

FIELD SERVICE

WORKORDER OPTIMIZATION

By

Ogiboyina Alekhya

21bq1a05g6@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 ⁱ Values Separated By Field Label Source ☐ Enter manually ☒ Detect from row * Field Labels Row Import **5 rows** of Data? ⁱ ☐ No, skip import ☒ Yes, import data Record Name Field ⁱ

Fields 7 of 7 to import ☐ Hide mapped fields

| IMPORT FILE FIELD NAME | SALESFORCE FIELD NAME | SALESFORCE FIELD TYPE | ADD TO LAYOUTS ⁱ | FIELD PREVIEW |
|------------------------|---|---------------------------------------|-------------------------------------|----------------------|
| ✓ Name | <input type="text" value="Name"/> | <input type="text" value="Text"/> | <input checked="" type="checkbox"/> | Raghu |
| ✓ Phone | <input type="text" value="Phone"/> | <input type="text" value="Phone"/> | <input checked="" type="checkbox"/> | 7892341560 |
| ✓ Email | <input type="text" value="Email"/> | <input type="text" value="Email"/> | <input checked="" type="checkbox"/> | 21bq1a05a2@vvit.net |
| ✓ Location | <input type="text" value="Location"/> | <input type="text" value="Picklist"/> | <input checked="" type="checkbox"/> | Hyderabad |
| ✓ Availability | <input type="text" value="Availability"/> | <input type="text" value="Picklist"/> | <input checked="" type="checkbox"/> | Available |
| ✓ Skills | <input type="text" value="Skills"/> | <input type="text" value="Picklist"/> | <input checked="" type="checkbox"/> | Machine Installation |

Back Next

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

Search Setup

Setup Home Object Manager

SETUP Object Manager 1 Items, Sorted by Label

Search: Schema Builder Create

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

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 ¹ Values Separated By Field Label Source ☐ Enter manually ☒ Detect from row * Field Labels Row Import 2 rows of Data? ¹ ☒ No, skip import ☐ Yes, import data Record Name Field ¹

Fields 7 of 7 to import ☐ Hide mapped fields

| IMPORT FILE FIELD NAME | | SALESFORCE FIELD NAME | SALESFORCE FIELD TYPE | ADD TO LAYOUTS ¹ | FIELD PREVIEW |
|------------------------|---|---|---|-------------------------------------|------------------------|
| ✓ WorkOrder ID | ✕ | <input type="text" value="WorkOrder ID"/> | <input type="text" value="Text"/> | <input checked="" type="checkbox"/> | WO-{0001} |
| ✓ Email | ✕ | <input type="text" value="Email"/> | <input type="text" value="Email"/> | <input checked="" type="checkbox"/> | example1@workorder.com |
| ✓ Service Type | ✕ | <input type="text" value="Service Type"/> | <input type="text" value="Text"/> | <input checked="" type="checkbox"/> | Maintenance |
| ✓ Description | ✕ | <input type="text" value="Description"/> | <input type="text" value="Picklist"/> | <input checked="" type="checkbox"/> | |
| ✓ Location | ✕ | <input type="text" value="Location"/> | <input type="text" value="Text Area (Long)"/> | <input checked="" type="checkbox"/> | Pune |
| ✓ Priority | ✕ | <input type="text" value="Priority"/> | <input type="text" value="Picklist"/> | <input checked="" type="checkbox"/> | Low |

After creating the WorkOrder Custom object it looks like the below

Setup Home Object Manager

Object Manager

1 Items, Sorted by Label

| 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

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

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

Custom Tabs

You can create new custom tabs to extend Salesforce functionality or to build new application functionality.

Custom Object tabs look and behave like the standard tabs provided with Salesforce. Web tabs allow you to embed external web applications and content within the Salesforce window. Visualforce tabs allow you to embed Visualforce pages. Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app. Lightning Page tabs allow you to add Lightning Pages to Lightning Experience and the mobile app.

Custom Object Tabs [New](#) [What Is This?](#)

| Action | Label | Tab Style | Description |
|--|-----------------------------|-----------|-------------|
| Edit Del | Assignments | Box | |
| Edit Del | Technician | Box | |
| Edit Del | WorkOrder | Box | |

Web Tabs [New](#) [What Is This?](#)

No Web Tabs have been defined

Visualforce Tabs [New](#) [What Is This?](#)

No Visualforce Tabs have been defined

Lightning Component Tabs [New](#) [What Is This?](#)

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

Task 3 :

Create a Lightning App

To create a lightning app page:

- Go to the setup page --> search “app manager” in quick find --> select “app manager” --> click on New lightning App.
- 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

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

App Details & Branding

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

App Details

* App Name [?]

