Lease Management

1. Project Overview

This project focuses on the development of a "Lease management" using Salesforce. This project is to design and implement a comprehensive lease management system within Salesforce that allows businesses to efficiently manage the lifecycle of lease agreements, from initial setup through payments, renewals, and terminations. The system should automate tasks, centralize lease data, and provide analytics for decision-making.

Through this project, the aim is to:

- Centralized Lease Data.
- Lease Lifecycle Management**.**
- Automated Reminders & Alerts**.**
- Payment Tracking**.**
- Custom Reporting & Analytics**.**

2. Objectives

Business Goals:

1. Increase Operational Efficiency.
2. Enhance Data Accuracy & Visibility.
3. Improve Financial Control & Reporting.
4. Ensure Compliance & Risk Reduction.

Specific Outcomes:

- A user-friendly bussiness portal integrated into Salesforce.
- Automated Lease Workflow Management.
- Improved Payment Tracking and Financial Visibility.
- Increased Compliance & Risk Management.

3. Salesforce Key Features and Concepts Utilized

1. **Salesforce Objects**

   Custom objects for

   - Tenant.
   - Property.
   - payment for tenant.
   - lease.

2. Process Automation

1. Payment Schedule workflow:

Automate the generation of payment schedules based on lease terms, payment frequency, and amounts.

- Flows for email sending for payment detail and approval processes.
- Apex class and Apex Triggers for avoid duplicate property .

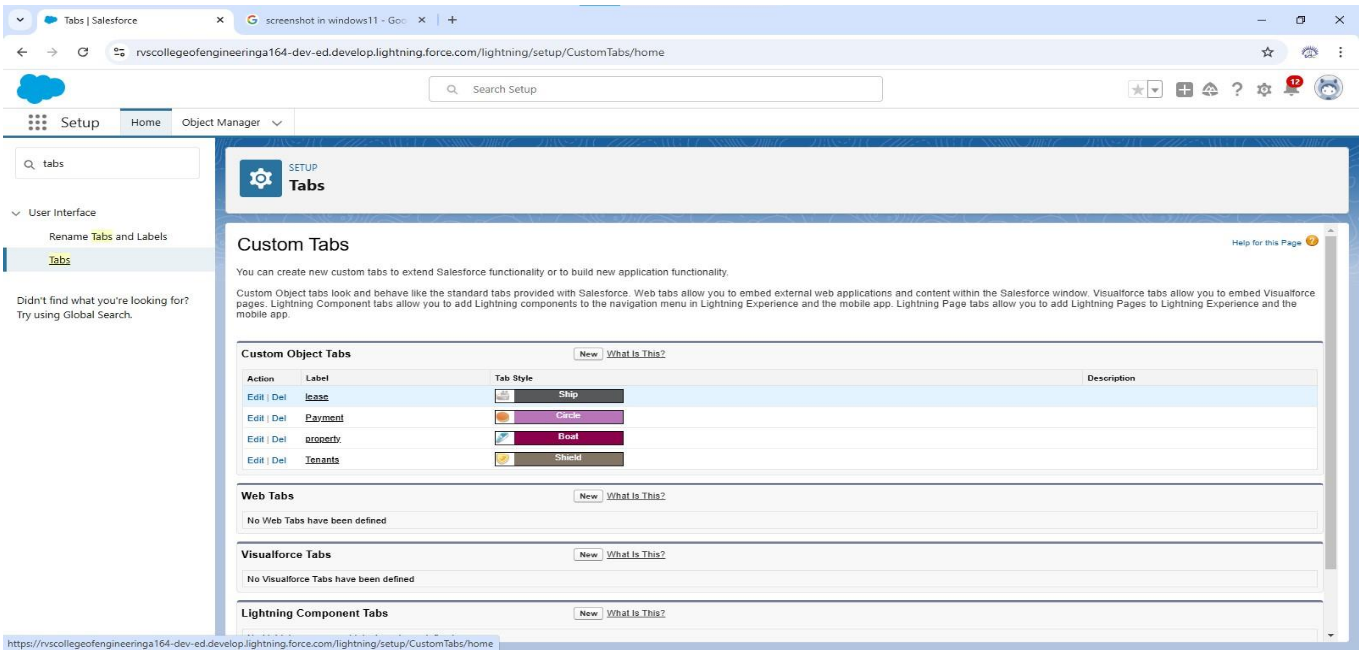
2.Billing & Payment Processing:

- Scheduled Flows can automatically create recurring billing records based on the lease schedule.
- Use **Apex triggers** to update the status of payments and send out reminders for overdue payments.

