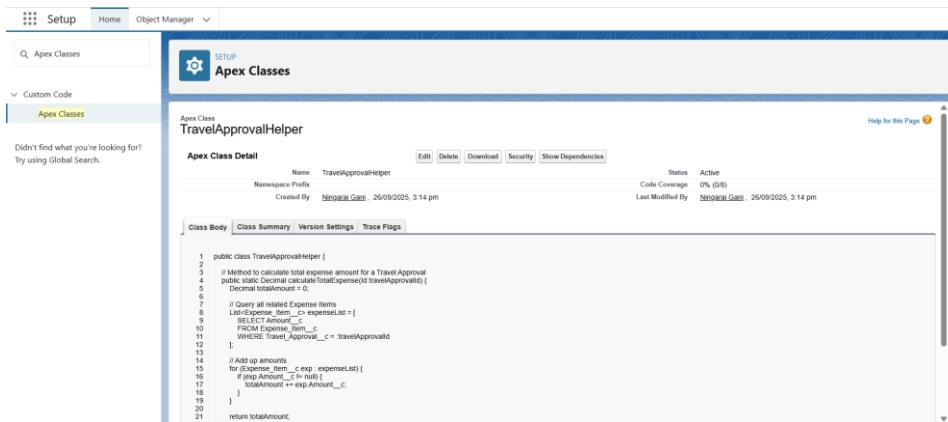


Phase 5: Apex Programming (Developer)

1. Classes & Objects

- **Use Case:** Create a helper class to calculate total travel expenses.
- Apex classes group logic into reusable units.
- Example: TravelApprovalHelper calculates total expense from related expense items.



Code:

```
public class TravelApprovalHelper {  
  
    // Method to calculate total expense amount for a Travel Approval  
    public static Decimal calculateTotalExpense(Id travelApprovalId) {  
        Decimal totalAmount = 0;  
  
        // Query all related Expense Items  
        List<Expense__c> expenseList = [  
            SELECT Amount__c  
            FROM Expense__c  
            WHERE Travel_Approval__c = :travelApprovalId  
        ];  
  
        // Add up amounts  
        for (Expense__c exp : expenseList) {  
            if (exp.Amount__c != null) {  
                totalAmount += exp.Amount__c  
            }  
        }  
    }  
  
    // Add up amounts  
    for (Expense__c exp : expenseList) {  
        if (exp.Amount__c != null) {  
            totalAmount += exp.Amount__c  
        }  
    }  
}
```

```

        totalAmount += exp.Amount__c;
    }

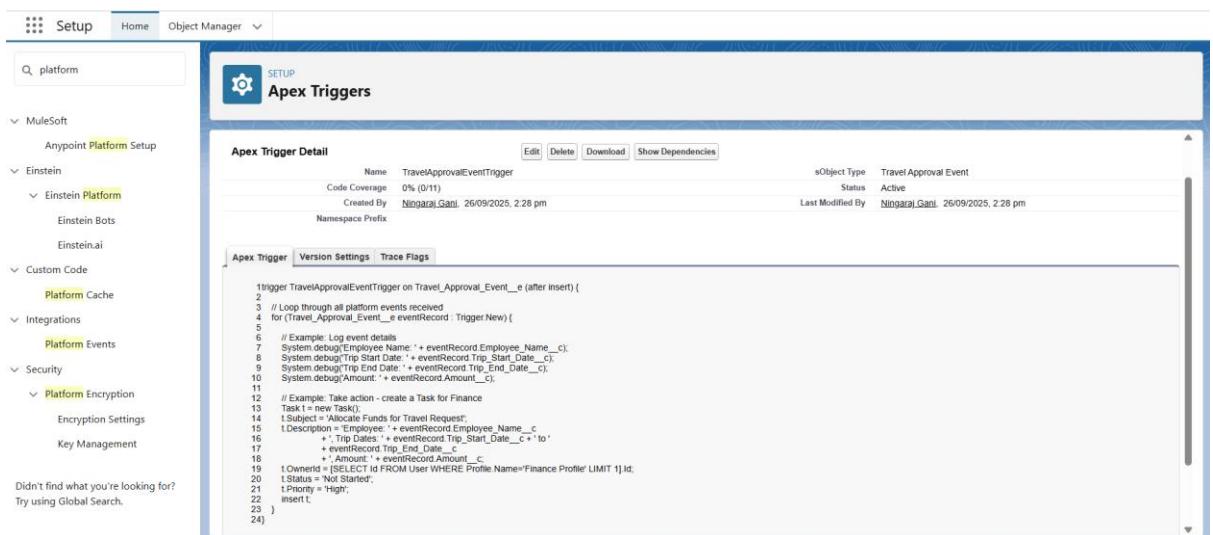
}

return totalAmount;
}

```

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

- Use Case:** Automatically create a Task when a Travel Request is approved.
- Triggers listen to changes in objects and execute logic.
- Example: On Travel_Approval__c, when Status__c = Approved, assign a Task to Finance.



