Objective:

The objective is to write code to dynamically check all the links under the table.

Task:

Launch the url https://www.moneycontrol.com

Click on Stocks link.

It will list Stock Price Quotes List with company name.

Get all the links and link text of the company under the list of Alphabet 'A' and print it on the console.

Assert the title using TestNG Assertion after clicking on 'Ashok Leyland' in the table of company name

Expected title of the 'Ashok Leyland page':

"Ashok Leyland Ltd. Stock Price, Share Price, Live BSE/NSE, Ashok Leyland Ltd. Bids Offers. Buy/Sell Ashok Leyland Ltd. news & tips, & F&O Quotes, NSE/BSE Forecast News and Live Quotes"

Source code:

```
File1: DisplayLinks.java
package com.ibm.dynamictable;
import java.util.concurrent.TimeUnit;
import org.openga.selenium.By;
import org.openga.selenium.WebDriver;
import org.openga.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openga.selenium.support.ui.WebDriverWait;
import org.testng.Assert;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;
import com.ibm.initialization.LaunchWebDriver;
import com.ibm.tablepages.CompaniesPage;
import com.ibm.tablepages.ContinuePage;
import com.ibm.tablepages.NewsPage;
import com.ibm.tablepages.StockQuotesPage;
public class DisplayLinks extends LaunchWebDriver {
      // To display all the links and link text
      @Test
      public void printLinks() {
             String url = data.get("urlmoneycontrol");
             String expectedTitle = data.get("exptitle");
             // To launch money control web application
             driver.get(url);
             NewsPage newspage = new NewsPage(driver, wait);
```

```
// To click on stocks link
             newspage.clickOnStocks();
             StockQuotesPage quotespage = new StockQuotesPage(driver, wait);
             // To click on A link
             quotespage.clickOnA();
             CompaniesPage companypage = new CompaniesPage(driver, wait);
             // To print links and link text of companies
             System.out.println(data.get("linksconsole"));
             companypage.printLinks();
             String actualTitle = companypage.clickonCompany();
             System.out.println("Title is " + actualTitle);
             // Assertion of title of the page
             Assert.assertTrue(actualTitle.contains(expectedTitle), "Assertion on
page title");
      }
}
File2: NewsPage.java
package com.ibm.tablepages;
import org.openga.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.openga.selenium.support.ui.WebDriverWait;
public class NewsPage {
      // Locating continue link
      @FindBy(xpath = "//a[@title='Stocks']")
      WebElement stocksEle;
      By stocks = By.xpath("//a[@title='Stocks']");
      WebDriverWait wait;
      WebDriver driver;
      public NewsPage(WebDriver driver, WebDriverWait wait) {
             PageFactory.initElements(driver, this);
             this.driver = driver;
             this.wait = wait;
      }
      // To click on Stocks link
```

```
public void clickOnStocks()
      {
             stocksEle.click();
      }
}
File3: StockQuotesPage.java
package com.ibm.tablepages;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.openga.selenium.support.ui.WebDriverWait;
public class StockQuotesPage {
      // Locating A link
      @FindBy(xpath = "//a[@title='A']")
      WebElement linkA;
      WebDriverWait wait;
      WebDriver driver;
      public StockQuotesPage(WebDriver driver, WebDriverWait wait) {
             PageFactory.initElements(driver, this);
             this.driver = driver;
             this.wait = wait;
      }
      // To click on A link
      public void clickOnA() {
             linkA.click();
      }
}
File4: CompaniesPage.java
package com.ibm.tablepages;
import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openga.selenium.support.FindAll;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
```

```
import org.openqa.selenium.support.ui.WebDriverWait;
public class CompaniesPage {
      WebDriverWait wait;
      WebDriver driver;
      @FindAll(@FindBy(tagName = "a"))
      List<WebElement> linksAll;
      @FindBy(xpath = "//a[@title='Ashok Leyland']")
      WebElement companylink;
      public CompaniesPage(WebDriver driver, WebDriverWait wait) {
