



Web App Design with React Week 3 Coding Assignment

Lisa Smith — 2021-10-05-fesd-nashua — Due: 2/12/22

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.

Description: I decided to create a Recipe Builder using the following url:

<https://crudcrud.com/api/467446f2b41f4ea58e1a553fd9faaf72>



Recipe Builder

The following are allowed:

- CREATE a Recipe — **POST** on path “/recipes”
- READ all Recipes — **GET** on “/recipes”
- UPDATE a Recipe — ADD & DELETE Ingredients — **PUT** on “recipes/recipeId”
- DELETE a Recipe — **DELETE**(recipeId)

Components: **RecipeList, Recipe, NewRecipeForm, NewIngredientForm**

Rest Implementation: **RecipeApi**

Object definition:

- **Recipe:** Name, Category, NumberServed, Ingredients=[]
- **Ingredient:** Name, Quantity, Units, Instructions

Screenshots of Code:

Recipe.js:

```
RecipeApi.js   JS NewRecipeForm.js   JS Recipe.js M X
FE-ReactWeek3-HW > recipe-builder > src > components > JS Recipe.js > Recipe
1  import React from 'react';
2  import { NewIngredientForm } from './NewIngredientForm';
3
4  // stateless or functional component
5  export const Recipe = (props) => {
6    const { recipe, updateRecipe } = props;
7
8    const deleteIngredient = (ingredientName) => {
9      const updatedRecipe = {
10        ...recipe,
11        ingredients: recipe.ingredients.filter(x => x.name !== ingredientName)
12      };
13      updateRecipe(updatedRecipe);
14    };
15
16    // Don't need a return, because it's a one-liner, and will automatically
17    // return the results.
18    // always add a new value in an array or an object...
19    const addNewIngredient = (ingredient) => updateRecipe({ ...recipe, ingredients: [...recipe.ingredients, ingredient]});
20
21    // bug here: I had "(" instead of "(" — rooms didn't render.
22    const ingredients = () => {
23      recipe.ingredients
24      ? <ul>
25        {recipe.ingredients.map((ingredient, index) => {
26          <li key={index}>
27            <label> <strong>{ingredient.name}</strong> {ingredient.quantity} {ingredient.units} <span>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</span>
28            <strong> &mdash; </strong> <span>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</span> {ingredient.instructions} </label>
29            <span>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</span>
30            <button className="btn-my-color rounded" onClick={e => deleteIngredient(ingredient.name)}>Delete Ingredient</button>
31            <br /> <br />
32          </li>
33        })}
34      </ul>
35      : null
36    )
37
38    // This works because: ingredients() is a function
39    // It is taking props: ingredients, recipeId, deleteIngredient function
40
41    return (
42      <div className="card-recipe">
43        <h1>{recipe.name}</h1>
44        <h4 className="tab"><strong>Category:</strong> {recipe.category} <span>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</span><strong> Number of People Served:</strong> {recipe.numberServed}</h4>
45        <br />
46        <h4><strong>Ingredients:</strong></h4>
47        {
48          ingredients ({ ingredients, recipeId: recipe_id, deleteIngredient})
49        }
50        <NewIngredientForm addNewIngredient={addNewIngredient} />
51      </div>
52    );
53  };
54
```



PROMINEO TECH

index.css

```
# index.css X # App.css # Recipe.js # NewRecipeForm.js # NewIngredientForm.js # RecipesList.js index.html # RecipeA
FE-ReactWeek3-HW > recipe-builder > src > # index.css > ul
1 body {
2   margin: 0;
3   color: #slateblue;
4   font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',
5     'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
6     sans-serif;
7   -webkit-font-smoothing: antialiased;
8   -moz-osx-font-smoothing: grayscale;
9 }
10
11 code {
12   font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',
13     monospace;
14 }
15
16 ul {
17   padding-left: 0px;
18 }
19
20 h1, h2, h3, h4, h5 {
21   color: #darkslateblue;
22   font-family: fantasy;
23   font-weight: 600;
24 }
25
26 button {
27   font-family: fantasy;
28 }
29
30 .recipe-div {
31   background: #lightsalmon;
32   border-width: 4;
33   border-radius: 20px;
34 }
35
36 div {
37   border-width: 4;
38   border-radius: 20px;
39 }
40
41 .new-ingredient {
42   background: #lightsalmon;
43 }
44
45 .img-recipe {
46   background-color: #lightsteelblue;
47   background-image: url("https://images.pexels.com/photos/616401/pexels-photo-616401.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500.png");
48 }
49
50 .new-recipe {
51   background-color: #lightsteelblue;
52 }
53
54 .single-recipe {
55   border-width: 4;
56   border-radius: 20px;
57   background: #lightsteelblue;
58 }
59
60 label {
61   color: #darkslateblue;
62 }
63
64 .btn-my-color {
65   color: #white;
66   background-color: #darkslateblue;
67 }
68
69 .tab {
70   margin-left: 2.5em
71 }
72
73 .round-corner {
74   border-width: 4;
75   border-radius: 20px;
76 }
```



