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">

    <title>movie reviews</title>

    <link rel="stylesheet" href="/mainprpj.css">

</head>

<body>

  <nav>

    <section id="header">

        <div class="header container">

          <div class="nav-bar">

            <div class="nav-list">

            <div class="logo">

              <center>  <img src="/logo.png" alt=""></center>

            </div>

              <ul>

                <li><button type="button" class="btn btn-outline-primary"><a href="#home" data-after="Home"><div class="card"><b>Home</b></div></button></a></li>

                  <li><button type="button" class="btn btn-outline-secondary"><a href="#cast" data-after="Cast"><div class="card"><b>Cast</b></div></button></a></li>

                 <li> <button type="button" class="btn btn-outline-success"><a href="#trailer" data-after="Trailer"><div class="card"><b>Trailer</b></div></button></a></li>

                 <li> <button type="button" class="btn btn-outline-danger"><a href="#songs" data-after="songs"><div class="card"><b>Songs</b></div></button></a></li>

                 <li> <button type="button" class="btn btn-outline-info"><a href="#watchnow" data-after="Watch\_Now"><div class="card"><b>Watch Now</b></div></button></a></li>

                 <li> <button type="button" class="btn btn-outline-light"><a href="#contactus" data-after="Contact Us"><div class="card"><b>Contact us</b></div></button></a></li>

              </ul>

            </div>

          </div>

          </nav>

        </div>

      </section>

      <section id="home">

      </section>

       <img class="backimg" src="/5086852.jpg"   />

       <section id="cast">

         <div class="containercast">

           <div class="card">

             <div class="content">

               <h2><img  class="castimg"src="/01.jfif" alt=""></h2>

               <h3>Pawan Kalyan</h3>

             </div>

           </div>

           <div class="card">

            <div class="content">

              <h2><img class="castimg" src="/02.jfif" alt=""></h2>

              <h3>Nivetha Thomas</h3>

            </div>

          </div>

          <div class="card">

            <div class="content">

              <h2><img  class="castimg"src="/03.jfif" alt=""></h2>

              <h3>Anjali</h3>

            </div>

          </div>

          <div class="card">

            <div class="content">

              <h2><img class="castimg" src="/04.jfif" alt=""></h2>

              <h3>Shruthi Hassan</h3>

            </div>

          </div>

           <div class="card">

            <div class="content">

              <h2><img  class="castimg"src="/05.jfif" alt=""></h2>

              <h3>Prakash Raj</h3>

            </div>

          </div>

          <div class="card">

            <div class="content">

              <h2><img class="castimg" src="/06.jfif" alt=""></h2>

              <h3>Vamsi Krishna</h3>

            </div>

          </div>

          <div class="card">

            <div class="content">

              <h2><img  class="castimg"src="/07.jfif" alt=""></h2>

              <h3>Mukesh Rishi</h3>

            </div>

          </div>

         </div>

       </section>

       <section id="trailer">

        <iframe width="10000" height="980"

        src="https://www.youtube.com/embed/P1xKV0Dmetg">

        </iframe>

       </section>

       <section id="songs">

          <div class="songscontainer">

            <div class="service-top">

              <h1 class="section-title">Vakeel Saab Songs</h1>

            </div>

            <div class="service-bottom">

              <div class="service-item">

                <div class="card"> <iframe width="420" height="315"

                  src="https://www.youtube.com/embed/-MAu6uPJ89o">

                  </iframe></div>

                <h2>Maguva Maguva</h2>

              </div>

              <div class="service-item">

                <div class="card"> <iframe width="420" height="315"

                  src="https://www.youtube.com/embed/SBMZA5-pe30">

                  </iframe></div>

                <h2>Sathyameva Jayathe</h2>

              </div>

              <div class="service-item">

                <div class="card"> <iframe width="420" height="315"

                  src="https://www.youtube.com/embed/OpgZzj\_Xjzw">

                  </iframe></div>

                <h2>Kanti Papa</h2>

              </div>

              <div class="service-item">

                <div class="card"> <iframe width="420" height="315"

                  src="https://www.youtube.com/embed/6nDPbwOAC8k">

                  </iframe></div>

                <h2>Kadhulu Kadhulu</h2>

              </div>

            </div>

          </div>

       </section>

<section id="watchnow">

  <a href="https://www.primevideo.com/">

  <img class="card" src="/icons8-amazon-prime-video-144.png"   /></a>

  <button class="card">  <a href="https://www.amazon.com/Vakeel-Saab-Pawan-Kalyan/dp/B093T6PHNX"> Watch Now </a></button>

