesforce. They help predict service demands, schedule tasks, and match the right technician with the right job.

3.**Predictive Analytics:** Using Salesforce's analytics capabilities, this feature forecasts future service needs. It analyzes past data to address potential issues before they arise.

4.**Internet of Things (IoT):** IoT devices provide real-time information from field equipment. This data is fed into Salesforce for proactive maintenance and quick responses to any issues.

Implementation Phases:

- **Salesforce Field Service Setup:** This phase includes setting up the core system for scheduling, dispatching, and real-time communication.
- **Integrating AI and ML:** AI and ML are used to improve scheduling and task assignments based on real-time data.
- **Predictive Analytics Integration:** This phase focuses on using historical data to predict service needs and plan accordingly.
- **IoT Integration:** Real-time data from field equipment is collected using IoT devices and connected to the Salesforce system for faster maintenance.

Potential Challenges:

- **Data Integration:** Combining data from different sources and legacy systems can be complex.
- **Change Management:** Managing transitions and getting everyone on board with new systems can be challenging.
- **Scalability:** The solution must be able to handle future growth and increased demand.
- **Security and Privacy:** Protecting sensitive customer and company data from security threats is critical.

Measurable Outcomes:

- **Efficiency Metrics:** Improvement in job completion time and resource utilization.
- **Customer Satisfaction:** Enhanced customer feedback due to timely service and better communication.
- **Operational Excellence:** Reduced errors, quicker response times, and better overall operations.

Functional Requirements:

- **Work Order Management:** The system will track and manage work orders.
- **Scheduling and Dispatching:** Assign jobs to the best-suited technicians.
- **Resource Management:** Manage the availability and skills of technicians.
- **Mobile Access:** Technicians can access the system on mobile devices.
- **Customer Communication:** Real-time updates and notifications to customers.
- **Analytics and Reporting**: Generate reports for better decision-making.
- **Integration:** Connect with other systems for seamless operations.
- **User Management and Security:** Control user access and protect data.
- **Maintenance and Support:** Provide ongoing support and system maintenance.

By fulfilling these requirements, the system will help streamline operations, enhance efficiency, and improve customer satisfaction, leading to the overall success of field service operations.

INDEX PAGE

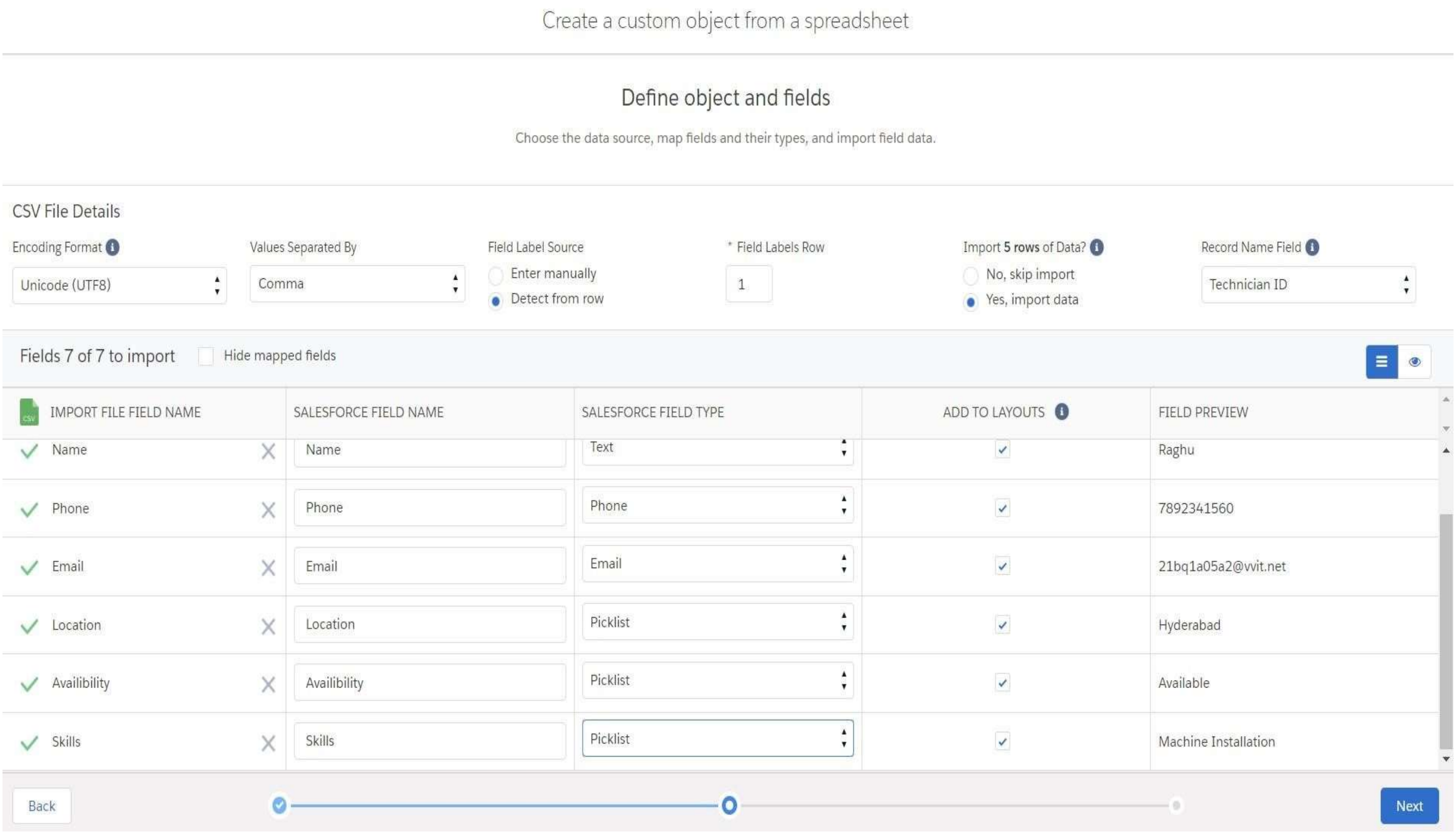
INTRODUCTION

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:

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

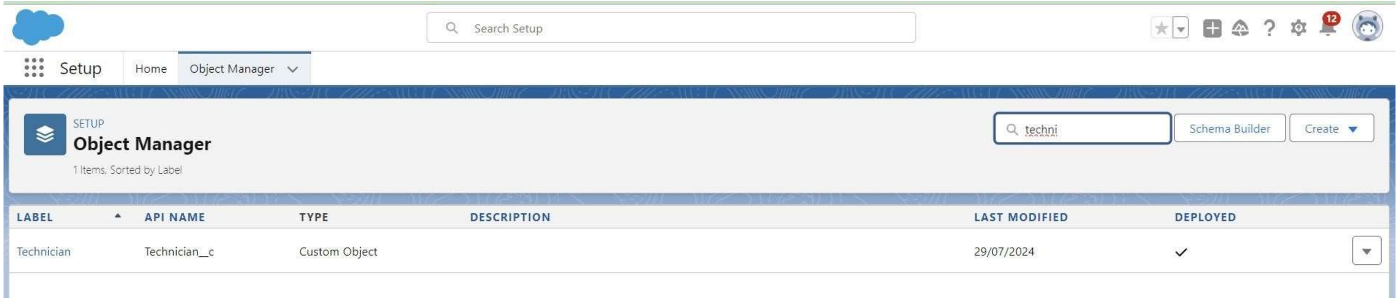
Import 5 rows of Data? ○ No, skip import ● Yes, import data

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
✓ Name	Name	Text	✓	Raghu
✓ Phone	Phone	Phone	✓	7892341560
✓ Email	Email	Email	✓	21bq1a05a2@vvit.net
✓ Location	Location	Picklist	✓	Hyderabad
✓ Availibility	Availibility	Picklist	✓	Available
✓ Skills	Skills	Picklist	✓	Machine Installation

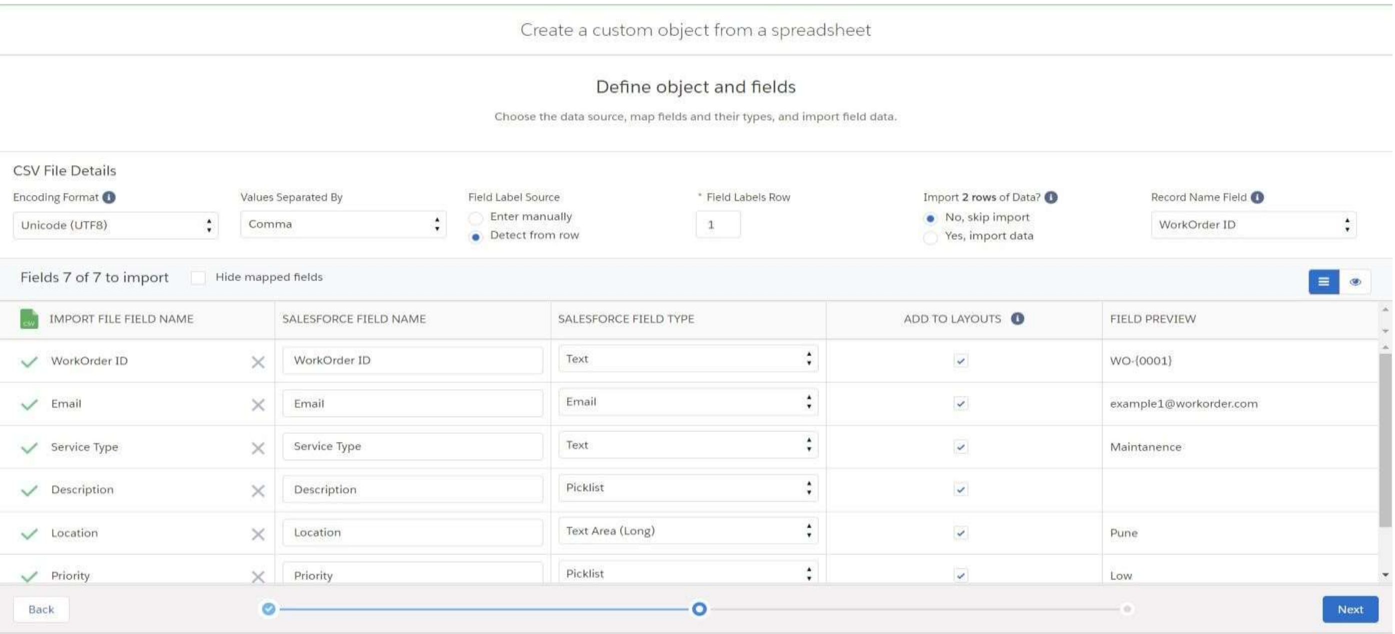
After creating technician details, the Quick box looks like the below



LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Technician	Technician__c	Custom Object		29/07/2024	✓

Create Work Order 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.
Back
Next

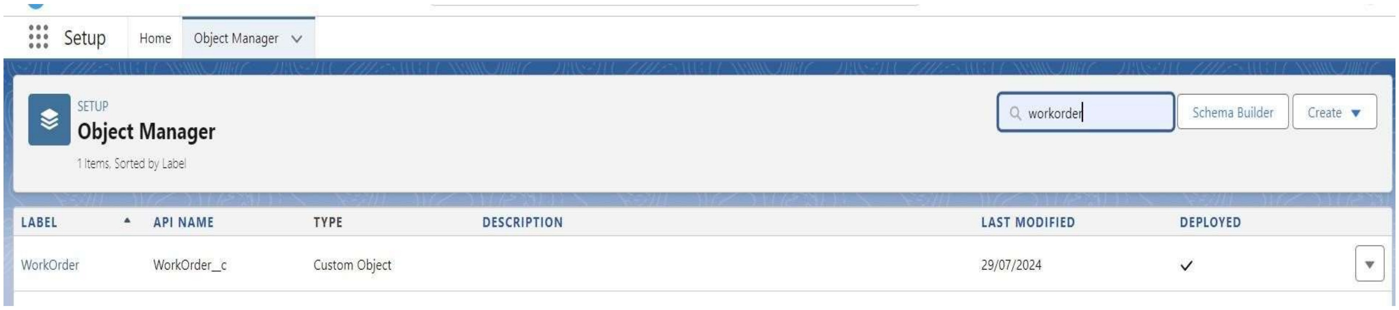
CSV File Details

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: WorkOrder 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
✓ WorkOrder ID	WorkOrder ID	Text	☑	WO-{0001}
✓ Email	Email	Email	☑	example1@workorder.com
✓ Service Type	Service Type	Text	☑	Maintanence
✓ Description	Description	Picklist	☑	
✓ Location	Location	Text Area (Long)	☑	Pune
✓ Priority	Priority	Picklist	☑	Low

