Contents

[Which app is used for testing 2](#_Toc204249287)

[How to start 2](#_Toc204249288)

[Folder Structure 2](#_Toc204249289)

[Start Server and Client via Commands 3](#_Toc204249290)

[Test Results 3](#_Toc204249291)

[Functional UI Automation using Playwright 3](#_Toc204249292)

[API Test Automation 4](#_Toc204249293)

[Test Plan 4](#_Toc204249294)

# Which app is used for testing

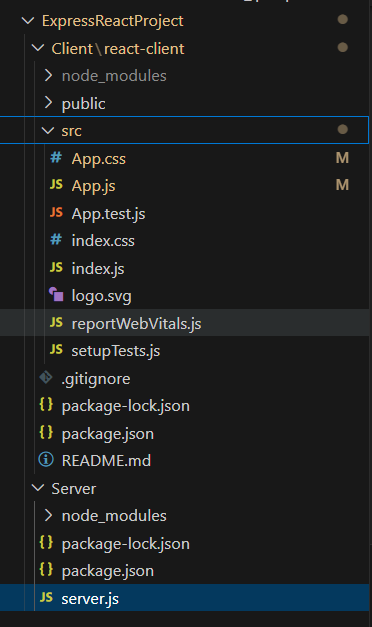
I have made a sample app having basic crud functionalities based on react and node js.

# How to start

Extract Contents from repo and import in vscode under one folder

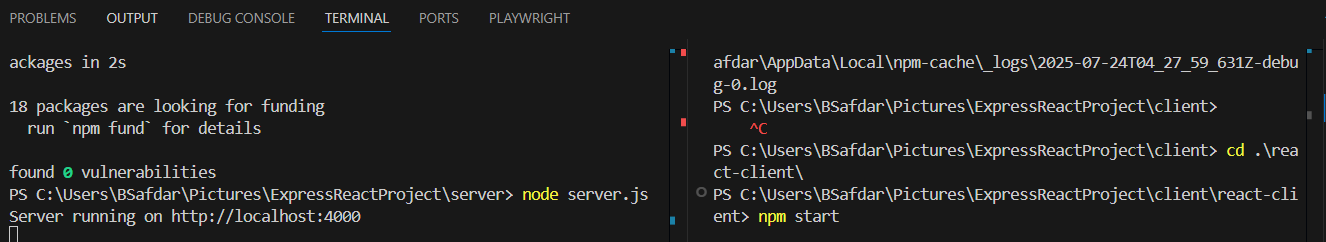
Navigate to the main directory ExpressReactproject

# Folder Structure



# Start Server and Client via Commands

1. node server.js ( in server directory)
2. npm start (in react-client directory)

