PIZZA ORDER FORM

CREATE A FORM LIKE THIS WITH VALIDATION AND SUBMISSION

The code

<!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>PIZZA HOUSE</title>

    <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=Montserrat:ital,wght@1,500&display=swap" rel="stylesheet" />

    <style>

        @import url("https://fonts.googleapis.com/css?family=Poppins:200,300,400,500,600,700,800,900&display=swap");

        \* {

            font-family: "Poppins", sans-serif;

            box-sizing: border-box;

*/\* border: 1px solid black; \*/*

        }

        body {

            width: 100%;

            height: 100%;

            font-size: 16px;

            background: #333333;

        }

        .container {

            height: 100vh;

            width: 100%;

            display: flex;

            align-items: right;

            justify-content: right;

            padding: 30px;

            cursor: default;

        }

        .card {

            position: relative;

            width: 15%;

            height: 60%;

            padding: 15px;

            border-radius: 20px;

            background-image: linear-gradient(135deg, #ff677d 10%, #cd6684 100%);

            box-shadow: 0 2px 20px rgba(0, 0, 0, 0.5);

            transform-style: preserve-3d;

            transform: perspective(1000px);

            display: flex;

            flex-flow: column;

            justify-content: right;

            align-items: center;

            margin: 20px;

*/\* overflow: hidden; \*/*

        }

        .shoe.container {

            height: 200px;

            width: 100%;

            display: flex;

            align-items: center;

            justify-content: right;

        }

        .shoe {

            width: 100%;

            transform: translateZ(50px);

            max-height: 145px;

            width: 240px;

            position: fixed;

            top: 10;

            left: 30;

        }

        .shoe img {

            width: 100%;

            height: 100%;

            object-fit: contain;

        }

        .category {

            transform: translatez(30px);

            font-weight: bolder;

            font-size: 20px;

            text-shadow: 0 2px 20px rgba(0, 0, 0, 0.5);

            align-self: start;

            margin-top: 150px;

        }

        .category .sub-head {

            font-size: 12px;

            color: rgba(255, 255, 255, 0.35);

            margin-bottom: 2px;

        }

        .description {

            transform: translateZ(25px);

            text-align: justify;

            text-shadow: 0 2px 20px rgba(0, 0, 0, 0.3);

        }

        body a {

            text-decoration: none;

            border: 0;

            outline: 0;

        }

        .read-more {

            margin-top: 10px;

            text-decoration: none;

            border: 0;

            outline: 0;

            height: 70%;

            padding: 10px 16px 16px;

            border-radius: 50px;

            box-shadow: 0 2px 20px rgba(0, 0, 0, 0.35);

            transform: translateZ(30px);

            font-size: 16px;

            font-weight: medium;

            box-shadow: 3px 3px 5px rgba(0, 0, 0, 0.3);

        }

        .read-more:hover {

            cursor: pointer;

            animation: shake 0.82s cubic-bezier(.36, .07, .19, .97) both;

            transform: translate3d(0, 0, 0);

            backface-visibility: hidden;

            perspective: 1000px;

        }

        @keyframes *shake* {

            10%,

            90% {

                transform: translate3d(-1px, 0, 0);

            }

            20%,

            80% {

                transform: translate3d(2px, 0, 0);

            }

            30%,

            50%,

            70% {

                transform: translate3d(-4px, 0, 0);

            }

            40%,

            60% {

                transform: translate3d(4px, 0, 0);

            }

        }

        body .js-tilt-glare {

            pointer-events: none;

            border-radius: 20px;

        }

        .nav\_links {

            padding: 0 20% 0 20%;

            margin-top: 0px;

*/\* box-shadow: 10px 10px 10px rgba(0, 0, 0, 0.5); \*/*

            list-style-type: none;

        }

        .nav\_links li {

            border-radius: 50px;

            align-items: center;

            display: inline-block;

            padding: 0px 10px;

*/\* box-shadow: 10px 0px 110px rgb(0 0 0 / 50%); \*/*

        }

        .nav\_links li:nth-child(1) {

            padding: 0 30px 0 0;

        }

        .nav\_links li a:hover {

            color: #000000;

        }

        nav {

            padding: 0 15px 0 0;