* Developer Name [?]

Description [?]

App Branding

Image [?]

Primary Color Hex Value [?]

Org Theme Options

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

App Launcher Preview

FS

Field Service WorkOrder O...

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

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

Navigation Items

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

Available Items

Q Type to filter list...

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

Selected Items

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

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

Note: select asset the custom object which we have created in the previous activity. 5. To Add User Profiles:

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

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 :

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.

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

Lookup Options

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

Lookup Filter

4.2 Manage your picklist values

11

Setup

Home

Object Manager

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 BAI KETHAVATH 29/07/2024, 7:07 pm |
| <input type="checkbox"/> Edit Del Deactivate | Nasik | Nasik | <input type="checkbox"/> | Assigned dynamically | JHANSI BAI KETHAVATH 29/07/2024, 7:15 pm |
| <input type="checkbox"/> Edit Del Deactivate | Warangal | Warangal | <input type="checkbox"/> | Assigned dynamically | JHANSI BAI KETHAVATH 29/07/2024, 7:15 pm |
| <input type="checkbox"/> Edit Del Deactivate | Nanded | Nanded | <input type="checkbox"/> | Assigned dynamically | JHANSI BAI KETHAVATH 29/07/2024, 7:15 pm |
| <input type="checkbox"/> Edit Del Deactivate | Pune | Pune | <input type="checkbox"/> | Assigned dynamically | JHANSI BAI KETHAVATH 29/07/2024, 7:15 pm |
| <input type="checkbox"/> Edit Del Deactivate | Hyderabad | Hyderabad | <input type="checkbox"/> | Assigned dynamically | JHANSI BAI KETHAVATH 29/07/2024, 7:15 pm |

Inactive Values

Delete Unused Values

Inactive Values Help

No Inactive Values values defined.

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 |

Setup > OBJECT MANAGER
WorkOrder

Active and inactive picklist values 4 (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 <input type="checkbox"/> Del <input type="checkbox"/> Deactivate | Value1 | Value1 | <input type="checkbox"/> | Assigned dynamically | JHANSI BAI KETHAVATH, 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 BAI KETHAVATH, 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 BAI KETHAVATH, 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 BAI KETHAVATH, 29/07/2024, 7:16 pm |

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

No inactive Values values defined.

Always show me more records per related list

Setup > OBJECT MANAGER
WorkOrder

Active and inactive picklist values 5 (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 <input type="checkbox"/> Del <input type="checkbox"/> Deactivate | Value1 | Value1 | <input type="checkbox"/> | Assigned dynamically | JHANSI BAI KETHAVATH, 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 BAI KETHAVATH, 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 BAI KETHAVATH, 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 BAI KETHAVATH, 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 BAI KETHAVATH, 29/07/2024, 7:17 pm |

