Neumorphism Analog Clock

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*="shortcut icon" *href*="favicon.png" *type*="image/x-icon">

*<!-- Adding css link -->*

    <link *rel*="stylesheet" *href*="style.css">

    <title>CLOCK</title>

</head>

* First add the simple doctype of HTML and then add the favicon and the CSS file link to it.

<body>

    <div *class*="clock">

        <div *class*="hour">

            <div *class*="hr"></div>

        </div>

        <div *class*="min">

            <div *class*="mn"></div>

        </div>

        <div *class*="sec">

            <div *class*="sc"></div>

        </div>

    </div>

Then the start of body tag inside it

* One div that contains the total clock inside it creates 3 more div for an hour min and sec. to show the sticks.

Now we have create the footer for credits

 <footer>

        <p>&copy; Gyana Ranjan Barik</p>

        <p>All Rights Reserved</p>

    </footer>

CSS

\* {

    margin: 0;

    padding: 0;

    box-sizing: border-box;

}

body {

    display: flex;

    justify-content: center;

    align-items: center;

    min-height: 100*vh*;

    background: #091921;

}

The above two CSS property is given for the whole body and the background color of the body.

By giving display flex we can add the aligned item and justify-content to center.

*.clock* {

    width: 350*px*;

    height: 350*px*;

    display: flex;

    justify-content: center;

    align-items: center;

    background: url(Clock\_photo.png);

    background-size: cover;

    border: 4*px* solid #091921;

    box-shadow: 0 -15*px* 15*px* rgba(0, 253, 143, 0.05), inset 0 -15*px* 15*px* rgba(0, 247, 255, 0.116), 0 -15*px* 15*px* rgba(16, 255, 235, 0.13), 0 -15*px* 15*px* rgba(179, 16, 16, 0.05);

    border-radius: 50*%*;

}

Now we give the CSS property to the class of clock

* First, we set its height and width here also we give its display property to make it in the center .and give one background image which will show our clock.
* Then for adding neomorphism to our clock we give it some border and the box-shadow and give some color to it.
* And here the background-size cover is given as “cover” for making it responsive it makes the photo responsive according to the size of the screen.

*.clock:before* {

    content: '';

    position: absolute;

    width: 15*px*;

    height: 15*px*;

    background: #fff;

    border-radius: 50*%*;

    z-index: 10000;

}

As the CSS property says clock before means this CSS property will be applied before the clock.

* The content used in the: before property to insert generate content here we want to add a simple round button to the middle of the clock png so we add this and give CSS property to it.
* We give it position absolute for make its size according to screen and give height and width and border radius for its round shape and give high z-index to show it first than others.

*.clock* *.hour*,

*.clock* *.min*,

*.clock.sec* {

    position: absolute;

}

Give position to all its sticks so it will adjust according to screen width.

*.clock* *.hour* {

    width: 160*px*;

    height: 160*px*;

}

*.clock* *.min* {

    width: 190*px*;

    height: 190*px*;

}

*.clock* *.sec* {

    width: 230*px*;

    height: 230*px*;

}

Giving height and weight to all the sticks.

*.hour*,

*.min*,

*.sec* {

    display: flex;

    justify-content: center;

    position: absolute;

    border-radius: 50*%*;

}

*.hour::before* {

    content: '';

    position: absolute;

    width: 8*px*;

    height: 80*px*;

    background: #ff105e;

    z-index: 10;

    border-radius: 6*px* 6*px* 0 0;

}

*.min::before* {

    content: '';

    position: absolute;

    width: 4*px*;

    height: 90*px*;

    background: #cbff10;

    z-index: 11;

    border-radius: 6*px* 6*px* 0 0;

}

*.sec::before* {

    content: '';

    position: absolute;

    width: 2*px*;

    height: 150*px*;

    background: #ffbf10;

    z-index: 12;

    border-radius: 6*px* 6*px* 0 0;

}

Now here we give the footer credit

footer {

    position: absolute;

    margin-top: 620*px*;

    color: rgb(155, 110, 110);

    text-align: center;

    font-weight: 400;

    line-height: 2;

}

Now here we give brain to our clock for working properly

<script ***"***>

*const sc = document.querySelector(*".sec "*)*;

*const mn = document.querySelector(*".min "*)*;

*const hr = document.querySelector(*".hour "*)*;

         setInterval(*function*(){

*let* time *= new Date()*;

*let* secs *= time.getSeconds() \* 6*;

*let* mins *= time.getMinutes() \* 6*;

*let* hrs *= time.getHours() \* 30*;

*sc*.*style*.*transform* *=* `rotateZ(${secs}deg)`;

*mn*.*style*.*transform* *=* `rotateZ(${mins}deg)`;

*hr*.*style*.*transform* *=* `rotateZ(${hrs*+*(mins*/*12)}deg)`;

         });

    </script>

First we three const variables and select the selectors of our CSS.

