

# **Experiment 1.1**

**Student Name:** Suhani Rawat

**Branch:** BE-AIT-CSE

**Semester:** 5th

**Subject Name:** Full Stack

**UID:** 23BAI70070

**Section/Group:** 23AML\_KRG-1

**Date of Performance:** 7 July 2025

**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:**
   1. Design a banking UI with balance display
   2. Implement deposit/withdrawal buttons
   3. Make the interface responsive using media queries
   4. Add interactive elements with JavaScript
   5. Validate form inputs
3. **Hardware Requirements:** 
   1. Processor: Intel i3/ Ryzen 3 or higher.
   2. RAM: 4GB minimum.
   3. Display: 1366x768 resolution.
4. **HTML:**

|  |
| --- |
| 1. <!DOCTYPE html> 2. <html lang="en"> 3. <head> 4. <meta charset="UTF-8"> 5. <meta name="viewport" content="width=device-width, initial-scale=1"> 6. <title>Banking Interface</title> 7. <link rel="stylesheet" href="style.css"> 8. </head> 9. <body> 10. <div class="bank-container"> 11. <h2>Bank Account</h2> 12. <div class="balance"> 13. Current Balance: ₹<span id="balance">0</span> 14. </div> 15. <div class="transaction"> 16. <input type="number" id="amount" placeholder="Enter amount"> 17. <div class="buttons"> 18. <button onclick="deposit()">Deposit</button> 19. <button onclick="withdraw()">Withdraw</button> 20. </div> 21. </div> 22. <div id="message" class="message"></div> 23. </div> 24. <div class="summary"> 25. <p>Total Deposited: ₹<span id="totalDeposit">0</span></p> 26. <p>Total Withdrawn: ₹<span id="totalWithdraw">0</span></p> 27. </div> 29. <div class="table"> 30. <table id="historyTable"> 31. <thead> 32. <tr> 33. <th>Type</th> 34. <th>Amount (₹)</th> 35. <th>Balance (₹)</th> 36. <th>Time</th> 37. </tr> 38. </thead> 39. <tbody></tbody> 40. </table> 41. </div> 42. <script src="script.js"></script> 43. </body> 44. </html> |

1. **CSS:**

|  |
| --- |
| 1. \* { 2. margin: 0; 3. padding: 0; 4. box-sizing: border-box; 5. } 6. body { 7. font-family: 'Segoe UI', sans-serif; 8. background-color: #f0f2f5; 9. display: flex; 10. flex-direction: column; 11. justify-content: center; 12. align-items: center; 13. height: 100vh; 14. } 15. .bank-container { 16. background-color: white; 17. padding: 30px; 18. border-radius: 12px; 19. box-shadow: 0 4px 15px rgba(0, 0, 0, 0.1); 20. width: 90%; 21. max-width: 400px; 22. text-align: center; 23. } 24. .balance { 25. font-size: 1.4rem; 26. margin: 20px 0; 27. color: #333; 28. } 29. .transaction input { 30. padding: 10px; 31. width: 80%; 32. margin-bottom: 10px; 33. border: 1px solid #ccc; 34. border-radius: 6px; 35. } 36. .buttons button { 37. padding: 10px 15px; 38. margin: 5px; 39. font-size: 1rem; 40. border: none; 41. border-radius: 6px; 42. cursor: pointer; 43. transition: background-color 0.3s ease; 44. } 45. button:hover { 46. background-color: #0077cc; 47. color: white; 48. } 49. .message { 50. margin-top: 10px; 51. color: red; 52. font-size: 0.95rem; 53. } 54. @media (max-width: 600px) { 55. .bank-container { 56. padding: 20px; 57. } 58. .balance { 59. font-size: 1.2rem; 60. } 61. .buttons button { 62. width: 100%; 63. margin-top: 5px; 64. } 65. .transaction input { 66. width: 100%; 67. } 68. } 69. .table { 70. margin-top: 20px; 71. overflow-x: auto; 72. } 73. table { 74. width: 100%; 75. border-collapse: collapse; 76. margin-top: 10px; 77. } 78. th, td { 79. padding: 8px 12px; 80. border: 1px solid #ccc; 81. text-align: center; 82. } 83. th { 84. background-color: #f2f2f2; 85. } 86. @media (max-width: 600px) { 87. th, td { 88. font-size: 0.85rem; 89. padding: 6px 8px; 90. } 91. } 92. .summary { 93. margin-top: 15px; 94. text-align: left; 95. font-weight: 500; 96. } 97. .summary p { 98. margin: 5px 0; 99. } 100. .deposit-row { 101. background-color: #e6ffed; 102. color: green; 103. } 104. .withdraw-row { 105. background-color: #ffe6e6; 106. 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 = '';  } |

1. **Learning Outcome:** 
   1. I learnt how to link and connect the script.js and css file with the Html.
   2. I learnt how to work with html file.
   3. I learnt how to write js file and css.