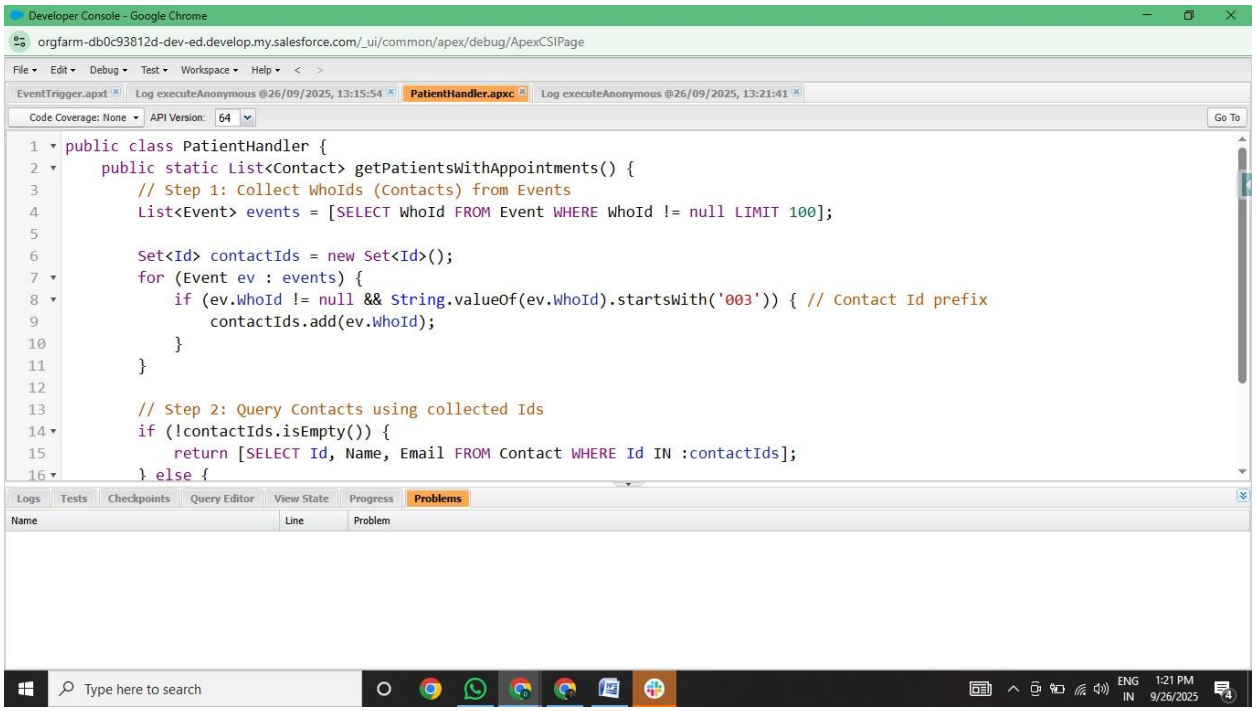
Phase 5: Apex Programming

Task 1: Apex Class

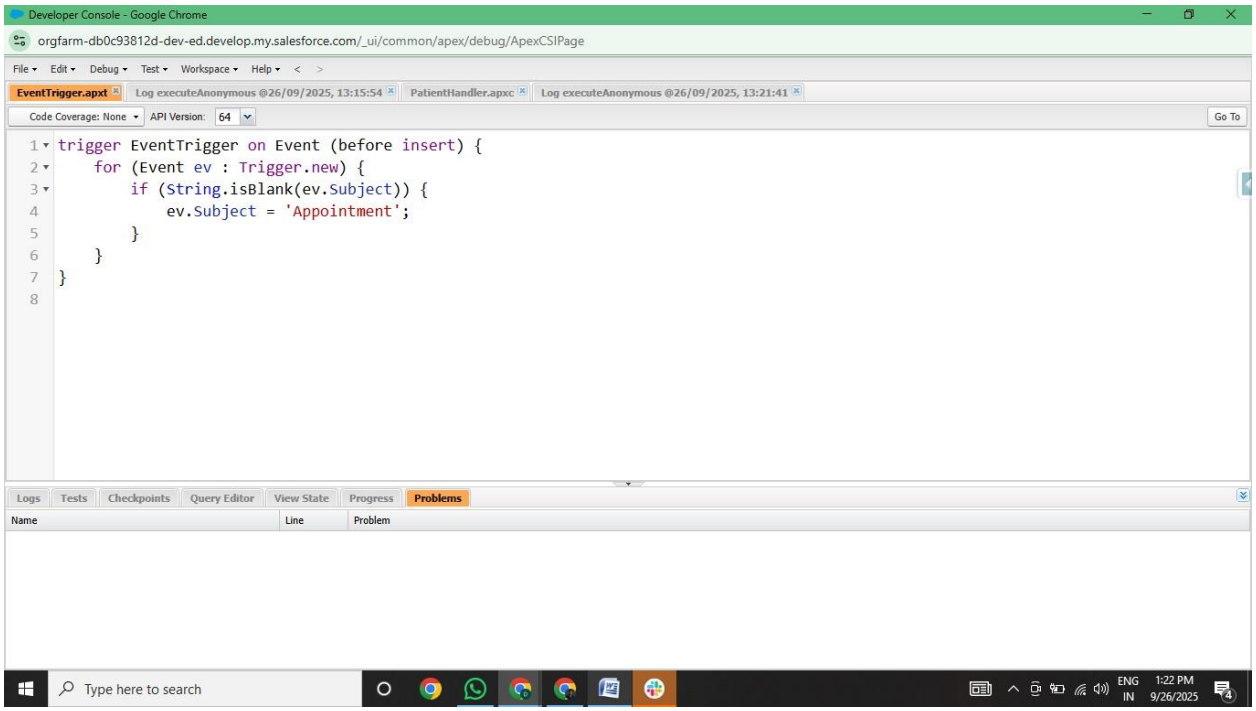
```
public class PatientHandler {

    public static List<Contact> getPatientsWithAppointments() {

        // Step 1: Collect WhoIds (Contacts) from Events

        List<Event> events = [SELECT WhoId FROM Event WHERE WhoId != null LIMIT 100];



        Set<Id> contactIds = new Set<Id>();

        for (Event ev : events) {

            if (ev.WhoId != null && String.valueOf(ev.WhoId).startsWith('003')) { // Contact Id
prefix

                contactIds.add(ev.WhoId);

            }

        }



        // Step 2: Query Contacts using collected Ids

        if (!contactIds.isEmpty()) {

            return [SELECT Id, Name, Email FROM Contact WHERE Id IN :contactIds];

        } else {

            return new List<Contact>();

        }

    }
```

