Interview Preparation - Image Info Systems Pvt Ltd

# Manual Testing (11 Questions & Answers)

1. Q1: What is Manual Testing?

Answer: Manual testing is testing software manually without automation tools. Testers execute test cases, observe results, and report defects.

1. Q2: Difference between Severity and Priority?

Answer: Severity = Impact of bug on system. Priority = Order in which bug should be fixed.  
Example: Spelling mistake on homepage is Low Severity but High Priority.

1. Q3: Explain Defect Life Cycle.

Answer: New → Assigned → Open → Fixed → Retest → Closed → Reopen (if still failing).

1. Q4: What is Regression Testing?

Answer: Re-running previously executed tests after a code change to ensure old functionality still works.

1. Q5: What is a Test Case?

Answer: A set of steps with input data and expected result to verify a specific feature works.

1. Q6: Explain Test Scenario vs Test Case.

Answer: Test Scenario = High-level functionality to test.  
Test Case = Step-by-step instructions to verify scenario.

1. Q7: What is Smoke Testing?

Answer: Basic test to check critical functionalities of the application before deeper testing.

1. Q8: What is Sanity Testing?

Answer: Focused testing after changes to ensure specific functionality works as expected.

1. Q9: What is Black Box and White Box Testing?

Answer: Black Box: Test without looking at internal code. White Box: Test with knowledge of internal logic.

1. Q10: What is Exploratory Testing?

Answer: Unscripted testing where tester explores application to find defects intuitively.

1. Q11: When do you stop testing?

Answer: When all test cases executed, critical bugs fixed, and exit criteria met.

# SQL / Database (11 Questions & Answers)

1. Q1: What is Primary Key and Foreign Key?

Answer: Primary Key uniquely identifies rows. Foreign Key references primary key in another table.

1. Q2: What is Normalization?

Answer: Process of structuring data to remove redundancy.  
1NF, 2NF, 3NF are commonly used forms.

1. Q3: What are Joins? Explain types.

Answer: Joins combine data from multiple tables.  
Types: INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN.

1. Q4: What is a View?

Answer: A virtual table created from a SELECT query. Used for security and simplification.

1. Q5: Write SQL to fetch employees in HR department.

Answer: SELECT \* FROM employees WHERE department='HR';

1. Q6: Find duplicates in a table.

Answer: SELECT column, COUNT(\*) FROM table GROUP BY column HAVING COUNT(\*) > 1;

1. Q7: Difference between WHERE and HAVING?

Answer: WHERE filters before grouping, HAVING filters after grouping (with aggregate functions).

1. Q8: What are Aggregate Functions?

Answer: COUNT, SUM, AVG, MIN, MAX used for summary calculations.

1. Q9: What is Index? Advantage?

Answer: Index improves data retrieval speed but may slow down insert/update operations.

1. Q10: What is a Subquery?

Answer: Query inside another query. Can be correlated or non-correlated.

1. Q11: What is a Stored Procedure?

Answer: Precompiled set of SQL statements stored in database for reuse.

# Automation Testing (11 Questions & Answers)

1. Q1: What is Automation Testing?

Answer: Testing using scripts/tools to execute test cases automatically.

1. Q2: When should we automate test cases?

Answer: When tests are repetitive, regression heavy, and need faster execution.

1. Q3: Which tools do you know?

Answer: Selenium WebDriver, TestNG/JUnit, Postman, (mention tools you worked on).

1. Q4: What is a Test Framework?

Answer: A set of guidelines for creating and managing test cases efficiently.  
Example: Data-driven, Keyword-driven, Hybrid.

1. Q5: How do you handle dynamic elements in Selenium?

Answer: Use XPath with contains(), CSS selectors, and explicit waits (WebDriverWait).

1. Q6: What are Locators in Selenium?

Answer: Mechanisms to identify web elements: ID, Name, Class, XPath, CSS Selector, LinkText, etc.

1. Q7: What is Implicit and Explicit Wait?

Answer: Implicit Wait: Global wait for all elements.  
Explicit Wait: Conditional wait for a specific element.

1. Q8: How do you generate reports?

Answer: Use ExtentReports or TestNG default reports to show pass/fail results.

1. Q9: Difference between Selenium RC and WebDriver?

Answer: WebDriver is faster, uses native browser automation, does not need server like Selenium RC.

