1. **What is Exploratory Testing?**

Exploratory testing is a simultaneous process of learning, test design, and test execution where testers actively explore the software without predefined test cases

When to Use Exploratory Testing:

* When time is limited
* When requirements are unclear or incomplete
* During early stages of development
* To explore usability or real-world usage issues

1. **What is Traceability Matrix**?

A traceability matrix is a document that maps and tracks the relationship between two or more project components.Test conditions should be able to be linked back to their sources in the test basis. This is known as traceability.

Types of Traceability Matrix:

1. Forward Traceability – Mapping of requirements to test cases

2. Backward Traceability – Mapping of test cases to requirements

3. Bi-directional Traceability – A good traceability matrix references both: - From test cases → requirements - From requirements → test cases, documents, and use cases

1. **What is boundary value testing?**

Boundary value testing is a black box testing technique where test case focused on the boundaries of input ranges because that's where defects often occur. Boundary value testing is a method which refines equivalence partitioning.

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

Equivalence Partitioning (EP) is a black box testing technique where input data is divided into equal classes or groups (called equivalence classes) so that testing one value from each group is enough to represent the whole group.

This testing can be used for all levels of testing.

It is the process of determining the optimum number of test by reviewing documents such as functional design specification and detailed design specification and identifying each input condition within function(s).

1. **What is Intigration testing?**

Integration Testing is a level of software testing where individual modules or components are combined and tested as a group to verify that they work together correctly.It comes after unit testing and before system testing in the software testing life cycle (STLC).

There are two level of testing

1. component integration testing

2.system integration testing

* Purpose of integration testing
* To check data flow between modules.
* To identify interface issues like incorrect data formats ,communication errors ,or function mismatches.
* Inadequate exeption handling cold cause issues.

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

A factor that could result in future negative consequences usually expressed as impact and likelihood. A risk could be any future event with negative consequence

There are two types of risk

1. Project risk

2. Product risk

1. **What is Alpha Testing?**

Alpha testing is a type of acceptance testing performed by the internal team of the organization before releasing the software application to the clients.

- It is always performed in a virtual environment.

1. **What is Beta Testing?**

Beta testing is a type of acceptance testing where the software is released to a limited group of real users outside the company to use in a real-world environment before the final release.

1. **What is Component Testing?**

A minimal software item that can be tested in isolation. It means a unit is the smallest testable part of software. It is a type of software testing where individual components or modules of the software are tested independently to verify that each one works correctly in isolation.

- It is important for detecting bugs early at the module level.

- Makes debugging easier.

1. **What is Functional System Testing**?

Testing based on an analysis of the specification of the functionality of a component or system.

It is a type of software testing where the entire system is tested as a whole to ensure that it meets the specified functional requirements.

It is focused on what the system does rather than how it does it.

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

Testing the attributes of a component or a system that do not relate to functionality.

It is a type of software testing that checks how the system performs, rather than what it does.

It focus on the quality attributes of the system like performance,usability,reliability and security instead of the specific function.

1. **What is GUI Testing**?

It is the software testing that checks the user interface of an application to ensure that it behaves and appears as expected to the end user.

It verifies the elements like buttons, menus, icons, text boxes, checkboxes, error messages, and navigation.

1. **What is Adhoc Testing?**

Adhoc testing is an informal testing type with an aim to break the system by randomly exploring the application.

The main aim of this testing is to find defects by random checking.

Adhoc testing can be achieved with the testing technique called Error Guessing

1. **What is Load Testing?**

It is a performance testing to check system behavior under load. This testing helps determine how the application behaves when multiple users access it simultaneously.

1. **What is Stress Testing?**

Stress testing is also known as endurance testing.

Stress testing is to test the system behavior under extreme conditions and is carried out till the system failure.

Stress testing tries to break the system by testing with overwhelming data or resources.

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

White Box testing is a software testing technique where the internal structure, design, and code of the application are tested directly.

White box testing is known as testing based analysis of the internal structure of the component or system.

Testers have full knowledge of the code and write test cases to check logic, paths, loops, conditions, and data flow.

