HTML Handle Bars.

{{!--

    Global Template: Exit Intent Pop-Up with Email Capture

    This global template displays a popup with an email capture when a user intends to exit the page.

    There are two default Styles: Dark text for lighter backgrounds or light text on darker backgrounds.

    Requirements:

    1) Set the content zone(s) to that defined in your Sitemap.

    2) Create a user attribute `emailAddress` to save the user's email to.

    Customizations:

    1) As a best practice, set the content zone to "global\_popup".

    See https://developer.evergage.com/web-integration/sitemap/contentzones for recommended names for content zones

    mapped via the Sitemap.

    2) Style options can be changed in Serverside Code.

    3) Styling of the Style options, including fallback background color, can be changed in CSS.

    4) Remove or change the default values for the configuration properties in Serverside Code.

    5) Once validated, the user's email address will be sent to Interaction Studio and set to the user's

    emailAddress attribute.

    6) Media query breakpoints can be changed in CSS.

    7) To support rich text, add the @richtext(true) decorator to configuration properties in Serverside Code.

    Handlebars expressions for these properties must be wrapped in three sets of curly braces (e.g. {{{expression}}} ).

    8) Change the value of the aria-label attributes as necessary.

    9) Change the error message in the .evg-error-msg. Styling can be changed in CSS.

--}}

<div id="evg-entry-intent-popup-email-capture" data-evg-campaign-id="{{campaign}}"

data-evg-experience-id="{{experience}}" data-evg-user-group="{{userGroup}}"

aria-label="Popup{{#if header}} - {{header}}{{/if}}" role="dialog" aria-modal="true" aria-hidden="false">

    {{#if lightbox}}

    <div class="evg-overlay" data-evg-dismissal></div>

    {{/if}}

    <div class="evg-popup evg-style-bg {{style.className}}">

      <div class="evg-left" width="50%">

          <img class="background-img" src="{{~imageUrl~}}" width="400px" height="410px">

      </div>

        <div class="evg-main-panel">

            <h1 class="evg-header evg-h1 evg-display-4 evg-style-text {{style.className}}">

                {{~header~}}

            </h1>

            {{#if subheader}}

            <h2 class="evg-subheader evg-h2 evg-style-text {{style.className}}">

                {{~subheader~}}

            </h2>

            {{/if}}

            <form class="evg-form" aria-label="Email Signup" role="form">

                   <div class="input-group evg-email-input-group">

                       <input class="evg-mobile" name="phone" type="string" id="entryPhoneInput" placeholder="Phone(Optional)" novalidate aria-label="mobile" tabindex="0">

                       <label class="floating-label evg-hide"  for="entryPhoneInput" data-evg-floating>Enter Phone Number</label>

                <span class="evg-error-msg-mobile evg-hide">

                    <div style="width:5%;display:inline-block">

                                        <svg xmlns="http://www.w3.org/2000/svg" width="14" height="14" fill="currentColor" class="bi bi-exclamation-circle bi-ex-error" viewBox="0 0 16 16">

                                                <path d="M8 15A7 7 0 1 1 8 1a7 7 0 0 1 0 14zm0 1A8 8 0 1 0 8 0a8 8 0 0 0 0 16z"/>

                                                <path d="M7.002 11a1 1 0 1 1 2 0 1 1 0 0 1-2 0zM7.1 4.995a.905.905 0 1 1 1.8 0l-.35 3.507a.552.552 0 0 1-1.1 0L7.1 4.995z"/>

                                            </svg>

                                    </div>

                  Please Enter a valid phone number.

                </span>

            <input class="evg-email" name="email" type="email" required placeholder="Email" novalidate aria-label="Email" aria-required="true" tabindex="0">

                                    <div class="input-group-append">

                                         <button class="evg-cta evg-btn evg-btn-primary evg-btn-lg {{style.className}}" id="exitEmailBtn" onsubmit="return false;"

                tabindex="1" type="button" aria-label="Submit">

                    {{~ctaText~}}

                </button>

                                    </div>

                                    <span class="evg-error-msg-empty-email evg-hide">

                                        <div style="width:5%;display:inline-block">

                                        <svg xmlns="http://www.w3.org/2000/svg" width="14" height="14" fill="currentColor" class="bi bi-exclamation-circle bi-ex-error" viewBox="0 0 16 16">

                                                <path d="M8 15A7 7 0 1 1 8 1a7 7 0 0 1 0 14zm0 1A8 8 0 1 0 8 0a8 8 0 0 0 0 16z"/>

                                                <path d="M7.002 11a1 1 0 1 1 2 0 1 1 0 0 1-2 0zM7.1 4.995a.905.905 0 1 1 1.8 0l-.35 3.507a.552.552 0 0 1-1.1 0L7.1 4.995z"/>

                                            </svg>

                                    </div>

                  Email address is required.

                </span>

                 <span class="evg-error-msg-invalid-email evg-hide">

                     <div style="width:5%;display:inline-block">

                                        <svg xmlns="http://www.w3.org/2000/svg" width="14" height="14" fill="currentColor" class="bi bi-exclamation-circle bi-ex-error" viewBox="0 0 16 16">

                                                <path d="M8 15A7 7 0 1 1 8 1a7 7 0 0 1 0 14zm0 1A8 8 0 1 0 8 0a8 8 0 0 0 0 16z"/>

                                                <path d="M7.002 11a1 1 0 1 1 2 0 1 1 0 0 1-2 0zM7.1 4.995a.905.905 0 1 1 1.8 0l-.35 3.507a.552.552 0 0 1-1.1 0L7.1 4.995z"/>

                                            </svg>

                                    </div>

                    Please Enter a valid email address.

                </span>

                                </div>

                           <!-- Footer -->

                                <div class="evg-footer" id="evg-footer" data-evg-footer-container>

                                    {{footerText}}

                                    <p>&nbsp;</p>

                                    <div>

                                        <div id="toolTipBox" class="evg-toolTip">By signing up with my mobile phone number, I agree to receive recurring automated promotional and personalized marketing text messages from Newell Brands Home Fragrance at the cell number used when signing up. marketing

                                            text messages from Newell Brands Home Consent is not a condition of any purchase. Reply HELP for help and STOP to cancel. Msg frequency varies. Msg and data rates may apply. View

                                            <a href="https://www.newellbrands.com/terms-of-use" id="entryTtTerms" target="\_blank">Terms</a> & <a href="https://privacy.newellbrands.com/index\_en.html" id="entryTtPrivacy" target="\_blank">Privacy</a>.

                                            <br><a href="#" id="ttClose" class="evg-ttClose">Close</a>

                                        </div>

                                        <div class=?toolTipCustom?>By submitting, I confirm I have read and accept your

                                            <a href="https://privacy.newellbrands.com/index\_en.html" id="entryPrivacy" target="\_blank">Privacy Statement</a> and I would like to receive marketing and/or promotional emails from Newell Brands Home Fragrance.

                                            <a href="#" id="lastTab">

                                                <span id="ttCircle">

                                                    <svg xmlns="http://www.w3.org/2000/svg" id="ttCircleSvg" width="11" height="11" class="bi bi-exclamation-circle" viewBox="0 0 16 16">

                                                        <path d="M8 15A7 7 0 1 1 8 1a7 7 0 0 1 0 14zm0 1A8 8 0 1 0 8 0a8 8 0 0 0 0 16z"/>

                                                        <path d="M7.002 11a1 1 0 1 1 2 0 1 1 0 0 1-2 0zM7.1 4.995a.905.905 0 1 1 1.8 0l-.35 3.507a.552.552 0 0 1-1.1 0L7.1 4.995z"/>

                                                    </svg>

                                                </span>

                                                <span id="ttCircleFill" style="display:none;">

                                                <svg xmlns="http://www.w3.org/2000/svg" id="ttCircleFillSvg" width="11" height="11" class="bi bi-exclamation-circle-fill" viewBox="0 0 16 16">

                                                    <path d="M16 8A8 8 0 1 1 0 8a8 8 0 0 1 16 0zM8 4a.905.905 0 0 0-.9.995l.35 3.507a.552.552 0 0 0 1.1 0l.35-3.507A.905.905 0 0 0 8 4zm.002 6a1 1 0 1 0 0 2 1 1 0 0 0 0-2z"/>

                                                </svg>

                                                </span>

                                            </a>

                                        </div>

                                    </div>

                                </div>

            </form>

        <!--commenting out opt out text <span class="evg-opt-out-msg evg-style-text {{style.className}}" tabindex="2" data-evg-dismissal>

                {{~optOutText~}}

                -->

            </span>

        </div>

         <div class="evg-confirm-panel evg-hide">

            <h1 class="evg-confirm-header evg-h1 evg-style-text {{style.className}}">

                {{~confirmationHeader~}}

            </h1>

            {{#if confirmationSubheader}}

            <h2 class="evg-confirm-subheader evg-h2 evg-style-text {{style.className}}">

                {{~confirmationSubheader~}}

            </h2>

            {{/if}}

                          <button class="evg-cta evg-btn evg-btn-primary evg-btn-lg {{style.className}} evg-submit-btn" data-evg-dismissal

                tabindex="3" type="button" aria-label="Submit"

        aria-label="Close Popup">    {{~ctaText2~}}

                </button>

        </div>

        <button class="evg-btn-dismissal evg-style-text {{style.className}}" data-evg-dismissal

        aria-label="Close Popup" tabindex="3">&times;</button>

    </div>

</div>

------------------------------------------ CSS-------------------------------------

/\* ==========================================================================

   Pop-up

========================================================================== \*/

/\*========================================================================

Tool Tip

==========================================================================\*/

#toolTipBox {

position: absolute;

    top: 96px !important;

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

    padding: 10px 0px;

        border: 2px solid #000000;

}

.background-image{

    width:95%;

}

#evg-entry-intent-popup-email-capture .evg-toolTip {

    display: none;

    color: #333;

    background: #fff;

    box-shadow: 0px 0px 6px #666;

    border: 1px solid #ccc;

    font-size: 12px;

    text-align: left;

    padding: 10px;

    width: 360px;

    position: absolute;

    z-index: 109;

    top: 70px;

    left: 52%;

}

.evg-ttClose {

    float: right;

    font-size: larger;

}

#evg-entry-intent-popup-email-capture .bi-exclamation-circle {

    fill: #4e616a;

}

#evg-entry-intent-popup-email-capture .bi-exclamation-circle-fill {

    fill: #4e616a;

}

#evg-entry-intent-popup-email-capture .evg-footer {

     padding: 15px;

     font-size: 10.8px;

    color: #696158;

    width: 100%;

}

#evg-entry-intent-popup-email-capture .bi-ex-error {

    fill: #bd1313;

}

#evg-entry-intent-popup-email-capture .evg-footer p {

    margin: 10px 0 0 0;

    word-wrap: break-word;

}

#evg-entry-intent-popup-email-capture .evg-footer a {

    color: #4e616a;

    text-decoration: none;

}

#evg-entry-intent-popup-email-capture .evg-footer-offer {

    border: 1px solid #aaa;

    background-color: #fff;

    color: #696158;

    width: 95%;

    font-size: 0.5em;

    margin-bottom: 0px;

    padding: 10px;

}

#evg-entry-intent-popup-email-capture .evg-footer-offer a {

    color: #000;

    text-decoration: underline;

}

#evg-entry-intent-popup-email-capture {

    position: relative;

    z-index: 10000000;

    background-color: #fff;

}

#evg-entry-intent-popup-email-capture .evg-overlay {

    position: fixed;

    top: 0;

    right: 0;

    bottom: 0;

    left: 0;

    opacity: .3;

    width: 100%;

    height: 100%;

    background-color: #fff;

}

#evg-entry-intent-popup-email-capture .evg-popup {

  position: fixed;

    top: 70%;

    left: 68.5%;

    width: 800px;

    height: 410px;

    padding: 0px;

    background-color: #fff;

    background-position: center bottom;

    background-size: cover;

    background-repeat: no-repeat;

    transform: translateX(-50%) translateY(-50%);

}

#evg-entry-intent-popup-email-capture .evg-btn-dismissal {

    position: absolute;

    top: 0;

    right: 0;

    padding: 0 .5rem;

    border: 1px solid transparent;

    background-color: transparent;

    font-size: 2rem;

    font-weight: 700;

    line-height: 1;

    cursor: pointer;

}

#evg-entry-intent-popup-email-capture .evg-main-panel,

#evg-entry-intent-popup-email-capture .evg-confirm-panel {

    display: -webkit-box;

    display: -ms-flexbox;

    display: flex;

    -webkit-box-orient: vertical;

    -webkit-box-direction: normal;

    -ms-flex-direction: column;

    flex-direction: column;

    -webkit-box-pack: center;

    -ms-flex-pack: center;

    justify-content: center;

    -webkit-box-align: center;

    -ms-flex-align: center;

    align-items: center;

    width: 50%;

     margin: -26% 0px 0% 47%;

    height: 0%;

    text-align: center;

}

#evg-entry-intent-popup-email-capture .evg-confirm-header {

    display: inline-block;

    margin: 0 0 .5rem;

