## Project Name: Display Bookshelves

## Code Implementation:

## Package Name: pageObjects

## BasePage.java:

## package pageobjects;

## import org.openqa.selenium.WebDriver;

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

## public class BasePage {

## public WebDriver driver;

## public BasePage(WebDriver driver) {

## this.driver=driver;

## PageFactory.initElements(driver,this);

## }

## }

## GiftCardPage.java:

## package pageobjects;

## import java.io.IOException;

## import org.openqa.selenium.WebDriver;

## import org.openqa.selenium.WebElement;

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

## import utilities.ExcelUtils;

## import utilities.Takescreenshot;

## public class GiftCardPage extends BasePage {

## public GiftCardPage(WebDriver driver) {

## super(driver);

## // TODO Auto-generated constructor stub

## }

## 

## //Locating Elements

## @FindBy(xpath="//ul[@class='\_2sedU']/li[3]")

## public WebElement Birthday\_card;

## 

## 

## @FindBy(xpath="//input[@placeholder='Amount']")

## public WebElement Amount;

## 

## @FindBy(xpath="//div[@class='\_3PNvG']/select[2]")

## public WebElement date\_dropdown;

## 

## @FindBy(xpath="//button[text()='Next']")

## public WebElement NextBtn;

## 

## @FindBy(xpath="//input[@name='recipient\_name']")

## public WebElement Receipent\_Name;

## 

## @FindBy(xpath="//input[@name='recipient\_email']")

## public WebElement Receipent\_Email;

## 

## @FindBy(xpath="//input[@name='recipient\_mobile\_number']")

## public WebElement Receipent\_Mobile;

## 

## @FindBy(xpath="//input[@name='customer\_name']")

## public WebElement Customer\_Name;

## 

## @FindBy(xpath="//input[@name='customer\_email']")

## public WebElement Customer\_Email;

## 

## @FindBy(xpath="//input[@name='customer\_mobile\_number']")

## public WebElement Customer\_mobile;

## 

## @FindBy(xpath="//input[@name='customer\_address']")

## public WebElement Customer\_address;

## 

## @FindBy(xpath="//input[@name='zip']")

## public WebElement Customer\_pincode;

## 

## @FindBy(xpath="//button[@type='submit' and text()='Confirm']")

## public WebElement ConfirmBtn;

## 

## Takescreenshot ts;

## String xfile=System.getProperty("user.dir")+"\\TestData\\DisplayBookShelves.xlsx";

## //Action Methods

## 

## public void clickonGiftcard() throws IOException {

## ts=new Takescreenshot(driver);

## Birthday\_card.click();

## ts.ScreenShot("BithdayCard");

## }

## 

## public void entercarddetails(String amt) {

## Amount.sendKeys(amt);

## }

## public void clickonNextBtn()

## {

## NextBtn.click();

## }

## public void addressform(String rec\_name,String rec\_email,String rec\_mobile,String yr\_name,String yr\_email,

## String yr\_mobile,String address,String pin) throws InterruptedException, IOException {

## ts=new Takescreenshot(driver);

## Receipent\_Name.sendKeys(rec\_name);

## Receipent\_Email.sendKeys(rec\_email);

## Receipent\_Mobile.sendKeys(rec\_mobile);

## Customer\_Name.sendKeys(yr\_name);

## Customer\_Email.sendKeys(yr\_email);

## Customer\_mobile.sendKeys(yr\_mobile);

## Customer\_address.sendKeys(address);

## Customer\_pincode.sendKeys(pin);

## Thread.sleep(3000);

## ts.ScreenShot("AddressForm");

## ConfirmBtn.click();

## }

## public void addressform2(String rec\_email) {

## Receipent\_Email.clear();

## Receipent\_Email.sendKeys(rec\_email);

## ConfirmBtn.click();

## }

## public void captureerrormessage() throws IOException {

## ts=new Takescreenshot(driver);

## ts.ScreenShot("Errormessage");

## String error=Receipent\_Email.getAttribute("validationMessage");

## ExcelUtils.setCellData(xfile,"Sheet3", 1, 0,error);

## System.out.println(error);

## }

## }

HomePage.java:

package pageobjects;

import java.io.IOException;

import java.util.List;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.interactions.Actions;

import org.openqa.selenium.support.FindBy;

import org.testng.Assert;

import utilities.ExcelUtils;

import utilities.Takescreenshot;

public class HomePage extends BasePage {

public HomePage(WebDriver driver) {

super(driver);

