Phase 3: Apex Classes Development

Detailed Step-by-Step Instructions

- **STEP 1 Create the Apex Controller Class**

1.1 Open VS Code and Your Project

-Open VS Code

-Make sure your LeaveTrackerApp project is open

-Verify you're connected to your Salesforce org (check bottom blue bar)

1.2 Create New Apex Class

-Right-click on force-app/main/default/classes folder

-Select: "SFDX Create Apex Class"

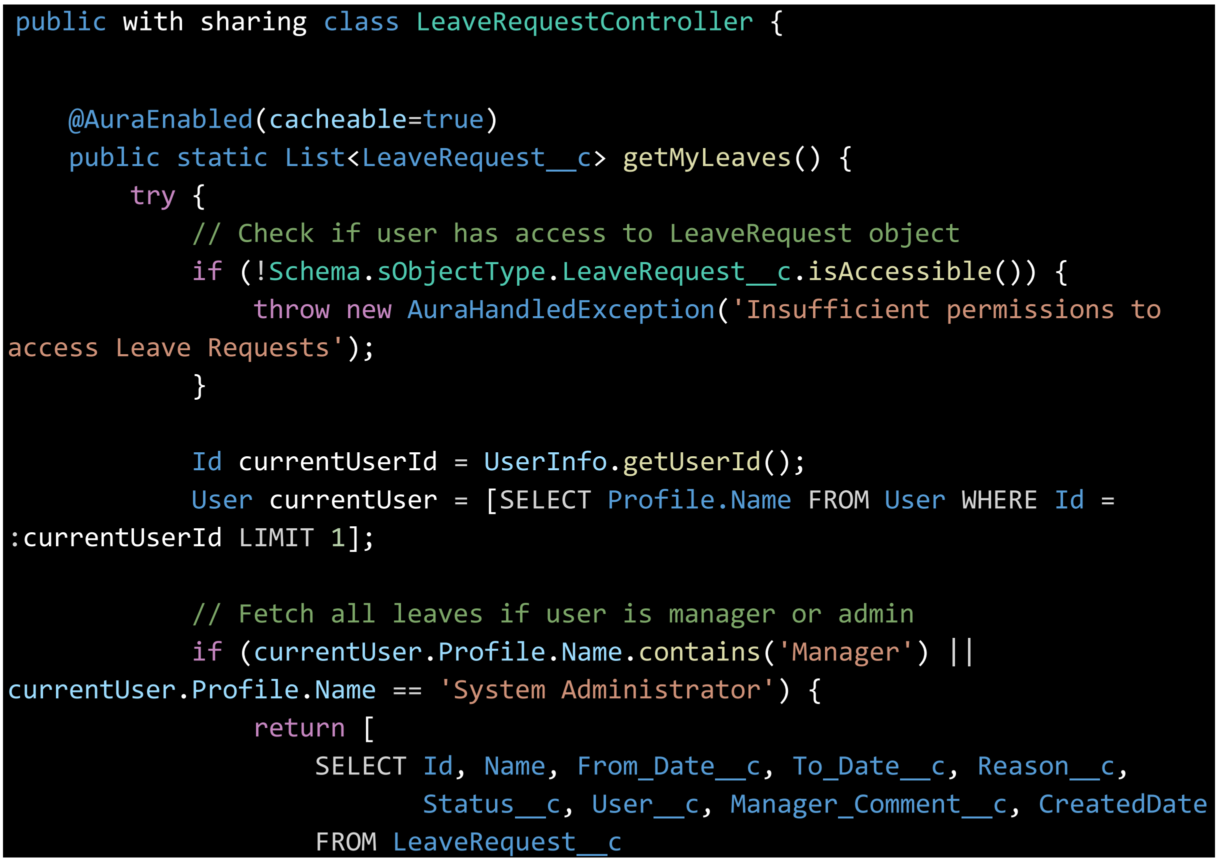
-Enter name:LeaveRequestController

-Press Enter to accept default location

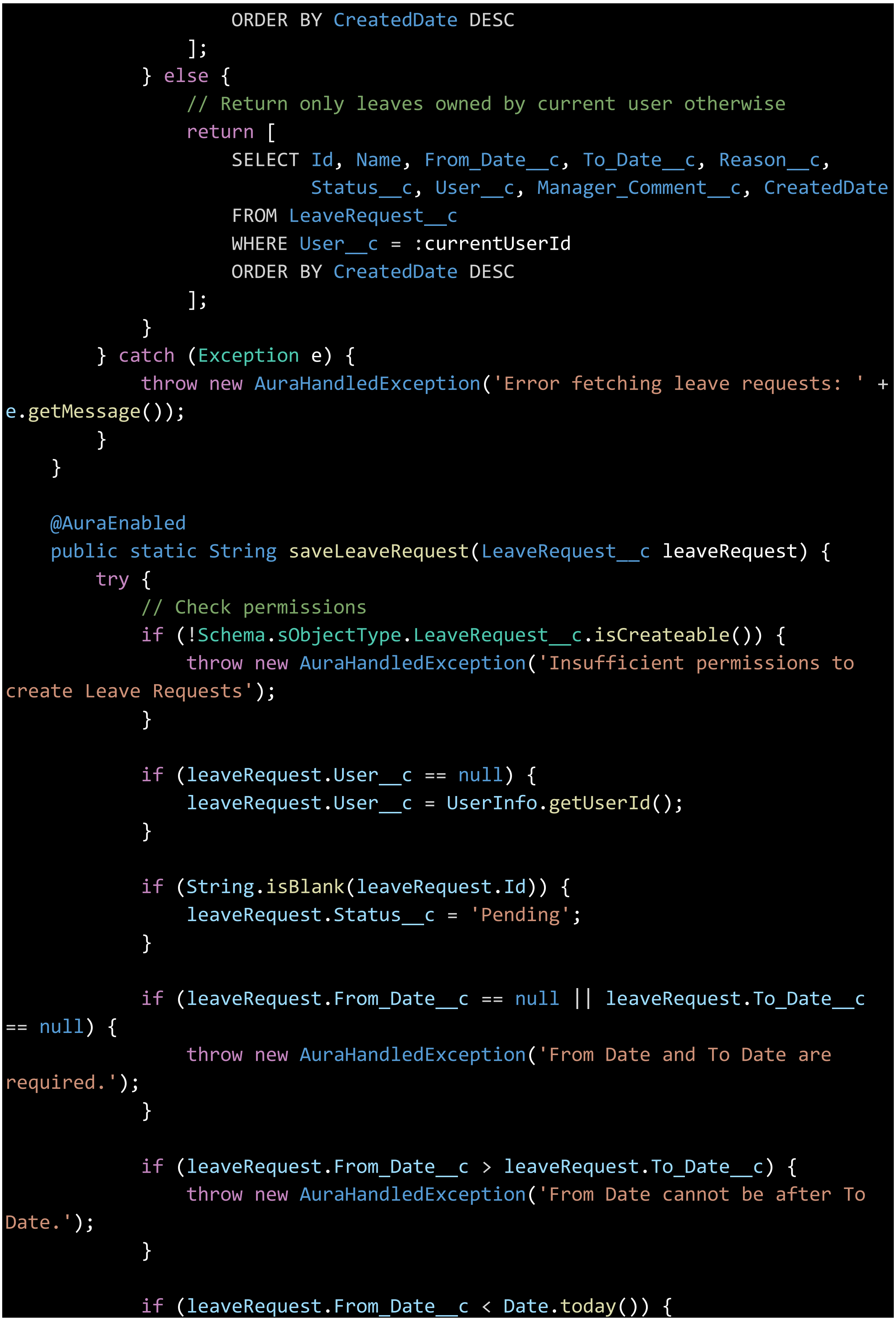
1.3 Replace Default Code

-VS Code will create a file with basic template. Replace ALL the content with this code

