Experiment 1

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

CODE:

```html
<!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>
  <style>
    body {
      font-family: sans-serif;
      background-color: #f4f4f4;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
      margin: 0;
    }
    .form-container {
      background: white;
      padding: 2em;
      border-radius: 8px;
      box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
      width: 100%;
      max-width: 400px;
    }
    h2 {
      text-align: center;
      color: #333;
    }
    .form-group {
      margin-bottom: 1em;
    }
    label {
      display: block;
      margin-bottom: 0.5em;
      font-weight: bold;
      color: #555;
    }
    input[type="text"],
    input[type="email"],
    input[type="number"] {
      width: 100%;
      padding: 0.8em;
      box-sizing: border-box;
      border: 1px solid #ccc;
      border-radius: 4px;
      font-size: 1em;
    }
    input:focus {
      border-color: #007BFF;
      outline: none;
      box-shadow: 0 0 5px rgba(0, 123, 255, 0.25);
    }
    button {
```

```
            width: 100%;
            padding: 0.8em;
            background-color: #007BFF;
            color: white;
            border: none;
            border-radius: 4px;
            font-size: 1.1em;
            cursor: pointer;
            transition: background-color 0.3s ease;
        }
        button:hover {
            background-color: #0056b3;
        }
        /* Style for HTML5 validation messages */
        input:invalid:not(:placeholder-shown) {
            border-color: #dc3545;
        }
        input:invalid:not(:placeholder-shown) + span::before {
            content: "Please enter a valid ";
            color: #dc3545;
            font-size: 0.8em;
            display: block;
            margin-top: 5px;
        }
    </style>
</head>
<body>

<div class="form-container">
    <h2>Student Registration</h2>
    <form action="#" method="post">

        <div class="form-group">
            <label for="name">Full Name:</label>
            <input type="text" id="name" name="name" required
                pattern="[A-Za-z\s]{3,}"
                title="Name must contain at least 3 letters and only alphabets and spaces."
                placeholder="Enter your full name">
        </div>

        <div class="form-group">
            <label for="email">Email Address:</label>
            <input type="email" id="email" name="email" required
                placeholder="Enter a valid email address">
        </div>

        <div class="form-group">
            <label for="age">Age:</label>
            <input type="number" id="age" name="age" required
                min="16" max="99"
                title="Age must be between 16 and 99."
                placeholder="Enter your age (16-99)">
        </div>

        <button type="submit">Register</button>

    </form>
</div>

</body>
</html>
```

Experiment 2

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

I'll create a responsive banking UI with deposit/withdrawal buttons and a balance display using HTML, CSS, and JavaScript. The UI will be simple and mobile-friendly, demonstrating basic functionality.

CODE:

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Simple Banking UI</title>
    <style>
        :root {
            --primary-color: #4CAF50;
            --secondary-color: #f44336;
            --background-color: #f0f2f5;
            --card-background: #ffffff;
            --text-color: #333;
            --light-text: #666;
            --border-radius: 8px;
            --box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
        }

        * {
            box-sizing: border-box;
            margin: 0;
            padding: 0;
        }

        body {
            font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue', sans-serif;
            background-color: var(--background-color);
            display: flex;
            justify-content: center;
            align-items: center;
            min-height: 100vh;
            color: var(--text-color);
            padding: 1rem;
        }

        .container {
            width: 100%;
            max-width: 500px;
            background: var(--card-background);
            border-radius: var(--border-radius);
            box-shadow: var(--box-shadow);
            padding: 2rem;
            text-align: center;
        }

        h1 {
            color: var(--text-color);
            margin-bottom: 0.5rem;
```

```css
        }

        .balance-display {
            background-color: var(--primary-color);
            color: white;
            padding: 1.5rem;
            border-radius: var(--border-radius);
            margin: 1.5rem 0;
        }

        .balance-label {
            font-size: 1rem;
            margin-bottom: 0.5rem;
            text-transform: uppercase;
            letter-spacing: 1px;
            font-weight: 500;
        }

        #balance {
            font-size: 2.5rem;
            font-weight: bold;
            display: block;
        }

        .transaction-form {
            display: flex;
            flex-direction: column;
            gap: 1rem;
            margin-top: 1.5rem;
        }

        .transaction-form input {
            width: 100%;
            padding: 0.8rem;
            border: 1px solid #ddd;
            border-radius: var(--border-radius);
            font-size: 1rem;
            transition: border-color 0.3s;
        }

        .transaction-form input:focus {
            outline: none;
            border-color: var(--primary-color);
        }

