

"Salesforce-Centric Project"

(Smartinternz)

FIELD SERVICE WORKORDER OPTIMIZATION

Devoloped by :

NIKHITHA VASIREDDY

Email: 2111cs010326@mallareddyuniversity.ac.in

ABSTRACT

The **Field Service Work Order Optimization** project leverages Salesforce to create an efficient and streamlined process for managing field service operations. This project involves setting up a Salesforce Developer Account and activating the necessary permissions to develop and deploy custom applications. Key custom objects—**Technician**, **WorkOrder**, and **Assignment**—are created to model the data structure required for the system. Custom tabs and a tailored Lightning App interface are designed to provide a user-friendly experience for managing these objects.

To support the solution, specific fields and relationships are defined within these objects, ensuring accurate data representation and interaction. A **Technician Profile** is created to manage permissions and access for different user types, enhancing security and user management. Additionally, custom **Apex Triggers** are developed to automate workflows, ensuring that work orders are efficiently assigned based on technician location, availability, and skill set.

The project also includes the design and implementation of **Reports & Dashboards** to provide actionable insights into the system's performance. These tools help in monitoring key metrics, facilitating data-driven decision-making, and identifying areas for continuous improvement. Overall, this Salesforce-based solution optimizes work order management, reduces operational costs, and enhances customer satisfaction through efficient resource allocation and real-time data analytics.

The project enhances team communication and collaboration through real-time updates and mobile access, enabling efficient field management of work orders. A feedback mechanism supports continuous improvement, ensuring responsive and flexible service. This solution boosts service quality, customer satisfaction, and future scalability.

INDEX PAGE

Topics	Page No
1.Salesforce <ul style="list-style-type: none">■ Creating Developer Account■ Account Activation	5 5
2.Object Creation <ul style="list-style-type: none">■ Create Technician Object■ Create Workorder Object■ Create Assignment Object	6 - 8 9 - 11 12 - 13
3.Tabs <ul style="list-style-type: none">■ Creating a Custom Tab	13 - 14
4.The Lightning App <ul style="list-style-type: none">■ Create a Lightning App	14 - 18
5.Fields & Relationship <ul style="list-style-type: none">■ Creating Lookup Field In Assignment Object■ Manage Your Picklist Values■ Manage Your Picklist Values■ Creating Formula Field In Workorder Object■ Creating Remaining Fields for the Respective Objects	18 - 22 22 - 23 23 - 25 25 - 27 27 - 30
6.Profiles <ul style="list-style-type: none">■ Technician Profile	30 - 33

INDEX PAGE

Topics	Page No
7.Users	
■ Create User	33 - 34
8.Apex Trigger	
■ Create an Apex Class	35 - 37
■ Create an Apex Trigger	37 - 38
■ Create an Apex Class	38 - 40
■ Create an Apex Trigger	40 - 41
■ Create an Apex Class	41 - 42
■ Create an Apex Trigger	42 - 43
■ Create an Asynchronous Apex Class	43 - 45
■ Create an Apex Schedule Class	45
■ Create a Schedule Apex	45 - 47
9.Reports & Dashboards	
■ Reports	47 - 49
■ Create Reports	49 - 53
1. WorkOrders Status Reports	49 - 50
2. Technician and Assignment Details Report	50 - 53
■ Dashboards	
1. Dashboard	53- 55
2. WorkOrder Status Dashboard	55 - 56
10.Conclusion	57

INTRODUCTION

TASK-1

Creating Salesforce Developer Org and Account Activation

Creating Salesforce Developer Org

1. **Visit Salesforce Developer Site:**
 - Go to the [Salesforce Developer website](#).
 - Click on "Sign Up" or "Sign Up for Free".
2. **Fill Out the Registration Form:**
 - Enter your details such as name, company, email, role, and country.
 - Choose a unique username (it must be in the form of an email address, but does not have to be a real email).
3. **Activate the Account:**
 - Salesforce will send a confirmation email to the provided email address.
 - Open the email and click on the confirmation link to activate your account.
4. **Set Up Your Password:**
 - After clicking the confirmation link, you will be prompted to set up a password and a security question.
 - Complete the setup and click "Save".
5. **Login to Developer Org:**
 - Go back to the Salesforce Developer website and log in using your username and password.

Account Activation

1. **Login to Salesforce:**
 - Use your credentials to log in to the Salesforce Developer Org.

TASK-2

Object Creation

1. Create Technician Object

1. Prepare the Technician CSV File

1. Download the Spreadsheet:

- Download the provided spreadsheet and open it in a spreadsheet editor like Microsoft Excel or Google Sheets.

2. Edit the Email Column:

- Edit the email column by providing your email for at least one or two records.

3. Save the File:

- Save the file as 'Technician.csv'.

2. Log into Salesforce and Access Setup

1. Log In:

- Go to your Salesforce Developer Org and log in with your credentials.

2. Access Setup:

- Click on the gear icon (⚙) in the upper right corner.
- Select "Setup" from the dropdown menu.

3. Navigate to Object Manager:

- Click on the "Object Manager" tab at the top of the Setup screen.

3. Create Custom Object from Spreadsheet

1. Click Create:

- In the Object Manager, click on the "Create" button.

2. Select Custom Object from Spreadsheet:

- From the dropdown menu, select "Custom Object from Spreadsheet."

3. Login with Salesforce:

- Click "Login With Salesforce."

4. Enter Salesforce Credentials:

- Enter the username and password for your Salesforce account.
- Click "Log In."

5. Grant Permissions:

- Click "Allow" to grant the necessary permissions.

4. Upload the CSV File

- 1. Click Upload:**
 - Click the "Upload" button.
- 2. Select Technician.csv File:**
 - Navigate to the location where you saved the `Technician.csv` file.
 - Select the file and upload it.
- 3. Auto-Detect Fields:**
 - Salesforce automatically detects the fields from the CSV file and populates the record data.
- 4. Set Record Name Field:**
 - Choose "Technician ID" as the Record Name field.
- 5. Verify Field Data Types:**
 - Ensure that all fields are matched with the correct data types, as per the data in the CSV.

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		Field Label Row		Import 5 rows of Data?		Record Name Field
Encoding Format	Values Separated By	Field Label Source		No, skip import	Yes, Import data	Technician ID
Unicode (UTF8)	Comma	Enter manually	1	<input type="radio"/>	<input checked="" type="radio"/>	
Fields 7 of 7 to import <input type="checkbox"/> Hide mapped fields						
IMPORT FILE FIELD NAME	SALESFORCE FIELD NAME	SALESFORCE FIELD TYPE	ADD TO LAYOUTS	FIELD PREVIEW		
✓ Technician ID	Technician ID	Text	<input checked="" type="checkbox"/>	T-0001		
✓ Name	Name	Text	<input checked="" type="checkbox"/>	Raghu		
✓ Phone	Phone	Phone	<input checked="" type="checkbox"/>	7892341560		
✓ Email	Email	Email	<input checked="" type="checkbox"/>	2111cs010326@mailareddyuniversity.ac.in.		
✓ Location	Location	Picklist	<input checked="" type="checkbox"/>	Hyderabad		
✓ Availability	Availability	Picklist	<input checked="" type="checkbox"/>	Available		
✓ Skills	Skills	Picklist	<input checked="" type="checkbox"/>	Machine Installation		

Back Next

5. Finalize the Creation

- 1. Click Next:**
 - Review the settings and mappings. Click "Next."
- 2. Enter Object Settings:**
 - Configure the object settings, such as label, plural label, and other optional settings as needed.

Create a custom object from a spreadsheet

Object properties

Almost finished! Time to define your object's attributes.

* Label

Technician

* Plural Label

Technicians

* API Name ⓘ

Technician

Object Description

> Advanced Settings

3. Click Finish:

- Click "Finish" to complete the creation process.

4. Confirmation:

- The Technician object is successfully created, and the data from the CSV file is imported into Salesforce.

Create a custom object from a spreadsheet

Nice Work!



Now you can add your object to a Lightning app. You might need to refresh the object list to see it.

Import Overview

Object Created
Technician

Fields Detected
7

Fields Created
7

Rows Detected
5

Rows Imported
5

[Import Another Object](#)

2. Create WorkOrder Object

1. Prepare the Work Order CSV File

- 1. Download the Spreadsheet:**
 - Obtain the provided Work Order spreadsheet and open it in a spreadsheet editor like Microsoft Excel or Google Sheets.
- 2. Review and Edit Data:**
 - Ensure that the data in the spreadsheet is accurate and complete. You may edit any fields if necessary.
- 3. Save the File:**
 - Save the file as `WorkOrder.csv`.

2. Log into Salesforce and Access Setup

- 1. Log In:**
 - Go to your Salesforce Developer Org and log in with your credentials.
- 2. Access Setup:**
 - Click on the gear icon () in the upper right corner.
 - Select "Setup" from the dropdown menu.
- 3. Navigate to Object Manager:**
 - Click on the "Object Manager" tab at the top of the Setup screen.

3. Create Custom Object from Spreadsheet

- 1. Click Create:**
 - In the Object Manager, click on the "Create" button.
- 2. Select Custom Object from Spreadsheet:**
 - From the dropdown menu, select "Custom Object from Spreadsheet."
- 3. Login with Salesforce:**
 - If prompted, click "Login With Salesforce."
- 4. Enter Salesforce Credentials:**
 - Enter the username and password for your Salesforce account.
 - Click "Log In."
- 5. Grant Permissions:**
 - Click "Allow" to grant the necessary permissions.

4. Upload the CSV File

- 1. Click Upload:**
 - Click the "Upload" button.

2. Select WorkOrder.csv File:

- Navigate to the location where you saved the `WorkOrder.csv` file.
- Select the file and upload it.

3. Auto-Detect Fields:

- Salesforce automatically detects the fields from the CSV file and populates the record data.

4. Set Record Name Field:

- Choose an appropriate field, such as "Work Order ID" or "Work Order Number," as the Record Name field.

5. Verify Field Data Types:

- Ensure that all fields are matched with the correct data types, as per the data in the CSV. Proper field mapping is crucial to ensure that data is imported correctly. Verify fields like:

- **Work Order Number:** Auto Number or Text
- **Description:** Text Area
- **Status:** Picklist
- **Priority:** Picklist
- **Due Date:** Date

Create a custom object from a spreadsheet

Define object and fields

Choose the data source, map fields and their types, and import field data.

CSV File Details

Encoding Format: Unicode (UTF8) Values Separated By: Comma Field Label Source: Enter manually * Field Labels Row: 1 Import 2 rows of Data? No, skip import Yes, import data Record Name Field: WorkOrder ID

Fields 7 of 7 to import Hide mapped fields

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

Back Next

5. Finalize the Creation

1. Click Next:

- Review the settings and mappings. Click "Next."

2. Enter Object Settings:

- Configure the object settings, such as label (Work Order), plural label (Work Orders), and other optional settings as needed.

Create a custom object from a spreadsheet

Object properties

Almost finished! Time to define your object's attributes.

* Label

WorkOrder - Sheet1

* Plural Label

WorkOrder - Sheet1

* API Name i

WorkOrder_Sheet1

Object Description

> Advanced Settings

- **Click Finish:**
 - Click "Finish" to complete the creation process.
- **Confirmation:**
 - The Work Order object is successfully created, and the data from the CSV file is imported into Salesforce.

Create a custom object from a spreadsheet

Nice Work!



Now you can add your object to a Lightning app. You might need to refresh the object list to see it.

Import Overview

Object Created
WorkOrder - Sheet1

Fields Detected
7

Fields Created
7

Rows Detected

Rows Imported

[Import Another Object](#)

3. Create Assignment Object

- **Access the Object Manager:**
 - Go to the **Setup** page in Salesforce.
 - In the **Quick Find** box, type **Object Manager** and click on it.
- **Create the Custom Object:**
 - In Object Manager, click on the **Create** button.
 - Select **Custom Object**.
- **Enter Object Details:**
 - **Label Name:** Enter Assignment.
 - **Plural Label Name:** Enter Assignments.
- **Set Record Name Label and Format:**
 - **Record Name:** Enter Assignment ID.
 - **Data Type:** Select Auto Number.
 - **Display Format:** Enter A-{0000}.

The screenshot shows the 'New Custom Object' page in the Salesforce Setup interface. The 'Custom Object Definition Edit' section contains the following information:

- Custom Object Information:** Label: Example: Account; Plural Label: Example: Accounts; Starts with vowel sound:
- The Object Name is used when referencing the object via the API:** Object Name: Example: Account; Description:
- Context Sensitive Help Setting:** Open the standard Salesforce.com Help & Training window (radio button selected) or Open a window using a Visualforce page.
- Content Name:**
- Enter Record Name Label and Format:** The Record Name appears in page layouts, key lists, module, and search results. For example, the Record Name for Account is "Account Name" and for Case it is "Case Number". Note that the Record Name field is always called "Name" when referenced via the API.
 - Record Name: Example: Account Name
 - Data Type: Warning: If you plan to insert a high volume of records in this object, via the API for example, use the Text data type.
 - Display Format: Example: A-{0000} What is This?
 - Starting Number:
- Optional Features:** Allow Reports, Allow Activities, Allow Chatter, Allow in Chatter Groups, Enable Licensing, Allow Sharing.
- Object Classification:** When these settings are enabled, this object is classified as an Enterprise Application object. When these settings are disabled, this object is classified as a Light Application object. Learn more.

