

WORKFORCE ALLOCATION MANAGEMENT SYSTEM

Devloped by :

TATIPARTHI AMOSH

Email:- 2111CS010040@mallareddyuniversity.ac.in

Project Overview:-

The **WORKFORCE ALLOCATION MANAGEMENT SYSTEM** 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 Management, 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 Management 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.

What you'll learn

1. Real Time Salesforce Project
2. Object & their relationship in Salesforce
3. Page Layout
4. Record Types
5. Validation Rules
6. Triggers
7. Flows
8. Reports
9. Dashboards

Project Flow:

Milestone 1 : Creation of developer account

Milestone 2 : Object Creation

Milestone 3 : Tabs

Milestone 4 : The Lightning App

Milestone 5 : Fields & Relationship

Milestone 6 : Profiles

Milestone 7 : Users

Milestone 8 : Apex Trigger

Milestone 9 : Reports & Dashboards

Milestone 10 : Conclusion

INDEX PAGE

Topics	Page No
Milestone 1-Salesforce	
● Creating Developer Account	5
● Account Activation	6
Milestone 2-Object Creation	
● Create Management Object	6-9
● Create Workorder Object	9-12
● Create Assignment Object	12-13
Milestone 3-Tabs	
● Creating a Custom Tab	14-15
Milestone 4-The Lightning App	
● Create a Lightning App	15-18
Milestone 5-Fields & Relationship	
● Creating Lookup Field In Assignment Object	18-21
● Manage Your Picklist Values	21-22
● Manage Your Picklist Values	23-24
● Creating Formula Field In Workorder Object	24-26
● Creating Remaining Fields for the Respective Objects	26-27

Milestone 6-Profiles

- Management Profile 28-30

Milestone 7-User

- Create User 30-32

Milestone 8-Apex Trigger

- Create an Apex Class 32-34
- Create an Apex Trigger 34-35
- Create an Apex Class 35-36
- Create an Apex Trigger 37
- Create an Apex Class 37-38
- Create an Apex Trigger 38-39
- Create an Asynchronous Apex Class 39-41
- Create an Apex Schedule Class 41
- Create a Schedule Apex 42-43

Milestone 9-Reports & Dashboards

- Reports 43-44
- Create Reports 44-45
- 1. WorkOrders Status Reports 45
- 2. Management and Assignment Details Report 45-49

- Dashboards
- 1. Dashboard 49-50
- 2. WorkOrder Status Dashboard 50-51

Milestone 10-Conclusion

52

INTRODUCTION

Milestone1-Creating Salesforce Developer Org and Account Activation

Activity 1: Creating Salesforce Developer Org

1. Visit Salesforce Developer Site:

- Go to the <https://developer.salesforce.com/signup>
- 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.

Activity 2:Account Activation

1. Login to Salesforce:

- Use your credentials to log in to the Salesforce Developer Org.

Milestone 2-Object Creation

Activity 1: Create Management Object

1. Prepare the Management 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 'Management.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 Management.csv File:

- Navigate to the location where you saved the Management.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 "Management 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

Encoding Format <i>i</i>	Values Separated By	Field Label Source	* Field Labels Row	Import 5 rows of Data? <i>i</i>	Record Name Field <i>i</i>
Unicode (UTF8)	Comma	<input type="radio"/> Enter manually <input checked="" type="radio"/> Detect from row	1	<input type="radio"/> No, skip import <input checked="" type="radio"/> Yes, Import data	Management ID

Fields 7 of 7 to import Hide mapped fields

IMPORT FILE FIELD NAME	SALESFORCE FIELD NAME	SALESFORCE FIELD TYPE	ADD TO LAYOUTS <i>i</i>	FIELD PREVIEW
✓ Management ID	Management ID	Text	<input checked="" type="checkbox"/>	T-0001
✓ Name	Name	Text	<input checked="" type="checkbox"/>	Amosh
✓ Phone	Phone	Phone	<input checked="" type="checkbox"/>	7892341560
✓ Email	Email	Email	<input checked="" type="checkbox"/>	2111cs010040@mailareddyuniiversity.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

Create a custom object from a spreadsheet

Object properties

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

* Label
Management

* Plural Label
Managements

* API Name *i*
Management

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	Management
Fields Detected	7
Rows Detected	5
Fields Created	7
Rows Imported	5

[Import Another Object](#)

Activity 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 (WorkOrders), 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

* Plural Label

* API Name

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	Fields Detected	Fields Created	Rows Detected	Rows Imported
WorkOrder	7	7		

[Import Another Object](#)

Activity 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.

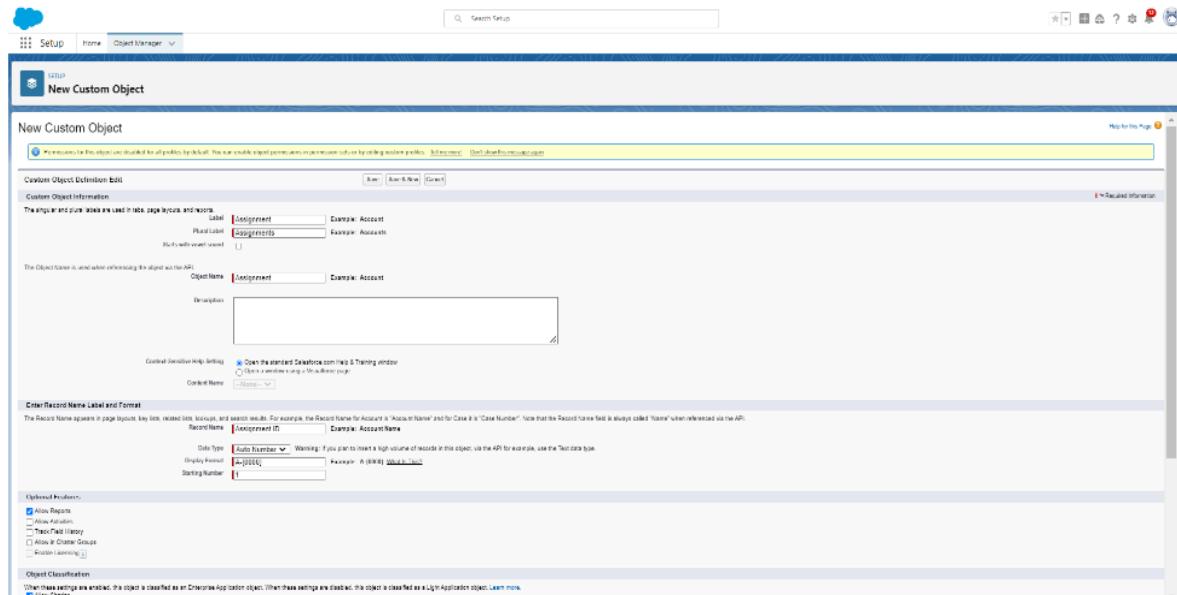
12

● 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}.

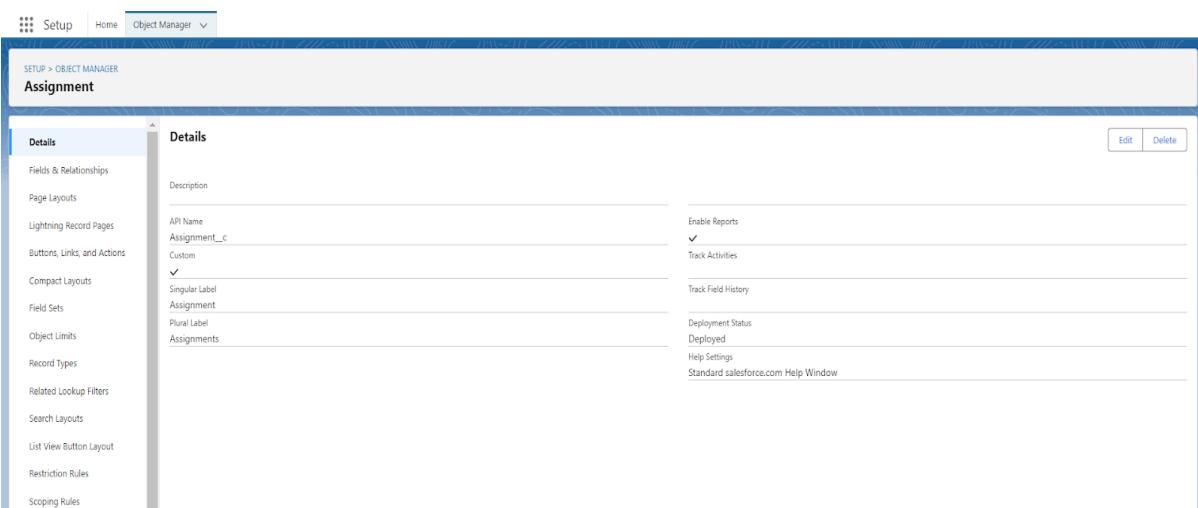


● Configure Optional Features:

- Check the boxes for Allow Reports and Allow Search to enable these features.

