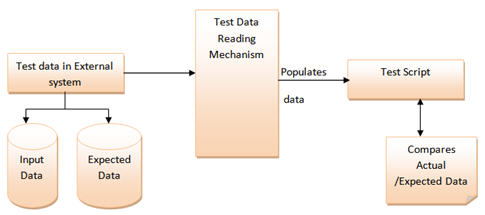
**Data Driven Testing Framework**

While automating or testing any application, at times it may be required to test the same functionality multiple times with the different set of input data. Thus, in such cases, we can’t let the test data embedded in the test script. Hence it is advised to retain test data into some external database outside the test scripts.

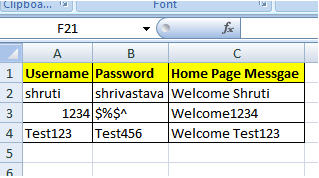
Data Driven Testing Framework helps the user segregate the test script logic and the test data from each other. It lets the user store the test data into an external database. The external databases can be property files, xml files, excel files, text files, CSV files, ODBC repositories etc. The data is conventionally stored in “Key-Value” pairs. Thus, the key can be used to access and populate the data within the test scripts.

***Note***: The test data stored in an external file can belong to the matrix of expected value as well as the matrix of input values.

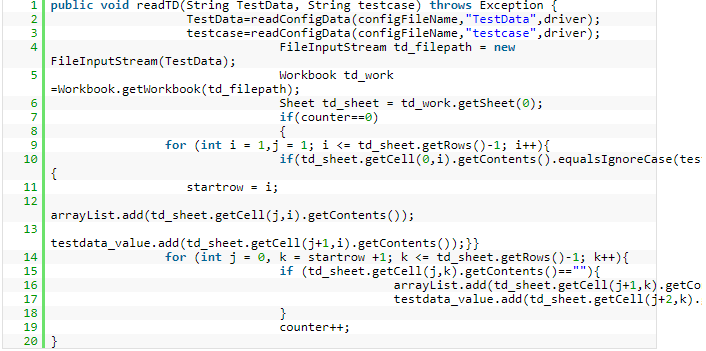


“Gmail – Login” Functionality.

**Step 1:** First and the foremost step are to create an external file that stores the test data (Input data and Expected Data). Let us consider an excel sheet for instance.



**Step 2:** The next step is to populate the test data into Automation test Script. For this purpose, several API’s can be used to read the test data.



The above method helps to read the test data and the below test step helps the user to type in the test data on the GUI.

*element.sendKeys(obj\_value.get(obj\_index));*

**Pros:**

1. The most important feature of this framework is that it considerably reduces the total number of scripts required to cover all the possible combinations of test scenarios. Thus, lesser amount of code is required to test a complete set of scenarios.
2. Any change in the test data matrix would not hamper the test script code.
3. Increases flexibility and maintainability
4. A single test scenario can be executed altering the test data values.

**Cons:**

1. The process is complex and requires an extra effort to come up with the test data sources and reading mechanisms.
2. Requires proficiency in a programming language that is being used to develop test scripts.