There are different types of white box testing

* Unit Testing - Testing individual functions or methods in isolation. Usually done by developers.
* Integration Testing - Testing the interaction between integrated modules.
* Code Coverage Testing - Measuring how much of the source code is tested.
* Path Testing - Testing all possible execution paths in the code.

1. **What black box testing? What are the different black box testing techniques?**

Testing , either functional or non-functional, without reference to the internal structure of the component or system. Specification-based testing technique is also known as ‘black-box’ testing

The tester have no longer knowledge about the system or component is structured inside the box.

In this testing the tester is concentrating on what the software does ,not how it does it.

Techniques of black box testing-

* Equivalence Partitioning-Divide inputs into valid and invalid groups.

Test only one value from each group — all other values are assumed to behave similarly.

* Boundary Value Analysis (BVA)- Focuses on values at the edge (boundaries) of valid input ranges.

Errors often occur at the boundaries.

* Decision Table Testing- Used when different combinations of inputs produce different results.

Represents logic using a table of conditions and actions.

* State Transition Testing-Based on state changes in the system in response to events or inputs.

Useful for systems with navigation (e.g. login/logout, form steps).

* Error Guessing- Based on experience and intuition of testers.

No formal method — guessing what kind of input might break the system.

1. **Mention what are the category of defects?**
2. Data Quality/Database Defects:

Related to incorrect data handling in the database.

Examples:

* Values not properly inserted/deleted.
* Wrong/null values saved.

1. Critical Functionality Defects:

Bugs that affect the most important functions of the application.

Examples:

* Exceptions that break main functionality.

1. Functionality Defects:

Affect how the system works (Functional behavior).

Examples:

* JavaScript errors.
* Buttons not working (e.g., Save, Cancel).
* Missing or Incorrect features.
* Infinite loops.

4)Security Defects:

Related to authentication and authorization problems.These are high priority defects.

Examples:

* Logging in with invalid credentials.
* Access to restricted pages.

1. User Interface (UI) Defects:

Issues with design, layout, and text.

Less severe but affect user.

Examples:

* Wrong error messages.
* Spelling Mistake.
* Alignment issues.

1. **What is big bang testing**?

Big Bang Testing is an integration testing approach where all components or modules are combined simultaneously and tested as a complete system, rather than integrating and testing them step-by-step.

Advantage of this testing is convenient for small system.

Disadvantage of this fault localization is difficult . it is time consuming and difficult to trace the cause of failures

1. **What is the purpose of exit criteria?**

Exit criteria defines the items that must be completed before testing can be concluded.

It is the set of predefined conditions or requirement that must be met before testing can be considered complete.

They help decide when to stop testing and ensure that the product is ready for release with acceptable quality and risk levels.

1. **When should regression testing be performed?**

As testing for dependencies between different components is a primary function of SIT testing, this area is often most subject to regression testing.

1. **What are 7 key principals? Explain in details?**

* Testing shows presence of Defects
* Exhaustive Testing is Impossible!
* Early Testing
* Defect Clustering
* The Pesticide Paradox
* Testing is Context Dependent
* Absence of Errors Fallacy

1. Testing shows presence of defects:

* Testing can show bugs are present, but never prove they are absent.
* Testing helps identify issues in the system but cannot guarantee 100% bug-free software.
* Even if no defects are found, it doesnot mean the software is perfect.

2. Exhaustive Testing is Impossible! :

* You cannot test everything.
* It is impractical to test all combinations of inputs, paths, and scenarios.
* Instead, testers use risk-based and prioritized approaches to focus on the most critical areas.

3.Early testing:

* Start testing early in the software development life cycle (SDLC)."
* Finding defects in the requirements or design phase is much cheaper than fixing them after deployment.
* Early involvement of testers helps improve product quality from the start.

4. Defect Clustering

* Most defects are found in a small number of module
* Often, 80% of bugs are found in 20% of the code
* Prioritizing those high-risk or complex areas improves testing efficiency.

5. Pesticide Paradox

* Same test cases won't find new bugs.
* Repeating the same tests over and over will eventually stop finding new bugs.
* To stay effective, test cases must be regularly reviewed and updated.

