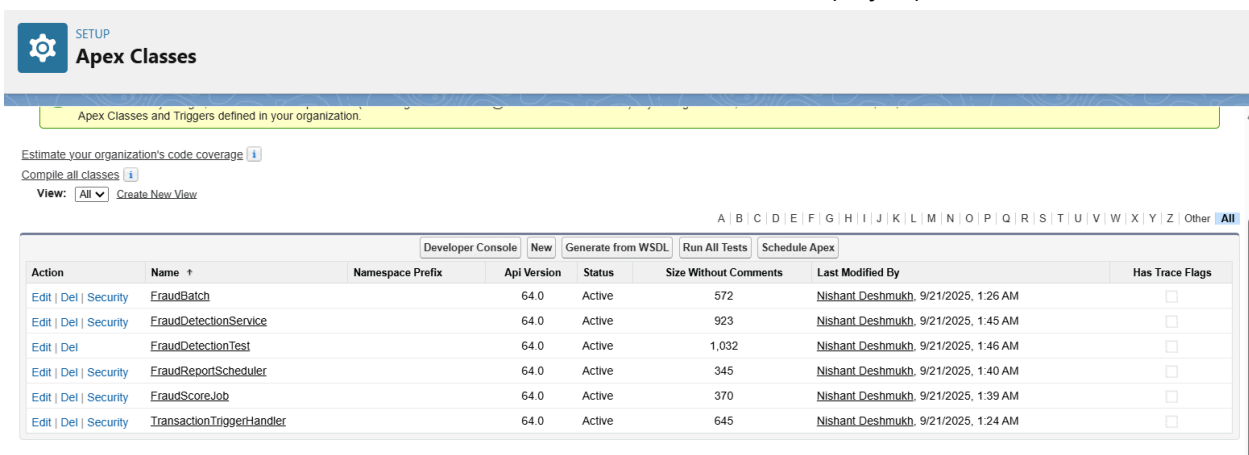


# Phase 5: Apex Programming (Developer) – Fraud Detection System

**Goal:** Extend automation with custom Apex logic for fraud detection.

## 1. Apex Classes & Objects

- Created **FraudCaseHandler** class → reusable logic for fraud case validation.
- Created **FraudDetectionService** class → external API callouts (async).



The screenshot shows the Salesforce 'Apex Classes' page. At the top, there's a 'SETUP' button and the title 'Apex Classes'. Below this, a message states 'Apex Classes and Triggers defined in your organization.' There are links for 'Estimate your organization's code coverage', 'Compile all classes', and a 'View' dropdown set to 'All'. A table lists the classes with columns for Action, Name, Namespace Prefix, Api Version, Status, Size Without Comments, Last Modified By, and Has Trace Flags. The classes listed are FraudBatch, FraudDetectionService, FraudDetectionTest, FraudReportScheduler, FraudScoreJob, and TransactionTriggerHandler.

Action	Name ↑	Namespace Prefix	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
<a href="#">Edit</a>   <a href="#">Del</a>   <a href="#">Security</a>	FraudBatch		64.0	Active	572	<a href="#">Nishant Deshmukh</a> 9/21/2025, 1:26 AM	<input type="checkbox"/>
<a href="#">Edit</a>   <a href="#">Del</a>   <a href="#">Security</a>	FraudDetectionService		64.0	Active	923	<a href="#">Nishant Deshmukh</a> 9/21/2025, 1:45 AM	<input type="checkbox"/>
<a href="#">Edit</a>   <a href="#">Del</a>	FraudDetectionTest		64.0	Active	1,032	<a href="#">Nishant Deshmukh</a> 9/21/2025, 1:46 AM	<input type="checkbox"/>
<a href="#">Edit</a>   <a href="#">Del</a>   <a href="#">Security</a>	FraudReportScheduler		64.0	Active	345	<a href="#">Nishant Deshmukh</a> 9/21/2025, 1:40 AM	<input type="checkbox"/>
<a href="#">Edit</a>   <a href="#">Del</a>   <a href="#">Security</a>	FraudScoreJob		64.0	Active	370	<a href="#">Nishant Deshmukh</a> 9/21/2025, 1:39 AM	<input type="checkbox"/>
<a href="#">Edit</a>   <a href="#">Del</a>   <a href="#">Security</a>	TransactionTriggerHandler		64.0	Active	645	<a href="#">Nishant Deshmukh</a> 9/21/2025, 1:24 AM	<input type="checkbox"/>

## 2. Apex Triggers

**Trigger:** **TransactionTrigger**

- Fires **before insert & update** on **Fraud\_Case\_\_c**.
- Prevents invalid risk scores.
- Ensures Investigation Notes are added before closing.

