Web App Design with React Final Project

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 an online API of your choice, or multiple APIs (or if no API can be found, using an array for in-memory storage is okay as well), create a React project of your choice. You will be working on this for the next three weeks.
2. Project must meet the following criteria:
    a. Use React Router and have at least 3 pages
    b. Use React Bootstrap or an alternative styling library
    c. Contain at least 10 components
    d. Allow for all CRUD operations

Screenshots of Code:

Index.js

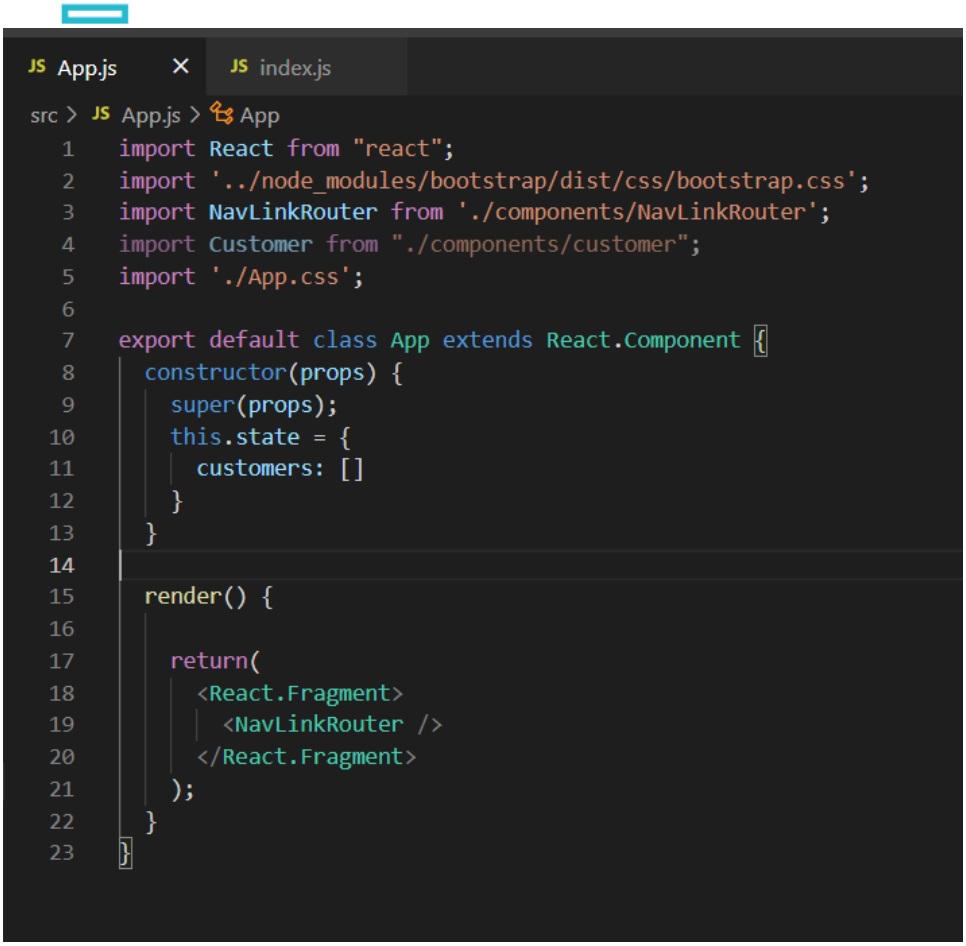
PROMINEO TECH

App.js index.js

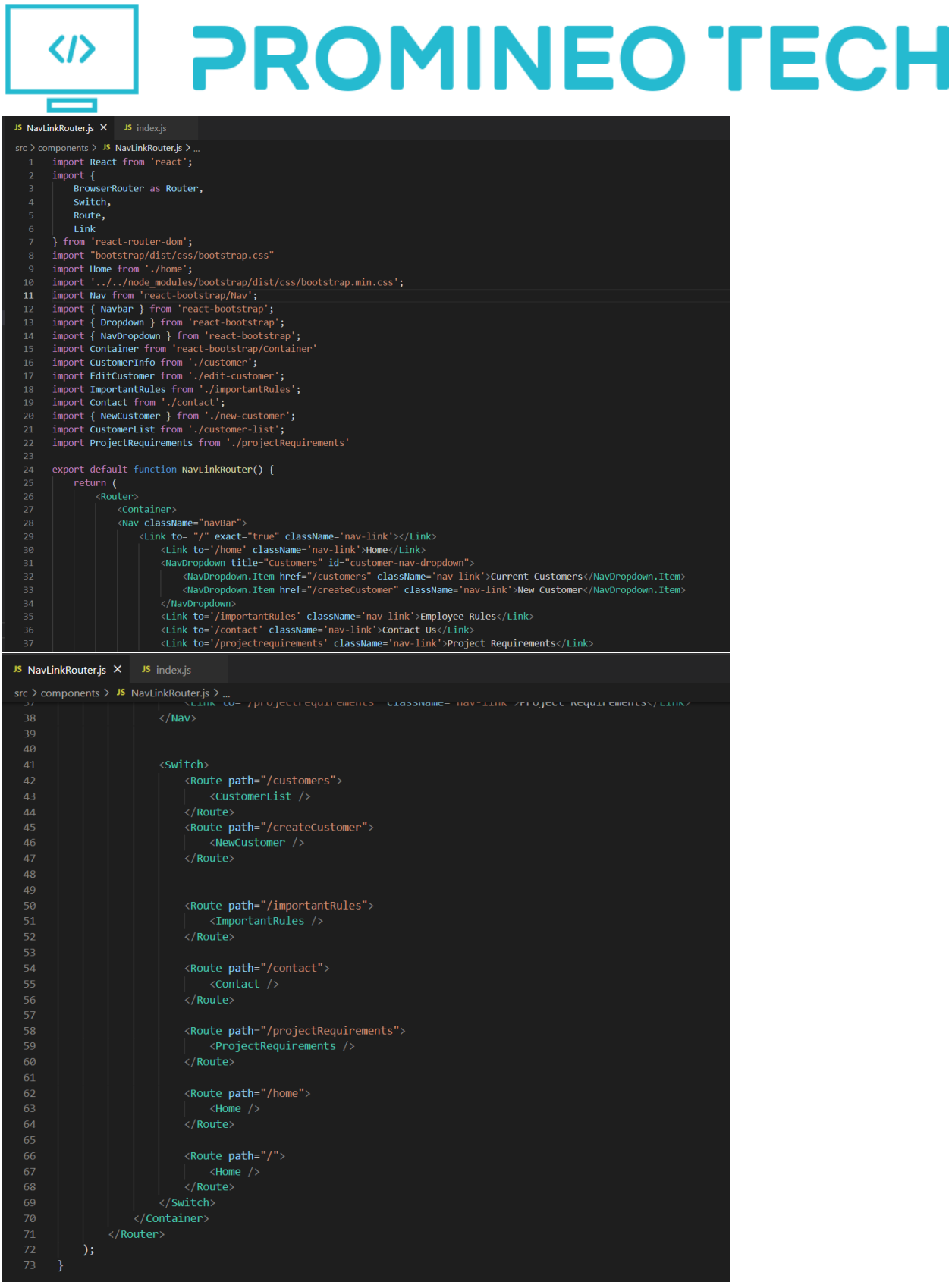
src > index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import { BrowserRouter } from 'react-router-dom';
import App from './App';
import reportWebVitals from './reportWebVitals';
// import { BrowserRouter } from 'react-router-dom';

ReactDOM.render(
  // <BrowserRouter>
   <React.StrictMode>
    <App />
   </React.StrictMode>,
  //  </BrowserRouter>
  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();

```

App.js



NavLinkRouter.js



NavLinkRouter.js | index.js

src > components > NavLinkRouter.js > ...

```
import React from 'react';
import {
    BrowserRouter as Router,
    Switch,
    Route,
    Link
} from 'react-router-dom';
import "bootstrap/dist/css/bootstrap.css"
import Home from './home';
import '../../node_modules/bootstrap/dist/css/bootstrap.min.css';
import Nav from 'react-bootstrap/Nav';
import { Navbar } from 'react-bootstrap';
import { Dropdown } from 'react-bootstrap';
import { NavDropdown } from 'react-bootstrap';
import Container from 'react-bootstrap/Container'
import CustomerInfo from './customer';
import EditCustomer from './edit-customer';
import ImportantRules from './importantRules';
import Contact from './contact';
import { NewCustomer } from './new-customer';
import CustomerList from './customer-list';
import ProjectRequirements from './projectRequirements'

export default function NavLinkRouter() {
    return (
        <Router>
            <Container>
            <Nav className="navBar">
                <Link to= "/" exact="true" className='nav-link'></Link>
                    <Link to='/home' className='nav-link'>Home</Link>
                    <NavDropdown title="Customers" id="customer-nav-dropdown">
                        <NavDropdown.Item href="/customers" className='nav-link'>Current Customers</NavDropdown.Item>
                        <NavDropdown.Item href="/createCustomer" className='nav-link'>New Customer</NavDropdown.Item>
                    </NavDropdown>
                    <Link to='/importantRules' className='nav-link'>Employee Rules</Link>
                    <Link to='/contact' className='nav-link'>Contact Us</Link>
                    <Link to='/projectrequirements' className='nav-link'>Project Requirements</Link>
                </Nav>


                <Switch>
                    <Route path="/customers">
                        <CustomerList />
                    </Route>
                    <Route path="/createCustomer">
                        <NewCustomer />
                    </Route>


                    <Route path="/importantRules">
                        <ImportantRules />
                    </Route>

                    <Route path="/contact">
                        <Contact />
                    </Route>

                    <Route path="/projectRequirements">
                        <ProjectRequirements />
                    </Route>

                    <Route path="/home">
                        <Home />
                    </Route>

                    <Route path="/">
                        <Home />
                    </Route>
                </Switch>
            </Container>
        </Router>
    );
}
```

WalkingAPI.js

PROMINEO TECH

NavLinkRouter.js | walkingApi.js M

src > rest > walkingApi.js > ...

```
const WALKING_ENDPOINT = 'https://crudcrud.com/api/90b59f2aad884721956652ce80c307b8/walking';
//need a crudcrud needs a /word at the end of it so it can create something, can be anything of your choosing.

class WalkingService {
    getAll = async() => {
        try {
            const resp = await fetch(WALKING_ENDPOINT);
            const data = await resp.json();
            return data;
        } catch(e) {
            console.error(`There was a problem getting your walk: ${e}`);
        }
    }
    update = async(customer) => {
        try {
            let updatedWithoutId = {
                customerName: customer.customerName,
                petName: customer.petName,
                petType: customer.petType,
                petBreed: customer.petBreed,
                address: customer.address,
                appointmentDay: customer.appointmentDay,
                appointmentTime: customer.appointmentTime,
                walkDuration: customer.walkDuration,
                customerNotes: customer.customerNotes,
                walkerNotes: customer.walkerNotes,

            }
            const resp = await fetch(`${WALKING_ENDPOINT}/${customer._id}`, {
                method: 'PUT',
                headers: {
                    'Content-Type': 'application/json'
                },
                body: JSON.stringify(updatedWithoutId)
            });
            return resp;
        } catch(e) {
```

NavLinkRouter.js | walkingApi.js M

src > rest > walkingApi.js > WalkingService > create > resp

```
            console.error(`There was a problem updating your walk: ${e}`)
        }
    }

    create = async(customer) => {
        try {
            console.log(WALKING_ENDPOINT);
            const resp = await fetch(`${WALKING_ENDPOINT}`, {
                method: 'POST',
                headers: ({
                    'Content-Type': 'application/json'
                }),
                // changed to call every part of the customer rather individually
                // because calling customer as a whole wasn't working.
                body: JSON.stringify({
                "customerName": customer.customerName,
                "petName" : customer.petName,
                "petType": customer.petType,
                "petBreed": customer.petBreed,
                "address": customer.address,
                "appointmentDay": customer.appointmentDay,
                "appointmentTime": customer.appointmentTime,
                "walkDuration": customer.walkDuration,
                "customerNotes": customer.customerNotes,
                "walkerNotes": customer.walkerNotes})
            });
            return resp;
        } catch(e) {
            console.error(`There was a problem creating your walk: ${e}`);
        }
    }

    delete = async(id) => {
        try {
            const resp = await fetch(`${WALKING_ENDPOINT}/${id}`, {
                method: 'DELETE',
                headers: {
```

NavLinkRouter.js | walkingApi.js M

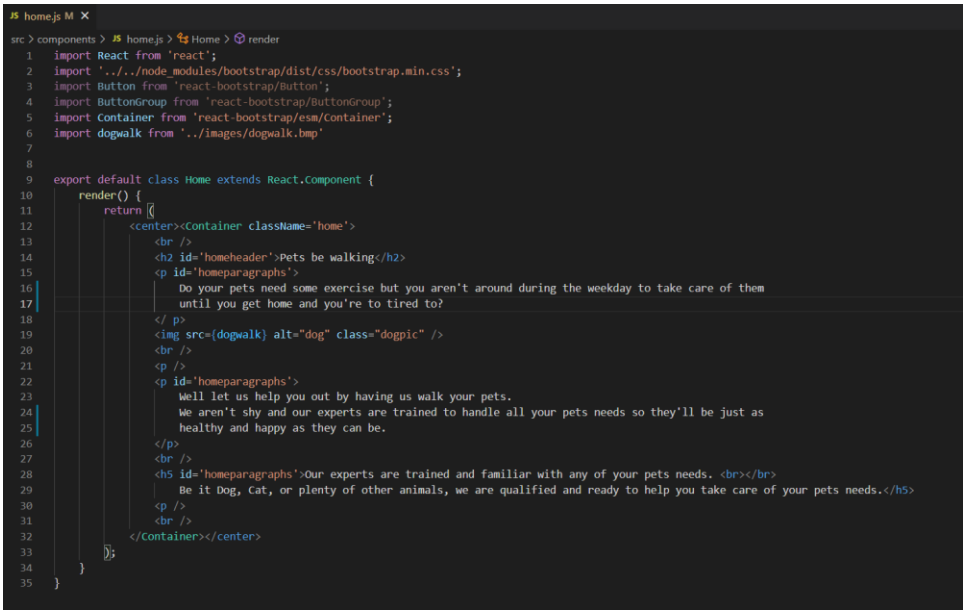
src > rest > walkingApi.js > WalkingService > create > resp

```
        try {
            const resp = await fetch(`${WALKING_ENDPOINT}/${id}`, {
                method: 'DELETE',
                headers: {
                    'Content-Type': 'application/json'
                }
            });
            return resp;
        } catch(e) {
            console.error(`There was a problem deleting your walk: ${e}`);
        }
    }
}

export const walkingService = new WalkingService();
```

Home.js

home.js M

src > components > home.js > Home > render



