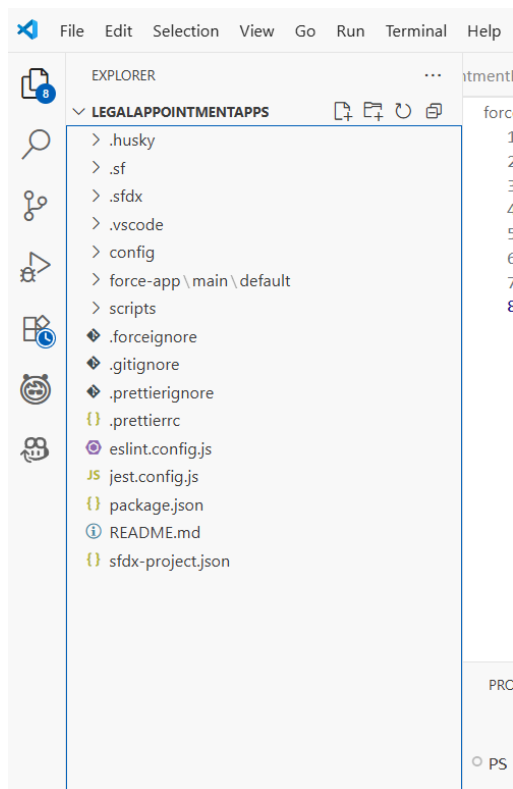# Phase 5: Apex Programming (Developer) – Online Appointment Booking for Legal Services
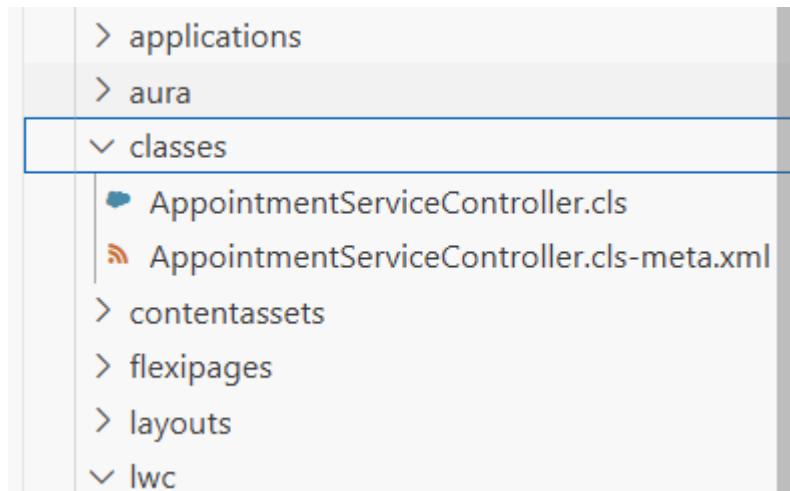
1. **Classes & Objects**

   o Create reusable Apex classes for business logic.

   o Example: AppointmentServiceController for handling appointment booking, lawyer availability, and client details.

   o Use **wrapper classes** if you need to pass multiple values between Lightning Web Components (LWCs) and Apex.

2. **Apex Triggers (before/after insert/update/delete)**

   o **Before Insert:** Validate that lawyer is available before saving an appointment.

   o **After Insert:** Automatically send appointment confirmation email (if not already handled by Flow).

   o **After Update:** If appointment status changes to *Cancelled*, release lawyer's availability slot.