After creating the Work Order Custom object it looks like the below

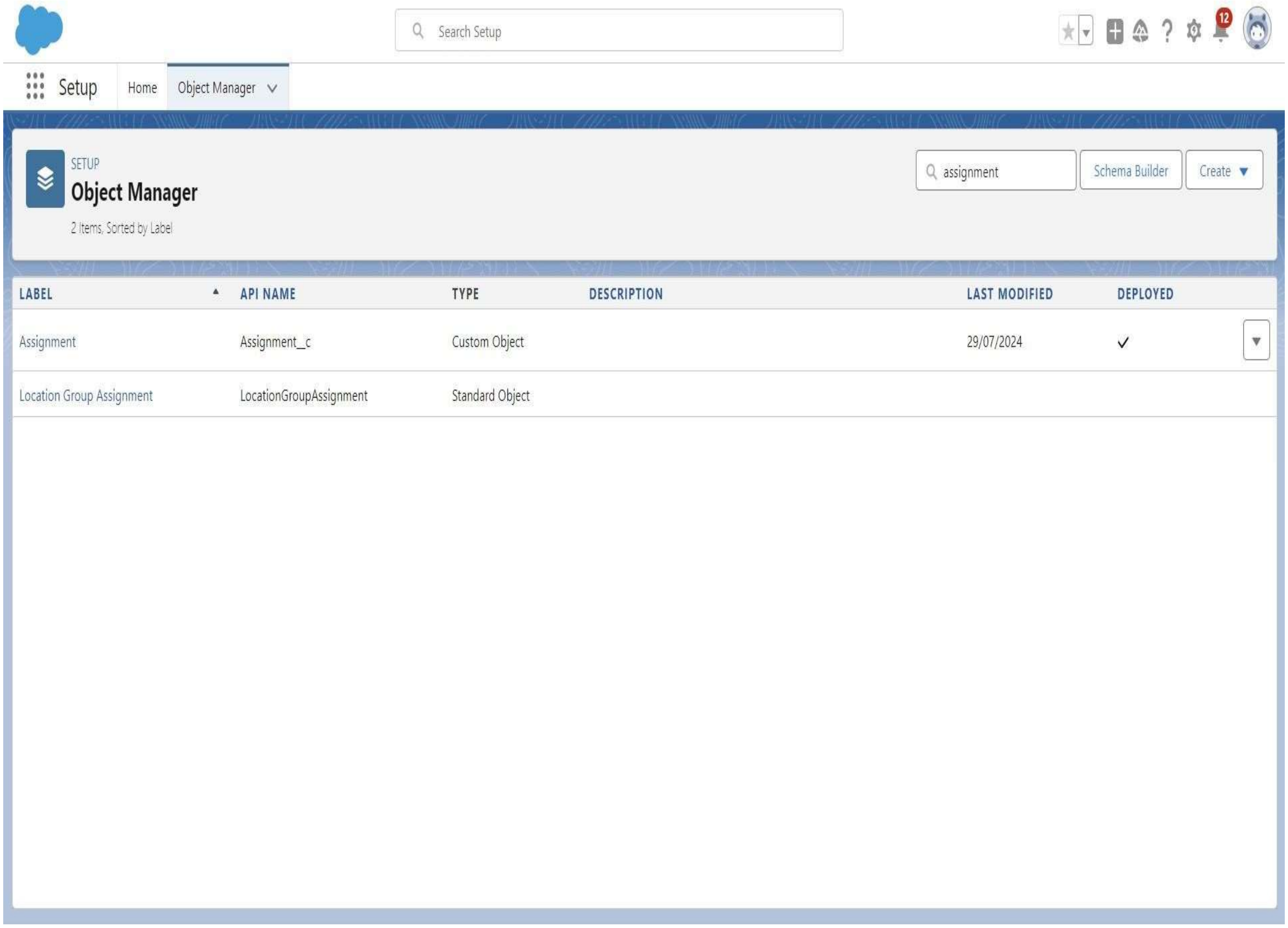


LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
WorkOrder	WorkOrder__c	Custom Object		29/07/2024	✓

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



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

Task 2:

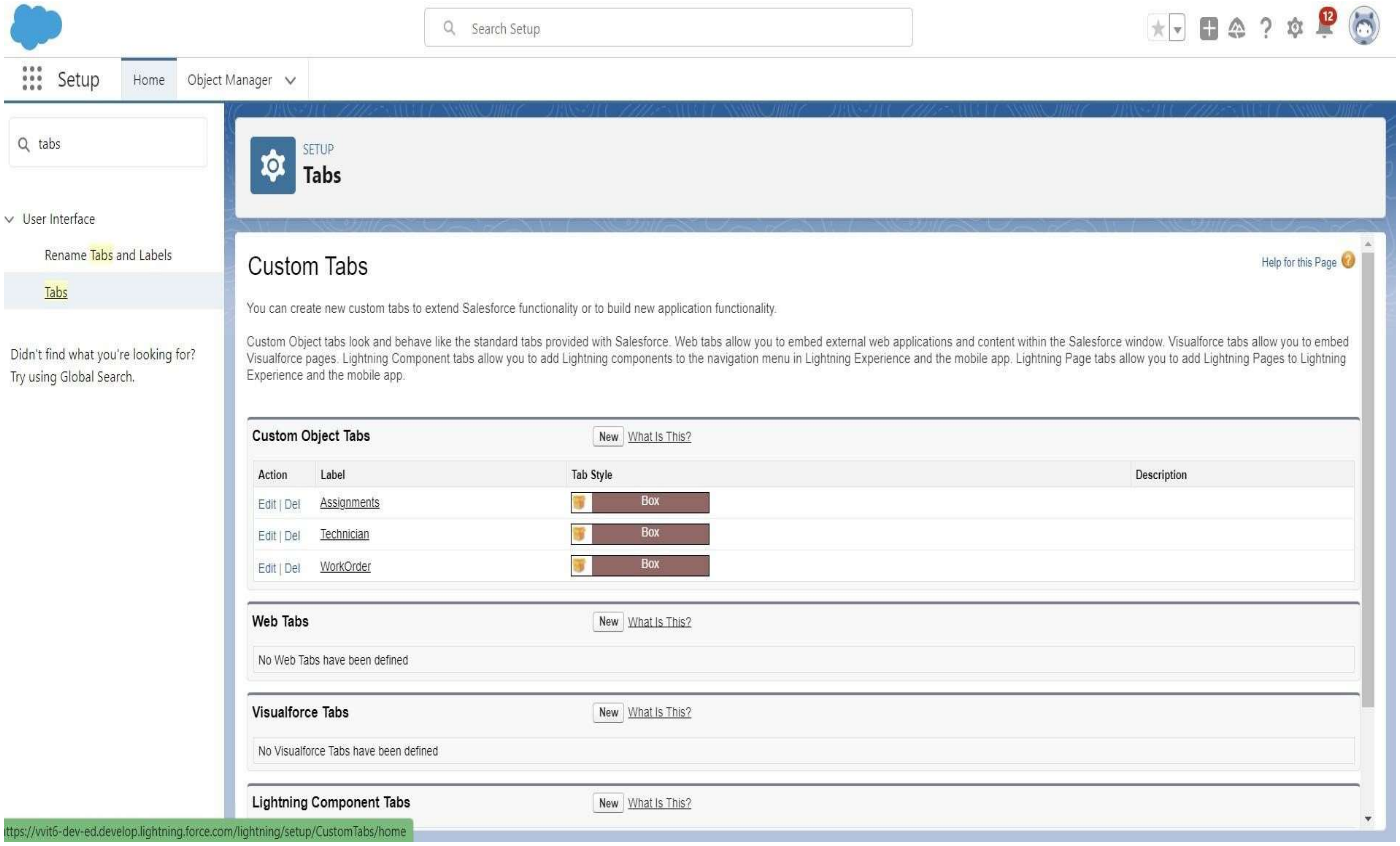
Creating a Custom Tab

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 Work Order & 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:


Search Setup
Setup
Home
Object Manager
tabs
User Interface
Rename Tabs and Labels
Tabs
Didn't find what you're looking for? Try using Global Search.
SETUP
Tabs
Help for this Page
ttps://vvit6-dev-ed.develop.lightning.force.com/lightning/setup/CustomTabs/home

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?

Task 3 :

Create a Lightning App

To create a lightning app page:

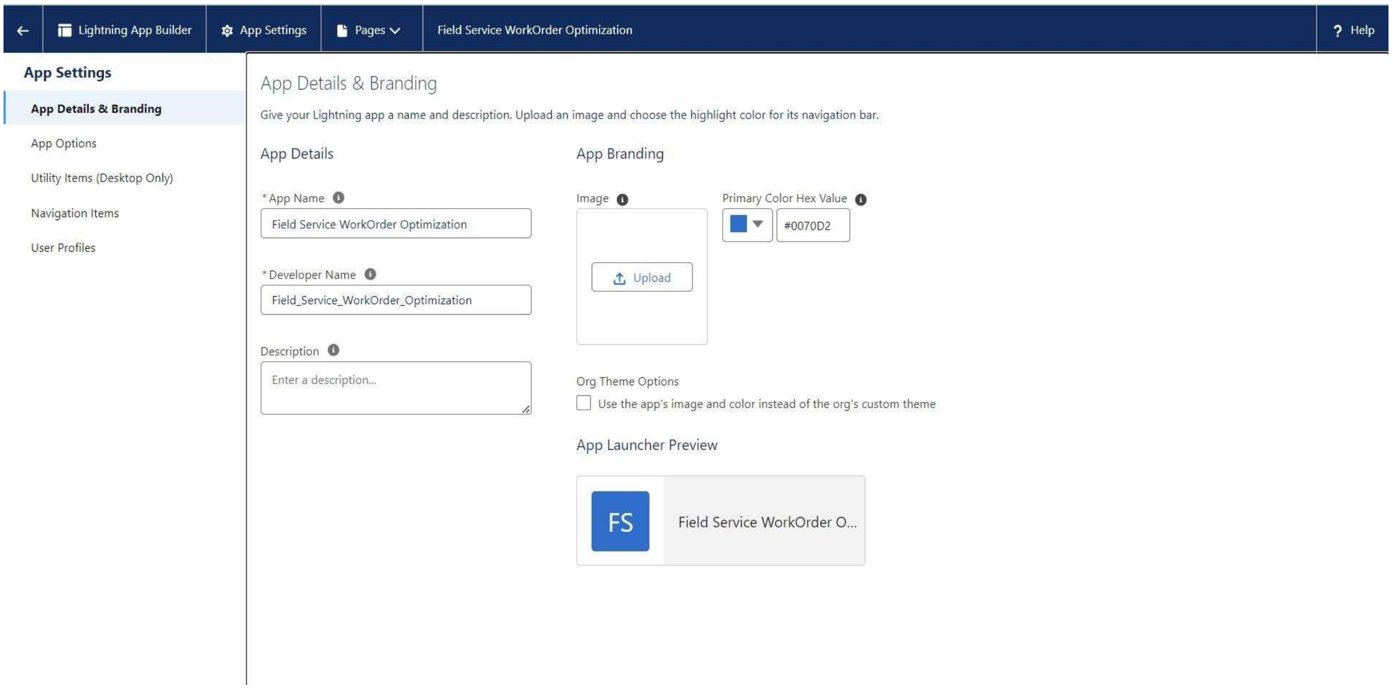
1. Go to the setup page --> search “app manager” in quick find --> select “app manager” --> click on New lightning App.
2. Fill the app name in app details and branding as follow

   App Name: Field Service Work Order 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

