I would suggest a few different types of tests for different occasions:

Review the requirement specification, it has been proven that the majority of bugs exist in the requirement specification. The early identifying the bugs from the spec, the cheaper and quicker for the development roll out.

Confirmation testing need to be carried out when a new feature or bug fix is made

Automation tests strategy as per below

Functional testing:

1. Unit tests for both backend and frontend components, these tests live with the production projects. These tests will definitely be triggered during the build process either manually run a build for a customer branch or merging the customer branch to the master branch in the build pipeline. It is recommended to execute the tests when making a code push from local to the remote repo.
2. DB integration tests for testing backend services with actual databases, these tests also live with the production projects. Recommend to use docker to stand up a database which has data similar to actual production data. These tests will definitely be triggered during the build process in the build pipeline. It depends on how long this test execution will take. If it is quick, then recommending to execute the tests when making a code push from local to the remote repo. Otherwise, they can also be triggered manually in the local environment.
3. API integration or Contract testing for the APIs. This is a grep box testing, which lives outside of the production projects. The testers need to know the contracts such as the endpoints, request headers and payload for sending a request. Also they need to know the expected response content such as response code, response payload, error messages if it is for negative tests. These tests should be triggered nightly or when making a merge.
4. E2E UI and E2E API testing. These tests are expensive, so we usually cover the happy paths for this level of testing. We can create a small number of E2EUI tests for most common user cases (examples for Buggy Cars Rating: Registration, Update Profile, login and access a model). If there are any minior variations for the user cases, then implement E2E API tests instead, because it does not need to start browser instances, waiting for page rendering, etc. The full set of these tests should run nightly, a setup of tests should be executed for smoke testing purposes when making a build. If the team is capable, cross browser and cross system testing should also be considered for E2E tests. To improve the performance, consider executing the tests in parallel.

The performance for unit test execution should be milliseconds to seconds. For DB integration and API testing, it should be done within 3 minutes. In general, it will be good if no more than 20 minutes considering the factors like slow networking and external systems dependencies for E2E UI and API full set testing. For small projects like Buggy Cars Rate, the benchmark for E2E testing is 5 minutes maximum.

The sizes of the tests for different types of automation tests should be E2EUI < E2EAP < API integration (Contract) < DB Integration < Unit

Non Functional testing:

1. Load and Stress testing for the APIs
2. Visual regression testing for the UI rendering
3. Pen testing for the security

Exploratory testing should always be carried out before any major release.

If it is possible, we can do alpha and beta testing for the new features. Also it is important to do the sanity testing in the Production environment whenever there is a release.