Automate the Web Application

--- Mohammad Hasnain Rajan

Steps to run the project

1. Clone the repository
2. Import In Eclipse
3. Install TestNg in Eclipse
4. Do maven Update
5. Open testing.xml
6. Update the driverLink system variable with the link to chromedriver in your system.
7. Right click and run as TestNg Suite.
8. The automated tests will test the website and write the results.
9. You can open emailable-report.html to view the results of the tests

Please Note:

- The register test uses a email to register on the website. So it will fail if the test is run the second time. In that case a proper error will be shown in the report so edit the emailAddrRegister parameter to a new email address.
- Please do not edit the emailAddrLogin, password, invPassword parameters. Editing them may lead to failure of tests. This is because I have created acoount with these specific details. Editing them will lead to failed login.
- If you get time out errors, reduce the thread-count according to the system you are running tests on

Steps To Complete the Project:

It involved writing test for the following use cases:

1. Steps for TestLogin :
    a. Start Chromedriver
    b. Go to Url http://automationpractice.com/
    c. Wait for the sign button to load
    d. Click on Sign in Button
    e. Wait for the page to load
    f. Click on email textbox
    g. Enter Email Id
    h. Click on password text box
    i. Enter Password
    j. Click on Sign In
    k. Wait for the Page to load
    l. Check the top bar for the registered name to verify successful login

2. Steps for TestInvalidLogin
    a. Start Chromedriver
    b. Go to Url http://automationpractice.com/
    c. Wait for the sign button to load
    d. Click on Sign in Button
    e. Wait for the page to load
    f. Click on email textbox
    g. Enter wrong Email Id
    h. Click on password text box
    i. Enter wrong Password
    j. Click on Sign In
    k. Wait for the Page to load
    l. Check the error message that authentication failed

3. Steps for TestLogout
    a. Click on Sign out Button
    b. Wait for the page to load
    c. Check the top bar for “Sign in” to verify successful logout

4. Steps for TestRegister
    a. Start Chromedriver
    b. Go to Url http://automationpractice.com/
    c. Wait for the sign button to load
    d. Click on Sign in Button
    e. Wait for the page to load
    f. Click on email textbox
    g. Enter new Email Id
    h. Click on Create new Account
    i. Throw exception if email Id already used
    j. Wait for gender radio button to become available
    k. Select Male
    l. Click on input controls to enter First name (By Id), Last Name (By Id), Password (By Id), birthdate (By xpath), Birth Month (By xpath), Birth year(By xpath), address First Name (By Id), Address Last Name (By Id), Address Line 1 (By Id), Address Line 2 (By Id), City (By Id), State (By xpath), postcode (By Id), country (By Id), phone number (By Id), mobile number (By Id), address alias (By Id)
    m. Click on the register button
    n. Wait for the page to load
    o. Verify name displayed in top bar to verify successful register

5. Steps for TestSearch
    a. Start Chromedriver
    b. Go to Url http://automationpractice.com/
    c. Wait for the page to load
    d. Click on search textbox
    e. Enter search query
    f. Press search button

g. Wait for the page to load
h. Assert name of first Product to be same as search query to verify correct search

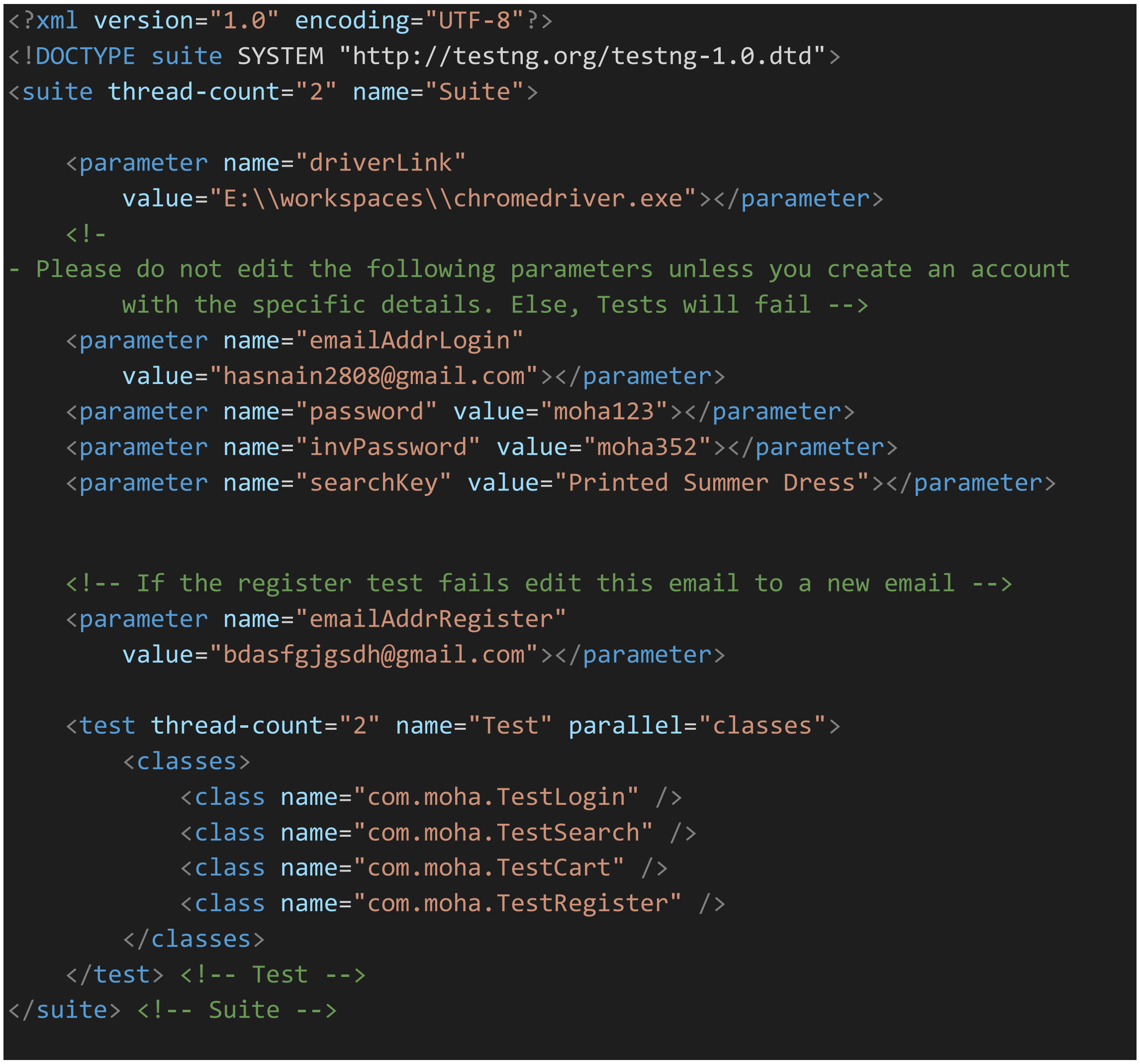
6. Steps for TestCart
    a. Start Chromedriver
    b. Go to Url http://automationpractice.com/
    c. Click on Sign in Button
    d. Wait for the page to load
    e. Click on email textbox
    f. Enter Email Id
    g. Click on password text box
    h. Enter Password
    i. Click on Sign In
    j. Wait for the Page to load
    k. Hover on a product
    l. Copy name of the product
    m. Press add to cart
    n. Check successfully added text to verify
    o. Click on cart
    p. Wait for the page to load
    q. Check the name of the product in the cart to verify correct cart added

Code:

