

Lab Exercise 5

Code:

#index.js

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

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

#app.js

```
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} />
    </div>
  );
}
```

```

        {notes.map((noteItem, index) => {
            return (
                <Note
                    key={index}
                    id={index}
                    title={noteItem.title}
                    content={noteItem.content}
                    onDelete={deleteNote}
                />
            );
        })}
        <Footer />
    </div>
);
}

export default App;

```

#create area.jsx

```

import React, { useState } from "react";

function CreateArea(props) {
    const [note, setNote] = useState({
        title: "",
        content: ""
    });

    function handleChange(event) {
        const { name, value } = event.target;

        setNote(prevNote => {
            return {
                ...prevNote,
                [name]: value
            };
        });
    }

    function submitNote(event) {
        props.onAdd(note);
        setNote({
            title: "",
            content: ""
        });
        event.preventDefault();
    }

    return (

```

```

    <div>
      <form>
        <input
          name="title"
          onChange={handleChange}
          value={note.title}
          placeholder="Title"
        />
        <textarea
          name="content"
          onChange={handleChange}
          value={note.content}
          placeholder="Take a note..."
          rows="3"
        />
        <button onClick={submitNote}>Add</button>
      </form>
    </div>
  );
}

export default CreateArea;

```

#footer.jsx

```

import React from "react";

function Footer() {
  const year = new Date().getFullYear();
  return (
    <footer>
      <p>Copyright © {year}</p>
    </footer>
  );
}

export default Footer;

```

#header.jsx

```

import React from "react";

function Header() {
  return (
    <header>
      <h1>Keeper</h1>
    </header>
  );
}

export default Header;

```

```
    </header>
  );
}

export default Header;
```

#note.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}>DELETE</button>
    </div>
  );
}

export default Note;
```

#index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Keeper App</title>
    <link font-family: "Montserrat", sans-serif;
    href="https://fonts.googleapis.com/css?family=McLaren|Montserrat&display=swap"
    rel="stylesheet" />
    <link rel="stylesheet" href="styles.css" />
  </head>

  <body>
    <div id="root"></div>
    <script src="../../src/index.js" type="text/jsx"></script>
  </body>
</html>
```

#style.css

```
* {
  padding: 0;
  margin: 0;
  box-sizing: border-box;
}

html {
  font-family: "Montserrat", sans-serif;
}

body {
  background: #eee;
  padding: 0 16px;
}

header {
  background-color: #f5ba13;
  margin: auto -16px;
  padding: 16px 32px;
  box-shadow: 0 0 10px 0 rgba(0, 0, 0, 0.3);
}

header h1 {
  color: #fff;
  font-family: "McLaren", cursive;
  font-weight: 200;
}

footer {
  position: absolute;
  text-align: center;
  bottom: 0;
  width: 100%;
  height: 2.5rem;
}

footer p {
  color: #ccc;
}

.note {
  background: #fff;
  border-radius: 7px;
  box-shadow: 0 2px 5px #ccc;
  padding: 10px;
  width: 240px;
  margin: 16px;
  float: left;
}

.note h1 {
  font-size: 1.1em;
}
```

```
    margin-bottom: 6px;
}

.note p {
    font-size: 1.1em;
    margin-bottom: 10px;
    white-space: pre-wrap;
    word-wrap: break-word;
}

.note button {
    position: relative;
    float: right;
    margin-right: 10px;
    color: #f5ba13;
    border: none;
    width: 36px;
    height: 36px;
    cursor: pointer;
    outline: none;
}

form {
    position: relative;
    width: 480px;
    margin: 30px auto 20px auto;
    background: #fff;
    padding: 15px;
    border-radius: 7px;
    box-shadow: 0 1px 5px rgb(138, 137, 137);
}

form input,
form textarea {
    width: 100%;
    border: none;
    padding: 4px;
    outline: none;
    font-size: 1.2em;
    font-family: inherit;
    resize: none;
}

form button {
    position: absolute;
    right: 18px;
    bottom: -18px;
    background: #f5ba13;
    color: #fff;
    border: none;
    border-radius: 50%;
}
```

```
width: 36px;
height: 36px;
box-shadow: 0 1px 3px rgba(0, 0, 0, 0.3);
cursor: pointer;
outline: none;
}
```

Output:

