



Experiment 1.1

Student Name: Rishi Jain

UID: 23BAI70569

Branch: BE-AIT-CSE

Section/Group: 23AML_KRG-1

Semester: 5th

Date of Performance: 7 July 2025

Subject Name: Full Stack

Subject Code: 23CSP-339

1. **Problem Title:** To create a responsive banking interface with deposit/withdrawal functionality using HTML5, CSS3, and JavaScript.
2. **Problem Description:**
 - a. Design a banking UI with balance display
 - b. Implement deposit/withdrawal buttons
 - c. Make the interface responsive using media queries
 - d. Add interactive elements with JavaScript
 - e. Validate form inputs
3. **Hardware Requirements:**
 - a. Processor: Intel i3/ Ryzen 3 or higher.
 - b. RAM: 4GB minimum.
 - c. Display: 1366x768 resolution.
4. **HTML:**

```
5. <!DOCTYPE html>
6. <html lang="en">
7. <head>
8.   <meta charset="UTF-8">
9.   <meta name="viewport" content="width=device-width,
    initial-scale=1">
10.    <title>Banking Interface</title>
11.    <link rel="stylesheet" href="style.css">
12.  </head>
```

```

13.     <body>
14.
15.         <div class="bank-container">
16.             <h2>Bank Account</h2>
17.             <div class="balance">
18.                 Current Balance: ₹<span
19.                     id="balance">0</span>
20.             </div>
21.             <div class="transaction">
22.                 <input type="number" id="amount"
23.                     placeholder="Enter amount">
24.                 <div class="buttons">
25.                     <button
26.                         onclick="deposit()">Deposit</button>
27.                     <button
28.                         onclick="withdraw()">Withdraw</button>
29.                 </div>
30.             </div>
31.             <div class="summary">
32.                 <p>Total Deposited: ₹<span
33.                     id="totalDeposit">0</span></p>
34.                 <p>Total Withdrawn: ₹<span
35.                     id="totalWithdraw">0</span></p>
36.             </div>
37.
38.             <div class="table">
39.                 <table id="historyTable">
40.                     <thead>

```

```

41.         <tr>
42.             <th>Type</th>
43.             <th>Amount (₹)</th>
44.             <th>Balance (₹)</th>
45.             <th>Time</th>
46.         </tr>
47.     </thead>
48.     <tbody></tbody>
49. </table>
50. </div>
51.
52.     <script src="script.js"></script>
53. </body>
54. </html>
55.

```

5. CSS:

```

6. * {
7.     margin: 0;
8.     padding: 0;
9.     box-sizing: border-box;
10. }
11.
12. body {
13.     font-family: 'Segoe UI', sans-serif;
14.     background-color: #f0f2f5;
15.     display: flex;
16.     flex-direction: column;
17.     justify-content: center;
18.     align-items: center;
19.     height: 100vh;
20. }
21.
22. .bank-container {