E:\workspaces\sel_prac_bnp_ecl\TestAutomation\pom.xml

```xml
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.moha</groupId>
  <artifactId>com.moha</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>TestAutomation</name>
  <description>Test automationpractice[dot]com</description>
    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <maven.compiler.source>1.7</maven.compiler.source>
        <maven.compiler.target>1.7</maven.compiler.target>
    </properties>
    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>4.11</version>
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>org.seleniumhq.selenium</groupId>
            <artifactId>selenium-java</artifactId>
            <version>4.0.0-alpha-6</version>
        </dependency>
        <dependency>
            <groupId>org.slf4j</groupId>
            <artifactId>slf4j-api</artifactId>
            <version>1.7.5</version>
        </dependency>
        <dependency>
            <groupId>org.slf4j</groupId>
            <artifactId>slf4j-log4j12</artifactId>
            <version>1.7.5</version>
        </dependency>
    </dependencies>
</project>
```

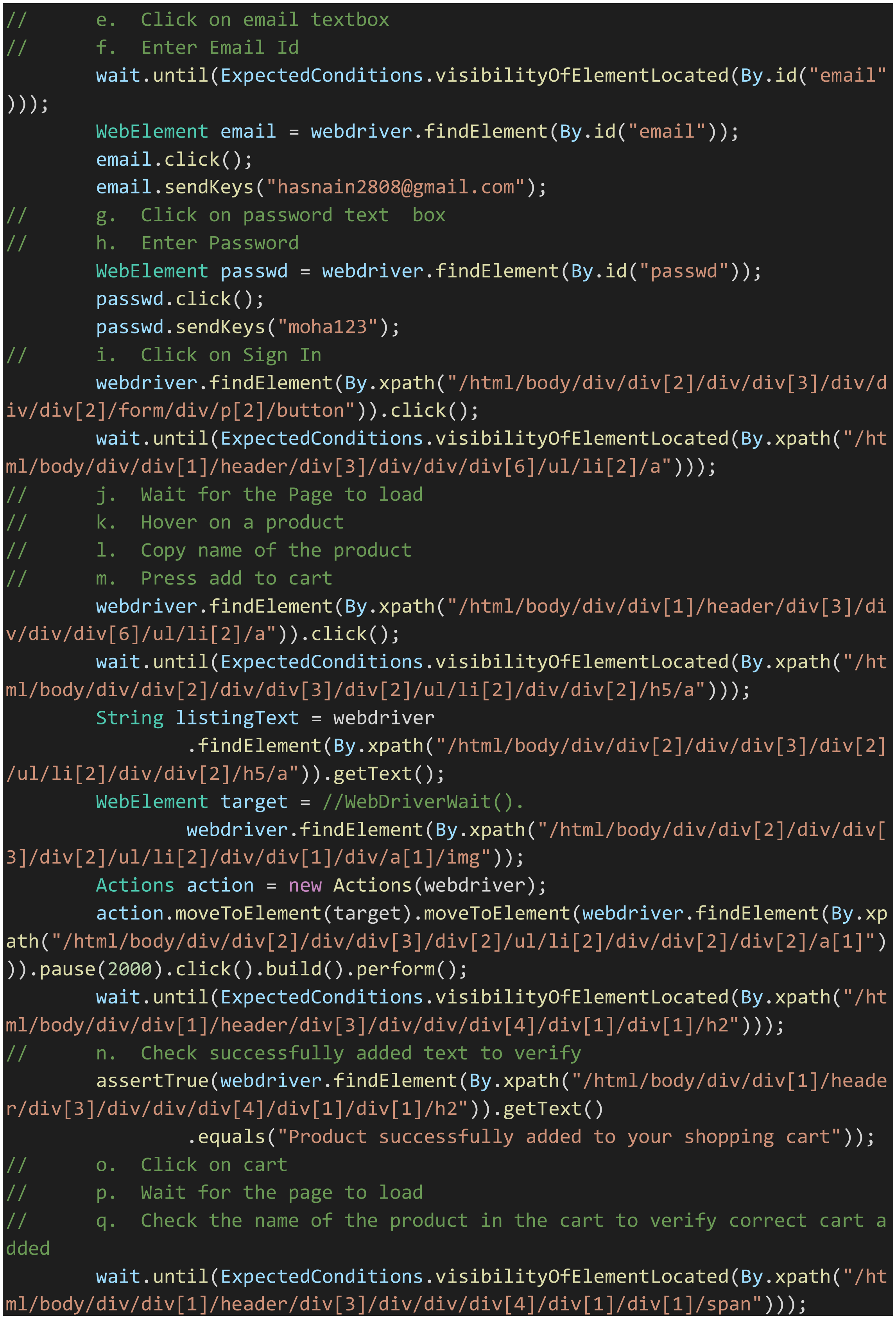
E:\workspaces\sel_prac_bnp_ecl\TestAutomation\testng.xml



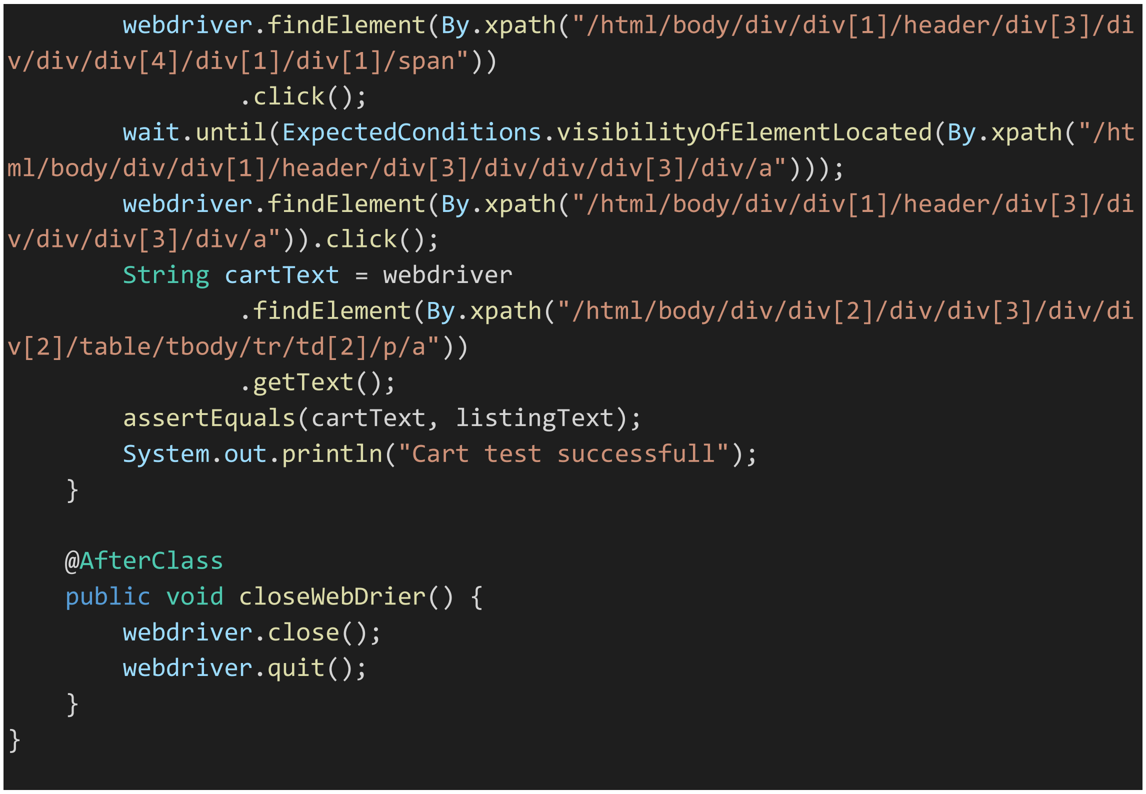
```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite thread-count="2" name="Suite">

    <parameter name="driverLink"
        value="E:\\workspaces\\chromedriver.exe"></parameter>
    <!-
- Please do not edit the following parameters unless you create an account
        with the specific details. Else, Tests will fail -->
    <parameter name="emailAddrLogin"
        value="hasnain2808@gmail.com"></parameter>
    <parameter name="password" value="moha123"></parameter>
    <parameter name="invPassword" value="moha352"></parameter>
    <parameter name="searchKey" value="Printed Summer Dress"></parameter>


    <!-- If the register test fails edit this email to a new email -->
    <parameter name="emailAddrRegister"
        value="bdasfgjgsdh@gmail.com"></parameter>

    <test thread-count="2" name="Test" parallel="classes">
        <classes>
            <class name="com.moha.TestLogin" />
            <class name="com.moha.TestSearch" />
            <class name="com.moha.TestCart" />
            <class name="com.moha.TestRegister" />
        </classes>
    </test> <!-- Test -->
</suite> <!-- Suite -->
```