PROMINEO TECH

RecipesList.js:

```
E-ReactWeek3-HW > recipe-builder > src > components > JS RecipesList.js > RecipesList > deleteRecipe
1 import React from 'react';
2 import { Recipe } from '../Recipe';
3 import { recipeApi } from '../rest/RecipeApi.js';
4 import { NewRecipeForm } from './NewRecipeForm';
5
6
7 export class RecipesList extends React.Component {
8   state = {
9     recipes: []
10  };
11
12  componentDidMount() {
13    this.fetchRecipes();
14  }
15
16  fetchRecipes = async () => {
17    const returnedRecipes = await recipeApi.get();
18    // console.log('In fetchRecipes: ' + returnedRecipes);
19    this.setState({ recipes: returnedRecipes });
20  };
21
22  createRecipe = async (newRecipe) => {
23    await recipeApi.post(newRecipe);
24    this.fetchRecipes();
25  };
26
27  updateRecipe = async (updatedRecipe) => {
28    //console.log('In updateRecipe: ' + updatedRecipe);
29    await recipeApi.put(updatedRecipe);
30    this.fetchRecipes();
31  };
32
33  deleteRecipe = async (recipeId) => {
34    await recipeApi.delete(recipeId);
35    this.fetchRecipes();
36  };
37
38  render() {
39    console.log('In render: ' + this.state.recipes);
40    const recipes = this.state.recipes
41    ? this.state.recipes.map((recipe) =>
42      <div className="round-corner container-fluid">
43        <div className="single-recipe border p-2">
44          <Recipe
45            key={recipe._id}
46            recipe={recipe}
47            updateRecipe={this.updateRecipe}
48          /><br />
49          <button className="btn-my-color rounded" onClick={e =>
50            this.deleteRecipe(recipe._id)}>Delete Recipe</button>
51          <br /><br />
52        </div>
53        <br /> <br />
54      </div>)
55    : null;
56    console.log('Recipes in RecipeList: ' + recipes);
57
58    return (
59      <div className="recipe-div p-2 border recipe-list">
60        <h1>Recipe Builder</h1>
61        <div>
62          <ul>
63            {recipes}
64          </ul>
65        </div>
66
67        <div>
68          <NewRecipeForm createRecipe={this.createRecipe}/>
69        </div>
70      </div>
71    );
72  };
73 }
```



```

1  # NewRecipeForm.js X      # NewIngredientForm.js      # RecipeApi.js
FE-React/Week3-HW > recipe-builder > src > components > # NewRecipeForm.js > NewRecipeForm > onSubmit
2  import React, { useState } from 'react';
3  import { Form } from 'react-bootstrap';
4
5  // importing { useState } allows us to use Hooks!
6
7  export const NewRecipeForm = (props) => {
8    const { createRecipe } = props;
9    const [name, setName] = useState('');
10   const [numberServed, setNumberServed] = useState('');
11   const [category, setCategory] = useState('');
12   const [ingredients, setIngredients] = useState('');
13
14   const handleNumberServedInput = (e) => {
15     const int = parseInt(e.target.value, 10);
16     // no negative numbers... if negative, pass in an empty string
17     setNumberServed(int >= 0 ? int : '');
18   } // end of handleQuantityInput
19
20   const onSubmit = (e) => {
21     // the default submit action WILL not occur!
22     e.preventDefault();
23     if (name && category) {
24       createRecipe(name, numberServed, category, ingredients);
25       setName('');
26       setCategory('');
27       setNumberServed('');
28       setIngredients('');
29     } else {
30       alert('Invalid input: Please re-enter ingredient!');
31       console.log('Invalid Recipe Input— needs to be re-entered!');
32     }
33   } // end of onSubmit
34
35   // create the JSX for this element — create the return statement
36
37   // name: everytime the text changes... this name value in the state is also being updated
38   // quantity: everytime the text changes... call handleQuantityInput
39   // units: everytime the text changes... this units value in the state is also being updated
40   // instructions: everytime the text changes... this instructions value in the state is also being updated
41
42   // the last part of tying the state variable back to the actual value of these inputs
43   // is "value={name}", "value={quantity}", "value={units}" & "value={instructions}"
44   // the onChange always updates the state value,
45   // and the state value updates the value in this input.
46   // So they stay tied together both ways!!!
47
48   return (
49     <div className="new-recipe">
50       <div className="p-2 border">
51         <h4><strong>Enter a New Recipe</strong></h4>
52         <Form onSubmit={onSubmit}>
53           <input
54             type="text"
55             placeholder="Recipe Name"
56             onChange={(e) => setName(e.target.value)}
57             value={name}
58           />
59           <span>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</span>
60           <input
61             type="text"
62             placeholder="Number of Servings"
63             onChange={handleNumberServedInput}
64             value={numberServed}
65           />
66           <span>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</span>
67           <input
68             type="text"
69             placeholder="Main, App, Dessert..."
70             onChange={(e) => setCategory(e.target.value)}
71             value={category}
72           />
73           <span>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</span>
74           <button
75             className="btn-my-color rounded"
76             type="submit"
77             onClick={onSubmit}>Add Recipe</button>
78         </Form>
79       </div>
80     </div>
81   ) // end of return
82 } // end of NewIngredientForm function

