Module–2(Manual Testing)

1.**What is Exploratory Testing?**

- Exploratory testing is a software testing technique that involves simultaneously learning, designing, and executing tests without a script. It's an agile technique that allows testers to be creative and adapt based on their observations of the system and user behaviours.
- Exploratory testing is an approach to software testing that is often described as simultaneous learning, test design, and execution. It focuses on discovery and relies on the guidance of the individual tester to uncover defects that are not easily covered in the scope of other tests.

2.**What is traceability matrix?**

- Traceability matrix is a table type document that is used in the development of software application to trace requirements. It can be used for both forward (from Requirements to Design or Coding) and backward (from Coding to Requirements) tracing. It is also known as Requirement Traceability Matrix (RTM) or Cross Reference Matrix (CRM).
- It is prepared before the test execution process to make sure that every requirement is covered in the form of a Test case so that we don't miss out any testing. In the RTM document, we map all the requirements and corresponding test cases to ensure that we have written all the test cases for each condition.

3.**What is Boundary value testing?**

- Boundary Value Analysis is based on testing the boundary values of valid and invalid partitions. The behavior at the edge of the equivalence partition is more likely to be incorrect than the behavior within the partition, so boundaries are an area where testing is likely to yield defects.

4.**What is Equivalence partitioning testing?**

- Equivalence partitioning is the process of defining the optimum number of tests by: Reviewing documents such as the Functional Design Specification and Detailed Design Specification, and identifying each input condition within a function, Selecting input data that is representative of all other data that would likely invoke the same process for that particular condition.

5.**What is Integration testing?**

- Integration Testing - Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems Integration Testing is a level of the software testing process where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.

6.**What determines the level of risk?**

- Risk considerations can include: financial implication of software being released that is un-tested (support costs / possible legal action) software being delivered late to market potential loss of Life (safety critical systems) potential loss of face

7.What is Alpha testing?

- It is always performed by the developers at the software development site. Sometimes it is also performed by Independent Testing Team. It is conducted for the software application and project. It is always performed in Virtual Environment. It is always performed within the organization. It is the form of Acceptance Testing

8.**What is beta testing?**

- Beta Testing (field testing) is performed and carried out by users or you can say people at their own locations and site using customer data. It is only a kind of Black Box Testing.

9.**What is component testing?**

- Component(Unit) – A minimal software item that can be tested in isolation. It means “A unit is the smallest testable part of software.”Component Testing – The testing of individual software components. Unit Testing is a level of the software testing process where individual units/components of a software/system are tested. The purpose is to validate that each unit of the software performs as designed

10**.What is functional system testing?**

- Functional testing is a type of software testing that validates the functionality of an application to ensure it behaves as expected. The main objective is to verify that the software performs according to its specified requirements.

11.**What is Non-Functional Testing?**

- Non-functional testing evaluates aspects of a software application that are not related to specific behaviors or functionalities. Instead, it focuses on the performance, usability, reliability, scalability, and other quality attributes of the system.

12.**What is GUI Testing?**

- GUI Testing is a type of software testing that focuses on evaluating the graphical elements of an application to ensure they function correctly and provide a positive user experience. This includes testing all visual components like buttons, icons, menus, text fields, checkboxes, and other elements that users interact with.

13.**What is Adhoc testing?**

- Ad hoc testing is an informal, unstructured testing approach where testers explore the application without following any predefined test cases or plans. The main goal of ad hoc testing is to find defects that might not be uncovered through formal testing methods.

14.**What is load testing?**

- Load testing is a type of performance testing that evaluates how a system behaves under expected and peak load conditions. It involves simulating a specific number of users or transactions to determine the application's response time, stability, and resource usage, ensuring it can handle anticipated traffic without performance degradation.

15.**What is stress Testing?**

- Stress testing is a performance testing technique that evaluates how a system behaves under extreme conditions beyond its normal operational capacity. The goal is to identify breaking points, assess system stability, and observe how it recovers from failure, ensuring it can handle unexpected spikes in load or stress.

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

- White box testing, also known as clear box testing, glass box testing, or structural testing, is a software testing method that involves examining the internal structure, design, and implementation of the software being tested.
- Types of White box testing:-

1. Unit Testing
2. Integration Testing
3. Code Coverage Testing
4. Statement Coverage
5. Branch Coverage
6. Function Coverage
7. Control Flow Testing
8. Data Flow Testing
9. Mutation Testing
10. Security Testing

17.**What is black box testing? What are the different black box testing techniques?**

- Black box testing is a software testing method that focuses on evaluating the functionality of an application without any knowledge of its internal code, structure, or implementation. In black box testing, testers assess the software based on its inputs and expected outputs, validating that the system behaves as specified in the requirements.

- The types of black box testing are:

2. Functional Testing
3. Non-Functional Testing
4. Regression Testing
5. Smoke Testing
6. Sanity Testing
7. Interface Testing
8. User Acceptance Testing (UAT)

18.Mention what are the categories of defects?

- the common categories of defects are:

1. Functional Defects
2. Performance Defects
3. Usability Defects
4. Compatibility Defects
5. Security Defects
6. Reliability Defects
7. Interface Defects
8. Localization Defects
9. Boundary Defects
10. Data Handling Defects

1**9.Mention what big bang testing is?**

- Big Bang Testing is an integration testing approach where all components or modules of a system are combined and tested as a whole at once, rather than integrating and testing them step-by-step. In this method, testing is typically performed after all parts of the system are developed, and then the system is evaluated in its entirety.

20.What is the purpose of exit criteria?

- The purpose of exit criteria in software testing is to define the conditions that must be met before testing activities can be considered complete and the software can move to the next phase or be released. Exit criteria help ensure that testing has adequately covered the application, and that it meets the required quality standards.

21.When should "Regression Testing" be performed?

- Regression testing should be carried out:

1. when the system is stable and the system or the environment changes
2. when testing bug-fix releases as part of the maintenance phase

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

1. Testing Shows Presence of Defects: Testing can only prove that defects are present in a system, not that they are absent. Even if no defects are found, this does not imply the software is error-free.

2. Exhaustive Testing is Impossible: It’s impractical to test all possible scenarios due to the vast number of inputs, outputs, and combinations. Instead, prioritized testing based on risk and critical functionalities is used.

3. Early Testing: The earlier testing begins in the software development lifecycle, the more time and cost savings can be realized. Early testing helps catch defects when they are cheaper and easier to fix.
4. Defect Clustering: Most defects are often found in a small number of modules or areas of the application. This principle suggests focusing testing efforts on these high-risk areas.
5. Pesticide Paradox: Repeating the same tests will not find new defects. Test cases need to be reviewed and updated regularly to uncover new defects and cover different scenarios.
6. Testing is Context-Dependent: Different types of software require different testing approaches based on their purpose and risk level. For instance, testing techniques for financial software differ from those for mobile games.
7. Absence of Errors Fallacy: Just because an application has no known defects does not mean it meets the user's needs or requirements. Testing must ensure the product also fulfills user expectations.

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

Quality Assurance (QA)	Quality Control (QC)	Tester
Focuses on improving processes to prevent defects	Involves identifying defects in the final product	Responsible for executing tests and identifying defects
Proactive and process-oriented	Reactive and product-oriented	Operates in both QA and QC activities
Ensures processes are efficient and error-free	Ensures product meets quality standards	Ensures the software meets user requirements and is defect-free
Process audits, standards, and methodologies	Inspection, testing, and defect reporting	Test execution, bug reporting, and verification

24.Difference between Smoke and Sanity?

Smoke Testing	Sanity Testing
Verifies basic functionality and stability	Validates specific bug fixes or changes
Broad and shallow; covers critical features	Narrow and deep; focuses on specific components
After a new build or major release	After minor updates or bug fixes
Short, usually the first test after a build	Short, but more focused on recent changes
Determines if build is stable enough for further testing	Determines if specific components are functioning correctly

25.Difference between verification and Validation

Verification	Validation
Ensures the product is being built correctly	Ensures the built product meets user needs
Process-oriented	Product-oriented
Reviews, inspections, walkthroughs	Testing, user acceptance testing
Conducted during development stages	Conducted after development, before release
To ensure compliance with design specifications	To ensure the final product meets requirements

26.Explain types of Performance testing.

1. Load Testing: Determines how the system performs under expected user load conditions to ensure it can handle anticipated traffic.
2. Stress Testing: Evaluates how the system behaves under extreme load conditions beyond normal operational capacity, identifying its breaking points.
3. Scalability Testing: Assesses the system's ability to scale up or down based on changing load conditions, ensuring it can accommodate varying levels of users or transactions.
4. Volume Testing: Tests the system's ability to handle large volumes of data, ensuring that data processing and storage do not degrade performance.
5. Endurance Testing (Soak Testing): Evaluates the system's performance over an extended period under sustained load, identifying memory leaks and performance degradation over time.

