1. What is Exploratory Testing?

Ans. Exploratory testing is a software testing approach that is less formalized and more focused on creativity, intuition, and adaptability compared to traditional scripted testing methods. In exploratory testing, the tester actively explores the software, designs test cases, and executes tests simultaneously. The primary goal is to discover defects or issues in the software and gain a deeper understanding of its behaviour.

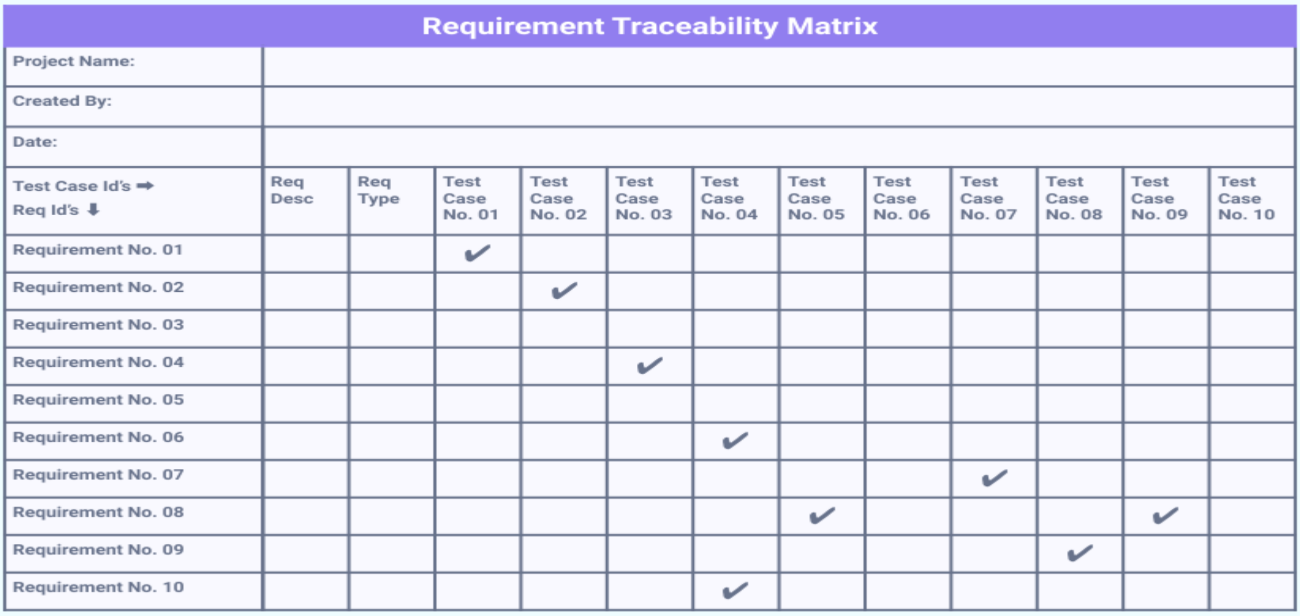
Exploratory testing is particularly useful in scenarios where requirements are unclear or constantly changing, and it allows testers to quickly adapt to evolving development processes. It complements scripted testing approaches and is often used in agile and iterative development environments. The results of exploratory testing can provide valuable feedback to improve test cases for future testing cycles.

2. What is traceability matrix?

Ans. A Traceability Matrix is a document used in software testing to ensure that all requirements have been tested and that there is complete test coverage across the entire software development life cycle. It establishes a link between requirements and the corresponding test cases, allowing for easy tracking and verification of the project's progress.

A Traceability Matrix acts as a bridge between requirements and test cases, ensuring that the testing process is aligned with the project's objectives and requirements are thoroughly validated through testing.

Also known as Requirement Traceability Matrix - RTM



Requirement Traceability Matrix

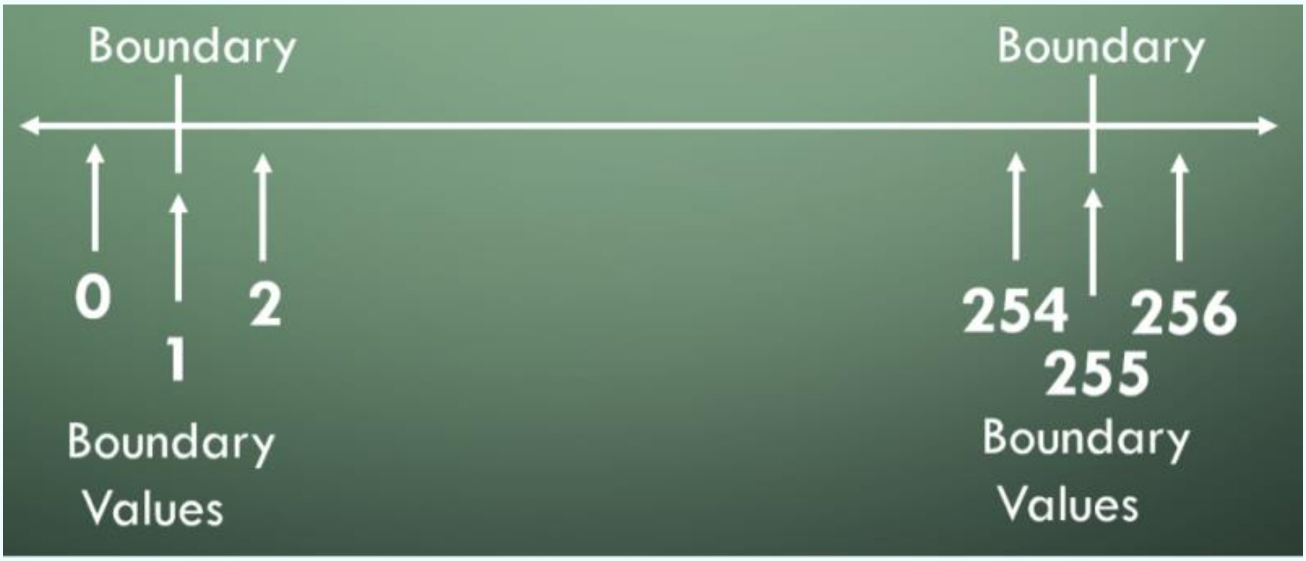
Project Name:												
Created By:												
Date:												
Test Case Id's ➡ Req Id's ⬇	Req Desc	Req Type	Test Case No. 01	Test Case No. 02	Test Case No. 03	Test Case No. 04	Test Case No. 05	Test Case No. 06	Test Case No. 07	Test Case No. 08	Test Case No. 09	Test Case No. 10
Requirement No. 01			✓									
Requirement No. 02				✓								
Requirement No. 03												
Requirement No. 04					✓							
Requirement No. 05												
Requirement No. 06						✓						
Requirement No. 07									✓			
Requirement No. 08							✓				✓	
Requirement No. 09										✓		
Requirement No. 10						✓						

