What is API automation framework?
RestAssured is a very powerful automation framework written in Java. It is designed to automate HTTP Requests – so REST APIs.

What is a page object model in selenium?
Page Object model is an **object** 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.
In this case we will use **Page** Factory to initialize web elements that are defined in **Page Objects**.

What is Page Factory in Selenium?
Page Factory is a class provided by ***Selenium WebDriver*** to implement the ***Page Object Model***. The *Page Object Repository* is separated from the *Test Methods* using the *Page Factory* concept.
Page Factory' is one way of implementing the 'Page Object Model'.

Subsequently, there are two simple steps through which we need to define and use ***Page Factory in Selenium projects:***

1. ***Using the @FindBy annotation***– *Unlike the regular approach of initializing web page elements using* ***FindElement*** *or* ***FindElements*****,** *the Page Factory uses the* ***@FindBy*** *annotation. The annotations used in Page Factory are descriptive. Moreover, they help improve code readability, which we will discuss in the next section. It provides the following syntax to locate the web elements:*

```
@FindBy(id="userName")
WebElement username;
```

2. ***Initializing the elements using initElements()***– *This is a static method used to initialize the web elements that we locate using the* ***@FindBy*** *or other annotation(s), thereby instantiating the page class.*

```
PageFactory.initElements(WebDriver driver,
java.lang.Class.pageObjectClass);
```

findElement()*: This method uniquely finds a web element on the web page.*
findElements()*: This method finds a list of web elements on the web page.*

Apache POI:-
We can do it as shown below using the ***HSSFWorkbook Class.*** If you are using MS-Office versions 97–2003 or ***XSSFWorkbook Class*** if MS-Office versions 2007 or later. In the below line, we use ***HSSFWorkbook*** as an Excel version is 97-2003.

```
File file =    new File("E:\\TestData\\TestData.xls");
FileInputStream inputStream = new FileInputStream(file);
HSSFWorkbook wb=new HSSFWorkbook(inputStream);
HSSFSheet sheet=wb.getSheet("STUDENT_DATA");
```

You can also create a sheet based upon the index using the ***getSheetAt*** (*int index*) method as shown below –

```
HSSFSheet sheet1=wb.getSheetAt(1);
HSSFRow row1=sheet.getRow(1);
sheet.getRow(1).getCell(1)
```

5. After you obtain the ***cell*** that contains the data, you can read the data in different formats like ***String, Date, Number*** using the different methods which are based upon the format of the cell you specify in the excel sheet.

- ***String*** – ***getStringCellValue****() [It can be used to read Name of the student from Excel]*
- ***Number*** – ***getNumericCellValue****() [It can be used to read the mobile number of the student]*
- ***Date*** – ***getDateCellValue****() [It can be used to read the Date of Birth of the student]*

What is the difference between Page Object Model (POM) and Page Factory:

Page Object is a class that represents a web page and hold the functionality and members.

Page Factory is a way to initialize the web elements you want to interact with within the page object when you create an instance of it.

Advantages of Page Object Model Framework:

- Code reusability –
- Code maintainability –
- Object Repository –
- Readability –

Q #11) What is a framework?

Answer: A framework is a set of the structure of the entire automation suite. It is also a guideline, which if followed can result in a structure that is easy to maintain and enhance.

These guidelines include:

- Coding standards
- Handling the test data
- Maintaining and handling the elements (object repository in QTP)
- Handling of environment files and properties file
- Reporting of data

- Handling logs

What are the attributes of a good framework?
Modular: The framework should be adaptable to change. Testers should be able to modify the scripts as per the environment or login information change.

- **Reusable:** The commonly used methods or utilities should be written in a common file that is accessible to all the scripts.
- **Consistent:** The suite should be written in a consistent format by following all the accepted coding practices.
- **Independent:** The scripts should be written in such a way that they are independent of each other. In case one test fails, it should not hold back the remaining test cases (unless it is a login page)
- **Logger:** It is good to have implemented the logging feature in the framework. This would help in case our scripts run for longer hours (say nightly mode), if the script fails at any point of time, having the log file will help us to detect the location along with the type of the error.
- **Reporting:** It is good to have the reporting feature automatically embedded into the framework. Once the scripting is done, we can have the results and reports sent via email.
- **Integration:** Automation Framework should be such that it is easy to integrate with other applications like continuous integration or triggering the automated script as soon as the build is deployed.

Can you do without a framework?
Answer: Frameworks are guidelines and not mandatory rules, so we can do without a framework, but if we create it and follow it, enhancing and maintaining would be easy to implement.

Q #17) Why do you want to keep this kind of information in a separate file and not directly in the code?
Answer: URL, Login, and passwords are the kind of fields that are used very often and these change as per the environment and authorization. In case we hardcode it into our code, we have to change it in every file which has its reference.
In case if there are more than 100 files, then it becomes very difficult to change all the 100 files and this, in turn, can lead to errors. So this kind of information is maintained in a separate file so that updating becomes easy.

Q #18) What are the different types of frameworks?
Answer: Different types of frameworks includes:

- Keyword-driven framework
- Data-Driven framework
- Hybrid Framework
- Linear Scripting

Q #22) How do you select which automation tool is best suited for you?

Answer: Selecting the automation tool depends upon various factors like:

- The scope of the application which we want to automate.
- Management overhead like cost and budget.
- Time to learn and implement the tool.
- Type of support available for the tool.
- Limitation of the tool

Q #28) When do you prefer Manual testing over Automation testing?
Answer: We prefer manual testing over automation testing in the following cases:

- The project is short-term and writing scripts will be time-consuming and costly when compared to manual testing.
- Flexibility is required. Automated test cases are programmed and run in a specific way of configurations.
- Usability testing needs to be performed.
- Applications/module is newly developed and has no previous test cases.
- Ad-hoc or exploratory testing needs to be performed.

Q #29) Is Automation testing in agile Methodology useful or not?
Answer: Automation testing is useful for regression, smoke or sanity testing. All these types of testing in the traditional waterfall model happen at the end of the cycle and sometimes if there are not many enhancements to the application, we might not even have to do regression testing.
Whereas, in agile methodology, every iteration requires executing the regression test case as some new functionalities is added.
Also, the regression suite itself keeps growing after each sprint as the functional test cases of the current sprint module need to be added to the regression suite for the next sprint.
Thus, Automation testing in agile methodology is very useful and helps in achieving maximum test coverage in less time of the sprint.

Q #30) List some advantages and disadvantages of Automation testing.
Answer:
Advantages:

- Fewer human resources
- Reusability
- More Test Coverage in less time
- Reliability
- Parallel execution of test cases
- Fast

Disadvantages:

- Development and maintenance time is more.
- Tool Cost
- Skilled resources are required.
- Environment setup
- Test Script debugging is an issue.

Q #31) List some advantages and disadvantages of Manual testing.
Answer:
Advantages:

- No environment setup needed.
- Programming knowledge is not required.
- Recommended for dynamically changing requirements.
- Allow human observation power to detect more bugs.
- The cost is less for short-term projects.
- Flexibility

Disadvantages:

- Difficult to perform complex calculations.
- Reusability
- Time taking
- High risk of human errors or mistakes.
- More human resources are required.

