

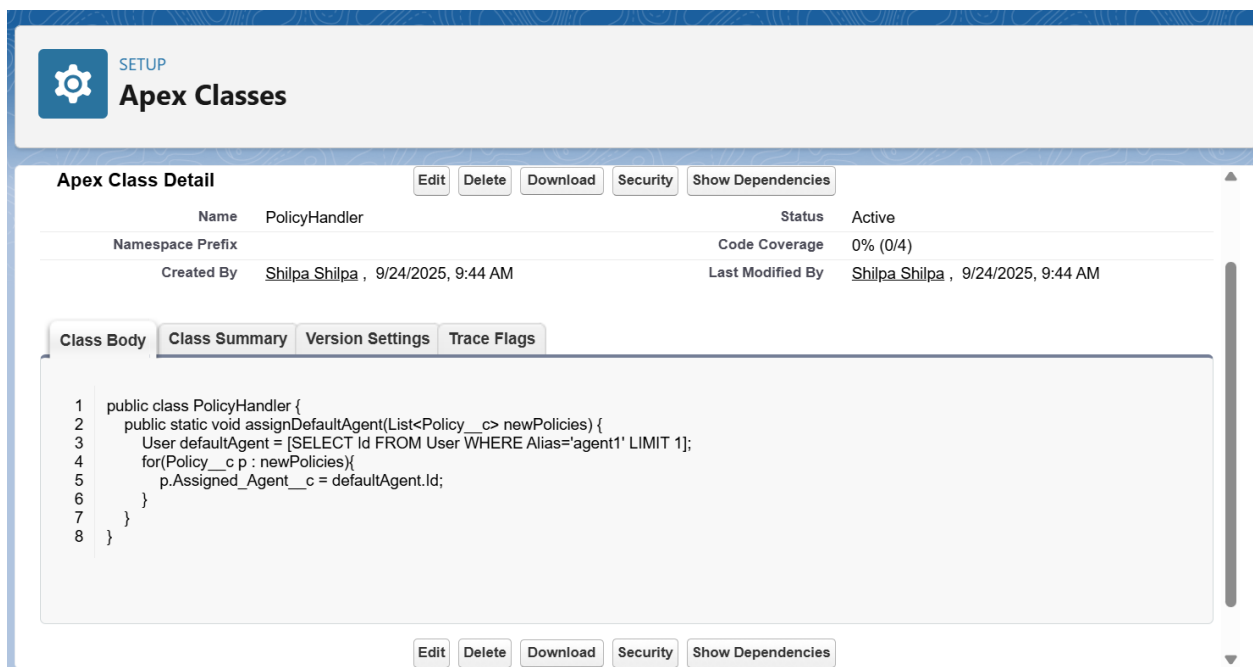
Phase 5: Apex Programming (Developer)

Objectives:

- Automate record processing
- Implement triggers for real-time operations
- Ensure data consistency and security

Apex Classes:

- Purpose: Automatically assigns a default agent when a new Policy is created.



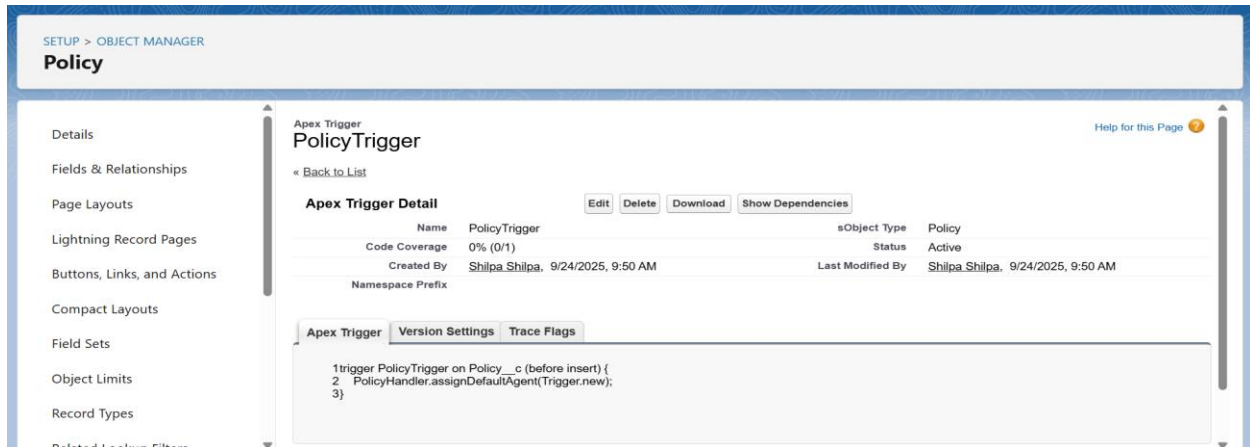
The screenshot displays the Salesforce 'Apex Classes' setup page. At the top, there's a 'SETUP' button and the title 'Apex Classes'. Below this, the 'Apex Class Detail' section shows the class name 'PolicyHandler' and its status 'Active'. A table provides details: 'Namespace Prefix' is empty, 'Code Coverage' is '0% (0/4)', 'Created By' is 'Shilpa Shilpa' on '9/24/2025, 9:44 AM', and 'Last Modified By' is also 'Shilpa Shilpa' on the same date and time. Below the table, there are tabs for 'Class Body', 'Class Summary', 'Version Settings', and 'Trace Flags'. The 'Class Body' tab is selected, showing the following Apex code:

```
1 public class PolicyHandler {
2     public static void assignDefaultAgent(List<Policy__c> newPolicies) {
3         User defaultAgent = [SELECT Id FROM User WHERE Alias='agent1' LIMIT 1];
4         for(Policy__c p : newPolicies){
5             p.Assigned_Agent__c = defaultAgent.Id;
6         }
7     }
8 }
```

