

# Field Service WorkOrder Optimization

## 1. Project Overview

The Field Service Work Order Optimization System is designed to streamline operations for companies providing installation and repair services. By leveraging a robust database and intelligent algorithms, the system efficiently assigns work orders to skilled technicians based on factors such as location, availability, and expertise.

A prioritization algorithm ensures that tasks are assigned strategically, maximizing efficiency and reducing delays. Automated communication features keep technicians informed in real-time, fostering seamless coordination. Additionally, analytics tools provide actionable insights to drive continuous improvement.

This solution aims to enhance operational efficiency, reduce costs, and elevate customer satisfaction in the fast-paced field service industry.

## 2. Objectives

### Business Goals:

- **Enhance Operational Efficiency**
  - Streamline work order assignment to reduce delays and improve service delivery times.
  - Optimize technician utilization by matching tasks to the most suitable and available resources.
- **Reduce Operational Costs**
  - Minimize travel time and fuel consumption through location-based task allocation.
  - Lower administrative overhead with automated scheduling and communication tools.
- **Improve Customer Satisfaction**
  - Ensure timely and accurate service delivery to enhance the customer experience.
  - Provide customers with real-time updates on service schedules and technician arrivals.
- **Maximize Resource Utilization**
  - Leverage technicians' specialized skills by assigning appropriate tasks based on expertise.
  - Increase technician productivity with better planning and reduced idle time.
- **Drive Continuous Improvement**
  - Utilize analytics to identify performance bottlenecks and areas for improvement.

- Implement data-driven decisions to refine workflows and enhance service quality.
- **Support Scalability**
  - Enable the system to handle growing workloads and accommodate future business expansion.
  - Integrate seamlessly with other business systems to support evolving needs.

## Specific Outcomes:

- **Improved Task Assignment Efficiency**
  - Reduction in average time taken to assign work orders to technicians.
  - Increased percentage of tasks assigned to the most suitable technicians based on skills and proximity.
- **Reduced Travel Time and Costs**
  - Decrease in average technician travel time per work order.
  - Reduction in fuel costs and vehicle wear-and-tear due to optimized route planning.
- **Higher Customer Satisfaction**
  - Increased customer ratings and positive feedback due to timely service.
  - Reduction in customer complaints related to delayed or missed appointments.
- **Increased Technician Productivity**
  - Higher number of completed work orders per technician per day.
  - Reduced idle time for technicians, resulting in better resource utilization.
- **Enhanced Communication**
  - Faster technician response times through real-time notifications and updates.
  - Improved coordination between dispatch teams and field technicians.
- **Data-Driven Decision Making**
  - Availability of detailed analytics for monitoring key performance indicators (KPIs), such as average resolution time and first-time fix rates.
  - Insights into trends and bottlenecks to facilitate continuous improvement.
- **Cost Savings**
  - Measurable reduction in operational expenses, including administrative overhead and logistical costs.
  - Improved profitability through more efficient use of resources.
- **Scalable Operations**
  - Ability to handle increased volumes of work orders without compromising efficiency or quality.
  - Seamless integration with existing and future enterprise systems.

### 3. Salesforce Key Features and Concepts Utilized

- **Custom Objects and Relationships**
  - **Work Order, Technician, and Assignment** objects created to model the field service process.
  - Relationships:
    - Lookup relationships between Work Order → Technician and Assignment → Work Order/Technician for linking related records.
- **Data Modeling and Fields**
  - Custom fields added to capture essential information:
    - **Work Order:** Status, Priority, Service Type, and Description.
    - **Technician:** Availability, Skills, Location, and Contact Details.
    - **Assignment:** Assignment Date, Completion Date, Technician ID, and Work Order ID.
- **UI Customization**
  - **Tabs:** Custom tabs created for easy access to Work Order, Technician, and Assignment records.
  - **Lightning App:** Consolidated these tabs into a unified interface for streamlined navigation.
- **Automation with Apex**
  - **Apex Triggers:**
    - Automated status updates (e.g., updating Work Order status when Assignment is marked completed).
    - Ensured Technician availability is updated after an Assignment is completed.
  - **Apex Classes:**
    - Implemented business logic for assigning technicians based on location, availability, and skills.
    - Utility methods for efficient operations and system integration.
- **Reports and Dashboards**
  - Created comprehensive **Reports:**
    - Monitor open Work Orders by status and priority.
    - Technician performance metrics like task completion rate.
    - Assignments completed within specified time frames.
  - Built **Dashboards** for real-time visualization of KPIs, including workload distribution, service efficiency, and customer satisfaction metrics.
- **Standard Salesforce Features**
  - **Profiles and Roles:** Defined access levels to secure sensitive data and limit access based on user roles (e.g., Dispatcher, Technician).
  - **Record Ownership:** Used the Owner field to track accountability for Work Orders and Assignments.

- **Chatter:** Enabled team collaboration on Work Order records for updates and discussions.
- **Picklists and Validation**
  - Standardized input with picklists for fields like Status, Priority, Service Type, and Technician Availability.
  - Added validation rules to ensure data consistency (e.g., mandatory fields before completing a Work Order).
- **Analytics and Metrics**
  - **Custom Dashboards:**
    - Track ongoing Assignments, high-priority Work Orders, and technician workload.
    - KPIs like first-time resolution rate and average response time visualized effectively.
- **Mobile Accessibility**
  - Leveraged Salesforce's mobile-ready features for technicians to access Assignments and update statuses in the field.
- **Security and Compliance**
  - Implemented role-based access control to secure records based on profiles.
  - Used field-level security to protect sensitive data like customer contact details.

## 4. Detailed Steps to Solution Design

### Create Objects From Spreadsheet

- **Create Technician Object:** The **Technician** object stores critical details about field technicians, including their availability, skills, and location. It features key fields such as Technician ID, Name, Email, Phone, Availability (Picklist), Location (Picklist), and Skills (Picklist). Relationships are established with the Work Order and Assignment objects via lookup fields to link technicians to specific tasks and assignments.

SETUP > OBJECT MANAGER

Technician

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Details

Description

API Name

Technician\_\_c

Custom

✓

Singular Label

Technician

Plural Label

Technicians

Enable Reports

✓

Track Activities

✓

Track Field History

Deployment Status

Deployed

Help Settings

Standard salesforce.com Help Window

Edit

Delete

- **Create WorkOrder Object:** The **WorkOrder** object represents service tasks, such as installations or repairs. It includes key fields like Work Order ID (Auto Number), Date, Description, Status (Picklist), Priority (Picklist), Service Type (Picklist), Location, and Email. Relationships are established with Technician and Assignment objects to link tasks to the appropriate resources.

