QUALITY ASSURANCE (QA) INTERVIEW QUESTIONS AND ANSWERS

- **What is Software Testing?**

Software testing is a process of executing a program or application with the intent of finding the software bugs. Match the system as well as business requirements with the software or not?

- **What are Phases of STLC (Software Testing Life Cycle)?**
  - **Requirement analysis**: - Gather details about testing priorities and focus.
  - **Test Planning**: - Preparation of test plan/strategy document for various types of testing.
  - **Test case development**: - This phase involves creation, verification and rework of test cases & test scripts.
  - **Environment Setup**: - Test environment decides the software and hardware conditions under which a work product is tested.
  - **Test Execution**: - During this phase test team will carry out the testing based on the test plans and the test cases prepared. Bugs will be reported back to the development team for correction and retesting will be performed.
  - **Test Cycle Closure**: - Testing team will meet, discuss and analyze testing artifacts to identify strategies that have to be implemented in future, taking lessons from the current test cycle.

- **What are Phases of SDLC (Software Development Life Cycle)?**
  - Requirement gathering and analysis.
  - Design.
  - Implementation or coding.
  - Testing.
  - Deployment.
  - Maintenance.

- **What is Agile Methodology?**
  - In the Agile methodology, each project is broken up into several 'Iterations'.
  - AGILE is a methodology that promotes continuous iteration of development and testing throughout the software development life cycle of the project. Both development and testing activities are concurrent unlike the Waterfall model.
  - All Iterations should be of the same time duration (between 2 to 8 weeks).
  - At the end of each iteration, a working product should be delivered.
  - Any remaining features that cannot be delivered in the first iteration will be taken up in the next iteration or subsequent iterations, based on priority.

- **What is eXtreme Programming (XP) Agile Methodology?**

  eXtreme Programming (XP): - Extreme Programming technique is very helpful when there is constantly changing demands or requirements from the customers or when they are not sure about the functionality of the system. It advocates frequent "releases" of the product in short development cycles, which inherently improves the productivity of the system and also introduces a checkpoint where any customer requirements can be easily implemented. The XP develops software keeping customer in the target.

- **Levels in Software Testing?**

  1. Unit Testing.
  2. Integration Testing.
  3. System Testing.
  4. Regression testing.
  5. User Acceptance Testing.
  6. Pilot/Field Testing.
  7. Installation or Production Testing.

- **Methods in Software Testing?**

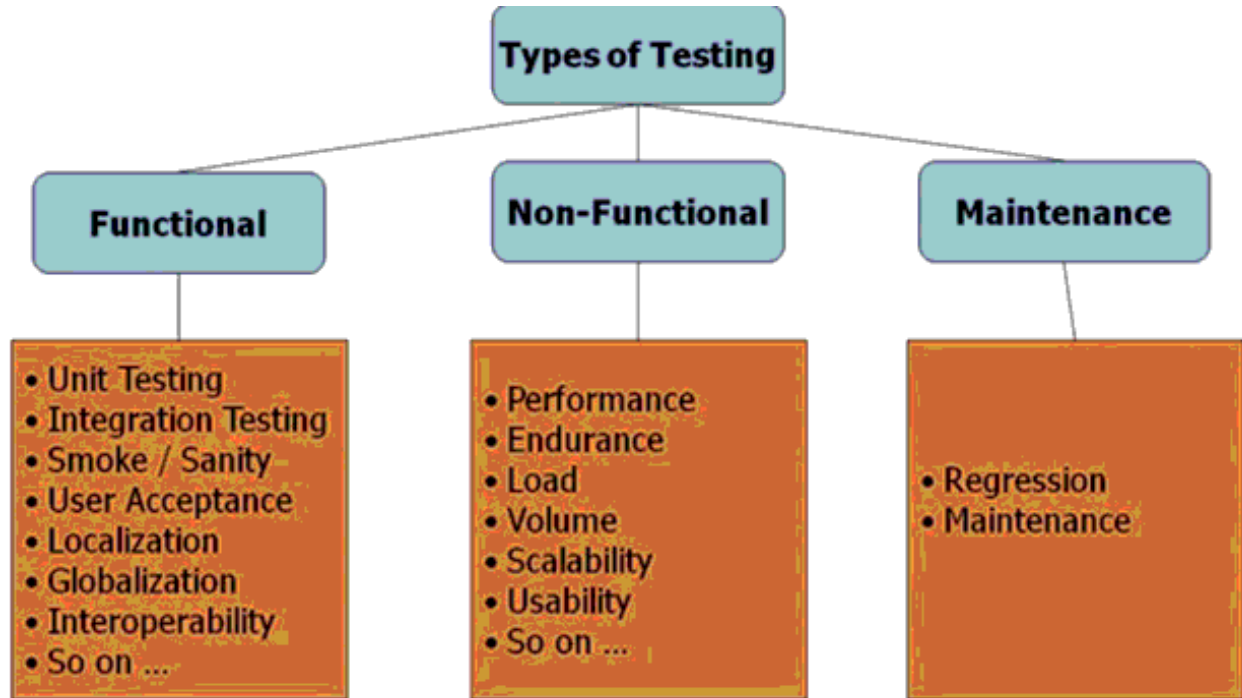| **Black-Box Testing**	**Grey-Box Testing**	**White-Box Testing**
The internal workings of an application need not be known.	The tester has limited knowledge of the internal workings of the application.	Tester has full knowledge of the internal workings of the application.

- **Types in Software Testing?**

Different types of tests (GUI testing, Functional testing, Regression testing, Smoke testing, load testing, stress testing, security testing, stress testing, ad-hoc testing) are carried out to complete system testing.

- **Web Application Testing.**

Functionality: -Test all links, Test Forms, Test cookies, Test Business workflow.
Usability: - Test the site navigation: - Menu, Buttons, Links. Test the content.
Interface: - Application, Web server, Database Server.
Database: - Test if any error are shown while executing queries.
Compatibility: - Browser compatibility test, operating system, Hardware.
Performance: - Site work under all loads.
Security: - Test unauthorized access to secure pages should not be permitted.

- **Types of Defects?**

Functionality Error, **Communication Error**, **Missing command Error, Calculation Error, Component Error, Control Error, Logical Error, Program Error, Message Error, Runtime Error, Typographical Error, Validation Error.**

- **Performance Testing, Load Testing, Stress Testing, Volume Testing?**

Volume Testing = Large amounts of data.
Performance testing= testing that is performed, to determine how fast some aspect of a system performs under a particular workload.
Load Testing = Large amount of users.
Stress Testing = Too many users, too much data, too little time and too little room.

- **What is Smoke & Sanity testing?**
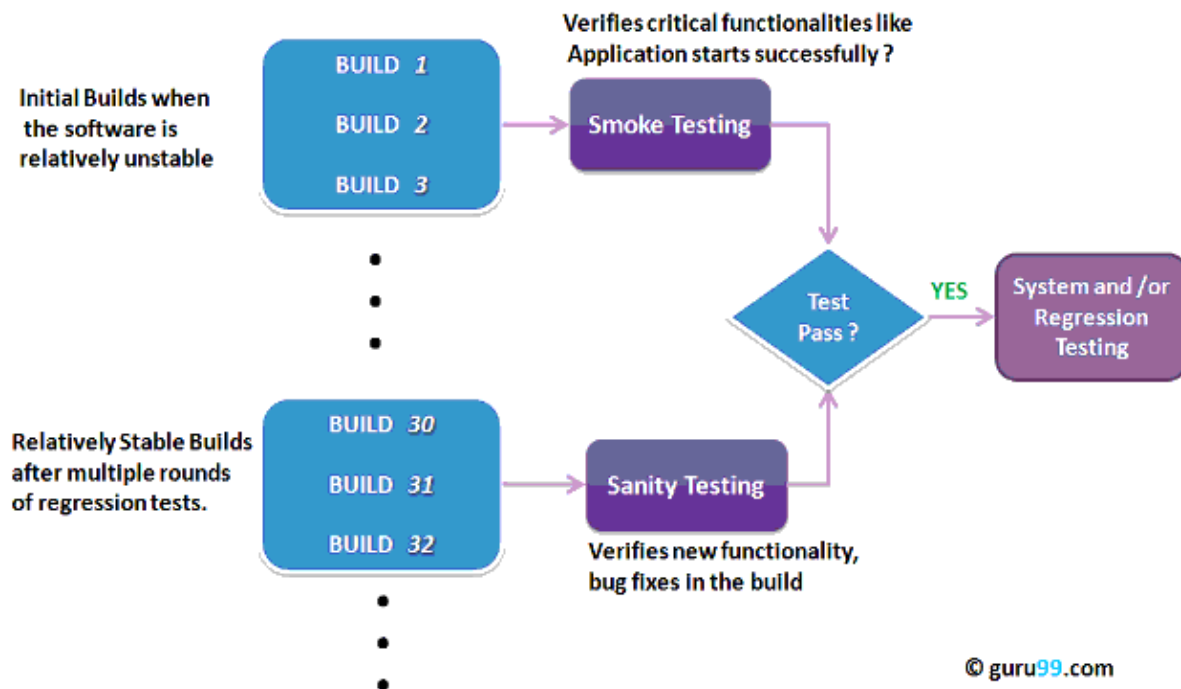
Example of smoke testing: -Verify that the application launches successfully, check that the GUI is responsive ... etc.

