

Day 5 - Test case solution document

Code Used:

BaseTest.java

```
package com.ibm.test;

import java.io.File;
import java.io.IOException;
import java.util.Date;
import org.apache.commons.io.FileUtils;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.testng.Assert;
import org.testng.Reporter;
import org.testng.annotations.Optional;
import org.testng.annotations.Parameters;
import org.testng.annotations.Test;
import com.ibm.pages.CategoryPage;
import com.ibm.pages.CouponPage;
import com.ibm.pages.DashboardPage;
import com.ibm.pages.DriverLaunch;
import com.ibm.pages.LoginPage;
import com.ibm.pages>ReturnsPage;
import com.ibm.pages.ShippingPage;
import com.ibm.pages.TabsPage;

public class BaseTest extends DriverLaunch
{
    @Test(testName="ShipLocationSearch")
    public void Day5TestCase() throws IOException
    {
        LoginPage login=new LoginPage(driver,wait);
        DashboardPage dash=new DashboardPage(driver,wait);
        ShippingPage ship=new ShippingPage(driver,wait);
        try
        {
            String url=hashData.get("url");
            String username=hashData.get("username");
            String password=hashData.get("password");
            String searchlocation=hashData.get("searchlocation");
            String noshiploc=hashData.get("noshiploc");
            //Navigating to Admin page of atozgroceries.com
            driver.get(url);
            //logging into the admin page
            login.EnterEmail(username);
            login.EnterPassword(password);
            login.clickLogin();
            //opening System from dashboard
            dash.OpenSystem();
            //opening Shipping under System
            dash.OpenShipping();
            //opening Shipping location under Shipping
```

```

        ship.OpenShippingLocation();
        //sending the search location for searching availability in the table
        String searchShipLoc=ship.searchData(searchlocation);
        //validating the availability of Search location
        if(searchShipLoc.contains(searchlocation))
        {
            //if search location is found
            Assert.assertTrue(searchShipLoc.contains(searchlocation), "Assertion on
Search data found!!");
            //reports the log in test output
            Reporter.log("Assertion on Search data found!!");
        }
        else
        {
            //if no search location is found
            Assert.assertTrue(searchShipLoc.contains(noshiploc), "Assertion on No
Matching Records!!");
            //reports the log in test output
            Reporter.log("Assertion on No Matching Records!!");
        }
    }
    catch(Exception e)// catching if any exception occurs and takes a screenshot
of the page
    {
        TakesScreenshot ts = (TakesScreenshot) driver;
        File file = ts.getScreenshotAs(OutputType.FILE);
        Date date = new Date();
        String currentDate = date.toString().replace(":", "-");
        FileUtils.copyFile(file, new File("./Screenshots/Day5_TestCase_Error_" +
currentDate + ".jpg"));
        System.out.println(e.getMessage());
        Assert.fail();
    }
    finally
    {
        login.ClickLogout();
    }
}
}

```

LoginPage.java

```

package com.ibm.pages;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.openqa.selenium.support.ui.WebDriverWait;

public class LoginPage {
    @FindBy(name="email")

```

```

WebElement emailEle;

@FindBy(name="pword")
WebElement passwordEle;

@FindBy(xpath="/html/body/div/div/div/div[2]/form/button")
WebElement loginEle;

@FindBy(xpath="//a[@title='Logout']")
WebElement logoutEle;

WebDriverWait wait;
WebDriver driver;

public LoginPage(WebDriver driver, WebDriverWait wait)
{
    PageFactory.initElements(driver, this);
}
public void EnterEmail(String username)
{
    emailEle.sendKeys(username);
}
public void EnterPassword(String password)
{
    passwordEle.sendKeys(password);
}
public void clickLogin()
{
    loginEle.click();
}
public void ClickLogout()
{
    logoutEle.click();
}
}

```

DashboardPage.java

```

package com.ibm.pages;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;

public class DashboardPage {

//Finding the WebElements
@FindBy(partialLinkText="Catalog")
WebElement catalogEle;

@FindBy(xpath="/html/body/div[1]/div[1]/div/ul/li[5]/a")
WebElement marketEle;
}