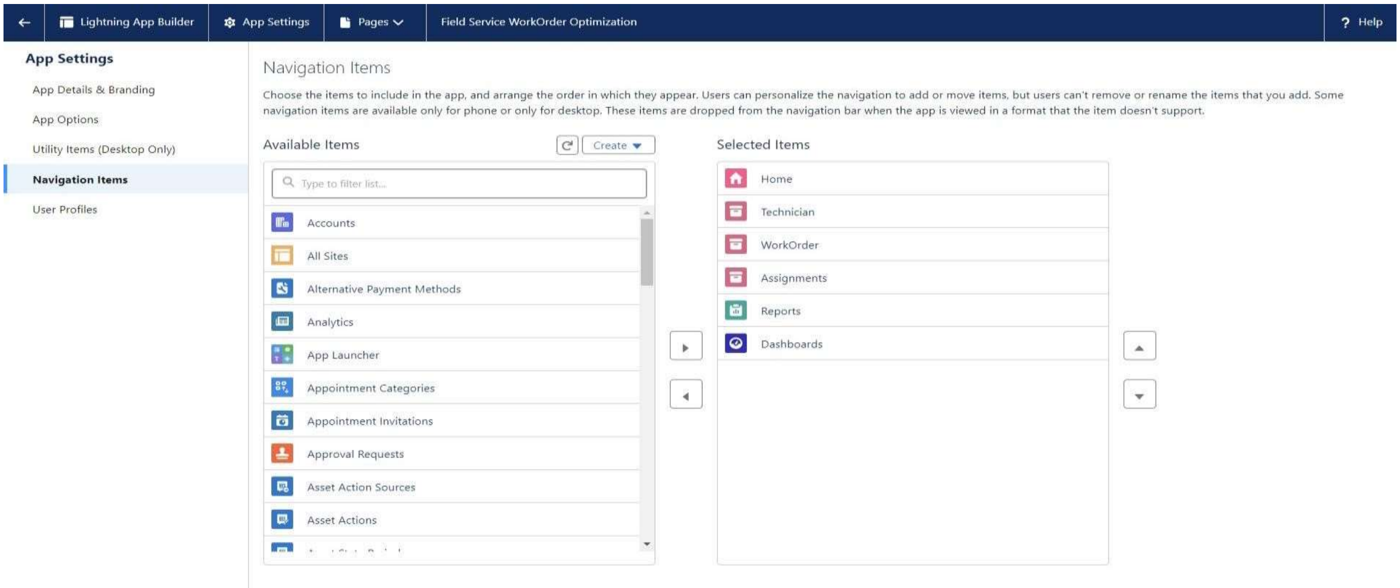
Org Theme Options

□ 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

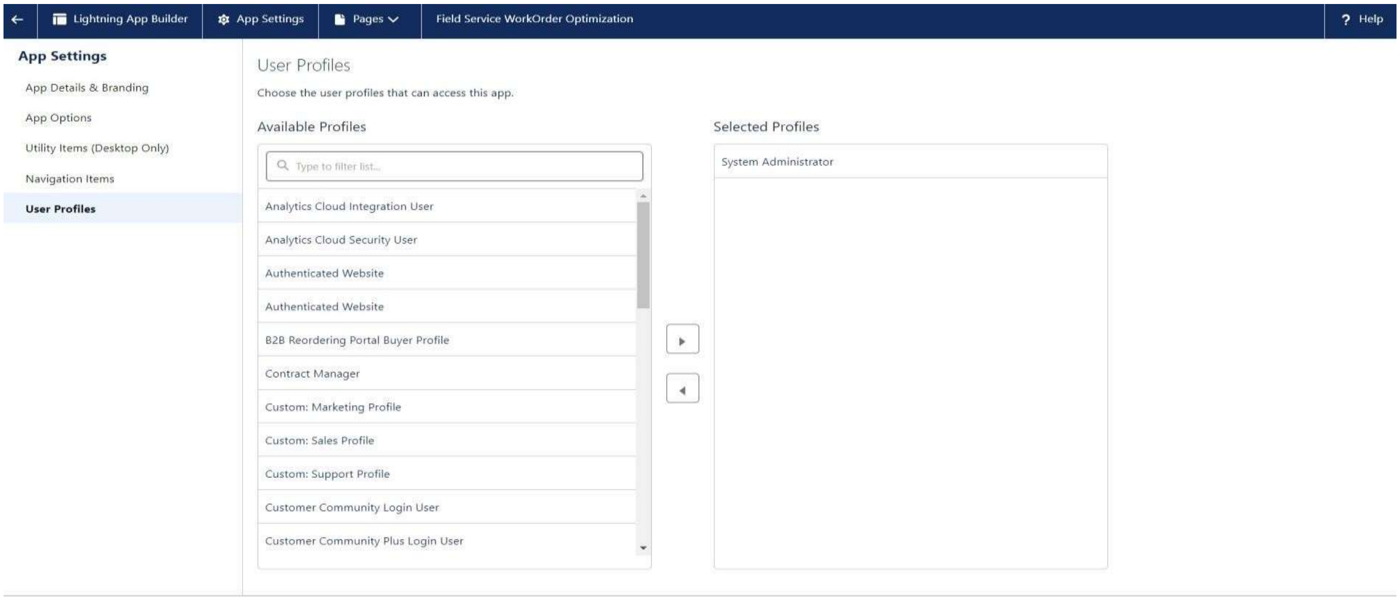
Selected Items: Home, Technician, WorkOrder, Assignments, Reports, Dashboards

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

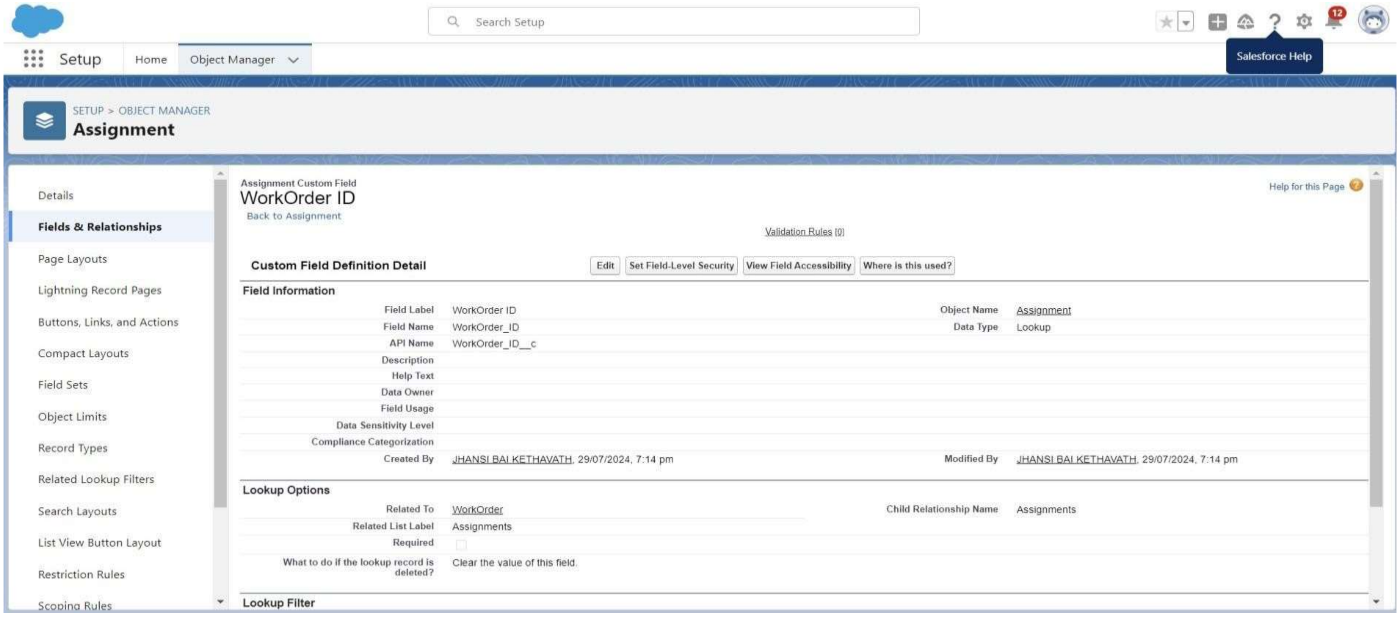
- System Administrator

This is the output after completion of following the aboveprocedure.

Task 4:

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

Back to Assignment

Custom Field Definition Detail

Field Information

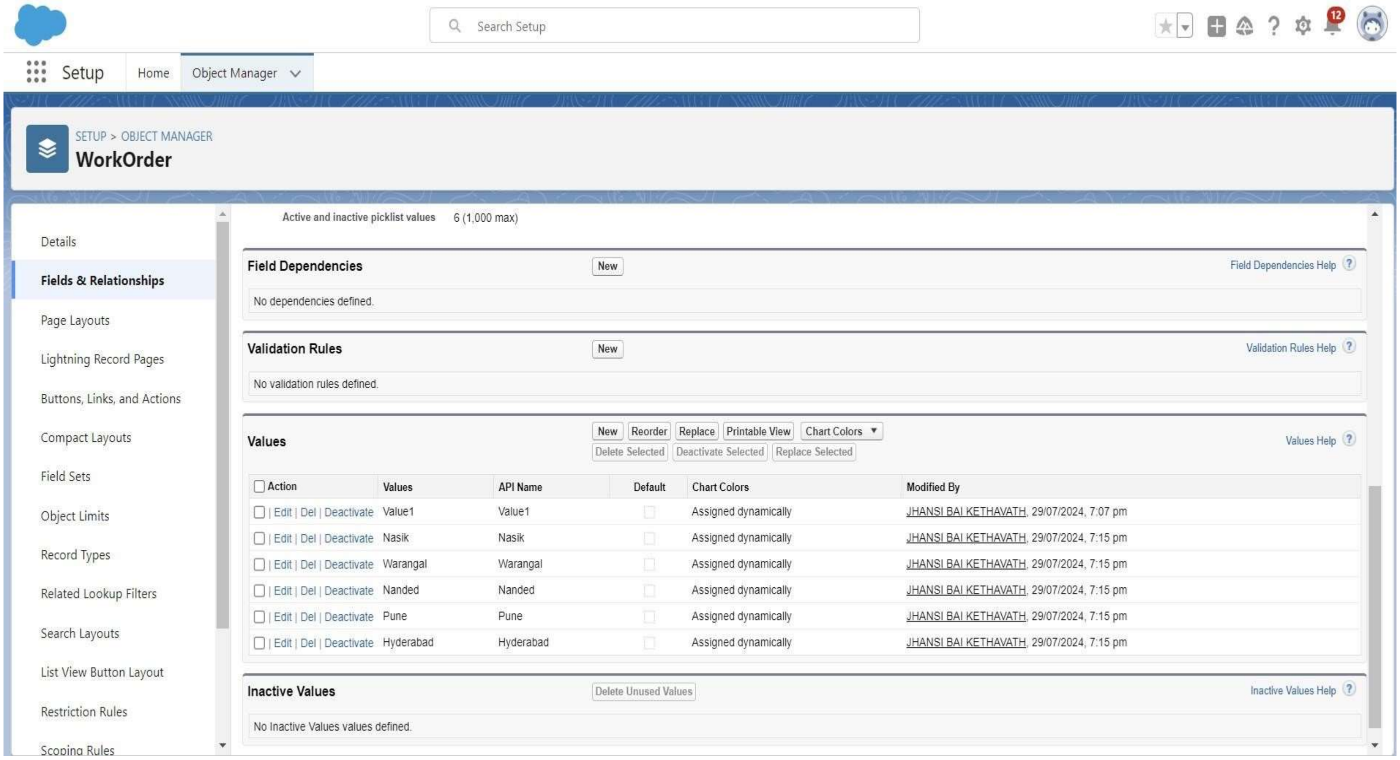
Field Label	WorkOrder ID	Object Name	Assignment
Field Name	WorkOrder_ID	Data Type	Lookup
API Name	WorkOrder_ID__c		
Description			
Help Text			
Data Owner			
Field Usage			
Data Sensitivity Level			
Compliance Categorization			
Created By	JHANSI BAI KETHAVATH, 29/07/2024, 7:14 pm	Modified By	JHANSI BAI KETHAVATH, 29/07/2024, 7:14 pm

Lookup Options

Related To	WorkOrder	Child Relationship Name	Assignments
Related List Label	Assignments		
Required	□		
What to do if the lookup record is deleted?	Clear the value of this field.		

Lookup Filter

Manage your picklist values



Active and inactive picklist values 6 (1,000 max)

Field Dependencies

No dependencies defined.

Validation Rules

No validation rules defined.

Values

□ Action	Values	API Name	Default	Chart Colors	Modified By
□ | Edit | Del | Deactivate	Value1	Value1	□	Assigned dynamically	JHANSI BAI KETHAVATH, 29/07/2024, 7:07 pm
□ | Edit | Del | Deactivate	Nasik	Nasik	□	Assigned dynamically	JHANSI BAI KETHAVATH, 29/07/2024, 7:15 pm
□ | Edit | Del | Deactivate	Warangal	Warangal	□	Assigned dynamically	JHANSI BAI KETHAVATH, 29/07/2024, 7:15 pm
□ | Edit | Del | Deactivate	Nanded	Nanded	□	Assigned dynamically	JHANSI BAI KETHAVATH, 29/07/2024, 7:15 pm
□ | Edit | Del | Deactivate	Pune	Pune	□	Assigned dynamically	JHANSI BAI KETHAVATH, 29/07/2024, 7:15 pm
□ | Edit | Del | Deactivate	Hyderabad	Hyderabad	□	Assigned dynamically	JHANSI BAI KETHAVATH, 29/07/2024, 7:15 pm

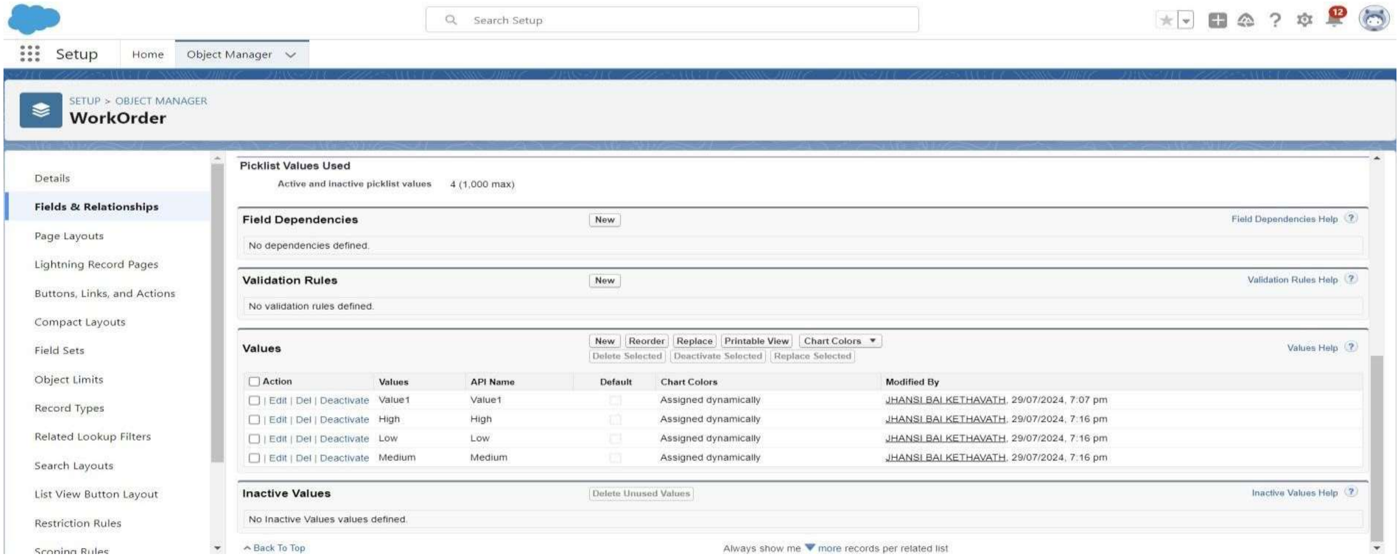
Inactive Values

No Inactive Values values defined.

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



Picklist Values Used

Active and inactive picklist values 4 (1,000 max)

Field Dependencies

No dependencies defined.

Validation Rules

No validation rules defined.

