**MODULE-4 (AUTOMATION CORE TESTING)**

**LOAD RUNNER UP AND SELENIUM IDE**

1. Which components have you used in Load Runner?

* LoadRunner is a comprehensive performance testing tool by Micro Focus that helps simulate user activity and monitor system performance. The primary components of LoadRunner include:
* VuGen (Virtual User Generator):
* Controller
* Load Generators (LGs)
* Analysis
* Agent Process
* Monitoring and Diagnostics

1. How can you set the number of V-users in Load Runner?

* In LoadRunner, you can set the number of Virtual Users (Vusers) through the Controller component. Here's a step-by-step guide on how to do this:
* Steps to Set the Number of Vusers in LoadRunner:
* Launch LoadRunner Controller:
* Create a New Scenario
* Add Vuser Scripts
* Configure the Number of Vusers
* Configure Vuser Initialization and Ramp-Up
* Save the Scenario
* Run the Scenario

1. What is Correlation?

* **Correlation** in LoadRunner is the method of capturing dynamic values from server responses and using them in subsequent client requests to ensure the realistic simulation of user behaviour and accurate performance testing results.

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

* Developing a Vuser (Virtual User) script in LoadRunner involves several key steps. These steps help ensure that the script accurately simulates user actions and interactions with the application under test. Here’s a comprehensive process for developing a Vuser script:
* Steps for Developing a Vuser Script:
* Planning and Preparation:
* Recording the Script
* Enhancing the Script
* Adding Think Time and Pacing
* Enhancing Script with Additional Logic (if needed)
* Running and Analysing
* Finalizing the Script

1. How Load Runner interacts with the application?

* LoadRunner interacts with an application to simulate real user behaviour and measure the performance of the system under load. This interaction involves several components and processes that work together to create, execute, and analyse test scenarios. Here's a detailed overview of how LoadRunner interacts with an application:
* Components Involved in LoadRunner Interaction
* VuGen (Virtual User Generator)
* Controller
* Load Generators (LGs)
* Analysis
* Steps of Interaction
* Script Recording
* Script Enhancement
* Scenario Configuration
* Load Generation
* Monitoring and Data Collection
* Result Analysis

1. How many VUsers are required for load testing?

* Determining the number of Virtual Users (VUsers) required for load testing depends on various factors, including the specific goals of the test, the expected usage patterns, the type of application, and the performance criteria you need to meet.
* The number of VUsers required for load testing will vary based on the application's specifics and the goals of the test. The process involves:
* Analysing user behaviour and peak usage patterns.
* Setting clear performance criteria.
* Running preliminary and incremental tests to identify the system's capacity and performance limits.

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

* Response time and throughput are two key performance metrics used in load testing and performance monitoring. While they are related, they measure different aspects of system performance.
* While response time and throughput are related, they are not directly proportional. In some cases, as throughput increases, response time may also increase, and vice versa.
* A system with high throughput may still have high response times if it's struggling to process a large number of concurrent requests.
* Conversely, a system with low throughput may have low response times if it's not under heavy load. The relationship between response time and throughput is influenced by various factors, including system resources, concurrency, network latency, application design, and workload characteristics.

1. To test the Performance testing on “Tops Technologies website”: - <https://www.saucedemo.com/>

Action ()

{

web\_url("gts1c3.der",

"URL=http://pki.goog/repo/certs/gts1c3.der",

"Resource=1",

"RecContentType=application/pkix-cert",

"Referer=",

"Snapshot=t5.inf",

LAST);

web\_url("gtsr1.der",

"URL=http://pki.goog/repo/certs/gtsr1.der",

"Resource=1",

"RecContentType=application/pkix-cert",

"Referer=",

"Snapshot=t6.inf",

LAST);

web\_url("gts1c3.der\_2",

"URL=http://pki.goog/repo/certs/gts1c3.der",

"Resource=1",

"RecContentType=application/pkix-cert",

"Referer=",

"Snapshot=t7.inf",

LAST);

web\_url("gtsr1.der\_2",

"URL=http://pki.goog/repo/certs/gtsr1.der",

"Resource=1",

"RecContentType=application/pkix-cert",

"Referer=",

"Snapshot=t8.inf",

LAST);

web\_url("RapidSSLTLSRSACAG1.crt",

"URL=http://cacerts.rapidssl.com/RapidSSLTLSRSACAG1.crt",

"Resource=1",

"RecContentType=application/pkix-cert",

"Referer=",

"Snapshot=t9.inf",

LAST);

return 0;

}

1. create a normal script of above website with correlate using hp default website.