</section>

<section id="contactus">

  <h1 >Contact Us</h1>

  <div class="containerus">

    <div class="card">

      <div class="content">

        <h2><img src="/icons8-telephone-128.png" alt=""></h2>

        <h3> +91 86666 68888 <br>

             +91 89999 98888

        </h3>

      </div>

    </div>

    <div class="card">

     <div class="content">

       <h2><img src="/icons8-composing-mail-96.png" alt=""></h2>

       <h3 class="mail">abcd@reddit.com <br>

           abcd@gmail.com

       </h3>

     </div>

   </div>

   <div class="card">

    <div class="content">

      <h2><img src="/icons8-instagram-96.png" alt=""></h2>

      <h3 class="insta">style\_designs3739</h3>

    </div>

  </div>

</section>

<section id="footer">

  <div class="footer container">

    <div class="brand"><center><h1>Movie <span>Reviews</span></h1></center></div>

  <center> <p>Copyright © 2021 Bhushan Shelke. All rights reserved</p></center>

  </div>

</section>

<script type="text/javascript" src="vanilla-tilt.js"></script>

<script type="text/javascript">

  VanillaTilt.init(document.querySelectorAll(".card"), {

    max: 25,

    speed: 400

  });

</script>

</body>

</html>

CSS:-

@import 'https://fonts.googleapis.com/css?family=Montserrat:300, 400, 700&display=swap';

\* {

    padding: 0;

    margin: 0;

    box-sizing: border-box;

}

html {

    font-size: 10px;

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

    scroll-behavior: smooth;

}

nav{

    margin: auto;

    padding: auto;

    width: 100%;

    height: auto;

}

.nav-bar{

    width: auto;

    visibility: initial;

    opacity: inherit;

    background-color: rgba(75, 47, 47, 0.041);

}

.nav-list ul {

    justify-content: space-evenly;

    display: flex;

    text-decoration: none;

    list-style: none;

}

.logo img{

    align-items: center;

    height: 100px;

    margin-bottom: 10px;

    opacity: 0.9;

    transition: 0.3s ease-in;

    z-index: 1;

}

.nav-list ul li button{

    font-size: 30px;

    text-decoration: none;

    justify-content: right;

    margin-bottom: 10px;

    border-radius: 10px;

    box-sizing: border-box;

    padding: 10px;

    background-color: rgb(238, 255, 0);

    box-shadow: 5px black;

}

.header {

    position: fixed;

    z-index: 1000;

    left: 0;

    top: 0;

    width: 100vw;

    height: auto;

}

.backimg{

    width: 100%;

    height: 1%;

    object-fit: cover;

    margin-top: 50px;

    position: relative;

    margin-bottom: 400px;

}

  .nav-list ul li a{

      list-style: none;

      text-decoration: none;

      color: black;

  }

  .nav-list ul li a :hover{

      transition: 0.3s ease-in;

      color: cornflowerblue;

  }

  #cast{

display: flex;

justify-content: center;

align-items: center;

background: linear-gradient(rgb(238, 255, 0),rgb(64, 64, 65));

margin-bottom: 400px;

  }

   #cast::before{

      content: '';

      position: absolute;

      top: 10000;

      left: 6000;

      width: 100%;

      height: 100%;

  }

  .containercast{

      position: relative;

      display: flex;

      justify-content: center;

      align-items: center;

      max-width: 1200px;

      flex-wrap: wrap;

      z-index: 1;

      margin-bottom: 4px;

  }

  .containercast .card{

      position: relative;

      width: 280px;

      height: 400px;

      margin: 30px;

      box-shadow: 20px 20px 50px rgba(0, 0, 0,0.5);

      border-radius: 15px;

      background: rgba(255, 255, 255, 0.1);

      overflow: hidden;

      display: flex;

      justify-content: center;

      border-top: 1px solid rgba(255,255, 255, 0.5);

      border-left: 1px solid rgba(255,255, 255, 0.5);

  }

   .containercast .card .content .castimg{

      padding: 20px;

      text-align: center;

      transform: translateY(100px);

      opacity: 10;

      transition: 0.5s;

  }

  .containercast.card:hover .content .castimg{

    transform: translateY(0px);

    opacity: 1;

