**Submitted by**:- kunal joshi

**Module-4 Automation Core Testing (Load Runner Up and Selenium IDE)**

**Q1). Which components have you used in Load Runner?**

Virtual User Generator (VuGen): This component is used to create scripts that simulate user interactions with the application under test. It allows you to record user actions or manually create scripts using various programming languages, such as C, Java, or JavaScript.

Controller: The Controller component is responsible for orchestrating and controlling the load testing scenarios. It allows you to define and manage virtual users, assign scripts, configure load generators, set up test scenarios, and monitor test execution.

Load Generators: Load generators are machines or instances that simulate virtual users by generating the load on the application. Load Runner distributes the virtual users across multiple load generators to generate realistic user loads.

Analysis: The Analysis component helps you analyze the results of load tests. It provides various graphs, reports, and statistics to help you understand system behavior under different load conditions. You can compare different test runs, identify performance bottlenecks, and generate reports for further analysis and troubleshooting.

Monitoring: Load Runner can integrate with various monitoring tools to capture system-level metrics during the test execution. These metrics can include server performance data, database metrics, network statistics, and more. By monitoring the system under load, you can identify potential performance issues and correlate them with test results.

**Q2 How can you set the number of Vusers in Load Runner?**

You can set the number of Vusers in the controller section while creating your scenarios. Many other advanced options like ramp-up, ramp-down of Vusers are also available in the Controller section.

**Q3 What is Correlation?**

Correlation is a concept in automation testing that refers to the process of capturing and extracting dynamic values from server responses and using them in subsequent requests. In web applications, dynamic values are often used to maintain session state, track user interactions, or ensure data integrity.

**Q4 What is the process for developing a Vuser Script?**

Identify the User Actions: Understand the user actions or business processes that you want to simulate in your performance test. This could include actions like logging in, navigating through the application, submitting forms, and logging out.

Record the Script: Use Load Runner's Virtual User Generator (VuGen) component to record the user interactions. Start recording, perform the identified user actions in the application, and stop recording once you have completed the scenario.

Enhance the Script: After recording, you will need to enhance the script to make it more robust and realistic. This involves modifying the script to handle dynamic values, such as session IDs or timestamps, through correlation. You may also need to add error-handling mechanisms and logic to simulate user think time or pacing between actions.

Parameterize the Script: If you want to simulate different user scenarios or test data variations, you'll need to parameterize the script. Parameterization involves replacing static values with variables that can be dynamically changed during test execution. For example, you can parameterize login credentials, input data, or URLs.

Validate the Script: Validate the script to ensure it accurately simulates the desired user actions. Execute the script locally or in a limited environment to identify any issues, such as incorrect correlation or missing steps. Debug and modify the script as necessary to ensure its correctness.

Customize and Fine-Tune: Load Runner provides customization options to tailor the script according to your testing requirements. You can modify the script to add additional logging, capture specific metrics, simulate different load profiles, or configure runtime settings.

Validate Parameters and Performance Goals: Review the script to verify that the required performance goals, such as response times or throughput, are correctly set. Ensure that the script is configured to capture the necessary performance metrics for analysis and evaluation.

Iterate and Refine: Script development is an iterative process. As you run load tests, analyze results, and identify issues, you may need to refine and enhance the script accordingly. This could involve fixing correlation issues, adjusting parameterization, or optimizing script logic to improve performance.

**Q5 How Load Runner interacts with the application?**

Recording: Load Runner's Virtual User Generator (VuGen) component is used to record the user actions performed on the application. During recording, VuGen captures the network traffic between the client and the server, including HTTP/HTTPS requests and responses.

User Actions: The user actions performed on the application, such as clicking buttons, filling out forms, or navigating through pages, are recorded by VuGen. These actions are typically performed by a human tester using a web browser or a client application.

Protocol Analysis: Load Runner supports various protocols for different types of applications, such as HTTP/HTTPS, Web Services, Java, .NET, Citrix, SAP, Oracle, and more. During recording, Load Runner analyzes the underlying protocol used by the application to understand the communication between the client and server.

