**Dr.D.Y.Patil Institute of Technology**

Pimpri, Pune-411018

A MINI-PROJECT REPORT

Selenium Automation testing On Library System

Submitted by

BCOA42 Anubhuti Rane

BCOA43 Sampada Gaonkar

**Department of Computer Engineering**

2019-2020

**Dr.D.Y.Patil Institute of Technology**

PIMPRI, PUNE-411018

**Department of Computer Engineering**

CERTIFICATE

Certified that the mini-project work entitled **“**Selenium Automation testing On Library System**”** is a bonafide workcarried out by

**Anubhuti Rane BCOA42**

**Sampada Gaonkar BCOA43**

The report has been approved as it satisfies the academic requirements in respect of mini-project work prescribed for the course.

……………...…………………………

**Subject In-charge**

**Content**

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **Name** | **Page No** |
| **1** | **Problem Statement** | **5** |
| **2** | **Introduction** | **6** |
| **3** | **Implementation** | **11** |
| **4** | **Conclusion** | **21** |

**List of Figure**

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **Name** | **Page No** |
| **1** | **User-Interface View** | **11** |
| **2** | **Failed Title Test Analysis** | **11** |
| **3** | **Page Link Test Result** | **12** |
| **4** | **User-Interface Of Login view** | **13** |
| **5** | **Assertion in Login Test** | **13** |
| **6** | **User- Interface of Registration Page** | **14** |
| **7** | **Test Result Of Registration Page** | **14** |
| **8** | **User- Interface of Search Page-1** | **15** |
| **9** | **User- Interface of Search Page-2** | **15** |
| **10** | **Test case For Search Page** | **16** |
| **11** | **The Xml file containing all Test Classes** | **16** |
| **12** | **Console Result** | **17** |
| **13** | **TestNG test analysis** | **17** |
| **14** | **Main Page** | **18** |
| **15** | **Total number of classes** | **18** |
| **16** | **Total run Time** | **19** |
| **17** | **Groups Of Test Cases** | **19** |
| **18** | **Chronological View** | **20** |

**I. Problem Statement**

Create a small web-based application by selecting relevant environment/platform and programming languages. Narrate concise Test Plan consisting features to be tested and bug taxonomy. Narrate scripts in order to perform regression tests. Identify the bugs using Selenium WebDriver and IDE generate test reports encompassing exploratory testing.

**II. Introduction**

**1. Manual Testing:**

***Manual Testing*** is a process of finding out the defects or bugs in a software program. In this method the tester plays an important role of end user and verifies that all the features of the application are working correctly. **The tester manually executes test cases without using any automation tools.**

**Life-Cycle of Manual Testing:**



**Disadvantages of Manual Testing :**

* Manual Testing requires more time or more resources, sometimes both.
* Performance testing is impractical in manual testing and no accuracy.
* Batch Testing is not possible, for each and every test execution Human user interaction is mandatory.
* Comparing large amount of data is impractical.

**2. Automation Testing:**

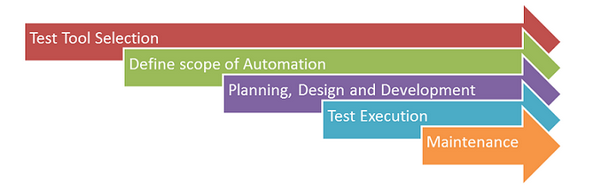
Automation Testing means using an automation tool to execute your test case suite. The automation software can also enter test data into the System Under Test, compare expected and actual results and generate detailed test reports. Using a test automation tool, it's possible to record this test suite and re-play it as required. Once the test suite is automated, no human intervention is required. This improved ROI of Test Automation.

The goal of Automation is to reduce the number of test cases to be run manually and not to eliminate Manual Testing altogether.

**Reasons to use Automation Testing:**

* Manual Testing of all workflows, all fields, all negative scenarios is time and money consuming
* It is difficult to test for multilingual sites manually
* Automation does not require Human intervention. You can run automated test unattended (overnight)
* Automation increases the speed of test execution
* Automation helps increase Test Coverage
* Manual Testing can become boring and hence error-prone.
* 70% faster than the manual testing
* Human Intervention is not required while execution Test Frequently and thoroughly

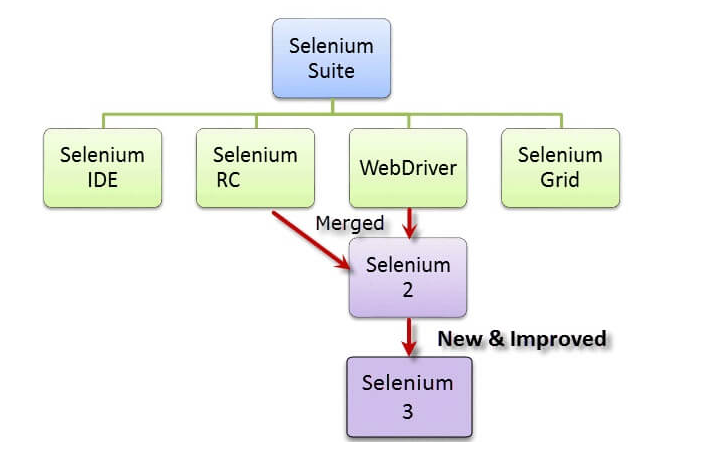
**Automation Testing Process:**



**3. Selenium Test Automation Tool:**

Selenium is a free (open source) automated testing suite for web applications across different browsers and platforms. It is quite similar to HP Quick Test Pro (QTP now UFT) only that Selenium focuses on automating web-based applications

Selenium is not just a single tool but a suite of software's, each catering to different testing needs of an organization. **It has four components.**



* **Selenium Integrated Development Environment (IDE)** is the **simplest framework** in the Selenium suite and is **the easiest one to learn.** It is a **Firefox plugin** that you can install as easily as you can with other plugins. However, because of its simplicity, Selenium IDE should only be used as a **prototyping tool.** If you want to create more advanced test cases, you will need to use either Selenium RC or WebDriver.
* **Selenium RC** was the **flagship testing framework** of the whole Selenium project for a long time. This is the first automated web testing tool that **allowed users to use a programming language they prefer**
* The **Selenium WebDriver** proves itself to be **better than both Selenium IDE and Selenium RC** in many aspects. It implements a more modern and stable approach in automating the browser's actions. WebDriver, unlike Selenium RC, does not rely on JavaScript for Automation**. It controls the browser by directly communicating with it.**
* **Selenium Grid** is a tool **used together with Selenium RC to run parallel tests** across different machines and different browsers all at the same time. Parallel execution means running multiple tests at once.

