

# Phase 5 : Apex Programming(Developer)

## Purpose

Phase 5 automates processes in the GreenFuture CRM system using **Apex**:

1. Automatically create **Carbon Credits** for completed projects.
  2. Notify project owners to review credits.
  3. Expire old credits after 1 year using **Batch Apex**.
  4. Integrate external **Carbon Price** data via HTTP callouts.
  5. Ensure all processes are **testable** and **validation-safe**.
- 

## Scenario Overview

1. **Scenario 1: Create Carbon Credits**
  - When a SustainabilityProject\_\_c record is marked **Completed**, a Carbon\_Credit\_\_c record is automatically generated based on project budget.
  - Example: Project “Tree Plantation” with budget \$10,000 → creates 10 Carbon Credits.
2. **Scenario 2: Notification Task**
  - After credit creation, a **Task** is created for the project owner:  
“Review Credits for Completed Project”
3. **Scenario 3: Expire Old Credits**
  - Credits older than 1 year are automatically marked **Inactive** via **Batch Apex**.
4. **Scenario 4: Carbon Price Integration**
  - Future method fetches the latest carbon price from an external API asynchronously for reporting.

## 1. Classes & Objects

**Purpose:** Create classes for handling Carbon Credits and project logic.

**Classes:** CarbonCreditManager.cls, SustainabilityProjectTriggerHandler.cls

**Steps:**

1. Open **Developer Console** → **File** → **New** → **Apex Class**.
2. Paste code for CarbonCreditManager.cls → Save.
3. Repeat for SustainabilityProjectTriggerHandler.cls.
4. Ensure all required fields and objects exist in Salesforce.

The screenshot shows the Salesforce Setup Apex Classes page. The left sidebar has sections for Email, Custom Code (with Apex Classes selected), Apex Settings, Apex Test Execution, Apex Test History, Apex Triggers, Environments, and Jobs. The main area displays a table of Apex classes with columns for Action, Name, Namespace Prefix, API Version, Status, Size Without Comments, Last Modified By, and Has Trace Flags. The table includes rows for CarbonCreditManager, CarbonCreditQueries, CarbonPriceService, ChangePasswordController, ChangePasswordControllerTest, CommunitiesLandingController, CommunitiesLandingControllerTest, CommunitiesLoginController, CommunitiesLoginControllerTest, CommunitiesSelfRegConfirmController, CommunitiesSelfRegConfirmControllerTest, CommunitiesSelfRegController, and CommunitiesSelfRegControllerTest.

The screenshot shows the Developer Console with the file CarbonCreditManager.apxc open. The code implements a class named CarbonCreditManager with a static void method createCreditsForCompletedProjects that takes a list of SustainabilityProject\_\_c objects as input. It iterates through the projects, creates a new Carbon\_Credit\_\_c object for each completed project, and adds it to a list of creditsToInsert. Finally, it inserts all the credits into the database. The code also includes a try block to handle any exceptions.

```

1 public with sharing class CarbonCreditManager {
2     public static void createCreditsForCompletedProjects(List<SustainabilityProject__c> projects) {
3         if (projects == null || projects.isEmpty()) return;
4
5         List<Carbon_Credit__c> creditsToInsert = new List<Carbon_Credit__c>();
6
7         for (SustainabilityProject__c p : projects) {
8             if (p.Status__c == 'Completed') {
9                 Carbon_Credit__c c = new Carbon_Credit__c();
10                c.Name = p.Name + ' Credit';
11                Decimal creditVal = 100;
12                if (p.Project_Budget__c != null && p.Project_Budget__c > 0) {
13                    creditVal = Math.max(1, (p.Project_Budget__c / 1000).setScale(0));
14                }
15                c.Credit_Amount__c = creditVal;
16                c.Type__c = 'Earned';
17                c.Date_Recorded__c = Date.today();
18                c.Project__c = p.Id;
19                c.Status__c = 'Active';
20                c.Source__c = 'Internal';
21                creditsToInsert.add(c);
22            }
23        }
24
25        if (!creditsToInsert.isEmpty()) {
26            try {

```

Fig.5.1. Apex Classes List and Code

## 2. Apex Triggers (before/after insert/update/delete)

**Purpose:** Trigger logic when Sustainability Projects are updated.

**Trigger:** SustainabilityProjectTrigger.trigger

**Steps:**

1. Open **Developer Console** → **File** → **New** → **Apex Trigger**.
2. Select object: **SustainabilityProject\_\_c**.
3. Paste trigger code → Save.
4. Verify trigger is active in **Setup** → **Object Manager** → **SustainabilityProject\_\_c** → **Triggers**.

The image contains two screenshots of the Salesforce interface. The top screenshot shows the 'Object Manager' page for the 'Sustainability Project' object. The sidebar on the left lists various object settings like Details, Fields & Relationships, Page Layouts, etc. The main area is titled 'Triggers' and shows one item: 'SustainabilityProjectTrigger'. The bottom screenshot shows the 'Apex Triggers' page for the 'SustainabilityProjectTrigger'. The sidebar on the left has a search bar and sections for Email, Custom Code (with 'Apex Classes' selected), Environments, and Jobs. The main area displays the 'Apex Trigger Detail' for 'SustainabilityProjectTrigger', showing details like Name, Status, and the trigger code itself.

