## ✅ 1. How to Manually Test a Website – Step-by-Step

### Step 1: Understand Requirements

* Read **Business Requirements Document (BRD)**
* Understand the **purpose, features, and workflow** of the website
* Clarify any doubts with stakeholders or developers

### Step 2: Prepare Test Cases

* List out test cases for each functionality
* Use a spreadsheet or test case management tool like TestRail/JIRA
* Include: Test case ID, description, steps, expected result, actual result, status

### Step 3: Set Up Testing Environment

* Browser(s): Chrome, Firefox, Safari, Edge
* Devices: Desktop, Tablet, Mobile
* Data: Sample test data (valid/invalid inputs)

### Step 4: Execute Test Cases

* Perform test steps manually
* Compare expected vs actual results
* Record bugs with screenshots and steps to reproduce

### Step 5: Report Defects

* Use a bug tracking tool like JIRA, Bugzilla, etc.
* Include:
  + Title
  + Steps to reproduce
  + Severity & priority
  + Screenshots or videos

### Step 6: Retest & Regression Test

* Retest after developer fixes the bugs
* Regression test other areas to ensure new bugs are not introduced

## 🔍 2. Aspects of Website Testing

| **Aspect** | **Description** |
| --- | --- |
| **Functional Testing** | Check each feature works as expected (e.g., login, search) |
| **Usability Testing** | Ensure user-friendly navigation, design, and layout |
| **Compatibility Testing** | Works across different browsers/devices/resolutions |
| **UI/UX Testing** | Check alignments, colors, font consistency, responsive design |
| **Security Testing** | Test for SQL injection, XSS, and HTTPS usage |
| **Performance Testing** (basic) | Page load speed, image optimization, etc. |
| **Validation Testing** | Input field validations, form submission behavior |
| **Link Testing** | No broken links, all hyperlinks redirect correctly |
| **Accessibility Testing** | Screen reader support, alt text for images, keyboard nav |
| **Error Handling** | Proper error messages shown for incorrect actions |

## 🧪 3. Example Test Scenarios for Manual Testing

### 🔐 Login Page

* Valid username and password
* Invalid username and/or password
* Empty fields and login attempt
* Password masking
* "Forgot Password" functionality
* Login with enter key or button click
* Session timeout behavior

### 📝 Form Validation

* Submit form with all valid data
* Submit with missing mandatory fields
* Email format validation
* Numeric fields accepting text
* Reset button clears fields

### 🛒 E-commerce (Cart)

* Add item to cart
* Remove item from cart
* Update quantity
* Checkout with empty cart
* Apply valid/invalid coupon code

### 🔗 Link Testing

* Check all navigation menu links
* Check all footer links
* Test external links open in new tab

### 🌐 Browser/Device Compatibility

* Test on Chrome, Firefox, Safari, Edge
* Responsive design on mobile and tablet
* Landscape vs portrait modes

### 📜 General UI/UX

* Logo and company name visible
* Consistent font, colors, spacing
* Proper alignment of elements
* Tooltip appears when hovering icons

## 🧾 Bonus: Checklist for Manual Web Testing

| **Item** | **Check** |
| --- | --- |
| Functionality | ✅ |
| UI/UX | ✅ |
| Validations | ✅ |
| Responsiveness | ✅ |
| Browser Compatibility | ✅ |
| Error Handling | ✅ |
| Security (basic) | ✅ |
| Link Navigation | ✅ |
| Performance (manual) | ✅ |
| Accessibility (basic) | ✅ |

In software testing, **testing techniques** are systematic methods used to design test cases. These techniques help ensure maximum coverage with minimum effort. They can be broadly categorized into:

**✅ 1. Black Box Testing Techniques**

Focus on **what** the system does, without knowing the internal code.

| **Technique** | **Description** | **Example** |
| --- | --- | --- |
| **Equivalence Partitioning** | Divide input into valid and invalid groups; test one from each group | Age field: valid (18–60), invalid (<18, >60) |
| **Boundary Value Analysis (BVA)** | Test at edges of input ranges | For range 1–100: test 0, 1, 100, 101 |
| **Decision Table Testing** | Test combinations of inputs & rules | Discounts based on user type and purchase amount |
| **State Transition Testing** | Test behavior based on states and transitions | Login: Logged Out → Logged In → Timeout |
| **Error Guessing** | Based on tester experience to guess error-prone areas | Submitting form without filling fields |
| **Use Case Testing** | Based on real user workflows | Buying a product, making payment, receiving confirmation |

**✅ 2. White Box Testing Techniques**

Focus on **how** the system works (internal logic/code-based).

| **Technique** | **Description** | **Example** |
| --- | --- | --- |
| **Statement Coverage** | Test every line of code at least once | Ensure every line in the login function runs |
| **Branch/Decision Coverage** | Test all IF/ELSE or decision points | Cover both true and false outcomes |
| **Condition Coverage** | Test all conditions in logical expressions | if (A && B) → test A=T/B=F, A=F/B=T, A=T/B=T |
| **Path Coverage** | Test all possible paths in the code | Covers all execution flows |

**✅ 3. Experience-Based Testing Techniques**

Based on tester’s skills, intuition, and product understanding.

| **Technique** | **Description** |
| --- | --- |
| **Exploratory Testing** | Freestyle testing while learning the system |
| **Ad-hoc Testing** | Informal, unstructured testing without documentation |
| **Checklist-Based Testing** | Use checklists of common defects or features |

**✅ 4. Gray Box Testing Techniques**

Combination of black-box + white-box knowledge (e.g., API structure + UI).

| **Technique** | **Description** |
| --- | --- |
| **Matrix Testing** | Test interaction between components/modules |
| **Regression Testing** | Test modified areas + surrounding impact |
| **Pattern Testing** | Identify repeating defect patterns (e.g., SQL injection points) |

**🧠 Summary Table**

| **Category** | **Techniques** |
| --- | --- |
| Black Box | Equivalence Partitioning, BVA, Decision Table, State Transition, Error Guessing, Use Case |
| White Box | Statement, Branch, Condition, Path |
| Experience-Based | Exploratory, Ad-hoc, Checklist |
| Gray Box | Regression, Matrix, Pattern |

Would you like me to provide **real-world examples for each technique**, or help create test cases using these techniques for your website or project?