● Save the Object:

- Click on Save to create the custom object.



Milestone 3-Tabs

Activity 1: 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.

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

New Custom Object Tab

Step 1. Enter the Details

Select an existing custom object or [create a new custom object now](#).

Action	Label	Tab Style	Description
Edit Del	Assignments	People	
Edit Del	Managements	Box	
Edit Del	WorkOrders	Box	

(Optional) Choose a Home Page Custom Link to show as a splash page the first time your users click on this tab.

Custom Tabs

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

Action	Label	Tab Style	Description
Edit Del	Assignments	People	
Edit Del	Managements	Box	
Edit Del	WorkOrders	Box	

Milestone 4 : The Lightning App

Activity 1: 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.

Lightning Experience App Manager					
App Name		Developer Name		Description	
	Last Modified Date	App Type	Visible...		
1 All Tabs	23/10/2024, 4:55 pm	Classic	✓	All Tabs	
2 Analytics Studio	23/10/2024, 4:55 pm	Classic	✓	Analytics Studio	
3 App Launcher	23/10/2024, 4:55 pm	Classic	✓	App Launcher	
4 Automation	23/10/2024, 4:55 pm	Lightning	✓	Automation	
5 Bolt Solutions	23/10/2024, 4:55 pm	Lightning	✓	Bolt Solutions	
6 Business Rules Engine	23/10/2024, 4:55 pm	Lightning	✓	Business Rules Engine	
7 Community	23/10/2024, 4:55 pm	Classic	✓	Community	
8 Content	23/10/2024, 4:55 pm	Classic	✓	Content	
9 Data Manager	23/10/2024, 4:55 pm	Lightning	✓	Data Manager	
10 Digital Experiences	23/10/2024, 4:55 pm	Lightning	✓	Digital Experiences	
11 Lightning Usage App	23/10/2024, 4:55 pm	Lightning	✓	Lightning Usage App	
12 Marketing CRM Classic	23/10/2024, 4:55 pm	Classic	✓	Marketing CRM Classic	
13 Platform	23/10/2024, 4:55 pm	Classic	✓	Platform	
14 Queue Management	23/10/2024, 4:55 pm	Lightning	✓	Queue Management	
15 Sales	23/10/2024, 4:55 pm	Classic	✓	Sales	
16 Sales	23/10/2024, 4:55 pm	Lightning	✓	LightningSales	
17 Sales Console	23/10/2024, 4:55 pm	Lightning	✓	Sales Console	
18 Salesforce Chatter	23/10/2024, 4:55 pm	Classic	✓	Chatter	
19 Salesforce Scheduler Setup	23/10/2024, 4:55 pm	Lightning	✓	Salesforce Scheduler Setup	
20 Service	23/10/2024, 4:55 pm	Classic	✓	Service	

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.

New Lightning App

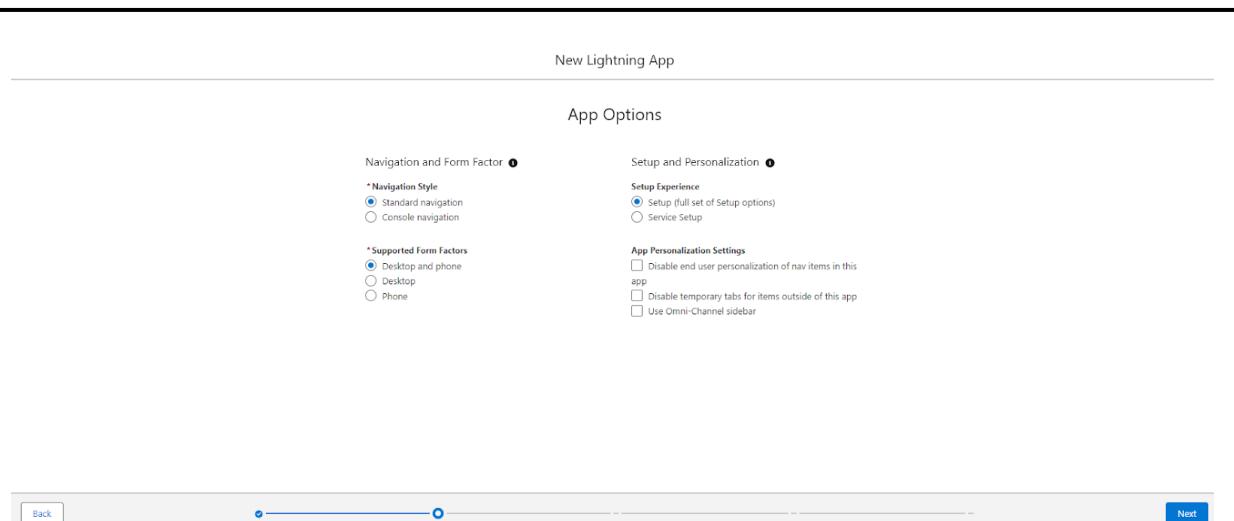
App Details & Branding

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

App Details	App Branding
*App Name <input type="text" value="Workforce Allocation Management System"/> *Developer Name <input type="text" value="Workforce_Allocation_Management_System"/> Description <input type="text" value="The Workforce Allocation Management System serves as a centralized platform for managing..."/>	Image <input type="file" value="Upload"/> Primary Color Hex Value <input type="text" value="#0070D2"/> Org Theme Options <input type="checkbox"/> Use the app's image and color instead of the org's custom theme
App Launcher Preview	

