1. Write 5 Test scripts using main method on Registration page in a different class.

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.testng.Assert;

import org.testng.annotations.Test;

public class DemoTestNG

{

public WebDriver driver = new FirefoxDriver();

String appUrl = &quot; https://accounts.google.com&quot.

public void gmailLogin()

{

driver.get(&quot;https://gmail.com&quot;);

driver.manage().window().maximize();

String expectedTitle = &quot; Sign in - Google Accounts &quot;;

String actualTitle = driver.getTitle();

Assert.assertEquals(expectedTitle,actualTitle);

WebElement username = driver.findElement(By.id(&quot;Email&quot;));

username.clear();

username.sendKeys(&quot;TestSelenium&quot;);

WebElement password = driver.findElement(By.id(&quot;Passwd&quot;));

password.clear();

password.sendKeys(&quot;password123&quot;);

WebElement SignInButton = driver.findElement(By.id(&quot;signIn&quot;));

SignInButton.click();

driver.close();

}

}

2. Write all the above 5 Test scripts in a single class using TestNG annotations.

public class test

{

public void beforeMethod()

{

System.out.println(" Before Method will execute before every test method");

}

public void afterMethod()

{

System.out.println("After Method will execute after every test method ");

}

public void beforeClass()

{

System.out.println("Before Class will always execute prior to Before Method and Test Method ");

}

public void afterClass()

{

System.out.println("After Class will always execute later to After Method and Test method");

}

public void beforeTest()

{

System.out.println("Before Test will always execute prior to Before Class, ,Before Method and Test Method ");

}

public void afterTest()

{

System.out.println("After Test will always execute later to After Method, After Class ");

}

public void beforeSuite()

{

System.out.println(“Before Suite will always execute prior to all annotations or tests in the suite.");

}

public void afterSuite()

{

System.out.println("After suite will always execute at last when all the annotations or test in the suite have run.");

}

public void testCase1()

{

System.out.println("This is my First Test Case 1");

}

public void testCase2()

{

System.out.println("This is my Second Test Case 2");

}

}

3. Run the individual test scripts and the entire suite testing.xml.

<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd" >

<suite name="Test-Suite" >

<test name="ToolsQA" >

<classes>

<class name="TestNG" />

</classes>

</test>

</suite>

4. Use all the TestNG annotations and run the scripts.

import org.testng.annotations.Test;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.Assert;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeTest;

public class Test1

{

public String baseUrl = "https://www.browserstack.com/";

String driverPath = "D:\\Selenium\\chromedriver.exe";

public WebDriver driver ;

public void launchBrowser()

{

System.out.println("launching Chrome browser");

System.setProperty("webdriver.chrome.driver", driverPath);

driver = new ChromeDriver();

driver.get(baseUrl);

}

public void verifyHomepageTitle()

{

String expectedTitle = "Most Reliable App & Cross Browser Testing Platform | BrowserStack";

String actualTitle = driver.getTitle();

Assert.assertEquals(actualTitle, expectedTitle);

}

@AfterTest

public void terminateBrowser(){

driver.close();

}

}

6. Create PageObjects class for all the page objects of an application.

import static org.testng.Assert.assertEquals;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

public class BrowserStackHomePage

{

WebDriver driver;

By Header=By.xpath("//h1");

By getStarted=By.xpath("//\*[@id='signupModalButton']");

public BrowserStackHomePage(WebDriver driver)

{

this.driver=driver;

}

public void veryHeader()

{

String getheadertext=driver.findElement(Header).getText();

assertEquals("App & Browser Testing Made Easy", getheadertext);

}

public void clickOnGetStarted()

{

driver.findElement(getStarted).click();

}

}

7. Create PageActions class for all the operations that are performed on the web elements.

public class Test\_Login

{

WebDriver driver;

PageActions\_Login actionLogin;

public void setup()

{

driver = new FirefoxDriver();

driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

driver.get("http://gmail.com/");

}

@Test(priority = 0)

public void SignIntoGMailInvalidPassword()

{

actionLogin = new PageActions\_Login(driver);

actionLogin.enterUserIDPassword("upadhyay40", "xyz");

String loginPageTitle = actionLogin.getWrongPasswordTextMessage();

Assert.assertTrue(loginPageTitle.contains("The email and password you entered don't match."));

}

8. Create a separate Test Suite file for each Module.

import org.testng.annotations.Test;

public class TestDatabase

{

@Test(groups = "db")

public void testConnectOracle()

