



SB8067 Salesforce Developer



S.Veerasamy Chettiar College of Engineering And
Technology,Puliyangudi.

Tittle : LEASE MANAGEMENT

Submitted by,
R. Mathialagan,
952622104031



**Smart
Internz**

Dear Student,

Greetings from SmartBridge,

Your team has successfully enrolled for the project. Please find the team details below

Team ID : NM2025TMID04732

Team Size : 4

Team Leader : Mathialagan R

Team member : Mohamed Sithick M

Team member : Mohamed hameem S

Team member : Sanjai S

Regards,
Team SmartBridge

LEASE MANAGEMENT

1. Ideation Phase

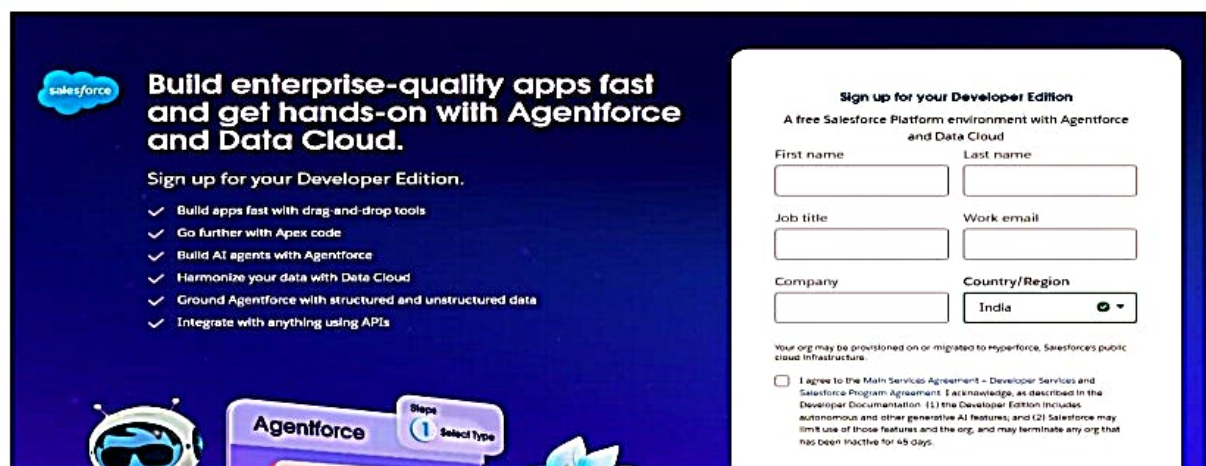
Management System is designed to simplify and automate the daily operations of a garage using Salesforce CRM.

Traditional garages depend on manual record-keeping for customer details, appointments, and billing, which leads to inefficiency and errors. The idea behind this project is to use Salesforce's cloud-based capabilities to:

- Schedule and track service appointments.
- Automate billing and feedback collection.
- Maintain transparency between garage owners, staff, and customers.

This phase focuses on identifying challenges such as delayed service tracking, missing customer data, and billing confusion — and proposing Salesforce automation as the solution.

1. Salesforce Developer Signup Page



The screenshot shows the Salesforce Developer Edition Signup Page. On the left, there is a blue banner with the Salesforce logo and the text "Build enterprise-quality apps fast and get hands-on with Agentforce and Data Cloud." Below this, it says "Sign up for your Developer Edition." and lists five benefits: "Build apps fast with drag-and-drop tools", "Go further with Apex code", "Build AI agents with Agentforce", "Harmonize your data with Data Cloud", and "Ground Agentforce with structured and unstructured data". At the bottom of the banner, there is a "Steps" section with "1 Select Type" highlighted. On the right, there is a white form titled "Sign up for your Developer Edition" with the subtitle "A free Salesforce Platform environment with Agentforce and Data Cloud". The form contains fields for "First name", "Last name", "Job title", "Work email", "Company", and "Country/Region" (with "India" selected). Below the form, there is a checkbox for "I agree to the Main Services Agreement - Developer Services and Salesforce Program Agreement" and a disclaimer about the use of the org.

Build enterprise-quality apps fast and get hands-on with Agentforce and Data Cloud.

