

A CRM Application to Manage the Mall

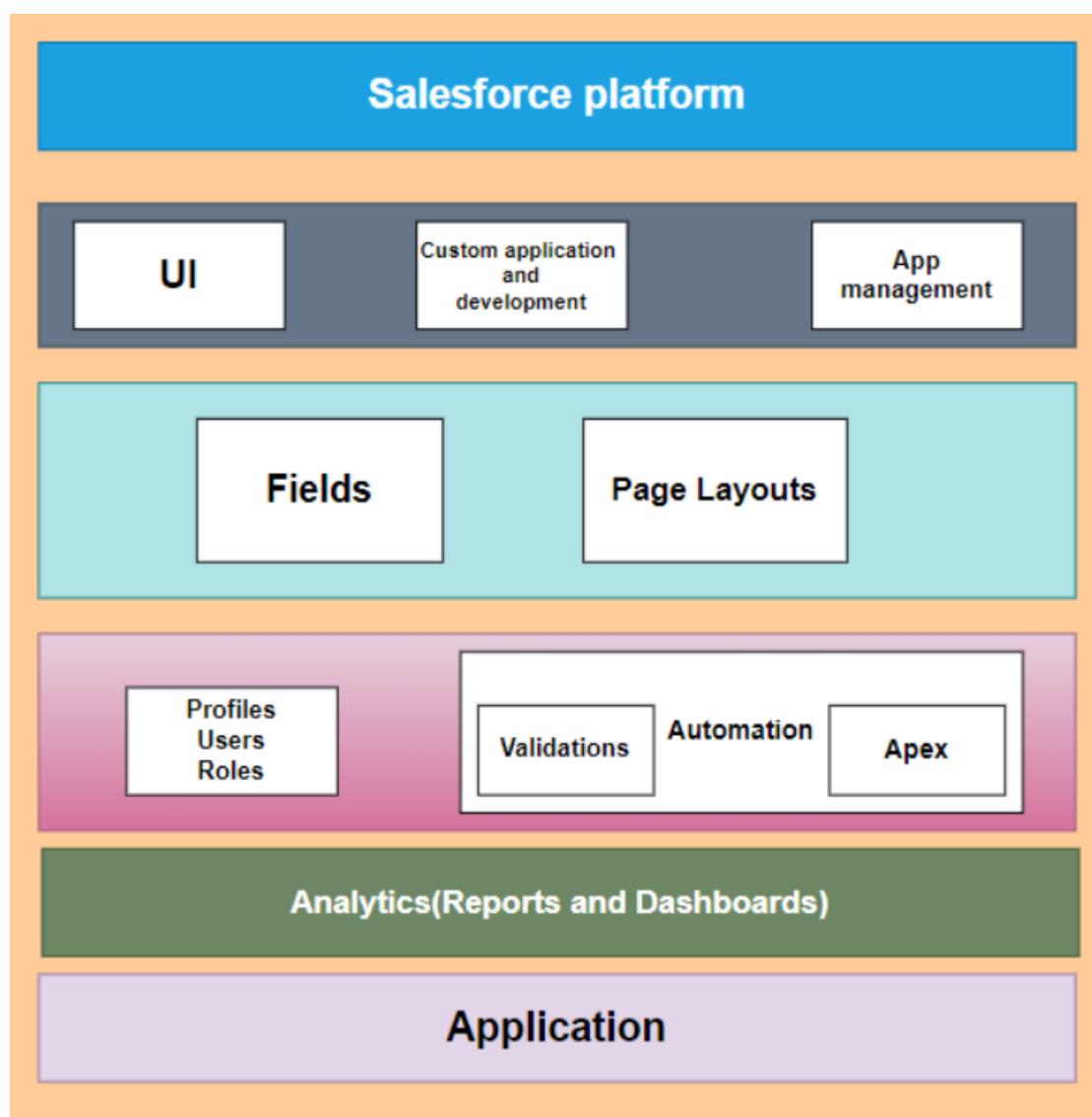
Project Description:

“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.

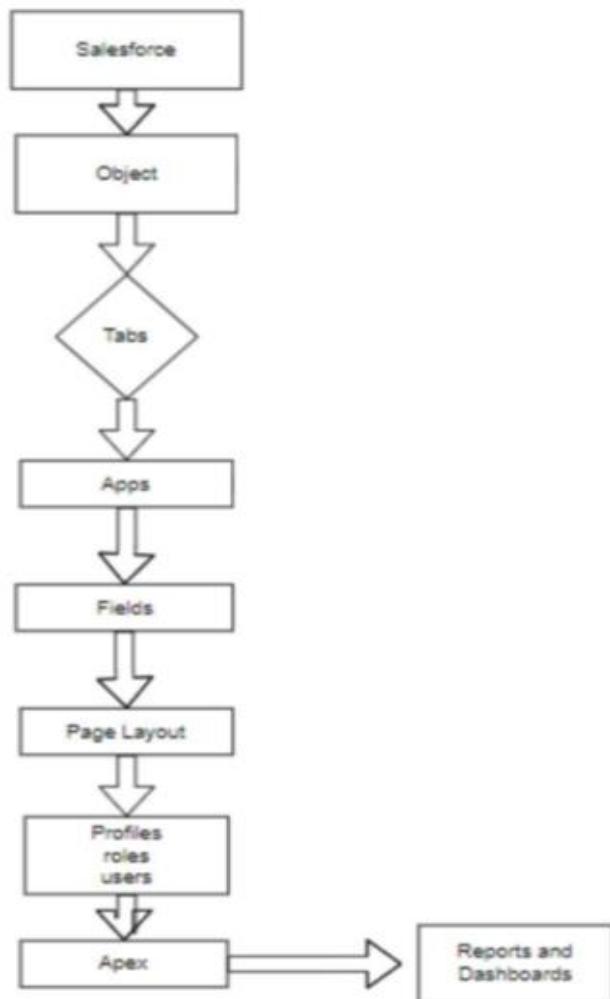
Short Description:

“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:



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, 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.

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, we 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 TheSmartBridge, you must have a Salesforce developer edition org in order to do all the required works which the CEO desires for TheSmartBridge.

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.

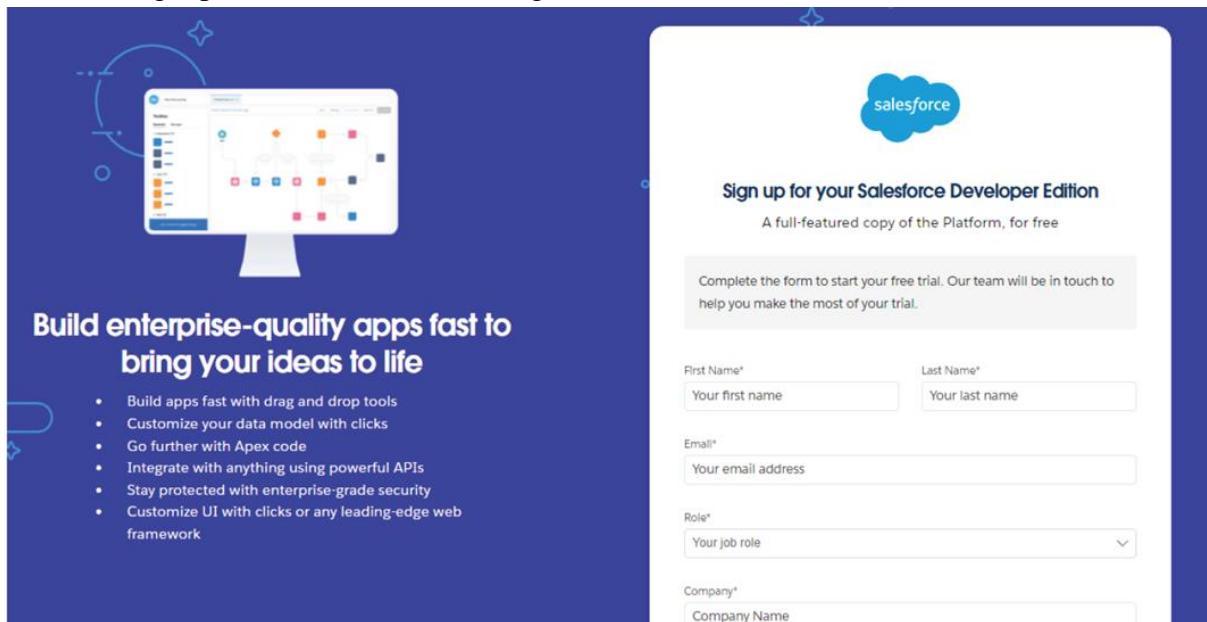
3	Enterprise	Meets the needs of large and complex businesses. It gives you advanced customization and administration tools, in addition to all the functionality available in Professional Edition, that can support large-scale deployments. Enterprise Edition also includes access to Salesforce APIs, so 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:

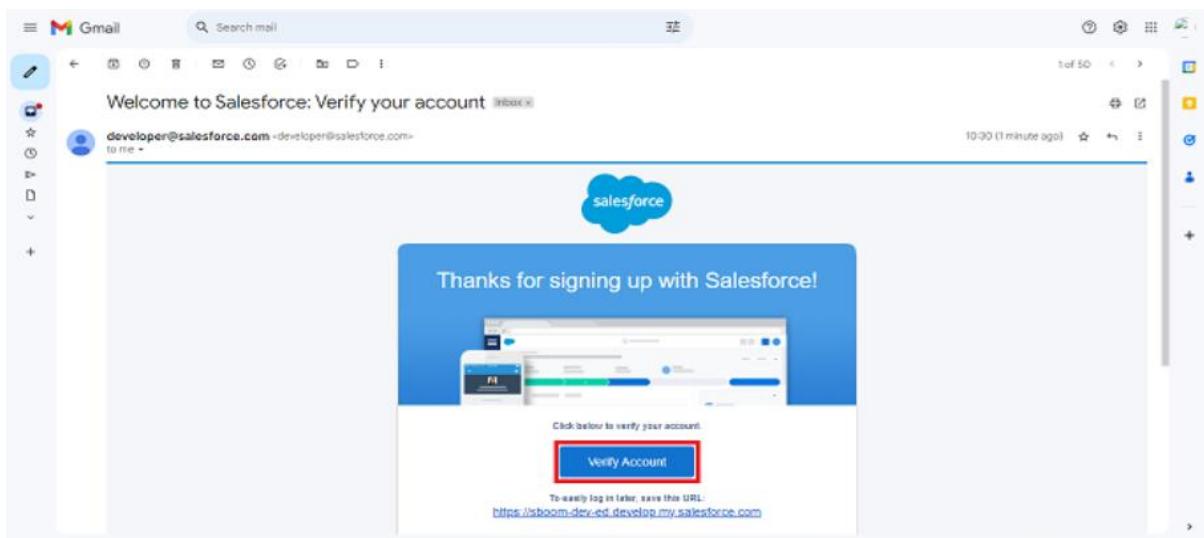


- 1) First name & Last name
- 2) Email
- 3) Role: Developer
- 4) Company: College Name

