Experiment 2

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

Code

1. Html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Online Banking</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>

  <div class="banking-container">
    <div class="header">
      <h1>Digital Bank</h1>
    </div>

    <div class="balance">
      <div class="balance-label">Current Balance</div>
      <div id="balance-amount">₹1,000.00</div>
    </div>

    <div class="controls">
      <input type="number" id="amount-input" placeholder="Enter amount">
      <div class="buttons-container">
        <button id="deposit-btn" class="btn">Deposit</button>
        <button id="withdraw-btn" class="btn">Withdraw</button>
      </div>
    </div>

    <div class="transactions">
      <h2>Transaction History</h2>
      <ul id="transaction-list">
        </ul>
    </div>
  </div>

  <script src="script.js"></script>
</body>
</html>
```

2. CSS-

```
/* --- Basic Setup --- */
:root {
  --primary-color: #007bff;
  --success-color: #28a745;
  --danger-color: #dc3545;
  --light-gray: #f8f9fa;
  --dark-gray: #343a40;
  --border-radius: 8px;
}

body {
  font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, Helvetica, Arial, sans-serif;
  background-color: var(--light-gray);
  margin: 0;
  display: flex;
  justify-content: center;
  align-items: center;
  min-height: 100vh;
}

/* --- Banking App Container --- */
.banking-container {
  width: 100%;
  max-width: 400px;
  background: #ffffff;
  border-radius: var(--border-radius);
  box-shadow: 0 8px 20px rgba(0,0,0,0.1);
  padding: 2rem;
  box-sizing: border-box;
  margin: 1rem;
}

/* --- Header --- */
.header {
  text-align: center;
  margin-bottom: 2rem;
  color: var(--dark-gray);
}

.header h1 {
  margin: 0;
  font-size: 1.8rem;
}

/* --- Balance Display --- */
.balance {
  background-color: var(--primary-color);
  color: white;
```

```
    padding: 1.5rem;
    border-radius: var(--border-radius);
    text-align: center;
    margin-bottom: 2rem;
}

.balance-label {
    font-size: 1rem;
    margin-bottom: 0.5rem;
    opacity: 0.8;
}

#balance-amount   {
    font-size:  2.5rem;
    font-weight: bold;
    letter-spacing: 1px;
}

/* --- Controls (Input & Buttons) --- */
.controls {
    display: flex;
    flex-direction: column;
    gap: 1rem;
}

#amount-input {
    width: 100%;
    padding: 0.75rem;
    border: 1px solid #ced4da;
    border-radius: var(--border-radius);
    font-size: 1.2rem;
    text-align: center;
    box-sizing: border-box;
}

/* Hide the number input arrows for a cleaner look */
#amount-input::-webkit-outer-spin-button,
#amount-input::-webkit-inner-spin-button {
    -webkit-appearance: none;
    margin: 0;
}
#amount-input[type=number] {
    -moz-appearance:  textfield;
}

