Create a new react app

Project link

https://github.com/shreyamalogi/Google-Keep-Clone

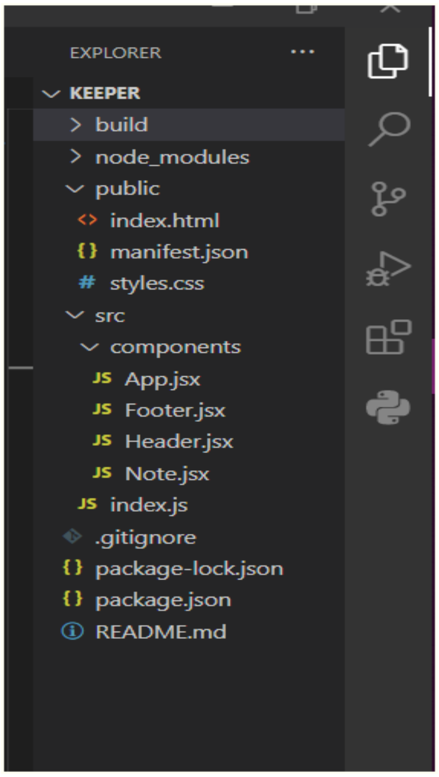
Click on commits, view versions by browsing files
Or git log and select the version you want to view

```
npx create-react-app my-app
cd my-app
npm start
```

Version 1- challenges to be solved

```
1. Create a new React app.
2. Create a App.jsx component.
3. Create a Header.jsx component that renders a <header> element
to show the Keeper App name in an <h1>.
4. Create a Footer.jsx component that renders a <footer> element
to show a copyright message in a <p> with a dynamically updated year.
5. Create a Note.jsx component to show a <div> element with a
<h1> for a title and a <p> for the content.
6. Make sure that the final website is styled like the example shown
here:
https://l1pp6.csb.app/

HINT: You will need to study the classes in teh styles.css file to
apply styling.
```

CLASSIC THINGS



```
EXPLORER ...
KEEPER
  > build
  > node_modules
  v public
    index.html
    manifest.json
    styles.css
  v src
    v components
      App.jsx
      Footer.jsx
      Header.jsx
      Note.jsx
    index.js
  .gitignore
  package-lock.json
  package.json
  README.md
```

All the react elements start with capital letter which is called pascal case

To render components to our app.js we need to import react then export the function

Version 1: Solution

App.jsx

```
import React from "react";

function App(){
    return <div>
        <h1>hello App</h1>
    </div>
}

export default App;
```

header.jsx

```
import React from "react";

function Header(){
  return <Header>
    <h1>GOOGLE KEEP </h1>
  </Header>
}

export default Header;
```

footer.jsx

```
import React from "react";

function Footer(){

    const currentYear = new Date().getFullYear()

    return<footer>
        <p>
        Copyright {currentYear} Shreya Malogi
    </p>

    </footer>
}

export default Footer;
```

note.jsx

To get css we need to put it under div className

```
import React from "react";

function Note(){
    return (
    <div className = "note">

    <h1>This is the title </h1>
    <p> This is the content </p>

    </div>
    )
}

export default Note;
```

WE CAN RENDER THE HEADER.JSX COMPONENT INTO OUR APP.JSX

Make sure that the html codes are always inside div
Component =function
The function which we are rendering will have a self closing tag