We will calculate the time with the help of our set interval function inside the function we create three variables that one for our time which will show time with help of the new Date() function.

And the other three functions are to get an hour, minutes, and seconds.

How we calculate

* FOR HOUR CHANGES IN THE CLOCK

for 12 hr it rotates 360deg

for 1 hr it will be 360/12 = 30deg

so we know how it will take for 60min it will rotate 30 deg to complete

so 1min ->(min/2) rotation (1/2)deg.

We don’t consider here the second because it changes to close or say no so we have not considered it

* FOR min CHANGES IN THE CLOCK.

for 60min it will rotate 360 deg

1 min = 6 deg

m minitues -> 6mdeg

* FOR SECOND CHANGES IN THE CLOCK

60 sec = 360deg

1 sec= 6deg.

Adding the time in Digital Format

 <div>

            <div *class*="clock\_text">

                <div *class*="clock\_text-hour" *id*="text-hour"></div>

                <div *class*="clock\_text-min" *id*="text-min"></div>

                <div *class*="clock\_text-ampm" *id*="text-ampm"></div>

            </div>

            <div *class*="clock\_date">

                <span *id*="date-day"></span>

                <span *id*="date-month"></span>

                <span *id*="date-year"></span>

            </div>

 </div>

* First, we create a div inside it we create three divs one div for showing the time and another div will show the year and month.
* Inside the time div with class name clock text inside it we create three div one for an hour to show one for min and one to show am and pm.
* Inside the date gives a class name that is clock\_date and creates a three-span to show day month and year.

Adding CSS To The digital clock format

*:root* {

*--biggest-font-size*: 3*rem*;

*--font-medium*: 500;

*--small-font-size*: 3*rem*;

*--title-color*: hsl(var(*--hue-color*), 53*%*, 15*%*);

*--tiny-font-size*: .625*rem*;

*--mb-0-25*: .45*rem*;

}

*.clock\_text* {

    display: flex;

    justify-content: center;

    margin-top: 560*px*;

    color: #10ffdf;

}

*.clock\_text-hour*,

*.clock\_text-min* {

    font-size: var(*--biggest-font-size*);

    font-weight: var(*--font-medium*);

    color: #bdb222;

}

*.clock\_text-ampm* {

    font-size: var(*--tiny-font-size*);

    color: var(*--title-color*);

    font-weight: var(*--font-medium*);

    margin-left: var(*--mb-0-25*);

}

*.clock\_date* {

    text-align: center;

    font-size: var(*--small-font-size*);

    font-weight: var(*--font-medium*);

    color: #bdb222;

}

: root is useful for declaring a variable for HTML element inside it declare some variables

Then we select the time class i.e clock\_text we give it some property to take it to the center of our page.

Similarly, we give to another class some CSS

JAVASCRIPT FOR DIGITAL CLOCK

*//  For digi clock*

*const texthour = document.getElementById(*'text-hour'*),*

*textmin = document.getElementById(*'text-min'*),*

*textAmPm = document.getElementById(*'text-ampm'*),*

*dateDay=document.getElementById(*'date-day'*),*

*datemonth=document.getElementById(*'date-month'*),*

*dateyear=document.getElementById(*'date-year'*)*

*const   clocktext = () =>*

*{*

*let* date *= new Date();*

*let* hh *= date.getHours(),*ampm

*let* mm *= date.getMinutes(),*

day *= date.getDate(),*

month *= date.getMonth(),*

year *= date.getFullYear()*

*// we change the hours from 24 to 12 and establish whether am or pm*

*if(*hh*>=12)*

*{*

hh*=*hh*-12;*

ampm*=*'pm'*;*

*}*

*else*

*{*

ampm*=*'Am'

*}*

*//   We detect when its 0 Am and transform to 12 AM*

*if(*hh*==0) {*hh*==12}*

*// Show a zero before hours*

*if(*hh*<10){*hh*=`0${*hh*}`}*

*// Show time*

*texthour.innerHTML = `${*hh*}:`*

*// Show a zero before the min*

*if(*mm*<10){*mm*=`0${*mm*}`}*

*// Show minutes*

*textmin.innerHTML = `${*mm*}`*

*// Show am or pm*

*textAmPm.innerHTML =*ampm*;*

*// We get the months and years and show it*

*let* months *= [*'jan'*,*'Feb'*,*'Mar'*,*'Apr'*,*'May'*,*'Jun'*,*'Jul'*,*'Aug'*,*'Sep'*,*'Oct'*,*'Nov'*,*'Dec'*]*

*// Show the day the month and the year*

*dateDay.innerHTML =*day*;*

*datemonth.innerHTML =`${*months*[*month*]},`;*

*dateyear.innerHTML =* year*;*

*}*

     setInterval(clocktext,1000) *//1000 = 1s*

‘’ AUTHOR – Gyana Ranajan Barik

PROJECT- Neumorphism Analog Calculator

TOOL -HTML CSS JAVASCRIPT ‘’