        .button-group {
            display: flex;
            gap: 1rem;
            flex-wrap: wrap;
        }

        .button-group button {
            flex: 1 1 150px; /* Allows buttons to grow and wrap */
            padding: 0.8rem;
            border: none;
            border-radius: var(--border-radius);
            font-size: 1rem;
            font-weight: bold;
            cursor: pointer;
            transition: background-color 0.3s ease;
        }

        #depositBtn {
            background-color: var(--primary-color);
            color: white;
```

```
            }

            #depositBtn:hover {
                background-color: #43a047;
            }

            #withdrawBtn {
                background-color: var(--secondary-color);
                color: white;
            }

            #withdrawBtn:hover {
                background-color: #e53935;
            }

            @media (max-width: 480px) {
                .container {
                    padding: 1.5rem;
                }

                .button-group {
                    flex-direction: column;
                }

                #balance {
                    font-size: 2rem;
                }
            }
        </style>
    </head>
    <body>

        <div class="container">
            <h1>My Bank Account</h1>

            <div class="balance-display">
                <span class="balance-label">Current Balance</span>
                <span id="balance">$1000.00</span>
            </div>

            <div class="transaction-form">
                <input type="number" id="amountInput" placeholder="Enter amount" min="1" step="0.01">
                <div class="button-group">
                    <button id="depositBtn">Deposit</button>
                    <button id="withdrawBtn">Withdraw</button>
                </div>
            </div>
        </div>

        <script>
            const balanceSpan = document.getElementById('balance');
            const amountInput = document.getElementById('amountInput');
            const depositBtn = document.getElementById('depositBtn');
            const withdrawBtn = document.getElementById('withdrawBtn');

            let currentBalance = 1000.00;

            function updateBalance(newBalance) {
                balanceSpan.textContent = `$${newBalance.toFixed(2)}`;
            }

            depositBtn.addEventListener('click', () => {
                const amount = parseFloat(amountInput.value);
                if (amount > 0) {
                    currentBalance += amount;
```

```
          updateBalance(currentBalance);
          amountInput.value = '';
        } else {
          alert('Please enter a valid amount to deposit.');
        }
      });

      withdrawBtn.addEventListener('click', () => {
        const amount = parseFloat(amountInput.value);
        if (amount > 0 && amount <= currentBalance) {
          currentBalance -= amount;
          updateBalance(currentBalance);
          amountInput.value = '';
        } else if (amount > currentBalance) {
          alert('Insufficient funds.');
        } else {
          alert('Please enter a valid amount to withdraw.');
        }
      });

      // Initial balance display
      updateBalance(currentBalance);
  </script>

</body>
</html>
```

Experiment 3

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

CODE:

HTML CODE:

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Admin Dashboard</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>

  <div class="dashboard" id="dashboard">
    <header class="header">
      <h2>Dashboard</h2>
      <div class="theme-switcher">
        <button id="theme-toggle">Switch Theme</button>
      </div>
    </header>

    <aside class="sidebar">
      <nav>
        <ul>
          <li><a href="#">Overview</a></li>
          <li><a href="#">Users</a></li>
          <li><a href="#">Products</a></li>
          <li><a href="#">Settings</a></li>
        </ul>
      </nav>
    </aside>

    <main class="main-content">
      <div class="card">
        <h3>Total Sales</h3>
        <p>$12,450</p>
      </div>
      <div class="card">
        <h3>New Users</h3>
        <p>1,234</p>
      </div>
      <div class="card">
        <h3>Active Projects</h3>
        <p>25</p>
      </div>
      <div class="card">
        <h3>Support Tickets</h3>
        <p>8</p>
      </div>
    </main>
  </div>

  <script src="script.js"></script>
</body>
</html>
```