Example of Sanity testing: - Is the surface level testing where QA engineer verifies that all the menus, functions, commands available in the product and project are working fine.

- **Difference between Defect Priority and Defect Severity?**

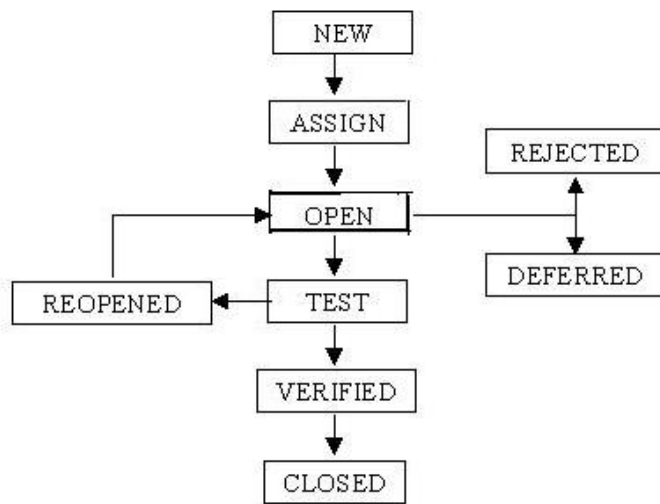| Defect Priority	Defect Severity
Priority is associated with scheduling.	Severity is associated with functionality or standards.
Priority indicates how soon the bug should be fixed.	Severity indicates the seriousness of the defect on the product functionality.

- **High severity low priority**: Logo of the company.
- **High severity high priority**: Submit button of login page not working or page not displaying.
- **Low severity high priority**: Cosmetic error or spelling mistake on login page. Its a small bug but has high priority as it's in the main login page.
- **Low severity low priority**: Spelling mistakes in text of home screen, Title name etc.

- **What is a Bug?**

When actual result deviates from the expected result while testing a software application or product then it results into a defect.
E.g.:- Application crash on clicking the SAVE button while creating a new user, hence unable to create a new user in the application.

- **Bug/Defect Life Cycle?**

```
        ┌─────────┐
        │   NEW   │
        └─────────┘
             │
             ▼
        ┌─────────┐
        │ ASSIGN  │
        └─────────┘      ┌──────────────┐
             │           │   REJECTED   │
             ▼           └──────────────┘
        ┌─────────┐            ▲
   ┌───▶│  OPEN   │────────────│
   │    └─────────┘            ▼
   │         │           ┌──────────────┐
   │         ▼           │   DEFERRED   │
┌──────────┐ ┌─────────┐ └──────────────┘
│ REOPENED │◀│  TEST   │
└──────────┘ └─────────┘
                  │
                  ▼
             ┌─────────┐
             │VERIFIED │
             └─────────┘
                  │
                  ▼
             ┌─────────┐
             │ CLOSED  │
             └─────────┘
```

- Tester finds the defect.
- Status assigned to defect- New.
- Defect is forwarded to Project Manager for analyze.
- Project Manager decides whether defect is valid or not.
- Here the defect is not valid- status given **Rejected**: - If the developer feels that the bug is not genuine, he rejects the bug. Then the state of the bug is changed to "rejected".
- Manager verifies whether a similar defect was raised earlier. If yes defect is assigned a status **Duplicate**:- If the bug is repeated twice or the two bugs mention the same concept of the bug, then one bug status is changed to "duplicate".
- **Deferred**: The bug, changed to deferred state means the bug is expected to be fixed in next releases. The reasons for changing the bug to this state have many factors. Some of them are priority of the bug may be low, lack of time for the release or the bug may not have major effect on the software.
- If no the defect is assigned to the developer who starts fixing the code. During this stage, the defect is assigned a status in- progress. Once the code is fixed. Defect is assigned a status fixed.
- Next the tester will re-test the code. In case, the test case passes the defect is closed. If the test cases fails again, the defect is re-opened and assigned to the developer.
- Consider a situation where during the 1st release of Flight Reservation a defect was found in Fax order that was fixed and assigned a status closed. During the second upgrade release the same defect again re-surfaced. In such cases, a closed defect will be re-opened.

- **Difference between defect, error, bug, failure and fault?**

A mistake in coding is called **error**.
Error found by tester is called **defect**.
Defect accepted by development team then it is called **bug**.
Build does not meet the requirements then it is **failure**.

- **Difference between Regression and Retesting?**

Retesting means testing the functionality or bug again to ensure the code is fixed. If it is not fixed, defect needs to be re-opened.

Regression testing means testing your software application when it undergoes a code change to ensure that the new code has not affected other parts of the software before its packet release. **Regression** ensures the original fault has been removed; **Regression** testing looks for unexpected side effects.

- **What is Requirement Traceability Matrix (RTM)?**

  - In testing this is the main part, this matrix will help to crosscheck whether the test cases as covered all the requirement specifications.
  - The main purpose of Requirement Traceability Matrix is to see that all test cases are covered so that no functionality should miss while testing.
  - Requirement Traceability Matrix. Requirement Traceability Matrix or RTM captures all requirements proposed by the client or development team and their traceability in a single document delivered at the conclusion of the life-cycle. In other words, it is a document that maps and traces user requirement with test cases.
  - Traceability Matrix means it provide mapping between user requirements ansd the test cases. Main Advantage is if we missed any test cases for the requirements at that time we can find out easy for what test cases missed for the what functionality by preparing the tracebility matrix.

- **What is Bug Leakage?**

A defect which exists during testing yet unfound by the tester which is eventually found by the QA manager/end-user is also called bug leakage.

- **What is Ad Hoc Testing?**

  - A testing phase where the tester tries to 'break' the system by randomly trying the system's functionality is called Ad Hoc testing. This can include negative testing also.
  - Ad-hoc testing can be achieved with the testing technique called Error Guessing.
  - Commonly used in software development, ad hoc testing is performed without a plan of action and any actions taken are not typically documented. Testers may not have detailed knowledge of product requirements. Ad hoc testing is also referred to as random testing and monkey testing.
  - **When to Execute Ad-hoc Testing:-**Ad-hoc testing can be performed when there is limited time to do exhaustive testing and usually performed after the formal test execution. Ad-hoc testing will be effective only if the tester has in-depth understanding about the System under Test.

  Types of ad-hoc testing:-
  - **Buddy Testing**: - Two buddies mutually work on identifying defects in the same module.
  - **Pair testing**: - Two testers are assigned modules, share ideas and work on the same machines to find defects.
  - **Monkey Testing**: - Monkey testing is a software testing technique in which the testing is performed on the system under test randomly.

- **What is Functionality Testing?**

Functionality testing subtest of functional testing. During this test test engineers are validates correctness and completeness of every functionality or requirement in the build with respect to functionality coverage's like GUI coverage, input domain coverage, error handling coverage, output coverage, service coverage, backend coverage.