SETUP > OBJECT MANAGER  
**WorkOrder**

Details	Details	Edit	Delete
Fields & Relationships	Description		
Page Layouts	API Name	Enable Reports	
Lightning Record Pages	WorkOrder__c	✓	
Buttons, Links, and Actions	Custom	Track Activities	
Compact Layouts	✓	Track Field History	
Field Sets	Singular Label	Deployment Status	
Object Limits	WorkOrder	Deployed	
Record Types	Plural Label	Help Settings	
Related Lookup Filters	WorkOrders	Standard salesforce.com Help Window	
Search Layouts			

- **Create Assignment Object:** The **Assignment** object tracks the allocation of technicians to specific work orders. It includes fields like Assignment ID (Auto Number), Assignment Date, Completion Date, Technician ID (Lookup), and Work Order ID (Lookup). This object links technicians to their assigned tasks, ensuring proper tracking of service completion.

SETUP > OBJECT MANAGER  
**Assignment**

Details	Details	Edit	Delete
Fields & Relationships	Description		
Page Layouts	API Name	Enable Reports	
Lightning Record Pages	Assignment__c	✓	
Buttons, Links, and Actions	Custom	Track Activities	
Compact Layouts	✓	Track Field History	
Field Sets	Singular Label	Deployment Status	
Object Limits	Assignment	Deployed	
Record Types	Plural Label	Help Settings	
Related Lookup Filters	Assignments	Standard salesforce.com Help Window	

## Create A Custom Tab

- **Assignment Object:** The **Assignment** tab is created to provide easy access to Assignment records. It allows users to view, manage, and track technician assignments for work orders. The tab is added to a **Lightning App** for seamless integration into the user interface, ensuring efficient navigation and task management.

- **WorkOrder and Technician Object:** The **Work Order** and **Technician** tabs are **defaultly created** as these objects are imported from a spreadsheet.

## Create The Lightning App

A **Lightning App** is created to consolidate the **Work Order**, **Technician**, and **Assignment** tabs into a unified interface. This app provides users with easy access to all relevant records and ensures streamlined navigation across different objects. It enhances the user experience by organizing tasks and data in a single, efficient workspace.

The screenshot shows the Lightning Experience App Manager interface. At the top, there's a 'SETUP' button and a 'Lightning Experience App Manager' title. On the right, there are buttons for 'New Lightning App' and 'New Connected App'. Below this, a table lists 24 items, sorted by App Name. The table has columns for App Name, Developer Name, Description, Last Modified, App Type, and Visibility. The 'Field Service WorkOrder Optimizati...' app is highlighted.

Below the table, there's a search bar and a navigation menu. The 'Field Service WorkOrder Optimizati...' app is selected, showing a dashboard with various reports and charts.

The dashboard includes:

- WorkOrders Status Reports:** A table showing work order details (ID, Assignee, Status, etc.).
- WorkOrders Status Reports:** A bar chart showing the record count for different work order types (Troubleshooting, Lane Management, Machine Installation, Maintenance).
- Technician and Assignment Details Report:** A bar chart showing the record count for different technicians (elina@gil.com, rishi@user3.com, rishikumar@user2.com, shikhar@user5.com).
- Recent Records:** A list of recent records for various reports.
- Today's Tasks:** A section for tasks due today.

## Create Fields and Relationships

- **Technician:** Fields like Technician ID, Skills, Availability, and Location. Lookup to Work Order and Assignment.
- **Work Order:** Fields like Work Order ID, Status, Priority, and Service Type. Lookup to Technician and Assignment.
- **Assignment:** Fields like Assignment ID, Assignment Date, and Completion Date. Lookup to Technician and Work Order.

● Technician:

SETUP > OBJECT MANAGER

Technician

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules

Scoping Rules

Fields & Relationships

10 Items, Sorted by Field Label

Q Quick Find

New

Deleted Fields

Field Dependencies

Set History Tracking

Availability	Availability__c	Picklist		
Created By	CreatedById	Lookup(User)		
Email	Email__c	Email		
Last Modified By	LastModifiedById	Lookup(User)		
Location	Location__c	Picklist		
Name	Name__c	Text(255)		
Owner	OwnerId	Lookup(User,Group)	✓	
Phone	Phone__c	Phone		
Skills	Skills__c	Picklist		
Technician ID	Name	Text(80)	✓	

● WorkOrder:

SETUP > OBJECT MANAGER

WorkOrder

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules

Scoping Rules

Fields & Relationships

11 Items, Sorted by Field Label

Q Quick Find

New

Deleted Fields

Field Dependencies

Set History Tracking

Date	Date__c	Formula (Date)		
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	Auto Number	✓	

● Assignment

SETUP > OBJECT MANAGER

Assignment

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules

Fields & Relationships

8 Items, Sorted by Field Label

Q Quick Find

New

Deleted Fields

Field Dependencies

Set History Tracking

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

# Create Technician Profile

The **Technician Profile** defines permissions and access for technicians in the Salesforce Org. It grants visibility to relevant objects like **Work Order** and **Assignment**, allowing technicians to view, update, and complete their tasks. Access is restricted to only records assigned to them, ensuring data security and streamlined operations.

SETUP

Profiles

Profile

Technician

Help for this Page

Users with this profile have the permissions and page layouts listed below. Administrators can change a user's profile by editing that user's personal information.

If your organization uses Record Types, use the Edit links in the Record Type Settings section below to make one or more record types available to users with this profile.

Login IP Ranges | Enabled Apex Class Access | Enabled Visualforce Page Access | Enabled External Data Source Access | Enabled Named Credential Access | Enabled External Credential Principal Access | Enabled Custom Metadata Type Access | Enabled Custom Setting Definitions Access | Enabled Flow Access | Enabled Service Presence Status Access | Enabled Custom Permissions

Profile Detail

EditCloneDeleteView Users

Name	Technician		
User License	Salesforce Platform	Custom Profile	✓
Description			
Created By	Chaitanya Nanepalli	Modified By	Chaitanya Nanepalli
	04/01/2025, 5:51 am		04/01/2025, 5:54 am

Page Layouts

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

# Create Users

SETUP

Users

All Users

Help for this Page

On this page you can create, view, and manage users.