3. What is Boundary value testing?

Ans. Boundary Value Testing is a software testing technique in which the values at the edges or boundaries of the input domain are tested to ensure that the application behaves as expected in these critical boundary conditions. The goal is to identify potential errors that may occur on or near the boundaries of valid input ranges.

The rationale behind boundary value testing is that errors often occur at the extremes of the input domain rather than in the middle. By testing values at the boundaries, testers aim to uncover issues related to data validation, range checking, and other boundary-specific behaviours.

By applying Boundary Value Testing, testers aim to uncover issues related to off-by-one errors, array bounds, input validation, and other scenarios where boundary conditions play a crucial role in system behaviour. This technique is widely used in unit testing, integration testing, and system testing to enhance the overall quality of software.



4. What is Equivalence partitioning testing?

Ans. Equivalence Partitioning is a software testing technique that divides the input space into partitions or groups of equivalent data, and then one representative value from each partition is chosen as a test case. The goal is to reduce the number of test cases while ensuring that each partition is tested, assuming that all values within a partition should behave in a similar way.

Equivalence Partitioning is particularly useful when dealing with large input domains where it is impractical to test every possible value. By identifying representative values from each equivalence class, testers can achieve good test coverage while minimizing the number of test cases, making it an efficient and effective testing technique.

5. What is Integration testing?

Ans. Integration testing is a level of software testing where individual units or components of a software system are combined and tested as a group. The purpose of integration testing is to verify that the interactions between these components work as intended when integrated into a larger system. It helps identify issues related to the interfaces, data flow, and communication between the integrated components.

Types of Integration Testing:

1. Bing Bang Integration Testing

2.Incremental Integration Testing

A) Top-Down Approach

B) Bottom-Up Approach

6. What determines the level of risk?

Ans. Risks are of two types

- Project Risks: - Ex. Project risk is Senior Team Member leaving the project abruptly
- Product Risk: - Ex. Product risks would be Flight Reservation system not installing in test environment.

7. What is Alpha testing?

Ans. Alpha testing is a type of software testing performed to identify and address issues within a software application before it is released to a broader audience. This testing phase is conducted by the internal development team, often in a controlled environment. The primary goal of alpha testing is to detect and rectify any defects or issues that may exist in the software's functionality, design, or usability. Alpha testing involves both white-box and black-box testing methods.

8. What is beta testing?

Ans. Beta testing is a type of software testing performed by a selected group of users, known as beta testers, before a product is released to the general public. The primary purpose of beta testing is to gather real-world feedback from users in diverse environments and uncover any issues or bugs that may not have been identified during earlier testing phases. It is performed in Real Time Environment. Beta testing involves black-box testing methods.

9. What is component testing?

Ans. Component testing, also known as Unit testing, Program Testing or Module testing, is a type of software testing in which individual components or modules of a software application are tested in isolation. The purpose of component testing is to ensure that each unit of the software, such as a function or a module, performs as intended. It is an essential part of the software development process and is typically conducted by developers during the development phase. Unit testing is performed by using the White Box Testing method.

10. What is functional system testing?

Ans. Functional system testing is a type of software testing that evaluates the overall functionality of a software application or system. It focuses on verifying that the software system behaves according to specified requirements and functions correctly in its intended environment. The user interface (UI) is a critical aspect of functional system testing.

11. What is Non-Functional Testing?

Ans. Non-functional testing is a type of software testing that assesses aspects of a system's performance and behaviour that are not related to specific functionalities or features.

Some key aspects of non-functional testing:

Performance, Load, Data volumes, Storage, Recovery, Usability, Stress

12. What is GUI Testing?

Ans. GUI testing, which stands for Graphical User Interface testing, is a type of software testing that focuses on verifying the graphical elements of a user interface. 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. This type of testing is particularly relevant for applications with graphical interfaces, such as desktop applications, web applications, and mobile applications.

13. What is Ad-hoc testing?

Ans. Ad-hoc testing is a type of software testing wherein experienced and good testers testing in random checks with the objective of breaking the system and uncovering defects. Ad-hoc testing is also called as Error Guessing.

Types of Ad-hoc testing:

- Buddy Testing: - Developer + Tester
- Pair testing: - Tester + Tester
- Monkey Testing: - Randomly test the product or application

14. What is load testing?

Ans. Load testing is a type of performance testing that involves system's performance under real-life load conditions and determine at what point the system's response time fails.

15. What is stress Testing?

Ans. Stress testing is a type of software testing that evaluates the stability and robustness of a system by subjecting it to extreme conditions or loads. The purpose of stress testing is to identify the system's breaking point and understand how it behaves under extreme stress. Stress testing is also known as endurance testing.

