Cucumber Interview Questions & Answers (50+)

Q1. What is Cucumber?

Cucumber is a Behavior Driven Development (BDD) tool used to write tests in plain English (Gherkin syntax).

Q2. What is Gherkin?

Gherkin is the domain-specific language used in Cucumber to describe application behavior in plain text.

Q3. What are Feature Files?

Files with .feature extension that contain scenarios written in Gherkin.

Q4. Difference between TDD and BDD?

TDD focuses on unit testing by developers, BDD focuses on system behavior involving both developers and business stakeholders.

Q5. What are the main components of Cucumber?

Feature files, Step Definitions, Runner Class, Hooks.

Q6. What is Background in Cucumber?

Background defines common steps executed before each scenario in a feature file.

Q7. What are Tags in Cucumber?

Tags like @Smoke, @Regression are used to group and filter scenarios for execution.

Q8. What are Hooks in Cucumber?

@Before and @After run before/after each scenario. @BeforeStep and @AfterStep run before/after each step.

Q9. How do you share test data between steps?

Using Scenario Context or Dependency Injection frameworks like PicoContainer/Spring.

Q10. What is the use of Cucumber Options?

@CucumberOptions in Runner class defines features, glue, tags, and reporting plugins.

Q11. Difference between Scenario and Scenario Outline?

Scenario runs once with specific data, Scenario Outline runs multiple times with Examples table data.

Q12. What is a Step Definition?

It is a Java method that links Gherkin steps with test automation code.

Q13. How do you generate reports in Cucumber?

Using plugins like pretty, html, json. Extent Reports and Allure can be integrated for rich reports.

Q14. What is Dry Run in Cucumber?

Checks if all steps in feature files have step definitions without executing the tests.

Q15. Can you run Cucumber with TestNG?

Yes, by extending AbstractTestNGCucumberTests. It also helps in parallel execution.

Q16. How do you handle parameterization in Cucumber?

Using Regular Expressions or Cucumber Expressions in Step Definitions.

Q17. What is DataTable in Cucumber?

It allows passing tabular data from feature files into step definitions.

Q18. How do you handle multiple browsers in Cucumber?

By parameterizing browser type in feature files or configuration files and launching WebDriver accordingly.

Q19. How do you skip scenarios in Cucumber?

By tagging them with @Ignore or excluding them in Runner class with tags.

Q20. What are Cucumber Expressions?

They are an alternative to regular expressions for defining step definitions in a simpler way.

Q21. How do you integrate Selenium with Cucumber?

By writing Selenium WebDriver code inside Step Definitions and using POM for maintainability.

Q22. How do you handle API testing in Cucumber?

By integrating RestAssured inside Step Definitions and writing feature files for API scenarios.

Q23. How do you validate data between UI/API and Database?

Fetch data from UI/API, fetch DB data via JDBC, and compare both for consistency.

Q24. How do you manage large test suites in Cucumber?

By modularizing feature files, using tags, and following POM for reusable step definitions.

Q25. How do you achieve parallel execution in Cucumber?

Using TestNG, JUnit with Maven Surefire, or Cucumber JVM parallel plugin.

Q26. What types of reports are commonly used in Cucumber?

Pretty, HTML, JSON, Extent Reports, Allure Reports.

Q27. How do you handle flaky tests in Cucumber?

Using explicit waits, retry mechanism, ensuring independent tests, and stable environments.

Q28. How do you integrate Cucumber with Jenkins?

Run via Maven command mvn clean test, publish HTML/JSON reports, integrate with Jenkins plugins.

Q29. What are Plugins in Cucumber?

They are reporting/configuration options specified in @CucumberOptions (e.g., pretty, json, html).

Q30. How do you log information in Cucumber tests?

By using loggers (Log4j/SLF4J) or capturing screenshots in hooks for failed scenarios.

Q31. Explain a login feature automation using Cucumber?

Feature files contain Given-When-Then steps, Step Definitions implement Selenium logic to enter username, password, and validate home page navigation.

Q32. How do you implement search functionality testing with Cucumber + API?

Use RestAssured in step definitions to send GET request and validate response with assertions.

Q33. How do you test user registration with multiple data sets?

Using Scenario Outline with Examples table for multiple usernames, emails, and passwords.

Q34. How do you handle database validation after a transaction?

Fetch DB records with JDBC after performing transaction and assert expected results.

Q35. How do you attach screenshots in Cucumber reports?

By capturing screenshots in @After hook and embedding them into reports using Scenario.attach.

Q36. How do you handle exceptions in Step Definitions?

Using try-catch blocks and logging errors, also marking scenario as failed gracefully.

Q37. How do you integrate Cucumber with CI/CD pipeline?

By using Jenkins/GitHub Actions to trigger mvn test and generate reports for stakeholders.

Q38. How do you reuse common steps in Cucumber?

By writing generic step definitions and using Background for common preconditions.

Q39. How do you perform cross-browser testing in Cucumber?

By integrating Selenium Grid or BrowserStack and parameterizing browser type in feature files.

Q40. How do you secure sensitive test data in Cucumber?

By using encrypted properties, environment variables, or external test data providers.

Q41. How do you integrate Cucumber with Spring?

By using Spring dependency injection for managing beans, drivers, and context sharing.

Q42. How do you implement tagging strategy in a large project?

By defining tags for Smoke, Regression, E2E, and critical paths for selective execution.

Q43. How do you handle asynchronous operations in Cucumber?

By using WebDriver waits in Selenium or Awaitility library in API tests.

Q44. How do you handle dynamic elements in Selenium + Cucumber?

By using dynamic locators (XPath/CSS) and explicit waits for stability.

Q45. What is the difference between Cucumber JVM and SpecFlow?

Cucumber JVM is for Java, SpecFlow is the .NET equivalent for BDD.

Q46. Can you integrate Cucumber with cloud services?

Yes, with BrowserStack, Sauce Labs, or AWS Lambda for scalable test execution.

Q47. How do you manage version control with Cucumber tests?

By maintaining feature files, step definitions, and reports in Git repository.

Q48. How do you handle configuration in Cucumber?

By using config.properties, YAML, or environment variables to avoid hardcoding.

Q49. How do you ensure maintainability of Cucumber framework?

By following POM, modular design, tagging strategy, and reusable step definitions.

Q50. What challenges did you face with Cucumber and how did you solve them?

Duplication of steps (solved by POM), long execution (parallel runs), flaky tests (explicit waits), reporting (Extent/Allure).

Q51. Bonus: What are the latest features of Cucumber?

Improved Cucumber Expressions, DataTable enhancements, and integration with modern reporting tools.