```
import React from 'react';
import '../../node_modules/bootstrap/dist/css/bootstrap.min.css';
import Button from 'react-bootstrap/Button';
import ButtonGroup from 'react-bootstrap/ButtonGroup';
import Container from 'react-bootstrap/esm/Container';
import dogwalk from '../images/dogwalk.bmp'


export default class Home extends React.Component {
    render() {
        return (
            <center><Container className='home'>
                <br />
                <h2 id='homeheader'>Pets be walking</h2>
                <p id='homeparagraphs'>
                    Do your pets need some exercise but you aren't around during the weekday to take care of them
                    until you get home and you're to tired to?
                </ p>
                <img src={dogwalk} alt="dog" class="dogpic" />
                <br />
                <p />
                <p id='homeparagraphs'>
                    Well let us help you out by having us walk your pets.
                    We aren't shy and our experts are trained to handle all your pets needs so they'll be just as
                    healthy and happy as they can be.
                </p>
                <br />
                <h5 id='homeparagraphs'>Our experts are trained and familiar with any of your pets needs. <br></br>
                    Be it Dog, Cat, or plenty of other animals, we are qualified and ready to help you take care of your pets needs.</h5>
                <p />
                <br />
            </Container></center>
        );
    }
}
```

Contact.js

home.js M contact.js

src > components > contact.js > Contact > render

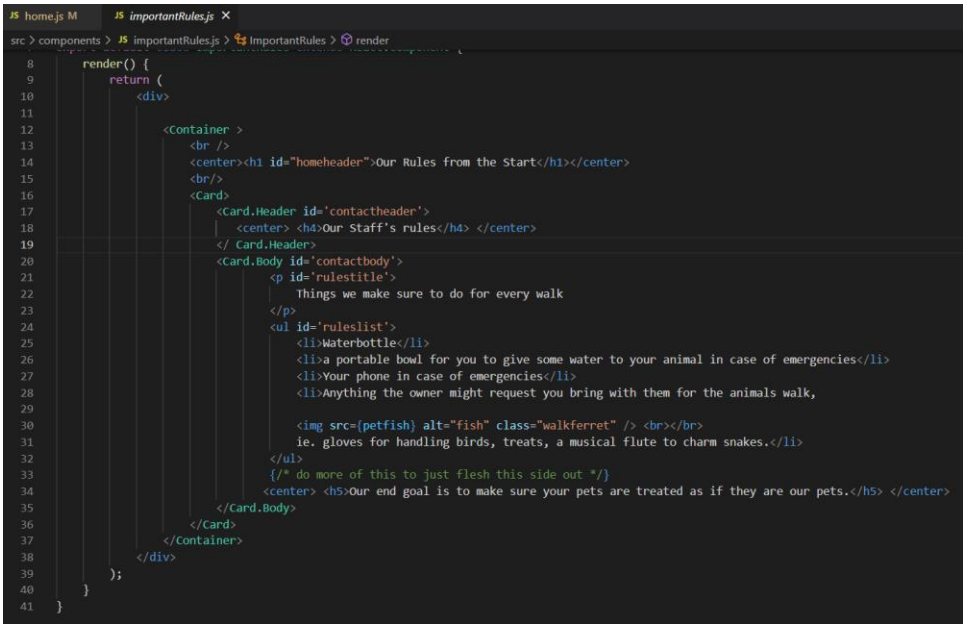
```
import React from 'react';
import '../../node_modules/bootstrap/dist/css/bootstrap.min.css';
import Button from 'react-bootstrap/Button';
import ButtonGroup from 'react-bootstrap/ButtonGroup';
import Card from 'react-bootstrap/Card';
import ListGroup from 'react-bootstrap/ListGroup';
import Container from 'react-bootstrap/Container';
import paw from '../images/paw.bmp'

export default class Contact extends React.Component {
    render() {

        return(
            <div>
                <Container>
                    <Card id="contact">
                        <Card.Header id='contactheader'> <center> <h3>Contact Us</h3> </center> </Card.Header>
                        <Card.Body id='contactbody'>
                            <center><h5>If you need to reach out to us at any time, <br /> here's where you can reach us</h5>
                            <br />
                            <p id='contactphone'>1-(978)-485-985-645-1</p>
                            <p id='contactemail'>petsforwalkies@gmail.gov</p> </center>
                            <center><img src={paw} alt="paw" class="pawprint" /> </center>
                             <br></br>
                        </Card.Body>
                    </Card>
                </Container>
            </div>
        );
    }
}
```

ImportantRules.js

home.js M importantRules.js

src > components > importantRules.js > ImportantRules > render

```
import React from 'react';
import { Card } from 'react-bootstrap';
import Container from 'react-bootstrap/esm/Container';
// import Media from 'react-bootstrap/Media';
import petfish from '../images/petfish.bmp';

export default class ImportantRules extends React.Component {
    render() {
        return (
            <div>

                <Container >
                    <br />
                    <center><h1 id="homeheader">Our Rules from the Start</h1></center>
                    <br/>
                    <Card>
                        <Card.Header id='contactheader'>
                            <center> <h4>Our Staff's rules</h4> </center>
                        </ Card.Header>
                        <Card.Body id='contactbody'>
                                <p id='rulestitle'>
                                    Things we make sure to do for every walk
                                </p>
                                <ul id='ruleslist'>
                                    <li>Waterbottle</li>
                                    <li>a portable bowl for you to give some water to your animal in case of emergencies</li>
                                    <li>Your phone in case of emergencies</li>
                                    <li>Anything the owner might request you bring with them for the animals walk,

                                    <img src={petfish} alt="fish" class="walkferret" /> <br></br>
                                    ie. gloves for handling birds, treats, a musical flute to charm snakes.</li>
                                </ul>
                                {/* do more of this to just flesh this side out */}
                               <center> <h5>Our end goal is to make sure your pets are treated as if they are our pets.</h5> </center>
                        </Card.Body>
                    </Card>
                </Container>
```



ProjectRequirements.js

PROMINEO TECH

projectRequirements.js M

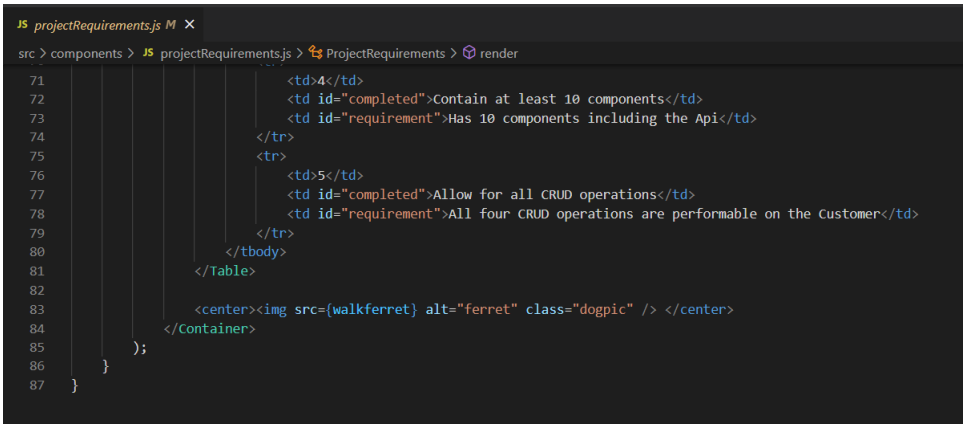
src > components > projectRequirements.js > ProjectRequirements > render

```
import React from 'react';
import Container from 'react-bootstrap/esm/Container';
import Table from 'react-bootstrap/Table';
import '../../node_modules/bootstrap/dist/css/bootstrap.css';
import walkferret from '../images/walkferret.bmp'

export default class ProjectRequirements extends React.Component {
    render() {
        return (
            <Container><br />
                <center><h3 id="titleprojectpage">Project's full requirements page</h3></center>
                <Table id='table'>
                    <thead>
                        <tr>
                            <th><h5>Color</h5></th>
                            <th><h5>What it means</h5></th>
                        </tr>
                    </thead>
                    <tbody>
                        <tr>
                            <td id="requirement"></td>
                            <td>Requirements</td>
                        </tr>
                        <tr>
                            <td id="completed"></td>
                            <td>Completed</td>
                        </tr>
                        <tr>
                            <td id="analysising"></td>
                            <td>Analysising</td>
                        </tr>
                        <tr>
                            <td id="debugging"></td>
                            <td>Debugging</td>
                        </tr>
                        <tr>
                            <td id="planning"></td>
```

projectRequirements.js M

src > components > projectRequirements.js > ProjectRequirements > render

```
                            <td>Planning Phase</td>
                        </tr>
                    </tbody>
                </Table>
                {/* <br /> */}

                {/* <center><img src={walkferret} alt="ferret" class="dogpic" /> </center> <br /> */}

                <Table>
                    <thead>
                        <tr>
                            <th><h5>#</h5></th>
                            <th><h5>Project's Requirement</h5></th>
                            <th><h5>Data</h5></th>
                        </tr>
                    </thead>
                    <tbody>
                        <tr>
                            <td>1</td>
                            <td id="completed">Use an Api or multiple Apis to create a react app</td>
                            <td id="requirement">https://crudcrud.com/api/90b59f2aad884721956652ce80c307b8/walking</td>
                        </tr>
                        <tr>
                            <td>2</td>
                            <td id="completed">Use React Router and have at least 3 pages</td>
                            <td id="requirement">Contains 6 pages and a dropdown within the Navbar</td>
                        </tr>
                        <tr>
                            <td>3</td>
                            <td id="completed">Use React Bootstrap or an alternative styling library</td>
                            <td id="requirement">Uses React-Bootstrap</td>
                        </tr>
                        <tr>
                            <td>4</td>
                            <td id="completed">Contain at least 10 components</td>
                            <td id="requirement">Has 10 components including the Api</td>
                        </tr>
```



New-customer.js

PROMINEO TECH

JS new-customer.js M

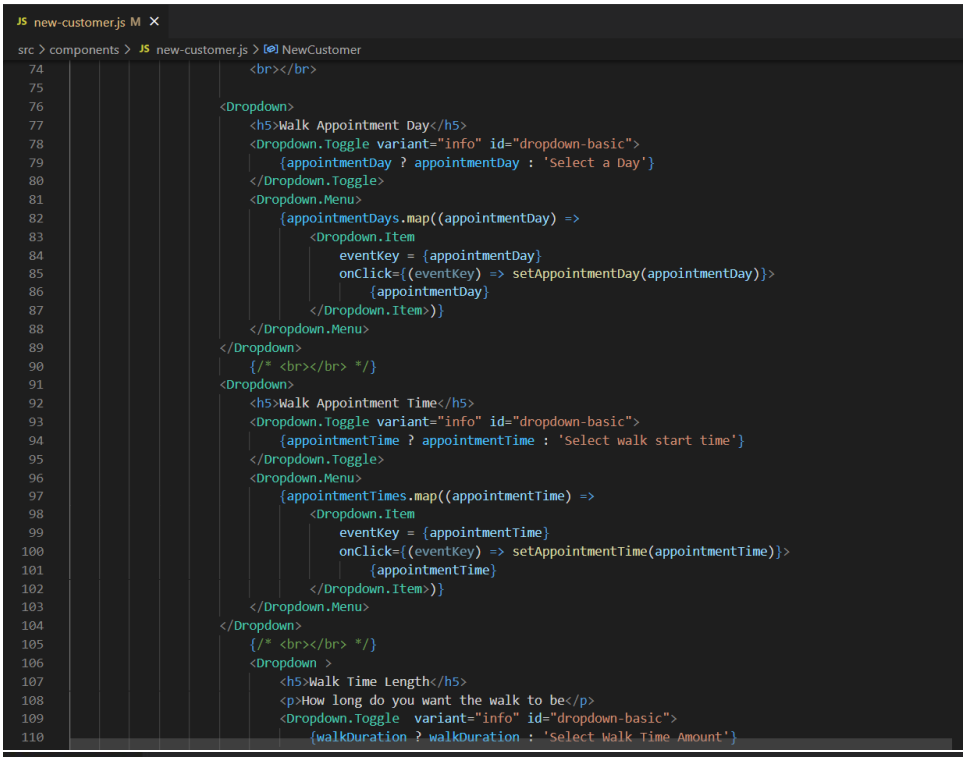
src > components > JS new-customer.js > NewCustomer

```
import React, { useState } from 'react';
import '../../node_modules/bootstrap/dist/css/bootstrap.css';
import Dropdown from 'react-bootstrap/Dropdown';
import Container from 'react-bootstrap/Container';
import { walkingService } from '../rest/walkingApi';

export const NewCustomer = (props) => {
    const [customerName, setCustomerName] = useState('');
    const [petName, setPetName] = useState('');
    const [petBreed, setPetBreed] = useState('');

    const [address, setAddress] = useState('');
    const [appointmentDay, setAppointmentDay] = useState('');
    const [appointmentTime, setAppointmentTime] = useState('');
    const [walkDuration, setWalkDuration] = useState('');
    const [customerNotes, setCustomerNotes] = useState('');
    const [walkerNotes, setWalkerNotes] = useState('');

    // const petTypes = ['Dog', 'Cat', 'Sugar Glider', 'Bird', 'Snake', 'Ferret', 'Other'];
    const appointmentDays = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday'];
    const appointmentTimes = ['10 am', '10:30 am', '11 am', '11:30 am', '12 pm', '12:30 pm', '1 pm', '1:30 pm', '2 pm', '2:30 pm'];
    const walkDurations = ['20 minutes','30 minutes', '45 minutes', '60 minutes', '75 minutes', '90 minutes'];

    const handleSubmit = (event) => {
        event.preventDefault();
        addNewCustomer({customerName, petName, petBreed, address, appointmentDay, appointmentTime, walkDuration, customerNotes, walkerNotes})
        console.log('The customers props are ' + customerName + ' ' + petName + ' ' + petBreed + ' ' + address + ' ' + appointmentDay + ' ' +
                appointmentTime + ' ' + walkDuration + ' ' + customerNotes + ' ' + walkerNotes + '.');
    }

    const _refreshData = async() => {
        const customers = await walkingService.getAll();
    }

    function componentDidMount() {
        _refreshData();
    }
```

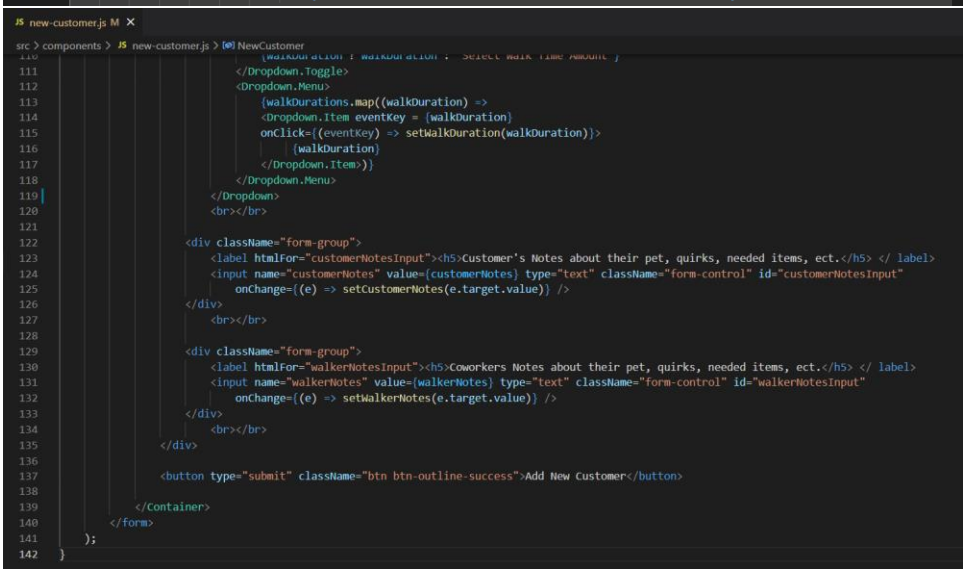
JS new-customer.js M

src > components > JS new-customer.js > NewCustomer

