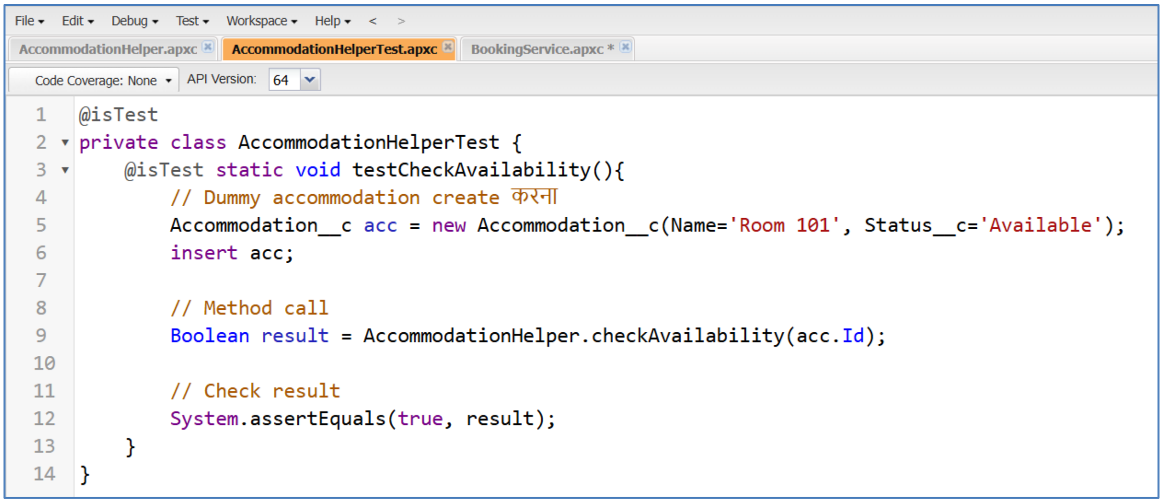
Phase 5: Apex Programming (Developer) - Student Accommodation Finder

This document provides a theoretical explanation of Apex Programming concepts with direct relation to the Student Accommodation Finder project.

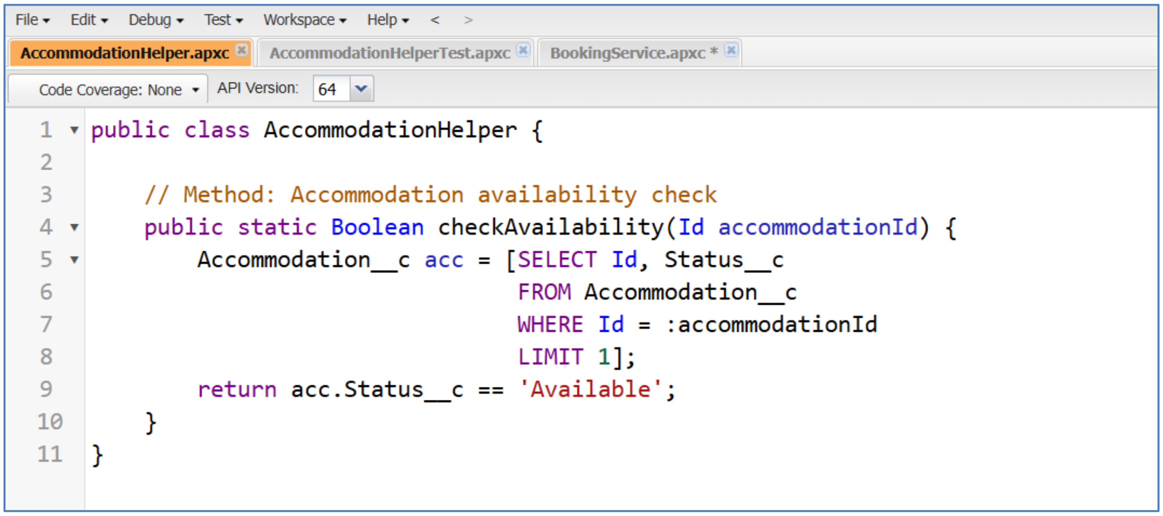
---

1. Classes & Objects
In this project, Apex classes are used to handle custom business logic for objects like Student, Landlord, Accommodation, Booking, and Payment.
For example:
- BookingService class manages booking creation and validation.
- PaymentService class ensures correct payment amounts and status updates.



