Project: Disaster Relief Resource Management CRM (ReliefConnect)

PHASE 5: Apex Programming (Developer)

🎯 Executive Summary

Phase 5 marks the transition from declarative configuration to programmatic development with Apex. The objective was to build a robust, scalable, and efficient backend architecture to handle complex business logic that is beyond the scope of declarative tools. During this phase, I implemented a best-practice trigger framework, created service classes to encapsulate logic, and built asynchronous processes to handle large data volumes and long-running operations. This work ensures the ReliefConnect application is not only intelligent but also performs reliably under load.

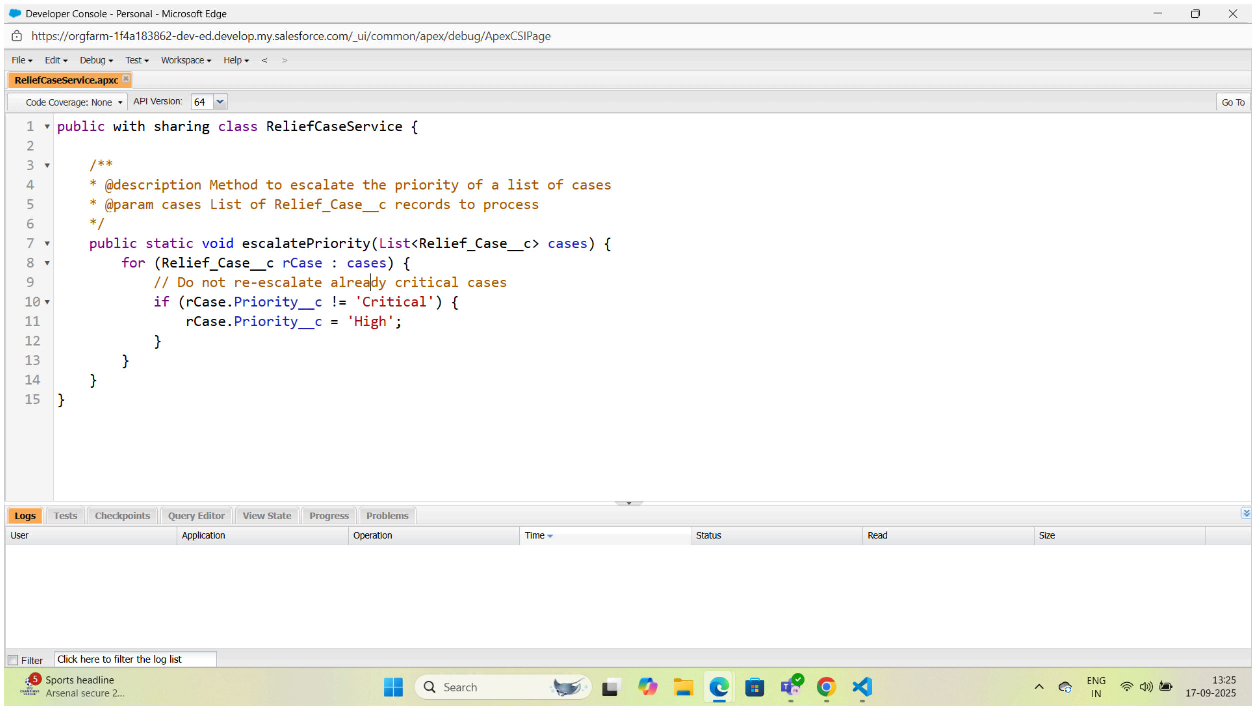
🏛️ Classes & Objects

I defined Apex classes to act as blueprints for objects and to contain the application's core business logic, separating it from the trigger invocation for better maintainability.

- **Implementation:** A primary service class, `ReliefCaseService`, was created to contain reusable methods related to `Relief_Case__c` records. This encapsulates logic that can be called from triggers, batch jobs, or other parts of the system.
- **Code Example:**

Apex

```
// Service class to handle business logic for Relief Cases
public with sharing class ReliefCaseService {

    // Method to escalate the priority of a list of cases
    public static void escalatePriority(List<Relief_Case__c> cases) {
        for (Relief_Case__c rCase : cases) {
            // Do not re-escalate already critical cases
            if (rCase.Priority__c != 'Critical') {
                rCase.Priority__c = 'High';
            }
        }
    }
```



Developer Console - Personal - Microsoft Edge
https://orgfarm-1f4a183862-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
ReliefCaseService.apxc
Code Coverage: None
API Version: 64
Go To
Logs Tests Checkpoints Query Editor View State Progress Problems
User Application Operation Time Status Read Size
Filter Click here to filter the log list

```
public with sharing class ReliefCaseService {

    /**
    * @description Method to escalate the priority of a list of cases
    * @param cases List of Relief_Case__c records to process
    */
    public static void escalatePriority(List<Relief_Case__c> cases) {
        for (Relief_Case__c rCase : cases) {
            // Do not re-escalate already critical cases
            if (rCase.Priority__c != 'Critical') {
                rCase.Priority__c = 'High';
            }
        }
    }
}
```

⚡ Apex Triggers & Trigger Design Pattern

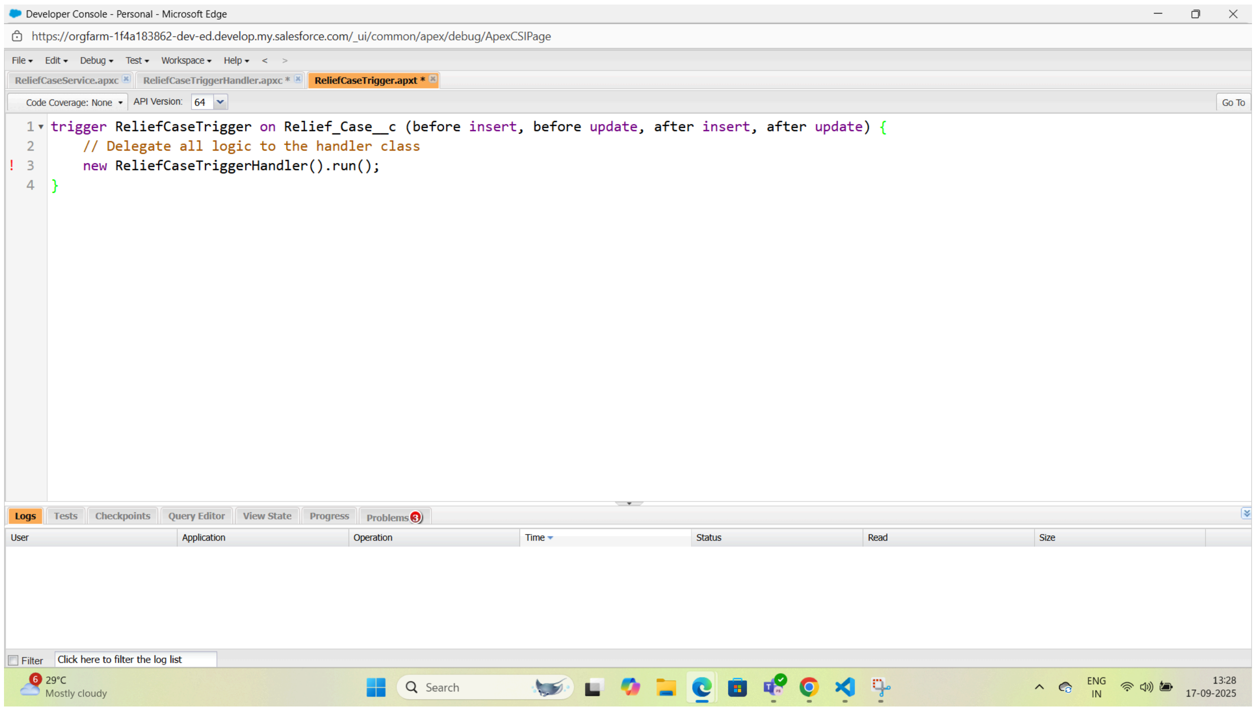
I implemented a "one-trigger-per-object" framework to manage all operations on Relief_Case__c. This pattern prevents unpredictable execution order and makes the code base manageable. The trigger's sole responsibility is to delegate the logic to a dedicated handler class.

- **Implementation:** A single trigger, ReliefCaseTrigger, was created on the Relief_Case__c object. This trigger calls the ReliefCaseTriggerHandler class, passing the trigger context variables (Trigger.new, Trigger.oldMap, etc.).