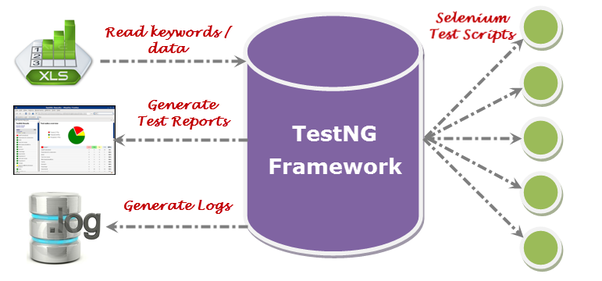
**4. TestNG**

TestNG is an automation testing framework in which NG stands for "Next Generation". TestNG is inspired from [JUnit](https://www.guru99.com/junit-tutorial.html) which uses the annotations (@).

* Using TestNG you can generate a proper report, and you can easily come to know how many test cases are passed, failed and skipped.
* You can execute failed test case separately.
* The TestNG provides an option, i.e., testng-failed.xml file in test-output folder. If you want to run only failed test cases means you run this XML file. It will execute only failed test cases.

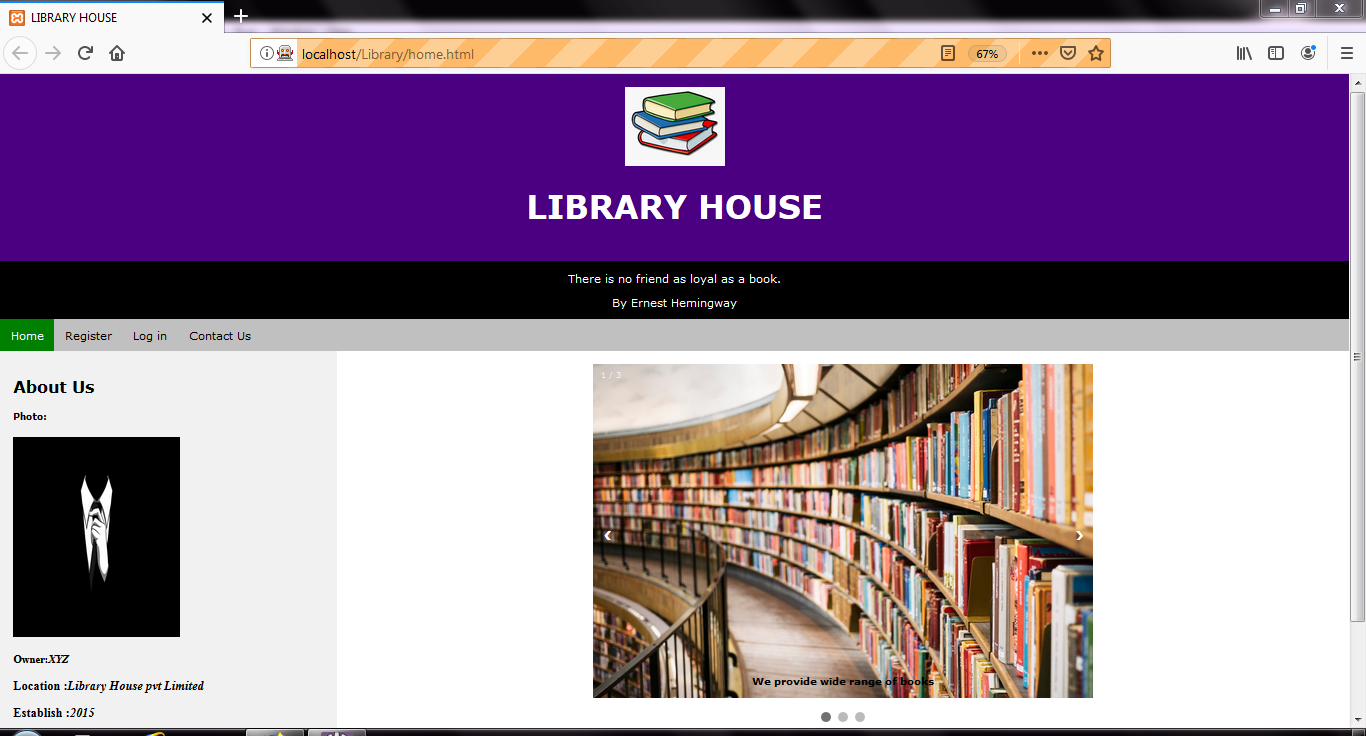
### Why Use TestNG with Selenium?

* Default Selenium tests do not generate a proper format for the test results. Using TestNG we can generate test results.
* Generate the report in a proper format including a number of test cases runs, the number of test cases passed, the number of test cases failed, and the number of test cases skipped.
* Multiple test cases can be grouped more easily by converting them into testng.xml file. In which you can make priorities which test case should be executed first.
* Using testng, you can execute multiple test cases on multiple browsers, i.e., cross browser testing.
* Annotations used in the testing are very easy to understand ex: @BeforeMethod, @AfterMethod, @BeforeTest, @AfterTest.