Action ()

{

web\_url("index.htm",

"URL=http://127.0.0.1:1080/WebTours/index.htm",

"Resource=0",

"Referer=",

"Snapshot=t1.inf",

"Mode=HTML",

LAST);

web\_url ("header.html",

"URL=http://127.0.0.1:1080/WebTours/header.html",

"Resource=0",

"Referer=http://127.0.0.1:1080/WebTours/index.htm",

"Snapshot=t2.inf",

"Mode=HTML",

LAST);

web\_url("welcome.pl",

"URL=http://127.0.0.1:1080/cgi-bin/welcome.pl?signOff=true",

"Resource=0",

"RecContentType=text/html",

"Referer=http://127.0.0.1:1080/WebTours/index.htm",

"Snapshot=t3.inf",

"Mode=HTML",

EXTRARES,

"Url=http://pki.goog/repo/certs/gts1c3.der", "Referer=", ENDITEM,

"Url=http://pki.goog/repo/certs/gtsr1.der", "Referer=", ENDITEM,

LAST);

lr\_save\_string(lr\_decrypt("6620c77998a0b4f6"), "PasswordParameter");

web\_submit\_data("login.pl",

"Action=http://127.0.0.1:1080/cgi-bin/login.pl",

"Method=POST",

"RecContentType=text/html",

"Referer=http://127.0.0.1:1080/cgi-bin/nav.pl?in=home",

"Snapshot=t4.inf",

"Mode=HTML",

ITEMDATA,

"Name=userSession", "Value=138794.996921831HVDQczHpzftVzzzHtciDfpzAfQcf", ENDITEM,

"Name=username", "Value=jojo", ENDITEM,

"Name=password", "Value={PasswordParameter}", ENDITEM,

"Name=login.x", "Value=47", ENDITEM,

"Name=login.x", "Value=8", ENDITEM,

"Name=JSFormSubmit", "Value=off", ENDITEM,

LAST);

return 0;

}

1. What is Automation Testing?

* Automation testing is a software testing technique that uses specialized tools and scripts to automatically execute test cases, compare actual outcomes with expected outcomes, and report results. This approach aims to reduce the manual effort required in repetitive testing tasks, improve test coverage, and enhance the accuracy and efficiency of the testing process.

1. Which Are the Browsers Supported by Selenium Ide?

* Selenium IDE (Integrated Development Environment) is primarily a Firefox extension. It's designed as a browser extension for Mozilla Firefox and is tightly integrated with the Firefox browser.
* Mozilla Firefox: Selenium IDE is available as an add-on for Firefox. It was originally developed as a Firefox plugin and continues to offer robust support for recording and executing test scripts in Firefox.
* Google Chrome: Selenium IDE can be installed as a Chrome extension from the Chrome Web Store. It provides a user-friendly interface for recording and playing back test scripts directly within the Chrome browser.
* Microsoft Edge: Selenium IDE may also have experimental support for Microsoft Edge, either through a dedicated Edge extension or compatibility with Chrome extensions (since Edge is based on Chromium).
* Other Browsers: Support for other browsers may vary, and it's recommended to check the latest documentation or announcements from the Selenium IDE project for updates on browser support.

1. What are the benefits of Automation Testing?

* Automation testing offers numerous benefits that significantly enhance the efficiency, reliability, and effectiveness of software testing processes. Here are the key advantages of automation testing:
* Speed and Efficiency
* Improved Accuracy
* Increased Test Coverage
* Reusability
* Cost Savings
* Early Detection of Defects
* Facilitates Continuous Integration and Continuous Deployment (CI/CD)
* Supports Agile and DevOps Practices
* Improved Product Quality
* Customer Satisfaction

1. What are the advantages of Selenium?

* Selenium is a powerful and widely used open-source automation testing framework for web applications. It provides several advantages that make it a preferred choice among testers and developers. Here are the key advantages of Selenium:
* Cross-Browser Compatibility
* Support for Multiple Programming Languages
* Integration with Testing Frameworks and Tools
* Open-Source and Active Community Support
* Flexibility and Extensibility
* Support for Parallel Testing
* Robust Browser Automation
* Rich Ecosystem of Tools and Libraries
* Suitable for Agile and DevOps Practices
* Industry Adoption and Maturity

1. Why testers should opt for Selenium and not QTP?

* Testers should select Selenium over QTP (UFT) due to Selenium's open-source nature, which eliminates licensing costs, making it more cost-effective for organizations. Selenium supports multiple browsers and operating systems, offers flexibility with various programming languages, and integrates seamlessly with CI/CD tools. Its active community ensures frequent updates and robust support. In contrast, QTP/UFT, as a commercial tool, incurs licensing expenses, supports fewer languages, and historically focuses more on Windows-based applications. Selenium's scalability, cross-platform compatibility, and adaptability to modern testing needs make it a preferred choice for agile teams aiming for efficient and flexible test automation solutions.