Hi this is Md. I have been working around 5 years of experience into IT and it’s totally into QA. I have experience Manual testing and Automation testing. But last 4 years I am majorly involved into automation activity and sometimes, they need to help Manual testing while I am doing manual testing in all phases of Software Development Life Cycle (SDLC) and Software Testing Life Cycle (STLC).

Throughout my career I worked as a manual and automation tester across various domains such as Bank and Insurance mostly for web-based applications where I performed front end and back end testing by developing test scenarios, test cases, test data, test strategy, test plans, test reports (based on acceptance criteria, and Wireframe), and also created automation test scripts using Selenium WebDriver API with Java.

I performed different types of testing like Smoke Testing, Sanity Testing, Regression Testing, Integration Testing, and Database Testing etc. Doing that I mostly worked in Agile methodology in both Scrum and Kanban projects.

As an automation engineer, I have experience in writing and analyzing different kinds of documentations and attending different Agile meetings whether they are an onshore or offshore team.

I am expert in UI Testing, I have hands on experience for back end testing as well. To validate Database, I used SQL queries with multiple Databases such as SQL, MySQL and Oracle. Other

In my career, I had a chance to work with different testing frameworks like Data Driven, Keyword Driven, Hybrid, BDD Cucumber etc. where I mostly used Maven as a project building tool and I used Junit, and TestNG as a testing framework. I used Jenkins as a CI/CD tool and log4j. To share our project within team members I got chance to work with different version control tools like Git and GitHub.

Even though I have good knowledge in API testing manually like Rest and Soap using Postman and SoapUI tools. Besides that, I execute Rest API using automation tools like Rest Assured framework.

In my last project, I have a good understanding with BDD (Behavior Driven Development) Framework where I write Cucumber Feature File using Gherkin Syntax like (Feature, Background, Scenario, Scenario Outline, Given, And, When, Then, and Example) and I implemented them using Page Object Model concept through Step Definition class and executed tests using Runner class configuring with Cucumber Options such as feature, glue, format, Dry Run etc.

My current framework is BDD framework with Cucumber approach. Basically, there are three stages:

**1) Designing the framework,**

**2) Implementing the framework,**

**3) Execution of the framework.**

We implemented maven project. It uses pom.xml file to maintain our dependencies such as selenium-standalone-server, selenium-java, TestNG, Apache poi etc. We have src/main/java which library package that contains configuration class which works as static class and have some final static variable and methods that we can utilize in the project, we have generic function class to store common functionalities that we use in our script, ExcelUtility class to write/read excel files, annotation transformer class to customize the annotation in runtime. Another packages is src/test/java where we write our test script using page object model pattern. Every web page has different class to store the element. It has some other sub-packages such as PageModel package contains location of the element, Page Object package we extends pageModel and contains actions that we want to perform on different elements. PageTest package contains tests for different pages and I define how to execute these tests. For configuring and executing tests we use testng.xml file.

The project has screenshot folder to store screenshots of failed test. Data folder to store our test data files. browserDriver folder to store drivers. configuration folder that contains properties file to store common data such as url, username, password etc. We have test-output folder where testNG reports are generated.

We have GitHub version control system to store our code. We have master branch and separated branches for each member(Release branch). When our test script is done, I run it locally to make sure it’s succeeded. We utilize Jenkins to run our test. We have devops team to set up pipeline and run our test in Jenkins.

I built my framework from scratch

I used java programing language with selenium web Driver for automation

Maven – Build tool

TestNG – java unit testing framework

Cucumber – BDD automation approach

GitHub – version control

Jenkins -CI

My current framework has

Generic library (all common code with methods)

Page Factory (all Xpaths inside and its act like a repository)

Utility package (all utility methods inside like highlight, screenshot, config file)

DB utility package (database connection with other methods to get value from database)

Excel utility package (to handle excel with apache poi)

For Cucumber:

Feature file (all test steps)

Step def (all steps code)

Runner file( it will run the feature file with step def and also I can use different cucumber options like plugin, dry run, monochrome, strict and hooks)

I execute my code when application in ready and stable

I push my code into GitHub and connect with jenkins for CI

I checked my build from maven or jenkins

Also generate report by the help of testNG and shire reports with my team members.

In short, that is all about myself .

If you would like to know anything specifically, I’ll be more than happy to elaborate.

\*\*\*\* What are different options available in cucumber and what are the purposes of each option?

1. dry Run=true; dry Run checks that all the steps of Feature file have in the Step Definition class.

2. Features = Features set the path of the Feature file. Where is the Feature file available?

3. glue= glue set the path of the step Definition class. Where is exactly step Definition class available?

4. tags= tags are used for running a specific scenario of the Feature file.

5. monochrome=true; monochrome displays the console output in proper readable format.

6. Strict= true; Strict will fail execution if there are any undefined steps in step Definition class.

7 plugin= plugin generates different types of reports. It can generate HTML report, XML report and Jason report

 

What is Page Object Model in Selenium?

Page Object Model, also known as POM, is a design pattern in Selenium that creates an object repository for storing all web elements. It is useful in reducing code duplication and improves test case maintenance.

In Page Object Model, consider each web page of an application as a class file. Each class file will contain only corresponding web page elements. Using these elements, testers can perform operations on the website under test.

### What is Page Factory in Selenium?

Page Factory is a class provided by [Selenium WebDriver](https://www.browserstack.com/guide/selenium-webdriver-tutorial) to support Page Object Design patterns. In Page Factory, testers use **@FindBy** annotation. The **initElements** method is used to initialize web elements.

* **@FindBy**: An annotation used in Page Factory to locate and declare web elements using different locators. Below is an example of declaring an element using **@FindBy**

@FindBy(id="elementId") WebElement element;

Difference Between Page Object Model and Page Factory in Selenium

|  |  |
| --- | --- |
| **Page Object Model** | **Page Factory** |
| Finding web elements using **By** | Finding web elements using **@FindBy** |
| POM does not provide lazy initialization | Page Factory does provide lazy initialization |
| Page Object Model is a design pattern | PageFactory is a class that provides the implementation of the Page Object Model design pattern |
| In POM, one needs to initialize every page object individually | In PageFactory, all page objects are initialized by using the **initElements()** method |