How do you deal with flaky tests?

Flaky Tests are mostly caused in test pipeline which mainly fails due to timeout synchronization or low latency over network, Environment related issue in specific time or with interrupted memory leaks.

Flaky tests can be handled in following ways

1. Visualizing Test Runs
    - While test-run visualizations alone can give us an idea of how well tests work in combination with multiple randomized test runs. It gives a clear depiction of whether flaky tests are increasing or decreasing over time.
2. Quarantining Flaky Tests
    - Creating a separate suite of flaky test cases. This suite helps us to save time from being duplicating part of work
3. Implementation of proper pre and post conditions for tests before run
    - Removal of all existing data and states before and after each test run provides effective solution
4. Implementation of effective timeouts
    - Use of dynamic waits helps in reducing issue with timeouts

Let's suppose there is a test pipeline taking about 1 hour to finish, what would you do to decrease the time of it?

There are 2 ways test pipeline can be optimized in terms of

1. Running tests in parallel
    - Instantiating test execution in parallel helps to reduce test pipeline execution time to 60% where all the test script gets executed independently and with different browser instances

2. Implementation of docker containers
    - Implementation of docker containers is a very timely and cost-effective solution where all browser instances are being created on single machine in the forms of containers and these container gets removed after completion of each automated test execution
    - Container creations are explicitly handled over ISO images of browsers which are provided over cloud from docker side. Test engineer need to run to docker-compose.yml file as part of test pipeline which contains ISO image path of browser type required to run automated tests.
    - Hence it reduces a significant time in instantiating browser instances very effectively and without any cause of failure.

Imagine you have the possibility to ask software engineers to develop tools for you that will increase your productivity as full-stack QA, please describe them your requirements

Implementation of Test Data Generator Tool

- Application should provide set of test data based on configuration provided for all the test environment like Test, Stagging, etc

Implementation of Automated Build Provider

- Tool will help in initiating test runs over different environments based on build Number provided.
- Test execution will run for each checked-in code build over CICD pipeline