        }

        .active {

            display: block;

        }

        li a {

            display: block;

            background: none;

            padding: 5px 16px;

            text-decoration: none;

            color: white;

        }

        li a:hover {

            background: linear-gradient(135deg, #ff677d 10%, #cd6684 100%);

            border-radius: 50px;

            transform: scale(1.1, 1.1);

            transition: transform .5s;

            box-shadow: 5px 10px 20px rgb(0 0 0 / 50%);

        }

        video {

            position: absolute;

            top: 0;

            left: 0;

            width: 100%;

            height: 100%;

            object-fit: cover;

            opacity: 0.8;

            z-index: -1;

        }

        .overlay {

            position: absolute;

            top: 0;

            left: 0;

            width: 100%;

            height: 100%;

            background: #f4037b1e;

            mix-blend-mode: overlay;

            z-index: -1;

        }

        .social {

            position: absolute;

            z-index: 10;

            bottom: 5px;

            display: flex;

            justify-content: center;

            align-items: bottom;

            margin-left: 40px;

        }

        .social li {

            list-style: none;

        }

        .social li a {

            display: inline-block;

            margin-right: 20px;

            filter: invert(1);

            transform: scale(0.5);

            transition: 0.5s;

        }

        .social li a:hover {

            transform: scale(0.5) translateY(-15px);

        }

        .log {

            width: 60px;

            height: 60px;

            position: absolute;

            left: 100px;

            top: -2px;

            display: flex;

        }

        .text-box {

            color: #fff;

            position: absolute;

            TOP: 250PX;

            margin-left: 100px;

        }

        .text-box p {

            font-size: 50px;

            font-weight: 600;

            margin: 0;

        }

        .text-box h1 {

            font-size: 190px;

            line-height: 160px;

            margin: -10px;

            color: transparent;

            -webkit-text-stroke: 1px #fff;

            background: url(*./x/back.png*);

            background-position: 0 0;

            animation: back 20s linear infinite;

        }

        .text-box h3 {

            font-size: 40px;

            font-weight: 500;

            margin: 0;

        }

        @media (max-width: 991px) {

            .showcase,

            .showcase header {

                padding: 40px;

            }

            .text h2 {

                font-size: 3em;

            }

            .text h3 {

                font-size: 2em;

            }

        }

        .display {

            align-items: center;

            border: none;

            padding: 0px 10px 0px 10px;

            border-radius: 20px;

            background: linear-gradient(135deg, #ff677d 10%, #cd6684 100%);

        }

    </style>

</head>

<body>

    <div class="log">

        <img src="pizza design logo.png" alt="pizza logo">

    </div>

    <div class="nav">

        <nav>

            <ul class="nav\_links">

                <li><a href="#home"><b>Home</b></a></li>

                <li><a href="#contact"><b>Contact</b></a></li>

                <li><a href="#about"><b>About</b></a></li>

            </ul>

        </nav>

    </div>

    <div class="text-box">

        <p>LIMITED TIME</p>

        <h1>50% <br>OFF</h1>

        <h3>ON VEGETARIAN PIZZA</h3>

    </div>

    <div class="video">

        <video src="pexels-taryn-elliott-7172273.mp4" muted loop autoplay></video></div>

    <div class="overlay"></div>

    <div class="container">

        <div class="card" data-tilt data-tilt-max="30" data-tilt-glare data-tilt-max-glare="0.55">

            <div class="shoe">

                <img src="./cheese-pizza-png-images-hd-png-play-pizza-png-750\_441.png" alt="" />

            </div>

            <div class="category">

                <div class="sub-head">Delivery</div>

                Vegetarian

            </div>

            <div class="description">

                a vegetarian preparation with San Marzano sauce, goat cheese and blanched spinach flavoured with roasted garlic.

            </div>

            <a href="#">

                <button class="read-more">Order Now</button>

            </a>

        </div>

        <div class="card card2" data-tilt data-tilt-max="30" data-tilt-glare data-tilt-max-glare="0.55">

            <div class="shoe">

                <img src="./Cheese-Pizza.png" alt="" />

            </div>

            <div class="category">

                <div class="sub-head">Category</div>

                Cheese

            </div>

            <div class="description">

                a cheese preparation with San Marzano sauce, goat cheese and blanched spinach flavoured with roasted garlic.

