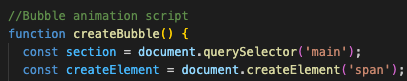
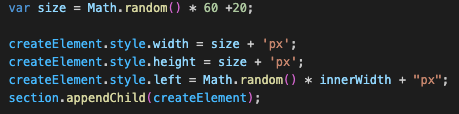
**Booking Form Bubble Animation Process**

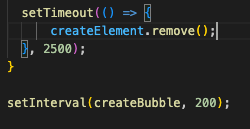
I firstly created a createBubble function, which retrieved the <main> element of the HTML file and created a <span> element.

I then added var size = Math.random() \* 60 + 20 to ensure max and min size of bubbles. Each time the code is run, the sizes will change. createElement.style.width & createElement.style.height are set to size + ‘px’ to

Generate random bubble sizes. createElement.style.left sets the horizontal position. It generates a random number between 0 & the width of the browser window so bubbles appear at a random horizontal position on the screen.

Section.appendChild(createElement) adds the <span> (bubble) element to the <main> element in the DOM, meaning the bubble is added to the page.

The final piece of JavaScript sets a timer so the bubble is removed after 2.5 seconds. The arrow function then runs after the delay which calls createElement.remove which removes the bubble from the DOM after 2.5 seconds.

setInterval(createBubble, 200) sets an interval that consistently calls the createBubble function every 200 milliseconds. A new bubble is created every 0.2 seconds and then will disappear after 2.5 seconds, so bubbles are constantly created after disappearing.

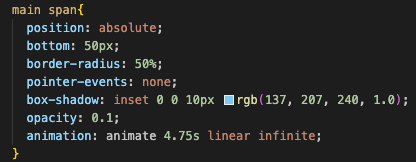
Although the javascript is complete at this point, the bubbles had to be styled in CSS in order to be visible

**CSS Styling For Bubbles**

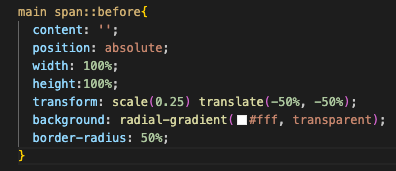
The below CSS styles the span elements (bubbles) that are created by the javascript code.

I positioned the bubbles relative to the <main> container and ensured they were placed 50px from the bottom of the container. Border-radius makes the bubble circular. Pointer-events:none ensures that the pointer does not interfere with the animation.

I included a box-shadow effect to make the bubbles look like they have a glow/shine. The opacity of 0.1 makes the bubble semi-transparent at the beginning of the animation.

Animation: animate 4.75s linear infinite moves the bubbles upwards and lasts 4.75 seconds. This process is repeated indefinitely.

I then used a ::before pseudo element to insert another circle into the bubble to give it a shine effect.

Again, I set the position as absolute to match the main span and this allowed me to place the pseudo element at specific coordinates inside the span (bubble). I set the width & height of the pseudo element to 100% to cover the entire area of the <span>. Scale 0.25 reduces the pseudo element to 25% of it’s original size, while translate (-50%, -50%) re-centers the element.

Finally, I added @keyframes. This rule defines how the bubbles should be animated over time.

It creates an animation (animate) that has already been applied in previous CSS rules. 0% indicates the start of the animation and transform: translateY(500%) mean that the bubble starts 5 times its own height below its original position. Opacity 1 means the bubble is fully visible.

Opacity 1 at 90% ensures the bubble remains fully visible up to this point.

Transform: translateY(-1700%) allows the bubble to move upwards by 1700% of its own height. As the value is negative, it moves the bubble out of view which gives the effect of it floating away. Opacity is still set to 1.



