

FIELD SERVICE

WORKORDER OPTIMIZATION

By

MOUNIKA MEKA

21bq1a05e2@vvit.net

ABSTRACT

Objective: 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

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

SI No.	Module or Tasks Labels	Page No.
1	<u>Task 1: Object</u> 1.1 Create Technician Object 1.2 Create WorkOrder Object 1.3 Create Assignment Object	5-7
2	<u>Task 2: Tabs</u> 2 Create a custom tab	8

3	<u>Task 3: The Lightning App</u> 3 Create a Lightning App	9-10
4	<u>Task 4: Fields & Relationship</u> 4.1 Creating Lookup Field in Assignment Object 4.2 Manage your picklist values 4.3 Manage your picklist values 4.4 Creating Formula Field in WorkOrder Object	11-14
5	<u>Task 5: Profiles</u> 5 Technician Profile	16
6	<u>Task 6: Users</u> 6 Create User	17
7	<u>Task 7: Apex Trigger</u> 7.1 Create an Apex Class 7.2 Create an Apex Trigger 7.3 Create an Apex Class 7.4 Create an Apex Trigger 7.5 Create an Apex Class 7.6 Create an Apex Trigger 7.7 Create an Asynchronous Apex Class 7.8 Create an Apex Schedule Class 7.9 Create an Schedule apex	18-28

8	<u>Task 8: Reports & Dashboards</u> 8.1 Report 8.2 Create Reports 8.3 Dashboard 8.4 Create Dashboards	29-31
---	--	-------

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.

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		21bq1a05e2@vvit.net
 Location	 Location	Picklist		Hyderabad
 Availability	 Availability	Picklist		Available
 Skills	 Skills	Picklist		Machine Installation

Back

Next

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

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Technician	Technician__c	Custom Object		20/07/2024	✓

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 Field Labels Row: 1 Import 2 rows of data: No, skip import 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-10001
✓ Email	Email	Email	✓	example1@workorder.com
✓ Service Type	Service Type	Text	✓	Maintenance
✓ Description	Description	Picklist	✓	
✓ Location	Location	Text Area (Long)	✓	Paris
✓ Priority	Priority	Picklist	✓	Low

Back Next

After creating the WorkOrder Custom object it looks like the below

Setup Home Object Manager

Object Manager

1 item, Sorted by Label

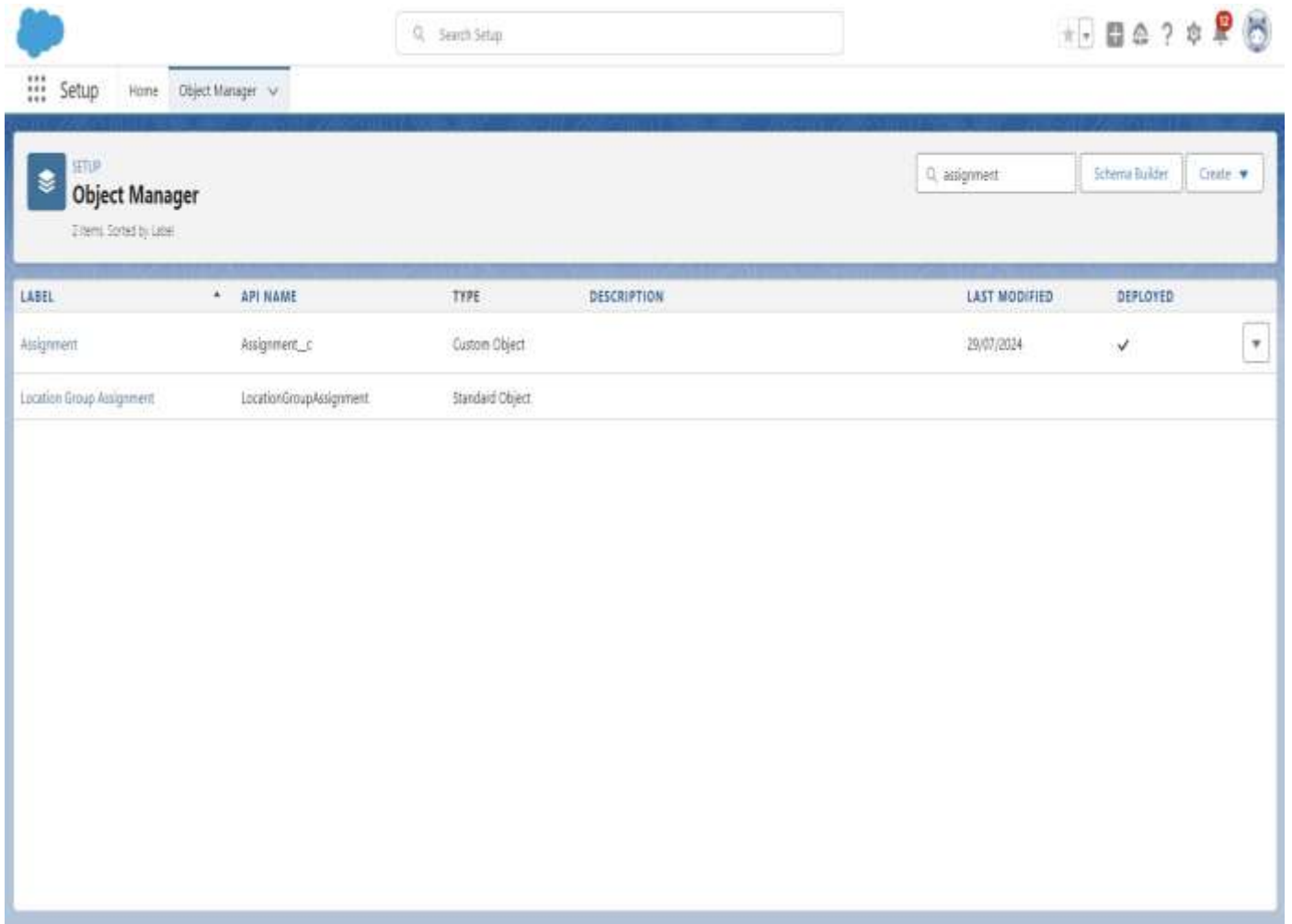
workorder Schema Builder Create

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

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



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	Assignment	New	
Edit / Del	Technician	New	
Edit / Del	WorkOrder	New	

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?](#)

