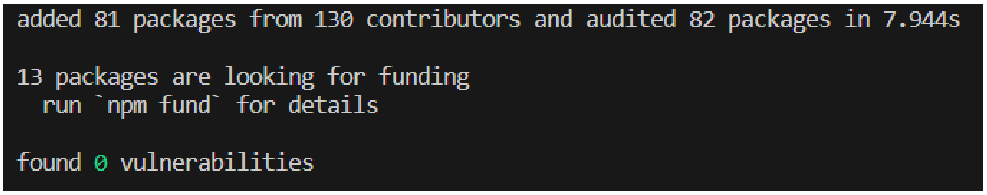
Limitation

I actually have no actual experience with React but I'll try my best to do the exercise.

Compiling and running the mock-api

npm install

After npm install there are no vulnerabilities

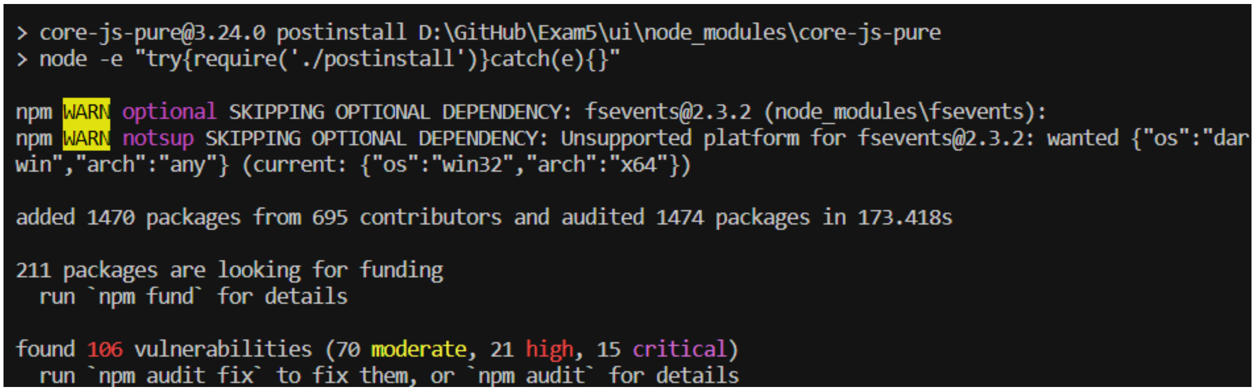


```
added 81 packages from 130 contributors and audited 82 packages in 7.944s

13 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

Compiling and running the ui

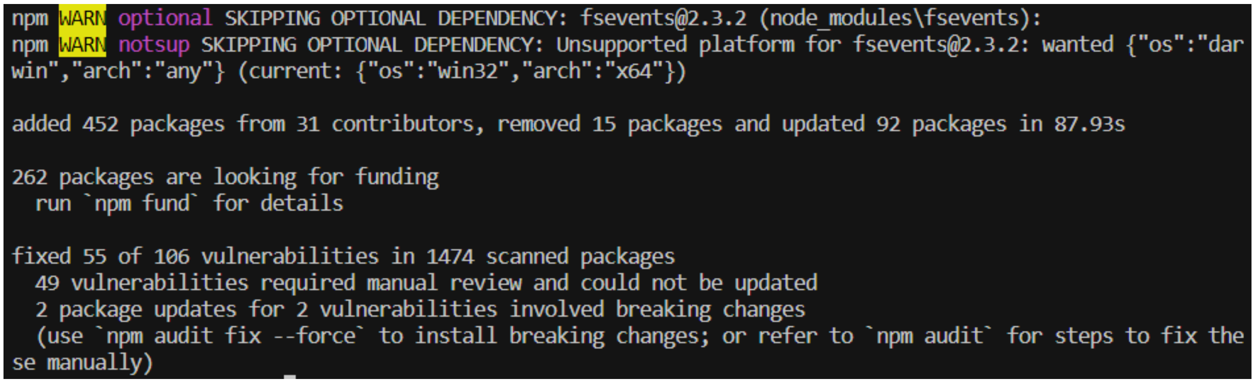
npm install

After npm install



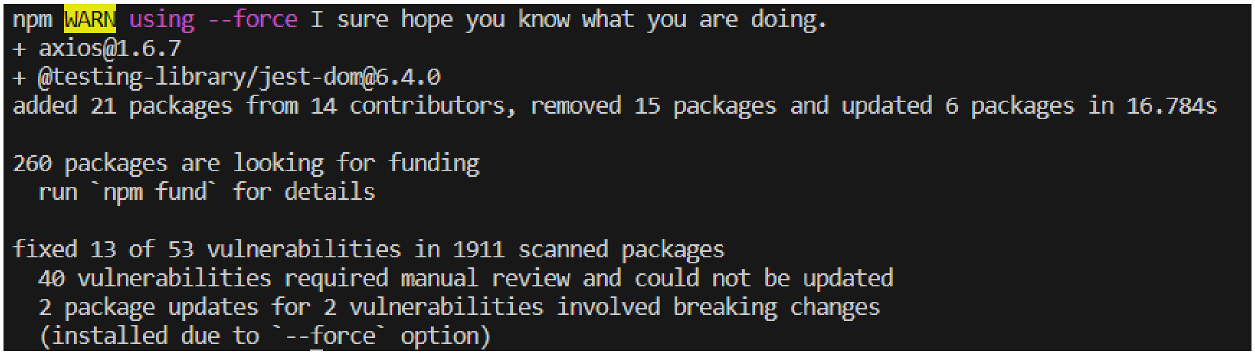
```
> core-js-pure@3.24.0 postinstall D:\GitHub\Exam5\ui\node_modules\core-js-pure
> node -e "try{require('./postinstall')}catch(e){}"