E:\workspaces\sel_prac_bnp_ecl\TestAutomation\src\test\java\com\moha\TestCart.java

```
package com.moha;

import static org.testng.Assert.assertEquals;
import static org.testng.Assert.assertTrue;

import java.time.Duration;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.interactions.Actions;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
import org.testng.annotations.AfterClass;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.Parameters;
import org.testng.annotations.Test;

public class TestCart {

    private WebDriver webdriver;
    private Duration timeout;
    private WebDriverWait wait;
    @BeforeClass
    @Parameters({"driverLink"})
    public void createWebDriver(String driverLink) {
        System.setProperty("webdriver.chrome.driver", driverLink);
//      a.  Start Chromedriver
        webdriver = new ChromeDriver();
        timeout = Duration.ofSeconds(5);
         wait = new WebDriverWait(webdriver, timeout );
    }
    @Test
    public void testCart() {
//      b.  Go to Url http://automationpractice.com/
        String actualUrl = "http://automationpractice.com/";
        webdriver.get(actualUrl);
//      c.  Click on Sign in Button
        wait.until(ExpectedConditions.visibilityOfElementLocated(By.linkText("Sign in")));
        webdriver.findElement(By.linkText("Sign in")).click();
//      d.  Wait for the page to load
```



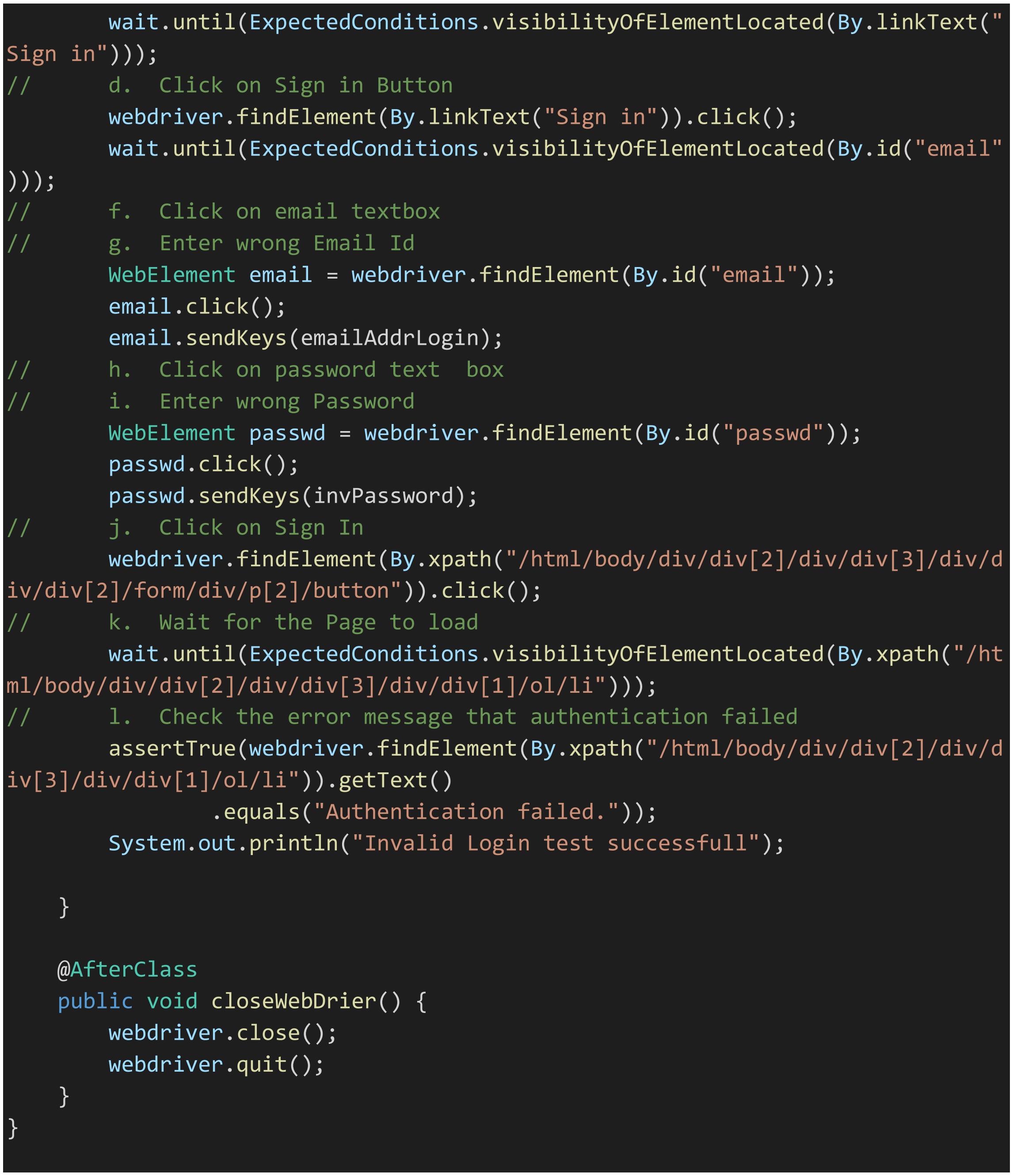
```
//      e.  Click on email textbox
//      f.  Enter Email Id
        wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("email"
)));
        WebElement email = webdriver.findElement(By.id("email"));
        email.click();
        email.sendKeys("hasnain2808@gmail.com");
//      g.  Click on password text  box
//      h.  Enter Password
        WebElement passwd = webdriver.findElement(By.id("passwd"));
        passwd.click();
        passwd.sendKeys("moha123");
//      i.  Click on Sign In
        webdriver.findElement(By.xpath("/html/body/div/div[2]/div/div[3]/div/d
iv/div[2]/form/div/p[2]/button")).click();
        wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("/ht
ml/body/div/div[1]/header/div[3]/div/div/div[6]/ul/li[2]/a")));
//      j.  Wait for the Page to load
//      k.  Hover on a product
//      l.  Copy name of the product
//      m.  Press add to cart
        webdriver.findElement(By.xpath("/html/body/div/div[1]/header/div[3]/di
v/div/div[6]/ul/li[2]/a")).click();
        wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("/ht
ml/body/div/div[2]/div/div[3]/div[2]/ul/li[2]/div/div[2]/h5/a")));
        String listingText = webdriver
                .findElement(By.xpath("/html/body/div/div[2]/div/div[3]/div[2]
/ul/li[2]/div/div[2]/h5/a")).getText();
        WebElement target = //WebDriverWait().
                webdriver.findElement(By.xpath("/html/body/div/div[2]/div/div[
3]/div[2]/ul/li[2]/div/div[1]/div/a[1]/img"));
        Actions action = new Actions(webdriver);
        action.moveToElement(target).moveToElement(webdriver.findElement(By.xp
ath("/html/body/div/div[2]/div/div[3]/div[2]/ul/li[2]/div/div[2]/div[2]/a[1]")
)).pause(2000).click().build().perform();
        wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("/ht
ml/body/div/div[1]/header/div[3]/div/div/div[4]/div[1]/div[1]/h2")));
//      n.  Check successfully added text to verify
        assertTrue(webdriver.findElement(By.xpath("/html/body/div/div[1]/heade
r/div[3]/div/div/div[4]/div[1]/div[1]/h2")).getText()
                .equals("Product successfully added to your shopping cart"));
//      o.  Click on cart
//      p.  Wait for the page to load
//      q.  Check the name of the product in the cart to verify correct cart a
dded
        wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("/ht
ml/body/div/div[1]/header/div[3]/div/div/div[4]/div[1]/div[1]/span")));
```



```
        webdriver.findElement(By.xpath("/html/body/div/div[1]/header/div[3]/div/div/div[4]/div[1]/div[1]/span"))
                .click();
        wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div/div[1]/header/div[3]/div/div/div[3]/div/a")));
        webdriver.findElement(By.xpath("/html/body/div/div[1]/header/div[3]/div/div/div[3]/div/a")).click();
        String cartText = webdriver
                .findElement(By.xpath("/html/body/div/div[2]/div/div[3]/div/div[2]/table/tbody/tr/td[2]/p/a"))
                .getText();
        assertEquals(cartText, listingText);
        System.out.println("Cart test successfull");
    }

    @AfterClass
    public void closeWebDrier() {
        webdriver.close();
        webdriver.quit();
    }
}
```