}

#evg-entry-intent-popup-email-capture .evg-confirm-subheader {

    display: inline-block;

    margin: 0 0 .5rem;

}

#evg-entry-intent-popup-email-capture .evg-confirm-panel .evg-btn {

    width: 200px;

    padding: 20px 10px;

}

#evg-entry-intent-popup-email-capture .evg-error-msg-mobile  .evg-error{

    color: #ff0000;

    font-size: 1rem;

    display: block;

}

/\*#evg-entry-intent-popup-email-capture .evg-opt-out-msg {

    font-size: 1rem;

    text-decoration: underline;

    cursor: pointer;

}\*/

/\* ==========================================================================

   Form

========================================================================== \*/

#evg-entry-intent-popup-email-capture .evg-form {

    display: -webkit-box;

    display: -ms-flexbox;

    display: flex;

    -webkit-box-orient: vertical;

    -webkit-box-direction: normal;

    -ms-flex-direction: column;

    flex-direction: column;

    -webkit-box-pack: center;

    -ms-flex-pack: center;

    justify-content: center;

    -webkit-box-align: center;

    -ms-flex-align: center;

    align-items: center;

    width: 100%;

    text-align: center;

}

#evg-entry-intent-popup-email-capture .evg-email {

    width: 75%;

    margin: .5rem 0;

    padding: .5rem 1rem;

    border: 1px solid #000001;

    font-size: .8rem;

    color: #333;

    text-align: left;

    display: block;

    height: 3rem;

}

#evg-entry-intent-popup-email-capture .evg-mobile {

     margin: .2rem 0;

    border: 1px solid #000001;

    font-size: .8rem;

    color: #333;

    text-align: left;

    display: block;

    width: 100%;

    height: 3rem;

}

/\* ==========================================================================

   Headers

========================================================================== \*/

#evg-entry-intent-popup-email-capture .evg-h1,

#evg-entry-intent-popup-email-capture .evg-h2 {

    margin: 0 0 .5rem;

    padding: 0;

    color: #000001;

    font-weight: 500;

    font-family: inherit;

    line-height: 1.2;

}

#evg-entry-intent-popup-email-capture .evg-h1.evg-display-4 {

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

    font-weight: bold;

    font-size: 20px;

    color: #696158;

    padding-bottom: 10px;

}

#evg-entry-intent-popup-email-capture .evg-h2 {

       font-weight: bold;

    font-size: 16px;

    color: #696158;

    padding-bottom: 10px;

    line-height: 1.5;

}

.evg-email-input-group {

    margin-top: 5px;

    width: 90%;

    left: 20px;

}

/\* ==========================================================================

   Call-To-Action Button

==========================================================================

#evg-entry-intent-popup-email-capture .evg-btn {

    display: inline-block;

    vertical-align: middle;

    padding: .5rem 1rem;

    border: 1px solid transparent;

    border-radius: 3rem;

    color: #215ca0;

    background-color: transparent;

    font-size: 1rem;

    font-weight: 400;

    line-height: 1.5;

    text-align: center;

    text-decoration: none;

    transition:

        color .15s ease-in-out,

        background-color .15s ease-in-out,

        border-color .15s ease-in-out,

        box-shadow .15s ease-in-out;

    cursor: pointer;

    -webkit-user-select: none;

    -moz-user-select: none;

    -ms-user-select: none;

    user-select: none;

}

#evg-entry-intent-popup-email-capture .evg-btn-primary {

    border-color: #215ca0;

    color: #fff;

    background-color: #215ca0;

    transition: background-color .15s, color .15s;\*/

}

#evg-entry-intent-popup-email-capture .evg-btn-lg {

    padding: .5rem 2rem;

    font-size: 1.25rem;

}

#evg-entry-intent-popup-email-capture .evg-cta {

    margin: .5rem 0;

}

/\* ==========================================================================

   Error Message

========================================================================== \*/

#evg-entry-intent-popup-email-capture  .evg-error-msg-invalid-email {

    color: #ff0000;

    font-size: 1rem;

}

#evg-entry-intent-popup-email-capture  .evg-error-msg-empty-email {

    color: #ff0000;

    font-size: 1rem;

}

#evg-entry-intent-popup-email-capture  .evg-error-msg-mobile {

    color: #ff0000;

    font-size: 1rem;

}

#evg-modal-email-capture .floating-label {

    color: #ff0000;

}

/\* ==========================================================================

   Style

========================================================================== \*/

#evg-entry-intent-popup-email-capture .evg-style-bg.evg-light-on-dark {

    background-color: #ffffff;

}

#evg-entry-intent-popup-email-capture

{

    background-color: #ffffff;

}

#evg-entry-intent-popup-email-capture .evg-style-bg.evg-dark-on-light {

    background-color: #fff;

}

#evg-entry-intent-popup-email-capture .evg-style-text.evg-light-on-dark {

    color: #000001;

}

#evg-entry-intent-popup-email-capture .evg-style-text.evg-dark-on-light {

    color: #343a40;

}

/\* ==========================================================================

   Visibility

========================================================================== \*/

#evg-entry-intent-popup-email-capture .evg-hide {

    visibility: hidden;

    display: none;

    width: 0;

    height: 0;

}

/\* ==========================================================================

   Media Queries

========================================================================== \*/

@media screen and (max-height: 500px) {

    #evg-entry-intent-popup-email-capture .evg-popup {

        height: 90vh;

    }

}

@media screen and (max-width: 575px) {

    #evg-entry-intent-popup-email-capture .evg-popup {

        width: 90vw;

    }

    #evg-entry-intent-popup-email-capture .evg-h1.evg-display-4 {

        font-size: 2.75rem;

    }

    #evg-entry-intent-popup-email-capture .evg-h2 {

        font-size: 1.2rem;

    }

    #evg-entry-intent-popup-email-capture .evg-btn-lg {

        font-size: 1rem;

    }

    #evg-entry-intent-popup-email-capture .evg-error-msg,

   /\* #evg-entry-intent-popup-email-capture .evg-opt-out-msg {

        font-size: .85rem;

    }\*/

}

#evg-entry-intent-popup-email-capture .evg-flex-container {

    display: flex;

}

#evg-entry-intent-popup-email-capture .evg-flex-child {

    flex: 1;

    width: 800px;

    height: auto;

}

#evg-entry-intent-popup-email-capture .flex-child:first-child {

    margin-right: 20px;

}

#evg-entry-intent-popup-email-capture .evg-left {

    background-image: url(https://image.e.yankeecandle.com/lib/fe3e157…/m/1/6b6d93b6-f6a1-4534-a896-1a67e02e647b.jpg);

    background-repeat: no-repeat;

    background-size: cover;

    background-position: left center;

    width: 400px;

    height: 400px;

}

#evg-entry-intent-popup-email-capture .evg-right {

    width: 400px;

    position: relative;

    background: white;

}

#evg-entry-intent-popup-email-capture .evg-confirm-panel .evg-right {

    display: flex;

    flex-direction: column;

    align-items: left;

}

#evg-entry-intent-popup-email-capture .evg-btn {

    vertical-align: middle;

    width: 100px;

    text-align: center;

    text-decoration: none;

    transition: color .15s ease-in-out, background-color .15s ease-in-out, border-color .15s ease-in-out, box-shadow .15s ease-in-out;

    cursor: pointer;

    -webkit-user-select: none;

    -moz-user-select: none;

    -ms-user-select: none;

    user-select: none;

}

#evg-entry-intent-popup-email-capture .evg-btn-primary {

    background-color: #4e616a;

    color: #fff;

    border: 1px solid #4e616a;

    border-color: #4e616a;

}

#evg-entry-intent-popup-email-capture .evg-btn-primary:focus {

    z-index: 4;

    outline: 1px dashed #4e616a;

    outline-offset: 4px;

}

-------------------------------------------------- client side code----------------------------------------------

/\*v\_20\_7 with API code merged\*/

(function() {

    const apiUrl = `https://dev.apigw.newellbrands.com/v1/sfmc\_registeremail`;

    const apiKey = 'vUA3aMbj7fUXOGwVhpVisVZrl5FkHdrE';

    const callingApplication = 'Yankee Candle';

    const optinVerbiage = 'test dev\_yankee';

    const listName = 'global';

    const requestIp = '10.10.10.1000';

    // setup date params

    const callingDate = buildDateString();

    const dateTime = buildDateString(true, true);

    const optinDate = buildDateString(true, true);

    const requestID =  buildUID(8);

    const source = 'popup-entry';

    const pageName =  'home';

    const siteUrl =  Evergage.getCurrentPage().source.url

    const region = 'US';

    const locale =  'en\_US';

    const subBrand = dataLayer[0].pageinfo.hierarchy\_parent;

    /\*\*

     \* @function buildBindId

     \* @param {Object} context

     \* @description Create unique bind ID based on the campaign and experience IDs.

     \*/

    function buildBindId(context) {

        return `${context.campaign}:${context.experience}`;

    }

    /\*ToolTip\*/

    function circleSwap(state) {

        if (state == 'fill') {

            $("#ttCircle").css("display", "none");

            $("#ttCircleFill").css("display", "inline");

        } else {

            $("#ttCircle").css("display", "inline");

            $("#ttCircleFill").css("display", "none");

        }

    }

    // Should exclamation point be outline or solid

    function circleSwap(state) {

        if (state == 'fill') {

            $("#ttCircle").css("display", "none");

            $("#ttCircleFill").css("display", "inline");

        } else {

            $("#ttCircle").css("display", "inline");

            $("#ttCircleFill").css("display", "none");

        }

    }

    // Show/Hide Tooltip

    function showHideTip() {

        var toolTipBox = document.getElementById("toolTipBox");

        let ttCircleY = document.getElementById("ttCircle").offsetTop;

        toolTipBox.style.position = "absolute";

        toolTipBox.style.top = (ttCircleY - 184) + "px";

        if (toolTipBox.style.display != "block") {

            circleSwap('fill');

            $("#toolTipBox").fadeIn();

        } else {

            $("#toolTipBox").hide();

            circleSwap('outline');

        }

    }

    /\*\*

     \* @function setConfirmationPanel

     \* @description Add click listener to the Call-To-Action button that validates the user email address,

     \* shows the Confirmation Panel, removes dismissal tracking from the "X" button and overlay, and sends

     \* an event to Interaction Studio to set the emailAddress attribute to the user email address.

     \*/

    function setConfirmationPanel() {

        Evergage.cashDom("#evg-entry-intent-popup-email-capture .evg-cta").on("click", () => {

            const email = Evergage.cashDom(".evg-form input[placeholder='Email']").val();

            let mobile = Evergage.cashDom(".evg-form input[placeholder='Phone(Optional)']").val();

            console.log('mobile is'+mobile);

            const emailRegex = RegExp(/^[.-\w]+@[\w\-]{3,}((.)\w{2,})+$/);

            const phoneRegex = RegExp(/^[\+]?[(]?[0-9]{3}[)]?[-\s\.]?[0-9]{3}[-\s\.]?[0-9]{4,6}$/);

            let finalValidator="";

            finalValidator=email?emailRegex.test(email)?"validEmail":"invalidEmail":"emptyEmail";

            finalValidator+="\_&\_"+(mobile?phoneRegex.test(mobile)?"validMobile":"invalidMobile":"emptyMobile");

            console.log('email, phone  is'+finalValidator);

            // analytics attempt

            dataLayer.push({

                'event': 'email-signup-attempt',

                'form\_field\_source': 'Lightbox'

            });

            switch(finalValidator){

                case "validEmail\_&\_validMobile":

                    emailValid = true;

                    mobileValid = true;

                    handleValid("email");

                    mobile = formatPhoneNumber(mobile);

                    console.log('formatted mobile is'+mobile);

                    setupApigeeApiCall(email, mobile);

                break;

                 case "validEmail\_&\_emptyMobile":

                    emailValid = true;

                    mobileValid = true;

                    handleValid("email");

                    setupApigeeApiCall(email, '');

                break;

                case "emptyEmail\_&\_validMobile":

                    handleError("emptyEmail");

                    emailValid = false;

                    mobileValid = true;

                break;

                case "validEmail\_&\_invalidMobile":

                       emailValid = true;

                       mobileValid = false;

                       handleError( "invalid\_mobile");

                break;

                case "invalidEmail\_&\_invalidMobile":

                    handleError( "invalid\_email\_&\_invalidMobile");

                    emailValid = false;

                    mobileValid = false;

                break;

                case "invalidEmail\_&\_validMobile":

                    handleError( "invalid\_email");

                    emailValid = false;

                    mobileValid = true;

                break;

                case "invalidEmail\_&\_emptyMobile":

                    handleError("invalid\_email");

                    emailValid = false;

                    mobileValid = true;

                break;

                case "emptyEmail\_&\_invalidMobile":

                    handleError("emptyEmail\_&\_invalidMobile");

                    emailValid = false;

                    mobileValid = false;

                break;

                case "emptyEmail\_&\_emptyMobile":

                    handleError("emptyEmail");

                    emailValid = false;

                    mobileValid = true;

                break;

                default:null;