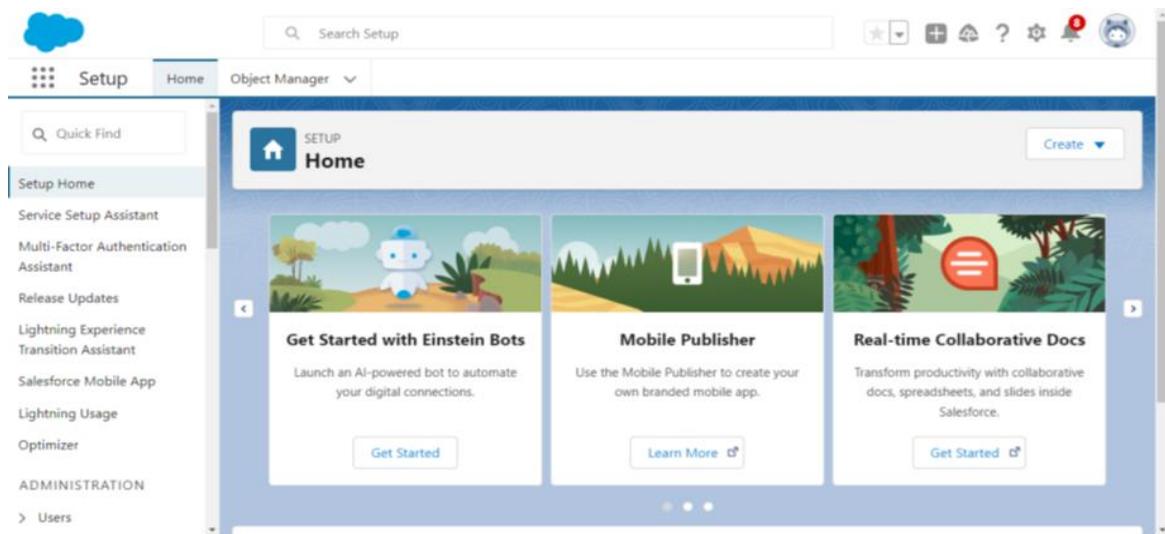
- 5) County: India
 - 6) Postal Code: pin code
 - 7) Username: should be a combination of your name and company
This need not be an actual email ID, you can give anything in the format:
username@organization.com
- Click on Sign me up after filling these.

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 form asks for a new password that is at least 8 characters long, containing one letter and one number. The "New Password" field contains "....." and is labeled "Good". The "Confirm New Password" field contains "....." and is labeled "Match". Below these fields is a "Security Question" section with the question "In what city were you born?". The "Answer" field contains "asdfghjkl". The entire form is highlighted with a red box.

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 TheSmartBridge, 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	
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 tabs, page layouts, and reports.

Label	Example: Account
Plural Label	Example: Accounts
Starts with vowel sound	<input type="checkbox"/>

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

Object Name	Example: Account
-------------	------------------

Description:

Context Sensitive Help Setting

- Open the standard Salesforce tabs 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 tabs, related lists, lookups, 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	Example: Account Name
-------------	-----------------------

Data Type

Text

Optional Features

- Allow Reports
- Allow Analytics
- Allow Fast 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' page in the Salesforce setup. A new object is being created with the following details:

- Label:** Lease Tracking
- Plural Label:** Leases Tracking
- Object Name:** Lease_Tracking
- Description:** (empty)
- Record Name:** Lease Tracking No
- Data Type:** Auto Number
- Display Format:** TT - {000000}

The 'Optional Features' section has 'Allow Reports' checked. The page includes standard Salesforce navigation and search bars at the top.

5. Allow search and Save.

The screenshot shows the 'Object Manager' page after saving the new object. The table displays one item:

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

The page includes standard Salesforce navigation and search bars at the top.

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. The search bar at the top contains the text "lead". A table below lists one item:

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. The search bar at the top contains the text "tenant". A table below lists two items:

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 TheSmartBridge 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.

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.

Action	Label	Tab Style	Description
Edit Del	Lease Tracking	Boat	
Edit Del	Tenant Issues	Computer	
Edit Del	Tenants	Apple	

Web Tabs

No Web Tab have been defined

Visualforce Tabs

No Visualforce Tabs have been defined

Lightning Component Tabs

No Lightning component tabs have been defined

Lightning Page Tabs

No Lightning Page Tabs have been defined

New Custom Object Tab

Step 1. Enter the Details

Choose the custom object for this new custom tab. Fill in other details.

New Custom Object Tab

Required information

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

Object	Tab Style
None	None

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

Splash Page Custom Link: None

Enter a short description.

Description:

Next | Cancel

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 includes sections for 'Custom Object Tabs', 'Web Tabs', 'Visualforce Tabs', 'Lightning Component Tabs', and 'Lightning Page Tabs'. Each section has a 'New' button and a 'What Is This?' link. The 'Custom Object Tabs' section lists three objects: Lease Tracking, Tenant Issues, and Tenants, each with a 'Tab Style' dropdown set to 'Basic'.

The screenshot shows the 'New Custom Object Tab' wizard, Step 1 of 3. It asks to enter details for a new custom tab for the 'Tenant Issue' object. The 'Object' dropdown is set to 'Tenant Issue' and the 'Tab Style' dropdown is set to 'Basic'. There is an optional field for a 'Splash Page Custom Link' and a 'Description' text area. A 'Next' button is visible at the bottom right.

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 Tabs setup page. In the left sidebar, under 'User Interface', 'Tabs' is selected. The main content area is titled 'Custom Tabs' and contains the following sections:

- Custom Object Tabs:** Displays three tabs: 'Leases_Tracking' (Red), 'Tenant_Issues' (Blue), and 'Tenants' (Yellow). Each tab has 'Edit' and 'Delete' buttons.
- Web Tabs:** Shows a message: "No Web Tabs have been defined".
- Visualforce Tabs:** Shows a message: "No Visualforce Tab have been defined".
- Lightning Component Tabs:** Shows a message: "No Lightning component tabs have been defined".
- Lightning Page Tabs:** Shows a message: "No Lightning Page Tabs have been defined".

The screenshot shows the 'New Custom Object Tab' creation wizard, Step 1 of 3. The left sidebar shows 'Tabs' is selected. The main content area is titled 'Step 1. Enter the Details' and includes the following fields:

- Select an existing custom object or create a new custom object now:** A dropdown menu for 'Object' is set to 'None'. A 'Create New Object' button is shown next to the dropdown.
- Tab Style:** A dropdown menu for 'Tab Style' is set to 'Standard'.
- (Optional) Choose a Home Page Custom Link to show as a splash page the first time your users click on this tab:** A dropdown menu for 'Splash Page Custom Link' is set to 'None'.
- Description:** A text input field with placeholder text 'Enter a short description.' and a 'Description' label.

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 Enrollment 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.

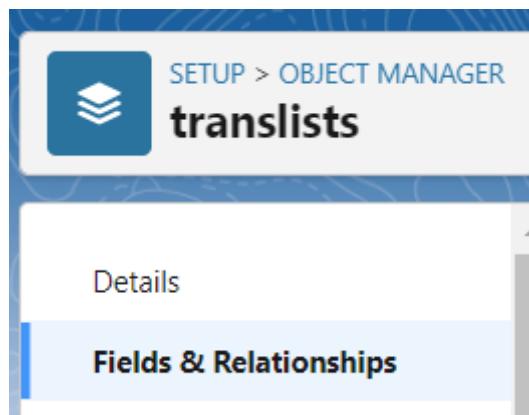
Use Case:

In Salesforce, fields and relationships 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 enroll 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 'New Custom Field' setup screen. The left sidebar shows 'Fields & Relationships' selected. The main form is titled 'Step 2. Enter the details' for creating a 'New Custom Field' under the 'Tenant' object. It includes fields for 'Field Label', 'Field Name', 'Description', 'Help Text', 'Required' (unchecked), 'Auto add to custom report type' (checked), 'Default Value' (with a formula editor link), and a note about formula syntax. Navigation buttons 'Previous', 'Next', and 'Cancel' are at the bottom right.

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_possession_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 with the path "SETUP > OBJECT MANAGER". Under "Tenant", the "Validation Rule Edit" screen is displayed. The "Rule Name" field contains "Phone Validation", which is highlighted with a red box. The "Error Condition Formula" field contains the formula "NOT(OR(REGEX(Phone_Number__c, "[^0-9]{10}")))" and is also highlighted with a red box. A sidebar on the right 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 "Example" field contains "Discount percent cannot exceed 30%". The "Error Message" field contains "Enter Valid 10 digit Phone number". The "Error Location" dropdown is set to "Top of Page". At the bottom are "Save", "Save & New", and "Cancel" buttons.

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()

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

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**

Errors: **Date**

Description:

Error Condition Formula

Example: [Discount_Percent_c>30](#) [More Examples...](#)

If this formula expression is true, display the text defined in the Error Message area

Functions

- ABS
- ACOS
- CEILMONTHS
- INT
- ASCII
- ASIN

Insert Selected Function

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

Help on this function

Check Syntax

- 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**

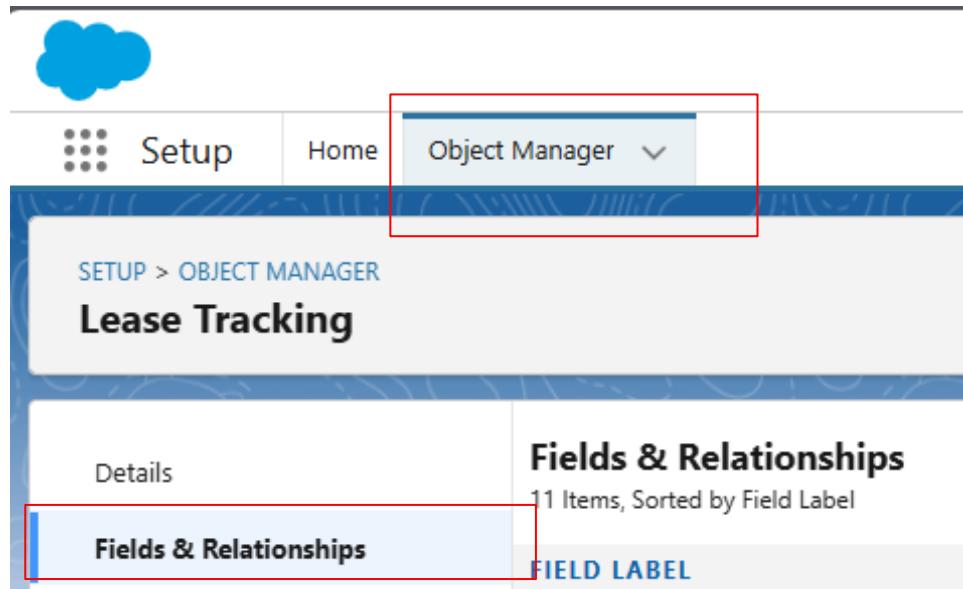
This error message can either appear at the top of the page or below a specific field on the page

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 screen in Salesforce. The title bar says 'Lease Tracking > New Custom Field'. The left sidebar lists various setup categories like Details, Fields & Relationships, Page Layouts, etc. The main form is titled 'Step 2. Enter the details' and is Step 2 of 4. It contains fields for 'Field Label' (with a placeholder 'Date of Possession'), 'Field Name' (placeholder 'DateofPossession'), 'Description', 'Help Text', 'Required' (unchecked), 'Auto add to custom report type' (unchecked), 'Always require a value in this field in order to save a record' (unchecked), 'Add this field to existing custom report types that contain this entity' (checked), and 'Default Value' (with a formula editor). The formula editor shows the code: 'Use formula editor. Enter text and positive values in double quotes ("04-Jan"). Include numbers without quotes (123) and use commas as decimals (.10), and express date values in the standard format: (Today + 7). To reference a field from a Custom Metadata type record use: \$CustomMetadata_Type__mdt__RecordName.FieldName'. Navigation buttons 'Previous', 'Next', and 'Cancel' are at the bottom right.

