**Document for Script.js**

In this file we have added frontend javascript.

Frontend javascript means the js which is only used in frontend.

1. **let addIngredientsBtn = document.getElementById('addIngredientsBtn'):** This line of code selects the element with id ‘addIngredientsBtn’ and stores the result in variable ‘addIngredientsBtn’.
2. **let ingredientList = document.querySelector('.ingredientList'):** This line selects the first element with class ‘.ingredientList’ and stores the result in the variable ‘ingredientList’.
3. **let ingredientDiv = document.querySelectorAll('.ingredientDiv')[0]:** This line select the element with class ‘.ingredientDiv’ but [0] represents that it will select only the first element having this class and store it in variable ‘ingredientDiv’.
4. **addIngredientsBtn.addEventListener('click', function(){**

**let newIngredients = ingredientDiv.cloneNode(true);**

**let input = newIngredients.getElementsByTagName('input')[0];**

**input.value =' ';**

**ingredientList.appendChild(newIngredients);**

**}):**

In these lines of code firstly we adding a click event listener to ‘addIngredientsBtn’.

What we are doing is, we are cloning the node which is stored in ‘ingredientDiv’ variable using predefined js function ‘cloneNode(true)’. Now this method copies the node and ‘true’ represents that we are slowing for deep copy.

**What is deep copy?**

Deep copy means the clone of node will be created with the child nodes of that node and also if that node had some input element and text associated with it will copy those too.

Here that node is the div with the class ‘ingredientDiv’ which is stored in ‘ingredientDiv’ variable.

We are storing the cloned div in ‘newIngredients’ variable.

**let input = newIngredients.getElementsByTagName('input')[0]:** Here, we are finding the input tag which is associated with newIngredients means the cloned div and [0] means we only need the first input tag. In this way it becomes very easy for us to manipulate that input tag.

So, we are storing the input tag received in ‘input’ variable.

**input.value =' ':** Here, we are setting the input to be empty so that the cloned input box becomes empty.

‘value’ represents the value attribute of input tag because anything we type inside input is stored inside the value attribute so that we can access it and manipulate it.

If we don’t use this then the new cloned input will by default have the value of first input box.

**ingredientList.appendChild(newIngredients):** In this line of code, we are appending the cloned node which is stored in variable ‘newIngredients’ to the div with class ‘.ingredientList’ which is stored in variable ‘ingredientList’.

**IN THIS WAY WE ARE ABLE TO ADD FUNCTIONALITY TO (+ Ingredient) BUTTON.**