.buttons-container {
    display: grid;
    grid-template-columns: 1fr 1fr;
    gap: 1rem;
}
```

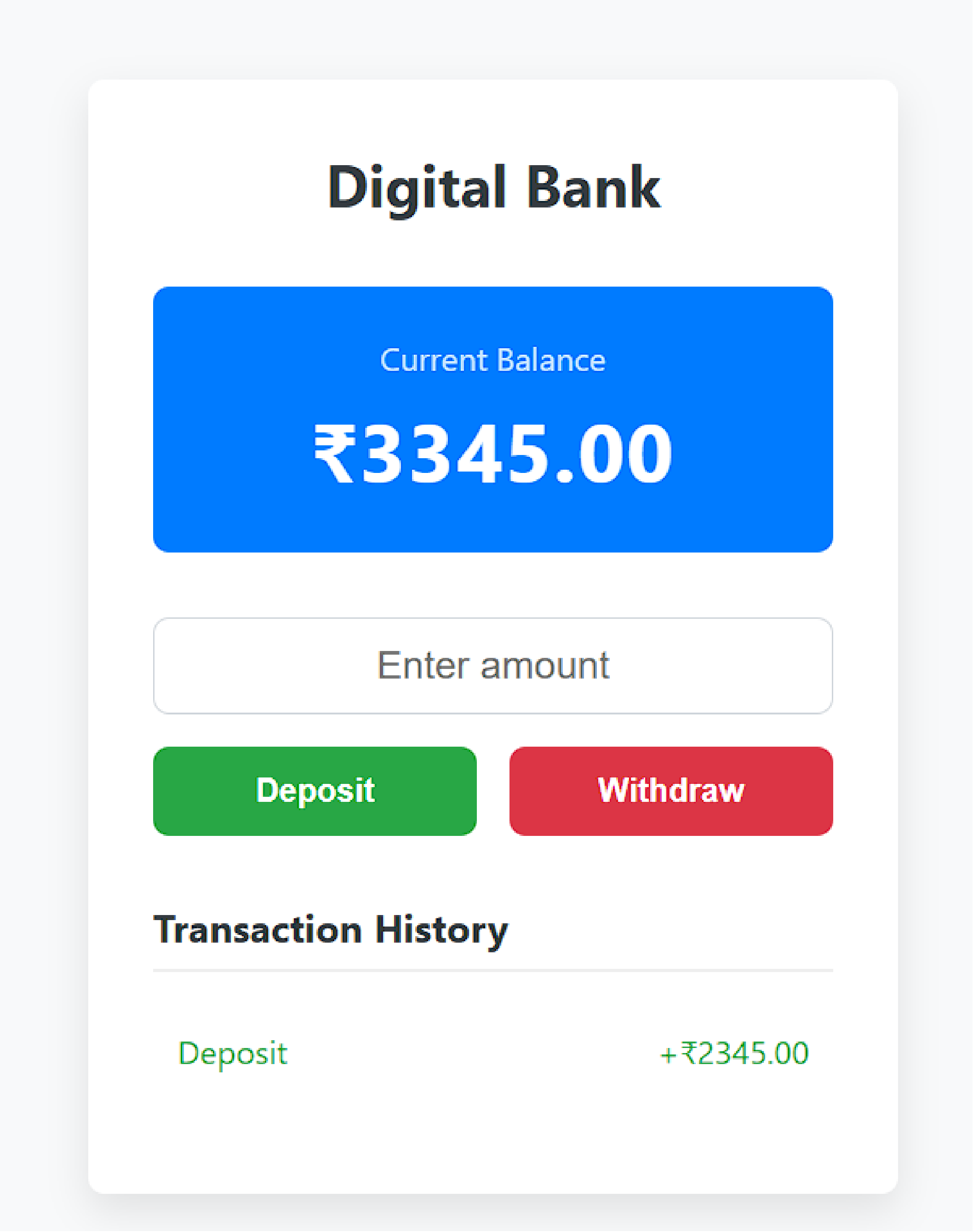
```css
.btn {
   padding: 0.8rem;
   border: none;
   border-radius: var(--border-radius);
   color: white;
   font-size: 1rem;
   font-weight: bold;
   cursor: pointer;
   transition: opacity 0.2s;
}

.btn:hover {
   opacity: 0.9;
}

#deposit-btn {
   background-color:  var(--success-color);
}

#withdraw-btn {
   background-color:  var(--danger-color);
}

/* --- Transaction History --- */
.transactions {
   margin-top: 2rem;
}
.transactions h2 {
   font-size: 1.2rem;
   color: var(--dark-gray);
   border-bottom: 2px solid #eee;
   padding-bottom: 0.5rem;
   margin-bottom: 1rem;
}
#transaction-list {
   list-style-type: none;
   padding: 0;
   max-height: 150px;
   overflow-y: auto;
}
#transaction-list li {
   display: flex;
   justify-content: space-between;
   padding: 0.75rem;
   border-bottom: 1px solid #eee;
}
 #transaction-list li:last-child {
   border-bottom: none;
}
```

```
.deposit {
  color: var(--success-color);
}
.withdrawal {
  color: var(--danger-color);
}
```

3. Javascript-

```
document.addEventListener('DOMContentLoaded', () => {
  // --- State ---
  let currentBalance = 1000.00;

  // --- Element References ---
  const balanceAmountEl = document.getElementById('balance-amount');
  const amountInputEl = document.getElementById('amount-input');
  const depositBtn = document.getElementById('deposit-btn');
  const withdrawBtn = document.getElementById('withdraw-btn');
  const transactionListEl = document.getElementById('transaction-list');

  // --- Functions ---

  /**
   * Updates the balance display with proper formatting.
   */
  function updateBalanceDisplay() {
    balanceAmountEl.textContent = `₹${currentBalance.toFixed(2)}`;
  }

  /**
   * Adds a transaction record to the history list.
   * @param {number} amount - The transaction amount.
   * @param {'deposit' | 'withdrawal'} type - The type of transaction.
   */
  function addTransaction(amount, type) {
    const li = document.createElement('li');
    const sign = type === 'deposit' ? '+' : '-';
    li.className = type === 'deposit' ? 'deposit' : 'withdrawal';
    li.innerHTML = `
      <span>${type.charAt(0).toUpperCase() + type.slice(1)}</span>
      <span>${sign}₹${amount.toFixed(2)}</span>
    `;
    transactionListEl.prepend(li); // Add new transactions to the top
  }

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

        if (isNaN(amount) || amount <= 0) {
            alert('Please enter a valid positive amount.');
            return;
        }

        currentBalance += amount;
        updateBalanceDisplay();
        addTransaction(amount, 'deposit');
        amountInputEl.value = ''; // Clear input
    }

