Phase 7: Integration & External Access

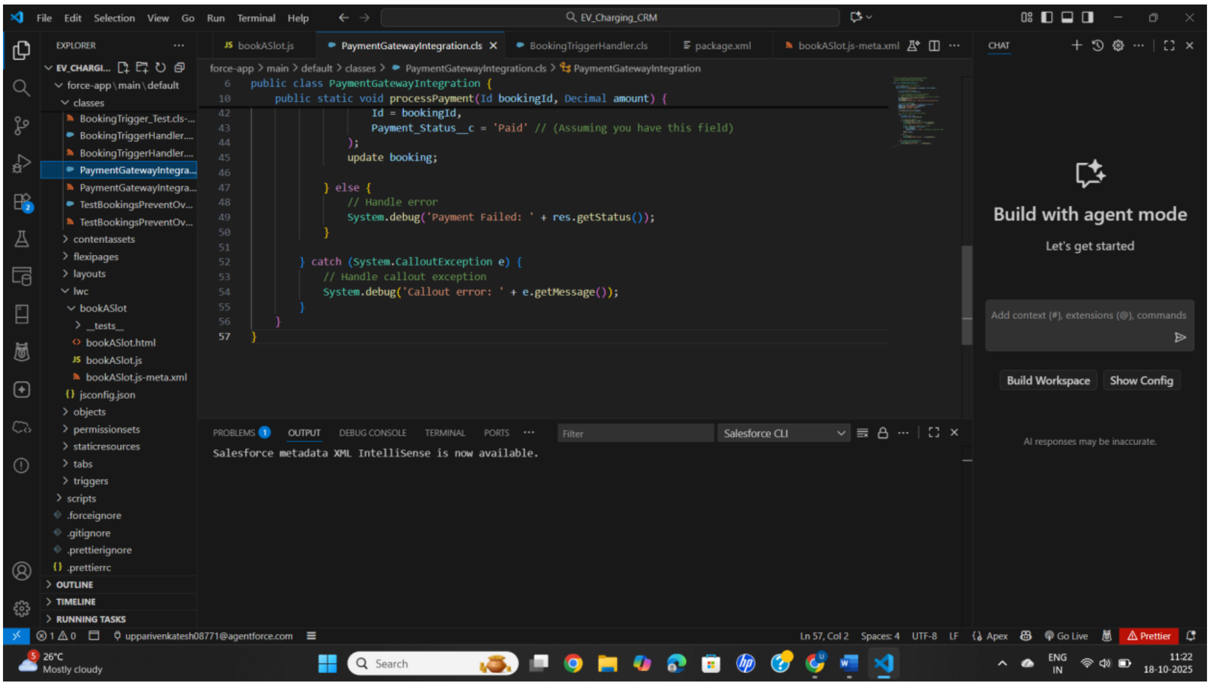
This phase demonstrates how Salesforce can interact with external systems, simulating a callout to a third-party payment gateway when a booking is confirmed.

Configuration for Callouts

To enable Salesforce to communicate with external endpoints, specific configurations are required.

1. **Remote Site Settings:**
    - **Purpose:** The traditional method to whitelist specific domains that Salesforce Apex code is allowed to call out to.
    - **Setup:** A Remote Site named Payment_API_Mock was created, allowing callouts to the URL https://httpbin.org (a free service for testing HTTP requests).
    - **Path:** Setup $\rightarrow$ Security $\rightarrow$ Remote Site Settings $\rightarrow$ New.
2. **Named Credentials (Preferred Method):**
    - **Purpose:** The modern, more secure, and flexible way to define callout endpoints and manage authentication separately from the code.
    - **Setup:** A Named Credential named Payment_API was created.
        - **Label:** Payment API
        - **URL:** https://httpbin.org
        - **Identity Type:** Anonymous (as httpbin.org requires no login)
        - **Authentication Protocol:** No Authentication
    - **Path:** Setup $\rightarrow$ Security $\rightarrow$ Named Credentials $\rightarrow$ New.

