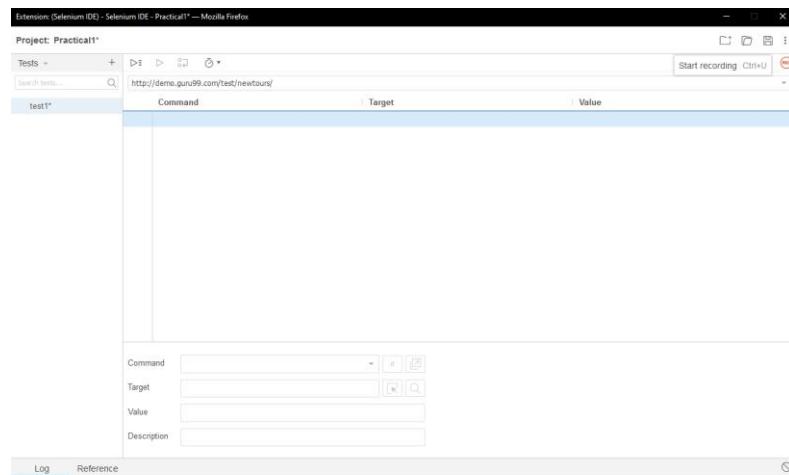


Practical No. 1

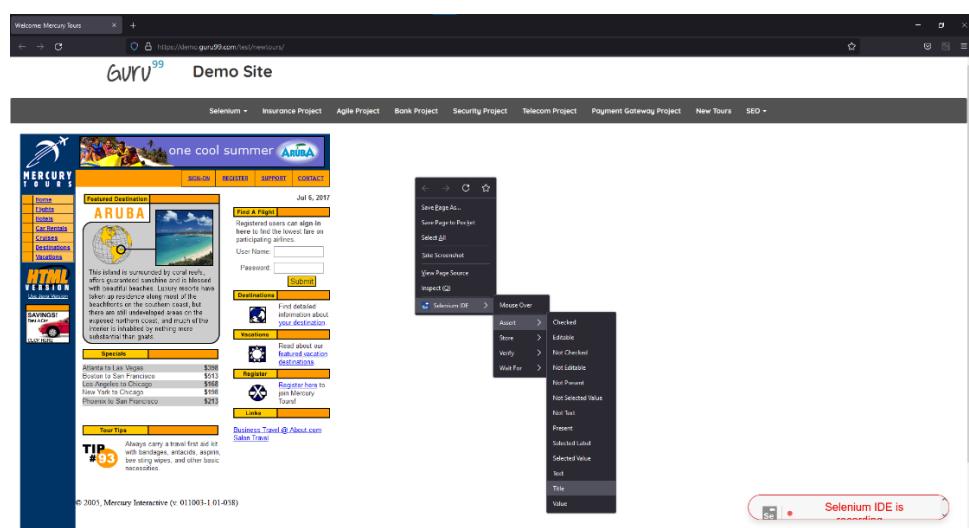
Aim:- Install Selenium IDE; Write a test suite containing minimum 4 test cases for different formats.

For demo.guru99.com →

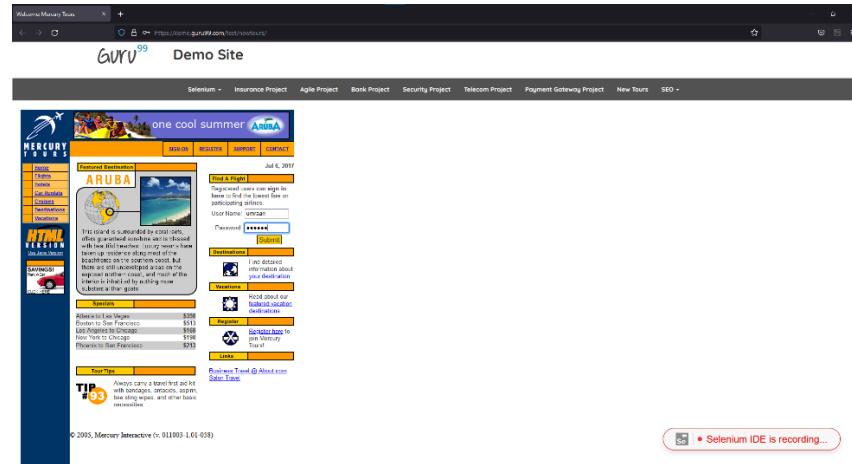
Step 1: Open Mozilla Firefox and search for <http://demo.guru99.com/test/newtours/>. Copy and paste the same URL into Selenium IDE, start recording subsequently.



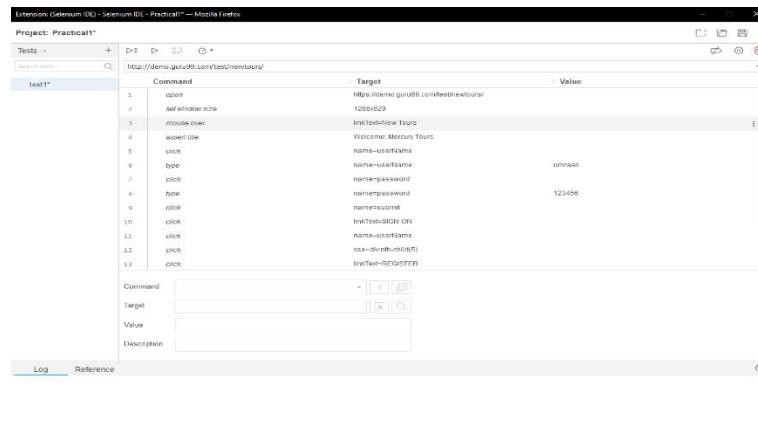
Step 2: Right click on the web page and cursor over Selenium IDE. Then, select Assert Title from the side bar.



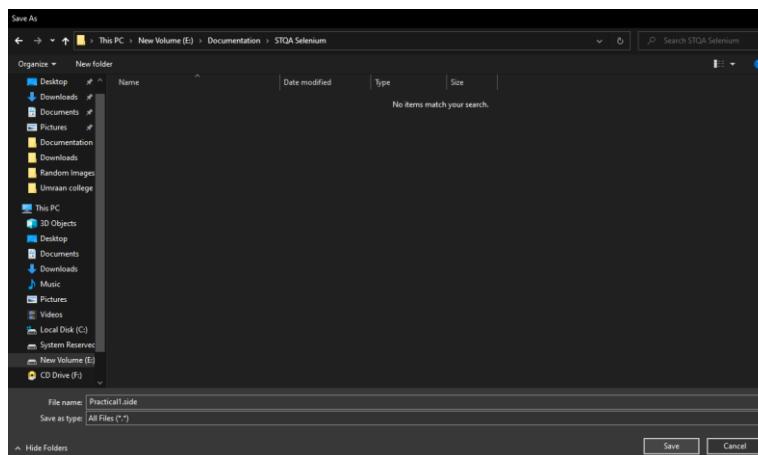
Step 3: Type anything into the user name and password blocks. Click Sign-In after entering.



Step 4: Abreast, you will see the commands in Selenium IDE which you had performed on website.



Step 5: Save your test case with a name followed by the `.side` extension.



Step 6: Go to the Selenium IDE and click onto Run current test then you will see the ‘test case has completed successfully’ in the Log section.

Index	Command	Target	Value
1	get	123456789	
2	mouse over	IntertourHome	
3	assert title	Welcome Mercury Tours	
4	click	name=username	
5	type	name=username	untrap
6	click	name=password	123456
7	type	name=password	123456
8	click	name=submit	
9	click	IntertourSIGNON	
10	click	name=username	
11	type	name=username	css-with-child()
12	click	css-with-child()	
13	click	IntertourREGISTER	
14	click	IntertourSIGNON	

Log
10. click on IntertourREGISTER OK
11. click on name=username OK
12. click on css-with-child() OK
13. click on IntertourREGISTER OK
14. click on IntertourSIGNON OK

"test1" completed successfully

For Gmail→

Step 1: Go to www.google.com and search for Gmail.

Step 2: Click on Gmail icon.

Step 3: Enter some ID and Password.

Step 4: Stop recording and then click on play.

For Yahoo→

Step 1: Go to www.yahoo.com and start recording.

Step 2: Perform some actions and then click on play button.

For Selenium→

Step 1: Go to <https://www.seleniumhq.org> and start recording.

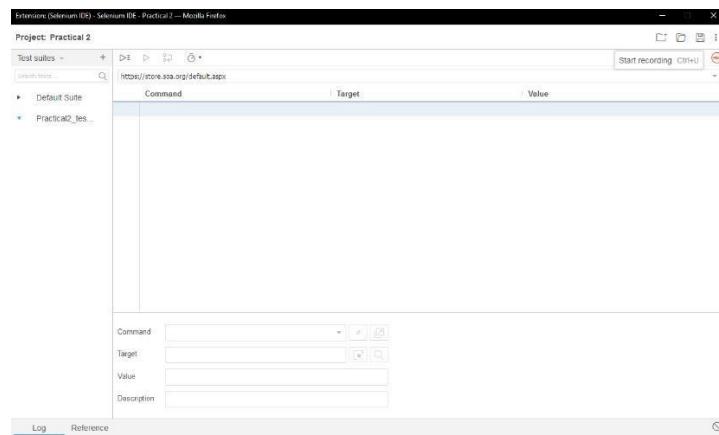
Step 2: Perform some actions and then click on play button.

Practical No. 2

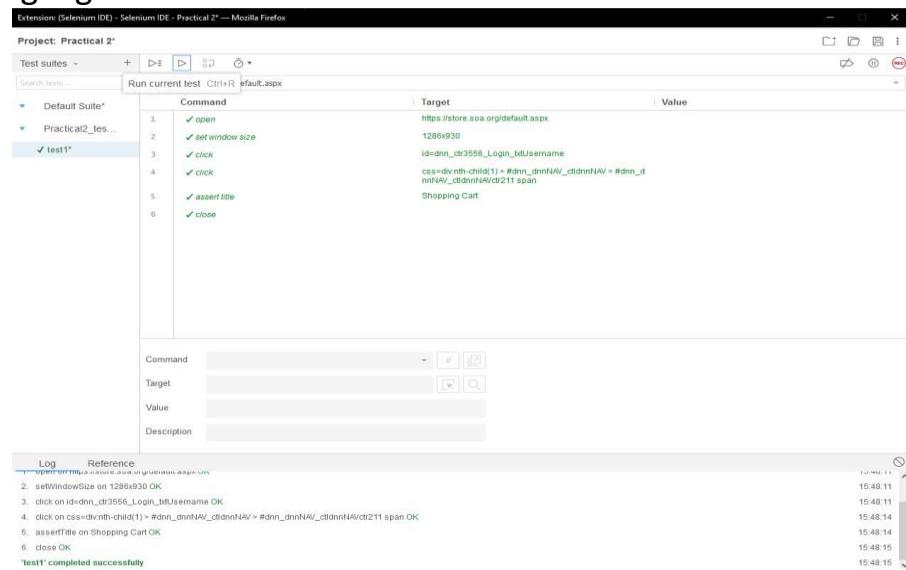
Aim: To conduct a test suite for any 2 websites.

A. For www.soastastore.com

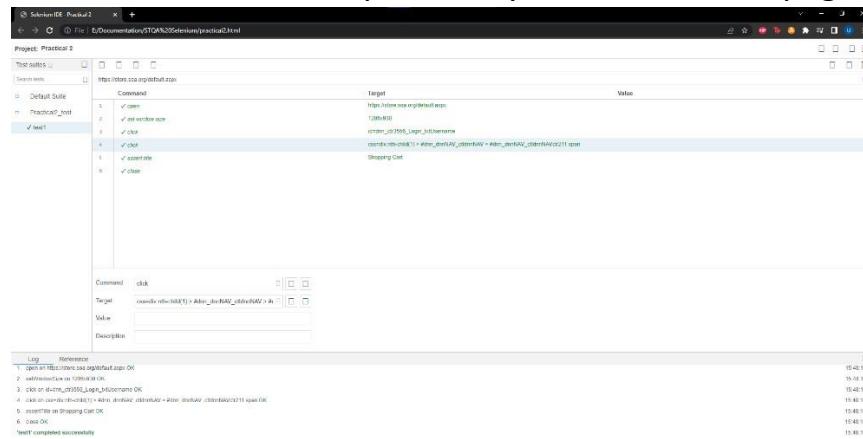
Step 1: Open the website and open selenium IDE. Copy the URL and paste it in the given space for further test. Then save & run the test to see log files.



Step 2: After every successful run it gives message “completed successfully” with green text highlighted.

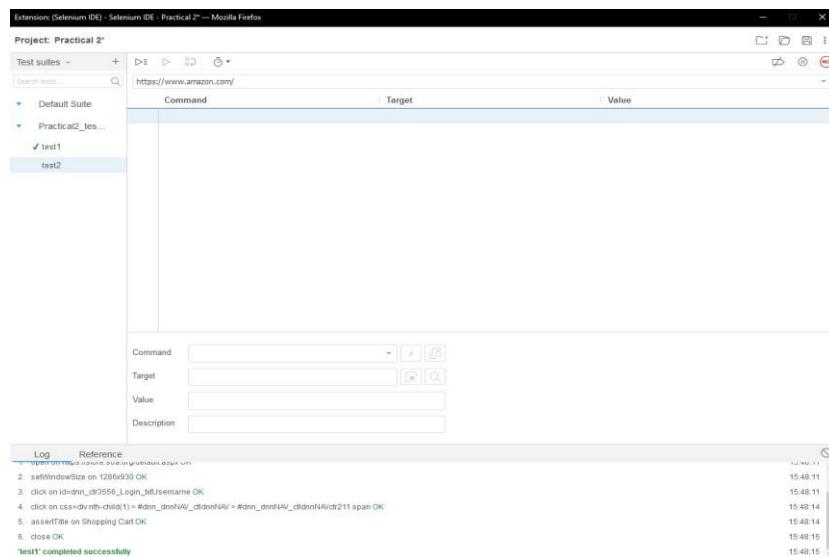


Step 3: Save it in the HTML file and open it on your browser webpage.



