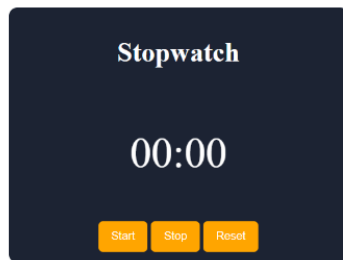


Problem statement:

With your knowledge gained on JS Timing Events and call-back design a Minute stopwatch with the time display that displays minute and seconds and having a button to start, stop and reset the time.



Solution:

Html Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Stopwatch</title>
  <!-- css file -->
  <link rel="stylesheet" href="stopwatch.css"/>
</head>
<body>
  <!-- main container -->
  <div class="container">
    <!-- heading -->
    <h1>Desktop Stopwatch</h1>
    <!-- paragraph for time -->
    <p class="stopwatch">
      <span id="min">00</span>:
      <span id="sec">00</span>
    </p>
    <!-- paragraph for buttons -->
    <p class="buttons">
      <button id="start-button">START</button>
      <button id="stop-button">STOP</button>
      <button id="reset-button">RESET</button>
    </p>
  </div>
  <!-- js file -->
  <script src="stopwatch.js" type="text/javascript"></script>
</body>
</html>
```

JS Code:

```
//initialization of min and sec variables
let min = 0;
let sec = 0;
//count variable to update second
let count = 0;
//watch variable to update and store time value
let watch;
//initialization of variables for updating the min and sec
const updateMin = document.getElementById('min');
const updateSec = document.getElementById('sec');
//initialization of variables for storing button events
const startBtn = document.getElementById('start-button');
const stopBtn = document.getElementById('stop-button');
const resetBtn = document.getElementById('reset-button');

//adding event for start button
startBtn.addEventListener('click', function () {
    //disabling the start button after it starts
    startBtn.disabled = true;
    //calling the function startTime for updating the clock
    watch = setInterval(startTime, 10);
})
//adding event for stop button
stopBtn.addEventListener('click', function () {
    //clearing the interval to stop the clock
    clearInterval(watch);
    //enabling the start button to restart the clock
    startBtn.disabled = false;
})
//adding event for reset button
resetBtn.addEventListener('click', function () {
    //clearing the interval to stop the clock
    clearInterval(watch);
    //enabling the start button to restart the clock
    startBtn.disabled = false;
    //resetting all the values and printing them as zero
    min = 0;
    sec = 0;
    updateMin.innerHTML = "00";
    updateSec.innerHTML = "00";
})
//startTime function definition to update stopwatch
function startTime() {
    //increment in the count variable
    count++;
    //checking condition for count variable to update seconds
    if (count > 99) {
        //increment in seconds
        sec++;
        //restarting count
        count = 0;
        //for adding zero before a single digit value
```

```

        updateSec.innerHTML = "0" + sec;
    }
    //if value is in two digits
    if (sec > 9) {
        updateSec.innerHTML = sec;
    }
    //if second completes its cycle of 60 seconds
    if (sec > 59) {
        //updating minute
        min++;
        sec = 0; //restarting seconds and count
        count = 0;
        updateMin.innerHTML = "0" + min;
    }
    //if value is more than single digit
    if (min > 9) {
        updateMin.innerHTML = min;
    }
}

```

CSS Code:

```

/* font import */
@import
url('https://fonts.googleapis.com/css2?family=Kanit:wght@100;200;300;400&display=swap');

/* universal selector */
* {
    font-family: 'Kanit', sans-serif;
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

body {
    background-image: url(https://images.unsplash.com/photo-1533090161767-e6ffed986c88?crop=entropy&cs=tinysrgb&fm=jpg&ixlib=rb-1.2.1&q=80&raw_url=true&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=cr
op&w=1469);
    background-size: cover;
}