- **Code Example (ReliefCaseTrigger):**

Apex

```
trigger ReliefCaseTrigger on Relief_Case__c (before insert, before update, after insert, after update)
{
    // Delegate all logic to the handler class
    new ReliefCaseTriggerHandler().run();
}
```


Developer Console - Personal - Microsoft Edge
https://orgfarm-1f4a183862-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
ReliefCaseService.apxc
ReliefCaseTriggerHandler.apxc *
ReliefCaseTrigger.apxt *
Code Coverage: None
API Version: 64
Go To
Logs
Tests
Checkpoints
Query Editor
View State
Progress
Problems 3
User
Application
Operation
Time
Status
Read
Size
Filter
Click here to filter the log list
29°C
Mostly cloudy
Search
ENG
IN
13:28
17-09-2025

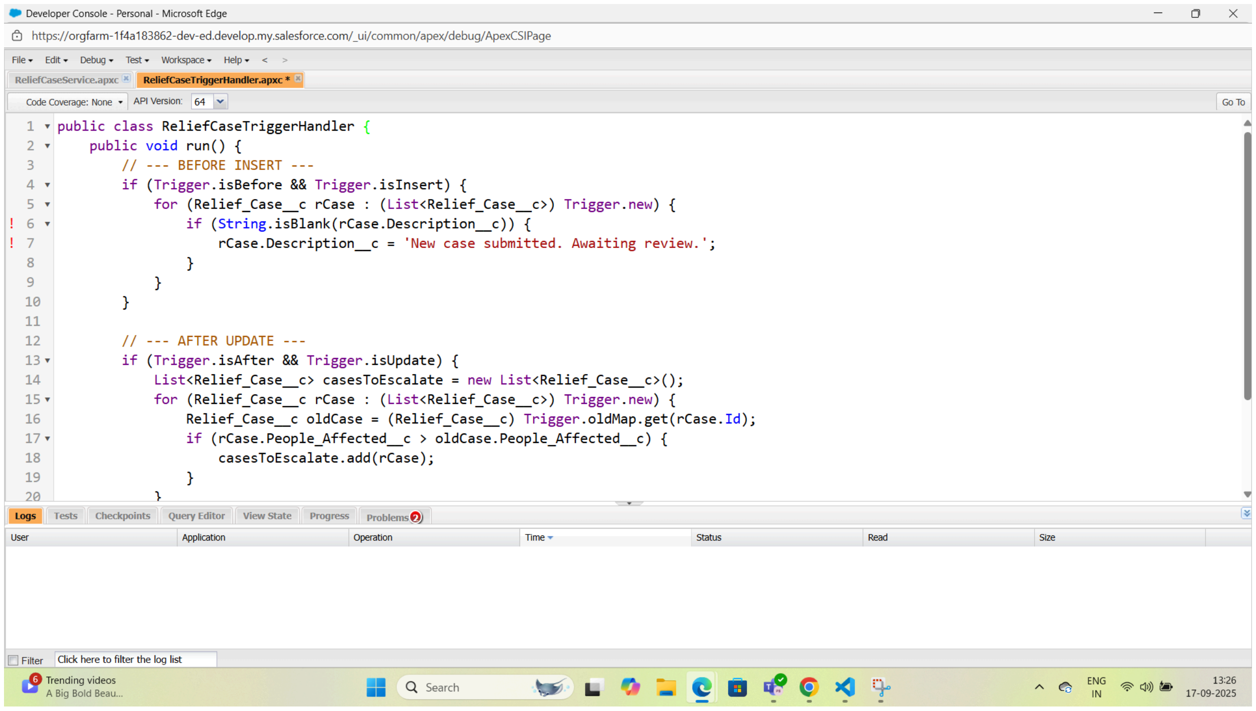
```
trigger ReliefCaseTrigger on Relief_Case__c (before insert, before update, after insert, after update) {
    // Delegate all logic to the handler class
    new ReliefCaseTriggerHandler().run();
}
```

- **Code Example (ReliefCaseTriggerHandler):**

Apex

```
public class ReliefCaseTriggerHandler {
    public void run() {
        // --- BEFORE INSERT ---
        if (Trigger.isBefore && Trigger.isInsert) {
            // Example: Set a default description on new cases
            for (Relief_Case__c rCase : (List<Relief_Case__c>) Trigger.new) {
                if (String.isBlank(rCase.Description__c)) {
                    rCase.Description__c = 'New case submitted. Awaiting review.';
                }
            }
        }

        // --- AFTER UPDATE ---
        if (Trigger.isAfter && Trigger.isUpdate) {
            // Example: Escalate priority if # of people increases
            List<Relief_Case__c> casesToEscalate = new List<Relief_Case__c>();
            for (Relief_Case__c rCase : (List<Relief_Case__c>) Trigger.new) {
                Relief_Case__c oldCase = (Relief_Case__c) Trigger.oldMap.get(rCase.Id);
                if (rCase.People_Affected__c > oldCase.People_Affected__c) {
```

```
                casesToEscalate.add(rCase);
            }
        }
        if (!casesToEscalate.isEmpty()) {
            ReliefCaseService.escalatePriority(casesToEscalate);
        }
    }
  }
}
```


Developer Console - Personal - Microsoft Edge
https://orgfarm-1f4a183862-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
ReliefCaseService.apxc
ReliefCaseTriggerHandler.apxc *
Code Coverage: None
API Version: 64
Go To
Logs
Tests
Checkpoints
Query Editor
View State
Progress
Problems 2
User
Application
Operation
Time
Status
Read
Size
Filter
Click here to filter the log list

```
public class ReliefCaseTriggerHandler {
    public void run() {
        // --- BEFORE INSERT ---
        if (Trigger.isBefore && Trigger.isInsert) {
            for (Relief_Case__c rCase : (List<Relief_Case__c>) Trigger.new) {
                if (String.isBlank(rCase.Description__c)) {
                    rCase.Description__c = 'New case submitted. Awaiting review.';
                }
            }
        }

        // --- AFTER UPDATE ---
        if (Trigger.isAfter && Trigger.isUpdate) {
            List<Relief_Case__c> casesToEscalate = new List<Relief_Case__c>();
            for (Relief_Case__c rCase : (List<Relief_Case__c>) Trigger.new) {
                Relief_Case__c oldCase = (Relief_Case__c) Trigger.oldMap.get(rCase.Id);
                if (rCase.People_Affected__c > oldCase.People_Affected__c) {
                    casesToEscalate.add(rCase);
                }
            }
```

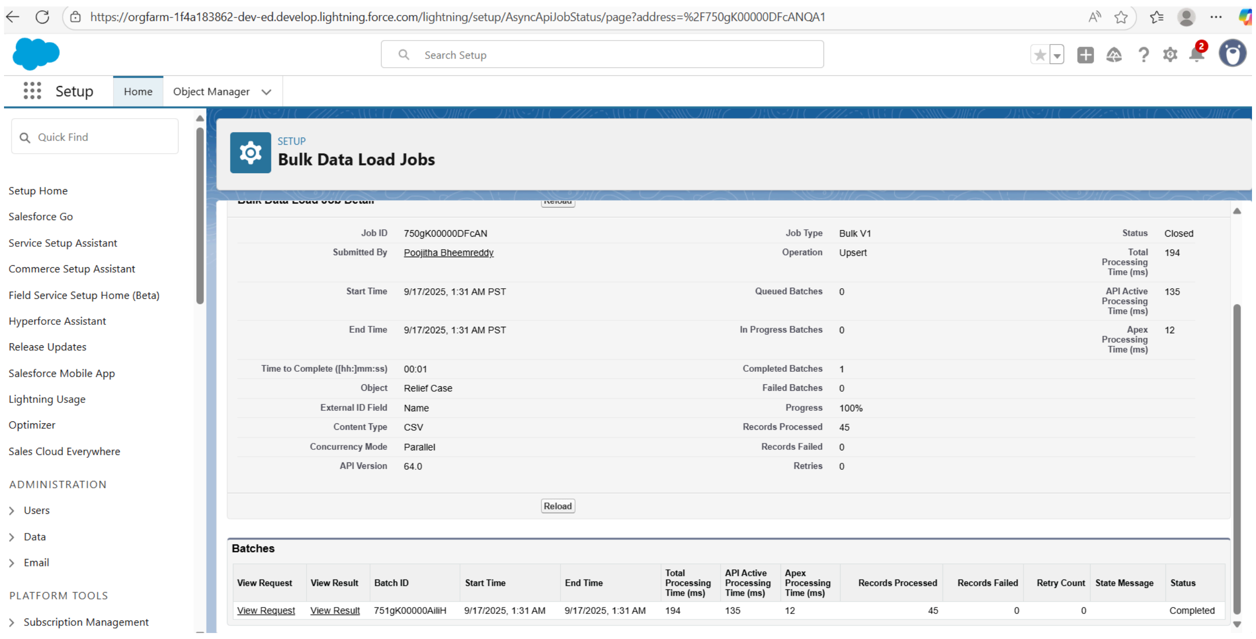
🔍 SOQL & SOSL

I used Salesforce Object Query Language (SOQL) for precise data retrieval and Salesforce Object Search Language (SOSL) for broad, multi-object text searches.

- **SOQL Example:** Used within a method to fetch all non-fulfilled relief cases.

Apex

```
List<Relief_Case__c> openCases = [
    SELECT Id, Name, Status__c, Priority__c
    FROM Relief_Case__c
    WHERE Status__c != 'Fulfilled'
    ORDER BY CreatedDate DESC
];
```

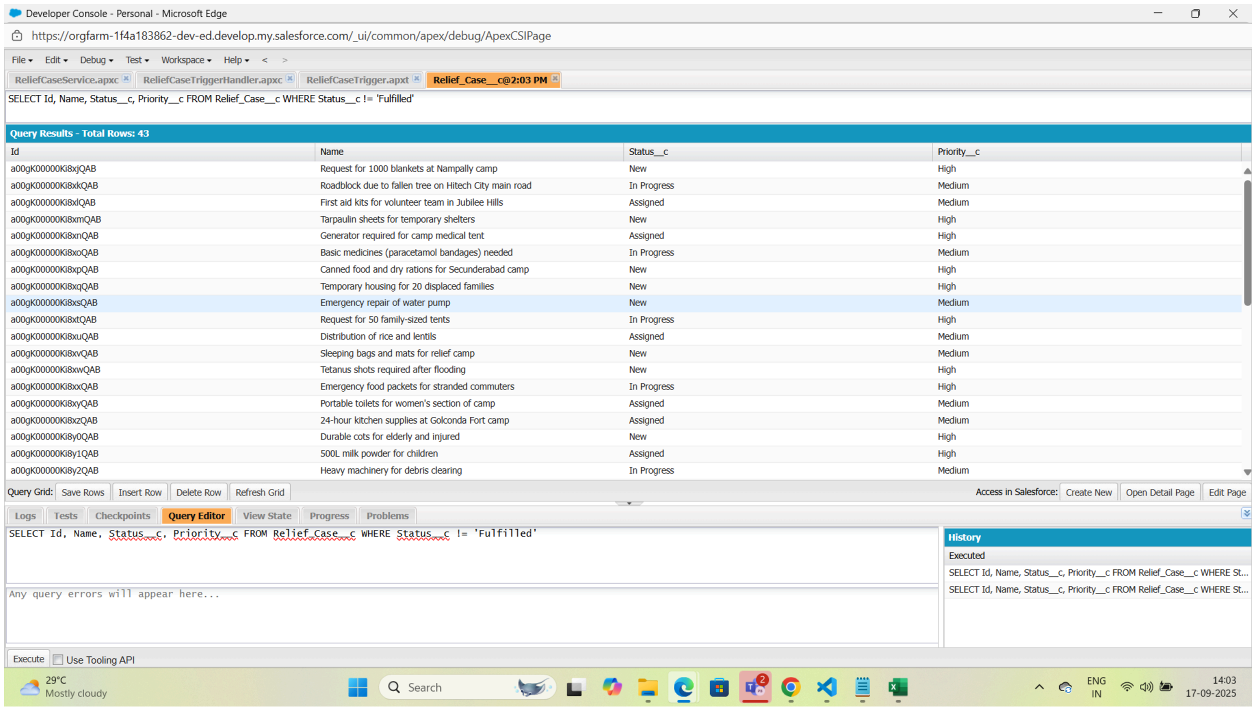

https://orgfarm-1f4a183862-dev-ed.develop.lightning.force.com/lightning/setup/AsyncApiJobStatus/page?address=%2F750gK00000DFcANQA1

