

CheckMate V2

Tony Baker Project

Tip: Starting development

Tip: Fixing the issue when you fetch for record to edit and then type in a field and all of your data disappears from other fields

Tip: Check for open port

Tip: MongoDB Commands – Visual Studio Code

Tip: Creating custom routes with express js

Tip: How to grab fields from MongoDB and dynamically populate a select

Tip: Splitting the Service and deploying to GCP

Tip: Running the app on a mac (when apache is running and webpack is an issue)

Tip: Deploying to Heroku (SEE THE NOTES ON CHECKMATE V2 – FINALLY GOT IT TO WORK)

Tip: Removing options from a select control

Tip: Comparing both versions of CheckMate

Tip: Fixing this error: - options.allowedHosts[0] should be a non-empty string. – See Notes in V2

Tip: Fixing Macbook deploy issues. – See Notes in V2

Tip: Enabling Authorization – JWT and REDUX – hiding the navigation based on REDUX

Tip: Fixing the issue with warning messages from reactbootstrap next table

Tip: Fixing the issue with warning messages from reactbootstrap next table

When working with bootstrapnext table, if you get the controlid needs to be unique ..., the way you fix it is: (You have to make sure that each column has a unique datafield name), for the edit columns, just add a 1 and 2 for datafield) because we are not surfacing up any data for them anyway, we are formatting them with font awesome images.

```

client > src > components > Search > SearchPharma.js > columns
122 //npm install react-bootstrap-table --save
123 const columns = [
124   {
125     dataField: '_id',
126     text: '_id',
127     sort: true
128   },
129   {
130     dataField: 'Name',
131     text: 'Name',
132     sort: true
133   },
134   {
135     dataField: 'Phone',
136     text: 'Phone',
137   },
138   {
139     dataField: 'Email',
140     text: 'Email',
141   },
142   {
143     dataField: '1',
144     text: 'Edit',
145     formatter: CellFormatter,
146     style: { width: '10px' }
147   },
148   {
149     dataField: '2',
150     text: 'Delete',
151     formatter: CellFormatterDelete,
152     style: { width: '10px' }
153   },
154 ];
155
156 const unique_id = uuidv4();
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172

```

```

138   {
139     dataField: 'ManagerName',
140     text: 'ManagerName',
141   },
142   {
143     dataField: 'Email',
144     text: 'Email',
145   },
146   {
147     dataField: 'id',
148     text: 'Edit',
149     formatter: CellFormatter,
150     style: { width: '10px' }
151 },
152 {
153   dataField: 'id',
154   text: 'Delete',
155   formatter: CellFormatterDelete,
156   style: { width: '10px' }
157 },
158 ],
159
160
161
162
163
164
165
166
167
168
169
170
171
172

```

return (
 <div className="container">

 <h1>
 Search Reps
 </h1>

 <Card>
 <Card.Body>
 <Card.Title>Enter Search Criteria</Card.Title>
 <Form>

You also need to make sure this is correct as well

```

</Button>
<br><br>
<hr></hr>
<Row>
  <Col sm={12}>
    <h2>Search Results</h2>

    <BootstrapTable
      striped
      hover
      keyField="_id"
      data={tblSearchResults}
      columns={columns}

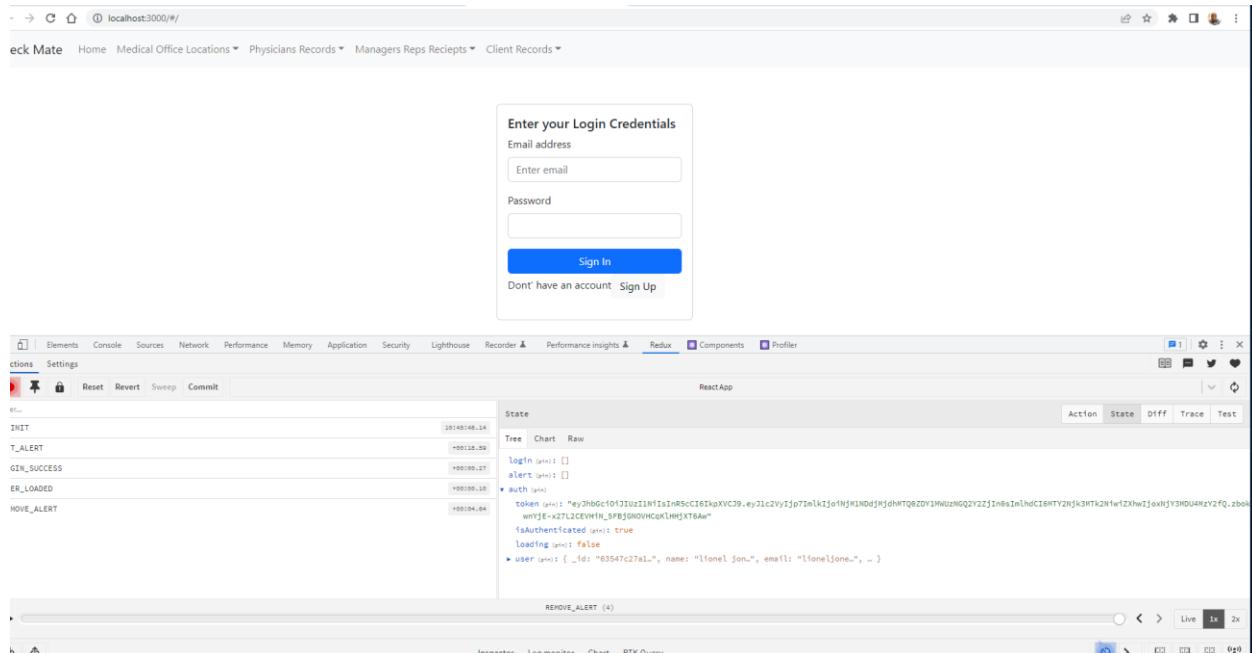
      pagination={paginationFactory({
        showTotal: true,
        firstPageText: "First",
        lastPageText: "Last",
      })}
    />

```

The key field needs to be the `_id` field (needs to be unique)

Tip: Enabling Authorization – JWT and REDUX – hiding the navigation based on REDUX

```
...  NavbarMain.js M X
client > src > NavbarMain > NavbarMain.js > NavbarMain
  1 import React, { Fragment } from 'react';
  2 import Nav from 'react-bootstrap/Nav';
  3 import { Navbar } from 'react-bootstrap';
  4 import { NavDropdown } from 'react-bootstrap';
  5 //import { withRouter } from 'react-router-dom';
  6 import { connect } from 'react-redux';
  7 import PropTypes from 'prop-types'
  8
  9
 10
 11 const NavbarMain = ({ auth, isAuthenticated }) => {
 12
 13   return (
 14     <div id="MasterContainer">
 15       <Fragment>
 16         {isAuthenticated !== null && isAuthenticated ? (
 17           <Navbar bg="light" expand="lg">...
 18           </Navbar>
 19         ) : (
 20           <p></p>
 21         )}
 22       </Fragment>
 23     </div>
 24   )
 25 }
 26
 27
 28 NavbarMain.propTypes = {
 29   isAuthenticated: PropTypes.bool
 30 }
 31
 32 const mapStateToProps = state => ({
 33   isAuthenticated: state.auth.isAuthenticated,
 34   auth: state.auth
 35 })
 36
 37 export default connect(mapStateToProps)(NavbarMain)
```



Tip: Comparing both versions of CheckMate

CheckMateV2 is my favorite:

Pros: It's a smaller bundle (it does not use more webpack (ver 5), it builds 200% faster

When you make changes, it reloads faster.

It does crash when you deploy to Heroku

It's official – Going with V2 – It deployed to Heroku – no errors and figuring out this culprit

```
2 app.use('/api/rep', require('./routes/api/rep'));
3 //http://localhost:5500/api/receipt
4 app.use('/api/receipt', require('./routes/api/receipt'));
5
6
7 //Server static assets in production
8 if(process.env.NODE_ENV === 'production')
9 {
10     //Set static folder (our public folder) - dist = VUE JS/ build = react
11     app.use(express.static('client/build'));
12     app.get('*', (req, res) => {
13         res.sendFile(path.resolve(__dirname, 'client', 'build', 'index.html'));
14     })
15 }
16
17
18 //HEROKU LOOKS AT THE process.env.PORT
```

```
server.js
37 //h
38 app
39 //h
40 app
41 //h
42 app
43 //h
44 app
45
46
47 //Se
48 if(r
49 {
50
51
52
53
54
55
56 }
57
58
```

Cons:

It has a huge set of warnings with bootstraptablenext (but it still works tho)

I will more than likely go with CheckMateV2 for the pass-off (It will be a lot easier for the devs to work with in terms of deployment)

Tip: Fixing this error: - options.allowedHosts[0] should be a non-empty string.

Tip: Fixing Macbook deploy issues

Tip: Comparing both versions of CheckMate

Tip: Running VSCode from command in terminal on mac m1

Tip: Using Google Meet for meetings

Tip: Getting the solution run in a docker container

Tip: Deploying to Heroku (Got it to work)

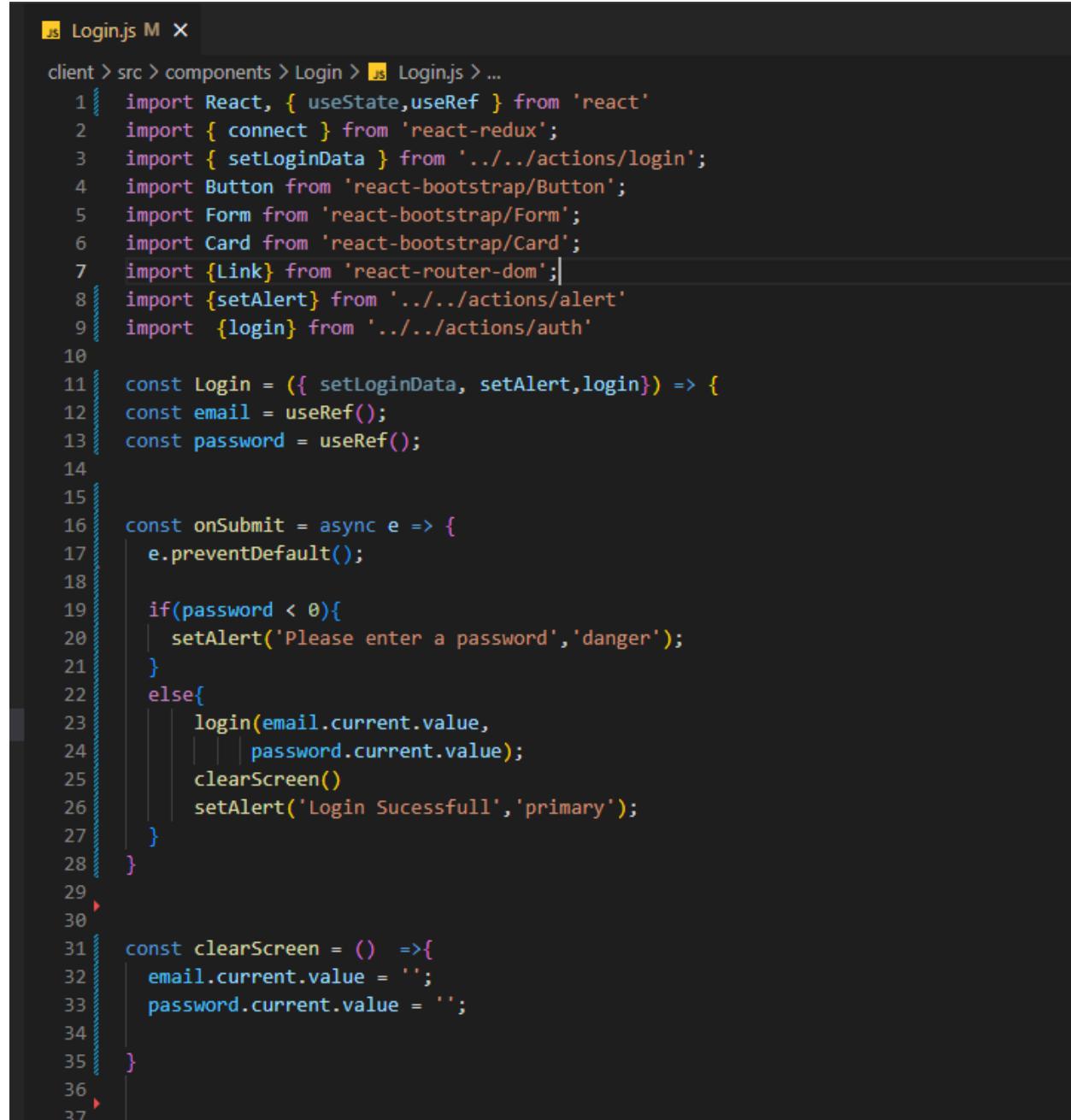
Tip: Splitting out the webservice to Heroku

Tip: Adding styles to the login screen

Tip: Using Controlled components

Tip: Using uncontrolled components

Sometimes react acts weird with controlled components, in this case we can use uncontrolled components



```
client > src > components > Login > Login.js > ...
1 import React, { useState, useRef } from 'react'
2 import { connect } from 'react-redux';
3 import { setLoginData } from '../../../../../actions/login';
4 import Button from 'react-bootstrap/Button';
5 import Form from 'react-bootstrap/Form';
6 import Card from 'react-bootstrap/Card';
7 import { Link } from 'react-router-dom';
8 import { setAlert } from '../../../../../actions/alert';
9 import { login } from '../../../../../actions/auth';

10
11 const Login = ({ setLoginData, setAlert, login }) => {
12   const email = useRef();
13   const password = useRef();

14
15   const onSubmit = async e => {
16     e.preventDefault();
17
18     if(password.current.value < 0){
19       setAlert('Please enter a password', 'danger');
20     }
21     else{
22       login(email.current.value,
23             password.current.value);
24       clearScreen()
25       setAlert('Login Sucessfull', 'primary');
26     }
27   }
28 }

29
30
31 const clearScreen = () =>{
32   email.current.value = '';
33   password.current.value = '';
34
35 }
```

```

return (
  <div className="container">
    <br />
    <h1>
      </h1>
    <br />
    <Card className="d-flex justify-content-center" style={{ width: '300px', position:'absolute',left:'40%'}} >
      <Card.Body>
        <Card.Title>Enter your Login Credentials</Card.Title>
        <Form
          onSubmit={(e) => {
            e.preventDefault();
            onSubmit(e)
          }}
        >
          <Form.Group className="mb-3" >
            <Form.Label>Email address</Form.Label>
            <Form.Control type="input"
              placeholder="Enter email"
              ref={email}
            />
          </Form.Group>

          <Form.Group className="mb-3" >
            <Form.Label>Password</Form.Label>
            <Form.Control
              type="password"
              ref={password}
            />
          </Form.Group>

          <Button variant="primary" type="submit" style={myStyles.buttonCustomStyle}>
            Sign In
          </Button>
        </Form>
        <p>
          Dont' have an account
          <Link to='/Register' className="btn btn-light">
            Sign Up
          <...>
        </p>
      </Card.Body>
    </Card>
  </div>
)

```

<https://shrouded-gorge-59241.herokuapp.com/#/>

Tip: Adding styles to the login screen

Look at the style for the card, this was used to center using absolute and flex

```

</h1>
<br />
<Card className="d-flex justify-content-center" style={{ width: '300px', position: 'absolute', left: '40%' }} >
  <Card.Body>
    <Card.Title>Enter your Login Credentials</Card.Title>
    <Form
      onSubmit={(e) => {
        e.preventDefault();
        login(e);
        setLoginData(userName, _password);
        setUserName('');
        setPassword('');
      }}
    >
      <Form.Group>
        <Form.Control type="text" placeholder="User Name" onChange={(e) => setUserName(e.target.value)} />
      </Form.Group>
      <Form.Group>
        <Form.Control type="password" placeholder="Password" onChange={(e) => setPassword(e.target.value)} />
      </Form.Group>
      <Button variant="primary" type="submit" style={myStyles.buttonCustomStyle}>
        Sign In
      </Button>
    </Form>
    <p>      p00149021@houstonisd.org Mag17615@7, yesterday • first commit ...
      Dont' have an account
      <Link to='/Register' className="btn btn-light">
        Sign Up
      </Link>
    </p>
  </Card.Body>
</Card>
</div>
);
}

const myStyles = {
  buttonPadLeft: {
    marginLeft: '2px'
  },
  smallerTextFields: {
    width: '300px',
  },
  buttonCustomStyle: {
    width: '100%'
  }
};

const mapStateToProps = state => ({

```

For the button, we use width: 100%

Position: absolute relative to the screen size, and it works with the left attribute (40% of the screen)
 d-flex justify.... Is from bootstrap 5- see the docs

```

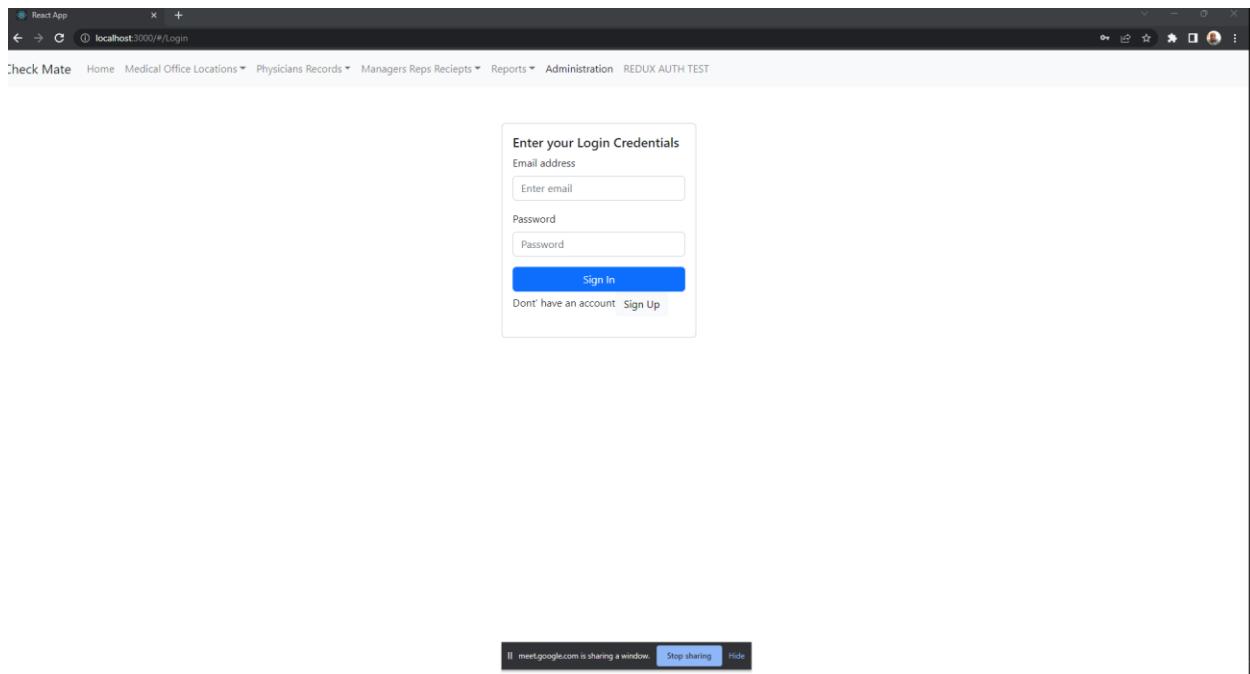
<Form.Group>
  <Form.Control type="text" placeholder="User Name" onChange={(e) => setUserName(e.target.value)} />
</Form.Group>
<Form.Group>
  <Form.Control type="password" placeholder="Password" onChange={(e) => setPassword(e.target.value)} />
</Form.Group>
<Button variant="primary" type="submit" style={myStyles.buttonCustomStyle}>
  Sign In
</Button>
</Form>
<p>      p00149021@houstonisd.org Mag17615@7, yesterday • first commit ...
  Dont' have an account
  <Link to='/Register' className="btn btn-light">
    Sign Up
  </Link>
</p>
</Card.Body>
</Card>
</div>
);

}

const myStyles = {
  buttonPadLeft: {
    marginLeft: '2px'
  },
  smallerTextFields: {
    width: '300px',
  },
  buttonCustomStyle: {
    width: '100%'
  }
};

const mapStateToProps = state => ({