CSS CODE:

```css
:root {
  --bg-color: #f0f2f5;
  --text-color: #333;
  --card-bg: #fff;
  --sidebar-bg: #2c3e50;
  --sidebar-text: #ecf0f1;
  --header-bg: #fff;
  --header-border: #ddd;
  --accent-color: #3498db;
  --box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
}

:root.dark-theme {
  --bg-color: #121212;
  --text-color: #e0e0e0;
  --card-bg: #1e1e1e;
  --sidebar-bg: #212121;
  --sidebar-text: #b0b0b0;
  --header-bg: #1e1e1e;
  --header-border: #333;
  --accent-color: #8e44ad;
  --box-shadow: 0 2px 4px rgba(0, 0, 0, 0.4);
}

body {
  font-family: sans-serif;
  margin: 0;
  background-color: var(--bg-color);
  color: var(--text-color);
  transition: background-color 0.3s, color 0.3s;
}

.dashboard {
  display: grid;
  grid-template-columns: 250px 1fr;
  grid-template-rows: auto 1fr;
  height: 100vh;
}

.header {
  grid-column: 2 / 3;
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 1rem 2rem;
  background-color: var(--header-bg);
  border-bottom: 1px solid var(--header-border);
  box-shadow: var(--box-shadow);
}

.sidebar {
  grid-row: 1 / 3;
  background-color: var(--sidebar-bg);
  color: var(--sidebar-text);
  padding: 1rem;
  box-shadow: var(--box-shadow);
}

.sidebar ul {
  list-style: none;
  padding: 0;
}
```

```css
.sidebar li a {
  display: block;
  padding: 0.8rem 1rem;
  color: var(--sidebar-text);
  text-decoration: none;
  transition: background-color 0.2s;
}

.sidebar li a:hover {
  background-color: var(--accent-color);
}

.main-content {
  padding: 2rem;
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
  gap: 1.5rem;
  overflow-y: auto;
}

.card {
  background-color: var(--card-bg);
  padding: 1.5rem;
  border-radius: 8px;
  box-shadow: var(--box-shadow);
  transition: transform 0.2s;
}

.card:hover {
  transform: translateY(-5px);
}

button {
  padding: 0.5rem 1rem;
  background-color: var(--accent-color);
  color: white;
  border: none;
  border-radius: 4px;
  cursor: pointer;
  transition: background-color 0.2s;
}

button:hover {
  background-color: #2980b9;
}

@media (max-width: 768px) {
  .dashboard {
    grid-template-columns: 1fr;
    grid-template-rows: auto auto 1fr;
  }

  .header {
    grid-column: 1 / 2;
  }

  .sidebar {
    grid-row: 2 / 3;
    grid-column: 1 / 2;
    padding: 0;
  }

  .sidebar nav ul {
    display: flex;
    justify-content: space-around;
```

```css
        padding: 0.5rem;
    }

    .main-content {
        grid-column: 1 / 2;
    }
}
```
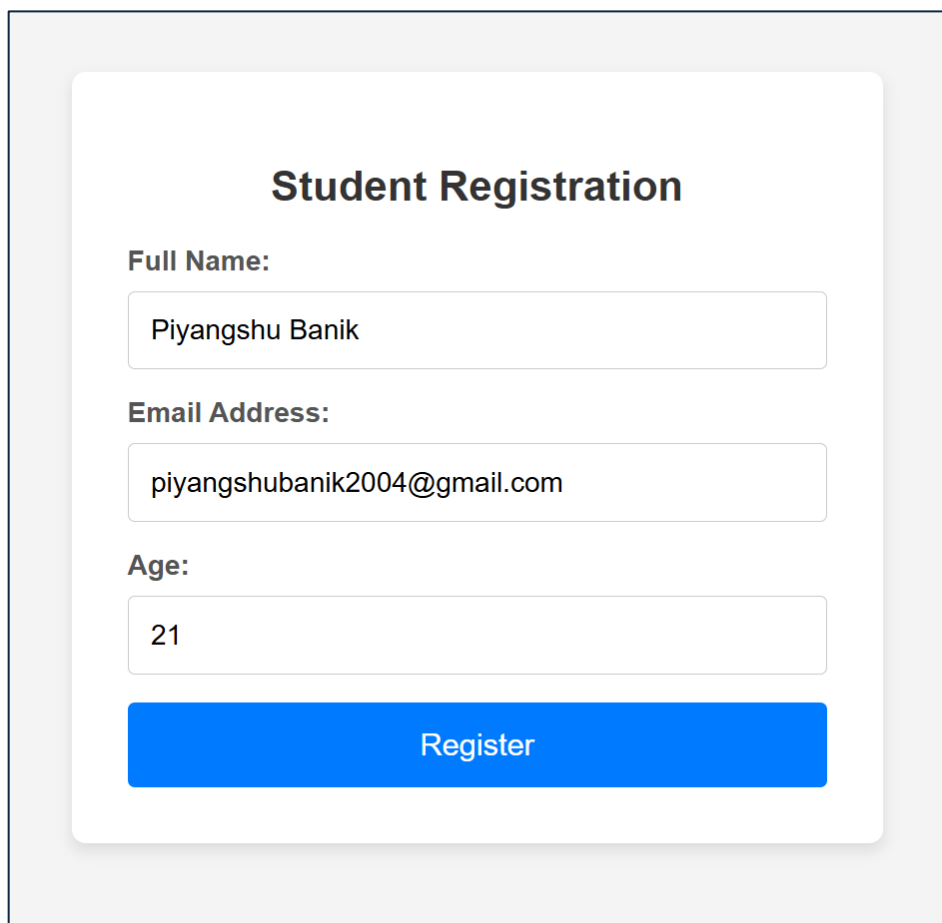
JAVASCRIPT CODE:

```javascript
const themeToggleBtn = document.getElementById('theme-toggle');
const body = document.body;
const currentTheme = localStorage.getItem('theme');

if (currentTheme) {
    body.classList.add(currentTheme);
}

themeToggleBtn.addEventListener('click', () => {
    body.classList.toggle('dark-theme');
    const newTheme = body.classList.contains('dark-theme') ? 'dark-theme' : '';
    localStorage.setItem('theme', newTheme);
});
```

