MODULE-2

Q1. What is exploratory testing?

Ans. It is a simultaneously learning, test design and test execution process.

- In this testing test planning, analysis, design and test execution are all done together and instantly.
- The tester will create or write down a test idea to give direction and explore the system while testing to further create critical, practical and useful for the successful testing of an application.
- Testing is often not recorded.
- Makes use of experience, heuristics and test patterns.
- Is carried out in time boxed intervals.
- Testing is based on a test charter that may include:
- Scope of the testing (in and out).
- The focus of exploratory testing is more on testing as a “thinking” activity.
- A brief description of how tests will be performed.
- Expected problems.
- More structured than Error guessing.
- Is carried out in time boxed intervals.
- Though the current trend in testing is to push for automation, exploratory testing is a new way of thinking. Automation has its limits.

- Is not random testing but it is Ad hoc testing with purpose of find bugs.

- Is structured and rigorous.

- Is cognitively (thinking) structured as compared to procedural structure of scripted testing. This structure comes from Charter, time boxing etc.

- Is highly teachable and manageable.

- Is not a technique but it is an approach. What actions you perform next is governed by what you are doing currently.

Q2. What is traceability matrix?

Ans. To protect against changes you should be able to trace back from every system component to the original requirement that caused its presence.

- A software process should help you keeping the virtual table up-to-date.
- Simple technique may be quite valuable (naming convention).
- Traceability Matrix (also known as Requirement Traceability Matrix - RTM) is a table which is used to trace the requirements during the Software development life Cycle.

It can be used for forward tracing (i.e. from Requirements to Design or Coding) or backward (i.e. from Coding to Requirements). There are many user defined templates for RTM.

- Each requirement in the RTM document is linked with its associated test case, so that testing can be done as per the mentioned requirements.
- Furthermore, Bug ID is also including and linked with its associated requirements and test case. The main goals for this matrix are:
    - Make sure Software is developed as per the mentioned requirements.
        - Helps in finding the root cause of any bug.
- Helps in tracing the developed documents during different phases of SDLC.
- A requirements traceability matrix is a document that traces and maps user requirements [requirement Ids from requirement specification document] with the test case ids. Purpose is to make sure that all the requirements are covered in test cases so that while testing no functionality can be missed.

Types of Traceability Matrix:

- Forward Traceability – Mapping of Requirements to Test cases.
- Backward Traceability – Mapping of Test Cases to Requirements.
- Bi-Directional Traceability - A Good Traceability matrix is the References from test cases to basis documentation and vice versa.

Pros of Traceability Matrix:

- Make obvious to the client that the software is being developed as per the requirements.
- To make sure that all requirements included in the test cases.
- To make sure that developers are not creating features that no one has requested.
- Easy to identify the missing functionalities.
- If there is a change request for a requirement, then we can easily find out which test cases need to update.
- The completed system may have “Extra” functionality that may have not been specified in the design specification, resulting in wastage of manpower, time and effort.

Cons of Traceability Matrix:

- No traceability or Incomplete Traceability Results into:
- Poor or unknown test coverage, more defects found in production.
- It will lead to miss some bugs in earlier test cycles which may arise in later test cycles. Then a lot of discussions arguments with other teams and managers before release.
- Difficult project planning and tracking, misunderstandings between different teams over project dependencies, delays, etc.

Q3. What is Boundary value testing?

Ans. The process of testing between extreme Ensor boundaries between partitions of the input values.

- Boundary value analysis is method which refines equivalence partitioning.
- So, these extreme ends like start-end, lower-upper, maximum-minimum, just inside-just outside values are called boundary values and the testing is called “Boundary testing”.

- The basic idea in boundary value testing is to select input variable values there:
    1. Minimum
    2. Maximum
    3. Just above the minimum
    4. A nominal value
    5. Just below the maximum
- In boundary value analysis you will check the boundary values like 0,1,10,11,99,100.
- In our earlier instead of checking one value for each partition you will check the values at the partitions like 0,1,10,11 and so on. As may you observe you test values at both valid and invalid boundaries. Boundary value analysis is also called range checking.
- BVA operates on the basis that experience shows us that errors are most likely to exist at the boundaries between partitions and in doing so incorporates a degree of negative testing into the test design.
- BVA Test cases are designed to exercise the software on and at either side of boundary values.
- Find the boundary and then test one value above and below it. Always results in two test cases per boundary for valid inputs and three tests cases per boundary for all inputs.

Q.4 What is Equivalence partitioning testing?

Ana. Divides input data of a software unit into partitions of equivalent data from which test cases can be derived.

- In principle, test cases are designed to cover each partition at least once.
- An advantage of this approach is reduction in the time required for testing software due to lesser number of test cases.
- If any one value from the test passes the test then the whole set of partition is considered pass or valid.
- The divided sets are called equivalence classes. Then we pick only one value from each partition for testing. The hypothesis behind this technique is that if one

condition value in a partition phases all others will also pass. If one conditions in a partition fails, all other conditions is that partition will fail.

- EP can be used for all Levels of Testing.
- The numbers fall into a partition where each would have the same, or equivalent, result i.e. an Equivalence Partition (EP) or Equivalence Class.
- EP says that by testing just one value we have tested the partition (typically a mid-point value is used). It assumes that:
    1. If one value finds a bug, the others probably will too
    2. If one doesn't find a bug, the others probably won't either
- In EP we must identify Valid Equivalence partitions and Invalid Equivalence partitions where applicable (typically in range tests).
- The Valid partition is bounded by the values 1 and 100.
- Time would be wasted by specifying test cases that covered a range of values within each of the three partitions, unless the code was designed in an unusual way.
- There are more effective techniques that can be used to find bugs in such circumstances (such as code inspection).
- EP can help reduce the number of tests from a list of all possible inputs to a minimum set that would still test each partition.
- If the tester chooses the right partitions, the testing will be accurate and efficient.
- EP is used to achieve good input and output coverage, knowing exhaustive testing is often impossible.
- It can be applied to human input, input via interfaces to a system, or interface parameters in integration testing.

Q.5 What is integration testing?

Ans. Testing performed to expose defect in the interfaces in the interactions between integrated components or systems.

- It is the level of the software testing process where individual units are combined and tested as group.
- The purpose is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in integration testing.
- It tests integration or interfaces between components, interaction to different part of the system such as an operating system, file system and hardware or interfaces between systems.
- Done by a specific integration tester or the team.
- There are 2 levels of integration testing:
    1. Component integration testing
    2. System integration testing

- **Component Integration Testing: -**

  It tests the integration between software components and is done after component testing.

  It is important to cover negative causes as well because components might make assumption with respect to the data.

  The following testing techniques are appropriate for integration testing:

  1. **Functional testing:**
  2. **Non-functional testing:**
- **System Integration Testing: -**

  It verifies the proper execution of software components and proper interfacing between components within the solution.

  The objective of SIT testing is to validate that all software module dependencies are functionally correct and that data integrity is maintained between separate modules for the entire solution.

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

- **Need of Integration Testing:**

  A module in general as designed by an individual software developer who understanding and programming logic may differ from other programmers. Integration testing became necessary to verify the software modules work in unity.

  at the time of module development, there wide chances of change in requirements by the clients. These new requirements may not be unit tested and hence integration testing become necessary.

  Interfaces of the software modules with the database could be erroneous external hardware interfaces. If any, could be erroneous inadequate expecting handling could cause issues.

- **Integration Testing methods: -**

  Any of black box testing, white box testing and gray box testing methods can be used. Normally the method depends on your definition of ‘unit’.

  These are two types of methods: -

  1. **Bing Bang Integration Testing**
  2. **Incremental Integration Testing**
     a. Top down approach
     b. Bottom up approach

- **Big Bang Integration Technique:**
    - In big bang integration testing all components or modules is integrated simultaneously after which everything is tested as a whole.
    - Big bang testing has the advantage that everything is finished before integration testing starts.
    - The major disadvantage is that in general it is time consuming and difficult to trace the cause of failures because of this late integration.
    - Here all components are integrated together at once, and then tested.
- **Advantages: -**
    - Convenient for small system.
- **Disadvantages: -**
    - Fault localization is difficult.
    - Since the integration testing can commence only after "all" the modules are designed, testing team well have less time for execution in the testing phase.
    - Since all modules are tested at once, high risk critical modules are not isolated and tested on priority.
- **Incremental testing:**
    - The incremental approach has the advantage that the defects are found early in a smaller assembly when it is relatively easy to defect the code.
    - A disadvantage is that it can be time consuming since stubs and drivers have to be developed and used in the test.
    - In this approach, testing is done by joining two or more modules that are logically related.
    - Then the other related modules are added and tested for the proper functioning. Process continue until all of the modules are joined and tested successfully.
    - This process is carried out by using dummy programs called stubs and drivers. Stubs and driver do not implement the entire programing logic of the software module but just simulate data communication with the calling module.
    - Stubs is called by the module under test.
    - Driver called the module to be tested**.**

a. Top down approach:

- Testing takes place from top to bottom. Following the control flow or architectural structure. Components or system are substituted by stubs.
- in top to down approach, testing takes place from top to bottom to down following the control flow of the software system.
- Takes helps of stubs for testing.

- **Advantages:**
    - fault localization is easier.
    - Possibility to obtain an early prototype.
    - critical modules are tested on priority major design flows could be found and fix first.

- **Disadvantages:**
    - Needs many stubs.
    - Modules are lower level are tested to inadequately.

b. Bottom up approach:

- Testing takes place from the bottom of the control flow upwards. Components or systems are they substituted be the drivers.
- In the bottom up strategy, each module at lower level is tested with higher modules until all modules are tested. It takes help of drivers for testing.

- **Advantages:**
    - Fault localization is easier.
    - No time is wasted waiting for all modules to be developed unlike big bang approach.
- **Disadvantages:**
    - Critical modules which control the flow of application are tested last and may be prone to defects.
    - Early prototype is not possible.

Q.6 What determines the level of risk?

Ans. Risks should be prioritised according to their level, which is obtained by assessing the likelihood of the event occurring and the impact of that event. Then, the residual level should be determined by considering the management response to the risk.

Q.7 What is alpha testing?

Ans. It is always performed by the developers at the software development sit.

- Sometimes it is also performed by the independent testing team.
- Alpha testing is not open to the market and public.
- It is conduct for the software application and project.
- It is always performed within the organization.
- It comes the under category which of both white box testing and black box testing.
- It is always performed in virtual environment.
- It is the form of Acceptance Testing.
- Alpha Testing is definitely performed and carried out at the developing organizations location with the involvement of developers.
- During this phase, the following will be tested in the application:
    - Spelling mistakes
    - Broken links
    - Cloudy direction

- Alpha Testing is always performed at the time of Acceptance Testing when developers test the product and project to check whether it meets the user requirements or not.
- It is always performed at the developer's premises in the absence of the users.
- It is considered as the User Acceptance Testing (UAT) which is done at developer's area.
- Unit testing, integration testing and system testing when combined are known as alpha testing.