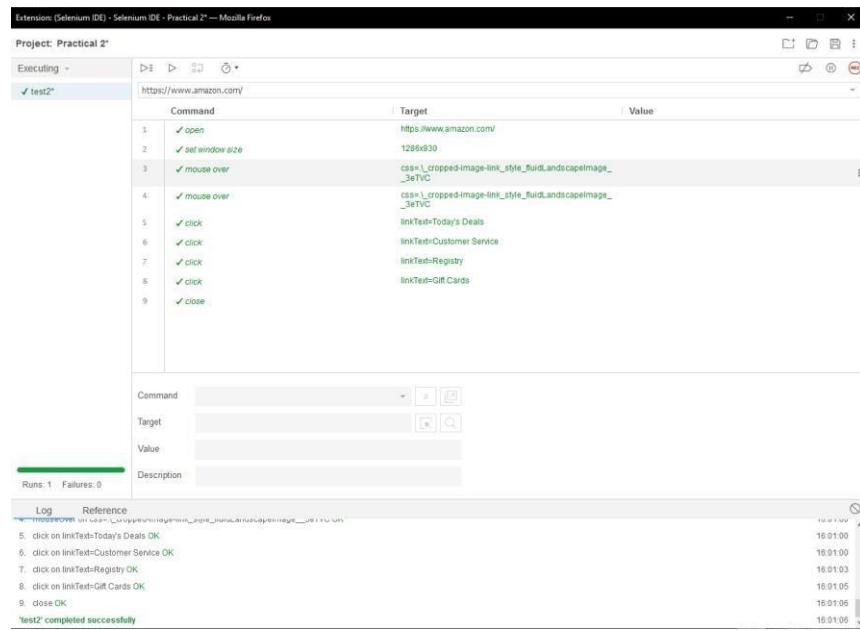
B. For www.amazon.com

Step1: Open the website and open selenium IDE. Copy the URL and paste it in the given space for further test. Then save & run the test to see log files.

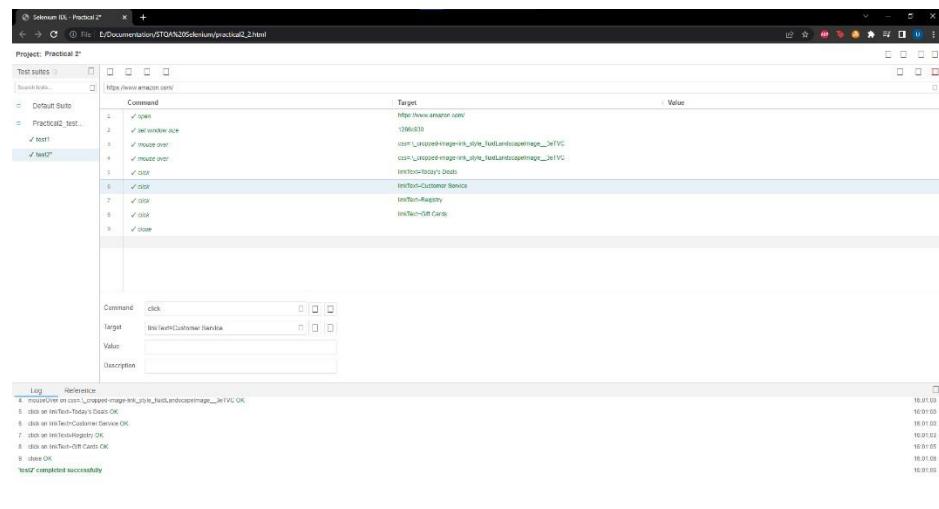


Step 2: After saving the project, click on play and see how commands execute...

Step 3: After successfully completing you will see message “completed successfully” with green text highlighted with log.



Step 4: Save it in the HTML file and open it on your browser webpage.

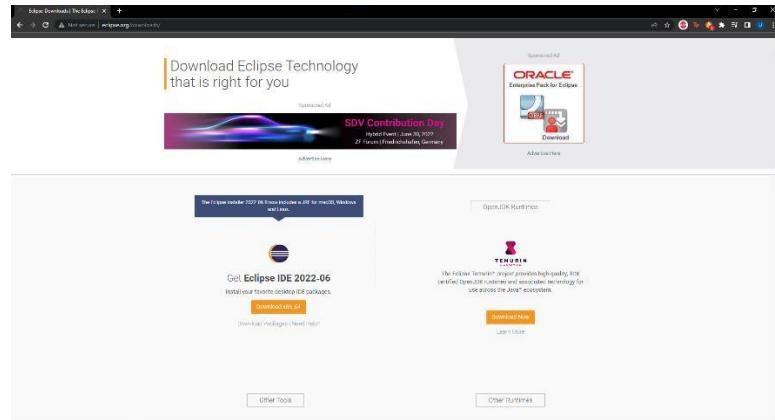


Practical No. 3

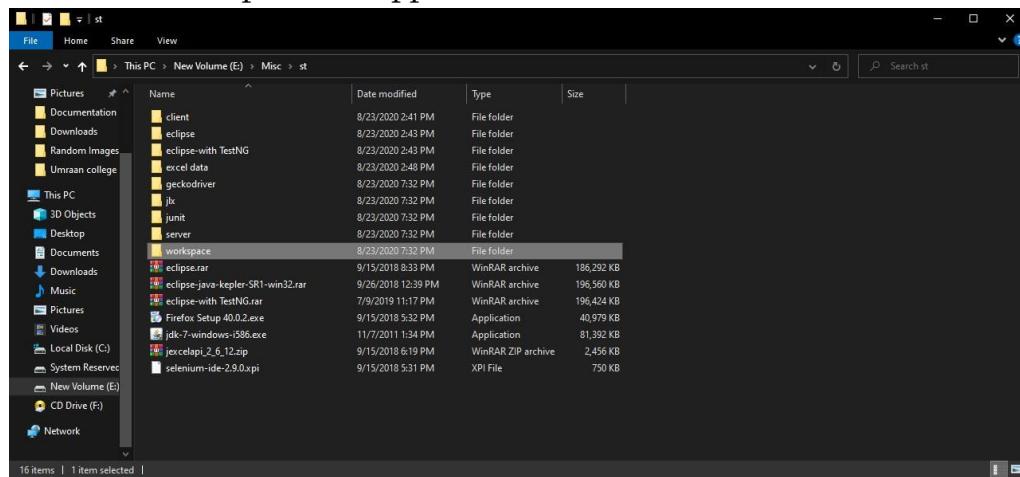
AIM: Install Selenium server and demonstrate it using a script in Java/PHP.

Download Eclipse

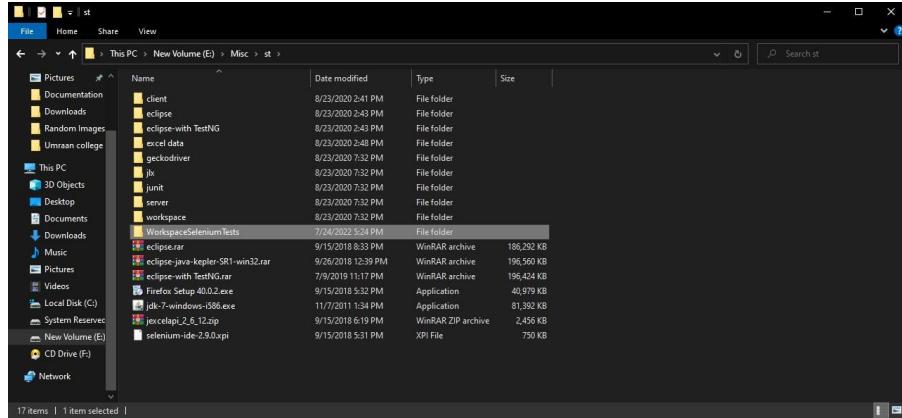
1. Go to URL – <http://www.eclipse.org/downloads/>
2. Select Eclipse IDE for Java Developers (Click on Windows 64 bit platform)



3. Click on OK button and save to a local drive (i.e. C: or D:, etc)
4. Unzip the downloaded zip file and rename that to Eclipse
5. Create one more folder “Eclipse-Workspace” (i.e. C:Eclipse-Workspace)in the same drive where Eclipse is unzipped and renamed.

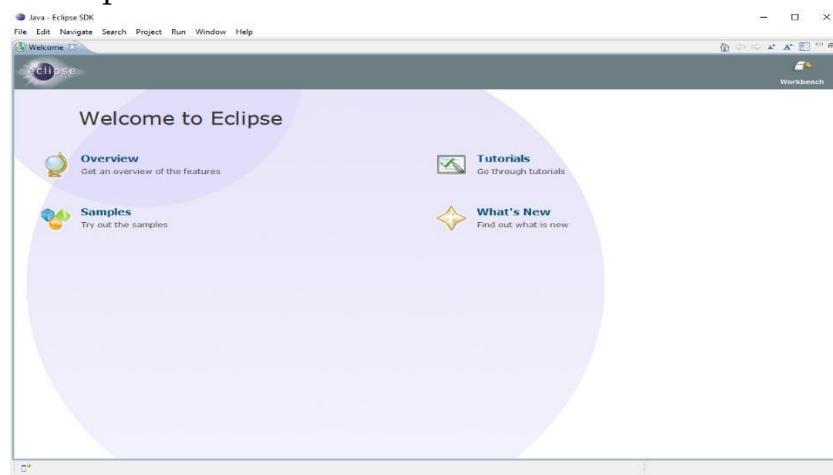


6. Create Eclipse desktop shortcut (go to C:Eclipse folder → right click Eclipse.exe and then click on “desktop create shortcut”).
1. Now we need to create a workspace folder → C:Eclipse-WorkspaceSeleniumTests



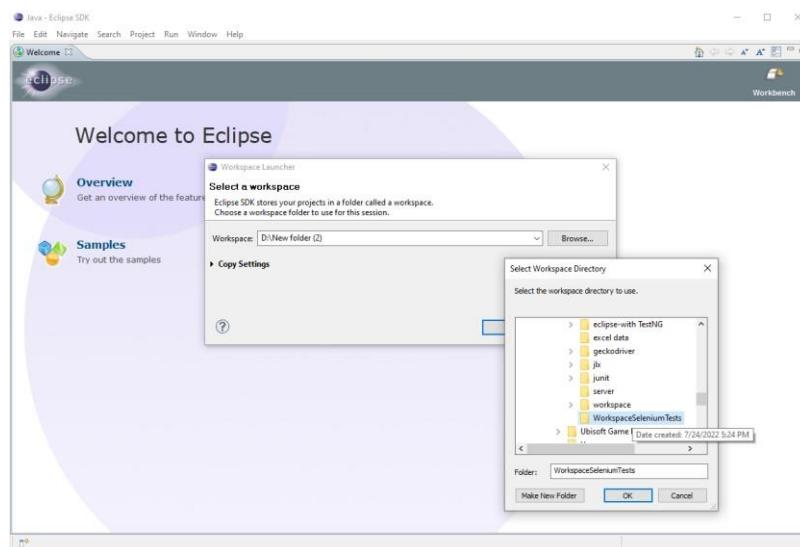
2. Double click on “Eclipse shortcut on Desktop”

3. This opens the Eclipse



4. Close Eclipse welcome screen

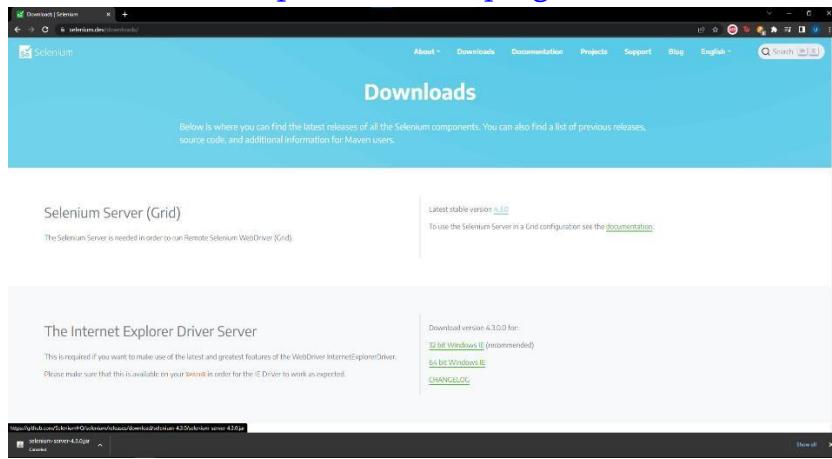
5. Click File menu → Launch Workspace → other 6. Now Select the C:Eclipse-WorkspaceSeleniumTests folder.



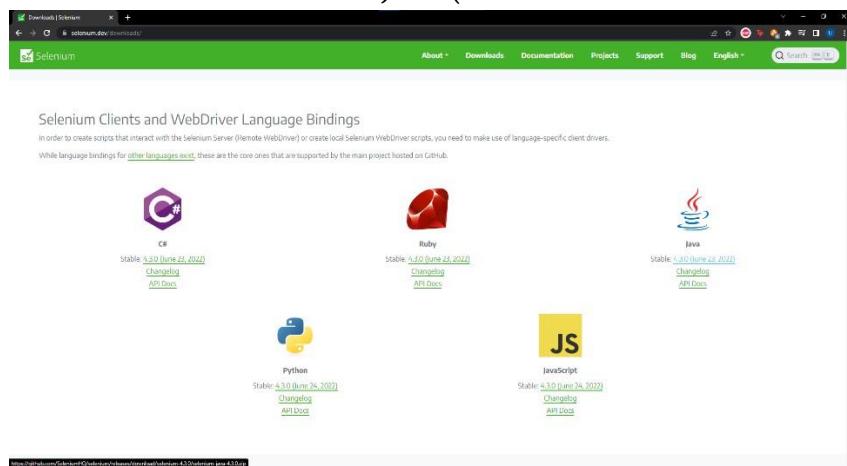
We have finished setting up the eclipse.