- **Configure Optional Features:**
 - Check the boxes for **Allow Reports** and **Allow Search** to enable these features.
- **Save the Object:**
 - Click on **Save** to create the custom object.

The screenshot shows the Salesforce Setup interface for managing objects. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main title is 'Assignment' under 'OBJECT MANAGER'. On the left, a sidebar lists various object configuration tabs. The 'Details' tab is active, showing the following details for the Assignment object:

- Description
- API Name: Assignment__c
- Singular Label: Assignment
- Plural Label: Assignments
- Enable Reports:
- Track Activities
- Track Field History
- Deployment Status: Deployed
- Help Settings: Standard salesforce.com Help Window

Buttons at the bottom right of the main area are 'Edit' and 'Delete'.

TASK-3

Tabs

Creating a Custom Tab

- **Navigate to Tabs Setup:**
 - Go to the **Setup** page in Salesforce.
 - In the **Quick Find** bar, type **Tabs** and click on **Tabs**.
- **Create a New Custom Object Tab:**
 - Under **Custom Object Tabs**, click on **New**.
- **Select the Assignment Object:**
 - In the **Object** dropdown, select **Assignment**.
 - Choose a **Tab Style** that you prefer.
- **Assign the Tab to Profiles:**
 - Click **Next** to move to the Add to Profiles page.
 - Keep the default settings to assign the tab to all profiles.
- **Add the Tab to Custom Apps:**
 - Click **Next** to move to the Add to Custom Apps page.
 - Keep the default settings to add the tab to all custom apps.
- **Save the Tab:**
 - Click **Save** to complete the creation of the custom tab for the "Assignment" object.

The screenshot shows the Salesforce Setup interface with the 'Tabs' page selected. A new custom object tab for 'Assignment' is being created, set to 'Guitar' style.

Tabs for the **WorkOrder** and **Technician** objects are created automatically in Salesforce.

The screenshot shows the Salesforce Setup interface with the 'Custom Tabs' page selected. It lists three custom object tabs:

Action	Label	Tab Style	Description
Edit Del	Assignments	Guitar	
Edit Del	Technicians	Box	
Edit Del	WorkOrder - Sheet1	Box	

TASK-4

The Lightning App

Create a Lightning App

1. Access App Manager:

- Go to the **Setup** page in Salesforce.
- In the **Quick Find** bar, type **App Manager** and select it.

2. Create a New Lightning App:

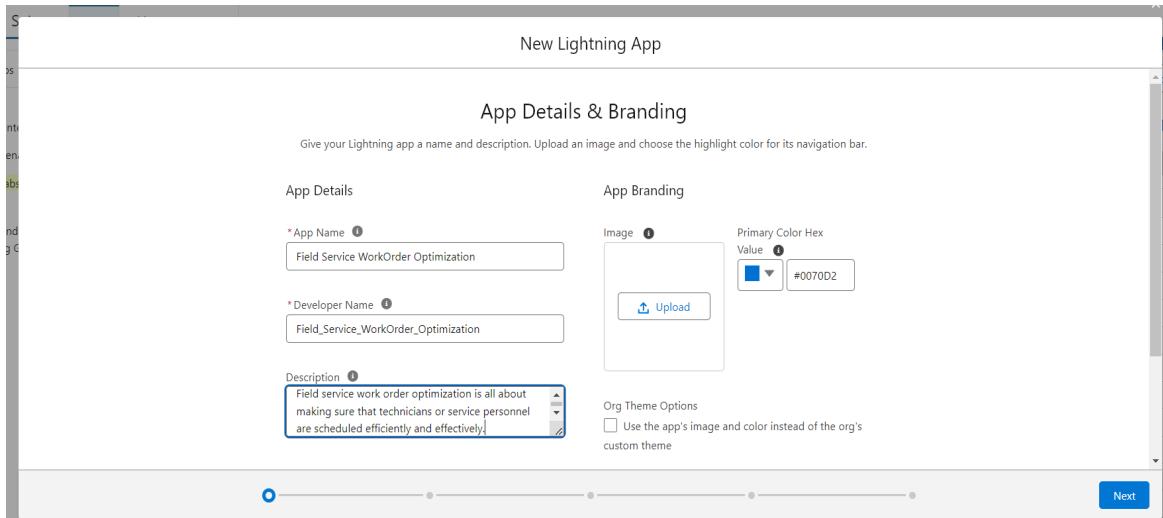
- Click on **New Lightning App**.

The screenshot shows the 'Lightning Experience App Manager' page. At the top, there's a search bar with 'app manager' typed in. Below the search bar are two buttons: 'New Lightning App' and 'New Connected App'. The main area is a table listing 22 items. The columns are: App Name, Developer Name, Description, Last Modified Date, App Type, and Visibility. The table includes rows for various apps like All Tabs, Analytics Studio, App Launcher, Automation, Bolt Solutions, Community, Content, Data Manager, Digital Experiences, Lightning Usage App, Marketing CRM Classic, Platform, Queue Management, and Sales. The 'Last Modified Date' column shows all entries were modified on 27/07/2024, 9:29 pm. The 'App Type' column indicates some are 'Classic' and others are 'Lightning'. The 'Visibility' column has checkmarks in some rows.

App Name	Developer Name	Description	Last Modified Date	App Type	Visibility
All Tabs	AllTabSet		27/07/2024, 9:29 pm	Classic	
Analytics Studio	Insights	Build CRM Analytics dashboards and apps	27/07/2024, 9:29 pm	Classic	
App Launcher	AppLauncher	App Launcher tabs	27/07/2024, 9:29 pm	Classic	
Automation	FlowsApp	Automate business processes and repetitive tasks.	27/07/2024, 9:32 pm	Lightning	
Bolt Solutions	LightningBolt	Discover and manage business solutions designed for your industry.	27/07/2024, 9:31 pm	Lightning	
Community	Community	Salesforce CRM Communities	27/07/2024, 9:29 pm	Classic	
Content	Content	Salesforce CRM Content	27/07/2024, 9:29 pm	Classic	
Data Manager	DataManager	Use Data Manager to view limits, monitor usage, and manage recipes.	27/07/2024, 9:29 pm	Lightning	
Digital Experiences	SalesforceCMS	Manage content and media for all of your sites.	27/07/2024, 9:29 pm	Lightning	
Lightning Usage App	LightningInstrumentation	View Adoption and Usage Metrics for Lightning Experience	27/07/2024, 9:29 pm	Lightning	
Marketing CRM Classic	Marketing	Track sales and marketing efforts with CRM objects.	27/07/2024, 9:29 pm	Classic	
Platform	Platform	The fundamental Lightning Platform	27/07/2024, 9:29 pm	Classic	
Queue Management	QueueManagement	Create and manage queues for your business.	27/07/2024, 9:29 pm	Lightning	
Sales	Sales	The world's most popular sales force automation (SFA) solution	27/07/2024, 9:29 pm	Classic	

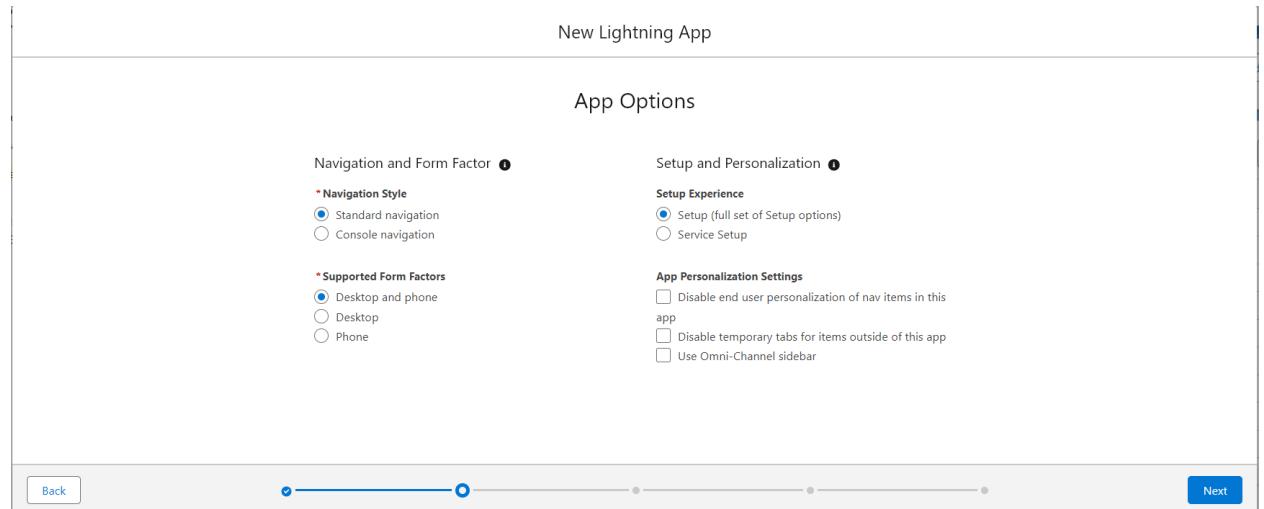
3. Fill in App Details and Branding:

- **App Name:** Enter Field Service WorkOrder Optimization.
- **Developer Name:** This field will auto-populate.
- **Description:** Provide a meaningful description of the app.
- **Image:** (Optional) Upload an image if desired, though it's not mandatory.
- **Primary Color Hex Value:** Leave this as the default value.
- Click **Next** to proceed.



4. App Options:

- On the **App Options** page, keep the default settings.
- Click **Next**.



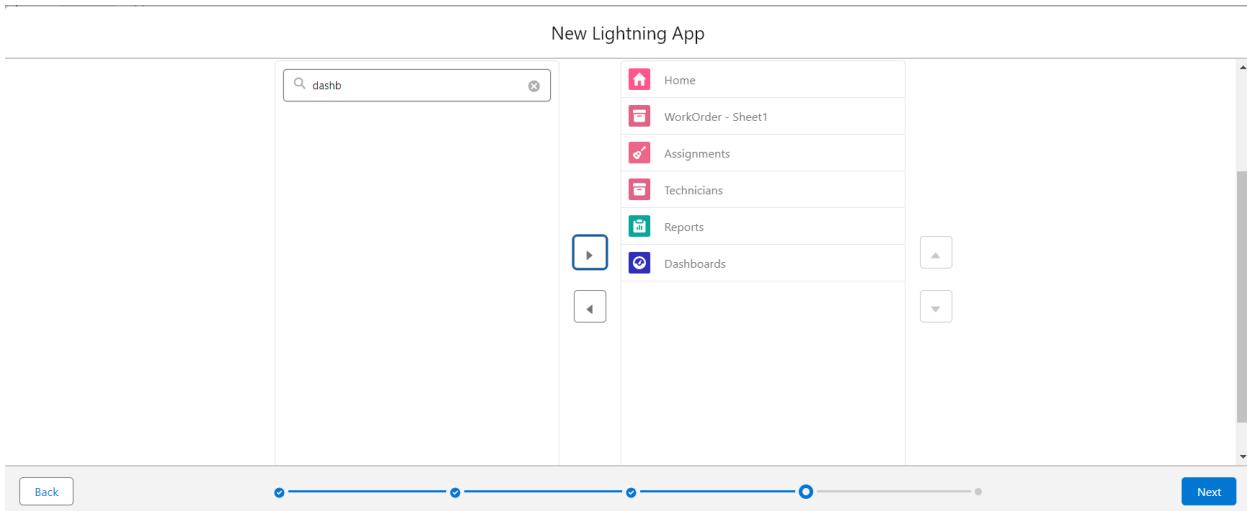
5. Utility Items:

- On the **Utility Items** page, keep the default settings.
- Click **Next**.

6. Add Navigation Items:

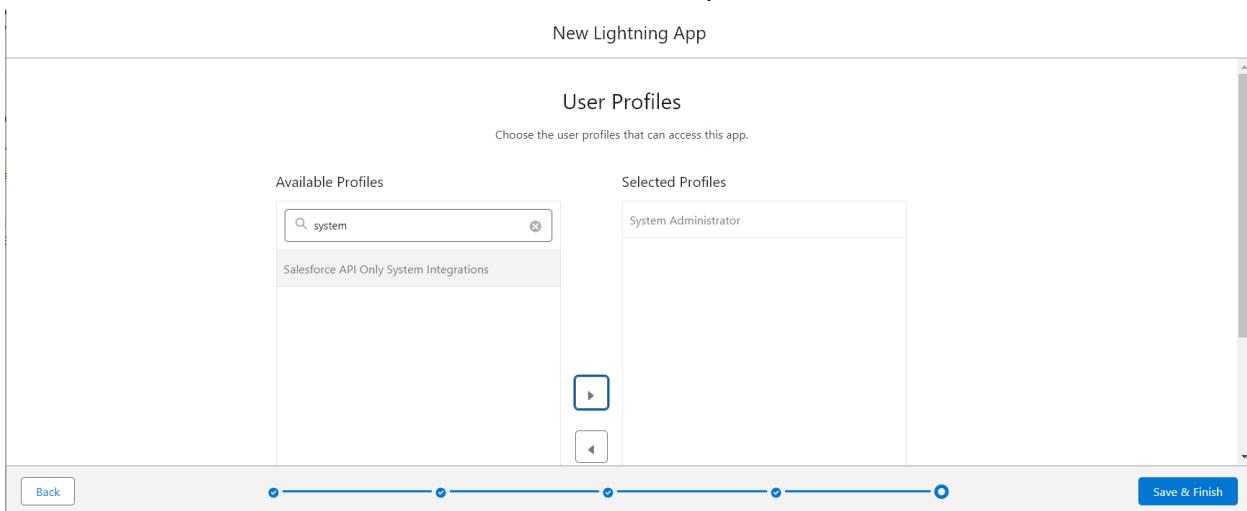
- In the **Navigation Items** section, search for the following items in the search bar:
 - Home
 - WorkOrder

- Technician
- Assignment
- Reports
- Dashboard
- Select each item and move it to the selected items section using the arrow button.
- **Note:** Make sure to select the **Assignment** custom object that you created in the Task 2 (Assignment Object Creation).
- Click **Next** to continue.