```
    }

    const addNewCustomer = async(customer) => {
        await walkingService.create(customer);
        _refreshData();
    }

    return (
        <form onSubmit={handleSubmit}>
            <Container id="customer-edit">
                <div className='form-group'>
                    <center><h2>New Customer form</h2></center>
                    <label htmlFor="customerNameInput"><h5>New Customer's Name</h5></label>
                    <input name="customerName" value={customerName} type="text" className="form-control" id="customerNameInput"
                    onChange={(e) => setCustomerName(e.target.value)} />
                    <div className="form-group">
                        <br></br>

                        <div className="form-group">
                            <label htmlFor="petNameInput"><h5>Pet's Name</h5></label>
                            <input name="petName" value={petName} type="text" className="form-control" id="petNameInput"
                            onChange={(e) => setPetName(e.target.value)} /><br></br>
                        </div>
                    </div>
                    <div />

                    <div className="form-group">
                        <label htmlFor="petBreedInput"><h5>Pet's Breed</h5></label>
                        <input name="petBreed" type="text" value={petBreed} className="form-control" id="petBreedInput"
                            onChange={(e) => setPetBreed(e.target.value)} /><br></br>
                    </div>

                    <div className="form-group">
                        <label htmlFor="addressInput"><h5>Customer's Address</h5> </ label>
                        <input name="address" value={address} type="text" className="form-control" id="addressInput"
                            onChange={(e) => setAddress(e.target.value)} />
                    </div>
                        <br></br>
```

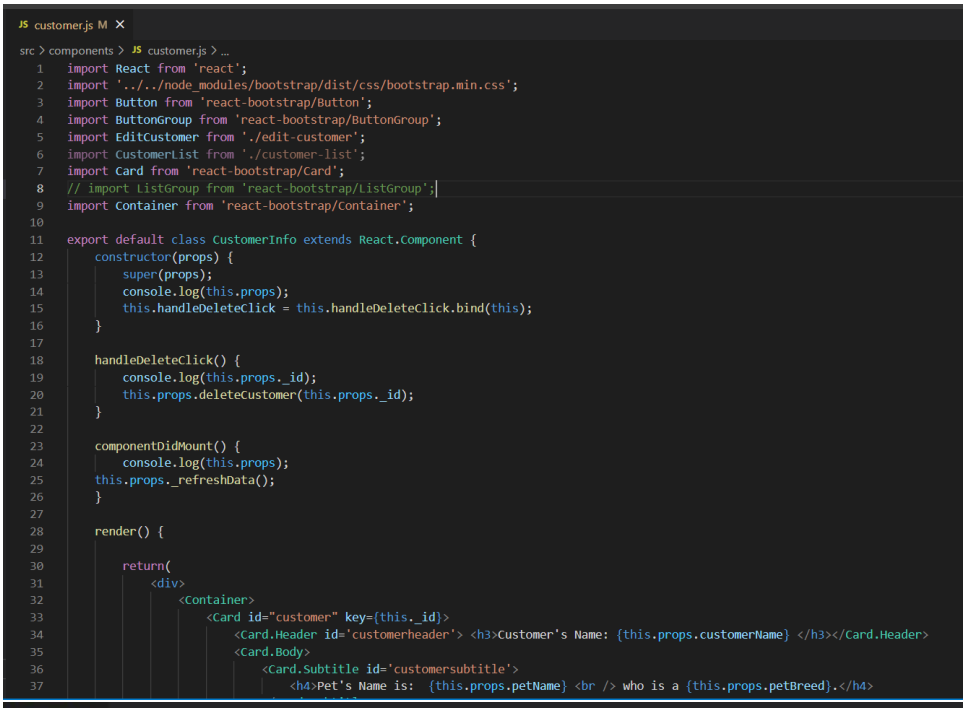


```
new-customer.js M
src > components > new-customer.js > NewCustomer
                        <br></br>

                    <Dropdown>
                        <h5>Walk Appointment Day</h5>
                        <Dropdown.Toggle variant="info" id="dropdown-basic">
                            {appointmentDay ? appointmentDay : 'Select a Day'}
                        </Dropdown.Toggle>
                        <Dropdown.Menu>
                            {appointmentDays.map((appointmentDay) =>
                                <Dropdown.Item
                                    eventKey = {appointmentDay}
                                    onClick={(eventKey) => setAppointmentDay(appointmentDay)}>
                                        {appointmentDay}
                                </Dropdown.Item>)}
                        </Dropdown.Menu>
                    </Dropdown>
                        {/* <br></br> */}
                    <Dropdown>
                        <h5>Walk Appointment Time</h5>
                        <Dropdown.Toggle variant="info" id="dropdown-basic">
                            {appointmentTime ? appointmentTime : 'Select walk start time'}
                        </Dropdown.Toggle>
                        <Dropdown.Menu>
                            {appointmentTimes.map((appointmentTime) =>
                                <Dropdown.Item
                                    eventKey = {appointmentTime}
                                    onClick={(eventKey) => setAppointmentTime(appointmentTime)}>
                                        {appointmentTime}
                                </Dropdown.Item>)}
                        </Dropdown.Menu>
                    </Dropdown>
                        {/* <br></br> */}
                        <Dropdown >
                            <h5>Walk Time Length</h5>
                            <p>How long do you want the walk to be</p>
                            <Dropdown.Toggle  variant="info" id="dropdown-basic">
                                {walkDuration ? walkDuration : 'Select Walk Time Amount'}
```

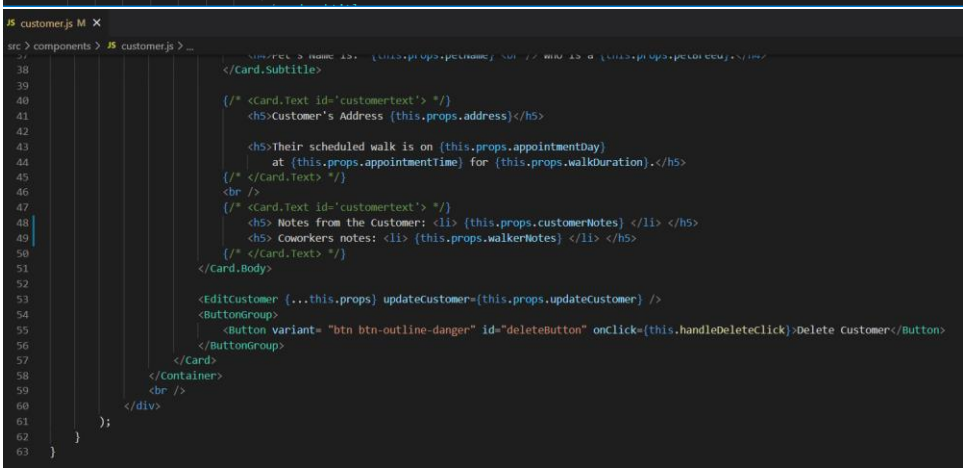


```
new-customer.js M
src > components > new-customer.js > NewCustomer
                            </Dropdown.Toggle>
                            <Dropdown.Menu>
                                {walkDurations.map((walkDuration) =>
                                <Dropdown.Item eventKey = {walkDuration}
                                onClick={(eventKey) => setWalkDuration(walkDuration)}>
                                    {walkDuration}
                                </Dropdown.Item>)}
                            </Dropdown.Menu>
                        </Dropdown>
                        <br></br>

                    <div className="form-group">
                        <label htmlFor="customerNotesInput"><h5>Customer's Notes about their pet, quirks, needed items, ect.</h5> </ label>
                        <input name="customerNotes" value={customerNotes} type="text" className="form-control" id="customerNotesInput"
                            onChange={(e) => setCustomerNotes(e.target.value)} />
                    </div>
                        <br></br>

                    <div className="form-group">
                        <label htmlFor="walkerNotesInput"><h5>Coworkers Notes about their pet, quirks, needed items, ect.</h5> </ label>
                        <input name="walkerNotes" value={walkerNotes} type="text" className="form-control" id="walkerNotesInput"
                            onChange={(e) => setWalkerNotes(e.target.value)} />
                    </div>
                        <br></br>
                </div>

                <button type="submit" className="btn btn-outline-success">Add New Customer</button>

            </Container>
        </form>
    );
}
```

Customer.js

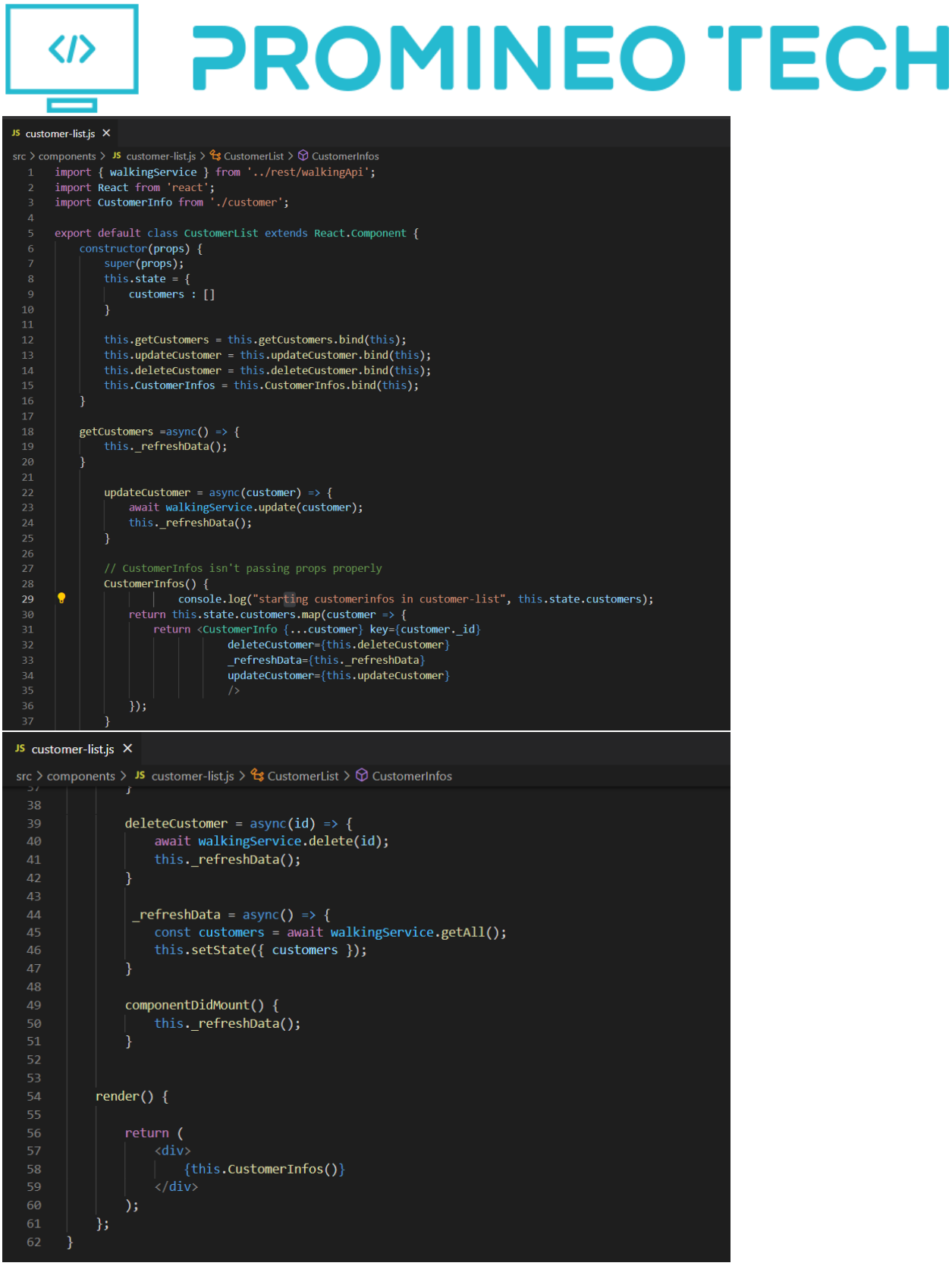


```
import React from 'react';
import '../../node_modules/bootstrap/dist/css/bootstrap.min.css';
import Button from 'react-bootstrap/Button';
import ButtonGroup from 'react-bootstrap/ButtonGroup';
import EditCustomer from './edit-customer';
import CustomerList from './customer-list';
import Card from 'react-bootstrap/Card';
// import ListGroup from 'react-bootstrap/ListGroup';
import Container from 'react-bootstrap/Container';

export default class CustomerInfo extends React.Component {
    constructor(props) {
        super(props);
        console.log(this.props);
        this.handleDeleteClick = this.handleDeleteClick.bind(this);
    }

    handleDeleteClick() {
        console.log(this.props._id);
        this.props.deleteCustomer(this.props._id);
    }

    componentDidMount() {
        console.log(this.props);
    this.props._refreshData();
    }

    render() {

        return(
            <div>
                <Container>
                    <Card id="customer" key={this._id}>
                        <Card.Header id='customerheader'> <h3>Customer's Name: {this.props.customerName} </h3></Card.Header>
                        <Card.Body>
                            <Card.Subtitle id='customersubtitle'>
                                <h4>Pet's Name is:  {this.props.petName} <br /> who is a {this.props.petBreed}.</h4>
```



```
                            </Card.Subtitle>

                            {/* <Card.Text id='customertext'> */}
                                <h5>Customer's Address {this.props.address}</h5>

                                <h5>Their scheduled walk is on {this.props.appointmentDay}
                                    at {this.props.appointmentTime} for {this.props.walkDuration}.</h5>
                            {/* </Card.Text> */}
                            <br />
                            {/* <Card.Text id='customertext'> */}
                                <h5> Notes from the Customer: <li> {this.props.customerNotes} </li> </h5>
                                <h5> Coworkers notes: <li> {this.props.walkerNotes} </li> </h5>
                            {/* </Card.Text> */}
                        </Card.Body>

