

```
1 using Microsoft.VisualStudio.TestTools.UnitTesting;
2 using OpenQA.Selenium;
3 using OpenQA.Selenium.Chrome;
4 using System;
5 using System.Collections.Generic;
6 using System.Net;
7 using System.Net.Security;
8 using System.Threading.Tasks;
9 using System.Security.Cryptography.X509Certificates;
10
11 namespace Selenium_assignment
12 {
13     [TestClass]
14     public class UnitTest1
15     {
16         [TestMethod]
17         public void question1()
18         {
19             IWebDriver driver = new ChromeDriver();//creating a chrome driver
20             driver.Navigate().GoToUrl("http://www.google.co.nz");//open
21             driver.FindElement(By.Name("q")).SendKeys("Taupo weather");// input
22             driver.FindElement(By.Name("q")).SendKeys(Keys.Enter);//search
23         }
24         [TestMethod]
25         public void question2()
26         {
27             IWebDriver driver = new ChromeDriver();
28             driver.Navigate().GoToUrl("https://www.trademe.co.nz/a/");// open
29             driver.FindElement(By.Id("search")).SendKeys("IT jobs");// input
30             driver.FindElement(By.Id("search")).SendKeys(Keys.Enter);// search
31         }
32         //[TestMethod]
33         //public void question3()
34         //{
35
36             // int valid_links = 0, broken_links = 0;
37             // IWebDriver driver = new ChromeDriver();
38             // driver.Url = "https://www.lambdatest.com/blog";
39             // using var client = new HttpClient();
40             // var links = driver.FindElements(By.TagName("a"));
41
42             // Console.WriteLine("Looking at all the URLs in LambdaTest :");
43             // /* Loop through all the urls */
```

```
44 //     foreach (var link in links)
45 //     {
46 //         if (!(link.Text.Contains("Email") || link.Text.Contains
47 //             ("https://www.linkedin.com") || link.Text == "" || link.Equals
48 //             (null)))
49 //         {
50 //             /* Get the URI */
51 //             HttpResponseMessage response = await
52 //             client.GetAsync(link.GetAttribute("href"));
53 //             System.Console.WriteLine($"URL:
54 //             {link.GetAttribute("href")} status is :{response.StatusCode}");
55 //             /* Reference - https://docs.microsoft.com/en-us/
56 //             dotnet/api/system.net.httpwebresponse.statuscode?
57 //             view=netcore-3.1 */
58 //             if (response.StatusCode == HttpStatusCode.OK)
59 //             {
60 //                 valid_links++;
61 //             }
62 //             else
63 //             {
64 //                 broken_links++;
65 //             }
66 //             catch (Exception ex)
67 //             {
68 //                 if ((ex is ArgumentNullException) ||
69 //                     (ex is NotSupportedException))
70 //                 {
71 //                     System.Console.WriteLine("Exception occured
72 //                     \n");
73 //                 }
74 //             }
75 //         }
76 //     }
77 //     /* Perform wait to check the output */
78 //     System.Threading.Thread.Sleep(2000);
79 //     Console.WriteLine("Detection of broken links completed with
80 //     " + broken_links + " broken links and " + valid_links + " valid
81 //     links");
82 //
83 // //IWebDriver driver = new ChromeDriver();
84 // ////driver.Navigate().GoToUrl("http://
85 // automationpractice.com");//open
86 // //driver.Navigate().GoToUrl("http://www.google.co.nz");
```

```
83
84     //    //IList<IWebElement> links = driver.FindElements(By.TagName
      ("a"));
85     //    //foreach (IWebElement link in links)
86     //    //{
87     //        Console.WriteLine(link.GetAttribute("herf"));
88     //        //var url = link.GetAttribute("href");
89     //        //IsLinkWorking(url);
90     //    }
91     //    //Console.Read();
92     //}
93
94
95
96     //bool IsLinkWorking(string url)
97     //{
98     //    HttpWebRequest request = (HttpWebRequest)
      HttpWebRequest.Create(url);
99
100    //    //You can set some parameters in the "request" object...
101    //    request.AllowAutoRedirect = true;
102
103    //    try
104    //    {
105    //        HttpResponseMessage response = (HttpResponse)
      request.GetResponse();
106    //        if (response.StatusCode == HttpStatusCode.OK)
107    //        {
108    //            Console.WriteLine("\r\nResponse Status Code is OK
      and StatusDescription is: { 0}", response.StatusDescription);
109    //            // Releases the resources of the response.
110    //            response.Close();
111    //            return true;
112    //        }
113    //        else
114    //        {
115    //            return false;
116    //        }
117    //    }
118    //    catch
119    //    { //TODO: Check for the right exception here
120    //        return false;
121    //    }
122    //}
123
124 }
125 }
126
```