

EXP-1:

Create a student registration form with HTML5 validation for name, email, and age inputs:

```
<!DOCTYPE html>

<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Student Registration Form</title>
    <script src="https://cdn.tailwindcss.com"></script>
    <!-- Importing the 'Poppins' font -->
        <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600;700&display=swap" rel="stylesheet">
<style>
    /* Applying the new font and custom validation colors */
    body {
        font-family: 'Poppins', sans-serif;
    }
    input:invalid {
        border-color: #f87171; /* red-400 */
    }
    input:valid {
        border-color: #34d399; /* emerald-400 */
    }
    /* Simple spinner for loading state */
    .spinner {
        border: 4px solid rgba(0, 0, 0, 0.1);
        width: 36px;
        height: 36px;
        border-radius: 50%;
    }
```

```

border-left-color: #14b8a6; /* teal-500 */
animation: spin 1s ease infinite;
}

@keyframes spin {
  0% { transform: rotate(0deg); }
  100% { transform: rotate(360deg); }
}


```

</style>

</head>

<body class="bg-gray-50 flex items-center justify-center min-h-screen p-4">

<div class="w-full max-w-md mx-auto bg-white rounded-2xl shadow-xl p-8">

<!-- Form Header -->

<div class="text-center mb-8">

<h1 class="text-3xl font-bold text-gray-800">Register for a Course</h1>

<p class="text-gray-500 mt-2">Let's get you started</p>

</div>

<!-- Registration Form -->

<form id="registrationForm" novalidate>

<!-- Name Input -->

<div class="mb-6">

<label for="name" class="block text-sm font-medium text-gray-700 mb-2">Full Name</label>

<input type="text" id="name" name="name" placeholder="e.g., Alex Doe" required minlength="3" class="w-full px-4 py-3 bg-gray-100 border-2 border-gray-200 rounded-lg focus:outline-none focus:ring-2 focus:ring-teal-400 focus:border-teal-400 transition duration-200">

</div>

<!-- Email Input -->

```
<div class="mb-6">
    <label for="email" class="block text-sm font-medium text-gray-700 mb-2">Email Address</label>
    <input type="email" id="email" name="email"
        class="w-full px-4 py-3 bg-gray-100 border-2 border-gray-200 rounded-lg focus:outline-none focus:ring-2 focus:ring-teal-400 focus:border-teal-400 transition duration-200"
        placeholder="you@example.com"
        required>
</div>

<!-- Age Input -->
<div class="mb-6">
    <label for="age" class="block text-sm font-medium text-gray-700 mb-2">Age</label>
    <input type="number" id="age" name="age"
        class="w-full px-4 py-3 bg-gray-100 border-2 border-gray-200 rounded-lg focus:outline-none focus:ring-2 focus:ring-teal-400 focus:border-teal-400 transition duration-200"
        placeholder="Enter your age"
        required
        min="16"
        max="100">
</div>

<!-- ✨ AI Course Suggestion Section (initially hidden) -->
<div id="course-suggestion-section" class="hidden mb-8 text-center">
    <button type="button" id="suggest-courses-btn" class="w-full bg-gray-700 text-white font-semibold py-2 px-4 rounded-lg hover:bg-gray-800 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-gray-600 transition duration-300">
        ✨ Suggest Courses For Me
    </button>
</div>

<!-- Submit Button -->
<button type="submit"
```

```
        class="w-full bg-teal-500 text-white font-bold py-3 px-4 rounded-lg hover:bg-teal-600 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-teal-500 transition duration-300 ease-in-out shadow-lg hover:shadow-xl">
```

```
        Register Now
```

```
    </button>
```

```
  </form>
```

```
</div>
```

```
<!-- Modal for Gemini API Response -->
```

```
  <div id="gemini-modal" class="fixed inset-0 bg-black bg-opacity-50 flex items-center justify-center p-4 hidden z-50">
```

```
    <div class="bg-white rounded-xl shadow-2xl w-full max-w-lg max-h-[90vh] flex flex-col">
```

```
      <div class="flex justify-between items-center p-4 border-b">
```

```
        <h3 id="modal-title" class="text-lg font-semibold text-gray-800">AI Response</h3>
```

```
        <button id="modal-close-btn" class="p-2 rounded-full hover:bg-gray-200 text-gray-500 hover:text-gray-800">
```

```
          &times;
```

```
        </button>
```

```
      </div>
```

```
      <div id="modal-content" class="p-6 overflow-y-auto">
```

```
        <!-- Loading spinner or content will be shown here -->
```

```
      </div>
```

```
      <div id="modal-footer" class="p-4 border-t text-right">
```

```
        <button id="modal-action-btn" class="bg-teal-500 text-white font-bold py-2 px-4 rounded-lg hover:bg-teal-600 transition">Close</button>
```

```
      </div>
```

```
    </div>
```

```
</div>
```

```
<script>
```

```
  // --- DOM Elements ---
```

```
  const form = document.getElementById('registrationForm');
```

```
  const ageInput = document.getElementById('age');
```

```

const courseSuggestionSection = document.getElementById('course-suggestion-section');

const suggestCoursesBtn = document.getElementById('suggest-courses-btn');

const geminiModal = document.getElementById('gemini-modal');

const modalTitle = document.getElementById('modal-title');

const modalContent = document.getElementById('modal-content');

const modalCloseBtn = document.getElementById('modal-close-btn');

const modalActionBtn = document.getElementById('modal-action-btn');

// --- Event Listeners ---

// Show course suggestion button when a valid age is entered
ageInput.addEventListener('input', () => {
  if (ageInput.checkValidity()) {
    courseSuggestionSection.classList.remove('hidden');
  } else {
    courseSuggestionSection.classList.add('hidden');
  }
});

// Handle form submission
form.addEventListener('submit', async function(event) {
  event.preventDefault();
  if (form.checkValidity()) {
    const studentName = document.getElementById('name').value.split(' ')[0];
    showModal('Registration Successful!');
    modalContent.innerHTML = `<div class="flex justify-center items-center h-32"><div class="spinner"></div></div>`;
    const prompt = `Generate a short, friendly, and encouraging welcome message for a new student named ${studentName} who just registered for a course. Mention that their registration is confirmed.`;
    const responseText = await callGeminiAPI(prompt);
  }
});

