

# Salesforce Project Phase-5

## Phase 5: Data Modeling & Relationships

### Classes & Objects


**Purpose:**

Apex classes are used to implement **complex logic, reusability, and automation** that Flow or standard configuration cannot handle easily. Classes encapsulate code into reusable blocks, while objects represent the data (records) they act upon.

**Class:** InternshipApplicationHandler

**Objects Used:** Internship\_Application\_\_c, Student\_\_c, Internship\_\_c.

**Use in Project:** Whenever a student submits an internship application, we want the **status field** of the application to be automatically set to "Submitted" if the user didn't fill it.

 **Apex Classes**

Apex Class

InternshipApplicationHandler

Help for this Page ?

Apex Class Detail

EditDeleteDownloadSecurityShow Dependencies

Name	InternshipApplicationHandler	Status	Active
Namespace Prefix		Code Coverage	0% (0/4)
Created By	Harshita Shrivastava , 9/25/2025, 5:13 AM	Last Modified By	Harshita Shrivastava , 9/25/2025, 5:13 AM

Class Body


Class SummaryVersion SettingsTrace Flags

```
1 public class InternshipApplicationHandler {
2
3     // Method to set default status
4     public static void setDefaultStatus(List<Internship_Application__c> apps) {
5         for(Internship_Application__c app : apps){
6             if(app.Status__c == null){
7                 app.Status__c = 'Submitted';
8             }
9         }
10    }
11
12 }
```

EditDeleteDownloadSecurityShow Dependencies

**Class:** RecommendationEngine

**Purpose :** Suggest internships to students based on skills or course.

 **Apex Classes**

Apex Class Help for this Page ?

**RecommendationEngine**

**Apex Class Detail** Edit Delete Download Security Show Dependencies

Name	RecommendationEngine	Status	Active
Namespace Prefix		Code Coverage	0% (0/3)
Created By	Harshita Shrivastava , 9/25/2025, 5:46 AM	Last Modified By	Harshita Shrivastava , 9/25/2025, 5:46 AM

Class Body Class Summary Version Settings Trace Flags

```
1 public class RecommendationEngine {
2     public static List<Internship__c> getRecommendedInternships(String skill){
3         return [SELECT Id, Name, Company__c, Skills_Required__c
4                 FROM Internship__c
5                 WHERE Skills_Required__c = :skill
6                 LIMIT 5];
7     }
8 }
```

Edit Delete Download Security Show Dependencies

## Apex Triggers

### Definition:

Triggers automate logic before or after record events (insert, update, delete). They allow actions to run automatically when records change.

### Trigger Design Pattern:

- The trigger **calls a Handler Class**.
- Keeps logic **centralized and maintainable**.
- Supports **bulkification** (process multiple records efficiently)
- • **Before triggers** are used to update fields **before saving** the record.
- • **After triggers** are used to perform actions on **related objects** or external processes after the record is saved.



SETUP

## Apex Triggers

Apex Trigger

### InternshipApplicationTrigger

[Help for this Page](#) ?

#### Apex Trigger Detail

[Edit](#) [Delete](#) [Download](#) [Show Dependencies](#)

Name	InternshipApplicationTrigger	sObject Type	Internship Application
Code Coverage	0% (0/2)	Status	Active
Created By	<a href="#">Harshita Shrivastava</a> , 9/25/2025, 6:43 AM	Last Modified By	<a href="#">Harshita Shrivastava</a> , 9/25/2025, 7:03 AM

Namespace Prefix

Apex Trigger

Version Settings

Trace Flags

```
1 trigger InternshipApplicationTrigger on Internship_Application__c (before insert) {
2     if (Trigger.isBefore && Trigger.isInsert) {
3         // Call the Apex class method to set default status
4         InternshipApplicationHandler.setDefaultStatus(Trigger.new);
5     }
6 }
```

[Edit](#) [Delete](#) [Download](#) [Show Dependencies](#)

### 3. SOQL & SOSL

SOQL (Salesforce Object Query Language)