Types of Stress Testing

- Application Stress Testing
- Transactional Stress Testing
- Systemic Stress Testing
- Exploratory Stress Testing

Stress Testing Tools

Stress Tester
Neo Load
App Perfect

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

Ans. White-box testing, also known as clear-box testing, open box testing, glass-box testing, or structural testing, is a software testing technique that examines the internal structure and logic of a software application.

Types of white box testing

- Statement coverage
- Decision coverage
- Condition coverage

Key characteristics of white-box testing

- Knowledge of Internal Code
- Testing at the Code Level
- Coverage of Code Paths
- Unit Testing and Integration Testing
- Path Coverage Metrics
- Performed by Developer

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

Ans. Black-box testing is a software testing method that focuses on assessing the functionality of a software application without examining its internal code or implementation details. Specification-based testing technique is also known as 'black-box' or input/output driven testing techniques.

Black box testing techniques

- Equivalence partitioning
- Boundary value analysis
- Decision tables
- State transition testing
- Use-case Testing
- Other Black Box Testing

Key characteristics of black-box testing

- No Knowledge of Internal Code
- Tests Based on Specifications
- Focus on Input and Output
- Functional and Non-functional Testing

- System-Level Testing
- Testing from the User's Perspective
- Perform by Tester

18. Mention what are the categories of defects?

Ans. Here are some common categories of defects:

- Data Quality/Database Defects
- Critical Functionality Defects
- Functionality Defects
- Security Defects
- User Interface Defects

19. Mention what big bang testing is?

Ans. Testing technique which integrates individual program modules only when everything is ready. It is performed by the testing teams.

20. What is the purpose of exit criteria?

Ans. Purpose of exit criteria is to define when we STOP testing either at the:

- End of all testing – i.e. product Go Live
- End of phase of testing (e.g. hand over from System Test to UAT)

21. When should "Regression Testing" be performed?

Ans. Regression testing should be performed whenever there are changes made to the software, such as code modifications, bug fixes, enhancements, or new feature implementations.

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

Ans. 7 key principles provide fundamental guidance for testing activities and are essential for understanding the core concepts of software testing.

Here are the seven key principles along with explanations:

1. Testing Shows the Presence of Defects: - Testing helps reveal the presence of defects, providing information about the quality of the software.
2. Exhaustive Testing is Impossible: - It is practically impossible to test every Combination of input.
3. Early Testing: - The principle of early testing advocates for starting the testing process as early as possible in the software development life cycle.
4. Defect Clustering: - If You have any error, make sure it will not spread in your system.
5. The Pesticide Paradox: - Test cases need to evolve and be regularly reviewed and updated to identify different types of defects.
6. Testing is Context Dependent: - Every Website having a different way to check.
7. Absence of Errors Fallacy: - When Testing is Perfect but requirement does not match.

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

Ans.

QA	QC	Tester
QA is a proactive and process-oriented approach to ensure that the processes and methodologies used in the software development life cycle are designed and implemented effectively.	QC is a reactive and product-oriented approach that involves activities and techniques used to full fill quality requirements and identify defects in the product.	Tester is the testing the software and identification of bugs/error/defects in the Software.
It is a subset of Software Test Life Cycle (STLC).	QC can be considered as the subset of Quality Assurance.	Testing is the subset of Quality Control.

24. Difference between Smoke and Sanity?

Ans.

Smoke Testing	Sanity Testing
Smoke Testing is performed to ascertain	Sanity Testing is done to check the

that the critical functionalities of the program is working fine	new functionality / bugs have been fixed
This testing is performed by the developers or testers	Sanity testing is usually performed by testers
Smoke testing is usually documented or scripted	Sanity testing is usually not documented and is unscripted
Smoke testing is a subset of Regression testing	Sanity testing is a subset of Acceptance testing
Smoke testing is like General Health Check Up	Sanity Testing is like specialized health check up

25. Difference between verification and Validation

Ans.

Verification	**Validation**
Verification is concerned with evaluating work products at various stages of the development process to ensure that they meet the specified requirements.	Validation is concerned with evaluating the software system during or at the end of the development process to ensure that it satisfies the specified requirements in its operational environment.
Verification activities take place during the development phase, before the actual testing begins.	Validation activities take place during the testing phase, where the software is tested against the user requirements to ensure that it behaves as expected.
The main goal of verification is to confirm that the software is being developed according to the requirements and design specifications.	The main goal of validation is to ensure that the final product meets the customer's expectations and functions correctly in the real-world environment.
Inspections, reviews, walkthroughs, and other static methods are used in verification to check documents, code, and design models.	Dynamic testing methods such as unit testing, integration testing, system testing, and acceptance testing are used in validation to check the software's functionality and performance.

26. Explain types of Performance testing.

Ans. Performance testing is a type of software testing that is conducted to evaluate the speed, responsiveness, stability, and scalability of a software application under different conditions. There are several types of performance testing.

1. Load Testing: - To evaluate the system's behaviour under a specific expected load.
2. Stress Testing: - To evaluate the system's behaviour under extreme conditions or beyond its maximum capacity.
3. Endurance Testing: - To evaluate the system's performance over an extended period to ensure its stability and reliability under sustained use.
4. Scalability Testing: - To assess the system's ability to handle increased load by adding resources, such as hardware or network bandwidth.
5. Spike Testing: - To assess the system's performance under a sustained load over an extended period to identify performance issues that may arise with prolonged use.
6. Volume Testing: - To assess how well the application handles a large amount of data.

