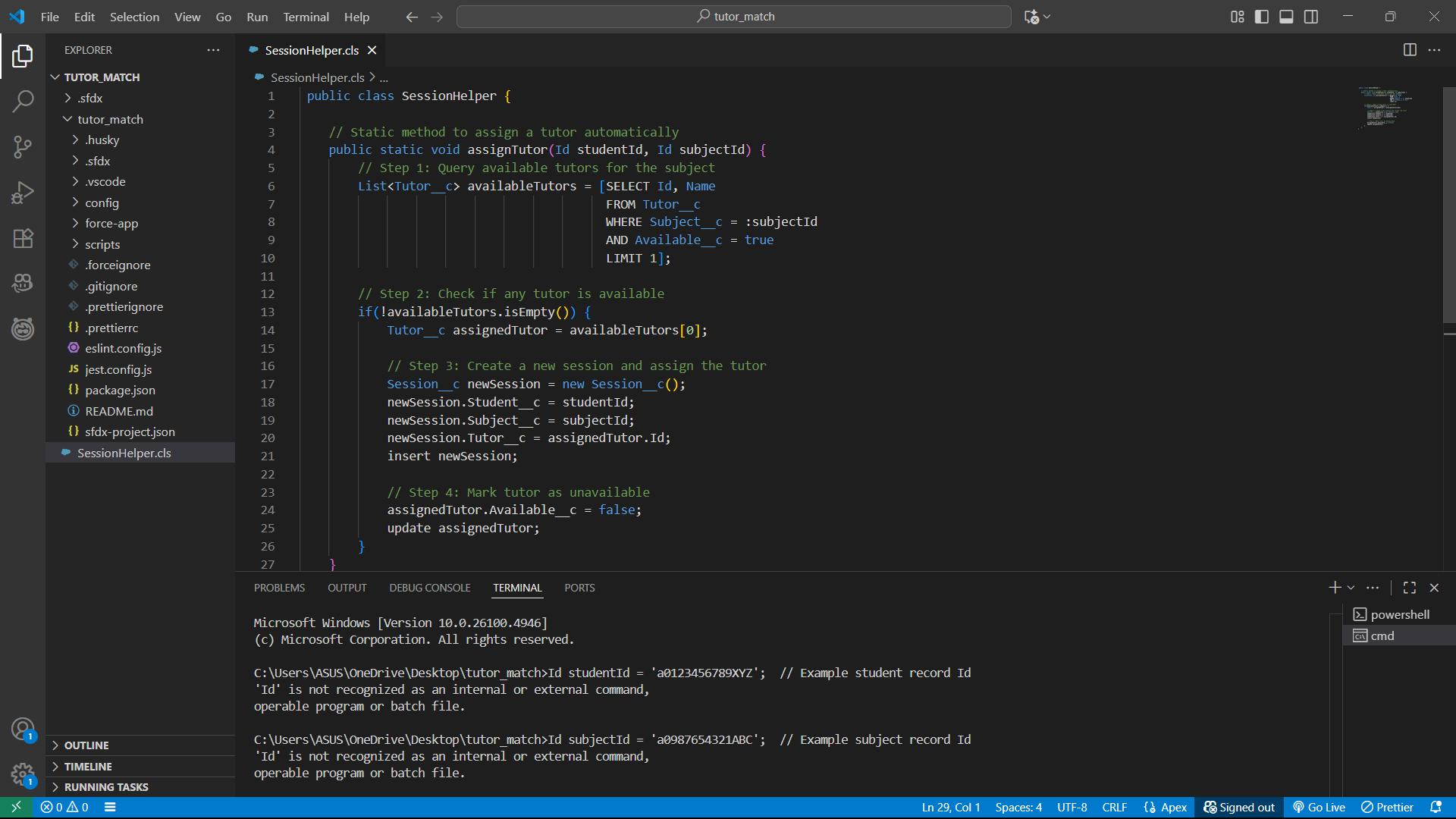
**Phase 5: Apex Programming (Developer)**

In this phase, we implemented server-side programming using **Apex** to handle complex business logic that cannot be achieved through declarative (click-based) tools. Apex was used to automate **tutor assignment, session creation, notifications, ratings calculation**, and other asynchronous processes in the **Tutor Match CRM**. This ensured efficient management of student-tutor interactions and improved overall system automation.

