PROMINEO TECH

Web App Design with React Week 3 Coding Assignment

Points possible: 70

Category	Criteria	% of Grade
Functionality	Does the code work?	25
Organization	Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear.	25
Creativity	Student solved the problems presented in the assignment using creativity and out of the box thinking.	25
Completeness	All requirements of the assignment are complete.	25

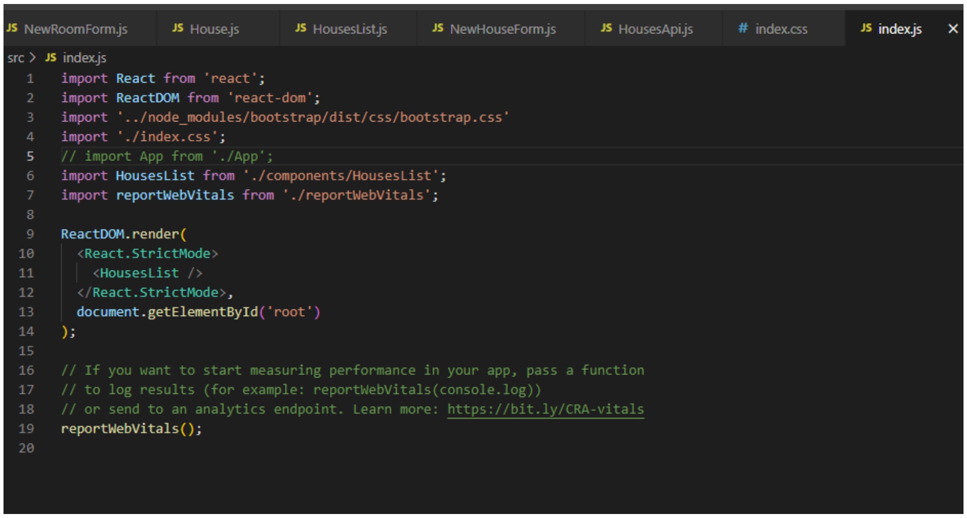
Instructions: In VS Code, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week's assignments and push this document, with your JavaScript project code, to the repository. Add the URL for this week's repository to this document where instructed and submit this document to your instructor when complete.

Coding Steps:

1. Using the Houses API, or any open API of your choice you can find online, create a single page that allows for all 4 crud operations to be performed on a resource from that API. Create a React component (or more, if needed) to represent the resource. Make all forms and other necessary UI pieces their own components as reasonable.

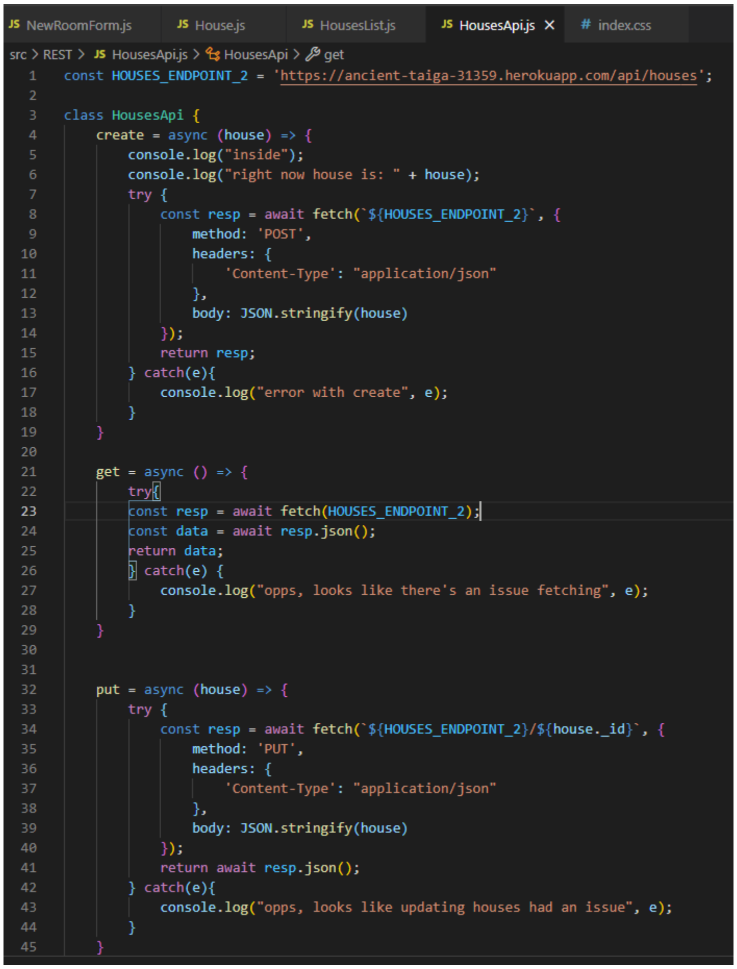
Screenshots of Code:

Index.js

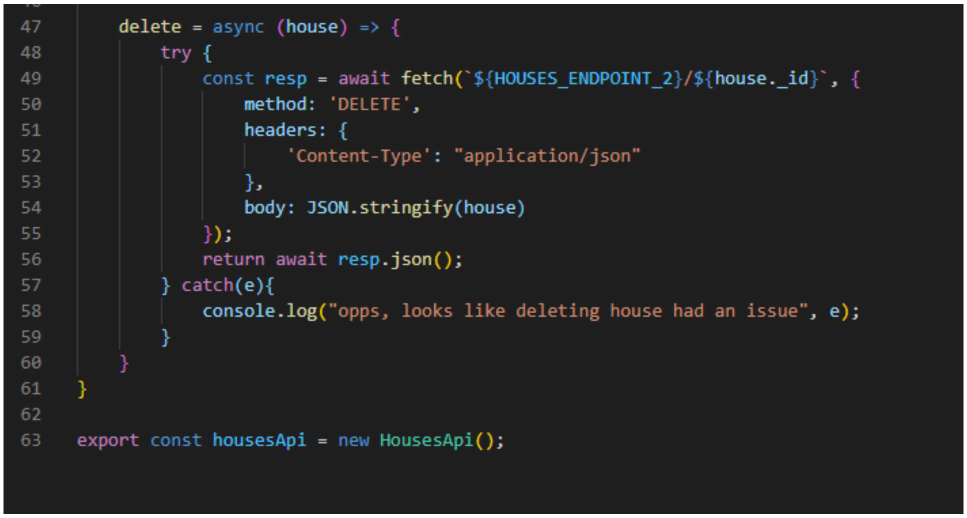


```
import React from 'react';
import ReactDOM from 'react-dom';
import '../node_modules/bootstrap/dist/css/bootstrap.css'
import './index.css';
// import App from './App';
import HousesList from './components/HousesList';
import reportWebVitals from './reportWebVitals';

ReactDOM.render(
  <React.StrictMode>
    <HousesList />
  </React.StrictMode>,
  document.getElementById('root')
);

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

HousesApi:

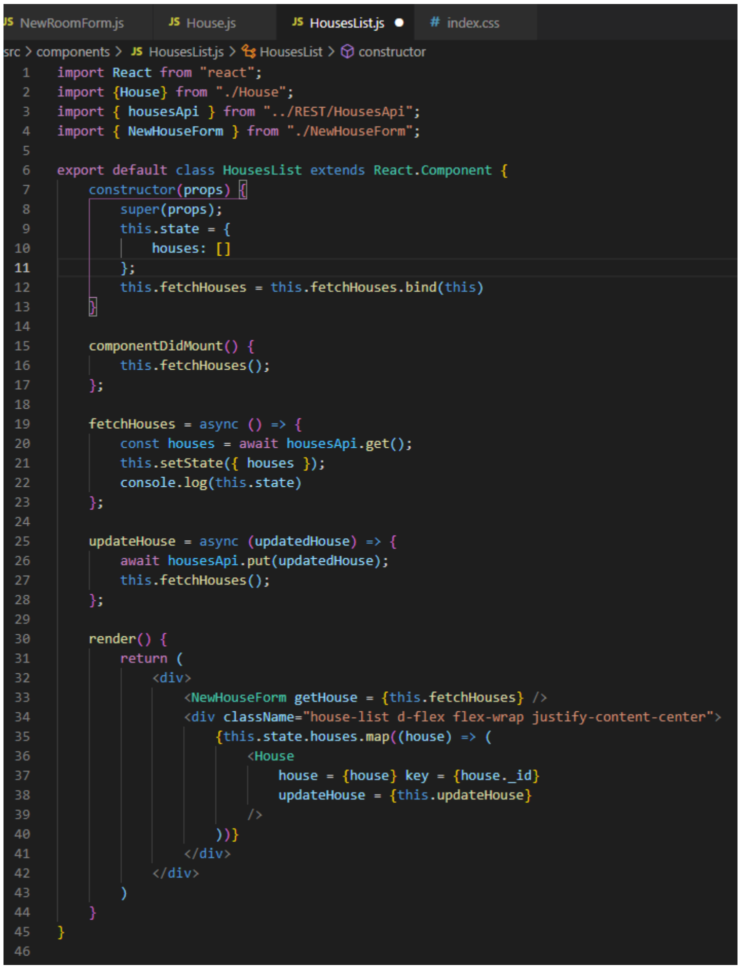


```
const HOUSES_ENDPOINT_2 = 'https://ancient-taiga-31359.herokuapp.com/api/houses';

class HousesApi {
    create = async (house) => {
        console.log("inside");
        console.log("right now house is: " + house);
        try {
            const resp = await fetch(`${HOUSES_ENDPOINT_2}`, {
                method: 'POST',
                headers: {
                    'Content-Type': "application/json"
                },
                body: JSON.stringify(house)
            });
            return resp;
        } catch(e){
            console.log("error with create", e);
        }
    }

    get = async () => {
        try{
        const resp = await fetch(HOUSES_ENDPOINT_2);
        const data = await resp.json();
        return data;
        } catch(e) {
            console.log("opps, looks like there's an issue fetching", e);
        }
    }