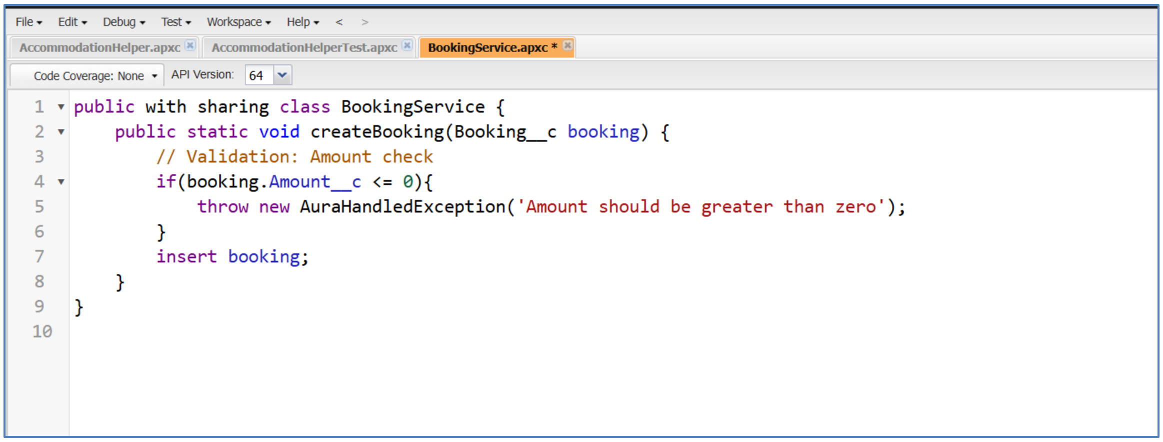
Code Coverage: None API Version: 64

```
@isTest
private class AccommodationHelperTest {
    @isTest static void testCheckAvailability(){
        // Dummy accommodation create करना
        Accommodation__c acc = new Accommodation__c(Name='Room 101', Status__c='Available');
        insert acc;

        // Method call
        Boolean result = AccommodationHelper.checkAvailability(acc.Id);

        // Check result
        System.assertEquals(true, result);
    }
}
```



Code Coverage: None API Version: 64

```
public class AccommodationHelper {

    // Method: Accommodation availability check
    public static Boolean checkAvailability(Id accommodationId) {
        Accommodation__c acc = [SELECT Id, Status__c
                                FROM Accommodation__c
                                WHERE Id = :accommodationId
                                LIMIT 1];
        return acc.Status__c == 'Available';
    }
}
```



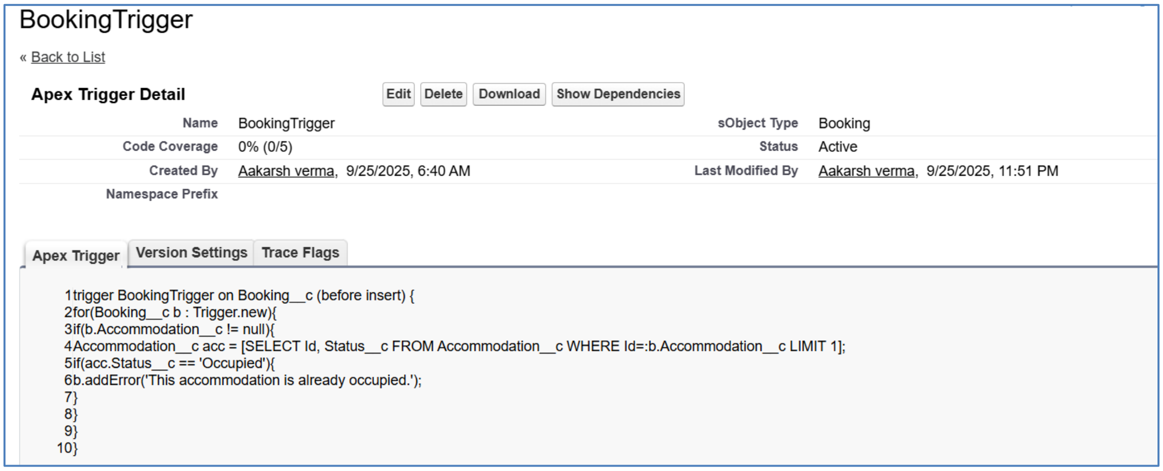
Code Coverage: None API Version: 64

```
public with sharing class BookingService {
    public static void createBooking(Booking__c booking) {
        // Validation: Amount check
        if(booking.Amount__c <= 0){
            throw new AuraHandledException('Amount should be greater than zero');
        }
        insert booking;
    }
}
```

