Carbon Footprint Tracker

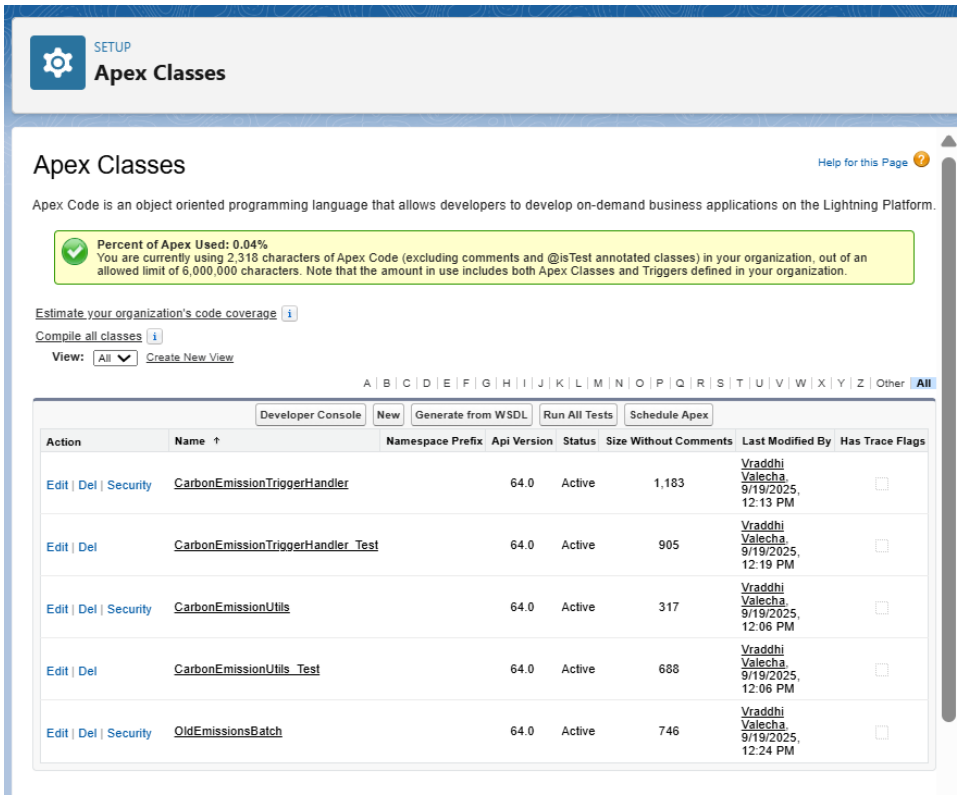
Phase 5: Apex Programming (Developer)

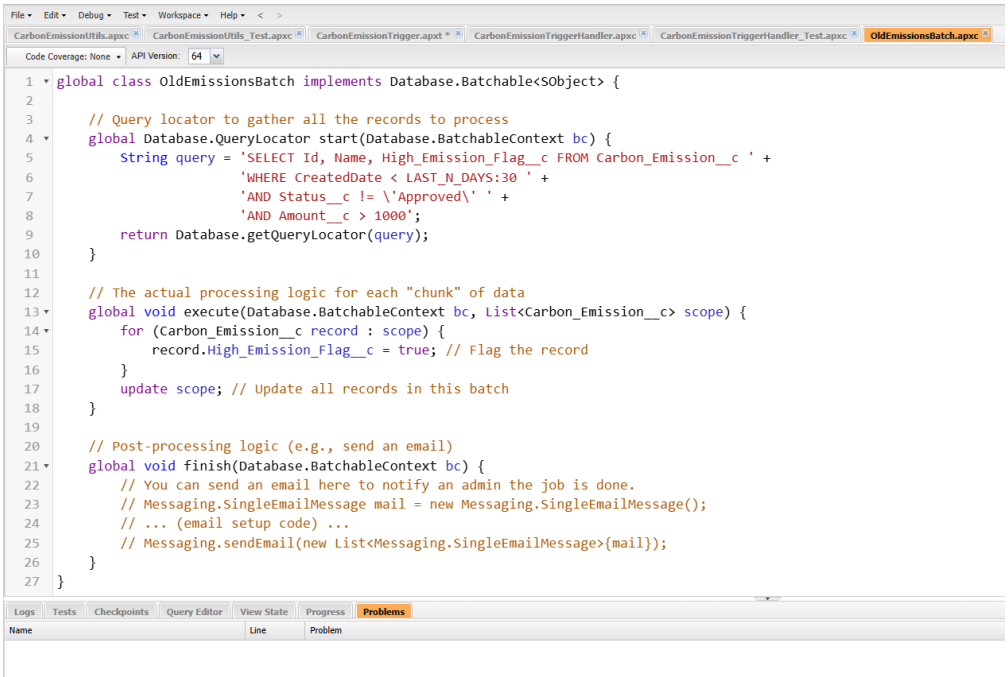
1. Classes & Objects

Developed:

- CarbonEmissionUtils: Utility class for unit conversions.
- CarbonEmissionTriggerHandler: Handler class for trigger logic.
- OldEmissionsBatch: Batch class for asynchronous processing.

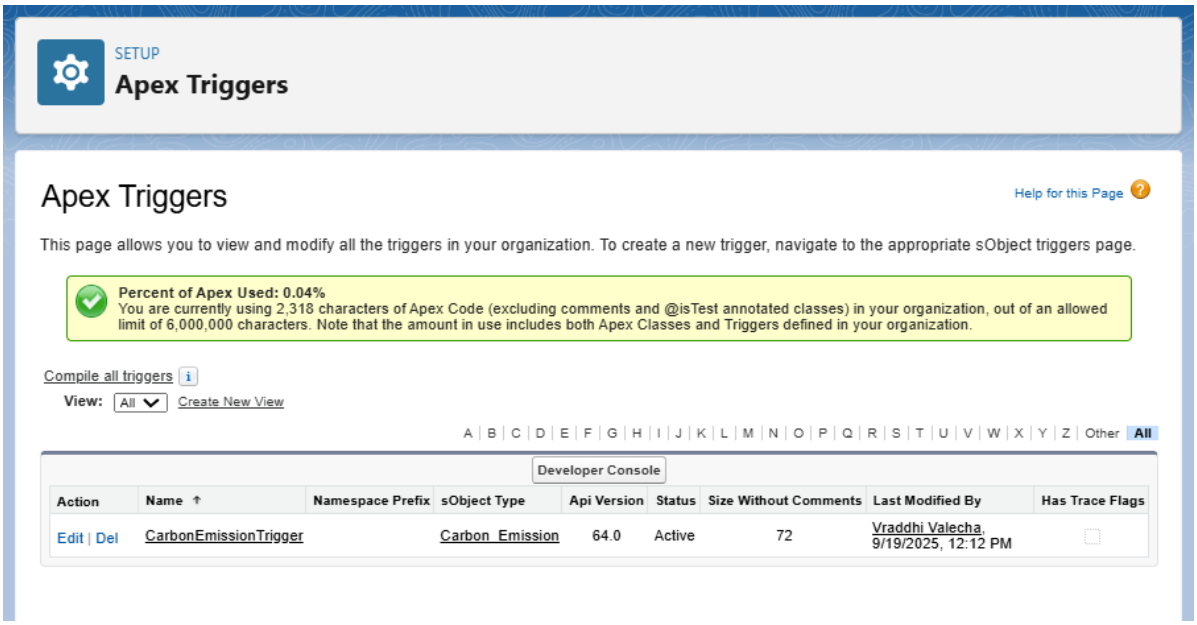
Verification: Setup → Apex Classes → All three classes present.

