

JIET GROUP OF INSTITUTIONS



Student Name Roll No.

Experiment No. Date

Software Testing -

1. Testing - Testing is a process of evaluating a system or its components with the intent to find whether it satisfies the specified requirement or not.

Testing is executing a system in order to identify any gaps, errors or missing requirements in contrary to the actual requirements.

Professions involved in testing are -

- a) Software Tester
- b) Software developer
- c) Project Lead / Manager
- d) End User

Start of Testing -

An early start to testing reduces the cost & time to rework & produce error free s/w that is delivered to client.

Testing is done in different forms at every phase of SDLC -

- During the requirement gathering phase, the analysis & verification of requirements are also considered as testing.
- Reviewing the design in the design phase with the intent to improve the design is also considered as testing.

JIET GROUP OF INSTITUTIONS



Student Name Roll No.

Experiment No. Date

- Testing performed by a developer on completion of the code is also categorized as testing.

Stop Testing -

The following aspects are to be considered for stopping the testing process -

- Testing deadlines.
- Completion of test case execution.
- Completion of functional & code coverage to a certain point.
- Bug rate falls below a certain level & no-high priority bugs are identified.
- Management decision.

Verification & Validation -

Verification

1. Verification addresses the concern : "Are you building it right?"

Validation

Validation addresses the concern : "Are you building the right thing?"

2. Ensures that the system meets all the functional requirements.
3. It occurs after verification.

Ensure that the functional requirements meet the intended behaviour.

3. It takes place first & includes the checking for documentation, & mainly involves the code etc.

JIET GROUP OF INSTITUTIONS



Student Name Roll No.

Experiment No. Date

- | | | |
|----|--|---|
| 4. | Done by developers. | Done by tester. |
| 5. | It has static activities, as it includes collecting reviews as it testing includes walkthrough, & inspect to verify a s/w. | It has dynamic activities, executing the s/w against the requirements. |
| 6. | It is an objective process no subjective decision should be made to verify a s/w. | It is a subjective process involves subjective decisions on how well a s/w works. |

Types of Testing -

- 1) Manual Testing - It includes testing s/w manually i.e without using any automated tools or any script.
- 2) Automated testing - Is when the tester writes scripts & uses another s/w to test the product.

System Testing Method-

1. Black box Testing - The internal workings of an applicatn need not be known.
→ Also known as closed box testing, data driven testing, or functional testing.
2. Grey box Testing - The tester has limited knowledge of the internal workings of the applicatn.
→ Also known as translucent testing, as the tester

JIET GROUP OF INSTITUTIONS



Student Name Roll No.

Experiment No. Date

has limited knowledge of the insides of the application.

3. White Box Testing - Tester has full knowledge of the internal workings of the application.
→ Also known as clear box testing, structure testing, or code based testing.

JIET GROUP OF INSTITUTIONS



Student Name Roll No.

Experiment No. Date

JUnit -

- The JUnit Platform serves as a foundation for launching testing frameworks on the JVM.
- It also defines the ~~testng~~ TestEngine API for developing a testing framework that runs on the platform.
- Furthermore, the platform provides a console Launcher to launch the platform from the command line & build plugins for Oracle & Maven as well as a JUnit 4 based Runner for running any TestEngine on the platform.
- JUnit Jupiter is the combination of the new programming model & extension model for writing test extensions in JUnit 5. The Jupiter sub-project provides a TestEngine for running Jupiter-based tests on the platform.
- JUnit Vintage provides a TestEngine for running JUnit 3 & JUnit 4 based tests on the platform.

JUnit 5 = JUnit Platform + JUnit Jupiter + JUnit Vintage.

JIET GROUP OF INSTITUTIONS



Student Name Roll No.:
Experiment No. Date

Objective - To test & run assertEquals Function.

System Used - Netbeans, Windows 7, JUnit 4.

Code -

```
Kriitika.java
package kriitika;
public class kriitika {
    public int add(int a, int b) {
        return (a+b);
    }
}
```

kriitikaTT.java

@ Test

```
public void testMain() {
    kriitika obj = new kriitika();
    assertEquals(10, obj.add(5,5));
}
```

Result - Test passed 100%.

JIET GROUP OF INSTITUTIONS



Student Name Roll No.

Experiment No. Date

Objective - To test & run assertTrue funct.

