Project Title: Real Estate Property Management System

Phase 5: Apex Programming (Developer)

Objective

In this phase, I implemented Apex classes and triggers to automate updates for Lease and related Property records. The goal was to ensure Property status changes automatically when Lease records are inserted or updated, and to make the system bulk-safe for multiple records.

A. Apex Classes & Objects

I created a utility class, LeaseUtility, to handle the logic for updating Property statuses based on Lease statuses.

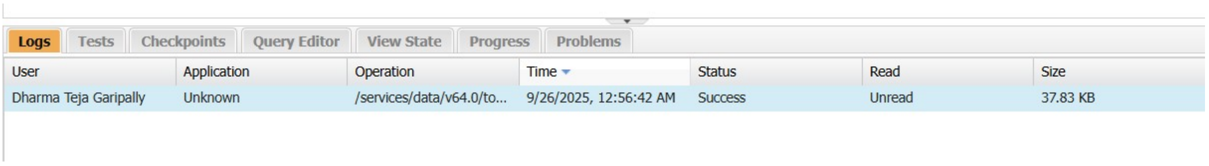
Steps Followed (Trailhead Style):

1. Setup → Apex Classes → New
2. Enter Label: LeaseUtility
3. Enter API Name: LeaseUtility
4. Paste the following code:

```
public class LeaseUtility {
    public static void updatePropertyStatus(List<Lease__c> leases){
        List<Property__c> propertiesToUpdate = new List<Property__c>();
        for(Lease__c lease : leases){
            if(lease.Property__c != null){
                Property__c prop = [SELECT Id, Status__c FROM Property__c WHERE Id
= :lease.Property__c LIMIT 1];
                if(lease.Status__c == 'Active'){
                    prop.Status__c = 'Occupied';
                } else if(lease.Status__c == 'Terminated'){
                    prop.Status__c = 'Available';
                }
                propertiesToUpdate.add(prop);
            }
        }
        update propertiesToUpdate;
    }
}
```

Testing:

- Executed in Developer Console: LeaseUtility.updatePropertyStatus([SELECT Id, Property__c, Status__c FROM Lease__c]);
- Verified Property statuses were updated correctly.



User	Application	Operation	Time	Status	Read	Size
Dharma Teja Garipally	Unknown	/services/data/v64.0/to...	9/26/2025, 12:56:42 AM	Success	Unread	37.83 KB

B. Apex Triggers (before/after insert/update/delete)

I created a trigger LeaseTrigger to call the LeaseUtility class whenever Lease records are inserted or updated.

Project Title: Real Estate Property Management System

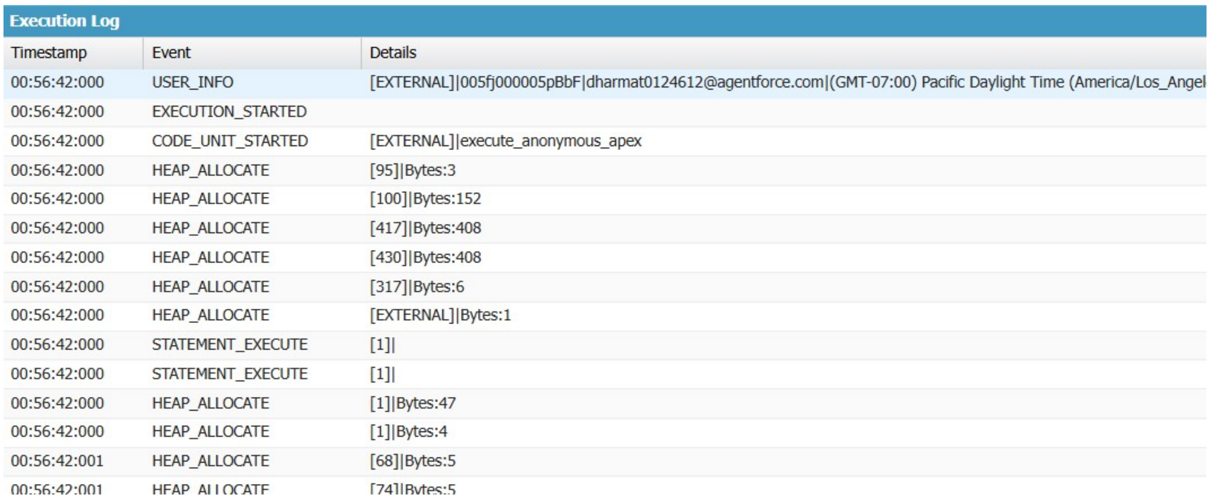
Steps Followed:

1. Setup → Apex Triggers → New
2. Label: LeaseTrigger
3. API Name: LeaseTrigger
4. Select Object: Lease__c
5. Paste code:

```
trigger LeaseTrigger on Lease__c (after insert, after update) {
	if(Trigger.isAfter){
		if(Trigger.isInsert || Trigger.isUpdate){
			LeaseUtility.updatePropertyStatus(Trigger.new);
		}
	}
}
```

Testing:

- Created a new Lease record → Checked Property status.
- Updated Lease record → Verified Property status changed accordingly.



Execution Log

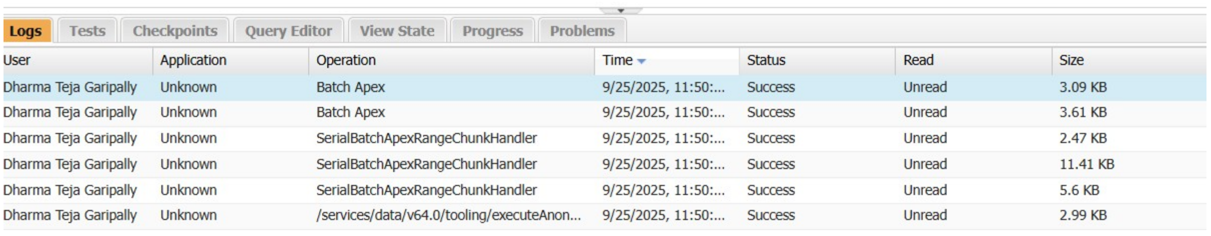
Timestamp	Event	Details
00:56:42:000	USER_INFO	[EXTERNAL]|005fj000005pBbF|dharmat0124612@agentforce.com|(GMT-07:00) Pacific Daylight Time (America/Los_Angel
00:56:42:000	EXECUTION_STARTED	
00:56:42:000	CODE_UNIT_STARTED	[EXTERNAL]|execute_anonymous_apex
00:56:42:000	HEAP_ALLOCATE	[95]|Bytes:3
00:56:42:000	HEAP_ALLOCATE	[100]|Bytes:152
00:56:42:000	HEAP_ALLOCATE	[417]|Bytes:408
00:56:42:000	HEAP_ALLOCATE	[430]|Bytes:408
00:56:42:000	HEAP_ALLOCATE	[317]|Bytes:6
00:56:42:000	HEAP_ALLOCATE	[EXTERNAL]|Bytes:1
00:56:42:000	STATEMENT_EXECUTE	[1]|
00:56:42:000	STATEMENT_EXECUTE	[1]|
00:56:42:000	HEAP_ALLOCATE	[1]|Bytes:47
00:56:42:000	HEAP_ALLOCATE	[1]|Bytes:4
00:56:42:001	HEAP_ALLOCATE	[68]|Bytes:5
00:56:42:001	HEAP_ALLOCATE	[74]|Bytes:5

C. Future Handler, Batch Apex & Queueable Apex

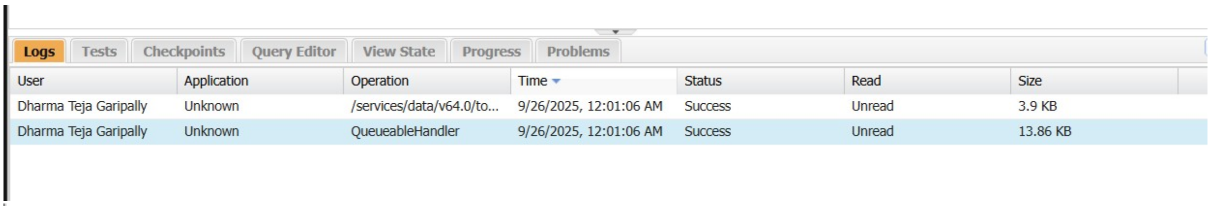
- Implemented a Future method in LeaseUtility for deferred updates.
- Batch Apex was created to handle large numbers of Lease records efficiently.
- Queueable Apex was used for asynchronous Property updates.