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

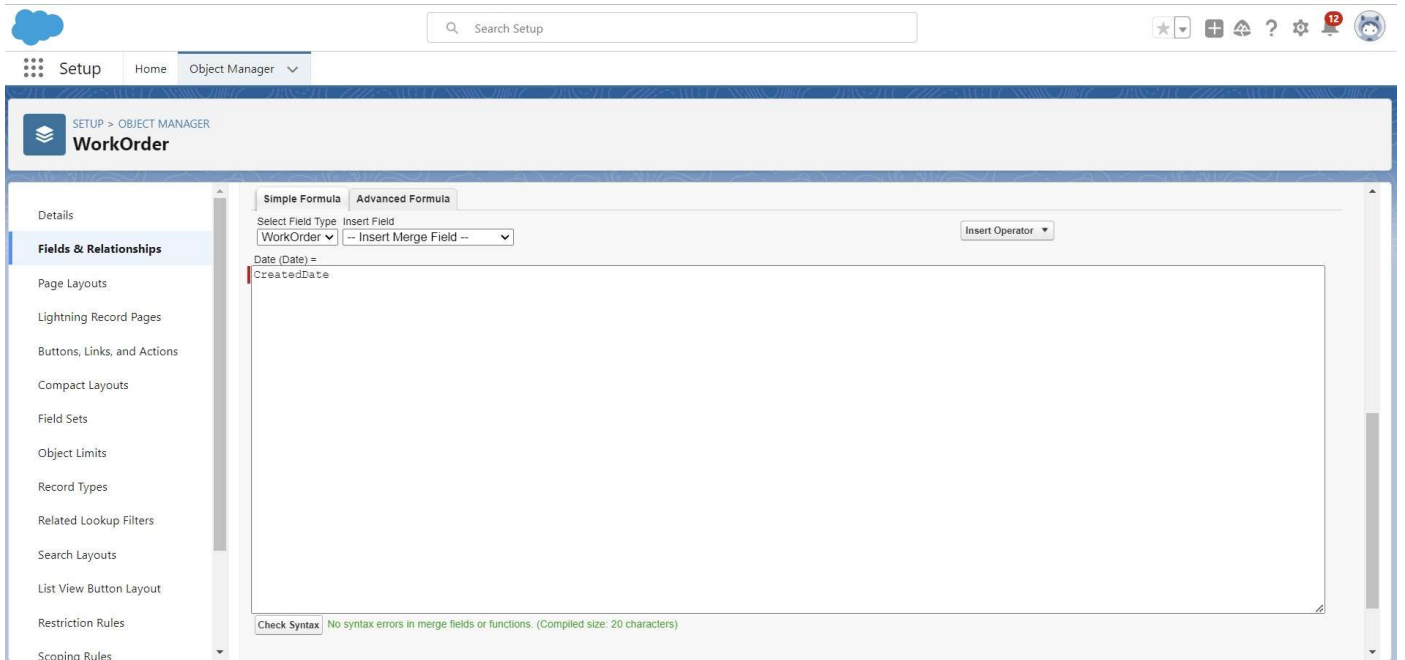
No inactive Values values defined.

Always show me more records per related list

4.4 Creating Formula Field in WorkOrder Object

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

1. Repeat steps 1 and 2 mentioned in activity 1
2. Select Data type as “Formula” and click Next.
3. Give Field Label and Field Name as “Date” and select formula return type as “Date” and click next.
4. Under Advanced Formula, write the formula and click “Check Syntax”
Formula: CreatedDate
5. Next--> Next--> Save.



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 | <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

8 Items, Sorted by Field Label

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

Task 5 :

Technician Profile

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

The screenshot shows the Salesforce Setup interface. On the left, there's a navigation menu with 'Setup', 'Home', and 'Object Manager'. Below it, a search bar contains 'profile'. The main content area is titled 'Profiles' and shows the 'Standard Platform User' profile. It includes a list of enabled permissions (e.g., Login IP Ranges, Apex Class Access, Visualforce Page Access, etc.) and a 'Profile Detail' section with buttons for 'Edit', 'Clone', and 'View Users'. The 'Page Layouts' section is expanded, showing a table of layouts for various objects. The table has two main sections: 'Standard Object Layouts' and 'Custom Object Layouts'. The 'Standard Object Layouts' section includes layouts for Global, Email Application, Home Page Layout, Account, Alternative Payment Method, and Appointment Invitation. The 'Custom Object Layouts' section includes layouts for Assignment, Technician, and WorkOrder. The 'Article Type Layouts' section is also visible at the bottom.

| Object | Layout | View Assignment |
|----------------------------|-----------------------------------|---------------------|
| 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, Prospector Users, Service, Embedded Service, Messaging for In-App and Web User Verification, and Process Automation. The main content area is titled 'Users' and shows the profile for 'Elina Gilbert'. At the top, there are links for various settings like Permission Set Assignments, Activation Required, Permission Set Group Assignments, Permission Set License Assignments, Personal Groups, Public Group Membership, Queue Membership, Team, Managers in the Role Hierarchy, OAuth Apps, Third-Party Account Links, Installed Mobile Apps, Authentication Settings for External Systems, Login History, and User Provisioning Accounts. Below these links, there's a 'User Detail' section with buttons for Edit, Sharing, Reset Password, Freeze, and View Summary. The profile information is organized into two columns. The left column contains fields like Name, Alias, Email, Username, Nickname, Title, Company, Department, Division, Address, Time Zone, Locale, Language, Delegated Approver, Manager, Receive Approval Request Emails, and Federation ID. The right column contains fields like Role, User License, Profile, Active, Marketing User, Offline User, Knowledge User, Flow User, Service Cloud User, Site.com Contributor User, Site.com Publisher User, WDC User, Mobile Push Registrations, Data.com User Type, Accessibility Mode (Classic Only), Debug Mode, and 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, Availibility__c, Name__c, Email__c FROM Technician__c]);
integer num2 = 0;
For(Technician__c T : allTechnician.values()){
    num2 = num2+1;
    if(maptotech.get(num2) != null){
        List<string> valofmap = maptotech.get(num2);
        system.debug('error 1 ----> the maptotech is empty ---> ' + maptotech.get(num2));
        if(valofMap.contains(t.Skills__c) && ValofMap.contains(t.Location__c) && t.Availibility__c ==
'Available'){
            techid.put(num2,t.Id);
        }
    }
}

