**Questions:**

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

Response from Rupesh: we can test/automate below scenarios in the Jupiter Toys application:

1. Login functionality-

* Verify successful login with valid username and password
* Verify login without entering either username or password
* Verify login with entering incorrect username and password

Graphical user interface, text, application, email

Description automatically generated

1. Verify all the links are functional or broken in the Jupiter Toys application

* ‘Jupiter Toys’, ‘Home’, ‘Shop’, ‘Contact’

1. Remove Item functionality in the Cart page

* Remove any item and verify if the Quantity, subtotal, and Total amounts are updated accordingly

Graphical user interface, text, application, email

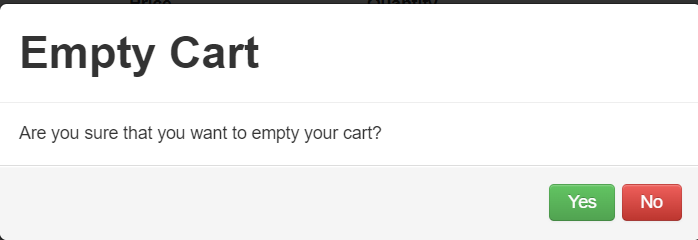
Description automatically generated

1. Verify ‘Start Shopping’ option from the Cart page when there no items in the Cart to validate successful navigation to Shop page

Graphical user interface, text, application, chat or text message

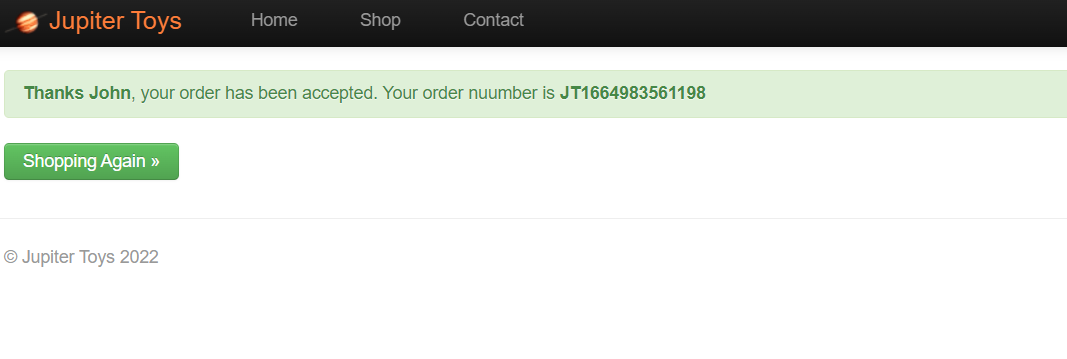
Description automatically generated

1. Verify Empty Cart functionality in the Cart Page



1. Verify Checkout functionality in the Cart page

* Verify mandatory fields messages in the Delivery Details page
* Verify successful submission of the Delivery details and verify submission message



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?

Response from Rupesh:

* We can use execute test cases on different remote machine parallelly to reduce overall execution time.
* We can try to test limited functionalities in a given test case, which will help us the automate test case execute quickly (pass or fail), if multiple functionalities are included chances of failure increases which may lead to re-execution effort post defect fixes.
* Limit usage of implicit waits like thread.sleep(), and usage of more explicit waits function to reduce test execution time
* Wherever possible using non-UI based testing approach like API tests as they are comparatively faster than UI Automation test cases.

1. How will your framework cater for this?

* Usage of correct Web locators, which may help to reduce execution time
* Minimize code duplication as each command takes certain amount of time to execute
* Limit repeated executions of Test cases which always passes

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

* When business users, product owners are actively involved in Test automation then we can use BDD approach so the Test automation use cases can be defined in more of a user story way/simple English which can be interpreted by Business users
* BDD may require good collaboration between developers, Testers and Business users which is seen more in Agile methodology, if in case teams is working as a separate automation team BDD may not suite.