**FRAMEWORK**

Currently i'm working on banking domain

in my project as per our company standards we are following Hybrid driven framework

1. Data Driven Framework

As per my project i have to test my application with multiple set of data if i do it manually means it is very tedious job So thats why i'm maintaining the data in different external resources like property files, excel files ,json files ,csv files and databases as per rules of automation while writting scripts we should not hard code we should get it from external resources

in Property file i'm maintaing the Common data which is common to each and every testscripts like username password url

And in excel sheets i'm maintaining the Test data for particular test script execution maintaining multiple sheets based on our project needs So we call it as data driven framework

1. Modular Driven Framework

Our application is very huge with multiple modules like customer ,staff ,manager ,savings ,loans difficulty to maintain so we are creating packages based on no.of modules present in our project So that i can easily maintain So we can call it as modular driven framework

1. Method Driven Framework

While automating test scripts we are making use of reusable methods (functions) .We are maintaining generic classes for reuse of Methods .We already know the code no need to rewrite it again our job is to reduce no of lines of code in script we have several advantages like code reusability code optimization maintainanace will be easy this approach we call it as method driven framework

1. Generic Utility

For writting reusable methods we are maintaining seperate package like generic utility seperate classes for writting reusable functions for maintaing java , excel ,property , json , database connection and webdriver methods in generic utility we are maintaining Base Class writting the testNG annotations Annotations are imp block in java used to provide special instructions to the

jvm during run time like @BeforeSuite we are writting here database connectinon configuration methods @BeforeTest and @BeforeClass writting the configuration methods before execution of test classes in real time we are writting the codes for launching the browser @BeforeMethod will execute before the execution of each and every test script in realtime it has Configuration methods for opening the browser preconditions for opening the browser

@AfterMethod configuration methods once after executing each and every test methods like logout from the application @AfterClass configuration methods for closing the browser after execution of each test classes @AfterSuite configuration methods to close the database connection . in BaseClass we are writting the configuration methods which are inherited by each and every test scripts here we are achieving Hierarichal inheritance

1. POM Classes/Pages

POM one of java design pattern preferred by google to store all web elements in seperate class based on no of webpages present in our project So no need to write xpath again and again we can write it once use it for several times later Here to identify the webelements we are making use of selenium annotations like @FindBy ,@FindBys and @FindAll annotations because we should not hard code elements we should get it from object repository with the help of these we can avoid stale element referance exception and achieve auto healing process so that script becomes more roboust and execution will take less time Here we are declaring web elements as private for security issues So here we are achieving abstraction concept making use of constructor to initialize the objects making use of helper methods like getters and setters methods to use the objects writting the business logics. As we are following agile process customer keeps on changing requirements and user interface may change thats why we are maintaining POM classes in our project

1. POM.xml

POM.xml is the one of project object module Here we are maintaining dependency jar files to supports for execution of test scripts When we use maven as build processing tool it will supports third party plug in tools simplifies the build processes and automating the builds execution of test scripts

1. Listener

@Listener is the one of imp feature in selenium additional annotation it will monitor test script execution during runtime and performs special operation whenever test case getting failed

as i'm implementing itestListener interface to take the screenshot of failed testscripts So we can easily analyze the failed test scripts at which point they got failed analsing the reports if the test scripts are got failed means we can retry it by implementing retry analyser interface to know about the exact reason for test failure whether it is for synchronisation issues ,browser or code issues

1. Emailable report.html

Once testscripts got executed it will automatically generate report in test output folder emailable html report We get to know about how many test scripts got executed ,passed failed and skipped test cases reason for exception we get to know about this

1. testNG

in my project I’m using testNG as unit testing tool it provide the more advantage and more features we can perform batch, group ,parallel and group execution and i can execute only failed testscripts .

Assertion one of important feature in selenium used validate actual and expected results of testscripts. Hard assert and soft assert

Annotations used to provide custom instructions to JVM during runtime