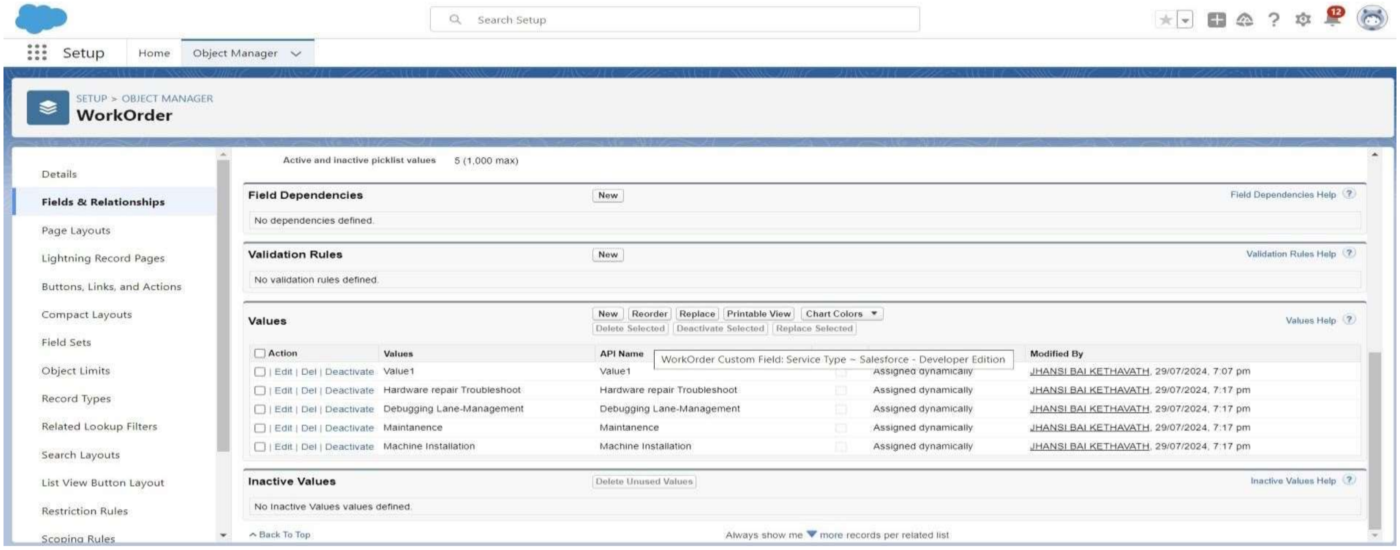
Values

Action	Values	API Name	Default	Chart Colors	Modified By
□ | Edit | Del | Deactivate	Value1	Value1	□	Assigned dynamically	JHANSI BAI KETHAVATH, 29/07/2024, 7:07 pm
□ | Edit | Del | Deactivate	High	High	□	Assigned dynamically	JHANSI BAI KETHAVATH, 29/07/2024, 7:16 pm
□ | Edit | Del | Deactivate	Low	Low	□	Assigned dynamically	JHANSI BAI KETHAVATH, 29/07/2024, 7:16 pm
□ | Edit | Del | Deactivate	Medium	Medium	□	Assigned dynamically	JHANSI BAI KETHAVATH, 29/07/2024, 7:16 pm

Inactive Values

No Inactive Values values defined.

Always show me more records per related list



Active and inactive picklist values 5 (1,000 max)

Field Dependencies

No dependencies defined.

Validation Rules

No validation rules defined.

Values

WorkOrder Custom Field: Service Type ~ Salesforce - Developer Edition

Action	Values	API Name			Modified By
□ | Edit | Del | Deactivate	Value1	Value1	□	Assigned dynamically	JHANSI BAI KETHAVATH, 29/07/2024, 7:07 pm
□ | Edit | Del | Deactivate	Hardware repair Troubleshoot	Hardware repair Troubleshoot	□	Assigned dynamically	JHANSI BAI KETHAVATH, 29/07/2024, 7:17 pm
□ | Edit | Del | Deactivate	Debugging Lane-Management	Debugging Lane-Management	□	Assigned dynamically	JHANSI BAI KETHAVATH, 29/07/2024, 7:17 pm
□ | Edit | Del | Deactivate	Maintanence	Maintanence	□	Assigned dynamically	JHANSI BAI KETHAVATH, 29/07/2024, 7:17 pm
□ | Edit | Del | Deactivate	Machine Installation	Machine Installation	□	Assigned dynamically	JHANSI BAI KETHAVATH, 29/07/2024, 7:17 pm

Inactive Values

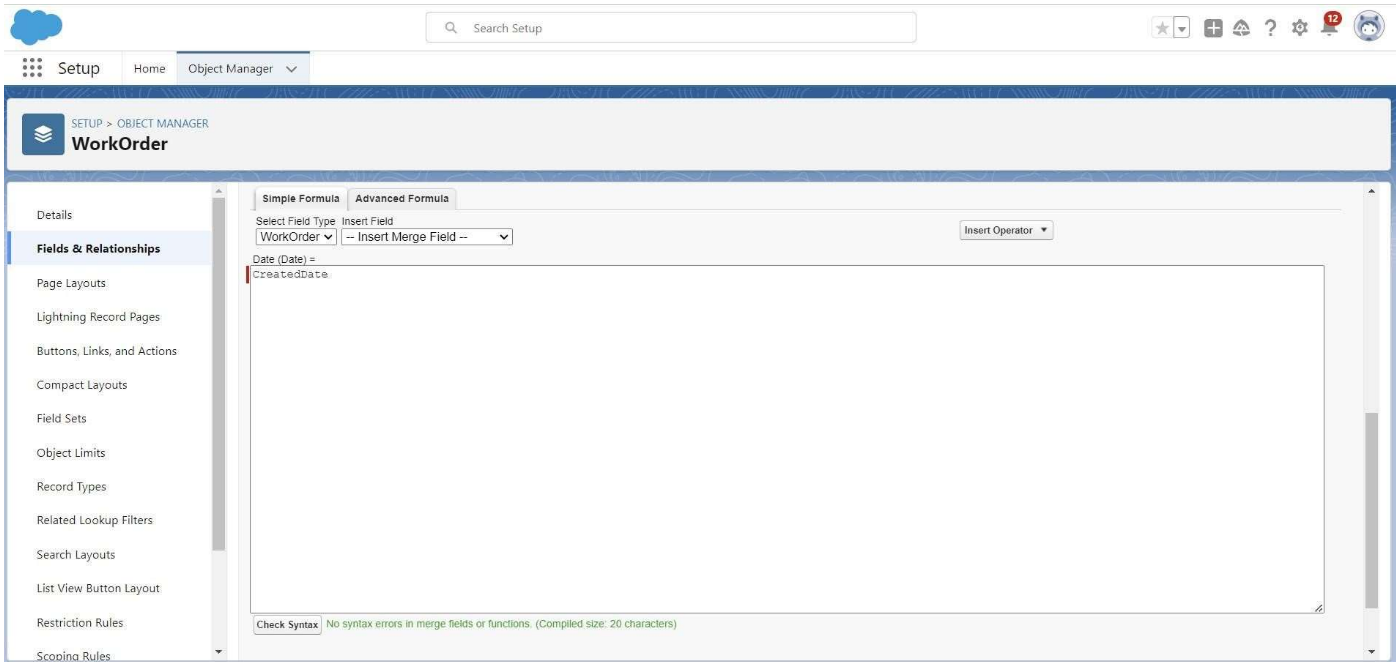
No Inactive Values values defined.

Always show me more records per related list

Creating Formula Field in Work Order 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.



Select Field Type: WorkOrder
Insert Field: -- Insert Merge Field --

Date (Date) =

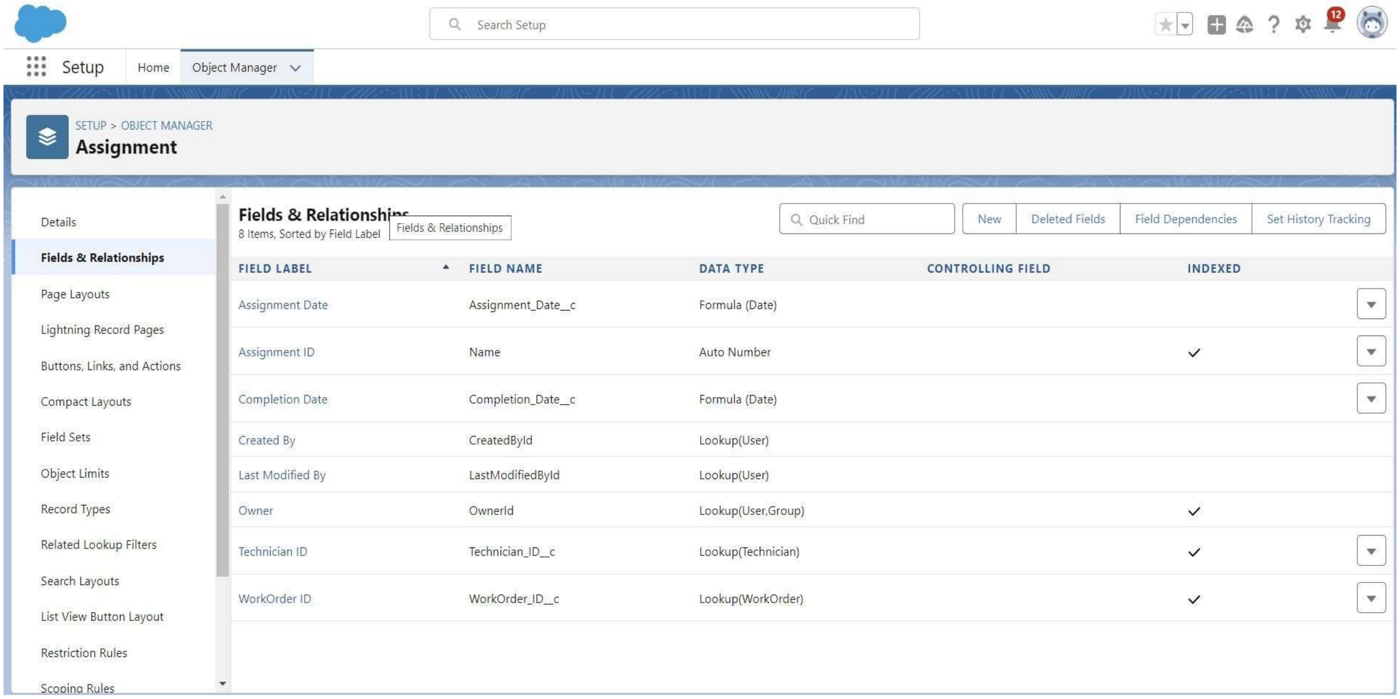
```
CreatedDate
```

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

Creating Remaining fields for the respective objects

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

<table>
<tr><th>SI 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
● Assignment Date
● Completion Date</td><td>Lookup (Technician)
Formula: return type: Date
(WorkOrder_ID__r. Date__c)
Formula: return type: Date
IF (ISPICKVAL (
WorkOrder_ID__r. Status__c ,
'Resolved'),
WorkOrder_ID__r.
LastModifiedDate,
NULL)</td></tr>
</table>
</td></tr>
</table>



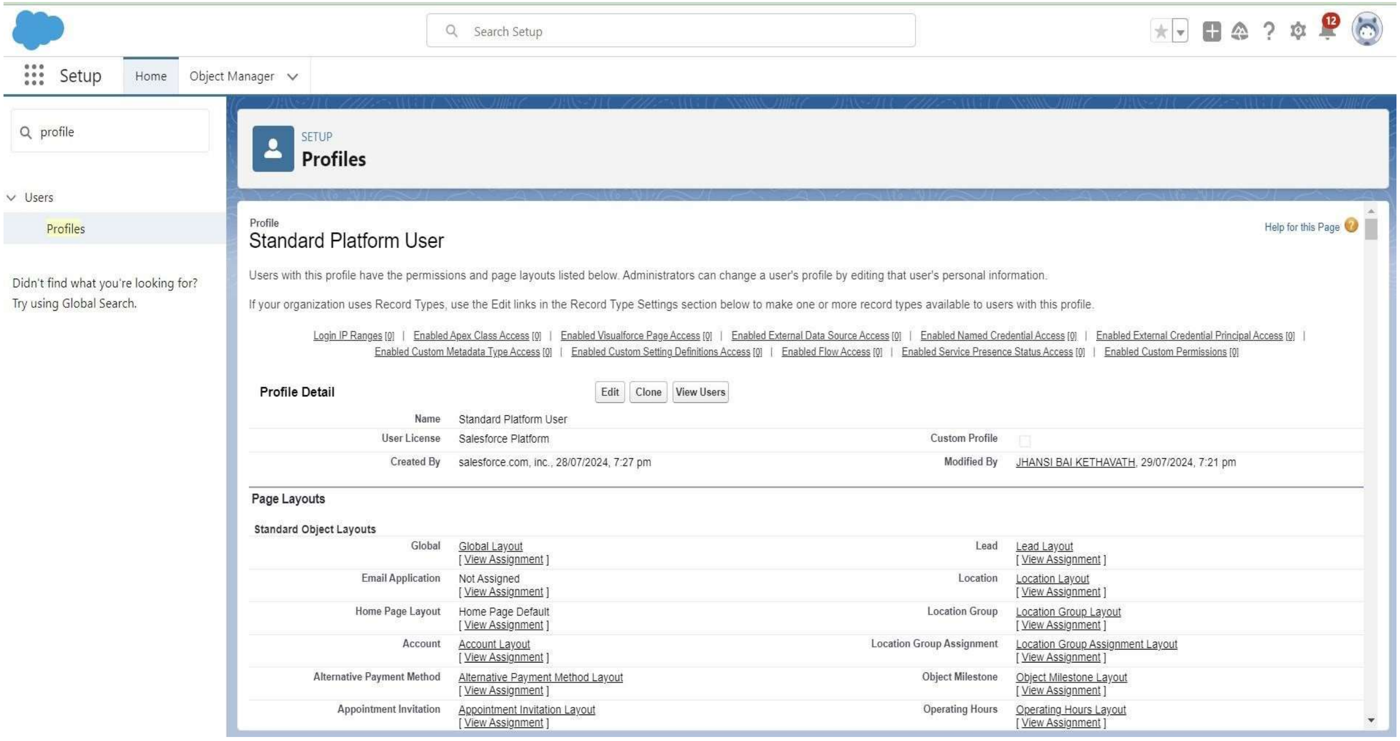
Fields & Relationships

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 Work Order object.
7. Click on Edit, enable the check box for the status field.
8. Click on Save.



