

FIELD SERVICE WORKORDER OPTIMIZATION

By

SOMINENI LAKSHMI PRIYANKA

21bq1a12f0@vvit.net

ABSTRACT

Field Service Work Order Optimization System

The Field Service Work Order Optimization System is designed to help a company efficiently manage installations and repair tasks. It uses a database to match work orders with skilled technicians based on factors like location, availability, and skills. This ensures that every job is assigned to the best technician. The system's algorithm prioritizes tasks, helping save time, reduce costs, and improve customer satisfaction.

Additionally, the system features automated communication, keeping technicians informed about job details in real-time. This reduces confusion and makes the workflow smoother. The built-in analytics offer insights into performance, enabling continuous improvement and better decision-making.

Key Technologies Used:

- **Salesforce Field Service:** Provides tools for scheduling, dispatching, and communication.
- **Artificial Intelligence & Machine Learning:** Predicts service demands and matches technicians to jobs based on various factors.
- **Predictive Analytics:** Forecasts service needs and identifies potential issues using historical data.
- **Internet of Things (IoT):** Uses real-time data from field devices for proactive maintenance.

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.

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.
- **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 addressing these needs, the system optimizes field service operations, reduces costs, and boosts customer satisfaction, creating a more efficient and effective work environment.

INDEX PAGE

SI No.	Module or Tasks Labels	Page No.
1	<u>Task 1: Object</u> <ul style="list-style-type: none"> ✓ Create Technician Object ✓ Create Work Order Object ✓ 1.3 Create Assignment Object 	5-7
2	<u>Task 2: Tabs</u> <ul style="list-style-type: none"> ✓ Create a custom tab 	8
3	<u>Task 3: The Lightning App</u> <ul style="list-style-type: none"> ✓ Create a Lightning App 	9-10
4	<u>Task 4: Fields & Relationship</u> <ul style="list-style-type: none"> ✓ 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 Work Order Object 	11-14
5	<u>Task 5: Profiles</u> <ul style="list-style-type: none"> ✓ Technician Profile 	16
6	<u>Task 6: Users</u> <ul style="list-style-type: none"> ✓ Create User 	17

7	<p><u>Task 7: Apex Trigger</u></p> <ul style="list-style-type: none"> ✓ Create an Apex Class ✓ Create an Apex Trigger ✓ Create an Apex Class ✓ Create an Apex Trigger ✓ Create an Apex Class ✓ Create an Apex Trigger ✓ Create an Asynchronous Apex Class ✓ Create an Apex Schedule Class ✓ Create an Schedule apex 	18-28
8	<p><u>Task 8: Reports & Dashboards</u></p> <ul style="list-style-type: none"> ✓ Report ✓ Create Reports ✓ Dashboard ✓ 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:

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
✓ Availability	×	Availability	Picklist	✓	Available
✓ Skills	×	Skills	Picklist	✓	Machine Installation

Back

Next

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



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

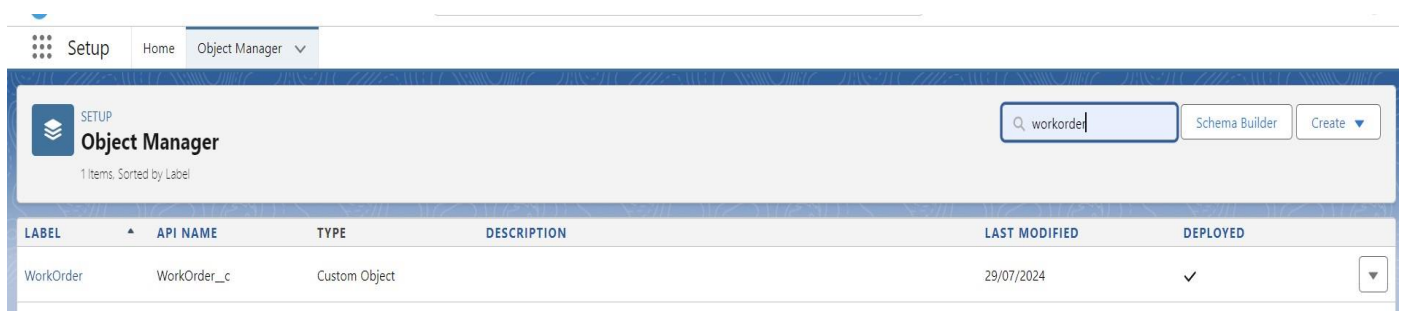
Encoding Format: Unicode (UTF8) | Values Separated By: Comma | Field Label Source: Detect from row | Field Labels Row: 1 | Import 2 rows of Data?: No, skip import | Record Name Field: WorkOrder ID