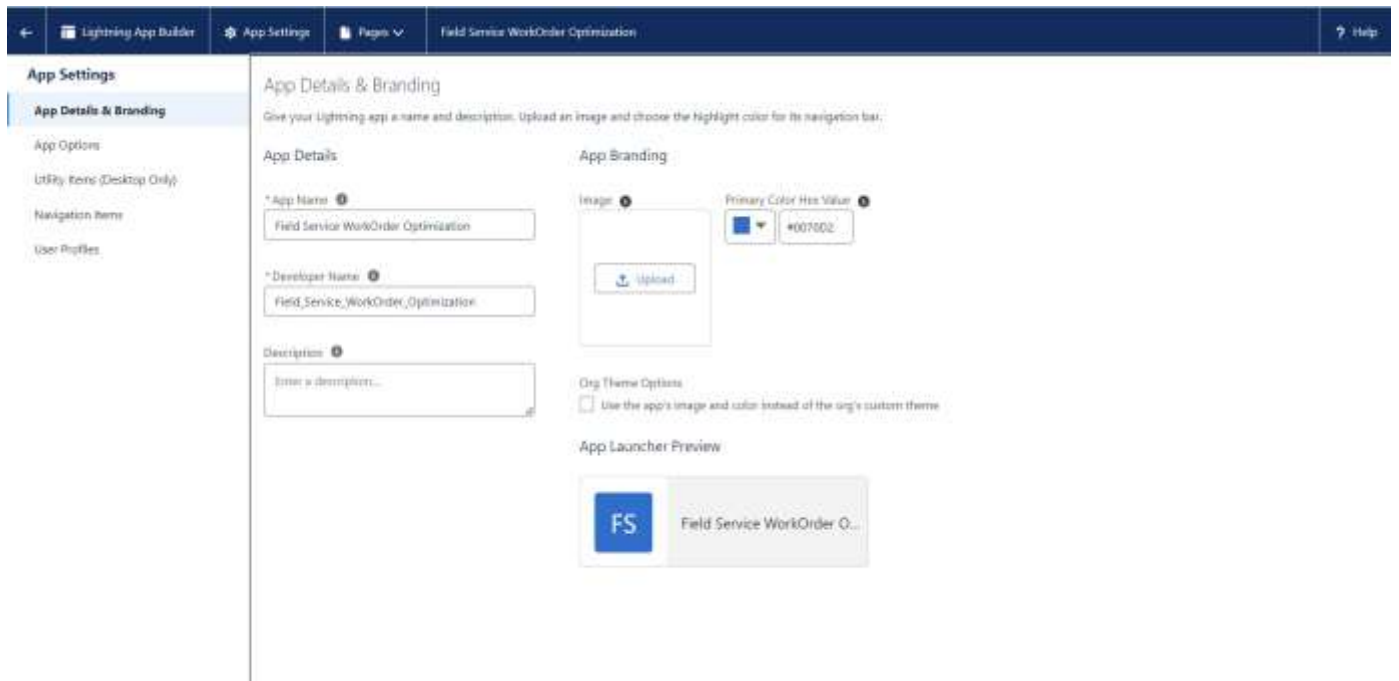
[https://www-dev-ed.trailhead.com/lightning/setup/CustomTabsHome](#)

Task 3 :

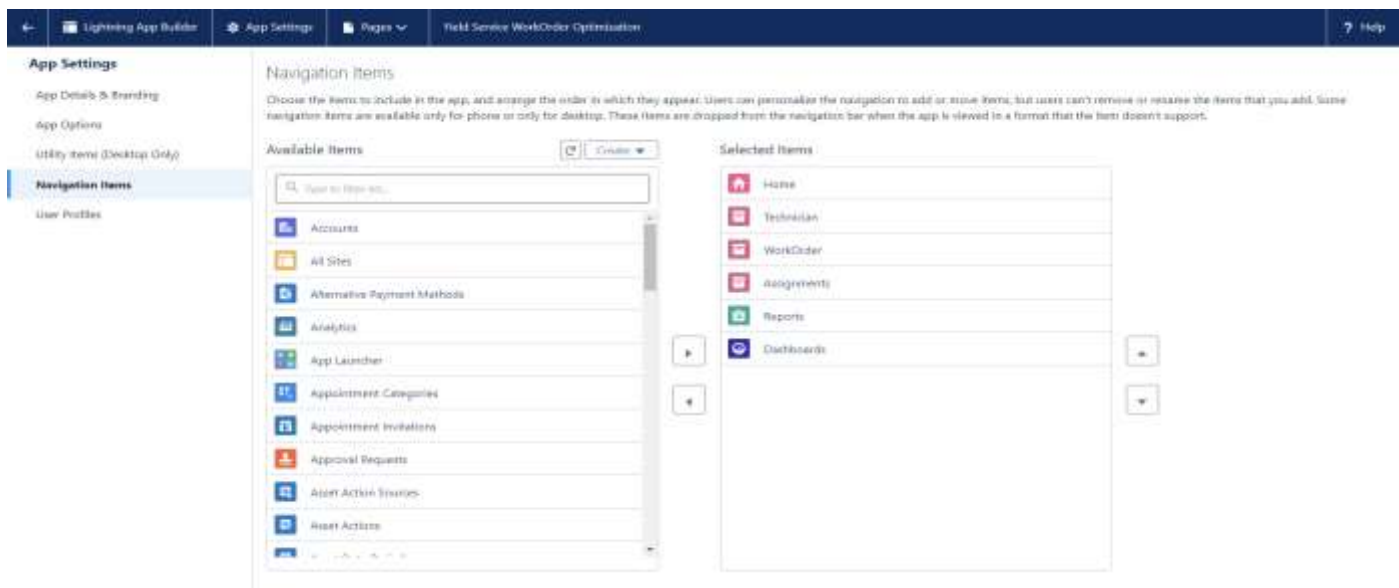
Create a Lightning App

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



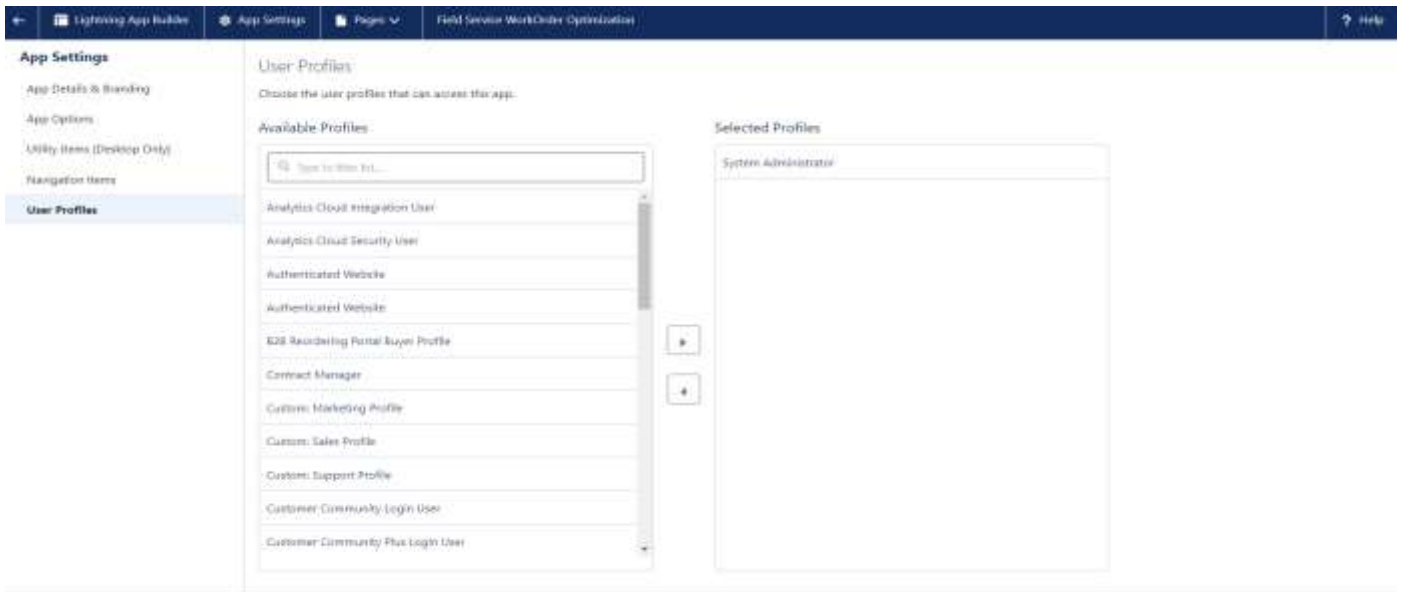
3. Then click Next --> (App option page) keep it as default --> Next --> (Utility Items) keep it as default --> Next
4. To Add Navigation Items:



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.