**Screenshot 1: Object Manager - Sustainability Project**

LABEL	API VERSION	SIZE WITHOUT COMMENTS	MODIFIED BY
SustainabilityProjectTrigger	64.0	223	Eshwari Santosh Lade, 27/09/2025, 3:13 pm

**Screenshot 2: Apex Triggers - SustainabilityProjectTrigger**

Apex Trigger Detail

Name	SustainabilityProjectTrigger	eObject Type	Sustainability Project
Code Coverage	0% (0/2)	Status	Active
Created By	Eshwari Santosh Lade, 27/09/2025, 3:12 pm	Last Modified By	Eshwari Santosh Lade, 27/09/2025, 3:13 pm
Namespace Prefix			

```
Trigger SustainabilityProjectTrigger on SustainabilityProject__c (after update) {
    for (Trigger__r && Trigger__u update) {
        SustainabilityProjectTriggerHandler__c afterUpdate(Trigger__r, Trigger__u);
    }
}
```

Fig.5.2. Apex Triggers List

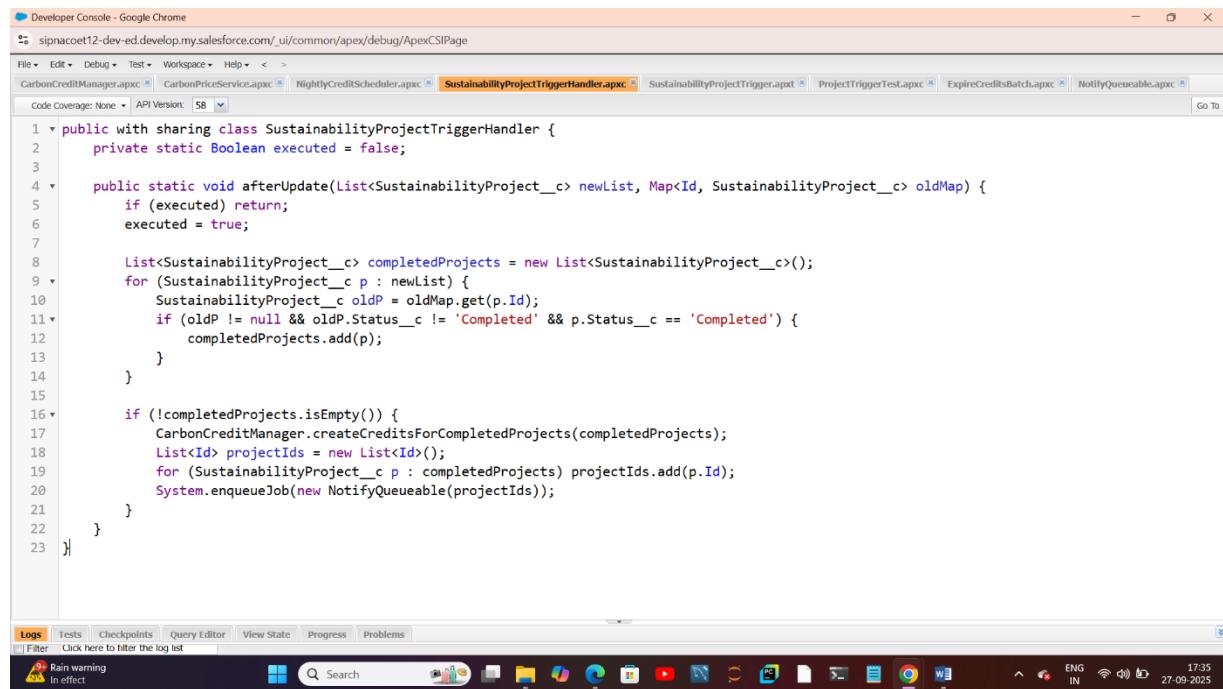
### 3. Trigger Design Pattern

**Purpose:** Use handler class to keep triggers clean.

**Handled in:** SustainabilityProjectTriggerHandler.cls

**Steps:**

- All trigger logic in handler class.
- Trigger simply calls handler method.



The screenshot shows the Salesforce Developer Console in Google Chrome. The title bar says 'Developer Console - Google Chrome'. The address bar shows the URL: 'sipnacoe12-dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage'. The tabs at the top include 'CarbonCreditManager.apxc', 'CarbonPriceService.apxc', 'NightlyCreditScheduler.apxc', 'SustainabilityProjectTriggerHandler.apxc' (which is the active tab), 'SustainabilityProjectTrigger.apxc', 'ProjectTriggerTest.apxc', 'ExpireCreditsBatch.apxc', and 'NotifyQueueable.apxc'. Below the tabs, there's a dropdown for 'Code Coverage: None' and 'API Version: 58'. The main area contains the Apex code for the SustainabilityProjectTriggerHandler class:

```
1 public with sharing class SustainabilityProjectTriggerHandler {
2     private static Boolean executed = false;
3
4     public static void afterUpdate(List<SustainabilityProject__c> newList, Map<Id, SustainabilityProject__c> oldMap) {
5         if (executed) return;
6         executed = true;
7
8         List<SustainabilityProject__c> completedProjects = new List<SustainabilityProject__c>();
9         for (SustainabilityProject__c p : newList) {
10             SustainabilityProject__c oldP = oldMap.get(p.Id);
11             if (oldP != null && oldP.Status__c != 'Completed' && p.Status__c == 'Completed') {
12                 completedProjects.add(p);
13             }
14         }
15
16         if (!completedProjects.isEmpty()) {
17             CarbonCreditManager.createCreditsForCompletedProjects(completedProjects);
18             List<Id> projectIds = new List<Id>();
19             for (SustainabilityProject__c p : completedProjects) projectIds.add(p.Id);
20             System.enqueueJob(new NotifyQueueable(projectIds));
21         }
22     }
23 }
```

At the bottom of the developer console, there are tabs for 'Logs', 'Tests', 'Checkpoints', 'Query Editor', 'View State', 'Progress', and 'Problems'. The 'Logs' tab is selected. A status bar at the bottom shows 'Rain warning In effect', system icons, and the date/time '27-09-2025 17:35'.