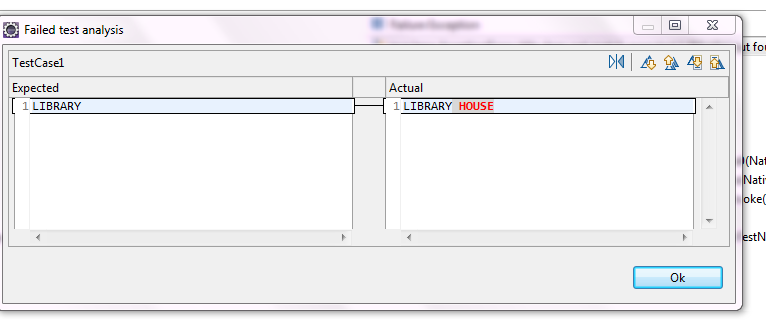


**III. Implementation**

1. **Testing of Home-Page (Title Test):**

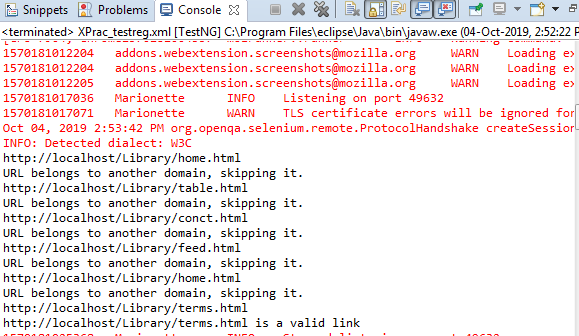
****

**Fiq 1: User-Interface View**

****

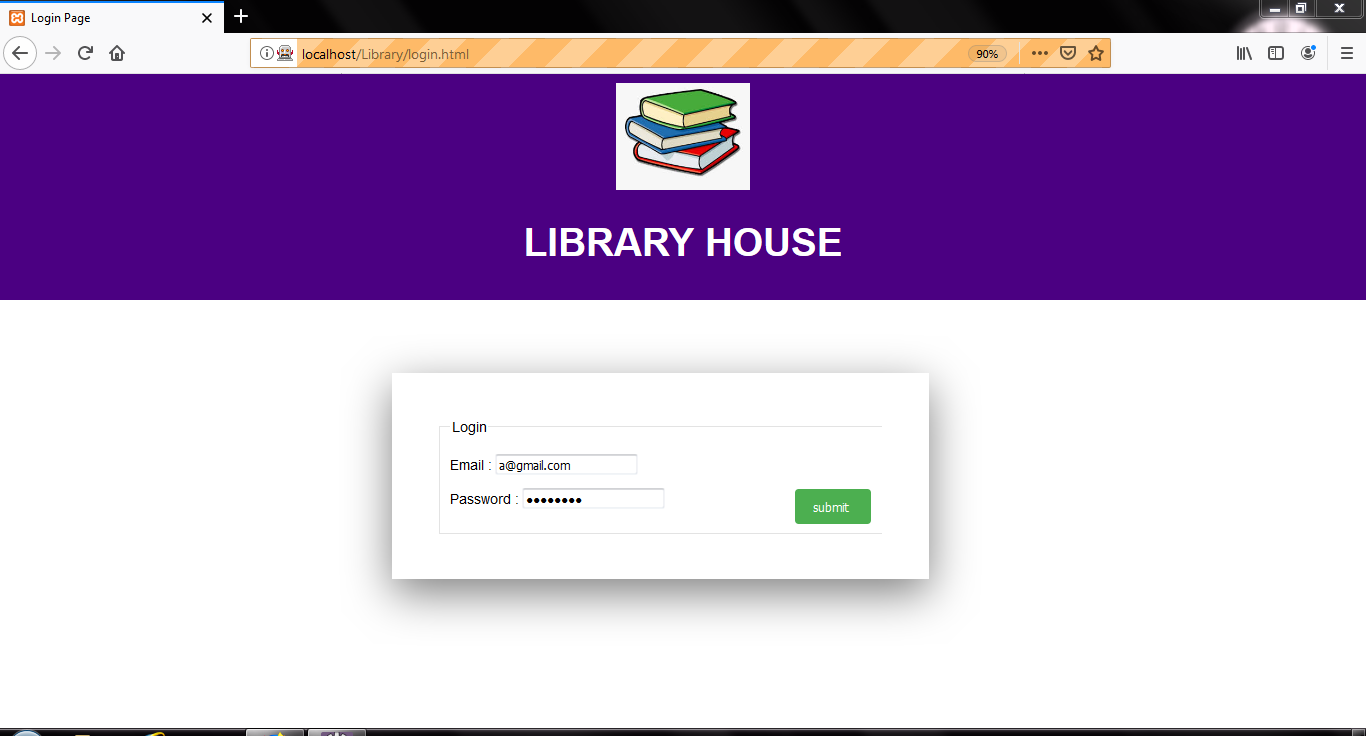
**Fiq 2: Failed Title Test Analysis**

1. **Testing of Page Links**

****

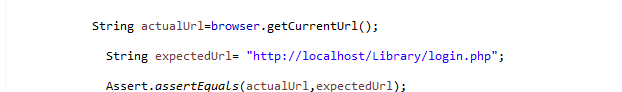
**Fiq 3: Page Link Test Result**

1. **Testing of Login Page**

****

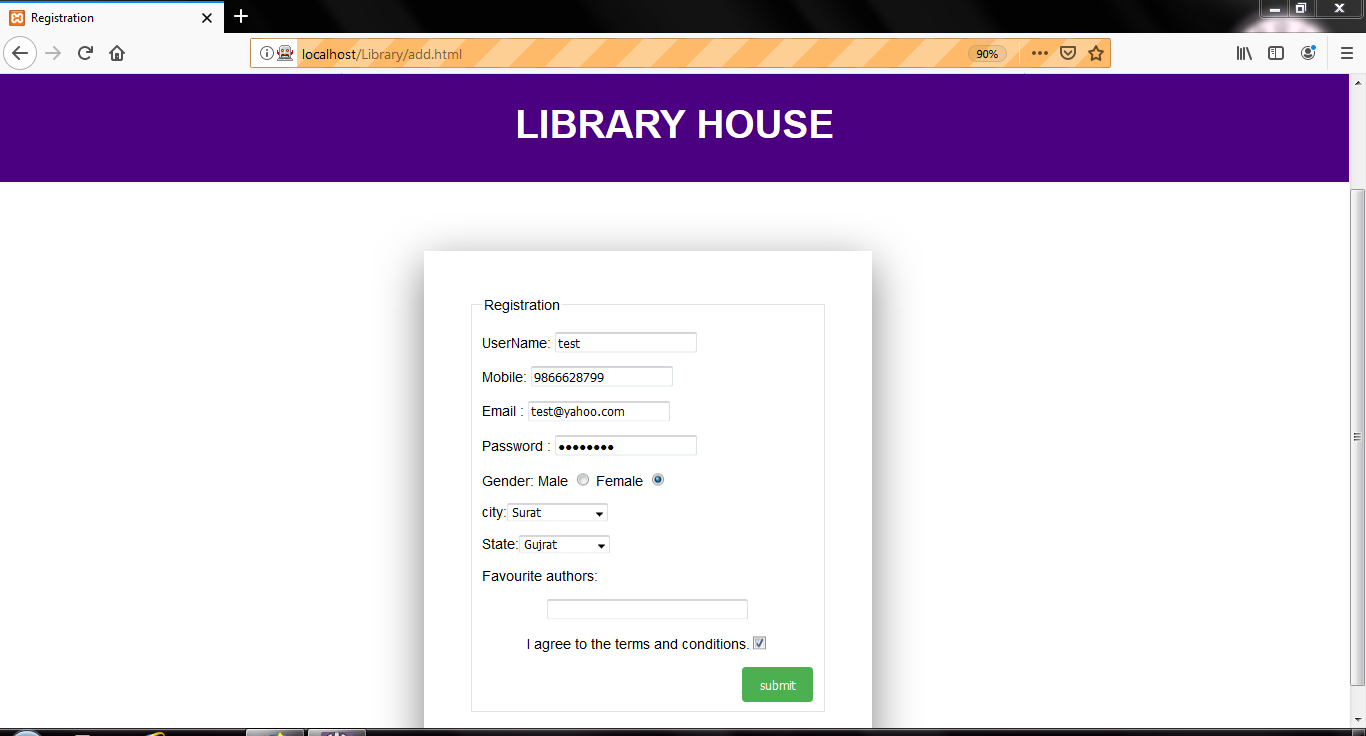
**Fiq 4 : User-Interface Of Login view**