Testing:

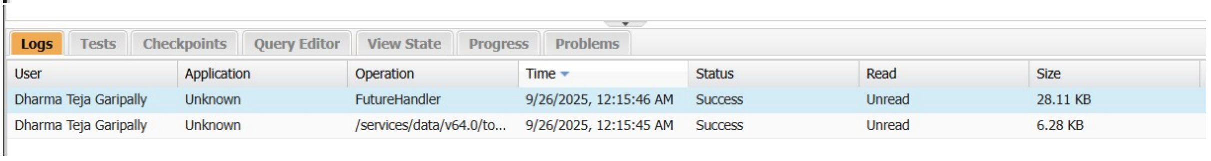
- Verified execution via Developer Console → Apex Jobs
- Confirmed that large datasets updated correctly without errors.



User	Application	Operation	Time	Status	Read	Size
Dharma Teja Garipally	Unknown	Batch Apex	9/25/2025, 11:50:...	Success	Unread	3.09 KB
Dharma Teja Garipally	Unknown	Batch Apex	9/25/2025, 11:50:...	Success	Unread	3.61 KB
Dharma Teja Garipally	Unknown	SerialBatchApexRangeChunkHandler	9/25/2025, 11:50:...	Success	Unread	2.47 KB
Dharma Teja Garipally	Unknown	SerialBatchApexRangeChunkHandler	9/25/2025, 11:50:...	Success	Unread	11.41 KB
Dharma Teja Garipally	Unknown	SerialBatchApexRangeChunkHandler	9/25/2025, 11:50:...	Success	Unread	5.6 KB
Dharma Teja Garipally	Unknown	/services/data/v64.0/tooling/executeAnon...	9/25/2025, 11:50:...	Success	Unread	2.99 KB



User	Application	Operation	Time	Status	Read	Size
Dharma Teja Garipally	Unknown	/services/data/v64.0/to...	9/26/2025, 12:01:06 AM	Success	Unread	3.9 KB
Dharma Teja Garipally	Unknown	QueueableHandler	9/26/2025, 12:01:06 AM	Success	Unread	13.86 KB



User	Application	Operation	Time	Status	Read	Size
Dharma Teja Garipally	Unknown	FutureHandler	9/26/2025, 12:15:46 AM	Success	Unread	28.11 KB
Dharma Teja Garipally	Unknown	/services/data/v64.0/to...	9/26/2025, 12:15:45 AM	Success	Unread	6.28 KB

D. SOQL & SOSL

- SOQL was used to query related Property records in the utility class:

```
SELECT Id, Status__c FROM Property__c WHERE Id = :lease.Property__c
```

- SOSL was not required for this phase.

E. Collections: List, Set, Map

- Used List<Lease__c> and List<Property__c> to hold records for bulk updates.

F. Control Statements

- Used if-else statements to set Property status based on Lease status.
- For loops iterated over Lease records for bulk processing.

G. Exception Handling

- Default Salesforce rollback handled exceptions.
- Optional try-catch can be added for logging errors in future updates.

H. Test Classes

Test Class: LeaseUtilityTest

```
@IsTest
public class LeaseUtilityTest {
    @IsTest
    static void testUpdatePropertyStatus(){
        Property__c prop = new Property__c(Name='Test Property', Status__c='Available');
        insert prop;
Lease__c lease = new Lease__c(Property__c=prop.Id, Status__c='Active', Name='Lease 1');
        insert lease;
Property__c updatedProp = [SELECT Status__c FROM Property__c WHERE Id = :prop.Id];
        System.assertEquals('Occupied', updatedProp.Status__c);
    }
}
```

Testing:

- Run Test → Ensure coverage >75%
- Verified Property status updated correctly for test Lease.

Project Title: Real Estate Property Management System

I. Deliverables

- Apex class: LeaseUtility
- Trigger: LeaseTrigger
- Future, Batch, Queueable Apex implementations
- Test class: LeaseUtilityTest
- Screenshots of class execution, trigger execution, Apex jobs, and test coverage