            }

        });

    }

     /\*\*

     \* @function setupapigeecall

     \* @param {Object} context

     \* @description Create request ,method and body in JSOn format to then pass to function callapigee

    \*\*/

       function setupApigeeApiCall(email, mobile) {

           console.log("function entered-->setupApigeeApiCall"+email);

        let headers = new Headers();

        headers.append('Content-Type', 'application/json');

        const body = JSON.stringify({

            'Header': {

                'APIKey': apiKey,

                'DateTime': dateTime,

                'RequestID': requestID,

                'Base': {

                    'CallingApplication': callingApplication,

                    'CallingDate': callingDate,

                    'SiteUrl': siteUrl,

                    'Source': source,

                    'RequestingIP': requestIp,

                    'ListName': listName,

                    'OptInVerbiage': optinVerbiage,

                    'PageName': pageName,

                    'OptInDate': optinDate,

                    'Region': region,

                    'sub\_brand': subBrand

                }

            },

            'WebFormDetails': {

                'Email': email,

                'Country': region

            },

            'SMSSignUp': {

                'mobilenumber': mobile,

                'subscribe': true,

                'resubscribe': true

            }

        });

        let requestOptions = {

            method: 'POST',

            headers,

            body,

            redirect: 'follow'

        };

        callApigeeApi(email, apiUrl, requestOptions, mobile);

    }

       /\*\*

         \* @function callApigee

         \* @param {Object} context

         \* @description call apigee and pass email and mobile(if avalable). Get teh contactid from call

       \*\*/

            function callApigeeApi(email, apiUrl, requestOptions, mobile) {

            console.log("function entered-->ApigeeApiCall");

            fetch(apiUrl, requestOptions)

            .then(response => {

                if(response.status!==200) {

                    throw new Error(response.status);

                }

                return response.text();

            })

            .then(result => {

            console.log(`%c${result} `, 'color:green');

                 // analytics email success

                 dataLayer.push({

                    'event': 'email-signup-success',

                    'form\_field\_source': 'Lightbox'

                });

                result = JSON.parse(result);

                if (mobile) {

                    if (!result.sms\_response && !result.sms\_response.results) {

                        if (result.sms\_response.results.length <= 0) {

                           // handleError(`There was an error adding your mobile number. Please check the format`, 'default');

                            return;

                        }

                    } else {

                        dataLayer.push({

                            'event': 'sms-signup-success',

                            'form\_field\_source': 'Lightbox'

                        });

                    }

                }

                generatePBD(result.contactResponse['sfdc.contact\_id'], email);

                updateIS(email, mobile,result.contactResponse['sfdc.contact\_id']);

            })

            .catch(error => {

                console.log(`%cError: ${error}`,  'color:red');

                // analytics fail

                 dataLayer.push({

                    'event': 'email-signup-fail',

                    'form\_field\_source': 'Lightbox',

                    'error\_message': error //Capture error number and text'

                });

                handleError(`There was an error connecting to marketing cloud. Error: ${error}`);

            });

    }

function generatePBD(contactId, email) {

        // Api Constants

        const auth = 'Basic c3RvcmVmcm9udDpOV0xlY29tbWVyY2U=';

        const authCookie = '\_\_cfduid=d6640e7a78a11050e5cc80a289093b3c51616691444; \_\_cfduid=dc4c4e5c7e48590ff88271971c7cb5a4c1616691444; \_\_cq\_dnt=1; dw\_dnt=1; dwanonymous\_e2b672b758c4413673702ba1fd48d98d=abitqlXr9CBcDztxeN1Vlw679W; dwsid=Mr0NoztkI5K29f3q73eLTka1w8OP1vXS9Ff3Tl6C8PceiL7eh-NdOaPVI0WDH1YGdWhvvSe\_bmq-A3URJukDNg==; sid=JQ5vcUINh3QZt3BlH3fr4lEGoRH2v1E0WxU';

        const PBDUrl = `https://us.dev.yankeecandle.com/on/demandware.store/Sites-homefragranceus-Site/en\_US/ContactUs-UpdateContactID`;

        // Headers

        let headers = new Headers();

        headers.append('Authorization', auth);

        headers.append('Cookie', authCookie);

        // Data

        let formdata = new FormData();

        formdata.append('contactId', contactId);

        formdata.append('email', email);

        // Options

        let requestOptions = {

            method: 'POST',

            headers: headers,

            body: formdata,

            redirect: 'follow'

        };

        fetch(PBDUrl, requestOptions)

            .then(response => {

                if (response.status !== 200) {

                    throw new Error(response.status);

                }

                return response.text();

            })

            .then(result => {

                console.log("PBD is success");

                result = JSON.parse(result);

                console.log ("check json" +result.payload);

                generateCookie(result.payload, 'custPBD', 360);

            })

            .catch(error => {

                handleError('There was an error generating PB Code');

                //console.log('error', error)

            });

    }

/\*\*

     \* @function generateCookie

     \* @description write pbd cookie

     \*/

    function generateCookie(payload, cname, xdays) {

        let d = new Date();

        d.setTime(d.getTime() + (xdays \* 24 \* 60 \* 60 \* 1000));

        const expires = 'expires=' + d.toUTCString();

        console.log("cookie expires on " + expires);

        if (getCookie(cname) === null) {

            document.cookie = cname + '=' + payload + ';' + expires + ';path=/';

            console.log('cookie created');

        } else {

            console.error('cookie exists');

        }

    }

    /\*\*

     \* @function getCookie

     \* @description find cookie by name

     \*/

    function getCookie(name) {

        var match = document.cookie.match(RegExp('(?:^|;\\s\*)' + name + '=([^;]\*)'));

        return match ? match[1] : null;

    }

/\*\*

     \* @function buildDateString

     \* @param {Boolean} addTime

     \* @description Use current day to setup a date variable

     \*/

    function buildDateString(addTime, label) {

        let d = new Date();

        let dd = String(d.getDate()).padStart(2, '0');

        let mm = String(d.getMonth() + 1).padStart(2, '0');

        let yyyy = d.getFullYear();

        let time = d.toLocaleTimeString();

        let timeLabel = (label) ? 'T' : ' ';

        time = time.replace('AM', '').replace('PM', '');

        return (addTime) ? `${mm}/${dd}/${yyyy}${timeLabel}${time}` : `${mm}/${dd}/${yyyy}`;

    }

    /\*\*

     \* @function buildUID

     \* @description create a random string uid

     \* @param {Number} length

     \*/

    function buildUID(length) {

        const chars = '0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ';

        let result = '';

        for (let i = length; i > 0; --i) result += chars[Math.floor(Math.random() \* chars.length)];

        return result;

    }

        /\*\*

             \* @function formatPhoneNumber

             \* @param {String} phoneNumberString

             \* @description normalizes phone number

             \*/

            function formatPhoneNumber(phoneNumberString) {

                let cleaned = ('' + phoneNumberString).replace(/\D/g, '');

                let match = cleaned.match(/^(1|)?(\d{3})(\d{3})(\d{4})$/);

                if (match) {

                    let intlCode = (match[1]) ? match[1] : '1';

                    return [intlCode, match[2], match[3], match[4]].join('');

                }

                return null;

            }

    /\*\*

     \* @function handleError

     \* @param {String} errorText

     \* @param {String} field

     \* @description Display error messages

     \*/

     function handleError(field) {

        switch(field){

            case "invalid\_mobile":

                Evergage.cashDom("#evg-entry-intent-popup-email-capture .evg-error-msg-mobile").removeClass("evg-hide");

                Evergage.cashDom("#evg-entry-intent-popup-email-capture .evg-error-msg-mobile").addClass("evg-error");

                Evergage.cashDom("evg-entry-intent-popup-email-capture .floating-label").removeClass("evg-hide");

                Evergage.cashDom("#evg-entry-intent-popup-email-capture .evg-confirm-panel").addClass("evg-hide");

                Evergage.cashDom("#evg-entry-intent-popup-email-capture .evg-error-msg-empty-email").addClass("evg-hide");

                      Evergage.cashDom("#evg-entry-intent-popup-email-capture .evg-error-msg-invalid-email").addClass("evg-hide");

            break;

            case "invalid\_email":

                console.log("handling email error block now-->");

                Evergage.cashDom(".evg-error-msg-invalid-email").removeClass("evg-hide")

                Evergage.cashDom(".evg-error-msg-invalid-email").addClass("evg-error");

                Evergage.cashDom("#evg-entry-intent-popup-email-capture .evg-error-msg-mobile").addClass("evg-hide");

                Evergage.cashDom("evg-entry-intent-popup-email-capture .floating-label").addClass("evg-hide");

                Evergage.cashDom("#evg-entry-intent-popup-email-capture .evg-error-msg-empty-email").addClass("evg-hide");

                Evergage.cashDom("#evg-entry-intent-popup-email-capture .evg-confirm-panel").addClass("evg-hide");

            break;

            case "invalid\_email\_&\_invalidMobile":

                Evergage.cashDom("#evg-entry-intent-popup-email-capture .evg-error-msg-mobile").removeClass("evg-hide")

                //Evergage.cashDom("#evg-modal-email-capture .floating-label").removeClass("evg-hide");;

                Evergage.cashDom("#evg-entry-intent-popup-email-capture .evg-error-msg-invalid-email").removeClass("evg-hide");

                Evergage.cashDom("#evg-entry-intent-popup-email-capture .evg-error-msg-invalid-email").addClass("evg-error");

                Evergage.cashDom("#evg-entry-intent-popup-email-capture .evg-error-msg-mobile").addClass("evg-error");

                Evergage.cashDom("#evg-entry-intent-popup-email-capture .evg-confirm-panel").addClass("evg-hide");

                Evergage.cashDom("#evg-entry-intent-popup-email-capture .evg-error-msg-empty-email").addClass("evg-hide");

            break;

            case "emptyEmail\_&\_invalidMobile":

                Evergage.cashDom("#evg-entry-intent-popup-email-capture .evg-error-msg-empty-email").removeClass("evg-hide");

                Evergage.cashDom("#evg-entry-intent-popup-email-capture .evg-error-msg-empty-email").addClass("evg-error");

                Evergage.cashDom("#evg-entry-intent-popup-email-capture .evg-error-msg-mobile").removeClass("evg-hide");

                Evergage.cashDom("#evg-entry-intent-popup-email-capture .evg-error-msg-mobile").addClass("evg-error");

                //Evergage.cashDom("#evg-modal-email-capture .floating-label").removeClass("evg-hide");

                Evergage.cashDom("#evg-entry-intent-popup-email-capture .evg-confirm-panel").addClass("evg-hide");

                Evergage.cashDom("#evg-entry-intent-popup-email-capture .evg-error-msg-invalid-email").addClass("evg-hide");

            break;

            case "emptyEmail":

                Evergage.cashDom("#evg-entry-intent-popup-email-capture .evg-error-msg-empty-email").removeClass("evg-hide");

                Evergage.cashDom("#evg-entry-intent-popup-email-capture .evg-error-msg-empty-email").addClass("evg-error");

                Evergage.cashDom("#evg-entry-intent-popup-email-capture .evg-confirm-panel").addClass("evg-hide");

                Evergage.cashDom("#evg-entry-intent-popup-email-capture .evg-error-msg-invalid-email").addClass("evg-hide");

                Evergage.cashDom("#evg-entry-intent-popup-email-capture .evg-error-msg-mobile").addClass("evg-hide");

              //Evergage.cashDom("#evg-modal-email-capture .floating-label").addClass("evg-hide");

            break;

            case "default":

                Evergage.cashDom("#evg-entry-intent-popup-email-capture .evg-error-msg-invalid-email")

                    .removeClass("evg-hide")

                    .addClass("evg-error");

                    Evergage.cashDom("#evg-entry-intent-popup-email-capture .evg-confirm-panel").addClass("evg-hide");

            break;

            default: null;

        }

    }

    /\*\*

     \* @function handleValid

     \* @param {String} field

     \* @description Hide error messages

     \*/

     function handleValid(field) {

        switch(field){

            case "mobile":

            console.log("handle valid called mobile??");

                Evergage.cashDom("#evg-entry-intent-popup-email-capture .evg-main-panel").addClass("evg-hide");

                Evergage.cashDom("#evg-entry-intent-popup-email-capture .evg-confirm-panel").removeClass("evg-hide");

                Evergage.cashDom(`

                    #evg-entry-intent-popup-email-capture .evg-btn-dismissal

                `).removeAttr("data-evg-dismissal");

            break;

            case "email":

            console.log("handle valid called??");

                Evergage.cashDom("#evg-entry-intent-popup-email-capture .evg-main-panel").addClass("evg-hide");

                Evergage.cashDom("#evg-entry-intent-popup-email-capture .evg-confirm-panel").removeClass("evg-hide");

                Evergage.cashDom(`

                    #evg-entry-intent-popup-email-capture .evg-btn-dismissal

                `).removeAttr("data-evg-dismissal");

            break;

            default:null;

        }

    }

        /\*\*

         \* @function updateIS

         \* @description update IS event to email sign up- popup and capture email, phone and contactid

         \*/

         function updateIS(email, mobile , contactId) {

            Evergage.sendEvent({

                action: "Email Sign Up - Popup",

                user: {

                    attributes: {

                        emailAddress: email,

                        email: email,

                        sfmcContactKey: contactId,

                        phone: mobile

                    }

                }

            });

        }

    /\*\*

     \* @function setDismissal

     \* @param {Object} context

     \* @description Add click listener to the overlay, "X" button, and opt-out text that removes the

     \* template from the DOM.