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

Ans. Error: - A mistake in coding is called error

Defect: - Error found by tester is called defect

Bug: - Defect accepted by development team then it is called bug

Failure: - Build does not meet the requirements then it is failure

28. Difference between Priority and Severity

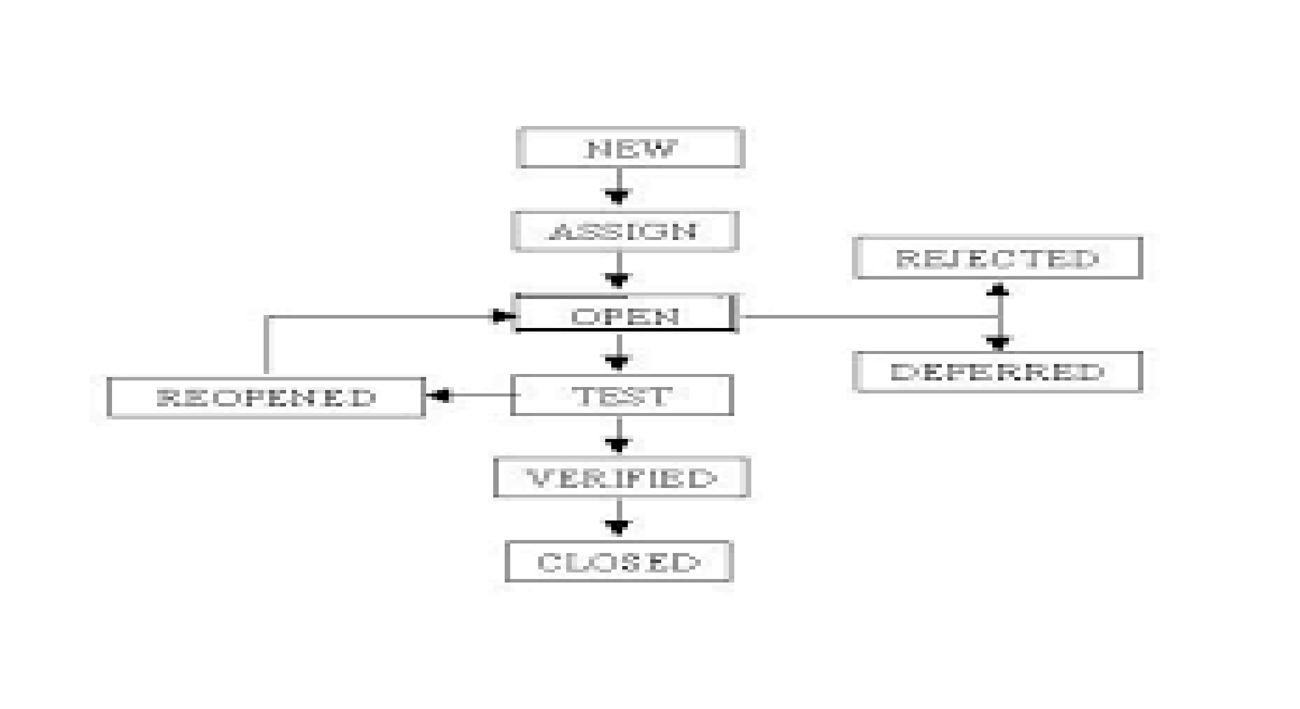
Ans.

Priority	**Severity**
Priority is about the business or project impact.	Severity is about the impact on the software functionality
Priority is more business-focused, considering project timelines and goals.	Severity is more technically focused, addressing the technical aspects of a defect
Priority is typically set by project managers or stakeholders.	Severity is often determined by testers or technical experts

Priority is Relative and Business-Focused.	Severity is absolute and Customer-Focused.

29. What is Bug Life Cycle?

Ans. The bug life cycle, also known as the defect life cycle, describes the various stages that a software bug goes through from the time it is identified to the time it is successfully fixed and verified.



The different phases of Bug life cycle are: -

1. New: - When a new defect is logged and posted for the first time. It is assigned a status as NEW.
2. Assigned: - The bug report is reviewed, and the responsibility for fixing it is assigned to a developer.
3. Open: - The developer begins investigating and fixing the bug.
4. Fixed: -The developer has implemented a fix for the bug.
5. Pending Retest: - The fixed code is ready for testing.
6. Retest: - The testing team retests the fixed code to ensure that the bug has been successfully resolved.
7. Closed: - The bug has been successfully fixed, retested, and verified. It is considered closed.
8. Reopen (Optional): - If the bug is found to persist after retesting, it may be reopened, and the cycle starts again.

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

Ans.

Functional testing	Non-Functional testing
Functional testing is concerned with verifying that the software functions as expected and that it meets the specified functional requirements.	Non-functional testing is concerned with attributes that are not related to specific behaviours or functions of the system. It focuses on qualities like performance, usability, reliability, and security.
The main purpose is to ensure that the software application behaves according to the documented design and requirements. It answers questions like "Does the software do what it's supposed to do?"	The primary purpose is to assess the overall system attributes and ensure that the software meets non-functional requirements, which are often related to user experience, system performance, and reliability.
Functional testing is concerned with the specific behaviours and functions of the software	Non-functional testing focuses on overall system attributes.
Functional testing is executed first	Non-functional testing should be performed after functional testing
Easy to do manual testing	Tough to do manual testing
Types of Functional tasting • Unit Testing • Smoke Testing • Sanity Testing • Integration Testing • White box testing • Black Box testing • User Acceptance testing • Regression Testing	Types of Nonfunctional testing • Performance Testing • Load Testing • Volume Testing • Stress Testing • Security Testing • Installation Testing • Penetration Testing • Compatibility Testing • Migration Testing

31. To create HLR & Test Case of

1. (Instagram, Facebook) only first page
2. Facebook Login Page: - https://www.facebook.com/

Ans. https://docs.google.com/spreadsheets/d/1kLRoAMWhYkmM0HJe0Z8a_v-4r2t8jSA2/edit?usp=sharing&ouid=106915935014132166214&rtpof=true&sd=true

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

Ans.

