Software testing interview questions pdf free download

As an employer and interviewer, it can be difficult to deal with good candidates from the less qualified. When you're interviewing, make sure you ask relevant questions, so your chosen candidate is not only professional and career-oriented, but also has goals and healthy interests outside the office. One of the first questions to ask as an employer is to lead you to learn more about who that person is. Ask the candidate to tell you about himself, his choice of education, his background and legacy. Each person has a different story, so ask to hear it. Ask the candidate why she chose this particular career or industry. For example, if a candidate is interviewed as legal secretary, ask about her interest in the right and interest in the position. You can easily identify from her answers if a candidate is pursuing the law because it is a passion or just an interview to get a job for money. Ask about the candidate's life goals. Goals can include jobs or career goals as well as personal goals. If a candidate's goal is to work effectively as part of a law firm team, you may have a good candidate. If, on the other hand, a candidate's goals include working from home or staying at dad's home, the candidate may not be the one you're looking for. While some employers want their employees to have a healthy lifestyle and hobbies outside of work, others don't care as long as the work is done. Candidates with wanted to talk about themselves, so they ask about their hobbies and interests outside of work. Use the answers to better understand the candidate. Ask a question concerning the choice of a candidate and the level of education. For example, if a candidate is for governor but has a degree in English literature, ask her how the education and skills she has learned will help her speak the position. The two issues that are common during the interview relate to the candidate's strengths and weaknesses. While a candidate can easily identify their strengths, weaknesses can be more of a problem because a candidate doesn't want weaknesses to take over and be the reason he hasn't received a job offer. Two more questions you should ask the candidate to deal with with previous experience. Ask the candidate about responsibilities or tasks in previous jobs. Then question her about the personal pleasure of the job. While the candidate may have been good at work, her answers will show if she doesn't enjoy the job. This can cause harm, especially if the candidate works directly with clients. The last question you have to ask a candidate is why you should hire him. This is the point of sale of the interview, as the candidate must explain why he believes he is qualified for the job. Most often ask automation testing interview questions for beginners and candidates at an expanded level: Automation testing plays a very important role in the entire software lifecycle. Most time when we want to for automation testing interviews, we focus only on specific issues of the tool. However, we must also consider the fact that learning and knowledge of the tool is only a means, and that is not the ultimate goal. So whenever we prepare to interview an automation tester, we should consider Automation in general and focus on the framework and steps involved. We all know that software testing is a very important part of software development. But, with rapidly growing methodologies and software development environments, it becomes difficult to manually check everything for use for a limited time along with cost constraints. Thus, automation testing is growing rapidly in the market to accelerate the pace of development. This tutorial includes leading interview questions on automation testing. I tried to cite short and quick questions that are very specific to automation in general and are not specific to any tool. Top 39 Automation Testing Interview QuestionsWe have looked at the main issues of automation and testing, as well as some extended questions for candidates of medium and expert level up to 2 to 5 years of experience. Q #1) What is automation? A: Automation is any action that can reduce human effort. Q #2) What is test automation? Answer: The process of using special software tools or scripts to perform test tasks such as entering data, performing test steps and comparing results, etc. is known as automation testing. Q #3) What can all be automated? Answer: Regression test kitSmoke / Sanity test suiteBuild deploymentSize the most data creationAuto-creation by GUI, like testing APIs and methods. Q #4) When is automation testing useful? Answer: Automation of testing is useful in the following cases:a) Regression testing: In case of bug fixes or implementation of a new module, we need to make sure that the functionality already implemented or unchanged is not affected. In this case, we end up running the regression test case multiple times. For example: After each change of request or correction of errors, after each iteration in case of a gradual approach to development, etc.) Dysfunctional testing: Testing of nonfunctional aspects of the program. For example, testing load or performance testing, etc. is very difficult for a person to track and analyze.c) Complex calculation checks or test scenarios that are prone to human errors.d) Re-performing the same tests: Sometimes we have to run the same set of test cases for another data set either after each build release or on multiple hardware, software or combinations of both. Automation of test cases in the above scenarios helps to achieve testing speed and minimize human errors. Q #5) How to identify test suitable for automation? Answer: Identifying appropriate test cases for automation is the most important step to automation. Q #6) Can you achieve 100% automation? Answer: 100% automation will be difficult to achieve there would be many edge test cases and some cases that are rare. Automation of these cases is not performed, which often do not add