To get more licenses, use the Your Account app. [Let's Go](#)

View: All Users | Edit | Create New View

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other | All

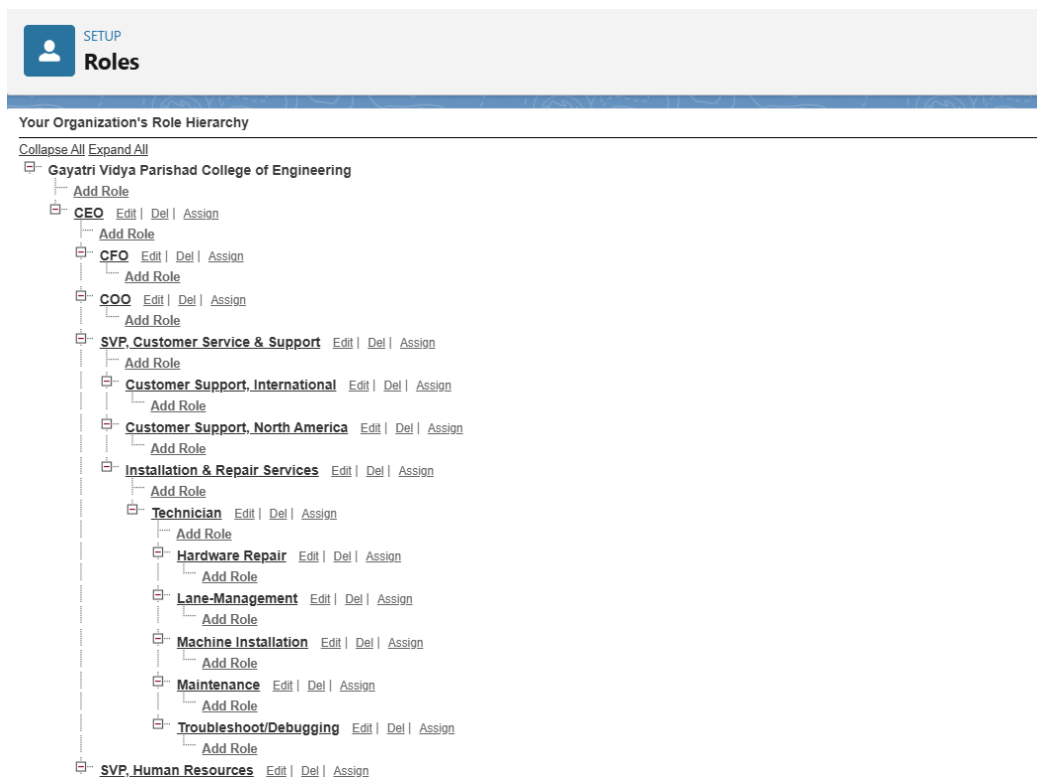
New UserReset Password(s)Add Multiple Users

Action	Full Name	Alias	Username	Role	Active	Profile
<input type="checkbox"/>   Edit	Chatter Expert	Chatter	chatty00dns000009ghiw2am.pmm5zost6ic@chatter.salesforce.com		✓	Chatter Free User
<input type="checkbox"/>   Edit	dhawan_sikhar	sdhaw	nanepalliachaitanya@gmail.com		<input type="checkbox"/>	Force.com - App Subscription User
<input type="checkbox"/>   Edit	Gilbert_Elina	eglib	elina@gill.com	Maintenance	✓	Technician
<input type="checkbox"/>   Edit	kohli_virat	vkohli	rohit@user3.com	Lane-Management	✓	Technician
<input type="checkbox"/>   Edit	Nanepalli_Chaitanya	CNane	chaitanya@gvpce.com		✓	System Administrator
<input type="checkbox"/>   Edit	Sharma_Rohit	rshar	rohitsharma@user2.com	Troubleshoot/Debugging	✓	Technician
<input type="checkbox"/>   Edit	smith_Steeve	ssmit	chaitu118c@user.com		✓	Force.com - Free User
<input type="checkbox"/>   Edit	User_Integration	integ	integration@00dns000009ghiw2am.com		✓	Analytics Cloud Integration User
<input type="checkbox"/>   Edit	User_Security	sec	insightssecurity@00dns000009ghiw2am.com		✓	Analytics Cloud Security User

New UserReset Password(s)Add Multiple Users



**Create Technician Role under Services and Maintenance and create roles under Technician and Assign it to Users:**



## Create Apex Classes & Triggers

- **WorkOrder Class:**

```

1  public class WorkOrderClass {
2      public static void workOrder(List<WorkOrder_> newListWorkOrder) {
3          // Map to store technician matching data
4          Map<Id, List<String>> workOrderCriteriaMap = new Map<Id, List<String>>();
5          List<WorkOrder_> validWorkOrders = new List<WorkOrder_>();
6          List<Assignment_> lstAssignment = new List<Assignment_>();
7
8          // Collect valid work orders and their matching criteria
9          for (WorkOrder_> workOrder : newListWorkOrder) {
10             if (workOrder.Service_Type__c != null && workOrder.Location__c != null) {
11                 List<String> criteria = new List<String>();
12                 criteria.add(workOrder.Service_Type__c);
13                 criteria.add(workOrder.Location__c);
14
15                 workOrderCriteriaMap.put(workOrder.Id, criteria);
16                 validWorkOrders.add(workOrder);
17             }
18         }
19
20         // Query all technicians
21         Map<Id, Technician_> allTechnicians = new Map<Id, Technician_>{
22             [SELECT Id, Name, Phone__c, Location__c, Skills__c, Availability__c, Email__c
23              FROM Technician__c WHERE Availability__c = 'Available']
24         };
25
26         // Map work orders to technician Ids
27         Map<Id, Id> workOrderToTechMap = new Map<Id, Id>();
28         for (Id workOrderId : workOrderCriteriaMap.keySet()) {
29             List<String> criteria = workOrderCriteriaMap.get(workOrderId);
30
31             for (Technician__c technician : allTechnicians.values()) {
32                 if (technician.Skills__c.contains(criteria[0]) &&
33                     technician.Location__c == criteria[1]) {
34                     workOrderToTechMap.put(workOrderId, technician.Id);
35                     allTechnicians.remove(technician.Id); // Remove to avoid assigning the same tech multiple times

```

```

36         break;
37     }
38 }
39 }
40
41 // Create assignments
42 for (WorkOrder__c workOrder : validWorkOrders) {
43     Id techId = workOrderToTechMap.get(workOrder.Id);
44     if (techId != null) {
45         Assignment__c assignment = new Assignment__c();
46         assignment.WorkOrder_ID__c = workOrder.Id; // Reference to WorkOrder__c
47         assignment.Technician_ID__c = techId; // Reference to Technician__c
48         lstAssignment.add(assignment);
49     }
50 }
51
52 // Insert assignments if any are valid
53 if (!lstAssignment.isEmpty()) {
54     insert lstAssignment;
55 } else {
56     System.debug('No valid assignments to insert.');