```

```

@FindBy(xpath="//*[contains(text(),'System')]")
WebElement systemEle;

@FindBy(xpath="//*[contains(text(),'Shipping')]")
WebElement shipEle;

WebDriver driver;
WebDriverWait wait;

//catalog page constructor
public DashboardPage(WebDriver driver, WebDriverWait wait)
{
    this.driver=driver;
    this.wait=wait;
    PageFactory.initElements(driver, this);
}

//Opencatalog method to click on catalog
public void OpenCatalog()
{
    wait.until(ExpectedConditions.visibilityOf(catalogEle));
    catalogEle.click();
}

//method for opening marketing
public void OpenMarketing()
{
    marketEle.click();
}
//method for opening system
public void OpenSystem()
{
    systemEle.click();
}
public void OpenShipping()
{
    shipEle.click();
}
}

```

ShippingPage.java

```

package com.ibm.pages;

import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.openqa.selenium.support.ui.WebDriverWait;

public class ShippingPage {

    //Finding Elements of Shipping Location

```

```

@FindBy(xpath="//*[contains(text(),'Shipping Location')]")
WebElement shipLocEle;

@FindBy(xpath="//*[@type='search']")
WebElement searchEle;

@FindBy(xpath="//*[@class='dataTables_empty']")
WebElement dataEmptyEle;

WebDriver driver;
WebDriverWait wait;
JavascriptExecutor js;
//Constructor of Shipping Location
public ShippingPage(WebDriver driver,WebDriverWait wait)
{
    this.driver=driver;
    this.wait=wait;
    js=(JavascriptExecutor) driver;
    PageFactory.initElements(driver, this);
}
//method to open Shipping Locations under Shipping
public void OpenShippingLocation()
{
    js.executeScript("arguments[0].scrollIntoView(true);", shipLocEle);
    shipLocEle.click();
}
//Method to search the given location
public String searchData(String searchLoc)
{
    //sending the search key in the Search input field
    searchEle.sendKeys(searchLoc);
    String searchPgSrc = driver.getPageSource();
    //if search key has data it returns pagesource
    if(searchPgSrc.contains(searchLoc))
    {
        searchPgSrc=driver.getPageSource();
    }
    else//if search key has no data it
    {
        searchPgSrc=dataEmptyEle.getText().trim();
    }
    return searchPgSrc;
}
}

DriverLaunch.java
package com.ibm.pages;

import java.io.IOException;
import java.util.HashMap;
import java.util.concurrent.TimeUnit;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

```

```

import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.ie.InternetExplorerDriver;
import org.openqa.selenium.support.ui.WebDriverWait;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.BeforeSuite;
import org.testng.annotations.Optional;
import org.testng.annotations.Parameters;

import com.ibm.utilities.PropertiesFileHandler;

public class DriverLaunch {

    public WebDriver driver;
    public WebDriverWait wait;
    public PropertiesFileHandler propFileHandle;
    public HashMap<String, String> hashData;

    @BeforeSuite
    public void preSetForTest() throws IOException
    {
        String file=".TestData/data.properties";
        propFileHandle =new PropertiesFileHandler();
        hashData=propFileHandle.getPropertiesAsMap(file);
    }
    @BeforeMethod
    @Parameters({"browser"})
    public void initialisation(@Optional("ch")String browser)
    {
        browserInitialization(browser);
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
        wait=new WebDriverWait(driver, 60);
    }
    @AfterMethod
    public void closingBrowser()
    {
        driver.quit();
    }

    public void browserInitialization(String browser)
    {
        switch(browser.toLowerCase())
        {
            case "ff":
                System.setProperty("webdriver.gecko.driver",
"./drivers/geckodriver.exe");
                driver=new FirefoxDriver();
                break;
            case "ch":
                System.setProperty("webdriver.chrome.driver",
"./drivers/chromedriver.exe");
                driver=new ChromeDriver();
                break;
        }
    }
}

```

```

        case "ie":
            System.setProperty("webdriver.ie.driver",
"./drivers/IEDriverServer.exe");
                driver=new InternetExplorerDriver();
                break;
            }
        }
    }
}

```

PropertiesFileHandler.java

```

package com.ibm.utilities;

import java.io.FileInputStream;
import java.io.IOException;
import java.util.HashMap;
import java.util.Properties;
import java.util.Set;

public class PropertiesFileHandler {

    public HashMap<String, String> getPropertiesAsMap(String file) throws
IOException {
        HashMap<String, String> magentoMap = new HashMap<String, String>();

        FileInputStream fileIn = new FileInputStream(file);
        Properties prop = new Properties();
        prop.load(fileIn);

        Set<Object> keysProp = prop.keySet();
        for (Object key : keysProp) {
            magentoMap.put(key.toString(), prop.getProperty(key.toString()));
        }
        prop.clear();
        return magentoMap;
    }

}