```



```

JS NewIngredientForm.js X JS RecipeApi.js
FE-ReactWeek3-HW > recipe-builder > src > components > JS NewIngredientForm.js > NewIngredientForm
1 import React, { useState } from 'react';
2 import { Form } from 'react-bootstrap';
3
4 // importing { useState } allows us to use Hooks!
5
6 export const NewIngredientForm = (props) => {
7   const [name, setName] = useState('');
8   const [quantity, setQuantity] = useState(undefined);
9   const [units, setUnits] = useState('');
10  const [instructions, setInstructions] = useState('');
11
12  const handleQuantityInput = (e) => {
13    const int = parseInt(e.target.value, 10);
14    setQuantity(int >= 0 ? int : '');
15  } // end of handleQuantityInput
16
17  const onSubmit = (e) => {
18    // the default submit action WILL not occur!
19    e.preventDefault();
20    if (name && quantity && units) {
21      // console.log("In OnSubmit Ingredients");
22      props.addNewIngredient({name, quantity, units, instructions});
23      setName('');
24      setQuantity('');
25      setUnits('');
26      setInstructions('');
27    } else {
28      console.log('Invalid Ingredient Input');
29    }
30  }; // end of onSubmit
31
32  // create the JSX for this element -- create the return statement
33
34  // the last part of tying the state variable back to the actual value of these inputs
35  // is "value={name}", "value={quantity}", "value={units}" & "value={instructions}"
36  // The onChange always updates the state value,
37  // & the state value updates the value in this input, so stay tied together both ways
38
39  return (
40    <div className="new-ingredient border p-2">
41      <h4><strong>Add a New Ingredient</strong></h4>
42      <Form onSubmit={onSubmit}>
43        <input
44          type='text'
45          placeholder='Name'
46          onChange={(e) => setName(e.target.value)}
47          value={name}
48        /> <span>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</span></span>
49        <input
50          type='text'
51          placeholder='How many -- #'
52          onChange={handleQuantityInput}
53          value={quantity}
54        /> <span>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</span></span>
55        <input
56          type='text'
57          placeholder='Tbsp, tsp, lb, oz, etc.'
58          onChange={(e) => setUnits(e.target.value)}
59          value={units}
60        /> <br /><br />
61        <input
62          type='text'
63          placeholder='Any instructions?'
64          onChange={(e) => setInstructions(e.target.value)}
65          value={instructions}
66        /> <span>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</span></span>
67        <button className='btn-my-color rounded' type='submit'>Add Ingredient</button>
68      </Form>
69    </div>
70  ); // end of return
71 }; // end of NewIngredientForm function

