ESTRUCTURA INICIAL PROYECTO

RAÍZ

DIST

-index.html

-styles.css

SRC

-timer.js

CÓDIGO

HTML

<!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">

<link rel="stylesheet" href="styles.css">

<title>TimerJS</title>

</head>

<body>

<div class="container">

<div class="clock">

<input class="number" type="text" maxlength="2" value="00"><span>hs</span><input class="number" type="text" maxlength="2" value="00"><span>min</span><input class="number" type="text" maxlength="2" value="00"><span>sec</span>

</div>

<div class="buttons">

<button id="play" onclick="startTimer()">Start</button>

<button id="stop" onclick="stopTimer()">Stop</button>

</div>

</div>

<script src="../src/timer.js"></script>

</body>

</html>

CSS

\* {

box-sizing: border-box;

padding: 0;

margin: 0;

}

html {

height: 100%;

}

body {

display: flex;

justify-content: center;

align-items: center;

height: 100%;

font-family: 'Roboto', sans-serif;

background: grey;

color: #ecf0f1;

}

.container {

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

background: black;

padding: 20px;

border-radius: 10px;

box-shadow: 0 0 20px rgba(0, 0, 0, 0.5);

}

.clock {

font-size: 32px;

text-shadow: -1px 0px 15px rgba(0,0,0,0.25);

color: #ecf0f1;

margin-bottom: 20px;

}

.number {

display: inline-block;

width: 140px;

font-size: 128px;

font-family: 'Roboto', sans-serif;

text-align: center;

text-shadow: -1px 0px 15px rgba(37, 27, 27, 0.25);

border: none;

color: #ecf0f1;

background-color: transparent;

}

.buttons button {

margin: 0 7px;

padding: 20px 40px;

border: none;

border-radius: 5px;

font-size: 18px;

color: #ecf0f1;

background-color: #3498db;

cursor: pointer;

transition: background-color 0.3s ease;

}

#play {

background-color: #2ecc71;

}

#stop {

background-color: #e74c3c;

}

#play:hover {

background-color: #27ae60;

}

#stop:hover {

background-color: #c0392b;

}

#play:active {

background-color: #1e8449;

}

#stop:active {

background-color: #96281b;

}

JAVASCRIPT

let inputs, clock, hours, minutes, seconds, repeater;

window.addEventListener('load', () => {

inputs = Array.from(document.getElementsByClassName('number')); // Obtiene los inputs

clock = document.querySelector('.clock'); // Obtiene el contenedor del reloj

});

function startTimer() {

parseTime(); // Obtiene los valores de los inputs

setTimer(); // Actualiza el timer visualmente

countdown(); // Inicia el contador

}

function setTimer() {

// Actualiza los valores de los inputs

inputs[0].value = hours > 9 ? hours : '0' + hours;

inputs[1].value = minutes > 9 ? minutes : '0' + minutes;

inputs[2].value = seconds > 9 ? seconds : '0' + seconds;

// Actualiza el título de la pestaña

document.title = `${hours > 9 ? hours : '0' + hours}:${minutes > 9 ? minutes : '0' + minutes}:${seconds > 9 ? seconds : '0' + seconds}`;

}

function parseTime() {

// Convierte los valores de los inputs a números

hours = Number(inputs[0].value);

minutes = Number(inputs[1].value);

seconds = Number(inputs[2].value);

}

function countdown() {

repeater = setInterval(runner, 1000); // Inicia el contador

}

function runner() {

if (seconds > 0) {

seconds--;

} else {

if (minutes > 0) {

seconds = 59;

minutes--;

} else {

if (hours > 0) {

seconds = 59;

minutes = 59;

hours--;

} else {

clearInterval(repeater); // Detiene el contador

alert("Time's up!"); // Muestra una alerta cuando el tiempo termina

}

}

}

setTimer(); // Actualiza el timer visualmente

}

function stopTimer() {

clearInterval(repeater); // Detiene el contador

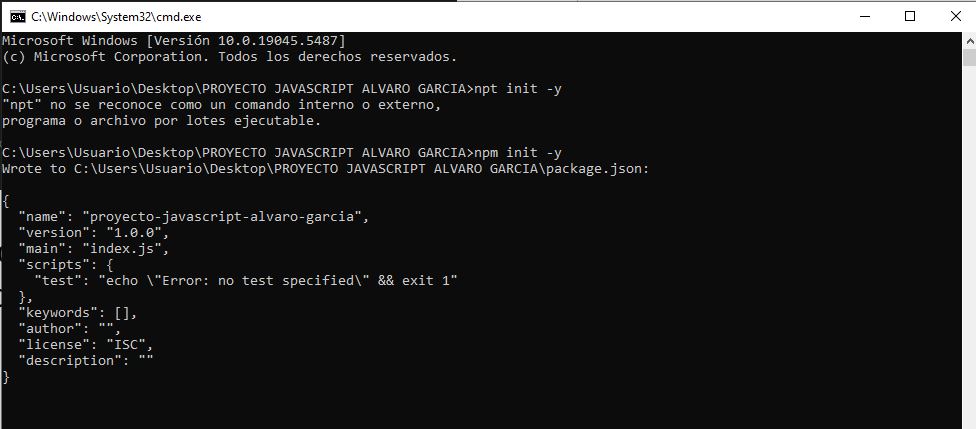
location.reload(); // Recarga la página

}

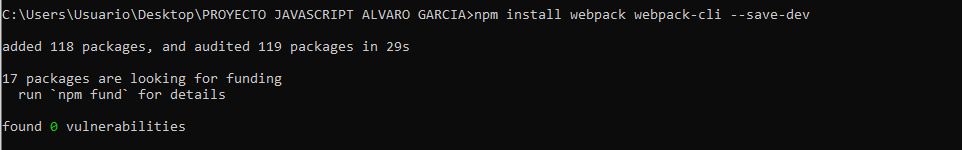
Con estos archivos creados

Abrimos el cmd (node instalado obligatoriamente)

