In today's software sector, automation testing is a full-fledged career. Automation testing professionals collaborate with the development team to meet project deadlines, which has resulted in a number of benefits and a lucrative career for modern-day professionals. Here are the 51+ Automation Testing Interview Questions you should know before your next interview.

Automation Testing Interview Questions:

The following are some of the most often asked questions in interviews for both freshers and QA professionals.

1) What is Automation?

Automation is a word for applications where reduce human efforts. There are include business process automation (BPA), IT automation, personal applications, business-related applications, E-commerce applications such as home automation, and more.

2) What is Automation Testing?

Automation testing is a process of using automation software tools to write a script to perform test cases such as entering data and comparing the results, etc. is known as Automation   testing.
The goal of Automation Testing is to reduce the number of test cases to be run manually so we can reduce human efforts

3) How do you identify the test cases which are suitable for test automation?

Identifying the particular test cases for automation is the most important part of automation. If the test case is stable then it is beat for automation.

4) What are the types of automation testing?

There are different types of automation testing. Like exploratory testing and some other that you can automate

Unit tests: The unit test is written by developers and test small units of code in isolation.

Integration tests: After performed Unit testing, all module is merged and then we are performing integration testing to verify how all component work combine together.

Regression tests: In regression testing, we are testing like updated code didn't break any existing working functionality.

Performance tests: Here verify that the application won't crash under heavy load and stress.

UI tests: Test there is all text and images are displayed properly, and there are no broken graphical elements on the page.

5) What is the difference between automation testing and manual testing?

Manual Testing	Automation Testing
A human executes the test cases one by one his/her self, without any software help it is called manual testing	Test cases are executed by any automation testing tool, without human efforts It is automated testing.
Useful for non-repeatable tests that involve human skills, involvement, and domain experience.	Useful for repeatable test cases where the application feature under test doesn't change frequently.
Good for accessibility and usability testing, as the tester can test the software as end-user point of view.	Good for regression testing to make sure that the software didn't break after updating code changes.
Can be slow and more time-consuming, and sometimes getting errors because of human misjudgment.	It's run by a computer, so there are no chances for error, and test case execution is fast.
Humans can test the software in a randomized manner, and it is also known as exploratory testing.	This type of testing is not possible in automation testing.
UI issues and some variances are easily found by a human tester.	In automation testing, we cannot cover UI issues.

performance Testing is impossible to test with extreme load by a human tester	Performance testing can be easily done by automation testing.
The tester doesn't need knowledge of programming.	To write automation test cases, the tester needs to have programming knowledge.

6) When is a good time to automate a test?

Under the following condition, it is a good time to automate a test

- The test cases are repeatable.
- The test cases under the automation testing don't change their functionality again and again.
- Some test cases are more time-consuming for a human tester.
- Some test cases are connected with complicated calculations.
- The test ensures the existing functionality didn't break after an update of the code change.

7) When will you avoid automated testing?

In the following situation, we will avoid automation testing

- When the Application Under Test and changes come regularly
- One time test cases
- Adhoc testing
- Automation testing is not useful for exploratory testing because a human tester can more explore the application with much better scenarios than a computer.
- Automation testing is also not used in UI testing

8) How do you choose a tool/framework for automated testing?

To perform any automation testing, we need to choose software tools or frameworks. There are many options to choose from many alternatives.

Here some are criteria based on that we can evaluate the tools.

- **Programmable (code-based) or code-less tools.**

Some tools require knowledge of programming, while some tools don't require, without any skill of programming allow to create test cases with visual assistance. so, in that situation, you can choose your tool based on your team's experience and skill-set.

- **Commercial vs. Open Source.**

There is a more difference in the price of the tools it is based on the feature which they are provided. Most commercial tools are more expensive, but they get tech support if you are facing any issue.

Open-Source software is free, but you have to search if you are in trouble.

- **Ease of use.**

Some Automation testing tools are difficult to using and required training before using this tool. Some are easy to understand we can use without any training.

Some of the most famous automation tools like Katalon Studio, Selenium, UFT, TestComplete, Testim, etc., and many more. You can choose anyone based on the requirement of your project, assess the team's skills, the experience of all team members, and comfort with the tool.

9) What are the different parts of a test automation framework?

Below is the different part of automation framework

Test Data Management

Generating test data is the main problem of automation testing. A good test case automation framework is making it easy to build test data for the application at the testing time

Testing Libraries

A Good automation framework provides libraries so test management make easy and also support different types of testing like integration testing, unit testing, and end to end testing

Testing Tools

1. Includes any GUI or command-line tools that make it useful for testers to run the set of tests again and again.
2. Testing tools are also useful for a tester to perform load and stress testing

10) Should you automate all testing?

We can not automate all types of testing. Some types of testing are only done by human testers like interface testing, usability testing, and accessibility testing.

Exploratory testing is another type of testing where a human tester can test better than automation testing. A human can provide more scenarios and tester more explore the software randomly and tried to find UI bugs easily as compare to automation testing.

Automation testing is useful for large projects where complex calculations and repeated test cases are available.

11) What is a test environment?

A test environment is a computer or a server on which a tester can test the software or applications. After the developer team builds the software, the tester installs it on this computer or on a server with all its required dependencies, like the production environment. This allows the tester to test the software or application with the real-world scenario.

Actually, the test environment is a copy of the production or live environment.