System Requirement - Netbeans, Windows 7, JUnit 4.

Code -

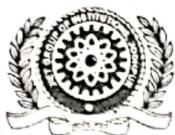
```
kritika1.java
package kritika ;
public class kritika1 {
    public boolean isEvenNumber(int n) {
        boolean result = false;
        if (n % 2 == 0)
            result = true ;
        }
        return (result);
    }
}
```

kritika1Test.java

```
@Test
public void testSomeMethod() {
    kritika1 obj = new kritika1();
    assertTrue(obj.isEvenNumber(4));
}
```

Result - Test passed 100.00%.

JIET GROUP OF INSTITUTIONS



Student Name Roll No.

Experiment No. Date

Objective - To test & run assertFalse functn.

System Requirement - Netbeans, JUnit4, Windows

Code -

kritikajava,

package kritika;

public class kritika2 {

 public boolean isEven (int n) {

 boolean result = false;

 if (n%2 == 0)

 result = true;

}

 return (result);

}

}

kritika2TT.java.

@Test

 public void TestSomeMethod() {

 kritika2 obj = new kritika2();

 assertFalse (obj.isEven(3));

}

}

Result - Test passed 100.00%.

JIET GROUP OF INSTITUTIONS



Student Name Roll No.

Experiment No. Date

Objective - To test & run assertNotNull function.

System Used - Netbeans, JUnit, Windows.

Code -

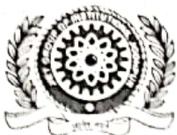
```
package kritika;
import java.util.*;
public class NotNull {
    public String notNull() {
        String s1 = "Not Null";
        return s1;
    }
}
```

NotNullIT.java.

② Test

```
public void testNotNull {
    NotNull obj = new NotNull();
    assertEquals("Not Null", obj.notNull());
}
```

Result - The Test passed 100.00 %.



JIET GROUP OF INSTITUTIONS

Student Name Roll No.

Experiment No. Date

Object - To test & implement assertNull() functn.

System Required - Windows 7, Netbeans, JUnit.

Code - Null.java

```
package krutika;
public class Null {
    public String null()
    { string sI = null;
        return(sI);
    }
}
```

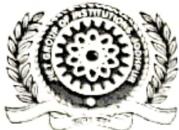
NullIT.java

@ Test

```
public void null()
{
    Null obj = new Null();
    assertNull(obj.null());
}
```

Result - The Test passed 100.00%.

JIET GROUP OF INSTITUTIONS



Student Name Roll No.:

Experiment No. Date

Object - To run & implement assertNotSame() funct.

System Used - Windows, JUnit, Netbeans.

Code - Notsame.java

```
package kritika;
public class Notsame {
    public String notsame (int a)
    {
        if (a == 0)
            { String s1 = "abc";
              return (s1);
            }
        else
            { String s1 = "bAc";
              return (s1);
            }
    }
}
```

NotsameIT.java

```
@Test
public void Test notsame()
{
    Notsame obj = new Notsame ();
    assertEquals (obj.notsame (0), obj.notsame (1));
}
```

Result - The test passed 100.00 % ;



JIET GROUP OF INSTITUTIONS

Student Name Roll No.

Experiment No. Date

Object- To run & implement assert same ()

System Used- Window 7, Netbeans, JUnit.

Code- Same.java

```
package kritika;
public class Same {
    public String same(int a)
    {
        if (a==0)
        {
            String s1 = "abc";
            return (s1);
        }
        else
        {
            String s1 = "Abc";
            return (s1);
        }
    }
}
```

Same JT.java.

@Test

```
public void testSame()
{
    Same obj = new Same();
    assertEquals(obj.same(0), obj.same(0));
}
```

~~See
Topic~~

Result - The Test passed 100.00%.

JIET GROUP OF INSTITUTIONS



Student Name Roll No.

Experiment No. 3 Date

Object - WAP that calculates the area & perimeter of the circle & find the coverage & test cases of the program.

Resources used - Windows 7, JUnit, Netbeans.