SETUP
Bulk Data Load Jobs

Job ID	750gK00000DFcAN	Job Type	Bulk V1	Status	Closed
Submitted By	Poojitha Bheemreddy	Operation	Upsert	Total Processing Time (ms)	194
Start Time	9/17/2025, 1:31 AM PST	Queued Batches	0	API Active Processing Time (ms)	135
End Time	9/17/2025, 1:31 AM PST	In Progress Batches	0	Apex Processing Time (ms)	12
Time to Complete ([hh:]mm:ss)	00:01	Completed Batches	1		
Object	Relief Case	Failed Batches	0		
External ID Field	Name	Progress	100%		
Content Type	CSV	Records Processed	45		
Concurrency Mode	Parallel	Records Failed	0		
API Version	64.0	Retries	0		

Batches

View Request	View Result	Batch ID	Start Time	End Time	Total Processing Time (ms)	API Active Processing Time (ms)	Apex Processing Time (ms)	Records Processed	Records Failed	Retry Count	State Message	Status
View Request	View Result	751gK00000AiliH	9/17/2025, 1:31 AM	9/17/2025, 1:31 AM	194	135	12	45	0	0		Completed


Developer Console - Personal - Microsoft Edge
https://orgfarm-1f4a183862-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

ReliefCaseService.apxc | ReliefCaseTriggerHandler.apxc | ReliefCaseTrigger.apxt | Relief_Case__c@2:03 PM

```
SELECT Id, Name, Status__c, Priority__c FROM Relief_Case__c WHERE Status__c != 'Fulfilled'
```

Query Results - Total Rows: 43

Id	Name	Status__c	Priority__c
a00gK00000Ki8xjQAB	Request for 1000 blankets at Nampally camp	New	High
a00gK00000Ki8xkQAB	Roadblock due to fallen tree on Hitech City main road	In Progress	Medium
a00gK00000Ki8xlQAB	First aid kits for volunteer team in Jubilee Hills	Assigned	Medium
a00gK00000Ki8xmQAB	Tarpaulin sheets for temporary shelters	New	High
a00gK00000Ki8xnQAB	Generator required for camp medical tent	Assigned	High
a00gK00000Ki8xoQAB	Basic medicines (paracetamol bandages) needed	In Progress	Medium
a00gK00000Ki8xpQAB	Canned food and dry rations for Secunderabad camp	New	High
a00gK00000Ki8xqQAB	Temporary housing for 20 displaced families	New	High
a00gK00000Ki8xsQAB	Emergency repair of water pump	New	Medium
a00gK00000Ki8xtQAB	Request for 50 family-sized tents	In Progress	High
a00gK00000Ki8xuQAB	Distribution of rice and lentils	Assigned	Medium
a00gK00000Ki8xvQAB	Sleeping bags and mats for relief camp	New	Medium
a00gK00000Ki8xwQAB	Tetanus shots required after flooding	New	High
a00gK00000Ki8xxQAB	Emergency food packets for stranded commuters	In Progress	High
a00gK00000Ki8xyQAB	Portable toilets for women's section of camp	Assigned	Medium
a00gK00000Ki8xzQAB	24-hour kitchen supplies at Golconda Fort camp	Assigned	Medium
a00gK00000Ki8y0QAB	Durable cots for elderly and injured	New	High
a00gK00000Ki8y1QAB	500L milk powder for children	Assigned	High
a00gK00000Ki8y2QAB	Heavy machinery for debris clearing	In Progress	Medium

Query Editor

```
SELECT Id, Name, Status__c, Priority__c FROM Relief_Case__c WHERE Status__c != 'Fulfilled'
```

Any query errors will appear here...

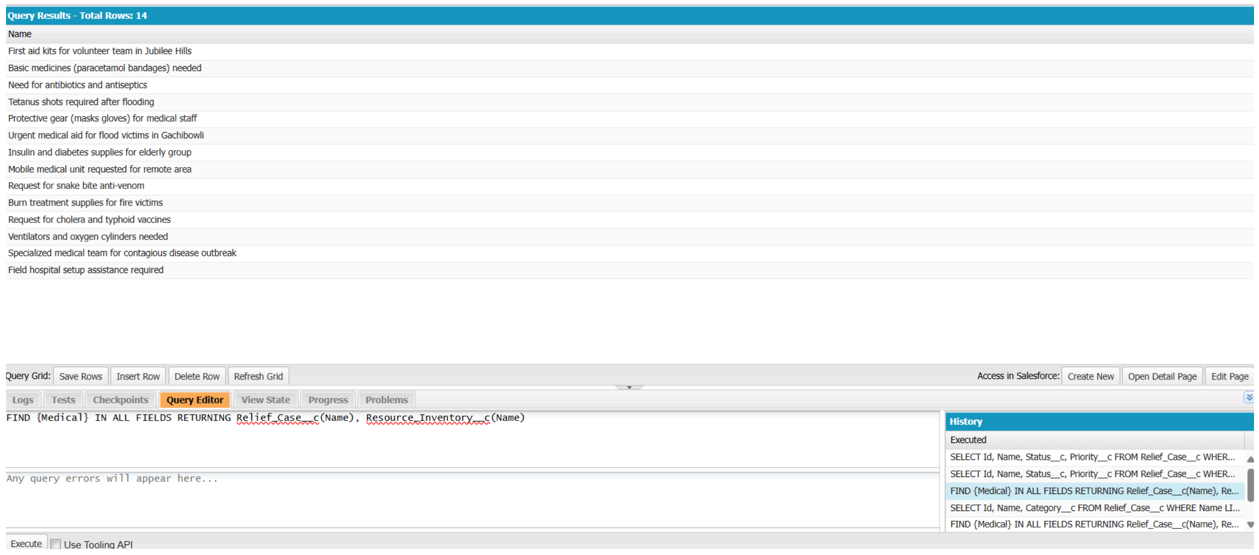
History
Executed
SELECT Id, Name, Status__c, Priority__c FROM Relief_Case__c WHERE St...
SELECT Id, Name, Status__c, Priority__c FROM Relief_Case__c WHERE St...

29°C Mostly cloudy · ENG IN · 14:03 17-09-2025

- **SOSL Example:** Used for a global search functionality to find a keyword across different objects.

Apex

```
String searchText = 'medical supplies';
List<List<SObject>> searchResults = [
    FIND :searchText IN ALL FIELDS
    RETURNING Relief_Case__c(Name), Resource_Inventory__c(Name)
];
```



Query Results - Total Rows: 14

Name
First aid kits for volunteer team in Jubilee Hills
Basic medicines (paracetamol bandages) needed
Need for antibiotics and antiseptics
Tetanus shots required after flooding
Protective gear (masks gloves) for medical staff
Urgent medical aid for flood victims in Gachibowli
Insulin and diabetes supplies for elderly group
Mobile medical unit requested for remote area
Request for snake bite anti-venom
Burn treatment supplies for fire victims
Request for cholera and typhoid vaccines
Ventilators and oxygen cylinders needed
Specialized medical team for contagious disease outbreak
Field hospital setup assistance required

```
FIND {Medical} IN ALL FIELDS RETURNING Relief_Case__c(Name), Resource_Inventory__c(Name)
```

```
Any query errors will appear here...
```

History

Executed
SELECT Id, Name, Status__c, Priority__c FROM Relief_Case__c WHER...
SELECT Id, Name, Status__c, Priority__c FROM Relief_Case__c WHER...
FIND {Medical} IN ALL FIELDS RETURNING Relief_Case__c(Name), Re...
SELECT Id, Name, Category__c FROM Relief_Case__c WHERE Name LI...
FIND {Medical} IN ALL FIELDS RETURNING Relief_Case__c(Name), Re...