     \* #evg-entry-intent-popup-email-capture .evg-opt-out-msg(dissnissselector code removed as opt out text removed)

     \*/

    function setDismissal(context) {

        const dismissSelectors = `

           #evg-entry-intent-popup-email-capture  .evg-submit-btn,

            #evg-entry-intent-popup-email-capture .evg-btn-dismissal,

            #evg-entry-intent-popup-email-capture .evg-opt-out-msg

            `;

        Evergage.cashDom(dismissSelectors).on("click", () => {

            Evergage.cashDom("#evg-entry-intent-popup-email-capture").remove();

            Evergage.sendEvent({

                    action: "Entry Popup dismissed",

                });

        });

    }

/\*apply function

  set a timeout of 3 seconds to display popup\*/

function apply(context, template) {

                return new Promise((res, rej) => {

                timer = window.setTimeout(() => {

                init(context, template);

                res(true);

            }, 3000);

        });

    }

    /\*function init\*/

     function init(context, template) {

        window.clearTimeout(timer);

        const html = template(context);

        Evergage.cashDom('body').append(html);

        document.getElementById("lastTab").addEventListener("click",showHideTip);

        document.getElementById("ttClose").addEventListener("click", showHideTip);

        setConfirmationPanel();

        setDismissal(context);

    }

    /\*reset fn\*/

    function reset(context, template) {

        Evergage.DisplayUtils.unbind(buildBindId(context));

        Evergage.cashDom("#evg-entry-intent-popup-email-capture").remove();

    }

    function control(context) {

            return context.contentZone && Evergage.DisplayUtils.bind(buildBindId(context)).pageInactive(3000).then(() => {

            return true;

    }

    registerTemplate({

        apply: apply,

        reset: reset,

        control: control

    });

})();

----------------------------------------------- Server Side Code-------------------------------------

export class StyleField {

    label: string;

    className: string;

}

export class ExitIntentPopupWithEmailCapture implements CampaignTemplateComponent {

    @header("Pop-Up Type")

    lightbox: boolean = true;

    @title("Background Image URL")

    imageUrl: string = "https://image.e.yankeecandle.com/lib/fe3e15717564047a741271/m/1/6b6d93b6-f6a1-4534-a896-1a67e02e647b.jpg";

    @subtitle("Define header and subheader text styling.")

    @options([

        { label: "Light on Dark", className: "evg-light-on-dark" },

        { label: "Dark on Light", className: "evg-dark-on-light" }

    ])

    style: StyleField = { label: "Light on Dark", className: "evg-light-on-dark" };

    header: string = "Before you go...";

    @subtitle("Optional text field")

    subheader: string = "UNLOCK 20% OFF YOUR ORDER.";

    @title("CTA Text")

    ctaText: string = "Submit";

    @title("CTA Text2")

    ctaText2: string = "Continue Shopping";

@title("Footer Text")

   footerText: string =

    "Minimum $25 purchase. Full price items only. Not valid on gift cards and sale items. Offer applies to first-time sign-ups only."

    //@title("Opt-Out Text")

    //@subtitle("Clicking this text closes the pop-up.")

    //optOutText: string = "No Thanks";

    @title("Confirmation Screen Header")

    @subtitle("Text appears upon successful email submission. Click CTA to preview.")

    confirmationHeader: string = "Thank You";

    @title("Confirmation Screen Subheader")

    @subtitle("Optional text field")

    confirmationSubheader: string = "Your 20% OFF offer is on its way to your inbox.";

    run(context: CampaignComponentContext) {

        return {};

    }

}

*[DEV] Exit Intent Pop-Up  with FE changes*

{{!--

    Global Template: Exit Intent Pop-Up with Email Capture

    This global template displays a popup with an email capture when a user intends to exit the page.

    There are two default Styles: Dark text for lighter backgrounds or light text on darker backgrounds.

    Requirements:

    1) Set the content zone(s) to that defined in your Sitemap.

    2) Create a user attribute `emailAddress` to save the user's email to.

    Customizations:

    1) As a best practice, set the content zone to "global\_popup".

    See https://developer.evergage.com/web-integration/sitemap/contentzones for recommended names for content zones

    mapped via the Sitemap.

    2) Style options can be changed in Serverside Code.

    3) Styling of the Style options, including fallback background color, can be changed in CSS.

    4) Remove or change the default values for the configuration properties in Serverside Code.

    5) Once validated, the user's email address will be sent to Interaction Studio and set to the user's

    emailAddress attribute.

    6) Media query breakpoints can be changed in CSS.

    7) To support rich text, add the @richtext(true) decorator to configuration properties in Serverside Code.

    Handlebars expressions for these properties must be wrapped in three sets of curly braces (e.g. {{{expression}}} ).

    8) Change the value of the aria-label attributes as necessary.

    9) Change the error message in the .evg-error-msg. Styling can be changed in CSS.

--}}

<div id="evg-exit-intent-popup-email-capture" data-evg-campaign-id="{{campaign}}"

data-evg-experience-id="{{experience}}" data-evg-user-group="{{userGroup}}"

aria-label="Popup{{#if header}} - {{header}}{{/if}}" role="dialog" aria-modal="true" aria-hidden="false">

    {{#if lightbox}}

    <div class="evg-overlay" data-evg-dismissal></div>

    {{/if}}

    <div class="evg-popup evg-style-bg {{style.className}}">

      <div class="evg-left" width="50%">

          <img class="background-img" src="{{~imageUrl~}}" width="400px" height="410px">

      </div>

        <div class="evg-main-panel">

            <h1 class="evg-header evg-h1 evg-display-4 evg-style-text {{style.className}}">

                {{~header~}}

            </h1>

            {{#if subheader}}

            <h2 class="evg-subheader evg-h2 evg-style-text {{style.className}}">

                {{~subheader~}}

            </h2>

            {{/if}}

            <form class="evg-form" aria-label="Email Signup" role="form">

                   <div class="input-group evg-email-input-group">

                       <input class="evg-mobile" name="phone" type="string" id="entryPhoneInput" placeholder="Phone(Optional)" novalidate aria-label="mobile" tabindex="0">

                       <label class="floating-label evg-hide"  for="entryPhoneInput" data-evg-floating>Enter Phone Number</label>

                <span class="evg-error-msg-mobile evg-hide">

                    <div style="width:5%;display:inline-block">

                                        <svg xmlns="http://www.w3.org/2000/svg" width="14" height="14" fill="currentColor" class="bi bi-exclamation-circle bi-ex-error" viewBox="0 0 16 16">

                                                <path d="M8 15A7 7 0 1 1 8 1a7 7 0 0 1 0 14zm0 1A8 8 0 1 0 8 0a8 8 0 0 0 0 16z"/>

                                                <path d="M7.002 11a1 1 0 1 1 2 0 1 1 0 0 1-2 0zM7.1 4.995a.905.905 0 1 1 1.8 0l-.35 3.507a.552.552 0 0 1-1.1 0L7.1 4.995z"/>

                                            </svg>

                                    </div>

                  Please Enter a valid phone number.

                </span>

            <input class="evg-email" name="email" type="email" required placeholder="Email" novalidate aria-label="Email" aria-required="true" tabindex="0">

                                    <div class="input-group-append">

                                         <button class="evg-cta evg-btn evg-btn-primary evg-btn-lg {{style.className}}" id="exitEmailBtn" onsubmit="return false;"

                tabindex="1" type="button" aria-label="Submit">

                    {{~ctaText~}}

                </button>

                                    </div>

                                    <span class="evg-error-msg-empty-email evg-hide">

                                        <div style="width:5%;display:inline-block">

                                        <svg xmlns="http://www.w3.org/2000/svg" width="14" height="14" fill="currentColor" class="bi bi-exclamation-circle bi-ex-error" viewBox="0 0 16 16">

                                                <path d="M8 15A7 7 0 1 1 8 1a7 7 0 0 1 0 14zm0 1A8 8 0 1 0 8 0a8 8 0 0 0 0 16z"/>

                                                <path d="M7.002 11a1 1 0 1 1 2 0 1 1 0 0 1-2 0zM7.1 4.995a.905.905 0 1 1 1.8 0l-.35 3.507a.552.552 0 0 1-1.1 0L7.1 4.995z"/>

                                            </svg>

                                    </div>

                  Email address is required.

                </span>

                 <span class="evg-error-msg-invalid-email evg-hide">

                     <div style="width:5%;display:inline-block">

                                        <svg xmlns="http://www.w3.org/2000/svg" width="14" height="14" fill="currentColor" class="bi bi-exclamation-circle bi-ex-error" viewBox="0 0 16 16">

                                                <path d="M8 15A7 7 0 1 1 8 1a7 7 0 0 1 0 14zm0 1A8 8 0 1 0 8 0a8 8 0 0 0 0 16z"/>

                                                <path d="M7.002 11a1 1 0 1 1 2 0 1 1 0 0 1-2 0zM7.1 4.995a.905.905 0 1 1 1.8 0l-.35 3.507a.552.552 0 0 1-1.1 0L7.1 4.995z"/>

                                            </svg>

                                    </div>

                    Please Enter a valid email address.

                </span>

                                </div>

                           <!-- Footer -->

                                <div class="evg-footer" id="evg-footer" data-evg-footer-container>

                                    {{footerText}}

                                    <p>&nbsp;</p>

                                    <div>

                                        <div id="toolTipBox" class="evg-toolTip">By signing up with my mobile phone number, I agree to receive recurring automated promotional and personalized marketing text messages from Newell Brands Home Fragrance at the cell number used when signing up. marketing

                                            text messages from Newell Brands Home Consent is not a condition of any purchase. Reply HELP for help and STOP to cancel. Msg frequency varies. Msg and data rates may apply. View

                                            <a href="https://www.newellbrands.com/terms-of-use" id="entryTtTerms" target="\_blank">Terms</a> & <a href="https://privacy.newellbrands.com/index\_en.html" id="entryTtPrivacy" target="\_blank">Privacy</a>.

                                            <br><a href="#" id="ttClose" class="evg-ttClose">Close</a>

                                        </div>

                                        <div class=?toolTipCustom?>By submitting, I confirm I have read and accept your

                                            <a href="https://privacy.newellbrands.com/index\_en.html" id="entryPrivacy" target="\_blank">Privacy Statement</a> and I would like to receive marketing and/or promotional emails from Newell Brands Home Fragrance.

                                            <a href="#" id="lastTab">

                                                <span id="ttCircle">

                                                    <svg xmlns="http://www.w3.org/2000/svg" id="ttCircleSvg" width="11" height="11" class="bi bi-exclamation-circle" viewBox="0 0 16 16">

                                                        <path d="M8 15A7 7 0 1 1 8 1a7 7 0 0 1 0 14zm0 1A8 8 0 1 0 8 0a8 8 0 0 0 0 16z"/>

                                                        <path d="M7.002 11a1 1 0 1 1 2 0 1 1 0 0 1-2 0zM7.1 4.995a.905.905 0 1 1 1.8 0l-.35 3.507a.552.552 0 0 1-1.1 0L7.1 4.995z"/>

                                                    </svg>

                                                </span>

                                                <span id="ttCircleFill" style="display:none;">

                                                <svg xmlns="http://www.w3.org/2000/svg" id="ttCircleFillSvg" width="11" height="11" class="bi bi-exclamation-circle-fill" viewBox="0 0 16 16">

                                                    <path d="M16 8A8 8 0 1 1 0 8a8 8 0 0 1 16 0zM8 4a.905.905 0 0 0-.9.995l.35 3.507a.552.552 0 0 0 1.1 0l.35-3.507A.905.905 0 0 0 8 4zm.002 6a1 1 0 1 0 0 2 1 1 0 0 0 0-2z"/>

                                                </svg>

                                                </span>

                                            </a>

                                        </div>

                                    </div>

                                </div>

            </form>

        <!--commenting out opt out text <span class="evg-opt-out-msg evg-style-text {{style.className}}" tabindex="2" data-evg-dismissal>