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.3.2 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.2: wanted {"os":"dar
win","arch":"any"} (current: {"os":"win32","arch":"x64"})

added 1470 packages from 695 contributors and audited 1474 packages in 173.418s

211 packages are looking for funding
  run `npm fund` for details

found 106 vulnerabilities (70 moderate, 21 high, 15 critical)
  run `npm audit fix` to fix them, or `npm audit` for details
```

After npm audit fix



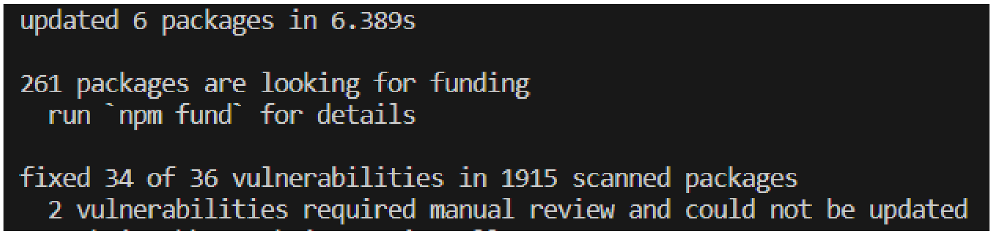
```
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.3.2 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.2: wanted {"os":"dar
win","arch":"any"} (current: {"os":"win32","arch":"x64"})

added 452 packages from 31 contributors, removed 15 packages and updated 92 packages in 87.93s

262 packages are looking for funding
  run `npm fund` for details

fixed 55 of 106 vulnerabilities in 1474 scanned packages
  49 vulnerabilities required manual review and could not be updated
  2 package updates for 2 vulnerabilities involved breaking changes
  (use `npm audit fix --force` to install breaking changes; or refer to `npm audit` for steps to fix the
se manually)
```

After npm audit fix –force



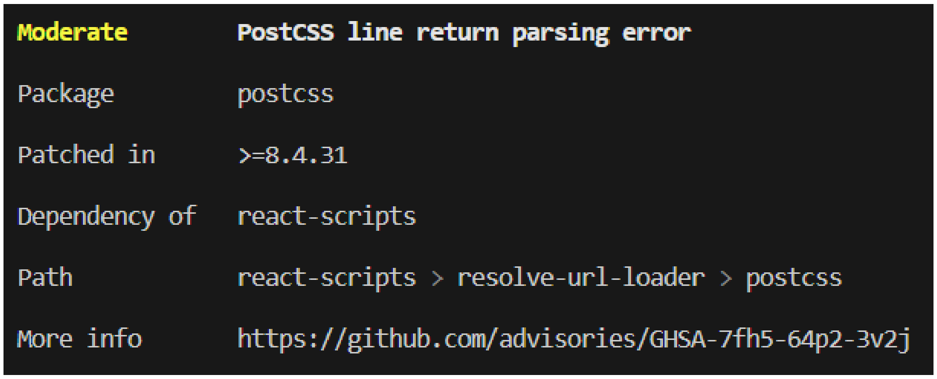
```
npm WARN using --force I sure hope you know what you are doing.
+ axios@1.6.7
+ @testing-library/jest-dom@6.4.0
added 21 packages from 14 contributors, removed 15 packages and updated 6 packages in 16.784s