value to the automated package. Q #7) How to solve a tool to use to automate testing in your projects? A: To identify a tool to automate testing in your project:a) Carefully understand your project requirements and identify the testing scenarios you want to automate.b) Find a list of tools that support your project requirements.c) Define your budget for the automation tool. Select tools within budget.d) Determine whether you already have qualified resources for the tools. If you don't have the necessary skilled resources, then determine the cost of learning existing resources or hiring new resources.e) Now compare each tool for key criteria such as: How easy is it to develop and maintain scenarios for a tool? Can a non-technical person also perform test cases with little training? Does the tool support different types of platforms, such as web, mobile, desktop, etc. based on your project requirements? Does the tool have test reporting functionality? If not, is it easy to customize for the tool? As a tool for cross-browser support for web applications? How many different types of testing can this tool support? How many languages does the tool support?f) Once you have compared the tools, select the tool that is in your budget and supports your project requirements, and gives you more benefits based on the key criteria mentioned above. Q #8) Currently I have no automation in place in my project, but now I want to implement automation, what would be my steps? A: First, determine what type of testing/test cases you want to automate. Define the ToolDesign frameworkC create utility files and environment files. Start scripting Identify and work on reports. Target time to improve and support scripts. Steps needed to get automation testing for the project include: Understand the benefits and disadvantages of automation testing and identify test scenarios that are suitable for automation. Choose the automation tool that works best for automating defined scenariosDiscover the expert tool to help you customize the tool and the necessary environment to perform test cases with the tool. Train the command so they can write scripts in the programming language supported by the tool. Create a test base or identify an existing one that meets your requirements. Write a progress plan for OS, browsers, mobile devices, and more. Write programming scripts for manual test cases to turn them into automatic test cases. Report the status of a test case using the tool reporting feature. Save for current changes or new features. Q #9) How do you decide which tool you should use? Answer: Completing which tool works best for the project requires a lot of brainstorming and and #10) Once you identify the tool, what will be your next steps? A: Once we finalize the tool, our next step will be to develop a framework. Q #11) What is a framework? Answer: The framework is a set of structures of the entire set of automation. It is also a benchmark that if followed can lead to a structure that is easy to maintain and improve. These guidelines include: Encoding standardsReporting test dataSecretion and processing of elements (object repositoris in QTP)Processing environment files and file propertiesReporting dataRucoga logsQ #12) What are the attributes of a good framework? Answer: Characteristics include: Modular: Frames must be adapted to changes. Testers should be able to modify scripts as environmental information changes or logins. Reuse: Frequently used methods or utilities must be written to a generic file available for all scenarios. Sequential: The package must be written in a sequential format, following all accepted coding practices. Independent: Scripts must be written in such a way that they are independent of each other. In case one test fails, it should not deter the remaining test cases (unless it's an login page)Logger: It's good to have implemented the logging feature within. This will help in case our scripts run for longer hours (say, night mode) if the script fails at any given time, having a log file will help us determine the location along with the type of error. Reporting: It's good that the reporting feature is automatically embedded in the framework. Once the scripts are made, we can have results and reports sent by email. Integration: The Automation Framework should be such as integrating seamlessly with other applications, such as continuous integration or running an automated script as soon as the assembly is deployed. Q #13) Can you do without a framework? A: Frameworks are guidelines, not binding rules, so we can do without a framework, but if we create it and follow it, then expansion and support would be easy to implement. Q #14) What are the different types of automation tool that you know? Answer: An open source tool like Selenium, JMeter, etc. Paid tools such as QTP, Load Runner, Ranorex, RFT and Rational Robot.Q #15) What is usually the framework structure? A: Usually the structure should have - (It will be different from project to project)Src folder (source) that has actual test scripts. A lib (library) folder that has all libraries and common methods. The folder is a class that has the entire class file (in the case of java). The log folder has log files. A file/folder with all the Ids.A files of web elements that contain the URL, environment, and sign-in information. Q #16) Where you will store information such as URLs, Password? A: This information should always be stored in a separate file. Q #17) Why do you want to save such information in a separate file instead of in the code? Answer: URLs, logins and passwords are the fields that are used very often, and these changes are by environment and authorization. In case we hardcode it into our code, we have to change it in every file that has its own help. In case there are more than 100 files, it becomes very difficult to change all 100 files and this in turn can lead to errors. Thus, such information is stored in a separate file to make the update easy. Q #18) What