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	CreatedBy	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	LastModifiedBy	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

Lease Tracking Validation Rule

Rule Name: **Possession_Validation**

Description:

Error Condition Formula

Example: `Discount_Percent__c>0.30` 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

Date_of_Possession_c < TODAY() + 60

Functions

- ABS
- ACOS
- ADDMONTHS
- AND
- ASCII
- ASIN

Check Syntax

- 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.

Error Message

Example: **Discount percent cannot exceed 30%**

This message will appear when Error Condition formula is true

Error Message: **Enter a date after 60 days**

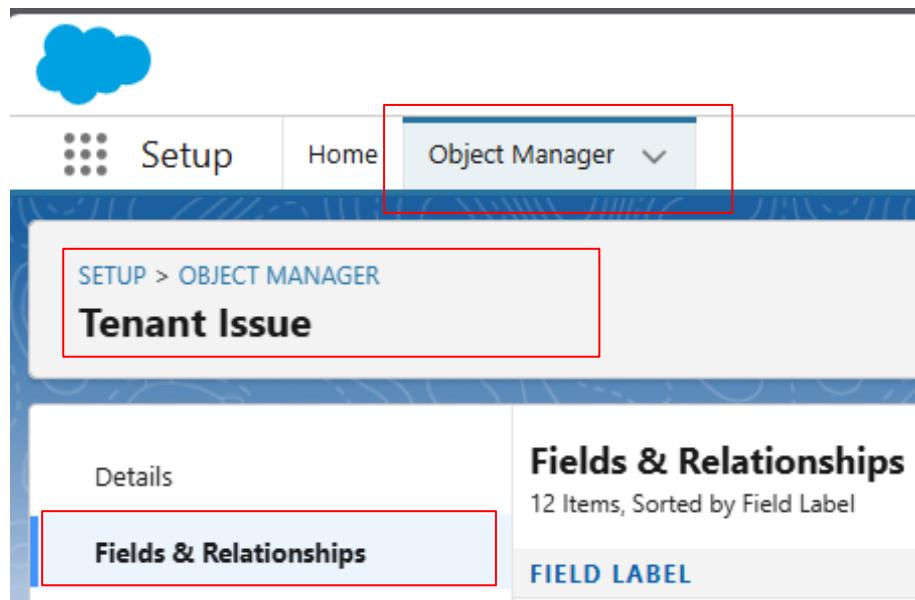
This error message can either appear at the top of the page or below a specific field on the page

Error Location: Top of Page Field **Date of Possession**

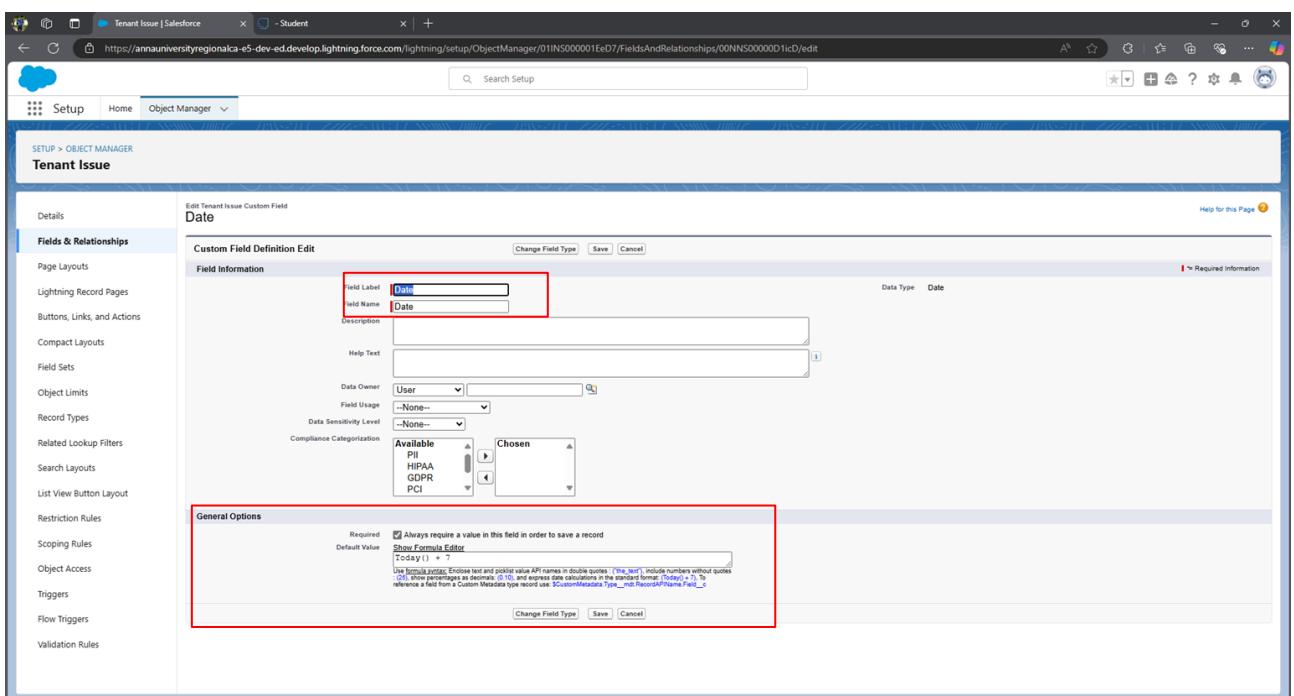
Save Save & New Cancel

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/01IN5000001EeD7/FieldsAndRelationships/view

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	CreatedById	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	LastModifiedById	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 color 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 TheSmartBridge organization, it's your duty to make sure every user of the organization is able to access the data modeling 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

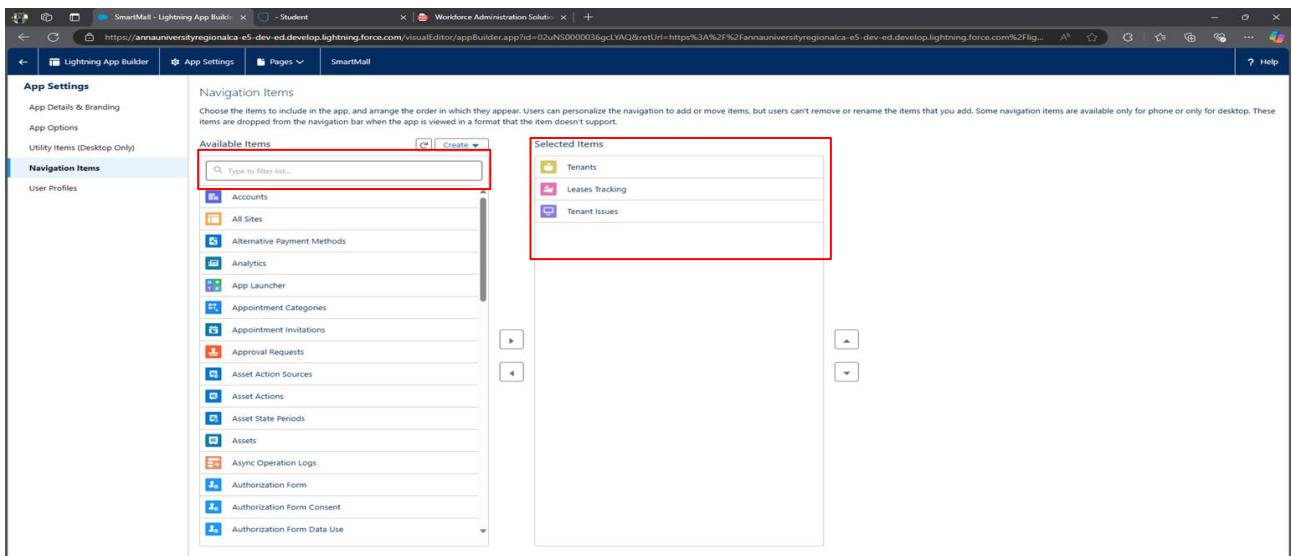
Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.

App Details	App Branding
<input type="text" value="SmartMall"/> * App Name <input type="text" value="SmartMall"/> Name your app...	<input type="text" value="SmartMall"/> Primary Color Hex <input type="text" value="#007002"/> <input type="button" value="Upload"/>
<input type="text" value="SmartMall"/> * Developer Name <input type="text" value="SmartMall"/> Enter a developer name...	
<input type="text" value="SmartMall"/> Description <input type="text" value="SmartMall"/> Enter a description...	<input type="checkbox"/> Org Theme Options <input type="checkbox"/> Use the app's image and color instead of the org's custom theme
App Launcher Preview	
<input type="button" value="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.

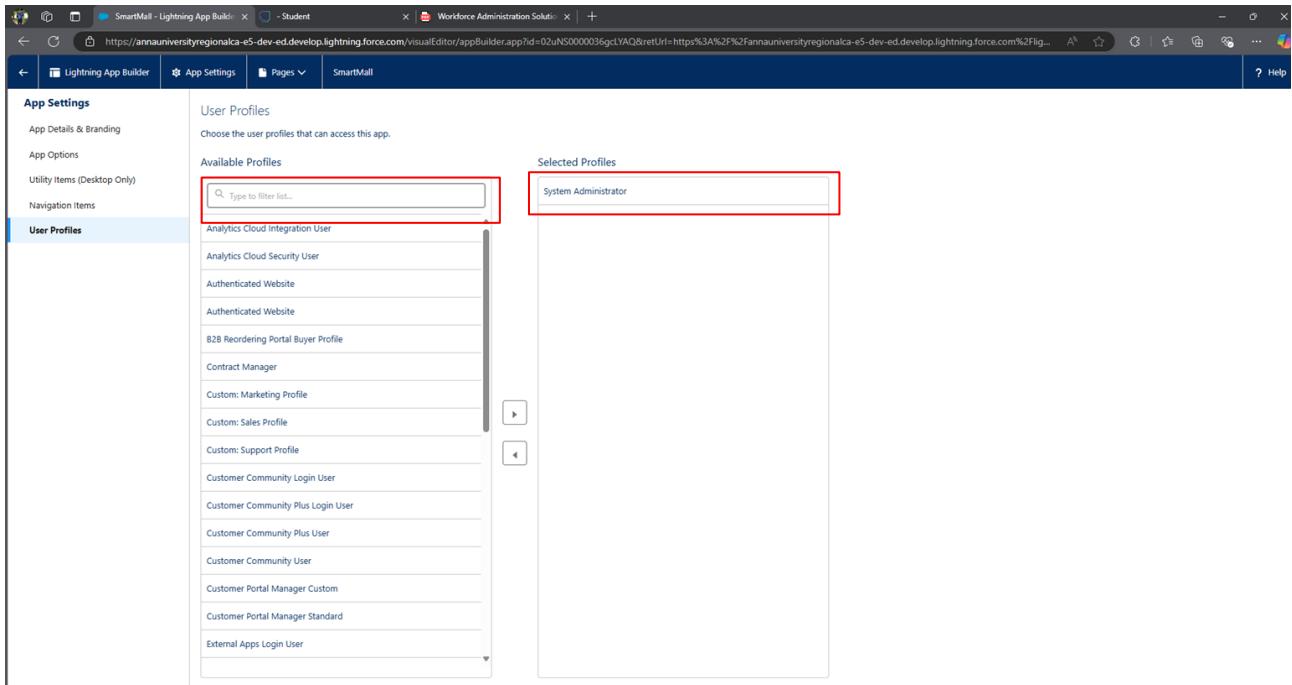
6. Click Next, Next and you can see a Navigation Items window like this:

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



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

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



- Click on Save and Finish.
- 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. The 'SmartMall' app is highlighted with a red box. The 'Setup' icon in the top-left corner is also highlighted with a red box. The 'SmartMall' app card displays its name and a brief description: 'SmartMall lets you manage your tenants and their leases in one place.' Below the app cards, there are sections for 'All Items' and 'Recently Viewed'.

