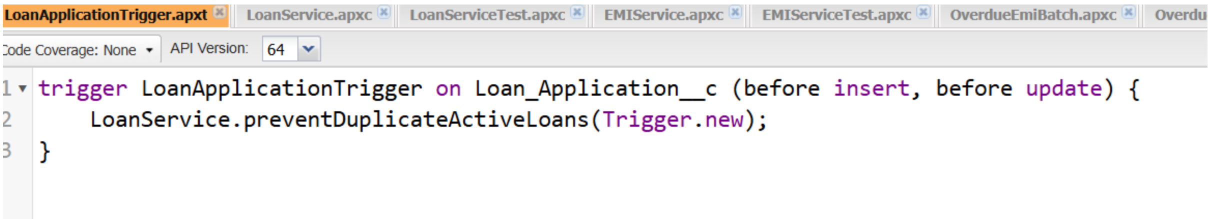
Phase 5 Apex Programming

A) Trigger & Handler – Prevent Duplicate Active Loans

- Prevents creating/updating multiple active loans for the same customer.
- Bulk-safe; handles multiple loan applications at once.

Trigger (LoanApplicationTrigger.trigger)



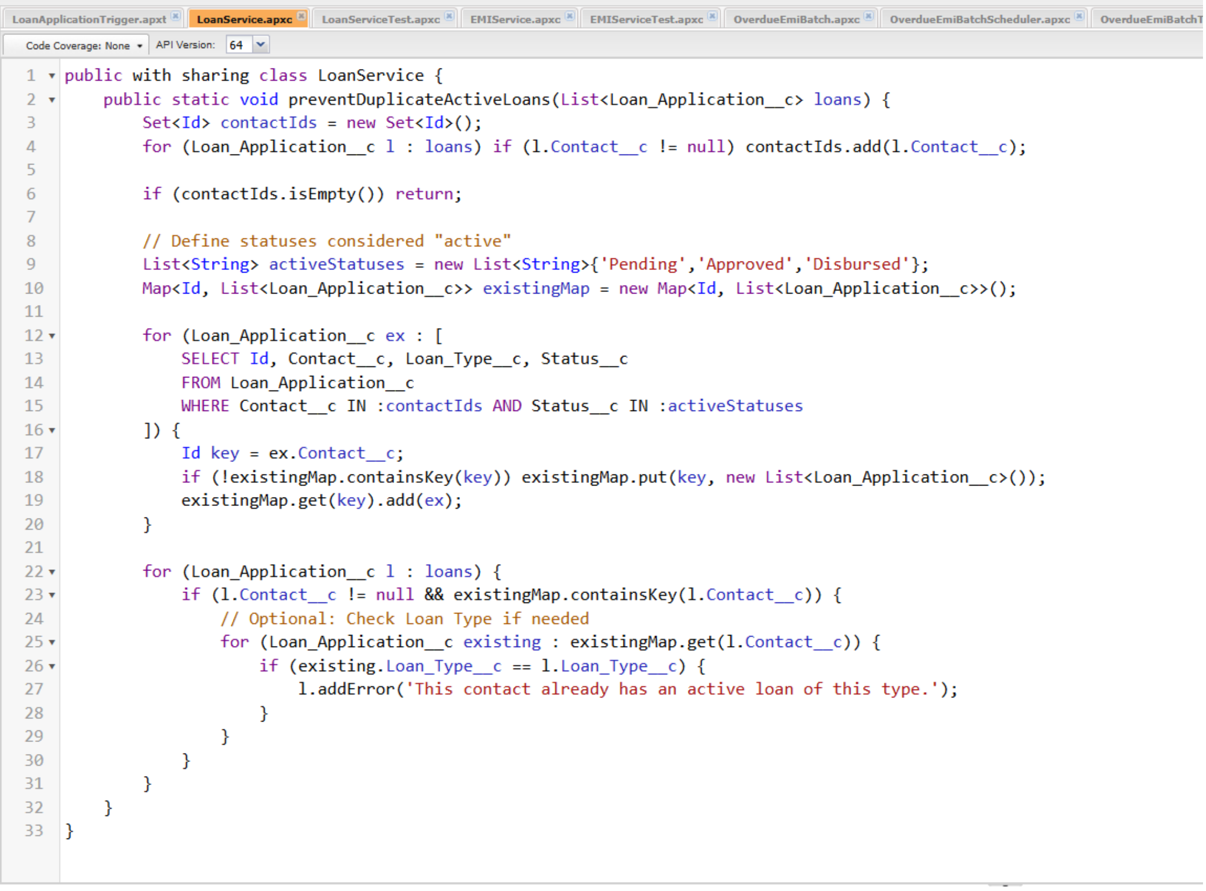
Code Coverage: None API Version: 64

```
trigger LoanApplicationTrigger on Loan_Application__c (before insert, before update) {
    LoanService.preventDuplicateActiveLoans(Trigger.new);
}
```

Handler (LoanApplicationHandler.cls)

Prevents creating/updating multiple active loans for the same customer.

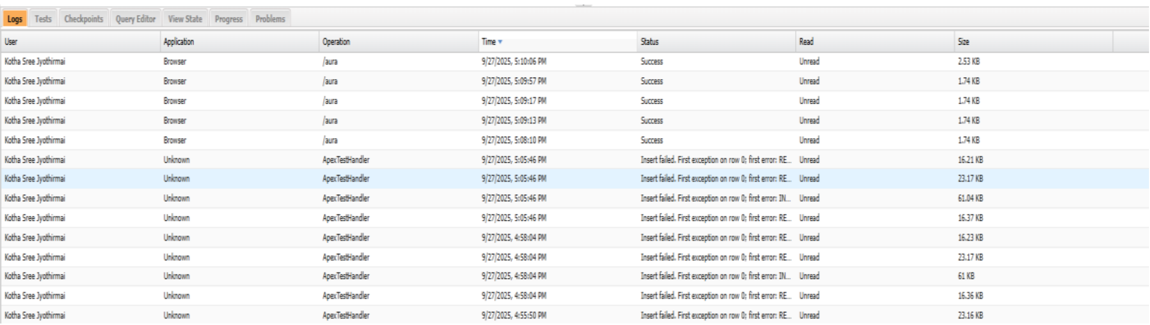
Bulk-safe; handles multiple loan applications at once.



