**Module 3 (Testing on Live Application)**

**1**)RDBMS (Relational Database Management System): RDBMS stands for Relational Database Management System. It is a software system that manages relational databases. It provides a structured approach to store, organize, and retrieve data using tables consisting of rows and columns. RDBMS ensures data integrity and supports various operations like insertion, deletion, and modification of data.

**2**)SQL (Structured Query Language): SQL stands for Structured Query Language. It is a programming language used for managing relational databases. SQL allows users to define, manipulate, and query data in the database. It provides commands for creating tables, inserting data, updating data, deleting data, and retrieving data from the database.

**3**)SQL Commands: Here are some common SQL commands:

* SELECT: Retrieves data from one or more tables.
* INSERT: Inserts data into a table.
* UPDATE: Modifies data in a table.
* DELETE: Deletes data from a table.
* CREATE: Creates a new table, view, or database.
* ALTER: Modifies the structure of a table or view.
* DROP: Deletes a table, view, or database.
* JOIN: Combines rows from two or more tables based on a related column.

**4**)Join: A join is used in SQL to combine rows from two or more tables based on a related column between them. It allows you to retrieve data from multiple tables in a single query. Joins are performed using the JOIN keyword and specify the columns that establish the relationship between the tables. Common types of joins include:

1. Inner Join: Returns only the rows that have matching values in both tables.
2. Left Join (or Left Outer Join): Returns all rows from the left table and matching rows from the right table.
3. Right Join (or Right Outer Join): Returns all rows from the right table and matching rows from the left table.
4. Full Join (or Full Outer Join): Returns all rows from both tables, including unmatched rows.
5. Cross Join (or Cartesian Join): Returns the Cartesian product of the two tables (all possible combinations).

**5**)Constraints in RDBMS: Constraints are rules defined on tables to maintain the integrity and consistency of the data. They impose restrictions on the data that can be inserted, updated, or deleted in a table. Common types of constraints include:

1. Primary Key: Ensures the uniqueness and integrity of a table's primary key column(s).
2. Foreign Key: Establishes a relationship between two tables by referencing the primary key of another table.
3. Unique: Ensures the uniqueness of values in one or more columns.
4. Not Null: Ensures that a column does not contain null values.
5. Check: Defines a condition that must be satisfied for data in a column.
6. Default: Specifies a default value for a column when no value is provided during insertion.

**6**)RDBMS vs. DBMS: RDBMS and DBMS both are database management systems, but there is a difference between them.

DBMS (Database Management System) is a software system that allows users to define, create, and manage databases. It provides basic functionalities like data storage, retrieval, and manipulation. DBMS may or may not follow the relational model.

RDBMS (Relational Database Management System) is a type of DBMS that follows the relational model. It organizes data into tables consisting of rows and columns and establishes relationships between tables using keys. RDBMS enforces the ACID properties (Atomicity, Consistency, Isolation, Durability) and provides support for SQL.

**7**)API Testing: API Testing refers to the process of testing application programming interfaces (APIs) to ensure their functionality, reliability, and performance. APIs enable communication between different software systems or components, allowing them to interact and exchange data

**8**)Types of API Testing:

1. Unit Testing: Tests individual API methods or functions to ensure their correctness and behavior.
2. Functional Testing: Verifies the functional requirements and behavior of the API.
3. Performance Testing: Evaluates the performance and responsiveness of the API under various load conditions.
4. Security Testing: Checks for vulnerabilities and ensures the API's security measures are effective.
5. Integration Testing: Tests the interaction between multiple APIs and verifies the data flow and functionality.
6. Load Testing: Determines how the API performs under expected and peak loads.
7. Stress Testing: Tests the API's stability and performance under extreme conditions.
8. Error Handling Testing: Tests the API's ability to handle and report errors gracefully.
9. Usability Testing: Evaluates the API's user-friendliness and ease of integration.

**9**)Responsive Testing: Responsive Testing refers to the process of testing a website or application to ensure it displays and functions correctly across different devices and screen sizes. The goal is to ensure that the user interface adapts and responds appropriately to different resolutions, orientations, and devices.

Tools for Responsive Testing:

1. Browser DevTools: Most modern web browsers like Chrome, Firefox, and Safari provide built-in developer tools that include device emulation and responsive testing features.
2. Responsive Design Mode: Browsers like Chrome and Firefox have a built-in responsive design mode that allows you to test and simulate different screen sizes.
3. CrossBrowserTesting: A cloud-based platform that provides real device testing for responsive designs across various browsers and devices.
4. BrowserStack: Another cloud-based platform that offers responsive testing on real devices and browsers.
5. LambdaTest: Provides cross-browser testing and responsive testing on cloud-based virtual machines.
6. ResponsiveTest.net: An online tool that allows you to test a website's responsiveness by entering its URL and selecting different device sizes.

**10**)Full form of .ipa and .apk: .ipa: The .ipa file extension stands for "iOS App Store Package." It is the file format used for distributing and installing applications on iOS devices like iPhones and iPads.

.apk: The .apk file extension stands for "Android Application Package." It is the file format used for distributing and installing applications on Android devices.