  }

  .containercast .card .content  .castimg {

      position: absolute;

      align-items: center;

      width: 70%;

      height: 55%;

      top: 10px;

      left: 40px;

  }

   .containercast .card .content  h3{

       margin-top: 60px;

      font-size: 3.0em;

      color: rgb(10, 3, 3);

      z-index: 1;

      font-weight: 700;

      font-family:Cambria, Cochin, Georgia, Times, 'Times New Roman', serif ;

    }

    #trailer{

        align-items: center;

        display: flex;

        justify-content: center;

        margin-bottom: 100px;

        margin-top: 100px;

        background: linear-gradient(rgb(238, 255, 0),rgb(64, 64, 65));

    } \*/

     #trailer iframe{

         height: 1000px;

        margin-bottom: 500px;

        margin-top: 400px;

        background: linear-gradient(rgb(238, 255, 0),rgb(64, 64, 65));

    }

    #trailer iframe::before{

        content: 'Trailer';

        position: absolute;

        top: 10000;

        left: 6000;

        width: 100%;

        height: 100%;

    }

    #songs{

        display: flex;

        justify-content: center;

        align-items: center;

        background: linear-gradient(rgb(238, 255, 0),rgb(64, 64, 65));

        margin-bottom: 400px;

          }

           #songs::before{

              content: '';

              position: absolute;

              top: 10000;

              left: 6000;

              width: 100%;

              height: 100%;

          }

          .service-item {

            position: relative;

            display: flex;

            justify-content: center;

            align-items: center;

            max-width: 1200px;

            flex-wrap: wrap;

            z-index: 1;

            margin-bottom: 4px;

        }

         .section-title {

             padding: 15px 15px;

            max-width: 500px;

            margin: 0 auto;

            font-size: 50px;

            text-decoration: none;

            text-align: center;

            border: 10px solid black;

            border-radius: 15px;

            box-shadow: 20px 20px 50px rgba(0, 0, 0,0.5);

        }

        #songs .songscontainer {

            flex-direction: row;

            text-align: center;

            max-width: 1500px;

            margin: 0 auto;

            padding: 100px 0;

        }

        .songscontainer .service-item{

            position: relative;

            left: 5px;

            width: 500px;

            height: 400px;

            margin: 30px;

            box-shadow: 20px 20px 50px rgba(0, 0, 0,0.5);

            border-radius: 15px;

            background: rgba(255, 255, 255, 0.1);

            overflow: hidden;

            display: flex;

            justify-content: center;

            border-top: 1px solid rgba(255,255, 255, 0.5);

            border-left: 1px solid rgba(255,255, 255, 0.5);

        }

        .service-item h2{

            font-size:  20px;

            font-weight: 1500;

            text-align: center;

            text-decoration: none;

        }

        #watchnow{

            margin: auto;

            display: flex;

            align-items: center;

            position: relative;

            justify-content: space-around;

            height: 650px;

            margin-bottom: 300px;

            background: linear-gradient(rgb(238, 255, 0),rgb(64, 64, 65));

        }

          #watchnow img{

              position: absolute;

          top: 30px;

          width: 25%;

          margin-top: 10px;

          margin-left: 11pc;

        }

        #watchnow button a{

            text-decoration: none;

            font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;

            color: black;

            font-weight: 5000px;

        }

        #watchnow button{

            display: inline-block;

            padding: 10px 30px;

            color: white;

            margin-right: 5pc;

            font-weight: 400;

            background-color: transparent;

            border: 4px solid rgb(15, 7, 9);

            font-size: 5rem;

            text-transform: uppercase;

            letter-spacing: .1rem;

            margin-top: 1pc;

            border-radius: 10px;

            transition: .3s ease;

            transition-property: background-color, color;

        }

        #watchnow button:hover{

            color: black;

            background: white;

        }

        #contactus{

            display: flex;

            justify-content: center;

            align-items: center;

            background: linear-gradient(rgb(238, 255, 0),rgb(64, 64, 65));

            margin-bottom: 1000px;

            height: 950px;

            margin: auto;

              }

               #contactus::before{

                  content: '';

                  position: absolute;

                  top: 10000;

                  left: 6000;

                  width: 100%;

                  height: 100%;

              }

              .containerus{

                  position: relative;

                  display: flex;

                  justify-content: center;

                  align-items: center;

                  max-width: 1200px;

                  flex-wrap: wrap;

                  z-index: 1;

                  margin-bottom: 4px;

