

**Coreline Engineering Solutions Website**

Documentation

Contents

[Description of Project 2](#_Toc147760504)

[Flow Diagram 2](#_Toc147760505)

[Components 2](#_Toc147760506)

[Functions 2](#_Toc147760507)

[Source Code 2](#_Toc147760508)

[HTML 2](#_Toc147760509)

[CSS 4](#_Toc147760510)

[JavaScript 14](#_Toc147760511)

# Description of Project

This project focus is a website for Coreline Engineering Solutions, the idea behind it is minimum like a business card, but still fun.

Company logo is in the shape of a cube therefore the idea of a rolling dice came to be. Now the user will have the opportunity to roll the dice through all the pages.

We have a total of 3 pages with content to be seen. With a quick access to a contact us list when user icon is pressed in the top right corner.

# Flow Diagram

The cube has 4 results that are displayed in a row first page being Home, secondly About, and then Service, the 4 page is a reset to landing page.

Cube:

* Home
* About
* Services
* Landing Page
* Contact Button has one can be pressed and will drop down a list where the user will be able to contact CES via Phone call, Email and LinkedIn.

Contact Button:

* Phone
* Email
* LinkedIn

# Components

## Source Code and Discription

### HTML

<!DOCTYPE html>

<html>

<head>

    <meta charset="UTF-8">

    <title>Coreline Engineering Solutions</title>

<!-- Font -->

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=Roboto&display=swap" rel="stylesheet">

<link rel="icon" type="image/png" href="./Pictures/Logo\_big.png"/>

<!-- CSS -->

  <link rel="stylesheet" type="text/css" href="./CSS/style.css">

</head>

<body class="container">

Where we place the Background Video on the website

  <!-- Video Background -->

  <video autoplay loop muted play-inline class="back-video">

    <source src="./Pictures/Map Background video.mp4" type="video/mp4">

  </video>

Here the Cube is being placed on the website with all its faces.

  <!-- Cube -->

  <div class="cube start">

    <div class="face front">

      <img id="frontImage" src="./Pictures/Cube Faces/S.png" alt="">

    </div>

    <div class="face top">

      <img id="topImage" src="./Pictures/Cube Faces/E.png" alt="">

    </div>

    <div class="face left">

      <img id="leftImage" src="./Pictures/Cube Faces/C.png" alt="">

    </div>

    <div class="face right">

      <img id="rightImage" src="./Pictures/Cube Faces/C.png" alt="">

    </div>

    <div class="face back">

      <img id="backImage" src="./Pictures/Cube Faces/S.png" alt="">

    </div>

    <div class="face bottom">

      <img id="bottomImage" src="./Pictures/Cube Faces/E.png" alt="">

    </div>

  </div>

Here we add the company logo when the website loads in and when all the content have been viewed.

  <!-- Company Name -->

  <div class ="logoName">

    <img src="./Pictures/Logo\_name.png" width="50%" height="auto" alt="">

  </div>

Here we add the Title and Text of every page content.

  <!-- Site info divs -->

  <div id="title"></div>

  <div id="context"></div>

Here we add the contact button where people can contact us with hyperlinks.

  <!-- Contact Details Button -->

  <div id = "contactdiv">

  <img id="contact-button" src="./Pictures/contactButtonImg.png" alt="Contact Button">

  <!-- Contact Details -->

  <div id="contactdetails">

    <h2><u>Contact us</u></h2>

    <p>

      <a href="tel:+27218791136">+2721 879 1136</a><br>

      <a href="mailto:info@corelineengineering.com">info@corelineengineering.com</a><br>

      <a href="https://www.linkedin.com/company/coreline-engineering-solutions/">Linkedin</a>

    </p>

  </div>

</div>

Here we add our JavaScript and its libraries for the website.

<script src="https://cdnjs.cloudflare.com/ajax/libs/hammer.js/2.0.8/hammer.min.js"></script>

<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

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

</body>

</html>

### CSS

Style of website so that it fills the whole screen.

body {

    display: flex;

    justify-content: center;

    align-items: center;

    height: 100vh;

    width: 100%;

    margin: 0px;

    overflow: hidden;

}

Making hyperlinks black and underline when hovered over.

a:link {

    text-decoration: none;

    color: black;

  }

  a:hover {

    text-decoration: underline;

    color: black;

  }

Putting logo name behind content.