Code:



```
public with sharing class LeaveRequestController {


    @AuraEnabled(cacheable=true)
    public static List<LeaveRequest__c> getMyLeaves() {
        try {
            // Check if user has access to LeaveRequest object
            if (!Schema.sObjectType.LeaveRequest__c.isAccessible()) {
                throw new AuraHandledException('Insufficient permissions to
access Leave Requests');
            }

            Id currentUserId = UserInfo.getUserId();
            User currentUser = [SELECT Profile.Name FROM User WHERE Id =
:currentUserId LIMIT 1];

            // Fetch all leaves if user is manager or admin
            if (currentUser.Profile.Name.contains('Manager') ||
currentUser.Profile.Name == 'System Administrator') {
                return [
                    SELECT Id, Name, From_Date__c, To_Date__c, Reason__c,
                           Status__c, User__c, Manager_Comment__c, CreatedDate
                    FROM LeaveRequest__c
```



```
                    ORDER BY CreatedDate DESC
                ];
            } else {
                // Return only leaves owned by current user otherwise
                return [
                    SELECT Id, Name, From_Date__c, To_Date__c, Reason__c,
                           Status__c, User__c, Manager_Comment__c, CreatedDate
                    FROM LeaveRequest__c
                    WHERE User__c = :currentUserId
                    ORDER BY CreatedDate DESC
                ];
            }
        } catch (Exception e) {
            throw new AuraHandledException('Error fetching leave requests: ' +
e.getMessage());
        }
    }

    @AuraEnabled
    public static String saveLeaveRequest(LeaveRequest__c leaveRequest) {
        try {
            // Check permissions
            if (!Schema.sObjectType.LeaveRequest__c.isCreateable()) {
                throw new AuraHandledException('Insufficient permissions to
create Leave Requests');
            }

            if (leaveRequest.User__c == null) {
                leaveRequest.User__c = UserInfo.getUserId();
            }

            if (String.isBlank(leaveRequest.Id)) {
                leaveRequest.Status__c = 'Pending';
            }

            if (leaveRequest.From_Date__c == null || leaveRequest.To_Date__c
== null) {
                throw new AuraHandledException('From Date and To Date are
required.');
            }

            if (leaveRequest.From_Date__c > leaveRequest.To_Date__c) {
                throw new AuraHandledException('From Date cannot be after To
Date.');
            }

            if (leaveRequest.From_Date__c < Date.today()) {
                throw new AuraHandledException('From Date cannot be in the
past.');
            }

            upsert leaveRequest;
            return leaveRequest.Id;

        } catch (Exception e) {
            throw new AuraHandledException('Error saving leave request: ' +
e.getMessage());
        }
    }

    @AuraEnabled
    public static void updateLeaveStatus(String recordId, String status,
String comment) {
        try {
            // Check permissions
            if (!Schema.sObjectType.LeaveRequest__c.isUpdateable()) {
                throw new AuraHandledException('Insufficient permissions to
update Leave Requests');
            }

            LeaveRequest__c leaveRequest = [
                SELECT Id, Status__c, Manager_Comment__c
                FROM LeaveRequest__c
                WHERE Id = :recordId
                LIMIT 1
            ];

            leaveRequest.Status__c = status;
            leaveRequest.Manager_Comment__c = comment;

            update leaveRequest;

        } catch (Exception e) {
            throw new AuraHandledException('Error updating leave status: ' +
e.getMessage());
        }
    }

    @AuraEnabled(cacheable=true)
    public static List<LeaveRequest__c> getLeaveRequestsForApproval() {
        try {
            if (!Schema.sObjectType.LeaveRequest__c.isAccessible()) {
                throw new AuraHandledException('Insufficient permissions to
access Leave Requests');
            }
            return [
                SELECT Id, Name, From_Date__c, To_Date__c, Reason__c,
                       Status__c, User__c, User__r.Name, Manager_Comment__c, CreatedDate
                FROM LeaveRequest__c
                WHERE Status__c = 'Pending'
                ORDER BY CreatedDate DESC
            ];
        } catch (Exception e) {
            throw new AuraHandledException('Error fetching leave requests for approval: ' + e.getMessage());
        }
    }
}
```

1.4 Save and Deploy

Save the file Ctrl+S

Right-click on the LeaveRequestController.cls file

Select: "SFDX Deploy Source to Org"

Wait for "Deploy Succeeded" message

- **STEP 2 Create the Test Class**

2.1 Create Test Class File

Right-click on force-app/main/default/classes folder

Select: "SFDX Create Apex Class"