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

- Error: A mistake in the code made by the developer, which may lead to incorrect logic or calculation.
- Defect: A flaw in the software that causes it to behave unexpectedly, identified during testing.
- Bug: A commonly used term for a defect; typically refers to an issue found during testing or use.
- Failure: Occurs when the software does not perform as expected due to a defect being encountered during execution.

28.Difference between Priority and Severity

Priority	Severity
Indicates how soon a defect should be fixed	Indicates the impact level of the defect
Business needs and timelines	Functional impact on the system
Handled by Project manager or client	Handled by Tester or developer

High priority but low severity for a typo on the homepage	High severity but low priority for a rarely used crash issue

29.What is Bug Life Cycle?

- New: A bug is identified and logged.
- Assigned: The bug is assigned to a developer for resolution.
- Open: The developer starts analyzing and working on the bug.
- Fixed: The bug is resolved by the developer.
- Retest: The tester rechecks the software to verify the fix.
- Verified: The fix is successful, and the bug is marked as verified.
- Reopen: If the bug persists, it is reopened for further investigation.
- Closed: If the bug is resolved and no longer present, it is marked as closed.
- Deferred: The bug is postponed for fixing in a future release.

30.Explain the difference between Functional testing and Non Functional testing

Functional Testing	Non-Functional Testing
Validates the software's functionality against requirements	Evaluates non-functional aspects like performance and usability
User actions, inputs, and expected outputs	Quality attributes like reliability and security
Black box, white box, regression	Load, stress, scalability
Verifying login functionality works as required	Measuring response time under high user load

31.To create HLR & TestCase of

1)(Instagram , Facebook) only first page

https://docs.google.com/spreadsheets/d/1754b_So3W0pVt6ZWhCw760iNjW_W9z34/edit?usp=drive_link&ouid=104762120838427984312&rtpof=true&sd=true

2) Facebook Login Page : https://www.facebook

https://docs.google.com/spreadsheets/d/1FkQ6VzVv2v_TxlmGtxcwVQBunWNpUK-U/edit?usp=drive_link&ouid=104762120838427984312&rtpof=true&sd=true

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

Software Development Life Cycle (SDLC)	Software Testing Life Cycle (STLC)
A series of phases involved in developing software from initial concept to deployment and maintenance.	A series of phases involved in testing the software to ensure it meets quality standards and is defect-free.
To create and deliver a functional, high-quality	To identify defects and verify that the software

software product.	meets specified requirements.
Common phases include: - Requirement Gathering - System Design - Implementation (Coding) - Testing - Deployment - Maintenance	Common phases include: - Requirement Analysis - Test Planning - Test Case Development - Test Environment Setup - Test Execution - Test Closure
Deliverables include software requirements specification (SRS), system design documents, code, user manuals, and the final software product.	Deliverables include test plans, test cases, test scripts, defect reports, test summary reports, and test closure documents.
Provides the foundation for the software and its functionality. Testing is a part of the SDLC.	Focuses on ensuring the quality of the software product created during the SDLC. All activities are centered on verification and validation.

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

Test Scenarios	Test Cases	Test Scripts
High-level descriptions of what needs to be tested, focusing on end-to-end functionality.	Detailed documentation of specific inputs, execution conditions, and expected outcomes for testing a particular feature or functionality.	Automated or manual step-by-step instructions or code used to execute a test case.
Ensures comprehensive coverage by identifying major areas and paths to be tested.	Provides detailed guidance on how to test specific aspects of functionality, ensuring consistent execution	Automates or formalizes the process of executing test cases, often with repeatability and efficiency.
High-level; broader in scope, without specific steps or inputs.	Medium-level; includes detailed steps, inputs, and expected results.	Low-level; very specific instructions or code that directly interacts with the system under test.
"Verify user login functionality."	- Precondition: User is on the login page. - Step 1: Enter a valid username and password. - Expected Result: User is logged in successfully.	Code that automates logging in with different username-password combinations and verifies successful login or failure messages.
Useful for brainstorming testing coverage, often used during initial planning phases.	Used as a basis for executing tests, either manually or via automation, with clear pass/fail criteria.	Typically used for automation or for detailed instructions in manual tests, executed by testers or automated tools.

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

A Test Plan is a detailed document that outlines the overall strategy, scope, objectives, resources, schedule, and activities required for testing a software product. It serves as a blueprint for the testing process and is essential for ensuring that all testing efforts are organized, efficient, and aligned with project goals.