E:\workspaces\sel_prac_bnp_ecl\TestAutomation\src\test\java\com\moha\TestLogin.java

```java
package com.moha;

import static org.testng.Assert.assertEquals;
import static org.testng.Assert.assertTrue;

import java.time.Duration;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
import org.testng.annotations.AfterClass;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.Parameters;
import org.testng.annotations.Test;

public class TestLogin {

    private WebDriver webdriver;
    private Duration timeout;
    private WebDriverWait wait;
    @BeforeClass
    @Parameters({"driverLink"})
    public void createWebDriver(String driverLink) {
//      a.  Start Chromedriver
        System.setProperty("webdriver.chrome.driver", driverLink);      webdriver = new ChromeDriver();
        timeout = Duration.ofSeconds(5);
         wait = new WebDriverWait(webdriver, timeout );
    }

    @Test(priority = 2)
    @Parameters({"emailAddrLogin", "password"})
    public void testValidLogin(String emailAddrLogin, String password) {
//      b.  Go to Url http://automationpractice.com/
        String actualUrl = "http://automationpractice.com/";
        webdriver.get(actualUrl);
//      c.  Wait for the sign button to load
//      d.  Click on Sign in Button
        wait.until(ExpectedConditions.visibilityOfElementLocated(By.linkText("Sign in")));
        webdriver.findElement(By.linkText("Sign in")).click();
        wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("email")));
//      f.  Click on email textbox
//      g.  Enter Email Id
        WebElement email = webdriver.findElement(By.id("email"));
        email.click();
        email.sendKeys(emailAddrLogin);
//      h.  Click on password text  box
//      i.  Enter Password
        WebElement passwd = webdriver.findElement(By.id("passwd"));
        passwd.click();
        passwd.sendKeys(password);
//      j.  Click on Sign In
        webdriver.findElement(By.xpath("/html/body/div/div[2]/div/div[3]/div/div/div[2]/form/div/p[2]/button")).click();
//      l.  Check the top bar for the registered name to verify successful log in
        assertTrue(webdriver.findElement(By.xpath("/html/body/div/div[1]/header/div[2]/div/div/nav/div[1]/a/span"))
                .getText().equals("Mohammad Hasnain Rajan"));
        System.out.println("Valid Login test successfull");
    }


    @Test(priority = 3)
    public void testLogOut() {
//      a.  Click on Sign out Button
        wait.until(ExpectedConditions.visibilityOfElementLocated(By.linkText("Sign out")));
        webdriver.findElement(By.linkText("Sign out")).click();
//      b.  Wait for the page to load
        wait.until(ExpectedConditions.visibilityOfElementLocated(By.linkText("Sign in")));
//      c.  Check the top bar for “Sign in” to verify successful logout
        String signin = webdriver.findElement(By.linkText("Sign in")).getText();
        assertEquals("Sign in", signin);
        System.out.println("Test Logout successfull");
    }

    @Test(priority = 1)
    @Parameters({"emailAddrLogin", "invPassword"})
    public void testInvalidLogin(String emailAddrLogin, String invPassword) {
//      b.  Go to Url http://automationpractice.com/
        String actualUrl = "http://automationpractice.com/";
        webdriver.get(actualUrl);
//      c.  Wait for the sign button to load
```



```
        wait.until(ExpectedConditions.visibilityOfElementLocated(By.linkText("
Sign in")));
//      d.  Click on Sign in Button
        webdriver.findElement(By.linkText("Sign in")).click();
        wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("email"
)));
//      f.  Click on email textbox
//      g.  Enter wrong Email Id
        WebElement email = webdriver.findElement(By.id("email"));
        email.click();
        email.sendKeys(emailAddrLogin);
//      h.  Click on password text  box
//      i.  Enter wrong Password
        WebElement passwd = webdriver.findElement(By.id("passwd"));
        passwd.click();
        passwd.sendKeys(invPassword);
//      j.  Click on Sign In
        webdriver.findElement(By.xpath("/html/body/div/div[2]/div/div[3]/div/d
iv/div[2]/form/div/p[2]/button")).click();
//      k.  Wait for the Page to load
        wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("/ht
ml/body/div/div[2]/div/div[3]/div/div[1]/ol/li")));
//      l.  Check the error message that authentication failed
        assertTrue(webdriver.findElement(By.xpath("/html/body/div/div[2]/div/d
iv[3]/div/div[1]/ol/li")).getText()
                .equals("Authentication failed."));
        System.out.println("Invalid Login test successfull");

    }

    @AfterClass
    public void closeWebDrier() {
        webdriver.close();
        webdriver.quit();
    }
}
```

E:\workspaces\sel_prac_bnp_ecl\TestAutomation\src\test\java\com\moha\TestRegister.java

```java
package com.moha;

import static org.testng.Assert.assertTrue;

import java.time.Duration;

import org.openqa.selenium.By;
import org.openqa.selenium.TimeoutException;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
import org.testng.annotations.AfterClass;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.Parameters;
import org.testng.annotations.Test;

public class TestRegister {

    private WebDriver webdriver;
    private Duration timeout;
    private WebDriverWait wait;

    @BeforeClass
    @Parameters({"driverLink"})
    public void createWebDriver(String driverLink) {
        System.setProperty("webdriver.chrome.driver", driverLink);
//      a.  Start Chromedriver
        webdriver = new ChromeDriver();
        timeout = Duration.ofSeconds(5);
        wait = new WebDriverWait(webdriver, timeout);
    }

    @Test
    @Parameters({ "emailAddrRegister" })
    public void testRegister(String emailAddrRegister) {
//      b.  Go to Url http://automationpractice.com/
        String actualUrl = "http://automationpractice.com/";
        webdriver.get(actualUrl);
//      c.  Wait for the sign button to load
//      d.  Click on Sign in Button
        webdriver.findElement(By.linkText("Sign in")).click();
//      f.  Click on email textbox
//      g.  Enter new Email Id
```