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-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



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

Object Manager

2 Items, Sorted by Label

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

Task 2:

Creating a Custom Tab

A user interface element in Salesforce that provides access to custom objects, records, or web content, enhancing navigation and organization of data within the Salesforce environment. To create a Tab:(Assignment)

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

Note: Tabs for 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:

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?](#)

<https://vvt6-dev-ed.develop.lightning.force.com/lightning/setup/CustomTabs/home>

Task 3 :

Create a Lightning App

To create a lightning app page:

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

Search Setup

Salesforce Help

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

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

Field Name WorkOrder_ID

API Name WorkOrder_ID__c

Description

Help Text

Data Owner

Field Usage

Data Sensitivity Level

Compliance Categorization

Object Name Assignment

Data Type Lookup

Created By JHANSI BALKETHAVATH 29/07/2024, 7:14 pm

Modified By JHANSI BALKETHAVATH 29/07/2024, 7:14 pm

Lookup Options

Related To WorkOrder

Related List Label Assignments

Required

Child Relationship Name Assignments

What to do if the lookup record is deleted? Clear the value of this field.

Lookup Filter

Manage your picklist values

Setup

Home

Object Manager

Search Setup

Star

Plus

Refresh

Help

Settings

12

Avatar

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

Active and inactive picklist values 6 (1,000 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
<input type="checkbox"/> Edit Del Deactivate	Value1	Value1	<input type="checkbox"/>	Assigned dynamically	JHANSI RAI KETHAVATH 29/07/2024, 7:07 pm
<input type="checkbox"/> Edit Del Deactivate	Nasik	Nasik	<input type="checkbox"/>	Assigned dynamically	JHANSI RAI KETHAVATH 29/07/2024, 7:15 pm
<input type="checkbox"/> Edit Del Deactivate	Warangal	Warangal	<input type="checkbox"/>	Assigned dynamically	JHANSI RAI KETHAVATH 29/07/2024, 7:15 pm
<input type="checkbox"/> Edit Del Deactivate	Nanded	Nanded	<input type="checkbox"/>	Assigned dynamically	JHANSI RAI KETHAVATH 29/07/2024, 7:15 pm
<input type="checkbox"/> Edit Del Deactivate	Pune	Pune	<input type="checkbox"/>	Assigned dynamically	JHANSI RAI KETHAVATH 29/07/2024, 7:15 pm
<input type="checkbox"/> Edit Del Deactivate	Hyderabad	Hyderabad	<input type="checkbox"/>	Assigned dynamically	JHANSI RAI KETHAVATH 29/07/2024, 7:15 pm

Inactive Values

Delete Unused Values

Inactive Values Help

No Inactive Values values defined.

Manage your picklist values:

Add following values to the respective fields in Wor Order object:

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

The top screenshot shows the 'Picklist Values Used' section for the WorkOrder object. It displays a table with 4 values. The bottom screenshot shows the same section with 5 values, including a formula field.