The screenshot shows the 'Tenants' list page for the SmartMall object. The 'Tenants' tab in the navigation bar is highlighted with a red box. The 'New' button in the top-right corner of the list header is also highlighted with a red box. 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	Paranjothi 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:			

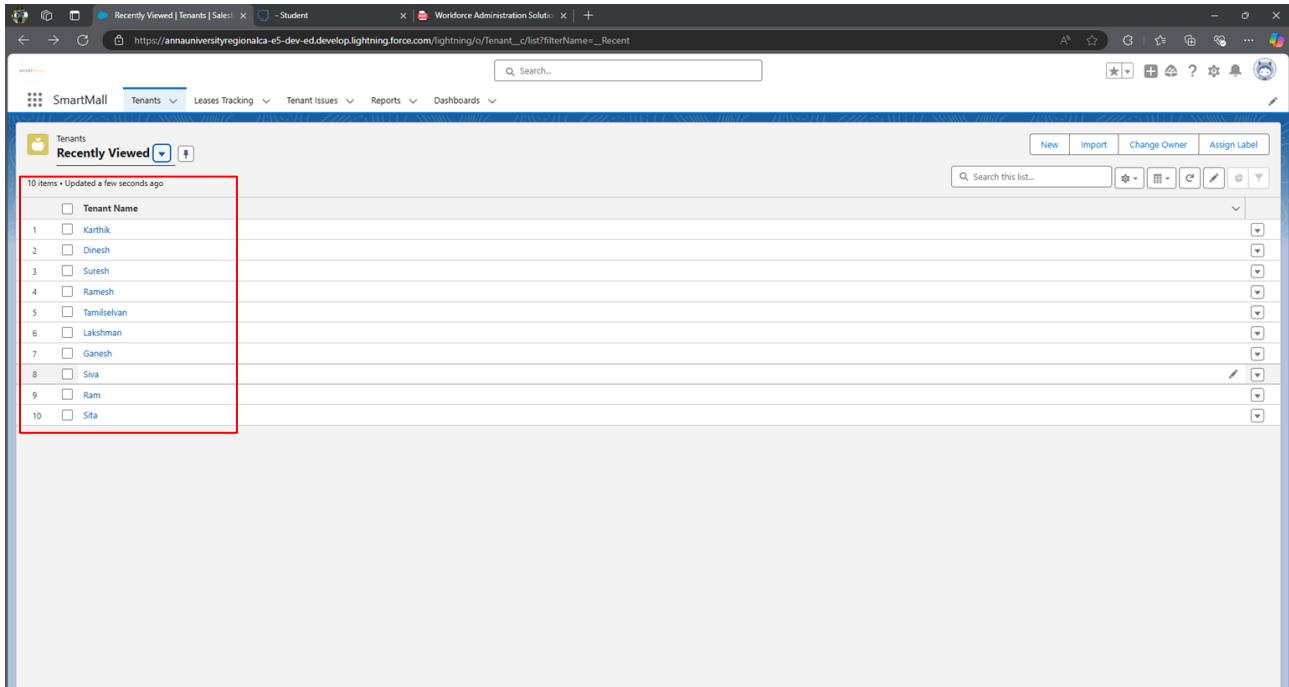
3. If you Enter a Phone Number of more or less than 10 digits it will show an error.
4. Similarly, if you enter DateofReg a Past date it will show an error.
5. After creating a record the page will look like this

The screenshot shows a Salesforce Lightning component for viewing a tenant record. The top navigation bar includes links for SmartMall, Tenants, Leases Tracking, Tenant Issues, Reports, and Dashboards. The main header displays the tenant's name, "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 screen, there are buttons for "New Contact", "Edit", and "New Opportunity". At the bottom, it shows the "Created By" and "Last Modified By" information, both listed as "Paranjothi Karthik M" on 10/11/2024 at 11:35 pm.

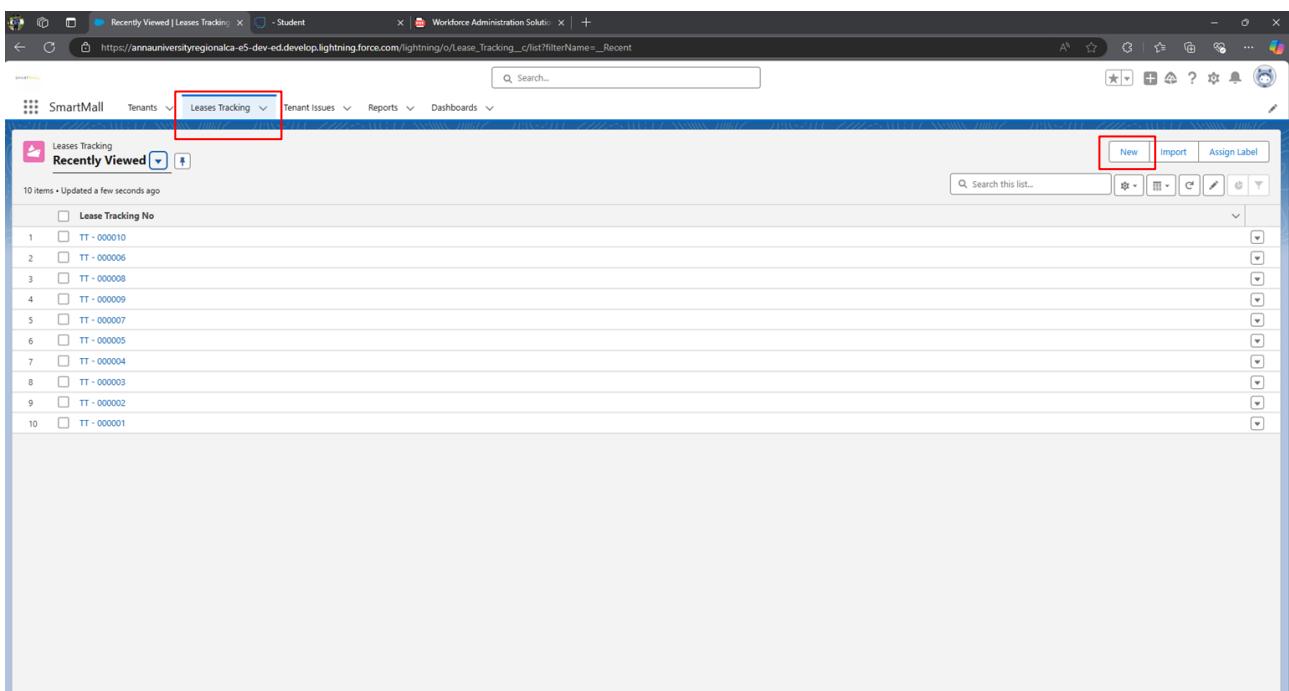
[Note]: Create at least 10 records in the tenant object.



The screenshot shows the Salesforce Lightning interface with the URL https://annauniversityregionalca-e5-dev-ed.lightning.force.com/lightning/o/Tenant__c/list?filterName=_Recent. The page title is "Tenants Recently Viewed". The list view displays 10 items, each with a checkbox and a tenant name. The names listed are: 1. Karthik, 2. Dinesh, 3. Suresh, 4. Ramesh, 5. Tamilselvan, 6. Lakshman, 7. Ganesh, 8. Siva, 9. Ram, 10. Sita. A red box highlights the first item in the list.

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 with the URL https://annauniversityregionalca-e5-dev-ed.lightning.force.com/lightning/o/Lease_Tracking__c/list?filterName=_Recent. The page title is "Leases Tracking Recently Viewed". The list view displays 10 items, each with a checkbox and a lease tracking number. The numbers listed are: 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. A red box highlights the "Leases Tracking" tab in the navigation bar, and another red box highlights the "New" button in the top right corner.

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... X

Edit TT - 000002

* = Required Information

Lease Tracking No
TT - 000002

Related Tenant
Suresh

* Date of Possession

Format: 31/12/2024

* End Date of Possession

* Total Year of Contract

* Total rent(Yearly)