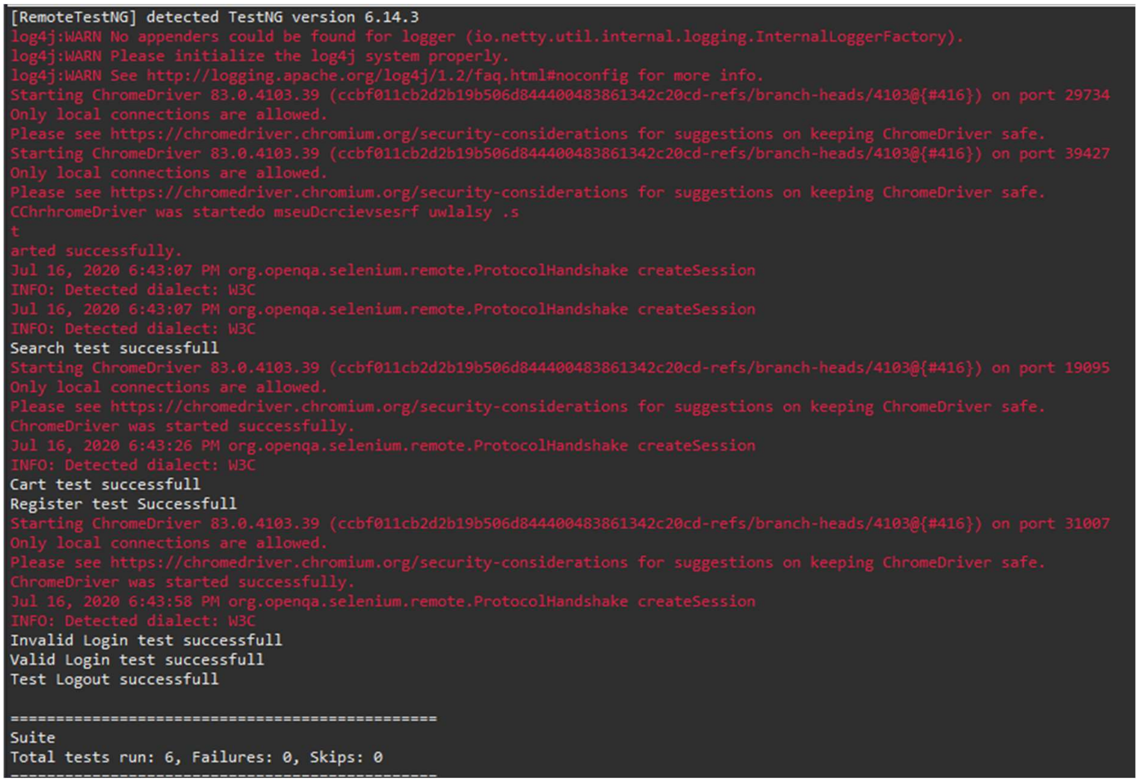
```
        wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("email_create")));
        WebElement email = webdriver.findElement(By.id("email_create"));
        email.click();
        email.sendKeys(emailAddrRegister);
//      h.  Click on Create new Account
        webdriver.findElement(By.id("SubmitCreate")).click();
//      i.  Throw exception if email Id already used
//      j.  Wait for gender radio button to become available
        try {
            wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("id_gender1")));

        } catch (TimeoutException e) {
            assertTrue(false, "Please use a different email Address, Account with this email id exists");
        }
//      k.  Select Male
        webdriver.findElement(By.id("id_gender1")).click();
//      l.  Click on input controls to enter First name (By Id), Last Name (By Id), Password (By Id), birthdate (By xpath), Birth Month (By xpath), Birth year(By xpath), address First Name (By Id), Address Last Name (By Id), Address Line 1 (By Id), Address Line 2 (By Id), City (By Id), State (By xpath), postcode (By Id), country (By Id), phone number (By Id), mobile number (By Id), address alias (By Id)
        WebElement firstName = webdriver.findElement(By.id("customer_firstname"));
        firstName.click();
        firstName.sendKeys("Babu");
        WebElement lastName = webdriver.findElement(By.id("customer_lastname"));
        lastName.click();
        lastName.sendKeys("Sharma");
        WebElement passwd = webdriver.findElement(By.id("passwd"));
        passwd.click();
        passwd.sendKeys("babu123");
        WebElement days = webdriver.findElement(By.id("days"));
        days.click();
        webdriver.findElement(By.xpath("//*[@id=\"days\"]/option[2]")).click();
        WebElement months = webdriver.findElement(By.id("months"));
        months.click();
        webdriver.findElement(By.xpath("//*[@id=\"months\"]/option[2]")).click();
        WebElement years = webdriver.findElement(By.id("years"));
        years.click();
        webdriver.findElement(By.xpath("//*[@id=\"years\"]/option[2]")).click();
        WebElement afirstName = webdriver.findElement(By.id("firstname"));
        afirstName.click();
        afirstName.sendKeys("Babu");
        WebElement alastName = webdriver.findElement(By.id("lastname"));
        alastName.click();
        alastName.sendKeys("Sharma");
        WebElement addr1 = webdriver.findElement(By.id("address1"));
        addr1.click();
        addr1.sendKeys("101, rahul complex");
        WebElement address2 = webdriver.findElement(By.id("address2"));
        address2.click();
        address2.sendKeys("raje Marg");
        WebElement city = webdriver.findElement(By.id("city"));
        city.click();
        city.sendKeys("Mumbai");
        WebElement id_state = webdriver.findElement(By.id("id_state"));
        id_state.click();
        webdriver.findElement(By.xpath("//*[@id=\"id_state\"]/option[12]")).cl
ick();
        WebElement postcode = webdriver.findElement(By.id("postcode"));
        postcode.click();
        postcode.sendKeys("00000");
        WebElement id_country = webdriver.findElement(By.id("id_country"));
        id_country.click();
        webdriver.findElement(By.xpath("//*[@id=\"id_country\"]/option[2]")).c
lick();
        WebElement phone = webdriver.findElement(By.id("phone"));
        phone.click();
        phone.sendKeys("9867519861");
        WebElement phone_mobile = webdriver.findElement(By.id("phone_mobile"))
;
        phone_mobile.click();
        phone_mobile.sendKeys("9867519261");
        WebElement alias = webdriver.findElement(By.id("alias"));
        alias.click();
        alias.sendKeys("address alias");
//      m.  Click on the register button
        wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("submit
Account")));
        webdriver.findElement(By.id("submitAccount")).click();
//      n.  Wait for the page to load
//      o.  Verify name displayed in top bar to verify successful register
        assertTrue(webdriver.findElement(By.xpath("//*[@id=\"header\"]/div[2]/
div/div/nav/div[1]/a/span")).getText()
                .equals("Babu Sharma"));
        System.out.println("Register test Successfull");
    }
    @AfterClass
    public void closeWebDrier() {
        webdriver.close();
        webdriver.quit();
    }
}
```