    put = async (house) => {
        try {
            const resp = await fetch(`${HOUSES_ENDPOINT_2}/${house._id}`, {
                method: 'PUT',
                headers: {
                    'Content-Type': "application/json"
                },
                body: JSON.stringify(house)
            });
            return await resp.json();
        } catch(e){
            console.log("opps, looks like updating houses had an issue", e);
        }
    }
```



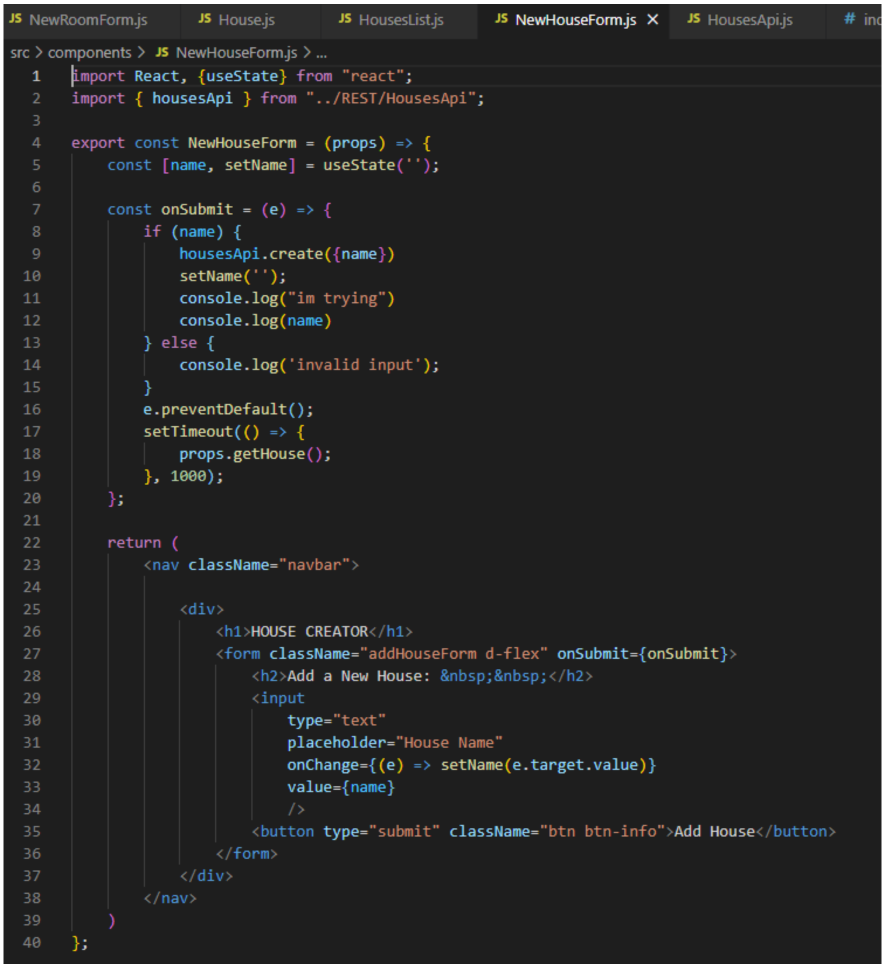
```
    delete = async (house) => {
        try {
            const resp = await fetch(`${HOUSES_ENDPOINT_2}/${house._id}`, {
                method: 'DELETE',
                headers: {
                    'Content-Type': "application/json"
                },
                body: JSON.stringify(house)
            });
            return await resp.json();
        } catch(e){
            console.log("opps, looks like deleting house had an issue", e);
        }
    }
}

export const housesApi = new HousesApi();
```

HousesList:


JS NewRoomForm.js
JS House.js
JS HousesList.js ●
# index.css
src > components > JS HousesList.js > HousesList > constructor

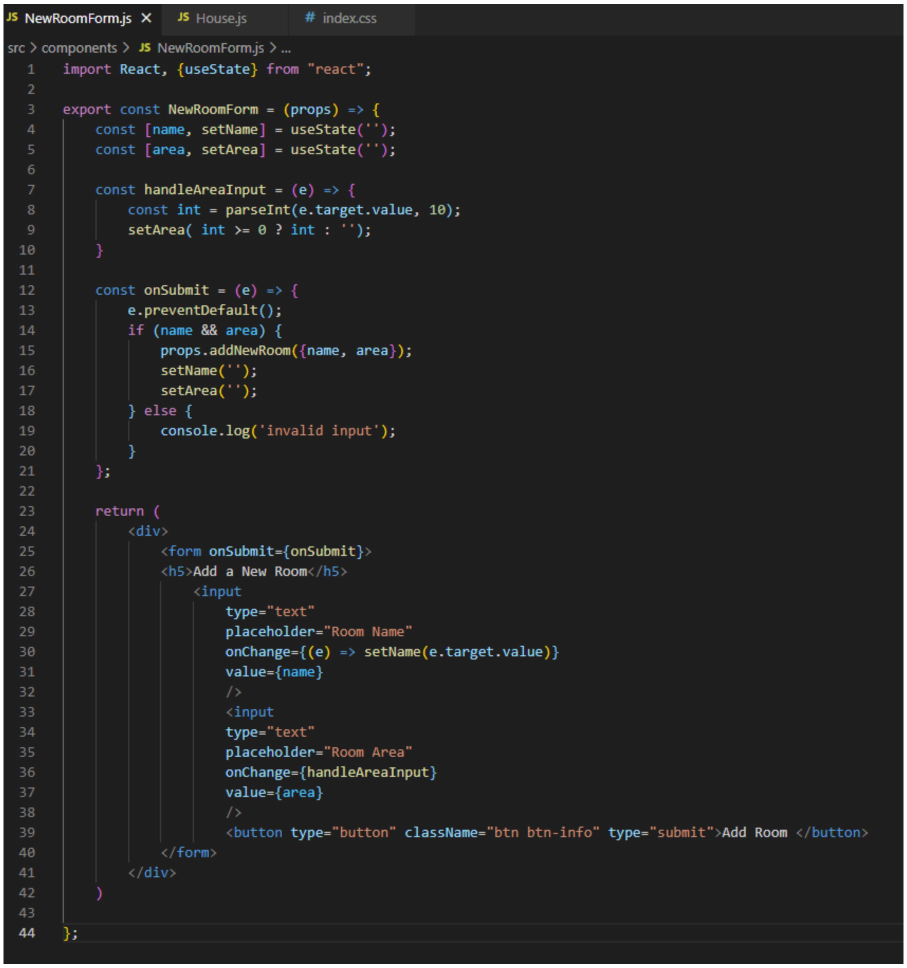
```
import React from "react";
import {House} from "./House";
import { housesApi } from "../REST/HousesApi";
import { NewHouseForm } from "./NewHouseForm";