* Amount Paid

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

Lease Tracking
TT - 000001

Related Details

Lease Tracking No
TT - 000001

Related Tenant
Dinesh

Date of Possession
01/11/2025

End Date of Possession
01/11/2026

Total Year of Contract
1

Total rent(Yearly)
2,000.00

Amount Paid
50,000

Amount to be paid
1,50,000.00

Email id

Created By
Paranjithi Karthik M. 11/11/2024, 11:19 am

Last Modified By
Paranjithi Karthik M. 11/11/2024, 11:19 am

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

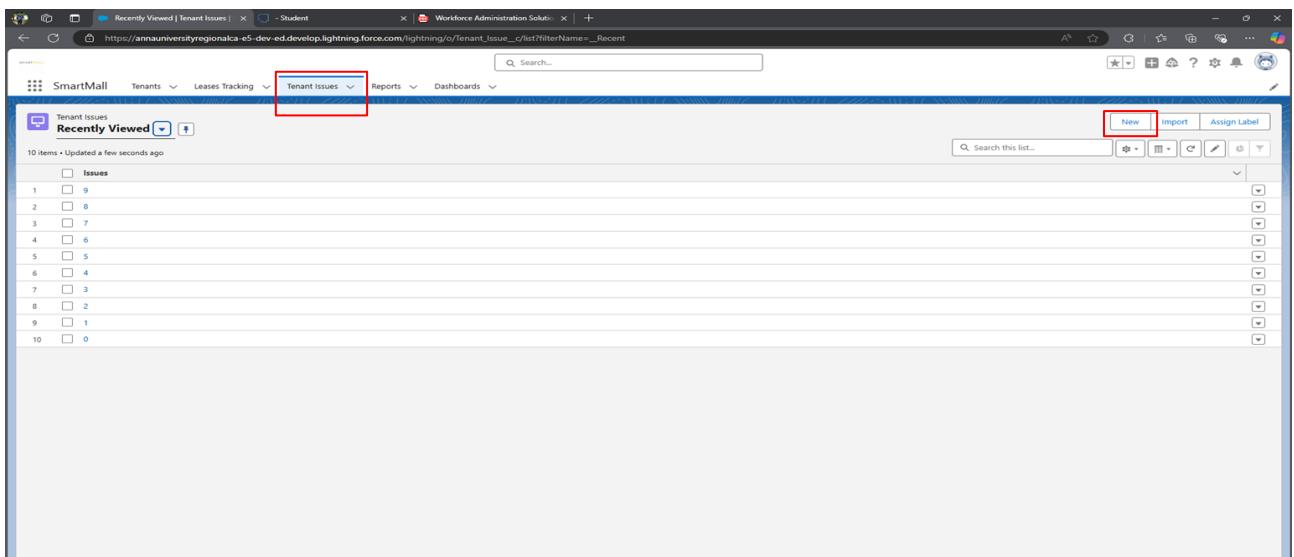
Leases Tracking
Recently Viewed

10 items • Updated a few seconds ago

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

Activity 3: Inserting Records in Tenants Issues object

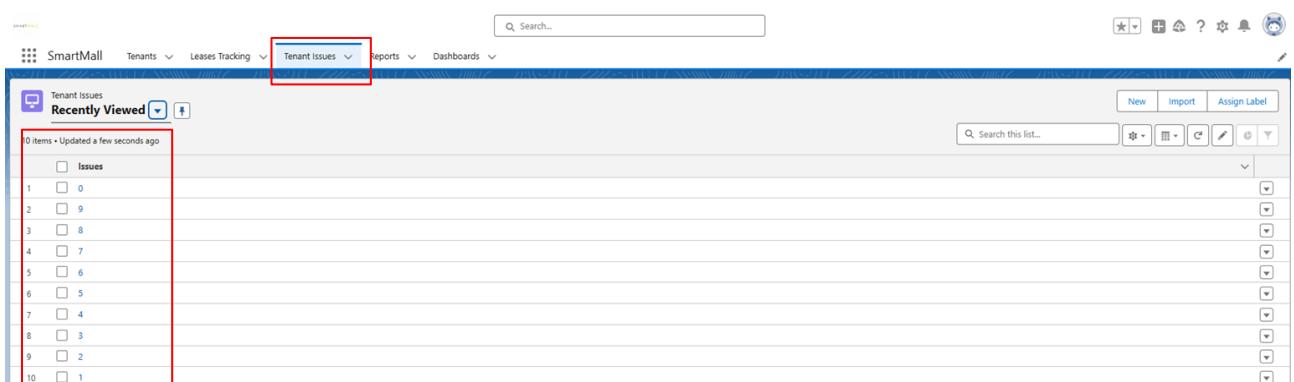
1. Click on the App Launcher and search Tenant Issues Object then click New in the right corner to create a record.
2. Fill every field with valid data.
3. Give each tenant related to each issue.
4. After creating a record the page will look like this



Edit 0

* = Required Information

Issues	0
Related tenant	Dinesh
*Issue Related to	
Available	Chosen
2) INFRASTRUCTURE 3) PLUMBING 4) RENT 5) OTHER	1) ELECTRICITY
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: ti1@gmail.com	
Cancel Save & New Save	



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. **Autolaunched 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.

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

- Setup** tab is selected in the top navigation bar.
- Flows** link under the **Process Automation** category is selected.
- New Flow** button in the top right corner of the main content area is highlighted with a red box.

The screenshot shows the "New Flow" creation page with the following details:

- Title:** New Flow
- Core** tab is selected in the top left.
- All + Templates** link is visible in the top left.
- Record-Triggered Flow** is selected and highlighted with a blue border, indicating it is the chosen type.
- Screen Flow**, **Schedule-Triggered Flow**, **Platform Event—Triggered Flow**, **Autolaunched Flow (No Trigger)**, and **Record-Triggered Orchestration** are listed below.
- Create** button is located in the bottom right corner.

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... View as Rich Text

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

Salesforce Sans ▾ 12 ▾ Font Color B I U Horizontal Line Vertical Line Table Image Text

Cancel Done

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

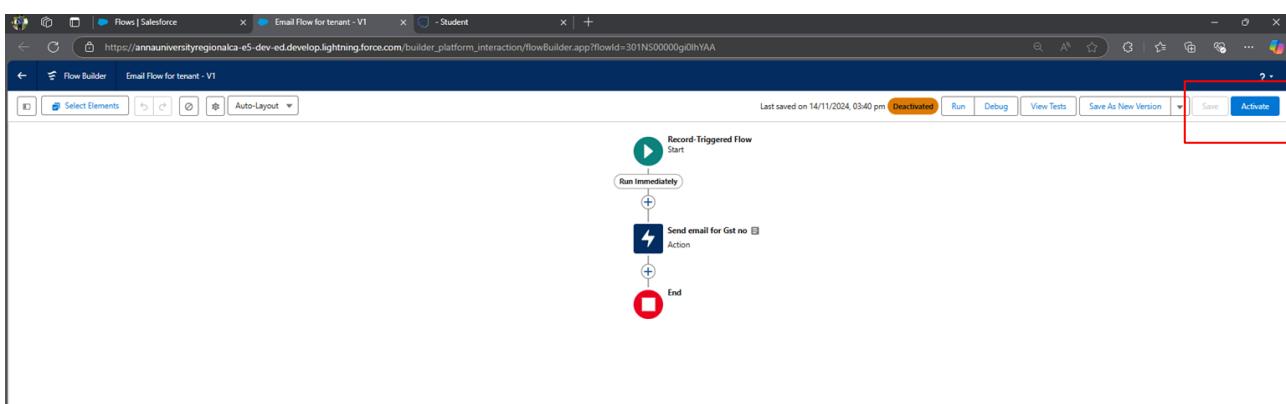
*Flow Label email flow

*Flow API Name email_flow

Description

Show Advanced

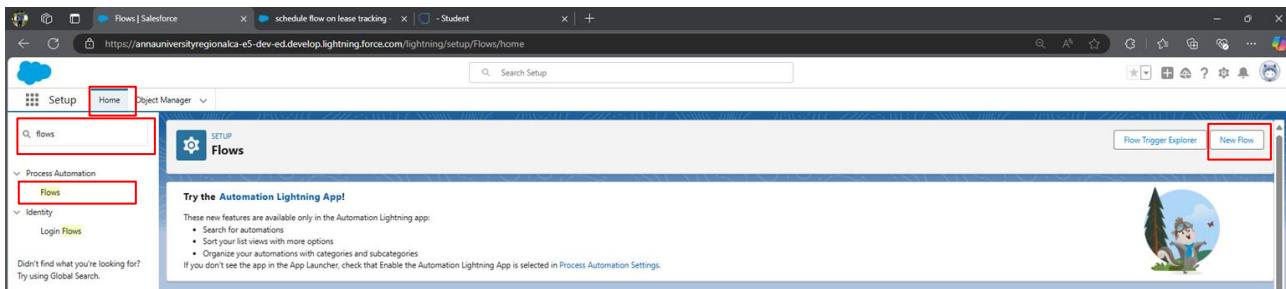
Cancel Save

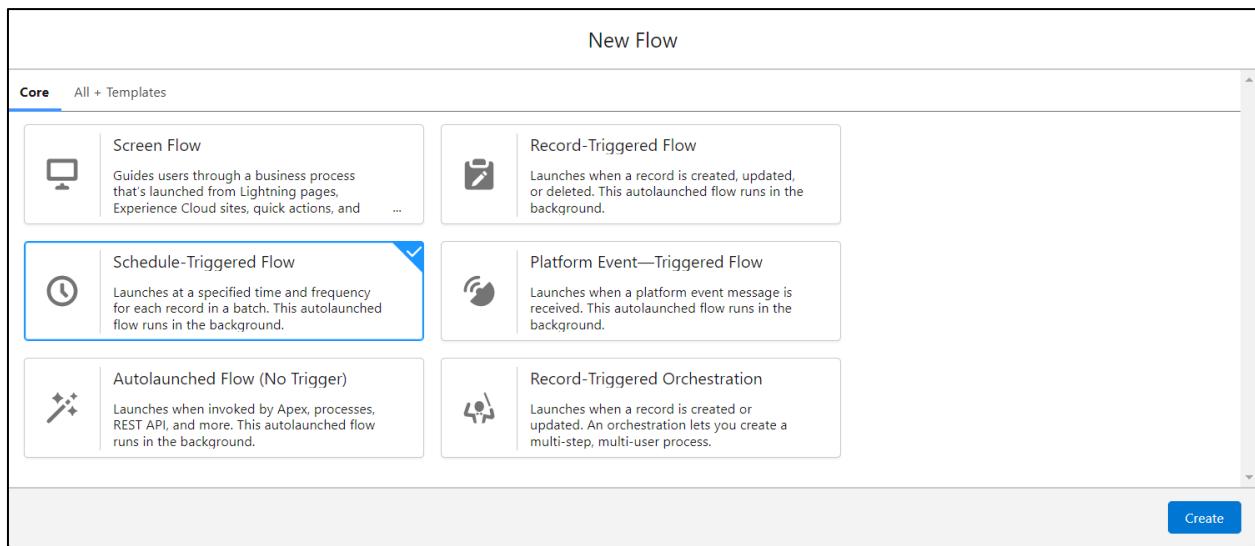


Activity 2: Create a Schedule Flow on Lease Management Object

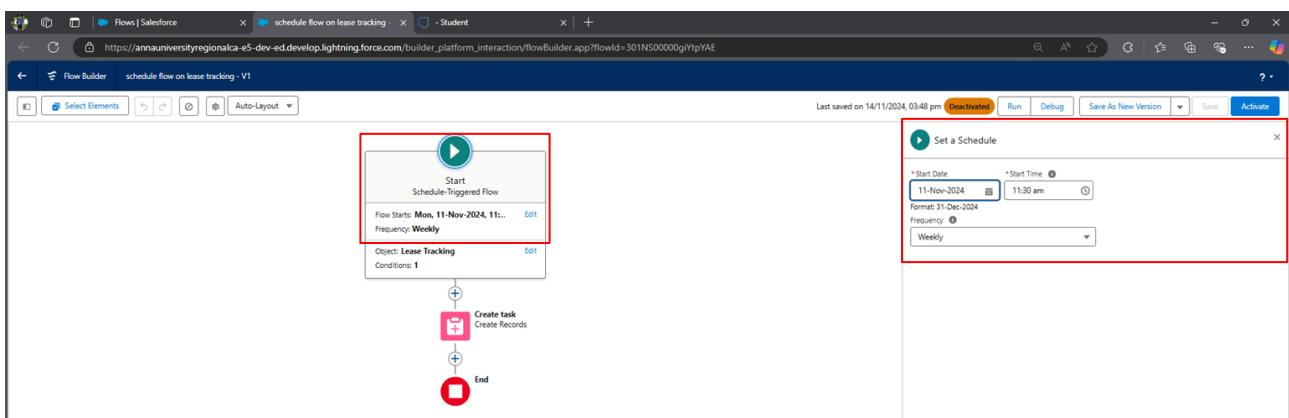
If the End Date is within the Next 1 year create a task to the Lease tracking weekly on every Monday.