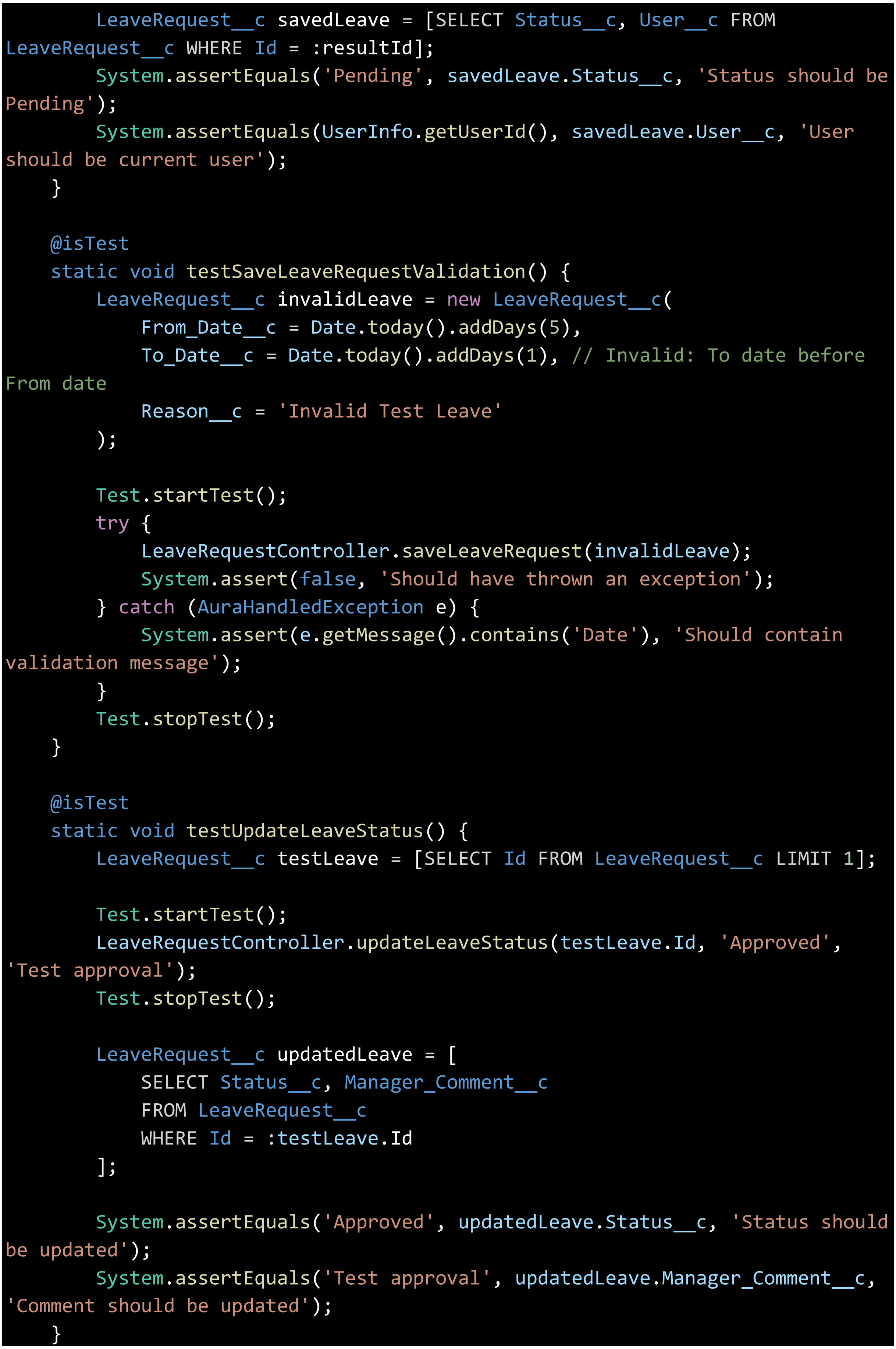
Enter name:LeaveRequestControllerTest

Press Enter

2.2 Replace with Test Code

Code:

```
@isTest
public class LeaveRequestControllerTest {

    @testSetup
    static void setupTestData() {
        // Create test user (using current user)
        User testUser = [SELECT Id FROM User WHERE Id = :UserInfo.getUserId()];

        // Create test leave requests
        List<LeaveRequest__c> testLeaves = new List<LeaveRequest__c>();

        testLeaves.add(new LeaveRequest__c(
            From_Date__c = Date.today().addDays(1),
            To_Date__c = Date.today().addDays(5),
            Reason__c = 'Test Vacation',
            Status__c = 'Pending',
            User__c = testUser.Id
        ));

        testLeaves.add(new LeaveRequest__c(
            From_Date__c = Date.today().addDays(10),
            To_Date__c = Date.today().addDays(15),
            Reason__c = 'Test Sick Leave',
            Status__c = 'Approved',
            User__c = testUser.Id,
            Manager_Comment__c = 'Approved for testing'
        ));

        insert testLeaves;
    }

    @isTest
    static void testGetMyLeaves() {
        Test.startTest();
        List<LeaveRequest__c> results = LeaveRequestController.getMyLeaves();
        Test.stopTest();

        System.assertNotEquals(null, results, 'Results should not be null');
        System.assertEquals(2, results.size(), 'Should return 2 leave
requests');
    }

    @isTest
    static void testSaveLeaveRequest() {
        LeaveRequest__c newLeave = new LeaveRequest__c(
            From_Date__c = Date.today().addDays(20),
            To_Date__c = Date.today().addDays(25),
            Reason__c = 'Test New Leave'
        );

        Test.startTest();
        String resultId = LeaveRequestController.saveLeaveRequest(newLeave);
        Test.stopTest();

        System.assertNotEquals(null, resultId, 'Should return an Id');
```



```
        LeaveRequest__c savedLeave = [SELECT Status__c, User__c FROM LeaveRequest__c WHERE Id = :resultId];
