**How Jest.mock() Works (Concise)**

**jest.mock() replaces an entire module with a mock version:**

jest.mock('../../Models/User');

1. **Intercepts the require/import** - When your code does require('../../Models/User'), Jest returns a mock instead of the real module
2. **Auto-mocks all functions** - Every function in the module becomes a Jest mock function (can be tracked and controlled)
3. **You control the behavior** - Use .mockResolvedValue(), .mockReturnValue(), etc. to define what the mocked functions return
4. **Tracks all calls** - Jest records every call to the mocked functions (arguments, call count, etc.)

**Example:**

jest.mock('../../Models/User'); // Mock the entire User model

User.create.mockResolvedValue({ \_id: '123' }); // Control what User.create returns

await registerUser(); // When this calls User.create(), it gets our mock value

expect(User.create).toHaveBeenCalledTimes(1); // Verify it was called

**Result:** Your tests run fast (no real database) and you control exactly what dependencies return. 🎯

**mockReturnValue() vs mockResolvedValue()**

**mockReturnValue()** - For **synchronous** functions (returns immediately)

createRandomString.mockReturnValue('abc123');

// When called: returns 'abc123' immediately

const result = createRandomString(); // 'abc123'

**mockResolvedValue()** - For **async** functions (returns a Promise)

User.create.mockResolvedValue({ \_id: '123' });

// When called: returns Promise.resolve({ \_id: '123' })

const result = await User.create(); // { \_id: '123' }

**Quick Rule:**

* Function returns a **value** → use mockReturnValue()
* Function returns a **Promise** (async/await) → use mockResolvedValue()

**In Your Test:**

createRandomString.mockReturnValue('7ycthivdrqvl'); // Sync function

argon2.hash.mockResolvedValue('$argon2id$hash'); // Async function (returns Promise)

User.create.mockResolvedValue({ \_id: '123' }); // Async function (returns Promise)

**Bonus:** mockRejectedValue() exists for async functions that should throw errors! 🎯