                {{~optOutText~}}

                -->

            </span>

        </div>

         <div class="evg-confirm-panel evg-hide">

            <h1 class="evg-confirm-header evg-h1 evg-style-text {{style.className}}">

                {{~confirmationHeader~}}

            </h1>

            {{#if confirmationSubheader}}

            <h2 class="evg-confirm-subheader evg-h2 evg-style-text {{style.className}}">

                {{~confirmationSubheader~}}

            </h2>

            {{/if}}

                          <button class="evg-cta evg-btn evg-btn-primary evg-btn-lg {{style.className}} evg-submit-btn" data-evg-dismissal

                tabindex="3" type="button" aria-label="Submit"

        aria-label="Close Popup">    {{~ctaText2~}}

                </button>

        </div>

        <button class="evg-btn-dismissal evg-style-text {{style.className}}" data-evg-dismissal

        aria-label="Close Popup" tabindex="3">&times;</button>

    </div>

</div>

/\* ==========================================================================

   Pop-up

========================================================================== \*/

/\*========================================================================

Tool Tip

==========================================================================\*/

#toolTipBox {

position: absolute;

    top: 96px !important;

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

    padding: 10px 0px;

        border: 2px solid #000000;

}

.background-image{

    width:95%;

}

#evg-exit-intent-popup-email-capture .evg-toolTip {

    display: none;

    color: #333;

    background: #fff;

    box-shadow: 0px 0px 6px #666;

    border: 1px solid #ccc;

    font-size: 12px;

    text-align: left;

    padding: 10px;

    width: 360px;

    position: absolute;

    z-index: 109;

    top: 70px;

    left: 52%;

}

.evg-ttClose {

    float: right;

    font-size: larger;

}

#evg-exit-intent-popup-email-capture .bi-exclamation-circle {

    fill: #4e616a;

}

#evg-exit-intent-popup-email-capture .bi-exclamation-circle-fill {

    fill: #4e616a;

}

#evg-exit-intent-popup-email-capture .evg-footer {

     padding: 15px;

     font-size: 10.8px;

    color: #696158;

    width: 100%;

}

#evg-exit-intent-popup-email-capture .bi-ex-error {

    fill: #bd1313;

}

#evg-exit-intent-popup-email-capture .evg-footer p {

    margin: 10px 0 0 0;

    word-wrap: break-word;

}

#evg-exit-intent-popup-email-capture .evg-footer a {

    color: #4e616a;

    text-decoration: none;

}

#evg-exit-intent-popup-email-capture .evg-footer-offer {

    border: 1px solid #aaa;

    background-color: #fff;

    color: #696158;

    width: 95%;

    font-size: 0.5em;

    margin-bottom: 0px;

    padding: 10px;

}

#evg-exit-intent-popup-email-capture .evg-footer-offer a {

    color: #000;

    text-decoration: underline;

}

#evg-exit-intent-popup-email-capture {

    position: relative;

    z-index: 10000000;

    background-color: #fff;

}

#evg-exit-intent-popup-email-capture .evg-overlay {

    position: fixed;

    top: 0;

    right: 0;

    bottom: 0;

    left: 0;

    opacity: .3;

    width: 100%;

    height: 100%;

    background-color: #fff;

}

#evg-exit-intent-popup-email-capture .evg-popup {

  position: fixed;

    top: 70%;

    left: 68.5%;

    width: 800px;

    height: 410px;

    padding: 0px;

    background-color: #fff;

    background-position: center bottom;

    background-size: cover;

    background-repeat: no-repeat;

    transform: translateX(-50%) translateY(-50%);

}

#evg-exit-intent-popup-email-capture .evg-btn-dismissal {

    position: absolute;

    top: 0;

    right: 0;

    padding: 0 .5rem;

    border: 1px solid transparent;

    background-color: transparent;

    font-size: 2rem;

    font-weight: 700;

    line-height: 1;

    cursor: pointer;

}

#evg-exit-intent-popup-email-capture .evg-main-panel,

#evg-exit-intent-popup-email-capture .evg-confirm-panel {

    display: -webkit-box;

    display: -ms-flexbox;

    display: flex;

    -webkit-box-orient: vertical;

    -webkit-box-direction: normal;

    -ms-flex-direction: column;

    flex-direction: column;

    -webkit-box-pack: center;

    -ms-flex-pack: center;

    justify-content: center;

    -webkit-box-align: center;

    -ms-flex-align: center;

    align-items: center;

    width: 50%;

     margin: -26% 0px 0% 47%;

    height: 0%;

    text-align: center;

}

#evg-exit-intent-popup-email-capture .evg-confirm-header {

    display: inline-block;

    margin: 0 0 .5rem;

}

#evg-exit-intent-popup-email-capture .evg-confirm-subheader {

    display: inline-block;

    margin: 0 0 .5rem;

}

#evg-exit-intent-popup-email-capture .evg-confirm-panel .evg-btn {

    width: 200px;

    padding: 20px 10px;

}

#evg-exit-intent-popup-email-capture .evg-error-msg-mobile  .evg-error{

    color: #ff0000;

    font-size: 1rem;

    display: block;

}

/\*#evg-exit-intent-popup-email-capture .evg-opt-out-msg {

    font-size: 1rem;

    text-decoration: underline;

    cursor: pointer;

}\*/

/\* ==========================================================================

   Form

========================================================================== \*/

#evg-exit-intent-popup-email-capture .evg-form {

    display: -webkit-box;

    display: -ms-flexbox;

    display: flex;

    -webkit-box-orient: vertical;

    -webkit-box-direction: normal;

    -ms-flex-direction: column;

    flex-direction: column;

    -webkit-box-pack: center;

    -ms-flex-pack: center;

    justify-content: center;

    -webkit-box-align: center;

    -ms-flex-align: center;

    align-items: center;

    width: 100%;

    text-align: center;

}

#evg-exit-intent-popup-email-capture .evg-email {

    width: 75%;

    margin: .5rem 0;

    padding: .5rem 1rem;

    border: 1px solid #000001;

    font-size: .8rem;

    color: #333;

    text-align: left;

    display: block;

    height: 3rem;

}

#evg-exit-intent-popup-email-capture .evg-mobile {

     margin: .2rem 0;

    border: 1px solid #000001;

    font-size: .8rem;

    color: #333;

    text-align: left;

    display: block;

    width: 100%;

    height: 3rem;

}

/\* ==========================================================================

   Headers

========================================================================== \*/

#evg-exit-intent-popup-email-capture .evg-h1,

#evg-exit-intent-popup-email-capture .evg-h2 {

    margin: 0 0 .5rem;

    padding: 0;

    color: #000001;

    font-weight: 500;

    font-family: inherit;

    line-height: 1.2;

}

#evg-exit-intent-popup-email-capture .evg-h1.evg-display-4 {

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

    font-weight: bold;

    font-size: 20px;

    color: #696158;

    padding-bottom: 10px;

}

#evg-exit-intent-popup-email-capture .evg-h2 {

       font-weight: bold;

    font-size: 16px;

    color: #696158;

    padding-bottom: 10px;

    line-height: 1.5;

}

.evg-email-input-group {

    margin-top: 5px;

    width: 90%;

    left: 20px;

}

/\* ==========================================================================

   Call-To-Action Button

==========================================================================

#evg-exit-intent-popup-email-capture .evg-btn {

    display: inline-block;

    vertical-align: middle;

    padding: .5rem 1rem;

    border: 1px solid transparent;

    border-radius: 3rem;

    color: #215ca0;

    background-color: transparent;

    font-size: 1rem;

    font-weight: 400;

    line-height: 1.5;

    text-align: center;

    text-decoration: none;

    transition:

        color .15s ease-in-out,

        background-color .15s ease-in-out,

        border-color .15s ease-in-out,

        box-shadow .15s ease-in-out;

    cursor: pointer;

    -webkit-user-select: none;

    -moz-user-select: none;

    -ms-user-select: none;

    user-select: none;

}

#evg-exit-intent-popup-email-capture .evg-btn-primary {

    border-color: #215ca0;

    color: #fff;

    background-color: #215ca0;

    transition: background-color .15s, color .15s;\*/

}

#evg-exit-intent-popup-email-capture .evg-btn-lg {

    padding: .5rem 2rem;

    font-size: 1.25rem;

}

#evg-exit-intent-popup-email-capture .evg-cta {

    margin: .5rem 0;

}

/\* ==========================================================================

   Error Message

========================================================================== \*/

#evg-exit-intent-popup-email-capture  .evg-error-msg-invalid-email {

    color: #ff0000;

    font-size: 1rem;

}

#evg-exit-intent-popup-email-capture  .evg-error-msg-empty-email {

    color: #ff0000;

    font-size: 1rem;

}

#evg-exit-intent-popup-email-capture  .evg-error-msg-mobile {

    color: #ff0000;

    font-size: 1rem;

}

#evg-modal-email-capture .floating-label {

    color: #ff0000;

}

/\* ==========================================================================

   Style

========================================================================== \*/

#evg-exit-intent-popup-email-capture .evg-style-bg.evg-light-on-dark {

    background-color: #ffffff;

}

#evg-exit-intent-popup-email-capture

{

    background-color: #ffffff;

}

#evg-exit-intent-popup-email-capture .evg-style-bg.evg-dark-on-light {

    background-color: #fff;

}

#evg-exit-intent-popup-email-capture .evg-style-text.evg-light-on-dark {

    color: #000001;

}

#evg-exit-intent-popup-email-capture .evg-style-text.evg-dark-on-light {

    color: #343a40;

}

/\* ==========================================================================

   Visibility

========================================================================== \*/

#evg-exit-intent-popup-email-capture .evg-hide {

    visibility: hidden;

    display: none;

    width: 0;

    height: 0;

}

/\* ==========================================================================

   Media Queries

========================================================================== \*/

@media screen and (max-height: 500px) {

    #evg-exit-intent-popup-email-capture .evg-popup {

        height: 90vh;

    }

}

@media screen and (max-width: 575px) {

    #evg-exit-intent-popup-email-capture .evg-popup {

        width: 90vw;

    }

    #evg-exit-intent-popup-email-capture .evg-h1.evg-display-4 {

        font-size: 2.75rem;

    }

    #evg-exit-intent-popup-email-capture .evg-h2 {

        font-size: 1.2rem;

    }

    #evg-exit-intent-popup-email-capture .evg-btn-lg {

        font-size: 1rem;

    }

    #evg-exit-intent-popup-email-capture .evg-error-msg,

   /\* #evg-exit-intent-popup-email-capture .evg-opt-out-msg {

        font-size: .85rem;

    }\*/

}

#evg-exit-intent-popup-email-capture .evg-flex-container {

    display: flex;

}

#evg-exit-intent-popup-email-capture .evg-flex-child {

    flex: 1;

    width: 800px;

    height: auto;

}

#evg-exit-intent-popup-email-capture .flex-child:first-child {

    margin-right: 20px;

}

#evg-exit-intent-popup-email-capture .evg-left {

    background-image: url(https://image.e.yankeecandle.com/lib/fe3e157…/m/1/6b6d93b6-f6a1-4534-a896-1a67e02e647b.jpg);

    background-repeat: no-repeat;

    background-size: cover;

    background-position: left center;

    width: 400px;

    height: 400px;

}

#evg-exit-intent-popup-email-capture .evg-right {

    width: 400px;

    position: relative;

    background: white;

}

#evg-exit-intent-popup-email-capture .evg-confirm-panel .evg-right {

    display: flex;

    flex-direction: column;

    align-items: left;

}

#evg-exit-intent-popup-email-capture .evg-btn {

    vertical-align: middle;

    width: 100px;

    text-align: center;

    text-decoration: none;

    transition: color .15s ease-in-out, background-color .15s ease-in-out, border-color .15s ease-in-out, box-shadow .15s ease-in-out;

    cursor: pointer;

    -webkit-user-select: none;

    -moz-user-select: none;

    -ms-user-select: none;

    user-select: none;

}

#evg-exit-intent-popup-email-capture .evg-btn-primary {

    background-color: #4e616a;

    color: #fff;

    border: 1px solid #4e616a;

    border-color: #4e616a;

}

#evg-exit-intent-popup-email-capture .evg-btn-primary:focus {

    z-index: 4;

    outline: 1px dashed #4e616a;

    outline-offset: 4px;

}

(function() {

    /\*declare global variables to be used \*/

    const pageExitMillis = 500;

    const apiUrl = `https://dev.apigw.newellbrands.com/v1/sfmc\_registeremail`; /\*dev apigee url\*/

    const apiKey = 'vUA3aMbj7fUXOGwVhpVisVZrl5FkHdrE';

    const callingApplication = 'Yankee Candle';

    const optinVerbiage = 'test dev\_yankee';

    const listName = 'global';

    const requestIp = '10.10.10.1000';

    // setup date params -functions used to get date/time values

    const callingDate = buildDateString();

    const dateTime = buildDateString(true, true);

    const optinDate = buildDateString(true, true);

    const requestID = buildUID(8);

    const source = 'popup-exit';

    const pageName = 'home';

    const siteUrl = Evergage.getCurrentPage().source.url

    const region = 'US';

    const locale = 'en\_US';

    const subBrand = dataLayer[0].pageinfo.hierarchy\_parent;

    /\*\*

     \* @function buildBindId

     \* @param {Object} context

     \* @description Create unique bind ID based on the campaign and experience IDs.

