

PROJECT REPORT

Naan Mudhalvan – Salesforce Developer

Project Title:
A CRM Application to Manage the Mall

Team Members:

Name	NM ID	32-digit NM ID
Naveen P	au712221205021	D69BC8CD087FCF49DBB42E60CFEF3095
Srikaran S	au712221205038	72CD71216FEAE48A5C746FB40B00A9E7
Mukesh Kumar Pandian M	au712221205018	72EBF622E5DEB93BA8241AD559A755AA
Santhosh Kumar P	Au712221205032	B23A374F5E48E9C9048434568AEB9BC5



ANNA UNIVERSITY
Park College Of Engineering And Technology

A CRM Application to Manage the Mall

Project Overview:

The Mall Management CRM Application is a comprehensive solution designed to streamline operations, enhance customer engagement, and improve the efficiency of mall management. This system enables mall administrators to manage retail tenants, track lease agreements, and monitor operational metrics seamlessly. The application simplifies complex workflows with features like tenant onboarding, rental billing, and maintenance request tracking. Additionally, it includes customer relationship management tools to capture and analyze shopper data, facilitate loyalty programs, and drive targeted marketing campaigns. Real-time dashboards and reporting capabilities provide actionable insights, helping stakeholders make informed decisions. The application is built with scalability and user-friendliness in mind, ensuring it adapts to malls of varying sizes and operational needs. Security and data integrity are prioritized through robust authentication mechanisms and secure cloud integration. Overall, the CRM application empowers mall managers to optimize resources, foster better tenant relations, and enhance the shopping experience for customers.

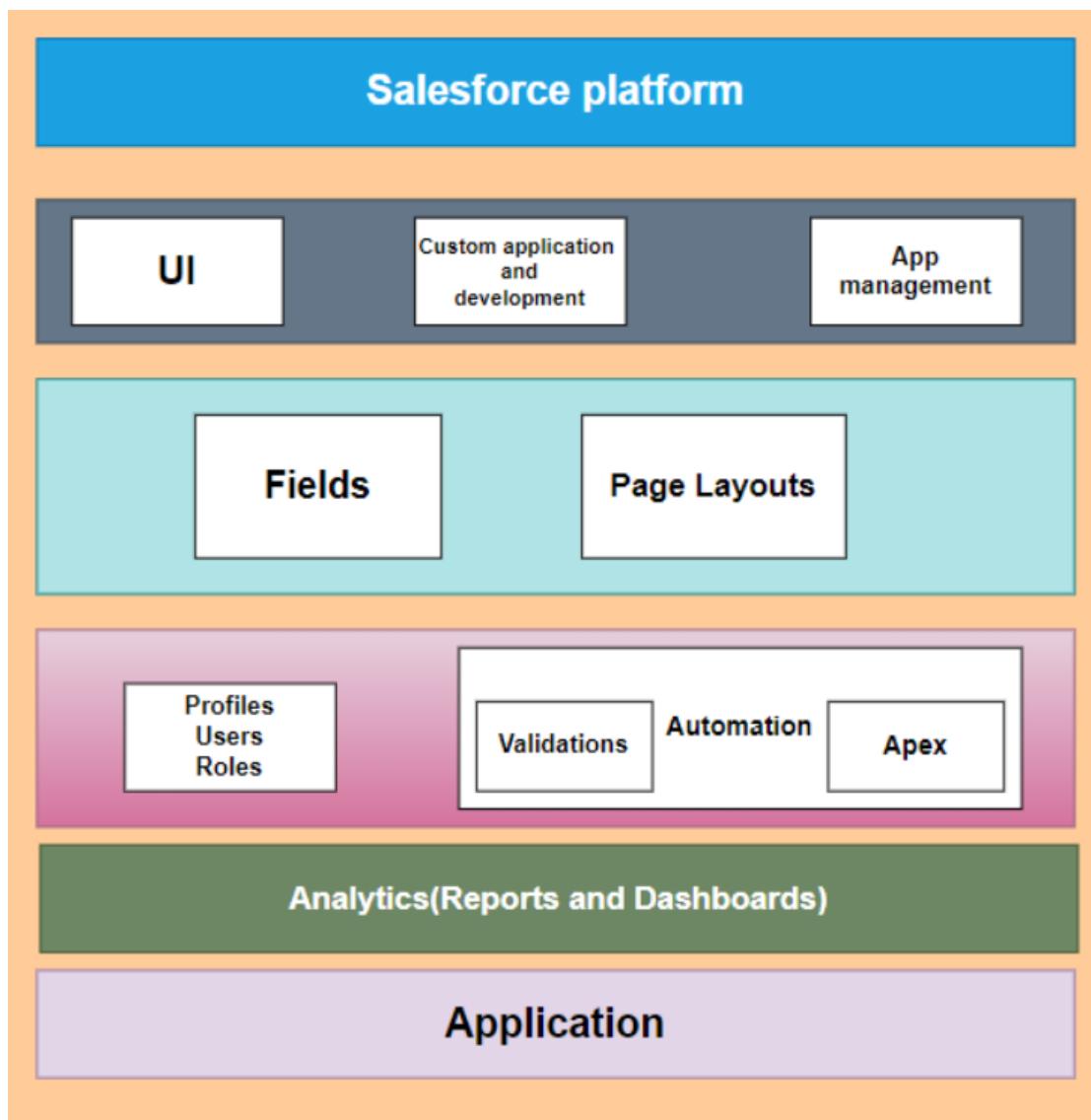
Project Description:

“A CRM application to manage the mall” project involves developing a comprehensive CRM application on Salesforce for managing and streamlining operations within a shopping mall. The primary objective is to centralize customer and tenant data, optimize communication, improve tenant relationships, enhance customer satisfaction, and ultimately drive revenue growth for the mall. The CRM will offer robust features to manage tenant interactions, track mall activities, analyse customer behaviours, and provide actionable insights for mall management.

Short Description:

“A CRM application to manage the mall” is a project to build a Salesforce-based solution that centralizes customer and tenant data, streamlines mall operations, and enhances tenant and customer relationships.

Technical Architecture:

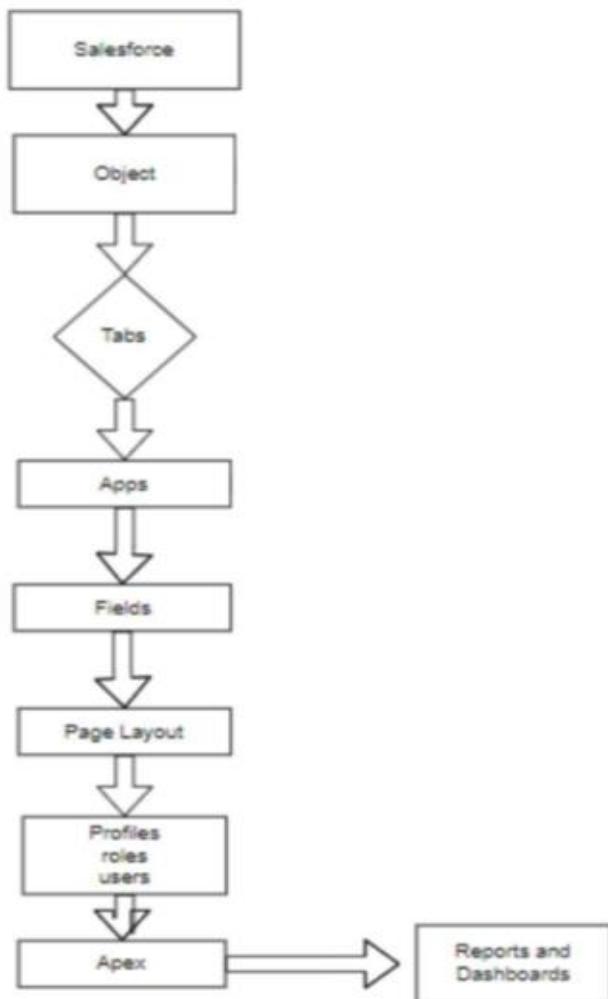


Objective:

Objectives of the Mall Management CRM Application:

1. Streamline Tenant Management: Centralize tenant onboarding, lease tracking, and rental billing processes to ensure efficient and transparent management.
 2. Enhance Customer Engagement: Implement tools to analyze shopper behavior, manage loyalty programs, and execute personalized marketing campaigns to improve customer retention.
 3. Optimize Operational Efficiency: Simplify maintenance requests, track inventory, and manage all resources effectively to reduce downtime and operational bottlenecks.
 4. Provide Actionable Insights: Utilize real-time dashboards and analytics to offer data-driven insights for informed decision-making by mall administrators.
 5. Ensure Data Security and Scalability: Build a secure, scalable platform capable of handling sensitive tenant and customer data while adapting to the needs of malls of different sizes.

Project Flow:



System Requirements:

- Windows 8 machine
- Install with two web browsers
- Bandwidth of 30mbps

What you'll learn

1. Real-Time Salesforce Project
2. Data Modelling
3. Creating an application
4. User Interface Customization
5. Importing bulk amounts of data
6. Security in Salesforce
7. Group Collaboration
8. Reports & Dashboards

Use Case:

The CRM application for mall management on Salesforce serves as a centralized platform to streamline tenant and customer interactions. Maintaining detailed tenant profiles and lease records, it simplifies lease renewals, payment tracking, and performance assessments. Mall managers can monitor store sales data, helping to analyse trends and optimize tenant placement and promotions. The system also enhances customer engagement, supporting targeted marketing based on customer preferences and visit history. Event and campaign management tools allow tenants to collaborate on mall-wide promotions, increasing customer footfall and loyalty. Additionally, customer feedback and issue resolution features ensure timely responses, improving customer satisfaction. Real-time analytics and dashboards give mall administrators insights into operational KPIs, supporting data-driven decision-making. With robust security and data protection, the CRM system enables secure, efficient, and insightful mall management for both tenants and administrators.

Detailed Steps to Solution Design

Milestone 1-Salesforce

Introduction:

Are you new to Salesforce? Not sure exactly what it is, or how to use it? Don't know where you should start on your learning journey? If you've answered yes to any of these questions, then you're in the right place. This module is for you.

Welcome to Salesforce! Salesforce is a game-changing technology, with a host of productivity-boosting features, that will help you sell smarter and faster. As you work toward your badge for this module, will take you through these features and answer the question, what is Salesforce, anyway?

What Is Salesforce?

Salesforce is your customer success platform, designed to help you sell, service, market, analyse, and connect with your customers.

Salesforce has everything you need to run your business from anywhere. Using standard products and features, you can manage relationships with prospects and customers, collaborate and engage with employees and partners, and store your data securely in the cloud.

So what does that mean? Well, before Salesforce, your contacts, emails, follow-up tasks, and prospective deals might have been organized something like this:

<https://youtu.be/r9EX3lGde5k>

Use Case:

Creating a Salesforce Developer Edition org allows developers to experiment, innovate, and build customized solutions within a controlled environment. With access to Salesforce's powerful development tools and features, developers can prototype, test, and refine their applications, empowering them to deliver robust and tailored solutions to meet unique business requirements. As a Salesforce Administrator for The Smart Bridge, you must have a Salesforce developer edition org in order to do all the required works which the CEO desires for The Smart Bridge.

Before creating our developer account, we must know what are the types of Editions Salesforce offers.

Types of Salesforce Editions:

1	Essentials	Designed for small businesses getting started with CRM to boost sales or service productivity. It includes a setup assistant and administration tools to customize your deployment as you grow.
2	Professional	Designed for businesses requiring full-featured CRM functionality. It includes straightforward and easy-to-use customization, integration, and administration tools to facilitate any small to midsize deployment.

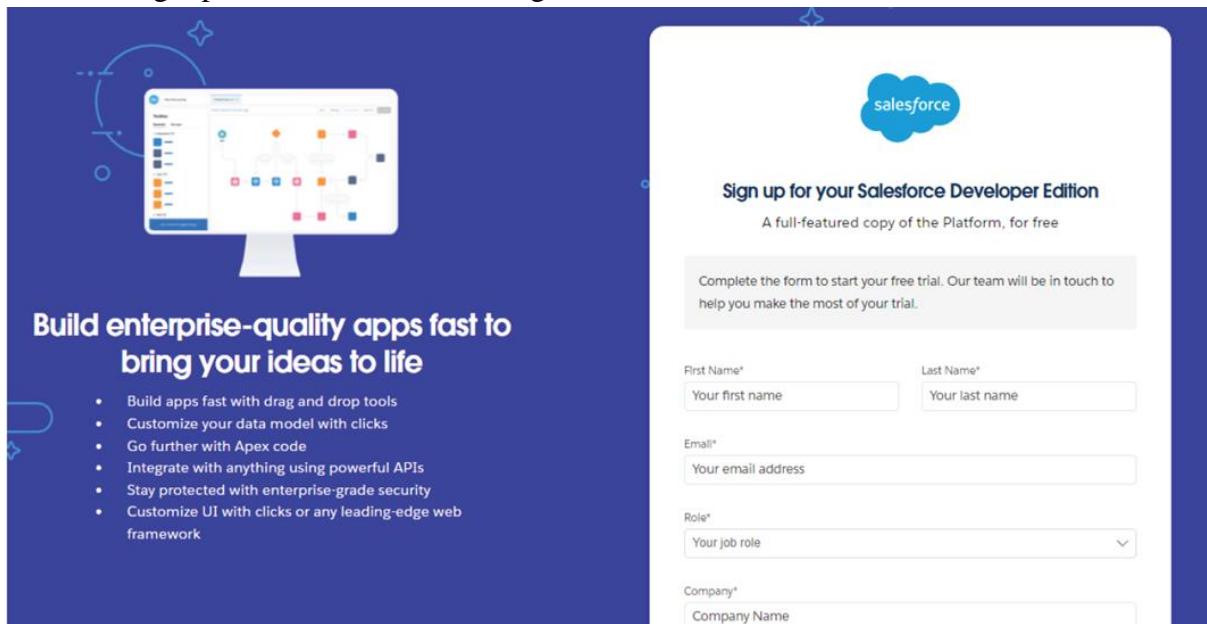
3	Enterprise	Meets the needs of large and complex businesses. It gives you advanced customization and administration tools, in addition to all the functionality available in Professional Edition, that can support large-scale deployments. Enterprise Edition also includes access to Salesforce APIs, so you can easily integrate with back-office systems.
4	Unlimited	Maximizes your success and extends it across the entire enterprise through the Lightning Platform. It gives you new levels of platform flexibility for managing and sharing all your information on demand. Includes all Enterprise Edition functionality, Premier Support, full mobile access, unlimited custom apps, increased storage limits, and other features.
5	Developer	Provides access to the Lightning Platform and APIs. It lets developers extend Salesforce, integrate with other applications, and develop new tools and applications. Developer Edition also provides access to many of the features available in Enterprise Edition

Let's begin with creating our Salesforce Developer Account.

Activity 1: Creating a Developer Account

Creating a developer org in Salesforce.

1. Go to <https://developer.salesforce.com/signup>
2. On the signup form, enter the following details:

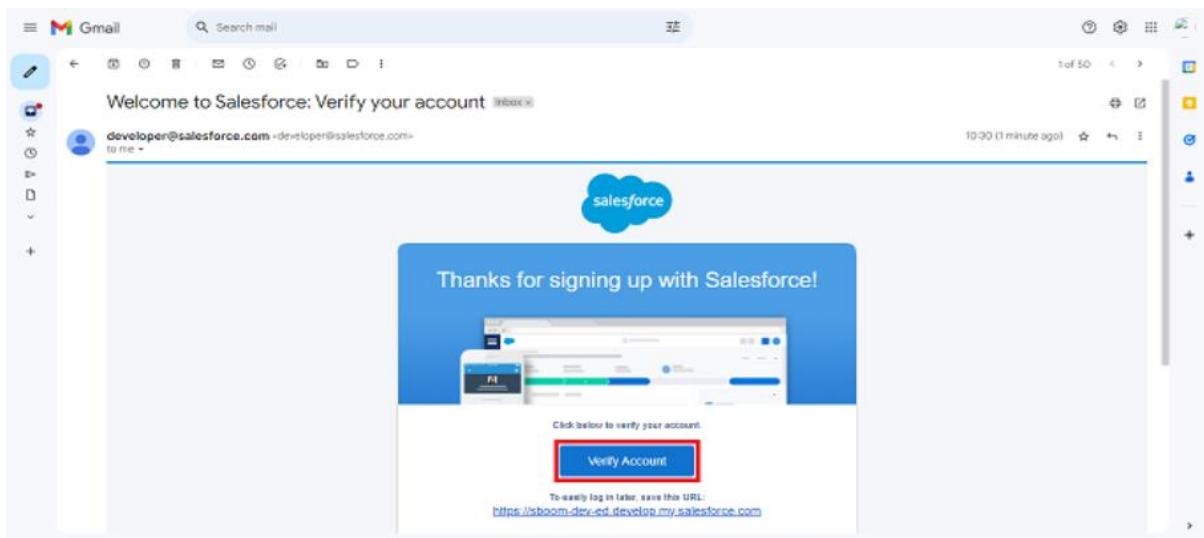


- 1) First name & Last name
- 2) Email
- 3) Role: Developer
- 4) Company: College Name

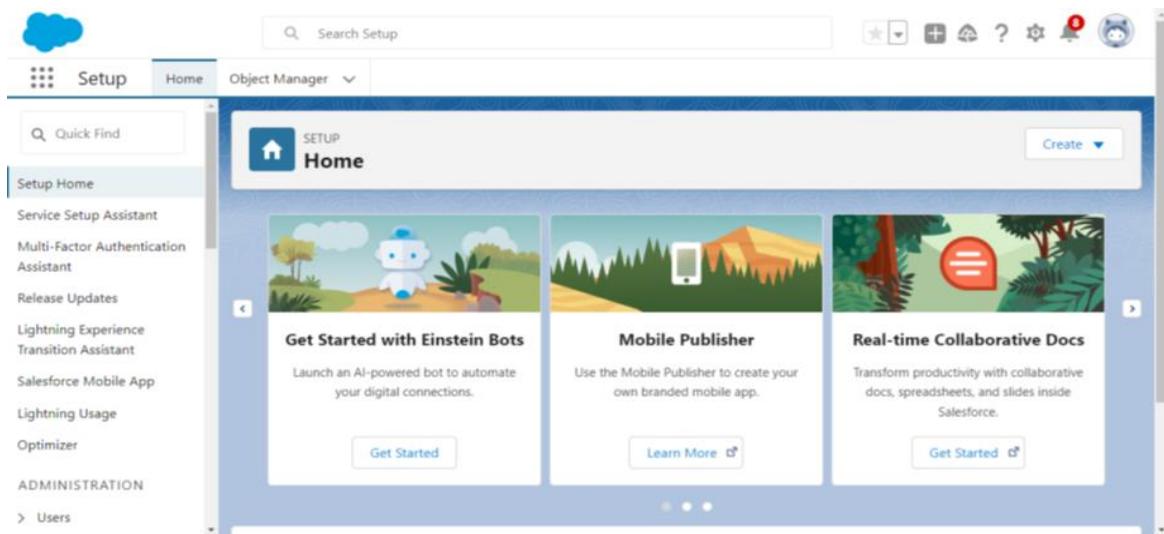
- 5) County: India
 - 6) Postal Code: pin code
 - 7) Username: should be a combination of your name and company
This need not be an actual email ID; you can give anything in the format:
username@organization.com
- Click on Sign me up after filling these.