Functionality testing is performed to verify that a software application performs and functions correctly according to design specifications. During functionality testing we check the core application functions, text input, menu functions and installation and setup on localized machines, etc.

It's part of Functional testing. Giving the input and checking the output. Testing the application against specifications. Functionality testing can be defined as testing the application to check its functionality with respect to the software requirements described under SRS document.

Functional testing: It is a mandatory testing level in system testing. During this test engineers are concentrating on validation of customer requirements means functionalities meet in build or not. It consists four subtests.

- **Format of Standard Test Cases?**

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
TU01	Check Customer Login with valid Data	1. Go to site http://demo.guru99.com 2. Enter UserId 3. Enter Password 4. Click Submit	Userid = guru99 Password = pass99	User should Login into application	As Expected	Pass

- **What are the different Methodologies in Agile Development Model?**

There are currently seven different agile methodologies that I am aware of:Extreme Programming (XP).
- Scrum.
- Lean Software Development.
- Feature-Driven Development.
- Agile Unified Process.
- Crystal.
- Dynamic Systems Development Model (DSDM).
- **Which are static and which are dynamic techniques?**Equivalence Partitioning: - dynamic
- Use Case Testing: - dynamic
- Data Flow Analysis: - Data Flow Analysis
- Exploratory Testing: - dynamic
- Decision Testing: - dynamic
- Inspections: - Data Flow Analysis

- **What is negative and positive testing?**

Testing Technique used for Positive and Negative Testing:
- Boundary Value Analysis.
- Equivalence Partitioning.

A **negative** test is when you put in an invalid input and receives errors. While a **positive** testing, is when you put in a valid input and expect some action to be completed in accordance with the specification.

- **An input field takes the year of birth between 1900 and 2004 what are the boundary values for testing this field?**

1899,1900,2004,2005

- **When should testing be stopped?**

It depends on the risks for the system being tested. There are some criteria bases on which you can stop testing.
- Deadlines (Testing, Release).
- Test budget has been depleted.
- Bug rate fall below certain level.
- Test cases completed with certain percentage passed.
- Alpha or beta periods for testing ends.
- Coverage of code, functionality or requirements are met to a specified point.

- **What is black box testing? What are the different black box testing techniques?**

Black box testing is the software testing method which is used to test the software without knowing the internal structure of code or program. This testing is usually done to check the functionality of an application. The different black box testing techniques are
* Equivalence Partitioning.
* Boundary value analysis.
* Cause effect graphing.
* **What is white box testing and list the types of white box testing?**

White box testing technique involves selection of test cases based on an analysis of the internal structure (Code coverage, branches coverage, paths coverage, condition coverage etc.) Of a component or system. It is also known as Code-Based testing or Structural testing. Different types of white box testing are:
* Statement Coverage.
* Decision Coverage.

* **Boundary value testing?**

Test boundary conditions on, below and above the edges of input and output equivalence classes. For instance, let say a bank application where you can withdraw maximum Rs.20, 000 and a minimum of Rs.100, so in boundary value testing we test only the exact boundaries, rather than hitting in the middle. That means we test above the maximum limit and below the minimum limit.

* **What is Fault Masking?**

Error condition hiding another error condition.

* **What is Equivalence partitioning testing?**

Equivalence partitioning testing is a software testing technique which divides the application input test data into each partition at least once of equivalent data from which test cases can be derived. By this testing method it reduces the time required for software testing.

* **What is verification and validation?**

Verifying process includes checking documents, design, code and program	It is a dynamic mechanism of testing and validating the actual product
It does not involve executing the code	It always involves executing the code
Verification uses methods like reviews, walkthroughs, inspections and desk- checking etc.	It uses methods like black box testing ,white box testing and non-functional testing
Whether the software conforms to specification is checked	It checks whether software meets the requirements and expectations of customer
It finds bugs early in the development cycle	It can find bugs that the verification process cannot catch
Target is application and software architecture, specification, complete design, high level and data base design etc.	Target is actual product
QA team does verification and make sure that the software is as per the requirement in the SRS document.	With the involvement of testing team validation is executed on software code.
It comes before validation	It comes after verification

* **What is the difference between UAT (User Acceptance Testing) and System testing?**

System testing: - System testing is finding defects when the system under goes testing as a whole, it is also known as end to end testing. In such type of testing, the application undergoes from beginning till the end.

User Acceptance Testing: - User Acceptance Testing (UAT) involves running a product through a series of specific tests which determines whether the product will meet the needs of its users.

- **Explain what is testing type and what are the commonly used testing type?**
- Unit Testing: Test the smallest code of an application.

- API Testing: Testing API created for the application.

- Integration Testing: Individual software modules are combined and tested.

- System Testing: Complete testing of system.

- Install/Uninstall Testing: Testing done from the point of client/customer view.

- Agile Testing: Testing through Agile technique.

- **In manual testing what are stubs and drivers?**

Why We Use Stubs And Drivers?

Stubs are dummy modules that are always distinguish as "**called programs**", or you can say that is handle in **top down integration testing** , it used when **sub programs** are under construction.
Stubs are considered as the dummy modules that always simulate the **low level modules**.

Drivers are also considered as the form of dummy modules which are always distinguished as "**calling programs**", that is handled in **bottom up integration testing**, it is only used when **main programs** are under construction.
Drivers are considered as the dummy modules that always simulate the **high level modules**.

Example of Stubs and Drivers is given below:-
For Example we have 3 modules login, home, and user module. Login module is ready and need to test it, but we call functions from home and user (which is not ready). To test at a selective module we write a short dummy piece of a code which simulates home and user, which will return values for Login, this piece of dummy code is always called Stubs and it is used in a top down integration. Considering the same Example above: If we have Home and User modules get ready and Login module is not ready, and we need to test Home and User modules Which return values from Login module, So to extract the values from Login module We write a Short Piece of Dummy code for login which returns value for home and user, So these pieces of code is always called Drivers and it is used in Bottom Up Integration.
Conclusion:-So it is fine from the above example that **Stubs** act "**called**" functions in **top down integration**. **Drivers** are "**calling**" Functions in bottom **up integration**.

- **Mention what are the different types of test coverage techniques?**
Different types of test coverage techniques include :-
Statement Coverage: It verifies that each line of source code has been executed and tested.
Decision Coverage: It ensures that every decision in the source code is executed and tested.
Path Coverage: It ensures that every possible route through a given part of code is executed and tested

- **Mention what is the difference between Pilot and Beta testing?**
Pilot testing – It is real world test done by the group of user before the final deployment to find as many defects as possible. The main purpose of pilot testing is to catch potential problems before they become costly mistakes.
Beta testing – It is the testing done by end users before the final release when the development and testing are essentially completed. The purpose is to find final problems and defects.

- **What is difference between Front End Testing and Back End testing?**

- **Front End Testing** is performed on the Graphical User Interface (GUI).whereas Back End Testing involves databases testing.
- **Front end** consist of web site look where user can interact whereas in case of **back end** it is the database which is required to store the data.
- When ender user enters data in GUI of the front end application, then this entered data is stored in the database. To save this data into the database we write SQL queries.

- **Mobile App Testing Important Test Scenarios.**
  - Check if the app works well with multiple carriers and multiple devices.
  - Usability of the features in a mobile screen.
  - Testing it in different mobile platforms – like Android and iOS.
  - Installations, uninstalling, launching the app with network and without network, testing functionality.
  - Network connections Wi-Fi, 2G, etc.
  - Logs at iOS iPhone configuration utility for Android Monitor.bat can be used for debugging.

- **Explain stress testing, load testing and volume testing?**
  - **Stress Testing**: When the load placed on the system is raised or accelerated beyond the normal range then it is known as Stress Testing.
  - **Load Testing**: Testing an application under heavy but expected load is known as Load Testing. Here, the load refers to the large volume of users, messages, requests, data, etc.
  - **Volume Testing**: The process of checking the system, whether the system can handle the required amounts of data, user requests, etc. is known as Volume Testing.
- **What the difference is between build and release in software testing?**
  - A "**build**" is a developed application for the customers that is given by development team to the software testing team. A "**release**" is an official launch of the application for the customers.