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

```

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

The screenshot shows the Salesforce Developer Console with the Apex class 'WorkOrderClass.apxc' open. The code defines a public class 'WorkOrderClass' with a static method 'workOrder' that takes a list of 'WorkOrder__c' objects. The method initializes a map 'maptotech', a counter 'num', and several lists. It then iterates through the input list, adding items to the lists and the map based on service type and location. Finally, it returns a map of technician IDs.

```

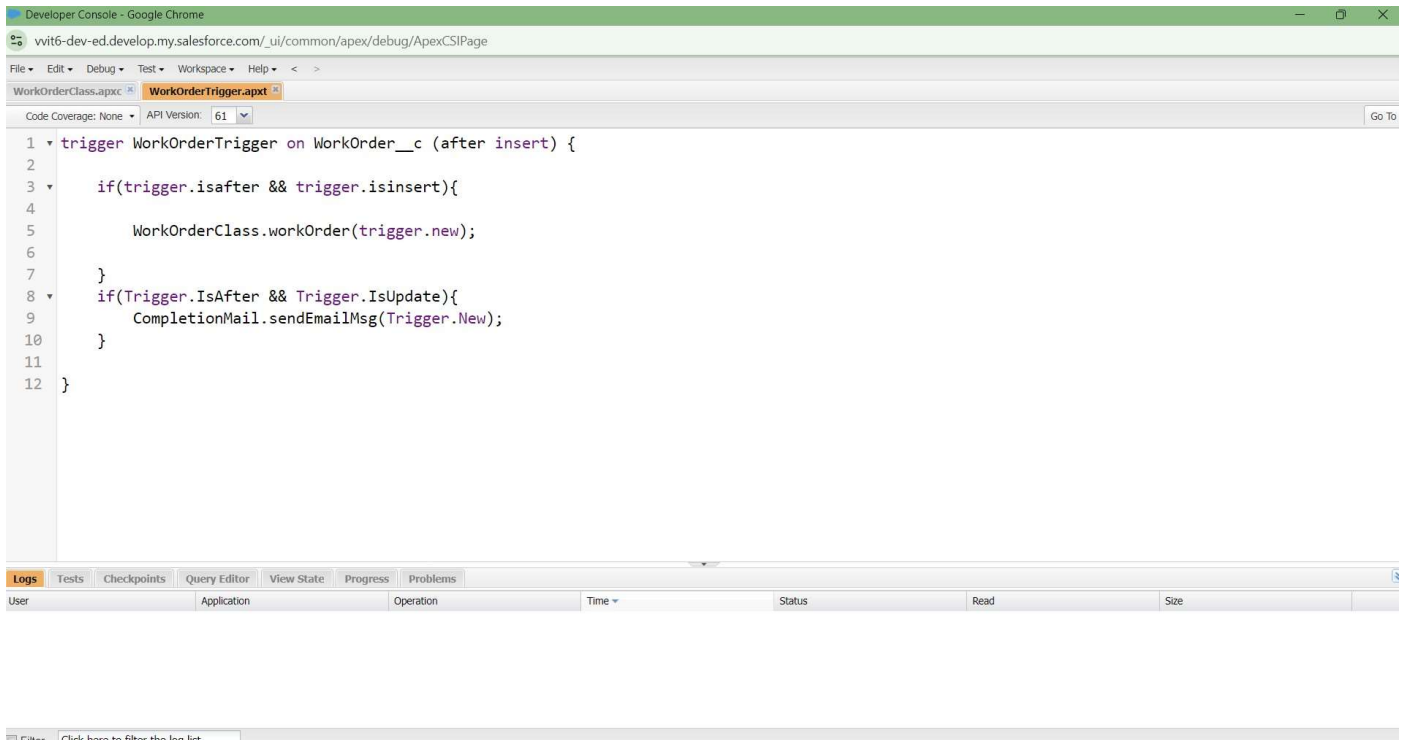
1 public class WorkOrderClass {
2     public static void workOrder(List<WorkOrder__c> newListWorkOrder){
3         Map<Integer, List<String>> maptotech = new map<Integer,List<String>>();
4         integer num = 0;
5         List<WorkOrder__c> properWo = new List<WorkOrder__c>();
6         List<Assignment__c> lstAssignment = new List<Assignment__c>();
7         List<Technician__c> technicianToAssignment = new List<Technician__c>();
8         for(WorkOrder__c iter : newListWorkOrder){
9             List<String> lststring = new List<string>();