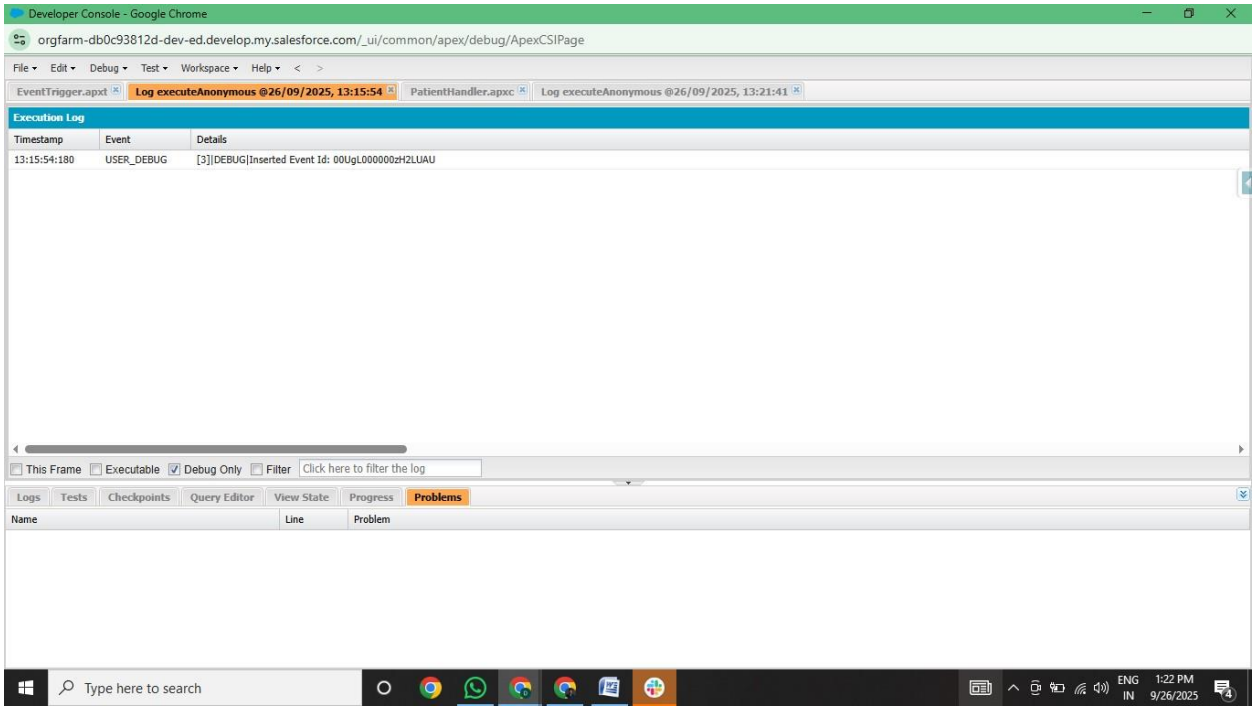
}



Task 2: Apex Trigger

```
trigger EventTrigger on Event (before insert) {
    for (Event ev : Trigger.new) {
        if (String.isBlank(ev.Subject)) {
            ev.Subject = 'Appointment';
        }
    }
}
```