```

```

        modalContent.innerHTML = `<div class="prose max-w-none">${formatResponse(responseText)}</div>`;

        modalActionBtn.textContent = 'Finish';

        modalActionBtn.onclick = () => {
            hideModal();
            form.reset();
            courseSuggestionSection.classList.add('hidden');
        };

    } else {
        console.log('Form is invalid');
    }
});

// Handle course suggestion button click
suggestCoursesBtn.addEventListener('click', async () => {
    const studentAge = ageInput.value;
    showModal('Suggesting Courses...');

    modalContent.innerHTML = `<div class="flex justify-center items-center h-32"><div class="spinner"></div></div>`;

    const prompt = `Based on a student's age of ${studentAge}, suggest three interesting and relevant courses they could take. Provide a one-sentence description for each course. Format the response as a simple list.`;

    const responseText = await callGeminiAPI(prompt);

        modalContent.innerHTML = `<div class="prose max-w-none">${formatResponse(responseText)}</div>`;

        modalActionBtn.textContent = 'Close';
        modalActionBtn.onclick = hideModal;
    });
}

```

```
// --- Modal Functions ---
const showModal = (title) => {
    modalTitle.textContent = title;
    geminiModal.classList.remove('hidden');
};

const hideModal = () => {
    geminiModal.classList.add('hidden');
};

modalCloseBtn.addEventListener('click', hideModal);
modalActionBtn.addEventListener('click', hideModal); // Default action

// --- Gemini API Call ---
async function callGeminiAPI(prompt, retries = 3, delay = 1000) {
    const apiKey = ""; // This will be handled by the environment
    const apiUrl = `https://generativelanguage.googleapis.com/v1beta/models/gemini-2.5-flash-preview-05-20:generateContent?key=${apiKey}`;
    const payload = { contents: [{ role: "user", parts: [{ text: prompt }] }] };

    try {
        const response = await fetch(apiUrl, {
            method: 'POST',
            headers: { 'Content-Type': 'application/json' },
            body: JSON.stringify(payload)
        });

        if (!response.ok) throw new Error(`HTTP error! status: ${response.status}`);
        const result = await response.json();
        const text = result.candidates?.[0]?.content?.parts?.[0]?.text;
    }
}
```

```

if (text) {
    return text;
} else {
    console.error("Unexpected API response structure:", result);
    return "Sorry, the AI returned an unexpected response format.";
}
} catch (error) {
    console.error("Error calling Gemini API:", error);
    if (retries > 0) {
        await new Promise(res => setTimeout(res, delay));
        return callGeminiAPI(prompt, retries - 1, delay * 2); // Exponential backoff
    }
    return "An error occurred while contacting the AI. Please try again later.";
}
}

// --- Formatting Helper ---
function formatResponse(text) {
    // Basic Markdown-like to HTML conversion
    return text
        .replace(/\*\*(.*?)\*\*/g, '<strong>$1</strong>')
        .replace(/^- (.*)$/gim, '<li class="ml-4 mb-2 list-disc">$1</li>')
        .replace(/\n/g, '<br>');
}
</script>
</body>
</html>

```

EXP-2:

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

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Banking UI</title>
  <script src="https://cdn.tailwindcss.com"></script>
  <link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;500;600;700;800&display=swap" rel="stylesheet">
<style>
  body {
    font-family: 'Inter', sans-serif;
  }
  .card-glow {
    box-shadow: 0 10px 15px -3px rgba(59, 130, 246, 0.2), 0 4px 6px -2px rgba(59, 130, 246, 0.1);
  }
</style>
</head>
<body class="bg-gray-900 flex items-center justify-center min-h-screen p-4">

  <div class="w-full max-w-sm mx-auto bg-gray-800 rounded-2xl shadow-2xl p-6 text-white">
    <!-- Header -->
    <div class="flex justify-between items-center mb-6">
      <h1 class="text-xl font-bold text-gray-200">My Account</h1>
      <div class="w-10 h-10 bg-blue-500 rounded-full flex items-center justify-center font-bold text-lg">
        JD
      </div>
    </div>
    <!-- Balance Display Card -->
```

```
<div class="bg-gradient-to-br from-blue-500 to-blue-700 rounded-xl p-6 mb-8 card-glow">
  <p class="text-sm text-blue-200 uppercase tracking-wider">Current Balance</p>
  <p id="balance" class="text-4xl font-extrabold mt-2 tracking-tight">$1,250.75</p>
</div>

<!-- Transaction Controls -->
<div>
  <h2 class="text-lg font-semibold text-gray-300 mb-4">New Transaction</h2>

  <!-- Amount Input -->
  <div class="mb-4">
    <label for="amount" class="sr-only">Amount</label>
    <div class="relative">
      <div class="absolute inset-y-0 left-0 pl-3 flex items-center pointer-events-none">
        <span class="text-gray-400">$</span>
      </div>
      <input type="number" id="amount" name="amount"
        class="w-full bg-gray-700 border-2 border-gray-600 rounded-lg pl-8 pr-4 py-3 text-white
placeholder-gray-400 focus:outline-none focus:ring-2 focus:ring-blue-500 focus:border-blue-500
transition">
        placeholder="0.00">
    </div>
  </div>

  <!-- Action Buttons -->
  <div class="grid grid-cols-2 gap-4">
    <button id="depositBtn" class="w-full bg-green-500 font-bold py-3 px-4 rounded-lg hover:bg-
green-600 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-offset-gray-800 focus:ring-
green-500 transition duration-300">
      Deposit
    </button>
  </div>
</div>
```

```
<button id="withdrawBtn" class="w-full bg-red-500 font-bold py-3 px-4 rounded-lg hover:bg-red-600 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-offset-gray-800 focus:ring-red-500 transition duration-300">  
    Withdraw  
</button>  
</div>  
</div>  
  
<!-- Message Area -->  
<p id="message" class="text-center text-yellow-400 mt-6 h-5"></p>  
  
</div>  
  
<script>  
    // --- DOM Elements ---  
  
    const balanceEl = document.getElementById('balance');  
  
    const amountInput = document.getElementById('amount');  
  
    const depositBtn = document.getElementById('depositBtn');  
  
    const withdrawBtn = document.getElementById('withdrawBtn');  
  
    const messageEl = document.getElementById('message');  
  
    // --- State ---  
  
    let currentBalance = 1250.75;  
  
    // --- Functions ---  
  
    /**  
     * Updates the balance display with proper formatting.  
     */  
    function updateBalanceDisplay() {  
        balanceEl.textContent = currentBalance.toLocaleString('en-US', {  
            style: 'currency',  
        })  
    }  
    withdrawBtn.addEventListener('click', () => {  
        const amount = amountInput.value;  
        if (amount === '') {  
            messageEl.textContent = 'Please enter an amount.';  
            return;  
        }  
        const newBalance = currentBalance - Number(amount);  
        currentBalance = newBalance;  
        updateBalanceDisplay();  
        messageEl.textContent = `Withdrawn: ${amount}. New balance: ${currentBalance}`;  
    })  
    depositBtn.addEventListener('click', () => {  
        const amount = amountInput.value;  
        if (amount === '') {  
            messageEl.textContent = 'Please enter an amount.';  
            return;  
        }  
        const newBalance = currentBalance + Number(amount);  
        currentBalance = newBalance;  
        updateBalanceDisplay();  
        messageEl.textContent = `Deposited: ${amount}. New balance: ${currentBalance}`;  
    })  
</script>
```

```

    currency: 'USD'
  });
}

/** 
 * Displays a message to the user and clears it after a delay.
 * @param {string} msg - The message to display.
 * @param {boolean} isError - If true, displays the message in red.
 */
function showMessage(msg, isError = false) {
  messageEl.textContent = msg;
  messageEl.style.color = isError ? '#f87171' : '#facc15'; // red-400 or yellow-400
  setTimeout(() => {
    messageEl.textContent = "";
  }, 3000);
}