Profile

Standard Platform User

Users with this profile have the permissions and page layouts listed below. Administrators can change a user's profile by editing that user's personal information.

If your organization uses Record Types, use the Edit links in the Record Type Settings section below to make one or more record types available to users with this profile.

Login IP Ranges [0] | Enabled Apex Class Access [0] | Enabled Visualforce Page Access [0] | Enabled External Data Source Access [0] | Enabled Named Credential Access [0] | Enabled External Credential Principal Access [0] | Enabled Custom Metadata Type Access [0] | Enabled Custom Setting Definitions Access [0] | Enabled Flow Access [0] | Enabled Service Presence Status Access [0] | Enabled Custom Permissions [0]

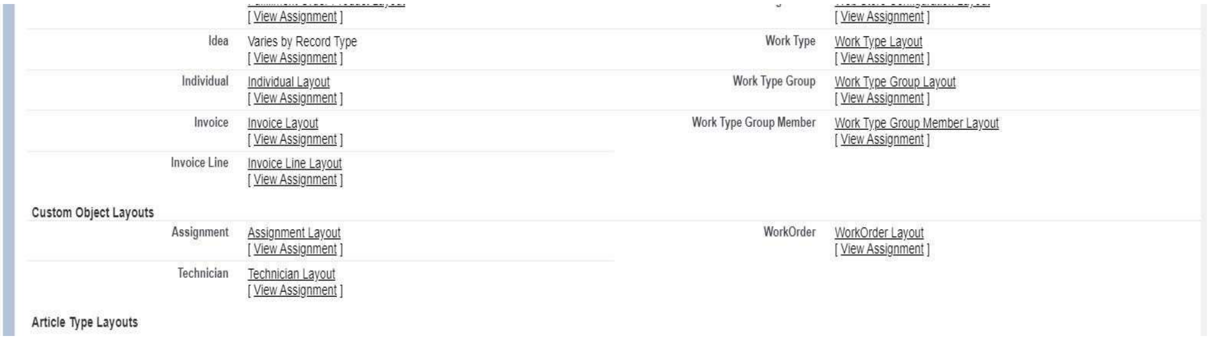
Profile Detail

Name	Standard Platform User		
User License	Salesforce Platform	Custom Profile	□
Created By	salesforce.com, inc., 28/07/2024, 7:27 pm	Modified By	JHANSI BAI KETHAVATH, 29/07/2024, 7:21 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]



	[View Assignment]		[View Assignment]
Idea	Varies by Record Type [View Assignment]	Work Type	Work Type Layout [View Assignment]
Individual	Individual Layout [View Assignment]	Work Type Group	Work Type Group Layout [View Assignment]
Invoice	Invoice Layout [View Assignment]	Work Type Group Member	Work Type Group Member Layout [View Assignment]
Invoice Line	Invoice Line Layout [View Assignment]		

Custom Object Layouts

Assignment	Assignment Layout [View Assignment]	WorkOrder	WorkOrder Layout [View Assignment]
Technician	Technician Layout [View Assignment]		

Article Type Layouts

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.

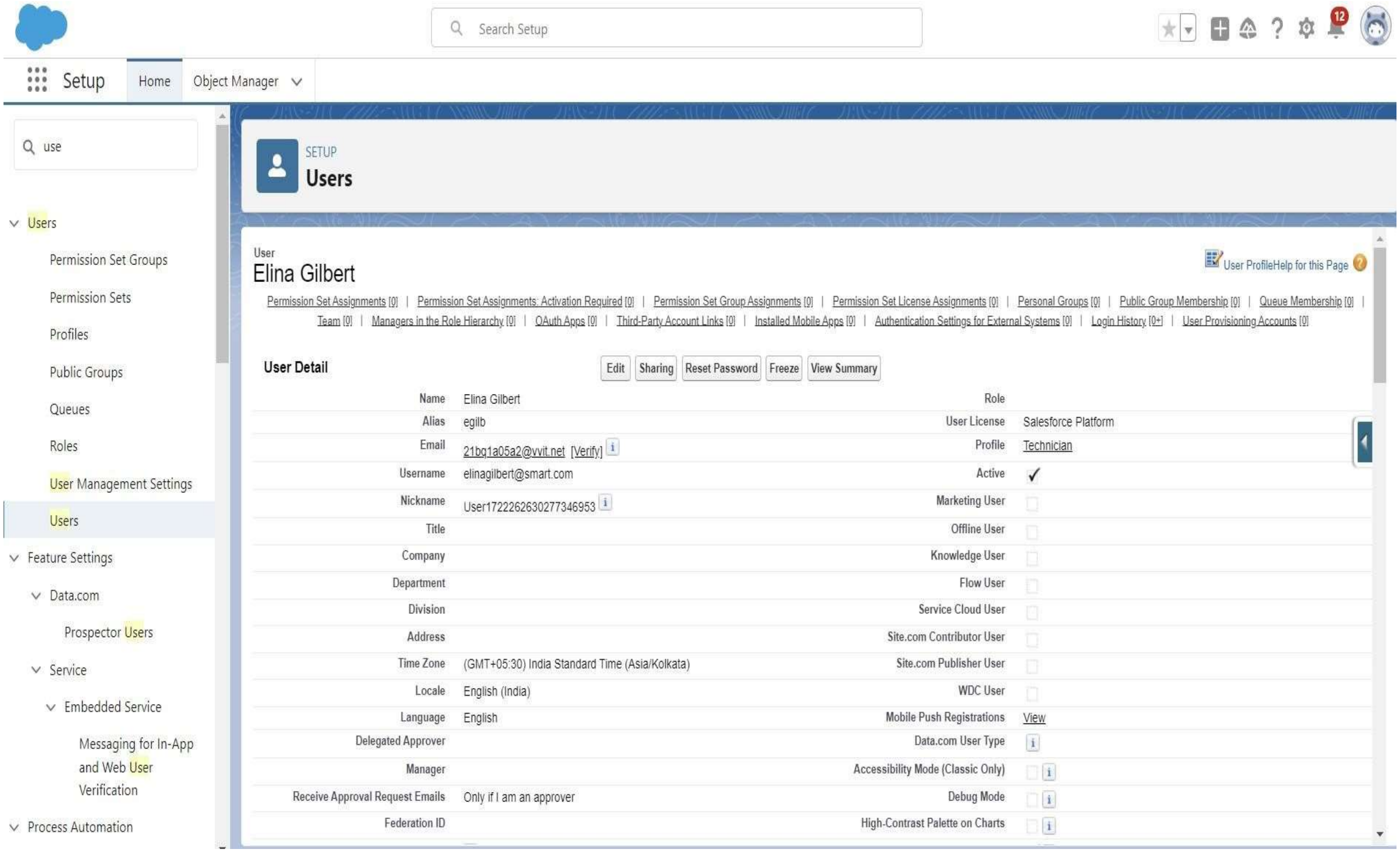
1. Go to setup --> type users in the quick find box --> select users --> click New user.
2. Fill in the fields

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



User

Elina Gilbert

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]

User Detail

Name	Elina Gilbert	Role	
Alias	egilb	User License	Salesforce Platform
Email	21bq1a05a2@vvit.net [Verify]	Profile	Technician
Username	elinagilbert@smart.com	Active	✓
Nickname	User1722262630277346953	Marketing User	□
Title		Offline User	□
Company		Knowledge User	□
Department		Flow User	□
Division		Service Cloud User	□
Address		Site.com Contributor User	□
Time Zone	(GMT+05:30) India Standard Time (Asia/Kolkata)	Site.com Publisher User	□
Locale	English (India)	WDC User	□
Language	English	Mobile Push Registrations	View
Delegated Approver		Data.com User Type	
Manager		Accessibility Mode (Classic Only)	□
Receive Approval Request Emails	Only if I am an approver	Debug Mode	□
Federation ID		High-Contrast Palette on Charts	□

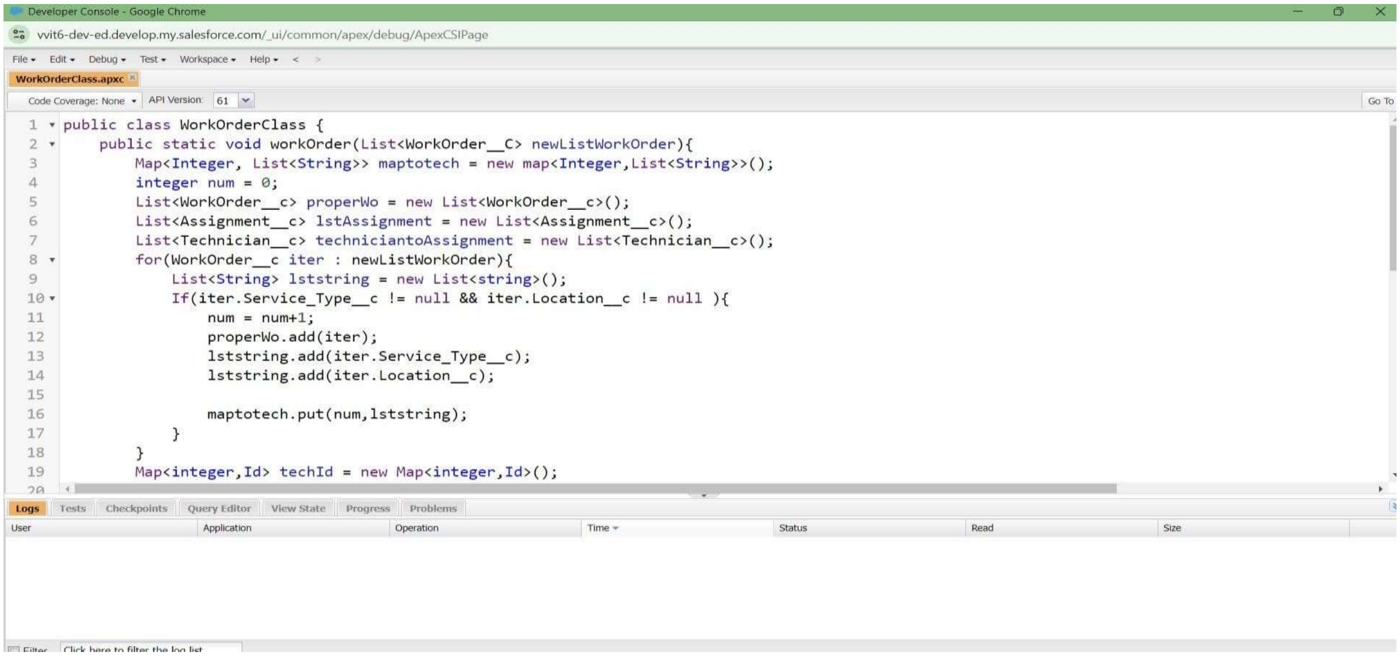
Task 7:

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)



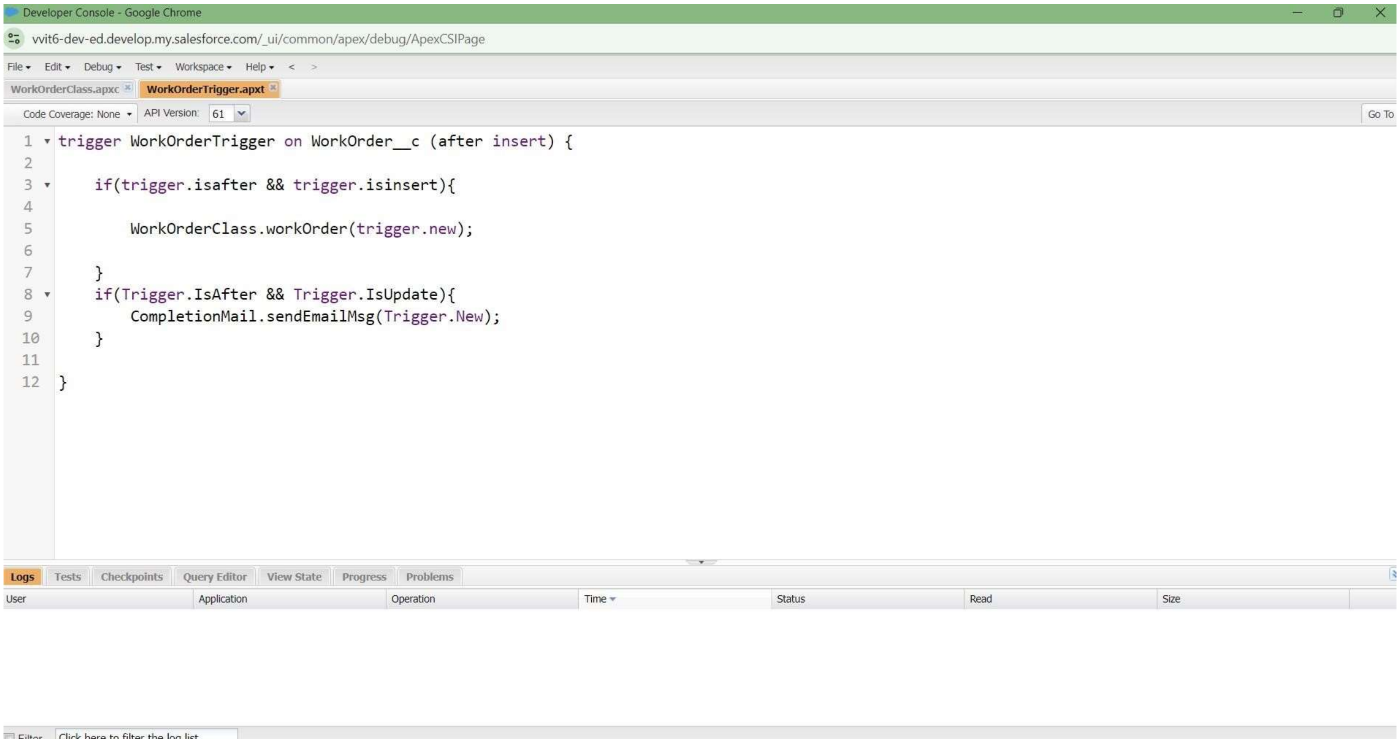
```
public class WorkOrderClass {
    public static void workOrder(List<WorkOrder__C> newListWorkOrder){
        Map<Integer, List<String>> maptotech = new map<Integer,List<String>>();
        integer num = 0;
        List<WorkOrder__c> properWo = new List<WorkOrder__c>();
        List<Assignment__c>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

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 object.
3. Click Submit.
4. Now write the code logic here
5. **Source Code:** trigger WorkOrderTrigger on WorkOrder__c (after insert) {