```

data.properties

For BRIGADE search location

```

url=https://atozgroceries.com/admin
username=demo@atozgroceries.com
password=456789
categoryname=Milk Products
tagtitle=Milk Products
sortorder=12
status=Enabled
show=All
tagname=Milk
tagsort=10
tagstatus=Enabled

```

```

tagrequiredmsg=fill out this field
tagSelectReqMsg=select an item in the list
couponcode=15OFF
expdeletemsg=successfully deleted
returnreason=small in size
searchlocation=BRIGADE
noshiploc>No matching records found

```

For No Matching records

```

url=https://atozgroceries.com/admin
username=demo@atozgroceries.com
password=456789
categoryname=Milk Products
tagtitle=Milk Products
sortorder=12
status=Enabled
show>All
tagname=Milk
tagsort=10
tagstatus=Enabled
tagrequiredmsg=fill out this field
tagSelectReqMsg=select an item in the list
couponcode=15OFF
expdeletemsg=successfully deleted
returnreason=small in size
searchlocation=BRIGADE
noshiploc>No matching records found

```

testng.xml

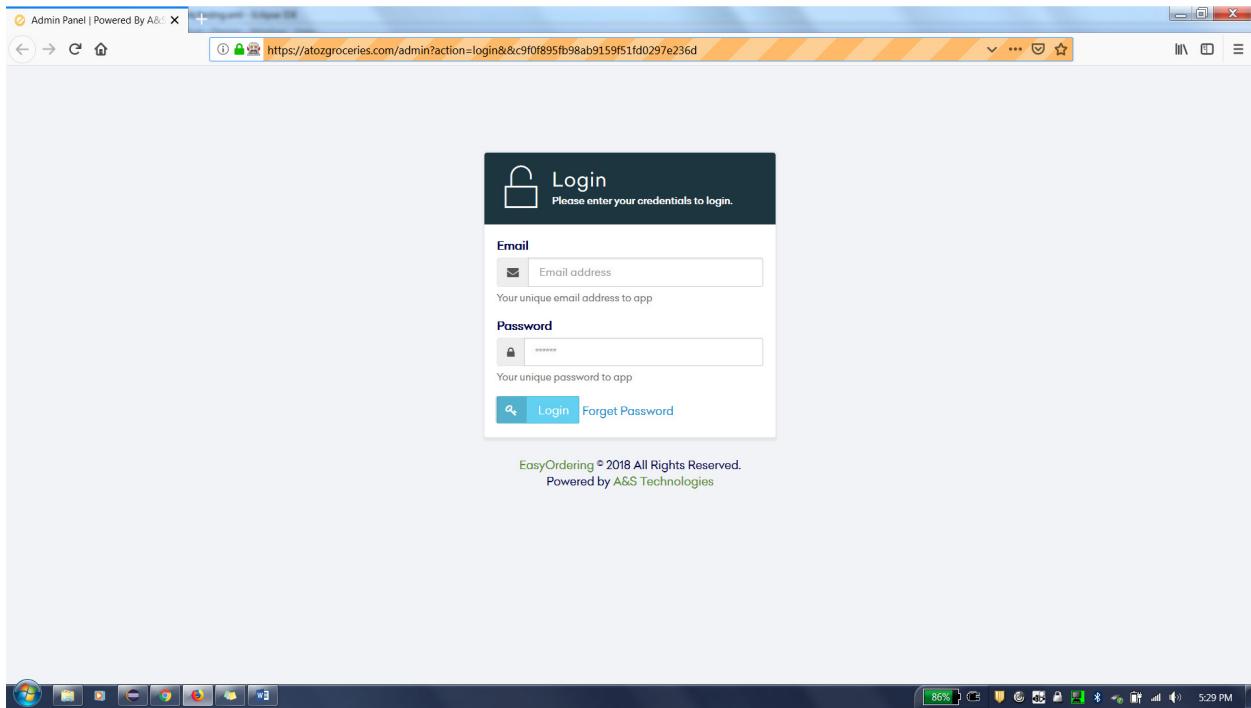
```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Suite">
  <test thread-count="5" name="Test">
    <parameter name="browser" value="ff"></parameter>
    <classes>
      <class name="com.ibm.test.BaseTest">
        <methods><include name="Day5TestCase"></include></methods>
      </class>
    </classes>
  </test> <!-- Test -->
</suite> <!-- Suite -->

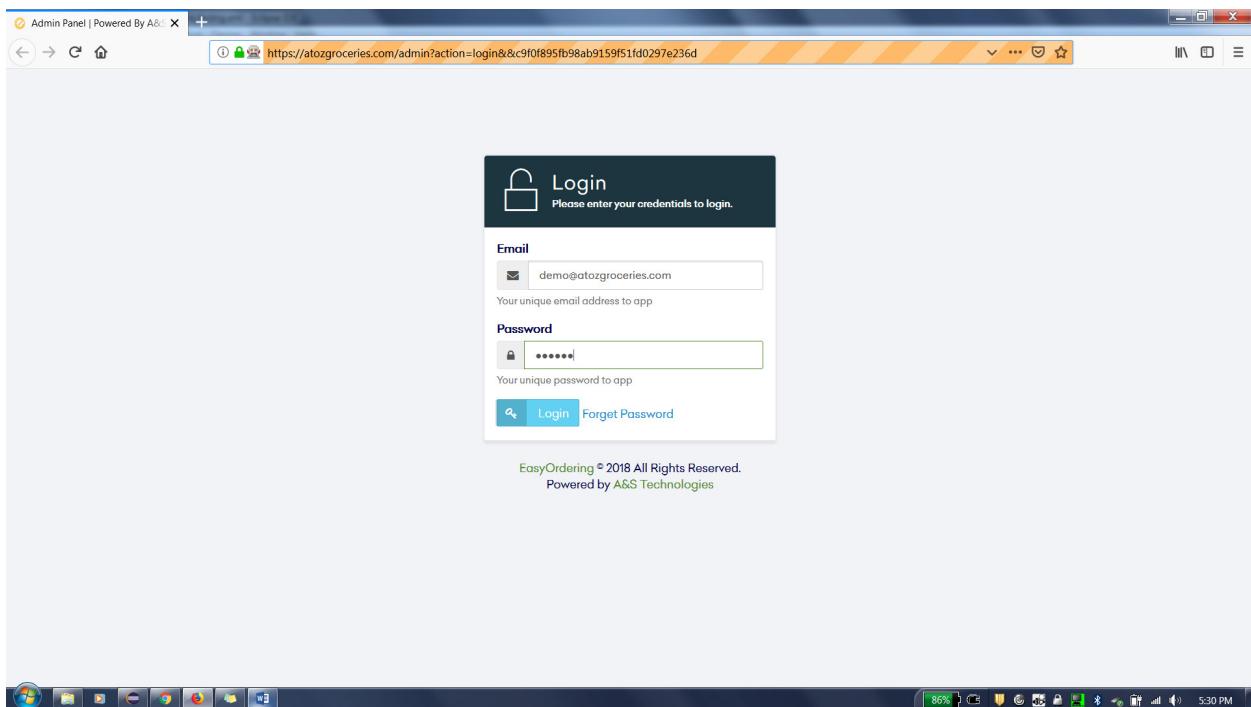
```

