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

19.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/

Test Case

Test Case ID: TC001
Test Case Title: Verify Access to Articles on the Art of Testing Website

Objective:
To ensure that users can access articles on the Art of Testing website without encountering errors.

Preconditions:

- The user has a stable internet connection.
- The Art of Testing website is live and operational.

Test Steps:

1. Open a web browser.
2. Navigate to the Art of Testing website (https://artoftesting.com/).
3. Locate the "Articles" section from the main menu.
4. Click on any article title to open it.

Expected Result:

- The article should load successfully without errors.
- The content of the article should be displayed correctly, with all formatting intact.

Post-Conditions:

- The user is able to read the article.
- The website continues to function normally for further navigation

44.Write a scenario of a pen

Test Scenario: Testing the Functionality of a Pen

Objective:
To ensure that the pen writes smoothly and reliably while assessing its design, durability, and overall user experience.

Preconditions:

1. The pen is new and filled with ink.
2. The user has access to suitable writing surfaces (e.g., paper, notebook).

Test Steps:

1. Visual Inspection:
    - Action: User inspects the pen for any visible defects (e.g., cracks, misalignment of parts, ink leakage).
    - Expected Result: The pen should appear intact and free from defects.
2. Check Ink Flow:
    - Action: User removes the cap (if applicable) and checks the ink flow by writing a few lines on paper.
    - Expected Result: The pen should write smoothly without skipping, blotting, or running out of ink.
3. Assess Comfort and Grip:
    - Action: User holds the pen to evaluate its grip and comfort during writing.
    - Expected Result: The pen should feel comfortable in hand, with an ergonomic design that supports extended writing.
4. Test Different Writing Angles:
    - Action: User writes at various angles and pressures to assess ink performance.
    - Expected Result: The pen should maintain consistent ink flow regardless of writing angle or pressure applied.
5. Check for Drying Time:
    - Action: User writes a sentence and observes the drying time of the ink on paper.
    - Expected Result: The ink should dry quickly enough to prevent smudging when the user moves their hand over the writing.
6. Assess Durability:
    - Action: User applies moderate pressure while writing to test the pen's durability.
    - Expected Result: The pen should withstand pressure without breaking or leaking ink.
7. Test Refill Mechanism (if applicable):
    - Action: User attempts to replace the ink cartridge or refill the pen.
    - Expected Result: The refill process should be straightforward, with clear instructions (if applicable).
8. Check Cap and Clip Functionality (if applicable):

- Action: User tests the pen cap for secure closure and checks the clip for attachment to notebooks or pockets.
- Expected Result: The cap should fit securely, and the clip should hold the pen firmly in place when attached.

9. Evaluate Aesthetic Design:
    - Action: User assesses the design elements (e.g., color, finish, branding) of the pen.
    - Expected Result: The pen should be visually appealing and align with user preferences.
10. User Feedback:
    - Action: User provides feedback on their overall satisfaction with the pen after testing its functionalities.
    - Expected Result: Feedback should indicate whether the pen meets usability expectations and any suggestions for improvement.

Post-Conditions:

- The pen is left in a usable state, ready for future writing tasks.
- Any issues encountered during testing are documented for review.

Expected Outcome:

- The pen should function effectively, providing a smooth writing experience while meeting user expectations for comfort, durability, and design.

.

45.Write a test scenario of only Whatsapp chat messages ˅

Here’s a test scenario to validate the functionality of WhatsApp chat messages:

Test Scenario: Sending and Receiving Chat Messages on WhatsApp

Objective:
To ensure that users can successfully send and receive chat messages, and that the messaging features function correctly.

Preconditions:

1. The WhatsApp application is installed and configured on the user’s device.
2. The user has an active internet connection.
3. The user has contacts available in their WhatsApp account for messaging.

Test Steps:

1. Open WhatsApp.
2. Select a Contact:
3. Send a Text Message:
    - Action: User types a text message in the message input field and presses the "Send" button.

- Expected Result: The message should be sent successfully, displaying a check mark indicating delivery.