Fig.5.3. Handler Pattern

### 4. SOQL & SOSL

**Purpose:** Query Salesforce records.

**Used in:** CarbonCreditManager.cls, Test Classes.

**Steps:**

- Query SustainabilityProject\_\_c to check status.
- Query Carbon\_Credit\_\_c for test validations.

```

1 public with sharing class CarbonCreditQueries {
2
3     /**
4      * Get all active Carbon_Credit__c records for a given project
5      */
6     public static List<Carbon_Credit__c> getcreditsForProject(Id projectId) {
7         if (projectId == null) return new List<Carbon_Credit__c>();
8         return [SELECT Id, Name, Credit_Amount__c, Type__c, Date_Recorded__c, Status__c
9                 FROM Carbon_Credit__c
10                WHERE Project__c = :projectId AND Status__c = 'Active'];
11    }
12 }

```

Fig.5.4. CarbonCreditQueries Code

## 5. Collections: List, Set, Map

**Purpose:** Store multiple records and handle bulk operations.

**Used in:** CarbonCreditManager.cls, NotifyQueueable.cls, ExpireCreditsBatch.cls

**Steps:**

- Create lists for credits to insert.
- Use maps for old/new project comparison in triggers.

```

1 public class NotifyQueueable implements Queueable {
2     private List<Id> projectIds;
3     public NotifyQueueable(List<Id> projectIds) {
4         this.projectIds = projectIds;
5     }
6
7     public void execute(QueueableContext ctx) {
8         if (projectIds == null || projectIds.isEmpty()) return;
9
10        List<Task> tasks = new List<Task>();
11        for (Id pid : projectIds) {
12            Task t = new Task(
13                Subject = 'Review Credits for Completed Project',
14                WhatId = pid,
15                OwnerId = UserInfo.getUserId(),
16                Status = 'Not Started',
17                Priority = 'Normal'
18            );
19            tasks.add(t);
20        }
21        if (!tasks.isEmpty()) insert tasks;
22    }
23 }

```

Fig.5.5. NotifyQueueable.cls Code

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is `sipnacoe12-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage`. The tab bar includes files like CarbonPriceService.apxc, NightlyCreditScheduler.apxc, SustainabilityProjectTriggerHandler.apxc, SustainabilityProjectTrigger.apxt, ProjectTriggerTestLapxc, **ExpireCreditsBatch.apxc**, NotifyQueueable.apxc, and CarbonCreditQueries.apxc. The code editor displays the following Apex code:

```
1 global class ExpireCreditsBatch implements Database.Batchable<sObject> {
2     global Database.QueryLocator start(Database.BatchableContext BC) {
3         Date cutOff = Date.today().addYears(-1);
4         return Database.getQueryLocator([
5             SELECT Id, Status__c, Date_Recorded__c
6             FROM Carbon_Credit__c
7             WHERE Status__c = 'Active' AND Date_Recorded__c <= :cutOff
8         ]);
9     }
10    global void execute(Database.BatchableContext BC, List<Carbon_Credit__c> scope) {
11        for (Carbon_Credit__c c : scope) {
12            c.Status__c = 'Inactive';
13        }
14        if (!scope.isEmpty()) update scope;
15    }
16    global void finish(Database.BatchableContext BC) {
17        System.debug('ExpireCreditsBatch finished at ' + Datetime.now());
18    }
19 }
20 }
```

The Query Editor tab is selected. The status bar shows weather (24°C Rain), search, file icons, and system info (ENG IN, 17:42, 27-09-2025).

Fig.5.6. ExpireCreditsBatch.cls

## 6. Control Statements

**Purpose:** Logic and error handling.

**Used in:** CarbonCreditManager.cls, CarbonPriceService.cls

### Steps:

- Validate project status before creating credit.
- Handle exceptions in insert/update operations.

```

1 public with sharing class CarbonCreditManager {
2     public static void createCreditsForCompletedProjects(List<SustainabilityProject__c> projects) {
3         if (projects == null || projects.isEmpty()) return;
4
5         List<Carbon_Credit__c> creditsToInsert = new List<Carbon_Credit__c>();
6
7         for (SustainabilityProject__c p : projects) {
8             if (p.Status__c == 'Completed') {
9                 Carbon_Credit__c c = new Carbon_Credit__c();
10                c.Name = p.Name + ' Credit';
11                Decimal creditVal = 100;
12                if (p.Project_Budget__c != null && p.Project_Budget__c > 0) {
13                    creditVal = Math.max(1, (p.Project_Budget__c / 1000).setScale(0));
14                }
15                c.Credit_Amount__c = creditVal;
16                c.Type__c = 'Earned';
17                c.Date_Recorded__c = Date.today();
18                c.Project__c = p.Id;
19                c.Status__c = 'Active';
20                c.Source__c = 'Internal';
21                creditsToInsert.add(c);
22            }
23        }
24
25        if (!creditsToInsert.isEmpty()) {
26            try {

```

Logs Tests Checkpoints Query Editor View State Progress Problems

Rain warning In effect

17:29 27-09-2025

Fig.5.7. CarbonCreditManager Loops & Conditions

## 7. Batch Apex

**Purpose:** Expire old Carbon Credits.

**Class:** ExpireCreditsBatch.cls

**Steps:**

Create Batch Apex class.