/** 
 * Handles the deposit logic.
 */
function handleDeposit() {
  const amount = parseFloat(amountInput.value);
  if (isNaN(amount) || amount <= 0) {
    showMessage('Please enter a valid positive amount.', true);
    return;
  }
  currentBalance += amount;
  updateBalanceDisplay();
  showMessage(`Successfully deposited ${amount.toLocaleString('en-US', { style: 'currency', currency: 'USD' })}.`);
  amountInput.value = "";
}

```

```
}

/**
 * Handles the withdrawal logic.
 */
function handleWithdraw() {
    const amount = parseFloat(amountInput.value);
    if (isNaN(amount) || amount <= 0) {
        showMessage('Please enter a valid positive amount.', true);
        return;
    }
    if (amount > currentBalance) {
        showMessage('Insufficient funds for this withdrawal.', true);
        return;
    }
    currentBalance -= amount;
    updateBalanceDisplay();
    showMessage(`Successfully withdrew ${amount.toLocaleString('en-US', { style: 'currency', currency: 'USD' })}.`);
    amountInput.value = "";
}

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

// --- Initial Call ---
// Initialize the display on page load.
updateBalanceDisplay();

</script>
</body>
```

```
</html>
```

EXP-3:

Develop an admin dashboard with CSS Grid layout and dynamic theme switching:

```
<!DOCTYPE html>

<!-- Default theme is light --&gt;

&lt;html lang="en" class="light"&gt;
  &lt;head&gt;
    &lt;meta charset="UTF-8"&gt;
    &lt;meta name="viewport" content="width=device-width, initial-scale=1.0"&gt;
    &lt;title&gt;Admin Dashboard&lt;/title&gt;
    &lt;script src="https://cdn.tailwindcss.com"&gt;&lt;/script&gt;
    &lt;link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;500;600;700&amp;display=swap" rel="stylesheet"&gt;
    &lt;!-- Lucide icons for a modern look --&gt;
    &lt;script src="https://unpkg.com/lucide@latest"&gt;&lt;/script&gt;
  &lt;style&gt;
    body {
      font-family: 'Inter', sans-serif;
    }
    /* Custom scrollbar for better aesthetics, especially in dark mode */
    ::-webkit-scrollbar {
      width: 8px;
    }
    ::-webkit-scrollbar-track {
      background-color: transparent;
    }
    ::-webkit-scrollbar-thumb {
      background-color: #a0aec0; /* gray-400 */
      border-radius: 10px;
    }
  &lt;/style&gt;
&lt;/head&gt;
&lt;body&gt;
  &lt;h1&gt;Welcome to Admin Dashboard&lt;/h1&gt;
  &lt;p&gt;This is a simple admin dashboard built with Tailwind CSS and dynamic theme switching.&lt;/p&gt;
  &lt;button&gt;Switch Theme&lt;/button&gt;
&lt;/body&gt;
&lt;/html&gt;</pre>
```

```

    border: 2px solid transparent;
    background-clip: content-box;
}

html.dark ::-webkit-scrollbar-thumb {
    background-color: #4a5568; /* gray-600 */
}

</style>

<script>
// Configure Tailwind CSS for dark mode toggling via a class
tailwind.config = {
    darkMode: 'class',
}
</script>

</head>

<body class="bg-gray-100 dark:bg-gray-900 text-gray-900 dark:text-gray-200 transition-colors duration-300">

    <!-- Main container using CSS Grid for the layout -->
    <div class="grid min-h-screen grid-cols-1 lg:grid-cols-[260px_1fr]">

        <!-- === Sidebar Navigation === -->
        <aside class="bg-white dark:bg-gray-800 shadow-lg hidden lg:flex lg:flex-col">
            <!-- Logo/Header -->
            <div class="p-6 flex items-center gap-3">
                <div class="p-2 bg-blue-600 rounded-lg">
                    <i data-lucide="command" class="text-white"></i>
                </div>
                <h1 class="text-xl font-bold">Dashboard</h1>
            </div>
        </aside>

        <!-- Navigation Links -->

```

```

<nav class="flex-1 px-4 py-2">

  <a href="#" class="flex items-center gap-3 px-4 py-3 text-gray-700 dark:text-gray-300 bg-blue-50 dark:bg-gray-700 rounded-lg font-semibold">
    <i data-lucide="layout-grid" class="w-5 h-5"></i> Overview
  </a>

  <a href="#" class="flex items-center gap-3 px-4 py-3 mt-2 text-gray-600 dark:text-gray-400 hover:bg-gray-100 dark:hover:bg-gray-700 rounded-lg">
    <i data-lucide="users" class="w-5 h-5"></i> Customers
  </a>

  <a href="#" class="flex items-center gap-3 px-4 py-3 mt-2 text-gray-600 dark:text-gray-400 hover:bg-gray-100 dark:hover:bg-gray-700 rounded-lg">
    <i data-lucide="shopping-cart" class="w-5 h-5"></i> Orders
  </a>