#### Purpose in your project:

Fetch records from a specific object like Internship\_\_c, Student\_\_c, or Internship\_Application\_\_c for filtering, reporting, or automation.

```
1 public class RecommendationEngine {
2
3     @AuraEnabled(cacheable=true)
4     public static List<Internship__c> getRecommendedInternships(Id studentId) {
5         // Get the student's skills
6         Student__c s = [SELECT Id, Skills__c FROM Student__c WHERE Id = :studentId LIMIT 1];
7
8         // Fetch top 5 internships matching the student's skill
9         List<Internship__c> recommended = [
10             SELECT Id, Name, Company__c, Skills_Required__c
11             FROM Internship__c
12             WHERE Skills_Required__c = :s.Skills__c
13             LIMIT 5
14         ];
15
16         return recommended;
17     }
18 }
```

## SOSL (Salesforce Object Search Language)

### Purpose in your project:

Search text across multiple objects/fields. Useful if a student or admin wants to **quickly search internships or applications by keyword**.

? **Use SOQL** when you know the object and fields you want (e.g., filter internships by skill).

? **Use SOSL** when you want a broader keyword search across multiple objects (e.g., student searching for internships, or admin searching for students & applications).

## Batch Apex

? **Fetch all Internship Application records with Status\_\_c = 'Submitted'.**

? **Update their status to Reviewed.**

Apex Class

### ReviewApplicationBatch

#### Apex Class Detail

Edit Delete Download Security Show Dependencies

Name ReviewApplicationBatch

Namespace Prefix

Created By Harshita Shrivastava , 9/25/2025, 10:04 AM

Class Body

Class Summary

Version Settings

Trace Flags

```
1 global class ReviewApplicationBatch implements Database.Batchable<SObject>, Database.Stateful {
2
3     // Step 1: Start - fetch all submitted applications
4     global Database.QueryLocator start(Database.BatchableContext BC) {
5         return Database.getQueryLocator([
6             SELECT Id, Status__c, Student__c, Internship__c
7             FROM Internship_Application__c
8             WHERE Status__c = 'Submitted'
9         ]);
10    }
11
12    // Step 2: Execute - process each chunk
13    global void execute(Database.BatchableContext BC, List<Internship_Application__c> scope) {
14        List<Internship_Application__c> appsToUpdate = new List<Internship_Application__c>();
15
16        for(Internship_Application__c app : scope) {
17            app.Status__c = 'Reviewed';
18            appsToUpdate.add(app);
19        }
20
21        if(!appsToUpdate.isEmpty()) {
22            update appsToUpdate; // Save changes
23        }
24    }
25
26    // Step 3: Finish - optional post-processing
27    global void finish(Database.BatchableContext BC) {
28        // Optional: send email or system notification to Placement Officers
29        System.debug('Batch completed: Internship Applications status updated to Reviewed.');
```

# Tesst Classes

Purpose: Validate logic, ensure code coverage  $\geq 75\%$

Recommendation Engine

Test Other Test Classes: Review Application Batch Test(same format)

Apex Class

## RecommendationEngineTest

### Apex Class Detail

[Edit](#)[Delete](#)[Download](#)[Run Test](#)[Show Dependencies](#)

Name RecommendationEngineTest

Namespace Prefix

Last Modified By [Harshita Shrivastava](#), 9/25/2025, 9:37 AM

[Class Body](#)[Class Summary](#)[Version Settings](#)[Trace Flags](#)

```
1  @isTest
2  public class RecommendationEngineTest {
3      @isTest
4      static void testGetRecommendedInternships() {
5          // Create Student with all required fields
6          Student__c s = new Student__c(
7              Name = 'Test User',           // Single Name field
8              Skills__c = 'Java',
9              Email__c = 'test@student.com', // REQUIRED
10             CGPA_GPA__c = 8.5             // REQUIRED
11         );
12         insert s;
13
14         // Create Internship with required fields
15         Internship__c i = new Internship__c(
16             Name = 'Java Internship',
17             Skills_Required__c = 'Java'
18         );
19         insert i;
20
21         // Call RecommendationEngine
22         List<Internship__c> recs = RecommendationEngine.getRecommendedInternships(s.Id);
23
24         // Assert
25         System.assert(!recs.isEmpty(), 'Internships should be recommended');
26     }
27 }
```

## 9. Best Practices

- Always bulkify triggers
- Avoid SOQL/DML inside loops
- Use Handler classes for logic
- Write robust test classes with positive/negative scenarios