4. Receive a Text Message:
    - Action: Another user sends a text message to the user’s WhatsApp account.
    - Expected Result: The user should receive a notification of the new message, which appears in the chat window.
5. Check Message Read Status:
    - Action: User opens the received message.
    - Expected Result: The message should change to "read" status, indicated by two blue check marks (if read receipts are enabled).

Post-Conditions:

- The chat messages sent and received are visible in the chat history.
- Any issues encountered during testing are documented for review.

Expected Outcome:

- The messaging features in WhatsApp should function as intended, allowing users to send, receive, edit, and delete messages seamlessly.

46.Write a Scenario of Pen ˅

Test Scenario: Testing the Functionality of a Pen

Objective:
To ensure that the pen writes smoothly and reliably while assessing its design, durability, and overall user experience.

Preconditions:

1. The pen is new and filled with ink.
2. The user has access to suitable writing surfaces (e.g., paper, notebook).

Test Steps:

1. Assess Comfort and Grip:
    - Action: User holds the pen to evaluate its grip and comfort during writing.
    - Expected Result: The pen should feel comfortable in hand, with an ergonomic design that supports extended writing.
2. Test Different Writing Angles:
    - Action: User writes at various angles and pressures to assess ink performance.
    - Expected Result: The pen should maintain consistent ink flow regardless of writing angle or pressure applied.
3. Assess Durability:
    - Action: User applies moderate pressure while writing to test the pen's durability.
    - Expected Result: The pen should withstand pressure without breaking or leaking ink.

Post-Conditions:

- The pen is left in a usable state, ready for future writing tasks.
- Any issues encountered during testing are documented for review.

Expected Outcome:

- The pen should function effectively, providing a smooth writing experience while meeting user expectations for comfort, durability, and design.

47.Write a Scenario of Pen Stand ˅

Test Scenario: Testing the Functionality of a Pen Stand

Preconditions:

1. The pen stand is placed on a flat, stable surface.
2. The pen stand is empty and ready for use.

Test Steps:

1. Check Stability:
    - Action: User gently shakes the pen stand to assess its stability on the surface.
    - Expected Result: The pen stand should remain stable and not tip over.
2. Test Pen Capacity:
    - Action: User attempts to insert multiple pens (e.g., 3-5) into the pen stand to check its capacity.
    - Expected Result: The pen stand should accommodate the number of pens without difficulty, holding them securely.
3. Check for Easy Access:
    - Action: User removes and replaces pens in the stand to evaluate accessibility.
    - Expected Result: The pens should be easy to access and replace, allowing smooth retrieval.
4. Assess Design:
    - Action: User evaluates the design of the pen stand for aesthetics and functionality (e.g., color, shape, material).
    - Expected Result: The pen stand should be visually appealing and fit well in its intended environment (e.g., desk, office).

Post-Conditions:

- The pen stand is left in a usable state with all pens securely placed in it.
- Any issues encountered during testing are documented for review.

Expected Outcome:

- The pen stand should function effectively, holding pens securely while providing a user-friendly experience and maintaining aesthetic appeal.

48.Write a Scenario of Door

Test Scenario: Testing the Functionality of a Door

Preconditions:

1. The door is installed and properly aligned in its frame.
2. The door is equipped with a handle and locking mechanism (if applicable).
3. The environment around the door is clear for safe operation.

Test Steps:

1. Open the Door:
    - Action: User grips the handle and attempts to open the door by pulling or pushing (depending on door type).
    - Expected Result: The door should open smoothly without excessive force or obstruction.
2. Close the Door:
    - Action: User closes the door to ensure it latches securely.
    - Expected Result: The door should close easily and latch without sticking or requiring excessive force.

Post-Conditions:

- The door is left in a secure position, either locked or unlocked as per the user's preference.
- Any issues encountered during testing are documented for maintenance or repair.

Expected Outcome:

- The door should function as intended, providing secure access and ease of operation while exhibiting durability and stability.

49.Write a Scenario of ATM

Test Scenario: Using an ATM for Cash Withdrawal

Objective:
To ensure that the ATM operates correctly, allowing users to perform a cash withdrawal transaction smoothly and securely.