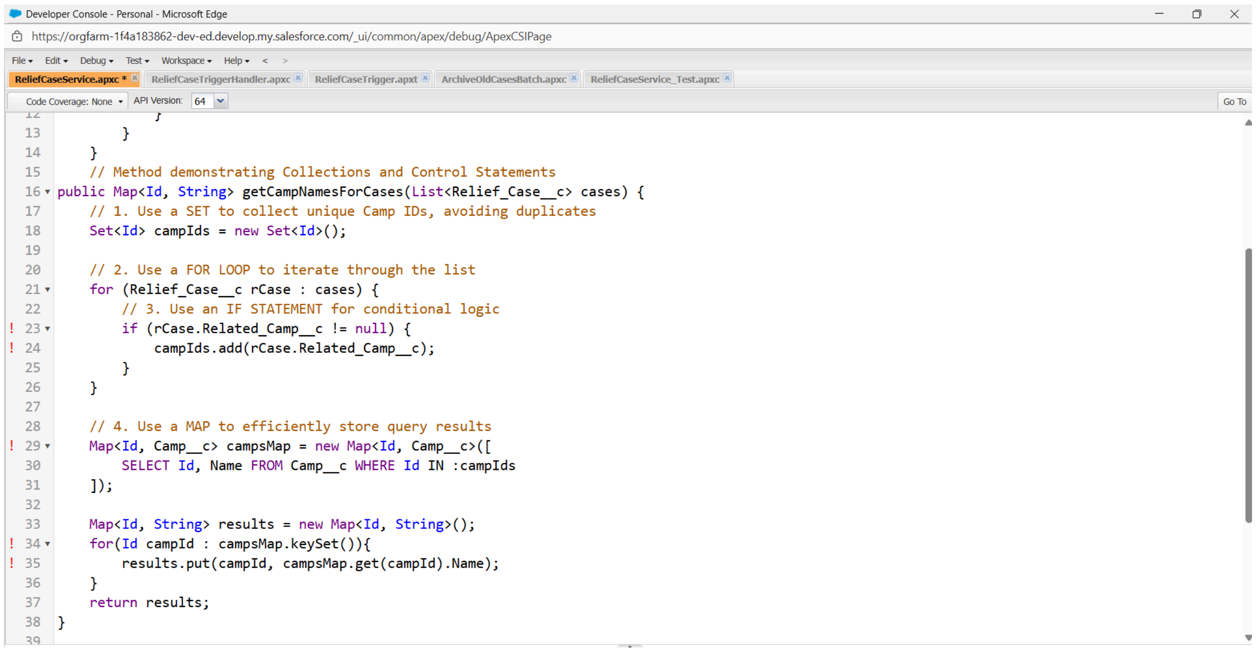
🗂️ Collections & Control Statements

I extensively used collections (**List, Set, Map**) and control statements (**if/else, for loops**) to handle records in bulk and implement conditional logic, which is critical for writing efficient, bulk-safe code.

- **Code Example:** This method demonstrates all three collection types and control statements. It processes a list of cases, gets their related camp details, and returns a map of Camp IDs to Camp Names.

Apex

```
public Map<Id, String> getCampNamesForCases(List<Relief_Case__c> cases) {
    // 1. Use a SET to collect unique Camp IDs, avoiding duplicates
    Set<Id> campIds = new Set<Id>();
    // 2. Use a FOR LOOP to iterate through the list
    for (Relief_Case__c rCase : cases) {
        // 3. Use an IF STATEMENT for conditional logic
        if (rCase.Related_Camp__c != null) {
            campIds.add(rCase.Related_Camp__c);
        }
    }

    // 4. Use a MAP to efficiently store query results
    Map<Id, Camp__c> campsMap = new Map<Id, Camp__c>([
        SELECT Id, Name FROM Camp__c WHERE Id IN :campIds
    ]);

    Map<Id, String> results = new Map<Id, String>();
    for(Id campId : campsMap.keySet()){
        results.put(campId, campsMap.get(campId).Name);
    }

    return results;
}
```


Developer Console - Personal - Microsoft Edge
https://orgfarm-1f4a183862-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File
Edit
Debug
Test
Workspace
Help
ReliefCaseService.apxc *
ReliefCaseTriggerHandler.apxc
ReliefCaseTrigger.apxt
ArchiveOldCasesBatch.apxc
ReliefCaseService_Test.apxc
Code Coverage: None
API Version: 64
Go To

```
            }
        }
    // Method demonstrating Collections and Control Statements
public Map<Id, String> getCampNamesForCases(List<Relief_Case__c> cases) {
    // 1. Use a SET to collect unique Camp IDs, avoiding duplicates
    Set<Id> campIds = new Set<Id>();

    // 2. Use a FOR LOOP to iterate through the list
    for (Relief_Case__c rCase : cases) {
        // 3. Use an IF STATEMENT for conditional logic
        if (rCase.Related_Camp__c != null) {
            campIds.add(rCase.Related_Camp__c);
        }
    }

    // 4. Use a MAP to efficiently store query results
    Map<Id, Camp__c> campsMap = new Map<Id, Camp__c>([
        SELECT Id, Name FROM Camp__c WHERE Id IN :campIds
    ]);

    Map<Id, String> results = new Map<Id, String>();
    for(Id campId : campsMap.keySet()){
        results.put(campId, campsMap.get(campId).Name);
    }
    return results;
}
```

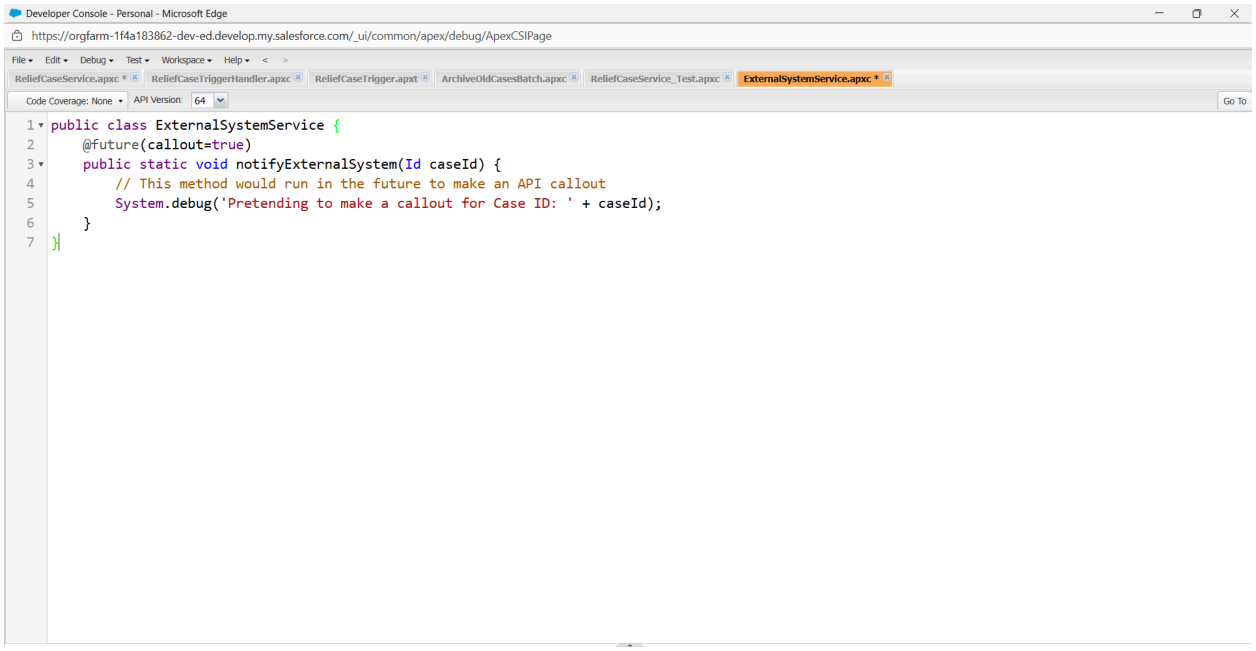
🔄 Asynchronous Apex Processing

To handle long-running operations and large data volumes without impacting the user experience or hitting governor limits, I implemented several types of asynchronous Apex.

- **@future Methods:** Used for simple, fire-and-forget operations, especially for callouts to external systems.

Apex

```
public class ExternalSystemService {
    @future(callout=true)
    public static void notifyExternalSystem(Id caseId) {
        // Pretend to make an API callout to an external logistics system
        // Http http = new Http();
        // ... (build request and send) ...
    }
}
```


Developer Console - Personal - Microsoft Edge
https://orgfarm-1f4a183862-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File
Edit
Debug
Test
Workspace
Help
ReliefCaseService.apxc *
ReliefCaseTriggerHandler.apxc
ReliefCaseTrigger.apxt
ArchiveOldCasesBatch.apxc
ReliefCaseService_Test.apxc
ExternalSystemService.apxc *
Code Coverage: None
API Version: 64
Go To

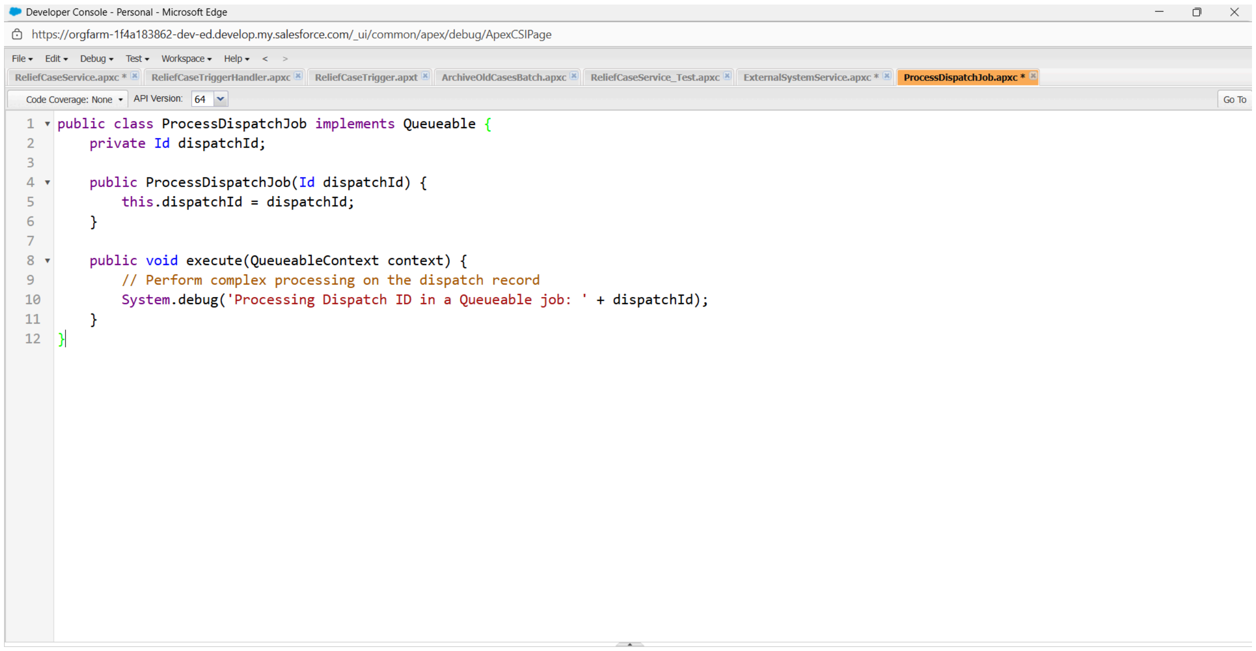
```
public class ExternalSystemService {
    @future(callout=true)
    public static void notifyExternalSystem(Id caseId) {
        // This method would run in the future to make an API callout
        System.debug('Pretending to make a callout for Case ID: ' + caseId);
    }
}
```

- **Queueable Apex:** Used for more complex async jobs that require chaining or access to more complex data types than future methods.