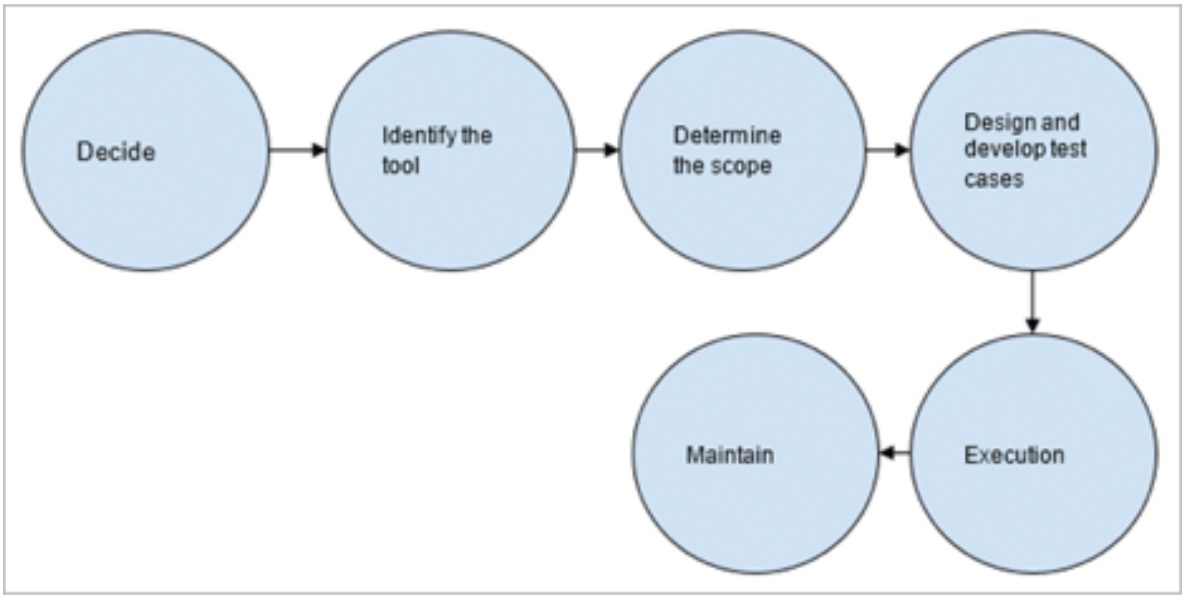
Q #34) Is Automation testing a Black box testing or White-box testing?
Answer: Automation testing is mostly a black box testing as we just program the steps that a manual tester performs for application under test without knowing the low-level design or code of the application.
Sometimes, automated test scripts need access to the database details that are used in the application under test or some more coding details and thus can be a type of white-box testing.
Thus automated testing can be both black or white box type of testing depending on the scenarios in which automation is performed.

Q #37) Which test cases can be automated?
Answer: Types of test cases which can be automated are:
(i) Smoke test cases: Smoke testing is also known as build verification testing. Smoke test cases are run every time when a new build is released to check the health of the build for acceptance to perform testing.
(ii) Regression Test Cases: Regression testing is the testing to ensure that previously developed modules are functioning as expected after a new module is added or a bug is fixed.
Regression test cases are very crucial in incremental software approach where a new functionality is added at each increment phase. In this case, regression testing is performed at each incremental phase.
(iii) Complex Calculation test cases: Test cases which involve some complex calculations to verify a field for an application fall into this category. Complex calculation results are more prone to human errors hence when automated they give accurate results.
(iv) Data-driven test cases: Test cases which have the same set of steps and run multiple times with the change of data are known as data-driven test cases. Automated testing for these kinds of test cases is quick and cost-effective.

(v) Non-functional test cases: Test cases like load tests and performance tests require a simulated environment with multiple users and multiple hardware or software combinations.
Setting up multiple environments manually is impossible for each combination or number of users. Automated tools can easily create this environment to perform non-functional testing easily.

Q #38) What are the phases in Automation testing Life Cycle?
Answer: The phases in Automation testing life Cycle include:

1. The decision to perform automation testing.
2. Identify and learn about the automation tool.
3. Determine the scope of automation testing.
4. Design and develop a test suite.
5. Test Execution
6. Maintenance of test scripts.



Q #39) What is an Automated test script?
Answer: An automated test script is a short program that is written in a programming language to perform a set of instructions to verify if the application is as per the requirements.

3. Why did you want to use Framework?
Answer:
In Test Automation Framework, we need to handle many files. To organize those files and systemically complete all tasks and achieve the goal successfully, we have to support any Framework.

. Which Framework you have used in your Selenium Project?
Answer:
Used Data-Driven Testing Framework where Page Object Model design pattern has been used along with Page Factory.

6. Where do you apply Object-Oriented Programming concepts in Test Automation Framework?
Answer:
OOPs is used in most of places while writing the Test Automation Framework. There are many OOPs concept that has been used for Test Automation Frameworks such as Abstraction, Polymorphism, Inheritance, Interface, Method Overloading and Overriding and Encapsulation.
In the case of Abstraction, we need to write locators like XPath, name, id, etc., inside Page Class for Page Object Model Design Pattern. In the case of Interface, Selenium uses WebDriver, which is an Interface. Likewise, the other OOPs concept has been used in Test Automation Framework in different scenarios.

8. What type of Test Cases did you prefer to be automated?
Answer:
The test cases I preferred to be automated are as below:

- Sanity Test Cases
- Regression Test Cases
- Smoke Test Cases

9. Why do you like to use Selenium?
Answer:

- Selenium is free to download and use and also open source.
- Cross-browser compatibility supports by Selenium.
- Multi-language support and very easy compatibility with Java.
- A large no of users are using Selenium as a Test Automation tool; that's why the help communities are larger compared to the other test Automation tools.

2) When will you automate a test?
Automation in preferred in following cases

- Repetitive Tasks
- Smoke and Sanity Tests
- Test with multiple data set
- Regression test cases

3) When will you not automate testing?
One should not automate in following cases

- When the Application Under Test changes frequently
- One time test cases
- Adhoc – Random testing

4) What are the steps involved in the Automation Process?
In the automation process, steps involved are

- Selecting the Test tool

- Define scope of automation
- Planning, design, and development
- Test execution
- Maintenance

6) In what condition we cannot use automation testing for the Agile method?

Automation testing is not useful for agile methods in following conditions

- When Agile testing always ask for changes in requirements
- When Exhaustive level of documentation is required in Agile
- Only suitable for those regression tests during agile testing like continuous integration

Different types of Locators in Selenium are as follows:

i. ID
ii. Name
iii. Class Name
iv. Tag Name
v. Link Text & Partial Link Text
vi. CSS Selector
vii. XPath

How To Write Dynamic XPath In Selenium WebDriver

Let's see different way of writing dynamic XPath in Selenium with examples:

1. Using Single Slash
2. Using Double Slash
3. Using Single Attribute
4. Using Multiple Attribute
5. Using AND
6. Using OR
7. Using contains()
8. Using starts_with()
9. Using text()
10. Using last()
11. Using position()

```
<ul id="automation">
  <li>Selenium</li>
  <li>QTP</li>
  <li>Sikuli</li>
</ul>
```

css="ul#automation li:nth-of-type(2)". QTP

css="ul#automation li:last-child" Sikuli

CSS Selector:-

- *Tag and ID*
  css=tag#id

- *Tag and Class*
  css=tag.class

- *Tag and Attribute*
  css=tag[attribute=value]
  css=input[name=Email]

- *Tag, Class, and Attribute*
  css=tag.class[attribute=value]

Maven vs Gradle

Gradle is a dependency management and a build automation tool that **was built upon the concepts of Ant and Maven.**
One of the first things we can note about Gradle is that it's not using XML files, unlike Ant or Maven.
Apache Maven is a dependency management and a build automation tool, primarily used for Java applications. **Maven continues to use XML files just like Ant but in a much more manageable way.**

Some significant benefits of Gradle are as following:

- Gradle allows us to write the build script with Java programing language.
- It is easy to use and maintain.
- It supports dependency management
- It provides high performance and scalable builds.
- Gradle integration process is quite easier.
- It supports a multi-project structure.
- It is easy to migrate to Gradle from Maven or other build tools.