Test scenarios	Test cases	Test script
A test scenario is a high-level description	A test case is a detailed set of conditions	Test script is the actual implementation or set of instructions used to execute a test case

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

Ans. Test plan is serves as a blueprint for the testing process and provides guidance on how testing activities will be conducted.

The information that should be covered:

- Introduction to the Test Plan document
- Assumptions when testing the application
- List of test cases included in Testing the application
- List of features to be tested
- What sort of Approach to use when testing the software
- List of Deliverables that need to be tested
- The resources allocated for testing the application
- Any Risks involved during the testing process
- A Schedule of tasks and milestones as testing is started

34. What is priority?

Ans. Priority refers to the level of importance assigned to a defect or a test case. It helps in determining the order in which tests should be executed or defects should be fixed.

35. What is severity?

Ans. Severity refers to the impact that a defect or issue has on the functionality of the software or system. It helps in classifying the level of seriousness or the extent to which a particular problem can adversely affect the system.

36. Bug categories are…

Ans. Bug categories are: -

- Security,
- Database,
- Functionality (Critical/General),
- UI

37. Advantage of Bugzilla.

Ans. Advantage: -

- Bugzilla is an open-source tool, which means it is freely available for use, modification, and distribution
- Organizations can customize it to meet their specific requirements without incurring additional costs.
- Bugzilla has robust email integration features.
- Bugzilla provides a web-based interface, making it accessible from any device with a web browser.
- Bugzilla offers advanced search capabilities, allowing users to create complex queries to find specific issues.
- Bugzilla allows administrators to define different roles and permissions for users.
- Bugzilla has been in existence for a long time and has been used by numerous organizations worldwide.

38. Difference between priority and severity

Ans. Same as 28.

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

40. Explain the difference between Authorization and Authentication in Web testing. What are the common problems faced in Web testing.

Ans. Authorization: - Authorization, on the other hand, is the process of determining what actions a user is allowed to perform after they have been authenticated.

Authentication: - Authentication ensures that a user is who they claim to be before granting access to specific resources or functionalities on a website.

The common problems faced in Web testing: -

- Data Security
- Secure Transmission
- Cross-Browser Compatibility
- Performance Issues
- User Interface (UI) and User Experience (UX)
- Compatibility Testing
- Functionality Issues
- Integration Testing
- Scalability
- Data Integrity and Validation
- Session Management

41. To create HLR & Test Case of Web Based (WhatsApp web, Instagram)
    1. WhatsApp Web: https://web.whatsapp.com/
    2. Instagram Web: https://www.instagram.com/accounts/login/

Ans. https://docs.google.com/spreadsheets/d/1kLRoAMWhYkmM0HJe0Z8a_v-4r2t8jSA2/edit?usp=sharing&ouid=106915935014132166214&rtpof=true&sd=true

42. To create HLR and Test Case on this Link: https://artoftesting.com/

Ans. https://docs.google.com/spreadsheets/d/1kLRoAMWhYkmM0HJe0Z8a_v-4r2t8jSA2/edit?usp=sharing&ouid=106915935014132166214&rtpof=true&sd=true

43. Write a scenario of only WhatsApp chat messages

Ans. Test scenario of WhatsApp chat messages:

- Check that contains the entire chat list.
- Check the displayed on the group chat list.
- Check displays the name of all contacts on the chat window.
- Ensure the "Send" button is functional.
- Verify that the recipient receives the sent message.
- Check for a notification indicating the arrival of a new message.
- Observe that the recipient sees a typing indicator.
- Confirm that the typing indicator disappears when the message is sent.
- Ensure that emoticons or emojis are displayed correctly on both ends.
- Confirm that each message displays a timestamp indicating when it was sent.
- Check the user can send and receive documents.
- Check the user can send and receive photos.
- Check the user can send and receive videos.
- Confirm that the recipient can download and view the media.
- Check the user can send and receive audio.
- Check that the user can send and receive contacts.
- Check the user can send and receive Location.
- Check the user can send and receive GIFs.
- Check the user can send and receive Stickers
- Verify the user can delete text, video, audio, locations, and documents.
- Check the user can send recorded voice mail.
- Ensure that the voice message is received and can be played.
- Check the user can delete the entire chat history.
- Check the user can see contact details.
- Verify the user can share images, links, and documents from media
- Verify the user can search specific chat history using the search option.
- Check the user can video call.
- Check the user can voice call.
- Check the user can change the wallpaper.
- Check the users have options like Report, Block, Clear Chat, Export Chat, and Add Shortcut.

44. Write a Scenario of Pen

Ans. Test Scenario of Pen:

- Ensure the pen is in good condition
- Examine the pen for any physical defects or damages.
- Check that the pen cap is present and fits securely.

- Confirm that the cap is easy to remove and securely fits when placed back on.
- Ensure that the pen has sufficient ink
- Test the ink color to match the specified color if applicable.
- Verify that the pen writes smoothly without skipping or blotting.
- Test the pen on various surfaces such as plain paper, textured paper, and cardboard.
- Experiment with different writing pressures to observe variations in line thickness.
- Verify that the pen maintains consistent ink flow without drying out.
- Conduct durability tests such as dropping the pen from a small height.
- The pocket clip is sturdy and functional.

45. Write a Scenario of Pen Stand

Ans. Test Scenario of Pen Stand:

- Ensure the pen stand is in good condition.
- The pens stand is free from physical defects.
- The stand accommodates the maximum specified number of pens.
- Pens of different sizes and styles fit securely in the stand.
- Pen insertion and removal are smooth and easy.
- The pen stand remains stable when loaded with pens.
- The material of the stand is durable and does not show signs of wear.
- The pen stand has an aesthetically pleasing design.
- Pens are easily accessible from the stand.
- Customizable elements, if any, work as intended.
- The pen stand is easy to clean and maintain.