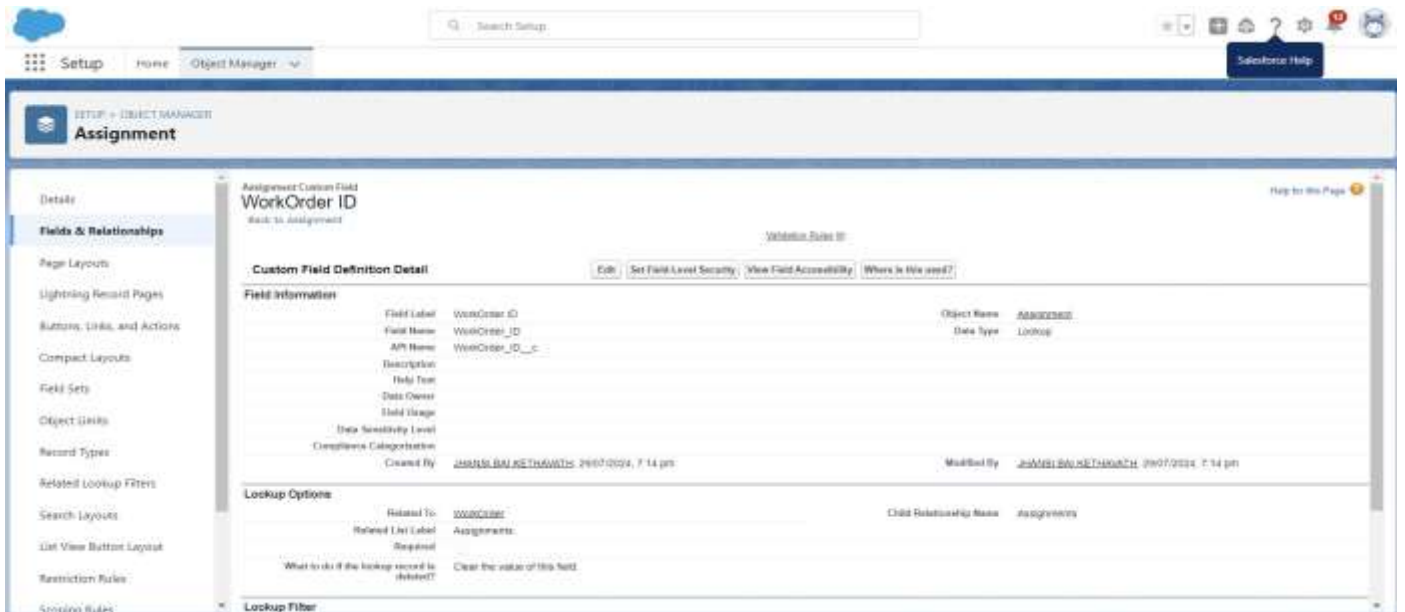


This is the output after completion of following the above procedure.

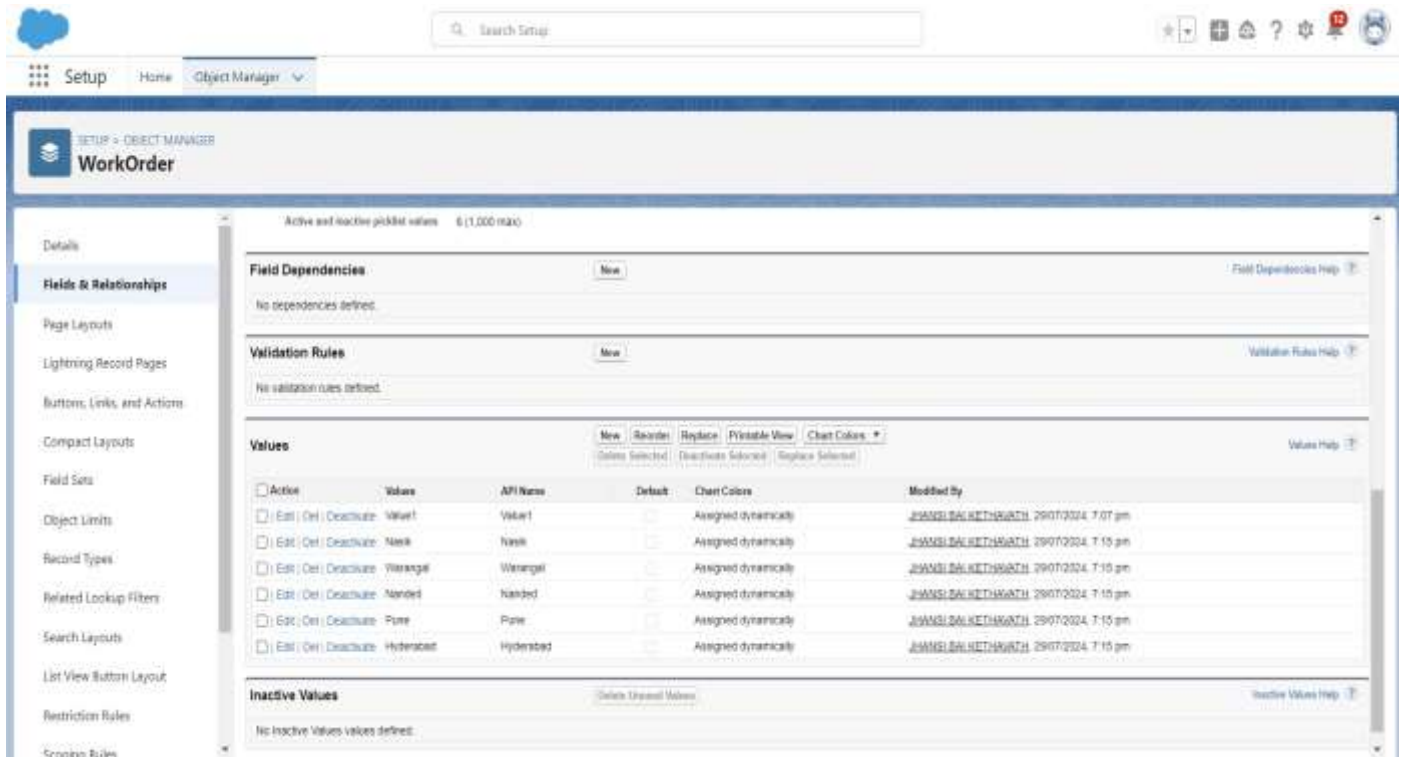
Task 4 :

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.