Q.Selenium assert or verify? Which one do you prefer?
Basic Difference: 'assert' terminates the execution whereas 'verify' marks the step as fail but continue the execution.

```
SoftAssert softAssert = new SoftAssert();

  //Assertion failing
  softAssert.fail("Failing first assertion");
  System.out.println("Failing 1");

  //Assertion failing
  softAssert.fail("Failing second assertion");
```

System.out.println("Failing 2");
//Collates the assertion results and marks test as pass or fail
softAssert.assertAll();

A. Annotations are lines of code that provide the configuration information. They are always preceded by @ symbol.

@BeforeSuite: Once, before all Tests in the suite.
@BeforeTest: Once, Prior to the first Test case in the TestNG file.
@BeforeClass: Once before the first Test method in the current class.
@BeforeMethod: Before each Test method.
@Test: Actual Test case, the business logic.
@AfterMethod: After each Test method.
@AfterClass: Once after all the Test methods in the current class.
@AfterTest: Once, after all Test cases in the TestNG file.
@AfterSuite: Once, after all Tests in the suite.

@BeforeGroups: The list of groups that this method will run before.
@AfterGroups: The list of groups that this method will run after.
@Parameters: to pass parameters in Test methods.
@DataProvider: Marks a method as supplying data for a Test method.
@Factory: Marks a method as a factory that returns objects that will be used by TestNG as Test classes.
@Listeners: Defines listeners on a Test class, helpful for logging purpose

Q.What does 'Abstract' mean in Java?
A. Abstraction is a process of hiding the implementation details and only show important functionality. 'abstract' keyword is used to create an abstract class and method. The purpose of an abstract class is to specify the default functionality of an object and let its sub-classes to explicitly implement that functionality. Thus, it stands as an abstraction layer that must be extended and implemented by the corresponding sub-classes.

- Abstract class can't be instantiated (it needs to be extended) << abstract class A {} >> | an abstract method contains a method signature, but no method body. << abstract void methodName(); >>
- Abstract class is used to provide default/common method implementation to all the subclasses.
- If you are extending any abstract class that have abstract methods, you must either provide the implementation of all abstract methods or make this sub-class abstract.
- An abstract class can have both abstract and non-abstract (or concrete) methods.
- If a class have abstract methods, then the class should also be abstract.

- An abstract class can have data member, abstract method, concrete method (body), constructor and even main() method.
- An abstract class may have static fields and static methods.
- Abstract method can never be final and static.

Abstract class and methods are usually declared where two or more subclasses are expected to do a similar thing in different ways through different implementations. Most common e.g. Shape as the abstract class >> its implementation provided by the Rectangle and Circle classes.
Note: Abstract classes are not Interfaces. They are different.

Q. What are some risks you mention in a Test Plan?
A. Some common risks every project may face/have faced,
- Timelines. The first & foremost. Any downtime or requirement changes may impact the timelines.
- Resourcing. Skilled employees & required hardware/software.
- Scope. Any scope change needs to be analyzed & assessed.
- Third-party Dependencies. Availability of third-party systems & teams.
- Environment. Environmental downtime.

Q.*One of the most common Testing situation – How do you handle timeline crunch? ⏳
- Consider some buffer in planning phases to accommodate the usual delays.
- The most common: Extended & Weekend working hours.
- Prioritize the Test efforts and communicate the same to all the stakeholders.
- Testing as possible in the given timelines.
- Highlight the delays to buy some extra time.
- Plan & make efficient use of working 9 hours.
- Convey the situation as-is to the stakeholders, asking for some time to complete testing.

Whatever be the approach, always take a retrospective look once the testing is complete in order to avoid delays the next time.

#Q 2) What are the fields in a Bug Report?
Answer: Following important fields should be included in a good Bug Report:
A unique ID
Defect Description: a short describing what the bug is.
Steps to Reproduce: details about how to arrive at the error, exact test data, the time at which defect was found(if applicable)
environment: any information that will help re-encounter the issue
Module/section of the application (if applicable)
Severity
Screenshot

Responsible QA: in case of any follow-up questions regarding this issue

Severity and Priority:-
Severity means how defect is affecting the functionality. Priority means how fast defect has to be fixed.
Severity is related to the quality standard. Priority is related to scheduling to resolve the problem
The priority status is set based on the customer requirements. For example: **If the company name is misspelled in the home page of the website, then the priority is high and severity is low to fix it**.

#Q 4) How to overcome the challenge of not having input documentation for testing?
A previous version of the application
Talk to the developers or the business analysts (when available) to get a confirmation on our understanding or clarifications in case of doubts.

What is Agile Testing?
The agile testing is a software practice, it is started from the beginning of the project with continuous integration between development and testing, unlike the waterfall method. It is a continuous development methodology, where the requirements evolve between the customer and the self–organizing teams.

How Agile Testing is beneficial?
There are some of the points which will clear that how agile testing is beneficial. Following are the points:

- *In this less documentation is required.*
- *Through daily meetings, the issues can be well determined at earlier stage.*
- *It saves time and money both.*
- *It provides regular feedback from the end-user.*

Basic fundamentals of Agile Testing
Some of the basic fundamentals of Agile Testing are:

- ***Test documentation reduced****– The agile testing reduces the test documentation and as it reduces the length of the documentation so, it makes the testers to focus on the test rather than on the details.*
- ***It is not a phase-*** *The team performs testing continuously and through continuous testing, it provides continuous progress.*
- ***Implementation-*** *The agile testing is performed while implementation compares to other testing methods that are performed after implementation.*
- ***Fixing defects-*** *The defects which are raised during the iteration are fixed within the same iteration and therefore keeping the code clean.*

What are the strategies of Agile Testing?
The agile testing life cycle has four stages:

- **Iteration 0** – *This stage includes the identifying of people for testing, installing the testing tools, etc*. **Construction Iterations** – *Most of the testing part is performed in this stage.*
- **Transition Phase** – *In the transition phase, the main purpose of this phase is to deploy our system successfully into production phase.*
- **Production phase** – *This is the last stage after the transition phase and in this the product is moved to the production stage.*

18. What is Test Suite?
Test Suite is a collection of test cases. The test cases which are intended to test an application.

19. What is Test Scenario?
Test Scenario gives the idea of what we have to test. Test Scenario is like a high-level test case.

20. What is Test Case?
Test cases are the set of positive and negative executable steps of a test scenario which has a set of pre-conditions, test data, expected result, post-conditions and actual results. Click here for more details.

100. What is the difference between a Standalone application, Client-Server application and Web application?
Standalone application:
one-tier architecture. Presentation, Business, and Database layer are in one system for a single user.
Client-Server Application:
two-tier architecture. Presentation and Business layer are in a client system and Database layer on another server. It works majorly in Intranet.
Web Application:
three-tier or n-tier architecture. The presentation layer is in a client system, a Business layer is in an application server and
Database layer is in a Database server.

Q #12) What is an XPath?
XPath is used to locate a web element based on its XML path.

Q #13) What is the difference between "/" and "//" in Xpath?
Single Slash "/" – Single slash is used to create Xpath with absolute path i.e. the xpath would be created to start selection from the document node/start node.
Double Slash "//" – Double slash is used to create Xpath with relative path i.e. the xpath would be created to start selection from anywhere within the

document.