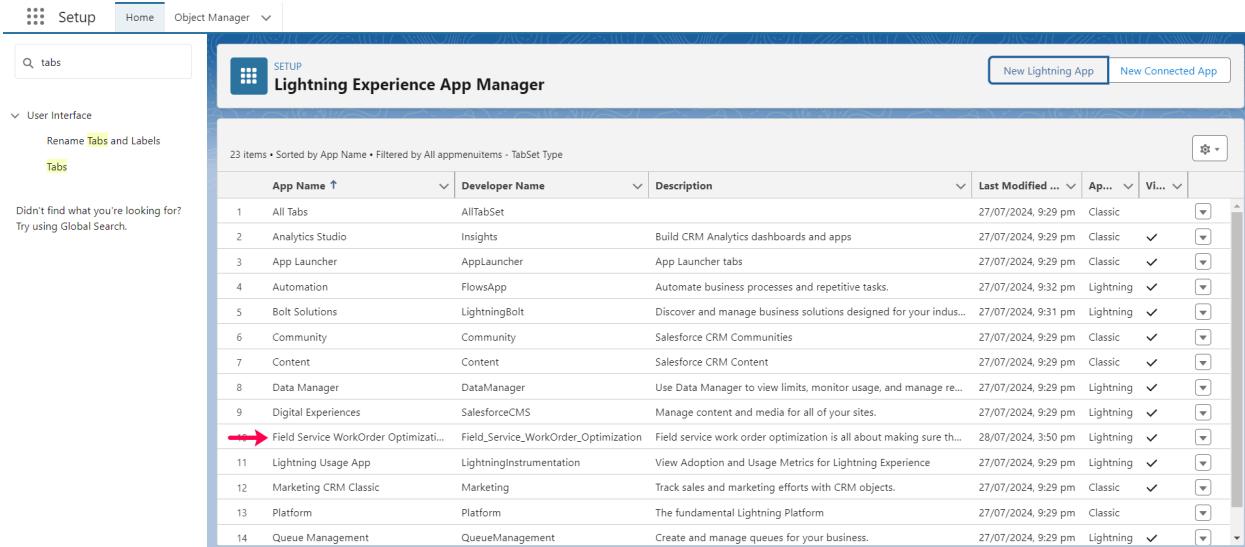
7. Add User Profiles:

- In the **User Profiles** section, search for **System Administrator** in the search bar.
- Click the arrow button to add it to the selected profiles.



8. Save & Finish:

- Click **Save & Finish** to complete the creation of the Lightning App Page.



The screenshot shows the Salesforce Lightning Experience App Manager. The left sidebar has a 'User Interface' section with 'Tabs' selected. A message says 'Didn't find what you're looking for? Try using Global Search.' The main area is titled 'Lightning Experience App Manager' and shows a table of 23 items. The columns are 'App Name', 'Developer Name', 'Description', 'Last Modified', 'Ap...', and 'Vi...'. One row, 'Field Service WorkOrder Optimizati...', is highlighted with a red arrow pointing to it. The table data is as follows:

App Name	Developer Name	Description	Last Modified	Ap...	Vi...
All Tabs	AllTabSet		27/07/2024, 9:29 pm	Classic	
Analytics Studio	Insights	Build CRM Analytics dashboards and apps	27/07/2024, 9:29 pm	Classic	
App Launcher	AppLauncher	App Launcher tabs	27/07/2024, 9:29 pm	Classic	
Automation	FlowsApp	Automate business processes and repetitive tasks.	27/07/2024, 9:32 pm	Lightning	
Bolt Solutions	LightningBolt	Discover and manage business solutions designed for your indu...	27/07/2024, 9:31 pm	Lightning	
Community	Community	Salesforce CRM Communities	27/07/2024, 9:29 pm	Classic	
Content	Content	Salesforce CRM Content	27/07/2024, 9:29 pm	Classic	
Data Manager	DataManager	Use Data Manager to view limits, monitor usage, and manage re...	27/07/2024, 9:29 pm	Lightning	
Digital Experiences	SalesforceCMS	Manage content and media for all of your sites.	27/07/2024, 9:29 pm	Lightning	
Field Service WorkOrder Optimizati...	Field_Service_WorkOrder_Optimization	Field service work order optimization is all about making sure th...	28/07/2024, 3:50 pm	Lightning	
Lightning Usage App	LightningInstrumentation	View Adoption and Usage Metrics for Lightning Experience	27/07/2024, 9:29 pm	Lightning	
Marketing CRM Classic	Marketing	Track sales and marketing efforts with CRM objects.	27/07/2024, 9:29 pm	Classic	
Platform	Platform	The fundamental Lightning Platform	27/07/2024, 9:29 pm	Classic	
Queue Management	QueueManagement	Create and manage queues for your business.	27/07/2024, 9:29 pm	Lightning	

TASK-5

Fields & Relationship

1. Creating Lookup Field in Assignment Object

To create a Lookup field in the "Assignment" object that relates to the "WorkOrder" object:

1. Access the Assignment Object:

- Go to the **Setup** page in Salesforce.
- Click on **Object Manager**.
- In the **Quick Find** bar, type Assignment and click on the **Assignment** object.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Assignment ID	Name	Auto Number		✓
Created By	CreatedBy	Lookup(User)		
Last Modified By	LastModifiedBy	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓

2. Create a New Lookup Field:

- In the Assignment object setup, click on **Fields & Relationships**.
- Click on **New** to create a new field.

3. Select Data Type:

- Select **Lookup Relationship** as the Data Type.
- Click on **Next**.

Step 1. Choose the field type

Assignment
New Custom Field

Specify the type of information that the custom field will contain.

Data Type

Lookup Relationship

Select one of the data types below.

None Selected

A system-generated sequence number that uses a display format you define. The number is automatically incremented for each new record.

Auto Number

A read-only field that derives its value from a formula expression you define. The formula field is updated when any of the source fields change.

Formula

A read-only field that displays the sum, minimum, or maximum value of a field in a related list or the record count of all records listed in a related list.

Roll-Up Summary

Creates a relationship that links this object to another object. The relationship field allows users to click on a lookup icon to select a value from a popup list. The other object is the source of the values in the list.

Master-Detail Relationship

Creates a specific type of master-detail relationship between this object (the child, or "detail") and another object (the parent, or "master") where:

- The master relationship field is required on all detail records.
- The ownership and sharing of a detail record are determined by the master record.
- When a user deletes the master record, all detail records are deleted.
- You can create rollup summary fields on the master record to summarize the detail records.

The relationship field allows users to click on a lookup icon to select a value from a popup list. The master object is the source of the values in the list.

4. Choose Related Object:

- For the field label related to, select the **WorkOrder** object.
- **Note:** Ensure that you are selecting the correct "WorkOrder" object (the one that matches the use case).

The screenshot shows the Salesforce Setup interface for the Assignment object. The left sidebar lists various setup options like Details, Fields & Relationships, Page Layouts, etc. The main area is titled 'Assignment New Relationship'. Step 2 is currently active, asking to choose the related object, which is set to 'WorkOrder - Sheet1'. Navigation buttons for Previous, Next, and Cancel are visible.

5. Define Field Label:

- **Field Label:** Enter WorkOrder ID.
- **Click Next.**

The screenshot shows Step 3 of the relationship setup. It defines the field label as 'WorkOrder ID', field name as 'WorkOrder_ID', and child relationship name as 'Assignments'. The 'Clear the value of this field' option is selected under 'What to do if the lookup record is deleted?'. The 'Add this field to existing custom report types that contain this entity' checkbox is checked under 'Auto add to custom report type'. Other fields like Description and Help Text are empty.

6. Finalize the Field Setup:

- **Click Next** through the subsequent screens, keeping the default settings.

Assignment

New Relationship

Step 4. Establish field-level security for reference field

Field Label: WorkOrder ID
 Data Type: Lookup
 Field Name: WorkOrder_ID
 Description:

Select the profiles to which you want to grant edit access to this field via field-level security. The field will be hidden from all profiles if you do not add it to field-level security.

Field-Level Security for Profile	Visible	Read-Only
Analytics Cloud Integration User	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Analytics Cloud Security User	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Contract Manager	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Cross Org Data Proxy User	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Custom: Marketing Profile	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Custom: Sales Profile	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Custom: Support Profile	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Step 5 of 6

Step 5. Add reference field to Page Layouts

Field Label: WorkOrder ID
 Data Type: Lookup
 Field Name: WorkOrder_ID
 Description:

Select the page layouts that should include this field. The field will be added as the last field in the first 2-column section of these page layouts. The field will not appear on any pages if you do not select a layout.

To change the location of this field on the page, you will need to customize the page layout.

Add Field Page Layout Name
 Assignment Layout

Step 6 of 6

Step 6. Add custom related lists

Field Label: WorkOrder ID
 Data Type: Lookup
 Field Name: WorkOrder_ID
 Description:

Specify the title that the related list will have in all of the layouts associated with the parent.
 Related List Label: Assignments

Select the page layouts that should include this field. The field will be added as the last field in the first 2-column section of these page layouts. The field will not appear on any pages if you do not select a layout.

To change the location of this field on the page, you will need to customize the page layout.

Add Related List Page Layout Name
 WorkOrder - Sheet1 Layout

Append related list to users' existing personal customizations

Finally, click on **Save** to finish.

The screenshot shows the Salesforce Setup interface. The top navigation bar includes a cloud icon, 'Setup', 'Home', 'Object Manager', and various system icons. The main content area is titled 'WorkOrder - Sheet1'. On the left, a sidebar lists setup categories like 'Page Layouts', 'Lightning Record Pages', etc., with 'Fields & Relationships' currently selected. The main table lists 10 items under 'Fields & Relationships', sorted by Field Label. The columns include Field Label, Field Name, Data Type, Controlling Field, and Indexed status.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Description	Description__c	Long Text Area(131072)		
Email	Email__c	Email		
Last Modified By	LastModifiedById	Lookup(User)		
Location	Location__c	Picklist		
Owner	OwnerId	Lookup(User,Group)		
Priority	Priority__c	Picklist		
Service Type	Service_Type__c	Picklist		
Status	Status__c	Picklist		
WorkOrder ID	Name	Type(0)		

The Lookup field "WorkOrder ID" is now created in the "Assignment" object, linking it to the "WorkOrder" object.

2. Manage your picklist values

To add new picklist values to the **Location** field in the **WorkOrder** object, follow these steps:

1. Access Setup:

- Log in to Salesforce and click the gear icon (⚙) in the upper right corner.
- Select "Setup" from the dropdown menu.

2. Navigate to Object Manager:

- In the Setup page, click on "Object Manager."
- Search for and select the **WorkOrder** object.

3. Go to Fields & Relationships:

- In the WorkOrder object setup, click on "Fields & Relationships."
- Find and select the **Location** field.

4. Add New Picklist Values:

- Scroll down to the "Values" section.
- Click the "New" button.

5. Enter New Values:

- Add the following values, one per line:
 - Nasik
 - Warangal
 - Nanded

SETUP > OBJECT MANAGER

WorkOrder - Sheet1

Fields & Relationships

Add Picklist Values
Location

Nasik
Warangal
Nanded

Help for this Page

Save Cancel

Save:

- Click "Save" to add the new values to the Location picklist.

Values						Values Help
Action	Values	API Name	Default	Chart Colors	Modified By	
<input type="checkbox"/> Edit Del Deactivate	Value1	Value1	<input type="checkbox"/>	Assigned dynamically	Nikhitha Vasireddy, 28/07/2024, 3:21 pm	
<input type="checkbox"/> Edit Del Deactivate	Nasik	Nasik	<input type="checkbox"/>	Assigned dynamically	Nikhitha Vasireddy, 28/07/2024, 4:23 pm	
<input type="checkbox"/> Edit Del Deactivate	Warangal	Warangal	<input type="checkbox"/>	Assigned dynamically	Nikhitha Vasireddy, 28/07/2024, 4:23 pm	
<input type="checkbox"/> Edit Del Deactivate	Nanded	Nanded	<input type="checkbox"/>	Assigned dynamically	Nikhitha Vasireddy, 28/07/2024, 4:23 pm	

3. Manage your picklist values

Adding Picklist Values to the WorkOrder Object in Salesforce

1. Priority Field:

- Navigate to the **Priority** field within the WorkOrder object.
- Scroll to the "Values" section, click "New," and add the value "High."