OUTPUTS
EXPERIMENT 1

Figure 1: OUTPUT OF EXPERIMENT 1

EXPERIMENT 2

Figure 2: OUTPUT OF EXPERIMENT 2 part a

Figure 3: OUTPUT OF EXPERIMENT 2 part b

EXPERIMENT 3

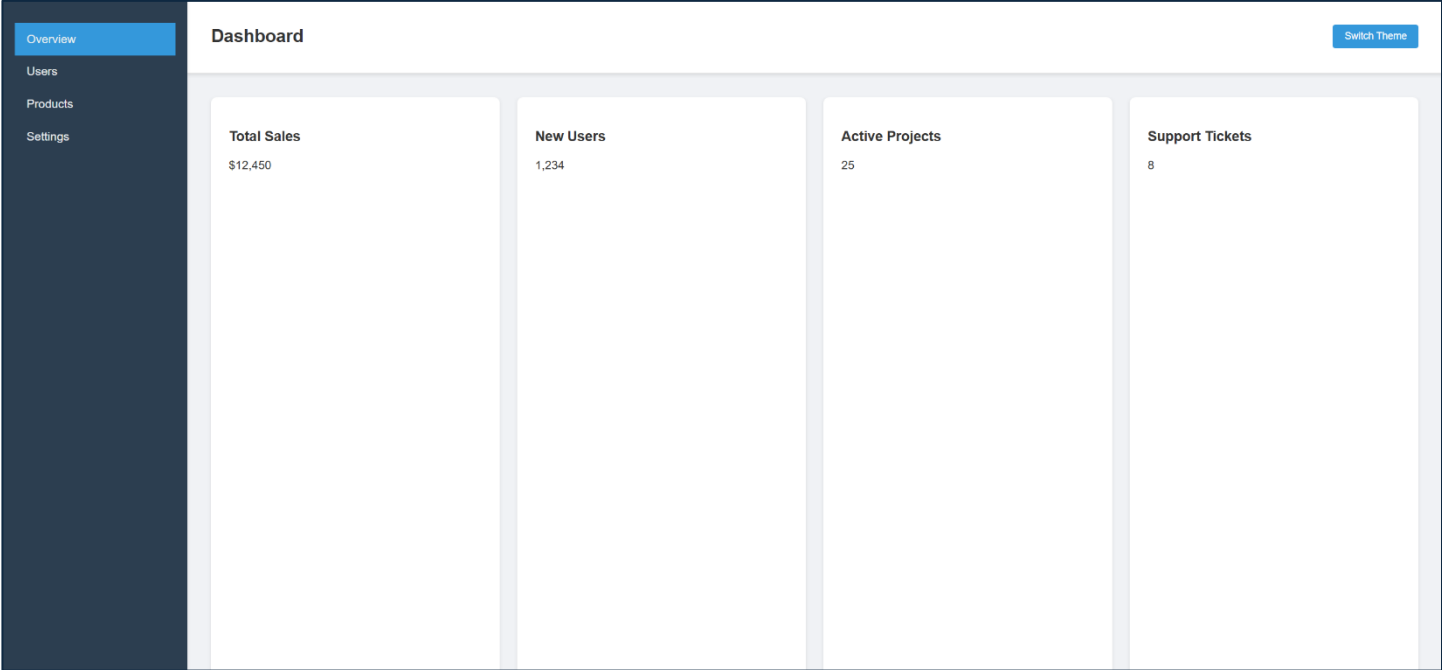| Overview	**Dashboard**			Switch Theme
Users				
Products				
Settings	**Total Sales**	**New Users**	**Active Projects**	**Support Tickets**
	$12,450	1,234	25	8

Figure 4: OUTPUT OF EXPERIMENT 3

	Total Sales	**New Users**	**Active Projects**	**Support Tickets**
	$12,450	1,234	25	8