# Assignment-4

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

1. **Virtual User Generator**

➢ I used VuGen to create automated performance test scripts. It allowed me to record user actions on the application and edit them by adding logic, parameterization (to vary input data), and correlation (to handle dynamic values).

2. **Controller**

➢ The Controller was used to design and execute performance test scenarios. I configured the number of virtual users, test duration, pacing, and think time. I also assigned scripts to different load generators to simulate concurrent users.

3. **Analysis**

➢ These are machines used to simulate the load on the application. I used them to distribute the virtual user load across different systems, which helped in testing how the application performs under stress from multiple locations or users

4. **Load Generators**

➢ After executing the tests, I used the Analysis component to review performance results. It helped me generate detailed graphs and reports that showed key metrics like response time, throughput, errors, and system resource usage. This helped in identifying bottlenecks and performance issues.

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

➢ Here's how it can be done**:**

1.Open the Controller and create or load a test scenario.

2.Add the script you want to run by clicking on “Add Scripts”.

3. In the “Groups” or “Scenario Groups” section, select the script and

● Set the number of Vusers you want to simulate for that script.
● You can do this in the “Vusers” column or through the Group Properties window.

4) You can also configure:

● Ramp-up: How quickly the Vusers should start
● Duration: How long the Vusers should stay active.
● Ramp-down: How the users should stop (gradually or all at once).

5) During the test execution, you can monitor the active Vusers in real time and adjust if needed.

**3. What is Correlation?**

➢ Correlation in LoadRunner is the process of handling dynamic values that are returned by the server during test script execution.

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

➢ The process of developing a Vuser (Virtual User) Script in LoadRunner involves several structured steps using the Virtual User Generator (VuGen). These steps ensure the script is reliable, reusable, and can accurately simulate real user behavior under load

**1. Identify Test Scenario**

● Understand the business process to be tested (e.g., login, search, checkout).
● Define the user actions, data inputs, and expected outputs.

**2. Select the Appropriate Protocol**
Choose the correct protocol for the application (e.g., Web HTTP/HTML, TruClient, Web Services, Citrix).

**3. Record the Script**
● Launch VuGen and start recording the business process.
● Perform the actions on the application exactly as a real user would.
● VuGen generates the script by capturing requests/responses.

**4. Enhance the Script 5. Replay and Debug**
● Run the script in VuGen to verify functionality.
● Use runtime viewer and log settings (standard/extended logs)
to debug and resolve errors.

**6. Save and Integrate into Scenario**
● Once the script is validated, save it and import it into the LoadRunner Controller.
● Use it to design load scenarios with defined virtual users and test conditions.

**7. Final Testing**
● Run the script under load, monitor results, and fine-tune based on application behavior

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

➢ LoadRunner interacts with the application under test by simulating real user behavior at the protocol level. It does this by generating virtual users (Vusers) that perform predefined actions, such as logging in, searching, or submitting forms, just like actual users would.

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

➢ The number of VUsers required for load testing depends on the expected number of concurrent users, test objectives (like load or stress testing), and the application's infrastructure capacity. It is typically based on real-world usage patterns during peak load conditions.

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

1. Response Time is the time taken by the system to process a single user request and send back a response.
2. Throughput is the number of requests processed by the system per unit of time (e.g., transactions per second).
3. When throughput is low, response time is usually fast because the system is lightly loaded.
4. As throughput increases (more users or requests), response time may increase due to system resource contention.
5. At peak capacity, a small increase in throughput can cause a large increase in response time.
6. Therefore, response time and throughput are inversely related when the system is heavily loade

**8.To test the Performance testing on “Tops Technologies website”**

**1. to Record all top level menu**

- Open LoadRunner VuGen (Virtual User Generator).
- Select the appropriate protocol, usually Web HTTP/HTML for web applications.
- Start recording by launching the browser from VuGen.
- Navigate to https://www.saucedemo.com/ and perform actions on all top-level menu items you want to test.
- Stop recording when all menu interactions are captured
- . Save the recorded script for further enhancements (parameterization, correlation, etc.).

**2. to Record minimum 10 Vuser on this website**

- Open the LoadRunner Controller.
- Create a new Scenario and add the previously recorded script.
- Set the number of Vusers to 10 in the scenario settings to simulate 10 virtual users.
- Configure ramp-up time and test duration as per requirements.
- Assign load generators if needed (local machine or remote load generators).
- Save the scenario.

**3. save all (Script,Design,Graph)**

- Save the VuGen script after recording and enhancing it.
- Save the Controller scenario design after configuring Vusers and test parameters.
- Execute the scenario to run the test.
- After the test completes, open LoadRunner Analysis.
- Generate and save performance graphs and reports showing response times, throughput, hits per second, errors, etc.
- Save all files (scripts, scenario designs, analysis reports) properly for submission and future reference.