export default class HousesList extends React.Component {
    constructor(props) {
        super(props);
        this.state = {
            houses: []
        };
        this.fetchHouses = this.fetchHouses.bind(this)
    }

    componentDidMount() {
        this.fetchHouses();
    };

    fetchHouses = async () => {
        const houses = await housesApi.get();
        this.setState({ houses });
        console.log(this.state)
    };

    updateHouse = async (updatedHouse) => {
        await housesApi.put(updatedHouse);
        this.fetchHouses();
    };

    render() {
        return (
            <div>
                <NewHouseForm getHouse = {this.fetchHouses} />
                <div className="house-list d-flex flex-wrap justify-content-center">
                    {this.state.houses.map((house) => (
                        <House
                            house = {house} key = {house._id}
                            updateHouse = {this.updateHouse}
                        />
                    ))}
                </div>
            </div>
        )
    }
}
```

New House Form:



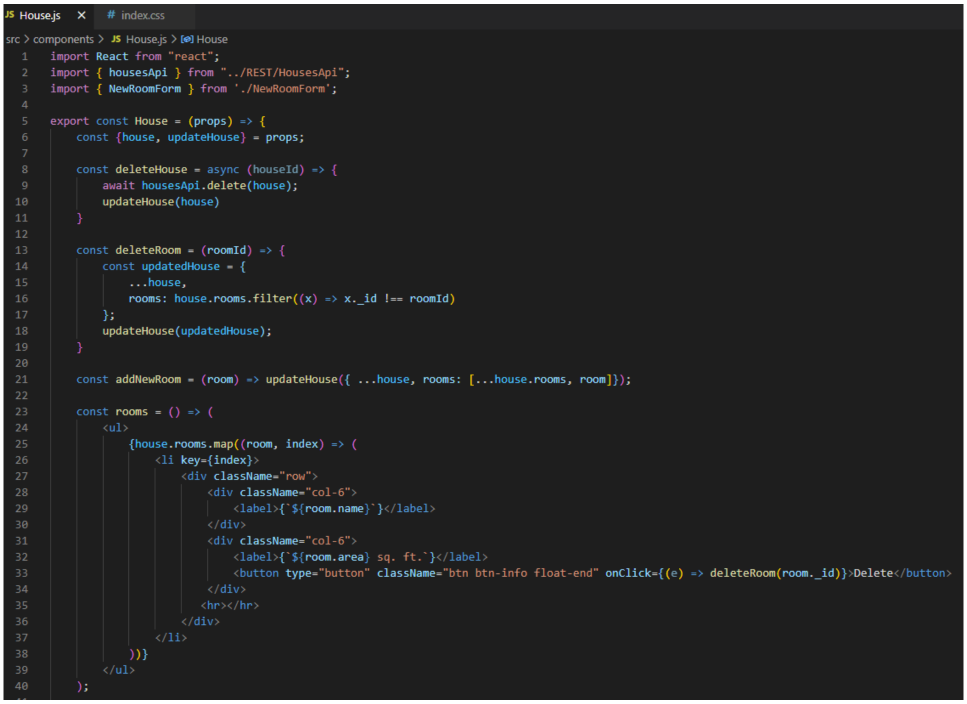
```
import React, {useState} from "react";
import { housesApi } from "../REST/HousesApi";

export const NewHouseForm = (props) => {
    const [name, setName] = useState('');

    const onSubmit = (e) => {
        if (name) {
            housesApi.create({name})
            setName('');
            console.log("im trying")
            console.log(name)
        } else {
            console.log('invalid input');
        }
        e.preventDefault();
        setTimeout(() => {
            props.getHouse();
        }, 1000);
    };

