FULL STACK

NAME: AMBIKA KASHYAP

CLASS: 23BCC1-A

UID:23BCC70026

EXPERIMENT-1.2

**AIM**

To design and implement a visually appealing and responsive **banking user interface** using only HTML and CSS that displays the current balance and includes Deposit and Withdraw buttons, ensuring proper alignment, spacing, and adaptability across different screen sizes.

**THEORY**

**HTML (HyperText Markup Language)** is used to structure web content by defining elements like headings, paragraphs, buttons, and containers.  
**CSS (Cascading Style Sheets)** styles these elements to control their layout, colors, fonts, and responsiveness.

**Key Concepts Used in This Experiment:**

1. **HTML Structure**
   * <div> elements are used for grouping UI components.
   * <h1> and <h2> provide headings and highlight balance information.
   * <button> elements represent Deposit and Withdraw options.
2. **CSS Styling**
   * **Flexbox** centers content horizontally and vertically.
   * **Background colors & gradients** create a professional look.
   * **Padding, margin, and border-radius** improve spacing and shape.
   * **Box-shadow** adds depth to the design.
3. **Responsive Design**
   * Media queries (@media) adjust layout for smaller devices.
   * Buttons switch from horizontal alignment to vertical stacking on mobile.
4. **Modern UI Principles**
   * Clean and minimal layout.
   * Consistent color scheme and typography.
   * Clear contrast for readability.

This approach ensures the UI works well on desktops, tablets, and mobile devices without JavaScript interactivity.

**PROCEDURE**

1. **Plan the Layout**
   * Decide on the sections: Title, Balance Display, Deposit Button, Withdraw Button.
   * Plan alignment so it appears centered on the page.
2. **Write the HTML Structure**
   * Create a main container <div> for the banking UI.
   * Inside, add a heading <h1>, a balance display section, and two buttons.
3. **Apply CSS Styling**
   * Use Flexbox to center the UI in the viewport.
   * Style the balance card with a light background, rounded corners, and padding.
   * Style buttons with colors, hover effects, and rounded edges.
4. **Add Responsiveness**
   * Use media queries to adjust layout for smaller screen sizes.
   * Stack buttons vertically when screen width is small.
5. **Test the Design**
   * Open in a browser and resize the window to ensure responsiveness.
   * Check that the design remains clean and readable across devices.

**CODE**

**index.html:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Responsive Banking UI</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<div class="bank-container">

<h1>My Bank</h1>

<div class="balance-card">

<p>Current Balance</p>

<h2>$5,000.00</h2>

</div>

<div class="button-group">

<button class="deposit-btn">Deposit</button>

<button class="withdraw-btn">Withdraw</button>

</div>

</div>

</body>

</html>

**style.css:**

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

font-family: Arial, sans-serif;

background: linear-gradient(to right, #0077b6, #0096c7);

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

}

.bank-container {

background: white;

padding: 2rem;

border-radius: 15px;

width: 350px;

text-align: center;

box-shadow: 0 4px 15px rgba(0,0,0,0.2);

}

.bank-container h1 {

margin-bottom: 1.5rem;

color: #023e8a;

}

.balance-card {

background: #f1f5f9;

padding: 1.5rem;

border-radius: 10px;

margin-bottom: 2rem;

}

.balance-card p {

font-size: 1rem;

color: #555;

}

.balance-card h2 {

margin-top: 0.5rem;

color: #023e8a;

}

.button-group button {

padding: 0.8rem 1.5rem;

border: none;

border-radius: 8px;

font-size: 1rem;

cursor: pointer;

margin: 0 0.5rem;

transition: 0.3s ease;

}

.deposit-btn {

background: #38b000;

color: white;

}

.deposit-btn:hover {

background: #2d6a4f;

}

.withdraw-btn {

background: #d00000;

color: white;

}

.withdraw-btn:hover {

background: #9d0208;

}

@media (max-width: 480px) {

.bank-container {

width: 90%;

padding: 1.5rem;

}

.button-group {

display: flex;

flex-direction: column;

gap: 10px;

