Internal Job Application Management System CRM

Prepared by : Gollapalli.Lakshmi Tulasi Jyothi Sri

Phase 5: Apex Programming (Developer)

Batch : 4

1.Overview

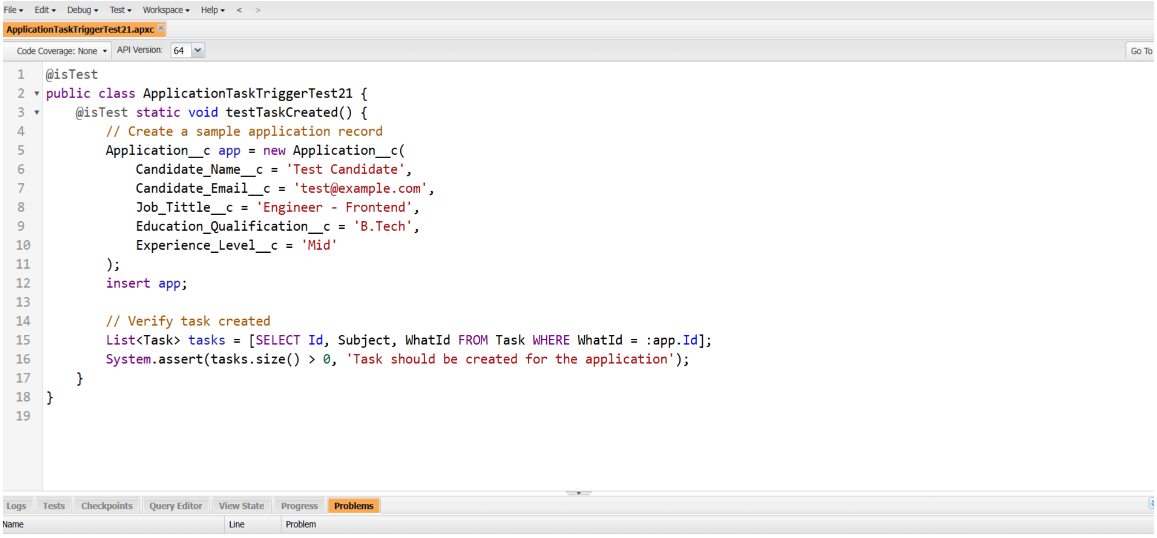
Phase 5 introduced developer-side enhancements using Apex Triggers and Test Classes. These automations added intelligence, such as creating follow-up tasks and ensuring data formatting. The test classes ensured deployment readiness and code coverage.

2. Trigger: Auto Task Creation

Developer Console → New Apex Trigger.

Name = ApplicationTaskTriggerTest21

sObject = Application__c.



```
@isTest
public class ApplicationTaskTriggerTest21 {
    @isTest static void testTaskCreated() {
        // Create a sample application record
        Application__c app = new Application__c(
            Candidate_Name__c = 'Test Candidate',
            Candidate_Email__c = 'test@example.com',
            Job_Tittle__c = 'Engineer - Frontend',
            Education_Qualification__c = 'B.Tech',
            Experience_Level__c = 'Mid'
        );
        insert app;

        // Verify task created
        List<Task> tasks = [SELECT Id, Subject, WhatId FROM Task WHERE WhatId = :app.Id];
        System.assert(tasks.size() > 0, 'Task should be created for the application');
    }
}
```