Now, we need to download Selenium RC server / client driver and configure that to Eclipse

1. Download Selenium server: <http://seleniumhq.org/download/>



2. Download Selenium Client driver for Java (from Selenium Client Drivers section)

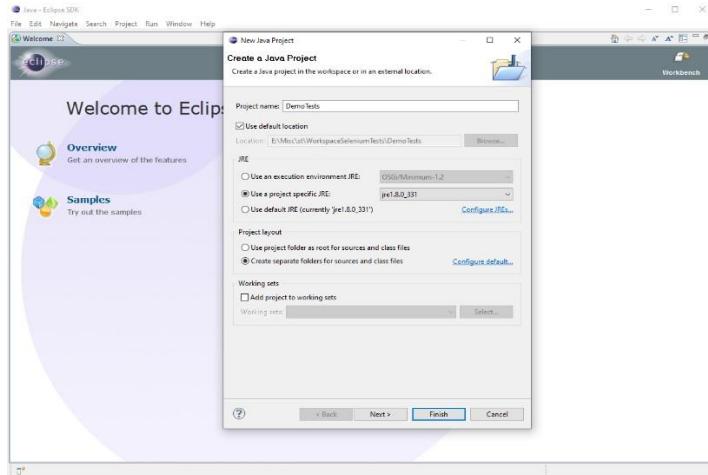


3. Create “Selenium” folder in C: drive and copy the Selenium-server.jar as well as unzip the Selenium Client driver (C:Selenium)

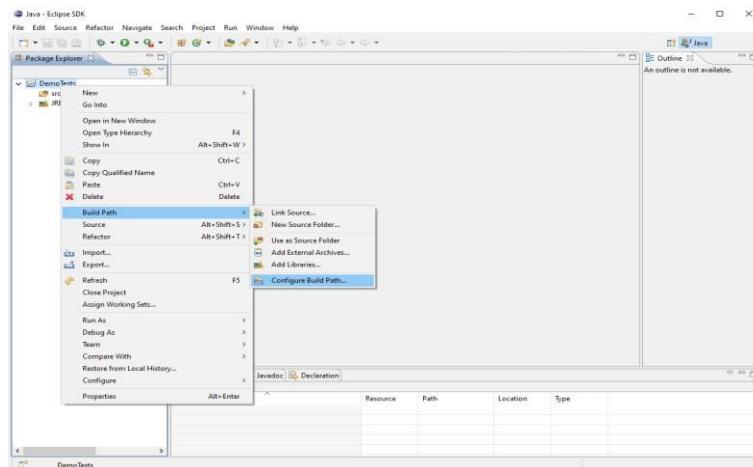
Downloading and unzipping the files into a folder is done.

We need to configure the appropriate Selenium Client driver Jar file to the Eclipse.

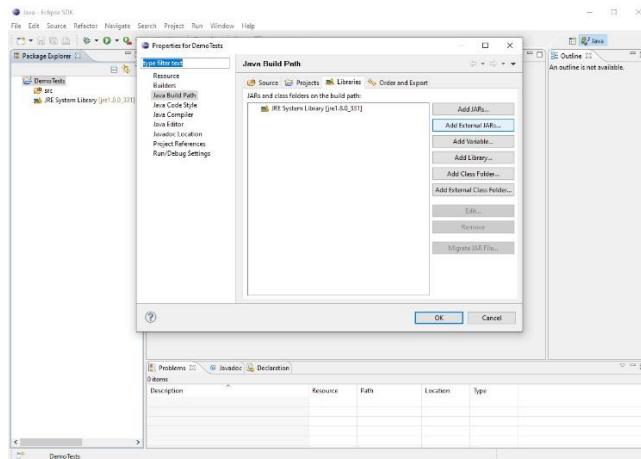
1. Go to Eclipse → Click File → New → Project (from various options need to select just “project”)
2. In Select Wizard → Click Java → “Java Project”
3. Give the project name (e.g. DemoTests)



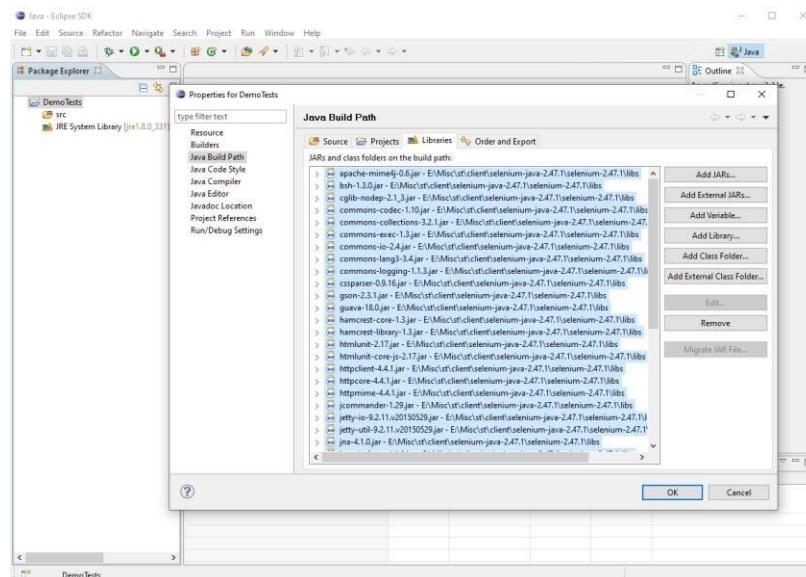
4. Click Finish – Click Yes
5. Now we are done with creation of project and need to configure the Selenium Client driver to this Project
6. Right Click “DemoTests” project
7. Click “Java Build Path”



8. Click Libraries tab
9. Click “Add External JARs” button



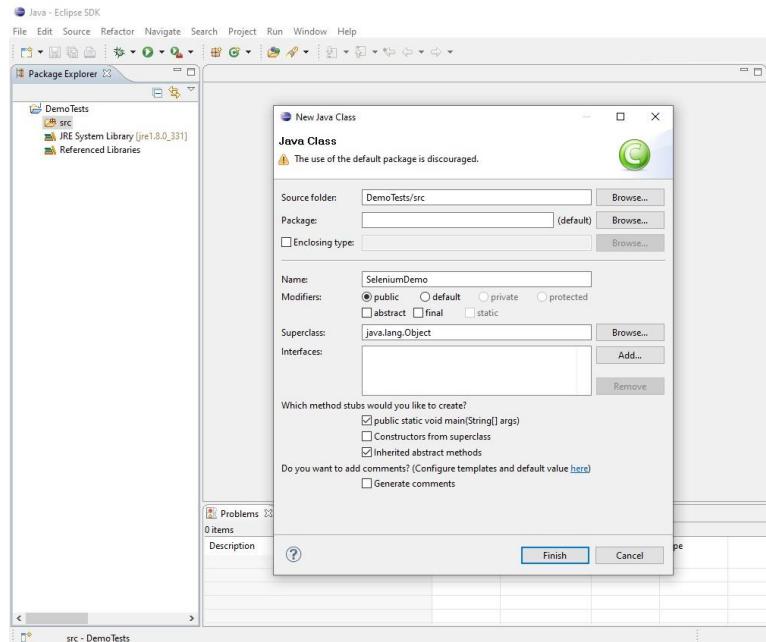
10. Select “Selenium Client Drivers” unzipped in C:Selenium folder (Selenium Server JAR file should not be added)



11. Click OK

12. Referenced libraries → contains both the Selenium Client driver jar files.

13. Create a new class file as “SeleniumDemo” in the “DemoTest” by right click on src folder.

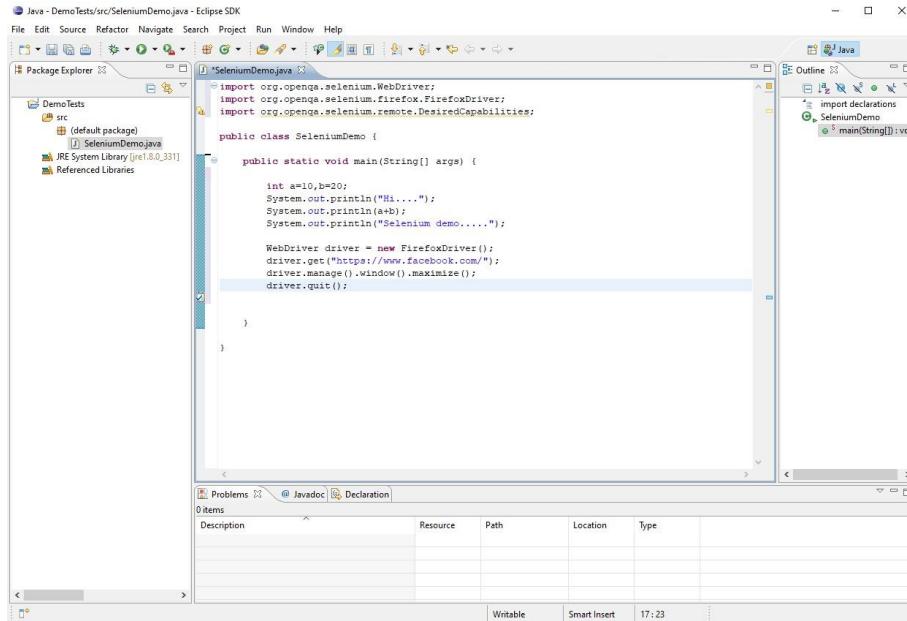


14. Copy the below code in the class file:-

```
import org.openqa.selenium.WebDriver; import
org.openqa.selenium.firefox.FirefoxDriver; import
org.openqa.selenium.remote.DesiredCapabilities; public
class Test
{
    public static void main(String args[])
    {
        int a=10,b=20;
        System.out.println("Hi....");
        System.out.println(a+b);
        System.out.println("Selenium demo.....");
```

```
        WebDriver driver = new FirefoxDriver();
        driver.get("https://www.facebook.com/");
        driver.manage().window().maximize();
        driver.quit();
```

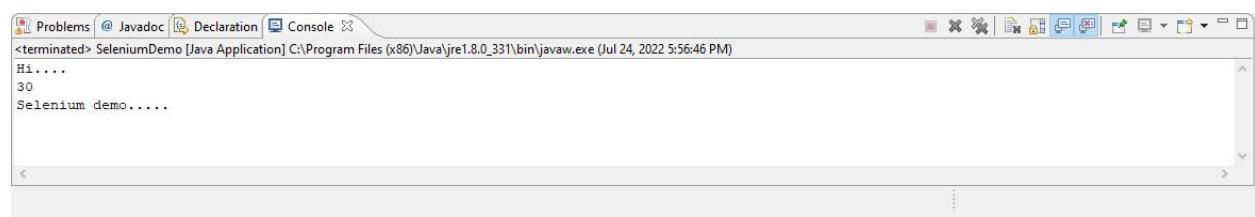
```
}
```



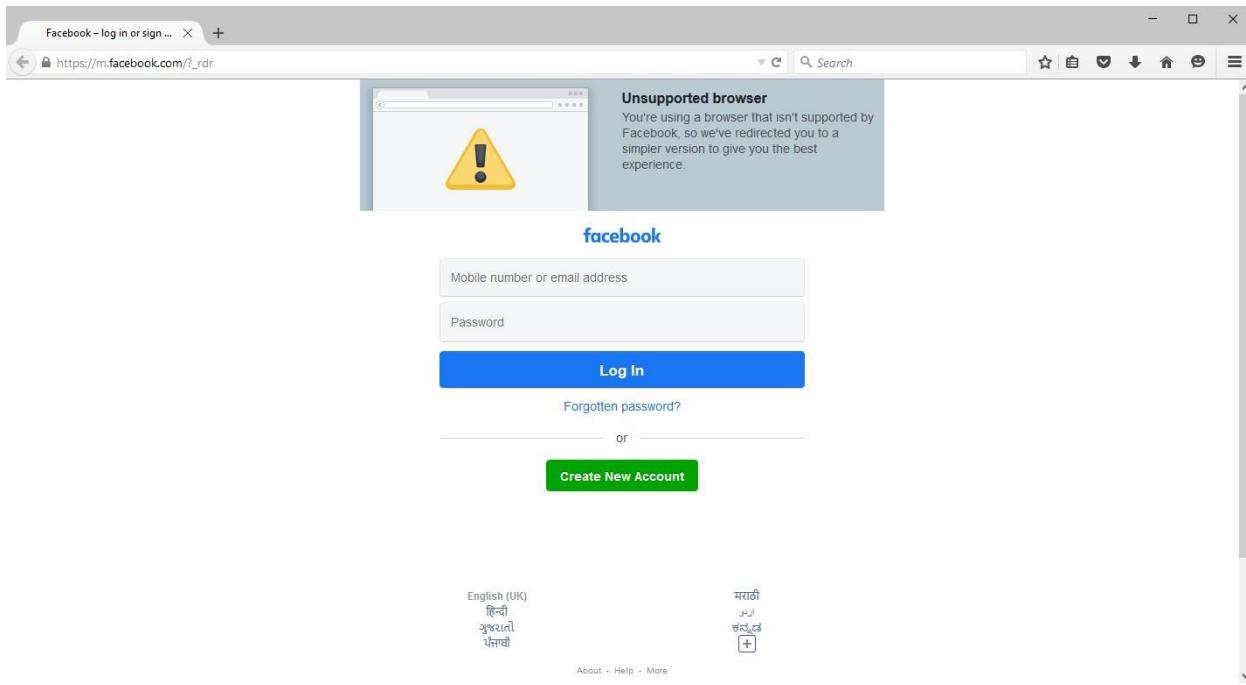
```
Java - DemoTests/src/SeleniumDemo.java - Eclipse SDK
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
DemoTests
src
  (default package)
    SeleniumDemo.java
JRE System Library [jre1.8.0_331]
Referenced Libraries
SeleniumDemo.java
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.remote.DesiredCapabilities;
public class SeleniumDemo {
    public static void main(String[] args) {
        int a=10,b=20;
        System.out.println("Hi....");
        System.out.println(a+b);
        System.out.println("Selenium demo.....");
        WebDriver driver = new FirefoxDriver();
        driver.get("https://www.facebook.com/");
        driver.manage().window().maximize();
        driver.quit();
    }
}
```

(If Firefox is unable to open, download [geckodriver](#) from the website, <https://github.com/mozilla/geckodriver/releases> according to the operating system, unzip the folder and keep the file along with gecko folder in the same folder wherever selenium jar files folder is there)

Output: -



```
Problems @ Javadoc Declaration Console
<terminated> SeleniumDemo [Java Application] C:\Program Files (x86)\Java\jre1.8.0_331\bin\javaw.exe (Jul 24, 2022 5:56:46 PM)
Hi....
30
Selenium demo.....
```



Pract 3

St client  selenium-java-2.47.1  ADD librarys selenium-java.2.47.1.jar , selenium-java.2.47.1-srcs.jar + all librarys in libs folder

code

```
package pk3;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;

public class p3 {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int a = 10, b = 20;
        System.out.println("Hi...");
        System.out.println(a+b);
        System.out.println("Selenium demo ...");
        WebDriver driver = new FirefoxDriver();
        driver.get("https://www.facebook.com/");
        driver.manage().window().maximize();
        driver.quit();
    }
}
```

Prac 4

St junit  junit-4.10.jar→ junit-4.10

TestJUnit.java

```
package p4;
```

```
import org.junit.Test;  
import static org.junit.Assert.assertEquals;  
  
public class TestJUnit {  
  
    @Test  
  
    public void testSetup(){  
  
        String str = "I am Done with Junit Setup";  
  
        assertEquals("I am Done with Junit Setup",str);  
  
    }  
  
}
```