// TODO Auto-generated constructor stub

}

//Locating Elements

@FindBy(xpath="//input[@id='search']")

public WebElement searchbar;

@FindBy(xpath="//button[@id='search\_button']")

public WebElement searchbtn;

@FindBy(xpath="//ul[@class='topnav bodytext']/li[contains(@class,'livingunit')]")

public WebElement Living\_tab;

@FindBy(xpath="//ul[@class='featuredLinksBar\_\_linkContainer']/li[3]/a")

public WebElement Giftcards;

@FindBy(xpath="//a[@href='javascript:void(0);' and @data-gaaction='popup.auth.close']")

public WebElement CloseButton;

@FindBy(xpath="//li[@class='topnav\_item livingunit']/div/div/ul/li/ul/li/a/span")

public List<WebElement> CollectionItemList;

String title=driver.getTitle();

Takescreenshot ts;

String xfile=System.getProperty("user.dir")+"\\TestData\\DisplayBookShelves.xlsx";

//Action Methods

public void verifypagetitle() {

Assert.assertEquals(title,"Buy Furniture Online and Get up to 50% Off | Shop Now - Urban Ladder");

}

public void Search(String key) throws IOException {

ts=new Takescreenshot(driver);

searchbar.sendKeys(key);

ts.ScreenShot("HomePage");

searchbtn.click();

}

public void Close() {

CloseButton.click();

}

public void hoveronlivingtab() throws IOException, InterruptedException {

ts=new Takescreenshot(driver);

Actions act=new Actions(driver);

act.moveToElement(Living\_tab).perform();

Thread.sleep(2000);

ts.ScreenShot("LivingTabs");

for(int i=0;i<11;i++) {

ExcelUtils.setCellData(xfile,"Sheet1", i+1, 0,CollectionItemList.get(i).getText());

System.out.println(CollectionItemList.get(i).getText());

}

}

public void clickonGiftcard() throws IOException {

ts=new Takescreenshot(driver);

Giftcards.click();

ts.ScreenShot("GiftCards");

}

}

SearchResults.java:

package pageobjects;

import java.io.IOException;

import java.time.Duration;

import java.util.List;

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 utilities.ExcelUtils;

public class SearchResults extends BasePage {

public SearchResults(WebDriver driver) {

super(driver);

// TODO Auto-generated constructor stub

}

//Locating Elements

@FindBy(xpath="//input[@id='filters\_availability\_In\_Stock\_Only']")

public WebElement ExcludeOutOfStockchkbx;

@FindBy(xpath="//li[@data-group='category']")

public WebElement Categories;

@FindBy(xpath="//input[@id='filters\_primary\_category\_Kids\_Bookshelves']")

public WebElement Kids\_Bookshelves;

@FindBy(xpath="//li[@data-group='price']")

public WebElement Price;

@FindBy(xpath="//div[@data-group='sorting']")

public WebElement SortBy;

@FindBy(xpath="//li[@class='option' and @data-key='price\_desc']")

public WebElement HightoLow;

@FindBy(xpath="//div[@class='product-info-block']/a/div/span[@itemprop='name']")

public List<WebElement> products\_name;

@FindBy(xpath="//div[@class='price-number']/span")

public List<WebElement> products\_price;

@FindBy(xpath="//div[@class='noUi-origin'][2]")

public WebElement UnderPrice;

@FindBy(xpath="//section[@class='header\_\_topBar\_sectionLeft']/a")

public WebElement Logo;

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

String xfile=System.getProperty("user.dir")+"\\TestData\\DisplayBookShelves.xlsx";

//Action Methods

public void SlidePrice() {

Actions act = new Actions(driver);

act.moveToElement(Price).perform();

mywait.until(ExpectedConditions.visibilityOf(UnderPrice));

act.moveToElement(UnderPrice).dragAndDropBy(UnderPrice, -215, 0).perform();

JavascriptExecutor js = (JavascriptExecutor)driver;

js.executeScript("arguments[0].setAttribute['style','left: 32.1767%;']",UnderPrice);