Q.8 What is beta testing?

Ans. it is always performed by the customer at their own site.

- It is not performed by independent testing team.
- Beta testing is always open to the market public.
- It is usually conduct for software product.
- It is performed in real time environment.
- It is always performed outside the organization.
- It is only kind of black box testing.
- It is also the form of Acceptance 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.
- Beta Testing is always performed at the time when software product and project are marketed.
- It is always performed at the user's premises in the absence of the development team.
- It is also considered as the User Acceptance Testing (UAT) which is done at customers or users' area.
- Beta testing can be considered "pre-release" testing.
- Pilot Testing is testing to product on real world as well as collect data on the use of product in the classroom.

Q.9 What is component testing?

Ans. component(unit)- "A unit is the smallest testable part of software.

- Unit testing is level of the software testing process where individual testing units/components of software/system are tested. The purpose is to validate that each unit of the software performs as designed.
- Sometimes known as unit testing, module testing or program testing.
- Unit testing frameworks, stubs, drivers and mock of fake objects are used to assist I in unit testing.

- Unit tests are typically written and run by the software developers to ensure that code meets its design and behaves as intended with debugging tool.
- It is a method by which individual units of source code are tested to determine if they are fit for use.
- A unit is the smallest testable part of an application like functions/procedures, classes and interfaces.
- The goal is to isolate each part of the program show that the individual parts are correct.
- It provides a strict, written contract that the piece of code must satisfy. As a result it affords several benefits.
- Unit tests find problems early in the development cycle.
- Unit testing is performed by using the white box testing method.
- Unit testing in Extreme Programming involves the extensive use of testing frameworks. A unit test framework is used in order to create automated unit tests. Unit testing frameworks are not unique to extreme programming, but they are essential to it. Below we look at some of what extreme programming brings to the world of unit testing:
    - Tests are written before the code
    - Rely heavily on testing frameworks
    - All classes in the application are tested
    - Quick and easy integration is made possible
- **Limitations of tests:**
    - Unit testing can't be expected to catch every error in a program. It is not possible to evaluate all execution paths even in the most trivial programs.
    - Unit testing by its very nature focuses on a unit of code. Hence it can't catch integration errors or broad system level errors.
- **Building Unit Test cases:**
    - Unit testing is commonly automated, but may still be performed manually. The IEEE does not favour one over the other. A manual approach to unit testing may employ a step-by-step instructional document.

Q.10 What is functional system testing?

Ans. A requirement that specifies a function that a system or system component must perform.

- A requirement may exist as a text document and/or a model.
- There are two types of techniques:
    - Requirement based functional testing
    - Process based testing

1. Functional system testing functionally as below:
    1. **Accuracy: -** provision of right or agreed results or effects.

2. **Interoperability: -** ability to interact with specified systems.
3. **Compliance: -** adhere to applicable standards, conventions, regulations or laws.
4. **Auditability:** - ability to provide adequate and accurate audit data.
5. **Suitability:** - presence and appropriateness of functions for specified tasks.

- **Requirement based testing:**
  - Testing against requirements and specifications.
  - Detailed user requirements.
  - System requirements functional specifications.
  - User documentation/instructions.
  - High level system testing.
  - Starts by using the most appropriate black box testing techniques.
  - May support this with white-box techniques (e.g. menu structures, web page navigation).
  - Risk based approach.
- **Business process-based testing:**
  - Expected user profile.
  - Business scenarios.
  - Use cases.
  - Testing should reflect the business environment and processes in which the system will operate.
  - Therefor, test cases should be based on real business processes.

Q.11 What is non-functional testing?

Ans. Non-functional testing checks the performance, reliability, scalability and other non-functional aspects of the software system.

- Non-functional testing should be performed after functional testing.
- Using tools will be affective for this testing.
- Performance parameter like speed, scalability are inputs to non-functional testing.
- Non-functional testing describes how good the product works.
- Tough to do manual testing.
- The non-functional aspects of a system are all the attributes other than business functionality, and are as important as the functional aspects. This includes:
  - the look and feel and ease of use of the system.
  - how quickly the system performs.
  - how much the system can do for the user.
- It is also about how easy and quick the system is to install.
- It is also about how robust it Is.
- It is also about how quickly the system can recover from a crash.

- Types of non-functional testing:
    1. Performance testing
    2. Load testing
    3. Volume testing
    4. Stress testing
    5. Security testing
    6. Installation testing
    7. Penetration testing
    8. Compatibility testing
    9. Migration testing

Q.12 What is GUI testing?

Ans. Graphical User Interface testing is the process of testing the systems GUI of the system under test.

- GUI testing involves checking the screens with the controls like menus, buttons, icons and all types of tool bar, menu bar, dialog boxes and windows etc.
- **What do you check in 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 message are display correctly.
    - Check for clear demarcation of different sections on screen.
    - Check font used in applicable is readable.
    - Check the alignment of the text is proper.
    - Check the color of the font and waiting message is aesthetically pleasing.
    - Check that the Images have good clarity.
    - Check that the images are properly aligned.
    - Check the positioning of GUI elements for deferent screen resolution.

- **GUI testing examples:**
    a. **Web based testing and desktop-based testing:**
        - The scrollbar should be enabled only when necessary.
        - Font size, style and color for headline, description, text, labels, infield data and grid into should be standard as specified in SRS.
        - The description text box should be multi-lined.
        - Enough space should be provided between field labels, columns, rows, error message etc.

b. **Mobile based testing:**
    - If mobile is in every orientation mode so display image and video properly.
    - Every app will display in responsive type.
    - Alignment should be applying properly of every field.
c. **Game based testing:**
    - Game infra design will be showing properly.
    - Game points or score will display proper with its background color.
    - Game sound manage with its background effect.
    - Can be also conducted in advance of designing page logouts or navigation menus.

Q.13 What is ad hock testing?

Ans. Ad hock testing is an informal testing type with an aim to break the system.

- It does not follow any test design techniques to create test cases.
- This testing is primarily performed if the knowledge of testers in the system under test is very high.
- Main aim of this testing is to find defects by random checking.
- Ad hock testing can be achieved with the testing technique called error guessing.
- Error guessing can be done by the people having enough experience on the system to “guess” the most likely source of errors.
- In fact, it does not create test cases altogether.
- Testers randomly test the application without any test cases or any business requirement document.
- Ad hock testing does not follow any structured way of testing and it is randomly done on any part of application.
- The error guessing is a technique where the experienced and good testers are encouraged to think of situations in which the software may not be able to cope.
- Some people seem to be naturally good at testing and others are good testers because they have a lot of experience either as a tester or working with a particular system and so are able to find out its weaknesses.
- This is why an error guessing approach, used after more formal techniques have been applied to some extent, can be very effective.
- It also saves a lot of time because of the assumptions and guessing made by the experienced testers to find out the defects which otherwise won’t be able to find.
- Using experience to postulate errors.
- Use Error Guessing to Complement Test Design Techniques.

- **Types of Ad hock testing:**
    1. **Buddy testing:**
        - Two buddies mutually work on identifying defects in the same module. Mostly one buddy will be from development team and another person will be from testing team. Buddy testing helps the testers develop better test cases and development team can be make design changes early. This testing usually after unit testing completion.
    2. **Pair testing:**
        - Two testers are assigned model, share ideas and work on the same machines to find defects. One person can execute the tests and another person can make notes on the findings. Roles of the person can be a tester and scriber during testing.
    3. **Monkey testing:**
        - randomly test the product or application without test cases with a goal to break a system.

Q.14 What is load testing?

Ans. It is a performance testing to check system behavior under load testing an application under heavy loads. Such as testing of website under a range of loads to determine at what point the systems response time degrades of fails.

- Load testing is a kind of performance testing which determines a systems performance under real-life load conditions. This testing helps determine how the application behave when multiple user access it simultaneously.
- This testing usually identifies: -
    - The maximum operating capacity of an application.
    - Determine whether current infrastructure is sufficient to run the application.
    - Sustainability of application with respect to peak user load.
    - Number of concurrent users that an application can support and scalability to allow more users to access it.
    - It is a type of non-functional testing. Load testing is commonly used for the client/server, web-based application- both intranet and internet.
- **Need for Load Testing:**
    - Some extremely popular sites have suffered serious downtimes when they get massive traffic volumes. E-commerce websites invest heavily in advertising campaigns, but not is load testing to ensure optimal system performance, when that marketing brings in traffic.
- **For Example:**
    - Popular toy store toysrus.com, could not handle the increased traffic generated by their advertising campaign resulting in loss of both marketing dollars and potential toy sales.

    - An Airline website was not able to handle 10000+ users during a festival offer.
    - Encyclopaedia Britannica declared free access to their online database as a promotional offer. They were not able to keep up with the onslaught of traffic for weeks.
    - Facebook (FB).
- **Why Load Testing:**
    - Load testing gives confidence in the system and its reliability and performance.
    - Load testing helps identify the bottlenecks in the system under heavy user stress scenarios before they happen in a production environment.
    - Load testing gives excellent protection against poor performance and accommodates complementary strategies for performance management and monitoring of a production environment.
- **Goals of Load Testing:**
    - Loading testing identifies the following problems before moving the application to market or Production:
        - Response time for each transaction.
        - Performance of the system components under various loads.
        - Performance of database components under different loads.
        - Network delay between the client and the server.
        - Software design issues.
        - Server configuration issues like web server, application server, database server etc.
        - Hardware limitation issues like CPU maximization, memory limitations, network bottleneck etc.
    - Load testing will determine whether system needs to be fine-tuned or modification of hardware and software is required to improve performance.
- **Pre-requisites for Load Testing:**
    - The chief metric for load testing is response time. Before you begin load testing, you must determine –
        - Whether the response time is already measured and compared – quantitative.
        - Whether the response time is applicable to the business process – relevant.
        - Whether the response time is justifiable – realistic.
        - Whether the response time is achievable – achievable.
        - Whether the response time is measurable using a tool or stopwatch – measurable.