        System.assertEquals('Pending', savedLeave.Status__c, 'Status should be Pending');
        System.assertEquals(UserInfo.getUserId(), savedLeave.User__c, 'User should be current user');
    }

    @isTest
    static void testSaveLeaveRequestValidation() {
        LeaveRequest__c invalidLeave = new LeaveRequest__c(
            From_Date__c = Date.today().addDays(5),
            To_Date__c = Date.today().addDays(1), // Invalid: To date before From date
            Reason__c = 'Invalid Test Leave'
        );

        Test.startTest();
        try {
            LeaveRequestController.saveLeaveRequest(invalidLeave);
            System.assert(false, 'Should have thrown an exception');
        } catch (AuraHandledException e) {
            System.assert(e.getMessage().contains('Date'), 'Should contain validation message');
        }
        Test.stopTest();
    }

    @isTest
    static void testUpdateLeaveStatus() {
        LeaveRequest__c testLeave = [SELECT Id FROM LeaveRequest__c LIMIT 1];

        Test.startTest();
        LeaveRequestController.updateLeaveStatus(testLeave.Id, 'Approved', 'Test approval');
        Test.stopTest();

        LeaveRequest__c updatedLeave = [
            SELECT Status__c, Manager_Comment__c
            FROM LeaveRequest__c
            WHERE Id = :testLeave.Id
        ];

        System.assertEquals('Approved', updatedLeave.Status__c, 'Status should be updated');
        System.assertEquals('Test approval', updatedLeave.Manager_Comment__c, 'Comment should be updated');
    }
    @isTest
    static void testGetLeaveRequestsForApproval() {
        Test.startTest();
        List<LeaveRequest__c> results =
LeaveRequestController.getLeaveRequestsForApproval();
        Test.stopTest();

        System.assertNotEquals(null, results, 'Results should not be null');
        // Should have at least 1 pending request from test data
        System.assert(results.size() >= 1, 'Should have pending requests');
    }
}
```

2.3 Deploy Test Class

Save the file Ctrl+S

Right-click on LeaveRequestControllerTest.cls

Select: "SFDX Deploy Source to Org"