     \*/

    function buildBindId(context) {

        return `${context.campaign}:${context.experience}`;

    }

    /\*\*

     \*@function custom ToolTip

     \*@param {Object} state

     \*@description Tootltip to display requisite text

     \*/

    function circleSwap(state) {

        if (state == 'fill') {

            $("#ttCircle").css("display", "none");

            $("#ttCircleFill").css("display", "inline");

        } else {

            $("#ttCircle").css("display", "inline");

            $("#ttCircleFill").css("display", "none");

        }

    }

    // Should exclamation point be outline or solid

    function circleSwap(state) {

        if (state == 'fill') {

            $("#ttCircle").css("display", "none");

            $("#ttCircleFill").css("display", "inline");

        } else {

            $("#ttCircle").css("display", "inline");

            $("#ttCircleFill").css("display", "none");

        }

    }

    // Show/Hide Tooltip

    function showHideTip() {

        var toolTipBox = document.getElementById("toolTipBox");

        let ttCircleY = document.getElementById("ttCircle").offsetTop;

        toolTipBox.style.position = "absolute";

        toolTipBox.style.top = (ttCircleY - 184) + "px";

        if (toolTipBox.style.display != "block") {

            circleSwap('fill');

            $("#toolTipBox").fadeIn();

        } else {

            $("#toolTipBox").hide();

            circleSwap('outline');

        }

    }

    /\*\*

     \* @function setConfirmationPanel

     \* @description Add click listener to the Call-To-Action button that validates the user email address,

     \* shows the Confirmation Panel, removes dismissal tracking from the "X" button and overlay, and sends

     \* an event to Interaction Studio to set the emailAddress attribute to the user email address.

     \*/

    function setConfirmationPanel() {

        Evergage.cashDom("#evg-exit-intent-popup-email-capture .evg-cta").on("click", () => {

            const email = Evergage.cashDom(".evg-form input[placeholder='Email']").val();

            let mobile = Evergage.cashDom(".evg-form input[placeholder='Phone(Optional)']").val();

            console.log('mobile is' + mobile);

            const emailRegex = RegExp(/^[.-\w]+@[\w\-]{3,}((.)\w{2,})+$/);

            const phoneRegex = RegExp(/^[\+]?[(]?[0-9]{3}[)]?[-\s\.]?[0-9]{3}[-\s\.]?[0-9]{4,6}$/);

            let finalValidator = "";

            finalValidator = email ? emailRegex.test(email) ? "validEmail" : "invalidEmail" : "emptyEmail";

            finalValidator += "\_&\_" + (mobile ? phoneRegex.test(mobile) ? "validMobile" : "invalidMobile" : "emptyMobile");

            console.log('email, phone  is' + finalValidator);

            // analytics attempt

            dataLayer.push({

                'event': 'email-signup-attempt',

                'form\_field\_source': 'Lightbox'

            });

            //call function setupApigeeApiCall when email, mobile are valid and email valid, mobile empty.

            //In other scenarios, fn handleerror is called

            switch (finalValidator) {

                case "validEmail\_&\_validMobile":

                    emailValid = true;

                    mobileValid = true;

                    handleValid("email");

                    mobile = formatPhoneNumber(mobile);

                    console.log('formatted mobile is' + mobile);

                    setupApigeeApiCall(email, mobile);

                    break;

                case "validEmail\_&\_emptyMobile":

                    emailValid = true;

                    mobileValid = true;

                    handleValid("email");

                    setupApigeeApiCall(email, '');

                    break;

                case "emptyEmail\_&\_validMobile":

                    handleError("emptyEmail");

                    emailValid = false;

                    mobileValid = true;

                    break;

                case "validEmail\_&\_invalidMobile":

                    emailValid = true;

                    mobileValid = false;

                    handleError("invalid\_mobile");

                    break;

                case "invalidEmail\_&\_invalidMobile":

                    handleError("invalid\_email\_&\_invalidMobile");

                    emailValid = false;

                    mobileValid = false;

                    break;

                case "invalidEmail\_&\_validMobile":

                    handleError("invalid\_email");

                    emailValid = false;

                    mobileValid = true;

                    break;

                case "invalidEmail\_&\_emptyMobile":

                    handleError("invalid\_email");

                    emailValid = false;

                    mobileValid = true;

                    break;

                case "emptyEmail\_&\_invalidMobile":

                    handleError("emptyEmail\_&\_invalidMobile");

                    emailValid = false;

                    mobileValid = false;

                    break;

                case "emptyEmail\_&\_emptyMobile":

                    handleError("emptyEmail");

                    emailValid = false;

                    mobileValid = true;

                    break;

                default:

                    null;

            }

        });

    }

    /\*\*

     \* @function setupapigeecall

     \* @param {Object} context

     \* @description Create request ,method and body in JSOn format to then pass to function callapigee

     \*\*/

    function setupApigeeApiCall(email, mobile) {

        console.log("function entered-->setupApigeeApiCall" + email);

        let headers = new Headers();

        headers.append('Content-Type', 'application/json');

        const body = JSON.stringify({

            'Header': {

                'APIKey': apiKey,

                'DateTime': dateTime,

                'RequestID': requestID,

                'Base': {

                    'CallingApplication': callingApplication,

                    'CallingDate': callingDate,

                    'SiteUrl': siteUrl,

                    'Source': source,

                    'RequestingIP': requestIp,

                    'ListName': listName,

                    'OptInVerbiage': optinVerbiage,

                    'PageName': pageName,

                    'OptInDate': optinDate,

                    'Region': region,

                    'sub\_brand': subBrand

                }

            },

            'WebFormDetails': {

                'Email': email,

                'Country': region

            },

            'SMSSignUp': {

                'mobilenumber': mobile,

                'subscribe': true,

                'resubscribe': true

            }

        });

        let requestOptions = {

            method: 'POST',

            headers,

            body,

            redirect: 'follow'

        };

        callApigeeApi(email, apiUrl, requestOptions, mobile);

    }

    /\*\*

     \* @function callApigee

     \* @param {Object} context

     \* @description call apigee and pass email and mobile(if avalable).

     \* Get the contactid from call and pass to IS as well as generate PBD cookie

     \*\*/

    function callApigeeApi(email, apiUrl, requestOptions, mobile) {

        console.log("function entered-->ApigeeApiCall");

        fetch(apiUrl, requestOptions)

            .then(response => {

                if (response.status !== 200) {

                    throw new Error(response.status);

                }

                return response.text();

            })

            .then(result => {

                console.log(`%c${result} `, 'color:green');

                // analytics email success

                dataLayer.push({

                    'event': 'email-signup-success',

                    'form\_field\_source': 'Lightbox'

                });

                result = JSON.parse(result);

                if (mobile) {

                    if (!result.sms\_response && !result.sms\_response.results) {

                        if (result.sms\_response.results.length <= 0) {

                            // handleError(`There was an error adding your mobile number. Please check the format`, 'default');

                            return;

                        }

                    } else {

                        dataLayer.push({

                            'event': 'sms-signup-success',

                            'form\_field\_source': 'Lightbox'

                        });

                    }

                }

                generatePBD(result.contactResponse['sfdc.contact\_id'], email);

                updateIS(email, mobile, result.contactResponse['sfdc.contact\_id']);

            })

            .catch(error => {

                console.log(`%cError: ${error}`, 'color:red');

                // analytics fail

                dataLayer.push({

                    'event': 'email-signup-fail',

                    'form\_field\_source': 'Lightbox',

                    'error\_message': error

                });

                handleError(`There was an error connecting to marketing cloud. Error: ${error}`);

            });

    }

    function generatePBD(contactId, email) {

        // Api Constants

        const auth = 'Basic c3RvcmVmcm9udDpOV0xlY29tbWVyY2U=';

        const authCookie = '\_\_cfduid=d6640e7a78a11050e5cc80a289093b3c51616691444; \_\_cfduid=dc4c4e5c7e48590ff88271971c7cb5a4c1616691444; \_\_cq\_dnt=1; dw\_dnt=1; dwanonymous\_e2b672b758c4413673702ba1fd48d98d=abitqlXr9CBcDztxeN1Vlw679W; dwsid=Mr0NoztkI5K29f3q73eLTka1w8OP1vXS9Ff3Tl6C8PceiL7eh-NdOaPVI0WDH1YGdWhvvSe\_bmq-A3URJukDNg==; sid=JQ5vcUINh3QZt3BlH3fr4lEGoRH2v1E0WxU';

        const PBDUrl = `https://us.dev.yankeecandle.com/on/demandware.store/Sites-homefragranceus-Site/en\_US/ContactUs-UpdateContactID`;

        // Headers

        let headers = new Headers();

        headers.append('Authorization', auth);

        headers.append('Cookie', authCookie);

        // Data

        let formdata = new FormData();

        formdata.append('contactId', contactId);

        formdata.append('email', email);

        // Options

        let requestOptions = {

            method: 'POST',

            headers: headers,

            body: formdata,

            redirect: 'follow'

        };

        fetch(PBDUrl, requestOptions)

            .then(response => {

                if (response.status !== 200) {

                    throw new Error(response.status);

                }

                return response.text();

            })

            .then(result => {

                console.log("PBD is success");

                result = JSON.parse(result);

                console.log("check json" + result.payload);

                generateCookie(result.payload, 'custPBD', 360);

            })

            .catch(error => {

                handleError('There was an error generating PB Code');

                //console.log('error', error)

            });

    }

    /\*\*

     \* @function generateCookie

     \* @description write pbd cookie

     \*/

    function generateCookie(payload, cname, xdays) {

        let d = new Date();

        d.setTime(d.getTime() + (xdays \* 24 \* 60 \* 60 \* 1000));

        const expires = 'expires=' + d.toUTCString();

        console.log("cookie expires on " + expires);

        if (getCookie(cname) === null) {

            document.cookie = cname + '=' + payload + ';' + expires + ';path=/';

            console.log('cookie created');

        } else {

            console.error('cookie exists');

        }

    }

    /\*\*

     \* @function getCookie

     \* @description find cookie by name

     \*/

    function getCookie(name) {

        var match = document.cookie.match(RegExp('(?:^|;\\s\*)' + name + '=([^;]\*)'));

        return match ? match[1] : null;

    }

    /\*\*

     \* @function buildDateString

     \* @param {Boolean} addTime

     \* @description Use current day to setup a date variable

     \*/

    function buildDateString(addTime, label) {

        let d = new Date();

        let dd = String(d.getDate()).padStart(2, '0');

        let mm = String(d.getMonth() + 1).padStart(2, '0');

        let yyyy = d.getFullYear();

        let time = d.toLocaleTimeString();

        let timeLabel = (label) ? 'T' : ' ';

        time = time.replace('AM', '').replace('PM', '');

        return (addTime) ? `${mm}/${dd}/${yyyy}${timeLabel}${time}` : `${mm}/${dd}/${yyyy}`;

    }

    /\*\*

     \* @function buildUID

     \* @description create a random string uid

     \* @param {Number} length

     \*/

    function buildUID(length) {

        const chars = '0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ';

        let result = '';

        for (let i = length; i > 0; --i) result += chars[Math.floor(Math.random() \* chars.length)];

        return result;

    }

    /\*\*

     \* @function formatPhoneNumber

     \* @param {String} phoneNumberString

     \* @description normalizes phone number

     \*/

    function formatPhoneNumber(phoneNumberString) {

        let cleaned = ('' + phoneNumberString).replace(/\D/g, '');

        let match = cleaned.match(/^(1|)?(\d{3})(\d{3})(\d{4})$/);

        if (match) {

            let intlCode = (match[1]) ? match[1] : '1';

            return [intlCode, match[2], match[3], match[4]].join('');

        }

        return null;

    }

    /\*\*

     \* @function handleError

     \* @param {String} errorText

     \* @param {String} field

     \* @description Display relevant error messages

     \*/

    function handleError(field) {

        switch (field) {

            case "invalid\_mobile":

                Evergage.cashDom("#evg-exit-intent-popup-email-capture .evg-error-msg-mobile").removeClass("evg-hide");

                Evergage.cashDom("#evg-exit-intent-popup-email-capture .evg-error-msg-mobile").addClass("evg-error");

                Evergage.cashDom("evg-exit-intent-popup-email-capture .floating-label").removeClass("evg-hide");

                Evergage.cashDom("#evg-exit-intent-popup-email-capture .evg-confirm-panel").addClass("evg-hide");

                Evergage.cashDom("#evg-exit-intent-popup-email-capture .evg-error-msg-empty-email").addClass("evg-hide");

                Evergage.cashDom("#evg-exit-intent-popup-email-capture .evg-error-msg-invalid-email").addClass("evg-hide");

                break;

            case "invalid\_email":

                Evergage.cashDom(".evg-error-msg-invalid-email").removeClass("evg-hide")

                Evergage.cashDom(".evg-error-msg-invalid-email").addClass("evg-error");

                Evergage.cashDom("#evg-exit-intent-popup-email-capture .evg-error-msg-mobile").addClass("evg-hide");

                Evergage.cashDom("evg-exit-intent-popup-email-capture .floating-label").addClass("evg-hide");