```

## ● AssigningEmail 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> technicians = new Map<Id, Technician__c>([SELECT Id, Phone__c, Location__c, Skills__c, Name__c, Email__c, Availability__c, Name FROM Technician__c]);
5
6         try {
7             for(Assignment__c con : assRec){
8                 if(con.Technician_ID__c != null){
9                     messaging.SingleEmailMessage mail = new messaging.SingleEmailMessage();
10                    List<String> sendTo = new List<String>();
11                    sendTo.add(technicians.get(con.Technician_ID__c).Email__c);
12                    mail.setToAddresses(sendTo);
13                    String subject = 'WorkOrder Assignment';
14                    mail.setSubject(subject);
15                    String body = 'The following WorkOrder has been assigned to you.';
16                    mail.setHTMLbody(body);
17                    myVar.add(mail);
18                }
19            }
20
21            // Send the emails if there are any to send
22            if (!myVar.isEmpty()) {
23                Messaging.sendEmail(myVar);
24            }
25        } catch(Exception e) {
26            system.debug('Error ----- ' + e.getMessage());
27        }
28    }
29 }

```

## ● CompletionMail Class:

```

1 public class CompletionMail {
2     @future
3     public static void sendEmailInFuture(List<String> emailAddresses, List<String> emailSubjects, List<String> emailBodies) {
4         List<messaging.SingleEmailMessage> emails = new List<messaging.SingleEmailMessage>();
5
6         for (Integer i = 0; i < emailAddresses.size(); i++) {
7             messaging.SingleEmailMessage mail = new messaging.SingleEmailMessage();
8             mail.setToAddresses(new List<String>{emailAddresses[i]});
9             mail.setSubject(emailSubjects[i]);
10            mail.setHTMLbody(emailBodies[i]);
11            emails.add(mail);
12        }
13
14        Messaging.sendEmail(emails);
15    }
16
17    public static void sendEmailMsg(List<WorkOrder__c> newWorkOrders, List<WorkOrder__c> oldWorkOrders) {
18        List<String> emailAddresses = new List<String>();
19        List<String> emailSubjects = new List<String>();
20        List<String> emailBodies = new List<String>();
21
22        for (Integer i = 0; i < newWorkOrders.size(); i++) {
23            WorkOrder__c newWorkOrder = newWorkOrders[i];
24            WorkOrder__c oldWorkOrder = oldWorkOrders[i];
25
26            if (newWorkOrder.Status__c == 'Resolved' && oldWorkOrder.Status__c != 'Resolved' && newWorkOrder.Email__c != null) {
27                emailAddresses.add(newWorkOrder.Email__c);
28                emailSubjects.add('Status Updated');
29                emailBodies.add('Dear Customer, <br><br>Your work order (ID: ' + newWorkOrder.Id + ') has been marked as resolved. Thank you for your patience.');

```

## TechnicianAvailabilityUpdater Class:

```
1 public class TechnicianAvailabilityUpdater {
2     public static void updateTechnicianAvailability(List<WorkOrder_c> newWorkOrders) {
3         // List to hold Technicians to update their availability
4         List<Technician_c> techniciansToUpdate = new List<Technician_c>();
5         List<Assignment_c> assignmentsToCreate = new List<Assignment_c>();
6
7         // Iterate through the updated WorkOrder records
8         for (WorkOrder_c newWorkOrder : newWorkOrders) {
9             // Check if the work order status is 'Resolved'
10            if (newWorkOrder.Status_c == 'Resolved') {
11                System.debug('Processing WorkOrder ID: ' + newWorkOrder.Id); // Debug log for processing
12
13                // Query for the related Assignment records to get the Technician associated with the WorkOrder
14                List<Assignment_c> assignments = [SELECT Technician_ID__c FROM Assignment__c WHERE WorkOrder_ID__c = :newWorkOrder.Id];
15
16                // Iterate over each assignment to get the technician and update availability
17                for (Assignment_c assignment : assignments) {
18                    if (assignment.Technician_ID__c != null) {
19                        Technician_c technician = [SELECT Id, Availability__c, Skills__c, Location__c FROM Technician__c WHERE Id = :assignment.Technician_ID__c LIMIT 1];
20
21                        // Update the Technician's Availability to 'Available' only if it's not already 'Available'
22                        if (technician != null && technician.Availability__c != 'Available') {
23                            technician.Availability__c = 'Available';
24                            techniciansToUpdate.add(technician);
25                            System.debug('Technician ' + technician.Id + ' marked as Available'); // Debug for technician availability update
26
27                            // Now that technician is available, check if they can be assigned to any new work orders
28                            // Query for any "New" status WorkOrders that need a technician
29                            List<WorkOrder_c> newStatusWorkOrders = [
30                                SELECT Id, Service_Type__c, Location__c
31                                FROM WorkOrder__c
32                                WHERE Status__c = 'New'
33                                AND Service_Type__c = :technician.Skills__c
34                                AND Location__c = :technician.Location__c
35                            ];
36
37                            // Create assignments for matching work orders
38                            for (WorkOrder_c workOrder : newStatusWorkOrders) {
39                                // Avoid assigning the same technician to the same work order twice
40                                if (!isTechnicianAssignedToWorkOrder(workOrder.Id, technician.Id)) {
41                                    Assignment_c newAssignment = new Assignment_c();
42                                    newAssignment.Technician_ID__c = technician.Id;
43                                    newAssignment.WorkOrder_ID__c = workOrder.Id;
44                                    assignmentsToCreate.add(newAssignment);
45                                    System.debug('Assignment created for Technician ' + technician.Id + ' and WorkOrder ' + workOrder.Id);
46                                }
47                            }
48                        }
49                    }
50                }
51            }
52        }
53
54        // Update Technicians' availability to 'Available' if necessary
55        if (!techniciansToUpdate.isEmpty()) {
56            update techniciansToUpdate;
57            System.debug('Technicians availability updated: ' + techniciansToUpdate.size()); // Debug for updates
58        }
59
60        // Create Assignments if any are valid
61        if (!assignmentsToCreate.isEmpty()) {
62            insert assignmentsToCreate;
63            System.debug('Assignments created: ' + assignmentsToCreate.size()); // Debug for assignments created
64        }
65    }
66
67    // Helper method to check if technician is already assigned to the work order
68    private static Boolean isTechnicianAssignedToWorkOrder(Id workOrderId, Id technicianId) {
69        List<Assignment_c> existingAssignments = [
70            SELECT Id
71            FROM Assignment__c
72            WHERE WorkOrder_ID__c = :workOrderId AND Technician_ID__c = :technicianId
73        ];
74        return !existingAssignments.isEmpty();
75    }
76 }
```

## ● CreateOrUpdateTechnicianQueueable Class:

```
1 public class CreateOrUpdateTechnicianQueueable implements Queueable {
2     private Set<Id> userIds;
3
4     // Constructor to pass user IDs
5     public CreateOrUpdateTechnicianQueueable(Set<Id> userIds) {
6         this.userIds = userIds;
7     }
8
9     public void execute(QueueableContext context) {
10        if (userIds == null || userIds.isEmpty()) {
11            System.debug('No User IDs provided for creation or update. Exiting Queueable execution.');
```

```

26 // Query existing Technician__c records based on Technician ID (UserName)
27 Map<String, Technician__c> existingTechniciansMap = new Map<String, Technician__c>{
28 [SELECT Id, Name, Phone__c, Email__c, Location__c, Availability__c, Skills__c
29 FROM Technician__c
30 WHERE Name IN :technicianIdsSet] // Match using Technician ID (UserName)
31 };
32
33 // List to hold Technician records for insertion or update
34 List<Technician__c> techniciansToUpsert = new List<Technician__c>();
35
36 for (User user : users) {
37     Technician__c technician = existingTechniciansMap.get(user.UserName); // Match using Technician ID
38
39     if (technician == null) {
40         // If no Technician exists, create a new one
41         technician = new Technician__c();
42         technician.Name = user.UserName; // Link Technician with UserName (unique identifier)
43     }
44
45     // Populate or update Technician fields
46     technician.Name__c = user.Name; // Technician Name
47     technician.Phone__c = user.Phone; // Technician Phone
48     technician.Email__c = user.Email; // Technician Email
49     technician.Location__c = user.City != null ? user.City : 'Default City'; // Use city from User or default
50     technician.Availability__c = 'Available'; // Default picklist value
51     technician.Skills__c = user.UserRole != null ? user.UserRole.Name : null; // Map the user role to skills
52
53     techniciansToUpsert.add(technician);
54 }
55
56 // Perform the DML operation to update existing records or insert new ones
57 try {
58     if (techniciansToUpsert.isEmpty()) {
59         // Use upsert, where Name (Technician ID) is treated as the external ID field for upsert
60         upsert techniciansToUpsert; // This will update existing records or create new ones if not found
61         System.debug('Technician records successfully inserted/updated.');
```

## ● AssingmentHandler Class:

```

1 public class AssignmentHandler {
2     public static void updateWorkOrderAndTechnician(List<Assignment__c> assRec){
3         // List to hold WorkOrders that need to be updated
4         List<WorkOrder__c> workOrdersToUpdate = new List<WorkOrder__c>();
5
6         // List to hold Technicians whose availability needs to be updated
7         List<Technician__c> techniciansToUpdate = new List<Technician__c>();
8
9         try {
10             for(Assignment__c con : assRec){
11                 if(con.Technician_ID__c != null){
12                     // Retrieve the related WorkOrder and update the status
13                     WorkOrder__c workOrder = [SELECT Id, Status__c FROM WorkOrder__c WHERE Id = :con.WorkOrder_ID__c LIMIT 1];
14
15                     if (workOrder != null && workOrder.Status__c != 'Resolved') {
16                         // Update the WorkOrder status to "Assigned/In Progress"
17                         workOrder.Status__c = 'Assigned/In Progress';
18                         workOrdersToUpdate.add(workOrder);
19                     }
20
21                     // Retrieve the technician and update availability to "Not Available"
22                     Technician__c technician = [SELECT Id, Availability__c FROM Technician__c WHERE Id = :con.Technician_ID__c LIMIT 1];
23                     if (technician != null && technician.Availability__c != 'Not Available') {
24                         technician.Availability__c = 'Not Available';
25                         techniciansToUpdate.add(technician);
26                     }
27                 }
28             }
29
30             // Update the WorkOrders if there are any changes
31             if (!workOrdersToUpdate.isEmpty()) {
32                 update workOrdersToUpdate;
33             }
34
35             // Update the Technicians if there are any changes
36             if (!techniciansToUpdate.isEmpty()) {
37                 update techniciansToUpdate;
38             }
39         } catch(Exception e) {
40             system.debug('Error ----- ' + e.getMessage());
41         }
42     }
43 }
```

## ● Assignment Trigger

```

1 trigger AssignmentTrigger on Assignment__c (after insert) {
2     if (Trigger.IsAfter && Trigger.IsInsert) {
3         // Call the method to send email
4         AssigningEmail.sendEmailmsg(Trigger.New);
5
6         // Call the method to update WorkOrder status and Technician availability
7         AssignmentHandler.updateWorkOrderAndTechnician(Trigger.New);
8     }
9 }
```

## • WorkOrder Trigger:

```

1 trigger WorkOrderTrigger on WorkOrder__c (after insert, after update) {
2
3     if (Trigger.isAfter && Trigger.isInsert) {
4         // Handling after insert
5         WorkOrderClass.workOrder(Trigger.new);
6     }
7
8     if (Trigger.isAfter && Trigger.isUpdate) {
9         // Prepare email details for the future method
10        List<String> emailAddresses = new List<String>();
11        List<String> emailSubjects = new List<String>();
12        List<String> emailBodies = new List<String>();
13
14        for (Integer i = 0; i < Trigger.new.size(); i++) {
15            WorkOrder__c newWorkOrder = Trigger.new[i];
16            WorkOrder__c oldWorkOrder = Trigger.old[i];
17
18            if (newWorkOrder.Status__c == 'Resolved' && oldWorkOrder.Status__c != 'Resolved' && newWorkOrder.Email__c != null) {
19                emailAddresses.add(newWorkOrder.Email__c);
20                emailSubjects.add('Status Updated');
21                emailBodies.add(
22                    'Dear Customer, <br><br>Your work order (ID: ' +
23                    newWorkOrder.Id +
24                    ') has been marked as resolved. Thank you for your patience.'
25                );
26            }
27        }
28
29        // Call the future method to send emails
30        if (!emailAddresses.isEmpty()) {
31            CompletionMail.sendEmailInFuture(emailAddresses, emailSubjects, emailBodies);
32        }
33
34        // Handling Technician availability update
35        TechnicianAvailabilityUpdater.updateTechnicianAvailability(Trigger.new);
36
37        // Create assignments for available technicians based on matching criteria
38        //WorkOrderClass.workOrder(Trigger.new);
39    }
40 }
41

```

## • CreateTechnicianOnUserCreation Trigger