Sign up for your Developer Edition.

- ✓ Build apps fast with drag-and-drop tools
- ✓ Go further with Apex code
- ✓ Build AI agents with Agentforce
- ✓ Harmonize your data with Data Cloud
- ✓ Ground Agentforce with structured and unstructured data
- ✓ Integrate with anything using APIs

Steps
1 Select Type

Sign up for your Developer Edition
A free Salesforce Platform environment with Agentforce and Data Cloud

First name

Last name

Job title

Work email

Company

Country/Region

Your org may be provisioned on or migrated to Hyperforce, Salesforce's public cloud infrastructure.

☐ I agree to the Main Services Agreement - Developer Services and Salesforce Program Agreement. I acknowledge, as described in the Developer Documentation: (1) the Developer Edition includes autonomous and other generative AI features; and (2) Salesforce may limit use of those features and the org, and may terminate any org that has been inactive for 45 days.

Figure 1.1: Salesforce Developer Org Signup Page

2.Salesforce Setup Home Page



2. Project Planning Phase

Lease Management in Salesforce is a customized solution designed to help organizations track, manage, and optimize real estate leases — whether for commercial, residential, or equipment leasing. By integrating lease operations into Salesforce, businesses can centralize tenant data, automate workflows.

modules planned:

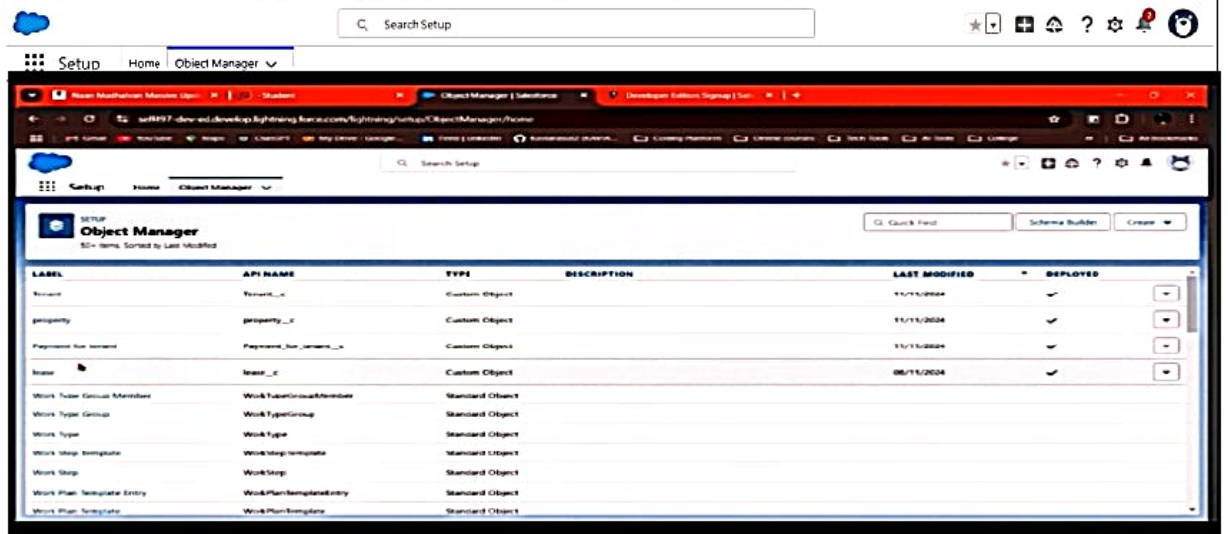
- 1.Streamline lease creation, approval, and renewal processes.**
- 2.Maintain a single source of truth for lease and tenant information.**
- 3.Automate rent invoicing and payment tracking.**
- 4.Improve visibility into lease expirations and revenue forecasting.**

- Salesperson: Limited access to assigned records.**

A structured timeline was followed:

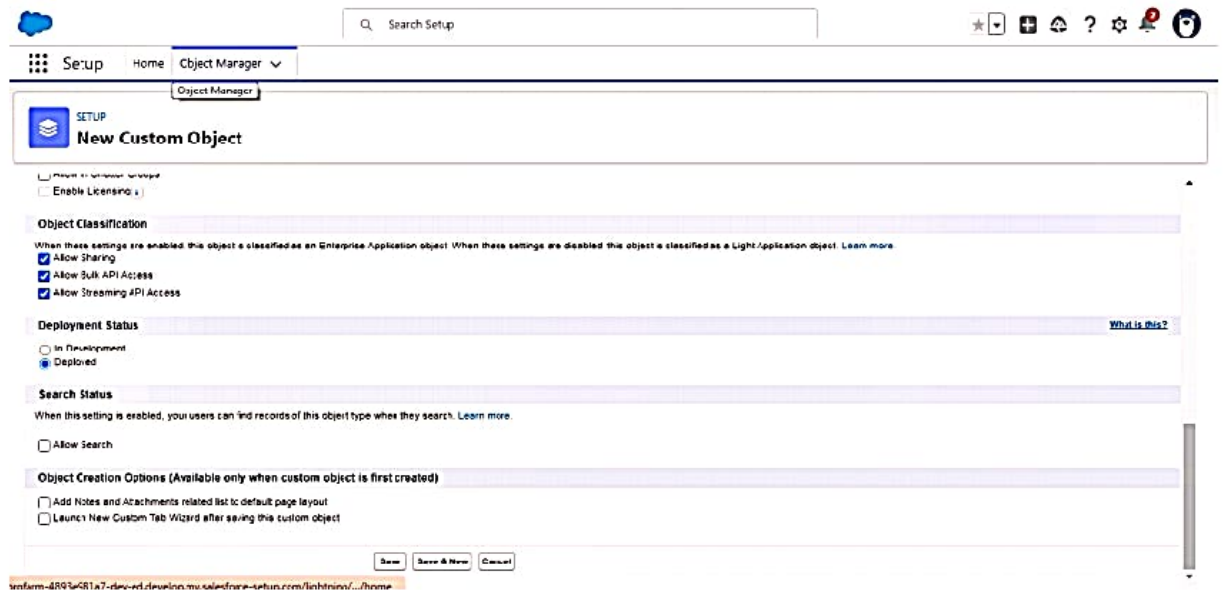
- 1. Object creation and setup**
- 2. Tab and Lightning app configuration**
- 3. Profile & Role Hierarchy setup**
- 4. Sharing settings and automation flows 5**
- . Testing and deployment**

1. Object Manager page – showing all custom objects.



Object Manager – Creating Custom Objects

2. List of all custom objects created.



3. Project Design Phase

The Project Design Phase is a crucial stage in developing the Lease Management System in Salesforce. It focuses on creating a blueprint of how the solution will function, look, and integrate within the Salesforce environment. This phase ensures that all system requirements are translated into a scalable, efficient, and user-friendly design.

1. System Architecture Design

Defines how different Salesforce components (Objects, Apex Classes, Lightning Web Components, Flows) interact.

2. User Interface (UI) & User Experience (UX) Design

Designs intuitive Lightning pages and custom components for easy navigation.

3. Data Model & Schema Design

Defines how lease-related data will be structured and stored in Salesforce.

1. Field creation (e.g., Gmail, Phone, Lookup).

The screenshot shows the Salesforce Setup interface for the 'property' object. The left sidebar contains a navigation menu with the following items: 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, and Restriction Rules. The main content area is titled 'property' and includes a 'Details' section with the following fields: Description, API Name (property__c), Custom (checked), Singular Label (property), and Plural Label (property). To the right of these fields are checkboxes for 'Enable Reports' (checked), 'Track Activities' (checked), 'Track Field History' (checked), 'Deployment Status' (Deployed), and 'Help Settings' (Standard Salesforce.com Help Window). 'Edit' and 'Delete' buttons are located in the top right corner.