Wait for success message

- **STEP 3 Run Tests**

3.1 Run Tests from VS Code

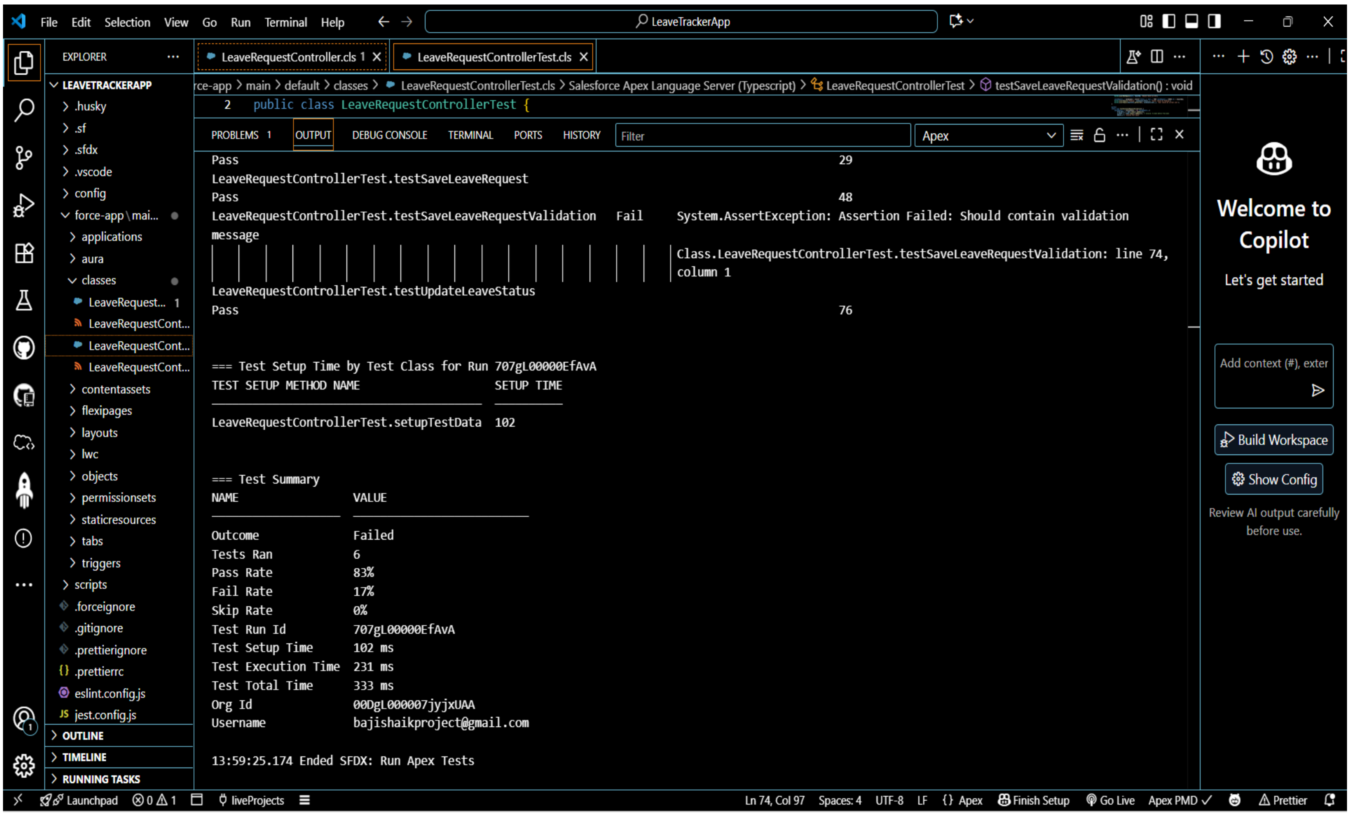
Right-click on

LeaveRequestControllerTest.cls

Select: "SFDX Run Apex Tests"

Wait for test results

Verify all tests pass and coverage is above 75%



```
  2   public class LeaveRequestControllerTest {
```

```
Pass                                                                                         29
LeaveRequestControllerTest.testSaveLeaveRequest
Pass                                                                                         48
LeaveRequestControllerTest.testSaveLeaveRequestValidation   Fail     System.AssertException: Assertion Failed: Should contain validation
message
│ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │Class.LeaveRequestControllerTest.testSaveLeaveRequestValidation: line 74,
                                    column 1
LeaveRequestControllerTest.testUpdateLeaveStatus
Pass                                                                                         76


=== Test Setup Time by Test Class for Run 707gL00000EfAvA
TEST SETUP METHOD NAME                     SETUP TIME
────────────────────────────────────────── ──────────
LeaveRequestControllerTest.setupTestData  102


=== Test Summary
NAME                VALUE
─────────────────── ───────────────────────────
Outcome             Failed
Tests Ran           6
Pass Rate           83%
Fail Rate           17%
Skip Rate           0%
Test Run Id         707gL00000EfAvA
Test Setup Time     102 ms
Test Execution Time 231 ms
Test Total Time     333 ms
Org Id              00DgL000007jyjxUAA
Username            bajishaikproject@gmail.com

13:59:25.174 Ended SFDX: Run Apex Tests
```

3.2 Alternative - Run Tests from Salesforce

Go to Setup in Salesforce

Search: "Apex Test Execution"

Click: "Apex Test Execution"

