# PROJECT EXECUTABLE FILES

## APEX TRIGGERS

## Use case:

The tenant and property are in a master-detail relationship, wherein each tenant is associated with only one property. When a tenant attempts to create a new record with an existing property, an error should be displayed, indicating that a tenant can have only one property.

Write a code to achieve this requirement using Salesforce developer skills to fulfill the Managers requirement.

## Create an Apex Trigger

To create a new Apex Class follow the below steps:Click on the file >> New ? Apex ClassA screenshot of a computer

AI-generated content may be incorrect.

Give the Apex Trigger name as “test”, and select “Tenant\_\_c” from the dropdown for sObject.

A screenshot of a computer

AI-generated content may be incorrect.

Click Submit.

Now write the code logic here

A screenshot of a computer

AI-generated content may be incorrect.

Trigger Code:

trigger test on Tenant\_\_c (before insert)

{

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

        testHandler.preventInsert(trigger.new);

    }

}

# 

# Apex Handler class

To create a new Apex Class follow the below steps:

Click on the file >> New >>Apex Class.

2. Enter class name as testHandler.

A screenshot of a computer code

AI-generated content may be incorrect.

Apex logic:

public class testHandler {

   public static void preventInsert(List<Tenant\_\_c> newlist) {

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

        for (Tenant\_\_c existingTenant : [SELECT Id, Property\_\_c FROM Tenant\_\_c WHERE Property\_\_c != null]) {

            existingPropertyIds.add(existingTenant.Property\_\_c);

        }

        for (Tenant\_\_c newTenant : newlist) {

            if (newTenant.Property\_\_c != null && existingPropertyIds.contains(newTenant.Property\_\_c)) {

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

            }

        }

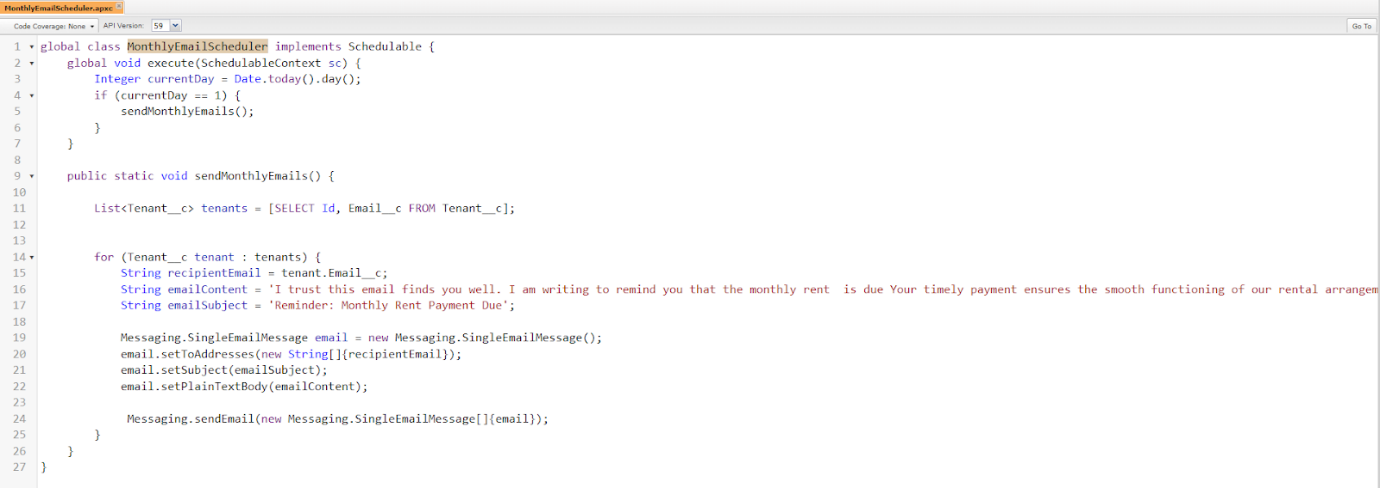
    }

}

APEX CLASS

1. To create a new Apex Class follow the below steps:  
Click on the file >> New >> Apex Class.

2. Enter class name as MonthlyEmailScheduler.



Apex logic:

global class MonthlyEmailScheduler implements Schedulable {

    global void execute(SchedulableContext sc) {

        Integer currentDay = Date.today().day();

        if (currentDay == 1) {

            sendMonthlyEmails();

        }

    }

    public static void sendMonthlyEmails() {

        List<Tenant\_\_c> tenants = [SELECT Id, Email\_\_c FROM Tenant\_\_c];

        for (Tenant\_\_c tenant : tenants) {

            String recipientEmail = tenant.Email\_\_c;

            String emailContent = 'I trust this email finds you well. I am writing to remind you that the monthly rent  is due Your timely payment ensures the smooth functioning of our rental arrangement and helps maintain a positive living environment for all.';

            String emailSubject = 'Reminder: Monthly Rent Payment Due';

            Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();

            email.setToAddresses(new String[]{recipientEmail});

            email.setSubject(emailSubject);

            email.setPlainTextBody(emailContent);

             Messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});

        }

    }

}