WorkOrder - Sheet1 Custom Field
Priority

Custom Field Definition Detail

Field Information

Field Label	Priority	Object Name	WorkOrder - Sheet1
Field Name	Priority	Data Type	Picklist
API Name	Priority__c		
Description			
Help Text			
Data Owner			
Field Usage			
Data Sensitivity Level			
Compliance Categorization			
Created By	Nikhitha Vasireddy, 28/07/2024, 3:21 pm	Modified By	Nikhitha Vasireddy, 28/07/2024, 3:21 pm

General Options

Required

Default Value

Click "Save."

Add Picklist Values

Priority

Add one or more picklist values below. Each value should be on its own line and it is used for both a value's label and API name.

If a value matches an inactive value's API name, that value is reactivated with its previous label.

If a value matches an inactive value's label but not the API name, a new value is created.

High

Save Cancel

2. Service Type Field:

- Navigate to the **Service Type** field.
- Scroll to the "Values" section, click "New," and add the following values:
 - Hardware repair
 - Troubleshoot/Debugging
 - Lane-Management

- Click "Save."

4. Creating Formula Field in WorkOrder Object

1. Repeat Steps 1 and 2 from the previous activity to navigate to the **Fields & Relationships** section of the WorkOrder object.
2. **Create a New Field:**
 - Click "New" to create a new field.
 - Select **Data Type** as "Formula" and click "Next."

SETUP > OBJECT MANAGER
Assignment

Fields & Relationships

Data Type

- None Selected
- Auto Number
- Formula
- Roll-Up Summary
- Lookup Relationship
- Master-Detail Relationship
- External Lookup Relationship
- Checkbox
- Currency

- **Enter Field Details:**

- **Field Label:** Enter "Date."
- **Field Name:** It will auto-populate as "Date."
- **Formula Return Type:** Select "Date."
- Click "Next."

SETUP > OBJECT MANAGER
Assignment

New Custom Field

Step 2. Choose output type

Step 2 of 5

Field Label: Date

Field Name: Date

Add this field to existing custom report types that contain this entity

Formula Return Type

- None Selected
- Checkbox
- Currency
- Date
- Date/Time

Enter the Formula:

- Under the **Advanced Formula** section, write the following formula:
CreatedDate
- Click "Check Syntax" to ensure the formula is correct.

The screenshot shows the Salesforce Setup interface for creating a new object. The left sidebar is titled 'Fields & Relationships'. The main area has a title 'Assignment' and a sub-section 'Insert Field'. A formula editor window is open, showing the expression 'Date (Date) = createdDate'. To the right of the formula is a list of function categories: ABS, ACOS, ADDMONTHS, AND, ASCII, ASIN. At the bottom of the formula editor, a message says 'Check Syntax | No syntax errors in merge fields or functions. (Compiled size: 20 characters)'.

Finalize and Save:

- Click "Next."
- Click "Next" again to skip through the visibility settings.
- Click "Save" to create the formula field.

The screenshot shows the final step of creating a new custom field 'Assignment'. The left sidebar is titled 'Fields & Relationships'. The main area is titled 'New Custom Field' and 'Step 5. Add to page layouts'. It shows a table with one row for the 'Assignment' field, which has a label 'Date', type 'Formula', name 'Date', and no description. Below the table, it says 'Select the page layouts that should include this field.' and lists 'Add Field' and 'Page Layout Name' with 'Assignment Layout' checked. At the bottom, there are buttons for 'Previous', 'Save & New', 'Save', and 'Cancel'.

5. Creating Remaining Fields For The Respective Objects

1. Technician ID Field

- **Field Type:** Lookup
- **Related To:** Technician

- **Steps:**

- Go to the **Fields & Relationships** section in the Assignment object.
- Click "New."
- Select **Data Type** as "Lookup Relationship" and click "Next."
- Choose **Related To** as "Technician" and click "Next."
- Set **Field Label** as "Technician ID" (Field Name auto-populates).
- Click "Next" and complete the remaining steps.
- Click "Save."

2. Assignment Date Field

- **Field Type:** Formula

- **Return Type:** Date

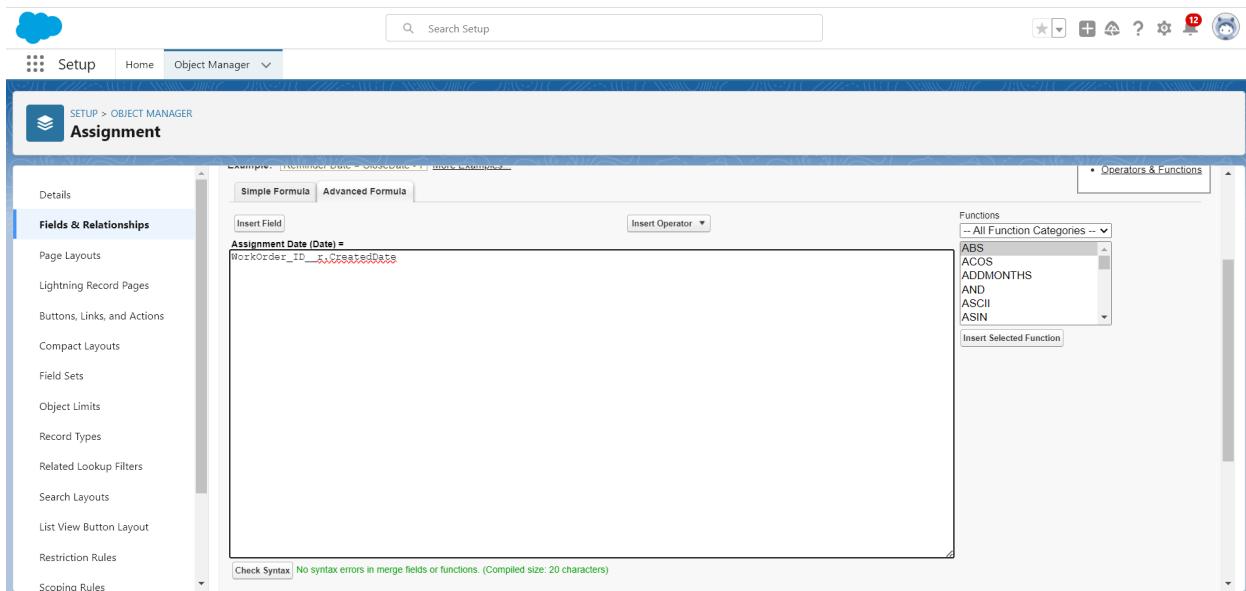
- **Formula:** `WorkOrder_ID__r.Date__c`

- **Steps:**

- Click "New" in the **Fields & Relationships** section.
- Select **Data Type** as "Formula" and click "Next."
- Set **Field Label** as "Assignment Date" (Field Name auto-populates).
- Choose **Formula Return Type** as "Date" and click "Next."
- In the **Advanced Formula** section, enter the formula:

WorkOrder_ID__r.Date__c

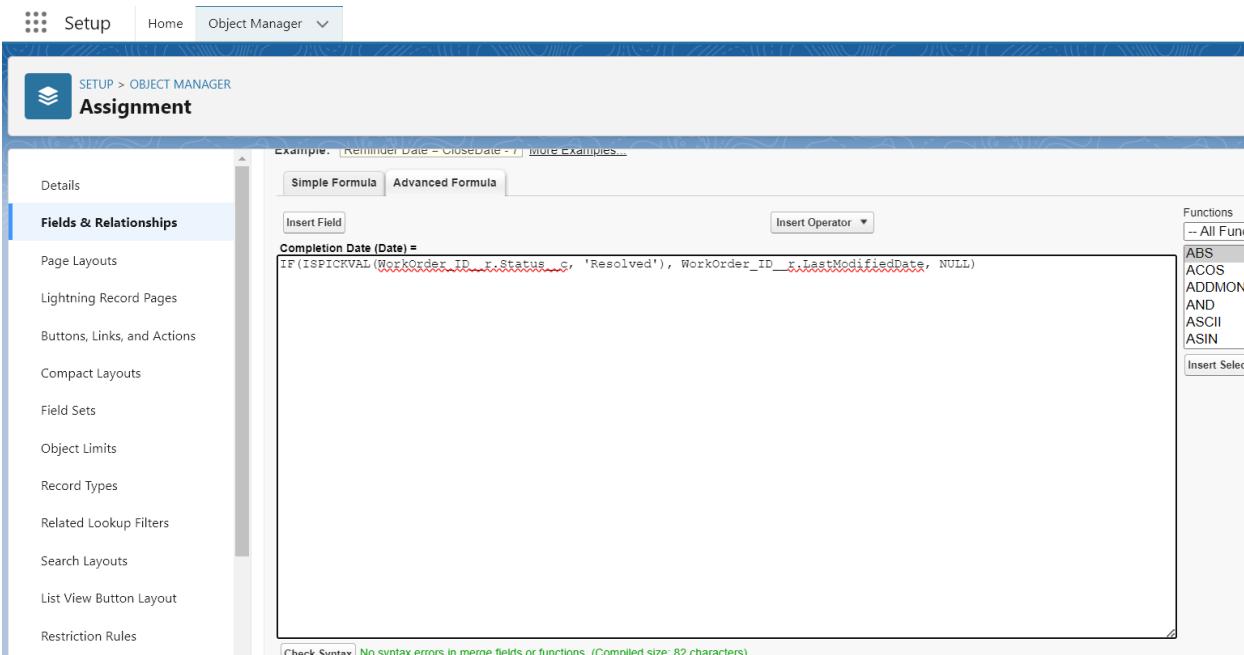
- Click "Check Syntax" to verify the formula.
- Click "Next," then "Next," and finally "Save."



3. Completion Date Field

- **Field Type:** Formula
- **Return Type:** Date
- **Formula:**

```
IF(ISPICKVAL(WorkOrder_ID__r.Status__c, 'Resolved'),  
    WorkOrder_ID__r.LastModifiedDate, NULL)
```
- **Steps:**
 - a. Click "New" in the **Fields & Relationships** section.
 - b. Select **Data Type** as "Formula" and click "Next."
 - c. Set **Field Label** as "Completion Date" (Field Name auto-populates).
 - d. Choose **Formula Return Type** as "Date" and click "Next."
 - e. In the **Advanced Formula** section, enter the formula:
**IF(ISPICKVAL(WorkOrder_ID__r.Status__c, 'Resolved'),
 WorkOrder_ID__r.LastModifiedDate, NULL)**
 - f. Click "Check Syntax" to verify the formula.
 - g. Click "Next," then "Next," and finally "Save."



These steps will create the **Technician ID**, **Assignment Date**, and **Completion Date** fields in the **Assignment** object with the specified data types and formulas.

SETUP > OBJECT MANAGER

Assignment

Fields & Relationships				
8 Items, Sorted by Field Label				
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Assignment ID	Name	Auto Number		✓
Completion Date	Completion_Date__c	Formula (Date)		▼
Created By	CreatedById	Lookup(User)		▼
Date	Date__c	Formula (Date)		▼
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 - Sheet1)		✓