Code - area.java,

```
package krutika;
public class area {
    float pi = 3.14f;
    float perimeter (float radius)
    {
        return (2 * pi * radius);
    }
}
```

```
float area-circle (float radius)
{
    return (pi * radius * radius);
}
```

AreaIT.java

@ Test

```
public void test_perimeter ()
{
    area obj = new area();
    assertEquals (62.8f, obj.perimeter (10.0f), 0.01);
}
```

@ Test

```
public void test_Area_circle ()
{
    area obj = new area();
    assertEquals (314, obj.area-circle (10.0f), 0.01);
}
```

Result - Test passed 100.0 %.

Page No. 13

21/01/2023



JIET GROUP OF INSTITUTIONS

Student Name Roll No.:

Experiment No. 4 Date

Object - To test & run assertArrayEqual().

Resources used - Windows 7, JUnit, Netbeans

Code - Testarray.java

```
package kritika;
import java.util.*;
public class Testarray {
    public int[] numbers(int n)
    {
        int [ ] arr = {1,2,3,4,5};
        int [ ] test = new int [ ] {1,2,3,4,5};
        if (n == 0)
            return arr;
        else
            return test;
    }
}
```

TestarrayIT.java.

```
@Test
public void testSomeMethod()
{
    Testarray obj = new Testarray();
    assertEquals(0, obj.numbers(0));
    assertEquals(5, obj.numbers(1));
}
```

Result - Test passed 100.00 %;

JIET GROUP OF INSTITUTIONS



Student Name Roll No.

Experiment No. Date

Object - Create a test case on class to find whether a given no. is prime or not.

Resources Used - Windows 7, JUnit, Netbeans

Code - Prime.java

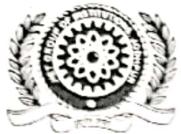
```
package kritika;
public class Prime {
    boolean isPrime(int n)
    { int flag = 0;
        for(int i=2; i<=n/2; i++)
        { if(n % i == 0)
            flag = 1;
        }
        if(flag == 1)
            return false;
        else
            return true;
    }
}
```

PrimeIT.java

```
@Test
public void TestIsPrime()
{
    Prime obj = new Prime();
    assertEquals(obj.isPrime(19));
}
```

Result - Test passed 100.00%.

JIET GROUP OF INSTITUTIONS



Student Name Roll No.

Experiment No. Date

Object - Create a Test case on class to find the given array is equal or not reverse of string.

Resources Used - Windows 7, JUnit, Netbeans.

Code - Palindromes.java.

```
package kritika;
public class Palindromes {
    static ispali (String s1)
    { String Builder sb = new String Builder (s1);
        sb. reverse ();
        return sb. toString ();
    }
}
```

PalindromesIT.java.

```
@Test
public void isTest Ispali ()
{
    Palindromes obj = new Palindromes ();
    assertEquals (obj. ispali ("baab"), "baab");
}
```

Result - Test passed 100.00%.

JIET GROUP OF INSTITUTIONS



Student Name Roll No.

Experiment No. Date

Object- Create a Test case to find sum of array.

Resources Used- windows 7, JUnit, Netbeans.

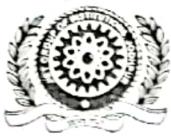
Code- sumArray.java.

```
package kritika
public class sumArray {
    int sumArray() {
        int num = 0
        int arr[] = {1, 2, 3, 4, 5};
        for (int i = 0; i < arr.length - 1; i++) {
            num = num + arr[i];
        }
        return num;
    }
}
```

sumArrayIT.java.

```
@Test
public void testSumArr() {
    SumArray obj = new SumArray();
    assertEquals(15, obj.sumArr());
}
```

Result- Test passed 100.00%.



JIET GROUP OF INSTITUTIONS

Student Name Roll No.

Experiment No. Date

Object - Create a test case on class to find fb series & put in array.

Resources Used - windows 7, JUnit, Netbeans

Code - Fbseries.java.

```
Package kritika;
public class Fbseries {
    int[] fibbo(int n)
    {
        int[] fibb = new int[n+2];
        fibb[0] = 0;
        fibb[1] = 1;
        for (int i=2; i<=n+1; i++)
            fibb[i] = fibb[i-1] + fibb[i-2];
        return fibb;
    }
}
```

FbseriesIT.java.

```
@Test
public void testfibbo()
{
    Fbseries obj = new Fbseries();
    int[] arr = {0, 1, 1, 2, 3, 5, 8};
    assertEquals(arr, obj.fibbo(5));
}
```

Result - Test passed 100.00 %.