Execute via **Developer Console** → **Execute Anonymous**:

Database.executeBatch(new ExpireCreditsBatch(), 200);

Check Carbon\_Credit\_\_c status updates.

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is `sipnacoe12-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage`. The tab bar includes files like CarbonPriceService.apxc, NightlyCreditScheduler.apxc, SustainabilityProjectTriggerHandler.apxc, SustainabilityProjectTrigger.apxt, ProjectTriggerTest.apxc, **ExpireCreditsBatch.apxc**, NotifyQueueable.apxc, and CarbonCreditQueries.apxc. The Query Editor tab is selected. The code editor contains the following Apex code:

```
1 global class ExpireCreditsBatch implements Database.Batchable<sObject> {
2     global Database.QueryLocator start(Database.BatchableContext BC) {
3         Date cutOff = Date.today().addYears(-1);
4         return Database.getQueryLocator([
5             SELECT Id, Status__c, Date_Recorded__c
6             FROM Carbon_Credit__
7             WHERE Status__c = 'Active' AND Date_Recorded__c <= :cutOff
8         ]);
9     }
10    global void execute(Database.BatchableContext BC, List<Carbon_Credit__c> scope) {
11        for (Carbon_Credit__c c : scope) {
12            c.Status__c = 'Inactive';
13        }
14        if (!scope.isEmpty()) update scope;
15    }
16    global void finish(Database.BatchableContext BC) {
17        System.debug('ExpireCreditsBatch finished at ' + Datetime.now());
18    }
19 }
20 }
```

The status bar at the bottom shows "Logs Tests Checkpoints **Query Editor** View State Progress Problems" and "24°C Rain". The system tray shows "ENG IN" and the date "27-09-2025".

Fig.5.7. ExpireCreditsBatch Code

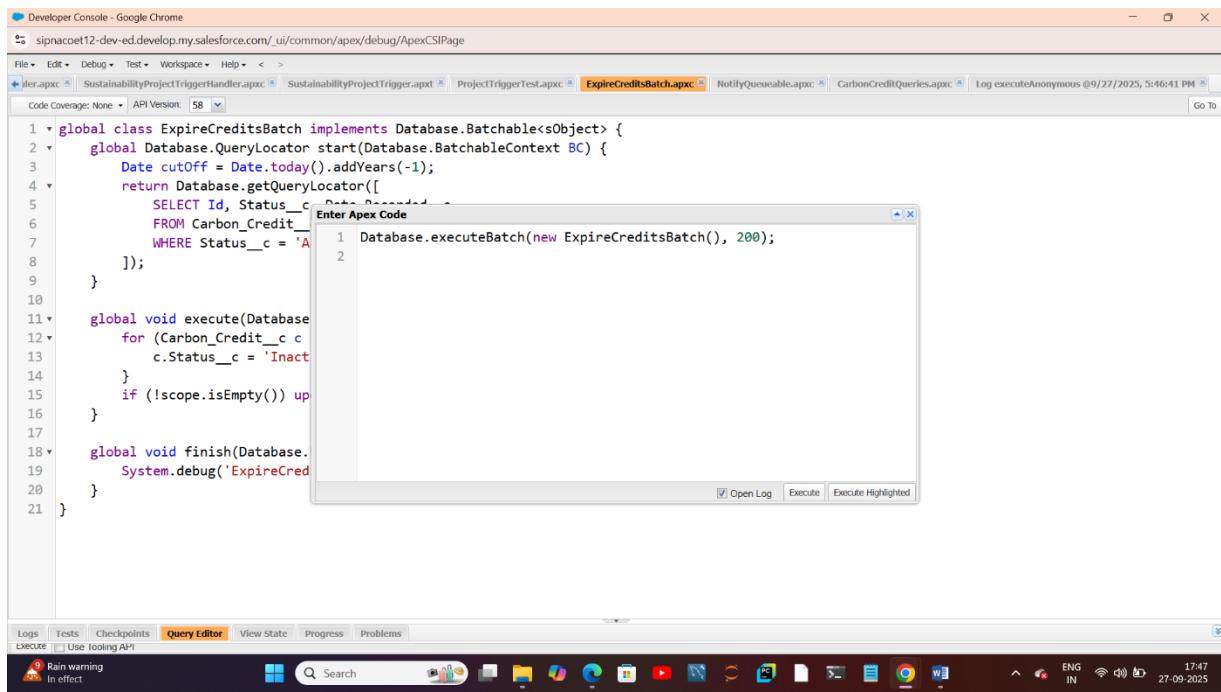


Fig.5.8. Execute Anonymous window.