```



PROMINEO TECH

RecipeApi.js:

```
JS RecipeApi.js X
FE-ReactWeek3-HW > recipe-builder > src > rest > JS RecipeApi.js > RecipeApi > delete
1 // all of the logic to make a network call for the HousesAPI
2 const RECIPE_ENDPOINT = 'https://crudcrud.com/api/467446f2b41f4ea58e1a553fd9faaf72/recipes';
3
4 // class to house a couple of functions or methods
5 class RecipeApi {
6   get = async () => {
7     try {
8       console.log(RECIPE_ENDPOINT);
9       const resp = await fetch(RECIPE_ENDPOINT);
10      console.log(resp);
11      const data = await resp.json();
12      console.log(data);
13      return data;
14    } catch(e) {
15      console.log('Oops, looks like fetchRecipes had an issue.', e);
16    }
17  }
18
19   post = async (recipe) => {
20     try {
21       const resp = await fetch(`${RECIPE_ENDPOINT}`, {
22         method: 'POST',
23         headers: {
24           'Content-Type': 'application/json'
25         },
26         body: JSON.stringify(recipe)
27       });
28       return await resp.json();
29     } catch(e) {
30       console.log('Oops, looks like posting recipes had an issue.', e);
31     }
32   }
33
34   put = async (recipe) => {
35     try {
36       console.log(RECIPE_ENDPOINT);
37       const resp = await fetch(`${RECIPE_ENDPOINT}/${recipe.id}`, {
38         method: 'PUT',
39         headers: {
40           'Content-Type': 'application/json'
41         },
42         body: JSON.stringify({ "name": recipe.name, "numberServed": recipe.numberServed, "category": recipe.category, "ingredients": recipe.ingredients })
43       });
44       console.log(resp);
45       return await resp.json();
46     } catch(e) {
47       console.log('Oops, looks like updating recipes had an issue.', e);
48     }
49   }
50
51   delete = async (recipeId) => {
52     try {
53       console.log(RECIPE_ENDPOINT);
54       const resp = await fetch(`${RECIPE_ENDPOINT}/${recipeId}`, {
55         method: 'DELETE'
56       });
57       console.log(resp);
58     } catch(e) {
59       console.log('Oops, looks like delete recipe had an issue.', e);
60     }
61   }
62 }
63
64 // create an instance of this class, and we can import that instance.
65 export const recipeApi = new RecipeApi();
```



PROMINEO TECH

Screenshots of Running Application:

Initial Screen (with one recip, no ingredients):

Recipe Builder

Buffalo Chicken Dip

Category: Appetizer Number of People Served: 12

Ingredients:

Add a New Ingredient

Name How many -- # Tbsp, tsp, lb, oz, etc.

Any instructions? Add Ingredient

Delete Recipe

Enter a New Recipe

Recipe Name Number of Servings Main, App, Dessert... Add Recipe

crudcrud.com view:

```
[{"_id": "6207cb8c38fd8d03e8973538", "name": "Buffalo Chicken Dip", "numberServed": 12, "category": "Appetizer", "ingredients": []}]
```




PROMINEO TECH

Entered information in NewIngredientForm:

Before Button Click:

Recipe Builder

Buffalo Chicken Dip

Category: Appetizer Number of People Served: 12

Ingredients:

Add a New Ingredient

Enter a New Recipe

After Button Click:

Recipe Builder

Buffalo Chicken Dip

Category: Appetizer Number of People Served: 12

Ingredients:

- ☐ Canned Chicken: 4 10-12 oz. cans ☐ Open, Drain and add to pan

Add a New Ingredient

Enter a New Recipe



PROMINEO TECH

Added lots of ingredients:

Before deletion of “Canned Chicken” — want to retype “instructions”:

Recipe Builder

Buffalo Chicken Dip

Category: Appetizer Number of People Served: 12

Ingredients:

- **Canned Chicken:** 4 10-12 oz. cans — Open, Drain and add to pan [Delete Ingredient](#)
- **Cream Cheese:** 2 blocks — Add to pan [Delete Ingredient](#)
- **Sour Cream:** 1 cup — Add to pan [Delete Ingredient](#)
- **White Vinegar:** 1 tsp — (Add only if you use Frank's Original Hot Sauce) [Delete Ingredient](#)
- **Frank's Buffalo Hot Sauce:** 8 Tbsp — (up to 1/2 cup if desired) Add to pan [Delete Ingredient](#)

Add a New Ingredient

[Add Ingredient](#)

[Delete Recipe](#)

After Delete Ingredient Button Click:

Recipe Builder

Buffalo Chicken Dip

Category: Appetizer Number of People Served: 12

Ingredients:

- **Cream Cheese:** 2 blocks — Add to pan [Delete Ingredient](#)
- **Sour Cream:** 1 cup — Add to pan [Delete Ingredient](#)
- **White Vinegar:** 1 tsp — (Add only if you use Frank's Original Hot Sauce) [Delete Ingredient](#)
- **Frank's Buffalo Hot Sauce:** 8 Tbsp — (up to 1/2 cup if desired) Add to pan [Delete Ingredient](#)

Add a New Ingredient

[Add Ingredient](#)

[Delete Recipe](#)



PROMINEO TECH

Add Canned Chicken again:

Recipe Builder

Buffalo Chicken Dip

Category: Appetizer Number of People Served: 12

Ingredients:

- Cream Cheese: 2 blocks — Add to pan Delete Ingredient
- Sour Cream: 1 cup — Add to pan Delete Ingredient
- White Vinegar: 1 tsp — (Add only if you use Frank's Original Hot Sauce) Delete Ingredient
- Frank's Buffalo Hot Sauce: 8 Tbsp — (up to 1/2 cup if desired) Add to pan Delete Ingredient
- Canned Chicken: 4 10-12 oz cans — Open, Drain & Add Chicken to pan Delete Ingredient

Add a New Ingredient

Add Ingredient

Delete Recipe

Full recipe:

Buffalo Chicken Dip

Category: Appetizer Number of People Served: 12

Ingredients:

- Cream Cheese: 2 blocks — Add to pan Delete Ingredient
- Sour Cream: 1 cup — Add to pan Delete Ingredient
- White Vinegar: 1 tsp — (Add only if you use Frank's Original Hot Sauce) Delete Ingredient
- Frank's Buffalo Hot Sauce: 8 Tbsp — (up to 1/2 cup if desired) Add to pan Delete Ingredient
- Canned Chicken: 4 10-12 oz cans — Open, Drain & Add Chicken to pan Delete Ingredient
- Raw Vegetables -- Carrots, Celery, etc.: 4 large — Cut up into sticks for dipping! Delete Ingredient
- Tortilla Chips: 1 bag — Serve for dipping Delete Ingredient

Add a New Ingredient

Add Ingredient

crudcrud.com:

```
<  →  ↻  crudcrud.com/api/467446f2b41f4ea58e1a553fd9faaf72/recipes  ☆  ⚙  👤  ⋮

[{"_id":"6207cb8c38fd8d03e8973538","name":"Buffalo Chicken Dip","numberServed":12,"category":"Appetizer","ingredients":
[{"name":"Cream Cheese","quantity":2,"units":"blocks","instructions":"Add to pan"},"name":"Sour
Cream","quantity":1,"units":"cup","instructions":"Add to pan"},"name":"White Vinegar","quantity":1,"units":"tsp","instructions":"
(Add only if you use Frank's Original Hot Sauce)"},"name":"Frank's Buffalo Hot Sauce","quantity":8,"units":"Tbsp","instructions":"
(up to 1/2 cup if desired) Add to pan"},"name":"Canned Chicken","quantity":4,"units":"10-12 oz cans","instructions":"Open, Drain &
Add Chicken to pan"},"name":"Raw Vegetables -- Carrots, Celery, etc.","quantity":4,"units":"large","instructions":"Cut up into
sticks for dipping!"},"name":"Tortilla Chips","quantity":1,"units":"bag","instructions":"Serve for dipping"}]]
```



PROMINEO TECH

Add a new recipe — Entered information:

Before Click:

Enter a New Recipe

Mulled Wine

24

Drinks

Add Recipe

After button click:

Mulled Wine

Category: Drinks Number of People Served: 24

Ingredients:

Add a New Ingredient

Name

How many -- #

Tbsp, tsp, lb, oz, etc.

