Documentation for Financial Transactions HTML Page

Contents

1 HTML 1

1.1 Preamble and head 1

2 Javascript 2

3 CSS 3

A HTML Source 4

B Javascript Source 8

C CSS Source 13

1 HTML

1.1 Preamble and head

```
<!DOCTYPE html>
```

This line declares that the document is an HTML5 document.

```
<head>
    <meta charset = "UTF-8"/>
    <liink rel="readsheet" type="text/css" href="./style.css"/>
    <script src="./script.js"></script>
</head>
```

`<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

3 CSS

A HTML Source

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset = "UTF-8"/>
        <link rel="stylesheet" type="text/css" href="./style.css"/>
        <script src="./script.js"></script>
    </head>
    <body>
        <article id="inputFields">
            <form onsubmit="return false" autocomplete="off">
                <section>
                    <label for="date">Date:</label><br/>
                    <input id="date" name="date" type="date" placeholder="
                       ↪  Date" required/>
                </section>

                <section>
                    <label for="account">Account Number:</label><br/>
                    <input id="account" name="account" type="text"
                       ↪  placeholder="Account Number" required/>
                </section>

                <section>
                    <label for="type">Transaction Type:</label><br/>
                    <select id="type" name="type">
                        <option value=""></option>
                        <option value="BUY">BUY</option>
                        <option value="SELL">SELL</option>
                        <option value="DIVIDEND">DIVIDEND</option>
                        <option value="INTEREST">INTEREST</option>
                        <option value="WITHDRAW">WITHDRAW</option>
                        <option value="DEPOSIT">DEPOSIT</option>
                    </select>
                </section>

                <section>
                    <label for="security">Security:</label><br/>
                    <input id="security" name="security" type="text"
                       ↪  placeholder="Security" required/>
                </section>

                <section>
                    <label for="amount">Amount:</label><br/>
                    <input id="amount" name="amount" type="text"
                       ↪  placeholder="Unit Amount" required/>
                </section>

                <section>
                    <label for="dAmount">$ Amount:</label><br/>
                    <input id="dAmount" name="dAmount" type="text"
                       ↪  placeholder="$ Amount" required/>
                </section>
                <section>
                    <button id="add" type="submit" onclick="
                       ↪ addTransactionButton();">Add Transaction</button
                       ↪ >
                    <button id="save" type="submit" hidden="true" onclick
                       ↪ ="saveChanges();">Save</button>
                    <button id="discard" type="button" hidden="true"
                       ↪ onclick="discardChanges();">Discard</button>
                </section>
            <form>
        </article>

        <article id="table">
            <table>
                <thead>
                    <tr>
                        <th>
                            <section>
                                Transaction ID
                            </section>
                            <section class="sort">
                                <button type=button onclick="sortTable(0,
                                   ↪ true)">^</button>
                                <button type=button onclick="sortTable(0,
                                   ↪ false)">v</button>
                            </section>
                        </th>
                        <th>
                            <section>
                                Date
                            </section>
                            <section class="sort">
                                <button type=button onclick="sortTable(1,
                                   ↪ true)">^</button>
                                <button type=button onclick="sortTable(1,
                                   ↪ false)">v</button>
                            </section>
                        </th>
                        <th>
                            <section>
                                Account Number
                            </section>
                            <section class="sort">
                                <button type=button onclick="sortTable(2,
                                   ↪ true)">^</button>
                                <button type=button onclick="sortTable(2,
                                   ↪ false)">v</button>
                            </section>
                        </th>
                        <th>
                            <section>
                                Transaction Type
                            </section>
                            <section class="sort">
                                <button type=button onclick="sortTable(3,
                                    ↪ true)">^</button>
                                <button type=button onclick="sortTable(3,
                                    ↪ false)">v</button>
                            </section>
                        </th>
                        <th>
                            <section>
                                Security
                            </section>
                            <section class="sort">
                                <button type=button onclick="sortTable(4,
                                    ↪ true)">^</button>
                                <button type=button onclick="sortTable(4,
                                    ↪ false)">v</button>
                            </section>
                        </th>
                        <th>
                            <section>
                                Amount
                            </section>
                            <section class="sort">
                                <button type=button onclick="sortTable(5,
                                    ↪ true)">^</button>
                                <button type=button onclick="sortTable(5,
                                    ↪ false)">v</button>
                            </section>
                        </th>
                        <th>
                            <section>
                                $ Amount
                            </section>
                            <section class="sort">
                                <button type=button onclick="sortTable(6,
                                    ↪ true)">^</button>
                                <button type=button onclick="sortTable(6,
                                    ↪ false)">v</button>
                            </section>
                        </th>
                        <th>
                            <section>
                                Cost Basis
                            </section>
                            <section class="sort">
                                <button type=button onclick="sortTable(7,
                                    ↪ true)">^</button>
                                <button type=button onclick="sortTable(7,
                                    ↪ false)">v</button>
                            </section>
                        </th>
                        <th>Actions</th>
                    </tr>
                </thead>
```