The screenshot shows the Developer Console in Google Chrome with the URL `sipnacoet12-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage`. The tab bar includes files like `ller.apxc`, `SustainabilityProjectTriggerHandler.apxc`, `SustainabilityProjectTrigger.apxc`, `ProjectTriggerTest.apxc`, `ExpireCreditsBatch.apxc`, `NotifyQueueable.apxc`, and `CarbonCreditQueries.apxc`. The main area is titled "Execution Log" and displays a table of log entries. The first entry is `USER_INFO` at timestamp 17:46:41:003, detailing a user from India Standard Time (Asia/Kolkata) with ID 005dM000000mgJ2j. Subsequent entries show various heap allocations and database interactions, such as `EXECUTION_STARTED`, `CODE_UNIT_STARTED`, `HEAP_ALLOCATE` for system methods, and `VARIABLE_ASSIGNMENT` for database batchable context. The log ends with `METHOD_EXIT` and `HEAP_ALLOCATE` for the `NotifyQueueable` class. The bottom of the screen shows the Windows taskbar with icons for File Explorer, Task View, and various applications, along with system status indicators like battery level and signal strength.

Fig.5.9. Query results showing expired credits.

## 8. Queueable Apex

**Purpose:** Notify Project Owner asynchronously.

**Class:** NotifyQueueable.cls

**Steps:**

- Enqueue job from Trigger Handler
- Tasks are created asynchronously.

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is `sipnacoe12-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage`. The tab bar at the top includes CarbonCreditManager.apxc, CarbonPriceService.apxc, NightlyCreditScheduler.apxc, SustainabilityProjectTriggerHandler.apxc, SustainabilityProjectTrigger.apxc, ProjectITTriggerTest.apxc, ExpireCreditsBatch.apxc, and NotifyQueueable.apxc (which is currently selected). Below the tabs, there's a code editor window with the following Apex code:

```
1 public class NotifyQueueable implements Queueable {
2     private List<Id> projectIds;
3     public NotifyQueueable(List<Id> projectIds) {
4         this.projectIds = projectIds;
5     }
6     public void execute(QueueableContext ctx) {
7         if (projectIds == null || projectIds.isEmpty()) return;
8
9         List<Task> tasks = new List<Task>();
10        for (Id pid : projectIds) {
11            Task t = new Task(
12                Subject = 'Review Credits for Completed Project',
13                WhatId = pid,
14                OwnerId = UserInfo.getUserId(),
15                Status = 'Not Started',
16                Priority = 'Normal'
17            );
18            tasks.add(t);
19        }
20        if (!tasks.isEmpty()) insert tasks;
21    }
22 }
```

Below the code editor, there are tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Logs tab is selected. The status bar at the bottom shows the date and time as 27-09-2025, 18:20.

Fig.5.10. NotifyQueueable Code

## 9. Scheduled Apex

**Purpose:** Nightly batch execution.

**Class:** NightlyCreditScheduler.cls

**Steps:**

1. Go to **Setup** → **Apex Classes** → **Schedule Apex**.
2. Select **NightlyCreditScheduler**.
3. Set schedule (e.g., every night 12:00 AM).

The screenshot shows the Salesforce Setup interface with the 'Apex Classes' page selected. The sidebar on the left has sections for Email, Custom Code (with 'Apex Classes' highlighted), Environments, and Jobs. A search bar at the top right contains 'Search Setup'. The main content area displays a table of Apex classes, with one row highlighted in blue. The table includes columns for Name, Status, Active, and Last Modified.

Name	Status	Active	Last Modified
LightningLoginFormController	Active	1,069	Eshwari Santosh Lade, 20/09/2020, 12:31 am
LightningLoginFormControllerTest	Active	1,064	Eshwari Santosh Lade, 25/09/2025, 12:31 am
LightningSelfRegisterController	Active	896	Eshwari Santosh Lade, 25/09/2025, 12:31 am
LightningSelfRegisterControllerTest	Active	5,031	Eshwari Santosh Lade, 25/09/2025, 12:31 am
MicrobatchSelfRegController	Active	1,779	Eshwari Santosh Lade, 25/09/2025, 12:31 am
MicrobatchSelfRegControllerTest	Active	496	Eshwari Santosh Lade, 25/09/2025, 12:31 am
MyProfilePageController	Active	1,403	Eshwari Santosh Lade, 25/09/2025, 12:31 am
MyProfilePageControllerTest	Active	2,507	Eshwari Santosh Lade, 25/09/2025, 12:31 am
NightlyCreditScheduler	Active	180	Eshwari Santosh Lade, 27/09/2025, 3:09 pm
NotifyQueueable	Active	726	Eshwari Santosh Lade, 27/09/2025, 6:18 pm
ProjectTriggerTest	Active	3,397	Eshwari Santosh Lade, 27/09/2025, 3:24 pm
SiteLoginController	Active	352	Eshwari Santosh Lade, 25/09/2025, 12:31 am
SiteLoginControllerTest	Active	393	Eshwari Santosh Lade, 25/09/2025, 12:31 am
SiteRegisterController	Active	1,524	Eshwari Santosh Lade, 25/09/2025, 12:31 am
SiteRegisterControllerTest	Active	559	Eshwari Santosh Lade, 25/09/2025, 12:31 am
SustainabilityProjectTriggerHandler	Active	1,005	Eshwari Santosh Lade, 27/09/2025, 6:17 pm

The screenshot shows the Salesforce Developer Console with the code editor open for the 'NightlyCreditScheduler.apxc' file. The code implements the `Schedulable` interface and contains a single method `execute` that performs a database batch operation.

```

global class NightlyCreditScheduler implements Schedulable {
    global void execute(SchedulableContext sc) {
        Database.executeBatch(new ExpireCreditsBatch(), 200);
    }
}