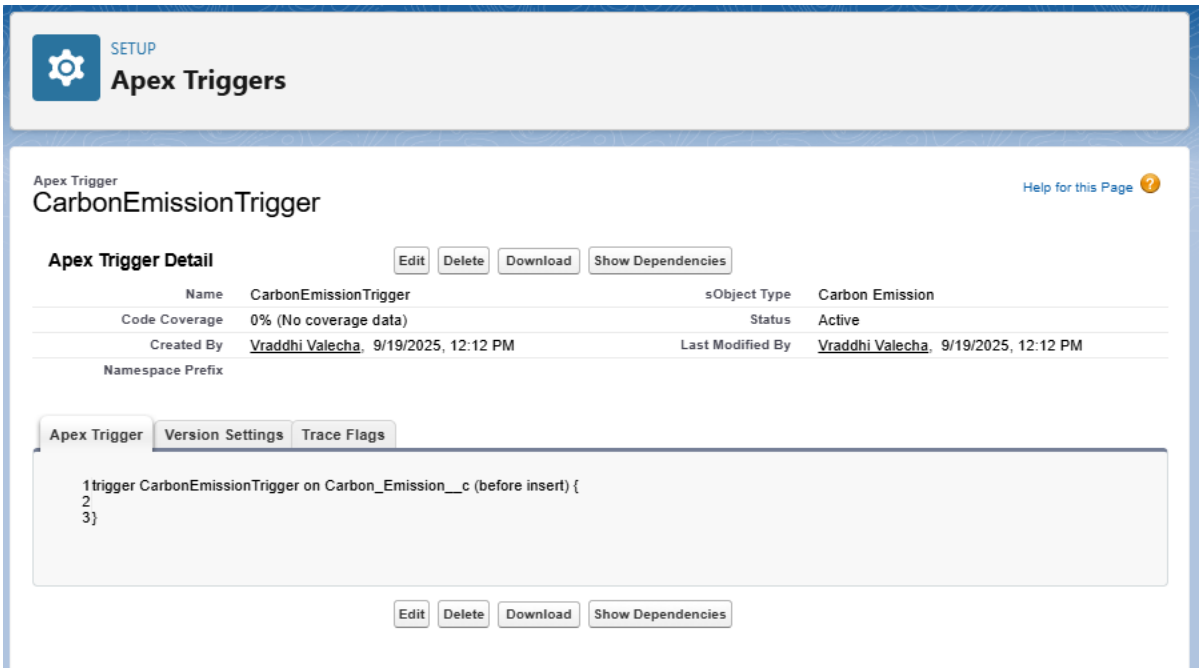


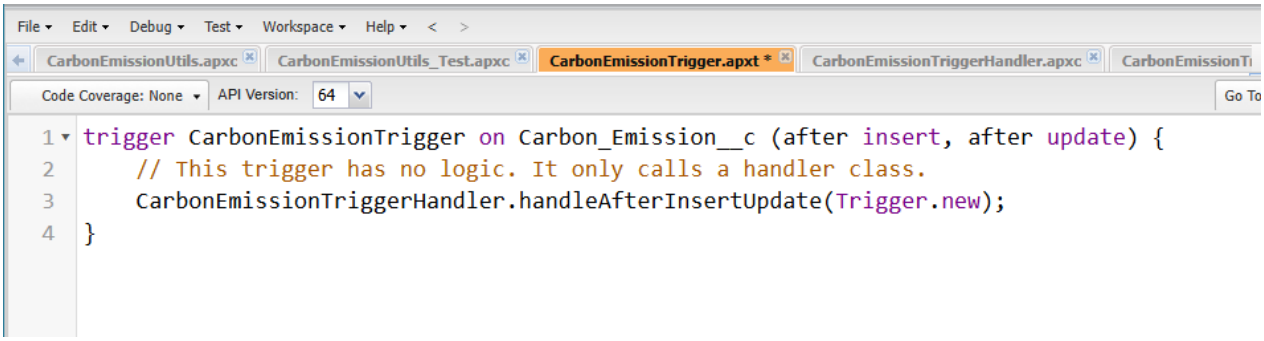


2. Apex Triggers & 3. Trigger Design Pattern

Developed: Trigger on Carbon_Emission__c for after insert, after update. Logic delegated to handler class.
Verification: Setup → Apex Triggers → CarbonEmissionTrigger contains single handler call.