{

System.out.println("testConnectOracle()");

}

@Test(groups = "db")

public void testConnectMsSQL()

{

System.out.println("testConnectMsSQL");

}

@Test(groups = "db-nosql")

public void testConnectMongoDB()

{

System.out.println("testConnectMongoDB");

}

@Test(groups = { "db", "brokenTests" })

public void testConnectMySQL()

{

System.out.println("testConnectMySQL");

}

}

10. Group the test cases using TestNG Groups attribute.

import org.testng.annotations.Test;

public class Personal\_loan

{

@Test(groups= {"SmokeTest"})

public void WebLoginPersonalLoan()

{

System.out.println("Web Login Personal Loan");

}

public void MobileLoginPersonalLoan()

{

System.out.println("Mobile Login Personal Loan");

}

public void APILoginPersonalLoan()

{

System.out.println("API Login Personal Loan");

}

}

11. Prioritize the test scripts using TestNG priority attribute.

import org.testng.annotations.Test;

public class NoPriorityEx

{

public void one()

{

System.out.println("First");

}

public void two()

{

System.out.println("Second");

}

public void three()

{

System.out.println("Third");

}

}

12. Create a BaseTest and have common code in the BaseTest, All Test Suites should inherit the BaseTest.

public abstract class BaseWeb

{

protected static WebDriver driver;

static void webdrivermanagerSetup()

{

WebDriverManager.chromedriver().setup();

driver = new ChromeDriver();

driver.get("http://eliasnogueira.com");

}

static void quitBrowser()

{

driver.quit();

}

}

13. Use loggers.

import java.io.FileInputStream;

import java.io.IOException;

import java.util.logging.ConsoleHandler;

import java.util.logging.FileHandler;

import java.util.logging.Handler;

import java.util.logging.Level;

import java.util.logging.LogManager;

import java.util.logging.Logger;

public class LoggingExample

{

static Logger logger = Logger.getLogger(LoggingExample.class.getName());

public static void main (String[] args)

{

try

{

LogManager.getLogManager().readConfiguration(new FileInputStream("mylogging.properties"));

}

catch (SecurityException | IOException e1)

{

e1. printStackTrace();

}

logger.setLevel(Level.FINE);

logger.addHandler(new ConsoleHandler());

logger.addHandler(new MyHandler());

try

{

Handler fileHandler = new FileHandler("/Users/pankaj/tmp/logger.log", 2000, 5); fileHandler.setFormatter(new MyFormatter());

fileHandler.setFilter(new MyFilter()); logger.addHandler(fileHandler);

for (int i=0; i<1000; i++)

{

//logging messages logger.log (Level.INFO, "Msg"+i);

}

logger.log (Level.CONFIG, "Config data");

}

catch (SecurityException | IOException e)

{

e.printStackTrace();

}

}

}

14. Create a config file and pass the constant values from config file.

[APP]

ENVIRONMENT = test

DEBUG = True

# Only accept True or False

[DATABASE]

USERNAME = xiaoxu

PASSWORD = xiaoxu

HOST = 127.0.0.1

PORT = 5432

DB = xiaoxu\_database

15. Know how to debug the Scripts.

using System;

using System.Collections.Generic;

namespace ConsoleApp\_FirstApp

{

class Program

{

static void Main (string [] args)

{

Console.WriteLine("Welcome to Galaxy News!");

IterateThroughList();

Console.ReadKey();

}

private static void IterateThroughList()

{

var theGalaxies = new List<Galaxy>

{

new Galaxy () {Name="Tadpole", MegaLightYears=400, GalaxyType=new GType('S')},

new Galaxy () {Name="Pinwheel", MegaLightYears=25, GalaxyType=new GType('S')},

new Galaxy () {Name="Cartwheel", MegaLightYears=500, GalaxyType=new GType('L')},

new Galaxy () {Name="Small Magellanic Cloud", MegaLightYears=.2, GalaxyType=new GType('I')},

new Galaxy () {Name="Andromeda", MegaLightYears=3, GalaxyType=new GType('S')},

new Galaxy () {Name="Maffei 1", MegaLightYears=11, GalaxyType=new GType('E')}

};

foreach (Galaxy theGalaxy in theGalaxies)

{

Console.WriteLine(theGalaxy.Name + ““ + theGalaxy.MegaLightYears + ", " + theGalaxy.GalaxyType);