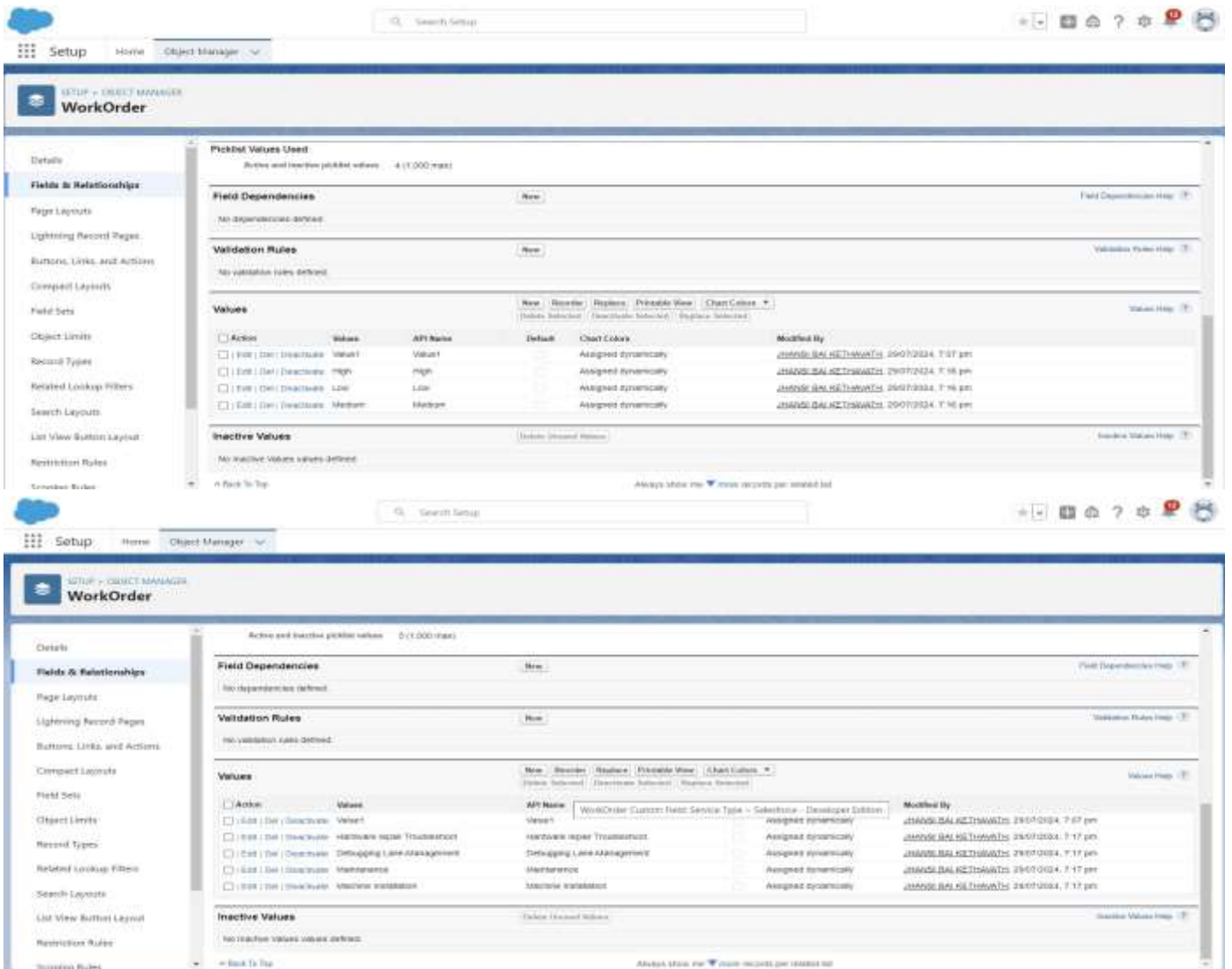
4.2 Manage your picklist values



4.3 Manage your picklist values :

Add following values to the respective fields in WorkOrder object:

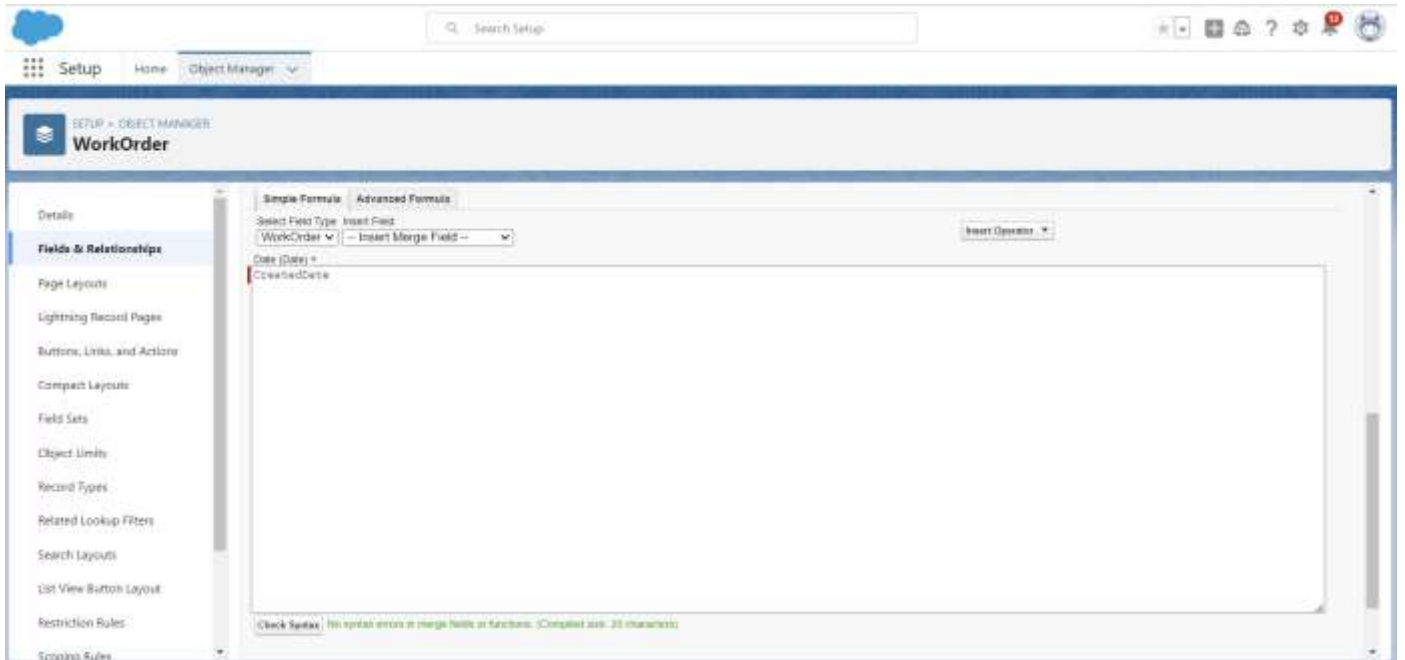
Field	Values
Priority	High
Service Type	Hardware repair Troubleshoot/Debugging Lane-Management



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.



4.5 Creating Remaining fields for the respective objects

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

SI No	Object Name	Field	
1	Assignment	Field Name	Datatype
		<ul style="list-style-type: none"> Technician ID Assignment Date Completion Date 	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)

The screenshot shows the Salesforce Setup interface, specifically the 'Object Manager' section for the 'Assignment' object. The left sidebar lists various setup options, with 'Fields & Relationships' selected. The main content area displays a table of fields for the 'Assignment' object, sorted by Field Label. The table has columns for Field Label, Field Name, Data Type, Controlling Field, and Indexed. The fields listed are: Assignment Date, Assignment ID, Completion Date, Created By, Last Modified By, Owner, Technician ID, and WorkOrder ID.

