1st Assignment

**1.What is test scenario:**

In software quality assurance (SQA), a test scenario is a specific set of test cases or test scripts that are designed to validate a particular aspect of a software application. Test scenarios are created based on the requirements and specifications of the software and are used to verify that the system behaves as expected under different conditions.

Test scenarios in SQA typically focus on a specific functionality or feature of the software and outline the steps, inputs, and expected results for testing that functionality.

**2.What is Test case:**

A test case is a detailed set of conditions or steps that are designed to test a specific aspect of a software application. Test cases are created based on requirements and specifications and outline the inputs, actions, and expected results for testing a particular functionality or feature of the software.

Each test case is typically written in a structured format and includes information such as the test case ID, description, preconditions, test steps, expected results, and actual results. Test cases are used by testers to verify that the software behaves as expected under different conditions and to identify any defects or issues that may be present.

A test case is a set of actions performed on a system to determine if it satisfies software requirements and functions correctly. The purpose of a test case is to determine if different features within a system are performing as expected and to confirm that the system satisfies all related standards, guidelines and customer requirements. The process of writing a test case can also help reveal errors or defects within the system.

Types of Testing :

**Non-Functional Testing:**

**1.Unit Testing:**

Unit testing is a quality assurance technique that involves testing the smallest functional units of code. Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually scrutinized for proper operation.

**2.Smoke Testing:**

Smoke testing is a software testing method that determines if a new software build is ready for the next testing phase. It's also known as build verification testing, confidence testing, sanity testing, or build acceptance test.

Smoke testing is performed in the early stages of development to identify minor issues that might delay the product's release. It ensures that all core functionalities of the program are working smoothly and cohesively

**3.Integration Testing:**

Integration testing is a software testing process that involves checking individual components of a software project to see if they work together as intended. It is the second level of the software testing process, following unit testing.

**4.Regression Testing:**

Regression Testing: Regression testing involves retesting the software after changes or updates to ensure that new code modifications have not introduced any defects or issues in existing functionality. Regression testing helps maintain the stability and reliability of the software over time.

Regression testing is a software testing technique that runs after code changes to ensure that the changes don't cause unintended breaks. It's a critical quality assurance practice in the software development lifecycle. Regression testing involves running functional and non-functional tests to ensure that a software application works as intended after any code changes, updates, revisions, improvements, or optimizations.

Regression testing is called regression testing because the verb regress means to return to a former state or condition, which in software, isn't considered a good thing

**5.Sanity Testing:**

Sanity testing is a basic test that verifies a system's basic functionality after major changes. It's a subset of regression testing and is performed before a full regression test.

**6.Cross browser Test:**

Cross-browser testing is a quality assurance (QA) process that verifies that web applications function as expected across different combinations of web browsers, operating systems, and devices. It's a type of non-functional testing that can be done manually or as part of a test automation strategy.

**7.Volume Test:**

Volume testing, also known as flood testing, is a type of software testing that evaluates a system's stability and response time. It involves transferring large amounts of data to test the data load capabilities of a product.

Volume testing is a non-functional performance test that analyzes the performance of applications that handle large amounts of data. For example, if a database is expected to grow in size, the database can be artificially increased to that size and the application's performance tested

Functional Testing:

**1.Performance Testing:**

Performance testing is a software quality assurance (QA) practice that evaluates how a system performs under a specific workload. It's a key part of software testing that ensures a system performs optimally under specified loads

**2.Load Testing:**

Load testing is a key part of software quality assurance (SQA). It is a process that helps determine how a system behaves under normal and high loads. Load testing helps to ensure that web applications and services perform optimally under heavy traffic

**3.Stress Testing:**

Stress testing is a type of software testing that evaluates how a system performs under extreme conditions. It's used to determine how well a system can handle heavy loads, network congestion, or other extreme conditions. Stress testing is performed while building software to understand its capacity, stability, and strength.

Stress testing can help identify potential errors, bottlenecks, or vulnerabilities that could affect the user experience. It can also help identify performance limitations, such as memory leaks, scalability issues, and concurrency problems

* **Black box Test:**

Black Box Testing: Black box testing focuses on testing the software's functionality without knowledge of its internal code or design. Testers design test cases based on input-output scenarios and user workflows to validate the software's behavior from a user's perspective

Black box testing is a software testing technique that evaluates software functionality from an end-user's perspective. It's a key method in software quality assurance.

In black box testing, the tester doesn't know the system's internal workings or code structure. Instead, they focus on the software's external behavior, such as the input and output it produces. The tester doesn't cover the inside details like code, server logic, or development method.

* **White Box test:**

White Box Testing: White box testing involves testing the internal structure, code, and logic of the software to ensure that it functions correctly at the code level. Testers have access to the source code and use this knowledge to design test cases targeting specific paths and conditions within the code.

White box testing is a software testing method that examines an application's internal structure, logic, and coding. It's also known as clear box testing, glass box testing, transparent box testing, and structural testing.

White box testing is different from black box testing, which tests an application's functionality. White box testing gives testers complete knowledge of the application being tested, including access to source code and design documents. This in-depth visibility allows white box testing to identify issues that are invisible to other types of testing.

* **Functional Test:**

Functional testing is a type of software testing that determines if an application's features work as per the software's requirements. It's a method used by software developers to perform quality assurance (QA).

Functional testing's goal is to ensure that the software functions correctly and as per the desired specifications. It does this by comparing each function to an organization's specifications to ensure that the software provides the output that an end user or business requires.