Any instructions?

Add Ingredient

Delete Recipe

Enter a New Recipe

Recipe Name

Number of Servings

Main, App, Dessert...

Add Recipe

crudcrud.com — proof that it
stored through the api:

```
https://crudcrud.com/api/4674...
crudcrud.com/api/467446f2b41f4ea58e1a553fd9faaf72/recipes

[{"_id": "6207cb8c38fd8d03e8973538", "name": "Buffalo Chicken Dip", "numberServed": 12, "category": "Appetizer", "ingredients": [{"name": "Cream Cheese", "quantity": 2, "units": "blocks", "instructions": "Add to pan"}, {"name": "Sour Cream", "quantity": 1, "units": "cup", "instructions": "Add to pan"}, {"name": "White Vinegar", "quantity": 1, "units": "tsp", "instructions": "(Add only if you use Frank's Original Hot Sauce)"}, {"name": "Frank's Buffalo Hot Sauce", "quantity": 8, "units": "Tbsp", "instructions": "(up to 1/2 cup if desired) Add to pan"}, {"name": "Canned Chicken", "quantity": 4, "units": "10-12 oz cans", "instructions": "Open, Drain & Add Chicken to pan"}, {"name": "Raw Vegetables -- Carrots, Celery, etc.", "quantity": 4, "units": "large", "instructions": "Cut up into sticks for dipping!"}, {"name": "Tortilla Chips", "quantity": 1, "units": "bag", "instructions": "Serve for dipping"}]}, {"_id": "6207d16738fd8d03e897353f", "name": "Mulled Wine", "numberServed": 24, "category": "Drinks", "ingredients": []}]
```



PROMINEO TECH

Delete a recipe:

After “Delete Recipe” click:

Recipe Builder

Buffalo Chicken Dip

Category: Appetizer Number of People Served: 12

Ingredients:

- Cream Cheese: 2 blocks — Add to pan **Delete Ingredient**
- Sour Cream: 1 cup — Add to pan **Delete Ingredient**
- White Vinegar: 1 tsp — (Add only if you use Frank's Original Hot Sauce) **Delete Ingredient**
- Frank's Buffalo Hot Sauce: 8 Tbsp — (up to 1/2 cup if desired) Add to pan **Delete Ingredient**
- Canned Chicken: 4 10-12 oz cans — Open, Drain & Add Chicken to pan **Delete Ingredient**
- Raw Vegetables -- Carrots, Celery, etc.: 4 large — Cut up into sticks for dipping! **Delete Ingredient**
- Tortilla Chips: 1 bag — Serve for dipping **Delete Ingredient**

Add a New Ingredient

Name How many -- # Tbsp, tsp, lb, oz, etc.

Any instructions? **Add Ingredient**

Delete Recipe

Enter a New Recipe

Recipe Name Number of Servings Main, App, Dessert... **Add Recipe**

crudcrud.com proof that it's deleted



```
[{"_id":"6207cb8c38fd8d03e8973538","name":"Buffalo Chicken Dip","numberServed":12,"category":"Appetizer","ingredients":[{"name":"Cream Cheese","quantity":2,"units":"blocks","instructions":"Add to pan"}, {"name":"Sour Cream","quantity":1,"units":"cup","instructions":"Add to pan"}, {"name":"White Vinegar","quantity":1,"units":"tsp","instructions":"(Add only if you use Frank's Original Hot Sauce)"}, {"name":"Frank's Buffalo Hot Sauce","quantity":8,"units":"Tbsp","instructions":"(up to 1/2 cup if desired) Add to pan"}, {"name":"Canned Chicken","quantity":4,"units":"10-12 oz cans","instructions":"Open, Drain & Add Chicken to pan"}, {"name":"Raw Vegetables -- Carrots, Celery, etc.","quantity":4,"units":"large","instructions":"Cut up into sticks for dipping!"}, {"name":"Tortilla Chips","quantity":1,"units":"bag","instructions":"Serve for dipping"}]}
```

URL to GitHub Repository: <https://github.com/sw-dev-lisa-s-nh/React-Week3>