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.

Setup Home Object Manager

Search Setup

profile

Users

Profiles

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

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.

Lead: IT Home (S) | Enabled: Auto Close Access (S) | Enabled: Viewforce Page Access (S) | Enabled: External Data Source Access (S) | Enabled: External Contentful Source (S) | Enabled: External Contentful Profile Access (S) | Enabled: Custom Metadata Page Access (S) | Enabled: Custom Settings Definition Access (S) | Enabled: Price Access (S) | Enabled: Service Presence Status Access (S) | Enabled: Custom Permissions (S)

Profile Detail Edit Clone View Users

Name: Standard Platform User
User License: Salesforce Platform
Created By: salesforce.com, inc. 28/07/2024, 7:27 pm
Custom Profile
Modified By: JHANGU BAL PUTHIAJATHI 28/07/2024, 7:24 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)	Opening Hours	Opening Hours Layout (View Assignment)

Custom Object Layouts

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)		
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

The screenshot shows the Salesforce Setup interface. On the left, the 'Users' section is expanded under 'User Management Settings'. The main area displays the details for a user named 'Elina Gilbert'. The user's profile is 'Technician'. The 'User Detail' section includes fields for Name, Alias, Email, Username, Nickname, Title, Company, Department, Division, Address, Time Zone, Locale, Language, and various roles and permissions. The 'Active' checkbox is checked. The 'Marketing User' checkbox is unchecked. The 'Office User' checkbox is unchecked. The 'Knowledge User' checkbox is unchecked. The 'Flow User' checkbox is unchecked. The 'Service Cloud User' checkbox is unchecked. The 'Site.com Contributor User' checkbox is unchecked. The 'Site.com Publisher User' checkbox is unchecked. The 'WDC User' checkbox is unchecked. The 'Mobile Push Registration' checkbox is unchecked. The 'Data.com User Type' is set to 'Data.com User Type'. The 'Accessibility Mode (Classic Only)' is set to 'Classic Only'. The 'Debug Mode' is set to 'Debug Mode'. The 'High Contrast Palette on Charts' is set to '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> IstAssignment = new List<Assignment__c>();
        List<Technician__c> techniciantoAssignment = new List<Technician__c>();
        for(WorkOrder__c iter : newListWorkOrder){
            List<String> Iststring = new List<string>();
            If(iter.Service_Type__c != null && iter.Location__c != null ){
```

```

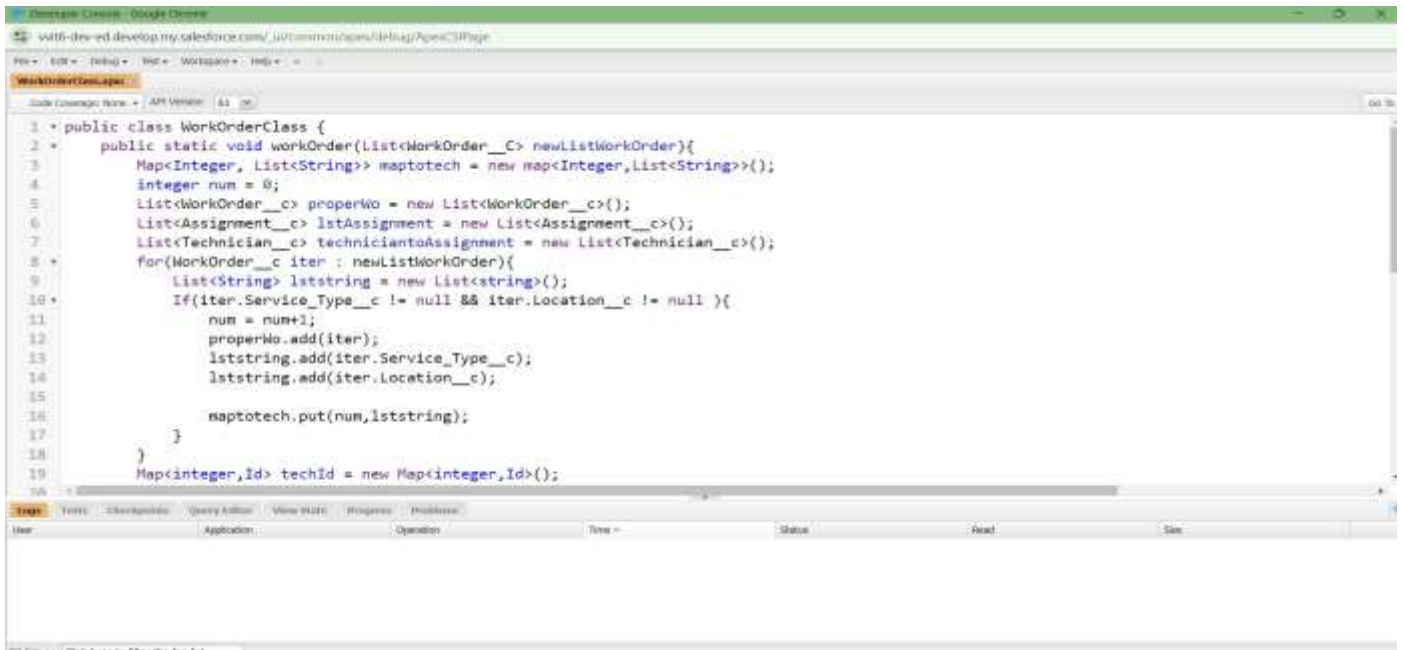
        num = num+1;
        properWo.add(iter);
        lststring.add(iter.Service_Type__c);
        lststring.add(iter.Location__c);

        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, Availability__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.Availability__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;
}
}
}