Rendering app.jsx into app.js

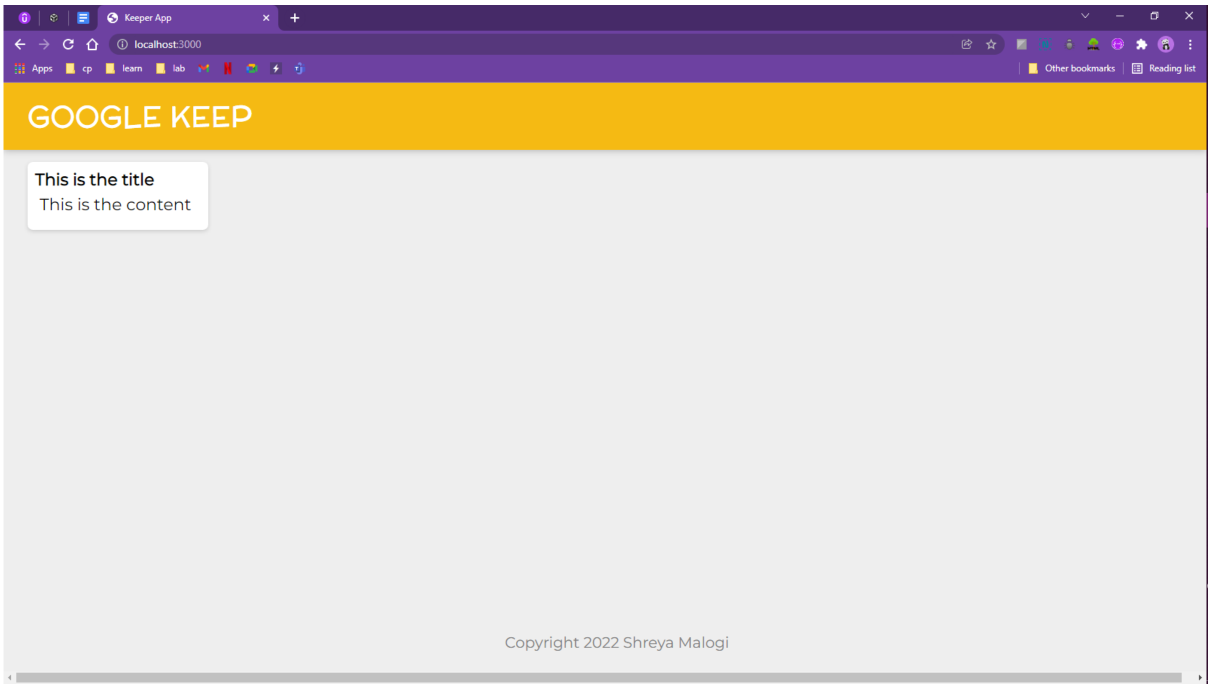
```
import React from "react";
import ReactDOM from "react-dom";
import App from "./components/App";

ReactDOM.render(
  <div>
    <App />
  </div>,
  document.getElementById("root")

);
```

Rendering header.jsx, footer.jsx, note.jsx into app.jsx

Importing all the other components to main app.jsx and then rendering it as html tags inside the function

```
import React from "react";
import Header from "./Header";
import Footer from "./Footer";
import Note from "./Note";

function App(){
    return (<div>
        <Header />
        <Note />
        <Footer />
    </div>
    );
}

export default App;
```

V1 Output :



GOOGLE KEEP

This is the title
This is the content

Copyright 2022 Shreya Malogi

Version 2

Rendering custom content through props
Passing the props from app.js to note.jsx
To render a single note then, in app.jsx we write customised

```
function App(){
    return (<div>
        <Header />
        <Note
            title = "This is the title "
            content = "This is the content"
         />
        <Footer />
    </div>
    );
}

export default App;
```

And in note.jsx we write the props

```
import React from "react";

function Note(props){
    return (
    <div className = "note">

    <h1>{props.title} </h1>
    <p> {props.content} </p>

    </div>
    )
}

export default Note;
```

Add a new file notes.js as an array of notes

```
const notes = [
  {
    key: 1,
    title: "Delegation",
    content:
      "Q. How many programmers does it take to change a light bulb? A.
None – It’s a hardware problem"
  },
  {
    key: 2,
    title: "Loops",
    content:
      "How to keep a programmer in the shower forever. Show him the
shampoo bottle instructions: Lather. Rinse. Repeat."
  },
  {
    key: 3,
    title: "Arrays",
    content:
      "Q. Why did the programmer quit his job? A. Because he didn't get
arrays."
  },
  {
    key: 4,
    title: "Hardware vs. Software",
    content:
      "What's the difference between hardware and software? You can hit
your hardware with a hammer, but you can only curse at your software."
  }
];

export default notes;
```

