

EXP 2: Build a responsive banking UI with deposit/withdrawal buttons and balance display.

Code :

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Banking UI</title>

  <!-- Tailwind CSS CDN -->

  <script src="https://cdn.tailwindcss.com"></script>

  <!-- Google Fonts: Inter -->

  <link rel="preconnect" href="https://fonts.googleapis.com">

  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

  <link
href="https://fonts.googleapis.com/css2?family=Inter:wght@400;500;600;700&display=swap" rel="stylesheet">

  <!-- Icon Library -->

  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/font-awesome@6.1.1/css/all.min.css">

  <style>

    body {

      font-family: 'Inter', sans-serif;

    }

  </style>

</head>

<body class="bg-gray-100 flex items-center justify-center min-h-screen">

  <!-- Main Banking App Container -->

  <div class="w-full max-w-sm mx-auto bg-white rounded-2xl shadow-2xl p-6 space-y-6">
```

```
<!-- Header -->

<div class="flex justify-between items-center">

  <h1 class="text-xl font-bold text-gray-800">My Account</h1>

  <i class="fa-solid fa-bell text-gray-400"></i>

</div>


<!-- Balance Display -->

<div class="bg-indigo-600 text-white p-6 rounded-xl">

  <p class="text-sm text-indigo-200">Current Balance</p>

  <p id="balance-display" class="text-4xl font-bold mt-1">$1,250.75</p>

</div>


<!-- Transaction Controls -->

<div class="space-y-4">

  <h2 class="text-lg font-semibold text-gray-700">New Transaction</h2>

  <!-- Amount Input -->

  <div>

    <label for="amount" class="sr-only">Amount</label>

    <div class="relative">

      <div class="pointer-events-none absolute inset-y-0 left-0 pl-3 flex items-center">

        <span class="text-gray-500">$</span>

      </div>

      <input

        type="number"

        id="amount"

        name="amount"

        placeholder="0.00"

      >

    </div>

  </div>

</div>
```

```
        class="block w-full pl-7 pr-12 py-3 bg-gray-50 border border-gray-300 rounded-
lg focus:outline-none focus:ring-2 focus:ring-indigo-500 transition"
```

```
    >
```

```
  </div>
```

```
</div>
```

```
<!-- Action Buttons -->
```

```
<div class="grid grid-cols-2 gap-4">
```

```
  <button id="deposit-btn" class="w-full flex items-center justify-center gap-2 py-3
px-4 border border-transparent rounded-lg text-sm font-medium text-white bg-green-500
hover:bg-green-600 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-green-500
transition">
```

```
    <i class="fa-solid fa-arrow-down"></i>
```

```
    Deposit
```

```
  </button>
```

```
  <button id="withdraw-btn" class="w-full flex items-center justify-center gap-2 py-3
px-4 border border-transparent rounded-lg text-sm font-medium text-white bg-red-500
hover:bg-red-600 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-red-500
transition">
```

```
    <i class="fa-solid fa-arrow-up"></i>
```

```
    Withdraw
```

```
  </button>
```

```
</div>
```

```
</div>
```

```
<!-- Message Area -->
```

```
<div id="message-box" class="text-center text-sm font-medium text-gray-600 h-
5"></div>
```

```
</div>
```

```
<script>
```

```
// --- DOM Elements ---
```

```
const balanceDisplay = document.getElementById('balance-display');
```

```
const amountInput = document.getElementById('amount');
```

```
const depositBtn = document.getElementById('deposit-btn');
```

```
const withdrawBtn = document.getElementById('withdraw-btn');
```

```
const messageBox = document.getElementById('message-box');
```

```
// --- State ---
```

```
let currentBalance = 1250.75;
```

```
// --- Functions ---
```

```
/**
```

```
 * Updates the balance display with proper formatting.
```

```
*/
```

```
function updateDisplay() {
```

```
    balanceDisplay.textContent = `$$${currentBalance.toLocaleString('en-US', {  
minimumFractionDigits: 2, maximumFractionDigits: 2 }}}`;
```

```
}
```

```
/**
```

```
 * Displays a message to the user for a short duration.
```

```
 * @param {string} text - The message to display.
```

```
 * @param {string} color - The color of the message (e.g., 'text-red-500').
```

```
*/
```

```
function showMessage(text, color = 'text-gray-600') {
```

```

    messageBox.textContent = text;

    messageBox.className = `text-center text-sm font-medium h-5 ${color}`;

    setTimeout(() => {
        messageBox.textContent = "";
    }, 3000); // Message disappears after 3 seconds
}

/**
 * Handles the deposit logic.
 */
function handleDeposit() {
    const amount = parseFloat(amountInput.value);

    if (isNaN(amount) || amount <= 0) {
        showMessage('Please enter a valid positive amount.', 'text-red-500');

        return;
    }

    currentBalance += amount;

    updateDisplay();

    showMessage(`Successfully deposited $$${amount.toFixed(2)}.`, 'text-green-500');

    amountInput.value = "";
}

/**
 * Handles the withdrawal logic.
 */
function handleWithdraw() {
    const amount = parseFloat(amountInput.value);

    if (isNaN(amount) || amount <= 0) {

```

```
        showMessage('Please enter a valid positive amount.', 'text-red-500');
        return;
    }
    if (amount > currentBalance) {
        showMessage('Insufficient funds for this withdrawal.', 'text-red-500');
        return;
    }
    currentBalance -= amount;
    updateDisplay();
    showMessage(`Successfully withdrew $$${amount.toFixed(2)}.`, 'text-blue-500');
    amountInput.value = '';
}