Information Covered in a Test Plan:

- Preparation of test plan/strategy document for various types of testing
- Test tool selection
- Test effort estimation
- Resource planning and determining roles and responsibilities.
- Training requirement

35.What is priority? ⌄

Priority is Relative and Business-Focused. Priority defines the order in which we should resolve a defect. Should we fix it now, or can it wait? This priority status is set by the tester to the developer mentioning the time frame to fix the defect. If high priority is mentioned then the developer has to fix it at the earliest. The priority status is set based on the customer requirements

36.What is severity? ⌄

Severity is absolute and Customer-Focused. It is the extent to which the defect can affect the software. In other words it defines the impact that a given defect has on the system.

37.Bug categories are…

The categories of software bugs:

1. Functional Bugs
2. Performance Bugs
3. Security Bugs
4. Usability Bugs
5. Compatibility Bugs
6. Syntax Bugs
7. Logical Bugs
8. Interface Bugs
9. Data Bugs
10. Integration Bugs

38.Advantage of Bugzila . ⌄

- The advantages of Buzilla are:-

  Advanced search capabilities
  E-mail Notifications
  Modify/file Bugs by e-mail
  Time tracking
  Strong security
  Customization

39.Difference between priority and severity

Priority	Severity
Impact of the defect on the system and users	Impact of the defect on the system and users
Business perspective on defect resolution	Technical aspect of the defect
High, Medium, Low	Critical, Major, Minor, Trivial
Graphical issue for a marketing launch (high priority)	Application crash (high severity)

40.What are the different Methodologies in Agile Development Model?

Here are the names of the different methodologies in the Agile development model:

1. Scrum
2. Kanban
3. Extreme Programming (XP)
4. Lean Software Development
5. Feature-Driven Development (FDD)
6. Crystal
7. Agile Unified Process (AUP)

41.Explain the difference between Authorization and Authentication in Web testing.

Authentication	Authorization
Verify the identity of the user	Determine what resources or actions the user can access
Involves checking credentials (e.g., username and password)	Involves checking user roles and permissions
Confirms who the user is	Defines what the user can do
Logging in to a web application	Granting access to specific pages or features based on user role

42.What are the common problems faced in Web testing?

Here are some common problems faced during web testing:

1. Cross-Browser Compatibility:
    - Issue: Different browsers (e.g., Chrome, Firefox, Safari, Edge) may render web pages differently, leading to inconsistencies in functionality and appearance.

    - Impact: Users may experience errors or suboptimal layouts depending on their browser choice.

2. Responsive Design:
    - Issue: Web applications need to work on a variety of devices and screen sizes (desktops, tablets, smartphones).
    - Impact: Failure to test responsiveness can result in poor user experience, such as elements being cut off or requiring excessive scrolling.
3. Dynamic Content:
    - Issue: Many web applications use dynamic content that changes based on user interactions or data from the server (e.g., AJAX calls).
    - Impact: This can lead to challenges in ensuring that all content is displayed correctly and that features work as intended during various states.
4. Security Vulnerabilities:
    - Issue: Web applications are often targets for security threats (e.g., SQL injection, cross-site scripting).
    - Impact: Insufficient security testing can lead to data breaches and compromise user information.
5. Performance Issues:
    - Issue: Websites can experience slow load times, especially under high traffic conditions.
    - Impact: Poor performance can lead to user frustration and increased bounce rates.
6. Integration with Third-Party Services:
    - Issue: Many web applications rely on third-party APIs and services (e.g., payment gateways, social media integration).
    - Impact: Changes or outages in these services can affect the functionality of the web application.
7. User Interface (UI) Consistency:
    - Issue: Maintaining a consistent look and feel across all pages can be challenging, especially in larger applications.
    - Impact: Inconsistent UI can confuse users and detract from the overall experience.
8. Accessibility Compliance:
    - Issue: Ensuring that the web application is accessible to users with disabilities (e.g., screen readers, keyboard navigation).
    - Impact: Non-compliance with accessibility standards can exclude a significant portion of users and lead to legal repercussions.
9. Session Management:
    - Issue: Managing user sessions, including timeouts and logout functionality.
    - Impact: Poor session management can lead to security risks and user inconvenience.
10. Data Validation and Error Handling:
    - Issue: Ensuring that data entered by users is validated correctly and that error messages are displayed appropriately.

- Impact: Inadequate validation can result in incorrect data being processed, while poor error handling can lead to user confusion.

11.Localization and Internationalization:

- Issue: Adapting the application for different languages and regions.
- Impact: Failure to properly localize can result in language issues and cultural insensitivity, affecting user engagement.

12.Automated Testing Challenges:

- Issue: While automated testing can improve efficiency, it may be difficult to set up and maintain.
- Impact: Tests can become outdated quickly, leading to missed issues if not regularly updated.