This screenshot shows the 'property' object details in Salesforce Setup, with additional configuration options. The 'Context Sensitive Help Setting' is set to 'Open the standard Salesforce.com Help & Training window'. The 'Compact Name' is 'Property'. The 'Enter Record Name Label and Format' section includes a text box for 'Record Name' (property Name) and a dropdown for 'Data Type' (Text). A warning message states: 'Warning: If you plan to insert a high volume of records in this object, via the API for example, use the Text data type'. The 'Optional Features' section includes checkboxes for 'Allow Reports' (checked), 'Allow Activities' (checked), 'Track Field History' (checked), 'Allow in Chatter Groups' (unchecked), and 'Enable Licensing' (unchecked). The 'Object Classification' section includes checkboxes for 'Allow Sharing' (checked), 'Allow Bulk API Access' (checked), and 'Allow Streaming API Access' (checked). A note at the bottom states: 'When these settings are enabled, this object is classified as an Enterprise Application object. When these settings are disabled, this object is classified as a Light Application object. Learn more'.

Custom Field Creation in Customer Details Object

2. Lightning App setup showing all tabs.

The screenshot shows the Salesforce Lightning App Setup page. The browser address bar displays the URL: `sell497-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/home`. The page header includes a search bar labeled "Search Setup" and navigation tabs for "Setup", "Home", and "Object Manager". The "Object Manager" tab is active, showing a list of 51+ items sorted by label. The list includes standard objects like Account, Activity, Address, and Asset. The table has columns for LABEL, API NAME, TYPE, DESCRIPTION, LAST MODIFIED, and DEPLOYED.

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Account	Account	Standard Object			
Activity	Activity	Standard Object			
Address	Address	Standard Object			
Alternative Payment Method	AlternativePaymentMethod	Standard Object			
API Anomaly Event Store	ApiAnomalyEventStore	Standard Object			
Appointment Category	AppointmentCategory	Standard Object			
Appointment Invitation	AppointmentInvitation	Standard Object			
Appointment Invitee	AppointmentInvitee	Standard Object			
Appointment Topic Time Slot	AppointmentTopicTimeSlot	Standard Object			
Asset	Asset	Standard Object			
Asset Action	AssetAction	Standard Object			
Asset Action Source	AssetActionSource	Standard Object			

This screenshot is identical to the one above, showing the Salesforce Lightning App Setup page with the Object Manager tab active. The browser address bar displays the URL: `sell497-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/home`. The page header includes a search bar labeled "Search Setup" and navigation tabs for "Setup", "Home", and "Object Manager". The "Object Manager" tab is active, showing a list of 51+ items sorted by label. The list includes standard objects like Account, Activity, Address, and Asset. The table has columns for LABEL, API NAME, TYPE, DESCRIPTION, LAST MODIFIED, and DEPLOYED.

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Account	Account	Standard Object			
Activity	Activity	Standard Object			
Address	Address	Standard Object			
Alternative Payment Method	AlternativePaymentMethod	Standard Object			
API Anomaly Event Store	ApiAnomalyEventStore	Standard Object			
Appointment Category	AppointmentCategory	Standard Object			
Appointment Invitation	AppointmentInvitation	Standard Object			
Appointment Invitee	AppointmentInvitee	Standard Object			
Appointment Topic Time Slot	AppointmentTopicTimeSlot	Standard Object			
Asset	Asset	Standard Object			
Asset Action	AssetAction	Standard Object			
Asset Action Source	AssetActionSource	Standard Object			

Garage Management Lightning App Setup

4. Requirement Analysis

In this phase, both functional and non-functional requirements of the system are clearly defined.

Functional Requirements:

- 1. Add, view, and manage Customer, Appointment, and Billing records.**
- 2. Automate updates — for example, when payment is completed, mark status as “Paid.”**
- 3. Send email alerts automatically upon successful billing.**
- 4. Allow users to generate reports and dashboards for performance tracking.**