  <a href="#" class="flex items-center gap-3 px-4 py-3 mt-2 text-gray-600 dark:text-gray-400 hover:bg-gray-100 dark:hover:bg-gray-700 rounded-lg">
    <i data-lucide="bar-chart-3" class="w-5 h-5"></i> Analytics
  </a>
</nav>

<!-- Sidebar Footer -->

<div class="p-4 border-t dark:border-gray-700">
  <a href="#" class="flex items-center gap-3 px-4 py-3 text-gray-600 dark:text-gray-400 hover:bg-gray-100 dark:hover:bg-gray-700 rounded-lg">
    <i data-lucide="settings" class="w-5 h-5"></i> Settings
  </a>
</div>
</aside>

<!-- === Main Content Area === -->

<main class="flex flex-col">

  <!-- Header -->
  <header class="flex items-center justify-between p-4 bg-white dark:bg-gray-800 shadow-md border-b dark:border-gray-700">

```

```
<h2 class="text-2xl font-bold">Dashboard Overview</h2>
<div class="flex items-center gap-4">
  <!-- Theme Switcher Button -->
  <button id="theme-toggle" class="p-2 rounded-full text-gray-500 dark:text-gray-400
  hover:bg-gray-100 dark:hover:bg-gray-700 focus:outline-none focus:ring-2 focus:ring-blue-500">
    <i id="theme-icon-light" data-lucide="sun" class="hidden"></i>
    <i id="theme-icon-dark" data-lucide="moon" class="hidden"></i>
    <span class="sr-only">Toggle theme</span>
  </button>
  <!-- User Profile -->
  <div class="w-10 h-10 bg-gradient-to-br from-blue-400 to-indigo-500 rounded-full flex
  items-center justify-center text-white font-bold">
    A
  </div>
</div>
</header>

<!-- Page Content -->
<div class="flex-1 p-6 lg:p-8 bg-gray-50 dark:bg-gray-800/50">
  <!-- Stat Cards -->
  <div class="grid grid-cols-1 sm:grid-cols-2 xl:grid-cols-4 gap-6 mb-8">
    <div class="bg-white dark:bg-gray-800 p-6 rounded-xl shadow-sm">
      <h3 class="text-sm font-medium text-gray-500 dark:text-gray-400">Total Revenue</h3>
      <p class="text-3xl font-bold mt-2">$58,392.41</p>
    </div>
    <div class="bg-white dark:bg-gray-800 p-6 rounded-xl shadow-sm">
      <h3 class="text-sm font-medium text-gray-500 dark:text-gray-400">New Users</h3>
      <p class="text-3xl font-bold mt-2">+1,204</p>
    </div>
    <div class="bg-white dark:bg-gray-800 p-6 rounded-xl shadow-sm">
      <h3 class="text-sm font-medium text-gray-500 dark:text-gray-400">Conversion Rate</h3>
      <p class="text-3xl font-bold mt-2">12.5%</p>
    </div>
  </div>
</div>
```

```

</div>

<div class="bg-white dark:bg-gray-800 p-6 rounded-xl shadow-sm">
  <h3 class="text-sm font-medium text-gray-500 dark:text-gray-400">Support Tickets</h3>
  <p class="text-3xl font-bold mt-2">27</p>
</div>
</div>

<!-- Main Content Grid -->
<div class="grid grid-cols-1 xl:grid-cols-3 gap-6">
  <!-- Sales Chart -->
  <div class="xl:col-span-2 bg-white dark:bg-gray-800 p-6 rounded-xl shadow-sm">
    <h3 class="font-semibold text-lg mb-4">Sales Performance</h3>
    <div class="h-80 bg-gray-100 dark:bg-gray-700 rounded-lg flex items-center justify-center">
      <p class="text-gray-400">[Chart Data Would Go Here]</p>
    </div>
  </div>
  <!-- Recent Activity -->
  <div class="bg-white dark:bg-gray-800 p-6 rounded-xl shadow-sm">
    <h3 class="font-semibold text-lg mb-4">Recent Activity</h3>
    <ul class="space-y-5">
      <li class="flex items-center gap-4">
        <div class="w-10 h-10 bg-green-100 dark:bg-green-900 text-green-600 dark:text-green-400 rounded-full flex items-center justify-center"><i data-lucide="dollar-sign" class="w-5 h-5"></i></div>
        <p class="flex-1 text-sm">New order #8124 placed by J. Doe.</p>
      </li>
      <li class="flex items-center gap-4">
        <div class="w-10 h-10 bg-blue-100 dark:bg-blue-900 text-blue-600 dark:text-blue-400 rounded-full flex items-center justify-center"><i data-lucide="user-plus" class="w-5 h-5"></i></div>
        <p class="flex-1 text-sm">A new user, Alex R., has registered.</p>
      </li>
    </ul>
  </div>
</div>

```

```
<li class="flex items-center gap-4">
    <div class="w-10 h-10 bg-red-100 dark:bg-red-900 text-red-600 dark:text-red-400 rounded-full flex items-center justify-center"><i data-lucide="alert-circle" class="w-5 h-5"></i></div>
    <p class="flex-1 text-sm">Server #3 is reporting high CPU usage.</p>
</li>
</ul>
</div>
</div>
</main>
</div>

<script>
// --- Initialize Lucide Icons ---
lucide.createIcons();

// --- Theme Switching Logic ---
const themeToggleBtn = document.getElementById('theme-toggle');
const lightIcon = document.getElementById('theme-icon-light');
const darkIcon = document.getElementById('theme-icon-dark');
const htmlEl = document.documentElement;

// Function to set the theme and update the icon
function setTheme(theme) {
    if (theme === 'dark') {
        htmlEl.classList.add('dark');
        lightIcon.classList.add('hidden');
        darkIcon.classList.remove('hidden');
    } else {
        htmlEl.classList.remove('dark');
        darkIcon.classList.add('hidden');
    }
}

setTheme('light');
themeToggleBtn.addEventListener('click', () => setTheme((htmlEl.classList.contains('dark')) ? 'light' : 'dark'));

```

```

        lightIcon.classList.remove('hidden');

    }

}

// Toggle theme on button click
themeToggleBtn.addEventListener('click', () => {
    const currentTheme = htmlEl.classList.contains('dark') ? 'dark' : 'light';
    const newTheme = currentTheme === 'dark' ? 'light' : 'dark';
    localStorage.setItem('theme', newTheme);
    setTheme(newTheme);
});

// Apply theme on initial load
document.addEventListener('DOMContentLoaded', () => {
    const savedTheme = localStorage.getItem('theme');
    const prefersDark = window.matchMedia('(prefers-color-scheme: dark)').matches;

    if (savedTheme) {
        setTheme(savedTheme);
    } else if (prefersDark) {
        setTheme('dark');
    } else {
        setTheme('light');
    }
});

</script>
</body>
</html>

```

OUTPUTS:

EXP-1:

Register for a Course

Let's get you started

Full Name

e.g., Alex Doe

Email Address

you@example.com

Age

Enter your age

Register Now

Figure 1 Registration

Registration Successful!

x

Welcome, Alex! Your registration for the course is confirmed – we're so glad to have you join us! We're excited for you to start and are here to support you.

Finish

Figure 2 Registration Successful

EXP-2:

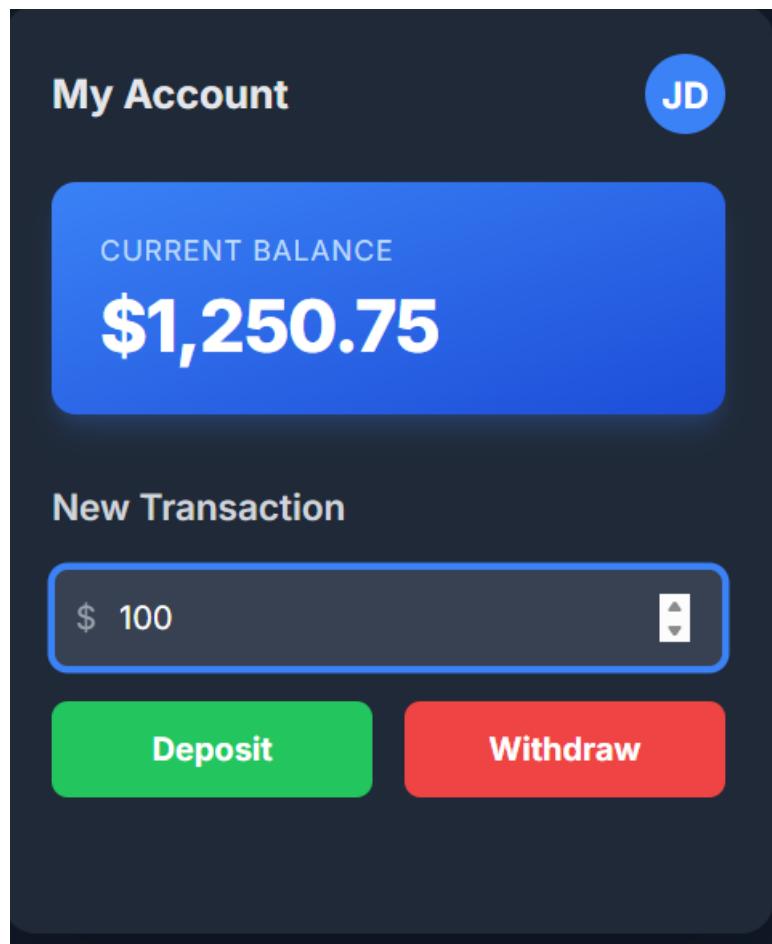


Figure 3 Banking Ui

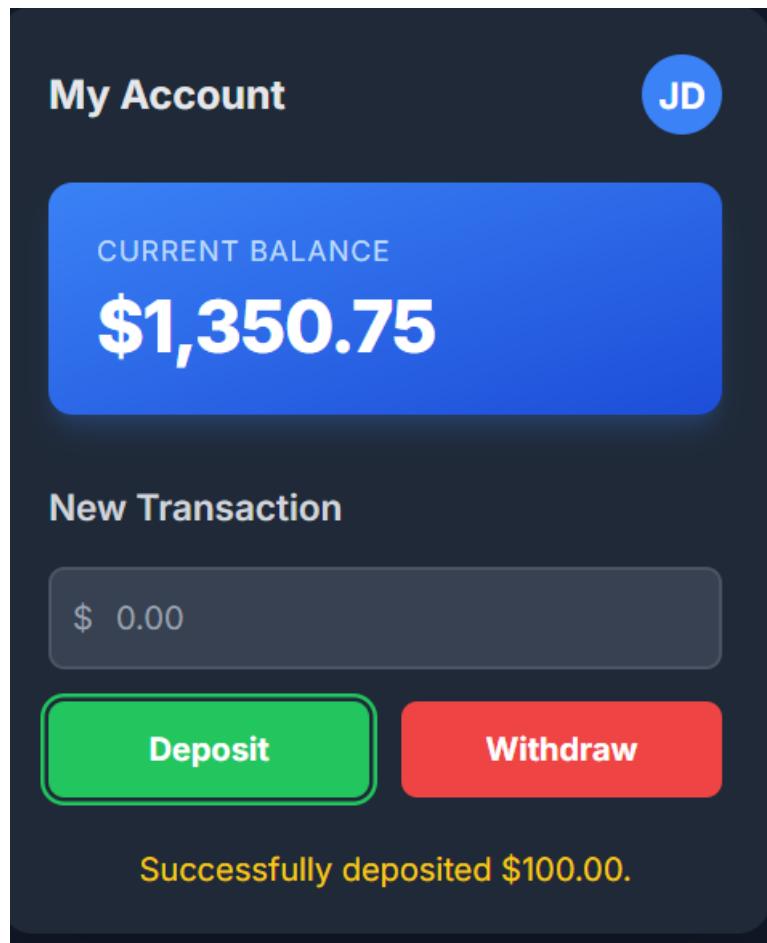


Figure 4 Deposit Successful

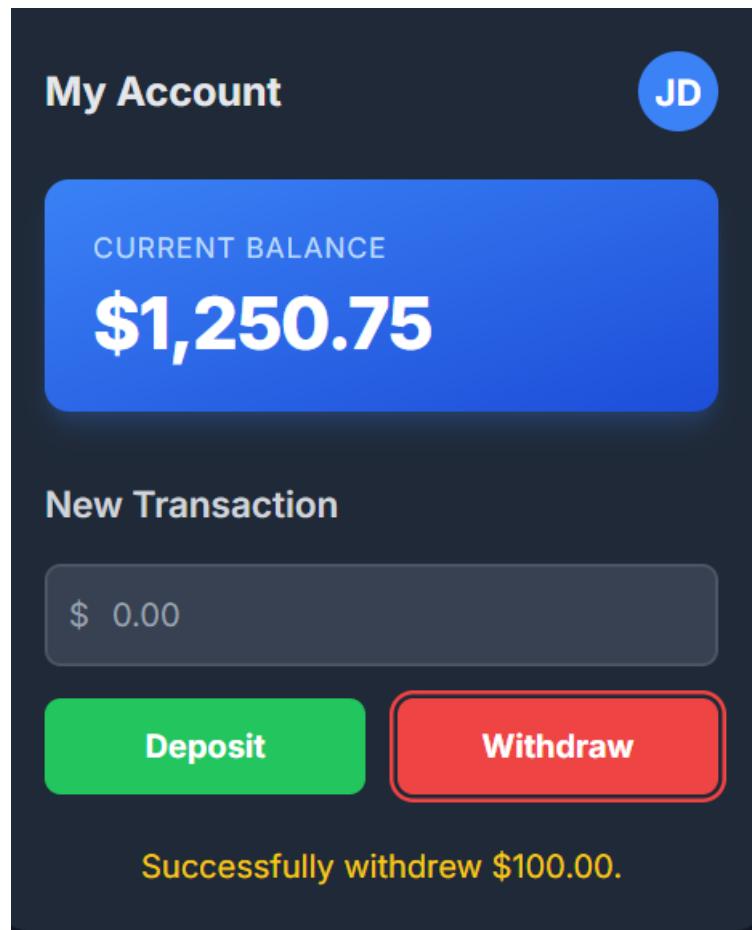


Figure 5 Withdraw Successful

EXP-3:

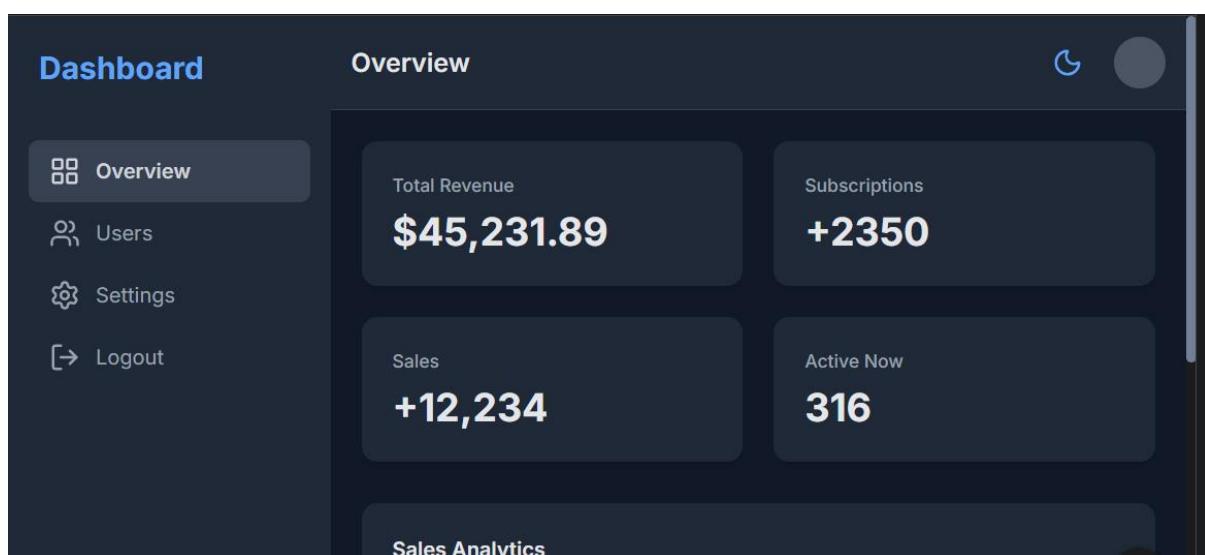


Figure 6 Dark Mode Dashboard

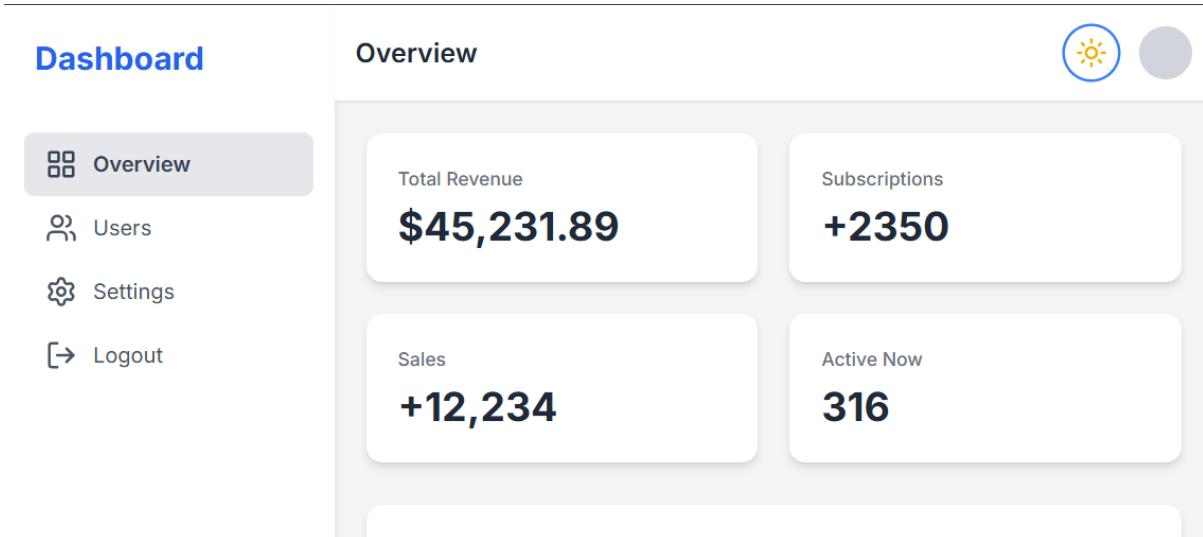


Figure 7 Light Mode Dashboard