Preconditions:

1. The ATM is powered on and operational.
2. The user has a valid bank card (debit or credit) and knows the associated PIN.
3. The ATM has sufficient cash available for withdrawal.

Test Steps:

1. Visual Inspection:
2. Insert Card:

3. Enter PIN:
4. Select Transaction Type:
5. Choose Withdrawal Amount:
6. Confirm Transaction:
    - Action: User confirms the transaction to proceed with the withdrawal.
    - Expected Result: The ATM processes the request and displays a message indicating that the transaction is being completed.
7. Dispense Cash:
    - Action: User waits for the ATM to dispense the cash.
    - Expected Result: The ATM dispenses the correct amount of cash without errors or jams.
8. Provide Receipt :
    - Action: User selects whether to receive a printed receipt.
    - Expected Result: The ATM prints the receipt and provides it to the user if requested.
9. End Transaction:
    - Action: User takes the cash, receipt and removes the bank card from the ATM.
    - Expected Result: The ATM should eject the card, and the transaction should end smoothly.
10. Check Account Balance (optional):
    - Action: User checks their account balance after the withdrawal.
    - Expected Result: The ATM should display the updated account balance reflecting the withdrawal.

Post-Conditions:

- The ATM is ready for the next user.
- The user’s account is updated to reflect the transaction.

Expected Outcome:

- The ATM should function correctly throughout the process, allowing the user to withdraw cash without issues.
- The transaction should be secure, and the user experience should be efficient and user-friendly.

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

Test Scenario: Operating a Microwave Oven

Preconditions:

1. The microwave oven is plugged in and powered on.
2. The user has access to the microwave's control panel.
3. The food item to be heated is suitable for microwave use and placed in a microwave-safe container.

Test Steps:

1. Visual Inspection:
2. Check Power Supply
3. Set Cooking Time:
4. Start Cooking:
    - Action: User presses the "Start" button to begin cooking.
    - Expected Result: The microwave should start, and the timer should begin counting down.
5. Monitor Cooking Process:
    - Action: User observes the cooking process through the microwave's door to ensure that the food is heating evenly (if applicable).
    - Expected Result: The microwave operates quietly, and the food should rotate (if the microwave has a turntable) without any unusual noises or interruptions.
6. Open the Door:
    - Action: User carefully opens the microwave door.
    - Expected Result: The door should open smoothly, and there should be no steam or pressure release issues.
7. Check Food Temperature:
    - Action: User checks the temperature of the heated food using a food thermometer (if applicable) or by touching it carefully.

- Expected Result: The food should be heated to the appropriate temperature for safe consumption.

Post-Conditions:

- The microwave oven is left clean and ready for the next use.
- Any issues encountered during the process are documented for maintenance.

Expected Outcome:

- The microwave should function as expected, heating the food efficiently and safely.
- The entire operation should be smooth, with the user able to set cooking times, monitor the process, and receive the heated food without complications.

52.Write a scenario of Coffee vending Machine

Test Scenario: Operating a Coffee Vending Machine

Objective:
To ensure that the coffee vending machine can successfully prepare and dispense different types of coffee based on user selections while functioning correctly throughout the process.

Preconditions:

1. The coffee vending machine is powered on and operational.
2. The machine is filled with coffee supplies and has adequate water.
3. The machine has a user interface that allows for selection.

Test Steps:

1. Visual Inspection:
    - Action: User inspects the coffee vending machine for cleanliness and any visible defects
    - Expected Result: The machine should be clean, and all parts should appear intact and operational.
2. Check Power Supply:
    - Action: User checks to ensure that the machine is powered on and the display is functioning.
    - Expected Result: The display should show the welcome message and available coffee options.
3. Select Coffee Type:
    - Action: User selects a coffee typE from the menu displayed on the screen.
    - Expected Result: The selected coffee type should be highlighted or confirmed on the display.
4. Check Payment Method:
    - Action: User selects a payment method and follows the prompts to complete the payment.