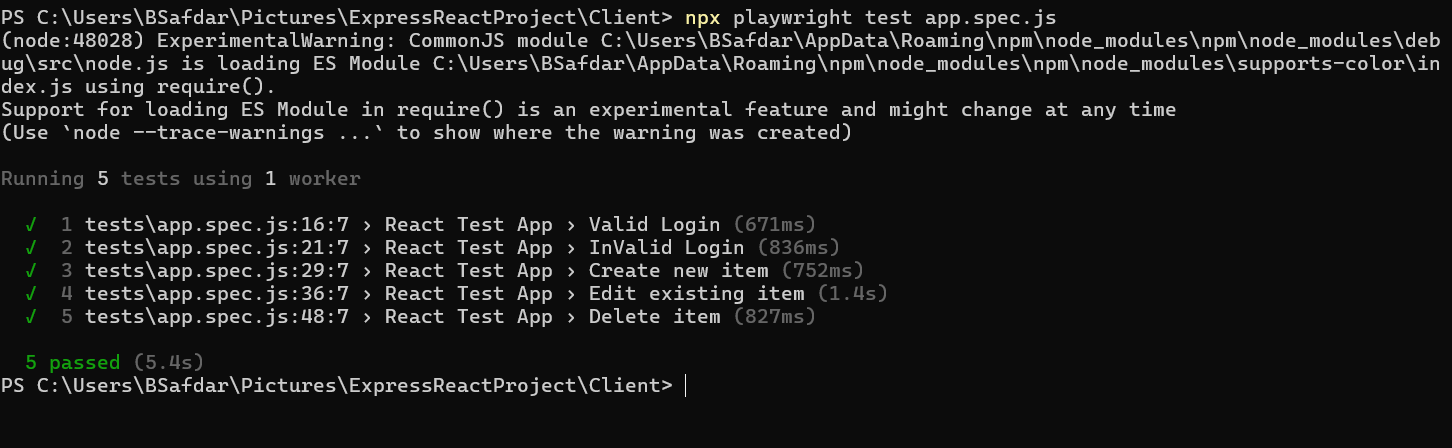
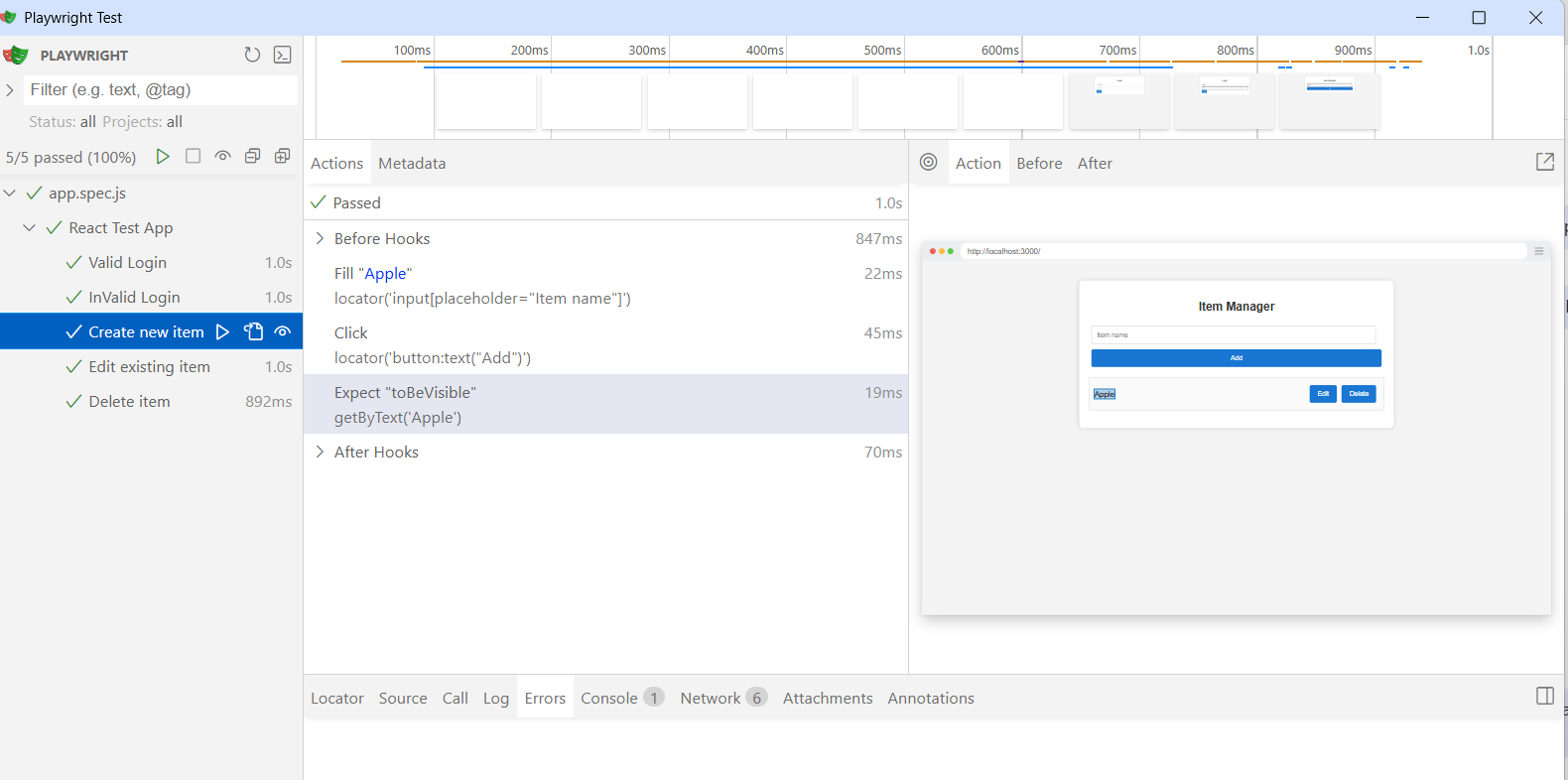