             PageFactory.initElements(driver, this);
             this.driver = driver;
             this.wait = wait;
      }
      // To print links and link text
      public void printLinks() {
             // Finding the total number of rows in the table.
             List<WebElement> rows = (List<WebElement>) driver
                           .findElements(By.xpath("//table[@class='pcq tbl
MT10']/tbody/tr"));
             int rowTotal = rows.size();
             List<WebElement> columns = (List<WebElement>) driver
                          .findElements(By.xpath("//table[@class='pcq_tbl
MT10']/tbody/tr[2]/td"));
             int colTotal = columns.size();
             System.out.println();
             for (int i = 2; i <= rowTotal; i++) {</pre>
                    for (int j = 1; j <= colTotal; j++) {</pre>
                          WebElement companyElt = driver.findElement(
                                        By.xpath("//table[@class='pcq_tbl
MT10']/tbody/tr[" + i + "]/td[" + j + "]/descendant::a"));
                          System.out.println(companyElt.getText().toString());
                          System.out.println(companyElt.getAttribute("href"));
                    }
             }
      }
      // To click on Ashok Leyland company
      public String clickonCompany()
      {
```

```
companylink.click();
             return driver.getTitle();
      }
}
File5: Launch Web Driver. java
package com.ibm.initialization;
import java.io.IOException;
import java.util.HashMap;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.WebDriver;
import org.openga.selenium.firefox.FirefoxDriver;
import org.openga.selenium.support.ui.WebDriverWait;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.BeforeSuite;
import com.ibm.utilities.PropertiesFileHandler;
public class LaunchWebDriver {
      public WebDriver driver;
      public WebDriverWait wait;
      public PropertiesFileHandler propFileHandler;
      public HashMap<String, String> data;
      @BeforeSuite
      public void preSetForTest() throws IOException {
             String file = "./TestData/dynamictable.properties";
             propFileHandler = new PropertiesFileHandler();
             data = propFileHandler.getPropertiesAsMap(file);
      }
      //Setting path for webdriver and launching web application
      @BeforeMethod
      public void initializeDriver()
      {
                          //Setting path for Firefox Driver
                          System.setProperty("webdriver.gecko.driver",
"./drivers/geckodriver.exe");
                          driver = new FirefoxDriver();
                          driver.manage().timeouts().implicitlyWait(30,
TimeUnit.SECONDS);
                          wait = new WebDriverWait(driver, 60);
                          driver.manage().window().maximize();
      }
      //To close the webdriver
      @AfterMethod
```

```
public void close()
      {
             driver.close();
      }
}
File6: PropertiesFileHandler.java
package com.ibm.utilities;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
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> dynamicMap = 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) {
                    dynamicMap.put(key.toString(), prop.getProperty(key.toString()));
             prop.clear();
             return dynamicMap;
      }
      public void setKeyAndValue(String file,String key,String value) throws
IOException
             FileInputStream fileIn = new FileInputStream(file);
             Properties prop = new Properties();
             prop.load(fileIn);
             prop.setProperty(key, value);
             FileOutputStream fOut=new FileOutputStream(file);
             prop.store(fOut, "Test Result");
             fOut.close();
             fileIn.close();
      }
```

```
}
```

File7:dynamictable.properties

```
url=https://datatables.net
message=The list of employees with age >=50 and salary >=200000
urlmoneycontrol=https://www.moneycontrol.com
linksconsole=The links and link text of the company under the list of Alphabet 'A':
exptitle=Ashok Leyland Ltd. Stock Price, Share Price, Live BSE/NSE, Ashok Leyland
Ltd. Bids Offers. Buy/Sell Ashok Leyland Ltd. news & tips, & F&O Quotes, NSE/BSE
Forecast News and Live Quotes
```