Activity 2: Account Activation

1. Go to the inbox of the email that you used while signing up. Click on the verify account to activate your account. The email may take 5-10mins.
2. Click on Verify Account
3. Give a password answer a security question and click on change password.

A screenshot of a "Change Your Password" form. The title is "Change Your Password". It asks for a new password and specifies requirements: "Enter a new password for lead@sb.com. Make sure to include at least: 8 characters, 1 letter, 1 number". There are two input fields: "New Password" and "Confirm New Password", both of which are highlighted with a red rectangle. Below these are "Security Question" and "Answer" fields, also highlighted with a red rectangle. At the bottom is a "Change Password" button, which is also highlighted with a red rectangle.

4. Then you will redirect to your salesforce setup page.



Milestone 2- Object

What Is an Object?

Salesforce objects are database tables that permit you to store data that is specific to an organization.

What are the types of Salesforce objects?

Salesforce objects are of two types:

- 1. Standard Objects:** Standard objects are the kind of objects that are provided by salesforce.com such as users, contracts, reports, dashboards, etc.
- 2. Custom Objects:** Custom objects are those objects that are created by users. They supply information that is unique and essential to their organization. They are the heart of any application and provide a structure for sharing data.

In Salesforce, a custom object is a database table that allows you to store data unique to your organization. While Salesforce provides standard objects (like Account, Contact, Opportunity, etc.) that cover common CRM functions, custom objects let you define and manage data that's specific to your business.

Use Case:

Creating an object in a Salesforce organization is essential for efficient data management and process automation. By defining custom objects, businesses can structure and store data specific to their needs, enabling streamlined workflows, personalized reporting, and enhanced user experiences. Objects serve as the foundation for organizing and leveraging critical information within Salesforce. As an Admin for The Smart Bridge, It's your responsibility to store the data as per the organisation needs.

Create Custom Objects

To store the data as per business requirements.

Activity 1: Create Tenant Object

1. From the setup page ==> Click on Object Manager ==> Click on Create ==> Click on Custom Object.

Object Manager					
Label	API Name	Type	Description	Last Modified	Deployed
Account	Account	Standard Object		06/11/2024	✓
Activity	Activity	Standard Object		06/11/2024	✓
Address	Address	Standard Object		06/11/2024	✓
Alternative Payment Method	AlternativePaymentMethod	Standard Object		06/11/2024	✓
API Anomaly Event Store	ApianomalyEventStore	Standard Object		06/11/2024	✓
Appointment Category	AppointmentCategory	Standard Object		06/11/2024	✓
Appointment Invitation	AppointmentInvitation	Standard Object		06/11/2024	✓
Appointment Invitee	AppointmentInvitee	Standard Object		06/11/2024	✓
Appointment Topic Time Slot	AppointmentTopicTimeSlot	Standard Object		06/11/2024	✓
Asset	Asset	Standard Object		06/11/2024	✓
Asset Action	AssetAction	Standard Object		06/11/2024	✓
Asset Action Source	AssetActionSource	Standard Object		06/11/2024	✓
Asset Relationship	AssetRelationship	Standard Object		06/11/2024	✓
Asset State Period	AssetStatePeriod	Standard Object		06/11/2024	✓
Assigned Resource	AssignedResource	Standard Object		06/11/2024	✓
Associated Location	AssociatedLocation	Standard Object		06/11/2024	✓
Async Operation Tracker	AsyncOperationTracker	Standard Object		06/11/2024	✓
Authorization Form	AuthorizationForm	Standard Object		06/11/2024	✓

2. Enter the label name: Tenant
3. Plural label name: Tenants
4. Enter Record Name Label and Format
 - Record Name: Tenant Name
 - Data Type: Text

New Custom Object

Custom Object Definition Edit

Custom Object Information

The singular and plural labels are used in table, page layouts, and reports.

Label	<input type="text" value="Tenant"/>	Example: Account
Plural Label	<input type="text" value="Tenants"/>	Example: Accounts

Maintain visual context:

The Object Name is used when referencing the object via the API.

Object Name	<input type="text" value="Tenant__c"/>	Example: Account
-------------	--	------------------

Description:

Content-Sensitive Help Setting: Open the standard Salesforce core Help & Training window Open a window using a Visualforce page

Content Name:

Enter Record Name Label and Format

The Record Name appears in page layouts, key lists, related lists, and search results. For example, the Record Name for Account is "Account Name" and for Case it is "Case Number". Note that the Record Name field is always called "Name" when referenced via the API.

Record Name	<input type="text" value="Tenant Name"/>	Example: Account Name
-------------	--	-----------------------

Data Type: Warning: If you plan to insert a high volume of records in this object, via the API for example, use the Text data type.

Optional Features

Allow Reports
 Allow Views
 Allow Page History
 Allow in Chatter Groups

5. Click on Allow reports.
6. Allow search and Save

Object Manager					
Label	API Name	Type	Description	Last Modified	Deployed
Tenant	Tenant__c	Custom Object		06/11/2024	✓
Tenant Issue	Tenant_Issue__c	Custom Object		06/11/2024	✓

Activity 2: Create Lease Tracking Object

1. Enter the label name ==>Lease Tracking
2. Plural label name ==> Leases Tracking
3. Enter Record Name Label and Format
 - Record Name: Lease Tracking No
 - Data Type: Auto Number
 - Display Format - TT - {000000}
4. Click on Allow reports.

The screenshot shows the 'Object Manager' section of the Salesforce setup. A specific object named 'Lease Tracking' is being edited. The 'Custom Object Definition Edit' screen displays the following details:

- Custom Object Information:** Shows the label 'Lease Tracking' and the plural label 'Leases Tracking'.
- Description:** An empty text input field.
- Context-Sensitive Help Setting:** Options include 'Open the standard Salesforce.com Help & Training window' (selected) and 'Open a window using a Visualforce page'.
- Enter Record Name Label and Format:** Shows the record name 'Lease Tracking No', data type 'Auto Number', and display format 'TT - {000000}'.
- Optional Features:** A checkbox for 'Allow Reports' is checked, while others like 'Allow Activities', 'Track Field History', and 'Allow in Chatter Groups' are unchecked.

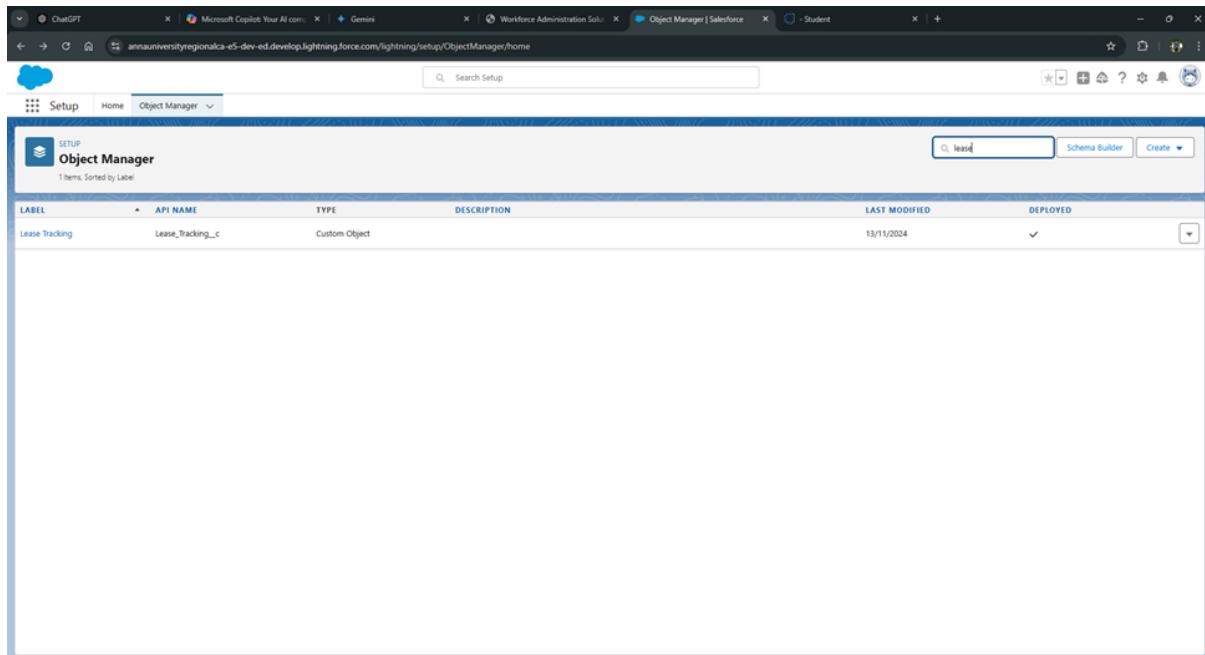
5. Allow search and Save.

The screenshot shows the 'Object Manager' table listing the 'Lease Tracking' object. The table has columns for LABEL, API NAME, TYPE, DESCRIPTION, LAST MODIFIED, and DEPLOYED. The data for 'Lease Tracking' is as follows:

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Lease Tracking	Lease_Tracking_c	Custom Object		13/11/2024	✓

Activity 3: Create Tenant Issues Object

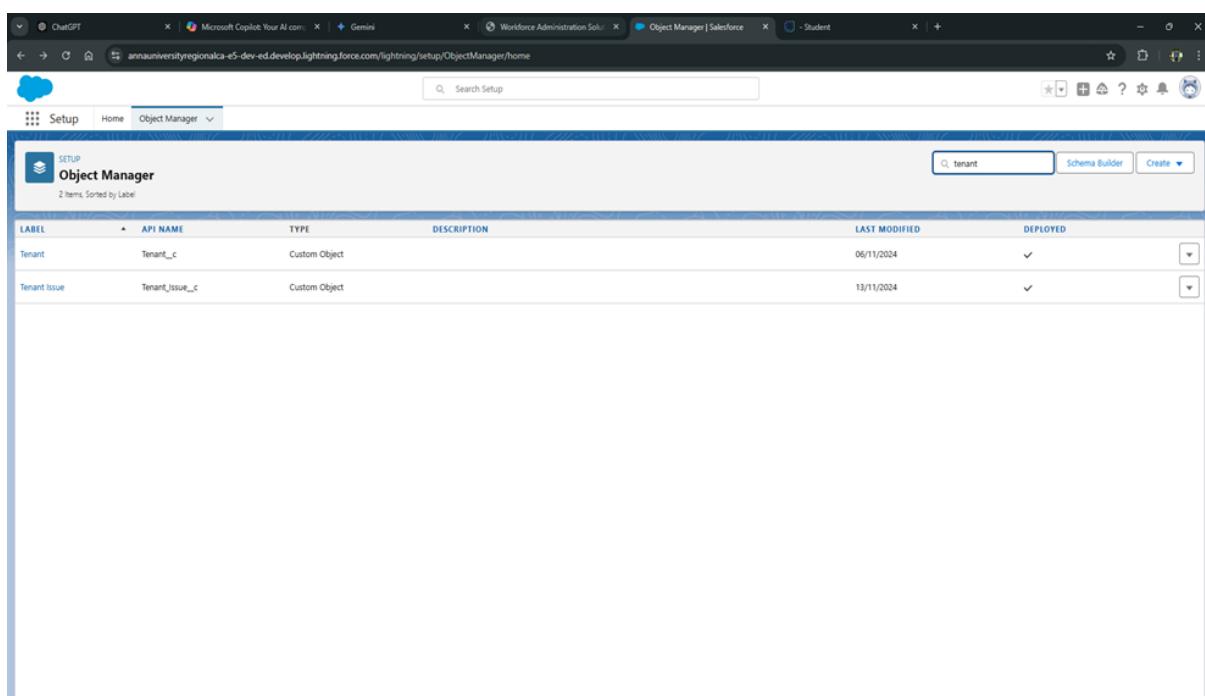
1. Enter the label name ==> Tenant Issue
2. Plural label name ==> Tenant Issues
3. Enter Record Name Label and Format
 - Record Name: Issues
 - Data Type: Auto number
4. Click on Allow reports.



The screenshot shows the Salesforce Object Manager interface. At the top, there are several tabs: Setup, Home, and Object Manager. The main title is "Object Manager" with a subtitle "1 items, Sorted by Label". A search bar at the top right contains the text "Lease". Below the header is a table with the following columns: LABEL, API NAME, TYPE, DESCRIPTION, LAST MODIFIED, and DEPLOYED. There is one row in the table:

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Lease Tracking	Lease_Tracking__c	Custom Object		13/11/2024	✓

5. Allow search and Save.



The screenshot shows the Salesforce Object Manager interface after creating two new objects. The search bar at the top right now contains the text "tenant". The table below shows two rows:

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Tenant	Tenant__c	Custom Object		06/11/2024	✓
Tenant Issue	Tenant_Issue__c	Custom Object		13/11/2024	✓

Milestone 3- Tabs

What is Tab?

A tab is like a user interface that is used to build records for objects and to view the records in the objects.

Types of Tabs:

1. Custom Tabs:

Custom object tabs are the user interface for custom applications that you build in salesforce.com. They look and behave like standard salesforce.com tabs such as accounts, contacts, and opportunities.

2. Web Tabs:

Web Tabs are custom tabs that display web content or applications embedded in the salesforce.com window. Web tabs make it easier for your users to quickly access content and applications they frequently use without leaving the salesforce.com application.

3. Visualforce Tabs:

Visualforce Tabs are custom tabs that display a Visualforce page. Visualforce tabs look and behave like standard salesforce.com tabs such as accounts, contacts, and opportunities.

4. Lightning Component Tabs:

Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app.

5. Lightning Page Tabs:

Lightning Page Tabs let you add Lightning Pages to the mobile app navigation menu.

Lightning Page tabs don't work like other custom tabs. Once created, they don't show up on the All Tabs page when you click the Plus icon that appears to the right of your current tabs. Lightning Page tabs also don't show up in the Available Tabs list when you customize the tabs for your apps.

Use Case:

Creating Objects and storing The Smart Bridge organization's data is the very first step in the requirements they want. Now to access the stored data by an employee from the organization Admin needs to create Tabs. By designing a dedicated Tab, businesses can improve user experience, simplify navigation, and provide quick access to critical information, enhancing productivity and ensuring efficient utilization of Salesforce's capabilities.

Activity 1: Create a custom tab for the tenant object

1. Go to setup page ==> type Tabs in Quick Find bar ==>click on tabs ==> New (under custom object tab).
2. Select Object (Tenant) ==> Select the tab style ==> Next (Add to profiles page) keep it as default ==> Next (Add to Custom App) uncheck the include tab ==> Save.
3. Make sure to append a tab to users' existing personal customizations is checked.

The screenshot shows the Salesforce Setup interface with the 'Tabs' tab selected under 'User Interface'. The main content area is titled 'Custom Tabs' and contains sections for 'Custom Object Tabs', 'Web Tabs', 'Visualforce Tabs', 'Lightning Component Tabs', and 'Lightning Page Tabs'. Under 'Custom Object Tabs', there is a table with three rows:

Action	Label	Tab Style	Description
Edit	Leases_Tracking	Basic	
Edit	Tenant_Basis	Computer	
Edit	Tenants	Apple	

The screenshot shows the 'New Custom Object Tab' wizard, Step 1: Enter the Details. The page title is 'New Custom Object Tab' and it indicates 'Step 1 of 3'. It has a required information notice. The 'Object' dropdown is set to 'None' and the 'Tab Style' dropdown is set to 'Basic'. There is an optional field for a 'Splash Page Custom Link' which is also set to 'None'. A text input field for 'Description' is present, and at the bottom right are 'Next' and 'Cancel' buttons.

