React has generally maintained a healthy lead on other Front-end frameworks. Learning it is helpful for a variety of reasons. An understanding of HTML & CSS is enough to get started.

A very quick and dirty introduction to what React does:

1. Enough support code that a full app can be built (framework, rather than library)
2. Pre-existing component library (you've seen Expo components if you've dug into the code for the mobile app at all; this is the parent of those) included
3. Companion libraries and the following useful features:
    a. Virtual DOMS
    b. JSX
    c. State management
    d. Independent reusable components
4. Uni-directional flow: child components do not automatically alter parents, but parent data can flow down to child components (this is very useful when tracking visual bugs).
5. Hooks and Redux (see state management) are used for application data that the entire app might use
6. Does not impose architecture pattern requirements on the program, unlike some other popular frameworks

Node.js has a command built in to build a react project and get started immediately (create-react-app).

UI
1: break down project functions into visual components based upon architecture requirements (React is flexible, use logical sense)
2: design components as independent sub-views (component per function intended? what views are useful?)
3: add components to navigation (nest and structure)

FRONTEND
1: set up state for components to interact using functions
2: set up ways for components to call code functions
3: design navigation from view to view (navigation is probably one of the hardest tasks to make robust and failure-proof; security is a concern here)

BACKEND
1: break down your design into components you want to have
2: design each component as its own system; while they can interweave, this should be at the data level, not at the code level (avoid codependency)
3: write code functions at the data level

This is less reliable at our project's design level given that we are coming up with the next things to work on, but we can break down smaller parts of the project this same way.

Resources:

1. https://www.w3schools.com/react/react_getstarted.asp
2. https://react.dev/reference/react