Xpath can be created in two ways:
Relative Xpath
Relative Xpath begins from the current location and is prefixed with a "//".
For example: //span[@class='Email']
Absolute Xpath
Absolute Xpath begins with a root path and is prefixed with a "/".
For example: /HTML/body/div/div[@id='Email']

Q #15) When should I use Selenium Grid?
Selenium Grid can be used to execute same or different test scripts on multiple platforms and browsers concurrently so as to achieve distributed test execution, testing under different environments and saving execution time remarkably.

Q #20) What are the different types of waits available in WebDriver?
There are two types of waits available in WebDriver:

1. Implicit Wait
2. Explicit Wait

Implicit Wait: Implicit waits are used to provide a default waiting time (say 30 seconds) between each test step/command across the entire test script.

driver.manage().timeouts().implicitlyWait(TimeOut, TimeUnit.SECONDS);

Explicit Wait: Explicit waits are used to halt the execution till the time a particular condition is met or the maximum time has elapsed. Unlike Implicit waits, explicit waits are applied for a particular instance only.
WebDriverWait wait = new WebDriverWait(WebDriverRefrence,TimeOut);
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//input[@id='name'])));

1. elementToBeClickable()
2. elementToBeSelected()
3. frameToBeAvaliableAndSwitchToIt()
4. invisibilityOfTheElementLocated()
5. invisibilityOfElementWithText()
6. presenceOfAllElementsLocatedBy()
7. presenceOfElementLocated()

Q #17) Which is the latest Selenium tool?
WebDriver

Q #18) How do I launch the browser using WebDriver?
The following syntax can be used to launch Browser:
*WebDriver driver = **new** FirefoxDriver();*

*WebDriver driver = **new** ChromeDriver();*
*WebDriver driver = **new** InternetExplorerDriver();*

Q #19) What are the different types of Drivers available in WebDriver?
The different drivers available in WebDriver are:

- FirefoxDriver
- InternetExplorerDriver
- ChromeDriver
- SafariDriver
- OperaDriver
- AndroidDriver
- IPhoneDriver
- HtmlUnitDriver

Q #21) How to type in a textbox using Selenium?
The user can use sendKeys("String to be entered") to enter the string in the textbox.
Syntax:
WebElement username = driver.findElement(By.id("Email"));
// entering username
username.sendKeys("sth");

Q #22) How can you find if an element in displayed on the screen?

1. isDisplayed()
2. isSelected()
3. isEnabled()

Syntax:
isDisplayed():
***boolean** buttonPresence = driver.findElement(By.id("gbqfba")).isDisplayed();*
isSelected():
***boolean** buttonSelected = driver.findElement(By.id("gbqfba")).isSelected();*
isEnabled():
***boolean** searchIconEnabled = driver.findElement(By.id("gbqfb")).isEnabled();*

Q #23) How can we get a text of a web element?
Syntax:
String Text = driver.findElement(By.id("Text")).getText();

Q #24) How to select value in a dropdown?
The value in the dropdown can be selected using WebDriver's Select class.
Syntax:
selectByValue:
*Select selectByValue = **new** Select(driver.findElement(By.id("SelectID_One")));*
*selectByValue.**selectByValue**("greenvalue");*
selectByVisibleText:
*Select selectByVisibleText = **new** Select*

(driver.findElement(By.id("SelectID_Two")));
*selectByVisibleText.s**electByVisibleTex**t("Lime");*
selectByIndex:
Select selectByIndex
*= **new** Select(driver.findElement(By.id("SelectID_Three")));*
*selectByIndex.**selectByIndex**(2);*
***getOptions()** return List<WebElements> inside DropDown*

Q #25) What are the different types of navigation commands?
Following are the navigation commands:
navigate().back() – The above command requires no parameters and takes back the user to the previous webpage in the web browser's history.
Sample code:
driver.navigate().back();
navigate().forward() – This command lets the user to navigate to the next web page with reference to the browser's history.
Sample code:
driver.navigate().forward();
navigate().refresh() – This command lets the user to refresh the current web page there by reloading all the web elements.
Sample code:
driver.navigate().refresh();
navigate().to() – This command lets the user to launch a new web browser window and navigate to the specified URL.
Sample code:
driver.navigate().to("https://google.com");

Q #26) How to click on a hyper link using linkText?

*driver.findElement(**By.linkText**("Google")).click();*
The command finds the element using link text and then click on that element and thus the user would be re-directed to the corresponding page.
The above-mentioned link can also be accessed by using the following command.
*driver.findElement(**By.partialLinkText**("Goo")).click();*
The above command finds the element based on the substring of the link provided in the parenthesis and thus partialLinkText() finds the web element with the specified substring and then clicks on it.

Q #27) How to handle frame in WebDriver?
An inline frame acronym as iframe is used to insert another document within the current HTML document or simply a web page into a web page by enabling nesting.
Select iframe by id
driver.switchTo().frame("ID of the frame");
Locating iframe using tagName
driver.switchTo().frame(driver.findElements(By.tagName("iframe").get(0));

Locating iframe using index
frame(index)
driver.switchTo().frame(0);
frame(Name of Frame)
driver.switchTo().frame("name of the frame");
frame(WebElement element)

//Switch back to main page
driver.switchTo().defaultContent();

//Switch back to parent frame
driver.switchTo().parentFrame();

Different b/w driver.get() and driver.navigate().to()

1. get() method **holds or waits till page is loaded** while navigate() does not.
2. get() method does not retain any **browsing history** while navigate() does.

Q #28) When do we use findElement() and findElements()?
findElement(): findElement() is used to find the first element in the current web page matching to the specified locator value. Take a note that only first matching element would be fetched.

Syntax:
WebElement element = driver.findElements(By.xpath("//div[@id='example']//ul//li"));
findElements(): findElements() is used to find all the elements in the current web page matching to the specified locator value. Take a note that all the matching elements would be fetched and stored in the list of WebElements.
Syntax:
List <WebElement> elementList = driver.findElements(By.xpath("//div[@id='example']//ul//li"));

Q #30) What is the difference between driver.close() and driver.quit command?
close(): WebDriver's close() method closes the web browser window that the user is currently working on or we can also say the window that is being currently accessed by the WebDriver. The command neither requires any parameter nor does it return any value.
quit(): Unlike close() method, quit() method closes down all the windows that the program has opened. Same as close() method, the command neither requires any parameter nor does is return any value.

Q #31) Can Selenium handle windows based pop up?
Selenium is an automation testing tool which supports only web application testing. Therefore, windows pop up cannot be handled using Selenium.

Q #32) How can we handle web-based pop-up?
WebDriver offers the users a very efficient way to handle these pop-ups using Alert interface. There are the four methods that we would be using along with the Alert interface.

- void dismiss() – The dismiss() method clicks on the "Cancel" button as soon as the pop-up window appears.
- void accept() – The accept() method clicks on the "Ok" button as soon as the pop-up window appears.
- String getText() – The getText() method returns the text displayed on the alert box.
- void sendKeys(String stringToSend) – The sendKeys() method enters the specified string pattern into the alert box.

Syntax:
// accepting javascript alert
Alert alert = driver.switchTo().alert();
alert.accept();

Q #33) How can we handle windows based pop up?
There are several third-party tools available for handling window based pop-ups along with the selenium like AutoIT, Robot class etc.