1. Q10: What is Continuous Integration?

Answer: Practice of merging code frequently and running automated builds/tests (e.g., Jenkins pipeline).

1. Q11: What is Headless Browser Testing?

Answer: Running tests without opening UI browser (e.g., Chrome Headless, PhantomJS). Useful in CI/CD.

# 🧪 Manual Testing / QA – Answers

**Q1: What is manual testing? What are its advantages and disadvantages?**  
**A:** Manual testing is testing software without automation tools, where a tester executes test cases manually.  
✅ **Advantages:** Finds usability issues, flexible, cheaper for small projects.  
❌ **Disadvantages:** Time-consuming, prone to human error, not ideal for regression.

**Q2: What are different types of manual testing?**  
**A:**

* **Functional Testing:** Verifies each feature works as expected.
* **Integration Testing:** Tests interactions between modules.
* **System Testing:** End-to-end testing of entire application.
* **Regression Testing:** Ensures old functionality works after changes.
* **User Acceptance Testing (UAT):** Validates software meets user needs.
* **Non-Functional Testing:** Performance, security, usability.

**Q3: Describe test case, test scenario, test plan. How do you write good test cases?**  
**A:**

* **Test Case:** Step-by-step actions + expected result.
* **Test Scenario:** High-level functionality to test.
* **Test Plan:** Document describing scope, approach, resources, schedule.  
  ✅ Good test cases: clear steps, valid/invalid data, expected results, re-usable.

**Q4: Difference between severity and priority?**  
**A:**

* **Severity:** Impact of bug on system.
* **Priority:** Urgency of fixing bug.  
  Example: UI color mismatch → Low severity, High priority (customer-facing).

**Q5: Explain Defect Life Cycle.**  
**A:**

1. New → 2. Assigned → 3. Open → 4. Fixed → 5. Retest → 6. Closed → (Reopen if issue persists).

**Q6: How do you decide when to stop testing?**  
**A:**

* All critical test cases executed
* High severity defects fixed
* Coverage goals met
* Deadline or exit criteria reached

**Q7: Black Box vs White Box Testing (with examples):**  
**A:**

* **Black Box:** No knowledge of internal code (e.g., testing login form).
* **White Box:** Test with knowledge of code logic (e.g., unit testing functions).

**Q8: Boundary Value Analysis, Equivalence Partitioning, Error Guessing**  
**A:**

* **BVA:** Test edges of input range (e.g., min, max, just outside limits).
* **EP:** Divide input into valid/invalid classes and test one value per class.
* **Error Guessing:** Based on tester’s intuition/experience.

**Q9: How do you ensure test coverage?**  
**A:** Map test cases to requirements using a **Requirement Traceability Matrix (RTM)** to ensure all requirements are tested.

**Q10: What is regression testing? When do you perform it?**  
**A:** Testing old functionality after new code changes to ensure nothing is broken. Performed after bug fixes, enhancements, builds.

**Q11: Scenario-based question (e.g., Login testing)**  
**A:**

* **Positive tests:** Valid username/password → login successful
* **Negative tests:** Wrong password, blank fields, SQL injection, special characters
* **UI tests:** Button alignment, error message clarity
* **Performance:** Multiple login attempts quickly
* **Security:** Check lockout after multiple failures

# 🗄️ SQL / Database – Answers

**Q1: Primary key, foreign key, constraints**  
**A:**

* **Primary Key:** Unique, not NULL (e.g., emp\_id).
* **Foreign Key:** References primary key in another table.
* **Constraints:** Rules like NOT NULL, UNIQUE, CHECK, DEFAULT.

**Q2: SQL to fetch common records between two tables**

SELECT \*

FROM table1

INNER JOIN table2

ON table1.id = table2.id;

**Q3: JOIN & Types**  
**A:**

* **INNER JOIN:** Matching rows only
* **LEFT JOIN:** All left + matching right rows
* **RIGHT JOIN:** All right + matching left rows
* **FULL JOIN:** All rows from both tables

**Q4: GROUP BY and HAVING**  
**A:**  
GROUP BY groups rows with same values.  
HAVING filters groups after aggregation.

SELECT department, COUNT(\*)

FROM employees

GROUP BY department

HAVING COUNT(\*) > 5;

