Apex Programming (Developer)

In this phase of the **Expense Tracker** project, I focused on integrating Salesforce custom objects with Apex logic and ensuring smooth deployment of all components. This phase connected the data model and automation to a working, code-driven process.

1. Creating a New SFDX Project

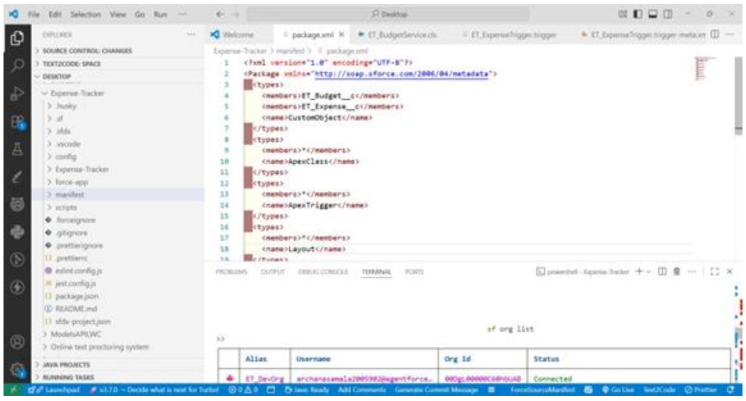
I started by setting up a Salesforce DX project and authorizing my Developer Org:

sfdx force:project:create -n Expense-Tracker

cd Expense-Tracker

sfdx auth:web:login -a ET_DevOrg

This created the project structure and linked it to my Dev Org using a short alias.



```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
<types>
    <members>ET_Budget__c</members>
    <members>ET_Expense__c</members>
    <name>CustomObject</name>
</types>
<types>
    <members>*</members>
    <name>ApexClass</name>
</types>
<types>
    <members>*</members>
    <name>ApexTrigger</name>
</types>
<types>
    <members>*</members>
    <name>Layout</name>
</types>
```

```
sf org list

| Alias     | Username                              | Org Id                 | Status    |
| ET_DevOrg | archanasamala2005982@agentforce…     | [illegible]            | Connected |
```

2. Verification of Objects and Fields

Before coding, I verified that all required custom objects and fields existed:

Objects:

- ET_Budget__c
- ET_Expense__c

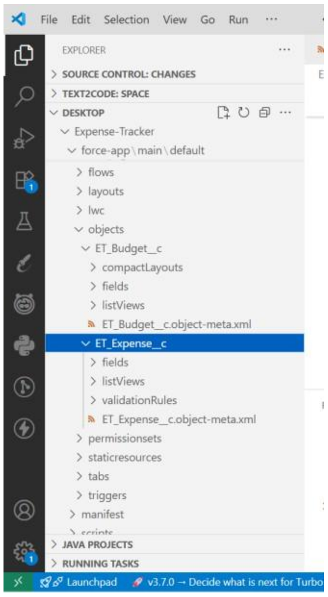
ET_Budget__c Fields:

- Threshold_Amount__c
- Total_Expenses_Apex__c

ET_Expense__c Fields:

- Amount__c
- Expense_Date__c
- Category__c
- Lookup relationship Budget__c pointing to ET_Budget__c

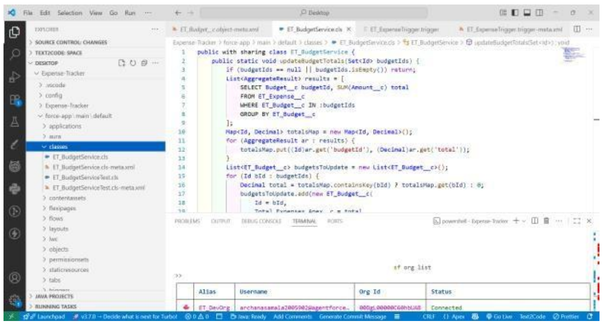
This ensured that all objects and fields were ready for Apex integration.



3. Development of Apex Class

I created a service class **ET_BudgetService.cls** to maintain accurate total expenses for each budget:

- The class calculates the sum of related expenses whenever a record is inserted, updated, or deleted.
- This ensures that the budget always reflects the current total expenses.



```
public with sharing class ET_BudgetService {
    public static void updateBudgetTotals(Set<Id> budgetIds) {
        if (budgetIds == null || budgetIds.isEmpty()) return;
        List<AggregateResult> results = [
            SELECT Budget__c budgetId, SUM(Amount__c) total
            FROM ET_Expense__c
            WHERE ET_Budget__c IN :budgetIds
            GROUP BY ET_Budget__c
        ];
        Map<Id, Decimal> totalsMap = new Map<Id, Decimal>();
        for (AggregateResult ar : results) {
            totalsMap.put((Id)ar.get('budgetId'), (Decimal)ar.get('total'));
        }
        List<ET_Budget__c> budgetsToUpdate = new List<ET_Budget__c>();
        for (Id bId : budgetIds) {
            Decimal total = totalsMap.containsKey(bId) ? totalsMap.get(bId) : 0;
            budgetsToUpdate.add(new ET_Budget__c(
                Id = bId,
                Total_Expenses_Apex__c = total
```

