

Software Testing Assignment - Module_2

1.What is Exploratory Testing?

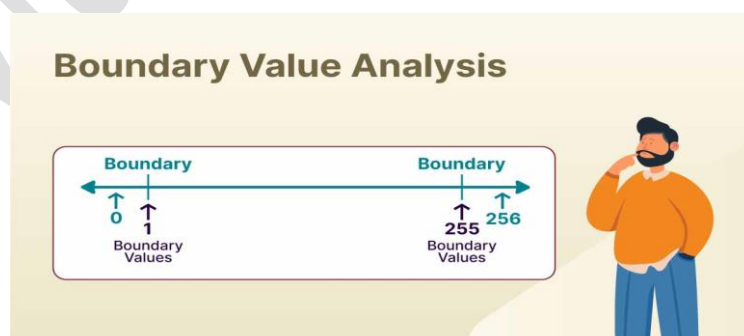
Exploratory testing is an unscripted, hands-on approach to testing software, where a tester:

- Learns about the software's features and functionality
- Designs and executes tests simultaneously
- Explores the software to identify defects and issues
- Uses their skills, experience, and creativity to test the software

2.What is traceability matrix?

traceability matrix is a document that demonstrates the relationship between requirements and other artifacts. It's used to prove that requirements have been fulfilled. And it typically documents requirements, tests, test results, and issues.

3.What is Boundary value testing?



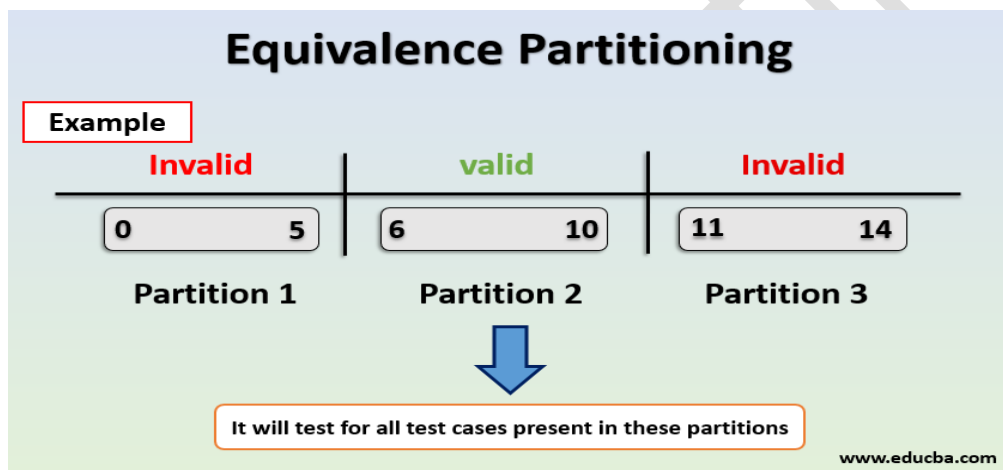
Boundary-value analysis is a software testing technique in which tests are designed to include representatives of boundary values in a range. This is the part of Black Box Testing Methods.

4. What is Equivalence partitioning testing?

This is the method of black box testing that is used to divide the long-range inputs into equivalence partitions and pick one representative from the partition for checking the whole range.

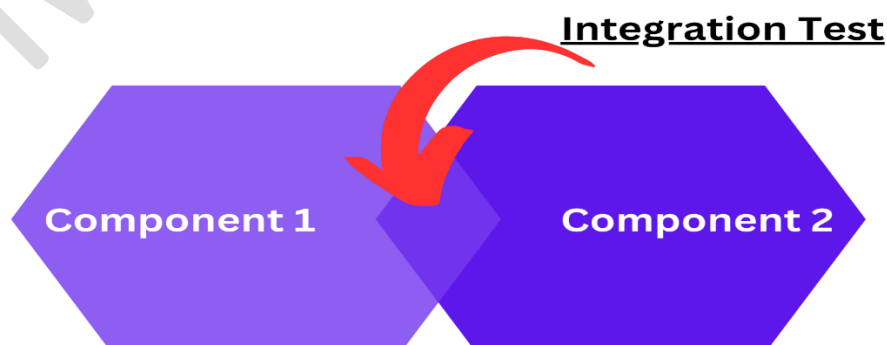
Ex. -Input field for:

- Partition 1: 0-5
- Partition 2: 6-10
- Partition 3: 11-14



5. What is Integration testing?

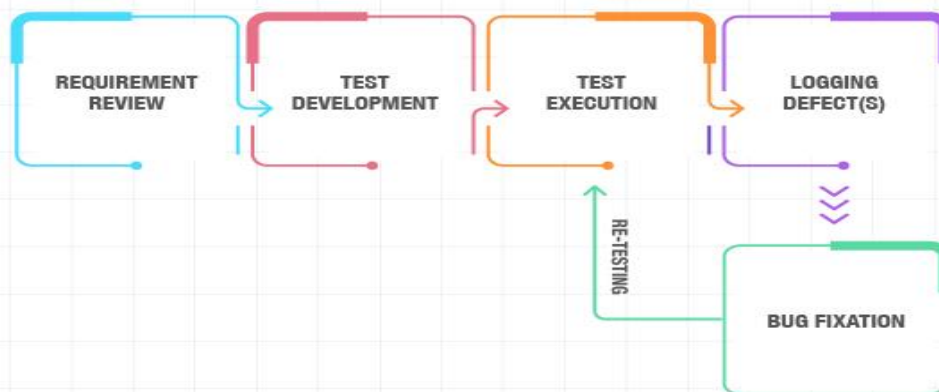
Integration Testing is a level of the software testing process where individual units are combined and tested as a group.



6. What is Alpha testing?

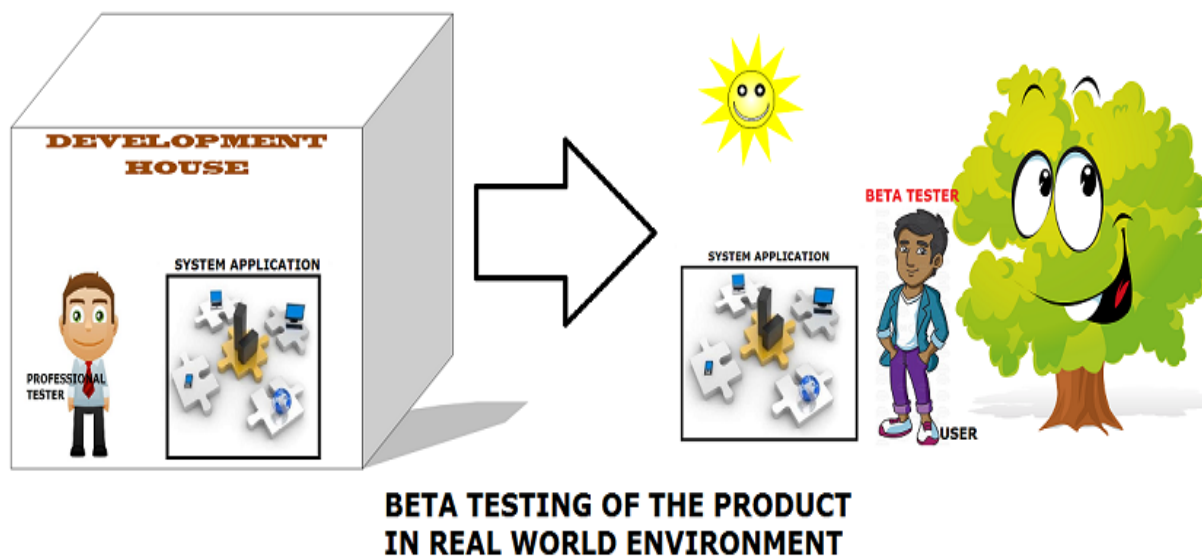
Alpha testing is a critical phase of software testing conducted by the development or QA team before beta testing. The purpose of alpha testing is to identify and resolve critical bugs and issues in the software before it is released to the public.

