

# AIM

Responsive Banking UI with Deposit/Withdrawal Buttons and Balance Display

## Objective

Design a visually appealing and responsive banking user interface using only HTML and CSS, helping you practice modern layout techniques and styling without adding interactivity.

## Task Description

Create a banking interface that includes:

- A balance display section (for example, a card or a highlighted box).
- A Deposit button.
- A Withdraw button.

## HTML

```
index.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 UI</title>
7    <link rel="stylesheet" href="style.css">
8  </head>
9  <body>
10   <div class="bank-container">
11     <h1>My Bank Account</h1>
12     <div class="balance-box">
13       <p>Current Balance</p>
14       <h2 id="balance">$0.00</h2>
15     </div>
16
17     <div class="actions">
18       <input type="number" id="amount" placeholder="Enter amount">
19       <div class="buttons">
20         <button onclick="deposit()">Deposit</button>
21         <button onclick="withdraw()">Withdraw</button>
22       </div>
23     </div>
24   </div>
25
```

```

26 <script>
27   let balance = 0;
28
29   function updateBalance() {
30     document.getElementById("balance").textContent = "$" + balance.toFixed(2);
31   }
32
33   function deposit() {
34     let amt = parseFloat(document.getElementById("amount").value);
35     if (!isNaN(amt) && amt > 0) {
36       balance += amt;
37       updateBalance();
38       document.getElementById("amount").value = "";
39     }
40   }
41
42   function withdraw() {
43     let amt = parseFloat(document.getElementById("amount").value);
44     if (!isNaN(amt) && amt > 0 && amt <= balance) {
45       balance -= amt;
46       updateBalance();
47       document.getElementById("amount").value = "";
48     }
49   }
50
51   updateBalance();
52 </script>
53 </body>

```

## CSS

```

1 * {
2   box-sizing: border-box;
3 }
4 body {
5   font-family: Arial, sans-serif;
6   margin: 0;
7   padding: 0;
8   background: #f3f4f6;
9   display: flex;
10  justify-content: center;
11  align-items: center;
12  height: 100vh;
13 }
14 .bank-container {
15   background: #ffff;
16   padding: 30px;
17   border-radius: 12px;
18   width: 90%;
19   max-width: 400px;
20   box-shadow: 0 2px 10px rgba(0,0,0,0.1);
21   text-align: center;
22 }
23 h1 {
24   font-size: 28px;
25   margin-bottom: 20px;
26 }
27 .balance-box {
28   background-color: #e7f8e5;
29   padding: 15px;
30   border-radius: 10px;
31   margin-bottom: 20px;
32 }
33 #balance {
34   font-size: 30px;
35   color: #2e7d32;
36   margin-top: 5px;
37 }

```

```

37 }
38 .actions input[type="number"] {
39   width: 100%;
40   padding: 10px;
41   font-size: 16px;
42   margin-bottom: 15px;
43   border-radius: 5px;
44   border: 1px solid #ccc;
45 }
46 .buttons {
47   display: flex;
48   flex-direction: column;
49   gap: 10px;
50 }
51 .buttons button {
52   width: 100%;
53   padding: 12px;
54   font-size: 16px;
55   font-weight: bold;
56   border: none;
57   border-radius: 6px;
58   color: white;
59   cursor: pointer;
60   transition: background-color 0.2s ease;
61 }
62 .buttons button:first-child {
63   background-color: #4CAF50;
64 }
65 .buttons button:first-child:hover {
66   background-color: #43a047;
67 }
68 .buttons button:last-child {
69   background-color: #f44336;
70 }
71 .buttons button:last-child:hover {
72   background-color: #e53935;
73 }
74

```

## OUTPUT



