

## EXPERIMENT-2

# AIM

To design and develop a responsive banking user interface using only HTML and CSS, which includes a balance display, deposit button, and withdraw button.

## OBJECTIVE

- Practice modern layout techniques using HTML and CSS.
- Create a responsive and visually clean UI that works on desktops, tablets, and mobile devices.
- Apply design principles like alignment, spacing, and centering using Flexbox and media queries.

## THEORY

HTML and CSS provide powerful tools for building visually appealing and responsive interfaces. In this task:

- HTML structures the content (balance, buttons).
- CSS Flexbox centers content both vertically and horizontally and manages layout.
- Media queries make the layout responsive across different screen sizes.
- The UI must be static—no interactivity with JavaScript.

**CODE**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Banking UI</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    * {
      box-sizing: border-box;
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="header">
      <h1>Banking UI</h1>
    </div>
    <div class="main">
      <div class="card">
        <div class="card-header">
          <h2>Welcome to Banking UI</h2>
        </div>
        <div class="card-body">
          <div class="form">
            <div class="input">
              <input type="text" value="Name" />
            </div>
            <div class="input">
              <input type="password" value="Password" />
            </div>
            <div class="input">
              <input type="password" value="Confirm Password" />
            </div>
            <div class="button">
              <button>Sign Up</button>
            </div>
          </div>
          <div class="form">
            <div class="input">
              <input type="text" value="Email" />
            </div>
            <div class="input">
              <input type="password" value="Password" />
            </div>
            <div class="input">
              <input type="password" value="Confirm Password" />
            </div>
            <div class="button">
              <button>Sign In</button>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</body>
</html>
```

```
body {
  font-family: Arial, sans-serif;
  background-color: #f0f2f5;
  margin: 0;
  padding: 0;
  display: flex;
  align-items: center;
  justify-content: center;
  height: 100vh;
}

.bank-ui {
  background-color: white;
  padding: 30px;
  border-radius: 10px;
  box-shadow: 0 4px 8px rgba(0,0,0,0.1);
  text-align: center;
  width: 300px;
}

.balance {
  font-size: 24px;
  font-weight: bold;
  color: #333;
  margin-bottom: 20px;
  padding: 15px;
  border: 2px solid #4CAF50;
  border-radius: 5px;
}

.button-group {
  display: flex;
  justify-content: space-between;
  gap: 10px;
}

button {
  padding: 10px;
  flex: 1;
  background-color: #4CAF50;
  color: white;
  border: none;
  border-radius: 5px;
  cursor: default;
}

@media (max-width: 500px) {
  .bank-ui {
    width: 90%;
  }

  .button-group {
```

```

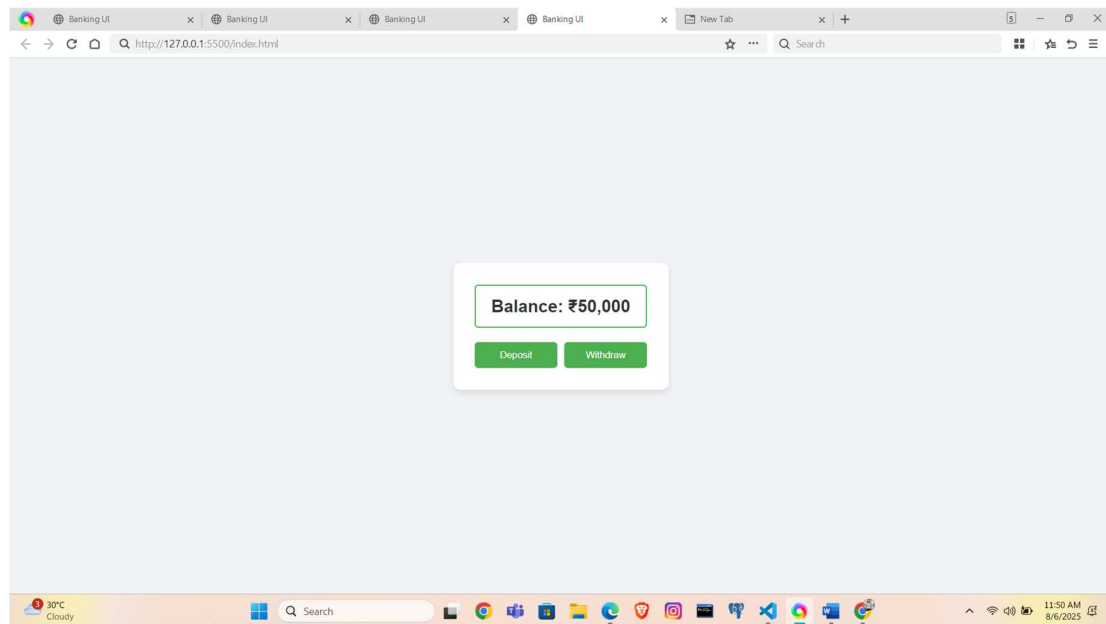
        flex-direction: column;
    }
}
</style>
</head>
<body>

<div class="bank-ui">
  <div class="balance">Balance: ₹50,000</div>
  <div class="button-group">
    <button>Deposit</button>
    <button>Withdraw</button>
  </div>
</div>

</body>
</html>

```

## OUTPUT



## LEARNING OUTCOME

- Use Flexbox for layout and centering.
- Build responsive web interfaces using media queries.
- Style HTML elements professionally using modern CSS practices.
- Create clean, structured UI components without using JavaScript.