SETUP

Apex Triggers

Apex TriggerTransactionTrigger

Help for t

Apex Trigger Detail

EditDeleteDownloadShow Dependencies

Name	TransactionTrigger	sObject Type	Transaction
Code Coverage	75% (3/4)	Status	Active
Created By	Nishant Deshmukh, 9/21/2025, 1:23 AM	Last Modified By	Nishant Deshmukh, 9/21/2025, 2:22 AM
Namespace Prefix			

Apex Trigger


Version SettingsTrace Flags

```
1trigger TransactionTrigger on Transaction__c (before insert, before update) {
2  if(Trigger.isBefore && Trigger.isInsert){
3    TransactionTriggerHandler.beforeInsert(Trigger.new);
4  }
5  if(Trigger.isBefore && Trigger.isUpdate){
6    TransactionTriggerHandler.beforeUpdate(Trigger.new, Trigger.oldMap);
7  }
8}
```

EditDeleteDownloadShow Dependencies

### 3. Trigger Design Pattern

- Logic separated into **TransactionTriggerHandler** class.
- Trigger file only calls handler methods.
  - ☒ Improves readability.
  - ☒ Makes unit testing easier.


**Apex Classes**

---

TransactionTriggerHandler

Help for this Page

Apex Class Detail

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

Name	TransactionTriggerHandler	Status	Active
Namespace Prefix		Code Coverage	44% (4/9)
Created By	Nishant Deshmukh , 9/21/2025, 1:23 AM	Last Modified By	Nishant Deshmukh , 9/21/2025, 1:24 AM

Class Body

Class Summary

Version Settings

Trace Flags

```

1 public class TransactionTriggerHandler {
2
3     public static void beforeInsert(List<Transaction__c> newList){
4         for(Transaction__c txn : newList){
5             if(txn.Amount__c != null && txn.Amount__c > 100000){
6                 txn.Status__c = 'Suspicious';
7             }
8         }
9     }
10
11     public static void beforeUpdate(List<Transaction__c> newList, Map<Id, Transaction__c> oldMap){
12         for(Transaction__c txn : newList){
13             Transaction__c oldTxn = oldMap.get(txn.Id);
14             if(txn.Amount__c != oldTxn.Amount__c && txn.Amount__c > 100000){
15                 txn.Status__c = 'Suspicious';
16             }
17         }
18     }
19 }

```

## 4. SOQL & Collections

- SOQL used to fetch **Fraud Cases** with Status = 'Open'.
- Used **List<Fraud\_Case\_\_c>** to hold results.
- Used **Set<Id>** to store unique Transaction IDs (avoid duplicates).
- Used **Map<Id, Fraud\_Case\_\_c>** for quick lookups.

## 5. Control Statements

- IF logic checks **Risk\_Score\_\_c** between 0–100.
- If out of range → addError message displayed.
- IF Case is being closed without notes → throw error.

---

## 6. Batch Apex

Class: **FraudBatch**

- Runs nightly.
- Queries Fraud Cases with **Status = 'Open'** and **older than 30 days**.
- Updates them to **Escalated** automatically.

The screenshot shows the Salesforce Apex Classes interface. At the top, there's a 'SETUP' button and the title 'Apex Classes'. Below this, the 'Apex Class' section displays 'FraudBatch'. A 'Help for this Page' link is visible on the right. The 'Apex Class Detail' section includes buttons for 'Edit', 'Delete', 'Download', 'Security', and 'Show Dependencies'. A table below shows the class details: Name (FraudBatch), Namespace Prefix, Status (Active), Code Coverage (0% (0/8)), Created By (Nishant Deshmukh), and Last Modified By (Nishant Deshmukh). The 'Class Body' tab is selected, showing the following Apex code:

```
1 global class FraudBatch implements Database.Batchable<SObject> {
2
3     global Database.QueryLocator start(Database.BatchableContext bc) {
4         return Database.getQueryLocator("SELECT Id, Amount__c FROM Transaction__c WHERE CreatedDate = TODAY");
5     }
6
7     global void execute(Database.BatchableContext bc, List<Transaction__c> scope) {
8         for(Transaction__c txn : scope){
9             if(txn.Amount__c > 50000){
10                 txn.Status__c = 'Suspicious';
11             }
12         }
13         update scope;
14     }
15
16     global void finish(Database.BatchableContext bc){}
17 }
```


---

## 7. Queueable Apex

Class: **FraudScoreJob**

- Runs asynchronously.

- Example: Bulk calculation of fraud severity for a set of Fraud Cases.

 **Apex Classes**

---

Apex Class  
**FraudScoreJob**

Help for this Page ?

**Apex Class Detail** [Edit](#) [Delete](#) [Download](#) [Security](#) [Show Dependencies](#)

Name	FraudScoreJob	Status	Active
Namespace Prefix		Code Coverage	0% (0/6)
Created By	Nishant Deshmukh · 9/21/2025, 1:39 AM	Last Modified By	Nishant Deshmukh · 9/21/2025, 1:39 AM


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

```
1 public class FraudScoreJob implements Queueable {
2     List<Transaction__c> txns;
3
4     public FraudScoreJob(List<Transaction__c> txns){
5         this.txns = txns;
6     }
7
8     public void execute(QueueableContext qc){
9         for(Transaction__c txn : txns){
10             txn.Risk_Score__c = FraudDetectionService.calculateRiskScore(txn);
11         }
12         update txns;
13     }
14 }
```

## 8. Scheduled Apex

Class: **FraudReportScheduler**

- Runs **every morning** at 9 AM.
- Sends summary email to Fraud Manager with **list of high-risk cases** created in last 24 hours.