TestRunner.java

```
package p4;  
  
import org.junit.runner.JUnitCore;  
import org.junit.runner.Result;  
import org.junit.runner.notification.Failure;  
  
import p4.TestJUnit;  
  
public class TestRunner {  
  
    public static void main(String[] args) {  
  
        // TODO Auto-generated method stub  
  
        Result result = JUnitCore.runClasses(TestJUnit.class);  
  
        for(Failure failure:result.getFailures()){  
  
            System.out.println(failure.toString());  
  
        }  
  
        System.out.println("Result=="+result.wasSuccessful());  
  
    }  
  
}
```

Pract 5

St@ testNG @eclipse

St@Jxl @jexcelapi @jxl jar

Right click on java project

code

```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import jxl.Sheet;
import jxl.Workbook;
import jxl.write.Label;
import jxl.write.WritableSheet;
import jxl.write.WritableWorkbook;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.Test;

public class c5 {
    @BeforeClass
    public void f1()
    {}
    @Test
        public void testImportexport1() throws Exception {
            FileInputStream fi = new
FileInputStream("C:\\st\\sampledata.xls");
            Workbook w = Workbook.getWorkbook(fi);
            Sheet s = w.getSheet(0);
            String a[][] = new String[s.getRows()][s.getColumns()];
            FileOutputStream fo = new
FileOutputStream("C:\\st\\result.xls");
            WritableWorkbook wwb = Workbook.createWorkbook(fo);
            WritableSheet ws = wwb.createSheet("result1", 0);
            for (int i = 0; i<s.getRows(); i++)
            {
                for (int j = 0; j <s.getColumns(); j++)
                {
                    a[i][j]=s.getCell(j,i).getContents();
                    Label l2=new Label(j,i,a[i][j]);
                    ws.addCell(l2);
                    Label l1=new Label(6,0,"Results");
                    ws.addCell(l1);
                }
            }
            for (int i = 1; i<s.getRows(); i++)
            {
                for (int j = 2; j <s.getColumns(); j++)
                {
                    a[i][j]=s.getCell(j,i).getContents();
                    int x=Integer.parseInt(a[i][j]);
                    if(x>35)
                    {
                        Label l1=new Label(6,i,"Pass");
                        ws.addCell(l1);
                    }
                    else
                    {
                        Label l1= new Label(6,i,"Fail");
                        ws.addCell(l1);
                        break;
                    }
                }
            }
        }
}
```

```
        }
        wwb.write();
        wwb.close();
    }
}
```

Prac 6

St² testNG ²eclipse

St²jlx ²jexcelapi ²jxl jar

Right click on java project ² properties ² Libraries ² Add library ² TestNG

```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import jxl.Sheet;
import jxl.Workbook;
import jxl.write.Label;
import jxl.write.WritableSheet;
import jxl.write.WritableWorkbook;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.Test;

public class c6 {
    @BeforeClass
    public void f1(){}
    @Test
    public void testImportexport1()throws Exception
    {
        FileInputStream fi=new FileInputStream("C:\\st\\sampledata.xls");
        Workbook w=Workbook.getWorkbook(fi);
        Sheet s=w.getSheet(0);
        String a[][]=new String[s.getRows()][s.getColumns()];
        FileOutputStream fo=new FileOutputStream("C:\\st\\result.xls");
        WritableWorkbook wwb=Workbook.createWorkbook(fo);
```

```

WritableSheet ws=wwb.createSheet("result1",0);

int c=0;

for(int i=0;i<s.getRows();i++)
{
    for(int j=0;j<s.getColumns();j++)
    {
        if(i>=1)
        {
            String b=new String();
            b=s.getCell(3,i).getContents();
            int x=Integer.parseInt(b);
            if(x<60)
            {
                c++;
                break;
            }
        }
        a[i][j]=s.getCell(j,i).getContents();
        Label l2=new Label(j,i-c,a[i][j]);
        ws.addCell(l2);
    }
    wwb.write();
    wwb.close();
}
}

```

Prac 7

St client  selenium-java-2.47.1  ADD librarys selenium-java.2.47.1.jar , selenium-java.2.47.1-srcs.jar + all librarys in libs folder

```

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;

public class C7 {

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        WebDriver driver=new FirefoxDriver();
        driver.get("http://www.google.com");
        java.util.List<WebElement> links=driver.findElements(By.tagName("a"));
        System.out.println("Total Links are"+links.size());
    }
}

}

```

prac 8

St client  selenium-java-2.47.1  ADD librarys selenium-java.2.47.1.jar , selenium-java.2.47.1-srcs.jar + all librarys in libs folder

```

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.Select;
public class C8 {

```

```

/**
 * @param args
 */
public static void main(String[] args) {
    // TODO Auto-generated method stub
    WebDriver driver=new FirefoxDriver();
    driver.get("https://www.facebook.com/reg/");
    Select se = new Select(driver.findElement(By.id("month")));
    java.util.List<WebElement> mylist=se.getOptions();
    mylist.size();
    System.out.println("Number Of Items="+mylist.size());
}

}

```

Prac 9

St² client ↳ selenium-java-2.47.1 ↳ ADD librarys selenium-java.2.47.1.jar , selenium-java.2.47.1-srcs.jar + all librarys in libs folder

```

package pk9;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
public class c9 {

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        WebDriver driver=new FirefoxDriver();
        driver.get("C:\\st\\Combo.html");
        int radiochk=0,checkboxchk=0;
        int radiounchk=0,checkboxunchk=0;
        java.util.List <WebElement>
els=driver.findElements(By.xpath("//input[@type='radio']"));
        for(WebElement el:els)

```

```

        {
            if(el.isSelected())
            {
                radiochk++;
            }
            else
            {
                radiounchk++;
            }
        }
        System.out.println("Radio Buttons");
        System.out.println("Total Checked items"+ radiochk);
        System.out.println("Total unChecked items"+ radiounchk);

    java.util.List<WebElement>ebox=driver.findElements(By.xpath("//input[@type='checkbox']"));
    for(WebElement el:ebox)
    {
        if(el.isSelected())
        {
            checkboxchk++;
        }
        else
        {
            checkboxunchk++;
        }
    }
    System.out.println("Checkboxes");
    System.out.println("Total Checked items"+ checkboxchk);
    System.out.println("Total unChecked items"+ checkboxunchk);
}
}

```

Combo.html

```

<!DOCTYPE html>

<html>
<body>
<form>
<h2>Text Input</h2>
First Name:<br>
<input type="text" name="Firstname">
</br>
Last Name:<br>
<input type="text" name="lastname">

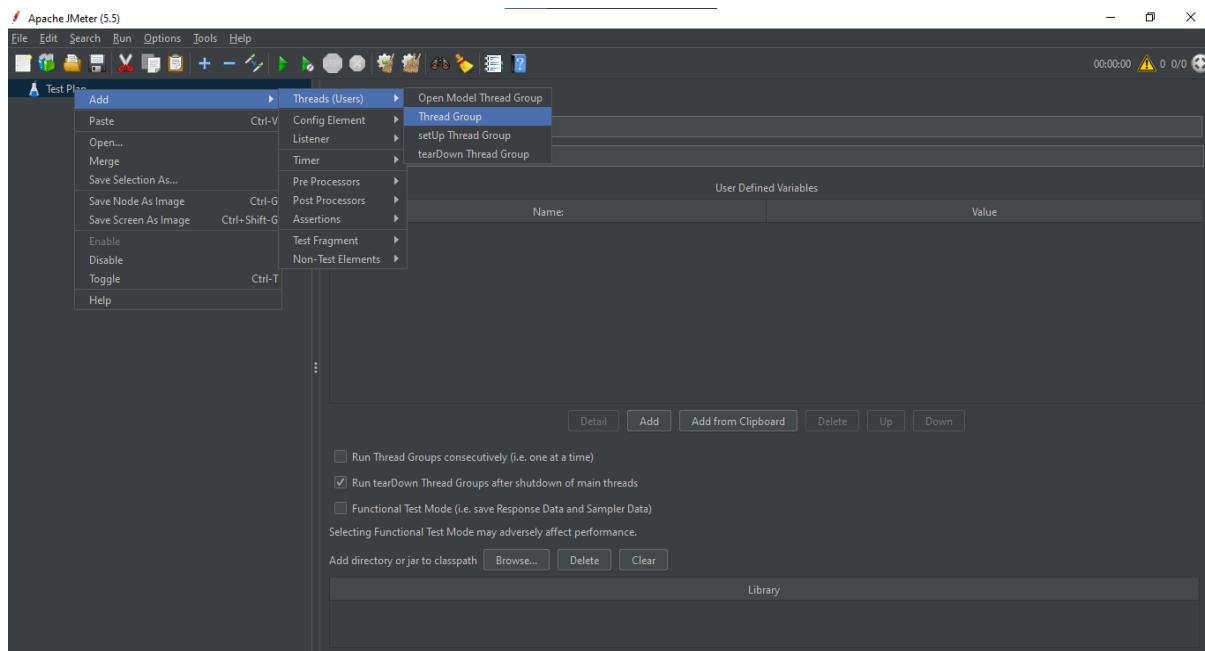
```

```
</br>
<h2>Select Gender</h2>
<input type="radio" name="gender" value="male" checked>Male<br>
<input type="radio" name="gender" value="female">Female<br>
<input type="radio" name="gender" value="others">Others<br>
<h2>Select Languages Known</h2>
<input type="checkbox" name="lang" value="Java" checked="checked">Java<br>
<input type="checkbox" name="lang" value="Php">Php<br>
<input type="checkbox" name="lang" value="ASP.net">.Net<br>
<input type="checkbox" name="lang" value="Python" checked="checked">Python<br>
<input type="submit" value="submit"><br>
</form>
</body>
</html>
```

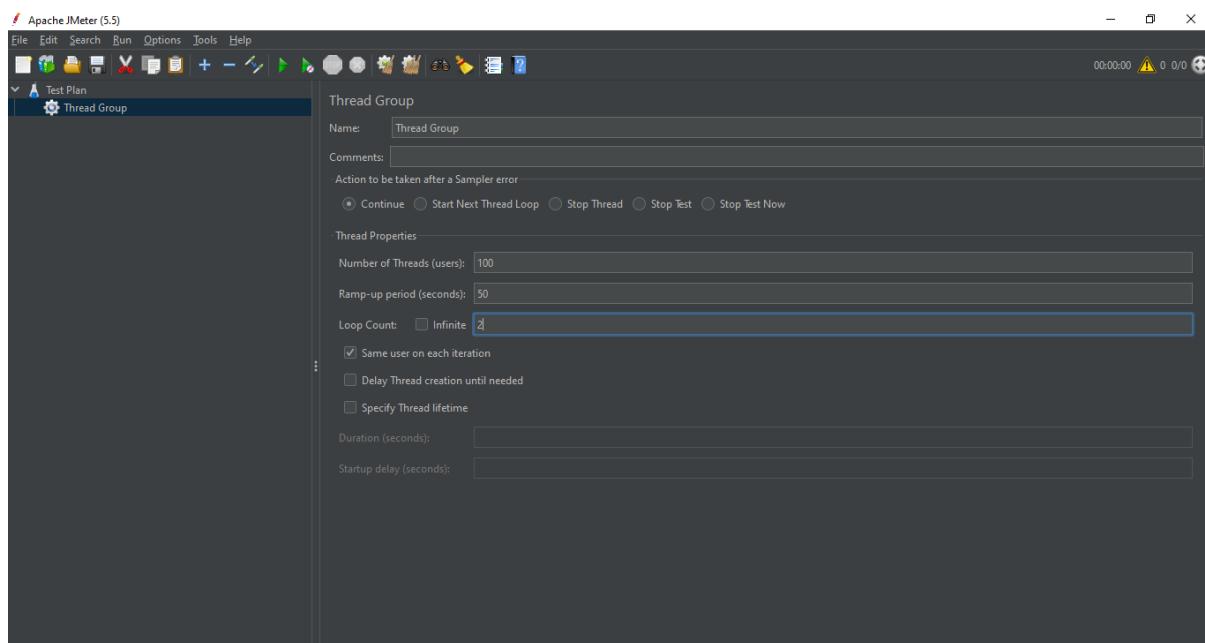
PRACTICAL 10

AIM: Load Testing Using JMeter.

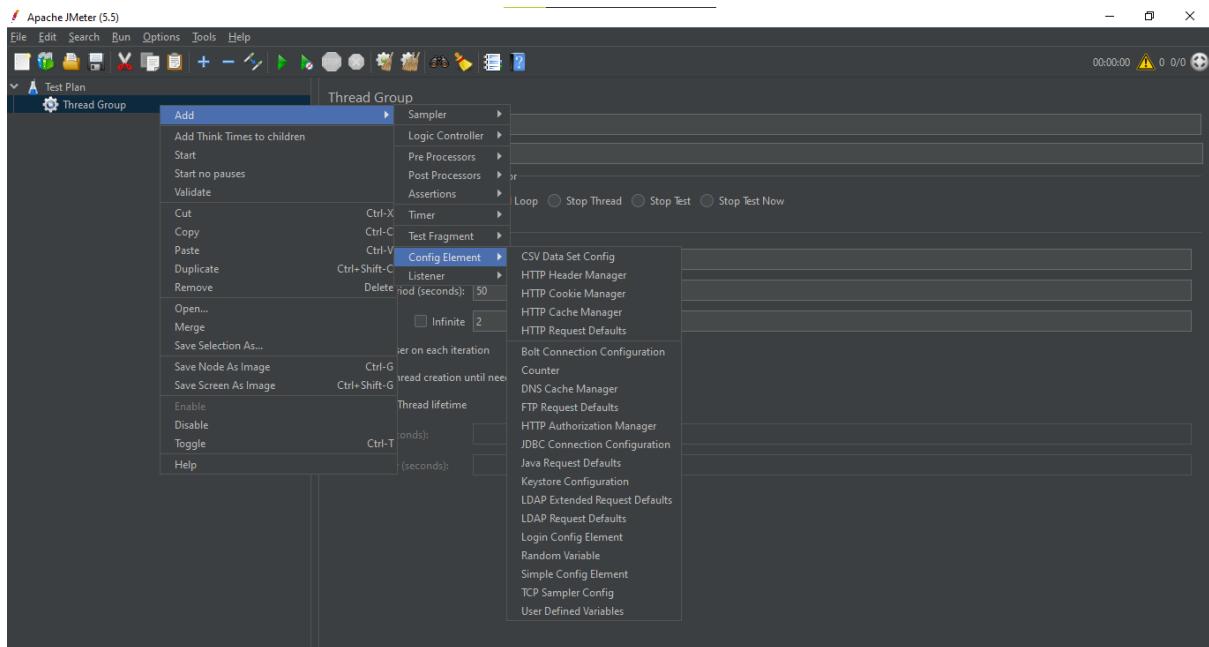
Step 1: Right click on the "Test Plan" and add a new thread group: Add -> Threads (Users) -> Thread Group