At the bottom of the class body, there are buttons for 'Edit', 'Delete', 'Download', 'Security', and 'Show Dependencies'.

Apex Triggers:

- Trigger Type: before insert
- Function: Calls PolicyHandler class to assign default agents.

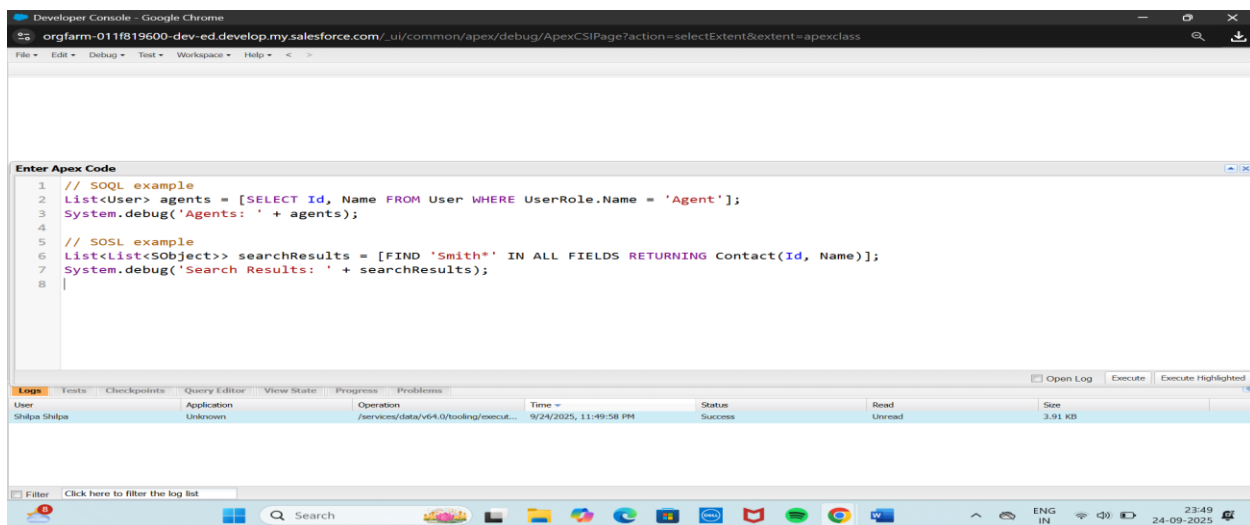


Trigger Design Patterns:

One Trigger per Object

Handler Classes

Bulk-Safe loops over Trigger.new



Collections (List, Set, Map):

- List: Ordered collection
- Set: Unique elements
- Map: Key-value pairs

The screenshot shows the Salesforce Developer Console with the following Apex code:

```

1 List<String> policyNames = new List<String>();
2 policyNames.add('Vehicle Policy');
3 policyNames.add('Health Policy');
4 System.debug('Policy Names List: ' + policyNames);
5
6 Set<String> uniqueAgents = new Set<String>();
7 uniqueAgents.add('Agent1');
8 uniqueAgents.add('Agent2');
9 uniqueAgents.add('Agent1'); // Duplicate, will be ignored
10 System.debug('Unique Agents Set: ' + uniqueAgents);
11
12 Map<Id, Policy__c> policyMap = new Map<Id, Policy__c>{
13     [SELECT Id, Name FROM Policy__c LIMIT 5]
14 };
15 System.debug('Policy Map: ' + policyMap);