            </div>

            <a href="#">

                <button class="read-more">Order Now</button>

            </a>

        </div>

        <div class="card card3" data-tilt data-tilt-max="30" data-tilt-glare data-tilt-max-glare="0.55">

            <div class="shoe">

                <img src="./Pepperoni-Pizza-PNG-Clipart.png" alt="" />

            </div>

            <div class="category">

                <div class="sub-head">Category</div>

                Pepperoni

            </div>

            <div class="description">

                a Pepperoni preparation with San Marzano sauce, goat cheese and blanched spinach flavoured with roasted garlic.

            </div>

            <a href="#">

                <button class="read-more">Order Now</button>

            </a>

        </div>

    </div>

    <section>

        <ul class="social">

            <li>

                <a href="#"><img src="https://i.ibb.co/x7P24fL/facebook.png" /></a>

            </li>

            <li>

                <a href="#"><img src="https://i.ibb.co/Wnxq2Nq/twitter.png" /></a>

            </li>

            <li>

                <a href="#"><img src="https://i.ibb.co/ySwtH4B/instagram.png" /></a>

            </li>

        </ul>

    </section>

    <div class="table100 ver1 m-b-110">

        <div class="table100-body js-pscroll">

            <table id="table" class="display" style=" width: 50% ">

                <thead>

                    <tr>

                        <th colspan="2">Pizza Shop 2.0</th>

                    </tr>

                </thead>

                <tbody>

                    <tr>

                        <td><label for="name">Name:</label></td>

                        <td>

                            <input type="text" id="name" />

                        </td>

                    </tr>

                    <tr>

                        <td>Pizza Topping</td>

                        <td>

                            <form class="toppings" id="form">

                                <fieldset name="userFields">

                                    <input type="radio" id="Supreme" name="Pizza" value="Supreme" />

                                    <label for="Supreme">Supreme</label>

                                    <input type="radio" id="Vegetarian" name="Pizza" value="Vegetarian" />

                                    <label for="Vegetarian">Vegetarian</label>

                                    <input type="radio" id="Hawaiian" name="Pizza" value="Hawaiian" />

                                    <label for="Hawaiian">Hawaiian</label>

                            </form>

                            </fieldset>

                        </td>

                    </tr>

                    <tr>

                        <td>Pizza Sauce</td>

                        <td>

                            <select>

              <option value="Tomato" selected>Tomato</option>

              <option value="Pepperroni">Pepperroni</option>

              <option value="Cheese">Cheese</option>

              <option value="Bread Sticks">Bread Sticks</option>

              <option value="Pine Apple">Pine Apple</option>

              <option value="Pepper">Pepper</option>

              <option value="Mashrooms">Mashrooms</option>

              <option value="Yeast">Yeast</option>

            </select>

                        </td>

                    </tr>

                    <tr>

                        <td>Optional Extras</td>

                        <td>

                            <form>

                                <input type="checkbox" id="Extra Cheese" name="Extra Cheese" value="Pizza" />

                                <label for="Extra Cheese">Extra Cheese</label>

                                <input type="checkbox" id="Gluten Free Base" name="Gluten Free Base" value="Pizza" />

                                <label for="Gluten Free Base">Gluten Free Base</label>

                            </form>

                        </td>

                    </tr>

                    <tr>

                        <td colspan="2">

                            <p style="margin: 0px 0;">Delivery Instructions:</p>

                            <textarea name="deliveryopts" style="width: 99.1%;" padding: 0; ">

                </textarea>

          </td>

        </tr>

        <tr>

          <td colspan="2 "><button type="submit ">Send my Order</button></td>

        </tr>

      </tbody>

    </table>

    </div>

        <script>

            var VanillaTilt = (function() {

                "use strict ";

                class t {

                    constructor(*t*, *e* = {}) {

                        if (!(*t* instanceof Node))

                            throw (

                                "Can 't initialize VanillaTilt because " + *t* + " is not a Node."

