1. **Which components have you used in Load Runner?**

**Ans.** – Load runner is a performance testing tool that consists of several components, including:

* Virtual user generator (VuGen): Used to record user actions and create scripts for simulating user activity.
* Controller: Used to organize, execute, and monitor tests by managing multiple virtual users.
* Load Generators: These simulate the actual users by generating the load on the system under test.
* Analysis: Provides in- depth analysis of test results, including graphs and repots to identify performance bottlenecks.
* Monitoring: Monitors the system under test during test execution to gather performance metrics.

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

**Ans. –** To set the number of Vusers in Load Runner, you typically use the Load Runner Controller Tool. Within the Controller, you can specify the number of vusers by configuring the “Number of Vusers” field in the scenario settings. You can either specify a fixed number of Vusers or use different load patterns to ramp up or down the number of Vusers over time.

1. **What is Correlation?**

**Ans.** - Correlation is the process of capturing dynamic data from server responses and replacing them with unique parameters during script execution. This ensures accurate simulation of user behavior by handling session – specific values like session IDs or timestamp.

1. **What is the process for developing a Vuser Script?**

**Ans. -** Following are the several steps to developing a Vuser script:

1. Understanding Requirements: Gather requirement from stakeholder to determine the scenarios to be simulated.
2. Recording: Use a performance testing tool to record user actions on the application under test.
3. Enhancing the Script: Modify the recorded script to handle dynamic data, parameterize inputs, and add logic for error handling and verification.
4. Correlation: Identify dynamic values in the scripts(such as session IDs or timestamps) and correlate them to ensure realistic simulation.
5. Parameterization: Replace hard coded values with parameters to simulate multiple users with varying data inputs.
6. Data – Driving: f necessary, integrate external data sources to drive the script’s input dynamically.
7. Script Validation: Validate the script to ensure it accurately reflects the desired user behavior and objectives.
8. Execution and Analysis: Execute the script against the application to simulate user load and analyze performance metrics to identify bottlenecks and area for optimization.
9. Iteration and optimization: Based on the analysis, refine the script as needed to improve performance and achieve desired testing goals.
10. Documentation: Document the script, including any parameterization, correlation or special configurations, for future reference and maintenance.
11. **How Load Runner interacts with the application?**

**Ans.-**  Load Runner interacts with an application by simulating multiple users accessing the application simultaneously. It does this by crating virtual users that emulate real user behavior, such as clicking buttons, filling out forms, and navigating through application. Load Runner then measures the application’s response time, throughout and resource utilization under various load condition to assess its performance and identify bottlenecks.

1. **How many VUsers are required for load testing?**

**Ans**.- The number of virtual users required for load testing depends on the various factors, including the expected user load on the application, the complexity of user transaction, and the performance goals of the test. For example, Time taken for 1 user to complete 1 transaction: [a] + [b ]+[c] = 6 second In 1 hour a user can do: 60\*60/6 = 600 transactions per hour, we need: 2,80,000/600 = 467 Vusers After Identifying the scripts and the total Vusers, plan the Load Runner test scenario for the same.

1. **What is the relationship between Response Time and Throughput?**

**Ans.** – Response time refers to the time taken for the system to respond to a user request. Throughout, on the other hand, measures the number of transactions processed by the system per unit of time. It indicates how many requests the system can handle in a given time frame, usually measured in requests per second. The relationship between response time and throughout is often inversely proportional. As the response time decreases, the throughout tends to increase, and vice versa. However, it’s essential to find the balance between these two metrics. Sometimes, increasing throughout may lead to an increase in response time due to resource contention or other factors.

1. **To test the Performance testing on “Tops Technologies website” :-** [**https://www.saucedemo.com**](https://www.saucedemo.com)

**Ans.-**