These allow the tester can make a duplicate copy to reproduce the same bug for reporting and the test can provide the exact steps to a developer to fix them.

Below are the some of prerequisites for a good test environment:

- A test server has a similar configuration, including software and hardware to match with the production environment
- Sample test data for testing with software.
- Test database with real data, it can be a copy of the production database.

12) What is browser automation?

Browser automation is one of the techniques for launching a web application in a browser programmatically and executing different actions by automation, just like a regular user would. Browser testing gives you the speed and efficiency that human testers cannot provide. There is a different popular tool for browser testing like Protractor, Cypress, and Selenium.

13) What is automated regression testing?

Software is never completed. The developers are continuedly updating features, functions and fixing bugs So because of that chance to break existing working functionality. And because of that tester need to perform regression testing.

Regression testing is a testing technique where makes sure that updated code is not affected by the existing working functionality and didn't break any working functionality. The main foal of regression testing is to make sure that previously tested and working functionality still works fine after adding new changes in code.

In automation regression testing, the tester runs the regression suit after every new release of the software. If a test case is a pass then the tester can move to the other type of testing and if a test case is failed then wait until the developer fixes this broken functionality.

14) What are some risks associated with automated testing?

There is more benefit of automation testing like efficient and fast, repeatable tests and also there are few risks with automation testing which are described below:

Negative ROI

- The automated test requires more code and expensive tools. Testers need to spend more time learning and implementing automated tests.
- As per the experience, the team realizes that the testing strategy, while the software is complex with continually changing features. Tester also has to keep changes continually.

Playing catch-up with the technology.

Before start the Automation testing, the testing team spend more time learning and try to code with different tools and technologies because experience is more required in automation testing and then realize its original promised value

Maintenance Risk

- Testers need to maintain code with all changing functionality and fixing the bugs.
- Instead of creating new features most of the time tester working on that test case which is failed

15) What are the different phases in an automation testing life cycle?

Automation testing has a life cycle same as the software testing life cycle and software development life cycle. Some of the major phases of the automation testing life cycle are described below:

1. Analysis of the scope of the testing.
2. Choose the automation framework and tools for the start automation.
3. Design test plan and test case execution strategy.
4. Set up test environment as per requirement.
5. Start development and execution of the test cases.
6. Generate test report and verify it.

16) Can you achieve 100% automation?

Achieve 100% automation is very difficult because there would be many different scenarios under the test cases. And more test cases are not executed for a long time. And will not add value to these test cases into an automated suit.

17) How to decide the tool that you should use for Automation testing in your projects?

1. You have to understand the project requirements and identify the test scenario which you want to automate.
2. Found best tools that support your project requirement.
3. Decide your budget for automation tools and as per that select tools under your budget.
4. Identify skilled resources if you have If you don't have then need to identify the cost of training of your existing resources or hiring new resource if resources are not available.

18) What is a framework?

A framework is a bunch of the design of the whole automation suite. It is additionally a rule, which whenever followed can bring about construction that is not difficult to keep up with and improve.

The Rules Include:

- Coding principles.
- Taking care of the test information
- Managing and taking care of the components of the framework
- Managing of environment documents and properties record
- Reporting of information
- Dealing with logs

19) What are the attributes of a good framework?

Read in the brief about attributes below:

- **Modular:** The structure ought to be versatile to change. A Person who is analyzing and testing should have the option to adjust the contents according to the climate or login data change.
- **Reusable:** The regularly utilized technique or utilities ought to be written in a standard file that is available to every one of the contents.
- **Consistent:** The suite ought to be written in a steady way by following all the acknowledged coding rehearses.
- **Independent:** The contents ought to be written so that they are free of one another.
- **Logger:** It is a great idea to have executed the logging highlight in the structure. This would help in the event that our contents run for longer hours (say daily

mode), if the content fails anytime, having the log document will assist us with identifying the area alongside the sort of defect.

- **Reporting:** It is a great idea to have the detailing highlight consequently inserted into the system. When the code run, we can have the outcomes and reports sent through email.

20) What are the various types of automation tools you are familiar with?

There are two types of automation tools available in the market which are as follow:

- **Open Source**: It includes Selenium, JMeter, etc.
- **Paid Tools**: It includes QTP, Load Runner, Ranorex, etc.

21) What are the different types of frameworks?

Different types of automation frameworks are as follow:

- Keyword-driven structure
- Data-driven structure
- Hybrid structure
- Linear Structure

22) Can you give some examples of effective coding methods for automation?

Good practices of coding include:

- Add Proper Comments
- Distinguish the reusable techniques and add them in a different file.
- Follow the language-particular coding protocol.
- Keep up with the test information in a different record.
- Run your contents routinely

23) Is there any type of test that you think should not be automated?

- Tests that are only rarely executed.
- Exploratory testing
- Test which is executed immediately when perform manually.

24) What criteria do you use to determine which automation tool is best for you?

Below factors help in selecting the best automation tools:

- The range of the application which we need to automate.
- The board overhead like expense and financial plan.
- Time to learn and execute the apparatus.
- Sort of help accessible for the device.
- Limitation of the tool.

25) What do you think is preventing testers from automation testing? Is there a way to overcome it?

The significant obstacle for analyser's is to study programming/coding when they need to automate. Since analyser's don't code, adjusting to coding is a bit hard for analyser's.