                            );

                        (this.width = null),

                        (this.height = null),

                        (this.left = null),

                        (this.top = null),

                        (this.transitionTimeout = null),

                        (this.updateCall = null),

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

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

                        (this.element = *t*),

                        (this.settings = this.extendSettings(*e*)),

                        (this.reverse = this.settings.reverse ? -1 : 1),

                        (this.glare = this.isSettingTrue(this.settings.glare)),

                        (this.glarePrerender = this.isSettingTrue(

                            this.settings["glare-prerender"]

                        )),

                        this.glare && this.prepareGlare(),

                            this.addEventListeners();

                    }

                    isSettingTrue(*t*) {

                        return "" === *t* || !0 === *t* || 1 === *t*;

                    }

                    addEventListeners() {

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

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

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

                        (this.onWindowResizeBind = this.onWindowResizeBind.bind(this)),

                        this.element.addEventListener(

                                "mouseenter",

                                this.onMouseEnterBind

                            ),

                            this.element.addEventListener("mousemove", this.onMouseMoveBind),

                            this.element.addEventListener(

                                "mouseleave",

                                this.onMouseLeaveBind

                            ),

                            this.glare &&

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

                    }

                    removeEventListeners() {

                        this.element.removeEventListener(

                                "mouseenter",

                                this.onMouseEnterBind

                            ),

                            this.element.removeEventListener(

                                "mousemove",

                                this.onMouseMoveBind

                            ),

                            this.element.removeEventListener(

                                "mouseleave",

                                this.onMouseLeaveBind

                            ),

                            this.glare &&

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

                    }

                    destroy() {

                        clearTimeout(this.transitionTimeout),

                            null !== this.updateCall && cancelAnimationFrame(this.updateCall),

                            this.reset(),

                            this.removeEventListeners(),

                            (this.element.vanillaTilt = null),

                            delete this.element.vanillaTilt,

                            (this.element = null);

                    }

                    onMouseEnter(*t*) {

                        this.updateElementPosition(),

                            (this.element.style.willChange = "transform"),

                            this.setTransition();

                    }

                    onMouseMove(*t*) {

                        null !== this.updateCall && cancelAnimationFrame(this.updateCall),

                            (this.event = *t*),

                            (this.updateCall = requestAnimationFrame(this.updateBind));

                    }

                    onMouseLeave(*t*) {

                        this.setTransition(),

                            this.settings.reset && requestAnimationFrame(this.resetBind);

                    }

                    reset() {

                        (this.event = {

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

                            pageY: this.top + this.height / 2,

                        }),

                        (this.element.style.transform =

                            "perspective(" +

                            this.settings.perspective +

                            "px) rotateX(0deg) rotateY(0deg) scale3d(1, 1, 1)"),

                        this.glare &&

                            ((this.glareElement.style.transform =

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

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

                    }

                    getValues() {

                        let t = (this.event.clientX - this.left) / this.width,

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

                        return (

                            (t = Math.min(Math.max(t, 0), 1)),

                            (e = Math.min(Math.max(e, 0), 1)), {

                                tiltX: (

*this*.*reverse* \*

                                    (*this*.*settings*.*max* / 2 - *t* \* *this*.*settings*.*max*)

                                ).*toFixed*(2),

                                tiltY: (

*this*.*reverse* \*

                                    (*e* \* *this*.*settings*.*max* - *this*.*settings*.*max* / 2)

                                ).*toFixed*(2),

                                percentageX: 100 \* *t*,

                                percentageY: 100 \* *e*,

                                angle: *Math*.*atan2*(

*this*.*event*.*clientX* - (*this*.*left* + *this*.*width* / 2), -(*this*.*event*.*clientY* - (*this*.*top* + *this*.*height* / 2))

                                    ) \*

                                    (180 / *Math*.*PI*),

                            }

                        );

                    }

                    updateElementPosition() {

                        let t = this.element.getBoundingClientRect();

                        (this.width = this.element.offsetWidth),

                        (this.height = this.element.offsetHeight),

                        (this.left = t.left),

                        (this.top = t.top);

                    }

                    update() {

                        let t = this.getValues();