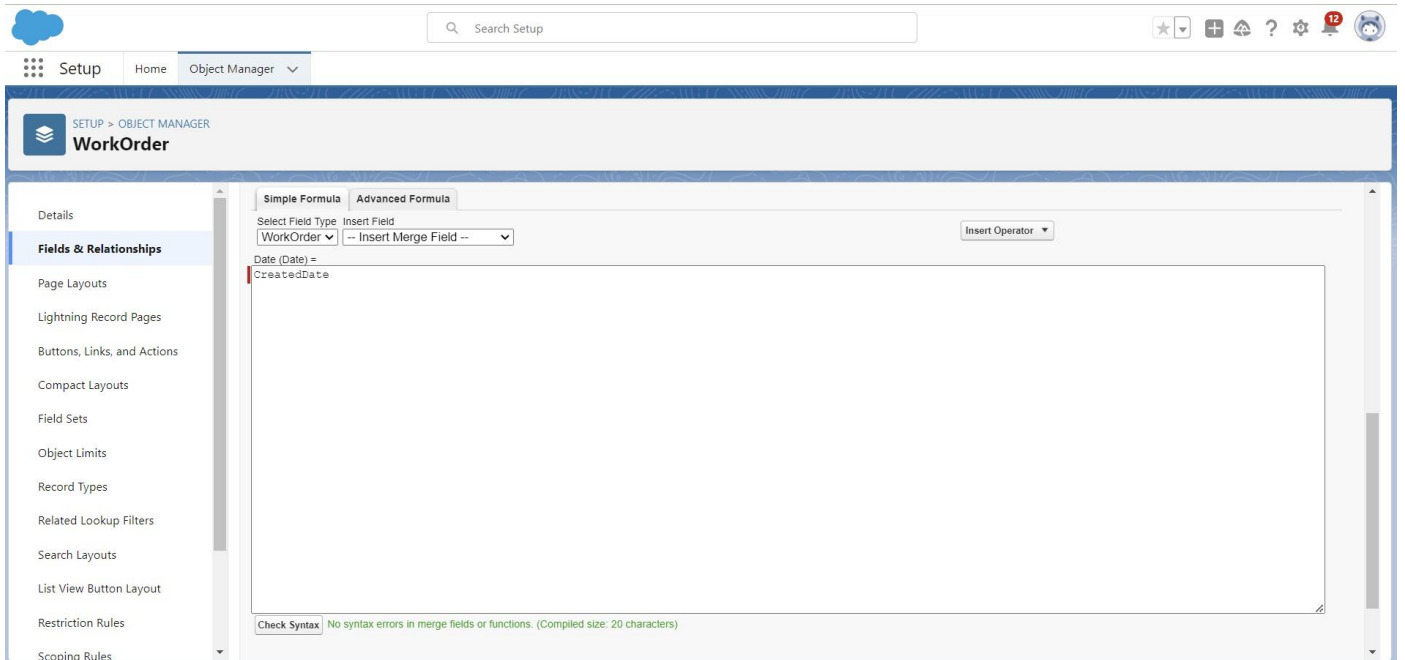
Action	Values	API Name	Default	Chart Colors	Modified By
<input type="checkbox"/> Edit <input type="checkbox"/> Del <input type="checkbox"/> Deactivate	Value1	Value1	<input type="checkbox"/>	Assigned dynamically	JHANSI BAIKETHAVATH, 29/07/2024, 7:07 pm
<input type="checkbox"/> Edit <input type="checkbox"/> Del <input type="checkbox"/> Deactivate	High	High	<input type="checkbox"/>	Assigned dynamically	JHANSI BAIKETHAVATH, 29/07/2024, 7:16 pm
<input type="checkbox"/> Edit <input type="checkbox"/> Del <input type="checkbox"/> Deactivate	Low	Low	<input type="checkbox"/>	Assigned dynamically	JHANSI BAIKETHAVATH, 29/07/2024, 7:16 pm
<input type="checkbox"/> Edit <input type="checkbox"/> Del <input type="checkbox"/> Deactivate	Medium	Medium	<input type="checkbox"/>	Assigned dynamically	JHANSI BAIKETHAVATH, 29/07/2024, 7:16 pm

Action	Values	API Name	Default	Chart Colors	Modified By
<input type="checkbox"/> Edit <input type="checkbox"/> Del <input type="checkbox"/> Deactivate	Value1	Value1	<input type="checkbox"/>	Assigned dynamically	JHANSI BAIKETHAVATH, 29/07/2024, 7:07 pm
<input type="checkbox"/> Edit <input type="checkbox"/> Del <input type="checkbox"/> Deactivate	Hardware repair Troubleshoot	Hardware repair Troubleshoot	<input type="checkbox"/>	Assigned dynamically	JHANSI BAIKETHAVATH, 29/07/2024, 7:17 pm
<input type="checkbox"/> Edit <input type="checkbox"/> Del <input type="checkbox"/> Deactivate	Debugging Lane-Management	Debugging Lane-Management	<input type="checkbox"/>	Assigned dynamically	JHANSI BAIKETHAVATH, 29/07/2024, 7:17 pm
<input type="checkbox"/> Edit <input type="checkbox"/> Del <input type="checkbox"/> Deactivate	Maintenance	Maintenance	<input type="checkbox"/>	Assigned dynamically	JHANSI BAIKETHAVATH, 29/07/2024, 7:17 pm
<input type="checkbox"/> Edit <input type="checkbox"/> Del <input type="checkbox"/> Deactivate	Machine Installation	Machine Installation	<input type="checkbox"/>	Assigned dynamically	JHANSI BAIKETHAVATH, 29/07/2024, 7:17 pm

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.