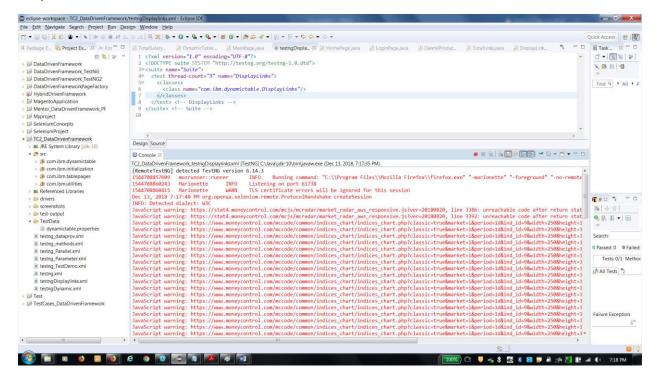
File8:testngDisplaylinks.xml

Result:

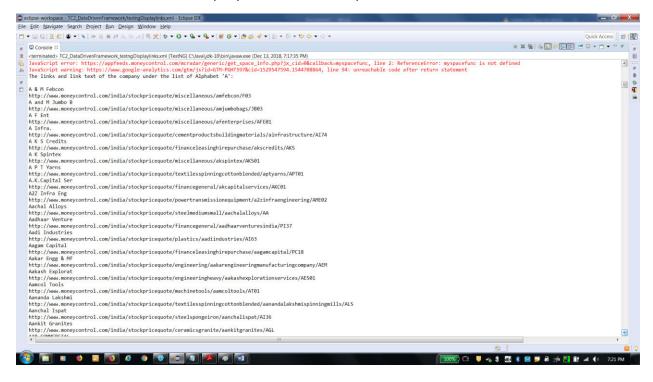
All the links and link text of the company under the list of Alphabet 'A' are printed on console and assertion is performed on expected page title.

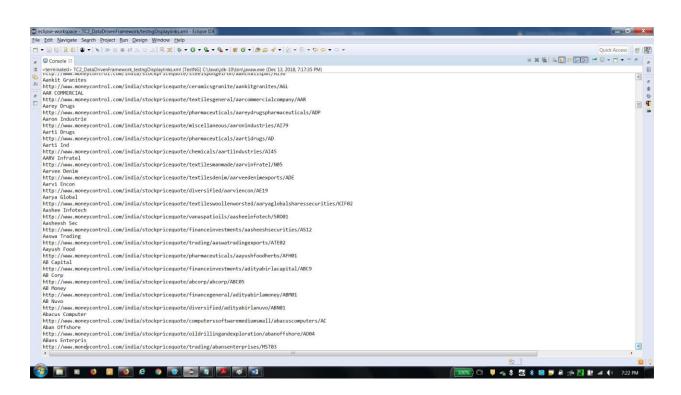
Screenshots:

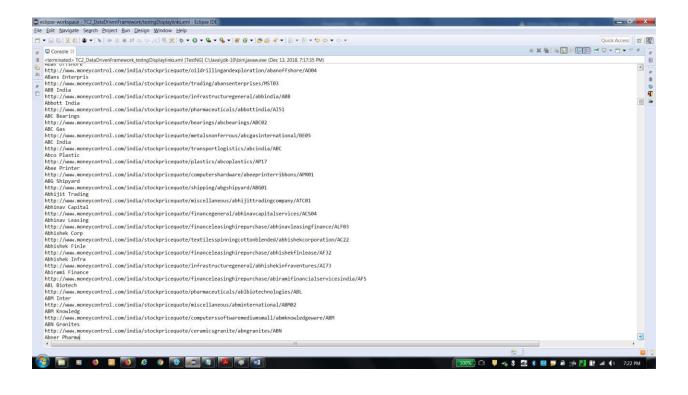
Output on console:

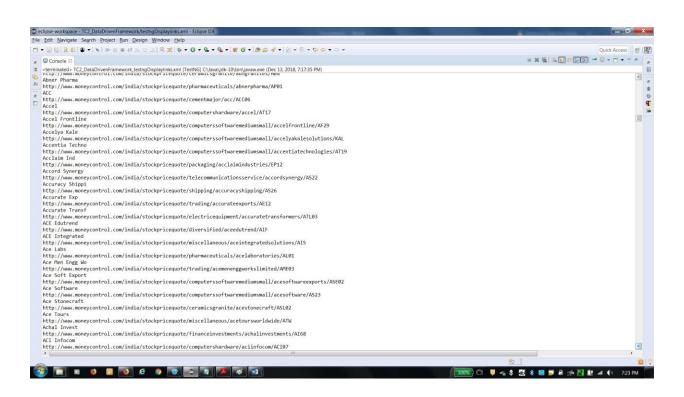


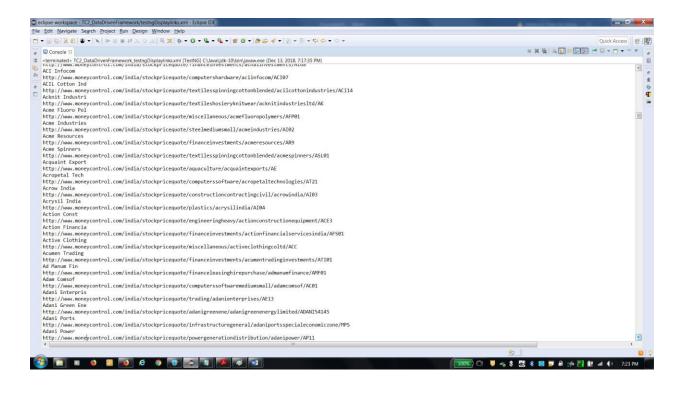
The links and link text of the company under the list of Alphabet 'A'