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





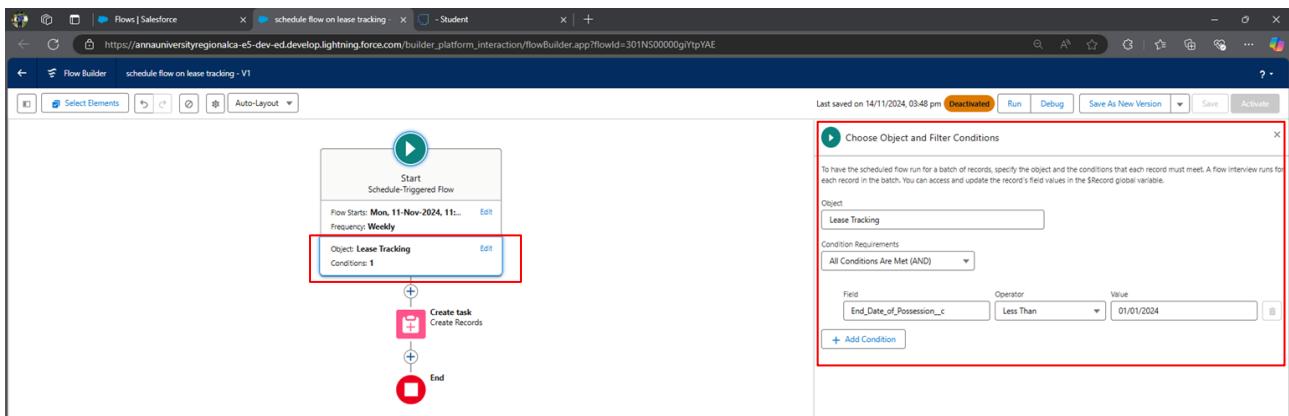
2. Set the Schedule Date - Any Monday, Time - 11.30, Frequency - Weekly.



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

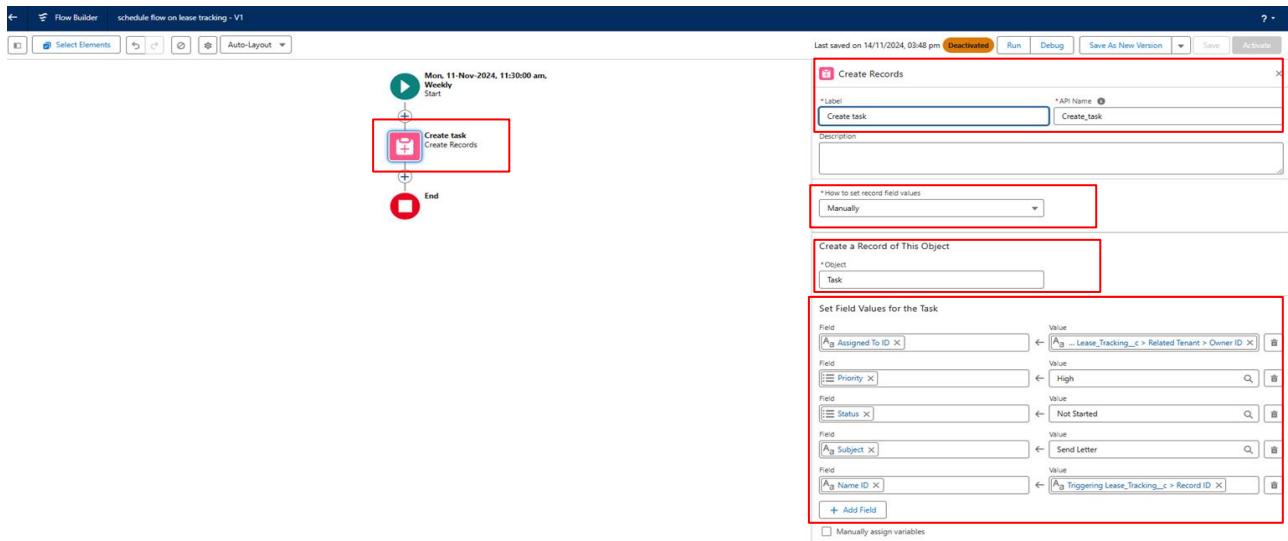
7. How many Records to create - one

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

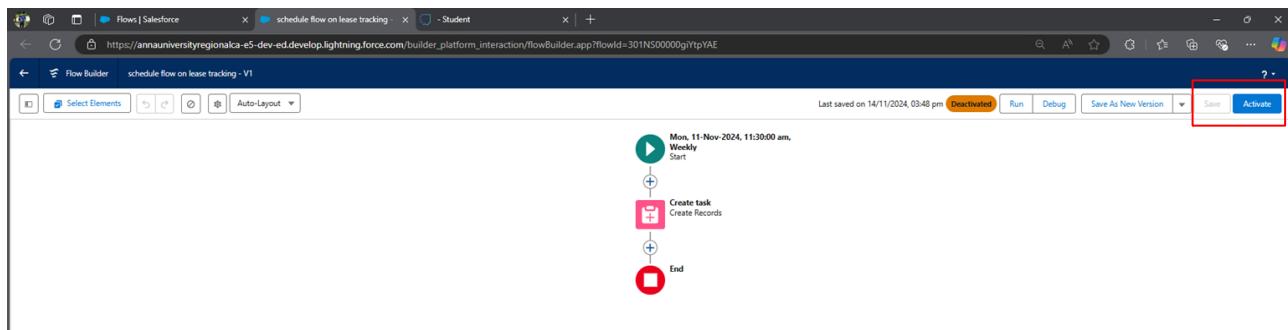
* Flow Label: schedule flow on lease tracking

* 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.

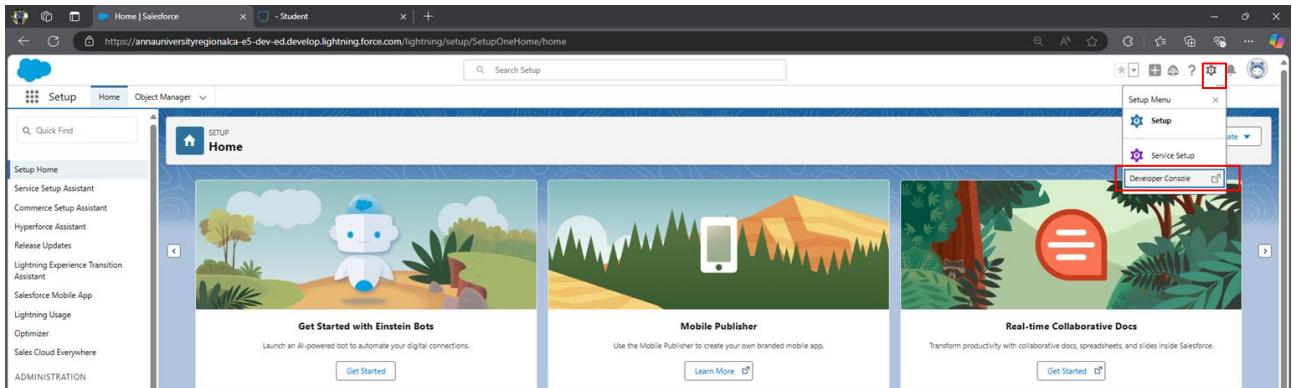
Enter the trigger name and the object to be triggered.

Use Case:

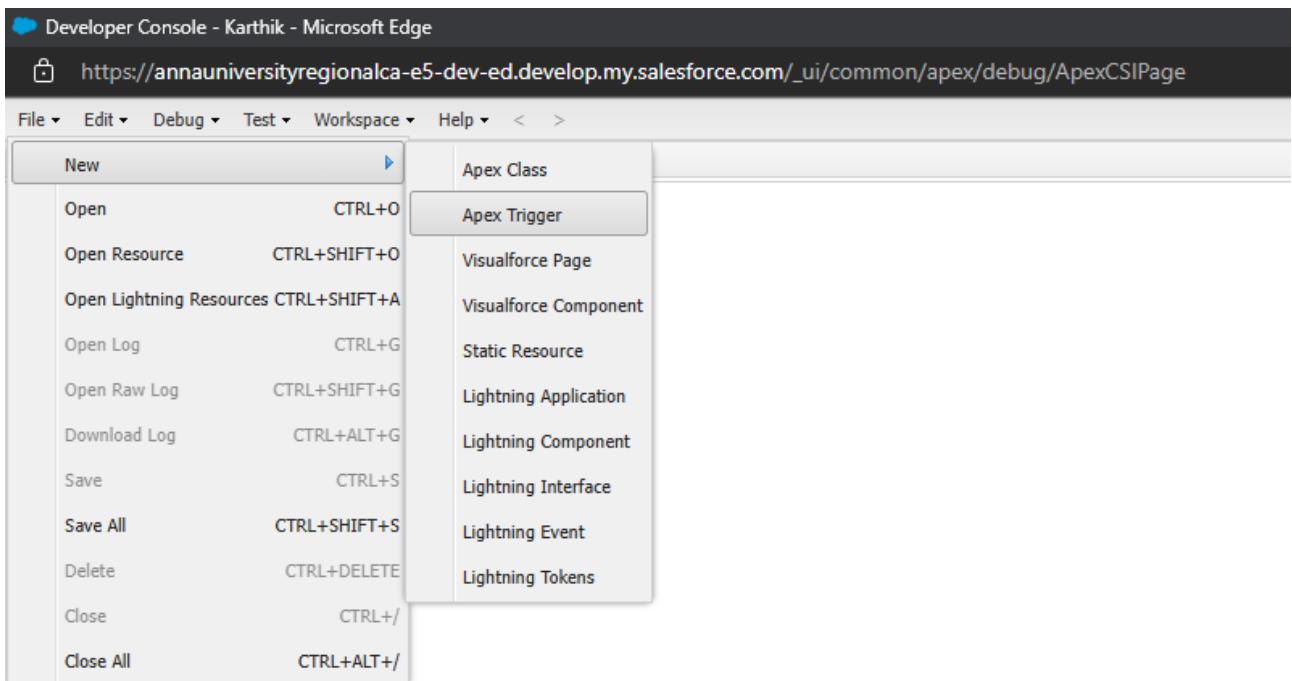
In Salesforce, Apex Triggers are powerful tools used to perform custom actions before or after specific database events, such as insertions, updates, or deletions on Salesforce objects. For instance, they can enforce business logic by automatically verifying data integrity and ensuring that no incomplete 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.

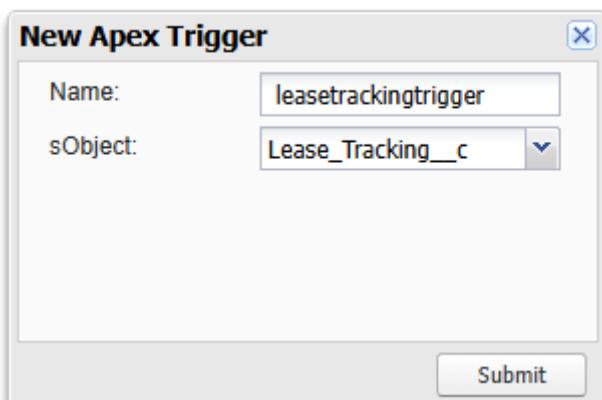
1. Click on the gear icon and click on the developer console.