10            If(iter.Service_Type__c != null && iter.Location__c != null ){
11                num = num+1;
12                properWo.add(iter);
13                lststring.add(iter.Service_Type__c);
14                lststring.add(iter.Location__c);
15
16                maptotech.put(num,lststring);
17            }
18        }
19        Map<integer,Id> techId = new Map<integer,Id>();
20    }
21 }

```

Below the code editor, there is a 'Logs' tab and a table with columns: User, Application, Operation, Time, Status, Read, and Size. The table is currently empty.

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)

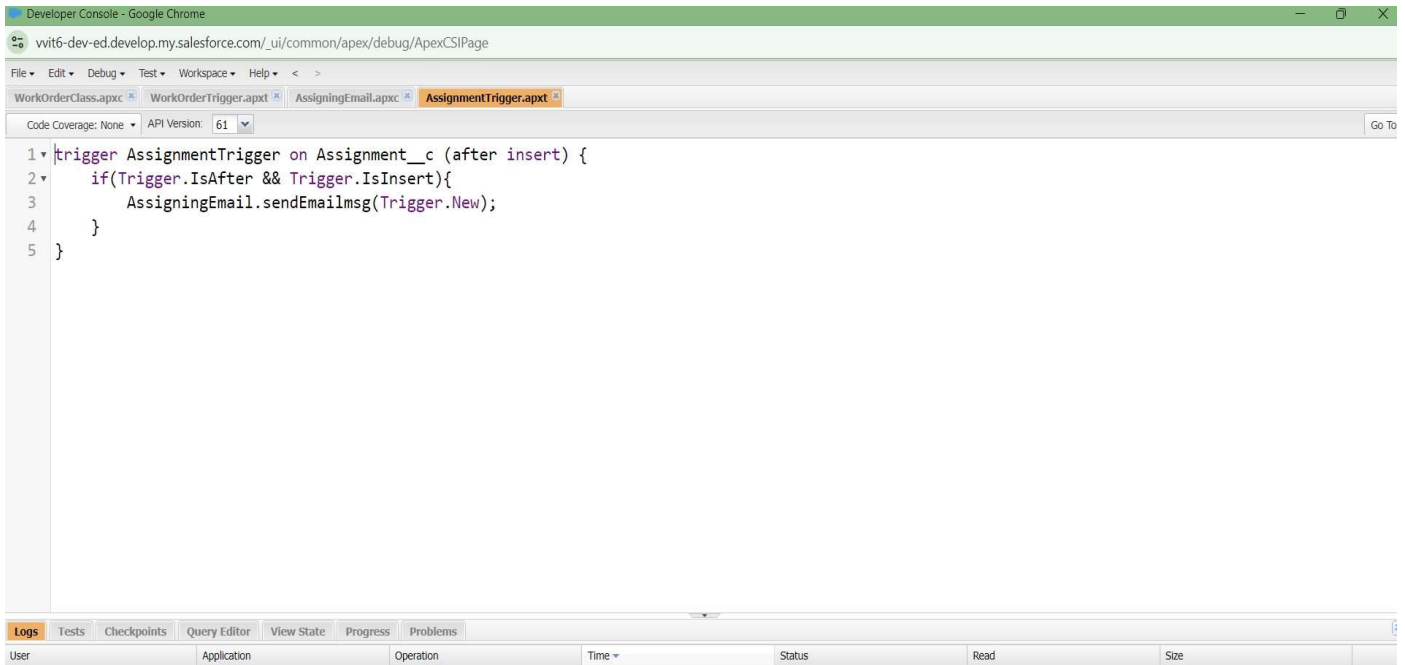
```
1 public class AssigningEmail {
2     public static void sendEmailmsg(List<Assignment__c> assRec){
3         List<messaging.SingleEmailMessage> myVar = new List<messaging.SingleEmailMessage>();
4         Map<id, Technician__c> technicians = new Map<id, Technician__c>([SELECT Id, Phone__c, Location__c, Skills__c, Name__c, Email__c, Availability__c FROM Technician__c WHERE Id IN :assRec]);
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';
            }
        }
    }
}

