

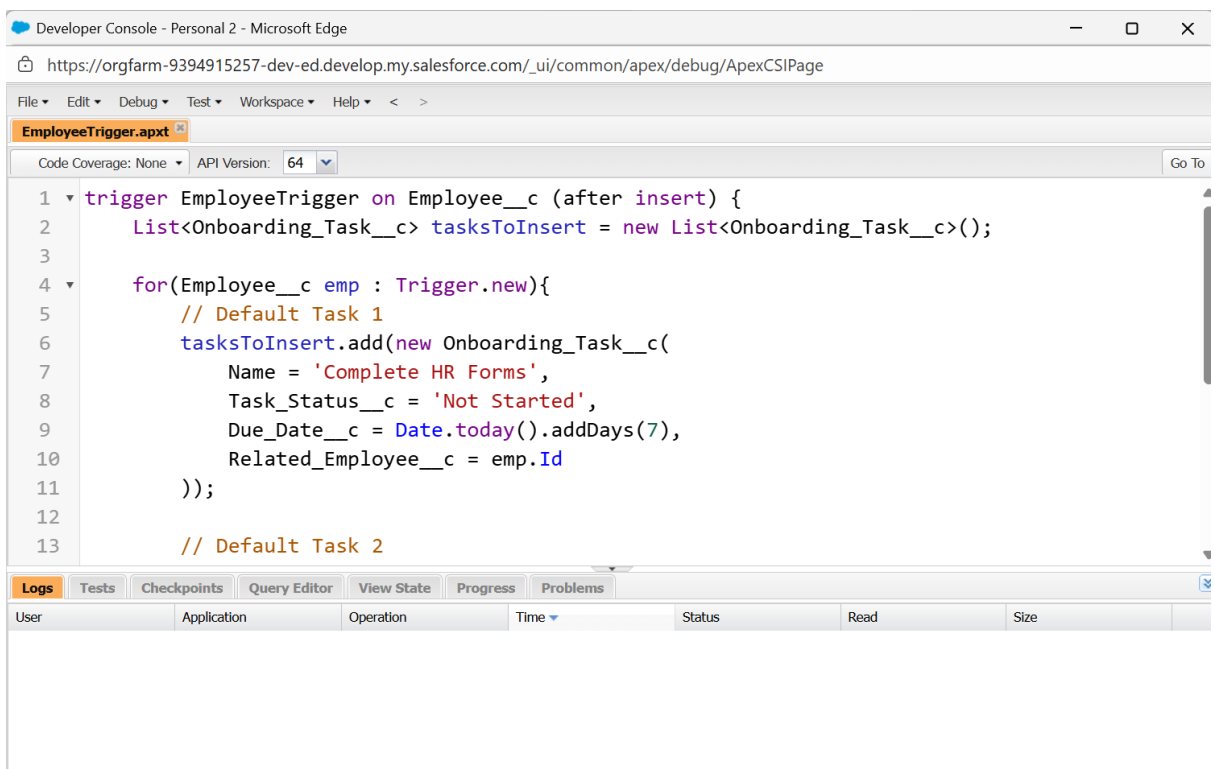
Phase 5: Apex Programming (Developer)

Objective

The purpose of this phase is to extend Salesforce functionality using **Apex programming**. In this project, Apex is used to automatically create onboarding tasks, assign training modules, and ensure code quality with test coverage. Apex enables logic that cannot be achieved through point-and-click tools alone.

Apex Trigger – Auto-Create Onboarding Tasks

- **Purpose:** Whenever a new Employee is created, assign default onboarding tasks automatically.
- **Trigger Code:**

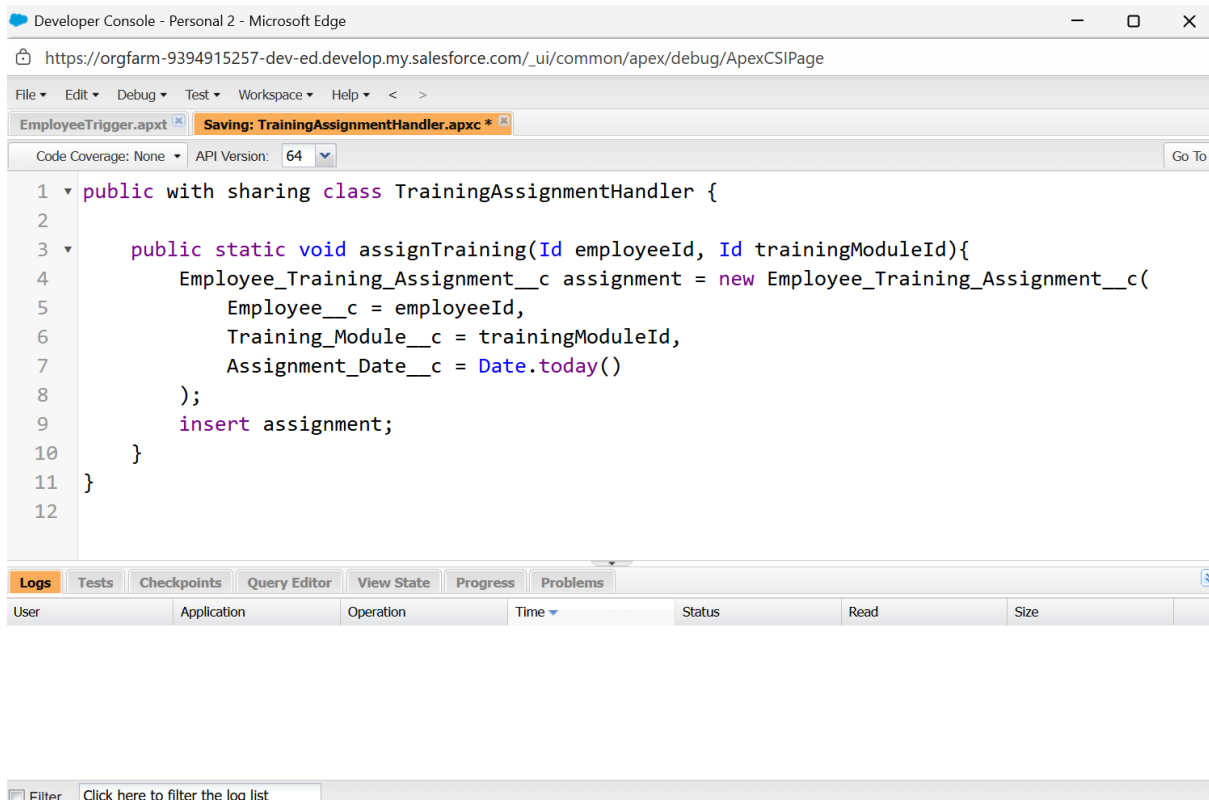


```
1 trigger EmployeeTrigger on Employee__c (after insert) {
2     List<Onboarding_Task__c> tasksToInsert = new List<Onboarding_Task__c>();
3
4     for(Employee__c emp : Trigger.new){
5         // Default Task 1
6         tasksToInsert.add(new Onboarding_Task__c(
7             Name = 'Complete HR Forms',
8             Task_Status__c = 'Not Started',
9             Due_Date__c = Date.today().addDays(7),
10            Related_Employee__c = emp.Id
11        ));
12
13        // Default Task 2
```