Activity 2: Create a custom tab for the Lease Tracking object

1. Go to setup page ==> type Tabs in Quick Find bar ==>click on tabs ==> New (under custom object tab).
2. Select Object (Lease Tracking) ==> Select the tab style ==> Next (Add to profiles page) keep it as default ==> Next (Add to Custom App) uncheck the include tab ==> Save.
3. Make sure to append a tab to users' existing personal customizations is checked.

The screenshot shows the 'Custom Tabs' section of the Salesforce Setup. It lists 'Custom Object Tabs' with three entries: 'Leases Trackeo' (Tab Style: Broad), 'Tenant Issues' (Tab Style: Computer), and 'Tenants' (Tab Style: Apple). Below this are sections for 'Web Tabs', 'Visualforce Tabs', 'Lightning Component Tabs', and 'Lightning Page Tabs', each showing a message indicating no tabs have been defined.

The screenshot shows the 'New Custom Object Tab' wizard, Step 1: Enter the Details. It asks to choose a custom object for the new tab. The 'Object' dropdown is set to 'Leases Trackeo'. The 'Tab Style' dropdown is set to 'Broad'. There is an optional field for a 'Splash Page Custom Link' which is currently empty. A note says '(Optional) Choose a Home Page Custom Link to show as a splash page the first time your users click on this tab.' At the bottom, there is a 'Description' input field and a 'Next' button.

Activity 3: Create a custom tab for the Tenant Issue object

1. Go to setup page ==> type Tabs in Quick Find bar ==>click on tabs ==> New (under custom object tab).
2. Select Object (Tenant Issues) ==> Select the tab style ==> Next (Add to profiles page) keep it as default ==> Next (Add to Custom App) uncheck the include tab ==> Save.
3. Make sure to append a tab to users' existing personal customizations is checked.

The screenshot shows the Salesforce Setup interface with the 'Tabs' page selected. The left sidebar shows 'User Interface' and 'Tabs'. The main content area is titled 'Custom Tabs' and contains sections for 'Custom Object Tabs', 'Web Tabs', 'Visualforce Tab', 'Lightning Component Tabs', and 'Lightning Page Tabs'. Under 'Custom Object Tabs', there is a table with three rows:

Action	Label	Tab Style	Description
Edit Del	Leases Tracking	Broad	
Edit Del	Tenant Issues	Computer	
Edit Del	Tenants	Apple	

The screenshot shows the 'New Custom Object Tab' wizard, Step 1 of 3. The title is 'Step 1. Enter the Details'. It asks to choose a custom object for the new tab. A dropdown menu shows 'Object' and 'None'. Below it is an optional section for a splash page custom link, with a dropdown set to 'None'. At the bottom, there is a 'Description' field and a 'Next' button.

Milestone 4 - Create Fields and Relationships

What is Field?

Fields in Salesforce are individual data points that belong to a Salesforce object (like Account, Contact, Opportunity, or custom objects). Each field in an object is akin to a column in a spreadsheet and is used to capture specific information about a record. Salesforce offers various types of fields, including:

- Standard Fields: Predefined fields available in every Salesforce org, such as Name, Created Date, Owner, etc.
- Custom Fields: User-defined fields created to meet specific business needs, allowing customization beyond the standard fields.
- Field Types: Fields come in various types, such as Text, Number, Date, Checkbox, Picklist, Lookup, and Formula fields, allowing data to be structured appropriately.

What is Relationships?

Relationships in Salesforce define how two or more objects are connected, enabling data to be linked and managed across different records. Salesforce supports several types of relationships:

- Lookup Relationship: A loosely coupled relationship where one object can link to another, but records can exist independently. For example, a Contact record may have a lookup relationship to an Account, but deleting the Account won't delete the Contact.
- Master-Detail Relationship: A tightly coupled relationship where one object depends on another. In a master-detail relationship, the detail (or child) record is deleted if the master (or parent) record is deleted. For example, Opportunity Line Items are dependent on an Opportunity.
- Many-to-Many Relationship: This relationship requires a junction object to link two objects in a many-to-many structure. For instance, a Course object and a Student object might be related in a many-to-many fashion using a Course Enrolment junction object.
- Hierarchical Relationship: A special type of relationship available only for the User object, allowing it to relate to itself in a hierarchical structure, like a manager and employee relationship.

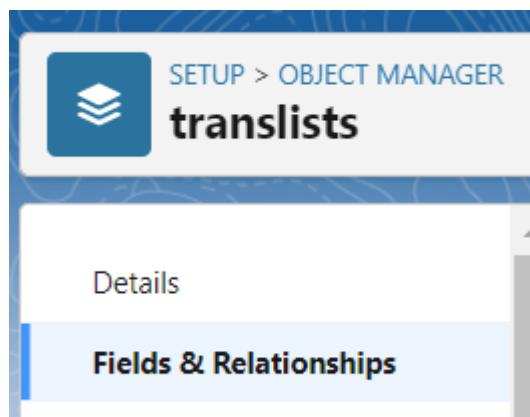
Use Case:

In Salesforce, fields and relationship's structure data for efficient tracking and management. Fields capture specific data points, like a Contact's 'Name' and 'Email', on individual records. Lookup Relationships link objects with flexible connections, such as associating Contacts with Accounts, while still keeping them independent. Master-Detail Relationships enforce dependency,

ensuring child records, like Opportunity Line Items, get deleted if the parent Opportunity is removed. Lastly, Many-to-Many Relationships use junction objects to associate objects like `Courses` and `Students`, allowing each student to enrol in multiple courses and vice versa.

Activity 1: Create Fields on the Tenant object

1. Setup and click on setup.
2. You will now be navigated to the setup page, click on object manager and search for the object “Tenant”.
3. Click on “Fields & Relationships” in the left panel.



4. Click on New and choose the data type Phone and first name: Phone Number.
5. Click next and fill in the following details in the mentioned.

A screenshot of the Salesforce setup interface for creating a new custom field. The page title is "SETUP > OBJECT MANAGER Tenant New Custom Field". The left sidebar shows "Fields & Relationships" is selected. The main form is titled "Step 2. Enter the details". It includes fields for "Field Label", "Field Name", "Description", "Help Text", "Required" (unchecked), "Auto add to custom report type" (checked), and "Default Value" (with a formula editor open). Buttons at the bottom include "Previous", "Next", and "Cancel".

6. Click Next, Next, and click on “Save and New”.

Note: Repeat the same steps to create the fields:

S No	Field Label	Data Type
1	Address	Text Area(255)
2	PAN Card	Text
3	Date of Reg	Date
4	Email	Email
5	GST No	Text
6	Registered License No	Text
7	Shop Act license No	Text
8	Status of possession	Pick List Pending Hand Over Renewal Needed Closed

Fields & Relationships
13 items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address_c	Text Area(255)		
Created By	CreatedById	Lookup(User)		
Date of Reg	Date_of_Reg_c	Date		
Email	Email_c	Email		
GST No	GST_No_c	Text(25)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		
PAN Card	PAN_Card_c	Text(10)		
Phone Number	Phone_Number_c	Phone		
Registered License No	Registered_License_No_c	Text(15)		
Shop Act license No	Shop_Act_license_No_c	Text(20)		
Status of possession	Status_of_posession_c	Picklist		
Tenant Name	Name	Text(80)		

Create Validation Rules for Tenant Object:

A) Create validation rule for Phone Number -

Note: - check if the Phone Number is valid having 10 digits if not then show an error.

1. Go to setup ==> click on Object Manager ==> type object name (Tenant) in quick find bar==>click on the object.
2. Click on the validation rule ==> click New.

Validation Rules
2 items, Sorted by Rule Name

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
Date_Validation	Date of Reg	Enter Valid Date	✓	Paranjithi Karthik M. 10/11/2024, 9:31 pm
Phone_Validation	Phone Number	Enter Valid 10 digit Phone number	✓	Paranjithi Karthik M. 10/11/2024, 8:57 pm

3. Enter the Rule name as “Phone Validation “.
4. Insert the Error Condition Formula as: -

NOT(OR(REGEX(Phone_Number__c , "^[0-9]{10}")))

The screenshot shows the Salesforce Setup interface under the Object Manager section for the 'Tenant' object. A validation rule is being created with the following details:

- Rule Name:** Phone Validation
- Description:** (empty)
- Error Condition Formula:**

```
NOT(OR(REGEX(Phone_Number__c, "^[0-9]{10}")))
```
- Functions:** A sidebar lists various functions like ABS, ACOS, ADDMONTHS, AND, ASCII, ASIN, etc.

- Enter the Error Message as “Enter Valid 10 digit Phone number”, select the Error location as Field and select the field as “Phone Number”, and click Save

The screenshot shows the 'Error Message' configuration screen. The error message is set to appear at the top of the page for the 'Phone Number' field:

- Error Message:** Enter Valid 10 digit Phone number
- Error Location:** Top of Page

B) Create a Validation rule for the Date of Reg :

Note:- check if the DateofReg is valid and is not a Date in the past.

- Go to setup ==> click on Object Manager ==> type object name(Tenant) in quick find bar==> click on the object.
- Click on the validation rule==> click New.
- Enter the Rule name as “Date Validation “.
- Insert the Error Condition Formula as: -

Formula: DateofReg__c < TODAY()

SETUP > OBJECT MANAGER

Tenant

Validation Rules
2 items, Sorted by Rule Name

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
Date_Validation	Date of Reg	Enter Valid Date	✓	Paranjithi Karthik M, 10/11/2024, 9:31 pm
Phone_Validation	Phone Number	Enter Valid 10 digit Phone number	✓	Paranjithi Karthik M, 10/11/2024, 8:57 pm

New

SETUP > OBJECT MANAGER

Tenant

Tenant Validation Rule

Define a validation rule by specifying an error condition and a corresponding error message. The error condition is written as a Boolean formula expression that returns true or false. When the formula expression returns true, the save will be aborted and the error message will be displayed. The user can correct the error and try again.

Validation Rule Edit

Rule Name: Date_Validation

Description:

Error Condition Formula

Example: Discount_Percent <= 30% More Examples...
Display an error if Discount is more than 30%
If this formula expression is true, display the text defined in the Error Message area

Functions

- ABS
- ACOS
- ACOSMONTHS
- AND
- ASCII
- ASIN

Insert Selected Function

ABS(number)
Returns the absolute value of a number, a number without its sign

Check Syntax

5. Enter the Error Message as “Enter Valid Date”, select the Error location as Field and select the field as “DateOfReg”, and click Save.

Error Message

Example: Discount percent cannot exceed 30%

This message will appear when Error Condition formula is true

Error Message Enter Valid Date

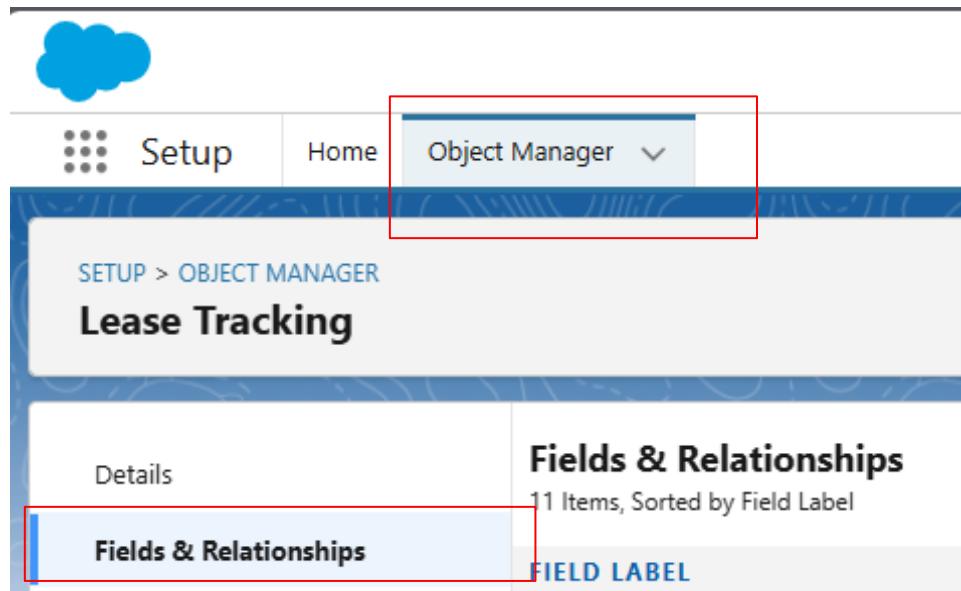
This error message can either appear at the top of the page or below a specific field on the page

Error Location Top of Page Field Date of Reg

Save **Save & New** **Cancel**

Activity 2: Create fields on the Lease Tracking Object

1. Setup and click on setup.
2. You will now be navigated to the setup page, click on object manager and search for the object “Lease Tracking”.
3. Click on “Fields & Relationships” in the left panel.



4. Click on New and choose the data type Date and first name: Date of Possession.
5. Click next and fill in the following details in the mentioned.

The screenshot shows the 'New Custom Field' creation wizard in Step 2. The left sidebar lists various setup categories like Page Layouts, Lightning Record Pages, etc. The main form is titled 'Lease Tracking New Custom Field'. It has a step navigation bar at the top: 'Step 2. Enter the details' (Step 2 of 4). The form fields include:

- Field Label: A text input field.
- Field Name: A text input field.
- Description: A text area for description.
- Help Text: A text area for help text.
- Required: A checkbox labeled 'Always require a value in this field in order to save a record'.
- Auto add to custom report type: A checkbox labeled 'Add this field to existing custom report types that contain this entity'.
- Default Value: A 'Show Formula Editor' button with a formula editor input field containing: `Use formula syntax. Enter text and positive or API names in double quotes ('One'). Include numbers without quotes (10). Use underscores as decimal. (10), and express date calculations in the standard format: Today + 7. To reference a field from a Custom Metadata type record use: $CustomMetadata_Type__mdt RecordAPIName__c`.

At the bottom right are 'Previous', 'Next', and 'Cancel' buttons.

6. Click Next, Next, and click on “Save and New”.

NOTE: - Fields in lease Tracking objects are as follows below data types:

S No	Field Label	Data Type
1	Related Tenant	Master-Detail Relationship (Related to - Tenant)
2	Date of Possession	Date
3	End Date of Possession	Date
4	Total Year of Contract	Number
5	Total rent(Yearly)	Number
6	Amount Paid	Number
7	Amount to be paid	Formula field (Total Rent - Amount Paid) Return Data Type- Number

Fields & Relationships					
	FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Page Layouts	Amount Paid	Amount_Paid__c	Number(18, 0)		
Lightning Record Pages	Amount to be paid	Amount_to_be_paid__c	Formula (Number)		
Buttons, Links, and Actions	Created By	CreatedById	Lookup(User)		
Compact Layouts	Date of Possession	Date_of_Possession__c	Date		
Field Sets	Email id	Email_Id__c	Email		
Object Limits	End Date of Possession	End_Date_of_Possession__c	Date		
Record Types	Last Modified By	LastModifiedById	Lookup(User)		
Related Lookup Filters	Lease Tracking No	Name	Auto Number		
Search Layouts	Related Tenant	Related_Tenant__c	Master-Detail(Tenant)		
List View Button Layout	Total rent(Yearly)	Total_rent_Yearly__c	Number(18, 0)		
Restriction Rules	Total Year of Contract	Total_Year_of_Contract__c	Number(18, 0)		
Scoping Rules					
Object Access					
Triggers					
Flow Triggers					
Validation Rules					

Create Validation Rule For Lease Tracking Object :

A) Create a Validation rule on the Date of Possession-

Note:- check if the Date of Possession is 60 days from today or not if not then show an error.

1. Go to setup ==> click on Object Manager ==> type object name(Lease tracking) in quick find bar==>click on the object.
2. Click on the validation rule ==> click New.

Validation Rules					
	RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
	Possession_Validation	Date of Possession	Enter a date after 60 days	✓	Paranjithi Karthik M, 10/11/2024, 9:52 pm

3. Enter the Rule name as “Possession Validation“.

- Insert the Error Condition Formula as: -

Formula: Date of Possession < TODAY() + 60

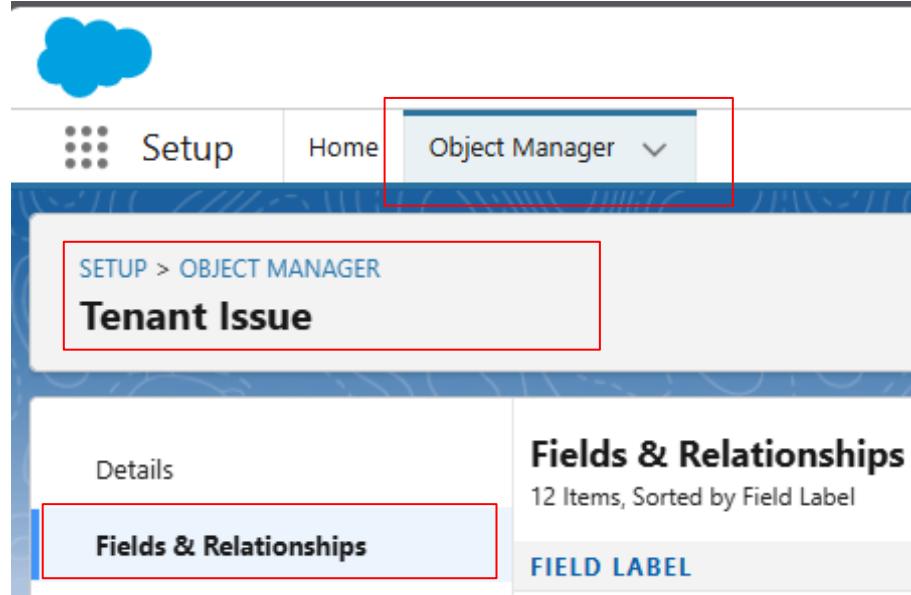
The screenshot shows the 'Lease Tracking Validation Rule' configuration screen. The 'Rule Name' is 'Possession_Validation'. The 'Error Condition Formula' is set to 'Date_of_Possession_c < TODAY() + 60'. A dropdown menu for functions is open, showing options like ABS, ACOS, ADDMONTHS, AND, ASCII, ASIN, etc.