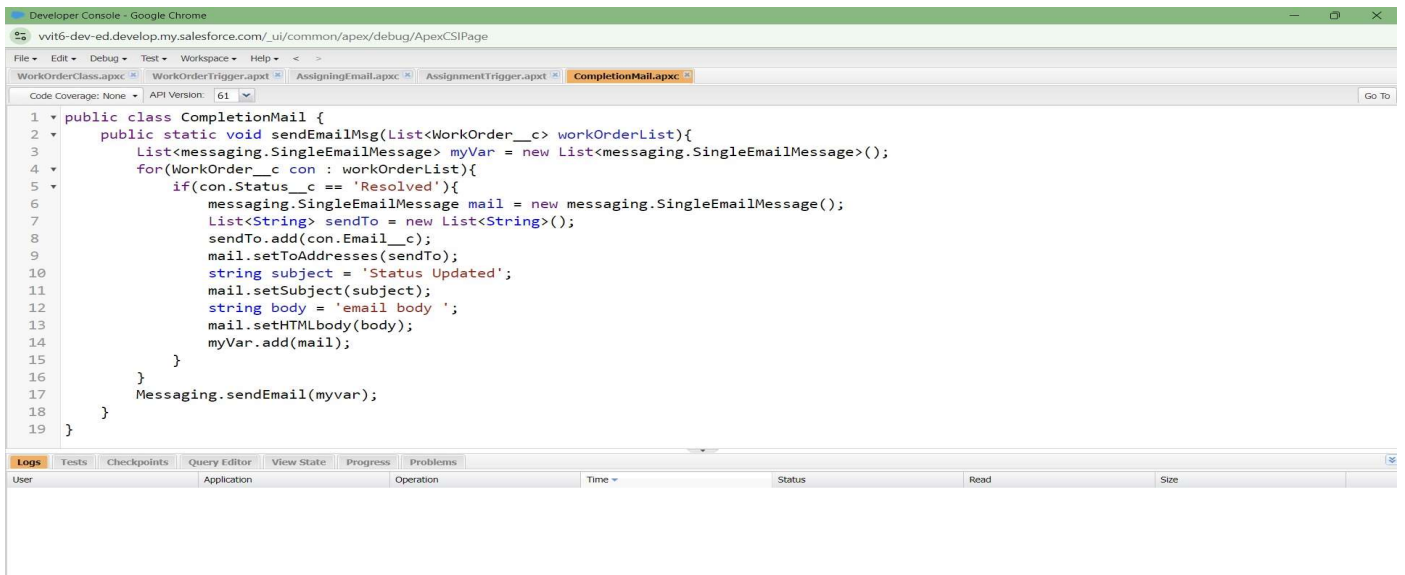
```

```

        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.