46. Write a Scenario of Door

Ans. Test Scenario of Door:

- Ensure the door is installed correctly.
- The door is free from physical defects.
- The door operates smoothly during opening and closing.
- The latch and lock mechanisms function correctly.
- The key (if applicable) operates the lock smoothly.
- Hinges are in good condition, and the door remains securely attached.
- The door seal (if applicable) provides effective weatherproofing.

- The door swings in the correct direction.
- Safety features are functional, and the door complies with safety standards.
- The door withstands pressure without warping.
- The door operates quietly without excessive noise.
- The door provides the intended level of security.
- Fire-rated doors comply with fire safety standards.

47. Write a Scenario of ATM

Ans. Test Scenario of ATM:

- Ensure the ATM is powered on and connected to the network.
- Verify that the ATM is stocked with sufficient cash.
- The ATM initializes without errors.
- Check whether the ATM provides all types of accounts for operations like Savings and the correct account.
- Card insertion and PIN entry are accepted without issues.
- Check whether the ATM provides all types of accounts for operations like Savings and the correct account.
- Verify whether the entered pin is encrypted or not.
- Check whether the ATM provides all types of accounts for operations like Savings and the correct account.
- Balance inquiry, cash withdrawal, deposit, and transfer transactions are processed accurately.
- Receipts contain accurate information.
- The card is returned promptly after each transaction.
- Security features prevent unauthorized access.
- Touchscreen (if applicable) is responsive.
- The ATM is accessible to users with disabilities.
- Transactions are processed within a reasonable timeframe.
- The ATM provides appropriate messages in case of network disruptions.
- Emergency procedures are effective.
- If the user enters more amounts than the account balance, then the user should get an error message.

48. When to used Usability Testing?

Ans. Usability testing is a versatile activity that can be employed at different stages of development to enhance the overall user experience and satisfaction with a software product.

49. What is the procedure for GUI Testing?

Ans. Here is a general procedure for GUI testing: -

- Understanding Requirements
- Identifying Test Scenarios
- Creating Test Cases
- Functional Testing
- Layout and Alignment
- Font and Text Verification
- Color and Style Verification
- Error Handling
- Usability Testing
- Compatibility Testing
- Performance Testing
- Accessibility Testing
- Cross-Browser Testing
- Localization Testing
- Documentation Verification
- Regression Testing
- Automation Testing
- Reporting and Tracking

50. Write a scenario of Microwave Owen

Ans. Test Scenario of Microwave Owen:

- Ensure the microwave oven is plugged in and properly connected.
- The display is illuminated and functional.
- The door closes securely, and the safety interlock is effective.
- Control panel buttons respond accurately.
- Time and power settings are displayed correctly.
- Cooking operations heat or cook food as expected.
- Pause and resume functions work seamlessly.
- The door opens easily after cooking.

- If applicable, the turntable rotates smoothly.
- The countdown timer alerts when it reaches zero.
- Sensor cooking features operate accurately.
- Child lock can be activated and deactivated.
- The exterior temperature remains within a safe range.
- The ventilation system operates effectively.
- Defrosting function (if applicable) is efficient.

51. Write a scenario of Coffee vending Machine

Ans. Test Scenario of Coffee vending Machine:

- User interface buttons or touch screen options function accurately.
- The cup dispenser reliably dispenses cups of the correct size.
- Ingredient levels are monitored, and low or empty levels are indicated.
- The machine accurately dispenses selected beverages.
- Customization options are reflected in the dispensed beverage.
- The hot water dispenser functions at the correct temperature.
- The grinding mechanism operates smoothly without issues.
- Dispensed beverage temperature is within a safe and enjoyable range.
- The payment system, if applicable, accepts payments accurately.
- Cleaning functions are efficient and do not affect beverage quality.
- The machine provides clear error messages and instructions.
- The machine operates in an energy-efficient manner.
- Users find the machine user-friendly and easy to operate.

52. Write a scenario of chair

Ans. Test Scenario of chair:

- Ensure the chair is assembled correctly.
- Verify that the chair is free from visible defects or damages.
- The chair is stable and does not wobble.
- The chair supports the specified weight capacity.
- The seat and backrest provide comfort during use.
- Armrests (if present) are sturdy and comfortable.
- Adjustable features operate smoothly.
- Chair materials show no signs of wear or deterioration.
- Casters (if present) roll smoothly without getting stuck.

- The chair operates quietly without squeaks or noises.
- Assembly is easy, and all components fit securely.
- Safety features, if any, operate as intended.
- The chair is easy to clean and maintain.
- The overall design and aesthetics meet expectations.
- Packaging is free from damage, and assembly instructions are included.

53. To Create Scenario (Positive & Negative)

    1. Facebook Chat on Mobile

- Positive test scenario:
    - Ensure users can log in successfully to the Facebook mobile app.
    - Confirm that users can access the chat feature without any issues.
    - Test the ability to send text messages to friends in the chat.
    - Verify that sent messages are delivered promptly and displayed correctly.
    - Test sending and receiving images, videos, and voice messages.
    - Confirm that media files are displayed correctly in the chat.
    - Verify that users can send stickers and emoticons.
    - Ensure that stickers are displayed properly on both ends.
    - Test the creation of group chats.
    - Confirm that users can add/remove members and customize group settings.
    - Verify that users receive timely notifications for new messages.
    - Test the display of message previews in notifications.
    - Ensure that read receipts are displayed when a message is read.
    - Confirm that users can enable/disable read receipts as per their preferences.
    - Test the initiation and reception of voice and video calls.
    - Ensure that voice and video calls are clear with minimal latency.
    - Verify that users can search for specific messages or contacts.
    - Confirm that search results are accurate and relevant.
    - Test the ability to view and respond to messages in offline mode.
    - Confirm that messages are synchronized once the device is back online.