TASK-6

Profiles

Creating Technician Profile

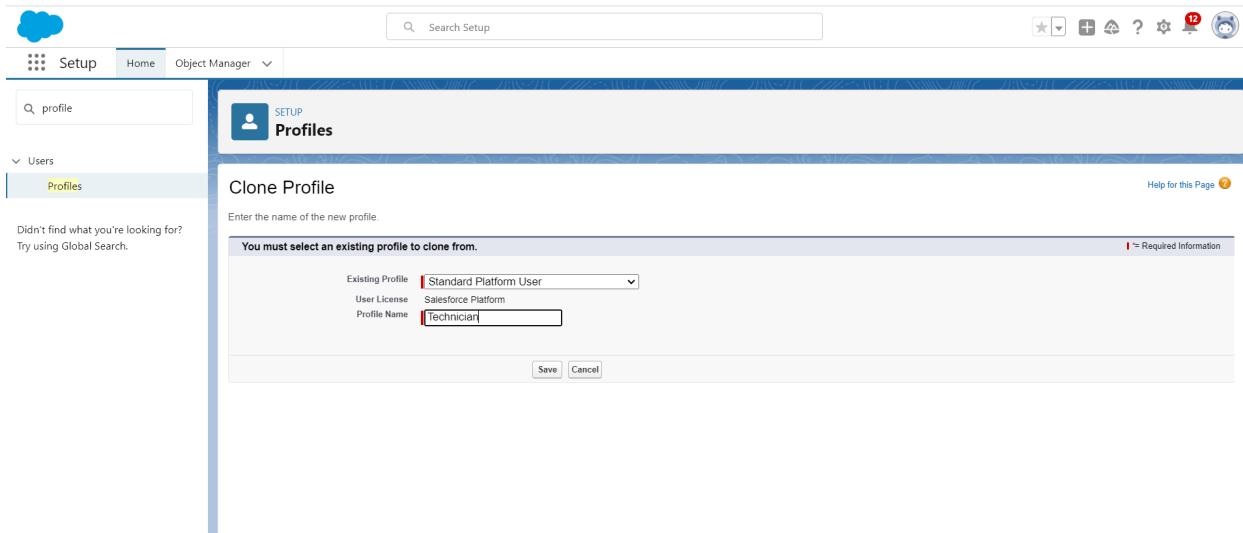
1. Create a New Profile

1. Go to Profiles:

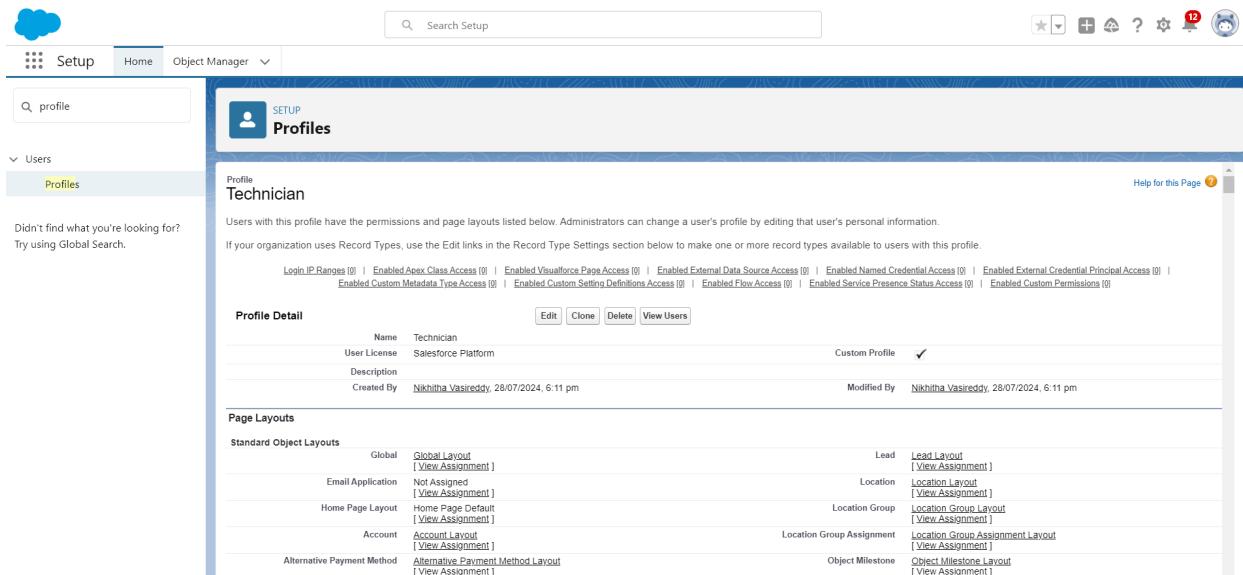
- In Setup, type "Profiles" in the Quick Find box and click on "Profiles."

2. Create New Profile:

- Click on "New Profile."
- Select **Standard Platform User** as the existing profile.
- Enter **Technician** as the Profile Name.



■ Click "Save."



2. Edit Profile Permissions

1. **Edit the Technician Profile:**
 - While still on the Technician profile page, click "Edit."
2. **Set Custom Object Permissions:**
 - Scroll down to the **Custom Object Permissions** section.
 - Set **Read Only** access for the **Technician**, **WorkOrder**, and **Assignment** objects.
3. **Save the Profile:**
 - Scroll down and click "Save."

The screenshot shows the Salesforce Setup interface under the Profiles section. On the left, there's a search bar and a sidebar with 'Users' and 'Profiles' selected. The main area has a title 'Profiles' with a blue icon. Below it is a 'Custom Object Permissions' section with two tables. The first table is for 'Assignments' and the second for 'Technicians'. Both tables have columns for 'Basic Access' (Read, Create, Edit, Delete) and 'Data Administration' (View All, Modify All). The second table also includes a 'WorkOrder - Sheet1' row. At the bottom are 'Session Settings' with options for session timeout and security level.

3. Configure Field-Level Security

1. Access Field-Level Security for WorkOrder:

- From the Technician profile detail page, scroll down to the **Custom Field-Level Security** section.
- Click "View" next to the **WorkOrder** object.

This screenshot shows the 'Custom Field-Level Security' page for the 'WorkOrder' object. It lists two profiles: 'Assignment' and 'Technician', each with a '[View]' link. The 'Technician' profile is currently selected.

1. Edit Status Field Security:

- Click "Edit" on the WorkOrder field-level security page.
- Enable the checkbox for the **Status** field.

This screenshot shows the 'WorkOrder - Sheet1 Field-Level Security for profile Technician' page. It lists various fields like Created By, Description, Email, etc., with their field types and current access levels (Read Access and Edit Access). The 'Status' field is highlighted with a checked checkbox under 'Edit Access'.

2. Save Changes:

- Click "Save" to apply the changes.

The screenshot shows the Salesforce Setup interface under the Profiles section. It displays the Field-Level Security configuration for the 'Technician' profile. The table includes columns for Field Name, Field Type, Read Access, and Edit Access. Most fields have 'Read Access' checked, while 'Edit Access' is mostly unchecked except for 'Owner' and 'WorkOrder ID'.

Field Name	Field Type	Read Access	Edit Access
Created By	Lookup	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Description	Long Text Area	<input type="checkbox"/>	<input type="checkbox"/>
Email	Email	<input type="checkbox"/>	<input type="checkbox"/>
Last Modified By	Lookup	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Location	Picklist	<input type="checkbox"/>	<input type="checkbox"/>
Owner	Lookup	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Priority	Picklist	<input type="checkbox"/>	<input type="checkbox"/>
Service Type	Picklist	<input type="checkbox"/>	<input type="checkbox"/>
Status	Picklist	<input checked="" type="checkbox"/>	<input type="checkbox"/>
WorkOrder ID	Text	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

TASK-7

Users

Create User

To create a new user with the **Technician** profile,

- **Access Users Setup:**
 - In Setup, type "Users" in the Quick Find box and select "Users."
- **Create a New User:**
 - Click "New User."
- **Fill in User Details:**
 - **First Name:** Elina
 - **Last Name:** Gilbert
 - **Alias:** (Enter an alias, e.g., EGilbert)
 - **Email:** (Enter your personal email address)
 - **Username:** (Enter a username in the form: text@text.text, e.g., elina@company.com)
 - **Nickname:** (Enter a nickname, e.g., ElinaG)
 - **Role:** (Select a role if needed or leave it as is)
 - **User License:** Select **Salesforce Platform**
 - **Profile:** Select **Technician**.

Save the User:

- Click "Save" to create the new user.

This will create a user named Elina Gilbert with the Technician profile and the specified user details.

User Detail		Edit Sharing Reset Password Freeze View Summary	
Name	Elina Gilbert	Role	Salesforce Platform
Alias	nikku	User License	Technician
Email	nikhithavasi1605@gmail.com [Verify]	Active	<input checked="" type="checkbox"/>
Username	nikku@salesforce.com	Marketing User	<input type="checkbox"/>
Nickname	nikku	Offline User	<input type="checkbox"/>
Title		Knowledge User	<input type="checkbox"/>
Company		Flow User	<input type="checkbox"/>
Department		Service Cloud User	<input type="checkbox"/>
Division		Site.com Contributor User	<input type="checkbox"/>
Address		Site.com Publisher User	<input type="checkbox"/>
Time Zone	(GMT+05:30) India Standard Time (Asia/Kolkata)	WDC User	<input type="checkbox"/>
Locale	English (India)	Mobile Push Registrations	View
Language	English	Data.com User Type	View
Delegated Approver		Accessibility Mode (Classic Only)	<input type="checkbox"/>
Manager		Debug Mode	<input type="checkbox"/>
Receive Approval Request Emails	Only if I am an approver	High-Contrast Palette on Charts	<input type="checkbox"/>
Federation ID			

TASK-8

Apex Trigger

1. Create an Apex Class

- **Access Developer Console:**
 - Log in to Salesforce.
 - Click on the gear icon (Setup) in the top right corner.
 - Select **Developer Console** from the dropdown. This will open a new console window.
- **Create a New Apex Class:**
 - In the Developer Console, go to the top left corner and click on **File**.
 - Select **New > Apex Class**.
 - In the popup window, enter the class name as WorkOrderClass.
 - Click **OK**.
- **Write the Code Logic:**
 - Replace the placeholder content in the class with the following code:

```
public class WorkOrderClass {  
    public static void workOrder(List<WorkOrder__c> newListWorkOrder){  
        Map<Integer, List<String>> maptotech = new Map<Integer,  
List<String>>();  
        Integer num = 0;  
        List<WorkOrder__c> properWo = new List<WorkOrder__c>();  
        List<Assignment__c> lstAssignment = new List<Assignment__c>();  
        List<Technician__c> technicianAssignment = new  
List<Technician__c>();  
  
        for (WorkOrder__c iter : newListWorkOrder) {  
            List<String> lststring = new List<String>();  
            if (iter.Service_Type__c != null && iter.Location__c !=  
null) {  
                num = num + 1;  
                properWo.add(iter);  
                lststring.add(iter.Service_Type__c);  
                lststring.add(iter.Location__c);  
                maptotech.put(num, lststring);  
            }  
        }  
    }  
}
```

```

    }

    Map<Integer, Id> techId = new Map<Integer, Id>();
    Map<Id, Technician__c> allTechnician = new Map<Id,
Technician__c>([SELECT Id, Name, Phone__c, Location__c, Skills__c,
Availability__c, Name__c, Email__c FROM Technician__c]);
    Integer num2 = 0;

    for (Technician__c T : allTechnician.values()) {
        num2 = num2 + 1;
        if (maptotech.get(num2) != null) {
            List<String> valofmap = maptotech.get(num2);
            System.debug('error 1 ----> the maptotech is empty ---> ' + maptotech.get(num2));
            if (valofmap.contains(T.Skills__c) &&
valofmap.contains(T.Location__c) && T.Availability__c == 'Available')
{
                techid.put(num2, T.Id);
            }
        }
    }

    Integer num3 = 0;
    for (WorkOrder__c W : properWo) {
        num3 = num3 + 1;
        Assignment__c A = new Assignment__c();
        A.WorkOrder_ID__c = W.Id;
        A.Technician_ID__c = techid.get(num3);
        lstAssignment.add(A);
    }

    if (!lstAssignment.isEmpty()) {
        insert lstAssignment;
    }
}
}

```

4. Save the Class:

- After entering the code, go to **File > Save** or press **Ctrl + S** to save your class.

This code defines a WorkOrderClass that processes a list of WorkOrder__c objects, assigns

appropriate technicians to work orders based on their skills, location, and availability, and creates Assignment__c records to map the work orders to technicians.

```

1 - public class WorkOrderClass {
2 -     public static void workOrder(List<WorkOrder_Sheet1__c> newListWorkOrder){
3 -         Map<Integer, List<String>> mapptech = new Map<Integer, List<String>>();
4 -         integer num = 0;
5 -         List<WorkOrder_Sheet1__c> properWo = new List<WorkOrder_Sheet1__c>();
6 -         List<Assignment__c> lstAssignment = new List<Assignment__c>();
7 -         List<Technician__c> tecniciantoAssignment = new List<Technician__c>();
8 -         for(Workorder_Sheet1__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 -                mapptech.put(num,lststring);
16 -            }
17 -        }
18 -        Map<integer,Id> techId = new Map<integer,Id>();
19 -        Map<Id,Technician__c> allTechnician = new Map<Id,Technician__c>([SELECT Id, Name, Phone__c, Location__c, Skills__c, Availability__c, Name__c FROM Technician__c]);
20 -        integer num2 = 0;
21 -        For(Technician__c t : allTechnician.values()){
22 -            num2 = num2+1;
23 -            if(mapptech.get(num2) != null){
24 -                List<string> valofmap = mapptech.get(num2);
25 -                system.debug('error 1 ----> the mapptech is empty ---> ' + mapptech.get(num2));
26 -                if(valofMap.contains(t.Skills__c) && ValoFMap.contains(t.Location__c) && t.Availability__c == 'Available'){
27 -                    techid.put(num2,t.Id);
28 -                }
29 -            }
30 -        }
31 -    }
32 -    integer num3 = 0;
33 -    For(WorkOrder_Sheet1__c w : properWo){
34 -        num3 = num3 + 1;
35 -        Assignment__c A = new Assignment__c();
36 -        A.WorkOrder_ID__c = W.Id;
37 -        A.Technician_ID__c = techid.get(num3);
38 -        lstAssignment.add(A);
39 -    }
40 -    If(!lstAssignment.IsEmpty()){
41 -        insert lstAssignment;
42 -    }
43 - }
44 - }
45 - }
46 - }

```

2. Create an Apex Trigger

- **Create a New Apex Trigger:**

- In the Developer Console, go to the top left corner and click on **File**.
- Select **New > Apex Trigger**.
- In the popup window, enter the trigger name as **WorkOrderTrigger**.
- In the dropdown list for **sObject**, select **WorkOrder__c**.
- Click **Submit**.

- **Write the Code Logic:**

- Replace the placeholder content in the trigger with the following code:

```
trigger WorkOrderTrigger on WorkOrder__c (after insert) {
    if (Trigger.isAfter && Trigger.isInsert) {
        WorkOrderClass.workOrder(Trigger.new);
    }
}
```

Save the Trigger:

- After entering the code, go to **File > Save** or press **Ctrl + S** to save your trigger.

```
trigger WorkOrderTrigger on WorkOrder__c (after insert) {
    if(trigger.isafter && trigger.isinsert){
        WorkOrderClass.workOrder(trigger.new);
    }
}
```

3. Create an Apex Class

- **Create a New Apex Class:**

- In the Developer Console, go to the top left corner and click on **File**.
- Select **New > Apex Class**.
- In the popup window, enter the class name as **AssigningEmail**.
- Click **OK**.

