**APEX TRIGGERS**

**Trigger Code:**

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

{

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

testHandler.preventInsert(trigger.new);

}

}

**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');

}

}

}

}

**SCHEDULE APEX CLASS**

**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});

}

}

}