Apex

```
public class ProcessDispatchJob implements Queueable {
    private Id dispatchId;

    public ProcessDispatchJob(Id dispatchId) {
        this.dispatchId = dispatchId;
    }

    public void execute(QueueableContext context) {
        // Perform complex processing on the dispatch record
        // Optionally, chain to another job:
```

```
        // System.enqueueJob(new AnotherJob());
    }
}
```

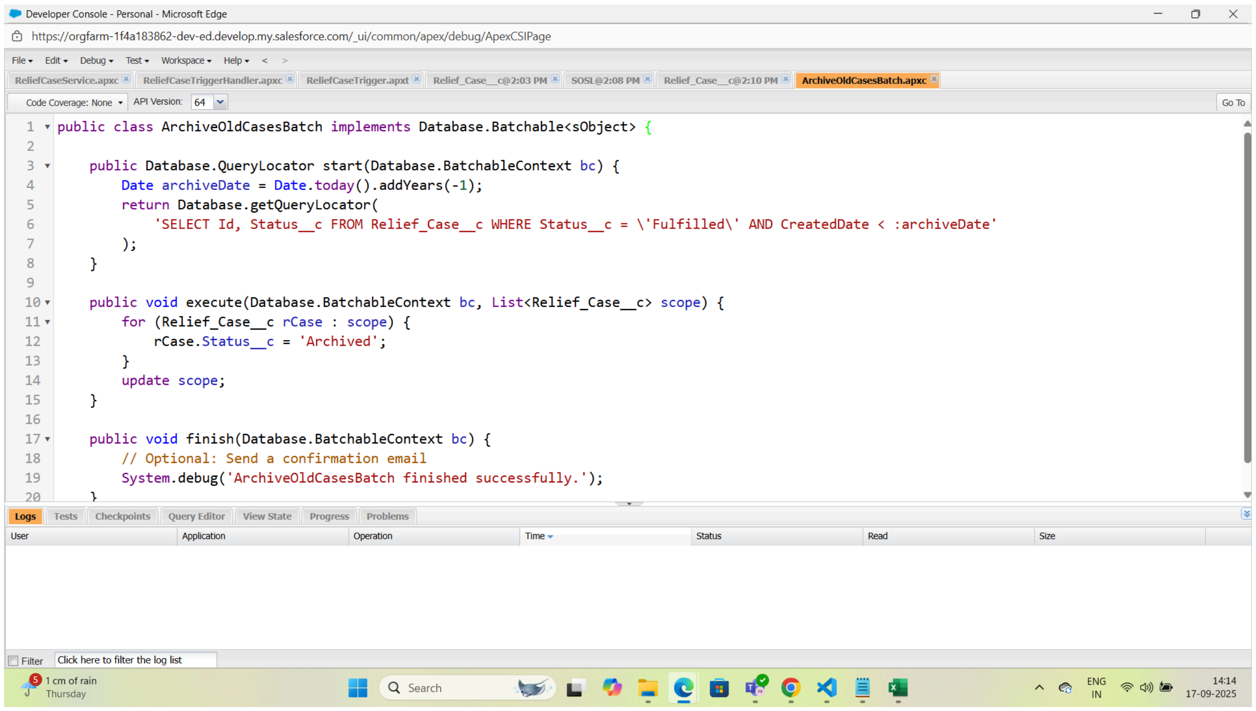

Developer Console - Personal - Microsoft Edge
https://orgfarm-1f4a183862-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
ReliefCaseService.apxc * ReliefCaseTriggerHandler.apxc ReliefCaseTrigger.apxt ArchiveOldCasesBatch.apxc ReliefCaseService_Test.apxc ExternalSystemService.apxc * ProcessDispatchJob.apxc *
Code Coverage: None API Version: 64
Go To

```
public class ProcessDispatchJob implements Queueable {
    private Id dispatchId;

    public ProcessDispatchJob(Id dispatchId) {
        this.dispatchId = dispatchId;
    }

    public void execute(QueueableContext context) {
        // Perform complex processing on the dispatch record
        System.debug('Processing Dispatch ID in a Queueable job: ' + dispatchId);
    }
}
```

- **Batch Apex:** The primary tool for processing thousands or millions of records.
    - **Implementation:** A batch class was created to run a nightly cleanup job, archiving old Relief_Case__c records.

Apex

```
public class ArchiveOldCasesBatch implements Database.Batchable<sObject> {

    public Database.QueryLocator start(Database.BatchableContext bc) {
        Date archiveDate = Date.today().addYears(-1);
        return Database.getQueryLocator(
            'SELECT Id, Status__c FROM Relief_Case__c WHERE Status__c = \'Fulfilled\' AND
CreatedDate < :archiveDate'
        );
    }

    public void execute(Database.BatchableContext bc, List<Relief_Case__c> scope) {
        for (Relief_Case__c rCase : scope) {
            rCase.Status__c = 'Archived';
        }
        update scope;
    }
    public void finish(Database.BatchableContext bc) {
        // Optional: Send a confirmation email
    }
}
```

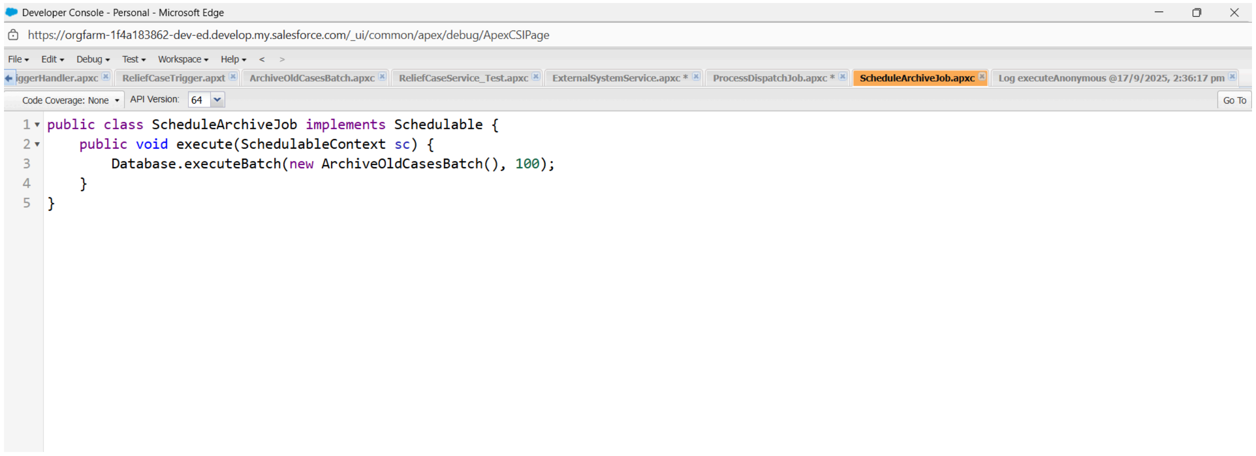

Developer Console - Personal - Microsoft Edge
https://orgfarm-1f4a183862-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
ReliefCaseService.apxc
ReliefCaseTriggerHandler.apxc
ReliefCaseTrigger.apxt
Relief_Case__c@2:03 PM
SOSL@2:08 PM
Relief_Case__c@2:10 PM
ArchiveOldCasesBatch.apxc
Code Coverage: None
API Version: 64
Go To
Logs Tests Checkpoints Query Editor View State Progress Problems
User Application Operation Time Status Read Size
Filter
Click here to filter the log list
1 cm of rain
Thursday
Search
ENG
IN
14:14
17-09-2025

```
public class ArchiveOldCasesBatch implements Database.Batchable<sObject> {

    public Database.QueryLocator start(Database.BatchableContext bc) {
        Date archiveDate = Date.today().addYears(-1);
        return Database.getQueryLocator(
            'SELECT Id, Status__c FROM Relief_Case__c WHERE Status__c = \'Fulfilled\' AND CreatedDate < :archiveDate'
        );
    }

    public void execute(Database.BatchableContext bc, List<Relief_Case__c> scope) {
        for (Relief_Case__c rCase : scope) {
            rCase.Status__c = 'Archived';
        }
        update scope;
    }

    public void finish(Database.BatchableContext bc) {
        // Optional: Send a confirmation email
        System.debug('ArchiveOldCasesBatch finished successfully.');
    }
```