Code:

```

trigger TravelApprovalEventTrigger on Travel_Approval_Event__e (after insert) {

    // Loop through all platform events received
    for (Travel_Approval_Event__e eventRecord : Trigger.New) {

        // Example: Log event details
        System.debug('Employee Name: ' + eventRecord.Employee_Name__c);
        System.debug('Trip Start Date: ' + eventRecord.Trip_Start_Date__c);
        System.debug('Trip End Date: ' + eventRecord.Trip_End_Date__c);
    }
}

```

```

System.debug('Amount: ' + eventRecord.Amount__c);

// Example: Take action - create a Task for Finance
Task t = new Task();
t.Subject = 'Allocate Funds for Travel Request';
t.Description = 'Employee: ' + eventRecord.Employee_Name__c
    + ', Trip Dates: ' + eventRecord.Trip_Start_Date__c + ' to '
    + eventRecord.Trip_End_Date__c
    + ', Amount: ' + eventRecord.Amount__c;
t.OwnerId = [SELECT Id FROM User WHERE Profile.Name='Finance Profile' LIMIT 1].Id;
t.Status = 'Not Started';
t.Priority = 'High';
insert t;
}

```

3. Trigger Design Pattern

- **Use Case:** Avoid writing all logic directly inside triggers.
- Instead, call helper classes from the trigger.
- Improves **readability, reusability, and testing.**

4. SOQL & SOSL

- **SOQL (Salesforce Object Query Language):** Query records.
 - Example: Fetch all Approved Travel Approvals.
- **SOSL (Salesforce Object Search Language):** Search across multiple objects.
 - Example: Search travel requests by employee name or trip purpose.

Code:

```

// Fetch all Approved Travel Approvals
List<Travel_Approval__c> approvedRequests = [
    SELECT Id, Name, Employee_Name__c, Status__c, Total_Expense__c
    FROM Travel_Approval__c
    WHERE Status__c = 'Approved'
];

```

```

for (Travel_Approval__c ta : approvedRequests) {
    System.debug('✓ Approved Travel: ' + ta.Name + ' | Employee: ' + ta.Employee_Name__c + ' | Expense: ' + ta.Total_Expense__c);
}

```

5. Collections: List, Set, Map

- **List:** Ordered collection (e.g., list of Travel Approvals).
- **Set:** Unique values (e.g., list of unique employee IDs requesting travel).
- **Map:** Key-value pairs (e.g., Map<EmployeeId, TravelApproval>).

6. Control Statements

- **Use Case:** Decision making in code.
- Example:
 - If Status = "Submitted", send for approval.
 - Else if Status = "Approved", notify Finance.

Code:

```

public class TravelApprovalHelper {

    // Method to categorize travel request by expense amount
    public static String getBudgetCategory(Decimal totalExpense) {
        String category;

        // if-else control
        if (totalExpense > 5000) {
            category = 'High Budget';
        } else if (totalExpense >= 1000 && totalExpense <= 5000) {
            category = 'Medium Budget';
        } else {
            category = 'Low Budget';
        }

        return category;
    }
}

```

```
// for loop example - display employee names in list

public static void printEmployeeNames(List<Travel_Approval__c> requests) {
    for (Travel_Approval__c ta : requests) {
        System.debug('Employee Name: ' + ta.Employee_Name__c);
    }
}
```

7. Exception Handling

- **Use Case:** Prevent app crashes and log errors.
- Example: If Expense API call fails, catch the exception and notify Admin.
- Syntax: try { ... } catch(Exception e) { ... } finally { ... }.

Code:

```
public class TravelApprovalHandler {

    public static void updateApprovalStatus(Id recordId, String newStatus) {
        try {
            // Query the record
            Travel_Approval__c ta = [SELECT Id, Status__c FROM Travel_Approval__c WHERE Id = :recordId
LIMIT 1];

            // Update status
            ta.Status__c = newStatus;
            update ta;

            System.debug(' ✅ Travel Approval updated successfully: ' + recordId);
        }
        catch (QueryException qe) {
            System.debug(' ⚠️ Query failed: ' + qe.getMessage());
        }
        catch (DmlException de) {
            System.debug(' ⚠️ DML failed: ' + de.getMessage());
        }
    }
}
```

```

        catch (Exception e) {
            System.debug('⚠️ Unexpected Error: ' + e.getMessage());
        }
    finally {
        System.debug('Process finished (success or failure).');
    }
}

```

8. Test Classes

- **Use Case:** Every Apex class/trigger must have test coverage.
- Example: Create TravelApprovalTest to test approval trigger logic.
- Ensures code works correctly before deployment.

Code:

```

@isTest
public class TravelApprovalHandlerTest {

    @isTest
    static void testUpdateApprovalStatus() {
        // Step 1: Create test Travel Approval record
        Travel_Approval__c ta = new Travel_Approval__c(
            Name = 'Test Trip',
            Status__c = 'Submitted'
        );
        insert ta;

        // Step 2: Call the handler method
        Test.startTest();
        TravelApprovalHandler.updateApprovalStatus(ta.Id, 'Approved');
        Test.stopTest();

        // Step 3: Query the updated record
    }
}

```

```
Travel_Approval__c updatedTa = [SELECT Id, Status__c FROM Travel_Approval__c WHERE Id = :ta.Id  
LIMIT 1];  
  
// Step 4: Assert the expected result  
System.assertEquals('Approved', updatedTa.Status__c, 'Status should be updated to Approved');  
}  
}
```