                        <EditCustomer {...this.props} updateCustomer={this.props.updateCustomer} />
                        <ButtonGroup>
                            <Button variant= "btn btn-outline-danger" id="deleteButton" onClick={this.handleDeleteClick}>Delete Customer</Button>
                        </ButtonGroup>
                    </Card>
                </Container>
                <br />
            </div>
        );
    }
}
```

Customer-list.js



customer-list.js

src > components > customer-list.js > CustomerList > CustomerInfos

```
import { walkingService } from '../rest/walkingApi';
import React from 'react';
import CustomerInfo from './customer';

export default class CustomerList extends React.Component {
    constructor(props) {
        super(props);
        this.state = {
            customers : []
        }

        this.getCustomers = this.getCustomers.bind(this);
        this.updateCustomer = this.updateCustomer.bind(this);
        this.deleteCustomer = this.deleteCustomer.bind(this);
        this.CustomerInfos = this.CustomerInfos.bind(this);
    }

    getCustomers =async() => {
        this._refreshData();
    }

        updateCustomer = async(customer) => {
            await walkingService.update(customer);
            this._refreshData();
        }

        // CustomerInfos isn't passing props properly
        CustomerInfos() {
                    console.log("starting customerinfos in customer-list", this.state.customers);
            return this.state.customers.map(customer => {
                return <CustomerInfo {...customer} key={customer._id}
                            deleteCustomer={this.deleteCustomer}
                            _refreshData={this._refreshData}
                            updateCustomer={this.updateCustomer}
                            />
            });
        }
```

customer-list.js

src > components > customer-list.js > CustomerList > CustomerInfos

```
        deleteCustomer = async(id) => {
            await walkingService.delete(id);
            this._refreshData();
        }

         _refreshData = async() => {
            const customers = await walkingService.getAll();
            this.setState({ customers });
        }

        componentDidMount() {
            this._refreshData();
        }


    render() {

        return (
            <div>
                {this.CustomerInfos()}
            </div>
        );
    };
}
```

Edit-customer.js

PROMINEO TECH

edit-customer.js M

src > components > edit-customer.js > EditCustomer > constructor

```
import React from "react";
import '../../node_modules/bootstrap/dist/css/bootstrap.min.css';
import  Dropdown from "react-bootstrap/Dropdown";
import  Container from "react-bootstrap/Container";
import walkferret from '../images/walkferret.bmp'
import DropdownItem from "react-bootstrap/esm/DropdownItem";

export default class EditCustomer extends React.Component {
    constructor(props) {
        super(props);
        this.state = {
            customerName: props.customerName,
            petName: props.petName,
            petType: props.petType,
            petBreed: props.petBreed,

            address: props.address,
            appointmentDay: props.appointmentDay,
            appointmentTime: props.appointmentTime,
            walkDuration: props.walkDuration,
            customerNotes: props.customerNotes,
            walkerNotes: props.walkerNotes,

            _id: props._id
        }

        this.petTypes = ['Dog', 'Cat', 'Sugar Glider', 'Bird', 'Snake', 'Ferret', 'Other'];
        this.appointmentDays = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday'];
        this.appointmentTimes = ['10 am', '10:30 am', '11 am', '11:30 am', '12 pm', '12:30 pm', '1 pm', '1:30 pm', '2 pm', '2:30 pm'];
        this.walkDurations = ['20 minutes','30 minutes', '45 minutes', '60 minutes', '75 minutes', '90 minutes'];

        this.handleChange = this.handleChange.bind(this);
        this.handleSubmit = this.handleSubmit.bind(this);
        this.setPetType = this.setPetType.bind(this);
        this.setAppointmentDay = this.setAppointmentDay.bind(this);
        this.setAppointmentTime = this.setAppointmentTime.bind(this);
        this.setWalkDuration = this.setWalkDuration.bind(this);
```

edit-customer.js M

src > components > edit-customer.js > EditCustomer > constructor

```
    }

    setPetType(petType) {
        this.setState({
            petType: petType
        });
        console.log("setting petType in edit customer")
    }

    setAppointmentDay(appointmentDay) {
        this.setState({
            appointmentDay: appointmentDay
        });
        console.log("setting appointmentDate in edit customer");
    }

    setAppointmentTime(appointmentTime) {
        this.setState({
            appointmentTime: appointmentTime
        });
        console.log("setting appointmentTIme in edit customer");
    }

    setWalkDuration(walkDuration) {
        this.setState({
            walkDuration: walkDuration
        });
        console.log("setting walkDuration in edit customer");
    }

    handleChange(event) {
        const target = event.target;
        const value = target.value;
        const field = target.id.replace('input_', '');

        this.setState({
            [field]: value
```

PROMINEO TECH

edit-customer.js M

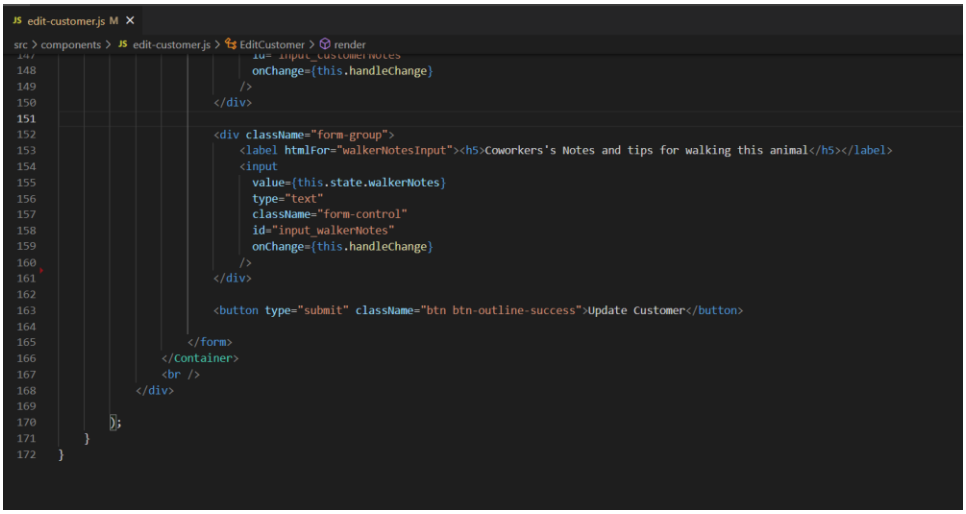
src > components > edit-customer.js > EditCustomer > constructor

```
            [field]: value
        });
        console.log("set state on " + field + " in edit customer");
    }

    handleSubmit(event) {
        this.props.updateCustomer(this.state);
        event.preventDefault();
    }

    render() {

        return (
            <div>

                {console.log(this.state._id)}
                <Container id="customer-edit" >
                    <form onSubmit={this.handleSubmit} ref={ref => {this.myForm=ref;}} >
                    {/* <p>*{this.state.appointmentDay}</p>
                    <p>*{this.state.appointmentTime}</p>
                    <p>*{this.state.walkDuration}</p> */}
                        <Dropdown>
                            <h5>Walk Day Appointment</h5>
                            <p>Timeslot for the walk on that specific day</p>
                            <Dropdown.Toggle variant="outline-secondary" id="dropdown-basic">
                                {this.state.appointmentDay ? this.state.appointmentDay : 'Select a Day'}
                            </Dropdown.Toggle>
                            <Dropdown.Menu>
                                {this.appointmentDays.map((appointmentDay) =>
                                <Dropdown.Item
                                    eventKey={appointmentDay}
                                    onClick={(eventKey) => this.setAppointmentDay(appointmentDay)}>
                                    {appointmentDay}
                                </Dropdown.Item> )}
                            </Dropdown.Menu>
                        </Dropdown>
```

edit-customer.js M

src > components > edit-customer.js > EditCustomer > constructor

```
                        <Dropdown >
                            <h5>Walk Appointment Time</h5>
                            <p>When do you want to walk to start at?</p>
                            <Dropdown.Toggle  variant="outline-secondary" id="dropdown-basic">
                                {this.state.appointmentTime ? this.state.appointmentTime : 'Select walk start time'}
                            </Dropdown.Toggle>
                            <Dropdown.Menu>
                                {this.appointmentTimes.map((appointmentTime) =>
                                <Dropdown.Item eventKey = {appointmentTime}
                                onClick={(eventKey) => this.setAppointmentTime(appointmentTime)}>
                                    {appointmentTime}
                                </Dropdown.Item>)}
                            </Dropdown.Menu>
                        </Dropdown>

                        <Dropdown >
                            <h5>Walk Time Length</h5>
                            <p>How long do you want the walk to be</p>
                            <Dropdown.Toggle  variant="outline-secondary" id="dropdown-basic">
                                {this.state.walkDuration ? this.state.walkDuration : 'Select Walk Time Amount'}
                            </Dropdown.Toggle>
                            <Dropdown.Menu>
                                {this.walkDurations.map((walkDuration) =>
                                <Dropdown.Item eventKey = {walkDuration}
                                onClick={(eventKey) => this.setWalkDuration(walkDuration)}>
                                    {walkDuration}
                                </Dropdown.Item>)}
                            </Dropdown.Menu>
                        </Dropdown>

                        <div className="form-group">
                            <label htmlFor="customerNotesInput"><h5>Customer's Notes or any special requests for their pet</h5></label>
                            <input
                              value={this.state.customerNotes}
                              type="text"
                              className="form-control"
                              id="input_customerNotes"
```


edit-customer.js M
src > components > edit-customer.js > EditCustomer > render
onChange={this.handleChange}
/>

<label htmlFor="walkerNotesInput"><h5>Coworkers's Notes and tips for walking this animal</h5></label>
<input
value={this.state.walkerNotes}
type="text"
className="form-control"
id="input_walkerNotes"
onChange={this.handleChange}
/>

<button type="submit" className="btn btn-outline-success">Update Customer</button>
</form>
</Container>

);
}
}

App.css

edit-customer.js M
App.css
src > App.css > .App
.App {
text-align: center;
}
.App-logo {
height: 40vmin;
pointer-events: none;
}
@media (prefers-reduced-motion: no-preference) {
.App-logo {
animation: App-logo-spin infinite 20s linear;
}
}
.App-header {
background-color: #282c34;
min-height: 100vh;
display: flex;
flex-direction: column;
align-items: center;
justify-content: center;
font-size: calc(10px + 2vmin);
color: white;
}
.App-link {
color: #61dafb;
}
@keyframes App-logo-spin {
from {
transform: rotate(0deg);
}
to {
transform: rotate(360deg);
}

edit-customer.js M
App.css
src > App.css > .App
3 }
4
5 .App-logo {
6 height: 40vmin;
7 pointer-events: none;
8 }
9
10 @media (prefers-reduced-motion: no-preference) {
11 .App-logo {
12 animation: App-logo-spin infinite 20s linear;
13 }
14 }
15
16 .App-header {
17 background-color: #282c34;
18 min-height: 100vh;
19 display: flex;
20 flex-direction: column;
21 align-items: center;
22 justify-content: center;
23 font-size: calc(10px + 2vmin);
24 color: white;
25 }
26
27 .App-link {
28 color: #61dafb;
29 }
30
31 @keyframes App-logo-spin {
32 from {
33 transform: rotate(0deg);
34 }
35 to {
36 transform: rotate(360deg);
37 }
38 }

Index.css

PROMINEO TECH

edit-customer.js M | index.css M

src > # index.css > ...

```
body {
  background-image: url(http://1-background.com/images/ultraviolet/ultraviolet-dark1-repeating-background-tile.jpg) !important;
  margin: 0;
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',
    'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
    sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}

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

.navBar {
  background-color: rgba(143, 45, 146, 0.6);
  padding: 10px;
  border: inset 5px rgba(137, 18, 141, 0.733);
}

.nav-link {
  color: aliceblue !important;
  text-shadow: 0 0 3px #000000, 0 0 5px #000000;
}

#home {
  /* text-align: center; */
  /* padding-left: 320px; */
  /* align-items: center; */
  background-color: rgba(180, 100, 255, 0.151);
  color: rgb(125, 192, 252);
  border-color: blue;
  max-width: 100%;
  max-height: 100%;
  margin: auto;
}
```

edit-customer.js M | index.css M

src > # index.css > ...

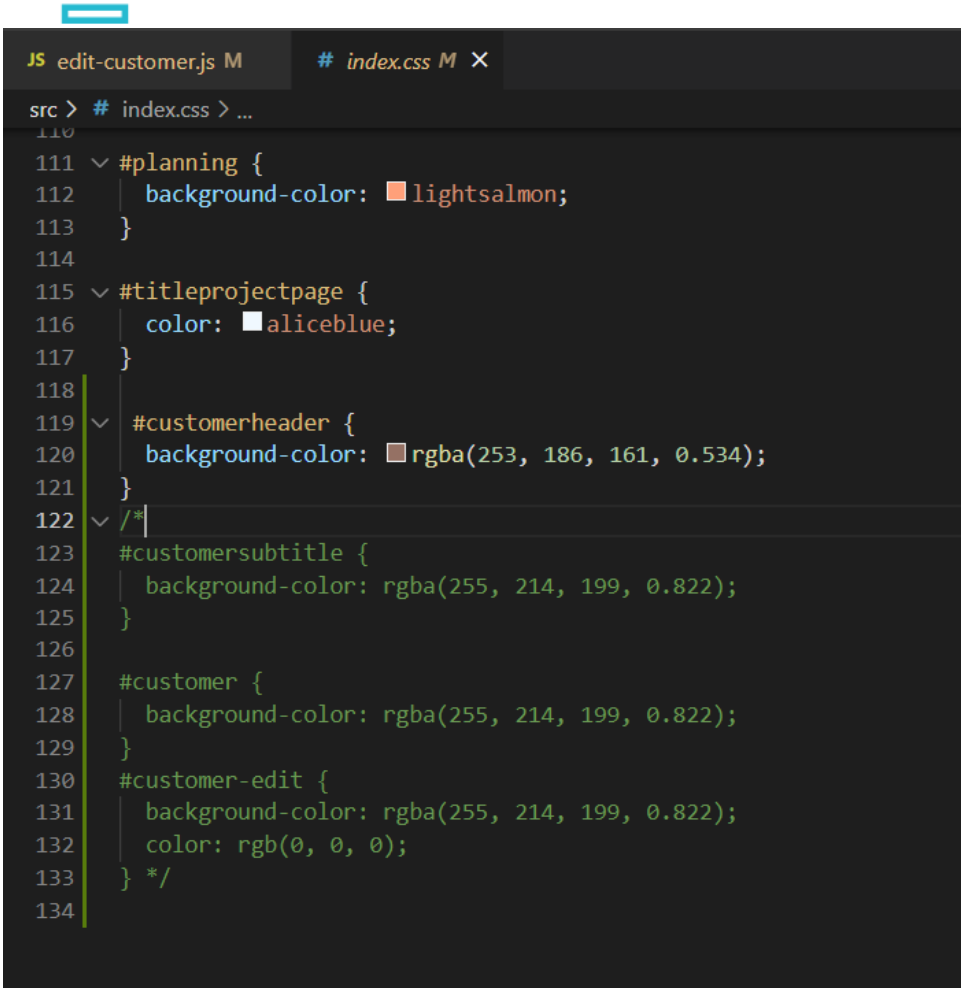
```
}

