Agile

Iterative & incremental process of building:  
Have different versions to show the progress and with that you can check with your user to see what the actual question/problem is. Every iteration should be potentially shippable.

Git is a version control tool:

- add collaborators  
- local / your machine  
- remote / GitHub is remote  
- it uses branches (always a master, there are also feature branches etc)  
- commit  
- push to remote & pull from remote  
- merge / or pull request  
- the master branch should be the branch that **always** have to work  
- where to merge the stuff then what you are collabing on and see if it still works? You do that in a ‘development’ branch  
  
Branches:  
- a master branch  
- a development branch  
- a feature branch (or called a bugfix maybe for other reasons).

If you feel that it’s done, you do a pull request to the dev branch.  
- When someone checks it and it’s approved, it becomes part of the dev branch  
- you can then pull development, run it on your pc and if it still works, you pull it to the master

GitHub flow  
~ Cd desktop  
~ create-react-app filename  
make new repo on github

~ Git remote add origin SSH key github  
~ git push origin master  
  
now we need a development branch  
~ git checkout -b development  
~ git push origin development

Now they’re still the same, but we have two branches  
Now we’re going to change the default branch so that we don’t merge things into master because this was the default.

~ git pull origin development (always check if you have the latest version, that’s how you start on your feature)

~ git checkout -b chore/security-audit (making a new feature to fix the security)  
~ npm audit –fix  
~ git add .   
~ git commit -m “removed vulnerabilities”

~ git push origin chore/security (ggpush, push shortkey when on same branch)  
merge the new feature onto the branch and make a pull request to the branch and eventually to master  
~ git pull origin development (or feature etc.) (ggpull, pull this branch from origin)  
  
Work in small steps so that you can merge often, this is a good process.   
  
Project on GitHub  
Go into project, create one, do a template (basic kenban): this way you can split up tasks and its already inside the whole project (Trello can also be used, optional)

**Create a React app  
1.** console: Create-react-app set-file-name-here  
**2.** console: npm install redux react-redux  
**3.** Create store.js in a src folder  
**4.** Create a src folder (if it’s not there), components and a reducers folder  
**5.** console: npm run start (run app local)  
  
**• Create a store**import { createStore } from 'redux'

import reducer from './reducer'

const store = createStore(reducer)

export default store  
  
**• Add a reducer file in src**rxreducer (ES7 add on command) Or from the reader:

const reducer = (state = [], action = {}) => {

switch (action.type) {

default:

return state

}

}

export default reducer

**• Add a Provider (to tell React where it can find the Redux store) in src/index.js**import store from './store';

import {Provider} from 'react-redux;  
  
ReactDOM.render(

<Provider store={store}>

<App />

</Provider>,

document.getElementById('root')

)

**• Connect your store with the DevTools by adjusting store.js**import { createStore } from 'redux'

import reducer from './reducer'

const enhancer = window.\_\_REDUX\_DEVTOOLS\_EXTENSION\_\_ && window.\_\_REDUX\_DEVTOOLS\_EXTENSION\_\_()

const store = createStore(reducer, enhancer)

export default store