Q #34) How to assert the title of the web page?
//verify the title of the web page
assertTrue("The title of the window is incorrect.",driver.getTitle().equals("Title of the page"));

Q #35) How to mouse hover on a web element using WebDriver?.
Sample Code:
Actions actions=new Actions(driver);
actions.moveToElement(driver.findElement(By.id("id of the dropdown"))).perform();
// Clicking on one of the items in the list options
WebElement subLinkOption=driver.findElement(By.id("id of the sub link"));
subLinkOption.click();

Q #40) What is TestNG and how is it better than Junit?
There are various advantages that make TestNG superior to JUnit. Some of

them are:

- Added advance and easy annotations
- Execution patterns can set
- Concurrent execution of test scripts
- Test case dependencies can be set

Q #41) How to set test case priority in TestNG?
Setting Priority in TestNG

Code Snippet

```
package TestNG;
import org.testng.annotations.*;
public class SettingPriority {
@Test(priority=0)
public void method1() {
}
@Test(priority=1)
public void method2() {
}
@Test(priority=2)
public void method3() {
}
}
```

Test Execution Sequence:

1. Method1
2. Method2
3. Method3

Q #42) What is a framework?
The framework is a constructive blend of various guidelines, coding standards, concepts, processes, practices, project hierarchies, modularity, reporting mechanism, test data injections etc. to pillar automation testing.

Q #44) What are the different types of frameworks?
Below are the different types of frameworks:

1. **Data Driven Testing Framework**: Data Driven Testing Framework helps the user segregate the test script logic and the test data from each other. It lets the user store the test data into an external database. The data is conventionally stored in "Key-Value" pairs. Thus, the key can be used to access and populate the data within the test scripts.
2. **Keyword Driven Testing Framework:** The Keyword Driven testing framework is an extension to Data-driven Testing Framework in a sense that it not only segregates the test data from the scripts, it also keeps the certain set of code belonging to the test script into an external data file.

3. **Hybrid Testing Framework:** Hybrid Testing Framework is a combination of more than one above mentioned frameworks. The best thing about such a setup is that it leverages the benefits of all kinds of associated frameworks.
4. **Behavior Driven Development Framework:** Behavior Driven Development framework allows automation of functional validations in an easily readable and understandable format to Business Analysts, Developers, Testers, etc.

Double click in Selenium
Double click in Selenium using Actions class
Actions actions = new Actions(driver);
WebElement elementLocator = driver.findElement(By.id("ID"));
actions.doubleClick(elementLocator).perform();

Right-click in Selenium
Right Click operation is also called Context Click in Selenium.

Actions actions = new Actions(driver);
WebElement elementLocator = driver.findElement(By.id("ID"));
actions.contextClick(elementLocator).perform();

3. Explain the different exceptions in Selenium WebDriver.
Exceptions in Selenium are similar to exceptions in other programming languages. The most common exceptions in Selenium are:

- **TimeoutException:** This exception is thrown when a command performing an operation does not complete in the stipulated time
- **NoSuchElementException:** This exception is thrown when an element with given attributes is not found on the web page
- **ElementNotVisibleException:** This exception is thrown when the element is present in DOM (Document Object Model), but not visible on the web page
- **StaleElementException:** This exception is thrown when the element is either deleted or no longer attached to the DOM

11. What is the use of JavaScriptExecutor?
JavaScriptExecutor is an interface which provides a mechanism to execute Javascript through the Selenium WebDriver. It provides "**executescript**" and "**executeAsyncScript**" methods, to run JavaScript in the context of the currently selected frame or window. An example of that is:
JavascriptExecutor js = (JavascriptExecutor) driver;
js.executeScript(Script,Arguments);

12. How to scroll down a page using JavaScript in Selenium?
We can scroll down a page by using window.scrollBy() function. Example:

((JavascriptExecutor) driver).executeScript("window.scrollBy(0,500)");

13. How to scroll down to a particular element?

To scroll down to a particular element on a web page, we can use the function **scrollIntoView()**. Example:

((JavascriptExecutor) driver).executeScript("arguments[0].scrollIntoView();", element);

18. What is Selenese?

Selenese is the set of selenium commands which are used to test your web application used By Selenium IDE

You can even make use of:

- **Actions**: Used for performing operations Execution of Actions generates events like click this link, select that option, type this box, etc. If an Action fails, or has a bug, the execution of current test is stopped.
- **Accessors:** Used for storing a value in a particular variable examine state of the application and store the results in variables.
- **Assertions**: Used as checkpoints. verify the state of the application. Assertions are generally used in three modes assert, verify and waitfor.
-

27. How to take screenshots in Selenium WebDriver?

You can take a *screenshot* by using the **TakeScreenshot** function. By using **getScreenshotAs()** method you can save that screenshot. Example:

File scrFile = ((TakeScreenshot)driver).getScreenshotAs(outputType.FILE);

32. Can we enter text without using sendKeys()?

Yes. We can enter/ send text without using **sendKeys()** method. We can do it using JavaScriptExecutor.

JavascriptExecutor jse = (JavascriptExecutor) driver;
jse.executeScript("document.getElementById('Login').value=Test text without sendkeys");

33. Explain how you will login into any site if it is showing any authentication popup for username and password?

WebDriverWait wait = new WebDriverWait(driver, 10);
Alert alert = wait.until(ExpectedConditions.alertIsPresent());
*alert.authenticateUsing(new UserAndPassword(**username**, **password**));*

34. Explain how can you find broken links in a page using Selenium WebDriver?

First, we have to use the anchor tags <a> to determine the different hyperlinks on the web page.

For each <a> tag, we can use the attribute 'href' value to obtain the hyperlinks and then analyze the response received for each hyperlink when used in **driver.get()** method.

```
public class BrokenLinks {
   public static void main(String[] args) {
      System.setProperty("webdriver.chrome.driver","./src/resources/
chromedriver");
      WebDriver driver = new ChromeDriver();
      driver.manage().window().maximize();
      driver.get("https://demoqa.com/broken");

      //Storing the links in a list and traversing through the links
      List<WebElement> links = driver.findElements(By.tagName("a"));

      // This line will print the number of links and the count of links.
      System.out.println("No of links are "+ links.size());

      //checking the links fetched.
      for(int i=0;i<links.size();i++)
      {
         WebElement E1= links.get(i);
         String url= E1.getAttribute("href");
         verifyLinks(url);
      }

      driver.quit();
   }


   public static void verifyLinks(String linkUrl)
   {
      try
      {
         URL url = new URL(linkUrl);

         //Now we will be creating url connection and getting the response code

         HttpURLConnection
httpURLConnect=(HttpURLConnection)url.openConnection();
         httpURLConnect.setConnectTimeout(5000);
         httpURLConnect.connect();
         if(httpURLConnect.getResponseCode()>=400)
         {
          System.out.println(linkUrl+" -
"+httpURLConnect.getResponseMessage()+"is a broken link");
         }

         //Fetching and Printing the response code obtained
```

```
        else{
           System.out.println(linkUrl+" -
"+httpURLConnect.getResponseMessage());
        }
      }catch (Exception e) {
    }
  }
}
```

37. What is parameterization in TestNG? How to pass parameters using testng.xml?

Parameterization is the technique of defining values in testng.xml file and sending them as parameters to the test class. This technique is especially useful when we need to pass multiple login credentials of various test environments. Take a look at the code below, in which “myName” is annotated as a parameter.

```
1  public class
2  ParameterizedTest1{
3   @Test
4   @Parameters("myName")
5   public void
6  parameterTest(String
7  myName) {

   System.out.println("Param
   eterized value is : " +
   myName);
   }
  }
```

To pass parameters using testng.xml file, we need to use ‘parameters’ tag. Look at the below code for example:

```
<?xml version="1.0"
encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "<a
href="http://testng.org/
testng-1.0.dtd">http://testng.org/
testng-1.0.dtd</a>" >
 <suite name=″CustomSuite">
  <test name=″CustomTest″>
   <parameter name="myName"
value=″John"/>
    <classes>
     <class
name="ParameterizedTest1" />
    </classes>
  </test>
 </suite>
```

38. Explain DataProviders in TestNG using an example. Can I call a single data provider method for multiple functions and classes?

