

FIELD SERVICE WORKORDER OPTIMIZATION

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.

SetupHomeObject Manager

Object Manager

1 Items, Sorted by Label

tec

Schema Builder

Create

Label	API Name	Type	Description	Last Modified	Deployed
Technician	Technician__c	Custom Object		20/11/2024	✓

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.

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 ✓ Maintenance

✓ Description × Description Picklist ✓

✓ Location × Location Text Area (Long) ✓ Pune

✓ Priority × Priority Picklist ✓ Low

Back

Next

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

Setup | Home | Object Manager ▾

SETUP

Object Manager

1 Items. Sorted by Label

Schema Builder
Create ▾

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

Setup Home Object Manager

Object Manager
2 Items, Sorted by Label

Search Setup

assignment Schema Builder Create

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

Task 2:

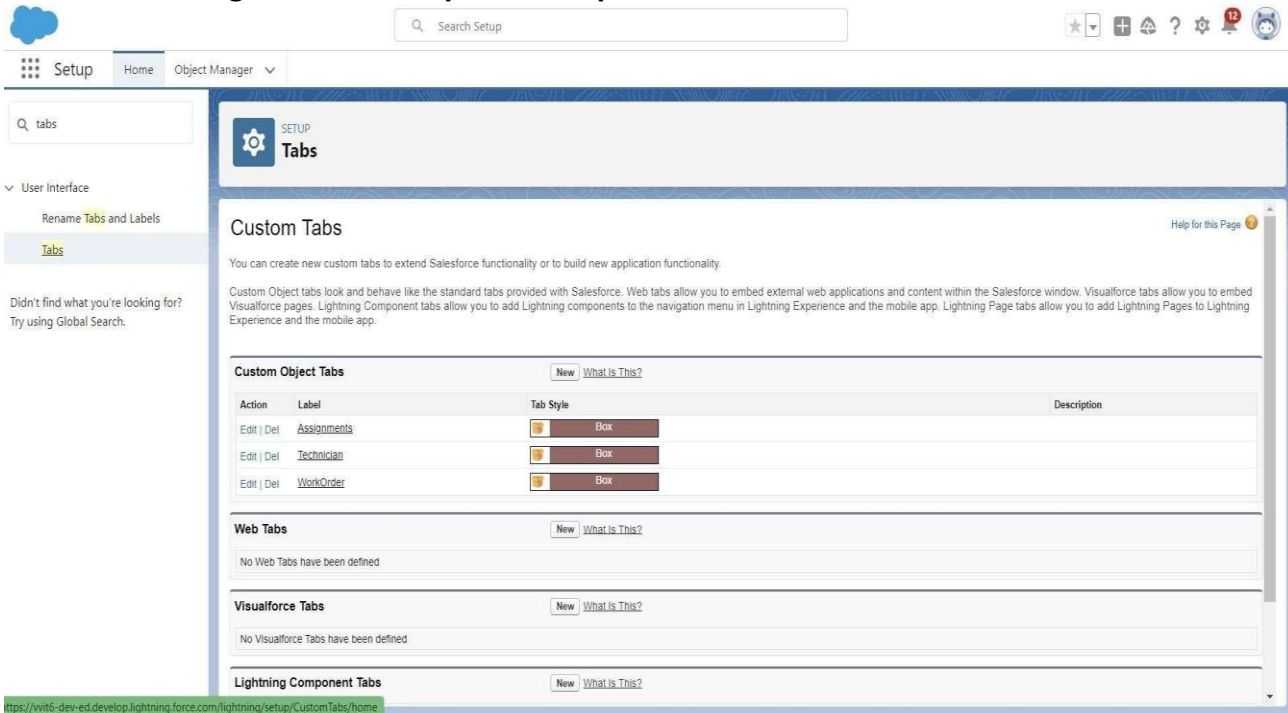
Creating a Custom Tab:

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:



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

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.

Lightning App Builder
App Settings
Pages
Field Service WorkOrder Optimization
? Help

App Settings
App Details & Branding
App Options
Utility Items (Desktop Only)
Navigation Items
User Profiles

User Profiles

Choose the user profiles that can access this app.

Available Profiles

Type to filter list...

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:

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.

Setup
Home
Object Manager

SETUP > OBJECT MANAGER
Assignment

Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Restriction Rules

Assignment Custom Field
WorkOrder ID
Back to Assignment

Validation Rules (0)

Custom Field Definition Detail
Edit
Set Field-Level Security
View Field Accessibility
Where is this used?

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	Chaitanya Lahari Bezwada, 21/11/2024, 3:34 pm	Modified By	Chaitanya Lahari Bezwada, 21/11/2024, 3:34 pm

Lookup Options

Related To	WorkOrder	Child Relationship Name	Assignments1
Related List Label	Assignments		
Required	<input type="checkbox"/>		
What to do if the lookup record is deleted?	Clear the value of this field.		

Manage your picklist values:

SETUP > OBJECT MANAGER

WorkOrder

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules

Scoping Rules

Field Dependencies [New](#) [Field Dependencies Help](#)

No dependencies defined.

Validation Rules [New](#) [Validation Rules Help](#)

No validation rules defined.