```



Tip: Splitting out the webservice to Heroku

Web service URL

<https://pure-gorge-49930.herokuapp.com/>

Modify your server.js

```
server.js
27 //primary CheckMate Routes
28 //http://localhost:5500/api/users
29 app.use('/api/users', require('./routes/api/users'));
30 //http://localhost:5500/api/auth
31
32 app.use('/api/auth', require('./routes/api/auth'));
33 //http://localhost:5500/api/pharma
34 app.use('/api/pharma', require('./routes/api/pharma'));
35 //http://localhost:5500/api/medicalGroup
36 app.use('/api/medicalGroup', require('./routes/api/medicalGroup'));
37 //http://localhost:5500/api/physicians
38 app.use('/api/physicians', require('./routes/api/physicians'));
39 //http://localhost:5500/api/manager
40 app.use('/api/manager', require('./routes/api/manager'));
41 //http://localhost:5500/api/rep
42 app.use('/api/rep', require('./routes/api/rep'));
43 //http://localhost:5500/api/receipt
44 app.use('/api/receipt', require('./routes/api/receipt'));
45
46
47 /*
48 //Server static assets in production
49 if(process.env.NODE_ENV === 'production')
50 {
51   //Set static folder (our public folder) - dist = VUE JS/ build = react
52   app.use(express.static('client/build'));
53   app.get('*',(req,res) => {
54     res.sendFile(path.resolve(__dirname,'client','build','index.html'));
55   })
56 }
57 */
58 // You, 1 second ago * Uncommitted changes
59
60 //HEROKU LOOKS AT THE process.env.PORT
61 const PORT = process.env.PORT || 5500;
62
63 app.listen(PORT, () => console.log(`Server started on port ${PORT}`));
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER GITLENS

remote: If you have apps using any of these resources, you must upgrade to paid plans by this d
gram by the end of September. Learn more at <https://blog.heroku.com/next-chapter>

And your package.json

```

{
  "name": "checkmateservice",
  "version": "1.0.0",
  "description": "Check Mate Service (2022)",
  "main": "server.js",
  "scripts": {
    "start": "node server"
  },
  "author": "Lionel Jones",
  "license": "MIT",
  "dependencies": {
    "bcryptjs": "^2.4.3",
    "config": "^3.3.7",
    "cors": "^2.8.5",
    "express": "^4.18.1",
    "express-validator": "^6.14.2",
    "gravatar": "^1.8.2",
    "jsonwebtoken": "^8.5.1",
    "mongoose": "^6.4.6",
    "mongoose-uniform-validator": "^3.1.0",
    "request": "^2.88.2"
  },
  "devDependencies": {
    "concurrently": "7.5.0",
    "nodemon": "2.0.20"
  }
}

```

<https://pure-gorge-49930.herokuapp.com/api/pharma/searchPharmaRecord>

POST https://pure-gorge-49930.herokuapp.com/api/pharma/searchPharmaRecord

Body (Pretty)

```

1 {
2   ...
3   "Name": "Memorial Health",
4   ...
5   "Email": "mmhealth@gmail.com",
6   ...
7   "SearchType": "Name"
8
9
10
11
12
13

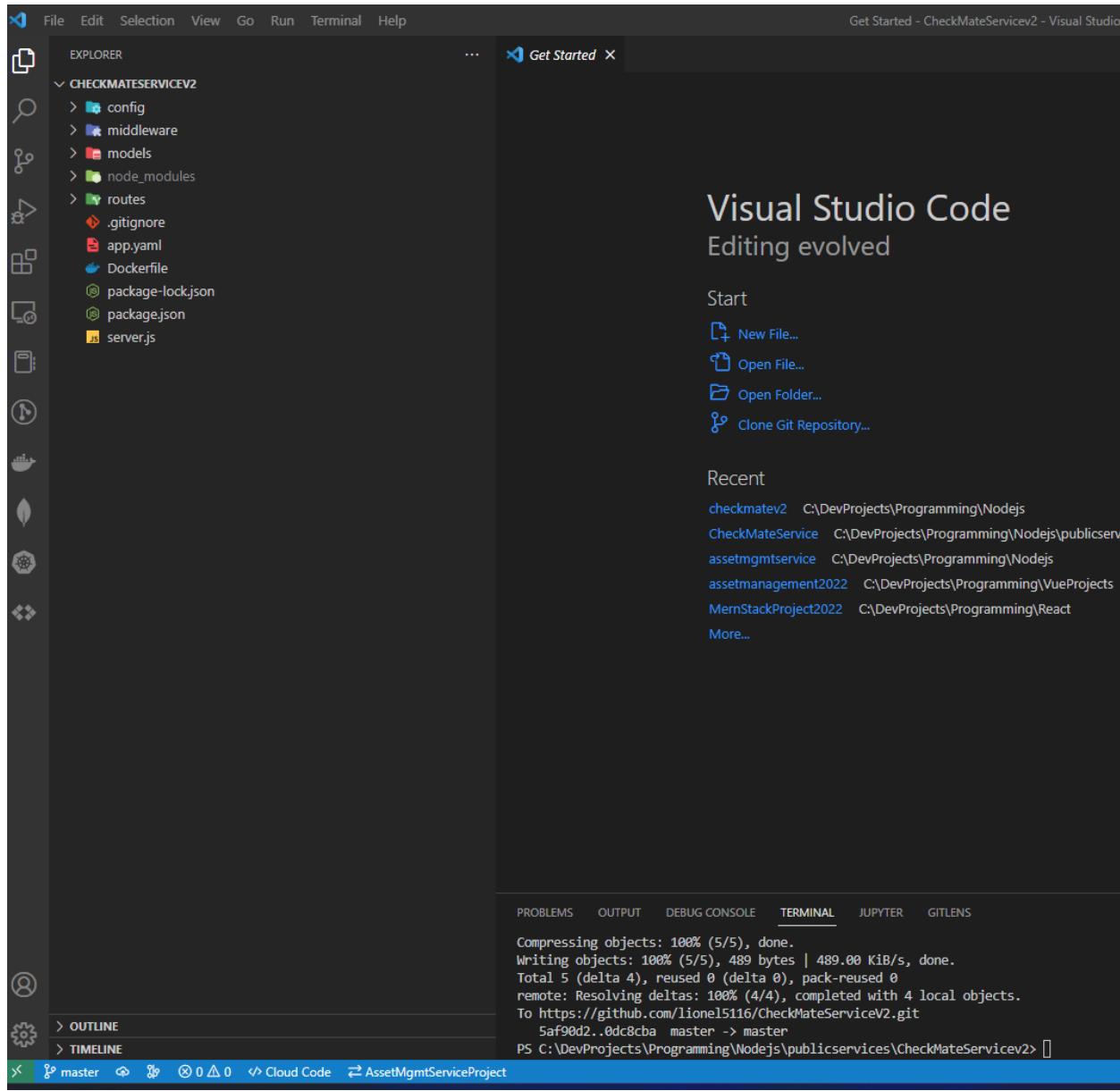
```

Body (JSON)

```

1 [
2   {
3     "_id": "63549190fe74bf00a27dcfa5",
4     "Name": "Memorial Health",
5     "Phone": "713-654-0987",
6     "Email": "mmhealth@gmail.com",
7     "Password": "1234567",
8     "Notes": "Good Pharma Company and I will stay employed with them",
9     "__v": 0
10   }
11 ]
12
13

```



Placed it in the project and it works

The screenshot shows a browser window with two tabs. The active tab is 'React App' at 'localhost:3000/#/SearchPharma'. It displays a search form with fields for 'Enter Search Criteria' (containing 'All'), 'Email' (containing 'Enter Email'), and 'Name' (containing 'Name'). A blue button labeled 'Search Records' is visible. Below the form is a section titled 'Search Results' containing two rows of data:

_id	Name	Phone	Email	Edit	Delete
63549190fe74bf00a27dcfa5	Memorial Health	713-654-0987	mmhealth@gmail.com		
6354a0dd5e97ead84326278	Herman Hospital System	713-876-0987	corsonmemorial@gmail.com		

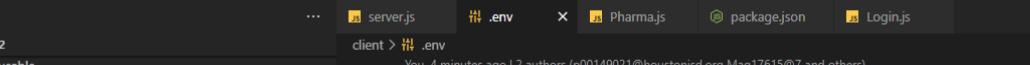
Below the table, a message says 'Showing rows 1 to 2 of 2' with a page number '1'.

The developer tools console in the top right shows several warnings from React regarding the 'key' prop:

- Warning: `control` is ignored on `FormControls` when `id` is specified.
- Warning: `control` is ignored on `FormControls` when `id` is specified.
- Warning: `control` is ignored on `FormControls` when `id` is specified.
- Warning: `control` is ignored on `FormControls` when `id` is specified.
- Warning: `control` is ignored on `FormControls` when `id` is specified.
- Warning: `control` is ignored on `FormControls` when `id` is specified.
- Warning: Encountered two children with the same key, 'id'. Keys should be unique so that components maintain their identity across updates. Non-unique keys may cause children to be duplicated and/or omitted.

At the bottom of the console, there is a note: 'Check the render method of "Body". See https://reactjs.org/warnings-keys for more information.'

Changed url to point to the service on Heroku



.env - checkmatev2 - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

client > .env

```
You, 4 minutes ago | 2 authors (p00149021@houstanisd.org Mag17615@7 and others)
1 https://cstroman.medium.com/using-environment-variables-with-react-webpack-c7a04bbf833e
2 REACT_APP_API_KEY = 123-123-123-123
3 REACT_APP_API_BASE =
4 #using port 5500
5 #REACT_APP_SERVICE_URL = 'http://localhost:5500/api'
6 #dont need the api for the heroku deployed service
7 REACT_APP_SERVICE_URL = 'https://pure-gorge-49930.herokuapp.com/api' You, 4 minutes ago *
```

And it works!!!!

The screenshot shows a web browser window with three tabs: 'lione5116/CheckMateServiceV2', 'shrouded-gorge-59241 | Heroku', and 'React App'. The main content area has a header with 'Check Mate' and navigation links: Home, Medical Office Locations, Physicians Records, Managers Reps Receipts, Reports, Administration, and REDUX AUTH TEST. Below the header is a section titled 'Search Pharmaceutical Companies'.

Enter Search Criteria

All

Email Name

Enter Email Name

Search Records

Search Results

_id	Name	Phone	Email	Edit	Delete
63549190fe74bf00a27dcfa5	Memorial Health	713-654-0987	mmhealth@gmail.com		
6354a04dd5e97ead84326278	Herman Hospital System	713-876-0987	corsonmemorial@gmail.com		

Showing rows 1 to 2 of 2

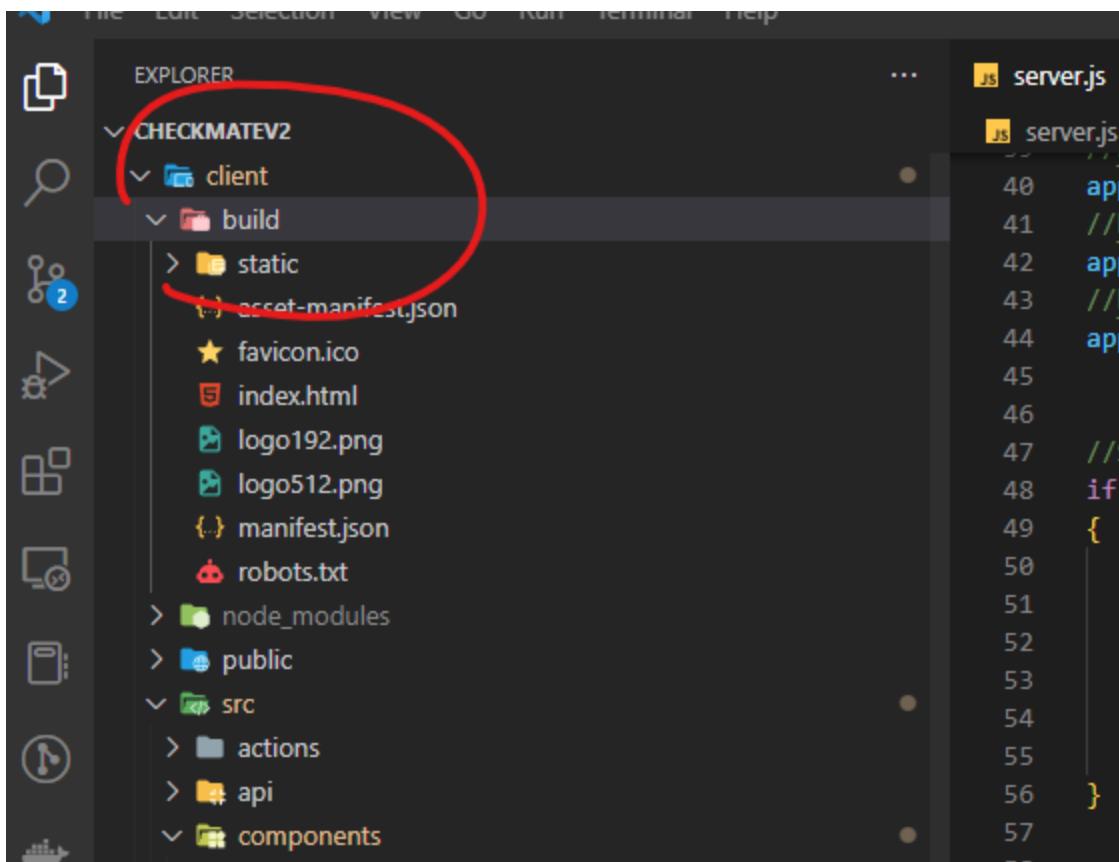
1

Tip: Deploying to Heroku (Got it to work)

Deploying to Heroku is super easy!!!

Most important step, do a build first to see where what your build location is:

npm run build



This determines lines 48 and 51 below

I made this mistake before, the /dist folder is when you are building a VUEJS app, like my assetmgmt2022 creates **VERY IMPORTANT !!!!!!!**

```

40 app.use('/api/manager', require('./routes/api/manager'));
41 //http://localhost:5500/api/rep
42 app.use('/api/rep', require('./routes/api/rep'));
43 //http://localhost:5500/api/receipt
44 app.use('/api/receipt', require('./routes/api/receipt'));

45

46

47 //Server static assets in production
48 if(process.env.NODE_ENV === 'production')
49 {
50     //Set static folder (our public folder) - dist = VUE JS/ build = react
51     app.use(express.static('client/build'));
52     app.get('*', (req, res) => {
53         res.sendFile(path.resolve(__dirname, 'client', 'build', 'index.html'));
54     })
55 }
56
57
58
59 //HEROKU LOOKS AT THE process.env.PORT

```

Next you do:

Heroku login

Heroku create

It creates a repo

Go to deploy on the dashboard

Install the Heroku CLI

Download and install the [Heroku CLI](#).

If you haven't already, log in to your Heroku account and follow the prompts to create a new SSH public key.

```
$ heroku login
```

Clone the repository

Use Git to clone shrouded-gorge-59241's source code to your local machine.

```
$ heroku git:clone -a shrouded-gorge-59241
$ cd shrouded-gorge-59241
```

Deploy your changes

Make some changes to the code you just cloned and deploy them to Heroku using Git.

```
$ git add .
$ git commit -am "make it better"
$ git push heroku master
```

Follow the instructions, do a

heroku git:remote -a shrouded-gorge-59241

```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
```

```
PS C:\DevProjects\Programming\Nodejs\checkmatev2> heroku git:remote -a shrouded-gorge-59241
set git remote heroku to https://git.heroku.com/shrouded-gorge-59241.git
```

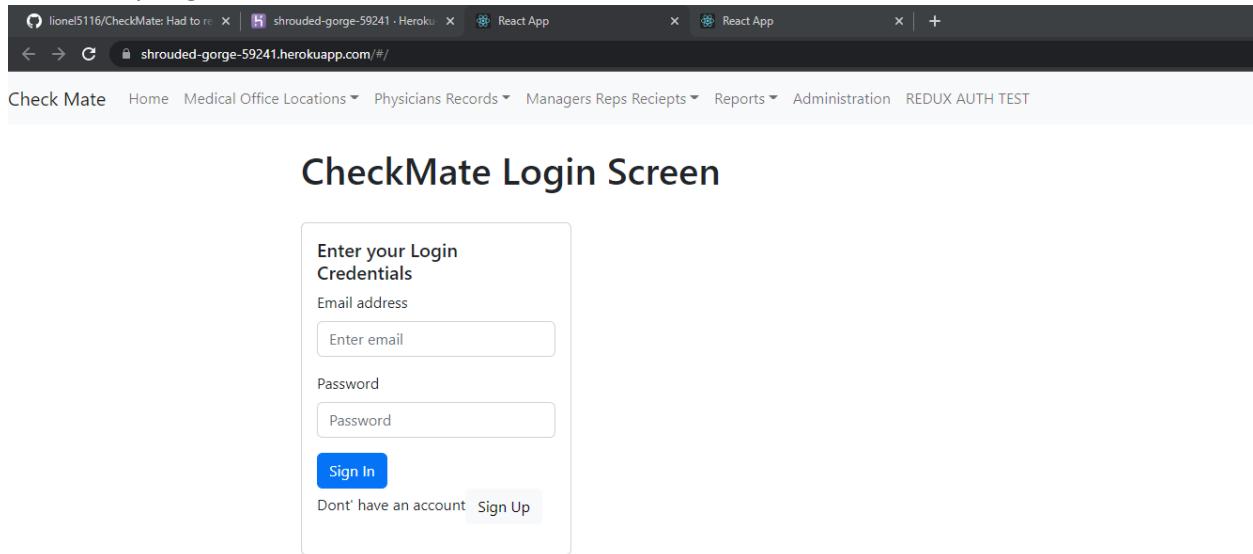
Then do a

git add .

git commit -am "make it better"

git push heroku master

And there you go

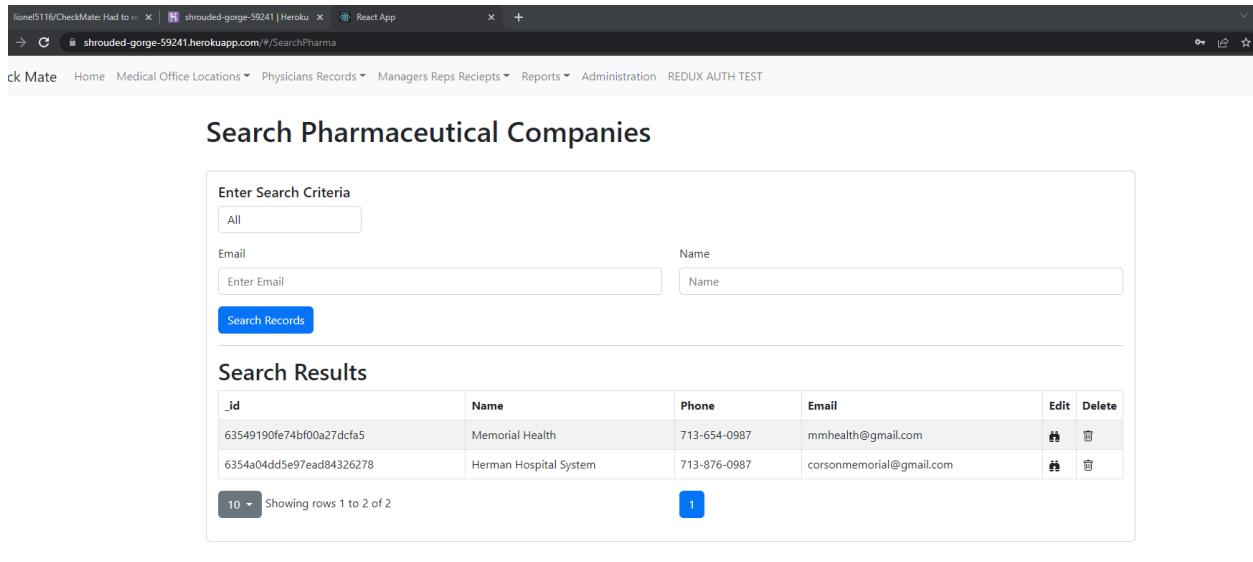


A screenshot of a web browser showing the 'CheckMate Login Screen'. The title bar says 'Check Mate' and has links for Home, Medical Office Locations, Physicians Records, Managers Reps Receipts, Reports, Administration, and REDUX AUTH TEST. The main content area has a form titled 'Enter your Login Credentials' with fields for Email address and Password, and a 'Sign In' button. Below the form are links for 'Dont' have an account' and 'Sign Up'.

<https://shrouded-gorge-59241.herokuapp.com/#/>

Ahd it WORKS!!!!

It took time for the service to start (I hit the endpoint to quick)



A screenshot of a web browser showing the 'Search Pharmaceutical Companies' screen. The title bar and navigation links are identical to the login screen. The main content area has a search form with fields for Email and Name, and a 'Search Records' button. Below the form is a table titled 'Search Results' with columns for _id, Name, Phone, Email, Edit, and Delete. Two rows of data are shown: 'Memorial Health' and 'Herman Hospital System'. At the bottom, there is a pagination control showing 'Showing rows 1 to 2 of 2' and a page number '1'.

Tip: Getting the solution run in a docker container

The screenshot shows a code editor interface with a dark theme. On the left is the Explorer sidebar, which lists a project named 'CHECKMATEV2' containing a 'client' folder, 'public', 'src' (with '.env', '.gitignore', 'package-lock.json', 'package.json', and 'README.md'), 'config', 'middleware', 'models', 'node_modules', 'routes' (with '.gitignore' and 'CheckMateV2.docx'), 'Dockerfile', 'package-lock.json', 'package.json', and 'server.js'. The 'Dockerfile' is selected and highlighted in the sidebar. The main panel displays the content of the 'Dockerfile':