   o **After Delete:** Free up lawyer's availability.
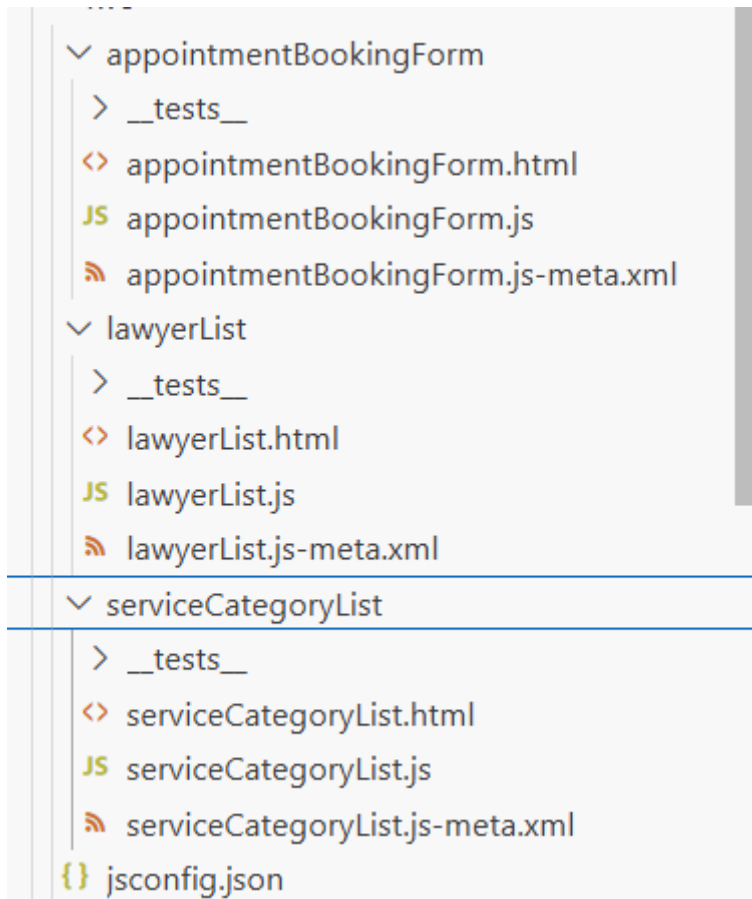
3.

3. Trigger Design Pattern

- o Implement a handler class to keep triggers clean.
- o Example:
  - AppointmentTrigger.trigger → delegates logic to AppointmentHandler.cls.
  - Ensures scalability and avoids recursion issues.

4. SOQL & SOSL

- o **SOQL Example:**
- o List<Lawyer__c> lawyers = [SELECT Id, Name, Experience__c, Location__c
- o   FROM Lawyer__c
- o   WHERE Service_Category__c = :selectedCategory];
- o **SOSL Example (search lawyers by name or location):**
- o List<List<SObject>> results = [FIND 'Delhi' IN ALL FIELDS RETURNING Lawyer__c(Id, Name, Location__c)];

5. Batch Apex

- o Process large appointment data in bulk (e.g., re-check lawyer availability for thousands of appointments).
- o Useful for data clean-up or system-wide updates.

```
∨ appointmentBookingForm
  > __tests__
  <> appointmentBookingForm.html
  JS appointmentBookingForm.js
  ∿ appointmentBookingForm.js-meta.xml
∨ lawyerList
  > __tests__
  <> lawyerList.html
  JS lawyerList.js
  ∿ lawyerList.js-meta.xml
∨ serviceCategoryList
  > __tests__
  <> serviceCategoryList.html
  JS serviceCategoryList.js
  ∿ serviceCategoryList.js-meta.xml
{} jsconfig.json
```

6. **Future Methods**

- Offload tasks like calling an external payment API or SMS gateway after booking confirmation.

7. **Exception Handling**

- Use try-catch-finally to prevent transaction failures.

- Example: Catch DML exceptions when inserting Appointment & Contact records.

8. **Test Classes**

- Write Apex test classes to cover triggers, controllers, and asynchronous logic.

- Ensure at least **75% coverage** before deploying to production.

- Example: Test booking an appointment with a new client vs existing client.

9. **Asynchronous Processing**

- Use **Batch, Queueable, Scheduled, and Future methods** to manage large-scale, time-consuming processes like:

    o   Sending thousands of appointment reminders.

    o   Syncing lawyer data with external legal databases.

Outcome of Phase 5

At the end of this phase:

- Business logic runs efficiently through Apex Classes & Triggers.

- Appointment scheduling is optimized with SOQL/SOSL queries.

- Lawyer availability updates in real-time.

- Asynchronous processes handle reminders, payments, and external integrations without slowing down the system.

- Test classes ensure reliability and deployment readiness.