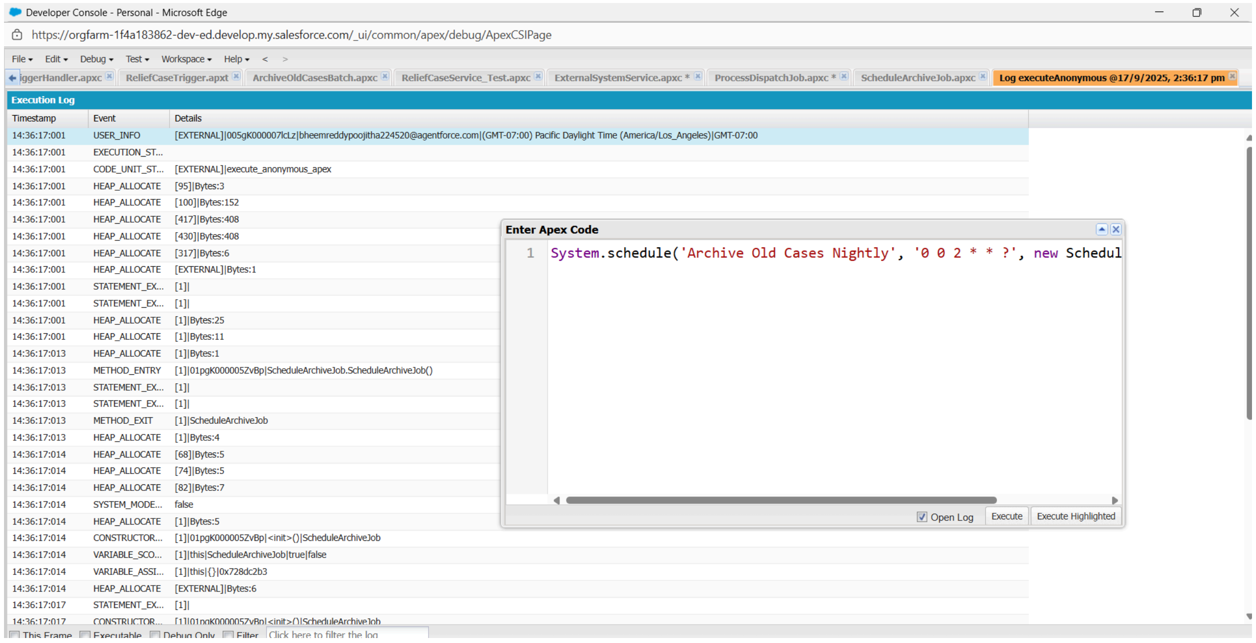
- **Scheduled Apex:** Used to run Apex code at a specific time.
  - **Implementation:** A schedulable class was created to invoke the ArchiveOldCasesBatch job every night.

Apex

```
public class ScheduleArchiveJob implements Schedulable {
    public void execute(SchedulableContext sc) {
        Database.executeBatch(new ArchiveOldCasesBatch(), 100);
    }
}
// To schedule this to run every night at 2 AM:
// System.schedule('Archive Old Cases Nightly', '0 0 2 * * ?', new ScheduleArchiveJob());
```

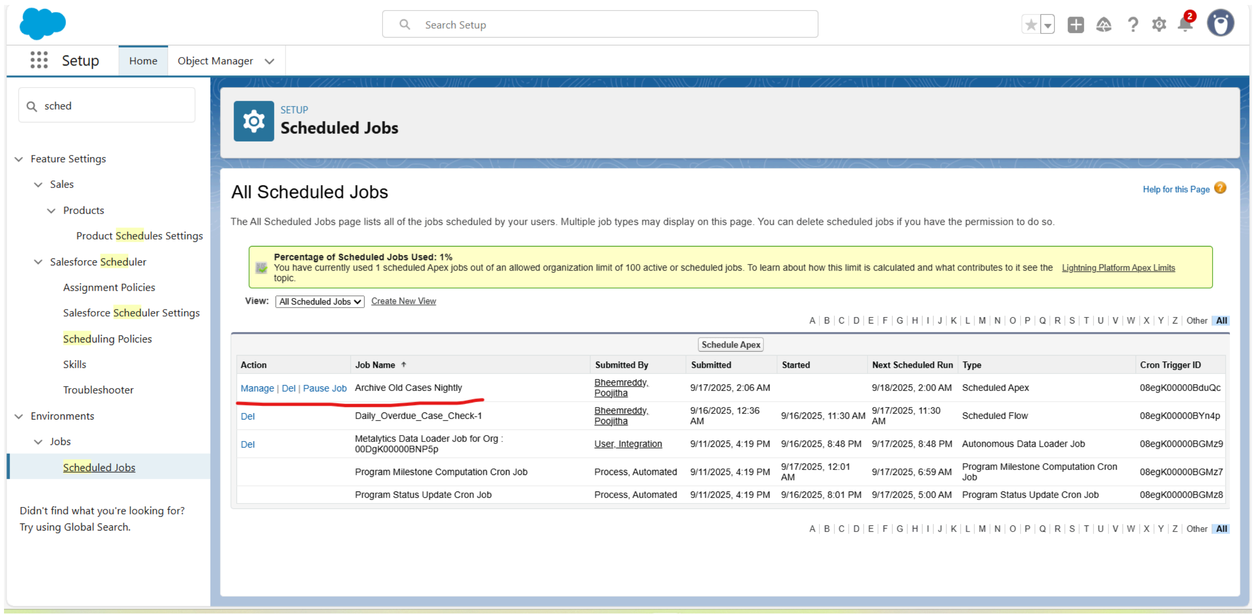

Developer Console - Personal - Microsoft Edge
https://orgfarm-1f4a183862-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
iggerHandler.apxc ReliefCaseTrigger.apxt ArchiveOldCasesBatch.apxc ReliefCaseService_Test.apxc ExternalSystemService.apxc * ProcessDispatchJob.apxc * ScheduleArchiveJob.apxc Log executeAnonymous @17/9/2025, 2:36:17 pm
Code Coverage: None API Version: 64 Go To

```
public class ScheduleArchiveJob implements Schedulable {
    public void execute(SchedulableContext sc) {
        Database.executeBatch(new ArchiveOldCasesBatch(), 100);
    }
}
```


Developer Console - Personal - Microsoft Edge
https://orgfarm-1f4a183862-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
iggerHandler.apxc ReliefCaseTrigger.apxt ArchiveOldCasesBatch.apxc ReliefCaseService_Test.apxc ExternalSystemService.apxc * ProcessDispatchJob.apxc * ScheduleArchiveJob.apxc Log executeAnonymous @17/9/2025, 2:36:17 pm
Execution Log
Enter Apex Code
Open Log
Execute
Execute Highlighted
This Frame
Executable
Debug Only
Filter
Click here to filter the log

```
Timestamp      Event            Details
14:36:17:001   USER_INFO        [EXTERNAL]|005gK000007lcLz|bheemreddypoojitha224520@agentforce.com|(GMT-07:00) Pacific Daylight Time (America/Los_Angeles)|GMT-07:00
14:36:17:001   EXECUTION_ST...
14:36:17:001   CODE_UNIT_ST...  [EXTERNAL]|execute_anonymous_apex
14:36:17:001   HEAP_ALLOCATE    [95]|Bytes:3
14:36:17:001   HEAP_ALLOCATE    [100]|Bytes:152
14:36:17:001   HEAP_ALLOCATE    [417]|Bytes:408
14:36:17:001   HEAP_ALLOCATE    [430]|Bytes:408
14:36:17:001   HEAP_ALLOCATE    [317]|Bytes:6
14:36:17:001   HEAP_ALLOCATE    [EXTERNAL]|Bytes:1
14:36:17:001   STATEMENT_EX...  [1]|
14:36:17:001   STATEMENT_EX...  [1]|
14:36:17:001   HEAP_ALLOCATE    [1]|Bytes:25
14:36:17:001   HEAP_ALLOCATE    [1]|Bytes:11
14:36:17:013   HEAP_ALLOCATE    [1]|Bytes:1
14:36:17:013   METHOD_ENTRY     [1]|01pgK000005ZvBp|ScheduleArchiveJob.ScheduleArchiveJob()
14:36:17:013   STATEMENT_EX...  [1]|
14:36:17:013   STATEMENT_EX...  [1]|
14:36:17:013   METHOD_EXIT      [1]|ScheduleArchiveJob
14:36:17:013   HEAP_ALLOCATE    [1]|Bytes:4
14:36:17:014   HEAP_ALLOCATE    [68]|Bytes:5
14:36:17:014   HEAP_ALLOCATE    [74]|Bytes:5
14:36:17:014   HEAP_ALLOCATE    [82]|Bytes:7
14:36:17:014   SYSTEM_MODE...   false
14:36:17:014   HEAP_ALLOCATE    [1]|Bytes:5
14:36:17:014   CONSTRUCTOR...   [1]|01pgK000005ZvBp|<init>()|ScheduleArchiveJob
14:36:17:014   VARIABLE_SCO...  [1]|this|ScheduleArchiveJob|true|false
14:36:17:014   VARIABLE_ASSI... [1]|this|{}|0x728dc2b3
14:36:17:014   HEAP_ALLOCATE    [EXTERNAL]|Bytes:6
14:36:17:017   STATEMENT_EX...  [1]|
14:36:17:017   CONSTRUCTOR...   [1]|01pgK000005ZvBp|<init>()|ScheduleArchiveJob
```

```
System.schedule('Archive Old Cases Nightly', '0 0 2 * * ?', new Schedul
```



All Scheduled Jobs

The All Scheduled Jobs page lists all of the jobs scheduled by your users. Multiple job types may display on this page. You can delete scheduled jobs if you have the permission to do so.

Percentage of Scheduled Jobs Used: 1%
You have currently used 1 scheduled Apex jobs out of an allowed organization limit of 100 active or scheduled jobs. To learn about how this limit is calculated and what contributes to it see the Lightning Platform Apex Limits topic.