                Evergage.cashDom("#evg-exit-intent-popup-email-capture .evg-error-msg-empty-email").addClass("evg-hide");

                Evergage.cashDom("#evg-exit-intent-popup-email-capture .evg-confirm-panel").addClass("evg-hide");

                break;

            case "invalid\_email\_&\_invalidMobile":

                Evergage.cashDom("#evg-exit-intent-popup-email-capture .evg-error-msg-mobile").removeClass("evg-hide")

                Evergage.cashDom("#evg-modal-email-capture .floating-label").removeClass("evg-hide");;

                Evergage.cashDom("#evg-exit-intent-popup-email-capture .evg-error-msg-invalid-email").removeClass("evg-hide");

                Evergage.cashDom("#evg-exit-intent-popup-email-capture .evg-error-msg-invalid-email").addClass("evg-error");

                Evergage.cashDom("#evg-exit-intent-popup-email-capture .evg-error-msg-mobile").addClass("evg-error");

                Evergage.cashDom("#evg-exit-intent-popup-email-capture .evg-confirm-panel").addClass("evg-hide");

                Evergage.cashDom("#evg-exit-intent-popup-email-capture .evg-error-msg-empty-email").addClass("evg-hide");

                break;

            case "emptyEmail\_&\_invalidMobile":

                Evergage.cashDom("#evg-exit-intent-popup-email-capture .evg-error-msg-empty-email").removeClass("evg-hide");

                Evergage.cashDom("#evg-exit-intent-popup-email-capture .evg-error-msg-empty-email").addClass("evg-error");

                Evergage.cashDom("#evg-exit-intent-popup-email-capture .evg-error-msg-mobile").removeClass("evg-hide");

                Evergage.cashDom("#evg-exit-intent-popup-email-capture .evg-error-msg-mobile").addClass("evg-error");

                Evergage.cashDom("#evg-modal-email-capture .floating-label").removeClass("evg-hide");

                Evergage.cashDom("#evg-exit-intent-popup-email-capture .evg-confirm-panel").addClass("evg-hide");

                Evergage.cashDom("#evg-exit-intent-popup-email-capture .evg-error-msg-invalid-email").addClass("evg-hide");

                break;

            case "emptyEmail":

                Evergage.cashDom("#evg-exit-intent-popup-email-capture .evg-error-msg-empty-email").removeClass("evg-hide");

                Evergage.cashDom("#evg-exit-intent-popup-email-capture .evg-error-msg-empty-email").addClass("evg-error");

                Evergage.cashDom("#evg-exit-intent-popup-email-capture .evg-confirm-panel").addClass("evg-hide");

                Evergage.cashDom("#evg-exit-intent-popup-email-capture .evg-error-msg-invalid-email").addClass("evg-hide");

                Evergage.cashDom("#evg-exit-intent-popup-email-capture .evg-error-msg-mobile").addClass("evg-hide");

                Evergage.cashDom("#evg-modal-email-capture .floating-label").addClass("evg-hide");

                break;

            case "default":

                Evergage.cashDom("#evg-exit-intent-popup-email-capture .evg-error-msg-invalid-email")

                    .removeClass("evg-hide")

                    .addClass("evg-error");

                Evergage.cashDom("#evg-exit-intent-popup-email-capture .evg-confirm-panel").addClass("evg-hide");

                break;

            default:

                null;

        }

    }

    /\*\*

     \* @function handleValid

     \* @param {String} field

     \* @description Hide error messages, show confirm panel

     \*/

    function handleValid(field) {

        switch (field) {

            case "mobile":

                console.log("handle valid called mobile??");

                Evergage.cashDom("#evg-exit-intent-popup-email-capture .evg-main-panel").addClass("evg-hide");

                Evergage.cashDom("#evg-exit-intent-popup-email-capture .evg-confirm-panel").removeClass("evg-hide");

                Evergage.cashDom(`

                    #evg-exit-intent-popup-email-capture .evg-btn-dismissal

                `).removeAttr("data-evg-dismissal");

                break;

            case "email":

                console.log("handle valid called??");

                Evergage.cashDom("#evg-exit-intent-popup-email-capture .evg-main-panel").addClass("evg-hide");

                Evergage.cashDom("#evg-exit-intent-popup-email-capture .evg-confirm-panel").removeClass("evg-hide");

                Evergage.cashDom(`

                    #evg-exit-intent-popup-email-capture .evg-btn-dismissal

                `).removeAttr("data-evg-dismissal");

                break;

            default:

                null;

        }

    }

    /\*\*

     \* @function updateIS

     \* @description update IS event to email sign up- popup and capture email, phone and contactid

     \*/

    function updateIS(email, mobile, contactId) {

        Evergage.sendEvent({

            action: "Email Sign Up - Popup",

            user: {

                attributes: {

                    emailAddress: email,

                    email: email,

                    sfmcContactKey: contactId,

                    phone: mobile

                }

            }

        });

    }

    /\*\*

     \* @function setDismissal

     \* @param {Object} context

     \* @description Add click listener to the overlay, "X" button, and opt-out text that removes the

     \* template from the DOM.

     \* #evg-exit-intent-popup-email-capture .evg-opt-out-msg(dissnissselector code removed as opt out text removed)

     \*/

    function setDismissal(context) {

        const dismissSelectors = `

           #evg-exit-intent-popup-email-capture  .evg-submit-btn,

            #evg-exit-intent-popup-email-capture .evg-btn-dismissal,

            #evg-exit-intent-popup-email-capture .evg-opt-out-msg

            `;

        Evergage.cashDom(dismissSelectors).on("click", () => {

            Evergage.cashDom("#evg-exit-intent-popup-email-capture").remove();

            Evergage.sendEvent({

                action: "Exit Popup dismissed",

            });

        });

    }

    function apply(context, template) {

        if (!context.contentZone) return;

        /\*\*

         \* The pageExit method waits for the user's cursor to exit through the top edge of the page before rendering the

         \* template after a set delay.

         \*

         \* Visit the Template Display Utilities documentation to learn more:

         \* https://developer.evergage.com/campaign-development/web-templates/web-display-utilities

         \*/

        return Evergage.DisplayUtils

            .bind(buildBindId(context))

            .pageExit(pageExitMillis)

            .then(() => {

                const html = template(context);

               if (Evergage.cashDom("#evg-modal-email-capture").length > 0)

                {

                        Evergage.cashDom('#evg-modal-email-capture').remove();

                }

                Evergage.cashDom("body").append(html);

                document.getElementById("lastTab").addEventListener("click", showHideTip);

                document.getElementById("ttClose").addEventListener("click", showHideTip);

                setConfirmationPanel();

                setDismissal(context);

                /\* document.getElementById("#evg-exit-intent-popup-email-capture .evg-ttClose").addEventListener("click",addClass("evg-hide"));\*/

            });

    }

    function reset(context, template) {

        Evergage.DisplayUtils.unbind(buildBindId(context));

        Evergage.cashDom("#evg-exit-intent-popup-email-capture").remove();

    }

    function control(context) {

        return context.contentZone &&

            Evergage.DisplayUtils

            .bind(buildBindId(context))

            .pageExit(pageExitMillis)

            .then(() => {

                if (context.contentZone) return true;

            });

    }

    registerTemplate({

        apply: apply,

        reset: reset,

        control: control

    });

})();

export class StyleField {

    label: string;

    className: string;

}

export class ExitIntentPopupWithEmailCapture implements CampaignTemplateComponent {

    @header("Pop-Up Type")

    lightbox: boolean = true;

    @title("Background Image URL")

    imageUrl: string = "https://image.e.yankeecandle.com/lib/fe3e15717564047a741271/m/1/6b6d93b6-f6a1-4534-a896-1a67e02e647b.jpg";

    @subtitle("Define header and subheader text styling.")

    @options([

        { label: "Light on Dark", className: "evg-light-on-dark" },

        { label: "Dark on Light", className: "evg-dark-on-light" }

    ])

    style: StyleField = { label: "Light on Dark", className: "evg-light-on-dark" };

    header: string = "Before you go...";

    @subtitle("Optional text field")

    subheader: string = "UNLOCK 20% OFF YOUR ORDER.";

    @title("CTA Text")

    ctaText: string = "Submit";

    @title("CTA Text2")

    ctaText2: string = "Continue Shopping";

@title("Footer Text")

   footerText: string =

    "Minimum $25 purchase. Full price items only. Not valid on gift cards and sale items. Offer applies to first-time sign-ups only."

    //@title("Opt-Out Text")

    //@subtitle("Clicking this text closes the pop-up.")

    //optOutText: string = "No Thanks";

    @title("Confirmation Screen Header")

    @subtitle("Text appears upon successful email submission. Click CTA to preview.")

    confirmationHeader: string = "Thank You";

    @title("Confirmation Screen Subheader")

    @subtitle("Optional text field")

    confirmationSubheader: string = "Your 20% OFF offer is on its way to your inbox.";

    run(context: CampaignComponentContext) {

        return {};

    }

}

userAttributes: string ="emailAddress,lastName";

    run(context: CampaignComponentContext) {

        let attrs = this.userAttributes.split(",")

        let attributesToSend = {}

        attrs.filter((value) => {

            return context.user.attributes.hasOwnProperty(value);

        }).map((value2) => {

            attributesToSend[value2] = {

                value: context.user.attributes[value2]

            }

        });

        return {

            attributes: attributesToSend

        };

{{attributes.lastName.value.value}}

Sitemap

Evergage.configure({

//Any global configurations if required

});

Evergage.init({

cookieDomain: "yankeecandle.com",

trackerUrl: "https://newell.evergage.com"

}).then(() => {

/\*Create site-map here.\*/

//Global Variable for Products from dataLayer

let dataLayerProduct;

function checkDataLayer(){

if(window.isDataLayer){

var products;

if( window.isDataLayer.pdp &&

window.isDataLayer.pdp.ecommerce &&

window.isDataLayer.pdp.ecommerce.detail &&

window.isDataLayer.pdp.ecommerce.detail.products)

{

products = window.isDataLayer.pdp.ecommerce.detail.products; //returning products from PDP

return products;

}

else if(window.isDataLayer.transaction &&

window.isDataLayer.transaction.products)

{

products = window.isDataLayer.transaction.products;

return products;

}

}

}

// Function for checking Facebook Login on checkout

function checkFbLogin(){

if(window.dataLayer){

for (let i = 0; i < window.dataLayer.length; i++) {

if ((window.dataLayer[i].event && window.dataLayer[i].event === 'facebook-login-success' && window.dataLayer[i].source && window.dataLayer[i].source === 'checkout')){

return true;

}

}

}

}

// Function for checking Guest checkout

function checkGuestCheckout(){

if(window.dataLayer){

for (let i = 0; i < window.dataLayer.length; i++) {

if ((window.dataLayer[i].event && window.dataLayer[i].event === 'checkout-steps' && window.dataLayer[i].step\_name && window.dataLayer[i].step\_name === 'Guest')){

return true;

}

}

}

}

// Function for checking dataLayer products, to be used at PDP & Order Confirmation Page.

function fetchDatalayerProduct() {

var sku = window.location.href.split("/").pop().split(".html").shift() || Evergage.cashDom(".product-detail").attr("data-pid");

let recentProduct;

const products = checkDataLayer();

if(sku == products[0].product\_sku){

recentProduct = products;

}

else{

recentProduct = getProductsFromOldDataLayer();

}

return recentProduct;

}

// condition for checking dataLayer on order Confirmation Page

if(dataLayer[0].pageinfo.page\_type === "Order Confirmation" || /\/Order\-Confirm/.test(window.location.href)){

dataLayerProduct = checkDataLayer();

}

if(dataLayer[0].pageinfo.page\_type === "Cart"){

dataLayerProduct = checkDataLayerAtCart();

}

// END

const config = {

// pagetypes, catalog object, global listners, contentzones

/\* Global Configurations of content zones and listeners \*/

global: {

//Global Content Zones below.

contentZones: [

{ name: "global\_popup"},

{ name: "global\_exit\_intent"}

],

//Global page level listeners below.

listeners: [

// Email Sign Up Popup - Listener

Evergage.listener("click", "[data-evg-submit]", () => {

var userEmail = Evergage.cashDom("[data-evg-email]").val();

if (typeof userEmail === 'string' && /^.+@.+\..+$/.test(userEmail)) {

Evergage.sendEvent({

action: "Email Sign Up - Popup",

user: {

attributes: {

email: userEmail,

emailAddress: userEmail

}

}

});

}

}),

// Email Sign Up Footer - Listener

Evergage.listener("click", ".btn.btn-footer-email-subscribe.js-emailsubscribe-component.h-100", () => {

var userEmail = Evergage.cashDom(".input-group.input-with-button.form-label input.form-control").val();

if (typeof userEmail === 'string' && /^.+@.+\..+$/.test(userEmail)) {

Evergage.sendEvent({

action: "Email Sign Up - Footer",

user: {

attributes: {

email: userEmail,

emailAddress: userEmail

}

}

});