```

Fig.5.11. NightlyCreditScheduler Code

## 10. Future Methods

**Purpose:** Asynchronous HTTP callout.  
**Class:** CarbonPriceService.cls

**Steps:**

- Annotate method with @future(callout=true).
- Call method from another class or anonymous Apex:

```
CarbonPriceService.fetchCarbonPrice('https://example.com/api');
```

Developer Console - Google Chrome  
sipnacoe12-dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage

File • Edit • Debug • Test • Workspace • Help • < >

CarbonPriceService.apxc [ ] HighlyCreditScheduler.apxc [ ] SustainabilityProjectTriggerHandler.apxc [ ] SustainabilityProjectTrigger.apxt [ ] ProjectTriggerTest.apxc [ ] ExpireCreditsBatch.apxc [ ] NotifyQueueable.apxc [ ] Log executeAnonymous

Code Coverage: None • API Version: 58 • Go To

```
1 public with sharing class CarbonPriceService {
2     @future(callout=true)
3     public static void fetchCarbonPrice(String endpoint) {
4         if (String.isBlank(endpoint)) return;
5         Http http = new Http();
6         HttpRequest req = new HttpRequest();
7         req.setEndpoint(endpoint);
8         req.setMethod('GET');
9         try {
10             HttpResponse res = http.send(req)
11             if (res.getStatusCode() == 200) {
12                 System.debug('Carbon price re
13             } else {
14                 System.debug('Carbon price er
15             }
16         } catch (CalloutException ce) {
17             System.debug('Callout failed: ' +
18         }
19     }
20 }
```

Enter Apex Code

```
1 CarbonPriceService.fetchCarbonPrice('https://example.com/api');
2
```

Open Log Execute Execute Highlighted

Logs Tests Checkpoints Query Editor View State Progress Problems

Filter Click here to filter the log list

18:26 27-09-2025 ENG IN 27-09-2025

Fig.5.12.Code snippet with @future

Developer Console - Google Chrome  
sipnacoe12-dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage

File • Edit • Debug • Test • Workspace • Help • < >

KC [ ] HighlyCreditScheduler.apxc [ ] SustainabilityProjectTriggerHandler.apxc [ ] SustainabilityProjectTrigger.apxt [ ] ProjectTriggerTest.apxc [ ] ExpireCreditsBatch.apxc [ ] NotifyQueueable.apxc [ ] Log executeAnonymous @9/27/2025, 6:26:29 PM

Execution Log

Timestamp	Event	Details
18:26:29:002	USER_INFO	[EXTERNAL]005dM000000OrjJZ[eshwarilade2005@gmail.com](GMT+05:30) India Standard Time (Asia/Kolkata)(GMT+05:30)
18:26:29:002	EXECUTION_ST...	
18:26:29:002	CODE_UNIT_ST...	[EXTERNAL]execute_anonymous_apex
18:26:29:002	HEAP_ALLOCATE	[95]Bytes:3
18:26:29:002	HEAP_ALLOCATE	[100]Bytes:152
18:26:29:002	HEAP_ALLOCATE	[417]Bytes:408
18:26:29:002	HEAP_ALLOCATE	[430]Bytes:408
18:26:29:002	HEAP_ALLOCATE	[317]Bytes:6
18:26:29:002	HEAP_ALLOCATE	[EXTERNAL]1Bytes:1
18:26:29:002	STATEMENT_EX...	[1]J
18:26:29:002	STATEMENT_EX...	[1]J
18:26:29:002	HEAP_ALLOCATE	[1]Bytes:18
18:26:29:002	HEAP_ALLOCATE	[1]Bytes:16
18:26:29:002	HEAP_ALLOCATE	[1]Bytes:4
18:26:29:002	HEAP_ALLOCATE	[1]Bytes:23
18:26:29:002	HEAP_ALLOCATE	[1]Bytes:4
18:26:29:002	HEAP_ALLOCATE	[1]Bytes:6
18:26:29:003	HEAP_ALLOCATE	[68]Bytes:5
18:26:29:003	HEAP_ALLOCATE	[74]Bytes:5
18:26:29:003	HEAP_ALLOCATE	[82]Bytes:7
18:26:29:003	HEAP_ALLOCATE	[EXTERNAL]1Bytes:8
18:26:29:003	HEAP_ALLOCATE	[EXTERNAL]1Bytes:8
18:26:29:044	CUMULATIVE_L...	
18:26:29:000	LIMIT_USAGE...	(default)
18:26:29:000	LIMIT_USAGE...	Number of SOQL queries: 0 out of 100
18:26:29:000	LIMIT_USAGE...	Number of query rows: 0 out of 50000
18:26:29:000	LIMIT_USAGE...	Number of SOSL queries: 0 out of 20
18:26:29:000	LIMIT_USAGE...	Number of DML statements: 0 out of 150

Open Log Execute Execute Highlighted

Logs Tests Checkpoints Query Editor View State Progress Problems

Filter Click here to filter the log list

18:26 27-09-2025 ENG IN 27-09-2025

Fig.5.13. Debug log showing response from callout.

