```
<!DOCTYPE html>

<html>
<head>
    <meta charset="utf-8">
    <title>Alphabet Concordance (arrays version)</title>
    <style type="text/css">
        /* css rules go here. */
        #output { visibility: hidden; }
        #text { height: 100px; width: 800px; overflow: scroll; font-size: larger; }
    </style>
    <script type="text/javascript">
        /* scripts go here */
        /* variables, constants, etc. */
        const alphabet="abcdefghijklmnopqrstuvwxyz ,.";
        var charCountArray=[];
        //________ends variable, constant decls.

        /* Event handlers, etc. */

        /*
         * This is the function called by the onload event
         * handler. Essentially, it reconstructs the
         * concordance table.
         */
        function initTable() {
            charCountArray = Array ( alphabet.length );
            for( var index=0; index < charCountArray.length; index++ ) {
                charCountArray[ index ] = 0 ;
            }
        }
        /*
         * called between input sessions ...
         */
        function resetTable () {
            for( var index=0; index < charCountArray.length; index++ ) {
                charCountArray[ index ]=0;
            }
        }
        // _________end general functions, event handlers, etc.

        /* String handling functions, procedures ...
         * stripping punctuation, lowercasing text, ... .
         */
        function parseStream ( text ) {
            var parsedText = (text.trim()).toLowerCase();
            /* brute force approach ... doesn't use RegEx. */
            return parsedText;
        }

        /*
         * Find the corresponding index of the character (if it's in the
         * alphabet), and set up that array value by one.
         */
        function analyzeText( text ) {
            var stream = parseStream( text );
            for( var charIndex=0; charIndex < stream.length; charIndex++ ) {
                /* for each alphabetic character, find the CharInt Pair
                 * entry in the array, and update the corresponding
                 * counter (see the object/class definition of
                 * CharIntPair, above)
                 */
                var index = alphabet.indexOf( stream.charAt( charIndex ) ); //findIndex( stream.charAt( charIndex ));
                if( index >= 0 ) {
                    charCountArray[ index ]++;
                }
            }
        }

        function charCountArrayToHTML( index ) {
```

```
71              return "<tr><td>" + alphabet.charAt( index ) + "</td><td>" + charCountArray[ index ] + "</td></tr>";
72          }
73          /*
74           * Event handler: onclick!
75           */
76          function doConcordance() {
77              var text = document.getElementById("text").value;
78              analyzeText( text );
79              updateTable();
80          }
81
82          function updateTable() {
83              var tableBody = "";
84              for( var index=0; index < charCountArray.length; index++ ) {
85                  tableBody += charCountArrayToHTML( index );
86              }
87              document.getElementById("entries").innerHTML=tableBody;
88              document.getElementById("output").style.visibility="visible";
89          }
90
91          function resetAll() {
92              document.getElementById("text").value="";
93              resetTable();
94              document.getElementById("output").style.visibility="hidden";
95
96          }
97      </script>
98  </head>
99
100 <body onload="initTable();">
101     <h1>Counting Strings</h1>
102     <p>A <dfn>concordance</dfn> is a mapping from objects, usually strings (words), to the number of times that each appears
     in a collection (usually a document).
103     For this exercise, you will enter text in the text box below, and write some scripts that construct a table that shows h
    ow many times each character from the
104     English alphabet appeared in that text. Note, many characters may appear 0 times. See if you can find a "pattern" whereb
    y certain characters appear
105     more often than others.</p>
106     <br>
107     <div id="input">
108         <textarea id="text" required autofocus>
109
110         </textarea>
111     </div>
112     <br>
113     <button id="checkIt" onclick="doConcordance();">Build Concordance Table</button> <button id="reset" onclick="resetA
    ll();">Reset and enter new text</button>
114 <br>
115 <div id="output">
116     <table summary="concordance">
117         <thead><tr><th>Character</th><th>Number of Occurrences</th></tr></thead>
118         <tbody id="entries">
119             <!-- entries go here as a row -->
120         </tbody>
121     </table>
122 </div>
123
124 </body>
125 </html>
```