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)

SETUP > OBJECT MANAGER
Assignment

Details

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.

Setup Profiles

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 [Edit](#) [Clone](#) [View Users](#)

Name	Standard Platform User		
User License	Salesforce Platform	Custom Profile	<input type="checkbox"/>
Created By	salesforce.com, inc., 28/07/2024, 7:27 pm	Modified By	JHANSI BAL 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]
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

The screenshot shows the Salesforce Setup interface. On the left, there's a sidebar with a search bar and a list of navigation items: Users, Feature Settings, Data.com, Service, Embedded Service, and Process Automation. The 'Users' item is selected. The main content area is titled 'User Detail' for 'Elina Gilbert'. It includes buttons for 'Edit', 'Sharing', 'Reset Password', 'Freeze', and 'View Summary'. Below these, there's a table of user details:

Name	Elina Gilbert	Role
Alias	eglb	User License
Email	21bq1a05a2@vvt.net [Verify]	Profile
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
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:

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 ){
```

```

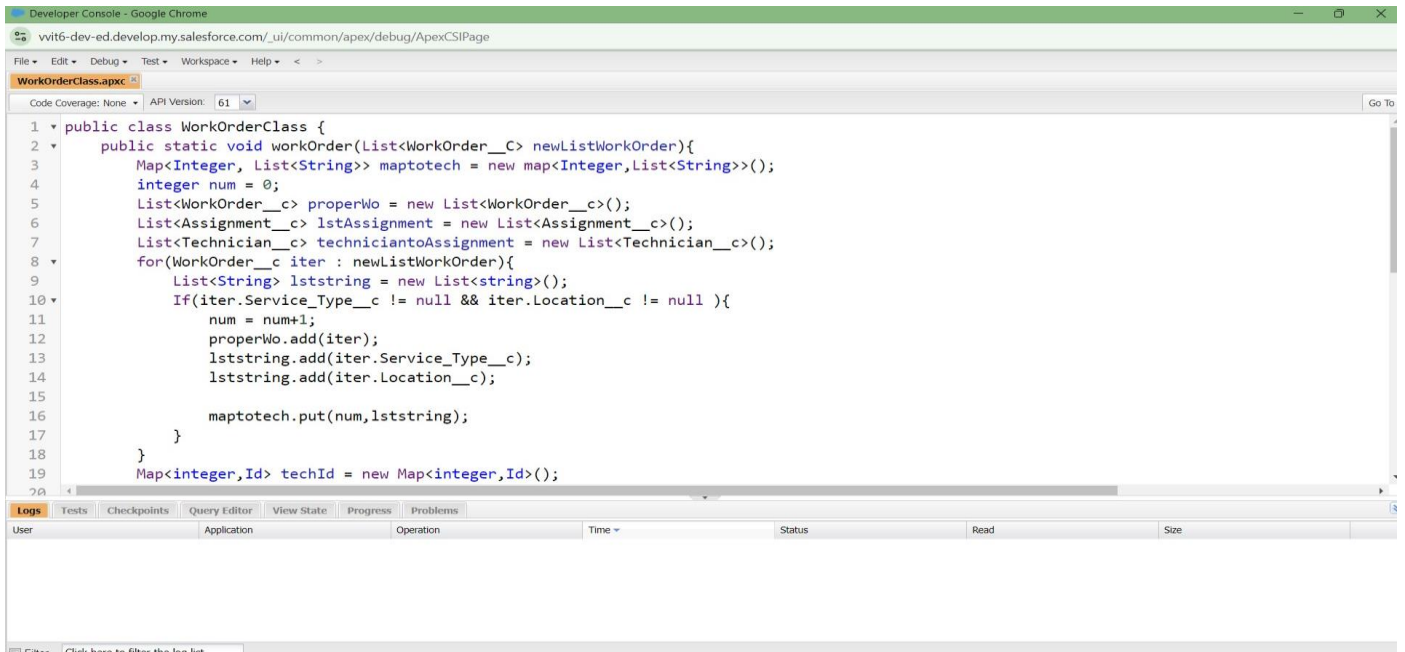
        num = num+1;
        properWo.add(iter);
        lststring.add(iter.Service_Type__c);
        lststring.add(iter.Location__c);