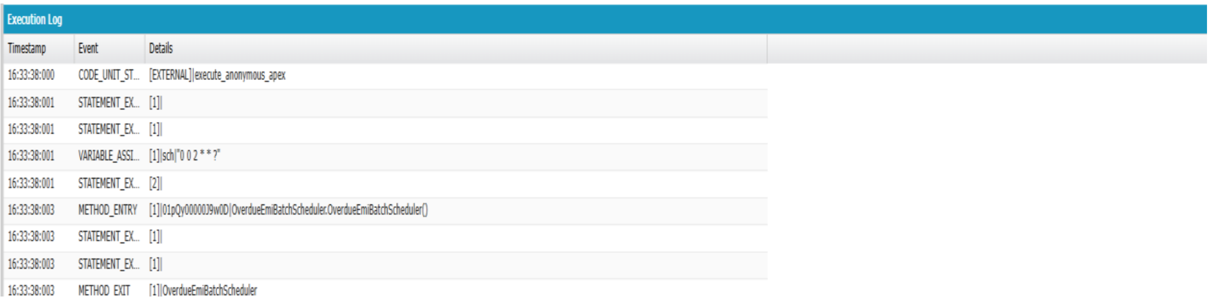
Code Coverage: None API Version: 64

```
public with sharing class LoanService {
    public static void preventDuplicateActiveLoans(List<Loan_Application__c> loans) {
        Set<Id> contactIds = new Set<Id>();
        for (Loan_Application__c l : loans) if (l.Contact__c != null) contactIds.add(l.Contact__c);

        if (contactIds.isEmpty()) return;

        // Define statuses considered "active"
        List<String> activeStatuses = new List<String>{'Pending','Approved','Disbursed'};
        Map<Id, List<Loan_Application__c>> existingMap = new Map<Id, List<Loan_Application__c>>();

        for (Loan_Application__c ex : [
            SELECT Id, Contact__c, Loan_Type__c, Status__c
            FROM Loan_Application__c
            WHERE Contact__c IN :contactIds AND Status__c IN :activeStatuses
        ]) {
            Id key = ex.Contact__c;
            if (!existingMap.containsKey(key)) existingMap.put(key, new List<Loan_Application__c>());
            existingMap.get(key).add(ex);
        }

        for (Loan_Application__c l : loans) {
            if (l.Contact__c != null && existingMap.containsKey(l.Contact__c)) {
                // Optional: Check Loan Type if needed
                for (Loan_Application__c existing : existingMap.get(l.Contact__c)) {
                    if (existing.Loan_Type__c == l.Loan_Type__c) {
                        l.addError('This contact already has an active loan of this type.');
                    }
                }
            }
        }
    }
}
```

Output:



User	Application	Operation	Time ▾	Status	Read	Size
Kotha Sree Jyothirmai	Browser	/aura	9/27/2025, 5:10:06 PM	Success	Unread	2.53 KB
Kotha Sree Jyothirmai	Browser	/aura	9/27/2025, 5:09:57 PM	Success	Unread	1.74 KB
Kotha Sree Jyothirmai	Browser	/aura	9/27/2025, 5:09:17 PM	Success	Unread	1.74 KB
Kotha Sree Jyothirmai	Browser	/aura	9/27/2025, 5:09:13 PM	Success	Unread	1.74 KB
Kotha Sree Jyothirmai	Browser	/aura	9/27/2025, 5:08:10 PM	Success	Unread	1.74 KB
Kotha Sree Jyothirmai	Unknown	ApexTestHandler	9/27/2025, 5:05:46 PM	Insert failed. First exception on row 0; first error: RE...	Unread	16.21 KB
Kotha Sree Jyothirmai	Unknown	ApexTestHandler	9/27/2025, 5:05:46 PM	Insert failed. First exception on row 0; first error: RE...	Unread	23.17 KB
Kotha Sree Jyothirmai	Unknown	ApexTestHandler	9/27/2025, 5:05:46 PM	Insert failed. First exception on row 0; first error: IN...	Unread	61.04 KB
Kotha Sree Jyothirmai	Unknown	ApexTestHandler	9/27/2025, 5:05:46 PM	Insert failed. First exception on row 0; first error: RE...	Unread	16.37 KB
Kotha Sree Jyothirmai	Unknown	ApexTestHandler	9/27/2025, 4:58:04 PM	Insert failed. First exception on row 0; first error: RE...	Unread	16.23 KB
Kotha Sree Jyothirmai	Unknown	ApexTestHandler	9/27/2025, 4:58:04 PM	Insert failed. First exception on row 0; first error: RE...	Unread	23.17 KB
Kotha Sree Jyothirmai	Unknown	ApexTestHandler	9/27/2025, 4:58:04 PM	Insert failed. First exception on row 0; first error: IN...	Unread	61 KB
Kotha Sree Jyothirmai	Unknown	ApexTestHandler	9/27/2025, 4:58:04 PM	Insert failed. First exception on row 0; first error: RE...	Unread	16.36 KB
Kotha Sree Jyothirmai	Unknown	ApexTestHandler	9/27/2025, 4:55:50 PM	Insert failed. First exception on row 0; first error: RE...	Unread	23.16 KB

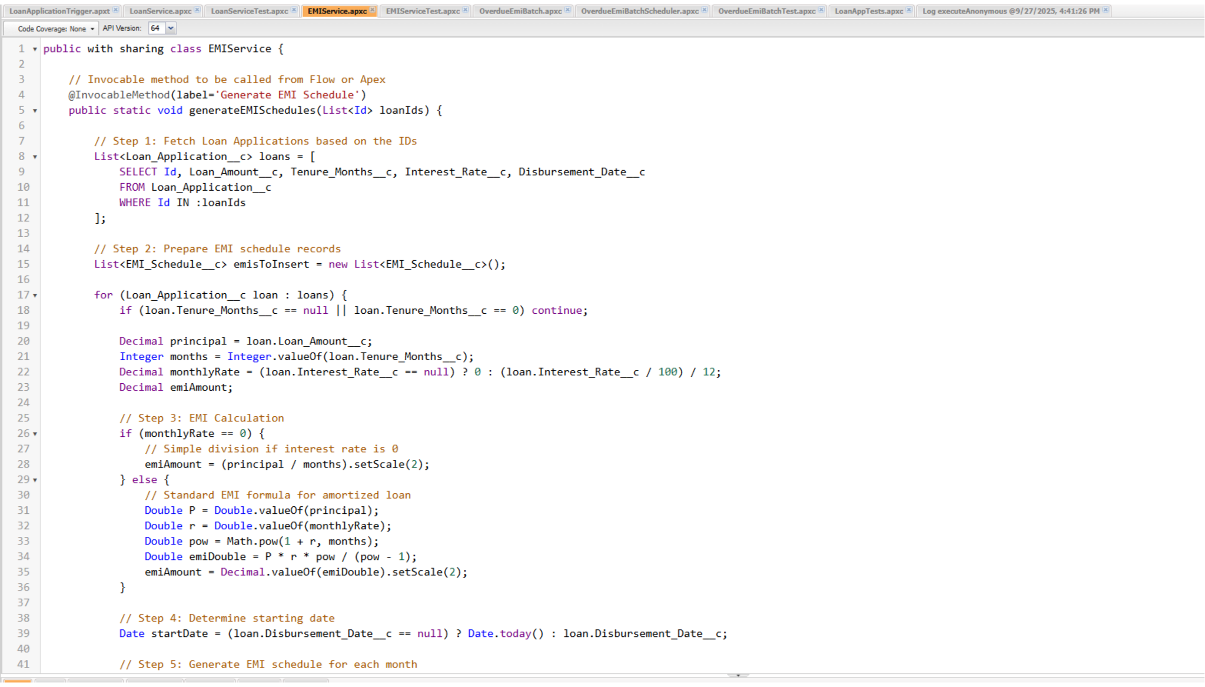

Execution Log

Timestamp	Event	Details
16:33:38:000	CODE_UNIT_ST...	[EXTERNAL]|execute_anonymous_apex
16:33:38:001	STATEMENT_EX...	[1]|
16:33:38:001	STATEMENT_EX...	[1]|
16:33:38:001	VARIABLE_ASSI...	[1]|sch|"0 0 2 * * ?"
16:33:38:001	STATEMENT_EX...	[2]|
16:33:38:003	METHOD_ENTRY	[1]|01pQy00000J9w0D|OverdueEmiBatchScheduler.OverdueEmiBatchScheduler()
16:33:38:003	STATEMENT_EX...	[1]|
16:33:38:003	STATEMENT_EX...	[1]|
16:33:38:003	METHOD_EXIT	[1]|OverdueEmiBatchScheduler

