**Project Name: Featured News**

Project Documentation:

Code Implementation:

Package Name: pageObjects

BasePage.java:

**package** PageObjects;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.support.PageFactory;

**public** **class** BasePage {

**protected** WebDriver driver; //Initialize the driver

// Constructor of the driver

**public** BasePage(WebDriver driver) {

**this**.driver = driver;

PageFactory.*initElements*(driver, **this**);

}

BeCognizantPage.java:

package PageObjects;

import java.time.Duration;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.WebDriverWait;

public class BeCognizantPage extends BasePage{

//Constructor

public BeCognizantPage(WebDriver driver) {

super(driver);

// TODO Auto-generated constructor stub

}

//Elements

//profile

@FindBy(id="O365\_MainLink\_MePhoto")

WebElement imgtag;

//User-name element

@FindBy(xpath = "//div[@class='mectrl\_dropdownbody expanded']//div[@class = 'mectrl\_name mectrl\_truncate']")

WebElement txt\_userName;

//Email element

@FindBy(xpath = "//div[@class='mectrl\_dropdownbody expanded']//div[@class = 'mectrl\_truncate']")

WebElement txt\_userEmail;

//news header

@FindBy(xpath="//div[@class='z\_a\_91bed31b l\_a\_91bed31b']//a[@data-automation-id='newsItemTitle']")

List<WebElement> list\_news;

//Actions

//click On User-Info-button

public void ClickOnProfile() throws InterruptedException

{

WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(30));

WebElement user= wait.until(ExpectedConditions.visibilityOfElementLocated((By.id("O365\_MainLink\_MePhoto"))));

wait.until(ExpectedConditions.visibilityOfElementLocated((By.xpath("//\*[@id='O365\_MainLink\_Settings']/div"))));

Thread.sleep(2000);

imgtag.click();

}

//check user-name is visible or not

public String userNameVisibility()

{

try {

String userName = txt\_userName.getText();

System.out.println("Username:" +userName);

return userName;

}

catch(Exception e)

{

return(e.getMessage());

}

}

//check user-Email is visible or not

public String userEmailVisibility()

{

try {

String userEmail = txt\_userEmail.getText();

System.out.println("Useremail:" +userEmail);

return userEmail;

}

catch(Exception e)

{

return(e.getMessage());

}

}

//check news header & tool-tip

public void newsHeaderVisibility()

{

// System.out.println("News Headline"+" "+"Tool-tip");

for(WebElement e:list\_news)

{

System.out.println("News Headline: "+e.getText());

System.out.println("Tooltip: "+e.getAttribute("title"));

}

}

}

NewsPage.java:

**package** PageObjects;

**import** java.io.IOException;

**import** java.time.Duration;

**import** java.util.List;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.JavascriptExecutor;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.interactions.Actions;

**import** org.openqa.selenium.support.FindBy;

**import** org.openqa.selenium.support.ui.ExpectedConditions;

**import** org.openqa.selenium.support.ui.WebDriverWait;

**import** Utility.ExcelUtils;

//import io.reactivex.rxjava3.functions.Action;

//import org.openqa.selenium.support.FindBy;

**public** **class** NewsPage **extends** BasePage {

//Constructor

**public** NewsPage(WebDriver driver) {

**super**(driver);

// **TODO** Auto-generated constructor stub

}

//elements

//@FindBy(xpath="//a[normalize-space() ='One year of Bluebolt: Advancing our spirit of innovation']")

//click on each news

@FindBy(xpath="//a[@id='news\_text\_title']")

List<WebElement> eachNews;

//News-Header

@FindBy(xpath="//div[@id='title\_text']")

WebElement txt\_heading;

//News-Heading-Details

@FindBy(xpath="//div[@data-automation-id='textBox']")

WebElement txt\_headingDetails;

//Associate details hover

@FindBy(xpath="//div[@class='lk\_ae\_9f38462c']")

WebElement associateDetails;

Actions action = **new** Actions(driver);

//HyperLink

@FindBy(tagName="a")

List<WebElement> hyperlink;

//Share-Button

@FindBy(xpath="//button[@data-automation-id='shareButton']")

WebElement sharebtn;

//Share-Option

@FindBy(xpath="//span[text()='Share page']/ancestor::ul")

WebElement shareOptions;

//Scroll-Body

@FindBy(xpath="//div[@class='fw\_aa\_9f38462c fw\_ap\_9f38462c mx\_ap\_9f38462c']")