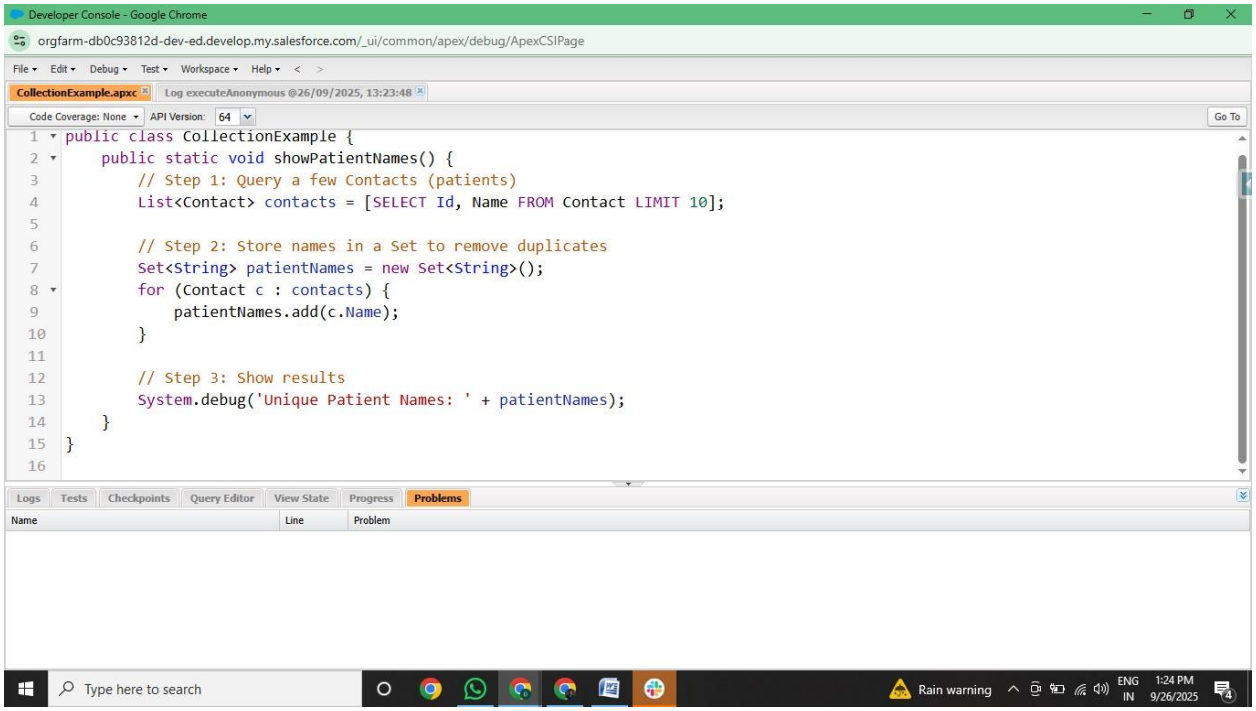


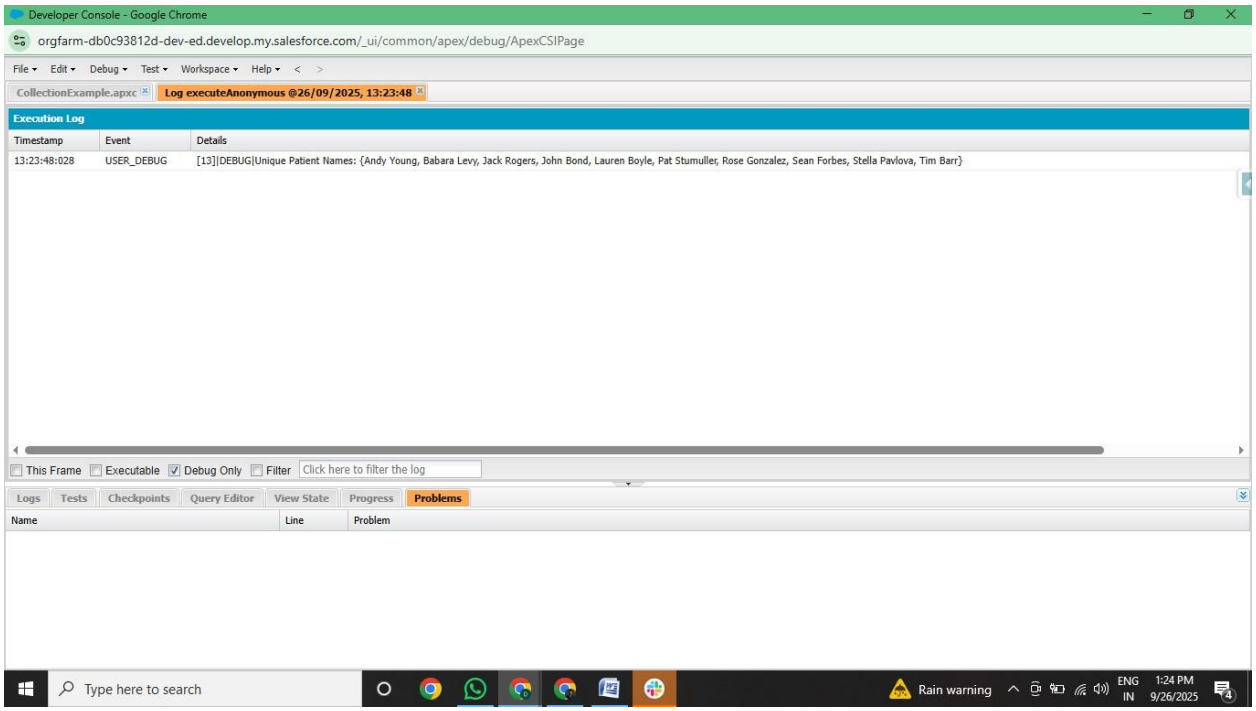
Task 3: Collections Example

Store patient names in a list and debug them.

```
public class CollectionExample {
    public static void showPatientNames() {
        // Step 1: Query a few Contacts (patients)
```

```
        List<Contact> contacts = [SELECT Id, Name FROM Contact LIMIT 10];

        // Step 2: Store names in a Set to remove duplicates
        Set<String> patientNames = new Set<String>();
        for (Contact c : contacts) {
            patientNames.add(c.Name);
        }

        // Step 3: Show results
        System.debug('Unique Patient Names: ' + patientNames);
    }
}
```