                        (this.element.style.transform =

                            "perspective(" +

                            this.settings.perspective +

                            "px) rotateX(" +

                            ("x" === this.settings.axis ? 0 : t.tiltY) +

                            "deg) rotateY(" +

                            ("y" === this.settings.axis ? 0 : t.tiltX) +

                            "deg) scale3d(" +

                            this.settings.scale +

                            ", " +

                            this.settings.scale +

                            ", " +

                            this.settings.scale +

                            ")"),

                        this.glare &&

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

                                (this.glareElement.style.opacity = `${

                  (t.percentageY \* this.settings["max-glare"]) / 100

                }`)),

                            this.element.dispatchEvent(

                                new CustomEvent("tiltChange", {

                                    detail: t,

                                })

                            ),

                            (this.updateCall = null);

                    }

                    prepareGlare() {

                        if (!this.glarePrerender) {

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

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

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

                            e.classList.add("js-tilt-glare-inner"),

                                t.appendChild(e),

                                this.element.appendChild(t);

                        }

                        (this.glareElementWrapper =

                            this.element.querySelector(".js-tilt-glare")),

                        (this.glareElement = this.element.querySelector(

                            ".js-tilt-glare-inner"

                        )),

                        this.glarePrerender ||

                            (Object.assign(this.glareElementWrapper.style, {

                                    position: "absolute",

                                    top: "0",

                                    left: "0",

                                    width: "100%",

                                    height: "100%",

                                    overflow: "hidden",

                                }),

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

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

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

                                    "transform-origin": "0% 0%",

                                    opacity: "0",

                                }));

                    }

                    updateGlareSize() {

                        Object.assign(this.glareElement.style, {

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

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

                        });

                    }

                    onWindowResizeBind() {

                        this.updateGlareSize();

                    }

                    setTransition() {

                        clearTimeout(this.transitionTimeout),

                            (this.element.style.transition =

                                this.settings.speed + "ms " + this.settings.easing),

                            this.glare &&

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

                            (this.transitionTimeout = setTimeout(() => {

                                this.element.style.transition = "";

                                this.glare && (this.glareElement.style.transition = "");

                            }, this.settings.speed));

                    }

                    extendSettings(*t*) {

                        let e = {

                                reverse: !1,

                                max: 35,

                                perspective: 1e3,

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

                                scale: "1",

                                speed: "300",

                                transition: !0,

                                axis: null,

                                glare: !1,

                                "max-glare": 1,

                                "glare-prerender": !1,

                                reset: !0,

                            },

                            i = {};

                        for (var s in e)

                            if (s in *t*) i[s] = *t*[s];

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

                            let t = this.element.getAttribute("data-tilt-" + s);

                            try {

                                i[s] = JSON.parse(t);

                            } catch (e) {

                                i[s] = t;

                            }

                        } else i[s] = e[s];

                        return i;

                    }

                    static init(*e*, *i*) {

*e* instanceof Node && (*e* = [*e*]),

*e* instanceof NodeList && (*e* = [].slice.call(*e*)),

*e* instanceof Array &&

*e*.forEach((*e*) => {

                                "vanillaTilt" in *e* || (*e*.vanillaTilt = new t(*e*, *i*));

                            });

                    }

                }

                return (

                    "undefined" != typeof document &&

                    ((window.VanillaTilt = t),

                        t.init(document.querySelectorAll("[data-tilt]"))),

*t*

                );

            })();

*//js/onsubmit\_event.js*

*// Below Function Executes On Form Submit*

function ValidationEvent() {

*// Storing Field Values In Variables*

var name = document.getElementById("name").value;

var email = document.getElementById("email").value;

var contact = document.getElementById("contact").value;

*// Regular Expression For Email*

var emailReg = /^([w-.]+@([w-]+.)+[w-]{2,4})?$/;

*// Conditions*

if (name != ' ' && email != ' ' && contact != ' ') {

if (email.match(emailReg)) {

if (document.getElementById("Supreme").checked || document.getElementById("Vegetarian").checked || document.getElementById("Hawaiian").checked) {

if (contact.length == 10) {

alert("All type of validation has done on OnSubmit event.");

return true;

} else {

alert("The Contact No. must be at least 10 digit long!");

return false;

}

} else {

alert("You must select pizza.....!");

return false;

}

} else {

alert("Invalid Email Address...!!!");

return false;

}

} else {

alert("All fields are required.....!");

return false;

}

}

        </script>

</body>

</html>

The visual