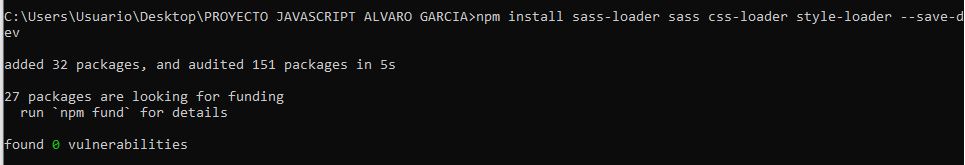
ejecutamos npm init -y para crear archivo json .-(/nodemodules/)-.

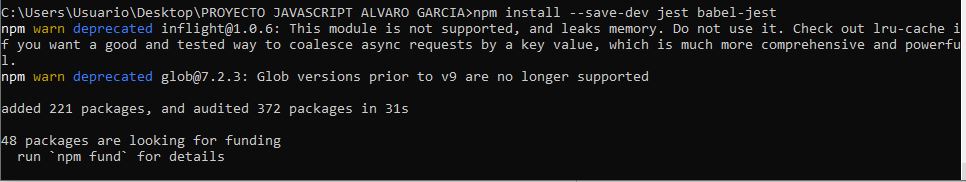


Ejecutamos npm install webpack webpack-cli --save-dev (instalar webpack bundler)



Ejecutamos npm install sass-loader sass css-loader style-loader --save-dev(para que funcione sass con webpack)



Ejecutamos npm install --save-dev jest babel-jest (jest con webpack)

Para que funcione bien el bundler de webpack hay que configurar los archivos package.json {

"name": "proyecto-javascript-alvaro-garcia",

"version": "1.0.0",

"main": "timer.js",

"scripts": {

"build": "webpack",

"test": "jest"

},

"keywords": [],

"author": "",

"license": "ISC",

"description": "",

"devDependencies": {

"babel-jest": "^29.7.0",

"css-loader": "^7.1.2",

"jest": "^29.7.0",

"sass": "^1.85.1",

"sass-loader": "^16.0.5",

"style-loader": "^4.0.0",

"webpack": "^5.98.0",

"webpack-cli": "^6.0.1"

},

"jest": {

"transform": {

"\\.js$": "babel-jest"

},

"moduleNameMapper": {

"^src/(.\*)$": "<rootDir>/src/$1"

}

}

}

creamos los scripts jest y webpack añadimos las devdependencies pertinentes y el apartado de jest (sacado de las páginas linkeadas a la prática)

Archivo webpack.config.js  
const path = require('path');

module.exports = {

mode: 'development',

entry: './src/timer.js',

output: {

filename: 'bundle.js',

path: path.resolve(\_\_dirname, 'dist'),

},

module: {

rules: [

{

test: /\.scss$/, // Procesar archivos SCSS

use: ['style-loader', 'css-loader', 'sass-loader'], // Cargar y compilar SCSS

},

{

test: /\.js$/, // Procesar archivos JavaScript

exclude: /node\_modules/,

use: 'babel-loader',

},

],

},

resolve: {

extensions: ['.js', '.jsx', '.json'], // Resolución de extensiones

},

};

Añadimos el entry en nuestro js principal

la parte de scss y jest para que funcionen respectivamente extensions para que identifique las extensiones presentes.

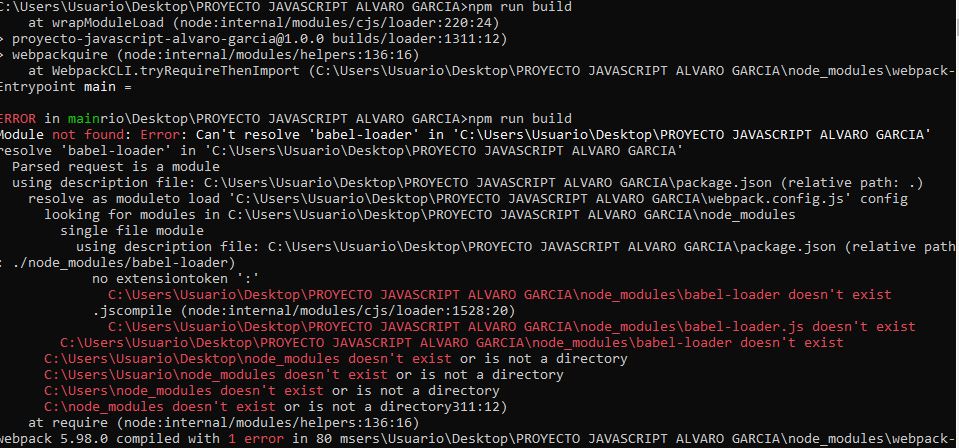
Creacion archivo config.babelrc con ese código  
{

"presets": ["@babel/preset-env"]

}

Sinceramente he tenido un montón de errores al hacerlo y he tenido que ayudarme un montón para que funcione y no salgan casi errores.

Por eso la explicación no es muy buena ya que hay muchas cosas que no me ha dado tiempo a entender  
  
Al ejecutar npm run build se ejecuta con un error en el jest ( no se solucionarlo aún)



pero se genera el archivo bundle.js