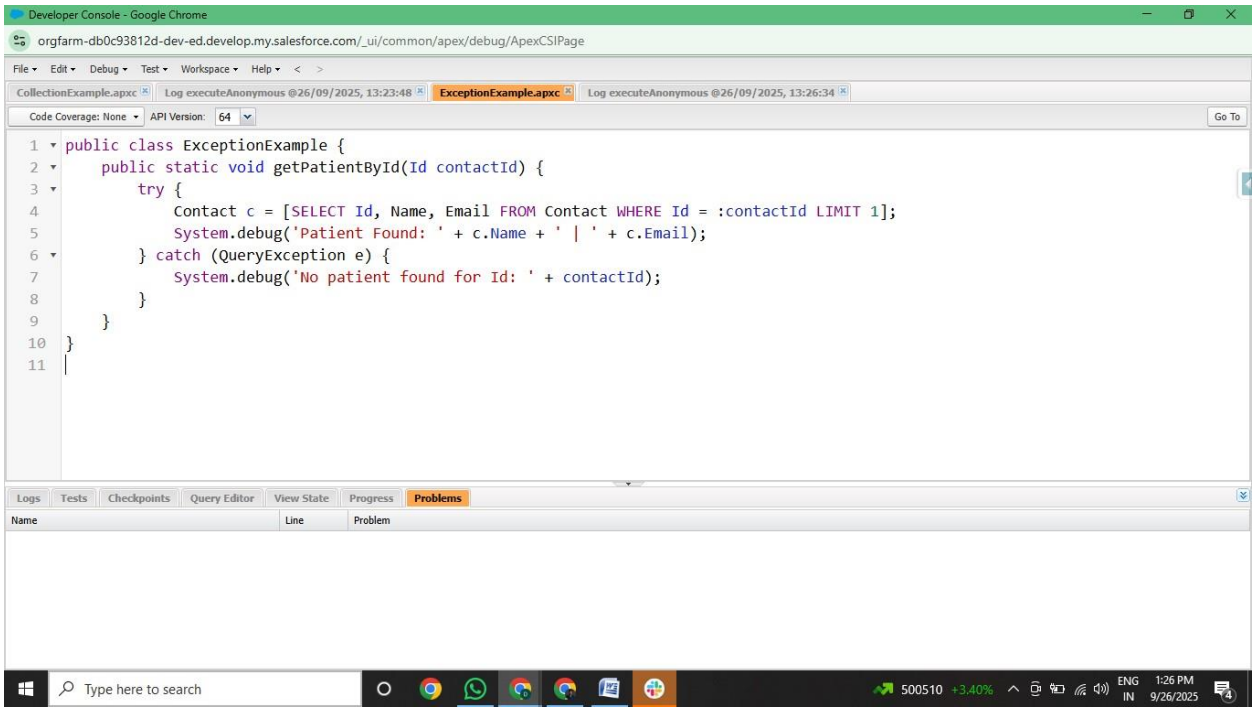


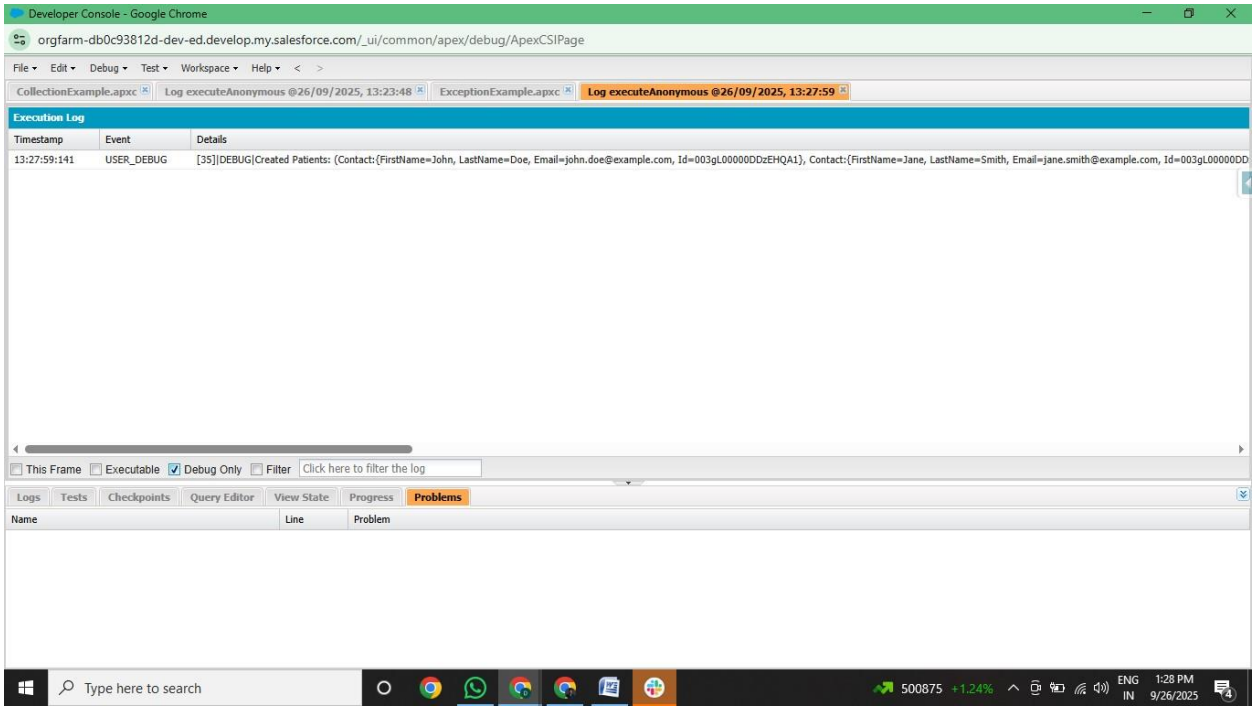


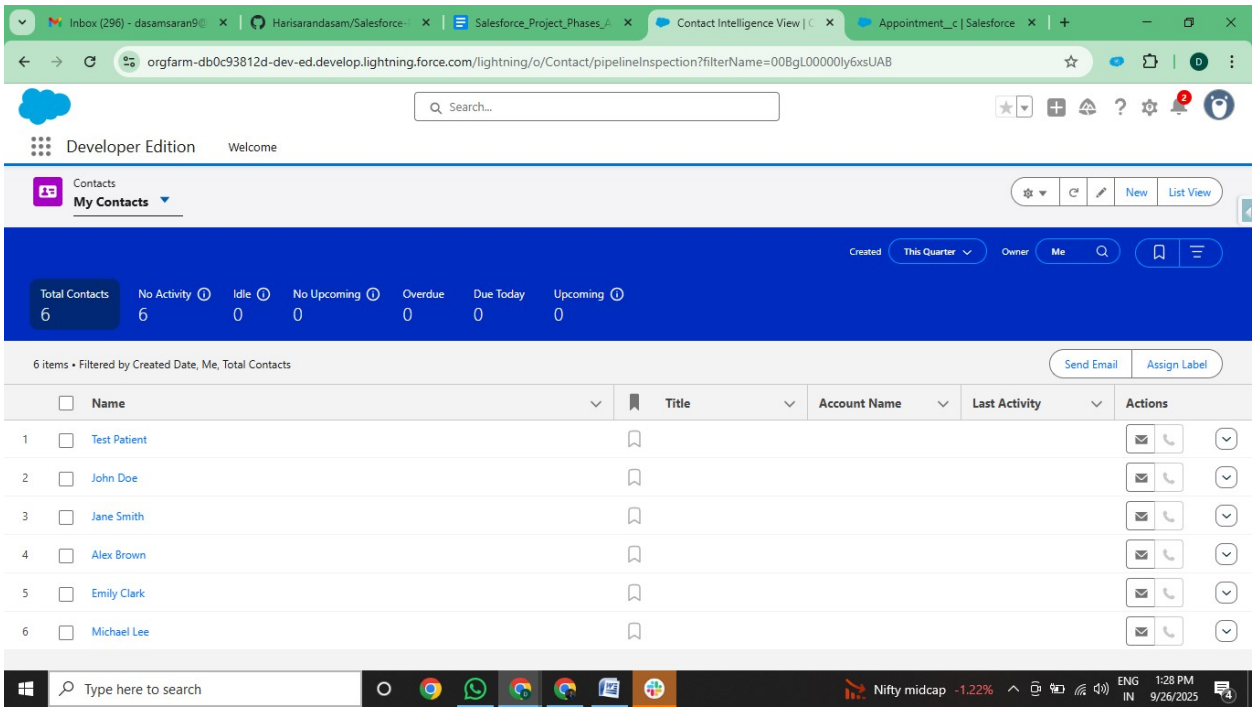
Task 4: Exception Handling

Handle missing records gracefully.

```
public class ExceptionExample {
    public static void getPatientById(Id contactId) {
        try {
            Contact c = [SELECT Id, Name, Email FROM Contact WHERE Id =
:contactId LIMIT 1];
            System.debug('Patient Found: ' + c.Name + ' | ' + c.Email);
        } catch (QueryException e) {
            System.debug('No patient found for Id: ' + contactId);
        }
    }
}
```


Developer Console - Google Chrome
orgfarm-db0c93812d-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
CollectionExample.apxc
Log executeAnonymous @26/09/2025, 13:23:48
ExceptionExample.apxc
Log executeAnonymous @26/09/2025, 13:26:34
Code Coverage: None
API Version: 64
Go To
public class ExceptionExample {
