I have tried to cover all the scenarios mention in the assignment. The following are the main components which I have used in Kennect assignment project.

* The programming language used in automation is JAVA
* Automation Tool Used :- Selenium
* Type of frame work – Test data driven frame work
* Test data= Excel File
* Config.properties file
* Design pattern= Page Object Model
* **packages**
* Utils (Common methods)
* Listeners (Failed Screen shots method)
* Config.properties =URL, Browser name, username & passwored.
* Reports=extent report
* TestNG file = Parallel execution
* Test runner file = to run all the test case methods with assertion
* Covered Cross browser testing on Edge
* For Run this project import in eclipse IDE and to setup system environment.
* After setup you can run by Testng.xml file for parallel execution as well as you can run **TestRunner class using TestNG**.
* Challenges:-I faced the challenges to choose some time in dropdowns like equipment, and add test as well as **click** on some of web elements to overcome this I uses Actions and JS classes.

**Explanation**  :-

1. **Language:** In our Selenium Project we are using Java language. Even though Selenium supports multiple languages, we are using Java language is just because most of the automation developers have knowledge on Selenium with Java.
2. **Type of Framework:** In our project, we are using Data-driven Framework by using Page Object Model design pattern (**POM**)with **Page Factory**.
3. **POM:** As per the Page Object Model, we have maintained a class for every web page. Each web page has a separate class and that class holds the functionality and members of that web page. Separate classes for every individual test.
4. **Packages**: We have separate packages for Pages and Tests. All the web page related classes come under the Pages package and all the tests related classes come under Tests package. For example, Home Page and Login Page have separate classes to store element locators. For the login test, there would be a separate class which calls the methods from the Home Page class and Login Page class.
5. **Test Base Class:** Test Base class (TestBase.java) deals with all the common functions used by all the pages. This class is responsible for loading the configurations from properties files, Initializing the WebDriver, Implicit Waits, Extent Reports, and also to create the object of FileInputStream() which is responsible for pointing towards the file from which the data should be read.
6. **Utility class** :Utility class (TestUtil.java) stores and handles the functions (The code which is repetitive in nature such as waits, actions, capturing screenshots, accessing excels, sending email, etc.,) which can be commonly used across the entire framework. The reason behind creating a utility class is to achieve reusability.
7. **Properties file**: This file (config.properties) stores the information that remains static throughout the framework such as browser-specific information, application URL, screenshots path, etc.
8. **Screenshots**: Screenshots will be captured and stored in a separate folder and also the screenshots of failed test cases will be added to the extent reports.
9. **Test Data**: All the historical test data will be kept in an excel sheet (controller.xlsx). By using ‘controller.xlsx’, we pass test data and handle datadriven testing. We use Apache POI to handle excel sheets.
10. **TestNG**: Using TestNG for Assertions, Grouping, and Parallel execution.
11. **Maven**: Using Maven for build, execution, and dependency purpose. Integrating the TestNG dependency in the POM.xml file and running this POM.xml file.
12. **version Control Tool**: We use Git as a repository to store our test scripts.
13. **Extent Reports**: For the reporting purpose, we are using Extent Reports. It generates beautiful HTML reports. We use the extent reports for maintaining logs and also to include the screenshots of failed test cases in the Extent Report.

Regards,

Neeraj Singh

Ndssp150@gmail.com/9930927788