

Documentation for Financial Transactions HTML Page

Contents

1	HTML	1
1.1	Preamble and head	1
1.2	Input Fields	1
1.3	Table	1
2	Javascript	2
2.1	getData()	2
2.2	validate()	2
2.3	generateId()	2
2.4	calculateCostBasis()	2
2.5	addTransaction()	2
2.6	deleteRow()	2
2.7	editRow()	2
2.8	saveChanges()	2
2.9	discardChanges()	2
2.10	sortTable()	2
3	CSS	3
3.1	Vertical Scrolling Table	3
3.2	Horizontal Scrolling on Overflow	3
3.3	Miscellaneous	3
A	HTML Source	4
B	Javascript Source	8
C	CSS Source	13

1 HTML

1.1 Preamble and head

```
1 <!DOCTYPE html>
```

This line declares that the document is an HTML5 document.

```
2 <head>
3   <meta charset = "UTF-8"/>
4   <link rel="stylesheet" type="text/css" href="./style.css"/>
5   <script src="./script.js"></script>
6 </head>
```

1.2 Input Fields

1.3 Table

`<head>` tags are used to contain meta information about the document. Within the `head` element:

- The first line defines the character set of the document.
- The second line defines the source of an external CSS document.
- The third line defines the source of an external Javascript document.

2 Javascript

2.1 `getData()`

2.2 `validate()`

2.3 `generateId()`

2.4 `calculateCostBasis()`

2.5 `addTransaction()`

2.6 `deleteRow()`

2.7 `editRow()`

2.8 `saveChanges()`

2.9 `discardChanges()`

2.10 `sortTable()`

3 CSS

3.1 Vertical Scrolling Table

3.2 Horizontal Scrolling on Overflow

3.3 Miscellaneous

A HTML Source

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset = "UTF-8"/>
5     <link rel="stylesheet" type="text/css" href="./style.css"/>
6     <script src="./script.js"></script>
7   </head>
8   <body>
9     <article id="inputFields">
10       <form onsubmit="return false" autocomplete="off">
11         <section>
12           <label for="date">Date:</label><br/>
13           <input id="date" name="date" type="date" placeholder="
14             ↪ Date" required/>
15         </section>
16
17         <section>
18           <label for="account">Account Number:</label><br/>
19           <input id="account" name="account" type="text"
20             ↪ placeholder="Account Number" required/>
21         </section>
22
23         <section>
24           <label for="type">Transaction Type:</label><br/>
25           <select id="type" name="type">
26             <option value=""></option>
27             <option value="BUY">BUY</option>
28             <option value="SELL">SELL</option>
29             <option value="DIVIDEND">DIVIDEND</option>
30             <option value="INTEREST">INTEREST</option>
31             <option value="WITHDRAW">WITHDRAW</option>
32             <option value="DEPOSIT">DEPOSIT</option>
33           </select>
34         </section>
35
36         <section>
37           <label for="security">Security:</label><br/>
38           <input id="security" name="security" type="text"
39             ↪ placeholder="Security" required/>
40         </section>
41
42         <section>
43           <label for="amount">Amount:</label><br/>
44           <input id="amount" name="amount" type="text"
45             ↪ placeholder="Unit Amount" required/>
46         </section>
47
48         <section>
49           <label for="dAmount">$ Amount:</label><br/>
50           <input id="dAmount" name="dAmount" type="text"
51             ↪ placeholder="$ Amount" required/>
52         </section>
53       </form>
54     </article>
55   </body>
56 </html>
```

