Automation frame work (HYDBRID):

This frame work supports following features:

1. Able to run with multiple suites or single suite or single test case
2. Able to run in different browsers parallel (ie,ff,cr…etc)
3. Able to run in different machines parallel (system1, system2…etc)
4. Handling test data using properties file (We can also do different approaches like excel, xml , dataproviders…)
5. Handling failed test cases and show proper reports using Take screenshot and TestNG reports

(We can do xslt or customized reports too…)

1. Build the frame work using ANT tool. (We can do through Maven…)
2. Separation of Business logic and web elements to handle POM pattern

Frame work structure:

1. Created java project with 4 packages : test, or, impl, utility
2. Test – all the java test cases with annotations

Generic – All common functions

Branch , Staff, Account all are features files

Or – all the java classes for webelements

Utility – all the utilities used for frame work

Impl – implanted methods for this feature

1. Conf – properties file for test cases
2. Lib – placed all jars related to frame work
3. Logs – supports for log4j properties details
4. Resources
   1. Input

Xml – dynamically generated testng xml suite files

Driver.xls – having test case, browser , machines data to launch applicaton

* 1. O
  2. utput

FinalResult.xls – having test cases results

1. Bat – running bat file with threadpoolsize command to run multiple suites
2. Build – dynamically create this folder using ant target
3. Run the hub in server and node in each client machine.

Working Flow of execution:

1. Build.xml

Run with targets like cleanup, init, compile,jar,createXML,run,createresults

1. createXML target generate dynamically testng xml files using utitlity based on the data given in Driver.xls
2. run all test cases and suites from bat file
3. generate customized results using testng-output fodler

Configuration for further:

Jenkin and GitHub:

Configure jenkin with git, github plugins

Integrate github and jenking using ssh or https (private or public keys)

Configure systems like ant or maven configuration

Created cron schedules and mail to based on settings

**To run this project:**

1. **Run Gurukula project**
2. **Run hub and node java -jar selenium-server-standalone-2.48.2.jar -role hub**
3. **download Gurukula.zip from Github**
4. **import the project into eclipse**
5. **update the file paths in common.properties**
6. **Add jar files into lib folder**
7. **Update the Gurukula\resources\input\Driver.xls with your machine name**
8. **Keep the Gurukula\bat\main.bat file on desktop and run based on instructions(I am not considering any inputs that you give for bat file)**

**If getting any errors and not able to run Navigate to Gurukula\machine1.xml and run the testng suite.**