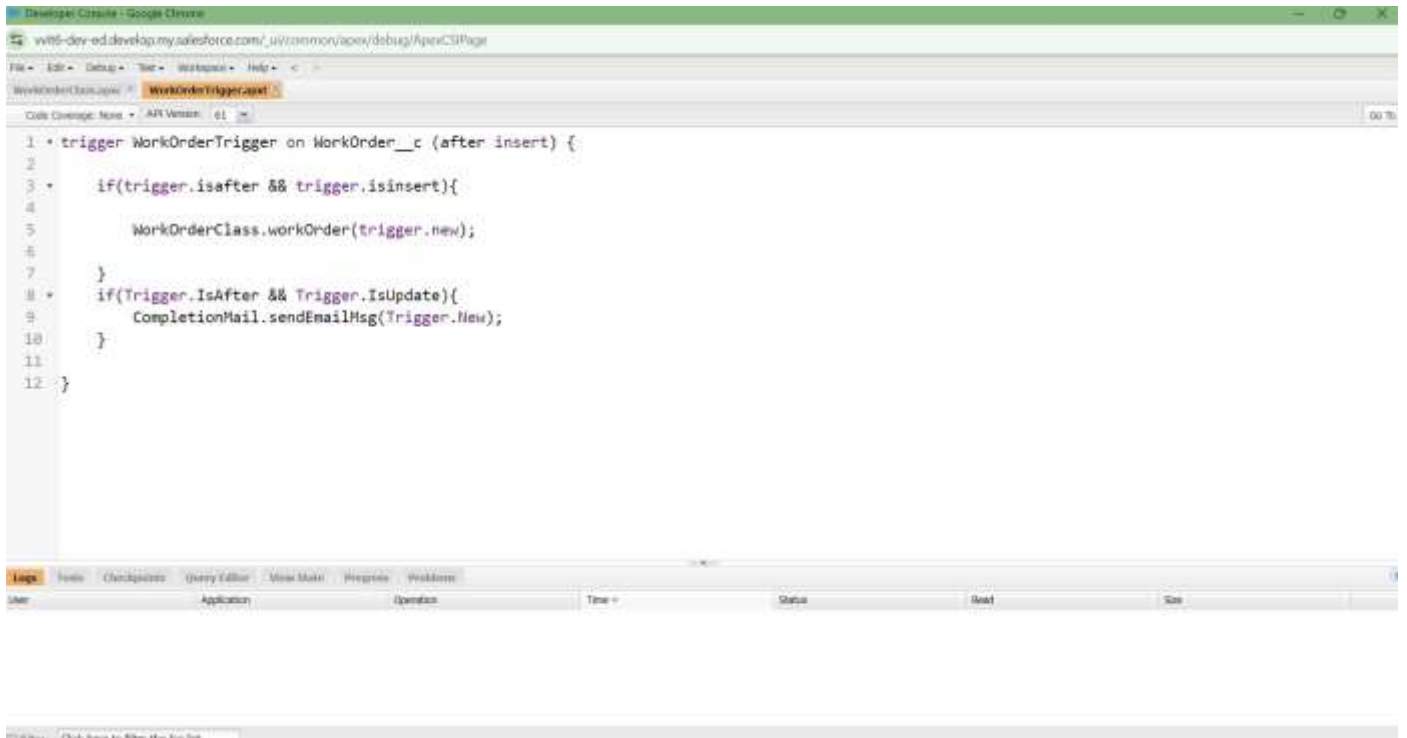
```

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



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)



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> technicians = new Map<id,Technician__c>([SELECT Id, Phone__c,
Location__c, Skills__c, Name__c, Email__c, Availability__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>();

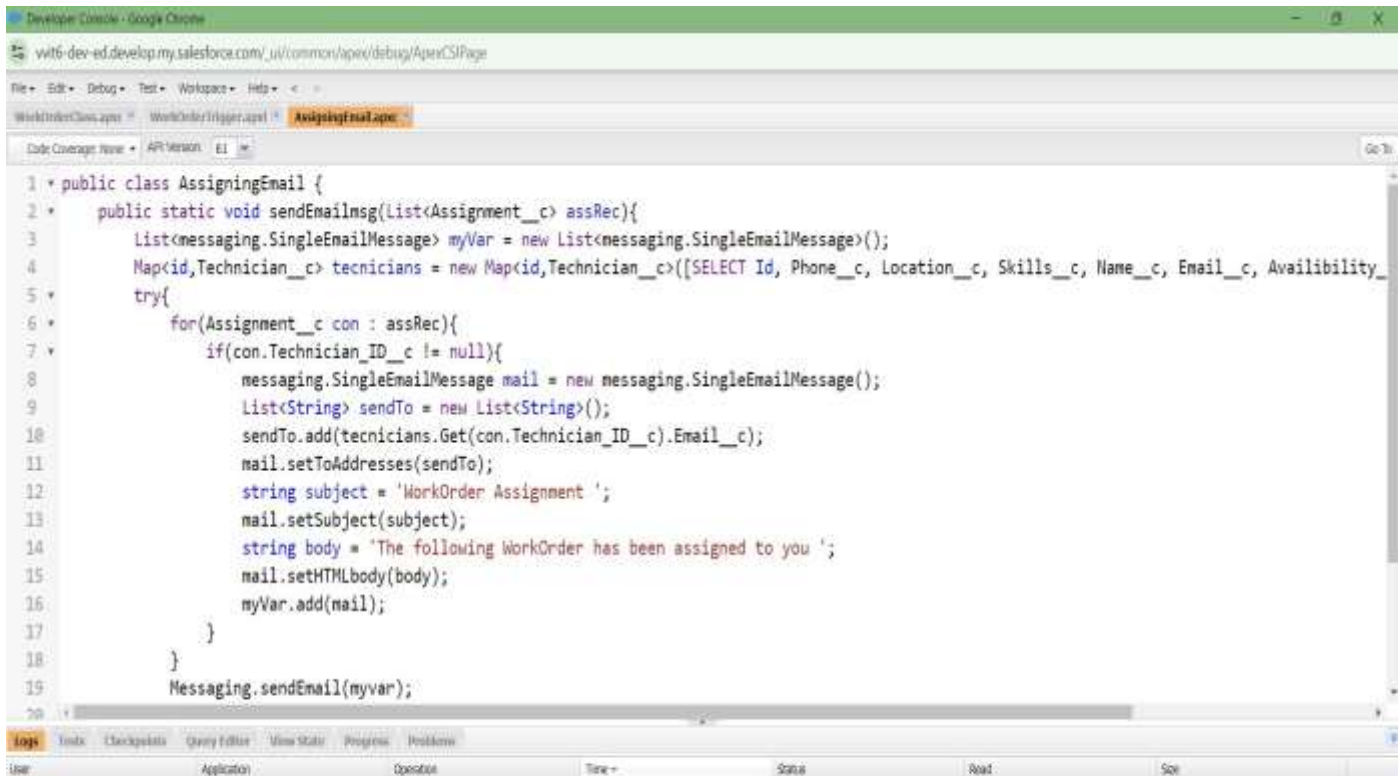
```

```

        sendTo.add(tecnicos.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)

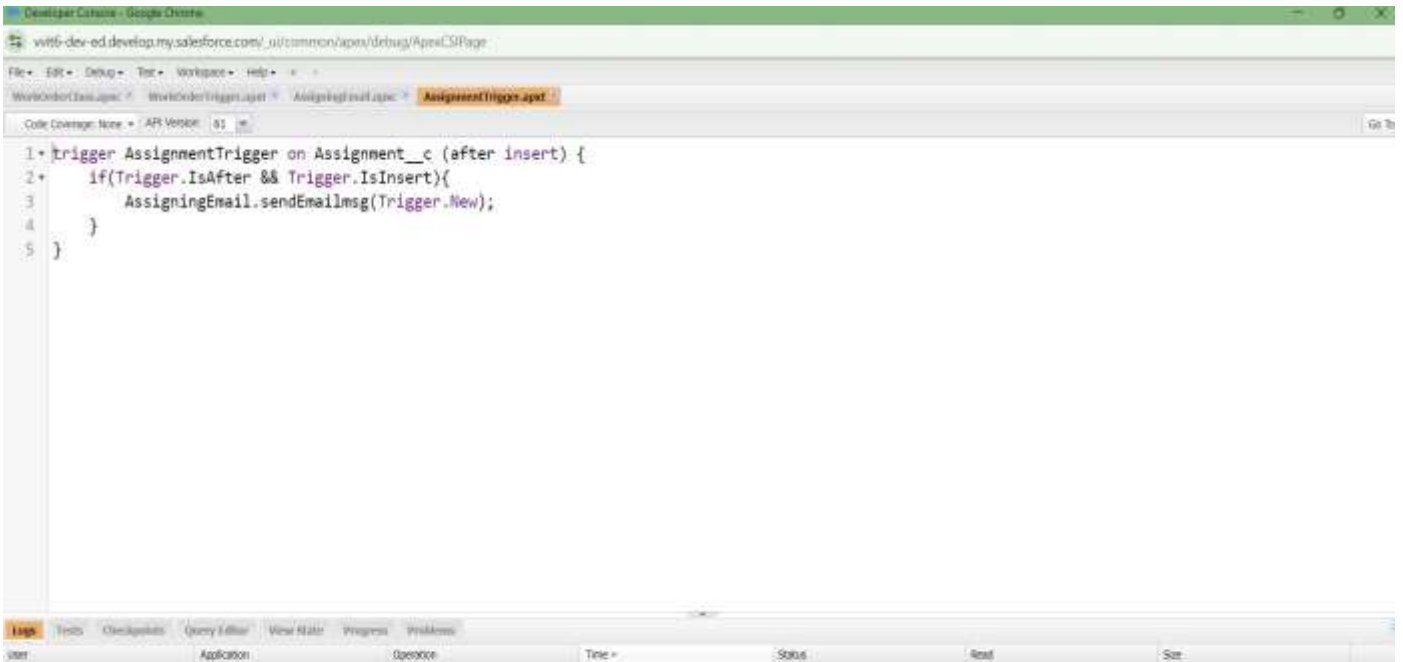