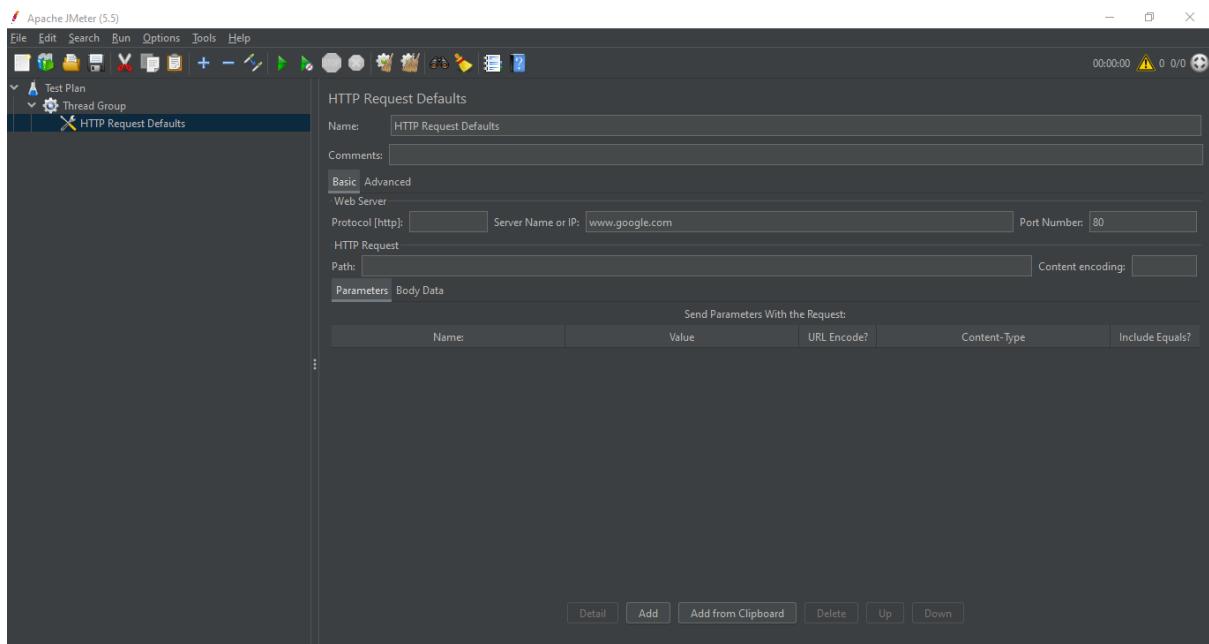
Step 2: Thread Group Property Window will be opened, Enter the following Properties.



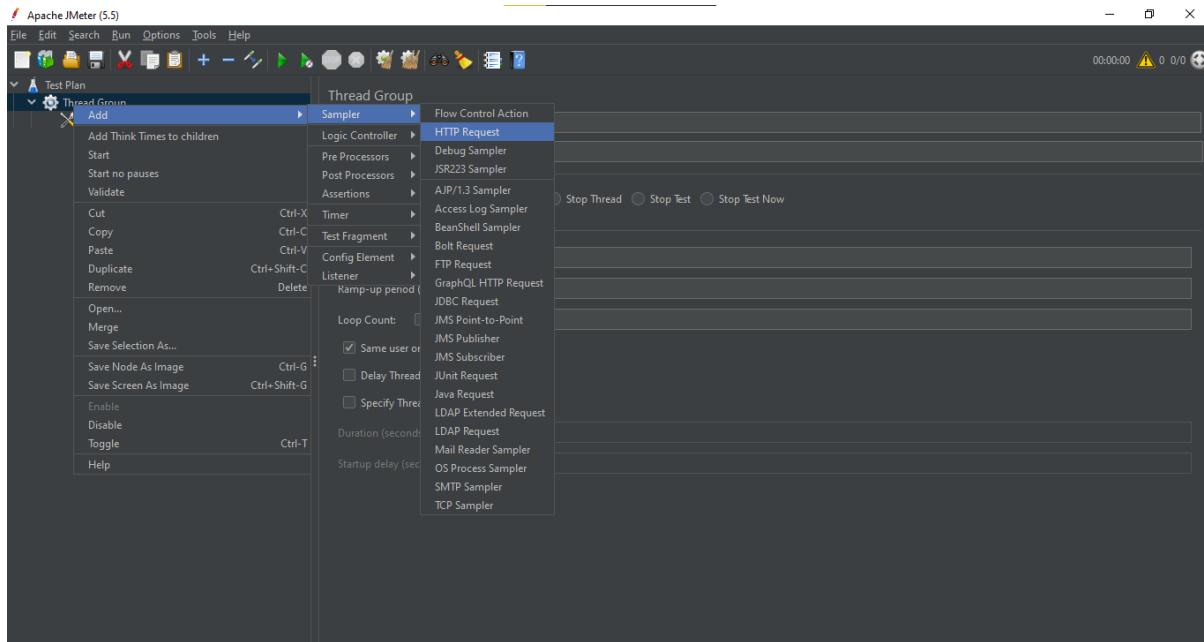
Step 3: Right Click on Thread Group then Add→Config Element→HTTP Request Defaults



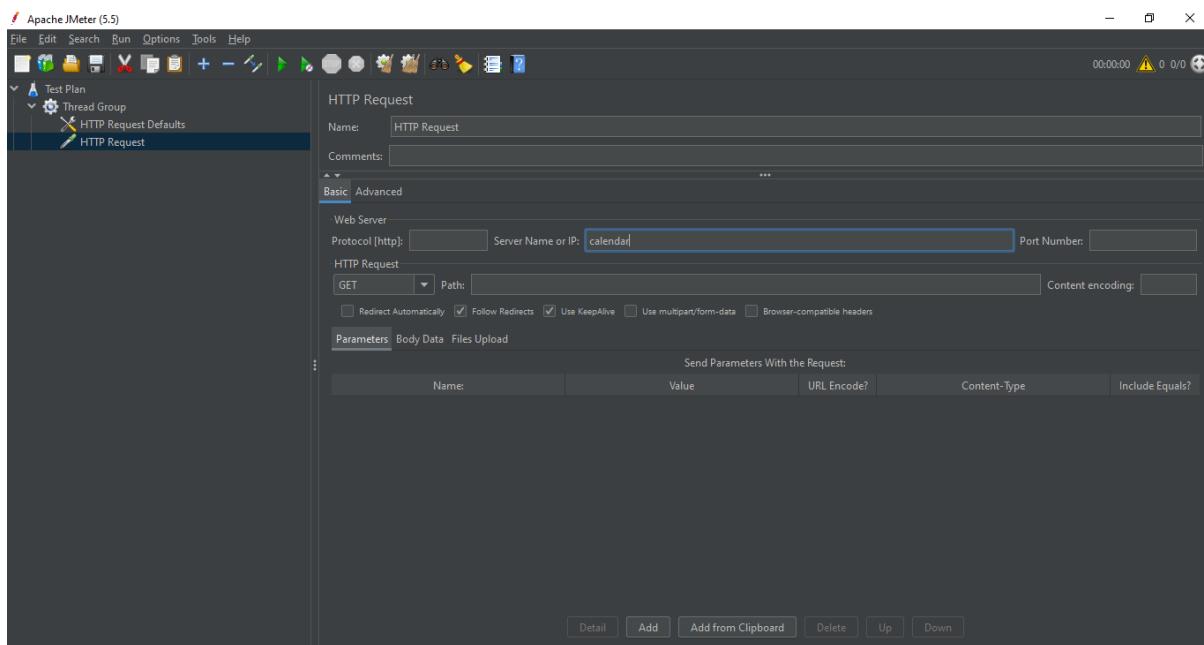
Step 4: Enter www.google.com in the path section, and Port Number as 80.



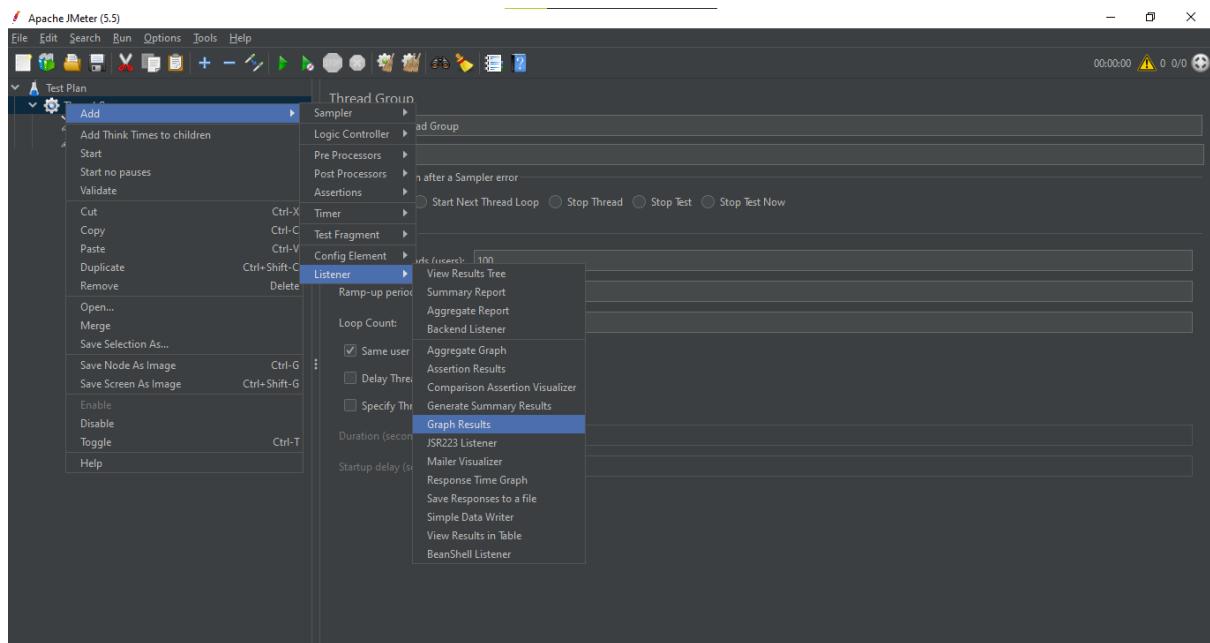
Step 5: Right Click on Thread Group → Add → Sampler → HTTP Request



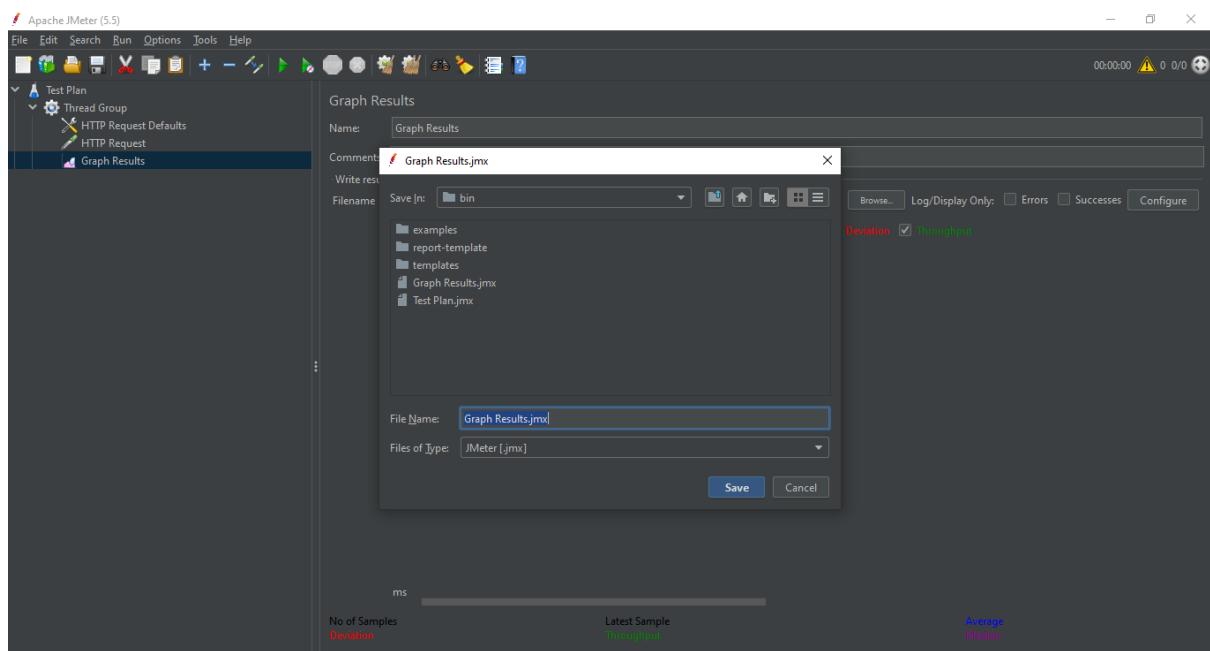
Step 6: Type calendar in the Path Section.

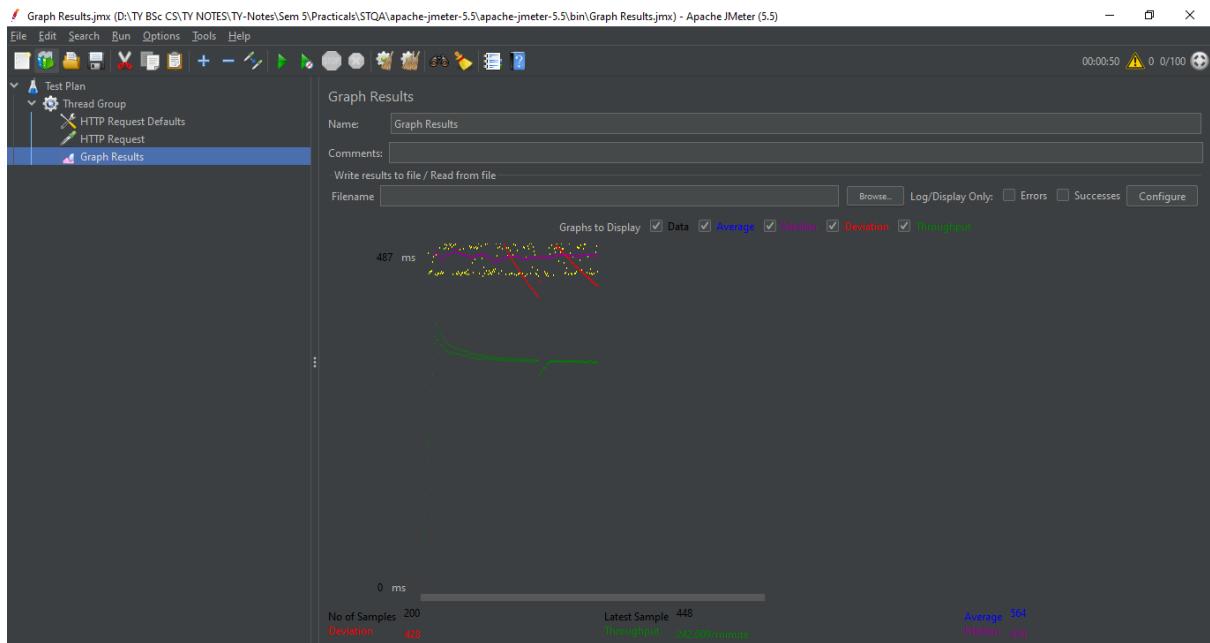


Step 7: Right Click on Thread Group → Add → Listener → Graph Results



Step 8: Save the Test before Running.