}

public void selectcategory() {

Actions act=new Actions(driver);

act.moveToElement(Categories).perform();

JavascriptExecutor js=(JavascriptExecutor)driver;

js.executeScript("arguments[0].click()",Kids\_Bookshelves);

}

public void outofstock() {

ExcludeOutOfStockchkbx.click();

}

public void sortbyfilter() {

Actions act=new Actions(driver);

act.moveToElement(SortBy).perform();

HightoLow.click();

}

public void displayproducts() throws IOException {

for(int i=0;i<3;i++) {

ExcelUtils.setCellData(xfile,"Sheet2", i+1, 0,products\_name.get(i).getText());

ExcelUtils.setCellData1(xfile,"Sheet2", i+1, 1,products\_price.get(i).getText());

System.out.println(products\_name.get(i).getText()+":"+products\_price.get(i).getText());

}

}

public void navigatetohomepage() {

JavascriptExecutor js=(JavascriptExecutor)driver;

js.executeScript("arguments[0].click()",Logo);

}

}

Test-File:

BaseClass.java:

package testbase;

import java.io.FileReader;

import java.io.IOException;

import java.time.Duration;

import java.util.Properties;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.chrome.ChromeOptions;

import org.openqa.selenium.edge.EdgeDriver;

import org.openqa.selenium.edge.EdgeOptions;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Parameters;

public class BaseClass {

public static WebDriver driver;

public Properties p;

//Launching the browers one by one

@BeforeTest

@Parameters("browser")

public void setup(String br) throws InterruptedException, IOException {

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

p=new Properties();

p.load(file);

ChromeOptions chromeOptions=new ChromeOptions();

chromeOptions.setExperimentalOption("excludeSwitches",new String[] {"enable-automation"});

chromeOptions.addArguments("--disable-notifications");

EdgeOptions edgeOptions=new EdgeOptions();

edgeOptions.setExperimentalOption("excludeSwitches",new String[] {"enable-automation"});

edgeOptions.addArguments("--disable-notifications");

//launching browser based on condition - locally

switch(br.toLowerCase())

{

case "chrome": driver=new ChromeDriver(chromeOptions); break;

case "edge": driver=new EdgeDriver(edgeOptions); break;

default: System.out.println("No matching browser..");

}

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

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

driver.get(p.getProperty("Url"));

}

//Closing the browser

@AfterTest

public void teardown() {

driver.quit();

}

}

Test Cases:

TC001\_HomePage.java:

package testcases;

import java.io.IOException;

import org.testng.annotations.\*;

import pageobjects.HomePage;

import testbase.BaseClass;

public class TC001\_HomePage extends BaseClass{

HomePage hm;

@Test(priority=1)

public void verifyPageTitle() {

hm=new HomePage(driver);

hm.verifypagetitle();

}

@Test(priority=2)

public void SearchBar() throws InterruptedException, IOException {

hm=new HomePage(driver);

hm.Search(p.getProperty("SearchKey"));

Thread.sleep(3000);

// hm.Dismissalert();

hm.Close();

}

@Test(priority=6)

public void HoverandclickonLivingTab() throws IOException, InterruptedException {

hm=new HomePage(driver);

hm.hoveronlivingtab();

}

@Test(priority=7)

public void clickonGiftcardsTab() throws IOException {

hm=new HomePage(driver);

hm.clickonGiftcard();

}

}

TC002\_SearchPageResults:

package testcases;

import java.io.IOException;

import org.testng.annotations.\*;

import pageobjects.SearchResults;

import utilities.Takescreenshot;

public class TC002\_SearchPageResults extends TC001\_HomePage {

SearchResults sr;

Takescreenshot ts;

@Test(priority=3)

public void applyfilters() throws InterruptedException, IOException {

sr=new SearchResults(driver);

ts=new Takescreenshot(driver);

Thread.sleep(2000);

ts.ScreenShot("SearchResults");

Thread.sleep(2000);

sr.outofstock();

sr.selectcategory();

Thread.sleep(2000);

sr.SlidePrice();

Thread.sleep(2000);

sr.sortbyfilter();

ts.ScreenShot("SearchResultsFiltered");

}

@Test(priority=4)

public void DisplaySearchResults() throws InterruptedException, IOException {

sr=new SearchResults(driver);

sr.displayproducts();