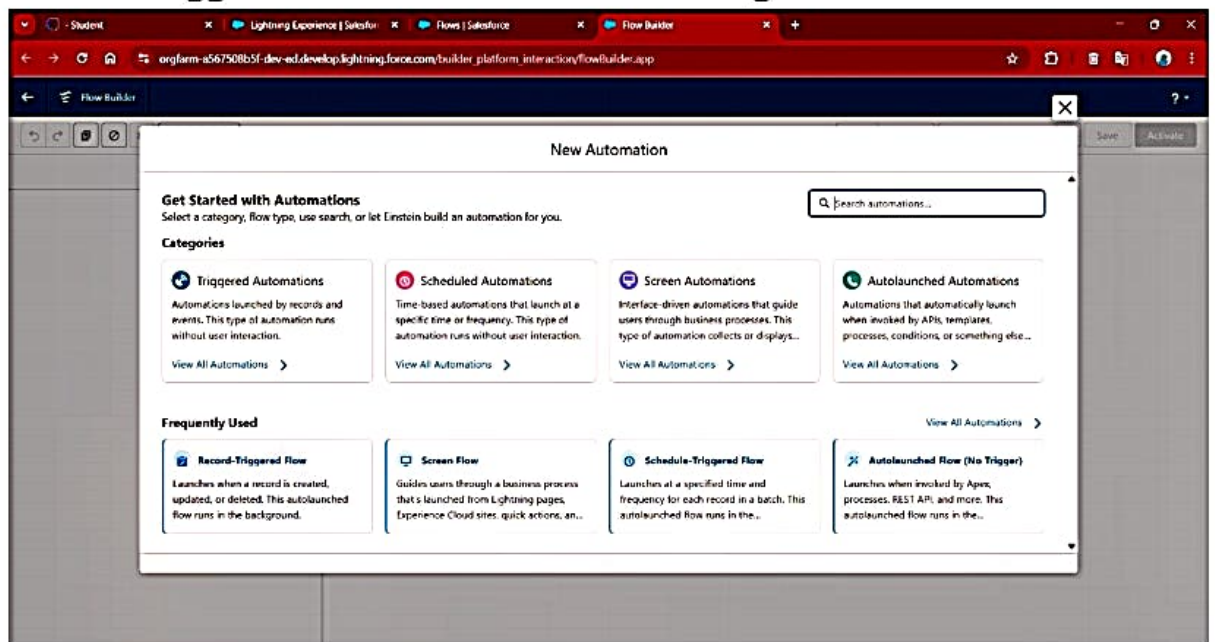
Non-Functional Requirements:

- 1. The system should be user-friendly with a simple interface.**
- 2. Data should be secured based on role-based access.**
- 3. Performance should be optimized for fast operations.**
- 4. Validation and duplicate prevention to ensure accuracy.**

Implementation Tools:

- Salesforce Objects, Flows, Apex Trigger, Validation & Duplicate Rules.**
- Reports and Dashboards for analysis.**

