Config.properties :

```
#Url: application website which we want to automate
url=https://relevel-ecomm-fe.herokuapp.com/

#Headless: mode in which we want to run the automation , yes to run the test
cases in background or headless mode ,
# No to run the test cases without headless mode
headless=No

#browser: chrome to run the test cases in chrome browser,
# safari to run the test case in safari browser
# Edge to run the test cases in edge browser
# Firefox to run the test cases in firefox browser
browser=CHROME

#implicitwait: time in seconds for selenium to search the web Elements
implicitWait = 10

#pageLoadTimeOut: time in seconds for the page to completely load
pageLoadTimeout = 20

#testDataFilePath: path of the test data file
testDataFilePath = src/main/resources/Electronics_Data.xlsx

#username : username of the login credentials
userName = test1

#password : password of the login credentials
password = test1
```

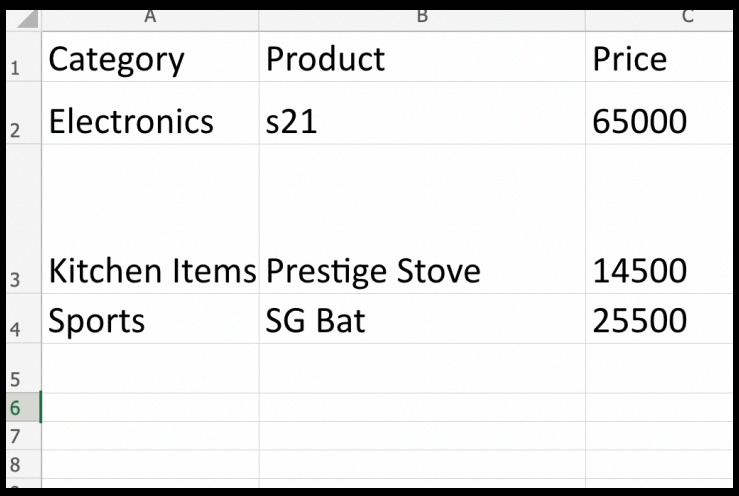
Log4j.properties:

```
# Root logger option
log4j.rootLogger=INFO, file, stdout

# Direct log messages to a log file
log4j.appender.file=org.apache.log4j.RollingFileAppender
log4j.appender.file.File=Testlog.log
log4j.appender.file.MaxFileSize=10MB
log4j.appender.file.MaxBackupIndex=10
log4j.appender.file.layout=org.apache.log4j.PatternLayout
log4j.appender.file.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p
%c{1}:%L - %m%n


# Direct log messages to stdout
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.Target=System.out
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p
%c{1}:%L - %m%n
```

Excel Sheet – Data File :



	A	B	C
1	Category	Product	Price
2	Electronics	s21	65000
3	Kitchen Items	Prestige Stove	14500
4	Sports	SG Bat	25500
5			
6			
7			
8			