}

}

}

public class Galaxy

{

public string Name {get; set; }

public double MegaLightYears {get; set; }

public object GalaxyType {get; set; }

}

public class GType

{

public GType(char type)

{

switch(type)

{

case 'S':

MyGType = Type.Spiral;

break;

case 'E':

MyGType = Type.Elliptical;

break;

case 'l':

MyGType = Type.Irregular;

break;

case 'L':

MyGType = Type.Lenticular;

break;

default:

break;

}

}

public object MyGType {get; set; }

private enum Type {Spiral, Elliptical, Irregular, Lenticular}

}

}

16. Run the scripts in all browsers.

import org.openqa.Selenium.WebDriver;

importorg.openqa.Selenium.htmlunit.HtmlUnitDriver;

import org.testng.Assert;

import org.testng.annotations.Test;

publicclassvefifyTestTitle

{

publicvoidverifyFacebookTitle()

{

WebDriver driver = newHtmlUnitDriver(true);

driver.get("http://www.facebook.com");

String facebook\_Title= driver.getTitle();

Assert.assertTrue(facebook\_Title.contains("Facebook"));

System.out.println(facebook\_Title);

}

}

17. Parameterize the scripts using excel, Running the same script with multiple set of test data.

Import org.openqa.selenium.By;

import org.openqa.selenium.Keys;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.Reporter;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.BeforeMethod;

import org.testng.annotations.DataProvider;

import org.testng.annotations.Test;

public class SimpleTest

{

WebDriver driver;

@DataProvider(name = "test-data")

public Object[][] dataProvFunc(){

return new Object[][]{

{"Lambda Test"},{"Automation"}

};

}

public void setUp()

{

System.out.println("Start test");

System.setProperty("webdriver.chrome.driver", "E:\\chromedriver.exe");

driver = new ChromeDriver();

String url = "https://www.google.com";

driver.get(url);

driver.manage().window().maximize();

}

@Test annotation

@Test(dataProvider ="test-data")

public void search(String keyWord)

{

WebElement txtBox = driver.findElement(By.xpath("//input[@class='gLFyf gsfi']"));

txtBox.sendKeys(keyWord);

Reporter.log("Keyword entered is : " +keyWord);

txtBox.sendKeys(Keys.ENTER);

Reporter.log("Search results are displayed.");

}

public void burnDown()

{

driver.quit();

}

}

18. Parameterize the scripts using DataProvider.

import org.openqa.selenium.By;

import org.openqa.selenium.Keys;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.Reporter;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.BeforeMethod;

import org.testng.annotations.Test;

public class TestClass

{

WebDriver driver;

public void setUp()

{

System.out.println("Start test");

System.setProperty("webdriver.chrome.driver", "E:\\chromedriver.exe");

driver = new ChromeDriver();

String url = "https://www.google.com";

driver.get(url);

driver.manage().window().maximize();

}

@Test(dataProvider ="test-data", dataProviderClass=DPClass.class)

public void search(String keyWord)

{

WebElement txtBox = driver.findElement(By.xpath("//input[@class='gLFyf gsfi']"));

txtBox.sendKeys(keyWord);

Reporter.log("Keyword entered is : " +keyWord);

txtBox.sendKeys(Keys.ENTER);

Reporter.log("Search results are displayed.");

}

public void burnDown()

{

driver.quit();

}

}

20. Configure extent Report.

import org.junit.AfterClass;

import org.junit.BeforeClass;

import org.junit.Test;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import com.relevantcodes.extentreports.ExtentReports;

import com.relevantcodes.extentreports.ExtentTest;

import com.relevantcodes.extentreports.LogStatus;

public class ExtentDemo

{

static ExtentTest test;

static ExtentReports report;

public static void startTest()

{

report = new ExtentReports(System.getProperty("user.dir")+"ExtentReportResults.html");

test = report.startTest("ExtentDemo");

}

public void extentReportsDemo()

{

System.setProperty("webdriver.chrome.driver", "D:SubmittalExchange\_TFSQAAutomation3rdpartychromechromedriver.exe");

WebDriver driver = new ChromeDriver();

driver.get("https://www.google.co.in");

if(driver.getTitle().equals("Google"))

{

test.log(LogStatus.PASS, "Navigated to the specified URL");

}

else

{

test.log(LogStatus.FAIL, "Test Failed");

}

}

public static void endTest()

{

report.endTest(test);

report.flush();

}

}