Values [New](#) [Reorder](#) [Replace](#) [Printable View](#) [Chart Colors](#) [Values Help](#)

[Delete Selected](#) [Deactivate Selected](#) [Replace Selected](#)

Action	Values	API Name	Default	Chart Colors	Modified By
Edit Del Deactivate	Pune	Pune	<input type="checkbox"/>	Assigned dynamically	Chaitanya Lahari Bezawada, 20/11/2024, 1:51 am
Edit Del Deactivate	Hyderabad	Hyderabad	<input type="checkbox"/>	Assigned dynamically	Chaitanya Lahari Bezawada, 20/11/2024, 1:51 am
Edit Del Deactivate	Nasik	Nasik	<input type="checkbox"/>	Assigned dynamically	Chaitanya Lahari Bezawada, 20/11/2024, 1:55 am
Edit Del Deactivate	Warangal	Warangal	<input type="checkbox"/>	Assigned dynamically	Chaitanya Lahari Bezawada, 20/11/2024, 1:55 am
Edit Del Deactivate	Nanded	Nanded	<input type="checkbox"/>	Assigned dynamically	Chaitanya Lahari Bezawada, 20/11/2024, 1:55 am

Inactive Values [Delete Unused Values](#) [Inactive Values Help](#)

Action	Values	API Name	Modified By
Del Activate	Value1	Value1	Chaitanya Lahari Bezawada, 20/11/2024, 1:51 am

[Back To Top](#) Always show me [more](#) records per related list

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

Setup > OBJECT MANAGER
WorkOrder

Fields & Relationships

Picklist Values Used
Active picklist values: 3 (1,000 max)
Inactive picklist values: 1 (4,000 max)

Field Dependencies [New](#) [Field Dependencies Help ?](#)
No dependencies defined.

Validation Rules [New](#) [Validation Rules Help ?](#)
No validation rules defined.

Values [New](#) [Reorder](#) [Replace](#) [Printable View](#) [Chart Colors](#) [Delete Selected](#) [Deactivate Selected](#) [Replace Selected](#) [Values Help ?](#)

Action	Values	API Name	Default	Chart Colors	Modified By
Edit Del Deactivate	Low	Low	<input type="checkbox"/>	Assigned dynamically	Chaitanya Lahari Bezawada, 20/11/2024, 1:51 am
Edit Del Deactivate	Medium	Medium	<input type="checkbox"/>	Assigned dynamically	Chaitanya Lahari Bezawada, 20/11/2024, 1:51 am
Edit Del Deactivate	High	High	<input type="checkbox"/>	Assigned dynamically	Chaitanya Lahari Bezawada, 20/11/2024, 1:55 am

Inactive Values [Delete Unused Values](#) [Inactive Values Help ?](#)

Action	Values	API Name	Modified By
Del Activate	Value1	Value1	Chaitanya Lahari Bezawada, 20/11/2024, 1:51 am

[Back To Top](#) Always show me [more](#) records per related list

Setup > OBJECT MANAGER
WorkOrder

Fields & Relationships

Field Dependencies [New](#) [Field Dependencies Help ?](#)
No dependencies defined.

Validation Rules [New](#) [Validation Rules Help ?](#)
No validation rules defined.

Values [New](#) [Reorder](#) [Replace](#) [Printable View](#) [Chart Colors](#) [Delete Selected](#) [Deactivate Selected](#) [Replace Selected](#) [Values Help ?](#)

Action	Values	API Name	Default	Chart Colors	Modified By
Edit Del Deactivate	Maintenance	Maintenance	<input type="checkbox"/>	Assigned dynamically	Chaitanya Lahari Bezawada, 20/11/2024, 1:51 am
Edit Del Deactivate	Machine Installation	Machine Installation	<input type="checkbox"/>	Assigned dynamically	Chaitanya Lahari Bezawada, 20/11/2024, 1:51 am
Edit Del Deactivate	Hardware repair	Hardware repair	<input type="checkbox"/>	Assigned dynamically	Chaitanya Lahari Bezawada, 20/11/2024, 1:55 am
Edit Del Deactivate	Troubleshoot/Debugging	Troubleshoot/Debugging	<input type="checkbox"/>	Assigned dynamically	Chaitanya Lahari Bezawada, 20/11/2024, 1:55 am
Edit Del Deactivate	Lane-Management	Lane-Management	<input type="checkbox"/>	Assigned dynamically	Chaitanya Lahari Bezawada, 20/11/2024, 1:55 am

Inactive Values [Delete Unused Values](#) [Inactive Values Help ?](#)

Action	Values	API Name	Modified By
Del Activate	Value1	Value1	Chaitanya Lahari Bezawada, 20/11/2024, 1:51 am

[Back To Top](#) 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.

The screenshot shows the Salesforce Setup interface for the 'WorkOrder' object. The 'Fields & Relationships' tab is active. The formula editor is open, showing a simple formula: 'Date (Date) =' followed by 'CreateDate'. The 'Check Syntax' button indicates no syntax errors.

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	<table><tr><th>Field Name</th><th>Datatype</th></tr><tr><td><ul style="list-style-type: none">• 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>	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)
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)					

SETUP > OBJECT MANAGER
Assignment

Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Restriction Rules
Scoping Rules

Fields & Relationships
8 Items, Sorted by Field Label

Quick Find New Deleted Fields Field Dependencies Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Assignment Date	Assignment_Date__c	Formula (Date)		<input type="checkbox"/>
Assignment ID	Name	Auto Number		<input checked="" type="checkbox"/>
Completion Date	Completion_Date__c	Formula (Date)		<input type="checkbox"/>
Created By	CreatedById	Lookup(User)		<input type="checkbox"/>
Last Modified By	LastModifiedById	Lookup(User)		<input type="checkbox"/>
Owner	OwnerId	Lookup(User/Group)		<input checked="" type="checkbox"/>
Technician ID	Technician_ID__c	Lookup(Technician)		<input checked="" type="checkbox"/>
WorkOrder ID	WorkOrder_ID__c	Lookup(WorkOrder)		<input checked="" type="checkbox"/>

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.

