**DAY 2 ASSIGNMENT – 3**

1. **DDD – Domain-Driven Design**

Understand the business deeply before coding. Focus on the "domain" (real-world rules).

Before you start coding, you talk to the business owner and learn things like:

* Users should be locked out after 3 failed logins.
* Some users (admins) should get SMS alerts on failed login.
* Passwords must be changed every 90 days.

You organize your code around these **real business rules**, not just technical things.

You may create:

* User, Admin, Login attempt classes

**2. FDD – Feature-Driven Development**

Break your app into small features and develop one by one.

**Example Features for Login:**

1. "User can enter username and password"
2. "System checks if user exists"
3. "Show error if wrong password"
4. "Redirect to homepage after success"

Each of these is treated as a **mini project**. Teams can work on different features in parallel.

**3. TDD – Test-Driven Development**

Write tests first, then write code to pass the tests.

**TDD** means:

1. **Write a test first** (before writing the actual code)
2. **Write the code** to pass the test
3. **Improve the code** (refactor)
4. Repeat for next feature

**Example (Login):**

* First: Write a test – "Login should succeed with correct password"
* Then: Write code to make that test pass
* Next: Add test – "Login should fail with wrong password"
* Continue building step by step

**BDD (Behaviour-Driven Development)**:

**Behaviour-Driven Development (BDD)** is a way of developing software that focuses on **how the software should behave from the user’s point of view**.

**Example : Online Shopping – Add to Cart**

* **Given** the user is on a product page
* **When** they click “Add to Cart”
* **Then** the product should appear in the cart

This describes user **behaviour**, not code!