.importantRules {
  background-color: rgb(248, 163, 163);
  viewport-fit: 100%;
  color: rgb(1, 1, 105);
}

#importantbody {
  color: rgb(0, 0, 129);
  background-color: rgb(250, 135, 135);
}

.dogpic {
  max-width: 65%;
  max-height: 65%;
  margin: auto;
}

#homeheader {
  color: aqua;
}

#homeparagraphs {
  color: rgb(255, 255, 255);
}

.walkferret {
  max-width: 65%;
  max-height: 65%;
  margin: auto;
}

#contactheader {
  color: rgb(0, 102, 102);
  background-color: bisque;
}
```

PROMINEO TECH

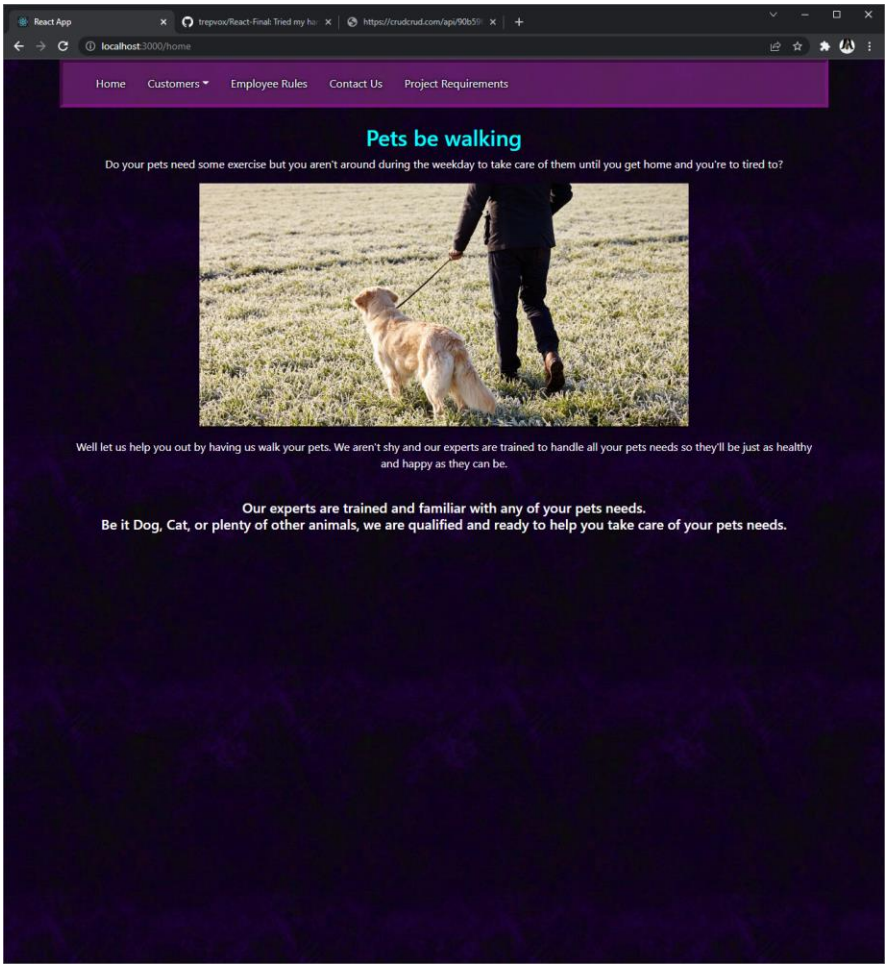
edit-customer.js M index.css M

src > # index.css > ...

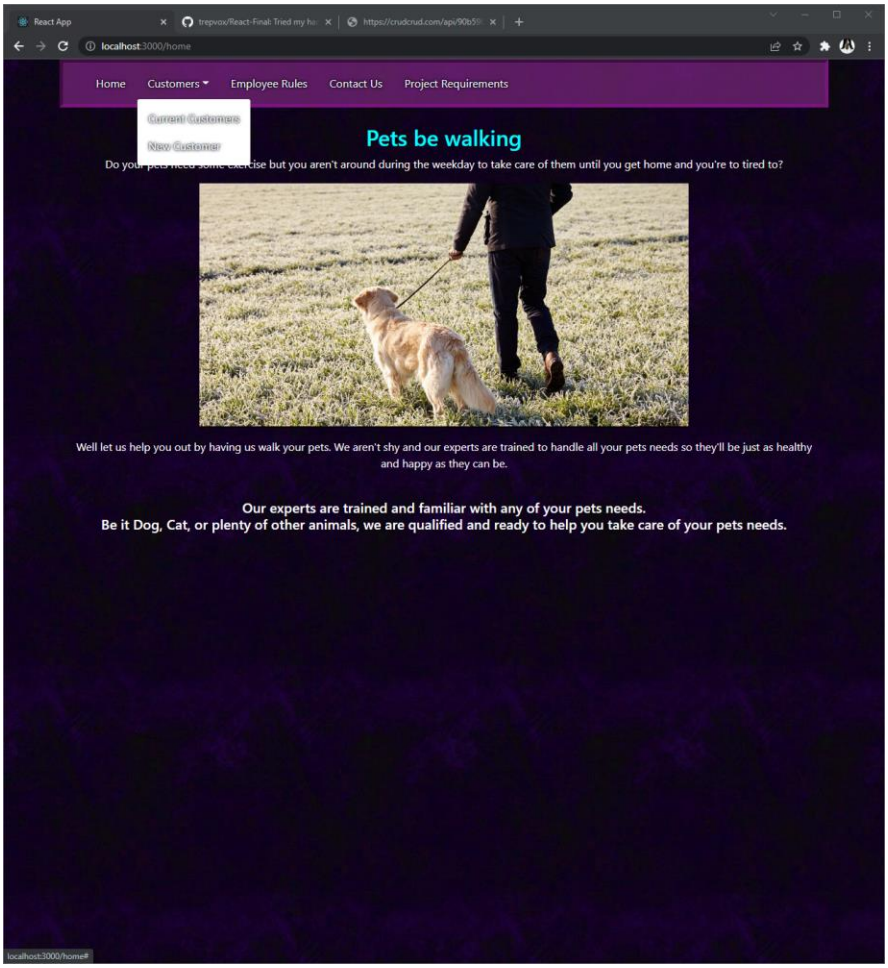
```
74
75  #contactbody {
76    color: rgb(0, 56, 102);
77    background-color: rgb(252, 244, 235);
78    border: curved;
79  }
80
81  .pawprint {
82    max-width: 25%;
83    max-height: 25%;
84    margin: auto;
85  }
86
87  Table {
88    background-color: rgb(157, 128, 177);
89  }
90
91  /* #table {
92    color: white;
93  } */
94
95  #requirement {
96    background-color: lightgreen;
97  }
98
99  #completed {
100   background-color: rgb(37, 170, 253);
101 }
102
103 #analysising {
104   background-color: rgb(241, 241, 141);