- **Questions List :-**
  - Why we need to do software testing?
  - What is sdlc? Is there any difference between sdlc and stlc?
  - What is Test case? What are all components of the Test case?
  - What about defect lifecycle?
  - What is severity/priority?
  - Example of severity 1 and low priority defect, sev3 and high priority defect?
  - What all challenges faced during the testing of last project/application?
  - Example of test scenarios for any real world product/application (Table/Wrist watch/Pencil)?
  - Types of software testing?
  - What is Test management tools? Examples?
  - Difference between Retesting and Regression testing?
  - How you prioritize your activities when you are near to project deadline?
  - When you will stop the testing?
  - What is load testing?
  - What is stress testing?
  - What is difference between mobile app testing and web application testing?
  - How do u test a web application?
  - Which method u use for testing? Agile?
  - What is agile testing? What is scrum methodology?
  - Difference between Re-testing and regression testing?
  - Difference between smoke and sanity testing?

- **Define performance and stress testing.**

Performance Testing: Performance Testing is performed to evaluate application performance under some load and stress condition. It is generally measured in terms of response time for the user

activity. It is designed to test the whole performance of the system at high load and stress condition.

Stress testing: It involves imposing the database with heavy loads. Such as, large numbers of users access the data from the same table and that table contains large number of records.

- **What are the typical problems in web testing?**

  - Functionality problems
  - User Interface related problems
  - Performance related problems
  - Database related problems
  - OS compatibility problems
  - Browser compatibility problems
  - Security related problems
  - Load related problem
  - Navigation problem

- **Write the test scenarios for testing a web site?**

First we have to assume that **Graphical User Interface (GUI) objects** and **elements** of a website together is One Test Scenario. Then, we have to check all the **links** and **buttons**. Then we have to **check all forms are working properly or not**. Prepare **Test Scenarios** of the forms of a webpage. We can identify 4 different types of Test Scenarios of a form: -

- Check the form with **valid data in all the fields**.
- Check the form with **invalid data which violate the validations of fields in the form**.
- Check the form by **leaving some mandatory fields in the form**.
- Check the form with **existing record details**.

While testing a website, which are the different configurations which will have to be considered?These configurations may demand for change in strategy of the webpage. The most important factors that need consideration are following: -

Hardware platform: some user may use the Mac platform, some may use Linux, while others may use Microsoft platform.

Browsers: browser and their versions also change the **layout** of the web page. Along with the browser versions, the different Plug-Ins also has to be taken into consideration. The resolution of the monitor also with color depth and text size is some of the other configurations.

- **What is the difference between authentication and authorization in web testing?**

- **Authentication** is the process with which the system identifies the user whereas authorization is the process after the **authentication** process.

- The **authentication** is used to ensure that the user is indeed a user, who he claims to be whereas in **authorization** system will decide whether a particular task can be performed by the user.

- There are different types of authentications, which can be used like **password based authentication**, **device based authentication** whereas in authorization there are two types **read only, and read write both**.

- **Explain the different between HTTP and HTTPS?**

- Hypertext Transfer Protocol is a protocol for information to be passed back and forth between web servers and clients.

- **HTTPS** is refers to the combination of a normal HTTP interaction over an encrypted Secure Sockets Layer (SSL) or Transport Layer Security (TLS) transport mechanism.

- **HTTP** use port number 80 whereas **HTTPS** use port number 443.

- **HTTP** can support the client asking for a particular file to be sent only if it has been updated after a certain date and time whereas **Hypertext Transfer Protocol over Secure Socket Layer** is built into its browser that **encrypts and decrypts user page requests as well as the pages that are returned by the Web server.**

- **The differences between Static and Dynamic website are following:**
  - A **static website** contains Web pages with fixed content where as in **Dynamic web site** content of the web page change with respect to time.
  - **Static website** are easy to create and **don't require** any **database design** but in case of **dynamic website** it require good knowledge to **develop** the website with **programming** and **database knowledge**.
  - In **static website** user **cannot communicate with other** and same information will be displayed to each user where as in **dynamic website** user may **communicate with each other**.

- **What is Cross Site Scripting?**

Cross Site Scripting is a **thread** in the **dynamic website**. It is also known as **XSS**. Cross site scripting occurs when a **web application gathers malicious data from a user**. The data is collected in the **hyperlink** form which contains **malicious content within it**. It allows **malicious code to be inserted into the web page**. The web page can be a **simple HTML code or a client side script**. When the malicious code is inserted in page and clicked by some user, the malicious code becomes a part of the web request of the user. This request can also execute on the user's computer and steal information.

- **What type of security testing you performed?**

To perform the security testing tester **try to attack the system**. This is the best way to **determine the lope hole in the security area of the application**. Most of the systems use **encryption technique to store passwords**. In this we have to try to get access to the system by using different combinations of passwords. Another common example of security testing is to find if the system is **vulnerable to SQL injection attacks**. While performing the security testing, tester cannot do any changes in any of the following: -

- **Configuration** of the **application** or the **server**

- **Services** running on the **server**
- **Existing user** or **customer data** hosted by the application

• **What are steps you will perform for securing a new web server?**

Some of the important steps to be carried out for securing the web server are following: -
- **Minimize** rights.
- Update **permissions**.
- **Delete default data** and **scripts**.
- Make use of **software firewall**.
- Enable and make use of IIS logging.
- Regular **backup**.

• **What is usability testing in web testing?**

Usability testing perform with reference to the **end user**. In usability testing we find **how easily end user can access the application**. In terms of websites and software applications, usability is defined as the ease at which a **person with no programming knowledge** can **use the software to complete the desired task**.

Usability is comprised of following: -
- **Learnability** is how easy it is for a new user to **accomplish tasks** the **first time they visit your website**.
- **Memorability** is how easy it is for someone to come back to using your website after they haven't used it for a period of time.
- **Efficiency** is how quickly users can **complete tasks** on your site after they are familiar with its use.

• **What is difference between Web based testing and windows based testing?**

Web based testing is concerned with the following: -
- **Broken Links** in the **web pages**.
- **Performance of web like response time**.
- **Graphical User Interface**.
- **Text** on the **page**
Where as in Windows based testing we look for: -
- **Functionality.**
- **Integration.**

• **What are the common bugs in web testing?**

In Web based testing following bugs are very common:
- Issues in **navigation** of application.
- **Usability**.
- **Cosmetic** Issues and **GUI** Issues.
- **Functional** Issues.

- **Performance** issues - How time it takes to **display the page to the user**.
- **Load** - How much **load** on application can **handle** at any **point** in **time**.
- **Stress** - At how much **load** application will **crash**.
- **Flow** of data - **Information** which is entered by **user** is **stored** in **correct format**.
- If proper static information is not displayed along with text fields to enter data.
- **Links are broken**, default focus is not set in forms, and **tab key not working**, all **key board short cuts are not fully functional.**

- **What is the difference between desktop application testing and web testing?**

Desktop Testing - Desktop application testing is **standalone** testing it is **independent** of the other **application** which are **executing** on the **different machines**. In this **application** testing, tester need not worry about **number** of **user**.

Web testing - Web testing is related to **client server**. Web testing needs to have many testing's like **Usability**, **GUI**, **Load** Testing, **Performance** Testing.

- **What is field validation in web testing?**

Field validation is used to ensure that only **correct data is entered into the field**. We can select validation options to make sure that **only correct format data** can be entered into a field correctly. When validation options are selected, we can use the FileMaker Pro to displays a message when user enter data in incorrectly format. For example, you can set an option to require that users enter a value in a field. The field validations check the format of the data. To ensure this we perform the validation testing in the website. Like the email field must contain the data in email@domain.com format.

- **What is focus testing in website?**

Focus testing is used to **test that when we open a webpage the cursor automatically blink on the particular field**. Like in the case of Gmail login page. When we open the Gmail login page the **cursor automatically blinks on the username filed**. This is the Focus testing in website.

- **Explain some web testing tools.**

Some of the webs testing tools are discussed below:
- **JMeter (load and performance tester):** JMeter is a **Java** desktop application which is used to **create the load test environment and measure performance of the application during load test**. It can be used for **examine the performance of static and dynamic website**.
- **Selenium (Web app testing tool):** Selenium contains several application like **Selenium IDE**, **Selenium Remote Control** and **Selenium Grid** to **examine** and **evaluate** the **web application**.

