Server:

* API Routes:
* ~~Get clients~~
* Get client
* PUT (edit) client
* ~~Write a script that saves all the JSON data from github to DB~~
* ~~Open to cross-origin calls~~
* Database:
* ~~One collection called Clients~~

Client:

* ~~/Clients component~~ **~~/clients~~**
* ~~Display all clients in a table~~
* ~~Navbar Component~~
* ~~(navigates between routes)~~
* Actions component **/actions**
* Update section
* Add section
* This will be the only way a user can update a client’s email/owner/sale status
* Use a separate request for data that only returns client ID, name and owner.
* Show a loader as the request is happening
* Dashboard/Analytics component **/analytics**
* Make a new request to /clients data route
* Read more about this section in the lesson, use Recharts for the charts
* Update client component **popup**, use PUT requests
* Pops up when a specific client is clicked
* The owner drop-down should be generated off your data
* Email type can be hard coded
* **Transfer** button changes client owner
* Send buttonchange emailType by the users selection
* **Declare** button should change sold property to true
* If the user tries to carry out an action without filling the appropriate data, don’t make a request and alert a message
* Use axios to handle all requests

Make sure that the data for each chart is stored in its component's state - this probably means **you will have to access props inside of your constructor** - to do so, make sure your constructor looks like this:

constructor(props) {

**super**(props)

//...use props.whatever here

**More features (explained in lesson)**