```

48         <section>
49             <button id="add" type="submit" onclick="
50                 ↪ addTransactionButton();">Add Transaction</button>
51                 ↪ >
52             <button id="save" type="submit" hidden="true" onclick
53                 ↪ ="saveChanges();">Save</button>
54             <button id="discard" type="button" hidden="true"
55                 ↪ onclick="discardChanges();">Discard</button>
56         </section>
57     <form>
58 </article>
59
60 <article id="table">
61     <table>
62         <thead>
63             <tr>
64                 <th>
65                     <section>
66                         Transaction ID
67                     </section>
68                     <section class="sort">
69                         <button type="button" onclick="sortTable(0,
70                             ↪ true)">^</button>
71                         <button type="button" onclick="sortTable(0,
72                             ↪ false)">v</button>
73                     </section>
74                 </th>
75                 <th>
76                     <section>
77                         Date
78                     </section>
79                     <section class="sort">
80                         <button type="button" onclick="sortTable(1,
81                             ↪ true)">^</button>
82                         <button type="button" onclick="sortTable(1,
83                             ↪ false)">v</button>
84                     </section>
85                 </th>
86                 <th>
87                     <section>
88                         Account Number
89                     </section>
90                     <section class="sort">
91                         <button type="button" onclick="sortTable(2,

```

```

92         <section class="sort">
93             <button type=button onclick="sortTable(3,
94                 ↪ true)">^</button>
95             <button type=button onclick="sortTable(3,
96                 ↪ false)">v</button>
97         </section>
98     </th>
99     <th>
100         <section>
101             Security
102         </section>
103         <section class="sort">
104             <button type=button onclick="sortTable(4,
105                 ↪ true)">^</button>
106             <button type=button onclick="sortTable(4,
107                 ↪ false)">v</button>
108         </section>
109     </th>
110     <th>
111         <section>
112             Amount
113         </section>
114         <section class="sort">
115             <button type=button onclick="sortTable(5,
116                 ↪ true)">^</button>
117             <button type=button onclick="sortTable(5,
118                 ↪ false)">v</button>
119         </section>
120     </th>
121     <th>
122         <section>
123             $ Amount
124         </section>
125         <section class="sort">
126             <button type=button onclick="sortTable(6,
127                 ↪ true)">^</button>
128             <button type=button onclick="sortTable(6,
129                 ↪ false)">v</button>
130         </section>
131     </th>
132     <th>
133         <section>
134             Cost Basis
135         </section>
136         <section class="sort">
137             <button type=button onclick="sortTable(7,
138                 ↪ true)">^</button>
139             <button type=button onclick="sortTable(7,
140                 ↪ false)">v</button>
141         </section>
142     </th>
143     <th>Actions</th>
144 </tr>
145 </thead>

```



```
136         <tbody id="tableBody">
137             </tbody>
138         </table>
139     </article>
140 </body>
141 </html>
```

B Javascript Source

```
1 function getData() {
2     var date = document.getElementById("date").value;
3     var account = document.getElementById("account").value;
4     var type = document.getElementById("type").value;
5     var security = document.getElementById("security").value;
6     var amount = document.getElementById("amount").value;
7     var dAmount = document.getElementById("dAmount").value;
8
9     amount = Number(amount);
10
11     if(dAmount[0] == '$') {
12         dAmount = dAmount.substr(1);
13     }
14     dAmount = Number(dAmount);
15
16     if(validate(date, account, type, security, amount, dAmount)) {
17         var costBasis = calculateCostBasis(amount, dAmount);
18         dAmount = '$' + dAmount.toFixed(2);
19
20         return [ date, account, type, security, amount, dAmount, costBasis
21                 ↪ ];
22     }
23     else return false;
24 }
25
26 function validate(date, account, type, security, amount, dAmount) {
27     if(!validateDate(date)) return false;
28     if(!validateAccount(account)) return false;
29     if(!validateType(type)) return false;
30     if(!validateSecurity(security)) return false;
31     if(!validateAmount(amount)) return false;
32     if(!validateDAmount(dAmount)) return false;
33
34     return true;
35 }
36
37 function validateDate(date) {
38     realDate = new Date();
39     inputDate = document.getElementById('date').valueAsNumber;
40
41     if(isNaN(inputDate)) {
42         alert('Error: No date specified');
43         return false;
44     }
45
46     if(realDate.valueOf() < inputDate) {
47         alert('Error: Date is in the future');
48         return false;
49     }
50
51     return true;
52 }
```