```

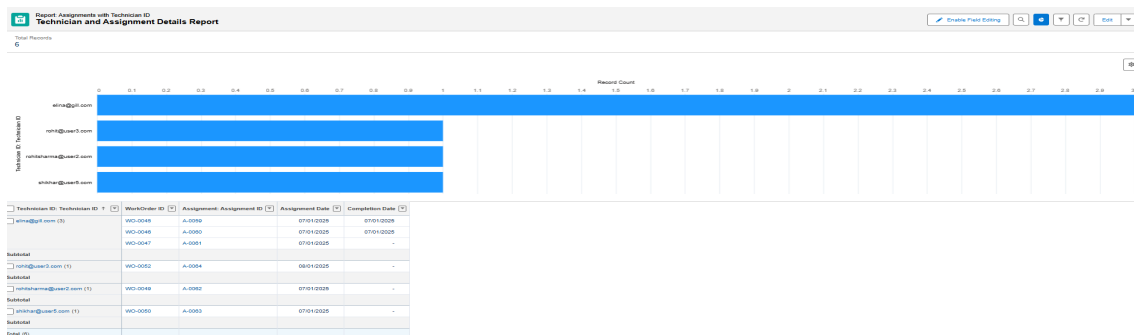
1 trigger CreateTechnicianOnUserCreation on User (after insert, after update) {
2     // Collect user IDs that need to create or update Technician records
3     Set<Id> technicianUserIds = new Set<Id>();
4
5     // Fetch the Profile ID for "Technician" to avoid multiple queries
6     Id technicianProfileId = [SELECT Id FROM Profile WHERE Name = 'Technician' LIMIT 1].Id;
7
8     // Check if the user has the Technician profile
9     for (User user : Trigger.new) {
10        if (user.ProfileId == technicianProfileId) {
11            technicianUserIds.add(user.Id);
12        }
13    }
14
15    // If there are users to process, call the queueable class
16    if (!technicianUserIds.isEmpty()) {
17        System.enqueueJob(new CreateOrUpdateTechnicianQueueable(technicianUserIds));
18    }
19 }

```

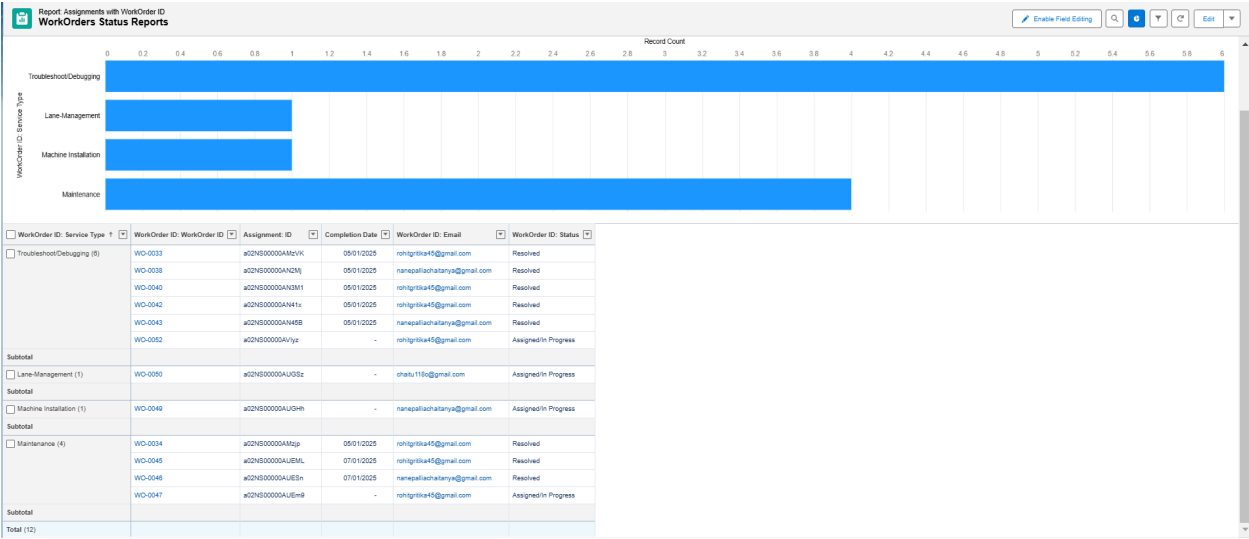
## Create Reports And Dashboards:

### Reports:

- **Technician and Assignment Details Report:** Monitors technician assignments, including dates and completion status.

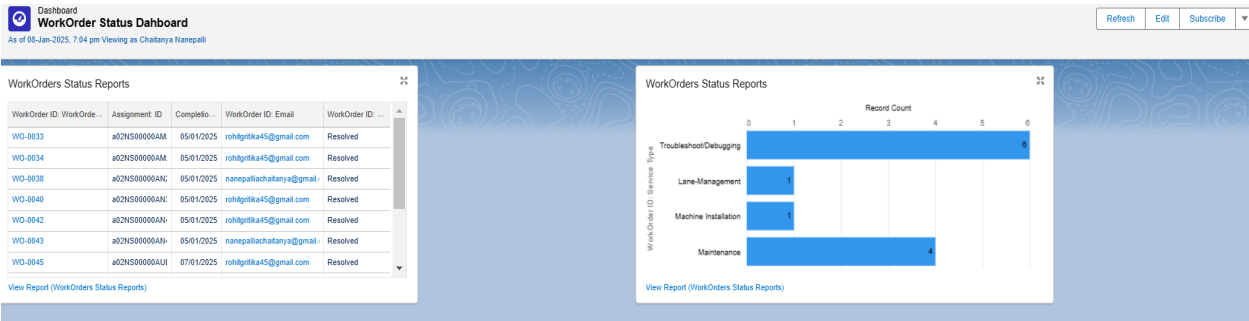


Work Order Report: Tracks work orders by status, priority, and service type.

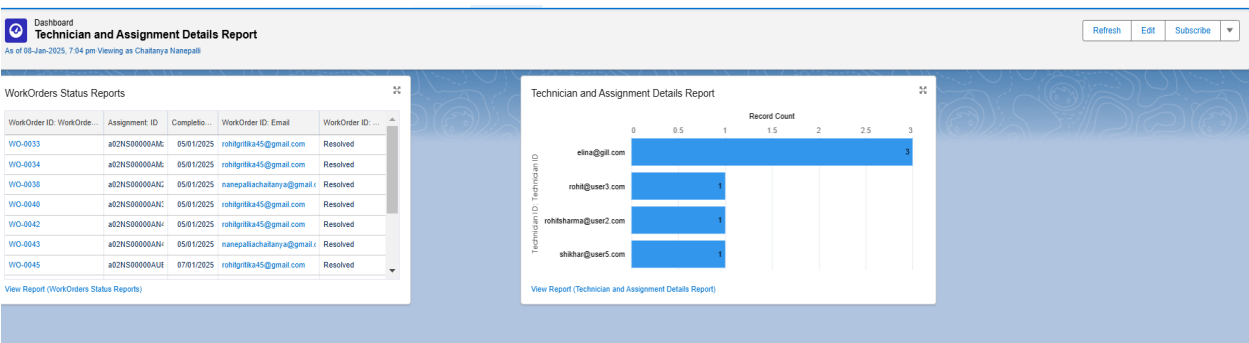


Dashboards:

- Work Order Status Dashboard:



- Technician and Assignment Details Dashboard:



## Testing and Validation:

- 1. Create users under Technician profile such that a record is created in Technician Object with the Users details:(or)  
Create users under Technician Profile directly:**


Edit [steeve@user6.com](#)

\* = Required Information

\* Technician ID

steeve@user6.com

Owner

 Chaitanya Nanepalli

Name

steeve smith

Phone

9848022338

Email

rohitgritika45@gmail.com

Location

Nasik


Availability

Available


Skills

Hardware Repair

Created By

 Chaitanya Nanepalli, 07/01/2025, 11:17 pm

Last Modified By

 Chaitanya Nanepalli, 07/01/2025, 11:17 pm

Cancel

Save & New

Save

Technicians

All Records

New

Change Owner

Import

Printable View

Assign Label

5 items • Sorted by Technician ID • Filtered by All technicians • Updated a few seconds ago

Q Search this list...

<div><input type="checkbox"/></div> Technician ID <div>↑</div>	<div><input type="checkbox"/></div> Name	<div><input type="checkbox"/></div> Phone	<div><input type="checkbox"/></div> Email	<div><input type="checkbox"/></div> Location	<div><input type="checkbox"/></div> Availability	<div><input type="checkbox"/></div> Skills	<div><input type="checkbox"/></div>	
1 <div><input type="checkbox"/></div>	<a href="#">elina@gill.com</a>	Elina Gilbert	123456780	<a href="#">chaitu118c@gmail.com</a>	Hyderabad	Available	Maintenance	<div><div></div></div>
2 <div><input type="checkbox"/></div>	<a href="#">rahul@user4.com</a>	KL Rahul	9848022338	<a href="#">nanepalliachaitanya@gmail.com</a>	Nasik	Available	Hardware Repair	<div><div></div></div>
3 <div><input type="checkbox"/></div>	<a href="#">rohit@user3.com</a>	Virat kohli	9848022336	<a href="#">rohitgritika45@gmail.com</a>	Pune	Available	Troubleshoot/Debugging	<div><div></div></div>
4 <div><input type="checkbox"/></div>	<a href="#">rohitsharma@user2.com</a>	Rohit Sharma	9848022334	<a href="#">nanepalliachaitanya@gmail.com</a>	Nanded	Available	Machine Installation	<div><div></div></div>
5 <div><input type="checkbox"/></div>	<a href="#">shikhar@user5.com</a>	shikhar dhawan	9848022337	<a href="#">chaitu118c@gmail.com</a>	Warangal	Available	Lane-Management	<div><div></div></div>

## 2. Create WorkOrder Record which is the work that Customer assigns us:

WorkOrders

Recently Viewed

WorkOrder "WO-0047" was created.

New

Change Owner

Import

Assign Label

0 items • Updated a few seconds ago

Search this list...

<input type="checkbox"/>	WorkOrder ID	Email	Service Type	Description	Location	Priority	Status	
1	<input type="checkbox"/> WO-0047	<a href="mailto:rohitratika45@gmail.com">rohitratika45@gmail.com</a>	Maintenance		Hyderabad		New	