**Q5: Aggregate functions**  
**A:** COUNT, SUM, AVG, MIN, MAX

SELECT COUNT(\*), AVG(salary), MAX(salary) FROM employees;

**Q6: Normalization (1NF, 2NF, 3NF)**  
**A:**

* **1NF:** Atomic columns, no repeating groups
* **2NF:** No partial dependency (applies to composite keys)
* **3NF:** No transitive dependency (non-key depends only on key)

**Q7: Indexes**  
**A:** Improve search speed but slow down insert/update due to reindexing.

**Q8: Subquery types**  
**A:**

* **Non-correlated:** Runs once, result used in main query
* **Correlated:** Runs for each row in outer query

**Q9: View, triggers, stored procedures**  
**A:**

* **View:** Virtual table from SELECT query
* **Trigger:** Auto-executes on INSERT/UPDATE/DELETE
* **Stored Procedure:** Predefined SQL logic for reuse

**Q10: Find duplicates**

SELECT name, COUNT(\*)

FROM employees

GROUP BY name

HAVING COUNT(\*) > 1;

**Q11: Total orders per product**

SELECT p.product\_name, COUNT(o.order\_id) AS total\_orders

FROM products p

JOIN orders o ON p.product\_id = o.product\_id

GROUP BY p.product\_name;

# 🤖 Automation Testing – Answers

**Q1: What is Automation Testing?**  
**A:** Running tests automatically using tools like Selenium instead of manually.

s

**Q2: When to automate / when not**  
**A:**  
✅ Automate repetitive, regression, data-heavy, stable tests  
❌ Don’t automate ad-hoc, rapidly changing features, one-time tests

**Q3: Tools used**  
**A:** Selenium WebDriver, TestNG, JUnit, Postman, Jenkins (mention what you know)

**Q4: Advantages & challenges**  
**A:**  
✅ Saves time, runs 24/7, reduces human error  
❌ High setup cost, script maintenance, tool limitations

**Q5: Test script / suite / framework**  
**A:**

* **Script:** Code for one test
* **Suite:** Collection of test scripts
* **Framework:** Reusable structure (Keyword/Data/Hybrid)

**Q6: CI/CD & automation fit**  
**A:** CI/CD automates build/test/deployment pipeline. Automated tests give quick feedback after every code commit.

**Q7: Handling dynamic elements**  
**A:** Use XPath functions (contains(), starts-with()), CSS selectors, explicit waits (WebDriverWait).

**Q8: Maintaining automation scripts**  
**A:** Use page object model (POM), reusable functions, update locators regularly, run smoke tests after UI change.

**Q9: Reporting in automation**  
**A:** Generate HTML reports using ExtentReports, Allure, or TestNG reports. Include screenshots for failed tests.

**Q10: Scenario-based automation (login → checkout)**  
**A:**

1. Launch browser
2. Open app
3. Automate login
4. Search for product, add to cart
5. Proceed to checkout, verify total
6. Validate confirmation message  
   ✅ Use Selenium + TestNG + Assertions

### ****Test Case for Pen****

**Test Case ID:** TC\_PEN\_001  
**Test Case Name:** Verify that the pen writes smoothly

| **Field** | **Details** |
| --- | --- |
| **Pre-condition** | Pen should have ink filled and be ready to use |
| **Test Steps** | 1. Take the pen  2. Write on a piece of paper |
| **Test Data** | Normal white paper |
| **Expected Result** | Pen should write smoothly, with clear and continuous ink flow, without breaks or leakage |
| **Post-condition** | Pen remains functional for further use |
| **Actual Result** | (Tester will fill this after testing) |
| **Status** | Pass / Fail |

### ****Test Case for Email****

**Test Case ID:** TC\_EMAIL\_001  
**Test Case Name:** Verify that user can send an email successfully

| **Field** | **Details** |
| --- | --- |
| **Pre-condition** | User must be logged into the email account |
| **Test Steps** | 1. Open email application/webmail  2. Click **Compose**  3. Enter a valid recipient email address  4. Enter subject and body  5. Click **Send** |
| **Test Data** | Recipient: test@example.com  Subject: "Test Email"  Body: "This is a test email." |
| **Expected Result** | Email should be sent successfully and appear in **Sent Items** folder |
| **Post-condition** | Recipient should receive the email in their inbox |
| **Actual Result** | (Tester will fill this after testing) |
| **Status** | Pass / Fail |