**Assertion test: PASSED**

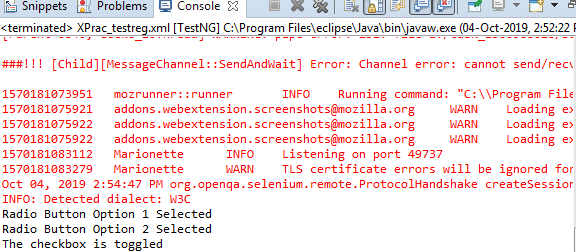
****

**Fiq 5: Assertion in Login Test**

1. **Testing of Registration Page**

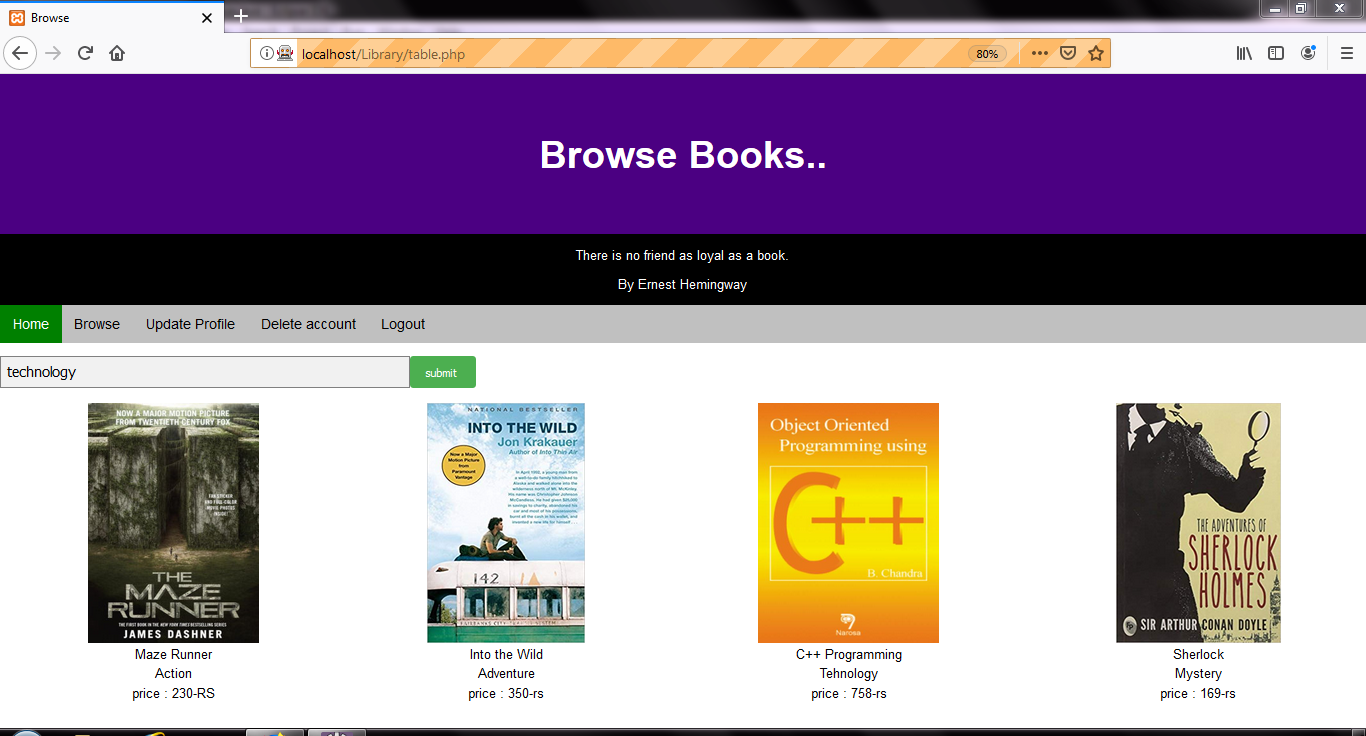


**Fiq 6: User- Interface of Registration Page**

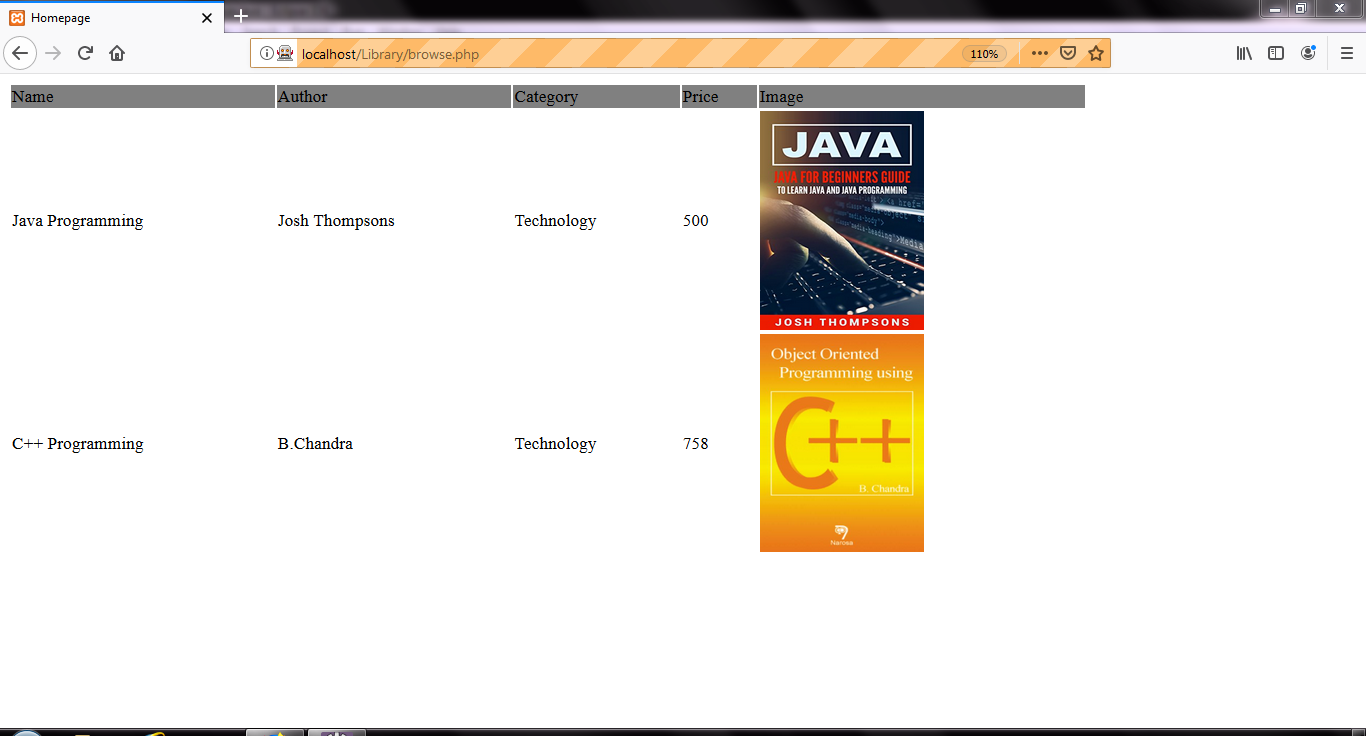
****

**Fiq 7: Test Result Of Registration Page**

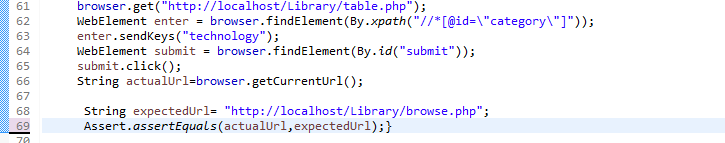