```
                <tbody id="tableBody">
                </tbody>
            </table>
        </article>
    </body>
</html>
```

B Javascript Source

```
function addTransactionButton() {
    var data = getData();
    if(data) {
        var date = data[0];
        var account = data[1];
        var type = data[2];
        var security = data[3];
        var amount = data[4];
        var dAmount = data[5];
        var costBasis = data[6];

        var id = generateId();

        addTransaction(id, date, account, type, security, amount, dAmount,
           ↪  costBasis);
    }
}

function addTransaction(id, date, account, type, security, amount, dAmount
   ↪ , costBasis) {
    var tableBody = document.getElementById('tableBody');
    var newRow = tableBody.insertRow(0);
    newRow.classList += "bodyRow";

    var actionsContent = "<button type='button' onclick='editRow(this)'>
       ↪ Edit</button> <button type='button' onclick='deleteRow(this)'>
       ↪ Delete</button>";
    var rowContents = [id, date, account, type, security, amount, dAmount,
       ↪  costBasis, actionsContent];

    for(var i = 0; i < rowContents.length; i++) {
        var newCell = newRow.insertCell(i);
        newCell.innerHTML = rowContents[i];
        if(i == 0) {
            newCell.classList += "idCell";
        }
    }
}

function getData() {
    var date = document.getElementById("date").value;
    var account = document.getElementById("account").value;
    var type = document.getElementById("type").value;
    var security = document.getElementById("security").value;
    var amount = document.getElementById("amount").value;
    var dAmount = document.getElementById("dAmount").value;

    amount = Number(amount);

    if(dAmount[0] == '$') {
        dAmount = dAmount.substr(1);
    }
    dAmount = Number(dAmount);

    if(validate(date, account, type, security, amount, dAmount)) {
        var costBasis = calculateCostBasis(amount, dAmount);
        dAmount = '$' + dAmount.toFixed(2);

        return [ date, account, type, security, amount, dAmount, costBasis
            ↪  ];
    }
    else return false;
}

function validate(date, account, type, security, amount, dAmount) {
    if(!validateDate(date)) return false;
    if(!validateAccount(account)) return false;
    if(!validateType(type)) return false;
    if(!validateSecurity(security)) return false;
    if(!validateAmount(amount)) return false;
    if(!validateDAmount(dAmount)) return false;

    return true;
}

function validateDate(date) {
    realDate = new Date();
    inputDate = document.getElementById('date').valueAsNumber;

    if(isNaN(inputDate)) {
        alert('Error: No date specified');
        return false;
    }

    if(realDate.valueOf() < inputDate) {
        alert('Error: Date is in the future');
        return false;
    }

    return true;
}

function validateAccount(account) {
    if(account == '') {
        alert('Error: Missing Account Number');
        return false;
    }

    return true;
}

function validateType(type) {
    if(type == '') {
        alert('Error: Missing Transaction Type');
        return false;
    }
    return true;
}

function validateSecurity(security) {
    if(security == ’’) {
        alert(’Error: Missing Security’);
        return false;
    }

    return true;
}

function validateAmount(amount) {
    if(amount == ’’) {
        alert(’Error: Missing Amount’);
        return false;
    }

    if(isNaN(amount)) {
        alert(’Error: Invalid Amount’);
        return false;
    }

    return true;
}

function validateDAmount(dAmount) {
    if(dAmount == ’’) {
        alert(’Error: Missing $ Amount’);
        return false;
    }

    if(isNaN(dAmount)) {
        alert(’Error: Invalid $ Amount’);
        return false;
    }

    return true;
}

function generateId() {
    var id = ’’;
    var idLength = 6;

    var characters = ’ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789’;
    var charactersLength = characters.length;

    var unique = false;

    while(!unique) {
        for(var i = 0; i < idLength; i++) {
            id += characters.charAt(Math.floor(Math.random() *
               ↪ charactersLength));
        }