// --- Event Listeners ---
depositBtn.addEventListener('click', handleDeposit);
withdrawBtn.addEventListener('click', handleWithdraw);

// --- Initial Display Update ---
// This ensures the initial balance is correctly formatted on page load.
updateDisplay();
</script>
</body>
</html>
```

OUTPUT :

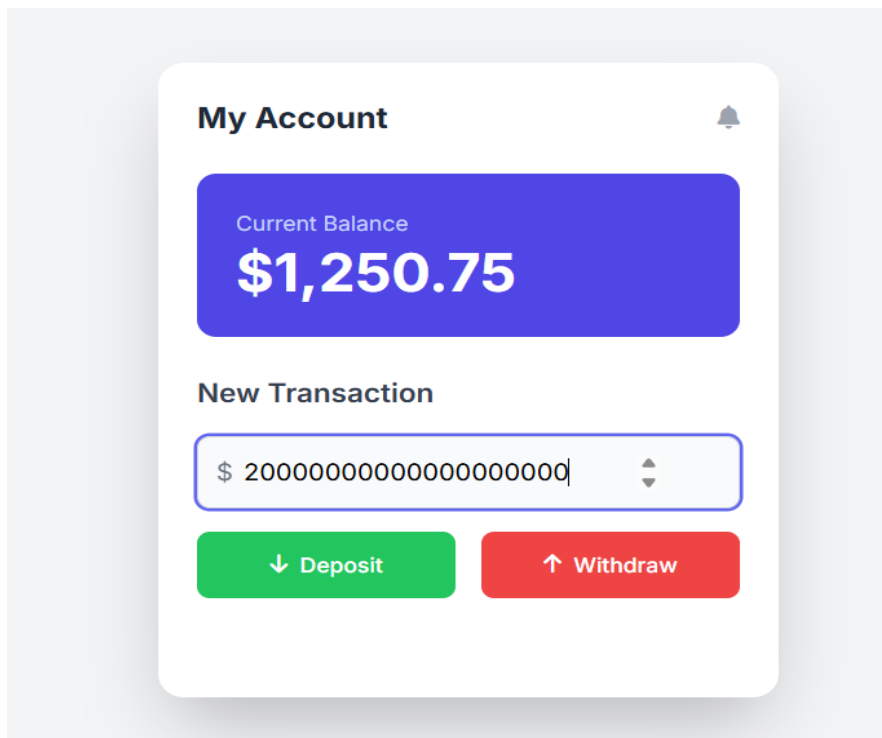


Figure 1 Deposit Money

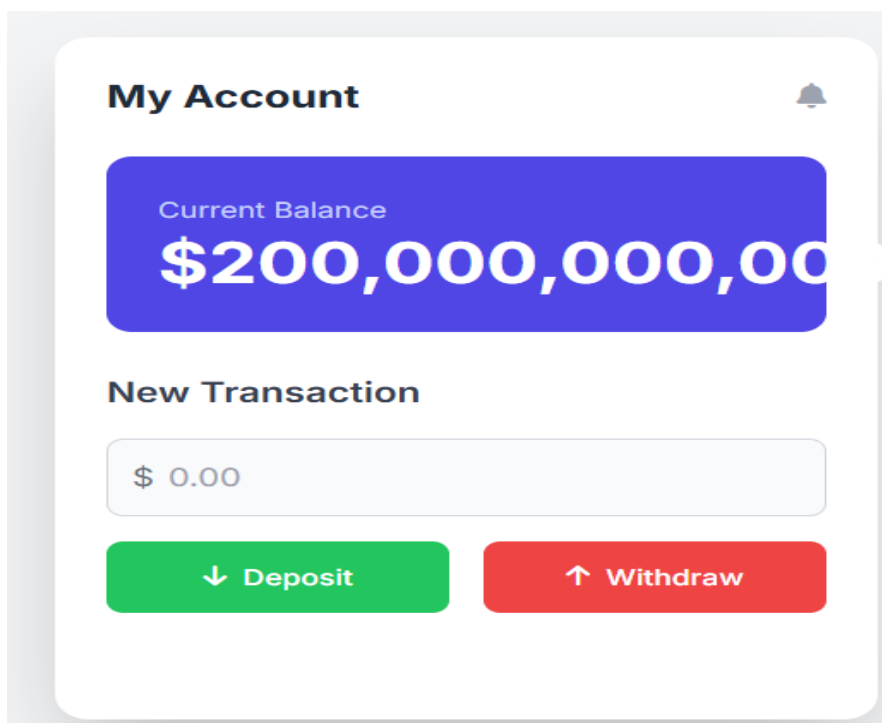


Figure 2 After Deposit