Select: LeaveRequestControllerTest

Run the tests

❖ STEP 4 Verification

4.1 Verify Classes Are Deployed

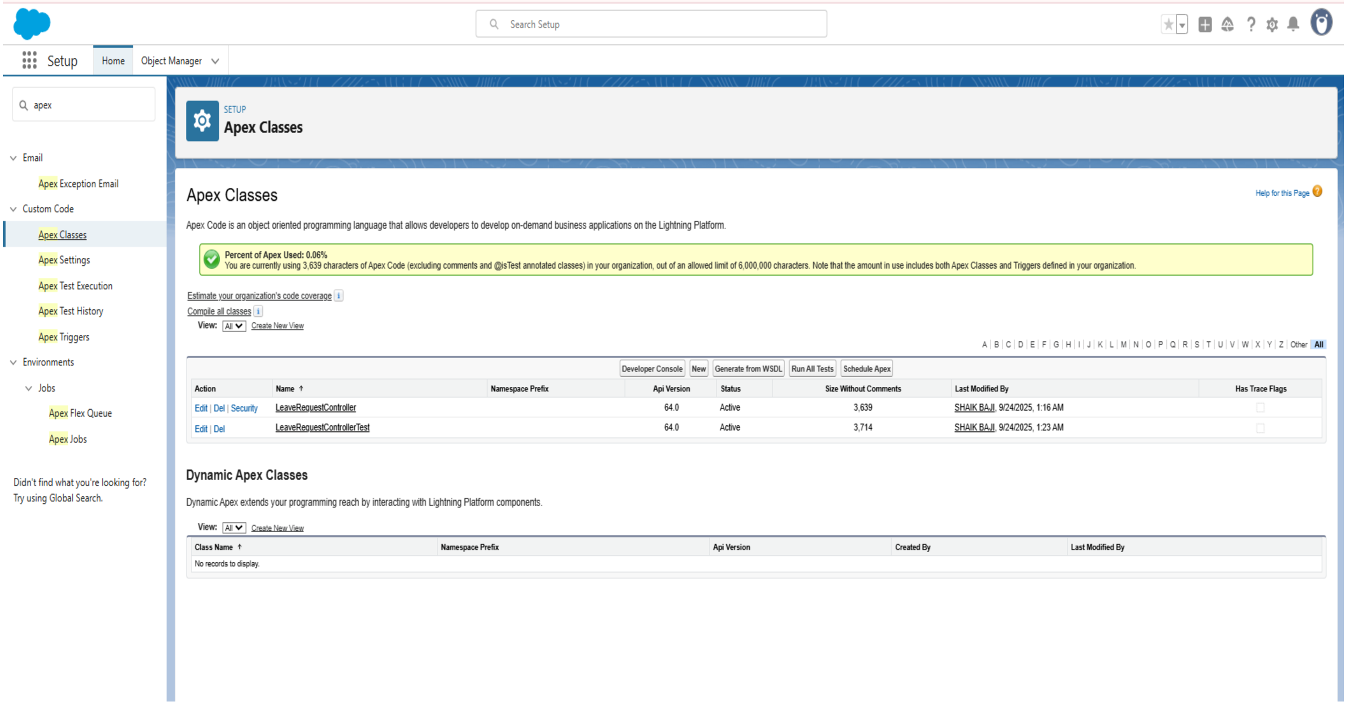
In Salesforce Setup, search for "Apex Classes"

Verify you see:

LeaveRequestController

LeaveRequestControllerTest

Check that both are deployed successfully



Apex Classes

Apex Classes

Apex Code is an object oriented programming language that allows developers to develop on-demand business applications on the Lightning Platform.

Percent of Apex Used: 0.06%
You are currently using 3,639 characters of Apex Code (excluding comments and @isTest annotated classes) in your organization, out of an allowed limit of 6,000,000 characters. Note that the amount in use includes both Apex Classes and Triggers defined in your organization.

Estimate your organization's code coverage

Compile all classes

View: All Create New View

Action	Name ↑	Namespace Prefix	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
Edit | Del | Security	LeaveRequestController		64.0	Active	3,639	SHAIK BAJI, 9/24/2025, 1:16 AM	□
Edit | Del	LeaveRequestControllerTest		64.0	Active	3,714	SHAIK BAJI, 9/24/2025, 1:23 AM	□

Dynamic Apex Classes

Dynamic Apex extends your programming reach by interacting with Lightning Platform components.

View: All Create New View

Class Name ↑	Namespace Prefix	Api Version	Created By	Last Modified By
No records to display.				