- **What is website testing?**

Website testing is a type of software testing which deal with the testing of the website. Website testing is performing on website to check the **functionality**, **performance**, **usability, database**

related **issues** and **browser** related **issues**. To perform this testing we have so many **tools** like **JMeter**, **Selenium**, **and QTP** etc.

- **How is Web Application Testing different to Desktop Application Testing?**
Web Applications are typically **hosted on a server** which we can access via a **web browser**, whereas **desktop applications** are installed on the **client's machine**.
This setup opens a whole new testing challenges: **Performance and Security testing** become important as the **application is open to a wide audience**. **Good design and usability** are also important.
Other important factors that come to play are testing on **multiple browsers, multiple devices, redirection and responsiveness**.
Also, we should not forget about **JavaScript**, **CSS**, **Cookies**, **W3C standards, traffic monitoring, third party tags testing,** all of which are important in Web Application Testing.

- **What are the HTTP response code blocks and what do they mean?**
After a request is sent to a server, there are different possible response codes which can be returned by the server:
The blocks are:
  - 2xx for **Success**, the most common one is **200** which means "OK".
  - 3xx for **Redirection**, the most common ones are **301** and **303** which mean "**Permanent Redirect**" and "**Redirect** for **Undefined Reason**", respectively.
  - 4xx for **Application Error**, the most common ones are **403** and **404** which mean "**Forbidden**" and "**Not Found**", respectively.
  - 5xx for **Server Error**, the most common one is **500** which means "**Server Error**".

- **How would you Test a Service Oriented Architecture (SOA) Web Application?**
The testing of web applications that communicate with a web service can be broken down in two parts:
  - **Testing of the Web Service in isolation:** Each web service has one or more functions which can be tested by **sending appropriate requests** and **analyzing** the **response** and **verifying correct data** is **returned** in the **response**. We can use tools such as **SoapUI** to test a **Soap Service** or **Rest Client** to test a **RESTful web service**.
  - **Integration Testing of Web Service with the Front End:** The **integration testing** is also important as it can **highlight** issues with **data** in the **request** and **display** of the **response**.

- **Suppose you have a Login form which is connected to an Authentication Web Service. What tests would you perform at which layer?**
  - All the **input/output validation should be tested at the API layer calling the Authentication Web Service**. Tests such as **valid/invalid username/password combinations as well as verifying correct error messages**.
  - The **location** of the **display** of **error messages**, their **color** and **font** should be tested at **login web page**. Also **JavaScript** and **Cookies** tests if applicable needs to be tested at **front-end login page**.

- **What is the difference between client-server testing and web based testing and what are things that we need to test in such applications?**
  Projects are broadly divided into two types of:
  - **2 tier applications**
  - **3 tier applications**
CLIENT / SERVER TESTING: -
This type of testing usually **done** for **2 tier applications** (usually **developed** for **LAN**).
Here we will be having **front**-**end** and **backend**.
The application launched on **front**-**end** will be having **forms** and **reports** which will be **monitoring** and **manipulating data.**

E.g. applications developed in **VB**, **VC++, Core Java**, C, C++, D2K, **PowerBuilder** etc.,
The **backend** for these **applications** would be **MS Access**, **SQL Server**, **Oracle**, **Sybase**, **MySQL**,
Quad base.
The tests performed on these types of applications would be
– **User interface testing.**
– **Manual support testing.**
– **Functionality testing.**
– **Compatibility testing & configuration testing.**
– **Intersystem testing.**

WEB TESTING: -
This is done for **3 tier applications** (**developed** for **Internet** / **intranet** / **xtranet**)
Here we will be having **Browser**, **web server** and **DB server**.
The **applications accessible** in **browser** would be **developed** in **HTML**, **DHTML**, **XML**, **JavaScript**
etc. (We can monitor through these applications).
Applications for the **web server** would be **developed** in **Java**, **ASP**, **JSP**, **VBScript**, **JavaScript**,
Perl, **Cold Fusion**, **PHP** etc. (All the **manipulations** are done on the **web server** with the **help** of
these **programs developed**).
The DBserver would be having oracle, sql server, sybase, mysql etc. (All data is stored in the database
available on the DB server)
The tests performed on these types of applications would be: -
– **User interface testing.**
– **Functionality testing.**
– **Security testing.**
– **Browser compatibility testing.**
– **Load / stress testing.**
– **Interoperability testing/intersystem testing.**
– **Storage and data volume testing.**

A web-application is a three-tier application.

1. **browser** (**monitors data**) [**monitoring** is done using **html**, **dhtml**, **xml**, **JavaScript**].
2. **webserver** (**manipulates data**) [**manipulations are done using programming languages** or
scripts like **adv java**, **asp**, **jsp**, **vbscript**, **JavaScript**, **Perl**, **ColdFusion**, **php**].
3. **database server** (**stores data**) [**data storage** and **retrieval** is done using **databases** like
oracle, **sql server**, **Sybase**, **MySQL**].

• **WHAT IS THE SCOPE OF WEB TESTING?**
Web testing is the name given to **Software Testing** that **focuses** on **testing** the **web**-**based**
applications.
Before **going** to the **production environment**, **test team performs exhaustive testing** of the
web applications. This **help** to **uncover different issues** in the **application** like the **functional**
discrepancy, **web application security**, **web service testing**, **problems during integration**,
environmental issues, and its **ability** to **handle** the **user load**. These issues if remain **uncovered**
may get **exposed** to the **public**. Thus, **efforts** are made in this **stage** of **testing** to **find** out the
possible bugs.

• **WHAT ARE THE MOST COMMON ISSUES SEEN IN WEB TESTING?**
1. **Functional** Issues.
2. Problems that occur while **navigating** an application.
3. **Usability** issues like **broken links**, form **fields missing default focus, tab key not working**
and all **keyboard shortcuts** not **fully functional**.
4. **Cosmetic** and **GUI** issues.
5. **Performance issues** – How much **time** it **takes** to **display** the **page** to the **user**.
6. **Load** – How much **load** an **application** can **handle** at any **point** in **time**.
7. **Stress** – **Load** that **causes** an **application** to **crash**.
8. The **flow** of **data** – **Storing** of **information entered** by the **user** in the **correct format**.
9. **Proper static information** is not displayed along with **text fields** to **enter data**.

- **EXPLAIN THE DIFFERENCE BETWEEN HTTP AND HTTPS?**

1. HTTP stands for **Hypertext Transfer Protocol** which is a **set** of **rules** for **passing** the **information back** and **forth between clients** and **web servers**. **HTTPS refers** to a **combination** of a **normal HTTP interaction** over an **encrypted Secure Sockets Layer** (SSL) or **Transport Layer Security** (TLS) **transport mechanism**.

2. HTTP uses **port** number **80** whereas, **HTTPS** uses **port** number **443**.

3. HTTP supports mechanism of **sending** a **file** to the **client** asking for it only if any **update** is there. Whereas, **HTTPS encrypts** user page **requests** and **decrypts** the **page response** that **web server returns**.

- **LIST DOWN THE KEY WEB APPLICATION TESTING TECHNIQUES?**

1. <u>Functional Testing</u> – includes **link testing**, **form validation**, **search operations**, and **navigation testing**.

2. <u>Security Testing</u> – includes **authorization check** for **secure pages**, **verify access control** and **user making direct entry** to the **internal page** is **redirect** to the **login page, check** if the **session expires** after it **remains idle** for a **pre-defined time** and **testing virus attacks**.

3. <u>Database Testing</u> – includes **verifying data integrity** on **creating**, **updating** or **deleting data** in the **database**, **obtain** a **correct result** on **executing heavy** queries, **retrieve data from** the **database** and **represent** on the **web pages correctly**.

4. <u>Performance Testing</u> – Check response times of **application under different speeds** of **connections**, **verify** if **site handles many simultaneous user requests** at the **same time**, check **how** the **site handles large input data from users** and **check** how the **site pulls through** if a **crash occurs** due to **peak load**.

Next, there are some which **ensure** a **good user**-**experience**.