```
  if(trigger.isafter && trigger.isinsert){
     WorkOrderClass.workOrder(trigger.new);
  }
}
```

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



```
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);
    }

}
```

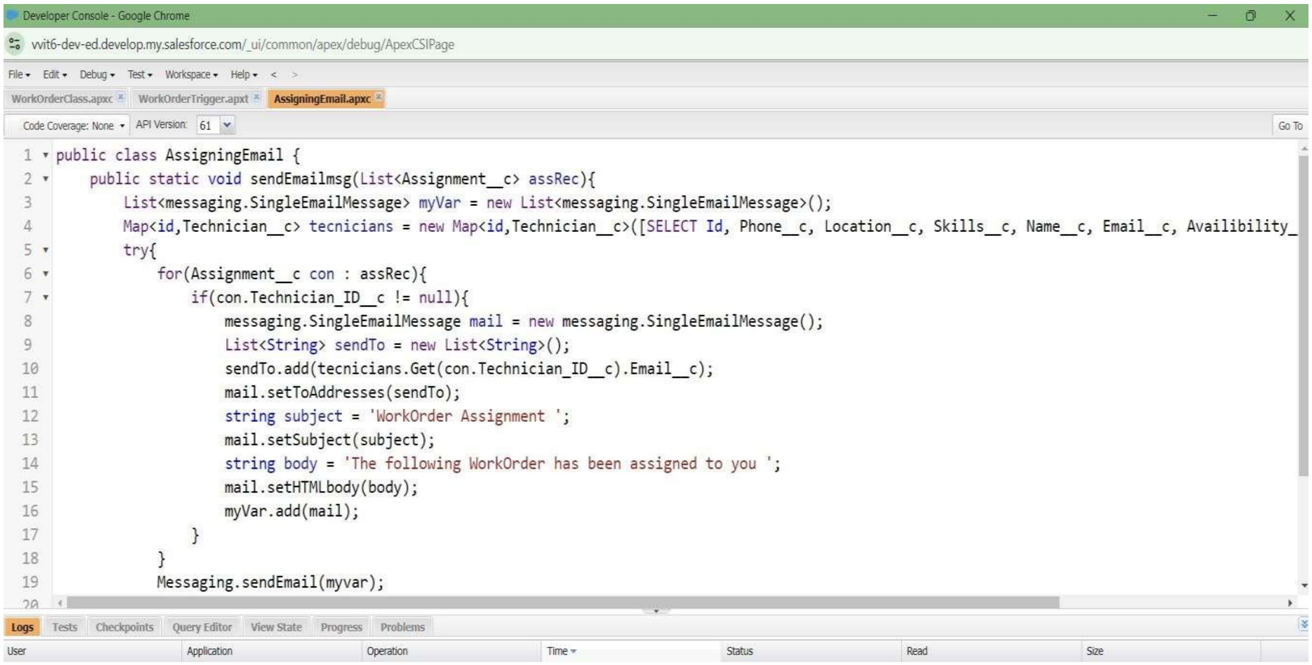
Create an Apex Class

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)



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
AssigningEmail.apxc
Code Coverage: None ▾
API Version: 61
Go To

```
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);
```

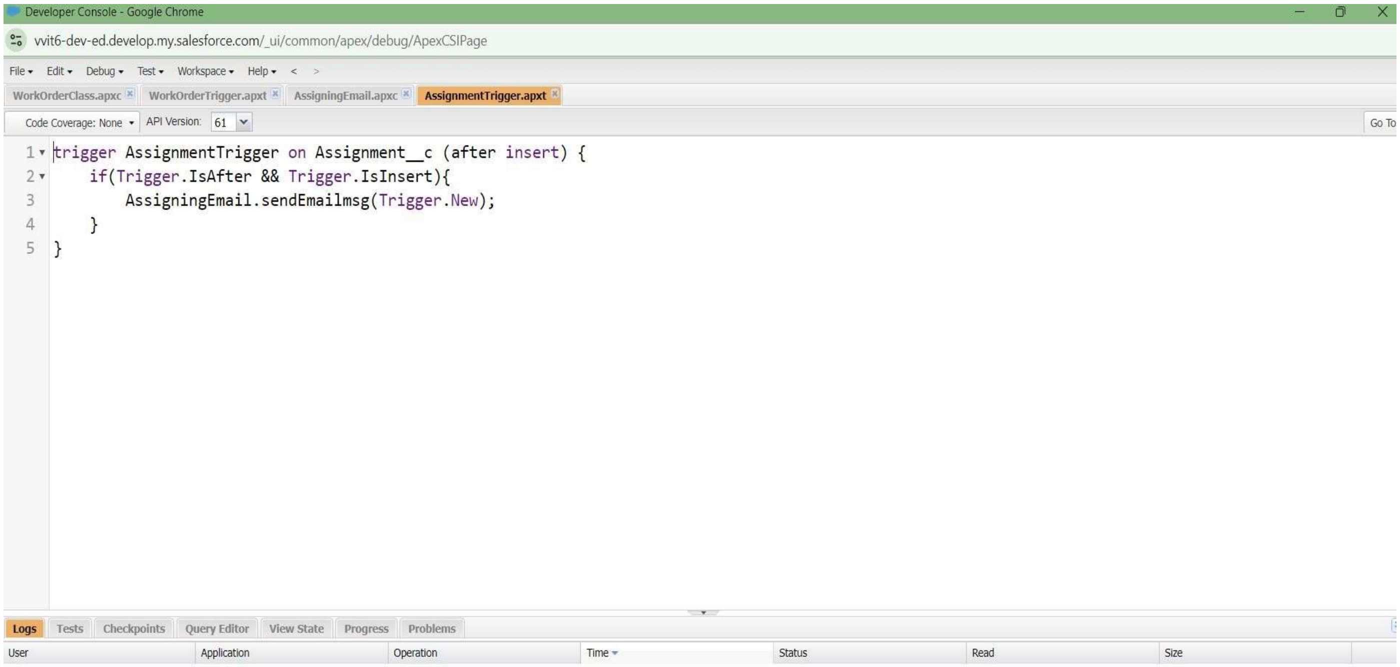
Create an Apex Trigger

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)


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
AssigningEmail.apxc
AssignmentTrigger.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

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

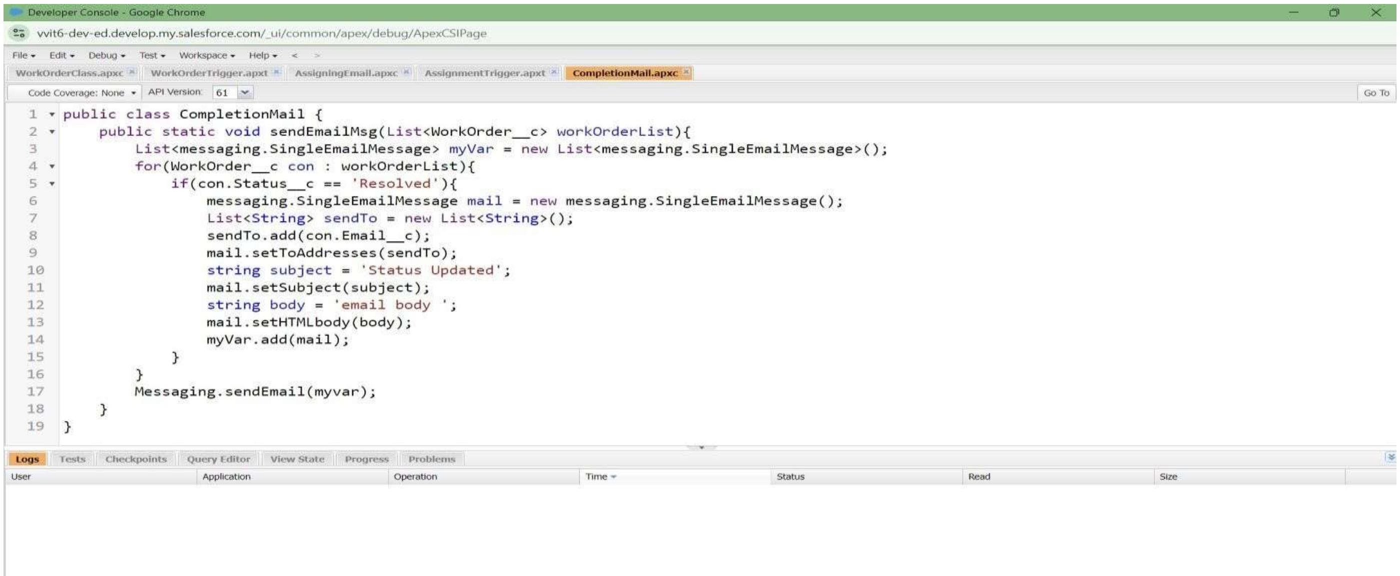
Create an Apex Class

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)



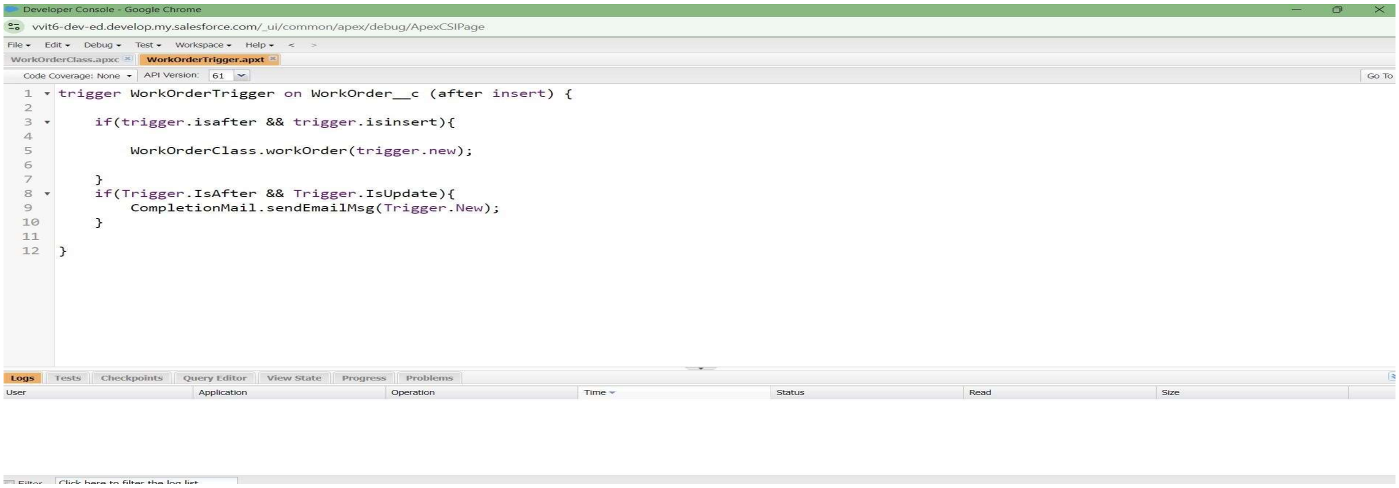
```
public class CompletionMail {
    public static void sendEmailMsg(List<WorkOrder__c> workOrderList){
        List<messaging.SingleEmailMessage> myVar = new List<messaging.SingleEmailMessage>();
        for(WorkOrder__c con : workOrderList){
            if(con.Status__c == 'Resolved'){
                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);
    }
}
```

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)