```
1 #!/bin/bash -c docker build -t checkmateappv2.latest .
2 #A
3 FROM node:16
4
5 #copy over main server file
6 #B
7 COPY server.js /server.js
8
9 COPY package*.json ./
10
11 #E
12 EXPOSE 5500 3000
13
14 #C copy directories for client
15 COPY client/ /client
16
17 #copy directories for service
18 #C
19 COPY config/ /config
20 COPY middleware/ /middleware
21 COPY models/ /models
22 COPY routes/ /routes
23
24
25 #F npm install on server
26 RUN npm install
27 |
28 #F npm install on the client
29 RUN npm install --force --prefix client
30 RUN npm run build --prefix client
31
32 #D
33 #ENTRYPOINT ["npm", "start"]
34 #ENTRYPOINT ["node", "server.js"]
35 ENTRYPOINT ["npm", "run", "dev"]
36
37
38 #A The base image to build upon
```

Notice line 13, this is how you expose both ports

3000 is for the client app

5500 is for the NodeJS app

Line 16 is copying the client folder's contents

line 30 ad 31 runs the install and build of the react app

line 36 runs your custom command – (npm run dev)

To run the image

```
● ● ●
lioneljones@MacBookAirM1 ~ % docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
checkmateappv2  latest   7a5366aa0d04  33 seconds ago  1.4GB
assetmgmtsvc   latest   5ed50c8e47cb  12 days ago   902MB
lioneljones@MacBookAirM1 ~ % docker run --name checkmateappv2-container -p 3000:3000 -p 5500:5500 -d checkmateappv2
d50b54f5f3428acc28cd53e4fc14534d7b97778a5dc5835cae3ea19cbb9f76d
lioneljones@MacBookAirM1 ~ % docker container ps
CONTAINER ID  IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES
lioneljones@MacBookAirM1 ~ % docker run --name checkmateappv2-container -p 3000:3000 -p 5500:5500 -d checkmateappv2
869ace6b2ca1a57fca5f030c99434a1dc5775f2b6cc255a65edd4a9fb07363e1
lioneljones@MacBookAirM1 ~ % docker run --name checkmateappv2-container -p 3000:3000 -p 5500:5500 -d checkmateappv2
6ee37b18d2f490e5fa36ee801df687a9fd4d16e74f2cbc1e17c0ed135ee62276
lioneljones@MacBookAirM1 ~ %
```

To run the image

```
docker run --name checkmateappv2-container -p 3000:3000 -p 5500:5500 -d checkmateappv2
```

The screenshot shows the Docker web interface with the following details:

- Containers** [Give Feedback](#)
- A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. [Learn more](#)
- Showing 1 items
- Search bar: Search
- Table headers: NAME, IMAGE, STATUS, PORT(S), STARTED, ACTIONS
- Table data:

	NAME	IMAGE	STATUS	PORT(S)	STARTED	ACTIONS
<input type="checkbox"/>	 checkmateappv2-container 6ee37b18d2f4	checkmateappv2:latest	Running	3000,5...	2 minutes ago	<input type="button"/> <input type="button"/> <input type="button"/>

And it works

localhost:3000/#/SearchPharma

Check Mate Home Medical Office Locations ▾ Physicians Records ▾ Managers Reps Receipts ▾ Reports ▾ Administration REDUX AUTH TEST

Search Pharmaceutical Companies

Enter Search Criteria

Email
Name

Search Results

_id	Name	Phone	Email	Edit	Delete
63549190fe74bf00a27dcfa5	Memorial Health	713-654-0987	mmhealth@gmail.com		
6354a04dd5e97ead84326278	Herman Hospital System	713-876-0987	corsonmemorial@gmail.com		

10 ▾ Showing rows 1 to 2 of 2
1

To run your image

```
docker run --name checkmateappv2-container -p 3000:3000 -p 5500:5500 -d
checkmateappv2
```

```
docker inspect checkmateappv2-container
```

To grab the id only

```
docker inspect --format="{{.Id}} checkmateappv2-container
```

But I get an error (need to research this)

To remove your image

```
docker rm 6ee37b18d2f490e5fa36ee801df687a9fd4d16e74f2cbc1e17c0ed135ee62276
```

Below is the end user process

If you already have a container that you ran. Then you stopped and restarted your machine. The container maintains its id and configuration. You will see it below

The screenshot shows the Docker Desktop interface for managing images. At the top, it displays "Images on disk" with a refresh icon, "Last refresh: less than a minute ago", "2 images", "2.31 GB total size", "1.4 GB / 2.31 GB in use", and a "Clean up" button. Below this, the "Images" section has tabs for "LOCAL" and "REMOTE REPOSITORIES". A search bar and a checkbox for "In use only" are present. The table lists two images:

NAME	TAG	IMAGE ID	CREATED	SIZE
assetmgmtsvc	latest	5ed50c8e47cb	12 days ago	902.43 MB
checkmateappv2	IN USE	62b427cecc78	about 2 hours ago	1.4 GB

The screenshot shows the Docker Desktop interface for managing containers. At the top, it displays "Containers" with a refresh icon, "Docker Desktop", "Upgrade plan", and a "Sign in" button. Below this, the "Containers" section has tabs for "Containers" and "Give Feedback". A message states: "A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. [Learn more](#)". The table lists one container:

NAME	IMAGE	STATUS	PORT(S)	STARTED	ACTIONS
checkmateappv2-container 61911b4c8195	checkmateappv2:latest	Exited (255)	3000,5...		

Just hit the start button

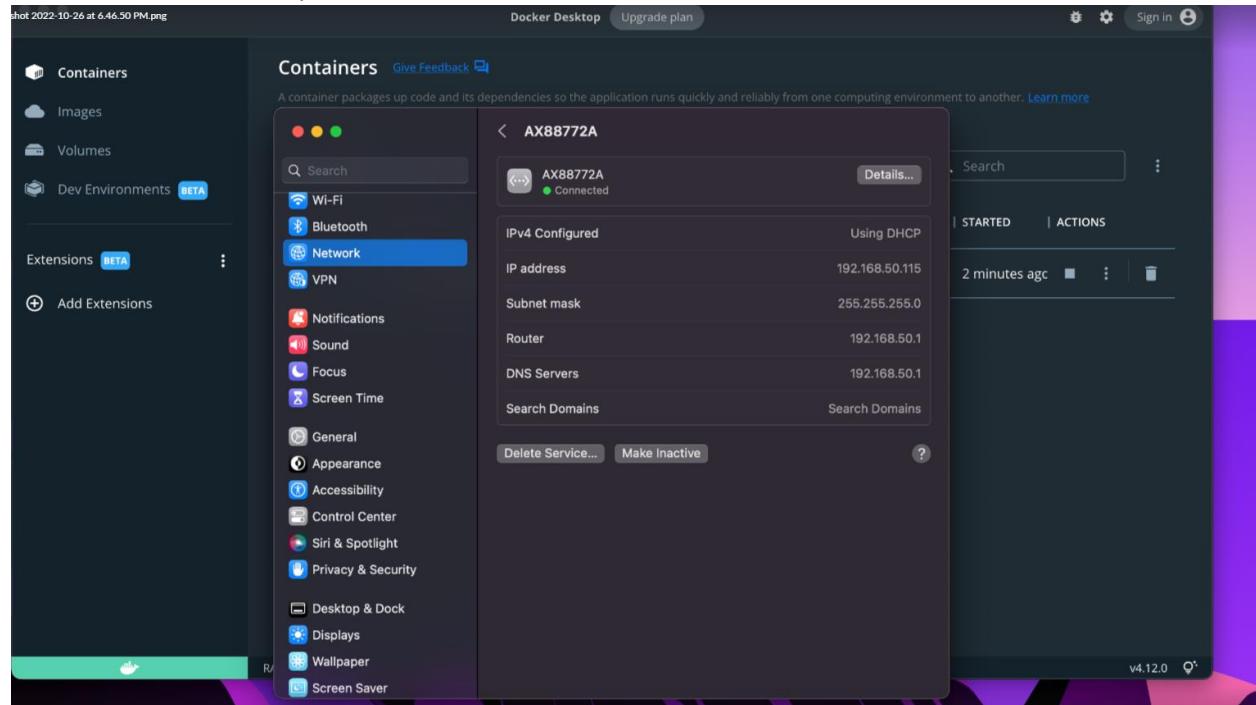
The screenshot shows the Docker Desktop interface for managing containers after starting the previously listed container. The container status has changed from "Exited (255)" to "Running". The table now shows:

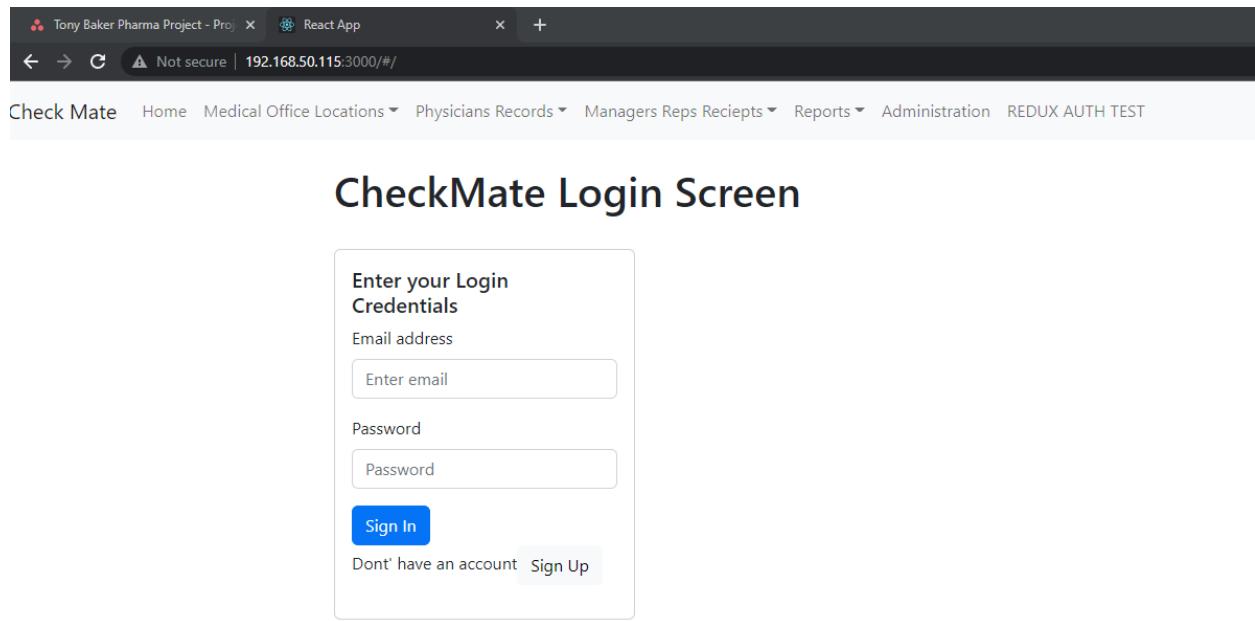
NAME	IMAGE	STATUS	PORT(S)	STARTED	ACTIONS
checkmateappv2-container 61911b4c8195	checkmateappv2:latest	Running	3000,5...	13 seconds ago	

And as you can see below, it still has its settings

```
lioneljones@MacBookAirM1 ~ % docker container ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
6191b4c8195 checkmateappv2 "npm run dev" 28 minutes ago Up 21 seconds 0.0.0.0:3000->3000/tcp, 0.0.0.0:5500->5500/tcp checkmateappv2-container
lioneljones@MacBookAirM1 ~ %
```

Then I did a test from my MAC at home:





But it's looking at localhost
Now if you wanted to use the service
This works:

POST http://localhost:5500/api/pharma/searchPharmaRecord

POST https://shrouded-gorge

POST http://192.168.50.115:5500/api/pharma/searchPharmaRecord

http://192.168.50.115:5500/api/pharma/searchPharmaRecord

POST http://192.168.50.115:5500/api/pharma/searchPharmaRecord

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {  
2   "Name": "Memorial Health",  
3   "Email": "mmhealth@gmail.com",  
4   "SearchType": "Name"  
5 }
```

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 {  
2   "pharma": [  
3     {  
4       "_id": "63549190fe74bf00a27dcfa5",  
5       "Name": "Memorial Health",  
6       "Phone": "713-654-0987",  
7       "Email": "mmhealth@gmail.com",  
8       "Password": "1234567",  
9       "Notes": "Good Pharma Company and I will stay employed with them",  
10      "__v": 0  
11    }  
12  ]  
13 }
```

But the app itself points to localhost (meaning that it thinks it's looking itself)

Tip: Using Google Meet for meetings

Google meet rocks.. it's free and much easier to use than teams, and it's completely free

<https://play.google.com/store/apps/details?id=co...> ::

Google Meet (original) - Apps on Google Play

Securely connect, collaborate, and celebrate from anywhere. With **Google Meet**, everyone can safely create and join high-quality video meetings for groups of ...

★★★★★ Rating: 4.1 · 2,207,524 votes · Free · Android · Business/Productivity

Videos ::



How to Use Google Meet - Detailed Tutorial

YouTube · Teacher's Tech
Mar 24, 2020

Tip: Running VSCode from command in terminal on mac m1

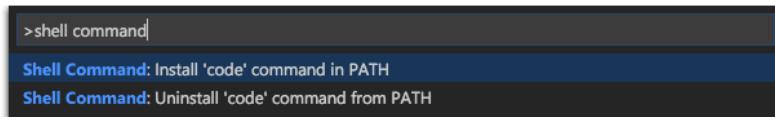
<https://code.visualstudio.com/docs/setup/mac#:~:text>You%20can%20also%20run%20VS,code'%20command%20in%20PATH%20command.>

- [Tweet this link](#)
- [Subscribe](#)
- [Ask questions](#)
- [Follow @code](#)
- [Request features](#)
- [Report issues](#)
- [Watch videos](#)

Launching from the command line

You can also run VS Code from the terminal by typing 'code' after adding it to the path:

- Launch VS Code.
- Open the Command Palette (**Cmd+Shift+P**) and type 'shell command' to find the Shell Command:
Install 'code' command in PATH command.



- Restart the terminal for the new **\$PATH** value to take effect. You'll be able to type 'code .' in any folder to start editing files in that folder.

Tip: Comparing both versions of CheckMate

CheckMateV2 is my favorite:

Pros: It's a smaller bundle (it does not use more webpack (ver 5), it builds 200% faster

When you make changes, it reloads faster.

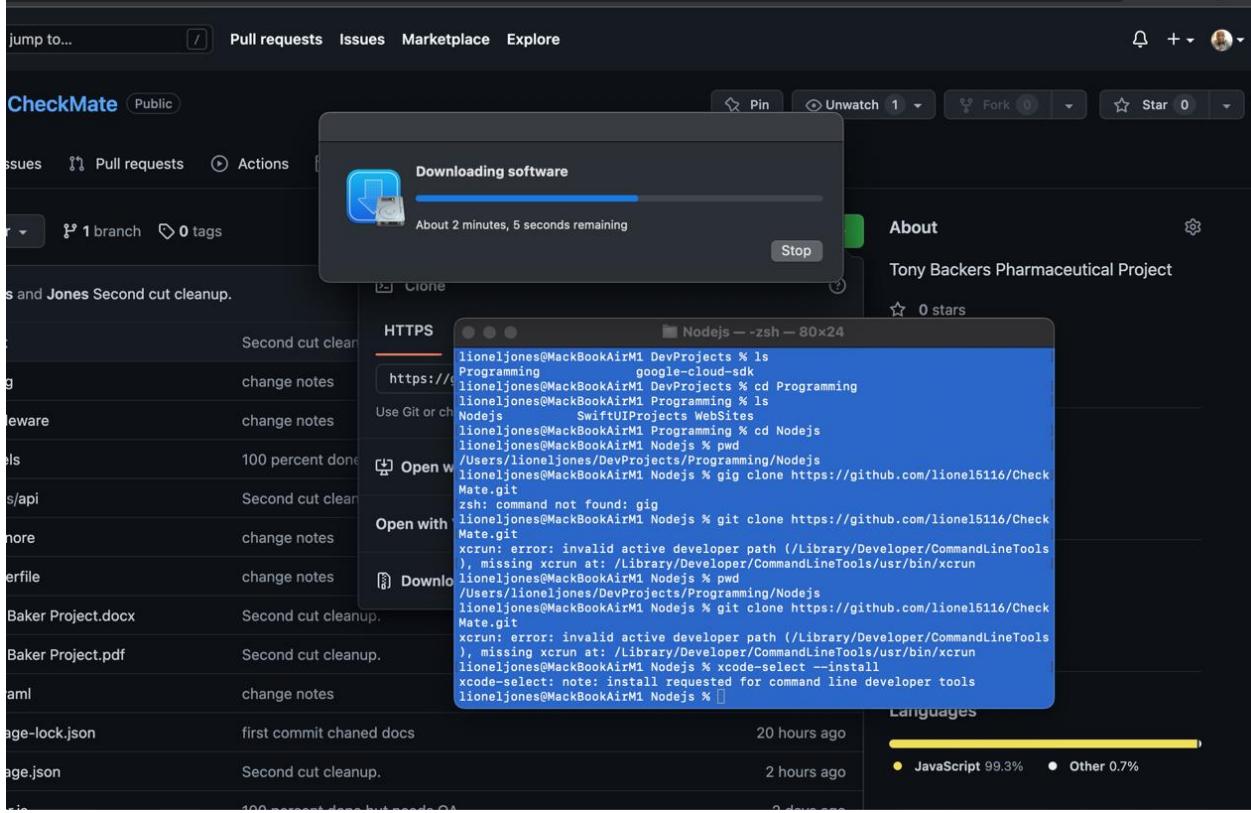
It does crash when you deploy to Heroku

Cons:

It has a huge set of warnings with bootstraptablenext (but it still works tho)

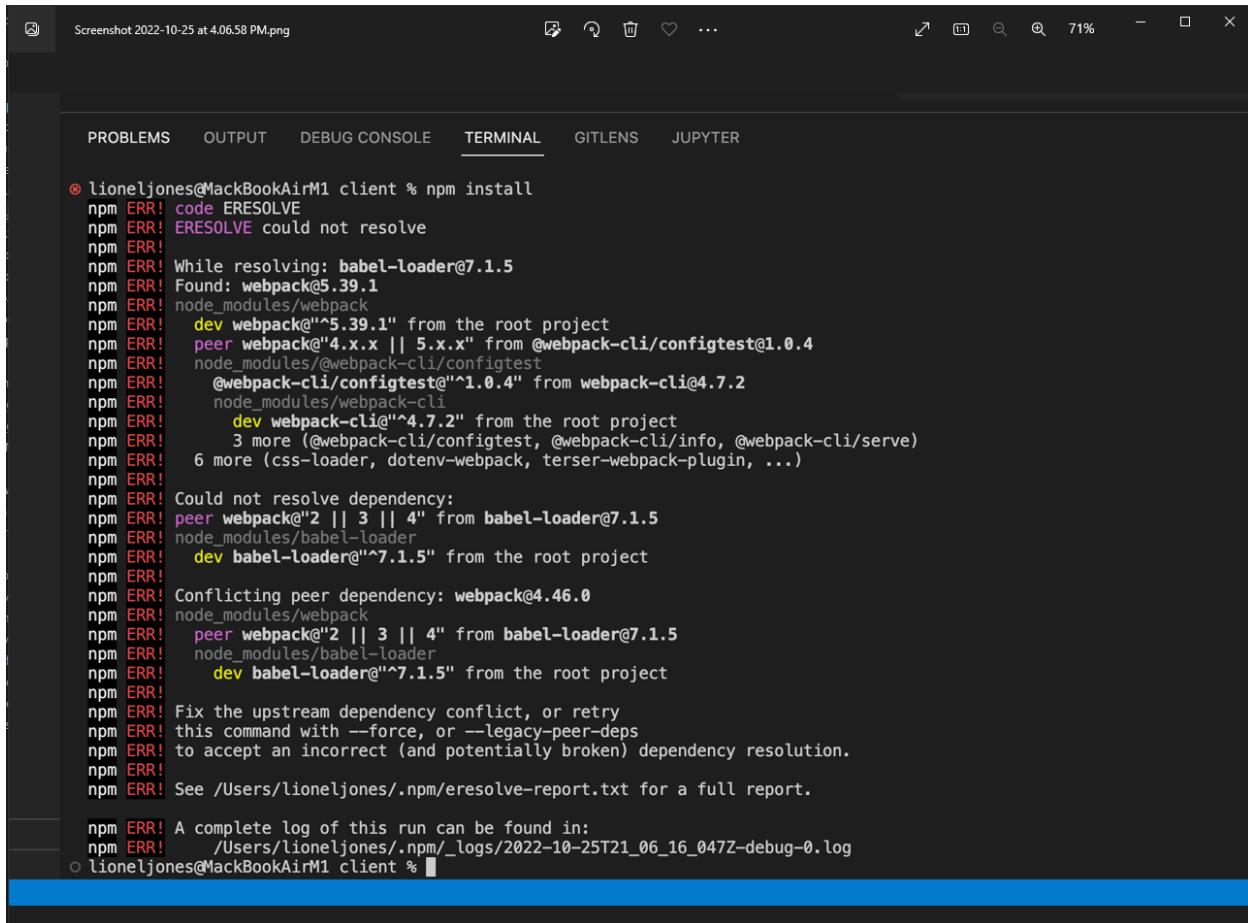
Tip: Fixing Macbook deploy issues

For the error below when trying to clone the repo



I had to run the command to update my xcode tools
xcode-select –install

This is common everywhere



Screenshot 2022-10-25 at 4.06.58 PM.png

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS JUPYTER