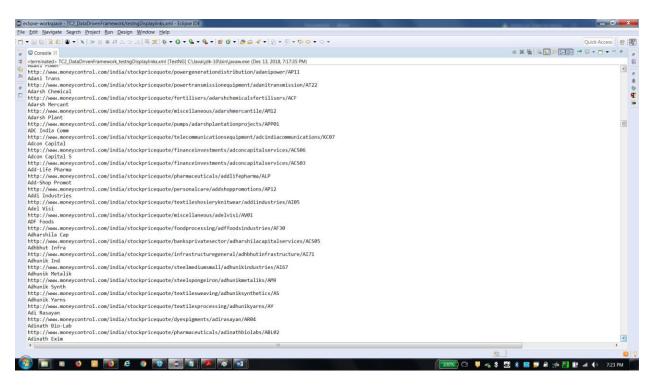


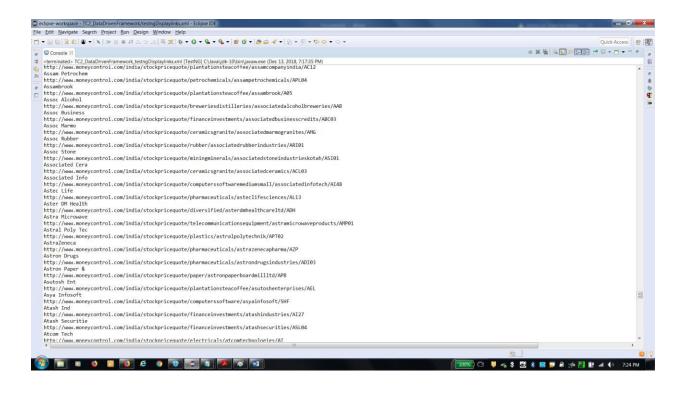


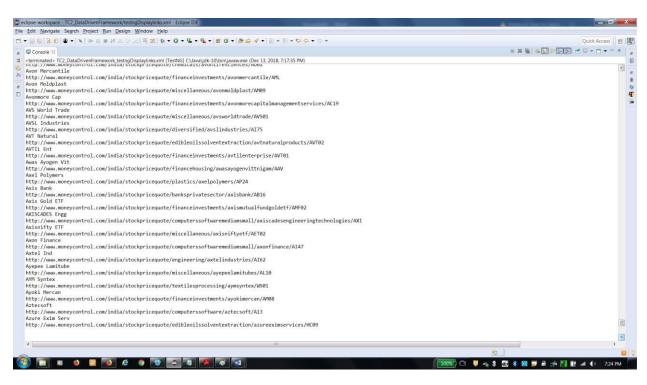


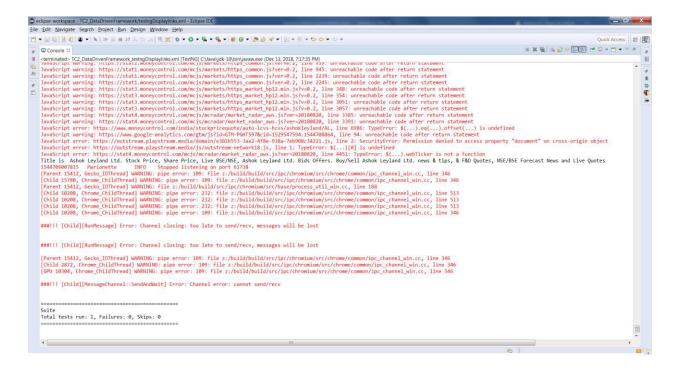


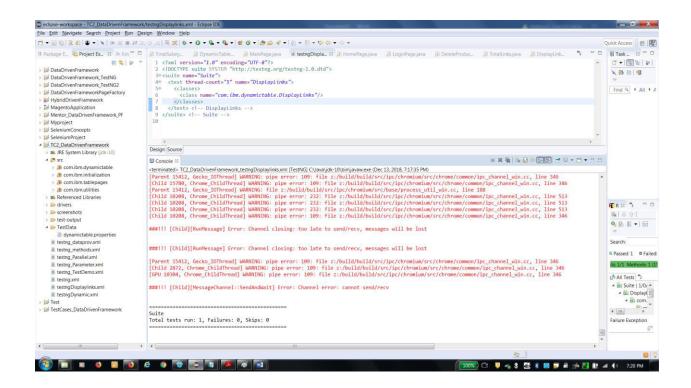










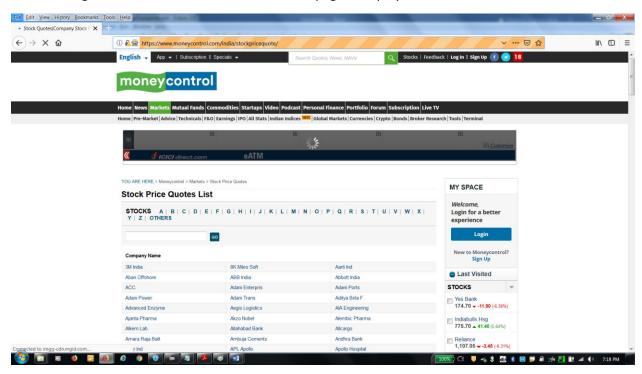


Display of the first page when the site for money control is launched.