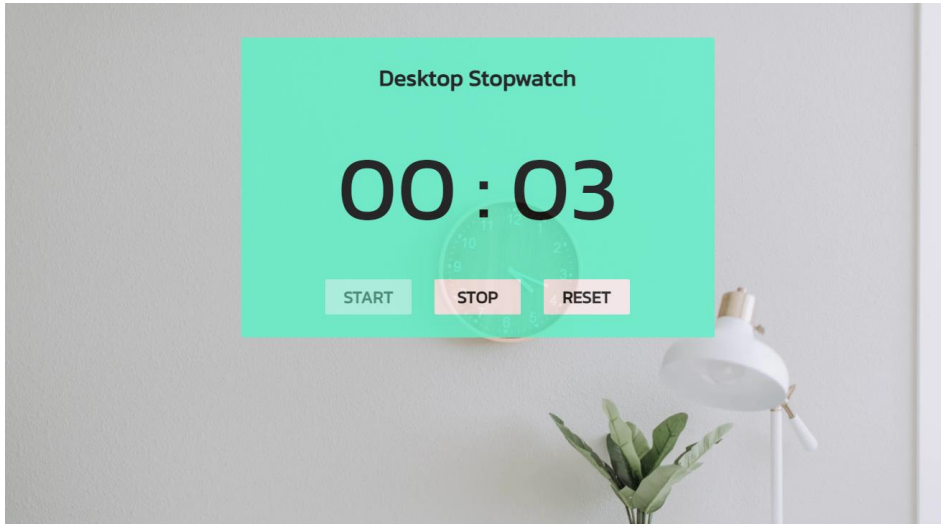
/* main container div styling */
.container {
    background-color: rgb(92, 242, 200);
    display: flex;
    flex-direction: column;
    padding: 2em;
    align-items: center;
    opacity: 0.8;
    width: 50%;
    margin: auto;
    margin-top: 3em;
    gap: 2em;
    border-radius: 0.2em;
}

```

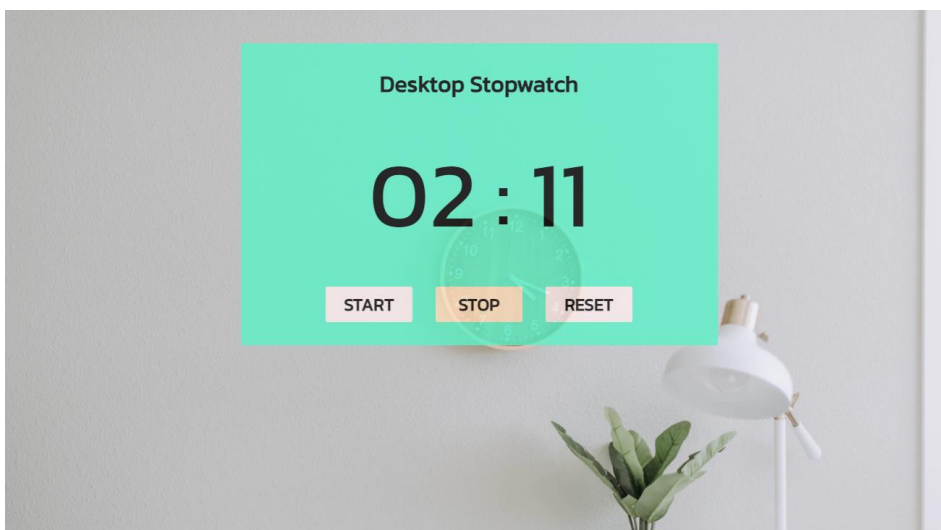
```
}
/* styling for stopwatch paragraph */
.stopwatch {
  font-size: 8em;
  display: flex;
  gap: 0.2em;
}
/* styling for buttons paragraph */
.buttons {
  display: flex;
  gap: 2em;
}

/* styling for buttons */
#start-button,
#stop-button,
#reset-button {
  font-size: 1.5em;
  width: 5em;
  cursor: pointer;
  border-style: none;
  background-color: rgb(252, 238, 238);
  font-weight: bold;
  border-radius: 0.1em;
  color: rgb(10, 10, 10);
  padding: 0.3em;
}
/* hover effect for buttons */
#start-button:hover,
#stop-button:hover,
#reset-button:hover {
  background-color: bisque;
}
/* disabled effect for start button when it is disable after start */
#start-button:disabled {
  opacity: 0.5;
  background-color: azure;
}
```

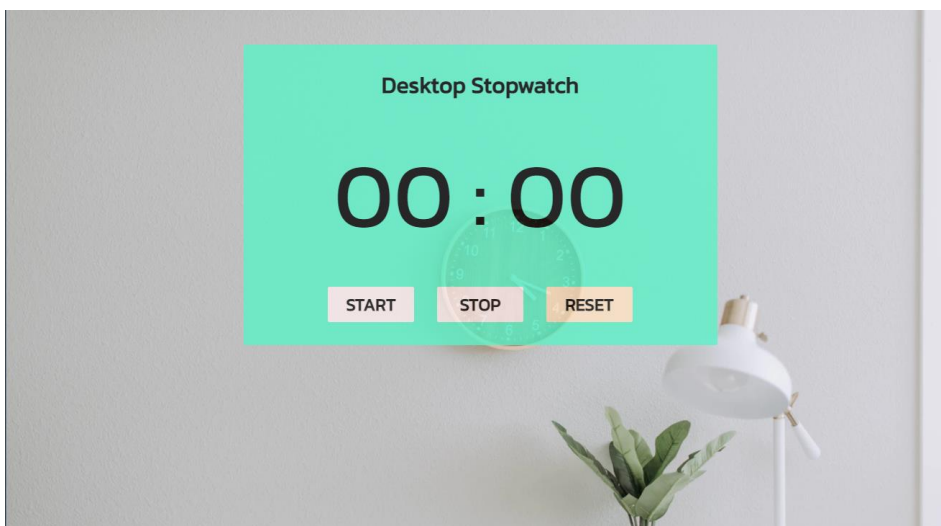
Output:



(Stopwatch started and the start button is disabled to avoid re-click)



(Stopwatch Stopped)



(Stopwatch Reset)