    - Hardware Platform Software Configuration:
        - Server machines
        - Processors
        - Memory
        - Disk storage
        - Load machines configuration
        - Network configuration
        - Operating system
        - Server software
- **Strategies of Load Testing:**
    - Manual Load Testing.
    - In house(organization) developed load testing tools.
    - Open source load testing tools.
    - Enterprise (record and play) load testing tools.
- **Load Testing Tools:**
    - Load runner
    - Web load
    - Astra load test
    - Reviews web load
    - Studio, rational site load
    - Silk performer
- **Pros and Cons of Load Testing:**
- **PROS:**
    - Performance bottlenecks identification before production.
    - Improves the scalability of the system.
    - Minimized risk related to system downtime.
    - Reduced costs of failure.
    - Increase customer satisfaction.
- **CONS:**
    - Need programming knowledge to use load testing tools.
    - Tools can be expensive as pricing depends on the number of virtual users supported.

Q.15 What is stress testing?

Ans. system is stress beyond its specifications to check how and when it fails. Performed under heavy load like putting large number beyond storage capacity, complex database queries, continuous input to system or database load.

- Stress testing is used to test the stability & reliability of the system. This test mainly determines the system on its robustness and error handling under extremely heavy load conditions.
- It even tests beyond the normal operating point and evaluates how the system works under those extreme conditions.

- Stress testing is done to make sure that the system would not crash under crunch situations.
- Stress testing is also known as endurance testing.
- Under Stress Testing, AUT is be stressed for a short period of time to know its withstanding capacity.
- Most prominent of stress testing is to determine the limit, at which the system or software or hardware breaks.
- It also checks whether system demonstrates effective error management under extreme conditions.
- The application under testing will be stressed when 5GB data is copied from the website and pasted in the notepad.
- Notepad is under stress and gives 'Not Responded' error message.
- Examples of stress conditions include:
    - Excessive volume in terms of either users or data; examples might include a denial of service (DoS) attack or a situation where a widely viewed news item prompts a large number of users to visit a Web site during a three-minute period.
    - Resource reduction such as a disk drive failure.
    - Application components fail to respond.
- **Need for stress testing:**
    - During festival time, an online shopping site may witness a spike in traffic, or when it announces a sale.
    - When a blog is mentioned in a leading newspaper, it experiences a sudden surge in traffic.
    - To check whether the system works under abnormal conditions.
    - Displaying appropriate error message when the system is under stress.
    - System failure under extreme conditions could result in enormous revenue loss.
    - It is better to be prepared for extreme conditions by executing Stress Testing.
- **Goal of Stress Testing:**
    - The goal of stress testing is to analyze the behavior of the system after failure. For stress testing to be successful, system should display appropriate error message while it is under extreme conditions.
    - To conduct stress testing, sometimes, massive data sets may be used which may get lost during stress testing.
    - The main purpose of stress testing is to make sure that the system recovers after failure which is called as recoverability.
- **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

- **Metric for stress testing:**
    - **Measuring Scalability & Performance**
        - **Pages per second-** Measures how many pages have been requested/second.
        - **Throughput-** Basic Metric – Response data size/Second.
        - **Rounds-** Number of times test scenarios has been planned Versus Number of times client has executed.
    - **Application Response:**
        - **Hit time-** Average time to retrieve an image or a page.
        - **Time to the first byte-** Time taken to return the first byte of data or information.
        - **Page time-** Time taken to retrieve all the information in a page.
    - **Failures:**
        - **Failed connection-** Number of failed connections refused by the client.
        - **Failed rounds-** Number of rounds it gets failed.
        - **Failed hits-** Number of failed attempts done by the system.

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

Ans. Testing based on an analysis of the internal structured of the components or system.

- Structured-based testing techniques is also known as '**white-box'** or '**glass-box'** testing techniques because here the testers require knowledge of how the software is implemented, how it works.
- In white-box testing the tester is concentrating on how the software does it.
    - For example, a structural technique may be concerned with exercising loops in the software.
- Different test cases may be derived to exercise the loop once, twice and many times. This may be done regardless of the functionality of the software.
- Structured-based techniques are also used in system and acceptance testing, but the structures are different.
    - For example, the coverage of menu options or major business transactions could be the structural element in system or acceptance testing.
- Testing based upon the structured of the code.
- Typically undertaken at Component and Component Integration Test phases by development teams.

- White box testing is the detailed investigation of internal logic and structured of the code.
- White box testing is also called glass testing or open box testing. In order to perform white box testing on an application, the tester needs to possess knowledge of the internal working of the code.
- These are the testing techniques are below:
    - Statement Coverage
    - Branch Coverage
    - Condition Coverage
    - Multiple Condition Coverage
    - Basis Path Testing
    - Loop Testing

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

Ans. Testing, either functional or non-functional, without reference to the internal structure of the component or system.

- Specification-based testing techniques is also known as '**Black-box**' or input/output driven testing techniques because they view the software as a black box with inputs and outputs.
- The testers have no knowledge of how the system or component is structured inside the box. In black box testing the tester is concentrating on what the software does, not how it does it.
- Specification-based techniques are appropriate all level of testing (component testing through to acceptance testing) where a specification exists.
- For example, when performing system or acceptance testing, the requirements specification or functional specification may form the basis of the tests.
- The technique of testing without having any knowledge of the interior workings of the application is Black Box testing.
- What a system does, rather than HOW it does it.
- Typically used at System Test phase, although can be useful throughout the test lifecycle.
- The tester is oblivious to the system architecture and does not have access to the source code.
- Typically, when performing a black box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.
- **Techniques of Black box Testing:**
    - There are four specification-based or black-box technique:
        - Equivalence partitioning
        - Boundary value analysis

- Decision tables
- State transition testing
- Use-case testing
- Other black box testing-syntax or pattern testing

Q.18 Mention what are the categories of defects?

Ans. The defect is the occurrence of variance between the accepted result and the actual result.

- Defect is some kind of error or mistakes which prevents the system to work smooth.
- There is different type of categories:
  - **Database defect:** deals with improper handling of data in the database.
  - **Examples:**
    - Values not deleted or inserted into the database properly.
    - Improper or null values inserted in place of actual values.
  - **Critical Functionality defects:** The occurrence of these bugs hampers the crucial functionality of the application.
  - Example**:** Exceptions.
  - **Functionality defects:** These defects affect the functionality of the application.
  - Example: all JavaScript errors, Buttons like save, delete, cancel not performing their intended functions.
  - **Security defects:** Application security defects generally involve improper handling of data sent from the user to the application. These defects are the most severe and given highest priority for a fix.
  - Examples: Authentication: Accepting and invalid username/password.
  - Authorization: Accessibility to pages though permission not given.
  - **User Interface Defects:** As the name suggests, the bugs deal with problems related to UI are usually considered less severe.
  - Examples: Spelling mistakes, Alignment problems, improper error/UI messages.

Q.19 Mention what big bang test is?

Ans. In big bang integration testing all components or modules is integrated simultaneously, after which everything is tested as a whole.

- Big bang testing has the advantages that everything is finished before integration testing starts.
- The major disadvantages are that in general it is time consuming and difficult to trace the cause of failures because of this late integration.
- Here all components are integrated together at once, and then tested.

- **Advantages:**
    - Convenient for small system.
- **Disadvantages:**
    - Fault localization is difficult.
    - Since the integration testing can commence only after “all” the modules are designed, testing team well have less time for execution in the testing phase.
    - Since all modules are tested at once, high risk critical modules are not isolated and tested on priority.

Q.20 What is the purpose of exit criteria?

Ans. how do we know when to stop testing?

- Run out of time?
- Run out of budget?
- The business tells you it went live last night!
- Boss says stop?
- All defects have been fixed?
- When out exit criteria have been meet?

- 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.
- Exit Criteria typically measures:
    - Thoroughness measures, such as coverage of requirements or of code or risk coverage.
    - Estimates of defect density or reliability measures. (e.g. how many defects open by category).
    - Cost.
    - Residual Risks, such as defects not fixed or lack of test coverage in certain areas.
    - Schedules - such as those based on time to market.

Q.21 When should "Regression Testing" be performed?

Ans. Regression testing should be done to ensure that existing functionality works as excepted after making changes to code, such as adding new features, fixing bugs of previously tested program, or improving performance.

- Regression testing is performed before each new instance of the product is deployed, guaranteeing that the program works perfectly in each setting. For instance, we need to make sure the product continues to perform properly in a production environment before we release it.
- It should be done to ensure that, after changes made in module it does not affecting the other.

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

Ans. General testing principles:

- Testing shows presence of defects
- Exhaustive Testing is impossible!
- Early Testing
- Defect Clustering
- The Pesticide Paradox
- Testing is Context Dependant
- Absence of Error Fallacy

- **Testing shows presence of Defects:**
  - Testing can show that defects are present, but cannot prove that there are no defects.
  - Testing reduce the probability of undiscovered defects remaining in the software but, even if no defects are found, it is not a proof of correctness.
  - We test to find faults.
  - As we find more defects, the probability of undiscovered defects remaining in a system reduces.
  - However, Testing cannot prove that there are no defects present.
- **Exhaustive Testing is Impossible:**
  - Testing everything including all combinations of inputs are preconditions is not possible.
  - So, instead of doing the exhaustive testing we can use risks and priorities to focus testing efforts.
  - For example-: In an application in one screen there are 15 input fields, each having 5 possible values, then to test all the valid combinations you would need 30 517 578 125 tests.
  - This is very unlikely that the project timescales would allow for this number of tests.
  - So, accessing and managing risk is one of the most important activities and reason for testing in any project.
  - We have learned that we cannot test everything (i.e. all combinations of inputs and preconditions).
  - That is, we must priorities our testing effort using a risk based apaproach.
- Why do not testing everything?
  - Exhaustive testing of complex software applications:
    - Requires enormous resources
    - Is too expensive
    - Takes too long
- Example:
  - System has 20 screens
  - Average 4 menus/screens
  - Average 3 options/menu

    - Average of 10 fields/screen
    - 2 types of input per field
    - Around 100 possible values
- Approximate total for exhaustive testing 20 x 4 x 3 x 10 x 2 x 100 = 480,000 tests
    - Test length = 1 sec then test duration = 17.7 days
    - Test length = 10 sec then test duration = 34 weeks
    - Test length = 1 min then test duration = 4 years
    - Test length = 10 min then test duration = 40 years
- **Early testing:**
    - Testing activities should start as early as possible in the software or system developments life cycle, and should be focused on defined objectives.
    - Testing activities should start as early as possible in the development life cycle.
    - This activity should be focused on defined objectives – outlined in the Test Strategy.
    - Remember from our Definition of Testing, that Testing doesn't start once the code has been written!
- **Defect clustering:**
    - A small number of modules contain most of the defects discovered during pre-release testing, or are responsible for the most operational failures.
    - Defects are not evenly spread in a system.
    - They are 'clustered'.
    - In other words, most defects found during testing are usually confined to a small number of modules.
    - Similarly, operational failures of a system are usually confined to a small number of modules.
    - An important consideration in test prioritisation!
- **Pesticide Paradox:**
    - If the same tests are repeated over and over again, eventually the same sets of test cases will no longer find any new defects.
    - To overcome this "pesticide paradox", the test cases need to be regularly reviewed and revised, and new different tests need to be written to exercise different parts of the software or system to potentially find more defects.
    - Testing identifies bugs, and programmers respond to fix them.
    - As bugs are eliminated by the programmers, the software improves.
    - As software improves the effectiveness of previous tests erodes.
    - Therefore, we must learn, create and use new tests based on new techniques to catch new bugs.
    - N.B It's called the "pesticide paradox" after the agricultural phenomenon, where bugs such as the boll weevil build up tolerance to pesticides, leaving you with the choice of ever-more powerful pesticides followed by ever-more powerful bugs or an altogether different approach.' – Bezier 1995.