```
① lioneljones@MacBookAirM1 client % npm install
npm ERR! code ERESOLVE
npm ERR! ERESOLVE could not resolve
npm ERR!
npm ERR! While resolving: babel-loader@7.1.5
npm ERR! Found: webpack@5.39.1
npm ERR! node_modules/webpack
npm ERR!   dev webpack@"^5.39.1" from the root project
npm ERR!   peer webpack@"4.x.x || 5.x.x" from @webpack-cli/configtest@1.0.4
npm ERR!   node_modules/@webpack-cli/configtest
npm ERR!     @webpack-cli/configtest@"1.0.4" from webpack-cli@4.7.2
npm ERR!     node_modules/webpack-cli
npm ERR!       dev webpack-cli@"^4.7.2" from the root project
npm ERR!       3 more (@webpack-cli/configtest, @webpack-cli/info, @webpack-cli/serve)
npm ERR!     6 more (css-loader, dotenv-webpack, terser-webpack-plugin, ...)
npm ERR!
npm ERR! Could not resolve dependency:
npm ERR!   peer webpack@"2 || 3 || 4" from babel-loader@7.1.5
npm ERR!   node_modules/babel-loader
npm ERR!     dev babel-loader@"^7.1.5" from the root project
npm ERR!
npm ERR! Conflicting peer dependency: webpack@4.46.0
npm ERR!   node_modules/webpack
npm ERR!     peer webpack@"2 || 3 || 4" from babel-loader@7.1.5
npm ERR!     node_modules/babel-loader
npm ERR!       dev babel-loader@"^7.1.5" from the root project
npm ERR!
npm ERR! Fix the upstream dependency conflict, or retry
npm ERR! this command with --force, or --legacy-peer-deps
npm ERR! to accept an incorrect (and potentially broken) dependency resolution.
npm ERR!
npm ERR! See /Users/lioneljones/.npm/eresolve-report.txt for a full report.

npm ERR! A complete log of this run can be found in:
npm ERR!   /Users/lioneljones/.npm/_logs/2022-10-25T21_06_16_047Z-debug-0.log
○ lioneljones@MacBookAirM1 client %
```

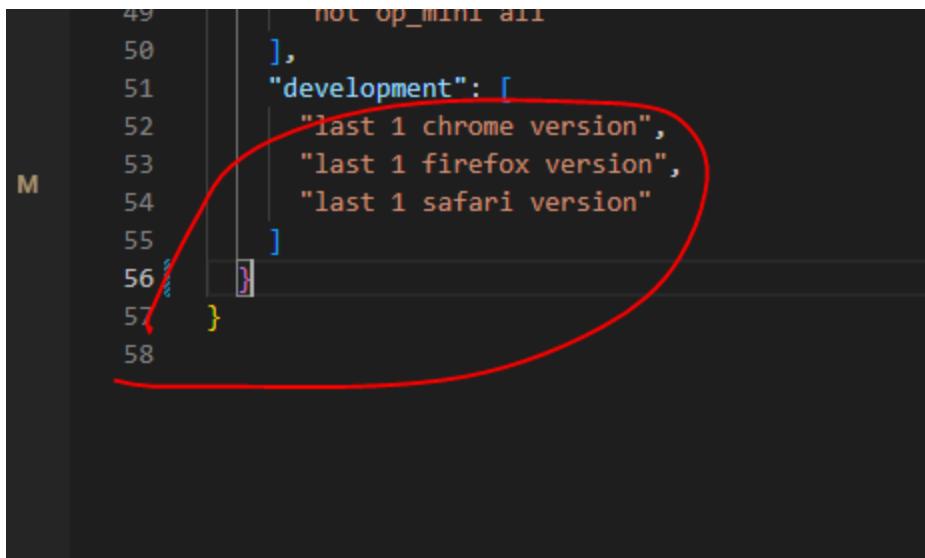
see the message

npm install -force

I have to use this everywhere, even in my deploy scripts for Heroku

Tip: Fixing this error: - options.allowedHosts[0] should be a non-empty string

While doing npm run start, I was getting ERRORS , no matter what I did. The way I fixed it was to remove the proxy entry from the package.json file



```
49      not op_mini all
50  ],
51  "development": [
52    "last 1 chrome version",
53    "last 1 firefox version",
54    "last 1 safari version"
55  ]
56 }
57 }
58 }
```

I am not use a proxy anymore anyway.

I decided to create another version of the app without webpack – causing too many issues

<https://reactjs.org/docs/create-a-new-react-app.html>

<https://react-bootstrap.github.io/getting-started/introduction/>

npm install react-bootstrap bootstrap

In Index.js

index.js - checkmatev2 - Visual Studio Code

src > index.js > ...

You, 1 second ago | 2 authors (p00149021@houstonisd.org Mag17615@7 and others)

```
1 import 'bootstrap/dist/css/bootstrap.min.css';
2 import 'bootstrap/dist/js/bootstrap.bundle.min';
3 import React from 'react';
4 import ReactDOM from 'react-dom/client';
5 import './index.css';
6 import App from './App';
7 import reportWebVitals from './reportWebVitals';

8

9 const root = ReactDOM.createRoot(document.getElementById('root'));
10 root.render(
11   <React.StrictMode>
12     <App />
13   </React.StrictMode>
14 );
15

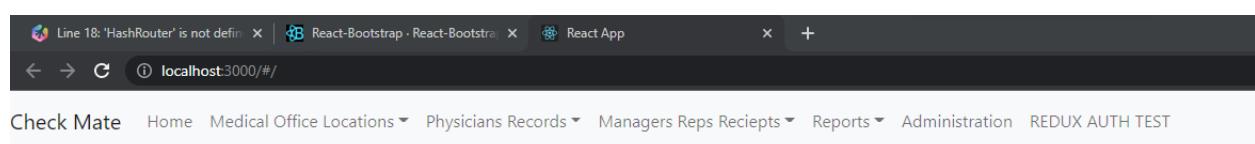
16 // If you want to start measuring performance in your app, pass a function
17 // to log results (for example: reportWebVitals(console.log))
18 // or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
19 reportWebVitals();
20
```

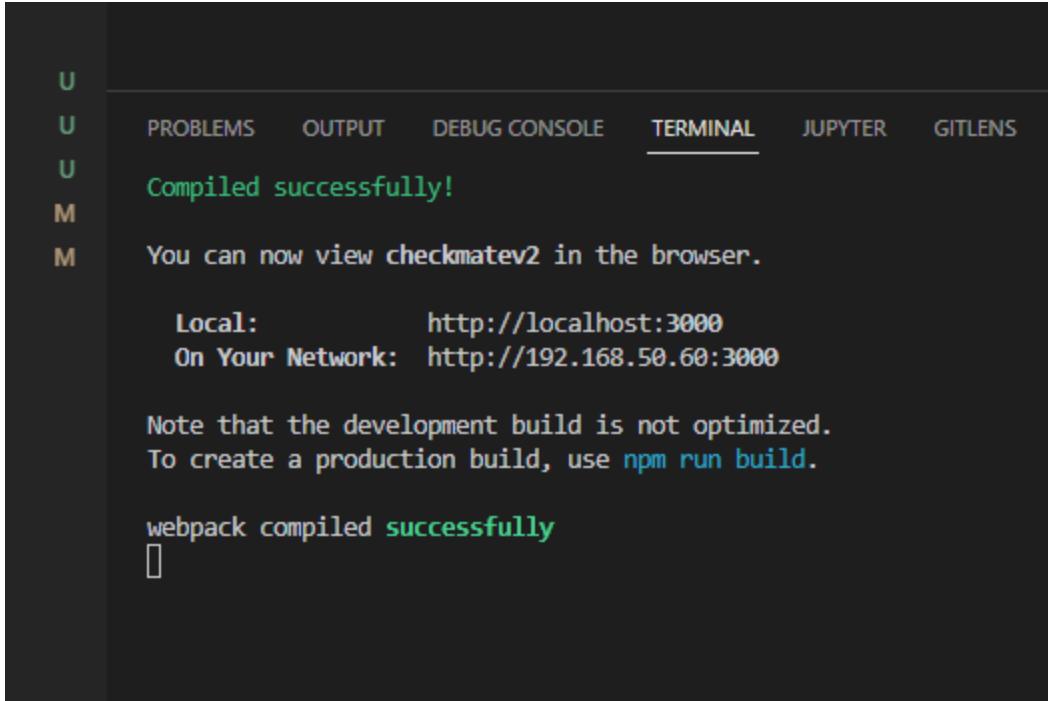
And so far, no warnings

```
npm install react-router-dom --save
```

```
npm install react-router-dom@5.2.0 react-router@5.2.0
```

And this is all we need to do to get the initial bootstrap to work





The screenshot shows the VS Code interface with the terminal tab selected. The terminal output is as follows:

```
U
U   PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    JUPYTER    GITLENS
U
M   Compiled successfully!
M
M   You can now view checkmatev2 in the browser.
M
      Local:          http://localhost:3000
      On Your Network:  http://192.168.50.60:3000
M
      Note that the development build is not optimized.
      To create a production build, use npm run build.
M
      webpack compiled successfully
[]
```

npm i axios

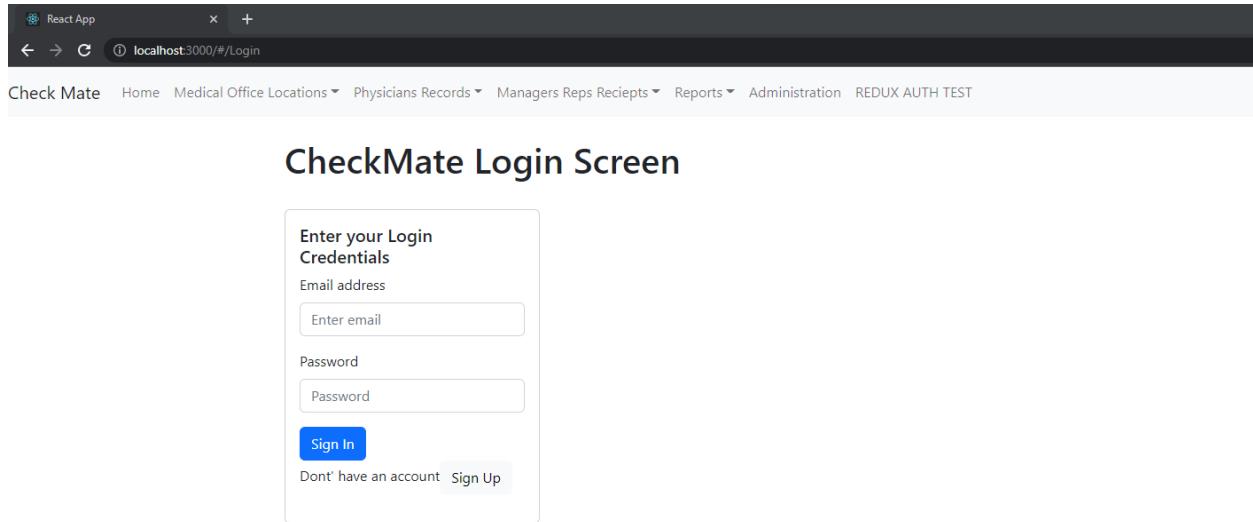
npm i redux-devtools-extension

Install all of this for REDUX

npm i react-router-dom redux react-redux redux-thunk moment react-moment

And after adding REDUX, it still builds fast

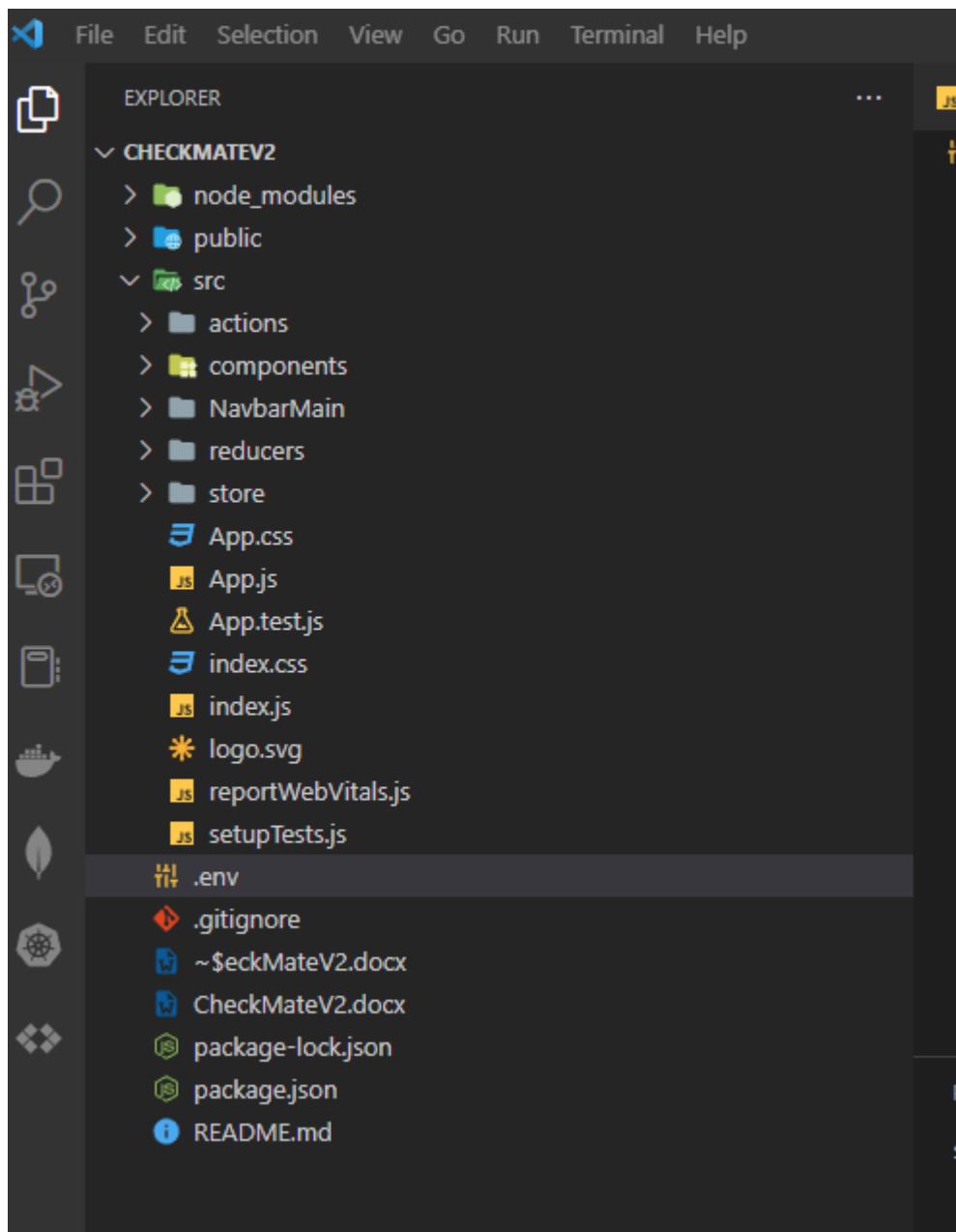
I have the base code:



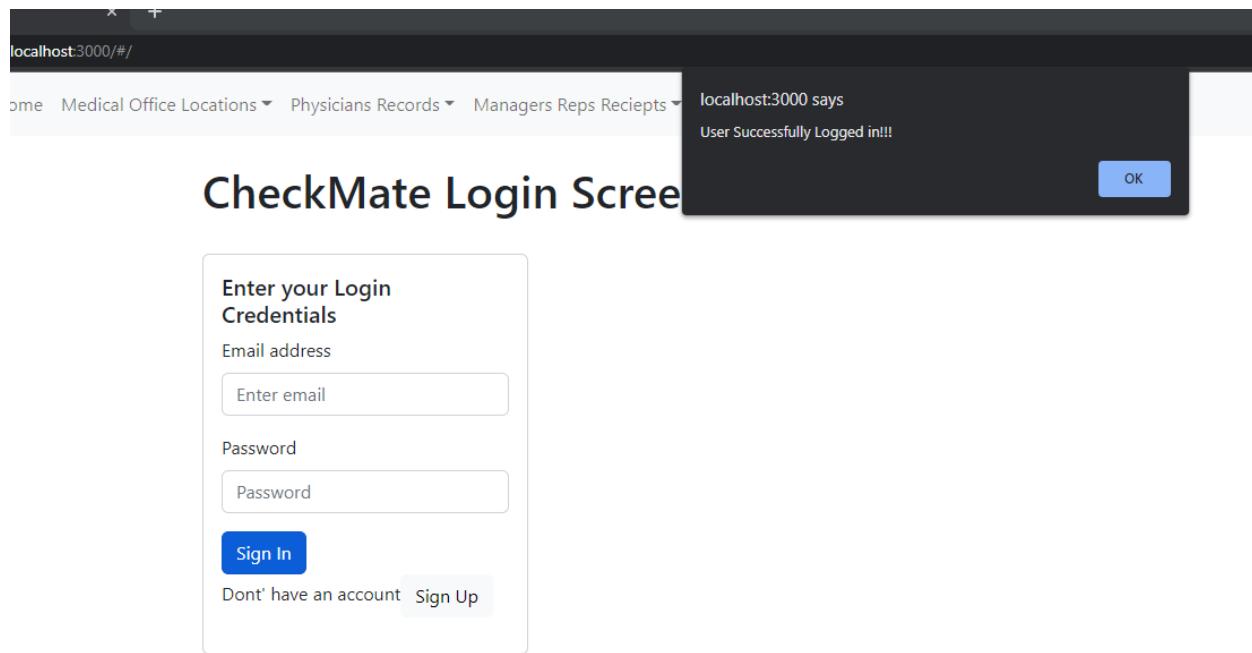
And its lightning fast

To allow for environment variables
npm i env-cmd

Make sure you put it at the root



So the login screen and register screen is finished



For the bootstrap tables, I had to use the –force with it

