



## **WEB Technologies LAB EXERCISE-HTML [CSS]**

### **1. Create a Personal Introduction Webpage using All Three CSS Styles**

**Objective:** Learn to differentiate between inline, internal, and external CSS and understand their scope and priority.

**Task:**

Build a simple webpage that includes your name, photograph, and a short paragraph about yourself.

- Use **inline CSS** to style your name (change color, font size).
- Use **internal CSS** (within `<style>` tag) to style the paragraph and background.
- Create a separate `.css` file and use **external CSS** to style the image and layout.

**Hint:**

Try giving the paragraph a different font using internal CSS and add a border to the image from the external CSS file.

#### Sample Output

**My Name is Sandeep**



Hello! I am currently doing research in EEG signal analysis and cognitive neuroscience. I enjoy coding, teaching, and learning new technologies.

Contact me at: sandeep@example.com

---

### **2. Design a Color-Themed Landing Page Using CSS Colors**

**Objective:** Practice working with various color models in CSS.

**Task:**

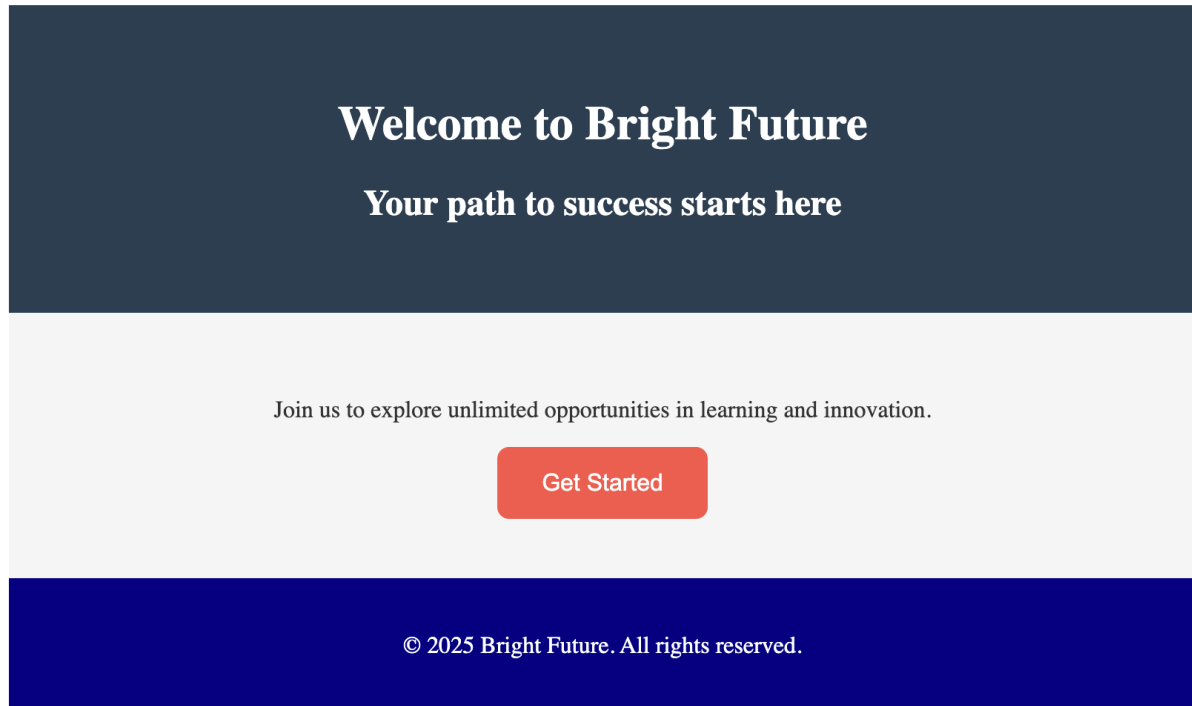
Design a small homepage layout with a heading, subheading, and a button.

- Use **hexadecimal**, **RGB**, and **named colors** in different sections.
- Assign different background colors to header, body, and footer.
- Make the button visually pop using color contrast.

**Hint:**

Try using `color:`, `background-color:`, and experiment with `rgba()` to add transparency.

**Sample Output**



### 3. Format an Article Page Using Class and ID Selectors

**Objective:** Use class and ID selectors to organize and reuse styles efficiently.

**Task:**

Write a simple blog post with a title, author info, multiple paragraphs, and a "Read more" button.

- Use an **ID selector** to uniquely style the title (e.g., bigger font, underline).
- Use **class selectors** for styling all paragraphs consistently.
- Create a class for the button and apply hover styling.

**Hint:**

ID starts with #, class starts with . — use them both in your CSS file. IDs are unique; classes can be reused.

**Sample Output**

## The Future of Web Development

By Jane Doe | August 1, 2025

Web development is constantly evolving with new frameworks, libraries, and tools emerging every year.

One of the biggest trends today is the rise of component-based frameworks like React, Vue, and Angular.

These frameworks allow developers to build dynamic, responsive interfaces more efficiently than ever before.

[Read More](#)

---

## 4. Visualize the CSS Box Model with Colored Sections

**Objective:** Understand the impact of margin, padding, border, and content.

**Task:**

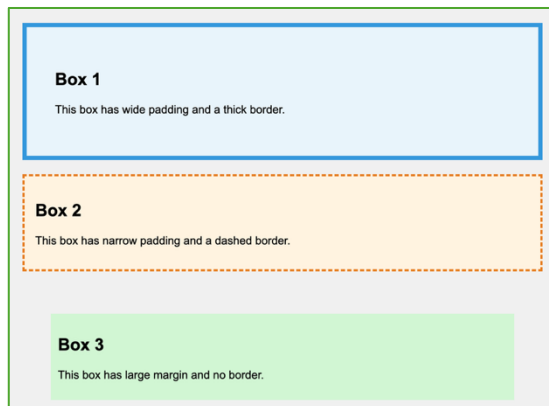
Design a simple layout with 3 different boxes showing margin, padding, and borders clearly. Each box should contain:

- A heading
- Some content
- Visually distinct spacing and border styles

**Hint:**

Use background colors to clearly differentiate between margin, border, and padding.

### Sample Output



---

## 5. Design a Stylish Profile Card with Shadow and Rounded Corners

**Objective:** Build a reusable UI card with modern CSS properties.

**Task:**

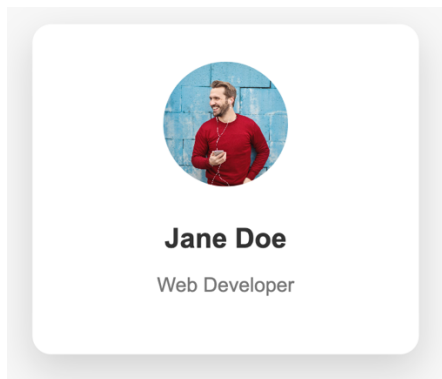
Create a profile card with an image, name, and job title.

- Add `border-radius` to make the image circular
- Use `box-shadow` to give the card depth
- Add a subtle `transition` effect when hovering over the card

**Hint:**

Use `transition: all 0.3s ease-in-out;` and `transform: scale(1.05)` on hover.

**Sample Output**



---

## 6. Arrange Multiple Elements Using CSS Positioning

**Objective:** Learn the difference between `static`, `relative`, `absolute`, and `fixed` positions.

**Task:**

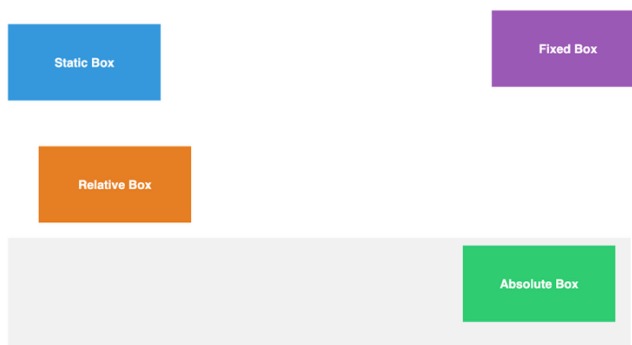
Create a page with three colored boxes placed at specific positions:

- One should scroll normally (`static`)
- One should stay fixed at the top right (`fixed`)
- One should move relative to its container (`relative`)
- One placed absolutely inside a wrapper (`absolute`)

**Hint:**

Use `top`, `left`, and `right` to place positioned elements. Use `z-index` if elements overlap.