DataProvider is a TestNG feature, which enables us to write DataDriven tests.
it supports DataDriven testing.
the same test method can run multiple times with different data-sets.
@DataProvider marks a method as supplying data for a test method. The annotated method must return an Object[] where each Object[] can be assigned to parameter list of the test method.

39. How to skip a method or a code block in TestNG?

@Test(enabled = false)

40. What is soft assertion in Selenium? How can you mark a test case as failed by using soft assertion?

To mark a test as failed with soft assertions, call **assertAll()** method at the end of the test.

2. How does TestNG allow you to state dependencies? Explain it with an example.

Dependency is a feature in TestNG that allows a test method to depend on a single or a group of test methods.
Syntax: @Test(dependsOnMethods = { “initEnvironmentTest” })

```
@Test(groups={“Car”})
public void drive(){
system.out.println(“Driving the
vehicle”);
}

@Test(dependsOnMethods={“drive
”},groups={cars})
public void changeGear() {
system.out.println("Change
Gears”);
}

@Test(dependsOnMethods={“chan
geGear”},groups={“Car”})
public void accelerate(){
system.out.println(“Accelerating”);
}
```

43. Explain what does @Test(invocationCount=?) and @Test(threadPoolSize=?) indicate.

@Test(invocationCount=?) is a parameter that indicates the number of times this method should be invoked.

@Test(threadPoolSize=?) is used for executing suites in parallel. Each suite can be run in a separate thread.

To specify how many times @Test method should be invoked from different threads, you can use the attribute **threadPoolSize** along with **invocationCount**. Example:

```
@Test(threadPoolSize = 3,
invocationCount = 10)
public void testServer() {
}
```

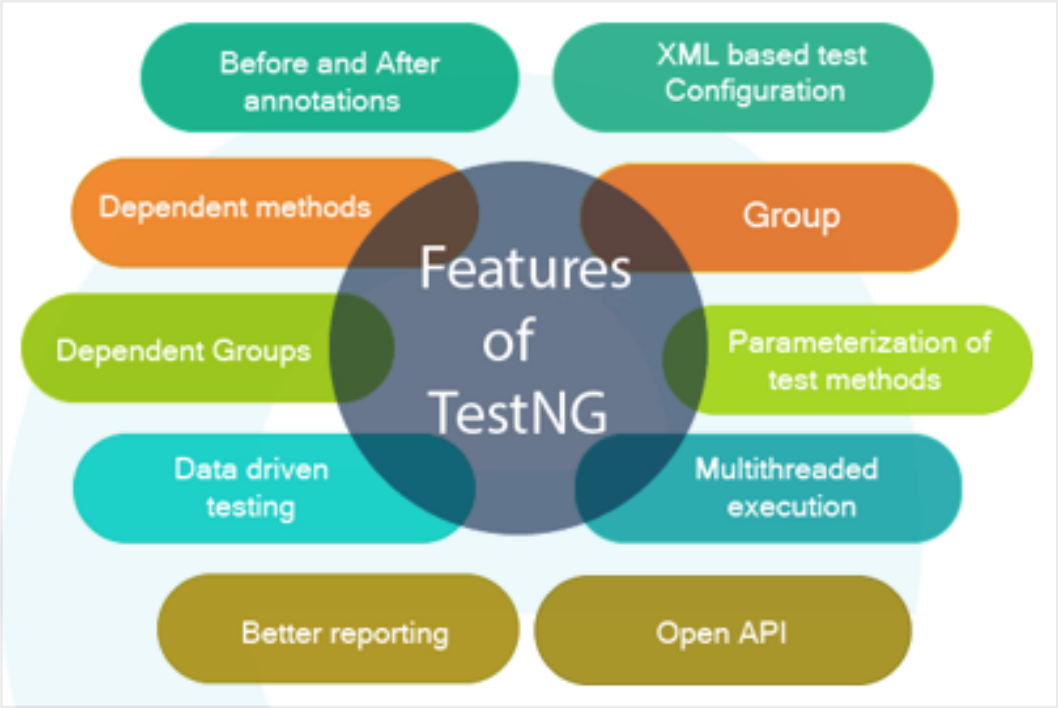
46. How do you achieve synchronization in WebDriver?

It is a mechanism which involves more than one components to work parallel with each other.

Synchronization can be classified into two categories:

- **Unconditional:** In this we just specify timeout value only. We will make the tool to wait until certain amount of time and then proceed further.
- **Conditional**: It specifies a condition along with timeout value, so that tool waits to check for the condition and then come out if nothing happens.

48. What are the features of TestNG?



Run Class Parallel in TestNG
<suite name="Regression" verbose="1" parallel="classes" thread-count="1">

Group test in TestNG
@Test(**description** = "Add User with 6 Characters Test", **enabled** = false, **groups** = {"Regression", "Smoke"})

Authentication Vs Authorization
Authentication is the process of verifying who a user is,
while **Authorization** is the process of verifying what they have access to.

What is **functional** and **Non-functional** testing?
Functional testing ensures that functions and features of the application work properly. Non-functional testing examines other aspects of how well the application works.

Functional testing types

- Unit testing.
- Component testing.
- Smoke testing.
- Sanity testing.
- Regression testing.
- Integration testing.
- API testing.
- UI testing.

Key Difference Between Sanity and Smoke Test

- Smoke Testing has a goal to verify “stability” whereas Sanity Testing has a goal to verify “rationality”.
- Smoke Testing is done by both developers or testers whereas Sanity Testing is done by testers.
- Smoke Testing verifies the critical functionalities of the system whereas Sanity Testing verifies the new functionality like bug fixes.
- Smoke testing is a subset of acceptance testing whereas Sanity testing is a subset of Regression Testing.
- Smoke testing is documented or scripted whereas Sanity testing isn’t.
- Smoke testing verifies the entire system from end to end whereas Sanity Testing verifies only a particular component.

Conver text file to string and pass it to body in API request by

new String(Files.readAllBytes(Paths.get(path)));

}

Can you achieve 100% automation?
Answer: 100% automation would be difficult to achieve because there would be many edge test cases and some cases that are executed seldom. Automating these cases which are not executed that often will not add value to the automated suite.

What are the types of automation testing?

1. Unit tests: These are written by software developers and test a unit of code in isolation.
2. Integration tests: These test how well different software components work with each other.
3. Regression tests: Verify that the new code didn't break any existing functionality.
4. Performance tests: Ensure that the software won't crash and perform reasonably under heavy load or stringent conditions.
5. UI tests: Ensure that the software uses a consistent user experience and no visual or graphical elements on the screen are broken.

What is a test environment?

A test environment is a computer or a server on which a tester tests the software. After the team builds the software, the tester installs it on this computer with all its dependencies, just like the production environment. This

allows the tester to test the software in a real-world scenario.

What is cross-browser testing?