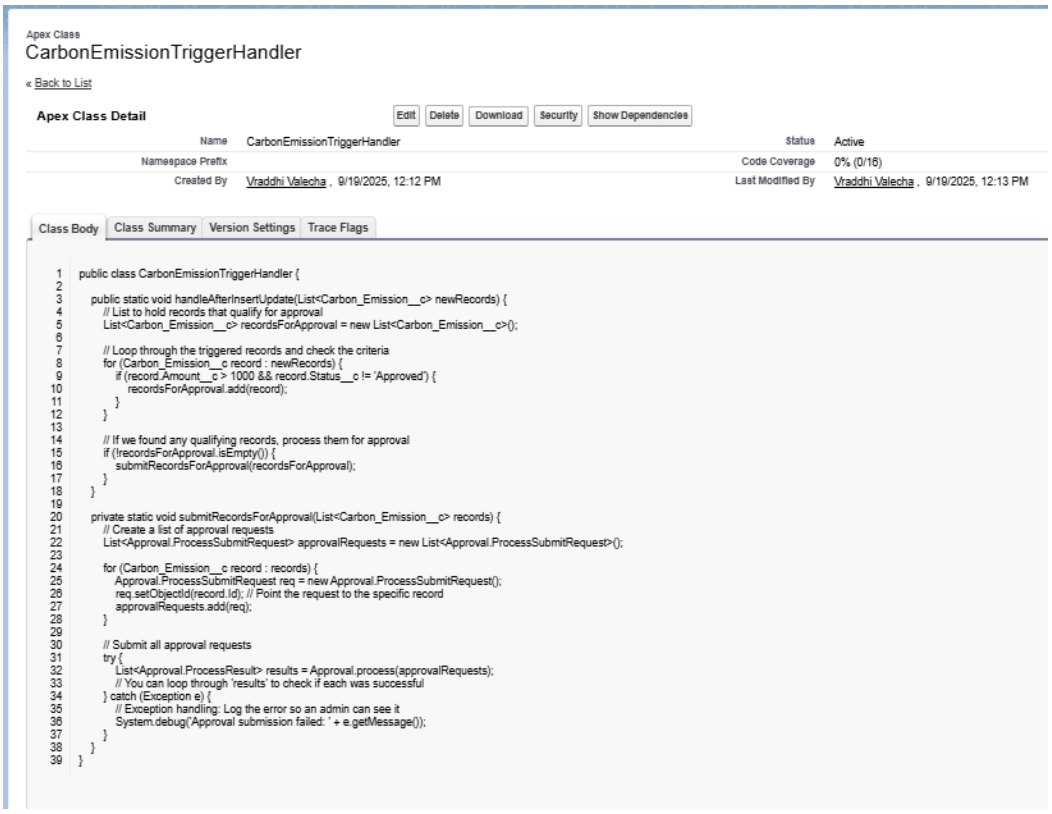


4. SOQL, 6. Control Statements, 7. Collections

Implemented:

- SOQL: Queries in batch class and tests.
- Collections: List<Carbon_Emission__c> to manage records.
- Control Statements: if for checks, for loops for processing.

Verification: Code in CarbonEmissionTriggerHandler and OldEmissionsBatch.



Apex Class

CarbonEmissionTriggerHandler

« Back to List

Apex Class Detail

Edit | Delete | Download | Security | Show Dependencies

Name	CarbonEmissionTriggerHandler	Status	Active
Namespace Prefix		Code Coverage	0% (0/16)
Created By	Vraddhi Valecha , 9/19/2025, 12:12 PM	Last Modified By	Vraddhi Valecha , 9/19/2025, 12:13 PM

Class Body | Class Summary | Version Settings | Trace Flags

```
1  public class CarbonEmissionTriggerHandler {
2
3      public static void handleAfterInsertUpdate(List<Carbon_Emission__c> newRecords) {
4          // List to hold records that qualify for approval
5          List<Carbon_Emission__c> recordsForApproval = new List<Carbon_Emission__c>();
6
7          // Loop through the triggered records and check the criteria
8          for (Carbon_Emission__c record : newRecords) {
9              if (record.Amount__c > 1000 && record.Status__c != 'Approved') {
10                 recordsForApproval.add(record);
11             }
12         }
13
14         // If we found any qualifying records, process them for approval
15         if (!recordsForApproval.isEmpty()) {
16             submitRecordsForApproval(recordsForApproval);
17         }
18     }
19
20     private static void submitRecordsForApproval(List<Carbon_Emission__c> records) {
21         // Create a list of approval requests
22         List<Approval.ProcessSubmitRequest> approvalRequests = new List<Approval.ProcessSubmitRequest>();
23
24         for (Carbon_Emission__c record : records) {
25             Approval.ProcessSubmitRequest req = new Approval.ProcessSubmitRequest();
26             req.setObjectId(record.Id); // Point the request to the specific record
27             approvalRequests.add(req);
28         }
29
30         // Submit all approval requests
31         try {
32             List<Approval.ProcessResult> results = Approval.process(approvalRequests);
33             // You can loop through 'results' to check if each was successful
34         } catch (Exception e) {
35             // Exception handling: Log the error so an admin can see it
36             System.debug('Approval submission failed: ' + e.getMessage());
37         }
38     }
39 }
```

```
// Query locator to gather all the records to process
global Database.QueryLocator start(Database.BatchableContext bc) {
    String query = 'SELECT Id, Name, High_Emission_Flag__c FROM Carbon_Emission__c ' +
                   'WHERE CreatedDate < LAST_N_DAYS:30 ' +
                   'AND Status__c != \'Approved\' ' +
                   'AND Amount__c > 1000';
    return Database.getQueryLocator(query);
}
```

