Day 12 of training

Animations with Transitions and Transforms

1. **Introduction to CSS Transitions**
    - Today's lesson was about making elements move and change smoothly on the screen using **CSS Transitions**. Transitions allow you to define how an element changes from one state to another over a period of time, rather than instantly.
    - **States:** Elements can exist in different states, such as a normal state, a :hover state (when the mouse is over it), or an :active state (when it's being clicked). Transitions define the animation between these states.
    - **transition-property:** Specifies the CSS property (or properties) to which a transition effect should be applied (e.g., background-color, font-size, all).
    - **transition-duration:** Defines how long the transition animation should take to complete (e.g., 2s for 2 seconds, 500ms for 500 milliseconds).
    - **transition-timing-function:** Specifies the speed curve of the transition effect. Common values include ease, linear, ease-in (starts slow, then fast), ease-out (starts fast, then slow), ease-in-out (slow start and end), and steps() (transition in discrete steps).
    - **transition-delay:** Defines when the transition effect will start. A value of 1s means the transition will wait 1 second after the event before starting.
    - **transition Shorthand:** All these properties can be combined into a single transition shorthand property for conciseness (e.g., transition: all 2s ease-in 0.5s;).
2. **CSS Transforms: Moving and Reshaping Elements**
    - **Transforms** allow you to visually manipulate elements in 2D or 3D space. They enable powerful effects like rotation, scaling, and movement.
    - **rotate():** Rotates an element around a fixed point (its origin). You specify the degree of rotation (e.g., rotate(45deg) for a 45-degree rotation). You can also specify rotation around x, y, or z axes (rotateX(), rotateY(), rotateZ()).

- **scale():** Increases or decreases the size of an element. A value of 2 doubles the size, 0.5 halves it. You can scale uniformly or specify different scaling factors for the X and Y axes (scaleX(), scaleY()).
- **translate():** Moves an element from its original position. You can specify movement along the X-axis, Y-axis, or both. For example, translate(50px, 100px) moves it 50px right and 100px down.
- **skew():** Skews an element along the X and Y axes, giving it a slanted appearance. You specify the degree of skew (e.g., skew(30deg)).

3. **z-index for Overlapping Elements**
    - When elements are positioned using absolute, relative, or fixed positioning, they can overlap. The z-index property determines the stacking order of these overlapping elements.
    - Elements with a higher z-index value are placed on top of elements with lower values. If z-index is not specified, elements are stacked according to their order in the HTML document (later elements appear on top).