- Negative test scenario:
    - Attempt to log in with incorrect credentials and verify appropriate error messages are displayed.
    - Test sending messages when the user has poor or no internet connectivity.

- Verify that appropriate error messages are displayed for failed message sending.
- Send corrupted media files and ensure the app handles them gracefully.
- Attempt to send media files exceeding the allowed size and verify error handling.
- Test adding a user who has blocked the initiator in a group chat.
- Verify that removing the last admin from a group triggers an appropriate warning.
- Disable notifications for the Facebook app and verify that users do not receive any notifications.
- Test scenarios where read receipts are not sent even after the message is read.
- Verify that turning off read receipts does not affect the display for the other user.
- Test making calls with poor network conditions and verify how the app handles call drops.
- Attempt to initiate a call with a user who has blocked the initiator.
- Test searching for a non-existent contact or message and verify the app's response.
- Attempt to search while offline and confirm that appropriate messages are displayed.
- Test responding to messages in offline mode and check for synchronization issues once online.
- Verify that sensitive actions (e.g., blocking a user) are restricted in offline mode.
- Test the application on different mobile devices and operating system versions to identify any compatibility issues.

2. Gmail (Receiving mail.)

➢ Positive test scenario:

- Ensure users can log in successfully to the Gmail app.
- Confirm that users can access the Inbox without any issues.
- Send an email from another account to the Gmail account being tested.
- Verify that the email is delivered to the Inbox.
- Send emails at different times and ensure they are delivered promptly.
- Verify that there is minimal delay between sending and receiving emails.
- Test the receipt of emails with various content types (text, HTML, images, attachments).

- Confirm that the content is displayed correctly within the email.
- Send emails with multiple recipients and verify that each recipient receives the email.
- Confirm that the email thread is correctly displayed for multiple recipients.
- Send multiple unread emails and verify that the unread email count is updated accurately.
- Confirm that the unread email count is displayed prominently.
- Ensure that users receive timely notifications for new emails.
- Test the display of email previews in notifications.
- Verify that emails are categorized correctly (Primary, Social, Promotions, etc.).
- Confirm that labels are applied as expected.
- Test searching for specific emails using keywords.
- Verify that search results are accurate and relevant
- Test the auto-refresh functionality to ensure new emails are automatically fetched.
- Confirm that users can manually refresh the Inbox to fetch new emails.
- Test marking emails as read or unread and verify that the status is updated instantly.
- Confirm that read/unread status syncs across multiple devices.
- Send emails with attachments of different types and sizes.
- Verify that users can open and download attachments without issues.
- Test starring emails and confirm that starred emails are highlighted appropriately.
- Verify that starred emails are easily accessible in the "Starred" or "Important" section.
- Test accessing previously received emails in offline mode.
- Positive: Confirm that users can compose emails in offline mode, and they are sent once the device is back online.
- Test receiving emails on multiple devices (e.g., mobile, desktop) simultaneously.
- Confirm that the email status and content sync across all devices.
- Test the effectiveness of Gmail's spam filter by sending test emails marked as spam.
- Verify that emails filtered by custom filters are organized into the correct folders.

➢ Negative test scenario:

- Attempt to log in with incorrect credentials and verify that appropriate error messages are displayed.
- Test logging in with a deactivated or suspended account.

- Introduce delays in the email delivery process and verify how the application handles delayed emails.
- Test receiving emails with large attachments and verify if the delivery is significantly delayed.
- Send emails with corrupted content or unsupported file types and verify how the application handles such emails.
- Test receiving emails with empty or missing content.
- Send emails with attachments exceeding the allowed size and verify if the application handles them appropriately.
- Test receiving emails with attachments that cannot be opened or downloaded.
- Disable notifications for the Gmail app and verify that users do not receive any notifications for new emails.
- Send emails with misleading content to test if the categorization algorithm misclassifies them.
- Test receiving emails that should be categorized but are not.
- Test searching for a specific email using incorrect or invalid keywords and verify the application's response.
- Attempt to search for emails while offline and check for inappropriate messages.
- Test marking emails as read or unread and verify if the status is not updated or synchronized correctly.
- Confirm that read/unread status does not sync across devices as expected.
- Test downloading attachments in poor network conditions and verify if the download fails or is incomplete.
- Attempt to download password-protected attachments and verify the application's response.
- Test accessing the Inbox in offline mode and verify if the application displays outdated or incorrect content.
- Attempt to compose and send emails in offline mode and verify the synchronization behavior.
- Test receiving emails on multiple devices and verify if there are discrepancies in the email status or content.
- Confirm that marking an email as read/unread on one device reflects incorrectly on another.
- Test the spam filter by sending emails that should not be marked as spam and verify if they are incorrectly filtered.
- Verify that custom filters do not misclassify or misplace emails.