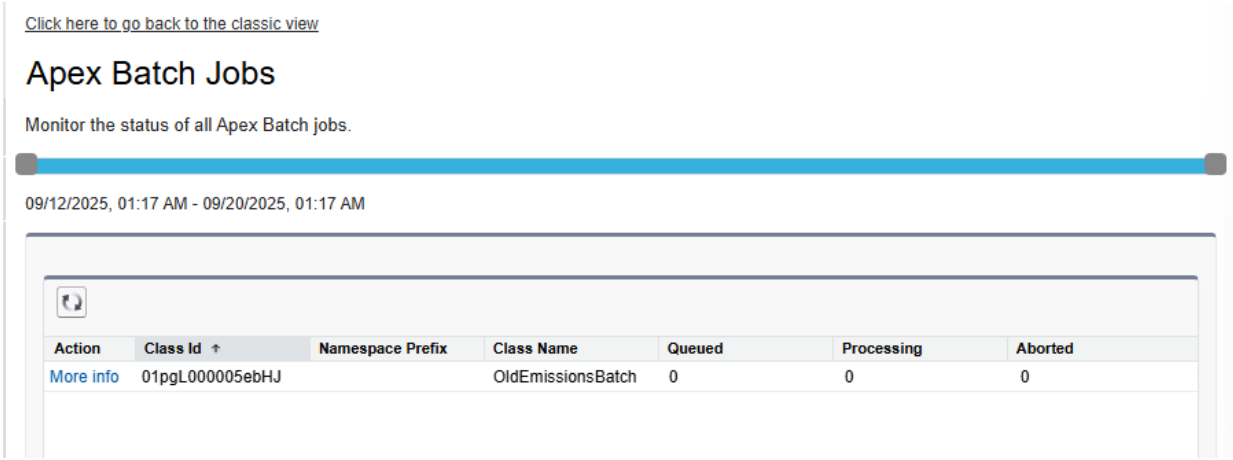
```
// Loop through the triggered records and check the criteria
for (Carbon_Emission__c record : newRecords) {
    if (record.Amount__c > 1000 && record.Status__c != 'Approved') {
        recordsForApproval.add(record);
    }
}
```

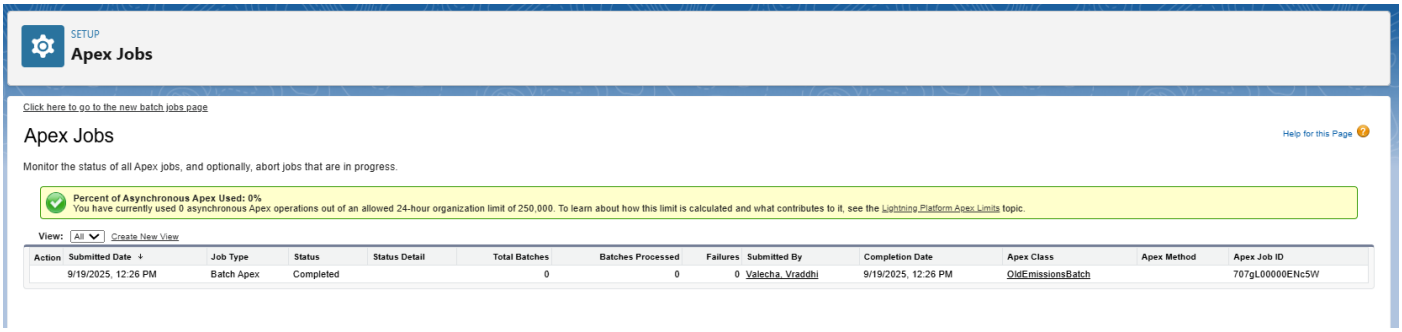
```
// List to hold records that qualify for approval
List<Carbon_Emission__c> recordsForApproval = new List<Carbon_Emission__c>();
```