1. **Testing of Search Page**

****

**Fiq 8 : User- Interface of Search Page-1**

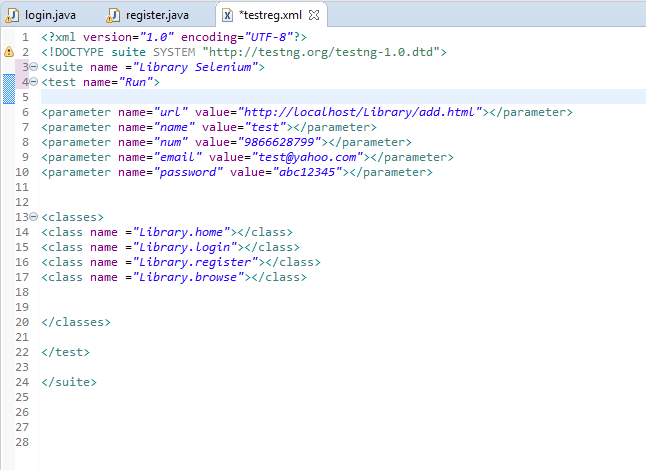
****

**Fiq 9: User- Interface of Search Page -2**

****

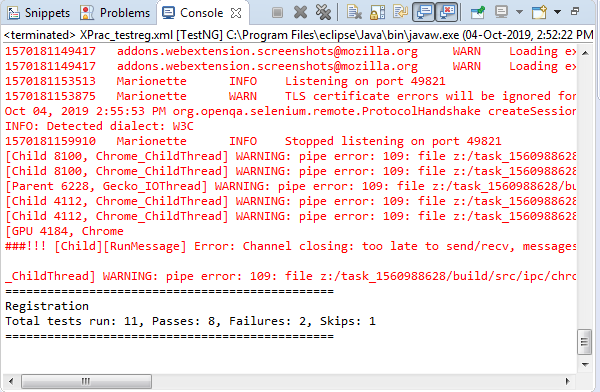
**Fiq 10 : Test case For Search Page**

1. **Xml file**

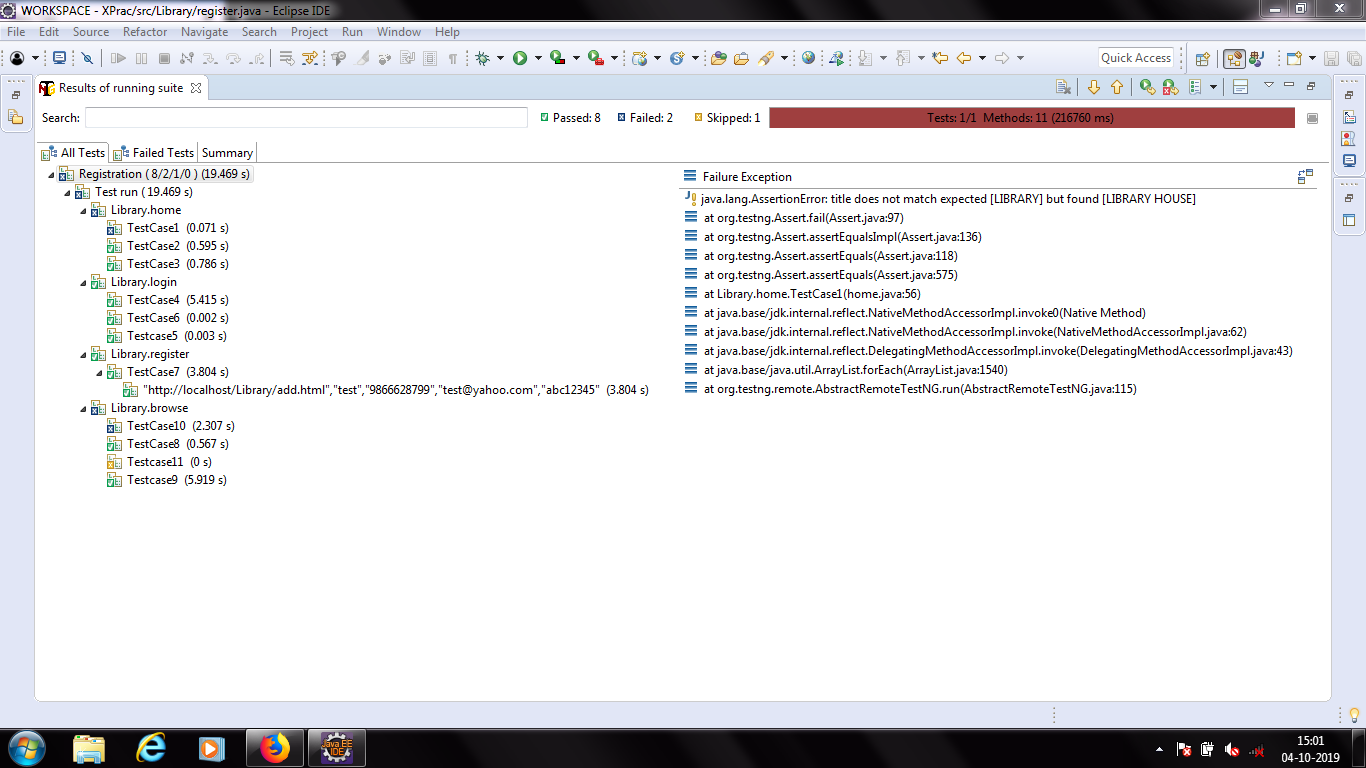