```
10  "bootstrap": "^5.2.2",
11  "env-cmd": "^10.1.0",
12  "moment": "^2.29.4",
13  "react": "^18.2.0",
14  "react-bootstrap": "^2.5.0",
15  "react-bootstrap-table-next": "^4.0.3",
16  "react-bootstrap-table2-filter": "^1.3.3",
17  "react-bootstrap-table2-paginator": "^2.1.2",
18  "react-bootstrap-table2-toolkit": "^2.1.3",
19  "react-dom": "^18.2.0",
20  "react-moment": "^1.1.2",
```

i:e npm install react-bootstrap-table2-toolkit –force
npm i uuid

Tip: Removing options from a select control

```
}

async function getManagerFirstAndLastName() {
    let _SEARCH_DATA = [];
    _SEARCH_DATA = await fetchManagerFirstAndLastName();
    console.log(_SEARCH_DATA)

    var _ddMangers = document.getElementById('ManagerName');
    removeOptions( _ddMangers )

    _ddMangers.options[_ddMangers.options.length] = new Option('--Select Manager--');
    for(const key in _SEARCH_DATA) {
        _ddMangers.options[_ddMangers.options.length] = new Option(_SEARCH_DATA[key].FirstName + ' ' + _SEARCH_DATA[key].LastName);
    }
}

function removeOptions(selectElement) {
    var i, L = selectElement.options.length - 1;
    for(i = L; i >= 0; i--) {
        selectElement.remove(i);
    }
}
```

Tip: Deploying to Heroku

These steps are followed from my other two deployments

We modify our first three files as shown below:

```
default.json
config> default.json > mongoURI
1 {
2     "mongoURI" : "mongodb+srv://lione15116:Mag17615%40@cluster0.jwcnt.mongodb.net/Checkmate?retryWrites=true&w=majority"
3 }
4 }

package.json
{
  "description": "Check Mate Service (2022)",
  "main": "server.js",
  "scripts": {
    "start": "node server",
    "server": "nodemon server",
    "client": "npm start --prefix client",
    "dev": "concurrently \"npm run server\" \"npm run client\"",
    "heroku-postbuild": "NPM_CONFIG_PRODUCTION=false npm install --prefix client && npm run build --prefix client"
  },
  "author": "Lionel Jones",
  "license": "MIT",
  "dependencies": {
    "bcryptjs": "^2.4.3",
    "config": "^3.3.7",
    "cors": "^2.8.5",
    "express": "^4.18.1",
  }
}

.gitignore
# config/default.json
node_modules/
node_modules/
```

Update (You have to use the –force) in the package.json, or Heroku will error out

```
SearchManager.js M Rep.js M package.json Receipt.js M
package.json > {} dependencies
1 {
2   "name": "checkmateservice",
3   "version": "1.0.0",
4   "description": "Check Mate Service (2022)",
5   "main": "server.js",
6   "scripts": {
7     "start": "node server",
8     "server": "nodemon server",
9     "client": "npm start --prefix client",
10    "dev": "concurrently \"npm run server\" \"npm run client\"",
11    "heroku-postbuild": "NPM_CONFIG_PRODUCTION=false npm install --force --prefix client && npm run build --prefix client"
12  },
13  "author": "Lionel Jones",
14  "license": "MIT",
15  "dependencies": [
16    "bcryptjs": "^2.4.3",
17    "config": "^3.3.7",
18    "cors": "^2.8.5",
19    "express": "^4.18.1",
20    "express-validator": "^6.14.2",
21    "gravatar": "^1.8.2",
22    "jsonwebtoken": "^8.5.1",
23    "mongoose": "^6.4.6",
24    "mongoose-unique-validator": "^3.1.0",
25    "request": "^2.88.2"
26  ],
27  "devDependencies": {
28    "concurrently": "^7.5.0",
29    "nodemon": "^2.0.20"
30  }
31}
```

default.json

.gitignore

package.json

See your AssetManagement2022 for the correct content

The Heroku tag we entered tells Heroku once we push it to Heroku to run `npm install` to install dependencies for each project and then run `npm run build` to build the project, and when it build the project, it will put everything in the client folder of our project.

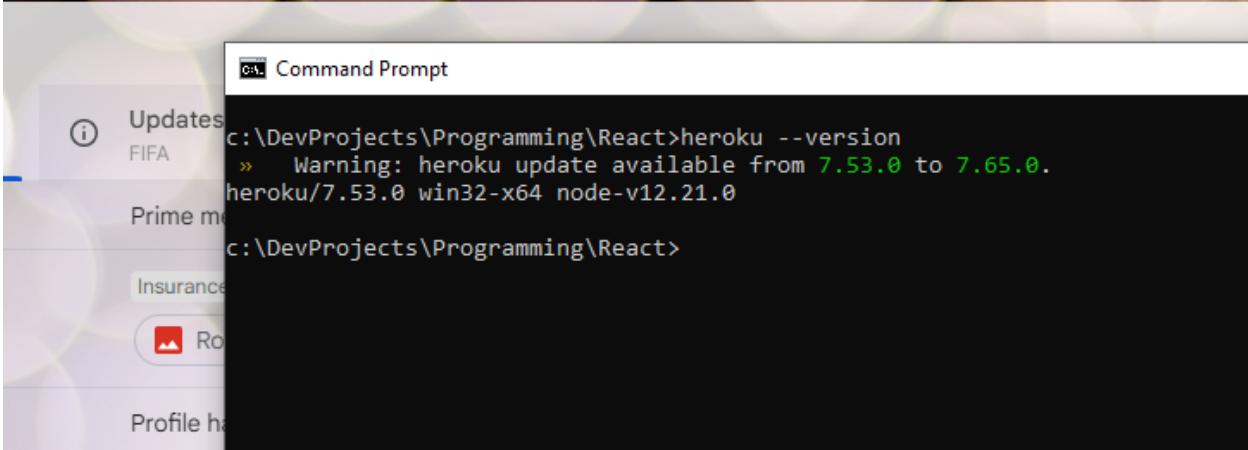
Then in your server.js file:

```
42 app.use('/api/rep', require('./routes/api/rep'));
43 //http://localhost:5500/api/receipt
44 app.use('/api/receipt', require('./routes/api/receipt'));
45
46
47 //Server static assets in production
48 if(process.env.NODE_ENV === 'production')
49 {
50     //Set static folder (our public folder)
51     app.use(express.static('client/dist'));
52     app.get('*', (req, res) => {
53         res.sendFile(path.resolve(__dirname, 'client', 'dist', 'index.html'));
54     })
55 }
56
57
58 //HEROKU LOOKS AT THE process.env.PORT
59 const PORT = process.env.PORT || 5500;
60
```

Next at the terminal, do a Heroku login

Heroku login

Check your version



Next create an app

Heroku create

If you go to the dashboard, you will see the app

fierce-plains-03819

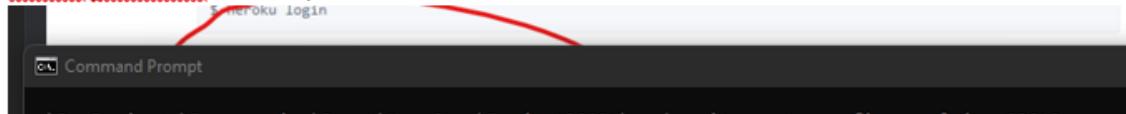
Next, go to your dashboard, click on your app
Go to the deploy tab

Go to the deploy tab

The screenshot shows the Heroku dashboard with the 'Deploy' tab selected. At the top, there are three buttons: 'Heroku Git' (with 'Use Heroku CLI'), 'GitHub' (with 'Connect to GitHub'), and 'Container Registry' (with 'Use Heroku CLI'). Below these buttons, there is a section titled 'Install the Heroku CLI' with a link to download it. A red circle highlights the 'heroku login' command in the terminal section. The terminal also shows commands for creating a new Git repository and initializing it, followed by a heroku git:remote command. A red circle highlights the 'heroku git:remote -a fierce-plains-03819' command. Below the terminal, there is a section titled 'Deploy your application' with instructions to commit code to a repository and push it to Heroku. A red circle highlights the 'heroku git:remote -a fierce-plains-03819' command again. A note at the bottom says: 'You can now change your main deploy branch from "master" to "main" for both manual and automatic deploys, please follow the instructions [here](#)'.

copy the remote command
heroku git:remote -a fierce-plains-03819

heroku git:remote -a fierce-plains-03819



Make sure you in your directory

A screenshot of a terminal window in a code editor. The terminal tab is selected at the top. The code in the terminal is:

```
56  }
57
58
59 //HEROKU LOOKS AT THE process.env.PORT
60 const PORT = process.env.PORT || 5500;
61
62 app.listen(PORT, () => console.log(`Server started on port ${PORT}`));
```

The output shows several messages from webpack and npm, indicating a warning and a successful exit. The command history is restored. The terminal window is running on Windows PowerShell.

Then type in the line from here:

A screenshot of a Heroku deployment guide. A red circle highlights the command line area where the user is instructed to type:

```
$ cd my-project/
$ git init
$ heroku git:remote -a fierce-plains-03819
```

Below the command line, there is a section titled "Deploy your application" with instructions to commit code to a repository and push it to Heroku. Another red circle highlights the "git" command line area.

A screenshot of a terminal window. A red circle highlights the command line area where the user runs:

```
heroku git:remote -a fierce-plains-03819
```

Below the command line, the terminal shows the output of the command, including a warning about the Heroku update available and the URL for the remote repository. A red circle highlights the "remote" part of the command line.

This adds a remote repo for Heroku

Then do all of your regular commits locally

```
git add -A  
git add .  
git commit -m "first commit"  
git push origin master
```

Now push to Heroku

```
git push Heroku master
```

After that, watch the build, it should succeed.

You can go to your Heroku dashboard and view your app and browse it.

Tip: Running the app on a mac (when apache is running and webpack is an issue)

On a mac, if you do a npm install on the client folder, you will get a npm run webpack error,
If you look at the error, it will tell you what command to run to get npm install to run
npm install -f (or npm install -force)

For Apache – The default port for Apache Is set to 8080

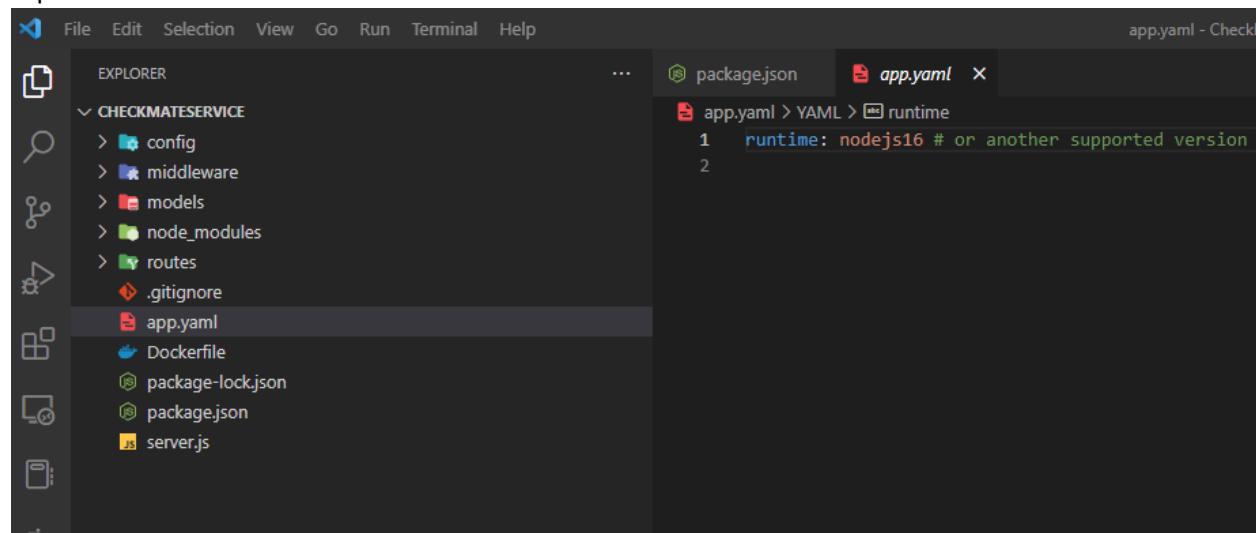
Stop Apache with:

```
brew services stop http
```

Then when you run your application, the default page will come up if cached, if that happens , do a page refresh and it will come up

Tip: Splitting the Service and deploying to GCP

I split out the service to it's own folder



<https://github.com/lionel5116/CheckMateService.git>

git remote add origin https://github.com/lionel5116/CheckMate.git

git add -A

git add .

git commit -m "first commit"

git push origin master

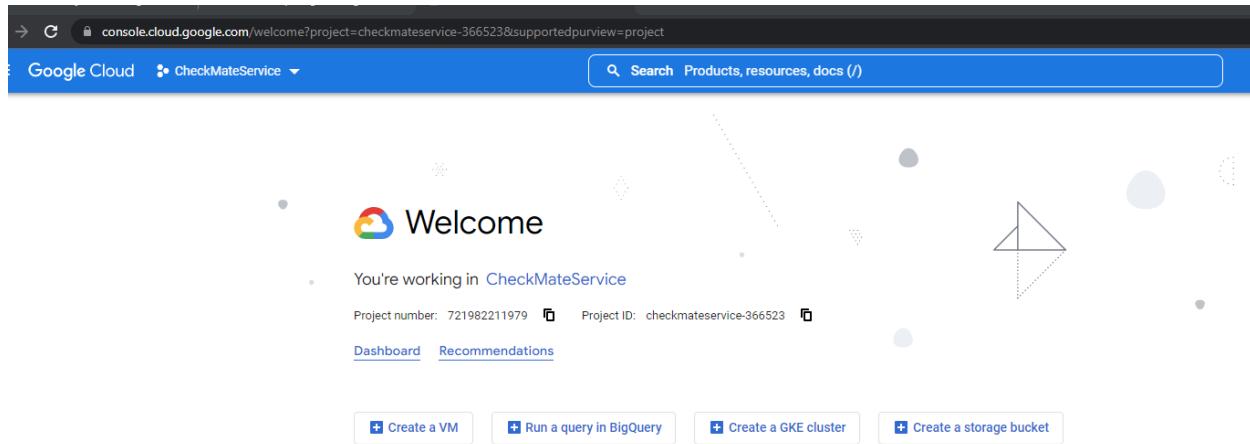
I made sure to include the app.yaml file

I went to Google Cloud and created a new project

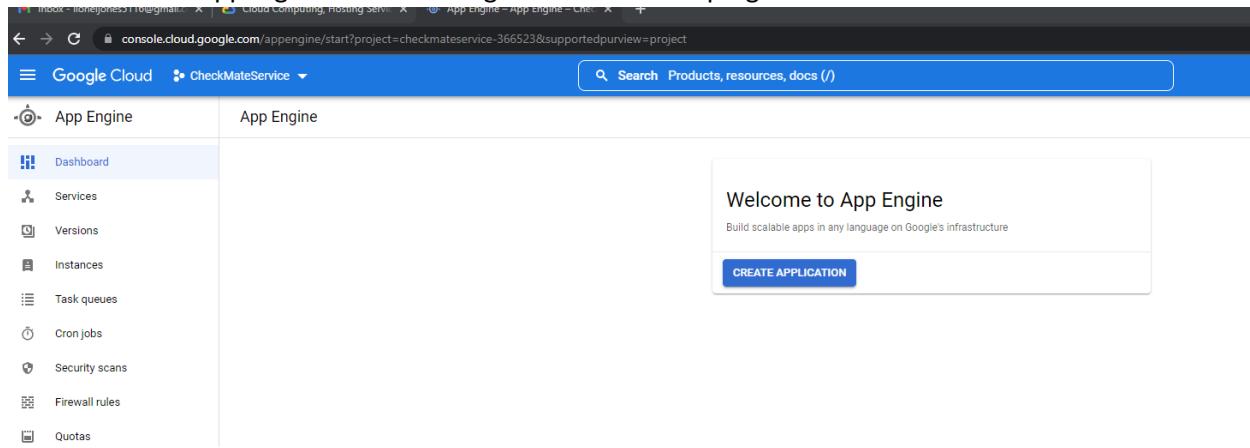
I logged into google cloud and created a new project

The screenshot shows the 'New Project' creation interface. At the top, there's a warning message: '⚠ You have 18 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)' with a 'MANAGE QUOTAS' button. Below that, the 'Project name *' field contains 'CheckMateService'. To the right of the input field is a question mark icon. Underneath, the 'Project ID' is listed as 'checkmateservice-366523. It cannot be changed later.' with an 'EDIT' link. The 'Location *' section shows 'No organization' with a 'BROWSE' button and a 'Parent organization or folder' dropdown. At the bottom are 'CREATE' and 'CANCEL' buttons.

When I finished I was in the new project context of checkmate service



I then selected AppEngine from the burger menu at the top right



When my app was created, I went to the shell

The screenshot shows the Google Cloud Platform App Engine dashboard for the 'CheckMateService' project. The left sidebar contains links for Dashboard, Services, Versions, Instances, Task queues, Cron jobs, Security scans, Firewall rules, Quotas, Release Notes, and Help. The main area has tabs for Resources, Get started, and I'll do this later. Under Resources, it shows Language (Node.js) and Environment (Standard). A section titled 'Deploy with Google Cloud SDK' includes a 'DOWNLOAD THE CLOUD SDK' button and commands for initializing the SDK (\$ gcloud init) and deploying the app (\$ gcloud app deploy). The bottom part shows a Cloud Shell terminal window with the command \$ ls -al, displaying a directory listing for the application files.