```

```
23.     background-color: white;
24.     padding: 30px;
25.     border-radius: 12px;
26.     box-shadow: 0 4px 15px rgba(0, 0, 0, 0.1);
27.     width: 90%;
28.     max-width: 400px;
29.     text-align: center;
30. }
31.
32. .balance {
33.     font-size: 1.4rem;
34.     margin: 20px 0;
35.     color: #333;
36. }
37.
38. .transaction input {
39.     padding: 10px;
40.     width: 80%;
41.     margin-bottom: 10px;
42.     border: 1px solid #ccc;
43.     border-radius: 6px;
44. }
45.
46. .buttons button {
47.     padding: 10px 15px;
48.     margin: 5px;
49.     font-size: 1rem;
50.     border: none;
51.     border-radius: 6px;
52.     cursor: pointer;
53.     transition: background-color 0.3s ease;
54. }
55.
56. button:hover {
```

```
57.     background-color: #0077cc;
58.     color: white;
59. }
60.
61. .message {
62.     margin-top: 10px;
63.     color: red;
64.     font-size: 0.95rem;
65. }
66.
67. @media (max-width: 600px) {
68.     .bank-container {
69.         padding: 20px;
70.     }
71.
72.     .balance {
73.         font-size: 1.2rem;
74.     }
75.
76.     .buttons button {
77.         width: 100%;
78.         margin-top: 5px;
79.     }
80.
81.     .transaction input {
82.         width: 100%;
83.     }
84. }
85.
86. .table {
87.     margin-top: 20px;
88.     overflow-x: auto;
89. }
90.
```

```
91.   table {
92.     width: 100%;
93.     border-collapse: collapse;
94.     margin-top: 10px;
95.   }
96.
97.   th, td {
98.     padding: 8px 12px;
99.     border: 1px solid #ccc;
100.    text-align: center;
101.  }
102.
103.  th {
104.    background-color: #f2f2f2;
105.  }
106.
107.  @media (max-width: 600px) {
108.    th, td {
109.      font-size: 0.85rem;
110.      padding: 6px 8px;
111.    }
112.  }
113.
114.  .summary {
115.    margin-top: 15px;
116.    text-align: left;
117.    font-weight: 500;
118.  }
119.
120.  .summary p {
121.    margin: 5px 0;
122.  }
123.
124.  .deposit-row {
```

```
125.     background-color: #e6ffed;
126.     color: green;
127. }
128.
129. .withdraw-row {
130.     background-color: #ffe6e6;
131.     color: red;
```

6. JavaScript:

```
let balance = 0;
let totalDeposit = 0;
let totalWithdraw = 0;

function updateBalance() {
    document.getElementById('balance').innerText = balance;
    document.getElementById('totalDeposit').innerText =
totalDeposit;
    document.getElementById('totalWithdraw').innerText =
totalWithdraw;
}

function showMessage(msg, isError = true) {
    const messageDiv = document.getElementById('message');
    messageDiv.innerText = msg;
    messageDiv.style.color = isError ? 'red' : 'green';

    setTimeout(() => {
        messageDiv.innerText = '';
    }, 3000);
}

function addToHistory(type, amount, currentBalance) {
    const table =
```

```

document.getElementById('historyTable').getElementsByTagName('tbody')[0];
const row = table.insertRow();

const typeCell = row.insertCell(0);
const amountCell = row.insertCell(1);
const balanceCell = row.insertCell(2);
const timeCell = row.insertCell(3);

const time = new Date().toLocaleTimeString();

typeCell.innerText = type;
amountCell.innerText = `₹${amount}`;
balanceCell.innerText = `₹${currentBalance}`;
timeCell.innerText = time;

// Add styling class
row.classList.add(type === "Deposit" ? 'deposit-row' : 'withdraw-row');
}

function deposit() {
  const amount =
parseFloat(document.getElementById('amount').value);
  if (isNaN(amount) || amount <= 0) {
    showMessage('Please enter a valid deposit amount.');
```

return;

balance += amount;

totalDeposit += amount;

updateBalance();

addToHistory("Deposit", amount, balance);

showMessage(`Deposited ₹\${amount}`, false);


```

    document.getElementById('amount').value = '';
}

function withdraw() {
    const amount =
parseFloat(document.getElementById('amount').value);
    if (isNaN(amount) || amount <= 0) {
        showMessage('Please enter a valid withdrawal amount.');
```

```

    return;
    }

    if (amount > balance) {
        showMessage('Insufficient balance!');
```

```

    return;
    }

    balance -= amount;
    totalWithdraw += amount;
    updateBalance();
    addToHistory("Withdraw", amount, balance);
    showMessage(`Withdrew ₹${amount}`, false);
    document.getElementById('amount').value = '';
}

```

132. Learning Outcome:

- a. I learnt how to link and connect the script.js and css file with the Html.
- b. I learnt how to work with html file.
- c. I learnt how to write js file and css.