- **Testing is Context Dependent:**
  - Testing is basically context dependent.
  - Testing is differently in different contexts.
  - Different kinds of site are tested differently.
  - For example:
    - Safety- critical software is tested differently from an e-commerce site.
  - Whilst, Testing can be 50% of development costs, in NASA's Apollo program it was 80% testing.
  - 3 to 10 failures per thousand lines of code (KLOC) typical for commercial software.
  - 1 to 3 failures per KLOC typical for industrial software.
  - 0.01 failure per KLOC for NASA shuttle code!
  - Also, different industries impose different testing standards.

- **Absence of Error Fallacy:**
  - If the system built is unusable and does not fulfil the user's needs and expectations then finding and fixing defects does not helps.
  - If we build a system and, in doing so, find and fix defects.
  - It doesn't make it a good system.
  - Even after defects have been resolved it may still be unusable and/or does not fulfil the users' needs and expectations.

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

Ans.

Quality Assurance	Quality Control	Tester
Activities which ensure the implementation of processes, procedures and standards in context to verification of developed software and intended requirements.	Activities which ensure the verification of developed software with respect to documented (or not in some cases) requirements.	Activities which ensure the identification of bugs/error/defects in the Software.
Focuses on processes and procedures rather then conducting actual testing on the system.	Focuses on actual testing by executing software with intended to identify bug/defect through implementation to procedures and processes.	Focuses on actual testing.
Process oriented activities.	Product oriented activities.	Product oriented activities.
Preventive activities.	It is a corrective process.	It is a 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.

Q.24 Difference between smoke and sanity?

Ans.

Smoke Testing	Sanity testing
Smoke Testing is performed to ascertain that the critical functionalities of the program is working fine.	Sanity Testing is done to check the new functionality/bugs have been fixed.
The objective of this testing is to verify “stability” of the system in order to with more rigorous testing.	The objective of this testing is to verify the “rationality” of the system in order to proceed with more rigorous testing.
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 usually not documented and is unscripted.
Smoke testing is a subset of Regression testing.	Sanity testing is a subset of Acceptance testing.
Smoke testing exercises the entire system from end to end.	Sanity testing exercises only the particular component of the entire system.
Smoke testing is like General Health Check Up.	Sanity testing is like specialized health check-up.

Q.25 difference between verification and validation?

Ans.

Criteria	Verification	Validation
Definition	The process of evaluating work-products (not the actual final product) of a development phase to determine whether they meet the specified requirements for that phase.	The process of evaluating software during or the end of the development process to determine whether it satisfies specified business requirements.
Objective	To ensure that the product is being built according to the requirements and design specifications. In other words, to ensure that work products meet their specified requirements.	To ensure that the product actually meets the users needs, and that the specifications were correct in the first place. In other words, to demonstrate that the product fulfills its intended use when placed in its intended environments.
Questions	Are we building the product, right?	Are we building the right product?
Evaluation items	Plan, Requirement specs, Design specs, Code, Test cases	The actual product/software.
Activities	Reviews, Walkthroughs, Inspections.	Testing

Q.26 Explain types of performance testing?

Ans. Types of performance testing:

- Load testing
- Stress testing
- Endurance testing
- Spike testing
- Volume testing
- Scalability testing

- **Load testing:**
  - Load testing involves simulating the excepted number of users to ensure the application can handle the anticipated load.

    - Load testing identifies the bottlenecks in breaking the system under various workloads and checks how the system reacts when load is gradually increased.
    - It helps identify performance bottlenecks and is used to determine the systems behaviour under normal and peak conditions.
    - Load testing does not break the system.
- **Stress testing:**
    - It even tests beyond the normal operating point and evaluates how the system works under those extreme conditions.
    - Stress testing is done to make sure that the system would not crash under crunch situations.
    - Stress testing is also known as endurance testing.
    - Stress testing tries to break the system by testing with overwhelming data or resources.
- **Endurance testing:**
    - Endurance testing involves running the system at normal load for an extended period to identify issues that might appear over time, such as memory leaks or performance degradation.
    - Ensure system stability or reliability.
- **Spike testing:**
    - Spike testing evaluates how the system reacts to sudden extreme increases in load.
    - This type of testing helps understand if the system can handle abrupt traffic sprites.
    - Evaluate performance and recovery time.
- **Volume testing:**
    - Volume testing involves testing the systems ability to handle a large volume of data.
    - Identify potential performance issues with data processing and retrieval.
- **Scalability testing:**
    - Scalability testing assesses how well the system can scale up or down in response to varying loads.
    - Evaluate the system's ability to scale.

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

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

- A discrepancy between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition.
- This can be a misunderstanding of the internal state of the software, an oversight in terms of memory management, confusion about the proper way to calculate a value, etc.

- **Defect:**
  - Error found by tester is called defect.
  - Commonly refers to several troubles with the software products, with its external behaviour or with its internal features.
- **Bug:**
  - Defect accepted by development team then it is called bug.
  - A fault in a program which causes the program to perform in an unintended or unanticipated manner.
  - See: anomaly, defect, error, exception, and fault. Bug is terminology of Tester.
- **Failure:**
  - Build does not meet the requirements then it is called failure.
  - The inability of a system or component to perform its required functions within specified performance requirements. See: bug, crash, exception, and fault.

Q.28 difference between priority and severity?

Ans.

Parameters	**Severity**	**Priority**
Definition	Severity is a term that denotes how severely a defect can affect the functionality of the software.	Priority is a term that defines how fast we need to fix a defect.
Parameters	Severity is a basically a parameter that denotes the total impact of a given defect an any software.	Priority is a basically a parameter that decides the order in which we should fix the defects.
Relation	Severity relates to the standards of quality.	Priority relates to the scheduling of defects to resolve them in software.
Value	The value of severity is objective.	The value of priority is subjective.
Changes of value	The value of severity changes continually from time to time.	The value of priority changes from time to time.
Who decides the defects	The testing engineer basically decides a defects severity level.	The product manager basically decides a defects priority level.
Types	There are 5 types of severities: Cosmetic, Minor, Major, Moderate and critical.	There are 3 types of priorities: High, Low and Medium.

Q.29 What is Bug Life Cycle?

Ans. A computer bug is an error, flaw, mistake, failure, or fault in a computer program that prevents it from working correctly or produces an incorrect result. Bugs arise from mistakes and errors, made by people, in either a programs source code or its design.

- The duration or time span between the first-time defects is found and the time that it is closed successfully, rejected, postponed or deferred is called as 'Defect Life Cycle'.
- When a bug is discovered, it goes through several states and eventually reaches one of the terminal states, where it becomes inactive and closed.
- The process by which the defect moves through the life cycle is depicted next slide.
- As you can see from above diagram, a defects state can be divided into Open or Closed.
- When a bug reaches one of the Closed or Terminal states, its lifecycle ends. Each state has one or more valid states to move to.
- This is to ensure that all necessary steps are taken to resolve or investigate that defect. For example, a bug should not move from submitted state to resolve state without having it open.
- In a typical scenario, as soon as a bug is identified, it is logged into the bug tracking system with status as Submitted. After ascertaining the validity of the defect, it is given the "Open" Status.

Q.30 Explain the difference between functional testing and non-functional testing?

Ans.

Functional testing	**Non-functional testing**
Functional testing is performed using the functional specification provided by the client and verifies the system against the functional requirements.	Non-functional testing checks the performance, reliability, scalability and other non-functional aspects of the software system.
Functional testing is executed first.	Non-functional testing should be performed after functional testing.
Manual testing or automation tools can be used for functional testing.	Using tools will be effective for this testing.
Business requirements are the inputs to functional testing.	Performance parameters like speed, scalability are inputs to non-functional testing.
Functional testing describes what the product does.	Non-functional testing describes how good the product works.

Easy to do manual testing.	Tough to do manual testing.
Types of functional testing are: • Unit testing • Smoke testing • Sanity testing • Integration testing • White box testing • Black box testing • User acceptance testing • Regression testing	Types of non-functional testing are: • Performance testing • Load testing • Volume testing • Stress testing • Security testing • Installation testing • Penetration testing • Compatibility testing • Migration testing

Q.31 To create HLR and Test Case of

- **(Instagram, Facebook) only first page**
- **Facebook login page**

Ans. INSTAGRAM

- **HLR:** https://github.com/vaghelaaryan/HLR/blob/main/HLR%20of%20INSTAGRAM.xlsx
- **TEST CASE:** https://github.com/vaghelaaryan/TEST-CASE-/blob/main/INSTAGRAM%20.xlsx

FACEBOOK

- **HLR:** https://github.com/vaghelaaryan/HLR/blob/main/HLR%20of%20FACEBOOK.xlsx
- **TEST CASE:** https://github.com/vaghelaaryan/TEST-CASE-/blob/main/FACEBOOK.xlsx

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

Ans.

SDLC (Software Development Life Cycle)	**STLC (Software Testing Life Cycle)**
SDLC is mainly related to software development.	STLC is mainly related to software testing.
Besides development other phases like testing is also include.	It focuses only on testing the software.
SDLC involves total six phases or steps.	STLC involves only five phases or steps.
In SDLC, development team makes the plans and designs based on the requirements.	In STLC, testing team (Test Lead or Test Architect) makes the plans and designs.

In SDLC, more number of members (developers) are required for the whole process.	In STLC, less number of members (testers) are needed.
Goal of SDLC is to complete successful development of software.	Goal of STLC is to complete successful testing of software.
It helps in developing good quality software.	It helps in making the software defects free.
SDLC phases are completed before the STLC phases.	STLC phases are performed after SDLC phases.
Post deployment support, enhancement, and update are to be included if necessary.	Regression tests are run by QA team to check deployed maintenance code and maintains test cases and automated scripts.
Creation of reusable software systems is the end of the result of SDLC.	A tested software system is the end result of STLC.

