1. The FAST In-built automation tool does not support the identification of UI-related issues.
2. Support and guidance from the FAST team are required for automating each test case, as there is no available documentation on how to configure and use the Orchestrators.
3. The FAST In-built tool does not assist in monitoring UI performance, as it accesses the database server directly through FAST Orchestrators.
4. Managing and maintaining test cases and test suites is complex and cumbersome.
5. The FAST tool does not support a Test-Driven Development (TDD) approach, and providing all test inputs in XML format is a tedious process.
6. New automation resources need to invest time in learning the FAST In-built tool before being able to work with it effectively.
7. Even after receiving knowledge transfer (KT) from the FAST team on how to use the In-built automation tool, support is still required to automate new functionalities or test cases, as only the FAST team is aware of which orchestrator should be used.
8. The Selenium Reporter gives a better understanding when there is a failure in the UI along with logging the failure, it also captures the screenshots for better understanding of the end user.
9. We start Automation with the policies that were generated from ALIP as an input parameter for our Input test data sheet and perform the required transactions on the policy whereas we’re not sure how FAST captures the policy values from ALIP to their test suites.
10. For every transaction/test case step, we’ve a dependency on FAST team to get to know the fields’ XPaths for validation. Also, that XPath’s value may not be a field value in the UI as it is fetched from the backend from the response payload.