**Phase 5: Apex Programming**

**Objective:**

Automate backend logic such as **updating leave balances, sending notifications, and enforcing approval rules** using Apex.

**Step 1: Create Apex Class – LeaveCalculator**

**Purpose:** Calculate remaining leave for employees.

**Steps:**

1. Developer Console → **File → New → Apex Class**
2. Enter Name: LeaveCalculator
3. Paste code:

public class LeaveCalculator {

public static Integer calculateRemaining(Integer totalLeave, Integer usedLeave) {

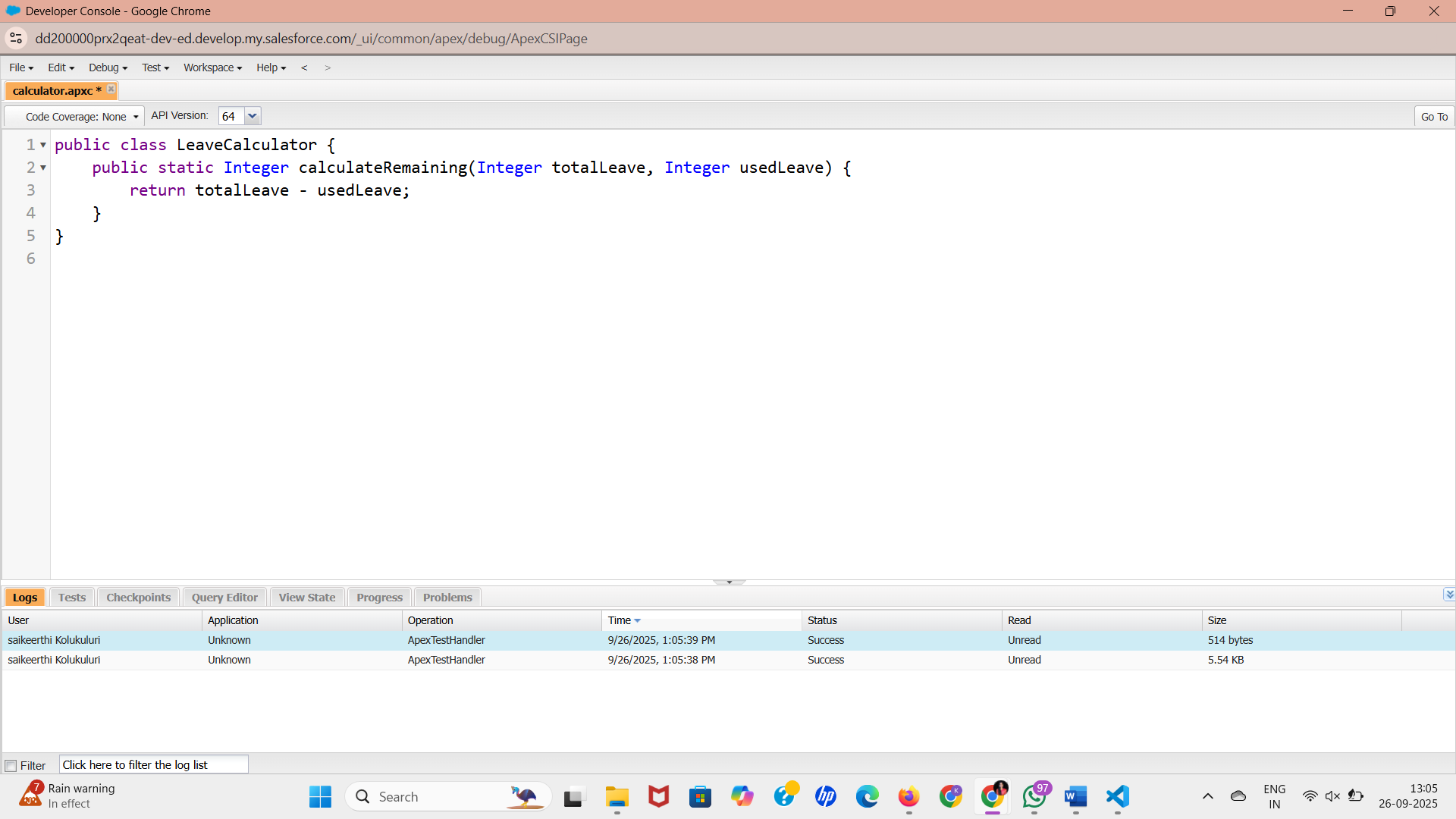
return totalLeave - usedLeave;

