Homework 2

Set-up

Download the starter file provided on canvas under Homework Two. Unzip the files and save them in your directory of choice. Open the "Homework2" folder in your code editor of choice. Use the terminal in your code editor to type the following commands. (Alternatively use the terminal, however, navigate to "homework2" folder before typing the following commands.)

```
npm install

npm audit fix

npm  start
```

You will see a browser tab open on your default browser(Chrome recommended) with the localhost information and a success message. As a warm-up, find the source code of the message displayed on the screen and delete it while also getting comfortable with the provided source code. (Hint: In case you can't figure it out chrome dev tools and React dev tools are a tap away.)

Navigate to GitHub on your browser of choice and create a new repository without a README. Follow the procedure highlighted in the first assignment to commit the current state of the project and push the changes.

Exercise.

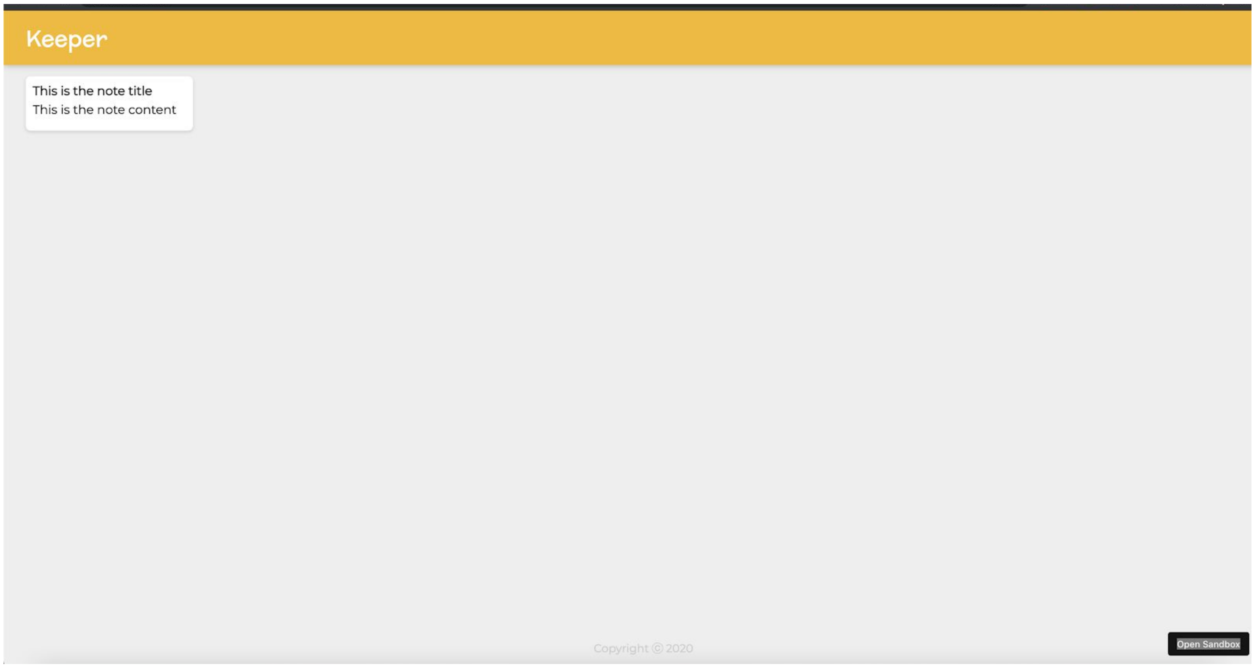
Section 1 - components

1. Create a Header.jsx component that renders a <header> to show the keeper app name in an h1(The styling has already been applied).

2. Create a Footer.jsx component that renders a <footer> to show the copyright message in a <p> with a dynamically updated year.

