**ASSIGNMENT 5**

**Module-4**

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

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

These components work together to facilitate the entire performance testing process, from script creation to test execution, monitoring, analysis, and reporting.

* Virtual user generator(vugen)
* Controller
* Load generator
* Agent process
* Analysis
* Monitoring tools

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

In LoadRunner, the number of Virtual Users (Vusers) can be set through the LoadRunner Controller. Here's steps:

1. Open Load Runner Controller

2. Create or Open a Scenario

3. Define Vuser Groups

4. Set the Number of Vusers

5. Distribute Vusers

6. Configure Ramp-Up

7. Run the Scenario

8. Monitor Execution.

3.**What is Correlation?**

Correlation, in the context of LoadRunner or performance testing, refers to the process of capturing and replacing dynamic values in a recorded script with parameters. Dynamic values are those that change with each user session or interaction, such as session IDs, authentication tokens, timestamps, or any other unique identifiers generated by the server.

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

Following development process in Vuser Script

1. Recording the Vuser script
2. Edit the Vuser script
3. Runtime setting.
4. Run the Vuser script in stand-alone mode.
5. Incorporate the Vuser script into a load runner scenario.

**5.How Load Runner interacts with the application?**

LoadRunner interacts with applications primarily through a process called performance testing. LoadRunner is a performance testing tool that simulates user activity on software applications, allowing testers to measure and analyse system performance under various conditions. LoadRunner provides a robust framework for assessing an application's performance under realistic conditions, helping organizations ensure that their software meets performance objectives and user expectations. Load runner is typically interacts with the an application through these points; scripting, parameterization, scenario design, controller setup, execution, monitoring, analysis, reporting,

**6.How many Vusers are required for load testing?**

Determining the number of Virtual Users (Vusers) required for load testing depends on several factors, including the objectives of the test, the nature of the application, the expected user load in production, and the available infrastructure. calculating No. of virtual users for our load test but when doing performance testing, never tested the application against any specific user set. Always perform different user tests to determine the system state at different loads. In independent testing cases the incremental approach to test the application by increasing the virtual users test by test.

**7.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.

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

}

**9.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;

}

**10. What is Automation Testing?**

Automation testing is a software testing technique that involves using automated tools and scripts to execute test cases and verify the behaviour and performance of software applications. Instead of manual intervention, automation testing relies on pre-written scripts to perform repetitive and complex test scenarios, thereby increasing testing efficiency, repeatability, and accuracy.

**11. 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.

1. **Mozilla Firefox:** Selenium IDE is primarily developed for Mozilla Firefox and offers the most comprehensive support for this browser.
2. **Google Chrome:** Selenium IDE also provides experimental support for Google Chrome. There is a separate Chrome extension available, but the functionality may be limited compared to the Firefox version, and it may still be in development or testing phases.
3. **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).
4. **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.

**12. What are the benefits of Automation Testing?**

Automation testing offers numerous benefits that contribute to improving the efficiency, effectiveness, and quality of software development processes. Here are some of the key benefits

Early detection defect

Faster time to market

Increased testing coverage

Repeatability and consistency

Improved accuracy

Support for continuous integration

Scalability

Enhanced developer productivity

**13.What are the advantages of Selenium?**

Selenium is a widely used open-source tool for automating web browsers. It offers several advantages that make it a preferred choice for automated testing of web applications. Selenium offers a powerful and flexible platform for automating web browser testing, helping organizations improve the efficiency, reliability, and quality of their web applications.

* **Open Source**: Selenium is an open-source tool, which means it is freely available for anyone to use. This makes it cost-effective for organizations and individuals.
* **Cross-platform Compatibility**: Selenium supports multiple operating systems like Windows, macOS, and Linux. It also works with different web browsers such as Chrome, Firefox, Safari, and Internet Explorer.
* **Language Support**: Selenium provides support for multiple programming languages including Java, C#, Python, Ruby, and JavaScript. This allows testers to choose a language they are familiar with and comfortable using.
* **Extensibility:** Selenium can be extended with third-party libraries and frameworks to enhance its functionality. For example, frameworks like TestNG and JUnit can be integrated with Selenium to create robust test automation solutions.
* **Large Community and Active Support**: Selenium has a vast community of users who actively contribute to its development and provide support through forums, blogs, and online communities. This ensures that users can find help and resources easily
* **Integration with CI/CD Tools**: Selenium can be seamlessly integrated with Continuous Integration and Continuous Deployment (CI/CD) tools such as Jenkins and Bamboo. This enables automated test execution as part of the software development and release pipeline.
* **Powerful Testing Capabilities**: Selenium supports a wide range of testing functionalities, including functional testing, regression testing, and browser compatibility testing. It also provides features for handling pop-ups, alerts, and dynamic web elements.
* **Flexibility:** Selenium offers flexibility in terms of test execution. It can run tests locally on the tester’s machine or on remote machines, allowing for distributed test execution.
* **Robust Test Reporting**: Selenium generates detailed test reports with information about test cases, test results, and execution logs. These reports help in identifying issues and tracking test progress.

**14. Why testers should opt for Selenium and not QTP?**

Choosing between Selenium and QTP (now known as UFT - Unified Functional Testing) depends on various factors, including the project requirements, budget, skillset of the testing team, and the specific needs of the organization.

The best test automation tool for you will depend on your specific needs and requirements. If you are looking for a commercial tool with a wide range of features and support for record-and-playback, then QTP/UFT is a good choice. If you are looking for an open-source tool that supports a wide range of programming languages and operating systems, then Selenium is a good choice.