}

}

1. Click **Save** → Compiles automatically

**Use:** Can be called by triggers or other classes.



**Step 2: Create Apex Trigger – Update Leave Balance**

**Purpose:** Automatically update the leave balance when a leave request is approved.

**Steps:**

1. Developer Console → **File → New → Apex Trigger**
2. Select Object: Leave\_Request\_\_c → Name: LeaveRequestTrigger
3. Paste code:

trigger LeaveRequestTrigger on Leave\_Request\_\_c (after update) {

for (Leave\_Request\_\_c lr : Trigger.new) {

if (lr.Status\_\_c == 'Approved') {

Leave\_Balance\_\_c lb = [SELECT Id, Remaining\_Leave\_\_c

FROM Leave\_Balance\_\_c

WHERE Employee\_\_c = :lr.Employee\_\_c

LIMIT 1];

lb.Remaining\_Leave\_\_c -= lr.Total\_Days\_\_c;

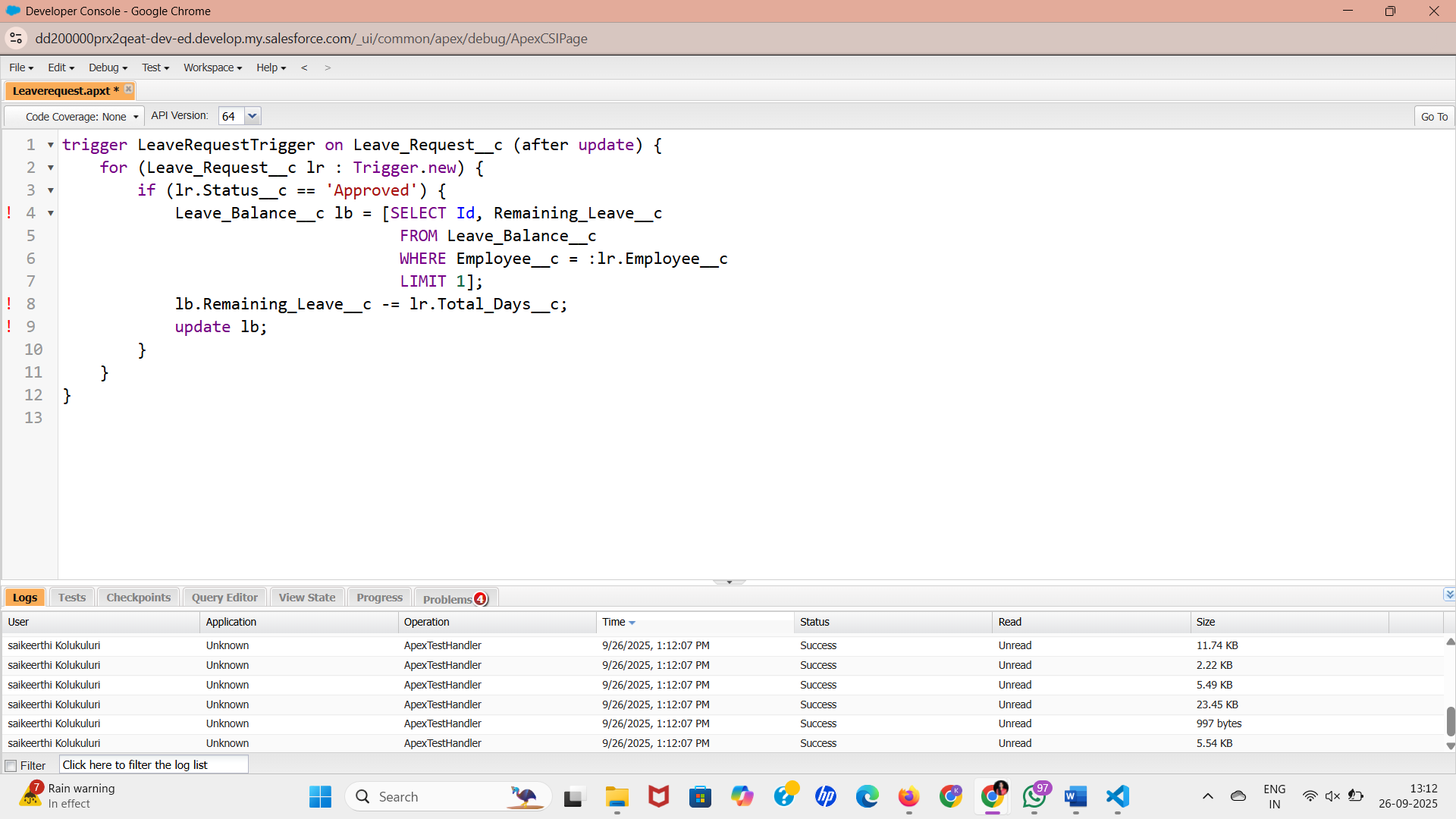
update lb;