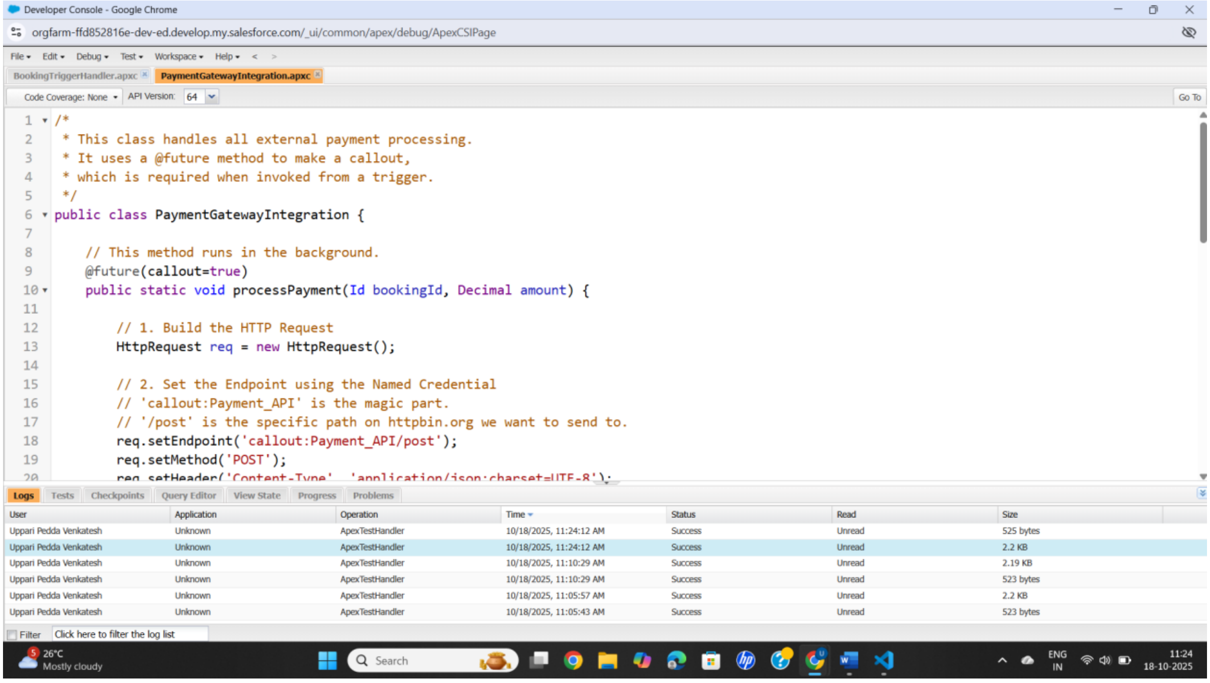
- **Usage in Code:** Apex code references this Named Credential using the special URL format callout:Payment_API/..., allowing the endpoint URL to be managed declaratively.



EV_Charging_CRM
bookASlot.js
PaymentGatewayIntegration.cls
BookingTriggerHandler.cls
package.xml
bookASlot.js-meta.xml
force-app > main > default > classes > PaymentGatewayIntegration.cls > PaymentGatewayIntegration
EXPLORER
CHAT
Build with agent mode
Let's get started
Build Workspace
Show Config
AI responses may be inaccurate.

```
public class PaymentGatewayIntegration {
    public static void processPayment(Id bookingId, Decimal amount) {
                    Id = bookingId,
                    Payment_Status__c = 'Paid' // (Assuming you have this field)
                );
                update booking;

            } else {
                // Handle error
                System.debug('Payment Failed: ' + res.getStatus());
            }

        } catch (System.CalloutException e) {
            // Handle callout exception
            System.debug('Callout error: ' + e.getMessage());
        }
    }
}
```

```
Salesforce metadata XML IntelliSense is now available.
```



Developer Console - Google Chrome
orgfarm-ffd852816e-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
BookingTriggerHandler.apxc
PaymentGatewayIntegration.apxc
Code Coverage: None
API Version: 64
Logs Tests Checkpoints Query Editor View State Progress Problems

```
/*
 * This class handles all external payment processing.
 * It uses a @future method to make a callout,
 * which is required when invoked from a trigger.
 */
public class PaymentGatewayIntegration {

    // This method runs in the background.
    @future(callout=true)
    public static void processPayment(Id bookingId, Decimal amount) {

        // 1. Build the HTTP Request
        HttpRequest req = new HttpRequest();

        // 2. Set the Endpoint using the Named Credential
        // 'callout:Payment_API' is the magic part.
        // '/post' is the specific path on httpbin.org we want to send to.
        req.setEndpoint('callout:Payment_API/post');
        req.setMethod('POST');
        req.setHeader('Content-Type', 'application/json;charset=UTF-8');
```

User	Application	Operation	Time	Status	Read	Size
Uppari Pedda Venkatesh	Unknown	ApexTestHandler	10/18/2025, 11:24:12 AM	Success	Unread	525 bytes
Uppari Pedda Venkatesh	Unknown	ApexTestHandler	10/18/2025, 11:24:12 AM	Success	Unread	2.2 KB
Uppari Pedda Venkatesh	Unknown	ApexTestHandler	10/18/2025, 11:10:29 AM	Success	Unread	2.19 KB
Uppari Pedda Venkatesh	Unknown	ApexTestHandler	10/18/2025, 11:10:29 AM	Success	Unread	523 bytes
Uppari Pedda Venkatesh	Unknown	ApexTestHandler	10/18/2025, 11:05:57 AM	Success	Unread	2.2 KB
Uppari Pedda Venkatesh	Unknown	ApexTestHandler	10/18/2025, 11:05:43 AM	Success	Unread	523 bytes

Apex Callout Example (Mock Payment Processing)

An Apex class was created to handle the callout asynchronously, as callouts cannot be made directly from triggers.

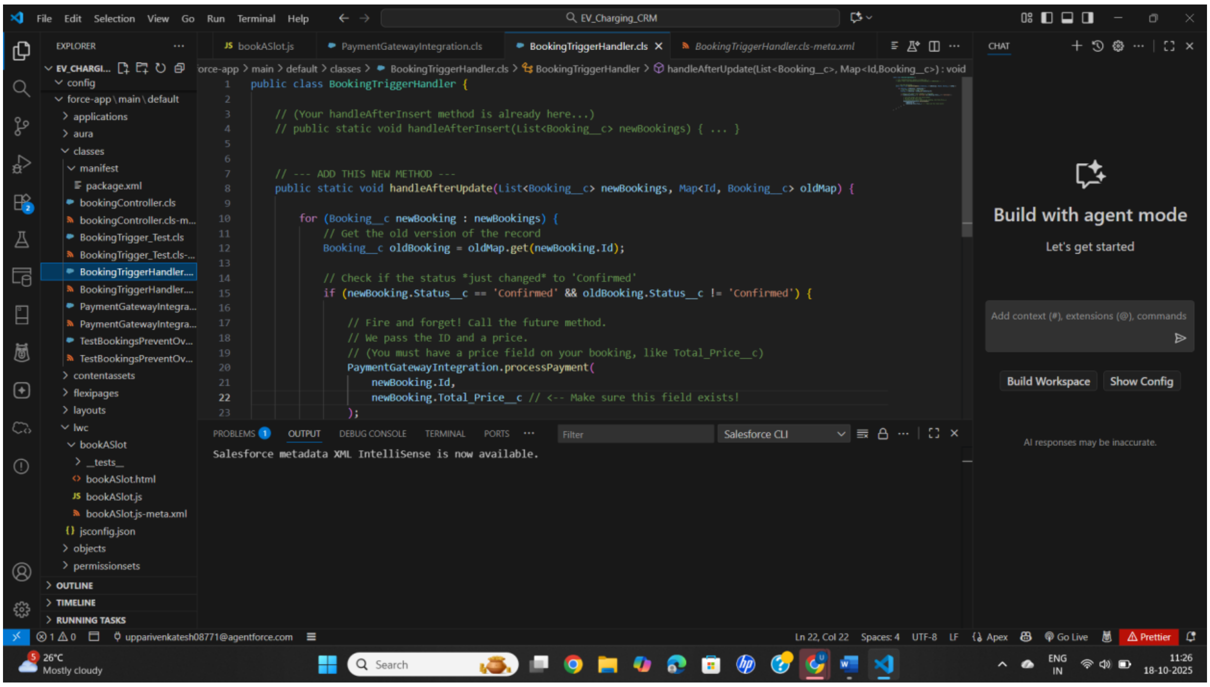
1. **Apex Class: PaymentGatewayIntegration**

- **Purpose:** Simulates sending booking details to an external payment processor.
- **Method: processPayment(Id bookingId, Decimal amount)**
    - **Annotation:** @future(callout=true) - This allows the method to make external calls and run asynchronously in the background when invoked (e.g., from a trigger).
    - **Logic:**
        1. Constructs an HttpRequest object.
        2. Sets the endpoint using the Named Credential: callout:Payment_API/post. The /post path tells httpbin.org to simply echo back the data sent.
        3. Sets the method to POST.
        4. Sets the Content-Type header to application/json.
        5. Creates a JSON body containing the bookingId and amount.
        6. Uses the Http class to send the request.
        7. Receives the HttpResponse.
        8. Checks the status code (expecting 200 for success from httpbin.org).
        9. Logs the response body using System.debug().
        10. If successful, performs a DML update on the original Booking__c record to set a field like Payment_Status__c to 'Paid'.
        11. Includes **try-catch block** for System.CalloutException to handle potential network or endpoint errors.

2. **Trigger Handler Modification: BookingTriggerHandler.handleAfterUpdate**

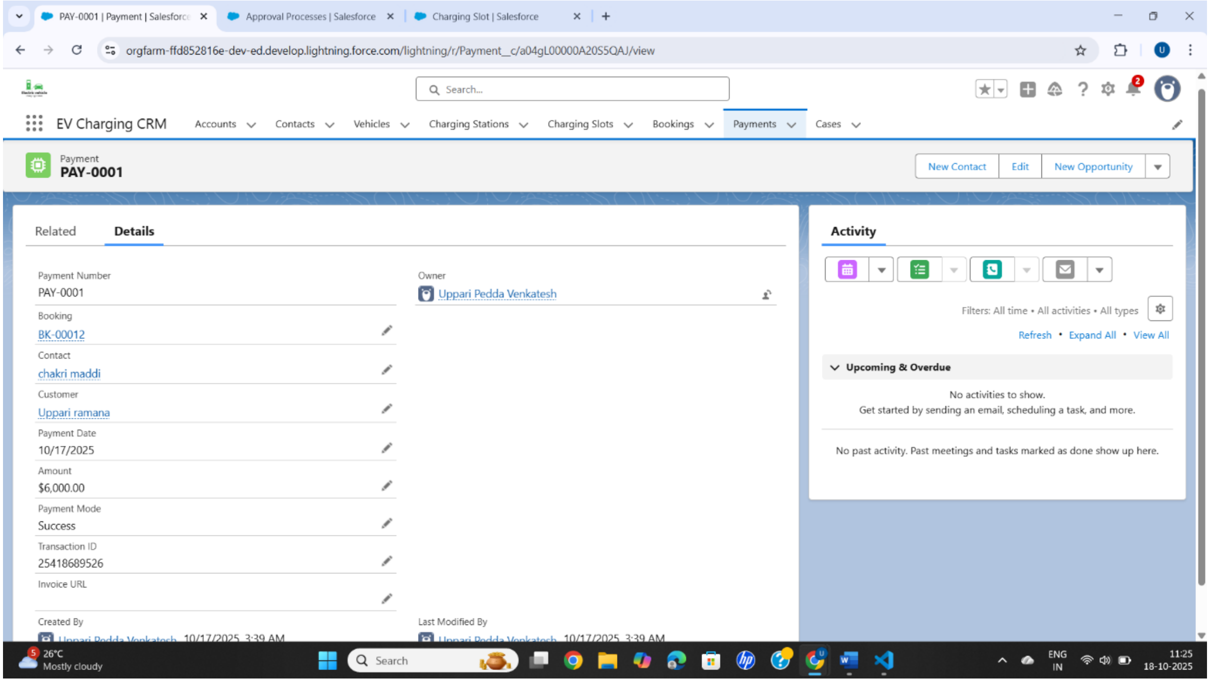
- **Purpose:** To invoke the payment callout when a booking is confirmed.