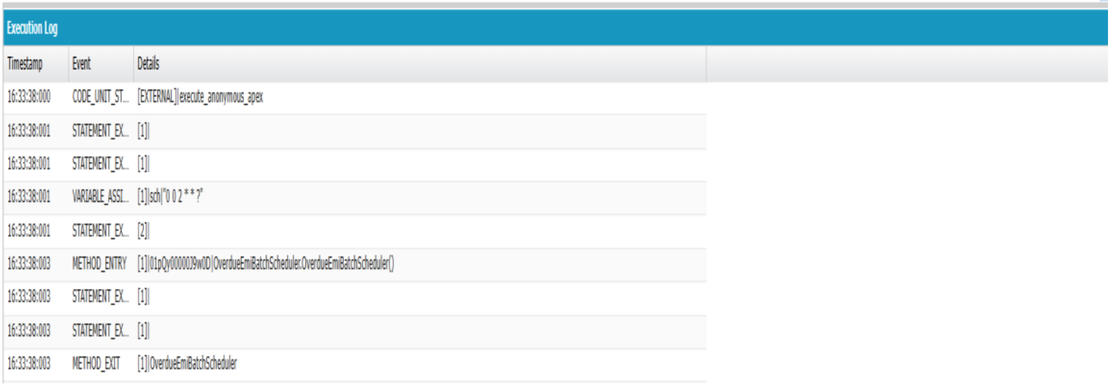
B) EMI Generation Service

- Generates amortized EMI schedules for approved loans.
- Can be invoked via Flow or directly in Apex.

EMIService.cls



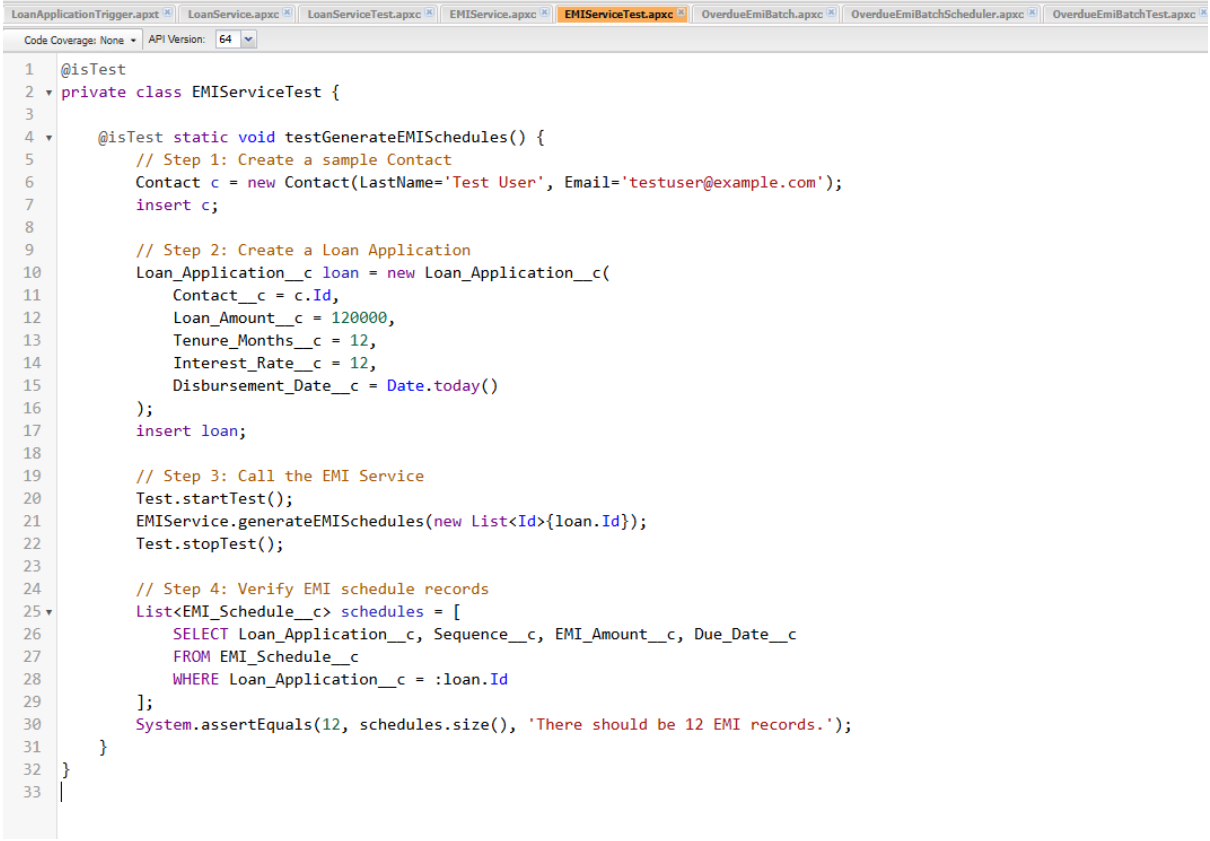
```
public with sharing class EMIService {

    // Invocable method to be called from Flow or Apex
    @InvocableMethod(label='Generate EMI Schedule')
    public static void generateEMISchedules(List<Id> loanIds) {

        // Step 1: Fetch Loan Applications based on the IDs
        List<Loan_Application__c> loans = [
            SELECT Id, Loan_Amount__c, Tenure_Months__c, Interest_Rate__c, Disbursement_Date__c
            FROM Loan_Application__c
            WHERE Id IN :loanIds
        ];

        // Step 2: Prepare EMI schedule records
        List<EMI_Schedule__c> emisToInsert = new List<EMI_Schedule__c>();

        for (Loan_Application__c loan : loans) {
            if (loan.Tenure_Months__c == null || loan.Tenure_Months__c == 0) continue;

            Decimal principal = loan.Loan_Amount__c;
            Integer months = Integer.valueOf(loan.Tenure_Months__c);
            Decimal monthlyRate = (loan.Interest_Rate__c == null) ? 0 : (loan.Interest_Rate__c / 100) / 12;
            Decimal emiAmount;

            // Step 3: EMI Calculation
            if (monthlyRate == 0) {
                // Simple division if interest rate is 0
                emiAmount = (principal / months).setScale(2);
            } else {
                // Standard EMI formula for amortized loan
                Double P = Double.valueOf(principal);
                Double r = Double.valueOf(monthlyRate);
                Double pow = Math.pow(1 + r, months);
                Double emiDouble = P * r * pow / (pow - 1);
                emiAmount = Decimal.valueOf(emiDouble).setScale(2);
            }

            // Step 4: Determine starting date
            Date startDate = (loan.Disbursement_Date__c == null) ? Date.today() : loan.Disbursement_Date__c;

            // Step 5: Generate EMI schedule for each month
```