  .logoName {

    z-index: -2;

  }

Putting Video behind content and making it fill screen.

.back-video{

    width: 100%;

    height: 100%;

    object-fit: cover;

    position: fixed;

    right:0;

    bottom: 0;

    z-index: -2;

}

Specifying Contact Button Details and look.

#contact-button {

    position: fixed;

    top: 20px;

    right: 20px;

    text-align: right;

    width: auto;

    height: 50px;

    border-color: rgba(0, 77, 144, 0);

    background-color: #a4a4a408;

    z-index: 1;

}

#contactdetails {

    display: none;

    position: fixed;

    margin-right: 50px;

    padding: 5px;

    top: 10px;

    right: 7%;

    width: 400px;

    background: radial-gradient(at 100% 50%, rgb(255, 255, 255) 200px, #ffffff00 75%);

    z-index: 2;

    text-align: right;

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

}

#contact-button img{

    width: 10%;

}

Specifying basic details on page content texts

/\* Text \*/

#title {

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

    display: none; /\* Hides the title and content text on start up   \*/

}

#context {

    background-image: url('/Pictures/Text Background.png');

    background-size: cover;

    background-position: center;

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

    display: none; /\* Hides the title and content text on start up   \*/

}

Creating a cube from the 6 faces and specifying their sizes

/\* Cube \*/

.container {

    position: relative;

    text-align: center;

}

.cube {

    position: fixed;

    transform-style: preserve-3d;

    transform: rotateX(0deg) rotateY(0deg);

}

/\* For still standing start \*/

.cube.start {

    transform: rotateX(-40deg) rotateY(45deg);

    top: 3.3em;

    left: 45%;

}

.face {

    position: absolute;

    width: 4em;

    height: 4em;

    background: rgb(253, 253, 253);

    text-align: center;

}

.face img {

    width: 100%;

    height: 100%;

}

.front  { transform: rotateY(0deg) translateZ(2em); }

.top    { transform: rotateX(90deg) translateZ(2em); }

.right  { transform: rotateY(90deg) translateZ(2em); }

.left   { transform: rotateY(-90deg) translateZ(2em); }

.bottom { transform: rotateX(-90deg) translateZ(2em); }

.back   { transform: rotateY(180deg) translateZ(2em); }

Cube movement functions to make the cube spin/move from point A to B.

/\* First Spins the cube \*/

@keyframes spin {

    0% {

        transform: rotateX(-40deg) rotateY(45deg) rotateZ(0deg);

    }

    100% {

        transform: rotateX(360deg) rotateY(360deg) rotateZ(360deg) ;

    }

}

@keyframes startshift {

    0% {transform: translateX(3.3em) ;}

    100% {transform: translateX(4.5em);}

}

/\* Other Spins the cube \*/

@keyframes otherSpin {

    0% {

        transform: rotateX(0deg) rotateY(0deg) rotateZ(0deg);

    }

    100% {

        transform: rotateX(360deg) rotateY(360deg) rotateZ(360deg) ;

    }

}

/\* Last page spin  \*/

@keyframes lastPageSpin {

    0% {

        transform: rotateX(360deg) rotateY(360deg) rotateZ(360deg);

    }

    100% {

        transform: rotateX(-40deg) rotateY(45deg) rotateZ(0deg);

    }

}

Changing specific content according to screen size and type

Media 1 and 2 for Desktops

/\* SCENES \*/

/\* Desktop screne \*/

@media (min-width: 1330px) and (min-height: 871px) {

    .cube {

        font-size: 4.5em;

        top: 4.5em;

        left: 35%;

    }

    #title {

        font-size: 3em;

        padding: 5px;

        width: 5.8em;

        height: 5.8em;

        position:absolute;

        top: 0.6em;

        left: 35%;

    }

    #context {

        font-size: 1em;

        padding: 10px 5px;

        width: 35.8em;

        height: 16.8em;

        position:absolute;

        top: 20.2em;

        left: 35%;

    }

    .logoName {

        position: absolute;

        top: 35em;

        left: 0em;

        right: 0em;

    }

}

/\* Desktop With height problem \*/

@media (min-width: 1330px) and (max-height: 870px) {

    .cube {

        font-size: 3.67em;

        top: 4em;

        left: 36%;

    }

    #title {

        font-size: 2.4em;

        padding: 5px;

        width: 5.85em;

        height: 5.85em;

        position:absolute;

        top: -0.15em;

        left: 36%;