3. Lease Notifications & Alerts:

Use **Process Builder** or **Flow** to send notifications or emails to tenants or leasing agents for important lease-related events like:

- Lease expiration
- Payment reminders
- Insurance verification dates
- Maintenance schedules



SETUP
Tabs

Custom Tabs

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

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

Custom Object Tabs

Action	Label	Tab Style	Description
Edit | Del	lease	Ship	
Edit | Del	Payment	Circle	
Edit | Del	property	Boat	
Edit | Del	Tenants	Shield	

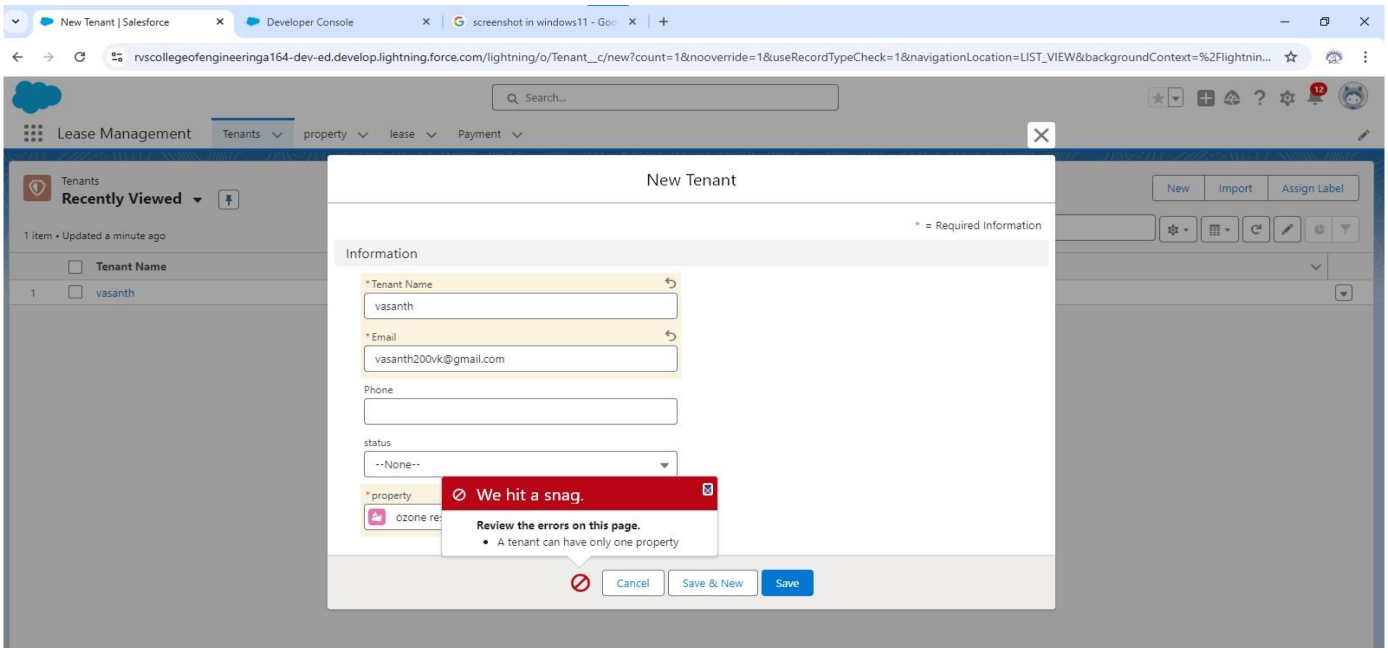
Web Tabs

No Web Tabs have been defined

Visualforce Tabs

No Visualforce Tabs have been defined

Lightning Component Tabs



New Tenant

* = Required Information

Information

*Tenant Name: vasanth

*Email: vasanth200vk@gmail.com

Phone:

status: --None--

*property: ozone re

We hit a snag.

Review the errors on this page.

- A tenant can have only one property

4.Detail steps to solution design

1. Lease Agreement (Custom Object)

Custom objects: Lease, property,payment for tenant,tenant.

