**TODO LIST:**

1. Being able to move the elements (books) around to different shelves (Currently reading, want to read, read)
2. When I click on “+”, I can search (regex)
3. Navigate back to the home page: react router history push replacing the state
4. Add book (if time permits): and have to choose which array to add to (concat?)

Finished:

1. Getting books from API into the books array
2. Now, create a method where you can both remove the book and add to a new state: Front-end.
3. Now, create a method where you can both remove the book and add to a new state: Back-end.

Keep the frame, but map each contact differently (horizontally): use <ul>

If state = shelf, then put it into the state

\_just write an update function, to update the shelf

\_ And then map it, the CSS will take care of it

\_3 components: The app page to handle all the display and route. The bookshelfs page which contain the shelves: 3 states (mapping all the shelves out) . book shelf page which handles the mapping of all books in 1 single shelf. Book page to handle the image, title etc & where the moving of the book is (select menu drop down, add value/ text)

+) And then a Search component

\_The title are basically a property

\_If I want to use .map, I have to use arrays

-Array of objects:

+) id: for key => key usually goes with id

+) title: to give out the title

+) filter: to see which book has that shelf

Can string

\_ What should go into Bookshelf?

+) Handle displaing of books

Todo more:

+) Do search: done by Fri night (work on it 2 hours tmr)

+) Re Route: Done by Sat

+) Format code into production code. Now the format looks messy

+) Start on REACT immediately by Friday, even if it’s just 30 mins.

\_What to do:

If there exists a query, then just normalize it: it takes both uppercase and lowercase

const match = new RegExp(escapeRegExp(this.state.query), 'i')

Then we’re gonna filter out from the books array

Check if that matches the query

If there’s no query, show showing books

Sort the books

Update qjuery, className, type= text, placeholder, value is the query. onChange: when someone puts something in, it update the query

Function update computer: trim() it

And clear query when you click on it

Then later display: If showingBooks.length isn’t the same, then (&&, show contacts <Now showing number of matches of {how many matches}, <span>

When I click, it shows all matches

///showingBooks maps out at bookgrid

format like the one in book.shelf