4. App Options:

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

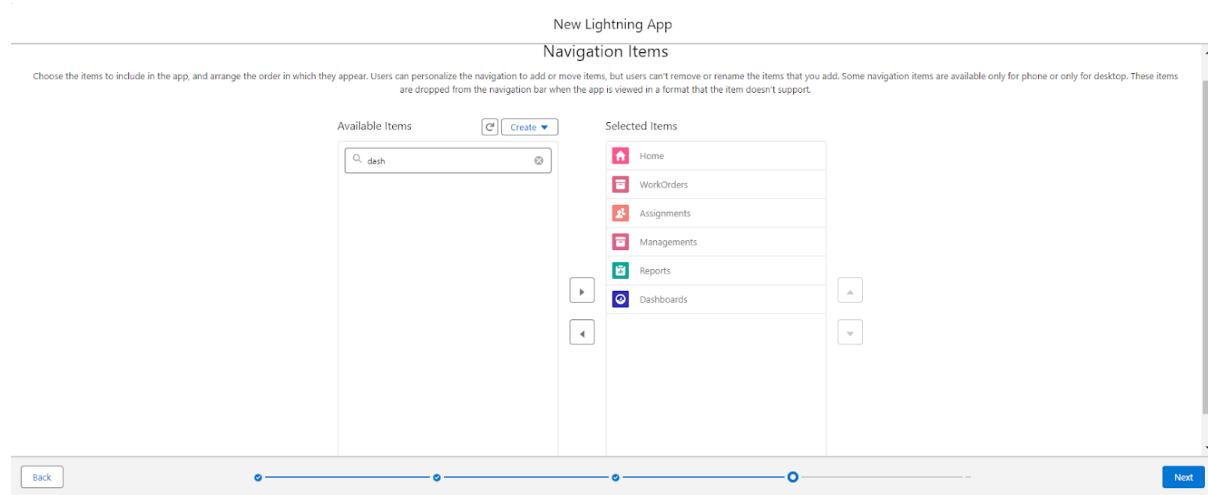


5. Utility Items:

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

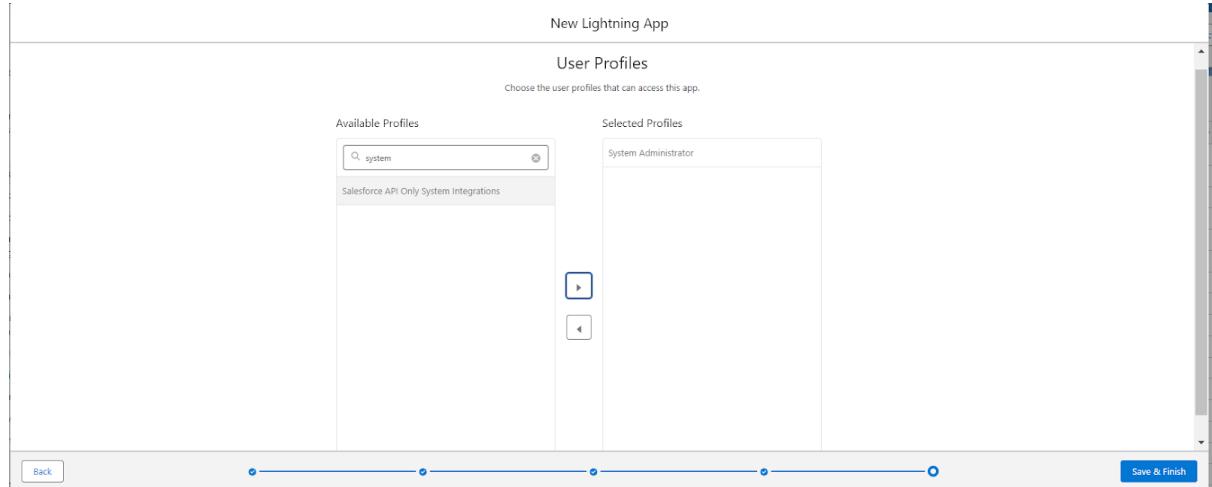
6. Add Navigation Items:

- 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 (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.

Lightning Experience App Manager						
App Name ↑		Developer Name		Description		
App Name	Developer Name	Description	Last Modified Date	App Type	Visible	
5 Bolt Solutions	LightningBolt	Discover and manage business solutions designed for your industry.	23/10/2024, 4:55 pm	Lightning	✓	
6 Business Rules Engine	ExpressionSetConsole	Create and maintain business rules that perform complex lookups and calculations.	23/10/2024, 4:55 pm	Lightning	✓	
7 Community	Community	Salesforce CRM Communities	23/10/2024, 4:55 pm	Classic	✓	
8 Content	Content	Salesforce CRM Content	23/10/2024, 4:55 pm	Classic	✓	
9 Data Manager	DataManager	Use Data Manager to view limits, monitor usage, and manage recipes.	23/10/2024, 4:55 pm	Lightning	✓	
10 Digital Experiences	SalesforceCMS	Manage content and media for all of your sites.	23/10/2024, 4:55 pm	Lightning	✓	
11 Lightning Usage App	LightningInstrumentation	View Adoption and Usage Metrics for Lightning Experience	23/10/2024, 4:55 pm	Lightning	✓	
12 Marketing CRM Classic	Marketing	Track sales and marketing efforts with CRM objects.	23/10/2024, 4:55 pm	Classic	✓	
13 Platform	Platform	The fundamental Lightning Platform	23/10/2024, 4:55 pm	Classic	✓	
14 Queue Management	QueueManagement	Create and manage queues for your business.	23/10/2024, 4:55 pm	Lightning	✓	
15 Sales	Sales	The world's most popular sales force automation (SFA) solution	23/10/2024, 4:55 pm	Classic	✓	
16 Sales	LightningSales	Manage your sales process with accounts, leads, opportunities, and more	23/10/2024, 4:55 pm	Lightning	✓	
17 Sales Console	LightningSalesConsole	(Lightning Experience) Lets sales reps work with multiple records on one screen	23/10/2024, 4:55 pm	Lightning	✓	
18 Salesforce Chatter	Chatter	The Salesforce Chatter social network, including profiles and feeds	23/10/2024, 4:55 pm	Classic	✓	
19 Salesforce Scheduler Setup	LightningScheduler	Set up personalized appointment scheduling.	23/10/2024, 4:55 pm	Lightning	✓	
20 Service	Service	Manage customer service with accounts, contacts, cases, and more	23/10/2024, 4:55 pm	Classic	✓	
21 Service Console	LightningService	(Lightning Experience) Lets support agents work with multiple records across customer service channels on one screen	23/10/2024, 4:55 pm	Lightning	✓	
22 Site.com	Sites	Build pixel-perfect, data-rich websites using the drag-and-drop Site.com application, and manage content and published sites.	23/10/2024, 4:55 pm	Classic	✓	
23 Subscription Management	RevenueCloudConsole	Get started automating your revenue processes	23/10/2024, 4:55 pm	Lightning	✓	
24 Workforce Allocation Manage...	Workforce_Allocation_Mana...	The Workforce Allocation Management System serves as a centralized platform for managing workforce resources, helping organ...	23/10/2024, 6:21 pm	Lightning	✓	

Milestone 5 : Fields & Relationship

Activity 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.

The screenshot shows the 'Assignment' object setup in the Salesforce Object Manager. The left sidebar has 'Field & Relationships' selected. The main area displays a table titled 'Fields & Relationships' with four items. The columns are 'FIELD LABEL', 'FIELD NAME', 'DATA TYPE', 'CONTROLLING FIELD', and 'INDEXED'. The data is as follows:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Assignment ID	Name	Auto Number		✓
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	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.

The screenshot shows the 'Step 1. Choose the field type' page for creating a new custom field. The left sidebar has 'Field & Relationships' selected. The main area is titled 'Assignment New Custom Field'. It shows a 'Data Type' section with several options:

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

 Detailed descriptions for each option are provided:

- 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 pop-up list. The other object is the source of the values in the list.
- Lookup Relationship: Creates a special type of parent-child relationship between this object (the child, or "detail") and another object (the parent, or "master") where:
 - The relationship field is required on all detail records.
 - The sharing and seeding of a detail record is controlled by the master record.
 - A user-defined master field of detail records are displayed.
 - You can create many 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 pop-up list. The master object is the source of the values in the list.
- Master-Detail Relationship: Creates a relationship that links this object to an external object whose data is stored outside the Salesforce org.
- External Lookup Relationship: Not described in the screenshot.

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).

Assignment
New Relationship

Step 2. Choose the related object

Select the other object to which this object is related.

Related To: WorkOrder

5. Define Field Label:

- Field Label: Enter WorkOrder ID.
- Click Next

Step 3. Enter the label and name for the lookup field

Field Label: WorkOrder ID

Field Name: WorkOrder_ID

Description:

Help Text:

Child Relationship Name: Assignments

Required: Always require a value in this field in order to save a record

What to do if the lookup record is deleted? Clear the value of this field. You can't choose this option if you make this field required.

Don't allow deletion of the lookup record that's part of a lookup relationship.

Auto add to custom report type: Add this field to existing custom report types that contain this entry

6. Finalize the Field Setup:

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

Step 4. Establish field-level security for reference field

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"/>
Contact 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"/>
Force.com - App Subscription User	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Force.com - Free User	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Gold Partner User	<input type="checkbox"/>	<input type="checkbox"/>
Identity User	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Marketing User	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Minimum Access - API Only Integrations	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Minimum Access - Salesforce	<input checked="" type="checkbox"/>	<input type="checkbox"/>

The top screenshot shows Step 5 of 6: "Add reference field to Page Layouts". It details a field named "WorkOrder ID" with type "Lookup" and name "WorkOrder_ID". It lists two page layouts: "Assignment Layout" and "Assignment Layout".

The bottom screenshot shows Step 6 of 6: "Add custom related lists". It details a related list titled "Assignments" for the "WorkOrder ID" field. It lists one layout: "WorkOrder Layout".