- **Write the Code Logic:**

- Replace the placeholder content in the class with the following 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(technicians.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: ' + con.WorkOrder_ID__c;  
mail.setHtmlBody(body);  
  
myVar.add(mail);  
}  
}  
Messaging.sendEmail(myVar);  
} catch (Exception e) {  
    System.debug('Error -----> ' + e.getMessage());  
}  
}  
}
```

Save the Class:

- After entering the code, go to **File > Save** or press **Ctrl + S** to save your class.

```

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> tecnicians = new Map<id,Technician__c>([SELECT Id, Phone__c, Locat
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(tecnicians.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             }
21             catch(exception e){
22                 system.debug('Error -----> ' + e.getMessage());
23             }
24         }
25     }
}

```

4. Create an Apex Trigger

- **Create a New Apex Trigger:**

- In the Developer Console, go to the top left corner and click on **File**.
- Select **New > Apex Trigger**.
- In the popup window, enter the trigger name as **AssignmentTrigger**.
- In the dropdown list for **sObject**, select **Assignment__c**.
- Click **Submit**.

- **Write the Code Logic:**

- Replace the placeholder content in the trigger with the following code:

```

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

```



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

5. Create an Apex Class

- **Create a New Apex Class:**

- In the Developer Console, go to the top left corner and click on **File**.
- Select **New > Apex Class**.
- In the popup window, enter the class name as **CompletionMail**.
- Click **OK**.

- **Write the Code Logic:**

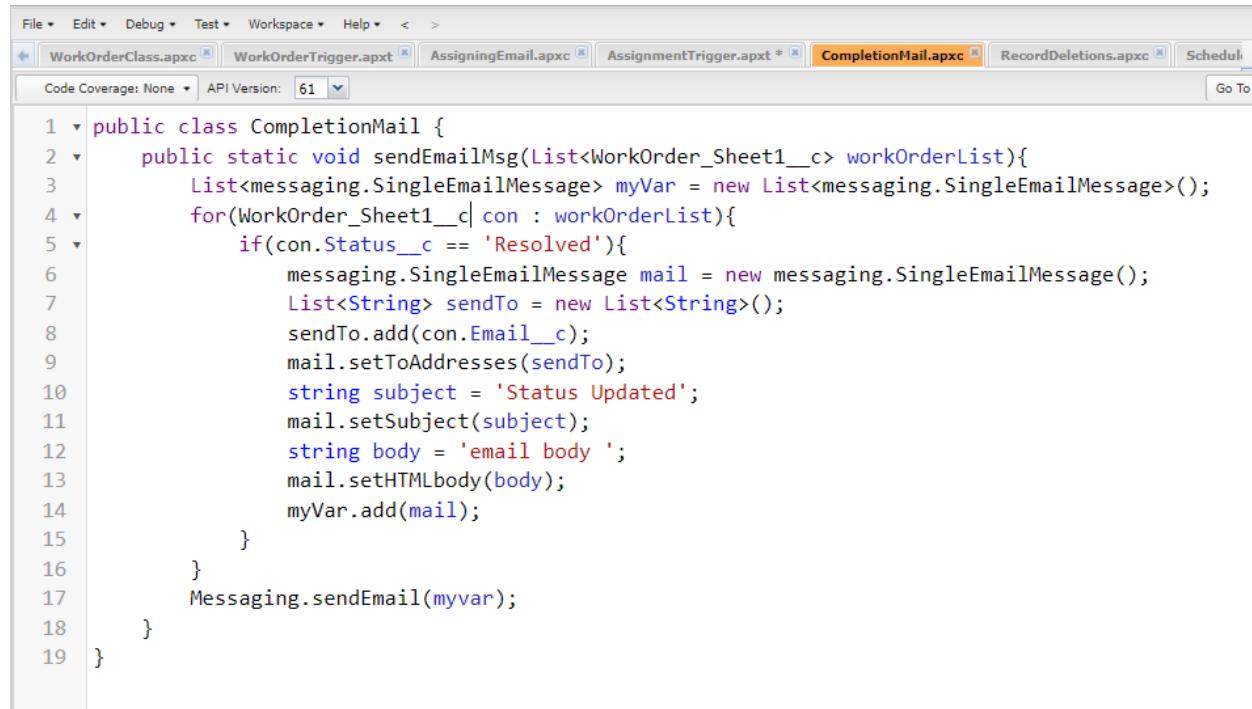
- Replace the placeholder content in the class with the following code:

```
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 = 'Your WorkOrder has been resolved.';
                mail.setHTMLBody(body);
                myVar.add(mail);
            }
        }
        Messaging.sendEmail(myVar);
    }
}
```

```
    }
}
```

Save the Class:

- After entering the code, go to **File > Save** or press **Ctrl + S** to save your class.



The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes File, Edit, Debug, Test, Workspace, Help, and tabs for various Apex classes like WorkOrderClass.apxc, WorkOrderTrigger.apxt, AssigningEmail.apxc, AssignmentTrigger.apxt, CompletionMail.apxc (which is currently selected), RecordDeletions.apxc, and Schedule. Below the tabs is a toolbar with Code Coverage: None, API Version: 61, and a Go To button. The main area displays the Apex code for the CompletionMail class:

```
1 public class CompletionMail {
2     public static void sendEmailMsg(List<WorkOrder_Sheet1__c> workOrderList){
3         List<messaging.SingleEmailMessage> myVar = new List<messaging.SingleEmailMessage>();
4         for(WorkOrder_Sheet1__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 }
```

6. Create an Apex Trigger

- Open the Existing Trigger:**
 - In the Developer Console, go to the top left corner and click on **File**.
 - Select **Open**.
 - In the popup window, click on the **Triggers** tab.
 - Select **WorkOrderTrigger** from the list and click on **Open**.
- Update the Code Logic:**
 - Replace the existing code in the trigger with the following:

```
trigger WorkOrderTrigger on WorkOrder__c (after insert, after update)
{
    if (Trigger.IsAfter && Trigger.IsInsert) {
        WorkOrderClass.workOrder(Trigger.New);
    }
    if (Trigger.IsAfter && Trigger.IsUpdate) {
```

```

        CompletionMail.sendEmailMsg(Trigger.New);
    }
}

```

```

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
WorkOrderClass.apxc WorkOrderTrigger.apxt AssigningEmail.apxc AssignmentTrigger.apxt * CompletionMail.apxc RecordDeletions.apxc Schedule...
Code Coverage: None API Version: 61 Go To
1 trigger WorkOrderTrigger on WorkOrder_Sheet1__c (after insert, after update) {
2
3     if(Trigger.IsAfter && Trigger.IsInsert){
4
5         WorkOrderClass.workOrder(trigger.new);
6     }
7     if(Trigger.IsAfter && Trigger.IsUpdate){
8         CompletionMail.sendEmailMsg(Trigger.New);
9     }
10 }

```

Save the Trigger:

- After entering the updated code, go to **File > Save** or press **Ctrl + S** to save your trigger.

7. Create an Asynchronous Apex Class

- **Create a New Apex Class:**
 - In the Developer Console, go to the top left corner and click on **File**.
 - Select **New > Apex Class**.
 - In the popup window, enter the class name as **RecordDeletion**.
 - Click **OK**.
- **Write the Code Logic:**
 - Replace the placeholder content in the class with the following code:

```

public class RecordDeletion implements Database.Batchable<SObject> {

    // Start method for defining the query
    public Database.QueryLocator start(Database.BatchableContext bc) {
        String query = 'SELECT Id FROM WorkOrder__c WHERE Status__c = \'Resolved\' AND Completed_Date__c < LAST_N_DAYS:30';
        return Database.getQueryLocator(query);
    }

    // Execute method to delete records in batches
}

```

```

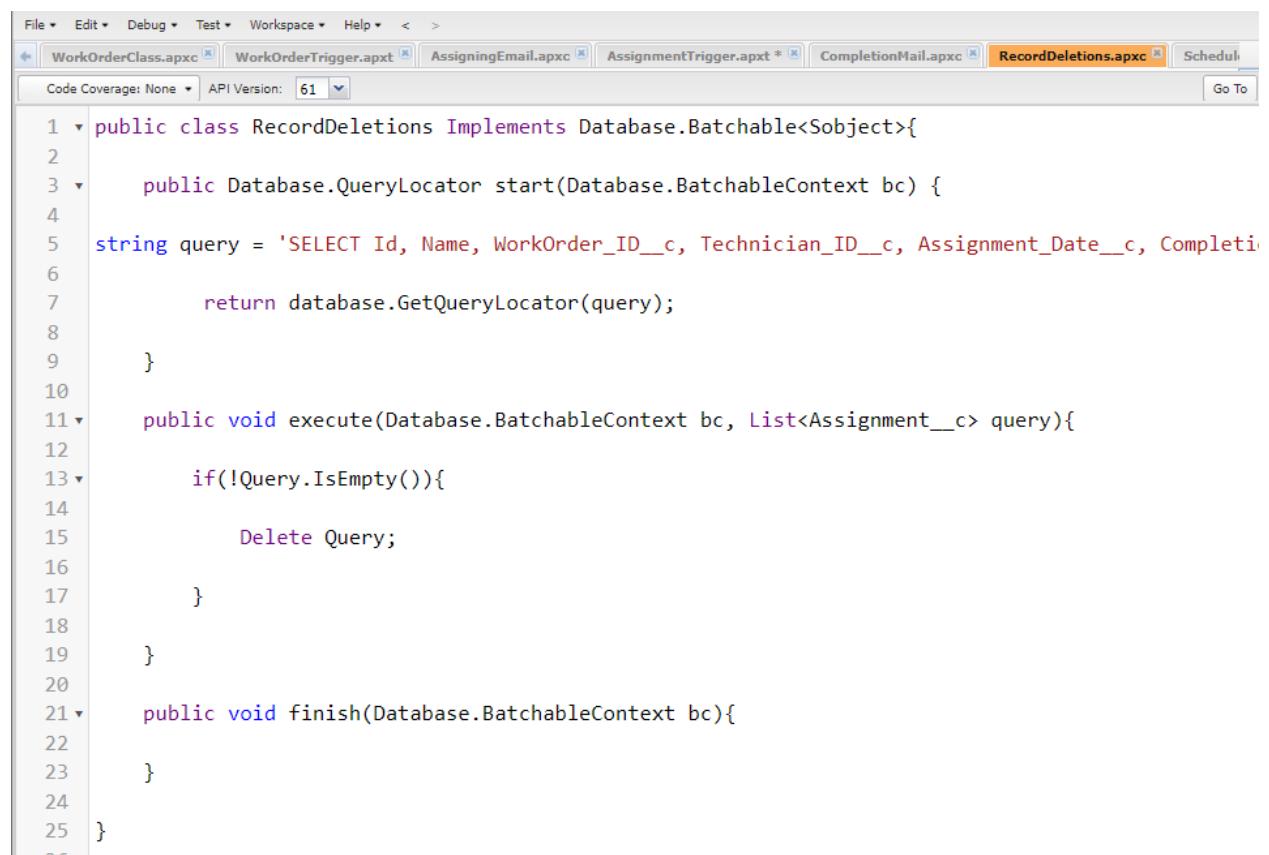
    public void execute(Database.BatchableContext bc,
List<WorkOrder__c> scope) {
        if (!scope.isEmpty()) {
            delete scope;
        }
    }

    // Finish method (empty as no special processing is needed after
completion)
    public void finish(Database.BatchableContext bc) {
        // Optional: You can log or send an email notification here if
needed
    }
}

```

Save the Class:

- After entering the code, go to **File > Save** or press **Ctrl + S** to save your class.