}

@Test(priority=5)

public void navigatebacktohomepage() {

sr=new SearchResults(driver);

sr.navigatetohomepage();

}

}

TC003\_GiftcardsPage:

package testcases;

import java.io.IOException;

import org.testng.annotations.Test;

import pageobjects.GiftCardPage;

public class TC003\_GiftcardsPage extends TC002\_SearchPageResults{

GiftCardPage gp;

@Test(priority=8)

public void choosegiftcard() throws IOException {

gp=new GiftCardPage(driver);

gp.clickonGiftcard();

}

@Test(priority=9)

public void entergiftcarddetails() {

gp=new GiftCardPage(driver);

gp.entercarddetails(p.getProperty("Amount"));

gp.clickonNextBtn();

}

@Test(priority=10)

public void filladdressform() throws InterruptedException, IOException {

gp=new GiftCardPage(driver);

gp.addressform(p.getProperty("Receipent\_Name"),p.getProperty("Receipent\_Email\_invalid"),p.getProperty("Receipent\_mobile"), p.getProperty("Customer\_Name"),p.getProperty("Customer\_Email"),p.getProperty("Customer\_mobile"),p.getProperty("Address"),p.getProperty("Pincode"));

Thread.sleep(3000);

gp.captureerrormessage();

gp.addressform2(p.getProperty("Receipent\_Email\_valid"));

}

}

Package Name: Utility

ExcelUtils.java:

**package** utilities;

**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*;

**public** **static** FileOutputStream *fo*;

**public** **static** XSSFWorkbook *wb*;

**public** **static** XSSFSheet *ws*;

**public** **static** XSSFRow *row*;

**public** **static** XSSFCell *cell*;

**public** **static** CellStyle *style*;

//Get row count

**public** **static** **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;

}

//Get cell count

**public** **static** **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;

}

//Get cell data

**public** **static** 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 in cell

**public** **static** **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*.createRow(rownum);

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

*cell*.setCellValue(data);

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

*wb*.write(*fo*);

*wb*.close();

*fi*.close();

*fo*.close();

}

**public** **static** **void** setCellData1(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();

}

//Fill cell with green color

**public** **static** **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.***GREEN***.getIndex());

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

*cell*.setCellStyle(*style*);

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

*wb*.write(*fo*);

*wb*.close();

*fi*.close();

*fo*.close();

}

//Fill cell with red color

**public** **static** **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();

}

}

ExtentReportManager.java:

package utilities:

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=new ExtentSparkReporter(System.getProperty("user.dir")+ "/reports/myReport.html");//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);

extent=new ExtentReports();

extent.attachReporter(sparkReporter);

extent.setSystemInfo("Computer Name","localhost");

extent.setSystemInfo("Environment","QA");

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

extent.setSystemInfo("os","Windows10");

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 enty 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 imgPath=new Takescreenshot(driver).ScreenShot(result.getName());

test.addScreenCaptureFromPath(imgPath);

}catch(Exception e) {

e.printStackTrace();

}

}

public void onTestFailure(ITestResult result) {

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

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

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

}

public void onTestSkipped(ITestResult result) {

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

test.assignCategory(result.getMethod().getGroups());

test.log(Status.SKIP, result.getName() + " got skipped");

test.log(Status.INFO, result.getThrowable().getMessage());

}

public void onFinish(ITestContext context) {

extent.flush();

}

}

Screenshots.java:

package utilities;

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 Takescreenshot extends BasePage {

public Takescreenshot(WebDriver driver) throws IOException {

super(driver);

// TODO Auto-generated constructor stub

}

//Taking screenshot of full webpage

public String ScreenShot(String path) throws IOException {

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

p+=path+".png";

//creating a object of TakeScreenshot interface

TakesScreenshot ts = (TakesScreenshot)driver;

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

File trg = new File(p);

//copying the src file to trg file

FileUtils.copyFile(src, trg);

return p;

}

}

Config. Properties:

Url=https://www.urbanladder.com/

SearchKey=BookShelves

Amount=5000

date=15

Receipent\_Name=Susmita

Receipent\_Email\_invalid=abjndgmail.com

Receipent\_Email\_valid=abc@gmail.com

Receipent\_mobile=9163807109

Customer\_Name=Moupali

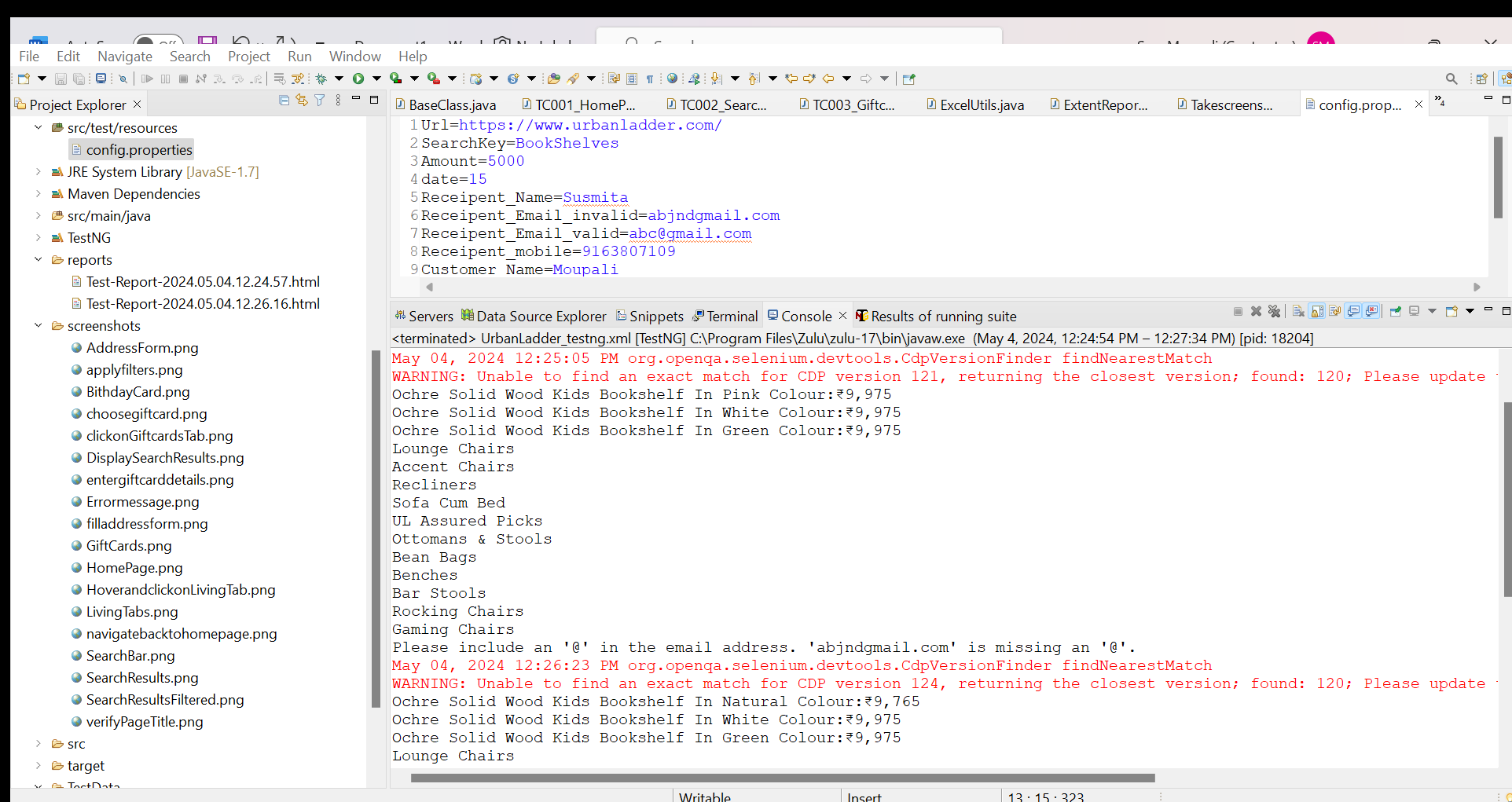
Customer\_Email=xyz21@gmail.com

Customer\_mobile=9748406309

Address=kolkata

Pincode=700013

Console Output:



Screenshots:

A screenshot of a gift card

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a website

Description automatically generated