Way to overcome it are as follow:

- Cooperating with designers while automating.
- Taking into account that automation is the obligation of the entire group and not just of the analyser's.
- Giving a devoted time and priority to automation.

26) What is an Automation testing framework?

An automation testing structure is liable for making a test utilization with an instrument to interface with the application under test, take input from a document, execute the experiments and create the reports for test execution. An automation testing system ought to be autonomous of the application and it ought to be not difficult to utilize, alter or expand.

27) What are the key components of an automated testing framework?

Below are important modules of the automation testing framework:

- **Test Assertion Tools:** This device will give a verification guarantee to testing the normal qualities in the application under test. For Example. TestNG, Junit, and so forth.

- **Data Setup:** Each experiment needs to take the client information either from the data set or from a record. So this module ought to take care of the data that is utilizing in the script.
- **Build Management Tool:** Structure should be created and sent for the utilization of making test scripts.
- **Continuous Integration Tool:** This device is needed for coordinating and sending the progressions done in the structure at every cycle.
- **Reporting Tool:** This device is used to obtain end results.
- **Logging Tool:** This device helps in the event that our contents run for longer hours (say daily mode), if the content fails anytime, having the log document will assist us with identifying the area alongside the sort of defect.

28) Explain some Automation testing tools?

Below are the [popular automation testing tools](#):

- **Selenium:** Selenium is a test system for web application automation testing. It upholds different programs and is OS-free. Selenium likewise supports different programming dialects like Java, C#, PHP, Ruby, and Perl, and so forth.
- **UFT:** It gives a wide scope of elements like APIs, web administrations, and so on and furthermore supports numerous stages like work areas, web, and portable. UFT scripts are written in the visual fundamental prearranging language.
- **Appium:** Appium is an open-source versatile application testing apparatus. It is utilized to test cross-stage, hybrid, native, and online portable applications.
- **Cucumber:** Cucumber is an open-source conduct-driven improvement device. It is utilized for electronic application automation testing and supports dialects like ruby, java, scala so on.

29) What are the various types of framework testing techniques?

Below are the different types of techniques:

- Modular Testing framework
- Data-Driven Testing framework
- Keyword-Driven Framework
- Hybrid Testing Framework

30) When should you use manual testing instead of automated testing?

Manual testing can be preferred over automation testing due to the following reasons:

- The task belongs to a short time period and composing content will take a lot of time to run and is expensive when contrasted with manual testing.

- Adaptability is required.
- Ease of use testing should be performed.
- Applications/module is recently evolved and has no past experiments.

31) List some advantages and disadvantages of Automation testing?

Pros:

Less HR

More Test Coverage significantly quicker

Dependability

Equal execution of tests.

Cons:

- Building and preserving time is more
- High Device Cost
- Knowledge of Code and skill required
- Environment arrangement

32) How will you automate test cases for an application's fundamental "login" functionality?

We need to follow the list of rules for writing test cases:

whenever we write test cases we should remember it should be a combination of both negative and positive test cases. Also, it's a mixture of functional, performance, UI, usability, and compatibility test cases.

Testcase for basic login functionality is as follows:

Functional test case:

Sr.No	Functional Test Cases	Type
1	verify that the user has been successfully Login to the application with valid credentials.	Positive

2	verify that the user unable to login application with an invalid credential.	Negative
3	verify login page without entering any details.	Negative
4	verify that when the user enters a password it should be in encrypted form.	Positive
5	verify time taken to Login application with valid credentials.	Performance & Positive
6	verify the login page and all its control in different browsers is working fine.	Cross Browser
7	verify the length of email id/user id.	Positive
8	verify the length of the password and whether it accepts special characters or not.	Negative
9	verify the implementation of the SSL certificate.	Positive

33) What do you think, Automation testing is White-box testing or Black-box testing?

Automation testing can be both Black box or white box sort of testing relying upon the situations wherein it is performed. In some situation test scripts needs to access information base i.e. Database specification that is utilized in the application under test or some additional coding specification and accordingly can be a sort of white-box testing.

34) How many test cases per day do you automate?

One cannot estimate the number of test cases you can automate at a particular time, the number may depending upon complexity, requirement, and time given for testing the application.

the estimate can depend on the Test case scenario's complexity and length. I did automate 3-6 test scenarios per day when the complexity is limited. Sometimes just 1 -2 test scenarios in a day when the complexity is high.

35) What factors determine the effectiveness of Automation testing?

Having a talented staff that includes professionals along with the newest and smartest graduates or fresher can help us determine successful completion of testing. We can measure the effectiveness of automation testing in relation to risk and then differentiate risk high to low priority and start the automation of high priority risk, Risk with high priority can be automated first also we can track the high priority risk that has been mitigated by the automated tests.

36) Which test cases can be automated?

A test case should be automated if:
The task is going to be repeated.

- It's going to save time.
- The requirements, the test, or the task are a low hazard, constant, and unlikely to change often.
- The test is subject to human error.
- The test is time-consuming.
- The test has significant downtime between steps.
- The test is repetitive.

37) Do you know the advantages and disadvantages of Automated testing?

Automation testing is a process of automating the manual testing process by a set of tools. This process is done by replacing the manual testing process with the collection of automated testing tools. Automation testing ensures the quality of software applications.

Automation Testing has the following advantages:

- Automated testing improves the coverage of testing as automated execution of test cases is faster than manual execution thus it saves resources and time as well.