The screenshot shows the Salesforce Setup interface. On the left, there is a navigation menu with options like 'Setup', 'Home', 'Object Manager', 'Users', 'Profiles', 'Data', 'Feature Settings', 'Decision Explorer', 'Functions', 'Marketing', and 'Sales'. The main content area is titled 'Profiles' and shows the 'Technician' profile. It includes a 'Profile Detail' section with fields for Name, User License, Description, and Created By. Below this is a 'Page Layouts' section with a table of standard object layouts.

Standard Object Layouts	Global	Global Layout [View Assignment]	Fulfillment Order Item Tax	Fulfillment Order Item Tax Layout [View Assignment]
Email Application	Not Assigned [View Assignment]		Fulfillment Order Product	Fulfillment Order Product Layout [View Assignment]
Home Page Layout	Home Page Default [View Assignment]		Idea	Varies by Record Type [View Assignment]
Account	Account Layout [View Assignment]		Individual	Individual Layout [View Assignment]
Alternative Payment Method	Alternative Payment Method Layout [View Assignment]		Invoice	Invoice Layout [View Assignment]
Appointment Initiation	Appointment Initiation Layout		Invoice Line	Invoice Line Layout

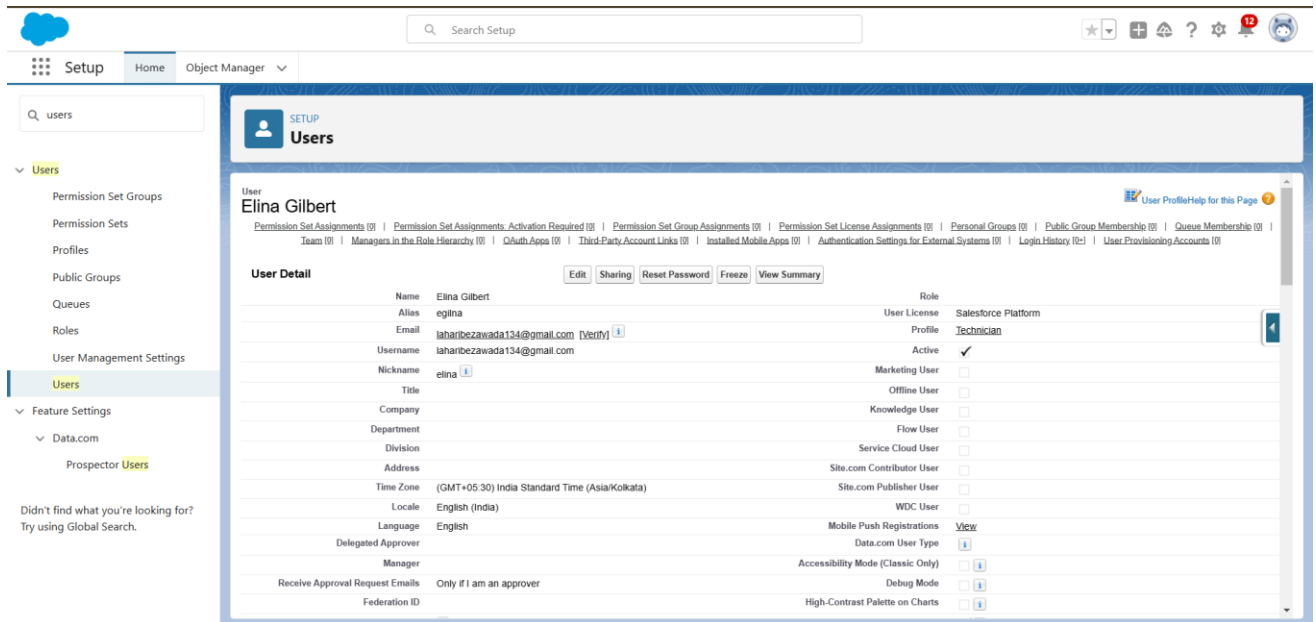
Task 6:

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



Task 7:

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>();
```

```

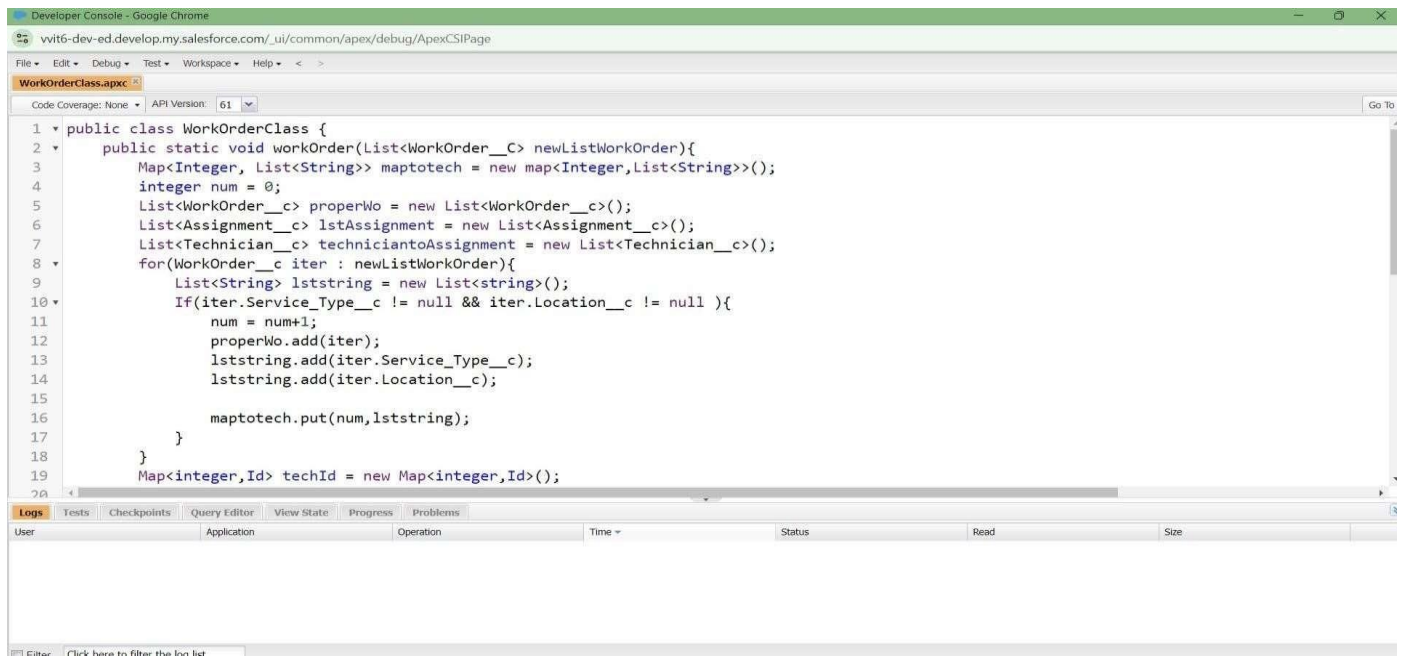
List<Assignment__c> lstAssignment = new List<Assignment__c>();
List<Technician__c> technicianToAssignment = new List<Technician__c>();
for(WorkOrder__c iter : newListWorkOrder){
    List<String> lststring = new List<string>();
    If(iter.Service_Type__c != null && iter.Location__c != null ){
        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, 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;
    }
}
}

