package Action\_Items;  
  
import Reusable\_Library.Reusable\_Methods;  
import jxl.Sheet;  
import jxl.Workbook;  
import jxl.read.biff.BiffException;  
import jxl.write.Label;  
import jxl.write.WritableSheet;  
import jxl.write.WritableWorkbook;  
import jxl.write.WriteException;  
import org.openqa.selenium.WebDriver;  
import org.testng.annotations.AfterSuite;  
import org.testng.annotations.BeforeSuite;  
import org.testng.annotations.Test;  
  
import java.io.File;  
import java.io.IOException;  
import java.util.ArrayList;  
  
public class Assignment\_6 {  
  
 //declare all variables  
 WebDriver driver;  
 Workbook readableFile;  
 Sheet readableSheet;  
 WritableWorkbook writableFile;  
 WritableSheet writableSheet;  
 int rowCount;  
  
 @BeforeSuite  
 public void setDriver() throws IOException, BiffException {  
 //setting the chrome driver  
 driver = Reusable\_Methods.*getDriver*();  
 readableFile = Workbook.*getWorkbook*(new File("src/main/resources/Kayak\_AI.xls"));  
 readableSheet = readableFile.getSheet(0);  
 writableFile = Workbook.*createWorkbook*(new File("src/main/resources/Kayak\_AI\_Results.xls"),readableFile);  
 //Step 4: define the writable sheet for the writable file  
 writableSheet = writableFile.getSheet(0);  
 //Step 5: get the count of the rows that are not empty in excel workbook  
 rowCount = writableSheet.getRows();  
 }//end of set driver  
  
 @Test  
 public void totalCharges() throws InterruptedException, WriteException {  
  
 //navigate to kayak homepage  
 driver.navigate().to("https://www.kayak.com");  
 driver.manage().window().maximize();  
 Thread.*sleep*(2000);  
 String actualTitle = driver.getTitle();  
 if (actualTitle == "Kayak"){  
 System.*out*.println(actualTitle);  
 } else {  
 System.*out*.println("The actual title is " + actualTitle);  
 }//end of if else condition  
  
 for(int i = 1;i < rowCount;i++){  
 //I need to get content of the cell  
 //column is hard coded and row is dynamic(i)  
 String airport = writableSheet.getCell(0, i).getContents();  
 String startDate = writableSheet.getCell(1, i).getContents();  
 String startTime = writableSheet.getCell(2, i).getContents();  
 String endDate = writableSheet.getCell(3, i).getContents();  
 String endTime = writableSheet.getCell(4, i).getContents();  
 String vehicleSize = writableSheet.getCell(5, i).getContents();  
  
 driver.navigate().to("https://www.kayak.com");  
 Thread.*sleep*(2000);  
 /\* below code is inspected before you clicking inside the search field  
 and I am using contains id property as initial portion of id value  
 is changing \*/  
 Reusable\_Methods.*click*(driver, "//\*[contains(@id,'pickup-display')]", "Pick up Location");  
 Reusable\_Methods.*sendKeys*(driver, "//\*[@name='pickup']",airport,"Airport");  
 /\* click on suggesting link once you type your airport  
 orelse it will block your calendar so you need to click on suggested link \*/  
 Reusable\_Methods.*click*(driver, "//\*[@class='smarty-item-icon']", "Choose Airport Link");  
 //now put like 1 to 2 seconds then click on start date calendar icon  
 Thread.*sleep*(2000);  
 //line below is clicking on calendar icon first for start date  
 Reusable\_Methods.*click*(driver,"//\*[contains(@id,'dateRangeInput-display-start-inner')]","Start Date");  
 Thread.*sleep*(1000);  
 Reusable\_Methods.*typeAndSubmit*(driver,"//\*[contains(@id,'pickup-date-input')]",startDate,"Start Date");  
 Thread.*sleep*(1000);  
 /\* element below is not interactable because you have to close the calendar so click anywhere outside  
 like clicking on title search for hundred \*\*/  
 Reusable\_Methods.*click*(driver,"//\*[@class='title dark']","Dark Title");  
 //put like 1 second  
 Thread.*sleep*(1000);  
 //before you select the value from start time you need to click on start time icon first  
 Reusable\_Methods.*click*(driver, "//\*[contains(@id,'start-time-select-display')]","Start Time");  
 /\* also before clicking on the value you should put 1.5 to 2 seconds to slow down selenium  
 orlese you will get element not interactable error \*\*/  
 Thread.*sleep*(2500);  
 //now click on start time value  
 Reusable\_Methods.*mouseClick*(driver,"//\*[contains(@id,'start-time-select-option') and @data-title='"+startTime+"']","Start Time");  
 Reusable\_Methods.*click*(driver,"//\*[contains(@id,'dateRangeInput-display-end-inner')]","End Date");  
 Thread.*sleep*(1000);  
 Reusable\_Methods.*typeAndSubmit*(driver,"//\*[contains(@id,'dropoff-date-input')]",endDate,"Start Date");  
 Thread.*sleep*(1000);  
 Reusable\_Methods.*click*(driver,"//\*[@class='title dark']","Dark Title");  
 Thread.*sleep*(1000);  
 Reusable\_Methods.*click*(driver, "//\*[contains(@id,'end-time-select-display')]","End Time");  
 Thread.*sleep*(2500);  
 Reusable\_Methods.*mouseClick*(driver,"//\*[contains(@id,'end-time-select-option') and @data-title='"+endTime+"']","End Time");  
 Reusable\_Methods.*click*(driver,"//\*[@class='v-c-p centre ']","Search Button");  
 Thread.*sleep*(2000);  
 Reusable\_Methods.*click*(driver,"//\*[@class='MPti-Content']","Small");  
 Thread.*sleep*(5000);  
 Reusable\_Methods.*clickByIndex*(driver,"//\*[contains(text(),'View Deal')]",2,"View Deal");  
 //need to store window handles command in as ArrayList to switch to different tab(s)  
 Thread.*sleep*(4000);  
 ArrayList<String> tabs = new ArrayList<>(driver.getWindowHandles());  
 //switch to second tab  
 driver.switchTo().window(tabs.get(1));  
 Thread.*sleep*(5000);  
 String totalCharge = Reusable\_Methods.*capturingText*(driver,"//\*[@class='sc-VJcYb drkFuw' or @class='qeek-tip-trigger h\_total-price' or @class='nowrap' or @class='Checkout-Common-Book-Rightrail-FourPartItem total']","Total Charge");  
 driver.close();//to close the window if you don't need it again  
 //switch back to original tab  
 driver.switchTo().window(tabs.get(0));  
 driver.manage().deleteAllCookies();  
 //store and send it to the empty column in excel writable file  
 Label label = new Label(8, i, totalCharge);  
 //adding this label to writable sheet  
 writableSheet.addCell(label);  
  
 }//end of for loop  
  
 }//end of total charges  
  
 @AfterSuite  
 public void closingInfo() throws IOException, WriteException {  
 writableFile.write();  
 writableFile.close();  
 }//end of closing info  
  
  
  
  
}//end of class