```

lionejones5116@cloudshell:~ (checkmateservice-366523)$ ls -al
total 88
drwxr-xr-x 11 lionejones5116 lionejones5116 4096 Oct 25 00:01 .
drwxr-xr-x  4 root      root      4096 Sep  1 16:35 ..
drwxr-xr-x  8 lionejones5116 lionejones5116 4096 Oct 17 19:02 assetmgmtservice
-rw-r--r--  1 lionejones5116 lionejones5116 776 Oct 25 00:01 .bash_history
-rw-r--r--  1 lionejones5116 lionejones5116 220 Jan  1 1970 .bash_logout
-rw-r--r--  1 lionejones5116 lionejones5116 3564 Aug 24 07:14 .bashrc
-rw-r--r--  1 lionejones5116 lionejones5116 21346 Oct 25 00:01 .botc
drwxr-xr-x  4 lionejones5116 lionejones5116 4096 Oct 17 12:16 .cache
drwxr-xr-x  3 lionejones5116 lionejones5116 4096 Sep  1 16:35 cloudshell_open
drwxr-xr-x  4 lionejones5116 lionejones5116 4096 Oct 17 18:14 .config
drwxr-xr-x  2 lionejones5116 lionejones5116 4096 Sep  1 16:35 .docker
drwxr-xr-x  3 lionejones5116 lionejones5116 4096 Oct 17 19:02 .local
drwxr-xr-x  4 lionejones5116 lionejones5116 4096 Oct 17 19:02 .npm
-rw-r--r--  1 lionejones5116 lionejones5116 807 Jan  1 1970 .npmrc
-rw-r--r--  1 lionejones5116 lionejones5116 913 Oct 25 00:00 README-cloudshell.txt
drwxr-xr-x  2 lionejones5116 lionejones5116 4096 Sep  1 16:35 .redhat
drwxr-xr-x  6 lionejones5116 lionejones5116 4096 Sep  1 16:35 .theia

```

And did a gcloud init

This made sure that I was in the correct path for the files that I would be pulling down from github. If you notice with this new application, you do not see any of the files from the assetmgmt2022 service.

That's why you have to
create a new project,
go to App engine,
create a new app
and then re-initialize

Next I am going to clone my repo

git clone <https://github.com/lionej5116/CheckMateService.git>

≡ Google Cloud • CheckMateService ▾

Search Products, resources, doc

App Engine Get started

Dashboard

- Services
- Versions
- Instances
- Task queues
- Cron jobs
- Security scans
- Firewall rules
- Quotas

Release Notes

Resources

Language — Node.js

Environment — Standard

Read App Engine Node.js Standard Environment [Documentation](#).

Visit [Github](#) for Node.js Standard Environment code samples.

I'LL DO THIS LATER

Deploy with

DOWNLOAD TH

Initialize your SDK

\$ gcloud ini

Deploy to App Eng

\$ gcloud app

CLOUD SHELL Terminal (checkmateservice-366523) + ▾

```
-rw-r--r-- 1 lioneljones5116 lioneljones5116 3564 Aug 24 07:14 .bashrc
-rw----- 1 lioneljones5116 lioneljones5116 21346 Oct 25 00:01 .boto
drwxr-xr-x 4 lioneljones5116 lioneljones5116 4096 Oct 17 12:16 .cache
drwxr-xr-x 3 lioneljones5116 lioneljones5116 4096 Sep 1 16:35 cloudshell_open
drwxr-xr-x 4 lioneljones5116 lioneljones5116 4096 Oct 17 18:14 .config
drwxr-xr-x 2 lioneljones5116 lioneljones5116 4096 Sep 1 16:35 .docker
drwxr-xr-x 3 lioneljones5116 lioneljones5116 4096 Oct 17 19:02 .local
drwxr-xr-x 4 lioneljones5116 lioneljones5116 4096 Oct 16 23:26 .npm
-rw-r--r-- 1 lioneljones5116 lioneljones5116 807 Jan 1 1970 .profile
-rw-r--r-- 1 lioneljones5116 lioneljones5116 913 Oct 25 00:00 README-cloudshell.txt
drwxr-xr-x 2 lioneljones5116 lioneljones5116 4096 Sep 1 16:35 .redhat
drwxr-xr-x 6 lioneljones5116 lioneljones5116 4096 Sep 1 16:35 .theia
lioneljones5116@cloudshell:~ (checkmateservice-366523)$ git clone https://github.com/lionel5116/CheckMateService.git
Cloning into 'CheckMateService'...
remote: Enumerating objects: 33, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (19/19), done.
remote: Total 33 (delta 11), reused 33 (delta 11), pack-reused 0
Receiving objects: 100% (33/33), 60.08 KiB | 1.82 MiB/s, done.
Resolving deltas: 100% (11/11), done.
lioneljones5116@cloudshell:~ (checkmateservice-366523)$ []
```

The screenshot shows the Google Cloud Shell interface with the title bar "Google Cloud" and "checkMateService". The main area is a code editor window titled "CLOUD SHELL Editor".

The "EXPLORER" sidebar on the left lists the project structure:

- LIONELJONES5116
- assetmgmtservice
- CheckMateService (selected)
- config
- middleware
- models
- routes
 - app.yaml
 - Dockerfile
 - package-lock.json
 - package.json
 - server.js
- cloudshell_open
- README-cloudshell.txt

The "PROBLEMS" tab in the bottom right shows the following file listing:

```
drwxr-xr-x 4 root      root          4096 Sep  1 16:35 ..
-rw----- 1 lioneljones5116 lioneljones5116 758 Oct 17 19:02 .bash_history
-rw-r--r-- 1 lioneljones5116 lioneljones5116 220 Jan  1 1970 .bash_logout
-rw-r--r-- 1 lioneljones5116 lioneljones5116 3564 Aug 24 07:14 .bashrc
drwxr-xr-x 4 lioneljones5116 lioneljones5116 4096 Oct 17 12:16 .cache
drwxr-xr-x 4 lioneljones5116 lioneljones5116 4096 Oct 17 18:14 .config
drwxr-xr-x 2 lioneljones5116 lioneljones5116 4096 Sep  1 16:35 .docker
drwxr-xr-x 3 lioneljones5116 lioneljones5116 4096 Oct 17 19:02 .local
drwxr-xr-x 4 lioneljones5116 lioneljones5116 4096 Oct 16 23:26 .npm
-rw-r--r-- 1 lioneljones5116 lioneljones5116 807 Jan  1 1970 .profile
drwxr-xr-x 2 lioneljones5116 lioneljones5116 4096 Sep  1 16:35 .redhat
drwxr-xr-x 6 lioneljones5116 lioneljones5116 4096 Sep  1 16:35 .theia
-rw-r--r-- 1 lioneljones5116 lioneljones5116 913 Oct 24 23:42 README-cloudshell.txt
drwxr-xr-x 8 lioneljones5116 lioneljones5116 4096 Oct 17 19:02 assetmgmtservice
drwxr-xr-x 3 lioneljones5116 lioneljones5116 4096 Sep  1 16:35 cloudshell_open
lioneljones5116@cloudshell:~ (checkmateService)$
```

You will see all of your projects (just like VSCode) – This is VSCode, but you are in the directory of your new service

cd into your checkmateservice folder

The terminal window shows the directory structure of the `CheckMateService` project. It includes subfolders for `config`, `middleware`, `models`, and `routes`, along with files like `app.yaml`, `Dockerfile`, `package-lock.json`, `package.json`, and `server.js`. The current working directory is `~/CheckMateService (checkmateservice)`. A command `ls -al` is run, displaying a detailed list of files and their permissions, sizes, and modification dates.

```
lioneljones5116@cloudshell:~/CheckMateService (checkmateservice)$ ls -al
total 208
drwxr-xr-x  7 lioneljones5116 lioneljones5116  4096 Oct 25 00:05 .
drwxr-xr-x 12 lioneljones5116 lioneljones5116  4096 Oct 25 00:05 ..
drwxr-xr-x  8 lioneljones5116 lioneljones5116  4096 Oct 25 00:05 .git
-rw-r--r--  1 lioneljones5116 lioneljones5116    50 Oct 25 00:05 .gitignore
-rw-r--r--  1 lioneljones5116 lioneljones5116   630 Oct 25 00:05 Dockerfile
-rw-r--r--  1 lioneljones5116 lioneljones5116   49 Oct 25 00:05 app.yaml
drwxr-xr-x  2 lioneljones5116 lioneljones5116  4096 Oct 25 00:05 config
drwxr-xr-x  2 lioneljones5116 lioneljones5116  4096 Oct 25 00:05 middleware
drwxr-xr-x  2 lioneljones5116 lioneljones5116  4096 Oct 25 00:05 models
-rw-r--r--  1 lioneljones5116 lioneljones5116 163308 Oct 25 00:05 package-lock.json
-rw-r--r--  1 lioneljones5116 lioneljones5116   611 Oct 25 00:05 package.json
drwxr-xr-x  3 lioneljones5116 lioneljones5116  4096 Oct 25 00:05 routes
-rw-r--r--  1 lioneljones5116 lioneljones5116  1829 Oct 25 00:05 server.js
lioneljones5116@cloudshell:~/CheckMateService (checkmateservice)$
```

do an npm install to install your modules

Then do a npm start

The terminal window shows the output of the `npm install` command. It lists several warnings about deprecated packages: `har-validator@5.1.5`, `querystring@0.2.0`, `uuid@3.4.0`, and `request@2.88.2`. After the install, it shows that 235 packages were added and audited in 11 seconds. It also lists 22 packages looking for funding. Finally, the `npm start` command is run, which starts the application and connects to MongoDB.

```
lioneljones5116@cloudshell:~/CheckMateService (checkmateservice)$ npm install
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated querystring@0.2.0: The querystring API is considered Legacy. new code should
npm WARN deprecated deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use M
s://v8.dev/blog/math-random for details.
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/r

added 235 packages, and audited 236 packages in 11s

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

found 0 vulnerabilities
lioneljones5116@cloudshell:~/CheckMateService (checkmateservice)$ npm start

> checkmateservice@1.0.0 start
> node server

Server started on port 5500
MongoDB Connected...
```

Now deploy

The screenshot shows a terminal window with a file tree on the left and a command line on the right. The file tree includes directories like assetmgmtservice, CheckMateService, config, middleware, models, node_modules, routes, and files like app.yaml, Dockerfile, package-lock.json, package.json, and server.js. The command line shows the user 'lioneljones5116' at a 'cloudshell' prompt, running the command 'gcloud app deploy'. The terminal has tabs for 'Problems' and 'checkmateservice'.

I got a permission denied error at first, because it was referencing a project with the wrong project ID. I just went back to the project selector, made sure I was in right context and directory at the shell.

Then I typed this

The screenshot shows a terminal window with a command line. The user 'lioneljones5116' runs 'gcloud config set project checkmateservice-366523'. The terminal also displays instructions to view the application in a web browser using 'gcloud app browse'.

This made sure I was in the correct context

Then I did a
gcloud app deploy

I got another error:

The screenshot shows a terminal window with a command line. The user 'lioneljones5116' runs 'gcloud app deploy'. The output shows an error message: 'ERROR: (gcloud.app.deploy) Error Response: [9] Cloud build 6c07042d-1129-4636-a362-d357c0314706 status: FAILURE'. It indicates an unexpected error occurred and provides a link to the build logs: 'https://console.cloud.google.com/cloud-build/builds;region=us-central1/6c07042d-1129-4636-a362-d357c0314706?project=721982211979'. The terminal also shows the descriptor path: '/home/lioneljones5116/CheckMateService/app.yaml'.

But this was standard error that I always get on first build

After typing deploy again, the build went through

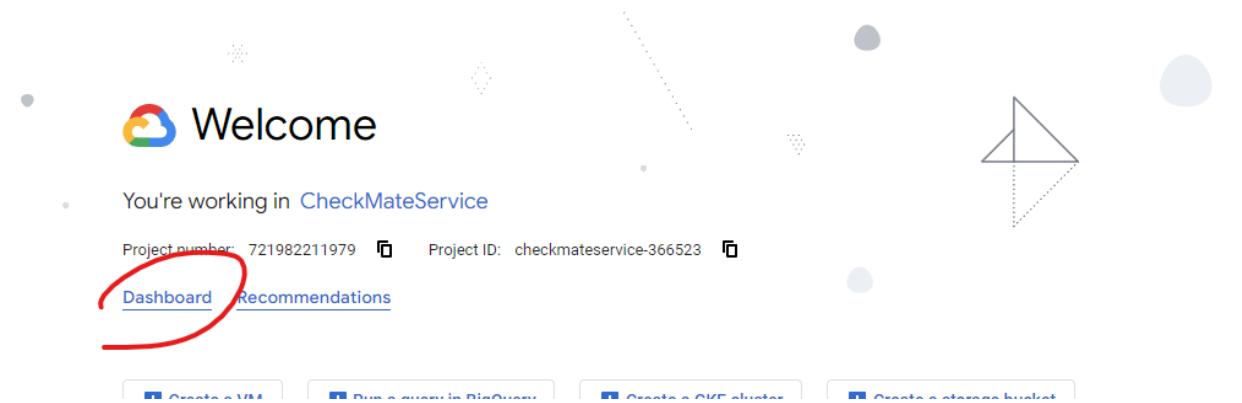
```
Do you want to continue (Y/n)? y

Beginning deployment of service [default]...
Uploading 0 files to Google Cloud Storage
100%
File upload done.
Updating service [default]...done.
Setting traffic split for service [default]...done.
Deployed service [default] to [https://checkmateservice-366523.uc.r.appspot.com]

You can stream logs from the command line by running:
$ gcloud app logs tail -s default

To view your application in the web browser run:
$ gcloud app browse
lioneljones5116@cloudshell:~/CheckMateService (checkmateservice-366523)$ gcloud app deploy^C
lioneljones5116@cloudshell:~/CheckMateService (checkmateservice-366523)$ 
```

I went to the dashboard



Clicked on App Engine

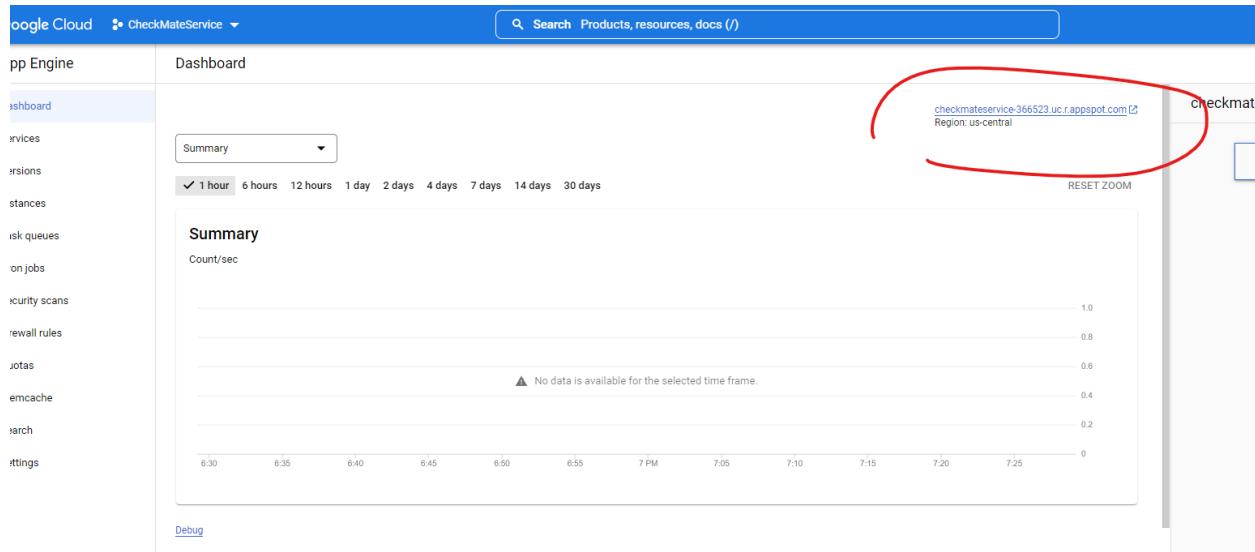
The screenshot shows the Google Cloud Platform dashboard. On the left, there's a sidebar with three dots. The main area has two main sections:

- App Engine**: This section includes a chart titled "Summary (count/sec)" with a Y-axis from 0 to 1.0 and an X-axis from 6 PM to 6:45. Below the chart is a link: "→ Go to the App Engine dashboard". This link is circled in red.
- API APIs**: This section includes a chart titled "Requests (requests/sec)" with a Y-axis from 0.8 to 1.0 and an X-axis from 6 PM to 6:45.

On the right side, there are several cards:

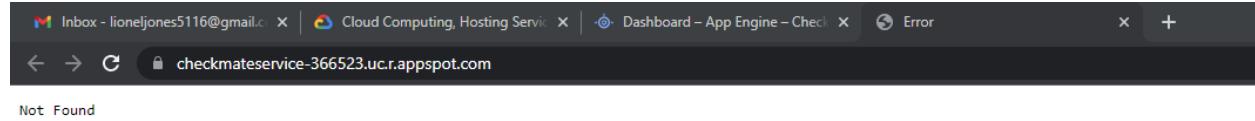
- Google Clicks**: Shows "All services no errors".
- Billing**: Shows "Estimated charges \$0.00".
- Monitoring**: Shows links to "Create my dashboard", "Set up alerting rules", and "Create uptime checks".

There is your link

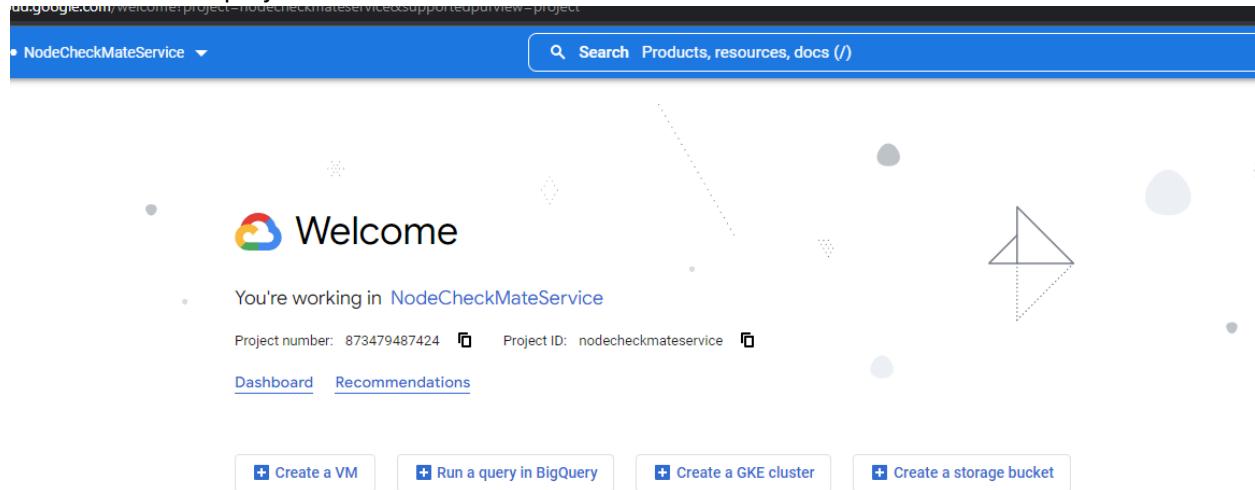


<https://checkmatestervice-366523.uc.r.appspot.com/>

When I browse, it says not found



I created another project



Quick access [?](#)

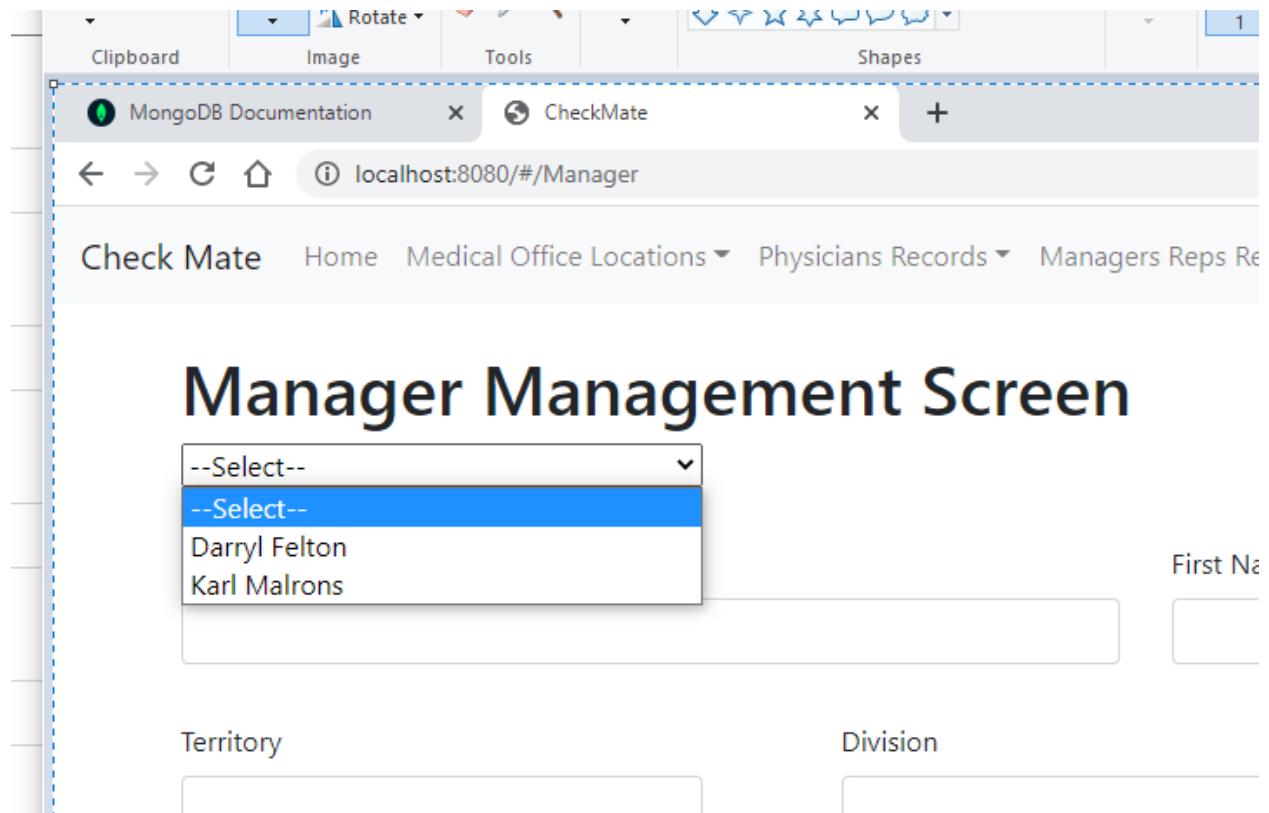
I removed the old folder

rm -rf CheckMateService/

Same process, still 404

Tip: How to grab fields from MongoDB and dynamically populate a select

In this scenario, I had to find a way to populate a select dynamically from a record where the values were in two columns (First Name and Last Name). The end result:



The solution

I created a custom route:

```
routes > api > manager.js > ⚡ router.get('/fetchManagerFirstAndLastName') callback
1  const express = require('express');
2  const router = express.Router();
3  const Manager = require('../models/Manager');
4  const config = require('config')
5  const { check, validationResult } = require('express-validator')
6
7  router.get('/fetchManagerFirstAndLastName', async(req,res) => {
8    try {
9
10      const manager = await Manager.find({}, {FirstName: 1, LastName: 1,_id: 0});
11      if(!manager) {
12        return res.status(404).json({msg:'Record not found'});
13      }
14      res.json(manager);
15
16    } catch (err) {
17      //console.log(err.message);
18
19      if(err.kind =='ObjectId') {
20        return res.status(404).json({msg:'Record not found'});
21      }
22      res.status(500).send('Server error')
23
24    }
25  })
26}
27
28 > router.post('/createManagerRecord', ...
29   });
30
31 > router.post('/searchManagerRecord', ...
```

The mongodb syntax returns the columns I needed (FirstName and Last Name)
Then I created an action

```
3 | export const fetchManagerFirstAndLastName =() =>
4 |   async (dispatch) => {
5 |
6 |     let serviceUrl = "";
7 |     serviceUrl = process.env.REACT_APP_SERVICE_URL + '/manager/fetchManagerFirstAndLastName'
8 |
9 |     try {
10 |       const config = {
11 |         headers: {
12 |           "Content-Type": "application/json",
13 |         },
14 |       };
15 |
16 |       const res = await axios.get(serviceUrl,config);
17 |       if (res.status === 200) {
18 |
19 |         return res.data
20 |       }
21 |       else {
22 |         dispatch(
23 |           setAlert(
24 |             "Could not find record",
25 |             "danger"
26 |           )
27 |         );
28 |         return []
29 |       }
30 |
31 |     } catch (err) {
32 |       console.log(err);
33 |       dispatch(
34 |         setAlert(
35 |           "Could not find record",
36 |             "danger"
37 |           )
38 |         );
39 |     }
40 |   };
41 | }
```

Then I made a call to grab the items and dynamically populate a select control

```
async function getManagerFirstAndLastName(e) {
  e.preventDefault();
  let _SEARCH_DATA = [];
  _SEARCH_DATA = await fetchManagerFirstAndLastName();
  console.log(_SEARCH_DATA)
  //ddMangers
  var _ddMangers = document.getElementById('ddMangers');

  _ddMangers.options[_ddMangers.options.length] = new Option('--Select--');
  for(const key in _SEARCH_DATA) {
    _ddMangers.options[_ddMangers.options.length] = new Option(_SEARCH_DATA[key].FirstName + ' ' + _SEARCH_DATA[key].LastName);
  }
}
```

```
</h1>
<Form onSubmit={e => onSubmit(e)}>

  <Row className="mb-3">
    <Form.Group as={Col}>
      <select class="form-select form-select-sm"
        aria-label=".form-select-sm example"
        style={{ width: 300 }}
        name='ddMangers'
        id='ddMangers'
      >
        </select>
    </Form.Group>
  </Row>
```

Tip: Creating custom routes with express js

Any custom route you create is done with
router.get(/<the name of your route..>

So for example:

```
routes > api > manager.js > router.get('/fetchManagerFirstAndLastName') callback
1 const express = require('express');
2 const router = express.Router();
3 const Manager = require('../models/Manager');
4 const config = require('config')
5 const { check, validationResult } = require('express-validator')
6
7 router.get('/fetchManagerFirstAndLastName', async(req,res) => {
8     try {
9
10
11         const manager = await Manager.find({}, {FirstName: 1, LastName: 1,_id: 0});
12         //const manager = await Manager.find();
13         if(!manager) {
14             return res.status(404).json({msg:'Record not found'});
15         }
16         res.json(manager);
17
18     } catch (err) {
19         //console.log(err.message);
20
21         if(err.kind =='ObjectId') {
22             return res.status(404).json({msg:'Record not found'});
23         }
24         res.status(500).send('Server error')
25
26     }
27 }
28 }
```

<http://localhost:5500/api/manager/fetchManagerFirstAndLastName>

The screenshot shows the Postman interface with a GET request to `http://localhost:5500/api/manager/fetchManagerFirstAndLastName`. The response body is a JSON array containing two documents:

```

1  [
2    {
3      "_id": "6355d6e81252bbcc02c342a6",
4      "FirstName": "Darryl",
5      "LastName": "Felton",
6      "Territory": "Houston",
7      "Division": "Galleria",
8      "District": "FT Bend",
9      "Phone": "713-876-0987",
10     "Mobile": "(832) 819-3682",
11     "Email": "dfetlition@yahoo.com",
12     "Notes": "Good Guy",
13     "__v": 0
14   },
15   {
16     "_id": "6355d7241252bbcc02c342aa",
17     "FirstName": "Karl",
18     "LastName": "Malrons",
19     "Territory": "Dallas",
20     "Division": "New Barrington",
21     "District": "Harris",
22     "Phone": "713-876-0987",
23     "Mobile": "832-987-2134",
24     "Email": "emarlonsman@gmail.com",
25     "Notes": "Always a good choice",
26     "__v": 0
27   }
28 ]

```

But one important note:

YOU HAVE TO HAVE ALL OF YOUR CUSTOM ROUTES BEFORE YOUR OTHER REGULAR GET,PUT,DELETE ETC.. ROUTES ARE YOU WILL GET AN ERROR:

```

Server started on port 5500
MongoDB Connected...
Cast to ObjectId failed for value "etchManagerFirstAndLastName" (type string) at path "_id" for model "Manager"
Tried casting "etchManagerFirstAndLastName" to type "ObjectID".

```

So your routes have to be in this order

```
server.js          manager.js M X

routes > api > manager.js > router.get('/fetchManagerFirstAndLastName') callback
1  const express = require('express');
2  const router = express.Router();
3  const Manager = require('../models/Manager');
4  const config = require('config')
5  const { check, validationResult } = require('express-validator')
6
7  > router.get('/fetchManagerFirstAndLastName', async(req,res) => {[...]
8  })
9
10 > router.post('/createManagerRecord', ...
11   );
12
13 > router.post('/searchManagerRecord', ...
14   );
15
16 > router.get('/:id', async(req,res) => { ...
17   })
18
19 > router.delete('/:id', async(req,res) => { ...
20   })
21
22 > router.put('/:id', async(req,res) => { ...
23   })
24
25
26 > module.exports = router;
```

Tip: MongoDB Commands – Visual Studio Code

```

File Edit Selection View Go Run Terminal Help
CONNECTIONS
  ✓ clustered@jwcnt.mongodb.net connected
    > AssetManagement2022
    > Checkmate
    > HuckeVUEProject
    > NutritionServices
    > TripTracker
    > _realm_sync
    > huckle
    > jwtextexample
    > mem
    > places
    > test
    > todo
    > admin
    > local
PLAYGROUNDS
No 'mongodt' playground files found in the workspace.
Create New Playground
Get Started PlaygroundMongoDB.mongodt
C:\> DevProjects\HUED\Docs\PlaygroundMongoDB.mongodt
Connected to cluster clustered@jwcnt.mongodb.net. Click here to change connection.
1 // To disable this template go to Settings | Mongodt | Use Default Template For Playground.
2 // Make sure you are connected to enable completions and to be able to run a playground.
3 // Use Ctrl+Space inside a snippet or a string literal to trigger completions.
4 // Select the database to use.
5 //use("mongodt");
6 //use("mongodtCodePlaygroundDB");
7
8 // The drop() command destroys all data from a collection.
9 // Make sure you run it against the correct database and collection.
10 //db.sales.drop();
11
12 // Insert a few documents into the sales collection.
13 /*
14 db.sales.insertMany([
15   { '_id': 1, 'item': 'abc', 'price': 10, 'quantity': 2, 'date': new Date('2014-03-01T08:00:00Z')
16   { '_id': 2, 'item': 'jkl', 'price': 20, 'quantity': 1, 'date': new Date('2014-03-01T09:00:00Z')
17   { '_id': 3, 'item': 'xyz', 'price': 10, 'quantity': 10, 'date': new Date('2014-03-01T10:00:00Z')
18   { '_id': 4, 'item': 'abc', 'price': 5, 'quantity': 20, 'date': new Date('2014-04-04T09:21:39.739Z')
19   { '_id': 5, 'item': 'abc', 'price': 10, 'quantity': 10, 'date': new Date('2014-04-04T21:23:13.33Z')
20   { '_id': 6, 'item': 'def', 'price': 7.5, 'quantity': 5, 'date': new Date('2015-06-04T05:00:13Z')
21   { '_id': 7, 'item': 'def', 'price': 7.5, 'quantity': 10, 'date': new Date('2015-09-10T08:43:00Z')
22   { '_id': 8, 'item': 'abc', 'price': 10, 'date': new Date('2016-02-06T20:20:13Z')
23 ];
24 }
25
26 // Run a find command to view items sold on April 4th, 2014.
27 db.sales.find({ date: { $gte: new Date('2014-04-04'), $lt: new Date('2014-04-05') } });
28
29 // Build an aggregation to view total sales for each product in 2014.
30 const aggregation = [
31   { $match: { date: { $gte: new Date('2014-01-01'), $lt: new Date('2015-01-01') } } },
32   { $group: { _id: '$item', totalSaleAmount: { $sum: { $multiply: [ '$price', '$quantity' ] } } } }
33 ];
34
35 // Run the aggregation and open a cursor to the results.
36 // Use toIterate() to exhaust the cursor to return the whole result set.
37 // You can use hasNext() / next() to iterate through the cursor page by page.
38 db.sales.aggregate(aggregation);
39
40
41
42 //https://www.mongodb.com/community/forums/t/mongodb-for-vs-code-wont-change-db/9278/2
43 //https://www.mongodb.com/docs/manual/reference/operator/query/
44 //https://stackoverflow.com/questions/25589113/how-to-select-a-single-field-for-all-documents-in-a-
45 use('Checkmate');
46 db.managers.find({ FirstName: 1, LastName: 1, _id: 0 });
47
48 db.managers.find({ LastName: {$eq: 'Marlons'} }).pretty()
49 db.managers.find({}).pretty()

```

The way playgrounds work is that you have to use the syntax as shown below. (Each run you do you have to specify what database you want to use at each run. The last command always is shown. It does show all lines, just the results from the last line

<https://www.mongodb.com/community/forums/t/mongodb-for-vs-code-wont-change-db/9278/2>

<https://www.mongodb.com/docs/manual/reference/operator/query/>

<https://stackoverflow.com/questions/25589113/how-to-select-a-single-field-for-all-documents-in-a-mongodb-collection>

```

use ('Checkmate');
//return specific columns (the _id:0 means do not return the id column)
db.managers.find({}, {FirstName: 1, LastName: 1, _id: 0});

```

```

//filter the result from managers where the last name = 'Marlons'
db.managers.find({LastName:{$eq:'Marlons'}}).pretty()

```

```

//show all records in the managers collection
db.managers.find({}).pretty()

```

So to run a command, highlight the syntax and hit the run button



```
use ('Checkmate');
db.managers.find({}, {FirstName: 1, LastName: 1,_id: 0});
```

Results:



```
{}
Playground Result ×
```

```
1 [           ← Clicked here
2 {
3   "FirstName": "Darryl",
4   "LastName": "Felton"
5 },
6 {
7   "FirstName": "Karl",
8   "LastName": "Malrons"
9 }
10 ]
```

This is a custom route that returns first and last name from the managers model

```
server.js          manager.js M X
routes > api > manager.js > router.get('/fetchManagerFirstAndLastName') callback
  1  const express = require('express');
  2  const router = express.Router();
  3  const Manager = require('../models/Manager');
  4  const config = require('config')
  5  const { check, validationResult } = require('express-validator')
  6
  7  router.get('/fetchManagerFirstAndLastName', async(req,res) => {
  8    try {
  9
 10
 11      const manager = await Manager.find({}, {FirstName: 1, LastName: 1,_id: 0});
 12      if(!manager) {
 13        return res.status(404).json({msg:'Record not found'});
 14      }
 15      res.json(manager);
 16
 17    } catch (err) {
 18      //console.log(err.message);
 19
 20      if(err.kind == 'ObjectId') {
 21        return res.status(404).json({msg:'Record not found'});
 22      }
 23      res.status(500).send('Server error')
 24
 25    }
 26  })
 27
```

http://localhost:5500/api/manager/fetchManagerFirstAndLastName

GET http://localhost:5500/api/manager/fetchManagerFirstAndLastName

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON ↻

```
1 [
2   {
3     "FirstName": "Darryl",
4     "LastName": "Felton"
5   },
6   {
7     "FirstName": "Karl",
8     "LastName": "Malrons"
9   }
10 ]
```

It's the same syntax as what I used in playgrounds

```
42 //https://www.mongodb.com/community/forums/t/mongodb-for-vs-code-wont-
43 //https://www.mongodb.com/docs/manual/reference/operator/query/
44 //https://stackoverflow.com/questions/25589113/how-to-select-a-single-
45 use ('Checkmate');
46 db.managers.find({}, {FirstName: 1, LastName: 1, _id: 0});
47
```

So the **MongoDB Syntax works !!!**, We are just using Mongoose objects, you just replace the db.managers. with Manager.

Tip: Check for open port

netstat -aon

Active Connections				
Proto	Local Address	Foreign Address	State	PID
TCP	0.0.0.0:80	0.0.0.0:0	LISTENING	4
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING	1304
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING	4
TCP	0.0.0.0:5040	0.0.0.0:0	LISTENING	8344
TCP	0.0.0.0:5700	0.0.0.0:0	LISTENING	4
TCP	0.0.0.0:7680	0.0.0.0:0	LISTENING	9260
TCP	0.0.0.0:9012	0.0.0.0:0	LISTENING	16496
TCP	0.0.0.0:27036	0.0.0.0:0	LISTENING	12024
TCP	0.0.0.0:32400	0.0.0.0:0	LISTENING	6712
TCP	0.0.0.0:49664	0.0.0.0:0	LISTENING	1092
TCP	0.0.0.0:49665	0.0.0.0:0	LISTENING	828
TCP	0.0.0.0:49666	0.0.0.0:0	LISTENING	2516
TCP	0.0.0.0:49667	0.0.0.0:0	LISTENING	3000
TCP	0.0.0.0:49668	0.0.0.0:0	LISTENING	4104
TCP	0.0.0.0:49679	0.0.0.0:0	LISTENING	1060
TCP	0.0.0.0:49682	0.0.0.0:0	LISTENING	4916
TCP	0.0.0.0:55645	0.0.0.0:0	LISTENING	39732
TCP	0.0.0.0:57621	0.0.0.0:0	LISTENING	39732
TCP	127.0.0.1:1434	0.0.0.0:0	LISTENING	6168
TCP	127.0.0.1:2015	0.0.0.0:0	LISTENING	5460
TCP	127.0.0.1:5939	0.0.0.0:0	LISTENING	4664
TCP	127.0.0.1:8080	0.0.0.0:0	LISTENING	23172
TCP	127.0.0.1:8884	0.0.0.0:0	LISTENING	4
TCP	127.0.0.1:27015	0.0.0.0:0	LISTENING	16688
TCP	127.0.0.1:27060	0.0.0.0:0	LISTENING	12024
TCP	127.0.0.1:32401	0.0.0.0:0	LISTENING	6712

For individual Port

netstat -ano | findstr :8080

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.22621.674]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>netstat -ano | findstr :8080
  TCP    127.0.0.1:8080        0.0.0.0:0          LISTENING      23172

C:\Windows\System32>
```

to kill

taskkill /PID 23172 /F

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.22621.674]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>netstat -ano | findstr :8080
  TCP    127.0.0.1:8080        0.0.0.0:0          LISTENING      23172

C:\Windows\System32>taskkill /PID 23172 /F
SUCCESS: The process with PID 23172 has been terminated.