Action	Job Name	Submitted By	Submitted	Started	Next Scheduled Run	Type	Cron Trigger ID
Manage | Del | Pause Job	Archive Old Cases Nightly	Bheemreddy, Poojitha	9/17/2025, 2:06 AM		9/18/2025, 2:00 AM	Scheduled Apex	08egK00000BduQc
Del	Daily_Overdue_Case_Check-1	Bheemreddy, Poojitha	9/16/2025, 12:36 AM	9/16/2025, 11:30 AM	9/17/2025, 11:30 AM	Scheduled Flow	08egK00000BYn4p
Del	Metalytics Data Loader Job for Org : 00DgK00000BNP5p	User, Integration	9/11/2025, 4:19 PM	9/16/2025, 8:48 PM	9/17/2025, 8:48 PM	Autonomous Data Loader Job	08egK00000BGMz9
	Program Milestone Computation Cron Job	Process, Automated	9/11/2025, 4:19 PM	9/17/2025, 12:01 AM	9/17/2025, 6:59 AM	Program Milestone Computation Cron Job	08egK00000BGMz7
	Program Status Update Cron Job	Process, Automated	9/11/2025, 4:19 PM	9/16/2025, 8:01 PM	9/17/2025, 5:00 AM	Program Status Update Cron Job	08egK00000BGMz8

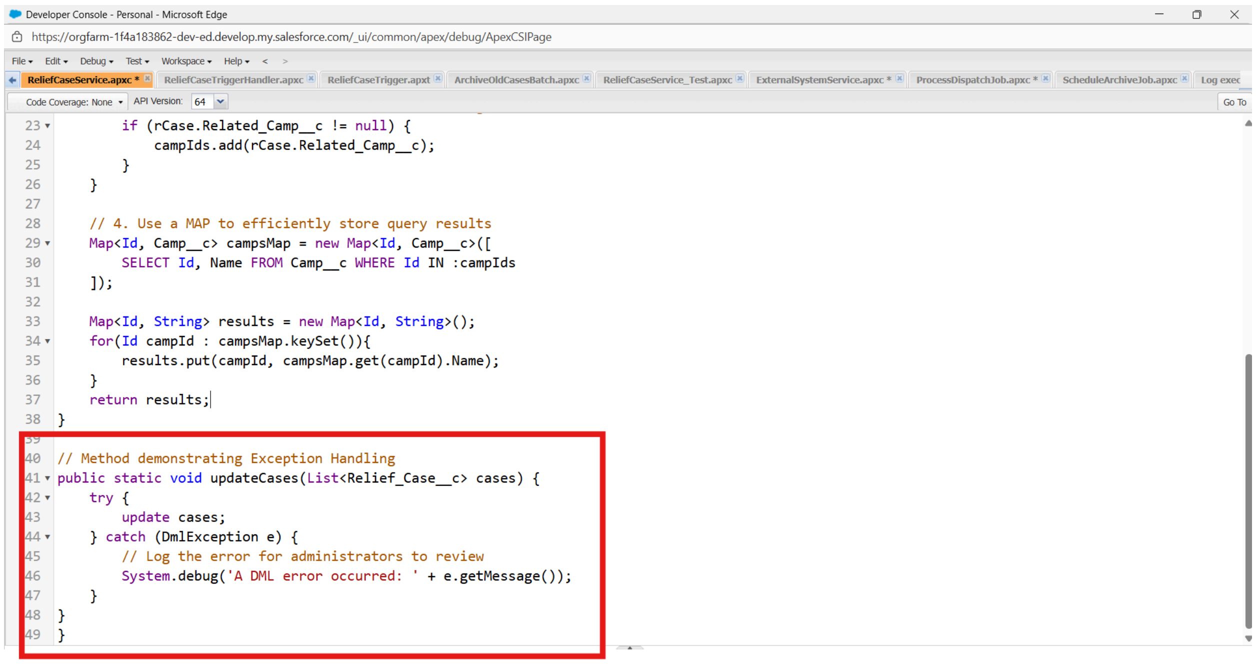
🛡️ Exception Handling

To ensure the application is robust and can gracefully handle unexpected errors (like a failed database operation), I implemented try/catch blocks in critical business logic.

- **Code Example:**

Apex

```
public static void updateCases(List<Relief_Case__c> cases) {
    try {
        update cases;
    } catch (DmlException e) {
        // Log the error for administrators to review
        System.debug('A DML error occurred: ' + e.getMessage());
        // Optionally, create a custom Log__c record
    }
}
```


Developer Console - Personal - Microsoft Edge
https://orgfarm-1f4a183862-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File
Edit
Debug
Test
Workspace
Help
ReliefCaseService.apxc *
ReliefCaseTriggerHandler.apxc
ReliefCaseTrigger.apxt
ArchiveOldCasesBatch.apxc
ReliefCaseService_Test.apxc
ExternalSystemService.apxc *
ProcessDispatchJob.apxc *
ScheduleArchiveJob.apxc
Log exec
Code Coverage: None
API Version: 64
Go To

```
        if (rCase.Related_Camp__c != null) {
            campIds.add(rCase.Related_Camp__c);
        }
    }

    // 4. Use a MAP to efficiently store query results
    Map<Id, Camp__c> campsMap = new Map<Id, Camp__c>([
        SELECT Id, Name FROM Camp__c WHERE Id IN :campIds
    ]);

    Map<Id, String> results = new Map<Id, String>();
    for(Id campId : campsMap.keySet()){
        results.put(campId, campsMap.get(campId).Name);
    }
    return results;
}

// Method demonstrating Exception Handling
public static void updateCases(List<Relief_Case__c> cases) {
    try {
        update cases;
    } catch (DmlException e) {
        // Log the error for administrators to review
        System.debug('A DML error occurred: ' + e.getMessage());
    }
}
}
```

🧪 Test Classes

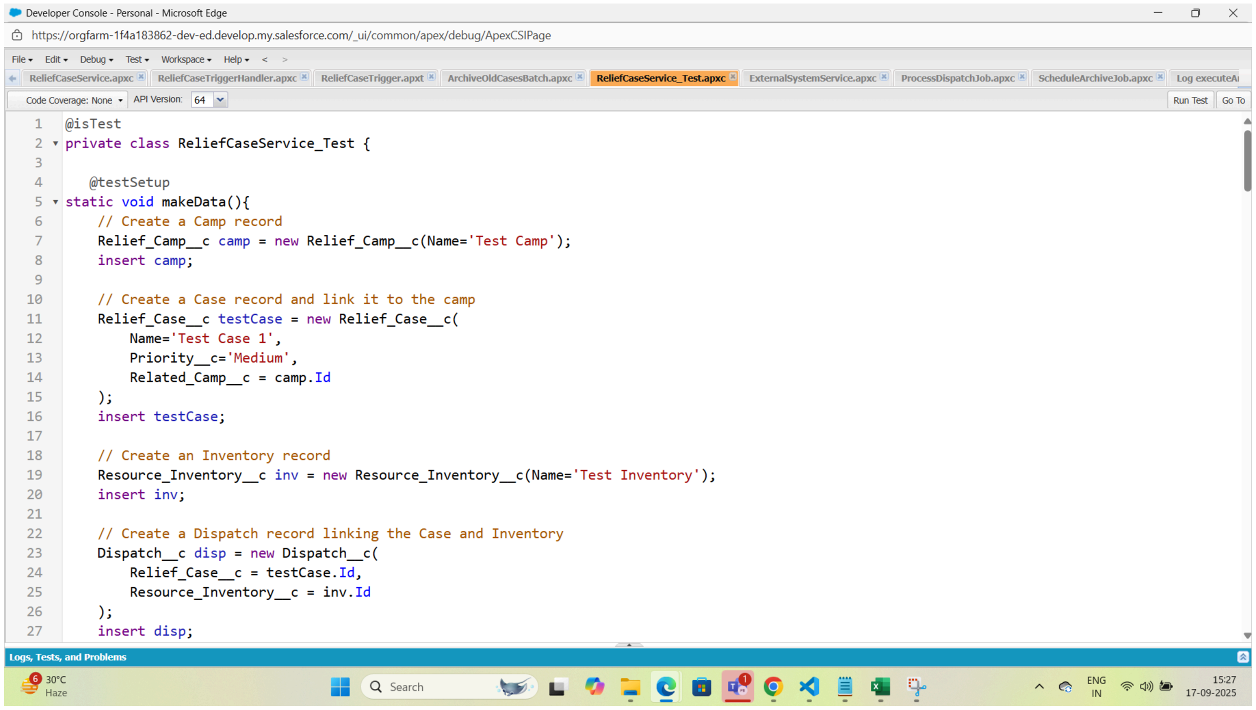
To guarantee code quality, reliability, and meet Salesforce's 75% code coverage requirement for deployment, I created dedicated test classes for all Apex logic.

- **Implementation:** Test classes were written using the @isTest annotation. Data is created in test setup methods (@testSetup), and business logic is invoked and verified using System.assert() statements.