28/02/2023

JIET GROUP OF INSTITUTIONS



Student Name Roll No.

Experiment No. 5 Date

Object - Create a test class to find arithmetic calculation

Resources used - Windows; Netbeans, JUnit.

Code - stvLabs.java

```
package krujika;
public class stvLabs{
    int add (int a, int b)
        return a+b;
    int sub (int a, int b)
        return a-b;
    int mul (int a, int b)
        return a*b;
    int div (int a, int b)
        return a/b;
}
```

~~stvLabsTest.java~~

```
public void testMain() {
    StvLabs obj = new StvLabs();
    assertEquals(20, obj.add(10, 10));
    assertEquals(20, obj.sub(30, 10));
    assertEquals(20, obj.mul(2, 10));
    assertEquals(20, obj.div(200, 10));
}
```

Result - Test passed 100.00%.



JIET GROUP OF INSTITUTIONS

Student Name Roll No.:

Experiment No. 6 Date

Object - Create a Test case on class to find largest element in an array.

Resources Used - Windows 7, JUnit, Netbeans

code - Large.java

```
package krullika;
public class Large {
    int [] arr = {10, 20, 30, 40, 50, 60};
    int high;
    int great()
    {
        for (int i=0; i<arr.length; i++)
        {
            if (i==0)
                high = arr[i];
            else
            {
                if (arr[i] > high)
                    high = arr[i];
            }
        }
        return high;
    }
}
```

LargeIT.java

```
@Test
```

```
public void testSomeMethod ()
```

```
    Large obj = new Large()
```

```
    assertEquals(60, obj.great());
```

```
}
```

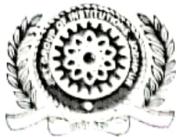
```
}
```

Result - Test passed 100.00%

Page No. 20

OP
21/03/22

JIET GROUP OF INSTITUTIONS



Student Name Roll No.

Experiment No. 7 Date

Object - Create a test case one class to sort an array.

Resource Used - JUnit, Windows 7, Netbeans.

Code - JK.java

```
package kavitika;
import java.util.*;
public class JK {
    int [] arrsort(int [] arr)
    {
        Arrays.sort(arr);
        return arr;
    }
}
```

JKIT.java

```
@ public void testArrSort()
{
    JK obj = new JK();
    int [] arr1 = {1, 2, 3, 4, 5};
    int [] arr2 = {5, 3, 4, 2, 1};
    assertEquals(arr1, obj.arrsort(arr2));
}
```

Result - Test passed 100.00%.

JIET GROUP OF INSTITUTIONS



Student Name Roll No.

Experiment No. 8. Date

Object - Create a test case on a class to find palindrome of no.

Resource Used - JUnit, Netbeans, Windows 7

Code - pali.java.

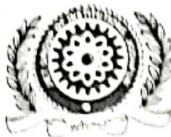
```
package krishika;
public class pali {
    boolean palin (int a)
    {
        int n = 0, r, temp = a;
        while (a != 0)
        {
            r = a % 10;
            n = n * 10 + r;
            a = a / 10;
        }
        if (temp == n)
            return true;
        else
            return false;
    }
}
```

paliTest.java.

```
@Test
public void TestPalin()
{
    pali obj = new pali();
    assertEquals(true, obj.palin(100));
}
```

Result - Test passed 100.00%.

JIET GROUP OF INSTITUTIONS



Student Name Roll No.

Experiment No. 9 Date

Objective -

JMeter - Performance testing by JMeter tool.

Theory -

JMeter - It works by simulating visitors to your application/service, by allowing users to create & send HTTP request to server. The server response data is then collected, & the statistical data is displayed visually for user in the form of charts & reports.

Steps to perform Load testing with JMeter

1. Verify system Requirements -

2. Download JMeter Binaries or Source Code -

Once you verify requirements, download the latest JMeter source file from jmeter.apache.org / download-jmeter.cgi