}

}

}

1. Click **Save** → Compiles automatically
2. Test by changing a leave request **Status → Approved** → Leave\_Balance updates



**Step 3: Create Test Class**

**Purpose:** Salesforce requires at least **75% code coverage** for deployment.

**Steps:**

1. Developer Console → **File → New → Apex Class** → Name: LeaveRequestTest
2. Paste code:

@isTest

public class LeaveRequestTest {

@isTest static void testLeaveApproval(){

User emp = [SELECT Id FROM User LIMIT 1];

Leave\_Request\_\_c lr = new Leave\_Request\_\_c(

Employee\_\_c = emp.Id,

Total\_Days\_\_c = 2,

Status\_\_c = 'Pending'

);

insert lr;

lr.Status\_\_c = 'Approved';

update lr;

// Verify update

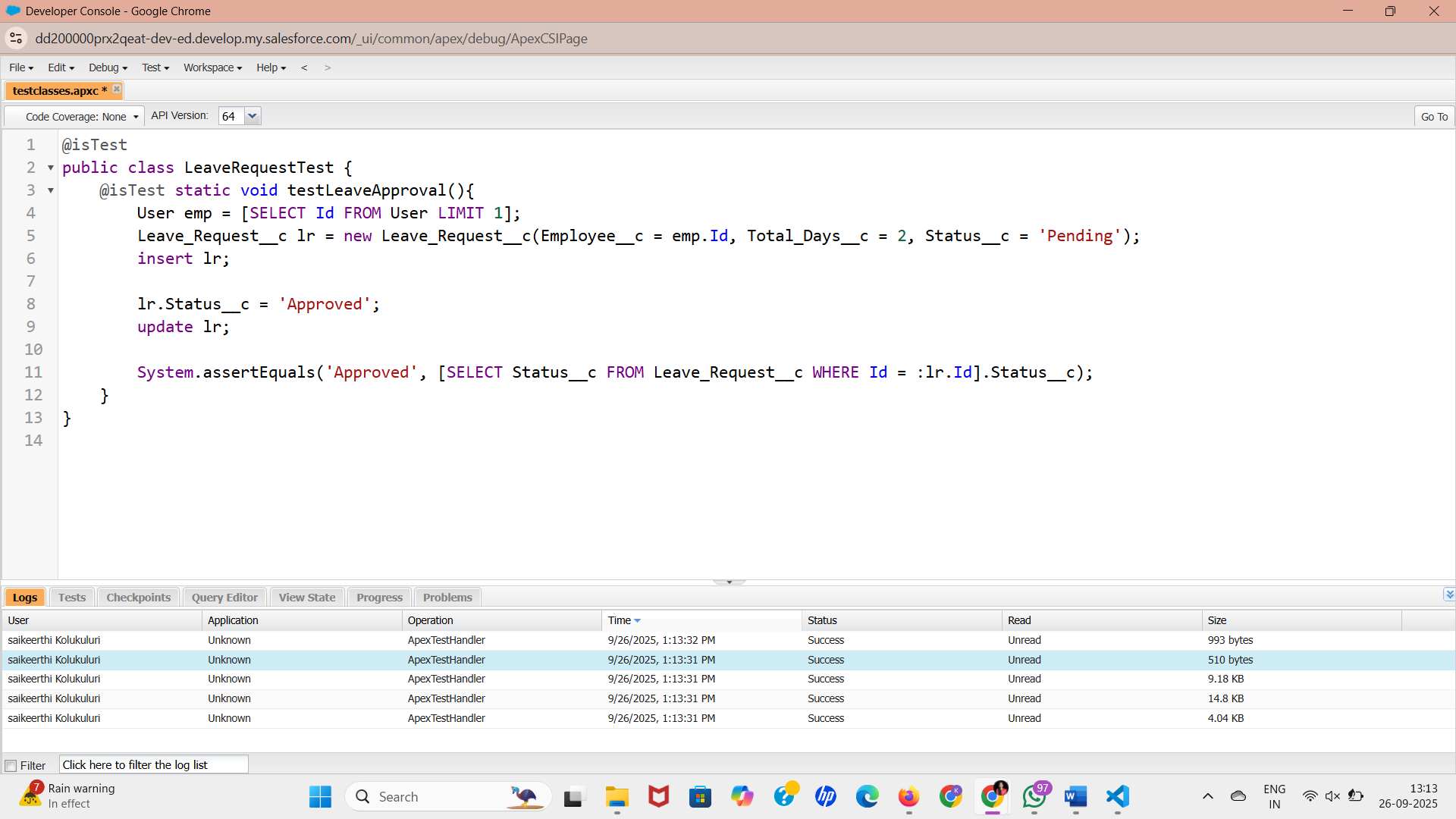
Leave\_Request\_\_c updatedLR = [SELECT Status\_\_c FROM Leave\_Request\_\_c WHERE Id = :lr.Id];

System.assertEquals('Approved', updatedLR.Status\_\_c);

}

}

1. Save → Compile
2. Run Test: **Test → New Run → Select LeaveRequestTest → Run**
3. Check Logs for **pass/fail**



**Step 4: Optional – Asynchronous Notification (Queueable Apex)**

**Purpose:** Send email to employee after approval without slowing system.

**Steps:**

1. Developer Console → File → New → Apex Class → Name: LeaveNotificationQueue
2. Paste code:

public class LeaveNotificationQueue implements Queueable {

public Leave\_Request\_\_c leaveRequest;

public LeaveNotificationQueue(Leave\_Request\_\_c lr){

this.leaveRequest = lr;

}

public void execute(QueueableContext context){

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

mail.setToAddresses(new String[]{leaveRequest.Employee\_\_r.Email});

mail.setSubject('Leave Approved');

mail.setPlainTextBody('Your leave request has been approved.');