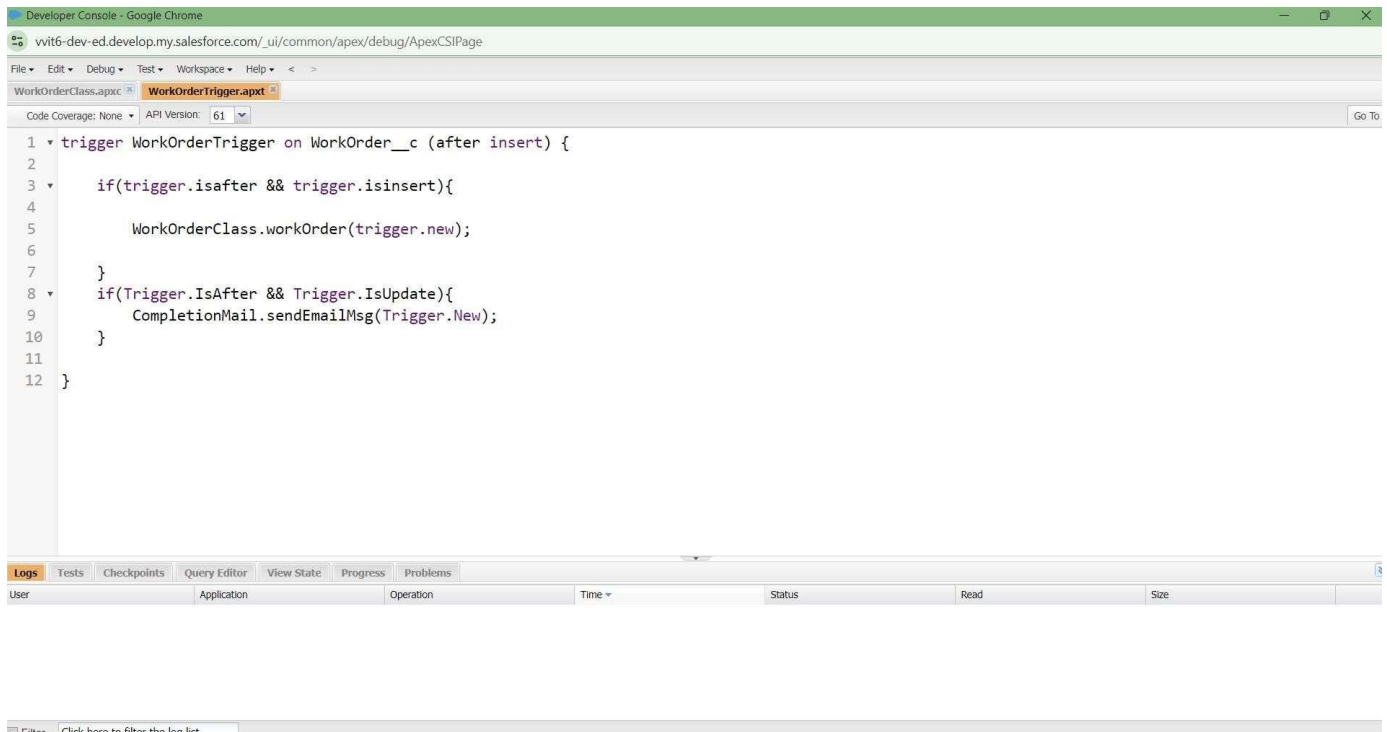
```

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



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)



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:**

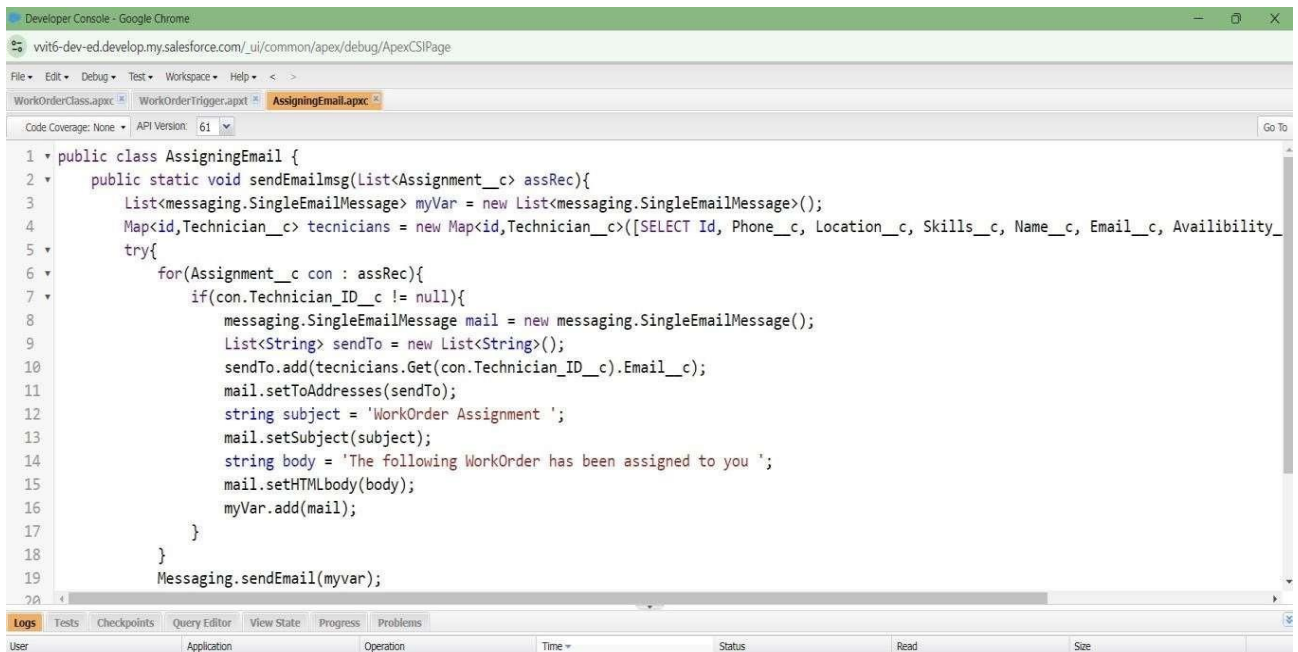
```
public class AssigningEmail {
    public static void sendEmailMsg(List<Assignment__c> assRec){
        List<messaging.SingleEmailMessage> myVar = new
        List<messaging.SingleEmailMessage>();
        Map<id,Technician__c> technicians = new Map<id,Technician__c>([SELECT Id,
        Phone__c, Location__c, 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)