        unique = true;
        for(var i = 0; i < document.getElementsByClassName('idCell').
           ↪ length; i++) {
            if(document.getElementsByClassName('idCell')[i].innerText ==
               ↪ id) {
                unique = false;
                break;
            }
        }
    }
    return id;
}

function calculateCostBasis(amount, dAmount) {
    costBasis = '$' + (dAmount / amount).toFixed(2);
    return costBasis;
}

function deleteRow(button) {
    var row = button.parentElement.parentElement;
    document.getElementById("tableBody").removeChild(row);

    if(document.getElementsByClassName('editing').length == 0) {
        document.getElementById('add').removeAttribute('hidden');
        document.getElementById('save').setAttribute('hidden', true);
        document.getElementById('discard').setAttribute('hidden', true);
    }
}

function editRow(button) {
    if(document.getElementsByClassName('editing').length > 0)
        document.getElementsByClassName('editing')[0].classList = "bodyRow
           ↪ ";

    var row = button.parentElement.parentElement;
    var rowContent = row.getElementsByTagName('td');
    row.classList = "bodyRow editing";

    document.getElementById('date').value = rowContent[1].innerText;
    document.getElementById('account').value = rowContent[2].innerText;
    document.getElementById('type').value = rowContent[3].innerText;
    document.getElementById('security').value = rowContent[4].innerText;
    document.getElementById('amount').value = rowContent[5].innerText;
    document.getElementById('dAmount').value = rowContent[6].innerText;

    document.getElementById('add').setAttribute('hidden', true);
    document.getElementById('save').removeAttribute('hidden');
    document.getElementById('discard').removeAttribute('hidden');
}

function saveChanges() {
    data = getData();
    if(data) {
        rowToEdit = document.getElementsByClassName('editing')[0];
        cellsToEdit = rowToEdit.getElementsByTagName('td');

        for(var i = 0; i < data.length; i++) {
            cellsToEdit[i + 1].innerHTML = data[i];
        }
        rowToEdit.classList = "bodyRow";
    }

    document.getElementById('add').removeAttribute('hidden');
    document.getElementById('save').setAttribute('hidden', true);
    document.getElementById('discard').setAttribute('hidden', true);
}

function discardChanges() {
    document.getElementsByClassName('editing')[0].classList = "bodyRow";

    document.getElementById('add').removeAttribute('hidden');
    document.getElementById('save').setAttribute('hidden', true);
    document.getElementById('discard').setAttribute('hidden', true);
}

function sortTable(column, ascending) {
    var tableBody = document.getElementById('tableBody');
    var rows = document.getElementsByClassName('bodyRow');

    var sorting = true;
    while(sorting) {
        sorting = false;
        for(var i = 0; i < (rows.length - 1); i++) {
            rowA = rows[i].getElementsByTagName('td')[column];
            rowB = rows[i + 1].getElementsByTagName('td')[column];

            var swap = false;

            if(ascending && rowA.innerHTML.toLowerCase() > rowB.innerHTML.
               ↪  toLowerCase()) swap = true;
            else if(!ascending && rowA.innerHTML.toLowerCase() < rowB.
               ↪  innerHTML.toLowerCase()) swap = true;

            if(swap) {
                sorting = true;
                rows[i].parentNode.insertBefore(rows[i + 1], rows[i]);
            }
        }
    }
}
```

C CSS Source

```
body {
    font-size: 14px;
}

#inputFields {
    padding: 10px 0;
    overflow-x: auto;
}

form {
    min-width: 1900px;
}

    form > section {
        width: 14%;
        display: inline-block;
    }

    input,
    select {
        min-width: 100px;
        width: 80%;
    }

    button {
        width: 40%;
    }

#table {
    max-height: 80vh;
    overflow: auto;
}

table {
    width: 100%;
    margin: auto;
    border-collapse: collapse;
}
    th {
        min-width: 200px;
        width: 10%;
        position: sticky;
        background: white;
        top: 0;
    }

    th > section {
        width: 80%;
        display: inline-block;
        padding: 0;
        margin: 0;
    }
```

```
.sort {
    width: 10%;
}

.sort > button {
    padding: 0;
    border: 0;
    display: block;
    width: 100%;
}

.editing {
    background-color: yellow;
}

#table,
table,
td,
th {
    box-shadow: 1px 1px black, inset 1px 1px black;
}
```