Alpha Testing



7. What is beta testing?

Beta testing is the last phase of the testing process before a product is released. The purpose of this phase is to evaluate the level of customer satisfaction with the product.



8.What is component testing?

Component testing, also known as unit testing or module testing, is a level of software testing that focuses on verifying the individual components or units of a system.

9.What is functional system testing?

Functional system testing meets the functional requirements and works as expected. It involves testing the system's functionality, user interactions, and business processes from start to finish.

10.What is Non-Functional Testing?

Non-functional testing is a type of software testing that evaluates the software's characteristics, properties, and behaviors that are not related to its functional requirements.

11.What is GUI Testing?

GUI testing is a software testing technique that checks the Graphical User Interface of the software application to ensure the application's functionality and features meet the business requirements. Some of the common things that are checked in GUI testing include like Size, position and width of the images, Error messages, Sections on screen, Font and color of the text, Alignment of the text and images etc.

12.What is Ad-hoc testing?

Ad-hoc testing is informal testing technique type. Main aim of this testing is find the defect with random checking and brake the system.

Type of ad-hoc testing.

Buddy testing

Pair tasting

Monkey testing

13.What is load testing?

This a type of performance testing, that determines how well a system, software product or application performs under real-life load conditions.

Stability + response time + applying load (app will withstand with designed no. of users)

14.What is stress Testing?

Stress testing is a type of software testing that helps to ensure the stability and reliability of a system. tress testing is performed until the system fails or crashes, which helps to identify the weak points and vulnerabilities of the system.

15.What is white box testing and list the types of white box testing?

White box testing is approach in which testing is based on an analysis of the internal structure. White box testing analyzes the internal code, internal design and working functionality. White box testing is also known as transparent testing an open ox testing.

There are three methods of black box testing:

1. **Statement coverage:** Statement coverage also known as line coverage and segment coverage. This method covers only true condition
2. **Decision/Branch coverage:** This method covers the both conditions true and false.
3. **Condition coverage:** Test all conditions (true/false) in each decision point.

16.What is black box testing? What are the different black box testing techniques?

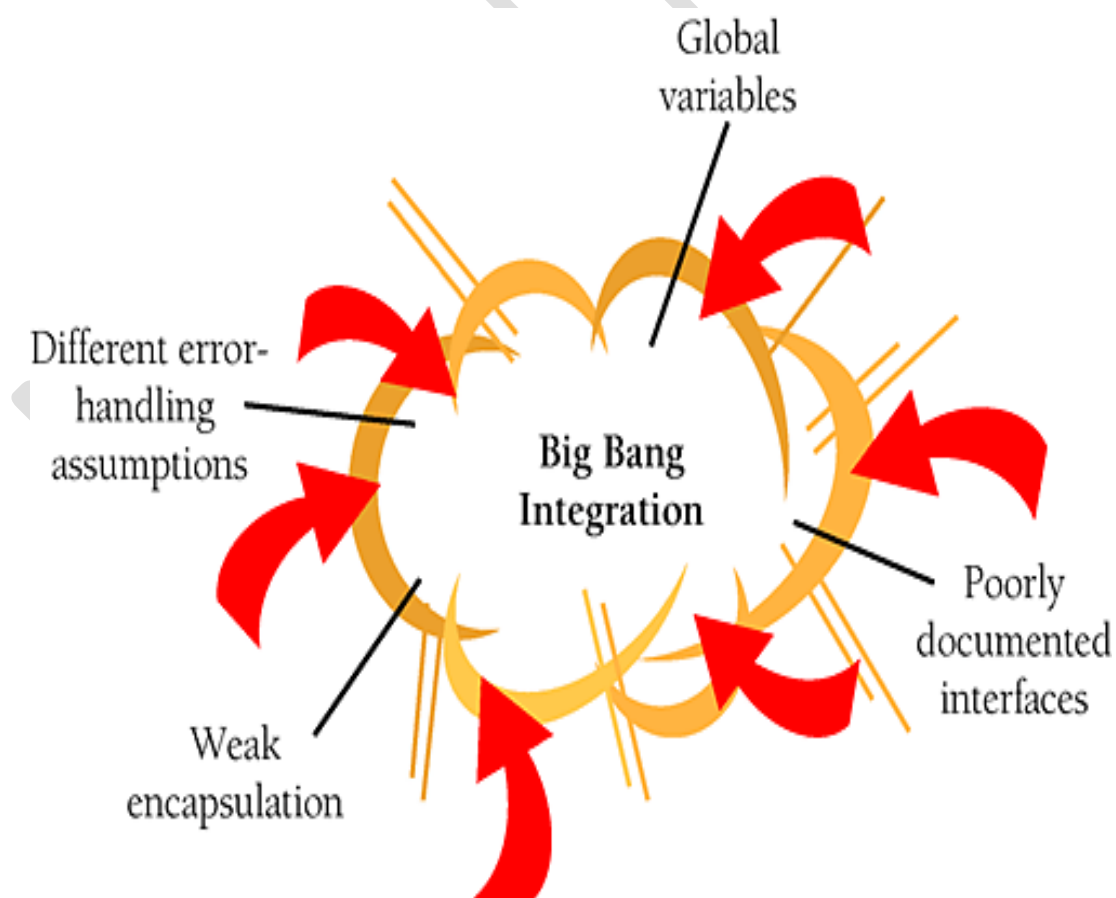
Black box testing, to check the application without having any knowledge of the internal part / source code. In black-box testing the tester is concentrating on what the software does, not how it does it.