****

**Fiq 11: The Xml file containing all Test Classes**

1. **Test Results of All Test-Cases**

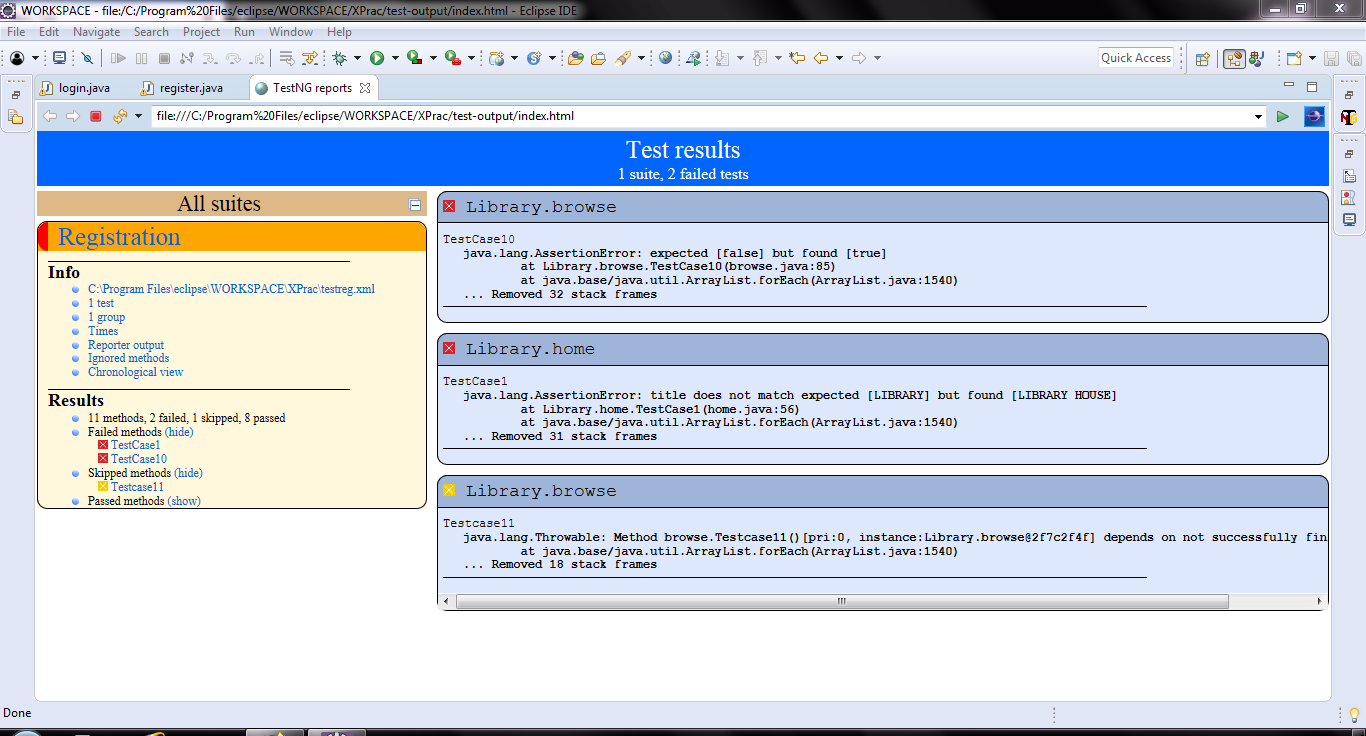
****

**Fiq 12.Console Result**

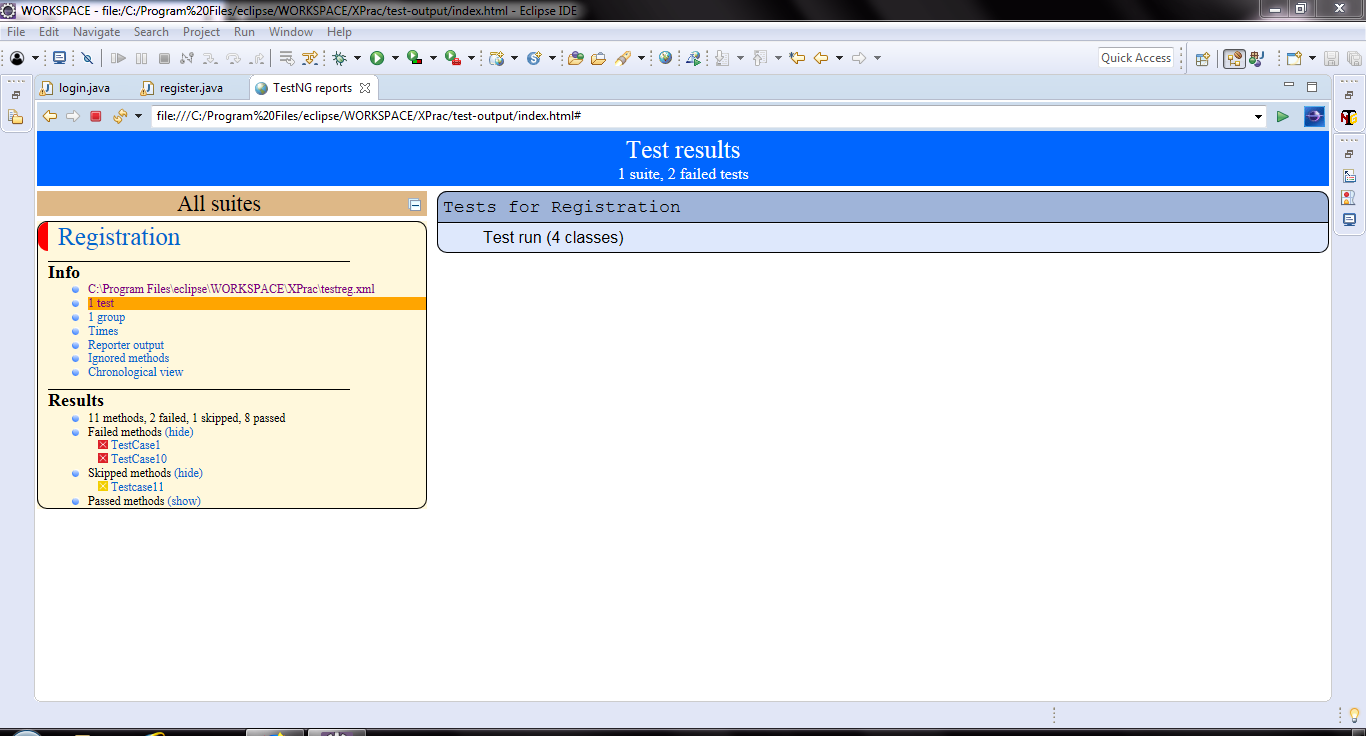
****

**Fiq 13: TestNG test analysis**

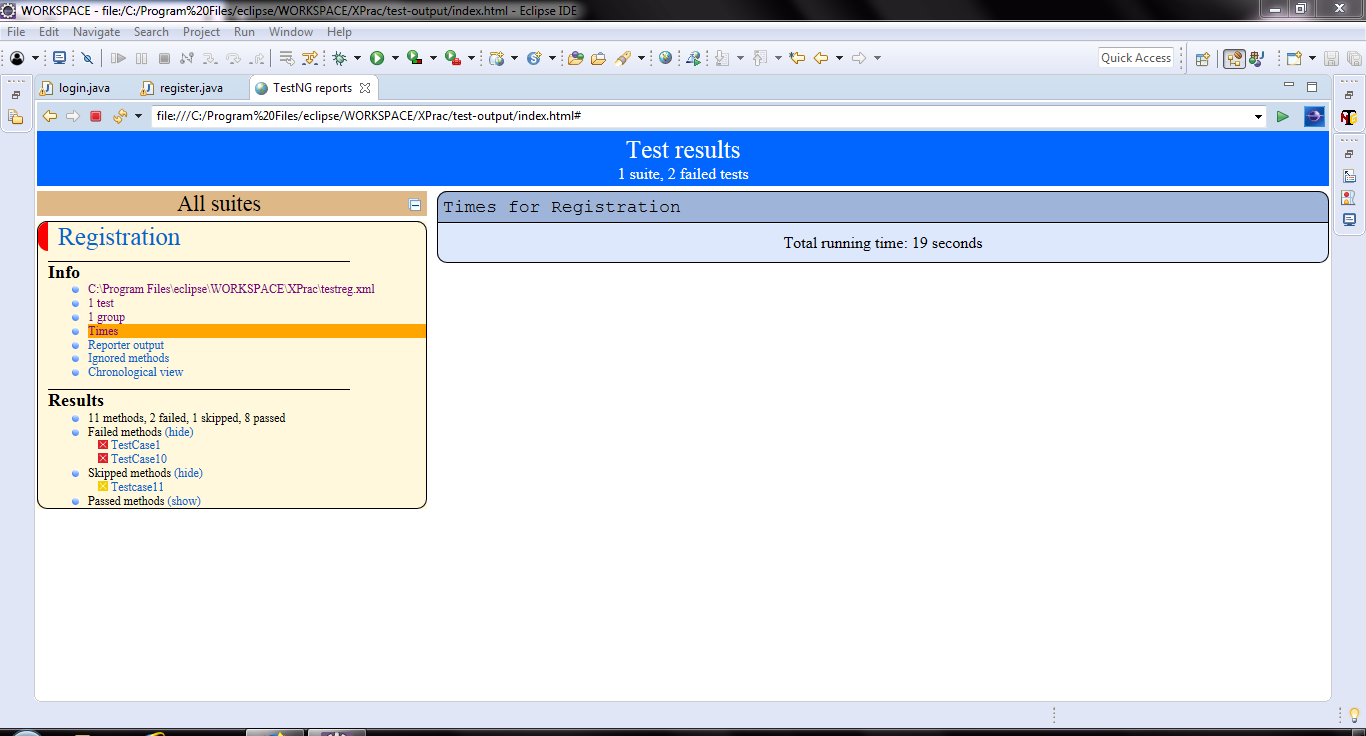
**8. HTML Report**

****

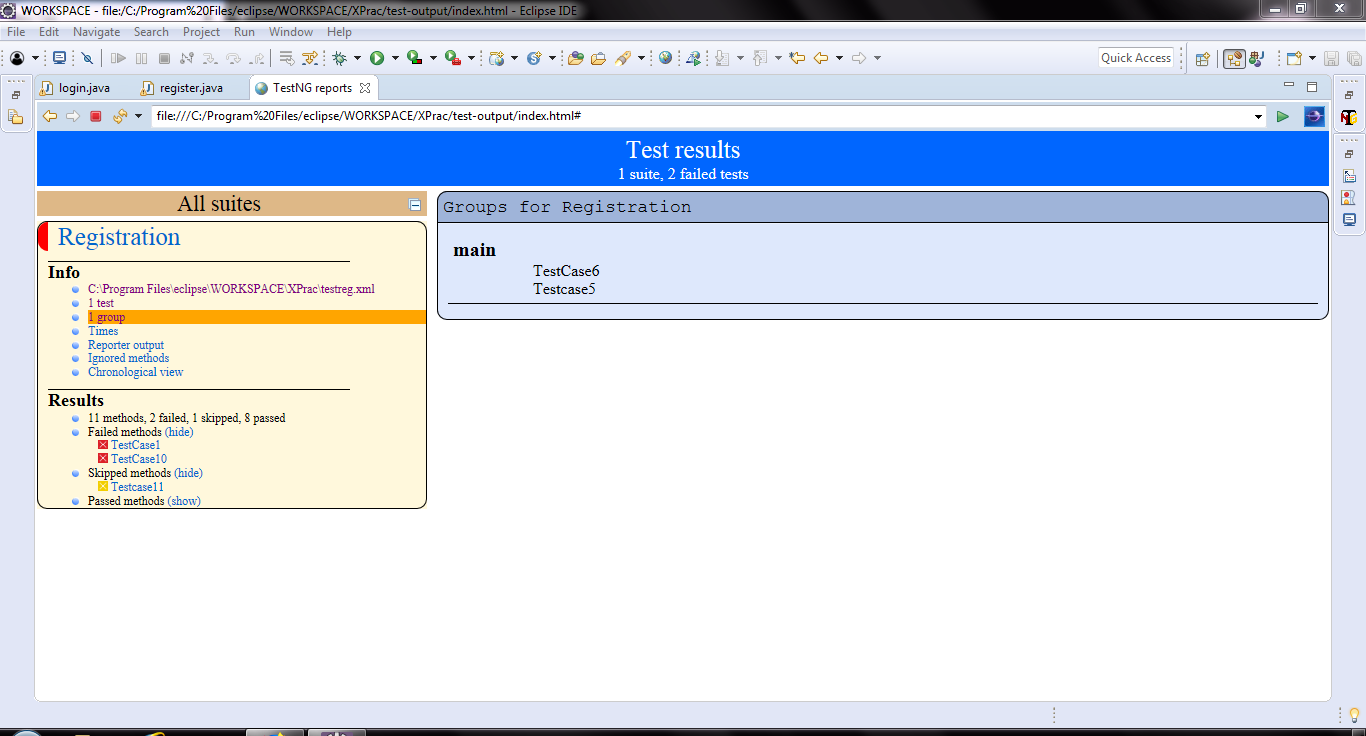