```
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);
    }

}
```

Create an Asynchronous Apex Class

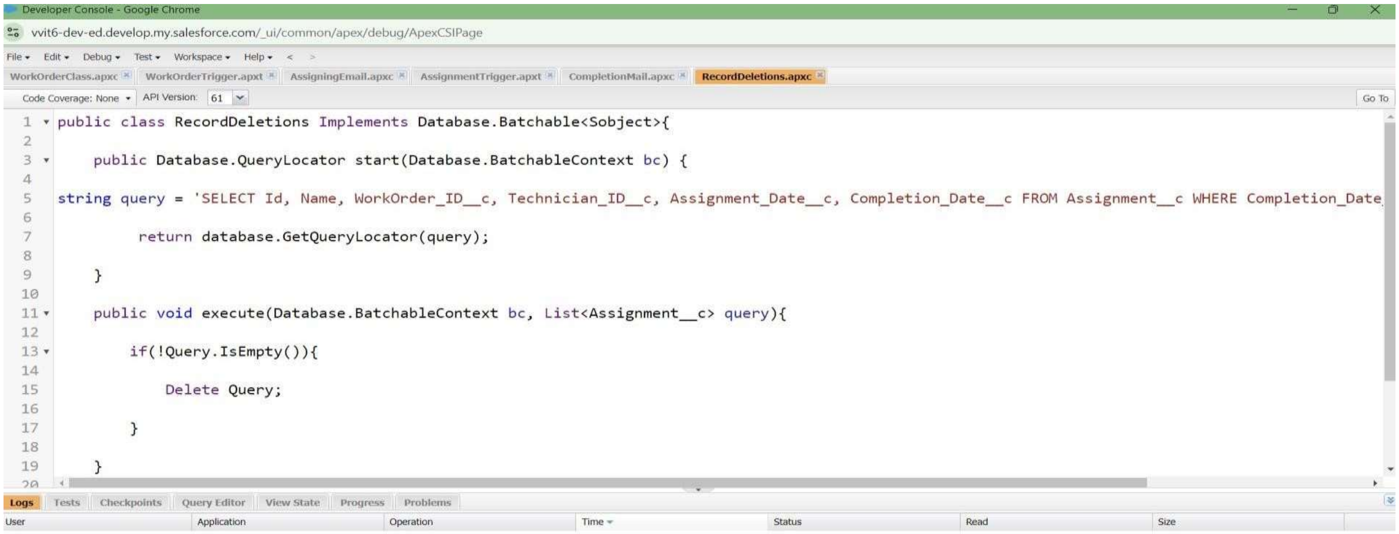
Create an Apex Class to Delete all the WorkOrder records which meets the following criteriaL

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){
 }
}
```

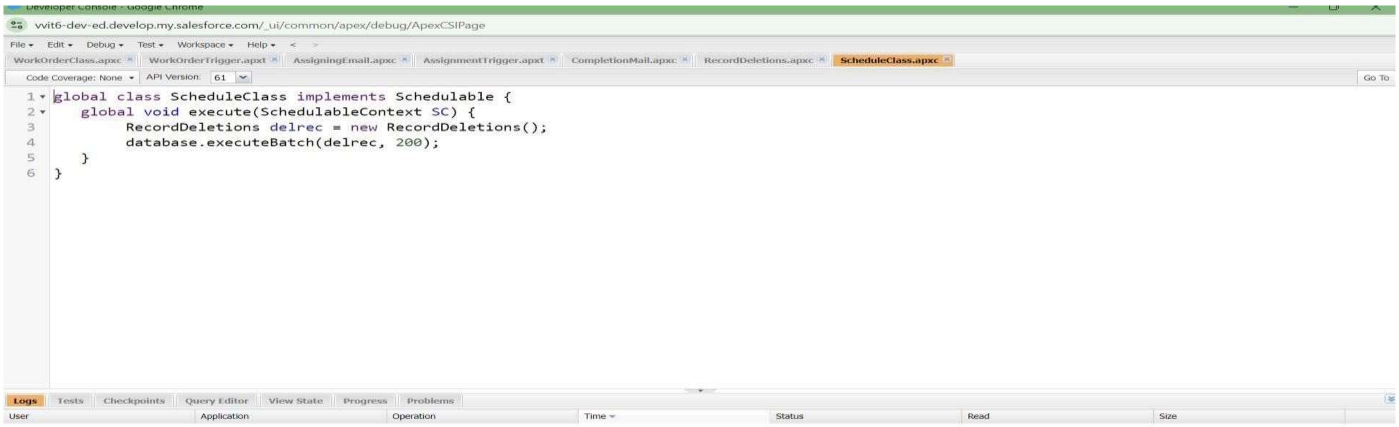
7. Save the code.(click on file --> Save)



```
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;

        }

    }
```

Create an Apex Schedule Class

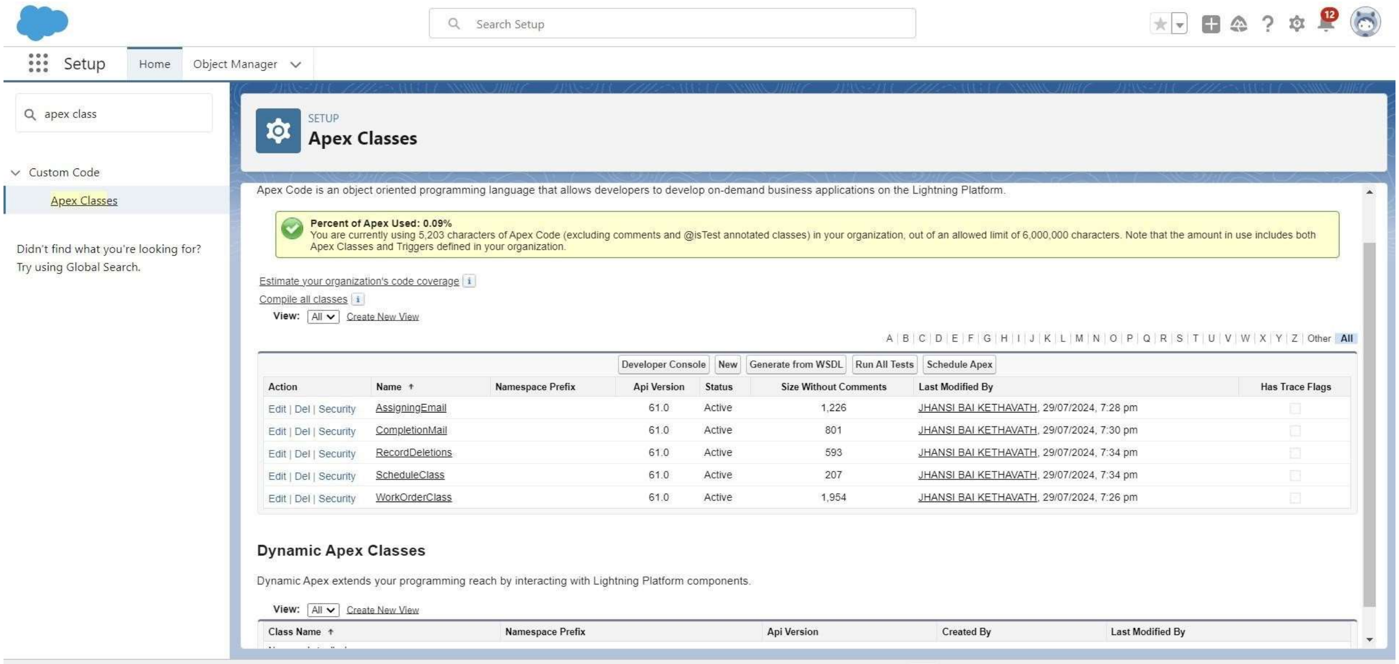
1. Go to Setup --> Click on the gear icon --> Select Developer Console.
2. Then we can see the Developer console. Click on the developer console and you will navigate to a new console window.
3. To create a new Apex Class follow the below steps: Click on the file --> New --> Apex Class.
4. Give the Apex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

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.


Search Setup
Setup
Home
Object Manager
apex class
Custom Code
Apex Classes
Didn't find what you're looking for? Try using Global Search.
SETUP
Apex Classes
Estimate your organization's code coverage
Compile all classes
View: All
Create New View
A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other | All
Developer Console
New
Generate from WSDL
Run All Tests
Schedule Apex

Apex Code is an object oriented programming language that allows developers to develop on-demand business applications on the Lightning Platform.

Percent of Apex Used: 0.09%
You are currently using 5,203 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	JHANSI BAI KETHAVATH, 29/07/2024, 7:28 pm	□
Edit | Del | Security	CompletionMail		61.0	Active	801	JHANSI BAI KETHAVATH, 29/07/2024, 7:30 pm	□
Edit | Del | Security	RecordDeletions		61.0	Active	593	JHANSI BAI KETHAVATH, 29/07/2024, 7:34 pm	□
Edit | Del | Security	ScheduleClass		61.0	Active	207	JHANSI BAI KETHAVATH, 29/07/2024, 7:34 pm	□
Edit | Del | Security	WorkOrderClass		61.0	Active	1,954	JHANSI BAI KETHAVATH, 29/07/2024, 7:26 pm	□

Dynamic Apex Classes

Dynamic Apex extends your programming reach by interacting with Lightning Platform components.

View: All Create New View

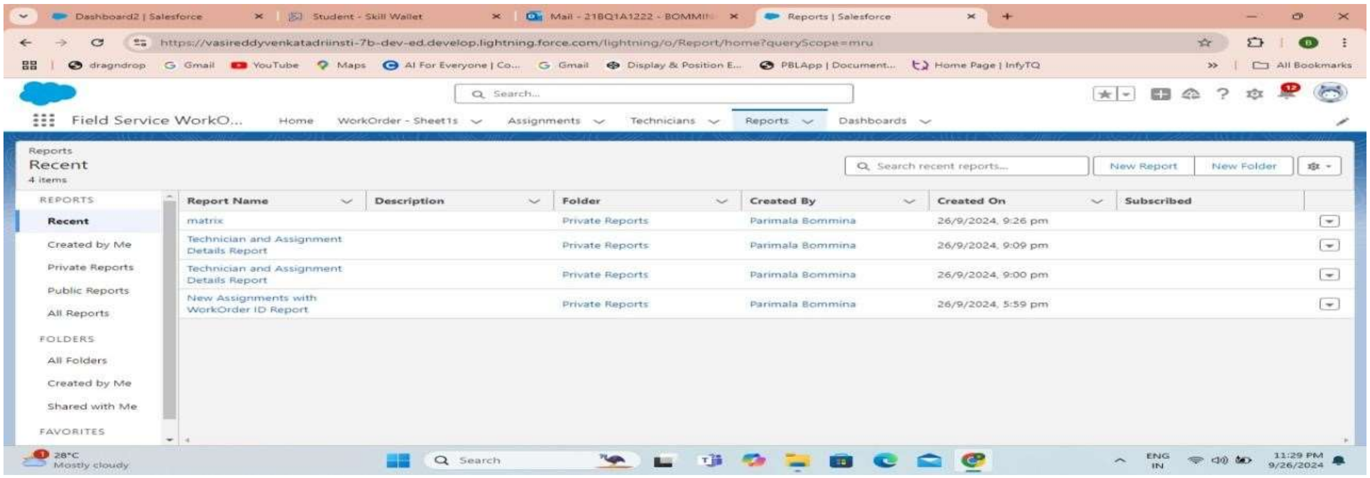
Class Name ↑	Namespace Prefix	Api Version	Created By	Last Modified By

Task 8:

Report

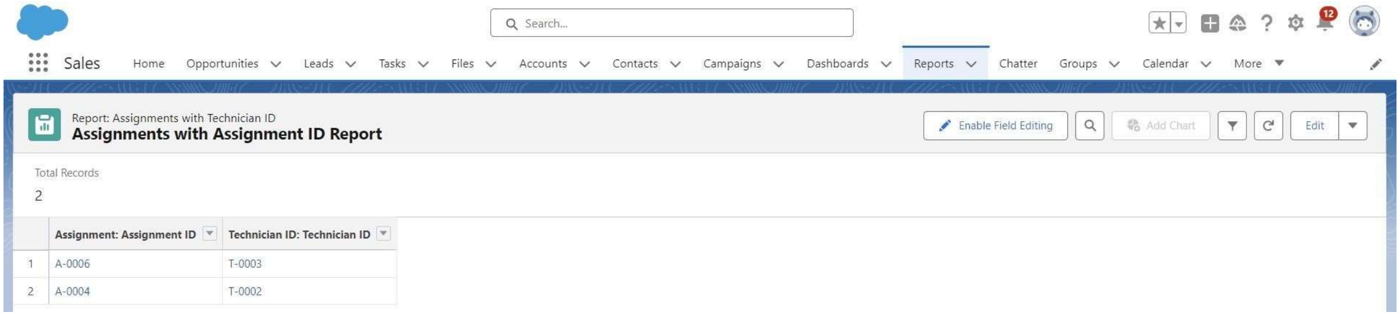
1. Go to the app --> click on the reports tab
2. Click New Report.
3. Select report type from category or from report type panel or from search panel --> click on start report.
4. Customize your report
5. Add fields from left pane as shown below
6. Grouped by workorder ID
7. Save or run it.