              }

              .containerus .card{

                  position: relative;

                  width: 280px;

                  height: 400px;

                  margin: 30px;

                  box-shadow: 20px 20px 50px rgba(0, 0, 0,0.5);

                  border-radius: 15px;

                  background: rgba(255, 255, 255, 0.1);

                  overflow: hidden;

                  display: flex;

                  justify-content: center;

                  border-top: 1px solid rgba(255,255, 255, 0.5);

                  border-left: 1px solid rgba(255,255, 255, 0.5);

              }

              #contactus h1{

                font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;

                font-size: 6.5em;

               justify-content: center;

                display: flex;

                padding-right: 35pc;                }

                #contactus .containerus .card .content h2{

                    padding-top: 10px;

                    align-content: center;

                    justify-content: center;

                }

                #contactus  .content h3{

                    font-size: 2.0em;

                    padding-top: 5pc;

                }

                #contactus  .content .mail{

                    font-size: 2.0em;

                    padding-top: 7pc;

                }

                #contactus  .content .insta {

                    font-size: 2.0em;

                    padding-top: 7pc;

                }

                #footer {

                    background-image: linear-gradient(60deg, #29323c 0%, #485563 100%);

                    margin-top: 5pc;

                    height: 9pc;margin-bottom: 1%;

                }

                #footer .footer {

                    flex-direction: column;

                    padding-top: 50px;

                }

              #footer .brand{

                  align-items: center;

                  display: flex;

                  flex-wrap: wrap;

              }

              #footer h1{

                  font-size: 4em;

                  color: rgba(255,255, 255, 0.5);

                  padding-left: 60pc;

                 /\* top: -8% \*/

                 ;             }

                 #footer h1 span{

                     color: seashell;

                 }

                #footer p {

                    color: white;

                    padding-top: 1pc;

                    font-size: 2.3rem;

                    align-items: center;

                }

JAVASCRIPT:-

var VanillaTilt = (function () {

'use strict';

class VanillaTilt {

  constructor(element, settings = {}) {

    if (!(element instanceof Node)) {

      throw ("Can't initialize VanillaTilt because " + element + " is not a Node.");

    }

    this.width = null;

    this.height = null;

    this.clientWidth = null;

    this.clientHeight = null;

    this.left = null;

    this.top = null;

    // for Gyroscope sampling

    this.gammazero = null;

    this.betazero = null;

    this.lastgammazero = null;

    this.lastbetazero = null;

    this.transitionTimeout = null;

    this.updateCall = null;

    this.event = null;

    this.updateBind = this.update.bind(this);

    this.resetBind = this.reset.bind(this);

    this.element = element;

    this.settings = this.extendSettings(settings);

    this.reverse = this.settings.reverse ? -1 : 1;

    this.glare = VanillaTilt.isSettingTrue(this.settings.glare);

    this.glarePrerender = VanillaTilt.isSettingTrue(this.settings["glare-prerender"]);

    this.fullPageListening = VanillaTilt.isSettingTrue(this.settings["full-page-listening"]);

    this.gyroscope = VanillaTilt.isSettingTrue(this.settings.gyroscope);

    this.gyroscopeSamples = this.settings.gyroscopeSamples;

    this.elementListener = this.getElementListener();

    if (this.glare) {

      this.prepareGlare();

    }

    if (this.fullPageListening) {

      this.updateClientSize();

    }

    this.addEventListeners();

    this.updateInitialPosition();

  }

  static isSettingTrue(setting) {

    return setting === "" || setting === true || setting === 1;

  }

  /\*\*

   \* Method returns element what will be listen mouse events

   \* @return {Node}

   \*/

  getElementListener() {

    if (this.fullPageListening) {

      return window.document;

    }

    if (typeof this.settings["mouse-event-element"] === "string") {

      const mouseEventElement = document.querySelector(this.settings["mouse-event-element"]);

      if (mouseEventElement) {

        return mouseEventElement;

      }

    }

    if (this.settings["mouse-event-element"] instanceof Node) {

      return this.settings["mouse-event-element"];

    }

    return this.element;