are the different types of framework? Answer: Different types of framework includes: Keyword-based frameworkSubsid-based frameworkHybride FrameworkLinear ScriptingQ #19) Can you tell some good coding practices during automation? A: Some of the good coding practices include: Add relevant comments. Define ways to reuse and write them to a separate file. Follow language-specific encoding rules. Save test data in a separate file. Run scripts regularly. Q #20) Any kind of test that you think shouldn't be automated? Answer: Tests that are rarely performed. Intelligence testingPerformance testingThest that is performed quickly when done manually. Q #21) Do you think that testing can only be done at the UI level? A: Today, when we go into Agile mode, testing is not limited to the UI layer. Early reviews are imperial for a flexible project. If we concentrate only on the UI layer, we are actually waiting for the interface to be developed and available for testing. Rather, we can test even before the interface is actually developed. We can directly test APIs or methods using tools such as Cucumber and FitNesse.In so we give feedback much early and test even before the user interface is developed. By following this approach, we will help us check only the GUI aspect of small cosmetic changes or some user interface checks and help developers by giving more time to fix bugs. Q #22) How to choose which automation tool works best for you? A: Choosing an automation tool depends on a variety of factors, such as: The amount of program we want to automate. Managing overheads like cost and budget. Time to learn and implement the tool. The type of support available for the tool. Limitations toolQ #23) What do you think keeps testers back to do automation? Is there a way to overcome it? Answer: The main obstacle for testers is to study programming/coding when they want to automate. Because testers don't encode, adapting to coding is a bit tricky for testers. We can overcome this through: Collaborate with developers on automation. Given that automation is the responsibility of the whole team and not only testers. special time and focusing on automation. Obtaining proper managerial support. You can save these issues of automation testing interviews as PDF and print for later reading. Q #24) What is an automation testing system? A: The framework, in general, is a set of guidelines. A set of guidelines, assumptions, assumptions, and coding practices to create a runtime in which tests will be automated, known as automation testing framework. The automation testing system is responsible for creating a test harness with a mechanism to connect with the application under the test, enter from a file, perform test cases and generate reports to perform the test. The automation testing system should be independent of the program, and it should be easy to use, change or expand. Q #25) What are the important modules of the automation testing system? Answer: Important modules for automating the testing framework are: Test Assertion Tool: This tool will provide statement statements to test the expected values in the program below the test. Such as. TestNG, Junit, etc. data settings: Each test case must take user data either from a database or from a file or embedded in a test script. The framework data module should take care of data consumption for test scenarios and global variables. Build Manager: The frame must be built and deployed to use test scenarios. Continuous Integration Tool: With CICD (continuous integration and continuous development) in place, a continuous integration tool is needed to integrate and deploy changes made within each iteration. Reporting Tool: The reporting tool is required to create a report that can be read after test cases are executed to better view steps, results, and errors. Logging Tool: The within logging tool helps to better debug errors and errors. Q #26) Explain some automation testing tools. Answer: Some of the well-known automation testing tools are explained below:(i) Selenium: Selenium is a test basis for web application automation testing. It supports multiple browsers and is independent of the OS. Selenium also supports various programming languages such as Java, C#, PHP, Ruby and Perl, etc. Selenium is an open source library suite that can be used to develop additional test frames or test scripts to test web applications. (ii) UFT: Unified Functional Testing is a licensed tool for functional testing. It provides a wide range of features such as APIs, web services, etc., and supports multiple platforms such as desktops, web pages and mobile devices. UFT scripts are written in visual basic scripting languages. (iii) Appium: Appium is an open source mobile app testing tool. It is used to automate testing on cross-platform, native, hybrid and web mobile applications. Appium automates any mobile app from any language with full access to API and DBs from test code. Appium is based on the client server architecture and has evolved from selenium. (iv) Cucumber: Cucumber is an open source development tool. It is used for automation of applications and supports languages such as ruby, java, scala, groovy, etc. Cucumber will read the executable specification file written in plain text and checks checks during the test for these specifications. For cucumber to understand scenarios in plain text, we must follow some of the basic syntax rules known as The Gherkin. (v) TestComplete: TestComplete is a licensed automated user interface testing tool for testing applications across platforms such as desktops, web and mobile devices, etc. This provides flexibility to record a test case in one browser and run it on multiple browsers and thus supports cross browser testing. TestComplete has a built-in object recognition algorithm that uniquely identifies an object and stores it in a repository. Q #27) What are the different types of framework testing methods? Answer: There are four types of methods to automate the testing of