2	<input type="checkbox"/> WO-0046	<a href="mailto:nanepalliachaitanya@gmail.com">nanepalliachaitanya@gmail.com</a>	Maintenance		Hyderabad	High	Assigned/In Progress	
3	<input type="checkbox"/> WO-0045	<a href="mailto:rohitratika45@gmail.com">rohitratika45@gmail.com</a>	Maintenance		Hyderabad	High	Resolved	
4	<input type="checkbox"/> WO-0043	<a href="mailto:nanepalliachaitanya@gmail.com">nanepalliachaitanya@gmail.com</a>	Troubleshoot/Debugging		Nasik	High	Resolved	

Case 1:If the workorder is assigned and there is a available technician then it assigns to then it assigns to the technician

Related

Details

Assignment ID

A-0059

WorkOrder ID

[WO-0045](#)

Technician ID


[elina@gill.com](#)

Assignment Date


07/01/2025

Completion Date


Owner

 [Chaitanya Nanepalli](#)

Created By

 [Chaitanya Nanepalli](#), 07/01/2025, 10:44 pm

Last Modified By

 [Chaitanya Nanepalli](#), 07/01/2025, 10:44 pm

<input type="checkbox"/> WorkOrder ID	<input type="checkbox"/> Email	<input type="checkbox"/> Service Type	<input type="checkbox"/> Description	<input type="checkbox"/> Location	<input type="checkbox"/> Priority	<input type="checkbox"/> Status
1 <input type="checkbox"/> WO-0045	<a href="#">rohitgritika45@gmail.com</a>	Maintenance		Hyderabad	High	Assigned/In Progress

<input type="checkbox"/> Technician ID	<input type="checkbox"/> Name	<input type="checkbox"/> Phone	<input type="checkbox"/> Email	<input type="checkbox"/> Location	<input type="checkbox"/> Availability	<input type="checkbox"/> Skills
<input type="checkbox"/> <a href="#">elina@gill.com</a>	Elina Gilbert	123456780	<a href="#">chaitu118c@gmail.com</a>	Hyderabad	Not Available	Maintenance
<input type="checkbox"/> <a href="#">rahul@user4.com</a>	KL Rahul	9848022338	<a href="#">nanepalliachaitanya@gmail.com</a>	Nasik	Available	Hardware Repair
<input type="checkbox"/> <a href="#">rohit@user3.com</a>	virat kohli	9848022336	<a href="#">rohitgritika45@gmail.com</a>	Pune	Available	Troubleshoot/Debugging

WorkOrder Assignment

Inbox



Chaitanya Nanepalli

via lbbw8g8gnpi329jw.7zaa.ns-9qhiw2am.ind56.bnc.salesforce.com

to me

10:44 PM (3 minutes ago)

The following WorkOrder has been assigned to you.

Reply

Forward

When a work order requirement matches with the technician Elina, it is assigned to her, an assignment record is created, and an email notification is sent to inform her about the assigned work. Upon assignment, the work order status is updated to **Assigned/In Progress**, and the technician's availability is updated to **Not Available**.

Case 2:If a work order is created and no matching technician is currently available



WorkOrders

Recently Viewed

WorkOrder "WO-0047" was created.

New

Change Owner

Import

Assign Label

Search this list...

1 items • Updated a few seconds ago

	WorkOrder ID	Email	Service Type	Description	Location	Priority	Status
1	WO-0047	rohitgritika45@gmail.com	Maintenance		Hyderabad		New
2	WO-0046	nanepalliachaitanya@gmail.com	Maintenance		Hyderabad	High	Assigned/In Progress
3	WO-0045	rohitgritika45@gmail.com	Maintenance		Hyderabad	High	Resolved
4	WO-0043	nanepalliachaitanya@gmail.com	Troubleshoot/Debugging		Nasik	High	Resolved

WorkOrders

Recently Viewed

New

Change Owner

Import

Assign Label

Search this list...

9 items • Updated a few seconds ago

	WorkOrder ID	Email	Service Type	Description	Location	Priority	Status
1	WO-0047	rohitgritika45@gmail.com	Maintenance		Hyderabad		New
2	WO-0046	nanepalliachaitanya@gmail.com	Maintenance		Hyderabad	High	Resolved
3	WO-0045	rohitgritika45@gmail.com	Maintenance		Hyderabad	High	Resolved
4	WO-0043	nanepalliachaitanya@gmail.com	Troubleshoot/Debugging		Nasik	High	Resolved
5	WO-0042	rohitgritika45@gmail.com	Troubleshoot/Debugging		Nasik	High	Resolved
6	WO-0040	rohitgritika45@gmail.com	Troubleshoot/Debugging		Nasik	High	Resolved
