|  |
| --- |
| File Run Process |
| 1. Download the project in zip format from Git repository |
| 2. Open the project GurukulaProject in Eclipse |
| 3. Expand src/main/java folder |
| 4. Expand com.org.runner package |
| 5. Double Click **runner.java** file |
| 6. Right click > select Run > JunitTest |

|  |
| --- |
| Execution Run Results |
| 1. After completing the execution |
| 2. Navigate to GurukulaProject\target\cucumber-html-report |
| 3. Open **index.html** file for execution results |

|  |
| --- |
| Logs: |
| 1. Navigate to GurukulaProject\log folder and clicking logging.log file |

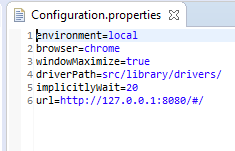
GurukulaScenarios.xlsx

- Please go through the features/ scenarios/ defects from this excel which is uploaded in the project folder

------------------------------------------------------------------------------------------

Project Details:

How it works?  
  
1. Provide the browse name, url in the config file (configs\configuration.properties)



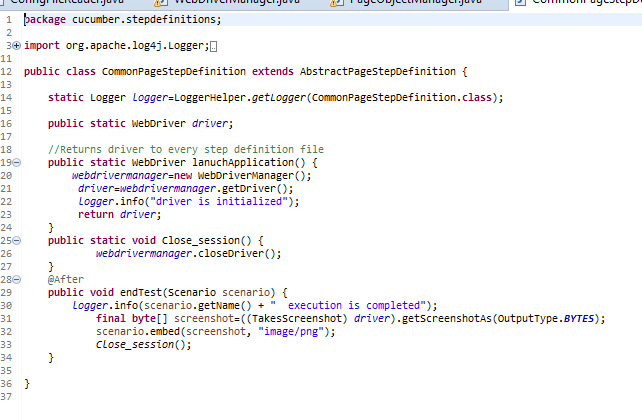
2. src/main/java/com.org.util/ConfigReader.java – Reads the browser name, driverpath, url details from configuration.properties file

3. src/main/java: Package com.org.managers / WebDriverManager.java

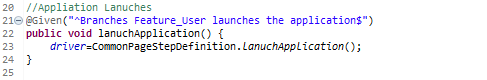
Creates the driver

4. src/mainjava/cucumber.stepdefinitions/CommonPageStepDefinition

This java file takes the driver from WebDriverManger and passes this driver to all the feature step definition files



5. Each step definition file would read the commonpagestepdefinition file to read the driver ex: below



===============================================================================

Project Framework:

- Build on cucumber framework with page object model

- Features files are resided in src/main/resources

* There are 11 feature files are created

- Step definition files resides in folder src/mainjava. Every step definition file relates to the feature which is created. It has one to one mapping.

- Pages

* As it is based on page object model. We have java files creaed for respective pages in the src/main/java folder under the package of com.org.pages
* Each step definition files would call the page objects to do the actions on that specific pages

- util package -

ApplicationFunctionLibrary- All common actions/reusable functions related to application specific things like password strength check are defined here

ConfigFileReader - driverpath, browsername are all read from the configs\config file through this java code

FunctionLibary: this has all reusable functions of clicking the button, reading the label text, providing input in the text field, reading drop down, webtable etc. All the java pages file would extend this library/java file to use the methods

LoggerHelper: This java class is created to read the details from configs/log4j.properties so that logs can be generated for the actions we do