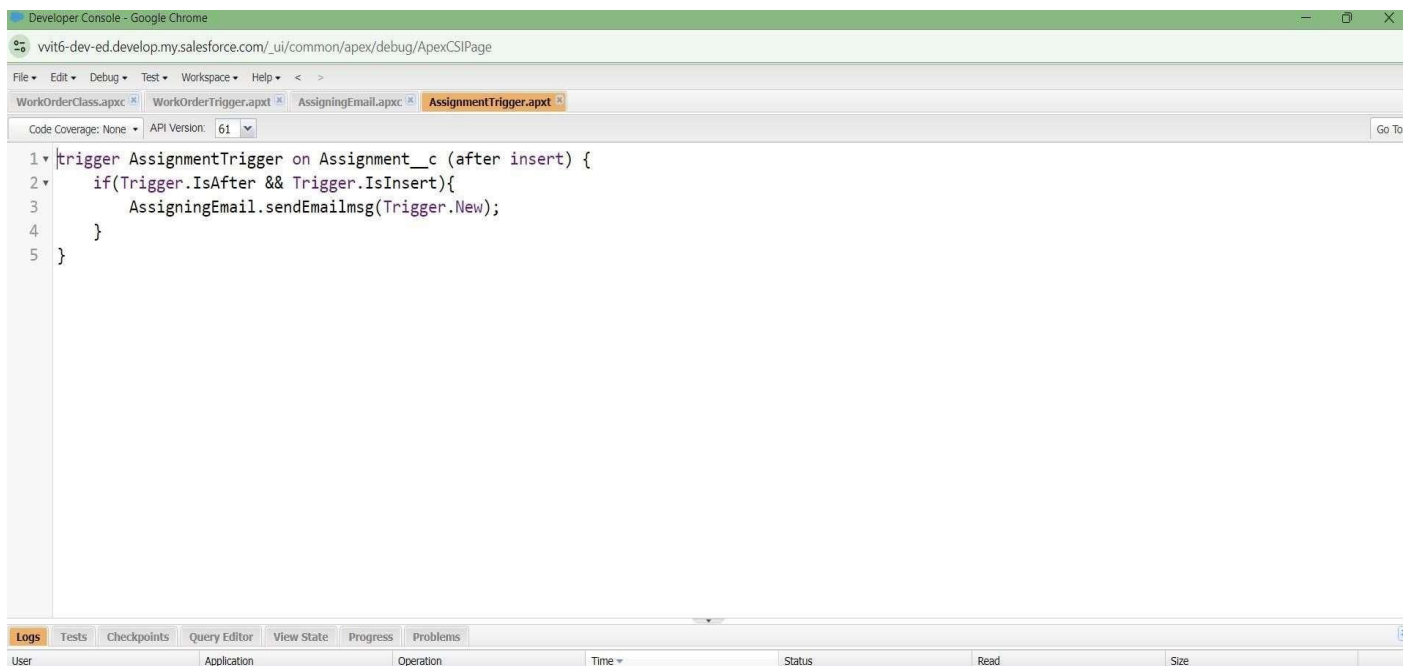


Create an Apex Trigger:

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)



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)

The screenshot shows the Salesforce Developer Console with the 'CompletionMail.apex' file open. The code defines a public class 'CompletionMail' with a static method 'sendEmailMsg' that takes a list of 'WorkOrder__c' objects. It iterates through the list, checks if the status is 'Resolved', and then creates and sends an email message with the subject 'Status Updated' and a body 'email body '.

```
1 public class CompletionMail {
2     public static void sendEmailMsg(List<WorkOrder__c> workOrderList){
3         List<messaging.SingleEmailMessage> myVar = new List<messaging.SingleEmailMessage>();
4         for(WorkOrder__c con : workOrderList){
5             if(con.Status__c == 'Resolved'){
6                 messaging.SingleEmailMessage mail = new messaging.SingleEmailMessage();
7                 List<String> sendTo = new List<String>();
8                 sendTo.add(con.Email__c);
9                 mail.setToAddresses(sendTo);
10                string subject = 'Status Updated';
11                mail.setSubject(subject);
12                string body = 'email body ' ;
13                mail.setHTMLbody(body);
14                myVar.add(mail);
15            }
16        }
17        Messaging.sendEmail(myVar);
18    }
19 }
```

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)

The screenshot shows the Salesforce Developer Console with the 'WorkOrderTrigger.apex' file open. The code defines a trigger 'WorkOrderTrigger' on the 'WorkOrder__c' object. It has two conditions: 'if(trigger.isafter && trigger.isinsert)' and 'if(trigger.IsAfter && Trigger.IsUpdate)'. The first condition calls 'WorkOrderClass.workOrder(trigger.new);' and the second condition calls 'CompletionMail.sendEmailMsg(Trigger.New);'.

```
1 trigger WorkOrderTrigger on WorkOrder__c (after insert) {
2
3     if(trigger.isafter && trigger.isinsert){
4         WorkOrderClass.workOrder(trigger.new);
5     }
6
7     if(trigger.IsAfter && Trigger.IsUpdate){
8         CompletionMail.sendEmailMsg(Trigger.New);
9     }
10 }
11
12 }
```

Create an Asynchronous Apex Class:

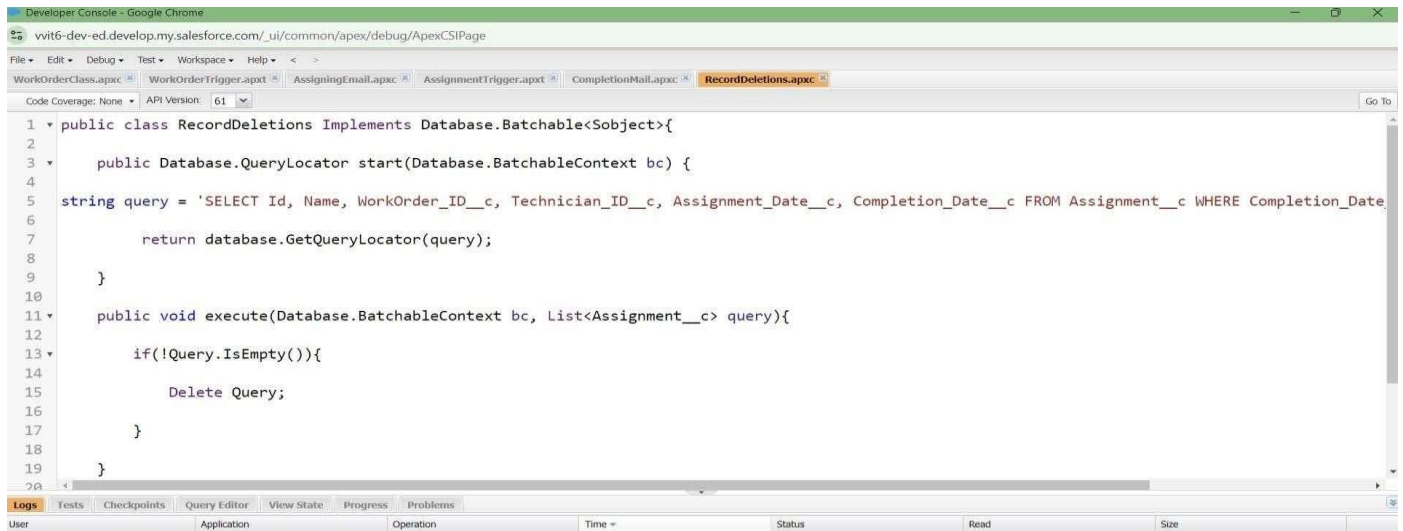
Create an Apex Class to Delete all the WorkOrder records which meets the following criteriaL