- Enter the Error Message as “Enter a date after 60 days”, select the Error location as Field select the field as “DateofPossession”, and click Save.

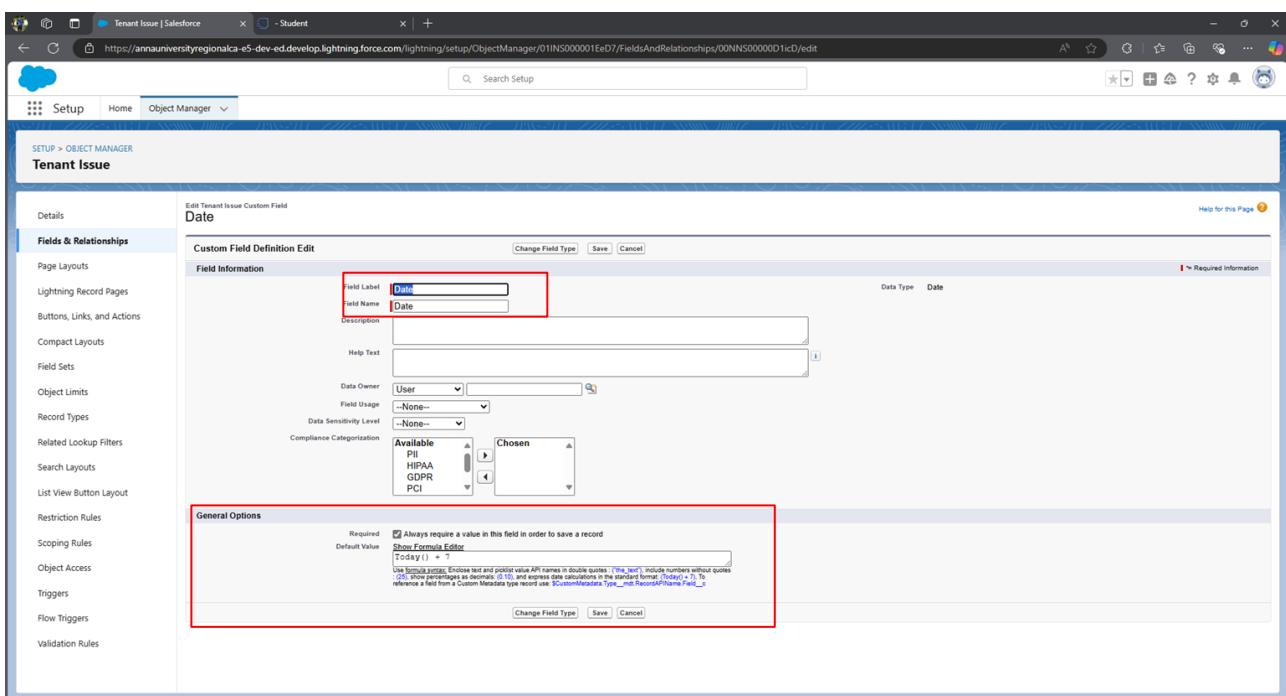
The screenshot shows the 'Error Message' configuration screen. The 'Error Message' field contains 'Enter a date after 60 days'. The 'Error Location' is set to 'Field Date of Possession'. The 'Save' button is visible at the bottom.

Activity 3: Create fields on Tenant Issues

- Setup and click on setup.
- You will now be navigated to the setup page, click on object manager and search for the object “lease Tracking”.
- Click on “Fields & Relationships” in the left panel.



4. Click on New and choose the data type Date and first name: Date of Possession.
5. Click next and fill in the following details in the mentioned.



6. Click Next, Next, and click on “Save and New”.

NOTE: Fields in lease Tenant Issues are as follows below data types-

S No	Field Label	Data Type
1	Related tenant	Master-Detail Relationship (Related to - Tenant)
2	Issue Related to	Multi Select Picklist 1) ELECTRICITY 2) INFRASTRUCTURE 3) PLUMBING 4) RENT 5) OTHER
3	Subject	Text Are (long)
4	Phone Number	Number
5	Status	Pick List 1. Not contacted 2. Open 3. In progress 4. Working 5. closed
6	Priority	Picklist 1. Low 2. Medium 3. High
7	Origin	Picklist 1. Phone 2. Mail 3. Web
8	Email id	Email
9	Date	Date (Default Value - Today())

Tenant Issue | Salesforce - Student

https://annauniversityregionalca-e5-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01INS000001EeD7/FieldsAndRelationships/view

Cloud Setup Home Object Manager

Search Setup

SETUP > OBJECT MANAGER

Tenant Issue

Fields & Relationships

12 items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	Lookup(User)		
Date	Date_c	Date		
Email id	Email_id_c	Email		
Issue Related to	Issue_Related_to_c	Picklist (Multi-Select)		
Issues	Name	Auto Number		
Last Modified By	LastModifiedBy	Lookup(User)		
Origin	Origin_c	Picklist		
Phone Number	Phone_Number_c	Number(18, 0)		
Priority	Priority_c	Picklist		
Related tenant	Related_tenant_c	Master-Detail(Tenant)		
Status	Status_c	Picklist		
Subject	Subject_c	Long Text Area(32768)		

Quick Find New Deleted Fields Field Dependencies Set History Tracking

Milestone 5 - Create the Lightning App

An app is a collection of items that work together to serve a particular function. In Lightning Experience, Lightning apps give users access to sets of objects, tabs, and other items all in one convenient bundle in the navigation bar.

Lightning apps let you brand your apps with a custom colour and logo. You can even include a utility bar and Lightning page tabs in your Lightning app. Members of your org can work more efficiently by easily switching between apps.

There are two types of Salesforce Applications:

- Standard Apps
- Custom Apps

Standard Apps:

Standard apps come with Salesforce as the default for every occurrence. Community, Call Center, Content, Sales, Marketing, Salesforce Chatter, Site.com, and App Launcher are included in these apps. The description, logo, and label of a standard app cannot be altered.

Custom Apps:

Custom apps are created according to the needs of a company. They can be made by putting custom and standard tabs together. Logos for custom apps can be changed.

Use Case:

Well done you have reached close to your organizational requirement by creating the objects to store the organization's data. Making a database for an organization is just not enough to reach out the requirements, the task is how the users at the organisation can access the objects you have created for them. As an Admin for The Smart Bridge organization, it's your duty to make sure every user of the organization is able to access the data modelling structure.

Activity 1: Steps to create a custom app in Salesforce

1. Go to setup, by clicking the gear icon present in the top right corner.
2. Navigate to the Home bar and in the quick find box, search for App.
3. Click on APP MANAGER.
4. You can notice the screen like this. Now click on New Lightning app You will find like this below.

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="SmartMall"/>	Image <input type="file"/>
*Developer Name <input type="text" value="SmartMall"/>	Primary Color Hex Value <input type="text" value="#0070D2"/>
Description <input type="text" value="SmartMall"/>	Org Theme Options <input type="checkbox"/> Use the app's image and color instead of the org's custom theme

App Launcher Preview

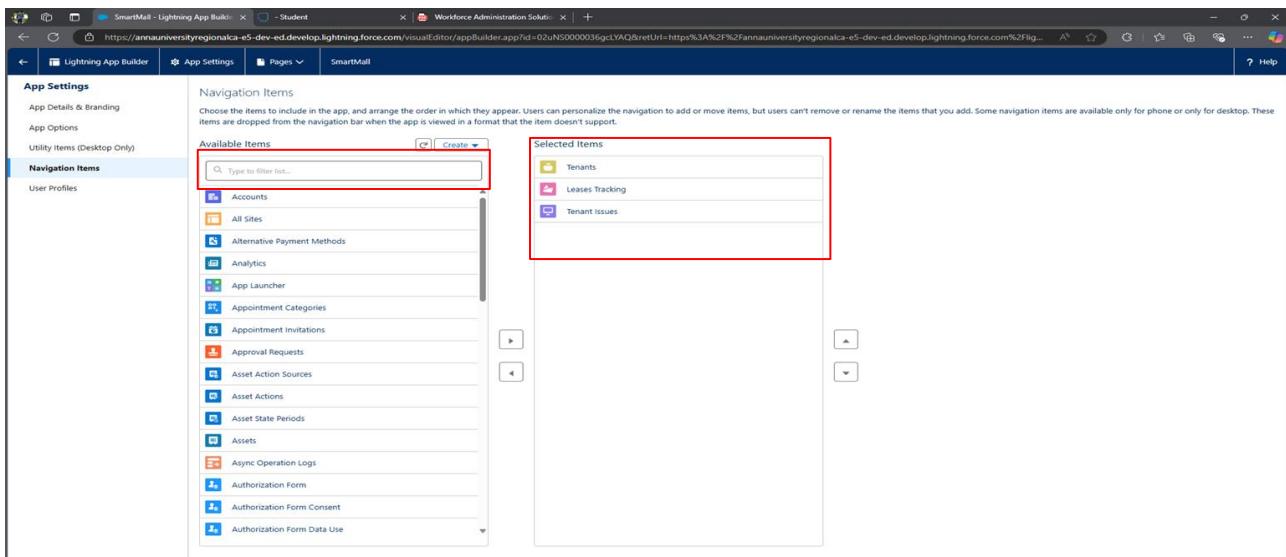


Next

5. Enter the App name(Here we entered ‘SmartMall’),the developer name gets automatically populated. If an image is required, you can browse the image and upload it.

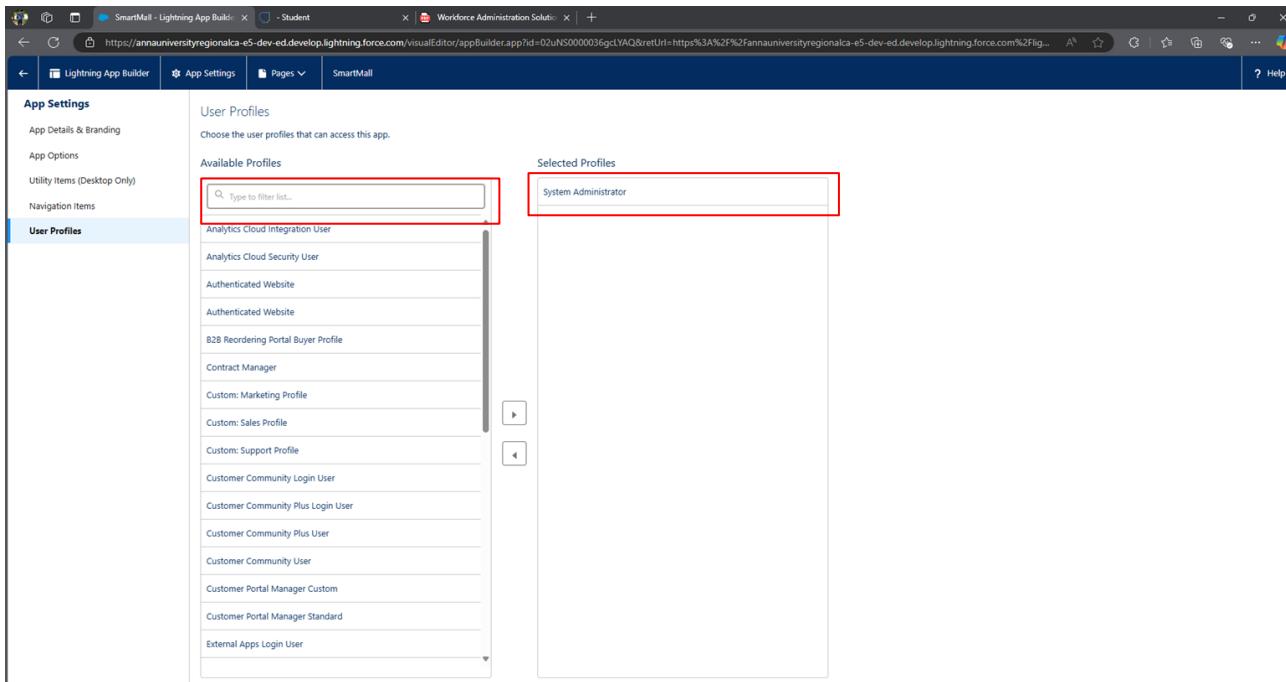
6. Click Next, Next and you can see a Navigation Items window like this:

7. In the filter list, enter Tenant, Lease Tracking, and Tenant issues, and move them to the Selected items from Available items.



8. Click on Next, and you will see User Profiles. This option is used when we want only certain profiles to have access to them.

9. Enter System Administrator in the filter box and add the system Administrator to the selected profile list.



10. Click on Save and Finish.

11. Now navigate to the App launcher and search for SmartMall and you can find the SmartMall app.

Milestone 6 - Record Insertion

Creating records in Salesforce is a fundamental and essential activity that serves multiple purposes, contributing to the effective management of data, streamlined processes, and the overall success of an organization.

Record insertion in Salesforce is the process of creating new records (like leads, accounts, contacts, opportunities, etc.) in the Salesforce database. There are various ways to insert records in Salesforce, depending on your needs and access:

1. **Manual Entry:** You can create records manually by navigating to a specific object (like Accounts or Contacts) and filling out the required fields.
2. **Data Import Wizard:** This tool is useful for bulk data entry. It allows you to import data from a CSV file into Salesforce and automatically map fields. It's ideal for simple inserts and is accessible from the Salesforce setup menu.
3. **Data Loader:** This is a more powerful tool for bulk data operations, suitable for inserting, updating, and deleting large data sets (up to millions of records). It also supports scheduling, so you can set up regular batch imports.
4. **APEX DML (Data Manipulation Language):** If you are a developer, you can use APEX code to insert records programmatically. This is helpful when building custom logic or automation around the record creation process.
5. **API Integration:** Salesforce provides a REST and SOAP API that allows third-party applications to insert records into Salesforce. This is often used for integrating Salesforce with other systems, enabling real-time data flow between platforms.

Use Case:

In Salesforce, record insertion is essential for maintaining up-to-date customer and operational data. For small-scale data entry, **manual entry** is straightforward and effective, allowing users to create individual records directly within Salesforce. For bulk data import, the **Data Import Wizard** offers a simple, guided experience suitable for moderate volumes of data, enabling users to upload CSV files and map fields automatically. When handling large-scale data imports or regularly scheduled updates, the **Data Loader** is a powerful tool that supports bulk data manipulation, including inserts, updates, and deletes. Developers needing custom logic for record insertion can utilize **APEX DML** to programmatically create records, embedding logic to automate processes. For integration with external systems, **API Integration** provides REST and SOAP endpoints, allowing

third-party applications to insert records into Salesforce in real-time, making it ideal for synchronizing data between platforms.

Activity 1: Inserting Records in Tenant Object

1. Click on the App Launcher and search Tenant Object then click New in the right corner to create a record.

The screenshot shows the Salesforce App Launcher interface. A red box highlights the 'Setup' icon in the top left. Below it, under 'All Apps', the 'SmartMall' app is highlighted with a red box. The app description reads: '(Lightning Experience) Lets support agents work with multiple records across customer service channels on one screen'. Other apps listed include Service, Marketing CRM Classic, Community, Salesforce Chatter, Content, Sales Console, Sales, Lightning Usage App, Digital Experiences, Bolt Solutions, and Salesforce Scheduler Setup.

The screenshot shows the 'Tenants' list page for the SmartMall app. A red box highlights the 'Tenants' tab in the top navigation bar. Another red box highlights the 'New' button in the top right corner of the list area. The list displays 10 items, each with a checkbox and a tenant name: Dinesh, Suresh, Ramesh, Tamizselvan, Lakshman, Ganesh, Siva, Ram, Sita, and Karthik.

2. Fill every field with valid data, especially the fields on which you have created a validation rule.

Search... X

Edit Karthik

* = Required Information

* Tenant Name	Karthik	Owner	Paranjithi Karthik M
* Phone Number	9629921185		
* Address	Tirupur		
* PAN Card	GTY5347M		
* Date of Reg	10/11/2024		
* Email	karthik2003@gmail.com		
* GST No	GSS1234		
* Registered License No	9500693		
* Shop Act license No	45624678763		
* Status of possession	Pending Hand Overed Renewal Needed Closed		
<input type="button" value="Cancel"/> <input type="button" value="Save & New"/> <input type="button" value="Save"/>			
Created By		Last Modified by	

3. If you Enter a Phone Number of more or less than 10 digits it will show an error.
4. Similarly, if you enter DateofReg a Past date it will show an error.
5. After creating a record the page will look like this

The screenshot shows the Salesforce Lightning interface for a tenant record. The top navigation bar includes links for SmartMall, Tenants, Leases Tracking, Tenant Issues, Reports, and Dashboards. The main header displays "Tenant Karthik". Below the header, there are two tabs: "Related" and "Details". The "Details" tab is selected, showing the following fields and their values:

- Tenant Name: Karthik
- Phone Number: 9629921185
- Address: Tirupur
- PAN Card: GTY5347M
- Date of Reg: 10/11/2024
- Email: karthik2003@gmail.com
- GST No: GSS1234
- Registered License No: 9500693
- Shop Act license No: 45624678763
- Status of possession: Pending Hand Overed Renewal Needed Closed

On the right side of the details section, there is an "Owner" field showing "Paranjithi Karthik M". At the bottom of the page, there are "Created By" and "Last Modified by" fields, both showing "Paranjithi Karthik M" with the timestamp "10/11/2024, 11:35 pm".