```

The execution log shows the following details:

Timestamp	Event	Details
23:30:13:002	USER_DEBUG	[4][DEBUG]Policy Names List: (Vehicle Policy, Health Policy)
23:30:13:002	USER_DEBUG	[10][DEBUG]Unique Agents Set: (Agent1, Agent2)
23:30:13:020	USER_DEBUG	[15][DEBUG]Policy Map: {a01gl000000XnffQAT=Policy__c:[Id=a01gl000000XnffQAT, Name=P-00001], a01gl000000Y6pyQAD=Policy__c:[Id=a01gl000000Y6pyQAD, Name=P-00002]}

The bottom section shows the 'Execute Anonymous' window with the same code and a table of execution results:

Log	Tests	Checkpoints	Query Editor	View State	Progress	Problems
Shipsa Shipsa	Unknown	Unknown	/services/data/v64.0/tooling/execute...	9/24/2025, 11:26:52 PM	Success	3.2 KB
Shipsa Shipsa	Unknown	Unknown	/services/data/v64.0/tooling/execute...	9/24/2025, 11:24:56 PM	Success	2.84 KB

Control Statements:

- if-else, for loops, while loops

The screenshot shows the Salesforce Developer Console with the following Apex code:

```

1 List<Claim__c> claims = [SELECT Id, Claim_Amount__c, Status__c FROM Claim__c LIMIT 5];
2
3 for(Claim__c c : claims){
4     if(c.Claim_Amount__c > 10000){
5         c.Status__c = 'Requires Manager Approval';
6     } else {
7         c.Status__c = 'Approved';
8     }
9     System.debug('Claim: ' + c.Id + ', Amount: ' + c.Claim_Amount__c + ', Status: ' + c.Status__c);
10 }
11

```

The execution log shows the following details:

Log	Tests	Checkpoints	Query Editor	View State	Progress	Problems
23:52:15:010	HEAP_ALLOCATE	[1]Bytes:0				
23:52:15:010	HEAP_ALLOCATE	[1]Bytes:4				
Shipsa Shipsa	Unknown	Unknown	/services/data/v64.0/tooling/execute...	9/24/2025, 11:52:15 PM	Success	2.93 KB
Shipsa Shipsa	Unknown	Unknown	/services/data/v64.0/tooling/execute...	9/24/2025, 11:51:29 PM	Success	2.93 KB
Shipsa Shipsa	Unknown	Unknown	/services/data/v64.0/tooling/execute...	9/24/2025, 11:49:58 PM	Success	3.91 KB

Batch Apex:

```

global class BatchPolicyUpdate implements Database.Batchable<sObject> {
    global Database.QueryLocator start(Database.BatchableContext BC){
        return Database.getQueryLocator('SELECT Id FROM Policy__c WHERE Status__c = NULL');
    }
    global void execute(Database.BatchableContext BC, List<Policy__c> scope){
        for(Policy__c p : scope){
            p.Status__c = 'Pending';
        }
        update scope;
    }
}

```

```

    global void finish(Database.BatchableContext BC){
}

```

The screenshot shows the Salesforce Developer Console with the Apex Editor displaying the following code:

```

1 BatchPolicyUpdate batchJob = new BatchPolicyUpdate();
2 Database.executeBatch(batchJob, 100);
3

```

The Logs tab is selected, showing a table of execution logs. The table has columns: User, Application, Operation, Time, Status, Read, and Size. The logs show the execution of the Batch Apex class.

User	Application	Operation	Time	Status	Read	Size
Shilpa Shilpa	Unknown	Batch Apex	9/24/2025, 11:55:10 PM	Success	Unread	3.08 KB
Shilpa Shilpa	Unknown	Batch Apex	9/24/2025, 11:55:10 PM	Success	Unread	3.48 KB
Shilpa Shilpa	Unknown	/services/data/v64.0/tooling/executeBatch	9/24/2025, 11:55:09 PM	Success	Unread	2.94 KB
Shilpa Shilpa	Browser	/setup/build/editApexClass.apexp	9/24/2025, 11:54:30 PM	Success	Unread	1.36 KB
Shilpa Shilpa	Browser	/setup/build/listApexClass.apexp	9/24/2025, 11:54:23 PM	Success	Unread	631 bytes
Shilpa Shilpa	Unknown	/services/data/v64.0/tooling/executeBatch	9/24/2025, 11:52:15 PM	Success	Unread	2.93 KB

Queueable Apex:

The screenshot shows the Salesforce Developer Console with the Apex Editor displaying the following code:

```

1 System.enqueueJob(new QueueablePolicyUpdate());
2

```

The Logs tab is selected, showing a table of execution logs. The table has columns: User, Application, Operation, Time, Status, Read, and Size. The logs show the execution of the Queueable Apex class.