    }

    #context {

        font-size: 1em;

        padding: 10px 5px;

        width: 42em;

        height: 13em;

        position:absolute;

        top: 14.9em;

        left: 30%;

    }

    .logoName {

        position: absolute;

        padding-top: 26em;

        left: 0em;

        right: 0em;

    }

}

Media 3 and 4 for Laptops

  /\* Laptop screne \*/

  @media (min-width: 971px) and (max-width: 1329px) and (min-height: 871px) {

    .cube {

        font-size: 3.67em;

        top: 4em;

        left: 26%;

    }

    #title {

        font-size: 2.4em;

        padding: 5px;

        width: 5.85em;

        height: 5.85em;

        position: absolute;

        top: -0.15em;

        left: 26%;

    }

    #context {

        font-size: 1em;

        padding: 10px 5px;

        width: 42em;

        height: 13em;

        position: absolute;

        top: 14.9em;

        left: 17%;

    }

    .logoName {

        position: absolute;

        padding-top: 16em;

        left: 0em;

        right: 0em;

    }

  }

  @media (min-width: 971px) and (max-width: 1329px) and (max-height: 870px) {

    .cube {

        font-size: 3.67em;

        top: 4em;

        left: 26%;

    }

    #title {

        font-size: 2.4em;

        padding: 5px;

        width: 5.85em;

        height: 5.85em;

        position:absolute;

        top: -0.15em;

        left: 26%;

    }

    #context {

        font-size: 1em;

        padding: 10px 5px;

        width: 42em;

        height: 13em;

        position:absolute;

        top: 14.9em;

        left: 17%;

    }

    .logoName {

        position: absolute;

        padding-top: 20em;

        left: 0em;

        right: 0em;

    }

  }

Media 5 and 6 for Tablets

  /\* Tablet screens \*/

  @media (min-width: 641px) and (max-width: 970px) and (min-height: 871px) {

    body {

        overflow: visible;

    }

    .cube {

        font-size: 3.2em;

        top: 4em;

        left: 36%;

    }

    #title {

        font-size: 2em;

        padding: 5px;

        width: 6.1em;

        height: 5.99em;

        position: absolute;

        top: -0.25em;

        left: 36%;

    }

    #context {

        font-size: 1em;

        padding: 10px 5px;

        width: 21em;

        height: 27em;

        position:absolute;

        top: 13em;

        left: 14%;

    }

    .logoName {

        position: absolute;

        padding-top: 6em;

        left: 0em;

        right: 0em;

    }

  }

  @media (min-width: 641px) and (max-width: 970px) and (max-height: 870px) {

    body {

        overflow: visible;

    }

    .cube {

        font-size: 3.2em;

        top: 4em;

        left: 36%;

    }

    #title {

        font-size: 2em;

        padding: 5px;

        width: 6.1em;

        height: 5.99em;

        position:absolute;

        top: -0.25em;

        left: 36%;

    }

    #context {

        font-size: 1em;

        padding: 10px 5px;

        width: 21em;

        height: 27em;

        position:absolute;

        top: 13em;

        left: 14%;

    }

    .logoName {

        position: absolute;

        padding-top: 14em;

        left: 0em;

        right: 0em;

    }

  }

Media 7 and 8 for Phones

  /\* Phone screens \*/

  @media (max-width: 640px) and (min-height: 871px) {

    body {

        overflow: visible;

    }

    .cube {

        font-size: 2.5em;

        top: 4em;

        left: 35%;

    }

    #title {

        font-size: 1.7em;

        padding: 5px;

        width: 5.5em;

        height: 5.5em;

        position:absolute;

        top: -0.25em;

        left: 35%;

    }

    #context {

        font-size: 1em;

        padding: 10px 5px;

        width: 18em;

        height: 32em;

        position:absolute;

        top: 9.75em;

        left: 8%;

    }

    .logoName {

        position: absolute;

        padding-bottom: 12em;

        left: -4em;

        right: 0em;

        width: 40em;

    }

  }

  @media (max-width: 640px) and (max-height: 870px) {

    body {

        overflow: visible;

    }

    .cube {

        font-size: 2.5em;

        top: 4em;

        left: 35%;

    }

    #title {

        font-size: 1.7em;

        padding: 5px;

        width: 5.5em;

        height: 5.5em;

        position:absolute;

        top: -0.25em;

        left: 35%;