```

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
WorkOrderClass.apxc [ ] WorkOrderTrigger.apxt [ ] AssigningEmail.apxc [ ] AssignmentTrigger.apxt * [ ] CompletionMail.apxc [ ] RecordDeletions.apxc [ ] Schedule...
Code Coverage: None ▾ API Version: 61 ▾ Go To
1 public class RecordDeletions Implements Database.Batchable<Sobject>{
2
3     public Database.QueryLocator start(Database.BatchableContext bc) {
4
5         string query = 'SELECT Id, Name, WorkOrder_ID__c, Technician_ID__c, Assignment_Date__c, Completi...
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
21    public void finish(Database.BatchableContext bc){
22
23    }
24
25 }
26

```

Running the Batch Job

To execute this batch Apex class, you would use the following code snippet in the Developer Console's Execute Anonymous window:

```
RecordDeletion batch = new RecordDeletion();
```

```
Database.executeBatch(batch);
```

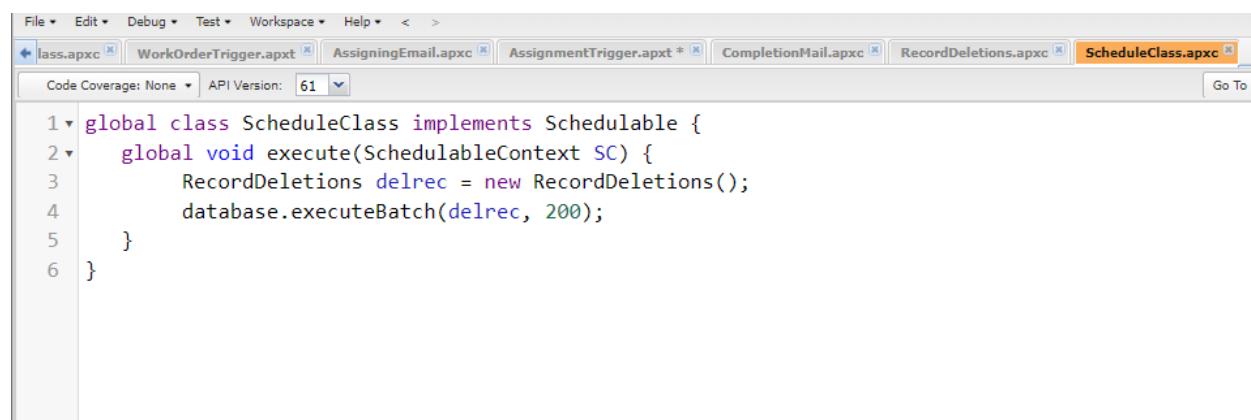
8. Create an Apex Schedule Class

- Create a New Apex Class:
 - In the Developer Console, go to the top left corner and click on **File**.
 - Select **New > Apex Class**.
 - In the popup window, enter the class name as **ScheduleClass**.
 - Click **OK**.
- Write the Code Logic:
 - Replace the placeholder content in the class with the following code:

```
global class ScheduleClass implements Schedulable {  
    global void execute(SchedulableContext SC) {  
        RecordDeletion delrec = new RecordDeletion();  
        database.executeBatch(delrec, 200);  
    }  
}
```

Save the Class:

- After entering the code, go to **File > Save** or press **Ctrl + S** to save your class.