There are four methods of black box testing.

1. **Equivalence partitioning (EP):** It is a method to drive the long-range inputs into equivalence partition and pick one representative from the partition for checking the whole range.
2. **Boundary value analysis:** To check the boundary values with valid range or invalid range.
3. **Decision tables:** Boolean expressions: true or false
Decision table shows the relationship between inputs and possible outputs are mapped together.
4. **State transaction/transaction testing:** To test the system by all given transaction stored in finite state machine. Any system where you get different outputs for the same input.

17.Mention what big-bang testing is?

That is a part of Integration Testing. Big Bang testing is a testing approach where all individual components or modules of a software application are tested together in a single.



18.What is the purpose of exit criteria?

The main purpose of exit criteria is all condition must be full fill before leaving the project. Ensure that the project has met its objectives and requirements.

19.When should "Regression Testing" be performed?

Regression testing should be performed in the following scenarios:

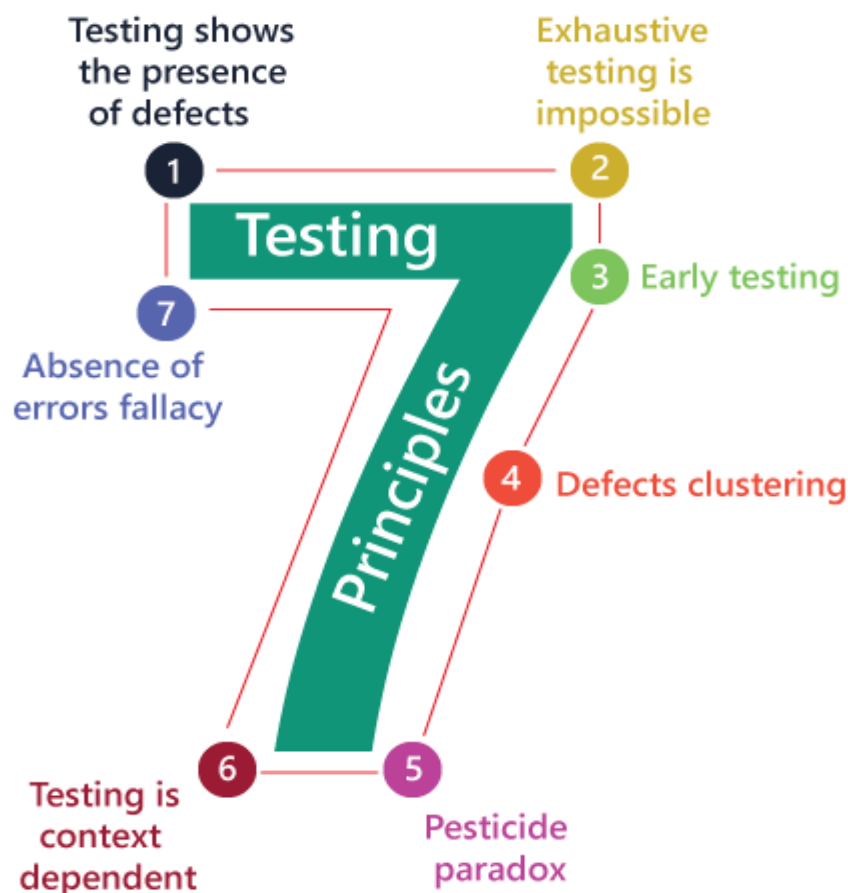
- When new functionality is added to the system
- When some defect is identified in the software and the code is debugged to fix it
- When the code is modified to optimize its working
- Whenever the production code is modified
- When a website's new feature is added
- When a bug is fixed by developers
- When there is an update in the database from one software to another