**Sample Output**



## 7. Build a Responsive Navigation Bar with Flexbox and Hover Effects

**Objective:** Create a responsive layout and enhance interactivity using pseudo-classes.

**Task:**

Create a navigation bar with 4–5 links (e.g., Home, About, Services, Contact).

- Use **Flexbox** to distribute them evenly
- Apply **:hover** effects (color change, underline, bold)
- Ensure it wraps nicely on smaller screens

**Hint:**

Use `display: flex, justify-content: space-around, and @media queries` for responsiveness.

**Sample Output**



## 8. Create a Blog Post Layout with Multi-Column Content

**Objective:** Use modern layout techniques like multi-column and overflow handling.

**Task:**

Design a blog post with:

- Multi-column text (split into 2 or 3 columns)
- A sidebar that remains scrollable
- Long content that demonstrates vertical scrolling with a fixed header

**Hint:**

Use `column-count, overflow, and height with overflow: auto` to create scrollable areas.

**Sample Output**



## My Tech Blog

### Sidebar

- Recent Posts
- Archives
- Tags
- Contact

### Understanding CSS Columns

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce varius, sapien sed mattis blandit, est libero vehicula sapien, non dignissim mi nisl sit amet leo. Proin at neque nec lacus iaculis commodo. Integer efficitur nisl vitae lorem laoreet, nec imperdiet sapien aliquam. Curabitur at mi nisi. Curabitur rhoncus erat ut erat luctus viverra. Sed tempor malesuada sapien.

Sed varius lacus justo, a vulputate sapien condimentum id. Nullam porttitor, nisl nec porttitor suscipit, elit dolor mattis risus, ut iaculis erat magna in velit. Donec nec libero in erat consectetur tincidunt. Duis volutpat mauris ac finibus rhoncus. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae.

## 9. Animate Buttons Using Transitions and Transformations

**Objective:** Introduce smooth transitions and CSS-based animations.

### Task:

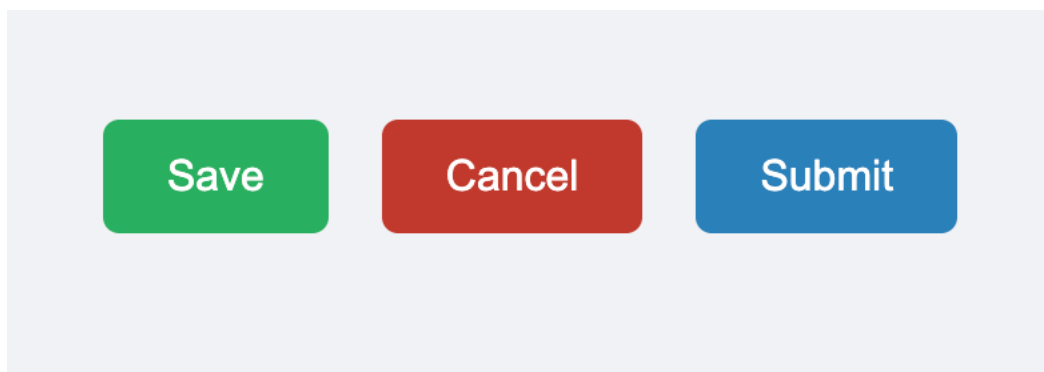
Create a set of buttons (e.g., Save, Cancel, Submit).

- On hover: change background color, scale up slightly
- On click: apply a temporary box-shadow
- Use transitions for smooth effects

### Hint:

Use `:hover`, `:active`, `transform`, and `transition` properties.

### Sample Output



## 10. Design a Web Page Using CSS3 Effects (Gradients, Web Fonts, Opacity)

**Objective:** Combine multiple modern CSS3 properties in a visually appealing page.

### Task:

Create a one-page portfolio layout with:

- Gradient background header
- Web fonts from Google Fonts
- Section with partially transparent images (`opacity`)
- Buttons with text effects like shadows or letter-spacing

**Hint:**

Use `background: linear-gradient(...)`, `@import` for fonts, and `text-shadow`, `letter-spacing`.

**Sample Output**