1. Record-triggered flow showing automation.

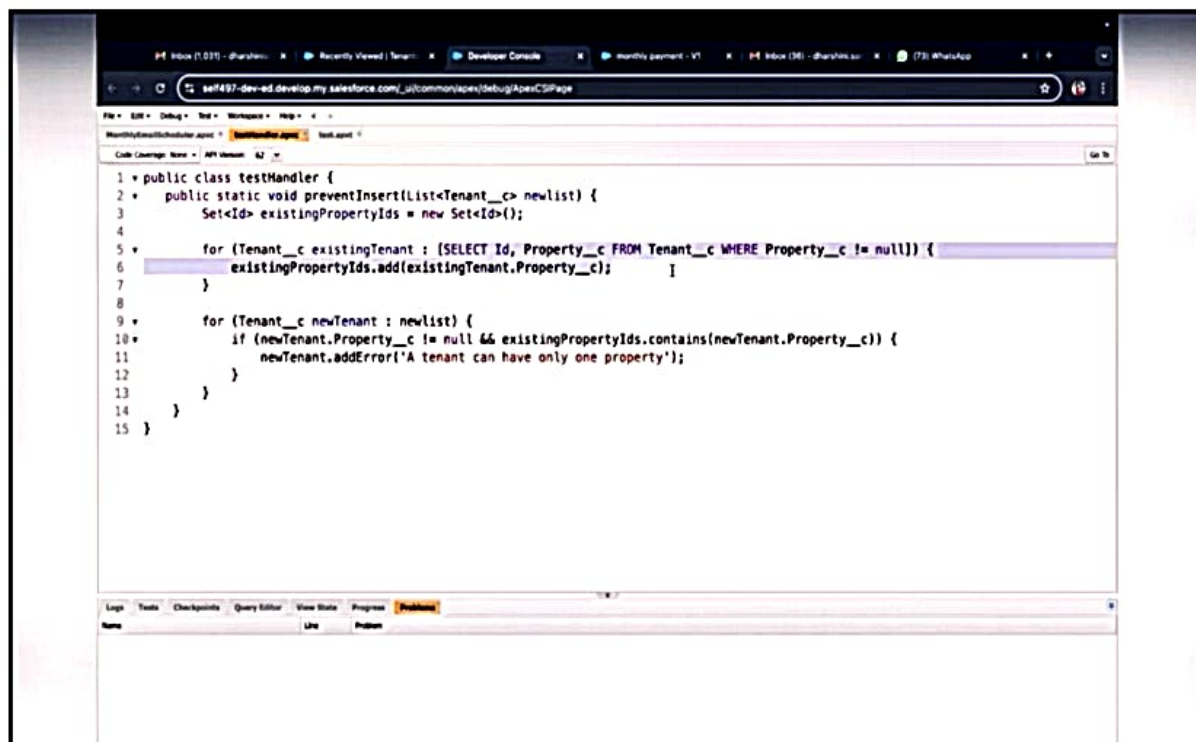
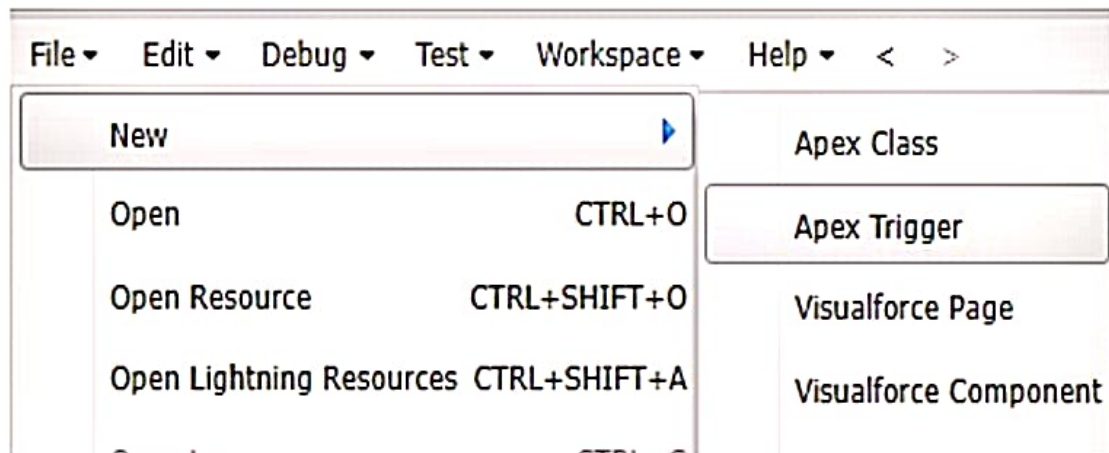


8. Api name : is auto populated

The screenshot shows the 'Edit Update Records' dialog in the Salesforce Flow Builder. The dialog has a title bar 'Edit Update Records'. Below the title bar, there is a description 'Update Salesforce records using values from the flow.' Below this, there are two input fields: '*Label' with the value 'Amount Update' and '*API Name' with the value 'Amount_Update'. Below these fields, there is a 'Description' field. Below the description field, there is a section titled '*How to Find Records to Update and Set Their Values' with four radio button options: 'Use the billing details and feedback record that triggered the flow' (selected), 'Update records related to the billing details and feedback record that triggered the flow', 'Use the IDs and all field values from a record or record collection', and 'Specify conditions to identify records, and set fields individually'. Below this section, there is a section titled 'Set Filter Conditions' with a dropdown menu 'Condition Requirements to Update Record' set to 'All Conditions Are Met (AND)'. At the bottom right, there are 'Cancel' and 'Done' buttons.

Record-Triggered Amount Update

2.Apex Trigger window in Developer Console.