**11**)To enable Developer Options mode on an Android device:

1. Go to the "Settings" menu on your Android device.
2. Scroll down and find "About phone" or "About device" (the exact name may vary depending on the device).
3. In the "About phone" section, look for the "Build number" or "Build version" option.
4. Tap on the "Build number" repeatedly (usually 7 times) until you see a message saying "You are now a developer" or something similar.
5. Once the developer mode is enabled, you can go back to the main Settings menu, and you will find the "Developer options" or "Developer settings" listed there

**12**)Components used in Load Runner: Load Runner consists of several components:

* Virtual User Generator (VuGen): Used to create Vuser scripts by recording user actions or manually coding scripts.
* Controller: Used to design and execute performance tests, define scenarios, and manage Vusers.
* Load Generator: Executes Vuser scripts and generates load on the target system.
* Analysis: Provides in-depth analysis of test results and generates reports.

**13**) Setting the number of Vusers in Load Runner: In Load Runner, the number of Vusers can be set in the Controller component. You can define the number of Vusers per script and configure the load distribution across different load generators.

**14**) Correlation: Correlation is the process of capturing and replacing dynamic values in a script with unique parameters. During script replay, dynamic values such as session IDs or timestamps need to be captured and replaced with appropriate values to emulate realistic user behavior.

**15**) Process for developing a Vuser Script: The process for developing a Vuser script typically involves the following steps:

* Recording: Use VuGen to record user actions on the application under test.
* Enhancing the script: Modify the script to handle dynamic values, parameterize data, and add business logic.
* Script execution: Validate the script by executing it in VuGen and ensuring it works as expected.
* Debugging: Identify and fix any errors or issues in the script.
* Performance tuning: Enhance the script to simulate realistic user behavior and optimize performance.

**16**) Load Runner's interaction with the application: Load Runner interacts with the application under test by emulating user actions through Vuser scripts. These scripts simulate user behavior, such as browsing web pages, submitting forms, or making API calls. Load Runner captures and replays these actions, allowing performance testing of the application under different load conditions.

**17**) Number of Vusers required for load testing: The number of Vusers required for load testing depends on various factors such as the expected user load, the complexity of user transactions, the system under test, and the available resources. It is determined through load testing planning, which involves analyzing the application's requirements and performance goals.

**18**) Relationship between Response Time and Throughput: Response time refers to the time taken by the system to respond to a user request. Throughput, on the other hand, represents the number of transactions or requests processed by the system per unit of time. The relationship between response time and throughput can vary depending on the system and the workload. In general, as the load on the system increases, the response time tends to increase while the throughput may decrease or stabilize.

**19)** Difference between hits/second and requests/second: Hits/second and requests/second are two different metrics used to measure the load on a system:

* Hits/second: Refers to the number of HTTP requests made to the server per second, including both page requests and resource requests (images, CSS files, etc.). Each element on a web page may generate multiple hits.
* Requests/second: Refers to the number of unique user requests made to the server per second. It represents the number of individual interactions with the server, irrespective of the number of hits generated

**20**)Automation Testing: Automation Testing refers to the use of software tools and scripts to perform test cases and validate the functionality of a system or application. It involves automating repetitive and manual testing tasks, such as executing test scripts, comparing actual results with expected results, and generating test reports. Automation Testing aims to improve the efficiency, accuracy, and coverage of testing processes.

**21**)Browsers Supported by Selenium IDE: Selenium IDE (Integrated Development Environment) is primarily a browser automation tool that allows users to record and playback interactions with a web browser. As of my knowledge cutoff in September 2021, Selenium IDE supports the following browsers:

* + Mozilla Firefox (with Selenium IDE extension)
  + Google Chrome (with Selenium IDE extension)
  + Microsoft Edge (with Selenium IDE extension)

Please note that browser support may change over time, so it's recommended to check the official Selenium documentation for the most up-to-date information.

**22**)Benefits of Automation Testing: Some key benefits of Automation Testing include:

* + Faster and efficient execution of test cases compared to manual testing.
  + Increased test coverage by running a large number of tests in a shorter time.
  + Improved accuracy and reliability by eliminating human errors.
  + Reusability of test scripts, reducing effort and time in regression testing.
  + Cost reduction in the long run due to reduced manual effort and increased productivity.
  + Enhanced software quality and faster time-to-market.

**23**)Advantages of Selenium: Selenium is a popular open-source automation testing framework that offers several advantages:

* + Platform Compatibility: Selenium supports multiple platforms such as Windows, macOS, and Linux, making it versatile for cross-platform testing.
  + Browser Compatibility: Selenium supports major web browsers like Firefox, Chrome, Safari, Edge, and more, allowing tests to be executed across different browsers.
  + Language Support: Selenium supports multiple programming languages like Java, Python, C#, Ruby, and more, enabling testers to use their preferred language for scripting.
  + Extensibility: Selenium provides a rich set of APIs and supports integration with other tools and frameworks, allowing testers to extend its capabilities as per their needs.
  + Large Community and Resources: Selenium has a vast community of users and developers, which means there is extensive documentation, forums, and resources available for learning and troubleshooting.

**24**)Selenium vs. QTP (UFT): Testers may choose Selenium over QTP (now known as UFT, or Unified Functional Testing) for several reasons:

* + Cost: Selenium is open-source and free to use, while QTP/UFT is a commercial tool that requires a license, making Selenium more cost-effective.
  + Language Support: Selenium supports multiple programming languages, whereas QTP/UFT primarily uses VBScript, limiting language options for testers.
  + Flexibility and Customization: Selenium provides more flexibility and customization options due to its open-source nature, allowing testers to tailor their automation frameworks as needed.
  + Community Support: Selenium has a larger and more active community, resulting in extensive support, frequent updates, and a wider range of available resources.
  + Web-Centric Testing: Selenium is specifically designed for web application testing and has strong capabilities in this area, while QTP/UFT offers broader application testing capabilities but with a stronger focus on desktop applications.