2. Click on the file and select New Apex Trigger



3. Name- leasetrackingtrigger, Object —> Lease_Tracking__c



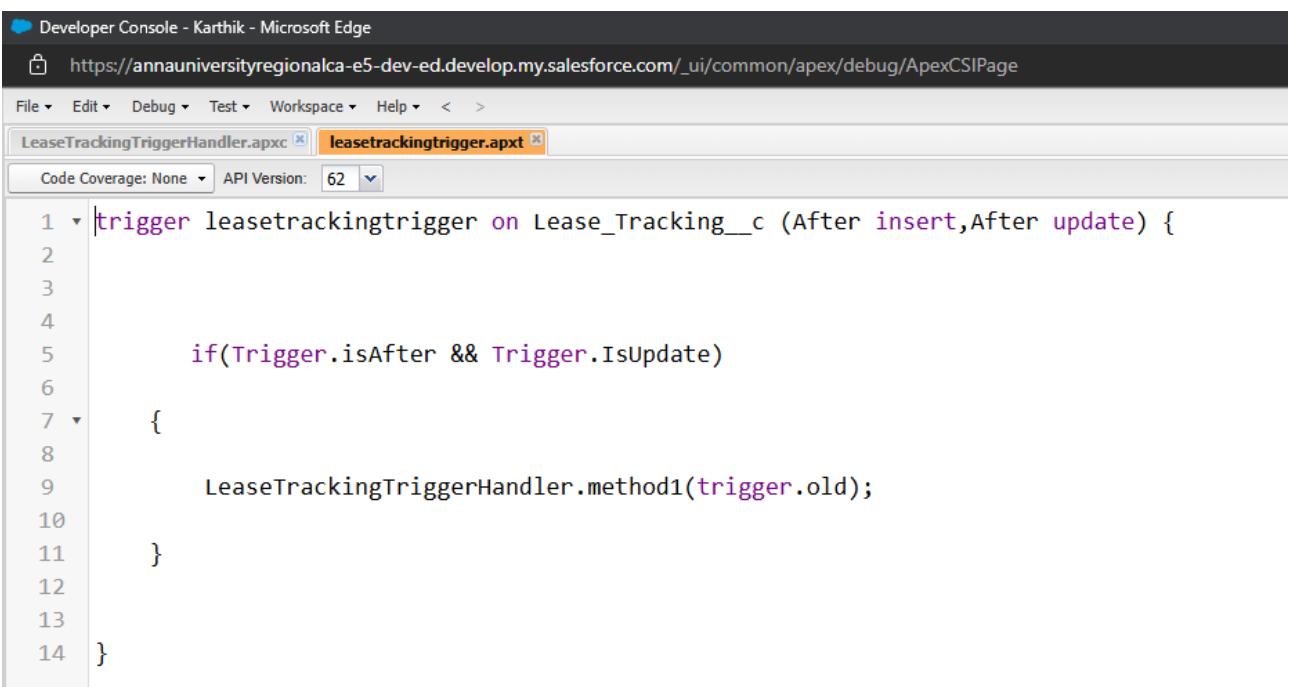
4. Use Event - After Insert and After Update and Use Trigger Context Variables as IsAfter and IsUpdate.

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 title is 'LeaseTrackingTriggerHandler.apxc' and the active tab is 'leasetrackingtrigger.apxt'. The code coverage 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);
}
}
}
}

```

The screenshot shows the Salesforce Developer Console interface. The top part displays the Apex code for the `LeaseTrackingTriggerHandler` class. The code defines a static method `method1` that iterates through a list of `Lease_Tracking__c` records. For each record where the `Amount_to_be_paid__c` is greater than half of the `Total_rent_Yearly__c`, it creates a `Messaging.SingleEmailMessage`, sets the recipient list, subject, and body, and then sends the email using `Messaging.sendEmail`. The bottom part of the screenshot shows the `Logs` tab, which is currently empty.

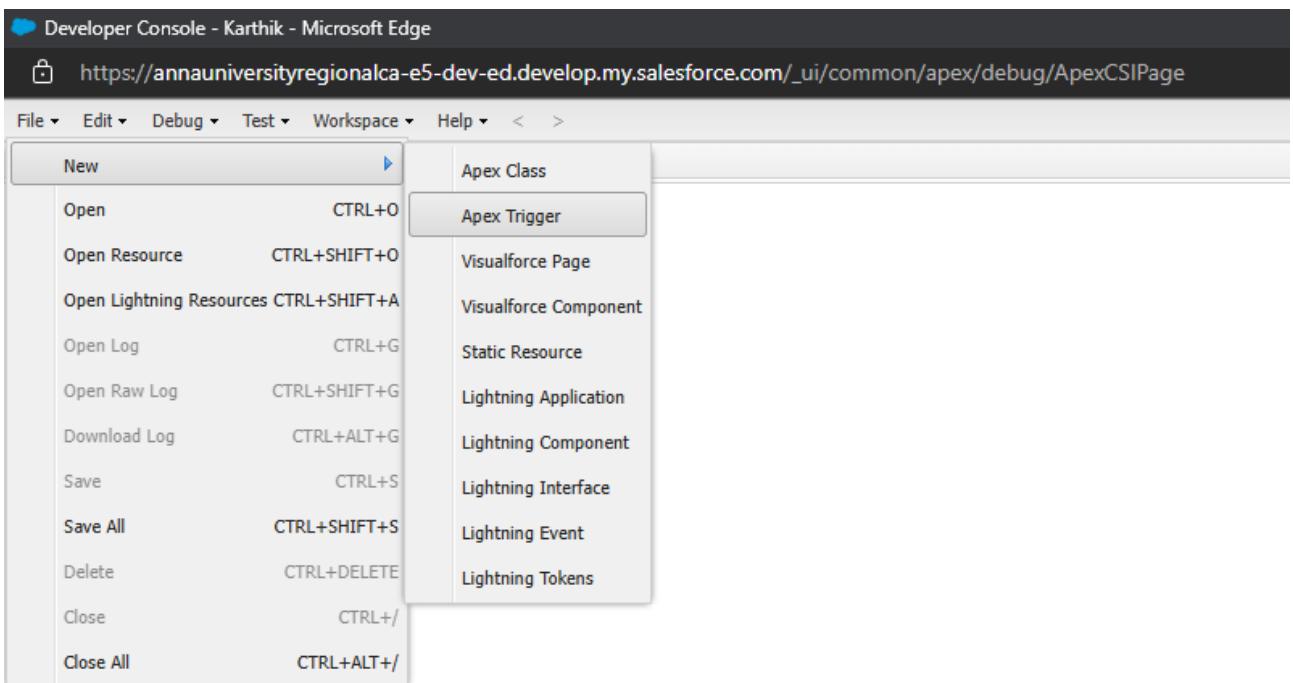
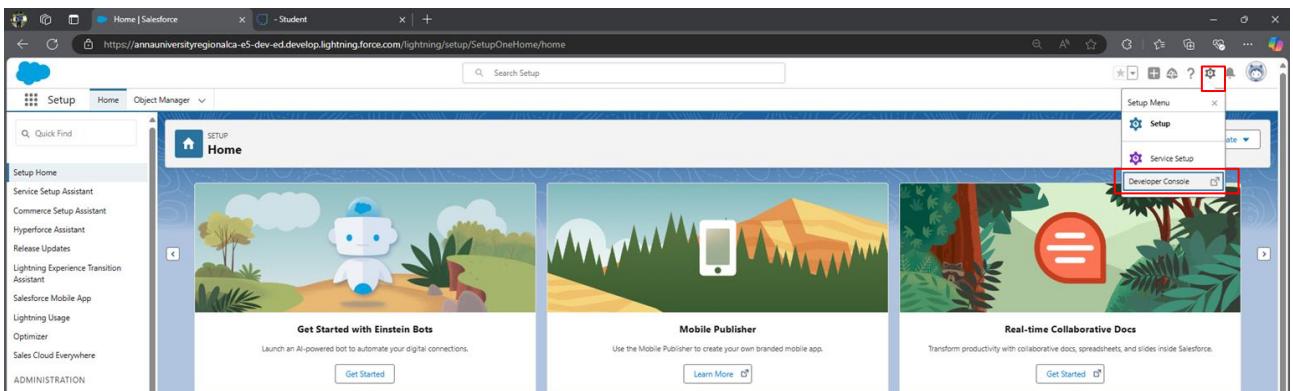
```

