Week 3: Deep Dive into CSS

- **Day 1:Box Model**

- Understand the CSS box model: content, padding, border, margin.

1) **Content** – It refers to the area where your actual text, images, or other media are displayed. The box model consists of several components that determine the size and structure of an element.

2) **Padding** – Padding refers to the space between the content of an element and its border.

3) **Border** - The border is a line that surrounds the padding and content of an element. It is positioned between the element's padding and margin.

4) **Margin** – Margin refers to the space between the border of two element or div. and difference between outside the border.

- Practice adjusting these properties.

- **Day 2: Display and Positioning**

- Learn about display property values like block, inline, inline-block, none.

1) **Block** - The Block property specifies the display behavior (the type of rendering box) of an element. The `block` display property value makes an element behave as a block-level element.

2) **Inline** – The Inline property value makes an element behave as an inline element. Inline elements are a key part of HTML and CSS, and they have specific behaviors and characteristics.

3) **Inline-block** – The inline-block display property value allows an element to be formatted as an inline element while retaining the characteristics of a block-level element

- Introduction to position property: static, relative, absolute, fixed.

1) **Static** – This property is used to specify how an element is positioned in a document. The 'static' value is the default value for the 'position' property, meaning an element is positioned according to the normal flow of the document.

2) **Relative** – The Relative property value positions an element relative to its normal position in the document flow.

3) **Absolute** – The Absolute property value removes an element from the normal document flow and positions it relative to its nearest positioned ancestor.

- **Day 3:Flexbox Basics**

- Learn about Flexbox and its properties.

.**What is Flexbox** - is a CSS layout model designed to provide an efficient way to lay out, align, and distribute space among items in a container, even when their size is unknown or dynamic. It allows for more flexible and responsive design, especially for web layouts.

.**Properties** – 1) **Flex direction** – It is a property that defines the direction in which flex items are placed in the flex container.

2) **Justify-content** – This property is used to align flex items along the main axis of the flex container.

3) **Align-item** – This property is used to align a flex container's lines within the flex container when there is extra space in the cross-axis, which is perpendicular to the main axis.

4) **Align-content** – This property controls how multiple lines of flex items are aligned within a flex container along the cross-axis, which is perpendicular to the main axis.

- Create a basic layout using Flexbox.

- **Day 4: Responsive Design**

- Understand the importance of responsive design.

.**What is responsive design** - Responsive design in CSS is crucial for creating websites and applications that adapt and function well across various devices and screen sizes.

- Introduction to media queries.

.**What is Media Query** – A media query is a feature that allows you to apply different styles to a document based on various attributes of the device on which the content is being displayed. Media queries enable responsive design by checking conditions such as screen width, device orientation, resolution, and more.

- **Day 5: CSS Grid Basics**

- Learn the fundamentals of CSS Grid.

.**What is CSS Grid** – CSS Grid is a powerful layout system that allows developers to design web pages in a two-dimensional grid format. It provides precise control over the layout of elements, enabling complex designs that were previously difficult to achieve with traditional CSS methods like floats and positioning.

- Create a basic layout using CSS Grid.

❖ Day 6:CSS Transitions and Animations

- Introduction to CSS transitions and animations.

. **What is Transition** – Transition allow you to smoothly change the property values of an element over a specified duration. This creates visual effects such as fading, sliding, or scaling elements in response to user interactions like hover or click events. Transitions make web interfaces more interactive and engaging without the need for JavaScript animations.

. **What is Animation** – Animation allow you to create more complex and dynamic visual effects compared to transitions. While transitions only enable property changes from one state to another, animations provide greater control over how elements move, change, and behave over time. CSS animations can animate multiple properties simultaneously and offer more advanced timing and looping options.

- Create simple animations using CSS.