Execution Log

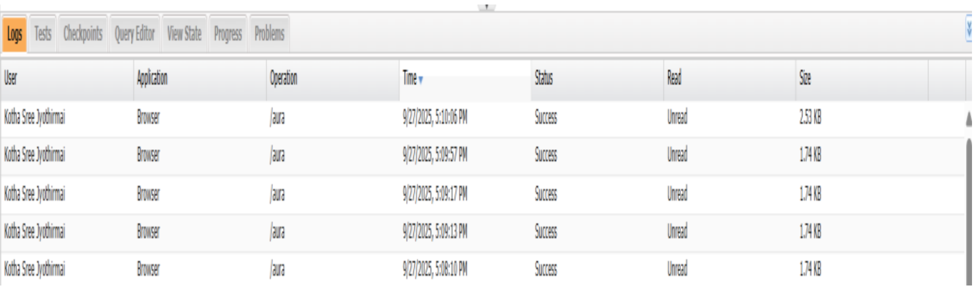
Timestamp	Event	Details
16:33:38:000	CODE_UNIT_ST...	[EXTERNAL]|execute_anonymous_apex
16:33:38:001	STATEMENT_EX...	[1]|
16:33:38:001	STATEMENT_EX...	[1]|
16:33:38:001	VARIABLE_ASSI...	[1]|sch|"0 0 2 * * ?"
16:33:38:001	STATEMENT_EX...	[2]|
16:33:38:003	METHOD_ENTRY	[1]|01pQy00000J9w0D|OverdueEmiBatchScheduler.OverdueEmiBatchScheduler()
16:33:38:003	STATEMENT_EX...	[1]|
16:33:38:003	STATEMENT_EX...	[1]|
16:33:38:003	METHOD_EXIT	[1]|OverdueEmiBatchScheduler

EMIServiceTest.cls

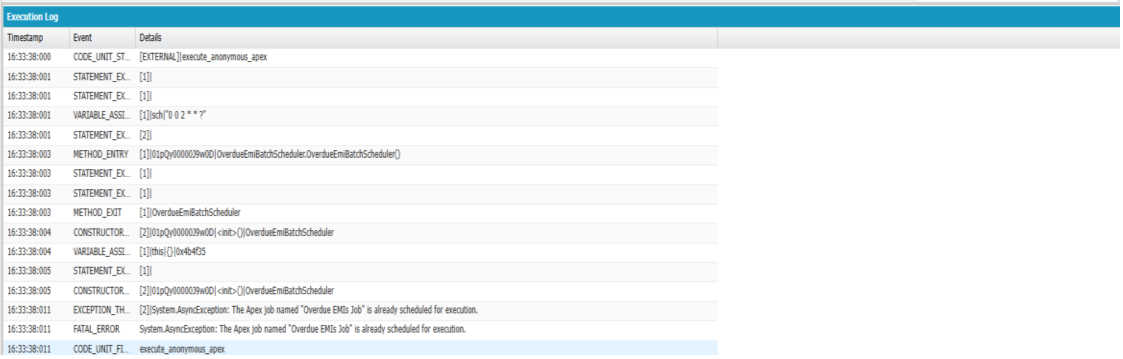


```
@isTest
private class EMIServiceTest {

    @isTest static void testGenerateEMISchedules() {
        // Step 1: Create a sample Contact
        Contact c = new Contact(LastName='Test User', Email='testuser@example.com');
        insert c;

        // Step 2: Create a Loan Application
        Loan_Application__c loan = new Loan_Application__c(
            Contact__c = c.Id,
            Loan_Amount__c = 120000,
            Tenure_Months__c = 12,
            Interest_Rate__c = 12,
            Disbursement_Date__c = Date.today()
        );
        insert loan;

        // Step 3: Call the EMI Service
        Test.startTest();
        EMIService.generateEMISchedules(new List<Id>{loan.Id});
        Test.stopTest();

        // Step 4: Verify EMI schedule records
        List<EMI_Schedule__c> schedules = [
            SELECT Loan_Application__c, Sequence__c, EMI_Amount__c, Due_Date__c
            FROM EMI_Schedule__c
            WHERE Loan_Application__c = :loan.Id
        ];
        System.assertEquals(12, schedules.size(), 'There should be 12 EMI records.');
    }
}

```

Output:



User	Application	Operation	Time	Status	Read	Size
Kotha Sree Jyothirmai	Browser	/aura	9/27/2025, 5:10:06 PM	Success	Unread	2.53 KB
Kotha Sree Jyothirmai	Browser	/aura	9/27/2025, 5:09:57 PM	Success	Unread	1.74 KB
Kotha Sree Jyothirmai	Browser	/aura	9/27/2025, 5:09:17 PM	Success	Unread	1.74 KB
Kotha Sree Jyothirmai	Browser	/aura	9/27/2025, 5:09:13 PM	Success	Unread	1.74 KB
Kotha Sree Jyothirmai	Browser	/aura	9/27/2025, 5:08:10 PM	Success	Unread	1.74 KB



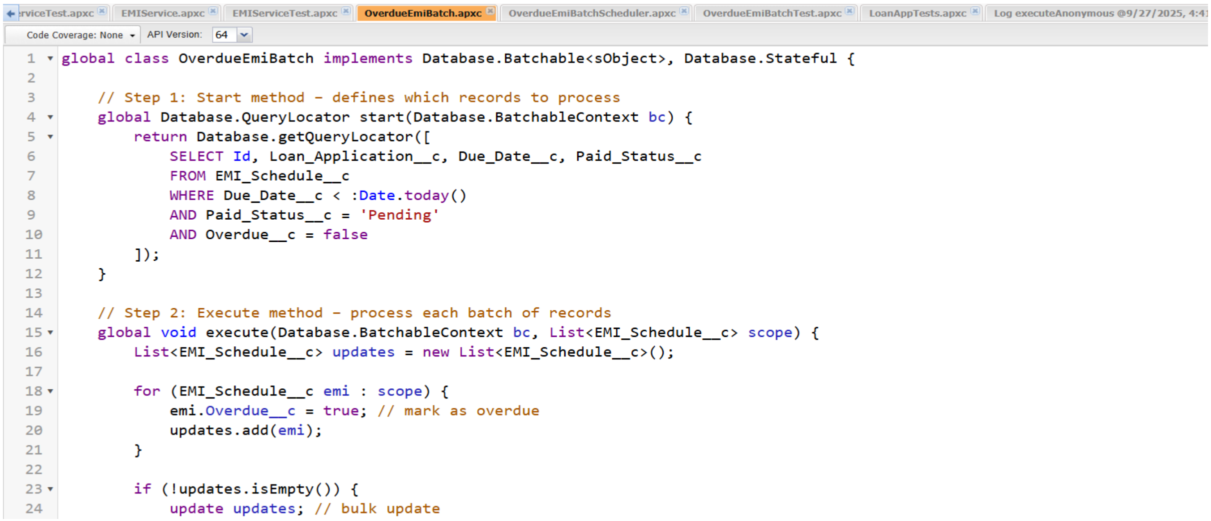
Execution Log

Timestamp	Event	Details
16:33:38:000	CODE_UNIT_ST...	[EXTERNAL]|execute_anonymous_apex
16:33:38:001	STATEMENT_EX...	[1]|
16:33:38:001	STATEMENT_EX...	[1]|
16:33:38:001	VARIABLE_ASSI...	[1]|sch|"0 0 2 * * ?"
16:33:38:001	STATEMENT_EX...	[2]|
16:33:38:003	METHOD_ENTRY	[1]|01pQy00000J9w0D|OverdueEmiBatchScheduler.OverdueEmiBatchScheduler()
16:33:38:003	STATEMENT_EX...	[1]|
16:33:38:003	STATEMENT_EX...	[1]|
16:33:38:003	METHOD_EXIT	[1]|OverdueEmiBatchScheduler
16:33:38:004	CONSTRUCTOR...	[2]|01pQy00000J9w0D|<init>()|OverdueEmiBatchScheduler
16:33:38:004	VARIABLE_ASSI...	[1]|this|{}|0x4b4f35
16:33:38:005	STATEMENT_EX...	[1]|
16:33:38:005	CONSTRUCTOR...	[2]|01pQy00000J9w0D|<init>()|OverdueEmiBatchScheduler
16:33:38:011	EXCEPTION_TH...	[2]|System.AsyncException: The Apex job named "Overdue EMIs Job" is already scheduled for execution.
16:33:38:011	FATAL_ERROR	System.AsyncException: The Apex job named "Overdue EMIs Job" is already scheduled for execution.
16:33:38:011	CODE_UNIT_FI...	execute_anonymous_apex