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

The screenshot shows the "Fields & Relationships" section for the "Assignment" object. It lists five fields:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Assignment ID	Name	Auto Number		✓
Created By	CreatedById	Lookup(User)		✓
Last Modified By	LastModifiedById	Lookup(User)		✓
Owner	OwnerId	Lookup(User/Group)		✓
WorkOrder ID	WorkOrder_ID_c	Lookup(WorkOrder)		✓

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

Activity 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

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The left sidebar lists various object settings like 'Details', 'Fields & Relationships' (which is selected), 'Page Layouts', 'Lightning Record Pages', etc. The main content area is titled 'Add Picklist Values' under 'Location'. It contains instructions: '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.' Below this is a text input field containing the values 'Nasik', 'Warangal', and 'Nanded', each on a new line. At the bottom are 'Save' and 'Cancel' buttons.

Save

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

The screenshot shows the 'Values' list for the Location field. The top navigation bar includes 'New', 'Reorder', 'Replace', 'Printable View', 'Chart Colors', and 'Values Help'. The table has columns: Action, Values, API Name, Default, Chart Colors, and Modified By. The data rows are:

Action	Values	API Name	Default	Chart Colors	Modified By
<input type="checkbox"/>	Edit Del Deactivate Value1	Value1	<input type="checkbox"/>	Assigned dynamically	Tatiparthi Amosh, 23/10/2024, 6:04 pm
<input type="checkbox"/>	Edit Del Deactivate Nasik	Nasik	<input type="checkbox"/>	Assigned dynamically	Tatiparthi Amosh, 23/10/2024, 6:34 pm
<input type="checkbox"/>	Edit Del Deactivate Warangal	Warangal	<input type="checkbox"/>	Assigned dynamically	Tatiparthi Amosh, 23/10/2024, 6:34 pm
<input type="checkbox"/>	Edit Del Deactivate Nanded	Nanded	<input type="checkbox"/>	Assigned dynamically	Tatiparthi Amosh, 23/10/2024, 6:34 pm

Activity 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".

The screenshot shows the Salesforce Object Manager interface for the WorkOrder object. On the left, a sidebar lists various setup options like Details, Fields & Relationships, Page Layouts, etc. The main area displays the 'Priority' field definition. It includes sections for Field Information (Field Label: Priority, Field Name: Priority, API Name: Priority_c), General Options (Required: checked, Default Value: High), and a note about Validation Rules. At the bottom right, it shows the object name (WorkOrder) and data type (Picklist). A 'Help for this Page' link is also present.

Click "Save".

The screenshot shows the 'Add Picklist Values' page for the Priority field. The sidebar remains the same. The main area has a heading 'Add Picklist Values' and a note: '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.' Below this is a text input field containing the value 'High'. At the bottom are 'Save' and 'Cancel' buttons.

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

The screenshot shows the Salesforce Object Manager interface for the WorkOrder object. A new custom field named 'Service Type' is being created. The 'Field & Relationships' tab is selected. The 'Custom Field Definition Detail' section shows the following details:

- Field Information:**
 - Field Label: Service Type
 - Field Name: Service_Type
 - API Name: Service_Type__c
 - Description: (empty)
 - Help Text: (empty)
 - Data Owner: (empty)
 - Field Usage: (empty)
 - Data Sensitivity Level: (empty)
 - Compliance Categorization: (empty)
- General Options:**
 - Required:
 - Default Value: (empty)

At the top right, there are buttons for 'Edit', 'Set Field-Level Security', 'View Field Accessibility', and 'Where is this used?'. The status bar at the bottom indicates 'Object Name: WorkOrder' and 'Data Type: Picklist'. The page title is 'SETUP > OBJECT MANAGER WorkOrder'.

- Click "Save."

The screenshot shows the same Salesforce Object Manager interface after saving the 'Service Type' field. The 'Fields & Relationships' tab is still selected. The 'Add Picklist Values' section now contains three entries:

- Hardware repair
- Troubleshoot/Debugging
- Lane-Management

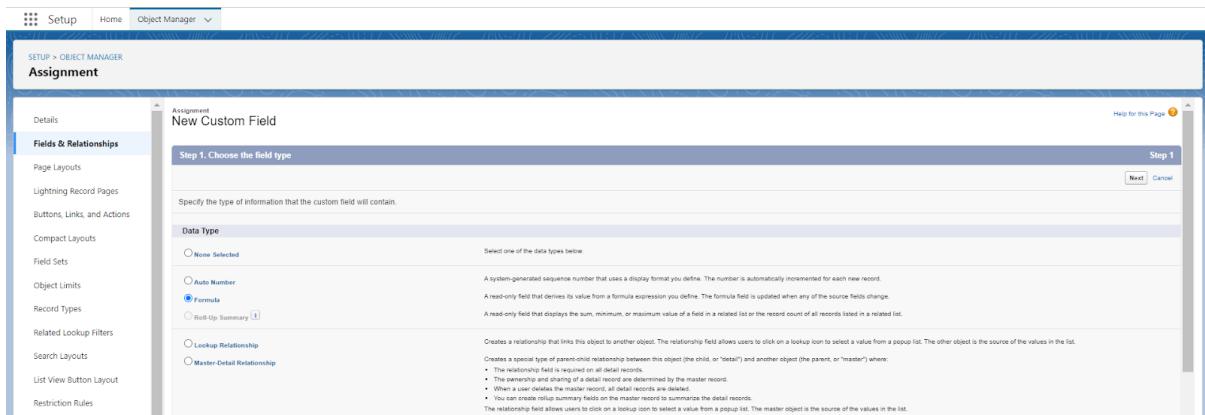
At the bottom right of the input area, there are 'Save' and 'Cancel' buttons. The page title is 'SETUP > OBJECT MANAGER WorkOrder'.

Activity 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.

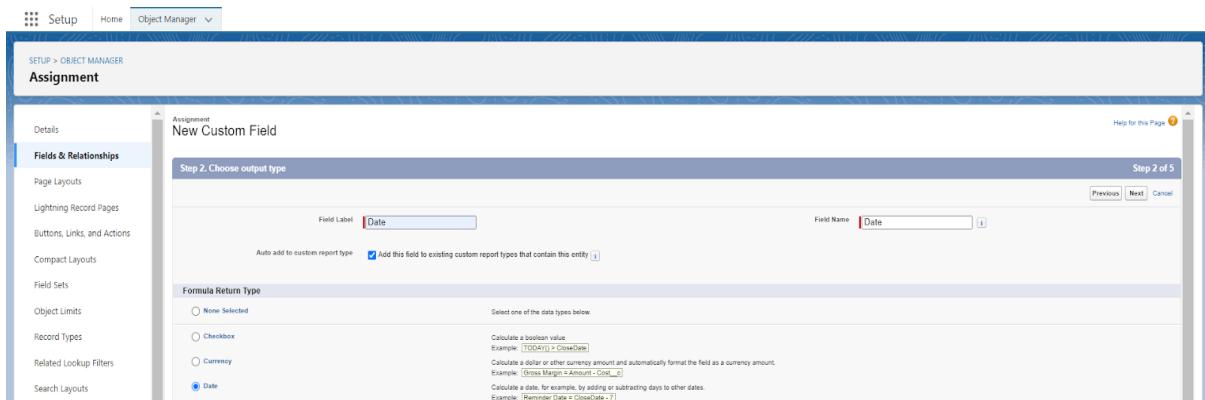
2. Create a New Field:

- Click "New" to create a new field.
- Select Data Type as "Formula" and click "Next."



- **Enter Field Details:**

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



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 Object Manager interface for creating a new field. The object is 'Assignment'. In the 'Fields & Relationships' section, a new formula field is being created. The formula is set to 'Date (Date) = CreatedDate'. A dropdown menu for functions is open, showing options like ABS, ACOS, ADDMONTHS, AND, ASCII, ASIN, etc. A note at the bottom says '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 'Step 5. Add to page layouts' screen of the custom field creation wizard. It's step 5 of 5. The field is named 'New Custom Field' and is of type 'Date'. Under 'Page Layouts', two checkboxes are selected: 'Add Field' and 'Assignment Layout'. A note says 'When finished, click Save & New to create more custom fields, or click Save if you are done.' Buttons at the bottom include 'Previous', 'Save & New', 'Save', and 'Cancel'.

