Automation Testing Interview Questions for Freshers

Automation Qa interview questions for freshers comprise basic knowledge. You may be asked questions like these:

1. When is a good time to automate a test?

Automation is beneficial when the test is repetitive, the feature's behavior is stable, it takes time for a human tester, the test involves complex computations, and it ensures the previous functionality remains intact after a new change.

2. Mention the parts of a test automation framework.

A test automation framework consists of tools, libraries, and guidelines. It includes test data sources, methods, and reusable models. The framework can be code-based or code-free and can be open-source or commercial.

3. Should you automate all testing?

No, not all testing can be automated. Some types of testing, like exploratory testing, usability testing, and user interface testing, require human intervention.

4. What is browser automation?

Browser automation involves using software to open a web application in a browser and perform actions automatically, mimicking human interaction.

5. What is cross-browser testing?

Cross-browser testing checks if an application functions correctly on different web browsers, such as Chrome, Firefox, Safari, and Internet Explorer.

6. What is automated regression testing?

Automated regression testing ensures that existing functionality still works after new code changes are introduced. It prevents the introduction of new defects.

7. Explain the test automation pyramid.

The test automation pyramid emphasizes having a large number of unit tests and fewer end-to-end UI tests. This approach provides better test coverage and faster test execution.

8. What is the role of an automation testing framework?

An automation testing framework provides guidelines and best practices for mastering test automation. It includes functional libraries, object details, test data sources, and methods.

9. What are the types of automation tests?

Common types include unit testing, GUI testing, functional testing, smoke testing, integration testing, and regression testing. Each serves a specific purpose in the testing process.

10. Can automation testing replace manual testing?

No, both have their roles. While automation is efficient for repetitive tasks, manual testing is crucial for exploratory, usability, and ad-hoc testing.

11. What are the key steps in the automation testing life cycle?

The steps include defining goals, selecting frameworks and tools, creating a test plan, setting up the environment, developing and executing test cases, and analyzing test reports.

12. Name some popular automation testing tools.

Selenium, Appium, UFT, Ranorex, and SoapUI are among the popular tools used for automation testing.

13. What factors should be considered when selecting an automation tool?

Project requirements, budget, features, reusability, reporting capabilities, and ease of use are important factors to consider.

14. Can we achieve 100% automation in testing?

No, some scenarios like CAPTCHA challenges and scenarios that require human intuition are not suitable for complete automation.

15. How do you ensure the effectiveness of automated tests?

Regularly review and update test scripts, maintain test records, analyze test reports, and ensure good coverage of software functionalities.

16. What are the main components of Selenium?

Selenium comprises Selenium WebDriver, Selenium IDE, and Selenium Grid. Selenium WebDriver is used to automate browser interactions, Selenium IDE is a record-and-playback tool, and Selenium Grid is used for parallel execution across multiple browsers.

17.What is a test case?

A test case is a detailed description of a test scenario, including input data, execution steps, and expected outcomes. It helps ensure that the software meets specified requirements.

18.How can you handle dynamic web elements in Selenium?

Dynamic elements have changing attributes. To handle them, you can use techniques like waiting for an element to be visible or clickable, using dynamic XPath, or leveraging explicit waits.

19. Explain data-driven testing.

Data-driven testing involves executing the same test case with multiple sets of test data to validate different scenarios. It helps uncover defects under various conditions.

20. What is the importance of version control in automation testing?

Version control (e.g., Git) helps track changes to test scripts, manage collaboration among team members, and maintain different versions of code. It ensures traceability and provides a history of changes.

21. What is a code review, and why is it essential?

A code review involves peers reviewing and analyzing code for quality, consistency, and adherence to coding standards. It helps identify bugs early, improve code quality, and share knowledge among team members.

22. How do you handle exceptions in automation testing?

Exceptions are handled using try-catch blocks in programming languages. In automation, exceptions may occur due to element not found, timeout, etc. Proper exception handling improves script robustness.

23. Explain the importance of parameterization in testing.

Parameterization allows you to run the same test with different data inputs. It enhances test coverage and helps identify defects in various scenarios.

24. What are assertions in automation testing?

Assertions are checkpoints that validate if an actual result matches the expected result. They ensure that the application functions correctly and help identify defects.

25. What is a test report, and what should it include?

A test report summarizes the results of test execution. It includes details about test cases executed, pass/fail status, defect details, test environment, and recommendations for further testing.

26. How would you ensure cross-browser compatibility testing?

Cross-browser testing involves running test cases on different browsers to ensure consistent behavior. It requires creating separate test cases for each browser and handling browser-specific issues.

27. What is continuous integration in automation testing?

Continuous Integration (CI) involves integrating code changes regularly into a shared repository. Automated tests are run after each integration, ensuring early detection of defects.

28. How do you handle dynamic content loading in Selenium?

Dynamic content loading can be handled using explicit waits or implicit waits. Explicit waits wait for a specific condition, while implicit waits wait for a certain time before throwing an exception.

29. Explain the difference between “findElement” and “findElements” in Selenium.

findElement returns the first matching element on the page, while findElements returns a list of all matching elements. findElement throws an exception if no element is found, while findElements returns an empty list.

30. What is the role of a Page Object Model (POM) in Selenium?

POM is a design pattern that abstracts web pages into classes. It improves code maintainability by separating page elements and their interactions from test scripts.