COP 4808-001: Full-Stack Web Development

Lab3

Due: Feb 9 at 11:59pm

GitHub Username: ccabreragarc

GitHub Repo URL:

#### **Step 0: Set up**

In this step, we'll create a new React project, install required dependencies, and remove the extra code we won't need for this lab.

* Create a new React project by running the command npm create vite@latest barista-app
* Follow the prompts and instructions the Terminal gives you. This project will be with the React framework in JavaScript.
* Then in your terminal, run the following commands:

cd barista-app  
npm install

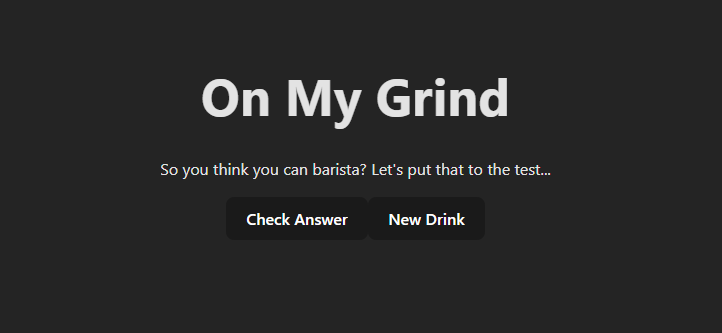
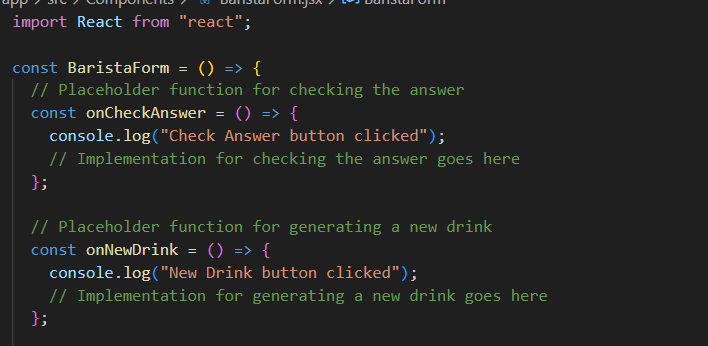
npm run dev



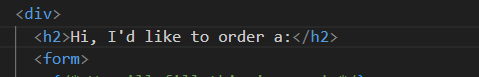
#### **Step 1: Create our React form structure**

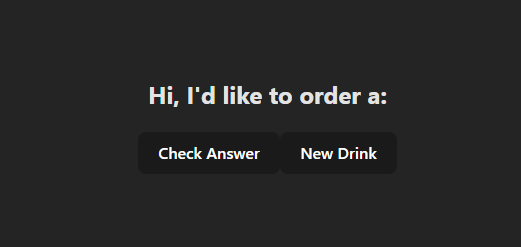
In this step, we will be creating our React form, which will allow users to input their guesses for the drink ingredients. We want to dynamically create this form.

* **Set Up the Component Structure:** In the src folder, create a Components folder and inside it, create BaristaForm.jsx. This will house the form for users to input their guesses for drink ingredients.
* **Import BaristaForm in App.jsx:** At the top of the App.jsx file, import the BaristaForm component. Then, within the return statement, add a div with a class name "title-container" that includes an h1 for the title "On My Grind", a paragraph about the challenge, and the BaristaForm component.
* **Initialize BaristaForm Component:** In BaristaForm.jsx, import React and set up the BaristaForm component with a basic return statement that includes a div for further content.
* **Add Form Elements:** Inside the div in the BaristaForm return statement, add form tags and two buttons - one for submitting the answer ("Check Answer") and another for generating a new drink ("New Drink"). Both buttons should be set up with their respective event handlers (to be added later).



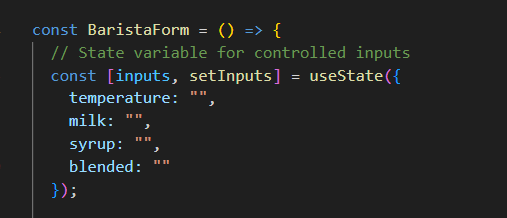
* **Add Title and Placeholder Functions:** Before the form tags, add an h2 element as the title for the form. Then, define empty arrow functions for onNewDrink() and onCheckAnswer() which will be linked to the buttons but do not contain functionality yet.