20.What is 7 key principles? Explain in detail.

The 7 Key Principles of Software Testing are fundamental guidelines that ensure effective testing and quality assurance. These principles are applied in the software testing industry. Here's a detailed explanation of each principles.

- **Testing Shows the Presence of Defects:** Testing can only expose the existence of defects. This principle said that testing is not a proof of correctness, but rather a way to identify errors.
- **Exhaustive Testing is Impossible:** It's impossible to test every possible scenario, input, or condition. It's time consuming.

- **Early Testing:** The earlier you test, the cheaper it is to fix defects. This principle highlights the importance of testing early in the development cycle to reduce costs and improve quality.
- **Defects Clustering:** A few specific areas of the software have a high concentration of bugs, defects and error.
- **The Pesticide paradox:** Repeatedly using the same test cases can lead to diminishing returns.
- **Testing is Context-Dependent:** Testing strategies and techniques vary depending on the project, technology, and requirements
- **Absence of errors is a fallacy:** Even if no defects are found, it doesn't mean the software is error-free. Defect free software is not as important as meets user requirement important.



21. Difference between QA v/s QC v/s Tester

| Quality Assurance | Quality Control | Tester |
|--|--|--|
| Focused on preventing defect and ensuring quality throughout the software development process. | Focuses on detecting and correcting defects in the software product. | Focuses on actual testing. |
| Quality Assurance (QA) is a process-oriented approach to ensuring that a product or service meets a set of quality standards | Quality Control (QC) is a product-oriented approach that focuses on identifying and fixing defects in the final product. | It includes activities that ensure the identification of bugs/error/defects in a software. |
| Involves testing, inspection, and validation to ensure the software meets the required standards. | Involves planning, designing, and implementing processes and procedures to ensure quality. | Product-oriented activities. |
| QA is process oriented | QC is product oriented. | Product-oriented activities. |
| QA is a managerial tool. | QC is a corrective tool. | |

22. Difference between Smoke and Sanity?

| Smoke testing | Sanity testing |
|---|---|
| To check the stability of the system. | To check the rationality of the system. |
| Smoke testing is scripted, documented, well-planned according to STLC | Sanity testing is unscripted, not documented, not well-planned. |
| This is the subset of acceptance testing | This is the subset of the regression testing. |
| That is performed by the developer & tester. | That is performed by tester only. |
| Ex. -To check the login functionality works, and the application doesn't crash or produce errors. | Ex. -To check the login functionality to work as expected with a focus on the logical flow and expected behavior. |