Step 9: Run the Test.

Practical No.1

Aim:- Implement Breadth First Search Algorithm for Romanian Map Problem.

CODE:-

```
import collections

def bfs(graph,root):

    seen,queue=set([root]),collections.deque([root])

    while queue:

        vertex=queue.popleft()

        visit(vertex)

        for node in graph[vertex]:

            if node not in seen:

                seen.add(node)

                queue.append(node)

def allpath(st,end,gr):

    todo=[(st,[st])]

    while len(todo):

        node,path=todo.pop(0)

        for next_node in gr[node]:

            if next_node in path:

                continue

            print('Ideal solution')

            elif next_node==end:

                yield path + [next_node]

            else:

                todo.append((next_node,path + [next_node]))


def visit(n):
```

```
print(n)

def bfs_shortest_path(graph, source, destination):

    checked=[]
    queue=[[source]]
    if source == destination:
        return "SOURCE IS DESTINATION :"
    while queue:
        path=queue.pop(0)
        node=path[-1]
        if node not in checked:
            neighbours = graph[node]

            for neighbour in neighbours:
                new_path=list(path)
                new_path.append(neighbour)
                queue.append(new_path)
                if neighbour == destination:
                    return new_path

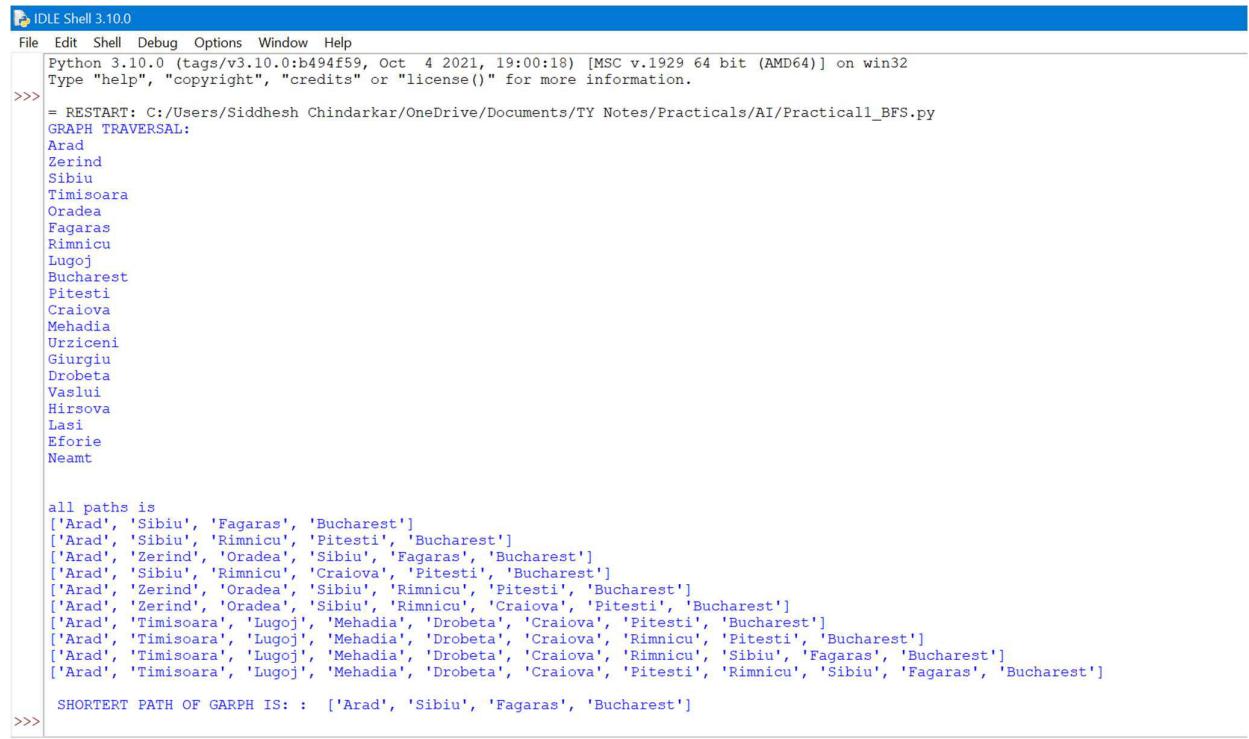
            checked.append(node)
    return "PATH DOES NOT EXIST :"

graph={'Oradea': ['Zerind', 'Sibiu'],
       'Zerind':['Oradea', 'Arad'],
       'Arad': ['Zerind','Sibiu','Timisoara'],
       'Timisoara': ['Arad', 'Lugoj'],
       'Lugoj': ['Timisoara', 'Mehadia'],
       'Mehadia': ['Lugoj', 'Drobeta'],
```

```
'Drobeta': ['Mehadia', 'Craiova'],
'Craiova': ['Drobeta', 'Rimnicu', 'Pitesti'],
'Pitesti': ['Rimnicu', 'Craiova', 'Bucharest'],
'Sibiu': ['Oradea', 'Fagaras', 'Rimnicu', 'Arad'],
'Fagaras': ['Sibiu', 'Bucharest'],
'Rimnicu': ['Sibiu', 'Pitesti', 'Craiova'],
'Bucharest': ['Urziceni', 'Giurgiu'],
'Giurgiu': ['Bucharest'],
'Urziceni': ['Bucharest', 'Vaslui', 'Hirsova'],
'Vaslui': ['Lasi', 'Urziceni'],
'Lasi': ['Neamt', 'Vaslui'],
'Neamt': ['Lasi'],
'Hirsova': ['Urziceni', 'Eforie'],
'Eforie': ['Hirsova']
}
```

```
print("GRAPH TRAVERSAL: ")
bfs(graph,'Arad')
print('\n\nall paths is')
[print (x) for x in allpath ('Arad','Bucharest', graph)]
print("\n SHORTEST PATH OF GRAPH IS: : ", bfs_shortest_path(graph, 'Arad','Bucharest'))
```

Output:-



```
Python 3.10.0 (tags/v3.10.0:b494f59, Oct  4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>> = RESTART: C:/Users/Siddhesh Chindarkar/OneDrive/Documents/TY Notes/Practicals/AI/Practical1_BFS.py
GRAPH TRAVERSAL:
Arad
Zerind
Sibiu
Timisoara
Oradea
Fagaras
Rimnicu
Lugoj
Bucharest
Pitesti
Craiova
Mehadia
Urziceni
Giurgiu
Drobeta
Vaslui
Hirsova
Lasi
Eforie
Neamt

all paths is
['Arad', 'Sibiu', 'Fagaras', 'Bucharest']
['Arad', 'Sibiu', 'Rimnicu', 'Pitesti', 'Bucharest']
['Arad', 'Zerind', 'Oradea', 'Sibiu', 'Fagaras', 'Bucharest']
['Arad', 'Sibiu', 'Rimnicu', 'Craiova', 'Pitesti', 'Bucharest']
['Arad', 'Zerind', 'Oradea', 'Sibiu', 'Rimnicu', 'Pitesti', 'Bucharest']
['Arad', 'Timisoara', 'Lugoj', 'Mehadia', 'Drobeta', 'Craiova', 'Pitesti', 'Bucharest']
['Arad', 'Timisoara', 'Lugoj', 'Mehadia', 'Drobeta', 'Craiova', 'Rimnicu', 'Pitesti', 'Bucharest']
['Arad', 'Timisoara', 'Lugoj', 'Mehadia', 'Drobeta', 'Craiova', 'Rimnicu', 'Sibiu', 'Fagaras', 'Bucharest']
['Arad', 'Timisoara', 'Lugoj', 'Mehadia', 'Drobeta', 'Craiova', 'Pitesti', 'Rimnicu', 'Sibiu', 'Fagaras', 'Bucharest']

SHORTERT PATH OF GARPH IS: : ['Arad', 'Sibiu', 'Fagaras', 'Bucharest']
```

Practical No.2

Aim:- Implement Depth First Search Algorithm for Romanian Map Problem.

CODE:-

```
graph={ 'Oradea': ['Zerind','Sibiu'],
        'Zerind': ['Oradea' , 'Arad'],
        'Arad' : ['Zerind' , 'Sibiu','Timisoara'],
        'Timisoara' : ['Arad' , 'Lugoj'],
        'Lugoj' : ['Timisoara' , 'Mehadia'],
        'Mehadia' : ['Lugoj' , 'Dobreta'],
        'Dobreta' : ['Mehadia' , 'Craiova'],
        'Craiova' : ['Dobreta' , 'Pitesti' , 'Rimnicu Vilcea'],
        'Pitesti' : ['Rimnicu Vilcea' , 'Craiova' , 'Bucharest'],
        'Rimnicu Vilcea' : ['Sibiu' , 'Pitesti' , 'Craiova' ],
        'Sibiu' : ['Oradea' , 'Rimnicu Vilcea' , 'Arad' , 'Fagaras'],
        'Fagaras' : ['Sibiu' , 'Bucharest'],
        'Bucharest' : ['Urziceni' , 'Giurgiu'],
        'Giurgiu' : ['Bucharest'],
        'Urziceni' : ['Bucharest' , 'Valsui' , 'Hirsova'],
        'Valsui' : ['Lasi' , 'Urziceni'],
        'Lasi' : ['Valsui' , 'Neamt'],
        'Neamt' : ['Lasi'],
        'Hirsova' : ['Urziceni' , 'Eforie'],
        'Eforie' : ['Hirsova'],
    }
```

```
def dfs(g, n, seen, d):
```

```
    if n not in seen:
```

```
        seen.append(n)
```

```
for i in g[n]:  
    if seen[-1] in d:  
        break  
    dfs(g, i, seen, d)  
  
return seen  
  
print("\n The nodes --> \n Oradea, Zerind, Arad, Timisoara, Arad, Lugoj, Mehadia, Dobreta,  
Craiova, Pitesti, Rimnicu Vilcea, Sibiu, Fagaras, Bucharest, Giurgiu,Urziceni, Valsui, Lasi,  
Neamt, Hirsova, Eforie')
```

```
X=input("\nEnter starting node:")  
Y=input("Enter goal node:")  
print("The path from Starting node to goal node is given by:")  
print(dfs(graph, X, [], Y))
```

Output:-

```
IDLE Shell 3.10.0  
File Edit Shell Debug Options Window Help  
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>> = RESTART: C:\Users\Siddhesh Chindarkar\OneDrive\Documents\TY Notes\Practicals\AI\Practical2_DFS.py  
The nodes -->  
Oradea, Zerind, Arad, Timisoara, Arad, Lugoj, Mehadia, Dobreta, Craiova, Pitesti, Rimnicu Vilcea, Sibiu, Fagaras, Bucharest, Giurgiu,Urziceni, Valsui, L  
asi, Neamt, Hirsova, Eforie  
Enter starting node:Dobreta  
Enter goal node:Arad  
The path from Starting node to goal node is given by:  
['Dobreta', 'Mehadia', 'Lugoj', 'Timisoara', 'Arad']  
>>>
```

Practical No.3

Aim:- Implement Depth Limited Search Algorithm for Romanian Map Problem.

Code:-

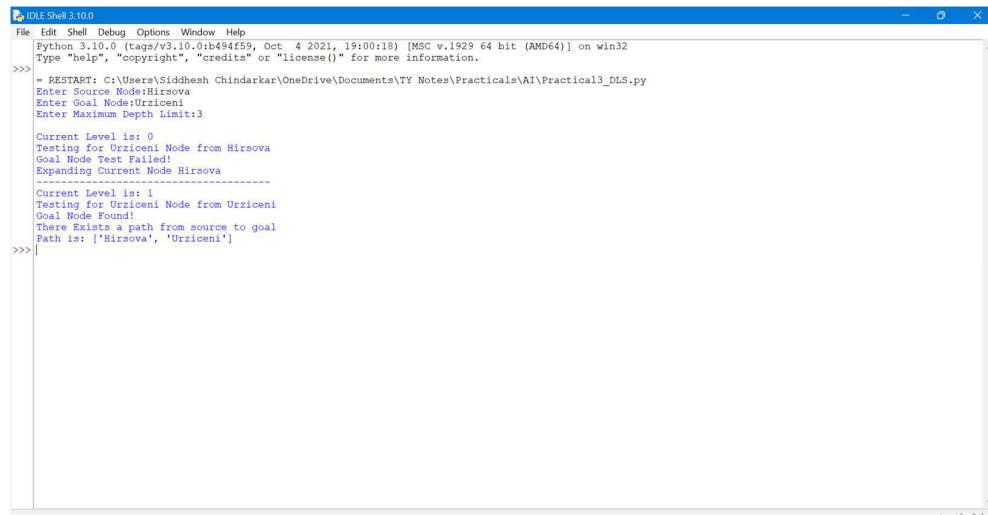
```
graph={  
    'Oradea': ['Zerind', 'Sibiu'],  
    'Zerind':['Oradea', 'Arad'],  
    'Arad': ['Zerind','Sibiu','Timisoara'],  
    'Timisoara': ['Arad', 'Lugoj'],  
    'Lugoj': ['Timisoara', 'Mehadia'],  
    'Mehadia': ['Lugoj', 'Drobeta'],  
    'Drobeta': ['Mehadia', 'Craiova'],  
    'Craiova': ['Drobeta', 'Rimnicu', 'Pitesti'],  
    'Pitesti': ['Rimnicu', 'Craiova', 'Bucharest'],  
    'Sibiu': ['Oradea', 'Fagaras', 'Rimnicu','Arad'],  
    'Fagaras': ['Sibiu', 'Bucharest'],  
    'Rimnicu': ['Sibiu', 'Pitesti','Craiova'],  
    'Bucharest': ['Urziceni','Giurgiu'],  
    'Giurgiu': ['Bucharest'],  
    'Urziceni': ['Bucharest', 'Vaslui', 'Hirsova'],  
    'Vaslui': ['Lasi', 'Urziceni'],  
    'Lasi': ['Neamt', 'Vaslui'],  
    'Neamt': ['Lasi'],  
    'Hirsova': ['Urziceni', 'Eforie'],  
    'Eforie': ['Hirsova']  
}  
  
def dls(s,g,path,level,max_depth):  
    print("Current Level is:",level)
```

```
print("Testing for",g+" "+ "Node from",s)
path.append(s)
e="Max Depth Limit Reached!"
while True:
    if level>max_depth:
        print("Current Level Reaches Maximum Depth")
        return False
    break
if s==g:
    print("Goal Node Found!")
    return path
print("Goal Node Test Failed!")
print("Expanding Current Node",s)
print("-----")
for neighbor in graph[s]:
    if dls(neighbor,g,path,level+1,max_depth):
        return path
    path.pop()
    return False
return False

s=input("Enter Source Node=")
g=input("Enter Goal Node=")
max_depth=int(input("Enter Maximum Depth Limit="))
print()
path=list()
output=dls(s,g,path,0,max_depth)
if (output):
```

```
print("There Exists a path from source to goal")  
print("Path is:",path)  
  
else:  
  
    print("No Path From Source to Goal in given depth Limit")
```