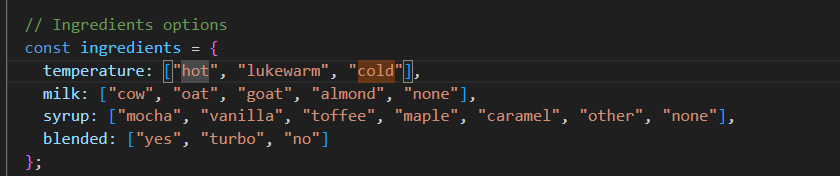


#### **Step 2: Dynamically create our form inputs with our different ingredient choices and place the user inputs into state variables**

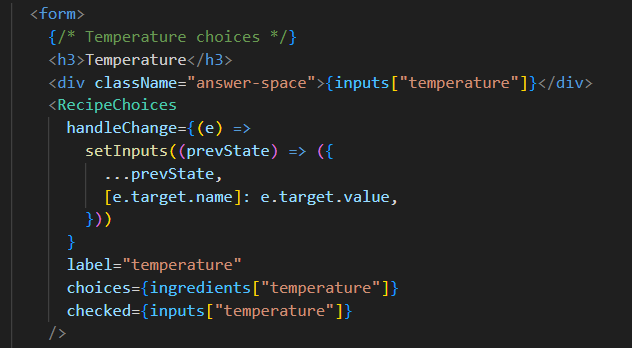
* **Implement State for User Choices:** Right after the BaristaForm component is defined, use useState to create a state variable named inputs to manage the user's selections for four basic ingredient categories: temperature, milk, syrup, and blended.



* **Define Ingredient Options:** Create a const ingredients variable below the state definition in BaristaForm.jsx to store arrays of choices for temperature, milk, syrup, and blended ingredients. Include some options as fakeouts not present in the actual drinks list to add complexity.



* **Create RecipeChoices Component:** In the Components folder, create a new file called RecipeChoices.jsx and set up a basic React component structure. This component will be responsible for displaying radio button inputs for each ingredient category.
* P**ass Props to RecipeChoices:** Modify the RecipeChoices component to accept four props: handleChange (for updating the parent component's state when a choice is made), label (describing the ingredient category), choices (the list of options for that category), and checked (to manage which option is currently selected). Use .map() within the component to render a radio button for each choice
* **Integrate RecipeChoices in BaristaForm:** In BaristaForm.jsx, for each ingredient category (temperature, milk, syrup, and blended), add a section that includes a title for the ingredient (using <h3> tags), a div to display the user's current selection (className="answer-space"), and an instance of the RecipeChoices component passing the relevant props for that category. This setup allows users to select their preferences via radio buttons, with the selections being reflected in the BaristaForm component's state.





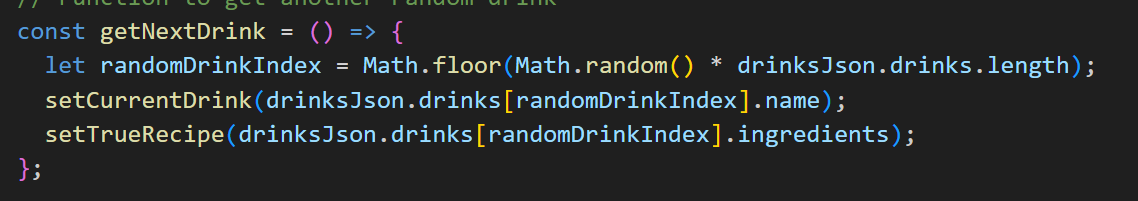
#### **Step 3: Importing our JSON drinks list to the form and adding the ability to grab new drinks from the list with the 'New Drink' button**

First, we need to import our file with our different drinks and their ingredients.

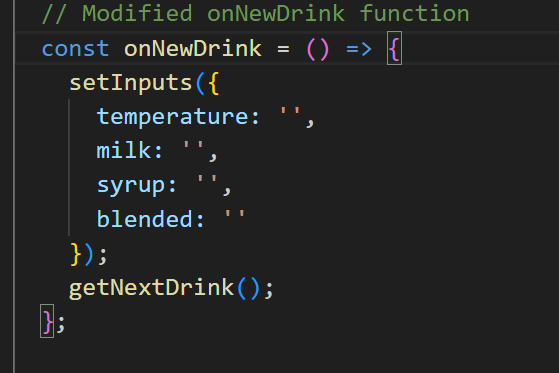
* At the top of BaristaForm.jsx, import the drinks.json file.



* Create state variables to keep track of what drink we have currently and what the true recipe is behind that drink. How can we create them using useState()?
* First, create a new function to select another random drink called getNextDrink().
* Within this function, we'll use Math to get a random number from 0 to the numerical length of the drinks list in our drinksJson.
* Set the currentDrink state variable to the name at the random index of the drinks list in our drinksJson.
* Set the trueRecipe state variable to the ingredients associated with that drink at the random index of the drinks list in our drinksJson.



* Back in our onNewDrink() function, we want to set our state variable for our ingredients to be empty, and then call getNextDrink()

;

The last thing we want to do now is render our currentDrink onto our page so that users can actually see it.

* Create a small div right under the "Hi, I'd like to order a:" title to do this with the class name "drink-container". Inside this div, add the following elements:
  + An h2 with the class name "mini-header" with the currentDrink state variable as its text.
  + A button with the type "new-drink-button" and class names "button" and "newdrink" that calls the onNewDrink() function when the button is clicked. Set the text for the button to "🔄"