[Note]: Create at least 10 records in the tenant object.

The screenshot shows the Salesforce Lightning interface for the Tenant object. The top navigation bar includes tabs for Recently Viewed, Tenants, Sales, Student, and Workforce Administration Solutions. The main content area displays a list titled "Tenants Recently Viewed" with 10 items. A red box highlights the first item, "Karthik". The list contains the following data:

Index	Tenant Name
1	Karthik
2	Dinesh
3	Suresh
4	Ramesh
5	Tamilselvan
6	Lakshman
7	Ganesh
8	Siva
9	Ram
10	Sita

Activity 2: Inserting Records in Lease Tracking Object

1. Click on the App Launcher and search Lease Tracking Object then click New in the right corner to create a record.

The screenshot shows the Salesforce Lightning interface for the Lease Tracking object. The top navigation bar includes tabs for Recently Viewed, Leases Tracking, Sales, Student, and Workforce Administration Solutions. The main content area displays a list titled "Leases Tracking Recently Viewed" with 10 items. A red box highlights the first item, "TT - 000010". The list contains the following data:

Index	Lease Tracking No
1	TT - 000010
2	TT - 000006
3	TT - 000008
4	TT - 000009
5	TT - 000007
6	TT - 000005
7	TT - 000004
8	TT - 000003
9	TT - 000002
10	TT - 000001

2. Fill every field with valid data, especially the fields on which you have created a validation rule, and Give each tenant related to each lease tracking.

Search...

Edit TT - 000002

* = Required Information

Lease Tracking No

TT - 000002

Related Tenant

Suresh

* Date of Possession

31/01/2025



Format: 31/12/2024

* End Date of Possession

31/01/2026



* Total Year of Contract

1

* Total rent(Yearly)

3,00,000

* Amount Paid

75,000

Amount to be paid

2,25,000.00

This field is calculated upon save

* Email id

Created By



Paranjothi Karthik M, 11/11/2024, 11:23 am

Last Modified By



Paranjothi Karthik M, 11/11/2024, 11:23 am

 Cancel Save & New Save

3. If you Enter a Date before the next 60 days it will show an error.
4. Similarly, if you enter Total rent and Amount Paid the Amount to be paid formula field will be added directly.
5. After creating a record the page will look like this

[Note]: Create at least 10 records in the Lease Tracking object.

Activity 3: Inserting Records in Tenants Issues object

1. Click on the App Launcher and search Tenant Issues Object then click New in the right corner to create a record.
2. Fill every field with valid data.
3. Give each tenant related to each issue.
4. After creating a record the page will look like this

Recently Viewed | Tenant Issues | Student | Workforce Administration Solution

Tenant Issues

SmartMall Tenants Leases Tracking Tenant Issues Reports Dashboards

Recently Viewed

10 items • Updated a few seconds ago

Issues

1	□	9
2	□	8
3	□	7
4	□	6
5	□	5
6	□	4
7	□	3
8	□	2
9	□	1
10	□	0

Search...

Edit 0

* = Required Information

Issues
0

Related tenant
Dinesh

*Issue Related to

Available Chosen

1) ELECTRICITY

2) INFRASTRUCTURE

3) PLUMBING

4) RENT

5) OTHER

Subject
regarding power cut during the peak time

*Phone Number
9,38,66,83,891

*Status
Open

*Priority
2. Medium

*Origin
2. Mail

Email id
t1@gmail.com

Cancel Save & New Save

Recently Viewed | Tenant Issues | Student | Workforce Administration Solution

Tenant Issues

SmartMall Tenants Leases Tracking Tenant Issues Reports Dashboards

Recently Viewed

10 items • Updated a few seconds ago

Issues

1	□	0
2	□	9
3	□	8
4	□	7
5	□	6
6	□	5
7	□	4
8	□	3
9	□	2
10	□	1

Note: Create at least 10 records in the tenant issues object.

Milestone 7 - Create Flows

In Salesforce, a **Flow** is an automation tool that allows users to build complex workflows and processes using a visual interface. It enables businesses to automate repetitive tasks, streamline workflows, and provide guided experiences for users. Flows can be used to collect data, update records, send emails, post to Chatter, and even make decisions based on conditions.

Salesforce provides different types of flows:

1. **Screen Flow:** Used to create interactive, guided experiences with screens, prompts, and forms, allowing users to enter data or make selections. It's ideal for step-by-step processes like onboarding or guided troubleshooting.
2. **Record-Triggered Flow:** Automatically runs when a record is created, updated, or deleted. It's similar to workflow rules and triggers and is useful for tasks like updating related records or sending alerts when certain conditions are met.
3. **Scheduled Flow:** Runs at specified times or intervals, allowing for batch processing of records. This can be useful for tasks like sending regular notifications or updating data at set intervals.
4. **Platform Event-Triggered Flow:** Activated by specific events published to Salesforce's platform event system, allowing flows to respond to external events or changes in real time.
5. **Auto launched Flow:** This doesn't have a user interface but can be triggered by other automation tools, like Process Builder, or even through Apex code. This is typically used for backend processing.

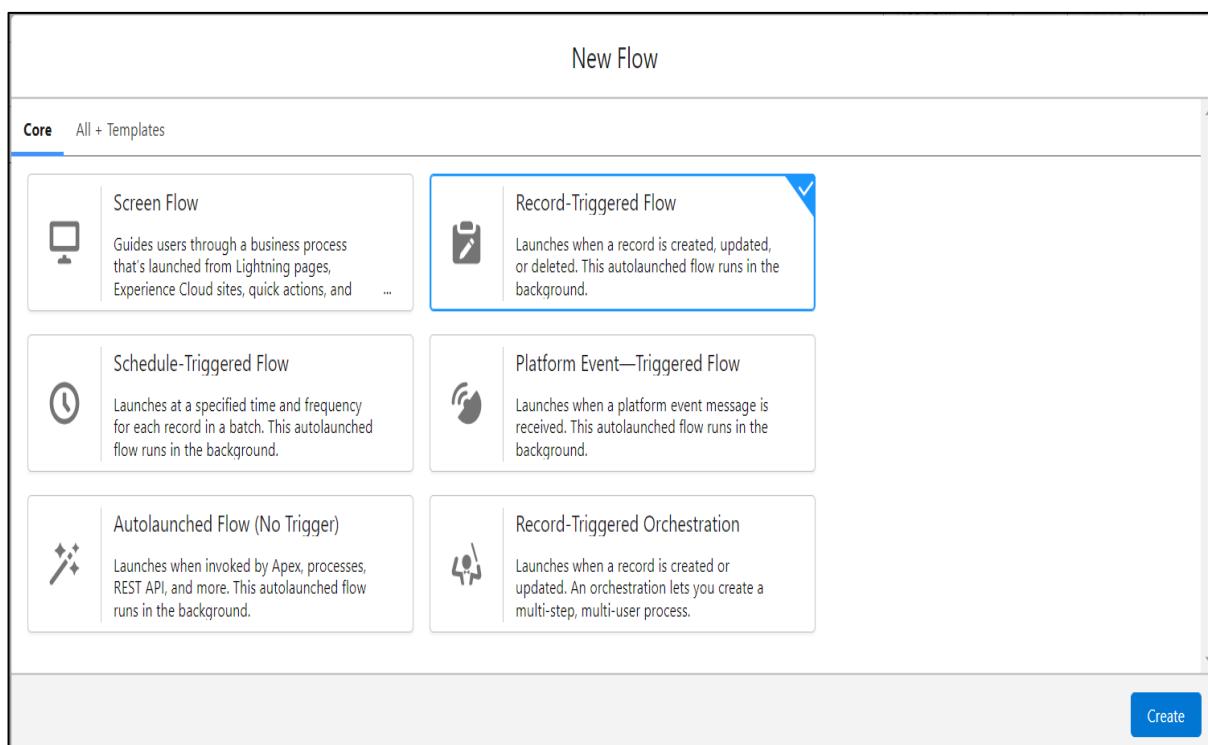
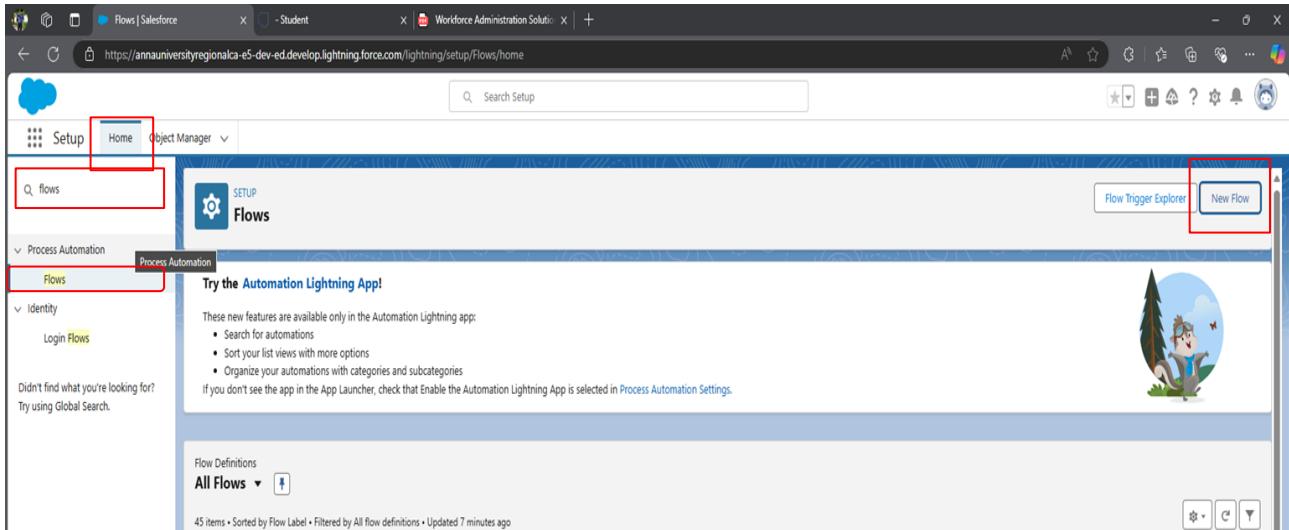
Use Case:

Imagine a company using Salesforce Flow to streamline its customer support onboarding process. With a Screen Flow, new agents can follow a guided series of prompts to familiarize themselves with support protocols, while Record-Triggered Flows automatically log and update customer records based on ticket status changes, ensuring all team members have the latest information. To further enhance efficiency, a Scheduled Flow could run nightly to batch-update records and send reminder emails to agents for unresolved cases, while a Platform Event-Triggered Flow initiates follow-up actions in real-time whenever a customer escalates an issue, ensuring that urgent cases receive immediate attention without manual intervention.

Activity 1: Create a Record Triggered flow on tenant Object

Whenever a tenant record is created and the GST No field in tenant Object is empty a mail should be sent to the tenant requesting the GST No.

1. To create a flow click on setup==> Flow ==> Click on New Flow==> Select Record Triggered Flow.



2. In Trigger the flow when Select A record is created
3. Select Condition required All Conditions are met (AND)
4. Select Field - GST_No__c, Operator - Is Null, Value - True.
5. Optimize the Flow for: Actions and Related Records

6. Add Element and choose ACTION in the search bar Search Send Email.
7. Label Name - Send email for GST no, Description - This email is to alert the tenant that he or she has not submitted the GST NO yet.

8. Include Body And Create a Resource - Text Template As below-
9. Include Recipient ID and from the profile select tenant Email ID.

New Resource

*Resource Type

Text Template

*API Name

body

Description

*Body ⓘ

Resource Picker

Insert a resource...

hello {!\$Record.Name}, we kindly request you to submit your GST details with us as soon as possible

Salesforce Sans 12

10. Include Subject and enter - Regarding your GST Details

Aa Recipient ID ⓘ Included

Aa Related Record ID Not Included

Rich-Text-Formatted Body Not Included

Aa Sender Email Address Not Included

Aa Sender Type Not Included

Aa Subject ⓘ Included

Use Line Breaks Not Included

11. Click on Save and Name the Flow as Email Flow for the tenant and Save.

12. Activate the flow.

Save the flow

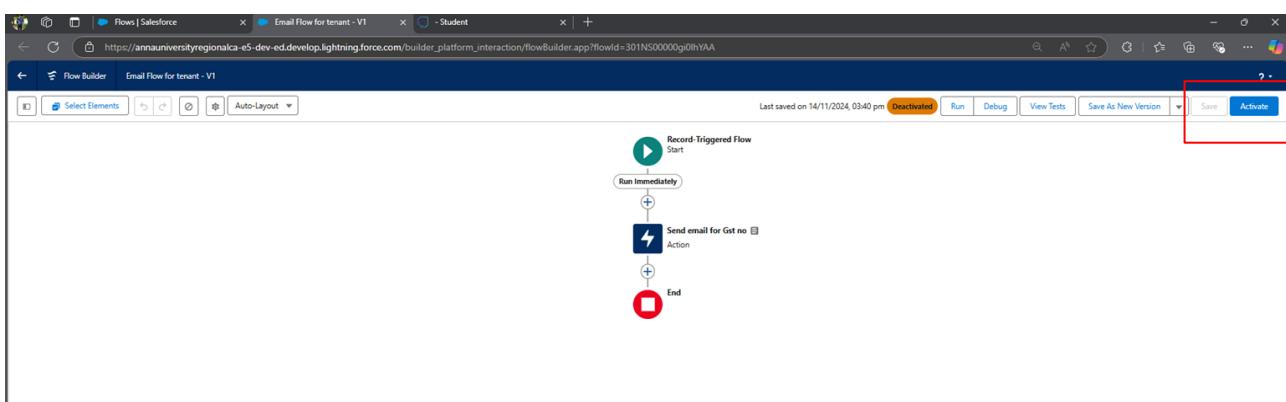
*Flow Label
email flow

*Flow API Name
email_flow

Description

Show Advanced

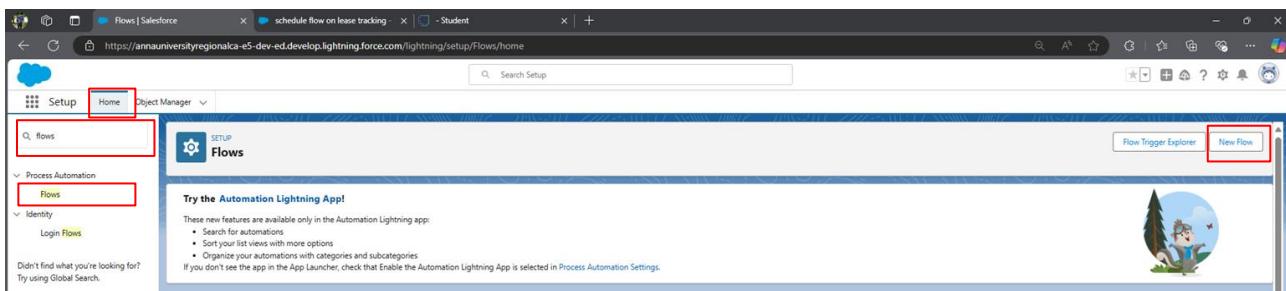
Cancel Save

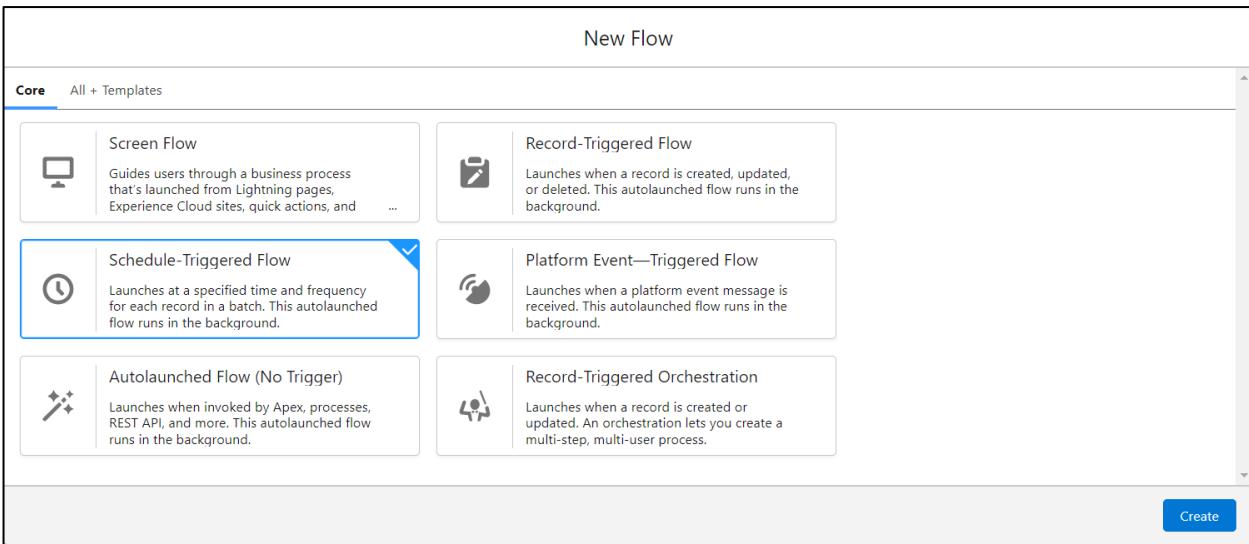


Activity 2: Create a Schedule Flow on Lease Management Object

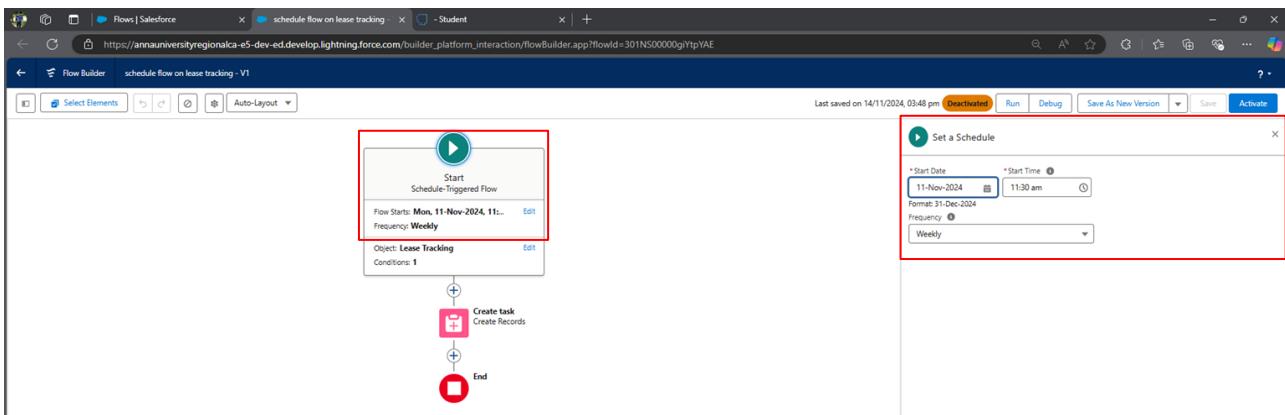
If the End Date is within the Next 1 year create a task to the Lease tracking weekly on every Monday.