6. Testing is Context Dependent

* The approach to testing depends on the type of software.
* For a banking app, focus may be on security;

7. Absence of Errors Fallacy

* A software that works perfectly but doesn’t meet business needs is still a failure.
* A bug-free system is useless if it doesn't solve the user’s actual problem.

1. **Difference between QA V/s QC v/s Tester?**

|  |  |  |
| --- | --- | --- |
| **(QA)**  **Quality Assurance** | **(QC)**  **Quality Control** | **Tester** |
| Focus on improving and managing the process of software development and testing. | Focuses on checking the product to ensure it meets quality standard. | Focuses on executing tests and identifying bugs in the software. |
| It is Process oriented activities | It is Product oriented activities | It is Product oriented activities |
| It is Preventive activities | It is Corrective process | It is Preventive process |
| 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 |
| Creating a company-wide testing process and checklist | Reviewing the final app to check if features work properly. | Testing login, Payment, or search functions and reporting errors. |

1. **Difference between smoke and sanity**

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Smoke Testing** | **Sanity Testing** |
| Purpose | To check basic functionality of the build | To verify specific functionality after changes. |
| When | After a new build is received | After bug fixes or minor changes |
| Depth | Shallow testing-checks overall working | Deep testing – checks focused features deeply |
| Example | Does the app open? Can you log in? | Is the search function working after fixing the bug? |
| Time | Quick and broad | Quick but more focused |
| Who Performs | Testeror developer | Mainly testers |

1. **Difference between verification and validation**

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Verification** | **Validation** |
| Definition | Verification is the process of checking whether the software is being built correctly by following requirements, design & coding. | Validation is the process of checking whether the developer software meets user need & expectation. |
| Happens | Verification happens before testing | Validation happens after and during testing. |
| Process Type | It is a Static process type. | It is a Dynamic process type. |
| Who Performs | Verification does developers and analysts | Validation is does QA and Tester teams. |
| Activities | Activities like Review, Walkthrough and Inspection. | Activities like Actual Testing (Manual and Automation) |
| Evaluation Items | Include like Plan, Requirement Specification, Design Specification, Code, Test cases. | Include like Test the actual Product/Software. |

1. **Explain types of Performance testing?**

Performance Testing checks how a software application behaves under different conditions such as load, stress, and scalability. It ensures that the system is fast, stable, and responsive**.**

There are different types of performance testing

1. Load Testing

checks how the system handles expected user load.

Example: 1000 users logging in at once on an e-commerce site.

2 Stress Testing

Tests the system beyond its maximum limit.

Example: Sending 10,000 requests per second to a server.

3 Spike Testing

Sudden and extreme increase in user load.

Example: Flash sales, ticket bookings, or breaking news sites.

4 Endurance Testing (Soak Testing)

Tests the system over a long period of time.

Example: Running a system for 24+ hours continuously with steady load.

5 Scalability Testing

Tests the application’s ability to scale up or down.

Example: Adding more users or hardware to see if the app can handle growth.

6 Volume Testing

Tests with large volumes of data.

Example: Uploading 1 million records to a database.

1. **What is error, Defect, Bug and Failure?**

A mistake in coding is called Error, Error found by tester team it called Defect, and defect is accepted by developers’ team it’s called Bug, and product or system it’s not meets the requirement it’s called Failure.

1. **Difference between Priority and severity?**

|  |  |
| --- | --- |
| **Priority** | **Severity** |
| How soon the bug should be fixed | How serious the impact of the bug is. |
| Set by Tester/Manager. | Set by Tester. |
| Focus on business need. | Focus on Technical issues |
| Ex. Spelling Mistake on home page (high priority, low severity | Ex. App crashes when clicking a button (high priority, low severity (if rarely used)) |

1. **What is Bug Life Cycle?**

The Bug Life Cycle (also called Defect Life Cycle) is the process a bug goes through from the time it is found until it is fixed and closed.

It helps teams track, manage, and resolve bugs systematically during the software development and testing process.

