**Page Object Model (POM)**

## **What is Page Object Model?**

****Page Object Model (POM)**** is a design pattern, popularly used in test automation that creates Object Repository for web UI elements. The advantage of the model is that it reduces code duplication and improves test maintenance.

Under this model, for each web page in the application, there should be a corresponding Page Class. This Page class will identify the WebElements of that web page and also contains Page methods which perform operations on those WebElements. Name of these methods should be given as per the task they are performing, i.e., if a loader is waiting for the payment gateway to appear, POM method name can be wait For Payment ScreenDisplay ().

**Advantages of Page Object Model :**

* ****Helps with easy maintenance****: POM is useful when there is a change in a UI element or there is a change in an action. An example would be: a drop-down menu is changed to a radio button. In this case, POM helps to identify the page or screen to be modified. As every screen will have different java files, this identification is necessary to make the required changes in the right files. This makes test cases easy to maintain and reduces errors.
* ****Helps with reusing code****: As already discussed, all screens are independent. By using POM, one can use the test code for one screen, and reuse it in another test case. There is no need to rewrite code, thus saving time and effort.
* ****Readability and Reliability of scripts****: When all screens have independent java files, one can easily identify actions that will be performed on a particular screen by navigating through the java file. If a change must be made to a certain section of code, it can be efficiently done without affecting other files.

## **Why Page Object Model?**

Starting an UI Automation in Selenium Web Driver is NOT a tough task. You just need to find elements, perform operations on it.

## **How to implement POM?**

Simple POM:

It’s the basic structure of Page object model framework where all Web Elements of the ****AUT**** and the method that operate on these Web Elements are maintained inside a class file.A task like verification should be separate as part of Test methods.

**POM Implementation:**

1. With Page Factory - Uses By().

No imports needed.

No cache storage.

1. Without Page Factory - Uses @FindBy().

Imports packages: package factory.

Cache lookup is faster.