1. To create a flow click on setup ==> Flow ==> Click on New Flow==> Select Schedule-Triggered Flow





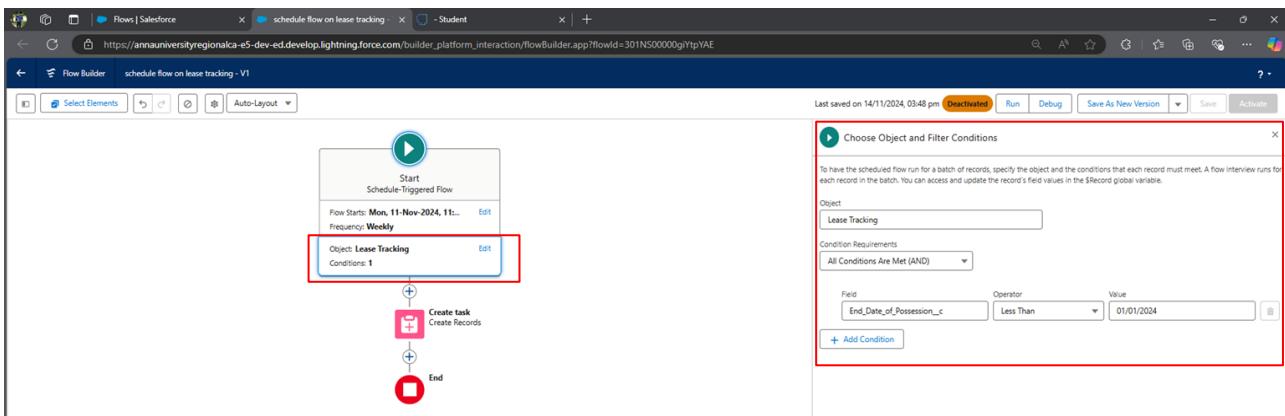
2. Set the Schedule Date - Any Monday, Time - 11.30, Frequency - Weekly.



3. Choose Object as Lease Tracking, Condition Requirements - All Conditions are met (AND)

4. Enter field as End_Date_of_Possession__c

5. Operator - Less than & Value - 1/1/2024



6. Create a Record and give label as Create task

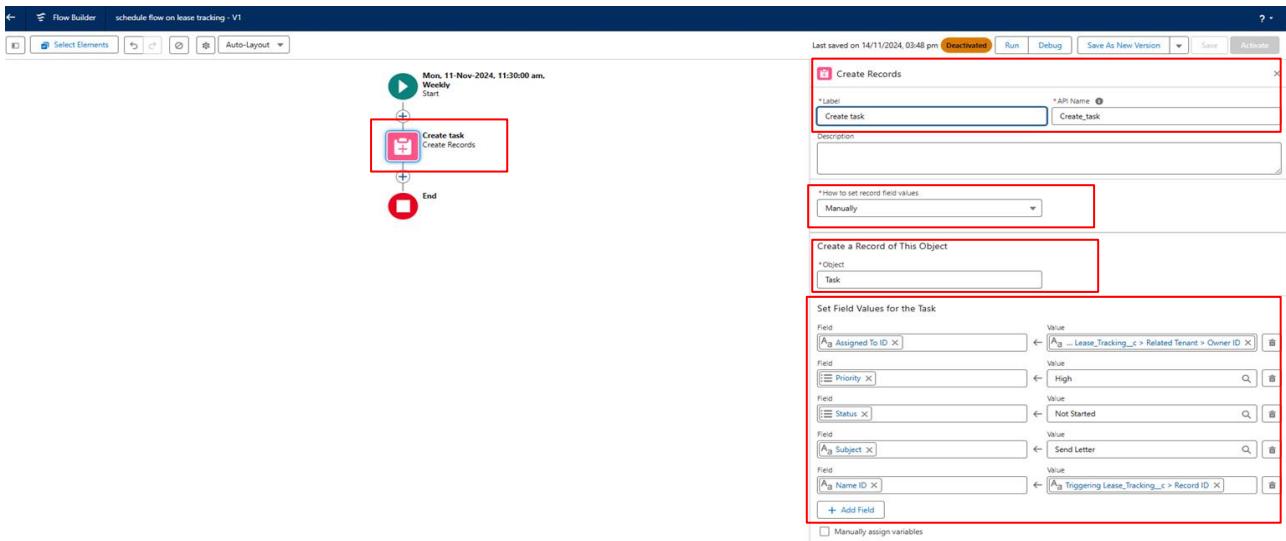
7. How many Records to create - one

8. How to Set the Record Fields - Use Separate Values

9. object - Task

10. Select the field and map them as below: -

11. Subject of the Task - your possession is going to end soon. Please Contact with Manager to renew your Possession or to End the contract.



12. Save the flow and label it as ‘schedule flow on lease tracking’

13. Activate the flow.

Save the flow

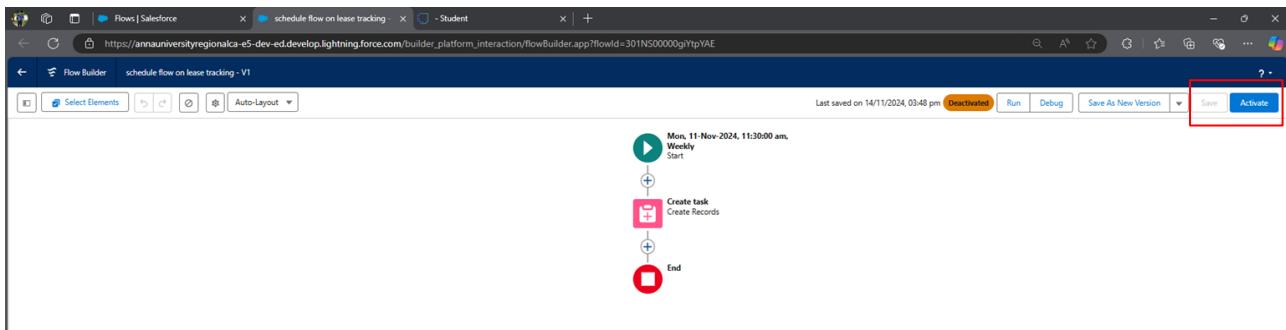
* Flow Label: schedule flow on lease tracking

* Flow API Name: schedule_flow_on_lease_tracking

Description:

Show Advanced

Cancel Save



Milestone 8 - Apex Triggers

Apex Triggers

A trigger is a set of Apex code that runs before or after DML (Data Manipulation Language) events.

A DML event could be a variety of data processing tasks that include the standard insert, update, and delete commands.

With Apex triggers, you can automate tasks that would otherwise be nearly impossible to accomplish using only the Salesforce user interface. Triggers enable you to create custom scripts that you can implement according to your needs, and the only limitation is your coding skills.

There are two Salesforce Apex trigger types:

Before triggers. These are helpful in cases that require a validation process before accepting a change. They run before any database changes. After triggers. These are helpful in cases where you need to modify your database records and when the necessary value is stored in other records. They run after any database changes. Both types will help you perform custom tasks and manage records effectively. They can help you perform bulk actions as they can handle several records simultaneously.

How to create a new trigger:

1. While still in the trailhead account, navigate to the gear icon in the top right corner.
2. Click on the developer console and you will be navigated to a new console window.
3. Click on the File menu in the toolbar, and click on the new Trigger.

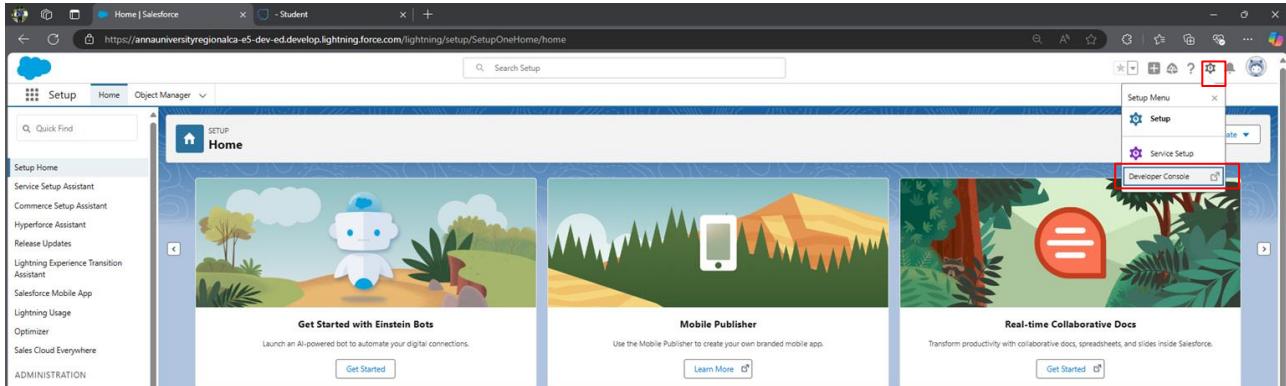
Enter the trigger name and the object to be triggered.

Use Case:

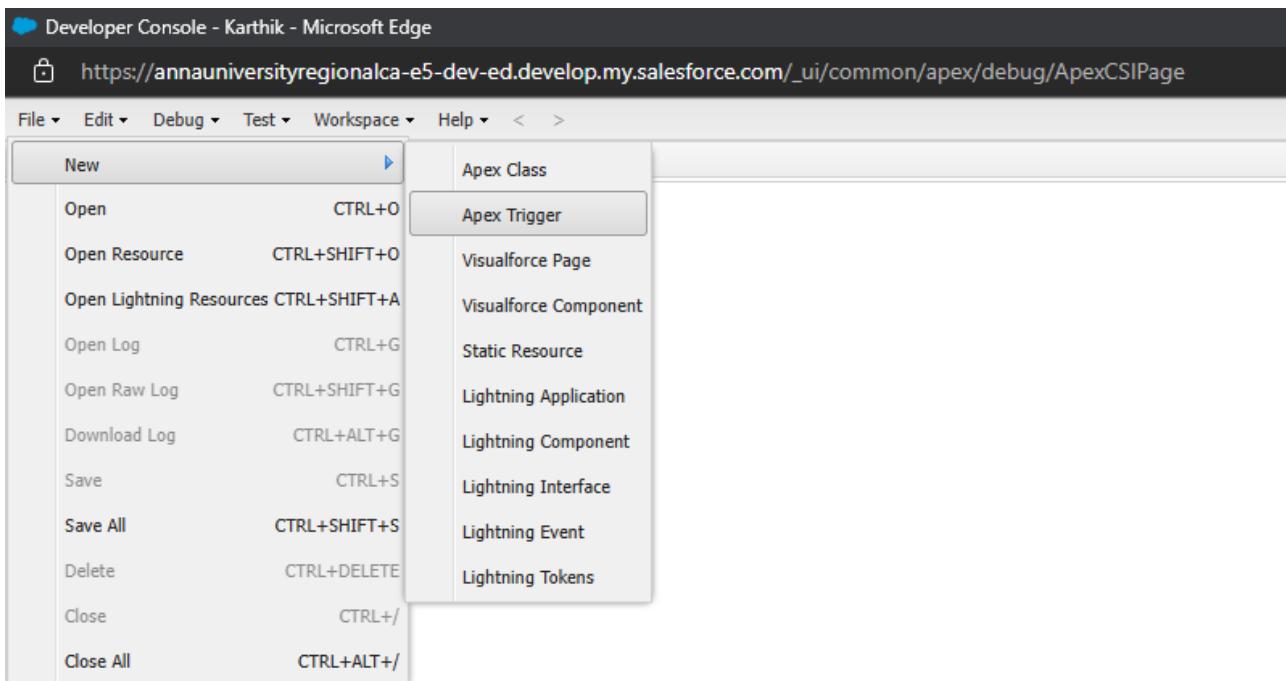
In Salesforce, Apex Triggers are powerful tools used to perform custom actions before or after specific database events, such as insertions, updates, or deletions on Salesforce objects. For instance, they can enforce business logic by automatically verifying data integrity and ensuring that no incomplete or incorrect records are saved in the system. Triggers can also automate processes such as creating related records; for example, when an Opportunity is created, a Trigger could automatically create related Task records, reminding team members to follow up. Additionally, Triggers help in maintaining data consistency across objects, such as updating parent Account information whenever a change is made to one of its child Contacts. This automation significantly enhances operational efficiency, reducing the need for manual intervention and helping teams to focus on strategic tasks.

Activity 1: Write an Apex Trigger to send an email if the tenant has not paid 50 Percent of Total Rent.

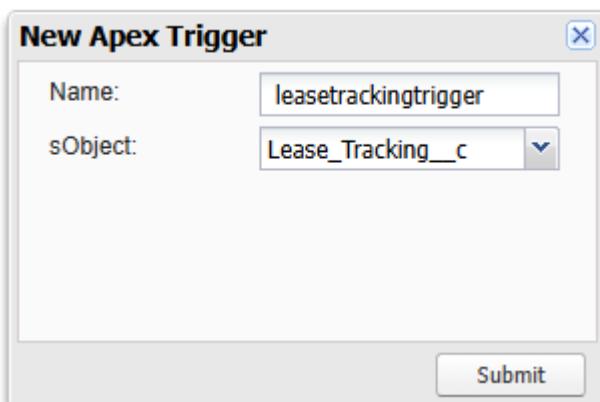
1. Click on the gear icon and click on the developer console.



2. Click on the file and select New Apex Trigger



3. Name- leasetrackingtrigger, Object —> Lease_Tracking__c



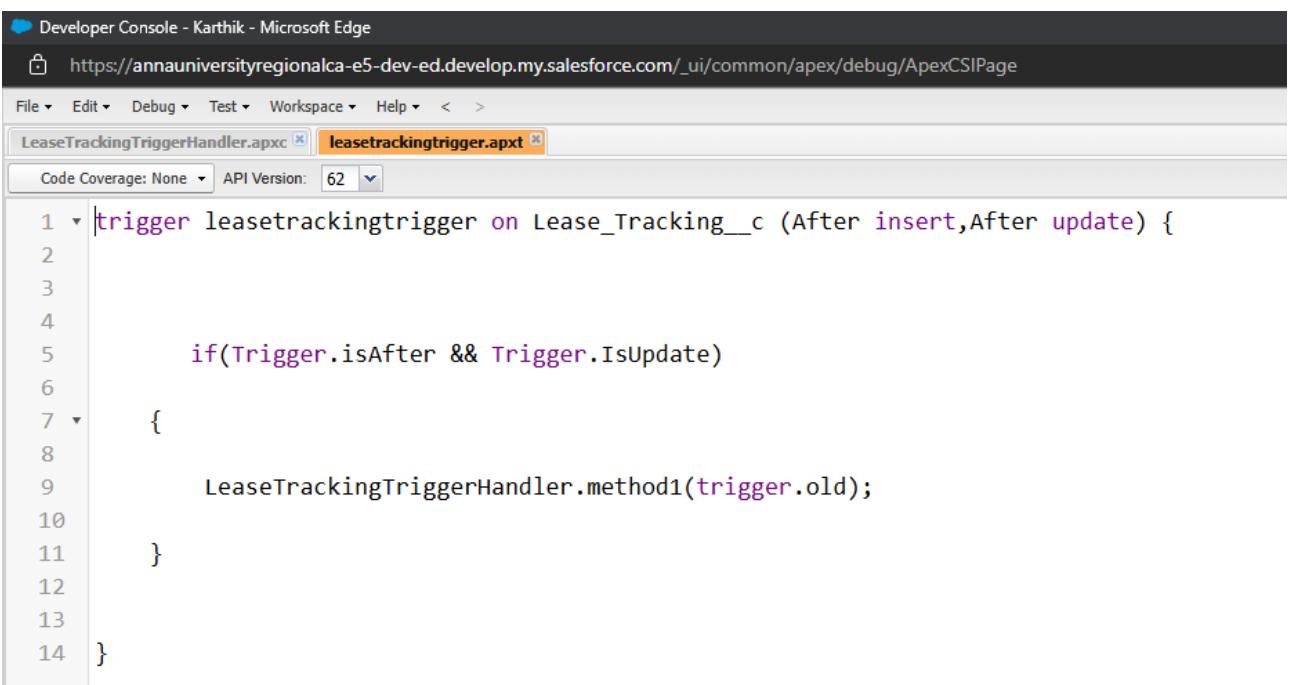
4. Use Event - After Insert and After Update and Use Trigger Context Variables as IsAfter and IsUpdate.