WebElement scrollbody;

//likes and views

@FindBy(xpath="//\*[@id='vpc\_Page.CommentsWrapper.internal.2610b4b4-550c-4548-b431-aa3ab709c184']/div/div/aside/div[1]/button/span/span/span")

WebElement totalLikes;

@FindBy(xpath="//\*[@id='vpc\_Page.CommentsWrapper.internal.2610b4b4-550c-4548-b431-aa3ab709c184']/div/div/aside/div[4]/button/span/span/span")

WebElement totalViews;

JavascriptExecutor jse = (JavascriptExecutor) driver;

//Actions

//click each news

**public** **void** clickEachNews(**int** newsno) **throws** IOException

{

eachNews.get(newsno).click();

ExcelUtils excel = **new** ExcelUtils();

String xlFile = System.*getProperty*("user.dir")+"\\testData\\testExcel.xlsx";

**for**(**int** j=0;j<eachNews.size();j++)

{

excel.setCellData(xlFile, "Sheet1", j+1, 0,txt\_heading.getText());

}

}

**public** **void** showNewsDetails() **throws** InterruptedException, IOException

{

//news header

System.***out***.println("Heading: "+txt\_heading.getText());

System.***out***.println("Details: "+txt\_headingDetails.getText());

//associate details

action.moveToElement(associateDetails).perform();

WebDriverWait wait = **new** WebDriverWait(driver, Duration.*ofSeconds*(10));

wait.until(ExpectedConditions.*visibilityOfElementLocated*(By.*xpath*("//div[@class='\_6sw9u lcpUo kc69i']")));

//validation of hyperlink

**for**(WebElement link:hyperlink)

{

String url = link.getAttribute("href");

**if**(url== **null** || url.isEmpty())

{

System.***out***.println(url);

System.***out***.println("url is empty");

}

**else**

{

System.***out***.println(url);

System.***out***.println("url is not empty");

}

}

//check share button is click-able or not

sharebtn.click();

//Thread.sleep(2000);

//check share options is visible or not

System.***out***.println(shareOptions.getText());

//check likes and views

WebElement ele = driver.findElement(By.*xpath*("//\*[@id='CommentsWrapper']"));

jse.executeScript("arguments[0].scrollIntoView();", ele);

Thread.*sleep*(3000);

System.***out***.println("Number of Likes: " + totalLikes.getText().split(" ")[0]);

System.***out***.println("Number of Views : " + totalViews.getText().split(" ")[0]);

System.***out***.println();

Thread.*sleep*(2000);

driver.navigate().back();

Thread.*sleep*(2000);

}

//count no of News pages

**public** **int** newsSize()

{

**return** eachNews.size();

}

**public** String getTxtHeading()

{

**return** txt\_heading.getText();

}

//Get back to previous News page

**public** **void** getBack()

{

driver.navigate().back();

}

}

Testing-File

Package Name: testBase

BaseClass.java:

**package** testBase;

**import** java.io.FileReader;

**import** java.io.IOException;

**import** java.time.Duration;

**import** java.util.Properties;

**import** java.util.Scanner;

**import** org.openqa.selenium.WebDriver;

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

**import** org.openqa.selenium.edge.EdgeDriver;

**import** org.testng.annotations.AfterTest;

**import** org.testng.annotations.BeforeTest;

**import** freemarker.log.Logger;

**public** **class** BaseClass {

**public** **static** WebDriver *driver*;

**public** Properties p;

**public** **int** choice;

**public** Logger logger;

Scanner sc;

@BeforeTest

**public** **void** setup() **throws** IOException

{

//loading properties file

FileReader file = **new** FileReader(".//src//test//resources//config.properties");

p = **new** Properties();

p.load(file);

sc = **new** Scanner(System.***in***);

//choosing browser

System.***out***.println("Press 1 : Chrome ");

System.***out***.println("Press 2 : Edge ");

System.***out***.println("Press 3 : Exit ");

choice = sc.nextInt();

//Switch Case to Choose the browser

**switch**(choice) {

**case** 1:

*driver* = **new** ChromeDriver(); // Launch Chrome

**break**;

**case** 2:

*driver* = **new** EdgeDriver(); // Launch Edge

**break**;

**case** 3:

System.***out***.println("Program Closed");

System.*exit*(0);