E:\workspaces\sel_prac_bnp_ecl\TestAutomation\src\test\java\com\moha\TestSearch.java

```
package com.moha;

import static org.testng.Assert.assertEquals;

import java.time.Duration;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
import org.testng.annotations.AfterClass;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.Parameters;
import org.testng.annotations.Test;

public class TestSearch {

    private WebDriver webdriver;
    private Duration timeout;
    private WebDriverWait wait;

    @BeforeClass
    @Parameters({"driverLink"})
    public void createWebDriver(String driverLink) {
        System.setProperty("webdriver.chrome.driver", driverLink);
//      a.  Start Chromedriver
        webdriver = new ChromeDriver();
        timeout = Duration.ofSeconds(5);
        wait = new WebDriverWait(webdriver, timeout);
    }

    @Test
    @Parameters({ "searchKey" })
    public void testSearch(String searchKey) {
```

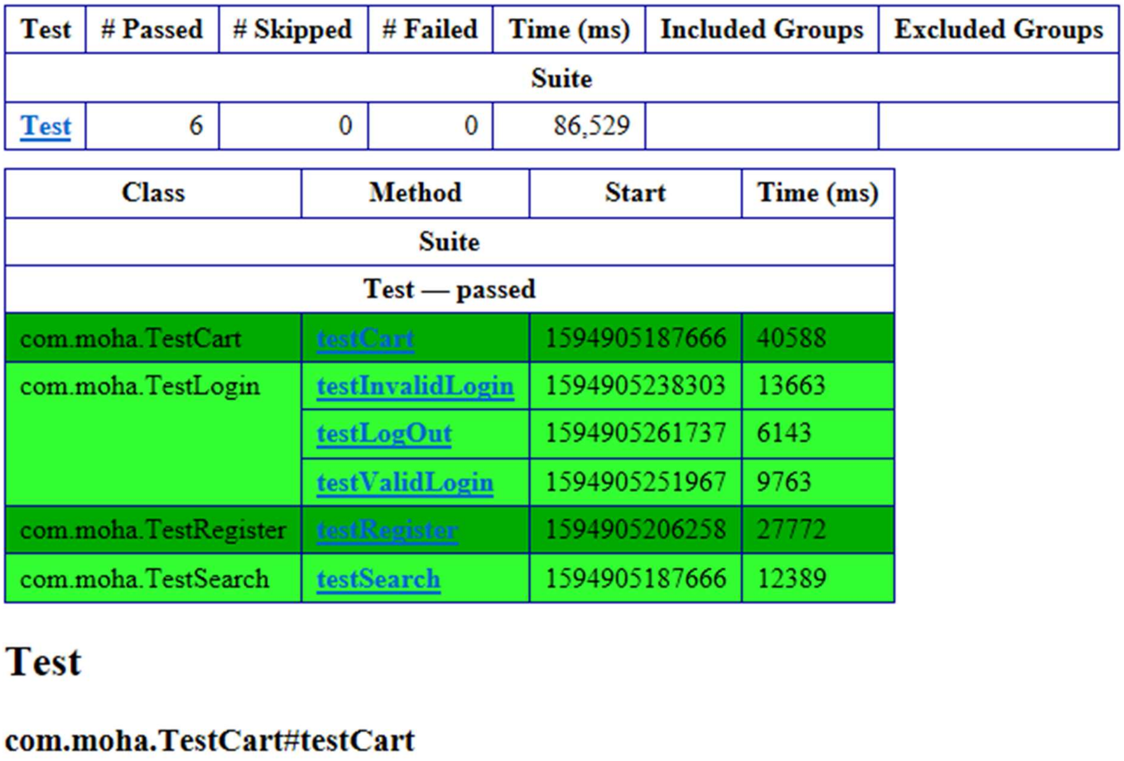
```
//      b.  Go to Url http://automationpractice.com/
        String actualUrl = "http://automationpractice.com/";
        webdriver.get(actualUrl);
//      d.  Click on search textbox
//      e.  Enter search query
        wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("search
_query_top")));
        String searchQuery = searchKey;
        WebElement search_query_top = webdriver.findElement(By.id("search_quer
y_top"));
        search_query_top.click();
        search_query_top.sendKeys(searchQuery);
//      f.  Press search button
        webdriver.findElement(By.xpath("/html/body/div/div[1]/header/div[3]/di
v/div/div[2]/form/button")).click();
//      g.  Wait for the page to load
//      h.  Assert name of first Product to be same as search query to verify
correct search
        wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("/ht
ml/body/div/div[2]/div/div[3]/div[2]/h1/span[1]")));
        String fromSearch = webdriver.findElement(By.xpath("/html/body/div/div
[2]/div/div[3]/div[2]/h1/span[1]"))
                .getText();
        assertEquals("\"" + searchQuery.toUpperCase() + "\"", fromSearch);
        String fromResults = webdriver
                .findElement(By.xpath("/html/body/div/div[2]/div/div[3]/div[2]
/ul/li[1]/div/div[2]/h5/a")).getText();
        assertEquals(fromResults, searchQuery);
        System.out.println("Search test successfull");
    }

    @AfterClass
    public void closeWebDrier() {
        webdriver.close();
        webdriver.quit();
    }
}
```

Terminal Output:



```
[RemoteTestNG] detected TestNG version 6.14.3
log4j:WARN No appenders could be found for logger (io.netty.util.internal.logging.InternalLoggerFactory).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 29734
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 39427
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
CChrhromeDriver was startedo mseuDcrcievsesrf uwlalsy .s
t
arted successfully.
Jul 16, 2020 6:43:07 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Jul 16, 2020 6:43:07 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Search test successfull
Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 19095
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Jul 16, 2020 6:43:26 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Cart test successfull
Register test Successfull
Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 31007
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Jul 16, 2020 6:43:58 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Invalid Login test successfull
Valid Login test successfull
Test Logout successfull

===============================================
Suite
Total tests run: 6, Failures: 0, Skips: 0
===============================================
```

emailable-report.html

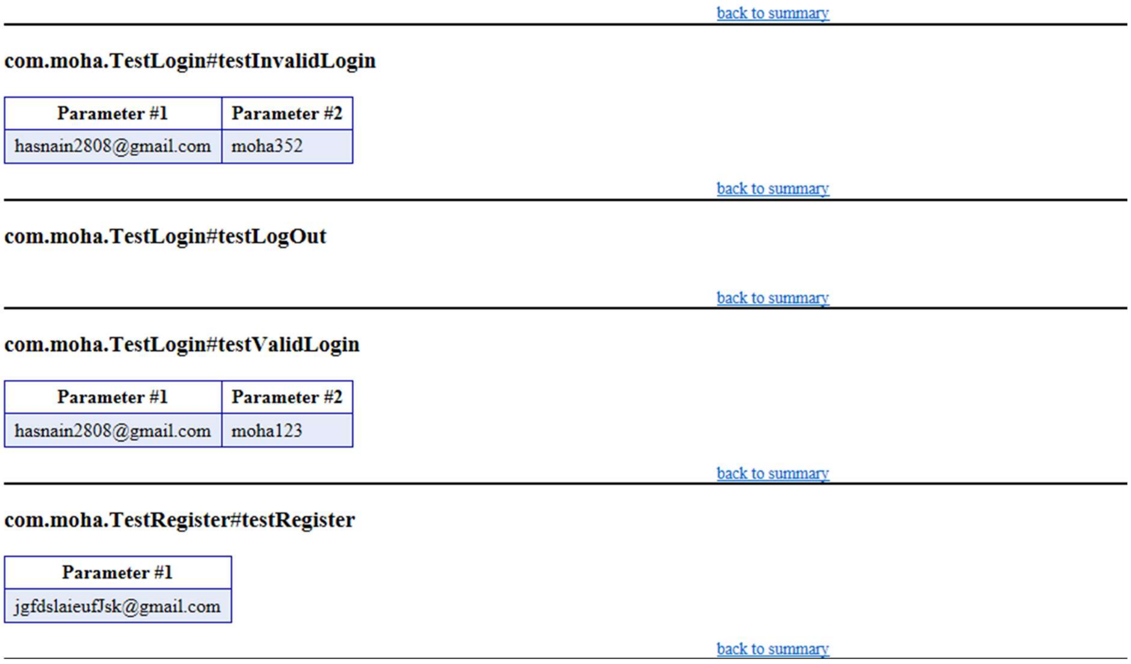


Test	# Passed	# Skipped	# Failed	Time (ms)	Included Groups	Excluded Groups
Suite						
Test	6	0	0	86,529		