- **Expected Output:** Two tasks (“Complete HR Forms” and “Attend Orientation Session”) are created automatically for every new employee.

Apex Class – Assign Training Modules

- **Purpose:** Provide a reusable method to assign a training module to an employee.
- **Class Code:**



The screenshot shows the Salesforce Developer Console interface. The top bar indicates the user is logged in as 'Personal 2' using 'Microsoft Edge'. The browser address bar shows the URL: https://orgfarm-9394915257-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. The console has a menu bar with 'File', 'Edit', 'Debug', 'Test', 'Workspace', and 'Help'. Below the menu, there are tabs for 'EmployeeTrigger.apxt' and 'Saving: TrainingAssignmentHandler.apxc'. The main editor area displays the following Apex code:

```
1 public with sharing class TrainingAssignmentHandler {  
2  
3     public static void assignTraining(Id employeeId, Id trainingModuleId){  
4         Employee_Training_Assignment__c assignment = new Employee_Training_Assignment__c(  
5             Employee__c = employeeId,  
6             Training_Module__c = trainingModuleId,  
7             Assignment_Date__c = Date.today()  
8         );  
9         insert assignment;  
10    }  
11 }  
12
```

Below the code editor, there is a toolbar with buttons for 'Logs', 'Tests', 'Checkpoints', 'Query Editor', 'View State', 'Progress', and 'Problems'. The 'Logs' button is currently selected. Below the toolbar, there is a table with columns: 'User', 'Application', 'Operation', 'Time', 'Status', 'Read', and 'Size'. At the bottom of the console, there is a 'Filter' button and a link that says 'Click here to filter the log list'.

Expected Output: A new record in **Employee Training Assignment** is created linking the Employee and Training Module.

Apex Test Class – Ensuring Code Coverage

- **Purpose:** Verify Apex logic works correctly and achieve minimum 75% code coverage.
- **Test Class Code:**

The screenshot shows the Salesforce Developer Console interface. The top bar indicates the environment is 'Developer Console - Personal 2 - Microsoft Edge'. The address bar shows the URL: https://orgfarm-9394915257-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. The workspace contains three tabs: 'EmployeeTrigger.apxt', 'TrainingAssignmentHandler.apxc' (which is the active tab), and 'TrainingAssignmentHandlerTest.apxc'. Below the tabs, the 'Code Coverage' is set to 'None' and the 'API Version' is '64'. The main editor displays the following Apex code for the 'TrainingAssignmentHandler' class:

```
1 public with sharing class TrainingAssignmentHandler {
2
3     public static void assignTraining(Id employeeId, Id trainingModuleId){
4         if(employeeId == null || trainingModuleId == null){
5             System.debug('EmployeeId or TrainingModuleId is null. Skipping assignment.');
```

```
6             return;
7         }
8
9         try {
10             Employee_Training_Assignment__c assignment = new Employee_Training_Assignment__c(
11                 Employee__c = employeeId,
12                 Training_Module__c = trainingModuleId,
13                 Assignment_Date__c = Date.today()
14             );
15
16             insert assignment;
17
18             System.debug('Training assigned successfully to employee: ' + employeeId);
19         } catch (Exception e) {
20             System.debug('Error while assigning training: ' + e.getMessage());
```

Expected Output: Test class runs successfully, passes all assertions, and provides coverage for the TrainingAssignmentHandler class.

Outcome of Phase 5

- Default onboarding tasks are automatically created for new employees using an **Apex Trigger**.
- Employees can be linked to training modules through a reusable **Apex Class**.
- A **Test Class** validates the logic and ensures that the code meets Salesforce deployment requirements with sufficient coverage.