Activity 5: Creating Remaining Fields For The Respective Objects

1. Management ID Field

- **Field Type:** Lookup
- **Related To:** Management
- **Steps:**
 - a. Go to the **Fields & Relationships** section in the Assignment object.
 - b. Click "**New**."
 - c. Select Data Type as "**Lookup Relationship**" and click "Next."

- d. Choose Related To as "**Management**" and click "Next."
- e. Set Field Label as "**Management ID**" (Field Name auto-populates).
- f. Click "**Next**" and complete the remaining steps.
- g. Click "**Save**".

2. Assignment Date Field

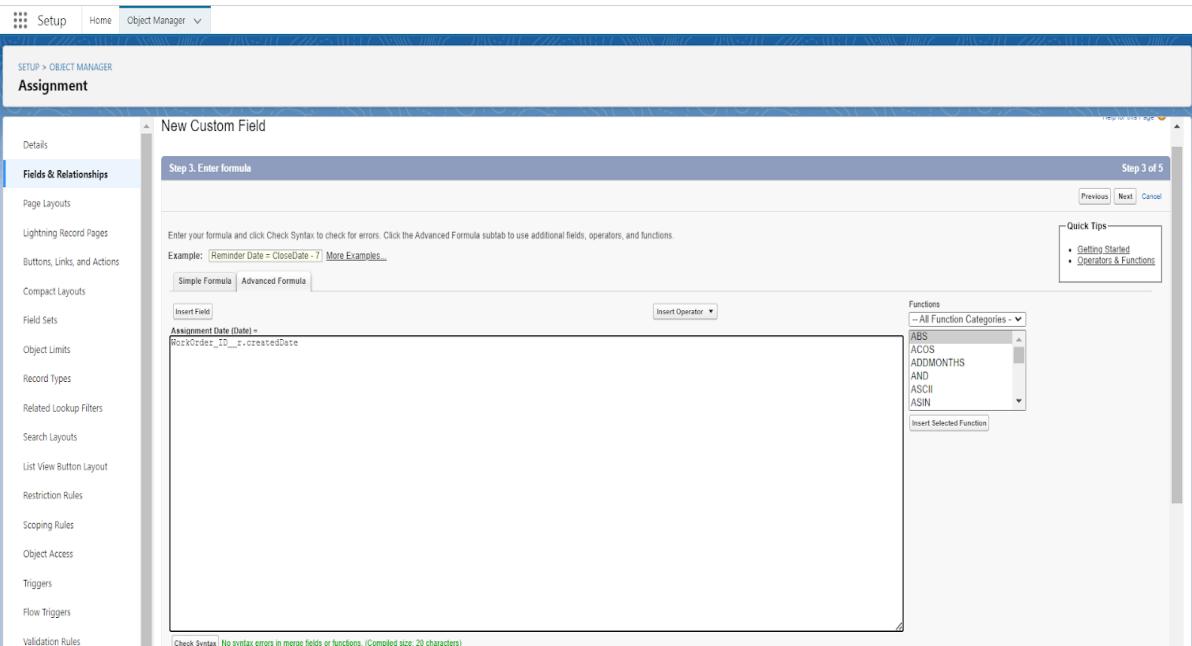
- **Field Type:** Formula
- **Return Type:** Date
- **Formula:** WorkOrder_ID__r.Date__c
- **Steps:**

- a. Click "New" in the Fields & Relationships section.
- b. Select Data Type as "**Formula**" and click "Next."
- c. Set Field Label as "**Assignment Date**" (Field Name auto-populates).
- d. Choose Formula Return Type as "**Date**" and click "**Next**".

e. In the Advanced Formula section, enter the formula:

WorkOrder_ID__r.Date__c

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



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

Milestone 6 : Profiles

Activity 1: Creating Management 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 Management as the Profile Name.

The screenshot shows the 'Clone Profile' page in the Salesforce Setup. The 'Existing Profile' dropdown is set to 'Standard Platform User'. The 'Profile Name' field contains 'Management'. At the bottom, there are 'Save' and 'Cancel' buttons.

- Click "Save."

The screenshot shows the 'Profile Management' page in the Salesforce Setup. The 'Name' field is set to 'Management'. The 'User License' dropdown is set to 'Salesforce Platform'. Under 'Page Layouts', it lists 'Standard Object Layouts' for various objects like Account, Contact, etc., with their respective global and email application layouts. Under 'Object Permissions', it lists permissions for various objects like Case, Lead, Opportunity, etc., with their respective global and email application permissions.

2. Edit Profile Permissions

1. Edit the Management Profile:

- While still on the Management profile page, click "Edit."

2. Set Custom Object Permissions:

- Scroll down to the Custom Object Permissions section.
- Set Read Only access for the Management, WorkOrder, and Assignment objects.

3. Save the Profile:

- Scroll down and click "Save."

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.

1. Edit Status Field Security:

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

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"/>

2. Save Changes:

- Click "Save" to apply the changes.

The screenshot shows the Salesforce Setup interface with the following details:

- Setup** tab selected in the top navigation bar.
- Profiles** section selected in the left sidebar under **Users**.
- WorkOrder Field-Level Security for profile Management** is displayed in the main content area.
- A table lists field-level security settings for the Management profile. The columns are: Field Name, Field Type, Read Access, and Edit Access.
- The table includes fields such as Created By, Description, Email, Last Modified By, Location, Owner, Priority, Service Type, Status, and WorkOrder ID.
- Checkmarks indicate which fields have specific access levels assigned.

Milestone 7 : Users

Activity 1: Create User

To create a new user with the **Management** 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 Management.

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.

Milestone 8 : Apex Trigger

Activity 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<Management__c> managementtoAssignment = new  
        List<Management__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,Management__c> allManagement = new  
        Map<Id,Management__c>([SELECT Id, Name, Phone__c, Location__c,  
        Skills__c, Availability__c, Name__c, Email__c FROM Management__c]);
```

```

integer num2 = 0;
For(Management__c T : allManagement.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.Management_ID__c = techid.get(num3);
    lstAssignment.add(A);
}
If(!lstAssignment.IsEmpty()){
    insert lstAssignment;
}
}
}

```

```

1 public class WorkOrderClass {
2     public static void workOrder(List<WorkOrder__c> newListWorkOrder){
3         Map<Integer, List<String>> mapTech = 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<Management__c> managementToAssignment = new List<Management__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                mapTech.put(num,lstString);
17            }
18        }
19        Map<integer,Id> techId = new Map<integer,Id>();
20        Map<Id,Management__c> allManagement = new Map<Id,Management__c>([SELECT Id, Name, Phone__c, Location__c, Skills__c, Availability__c, Name__c, Email__c FROM Management__c]);
21        integer num2 = 0;
22        For(Management__c t : allManagement.values()){
23            num2 = num2+1;
24            if(maptotech.get(num) != null){
25                List<string> valOfMap = mapTech.get(num);
26                system.debug('error 1 ----> the maptoTech is empty ...> ' + mapTech.get(num2));
27                if(valOfMap.contains(t.skills__c) && valOfMap.contains(t.Location__c) && t.Availability__c == 'Available'){
28                    techId.put(num2,t.Id);
29                }
30            }
31        }
32        integer num3 = 0;
33        For(WorkOrder__c W : properWo){
34            num3 = num3 + 1;
35            Assignment__c A = new Assignment__c();
36            A.WorkOrder_ID__c = W.Id;
37            A.Management_ID__c = techId.get(num3);
38            lstAssignment.add(A);
39        }
40    }
41    If(!lstAssignment.IsEmpty()){
42        insert lstAssignment;
43    }
44 }

```

Activity 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:

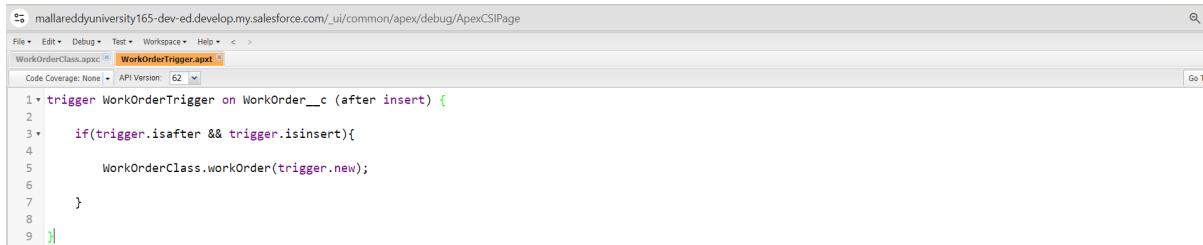
```