Step by step output:

1. Navigating to the Admin Page of “atozgroceries”



2. Providing login details



3. After logging into the webpage

The screenshot shows the Admin Panel of the easyordering system. The dashboard features four main statistics boxes: 'Total Orders' (0), 'Total Sales' (₹589.38), 'Total Customers' (0), and 'Total Products' (41). Below these is a section titled 'Latest Orders' with a table listing four pending orders:

#	Order ID	Customer	Status	Total	Date Added	Action
1	#4		Pending	₹3,570.00	2018-12-18	
2	#3		Pending	₹3,570.00	2018-12-18	
3	#2		Pending	₹3,570.00	2018-12-18	
4	#1		Pending	₹1,425.00	2018-12-10	

4. Opening System

The screenshot shows the Admin Panel of the easyordering system. The dashboard features four main statistics boxes: 'Total Orders' (4), 'Total Sales' (₹3,135.00), 'Total Customers' (4), and 'Total Products' (222). The 'System' menu item in the sidebar is highlighted with a red box. Below these is a section titled 'Latest Orders' with a table listing four pending orders:

#	Order ID	Customer	Status	Total	Date Added	Action
1	#4		Pending	₹3,570.00	2018-12-18	
2	#3		Pending	₹3,570.00	2018-12-18	
3	#2		Pending	₹3,570.00	2018-12-18	
4	#1		Pending	₹1,425.00	2018-12-10	

5. Opening Shipping

The screenshot shows the Admin Panel Dashboard of easyordering. On the left, a sidebar menu includes options like Dashboard, Catalog, Sales, Customers, Marketing, System (with sub-options for Settings, Users, Shipping, Shipping Methods, Shipping Locations, Shipping Pincode, Order Statuses, Weight Class, Returns, and Manage Backup), Reports, and a bottom section for Reports. The main area displays four cards: 'Total Orders' (4), 'Total Sales' (₹3,135.00), 'Total Customers' (4), and 'Total Products' (222). Below these are sections for 'Latest Orders' and 'Recent Activity'. The 'Latest Orders' table lists four pending orders with IDs #4, #3, #2, and #1, each with a view more link.