        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)



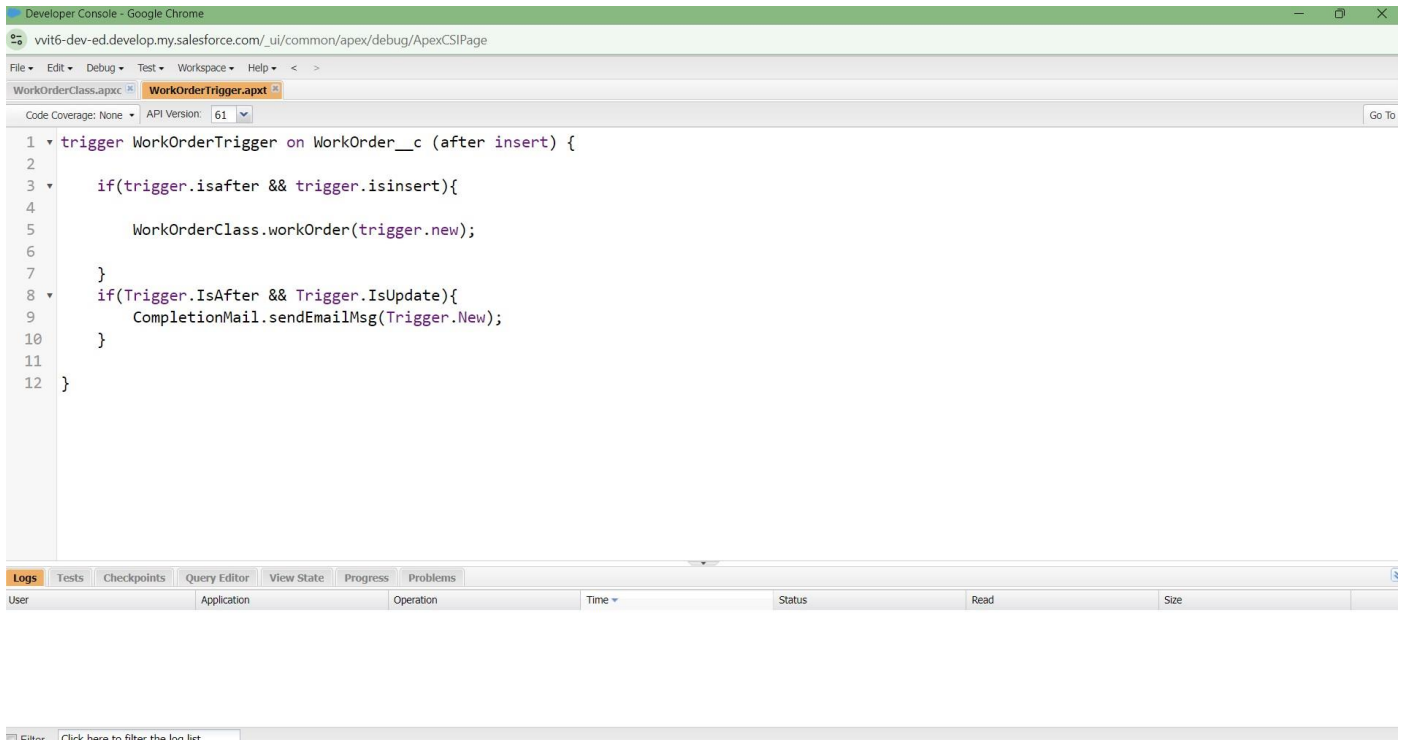
Create an Apex Trigger

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 **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)



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>();