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



Reports
Recent
4 items

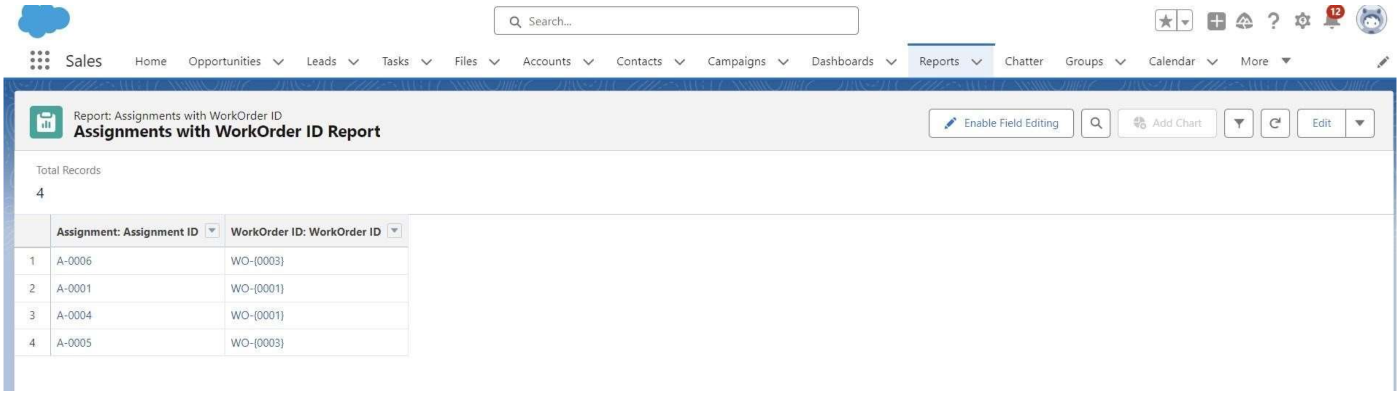
Report Name	Description	Folder	Created By	Created On	Subscribed
matrix		Private Reports	Parimala Bommina	26/9/2024, 9:26 pm	
Technician and Assignment Details Report		Private Reports	Parimala Bommina	26/9/2024, 9:09 pm	
Technician and Assignment Details Report		Private Reports	Parimala Bommina	26/9/2024, 9:00 pm	
New Assignments with WorkOrder ID Report		Private Reports	Parimala Bommina	26/9/2024, 5:59 pm	



Report: Assignments with Technician ID
Assignments with Assignment ID Report

Total Records
2

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



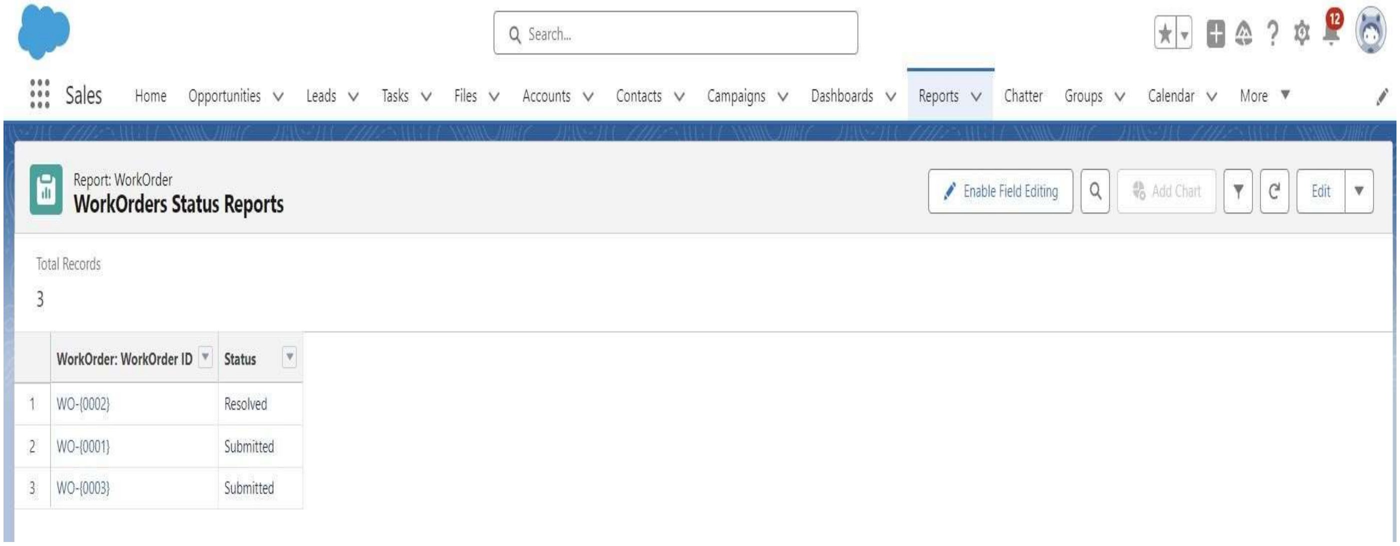
Report: Assignments with WorkOrder ID
Assignments with WorkOrder ID Report

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}

Create Reports

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



Report: WorkOrder

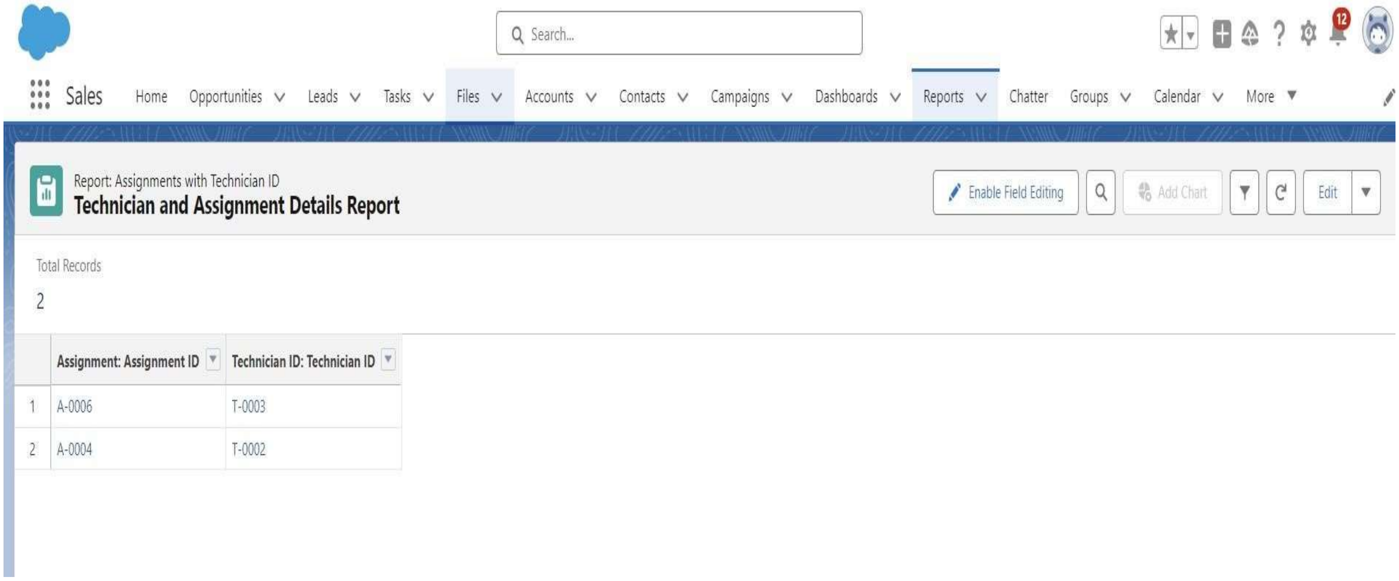
WorkOrders Status Reports

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

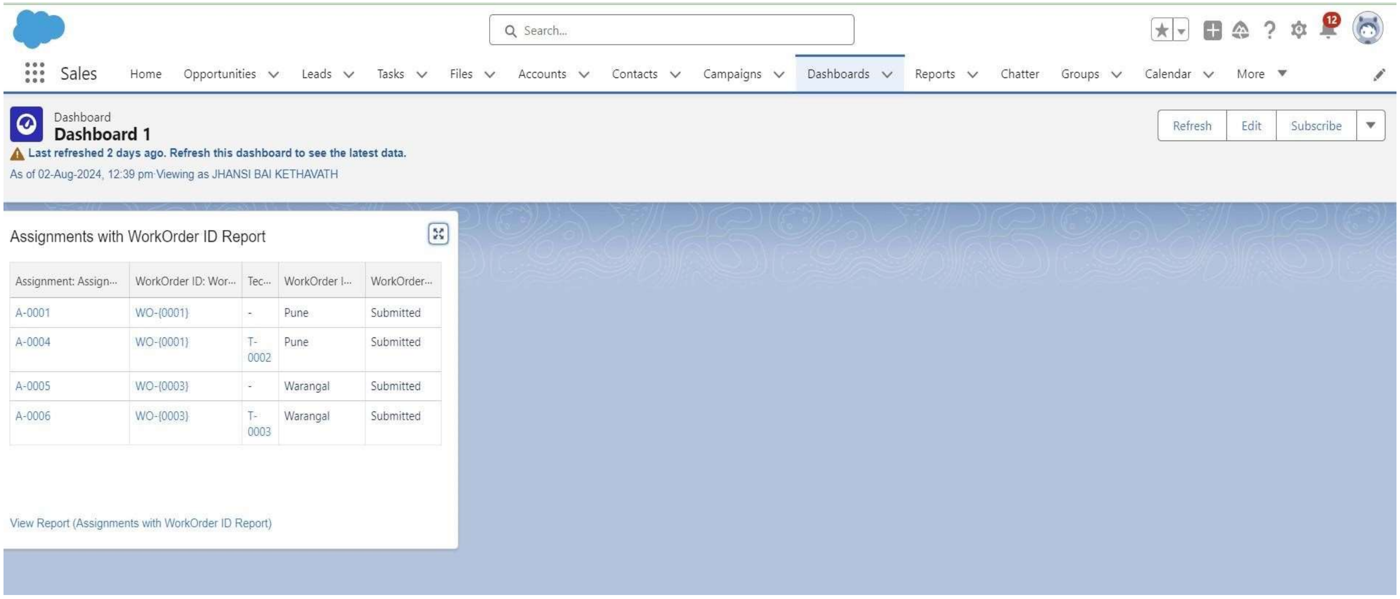
Total Records

2

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

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
Dashboard 1
Last refreshed 2 days ago. Refresh this dashboard to see the latest data.
As of 02-Aug-2024, 12:39 pm Viewing as JHANSI BAI KETHAVATH

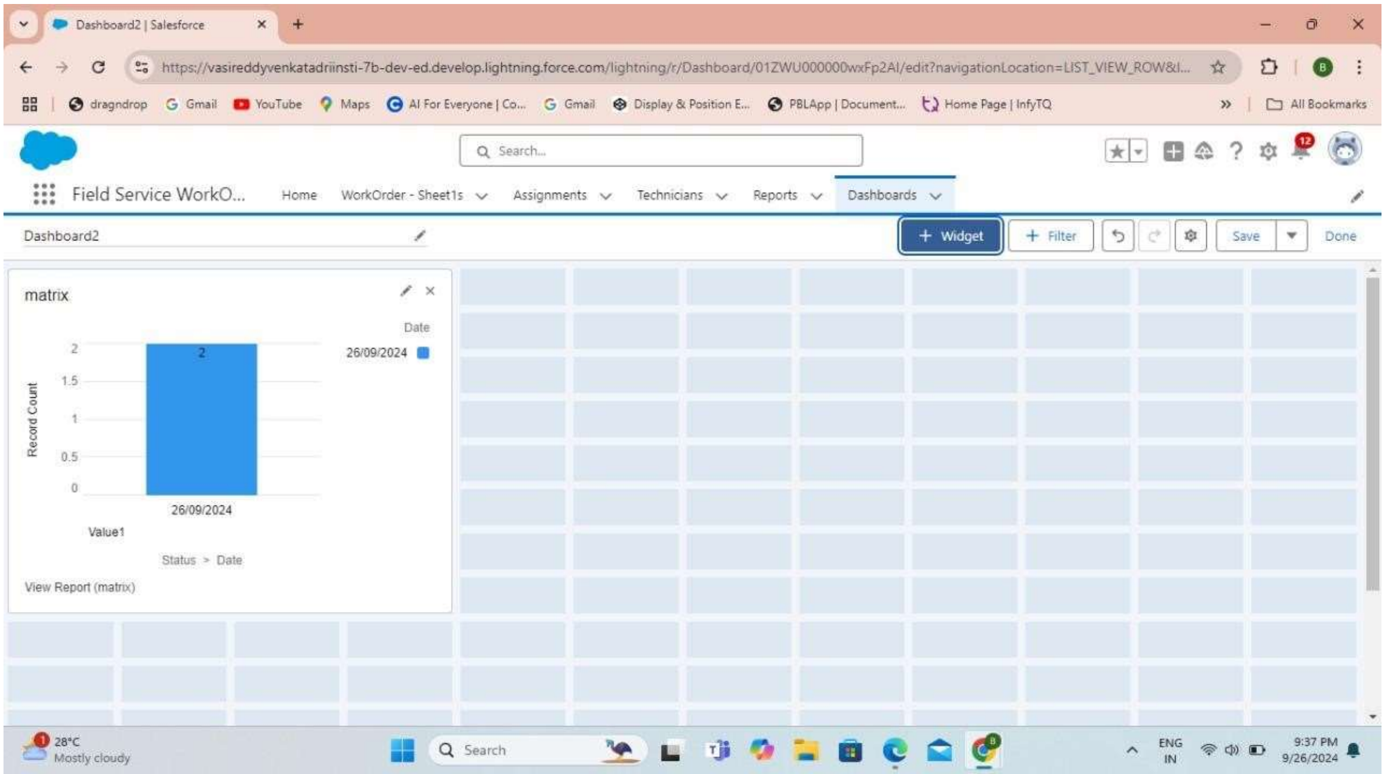
Assignments with WorkOrder ID Report

Assignment: Assign...	WorkOrder ID: Wor...	Tec...	WorkOrder I...	WorkOrder...
A-0001	WO-{0001}	-	Pune	Submitted
A-0004	WO-{0001}	T-0002	Pune	Submitted
A-0005	WO-{0003}	-	Warangal	Submitted
A-0006	WO-{0003}	T-0003	Warangal	Submitted

View Report (Assignments with WorkOrder ID Report)

Create Dashboards

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



THANK YOU