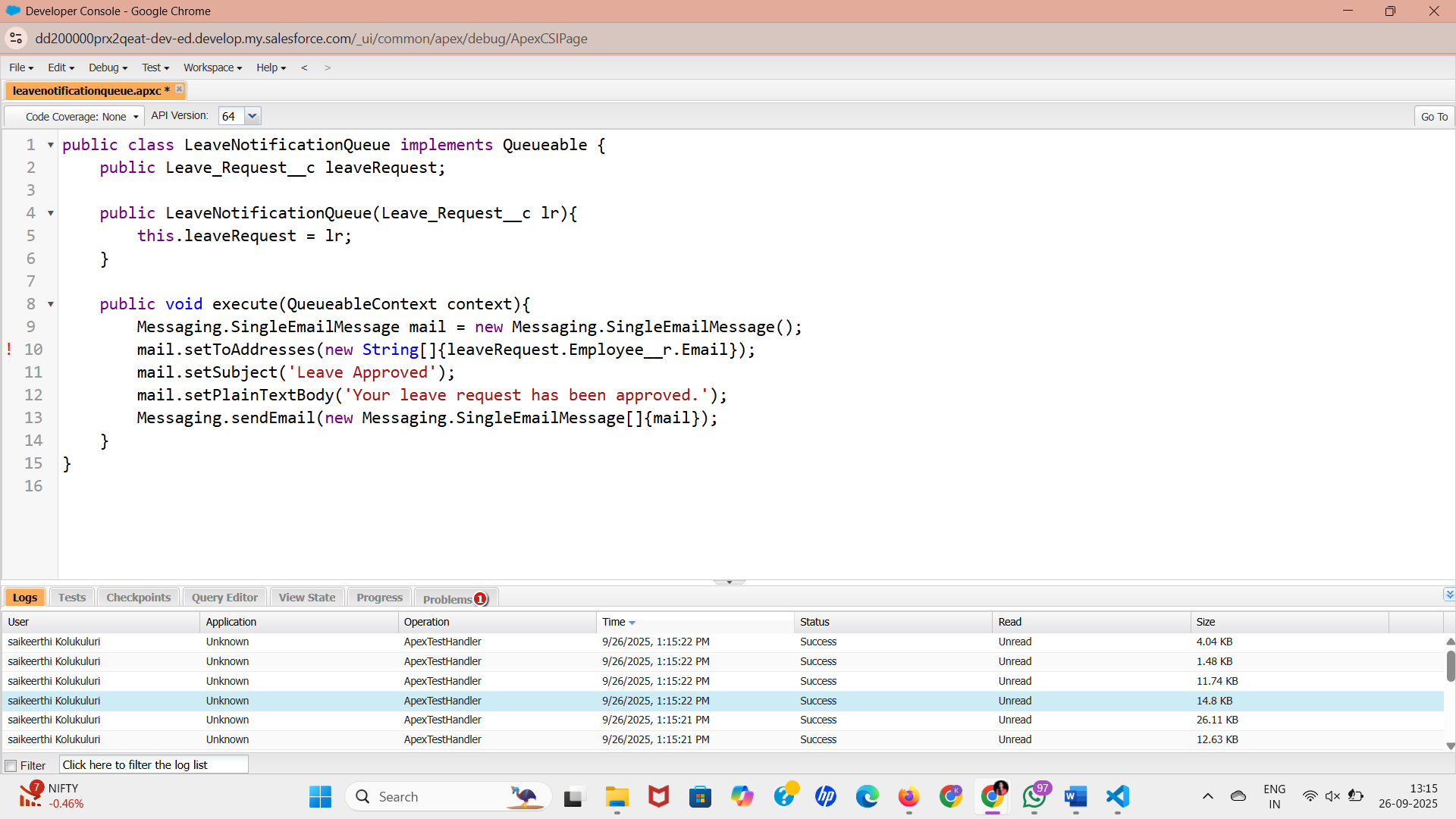
Messaging.sendEmail(new Messaging.SingleEmailMessage[]{mail});

}

}

1. Save → Compile
2. Call in trigger after leave approval:

System.enqueueJob(new LeaveNotificationQueue(lr));



**Step 5: Debugging & Logs**

* Developer Console → **Logs → Check Execution**
* Use System.debug() to print variable values during testing

System.debug('Remaining Leave: ' + lb.Remaining\_Leave\_\_c);