**Final Assignment**

**Submitted by:**

**Iqra Shafiq**

**Submitted to:**

**Dr. Usman**

**Riphah International University Lahore**

**Q#01:**

**What is Test Automation Framework?**

Test Automation Framework as a set of guidelines for creating and designing test cases. It is a conceptual part of the automated testing that helps testers to use resources more efficiently. A framework is defined as a set of rules or best practices that can be followed in a systematic way that ensures to deliver the desired results.

Typically, a broader description of test automation framework shows that it consists of a set of processes, tools, and protocols that can be collectively used for automated testing of software applications.

An automation testing framework is a platform developed by integrating various hardware, software resources along with using various tools for automation testing and web service automation framework, based on a qualified set of assumptions. This framework enables efficient design and development of automated test scripts and ensures reliable analysis of issues or bugs for the system or application under test (AUT).

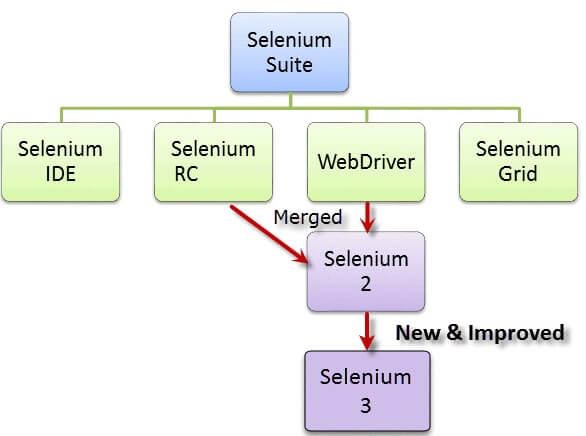
The important functions of software testing automation frameworks are broadly defined as they are effectively used to identify objects and arrange them to be reused in test scripts, perform some action on these identified objects and further also used to evaluate these objects to get the expected results.

# What is Selenium?

**Selenium** is a free (open-source) automated testing framework used to validate web applications across different browsers and platforms. You can use multiple programming languages like Java, C#, Python etc to create Selenium Test Scripts. Testing done using the Selenium testing tool is usually referred to as Selenium Testing.

Selenium Software is not just a single tool but a suite of software, each piece catering to different Selenium QA testing needs of an organization. Here is the list of tools

* Selenium Integrated Development Environment (IDE)
* Selenium Remote Control (RC)
* WebDriver
* Selenium Grid



# How does Selenium work?

Selenium is one of the most popular automation testing tools in the market. As an automation testing professional, one must be aware of the fundamentals of Selenium working.

Let’s start by looking at the automation testing using selenium. You use selenium tool to write test scripts, which are executed by Java. Selenium is not an executable application (like MS word or Excel) and cannot run on its own. It needs a programming environment to work. Some of the programming environments supported by Selenium are:

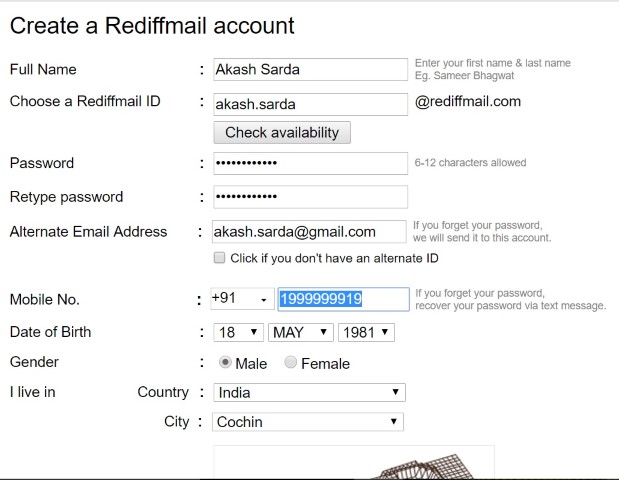
* Java
* Ruby
* PHP
* C# (.NET) & more…

In order to automate the testing, you need to run the test cases using script. What does a test case contain?

a) It contains a condition for a field on the screen

b) It has data to test the condition

In order to automate the testing, we need to write code to put the test data in a particular field in the screen and validate the condition.  If the condition is found to be TRUE, the test case passes else it fails.

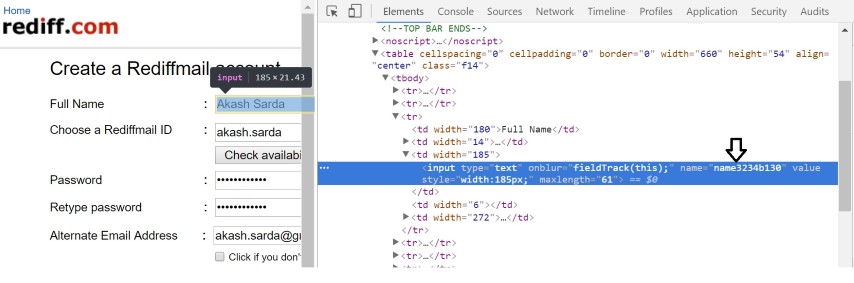


## **How does Selenium script put the data in the fields?**

In order to put test data in the fields on the screen, the script must identify the fields. To identify these fields, selenium scripts use locators. Locators, as the name suggests, helps in locating fields on any web page. Every field on the web page is identified as Web Element. So whether it’s text box or a dropdown list, it’s essentially a web element.