With web applications, you don't know in advance which browsers your users will use. Hence, it's crucial to test the web application or the website on multiple major browsers running on different operating systems.
Cross-browser testing is a type of browser automation testing where the tester verifies if the web application will work smoothly on different browsers. Some of the popular browsers include Google Chrome, Mozilla Firefox, Internet Explorer, Safari, etc.

How do you automate the testing of CAPTCHA?

It's not possible to automate the testing of CAPTCHA. That is the goal behind any good CAPTCHA strategy. By definition, a computer can't automate it. If it could, then it's not a good challenge that you can use in your application.

What is the difference between build and release?

Build: It is ***software that is given to the testing team by the development team.***
Release: It is ***software that is handed over to the customer by the tester or developer.***

What is bug leakage and bug release?

Bug release is when software or an application is **handed over to the testing team knowing that the defect is present** in a release. During this the priority and severity of bug is low, as **bug can be removed before the final handover**.
Bug leakage is something, when the **bug is discovered by the end users or customer, and not detected by the testing team** while testing the software.

What does the test strategy include?

The test strategy includes an introduction, resource, scope and schedule for test activities, test tools, test priorities, test planning and the types of test that has to be performed.

What is the strategy for Automation Test Plan?

- The strategy for Automation Test Plan
- Preparation of Automation Test Plan
- Recording the scenario
- Error handler incorporation
- Script enhancement by inserting check points and looping constructs
- Debugging the script and fixing the issues
- Rerunning the script
- Reporting the result

Explain stress testing, load testing and volume testing?

- Load Testing: Testing an application under **heavy but expected load i**s known as Load Testing. Here, the load refers to the large volume of users, messages, requests, data, etc.
- Stress Testing: When the load placed on the system is **raised or accelerated beyond the normal rang**e then it is known as Stress Testing.
- Volume Testing: The process of checking the system, whether the system can **handle the required amounts of data,** user requests, etc. is known as Volume Testing.

What is a 'USE' case and what does it include?

The document that describes, the user action and system response, for a particular functionality is known as USE case. It includes revision history, table of contents, flow of events, cover page, special requirements, pre-conditions and post-conditions.

What is Integration testing?

Integration testing is a software testing method that combines and tests individual application components. It is generally performed after unit and functional Testing.

What is the meaning of End-To-End Testing?

End To End testing is a method of testing an application that helps you to ensure whether it is acting as expected to be working. That should be used to test the application flow from the start to the end.
This testing method aims to examine the entire system's flow. It also confirms that the data integrity is maintained between the different system components and the systems.

What is Grey Box Testing?

Grey box is the combination of White Box and Black Box Testing. Testers involved in this type of Testing should have access to the design documents. It helps to create better test cases in this process.

How is validation different from verification?

Verification	Validation
Verification means checking the documents, languages, designs, and other programming things.	Validation means testing the actual product.
Verification is a type of static Testing.	Validation is a type of dynamic Testing.
It does not need you to execute the code.	It requires code execution.

It includes checking documents delivered by humans.	It includes the execution of a program executed by a computer.

Authentication verifies who the user is. **Authorization** determines what resources a user can access

In the authentication process, the identity of users are checked for providing the access to the system.	While in authorization process, a the person's or user's authorities are checked for accessing the resources.
In the authentication process, users or persons are verified.	While in this process, users or persons are validated.
It is done before the authorization process.	While this process is done after the authentication process.
It needs usually the user's login details.	While it needs the user's privilege or security levels.
Authentication determines whether the person is user or not.	While it determines **What permission does the user have?**
Generally, transmit information through an **ID Token.**	Generally, transmit information through an **Access Token**.
The OpenID Connect (OIDC) protocol is an authentication protocol that is generally in charge of user authentication process.	The OAuth 2.0 protocol governs the overall system of user authorization process.
Popular Authentication Techniques- • Password-Based Authentication • Passwordless Authentication • 2FA/MFA (Two-Factor Authentication / Multi-Factor Authentication) • Single sign-on (SSO) • Social authentication	Popular Authorization Techniques- • Role-Based Access Controls (RBAC) • JSON web token (JWT) Authorization • SAML Authorization • OpenID Authorization • OAuth 2.0 Authorization
The user authentication is visible at user end.	The user authorization is not visible at the user end.

The user authentication is identified with username, password, face recognition, retina scan, fingerprints, etc.	The user authorization is carried out through the access rights to resources by using roles that have been pre-defined.
Example: Employees in a company are required to authenticate through the network before accessing their company email.	**Example:** After an employee successfully authenticates, the system determines what information the employees are allowed to access.

What is Client-side Validation?

Client-side validation is done at the browser level, where the user's input is validated at the browser itself without the involvement of the server.

What is Acceptance testing?

Acceptance Testing is a type of Testing performed by the end-user or the client. It verifies or accepts the software system before moving the software apps to the production environment.

What is Unit Testing?

Unit Testing is a kind of software testing when individual units or components of a software are tested. This type of Testing is conducted to check whether the source code's modules are working correctly.

What is data-driven testing?

The aim of data-driven automation testing is to simplify the testing process involving complex and huge data sets. In Data Driven Testing, the test data includes input, expected output, and a result field. These fields are listed in CSV files, excel files, text files, XML files, etc. These files are then fed to the automation tool for execution, which compares the expected and actual data. Then the obtained results are documented in the result field.

What are the four elements that you have to pass in Selenium?

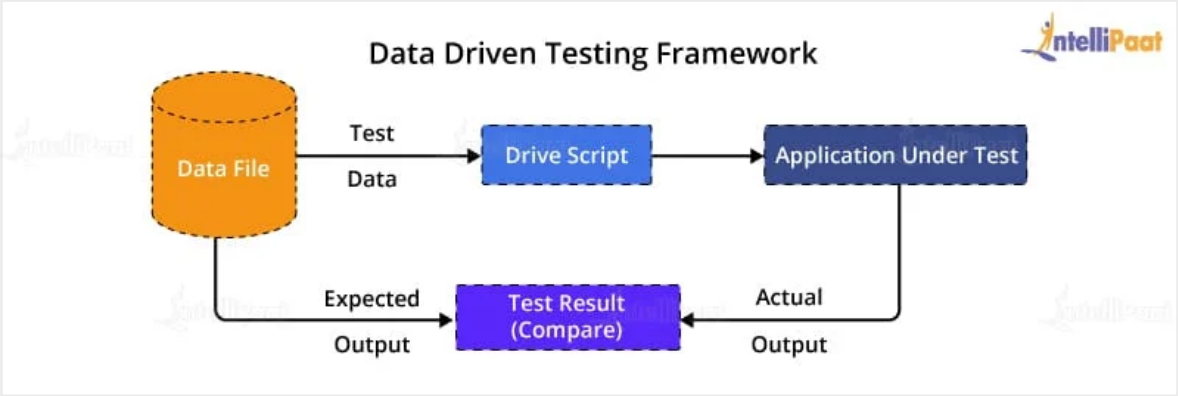
Four parameters that need to be passed in Selenium are:

- Host
- Port number
- Browser
- URL

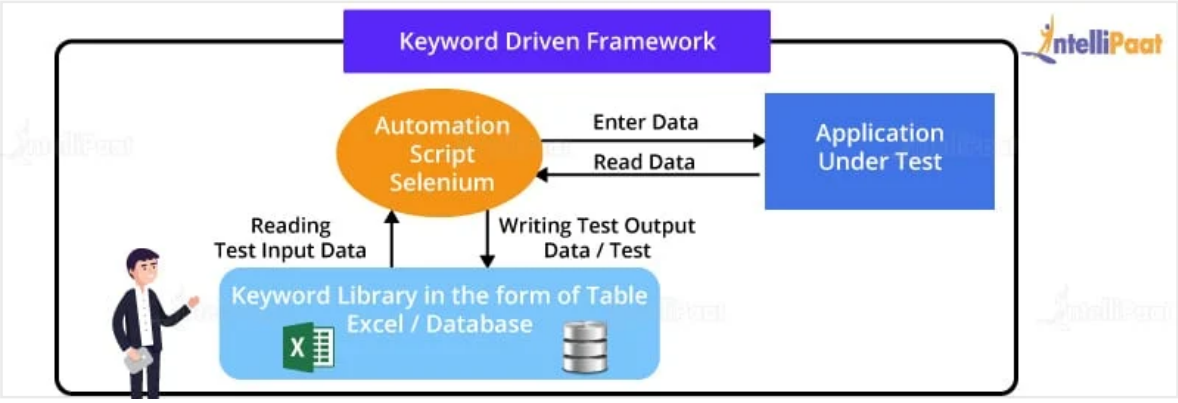
What are data-driven framework and keyword-driven framework?

A data-driven framework in Selenium is an approach of separating a 'dataset'

from the actual 'test case' (code). This framework is completely dependent on the input test data. The test data is inserted from external sources, such as from an Excel file, a CSV file, or from any database. It also allows us to easily control how much data needs to be tested. We can easily increase the number of test parameters by adding more username and password fields to the Excel file (or other sources).



A keyword-driven framework is an extension to the data-driven testing framework in the sense that it not only isolates the test data from the scripts but also keeps the particular section of the code belonging to the test script in an external data file. These sets of code are known as keywords, and hence the framework is so named. Keywords are self-guiding and work based on what actions need to be performed on the application.



What is the difference between getwindowhandles() and getwindowhandle()?

- **getwindowhandles():** It is used to get the address of all open browsers, and its return data type is Set<String>.
- **getwindowhandle()**: It is used to get the address of the current browser where the control is, and its return type is a string data type.

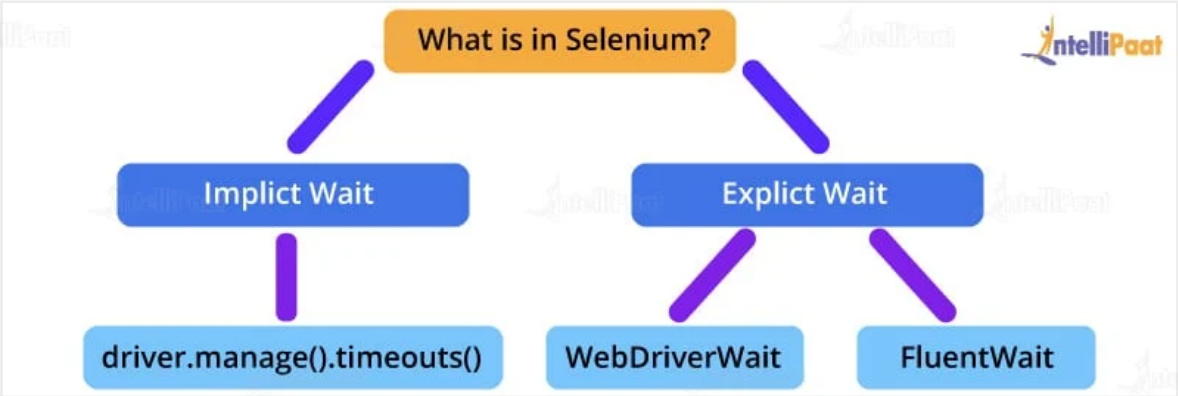
What is the difference between setSpeed() and sleep() methods?

Both setSpeed() and Sleep() in Selenium are used to delay the speed of execution.

- **setSpeed**: Sets the execution speed with a delay of milliseconds, followed by the Selenium operation. By default, the delay is 0 milliseconds.
- **sleep**: Causes the suspension of execution of the current thread for a specified period.

What are the different types of waits available in WebDriver?

There are two types of waits available in WebDriver:



- **Implicit wait**: These waits are used to provide a default waiting time (say, 30 seconds) between the consecutive test steps across the entire test script. Hence, the subsequent test step would only be executed when the 30 seconds are over after executing the previous test step.
- **Explicit wait**: These waits are used to halt the execution until a particular condition is met or the maximum time has elapsed.

Explicit waits are instantiated for a particular instance only, whereas implicit waits are not.

In Selenium, what are breakpoints and start points?

Breakpoints: Breakpoints are used to stall the execution of the test. The execution will stop whenever a breakpoint is implemented, and this will help us check whether the code is working properly or not.
Start points: Start points are the points from where the execution should begin. Start points can be used when we want to run the test script from the middle of the code or after a breakpoint.

How do you find broken links in Selenium WebDriver?

We can detect whether the given links are broken or not by using the following process:

1. First, accumulate all the links present on a web page using the <a> anchor

tag. For each <a> tag, use the attribute 'href' value to obtain the hyperlink
2. Send HTTP requests for each link and verify the HTTP response code
3. Based on the HTTP response code, determine if the link is valid or broken. Then, use the driver.get() method to navigate to a URL, which will respond with a status of 200 – OK (200 – OK indicates that the link is working). If we get any other status, then it indicates that the link is broken
4. Repeat the same process for all the links captured

StaleElementReferenceException
This exception occurs when the web element gets detached from the current DOM.
Root Cause:
JavaScript or JS library has deleted an element and replaced it with one with the same ID or attributes.

org.openqa.selenium.StaleElementReferenceException

This exception says that a web element is no longer present in the web page. This error is not the same as ElementNotVisibleException.
StaleElementReferenceException is thrown when an object for a particular web element was created in the program without any problem and however; this element is no longer present in the window. This can happen if there was a

- **Navigation to another page**
- **DOM has refreshed**
- **A frame or window switch**

WebElement firstName = driver.findElement(By.id("firstname"));
driver.switchTo().window(Child_Window);
element.sendKeys("Aaron");
In the code above, object firstName was created and then the window was switched. Then, WebDriver tries to type 'Aaron' in the form field. In this case StaleElementReferenceException is thrown.
Avoiding and Handling*: Confirm that we are trying to do the action in the correct window. To avoid issues due to DOM refresh, we can use Dynamic Xpath*
Let's discuss another example.
Say 'id' of a username field is 'username_1' and the XPath will be *// *[@id='firstname_1?]*. When you open the page again the 'id' might change say to *''firstname* _11'. In this case, the test will fail because the WebDriver could not find the element. In this case, StaleElementReferenceException will be thrown.
In this case, we can use a dynamic xpath like,

```
try {
driver.findElement(By.xpath("//
*[contains(@id,firstname')]")).sendKeys("Aaron");
} catch (StaleElementReferenceException e)
```

In the example above dynamic XPATH is used and if the exception is still found,

it is caught.

Cookies In Selenium:

```
 driver.manage().addCookie(new Cookie("foo", "bar"));
          // Get cookie details with named cookie 'foo'
          Cookie cookie1 = driver.manage().getCookieNamed("foo");

Set<Cookie> cookies = driver.manage().getCookies();
          System.out.println(cookies);

driver.manage().deleteCookieNamed("test1");

          /*
           Selenium Java bindings also provides a way to delete
           cookie by passing cookie object of current browsing context
           */
          driver.manage().deleteCookie(cookie1);

driver.manage().deleteAllCookies();
```