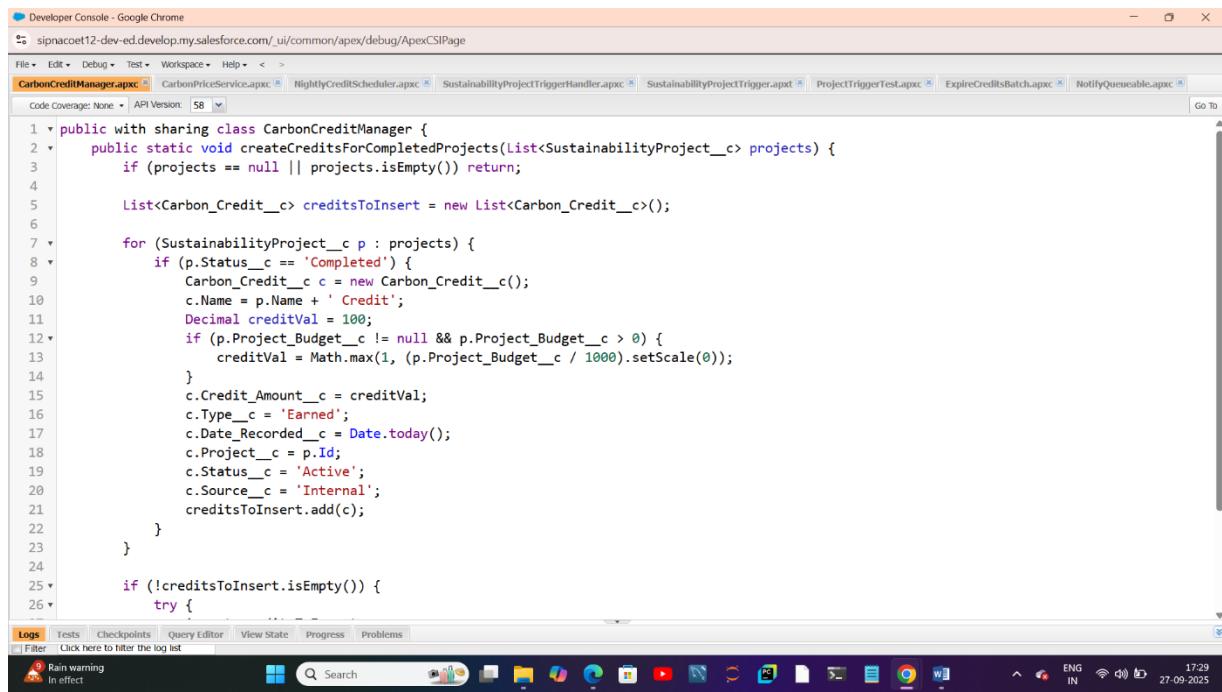
## 11.Exception Handling

**Purpose:** Handle DML and callout exceptions.

**Used in:** CarbonCreditManager.cls, CarbonPriceService.cls

**Steps:**

- Wrap insert/update in try-catch.
- Log errors using System.debug().



The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is `sipnacoe12-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage`. The tabs at the top include CarbonCreditManager.apxc, CarbonPriceService.apxc, and others. The code editor displays the following Apex code:

```
1 public with sharing class CarbonCreditManager {
2     public static void createCreditsForCompletedProjects(List<SustainabilityProject__c> projects) {
3         if (projects == null || projects.isEmpty()) return;
4
5         List<Carbon_Credit__c> creditsToInsert = new List<Carbon_Credit__c>();
6
7         for (SustainabilityProject__c p : projects) {
8             if (p.Status__c == 'Completed') {
9                 Carbon_Credit__c c = new Carbon_Credit__c();
10                c.Name = p.Name + ' Credit';
11                Decimal creditVal = 100;
12                if (p.Project_Budget__c != null && p.Project_Budget__c > 0) {
13                    creditVal = Math.max(1, (p.Project_Budget__c / 1000).setScale(0));
14                }
15                c.Credit_Amount__c = creditVal;
16                c.Type__c = 'Earned';
17                c.Date_Recorded__c = Date.today();
18                c.Project__c = p.Id;
19                c.Status__c = 'Active';
20                c.Source__c = 'Internal';
21                creditsToInsert.add(c);
22            }
23        }
24
25        if (!creditsToInsert.isEmpty()) {
26            try {

```

The status bar at the bottom shows "Logs Tests Checkpoints Query Editor View State Progress Problems" and a "Rain warning In effect" message. The system status bar indicates "ENG IN" and the date "27-09-2025".

Fig.5.14. Code snippet of try-catch block.

The screenshot shows the Salesforce Developer Console in Google Chrome. The title bar reads "Developer Console - Google Chrome". The address bar shows the URL "sipnacoe12-dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage". The tab bar includes "KCS", "NightlyCreditScheduler.apxc", "SustainabilityProjectTriggerHandler.apxc", "SustainabilityProjectTrigger.apxc", "ProjectTriggerTest.apxc", "ExpireCreditsBatch.apxc", "NotifyQueueable.apxc", and "Log executeAnonymous @9/27/2025, 6:26:29 PM". The main area has a header "Execution Log" with columns "Timestamp", "Event", and "Details". The log lists numerous events from 18:26:29:002 to 18:26:29:044, including USER\_INFO, EXECUTION\_ST..., CODE\_UNIT\_ST..., HEAP\_ALLOCATE, STATEMENT\_EX..., and LIMIT\_USAGE... entries. A modal window titled "Enter Apex Code" is open, containing the code "1 CarbonPriceService.fetchCarbon" and "2". At the bottom of the developer console, there are tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The status bar at the bottom right shows "Logs", "Tests", "Checkpoints", "Query Editor", "View State", "Progress", "Problems", "Filter", "Click here to filter the log", "18:29", "ENG IN", "27-09-2025", and a battery icon.