C:\Windows\System32>
```

Tip: Fixing the issue when you fetch for record to edit and then type in a field and all of your data disappears from other fields

When you bring back a record, you have to set the field values like this

```
async function fetchSingleRecordByRecordID(id) {
  let _SEARCH_DATA = [];

  _SEARCH_DATA = await getPharmaRecord(id);

  setID(id)

  var _Name = document.getElementById('Name')
  _Name.value = _SEARCH_DATA.Name;
  formData.Name = _Name.value;

  var _Phone = document.getElementById('Phone')
  _Phone.value = _SEARCH_DATA.Phone;
  formData.Phone= _Phone.value;

  var _Email = document.getElementById('Email')
  _Email.value = _SEARCH_DATA.Email;
  formData.Email= _Email.value;

  var _password = document.getElementById('Password')
  _password.value = _SEARCH_DATA.Password;
  formData.Password= _password.value;

  var _Notes = document.getElementById('Notes')
  _Notes.value = _SEARCH_DATA.Notes;
  formData.Notes= _Notes.value;
}

}
```

If not, when you type in one field, it will wipe out your model
I don't need the code below, I am just keeping it for reference

```
/*
function onChange (e){
  e.preventDefault();
  const { name, value } = e.target;

  switch (name) {
    case 'Name':
      setFormData({ ...formData, Name: value });
    }
  }
}
```

```
        break;
    case 'Phone':
        setFormData({ ...formData, Phone: value });
        break;
    case 'Email':
        setFormData({ ...formData, Email: value });
        break;
    case 'Password':
        setFormData({ ...formData, Password: value });
        break;
    case 'Notes':
        setFormData({ ...formData, Notes: value });
        break;
    default:
        break;
    }

    // console.log(user)
}
*/

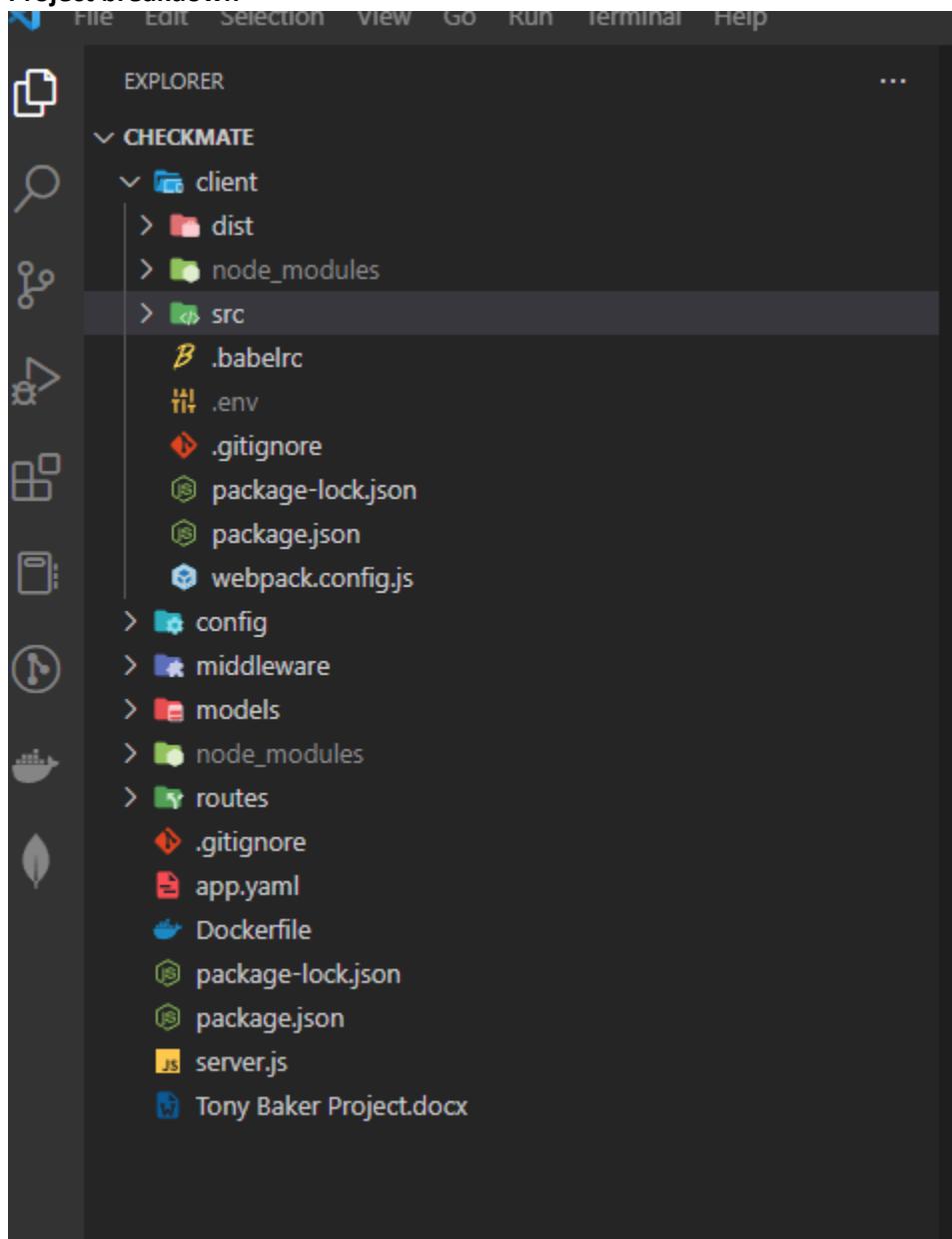
```

Tip: Starting development

Git repo

<https://github.com/lionel5116/CheckMate.git>

Project breakdown



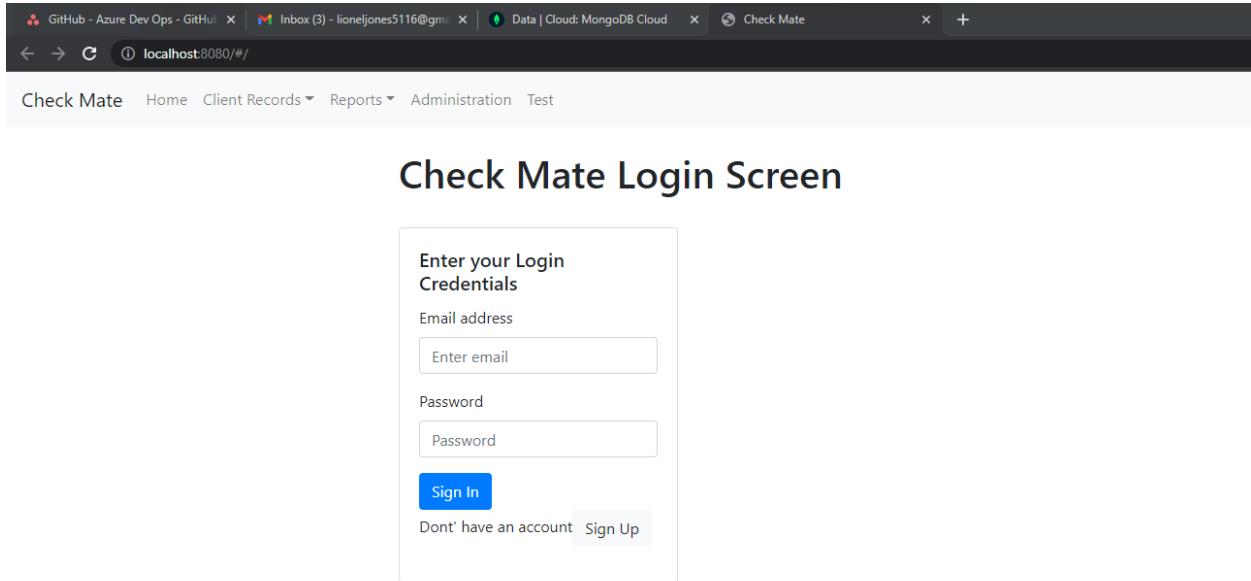
Combined directory with

React Client App

NodeJs Service

I have authentication working

SCREENS



Check Mate Registration

User Name
lioneljones5116@gmail.com

Email address
Enter email

Password

Sign Up

Already have an account ?
Sign In

Search Pharmaceutical Companies

Add Pharmaceutical Records

Search Pharmaceutical Records

Add Medical Group Records

Search Medical Group Records

Search Criteria

All

Email
Enter Email

Name
Name

Search Records

_id	Name	Phone	Email	Edit	Delete
63549190fe74bf00a27dcfa5	Memorial Health	713-654-0987	mmhealth@gmail.com		
6354a04dd5e97ead84326278	Herman Hospital System	713-876-0987	corsonmemorial@gmail.com		

10 ▾ Showing rows 1 to 2 of 2

1

Physicians Management Screen

-select- ▾

First Name

MI

Last Name

NPI

License ID

Email

Phone

Password

Confirm Password

Notes

Submit

Update Record

The Rep Screen

X CheckMate X +

t8080/#/Rep?id=63569b7d0fc9330d56e8c006

Medical Office Locations ▾ Physicians Records ▾ Managers Reps Receipts ▾ Reports ▾ Administration REDUX AUTH TEST

Rep Management Screen

Karl Malrons

Last Name	First Name
David	Michaels
Territory	Phone
Washington Avenue	987-098-4653
Email	Annual Budget
dmicals@yahoo.com	75000
Notes	
Hmm	

Submit **Update Record**

The drop-down is from the Managers table – **Every Rep is associated with a Manager**

On the Search Rep Screen, you can find all reps that are associated with a Manager

Search Reps

Enter Search Criteria

ManagerName

Manager Name

Darryl Felton

Email

Enter Email

Territory

Territory

Search Records

Search Results

_id	FirstName	LastName	Territory	ManagerName	Email	Edit	Delete
6356962a2a2059745df153fa	Jones	Lionel	Beltway 8	Darryl Felton	ljones@yahoo.com		
6356dfc7661a7f5b717874d9	Goode	Karle	Deason Area	Darryl Felton	kgoode@yahoo.com		

Showing rows 1 to 2 of 2

1

Document Start Date:

October 20, 2022

Purpose of the application:

This application is responsible for tracking and compliance with Pharmaceutical expenses that reps incur while negotiating a drug with doctors

Use Case(s)- Needs

The application needs to a SAS (Software as a Service Application)

In needs to be a subscription based model application.

User Count(s)

The future number of users can exceed 10,000+ (once the business get's going)

Scalability

The application has to be designed to scale

Future Enhancement (Requests)

Client wants the ability for the application to integrate with third party products (i:e SAP)

Software Architecture

The application needs to be a multi-tier application

Front End (Either React or VueJS)

Middle Tier: NodeJS or .Net

Database: SQL Server or MongoDB

Payment Processing – Stripe

localhost:8080/#/CheckOutForm

?022 Assets ▾ Travels ▾ Search ▾ Reports ▾

Subscribe Logout

Payment/Subscription Details

Name Email

Address

City State Zip

Credit Card

Pay Now

Subscribe \$100.00

checkout.stripe.com/c/pay/ppage_1LvNB0KWeLMKqrB6FwZ5cTx1#fidkdWxOYHwnPyd1b|pxYHZxWjA0SXN|PUxOUmBJSE50d0czQ1d8MUJWSkZWV3Z2amRVVzxbm2dNd0dsZ3Nwa1dSSFFUvF%2FXVZ3U2FGY1dKdExKd0YyQGK:

Lionel Jones Software **TEST MODE**

Annual Developer Subscription
\$100.00

This is a test subscription model for development purposes



G Pay

Or pay with card

Email

Card information

1234 1234 1234 1234	
MM / YY	CVC

Name on card

Country or region

United States

ZIP

Save my info for secure 1-click checkout

Enter your phone number to pay faster on Lionel Jones Software and thousands of sites.

(201) 555-0123

[link](#) • [More info](#)

Pay

Key Features:

It already has JWT authentication wired up

It has NodeJS Service that fully configured

VueJS is easy to code in

The project is set up to split into multiple parts (you split the client project from the nodejs api) and deploy them separately or together.

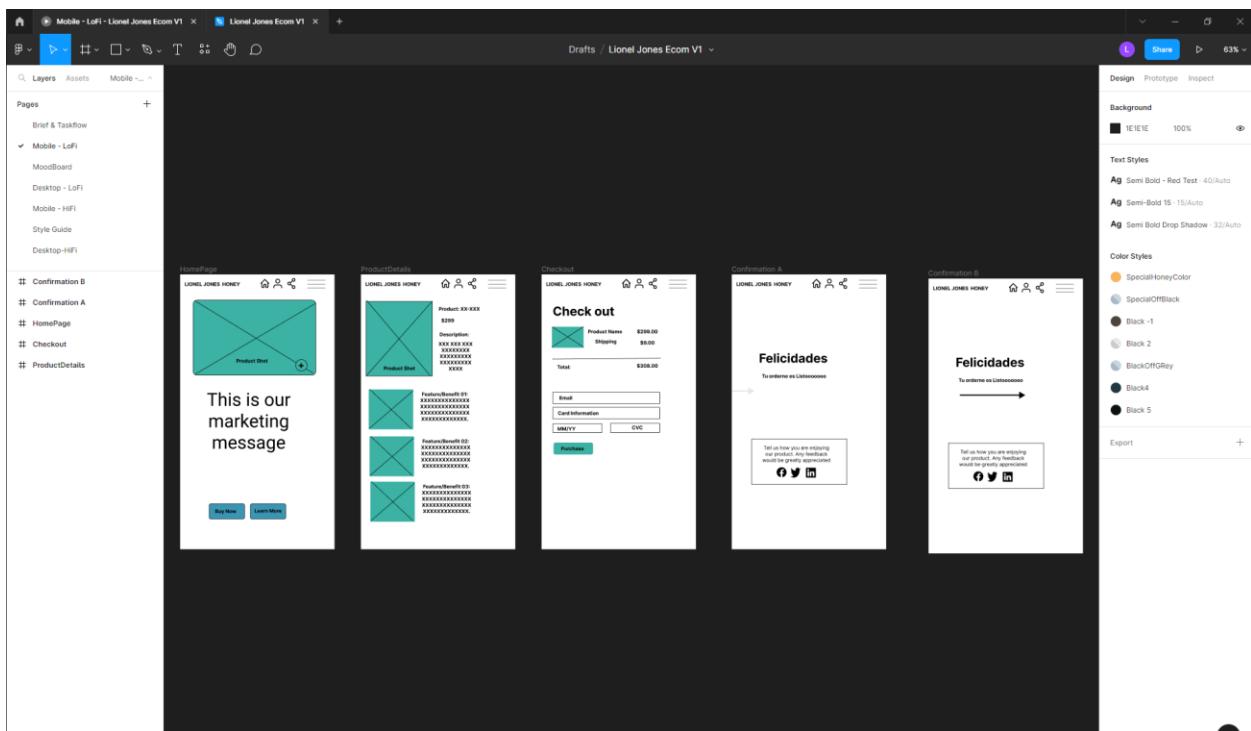
It's a good stack (MEVN)

Rapid development time

Suggestions for UI/UX

For the UX design, I would suggest getting a good UX design person

I would have them use FIGMA



Then turn over the design to the developers for cleanup toward the middle->end of development after the functionality – MVP (Minimal Viable Product) has been developed.

Hosting

Heroku

For low cost hosting (Say for MEVN) stack

I would go with Heroku

The screenshot shows the Heroku dashboard for the app 'vast-scrubland-03341'. At the top, there's a navigation bar with links for Overview, Resources, Deploy, Metrics, Activity, Access, and Settings. Below this is a banner with a link to 'Heroku Pipelines' and a 'Create a Heroku Pipeline' button. The main area is divided into several sections: 'Metrics (last 24hrs)' showing response time, throughput, and memory usage; 'Installed add-ons' showing \$0.00/month; 'Dyno formation' showing \$7.00/month; and 'Collaborator activity' showing 10 deployments. The 'Latest activity' section is filled with deployment logs from the user 'lionejones5116@gmail.com'.

It is an easy process in terms of deployment, cost etc. (This has been proven).

POC – MVP

If you wanted to put together a quick MVP for proof of concept, a good VUE = MEVN stack project would be good.

Azure

Azure is a good choice. Especially if we go with the React/SQL Based project. It's Microsoft so porting the Database backend would be seamless. – **Azure is a bit more costly**, but since the application is expected to scale rapidly, this would be a good choice.

The screenshot shows the Azure free account landing page. At the top, there's a navigation bar with various tabs. Below it, a large banner features the text "Build in the cloud with an Azure free account" and "Create, deploy, and manage applications across multiple clouds, on-premises, and at the edge". It includes a "Start free" button and a "Pay as you go >" link. To the right is a graphic of a laptop displaying the Azure portal interface with several service icons. Below the banner are three callout boxes: "Popular services free for 12 months" (View all services), "40+ other services free always" (View all services), and "Start with \$200 Azure credit" (30 days to use in addition to free services).

Here's just some of what you can do with Azure

Final thoughts/Suggestions

Since I can't find any developers that have the bandwidth

If I start the project with all of the preliminary design/coding. You could get another group of developers to take the existing code and finish the project. It would be designed well enough for someone to take it over. Just would have to find a good React or VueJS developer (they are fairly common).

I would be willing to:

For a React/.NET/SQL based project

Start the project:

Deliverables:

The web service

Database Design

The majority of the screens with CRUD (Create, Read, Update, Delete) records

A search screen with results, Export out the results to excel for reporting

Payment Processing Screen

Total Estimated Hours: 80 @ 100.00 per hour \$8,000.00 (50% down to start), 50% at time of deliverable

Advantage(s):

React, SQL Server, .NET – a good scalable stack that any front-end developer (with .net) skills could pick up and finish.

The up front cost is cheap for what you are getting

For a MEVN based project (More of a MVP that could scale**) depending on preference**

Start the project:

Deliverables:

The web service

Database Design

The majority of the screens with CRUD (Create, Read, Update, Delete) records

A search screen with results

Payment Processing Screen

Total Estimated Hours: 80 @ 100.00 per hour \$8,000.00 (50% down to start), 50% at time of deliverable

Advantage(s):

It's a good tech stack (VueJS is easier to code in)

Cheaper to Host and get something up pretty quick for an MVP

The up front cost is cheap for what you are getting

Work that would be left

Authentication:

(This can already be done with the preliminary coded project – The project that I would develop)

If you went with the MEVN stack – Heroku (JWT Authentication is already wired up). We would just have to change to fit specific for the business.

If React, the authentication can be written in a different way (Even for JWT above).

Charting/Metrics

U/X Design (Getting the UI cleaned up)

Estimated cost left to finish (Rough), Depending on who you go with

Anywhere between **\$20,000 - \$30,000 +-.** You get bids to save \$\$\$

Good Sites to shop for developers:

<https://www.upwork.com/>

<https://www.toptal.com/>

Current App Screen Design – DONE***

Pharma Table/Screen:

Create the profile for the Pharma Company

Fields:

Name

Phone

Email

Password

Confirm Password

Medical Group Screen (i.e Herman , Methodist) – DONE***

Name

Address

Office

Mobile

Email

Password

Confirm Password

Physicians – DONE***

Title

First Name

M.I

Last Name

NPI

License ID

Doctor Email

Password

Confirm Password

Manager Screen – DONE*

First

Last
Territory Code/Name
Division
District
Phone
Mobile
Email

Rep Profile = DONE – DONE **
Manager Name (Dropdown)
First Name
Last Name
Territory Name/ Number
Phone
Email
Annual Budget

Receipts Screen– DONE **
Rep Dropdown
Restaurant
Amount
Shared Amount
Date
Event
Doctors
Notes
Upload Image