    return (
        <nav className="navbar">

            <div>
                <h1>HOUSE CREATOR</h1>
                <form className="addHouseForm d-flex" onSubmit={onSubmit}>
                    <h2>Add a New House: &nbsp;&nbsp;</h2>
                    <input
                        type="text"
                        placeholder="House Name"
                        onChange={(e) => setName(e.target.value)}
                        value={name}
                        />
                    <button type="submit" className="btn btn-info">Add House</button>
                </form>
            </div>
        </nav>
    )
};
```

New Room Form:

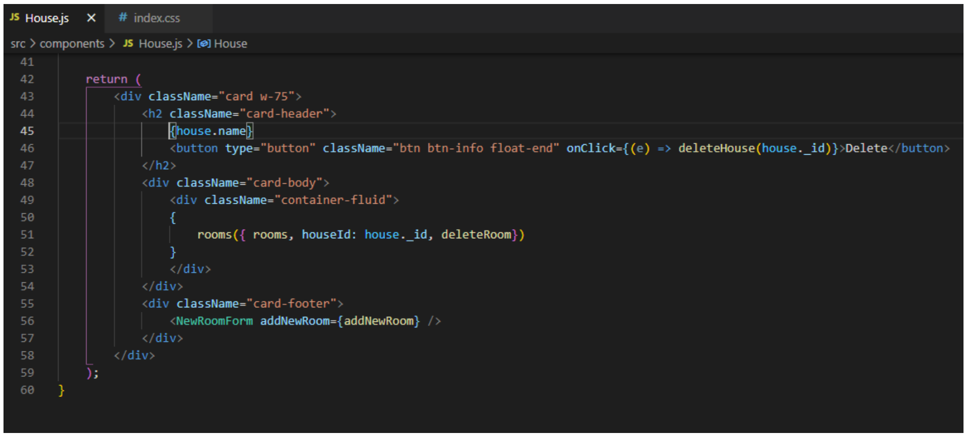


```
import React, {useState} from "react";

export const NewRoomForm = (props) => {
    const [name, setName] = useState('');
    const [area, setArea] = useState('');

    const handleAreaInput = (e) => {
        const int = parseInt(e.target.value, 10);
        setArea( int >= 0 ? int : '');
    }

    const onSubmit = (e) => {
        e.preventDefault();
        if (name && area) {
            props.addNewRoom({name, area});
            setName('');
            setArea('');
        } else {
            console.log('invalid input');
        }
    };

    return (
        <div>
            <form onSubmit={onSubmit}>
            <h5>Add a New Room</h5>
                <input
                    type="text"
                    placeholder="Room Name"
                    onChange={(e) => setName(e.target.value)}
                    value={name}
                    />
                    <input
                    type="text"
                    placeholder="Room Area"
                    onChange={handleAreaInput}
                    value={area}
                    />
                    <button type="button" className="btn btn-info" type="submit">Add Room </button>
            </form>
        </div>
    )

};
```

House:

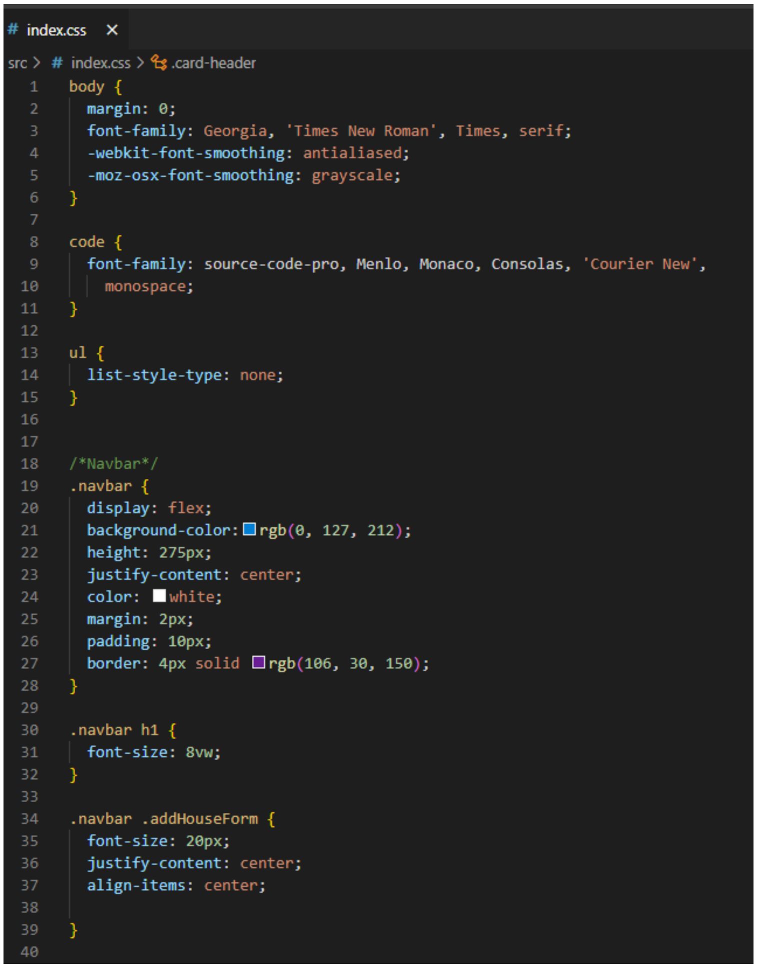


```jsx
import React from "react";
import { housesApi } from "../REST/HousesApi";
import { NewRoomForm } from './NewRoomForm';

export const House = (props) => {
    const {house, updateHouse} = props;

    const deleteHouse = async (houseId) => {
        await housesApi.delete(house);
        updateHouse(house)
    }

    const deleteRoom = (roomId) => {
        const updatedHouse = {
            ...house,
            rooms: house.rooms.filter((x) => x._id !== roomId)
        };
        updateHouse(updatedHouse);
    }

    const addNewRoom = (room) => updateHouse({ ...house, rooms: [...house.rooms, room]});

