**What is a Framework*?***

To design a system or a platform using that platform we solve our business problem/achieve our goals in a systematic way.

In the context of automation our business is to create as many test cases as we can automate, so we need a framework for that in which we can convert our manual testcases in to automated testcases

**Tell me some popular Test Automation Frameworks?**

* Linear Scripting Framework
* Modular Testing Framework
* Data Driven Testing Framework
* Keyword Driven Testing Frameworks
* Hybrid Testing Framework
* Behavior Driven Development Framework

**Linear Scripting Framework:**

Linear Scripting Framework is a basic level test automation framework which is in the form of ‘Record and Playback’ in a linear fashion. This framework is also known as ‘Record and Playback’ framework.

### ****Modular Testing Framework:****

In the modular testing framework, testers create test scripts on module wise by breaking down the complete application under test into smaller, independent tests. In simple words, testers divide the application into multiple modules and create test scripts individually.

### ****Data-driven Framework:****

Data-driven test automation framework is focused on separating the test scripts logic and the test data from each other. Allows us to create test automation scripts by passing different sets of test data. The test data set is kept in the external files or resources such as MS Excel Sheets, MS Access Tables, SQL Database, XML files etc., The test scripts connect to the external resources to get the test data. By using this framework we could easily make the test scripts work properly for different sets of test data. This framework significantly reduces the number of test scripts compared to module-based framework.

This framework gives more test coverage with reusable tests and flexibility in the execution of tests only when required and by changing only the input test data and reliable in terms of no impact on tests by changing the test data but it has its own drawbacks such as testers who work on this framework needs to have hands-on programming knowledge to develop test scripts

### ****Keyword Driven Testing Framework:****

It is also known as table-driven testing or action word based testing. In Keyword-driven testing, we use a table format to define keywords or action words for each function or method that we would execute.

### ****Hybrid Driven Testing Framework:****

Hybrid Test automation framework is the combination of two or more frameworks mentioned above. It attempts to leverage the strengths and benefits of other frameworks for the particular test environment it manages. Most of the teams are building this hybrid driven framework in the current market.

### ****Behavior Driven Development Testing Framework:****

The purpose of this Behavior Driven Development framework is to create a platform which allows everyone (such as Business Analysts, Developers, Testers etc,) to participate actively. It requires increased collaboration between Development and Test Teams.  It doesn’t require the users to be acquainted with a programming language. We use non-technical, natural language to create test specifications. Some of the tools available in the market for Behavior Driven Development is [JBehave](http://jbehave.org/), [Cucumber](https://cucumber.io/), etc.,

**Why Framework?**

In a test automation project, we do perform different tasks by using different types of files. To organize and manage all the files and to finish all the tasks in a systematic approach we use a framework.

**What are the advantages of using Test Automation Framework?**

1. Saves time and money. Automation testing is faster in execution
2. Reusability of code. Create one time and execute multiple times with less or no maintenance
3. Easy reporting. It generates automatic reports after test execution
4. Easy for compatibility testing. It enables parallel execution in combination of different OS and browser environments
5. Low cost maintenance. It is cheaper compared to manual testing in a long run
6. Automated testing is more reliable
7. Automated testing is more powerful and versatile
8. It is mostly used for regression testing. Supports execution of repeated test cases
9. Minimal manual intervention. Test scripts can be run unattended
10. Maximum coverage. It helps to increase the test coverage