Class	Method	Start	Time (ms)
Suite			
Test — passed			
com.moha.TestCart	testCart	1594905187666	40588
com.moha.TestLogin	testInvalidLogin	1594905238303	13663
	testLogOut	1594905261737	6143
	testValidLogin	1594905251967	9763
com.moha.TestRegister	testRegister	1594905206258	27772
com.moha.TestSearch	testSearch	1594905187666	12389

Test

com.moha.TestCart#testCart



back to summary

com.moha.TestLogin#testInvalidLogin

Parameter #1	Parameter #2
hasnain2808@gmail.com	moha352

back to summary

com.moha.TestLogin#testLogOut

back to summary

com.moha.TestLogin#testValidLogin

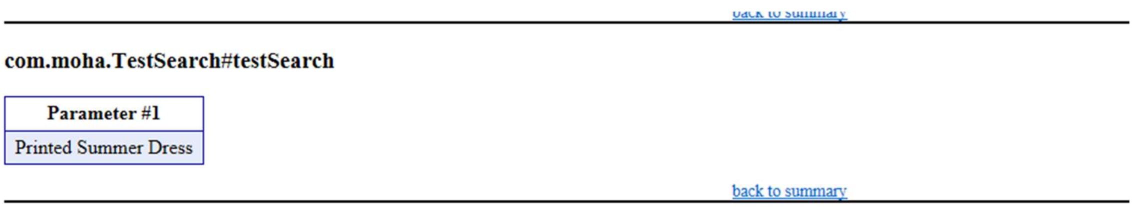
Parameter #1	Parameter #2
hasnain2808@gmail.com	moha123

back to summary

com.moha.TestRegister#testRegister

Parameter #1
jgfdslaieufJsk@gmail.com

back to summary



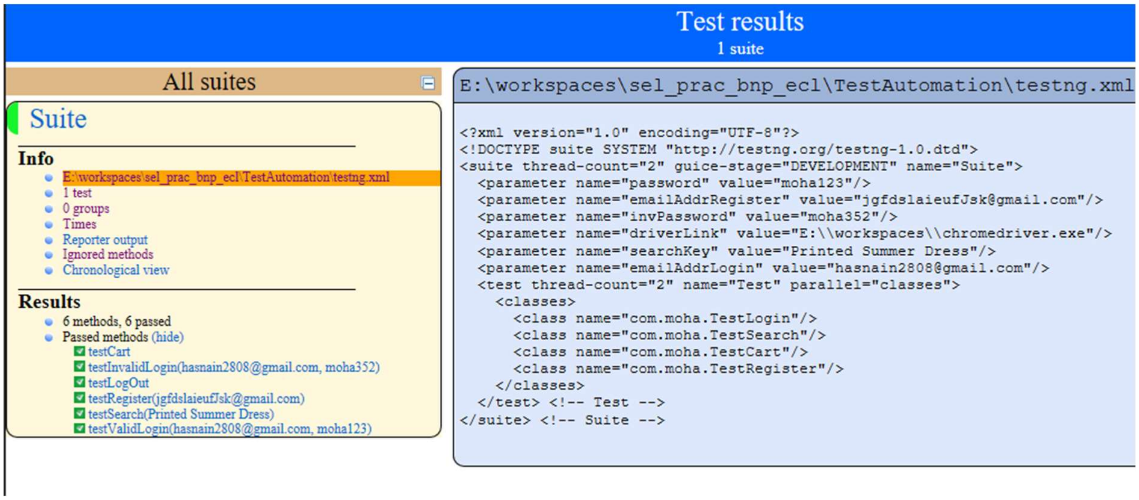
back to summary

com.moha.TestSearch#testSearch

Parameter #1
Printed Summer Dress

back to summary

Full Report



Test results
1 suite

All suites

Suite

Info

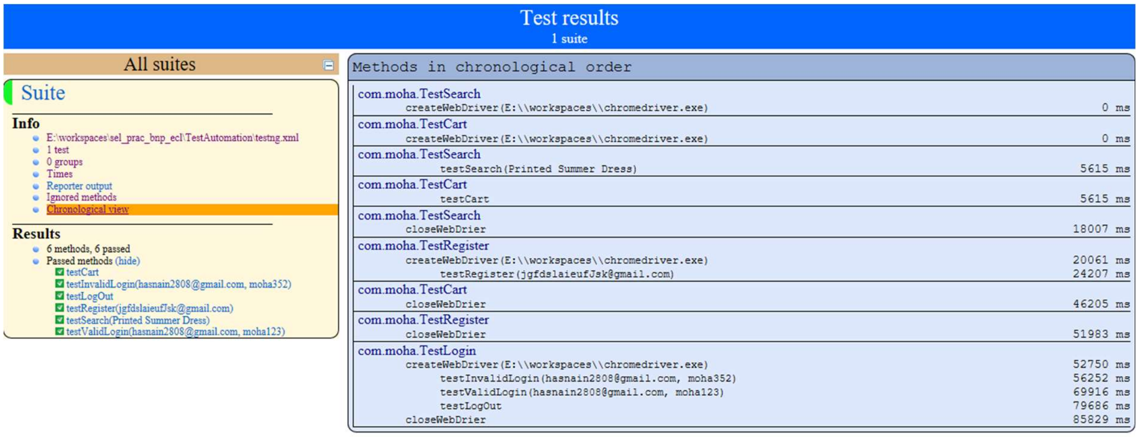
- E:\workspaces\sel_prac_bnp_ecl\TestAutomation\testng.xml
- 1 test
- 0 groups
- Times
- Reporter output
- Ignored methods
- Chronological view

Results

- 6 methods, 6 passed
- Passed methods (hide)
  - ☑ testCart
  - ☑ testInvalidLogin(hasnain2808@gmail.com, moha352)
  - ☑ testLogOut
  - ☑ testRegister(jgfdslaieufJsk@gmail.com)
  - ☑ testSearch(Printed Summer Dress)
  - ☑ testValidLogin(hasnain2808@gmail.com, moha123)

