

Documentation for Financial Transactions HTML Page
Jason N.
June 1, 2020

Contents

1	HTML	1
1.1	Preamble and head	1
1.1.1	Preamble	1
1.1.2	meta charset	1
1.1.3	link rel="stylesheet"	1
1.1.4	Scripts	1
1.2	Inputs	1
1.2.1	Labels	2
1.2.2	Date	2
1.2.3	Text	2
1.2.4	List	2
1.2.5	File	2
1.2.6	Buttons	3
1.3	Filters	3
1.3.1	Tooltip	3
1.3.2	Checkbox	3
1.4	Options	3
1.5	Table	3
1.5.1	thead	3
1.5.2	tbody	3
2	Main Javascript	4
2.1	getData()	4
2.2	validate()	4
2.2.1	Check empty	4
2.2.2	Check NaN	4
2.2.3	Check date	4
2.3	generateId()	4
2.4	calculateCostBasis()	4
2.5	addTransaction()	4
2.6	deleteRow()	4
2.7	editRow()	4
2.8	saveChanges()	4
2.9	discardChanges()	4
2.10	sortTable()	4
3	Secondary Javascript	5
3.1	firebaseScript.js	5
3.2	googleApiScript.js	5
3.3	imageFirestore.js	5
3.4	localStorageScript.js	5
3.5	mysqlScript.js	5
4	CSS	6
4.1	Vertical Scrolling Table	6
4.2	Horizontal Scrolling on Overflow	6
4.3	Miscellaneous	6
4.3.1	Sort buttons	6
4.3.2	Editing highlight	6
4.3.3	Table borders	6

1 HTML

1.1 Preamble and head

1.1.1 Preamble

Declares the document as HTML5.

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

1.1.2 meta charset

Specifies that characters in the file are encoded in UTF-8.

```
4 <meta charset = "UTF-8"/>
```

1.1.3 link rel="stylesheet"

Imports the CSS file.

```
5 <link rel="stylesheet" type="text/css" href="./style.css"/>
```

1.1.4 Scripts

Imports the main Javascript file, responsible for the table and UI.

```
7 <script src="./script.js"></script>
```

Imports the Google API library.

```
8 <script src="https://apis.google.com/js/api.js"></script>
```

Imports other Javascript files, responsible for database management.

```
9 <script src="./googleApiScript.js"></script>
10 <script src="./mysqlScript.js"></script>
11 <script src="./localStorageScript.js"></script>
12 <script src="./imageFirestore.js"></script>
```

1.2 Inputs

Disables autocomplete which remembers past user input by default. `return false` specifies that no POST request should be made to the server.

```
20 <form onsubmit="return false" autocomplete="off">
```

1.2.1 Labels

Identifies the purpose of the field to the user, allows the user to select the field by clicking the label. This element is also used by accessibility tools to identify the field.

```
22 <label for="date">Date:</label><br/>
```

1.2.2 Date

The `date` input type is supported by most modern browsers and provides an intuitive UI for selecting dates. It also includes methods for converting or verifying the `Date` object.

```
13 <input id="date" name="date" type="date" placeholder="yyyy-mm-dd"/>
```

1.2.3 Text

The `text` input type allows the user to input a string. For numbers, this string has to be parsed in Javascript.

```
28 <input id="account" name="account" list="accountsList" type="text"
    ↪ placeholder="Account Number"/>
```

1.2.4 List

Lists are created using the `select` element, containing `option` elements. Each option has a `value` which is used in Javascript, and `innerText` which is seen by the user.

```
34 <label for="type">Transaction Type:</label><br/>
35 <select id="type" name="type">
36   <option value=""></option>
37   <option value="BUY">BUY</option>
38   <option value="SELL">SELL</option>
39   <option value="!DIVIDEND">DIVIDEND</option>
40   <option value="!INTEREST">INTEREST</option>
41   <option value="!WITHDRAW">WITHDRAW</option>
42   <option value="!DEPOSIT">DEPOSIT</option>
43 </select>
```

1.2.5 File

Files are uploaded using the `file` input type. The `multiple` attribute allows the user to upload multiple files, which are interpreted as an array of files in Javascript.

```
64 <label id="fileUploadLabel" for="fileUpload">Upload file</label>
65 <input id="fileUpload" name="fileUpload" type="file" onchange="
    ↪ fileUploadChanged();" multiple/>
```

1.2.6 Buttons

Buttons with the `submit` type can be used to check that all required sections are complete and highlight them in red. These buttons can also be used to send a `POST` request to a server if desired. The `onclick` attribute specified the function and parameters that should be executed when pressed.

```
70 <button id="add" type="submit" onclick="addTransactionButton();">Add  
    ↪ Transaction</button>
```

1.3 Filters

Filter HTML elements are handled exactly the same as their counterparts in the input section. Some fields have two elements to handle a lower and upper bound, but these are handled solely in Javascript.

1.3.1 Tooltip

The `span` element is a generic container. The `title` attribute will display its value as a tool tip when the element is hovered.

```
118 <span title="Enter search terms here. Separate search terms with && or ||  
    ↪ for AND and OR statements, respectively. Exclusive filters are  
    ↪ marked by a leading !. Use || to filter by multiple securities (e.g.  
    ↪ SPY || TLT) and && to exclude multiple securities (e.g. !SPY && !  
    ↪ TLT). ">?</span><br/>
```

1.3.2 Checkbox

The input type `checkbox` provides a toggleable input field which can be evaluated as `true` or `false` with Javascript.

```
153 <label for="filterNa">Filter N/A:</label>  
154 <input id="filterNA" name="filterNA" type="checkbox"/>
```

1.4 Options

The options section uses buttons, text inputs, a file input, and drop down menus, which are described in the inputs section. The special handling of these elements is done in Javascript.

1.5 Table

1.5.1 `thead`

1.5.2 `tbody`

2 Main Javascript

2.1 `getData()`

2.2 `validate()`

2.2.1 Check empty

2.2.2 Check NaN

2.2.3 Check date

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 Secondary Javascript

3.1 firebaseScript.js

3.2 googleApiScript.js

3.3 imageFirestore.js

3.4 localStorageScript.js

3.5 mysqlScript.js

4 CSS

4.1 Vertical Scrolling Table

4.2 Horizontal Scrolling on Overflow

4.3 Miscellaneous

4.3.1 Sort buttons

4.3.2 Editing highlight

4.3.3 Table borders