105 }
106
107 #debugging {
108   background-color: lightpink;
109 }
110
```



Screenshots of Running Application:

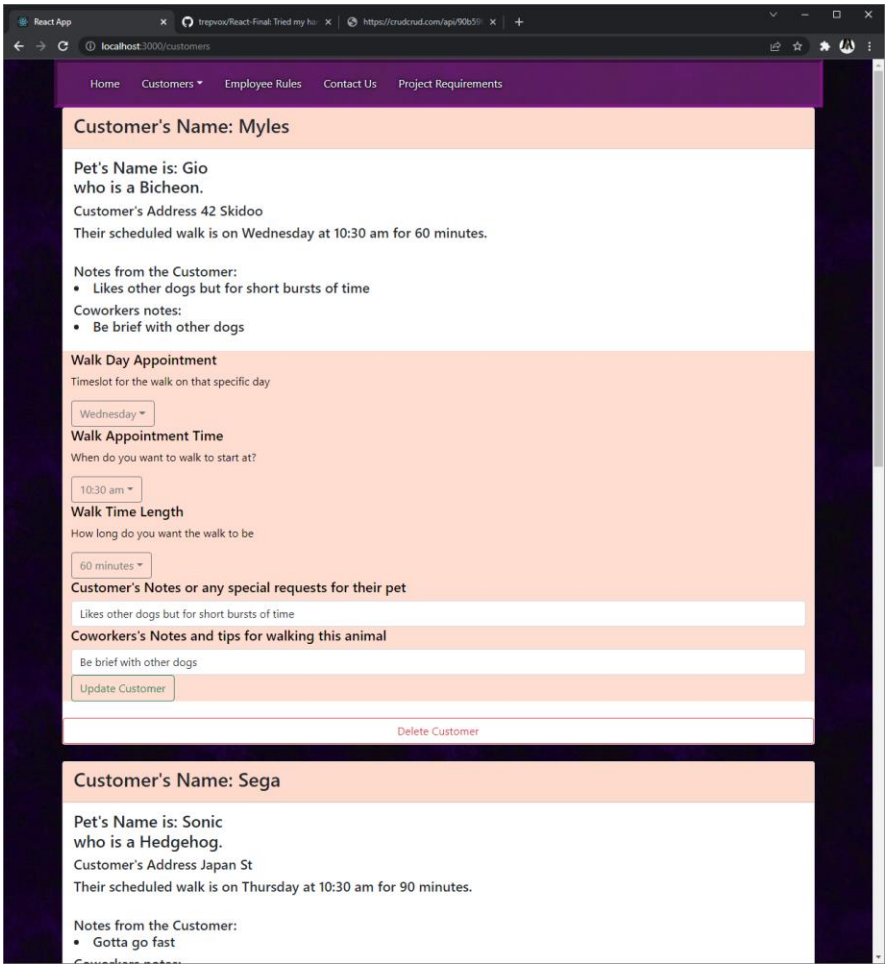


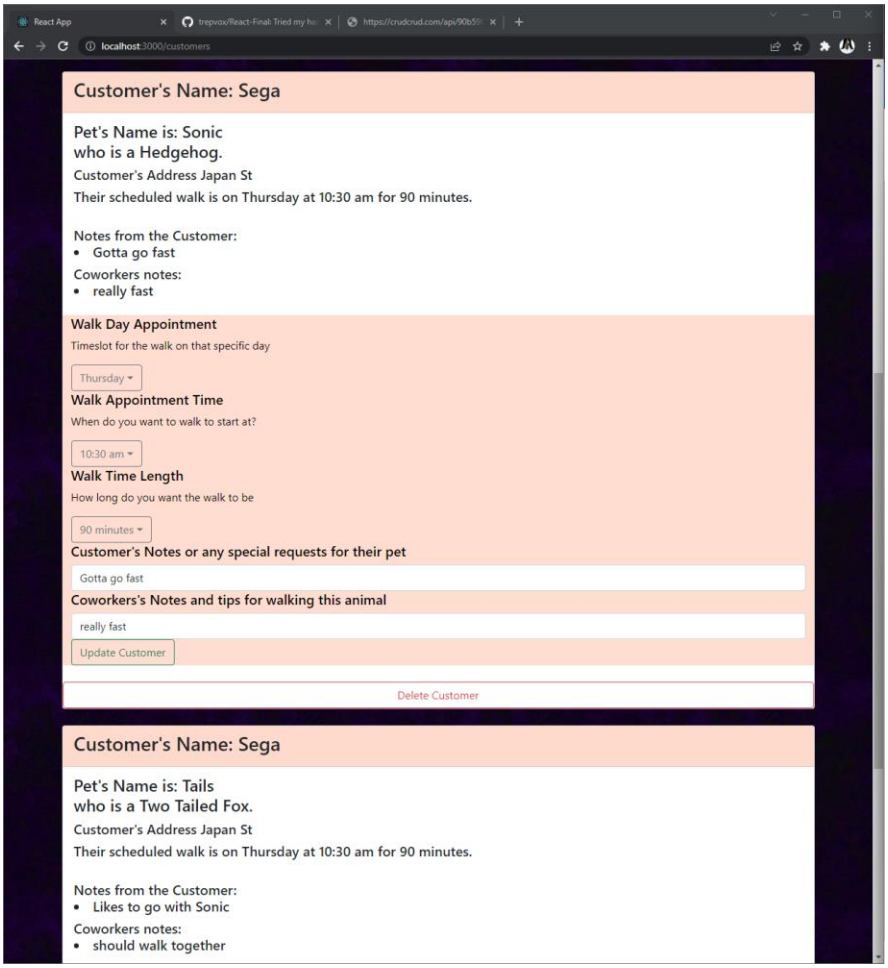
Showing the dropdown for Customers works



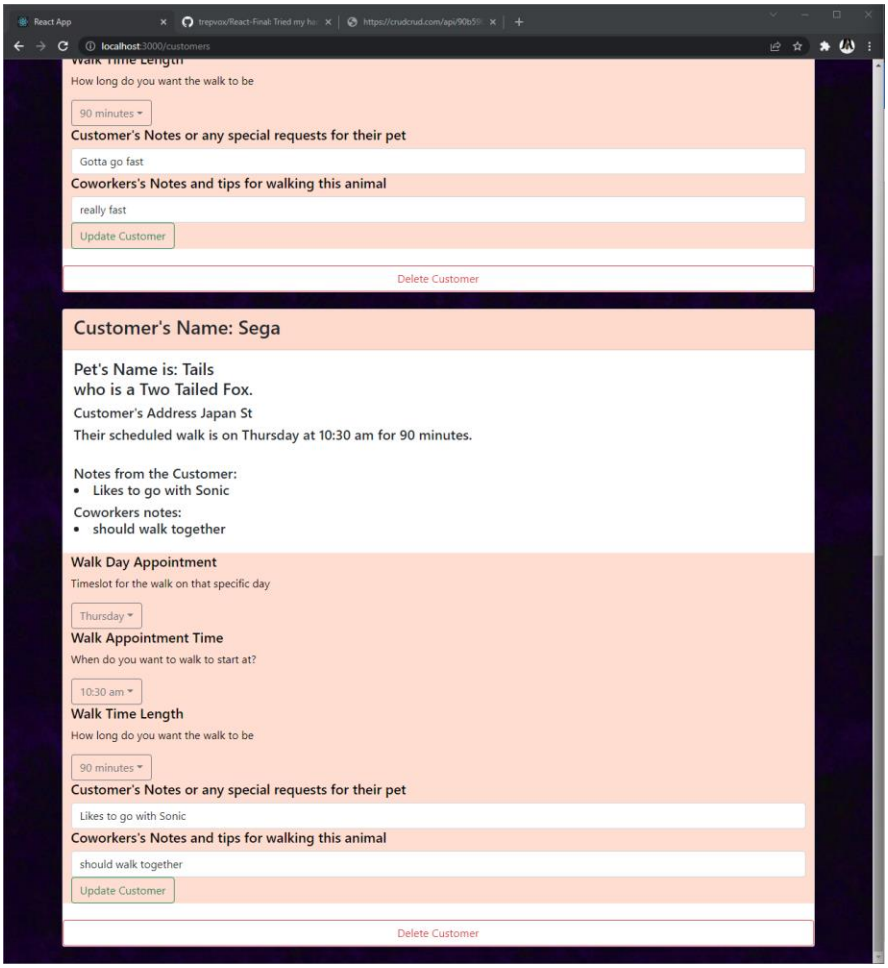
Current Customers

PROMINEO TECH

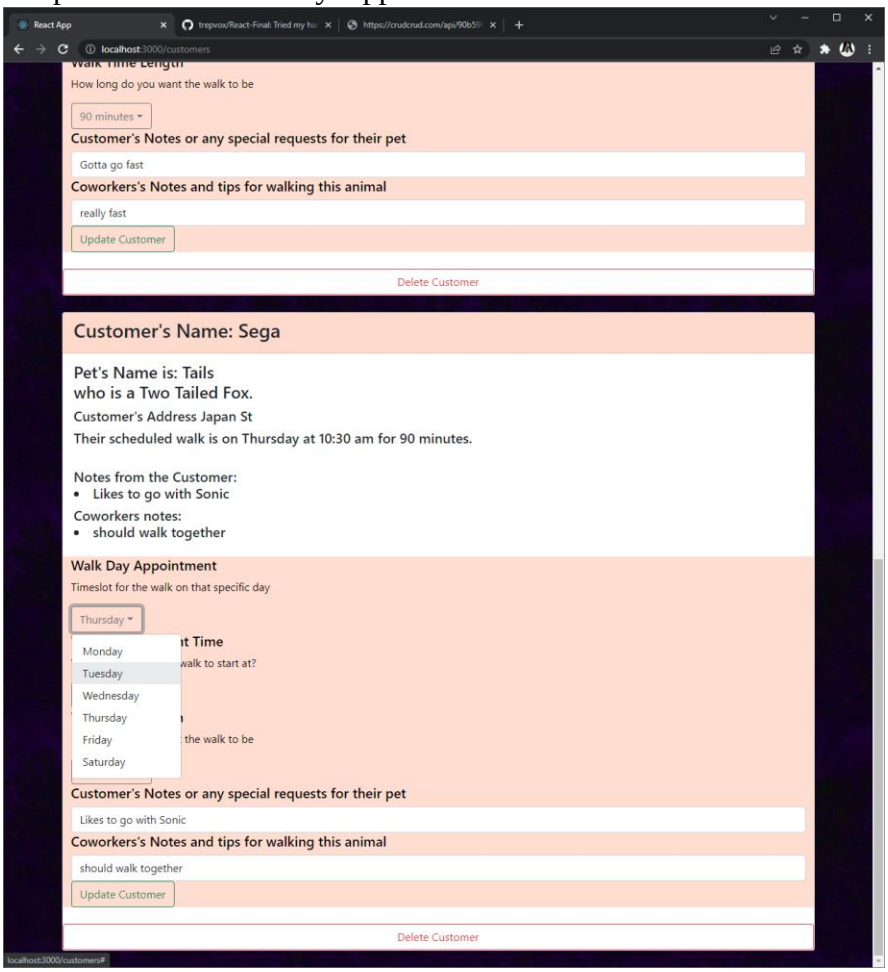



React App
localhost:3000/customers
Customer's Name: Sega
Pet's Name is: Sonic
who is a Hedgehog.
Customer's Address Japan St
Their scheduled walk is on Thursday at 10:30 am for 90 minutes.
Notes from the Customer:
Gotta go fast
Coworkers notes:
really fast
Walk Day Appointment
Timeslot for the walk on that specific day
Thursday
Walk Appointment Time
When do you want to walk to start at?
10:30 am
Walk Time Length
How long do you want the walk to be
90 minutes
Customer's Notes or any special requests for their pet
Gotta go fast
Coworkers's Notes and tips for walking this animal
really fast
Update Customer
Delete Customer
Customer's Name: Sega
Pet's Name is: Tails
who is a Two Tailed Fox.
Customer's Address Japan St
Their scheduled walk is on Thursday at 10:30 am for 90 minutes.
Notes from the Customer:
Likes to go with Sonic
Coworkers notes:
should walk together

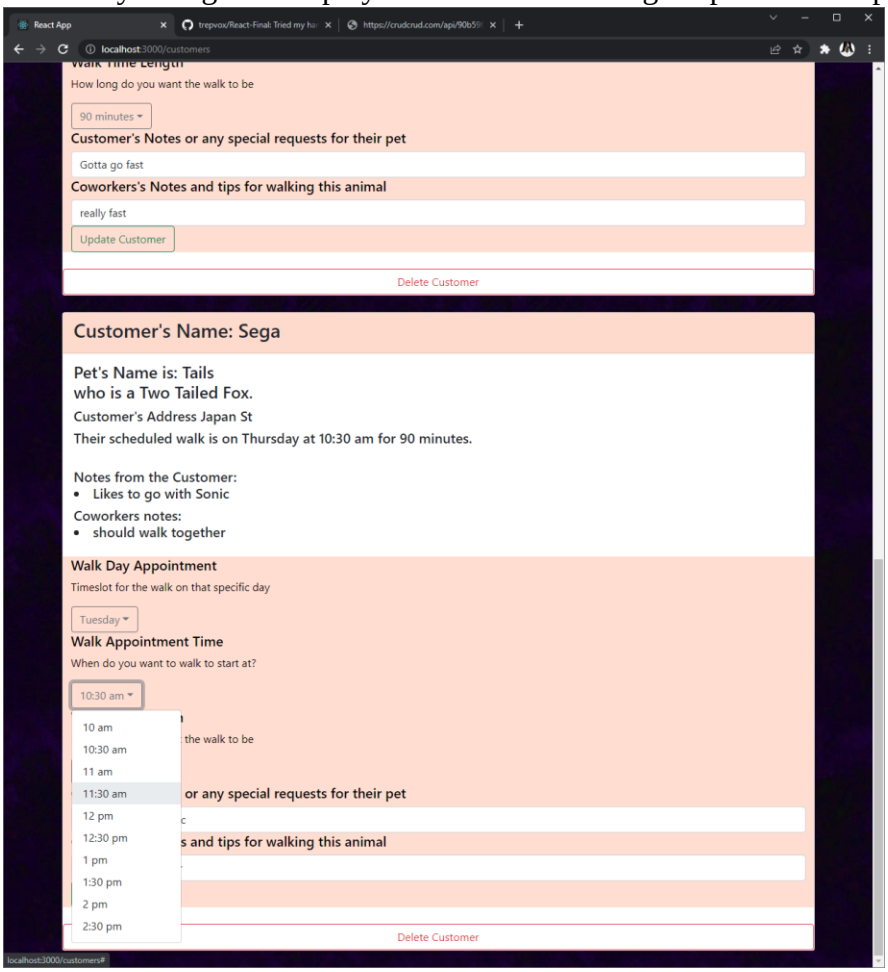
PROMINEO TECH



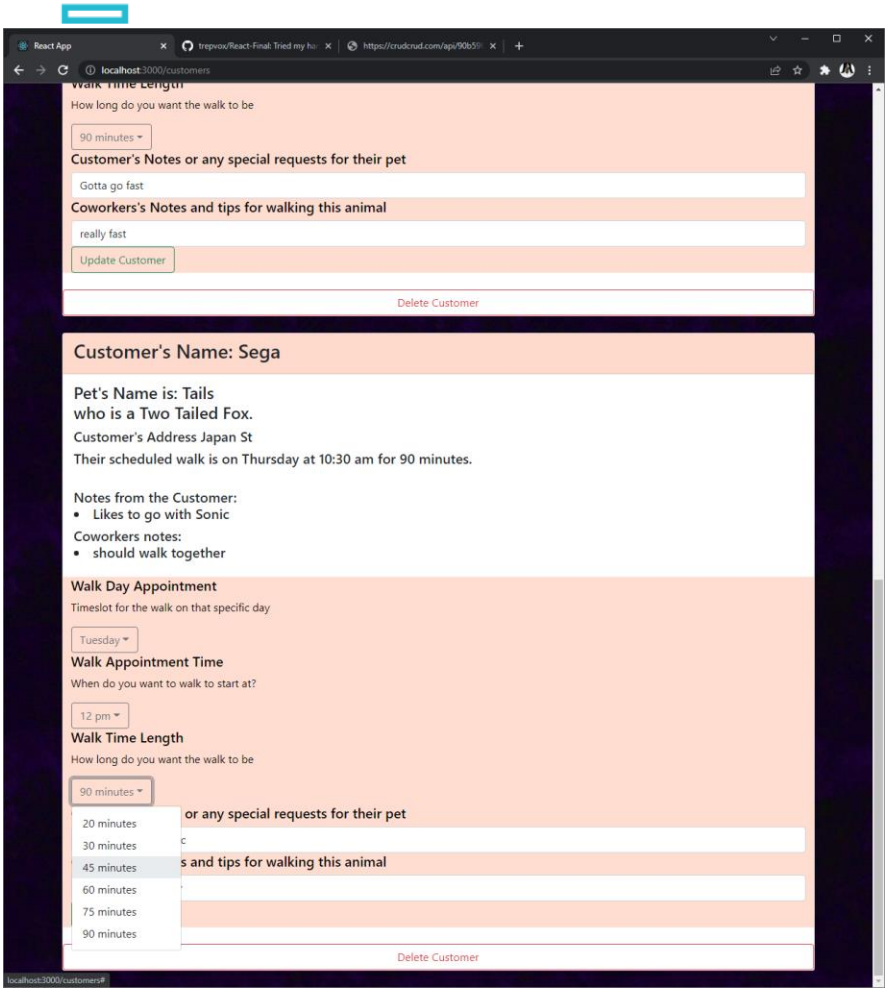
Dropdown for Walk Day Appointments



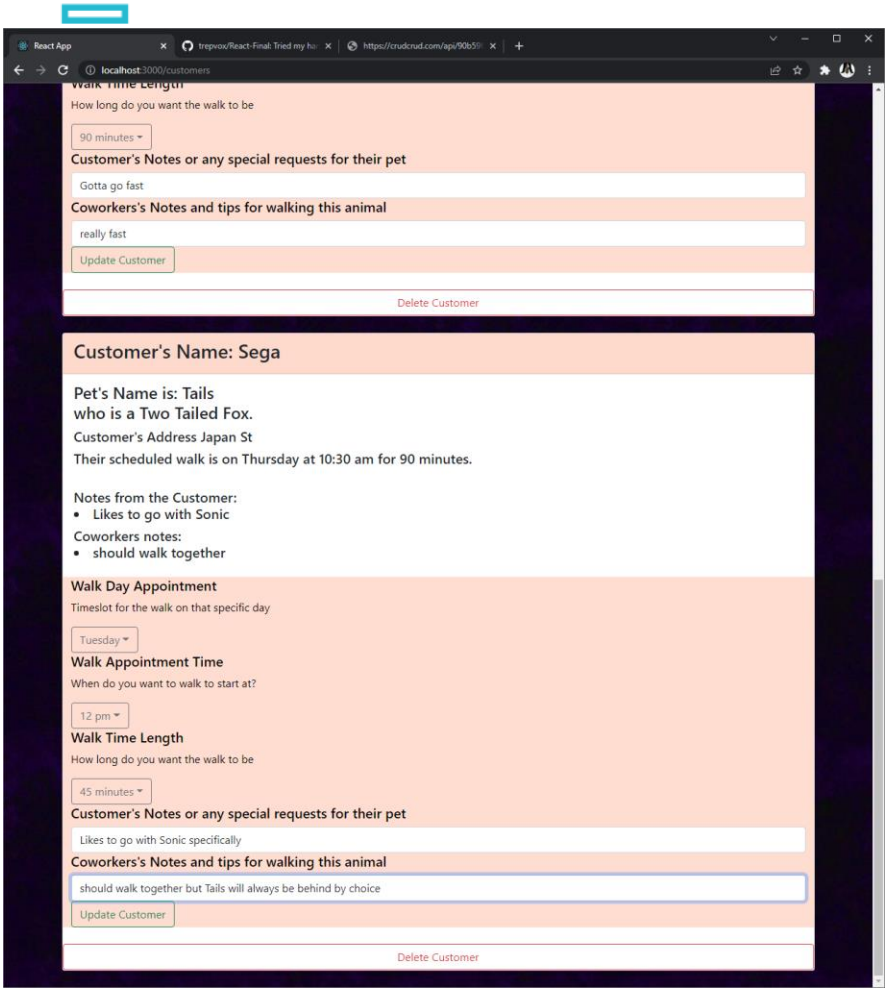
Walk Day changes to display selected and showing drop down for appointment time



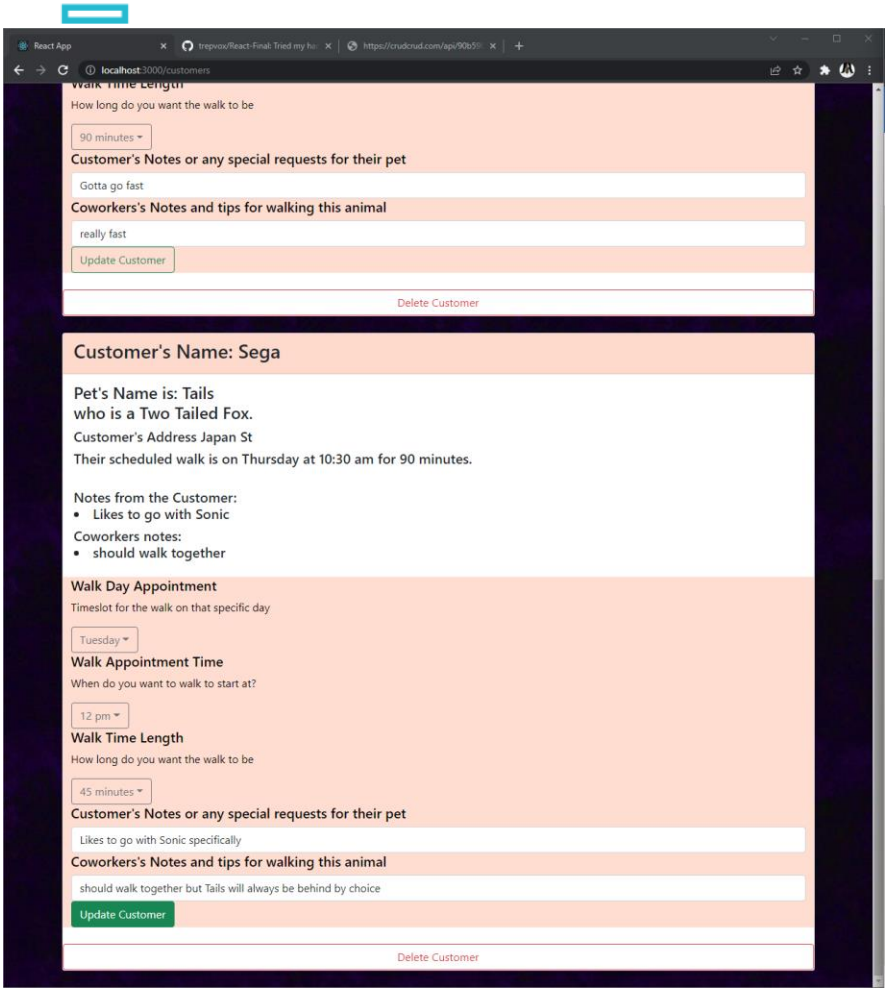
Appointment time shows selected and showing dropdown for time length



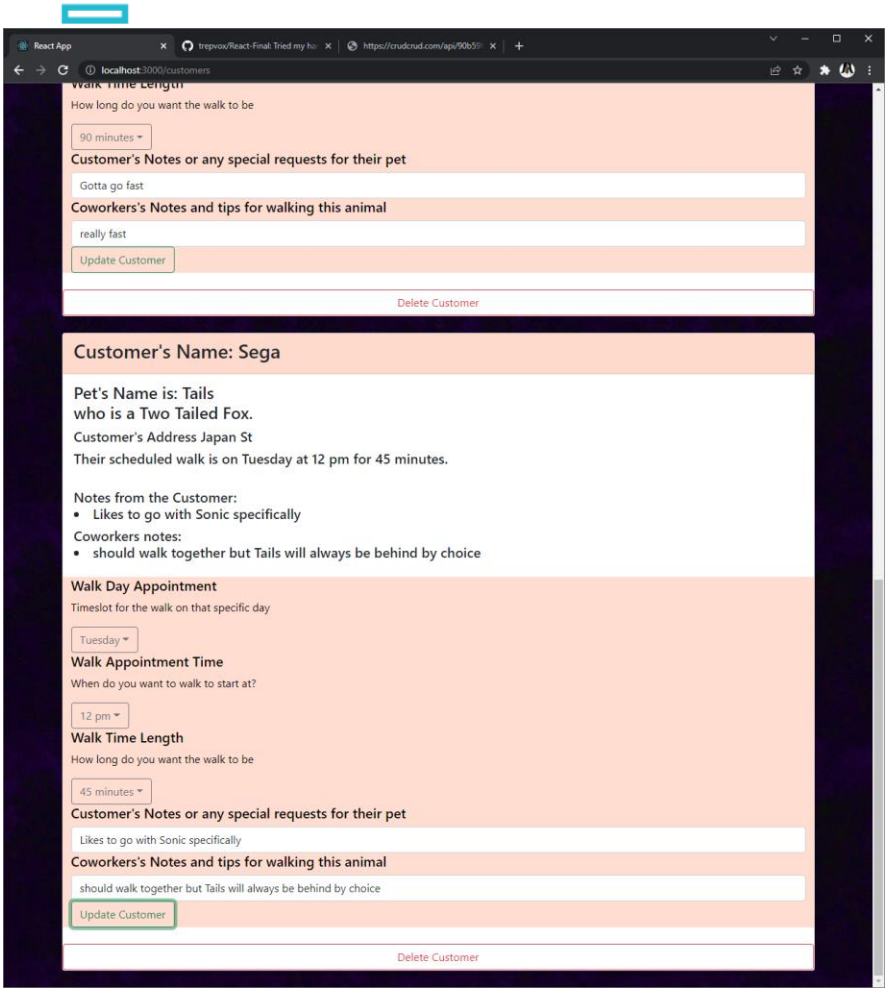
Showing the notes section both can be edited as well if need be.



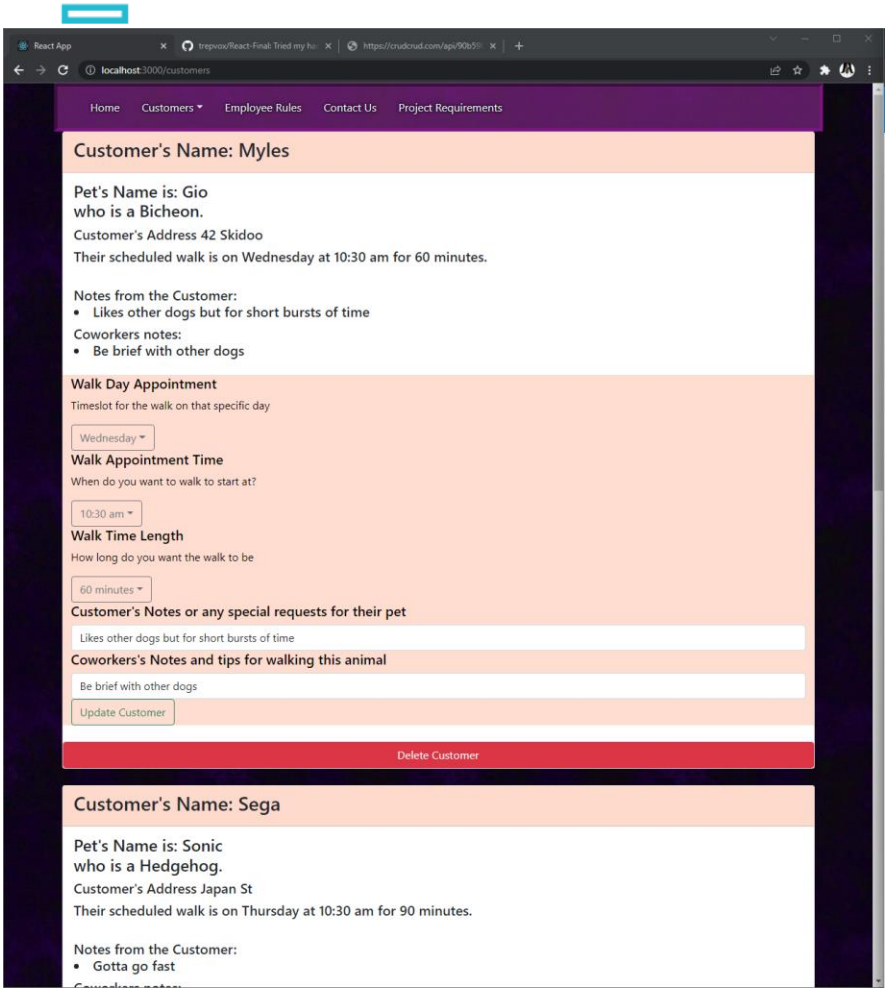
Hovering over update changes color



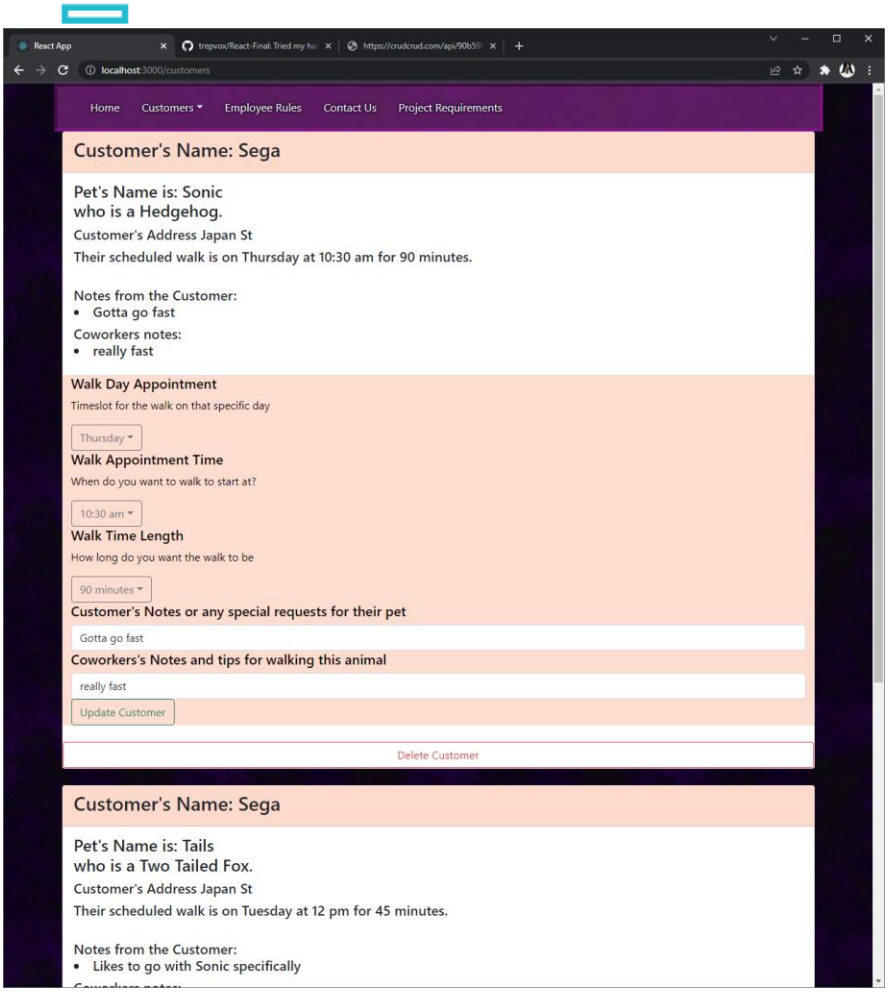
Update works displaying the changes in fact went up.



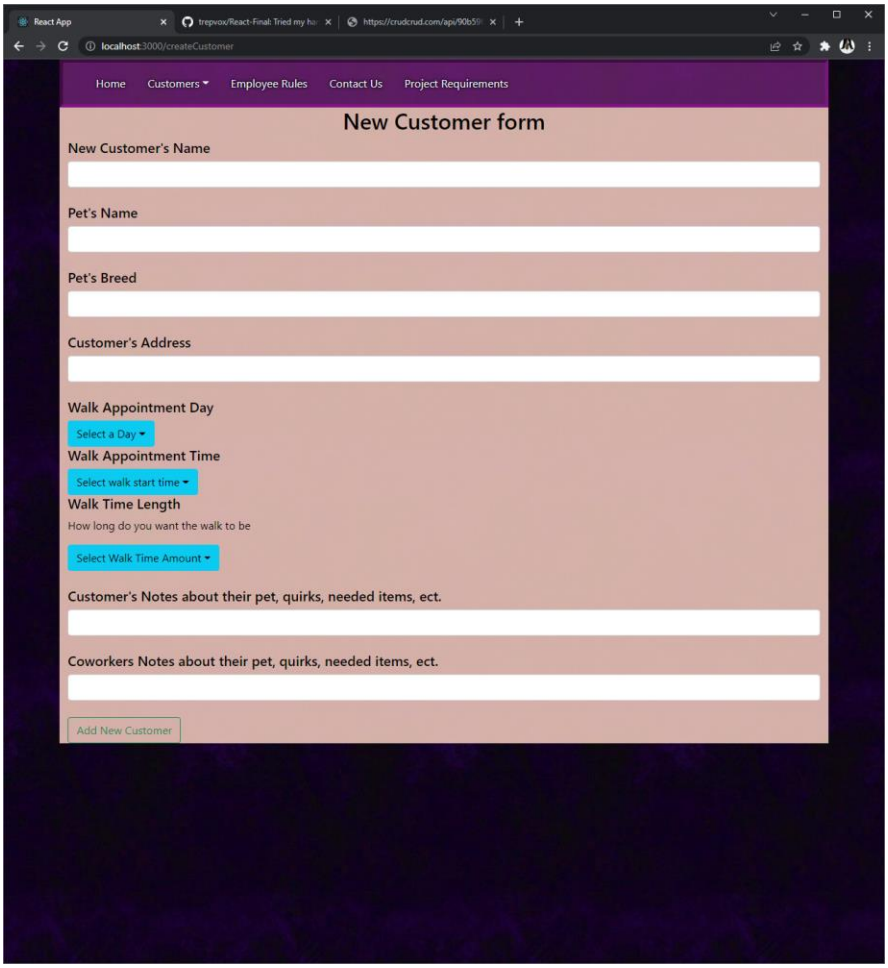
Hovering over Delete lights it up

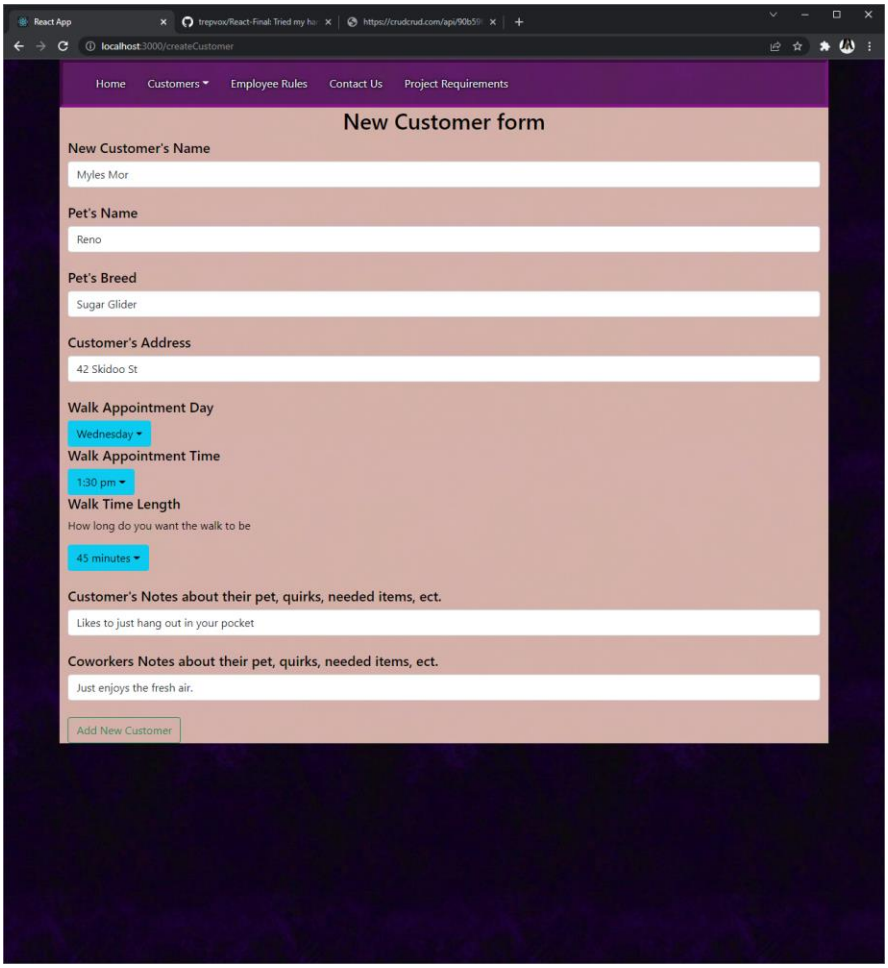


Clicking delete removes that customer

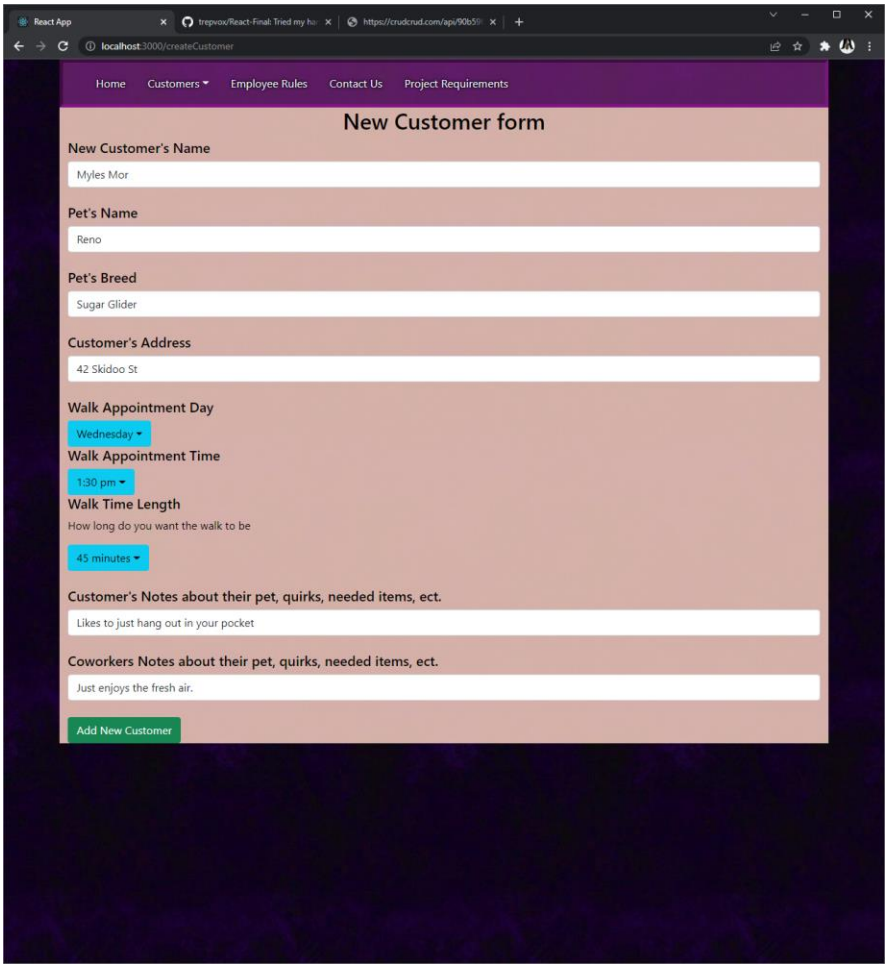


New Customer

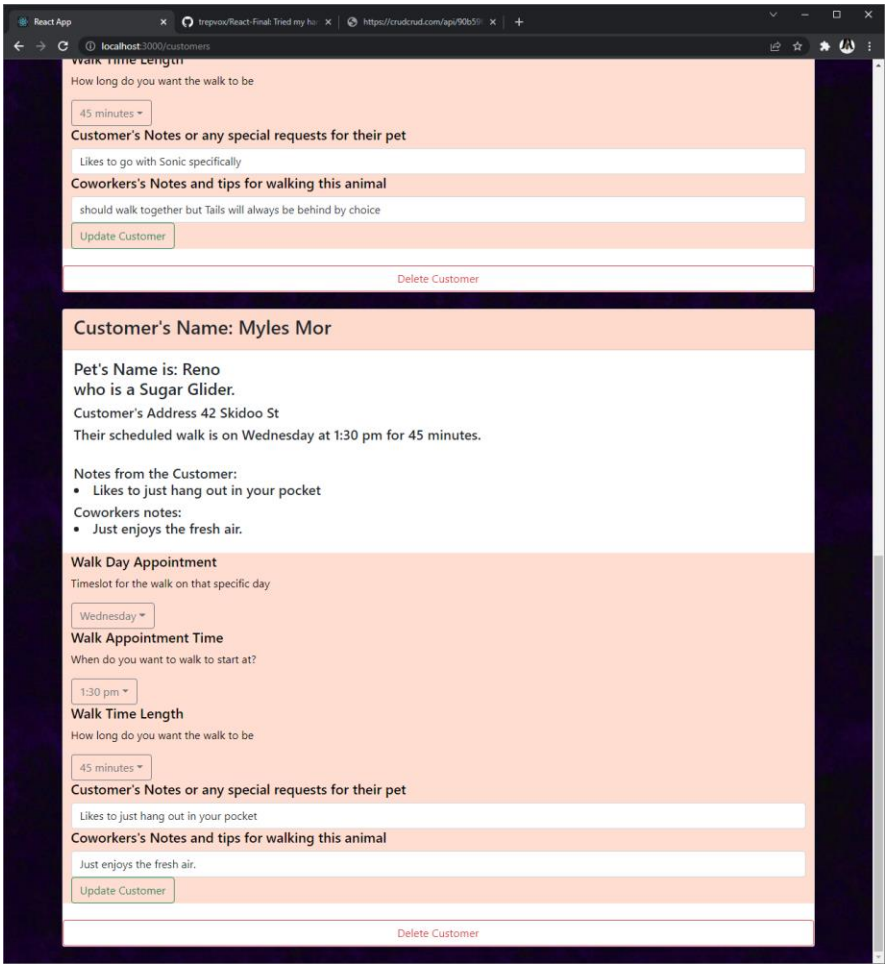

React App
localhost:3000/createCustomer
Home
Customers
Employee Rules
Contact Us
Project Requirements
New Customer form
New Customer's Name
Pet's Name
Pet's Breed
Customer's Address
Walk Appointment Day