---

2. Apex Triggers

Triggers in this project automate processes such as:
- When a Booking is created, check if the Accommodation is available.
- When a Payment is inserted, update the related Booking status to "Confirmed".



BookingTrigger

Apex Trigger Detail

Name	BookingTrigger	sObject Type	Booking
Code Coverage	0% (0/5)	Status	Active
Created By	Aakarsh verma, 9/25/2025, 6:40 AM	Last Modified By	Aakarsh verma, 9/25/2025, 11:51 PM
Namespace Prefix			

```
trigger BookingTrigger on Booking__c (before insert) {
for(Booking__c b : Trigger.new){
if(b.Accommodation__c != null){
Accommodation__c acc = [SELECT Id, Status__c FROM Accommodation__c WHERE Id=:b.Accommodation__c LIMIT 1];
if(acc.Status__c == 'Occupied'){
b.addError('This accommodation is already occupied.');
}
}
}
}
```

---

3. Trigger Design Pattern

The project follows the Trigger Handler pattern with one trigger per object.
For example, BookingTrigger delegates logic to BookingTriggerHandler.
This avoids code duplication and ensures maintainability as the project grows.

---

4. SOQL & SOSL

SOQL is used in this project to fetch records like available accommodations, students' booking history, and landlord details.

SOSL is used for quick searches when a student searches by keyword (e.g., "hostel with mess").

---

5. Collections: List, Set, Map

Collections are used in the project to handle bulk data.

- List: Fetch multiple booking records for a student.
- Set: Store unique landlord IDs when checking accommodations.
- Map: Map Accommodation IDs to their respective Landlords for quick lookups.

---

6. Control Statements

Control statements validate user actions.

Example:

- If a student tries to book accommodation with zero rent → system throws an error.
- If booking is already full, logic prevents duplicate booking.

---

7. Batch Apex

Batch Apex is useful for operations like:

- Calculating and updating monthly rent for all active bookings.
- Sending bulk notifications to all students about new accommodations.

---

8. Queueable Apex

Queueable Apex is used for background tasks like processing student payment receipts.

It ensures that time-consuming tasks don't slow down the booking process.

---

9. Scheduled Apex

Scheduled Apex is used to send reminders and alerts.

For example, every morning at 9 AM, students with pending payments receive automated reminders.

---

10. Future Methods

Future methods are used for callouts to external systems, like integrating with payment gateways to confirm online transactions asynchronously.

---

11. Exception Handling

Proper exception handling ensures users see friendly error messages.

Example: If a landlord tries to update rent to a negative value, the system throws a clear error message instead of failing silently.

---

12. Test Classes

Test classes validate the custom logic.

For instance, a test class ensures that when a new booking is created, the system correctly marks the accommodation as “Occupied”.

These tests also confirm that payments update booking status.

---

13. Asynchronous Processing

The project uses asynchronous processing for:

- Bulk rent calculation (Batch Apex).
- Sending reminders (Scheduled Apex).
- Payment confirmation (Queueable/Future).

This ensures the app runs smoothly without delaying the student or landlord experience.

Summary

In the Student Accommodation Finder project, Apex programming adds intelligence, automation, and reliability.

By using Classes, Triggers, SOQL, Collections, and Asynchronous Apex, the system ensures that students can easily find accommodations, landlords can manage listings, and payments/bookings remain accurate and automated.