```

52
53 function validateAccount(account) {
54     if(account == '') {
55         alert('Error: Missing Account Number');
56         return false;
57     }
58
59     return true;
60 }
61
62 function validateType(type) {
63     if(type == '') {
64         alert('Error: Missing Transaction Type');
65         return false;
66     }
67
68     return true;
69 }
70
71 function validateSecurity(security) {
72     if(security == '') {
73         alert('Error: Missing Security');
74         return false;
75     }
76
77     return true;
78 }
79
80 function validateAmount(amount) {
81     if(amount == '') {
82         alert('Error: Missing Amount');
83         return false;
84     }
85
86     if(isNaN(amount)) {
87         alert('Error: Invalid Amount');
88         return false;
89     }
90
91     return true;
92 }
93
94 function validateDAmount(dAmount) {
95     if(dAmount == '') {
96         alert('Error: Missing $ Amount');
97         return false;
98     }
99
100     if(isNaN(dAmount)) {
101         alert('Error: Invalid $ Amount');
102         return false;
103     }
104
105     return true;

```

```

106 }
107
108 function generateId() {
109     var id = '';
110     var idLength = 6;
111
112     var characters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789';
113     var charactersLength = characters.length;
114
115     var unique = false;
116
117     while(!unique) {
118         for(var i = 0; i < idLength; i++) {
119             id += characters.charAt(Math.floor(Math.random() *
120                 ↪ charactersLength));
121         }
122
123         unique = true;
124         for(var i = 0; i < document.getElementsByClassName('idCell').
125             ↪ length; i++) {
126             if(document.getElementsByClassName('idCell')[i].innerText ==
127                 ↪ id) {
128                 unique = false;
129                 break;
130             }
131         }
132     }
133     return id;
134 }
135
136 function calculateCostBasis(amount, dAmount) {
137     costBasis = '$' + (dAmount / amount).toFixed(2);
138     return costBasis;
139 }
140
141 function addTransaction(id, date, account, type, security, amount, dAmount
142     ↪ , costBasis) {
143     var tableBody = document.getElementById('tableBody');
144     var newRow = tableBody.insertRow(0);
145     newRow.classList += "bodyRow";
146
147     var actionsContent = "<button type='button' onclick='editRow(this)'"
148         ↪ "Edit</button> <button type='button' onclick='deleteRow(this)'"
149         ↪ "Delete</button>";
150     var rowContents = [id, date, account, type, security, amount, dAmount,
151         ↪ costBasis, actionsContent];
152
153     for(var i = 0; i < rowContents.length; i++) {
154         var newCell = newRow.insertCell(i);
155         newCell.innerHTML = rowContents[i];
156         if(i == 0) {
157             newCell.classList += "idCell";
158         }
159     }
160 }

```

```

153 }
154
155 function addTransactionButton() {
156     var data = getData();
157     if(data) {
158         var date = data[0];
159         var account = data[1];
160         var type = data[2];
161         var security = data[3];
162         var amount = data[4];
163         var dAmount = data[5];
164         var costBasis = data[6];
165
166         var id = generateId();
167
168         addTransaction(id, date, account, type, security, amount, dAmount,
            ↪ costBasis);
169     }
170 }
171
172 function deleteRow(button) {
173     var row = button.parentElement.parentElement;
174     document.getElementById("tableBody").removeChild(row);
175
176     if(document.getElementsByClassName('editing').length == 0) {
177         document.getElementById('add').removeAttribute('hidden');
178         document.getElementById('save').setAttribute('hidden', true);
179         document.getElementById('discard').setAttribute('hidden', true);
180     }
181 }
182
183 function editRow(button) {
184     if(document.getElementsByClassName('editing').length > 0)
185         document.getElementsByClassName('editing')[0].classList = "bodyRow
            ↪ ";
186
187     var row = button.parentElement.parentElement;
188     var rowContent = row.getElementsByTagName('td');
189     row.classList = "bodyRow editing";
190
191     document.getElementById('date').value = rowContent[1].innerText;
192     document.getElementById('account').value = rowContent[2].innerText;
193     document.getElementById('type').value = rowContent[3].innerText;
194     document.getElementById('security').value = rowContent[4].innerText;
195     document.getElementById('amount').value = rowContent[5].innerText;
196     document.getElementById('dAmount').value = rowContent[6].innerText;
197
198     document.getElementById('add').setAttribute('hidden', true);
199     document.getElementById('save').removeAttribute('hidden');
200     document.getElementById('discard').removeAttribute('hidden');
201 }
202
203 function saveChanges() {
204     data = getData();

```