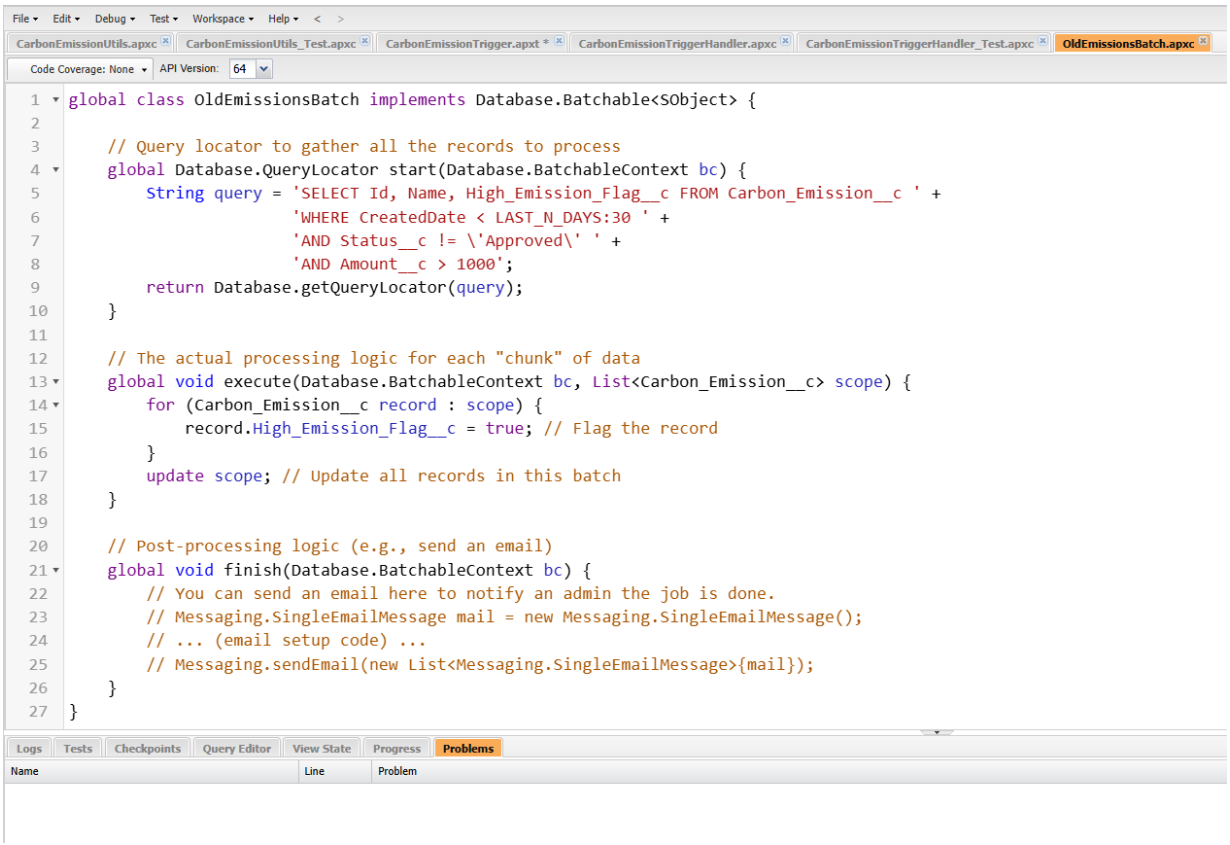
---

5. Batch Apex

Developed: OldEmissionsBatch to flag old unapproved records.
Verification: Setup → Apex Jobs → Completed job for OldEmissionsBatch