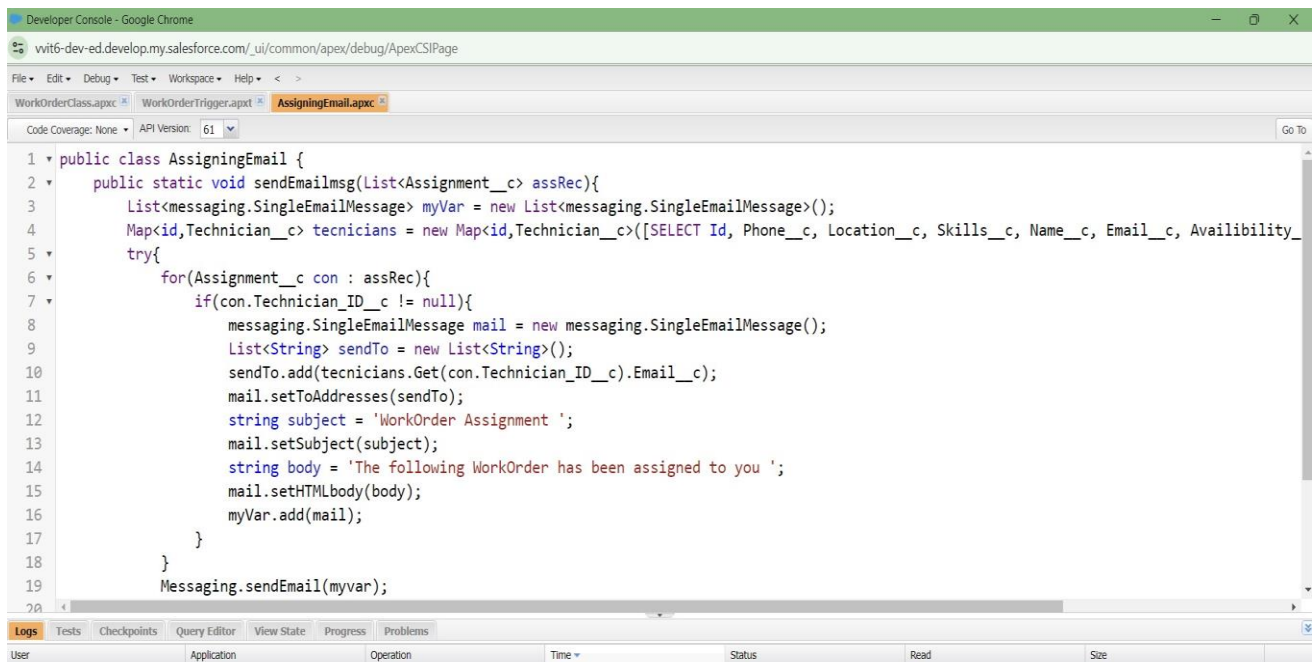
```

```

        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)