}

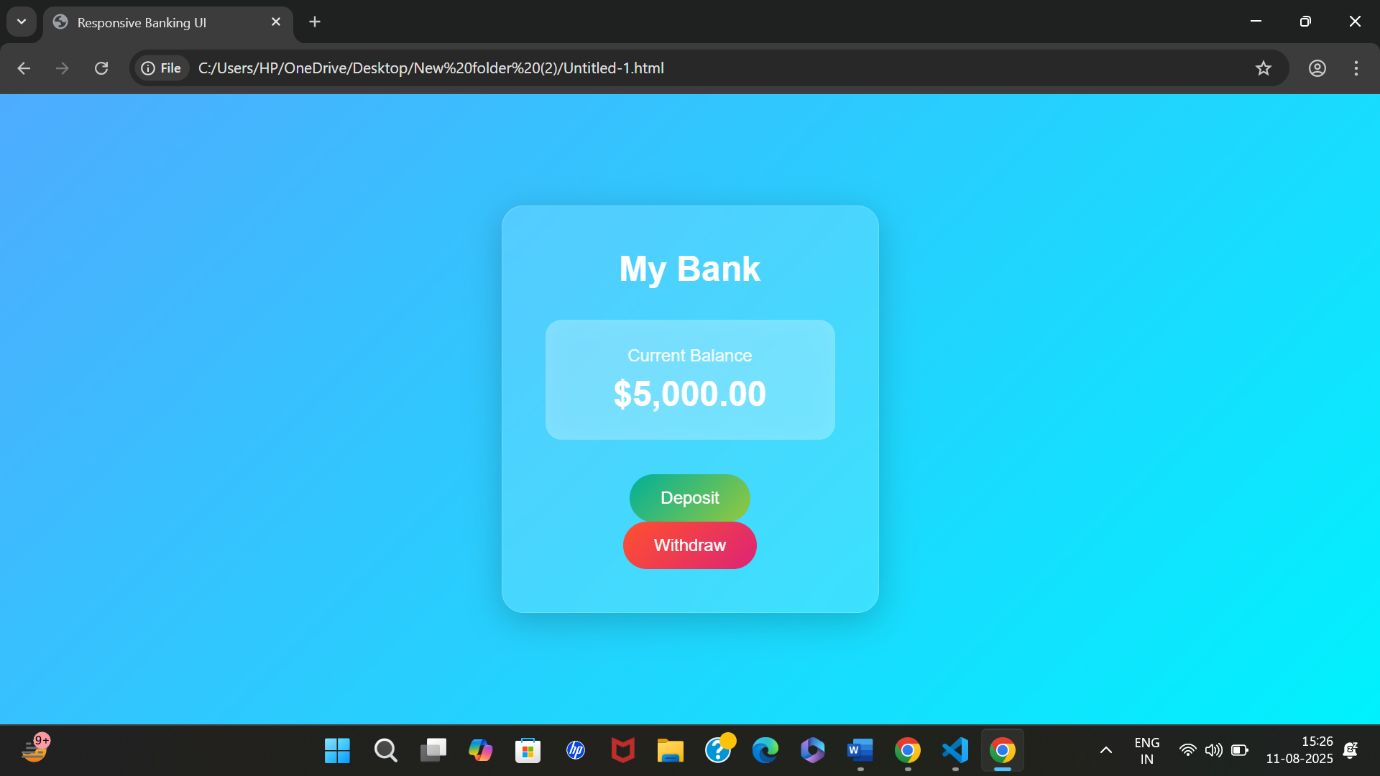
}

**LEARNING OUTCOMES:**

After completing this experiment, the learner will be able to:

1. **Structure a web page** using HTML tags like <div>, <h1>, <h2>, <p>, and <button>.
2. **Style UI components** using CSS properties such as colors, padding, margin, border-radius, and box-shadow.
3. **Center elements** using CSS Flexbox for better alignment and responsiveness.
4. **Apply responsive design** techniques using media queries for mobile compatibility.
5. **Create a professional UI** layout without requiring JavaScript interactivity.

**OUTPUT:**

****