JS Capstone Assignment – 2

Name and Batch: Atharva Athanikar 72287317E DTS Batch-3

QUESTION 2(20 Marks)

Design a Simple Bank Management System that allows users to manage their bank accounts. The system should enable the creation of accounts, depositing money, withdrawing money, and checking account balances. Use ES6 modules for code organization and implement a class to encapsulate the bank account functionalities.

• ES6 Module Structure (5 Marks)

• Class Implementation (5 Marks)

• Methods for Managing Accounts (5 Marks)

• Functional User Interactions (5 marks)

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>Bank Management System</title>

  <style>

    body {

      font-family: Arial, sans-serif;

      display: flex;

      justify-content: center;

      align-items: center;

      height: 100vh;

      margin: 0;

      background-color: #6cc4e7;

    }

    .container {

      background-color: white;

      padding: 20px;

      border-radius: 8px;

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

      text-align: center;

    }

    button {

      padding: 10px 20px;

      margin: 10px;

      cursor: pointer;

      background-color: #007bff;

      color: white;

      border: none;

      border-radius: 5px;

    }

    input {

      padding: 8px;

      margin: 5px;

      border-radius: 5px;

      border: 1px solid #ccc;

    }

    button:hover {

      background-color: #0056b3;

    }

    h1, h2 {

      color: #333;

    }

    p {

      margin: 5px 0;

    }

    #transaction-section {

      display: none;

      margin-top: 20px;

    }

  </style>

</head>

<body>

  <div class="container">

    <h1>Bank Management System</h1>

    <div id="account-section">

      <button id="create-account-btn">Create Account</button>

      <div id="account-form" style="display:none;">

        <input type="text" id="account-name" placeholder="Enter Account Name" required>

        <input type="text" id="account-number" placeholder="Enter Account Number" required>

        <input type="email" id="account-email" placeholder="Enter Email" required>

        <input type="text" id="account-address" placeholder="Enter Address" required>

        <input type="number" id="initial-deposit" placeholder="Enter Initial Deposit" required>

        <button id="submit-account-btn">Submit</button>

      </div>

    </div>

    <div id="account-actions" style="display:none;">

      <h2>Welcome, <span id="account-name-display"></span></h2>

      <p>Account Number: <span id="account-number-display"></span></p>

      <p>Email: <span id="account-email-display"></span></p>

      <p>Address: <span id="account-address-display"></span></p>

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

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

      <button id="check-balance-btn">Check Balance</button>

      <p id="balance">Balance: $0</p>

    </div>

    <div id="transaction-section" style="display:none;">

      <h3 id="transaction-title">Enter amount of money to deposit or withdraw</h3>

      <input type="number" id="amount" placeholder="Enter amount" required>

      <button id="confirm-transaction-btn">Confirm</button>

      <button id="cancel-transaction-btn">Cancel</button>

    </div>

  </div>

  <script type="module" src="app.js"></script>

</body>

</html>

app.js:

// BankAccount class implementation

class BankAccount {

    constructor(accountName, accountNumber, email, address, initialDeposit) {

      this.accountName = accountName;

      this.accountNumber = accountNumber;

      this.email = email;

      this.address = address;

      this.balance = parseFloat(initialDeposit) || 0;

    }

    // Deposit method

    deposit(amount) {

      if (amount > 0) {

        this.balance += amount;

        return true;

      }

      return false;

    }

    // Withdraw method

    withdraw(amount) {

      if (amount > 0 && this.balance >= amount) {

        this.balance -= amount;

        return true;

      }

      return false;

    }

    // Check balance

    checkBalance() {

      return this.balance;

    }

    // Get account details

    getAccountDetails() {

      return {

        accountName: this.accountName,

        accountNumber: this.accountNumber,

        email: this.email,

        address: this.address,

        balance: this.balance

      };

    }

  }

  let account = null;

  const createAccountBtn = document.getElementById('create-account-btn');

  const accountForm = document.getElementById('account-form');

  const submitAccountBtn = document.getElementById('submit-account-btn');

  const accountActions = document.getElementById('account-actions');

  const accountNameDisplay = document.getElementById('account-name-display');

  const accountNumberDisplay = document.getElementById('account-number-display');

  const accountEmailDisplay = document.getElementById('account-email-display');

  const accountAddressDisplay = document.getElementById('account-address-display');

  const depositBtn = document.getElementById('deposit-btn');

  const withdrawBtn = document.getElementById('withdraw-btn');

  const checkBalanceBtn = document.getElementById('check-balance-btn');

  const balanceDisplay = document.getElementById('balance');

  const transactionSection = document.getElementById('transaction-section');

  const confirmTransactionBtn = document.getElementById('confirm-transaction-btn');

  const cancelTransactionBtn = document.getElementById('cancel-transaction-btn');

  const amountInput = document.getElementById('amount');

  const transactionTitle = document.getElementById('transaction-title');

  // Account Creation

  createAccountBtn.addEventListener('click', () => {

    accountForm.style.display = 'block';

  });

  // Submit Account Details

  submitAccountBtn.addEventListener('click', () => {

    const accountName = document.getElementById('account-name').value;

    const accountNumber = document.getElementById('account-number').value;

    const accountEmail = document.getElementById('account-email').value;

    const accountAddress = document.getElementById('account-address').value;

    const initialDeposit = document.getElementById('initial-deposit').value;

    if (accountName && accountNumber && accountEmail && accountAddress && initialDeposit) {

      account = new BankAccount(accountName, accountNumber, accountEmail, accountAddress, initialDeposit);

      const accountDetails = account.getAccountDetails();

      accountNameDisplay.textContent = accountDetails.accountName;

      accountNumberDisplay.textContent = accountDetails.accountNumber;

      accountEmailDisplay.textContent = accountDetails.email;

      accountAddressDisplay.textContent = accountDetails.address;

      balanceDisplay.textContent = `Balance: $${accountDetails.balance}`;

      accountForm.style.display = 'none';

      accountActions.style.display = 'block';

    } else {

      alert("Please fill in all fields.");

    }

  });

  // Handle Deposit Button

  depositBtn.addEventListener('click', () => {

    transactionTitle.textContent = "Enter amount of money to deposit into your account"; // Update title for deposit

    transactionSection.style.display = 'block';

    confirmTransactionBtn.onclick = () => {

      const amount = parseFloat(amountInput.value);

      if (amount > 0 && account.deposit(amount)) {

        balanceDisplay.textContent = `Balance: $${account.checkBalance()}`;

        transactionSection.style.display = 'none';

        amountInput.value = '';

      } else {

        alert("Invalid deposit amount.");

      }

    };

  });

  // Handle Withdraw Button

  withdrawBtn.addEventListener('click', () => {

    transactionTitle.textContent = "Enter amount of money to withdraw from your account"; // Update title for withdrawal

    transactionSection.style.display = 'block';

    confirmTransactionBtn.onclick = () => {

      const amount = parseFloat(amountInput.value);

      if (amount > 0 && account.withdraw(amount)) {

        balanceDisplay.textContent = `Balance: $${account.checkBalance()}`;

        transactionSection.style.display = 'none';

        amountInput.value = '';

      } else {

        alert("Insufficient balance or invalid withdrawal amount.");

      }

    };

  });

  // Check Balance

  checkBalanceBtn.addEventListener('click', () => {

    alert(`Current Balance: $${account.checkBalance()}`);

  });

  // Cancel Transaction

  cancelTransactionBtn.addEventListener('click', () => {

    transactionSection.style.display = 'none';

    amountInput.value = '';

  });

Output:















