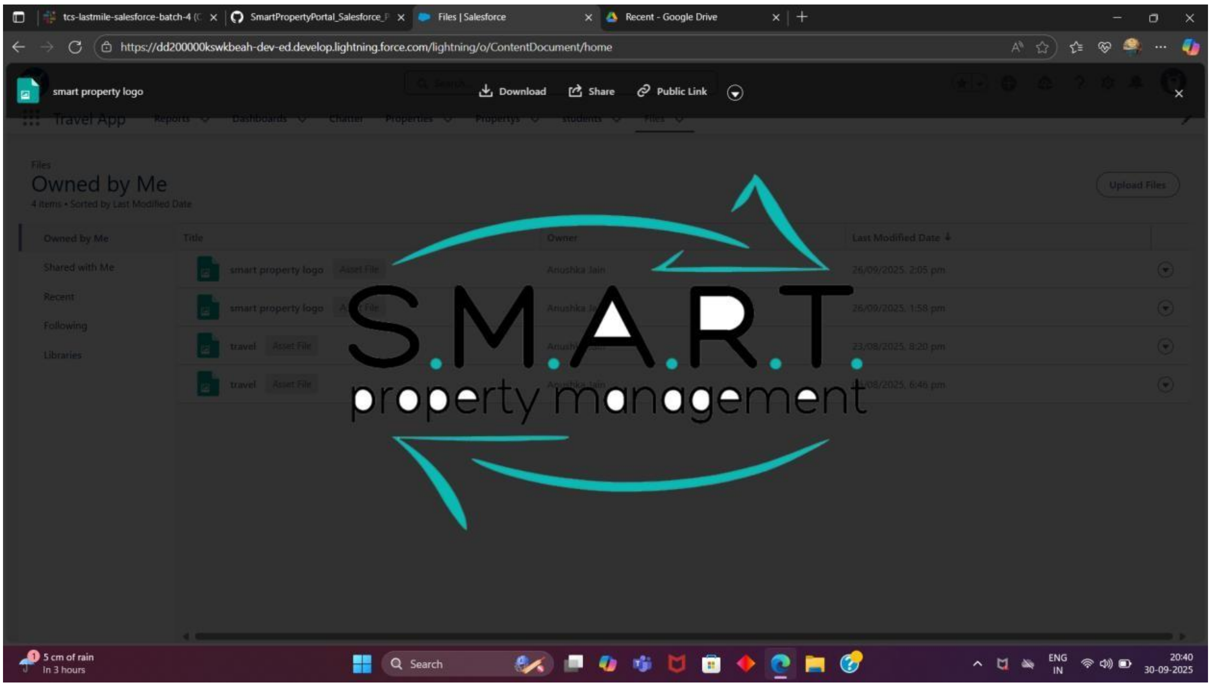
Phase 5: Apex Programming (Developer)

Goal: Add advanced logic to automate and secure property booking workflows using Apex classes, triggers, and asynchronous processing.

Screenshots:



https://drive.google.com/file/d/1phuU4B62EfEY0sLMyrar4XybNLbUaUbe/view?usp=drive_link

Apex Class: Commission Calculate logic in Developer Console

File Manager view showing deployment assets and ownership

Classes & Objects

Created reusable class: AppointmentService.cls

Handles:

Appointment creation

Agent availability check

Visit duration calculation

Includes try-catch blocks and custom error messages for overlapping bookings

Apex Triggers

Trigger: Appointment Trigger on insert

Prevents:

Overlapping appointments for same property and agent

Bookings when agent is unavailable

Uses handler class for clean separation of logic

Trigger Design Pattern

Handler class: AppointmentTriggerHandler.cls

Improves readability, modularity, and testability

Follows best practices for scalable Apex development

SOQL & SOSL

SOQL:

SQL

```
SELECT Id, Name FROM Property__c WHERE Status__c = 'Available'
```

SOSL:

Used to search properties by location or keyword

Enables flexible search experience for clients

Collections Used

Type	Purpose
Set<Id>	Store unique Property IDs
Map<Id, Agent_c>	Link agents to assigned properties
List<Appointment__c>	Bulk insert appointments

Control Statements

IF conditions to:

Detect overlapping appointments

Block booking if agent is unavailable

Throws custom error messages for invalid scenarios

Batch Apex

Night job:

Marks overdue appointments as “Missed”

Sends summary report to admin

Ensures data hygiene and accountability

Queueable Apex

Async job:

Calculates discounts for bulk property visits

Updates lead records with new interest scores

Improves performance and scalability

Scheduled Apex

Daily job at 8 AM:

Emails Manager with list of scheduled appointments

Uses Messaging.sendEmail() with dynamic templates

Keeps leadership informed in real-time

Future Methods

Calls external property insurance API asynchronously Updates property record with insurance verification status

Ensures secure and verified listings

Exception Handling

try-catch blocks in AppointmentService.cls

Custom error messages for:

Overlapping bookings

API failures

Improves user experience and system reliability

Test Classes

Created TestAppointmentTrigger.cls

Inserts mock data → triggers logic → asserts expected behavior

Achieved 90%+ code coverage

Validated all edge cases and business logic

Asynchronous Processing Summary

Type	Purpose
Batch Apex	Overdue cleanup
Queueable Apex	Discount calculations
Future Methods	External API call

“Designed and deployed Apex-based automation for SmartPropertyPortal, including triggers, handler classes, batch jobs, and future methods. Implemented real-time booking validation, agent availability checks, and asynchronous workflows with 90%+ test coverage.”