- There is less dependability of testing on the availability of tester once the software is automated.
- It provides more Generalization as automated tests can be run all-time in 24*7 environments.
- It takes fewer resources as compared to manual testing.
- Automation testing provides more reliability of testing like stress testing, load, and performance testing which is not more reliable than manual testing.
- With the help of automation testing, we can produce maximum test data to cover a large number of input and check the expected output also there are fewer chances of getting errors while using automation testing.

Automation Testing has the following disadvantages:

- Automation testing is more expensive as compared to manual testing so most of the organization will choose manual testing as compared to automation testing.
- Automation testing needs more domain knowledge and needs more professional and skilled people so they can automate applications more effectively.
- Sometimes, it's inconvenient to decide what to automate first and who will automate.

38) What is an Automated test script?

Test Scripts are line-by-line descriptions that contain information about the system transactions that should be performed to validate the application or system under test. The test script is a combination of different test steps that should be followed to test the functionality.

How to write a test script?

1. Record/playback:

In this method, the tester needs to write code instead of just recording the user's actions. However, if we find some issue then the tester will need to do some coding to fix that and check it works as per expectation.
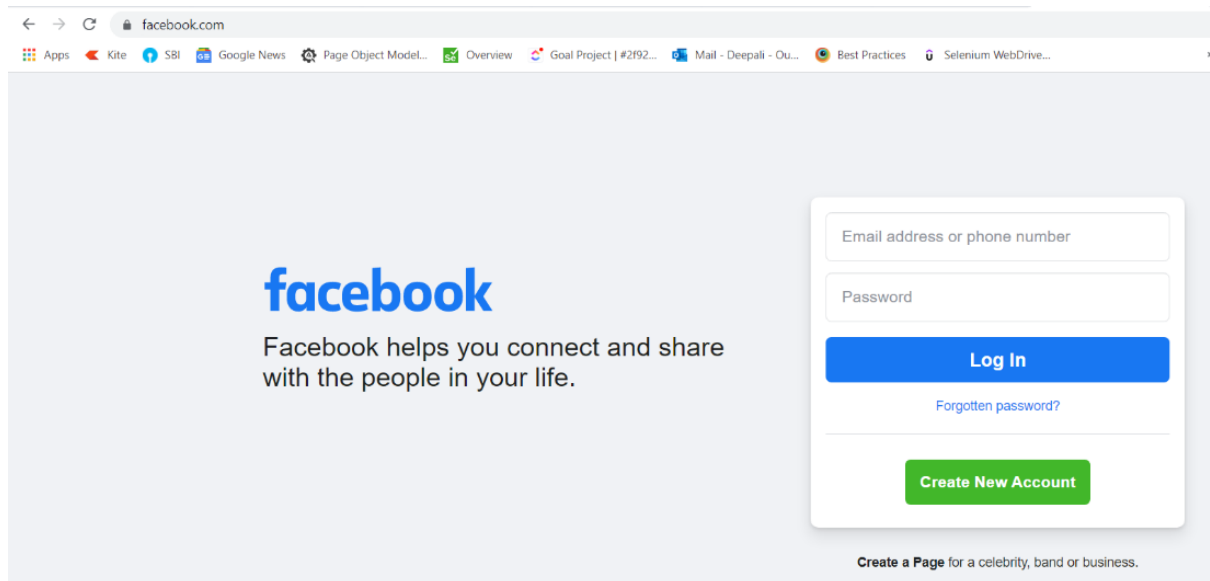
2. Keyword/data-driven scripting:

In this method, there is a dissociation between the testing team and the development team. In the Data-driven scripting technique, we keep data in different repositories.

Example of a Test script:
Log in Script

Identify the email and password locator to perform the login functionality by their X-path or CSS element and ensure that

- username and password fields should be visible.
- Next, type the username and password.
- Make sure the user enters a valid email and password.
- Identify the Login locator and click on the login button, and verify the login screen.
- Next, type the username and password.
- verify after clicking on the login button the homepage should load properly.

39) What is an XPath?

In this method, there is a dissociation between the testing team and the development team. In the Data-driven scripting technique, we keep data in different repositories.

X-path: X-path is used to identify a locator in a webpage, firstly we tried to find the element using id, by CSS selector, etc, if the element is not found using this we used x-path to find an element on a webpage.

The basic format of x-path:
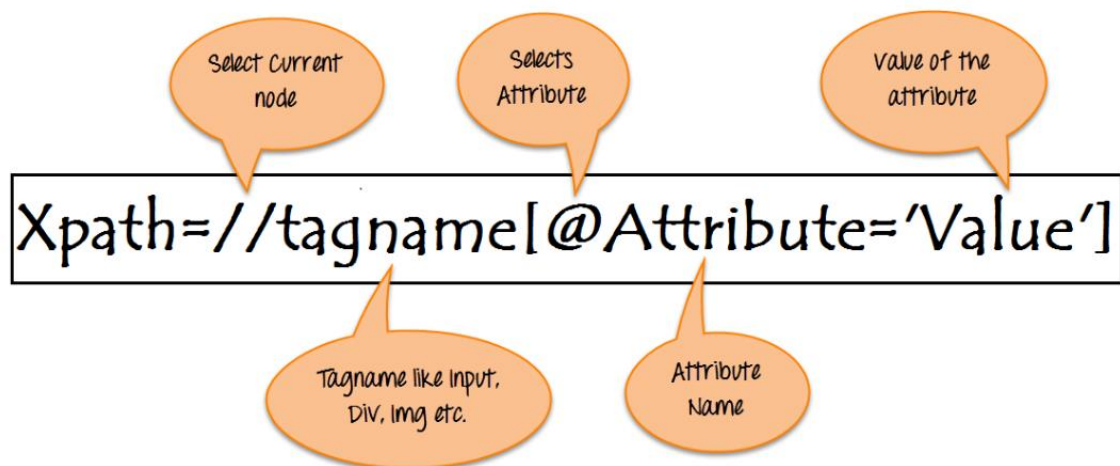
Syntax: //tagname[@Attribute='value']