Then in app.jsx

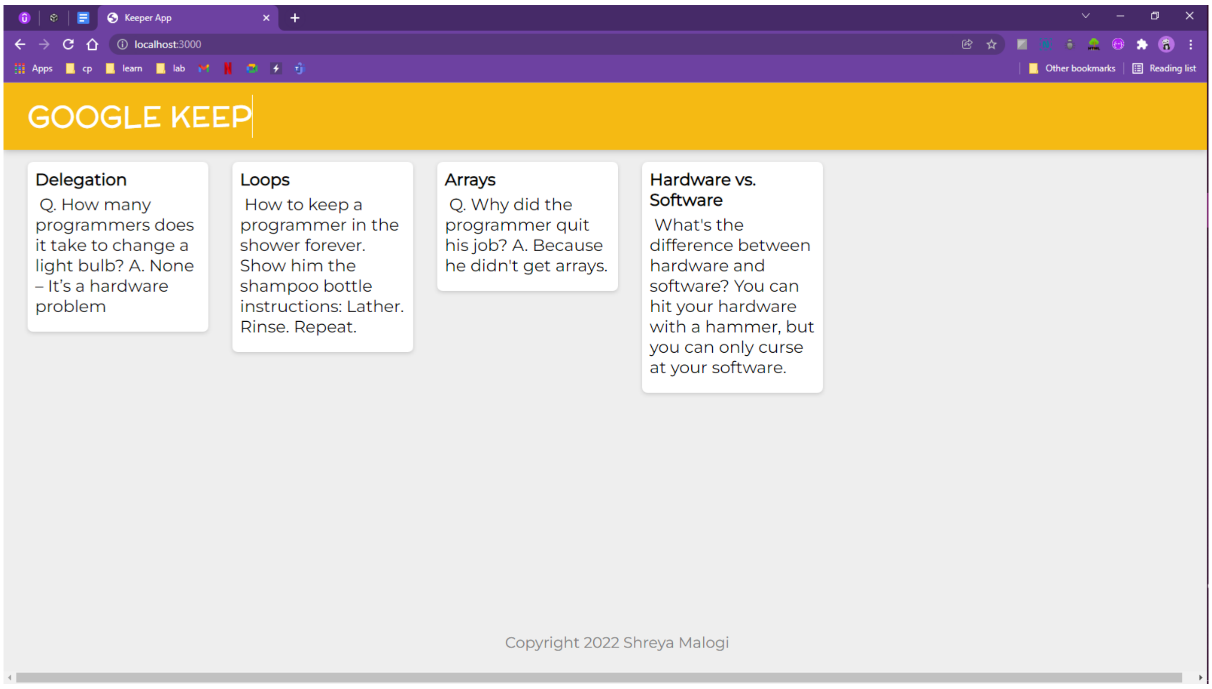
```
import React from "react";
import Header from "./Header";
import Footer from "./Footer";
import Note from "./Note";
import notes from "../notes";

//keep a function which creates notes and we gonna pass a single note
item into it and renders a custom  note component
//and which will return a note component which will have props
//this title and content props must be there is notes.js
//Whenever we want to loop through or map through a dynamic array we
must have a key


function createNotes(noteItem){
    return <Note
         key = {noteItem.key}
```

```
        title = {noteItem.title}
        content = {noteItem.content}
    />
}


//mapping our notes with create notes
function App(){
    return (<div>
        <Header />
            {notes.map(createNotes)}
        <Footer />
    </div>
    );
}

export default App;
```

Refactoring it as

```
function App(){
    return (
        <div>
        <Header />
        {notes.map ((noteItem) =>
        <Note
         key = {noteItem.key}
         title = {noteItem.title}
         content = {noteItem.content}
        />
    )}

        <Footer />
    </div>
    );
}

export default App;
```

Output:



GOOGLE KEEP

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.

Copyright 2022 Shreya Malogi

Version 3

```
CHALLENGE:
1. Implement the add note functionality.
- Create a constant that keeps track of the title and content.
- Pass the new note back to the App.
- Add a new note to an array.
- Take an array and render separate Note components for each item.

2. Implement the delete note functionality.
- Callback from the Note component to trigger a delete function.
- Use the filter function to filter out the item that needs
deletion.
- Pass a id over to the Note component, pass it back to the App
when deleting.

This is the end result you're aiming for:
https://pogqj.csb.app/
```

App.jsx

```jsx
import React from "react";
import Header from "./Header";
import Footer from "./Footer";
import Note from "./Note";
import CreateArea from "./CreateArea";

function App() {
    function addNote(note){

    }
  return (
    <div>
      <Header />
      <CreateArea onAdd={addNote} />

      <Note key={1} title="Note title" content="Note content" />
      <Footer />
    </div>
  );
}

export default App;
```