* New: When a new defect is logged and posted for the first time. It is assigned a status as NEW.
* Assigned: Once the bug is posted by the tester, the lead of the tester approves the bug and assigns the bug to the developer team
* Open: The developer starts analyzing and works on the defect fix
* Fixed: When a developer makes a necessary code change and verifies the change, he or she can make bug status as “Fixed.”
* Pending retest: Once the defect is fixed the developer gives a particular code for retesting the code to the tester. Since the software testing remains pending from the testers end, the status assigned is “pending retest.”
* Retest: Tester does the retesting of the code at this stage to check whether the defect is fixed by the developer or not and changes the status to “Re-test.”
* Verified: The tester re-tests the bug after it got fixed by the developer. If there is no bug detected in the software, then the bug is fixed and the status assigned is “verified.”
* Reopen: If the bug persists even after the developer has fixed the bug, the tester changes the status to “reopened”. Once again the bug goes through the life cycle.
* Closed: If the bug is no longer exists then tester assigns the status “Closed.”
* Duplicate: If the defect is repeated twice or the defect corresponds to the same concept of the bug, the status is changed to “duplicate.”
* Rejected: If the developer feels the defect is not a genuine defect then it changes the defect to “rejected.”
* Deferred: If the present bug is not of a prime priority and if it is expected to get fixed in the next release, then status “Deferred” is assigned to such bugs
* Not a bug:If it does not affect the functionality of the application then the status assigned to a bug is “Not a bug”.

1. **Explain difference between functional testing and Non functional testing?**

|  |  |
| --- | --- |
| **Functional Testing** | **Non-Functional Testing** |
| Verifies the action and operation of the application, | Verifies performance, reliability, scalability of the application. |
| Based on Business logic and requirements. | Based on performance standards and quality attributes. |
| Manual or automation testing is done. | Mostly tool-based testing (e.g., performance testing tools). |
| Ex. Smoke testing, Regression Testing, Integration Testing. | Ex. Performance Testing, Load Testing, Stress Testing. |

1. **To create HLR & Test case of (instagram, Facebook) First page and chat functionality.**

Test Case of instagram facebook- [Click here](https://docs.google.com/spreadsheets/d/1ByzQDbzg0jqEt5qdwfeynip7GTc3KSMS/edit?usp=sharing&ouid=106452846719119311361&rtpof=true&sd=true)

1. **What is the difference between the STLC and SDLC?**

|  |  |  |
| --- | --- | --- |
| **Aspect** | **SDLC (Software Development Life Cycle)** | **STLC (Software Testing Life Cycle)** |
| Focus | Entire Software development process | Only the software testing process |
| Goal | Develop a complete, functional software product | Ensure the product is tested and meets quality standards |
| Phases include | Requirement Gathering  Design  Coding  Testing  Maintenance | Requirement Analysis  Test Planning  Test Case Design  Execution  Closure |
| Start when | Project Start | After requirements are defined |
| Involves | Developers, testers, business analysts, project managers | Mainly testers and QA team |
| Deliverable | Working software application | Tested product with test reports and defect logs |

1. **What is difference between test scenario, test cases and test script?**

|  |  |  |
| --- | --- | --- |
| **Test Scenario** | **Test Cases** | **Test Script** |
| High-level functionality to be tested | Detailed steps to verify a scenario | Code to automate the test case |
| Identify all testable areas | Validate specific condition and workflow | Automate execution for efficiency |
| Broad and general | Specific and detailed | Very detailed and technical |
| Written in simple language | Written in step-by-step format | Write in programming/scripting language. |
| Used by Testers and QA leads | Used by Testers | Used by Automation testers/developers |
| EX. Check User login | EX. Open login page-Enter valid details-click login-check redirection | Selenium code to perform login and validate successful login |

1. **Explain what test plan is ? what is the information that should be covered**

A Test Plan is a detailed document that describes the strategy, scope, objectives, resources, schedule, and activities for testing a software product.

It acts as a blueprint or roadmap for the testing process and helps ensure that testing is well-organized and covers all aspects of the application.

