**Phase 7 – Integration & External Access**

**Project:** Smart Internship & Placement Management System  
**Domain:** Salesforce CRM | Education & Recruitment

**Objective**

Phase 7 demonstrates the concept of **external system integration** in Salesforce using **Apex REST API callouts**. The goal is to fetch internship data from an external source and make it available within Salesforce. This phase illustrates how Salesforce can communicate with external services programmatically, enhancing the system’s functionality beyond its internal database.

**Reason for Using Mock API**

* As a beginner-level implementation, we do not have access to a real external internship API.
* The **mock API** simulates external data, allowing demonstration of the integration process without requiring a live endpoint.
* This ensures that:
  + Apex callouts are correctly implemented and tested.
  + Unit tests can be executed successfully using Salesforce testing framework.
  + The concept of **integration and external access** is clear for demonstration purposes.

**Apex Classes and Their Purpose**

**1. InternshipService**

public class InternshipService {

public class Internship {

public String title;

public String company;

public String location;

public Internship(String t, String c, String l) {

this.title = t;

this.company = c;

this.location = l;

}

}

public static List<Internship> fetchInternships() {

HttpRequest req = new HttpRequest();

req.setEndpoint('https://api.example.com/internships');

req.setMethod('GET');

Http http = new Http();

HttpResponse res = http.send(req);

InternshipResponse response = (InternshipResponse) JSON.deserialize(res.getBody(), InternshipResponse.class);

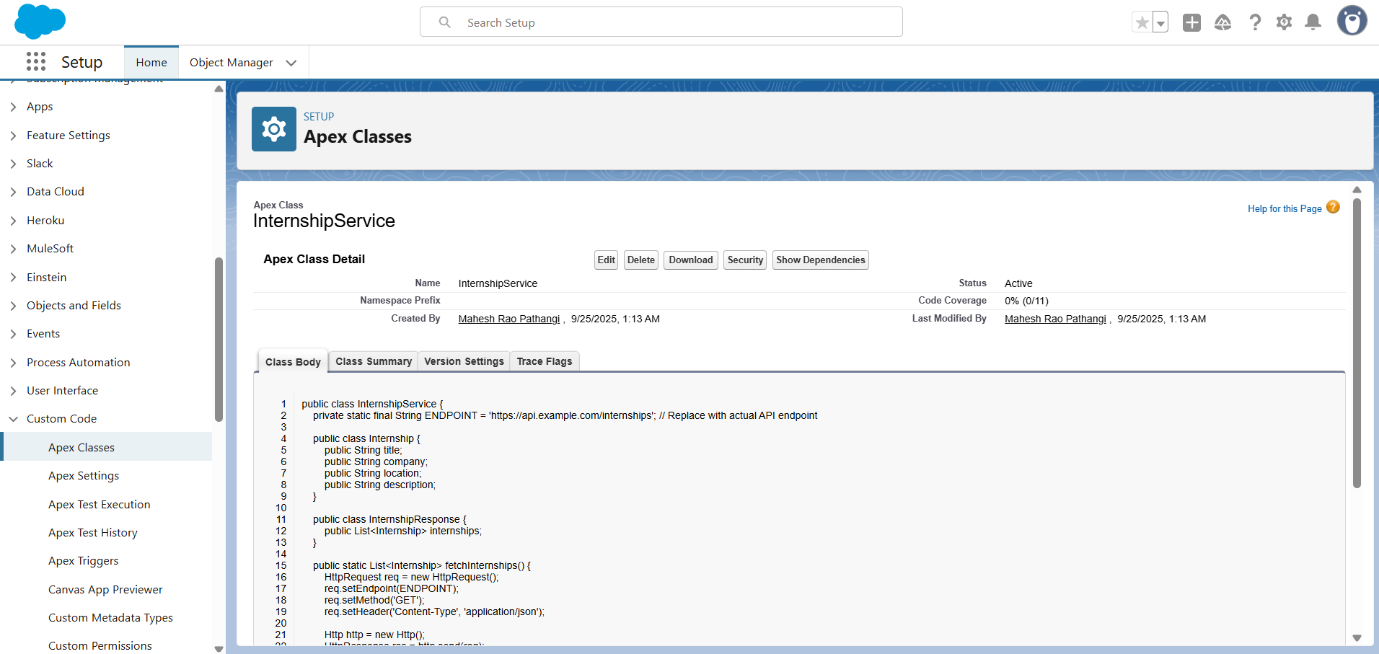
return response.internships;