    const rooms = () => (
        <ul>
            {house.rooms.map((room, index) => (
                <li key={index}>
                    <div className="row">
                        <div className="col-6">
                            <label>{`${room.name}`}</label>
                        </div>
                        <div className="col-6">
                            <label>{`${room.area} sq. ft.`}</label>
                            <button type="button" className="btn btn-info float-end" onClick={(e) => deleteRoom(room._id)}>Delete</button>
                        </div>
                       <hr></hr>
                    </div>
                </li>
            ))}
        </ul>
    );
```

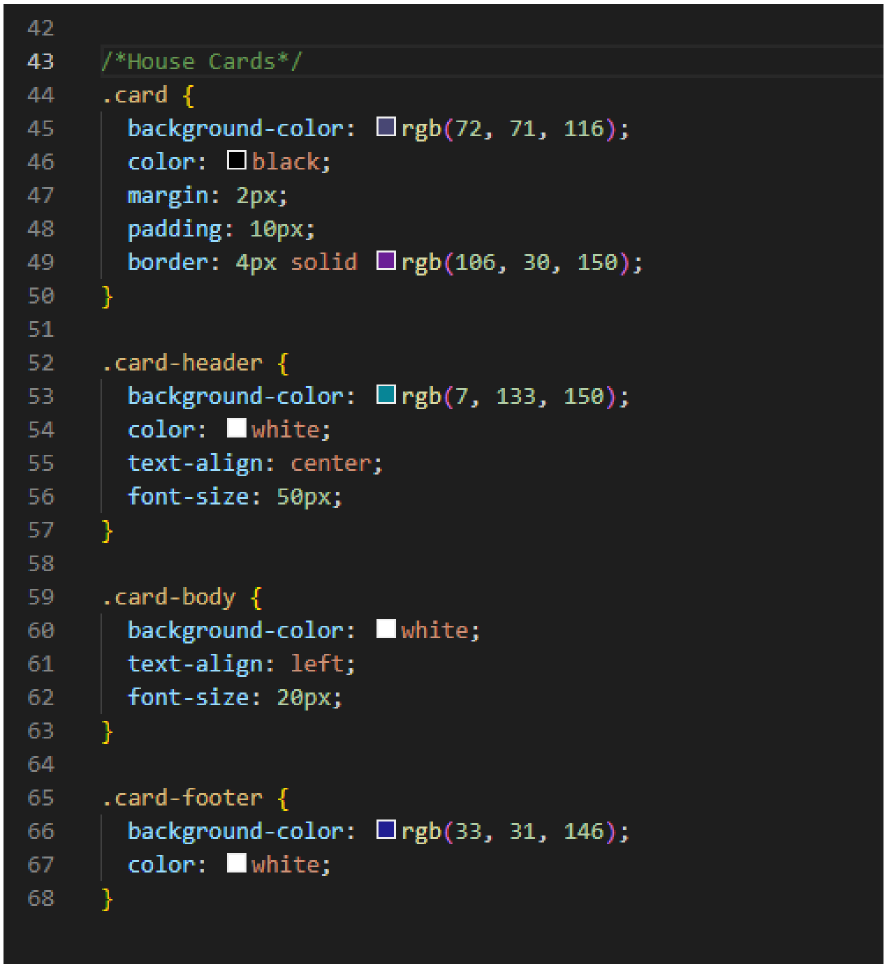


```jsx

    return (
        <div className="card w-75">
            <h2 className="card-header">
                {house.name}
                <button type="button" className="btn btn-info float-end" onClick={(e) => deleteHouse(house._id)}>Delete</button>
            </h2>
            <div className="card-body">
                <div className="container-fluid">
                {
                    rooms({ rooms, houseId: house._id, deleteRoom})
                }
                </div>
            </div>
            <div className="card-footer">
                <NewRoomForm addNewRoom={addNewRoom} />
            </div>
        </div>
    );
}
```

CSS:



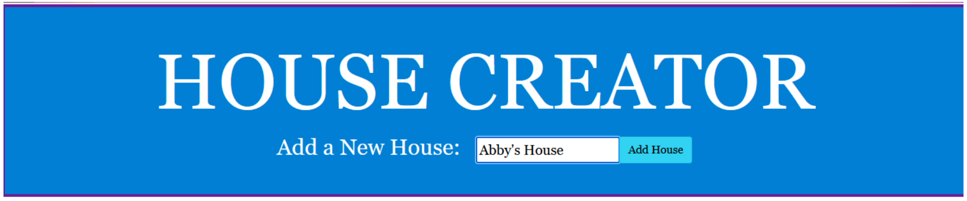
```css
body {
  margin: 0;
  font-family: Georgia, 'Times New Roman', Times, serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}

code {
  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',
    monospace;
}

ul {
  list-style-type: none;
}


/*Navbar*/
.navbar {
  display: flex;
  background-color: rgb(0, 127, 212);
  height: 275px;
  justify-content: center;
  color: white;
  margin: 2px;
  padding: 10px;
  border: 4px solid rgb(106, 30, 150);
}

.navbar h1 {
  font-size: 8vw;
}

.navbar .addHouseForm {
  font-size: 20px;
  justify-content: center;
  align-items: center;

}

```



```css
/*House Cards*/
.card {
  background-color: rgb(72, 71, 116);
  color: black;
  margin: 2px;
  padding: 10px;
  border: 4px solid rgb(106, 30, 150);
}

.card-header {
  background-color: rgb(7, 133, 150);
  color: white;
  text-align: center;
  font-size: 50px;
}