7	WO-0038	nanepalliachaitanya@gmail.com	Troubleshoot/Debugging		Nasik	High	Resolved
8	WO-0034	rohitgritika45@gmail.com	Maintenance		Nanded	High	Resolved
9	WO-0033	rohitgritika45@gmail.com	Troubleshoot/Debugging		Nasik	High	Resolved

Cancel

Save

WorkOrders

Recently Viewed

New

Change Owner

Import

Assign Label

Search this list...

9 items • Updated a few seconds ago

	WorkOrder ID	Email	Service Type	Description	Location	Priority	Status
1	WO-0046	nanepalliachaitanya@gmail.com	Maintenance		Hyderabad	High	Resolved
2	WO-0047	rohitgritika45@gmail.com	Maintenance		Hyderabad		Assigned/In Progress
3	WO-0045	rohitgritika45@gmail.com	Maintenance		Hyderabad	High	Resolved
4	WO-0043	nanepalliachaitanya@gmail.com	Troubleshoot/Debugging		Nasik	High	Resolved
5	WO-0042	rohitgritika45@gmail.com	Troubleshoot/Debugging		Nasik	High	Resolved
6	WO-0040	rohitgritika45@gmail.com	Troubleshoot/Debugging		Nasik	High	Resolved
7	WO-0038	nanepalliachaitanya@gmail.com	Troubleshoot/Debugging		Nasik	High	Resolved
8	WO-0034	rohitgritika45@gmail.com	Maintenance		Nanded	High	Resolved
9	WO-0033	rohitgritika45@gmail.com	Troubleshoot/Debugging		Nasik	High	Resolved

If a work order is created and no matching technician is currently available, the work order remains in a pending state. Once a suitable technician becomes available, the system automatically assigns the work order to them, creates an assignment record, sends an email notification to the technician, updates the work order status to **Assigned/In Progress**, and sets the technician's availability to **Not Available**.

### 3.If the work order is completed:

Status Updated Inbox x

Chaitanya Nanepalli 322103310162@gvpce.ac.in via j5wm0x06vrd0qkro.hz7o4uw.ns-9qhiw2am.ind56.bnc.salesforce.com to me

10:45 PM (0 minutes ago)

Dear Customer,

Your work order (ID: a01NS00000t7E8EYAU) has been marked as resolved. Thank you for your patience.

Reply

Forward

<input type="checkbox"/>	WorkOrder ID	Email	Service Type	Description	Location	Priority	Status	
1	<input type="checkbox"/> WO-0045	rohitgritika45@gmail.com	Maintenance		Hyderabad	High	Resolved	
2	<input type="checkbox"/> WO-0043	nanepalliachaitanya@gmail.com	Troubleshoot/Debugging		Nasik	High	Resolved	
3	<input type="checkbox"/> WO-0042	rohitgritika45@gmail.com	Troubleshoot/Debugging		Nasik	High	Resolved	
4	<input type="checkbox"/> WO-0040	rohitgritika45@gmail.com	Troubleshoot/Debugging		Nasik	High	Resolved	
5	<input type="checkbox"/> WO-0038	nanepalliachaitanya@gmail.com	Troubleshoot/Debugging		Nasik	High	Resolved	
6	<input type="checkbox"/> WO-0034	rohitgritika45@gmail.com	Maintenance		Nanded	High	Resolved	
7	<input type="checkbox"/> WO-0033	rohitgritika45@gmail.com	Troubleshoot/Debugging		Nasik	High	Resolved	

Cancel

Save

Technicians

All Records

New

Change Owner

Import

Printable View

Assign Label

5 Items • Sorted by Technician ID • Filtered by All technicians • Updated a few seconds ago

Search this list...

<input type="checkbox"/>	Technician ID	Name	Phone	Email	Location	Availability	Skills	
1	<input type="checkbox"/> elina@gill.com	Elina Gilbert	123456780	chaitu118c@gmail.com	Hyderabad	Available	Maintenance	
2	<input type="checkbox"/> rahul@user4.com	KL Rahul	9848022338	nanepalliachaitanya@gmail.com	Nasik	Available	Hardware Repair	
3	<input type="checkbox"/> rohit@user3.com	virat kohli	9848022336	rohitgritika45@gmail.com	Pune	Available	Troubleshoot/Debugging	
4	<input type="checkbox"/> rohitsharma@user2.com	Rohit Sharma	9848022334	nanepalliachaitanya@gmail.com	Nanded	Available	Machine Installation	
5	<input type="checkbox"/> shikhar@user5.com	shikhar dhawan	9848022337	chaitu118c@gmail.com	Warangal	Available	Lane-Management	

If the work is completed, the technician updates the work order status to **Resolved**. Upon this update, the system automatically changes the technician's availability status to **Available**.