5. <u>Usability Testing</u> – includes **navigation testing** to **verify** that the **Menus**, **Links** or **buttons** on **web pages** move to **correct pages**, **content testing** to **identify** all the **spelling** and **grammatical errors present** on the page.

6. <u>Compatibility Testing</u> – includes **verification** of **OS compatibility**, **browser compatibility**, and **mobile browsing**.

7. <u>Interface Testing</u> – verifies that **communication** towards all the **three servers** – **Web**, **Application**, and **Database Server** is **working fine**. Check if any **request interrupts** in-**between** then how the **application** is **responding**. Handle any **error** from **web** or **database server** to the **application server** and **display** the **correct error message** to the **user**.

- **LIST THE MAIN DIFFERENCES BETWEEN CLIENT-SERVER, THE WEB AND DESKTOP APPLICATIONS.**

<u>DESKTOP APPLICATION:</u>

1. The **application runs** in **single memory** with **Front**-**end** and **Back**-**end** in **one place**.

2. It has a **single user only**.

<u>CLIENT/SERVER APPLICATION:</u>

1. The **application runs** on **two** or **more machines**.

2. It is **menu**-**driven**.

3. Works in **connected mode** (that means **connection exists until logout**).

4. It has a **limited number** of **users**.

5. it has **less number** of **network issues** as **compared** to the **web app**.

<u>WEB APPLICATION:</u>

1. The **application** runs on **two** or **more machines**.

2. It is **URL**-**driven**.

3. It uses a **web browser** as the **client interface**.

4. Works in **disconnected mode** (stateless).

5. It has an **unlimited number** of **users**.

6. It has many **issues** like **hardware compatibility**, **browser compatibility**, **version compatibility**, **security issues**, and **performance issues**.

- **CAN YOU TELL FEW SCENARIOS FOR TESTING A WEB APPLICATION?**

1- FUNCTIONALITY VERIFICATION.

A- Testing Links which includes.

i. Internal Links.

ii. External Links.

iii. Mail Links.

iv. Broken Links.

B- Form validation includes.

i. Field validation.

ii. Display error message for wrong input.

iii. Verification of optional and mandatory fields.

C- Database.

i. Testing the database integrity.

D- Cookies.

i. Testing will be done on the client system side, on the temporary Internet files.

2- PERFORMANCE VERIFICATION.

Performance testing can be **applied** to **understand** the **web site's scalability**, or to **benchmark** the **performance** in the **environment** of **third party products** such as **servers** and **middleware** for **potential purchase**.

A- Connection Speed.

Testing done on various **networks** like **Dial**-**up**, **ISDN**.

B- Load.

i. A **Huge number** of **users accessing** the **application** at the **same time**.

ii. Check for **peak loads** and how the **system behaves**.

iii. A **Large amount** of **data accessed** by a **user**.

C- Stress.

i. Apply continuous load on the **system**.

ii. Verify the **performance** of **memory**, **CPU**, and **file handling**.

3- USABILITY.

The **characteristics** of a **system** are **measured**.

i. Ease of **use**.

ii. Navigation.

iii. Content completeness and **correctness**.

iv. General appearance.

4- SERVER SIDE INTERFACE.

To **verify** that **communication** is **proper** on this **interface**. **Test** the **compatibility** of the **server** with **software**, **hardware**, **network**, and the **database**.

5- CLIENT SIDE COMPATIBILITY.

Testing is **done** on **different platforms** using **various browsers**.

6- SECURITY.

Security testing involves the following.

i. Network Scanning.

ii. Vulnerability Scanning.

iii. Password Cracking.

iv. Log Review.

v. Integrity Checkers.

vi. Virus Detection.

- **WHAT ARE THE DIFFERENT CONFIGURATIONS WHICH TESTER SHOULD CONSIDER WHILE TESTING A WEB APPLICATION?**

1. **Hardware** **platform**: **different users** may use different **platforms** like **Mac**, **Linux**, and **Microsoft**.

2. **Browser:** The **layout** of the **web page changes** on the **different browser**. Also, **browser version** and **plug**-**ins** are other important factors. The **resolution** of the **monitor** along with **color depth** and **text size** is some of the other **configurations**.

- **WHAT IS THE DIFFERENCE BETWEEN AUTHENTICATION AND AUTHORIZATION IN WEB TESTING?**

1. Authentication is the **process** which **empowers** the **system** to **identify** the **user**. Whereas, **authorization happens** after **authenticating** the **user**.

2. The **authentication ensures** that the **user** is **indeed** a **valid user**, **who** he **claims** to be. Whereas, in **authorization system** will **decide** whether a **user** is **entitled** to **perform** a **particular task**.

3. There are different ways of doing **authentication** like **password**-**based** and **device**-**based**. However, **authorization** is of **two types read**-**only** and **read**-**write both**.

- **WHAT IS THE DIFFERENCE BETWEEN THE STATIC AND DYNAMIC WEBSITE?**

1. A Static website contains web pages with fixed content whereas, in Dynamic website content of the web page change with respect to time.

2. It's not difficult to build a static website as you don't require any database design. But developing a dynamic website requires good programming resources and database knowledge.

3. A static website doesn't support user communication as it displays same information to each one of them. Whereas, in the case of dynamic websites, users may communicate with each other.

- **EXPLAIN CROSS SITE SCRIPTING?**

Ans. Cross Site Scripting is a thread in the dynamic website. It is also known as XSS. It occurs when a web application gathers malicious data from a user.

Data collection happens in a hyperlinked form which contains malicious content within it. It allows insertion of malicious code into the web page. The web page can be a simple HTML code or a client side script.

Suppose a user, accidently clicks on a web page containing the malicious code then that code becomes part of the web-request of the user. This request can also execute on the user's computer and steal confidential information.

- **LIST DOWN THE ACTIONS NECESSARY FOR SECURING A NEW WEB SERVER?**

Ans. Following are some of the important steps that a tester should check for securing a web server.

1. Limit the user rights as per their roles.

2. Update user permissions for resources required.

3. Clean default data and scripts stashed on the server.

4. Use a Software firewall on the server.

5. Enabling and making use of IIS logging.

6. Taking Regular backups.

- **WHAT IS FIELD VALIDATION IN WEB TESTING?**

Ans. Field validation is done to ensure that user enters only correct data into the fields present on any web page. We can select a variety of validation option that depends on the type of data user may enter into the field. We can also ask to display an error message if the user enters an incorrect value. For example, you can set an option to make it mandatory for the user to enter a value in the field else, an error message is displayed. It performs data validation like email field must contain the data in*email@domain.com* format.

- **WHAT ARE DIFFERENT TOOLS THAT YOU USE FOR WEB TESTING?**

Ans. Below is the list of most popular web testing tools available nowadays. And we are using them frequently to run web tests.

1. JMeter (mainly load and performance testing tool) – It is a Java desktop application which is useful to create the load test environment. It measures the performance of the application during the load

test. And it is capable of testing the performance of both the static and dynamic website.

2. Selenium (Web app testing tool) – It is a framework for automated testing of web applications. It's not just a single tool but a suite of software each catering to different testing needs. Its components include Selenium IDE, Selenium Remote Control, Selenium Webdriver and Selenium Grid to examine and evaluate the web application. Nowadays, Selenium RC and Webdriver have merged into Selenium

- **DO YOU KNOW ABOUT THE TOOLS USED FOR ANALYZING HTTP TRAFFIC?**

Ans. Yes, we often need to track the HTTP requests that flows from the browser to the downstream web server. Also, by probing the network traffic, we can get to the details of each request/response and cross verify.

Next, many tools are available in the market for monitoring the Web traffic. I can name some of them which are open source and frequently used.

1. Wireshark – It's to check all data that passes through the network.
2. Fiddler – It is more useful for analyzing the HTTP/s only traffic.
3. Live HTTP Headers – It's a Firefox add-on which is helpful for previewing the HTTP headers.
4. Firebug – It's the most popular Firefox plugin which can monitor AJAX transactions. Though, it's essential feature is to fetch the Web element locators.
5. Browser Mob Proxy – It adds additional functionality into Selenium Webdriver for running automated tests.