- Expected Result: The payment process should complete without errors, and the user should receive a confirmation on the display.

5. Dispense Coffee:
    - Action: User presses the Start button to begin the coffee-making process.
    - Expected Result: The machine should start brewing the coffee, and the display should show the progress of the brewing process
6. Monitor Brewing Process:
    - Action: User observes the brewing process to ensure it proceeds without errors
    - Expected Result: The machine operates smoothly, and there are no error alerts or interruptions.
7. Receive Coffee:
    - Action: Once brewing is complete, the user waits for the coffee to be dispensed into the cup.
    - Expected Result: The machine dispenses the coffee into the designated cup or container without spills or blockages.
8. Check Quality of Coffee:
    - Action: User tastes the coffee to assess its quality (e.g., strength, temperature, flavor).
    - Expected Result: The coffee should match the selected options and taste fresh and at the appropriate temperature.

Post-Conditions:

- The coffee vending machine is ready for the next user.
- Any issues encountered during the process are documented for maintenance.

Expected Outcome:

- The coffee vending machine should function as intended, successfully preparing and dispensing the selected coffee.
- The entire process should be smooth, from selection to receipt of the coffee, with all functionalities operating correctly.

53.Write a scenario of chair

Test Scenario: Evaluating Comfort of a Chair

Objective:
To ensure that the chair is comfortable, and meets design specifications for use in various settings.

Preconditions:

1. The chair is fully assembled and in a location where it can be tested
2. The user is available to test the chair for comfort and functionality.

Test Steps:

1. Sit on the Chair:
    - Action: User sits on the chair to evaluate comfort.
    - Expected Result: The chair should support the user comfortably without any immediate discomfort.
2. Check Seat Height:
    - Action: If the chair has adjustable height, the user adjusts the seat to their preferred height and sits again.
    - Expected Result: The height adjustment mechanism should work smoothly, and the user should feel comfortable at the adjusted height.
3. Evaluate Back Support:
    - Action: User leans back against the backrest to assess lumbar support and overall comfort.
    - Expected Result: The backrest should provide adequate support, promoting a healthy posture without causing discomfort.
4. Test Armrests:
    - Action: User rests their arms on the armrests (if the chair has them) and adjusts them if they are adjustable.
    - Expected Result: The armrests should be at a comfortable height and width, allowing the user to rest their arms without strain..

Post-Conditions:

- The chair is assessed for comfort, and design appeal.
- Any defects or issues encountered during testing are noted for review.

Expected Outcome:

- The chair should meet all functional and comfort requirements, providing a pleasant user experience.
- The chair should be stable, adjustable, and durable, with a design that is visually appealing in its intended setting.

54.To Create Scenario (Positive & Negative)

Online shopping to buy product (flipkart)

Test Scenario: Purchasing a Product on Flipkart

Objective:
To ensure that a user can successfully search for, select, and purchase a product on Flipkart, with a smooth experience from product selection to order confirmation.

Preconditions:

1. The user has a registered account on Flipkart.
2. The user is logged into their account.

3. The user has a valid payment method linked to their account or is ready to enter one during checkout.

Test Steps:

1. Search for the Product:
    - Action: User enters a product name IPHONE 13 into the search bar and presses enter.
    - Expected Result: The search results display a list of products matching the search term, including the “iPhone 13” and related items.
2. Select the Product:
    - Action: User clicks on the desired product
    - Expected Result: The product details page loads, showing the product’s specifications, price, availability, and customer reviews.
3. Add the Product to Cart:
    - Action: User clicks on the “Add to Cart” button.
    - Expected Result: The product is added to the user’s cart, and a confirmation message appears, showing the cart summary.
4. Proceed to Checkout:
    - Action: User clicks on the cart icon and selects “Proceed to Checkout.”
    - Expected Result: The checkout page loads, showing the selected product, price breakdown, and shipping information.
5. Enter Shipping Details:
    - Action: User confirms or enters the shipping address.
    - Expected Result: The correct shipping address is displayed on the checkout page, and the user can see estimated delivery dates.
6. Select Payment Method:
    - Action: User chooses a payment method e.g., credit/debit card, net banking, or UPI and enters the required details.
    - Expected Result: The payment details are entered, and the user is prompted to confirm the payment.
7. Confirm Order:
    - Action: User reviews the order summary and clicks on the “Place Order” or “Confirm” button.
    - Expected Result: The payment is processed, and the user receives a confirmation message indicating that the order has been placed successfully.
8. Verify Order Confirmation:
    - Action: User checks their email or Flipkart account for an order confirmation email or notification.
    - Expected Result: An order confirmation is received, detailing the order number, product details, delivery estimate, and total cost.
9. Check Order Status:
    - Action: User navigates to the “My Orders” section on Flipkart to verify the order status.