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);
    }
}
```

Activity 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,Management__c> managements = new
        Map<id,Management__c>([SELECT Id, Phone__c, Location__c, Skills__c,
        Name__c, Email__c, Availability__c, Name FROM Management__c]);
        try{
            for(Assignment__c con : assRec){
                if(con.Management_ID__c != null){
                    messaging.SingleEmailMessage mail = new
                    messaging.SingleEmailMessage();
                    List<String> sendTo = new List<String>();
                    sendTo.add(managements.get(con.Management_ID__c).Email__c);
                    mail.setToAddresses(sendTo);
                    string subject = 'WorkOrder Assignment ';
```

```

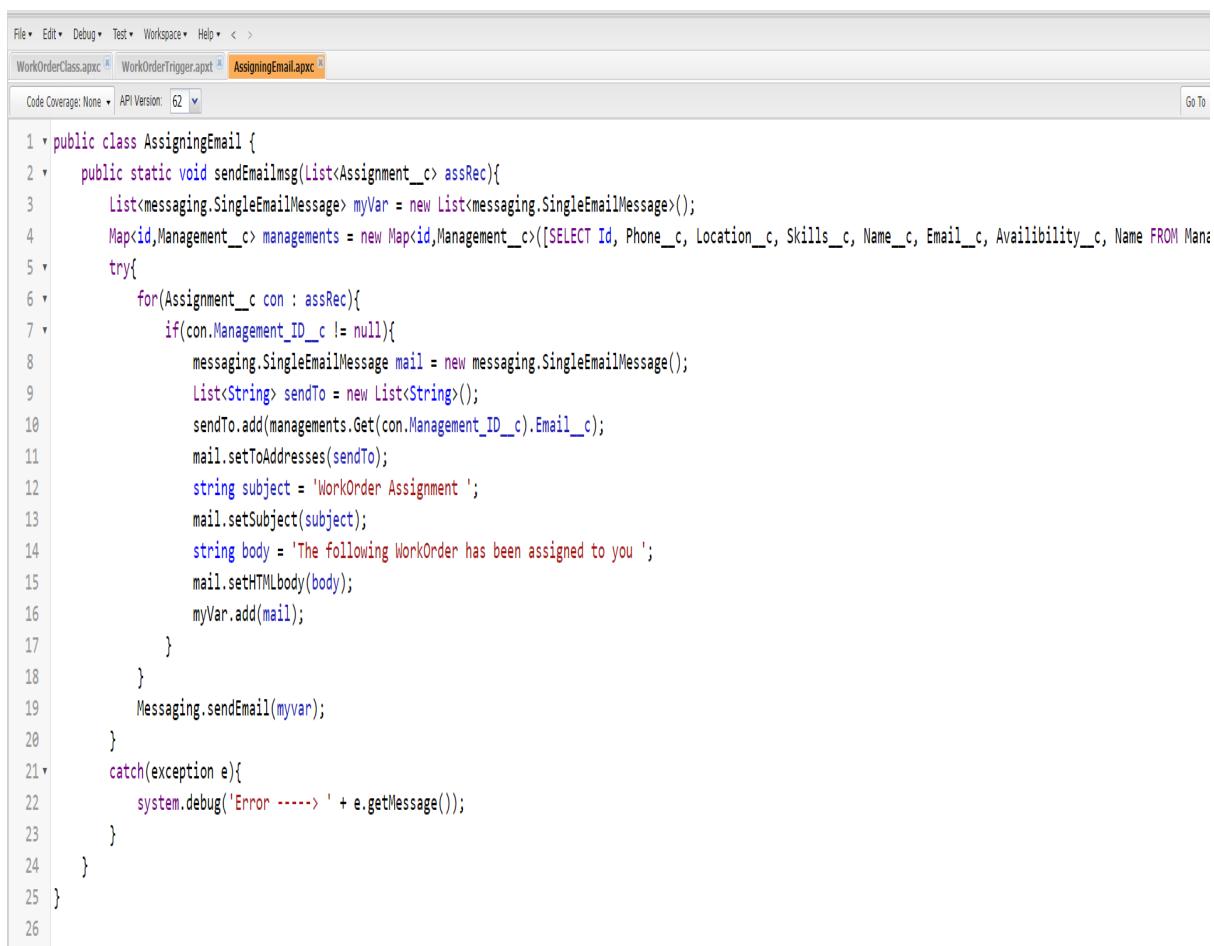
        mail.setSubject(subject);
        string body = 'The following WorkOrder has been assigned to you ';
        mail.setHTMLbody(body);
        myVar.add(mail);
    }
}

Messaging.sendEmail(myvar);
}
catch(exception e){
    system.debug('Error -----> ' + e.getMessage());
}
}
}

```

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,Management__c> managements = new Map<id,Management__c>([SELECT Id, Phone__c, Location__c, Skills__c, Name__c, Email__c, Availability__c, Name FROM Management__c]);
5         try{
6             for(Assignment__c con : assRec){
7                 if(con.Management_ID__c != null){
8                     messaging.SingleEmailMessage mail = new messaging.SingleEmailMessage();
9                     List<String> sendTo = new List<String>();
10                    sendTo.add(managements.Get(con.Management_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 }

```

Activity 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);  
}  
}
```



Activity 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') {
```

```

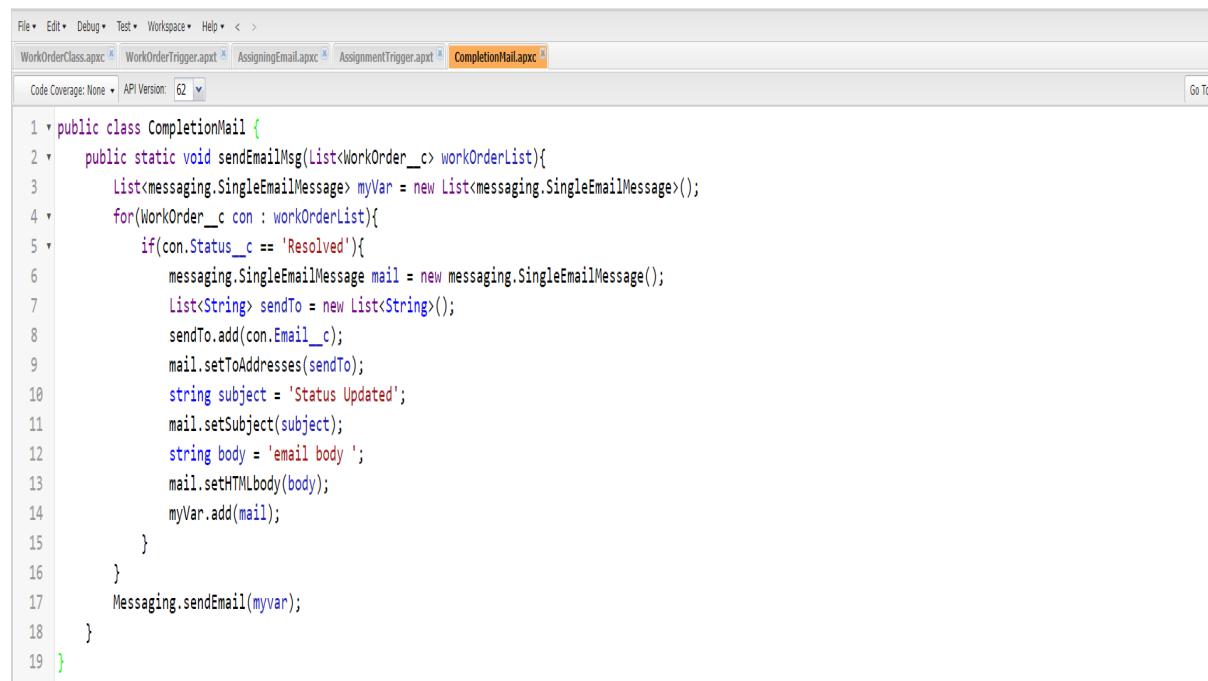
messaging.SingleEmailMessage mail = new
messaging.SingleEmailMessage();
List<String> sendTo = new List<String>();
sendTo.add(con.Email__c);
mail.setToAddresses(sendTo);
String subject = 'Status Updated';
mail.setSubject(subject);
String body = 'Your WorkOrder has been resolved.';
mail.setHTMLBody(body);
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 'WorkOrderClass.apxc', 'WorkOrderTrigger.apxt', 'AssigningEmail.apxc', 'AssignmentTrigger.apxt', and 'CompletionMail.apxc' (which is highlighted). Below the tabs, there's a dropdown for 'Code Coverage: None' and 'API Version: 62'. On the right, there's a 'Go To' button. The main area contains the Apex code for the 'CompletionMail' class:

```

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