- **WHAT ARE THE TESTS THAT YOU'LL RUN FOR TESTING A LOGIN FORM WHICH USES A WEB SERVICE FOR AUTHENTICATION?**

Ans. For testing such a user case, we need to verify both the Web service and the Login test form. So we can design the test cases in the following manner.

WEB SERVICE TESTING.
1. First of all, we'll cover the testing of Web service API for input/output validation.
2. We'll execute cases including valid/invalid username/password combinations as well as verify the correct error messages.

LOGIN UI TESTING.
1. Consequently, there will be tests for login web page to check the location of the display of error messages, their color, and font.
2. Also, if the login page uses any Javascript or Cookies, then we'll add cases to test the applicable functionality.

- **Describe yourself as QA Engineer?**

I started out my career as a QA engineer in the year _____. Since then I have been working on a variety of platforms and operating systems including Windows 7, Win 2K8, Win 2012 and different flavors of Linux such as Ubuntu, RHEL, Suse etc. During my stint as a test engineer, I have conducted validation of different kind of applications such as Java, Visual basics, C, C++, .NET etc. I've hands-on experience in testing client-server applications, web-based applications and many other programming languages.

Being a QA engineer, I do have experience in preparing test plans, writing test Cases. I use to attend several meetings with project managers, business analysts and sometimes with clients.

While thinking of different kinds of testing I have explored I should mention is Smoke Testing, Integration Testing, Regression Testing, Black box or UAT Testing, I have given all of them a shot. Writing a defect is also one of the important areas where I put special stress on. I would always assess, reassess, and test them thoroughly before passing it on.

- **How do you define QA, QC and Software Testing?**

2.1- Quality Assurance (**QA**) QA refers to the planned and systematic way of monitoring the quality of process which is followed to produce a quality product. QA tracks the outcomes and adjusts the process to meet the expectation.
2.2- Quality Control (**QC**) Concern with the quality of the product. QC finds the defects and suggests improvements. The process set by QA is implemented by QC. The QC is the responsibility of the tester.
2.3- Software Testing is the process of ensuring that product which is developed by the developer meets the user requirement. The motive to perform testing is to find the bugs and make sure that they get fixed.

- **How do you define a good test case?**

A **good test case ensures** that **all positive scenarios** and **negative scenarios** are **covered**. Good test **cases** do only **one thing** at a **time**. They **don't overlap** and are **less complicated**. They must be 'atomic'.

Characteristics of a good test case.

3.1- __Title__ A clear and one liner title to show the intent the test case.
3.2- __Purpose__ A brief explanation of the reason the test case was developed.
3.3-__Description__ A representation in words of the nature and characteristics of the test case.
3.4-__Test objects__ An unambiguous feature or module is being test.
3.5-__Preconditions__ Conditions that must be met before this test is executed.

- **How do you perceive a test plan? What does it usually consist of?**

A test plan is a test life cycle document which analyses the resource, scope, approach and schedule of several testing activities. It will help you to find items that need to be tested, its features that need further testing, the risks that come with some and the solutions as well.

It consists of history, contents, introduction, scope, overview and approach. The risks and assumptions are not left out either.

- **Assume you have a test plan with over 1000 test cases. How do you make sure what should be automated and what should still be done manually?**

In this scenario, we will focus on test case priority and the feasibility of automation for the test case in question.

5.1- Which scenarios are tedious and take a lot of time doing manually?
5.2- Which scenarios have been missed in the past?
5.3- Where did we see device differences due to fragmentation?
5.4- Which parts of our application are prone to regression?
5.5- Which test cases are complicated and would take a lot of time to establish an automated test case for?
5.6- Which parts of the app will likely change over the next few weeks? (If certain parts of the app are about to be changed, we recommend not to start with automated testing for these cases.)
5.7- Which test cases are best done manually as part of "explorative" testing and testing the user experience?

- **How do you determine which devices and OS versions we should test on?**

Good candidates will point to app analytics as the best measure, looking for the most used devices for their particular app. Another good answer would be to check the app reviews where people might have complained about specific issues happening on their devices.

- **Define a Test Case and a Use Case? What do they consist of?**

A test case is again a document which gives you a step by step detailed idea on how you can test an application. It usually comprises of results (pass or fail), remarks, steps, outputs and description.
A use case on the other is a document of another kind. It helps you understand the actions of the user and the response of the system found in a particular functionality. It comprises of the cover page, revision, contents, exceptions, and pre-and post conditions.

- **What is Test ware?**

Test ware is the subset of software, which helps in performing the testing of application. It is a term given to the combination of software application and utilities which is required for testing a software package.

- **What is a test strategy?**

A test strategy helps you understand the process of testing in every software development cycle. It has been made in such a way that all project managers and developers will be informed about some of the most important issues of testing.

Test strategy includes introduction, resource, scope and schedule for test activities, test tools, test priorities, test planning and the types of test that has to be performed.

- **What are the contents in test plans and test cases?**
  - **Testing objectives.**
  - **Testing scope.**
  - **Testing the frame.**
  - **The environment.**
  - **Reason for testing.**
  - **The criteria for entrance and exit.**
  - **Deliverables.**
  - **Risk factors.**

ADVANCE QA INTERVIEW QUESTIONS AND ANSWERS.

- **What is the strategy for Automation Test Plan?**

11.1- Strategy for Automation Test Plan.
11.2- Preparation of Automation Test Plan.
11.3- Recording the scenario.
11.4- Error handler incorporation.
11.5- Script enhancement by inserting check points and looping constructs.
11.6- Debugging the script and fixing the issues.
11.7- Rerunning the script.
11.8- Reporting the result.

- **What are test driver and test stub and why it is required?**

12.1- A stub is called from the software component to be tested, it is used in top down approach.
12.2- The driver calls a component to be tested, it is used in bottom up approach.
It is required when we need to test the interface between modules X and Y and we have developed only module X. So we cannot just test module X but if there is any dummy module we can use that dummy module to test module X.
Now module B cannot receive or send data from module A directly, so in this case we have to transmit data from one module to another module by some external features. This external feature is referred as Driver.

- **List out the roles of software Quality Assurance engineer?**

A software quality assurance engineer has to perform following tasks:
13.1- Writing source code.
13.2- Software design.
13.3- Control of source code.
13.4- Reviewing code.
13.5- Change management.
13.6- Configuration management.
13.7- Integration of software.
13.8- Program testing.
13.9- Release management process.

- **To what extent should developers do their own testing or do you believe testing is the responsibility of the QA team?**

The answer to this question is really depending on the business environment you are working in. in today's emerging test scenario it is also the developer's responsibility to perform at least some of his own code testing. Though it is not expected that he will have the capacity or that his focus should be to run through large test plans or test on a large stack of devices. However, without the responsibility to review and test his code, a sense of ownership will not develop.

We believe that results will improve more if all parties have access to test cases and are able to run and access them regularly to verify if the latest changes brought any regression.

- **What's your experience using Continuous Integration as part of the development process?**

If this is applicable to your company, it is a great thing to hear that a candidate has worked with Jenkins or Bamboo CI. If he has set up these systems and can give recommendations to you on what worked and did not work in his previous jobs, the candidate has earned himself not only bonus points, but a merit badge or two.

- **How do you define the bug life cycle?**

Bug life cycle basically comprises of numerous statuses of an error during its life cycle. A few examples are open, deferred, solved, reopened, fixed, solved and closed. You may also speak about this process and the way in which you monitor and determine the status with the help of several points.

- **Do you know about bug leakage and bug release?**

Bug release is when software or an application is handed over to the testing team knowing that the defect is present in a release. During this the priority and severity of bug is low, as bug can be removed before the final handover.
Bug leakage is something, when the bug is discovered by the end users or customer, and missed by the testing team to detect, while testing the software.

- **Tell us about the best bug of your test career?**

Well since there are so many quality bugs I've discovered in my testing career that I can't really remember the best one I found. What always surprises me is that you find so many different kinds of bugs so quick. It showed me that having multiple competences on the team is a great assets while testing. The latest bug hunt I did was conducted on a product application which was already on the market for some time. Still we found 21 bugs in 7 minutes! And yes even a crash! That is what amazes me.