In note.jsx

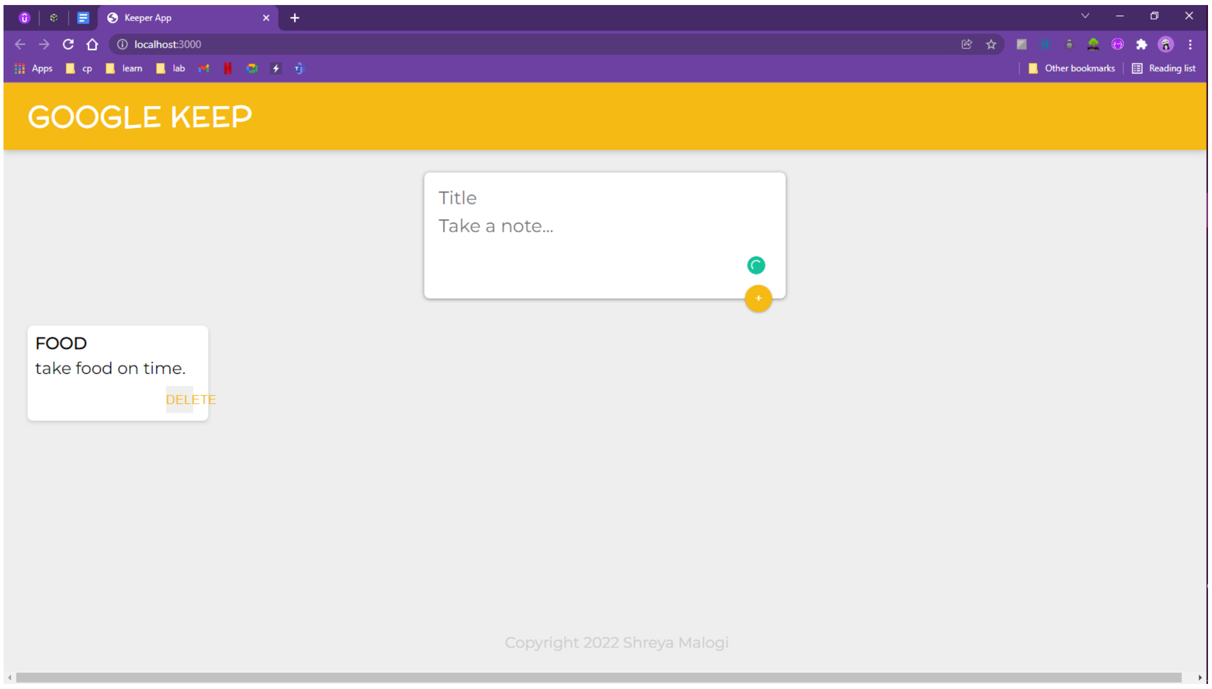
```jsx
import React from "react";

function Note(props) {
  function handleClick() {
    props.onDelete(props.id);
  }

  return (
    <div className="note">
      <h1>{props.title}</h1>
      <p>{props.content}</p>
      <button onClick={handleClick}>-</button>
    </div>
  );
}

export default Note;
```

Delete notes.js
Create createArea.jsx

```
import React, { useState } from "react";
import Header from "./Header";
import Footer from "./Footer";
import Note from "./Note";
import CreateArea from "./CreateArea";

function App() {
  const [notes, setNotes] = useState([]);

  function addNote(newNote) {
    setNotes(prevNotes => {
      return [...prevNotes, newNote];
    });
  }

  function deleteNote(id) {
    setNotes(prevNotes => {
      return prevNotes.filter((noteItem, index) => {
        return index !== id;
      });
    });
  }

  return (
    <div>
      <Header />
      <CreateArea onAdd={addNote} />
      {notes.map((noteItem, index) => {
        return (
          <Note
```

```
            key={index}
            id={index}
            title={noteItem.title}
            content={noteItem.content}
            onDelete={deleteNote}
          />
        );
      })}
      <Footer />
    </div>
  );
}

export default App;
```