3. Create a Note.jsx component to show a <div> element with <h1> for a title and a <p> for the content

4. Use the CSS classes in the styles.css to apply styling to achieve this look below. (Assume the grey button at the bottom right written "open sandbox")



This is the note title
This is the note content

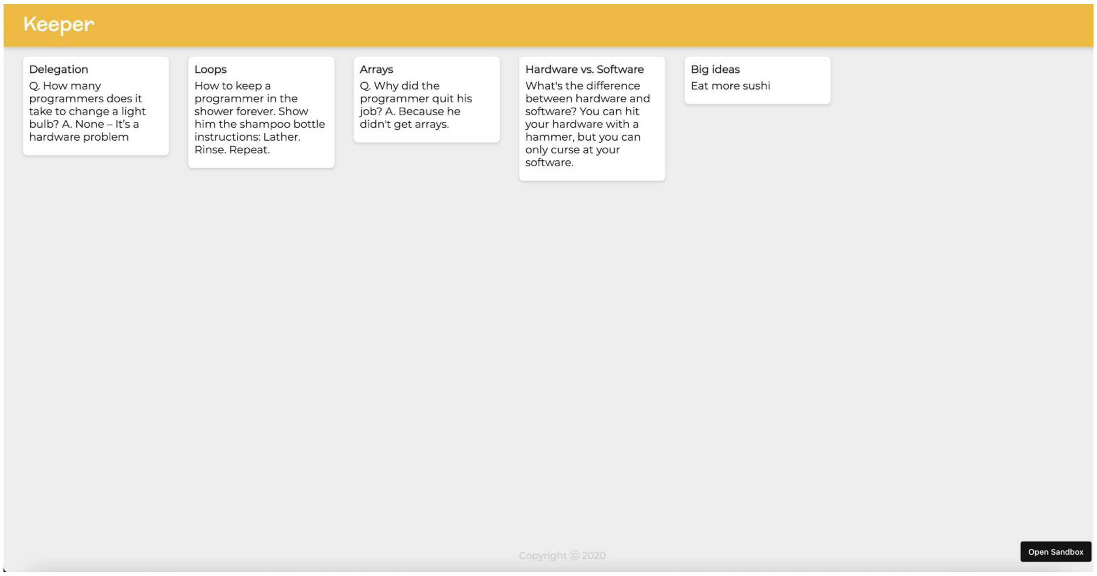
5. Once you are satisfied with the current progress, add all the changes through(git add .) and commit with the message “completed section1” and push them to your remote GitHub repository.

Section 2 - props and data mapping.

1. Render all the notes inside notes.js as a separate Note Component.

2. Confirm that everything matches the image shown below of the final iteration.

3. Once you are satisfied with the progress, add all the changes through(git add .) and commit with the message “completed section2” and push them to your remote GitHub repository.



Delegation

Q. How many programmers does it take to change a light bulb? A. None – It's a hardware problem

Loops

How to keep a programmer in the shower forever. Show him the shampoo bottle instructions: Lather. Rinse. Repeat.

Arrays

Q. Why did the programmer quit his job? A. Because he didn't get arrays.

Hardware vs. Software

What's the difference between hardware and software? You can hit your hardware with a hammer, but you can only curse at your software.

Big ideas

Eat more sushi

Copyright © 2020

Submission.

After pushing and making any changes to your project(including but not limited to optionally customizing the CSS to match any design you are interested in.) submit the link to the GitHub repository on canvas under homework2.

Further Reading.

You may have noticed we did not explicitly create a react app. What ***npm install*** does is go through the **package.json** and install all the dependencies listed and all the support libraries they require into a folder called **node-modules** that appears once you run the command. **package-lock.json is** automatically generated for any operations where npm modifies either the **node_modules** tree or **package.json**. It describes the exact tree that was generated, such that subsequent installs are able to generate identical trees, regardless of intermediate dependency updates. We ran ***npm audit fix,*** to fix and update any system-dependent dependencies. Then finally ***npm start*** runs the react scripts.

Further, you will notice that we have a **.gitignore** file that has ***node-modules,*** and ***package-lock.json*** listed. This is just the way to tell git to ignore these files when adding our files to local repositories and subsequently when they are pushed to GitHub.

Why????????.

Package.json saves the expense of uploading huge files of system-dependent node-modules into GitHub. Instead, you tell git to ignore them and once any developer clones your 1MB project on Github, they can simply run ***npm install*** and fully build the 100MB project locally. Taking a deeper look, the node-modules you download are also packages by other developers. To get the full picture, read here.