User	Application	Operation	Time	Status	Read	Size
Shilpa Shilpa	Unknown	QueueableHandler	9/24/2025, 11:56:38 PM	Success	Unread	3.37 KB
Shilpa Shilpa	Unknown	/services/data/v64.0/tooling/executeBatch	9/24/2025, 11:56:36 PM	Success	Unread	2.72 KB
Shilpa Shilpa	Browser	/setup/build/editApexClass.apexp	9/24/2025, 11:56:18 PM	Success	Unread	1.16 KB
Shilpa Shilpa	Browser	/setup/build/listApexClass.apexp	9/24/2025, 11:56:11 PM	Success	Unread	631 bytes
Shilpa Shilpa	Unknown	Batch Apex	9/24/2025, 11:55:10 PM	Success	Unread	3.48 KB
Shilpa Shilpa	Unknown	Batch Apex	9/24/2025, 11:55:10 PM	Success	Unread	3.08 KB

Scheduled Apex:

The screenshot shows the Salesforce Developer Console with the following components:

- Developer Console - Google Chrome:** The top window showing the URL `orgfarm-011f819600-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage?action=selectExtent&extent=apexclass`.
- Execute Anonymous - Google Chrome:** The bottom window showing the code:

```
1 System.schedule('Daily Policy Update', '0 0 2 * * ?', new ScheduledPolicyUpdate());
2
```
- Log List:** A table showing the execution history of the scheduled job.

User	Application	Operation	Time	Status	Read	Size
Shilpa Shilpa	Unknown	/services/data/v64.0/tooling/executeAnonymous	9/24/2025, 11:58:24 PM	Success		2.86 KB
Shilpa Shilpa	Browser	/setup/build/editApexClass.apexp	9/24/2025, 11:58:11 PM	Success	Unread	1.16 KB
Shilpa Shilpa	Browser	/setup/build/editApexClass.apexp	9/24/2025, 11:58:03 PM	Success	Unread	631 bytes
Shilpa Shilpa	Unknown	QueueableHandler	9/24/2025, 11:56:38 PM	Success	Unread	3.37 KB
Shilpa Shilpa	Unknown	/services/data/v64.0/tooling/executeAnonymous	9/24/2025, 11:56:36 PM	Success		2.72 KB
Shilpa Shilpa	Browser	/setup/build/editApexClass.apexp	9/24/2025, 11:56:18 PM	Success	Unread	1.16 KB

Future Methods:

The screenshot shows the Salesforce Developer Console with the following components:

- Developer Console - Google Chrome:** The top window showing the URL `orgfarm-011f819600-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage?action=selectExtent&extent=apexclass`.
- Execute Anonymous - Google Chrome:** The bottom window showing the code:

```
1 // Get list of Policy Ids
2 List<Policy__c> policies = [SELECT Id FROM Policy__c LIMIT 5];
3 List<Id> policyIds = new List<Id>();
4
5 for(Policy__c p : policies){
6     policyIds.add(p.Id);
7 }
8
9 // Call the future method
10 PolicyEmailHandler.sendPolicyEmail(policyIds);
11
```
- Execution Log:** A table showing the execution history of the future method.