- **What is your view on acceptance testing, when it is done and who does it?**

Acceptance Testing is a software testing checkpoint where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery. Formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system.

1- When is it performed?
Acceptance Testing is carried out after System Testing and before making the system available for actual use.

2- Who performs it?
Internal Acceptance Testing (Aka Alpha Testing) is done by members of the organization that has produced the software but who are not directly involved in the project (Development or Testing). Commonly, it is the members of Product Management, Pre-Sales and/or Tech Support.
External Acceptance Testing is performed by the product consumers who are not employees of the organization that developed the software. They can be some technical people from the client-side or the actual end users.

- **What is your experience in dealing with your team members, how do you plan it?**

When you work for an organization be it medium or large, it is almost likely that you won't be the only one in the team. And there are times when you find it very difficult and frustrating while dealing with the team members. There could be arguments, differences and misunderstandings and some will also try to ignore the others. But my purpose always is to look beyond all of this. I perceive it like we are a team and we should work together to reach a common goal. I've learnt to be friendly with my team mates and some times invite them over for coffee. As a human, it is very important to share feelings and have important discussions and that is exactly what I intend to do. This is something that not only me but everyone else in a working environment should apply.

- **WHAT IS REQUIREMENT TRACEABILITY MATRIX?**

Ans# The Requirements Traceability Matrix (RTM) is a tool that maintains a mapping of the test scenarios with the requirements. That's how it ensures that the Test Plan covers all the requirements and refers to the latest version of requirements.

102. EXPLAIN THE DIFFERENCE BETWEEN PILOT AND BETA TESTING?

Ans# Read the following points to know the difference between Pilot and Beta testing.
1. We do the beta test when the product is about to release to the customer whereas pilot testing takes place in the earlier phase of the development cycle.
2. In the beta test, testing application is given to few users to make sure that application meet the customer requirements and does not contain any showstopper bug. Whereas, in the pilot test, few members of the testing team work at the Customer site to set up the product. They give their feedback also to improve the quality of the end product.

103. DESCRIBE HOW TO PERFORM RISK ANALYSIS DURING SOFTWARE TESTING?

Ans# Risk analysis is the process of identifying the hidden issues that may derail the successful delivery of the application. It also prioritizes the sequence of resolving the identified risks for testing purpose.

Following are some of the risks that are of concern to the QA.
1**.** New Hardware.
2. New Technology.
3. New Automation Tool.
4. The sequence of code delivery.
5. Availability of test resources for the application.
We prioritize them into three categories which are as follows.
1. High magnitude: Impact of the bug on the other functionality of the application.
2. Medium: it is tolerable in the application but not desirable.
3. Low: it is tolerable. This type of risk has no impact on the company business.

104. DO YOU KNOW ABOUT SILK TEST? EXPLAIN IN YOUR WORDS.

Ans# Here are some facts about the Silk tool.
1. It's a tool developed for performing the regression and functionality testing of the application.
2. It benefits when we are testing Window based, Java, the web, and the traditional client/server applications.
3. Silk Test help in preparing the test plan and managing them to provide the direct accessing of the database and validation of the field.

105. WHAT IS THE DIFFERENCE BETWEEN MASTER TEST PLAN AND TEST PLAN?

Ans# The difference between Master Plan and Test Plan can be described using following points.
1. Master Test Plan contains all the test scenarios and risks prone areas of the application. Whereas, Test Plan document contains test cases corresponding to test scenarios.
2. Master Test Plan captures each and every test to be run during the overall development of application whereas test plan describes the scope, approach, resources and schedule of performing the test.
3. MTP includes test scenarios to be executed in all the phases of testing that run during the complete life cycle of the application development. Whereas, a separate Test Plan exists for each phase of testing like Unit, Functional, and System which contains the test cases related to that type only.
4. Only for big projects, we need a Master Test Plan which requires execution in all phases of testing. However, preparing a basic Test Plan is enough for small projects.

106. HOW DO YOU HANDLE A NON-REPRODUCIBLE BUG?

Ans# Following bugs lie under the non-reproducible category.
1. Defects observed due to low memory issue.
2. Bugs raised due to address pointing to a memory location that does not exist.
3. The race condition is an error scenario which occurs when the timing of one event impacts another executing in a sequence.
A tester can take the following actions to handle the non-reproducible bugs.

1. Execute test steps that are close to the error description.
2. Evaluate the test environment.
3. Examine and evaluate test execution results.
4. Keeps the resources & time constraints under check.

107. HOW DO YOU PERFORM AUTOMATED TESTING IN YOUR ENVIRONMENT?

Ans# Automation Testing is a process of executing tests automatically. It reduces the human intervention to a great extent. We use different test automation tool like QTP, Selenium, and WinRunner. These tools help in speeding up the testing tasks.
Using the above tools we can create test scripts to verify the application automatically. After completing the test execution, these tools also generate the test reports.

108. WHAT FACTORS MAKE YOU CHOOSE AUTOMATED TESTING OVER MANUAL TESTING?

Ans# The choice of automated testing over manual testing depends on the following factors.
1. The frequency of execution of a test case.
2. The test case contains repetitive test steps.
3. Time Comparison (time for preparing and running automated script for the first time is much less than manual execution time).
4. Reusability of Automation Script.
5. No frequent change in the execution environment.
6. Availability of automated test reports for every execution.
7. Small releases like service packs which include a minor bug fix. In such cases, regression type of cases is sufficient for validation.

109. HOW DO YOU DEFINE TEST DRIVER AND TEST STUB? EXPLAIN USING AN EXAMPLE.

Ans# The test driver is a piece of code that calls a software component under test. It is useful in testing that follows the bottom-up approach.
Test stub is a dummy program that integrates with an application to complete its functionality. These are relevant for testing that uses the top-down approach.
Let's take an example.
1. Let's say there is a scenario to test the interface between modules A and B. We have developed only module-A. Then we can test module-A only if we have real module-B or a dummy module for it. In this case, we call module-B as the Test Stub.
2. Now module-B can't send or receive data directly from module-A. In such scenario, we've to move data from one module to another using some external features called Test Driver.

110. WHAT ARE THE ESSENTIAL QUALITIES OF AN EXPERIENCED QA OR TEST LEAD?

Ans# Every QA or Test Lead should have the following qualities.

- Thorough knowledge of Software testing processes.
- Ability to accelerate teamwork to increase productivity.
- Improve coordination between QA and Dev engineers.
- Provide ideas to refine the QA processes.
- Ability to conduct RCA meetings and draw conclusions.
- Excellent written and interpersonal communication skills.
- Quick learner and able to groom the team members.

Interview Questions from Youtube: -
- **What's Test Plan?**
    - Is a document that describes the objectives, scope, approach and focus of a software testing effort.
- **What's Test Case?**
    - A test case is a document that describes an input, action or event and an expected response to determine if a feature of an application working correctly.
- **What should be done after a Bug is found?**

- The bug needs to be communicated and assigned to developers that can fix it. After the problem is resolved, fixes should be re-tested and determinations made regarding requirements for regression testing to check that fixes didn't create problems elsewhere.
- **What is Configuration management?**
  - Configuration management covers the processes used to control, coordinate and track: code, requirements, designs, tools/compilers/libraries/patches, changes made to them, and who makes the changes.
- **How does a client/server environment affect testing?**
  - Client/Server applications can be quite complex due to the multiple dependencies among clients, Data communications, hardware and servers.
- **How can World Wide Web sites be tested?**
  - Web sites are essentially client/server applications – with web servers and 'browser' clients. Consideration should be given to the interaction between html pages, TCP/IP communications, internet connections, firewalls, applications that run-in web pages.
- **What is Extreme Programming and what's it got to do with testing?**
  - Extreme Programming(XP) is a software development approach for small teams on risk-prone projects with unstable requirements.
- **What is Verification?**
  - Verification typically involves reviews and meetings to evaluate documents, plans, code, requirements, issues lists, walkthroughs and inspection meetings.
- **What is Validation?**
  - Validation typically involves actual testing and takes place after verification is completed.
- **Which methodologies are used in your project?**
  - Agile development methodology.