Output


Keeper App

localhost:3000

Apps cp learn lab

Other bookmarks Reading list

GOOGLE KEEP

Title

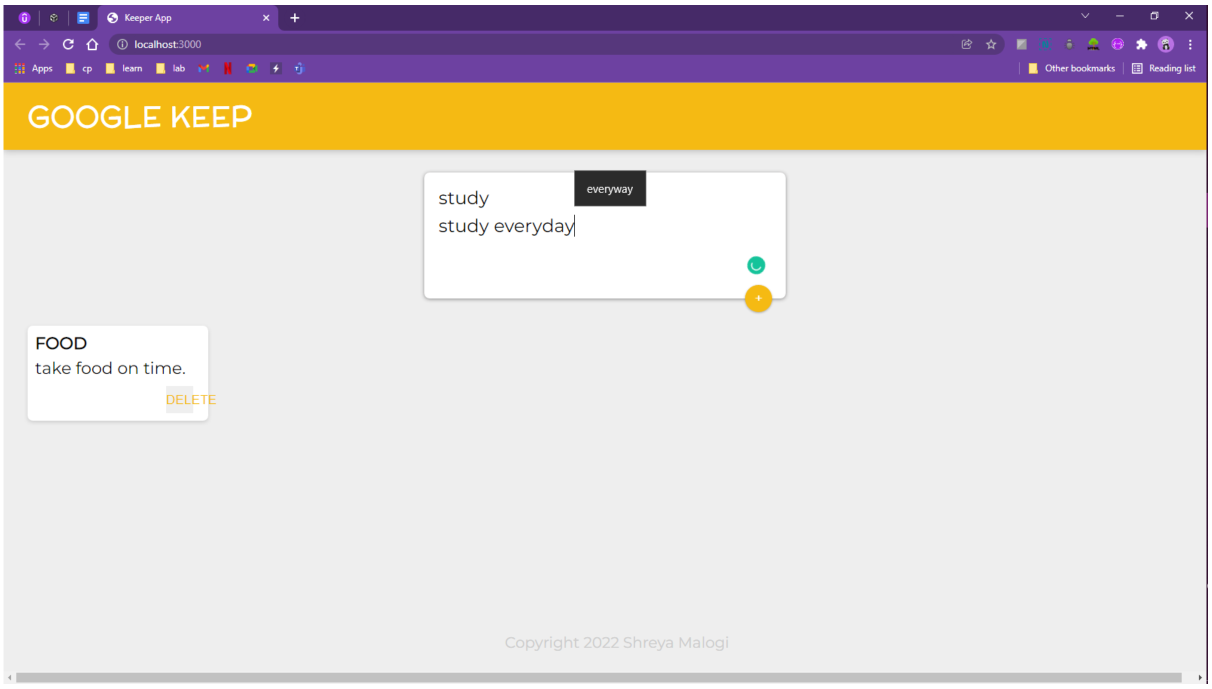
Take a note...

FOOD

take food on time.