```

Activity 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
Code Coverage: None API Version: 62 Go To
1 trigger WorkOrderTrigger on WorkOrder__c (after insert) {
2
3 if(trigger.isafter && trigger.isinsert){
4
5     WorkOrderClass.workOrder(trigger.new);
6
7 }
8 if(Trigger.IsAfter && Trigger.IsUpdate){
9     CompletionMail.sendEmailMsg(Trigger.New);
10 }
11 }
```

Save the Trigger:

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

Activity 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 batch
    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.apxc AssigningEmail.apxc AssignmentTrigger.apxc CompletionMail.apxc RecordDeletions.apxc
Code Coverage: None API Version 62 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, Management_ID__c, Assignment_Date__c, Completion_Date__c FROM Assignment__c WHERE Completion_Date__c = LAST_N_DAYS:30';
6
7         return database.GetQueryLocator(query);
8     }
9
10    public void execute(Database.BatchableContext bc, List<Assignment__c> query){
11
12        if(!query.IsEmpty()){
13
14            Delete query;
15
16        }
17
18    }
19
20    public void finish(Database.BatchableContext bc){
21
22    }
23
24 }

```

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);
```

Activity 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 < >
WorkOrderClass.apxc WorkOrderTrigger.apxt AssigningEmail.apxc AssignmentTrigger.apxt CompletionMail.apxc RecordDeletions.apxc ScheduleClass.apxc
Code Coverage: None API Version: 62 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 }
```

Activity 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. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. A search bar on the left contains 'apex class'. Under 'Custom Code', 'Apex Classes' is selected. A message says 'Didn't find what you're looking for? Try using Global Search.' Below this, a progress bar indicates 'Percent of Apex Used: 0.09%' and 'You are currently using 5,241 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.' A 'Schedule Apex' button is located at the top right of the main content area. The main content displays a table of Apex classes with columns: Action, Name, Namespace Prefix, Api Version, Status, Size Without Comments, Last Modified By, and Has Trace Flags. The table lists several classes like AssignmentEmail, CompletionMail, RecordDeletions, ScheduleClass, WorkOrderClass, and WorkOrderTroyer, all marked as Active. At the bottom, there's a section for 'Dynamic Apex Classes' with a note about interacting with Lightning Platform components, a table header for 'Class Name', 'Namespace Prefix', 'Api Version', 'Created By', and 'Last Modified By', and a message stating 'No records to display.'

• Enter Job Details:

- Job Name: Enter DeleteAssignmentSchedule.
- Apex Class: Click on the lookup icon (Q) 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.

The screenshot shows the 'Schedule Apex' page in the Salesforce Setup. The job name is set to 'DeleteAssignmentSchedule' and the apex class is 'ScheduleClass'. The frequency is set to 'Monthly' on the first day of every month. The start date is '23/10/2024' and the end date is '23/11/2024'. The preferred start time is '12:00 pm'.

Milestone 9 : Reports & Dashboards

Activity 1: Report

Steps to Create a Report

1. Access the App:

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

The dashboard shows a 'Quarterly Performance' chart with data for Closed (\$6,95,000), Open (>70%) (\$2,70,000), and Goal. Below the chart are sections for 'Today's Events' (free and clear) and 'Today's Tasks' (nothing due). The 'Recent Records' section shows a recent record for Eline Gilbert. The 'Key Deals - Recent Opportunities' section indicates no deals yet. The 'Assistant' sidebar lists new lead assignments for various users.

- **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 "Report Assignments with WorkOrder ID" under the "New Assignments with WorkOrder ID Report" section. The report displays 10 total records. The columns include Assignment ID, Assignment Assignment ID, WorkOrder ID, WorkOrder Email, WorkOrder Service Type, WorkOrder Location, and WorkOrder Record ID. The data rows show various assignments with their respective details such as WorkOrder ID (e.g., A-0010, A-0008, A-0005, etc.), Email (e.g., ravi@gmail.com, 004@gmail.com, rya@gmail.com, amosh@gmail.com, etc.), Service Type (e.g., Hardware repair, TroubleshootDebugging), Location (e.g., Nanded, Warangal, Nasik), and Record ID (e.g., a01Qy00000a5hQ3, a01Qy00000a5hQ5L, a01Qy00000a5hU8, etc.).

Total Records	10
Report Assignments with WorkOrder ID	
New Assignments with WorkOrder ID Report	
Enable Field Editing	Q Add Chart
C	Edit
Assignment ID	A-0010
Assignment Assignment ID	W-112
WorkOrder ID	ravi@gmail.com
WorkOrder Email	Hardware repair
WorkOrder Service Type	Nanded
WorkOrder Location	a01Qy00000a5hQ3
WorkOrder Record ID	
Subtotal	
a01Qy000005221 (1)	A-0008
W-008	004@gmail.com
TroubleshootDebugging	Warangal
	a01Qy00000a5hQ5L
Subtotal	
a01Qy00000525v (1)	A-0005
W-115	rya@gmail.com
TroubleshootDebugging	Nasik
	a01Qy00000a5hU8
Subtotal	
a01Qy0000059eq (1)	A-0003
W-008	004@gmail.com
TroubleshootDebugging	Warangal
	a01Qy00000a5hQ5L
Subtotal	
a01Qy000005aeH (1)	A-0001
W-001	amosh@gmail.com
Hardware repair	Nasik
	a01Qy00000a5gQz
Subtotal	
a01Qy000005auP (1)	A-0002
W-002	raju@gmail.com
TroubleshootDebugging	Nanded
	a01Qy00000a5QzV
Subtotal	
a01Qy000005bev (1)	A-0004
W-112	ravi@gmail.com
Hardware repair	Nanded
	a01Qy00000a5hQ3
Subtotal	
a01Qy000005bk1 (1)	A-0006
W-001	amosh@gmail.com
Hardware repair	Nasik
	a01Qy00000a5gQz
Subtotal	
a01Qy000005bqT (1)	A-0007
W-002	raju@gmail.com
TroubleshootDebugging	Nanded
	a01Qy00000a5hQ5L
Subtotal	
a01Qy00000554z (1)	A-0009
W-115	rya@gmail.com
TroubleshootDebugging	Nasik
	a01Qy00000a5hU8
Subtotal	
Total (10)	

Activity 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.