Trigger: -

CODE SNIPPET :

trigger leasetrackingtrigger on Lease_Tracking__c (After insert, After update)

```
{
    if(Trigger.isAfter && Trigger.IsUpdate)
    {
        LeaseTrackingTriggerHandler.method1(trigger.old);
    }
}
```



The screenshot shows the Salesforce Developer Console in Microsoft Edge. The URL is https://annauniversityregionalca-e5-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. The tab bar shows 'LeaseTrackingTriggerHandler.apxc' and 'leasetrackingtrigger.apxt'. The code coverage dropdown is set to 'None' and the API version is 62. The code itself is identical to the one above, with line numbers 1 through 14.

```

1 trigger leasetrackingtrigger on Lease_Tracking__c (After insert,After update) {
2
3
4
5     if(Trigger.isAfter && Trigger.IsUpdate)
6
7     {
8         LeaseTrackingTriggerHandler.method1(trigger.old);
9     }
10
11
12
13
14 }
```

Trigger Handler: -

1. Create an apex class and Name it LeaseTrackingTriggerHandler

CODE SNIPPET: -

```
public class LeaseTrackingTriggerHandler
{
    public static void method1(List<Lease_Tracking__c> lt1)
    {
        for(Lease_Tracking__c lt2: lt1 )
        {
            if(lt2.Amount_to_be_paid__c > (lt2.Total_rent_Yearly__c)/2)
            {
```

```

Messaging.SingleEmailMessage M = New Messaging.SingleEmailMessage();
List<String> ToADD = New List<String>{lt2.Email_id__c}
    M.setToAddresses(ToADD);
    M.setSubject('Regarding the Pending Rent');
    M.setPlainTextBody('Hello, This is an Reminder for you to complete your due rent by
the end this month, your due rent thatneeds to be paid is' +lt2.Amount_to_be_paid__c);
    List<Messaging.Email> AB = New List<Messaging.Email>{ };
    AB.add(M);
    Messaging.sendEmail(AB);
}

}

}

}

```

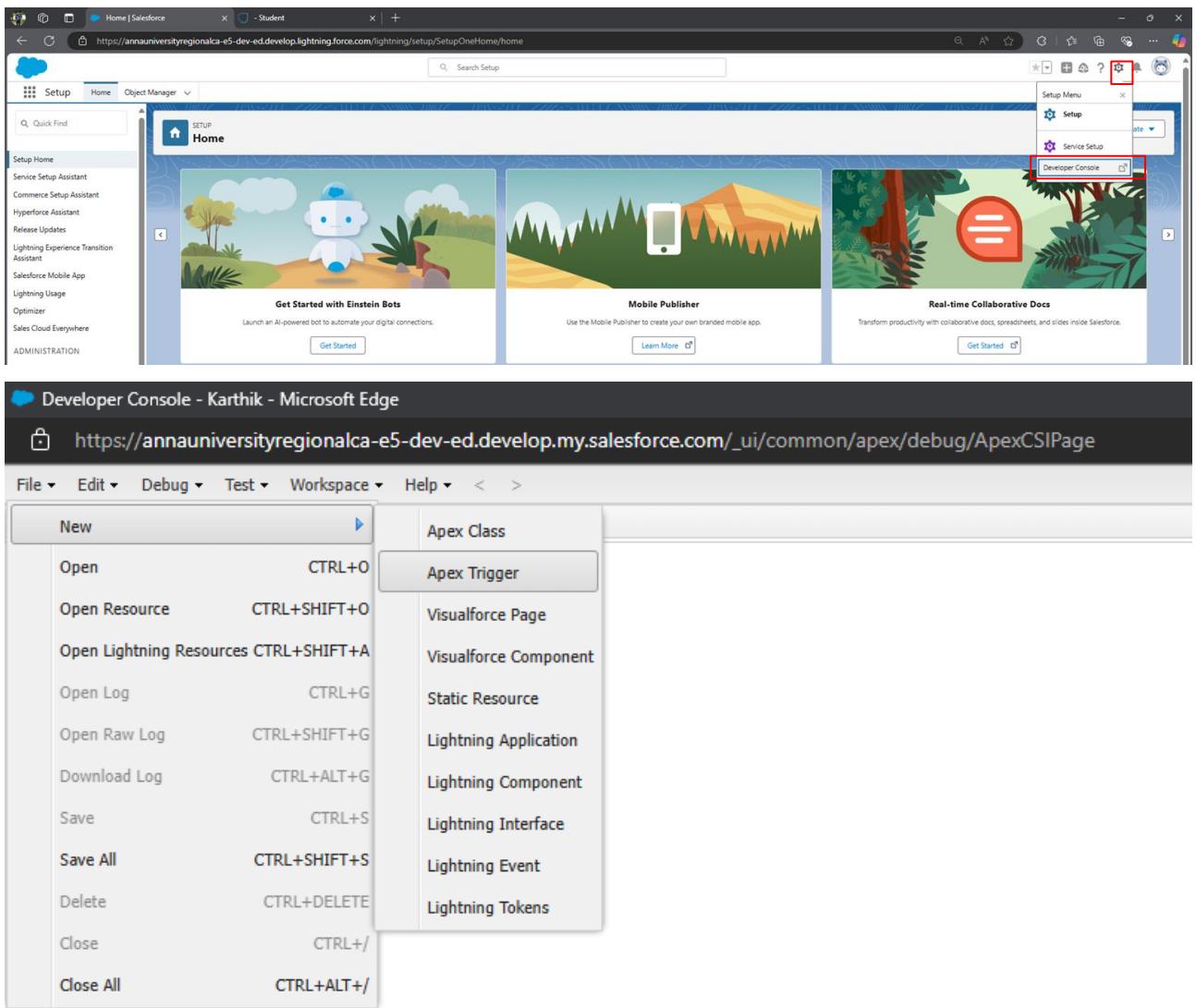
```

Developer Console - Karthik - Microsoft Edge
https://annaundergroundca-e5-dev-ed.develop.my.salesforce.com/u/common/apex/debug/ApexCSPage
File Edit Debug Test Workspace Help
LeaseTrackingTriggerHandler.apxc leasetrackingtrigger.apxt
Code Coverage: None API Version: 62 Go To
1 public class LeaseTrackingTriggerHandler {
2
3
4
5     public static void method1(List<Lease_Tracking__c> lt1)
6     {
7
8         for(Lease_Tracking__c lt2: lt1 )
9         {
10
11             if(lt2.Amount_to_be_paid__c > (lt2.Total_rent_Yearly__c)/2)
12             {
13
14                 Messaging.SingleEmailMessage M = New Messaging.SingleEmailMessage();
15
16
17                 List<String> ToADD = New List<String>{lt2.Email_id__c};
18
19
20
21                 M.setToAddresses(ToADD);
22
23
24                 M.setSubject('Regarding the Pending Rent');
25
26
27
28
Logs Tests Checkpoints Query Editor View State Progress Problems
User Application Operation Time Status Read Size
Filter Click here to filter the log list

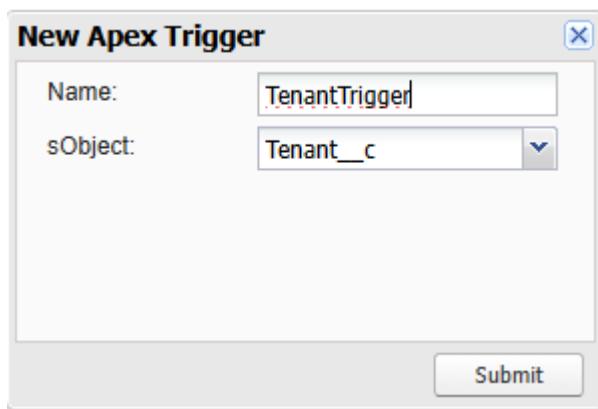
```

Activity 2: Write an Apex Trigger on Tenant Object to Show an error if the pan card is invalid.

1. Click on the gear icon and click on the developer console.
2. Click on the file and select New Apex Trigger



3. Name- TenantTrigger, Object - Tenant



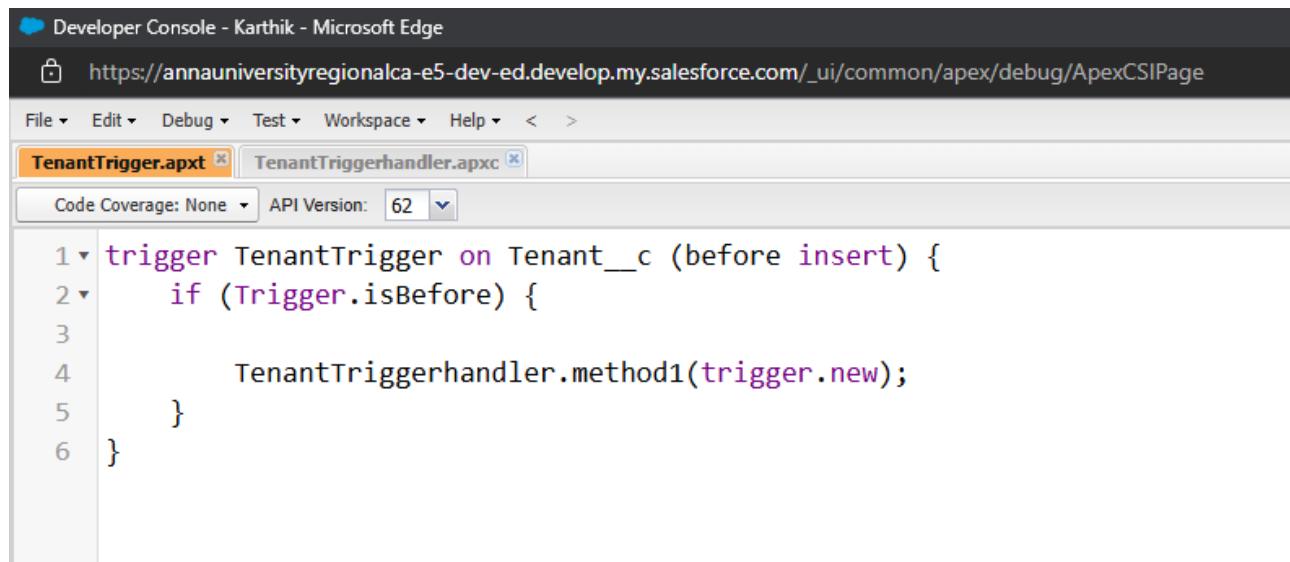
4. Use Events - Before insert and Trigger context Variable - IsBefore

Trigger: -

CODE SNIPPET: -

```
trigger TenantTrigger on Tenant__c (before insert)
```

```
{  
    if(Trigger.isBefore)  
    {  
        TenantTriggerhandler.method1(Trigger.New);  
    }  
}
```



The screenshot shows the Salesforce Developer Console in Microsoft Edge. The URL is https://annauniversityregionalca-e5-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. The tab bar shows 'TenantTrigger.apxt' is active. The code editor displays the trigger code:

```
1 trigger TenantTrigger on Tenant__c (before insert) {  
2     if (Trigger.isBefore) {  
3         TenantTriggerhandler.method1(trigger.new);  
4     }  
5 }  
6 }
```

Trigger Handler: -

1. Create an apex class and Name it TenantTriggerhandler

CODE SNIPPET: -

```
public class TenantTriggerhandler
```

```
{  
    public static void method1(List<Tenant__c> te)  
    {  
        for (Tenant__c tenant: te)  
        {  
            if(tenant.Pan_Card_no__c.length() > 10)  
            {  
                tenant.addError("This Pan Card number is invalid, Please Enter Valid Pan Card number");  
            }  
        }  
    }  
}
```

```
        }
    }
}
}
```

The screenshot shows the Salesforce Developer Console interface. The title bar reads "Developer Console - Karthik - Microsoft Edge" and the URL is "https://arunauniversityregionca-e5-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCsIPage". The tabs at the top include "File", "Edit", "Debug", "Test", "Workspace", "Help", and "TenantTrigger.apinx" (which is highlighted). Below the tabs, there are dropdowns for "Code Coverage: None" and "API Version: 62". A "Go To" button is also present. The main content area displays the Apex code for the TenantTriggerhandler class:

```
1 public class TenantTriggerhandler {
2
3     public static void Method1(List<Tenant__c> te) {
4         for (Tenant__c tenant : te) {
5             if (tenant.PAN_Card__c.length() > 10) {
6                 tenantaddError('This PAN Card number is invalid. Please enter a valid PAN Card number.');
7             }
8         }
9     }
10 }
```

Below the code editor, there is a navigation bar with tabs: "Logs" (which is selected), "Tests", "Checkpoints", "Query Editor", "View State", "Progress", and "Problems". Under the "Logs" tab, there is a table with columns: "User", "Application", "Operation", "Time", "Status", "Read", and "Size". The table is currently empty. At the bottom of the logs section, there is a "Filter" input field and a placeholder text "Click here to filter the log list".

Milestone 9 - Asynchronous Apex

Asynchronous Apex in Salesforce refers to a programming paradigm where code execution is detached from the immediate context and occurs independently, typically in the background. This approach is designed to handle long-running processes, heavy computations, or tasks that should not block user interactions.

Asynchronous Apex in Salesforce allows developers to perform operations that run in the background without blocking the user interface or delaying immediate actions. This type of processing is essential for handling large data volumes and time-consuming tasks, like making external web service calls, processing batches of records, or executing complex calculations. A common use case for Asynchronous Apex is data processing, where a batch job can update thousands of records without hitting governor limits, ensuring that Salesforce's resources are efficiently managed. Another use case involves scheduling tasks, such as nightly updates or reporting, where the job can be executed outside of peak hours, reducing the impact on system performance. By using Asynchronous Apex, organizations can improve the performance of their Salesforce org, maintaining responsiveness for users while handling critical background tasks.

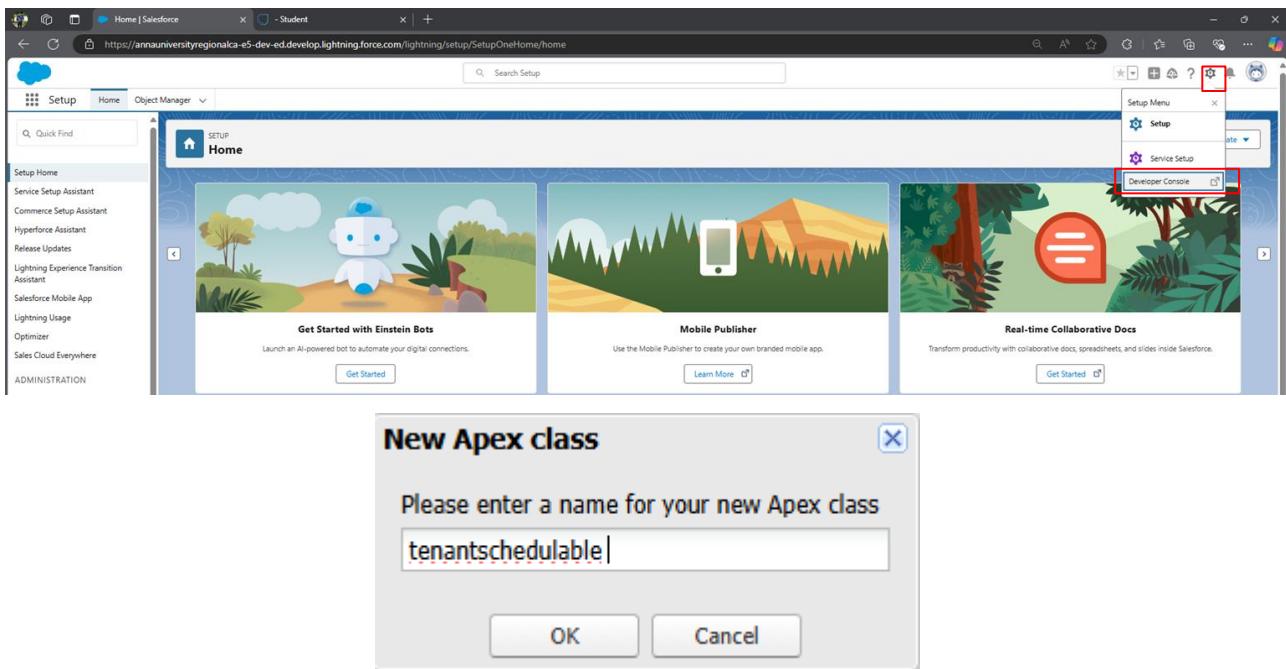
Use Case:

Asynchronous Apex in Salesforce is used to handle long-running or complex operations that don't need to finish immediately, allowing the system to manage resources efficiently. For example, it's ideal for processing large data sets, such as updating thousands of records in the background without impacting the user experience or hitting governor limits. Asynchronous Apex is also useful for integrating with external systems; it can make web service callouts to other platforms and handle responses later, without holding up the main process. In scenarios like scheduled reports, Asynchronous Apex enables automated, time-based actions, such as sending daily summaries of account activities without user intervention. This functionality allows Salesforce to scale operations, maintain system performance, and ensure that heavy tasks are completed in a timely manner without burdening the user interface.