```
File Edit Debug Test Workspace Help  
lass.apxc WorkOrderTrigger.apxt AssigningEmail.apxc AssignmentTrigger.apxt * CompletionMail.apxc RecordDeletions.apxc ScheduleClass.apxc  
Code Coverage: None API Version: 61 Go To  
1 ▾ global class ScheduleClass implements Schedulable {  
2 ▾   global void execute(SchedulableContext SC) {  
3       RecordDeletions delrec = new RecordDeletions();  
4       database.executeBatch(delrec, 200);  
5   }  
6 }
```

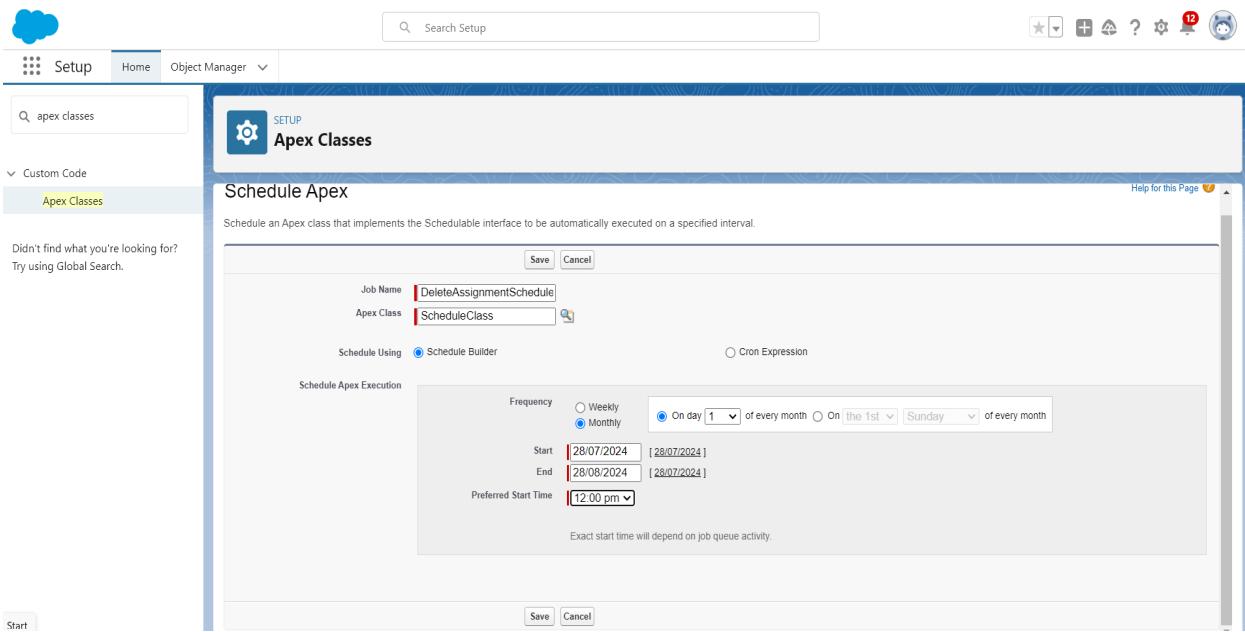
9. Create a Schedule Apex

- **Search for Apex Classes:**
 - In the Quick Find box on the left, type "Apex Classes."
 - Click on **Apex Classes** under the **Custom Code** section.
- **Schedule the Apex Class:**
 - On the **Apex Classes** page, click on the **Schedule Apex** button.

The screenshot shows the Salesforce Apex Classes page. At the top, there are navigation links: Setup, Home, Object Manager, and a search bar containing 'Q apex classes'. Below the search bar, a sidebar shows 'Custom Code' with 'Apex Classes' selected. A message says 'Didn't find what you're looking for? Try using Global Search.' The main content area has a title 'Apex Classes' and a sub-section 'Apex Code is an object oriented programming language that allows developers to develop on-demand business applications on the Lightning Platform.' A green box displays 'Percent of Apex Used: 0.09%' and a note about character usage. Below this is a table with columns: Action, Name, Namespace Prefix, Api Version, Status, Size Without Comments, Last Modified By, and Has Trace Flags. The table lists five Apex classes: AssigningEmail, CompletionMail, RecordDeletions, ScheduleClass, and WorkOrderClass, all created by Nikhitha Vasireddy on 28/07/2024. At the bottom, there's a section titled 'Dynamic Apex Classes' with a note about extending the platform's reach.

Action	Name	Namespace Prefix	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
Edit Del Security	AssigningEmail		61.0	Active	1,226	Nikhitha Vasireddy, 28/07/2024, 6:52 pm	<input type="checkbox"/>
Edit Del Security	CompletionMail		61.0	Active	815	Nikhitha Vasireddy, 28/07/2024, 7:10 pm	<input type="checkbox"/>
Edit Del Security	RecordDeletions		61.0	Active	593	Nikhitha Vasireddy, 28/07/2024, 7:17 pm	<input type="checkbox"/>
Edit Del Security	ScheduleClass		61.0	Active	207	Nikhitha Vasireddy, 28/07/2024, 7:18 pm	<input type="checkbox"/>
Edit Del Security	WorkOrderClass		61.0	Active	1,989	Nikhitha Vasireddy, 28/07/2024, 7:13 pm	<input type="checkbox"/>

- **Enter Job Details:**
 - **Job Name:** Enter DeleteAssignmentSchedule.
 - **Apex Class:** Click on the lookup icon (🔍) next to the field and select ScheduleClass.
- **Set Frequency:**
 - **Frequency:** Select Monthly.
 - **Preferred Start Time:** Choose any preferred time from the dropdown.
- **Save the Schedule:**
 - After setting up the job details and frequency, click on **Save**.



TASK-9

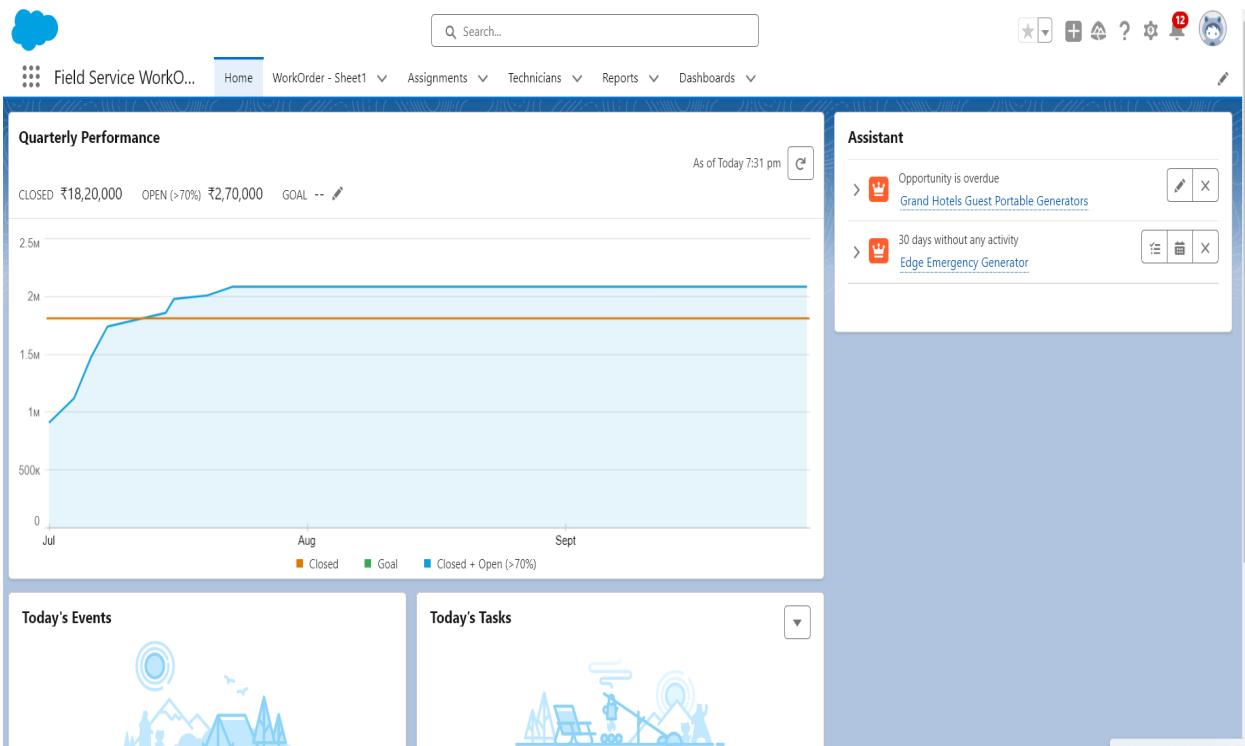
Reports and Dashboards

1. Report

Steps to Create a Report

1. Access the App:

- Log in to Salesforce.
- Go to the app that contains the **Reports** tab.



- **Click on the Reports Tab:**
 - In the app's navigation bar, click on the **Reports** tab.
- **Create a New Report:**
 - On the Reports page, click on the **New Report** button.
- **Select Report Type:**
 - In the report creation window, you can either:
 - Select a report type from the **Category** section.
 - Use the **Report Type** panel.
 - Use the **Search** panel to find the desired report type.
 - After selecting the appropriate report type, click on **Start Report**.
- **Customize Your Report:**
 - **Add Fields:** On the left pane, you'll see available fields. Drag and drop the fields you want to include in your report.
 - **Group by WorkOrder ID:** To group the report by WorkOrder__c ID, drag the WorkOrder__c ID field from the left pane to the "Group Rows" section.
- **Save or Run the Report:**
 - Once you've customized your report, you can either:
 - **Run** the report by clicking on the **Run** button.
 - **Save** the report by clicking on the **Save** button. When saving, you'll be prompted to provide a report name and select a folder to save it in.

The screenshot shows a Salesforce report titled "New Assignments with WorkOrder ID Report". The report displays 8 total records. The columns are: WorkOrder ID: Owner Name, WorkOrder ID: Record ID, Assignment: Assignment ID, WorkOrder ID: WorkOrder ID, WorkOrder ID: Service Type, and WorkOrder ID: Location. The data is grouped by owner name, with subtotals for each group. The rows show various assignments for different work orders, categorized by service type like Hardware repair, Troubleshoot/Debugging, and Lane-Management.

WorkOrder ID: Owner Name	WorkOrder ID: Record ID	Assignment: Assignment ID	WorkOrder ID: WorkOrder ID	WorkOrder ID: Service Type	WorkOrder ID: Location
Nikhitha Vasireddy (8)	a04dM000000etc5 (2)	A-0014 A-0007	1001 1001	Hardware repair Hardware repair	Nasik Nasik
	Subtotal				
	a04dM000000etdh (2)	A-0015 A-0008	1002 1002	Troubleshoot/Debugging Troubleshoot/Debugging	Warangal Warangal
	Subtotal				
	a04dM000000etfj (2)	A-0016 A-0009	1003 1003	Lane-Management Lane-Management	Nanded Nanded
	Subtotal				
	a04dM000000etgv (1)	A-0017	1004	Hardware repair	Nasik
	Subtotal				
	a04dM000000etgX (1)	A-0018	1005	Troubleshoot/Debugging	Warangal
	Subtotal				

Row Counts: Detail Rows: Subtotals: Grand Total:

2. Create Reports

i). Create a report with report type: “WorkOrders Status Reports”.

- **Select Report Type:**
 - In the **New Report** window, use the **Search** bar or browse through the **Report Types** panel to find “**WorkOrders Status Reports**”.
 - Click on the **WorkOrders Status Reports** report type.
 - Click **Start Report** to open the report builder.
- **Customize Your Report:**
 - **Add Fields:** Drag and drop the relevant fields from the left pane into the report. Common fields might include:
 - WorkOrder_c ID
 - Status_c
 - Completion_Date_c
 - Service_Type_c
 - Any other relevant fields for work order status.

Save or Run the Report:

- **Run the Report:** Click on the **Run** button to view the report results.
- **Save the Report:** Click on **Save**.

Report: WorkOrder - Sheet1

Work Order Status Report

Total Records
5

WorkOrder - Sheet1: WorkOrder ID	Priority	Email	Service Type	WorkOrder - Sheet1: Created By	Status
a04dM00000etc5	High	nikhitha@gmail.com	Hardware repair	Nikhitha Vasireddy	Value1
Subtotal					
a04dM00000etdh	High	pragna@gmail.com	Troubleshoot/Debugging	Nikhitha Vasireddy	Value1
Subtotal					
a04dM00000etfj	High	amosh@gmail.com	Lane-Management	Nikhitha Vasireddy	Value1
Subtotal					
a04dM00000etgv	High	ajay@gmail.com	Hardware repair	Nikhitha Vasireddy	Value1
Subtotal					
a04dM00000etiX	High	rahul@gmail.com	Troubleshoot/Debugging	Nikhitha Vasireddy	Value1
Subtotal					
Total (5)					

ii). Create a report with report type: “**Technician and Assignment Details Reports**”.

- **Access Report Types:**
 - Log in to Salesforce.
 - Click on the gear icon (Setup) in the top right corner.
 - In the Quick Find box, type “**Report Types**”.
 - Click on **Report Types** under the **Reports and Dashboards** section.
- **Create a New Custom Report Type:**
 - Click on the **New Custom Report Type** button.
- **Define the Report Type:**
 - **Primary Object:** Select **Technician__c** as the primary object.
 - **Report Type Label:** Enter a name like “**Technician and Assignment Details**”.
 - **Report Type Name:** This will auto-fill based on the label.
 - **Description:** Provide a description, such as “**Report including technician details and their assignments**.”
 - **Store in Category:** Choose a category like “**Other Reports**” or “**Custom Reports**”.

Define Report Relationships:

- Click on **Next** to proceed to define the relationships.
- Click on **Add Related Object** to link the Assignment__c object.
- Choose **Assignment__c** from the list of related objects.
- Define the relationship type. For example, if Assignment__c has a lookup to Technician__c, select **“Each Technician__c record may or may not have related Assignment__c records”**.

Objects Used in Report Type



Technician



Assignment

Save the Report Type: Click **Save** to create the custom report type.

- Go to the app that contains the **Reports** tab.
- Click on the **Reports** tab.
- **Create a New Report:**
 - Click on **New Report**.
- **Select Report Type:**
 - In the **New Report** window, find and select your newly created report type **"Technician and Assignment Details"**.
 - Click **Start Report** to open the report builder.
- **Customize the Report:**
 - **Add Fields:** Drag and drop fields from the **Technician__c** and **Assignment__c** objects into the report. Examples include:
 - From **Technician__c**:
 - Name
 - Email__c
 - Phone__c

- Skills__c
- From Assignment__c:
 - ID
 - WorkOrder_ID__c
 - Assignment_Date__c

Group By Fields: Drag fields like Technician__c Name to the "Group Rows" section if needed.

Save or Run the Report:

- **Run the Report:** Click on **Run** to view the report results.
- **Save the Report:** Click on **Save**. Enter a report name like "**Technician and Assignment Details Report**".

The screenshot shows a Salesforce report titled "Report: Technician and Assignment Details Reports" with the specific title "New Technician and Assignment Details". The report lists five technician records:

Technician ID	Name	Email	Phone	Skills	Created By: Full Name	Assignment ID	WorkOrder ID: WorkOrder ID	Created Date
a03dM0000043sBG (1)	Amosh (1)	amosh@gmail.com	8765432198	Machine Installation	Nikhitha Vasireddy	a05dM000005oQSF	1005	01/08/2024
Subtotal								
a03dM0000043vPF (1)	Pragna (1)	pragna@gmail.com	8654321769	Machine Installation	Nikhitha Vasireddy	a05dM000005oQKb	1001	01/08/2024
Subtotal								
a03dM0000043vc9 (1)	Rahul (1)	rahul@gmail.com	7654321876	Lane-Management	Nikhitha Vasireddy	a05dM000005oAob	1002	01/08/2024
Subtotal								
a03dM0000043vib (1)	Ajay (1)	ajay@gmail.com	9876453219	Hardware Repair	Nikhitha Vasireddy	a05dM000005o6de	1003	01/08/2024
Subtotal								
a03dM0000043vip (1)	Navya (1)	navya@gmail.com	7658543218	Maintenance	Nikhitha Vasireddy	a05dM000005oFCJ	1004	01/08/2024
Subtotal								

At the bottom of the report, there are checkboxes for Row Counts, Detail Rows, Subtotals, and Grand Total.

By creating this custom report type with **Technician__c** as the primary object, you can generate a detailed report that includes both technician and assignment details. This customization will help you analyze technician performance and assignment data effectively.

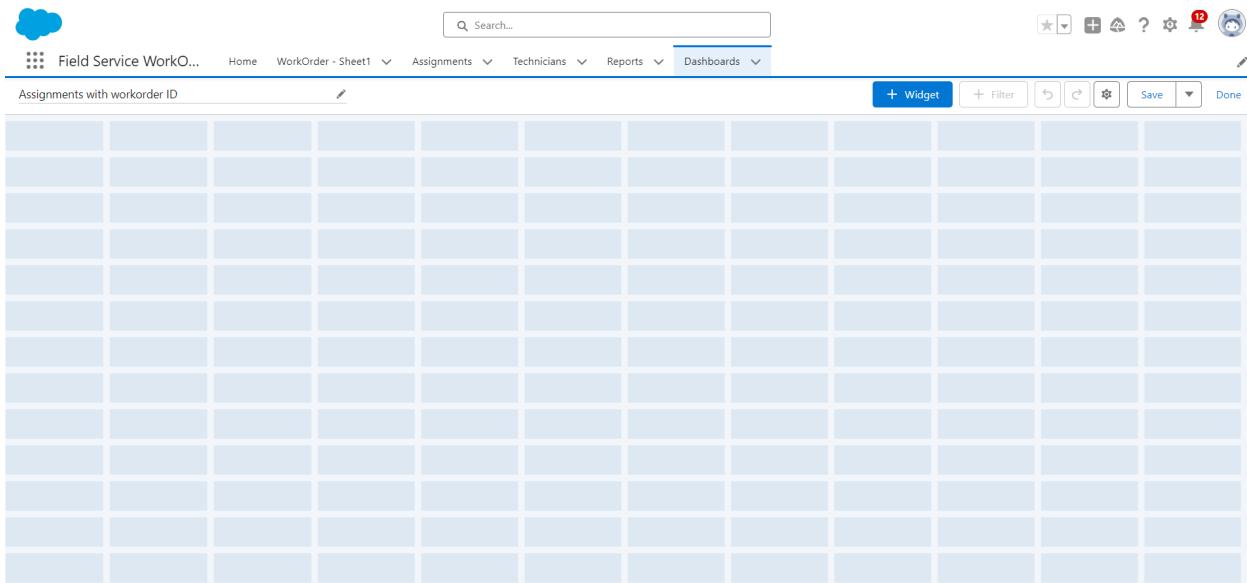
3. Dashboard

Steps to Create a Dashboard

1. Access the Dashboards Tab:

- Log in to Salesforce.

- Navigate to the app that contains the **Dashboards** tab.
 - Click on the **Dashboards** tab.
- 2. Create a New Dashboard:**
- Click on the **New Dashboard** button.
- 3. Name Your Dashboard:**
- Enter a name for your dashboard in the **Name** field. “**Assignments with WorkOrder ID Dashboard**”.
- **Click on Create:**
 - After entering the name and description, click on **Create** to start building the dashboard.
- **Add a Component to the Dashboard:**
 - Click on **+ Add Component** to add a new component to your dashboard.



- **Select a Report:**
 - Choose a report that you created earlier.
 - Click on **Select** to choose the report.
- **Configure the Component:**
 - After selecting the report, you'll be prompted to choose the type of component (e.g., chart, table, gauge).
 - Configure the component by adjusting settings like the chart type, data display options, and formatting.
- **Add the Component:**
 - Click **Add** to add the configured component to your dashboard.
- **Save the Dashboard:**
 - Once you've added all desired components and configured the layout, click **Save** to save your dashboard.
- **Finish:**

- Click on **Done** to finish creating your dashboard.



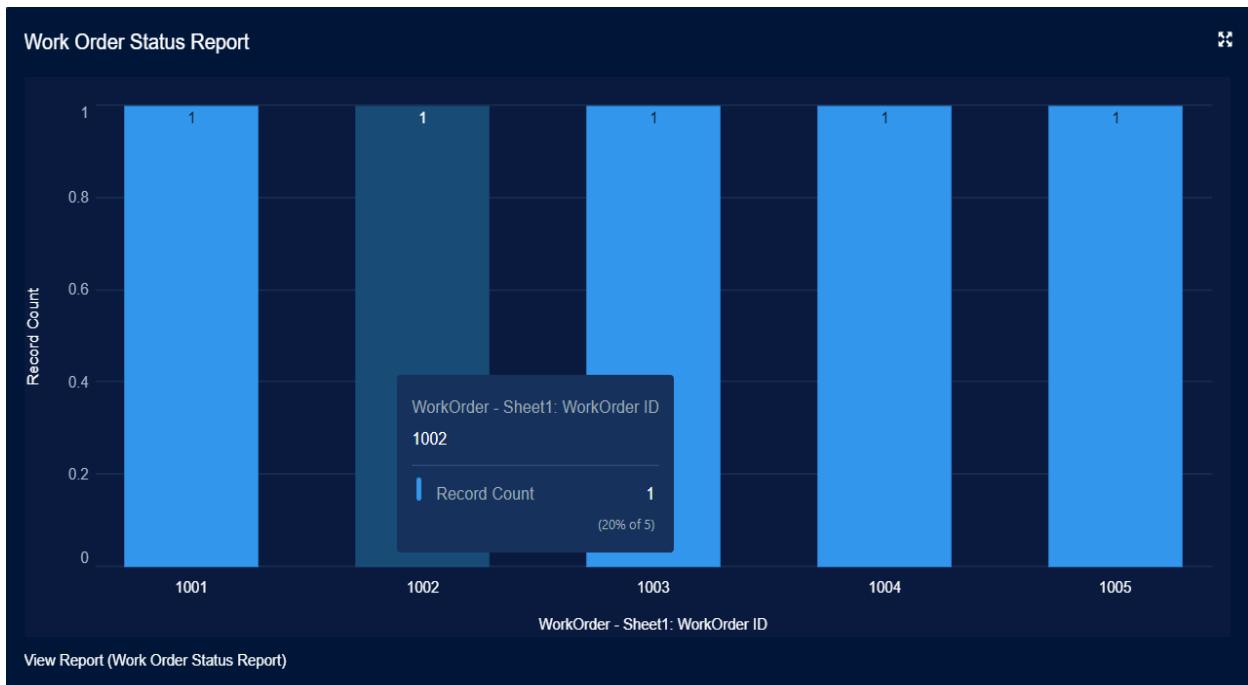
By following these steps, we can create a Salesforce dashboard that visualizes data from the reports you've generated. This dashboard will help you monitor and analyze key metrics related to work orders, technicians, and assignments effectively.

4. Create Dashboards

Create another Dashboard which shows the details of completed workorder status in a vertical bar graph.

- **Create a New Dashboard:**
 - Click on the **New Dashboard** button.
- **Name Your Dashboard:**
 - Enter a name for your dashboard. For example, “ **WorkOrder Status Dashboard**”.
 - Optionally, provide a description to explain the purpose of the dashboard.
- **Click on Create:**
 - After entering the name and description, click **Create** to start building the dashboard.
- **Add a Component to the Dashboard:**

- Click on **+ Add Component** to add a new component to your dashboard.
- **Select the Report:**
 - Choose the report that provides details about completed work orders. For example, select the report type you created previously or a relevant existing report.
 - Click **Select** to choose the report.
- **Configure the Vertical Bar Graph:**
 - **Choose Component Type:** In the component settings, select **Vertical Bar Chart** as the type of chart.
- **Add the Component:**
 - Click **Add** to add the vertical bar graph component to your dashboard.
- **Save the Dashboard:**
 - Once you've configured and added the component, click **Save** to save your dashboard.
- **Finish:**
 - Click **Done** to finish creating your dashboard.



CONCLUSION

The **Field Service Workorder Optimization** project has successfully implemented a comprehensive set of **Salesforce configurations and automations** to enhance field service operations. By creating custom **objects**, defining **relationships**, implementing **Apex classes and triggers**, and setting up **reports** and **dashboards**, this project optimizes the management of work orders and technician assignments. These enhancements improve data accuracy, streamline processes, and provide valuable insights for better decision-making.

This project setup is designed to be scalable and adaptable, supporting ongoing improvements and adjustments based on evolving business needs and operational goals.

THANK YOU