Fig. 5.15. Debug log showing caught exception.

## 12. Test Classes

**Purpose:** Validate all Apex functionality.

**Classes:** ProjectTriggerTest.cls, ExpireCreditsBatchTest.cls

**Steps:**

1. Create test records using safe templates.
2. Run Test.startTest() and Test.stopTest().
3. Assert expected outcomes: Carbon Credit created, expired, or task assigned.

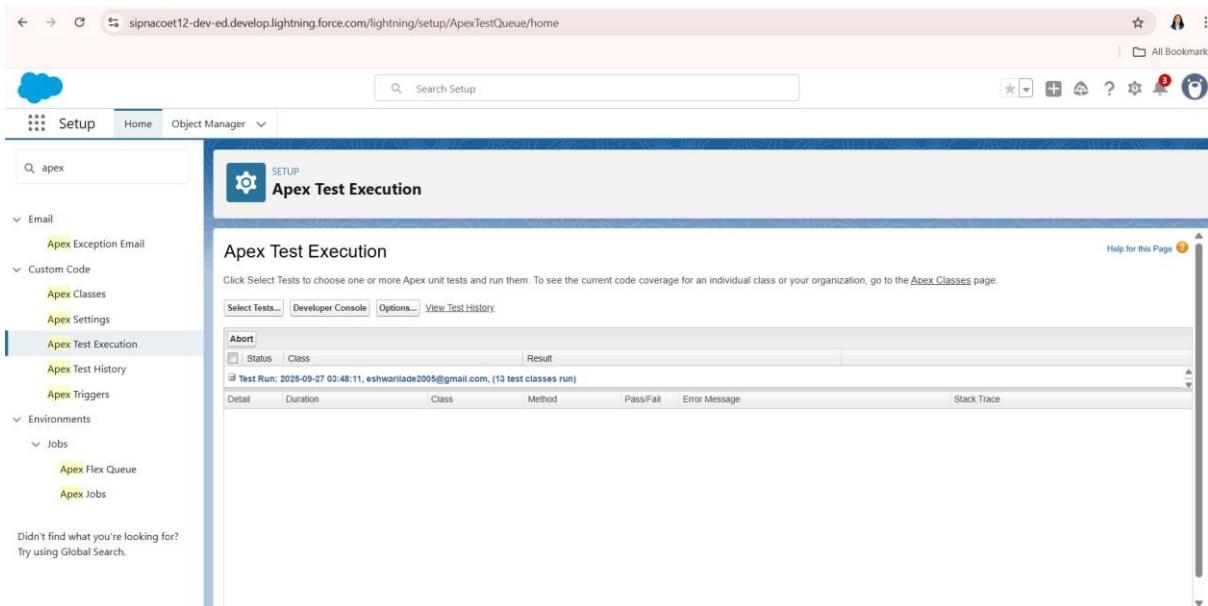


Fig.5.11. Apex Test Execution

User	Application	Operation	Time	Status	Read	Size
Eshwari Santosh Lade	Unknown	ApexTestHandler	9/27/2025, 6:37:47 PM	Success	Unread	5.93 KB
Eshwari Santosh Lade	Unknown	ApexTestHandler	9/27/2025, 6:37:46 PM	Success	Unread	5.39 KB
Eshwari Santosh Lade	Unknown	ApexTestHandler	9/27/2025, 6:37:46 PM	Success	Unread	4.98 KB
Eshwari Santosh Lade	Unknown	ApexTestHandler	9/27/2025, 6:37:46 PM	Success	Unread	16.17 KB
Eshwari Santosh Lade	Unknown	ApexTestHandler	9/27/2025, 6:37:46 PM	Success	Unread	11 KB
Eshwari Santosh Lade	Unknown	ApexTestHandler	9/27/2025, 6:37:45 PM	Success	Unread	94.83 KB
Eshwari Santosh Lade	Unknown	ApexTestHandler	9/27/2025, 6:37:45 PM	Success	Unread	13.76 KB
Eshwari Santosh Lade	Unknown	ApexTestHandler	9/27/2025, 6:37:45 PM	Success	Unread	4.8 KB
Eshwari Santosh Lade	Unknown	ApexTestHandler	9/27/2025, 6:37:45 PM	Success	Unread	5.64 KB
Eshwari Santosh Lade	Unknown	ApexTestHandler	9/27/2025, 6:37:43 PM	Success	Unread	3.59 KB
Eshwari Santosh Lade	Unknown	ApexTestHandler	9/27/2025, 6:37:43 PM	Success	Unread	16.87 KB
Eshwari Santosh Lade	Unknown	ApexTestHandler	9/27/2025, 6:37:42 PM	Success	Unread	22.66 KB

Fig.5.12. Test Results

## 13. Asynchronous Processing

**Purpose:** Handle processes without blocking UI.

**Used in:** NotifyQueueable.cls, ExpireCreditsBatch.cls, NightlyCreditScheduler.cls, CarbonPriceService.cls

**Steps:**

- Queueable jobs for notifications.
  - Batch Apex for expiration.
  - Scheduled Apex for automation.
  - Future methods for callouts.
- 
- Covered with **Batch, Queueable, Scheduled**