

Steps for Execution:

To execute test scripts please follow below steps:

1) Install Java & maven and set their respective paths in system variables.

2) Clone project (git clone <https://github.com/unkannan/GurukulaApplicationAutomation.git>)

3) Clean and compile project using

mvn clean compile

4) To execute automation scripts, execute any of the following commands from command prompt:

i. To execute all automation scripts (by default it will run on firefox)

mvn test

ii. To execute on particular browser, execute below command with parameter as 'firefox' or 'chrome'

mvn test -Dbrowser=<browser-name>

for e.g.: *mvn test -Dbrowser=firefox*

mvn test -Dbrowser=chrome

iii. To execute test scripts for particular functionality execute below command with parameter as [branch, login, misc, password, register, resetPassword, searchBranch, searchStaff, sessions, settings, staff, welcome]

mvn test -Dgroups=<group-name>

for e.g.: *mvn test -Dgroups=branch*

iv. To execute test scripts of any particular group on any particular browser

mvn test -Dbrowser=<browser-name> -Dgroups=<group-name>

for e.g.: *mvn test -Dbrowser=firefox -Dgroups=branch*

mvn test -Dbrowser=chrome -Dgroups=login

Scope of Testing:

In-Scope:

1) Functional Testing:

i. Manual and exploratory testing

ii. Automation testing

2) Non-Functional Testing:

i. UI testing performing various actions and also to confirm that application is easy to use.

ii. Cross-browser testing to verify that application works as intended on supported browsers.

Out of Scope:

1) Performance testing

2) Security testing

Test Strategy:

This section lists down the strategy to ensure the quality of Gurukula application via manual & automation testing.

- 1) Manually explore application to understand basic functionalities.
- 2) Initially identified the features in application.
- 3) Designed the test scenarios for testing of various features in application.
- 4) Identify automatable test scenarios and mark the test cases accordingly.
- 5) Design test scripts for automation testing
- 6) Execute both manual and automation tests against application in test.
- 7) Analyse test results to identify defects in application.

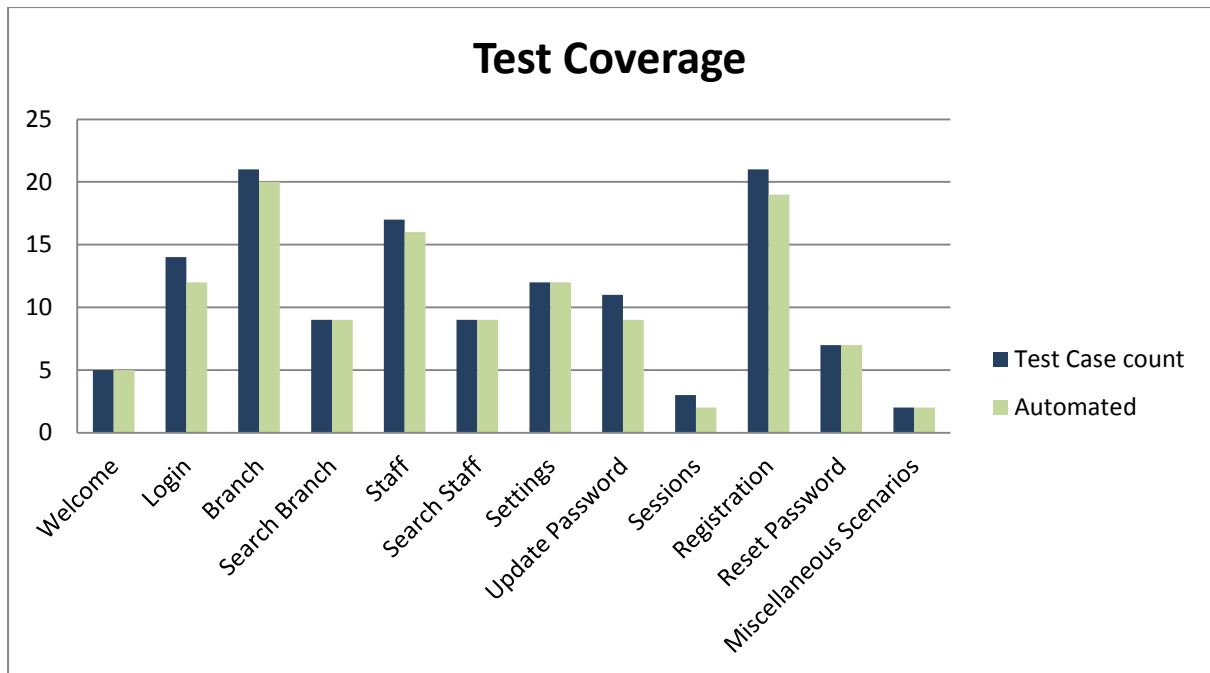
Automation tools used:

List of all the software and tools required for developing and executing automation framework.

- 1) Java (version 1.8)
- 2) Selenium (version 3.141.59)
- 3) TestNG (version 6.14.3)
- 4) Maven (version 4.0)
- 5) Log4j (version 1.2.17)
- 6) Google Chrome (version 76.0.3809.100)
- 7) Mozilla Firefox (version 68.0.2 (64-bit))
- 8) Internet Explorer 11 (version 11.0.9600.19326CO)

Test Deliverables:

1. Test Scenarios: For details refer "TestScenarios" sheet in GurukulaApplication Report.xlsx
2. Automation Feasibility: For details refer "TestScenarios" sheet in GurukulaApplication Report.xlsx



Test Coverage				
S.No.	Functionality	Test Case count	Automated	Percentage
F1	Welcome	5	5	100
F2	Login	14	12	85.71
F3	Branch	21	20	95.24
F4	Search Branch	9	9	100
F5	Staff	17	16	94.12
F6	Search Staff	9	9	100
F7	Settings	12	12	100
F8	Update Password	11	9	81.82
F9	Sessions	3	2	66.67
F10	Registration	21	19	90.48
F11	Reset Password	7	7	100
F12	Miscellaneous Scenarios	2	2	100
	Total	131	122	93.13

Automated Details: Automation test scripts are available at Github location <https://github.com/unkannan/GurukulaApplicationAutomation.git>.

4) Defect List: List of defects found while executing manual & automated test scenarios is reported in 'Defects' worksheet in " **GurukulaApplication Report.xlsx**".

Project Structure

- implemented in POM model (page object model)
- POM.xml – All dependencies are included in POM.xml
- drivers - folder have chrome, iexplore, firefox drivers
- log: auto-generated log file
- com.org: **package contains**
 - o enums – all enums project related as well the constant message used in the application
 - o pages - all repsecitve pages locators, methods used in the test methods
 - o tests - each file contains test the functionality end to end
 - o utility – to initiate driver, read properties file
 - o resources – properties used like url, username, password and log4j file to log

Reports:

1. Scenarios, defects are listed in the excel 'GurukulaApplication Report.xlsx'
2. Screenshots
 - Framework captures screenshot of every failed test scripts. All screenshots are stored at \target\screenshots.
3. TestNG report is available in \target\surefire-reports\index.html