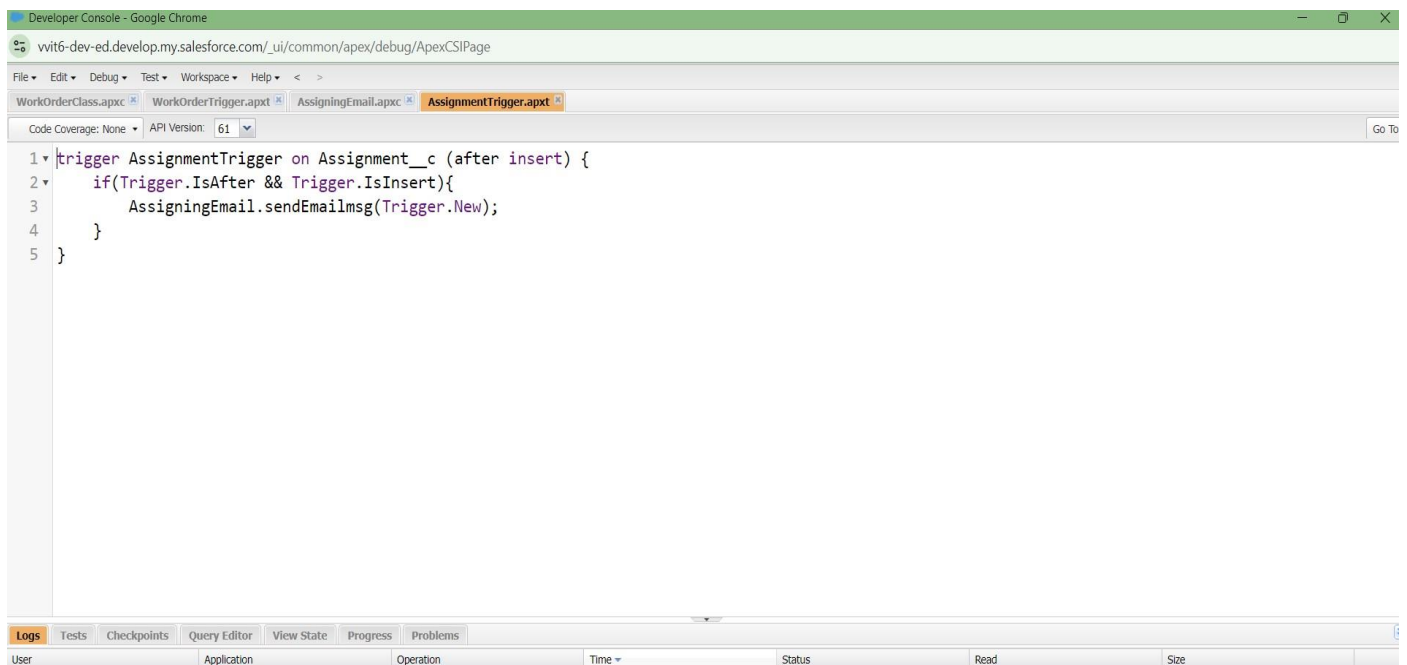


Create an Apex Trigger

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 file explorer at the top displaying several files: WorkOrderClass.apxc, WorkOrderTrigger.apxt, AssigningEmail.apxc, AssignmentTrigger.apxt, and CompletionMail.apxc. The CompletionMail.apxc file is selected and open in the editor. The code defines a public class CompletionMail with a static method sendEmailMsg that takes a List of WorkOrder__c objects and sends an email to each record where the status is 'Resolved'. The email subject is 'Status Updated' and the body is '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 file explorer at the top displaying WorkOrderClass.apxc and WorkOrderTrigger.apxt. The WorkOrderTrigger.apxt file is selected and open in the editor. The code defines a trigger on WorkOrder__c that calls WorkOrderClass.workOrder(trigger.new) when a record is inserted and CompletionMail.sendEmailMsg(trigger.New) when a record is updated.

```

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

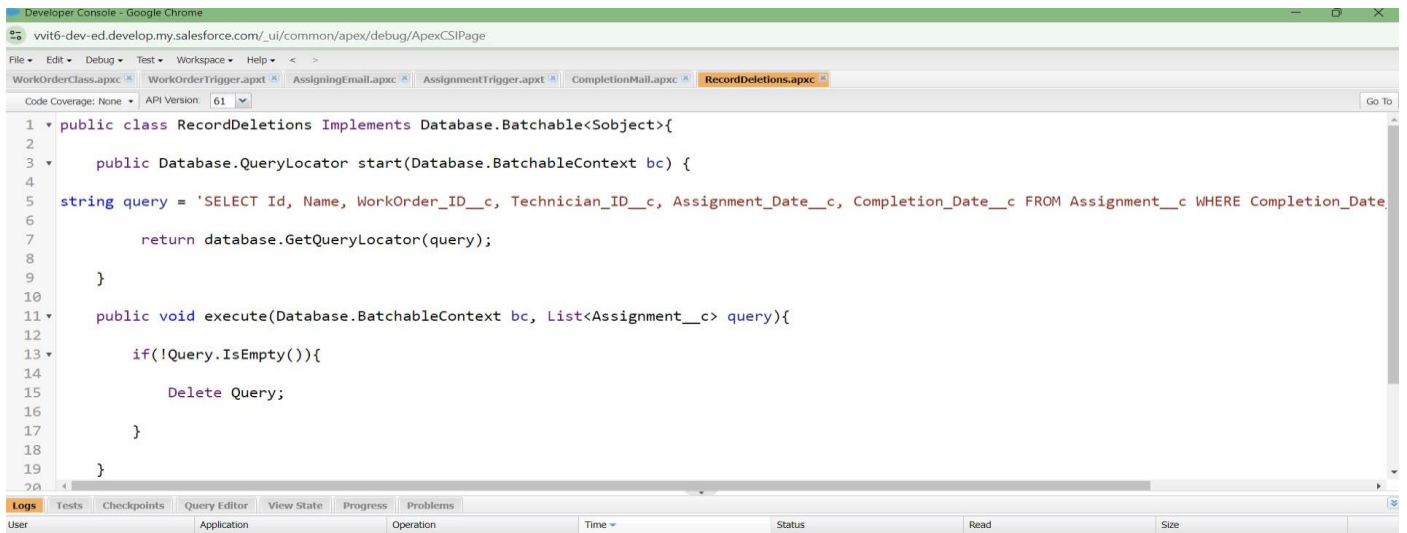
1. Completed date should be more than 30 days.

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)