  }

  /\*\*

   \* @return {Node}

   \*/

  addEventListeners() {

    this.onMouseEnterBind = this.onMouseEnter.bind(this);

    this.onMouseMoveBind = this.onMouseMove.bind(this);

    this.onMouseLeaveBind = this.onMouseLeave.bind(this);

    this.onWindowResizeBind = this.onWindowResize.bind(this);

    this.onDeviceOrientationBind = this.onDeviceOrientation.bind(this);

    this.elementListener.addEventListener("mouseenter", this.onMouseEnterBind);

    this.elementListener.addEventListener("mouseleave", this.onMouseLeaveBind);

    this.elementListener.addEventListener("mousemove", this.onMouseMoveBind);

    if (this.glare || this.fullPageListening) {

      window.addEventListener("resize", this.onWindowResizeBind);

    }

    if (this.gyroscope) {

      window.addEventListener("deviceorientation", this.onDeviceOrientationBind);

    }

  }

  /\*\*

   \*/

  removeEventListeners() {

    this.elementListener.removeEventListener("mouseenter", this.onMouseEnterBind);

    this.elementListener.removeEventListener("mouseleave", this.onMouseLeaveBind);

    this.elementListener.removeEventListener("mousemove", this.onMouseMoveBind);

    if (this.gyroscope) {

      window.removeEventListener("deviceorientation", this.onDeviceOrientationBind);

    }

    if (this.glare || this.fullPageListening) {

      window.removeEventListener("resize", this.onWindowResizeBind);

    }

  }

  destroy() {

    clearTimeout(this.transitionTimeout);

    if (this.updateCall !== null) {

      cancelAnimationFrame(this.updateCall);

    }

    this.reset();

    this.removeEventListeners();

    this.element.vanillaTilt = null;

    delete this.element.vanillaTilt;

    this.element = null;

  }

  onDeviceOrientation(event) {

    if (event.gamma === null || event.beta === null) {

      return;

    }

    this.updateElementPosition();

    if (this.gyroscopeSamples > 0) {

      this.lastgammazero = this.gammazero;

      this.lastbetazero = this.betazero;

      if (this.gammazero === null) {

        this.gammazero = event.gamma;

        this.betazero = event.beta;

      } else {

        this.gammazero = (event.gamma + this.lastgammazero) / 2;

        this.betazero = (event.beta + this.lastbetazero) / 2;

      }

      this.gyroscopeSamples -= 1;

    }

    const totalAngleX = this.settings.gyroscopeMaxAngleX - this.settings.gyroscopeMinAngleX;

    const totalAngleY = this.settings.gyroscopeMaxAngleY - this.settings.gyroscopeMinAngleY;

    const degreesPerPixelX = totalAngleX / this.width;

    const degreesPerPixelY = totalAngleY / this.height;

    const angleX = event.gamma - (this.settings.gyroscopeMinAngleX + this.gammazero);

    const angleY = event.beta - (this.settings.gyroscopeMinAngleY + this.betazero);

    const posX = angleX / degreesPerPixelX;

    const posY = angleY / degreesPerPixelY;

    if (this.updateCall !== null) {

      cancelAnimationFrame(this.updateCall);

    }

    this.event = {

      clientX: posX + this.left,

      clientY: posY + this.top,

    };

    this.updateCall = requestAnimationFrame(this.updateBind);

  }

  onMouseEnter() {

    this.updateElementPosition();

    this.element.style.willChange = "transform";

    this.setTransition();

  }

  onMouseMove(event) {

    if (this.updateCall !== null) {

      cancelAnimationFrame(this.updateCall);

    }

    this.event = event;

    this.updateCall = requestAnimationFrame(this.updateBind);

  }

  onMouseLeave() {

    this.setTransition();

    if (this.settings.reset) {

      requestAnimationFrame(this.resetBind);

    }

  }

  reset() {

    this.event = {

      clientX: this.left + this.width / 2,

      clientY: this.top + this.height / 2

    };

    if (this.element && this.element.style) {

      this.element.style.transform = `perspective(${this.settings.perspective}px) ` +

        `rotateX(0deg) ` +

        `rotateY(0deg) ` +

        `scale3d(1, 1, 1)`;

    }

    this.resetGlare();

  }

  resetGlare() {

    if (this.glare) {

      this.glareElement.style.transform = "rotate(180deg) translate(-50%, -50%)";

      this.glareElement.style.opacity = "0";

    }

  }

  updateInitialPosition() {

    if (this.settings.startX === 0 && this.settings.startY === 0) {

      return;

    }

    this.onMouseEnter();

    if (this.fullPageListening) {

      this.event = {

        clientX: (this.settings.startX + this.settings.max) / (2 \* this.settings.max) \* this.clientWidth,

        clientY: (this.settings.startY + this.settings.max) / (2 \* this.settings.max) \* this.clientHeight