Every web element is identified by a locator id. You can find the locator id using simple browser “Inspect” command, as shown below:



So for writing automation script, you just need to get hold of every field on the screen (under test) and use script to enter required test data.

**Why do we need Selenium?**

As we now are familiar with the makeup of Selenium and the tools at hand, let’s take a look at the main points or benefits of Selenium and why it is a good tool for automation testing:

* Open Source: Selenium is open source, this means that no licensing or cost is required, it is totally free to download and use. This is not the case for many other automation tools out there.
* Mimic User Actions: As stated earlier, Selenium WebDriver is able to mimic user input, in real scenarios, you are able to automate events like key presses, mouse clicks, drag and drop, click and hold, selecting and much more.
* Easy Implementation: Selenium WebDriver is known for being a user-friendly automation tool. Selenium being Open Source means that users are able to develop extensions for their own needs.
* Tool for every scenario: As mentioned earlier, Selenium is a suite of tools, and you will most likely find something that fits your scenario and your way of working.
* Language Support: One big benefit is multilingual support. Selenium supports all major languages like Java, JavaScript, Python, Ruby, C sharp, Perl, .Net and PHP, giving the developer a lot of freedom and flexibility.
* Browser, Operating System & Device support: Selenium supports many different browsers Chrome, Firefox, Opera, Internet Explorer, Edge, and Safari as well as operating systems (Windows, Linux, Mac)
* Framework Support: Selenium also supports a multitude of frameworks like Maven, Junit, TestNG to make it easier to automate testing. CI and CD tools like Jenkins is also supported, for automating the deployment process.
* Reusability: Scripts written for WebDriver is cross-browser compatible. Testers can therefore run multiple testing scenarios with the same base.
* Community Support: The Selenium community is quite active and open. Therefore, there is a lot of information and help available when needed.
* Advanced User Input: With WebDriver it is possible to request clicking of the browser back and front buttons. A practical feature when testing money transfer applications for example. This feature is not found in many tools, especially open source.

**Q#02**

**The most common tools that are used for configuration management are packer and ansible. You need to concisely compare both of them**

**Ansible vs Packer:**

What is Ansible? *Radically simple configuration-management, application deployment, task-execution, and multi-node orchestration engine*. Ansible is an IT automation tool. It can configure systems, deploy software, and orchestrate more advanced IT tasks such as continuous deployments or zero downtime rolling updates. Ansible’s goals are foremost those of simplicity and maximum ease of use.

What is Packer? *Create identical machine images for multiple platforms from a single source configuration*. Packer automates the creation of any type of machine image. It embraces modern configuration management by encouraging you to use automated scripts to install and configure the software within your Packer-made images.

Ansible belongs to "Server Configuration and Automation" category of the tech stack, while Packer can be primarily classified under "Infrastructure Build Tools".

Some of the features offered by Ansible are:

* Ansible's natural automation language allows sysadmins, developers, and IT managers to complete automation projects in hours, not weeks.
* Ansible uses SSH by default instead of requiring agents everywhere. Avoid extra open ports, improve security, eliminate "managing the management", and reclaim CPU cycles.
* Ansible automates app deployment, configuration management, workflow orchestration, and even cloud provisioning all from one system.

On the other hand, Packer provides the following key features:

* Super fast infrastructure deployment. Packer images allow you to launch completely provisioned and configured machines in seconds, rather than several minutes or hours.
* Multi-provider portability. Because Packer creates identical images for multiple platforms, you can run production in AWS, staging/QA in a private cloud like OpenStack, and development in desktop virtualization solutions such as VMware or VirtualBox.
* Improved stability. Packer installs and configures all the software for a machine at the time the image is built. If there are bugs in these scripts, they'll be caught early, rather than several minutes after a machine is launched.

[Test Automation Frameworks - Why, Types, Benefits, Approach (testingxperts.com)](https://www.testingxperts.com/blog/test-automation-frameworks#What%20is%20Test%20Automation%20Framework?)

[What is Selenium? Introduction to Selenium Automation Testing (guru99.com)](https://www.guru99.com/introduction-to-selenium.html)

[How does Selenium work | Selenium Blog | Techcanvass](https://techcanvass.com/automation-blog/how-does-selenium-work/)

[Ansible vs Packer | What are the differences? (stackshare.io)](https://stackshare.io/stackups/ansible-vs-packer)