c) Overdue Batch – Scheduled Nightly

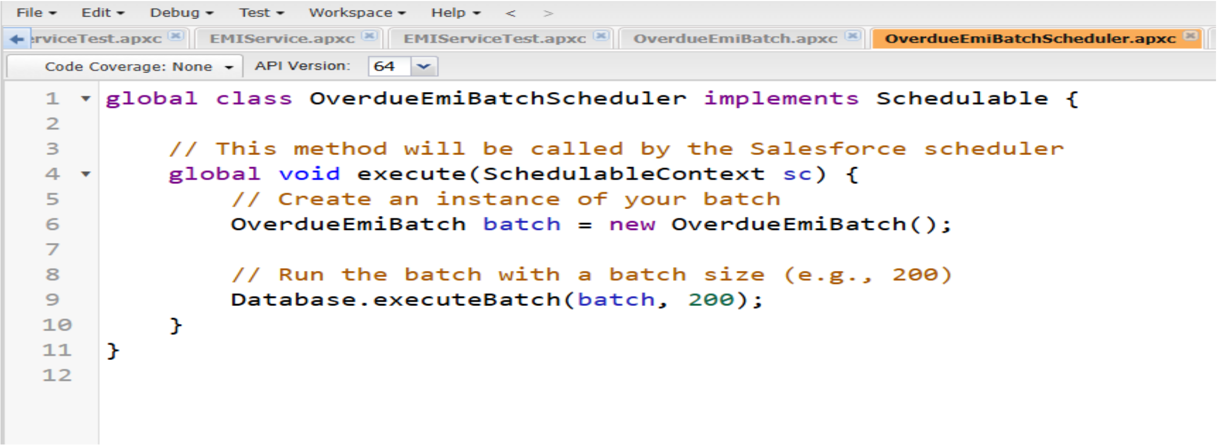
- Automatically marks EMIs overdue nightly.
- Bulk-safe and efficient.

OverdueEmiBatch.cls



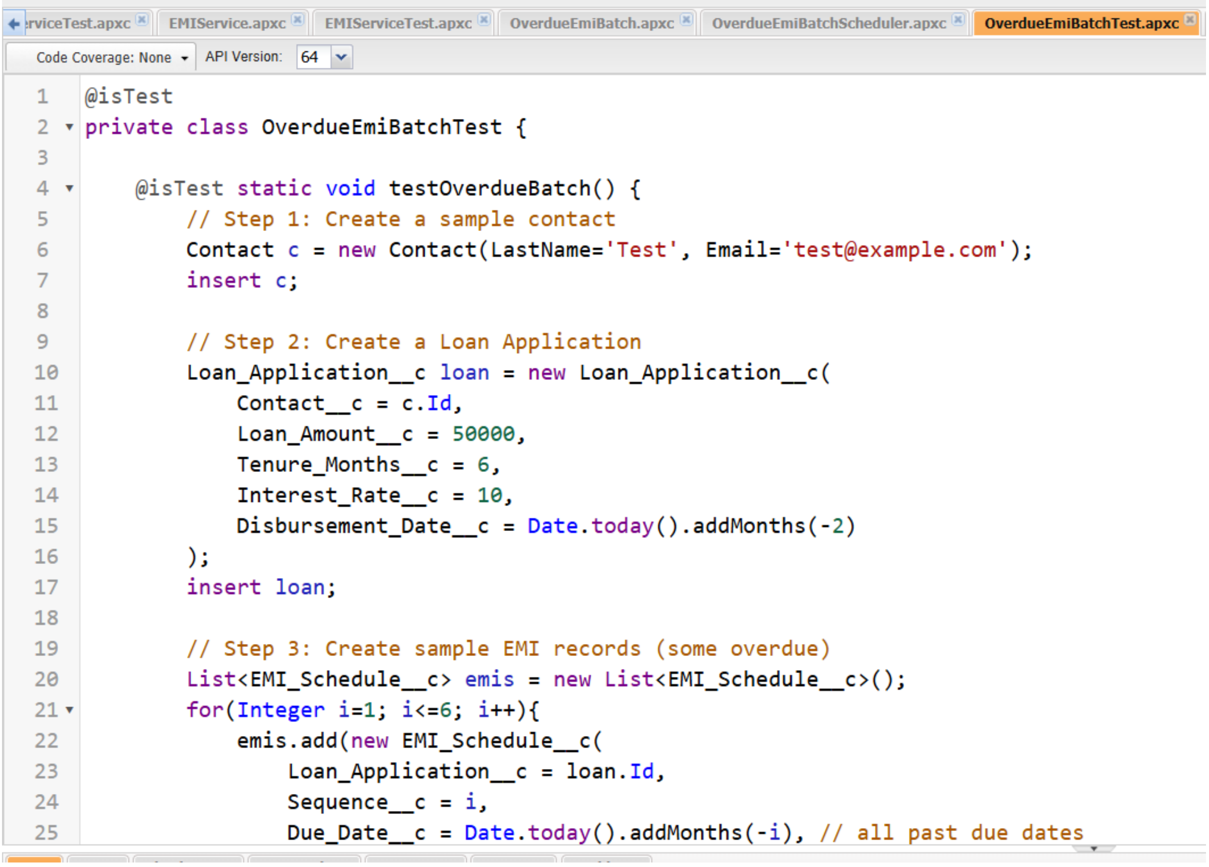
```
global class OverdueEmiBatch implements Database.Batchable<sObject>, Database.Stateful {

    // Step 1: Start method - defines which records to process
    global Database.QueryLocator start(Database.BatchableContext bc) {
        return Database.getQueryLocator([
            SELECT Id, Loan_Application__c, Due_Date__c, Paid_Status__c
            FROM EMI_Schedule__c
            WHERE Due_Date__c < :Date.today()
            AND Paid_Status__c = 'Pending'
            AND Overdue__c = false
        ]);
    }

    // Step 2: Execute method - process each batch of records
    global void execute(Database.BatchableContext bc, List<EMI_Schedule__c> scope) {
        List<EMI_Schedule__c> updates = new List<EMI_Schedule__c>();

        for (EMI_Schedule__c emi : scope) {
            emi.Overdue__c = true; // mark as overdue
            updates.add(emi);
        }

        if (!updates.isEmpty()) {
            update updates; // bulk update
```

Scheduler.cls



```
global class OverdueEmiBatchScheduler implements Schedulable {

    // This method will be called by the Salesforce scheduler
    global void execute(SchedulableContext sc) {
        // Create an instance of your batch
        OverdueEmiBatch batch = new OverdueEmiBatch();

        // Run the batch with a batch size (e.g., 200)
        Database.executeBatch(batch, 200);
    }
}