Apex Trigger for Amount Distribution

5. Performance Testing

After implementation, the system underwent multiple tests Performed:

Performance testing is the process of evaluating the speed, responsiveness, stability, and scalability of the Lease Management application built on Salesforce. It ensures that the system performs efficiently under expected user load and data volume, especially when handling multiple leases, payments, and contract updates simultaneously.

1. Objectives of Performance Testing

Validate System Efficiency: Ensure that lease operations (such as creating, updating, or renewing leases) run smoothly without delays

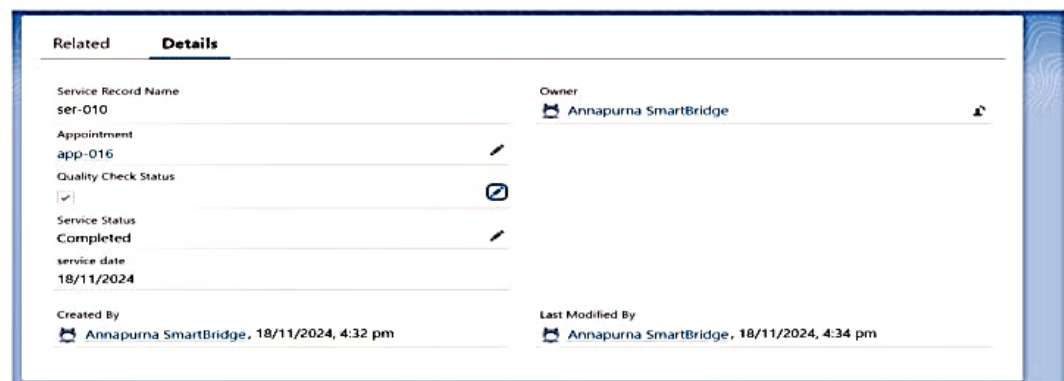
.2. Key Areas Tested

Lease Record Operations: Create, edit, delete, and search lease records.

Data Volume Testing: Handling large datasets (thousands of lease agreements).

1. Created Status records for all objects.

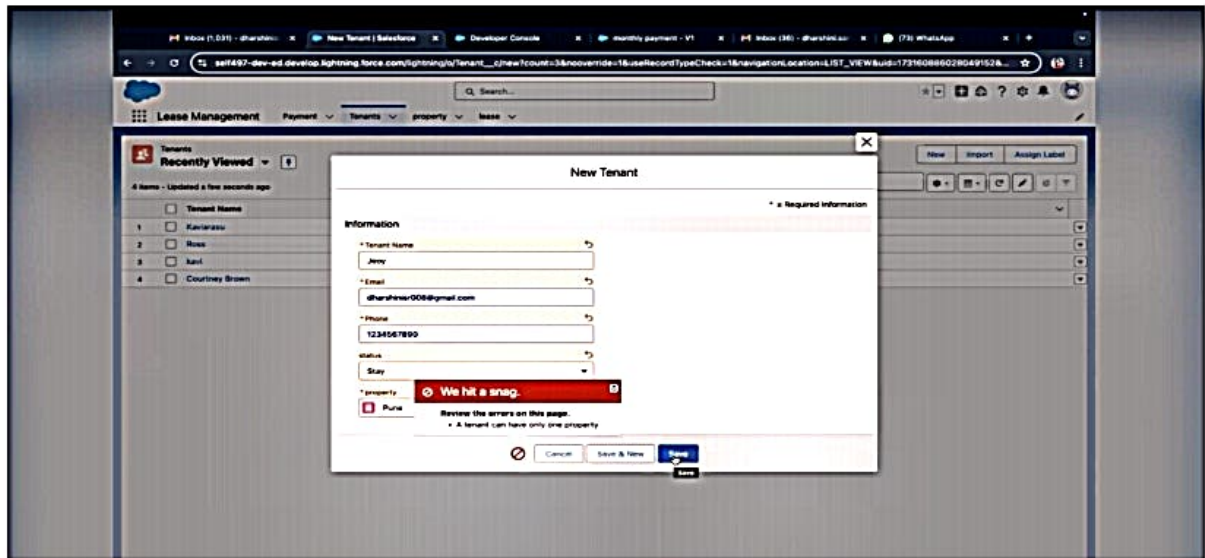
6. Now automatically Service status will be moved to completed.