public static void getPatientById(Id contactId) {
try {
Contact c = [SELECT Id, Name, Email FROM Contact WHERE Id = :contactId LIMIT 1];
System.debug('Patient Found: ' + c.Name + ' | ' + c.Email);
} catch (QueryException e) {
System.debug('No patient found for Id: ' + contactId);
}
}
}
Logs
Tests
Checkpoints
Query Editor
View State
Progress
Problems
Name
Line
Problem
Type here to search
500510 +3.40%
ENG
IN
1:26 PM
9/26/2025


Developer Console - Google Chrome
orgfarm-db0c93812d-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
CollectionExample.apxc
Log executeAnonymous @26/09/2025, 13:23:48
ExceptionExample.apxc
Log executeAnonymous @26/09/2025, 13:27:59
Execution Log
Timestamp
Event
Details
13:27:59:141
USER_DEBUG
[35]|DEBUG|Created Patients: (Contact:{FirstName=John, LastName=Doe, Email=john.doe@example.com, Id=003gL00000DDzEHQA1}, Contact:{FirstName=Jane, LastName=Smith, Email=jane.smith@example.com, Id=003gL00000DD
This Frame
Executable
Debug Only
Filter
Click here to filter the log
Logs
Tests
Checkpoints
Query Editor
View State
Progress
Problems
Name
Line
Problem
Type here to search
500875 +1.24%
ENG
IN
1:28 PM
9/26/2025



Task 5: Test Class

```
@isTest
public class EventTriggerTest {
    @isTest
    static void testEventInsert() {
        Test.startTest();
            // Create a new Event (Appointment)
            Event ev = new Event(
                StartDateTime = System.now().addHours(1),
                EndDateTime   = System.now().addHours(2)
            );
            insert ev;
        Test.stopTest();

        // Verify trigger logic worked
        Event evDb = [SELECT Id, Subject FROM Event WHERE Id = :ev.Id];
        System.assertEquals('Appointment', evDb.Subject,
            '✅ Subject should be auto-set to Appointment by trigger');
    }
}
```