- **// :** "//" is used to Select current node.
- **Tagname:** Tag name of the particular node.
- **@:** select attribute
- **Attribute:** attribute name of the node.
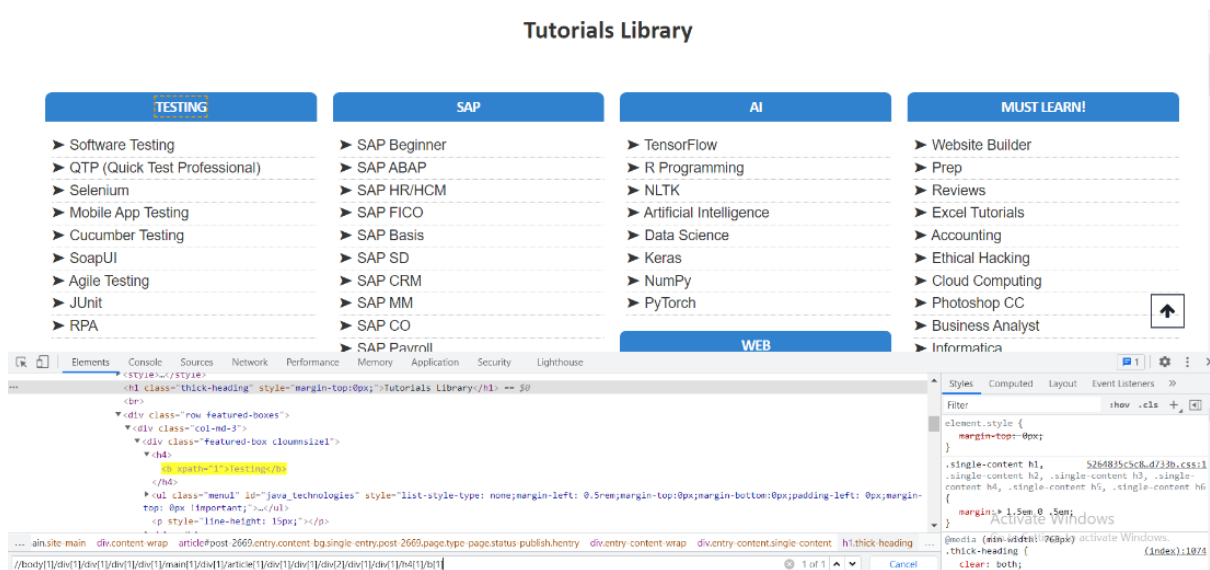- **Value:** value attribute contains.

Type Of X-Path:

1. Absolute X-path

2. Relative X-path

1. Absolute X-path: It's the direct way to find the element, but the disadvantage of the absolute XPath is that if there are any changes made in the path of the element then that XPath gets failed.
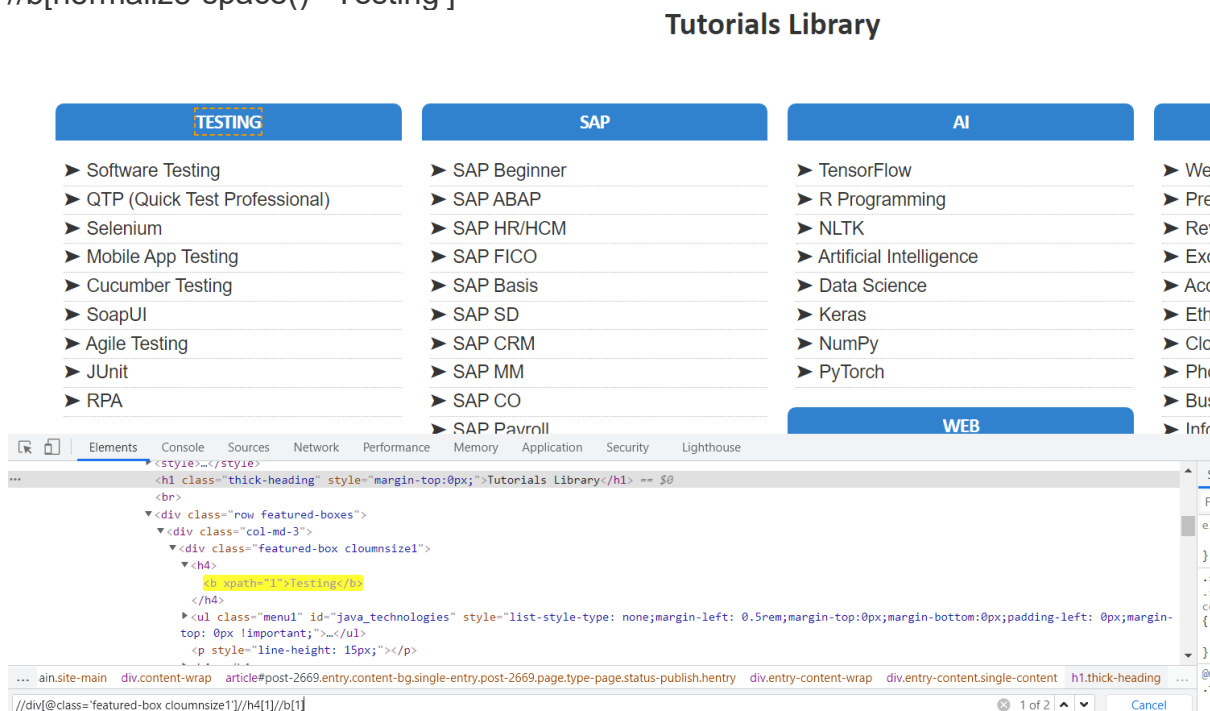
Example:

//body[1]/div[1]/div[1]/div[1]/div[1]/main[1]/div[1]/article[1]/div[1]/div[1]/div[2]/div[1]/div[1]/h4[1]/b[1]

2. Relative X-path: It starts with a double forward-slash (//). we can search elements anywhere on the webpage, using relative XPath.

**Example:**

//b[normalize-space()='Testing']

# 40) Do you know, what is the difference between "/" and "//"?

Single slash and double slash are always used in a path expression, by using this symbol we usually find elements or attributes in an XML tree. A path manifestation

consists of one or more steps that direct the XPath processor along one or more subdivide of the tree to the nodes that will be selected.

Usually, single slash and double slash are generally used to create XPath by which we can find any element/locator on a webpage. single slash is used in absolute XPath which selects an element from the root node.
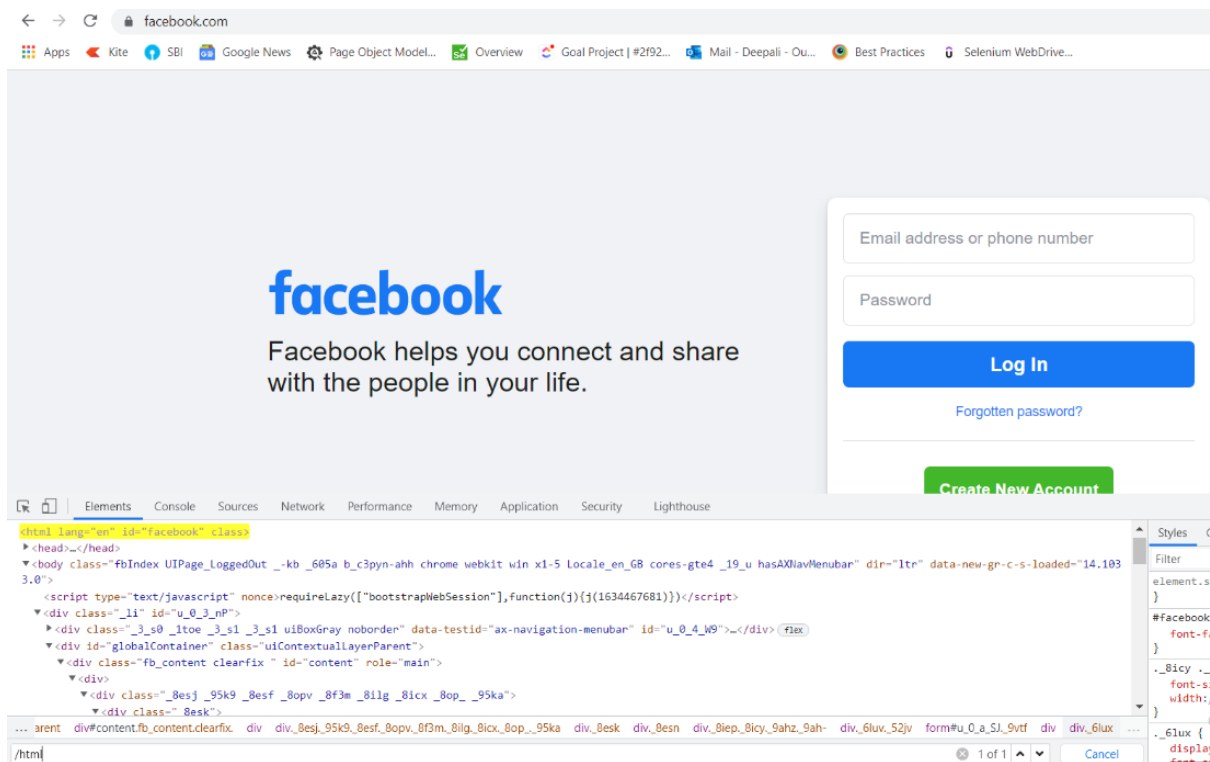
double slash is used to find out relative XPath which finds an element from anywhere on the webpage.
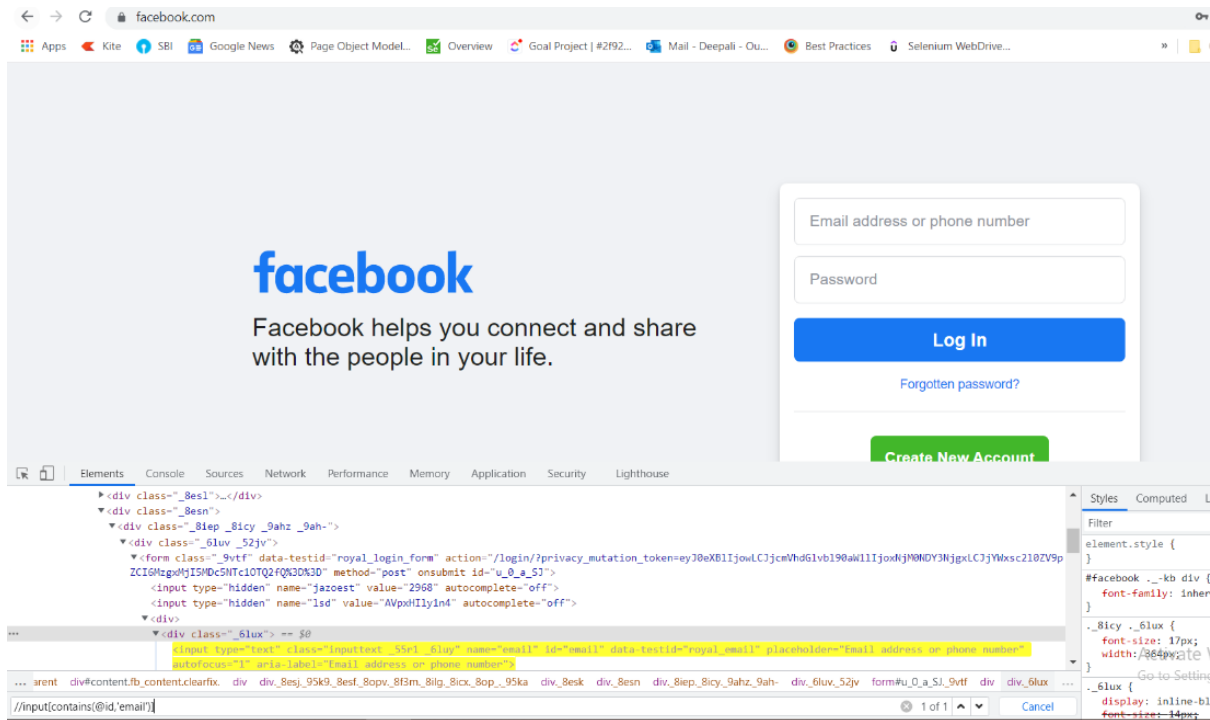
41) Explain Tree View in Automation Testing?