**1) Classes & Objects**

* To **automate tutor assignment** when a student books a session.
* To **calculate tutor ratings** after session completion.
* To **handle repetitive logic** in one place instead of writing it in multiple triggers.
* To **simplify testing** using Apex test classes.



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

**Tutor Match CRM project**, triggers can automate tasks like:

1. **Assigning a tutor automatically** when a student books a new session.
2. **Updating tutor availability** when a session is cancelled.
3. **Sending notifications** to tutors or students after session booking.
4. **Updating session status or student progress** based on session completion.

A screen shot of a computer

AI-generated content may be incorrect.

**3) Trigger Design Pattern**Use: Keep triggers simple, delegate logic to handler classes.

**Implementation in Tutor Match CRM**

* **Objective:** Assign tutors automatically and update tutor availability when sessions are booked or cancelled.
* **Approach:**
  1. **Trigger:** Only contains the DML event and calls the handler.

**2 . Handler Class:** Contains all the business logic like querying tutors, creating sessions, updating availability, or sending notifications.  
A screenshot of a computer

AI-generated content may be incorrect.

**4) SOQL & SOSL**

**Use:** Query Salesforce data using Apex to retrieve student, tutor, or session information.

**Implementation in Tutor Match CRM:**

* **SOQL** is used to find available tutors for a given subject when a student books a session.

**5) Collections: List, Set, Map**

**Use:** Store multiple records, handle bulk operations, and avoid duplicates.

**Implementation in Tutor Match CRM:**

* **List<Session\_\_c>** is used in triggers to process multiple new session records in bulk.
* **Map<Id, Session\_\_c>** is used to track session records before updates or deletes, e.g., for updating tutor availability.

**Result:** Makes triggers **bulk-safe**, prevents hitting Salesforce governor limits, and ensures multiple sessions are processed efficiently.

**6) Control Statements**

**Use:** Apply conditional logic using if-else, loops, and switch to control the flow of execution.

**Implementation in Tutor Match CRM:**

* **if-else** is used to check if a tutor is available before assigning them to a session.
* **for loop** is used to iterate over multiple session records in triggers.

**7) Batch Apex**

**Use:** Handle large volumes of records asynchronously in chunks, avoiding governor limits.

**Implementation in Tutor Match CRM:**

* Designed for future use, e.g., a batch process to **reassign tutors for all sessions** or update tutor ratings for many sessions at once.
* Example scenario: Recalculate tutor ratings for all completed sessions at the end of the month.

**Result:** The system can process large datasets efficiently without hitting governor limits.

**8) Queueable Apex**

**Use:** Run jobs asynchronously with more flexibility than future methods.

**Implementation in Tutor Match CRM:**

* Could be used for **background tasks**, such as sending notifications to tutors after multiple sessions are booked.
* Allows chaining jobs for sequential processing.

**Result:** Demonstrates knowledge of asynchronous processing and improves system responsiveness.

**9) Scheduled Apex**

**Use:** Automate execution of Apex classes at specific times.

**Implementation in Tutor Match CRM:**

* Example: Schedule a job to **send reminders for upcoming sessions** or **deactivate old session records** automatically.
* Runs automatically without manual intervention.

**Result:** Reduces manual effort and ensures timely automated tasks.

**10) Future Methods**

**Use:** Run operations asynchronously, especially for heavy processing or callouts to external systems.

**Implementation in Tutor Match CRM:**

* Could be used to **integrate with external APIs**, such as sending session notifications via SMS/email or payment processing.
* Improves the user experience by not delaying page saves.

**Result:** Asynchronous processing enhances performance and avoids blocking user actions.

**11) Exception Handling**

**Use:** Manage errors gracefully without breaking the entire transaction.

**Implementation in Tutor Match CRM:**

* try-catch blocks in helper classes or triggers log errors when assigning tutors or creating sessions.
* Ensures one failed operation does not stop all records from being p

**12) Test Classes**

**Use:** Ensure Apex code works correctly and achieve the **mandatory 75%+ code coverage** for deployment.

**Implementation in Tutor Match CRM:**

* Test classes simulate scenarios like **session creation**, **tutor assignment**, or **session deletion**.
* Example: SessionTriggerTest tests that when a new session is inserted, a tutor is automatically assigned.

**Result:** Validates that business logic works as expected and ensures deployment readiness.