Output:-



```
IDLE Shell 3.10.0  
File Edit Shell Debug Options Window Help  
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
=>>> = RESTART: C:\Users\Siddhesh Chindarkar\OneDrive\Documents\TY Notes\Practicals\AI\Practical3_DLS.py  
Enter Source Node:Hirsova  
Enter Goal Node:Urziceni  
Enter Maximum Depth Limit:3  
  
Current Level is: 0  
Testing for Urziceni Node from Hirsova  
Goal Node Test Failed!  
Expanding Current Node Hirsova  
-----  
Current Level is: 1  
Testing for Urziceni Node from Urziceni  
Goal Node Found!  
There Exists a path from source to goal  
Path is: ['Hirsova', 'Urziceni']  
=>>>
```

Practical No.4

Aim:- Implement Iterative Deep Depth First Search for Romanian Map Problem.

Code:-

```
graph = {'Oradea': ['Zerind', 'Sibiu'],
         'Zerind': ['Oradea', 'Arad'],
         'Arad': ['Zerind', 'Sibiu', 'Timisoara'],
         'Timisoara': ['Arad', 'Lugoj'],
         'Lugoj': ['Timisoara', 'Mehadia'],
         'Mehadia': ['Lugoj', 'Drobeta'],
         'Drobeta': ['Mehadia', 'Craiova'],
         'Craiova': ['Drobeta', 'Rimnicu', 'Pitesti'],
         'Pitesti': ['Rimnicu', 'Craiova', 'Bucharest'],
         'Sibiu': ['Oradea', 'Fagaras', 'Rimnicu', 'Arad'],
         'Fagaras': ['Sibiu', 'Bucharest'],
         'Rimnicu': ['Sibiu', 'Pitesti', 'Craiova'],
         'Bucharest': ['Urziceni', 'Giurgiu'],
         'Giurgiu': ['Bucharest'],
         'Urziceni': ['Bucharest', 'Vaslui', 'Hirsova'],
         'Vaslui': ['Lasi', 'Urziceni'],
         'Lasi': ['Neamt', 'Vaslui'],
         'Neamt': ['Lasi'],
         'Hirsova': ['Urziceni', 'Eforie'],
         'Eforie': ['Hirsova']
     }
```

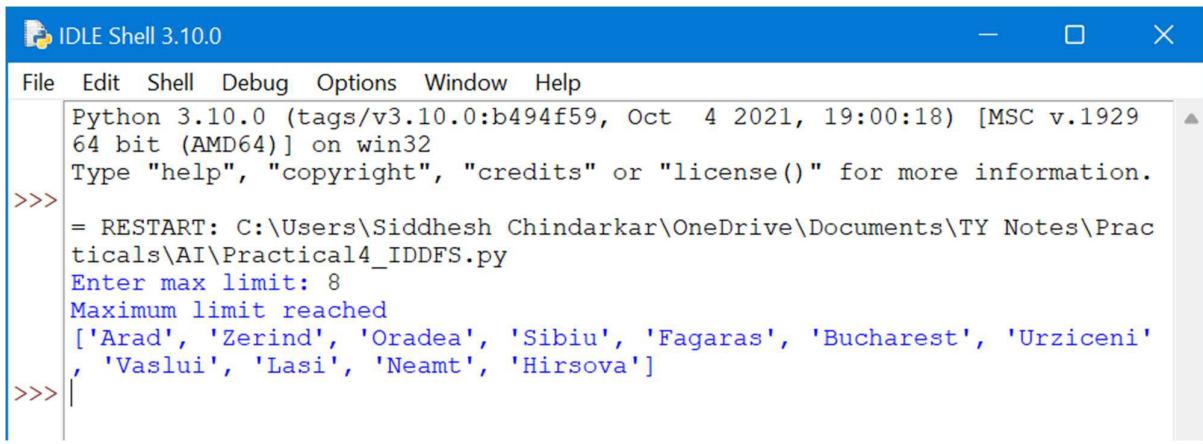
```
def iddfs(g, n, seen, dst, dep, lim):
```

```
    if n not in seen:
```

```
seen.append(n)
if dep <= lim:
    for i in g[n]:
        if seen[-1] is dst:
            return seen
    iddfs(g, i, seen, dst, dep + 1, lim)
else:
    print("Maximum limit reached")
return None

print(iddfs(graph, 'Arad', [], 'Hirsova', 0, int(input("Enter max limit: "))))
```

Output:-



```
IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929
64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Siddhesh Chindarkar\OneDrive\Documents\TY Notes\Practicals\AI\Practical4_IDDFS.py
Enter max limit: 8
Maximum limit reached
['Arad', 'Zerind', 'Oradea', 'Sibiu', 'Fagaras', 'Bucharest', 'Urziceni',
 'Vaslui', 'Lasi', 'Neamt', 'Hirsova']
>>> |
```

Practical No.5

Aim:- Implement Recursive Best First Search for Romanian Map Problem.

Code:-

```
graph = {  
    'Oradea': ({'Zerind':71, 'Sibiu':151},380),  
    'Zerind': ({'Oradea':71, 'Arad':75},374),  
    'Arad': ({'Zerind':75, 'Sibiu':140, 'Timisoara':118},366),  
    'Timisoara': ({'Arad':118, 'Lugoj':111},329),  
    'Lugoj': ({'Timisoara':111, 'Mehadia':70},244),  
    'Mehadia': ({'Lugoj':70, 'Dobreta':75},241),  
    'Sibiu': ({'Oradea':151, 'Rimnicu Vilcea':80, 'Arad':140, 'Fagaras':99},253),  
    'Fagaras': ({'Sibiu':99, 'Bucharest':211},176),  
    'Rimnicu Vilcea': ({'Sibiu':80, 'Pitesti':97, 'Craiova':146},193),  
    'Bucharest': ({'Urziceni':85,'Fagaras':211,'Pitesti':101,'Giurgiu':90},0),  
    'Dobreta': ({'Mehadia':75, 'Craiova':120},242),  
    'Craiova': ({'Dobreta':120, 'Pitesti':138, 'Rimnicu Vilcea':97},160),  
    'Pitesti': ({'Rimnicu Vilcea':97, 'Craiova':138, 'Bucharest':101},100),  
    'Urziceni': ({'Bucharest':85, 'Valsui':142, 'Hirsova':98},80),  
    'Giurgiu': ({'Bucharest':90},77),  
    'Valsui': ({'Lasi':92, 'Urziceni':142},199),  
    'Hirsova': ({'Urziceni':98, 'Eforie':86},151),  
    'Lasi': ({'Valsui':92, 'Neamt':87},226),  
    'Eforie': ({'Hirsova':86},161),  
    'Neamt': ({'Lasi':87},234)  
}
```

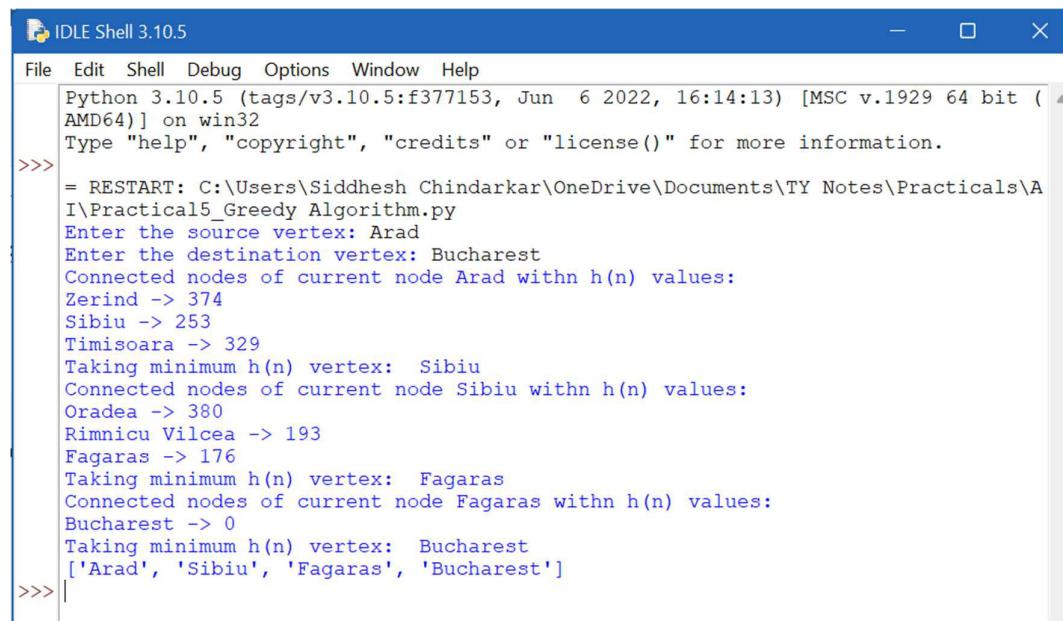
```
def greedy_search_rec(graph,prev,dst,path,q):  
    # n:(h(n))  
    print("Connected nodes of current node",prev,"withn h(n) values: ")
```

```
for n in graph[prev][0]: #neighbour list prev=Arad, -> Z,S,T
    if n not in path:
        q[n] = graph[n][1] #n=z [1]=374
        print(n,"->",q[n])

while q:
    mn = min(q,key=q.get)
    print("Taking minimum h(n) vertex: ",mn)
    #print(mn)
    if dst == mn:
        return path + [dst]
    #del q[mn]
    new_path = greedy_search_rec(graph,mn,dst,path + [mn],q)
    if new_path:
        return new_path
    return []

sourec = input("Enter the source vertex: ")
dest = input("Enter the destination vertex: ")
path = greedy_search_rec(graph,sourec,dest,[sourec],{})
if path:
    print(path)
else:
    print("Path not found")
```

Output:-



The screenshot shows the IDLE Shell 3.10.5 interface with the following output:

```
File Edit Shell Debug Options Window Help
Python 3.10.5 (tags/v3.10.5:f377153, Jun  6 2022, 16:14:13) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>> = RESTART: C:\Users\Siddhesh Chindarkar\OneDrive\Documents\TY Notes\Practicals\AI\Practical5_Greedy Algorithm.py
Enter the source vertex: Arad
Enter the destination vertex: Bucharest
Connected nodes of current node Arad withn h(n) values:
Zerind -> 374
Sibiu -> 253
Timisoara -> 329
Taking minimum h(n) vertex: Sibiu
Connected nodes of current node Sibiu withn h(n) values:
Oradea -> 380
Rimnicu Vilcea -> 193
Fagaras -> 176
Taking minimum h(n) vertex: Fagaras
Connected nodes of current node Fagaras withn h(n) values:
Bucharest -> 0
Taking minimum h(n) vertex: Bucharest
['Arad', 'Sibiu', 'Fagaras', 'Bucharest']
```

Practical No.6

Aim:- Implement A* Algorithm for Romanian Map Problem.

Code:-

```
graph={  
    "O":({"Z":71,"S":151},380),  
    "Z":({"O":71,"A":75},374),  
    "A":({"Z":75,"S":140,"T":118},366),  
    "T":({"A":118,"L":111},329),  
    "L":({"T":111,"M":70},244),  
    "M":({"L":70,"D":75},241),  
    "S":({"O":151,"F":99,"RV":80,"A":140},253),  
    "F":({"S":99,"B":211},176),  
    "RV":({"S":80,"P":97,"C":146},193),  
    "B":({"F":211,"P":101,"U":85,"G":90},0),  
    "P":({"RV":97,"C":138,"B":101},100),  
    "C":({"RV":146,"P":138,"D":120},160),  
    "D":({"M":75,"C":120},242),  
    "U":({"B":85,"V":142,"":98},80),  
    "G":({"B":90},77),  
    "V":({"L":92,"U":142},199),  
    "H":({"U":98,"E":86},151),  
    "I":({"V":92,"N":87},226),  
    "E":({"H":86},161),  
    "N":({"L":87},234)  
}  
  
def get_min(q):  
    mn=(0,(float("INF")))  
    for i in q:  
        if sum(q[i])<sum(mn[1]):
```

```
mn=(i,q[i])
return mn[0]

def a_star(graph,prev,dst,path,pcost,q):
    print("Connected nodes of current nodes",prev,"with h(n) values:")
    for n in graph[prev][0]:
        if n not in path:
            q[n]=(graph[n][1],graph[prev][0][n])
            print(n,"-->",q[n])
            add1=sum(q[n])
            path_cost=pcost+add1
            print("A* value for ",n,"is:",path_cost)

    while q:
        mn=get_min(q)
        print("Selectiong Minimum vertex:",mn)
        print("_____")
        if dst==mn:
            return path+[dst]
        pc=pcost+q[mn][1]
        print("Previous path cost:",pc)
        new_path=a_star(graph,mn,dst,path+[mn],pc,q)
        if new_path:
            return new_path
    return[]

source=input("Enter Source vertex:")
dest=input("Enter destination vertex:")
heuristic=int(input("Enter given heuristic value for source:"))
path=a_star(graph,source,dest,[],0,{source:(heuristic,0)})
if path:
    print(path)
```

else:

```
    print("Path is not found")
```

Output:-

```
IDLE Shell 3.10.5
File Edit Shell Debug Options Window Help
Python 3.10.5 (tags/v3.10.5:f377153, Jun 6 2022, 16:14:13) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>> = RESTART: C:\Users\Siddhesh Chindarkar\OneDrive\Documents\TY Notes\Practicals\AI\Practical6_Astar Algorithm.py
Enter Source vertex:A
Enter destination vertex:B
Enter given heuristic value for source:399
Connected nodes of current nodes A with h(n) values:
Z --> (374, 75)
A* value for Z is: 449
S --> (253, 140)
A* value for S is: 393
T --> (329, 118)
A* value for T is: 447
Selecting Minimum vertex: S

Previous path cost: 140
Connected nodes of current nodes S with h(n) values:
O --> (380, 151)
A* value for O is: 671
F --> (176, 99)
A* value for F is: 415
RV --> (193, 80)
A* value for RV is: 413
A --> (366, 140)
A* value for A is: 646
Selecting Minimum vertex: RV

Previous path cost: 220
Connected nodes of current nodes RV with h(n) values:
P --> (100, 97)
A* value for P is: 417
C --> (160, 146)
A* value for C is: 526
Selecting Minimum vertex: P
```

```
IDLE Shell 3.10.5
File Edit Shell Debug Options Window Help
Enter destination vertex:B
Enter given heuristic value for source:399
Connected nodes of current nodes A with h(n) values:
Z --> (374, 75)
A* value for Z is: 449
S --> (253, 140)
A* value for S is: 393
T --> (329, 118)
A* value for T is: 447
Selecting Minimum vertex: S

Previous path cost: 140
Connected nodes of current nodes S with h(n) values:
O --> (380, 151)
A* value for O is: 671
F --> (176, 99)
A* value for F is: 415
RV --> (193, 80)
A* value for RV is: 413
A --> (366, 140)
A* value for A is: 646
Selecting Minimum vertex: RV

Previous path cost: 220
Connected nodes of current nodes RV with h(n) values:
P --> (100, 97)
A* value for P is: 417
C --> (160, 146)
A* value for C is: 526
Selecting Minimum vertex: P

Previous path cost: 317
Connected nodes of current nodes P with h(n) values:
C --> (160, 138)
A* value for C is: 615
B --> (0, 101)
A* value for B is: 418
Selecting Minimum vertex: B

['S', 'RV', 'P', 'B']
>>>
```