the framework. They are: (i) Modular testing system: This structure is built on the concept of abstraction. In this frame, the tester creates scripts for each module under the test individually, and then these scenarios are merged in a hierarchical manner to create large test cases. It creates an abstraction layer between modules, so any changes in test scripts for one module do not affect any other modules. Advantages of this base: Easier maintenance and scalability of test cases. It is easier and faster to create test cases with the help of script modules. Disadvantages:Test cases have data embedded in them. Therefore, executing the same test script with different data is a big change at the script level. (ii) Based on the testing framework: As part of data-driven testing, inputs and expected source data corresponding to input are stored in a file or database, and the automated script runs the same set of test steps for multiple data sets. With this framework, we can run multiple test cases where only inputs differ and the execution steps are the same. Benefits: Reduces the number of test scenarios that you want to run. We perform the same scenario several times with different data. Less coding to automate testing. More flexibility to support and fix errors or improve functionality. Test data can be created before the automated testing system is ready. Disadvantages: Only similar test cases with the same set of execution steps can be combined for multiple data sets. Another set of execution steps requires another test case. (iii) Keyword-based testing framework: This is an application-independent testing structure that uses data tables and understandable keywords. Keywords explain the actions that will be performed in the app below the test, and the datasheet provides input and expected source data. Keyword-based testing is an increase in data-driven testing. Benefits: Less coding and the same scenario may be for multiple data sets. You don't need to use existing keywords for actions to create a test case. The same keywords can be used in several test cases. Disadvantages:This structure is more complex because it to take care of keyword actions as well as data entry. Test cases get bigger and more complex, thereby influencing the preservation of the same. (iv) Hybrid Testing Framework: this structure is a combination of all of the aforementioned test frameworks (modular, data-driven and keyword-based). In this framework, test cases are developed from modular scenarios by combining them into a modular testing framework. Each test case uses a driver script that uses a data file, both within a data-driven framework and a keyword-based action file. Benefits: Modular and easy to maintain. Less coding can take care of more test cases. One test case can be performed with multiple data sets. Disadvantages: A complex for reading, maintaining and enhancing. Q #28) When do you prefer manual testing over automation testing? A: We prefer manual testing to automate testing in the following cases: Project short term and script writing will be time-consuming and expensive when compared to manual testing. Flexibility is required. Automated test cases are programmed and configurations are launched in some way. Usabilitis testing should be carried out. Applications/modules are newly developed and have no previous test cases. Laboratory or reconnaissance testing is required. Q #29) Is automation testing in flexible methodology useful or not? Answer: Automation testing is useful for regression, smoke or sanity checks. All of these types of testing in the traditional waterfall model occur at the end of the cycle, and sometimes if there are not many improvements to apply, we may not even have to do regression testing. While in a flexible methodology, each iteration requires a regression test case as some new functionality is added. In addition, the regression set itself continues to grow after each sprint, as the functional test cases of the current sprint module need to be added to the regression kit for the next sprint. Thus, automation of testing in flexible methodology is very useful and helps in achieving maximum coverage of the test in less sprint time. Q #30) List some of the benefits and disadvantages of automation testing. Answer:Benefits: Fewer human resourcesAdding the coverage of the test at less timeRelative execution of test casesFastDisadvantages: Development and maintenance time is greater. CostSkilled tool resources required. Configuring the environmentSawing the script is an issue. Q #31) List some of the advantages and disadvantages of manual testing. Answer:Benefits: No environment setup required. Knowledge of programming is optional. Recommended for dynamically changing requirements. Allow human observation authorities to identify more errors. The cost is less for short-term projects. FlexibilityDesamants: Difficult to perform complex calculations. useTime acceptance of high risk of human errors or errors. More human resources are needed. Q #32) Can we do automation testing without a framework? If so, why do we need a framework? Answer: Yes, we can perform automation even without using a framework. We can simply understand the tool we use to automate and program programming steps that support tools. If we automate test cases without a framework, there will be no consistency in programming scenarios for test cases. A framework is needed to give a set of guidelines that everyone must follow to preserve readability, reuse and consistency in test scenarios. The framework also provides one common basis for reporting and logging functionality. Q #33) How will you automate the basic login functionality of test cases for the program? Answer: Assuming that the automation tool and framework are already in place of the test environment. To check the basic Sign in: Understand project requirement: Sign-in functionality will have a user name text box, password text box, and sign-in button. Define Test Scenarios: To sign in