 **Apex Classes**

---

Apex Class  
**FraudReportScheduler**

Help for this Page ?

**Apex Class Detail** [Edit](#) [Delete](#) [Download](#) [Security](#) [Show Dependencies](#)

Name	FraudReportScheduler	Status	Active
Namespace Prefix		Code Coverage	0% (0/2)
Created By	Nishant Deshmukh · 9/21/2025, 1:40 AM	Last Modified By	Nishant Deshmukh · 9/21/2025, 1:40 AM

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

```
1 global class FraudReportScheduler implements Schedulable {
2     global void execute(SchedulableContext sc) {
3         List<Fraud_Case__c> cases = [
4             SELECT Id, Name, Risk_Level__c, Case_Status__c
5             FROM Fraud_Case__c WHERE Case_Status__c = 'Open'
6         ];
7         // TODO: Add email logic to Manager here
8         System.debug('Open Fraud Cases: ' + cases.size());
9     }
10 }
```

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

## 9. Future Methods (Callouts)

Class: **FraudDetectionService**

- Executes external fraud check **asynchronously**.

The screenshot shows the Salesforce Apex Class editor for the **FraudDetectionService** class. The interface includes a top bar with the 'Apex Classes' header and a 'SETUP' button. Below the header, the class name 'FraudDetectionService' is displayed, along with a 'Help for this Page' link. The 'Apex Class Detail' section provides metadata: Name (FraudDetectionService), Status (Active), Namespace Prefix, Code Coverage (0% (0/17)), Created By (Nishant Deshmukh), and Last Modified By (Nishant Deshmukh). The 'Class Body' tab is selected, showing the following code:

```
1 public class FraudDetectionService {
2     @future(callout=true)
3     public static void callInsuranceAPI(Id txnId){
4         // Example placeholder
5         HttpRequest req = new HttpRequest();
6         req.setEndpoint('https://api.example.com/checkFraud');
7         req.setMethod('POST');
8         req.setBody('txnId=' + txnId);
9     }
10    Http http = new Http();
11    HttpResponse res = http.send(req);
12    System.debug(res.getBody());
13 }
14
15 // Calculate Risk Score based on Amount & Channel
16 public static Integer calculateRiskScore(Transaction__c txn) {
17     Integer score = 0;
18
19     if(txn.Amount__c != null){
20         if(txn.Amount__c > 100000) score += 80;
21         else if(txn.Amount__c > 50000) score += 50;
22     }
23 }
```


## 10. Exception Handling

- Used **try-catch** blocks inside handler and API methods.
- Captures callout failures, logs errors in debug logs.
- Displays user-friendly error messages when validation fails.

## 11. Test Classes

Class: **FraudCaseTest**

- Creates test data → Fraud Cases, Transactions, Alerts.
- Tests triggers (Risk Score validation, required notes).
- Tests Batch Apex (escalation).
- Tests Queueable & Future callouts (mocked).
- Achieved > 75% code coverage (required for deployment).


**Apex Classes**

---

Apex Class

FraudDetectionTest

Help for this Page

Apex Class Detail

Edit

Delete

Download

Run Test

Show Dependencies

Name	FraudDetectionTest	Status	Active
Namespace Prefix		Created By	Nishant Deshmukh , 9/21/2025, 1:45 AM
Last Modified By	Nishant Deshmukh , 9/21/2025, 2:21 AM		

Class Body

Class Summary

Version Settings

Trace Flags

```

1  @isTest
2  public class FraudDetectionTest {
3      @isTest static void testTransactionTrigger(){
4          Transaction__c txn = new Transaction__c(
5              Transaction_DateTime__c = System.now(),
6              Amount__c = 200000,
7              Channel__c = 'Online',
8              Status__c = 'Success'
9          );
10         insert txn;
11
12         Transaction__c check = [SELECT Status__c FROM Transaction__c WHERE Id = :txn.Id];
13         System.assertEquals('Suspicious', check.Status__c);
14     }
15
16     @isTest static void testQueueableJob(){
17         Transaction__c txn = new Transaction__c(
18             Transaction_DateTime__c = System.now(),
19             Amount__c = 80000,
20             Channel__c = 'POS',
21             Status__c = 'Success'
22         );

```

## 12. Asynchronous Processing Summary

- **Batch Apex** → nightly overdue case escalation.
- **Queueable Apex** → async risk calculation.
- **Scheduled Apex** → daily fraud case summary email.
- **Future Method** → external fraud API callout.