* Test Plan ID -Unique identifier
* Introduction -Overview of the project and test goals
* Scope - What will and will not be tested
* Test Strategy -Types of testing (manual, automation, functional, etc.)
* Test Environment - Hardware, software, tools needed
* Test Deliverables - Documents and reports to be delivered
* Resources & Roles - Who is responsible for what
* Schedule - Timeline of testing activities
* Entry & Exit Criteria-When to start and stop testing
* Risks and Contingencies - Known risks and backup plans

1. **What is priority?**

In software testing, Priority refers to how soon a defect or issue should be fixed based on its business or customer impact.It is assigned by testers or project managers based on how quickly the issue needs to be addressed, regardless of its severity

* Types of priority:
* Low:Defect can be fixed later it has minimal impact and doesn’t affect major functions. Ex. Type on about page.
* Medium**:** Defect should be fixed soon but is not critical it affects some functions but has workaround. Ex. Profile picture upload fails.
* High:Defect must be fixed quickly it affects important features or customer requirements. Ex. Payment crash.
* Critical:Defect needs an immediate fix it blocks key functions or system use. Ex. App won’t open.

1. **What is severity?**

It shows how serious the defects impact is on the system. Its set from a technical or tester point of view.

Severity can be of following types:

* Major:System or component stops data may be corrupted but workaround exists. Ex. System crash, data loss.
* Moderate:System gives wrong or incomplete results, no crash. Ex. Main feature broken, no workaround.
* Minor:Small issue, system works fine with a workaround. Ex. Small issues, has a workaround.
* Cosmetic:Only affects look/feel not functionality. Ex. Very small issues, like UI alignment or spelling mistake.

1. **Bug categories are…**
2. Data quality/ Data defect
3. Critically functionality defect
4. Functionality defect
5. Security defect
6. User interface defect
7. **Advantage of Bugzilla.**

* It is free and open-source.
* Easy to report and track bugs.
* Supports email notifications for updates.
* Allows setting bug priorities and severity.
* Helps manage projects and track progress.
* Offers good search and filter options.
* Supports multiple users and permissions.
* Generates useful bug and activity reports**.**

1. **Difference between Priority and severity.**

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Priority** | **Severity** |
| Meaning | Urgency to fix the bug | Impact of the bug on functionality |
| Focus | Business or customer urgency | Technical seriousness of the defect |
| Decided by | Project Manager, Client, Product Owner | Tester, Developer, QA lead |
| Ex1 | Wrong company name on homepage->High | Login button not working->Critical |
| Ex2 | Minor UI issue on main page->Medium | App crashes on specific action->Major |
| Ex3 | Small typo on contact page->Low | Incorrect calculation in billing->Major |
| Levels | High, Medium, Low, Critical | Major, Moderate, Minor, Cosmetic |

1. **What are the different Methodologies in Agile development Model?**

* Scrum
* Kanban
* Extreme Programming (XP)
* Lean
* Crystal
* Dynamic Systems Development Method (DSDM)
* Feature-Driven Development (FDD)

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

**Authentication**: Checks who the user is (example: username, password, OTP).

**Authorization**: Checks what the user is allowed to do (example: user can

view profile but not admin panel).

**Common problems faced in web testing:**

* Broken links
* Slow page loading
* Poor UI on different devices or browsers
* Security issues (like SQL injection, XSS)
* Session timeout or logout not working
* Form input validation errors
* Incorrect error or success messages
* Compatibility issues on different browsers or screen sizes

1. **To create HLR and Testcase of web Based (WhatsApp web)**

Whatsapp web- [Click here](https://docs.google.com/spreadsheets/d/1zDaGBn3OviiMyUhXBEiq_poRI41ZTEAS/edit?usp=sharing&ouid=106452846719119311361&rtpof=true&sd=true)

1. **Write Test cases of only Whatsapp chat messages**

[**“Click here”**](https://docs.google.com/spreadsheets/d/1h5txgj-UngfGkz8JQ8YAliNSGMbIoKvP/edit?usp=sharing&ouid=106452846719119311361&rtpof=true&sd=true)