    }

    #context {

        font-size: 1em;

        padding: 10px 5px;

        width: 18em;

        height: 32em;

        position:absolute;

        top: 9.75em;

        left: 8%;

    }

    .logoName {

        position: absolute;

        padding-top: 7em;

        left: -4em;

        right: 0em;

        width: 40em;

    }

  }

### JavaScript

Variable declaration

const cube = $('.cube');

var currentIndex = 0;

let scrolling = false;

var contactshow = false;

Page Content:

Every array is a page with its title, logo and context the cube will cycle through this content when cube spins

// Page Content

var pages = [

    {

        title: '<h2> </h2>',

        image: './Pictures/Blank.png',

        context: ' '

    },

    {

        title: '<h2>Home</h2>',

        image: './Pictures/Logo\_big.png',

        context: '<h3>Welcome to Coreline Engineering Solutions</h3><br>At Coreline Engineering Solutions, we are dedicated to revolutionizing the telecommunications industry through our expert design services. As a South African based provider, we bring a unique perspective to the global telecommunications outside plant design market. Our team of highly skilled engineers is focused on delivering innovative and efficient solutions to connect communities and businesses around the world. Join us in our mission to build a connected future.'

    },

    {

        title: '<h2>About</h2>',

        image: './Pictures/Logo\_big.png',

        context: '<br><br>At Coreline Engineering Solutions, we are dedicated to revolutionizing the telecommunications industry through our expert design services. As a South African based provider, we bring a unique perspective to the global telecommunications outside plant design market.<br><br> Our team of highly skilled engineers is focused on delivering innovative and efficient solutions to connect communities and businesses around the world. Join us in our mission to build a connected future.'

    },

    {

        title: '<h2>Services</h2>',

        image: './Pictures/Logo\_big.png',

        context: '<br><br>Our team specializes in access planning, permits, fiber optic engineering, and transmission planning. We excel in fiber optic system design, and route surveys, ensuring comprehensive project details for cost estimation and implementation. We utilize GIS mapping, AutoCAD, Google Earth, Global Mapper, and MapSource for our work.'

    },

];

This is the contact button top right, when pressed the drop down appears (Slides down) with the company contact details

//Contact Button

var contactshow = false;

        $('#contact-button').click(function () {

            if (contactshow === true) {

                $('#contactdetails').slideUp(500);

                contactshow = false;

            } else {

                $('#contactdetails').slideDown(500);

                contactshow = true;

            }

        });

Puts objects in an array so they listen and do the same action. Swiping the cube, title or context results in the cube spinning to the next page.

// Phone Swipe Listener

var cubeListen = $('.cube')[0];

var titleListen = $('#title')[0];

var contextListen = $('#context')[0];

var phoneListeners = [cubeListen, titleListen, contextListen];

phoneListeners.forEach(function (plistener) {

  var hammer = new Hammer(plistener);

  hammer.on('swipe', function (event) {

    if (!scrolling) {

      scrolling = true;

      if (event.direction === Hammer.DIRECTION\_LEFT) {

        // Next page = Swipe left

        currentIndex = (currentIndex + 1) % pages.length;

        if (currentIndex === 0) {

          $('.cube').addClass('end');

          preSpinPrep();

        } else {

          preSpinPrep();

        }

When swipe in the opposite direction results in the page going backward.

      } else if (event.direction === Hammer.DIRECTION\_RIGHT) {

        // Previous page = Swipe right

        currentIndex = (currentIndex - 1 + pages.length) % pages.length;

        if (currentIndex === 0) {

          $('.cube').addClass('end');

          preSpinPrep();

        } else {

          preSpinPrep();

        }

      }

      setTimeout(() => {

        scrolling = false;

      }, 5000);

    }

  });

});

When the user swipes the opposite directions the cube will go a page back instead.

        } else if (event.direction === Hammer.DIRECTION\_RIGHT) {

            // Previous page = Swipe right

            currentIndex = (currentIndex - 1 + pages.length) % pages.length;

            if (currentIndex === 0) {

                $('.cube').addClass('end');

                preSpinPrep();

            } else {

                preSpinPrep();

            }

        }

        setTimeout(() => {

            scrolling = false;

        }, 5000);

    }

});

Here we listen for computer actions with a mouse and apply it to cube.