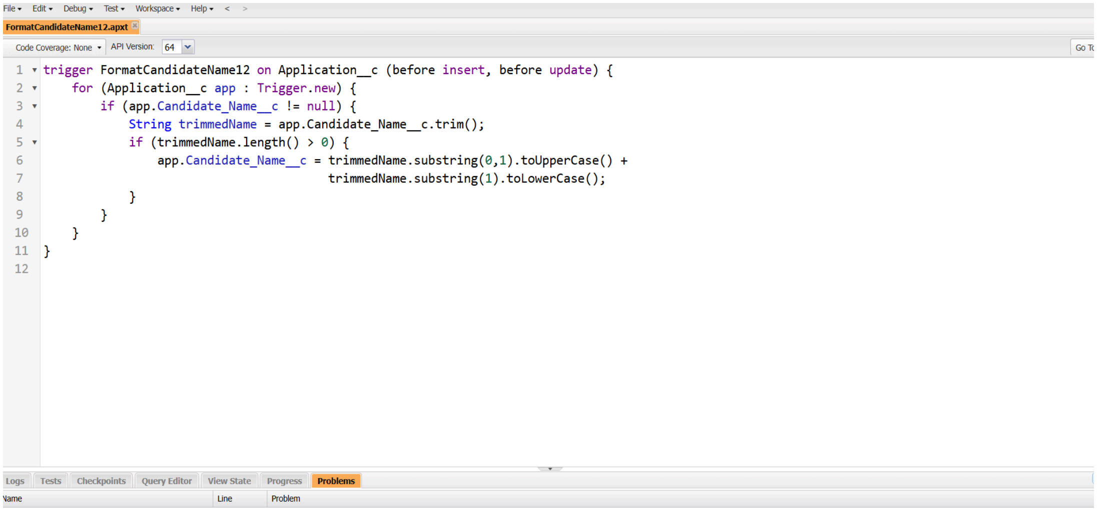
3.Trigger: Auto-Format Candidate Name

In Developer Console → File → New → Apex Trigger.

Name = FormatCandidateName12

sObject = Application__c.

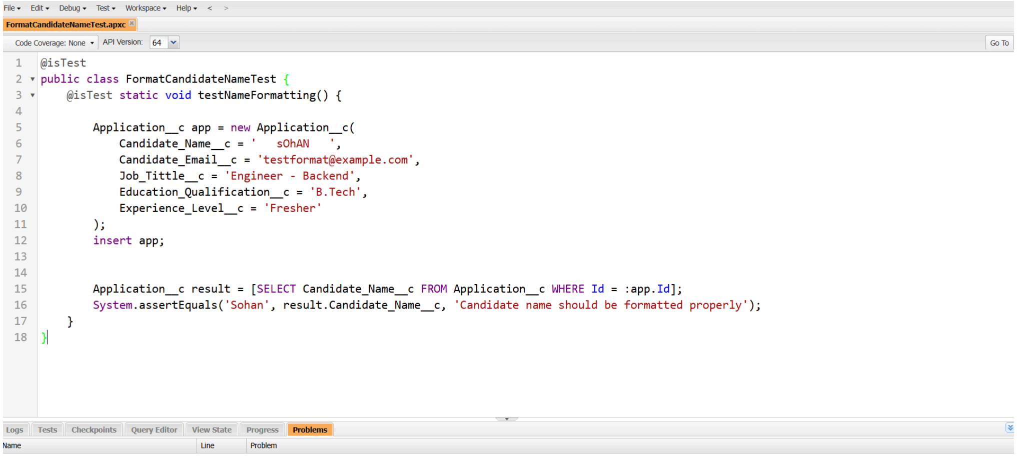
Click Submit.


File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
FormatCandidateName12.apxt
Code Coverage: None ▾ API Version: 64
Go To
Logs Tests Checkpoints Query Editor View State Progress Problems
Name Line Problem

```
trigger FormatCandidateName12 on Application__c (before insert, before update) {
    for (Application__c app : Trigger.new) {
        if (app.Candidate_Name__c != null) {
            String trimmedName = app.Candidate_Name__c.trim();
            if (trimmedName.length() > 0) {
                app.Candidate_Name__c = trimmedName.substring(0,1).toUpperCase() +
                                        trimmedName.substring(1).toLowerCase();
            }
        }
    }
}
```

5.Back in Developer Console → File → New → Apex Class.

Name: FormatCandidateNameTest


File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
FormatCandidateNameTest.apxc
Code Coverage: None ▾ API Version: 64
Go To
Logs Tests Checkpoints Query Editor View State Progress Problems
Name Line Problem

```
@isTest
public class FormatCandidateNameTest {
    @isTest static void testNameFormatting() {

        Application__c app = new Application__c(
            Candidate_Name__c = '    sOhAN    ',
            Candidate_Email__c = 'testformat@example.com',
            Job_Tittle__c = 'Engineer - Backend',
            Education_Qualification__c = 'B.Tech',
            Experience_Level__c = 'Fresher'
        );
        insert app;


        Application__c result = [SELECT Candidate_Name__c FROM Application__c WHERE Id = :app.Id];
        System.assertEquals('Sohan', result.Candidate_Name__c, 'Candidate name should be formatted properly');
    }
}
```

Why

- To provide an easy interface for HR and Job Tracker users.
- Created Job Application Tracker App with tabs for Applications, Reports, Dashboards.
- Centralized navigation for all CRM activities.

Phase 5 Outcome

In this phase, we enhanced the project with developer capabilities.

- Built an Apex Trigger on Application__c → automatically creates a follow-up Task for HR whenever a new application is inserted.

- Added a Before Trigger to auto-format candidate names for data consistency.

- Wrote a Test Class to validate triggers and ensure deployment readiness.