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)

The screenshot shows the Salesforce Developer Console in a Google Chrome browser. The address bar displays 'vvt6-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage'. The top navigation bar includes 'File', 'Edit', 'Debug', 'Test', 'Workspace', and 'Help'. Below this, several tabs are open: 'WorkOrderClass.apex', 'WorkOrderTrigger.apxt', 'AssigningEmail.apex', 'AssignmentTrigger.apxt', 'CompletionMail.apex', and 'RecordDeletions.apex'. The 'RecordDeletions.apex' tab is active, showing the following Apex code:

```
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__c <= :{!CompletionDate}';
6
7         return database.GetQueryLocator(query);
8     }
9
10    public void execute(Database.BatchableContext bc, List<Assignment__c> query){
11
12        if(!query.isEmpty()){
13
14            Delete query;
15        }
16    }
17 }
18
19 }
```

The bottom of the console shows a 'Logs' tab and a table with columns: 'User', 'Application', 'Operation', 'Time', 'Status', 'Read', and 'Size'.

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)

```
Developer Console - vogue-urhome
vvt6-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 CompletionMail.apxc RecordDeletions.apxc ScheduleClass.apxc
Code Coverage: None API Version: 61 Go To
1 global class ScheduleClass implements Schedulable {
2     global void execute(SchedulableContext SC) {
3         RecordDeletions delrec = new RecordDeletions();
4         database.executeBatch(delrec, 200);
5     }
6 }
```

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.

Setup Home Object Manager

Scheduled Jobs

All Scheduled Jobs

The All Scheduled Jobs page lists all of the jobs scheduled by your users. Multiple job types may display on this page. You can delete scheduled jobs if you have the permission to do so.

Percentage of Scheduled Jobs Used: 1%
You have currently used 1 scheduled Apex jobs out of an allowed organization limit of 100 active or scheduled jobs. To learn about how this limit is calculated and what contributes to it see the [Lightning Platform Apex Limits](#) topic.

View: All Scheduled Jobs Create New View

Action	Job Name	Submitted By	Submitted	Started	Next Scheduled Run	Type	Cron Trigger ID
Manage Del Pause Job	DeleteAssignmentSchedule	Bezwada, Chaitanya Labani	21/11/2024, 3:44 pm		01/12/2024, 4:00 pm	Scheduled Apex	08eNS0000048545
Del	Metalytics Data Loader Job for Org : 00DNS000006mk1	User Integration	18/11/2024, 12:02 pm	21/11/2024, 7:03 pm	22/11/2024, 7:03 pm	Autonomous Data Loader Job	08eNS00000480Ws
	Program Milestone Computation Cron Job	Process, Automated	18/11/2024, 12:02 pm	21/11/2024, 8:29 pm	22/11/2024, 1:29 am	Program Milestone Computation Cron Job	08eNS00000480Wq
	Program Status Update Cron Job	Process, Automated	18/11/2024, 12:02 pm	21/11/2024, 6:30 pm	22/11/2024, 9:30 am	Program Status Update Cron Job	08eNS00000480Wr

Apex Classes

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.

Estimate your organization's code coverage [i](#)
Compile all classes [i](#)
View: [All](#) [Create New View](#)

Action	Name ↑	Namespace Prefix	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
Edit Del Security	AssigningEmail		62.0	Active	1,226	Chaitanya Lahari Bezawada , 21/11/2024, 3:40 pm	<input type="checkbox"/>
Edit Del Security	CompletionMail		62.0	Active	801	Chaitanya Lahari Bezawada , 21/11/2024, 3:42 pm	<input type="checkbox"/>
Edit Del Security	RecordDeletions		62.0	Active	593	Chaitanya Lahari Bezawada , 21/11/2024, 3:43 pm	<input type="checkbox"/>
Edit Del Security	ScheduleClass		62.0	Active	207	Chaitanya Lahari Bezawada , 21/11/2024, 3:44 pm	<input type="checkbox"/>
Edit Del Security	WorkOrderClass		62.0	Active	1,954	Chaitanya Lahari Bezawada , 21/11/2024, 3:39 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
No records to display.				

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.

Field Service WorkO...

HomeWorkOrderAssignmentsTechnicianReportsDashboards

Search...

Star

+

🔒

?

⚙️

12

👤

Reports

Recent

4 items

Search recent reports...

New Report

New Folder

⚙️

REPORTS

Report NameDescriptionFolderCreated ByCreated OnSubscribed

Recent

Technician and Assignment Details ReportPrivate ReportsChaitanya Lahari Bezawada20/11/2024, 2:10 am

Created by Me

WorkOrders Status ReportPrivate ReportsChaitanya Lahari Bezawada20/11/2024, 2:11 am

Private Reports

Assignments with WorkOrder ID ReportPrivate ReportsChaitanya Lahari Bezawada21/11/2024, 3:46 pm

Public Reports

Assignments with Technician ID ReportPrivate ReportsChaitanya Lahari Bezawada21/11/2024, 9:33 pm

All Reports

FOLDERS

All Folders

Created by Me

Shared with Me

FAVORITES

All Favorites

Field Service WorkO...

