Functional Testing :

Definition:
Functional testing is a type of software testing where we check if the application works as expected according to the specified requirements. It focuses on what the system **does** rather than how it does it.

Why Do We Need Functional Testing?

- To verify that each feature of the software works correctly.
- To ensure that the software meets the user's needs and requirements.
- To check the functionality from the user's point of view.

Key Aspects of Functional Testing

1. **Based on Requirements**:
    - Functional testing uses the **requirements** of the software as a guide. We test to see if the application performs the tasks it’s supposed to.
2. **Focus on Output**:
    - It only checks the output of the system, not how the system gets that output.
3. **Testing User Actions**:
    - Tests are based on actions users will perform, like entering data, clicking buttons, and navigating pages.

Types of Functional Testing

1. **Unit Testing**:
    - Tests individual components or "units" to ensure they work correctly.
2. **Integration Testing**:
    - Tests combinations of modules to see if they interact correctly (covered earlier).
3. **Smoke Testing**:
    - A quick check to ensure basic functions work before detailed testing begins.
4. **Sanity Testing**:
    - Tests small sections after minor changes to see if everything still works as expected.
5. **Regression Testing**:

- Checks if new updates or bug fixes haven't affected existing features.

6. **User Acceptance Testing (UAT)**:
    - Conducted to ensure that the software meets the user's requirements and is ready for release.

Steps in Functional Testing

1. **Identify Test Requirements**:
    - List the features and functions that need to be tested according to the requirements.
2. **Create Test Scenarios**:
    - Outline actions a user might take to test each function.
3. **Design Test Cases**:
    - Write step-by-step instructions (test cases) to check if each function works as expected.
4. **Execute Test Cases**:
    - Run each test case to check if the software works correctly.
5. **Compare Results**:
    - Verify that the actual output matches the expected output.

Example of Functional Testing

Imagine testing a **login page**:

- Enter a correct username and password – check if it logs in successfully.
- Enter an incorrect password – check if it shows an error.
- Leave username or password blank – check if it shows a warning.

Benefits of Functional Testing

- **Ensures Requirements Are Met**: Confirms that all expected features work.
- **Improves User Satisfaction**: By testing from a user's perspective, it helps make the application user-friendly.
- **Reduces Risk of Bugs**: Catches errors in how the system performs essential functions.