1. **Write test case on whatsapp goup chat**

**Group chat-** [**click here**](https://docs.google.com/spreadsheets/d/1h5txgj-UngfGkz8JQ8YAliNSGMbIoKvP/edit?usp=sharing&ouid=106452846719119311361&rtpof=true&sd=true)

1. **Write a Scenario of Pen**

1 Check pen type it ink pen , gel pen ,or point pen.

1. Check the weight of pen.
2. Verify pen has cap or not.
3. Check the ink color of pen Red, Blue, or Black .
4. Check the pen for leakage under pressure No ink leaks when pressure is applied .
5. Verify if the pen works on different types of paper. Pen writes on notebook, printer, etc.
6. Check how long the pen writes .
7. Test the grip and comfort during long writing , Pen is comfortable for extended use .
8. Check writing is erasable or permanent.
9. Verify pen point is thick or thin.
10. Verify the pen label colour match with ink.
11. Check pen support multiple refills or not.

1. **Write a Scenario of Pen Stand**
2. Check material of pen stand it may be plastic , stainless steel,wooden.
3. Check stand is how long durable and maximum capacity of pen can store.
4. Verify the colour of stand is suitable.
5. Verify stand has sharp edges or corners, safe to use .
6. Check if it holds additional items like pencils or markers.
7. Check if the pen stand stays steady on the table and doesn’t fall over.
8. Check if the pen stand is easy to clean (remove dust or ink marks easily).
9. Check the weight of the stand is light so it can easy to move.
10. **Write a Scenario of Door**
11. Check material of door is wooden, aluminium, or stainless steel or iron.
12. Check the door is open single side or double side.
13. Check if the door open or close outside or inside.
14. Check the door is sliding door or rotating door open.
15. Check if the door size is matches with the measurement.
16. Check if the design of door is as per requirement.
17. Check what type of door lock like electric lock latch lock, or key lock.
18. Check the door has pin hole for see outside the door.
19. Check if the door has a stopper or not.
20. Check if the door can close automatically using a spring or similar system.
21. Verify the door is suitable in all weather in cold hot or rainy season.
22. Check if the door is not make noise when open and shut the door.
23. **Write a Scenario of ATM**
24. Verify the UI design of ATM machine like colour, design ,size material of atm machine
25. Check ATM machine has touch screen , button or both type.
26. Check if ATM machine button type than button is smoothly operated.
27. Insert a valid ATM/debit card , Card is accepted, and language selection screen appears.
28. Enter correct pin than transaction is proceed.
29. Enter pin is incorrect than transaction error and message show on display that invalid pin.
30. Enter pin limit is only three time after that message show on display card is blocked.
31. Verify withdraw cash with sufficient balance.
32. Verify the user can select account type like saving account or current account.
33. Check user can view account balance .
34. Check the user can not withdraw more than available balance in account
35. Verify user transaction limit of withdraw cash in one day.
36. Check ATM doesnot work with expired card error message shown on display.
37. Check user can change pin, deposit money in limit of account.
38. Check ATM restarts or crashes during transaction , Card is returned; transaction is rolled back .
39. **When to used Usablity Testing?**

Usability Testing is used when you want to evaluate how easy, intuitive, and user-friendly your application or system is for real users.

* During UI/UX Design Phase
* Before Launching a New Product or Feature
* When Revamping or Redesigning a Product
* For Accessibility Verification
* To Validate User Requirements
* After Completing Development

1. **What is the procedure for GUI testing?**

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.