The screenshot shows the Salesforce interface for a 'Report WorkOrder Status Report'. The top navigation bar includes 'Workforce Allocatio...', 'Home', 'WorkOrder', 'Assignments', 'Managements', 'Reports' (which is selected), and 'Dashboards'. Below the navigation is a toolbar with 'Enable Field Editing', 'Search', 'Add Chart', 'Print', 'Edit', and 'Download' buttons. The main area displays a table titled 'Report WorkOrder Status Report' with 5 total records. The columns are: WorkOrder: WorkOrder ID, Email, Location, Service Type, Priority, Status, and WorkOrder: Created Date. The data rows are:

	WorkOrder: WorkOrder ID	Email	Location	Service Type	Priority	Status	WorkOrder: Created Date
W-001 (1)	amosh@gmail.com	Nasik	Hardware repair	High	Value1		24/10/2024
Subtotal							
W-002 (1)	ravi@gmail.com	Nanded	Troubleshoot/Debugging	High	Value1		24/10/2024
Subtotal							
W-006 (1)	004@gmail.com	Warangal	Troubleshoot/Debugging	High	Value1		24/10/2024
Subtotal							
W-112 (1)	ravi@gmail.com	Nanded	Hardware repair	High	Value1		24/10/2024
Subtotal							
W-115 (1)	riya@gmail.com	Nasik	Troubleshoot/Debugging	High	Value1		24/10/2024
Subtotal							
Total (5)							

ii). Create a report with report type: "Management 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 Management__c as the primary object.
- Report Type Label: Enter a name like "Management 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".

The screenshot shows the 'Report Types' setup page. At the top, there's a 'SETUP' icon and the title 'Report Types'. Below the title, there's a section titled 'Report Type Focus' with a note about specifying the primary object. The 'Primary Object' dropdown is set to 'Management'. The 'Identification' section contains fields for 'Report Type Label' (set to 'Management and Assignment Details'), 'Report Type Name' (set to 'Management_and_Assignn'), and 'Description' (set to 'Management and Assignment Details Report'). The 'Store in Category' dropdown is set to 'Other Reports'. Under the 'Deployment' section, it says 'A report type with deployed status is available for use in the report wizard. While in development, report types are visible only to authorized administrators and their delegates.' The 'Deployment Status' radio buttons are set to 'In Development'.

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 Management__c, select "Each Management__c record may or may not have related Assignment__c records".

SETUP Report Types

New Custom Report Type Management and Assignment Details Help for this Page ?

Step 2. Define Report Records Set Step 2 of 2

This report type will generate reports about Managements. You may define which related records from other objects are returned in report results by choosing a relationship to another object.

A Managements Primary Object

B Assignments

A to B Relationship:

- Each "A" record must have at least one related "B" record.
- "A" records may or may not have related "B" records.

(Click to relate another object)

Previous Save Cancel

SETUP Report Types

Custom Report Type Management and Assignment Details Help for this Page ?

Management and Assignment Details

Below is the information for this custom report type. You can click the buttons on this page to preview or update information for the custom report type.

Custom Report Type Definition		Report Type Category	Other Reports
Report Type Label	Management and Assignment Details	Report Type Category	Other Reports
Report Type Name	Management_and_Assignment_Details	Deployment Status	In Development
Description	Management and Assignment Details Report	Created By	Tatiparthi Amosh
Created By	Tatiparthi Amosh, 24/10/2024, 9:26 pm	Modified By	Tatiparthi Amosh, 24/10/2024, 9:26 pm

Object Relationships Edit Object Relationships Help ?

Managements (A)

with at least one related record from **Assignments (B)**

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 "Management 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”.

Management ID	Assignment ID	Name	Location	Skills	Phone	Email	Availability
M-1000 (1)	A-0006	Amosh	Hyderabad	Maintenance	7654321876	amosh@gmail.com	Available
M-1004 (1)	A-0007	Ajay	Nasik	Lane-Management	9876453219	ajay@gmail.com	Available
M-1005 (1)	A-0008	Nikku	Pune	Troubleshoot/Debugging	7654321875	nikku@gmail.com	Available
M-1012 (1)	A-0009	Riya	Warangal	Maintenance	8769546895	riya@gmail.com	Not Available
M-1017 (1)	A-0010	Anil	Nanded	Machine Installation	8889996662	anil@gmail.com	Available

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

Activity 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:

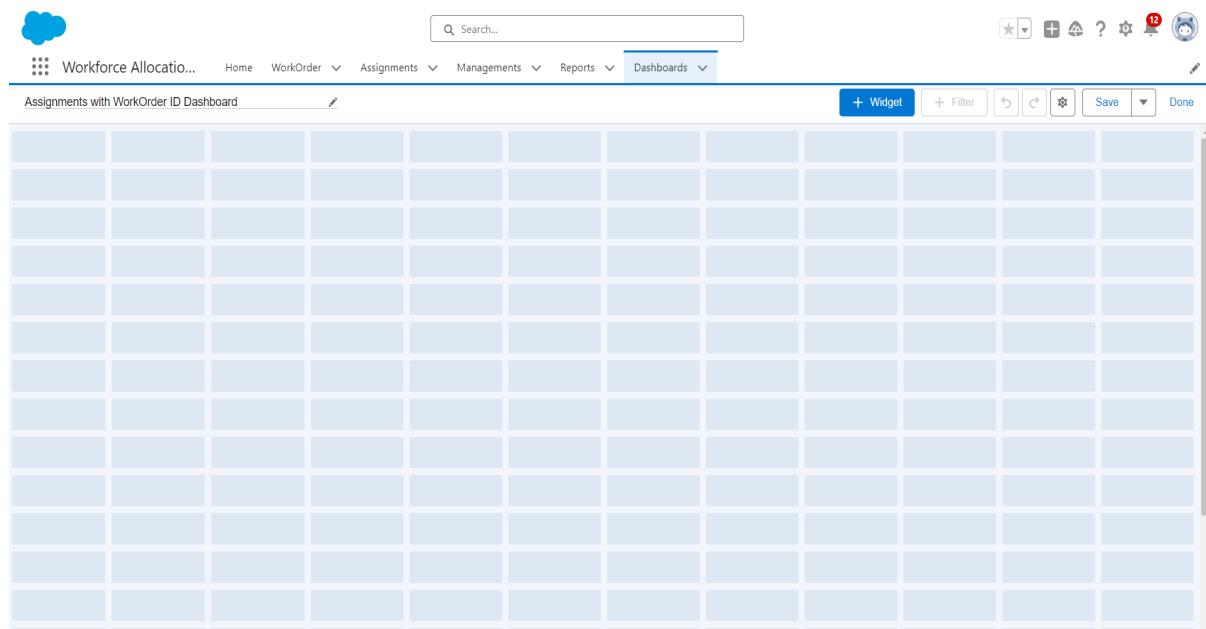
- Enter a name for your dashboard in the Name field. “Assignments with WorkOrder ID Dashboard”.

• 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.

Activity 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:**

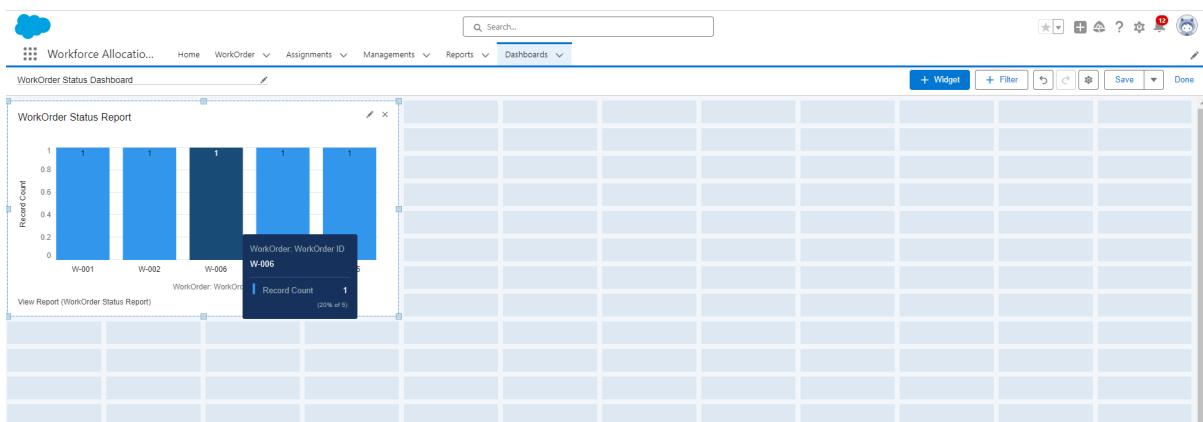
- Click Add to add the vertical bar graph component to your dashboard.

- **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 **Workforce allocation Management System** 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