Package DataProvider –> ConfigfileReader Class :

```
package dataProvider;

import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.util.Properties;

public class ConfigFileReader {

    private static Properties properties;

    private static final String propertyFilePath ="config.properties";

    static
    {
        BufferedReader bufferedReader;

        try
        {
            FileReader fileReader= new FileReader(propertyFilePath);
            bufferedReader = new BufferedReader(fileReader);
            properties = new Properties();

            try {
                properties.load(bufferedReader);
                bufferedReader.close();
            }
            catch (Exception e)
            {
                System.out.println("exception while reading the file:"+e);
            }

        }
        catch (FileNotFoundException e)
```

```
        {
            System.out.println("File not found at the defined location :"+e);
        }

    }

    public static String getUrl()
    {
        String url = properties.getProperty("url");
        if(url!=null)
            return url;

        else
            throw new RuntimeException("url is not specified in the
config.properties file");
    }

    public static String getMode()
    {
        String mode =properties.getProperty("headless");
        if(mode!=null)
            return mode;
        else
            throw new RuntimeException("headless mode is not defined in to
config.properties file");
    }

    public static String getBrowser()
    {
        String browser = properties.getProperty("browser");
        if(browser!=null)
            return browser;
        else
            throw new RuntimeException("browser is not defined in to
config.properties file");
    }

    public static String getImplicitWait()
    {
        String implicitWait = properties.getProperty("implicitWait");
        if(implicitWait!=null)
            return implicitWait;
        else
            throw new RuntimeException("implicit wait is not defined in to
config.properties file");
    }

  public static String getPageLoadTimeOut()
  {
     String explicitWait = properties.getProperty("pageLoadTimeout");
     if(explicitWait!=null)
         return explicitWait;
     else
         throw new RuntimeException("explicit wait is not defined in to
config.properties file");
  }

  public static String getTestDataFilePath()
  {
      String testDataFilePath = properties.getProperty("testDataFilePath");
      if(testDataFilePath!=null)
          return testDataFilePath;
      else
          throw new RuntimeException("test data file path is not defined in to
config.properties file");
  }
```

```
  public static String getUserName()
  {
      String userName = properties.getProperty("userName");
      if(userName!=null)
          return userName;
      else
          throw new RuntimeException("username is not defined in to
config.properties file");

  }

  public static String getPassword()
  {
      String password = properties.getProperty("password");
      if(password!=null)
          return password;
      else
          throw  new RuntimeException("password is not defined in to
config.properties file");
  }
}
```

Package DataProvider ->ReadWriteExcelClass:

```
package dataProvider;

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

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;

public class ReadWriteExcel {

    FileInputStream fileInputStream;
    XSSFWorkbook workbook;
    XSSFSheet sheet;

    public ReadWriteExcel() throws IOException {
        fileInputStream= new
FileInputStream(ConfigFileReader.getTestDataFilePath());
        workbook = new XSSFWorkbook(fileInputStream);
    }

    public XSSFSheet getSheet(String sheetName) {

        sheet = workbook.getSheet("Sheet1");
        return sheet;
    }
}
```

ObjectManagerPackage -> DriverManager Class :

```java
package ObjectManager;

import dataProvider.ConfigFileReader;
import io.github.bonigarcia.wdm.WebDriverManager;
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.firefox.FirefoxDriver;
import org.openqa.selenium.safari.SafariDriver;

import java.time.Duration;
import java.util.concurrent.TimeUnit;

public class DriverManager {

    private  static WebDriver driver;

    public static WebDriver getDriver()
    {
        if(driver==null)
            createDriver();
            return driver;

    }

    private static void createDriver() {

        switch (ConfigFileReader.getBrowser().toUpperCase())
        {
            case "CHROME":

                ChromeOptions options = new ChromeOptions();
                if(ConfigFileReader.getMode().equalsIgnoreCase("YES"))
                {
                    options.addArguments("--headless");
                }
                WebDriverManager.chromedriver().setup();
                driver=new ChromeDriver(options);
                System.out.println("created the chrome driver");
                break;

            case "FIREFOX" :
                WebDriverManager.firefoxdriver().setup();
                driver=new FirefoxDriver();
                System.out.println("created the firefox driver");
                break;

            case "SAFARI" :

                WebDriverManager.safaridriver().setup();
                driver= new SafariDriver();
                System.out.println("created the safari driver");
                break;

            case "EDGE" :
                WebDriverManager.edgedriver().setup();
                driver= new EdgeDriver();
                System.out.println("created the edge driver");
                break;
```

```
            default:
                System.out.println("no matching browser found");
                break;

        }

driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(Long.parseLong(Con
figFileReader.getImplicitWait())));

driver.manage().timeouts().pageLoadTimeout(Duration.ofSeconds(Long.parseLong(Co
nfigFileReader.getPageLoadTimeOut())));
        driver.manage().window().maximize();
        System.out.println("maximized the browser");

    }
}
```