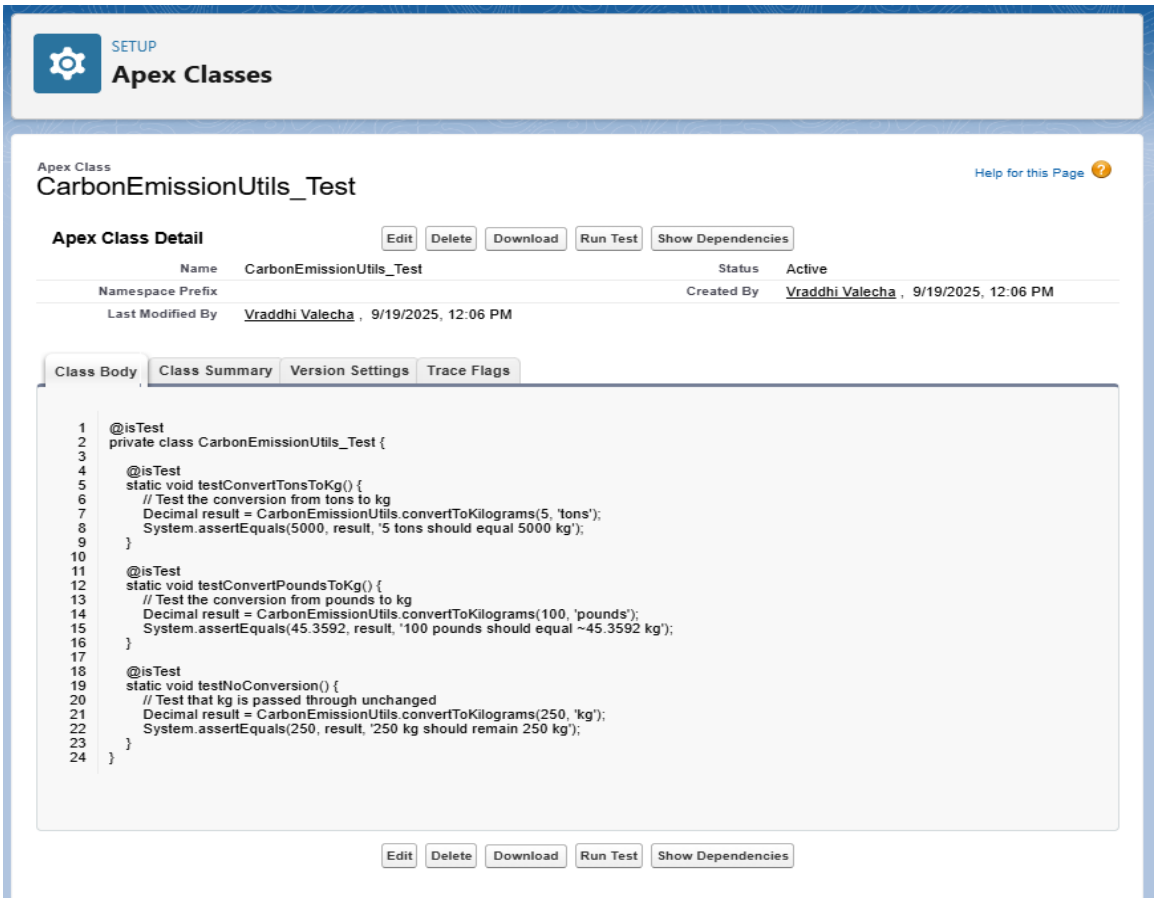
8. Test Classes

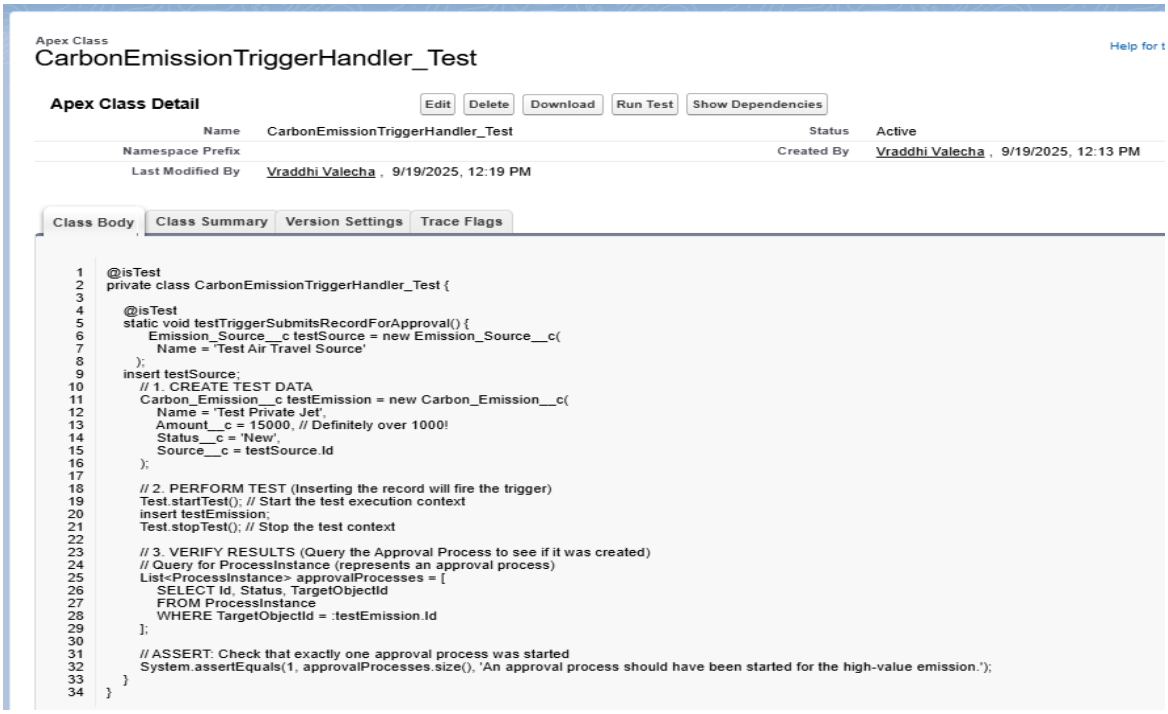
Developed:

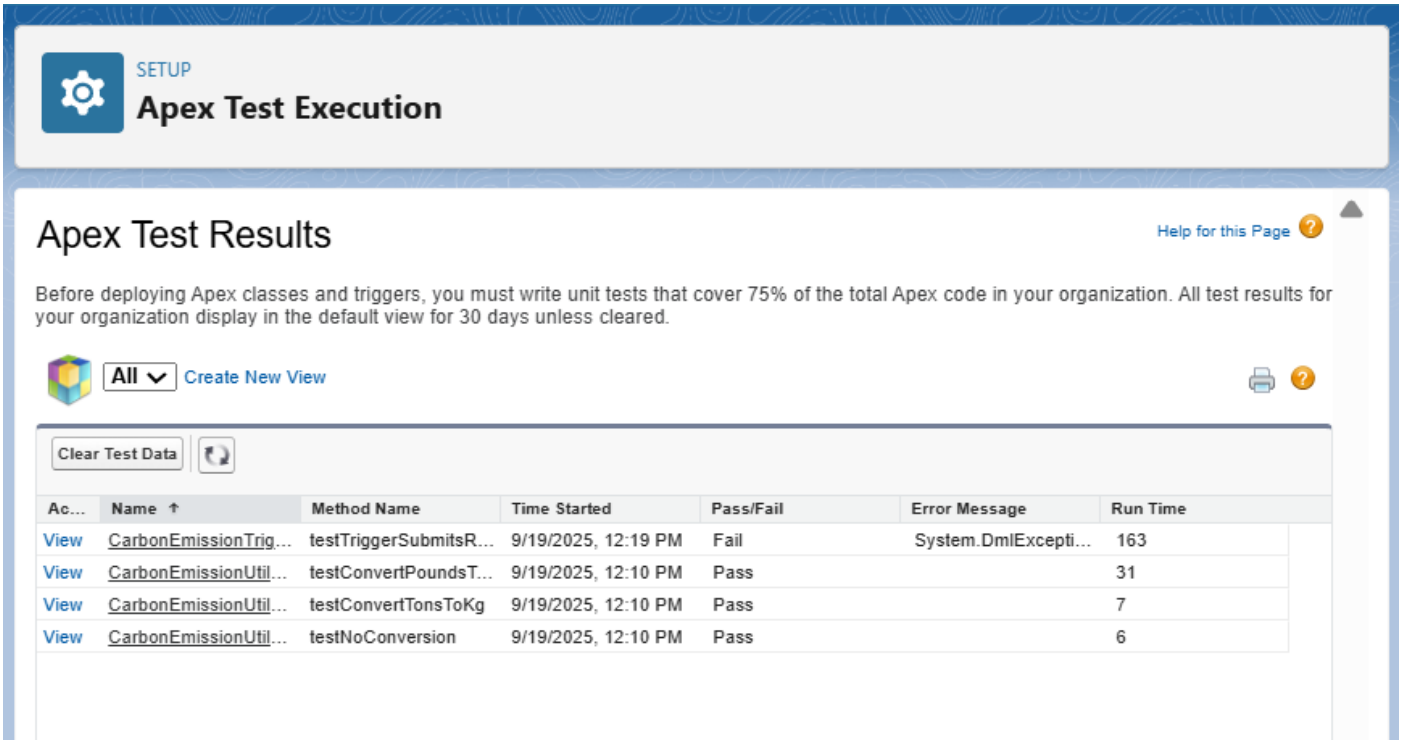
- Test classes for utilities and triggers (>75% coverage).

Verification:

- Code in CarbonEmissionUtils_Test and CarbonEmissionTriggerHandler_Test.
- Setup → Apex Test Execution → Successful test runs.







9. Exception Handling

Developed:

- try-catch block in handler for error handling.

Verification:

- try-catch in handler code.

```
29
30          // Submit all approval requests
31          try {
32              List<Approval.ProcessResult> results = Approval.process(approvalRequests);
33              // You can loop through 'results' to check if each was successful
34          } catch (Exception e) {
35              // Exception handling: Log the error so an admin can see it
36              System.debug('Approval submission failed: ' + e.getMessage());
37          }
38      }
39  }
```