3. JMeter Installation process - Extract the file on desired location.

4. In the extracted folder find bin folder & open it. In bin folder, run JMeter.exe.

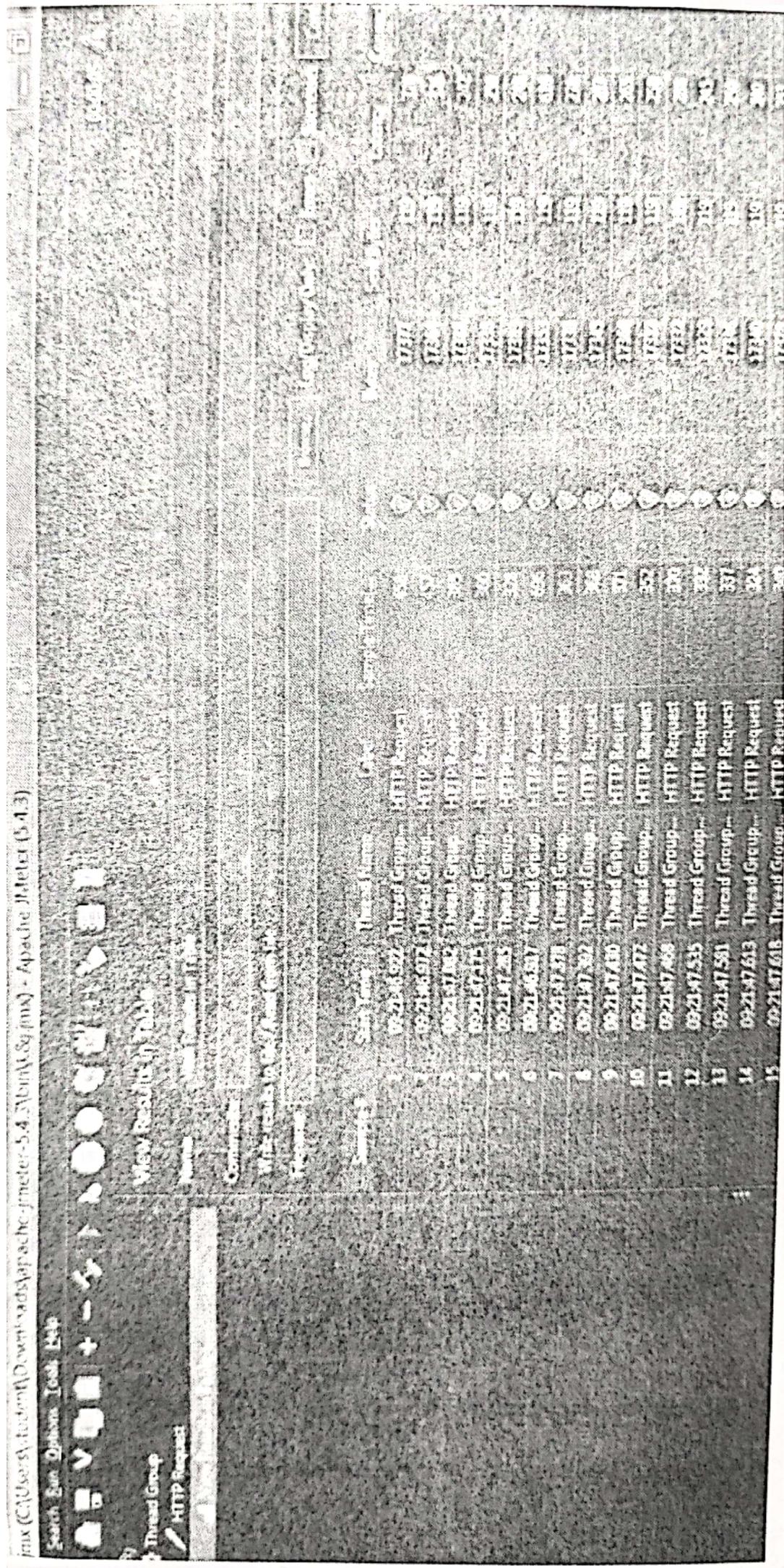
4. JMeter User Interface - Test Plan Window opens, which include access to the menu & Toolbar. Menu Tool Bar provides quicker access to some of the tools used during the load test creation & configuration process. The main section, ~~and Editor sectn~~, where we can see & configure Test plan elements & field of load test.

5. Create a Load Test plan -

(i) Give name to Test plan & save it.

(ii) In the left navigation window, right click on name of your test plan.

(iii) Click Add > Threads (Users) > Thread group.



JIET GROUP OF INSTITUTIONS



Student Name Roll No.