HomeWorkOrderAssignmentsTechnicianReportsDashboards

Search...

Star

+

🔒

?

⚙️

12

👤

Field Service WorkO...

HomeWorkOrderAssignmentsTechnicianReportsDashboards

Search...

Star

+

🔒

?

⚙️

12

👤

Report: Assignments with WorkOrder ID

Assignments with WorkOrder ID Report

Enable Field Editing

Search

Add Chart

Filter

Refresh

Edit

Total Records
3

	Assignment: Assignment ID	Technician ID
1	A-(0002)	-
2	A-(0001)	-
3	A-(0000)	-

Create Reports

- Create a report with report type: "Work Orders Status Reports".

Sales

HomeOpportunitiesLeadsTasksFilesAccountsContactsCampaignsDashboardsReportsChatterGroupsCalendarMore

Search...

Star

+

🔒

?

⚙️

12

👤

Report: WorkOrder

WorkOrders Status Reports

Enable Field Editing

Search

Add Chart

Filter

Refresh

Edit

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”.

The screenshot shows the Salesforce Reports interface. The top navigation bar includes the Salesforce logo, a search bar, and various utility icons. The main navigation menu is open, showing options like Sales, Home, Opportunities, Leads, Tasks, Files, Accounts, Contacts, Campaigns, Dashboards, Reports, Chatter, Groups, Calendar, and More. The 'Reports' tab is selected, and a report titled 'Report: Assignments with Technician ID' is displayed. The report is named 'Technician and Assignment Details Report'. It shows 'Total Records' as 2. Below the record count is a table with two columns: 'Assignment: Assignment ID' and 'Technician ID: Technician ID'. The table contains two rows of data.

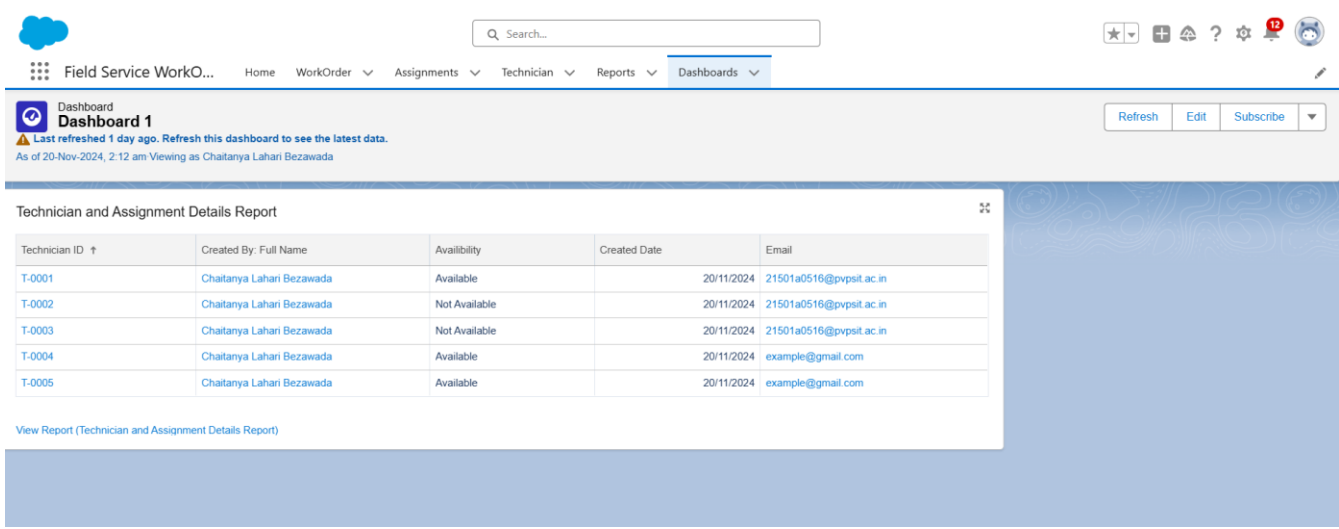
	Assignment: Assignment ID	Technician ID: Technician ID
1	A-0006	T-0003
2	A-0004	T-0002

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.

The screenshot shows the Salesforce Dashboards interface. The top navigation bar includes the Salesforce logo, a search bar, and various utility icons. The main navigation menu is open, showing options like Field Service WorkO..., Home, WorkOrder, Assignments, Technician, Reports, and Dashboards. The 'Dashboards' tab is selected. The interface shows a list of recent dashboards. The table has columns for Dashboard Name, Description, Folder, Created By, Created On, and Subscribed. There are two dashboards listed: 'Dashboard 2' and 'Dashboard 1'. The left sidebar shows navigation options for Dashboards, Folders, and Favorites.

DASHBOARDS	Dashboard Name	Description	Folder	Created By	Created On	Subscribed
Recent	Dashboard 2		Private Dashboards	Chaitanya Lahari Bezawada	20/11/2024, 2:17 am	
Created by Me	Dashboard 1		Private Dashboards	Chaitanya Lahari Bezawada	20/11/2024, 2:12 am	



Create Dashboards

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