      };

    } else {

      this.event = {

        clientX: this.left + ((this.settings.startX + this.settings.max) / (2 \* this.settings.max) \* this.width),

        clientY: this.top + ((this.settings.startY + this.settings.max) / (2 \* this.settings.max) \* this.height)

      };

    }

    let backupScale = this.settings.scale;

    this.settings.scale = 1;

    this.update();

    this.settings.scale = backupScale;

    this.resetGlare();

  }

  getValues() {

    let x, y;

    if (this.fullPageListening) {

      x = this.event.clientX / this.clientWidth;

      y = this.event.clientY / this.clientHeight;

    } else {

      x = (this.event.clientX - this.left) / this.width;

      y = (this.event.clientY - this.top) / this.height;

    }

    x = Math.min(Math.max(x, 0), 1);

    y = Math.min(Math.max(y, 0), 1);

    let tiltX = (this.reverse \* (this.settings.max - x \* this.settings.max \* 2)).toFixed(2);

    let tiltY = (this.reverse \* (y \* this.settings.max \* 2 - this.settings.max)).toFixed(2);

    let angle = Math.atan2(this.event.clientX - (this.left + this.width / 2), -(this.event.clientY - (this.top + this.height / 2))) \* (180 / Math.PI);

    return {

      tiltX: tiltX,

      tiltY: tiltY,

      percentageX: x \* 100,

      percentageY: y \* 100,

      angle: angle

    };

  }

  updateElementPosition() {

    let rect = this.element.getBoundingClientRect();

    this.width = this.element.offsetWidth;

    this.height = this.element.offsetHeight;

    this.left = rect.left;

    this.top = rect.top;

  }

  update() {

    let values = this.getValues();

    this.element.style.transform = "perspective(" + this.settings.perspective + "px) " +

      "rotateX(" + (this.settings.axis === "x" ? 0 : values.tiltY) + "deg) " +

      "rotateY(" + (this.settings.axis === "y" ? 0 : values.tiltX) + "deg) " +

      "scale3d(" + this.settings.scale + ", " + this.settings.scale + ", " + this.settings.scale + ")";

    if (this.glare) {

      this.glareElement.style.transform = `rotate(${values.angle}deg) translate(-50%, -50%)`;

      this.glareElement.style.opacity = `${values.percentageY \* this.settings["max-glare"] / 100}`;

    }

    this.element.dispatchEvent(new CustomEvent("tiltChange", {

      "detail": values

    }));

    this.updateCall = null;

  }

  prepareGlare() {

    if (!this.glarePrerender) {

      // Create glare element

      const jsTiltGlare = document.createElement("div");

      jsTiltGlare.classList.add("js-tilt-glare");

      const jsTiltGlareInner = document.createElement("div");

      jsTiltGlareInner.classList.add("js-tilt-glare-inner");

      jsTiltGlare.appendChild(jsTiltGlareInner);

      this.element.appendChild(jsTiltGlare);

    }

    this.glareElementWrapper = this.element.querySelector(".js-tilt-glare");

    this.glareElement = this.element.querySelector(".js-tilt-glare-inner");

    if (this.glarePrerender) {

      return;

    }

    Object.assign(this.glareElementWrapper.style, {

      "position": "absolute",

      "top": "0",

      "left": "0",

      "width": "100%",

      "height": "100%",

      "overflow": "hidden",

      "pointer-events": "none"

    });

    Object.assign(this.glareElement.style, {

      "position": "absolute",

      "top": "50%",

      "left": "50%",

      "pointer-events": "none",

      "background-image": `linear-gradient(0deg, rgba(255,255,255,0) 0%, rgba(255,255,255,1) 100%)`,

      "width": `${this.element.offsetWidth \* 2}px`,

      "height": `${this.element.offsetWidth \* 2}px`,

      "transform": "rotate(180deg) translate(-50%, -50%)",

      "transform-origin": "0% 0%",

      "opacity": "0",

    });

  }

  updateGlareSize() {

    if (this.glare) {

      Object.assign(this.glareElement.style, {

        "width": `${this.element.offsetWidth \* 2}`,

        "height": `${this.element.offsetWidth \* 2}`,

      });

    }

  }

  updateClientSize() {

    this.clientWidth = window.innerWidth

      || document.documentElement.clientWidth

      || document.body.clientWidth;

    this.clientHeight = window.innerHeight

      || document.documentElement.clientHeight

      || document.body.clientHeight;

