///////////////////import packages

import React from 'react';

import { useState, useEffect } from 'react';

/////////////////////Create Timer.js and declare variables

const Timer = () => {

// eslint-disable-next-line

const [days, setDays] = useState(0);

// eslint-disable-next-line

const [hours, setHours] = useState(0);

// eslint-disable-next-line

const [minutes, setMinutes] = useState(0);

// eslint-disable-next-line

const [seconds, setSeconds] = useState(0);

//////////////declaration of variables

const deadline = "December, 31, 2023";

const getTime = () => {

const time = Date.parse(deadline) - Date.now();

setDays(Math.floor(time / (1000 \* 60 \* 60 \* 24)));

setHours(Math.floor((time / (1000 \* 60 \* 60)) % 24));

setMinutes(Math.floor((time / 1000 / 60) % 60));

setSeconds(Math.floor((time / 1000) % 60));

};

////This useEffect hook is used for state updation.

useEffect(() => {

const interval = setInterval(() => getTime(deadline), 1000);

return () => clearInterval(interval);

}, []);

/////////////////html rendering

return (

<div className="timer" role="timer">

<div className="col-4">

<div className="box">

<p id="day">{days < 10 ? "0" + days : days}</p>

<span className="text">Days</span>

</div>

</div>

<div className="col-4">

<div className="box">

<p id="hour">{hours < 10 ? "0" + hours : hours}</p>

<span className="text">Hours</span>

</div>

</div>

<div className="col-4">

<div className="box">

<p id="minute">{minutes < 10 ? "0" + minutes : minutes}</p>

<span className="text">Minutes</span>

</div>

</div>

<div className="col-4">

<div className="box">

<p id="second">{seconds < 10 ? "0" + seconds : seconds}</p>

<span className="text">Seconds</span>

</div>

</div>

</div>

);

};

export default Timer;

/////////////////////////index.js

import React from 'react';

import ReactDOM from 'react-dom/client';

import './styles/newyearstyles.css';

import App from './App';

import reportWebVitals from './reportWebVitals'; //  **reportWebVitals** is a javascript function that is introduced for measuring the performance of our app

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

// Strict mode is a **set of development tools that help you catch potential problems in your code before** they become actual bugs.

<React.StrictMode>

<div classname='app' id='app'>

<h1>New Year Count Down..</h1>

<App />

</div>

</React.StrictMode>

///////////////////////////Styles.css

\* {

box-sizing: border-box;

margin: 0;

padding: 0;

}

html,

body {

height: 100%;

background-color: rgb(238, 163, 102);

}

#app {

background-color: rgb(238, 163, 102);

color: rgb(255, 255, 255);

font-family: 'Open Sans', sans-serif;

height: 100vh;

text-align: center;

margin-top: 100px;

}

.header {

font-size: calc(20px + 2vmin);

line-height: 2;

text-transform: capitalize;

}

.container {

align-items: center;

display: flex;

flex-direction: column;

height: 100%;

justify-content: center;

margin-top: 30px;

}

.timer {

background-color: rgba(173, 66, 47, 0.856);

display: inline-block;

margin-top: 30px;

padding: 0px;

text-align: center;

width: 400px;

}

.col-4 {

width: 25%;

float: left;

}

.box {

border-right: solid 1px rgba(255, 255, 255, 0.795);

font-weight: 300;

padding: 10px;

}

.col-4:last-child .box {

border-right-color: transparent;

}

.box p {

font-size: calc(16px + 2vmin);

margin: 0;

}

.text {

font-size: 14px

}

#img{

margin-left: 10px;

}