.card-body {
  background-color: white;
  text-align: left;
  font-size: 20px;
}

.card-footer {
  background-color: rgb(33, 31, 146);
  color: white;
}
```

Screenshots of Running Application:

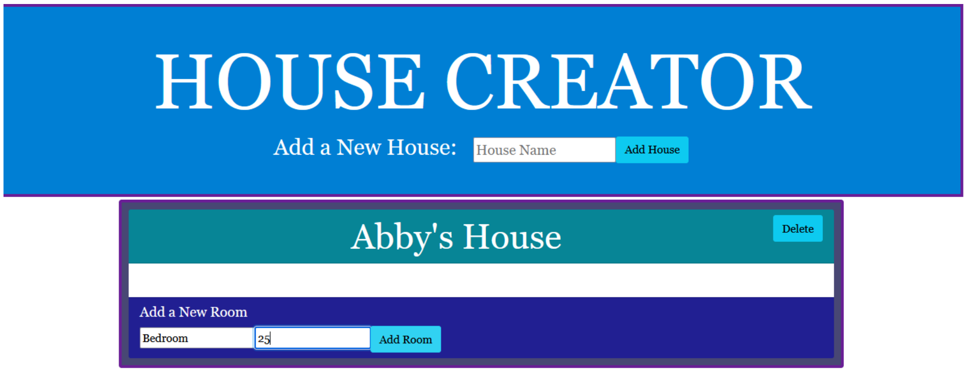
Create a house:



HOUSE CREATOR

Add a New House: Abby's House

Newly created house and create a room:



HOUSE CREATOR

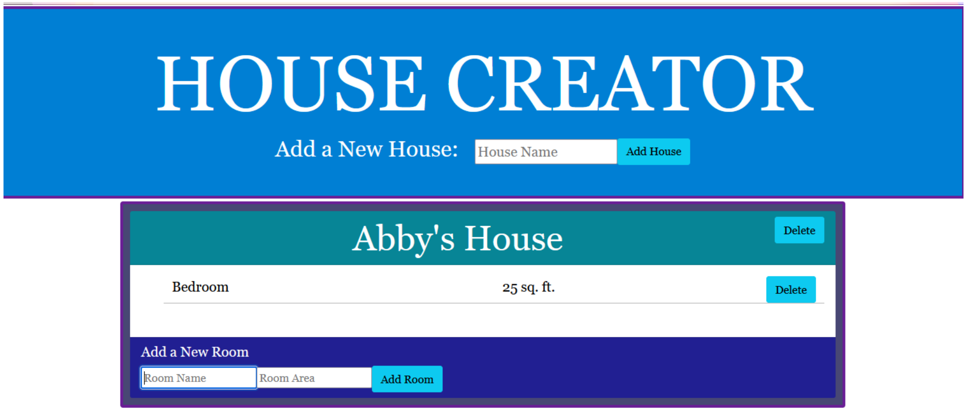
Add a New House: House Name

Abby's House

Add a New Room

Bedroom 25

Created room:



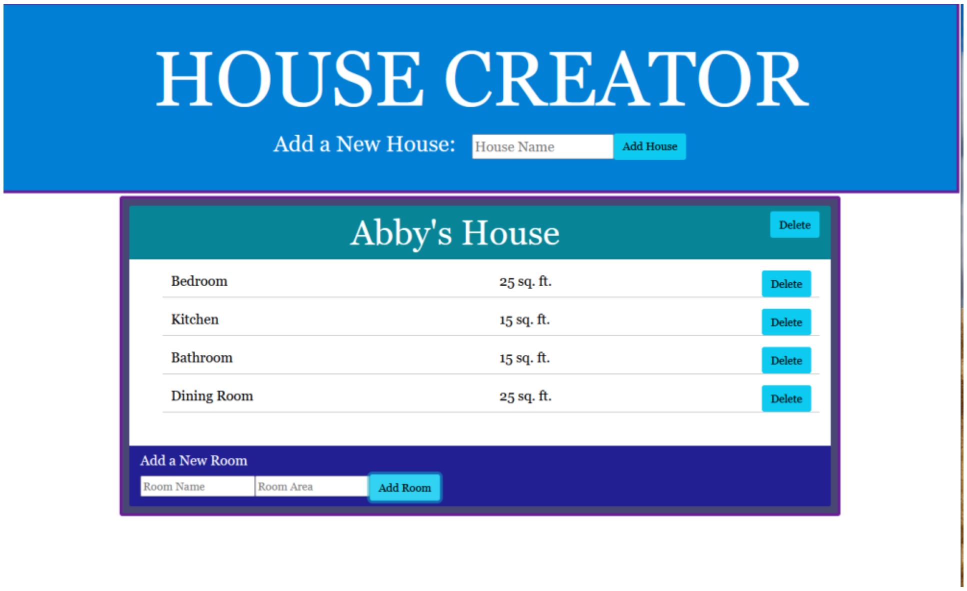
HOUSE CREATOR

Add a New House:

Abby's House

Bedroom	25 sq. ft.	Delete

Read the rooms of a house:



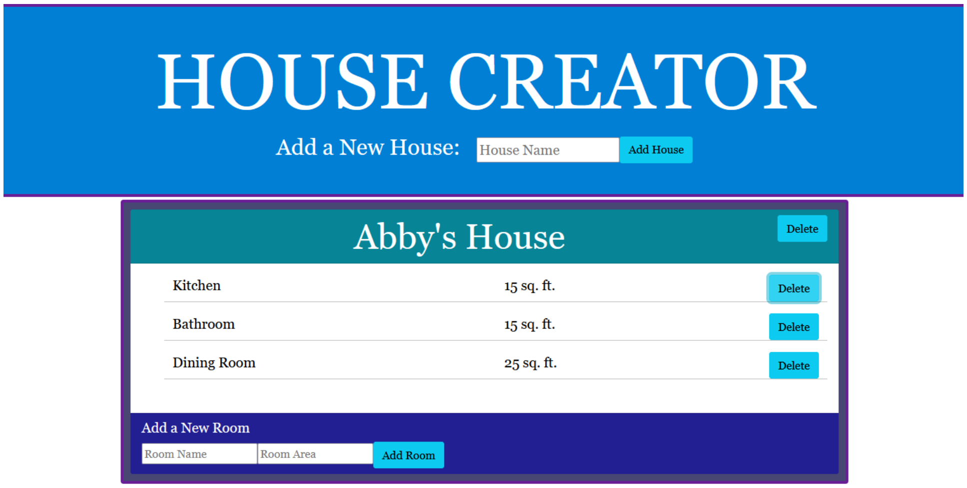
HOUSE CREATOR

Add a New House:

Abby's House

Bedroom	25 sq. ft.	Delete
Kitchen	15 sq. ft.	Delete