A tree view in Quick Test Pro has statements in" keywords "or we can say statements in a "tree structure". In a tree view, the statements are in a tree structure. Here, the statements are nothing but steps that were taken to test some features or modules of an application to determine it works as per user requirements.

42) What Features will you look out for in an Automation Tool?

- Test Environment support
- easy to use
- debugging ability
- Robust and secure
- Image testing abilities
- Identification of object
- Support database testing
- Support multiple frameworks

43) Name the different Scripting Techniques for Automation Testing?

Automation scripting technique defines different parameters like how to treat the object repository, the coding standards followed during automation, and data handling, among different modules, the automation scripting we should follow for test automation are:

1] Linear Scripting Technique:

This technique is the simplest of all the scripting techniques which follow a simple 'Record n Playback approach, where each test case is executed manually step by step so, we have the QA team record every step which is executed.

Pros

- Very low development cost
- Requires less programming knowledge
- No test planning is required for testing

Cons

- Maintenance cost is very high
- Non-structured way of programming

2]   Structured   Scripting   Technique
The structured scripting technique involves the use of control structures(like if-else, switch case, for, while conditions) in the scripts that help QA teams to regulate the flow of the test scripts.

Pros

- add some control statement to scripting
- Easy to maintain scripts
- reusability of scripts

Cons

- high cost
- data is hardcoded
- less secure

3] Data-Driven Scripting Technique

The Data-driven scripting technique is all about separating data and script in different places. this method keeps data into an external repository as files.

Pros

- Different scenarios can be easily executed
- Modifications in test script do not affect Test Data

Cons

- time-consuming

4] Shared Scripting Technique
In shared scripting system of working is different from the other scripting techniques it involves sharing scripts that describe an application Deportment. we can use this script if we have common functionality.

Pros

- Structured scripts
- Low cost

Cons

- Requires programming knowledge
- Requires planning

44) Can automation testing completely replace manual testing?

Automation testing has received much response as compared to manual testing nowadays. In today's world, many developers and testers prefer automation testing to make their life easier. One of the primary reasons automated testings becomes popular nowadays is the ability to save time, However, test automation does not have the ability to fully replace manual testing, the reason behind this is following:

Usability Testing is one of the major challenges in automation testing, usability tests are just not possible with automation testing because usability testing requires a human to check the ease of use while using the application. we can't train a computer to identify usability respective to humans. also, we cannot automate captcha in automation if we are going to think to skip captcha we cannot proceed further we need humans to proceed further. These are some limitations in automation testing.

The presence of automation testing and manual testing forces us to think about our choice, of tools, number of resources, their cost, and the benefit provided by different tools. Is automated testing replacing manual testing the answer is No, Manual testing helps us understand the entire problem and explore new things. Automated testing helps us save time.

45) What is your strategy for automating web application testing?