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

```
Action()
{
 // Start the transaction for login

// Open the login page
web_url("Open_Login_Page",
"URL=https://www.saucedemo.com/"
"TargetFrame=",
"Resource=0",
"RecContentType=text/html",
"Referer=",
"Snapshot=t1.inf",
"Mode=HTML",
LAST);


// Correlate a dynamic token (example placeholder) // Usually,
 you capture a token or session ID from server response
 web_reg_save_param("session_id",
 "LB=name=\"sessionToken\" value=\""
, "RB=\"",
"Search=Body",
LAST);

// Submit login form with correlated session token (if applicable)
web_submit_data("login",
"Action=https://www.saucedemo.com/",
"Method=POST",
"RecContentType=text/html",
"Referer=https://www.saucedemo.com/",
ITEMDATA,
"Name=user-name", "Value=standard_user", ENDITEM,
"Name=password", "Value=secret_sauce", ENDITEM,
// Use correlated session token if required
//"Name=sessionToken", "Value={session_id}",
ENDITEM, LAST);

lr_end_transaction("Login", LR_AUTO);

// Start transaction to open inventory page
lr_start_transaction("Open_Inventory_Page");
web_url("Inventory",
"URL=https://www.saucedemo.com/inventory.html",
"TargetFrame=",
"Resource=0",
"RecContentType=text/html",
"Referer=https://www.saucedemo.com/",
"Snapshot=t3.inf",
"Mode=HTML",
LAST);
lr_end_transaction("Open_Inventory_Page", LR_AUTO);
 return 0;
}
```

### 10.What is Automation Testing?

- Automation Testing is the process of using specialized software tools to execute pre-scripted tests on a software application automatically.
- it helps verify that the application behaves as expected, reduces manual effort, and increases test accuracy and efficiency.
- Automation Testing is commonly used for repetitive, regression, and load tests to save time and improve test coverage.

### 11. Which Are The Browsers Supported By Selenium Ide?

- Selenium IDE primarily supports the following browsers as extensions or add-ons:

1. Mozilla Firefox (originally the first supported browser)
2. Google Chrome (supported via official Selenium IDE extension)

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

**Advantages:**

- 70% faster than the manual testing
- Wider test coverage of application features
- Reliable in results
- Ensure Consistency
- Saves Time and Cost
- Improves accuracy
- Human Intervention is not required while execution
- Increases Efficiency

- Better speed in executing tests
- Reusable test scripts
- Test Frequently and thoroughly
- More cycle of execution can be achieved through automation

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

**-> Tester should opt for Selenium reason is as the following.**

1) **Open Source and Free:** Selenium is free to use, while QTP (now UFT) is a licensed, paid tool.
2) **Supports Multiple Browsers**: Selenium works across many browsers (Chrome, Firefox, Edge, Safari), whereas QTP has limited browser support.

3) **Supports Multiple Programming Languages**: Selenium supports Java, Python, C#, Ruby, JavaScript, etc., offering more flexibility; QTP primarily supports VBScript.

4) **Cross-Platform Support**: Selenium runs on Windows, Linux, and MacOS, whereas QTP is Windows-only.

5) **Better for Web Applications**: Selenium is designed specifically for web automation, while QTP is more general-purpose but heavier.

6) **Large Community and Support**: Selenium has a vast user community and frequent updates due to its open-source nature.

**15. To validate the tops technologies website Contact us page and enter your friend detail at last “Login and sidemenu” https://www.saucedemo.com/**

Side file

```
{
  "id": "12345-67890",
  "version": "2.0",
  "name": "Tops and SauceDemo Validation",
  "url": "https://www.tops-int.com/",
  "tests": [
    {
      "id": "test1",
      "name": "Validate Tops Technologies Contact Us Page",
      "commands": [
        {"command": "open", "target": "https://www.tops-int.com/contact-us/", "value": ""},
        {"command": "type", "target": "name=name", "value": "Rahul"},
        {"command": "type", "target": "name=email", "value": "rahul@gmail.com"},
        {"command": "type", "target": "name=phone", "value": "9876543210"},
        {"command": "type", "target": "name=message", "value": "Interested in Python course."},
        {"command": "click", "target": "xpath=//button[contains(text(),'Submit')]", "value": ""},
        {"command": "pause", "target": "3000", "value": ""},
        {"command": "open", "target": "https://www.saucedemo.com/", "value": ""},
        {"command": "type", "target": "id=user-name", "value": "standard_user"},
        {"command": "type", "target": "id=password", "value": "secret_sauce"},
        {"command": "click", "target": "id=login-button", "value": ""},
        {"command": "pause", "target": "2000", "value": ""},
        {"command": "click", "target": "id=react-burger-menu-btn", "value": ""},
        {"command": "pause", "target": "2000", "value": ""}
      ]
    }
  ]
}
```

**JAVA FILE:**

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

public class TopsAndSauceDemoTest {
    public static void main(String[] args) throws InterruptedException {

        System.setProperty("webdriver.chrome.driver", "C:\\chromedriver\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();

        driver.get("https://www.tops-int.com/contact-us/");
        driver.manage().window().maximize();

        driver.findElement(By.name("name")).sendKeys("Rahul");
        driver.findElement(By.name("email")).sendKeys("rahul@gmail.com");
        driver.findElement(By.name("phone")).sendKeys("9876543210");
        driver.findElement(By.name("message")).sendKeys("Interested in Python course.");

driver.findElement(By.xpath("//button[contains(text(),'Submit')]"));
        submitButton.click();
        Thread.sleep(3000);


        driver.get("https://www.saucedemo.com/");
        driver.findElement(By.id("user-name")).sendKeys("standard_user");
        driver.findElement(By.id("password")).sendKeys("secret_sauce");
        driver.findElement(By.id("login-button")).click();
        Thread.sleep(2000);


        driver.findElement(By.id("react-burger-menu-btn")).click();
        Thread.sleep(2000);


        System.out.println("Side menu opened successfully.");


        driver.quit();
    }
}
```