Frameworks of choice:

Jest:

On both frontend and backend we use jest for unit and integration within our application as it is a very versatile framework with powerful features such as mocking and snapshots. It is also very compatible with react applications and has built-in code coverage support functionality. Alternative frameworks we considered were Mocha and Chai, both popular javascript testing frameworks, however neither provides built-in mocking capabilities and would have required more setup within our react native app than jest.

Playwright:

We are using Playwright for end to end testing in our application. The reason we chose a browser automation and testing framework was due to the responsiveness of Playwright to any changes in the codebase as opposed to end to end testing frameworks for mobile applications, e.g Detox and Appium typically require builds of the application to run tests. It also provides extensive control over multiple windows which is helpful in testing our auth flow in a pure end to end fashion.

While Detox is designed for React Native, its expo compatibility is lacking, being exclusively a community driven effort which remains unmaintained and unsupported by Detox. This would be inefficient for tweaking tests as opposed to the responsiveness of Playwright to any changes in the codebase.

Automated testing:

Our repository runs automated tests using github actions on every pull request into our develop branch to ensure that no code with failing tests is allowed into the branch. We also utilise code coverage as a metric to determine if a pull request is allowed into the branch. A pull request may not reduce code coverage and a minimum percentage of new code must have passing tests to be eligible for merging.

- **Code Coverage:** https://app.codecov.io/github/COS301-SE-2024/TuneIn
- **Automated Test Workflow:** https://github.com/COS301-SE-2024/TuneIn/blob/develop/.github/workflows/test.yml
- **Frontend Tests:** https://github.com/COS301-SE-2024/TuneIn/tree/develop/frontend/__tests__

- **Backend Tests (spec files in folders):**
  https://github.com/COS301-SE-2024/TuneIn/tree/develop/backend/src/modules