**break**;

**default**:

System.***out***.println("Please Select the correct choice");

System.*exit*(0);

}

*driver*.manage().timeouts().implicitlyWait(Duration.*ofSeconds*(30));

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

*driver*.get(p.getProperty("appURL"));

}

//Quit the browser

@AfterTest

**public** **void** tearDown()

{

*driver*.quit();

}

}

Package Name: testCases

TC\_001\_BeCognizantPage.java

**package** testCases;

**import** java.io.IOException;

**import** org.testng.Assert;

**import** org.testng.annotations.Test;

**import** PageObjects.BeCognizantPage;

**import** Utility.ScreenShots;

**import** testBase.BaseClass;

**public** **class** TC\_001\_BeCognizantPage **extends** BaseClass{

@Test

**public** **void** beCognizantValidation() **throws** InterruptedException, IOException

{

//profile validation

BeCognizantPage bcp = **new** BeCognizantPage(*driver*);

bcp.ClickOnProfile();

Thread.*sleep*(8000);

ScreenShots ss = **new** ScreenShots(*driver*);

String userName = bcp.userNameVisibility();

Assert.*assertEquals*(userName,p.getProperty("uname"));

ss.ScreenShot(userName);

String userEmail = bcp.userEmailVisibility();

Assert.*assertEquals*(userEmail, p.getProperty("uemail"));

ss.ScreenShot(userEmail);

//News Header and tool-tip validation

bcp.newsHeaderVisibility();

}

}

TC\_002\_NewsPage.java:

**package** testCases;

**import** java.io.IOException;

**import** org.testng.annotations.Listeners;

**import** org.testng.annotations.Test;

**import** PageObjects.NewsPage;

**import** Utility.ScreenShots;

**import** testBase.BaseClass;

@Listeners(Utility.ExtentReportManager.**class**)

**public** **class** TC\_002\_NewsPage **extends** BaseClass {

@Test

**public** **void** newsPageValidation() **throws** InterruptedException, IOException

{

ScreenShots ss = **new** ScreenShots(*driver*);

NewsPage nsp = **new** NewsPage(*driver*);

**int** newssize = nsp.newsSize();

**for**(**int** i=0;i<newssize-1;i++)

{

nsp.clickEachNews(i);

ss.ScreenShot("News"+i);

nsp.showNewsDetails();

}

}

}

Utility File:

Apache POI Connection:

**package** Utility;

**import** java.io.FileInputStream;

**import** java.io.FileOutputStream;

**import** java.io.IOException;

**import** org.apache.poi.ss.usermodel.CellStyle;

**import** org.apache.poi.ss.usermodel.DataFormatter;

**import** org.apache.poi.ss.usermodel.FillPatternType;

**import** org.apache.poi.ss.usermodel.IndexedColors;

**import** org.apache.poi.xssf.usermodel.XSSFCell;

**import** org.apache.poi.xssf.usermodel.XSSFRow;

**import** org.apache.poi.xssf.usermodel.XSSFSheet;

**import** org.apache.poi.xssf.usermodel.XSSFWorkbook;

**public** **class** ExcelUtils {

**public** **static** FileInputStream *fi*; // FileInputStream Global Variable

**public** **static** FileOutputStream *fo*; // FileOutputStream Global Variable

**public** **static** XSSFWorkbook *wb*; // Excel Workbook Global Variable

**public** **static** XSSFSheet *ws*; // Excel Sheet Global Variable

**public** **static** XSSFRow *row*; // Excel Row Global Variable

**public** **static** XSSFCell *cell*; // Excel Cell Global Variable

**public** **static** CellStyle *style*; // Excel style Global Variable

//count rows present in excel sheet

**public** **int** getRowCount(String xlfile,String xlsheet) **throws** IOException

{

*fi*=**new** FileInputStream(xlfile);

*wb*=**new** XSSFWorkbook(*fi*);

*ws*=*wb*.getSheet(xlsheet);

**int** rowcount=*ws*.getLastRowNum();

*wb*.close();

*fi*.close();

**return** rowcount;

}

// count total no. of cells in the row

**public** **int** getCellCount(String xlfile,String xlsheet,**int** rownum) **throws** IOException

{

*fi*=**new** FileInputStream(xlfile);

*wb*=**new** XSSFWorkbook(*fi*);

*ws*=*wb*.getSheet(xlsheet);