  }

  onWindowResize() {

    this.updateGlareSize();

    this.updateClientSize();

  }

  setTransition() {

    clearTimeout(this.transitionTimeout);

    this.element.style.transition = this.settings.speed + "ms " + this.settings.easing;

    if (this.glare) this.glareElement.style.transition = `opacity ${this.settings.speed}ms ${this.settings.easing}`;

    this.transitionTimeout = setTimeout(() => {

      this.element.style.transition = "";

      if (this.glare) {

        this.glareElement.style.transition = "";

      }

    }, this.settings.speed);

  }

  /\*\*

   \* Method return patched settings of instance

   \* @param {boolean} settings.reverse - reverse the tilt direction

   \* @param {number} settings.max - max tilt rotation (degrees)

   \* @param {startX} settings.startX - the starting tilt on the X axis, in degrees. Default: 0

   \* @param {startY} settings.startY - the starting tilt on the Y axis, in degrees. Default: 0

   \* @param {number} settings.perspective - Transform perspective, the lower the more extreme the tilt gets

   \* @param {string} settings.easing - Easing on enter/exit

   \* @param {number} settings.scale - 2 = 200%, 1.5 = 150%, etc..

   \* @param {number} settings.speed - Speed of the enter/exit transition

   \* @param {boolean} settings.transition - Set a transition on enter/exit

   \* @param {string|null} settings.axis - What axis should be disabled. Can be X or Y

   \* @param {boolean} settings.glare - What axis should be disabled. Can be X or Y

   \* @param {number} settings.max-glare - the maximum "glare" opacity (1 = 100%, 0.5 = 50%)

   \* @param {boolean} settings.glare-prerender - false = VanillaTilt creates the glare elements for you, otherwise

   \* @param {boolean} settings.full-page-listening - If true, parallax effect will listen to mouse move events on the whole document, not only the selected element

   \* @param {string|object} settings.mouse-event-element - String selector or link to HTML-element what will be listen mouse events

   \* @param {boolean} settings.reset - false = If the tilt effect has to be reset on exit

   \* @param {gyroscope} settings.gyroscope - Enable tilting by deviceorientation events

   \* @param {gyroscopeSensitivity} settings.gyroscopeSensitivity - Between 0 and 1 - The angle at which max tilt position is reached. 1 = 90deg, 0.5 = 45deg, etc..

   \* @param {gyroscopeSamples} settings.gyroscopeSamples - How many gyroscope moves to decide the starting position.

   \*/

  extendSettings(settings) {

    let defaultSettings = {

      reverse: false,

      max: 15,

      startX: 0,

      startY: 0,

      perspective: 1000,

      easing: "cubic-bezier(.03,.98,.52,.99)",

      scale: 1,

      speed: 300,

      transition: true,

      axis: null,

      glare: false,

      "max-glare": 1,

      "glare-prerender": false,

      "full-page-listening": false,

      "mouse-event-element": null,

      reset: true,

      gyroscope: true,

      gyroscopeMinAngleX: -45,

      gyroscopeMaxAngleX: 45,

      gyroscopeMinAngleY: -45,

      gyroscopeMaxAngleY: 45,

      gyroscopeSamples: 10

    };

    let newSettings = {};

    for (var property in defaultSettings) {

      if (property in settings) {

        newSettings[property] = settings[property];

      } else if (this.element.hasAttribute("data-tilt-" + property)) {

        let attribute = this.element.getAttribute("data-tilt-" + property);

        try {

          newSettings[property] = JSON.parse(attribute);

        } catch (e) {

          newSettings[property] = attribute;

        }

      } else {

        newSettings[property] = defaultSettings[property];

      }

    }

    return newSettings;

  }

  static init(elements, settings) {

    if (elements instanceof Node) {

      elements = [elements];

    }

    if (elements instanceof NodeList) {

      elements = [].slice.call(elements);

    }

    if (!(elements instanceof Array)) {

      return;

    }

    elements.forEach((element) => {

      if (!("vanillaTilt" in element)) {

        element.vanillaTilt = new VanillaTilt(element, settings);

      }

    });

  }

}

if (typeof document !== "undefined") {

  /\* expose the class to window \*/

  window.VanillaTilt = VanillaTilt;

  /\*\*

   \* Auto load

   \*/

  VanillaTilt.init(document.querySelectorAll("[data-tilt]"));

}

return VanillaTilt;

}());