23. Difference between verification and Validation

| Verification | Validation |
|--|--|
| Verification does not involve code execution | Validation involves code execution. |
| The verifying process includes checking documents, design, code, and program | Validation is a dynamic mechanism of testing and validating the actual product |
| It finds bugs early in the development cycle | It can find bug that the verification process can't find the bug |
| It comes before validation | It comes after verification |

24. Explain types of Performance testing.

Performance testing is a type of software testing that evaluates the performance of a system or application under various conditions, such as:

- **Load:** Testing with a large number of users or transactions
- **Stress:** Testing beyond normal limits to identify breaking points
- **Endurance:** Testing for prolonged periods to evaluate stability
- **Scalability:** Testing with increased load or users to evaluate performance

25. What is Error, Defect, Bug and failure?

Error: A mistake in code/incorrect value

Defect: When the tester finds the error, it is called a defect.

Bug: When defect is accepted by developer or developer team, defect is called a bug.

Failure: Software failure refers to a situation where a software system or application does not perform as expected, does not meet user requirement.

26.Explain the difference between Functional testing and Non-Functional testing

| Functional testing | Non-Functional testing |
|--|---|
| Ensure the software performs its intended functions correctly. | Evaluate the software's performance, security, usability, and other non-functional aspects. |
| Easy to do manual & automation with Functional Testing. | Tough to do manual testing with Non-functional testing. |
| First to execute Functional. | Non-functional should be executed after Functional testing. |
| Functional testing describes what the product does. | Nonfunctional testing describes how good the product works. |

27.To create HLR & Test-Case of (Instagram, Facebook) only first page.

| |
|---|
| HLR & Test-Case of Facebook → click here |
| HLR & Test-Case of Instagram → click here |

28.What is the difference between the STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle)?

The STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle) are both crucial processes in software development, but they serve different purposes and focus on different aspects of the software development and testing process. Here are the key differences between STLC and SDLC:

SDLC (Software Development Life Cycle):

1. **Purpose:** SDLC is a process used by software development teams to plan, design, develop, test, and deploy software applications.
2. **Focus:** It focuses on the entire software development process, from the initial concept and requirements gathering to the final deployment and maintenance phases.
3. **Phases:** SDLC typically consists of phases such as:
 - **Requirements Gathering:** Gathering and documenting software requirements from stakeholders.
 - **Design:** Creating a detailed blueprint or design of the software architecture.
 - **Development:** Writing code and implementing the software based on the design.
 - **Testing:** Testing the software to identify and fix defects or issues.
 - **Deployment:** Releasing the software to users or customers.
 - **Maintenance:** Making updates, fixing bugs, and enhancing the software as needed after deployment.
4. **Involvement:** SDLC involves various stakeholders including developers, testers, project managers, business analysts, and end-users.
5. **Goal:** The primary goal of SDLC is to produce high-quality software that meets stakeholder requirements, within budget and time constraints.

STLC (Software Testing Life Cycle):

1. **Purpose:** STLC is a subset of the overall SDLC, specifically focusing on the testing activities throughout the software development process.
2. **Focus:** It focuses on planning, designing, executing, and managing the testing phase of the software development.
3. **Phases:** STLC typically includes phases such as:
 - **Requirement Analysis:** Analyzing the requirements from a testing perspective.
 - **Test Planning:** Creating a test plan that outlines the testing strategy, scope, resources, and schedule.
 - **Test Design:** Designing test cases and test scripts based on requirements and design documents.
 - **Test Execution:** Executing test cases, reporting defects, and retesting fixed defects.

- **Test Closure:** Evaluating test completion criteria, preparing test closure reports, and archiving test ware.
- 4. **Involvement:** STLC primarily involves testers, although collaboration with developers and other stakeholders is also necessary.
- 5. **Goal:** The goal of STLC is to ensure that the software meets quality standards, functions correctly according to specifications, and is ready for deployment.