*row*=*ws*.getRow(rownum);

**int** cellcount=*row*.getLastCellNum();

*wb*.close();

*fi*.close();

**return** cellcount;

}

// read data from excel sheet method

**public** String getCellData(String xlfile,String xlsheet,**int** rownum,**int** colnum) **throws** IOException

{

*fi*=**new** FileInputStream(xlfile);

*wb*=**new** XSSFWorkbook(*fi*);

*ws*=*wb*.getSheet(xlsheet);

*row*=*ws*.getRow(rownum);

*cell*=*row*.getCell(colnum);

String data;

**try**

{

//data=cell.toString();

DataFormatter formatter = **new** DataFormatter();

data = formatter.formatCellValue(*cell*);

**return** data;

}

**catch** (Exception e)

{

data="";

}

*wb*.close();

*fi*.close();

**return** data;

}

// write data on excel sheet cells

**public** **void** setCellData(String xlfile,String xlsheet,**int** rownum,**int** colnum,String data) **throws** IOException

{

*fi*=**new** FileInputStream(xlfile);

*wb*=**new** XSSFWorkbook(*fi*);

*ws*=*wb*.getSheet(xlsheet);

*row*=*ws*.getRow(rownum);

*cell*=*row*.createCell(colnum);

*cell*.setCellValue(data);

*fo*=**new** FileOutputStream(xlfile);

*wb*.write(*fo*);

*wb*.close();

*fi*.close();

*fo*.close();

}

// filling green colour in cells

**public** **void** fillGreenColor(String xlfile,String xlsheet,**int** rownum,**int** colnum) **throws** IOException

{

*fi*=**new** FileInputStream(xlfile);

*wb*=**new** XSSFWorkbook(*fi*);

*ws*=*wb*.getSheet(xlsheet);

*row*=*ws*.getRow(rownum);

*cell*=*row*.getCell(colnum);

*style*=*wb*.createCellStyle();

*style*.setFillForegroundColor(IndexedColors.***LIGHT\_GREEN***.getIndex());

*style*.setFillPattern(FillPatternType.***SOLID\_FOREGROUND***);

*cell*.setCellStyle(*style*);

*fo*=**new** FileOutputStream(xlfile);

*wb*.write(*fo*);

*wb*.close();

*fi*.close();

*fo*.close();

}

//filling red colour in cells

**public** **void** fillRedColor(String xlfile,String xlsheet,**int** rownum,**int** colnum) **throws** IOException

{

*fi*=**new** FileInputStream(xlfile);

*wb*=**new** XSSFWorkbook(*fi*);

*ws*=*wb*.getSheet(xlsheet);

*row*=*ws*.getRow(rownum);

*cell*=*row*.getCell(colnum);

*style*=*wb*.createCellStyle();

*style*.setFillForegroundColor(IndexedColors.***RED***.getIndex());

*style*.setFillPattern(FillPatternType.***SOLID\_FOREGROUND***);

*cell*.setCellStyle(*style*);

*fo*=**new** FileOutputStream(xlfile);

*wb*.write(*fo*);

*wb*.close();

*fi*.close();

*fo*.close();

}

}

ExtentReport Code implementation:

**package** Utility;

**import** java.text.SimpleDateFormat;

**import** java.util.Date;

**import** java.util.List;

**import** org.testng.ITestContext;

**import** org.testng.ITestListener;

**import** org.testng.ITestResult;

**import** com.aventstack.extentreports.ExtentReports;

**import** com.aventstack.extentreports.ExtentTest;

**import** com.aventstack.extentreports.Status;

**import** com.aventstack.extentreports.reporter.ExtentSparkReporter;

**import** com.aventstack.extentreports.reporter.configuration.Theme;

**import** testBase.BaseClass;

**public** **class** ExtentReportManager **extends** BaseClass **implements** ITestListener{

**public** ExtentSparkReporter sparkReporter; // UI of the report

**public** ExtentReports extent; //populate common info on the report

**public** ExtentTest test; // creating test case entries in the report and update status of the test methods

**public** **void** onStart(ITestContext context) {

String timeStamp = **new** SimpleDateFormat("yyyy.MM.dd.HH.mm.ss").format(**new** Date());// time stamp

