

Experiment 2

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

I'll create a responsive banking UI with deposit/withdrawal buttons and a balance display using HTML, CSS, and JavaScript. The UI will be simple and mobile-friendly, demonstrating basic functionality.

CODE:

```
<!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>
  <style>
    :root {
      --primary-color: #4CAF50;
      --secondary-color: #f44336;
      --background-color: #f0f2f5;
      --card-background: #ffffff;
      --text-color: #333;
      --light-text: #666;
      --border-radius: 8px;
      --box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
    }

    * {
      box-sizing: border-box;
      margin: 0;
      padding: 0;
    }

    body {
      font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, 'Open
      Sans', 'Helvetica Neue', sans-serif;
      background-color: var(--background-color);
      display: flex;
      justify-content: center;
      align-items: center;
      min-height: 100vh;
      color: var(--text-color);
      padding: 1rem;
    }

    .container {
      width: 100%;
      max-width: 500px;
      background: var(--card-background);
      border-radius: var(--border-radius);
      box-shadow: var(--box-shadow);
      padding: 2rem;
      text-align: center;
    }

    h1 {
      color: var(--text-color);
      margin-bottom: 0.5rem;
```

```
}

.balance-display {
  background-color: var(--primary-color);
  color: white;
  padding: 1.5rem;
  border-radius: var(--border-radius);
  margin: 1.5rem 0;
}

.balance-label {
  font-size: 1rem;
  margin-bottom: 0.5rem;
  text-transform: uppercase;
  letter-spacing: 1px;
  font-weight: 500;
}

#balance {
  font-size: 2.5rem;
  font-weight: bold;
  display: block;
}

.transaction-form {
  display: flex;
  flex-direction: column;
  gap: 1rem;
  margin-top: 1.5rem;
}

.transaction-form input {
  width: 100%;
  padding: 0.8rem;
  border: 1px solid #ddd;
  border-radius: var(--border-radius);
  font-size: 1rem;
  transition: border-color 0.3s;
}

.transaction-form input:focus {
  outline: none;
  border-color: var(--primary-color);
}

.button-group {
  display: flex;
  gap: 1rem;
  flex-wrap: wrap;
}

.button-group button {
  flex: 1 1 150px; /* Allows buttons to grow and wrap */
  padding: 0.8rem;
  border: none;
  border-radius: var(--border-radius);
  font-size: 1rem;
  font-weight: bold;
  cursor: pointer;
  transition: background-color 0.3s ease;
}

#depositBtn {
  background-color: var(--primary-color);
  color: white;
```

```

}

#depositBtn:hover {
  background-color: #43a047;
}

#withdrawBtn {
  background-color: var(--secondary-color);
  color: white;
}

#withdrawBtn:hover {
  background-color: #e53935;
}

@media (max-width: 480px) {
  .container {
    padding: 1.5rem;
  }

  .button-group {
    flex-direction: column;
  }

  #balance {
    font-size: 2rem;
  }
}
</style>
</head>
<body>

<div class="container">
  <h1>My Bank Account</h1>

  <div class="balance-display">
    <span class="balance-label">Current Balance</span>
    <span id="balance">$1000.00</span>
  </div>

  <div class="transaction-form">
    <input type="number" id="amountInput" placeholder="Enter amount" min="1" step="0.01">
    <div class="button-group">
      <button id="depositBtn">Deposit</button>
      <button id="withdrawBtn">Withdraw</button>
    </div>
  </div>
</div>

<script>
const balanceSpan = document.getElementById('balance');
const amountInput = document.getElementById('amountInput');
const depositBtn = document.getElementById('depositBtn');
const withdrawBtn = document.getElementById('withdrawBtn');

let currentBalance = 1000.00;

function updateBalance(newBalance) {
  balanceSpan.textContent = `$$${newBalance.toFixed(2)}`;
}

depositBtn.addEventListener('click', () => {
  const amount = parseFloat(amountInput.value);
  if (amount > 0) {
    currentBalance += amount;
  }
});

```

```
        updateBalance(currentBalance);
        amountInput.value = "";
    } else {
        alert('Please enter a valid amount to deposit.');
```

```
    }
});
```

```
withdrawBtn.addEventListener('click', () => {
    const amount = parseFloat(amountInput.value);
    if (amount > 0 && amount <= currentBalance) {
        currentBalance -= amount;
        updateBalance(currentBalance);
        amountInput.value = "";
    } else if (amount > currentBalance) {
        alert('Insufficient funds.');
```

```
    } else {
        alert('Please enter a valid amount to withdraw.');
```

```
    }
});
```

```
// Initial balance display
updateBalance(currentBalance);
</script>
```

```
</body>
</html>
```

OUTPUTS

EXPERIMENT 2

My Bank
Simple Banking Made Easy

Your Balance

\$5776.00

Make a Transaction

Enter Amount:

Deposit Money

Withdraw Money

Recent Transactions

Deposit

3:16 PM

+\$5676.00

Figure 1: OUTPUT OF EXPERIMENT 2 part a

My Bank

Simple Banking Made Easy

Your Balance

\$100.00

Make a Transaction

Enter Amount:

0.00

Deposit Money

Withdraw Money

Recent Transactions

Withdraw

3:16 PM

-\$5676.00

Deposit

3:16 PM

+\$5676.00

Figure 2: OUTPUT OF EXPERIMENT 2 part b