6-step that should be considered every time when we are automating a web application are:

1. Functional testing: The first step which we should consider while automating web application is functional testing, functional testing ensure that the functionality of web application is working properly and it returns expected output.

Typically, functional testing includes:

- the identification of functionality and output
- data input and entry
- the execution of the test case
- Analysis of the actual and expected result

2. Usability Testing: Usability testing refers to the ease of use, usability is a way to see how easy to use something in terms of user perspective. Usability testing determines how easily a user can access the application and work on it, it's being observed by a researcher, to see where user encounter problems and experience confusion, if more user encounters the same issue, recommendations will be made to overcome with this issue.

3. Interface Testing: Interface Testing is defined as a software testing type that ensures there is proper infusion between two different software systems or applications. A connection that integrates two components is known as an interface. Interface in a computer world could be anything like API's or it can be a web service. Testing of these connecting services is referred to as Interface Testing. An interface is software that consists of sets of commands or it can be other attributes that enable infusion between a user and application or device.

4. Compatibility Testing: Compatibility testing ensures that our software application is compatible with different browsers and in different devices and applications working fine in different platforms. There are different elements for compatibility testing:

1. Browser compatibility: Browser compatibility makes sure that our application is functioning correctly across the different browsers on different devices. This includes checking that JavaScript, AJAX, WebSocket's, browser notifications, and authentication requests are working as designed.
2. Operating system compatibility: Operating system compatibility ensures that our web application runs in different operating systems and it works fine as expected. it checks that it runs smoothly on Windows, macOS, Linux, and UNIX.
3. Mobile compatibility: it Ensuring that our application runs on different devices and functions as expected in different Android, iOS platforms.

5. Performance Testing: After ensuring that our applications are working as expected on different platforms, it's time to check the performance of web applications. In performance testing, we test our web application under heavy load and stress and check how our application reacts to heavy load and determine the breaking point of our application.

6. Security Testing: Security testing makes sure that our application is secure against unauthorized access and harmful actions through viruses or other malicious software.

Security testing has the following activities:

- It makes sure that secure pages, data can be accessed after a license.
- Verify that the web applications' follow proper SSL.
- Make sure that restricted files cannot be downloaded without a proper license.

46) What is your strategy for automating web application testing?

- **Prevention Rather Than Detection:** we should have prevented bugs before it introduces in our application. we should more focus on preventing the introduction of defects in the application first.
- **Regression Testing:** it is the heart of the test automation strategy. regression testing ensures that after some fixes our application is working fine and it does not affect the already working functionality.
- **Smoke Test:** the idea behind the smoke test pack is to catch the most obvious issues, it makes sure that our application loads properly and common user flow is executed properly, smoke tests are quick and cannot be more than 5 minutes which use to check our application works as expected and something major is not working. The smoke test pack runs after every deployment and can be a mixture of API and GUI tests.
- **Application Testing:** An e-commerce application can be split into different applications or apps that provide different functionalities. The concept of Application Testing is a group of tests that use to test the functionality of an app are organized together and run against the desired app/application. This pack will be useful when a team wants to release an individual app and would like to know whether it is functioning correctly.

the methodologies allow quick detection of bugs when they are introduced in the system and results in feedback for development.

47) What are some good coding principles to keep in mind when automating?

Do not rely ONLY on UI test automation try to test load, performance, and usability testing of the application.

- Consider using a BDD framework, which is very helpful
- Always use test design patterns and principles.
- Make your test automation framework portable.
- Name your tests wisely.
- Divide tasks based on skill
- Remove uncertainty
- Pick the right tool

- Test on real devices
- Keep Records for Better Debugging
- Use Data-Driven Tests
- Early and Frequent Testing
- Prioritize Detailed & Quality Test Reporting.

48) What do you do now that you've found your automation test tool?

After identifying the automation testing tool the next step should be a resource and test planning, after that, we can look for what reporting tool we can use in our automation. A better test planning results in the early execution of the test which will result in a high-quality product.

49) What is the main reason for testers to refrain from automation? How can they overcome it?

the main reason for testers to refrain from automation is to learn programming/coding when they want to automate software applications, as we all know that testers have less knowledge of programming/coding, so this will become challenging for the tester.

We can overcome it by:

- Collaborating with the development team when automating.
- Considering that automating software applications is the responsibility of the entire team and not only of the testers.
- Giving a dedicated team, sufficient time and focus on automation.
- Getting proper management support from the team and supporting each other.

50) Name important modules of the automation testing Framework

- Linear Automation Framework.
- Modular Based Testing Framework.
- Library Architecture Testing Framework.
- Data-Driven Framework.
- Keyword-Driven Framework.
- Hybrid Testing Framework.

51) What is Captcha?

this is a process where the user has to be asked to perform a certain operation such as enter the text written in the image or click the same image from a set of the different images that match the required criteria, or click on the checkbox.

captcha is usually used to check whether the user is a human or robot. since this task is dynamic, and it can only be completed by humans only with human cognizance and correct interpretation of information-areas where artificial intelligence of bots tends to fail.

52) Can we automate CAPTCHA or RECAPTCHA?

CAPTCHA can be automated if you are able to decode the image using OCR, OCR stands for Optical Character Recognition. This will be the only solution to bypass CAPTCHA but it will not automate it. Users will always have to enter the CAPTCHA themselves while we can automate another text field.