

Planit Technical Assessment – Automation

1. What other possible scenarios would you suggest for testing the Jupiter Toys application?

Following are the possible scenarios to test Jupiter Toys application:

- Verify login functionality
- Verify quantity, subtotal of each product and total of all products when quantity is updated in the cart page
- Verify delete or empty cart button
- Verify check out page and validate errors for mandatory fields
- Verify payment process

2. Jupiter Toys is expected to grow and expand its offering into books, tech, and modern art. We are expecting the of tests will grow to a very large number.

1. What approaches could you used to reduce overall execution time?

Following are some of the approaches to improve execution time of automation tests:

- **Parallel testing** can be used to run automated tests on different device-browser-OS combination simultaneously, so that the entire test suite can be completed in much lesser time.
- Use **swift locators** which can quickly find an element such as ID, name, xpath and CSS selector etc.
- Using **explicit waits** is another best way to reduce execution time, it enables the next step in a script to execute as soon as the preceding step is complete.
- **Automate tests only when needed**, since few test cases can be done more quickly through manual testing.
- Test scripts should be **optimized** as much as possible.

2. How will your framework cater for this?

- Make sure all configuration files are updated accordingly.
- Make necessary changes in the framework to support upgraded version of the application.
- Add any required libraries or jar files.
- Prepare test data for any newly added or updated features.
- Run all previous test cases and make sure all tests are passed.
- Repeat across your various environments like development, QA, UAT, whatever your process involves.

3. Describe when to use a BDD approach to automation and when NOT to use BDD.

BDD is an evolution of TDD. Writing scenarios from the beginning enforces quality-first and test-first mindsets. BDD automation can run scenarios to fail until the feature is implemented and causes tests to pass. BDD framework allows an automation tester to easily turn test scenarios into automated tests and initiate the scripting with the right

approach. The steps are already given by the scenarios – the automation tester simply needs to write a method/function to perform each step's operations.

BDD should be used for verifying the most important parts of the application using end-to-end tests and should also be used to verify the wanted behavior using integration tests. BDD provides a path to overcome the gap between the technical and the non-technical teams because the test cases are commonly written in simple text, i.e., English. The main advantage of BDD is the clearer approach which is easier to understand.

Since communication is essential between the parties - user/client and the development team, absence of any one of them, can cause the process to have ambiguities and lack of answers to the questions/doubts raised by either side. It would be better not to go for BDD approach when there is a lack of communication within the team.