6. Opening Shipping Locations

The screenshot shows the 'Shipping Location' list page in the Admin Panel. The sidebar menu is identical to the dashboard. The main content area is titled 'Shipping Location List' and contains a table of shipping locations. The table has columns for #, City Name, Sort Order, Status, and Action. The data in the table is as follows:

#	City Name	Sort Order	Status	Action
1	RAJARAJESHWARI NAGAR	1	Enabled	Action
2	NEW THIPPASANDRA	1	Disabled	Action
3	TestLocation	1	Enabled	Action
4	NEELASANDAR	1	Disabled	Action
5	NAYANDAHALLI	1	Enabled	Action
6	NANDINI LAYOUT	1	Disabled	Action
7	NAGAETTY HALLI	1	Disabled	Action
8	NAGASANDRA	1	Disabled	Action

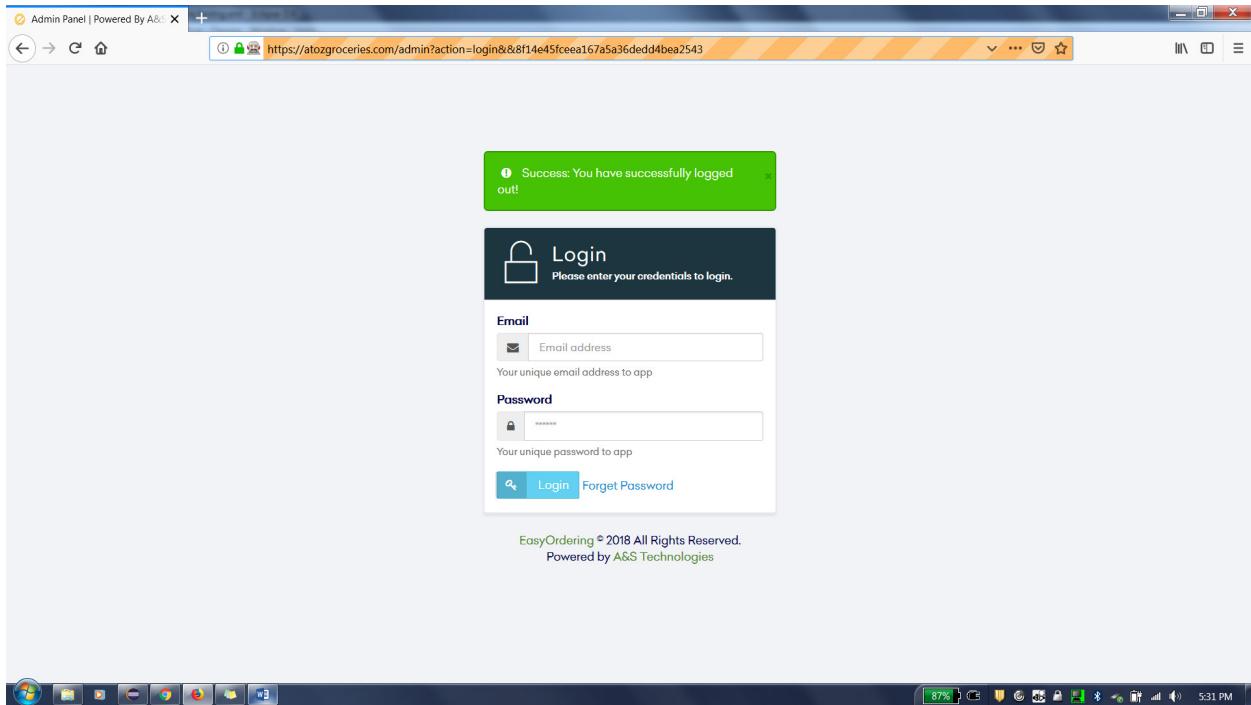
7. Sending search location – BRIGADE

The screenshot shows the Admin Panel of the easyordering system. The left sidebar has a 'System' section expanded, containing 'Settings', 'Users', 'Shipping' (which is selected), and other options like 'Reports'. The main content area is titled 'Shipping Location' and shows a table titled 'Shipping Location List'. A search bar at the top right contains the text 'BRIGADE'. The table has columns: '#', 'City Name', 'Sort Order', 'Status', and 'Action'. One row is visible: '#105' with 'BRIGADE ROAD' in the 'City Name' column, '1' in 'Sort Order', 'Disabled' in 'Status', and a blue 'Action' button. The status bar at the bottom shows battery level at 87% and the time as 5:34 PM.

8. Sending search location – BRIGDE for Validating No records found

This screenshot is similar to the previous one but shows a different search result. The search bar now contains 'BRIGDE'. The table below it displays the message 'No matching records found' instead of a list of results. The rest of the interface and status bar are identical to the first screenshot.

9. Logging out successfully after testing



10. Output from console

A screenshot of the Eclipse IDE interface. The left side shows the "Package Explorer" with various Java files like "BaseTest.java", "Type.properties", "testng.xml", "ShippingPage.java", "ReturnsPage.java", "DashboardPage.java", and "DriverLaunch.java". The right side shows the "Console" view which displays the output of a TestNG run. The output includes XML configuration code for testng.xml, followed by a series of log messages from the "Marionette" process, including warnings about certificates and socket errors. It also shows Java errors related to file paths and protocol handshake issues. The Eclipse toolbar and status bar are visible at the bottom.

Day 5 Test Case - Emailable report – if data available