Practical No.7

Aim:- Implement Naïve Bayes Learning Algorithm for Restaurant Waiting Problem.

Code:-

```
#import operator

#data set => already taken for prediction

dataset = {

    "Ans":      ["Yes","No","Yes","Yes","No","Yes","Yes","No","Yes","No","No","Yes"],

    "Alternate": ["Yes","Yes","No","No","Yes","No","No","Yes","Yes","No","Yes"],

    "Bar":       ["No","Yes","Yes","Yes","No","No","Yes","No","No","Yes","Yes","No"],

    "Fri/Sat":   ["Yes","No","No","No","Yes","No","Yes","Yes","Yes","No","No","Yes"],

    "Hungry":    ["No","Yes","No","Yes","No","Yes","No","Yes","No","Yes","No","Yes"],

    "Patrons":   ["Some","Full","Full","None","Full","Some","Some","Full","Full","Some","Full","Some"],

    "Price":     ["High","Low","Low","Low","High","High","High","Low","High","Low","High","Low"],

    "Raining":   ["Yes","Yes","No","No","No","No","Yes","No","Yes","No","No"],

    "Reservation": ["Yes","Yes","No","Yes","No","No","Yes","No","Yes","No","Yes"],

    "Type":      ["French","Thai","Burger","Italian","Italian","Thai","French","Thai","Burger","Italian","Burge
r","French"],

    "WaitEstimate": ["10-30","0-10",>"60","30-60","10-30","0-10",>"60","30-60","30-60",>"60","10-30","0-10"]
}
```

#input data to test or predict

```
test_case={

    "Alternate":  "Yes",

    "Bar":        "No",

    "Fri/Sat":    "No",

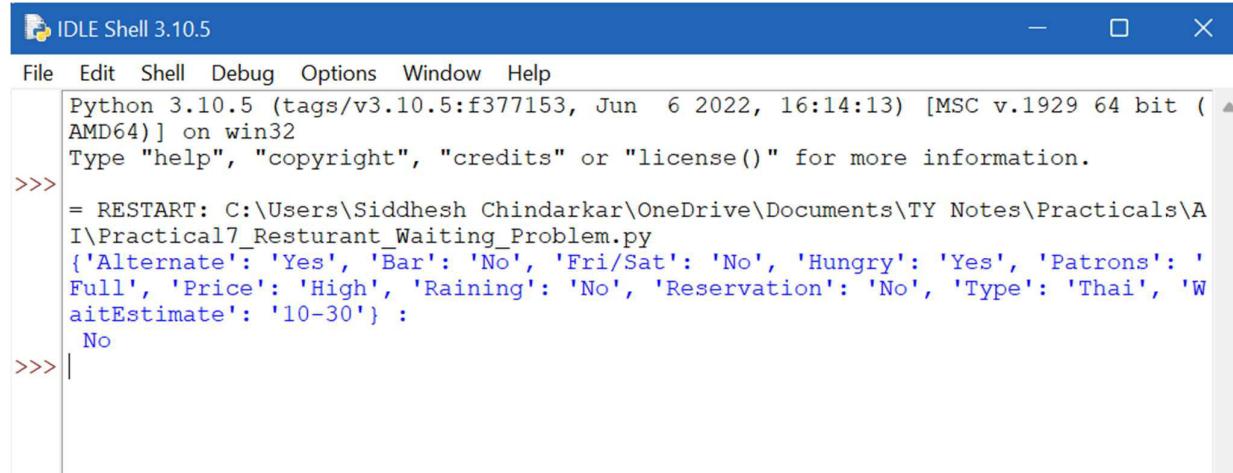
    "Hungry":     "Yes",
```

```
"Patrons": "Full",
"Price": "High",
"Raining": "No",
"Reservation": "No",
>Type": "Thai",
"WaitEstimate": "10-30"
}
```

```
def build_probs(ds, test_case):
    ans = ds["Ans"]#output attribute (ans)
    length = len(ans) #total length of output(ans)
    ans_set = set(ans) #unique (individual) classes yes and no
    count_ans = {k: ans.count(k) for k in ans_set}
    calc_prob = {k: count_ans[k] / length for k in ans_set}
    for ft in ds:
        if ft != "Ans":
            counts = {attr: {k: 0 for k in ans_set} for attr in set(ds[ft])}
            for i in range(length):
                counts[ds[ft][i]][ans[i]] +=1
            for k in ans_set:
                calc_prob[k] *= counts[test_case[ft]][k]/ count_ans[k]
    print(test_case,":\n",max(calc_prob, key=calc_prob.get))

build_probs(dataset, test_case)
```

Output:-



The screenshot shows a Python shell window titled "IDLE Shell 3.10.5". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following Python code and its output:

```
Python 3.10.5 (tags/v3.10.5:f377153, Jun  6 2022, 16:14:13) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>> = RESTART: C:\Users\Siddhesh Chindarkar\OneDrive\Documents\TY Notes\Practicals\A
I\Practical7_Resturant_Waiting_Problem.py
{'Alternate': 'Yes', 'Bar': 'No', 'Fri/Sat': 'No', 'Hungry': 'Yes', 'Patrons': 'Full',
 'Price': 'High', 'Raining': 'No', 'Reservation': 'No', 'Type': 'Thai', 'WaitEstimate': '10-30'} :
    No
>>> |
```

Practical No.8

Aim:- Implement Decision Tree Algorithm for the Restaurant Waiting Problem.

Code:-

```
import pandas as pd

from sklearn.tree import DecisionTreeClassifier

from sklearn .model_selection import train_test_split

from sklearn import metrics

from matplotlib import pyplot as plt

from sklearn import tree

col_names=['Reservation','Raining','BadService','Satur','Result']

hotldata=pd.read_csv("dtree.csv",header=None,names=col_names)

feature_cols=['Reservation','Raining','BadService','Satur']

X=hotldata[feature_cols]

Y=hotldata.Result

X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.3,random_state=1)

print(hotldata)

clf=DecisionTreeClassifier(criterion="entropy",max_depth=5)

clf=clf.fit(X_train,Y_train)

Y_pred=clf.predict(X_test)

print("ytest=",X_test)

print("ypred=",Y_pred)

print("Accuracy:",metrics.accuracy_score(Y_test,Y_pred))

fig=plt.figure(figsize=(25,20))

t=tree.plot_tree(clf,feature_names=feature_cols,class_names=['Leave','Wait'],filled=True)

fig.savefig("decistion_tree.png")
```

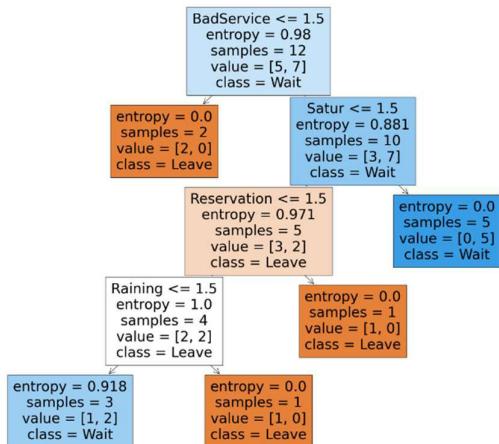
Output:-

```

IDLE Shell 3.10.5
File Edit Shell Debug Options Window Help
Python 3.10.5 (tags/v3.10.5:f377153, Jun  6 2022, 16:14:13) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>>
= RESTART: C:\Users\Siddhesh Chindarkar\OneDrive\Documents\TY Notes\Practicals\AI\Practical8_Decision_Tree.py
      Reservation  Raining  BadService  Satur  Result
0          1         1         2         1    Wait
1          2         2         2         2    Wait
2          2         1         1         1   Leave
3          1         1         1         1    Wait
4          2         1         2         1   Leave
5          2         1         2         2    Wait
6          1         2         2         1   Leave
7          2         1         2         2    Wait
8          1         1         2         1   Leave
9          1         2         1         2   Leave
10         2         1         2         2    Wait
11         1         2         2         1   Leave
12         2         1         2         2    Wait
13         1         1         2         1   Leave
14         1         2         1         2   Leave
15         1         1         2         1    Wait
16         2         2         2         2    Wait
17         2         1         1         1   Leave
ytest=
      Reservation  Raining  BadService  Satur
6          1         2         2         1
3          1         1         1         1
13         1         1         2         1
2          2         1         1         1
14         1         2         1         2
7          2         1         2         2
ypred= ['Leave' 'Leave' 'Wait' 'Leave' 'Leave' 'Wait']
Accuracy: 0.6666666666666666
>>> |

```



Practical No.9

Aim:- Implement Majority Voting Classifier in Ensemble Learning.

Code:-

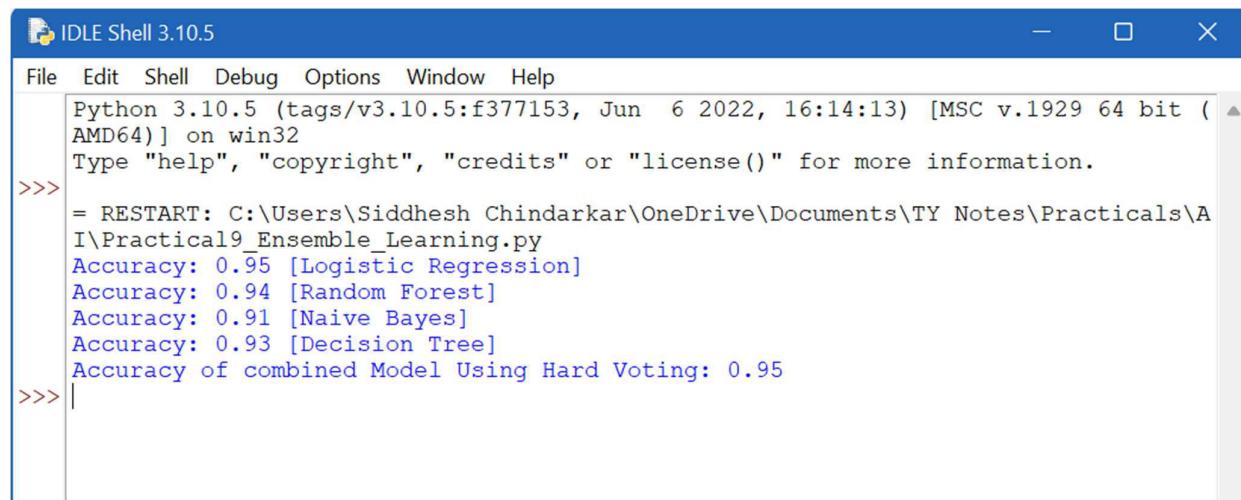
```
import numpy as np
from numpy import *
from sklearn import datasets
from sklearn import model_selection
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import VotingClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
import warnings
warnings.filterwarnings("ignore")

iris=datasets.load_iris()#iris dataset 5cols
x,y=iris.data[:,1:3],iris.target#first paramete for row, and second for col,here : in row is for all row
m1=LogisticRegression(random_state=1)
m2=RandomForestClassifier(random_state=1)
m3=GaussianNB()
m4=DecisionTreeClassifier()

labels=['Logistic Regression', 'Random Forest', 'Naive Bayes', 'Decision Tree']
for m, label in zip([m1,m2,m3,m4],labels):
    scores=model_selection.cross_val_score(m,x,y,cv=5,scoring='accuracy')
    print("Accuracy: %0.2f [%s]" %(scores.mean(), label))
voting_clf_hard=VotingClassifier(estimators=[(labels[0],m1),
                                              (labels[1],m2),
```

```
(labels[2],m3),  
(labels[3],m4)],  
voting='hard')  
  
scores1=model_selection.cross_val_score(voting_clf_hard,x,y,cv=5,scoring='accuracy')  
print("Accuracy of combined Model Using Hard Voting: %0.2f" %(scores1.mean()))
```

Output:-



```
IDLE Shell 3.10.5  
File Edit Shell Debug Options Window Help  
Python 3.10.5 (tags/v3.10.5:f377153, Jun 6 2022, 16:14:13) [MSC v.1929 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>> = RESTART: C:\Users\Siddhesh Chindarkar\OneDrive\Documents\TY Notes\Practicals\AI\Practical19_Ensemble_Learning.py  
Accuracy: 0.95 [Logistic Regression]  
Accuracy: 0.94 [Random Forest]  
Accuracy: 0.91 [Naive Bayes]  
Accuracy: 0.93 [Decision Tree]  
Accuracy of combined Model Using Hard Voting: 0.95  
>>> |
```