- **Status** (Picklist: Active, Expired, Renewed, Terminated) **–** Current status of the lease.
- **Lease Name** (Text) **–** Unique identifier for the lease.

2.Tenant (Custom Object)

- **Tenant Name** (Text) – Name of the tenant or business.
- **Tenant Type** (Picklist: Individual, Business) – Type of tenant.
- **Contact Information** (Phone, Email, Address) – Tenant’s contact details.

3. Relationships Between Objects

- **Lease Agreement** has a **Lookup relationship** to **Tenant** (Account or Custom Tenant object) and **Property** (Custom object).
- **Lease Payment** has a **Lookup relationship** to **Lease Agreement**.

4.Lease Payment (Custom Object)

- **Payment Date** (Date) – Date when the payment is due.
- **Amount Paid** (Currency) – The amount the tenant paid.
- • **Payment Method** (Picklist: Credit Card, Bank Transfer, Cash, etc.) – Method used to make the payment.

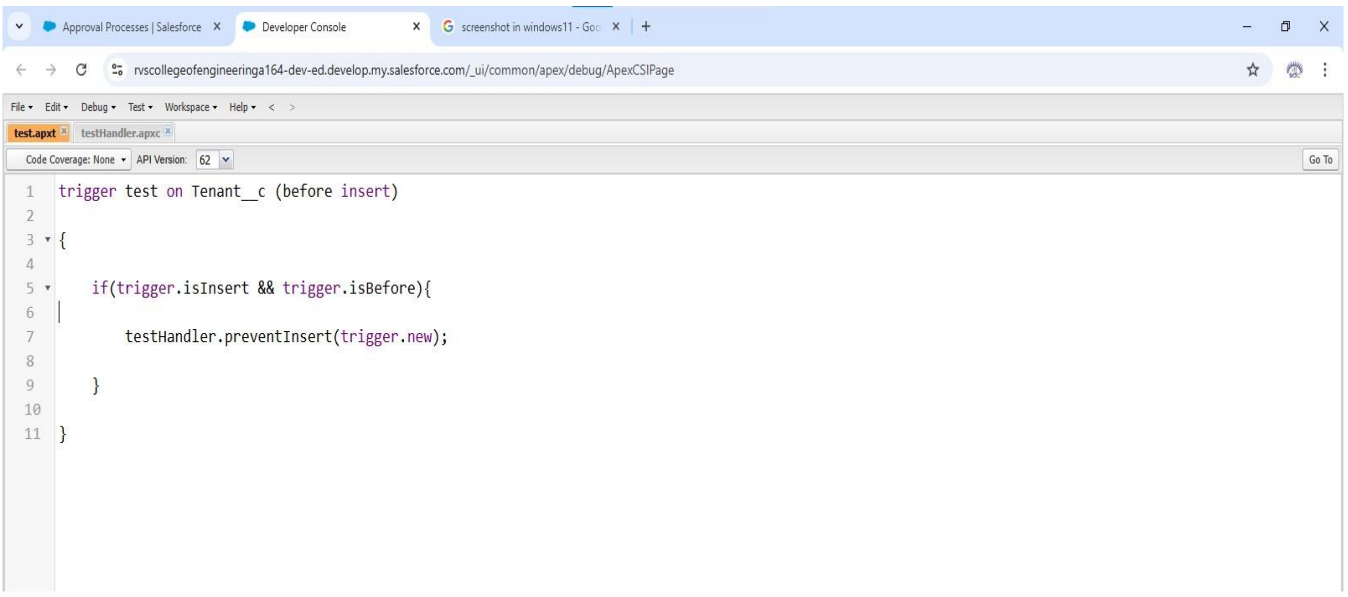
5.Testing and validation

1.User Interface Testing:

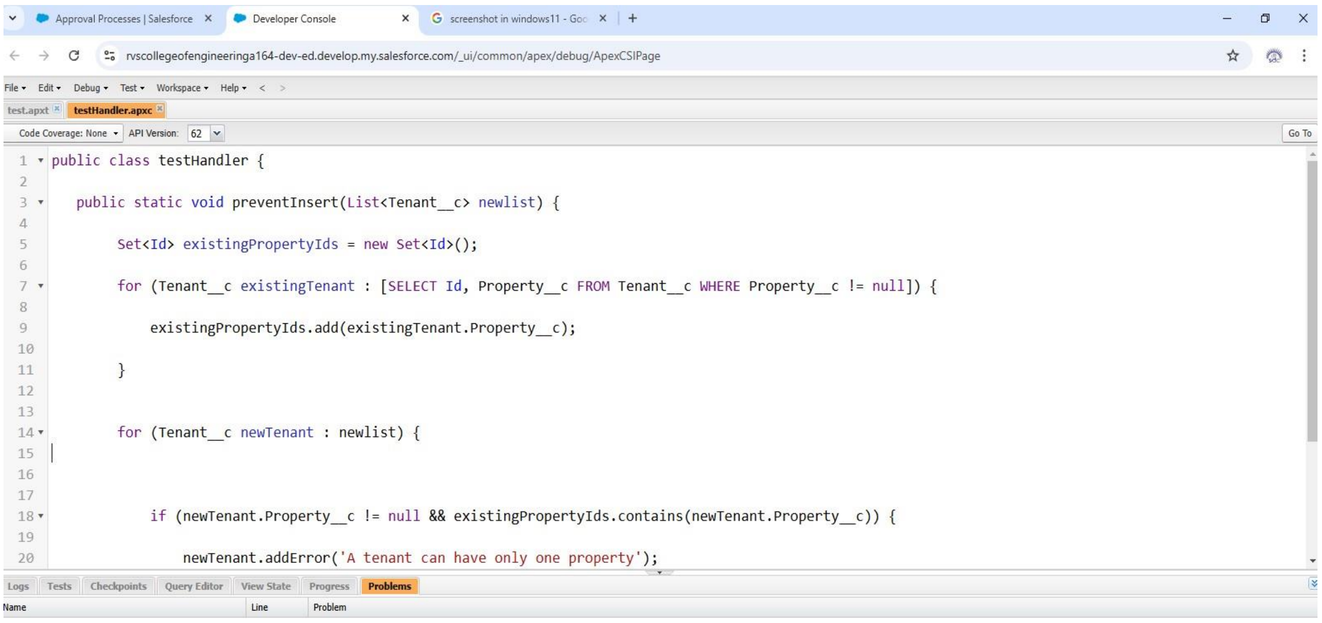
- Validated UI components across desktop and mobile platforms.
- Tested workflows for booking, inventory updates, and reporting.

2.Integration Testing:

- Validated integrations with external systems like payment gateways and SMS services.


Approval Processes | Salesforce
Developer Console
screenshot in windows11 - Goo
rvscollegeofengineeringa164-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
test.apxt
testHandler.apxc
Code Coverage: None
API Version: 62
Go To

```
trigger test on Tenant__c (before insert)

{

   if(trigger.isInsert && trigger.isBefore){

       testHandler.preventInsert(trigger.new);

   }

}
```



```
public class testHandler {

  public static void preventInsert(List<Tenant__c> newlist) {

      Set<Id> existingPropertyIds = new Set<Id>();

      for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {

          existingPropertyIds.add(existingTenant.Property__c);

      }


      for (Tenant__c newTenant : newlist) {



          if (newTenant.Property__c != null && existingPropertyIds.contains(newTenant.Property__c)) {

              newTenant.addError('A tenant can have only one property');
```

Approval Processes | Salesforce
Developer Console
screenshot in windows11 - Goo
rvscollegeofengineeringa164-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
test.apxt
testHandler.apxc
Code Coverage: None
API Version: 62
Go To
Logs Tests Checkpoints Query Editor View State Progress Problems
Name Line Problem

6.Key Scenarios Addressed by Salesforce in the Implementation Project

3. Lease Renewals & Extensions

- To identify leases that are due for renewal.
- Lease expired can renewal and extension.

2.Automated Email Notifications

- Real-time updates to inventory upon part usage during services.

3.Service Tracking:

- Technicians can update service status, and customers receive live updates.
-

4.Compliance Tracking with Custom Fields:

- Use custom fields and validation rules to track compliance-related data, such as whether tenants have submitted proof of insurance, completed necessary maintenance, or adhered to property usage guidelines.

7.Conclusion

The lease management built on Salesforce enhances the operational efficiencies, enhanced automation, and improved customer experience for organizations involved in property leasing, equipment leasing, or any business managing complex lease agreements. Salesforce offers a robust and flexible platform that can be customized to address a wide range of challenges .