String repName = "Test-Report-" + timeStamp + ".html";

sparkReporter = **new** ExtentSparkReporter(".\\reports\\" + repName);// specify location of the report

sparkReporter.config().setDocumentTitle("Automation Report"); // TiTle of report

sparkReporter.config().setReportName("Functional Testing"); // name of the report

sparkReporter.config().setTheme(Theme.***DARK***); // Theme of the report

extent=**new** ExtentReports();

extent.attachReporter(sparkReporter);

extent.setSystemInfo("Environment","https://be.cognizant.com");

extent.setSystemInfo("Tester Name","Moupali Sen(2317616)");

extent.setSystemInfo("os","Windows11");

extent.setSystemInfo("Browser name","Chrome,Edge");

List<String> includedGroups = context.getCurrentXmlTest().getIncludedGroups();

**if** (!includedGroups.isEmpty()) {

extent.setSystemInfo("Groups", includedGroups.toString());

}

}

**public** **void** onTestSuccess(ITestResult result) {

test = extent.createTest(result.getName()); // create a new entry in the report

test.assignCategory(result.getMethod().getGroups()); // to display groups in report

test.log(Status.***PASS***, "Test case PASSED is:" + result.getName()); // update status p/f/s

**try** {

String s = **new** ScreenShots(*driver*).ScreenShot(result.getName());

test.addScreenCaptureFromPath(s);

}**catch**(Exception e) {

e.printStackTrace();

}

}

**public** **void** onTestFailure(ITestResult result) {

test = extent.createTest(result.getName());

test.assignCategory(result.getMethod().getGroups()); // to display groups in report

test.log(Status.***FAIL***, "Test case FAILED is:" + result.getName());

test.log(Status.***FAIL***, "Test Case FAILED cause is: " + result.getThrowable());

**try** {

String s = **new** ScreenShots(*driver*).ScreenShot(result.getName());

test.addScreenCaptureFromPath(s);

}**catch**(Exception e) {

e.printStackTrace();

}

}

**public** **void** onTestSkipped(ITestResult result) {

test = extent.createTest(result.getName());

test.assignCategory(result.getMethod().getGroups()); // to display groups in report

test.log(Status.***SKIP***, "Test case SKIPPED is:" + result.getName());

}

**public** **void** onFinish(ITestContext context) {

extent.flush();

}

}

ScreenShots:

**package** Utility;

**import** java.io.File;

**import** java.io.IOException;

**import** org.apache.commons.io.FileUtils;

**import** org.openqa.selenium.OutputType;

**import** org.openqa.selenium.TakesScreenshot;

**import** org.openqa.selenium.WebDriver;

**import** PageObjects.BasePage;

**public** **class** ScreenShots **extends** BasePage{

**public** ScreenShots(WebDriver driver) {

// **TODO** Auto-generated constructor stub

**super**(driver);

}

//Whole Page Screenshot

**public** String ScreenShot(String path) **throws** IOException {

String p = System.*getProperty*("user.dir")+"\\screenshots\\";

p+=path+".png";

TakesScreenshot ts = (TakesScreenshot)driver;

File src = ts.getScreenshotAs(OutputType.***FILE***);

File trg = **new** File(p);

FileUtils.*copyFile*(src, trg);

**return** p;

}

}

Testing XML:

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

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

<suite name=*"Suite"*>

<test thread-count=*"5"* name=*"Test"*>

<classes>

<class name=*"testCases.TC\_001\_BeCognizantPage"*/>

<class name=*"testCases.TC\_002\_NewsPage"*/>

</classes>

</test> <!-- Test -->

</suite> <!-- Suite -->

Config. properties:

appURL=https://be.cognizant.com

uname=Sen, Moupali (Contractor)

[uemail=2317616@cognizant.com](mailto:uemail=2317616@cognizant.com)

Dependencies:

Pom.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>Featured\_News</groupId>

<artifactId>Featured\_News</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-java</artifactId>

<version>4.15.0</version>

</dependency>

<!-- https://mvnrepository.com/artifact/io.github.bonigarcia/webdrivermanager -->

<dependency>

<groupId>io.github.bonigarcia</groupId>

<artifactId>webdrivermanager</artifactId>

<version>5.5.3</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.apache.poi/poi -->

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi</artifactId>

<version>5.2.2</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml -->

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi-ooxml</artifactId>

<version>5.2.2</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.apache.logging.log4j/log4j-core -->

<dependency>

<groupId>org.apache.logging.log4j</groupId>

<artifactId>log4j-core</artifactId>

<version>2.20.0</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.apache.logging.log4j/log4j-api -->