4.2 Test Methods Work

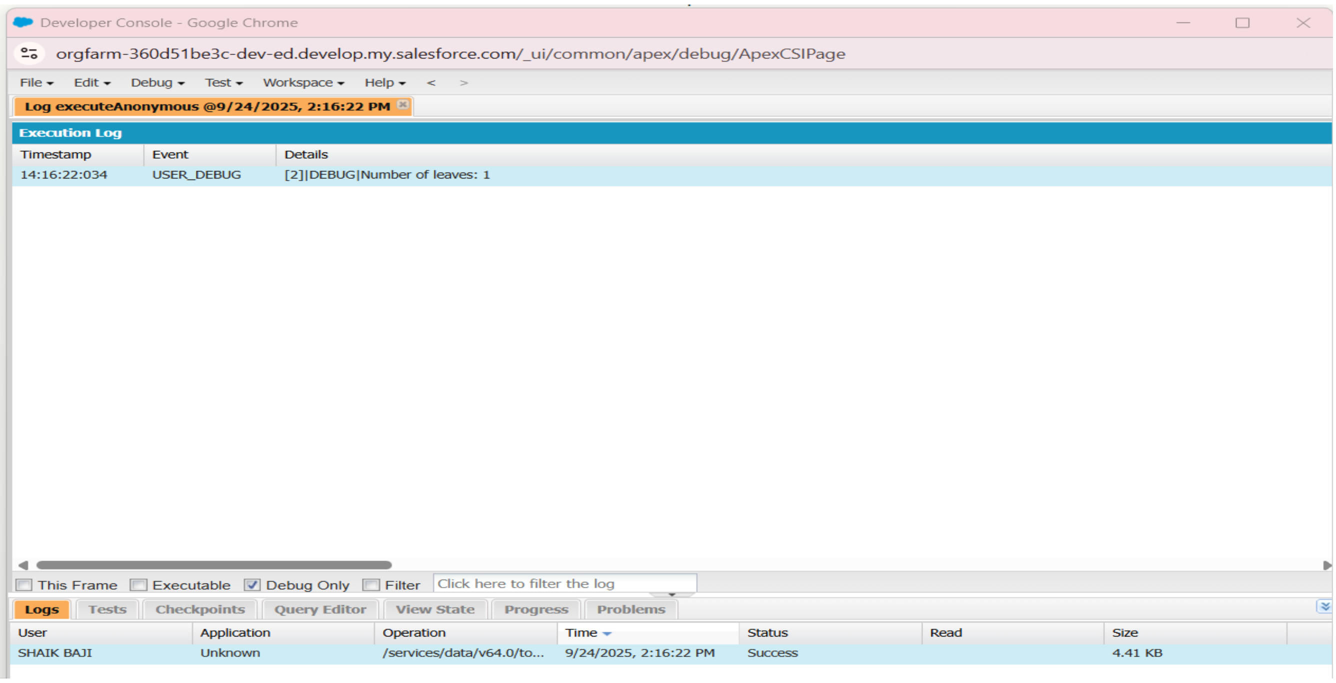
Go to Developer Console in Salesforce

Execute Anonymous Code:

List&lt;LeaveRequest__c&gt; leaves = LeaveRequestController.getMyLeaves();

System.debug('Number of leaves: ' + leaves.size());

Verify it runs without errors


Developer Console - Google Chrome
orgfarm-360d51be3c-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File
Edit
Debug
Test
Workspace
Help
Log executeAnonymous @9/24/2025, 2:16:22 PM
Execution Log
□ This Frame
□ Executable
☑ Debug Only
□ Filter
Click here to filter the log
Logs
Tests
Checkpoints
Query Editor
View State
Progress
Problems

Timestamp	Event	Details
14:16:22:034	USER_DEBUG	[2]|DEBUG|Number of leaves: 1

User	Application	Operation	Time	Status	Read	Size
SHAIK BAJI	Unknown	/services/data/v64.0/to...	9/24/2025, 2:16:22 PM	Success		4.41 KB

Outcomes:

- Success Criteria for Phase 3
- LeaveRequestController.cls deployed successfully
- LeaveRequestControllerTest.cls deployed successfully
- All tests passing with 75%+ code coverage
- Apex methods working when tested
- No deployment errors in VS Code

What's Next?

Phase 4 :Create the Lightning Web Component User Interface