//Computer Scroll and Click Listener

function setupInputListener(selector) {

    $(selector).on('wheel mousedown', (event) => {

        if ((event.type === 'wheel' || (event.type === 'mousedown' && event.button === 0)) && !scrolling) {

            scrolling = true;

            debouncedScroll(event);

            setTimeout(() => {

                scrolling = false;

            }, 4000);

        }

    });

}

setupInputListener('.cube');

setupInputListener('#title');

setupInputListener('#context');

Debounce function blocks the functions from listening at the user during the given time so that the animation can finish before running again while it already busy running reducing visual bugs

// Debounce Function

function debounce(func, wait) {

    let timeout;

    return function () {

        const context = this;

        const args = arguments;

        clearTimeout(timeout);

        timeout = setTimeout(() => {

            func.apply(context, args);

        }, wait);

    };

}

Here we change to next page according to what the computer listener received and choose the next page.

// Page Cycle Function

const debouncedScroll = debounce((event) => {

    // User scrolled Up

    if (event.originalEvent.deltaY < 0) {

        currentIndex = (currentIndex - 1 + pages.length) % pages.length;

    } else if (event.originalEvent.deltaY > 0) {

        currentIndex = (currentIndex + 1) % pages.length;

    } else if (event.type === 'mousedown' && event.button === 0) {

        // User scrolled Up

        currentIndex = (currentIndex + 1) % pages.length;

    }

If the class reaches index of 0 the class cube class must gain the class of end so that the cube will return to its first closed state.

    if (currentIndex === 0) {

        $('.cube').addClass('end');

    }

    preSpinPrep();

}, 200);

The Pre Spin Function Hides the text before a spin starts and then launches the spin function if it not the first spin the function will run the reverse function to change the content back to a cube before it spins again

//Pre spin Prepare

function preSpinPrep() {

    //Hides Text for content change

    $('#title').fadeOut(400, function() {

        $('#title').hide();

    });

    $('#context').fadeOut(400, function() {

        $('#context').hide();

    });

    setTimeout(() => {

        if (window.matchMedia("(max-width: 970px)").matches) { // phone

            if ($('.cube').hasClass('start')) {

                spin();

            } else {// Change flat Cube back to cube

                reversePcube(function() {

                    // spin when cube is cube again

                    spin();

                });

            }

        } else { // computer

            if ($('.cube').hasClass('start')) {

                spin();

            } else {// Change flat Cube back to cube

                reverseCcube(function() {

                    // spin when cube is cube again

                    spin();

                });

            }

        }

    }, 500);

}

The spin function starts off by forcing the text to be hidden to prevent visual bugs.

//Spins the cube acordingly

function spin() {

    //Force Hides text

    $('#title').hide();

    $('#context').hide();

We have 3 different spins to ensure for more smooth transitions regarding where the cube last has been:

First spin happens if we have the start class on this will also hide the company logo and starts the spinning and then opens the cube. It will then fade in the text on top.

    if ($('.cube').hasClass('start')) {

        //Hide Logo

        setTimeout(() => {

            $('.logoName').fadeOut(400);

        }, 100);

        cube.css('animation', 'spin 1s linear infinite');

        $(".cube").removeClass("start");

        setTimeout(() => {

            cube.css('animation', 'none');

            //Timeout = space for cube to fold open

            setTimeout(() => {

                if (window.matchMedia("(max-width: 970px)").matches) {

                    // phone

                    mobileDisplay();

                } else {

                    // computer

                    computerDisplay();

                }

            }, 200);

            setTimeout(() => {

                $('#title').html(pages[currentIndex].title);

                $('#title').fadeIn(500);

                $('#context').html(pages[currentIndex].context);

                $('#context').fadeIn(500);

            }, 1500);

        }, 1000);

The second spin is the last spin that must have the class end to be called. This will spin and then shows the company logo.