```

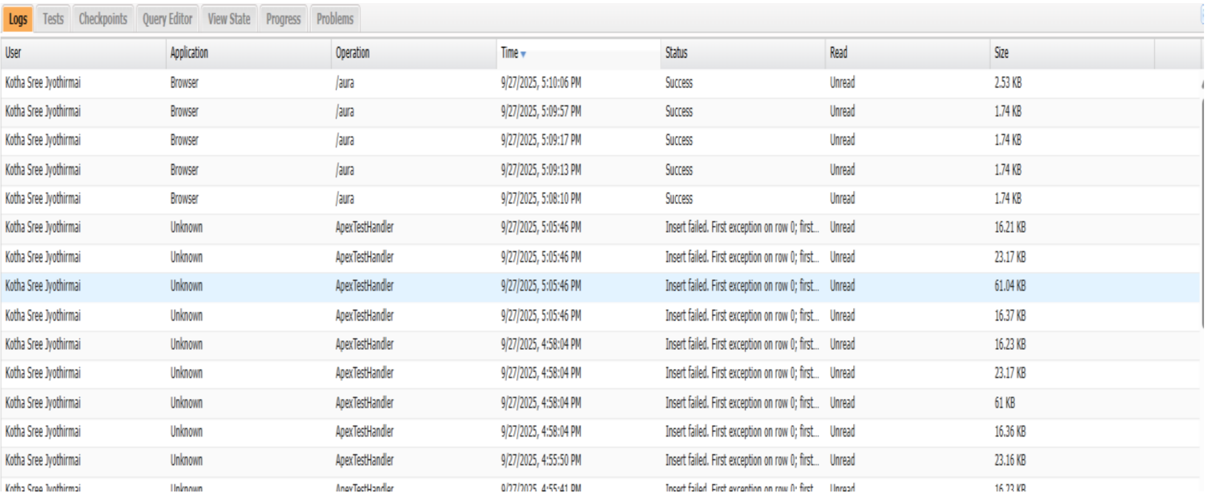
Test class



Code Coverage: None API Version: 64

```
@isTest
private class OverdueEmiBatchTest {

    @isTest static void testOverdueBatch() {
        // Step 1: Create a sample contact
        Contact c = new Contact(LastName='Test', Email='test@example.com');
        insert c;

        // Step 2: Create a Loan Application
        Loan_Application__c loan = new Loan_Application__c(
            Contact__c = c.Id,
            Loan_Amount__c = 50000,
            Tenure_Months__c = 6,
            Interest_Rate__c = 10,
            Disbursement_Date__c = Date.today().addMonths(-2)
        );
        insert loan;

        // Step 3: Create sample EMI records (some overdue)
        List<EMI_Schedule__c> emis = new List<EMI_Schedule__c>();
        for(Integer i=1; i<=6; i++){
            emis.add(new EMI_Schedule__c(
                Loan_Application__c = loan.Id,
                Sequence__c = i,
                Due_Date__c = Date.today().addMonths(-i), // all past due dates
```