260 packages are looking for funding
  run `npm fund` for details

fixed 13 of 53 vulnerabilities in 1911 scanned packages
  40 vulnerabilities required manual review and could not be updated
  2 package updates for 2 vulnerabilities involved breaking changes
  (installed due to `--force` option)
```

After npm audit fix again, was able to reduce the vulnerabilities to 2

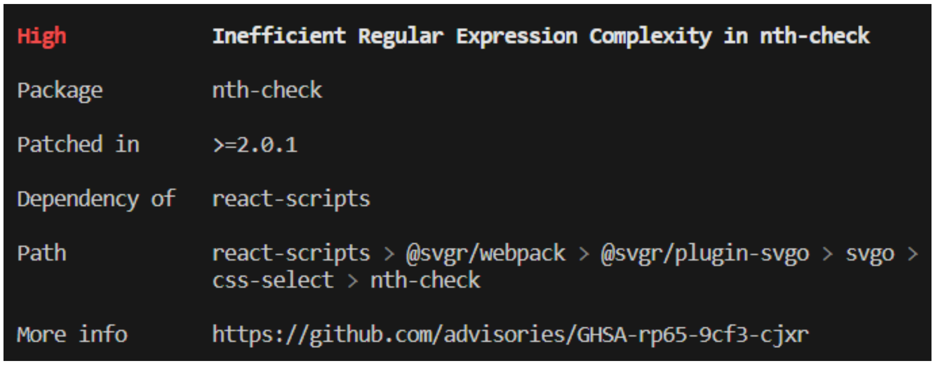


```
updated 6 packages in 6.389s

261 packages are looking for funding
  run `npm fund` for details

fixed 34 of 36 vulnerabilities in 1915 scanned packages
  2 vulnerabilities required manual review and could not be updated
```

Last 2 vulnerabilities are:



```
Moderate        PostCSS line return parsing error

Package         postcss

Patched in      >=8.4.31

Dependency of   react-scripts

Path            react-scripts > resolve-url-loader > postcss

More info       https://github.com/advisories/GHSA-7fh5-64p2-3v2j
```



```
High            Inefficient Regular Expression Complexity in nth-check

Package         nth-check

Patched in      >=2.0.1

Dependency of   react-scripts

Path            react-scripts > @svgr/webpack > @svgr/plugin-svgo > svgo >
                css-select > nth-check

More info       https://github.com/advisories/GHSA-rp65-9cf3-cjxr
```

Solution

Requirements

1. Create a client
    a. All fields are required
2. Search for a client
    a. Searching in the "search field" should filter the list of clients by their first or last name
    b. Example: John Stevens and Steven Smith should both show up if a user searches "steven"
    c. Example: John Stevens should show up if a user searches "john"

Actual Work

- Finished item 2.
- Finished item 1
    - I started first by calling the create api
    - Then added the actual UI based in the mock design

Extras

- Quality and best practices. I think I have applied none
- How close to the designs is your submission? Maybe fairly close
    - If you needed to change something in the future (size/color of buttons), how easy would it be? Maybe not so easy but not hard too, given I do not have an actual experience in react I think I did fair in this exercise
    - How does this look on different devices? I made it a little responsive but may still need more work
- How can your submission allow for scalability? I may have not applied any
    - What if a customer has thousands of records? We can put pagination but I did not add it
    - How is state managed as the codebase grows? Based on what I've understood, React's useState is the best option for local state management. If you need a global state solution, the most popular ones are Redux, MobX, and the built-in Context API. Your choice will depend on the size of your project, your needs, and your engineers' expertise.
    - How can we support multiple countries? I've seen a solution wherein it uses the react-intl library
- How can you ensure the app behaves as you intend it to? Based on how I did, react aims to allow developers to create fast user interfaces for websites and applications alike easily.
- How intuitive is the behavior of the app? It may not be intuitive as when I finalized the index.tsx I just put all the components into 1 file where it can be put into multiple files.

Additional (Not in requirements and extras)

- Made it a little responsive but still needs more work.

Notes

When I saw the requirements I jumped right away with the changes for the UI. But I didn’t check that it was made in react and I have done mostly in Angular for the FE in my previous works. I just did trial and error and read through material-ui.