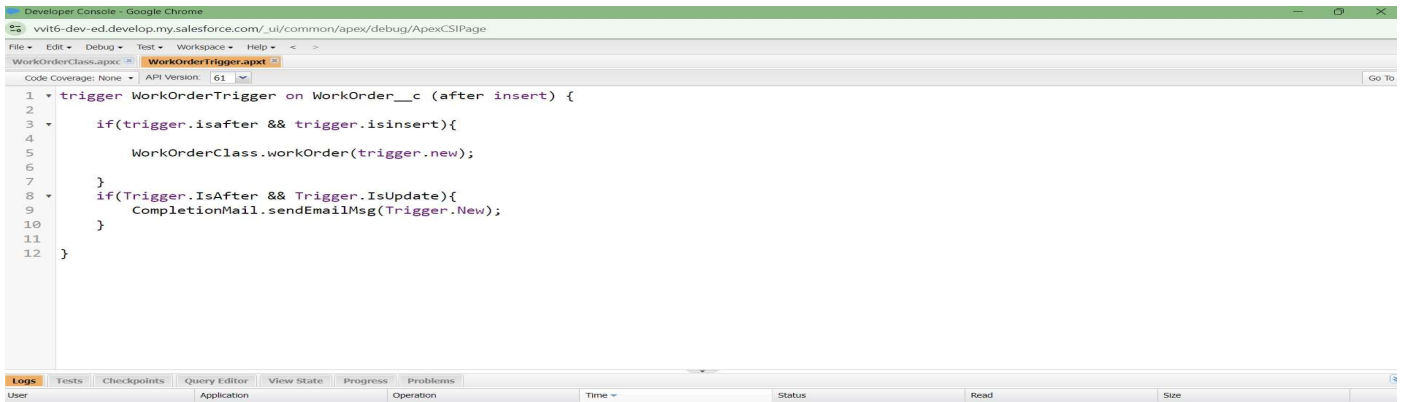
4. WorkOrderClass.workOrder(trigger.new);

```

    }
    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

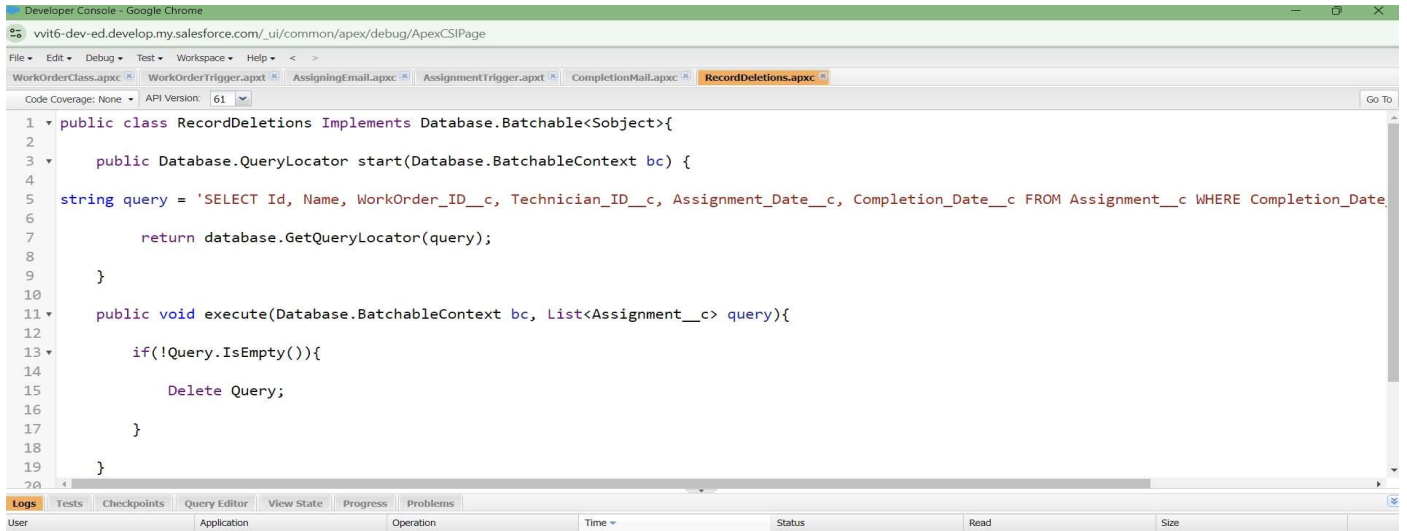
1. Completed date should be more than 30 days.
2. Status should be 'Resolved'. Create an Apex Class
1. Go to Setup --> Click on the gear icon --> Select Developer Console.
2. Then we can see the Developer console. Click on the developer console and you will navigate to a new console window.
3. To create 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)



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.

Search Setup

Setup Home Object Manager

apex class

Custom Code

Apex Classes

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

SETUP Apex Classes

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 [\[i\]](#)
[Compile all classes \[i\]](#)
View: [All](#) [Create New View](#)

| Action | Name | Namespace Prefix | Developer Console | New | Generate from WSDL | Run All Tests | Schedule Apex | Size Without Comments | Last Modified By | Has Trace Flags |
|---|---------------------------------|------------------|-------------------|--------|--------------------|---------------|---------------|-----------------------|--|--------------------------|
| Edit Del Security | AssigningEmail | | 61.0 | Active | | | | 1,226 | JHANSI BAI KETHAVATH 29/07/2024, 7:28 pm | <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 :

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 is expanded to 'Created by Me' under 'REPORTS'. The main table lists two reports: 'Assignments with Assignment ID Report' and 'Assignments with WorkOrder ID Report', both created by JHANSI BAI KETHAVATH on 29/7/2024.

The second screenshot shows the 'Assignments with Assignment ID Report' in execution. The report title is 'Report: Assignments with Technician ID Assignments with Assignment ID Report'. It shows 2 total records in a table with 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 |