43. To create HLR and TestCase on this Link. https://artoftesting.com/

https://docs.google.com/spreadsheets/d/1VdCw6jEeYraSmxmMdiCnIfF7DIOg5gLp/edit?usp=drive_link&ouid=104762120838427984312&rtpof=true&sd=true

44.Write a scenario of a pen

https://docs.google.com/spreadsheets/d/1IdquCwcbdd98x4JiFDltHoq3ewo-NKoU/edit?usp=drive_link&ouid=104762120838427984312&rtpof=true&sd=true

45.Write a test scenario of only Whatsapp chat messages

https://docs.google.com/spreadsheets/d/1AiY0CaIKYvQAxp44_tEcBISUrcb3Wy90/edit?usp=drive_link&ouid=104762120838427984312&rtpof=true&sd=true

46.Write a Scenario of Pen Stand

https://docs.google.com/spreadsheets/d/1jvT_1aso7pZehoxXLXoI-5Q_hI5woF-h/edit?usp=drive_link&ouid=104762120838427984312&rtpof=true&sd=true

47.Write a Scenario of Door

https://docs.google.com/spreadsheets/d/1vsW7yAC6LjS4keQP3Vq5pm2gI8WhRaQB/edit?usp=drive_link&ouid=104762120838427984312&rtpof=true&sd=true

48.Write a Scenario of ATM

https://docs.google.com/spreadsheets/d/1u4sAv_YMUBrfmYTtteJ0A1CPHaJbKYG2/edit?usp=drive_link&ouid=104762120838427984312&rtpof=true&sd=true

49.When to used Usablity Testing?

Usability testing is a technique used to evaluate a product's user interface by observing real users as they interact with it. The goal is to identify any usability issues, understand user behavior, and gather feedback to improve the overall user experience, ensuring that the product is intuitive, efficient, and satisfying to use.

50.What is the procedure for GUI Testing?

The procedure of GUI testing is:-

Check all the GUI elements for size, position, width, length and acceptance of characters or numbers. For instance, you must be able to provide inputs to the input fields.
Check you can execute the intended functionality of the application using the GUI Check Error Messages are displayed correctly
Check for Clear demarcation of different sections on screen
Check Font used in application is readable
Check the alignment of the text is proper
Check the Color of the font and warning messages is aesthetically pleasing
Check that the images have good clarity
Check that the images are properly aligned
Check the positioning of GUI elements for different screen resolution

51.Write a scenario of Microwave Owen

https://docs.google.com/spreadsheets/d/12jklXc2F0bpO5GdTbbCLqNbQJlqA0yuT/edit?usp=drive_link&ouid=104762120838427984312&rtpof=true&sd=true

52.Write a scenario of Coffee vending Machine

https://docs.google.com/spreadsheets/d/1IgkyFzlyeX1HXLIadk4f7vGI1PmNhije/edit?usp=drive_link&ouid=104762120838427984312&rtpof=true&sd=true

53.Write a scenario of chair

https://docs.google.com/spreadsheets/d/19NcFWlqAgKUiT10YVMbH0cjH4Kz_3ENH/edit?usp=drive_link&ouid=104762120838427984312&rtpof=true&sd=true

54.To Create Scenario (Positive & Negative)

Online product shopping using flipkart

https://docs.google.com/spreadsheets/d/10es5smaJa2lTVzO1amPndnpDYP06l78a/edit?usp=drive_link&ouid=104762120838427984312&rtpof=true&sd=true

55.Write a Scenario of Wrist Watch

https://docs.google.com/spreadsheets/d/1OQbwFUM7RteR1A0wsrksO_etDdiErztO/edit?usp=drive_link&ouid=104762120838427984312&rtpof=true&sd=true

56.Write a Scenario of Lift

https://docs.google.com/spreadsheets/d/1EtGHX6QAZD47YakHQ0ldHp0B72Jo9NC2/edit?usp=drive_link&ouid=104762120838427984312&rtpof=true&sd=true

57.Write a Scenario of whatsapp Group (generate group)

https://docs.google.com/spreadsheets/d/13ikddFd9-euUbNmwdMyzWtl0dGtXj9SY/edit?usp=drive_link&ouid=104762120838427984312&rtpof=true&sd=true

58.Write a Scenario of Whatsapp payment

https://docs.google.com/spreadsheets/d/1awHR0S_vlbL3LcGP1Tvf470HdX-zLex2/edit?usp=drive_link&ouid=104762120838427984312&rtpof=true&sd=true