3. Online shopping to buy product (Flipkart)

- Positive test scenario:
  - Ensure users can log in successfully to their Flipkart account.
  - Confirm that users can log in using various authentication methods (email, phone number, OTP).
  - Verify that users can search for a product using the search bar.
  - Test search filters and sorting options to refine product results.
  - Ensure users can view detailed product information (price, specifications, reviews).
  - Confirm that users can add a product to their shopping cart.
  - Verify that the selected product is added to the shopping cart.
  - Test the ability to update the quantity of products in the cart.
  - Confirm that users can proceed to checkout from the shopping cart.
  - Test different payment options (credit card, debit card, net banking, UPI) and ensure successful transactions.
  - Verify that users can enter and save shipping information.
  - Test scenarios where users choose different shipping options (standard, express) and confirm delivery dates.
  - Confirm that users can review the order summary before finalizing the purchase.
  - Verify that applicable discounts or promotional offers are applied correctly.
  - Test the final step of placing the order and ensure users receive an order confirmation.
  - Verify that users can track their order status after placement.
  - Test if the order details are correctly saved in the user's order history.
  - Confirm that users can view and print order invoices.
  - Verify that users receive order confirmation emails.
  - Test the email notification for order shipment and delivery.
  - Test the ability to leave product reviews and ratings.
  - Verify that reviews are correctly displayed on the product page.

- Negative test scenario:
  - Attempt to log in with incorrect credentials and verify appropriate error messages.
  - Test logging in with a deactivated or non-existent account.

- Search for a product that is out of stock and verify that it is appropriately labeled.
- Attempt to add an out-of-stock product to the cart and confirm the application's response.
- Test adding a product to the cart with an unusually large quantity and verify how the application handles it.
- Attempt to add a product to the cart with a network interruption and verify error messages.
- Test making a payment with invalid credit card details and verify the application's response.
- Simulate payment failures (e.g., insufficient funds) and confirm the error messages.
- Attempt to proceed to checkout without entering mandatory shipping information and verify error messages.
- Enter invalid shipping details and confirm that the application handles them appropriately.
- Test scenarios where users abandon the checkout process and verify if the application saves the items in the cart.
- Introduce delays in sending order confirmation or shipment notification emails and verify user expectations.
- Simulate high traffic on the website and test how the application handles performance under load.
- Introduce network latency and test the responsiveness of the application.

54. Write a Scenario of Wrist Watch

Ans. Test Scenario of Wrist Watch:

- Ensure the wristwatch is properly set to the correct time.
- Verify that the wristwatch is free from visible defects or damages.
- Watch hands move smoothly without jerking or skipping.
- The wristwatch maintains accurate timekeeping.
- Date and day functionality, if present, operate correctly.
- The crown operates smoothly without stiffness.
- Luminous elements glow adequately in low-light conditions.
- Water-resistant watches maintain their water resistance.
- The watch strap is durable and free from wear or damage.
- The clasp securely fastens and unfastens without issues.

- The watch continues to function after a gentle shock resistance test.
- Battery-powered watches last within the expected duration.
- Chronograph functionality, if present, operates accurately.
- Scratch-resistant coatings effectively protect the watch face.
- The wristwatch is comfortable and wearable for an extended period.

55. Write a Scenario of Lift (Elevator)

Ans. Test Scenario of Lift (Elevator):

- Ensure the lift is installed and connected to a power source.
- Verify that the lift is clean and free from any visible defects.
- The lift initializes without errors.
- Doors open and close smoothly without hesitation.
- Safety features, including sensors, operate as intended.
- The lift travels to selected floors without issues.
- The emergency stop button halts the lift's movement.
- The alarm system generates a noticeable alert.
- The lift operates within the specified weight limits.
- Doors reverse upon detecting an obstruction.
- Movement is stable, and the ride is comfortable.
- Doors open automatically upon reaching the selected floor.
- Indicator lights and displays accurately show the current status.
- The lift responds appropriately to a simulated power failure.
- The lift is accessible to users with disabilities.
- Components are easy to access for cleaning and maintenance.
- Users find the lift user-friendly and comfortable.

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

Ans. Test Scenario of WhatsApp Group (generate group):

- The WhatsApp application opens without errors.
- The "Chats" tab is accessible within the application.
- Contacts are successfully selected to initiate the group creation.
- The group name is entered and conforms to WhatsApp guidelines.
- A group profile picture, if set, is displayed correctly.
- Group privacy settings are adjusted as intended.
- The "Create" or "Next" button finalizes the group creation process.

- A confirmation message is displayed, indicating the successful creation of the group.
- The user is added as a member of the newly created group.
- Invited participants, if any, are successfully added to the group.
- The group info/settings accurately reflect the configured group name and settings.
- Test messages are delivered successfully within the group.
- Check if an admin can add others as Admin.
- Check if an admin can add others as Admin.

57. Write a Scenario of Instagram (video call with chat)

Ans. Test Scenario of Instagram (video call with chat):

- The Instagram application opens without errors.
- The login process is successful, and the user is authenticated.
- Direct messages are accessible within the application.
- A contact is selected to initiate a video call.
- The video call is initiated successfully, and the contact accepts the call.
- Chat messages can be sent and received during the ongoing video call.
- Toggling between the video call and chat windows is seamless.
- The video call is ended successfully.
- The call history accurately records the video call.
- Chat history with the contact is saved and accessible.
- Notifications related to the video call and chat are received.
- The process is repeatable with different contacts.

58. Write a Scenario of WhatsApp payment

Ans. Test Scenario of WhatsApp payment:

- Ensure users can link their bank accounts to WhatsApp.
- Test different types of supported banks.
- Test the process of setting up a transaction
- Confirm that users can send money successfully to their contacts.
- Test transactions with different amounts.
- Check that users receive confirmation messages after completing a transaction.
- Verify the accuracy of the transaction details in the confirmation message.

- Test the cancellation process for a transaction before and after completion.
- Ensure proper notifications are sent to both parties when a transaction is canceled.
- Attempt to perform unauthorized transactions and ensure they are blocked.
- Verify that users can view their transaction history.
- Confirm that users can check their WhatsApp Payments balance.
- Ensure error messages are clear and guide users on how to resolve issues.
- Test WhatsApp Payments integration with the overall WhatsApp application.
- Verify that users can seamlessly switch between regular messaging and payments.
- Test on different devices and screen sizes.