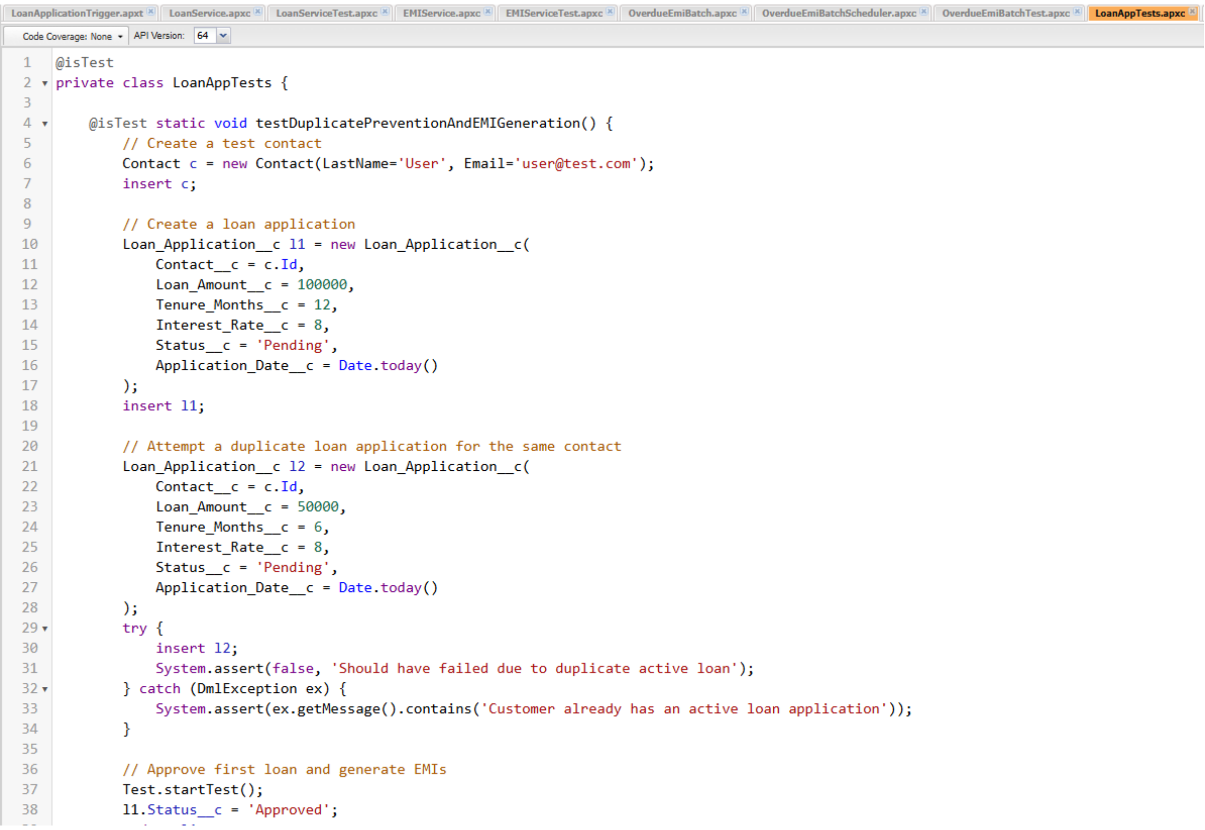
Output



User	Application	Operation	Time	Status	Read	Size
Kotha Sree Jyothirmai	Browser	/aura	9/27/2025, 5:10:06 PM	Success	Unread	2.53 KB
Kotha Sree Jyothirmai	Browser	/aura	9/27/2025, 5:09:57 PM	Success	Unread	1.74 KB
Kotha Sree Jyothirmai	Browser	/aura	9/27/2025, 5:09:17 PM	Success	Unread	1.74 KB
Kotha Sree Jyothirmai	Browser	/aura	9/27/2025, 5:09:13 PM	Success	Unread	1.74 KB
Kotha Sree Jyothirmai	Browser	/aura	9/27/2025, 5:08:10 PM	Success	Unread	1.74 KB
Kotha Sree Jyothirmai	Unknown	ApexTestHandler	9/27/2025, 5:05:46 PM	Insert failed. First exception on row 0; first...	Unread	16.21 KB
Kotha Sree Jyothirmai	Unknown	ApexTestHandler	9/27/2025, 5:05:46 PM	Insert failed. First exception on row 0; first...	Unread	23.17 KB
Kotha Sree Jyothirmai	Unknown	ApexTestHandler	9/27/2025, 5:05:46 PM	Insert failed. First exception on row 0; first...	Unread	61.04 KB
Kotha Sree Jyothirmai	Unknown	ApexTestHandler	9/27/2025, 5:05:46 PM	Insert failed. First exception on row 0; first...	Unread	16.37 KB
Kotha Sree Jyothirmai	Unknown	ApexTestHandler	9/27/2025, 4:58:04 PM	Insert failed. First exception on row 0; first...	Unread	16.23 KB
Kotha Sree Jyothirmai	Unknown	ApexTestHandler	9/27/2025, 4:58:04 PM	Insert failed. First exception on row 0; first...	Unread	23.17 KB
Kotha Sree Jyothirmai	Unknown	ApexTestHandler	9/27/2025, 4:58:04 PM	Insert failed. First exception on row 0; first...	Unread	61 KB
Kotha Sree Jyothirmai	Unknown	ApexTestHandler	9/27/2025, 4:58:04 PM	Insert failed. First exception on row 0; first...	Unread	16.36 KB
Kotha Sree Jyothirmai	Unknown	ApexTestHandler	9/27/2025, 4:55:50 PM	Insert failed. First exception on row 0; first...	Unread	23.16 KB
Kotha Sree Jyothirmai	Unknown	ApexTestHandler	9/27/2025, 4:55:41 PM	Insert failed. First exception on row 0; first...	Unread	16.73 KB

Test Classes

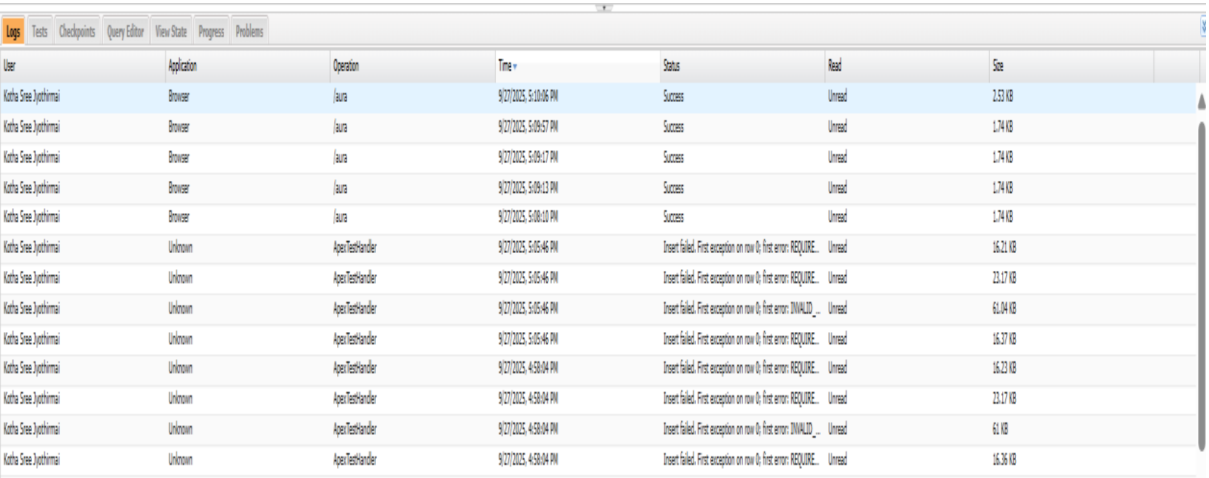
- **Covers duplicate loan prevention.**
- **Verifies correct EMI generation.**
- **Ensures bulk-safe operations.**



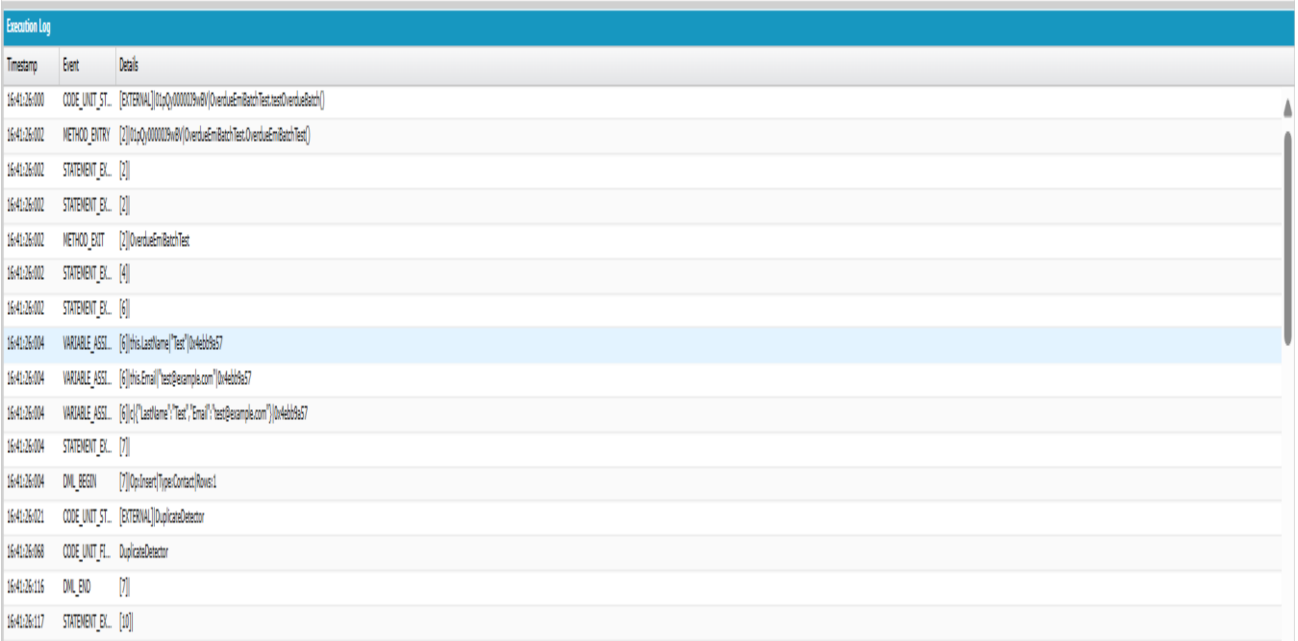
Code Coverage: None API Version: 64

```
@isTest
private class LoanAppTests {

    @isTest static void testDuplicatePreventionAndEMIGeneration() {
        // Create a test contact
        Contact c = new Contact(LastName='User', Email='user@test.com');
        insert c;

        // Create a loan application
        Loan_Application__c l1 = new Loan_Application__c(
            Contact__c = c.Id,
            Loan_Amount__c = 100000,
            Tenure_Months__c = 12,
            Interest_Rate__c = 8,
            Status__c = 'Pending',
            Application_Date__c = Date.today()
        );
        insert l1;

        // Attempt a duplicate loan application for the same contact
        Loan_Application__c l2 = new Loan_Application__c(
            Contact__c = c.Id,
            Loan_Amount__c = 50000,
            Tenure_Months__c = 6,
            Interest_Rate__c = 8,
            Status__c = 'Pending',
            Application_Date__c = Date.today()
        );
        try {
            insert l2;
            System.assert(false, 'Should have failed due to duplicate active loan');
        } catch (DmlException ex) {
            System.assert(ex.getMessage().contains('Customer already has an active loan application'));
        }

        // Approve first loan and generate EMIs
        Test.startTest();
        l1.Status__c = 'Approved';
```

Output:

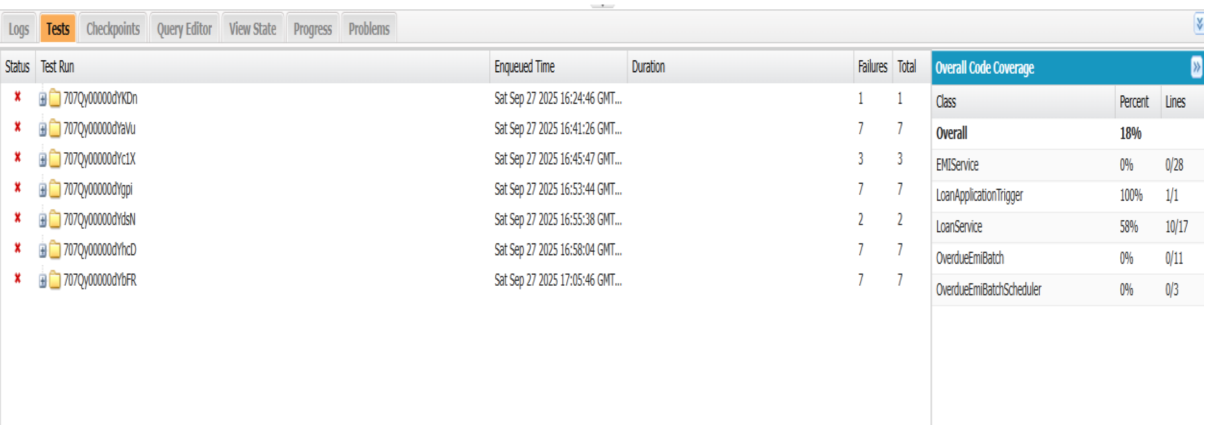


User	Application	Operation	Time	Status	Read	Size
Kotha Sree Jyothirmai	Browser	/aura	9/27/2025, 5:10:06 PM	Success	Unread	2.53 KB
Kotha Sree Jyothirmai	Browser	/aura	9/27/2025, 5:09:57 PM	Success	Unread	1.74 KB
Kotha Sree Jyothirmai	Browser	/aura	9/27/2025, 5:09:17 PM	Success	Unread	1.74 KB
Kotha Sree Jyothirmai	Browser	/aura	9/27/2025, 5:09:13 PM	Success	Unread	1.74 KB
Kotha Sree Jyothirmai	Browser	/aura	9/27/2025, 5:08:10 PM	Success	Unread	1.74 KB
Kotha Sree Jyothirmai	Unknown	ApexTestHandler	9/27/2025, 5:05:46 PM	Insert failed. First exception on row 0; first error: REQUIRE...	Unread	16.21 KB
Kotha Sree Jyothirmai	Unknown	ApexTestHandler	9/27/2025, 5:05:46 PM	Insert failed. First exception on row 0; first error: REQUIRE...	Unread	23.17 KB
Kotha Sree Jyothirmai	Unknown	ApexTestHandler	9/27/2025, 5:05:46 PM	Insert failed. First exception on row 0; first error: INVALID_...	Unread	61.04 KB
Kotha Sree Jyothirmai	Unknown	ApexTestHandler	9/27/2025, 5:05:46 PM	Insert failed. First exception on row 0; first error: REQUIRE...	Unread	16.37 KB
Kotha Sree Jyothirmai	Unknown	ApexTestHandler	9/27/2025, 4:58:04 PM	Insert failed. First exception on row 0; first error: REQUIRE...	Unread	16.23 KB
Kotha Sree Jyothirmai	Unknown	ApexTestHandler	9/27/2025, 4:58:04 PM	Insert failed. First exception on row 0; first error: REQUIRE...	Unread	23.17 KB
Kotha Sree Jyothirmai	Unknown	ApexTestHandler	9/27/2025, 4:58:04 PM	Insert failed. First exception on row 0; first error: INVALID_...	Unread	61 KB
Kotha Sree Jyothirmai	Unknown	ApexTestHandler	9/27/2025, 4:58:04 PM	Insert failed. First exception on row 0; first error: REQUIRE...	Unread	16.36 KB



Execution Log

Timestamp	Event	Details
16:41:26:000	CODE_UNIT_ST...	[EXTERNAL]|01pQy0000029wBV|OverdueEmiBatchTest.testOverdueBatch()
16:41:26:002	METHOD_ENTRY	[2]|01pQy0000029wBV|OverdueEmiBatchTest.OverdueEmiBatchTest()
16:41:26:002	STATEMENT_EX...	[2]|
16:41:26:002	STATEMENT_EX...	[2]|
16:41:26:002	METHOD_EXIT	[2]|OverdueEmiBatchTest
16:41:26:002	STATEMENT_EX...	[4]|
16:41:26:002	STATEMENT_EX...	[6]|
16:41:26:004	VARIABLE_ASSI...	[6]|this.LastName|"Test"|0x4ebb9a57
16:41:26:004	VARIABLE_ASSI...	[6]|this.Email|"test@example.com"|0x4ebb9a57
16:41:26:004	VARIABLE_ASSI...	[6]|c|{"LastName":"Test","Email":"test@example.com"}|0x4ebb9a57
16:41:26:004	STATEMENT_EX...	[7]|
16:41:26:004	DML_BEGIN	[7]|Op:Insert|Type:Contact|Rows:1
16:41:26:021	CODE_UNIT_ST...	[EXTERNAL]|DuplicateDetector
16:41:26:068	CODE_UNIT_FI...	DuplicateDetector
16:41:26:116	DML_END	[7]|
16:41:26:117	STATEMENT_EX...	[10]|



Status	Test Run	Enqueued Time	Duration	Failures	Total
✘	707Qy00000dYKDn	Sat Sep 27 2025 16:24:46 GMT...		1	1
✘	707Qy00000dYaVu	Sat Sep 27 2025 16:41:26 GMT...		7	7
✘	707Qy00000dYc1X	Sat Sep 27 2025 16:45:47 GMT...		3	3
✘	707Qy00000dYgpi	Sat Sep 27 2025 16:53:44 GMT...		7	7
✘	707Qy00000dYdsN	Sat Sep 27 2025 16:55:38 GMT...		2	2
✘	707Qy00000dYhcD	Sat Sep 27 2025 16:58:04 GMT...		7	7
✘	707Qy00000dYbFR	Sat Sep 27 2025 17:05:46 GMT...		7	7

Overall Code Coverage

Class	Percent	Lines
Overall	**18%**	
EMIService	0%	0/28
LoanApplicationTrigger	100%	1/1
LoanService	58%	10/17
OverdueEmiBatch	0%	0/11
OverdueEmiBatchScheduler	0%	0/3

Collections:

In Apex, collections are lists, sets, and maps. In this test class, you're using:

List<Id>

EMIService.generateEMISchedules(new List<Id>{l1.Id});

- A List collection is created with a single Loan Id (l1.Id).
- Passed to EMIService so it can bulk-generate EMI schedules for one or more loans.
- Shows bulkification best practice.

Control Statements Used

Control statements control flow of logic in your code. In this test class, you’re using:

1. try-catch block

```
try {
   insert l2;
   System.assert(false, 'Should have failed due to duplicate active loan');
} catch (DmlException ex) {
   System.assert(ex.getMessage().contains('Customer already has an active loan application'));
}
```

- Purpose: Handle the expected DML exception.
- try → attempts to insert a duplicate loan.
- catch → catches the DmlException when trigger blocks it.
- Ensures negative testing is validated.