1. **Write a scenario of Microwave Oven?**
2. Check if microwave size is match with the given specification.
3. Check if the material used in the microwave is safe for cooking.
4. Check if the internal space and material of steel used as specification.
5. Check if microwave temperature is correct for prepare food.
6. Check microwave maximum and minimum temperature capacity
7. Check microwave operate smoothly with solid and liquid food items.
8. Verify the power cord of oven is long enough.
9. Verify the door of Oven is open properly and close completely
10. Verify the glass of door is tolerance of maximum heat.
11. Verify the button given on oven is operate smoothly and instruction given on that read clearly.
12. Verify the speed of rotation plate is proper so food not spill over the oven.
13. **Write a scenario of coffee vending machine**
14. Verify the dimension of the machine as per specification.
15. Verify the material of inner body and outer body is as per specification.
16. Verify the machine’s color and company name and instruction is completely visible.
17. Check the quantity of milk water and powder of coffee is as per cup input proper or not.
18. Verify the voltage of machine require.
19. Verify the machine should not leak when power is off.
20. Check digital display give correct information.
21. Check the power button and other on and off button work properly.
22. Verify the coffee temperature is same every time
23. Check the quantity of coffee should poured in single operation
24. Verify the machine give a sign of indicator when machine is out of ingredients.
25. Verify the extra passage for drain extra coffee.
26. Verify he machine works properly in different weather like summer cold.
27. Verify the machine should not make too much noise when in operation.
28. **Write a scenario of chair**
29. Check if the chair is stable and can hold the weight of an average person.
30. Check the material of the chair like wood, plastic, etc
31. Check if the chair can be used as an office chair or home chair.
32. check if there is back support in chair.
33. Check support for hand in chair
34. Check the colour and size of chair.
35. Check the backside chair has cushion or not.
36. Check the durability of material of chair load capacity of chair.
37. Check the dimension and weight of the chair.
38. Check the height of the chair from floor.
39. Check how the chair reacts when washed with water.
40. **Create Test Cases on Compose Mail Functionality**

Compose mail- [Click here](https://docs.google.com/spreadsheets/d/1OhH73piOlsPjPo1pzjIPdGfEfqeuNeoQ/edit?usp=sharing&ouid=106452846719119311361&rtpof=true&sd=true)

1. . **Online shopping to buy product (flipkart)**

**Flipcart –** [**click here**](https://docs.google.com/spreadsheets/d/1OhH73piOlsPjPo1pzjIPdGfEfqeuNeoQ/edit?usp=sharing&ouid=106452846719119311361&rtpof=true&sd=true)

1. **Write a Scenario of Wrist Watch**

1. Verify the watch is digital or analog.
2. Verify the material of watch and its strap
3. Verify the watch is waterproof or not
4. Check the weight and durability of watch.
5. Check the number in dial are visible or not.
6. For analog, check if the second, minute, and hour hands show the correct time.
7. For digital, check if hours, minutes, seconds are shown correctly.
8. Check watch display with date and time or not
9. Verify the brand name is visiable in dial or not.
10. Check the alarm ,timer ,and stopwatch functionalty.
11. Verify watch comes with guaranty or warranty period time.
12. In the case of a digital watch, verify the format of the watch 12 hours or 24 hours.
13. Verify the dial glass material plastic or glass .and also check its durability.
14. **Write a Scenario of Lift(Elevator)**
15. Verify the elevator’ s dimension as per specification.
16. Verify the type of metal used in outside and inside the lift
17. Verify the door of the lift as per specification
18. Verify the capacity of lift in term of total weight
19. Check the button of open close the door and number of floors
20. Verify the lift moves to the floor which button is pressued
21. Verify that the lift stop when two button press at same time
22. Verify that at the emergency button to contact official
23. Verify lighting and fan in the lift
24. Verify that the lift interior is fully ventilation
25. Verify that in the case of multiple button press than every floor stop the lift
26. Verify that the lift is overloaded than warning alert is on display or voice
27. Verify that the lift doors don’t open while moving
28. Check if the lift works when the door open button is pressed before reaching the floor.
29. Verify that the lift stops safely during a power failure.
30. **Write a Scenario of Whatsapp payment**
31. Verify that the Whatsapp is latest version
32. Verify that the payment option is shown into attachment menu
33. Verify the user bank account is linked with the whatsapp
34. Verify the user select number from contact list
35. Verify the payment is done app ask for the UPI
36. Check if the payment completes successfully after entering the correct
37. Verify that the user and receiver both get payment confirmation messages
38. Check app shows payment history shown both sender and receiver
39. Verify that the if user input incorrect pin than payment failed
40. Verify that the internet is off than payment is not done
41. Verify that the app shows a proper message if the bank server is down or busy.