- Expected Result: The new order is listed with an initial status

Post-Conditions:

- The order is successfully recorded in the user's account.
- The user receives a confirmation email or notification with order details.
- The user can view the order status in the "My Orders" section.

Expected Outcome:

- The user is able to search for, select, and purchase the desired product without encountering errors.
- All pages and features function as expected, and the user receives timely notifications regarding their order.

55.Write a Scenario of Wrist Watch

Test Scenario: Checking and Adjusting Time and Date on a Wristwatch

Objective:
To ensure that the wristwatch accurately displays the current time and date, and that the user can adjust both settings correctly.

Preconditions:

1. The wristwatch is operational and has sufficient battery power.
2. The watch has both time and date display features.
3. The user has access to the controls for adjusting time and date.

Test Steps:

1. Check Current Time Display:
    - Action: User looks at the wristwatch to check the current time.
    - Expected Result: The watch displays the correct current time, accurate to the minute.
2. Check Current Date Display:
    - Action: User verifies the date display on the watch.
    - Expected Result: The watch displays the correct date-day, month, and year.
3. Verify Accuracy:
    - Action: After a few minutes, the user checks the time and date again to confirm accuracy.
    - Expected Result: The watch maintains the correct time and date without further drift or errors.

Post-Conditions:

- The wristwatch displays the correct time and date.
- The watch returns to its normal operating mode and does not continue to adjust or drift from the set time.

56.Write a Scenario of Lift

Test Scenario: Using the Elevator to Travel Between Floors

Objective:
To ensure that the elevator can successfully transport a user from one floor to another upon request and display the correct status throughout the process.

Preconditions:

1. The elevator is powered on and operational.
2. The user is on a specific floor where the elevator is accessible.
3. There are no emergency situations or maintenance activities affecting the elevator.

Test Steps:

1. Call the Elevator:
    - Action: User presses the call button on their current floor 1 to summon the elevator.
    - Expected Result: The elevator arrives at the floor, and the door opens.
2. Select Destination Floor:
    - Action: User enters the elevator and presses the button for their destination floor 5.
    - Expected Result: The elevator button for Floor 5 lights up, and the doors close.
3. Elevator Movement:
    - Action: The elevator begins moving towards the selected floor, displaying the current floor number as it passes each floor.
    - Expected Result: The elevator moves smoothly without sudden stops or unusual noises. The floor indicator updates accurately as the elevator moves.
4. Arrive at Destination Floor:
    - Action: The elevator reaches Floor 5, and the doors open automatically.
    - Expected Result: The elevator stops at Floor 5, and the floor indicator shows “5”. The doors open fully, allowing the user to exit.
5. Exit the Elevator:
    - Action: User exits the elevator.
    - Expected Result: Once the user has exited, the elevator waits for a few seconds before closing the doors and becoming available for the next call.
6. Verify Return to Idle State:
    - Action: User observes the elevator after exiting to ensure it returns to its idle state, ready to respond to the next call.
    - Expected Result: The elevator doors close after a delay, and the elevator remains on Floor 5, idle and ready for the next user request.

Post-Conditions:

- The elevator is idle and available for the next user.
- No errors, alarms, or interruptions occur during the elevator’s operation.

Expected Outcome:

- The elevator correctly responds to the call and transports the user to the selected floor.
- The floor indicators and button lights display accurately throughout the operation.
- The elevator returns to an idle state once the user exits.