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

Ans.

Test Scenarios	**Test Cases**	**Test Scripts**
Test scenario is the functionality that can be tested.	Test cases is the set of actions to be executed to verify particular functionality.	Test script is the set of instructions to test an app automatically using programming.
Test scenario is derived from the business requirements and software requirements.	Test cases is the mostly derived from the test scenarios.	Test scripts is mostly derived from the test cases.
Helps to test end to end functionality in agile way.	Helps in exhaustive testing of a system.	Helps to test specific thing repeatedly.
It is mostly focused on what to test.	It is focused on what to test and how to test.	It is focused on the excepted result.
It includes an end to end functionality to be tested.	It includes a test steps, data, excepted result for testing etc.	It includes different commands to develop test script.
Allows the quickly assessing the testing scope.	Allows the detecting the errors and defects.	Allows carrying out an automatic execution of test cases.
The main task is to check the full functionality of a software application.	The main task is to verify compliance with the standards, guidelines, and customer requirements.	The main task is to verify that nothing is skipped and results are true as the desired testing plan.
Takes less time and fewer resources to create.	Requires more resources and time.	Require less time for testing but more resources for scripts creating and updating.

Q.34 Explain what test plan is? What is the information that should be covered?

Ans. A test plan is a detailed document that outlines the strategy, approach, resources, schedule and scope of testing activities for a project or product. It serves as a blueprint for the testing process, ensuring that all aspects of the testing are considered and documented. A well-crafted test plan helps to ensure that testing is systematic, thorough and aligned with project goals.

- A comprehensive test plan typically includes the following information:
    - **Purpose:** why the test plan is needed and what it aims to achieve
    - **Scope:** What will be tested and what won't be tested.
    - **Test strategy:** different stages of testing (unit, integration, system). Kinds of testing (functional, performance). General method of testing (black-box, white-box).
    - **Features to be Tested:** Specific features and functionalities that will undergo testing.
    - **Features not to be Tested:** Features or Functionalities that won't be tested.
    - **Test Design and Execution:** How test cases will be created. Data needed for testing. Hardware and Software needed.
    - **Test Items:** List of things (features or components) that will be tested.
    - **Schedule:** timeline for testing activities.
    - **Pass/Fail criteria:** How to decide if a test has passes or failed.
    - **Resources:** people involved and their roles and any training needed.
    - **Test deliverables:** Documents and reports that will be produced.
    - **Approvals:** Who will receive and approved the test plan.
    - **Risks and contingencies:** Possible problems and backup plans.
    - **Test Environment:** specifies the hardware, software, and other resources needed for testing.

Q.35 What is priority?

Ans. priority is Relative and Business Focused. Priority defines the order in which we should resolve a defect. Should we fix it now, or can wait it? 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.

- **For example:** If the company name is misspelled in the home page of the website, then the priority is high and severity is low to fix it.
- **Priority can be following types:**
    - **Low:** The defect is an irritant which should be repaired, but repair can be deferred until after more serious defects has been fixed.
    - **Medium:** The defect can be resolved in the normal course of development activities. It can wait until a new build or version created.

- **High:** The defect must be resolved as soon as possible because the defect is affecting the application or the product severely. The system cannot be used until the repair has been done.
- **Critical:** Extremely urgent, resolve immediately.

Q.36 What is severity?

Ans. 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.

- **For example:** if an application or web page crashes when a remote link is clicked, in this case clicking the remote link by a user is rare but the impact of an application crashing is server. So, the severity is high but priority is low.
- **Severity can be following types:**
  - **Critical:** The defect that results in the termination of the complete system or one or more component of the system and causes extensive corruption of the data. The failed function is unusable and there is no acceptable alternative method to achieve the required results then the severity will be stated as critical.
  - **Major (High):** The defect that result in the termination of the complete system or one or more component of the system and causes extensive corruption of the data. The failed function is unusable but there exists an acceptable alternative method to achieve the required results then the severity will be stated as major.
  - **Moderate (Medium):** The defect that does not result in the termination, but causes the system to produce incorrect, incomplete or inconsistence results then the severity will be stated as moderate.
  - **Minor (low):** The defect that does not results in the termination and does not damage the usability of the system and the desired result can be easily obtained by working around the defects then the severity is stated as minor.
  - **Cosmetic:** The defect that is related to the enhancement of the system where the changes are related to the look and field of the application then the severity is stated as cosmetic.

Q.37 Bug categories are…

Ans. Bug categories can very depend on the context, but generally, bugs in software development are categorized based on their impact, severity and the area of the system where they occur. Here are some common bug categories:

- **Critical Bugs:** These bugs cause the system to crash, result in data loss, or make the software unusable. They often require immediate attention.
- **Major Bugs:** These bugs significantly impair the functionality of the software or cause major inconvenience to users. They need to be fixed urgently but might not be as severe as critical bugs.
- **Minor Bugs:** These bugs have minimal impact on the functionality of the software and may cause minor inconveniences or cosmetic issues. They are usually fixed in routine maintenance.
- **Cosmetic Bugs:** These bugs don't affect the functionality of the software but impact its appearance or user experience. They are often low priority.
- **UI/UX Bugs:** These bugs affect the user interface or user experience, such as layout issues, inconsistent design elements, or confusing user interactions.
- **Performance Bugs:** These bugs degrade the performance of the software, causing it to run slowly or consume excessive resources.
- **Security Bugs:** These bugs create vulnerabilities in the software that can be exploited by attackers to gain unauthorized access, manipulate data, or disrupt operations.
- **Compatibility Bugs:** These bugs occur when the software behaves differently on different platforms or fails to work with other software components as excepted.
- **Regression Bugs:** These bugs appear after a new version of the software is released and cause features that previously worked correctly to malfunction.
- **Documentation Bugs:** These bugs involve errors or inconsistences in the software documentation, such as missing or outdated instructions, misleading information, or grammatical errors.

Q.38 Advantage of Bugzilla?

Ans. Bugzilla is a defect tracking tool; however, it can be used as a test management tool.

- Bugzilla is an open source issue/bug tracking system that allows developers effectively to keep track of outstanding problems with their product.
- **Advantages are as mentioned below;**
    - It is easy in usage and its user interface is understandable for people without technical knowledge.
    - It reports in a verity of formats and types.
    - It can track the time that is taken to fix the bug.
    - Bugzilla comes with both basic and advanced searching mechanism. Using this you can search the details of bugs as you wish.
    - It can report in multiple type and format like charts, graph or HTML, CSV, XML.

Bugzilla have duplicate bug detection features as it automatically tracks the bugs similar to the one you are searching for.

Q.39 Difference between priority and severity?

Ans.

Parameters	Severity	Priority
Definition	Severity is a term that denotes how severely a defect can affect the functionality of the software.	Priority is a term that defines how fast we need to fix a defect.
Parameter	Severity is basically a parameter that denotes the total impact of a given defect on any software.	Priority is basically a parameter that decides the order in which we should fix the defects.
Relation	Severity relates to the standards of quality.	Priority relates to the scheduling of defects to resolve them in software.
Value	The value of severity is objective.	The value of priority is subjective.
Change of value	The value of severity changes continually from time to time.	The value of priority changes from time to time.
Who decides the defects	The testing engineer basically decides a defects severity level.	The product manager basically decides a defects priority level.
Types	There are 5 types of severities: Cosmetic, Minor, Moderate, Major and Critical.	There are 3 types of priorities: High, Medium and Low.

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

Ans. Agile development is a broad approach to software development that emphasizes iterative development, flexibility, collaboration, and customer feedback. There are several methodologies within the Agile framework.

- **Scrum:** Scrum is perhaps the most well-known agile methodology. It divides work into time-boxed iterations called sprints, typically lasting two to four weeks. Scrum teams have specific roles, including a product owner, Scrum master and Development Team.
- **Kanban:** Kanban focuses on visualizing the work flow and limiting work in progress (WIP). Work items are represented as cards on a board, moving through columns that represent different stages of the development process. Kanban provides a more continues flow of work compared to Scrums iterations.

- **Extreme Programing (XP):** XP emphasizes technical excellence and customer satisfaction. It includes practices such as test-driven development (TDD), pair programing, continues integration and frequent releases. XP encourages close collaboration between developers and customers.
- **Lean Software Development:** Lean principles, derived from manufacturing, are applied to software development. Lean focuses on eliminating waste, amplifying learning, empowering teams, and delivering as fast as possible. It emphasizes delivering value to customers quickly.
- **Crystal:** Crystal methodologies are a family of agile methodologies developed by Alistair Cockburn. Each Crystal methodology is tailored to specific project characteristics, such as team size, system criticality and project priorities. Crystals range from Crystal Clear (for small teams) to Crystal Diamond (for large, mission critical projects).
- **Dynamic Systems Development Method (DSDM):** DSDM is an Agile methodology that provides a framework for delivering projects quickly and efficiently. It focuses on frequent delivery of products in a controlled and structured manner. DSDM emphasizes active user involvement and prioritizes functionality.
- **Feature-Driven Development (FDD):** FDD is an iterative and incremental software development methodology. It's based on the idea of designing and building features incrementally. FDD is suitable for large teams and complex projects, and it emphasizes domain modelling, iterative development, and feature ownership.

Q.41 Explain the difference between Authorization and Authentication in Web testing. What are the common problems faced in Web testing?

Ans.

Authentication	**Authorization**
In the authentication process, the identity of the users is checked for providing the access to the system.	While in authorization process, the persons and users' authorities are checked for accessing the resources.
In the authentication process, users or persons are verified.	While in this process, users or persons are validated.
It is done before the authorization process.	While this process is done after the authentication process.
It needs usually the user's login details.	While it needs the user's privilege or security levels.
Authentication determines whether the person is user or not.	While it determines what permission does the user have?
Generally, transmit information through and ID token.	Generally, transmit information through an Access token.

The Open ID Connect (OIDC) protocol is an authentication protocol that is generally in charge of user authentication process.	The OAuth 2.0 protocol governs the overall system of user authorization process.
Popular Authentication Techniques- • Password-Based Authentication • Password less authentication • 2FA/MFA (Two-Factor Authentication/ Multi-Factor Authentication) • Single sign-on (SSO) • Social authentication	Popular Authorization Techniques- • Role-Based Access Control (RBAC) • JSON web token (JWT) Authorization • SAML Authorization • OpenID Authorization • OAuth 2.0 Authorization
The authentication credentials can be changed in part as and when required by the user.	The authorization permission cannot be changed by user as these are granted by the owner of the system and only he/she has the access to change it.
The user authentication is visible at user end.	The user authorization is not visible at the user end.
The user authentication is identified with username, password, face recognition, retina scan, fingerprints etc.	The user authorization is carried out through the access rights to resources by using roles that have been pre-defined.