The screenshot shows a web browser window with the title "TestNG Report". The URL is "file:///C:/Harisha/Selenium_Training/EclipseWorkSpace/SeleniumFramework/test-output/emailable-report.html". The page content includes a summary table for "Test" and "Suite", and a detailed table for "Suite". A message box displays "Assertion on Search data found!!".

Test	# Passed	# Skipped	# Failed	Time (ms)	Included Groups	Excluded Groups
Test	1	0	0	21,365		

Class	Method	Start	Time (ms)
Suite			
Test — passed			
com.ibm.test.BaseTest	Day5TestCase	1545221053663	14864

Test
com.ibm.test.BaseTest#Day5TestCase
Messages
Assertion on Search data found!!

[back to summary](#)



Day 5 Test Case – index report – if data available

The screenshot shows a web browser window with the title "TestNG Report". The URL is "file:///C:/Harisha/Selenium_Training/EclipseWorkSpace/SeleniumFramework/test-output/index.html#". The page content includes a sidebar for "All suites" and a main panel for "Reporter output for Suite". A message box displays "Assertion on Search data found!!".

All suites

Info

- C:\Harisha\Selenium_Training\EclipseWorkSpace\SeleniumFramework\testing.xml
- 1 test
- 0 groups
- Times
- Reporter output
- Ignored methods
- Chronological view

Results

- 1 method, 1 passed
- Passed methods (hide)
 - Day5TestCase

Reporter output for Suite

Day5TestCase

Assertion on Search data found!!



Day 5 Test Case - Emailable report – if data not available

The screenshot shows a web browser window with three tabs: 'TestNG Report', 'TestNG reports', and 'TestNG Report'. The URL is file:///C:/Harisha/Selenium_Training/EclipseWorkSpace/SeleniumFramework/test-output/emailable-report.html. The main content area displays a TestNG report for 'Day5TestCase'. A summary table at the top shows 1 test passed, 0 skipped, 0 failed, and a total time of 23.521 ms. Below this is a 'Suite' table with one row for 'Day5TestCase' which is marked as 'passed'. Under the 'Test' section, there is a 'Messages' box containing the error message 'Assertion on No Matching Records!!'. A link 'back to summary' is visible at the bottom.



Day 5 Test Case – index report – if data not available

The screenshot shows a web browser window with three tabs: 'TestNG Report', 'TestNG reports', and 'TestNG Report'. The URL is file:///C:/Harisha/Selenium_Training/EclipseWorkSpace/SeleniumFramework/test-output/index.html#. The main content area displays a TestNG report for 'Day5TestCase'. A summary table at the top shows 1 suite. The 'Test results' section shows 1 suite with 1 method passed. The 'Info' section provides details about the test configuration. A 'Reporter output for Suite' box contains the error message 'Assertion on No Matching Records!!'.



*** End of the document ***