- **Logic:** Within the handleAfterUpdate method, a check was added: if a Booking__c record's Status__c changes from any value *to* 'Confirmed', the PaymentGatewayIntegration.processPayment(bookingId, totalPrice) future method is called, passing the necessary details.


EV_Charging_CRM
bookASlot.js
PaymentGatewayIntegration.cls
BookingTriggerHandler.cls
BookingTriggerHandler.cls-meta.xml
force-app > main > default > classes > BookingTriggerHandler.cls > BookingTriggerHandler > handleAfterUpdate(List<Booking__c>, Map<Id,Booking__c>) : void
Build with agent mode
Let's get started
Build Workspace
Show Config

```
public class BookingTriggerHandler {

    // (Your handleAfterInsert method is already here...)
    // public static void handleAfterInsert(List<Booking__c> newBookings) { ... }


    // --- ADD THIS NEW METHOD ---
    public static void handleAfterUpdate(List<Booking__c> newBookings, Map<Id, Booking__c> oldMap) {

        for (Booking__c newBooking : newBookings) {
            // Get the old version of the record
            Booking__c oldBooking = oldMap.get(newBooking.Id);

            // Check if the status *just changed* to 'Confirmed'
            if (newBooking.Status__c == 'Confirmed' && oldBooking.Status__c != 'Confirmed') {

                // Fire and forget! Call the future method.
                // We pass the ID and a price.
                // (You must have a price field on your booking, like Total_Price__c)
                PaymentGatewayIntegration.processPayment(
                    newBooking.Id,
                    newBooking.Total_Price__c // <-- Make sure this field exists!
                );
```

```
Salesforce metadata XML IntelliSense is now available.
```


PAY-0001 | Payment | Salesforce
Approval Processes | Salesforce
Charging Slot | Salesforce
orgfarm-ffd852816e-dev-ed.develop.lightning.force.com/lightning/r/Payment__c/a04gL00000A20S5QAJ/view
EV Charging CRM
Accounts
Contacts
Vehicles
Charging Stations
Charging Slots
Bookings
Payments
Cases
New Contact
Edit
New Opportunity
Related
Details
Activity

Payment
PAY-0001

Payment Number	PAY-0001
Booking	BK-00012
Contact	chakri maddi
Customer	Uppari ramana
Payment Date	10/17/2025
Amount	$6,000.00
Payment Mode	Success
Transaction ID	25418689526
Invoice URL	
Owner	Uppari Pedda Venkatesh
Created By	Uppari Pedda Venkatesh, 10/17/2025, 3:39 AM
Last Modified By	Uppari Pedda Venkatesh, 10/17/2025, 3:39 AM

Filters: All time • All activities • All types

Upcoming & Overdue

No activities to show.
Get started by sending an email, scheduling a task, and more.

No past activity. Past meetings and tasks marked as done show up here.