Common problems faced in web testing:

- **Performance issues:** Testing for load times, responsiveness and performance under stress. Identifying and resolving bottlenecks that cause slow performance.
- **Security Vulnerabilities:** Ensuring the web application is secure from common threats like SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF) etc.
- **Usability issues:** Verifying that the user interface is intuitive and user friendly. Ensuring that navigation is smooth and users can easily complete their tasks.
- **Cross-Browser Compatibility:** Different Browsers (Chrome, Firefox, Safari etc.) render web pages differently due to variations in their rendering engines. Ensuring that the web application works consistently across all major browsers can be challenging.
- **Functionality issues:** Verifying that all features and functionalities of the web application work as intended.
- **Database Issues:** Ensuring that the web application interacts correctly with the database. Testing for data integrity, proper handling for data transaction and checking for potential database-related issues.

Q.42 To create HLR and Test Case of Web Based (WhatsApp web, Instagram)

1. WhatsApp Web:

- **HLR:** https://github.com/vaghelaaryan/HLR/blob/main/HLR%20of%20WHATSAPP%20WEB.xlsx

- **TEST CASE:** https://github.com/vaghelaaryan/TEST-CASE-/blob/main/WHATSAPP%20WEB.xlsx

2. Instagram Web:

- **HLR:** https://github.com/vaghelaaryan/HLR/blob/main/HLR%20of%20INSTAGRAM.xlsx
- **TEST CASE:** https://github.com/vaghelaaryan/TEST-CASE-/blob/main/INSTAGRAM%20.xlsx

3. Art of Testing:

- **HLR:** https://github.com/vaghelaaryan/HLR/blob/main/HLR%20of%20ART%20OF%20TESTING.xlsx
- **TEST CASE:** https://github.com/vaghelaaryan/TESTCASE/blob/main/ART%20OF%20TESTING.xlsx

Q.43 Write a scenario of only WhatsApp chat messages.

Ans. Verify that the user can download the WhatsApp application.

- Verify that user can register with new mobile number.
- Verify that for a new mobile number user will get a verification code on his mobile and filling in the same verifies the new user account.
- Verify that the chat window contains the entire chat lists.
- Verify that the chat window shows the contact numbers whose numbers are not saved in mobile
- Verify that clicking on one chat contact then a new window should open with history.
- Verify that user can see their all saved mobile numbers in WhatsApp chat.
- Verify that user can see their search option on chat window.
- Verify that user can see their WhatsApp logo is clearly visible or not.
- Verify that user can see the QR code is visible or not on chat window.
- Verify that user can see their camera functionality is visible or not on chat window.
- Verify that chat window displayed with all contacts with DP or without DP.
- Check the maximum number of incorrect attempts allowed while filling out the verification code.
- Verify that the chat window is displayed on the group chat list.
- Verify that the user can see all delivered and received messages.
- Verify that the user can see their read or send time of messages.
- Verify that chat window displays the last updated chatting time.
- Verify that the user can see the name of all contacts on the chat window.
- Verify that the user can send and receive text messages in the individual chat box.
- Verify that the user can send and receive the documents in the individual chat box.

- Verify that the user can send and receive photos in the individual chat box.
- Verify that the user can send and receive videos in the individual chat box.
- Verify that the user can send and receive audio in the individual chat box.
- Verify that the user can send and receive emotions icons in individual chat box.
- Verify that the user can send and receive contacts in individual chat box.
- Verify that the user can send and receive Location in individual chat box.
- Verify that the user can send and receive GIF in individual chat box.
- Verify that the user can send and receive Stickers in individual chat box.
- Verify that the user can delete text, video, audio, locations, documents, in individual chat box.
- Verify that the user can send and receive emojis in individual chat box.
- Verify that the user can send and receive recorded voice mail in individual chat box.
- Verify that the user can delete the entire chat history in individual chat box.
- Verify that the user is able to see contact details in individual chat box.
- Verify that the user is able to share images, links and documents from media in individual chat box.
- Verify that the user is able to search specific chat history using search option in individual chat box.
- Verify that the user can send and receive HD photos in individual chat box.
- Verify that the user can check the payment details.
- Verify that the user is able to mute the notification in individual chat box.
- Verify that user is able to change the wallpaper.
- Verify that user is able to change the wallpaper in individual chat box.
- Verify that the user has options like report, block, clear chat, export chat and add shortcut.
- Verify that registering an existing mobile number for new user account.
- Verify that user can set DP and status on WhatsApp.
- Verify that user can update the DP and WhatsApp.
- Verify that the user can send messages to any individual from his contact lists.
- Verify that user can create group.
- Verify that user can send money to other with WhatsApp payment.
- Verify that user can video call any individual from his contact list.
- Verify that user can share and receive contact with another person.
- Verify that user can set how many people can see status.
- Verify that user can create a group by adding multiple people from his contact list.
- Verify that the user makes WhatsApp calls to the person in his contact list.
- Verify that the user can receive WhatsApp calls from the person in his contact list.
- Verify that the user can archive chats in an individual or group chat.
- Verify that user can set a password to unlock WhatsApp.
- Verify that the user can unlock WhatsApp with fingerprint sensor.
- Verify that user can create a community with another persons.
- Verify that the user can see the multiple channels in WhatsApp status page.

- Verify that the user can create and find channels in WhatsApp.
- Verify that the user can see channels functionality like explore, popular and new etc.
- Verify that the user can create new broadcast in WhatsApp.
- Verify that the user can link devices with WhatsApp link or QR code.
- Verify that the users can block a user to prevent any message from getting received from the blocked contacts.
- Verify that the user can share their WhatsApp status with Facebook account.
- Verify that the user can set the theme like dark and white.
- Verify that the user can set their avatar in WhatsApp.
- Verify that the user can set app language like Hindi and English etc.
- Verify that the user can see their account details in WhatsApp settings.
- Check that in account details user can see security notifications, verify that the user can set their security notifications.
- Verify that the user can see their email address in account details.
- Verify that the user can see their two-step verification in account details.
- Verify that the user can change the number in account details.
- Verify that the user can add and delete accounts.
- Verify that the user can hide their online and last seen option.
- Verify that the user can send their live location in individual chat box.
- Verify that the user can hide their profile photo.
- Verify that the user can see their blocked contacts.
- Verify that user can have chat lock option for particular person in chat box.
- Verify that the user can marks chats as favourites and access all chats marked as favourites from the ‘favourites’ section.
- Verify that the user can send and receive chats in the secondary languages available.
- Verify that users can clear their complete chat history in an individual or group chat.
- Verify that ‘chats’ window contains all the chat list with DP and name and last message preview of the other person with whom chat was initiated.
- Verify that the user can check the message delivered and read the time for a message in the ‘Message info’ section.
- Verify that the user can send and receive the message in group chats.
- Verify that the user can send their photo with document to see clear picture.
- Verify that the user can call group video call with multiple people.
- Verify that the user can change the group DP by group admin.
- Verify that the user can set the wallpaper for individual chats.
- Verify that the user can delete the message with delete everyone or delete for me option is available.
- Verify that proper messages delivered or not.
- Verify that user can check another person profile and properly show or not.
- Verify that user can use microphone for send voice messages.
- Verify that user can record message delivered properly or not.
- Verify that emoji button properly shows or not.

- Verify that user can check camera button is properly working or not.
- Verify that user can check Gallery option and check the proper photos and videos show or not.
- Verify that user can check proper map visible or not in location option.
- Verify that user can check Contact option for their phones all contacts show or not.
- Verify that user can check Poll option properly work or not.
- Verify that user can check search option for any message search.
- Verify that user can check Disappearing messages for make messages in chat disappear or not.
- Verify that user can check message timer in disappearing messages properly work or not.
- Verify that user can send online payment in WhatsApp.
- Verify that user can check simple call properly work or not.
- Verify that user can check video call properly work or not without mobile data.
- Verify that user can check simple call properly work or not without mobile data.
- Verify that user can check mute notification option properly work or not.
- Verify that user can select any message and check information such as when the message was seen and delivered.
- Verify that select any message and user can copy that message or not.
- Verify that select any message and user can pin that message or not.
- Verify that select any message and user can star that message or not.
- Verify that select any message and user can forward and backward message or not.
- Verify that select any message and user can delete message and show the emojis or not.
- Verify that user can pin on chats.
- Verify that user can edit their message only when he sends it and cannot edit it after some time.
- Verify that user can check Keyboard is properly work or not.
- Check if we hide a message, user can check whether locked chats proper work or not.
- Verify that user can search any chats.
- Verify that user can check the setting of the entire WhatsApp or not.
- Verify that user can check chat with business properly work or not.
- Verify that user can check payment methods working or not.
- Verify that user can see archived chats.
- Verify that user can check status privacy.
- Verify that user can create call link or not.
- Verify that user can check different type of font in type a status.
- Verify that user can see any number select and then mute their status or not.
- Verify that user can check calls are end-to-end encrypted or not.
- Verify that user can check who can see my personal information button properly work or not.

- Verify that user can check default message timer.
- Verify that user can check read receipts for messages properly work or not.
- Verify that user can check advanced for protect IP address in calls or not.
- Verify that user can check delete avatar or not.
- Verify that user can check media visibility.
- Verify that font size can be changed such as small, large etc.
- Verify that user can check notification option properly working or not.
- Verify that user can conversation tones means play sounds for incoming and outgoing messages.
- Verify that user can check should it vibrate or not when message comes.
- Verify that user can check media auto download when using mobile data for photos.
- Verify that user can check when connected on WIFI proper work or not.
- Verify that user can check voice messages are always automatically downloaded or not.
- Verify that user can check help centre, get help and contact us properly work or not.
- Verify that user can check terms and privacy policy.
- Verify that user can check WhatsApp is not working without internet or not.
- Verify that user can see all contacts profile.
- Check if WhatsApp is open in another device, we can log out it from mobile phone or not.
- Verify that user can check WhatsApp app colour.
- Verify that user can check contacts verify security code properly work or not.
- Verify that user can check when roaming for no media.

Q.44 Write a scenario of Pen.

Ans. Verify that the type of pen like ballpoint pen, ink pen or gel pen.