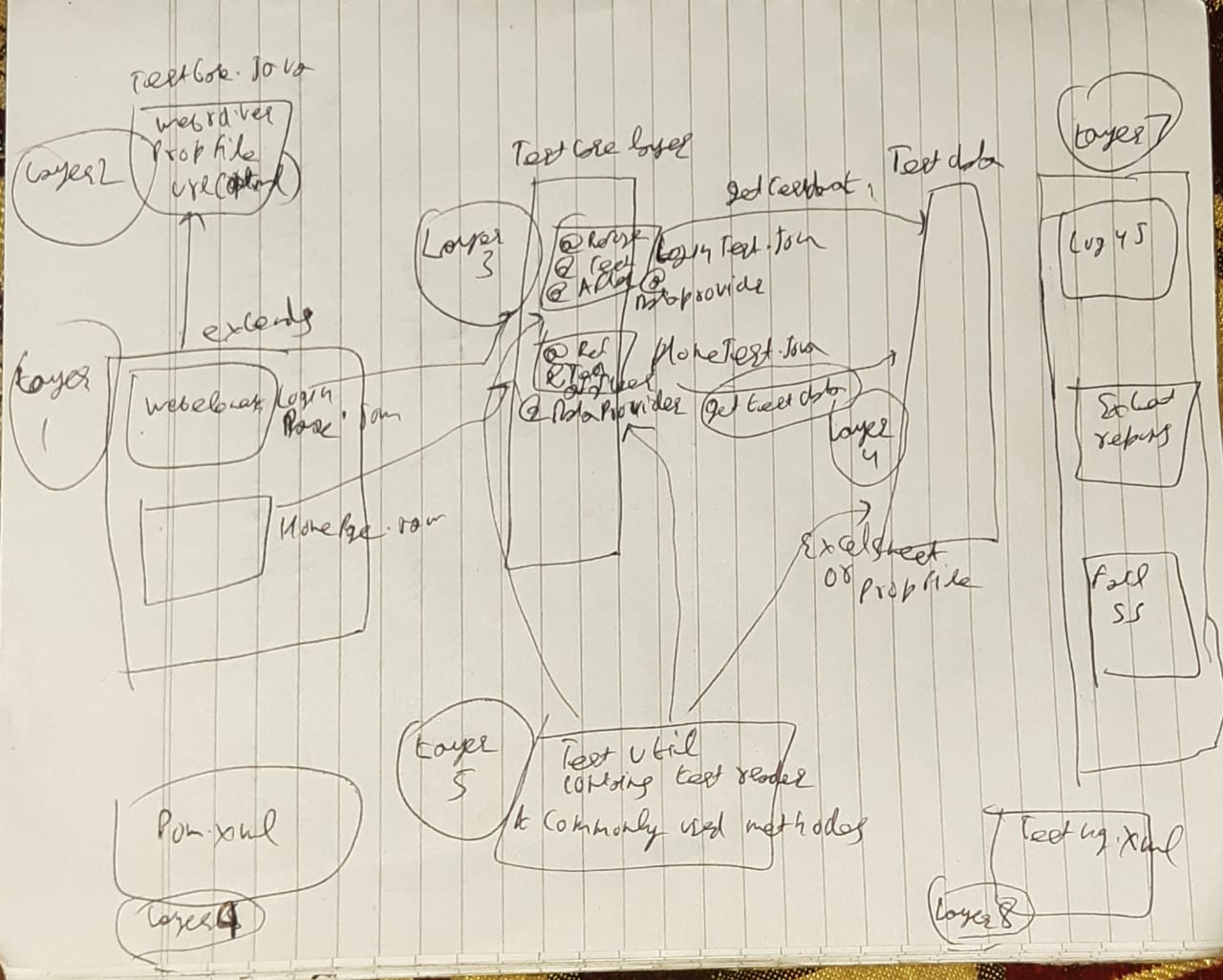
**Which Test Automation Framework you are using and why?**

Some of the Test Automation Frameworks are:

* Data Driven Testing Framework
* Hybrid Testing Framework

**Can you explain the Framework which you have used in your Selenium Project?**

I worked on POM which is made using data driven approch, which is on JAVA language.



**What type of test cases do you pick up to automate?**

I focus on the test cases which should be executed in a repetitive manner such as regression test cases, smoke and sanity test cases

**What type of test cases you won’t pick up to automate?**

Before picking up the test cases to automate, I do check whether the application is stable or not. So based on this, I don’t pickup test cases when the AUT(Applcation under test) changes frequently and the test cases which I run rarely and run only one time. When I do usability and exploratory testing.

**How many test cases you have automated per day?**

It depends on Test case scenario complexity and length. I did automate 2-5 test scenarios per day when the complexity is limited. Sometimes just 1 or fewer test scenarios in a day when the complexity is high.

**Extent report in selenium ?**

1. **Add maven dependency**
2. **Create a package ex:** com.qa.extentReportListner and add java class in it **ex:** ExtentReportListnerNG(IT S A TEMPLATE)
3. ExtentReportListnerNG class **implements** IReporter
4. Add listeners in testing.xml file

<listeners>

<listener

class-name=*"com.qa.extentReportListner.ExtentReportListnerNG"* />

</listeners>

**what is page object model?**

**Page Object model** is design **pattern** in Selenium, where web **pages** are represented as classes, and the various elements on the **page** are defined as variables on the class.

**Pagefactory?**

The ***PageFactory*** Class in Selenium is an extension to the Page Object design pattern.

It is used to **initialize elements** of a Page class **without** having to use **‘FindElement’ or ‘FindElements’**.

It is coming from **org.openqa.selenium.support** package

@FindBy(xpath="(//input[@name='email'])[1]")

WebElement UName;

@FindBy(how=How.***ID***,using="userpass") //standard way to define @findby is using How

WebElement Pswd;

Loginpage loginpage = PageFactory.***initElements***(*driver*, Loginpage.**class**); //return page object of Loginpage class

**CacheLookup:**

Cachelookup increase the performance of a testcase, because it will not check a element on the webpage again, it will take from the cache memory and start processing

We use cachelookup only with those webelements which can not be change

@FindBy(xpath = "//button[@type='submit']")

@CacheLookup

WebElement Login;

# What is the difference between Page object model and Page factory?

# Where you have applied OOPS in Automation Framework

<https://www.youtube.com/watch?v=g3r5KK2Acx8>