functionality, possible test scenarios: Blank user name and passwordInvalid user name and password Valid user name and invalid passwordAdditone user name and passwordPrepare the data entry file with data that corresponds to each scenario. Run the tool from the program. Specify the user name, password field, and sign-in button. For each test scenario, obtain data from the data file and enter it in the appropriate fields. Once you're entering data, the sign-in button is pressed. Check the error message for negative scenarios and success messages for positive scenarios in the test script by using statements. Run the test package, and then create a report. Q #34) Automation testing black box testing or white-box testing? A: Automate testing is basically black box testing as we simply program the steps that the manual tester performs for use under the test without knowing the low-level design or code of the program. Sometimes automated test scripts require access to database parts used in the app under the test or some more detailed coding details and can thus be a type of white box testing. Thus, automated testing can be both black and white type of testing depending on the scenarios in which automation is performed. Q #35) How many test cases are you automated per day? A: Well, the number depends on the complexity of the test cases. When the difficulty was limited, I was able to automate 5 to 6 test cases a day. Sometimes I be able to automate only one test case for complex scenarios. I also split my test cases into various components such as, take input, do calculation, check output, etc. in case of very complex scenarios and took 2 or more days. Q #36) What factors determine the effectiveness of automation testing? Answer: Some of the factors that determine the effectiveness of automation Time saved by using running scripts over manual execution of test cases. Defects DetectedInstant coating or code coverageEming the time or time of developmentState developmentMore

ReuseQualification of software under testQ #37) What test cases can be automated? Answer: Types of test cases that can be automated: (i) Cases of smoke testing: Smoke testing is also known as assembly verification testing. Smoke test cases run whenever a new build is released to test the health of the assembly for adoption to perform testing. (ii) Regressive test cases: Regression testing is testing to make sure previously developed modules function properly after adding a new module or fixing an error. Regressive test cases are very important in the incremental approach of the software, where new functionality is added at every stage of the increment. In this case, regression testing is carried out in each incremental phase. (iii) Complex calculated test cases: Test cases, which include some complex calculations to validate the field for the program, fall into this category. Complex calculation results are more prone to human errors, hence when automated they produce accurate results. (iv) Data-driven test cases: Test cases that have the same set of steps and work multiple times with data changes known as data-driven test cases. Automated testing for these types of test cases is fast and cost effective. (v) Dysfunctional test cases: Test cases, such as load tests and performance tests, require a simulated environment with multiple users and multiple combinations of hardware or software. You cannot manually configure multiple environments for each combination or number of users. Automated tools can easily create this environment for easy nonfunctional testing. Q #38) What are the stages in lifecycle testing automation? Answer: Stages of lifecycle testing automation include: Solutions to perform automation testing. Identify and learn about the automation tool. Determine how much automation is tested. Develop and develop a test kit. Execution checkErring The sewing of test scripts. Q #39) What is an automatic testing scenario? Answer: An automated test script is a short program that is written in the programming language to perform a set of instructions on the application under the test to check if the application is in accordance with the requirements. This program at startup gives the test results how to pass or does not depend on if the application is as expected. ConclusionThon are the main issues that do not ardate the tool of automation or programming language. Automation of interview testing also includes a tool and programming language specific issues depending on the tool you worked with. Most interview questions of Yiv test automation are based on the frameworks you develop, so it is recommended to carefully create and understand your Base. When I interview and the candidate answered my question about the framework, I also prefer to ask a specific language question (java kernel in my case). Questions begin with the basics of Java to write the logic of some basic scenarios, such as: How would you extract a set of from this line? How would you extract the URL? On any web page, in any frame, the number of links and its contents change dynamically, how would you handle it? How do you handle images and flash objects? How do I find a word in a line? The answers to all these interview automation test questions are very specific to the tool/language you use for automation. So, before you go for an interview, clean your programming skills. In case you didn't get a chance to create your frame and someone else created it and then take some time to understand it carefully before sitting for an interview. Some tips for automating interview testing will be: Know your tool carefully. Learn the locator techniques used by your tool. Practice programming using the language you use to automate testing. Learn the frames and its components. It's always profitable if you've been involved in developing your framework. So, be careful with the modules within which you worked. I hope these questions will be useful for you to prepare for test automation interviews. Interview.