Experiment 2

AIM: Responsive Banking UI with Deposit/Withdrawal Buttons and Balance Display

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Simple Banking UI</title>
  <!-- Tailwind CSS CDN -->
  <script src="https://cdn.tailwindcss.com"></script>
  <style>
    body {
      font-family: Arial, sans-serif;
    }
  </style>
</head>
<body class="bg-gray-100 flex items-center justify-center min-h-screen">

  <div class="bg-white p-8 rounded-lg shadow-lg w-full max-w-md">
    <h1 class="text-2xl font-bold mb-6 text-center">🏦 Simple Banking
UI</h1>

    <!-- Balance Display -->
    <div class="bg-blue-100 p-4 rounded-lg mb-6 text-center">
      <h2 class="text-lg font-semibold">Current Balance</h2>
      <p id="balance" class="text-3xl font-bold text-blue-600">₹0.00</p>
    </div>

    <!-- Deposit Section -->
    <div class="mb-4">
      <label class="block font-medium mb-2">Deposit Amount</label>
      <input type="number" id="depositAmount" placeholder="Enter
amount"
          class="border border-gray-300 p-2 rounded-lg w-full mb-2">
```

```
        <button onclick="deposit()"
            class="bg-green-500 text-white px-4 py-2 rounded-lg w-full
hover:bg-green-600">
          Deposit
        </button>
      </div>

      <!-- Withdraw Section -->
      <div>
        <label class="block font-medium mb-2">Withdraw Amount</label>
        <input type="number" id="withdrawAmount" placeholder="Enter
amount"
            class="border border-gray-300 p-2 rounded-lg w-full mb-2">
        <button onclick="withdraw()"
            class="bg-red-500 text-white px-4 py-2 rounded-lg w-full
hover:bg-red-600">
          Withdraw
        </button>
      </div>

      <!-- Message -->
      <p id="message" class="text-center mt-4 font-semibold"></p>
    </div>

    <script>
      let balance = 0;

      function updateBalanceDisplay() {
        document.getElementById('balance').textContent =
`₹${balance.toFixed(2)}`;
      }

      function deposit() {
        let amount =
parseFloat(document.getElementById('depositAmount').value);
        if (isNaN(amount) || amount <= 0) {
          showMessage("Please enter a valid deposit amount.", "text-
yellow-500");
          return;
        }
```

```
            balance += amount;
            updateBalanceDisplay();
            showMessage(`Successfully deposited ₹${amount.toFixed(2)}`,
"text-green-500");
            document.getElementById('depositAmount').value = "";
        }

        function withdraw() {
            let amount =
parseFloat(document.getElementById('withdrawAmount').value);
            if (isNaN(amount) || amount <= 0) {
                showMessage("Please enter a valid withdrawal amount.", "text-
yellow-500");
                return;
            }
            if (amount > balance) {
                showMessage("Insufficient funds!", "text-red-500");
                return;
            }
            balance -= amount;
            updateBalanceDisplay();
            showMessage(`Successfully withdrew ₹${amount.toFixed(2)}`,
"text-blue-500");
            document.getElementById('withdrawAmount').value = "";
        }

        function showMessage(text, colorClass) {
            let messageEl = document.getElementById('message');
            messageEl.textContent = text;
            messageEl.className = `text-center mt-4 font-semibold
${colorClass}`;
        }

        updateBalanceDisplay();
    </script>

</body>
        </html>
```

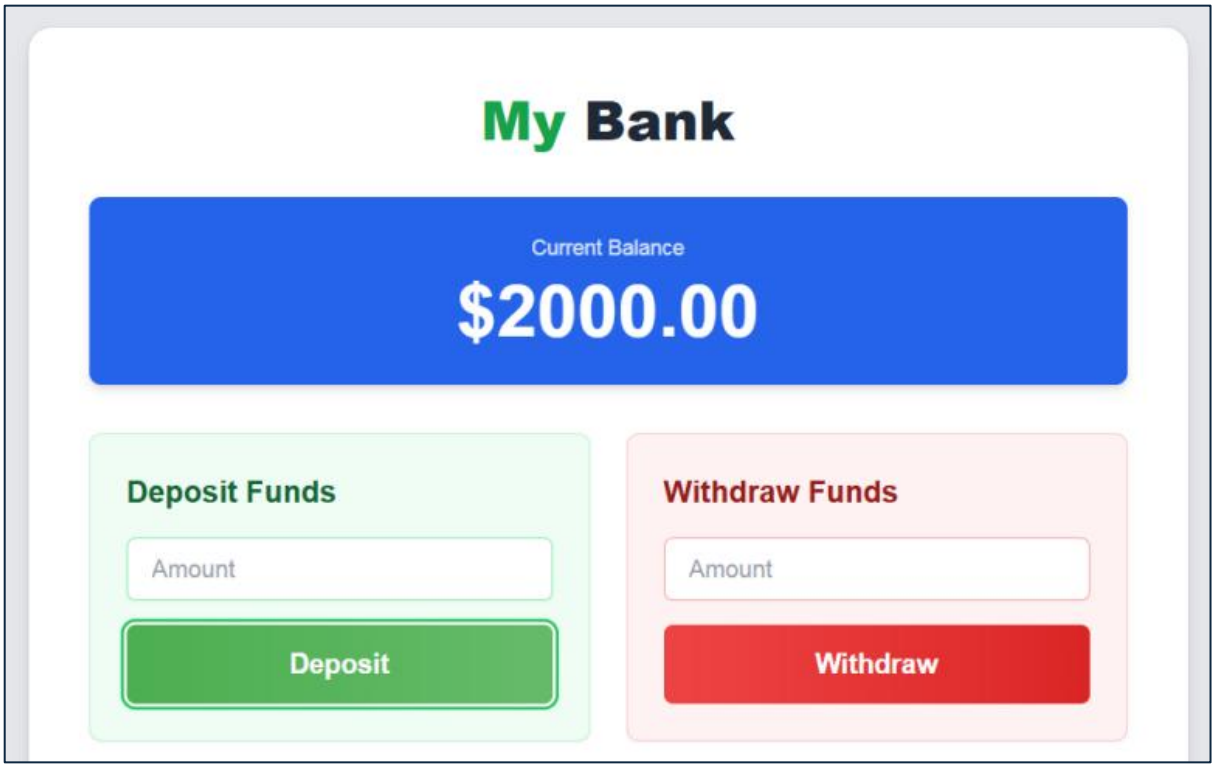

Figure 1: Experiment 2 Banking UI output