```
1 public class AssigningEmail {
2     public static void sendEmailmsg(List<Assignment__c> assRec){
3         List<messaging.SingleEmailMessage> myVar = new List<messaging.SingleEmailMessage>();
4         Map<id, Technician__c> technicians = new Map<id, Technician__c>([SELECT Id, Phone__c, Location__c, Skills__c, Name__c, Email__c, Availability__c FROM Technician__c]);
5         try{
6             for(Assignment__c con : assRec){
7                 if(con.Technician_ID__c != null){
8                     messaging.SingleEmailMessage mail = new messaging.SingleEmailMessage();
9                     List<String> sendTo = new List<String>();
10                    sendTo.add(technicians.Get(con.Technician_ID__c).Email__c);
11                    mail.setToAddresses(sendTo);
12                    string subject = 'WorkOrder Assignment';
13                    mail.setSubject(subject);
14                    string body = 'The following WorkOrder has been assigned to you';
15                    mail.setHTMLbody(body);
16                    myVar.add(mail);
17                }
18            }
19            Messaging.sendEmail(myVar);
20        } catch (Exception e) {}
21    }
22 }
```

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)



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)



7.6 Create an Apex Trigger

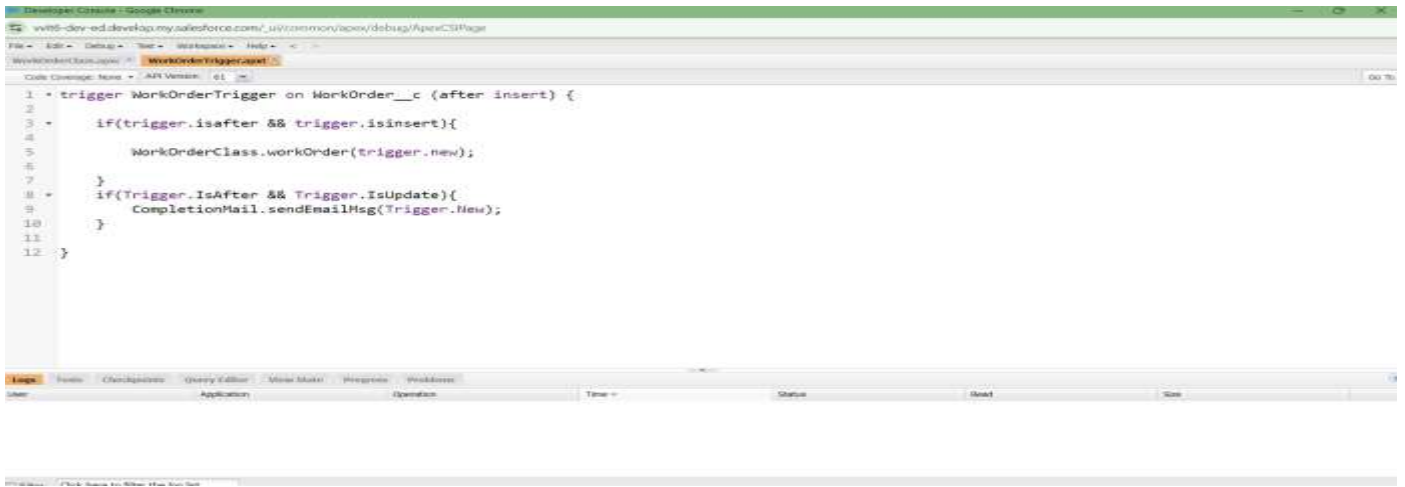
1. Click on the file --> Open.
2. A pop up window opens click on Triggers, then select “WorkOrderTrigger” and click on “Open”
3. Now write the code logic here.

```

4. WorkOrderClass.workOrder(trigger.new);
    }
    if(trigger.IsAfter && trigger.IsUpdate){
        CompletionMail.sendEmailMsg(trigger.New);
    }
}
}

```

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



7.7 Create an Asynchronous Apex Class

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

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)



The screenshot shows the Salesforce Developer Console with the Apex class `RecordDeletions` open. The class implements the `Database.Batchable<Sobject>` interface. The code is as follows:

```
1 public class RecordDeletions implements Database.Batchable<Sobject>{
2
3     public Database.QueryLocator start(Database.BatchableContext bc) {
4
5         string query = 'SELECT Id, Name, WorkOrder_ID__c, Technician_ID__c, Assignment_Date__c, Completion_Date__c FROM Assignment__c WHERE Completion_Date
6
7         return database.GetQueryLocator(query);
8
9     }
10
11     public void execute(Database.BatchableContext bc, list<Assignment__c> query){
12
13         if(!Query.IsEmpty()){
14
15             Delete Query;
16
17         }
18     }
19 }
```

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)