The screenshot shows a Salesforce record detail page for a Service Record. The page has two tabs: 'Related' and 'Details', with 'Details' being the active tab. The record information is organized into two columns. The left column contains fields for 'Service Record Name' (ser-010), 'Appointment' (app-016), 'Quality Check Status' (with a checkmark icon), 'Service Status' (Completed), and 'service date' (18/11/2024). The right column contains 'Owner' (Annapurna SmartBridge) and 'Last Modified By' (Annapurna SmartBridge, 18/11/2024, 4:34 pm). At the bottom, the 'Created By' field shows 'Annapurna SmartBridge, 18/11/2024, 4:32 pm'. Edit icons (pencil) are visible next to the Appointment, Quality Check Status, and Service Status fields.

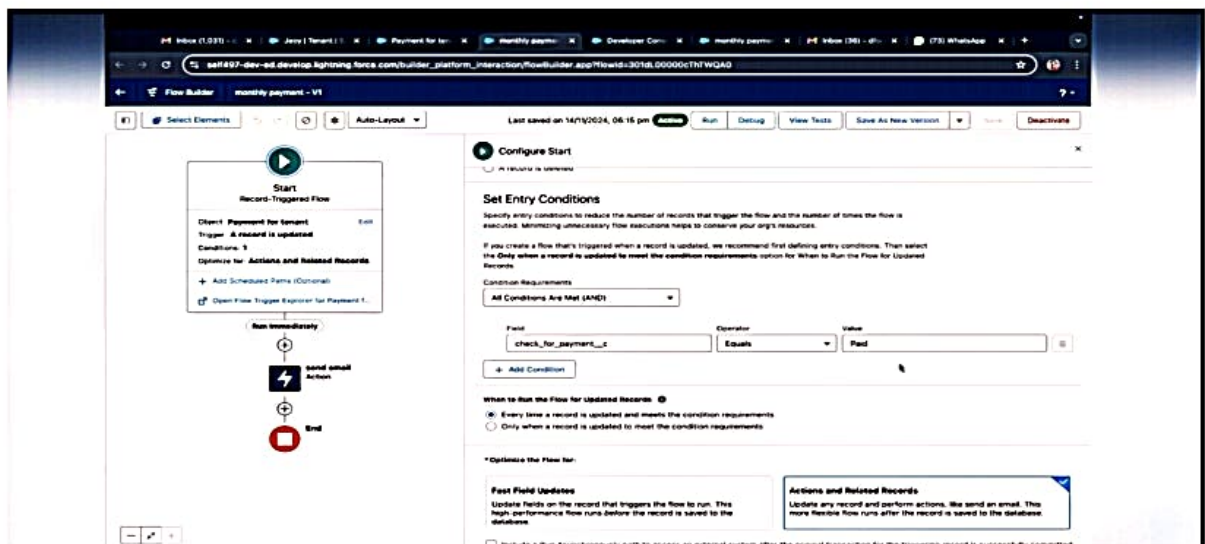
Details	
Service Record Name ser-010	Owner Annapurna SmartBridge
Appointment app-016	
Quality Check Status [✓]	
Service Status Completed	
service date 18/11/2024	
Created By Annapurna SmartBridge, 18/11/2024, 4:32 pm	Last Modified By Annapurna SmartBridge, 18/11/2024, 4:34 pm

2.Report tab showing generated report.



Service Information Report

3.Dashboard line chart view.



Service Rating Dashboard

Conclusion:

The Lease Management System built on Salesforce provides an efficient, transparent, and scalable solution for managing lease agreements, tenants, payments, and property details in one centralized platform. By leveraging Salesforce's powerful automation tools, custom objects, and analytics, the system enhances operational efficiency, reduces manual errors, and ensures better decision-making. align with business needs, user convenience, and Salesforce best practices. The final outcome demonstrates how Salesforce can transform traditional lease management into a smart, data-driven process with real-time insights, automated workflows, and seamless collaboration between teams.