- Verify the which type of colour of pen like black, blue, green and yellow etc.
- Verify that user can write clearly on different types of papers.
- Verify pen is with the cap or without the cap.
- Verify the which type of pen ink is like blue, green and red etc.
- Verify the pen writes smoothly on the surface or paper.
- Verify that user can check the pens ink is licking or not.
- Verify the written text by pen is waterproof or not.
- Verify the written text by pen is erasable or not.
- Verify the pen is support for multiple refills or not.
- Verify the pens point is breakable or not when is throw the surface.
- Verify the strength and length of the pens outer body.
- Verify the size of tip of the pen.
- Verify that the user can change the refill of the pen easily if pen is gel pen.
- Verify that the user is able to refill the pen with all the types of ink.

- Verify the size of the tip of pen.
- Verify the grip of the pen, and whether it provides friction for the user to comfortably grip the pen.
- Verify that the mechanism to refill the pen is easy to operate for ink pens.
- Verify that the functioning of the pen by applying normal pressure during writing.
- Check if we throw the pen on surface, verify that pens point is break or not.
- Verify that user can check pens soft or hard body.
- Verify that it should be checked whether the pen breaks at the bottom or not.
- Verify that user can check which company's pen.

Q.45 Write scenario of Pen Stand.

Ans. Verify that different types of colour of pen stand like red, black and green etc.

- Verify that different types of pen stand like rotating and multi-functional etc.
- Verify that we can put pen and pencil both in pen stand.
- Verify that pen stand holding multiple pens.
- Verify that pen stands size, width and length is proper or not.
- Verify that pen stand is breakable or not when we throw to the surface.
- Verify that different types of material of pen stand like wood, metal and plastic etc.
- Verify that pen stand place on a flat surface.
- Verify that we can add multiple pens in pen stand until it falls over.
- Verify that how many pens are holding by pen stand before falling.
- Verify that we can throw the pen in pen stand from different heights.
- Verify that when we throw the pen from different heights, check that pen stand is damage or not.
- Verify that user can check stands shape.
- Verify that it should be checked whether the stands breaks at the bottom or not.
- Verify that it should be checked whether bottom surface.
- Verify that it should be checked whether stand Is transparent or not.
- Verify if the material is resistant to scratches, stains, or other forms of damage that may occur during regular use.
- Verify Ensure that the pen stand remains stable and balanced, even when loaded with items unevenly.
- Verify that the pen stand is sturdy and stable.
- Verify Ensure that the pen stand does not scratch or damage the desk surface during normal use.
- Verify if the base of the pen stand provides adequate support to prevent tipping or wobbing.
- Consider user reviews and ratings to understand overall satisfaction with the pen stand.
- Verify any visible defects or damages in the material.

Q.46 Write scenario of Door.

Ans. Verify is the door is single door.

- Verify that doors open inwards or outwards.
- Verify that the types of door like glass, wood and steel doors etc.
- Verify that the types of door lock like finger print, handle lock, password lock and face sensor lock etc.
- Verify that the door is sliding door or not.
- Verify that the different types of colours of door.
- Verify that the doors handle is steel, wood and glass etc.
- Verify that the dimensions of doors are as per the specifications.
- Verify that the door has a stopper or not.
- Verify that the door is closing automatically or not.
- Verify that the doors outer body works in different types of whether.
- Verify that the door is making some noise when is opened or closed.
- Verify that the doors width, length and size as per specifications.
- Verify that the door is having peak-hole or not.
- Verify that how much force to pull or push required when open or closed the door.
- If we applied more amount of force to door for open, then check is breakable or not.
- Verify the door condition in different climatic conditions like temperature, humidity etc.
- Verify the position, quality and strength of hinges.
- Verify that the door is bi-folded or rotated door.
- Verify the material used for the doors body.
- Verify that colour of the door is specified or nor.
- Verify the number of locks in the door interior side or exterior side.
- Verify that the doors all parts like hinges and handles are fixed properly or not.
- Verify that we can use handle to open the door.
- Verify that door is transparent base or not.
- Verify that door functions open or close through biometrics machine.

Q.47 Write scenario of ATM.

Ans. Verify that power backup should be present at ATM.

- Verify that card reader should be present.
- Verify that receipt printer should be available and working properly.
- Verify that cash dispenser is working as excepted.
- Verify the type of ATM machine, if it has a touch screen, both keypad and buttons only, or both.
- Verify that user can check credit card swipe machine properly work or not.

- Verify that user can check withdraw option properly work or not.
- Verify that user can check deposit option properly work or not.
- Verify that the key pad should be working and covered.
- Verify that buttons are displayed on screen of ATM machine.
- Verify the font of text on the screen, it should be clearly visible.
- Verify that when card is inserted in ATM, pin should be asked from user.
- Verify that user can check current account option work or not.
- Verify that user can check saving account option properly work or not.
- Verify that user can check No print option properly work or not.
- Verify that user can type pin or not.
- Verify that user can pin change option or not.
- Verify that when user enters incorrect pin for a particular number of times, the card is blocked.
- Verify that when user enters correct pin, the user's details should be displayed on ATM screen.
- Verify that if doesn't enter amount in round off digits, error message is displayed.
- Verify that how much time has taken in a transaction.
- Verify that the user can check withdraw 10000 option is properly work or not.
- Verify that the user can check machine body.
- Verify that the user can check all buttons properly work or not.
- Verify that the user can check pin keyboard properly work or not.
- Verify how much time is taken system to logout user.
- Verify that message is displayed when the cash in ATM is finished.
- Verify that user is presented with an option to select language of operation.
- Verify that error message is displayed when entered amount is greater than account balance.
- Verify that user can check which bank ATM machine.
- Verify that screen properly visible or not.
- Verify that the user's session timeout is maintained.
- Check that the correct amount of money gets withdrawn as entered by the user for cash withdrawal.
- Verify that withdraw limits are set or not.
- Verify that user can check current balance properly show or not.
- Verify that the touch of the ATM screen is smooth and operational.
- Verify that user is able to use card of another bank on the ATM.
- Verify the functionality by entering a wrong pin number for a particular number of times.
- Verify the card ATM machine functionality by inserting an expired card.

Q.48 When to used Usability Testing?

Ans. All fields on page (For Example, Text box, Radio options, drop-down lists) should be aligned properly.

- The user should not be able to type in drop-down select lists.
- Tab and shift +Tab order should work properly.
- All buttons on a page should be accessible by keyboard shortcuts and the user should be able to perform all operations using a keyboard.
- All buttons on a page should be accessible by keyboard shortcuts and the user should be able to perform all operations using a keyboard…
- All pages should have a title.
- Conformation messages should be displayed before performing any update and delete operation.
- Hourglass should be displayed when the application is busy.
- Page text should be left-justified.
- The user should be able to select only one radio option and any combination for checkboxes.

Q.49 What is the procedure for GUI Testing?

Ans. Graphical User Interface (GUI) testing is the process of testing the systems GUI of the system under test. GUI testing involves checking the screens with the control like menus, buttons, icons, and all types of bars- tool bar, menu bar, dialog boxes and windows etc.

- **WHAT DO YOU CHECK IN 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 Colour 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.

Q.50 Write a scenario of Microwave Owen.

Ans. Verify that type of microwave Owen like solo, grill or convection.

- Verify that user can check which company model.
- Verify that the company name is properly displayed or not.
- Verify that user can check old model or new model.

- Verify that the brand logo is properly displayed on the microwave own or not.
- Verify that user can check machine body type.
- Verify that which material used in this machine.
- Verify the size of the microwave oven.
- Verify the colour of the microwave oven.
- Verify that all function properly works or not.
- Verify that the oven heats the food at the desired temperature properly.
- Verify that the oven heats food at the desired temperature within a specified time duration.
- Verify that the capacity of the microwave oven.
- Verify that the compact design of the microwave oven.
- Verify the ovens functioning with the maximum and minimum attainable temperature.
- Verify that the power cord of the oven is long enough.
- Verify that glass is turn table or not.
- Verify that weight of the microwave oven.
- Verify that the ovens place rotation speed is optimal and not too high to spoil the food kept over it.
- Verify that the usage instruction or user manuals have clear instructions.
- Verify that dimensions of the microwave oven.
- Verify that voltage of the microwave oven.
- Verify that the temperature regulator is smooth to operate.
- Verify that the temperature regulator works correctly.
- Check the ovens functionality with different kinds of food like solid and liquid.
- Verify that the ovens door gets closed properly.
- Verify that the ovens door opens smoothly or not.
- Verify that the next written over the ovens body is clearly readable.
- Verify that batteries are required or not.
- Verify that food is properly reheating or not.
- Verify that glass rotation speed is as excepted.
- Verify that disconnecting power while cooking is in progress.
- Verify that different kind of food at different temperature.
- Verify that the alarm sound system is properly working or not.
- Verify that food is grilled properly or not.
- Verify that all buttons are properly worked or not.

Q.51 Write a scenario of Coffee Wending Machine.

Ans. verify that user can check outer body.

- Verify that user can check inner body.

- Verify that the dimensions of the coffee vending machine are as per the specification.
- Verify that the machines body colour as well brand is correctly visible and as per the specification.
- Verify that the switch on/off button properly visible or not.
- Verify that there is a description book with machine or not.
- Verify that the input mechanism for coffee ingredients- milk, water, coffee beans/powder.
- Verify that the quantity of hot water, milk, coffee powder per serving is correct.
- Verify that machine material like metal, steel or etc.
- Verify that machine is with wire or wireless.
- Verify that coffee should not leak when not in operation.
- Verify that amount of coffee served in single-serving is as per specification.
- Verify that the user can set the temperature in the machine.
- Check the steam hot water button.
- Verify that the digital display displays correct information.
- Verify for the indicator lights when the machine is switched on-off.
- Check the machine has a grinder featured or not.
- Verify that user can check steam wand feature or not.
- Verify that the functioning of all the buttons work properly when pressed.
- Verify that user can check automatic machine is properly work or not.
- Verify that each button has an image/text with it, indicating the task it performs.
- Verify that system should display an error when it runs out of ingredients.
- Verify that pressing the coffee button multiple times leads to multiple serving of coffee.
- Verify that there is the passage for residual/extra coffee in the machine.
- Verify that machine should not make to much sound when in operation.
- Verify the performance of the machine when used continuously until the ingredients run out of the requirements.
- Verify the functioning of coffee machine with a lesser or higher voltage than required.
- Verify for guarantee and warranty of the machine, in case provided.
- Verify for the cup holder dimension as per specification/ or market standard.
- Verify the functioning of the coffee machine if the ingredient container’s capacity is exceeded.
- Verify the performance of the machine when used continuously until the ingredients run out of the requirements.
- Verify that user can check all buttons.
- Check the machine has a mixing chamber feature or not.
- Verify that user can check battery properly work or not.
- Verify that user can check drink option.
- Verify that user can check water quantity.

- Verify that how many years is the warranty.
- Verify that user can check magnetic sealing door properly open or not.
- Check the machine is reusable filter or not.