Activity 1: Schedule Apex

Delete the Tenant Records Monthly whose Status Of Possession is closed.

1. Create a class with name tenantschedulable
2. Give extension Schedulable to the class.
3. Open the Anonymous Window.



4. Schedule the class-

```
tenantschedulable a = new tenantschedulable();
string cron = '0 0 0 1 * ? *';
system.schedule('Delete the records monthly', cron, a);
```

CODE SNIPPET: -

```
public class tenantschedulable implements Schedulable
{
    public void execute (Schedulablecontext sc)
    {
        list<Tenant__c> ten = [SELECT Id, Status_of_Possession__c FROM Tenant__c ];
        list<Tenant__c> tenantstodelete = New List<Tenant__c>();

        for (Tenant__c te: ten)
        {
            if(te.Status_of_Possession__c == 'Closed')
            {
                tenantstodelete.add(te);
            }
        }
        Delete tenantstodelete;
    }
}
```

Developer Console - Karthik - Microsoft Edge
https://annauniversityregionalca-e5-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help

tenantschedulable.apcpx

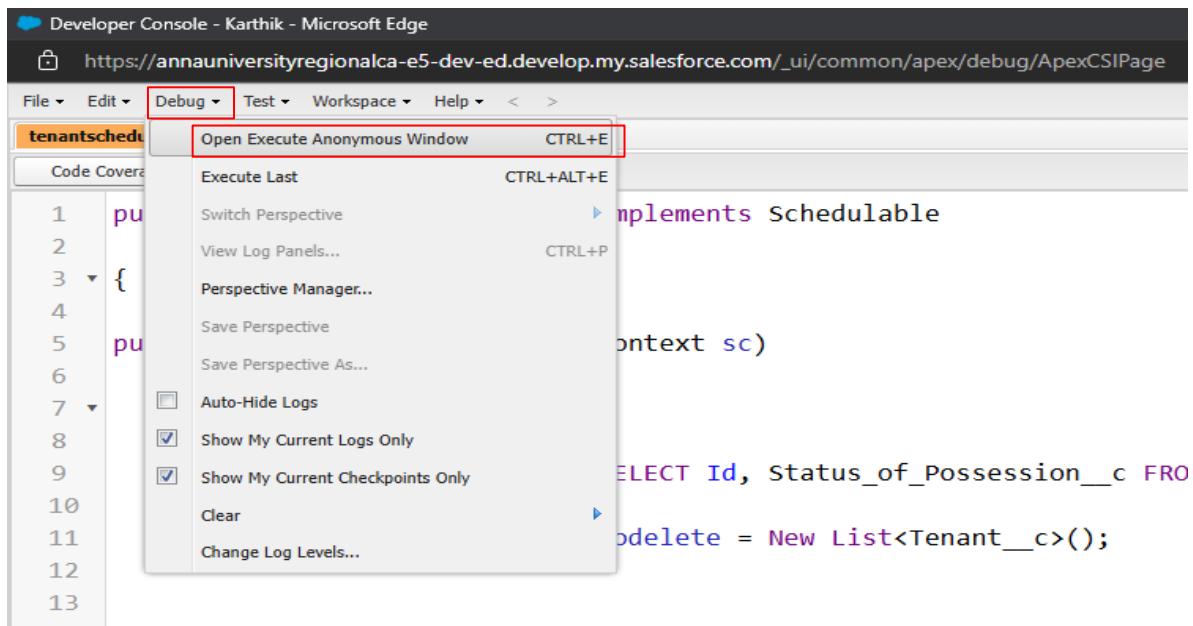
Code Coverage: None API Version: 62 Go To

```
1 public class tenantschedulable implements Schedulable
2 {
3     public void execute(Schedulablecontext sc)
4     {
5         list<Tenant__c> ten = [SELECT Id, Status_of_Possession__c FROM Tenant__c ];
6         list<Tenant__c> tenantstodelete = New List<Tenant__c>();
7
8         for(Tenant__c te: ten)
9         {
10            if(te.Status_of_Possession__c == 'Closed')
11            {
12                tenantstodelete.add(te);
13            }
14        }
15    }
16
17    void delete()
18    {
19        delete tenantstodelete;
20    }
21
22 }
```

Logs Tests Checkpoints Query Editor View State Progress Problems

User Application Operation Time Status Read Size

Filter Click here to filter the log list



Enter Apex Code

```
1 tenantschedulable a = new tenantschedulable();
2
3 string cron = '0 0 0 1 * ? * ';
4
5 system.schedule('Delete the records monthly', cron, a);
```

Open Log Execute Execute Highlighted

Milestone 10 - Create Reports And Dashboards

Salesforce Reports and Dashboards are powerful tools that empower users to visualize and analyse data within the Salesforce platform. They play a crucial role in providing insights, monitoring performance, and making informed business decisions.

Reports:

Reports give you access to your Salesforce data. You can examine your Salesforce data in almost infinite combinations, display it in easy-to-understand formats, and share the resulting insights with others. Before building, reading, and sharing reports, review these reporting basics.

Types of Reports in Salesforce

1. Tabular
2. Summary
3. Matrix
4. Joined Reports

Use Case:

The CEO of an organization wants to have brief data on employees working, project intake, project progress, Assets assigned, and the conditions of the Assets assigned. So he can have a clear picture of his organization and be able to make any decisions required based on this data. So he calls you on this task and wants you to represent the data appropriately.

Dashboards:

Dashboards help you visually understand changing business conditions so you can make decisions based on the real-time data you've gathered with reports. Use dashboards to help users identify trends, sort out quantities, and measure the impact of their activities. Before building, reading, and sharing dashboards, review these dashboard basics.

Use Case:

As an Admin for the organization, you keep pushing yourself to reach out the business requirements to take the organization to peak heights and all your superiors are very much impressed with your efforts and work dedication. In addition with reports, you make an ease for the CEO to view the reports with data visualization. So he doesn't have to search for the data he wants during the meetings.

Activity 1: Create a Report of lease Management Records

The Manager needs a report that shows the tenant and their joining date and their Remaining payment details also group this by date of Registration, and make a bucket list of the remaining amount as greater than 1000000 red, less than 1000, and greater than 500000 as blue and less than equal to 500000 as yellow.

The screenshot shows the SmartMall application interface. At the top, there is a navigation bar with links for SmartMall, Tenants, Leases Tracking, Tenant Issues, Reports, and Dashboards. A 'Recently Viewed' section lists tenants with their names and IDs. A modal window titled 'Edit SmartMall App Navigation Items' is open, showing a list of navigation items: Tenants, Leases Tracking, Tenant Issues, Reports, and Dashboards. An 'Add More Items' button is highlighted with a red box. Below this, another modal window titled 'Add Items' is open, showing a list of available items under the 'All' tab. The items listed include Accounts, Alternative Payment Methods, Analytics, App Launcher, Appointment Categories, Appointment Invitations, Approval Requests, Asset Action Sources, Asset Actions, Asset State Periods, Assets, Async Operation Logs, Authorization Form, Authorization Form Consent, Authorization Form Data Use, Authorization Form Text, Background Operations, and Business Brands. A search bar at the top of the 'Add Items' modal is empty. At the bottom of the 'Add Items' modal are 'Cancel' and 'Add Nav Items' buttons.

1) Create a new Folder and name it as MallReports

2) Click On new report ==> Select object Activities with LeaseTracking ==> Click on start report

3) Click on the Amount to be paid column click on bucket this list and name it as the Remaining amount

- 4) Save the report named as lease report and Save it in MallReports.

Activity 2: Create a Report on Tenant issue Records

Now the manager is asking for a report on issues that have not been contacted or opened yet and have a high priority which is directly encountered by Phone and Mail and the date of issue is from the last 7 days.

The screenshot shows the Salesforce Reports interface. On the left, there's a sidebar with categories like Reports, Recent, and Reports. Under Reports, 'Recent' is selected, showing four items: 'Leases Tracking Report', 'Tenant Issues Report', 'Tenants with Leases Tracking', and 'Tenant Details'. A red box highlights the 'New Folder' button at the top right of the main report list area. In the center, a modal window titled 'Create folder' is open, also with a red border. It has two input fields: 'Folder Label' and 'Folder Unique Name', both currently empty. At the bottom of the modal are 'Cancel' and 'Save' buttons. Below the modal, the main report list shows the same four recent reports again. To the right of the main list, there's a search bar and a 'New Report' button. At the very bottom of the page, another modal window titled 'Create Report' is open, also with a red border. This modal has a sidebar on the left with categories like 'Recently Used' and 'All'. The main area lists various report types under 'Select a Report Type', such as 'Accounts & Contacts', 'Opportunities', 'Customer Support Reports', etc., each with a 'Category' column. A red box highlights the 'All' category in the sidebar. The 'Create Report' modal also has a search bar at the top.

1. Click On new report ==> Select object Activities with Tenant issue==> Click on start report
2. Click on save, enter Name - Issue Report

- Choose the folder Mall Reports and save.

Activity 3: Create a Report on Tenant Records

Now, The Manager wants a Report that shows all the pending possessions and also shows the tenant's Pan Card no and GST NO and group date of reg by column and row by Status of Possession.

- Click On new report ==> Select object Tenants ==> Click on start report
- Choose the folder Mall Reports.
- Save the report and Name it Tenant Details.

Activity 4: Create a Dashboard

Very Good, You have created multiple reports but now for better convenience, the owner wants a Dashboard that shows the data of these reports, So Create a Dashboard and follow the instructions below -

- 1) To Create a Dashboard first create a folder to store the dashboard and save it as Mall Dashboard.

Create folder

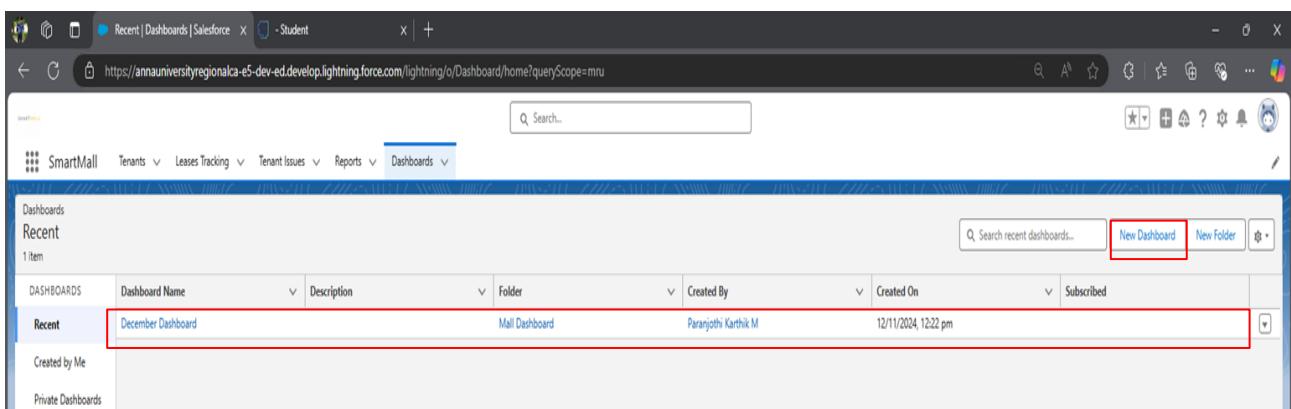
* Folder Label
Mall Dashboard

* Folder Unique Name
MallDashboard

Cancel Save

- 2) Now click on New Dashboard

- 3) Enter Name - December Dashboard, select the Mall Dashboard folder, and click on Create



The screenshot shows the Salesforce Lightning interface for creating a new dashboard. The browser address bar indicates the URL is <https://annauniversityregionalca-e5-dev-ed.lightning.force.com/lightning/o/Dashboard/home?queryScope=mru>. The page title is "Recent | Dashboards | Salesforce". The top navigation bar includes links for SmartMall, Tenants, Leases Tracking, Tenant Issues, Reports, and Dashboards. The main content area displays a table titled "Dashboards" under the "Recent" tab. The table has columns: Dashboard Name, Description, Folder, Created By, Created On, and Subscribed. A single row is visible, showing "December Dashboard" under "Dashboard Name", "Mall Dashboard" under "Folder", "Paranjithi Karthik M" under "Created By", and "12/11/2024, 12:22 pm" under "Created On". A red box highlights the "New Dashboard" button in the top right corner of the table header.

- 4) Click on + Component and Select Tenant Details, Display as Horizontal Bar Chart,

Dimensions - Height * width = 10*12.

- 5) Click on + Component and Select Issue Report, Display as Lightning Table,

Dimensions - Height * width = 8*6.

- 6) Click on + Component and Select lease report, Display as Lightning Table,

Dimensions - Height * width = 8*6

- 7) Click on Save and Done.

Output:

The screenshot displays the December Dashboard in Salesforce. The top navigation bar includes links for SmartMall, Tenants, Leases Tracking, Tenant Issues, Reports, and Dashboards. The dashboard header shows "December Dashboard" and the date "As of 14-Nov-2024, 8:06 pm Viewing as Parangothi Kartik M".

Tenant Details: A chart titled "Record Count" showing the count of tenants over time. The x-axis represents the date from 10/11/2024 to 11/11/2024. The y-axis represents the record count from 0 to 9. The chart shows a single data point at 10/11/2024 with a value of 1.

Tenant Issues Report: A table listing tenant issues categorized by origin (Phone, Mail) and priority (Low, Medium, High). The columns include Origin, Priority, Status, and Tenant Name. The data is as follows:

Origin	Priority	Status	Tenant Name
1. Phone	2. Medium	Open	Suresh
1. Phone	2. Medium	Working	Sita
1. Phone	1. Low	Working	Ram
2. Mail	2. Medium	Open	Dinesh
2. Mail	3. High	In progress	Ramesh
2. Mail	2. Medium	Open	Tamilselvan
2. Mail	3. High	In progress	Siva

Leases Tracking Report: A table listing lease tracking information. The columns include Tenant Name, Lease Tracking No., Amount to be paid, and Date of Possession. The data is as follows:

Tenant Name	Lease Tracking No.	Amount to be paid	Date of Possession
Dinesh	TT - 000001	150.00k	01/11/2025
Ganesh	TT - 000010	100.00k	01/08/2025
Karthik	TT - 000009	200.00k	01/07/2025
Lakshman	TT - 000005	232.00k	01/02/2025
Ram	TT - 000008	200.00k	01/05/2025
Ramesh	TT - 000003	155.00k	01/02/2025
Sita	TT - 000007	275.00k	01/04/2025

Key Scenarios Addressed by Salesforce in the Implementation Project:

- Centralized Data Management:** Salesforce provides a unified platform to store and manage tenant, customer, and operational data, eliminating data silos and enabling easy access for stakeholders.
- Enhanced Customer Engagement:** Tools like Marketing Cloud and Customer 360 help create personalized marketing campaigns, loyalty programs, and shopper engagement strategies based on detailed customer insights.
- Seamless Tenant Relationship Management:** Salesforce enables efficient tracking of lease agreements, rent payments, and communications with tenants, ensuring smooth tenant relationships and streamlined processes.
- Automation of Business Processes:** Through Salesforce's automation features, tasks like rental invoicing, maintenance request tracking, and inventory management are automated, reducing manual effort and improving accuracy.
- Real-Time Analytics and Reporting:** Salesforce dashboards and reporting capabilities provide actionable insights on key performance indicators, helping management make informed decisions and optimize operations.
- Scalable and Secure Platform:** Salesforce ensures scalability to support malls of varying sizes and employs robust security measures to protect sensitive data and ensure compliance with industry standards.

Conclusion of the Mall Management CRM Application

1. **Improved Operational Efficiency:** The CRM application simplifies complex workflows, automates repetitive tasks, and streamlines mall management processes, leading to increased productivity.
2. **Enhanced Customer and Tenant Satisfaction:** By facilitating better engagement through personalized marketing, loyalty programs, and responsive tenant management, the application fosters stronger relationships with both shoppers and tenants.
3. **Data-Driven Decision-Making:** Real-time analytics and reporting empower mall administrators to make informed decisions, optimize resource allocation, and identify growth opportunities.
4. **Scalability and Adaptability:** The CRM is designed to grow with the mall's needs, making it suitable for various sizes and operational complexities while maintaining high performance and reliability.
5. **Security and Compliance:** With robust data protection measures and adherence to industry standards, the application ensures the safety and privacy of sensitive information, building trust among users.