}

public class InternshipResponse {

public List<Internship> internships;

}

}  
  


**2. InternshipServiceMock**

@IsTest

global class InternshipServiceMock implements HttpCalloutMock {

global HTTPResponse respond(HTTPRequest req) {

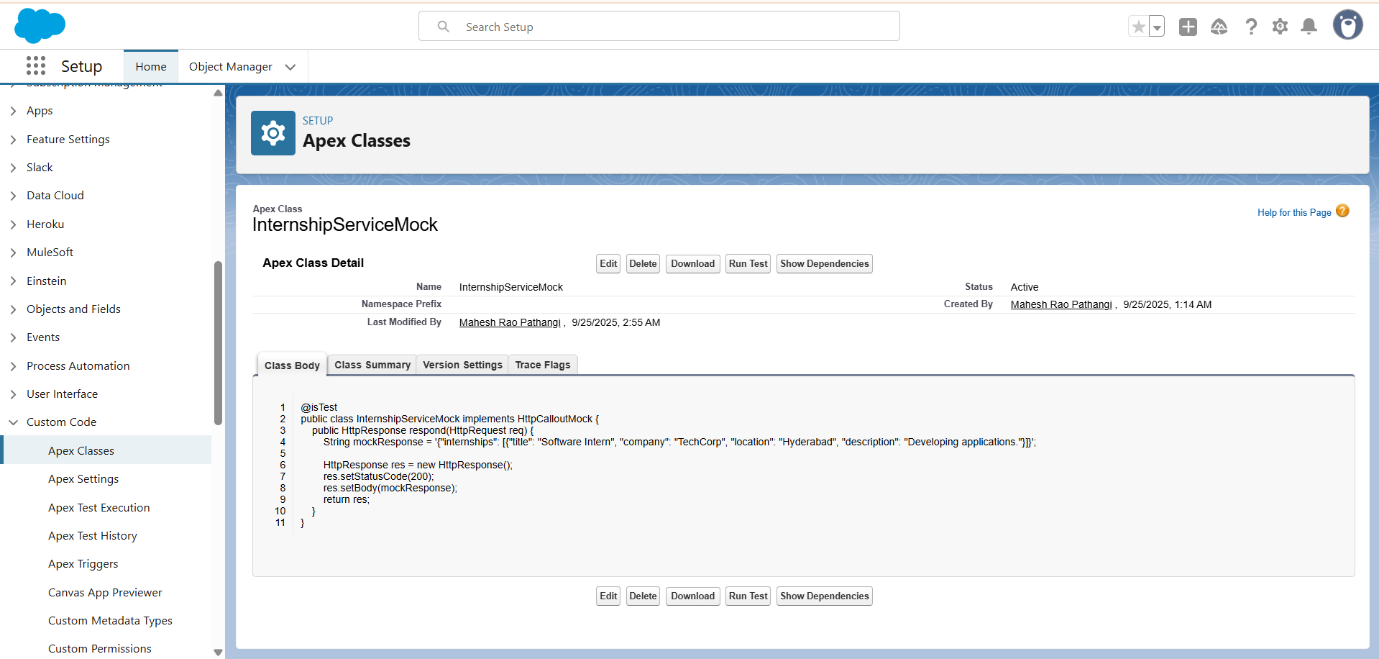
HttpResponse res = new HttpResponse();

res.setStatusCode(200);

res.setBody('{"internships":[{"title":"Software Intern","company":"TechCorp","location":"Hyderabad"},{"title":"Marketing Intern","company":"MarketX","location":"Bangalore"},{"title":"Design Intern","company":"Creatives","location":"Chennai"}]}');

return res;