**Fiq 14. Main Page**

****

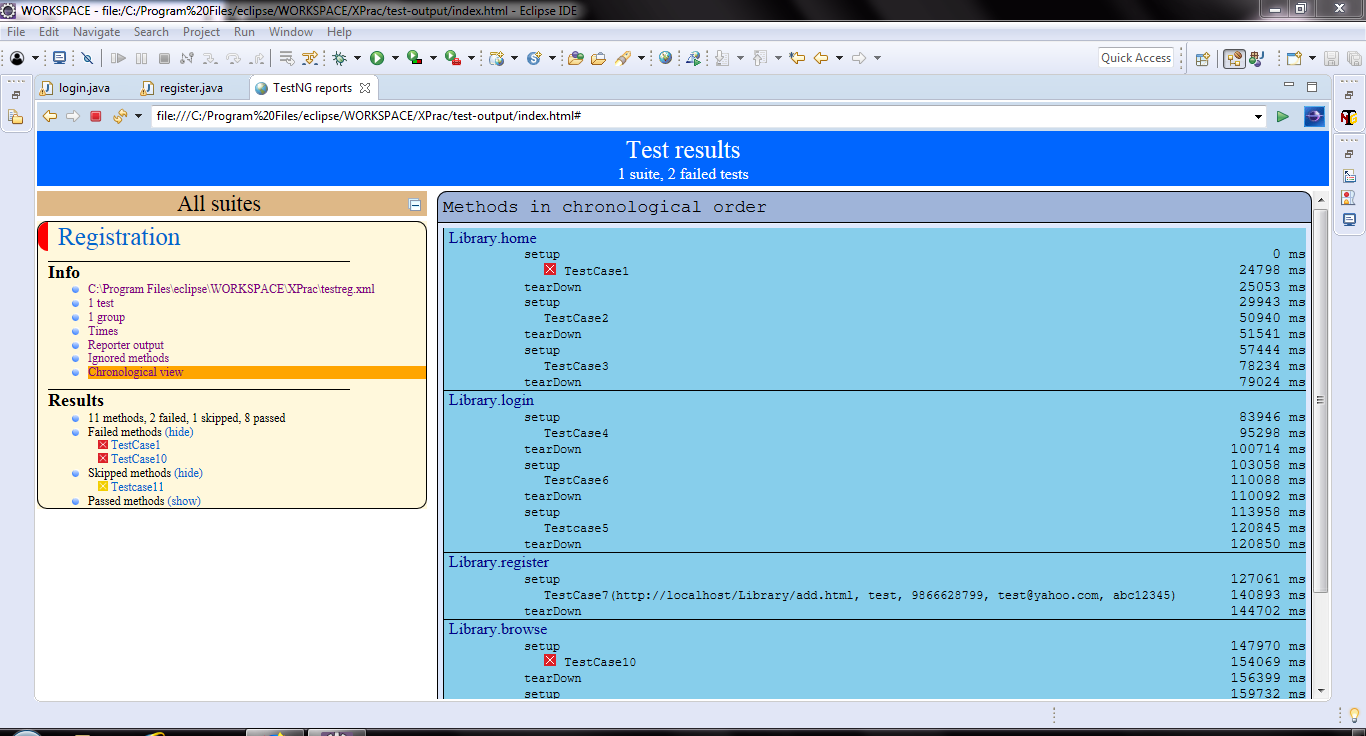
**Fiq 15: Total number of classes**

****

**Fiq 16. Total Run Time**

****

**Fiq 17: Groups Of Test Cases**

****

**Fiq 18: Chronological View**

**IV. Conclusion**

While TestNG with Selenium is useful and effective. It should follow a risk-based approach to balance the testing effort with consequences of software failure. Architectural and design-level risk analysis provide the right context to plan and perform testing. This system tests for all types of Test Cases with various possibilities and shows which part of the design of the webpage does not follow the testing standards