<dependency>

<groupId>org.apache.logging.log4j</groupId>

<artifactId>log4j-api</artifactId>

<version>2.20.0</version>

</dependency>

<!-- https://mvnrepository.com/artifact/commons-io/commons-io -->

<dependency>

<groupId>commons-io</groupId>

<artifactId>commons-io</artifactId>

<version>2.11.0</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.apache.commons/commons-lang3 -->

<dependency>

<groupId>org.apache.commons</groupId>

<artifactId>commons-lang3</artifactId>

<version>3.12.0</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.testng/testng -->

<dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

<version>7.7.1</version>

<scope>test</scope>

</dependency>

<!-- https://mvnrepository.com/artifact/com.aventstack/extentreports -->

<dependency>

<groupId>com.aventstack</groupId>

<artifactId>extentreports</artifactId>

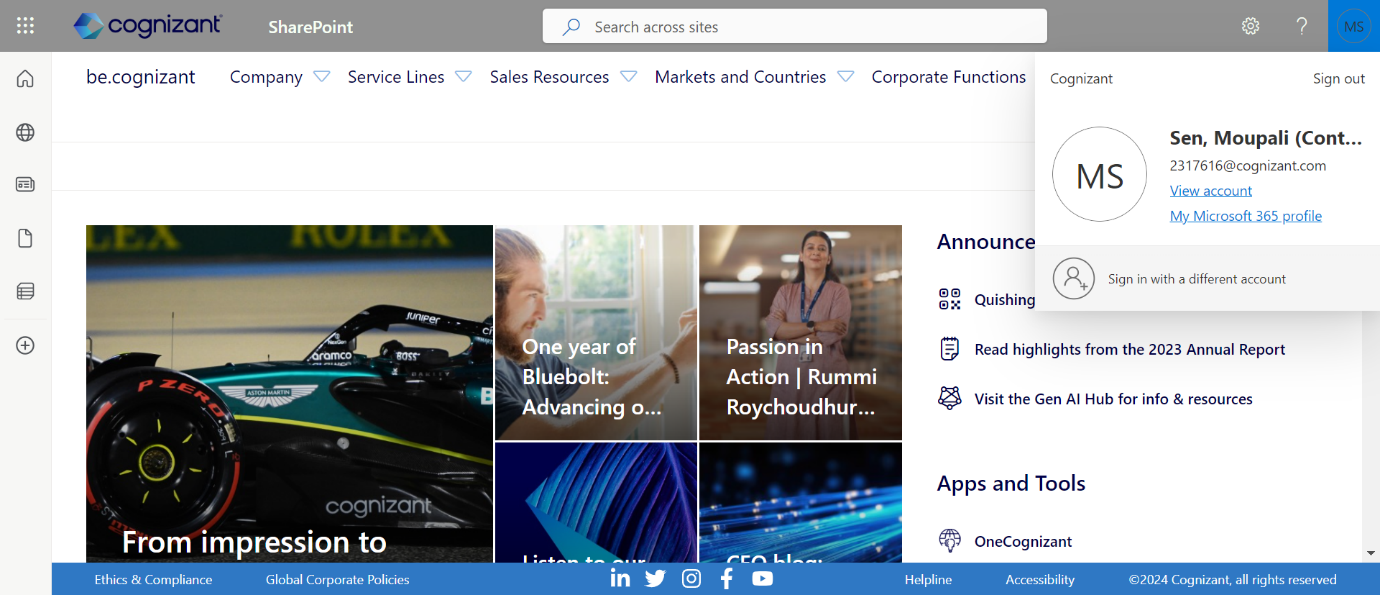
<version>5.0.9</version>

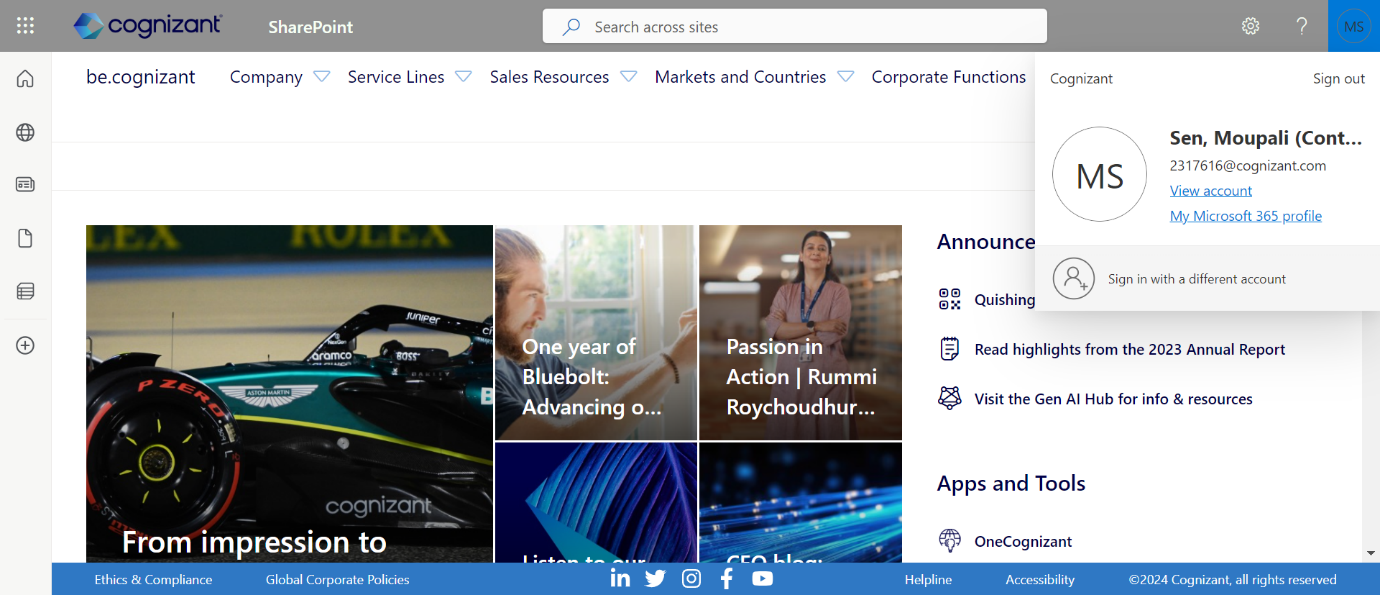
</dependency>

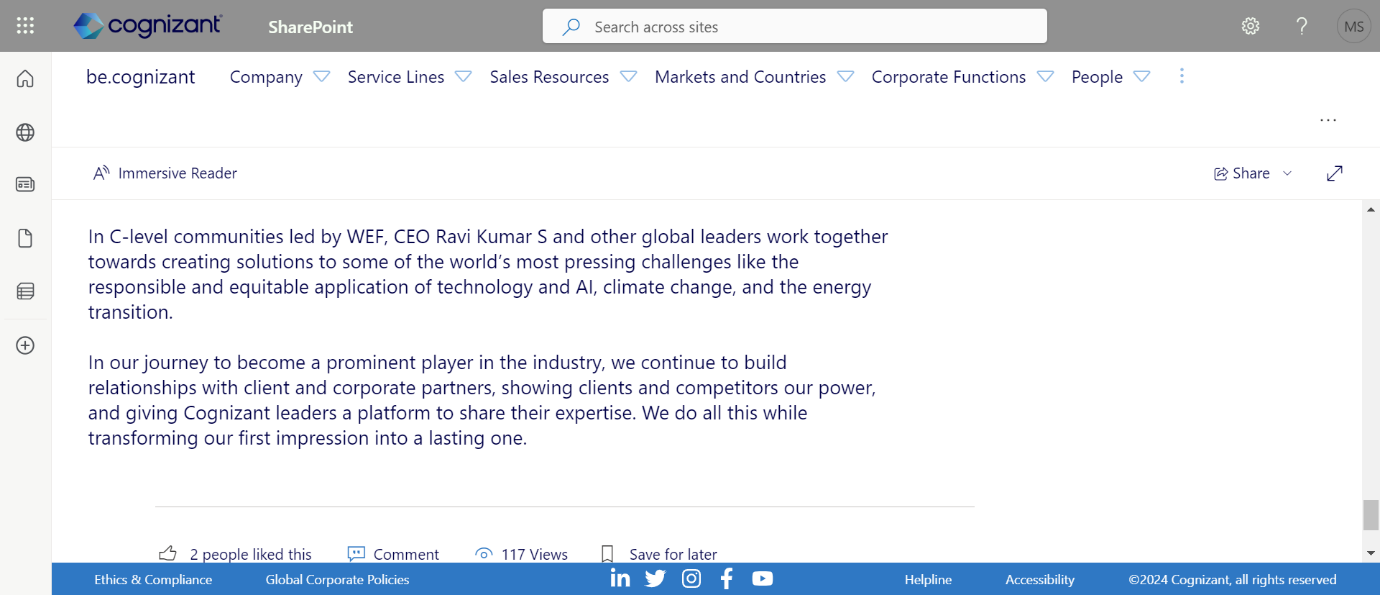
</dependencies>

</project>

Screenshots.png:







A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

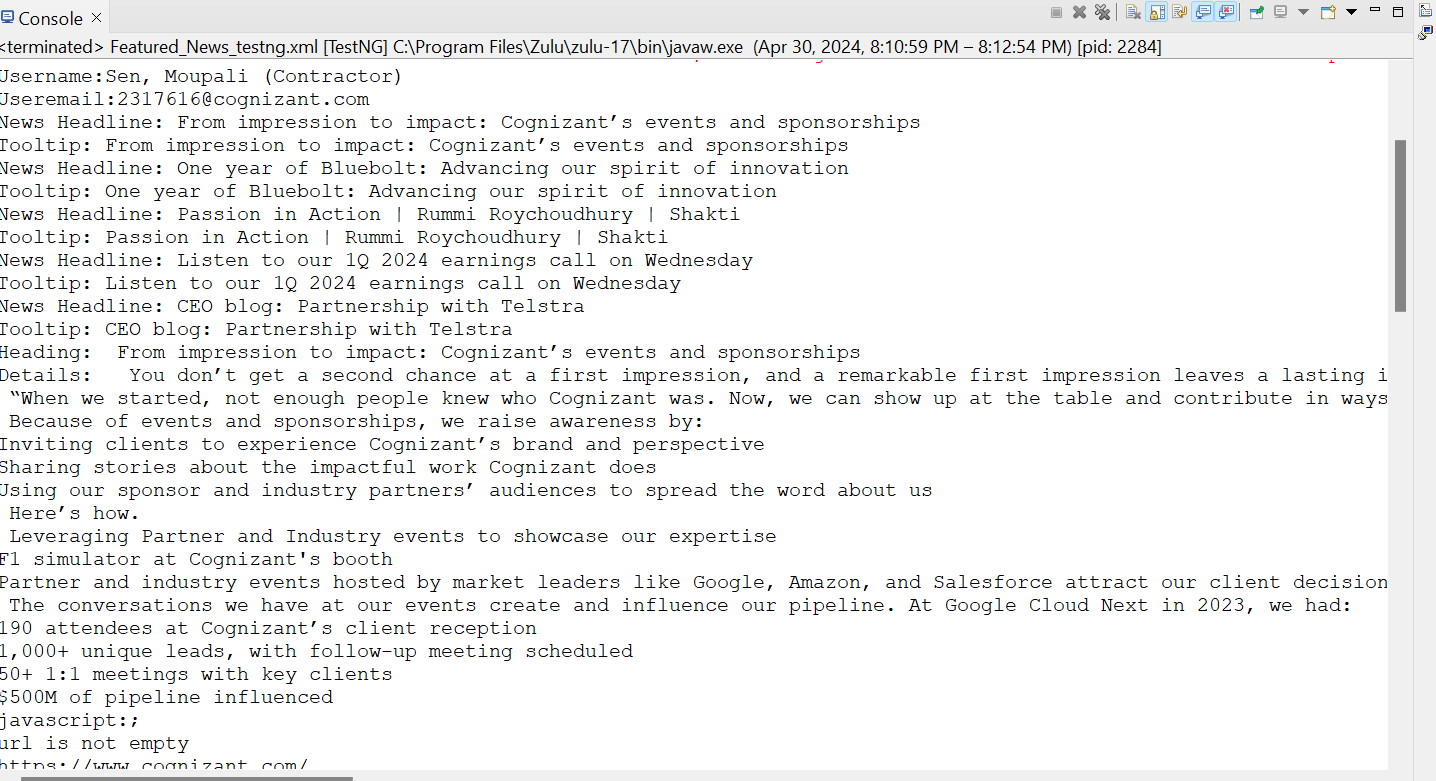
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Console Output:



A screenshot of a computer

Description automatically generated