Select a Day
Walk Appointment Time
Select walk start time
Walk Time Length
How long do you want the walk to be
Select Walk Time Amount
Customer's Notes about their pet, quirks, needed items, ect.
Coworkers Notes about their pet, quirks, needed items, ect.
Add New Customer



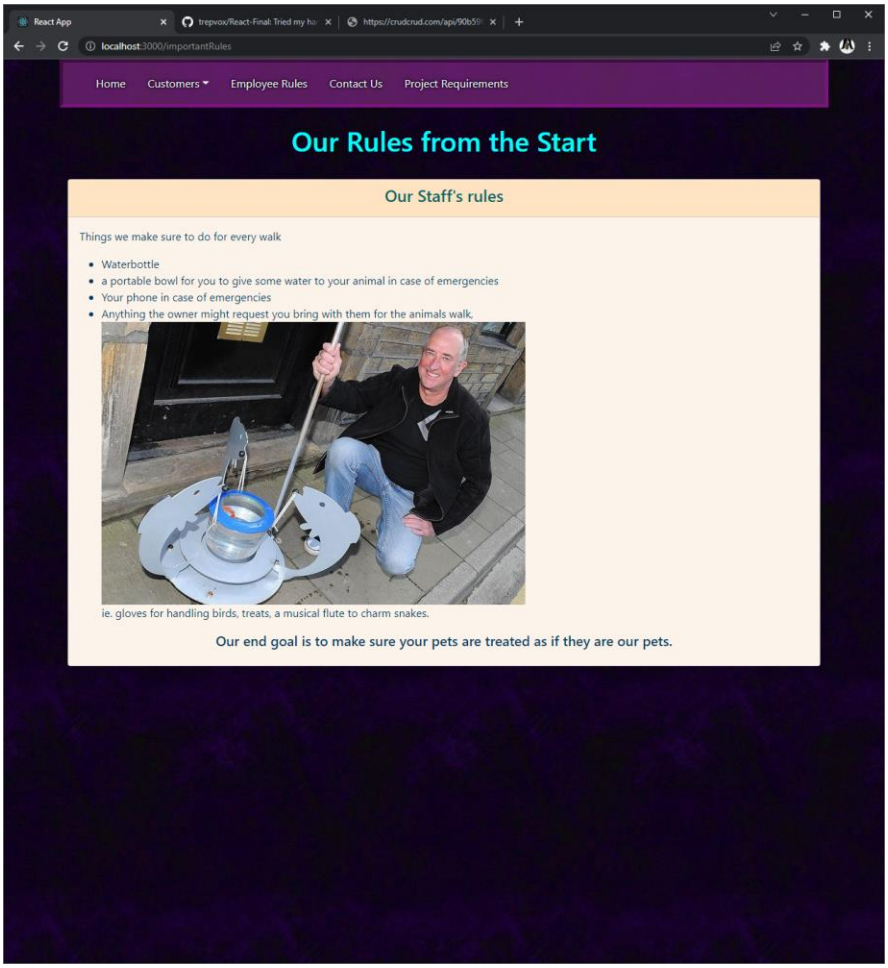
Clicking Add New Customer turns it green and pushes the customer to current customers



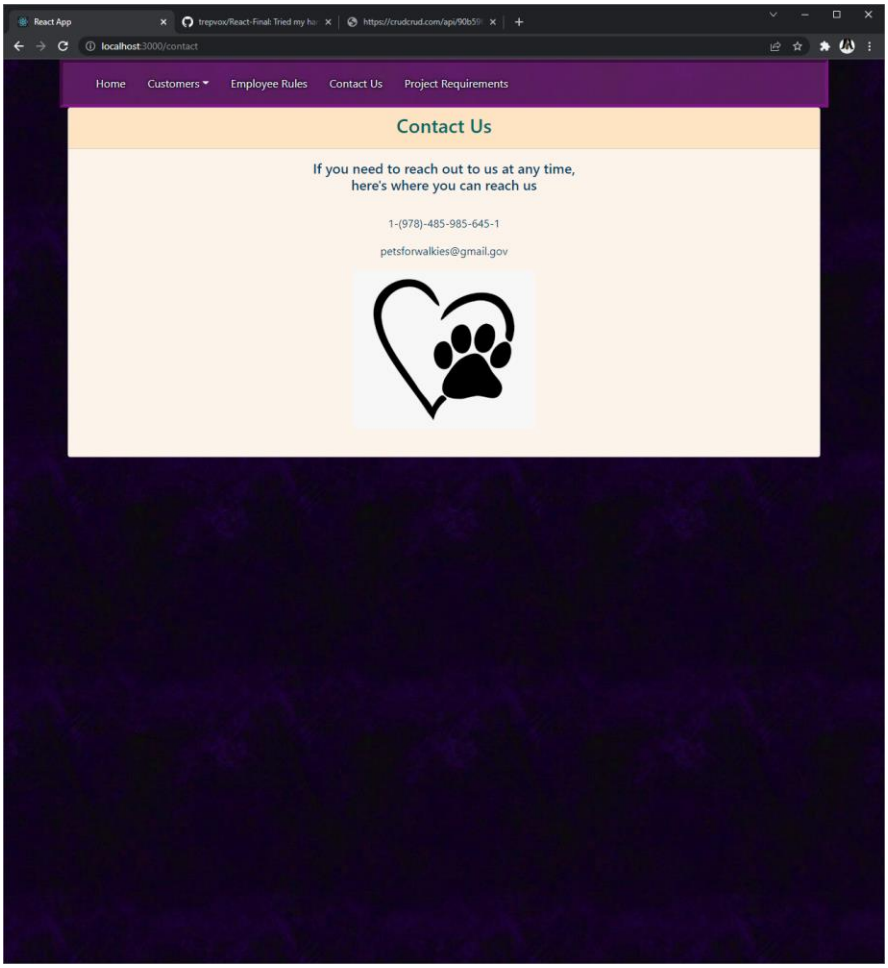
New customer has been added



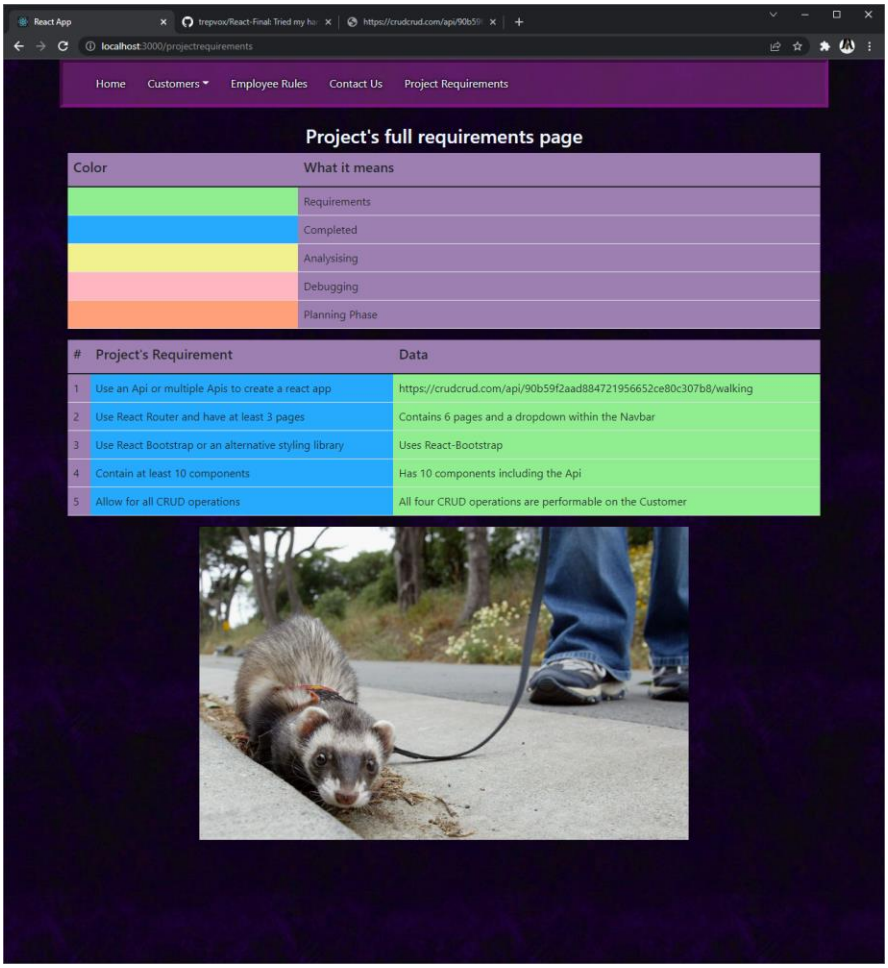
Employee Rules


React App
localhost:3000/importantRules
Home
Customers
Employee Rules
Contact Us
Project Requirements
Our Rules from the Start
Our Staff's rules
Things we make sure to do for every walk
Waterbottle
a portable bowl for you to give some water to your animal in case of emergencies
Your phone in case of emergencies
Anything the owner might request you bring with them for the animals walk.
ie. gloves for handling birds, treats, a musical flute to charm snakes.
Our end goal is to make sure your pets are treated as if they are our pets.

Contact Us



Project Requirements



URL to GitHub Repository:

https://github.com/trepvox/React-Final