    // Check if this is the last spin

    } else if ($('.cube').hasClass('end')) {

        cube.css('animation', 'lastPageSpin 1s linear infinite');

        setTimeout(() => {

            cube.css('animation', 'none');

        }, 1000);

        setTimeout(() => {

            $('.logoName').fadeIn(800);

        }, 100);

        $(".cube").addClass("start");

        $(".cube").removeClass("end");

    } else {

The last spin is the general spin from page to page that shows the text in a fade in method.

        cube.css('animation', 'otherSpin 1s linear infinite');

        setTimeout(() => {

            cube.css('animation', 'none');

            //Timeout = space for cube to fold open

            setTimeout(() => {

                if (window.matchMedia("(max-width: 970px)").matches) {

                    // phone

                    mobileDisplay();

                } else {

                    // computer

                    computerDisplay();

                }

            }, 200);

            setTimeout(() => {

                $('#title').html(pages[currentIndex].title);

                $('#title').fadeIn(500);

                $('#context').html(pages[currentIndex].context);

                $('#context').fadeIn(500);

            }, 1500);

        }, 1000);

    }

}

Mobile Display Function is the function that is called to open the cube for mobile form. Starts by changing the pictures to blank then showing displaying the cube one by one as the move out and then the location where the faces are going to end at.

// Cube Rotation Mobile

function mobileDisplay() {

    //Stops spinning hide all

    $('.top, .right, .left, .bottom, .back').hide();

    //Change images to Blank and logo

    $('#frontImage').attr('src', './Pictures/Cube Faces/Blank.png');

    $('#topImage').attr('src', './Pictures/Cube Faces/Blank.png');

    $('#leftImage').attr('src', './Pictures/Cube Faces/Blank.png');

    $('#rightImage').attr('src', './Pictures/Cube Faces/Logo.png');

    $('#backImage').attr('src', './Pictures/Cube Faces/Blank.png');

    $('#bottomImage').attr('src', './Pictures/Cube Faces/Blank.png');

    setTimeout(() => {

        $('.top').fadeIn(200, function() {

            $('.top').show();

            $('.left').fadeIn(200, function() {

                $('.left').show();

                $('.bottom').fadeIn(200, function() {

                    $('.bottom').show();

                    $('.right').fadeIn(200, function() {

                        $('.right').show();

                        $('.back').fadeIn(200, function() {

                            $('.back').show();

                        });

                    });

                });

            });

        });

    });

    // Reset shape to cube

    $(".front").css("transform", "rotateY(0deg) translateZ(2em) translateY(-0.1em)");

    $(".top").css("transform", "rotateX(180deg) rotateZ(0deg) translateY(5.2em) translateZ(-2em) translateY(-1em)");

    $(".right").css("transform", "rotateY(180deg) rotateX(180deg) rotateZ(180deg) translateZ(2em) translateX(4.1em) translateY(4em)");

    $(".left").css("transform", "rotateY(0deg) translateZ(2em) translateX(-4.1em) translateY(-0.1em)");

    $(".bottom").css("transform", "rotateX(0deg) translateZ(2em) translateY(4em)");

    $(".back").css("transform", "rotateY(0deg) rotateX(0deg) translateZ(2em) translateY(8.1em)");

}

Reverse will close the cube again by hiding the faces of the cube one by one and transform it back to a 3D cube.

//Reverse Cube for Computer

function reversePcube(callback) {

    setTimeout(() => {

        // Fade out and hide the back element

        $('.back').fadeOut(200, function () {

            $('.back').hide();

            // Fade out and hide the bottom element

            $('.right').fadeOut(200, function () {

                $('.right').hide();

                // Fade out and hide the right element

                $('.bottom').fadeOut(200, function () {

                    $('.bottom').hide();

                    // Fade out and hide the left element

                    $('.left').fadeOut(200, function () {

                        $('.left').hide();

                        // Fade out and hide the top element

                        $('.top').fadeOut(200, function () {

                            $('.top').hide();

                            // Reset shape to cube

                            $(".front").css("transform", "rotateY(0deg) translateZ(2em)");

                            $(".top").css("transform", "rotateX(90deg) translateZ(2em)");

                            $(".right").css("transform", "rotateY(90deg) translateZ(2em)");

                            $(".left").css("transform", "rotateY(-90deg) translateZ(2em)");

                            $(".bottom").css("transform", "rotateX(-90deg) translateZ(2em)");

                            $(".back").css("transform", "rotateY(180deg) translateZ(2em)");

                            // Reset image faces

                            $("#frontImage").attr('src', './Pictures/Cube Faces/S.png');

                            $("#topImage").attr('src', './Pictures/Cube Faces/E.png');

                            $("#rightImage").attr('src', './Pictures/Cube Faces/C.png');

                            $("#leftImage").attr('src', './Pictures/Cube Faces/C.png');

                            $("#bottomImage").attr('src', './Pictures/Cube Faces/E.png');

                            $("#backImage").attr('src', './Pictures/Cube Faces/S.png');

                            // Show the hidden elements

                            $('.top, .right, .left, .bottom, .back').show();

                            // Call the callback function when animations are complete

                            if (typeof callback === 'function') {

                                callback();

                            }

                        });

                    });

                });

            });

        });

    }, 200);

}

Computer Display Function is the function that is called to open the cube for computer form. Starts by changing the pictures to blank then showing displaying the cube one by one as the move out and then the location where the faces are going to end at.