E:\workspaces\sel_prac_bnp_ecl\TestAutomation\testng.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite thread-count="2" guice-stage="DEVELOPMENT" name="Suite">
  <parameter name="password" value="moha123"/>
  <parameter name="emailAddrRegister" value="jgfdslaieufJsk@gmail.com"/>
  <parameter name="invPassword" value="moha352"/>
  <parameter name="driverLink" value="E:\\workspaces\\chromedriver.exe"/>
  <parameter name="searchKey" value="Printed Summer Dress"/>
  <parameter name="emailAddrLogin" value="hasnain2808@gmail.com"/>
  <test thread-count="2" name="Test" parallel="classes">
    <classes>
      <class name="com.moha.TestLogin"/>
      <class name="com.moha.TestSearch"/>
      <class name="com.moha.TestCart"/>
      <class name="com.moha.TestRegister"/>
    </classes>
  </test> <!-- Test -->
</suite> <!-- Suite -->
```



Test results

1 suite

All suites

Suite

Info

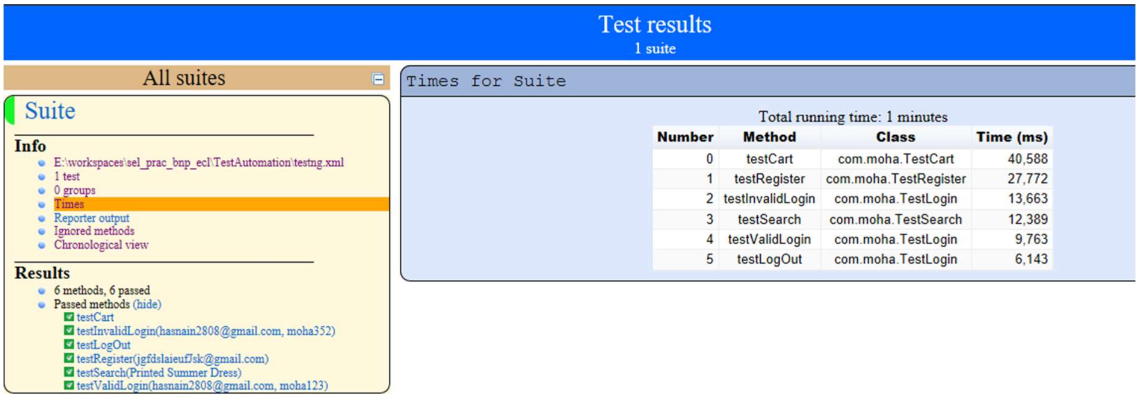
- E:\workspaces\sel_prac_bnp_ecl\TestAutomation\testng.xml
- 1 test
- 0 groups
- Times
- Reporter output
- Ignored methods
- Chronological view

Results

- 6 methods, 6 passed
- Passed methods (hide)
  - ☑ testCart
  - ☑ testInvalidLogin(hasnain2808@gmail.com, moha352)
  - ☑ testLogOut
  - ☑ testRegister(jgfdslaieufJsk@gmail.com)
  - ☑ testSearch(Printed Summer Dress)
  - ☑ testValidLogin(hasnain2808@gmail.com, moha123)

Methods in chronological order

```
com.moha.TestSearch
      createWebDriver(E:\\workspaces\\chromedriver.exe)                       0 ms
com.moha.TestCart
      createWebDriver(E:\\workspaces\\chromedriver.exe)                       0 ms
com.moha.TestSearch
           testSearch(Printed Summer Dress)                                5615 ms
com.moha.TestCart
           testCart                                                        5615 ms
com.moha.TestSearch
      closeWebDrier                                                       18007 ms
com.moha.TestRegister
      createWebDriver(E:\\workspaces\\chromedriver.exe)                   20061 ms
           testRegister(jgfdslaieufJsk@gmail.com)                         24207 ms
com.moha.TestCart
      closeWebDrier                                                       46205 ms
com.moha.TestRegister
      closeWebDrier                                                       51983 ms
com.moha.TestLogin
      createWebDriver(E:\\workspaces\\chromedriver.exe)                   52750 ms
           testInvalidLogin(hasnain2808@gmail.com, moha352)               56252 ms
           testValidLogin(hasnain2808@gmail.com, moha123)                 69916 ms
           testLogOut                                                     79686 ms
      closeWebDrier                                                       85829 ms
```



Test results

1 suite

All suites

Suite

Info

- E:\workspaces\sel_prac_bnp_ecl\TestAutomation\testng.xml
- 1 test
- 0 groups
- Times
- Reporter output
- Ignored methods
- Chronological view

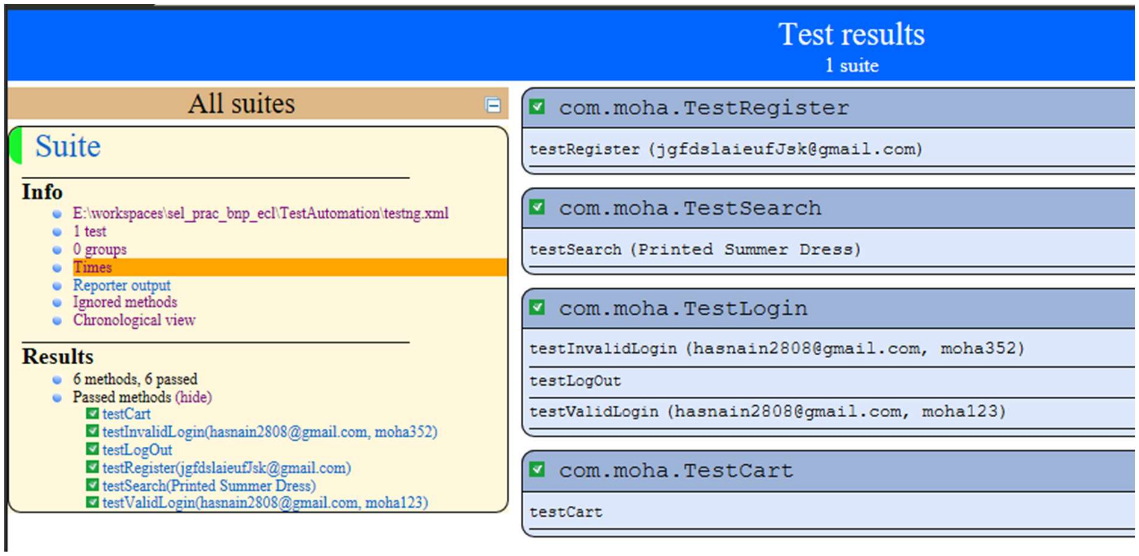
Results

- 6 methods, 6 passed
- Passed methods (hide)
  - ☑ testCart
  - ☑ testInvalidLogin(hasnain2808@gmail.com, moha352)
  - ☑ testLogOut
  - ☑ testRegister(jgfdslaieufJsk@gmail.com)
  - ☑ testSearch(Printed Summer Dress)
  - ☑ testValidLogin(hasnain2808@gmail.com, moha123)

Times for Suite

Total running time: 1 minutes

Number	Method	Class	Time (ms)
0	testCart	com.moha.TestCart	40,588
1	testRegister	com.moha.TestRegister	27,772
2	testInvalidLogin	com.moha.TestLogin	13,663
3	testSearch	com.moha.TestSearch	12,389
4	testValidLogin	com.moha.TestLogin	9,763
5	testLogOut	com.moha.TestLogin	6,143



Test results

1 suite

All suites

Suite

Info

- E:\workspaces\sel_prac_bnp_ecl\TestAutomation\testng.xml
- 1 test
- 0 groups
- Times
- Reporter output
- Ignored methods
- Chronological view

Results

- 6 methods, 6 passed
- Passed methods (hide)
  - ☑ testCart
  - ☑ testInvalidLogin(hasnain2808@gmail.com, moha352)
  - ☑ testLogOut
  - ☑ testRegister(jgfdslaieufJsk@gmail.com)
  - ☑ testSearch(Printed Summer Dress)
  - ☑ testValidLogin(hasnain2808@gmail.com, moha123)

☑ com.moha.TestRegister

testRegister (jgfdslaieufJsk@gmail.com)

☑ com.moha.TestSearch

testSearch (Printed Summer Dress)

☑ com.moha.TestLogin

testInvalidLogin (hasnain2808@gmail.com, moha352)

testLogOut

testValidLogin (hasnain2808@gmail.com, moha123)

☑ com.moha.TestCart

testCart