Experiment No. Date

6. Create Thread Group - Within the Thread group dialogue box, you can set & adjust multiple Thread properties, such as No. of threads (users), Ramp up period (in sec) & Loop count (how many Test iteratn), as well as additional actⁿ like delays, test start & stop time & actⁿ to take after sampler error.

7. Configured sampler - To add sampler, right click on Thread group in navigatⁿ window. Click on add > sampler > HTTP Request.

Give the server name/ IP which is to be tested in the test plan.
Eg → www.google.com

8. Configure Listener - In the left navigatⁿ pane, right click on HTTP request & click on Add > Listener > View Results in Table.

9. Executing the Test Plan - In the main Toolbar, click on 'start' button. The results will automatically get appended in the table.

Note → We can view results in different visual forms by adding different types of listeners. Eg - Response Time graph etc.

10. Review the load test results

Result - Performance testing done successfully.

JIET GROUP OF INSTITUTIONS



Student Name Roll No.

Experiment No. IO Date

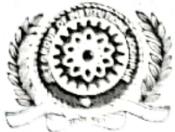
Objective - Automatⁿ Testing by Selenium.

Theory - Selenium IDE - It is a browser add-on that makes it easy to record & play ^{back} that test in browser. IDE is implemented as web browser extension. It allows you to record the test & edit the test script, move the commands around & debug. A Tester can author tests even without learning a test script language.

Process to automatⁿ testing by selenium -

1. Installing - Open Chrome, search selenium IDE. Find page that leads to chrome.google.com/... click on the add extension button.
2. Recording Test - Go to plugins in chrome & click on selenium IDE
 - In the selenium window, click on "record a new test in a new project"
 - Provide a name for your project.
 - Specify a valid base URL for your project. For eg - <http://thebestsite.phptravels.com/demo>
 - Click on start Recording button.
3. Performing tasks to record in test - You can perform multiple actⁿ which selenium will record such as enter text, login, logout etc.
4. Verify & Validate - After performing actⁿ stop recording. Provide a name to test. You'll be able to see the commands & their respective target locatⁿ, add values if any in the table. You can verify the test by clicking on run current test & the

JIET GROUP OF INSTITUTIONS



Student Name Roll No.

Experiment No. Date

commands will execute one by one automatically.

Result - Automat* testing by selenium IDE done successfully.

89
20/4/14

JIET GROUP OF INSTITUTIONS



Student Name Roll No.

Experiment No. Date

Objective - Manual Testing by Selenium

Theory -

Testing- Procedure -

→ We have to write commands and their target locatⁿ manually & the Test will run accordingly.

→ Give a name to test.

→ Click set command as command, target, value as below -

+	<u>Command</u>	<u>Target</u>	<u>Value</u>
a.	open	https://www.google.com	
b.	set window size	1024x620	
c.	click	name=g	
d.	Type	name=g	selenium Webdriver
e.	send keys	name=g	KEY-ENTER
f.	click	css#kao>.g:nth-child(1).LC2016	
g.	close		

→ Set Test execution speed to slow & run the Test & verify results.

Result - Manual Testing by selenium done successfully.

JIET GROUP OF INSTITUTIONS

INDEX

Name of the laboratory ... S.T.V. Lab Code ... B.G.S.Y-22

S.No.	Name of Experiment	Date of		Page No.	Remarks	Signature of Teacher with date
		Allotment	Completion			
1.	Assignment - 2 (Assert func ⁿ)	14/02	14/02	6-12		SD 21/02/22
2.	Introduct	14/02	14/02	1-5		SD 21/02/22
3.	Assignment - 3 (Assert func ⁿ)	21/02	21/02	13.		SD 21/02/22
4.	Assignment - 4 (Assert array Eq)	28/02	28/02	14-18		SD 21/02/22
5.	Assignment - 5 (assert Test)	28/02	28/02			SD 28/02/22
6.	Assignment - 6 (Reverse of string Test)	21/03	21/03	20.		SD 21/03/22
7.	Assignment - 7 (Test of sum of array) arithmetic calculat ⁿ , find largest element, sort an array	23/03	23/03.	21		SD 21/03/22
8.	Assignment - 8 (palindrome of no.)	23/03	23/03	22.		
9.	Inter testing	23/03	18/04	23-24 25		
10.	Automation testing	18/04	20/04	25-26		SD 20/04/22
11.	Manual testing	20/4	20/04	27		