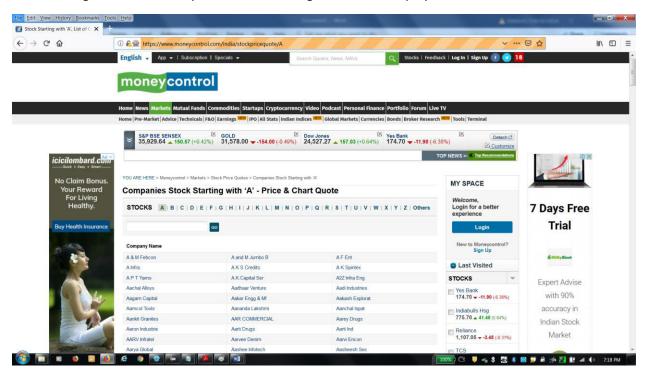


100% C3 U 🐇 \$ 🐼 🕸 🗷 🖟 🖺 🏥 🖽 🜓 al (i) 7:17 PM

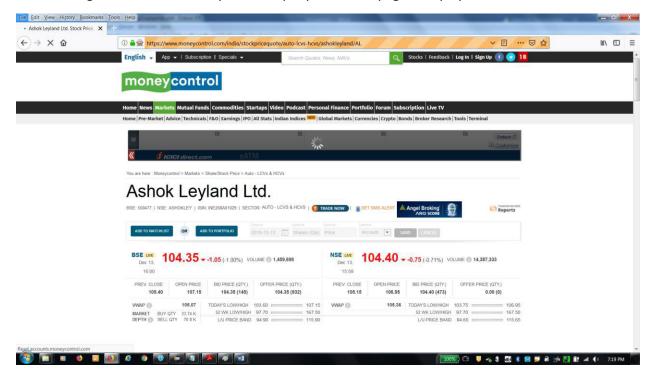
After clicking on stocks link 'Stock Price Quote List' page is displayed

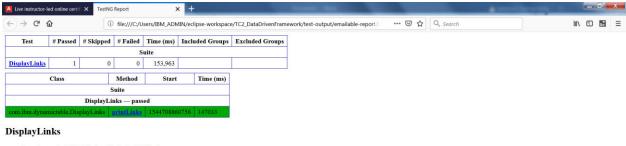


After clicking on link 'A', companies stock starting with 'A' are dispalyed



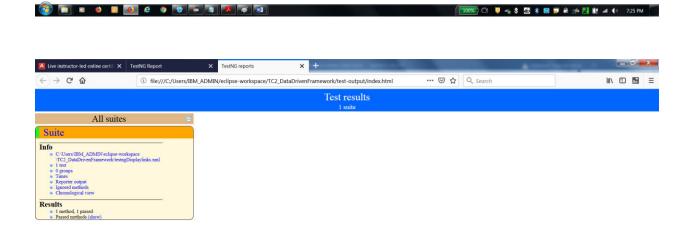
After clicking on link for 'Ashok Leyland' company the below page is displayed





com.ibm.dynamictable.DisplayLinks#printLinks







Flow Information:

Initially a property file 'dynamictable.properties' is created in Test Data folder of the project and list of key, values for url of money control site, expected page title is stored in this property file.

Under the class 'LaunchWebDriver', a Test NG annotation 'Before Suite' is used to invoke the method 'preSetFortTest' which is used to instantiate object for class PropertiesFileHandler to return the list of these key, values in to a HashMap.

The testng annotation 'BeforeMethod' is used with method initializeDriver' to set the relative path and instantiating webdriver for firefox under LaunchWebDriver class.

The class NewsPage uses the annotation FindBy to locate the link 'Stocks'. Click method is used to click on this link. PageFactory is used in the constructor of this class to initialize elements.

The class StocksQuotePage uses the annotation FindBy to locate the link 'A'. Click method is used to click on this link. PageFactory is used in the constructor of this class to initialize elements.

The class CompanyPage uses the annotation FindByAll to locate the all the web links under the displayed table and are stored in a list. The link for 'Ashok Ley land' company is also located . The method 'printLinks' to find the total number of rows and columns in the table. In side a for loop each company link is located using xpath and then company link and link text are displayed on console. The method clickonCompany is used to click on the link for AshokLeyland using click method and title of the page is returned..

The annotation @Test is used with the class DisplayLinks which is defined with method printLinks that uses get method is to launch the web application for money control and to instantiable objects for above classes and calling above mentioned methods as required. Assert is used to verify the page title.

The testngDisplaylinks.xml is defined with class name and test name.