- **Code Example (Testing the ReliefCaseService):**

Apex

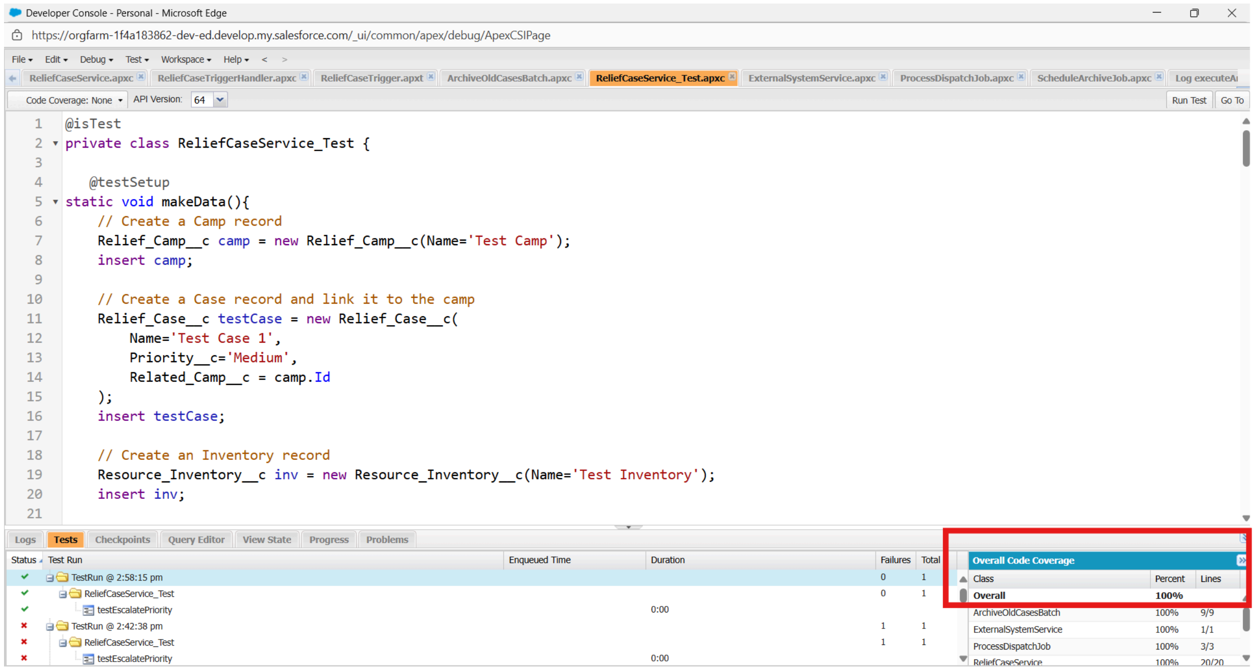
```
@isTest
private class ReliefCaseService_Test {

    @testSetup
    static void makeData() {
        // Create a Camp record
        Camp__c camp = new Camp__c(Name = 'Test Camp');
        insert camp;

        // Create a Case record and link it to the camp
        Relief_Case__c testCase = new Relief_Case__c(
            Name = 'Test Case 1',
            Priority__c = 'Medium',
            People_Affected__c = 10,
            Related_Camp__c = camp.Id
        );
        insert testCase;

        // Create an Inventory record
        Resource_Inventory__c inv = new Resource_Inventory__c(Name = 'Test Inventory');
        insert inv;

        // Create a Dispatch record linking the Case and Inventory
        Dispatch__c disp = new Dispatch__c(
            Relief_Case__c = testCase.Id,
            Resource_Inventory__c = inv.Id
        );
        insert disp;
    }

    //-- Tests for ReliefCaseService Methods --//
    @isTest
    static void testEscalatePriority() {
        Relief_Case__c testCase = [SELECT Id, Priority__c FROM Relief_Case__c LIMIT 1];

        Test.startTest();
        ReliefCaseService.escalatePriority(new List<Relief_Case__c>{ testCase });
        Test.stopTest();
        Relief_Case__c updatedCase = [SELECT Priority__c FROM Relief_Case__c WHERE Id =
:testCase.Id];
        System.assertEquals('High', updatedCase.Priority__c, 'The priority should have been escalated
to High.');
    }

    @isTest
    static void testEscalatePriority_AlreadyCritical() {
        Relief_Case__c criticalCase = new Relief_Case__c(Name = 'Critical Test Case', Priority__c =
'Critical');
        insert criticalCase;

        Test.startTest();
        ReliefCaseService.escalatePriority(new List<Relief_Case__c>{ criticalCase });
        Test.stopTest();

        Relief_Case__c resultCase = [SELECT Id, Priority__c FROM Relief_Case__c WHERE Id =
:criticalCase.Id];
        System.assertEquals('Critical', resultCase.Priority__c, 'The priority should remain Critical.');
    }

    @isTest
    static void testGetCampNamesForCases() {
        Relief_Case__c testCase = [SELECT Id, Related_Camp__c FROM Relief_Case__c LIMIT 1];

        Test.startTest();
        Map<Id, String> campNamesMap = new ReliefCaseService().getCampNamesForCases(new
List<Relief_Case__c>{ testCase });
        Test.stopTest();

        System.assert(!campNamesMap.isEmpty(), 'The map should not be empty.');
        System.assertEquals(1, campNamesMap.size(), 'The map should contain one entry.');
    }

    @isTest
    static void testGetCampNames_NullCamp() {
        Relief_Case__c caseWithNullCamp = new Relief_Case__c(Name = 'Test Case with Null
Camp');
        insert caseWithNullCamp;

        Test.startTest();
        Map<Id, String> campNamesMap = new ReliefCaseService().getCampNamesForCases(new
List<Relief_Case__c>{ caseWithNullCamp });
        Test.stopTest();

        System.assert(campNamesMap.isEmpty(), 'Map should be empty when case has no camp.');
    }

    @isTest
    static void testUpdateCases() {
        Relief_Case__c testCase = [SELECT Id, Priority__c FROM Relief_Case__c LIMIT 1];
        testCase.Priority__c = 'Low';

        Test.startTest();
        ReliefCaseService.updateCases(new List<Relief_Case__c>{ testCase });
        Test.stopTest();

        Relief_Case__c updatedCase = [SELECT Id, Priority__c FROM Relief_Case__c WHERE Id =
:testCase.Id];
        System.assertEquals('Low', updatedCase.Priority__c, 'The priority should have been updated
to Low.');
    }

    // Verifies the updateCases method gracefully handles DML exceptions.
    @isTest
    static void testUpdateCases_Exception_Foolproof() {
        Relief_Case__c caseWithoutId = new Relief_Case__c(Name='Bad Case');

        Test.startTest();
        ReliefCaseService.updateCases(new List<Relief_Case__c>{ caseWithoutId });
        Test.stopTest();

        System.assert(true, 'The method should handle the DML exception for a record without an ID.');
    }

    //-- Test for ReliefCaseTriggerHandler --//
    @isTest
    static void testTriggerHandler_EscalateOnUpdate() {
        Relief_Case__c testCase = [SELECT Id, People_Affected__c, Priority__c FROM
Relief_Case__c LIMIT 1];
        testCase.People_Affected__c = 20;

        Test.startTest();
        update testCase;
        Test.stopTest();

        Relief_Case__c updatedCase = [SELECT Id, Priority__c FROM Relief_Case__c WHERE Id =
:testCase.Id];
        System.assertEquals('High', updatedCase.Priority__c, 'Priority should be escalated when
people affected increases.');
    }
    //-- Tests for Asynchronous Classes --//
    @isTest
    static void testFutureMethod() {
        Relief_Case__c testCase = [SELECT Id FROM Relief_Case__c LIMIT 1];

        Test.startTest();
        ExternalSystemService.notifyExternalSystem(testCase.Id);
        Test.stopTest();

        System.assert(true, 'Future method was called successfully.');
    }

    @isTest
    static void testQueueableJob() {
        Dispatch__c disp = [SELECT Id FROM Dispatch__c LIMIT 1];

        Test.startTest();
        System.enqueueJob(new ProcessDispatchJob(disp.Id));
        Test.stopTest();

        System.assert(true, 'Queueable job was enqueued successfully.');
    }

    @isTest
    static void testBatchJob() {
        Relief_Case__c oldCase = new Relief_Case__c(Name = 'Old Fulfilled Case', Status__c =
'Fulfilled');
        insert oldCase;
        Test.setCreatedDate(oldCase.Id, Date.today().addYears(-2));

        Test.startTest();
        Database.executeBatch(new ArchiveOldCasesBatch());
        Test.stopTest();

        Relief_Case__c updatedCase = [SELECT Status__c FROM Relief_Case__c WHERE Id =
:oldCase.Id];
        System.assertEquals('Archived', updatedCase.Status__c, 'Batch job should have archived the
old case.');
    }

    @isTest
    static void testScheduledBatchJob() {
        Test.startTest();
        System.schedule('Test Archive Job', '0 0 2 * * ?', new ScheduleArchiveJob());
        Test.stopTest();
        System.assert(true, 'The scheduled job should run without errors.');
    }
}
```



Developer Console - Personal - Microsoft Edge

https://orgfarm-1f4a183862-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

```
@isTest
private class ReliefCaseService_Test {

   @testSetup
static void makeData(){
    // Create a Camp record
    Relief_Camp__c camp = new Relief_Camp__c(Name='Test Camp');
    insert camp;

    // Create a Case record and link it to the camp
    Relief_Case__c testCase = new Relief_Case__c(
        Name='Test Case 1',
        Priority__c='Medium',
        Related_Camp__c = camp.Id
    );
    insert testCase;

    // Create an Inventory record
    Resource_Inventory__c inv = new Resource_Inventory__c(Name='Test Inventory');
    insert inv;

    // Create a Dispatch record linking the Case and Inventory
    Dispatch__c disp = new Dispatch__c(
        Relief_Case__c = testCase.Id,
        Resource_Inventory__c = inv.Id
    );
    insert disp;
```



Developer Console - Personal - Microsoft Edge

https://orgfarm-1f4a183862-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

```
@isTest
private class ReliefCaseService_Test {

   @testSetup
static void makeData(){
    // Create a Camp record
    Relief_Camp__c camp = new Relief_Camp__c(Name='Test Camp');
    insert camp;

    // Create a Case record and link it to the camp
    Relief_Case__c testCase = new Relief_Case__c(
        Name='Test Case 1',
        Priority__c='Medium',
        Related_Camp__c = camp.Id
    );
    insert testCase;

    // Create an Inventory record
    Resource_Inventory__c inv = new Resource_Inventory__c(Name='Test Inventory');
    insert inv;
```

Status	Test Run	Enqueued Time	Duration	Failures	Total
✓	TestRun @ 2:58:15 pm			0	1
✓	ReliefCaseService_Test			0	1
✓	testEscalatePriority		0:00		
✗	TestRun @ 2:42:38 pm			1	1
✗	ReliefCaseService_Test			1	1
✗	testEscalatePriority		0:00		

Overall Code Coverage

Class	Percent	Lines
Overall	**100%**	
ArchiveOldCasesBatch	100%	9/9
ExternalSystemService	100%	1/1
ProcessDispatchJob	100%	3/3
ReliefCaseService	100%	20/20