Q.52 Write a scenario of Chair.

Ans. Verify that the chair is moveable or not.

- Verify that different types of colour of a chair like blue, red and black.
- Verify that the chair is comfortable for sitting or not.
- Verify that the chair has wheel or not.
- Verify that the different types of material used like leather, wood and metal etc.
- Verify that the back support of the chair.
- Verify that the support for hands in the chair.
- Verify that the cushion is provided for the chair or not.
- Verify that the after sitting on the chair the legs are level to the floor.
- Verify the height of the chairs seat from the floor.
- Verify the weight of the chair as per the specification.
- Verify the dimension of the chair as per the requirements.
- Verify that the chair is stable to take human load.
- Verify if the chairs material is brittle or not.
- Verify that the chair can handle different weights.
- Verify that the all parts of the chair work well.
- Verify that the roll of the chair on different floors then we check the wheels work well or not.
- Verify the chair is spin easily without the noise or not.
- Verify that usability of the chair as an office chair, normal household chair.
- Verify condition when washed with water or effect of water on chair.
- Verify whether distance between all legs it should be same.
- Verify by design because there is a chair with 3 legs or modern design when two and two are different distance.
- Verify the wheels are turning 360 degrees or not.
- Verify that user can check chair colour.
- Verify that user can check the seat of the chair go up or down.
- Verify that wheels are oiled well or not.
- Verify edged of chair is needed to be in round shape or sharp.
- Verify that the main function check stability.
- Verify that chair can be stored with other chairs on top or bottom.

Q.53 To Create Scenario (Positive and Negative)

1. Gmail (Receiving Mail)

2. Online shopping to buy product (flip kart).

Ans. 1. Gmail (Receiving Mail):

- Verify that a newly received email is displayed as highlighted in the inbox section.
- Verify that a newly received email has correctly displayed sender email id or name, mail subject and mail body (trimmed to a single line).
- Check if the compose button is available and clickable.
- Ensures users can create and send a new email successfully.
- Verify that the user receives the email in their inbox.
- Verify that on clicking the newly received email, the user is navigated to email content.
- Verify that users can add multiple recipients to an email.
- Verify that starred or flagged emails are visually marked in the inbox.
- Verify that user can check the email contents are correctly displayed with the desired source format.
- Verify that any attachments are attached to the email and are downloadable.
- Verify that the attachments are scanned for viruses before download.
- Verify that user can check all the emails marked as read are not highlighted.
- Verify the functionality of the primary, social, promotions and other categorized tabs.
- Verify that the inbox supports a conversation view for email threads.
- Verify to check if the user can delete the email from the inbox, which is then moved to the spam folder.
- Verify that the all received emails get piled up in the ‘inbox’ section and get deleted in cyclic fashion based on the size availability.
- Verify that the email can be received from non-Gmail email ids like- yahoo, Hotmail etc.
- Verify that all the emails read as well as unread have a mail read time appended at the end on the email list displayed in the inbox section.
- Verify that the count of unread emails is displayed alongside ‘inbox’ text in the left sidebar of Gmail.
- Verify that the email recipients in cc are visible to all users.
- Verify that the email recipients in bcc are not visible to the user.
- Verify that the unread email count increases by one on receiving a new email.
- Verify that the unread email count decreases by one on reading an email (marking an email as read).

2.Online shopping to buy product (flip kart):

- Verify that the initiation of the buy flow.
- Verify that the accuracy of product details.
- Verify that the responsiveness to changes in quantity.
- Verify that the accuracy and function of applied discounts.
- Verify that the behaviour with out of stock items.

- Verify that on the product page, the user can select the desired attribute of the product e.g. size, colour etc.
- Verify that the adding products from different categories.
- Verify that the user can successfully buy more than one product that were added to his/her cart.
- Verify that users can add products to the which list.
- Verify that the user can not add more than the available inventory of the product.
- Verify that the user can see the previously added products on the cart page, after signing in to the application.
- Verify that the cash on delivery option of payment is working properly or not.
- Verify that the limit to the number of products a user can buy is working correctly. Also, an error message can display, preventing the user from buying more than the limit.
- Verify that the different prepaid methods of payments are working properly or not.
- Verify that the delivery can be declined during checkout for the places where shipping is not available.
- Verify that the behaviour with saved addresses.
- Verify that the product return functionality works correctly.
- Verify the responsiveness of the payment selection step.
- Verify that the visibility and accuracy of delivery options.
- Verify that the cancelling the payment process midway.
- Verify that the accuracy of order summary details.
- Verify that the redirection on the order conformation page.
- Verify the responsiveness to changes in user location.

Q.54 Write a Scenario of Wrist Watch.

Ans. Verify the different types of watch like analog and digital etc.

- Verify the different types of colour like red, black and green etc.
- Verify the material of the watch and strip.
- Verify the weight of the watch.
- Verify that the watch is waterproof or not.
- Verify the watch is smart watch or not.
- Verify the different types of company Sonata, Titan and Rolex etc.
- Verify that watch is for man and women.
- Verify that the watch dial number is properly visible or not.
- Verify that the watch is having date and time display or not.
- Verify that the alarm sound is ticking or not.
- Verify that the shape of dial is as per the requirements.
- Verify that the battery requirement of the watch.

- Verify that the dimension of the watch as per the requirements.
- Verify that the brand of watch and check if it visible on display.
- Verify that the date, day and time is properly visible on display.
- Verify that the dials glass and plastic is resistance to minor scratches or not.
- Verify that the colour of the text displayed in the watch like date, day, time and other information.
- Verify that the watch is comfortable to wear.
- Verify that all features of the watch work properly.
- Verify that the watch's appearance and design.
- Verify that the how the watch affects the environments.
- Verify that the compare watch with similar watch products.
- Verify that the chain material and belt for damage.
- Verify that the belt or chain used is comfortable or not and its length.
- Verify that the format of the watch 12 hours or 24 hours.
- Verify that the clock is having stopwatch, timers, and alarm functionality or not.

Q.55 Write a Scenario of lift (Elevator).

Ans. Verify that the lift has different types of colours.

- Verify that the different types of door elevator door and telescopic doors.
- Verify the dimension of the lift.
- Verify the type of door of the lift as per the requirements.
- Verify the lift is showing proper information like fan, light and buttons.
- Verify that the lifts fan is properly working or not.
- Verify that the lifts light is properly working or not.
- Verify that the lifts all buttons properly working or not.
- Verify that the lifts close and open button is properly working or not.
- Verify that the lift is automatically close or open or not.
- Verify that how much weight or members holding by the lift.
- Verify the type of metal used in the in the lift interior and exterior.
- Verify that the maximum weight capacity of left.
- Verify that the height, width and length of lift.
- Verify that the buttons are present according to the number of floors in building.
- Verify that blind person should be able to use the buttons.
- Verify that the buttons to open and close lift door should be present.
- Verify that the buttons should be self-explanatory.
- Verify that the fan is present in lift or not.
- Verify that the controls are guided by a voice message.
- Verify that the performance of lift, the time taken to go a particular floor.
- Verify that the light is present in lift or not.

- Verify that the emergency button and contact details should be present in lift.
- Verify the behaviour of lift in case of power failure, lift should not free fall or should not stop in between two floors.
- Verify that lift moves to particular floor when the floors button is pressed.
- Verify that the door should not be opened if open is door button is pressed in between two floors.
- Verify that lift comes to floor is up/down button are pressed at a particular floor.
- Verify that backup mechanism should be present in case of power loss.
- Verify that the lift door closes or not if an object is placed in between the door, it should not close.
- Verify that the door should not open while the lift is in motion.
- Verify that the time for which the door remains open at floor.
- Verify that the behaviour by pressing stop button before reaching the specific floor.
- Verify that the behaviour by pressing open button while lift is moving.
- Verify that in case multiple floor number button is clicked, lift should stop at each floor.
- Verify that in case of capacity limit is reached users are prompted with warning alert.
- Verify that the behaviour of lift when there is smoke or fire inside the lift.
- Verify that the lift user is prompted with current floor and direction information the lift is moving towards.
- Verify that the behaviour of lift when the capacity of lift exceeds.

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

Ans. Verify that user is able to create a new group or not.

- Verify that user is able to add multiple contacts from contacts list.
- Verify that user is able to insert group name and select image for WhatsApp group DP.
- Verify that user is able to add and remove contacts from group.
- Verify that the user is able to delete a group.
- Verify that user can send and receive text messages in group.
- Verify that user can send and receive documents in group chat box.
- Verify that user can send and receive photos in group chat box.
- Verify that user can send and receive videos in group chat box.
- Verify that user can send and receive audio in group chat box.
- Verify that user can send and receive emotions icons in group chat box.
- Verify that user can send and receive contacts in group chat box.
- Verify that user can send and receive location in group chat box.
- Verify that user can send and receive GIF in group chat box.
- Verify that user can send and receive stickers in group chat box.
- Verify that user can delete text, video, audio, location, documents in group chat box.
- Verify that user can send recorded voice mail in group chat box.

- Verify that user is able to make multiple video call in group chat box.
- Verify that user is able to see the group contact information from group info in group chat box.
- Verify that the user is able to shared images, links and documents is group media in group chat box.
- Verify that the user is able to search specific chat history using search option ingroup chat box.
- Verify that the user is able to mute the group in group chat box.
- Verify that users have options like report, block, clear chat, export chat and add shortcut.

Q.57 Write a Scenario of WhatsApp payment.

Ans. Verify that users can register for WhatsApp payment.

- Verify that users can link their bank account to WhatsApp for transaction.
- Verifying the users phone number for WhatsApp payment.
- Verify that users can select a contact from their address book to send money to.
- Verify that users can review the transaction details before confirming.
- Verify that sending money from one user to another within the WhatsApp application.
- Verify that users can enter the amount they wish to send and add a note if necessary.
- Verify that users receive a notification when they have received money.
- Verify that users can view their transaction history.
- Verify that receiving money from another user.
- Verify that the transaction details are visible within the chat interface.
- Verify that security features such as PIN or biometric authentication before making a payment.
- Verify that adding a bank account for transaction.
- Verify that transaction history includes details such as date, time, amount, sender/receiver, and transaction status (completed, pending, failed etc.).
- Verify that users receive confirmation messages for each transaction, including details about the recipient and amount set.
- Verify that transaction is processed promptly and reliably without delays or errors.
- Verify that there are limits on the amount of money that can be sent or received per transaction and per day.
- Verify that users can add multiple bank accounts if supported.
- Verify that performance of the payment feature under various network conditions (3G, 4G, 5G, Wi-Fi).
- Verify that users can edit or remove bank accounts as needed.