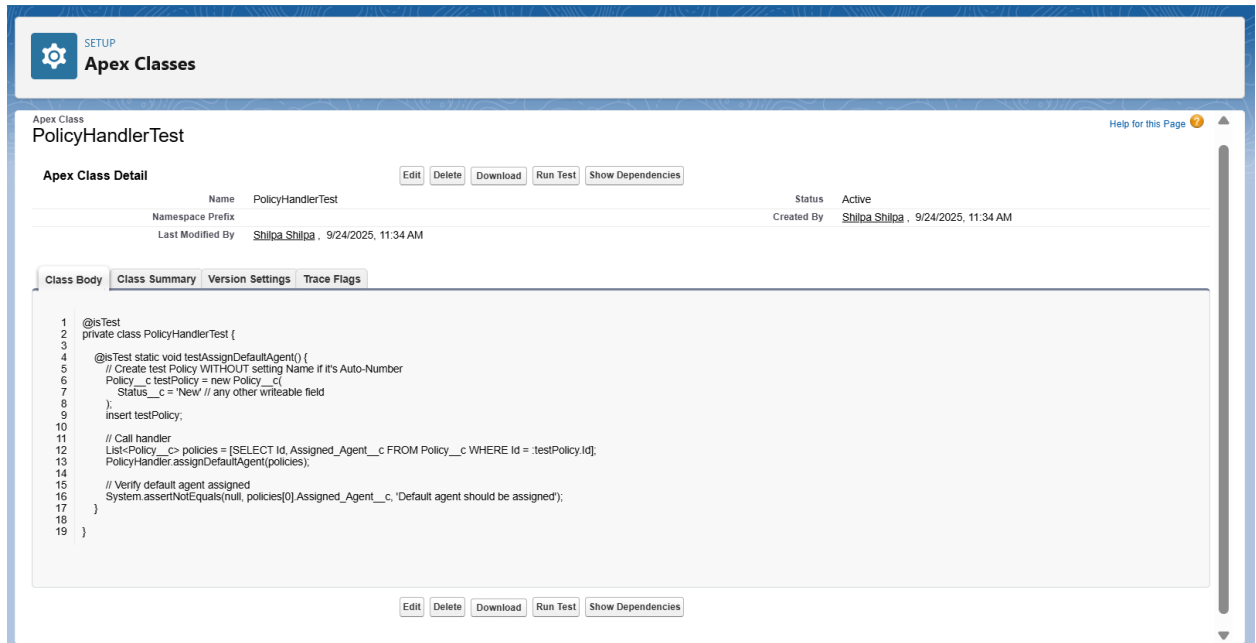
Timestamp	Event	Details
00:00:45:052	USER_DEBUG	[7]DEBUG Sending email for Policy: P-00001
00:00:45:052	USER_DEBUG	[7]DEBUG Sending email for Policy: P-00002

User	Application	Operation	Time	Status	Read	Size
Shilpa Shilpa	Unknown	/services/data/v64.0/tooling/executeAnonymous	9/25/2025, 12:00:45 AM	Success	Unread	4.28 KB
Shilpa Shilpa	Unknown	FutureHandler	9/25/2025, 12:00:45 AM	Success		4.6 KB
Shilpa Shilpa	Browser	/setup/build/editApexClass.apexp	9/24/2025, 11:59:29 PM	Success	Unread	1.16 KB
Shilpa Shilpa	Browser	/setup/build/editApexClass.apexp	9/24/2025, 11:59:22 PM	Success	Unread	626 bytes
Shilpa Shilpa	Unknown	/services/data/v64.0/tooling/executeAnonymous	9/24/2025, 11:58:24 PM	Success		2.86 KB
Shilpa Shilpa	Browser	/setup/build/editApexClass.apexp	9/24/2025, 11:58:11 PM	Success	Unread	1.16 KB

Exception Handling:

```
try {
    insert newPolicies;
} catch(DmlException e) {
    System.debug('Error inserting policies: ' + e.getMessage());
}
```

Test Classes:



The screenshot shows the Salesforce Apex Classes interface for a class named **PolicyHandlerTest**. The page includes a header with the Salesforce logo and a "SETUP" button. Below the header, the class name "PolicyHandlerTest" is displayed, along with a "Help for this Page" link. The "Apex Class Detail" section shows the class name, namespace prefix, status (Active), and creation/modification details. The "Class Body" tab is selected, displaying the following Apex code:

```
1  @isTest
2  private class PolicyHandlerTest {
3
4      @isTest static void testAssignDefaultAgent() {
5          // Create test Policy WITHOUT setting Name if it's Auto-Number
6          Policy__c testPolicy = new Policy__c(
7              Status__c = 'New' // any other writeable field
8          );
9          insert testPolicy;
10
11         // Call handler
12         List<Policy__c> policies = [SELECT Id, Assigned_Agent__c FROM Policy__c WHERE Id = :testPolicy.Id];
13         PolicyHandler.assignDefaultAgent(policies);
14
15         // Verify default agent assigned
16         System.assertNotEquals(null, policies[0].Assigned_Agent__c, 'Default agent should be assigned');
17     }
18 }
19 }
```

At the bottom of the class body, there are buttons for "Edit", "Delete", "Download", "Run Test", and "Show Dependencies".