```

1 // Global class ScheduleClass implements Schedulable {
2 //     global void execute(SchedulableContext SC) {
3 //         RecordDeletions delrec = new RecordDeletions();
4 //         database.executeBatch(delrec, 200);
5 //     }
6 // }

```

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.

Apex Classes

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: 5.88%
You are currently using 5,323 characters of Apex Code (excluding comments and @test annotated classes) in your organization, out of an allowed limit of 9,000,000 characters. Note that the amount in use includes both Apex Classes and Triggers defined in your organization.

Estimate your organization's code coverage: [\[?\]](#)

Control all classes: [\[?\]](#)

View: [\[All\]](#) [\[Create New View\]](#)

Action	Name	Namespace Prefix	API Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
Get Del Security	AssignmentClass		61.0	Active	1,326	JHANSI RAJ KETHAWATH 29/07/2024 7:28 pm	<input type="checkbox"/>
Get Del Security	ContestDetail		61.0	Active	901	JHANSI RAJ KETHAWATH 29/07/2024 7:30 pm	<input type="checkbox"/>
Get Del Security	RecordDeletions		61.0	Active	993	JHANSI RAJ KETHAWATH 29/07/2024 7:34 pm	<input type="checkbox"/>
Get Del Security	ScheduleClass		61.0	Active	207	JHANSI RAJ KETHAWATH 29/07/2024 7:34 pm	<input type="checkbox"/>
Get Del Security	WorkOrderClass		61.0	Active	1,354	JHANSI RAJ KETHAWATH 29/07/2024 7:20 pm	<input type="checkbox"/>

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 :

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.

The first screenshot shows the 'Reports' page in Salesforce. The left sidebar lists various report categories, and the main area displays a list of reports created by the user. The second screenshot shows the 'Assignments with Assignment ID Report' with 2 total records. The third screenshot shows the 'Assignments with WorkOrder ID Report' with 4 total records.

Report: Assignments with Assignment ID Report

	Assignment: Assignment ID	Technician ID: Technician ID
1	A-0008	T-0003
2	A-0004	T-0002

Report: Assignments with WorkOrder ID Report

	Assignment: Assignment ID	WorkOrder ID: WorkOrder ID
1	A-0008	WO-0003
2	A-0007	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".

The screenshot shows the Salesforce Reports page for a report named 'WorkOrders Status Reports'. The report is based on the 'WorkOrder' object. The table displays 3 records with columns for 'WorkOrder ID' and 'Status'.

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

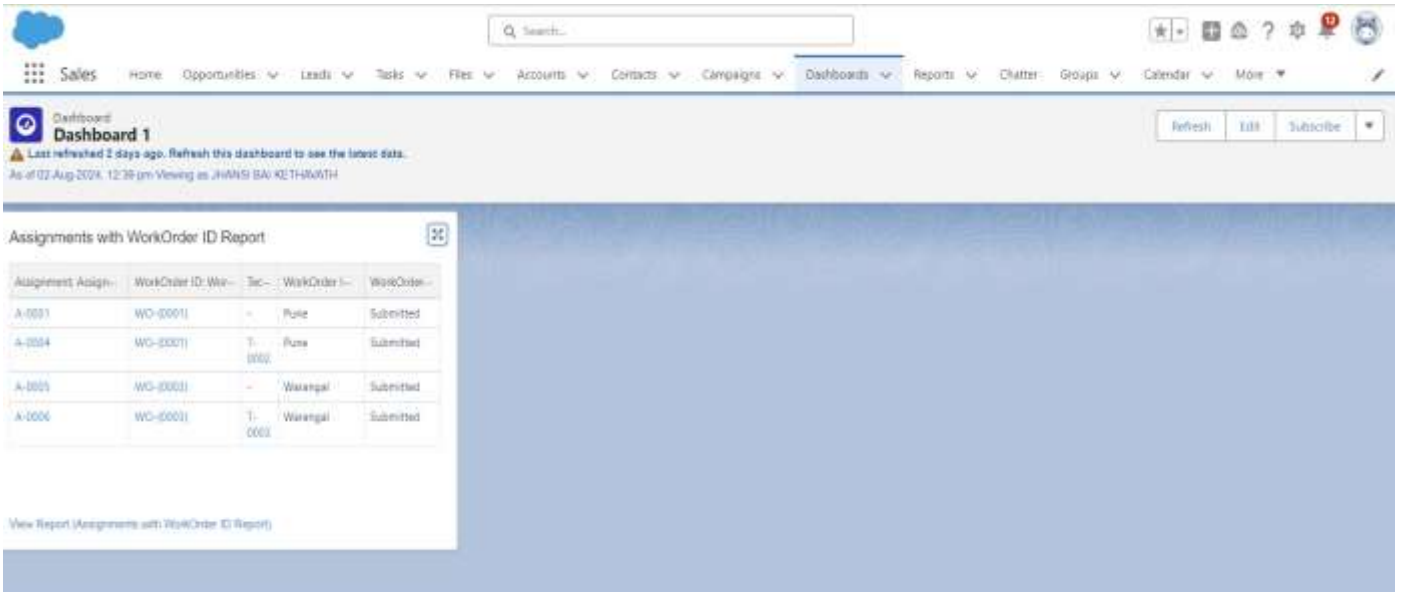
2. Create a report with report type: “Technician and Assignment Details Reports”.

The screenshot shows the Salesforce Reports page for a report named 'Technician and Assignment Details Report'. The report is based on the 'Assignment' object. The table displays 2 records with columns for 'Assignment ID' and 'Technician ID'.

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

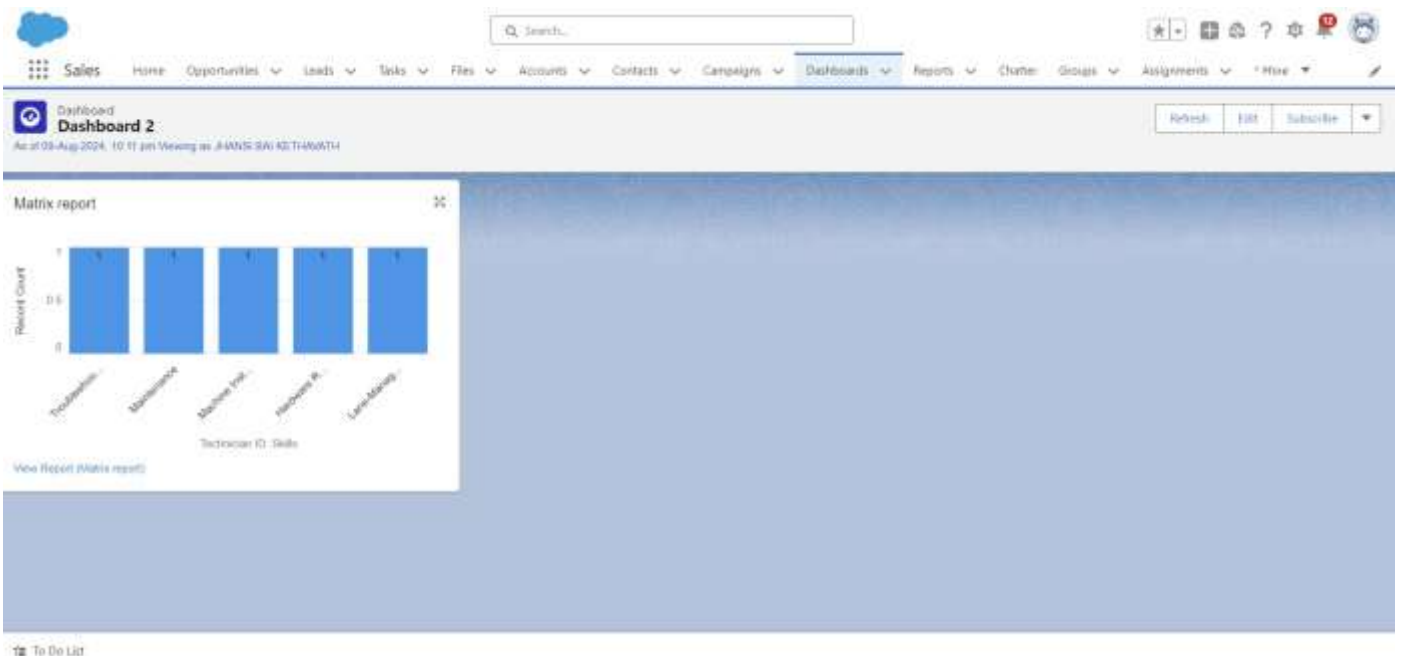
8.3 Dashboard

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.



8.4 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*