Script Generation: Based on the recorded user actions and protocol analysis, Load Runner generates a script that represents the simulated user interactions with the application. The script is typically written in a programming language such as C, Java, or JavaScript, depending on the chosen protocol and script type.

Script Replay: Load Runner's Controller component is responsible for replaying the generated script. The Controller distributes the generated load across multiple load generators, which simulate virtual users (Vusers) by replaying the script.

Request Generation: During the replay phase, the Controller generates requests based on the script and sends them to the application's server. These requests simulate the actions performed by the Vusers, such as submitting forms, making API calls, or interacting with the application's user interface.

Response Capturing: As the application's server responds to the requests, Load Runner captures the responses. This includes response headers, content, cookies, and other relevant data. The captured responses are used for performance measurement, validation, and analysis.

Performance Monitoring: Load Runner can also integrate with various monitoring tools to capture system-level metrics during the test execution. This allows you to monitor the performance and behavior of the application under load, including server performance, database metrics, network statistics, and more.

**Q6 How many VUsers are required for load testing?**

Define your testing goals: Determine what you want to achieve with your load testing. Are you trying to identify performance bottlenecks, measure system capacity, or validate specific performance requirements?

Analyze user behavior: Understand the expected usage patterns and behavior of your target users. This includes factors such as the number of concurrent users, their activities, and the frequency of their interactions with the system.

Set performance goals: Define the performance goals for your application, such as response time thresholds, throughput requirements, or maximum acceptable error rates.

Conduct a pilot test: Start with a smaller number of VUsers and perform a pilot test to observe the system's behavior under load. Gradually increase the number of VUsers until you start seeing performance degradation or reach your defined performance goals.

Analyze test results: Monitor the system's performance metrics during the test, such as response times, CPU and memory usage, database performance, etc. Identify any performance bottlenecks or issues that arise.

Scalability testing: If your goal is to measure the system's scalability, gradually increase the number of VUsers beyond the expected user load to determine the maximum capacity the system can handle.

Consider realistic load scenarios: It's essential to simulate realistic load scenarios that mimic the expected user behavior. For example, if your application has peak usage periods, simulate that load during testing.

Consult load testing guidelines: Various load testing guidelines and best practices are available that can provide recommendations for determining the number of VUsers based on factors such as server capacity, response time goals, and desired system behavior.

**Q7 What isthe relationship between Response Time and Throughput?**

Response time and throughput are related. The response time for an average transaction tends to decrease as you increase overall throughput. However, you can decrease the response time for a specific query, at the expense of overall throughput, by allocating a disproportionate amount of resources to that query.

**Q8 What is Automation Testing?**

Automated testing is a process that validates if software is functioning appropriately and meeting requirements before it is released into production. This software testing method uses scripted sequences that are executed by testing tools.

**Q9 Which Are The Browsers Supported By Selenium Ide?**

Selenium IDE has add-ons for Firefox and Chrome browsers. Selenium IDE comes with a rich set of commands that are powered by Selenese, and it allows you to record and test different interactions of a web application with the browser.

**Q10 What are the benefits of Automation Testing?**

Automation speeds up test cycles and removes repetitive monotonous test cases and allows better test case scenarios by improving Faster communication among designers, coders, and customers. Potential glitches can be easily rectified. Thus, automation provides a better quality of the end product.

**Q11 What are the advantages of Selenium?**

Selenium is an open-source automation testing tool and it is free of cost to use. Selenium provides high tester flexibility to write advanced and complex test cases. Supports test case execution on multiple operating systems such as Windows, Linux, Android, Mac, and iOS.

**Q12 Why testers should opt for Selenium and not QTP?**

Selenium, however, supports a wide range of programming languages. QTP/UFT test scripts run only on the Windows environment. They cannot be run across all browsers. On the other hand, Selenium is OS independent and allows test scripts to run across all browsers