Bathroom	15 sq. ft.	Delete
Dining Room	25 sq. ft.	Delete

Delete a room:



HOUSE CREATOR

Add a New House: House Name Add House

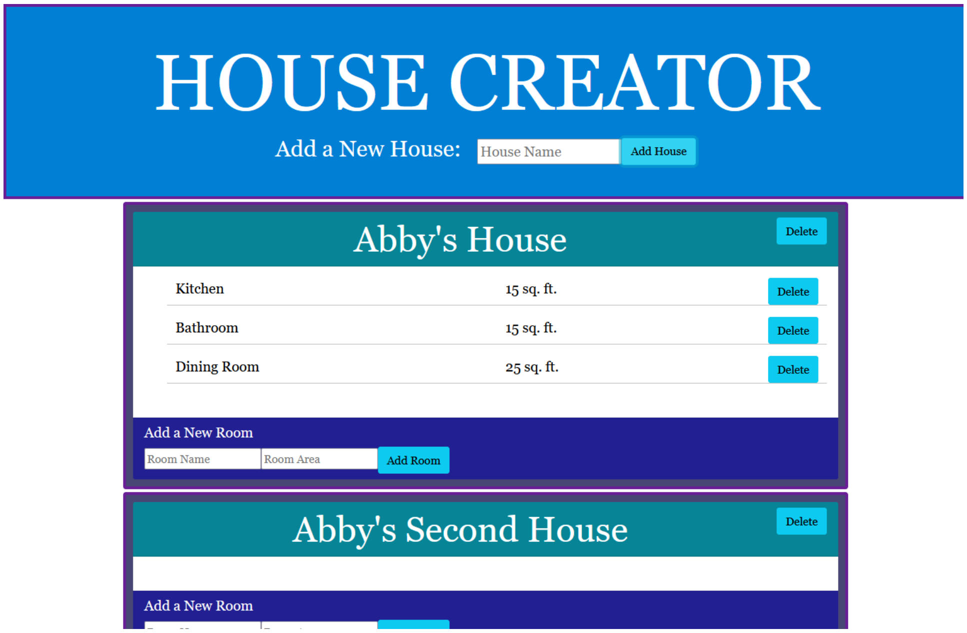
Abby's House Delete

Kitchen	15 sq. ft.	Delete
Bathroom	15 sq. ft.	Delete
Dining Room	25 sq. ft.	Delete

Add a New Room

Room Name Room Area Add Room

Read multiple houses, and delete a house:



HOUSE CREATOR

Add a New House: House Name Add House

Abby's House Delete

Kitchen	15 sq. ft.	Delete
Bathroom	15 sq. ft.	Delete
Dining Room	25 sq. ft.	Delete

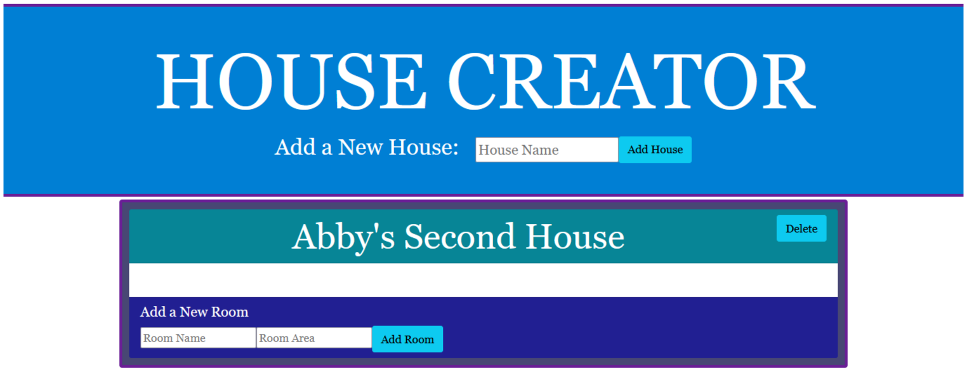
Add a New Room

Room Name Room Area Add Room

Abby's Second House Delete

Add a New Room

PROMINEO TECH



URL to GitHub Repository:

https://github.com/aheiser2/Week-15-Assignment