**A) Testing with Jest (RECOMMENDED)**

* Usually we need many tools for the ReactJS unit testing , at least the below tools
* Mocha, Karma, Jasmin (unit tests)
* Chai (assertions)
* Istanbul (coverage)
* Protractor(E2E for angular built on jasmine)

* All these replaced by JEST
* jest (unit tests, assertions, mocks, coverage, live test reload, )
* "Zero configuration" tool  
  src and test folder don’t have to be the same, (tests are closer to the source code in jest)
* By default — jest searches all the directories named \_tests\_ and runs all the js files inside
* jest can be set to mock the module/files in \_mocks\_ folder  
  so that you dont have to create them manually for development and testing
* jest --watch(Rerun the tests automatically, when the code changes)
* jest is fasts (runs parallel tests, runs previously failed tests first, )
* integrates with babel to compile JS code using babel (npm i -d babel-jest)
* jest --coverage(outputs coverage report (default txt), but can be configured to output html reports)
* jest --coverage --watch (to run the uncovered scripts)
* Coverage threshold if not met, jest will fails the test (ex: 90%)
* Error msgs are helpful, visual and colour coded, correct stack traces
* integrated timer mocks (call back functions)
* snapshot testing (jest -u to update \_snapshot\_ files)  
  Ex: converting DB to API
* support for promises and async/await
* Automatically resets global state(global variable) for every test
* jest comes pre-configured in "Create-react-app" and "react-native" projects
* Automatically converts code from other frameworks like Mocha, Tape to Jest (jest-code mods)

**B) Testing with Mocha/Chai?**

* #grouping similar tests (Test suite)  
  describe ('App',() =>),   
   it(test specific attribute) ,   
   #assertion  
   expect(component).to.have.class('comment-box');  
   expect(component).to. Contain('React Simple Starter');  
   expect(component. Find('button')).to.exist;
* mocha(suite executor/runner) collects the tests first, and queues them,   
  (Automatically reruns the tests when either the source code or test code is changed, this can be done using "nodemon --exec \"npm test\"" (slashes in windows)
* tests uses jquery — chai library (<https://github.com/chaijs/chai-jquery>)

**C) Other Unit testing tools**

* Karma/Jasmin and Istanbul are mainly for unit testing of AngularJS Application
* Karma(js test runner, and generate reports, supports other FWS)
* Jasmine(js test fw, used to write test scripts, no dependency on DOM, assertion based testing using matchers
* Phantomjs (headless browser, no GUI, protractor(E2E for angular built on jasmine)  
  Angular Mocks(lib allows angular services to be mocked)
* Function coverage, branch coverage, Line/statement coverage
* Improving code coverage by using TDD approach

 CONCLUSION: Jest is the recommended tool when compared with Mocha or any other tools, as it handles everything in one single tool.

References:

<https://auth0.com/blog/testing-react-applications-with-jest/>

<https://www.npmjs.com/package/jest-enzyme>

<https://facebook.github.io/jest/>

<https://www.youtube.com/watch?v=tvy0bSgwtTo>