29.What is the difference between test scenarios, test cases, and test script?

| Test Scenario | Test Case | Test Script |
|---|--|--|
| Is any functionality that can be tested. | Is a set of action executed to verify particular feature or functionality. | Is a set of instructions to test an app automatically. |
| Helps to test end to end functionality in an Agile Way. | Helps in executing testing an app. | Helps to test specific things repeatedly. |
| Includes an end-to-end functionality to be tested. | Includes test step, data, expected result for testing. | Includes different commands to develop a script. |
| Allow quickly assessing the testing scope. | Allow detecting error and defects. | Allowing carrying out an automatic execution of test case. |

30.Explain what Test Plan is? What is the information that should be covered.

A Test Plan is a detailed document outlining the approach, scope, and timeline for testing a software application or system. It serves as a guide for the testing process, ensuring that the testing is thorough, efficient, and effective.

A comprehensive Test Plan should cover the following information:

1. **Test Objectives:** Clearly state the goals and objectives of testing.
2. **Scope:** Define what is included and excluded from testing.
3. **Test Environment:** Describe the hardware, software, and network settings for testing.
4. **Test Approach:** Outline the testing methodology, techniques, and tools.
5. **Test Cases:** List the specific test scenarios and expected results.

- 6. Test Data:** Identify the data required for testing and how it will be obtained.
- 7. Test Schedule:** Provide a timeline for testing, including milestones and deadlines.
- 8. Resources:** Identify the personnel, equipment, and budget required for testing.
- 9. Risks and Assumptions:** Document potential risks and assumptions made during testing.
- 10. Test Deliverables:** Define the outputs and artifacts expected from testing, such as test reports and defect logs.
- 11. Test Criteria:** Establish the criteria for determining when testing is complete and successful.

31.To create HLR & Test-Case of Web-Based (WhatsApp-web, Instagram-web)

| | | |
|--------------------------------------|---|----------------------------|
| 1. HLR & Test-Case of WhatsApp-web | → | click here |
| 2. HLR & Test-Case of Instagram-web | → | click here |
| 3. HLR & Test-Case of Art of Testing | → | click here |

32.Write a scenario of only WhatsApp chat messages

| | | |
|----------------------|---|----------------------------|
| Scenario of WhatsApp | → | click here |
|----------------------|---|----------------------------|

33.Write a Scenario of Pen

| | | |
|---|---|----------------------------|
| Positive and Negative Scenario of the Pen | → | click here |
|---|---|----------------------------|

34.Write a Scenario of Pen Stand

| | | |
|---|---|----------------------------|
| Negative and Positive Scenario of the Pen Stand | → | click here |
|---|---|----------------------------|

35.Write a Scenario of Door

| | | |
|--|---|----------------------------|
| Negative and Positive Scenario of the Door | → | click here |
|--|---|----------------------------|

36.Write a Scenario of ATM

| | | |
|---|---|----------------------------|
| Negative and Positive Scenario of the ATM | → | click here |
|---|---|----------------------------|

37. Write a scenario of Microwave Owen

Negative and Positive Scenario of the Microwave Owen → [click here](#)

38. Write a scenario of Coffee vending Machine

Negative and Positive Scenario of the coffee vending machine → [click here](#)

39. What is the procedure for GUI Testing?

GUI testing involves checking the screens with the controls like menus, buttons, icons, and all types of bars – tool bar, menu bar, dialog boxes and windows etc.

Font size, style, & readable font, image resolution & clarity, size,
Error message window, color combination

Approach of GUI Testing

MANUAL BASED TESTING
RECORD AND REPLAY
MODEL BASED TESTING

40. Write a scenario of chair

Negative and Positive Scenario of the Chair → [click here](#)

41. To Create Scenario (Positive & Negative)

1. For Gmail (Receiving Gmail) → [click here](#)
2. For Online shopping to buy product (flip-kart) → [click here](#)

42. Write a Scenario of Wrist Watch

Negative and Positive Scenario of Wrist Watch → [click here](#)

43. Write a Scenario of Lift (Elevator)

Negative and Positive Scenario of Lift → [click here](#)

44. Write a Scenario of WhatsApp Group (generate group)

Negative and Positive Scenario WhatsApp Group → [Click here](#)