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

        if (isNaN(amount) || amount <= 0) {
            alert('Please enter a valid positive amount.');
            return;
        }

        if (amount > currentBalance) {
            alert('Insufficient funds for this withdrawal.');
            return;
        }

        currentBalance -= amount;
        updateBalanceDisplay();
        addTransaction(amount, 'withdrawal');
        amountInputEl.value = ''; // Clear input
    }

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

    // --- Initial Render ---
    // Set the initial balance on page load
    updateBalanceDisplay();
});
```

Outputs
Experiment 2



Digital Bank

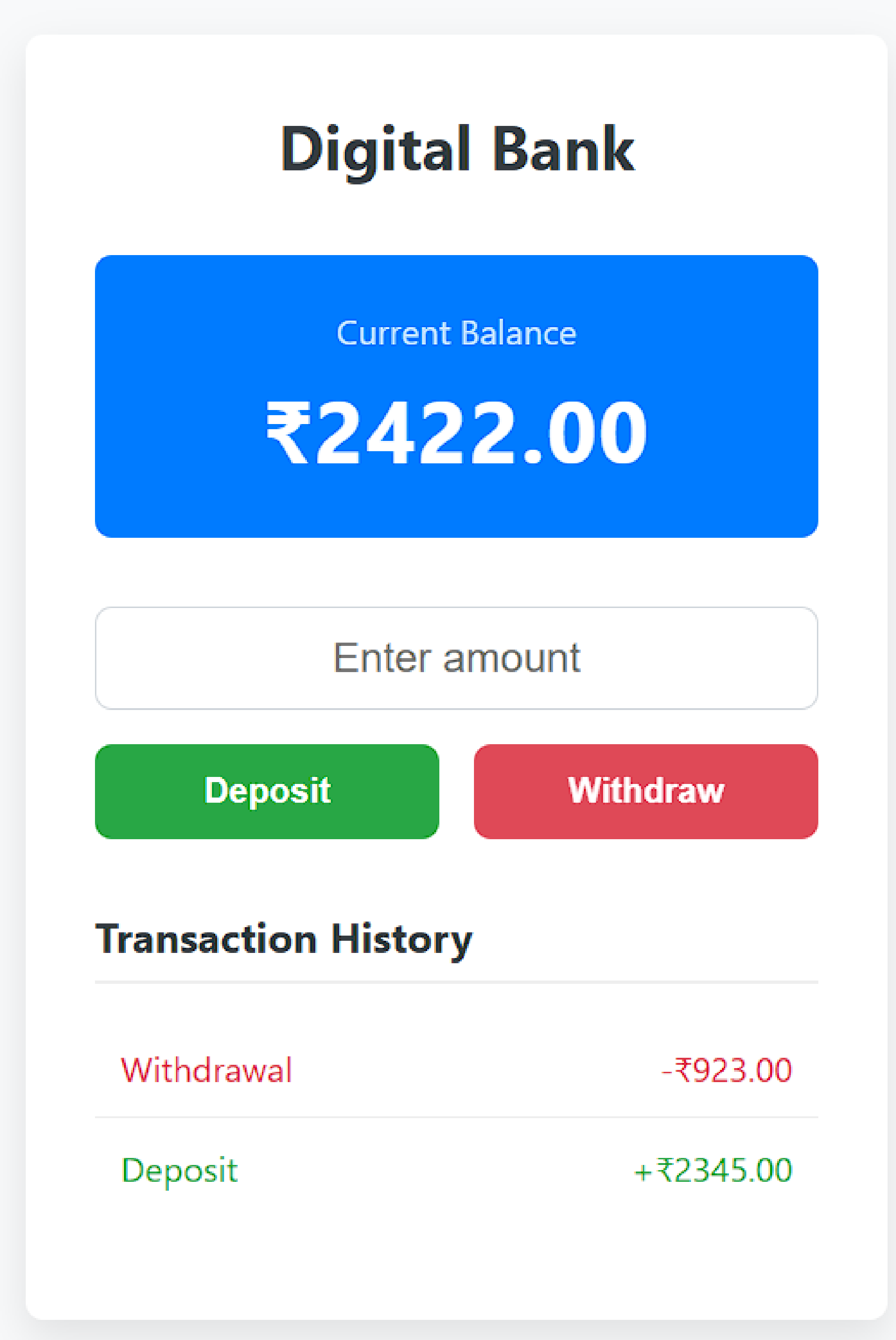
Current Balance

₹3345.00

Transaction History

Deposit +₹2345.00

Figure 1 Experiment 2. Deposited



Digital Bank

Current Balance

₹2422.00

Transaction History

Withdrawal	-₹923.00
Deposit	+₹2345.00

Figure 2 Experiment 2 (withdrawal)