}

}  
  


**3. InternshipServiceTest**

@IsTest

public class InternshipServiceTest {

@IsTest

static void testFetchInternships() {

Test.setMock(HttpCalloutMock.class, new InternshipServiceMock());

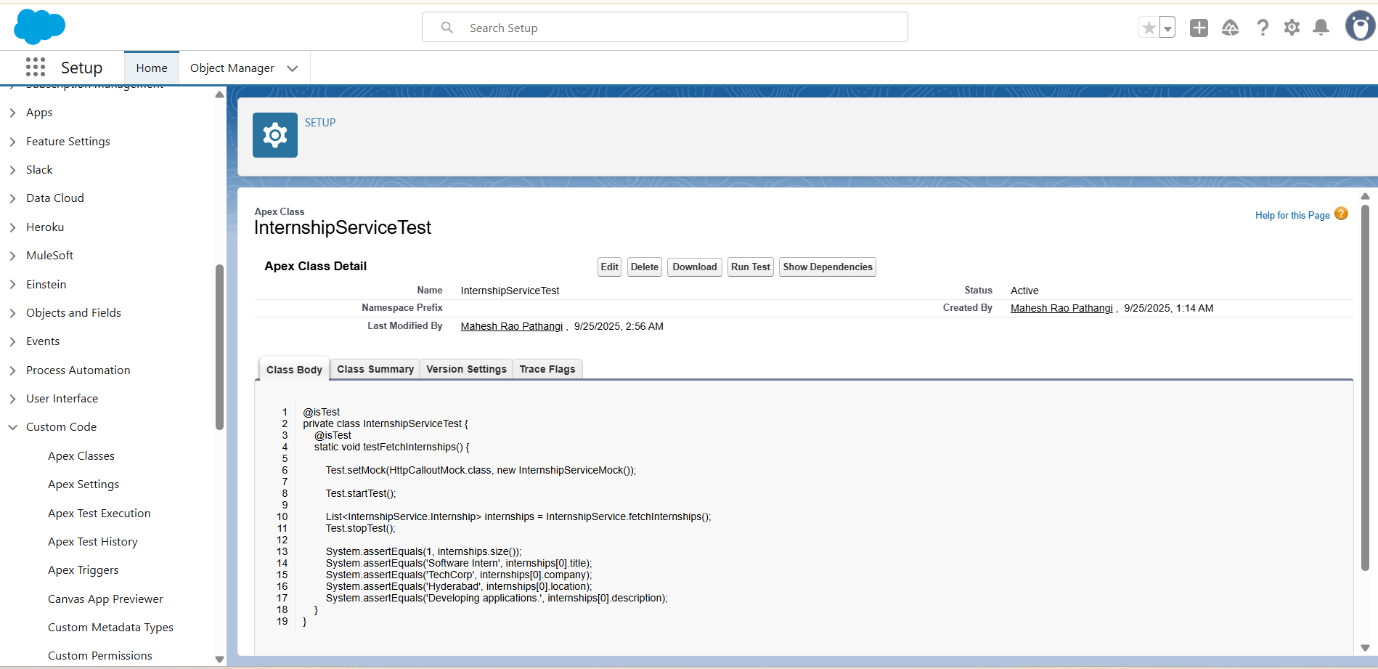
Test.startTest();

List<InternshipService.Internship> internships = InternshipService.fetchInternships();

Test.stopTest();

System.assertEquals(3, internships.size());

}

}  
  


**Procedure For Phase 7**

1. Open InternshipService, InternshipServiceMock, and InternshipServiceTest in **Setup → Apex Classes**.
2. Explain the role of each class and how they interact.
3. Navigate to **Setup → Apex Test Execution**, run InternshipServiceTest, and show that the test **passes** and returns mock internship data.
4. Highlight that InternshipServiceMock simulates external API responses for testing purposes.
5. Optionally explain that the fetched data can be used in a **Flow or Lightning component** for displaying internships to students.