DELETE

Copyright 2022 Shreya Malogi


Keeper App

localhost:3000

Apps cp learn lab

Other bookmarks Reading list

GOOGLE KEEP

study

everyway

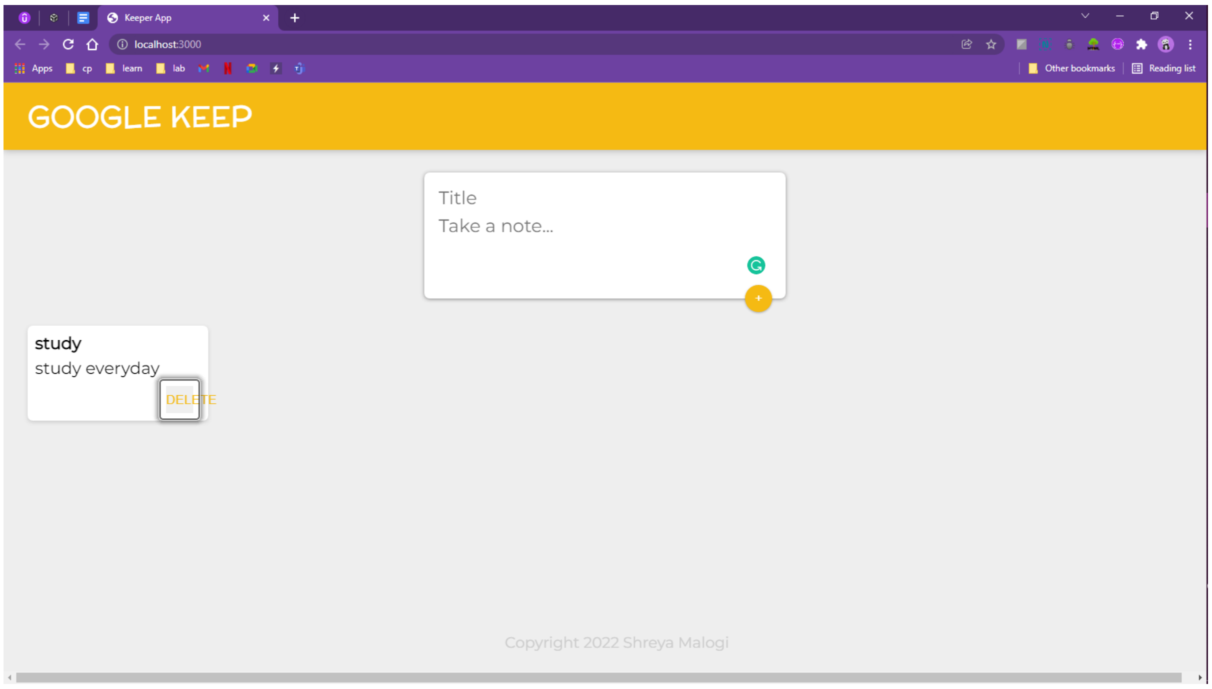
study everyday

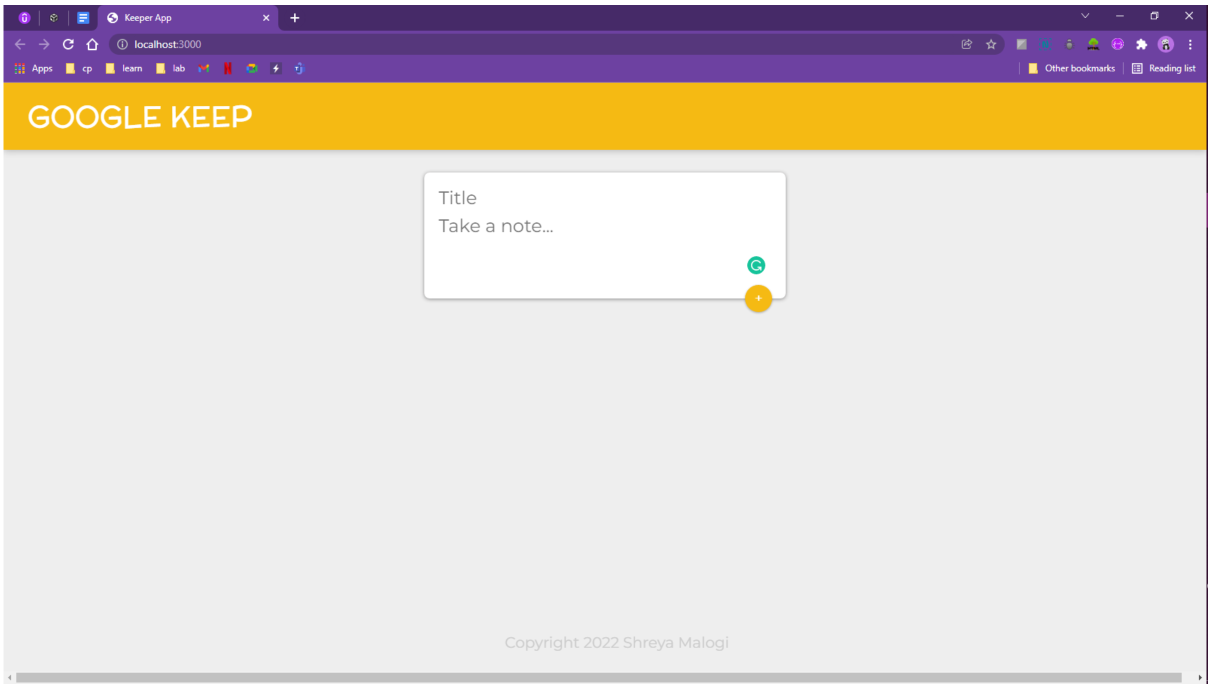
FOOD

take food on time.

