**These are all the NPM installs I did. It might work for everyone if you just do “npm install” in the Express file, and “npm install” in the Client(React) folder. But if It gives errors, this is a complete list. It should come up with a page with 2 links, Home and Secret. Home should say “Welcome” and Secret should link to “The password is potato”**

**RUN IN CMD WITH NPM START – shows on localhost:3000**

**BACKEND**

npm install

npm install –save-dev nodemon

npm install --save sqlite3

npm install --save sequelize

npm install --save bcryptjs (i installed bcryptjs but we are not using it yet)

npm install concurrently

**FRONT END (CLIENT FOLDER)**

npm install

npm install -g react-router-dom (I think this only needs installed once since it is a global install... if getting error run npm install react-router-dom in client folder)

**LIST OF STEPS OF WHAT WAS DONE--**

Created Express backend just like Woz lessons showed and created models/migrations for database.

Created the React frontend file (client) inside the express file with “npx create-react-app client” in command prompt.

“Concurrently” which was installed above, allows Nodemon and npm start to run at the same time so the Frontend and Backend will sync together and run at once. In command prompt inside the Express file(make sure you’re not in the client file) you have to run “npm start” and the frontend and backend will run at localhost 3000 instead of having to run both separately. Concurrently has to be put into the Express package.json. It is on line 6 in our code with the “start” for scripts.

React app is running at localhost 3000 and Express is running on 3000. Cannot have 2 running on the same port. Went in the Express “bin” folder and changed the var port to 3005 so Express does not run on same port as React.

In the client package.json, we needed to add a “proxy” at the end to port 3005, so the React App knows that it can run 3005 (which is our Express backend)

Changed the client/src/index.js to use the react-router-dom

router.get on Express routes/index.js links React components.

client App.js is where the routes are (React) and top you need to import your routes (ex: Home, Secret) depending on where the pages are located.

the fetch on each React component is your link on the Express index.js

routes/users.js on Express has a post with the requirements of the sign up. No actual form was created yet, but the post is connected to the database and can be tested in Postman. Make sure var models is defined at the top because that is needed to send the info to the database.

**In Postman to test the Route and make sure it posts to the database ---**

-Change drop down menu to POST

-The post route is [**http://localhost:3000/users/api/register**](http://localhost:3000/users/api/register)

**-** Go to “Body”

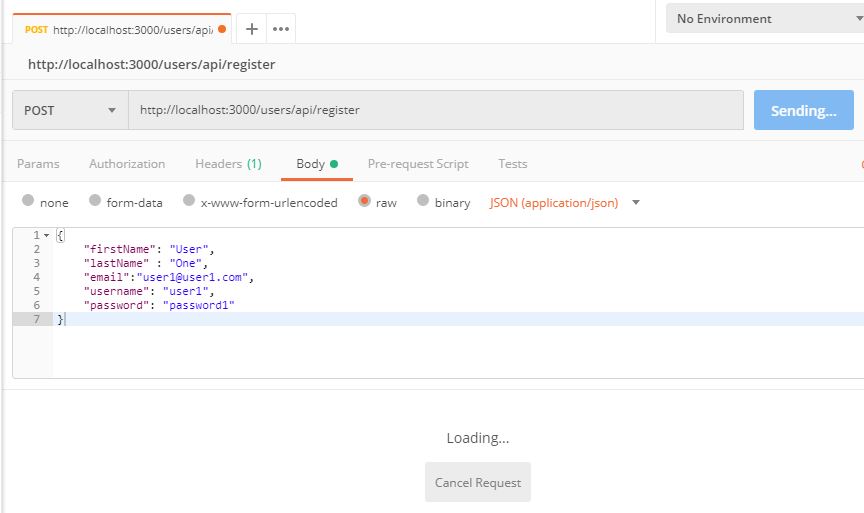
-Change to “raw”

-Click the “text” dropdown on the right and change to “JSON”

- This is where you can post your information to test the route. It has to be written in a JSON object in the same format that we wrote it in the “users” route. **The application must be running or it will not get a response.**

**-**To see if the post worked, load the database in SQLectron and run **select \* from users; …** you’ll have to create different information now that user1 already exists.

**Pictures Below On Postman and SQLectron**



Sqlectron

