1. **BIG ISSUE** : browser.quit() has been used after Assert statement. This line becomes unreachable when assertion fails. This leaves browser instances open even after test is completed. This becomes a big problem when the tests are run in server using Jenkins. This opens many browser instances and slows down server performance.
2. **IMPORTANT** : Refactored code to handle auto-suggestions. Changed findElements to findElement. Reason is whenever only one element is used out of a list of elements its better to go with findElement only. findElements can be used when we use more than 2 elements out of list like handling Web Tables etc., This effects performance of the code. We can observe the performance variations when the list of elements contains 1000+ of elements(Ex: 1000+ rows of a webtable)
3. Unused import statements should be removed. This increases the performance of the Java application.
4. Code is not designed in re-usable way. Methods developed in each test can be moved to a common library and used from there.
5. Data is hard coded within tests. The data should be driven from files like Excel, txt, properties file. This helps in configuring the data easily from one place
6. Hard waits has been used which increases test time. Selenium provides wait mechanisms like Explicit, Implicit and Fluent waits which reduces wait time and provides a variety of conditions that can be used in order to wait to meet the condition.
7. Locator 'To Location' field is not correct in FlightBookingTest.java
8. Page Objects and test classes should be separate classes in order for the Page Objects to be available for multiple tests ie., Reusable Page Objects
9. Selecting date in Date picker is hardcoded (19), this should made dynamically select date.
10. Application URL is hardcoded. This creates a problem when our client has more environments for the application like QA, DEV, UAT etc., Application URL should be maintained in a configurable file.
11. Java docs and proper comments are not implemented. Implementing Java docs and comments helps in understanding the framework and quick ramp-up of new resources onto work.