// Cube Rotation Computer

function computerDisplay() {

    //Stops spinning hide all

    $('.top, .right, .left, .bottom, .back').hide();

    //Change images to Blank and logo

    $('#frontImage').attr('src', './Pictures/Cube Faces/Blank.png');

    $('#topImage').attr('src', './Pictures/Cube Faces/Blank.png');

    $('#leftImage').attr('src', './Pictures/Cube Faces/Blank.png');

    $('#rightImage').attr('src', './Pictures/Cube Faces/Blank.png');

    $('#backImage').attr('src', './Pictures/Cube Faces/Logo.png');

    $('#bottomImage').attr('src', './Pictures/Cube Faces/Blank.png');

    setTimeout(() => {

        $('.top').fadeIn(200, function() {

            $('.top').show();

            $('.left').fadeIn(200, function() {

                $('.left').show();

                $('.right').fadeIn(200, function() {

                    $('.right').show();

                    $('.bottom').fadeIn(200, function() {

                        $('.bottom').show();

                        $('.back').fadeIn(200, function() {

                            $('.back').show();

                        });

                    });

                });

            });

        });

    });

    // Reset shape to cube

    $(".front").css("transform", "rotateY(0deg) translateZ(2em)");

    $(".top").css("transform", "rotateX(180deg) rotateZ(0deg) translateY(4.1em) translateZ(-2em)");

    $(".right").css("transform", "rotateY(180deg) rotateX(180deg) rotateZ(180deg) translateZ(2em) translateX(4.1em)");

    $(".left").css("transform", "rotateY(0deg) translateZ(2em) translateX(-4.1em) translateY(0em)");

    $(".bottom").css("transform", "rotateX(0deg) translateZ(2em) translateX(8.2em)");

    $(".back").css("transform", "rotateY(0deg) rotateX(0deg) translateZ(2em) translateX(4.1em) translateY(4.1em)");

}

Reverse will close the cube again by hiding the faces of the cube one by one and transform it back to a 3D cube.

//Reverse Cube for Computer

function reverseCcube(callback) {

    setTimeout(() => {

        // Fade out and hide the back element

        $('.back').fadeOut(200, function () {

            $('.back').hide();

            // Fade out and hide the bottom element

            $('.bottom').fadeOut(200, function () {

                $('.bottom').hide();

                // Fade out and hide the right element

                $('.right').fadeOut(200, function () {

                    $('.right').hide();

                    // Fade out and hide the left element

                    $('.left').fadeOut(200, function () {

                        $('.left').hide();

                        // Fade out and hide the top element

                        $('.top').fadeOut(200, function () {

                            $('.top').hide();

                            // Reset shape to cube

                            $(".front").css("transform", "rotateY(0deg) translateZ(2em)");

                            $(".top").css("transform", "rotateX(90deg) translateZ(2em)");

                            $(".right").css("transform", "rotateY(90deg) translateZ(2em)");

                            $(".left").css("transform", "rotateY(-90deg) translateZ(2em)");

                            $(".bottom").css("transform", "rotateX(-90deg) translateZ(2em)");

                            $(".back").css("transform", "rotateY(180deg) translateZ(2em)");

                            // Reset image faces

                            $("#frontImage").attr('src', './Pictures/Cube Faces/S.png');

                            $("#topImage").attr('src', './Pictures/Cube Faces/E.png');

                            $("#rightImage").attr('src', './Pictures/Cube Faces/C.png');

                            $("#leftImage").attr('src', './Pictures/Cube Faces/C.png');

                            $("#bottomImage").attr('src', './Pictures/Cube Faces/E.png');

                            $("#backImage").attr('src', './Pictures/Cube Faces/S.png');

                            // Show the hidden elements

                            $('.top, .right, .left, .bottom, .back').show();

                            // Call the callback function when animations are complete

                            if (typeof callback === 'function') {

                                callback();

                            }

                        });

                    });

                });

            });

        });

    }, 200);

}