```

205     if(data) {
206         rowToEdit = document.getElementsByClassName('editing')[0];
207         cellsToEdit = rowToEdit.getElementsByTagName('td');
208
209         for(var i = 0; i < data.length; i++) {
210             cellsToEdit[i + 1].innerHTML = data[i];
211         }
212         rowToEdit.classList = "bodyRow";
213     }
214
215     document.getElementById('add').removeAttribute('hidden');
216     document.getElementById('save').setAttribute('hidden', true);
217     document.getElementById('discard').setAttribute('hidden', true);
218 }
219
220 function discardChanges() {
221     document.getElementsByClassName('editing')[0].classList = "bodyRow";
222
223     document.getElementById('add').removeAttribute('hidden');
224     document.getElementById('save').setAttribute('hidden', true);
225     document.getElementById('discard').setAttribute('hidden', true);
226 }
227
228 function sortTable(column, ascending) {
229     var tableBody = document.getElementById('tableBody');
230     var rows = document.getElementsByClassName('bodyRow');
231
232     var sorting = true;
233     while(sorting) {
234         sorting = false;
235         for(var i = 0; i < (rows.length - 1); i++) {
236             rowA = rows[i].getElementsByTagName('td')[column];
237             rowB = rows[i + 1].getElementsByTagName('td')[column];
238
239             var swap = false;
240
241             if(ascending && rowA.innerHTML.toLowerCase() > rowB.innerHTML.
242                 ↪ toLowerCase()) swap = true;
243             else if(!ascending && rowA.innerHTML.toLowerCase() < rowB.
244                 ↪ innerHTML.toLowerCase()) swap = true;
245
246             if(swap) {
247                 sorting = true;
248                 rows[i].parentNode.insertBefore(rows[i + 1], rows[i]);
249             }
250         }
251     }

```

C CSS Source

```
1 body {
2     font-size: 14px;
3 }
4
5 #inputFields {
6     padding: 10px 0;
7     overflow-x: auto;
8 }
9
10 form {
11     min-width: 1900px;
12 }
13
14     form > section {
15         width: 14%;
16         display: inline-block;
17     }
18
19     input,
20     select {
21         min-width: 100px;
22         width: 80%;
23     }
24
25     button {
26         width: 40%;
27     }
28
29 #table {
30     max-height: 80vh;
31     overflow: auto;
32 }
33
34 table {
35     width: 100%;
36     margin: auto;
37     border-collapse: collapse;
38 }
39     th {
40         min-width: 200px;
41         width: 10%;
42         position: sticky;
43         background: white;
44         top: 0;
45     }
46
47     th > section {
48         width: 80%;
49         display: inline-block;
50         padding: 0;
51         margin: 0;
52     }
```

```
53
54     .sort {
55         width: 10%;
56     }
57
58     .sort > button {
59         padding: 0;
60         border: 0;
61         display: block;
62         width: 100%;
63     }
64
65     .editing {
66         background-color: yellow;
67     }
68
69     #table,
70     table,
71     td,
72     th {
73         box-shadow: 1px 1px black, inset 1px 1px black;
74     }
```