1 What is Smart Contract Testing?

Testing checks **whether your smart contract works correctly** before deploying it to blockchain.

We test:

- Functions
- Conditions (`require`)
- Events
- State changes

---

2 Tools Used

- **Hardhat** → Development environment
- **Mocha** → Test runner (runs tests)
- **Chai** → Assertion library (`expect`, `to.equal`)

Tests are written in **JavaScript**, not Solidity.

---

3 Create a Sample Contract (Simple)

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract Counter {
    uint public count;

    function increment() public {
        count += 1;
    }

    function decrement() public {
        require(count > 0, "Count is zero");
        count -= 1;
    }
}
```

---

4 Project Setup (One Time)

```
mkdir test-demo
cd test-demo
npm init -y
npm install --save-dev hardhat
npx hardhat
```

Choose **“Create a basic sample project”**

---

5 Write Test File

`test/Counter.js`

```
const { expect } = require("chai");

describe("Counter Contract", function () {
  let counter;

  beforeEach(async function () {
    const Counter = await ethers.getContractFactory("Counter");
    counter = await Counter.deploy();
    await counter.deployed();
  });

  it("Initial count should be zero", async function () {
    expect(await counter.count()).to.equal(0);
  });

  it("Should increment count", async function () {
    await counter.increment();
    expect(await counter.count()).to.equal(1);
  });

  it("Should decrement count", async function () {
    await counter.increment();
    await counter.decrement();
    expect(await counter.count()).to.equal(0);
  });

  it("Should fail when decrementing zero", async function () {
    await expect(counter.decrement()).to.be.revertedWith("Count is zero");
  });
});
```

---

6 Run Tests

```
npx hardhat test
```

If all tests pass → Contract works correctly

---

7 Important Keywords (Easy Meaning)

Keyword	Meaning
`describe()`	Group of tests
`it()`	Single test case
`beforeEach()`	Runs before every test
`expect()`	Check expected result
`revertedWith()`	Check error message

---

8 What We Actually Tested

Initial value
Function logic
State change
Error handling (`require`)