Autocomplete with JavaScript

Provide suggestions from input field - Autocomplete on form field input

1. The HTML code defines an input field with an ID of inputField and a container div with an ID of listContainer.
2. The JavaScript code defines a pre-populated array of values called prePopulatedArray.
3. It then gets the input field and list container using the document.getElementById() method and assigns them to variables inputField and listContainer, respectively.
4. An event listener is added to the input field using the addEventListener() method. Whenever the user types something in the input field, this event listener is triggered.
5. Inside the event listener, the list container is cleared using the innerHTML property.
6. The current input value is obtained using the value property of the input field and converted to lowercase using the toLowerCase() method.
7. The pre-populated array is filtered based on the input value using the filter() method. This method creates a new array with all elements that pass the test implemented by the provided function. In this case, the test is whether the lowercase version of each item in the array starts with the current input value.
8. For each item in the filtered array, a new li element is created using the document.createElement() method and added to the list container using the appendChild() method.
9. Finally, the textContent property of the new li has the values from the array listed that match the results.

ap

- Apple

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
    <input id="inputField">
    <div id="listContainer"></div>
    <script>
        // Define the pre-populated array of values
        const prePopulatedArray = ["Apple", "Banana",
"Cherry", "Date", "Elderberry", "Fig", "Grape"];

        // Get the input field and list container
        const inputField =
document.getElementById("inputField");
        const listContainer =
document.getElementById("listContainer");

        // Add an event listener to the input field
        inputField.addEventListener("input", function
() {
            // Clear the list container
            listContainer.innerHTML = "";

            // Get the current input value
            const inputValue =
inputField.value.toLowerCase();
```

```
            // Filter the pre-populated array based on
the input value
            const filteredArray =
prePopulatedArray.filter(function (item) {
                return
item.toLowerCase().startsWith(inputValue);
            });

            // Add each filtered item to the list
container
            filteredArray.forEach(function (item) {
                const listItem =
document.createElement("li");
                listItem.textContent = item;
                listContainer.appendChild(listItem);
            });
        });
    </script>
</body>
</html>
```

- <!DOCTYPE html> is a document type declaration that specifies the version of HTML used in the document.
- <html> is the root element of an HTML document.
- <head> is the element that contains metadata about the document, such as the title and links to external stylesheets and scripts.
- <body> is the element that contains the visible content of the document.
- <input id="inputField"> is an HTML input element where the user can type in their search query.

- is an HTML div element where the filtered list of values will be displayed.
- The script section contains JavaScript code that defines the search and filter functionality.
- const prePopulatedArray = ["Apple", "Banana", "Cherry", "Date", "Elderberry", "Fig", "Grape"]; defines a pre-populated array of values that the user can select from.
- const inputField = document.getElementById("inputField"); gets a reference to the input field element using its id.
- const listContainer = document.getElementById("listContainer"); gets a reference to the list container element using its id.
- inputField.addEventListener("input", function () { ... }); adds an event listener to the input field that listens for changes in its value.
- listContainer.innerHTML = ""; clears the list container before adding the filtered items.
- const inputValue = inputField.value.toLowerCase(); gets the current value of the input field and converts it to lowercase for case-insensitive comparison.
- const filteredArray = prePopulatedArray.filter(function (item) { ... }); filters the pre-populated array based on whether each item starts with the input value.
- filteredArray.forEach(function (item) { ... }); loops through each filtered item and creates a list item element for it.
- listItem.textContent = item; sets the text content of the list item to the filtered item.

Input field letter match for list items

a

- Apple
- Banana
- Orange
- Mango
- Pineapple
- Peach

Here's an example of how you can implement a search and filter functionality in JavaScript to enable users to quickly find and select from a pre-populated list of values:

HTML:

```
<input type="text" id="myInput" onkeyup="myFunction()"
placeholder="Search for names..">
<ul id="myUL">
  <li><a href="#">Apple</a></li>
  <li><a href="#">Banana</a></li>
  <li><a href="#">Orange</a></li>
  <li><a href="#">Mango</a></li>
  <li><a href="#">Lemon</a></li>
  <li><a href="#">Pineapple</a></li>
  <li><a href="#">Kiwi</a></li>
  <li><a href="#">Peach</a></li>
</ul>
```

JavaScript:

```
function myFunction() {
  // Declare variables
  var input, filter, ul, li, a, i, txtValue;
```

```
  input = document.getElementById('myInput');
  filter = input.value.toUpperCase();
  ul = document.getElementById('myUL');
  li = ul.getElementsByTagName('li');

  // Loop through all list items, and hide those who
don't match the search query
  for (i = 0; i < li.length; i++) {
    a = li[i].getElementsByTagName('a')[0];
    txtValue = a.textContent || a.innerText;
    if (txtValue.toUpperCase().indexOf(filter) > -1) {
      li[i].style.display = '';
    } else {
      li[i].style.display = 'none';
    }
  }
}
```

Explanation:

The HTML code defines an input field where the user can type in their search query, and a list of pre-populated values that the user can select from.
The JavaScript code defines a function myFunction() that gets called whenever the user types in the input field.
Within the function, we first get the user's search query and convert it to uppercase so that the search is case-insensitive.
We then loop through all the list items and check if their text content contains the search query. If it does, we display the list item; otherwise, we hide it.
The style.display property is used to show or hide the list items, depending on whether they match the search query or not.

Note that this is just a basic example and can be extended and customized based on your specific use case.