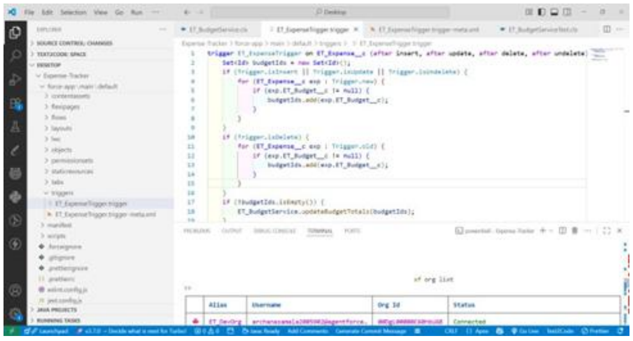
```
sf org list

   Alias      Username                            Org Id               Status
   ET_DevOrg  archanasamala2005902@agentforce.    00DgL00000C60hbUAB   Connected
```

4. Creation of Trigger

A trigger **ET_ExpenseTrigger.trigger** was created on the ET_Expense__c object:

- The trigger connects ET Expense records to ET Budget, ensuring that any changes in expenses are reflected in the budget totals.



```
trigger ET_ExpenseTrigger on ET_Expense__c (after insert, after update, after delete, after undelete)
    Set<Id> budgetIds = new Set<Id>();
    if (Trigger.isInsert || Trigger.isUpdate || Trigger.isUndelete) {
        for (ET_Expense__c exp : Trigger.new) {
            if (exp.ET_Budget__c != null) {
                budgetIds.add(exp.ET_Budget__c);
            }
        }
    }
    if (Trigger.isDelete) {
        for (ET_Expense__c exp : Trigger.old) {
            if (exp.ET_Budget__c != null) {
                budgetIds.add(exp.ET_Budget__c);
            }
        }
    }
    if (!budgetIds.isEmpty()) {
        ET_BudgetService.updateBudgetTotals(budgetIds);
```

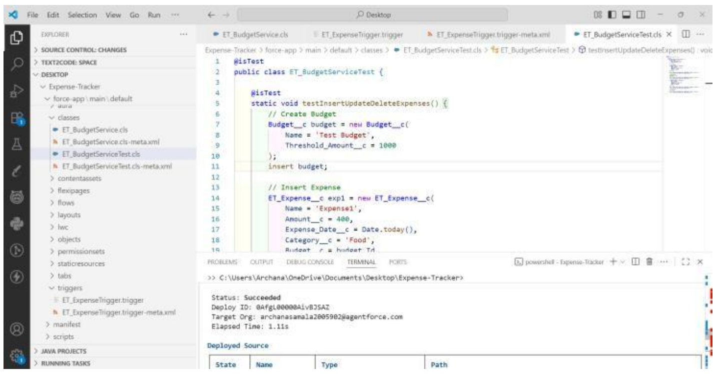
```
sf org list

   Alias      Username                            Org Id               Status
   ET_DevOrg  archanasamala2005902@agentforce.    [illegible]          Connected
```

5. Writing Test Class

I developed a test class **ET_BudgetServiceTest.cls** to validate functionality and achieve code coverage:

- Created test data for budgets and expenses.
- Tested scenarios including inserting, updating, deleting, and adding multiple expenses.
- Used SOQL queries and System.assertEquals() to verify that the budget totals were updated correctly.
- Ensured proper code coverage and verified trigger and service class logic.



```
@isTest
public class ET_BudgetServiceTest {

    @isTest
    static void testInsertUpdateDeleteExpenses() {
        // Create Budget
        Budget__c budget = new Budget__c(
            Name = 'Test Budget',
            Threshold_Amount__c = 1000
        );
        insert budget;

        // Insert Expense
        ET_Expense__c exp1 = new ET_Expense__c(
            Name = 'Expense1',
            Amount__c = 400,
            Expense_Date__c = Date.today(),
            Category__c = 'Food',
            Budget__c = budget.Id
```

```
>> C:\Users\Archana\OneDrive\Documents\Desktop\Expense-Tracker>

Status: Succeeded
Deploy ID: 0AfgL00000AivBJSAZ
Target Org: archanasamala2005902@agentforce.com
Elapsed Time: 1.11s

Deployed Source
| State | Name | Type | Path |
```

6. Deployment Process

Deployment was performed using the Salesforce CLI:

Steps followed:

1. Deploy custom objects: ET_Budget__c and ET_Expense__c
2. Deploy Apex classes: ET_BudgetService and ET_BudgetServiceTest
3. Deploy triggers: ET_ExpenseTrigger
4. Run Apex tests to confirm successful execution and coverage

Deployment Commands Used:

```
sf project deploy start --metadata CustomObject:ET_Budget__c --target-org ET_DevOrg

sf project deploy start --metadata CustomObject:ET_Expense__c --target-org ET_DevOrg

sf project deploy start --source-dir force-app/main/default/classes --target-org ET_DevOrg

sf project deploy start --source-dir force-app/main/default/triggers --target-org ET_DevOrg
```

7. Outcome of Phase 5

By the end of this phase, I was able to:

- Successfully deploy custom objects, Apex classes, and triggers into Salesforce.
- Ensure that budgets reflect total expenses in real-time.

- Gain hands-on experience in debugging deployment errors and resolving metadata issues.

This phase was a key milestone, connecting all previous setups into a fully functional, automated process in Salesforce.