The third screenshot shows the 'Assignments with WorkOrder ID Report' in execution. The report title is 'Report: Assignments with WorkOrder ID Assignments with WorkOrder ID Report'. It shows 4 total records in a table with columns 'Assignment: Assignment ID' and 'WorkOrder ID: WorkOrder ID'.

| | Assignment: Assignment ID | WorkOrder ID: WorkOrder ID |
|---|---------------------------|----------------------------|
| 1 | A-0006 | WO-[0003] |
| 2 | A-0001 | WO-[0001] |
| 3 | A-0004 | WO-[0001] |
| 4 | A-0005 | WO-[0003] |

8.2 Create Reports

1. Create a report with report type: "WorkOrders Status Reports".

The screenshot shows the Salesforce Reports page for a report titled "WorkOrders Status Reports". The report type is "WorkOrder". The interface includes a search bar, navigation tabs (Sales, Home, Opportunities, Leads, Tasks, Files, Accounts, Contacts, Campaigns, Dashboards, Reports, Chatter, Groups, Calendar, More), and a report header with options like "Enable Field Editing", "Add Chart", and "Edit". Below the header, it shows "Total Records: 3". A table displays the data:

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

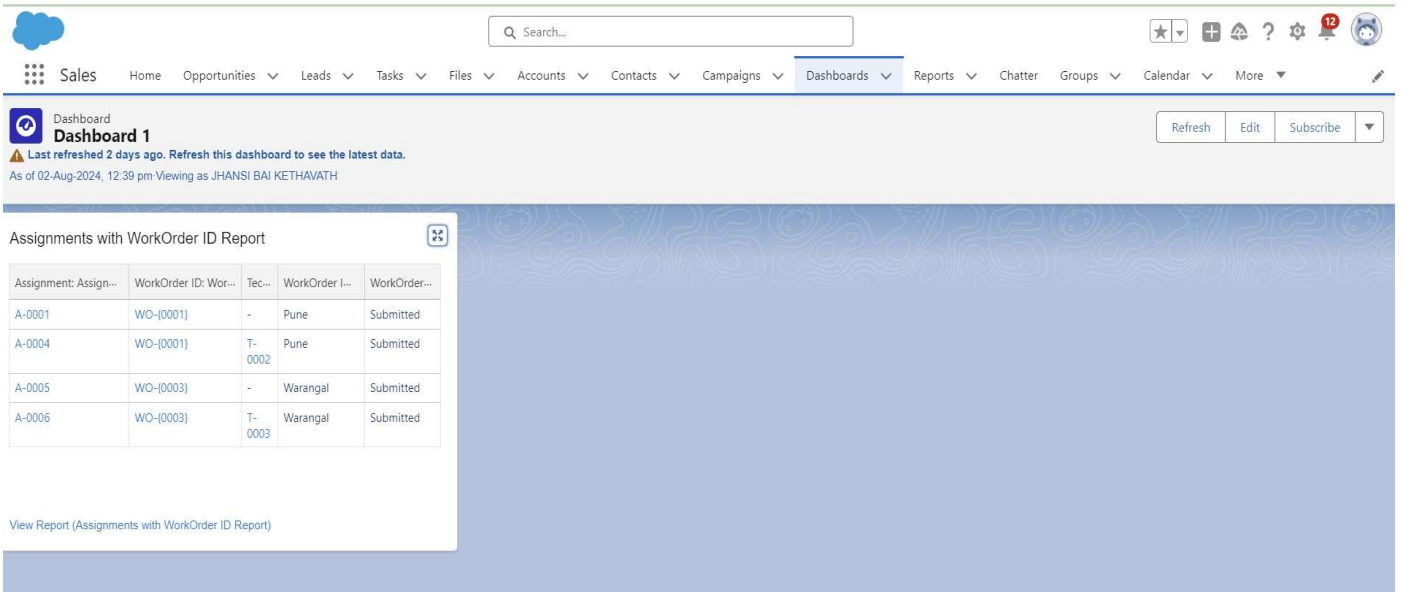
2. Create a report with report type: "Technician and Assignment Details Reports".

The screenshot shows the Salesforce Reports page for a report titled "Technician and Assignment Details Report". The report type is "Assignments with Technician ID". The interface includes a search bar, navigation tabs (Sales, Home, Opportunities, Leads, Tasks, Files, Accounts, Contacts, Campaigns, Dashboards, Reports, Chatter, Groups, Calendar, More), and a report header with options like "Enable Field Editing", "Add Chart", and "Edit". Below the header, it shows "Total Records: 2". A table displays the data:

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

8.3 Dashboard

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



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.

