

# **WEEK 1: BASIC TASKS – QA FOUNDATIONS & MANUAL TESTING**

---

## **What is Software Testing?**

Software testing checks whether a software application works as expected and is free from defects. Its goal is to ensure quality, reliability, and a smooth user experience before release.

---

## **Types of Software Testing**

### **Manual Testing**

Testing performed manually by a tester without using automation tools. Best for UI, usability, and exploratory testing; no programming required.

### **Automation Testing**

Testing performed using scripts and tools like Selenium or Cypress. Ideal for repetitive and regression tests; requires programming knowledge.

---

## **Black-Box Testing vs White-Box Testing**

### **Black-Box Testing**

Tester evaluates software functionality without knowing internal code. Focuses on input/output and user perspective.

### **White-Box Testing**

Tester examines internal code, logic, paths, and conditions. Used mainly by developers to test code correctness.

---

## **SDLC vs STLC**

### **SDLC (Software Development Life Cycle)**

Full process of planning, designing, developing, testing, deploying, and maintaining software. Goal: Build a complete, high-quality software product.

### **STLC (Software Testing Life Cycle)**

Testing-focused process to ensure software quality.

Goal: Identify defects and verify functionality.

---

## Test Levels

### Unit Testing

Tests individual components or functions.

Performed by developers to check internal logic.

### Integration Testing

Tests interaction between multiple modules.

Ensures modules work together correctly.

### System Testing

Tests the complete integrated system.

Validates end-to-end functionality from a user perspective.

### Acceptance Testing

Final testing to confirm software meets business and user requirements.

Performed by clients, end users, or QA before release.

---

## Test Artifacts

**Test Plan:** Document defining scope, approach, resources, and schedule of testing.

**Test Case:** Step-by-step instructions to verify a functionality with expected results.

**Test Scenario:** High-level description of what to test for complete coverage.

**Test Report:** Summary of testing activities and overall quality status.

**Bug Report:** Detailed document describing defects for developers to fix.

---

## Types of Testing

**Functional Testing:** Verifies software works according to requirements.

**Regression Testing:** Ensures new changes don't break existing functionality.

**Smoke Testing:** Quick check of critical features to confirm build stability.

**Sanity Testing:** Quick verification after minor changes or bug fixes.

**UAT (User Acceptance Testing):** Confirms software meets business and user needs.

---

# Non-Functional Testing

**Performance Testing:** Checks speed, responsiveness, and stability under load.

**Security Testing:** Identifies vulnerabilities and protects data from threats.

**Compatibility Testing:** Ensures software works on different devices, browsers, and OS.

**Usability Testing:** Evaluates ease of use, navigation, and overall user experience.

## Hands-On Tasks – QA Manual Testing

### 1. Test Cases for a Login Form

<u>Test Case ID</u>	<u>Title</u>	<u>Scenario</u>	<u>Expected Result</u>	<u>Actual Result</u>	<u>Status</u>	<u>QA Comment</u>
TC01	Valid Login	Enter valid username & password	User should log in successfully	User successfully logged in	Pass	Login functionality works as expected
TC02	Invalid Password	Enter valid username & wrong password	Error message: "Invalid credentials"	Error message displayed correctly	Pass	App handles wrong password correctly
TC03	Empty Username	Leave username empty & enter password	Error message: "Username required"	Error message displayed correctly	Pass	Username validation works properly
TC04	Empty Password	Enter username & leave password empty	Error message: "Password required"	Error message displayed correctly	Pass	Password validation works properly
TC05	Password Masking	Enter password	Password should be hidden/masked	Password is masked (****)	Pass	Password input is secure and masked

### 2. Sample Bug Report (Excel / Google Sheets)

<u>Bug ID</u>	<u>Summary</u>	<u>Steps to Reproduce</u>	<u>Expected Result</u>	<u>Actual Result</u>	<u>Severity</u>	<u>Status</u>	<u>Comments</u>
BUG01	Login allows empty password	1. Go to login page 2. Enter username 3. Leave password empty 4. Click Login	Error message "Password required"	User logged in without password	High	Open	Critical security issue
BUG02	Forgot password link not working	Click on "Forgot Password" link on login page	Redirects to reset password page	Nothing happens	Medium	Open	Needs developer fix

### **3. Test Scenarios for a Live App (e.g., Gmail / Trello)**

1. Verify user can log in with valid credentials.
  2. Verify error message on invalid login.
  3. Verify "Forgot Password" workflow.
  4. Verify account logout functionality.
  5. Verify email/task creation works correctly.
  6. Verify attachments can be uploaded successfully.
  7. Verify UI elements (buttons, links, icons) are visible and clickable.
  8. Verify notifications are received on new emails/tasks.
  9. Verify mobile responsiveness of the app.
  10. Verify search functionality returns accurate results.
-