```
1 public class RecordDeletions Implements Database.Batchable<Subject>{
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     }
20 }
```

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 }
```

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

apex class

Custom Code

Apex Classes

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

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: 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

Compile all classes

View: All Create New View

Action	Name	Namespace Prefix	Developer Console	New	Generate from WSDL	Run All Tests	Schedule Apex	Has Trace Flags
Edit Del Security	AssigningEmail		61.0	Active	1,226		JHANSI BAI KETHAVATH, 29/07/2024, 7:28 pm	<input type="checkbox"/>
Edit Del Security	CompletionMail		61.0	Active	801		JHANSI BAI KETHAVATH, 29/07/2024, 7:30 pm	<input type="checkbox"/>
Edit Del Security	RecordDeletions		61.0	Active	593		JHANSI BAI KETHAVATH, 29/07/2024, 7:34 pm	<input type="checkbox"/>
Edit Del Security	ScheduleClass		61.0	Active	207		JHANSI BAI KETHAVATH, 29/07/2024, 7:34 pm	<input type="checkbox"/>
Edit Del Security	WorkOrderClass		61.0	Active	1,954		JHANSI BAI KETHAVATH, 29/07/2024, 7:26 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:

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 screenshot shows the Salesforce Reports interface. The top navigation bar includes links for Sales, Home, Opportunities, Leads, Tasks, Files, Accounts, Contacts, Campaigns, Dashboards, Reports (active), Chatter, Groups, Assignments, and More. A search bar is located in the top right. Below the navigation bar, the 'Recent' reports section is displayed, showing a list of 5 items. The list includes columns for Report Name, Description, Folder, Created By, Created On, and Subscribed. The reports listed are:

Report Name	Description	Folder	Created By	Created On	Subscribed
New WorkOrder -graph		Private Reports	LAKSHMI PRIYANKA SOMINENI	26/9/2024, 10:24 pm	
WorkOrders Status Reports		Private Reports	LAKSHMI PRIYANKA SOMINENI	26/9/2024, 10:20 pm	
Technician and Assignment Details Report		Private Reports	LAKSHMI PRIYANKA SOMINENI	26/9/2024, 10:23 pm	
New Report_technician		Private Reports	LAKSHMI PRIYANKA SOMINENI	26/9/2024, 10:17 pm	
New Assignments with WorkOrder ID Report		Private Reports	LAKSHMI PRIYANKA SOMINENI	26/9/2024, 6:31 pm	

The interface also includes a sidebar on the left with options for 'Recent', 'Created by Me', 'Private Reports', 'Public Reports', 'All Reports', 'FOLDERS', 'All Folders', 'Created by Me', 'Shared with Me', and 'FAVORITES'. A 'To Do List' icon is visible at the bottom left of the interface.

Report: Assignments with Technician ID

Assignments with Assignment ID Report

Enable Field Editing Search Add Chart Filter Refresh Edit

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

Enable Field Editing Search Add Chart Filter Refresh Edit

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

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 'Sales', 'Home', 'Opportunities', 'Leads', 'Tasks', 'Files', 'Accounts', 'Contacts', 'Campaigns', 'Dashboards', 'Reports', 'Chatter', 'Groups', 'Calendar', and 'More'. The 'Reports' tab is selected. Below the navigation bar, the report title is 'Report: Assignments with Technician ID' and 'Technician and Assignment Details Report'. There are buttons for 'Enable Field Editing', 'Add Chart', and 'Edit'. The 'Total Records' section shows '2'. Below this is a table with two columns: 'Assignment: Assignment ID' and 'Technician ID: Technician ID'.

	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 is the same as the previous screenshot. The 'Dashboards' tab is selected. Below the navigation bar, the dashboard title is 'Dashboard 1'. There is a warning icon and text: 'Last refreshed 2 days ago. Refresh this dashboard to see the latest data.' Below this is the text 'As of 02-Aug-2024, 12:39 pm Viewing as JHANSI BAI KETHAVATH'. There are buttons for 'Refresh', 'Edit', and 'Subscribe'. Below this is a component titled 'Assignments with WorkOrder ID Report'. It contains a table with five columns: 'Assignment: Assign...', 'WorkOrder ID: Wor...', 'Tec...', 'WorkOrder I...', and 'WorkOrder...'. The table has four rows of data.

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.