# Test Results

## Functional UI Automation using Playwright

**Command: npx playwright test app.spec.js --ui (for running in UI mode)**

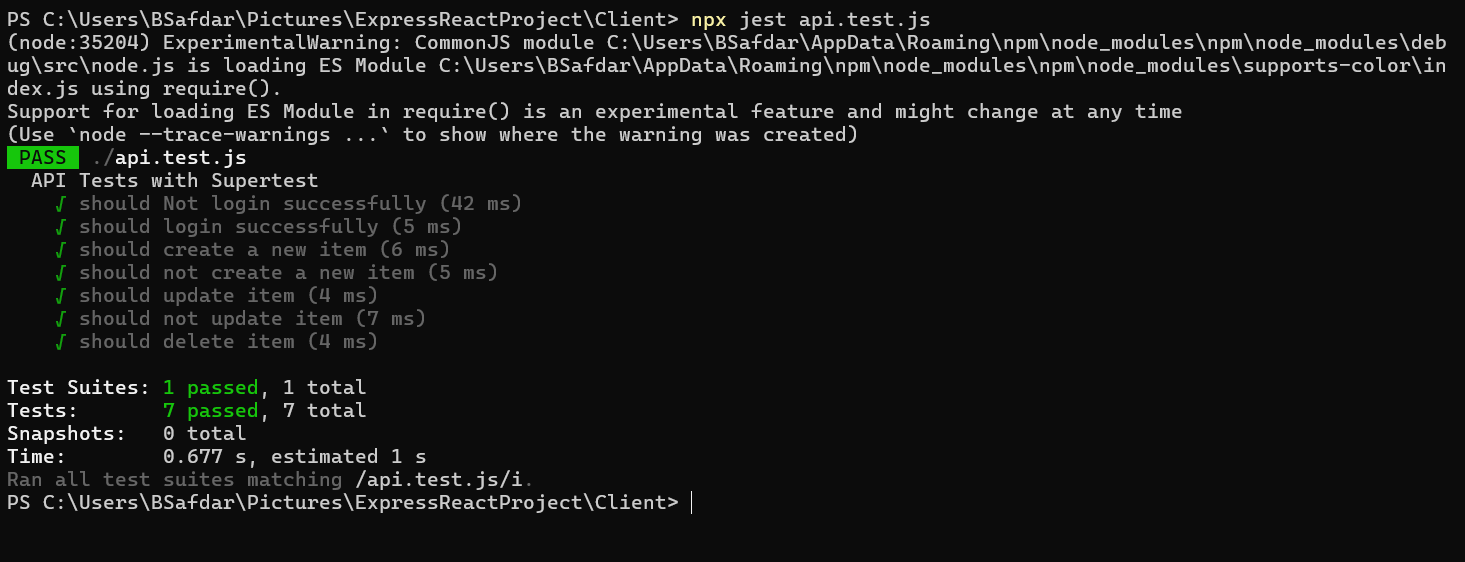
**Command: npx playwright test app.spec.js**

1. 
2. 

## API Test Automation

I have used super test for API testing.

**Command: npx jest api.test.js**

* 

# Test Plan

Testing basic level Positive and Negative Scenarios against the custom sample app.

* What is being tested? Sample app with basic crud operations

* Test coverage areas? Create, Read, Update and Delete
* Tools used and why? **Playwirght** and **supertest** is used as it complies with the overall stack of Node js and React js
* How to run the tests? Details have been mentioned above
* Any assumptions or limitations?
  1. Node js was already installed in my system so, assuming it would be installed on the system where it would be tested
  2. I would suggest to run Playwright tests first then the API ones just to see actions more clearly
  3. Only limited scenarios are handled because of time constraint