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

Test Scenario: Creating a New WhatsApp Group

Objective:
To ensure that a user can successfully create a new WhatsApp group, add members, and set a group name and profile picture.

Preconditions:

1. The user has an active WhatsApp account.
2. The user has at least two contacts available on WhatsApp to add to the group.
3. The user has a stable internet connection.

Test Steps:

1. Open WhatsApp and Navigate to Group Creation:
    - Action: User opens WhatsApp, navigates to the “Chats” tab, and taps on the “New Chat” icon. The user then selects “New Group” from the options.
    - Expected Result: The “Add Participants” screen appears, allowing the user to select contacts for the group.
2. Select Group Members:
    - Action: User selects two or more contacts from their WhatsApp contact list to add to the group and taps the “Next” button.
    - Expected Result: The “New Group” screen appears, prompting the user to enter the group name and set a profile picture.
3. Enter Group Name:
    - Action: User enters a name for the group.
    - Expected Result: The group name field updates to display the entered text.
4. Set Group Profile Picture :
    - Action: User taps on the camera icon to upload or take a photo for the group profile picture.
    - Expected Result: The selected photo is displayed as the group profile picture.
5. Create the Group:
    - Action: User taps the “Create” button to finalize the group creation.
    - Expected Result: The group is created, and the user is redirected to the new group chat interface.
6. Verify Group Creation:
    - Action: User checks the group chat to confirm the presence of the group name, profile picture, and added members.

- Expected Result: The group chat displays the group name at the top, along with the profile picture. The participants list includes the selected contacts.

Post-Conditions:

- The new group is visible in the user's chat list.
- All selected members are added to the group.
- Group members can participate in the chat.

Expected Outcome:

- The group is created successfully, with the selected members, group name, and profile picture.
- All group members are notified of their addition to the group and can access and interact in the group chat.

58.Write a Scenario of Whatsapp payment

Test Scenario: Validating WhatsApp Payment Process

Objective:
To ensure that the user can successfully send a payment to a contact through WhatsApp and both the sender and receiver receive appropriate notifications.

Preconditions:

1. The sender User A and receiver User B have active WhatsApp accounts.
2. Both User A and User B have set up WhatsApp Payments linked to their respective UPI accounts.
3. User A has sufficient funds in their account for the transaction.

Test Steps:

1. Open Chat:
    - Action: User A opens WhatsApp and navigates to the chat with User B.
    - Expected Result: User A sees the chat interface with User B.
2. Initiate Payment:
    - Action: User A taps on the "₹" icon at the bottom of the chat screen.
    - Expected Result: The payment interface opens, allowing User A to enter the payment amount.
3. Enter Payment Details:
    - Action: User A enters the amount (e.g., ₹500) and adds a note, "Payment for lunch."
    - Expected Result: The payment screen displays the entered amount and note, with an option to proceed.
4. Confirm Payment:
    - Action: User A taps on "Pay" and enters their UPI PIN to authenticate the transaction.
    - Expected Result: The payment processing screen appears, and User A receives a confirmation message upon successful payment.
5. Verify Payment Confirmation:

- Action: Check the chat interface for a confirmation message indicating the payment status.
- Expected Result: User A sees a message in the chat confirming that ₹500 has been sent to User B.

6. Receiver Notification:
    - Action: User B checks their WhatsApp for a notification regarding the payment.
    - Expected Result: User B receives a notification in their chat with User A that ₹500 has been received.
7. Verify Bank Transaction:
    - Action: Both User A and User B check their respective bank accounts for the transaction update.
    - Expected Result: User A sees ₹500 debited from their account, and User B sees ₹500 credited to theirs.

Post-Conditions:

- User A’s balance is updated to reflect the payment deduction.
- User B’s balance is updated to reflect the payment credit.

Expected Outcome:

- The payment transaction should be processed smoothly without errors.
- Both users should receive notifications about the payment status in their WhatsApp chats.
- Bank accounts should reflect the correct transaction amounts for both users.