DELETE

Copyright 2022 Shreya Malogi

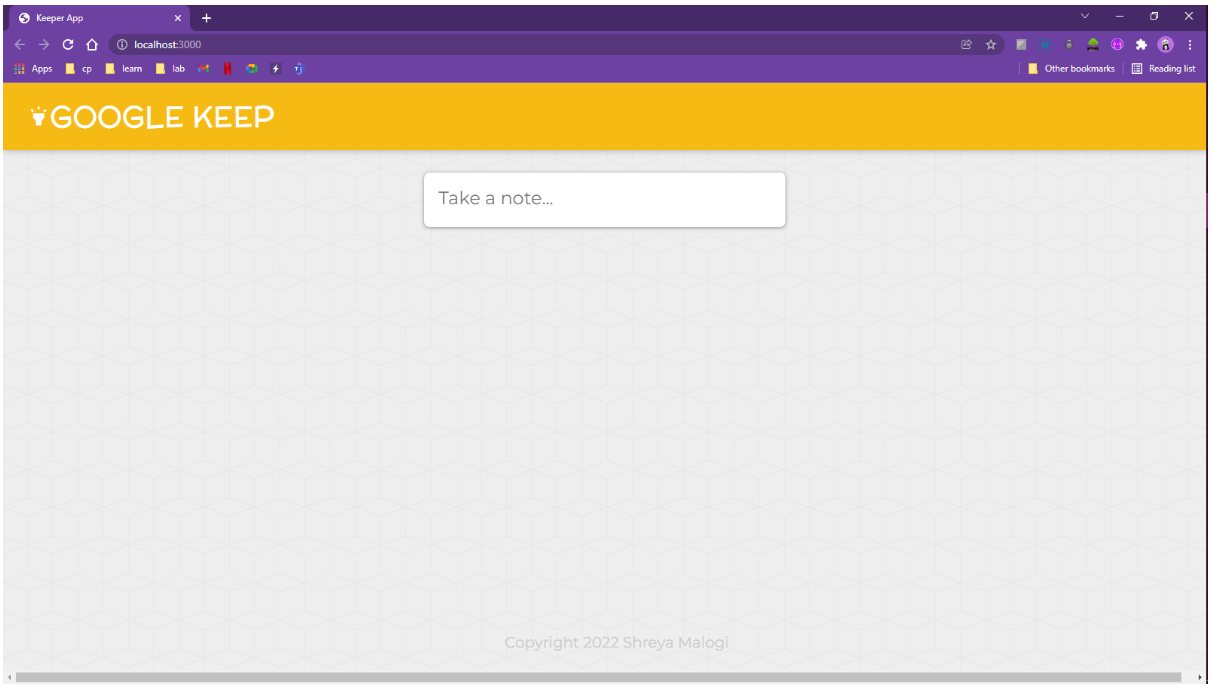


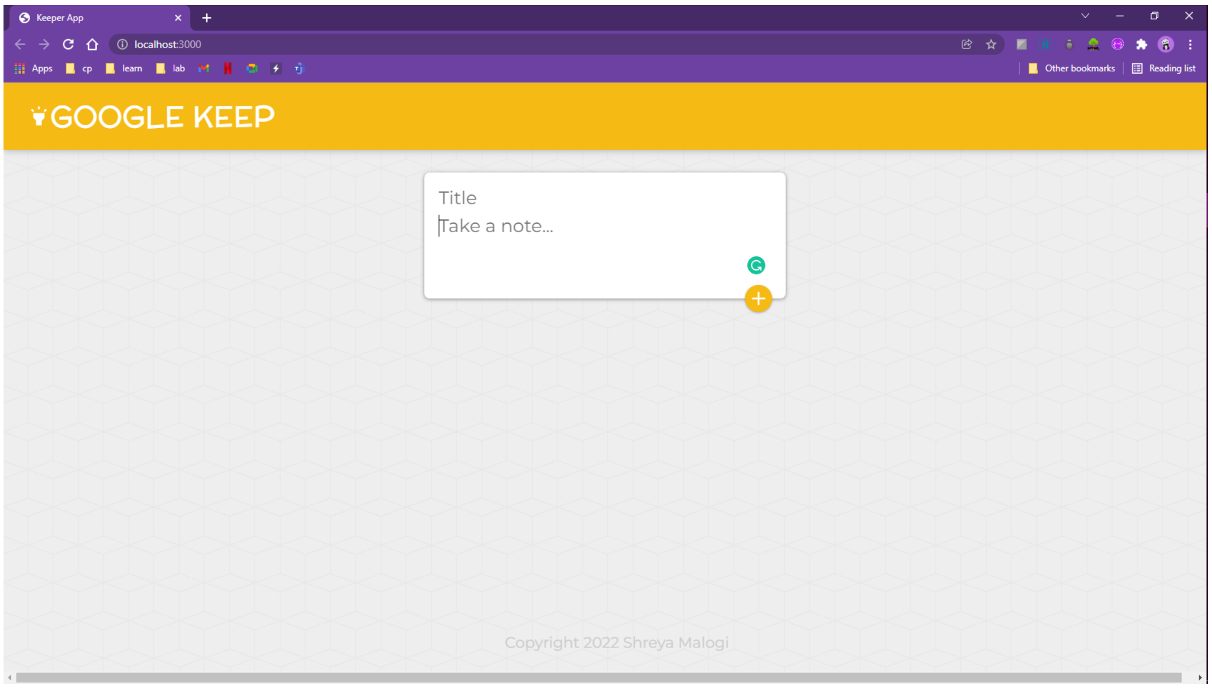


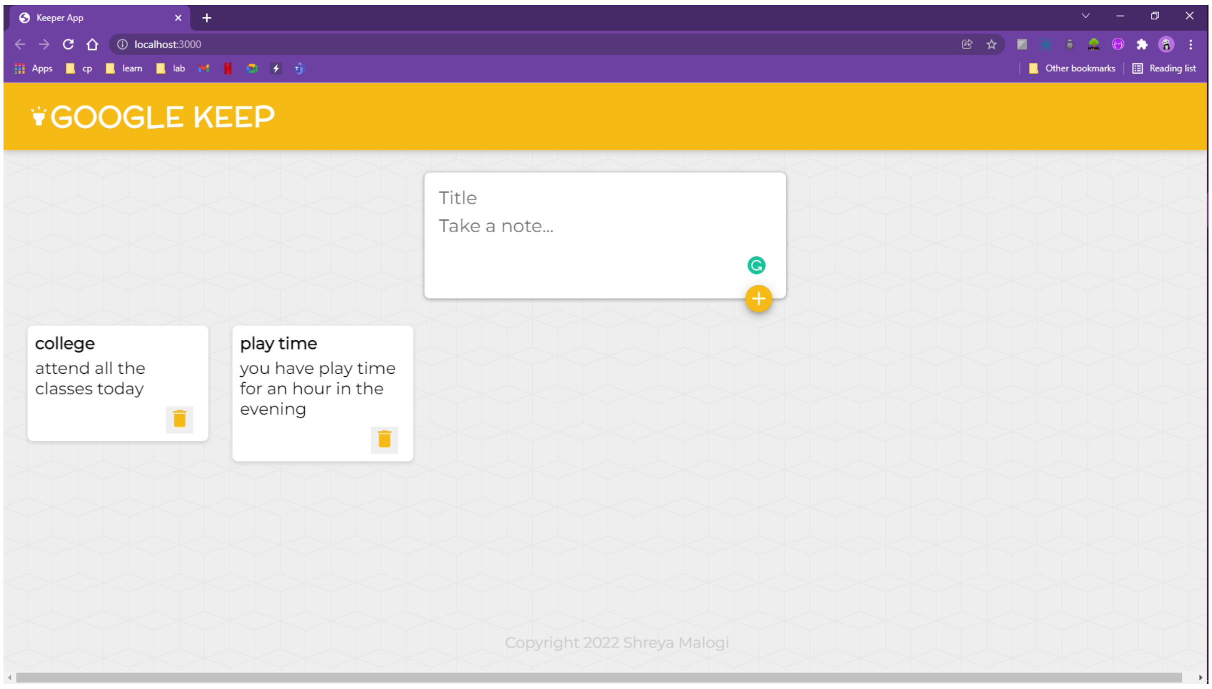
Version 4

Install dependencies

[material ui](#) head over to docs and try to implement importing the icons


Keeper App
localhost:3000
GOOGLE KEEP
Take a note...
Copyright 2022 Shreya Malogi


Keeper App
localhost:3000
GOOGLE KEEP
Title
Take a note...
Copyright 2022 Shreya Malogi



college
attend all the classes today

play time
you have play time for an hour in the evening

Copyright 2022 Shreya Malogi