**Aim :-** To design and implement a full stack web application using frontend, backend, and database technologies, demonstrating the core concepts of modern web development.

**Objectives :-** 1) To understand the architecture of full stack development (client-server model).

2) To develop a responsive frontend using HTML, CSS, and JavaScript (with frameworks like React/Angular).

3) To implement a backend server using Node.js/Express.js for handling API requests.

**Description :-** This experiment involves building a complete full stack web application from scratch as part of the Bytexl training module. It begins with designing the frontend interface using modern web technologies such as HTML, CSS, and JavaScript or frameworks like React.js.

**HTML:-**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>Simple Bank Interface</title>

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

</head>

<body>

  <div class="bank-interface">

    <h2>Account Balance: <span id="balance">$1000</span></h2>

    <input type="number" id="amount" placeholder="Enter amount" min="0">

    <div class="btn-group">

      <button onclick="deposit()">Deposit</button>

      <button onclick="withdraw()">Withdraw</button>

    </div>

    <p id="message"></p>

  </div>

  <script src="script.js"></script>

</body>

</html>

JAVA SCRIPT:

let balance = 1000;

function updateUI(message = '', isError = false) {

  document.getElementById('balance').textContent = `$${balance}`;

  document.getElementById('amount').value = '';

  const msg = document.getElementById('message');

  msg.textContent = message;

  msg.style.color = isError ? '#d01919' : 'green';

}

function deposit() {

  const amount = Number(document.getElementById('amount').value);

  if (amount > 0) {

    balance += amount;

    updateUI(`Deposited $${amount.toFixed(2)} successfully.`, false);

  } else {

    updateUI('Please enter a valid amount to deposit.', true);

  }

}

function withdraw() {

  const amount = Number(document.getElementById('amount').value);

  if (amount > 0 && amount <= balance) {

    balance -= amount;

  updateUI(`Withdrawn $${amount.toFixed(2)} successfully.`, false);

  } else if (amount > balance) {

    updateUI('Insufficient funds!', true);

  } else {

    updateUI('Please enter a valid amount to withdraw.', true);

  }

}

updateUI();

CSS:

:root {

  --primary-color: #2c3e50;

  --secondary-color: #3498db;

}

body {

  background: #fbf3f3;

  font-family: Arial, sans-serif;

  margin: 0;

  padding: 0;

}

.bank-interface {

  display: grid;

  gap: 1rem;

max-width: 500px;

  margin: 2rem auto;

  padding: 2rem;

  border-radius: 8px;

  box-shadow: 0 2px 10px rgba(0,0,0,0.1);

  background: #fff;

}

.bank-interface h2 {

  margin-top: 0;

  font-size: 1.4em;

  color: var(--primary-color);

}

input[type="number"] {

  width: 100%;

  padding: 0.5em;

  border: 1px solid #cdd5df;

  border-radius: 4px;

  font-size: 1em;

}

.btn-group {

  display: flex;

  gap: 1em;

}

button {

  flex: 1;

  padding: 0.7em 1em;

  background: var(--secondary-color);

  color: #fff;

  border: none;

  border-radius: 4px;

  font-weight: bold;

  cursor: pointer;

  transition: background .2s;

}

button:hover {

  background: #217dbb;

}

#message {

  min-height: 1.5em;

  font-size: 1em;

  color: #d01919;

font-weight: 500;

}

@media (max-width: 600px) {

  .bank-interface {

    padding: 1rem;

    margin: 1rem;

  }

}

@media (max-width: 400px) {

  .bank-interface {

    padding: 0.5rem;

    margin: 0.5rem;

  }

  input[type="number"] {

    font-size: 0.9em;

  }

  button {

    padding: 0.5em;

    font-size: 0.9em;

  }

}