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

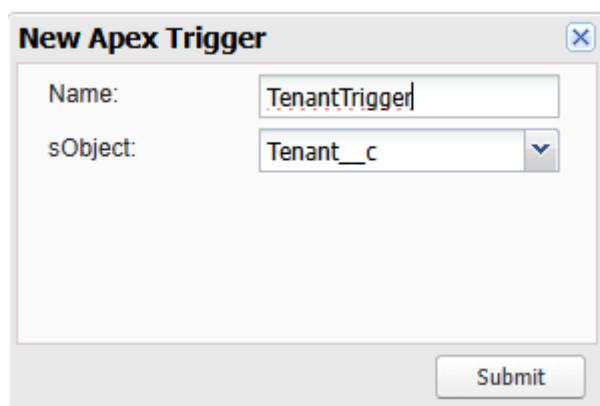
```

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

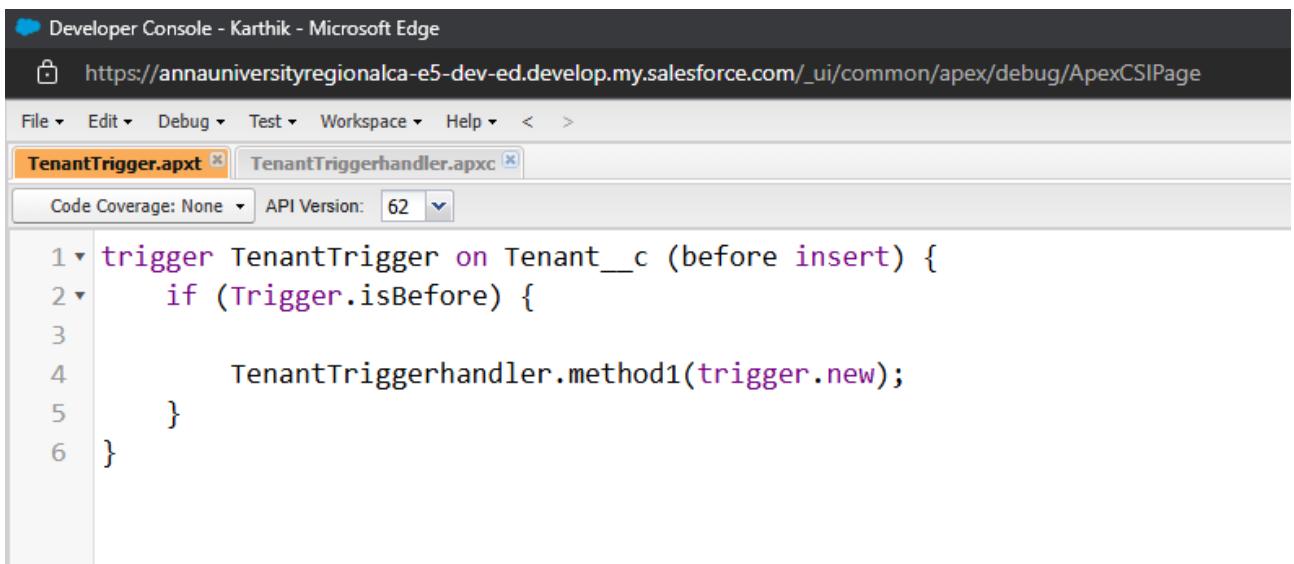


4. Use Events - Before insert and Trigger context Variable - IsBefore

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 tabs at the top are 'TenantTrigger.apxt' (selected) and 'TenantTriggerhandler.apxc'. Below the tabs, there are dropdown menus for 'File', 'Edit', 'Debug', 'Test', 'Workspace', and 'Help'. A status bar shows 'Code Coverage: None' and 'API Version: 62'. The main area contains the Apex trigger code:

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

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. At the top, 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 a dropdown for TenantTriggerHandler.apc. Below the tabs, there's a "Code Coverage: None" and "API Version: 62" dropdown, and a "Go To" button. The main area displays the Apex code for the TenantTriggerHandler class:

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

Below the code editor is a "Logs" tab, which is currently selected. The logs table has columns for User, Application, Operation, Time, Status, and Size. There are no logs listed in the table. At the bottom of the logs section, there is a "Filter" input field with the placeholder "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.

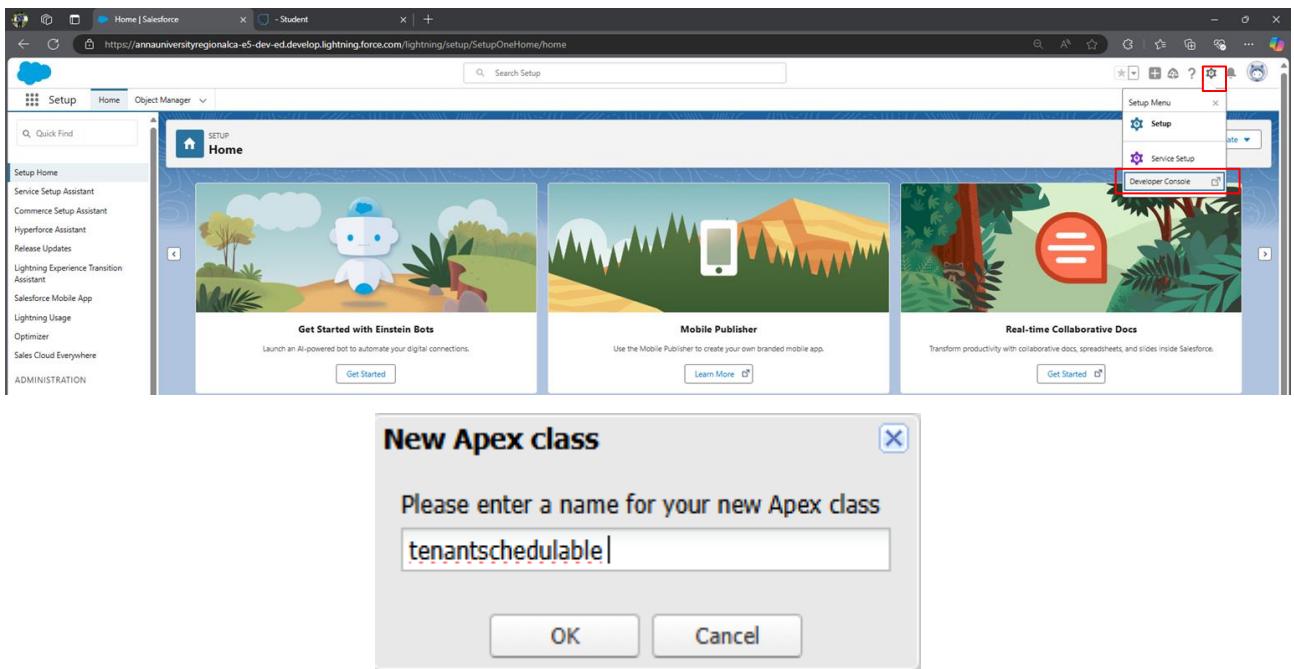
Use Case:

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
File Edit Debug Test Workspace Help https://annauniversityregionalca-e5-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

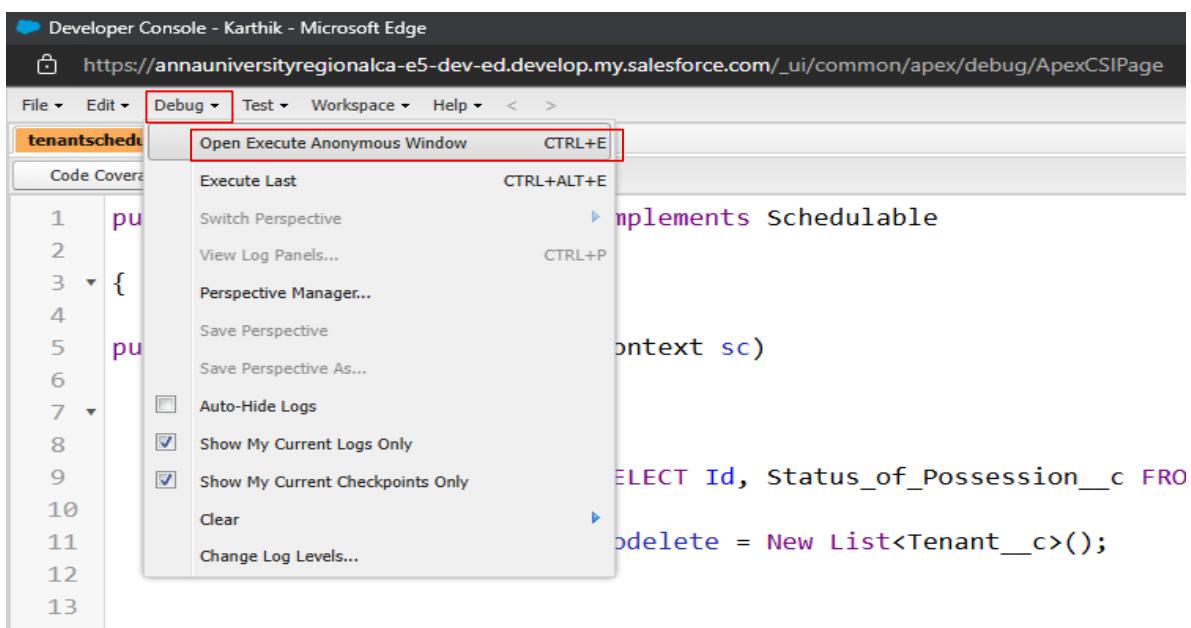
Code Coverage: None API Version: 62 Go To

tenantschedulable.apc

```
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    public void delete()
18    {
19        delete(tenantstodelete);
20    }
21
22 }
```

Logs Tests Checkpoints Query Editor View State Progress Problems 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 analyze 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 displays the SmartMall application interface. At the top, there is a navigation bar with various tabs like 'SmartMall', 'Tenants', 'Leases Tracking', 'Tenant Issues', 'Reports', and 'Dashboards'. A search bar is located at the top right. A context menu icon (pencil) is highlighted with a red box in the top right corner of the header.

A modal window titled 'Edit SmartMall App Navigation Items' is open. It contains a list of 'NAVIGATION ITEMS (5)' which include 'Tenants', 'Leases Tracking', 'Tenant Issues', 'Reports', and 'Dashboards'. There is a button labeled 'Add More Items' with a red box around it. Below the list are 'Cancel' and 'Save' buttons.

Below the main interface, another modal window titled 'Add Items' is open. On the left, there is a sidebar with sections 'Available Items', 'Favorites', and 'All' (which is selected). A search bar at the top of the list says 'Search all items...'. The list contains 21 items, each with a plus sign icon and a label: 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.

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

Report Type Name	Category
Accounts	Standard
Contacts & Accounts	Standard
Accounts with Partners	Standard
Account with Account Teams	Standard
Accounts with Contact Roles	Standard
Accounts with Assets	Standard
Contacts with Assets	Standard
Account History	Standard
Contact History	Standard
D&B Company with and without Accounts	Standard
Opportunities	Standard
Opportunities with Products	Standard
Opportunities with Contact Roles	Standard
Opportunities with Partners	Standard
Opportunities with Competitors	Standard
Opportunity History	Standard
Opportunity Field History	Standard
Opportunity Trends	Standard
Opportunities with Contact Roles and Products	Standard

Date of Reg	Tenant: Tenant Name	Lease Tracking: Lease Tracking No	Amount to be paid	Date of Possession
10/11/2024 (1)	Karthik	TT - 000009	2,00,000.00	01/07/2025
			2,00,000.00	
Subtotal				
11/11/2024 (9)	Sita	TT - 000007	2,75,000.00	01/04/2025
	Ganesh	TT - 000010	1,00,000.00	01/08/2025
	Suresh	TT - 000002	2,25,000.00	31/01/2025
	Ramesh	TT - 000003	1,55,000.00	01/02/2025
	Lakshman	TT - 000005	2,32,000.00	01/03/2025
	Siva	TT - 000008	1,80,000.00	01/06/2025
	Tamilselvan	TT - 000004	2,35,000.00	01/03/2025
	Dinesh	TT - 000001	1,50,000.00	01/11/2025
	Ram	TT - 000006	2,00,000.00	01/05/2025
Subtotal			17,52,000.00	
Total (10)			19,52,000.00	

- 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 two overlapping dialogs on a Salesforce interface:

- Create folder Dialog:** This dialog is titled "Create folder". It contains two input fields: "Folder Label" and "Folder Unique Name", both of which are currently empty. At the bottom are "Cancel" and "Save" buttons. The entire dialog is highlighted with a red box.
- Create Report Dialog:** This dialog is titled "Create Report". On the left is a sidebar with "Category" sections: "Recently Used" (with "All" selected), "Accounts & Contacts", "Opportunities", "Customer Support Reports", "Leads", "Campaigns", "Activities", "Contracts and Orders", "Price Books, Products and Assets", "Administrative Reports", "File and Content Reports", "Individuals", and "Other Reports". Below this is a section for "Hidden Report Types". On the right is a main table titled "Select a Report Type" with columns for "Report Type Name", "Category", and a dropdown menu. The table lists various report types such as Accounts, Contacts & Accounts, Accounts with Partners, etc., each with a standard category and a dropdown menu. The entire "Create Report" dialog is also highlighted with a red box.

- Click On new report ==> Select object Activities with Tenant issue==> Click on start report
- 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

DASHBOARDS	Dashboard Name	Description	Folder	Created By	Created On	Subscribed
Recent	December Dashboard		Mall Dashboard	Paranjithi Karthik M	12/11/2024, 12:22 pm	
Created by Me						
Private Dashboards						

- 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.

December Dashboard | Salesforce - Student

https://annauniversityregionalca-e5-dev-ed.lightning.force.com/lightning/r/Dashboard/01ZNS00000150pJ2AS/view?queryScope=userFolders

SmartMall Tenants Leases Tracking Tenant Issues Reports Dashboards

December Dashboard

As of 14-Nov-2024, 8:06 pm Viewing as Parangudi Karthik M

Refresh Edit Subscribe

Tenant Details

Date of Reg > Status of possession

Pending Hand Overed Renewal Needed Closed 10/11/2024 11/11/2024 Record Count 9

View Report (Tenant Details)

Tenant Issues Report

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

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	01/06/2025
Karthik	TT - 000009	200.00k	01/07/2025
Lakshman	TT - 000005	232.00k	01/02/2025
Ram	TT - 000006	200.00k	01/05/2025
Ramesh	TT - 000003	155.00k	01/02/2025
Sita	TT - 000007	275.00k	01/04/2025

View Report (Leases Tracking Report)