}

}),

// Email Sign Up Popup - Listener

Evergage.listener("click", ".js-lightbox-submit.btn.btn-primary" , () => {

var userEmail = Evergage.cashDom(".input-group.input-with-button.col-12.col-md-10.p-0 input.js-lightbox-email.form-control").val();

if (typeof userEmail === 'string' && userEmail) {

Evergage.sendEvent({

action: "Email Sign Up - Popup",

user: {

attributes: {

email: userEmail,

emailAddress: userEmail

}

}

});

}

}),

// Search Term for Product Suggestion

Evergage.listener("click", ".col-12.item.li-padding.suggested-item-section" , () => {

var searchTerm = Evergage.cashDom('input.form-control.search-field').val();

if (searchTerm) {

Evergage.sendEvent({

action: "Site Search Click",

user: {

attributes: {

lastSearchTerm: searchTerm

}

}

});

}

}),

//Search Term for Typed Search

Evergage.listener("click", ".btn.search-submit-button.input-button" , () => {

var searchTerm = Evergage.cashDom('input.form-control.search-field').val();

if (searchTerm) {

Evergage.sendEvent({

action: "Site Search Click",

user: {

attributes: {

lastSearchTerm: searchTerm

}

}

});

}

})

]

},

// default pageType Name - supressing Null values.

pageTypeDefault: {

name: "default"

},

// PageTypes block to be defined below

pageTypes: [

// Home Page Config

{

name: "home\_page",

action: "Homepage View",

isMatch: () => {return dataLayer[0].pageinfo.page\_type === "Yankee Candle Homepage" ||

/\/Home-Show/.test(window.location.href)},

contentZones: [{

// name: "home\_hero", selector: ".banner-container[role=banner]:nth-of-type(1)"

// name: "home\_hero", selector: ".banner-container.row.d-flex.justify-content-center.position-relative.dark-theme.mediumHero-banner"

name: "home\_hero", selector: ".banner-container[data-pd-element-selector=banner]"

}]

},

{

name:"personalized\_candles\_landing\_page",

action: "Personalized Candles Landing Page View",

isMatch: () => {return dataLayer[0].pageinfo.hierarchy\_sub\_category1 === "Candles" && dataLayer[0].pageinfo.hierarchy\_sub\_category2 === "Personalized Candles LP"}

},

//PDP View

{

name: "product\_detail\_page",

action: "Product Detail View",

isMatch: () => {return dataLayer[0].pageinfo.page\_type === "PDP";},

catalog: {

Product: {

\_id: () => {

const products = DatalayerProduct; // compute the product based on Variant switch SPA

var sku;

if (products !== undefined) {

sku = products[0].product\_sku;

return sku;

}

else if (sku === null || sku === undefined || sku === "") {

sku = window.location.href.split("/").pop().split(".html").shift() || Evergage.cashDom(".product-detail").attr("data-pid");

return sku;

}

},

priceDescription: () => {

const products = fetchDatalayerProduct(); // compute the product based on Variant switch SPA

if (products.length > 0) {

return products[0].dimension54;

}

},

categories: () => {

var cat1 = dataLayer[0].pageinfo.hierarchy\_sub\_category1.length?dataLayer[0].pageinfo.hierarchy\_sub\_category1:"";

cat1 = cat1.replace(/&/g, "&amp;").replace(/&amp;/g,"and");

var cat2 = dataLayer[0].pageinfo.hierarchy\_sub\_category2.length?"|"+dataLayer[0].pageinfo.hierarchy\_sub\_category2:"";

cat2 = cat2.replace(/&/g, "&amp;").replace(/&amp;/g,"and");

var cat3 = dataLayer[0].pageinfo.hierarchy\_sub\_category3.length?"|"+dataLayer[0].pageinfo.hierarchy\_sub\_category3:"";

cat3 = cat3.replace(/&/g, "&amp;").replace(/&amp;/g,"and");

var cat4 = dataLayer[0].pageinfo.hierarchy\_sub\_category4.length?"|"+dataLayer[0].pageinfo.hierarchy\_sub\_category4:"";

cat4 = cat4.replace(/&/g, "&amp;").replace(/&amp;/g,"and");

return [(cat1+cat2+cat3+cat4).toUpperCase()];

},

dimensions: {

Color: () => {

var selectedColor = Evergage.cashDom("div.color-attribute[aria-checked='true']").attr("id");

if(selectedColor !== null){

return [selectedColor];}

},

SubBrand: () =>{const subbrand = dataLayer[0].pageinfo.hierarchy\_parent;

return [subbrand]},

HierarchySubCategoryOne: ()=>{var subcat1 = dataLayer[0].pageinfo.hierarchy\_sub\_category1;

subcat1 = subcat1.replace(/&/g, "&amp;").replace(/&amp;/g,"and");

return [subcat1]},

HierarchySubCategoryTwo: ()=>{var subcat2 = dataLayer[0].pageinfo.hierarchy\_sub\_category2;

subcat2 = subcat2.replace(/&/g, "&amp;").replace(/&amp;/g,"and");

if(subcat2!== null || subcat2 !== ""){return [subcat2]}

else{return []}

}

,

HierarchySubCategoryThree: ()=>{const subcat3 = dataLayer[0].pageinfo.hierarchy\_sub\_category3;

if(subcat3!== null || subcat3 !== ""){return [subcat3]}

else{return []}},

HierarchySubCategoryFour: ()=>{const subcat4 = dataLayer[0].pageinfo.hierarchy\_sub\_category4;

if(subcat4!== null || subcat4 !== ""){return [subcat4]}

else{return []}}

}

}

},

listeners:[

Evergage.listener("click", "button.add-to-cart.btn.btn-primary", () => {

var lineItem = Evergage.util.buildLineItemFromPageState("select[id\*=quantity-1");

const products = fetchDatalayerProduct(); // compute the product based on Variant switch SPA

lineItem.sku = { \_id: () => {

var sku;

if (products !== undefined) {

sku = products[0].product\_sku;

return sku;

}

else if (sku === null || sku === undefined || sku === "") {

sku = window.location.href.split("/").pop().split(".html").shift() || Evergage.cashDom(".product-detail").attr("data-pid");

return sku;

}

} };

lineItem.price = products[0].price;

lineItem.SubBrand = products[0].brand;

Evergage.sendEvent({

itemAction: Evergage.ItemAction.AddToCart,

action: "Add To Cart",

cart: {

singleLine: {

Product: lineItem

}

}

});

}),

Evergage.listener("click", ".btn.btn-tertiary.add-registry.gift-registry", () => {

Evergage.sendEvent({

action: "AddToRegistry",

catalog: {

Product:{

\_id: () => {

const products = dataLayerProduct;

var sku;

if (products !== undefined) {

sku = products[0].product\_sku;

return sku;

}

}

}

}

});

})

]

},

//Login and Registration Page Config

{

name: "login\_page",

action: "Login",

isMatch: () => /\/login/.test(window.location.href),

onActionEvent: (actionEvent) => {

if (actionEvent.action === "Login") {

Evergage.cashDom("form[name='login-form'] button").on("click", () => {

const email = Evergage.cashDom('#login-form-email').val().trim();

if (email && /^.+@.+\..+$/.test(email)) {

Evergage.sendEvent({

action: "Log In Attempt",

user: {

attributes:

{

email : email,

emailAddress: email,

contactId : dataLayer[0].pageinfo.newell\_brands\_user\_id,

sfmcContactKey : dataLayer[0].pageinfo.newell\_brands\_user\_id

}

}

});

}

});

}

return actionEvent;

},

listeners:[

Evergage.listener("click", ".login-tabs", () => {

if($(".nav-link.active").attr("id") == "login-tab"){

Evergage.cashDom("form[name='login-form'] button").on("click", () => {

const email = Evergage.cashDom('#login-form-email').val().trim();

if (email && /^.+@.+\..+$/.test(email)) {

Evergage.sendEvent({action: "Log In Attempt", user: {

attributes:

{

email : email,

emailAddress: email

}

} });

}

});

}

else if($(".nav-link.active").attr("id") == "register-tab"){

Evergage.cashDom("form[name='dwfrm\_profile'] button").on("click", () => {

const email = Evergage.cashDom('#registration-form-email').val().trim();

const fname = Evergage.cashDom('#registration-form-fname').val().trim();

const lname = Evergage.cashDom('#registration-form-lname').val().trim();

if (email && fname && lname) {

Evergage.sendEvent({

action: "Create Account Attempt",

user: {

attributes:

{

email : email,

emailAddress: email,

firstName : fname,

lastName : lname,

contactId : dataLayer[0].pageinfo.newell\_brands\_user\_id,

sfmcContactKey : dataLayer[0].pageinfo.newell\_brands\_user\_id

}

}

});

}

});

}

else{

console.log('fail login and sign up');

}

}),

Evergage.listener("click", "form[name='dwfrm\_profile'] button", () => {

const email = Evergage.cashDom('#registration-form-email').val().trim();

const fname = Evergage.cashDom('#registration-form-fname').val().trim();

const lname = Evergage.cashDom('#registration-form-lname').val().trim();

if (email && fname && lname) {

Evergage.sendEvent({

action: "Create Account Attempt",

user: {

attributes:

{

email : email,

emailAddress: email,

firstName : fname,

lastName : lname,

contactId : dataLayer[0].pageinfo.newell\_brands\_user\_id,

sfmcContactKey : dataLayer[0].pageinfo.newell\_brands\_user\_id

}

}

});

}

}),

Evergage.listener("click","form[name='login-form'] button",()=>{

const email = Evergage.cashDom('#login-form-email').val().trim();

if (email && /^.+@.+\..+$/.test(email)) {

Evergage.sendEvent({

action: "Log In Attempt",

user: {

attributes:

{

email : email,

emailAddress: email,

contactId : dataLayer[0].pageinfo.newell\_brands\_user\_id,

sfmcContactKey : dataLayer[0].pageinfo.newell\_brands\_user\_id

}

}

});

}

})

]

},

//My Account Page Config

{

name: "account\_page",

action: "Account",

isMatch: () => /\/account/.test(window.location.href) && /registration=/.test(window.location.search),

onActionEvent: (actionEvent) => {

if (actionEvent.action === "Account") {

setTimeout(function(){

const contactID = dataLayer[0].userinfo.newell\_brands\_user\_id;

if (contactID) {

Evergage.sendEvent({action: "Log In Success",

user: {

attributes: {

contactId : contactID,

sfmcContactKey : contactID

}

}

});

}

}, 1000);

}

return actionEvent;

}

},

//Contact Us Page Config

{

name: "contact\_us",

action: "Contact Us page visit",

isMatch: () => /\/contact-us/.test(window.location.href),

listeners: [

Evergage.listener("mousedown","#contact-us-form button.subscribe-contact-us", () => {

const email = Evergage.cashDom('input#contactus-email').val().trim();

// const email = Evergage.cashDom('#contactus-email').val().trim();

const fName = Evergage.cashDom('input#contactus-firstName').val().trim();

const lName = Evergage.cashDom('input#contactus-lastName').val().trim();

const phoneNo = Evergage.cashDom('input#contactus-phoneNumberRequired').val().trim() || Evergage.cashDom('input#contactus-phoneNumber').val().trim();

if (email) {

Evergage.sendEvent({

action: "Contact Us Form Submit",

user: {

attributes:

{

email : email,

emailAddress: email,

firstName : fName,

lastName : lName,

phone : phoneNo

}

}

});

}

}),

],

},

//Product Registration Page Config

{

name: "product\_registration",

action: "Product Registration",

isMatch: () => /\/product-registration/.test(window.location.href),

listeners: [

Evergage.listener("click","#product-registration-form button", () => {

var fName = Evergage.cashDom('#contactus-firstName').val().trim();

var lName = Evergage.cashDom('#contactus-lastName').val().trim();

var email = Evergage.cashDom('#product-registration-email').val().trim();

var phoneNo = Evergage.cashDom('input#product-registration-phoneNumber').val().trim();

if (email || phoneNo) {

Evergage.sendEvent({

action: "Product Registration Submit",

user: {

attributes:{

email: email,

emailAddress: email,

phone: phoneNo ,

firstName: fName,

lastName: lName,

contactId : dataLayer[0].pageinfo.newell\_brands\_user\_id,

sfmcContactKey : dataLayer[0].pageinfo.newell\_brands\_user\_id

}

}

});

}

}),

],

},

// Category Page Config

{

name: "category\_page",

action: "Category Page View",

isMatch: () => {return dataLayer[0].pageinfo.page\_type === "PLP"},

catalog: {

Category:{

\_id: () => {

const cat1 = dataLayer[0].pageinfo.hierarchy\_sub\_category1.length?dataLayer[0].pageinfo.hierarchy\_sub\_category1:"";

const cat2 = dataLayer[0].pageinfo.hierarchy\_sub\_category2.length?"|"+dataLayer[0].pageinfo.hierarchy\_sub\_category2:"";

const cat3 = dataLayer[0].pageinfo.hierarchy\_sub\_category3.length?"|"+dataLayer[0].pageinfo.hierarchy\_sub\_category3:"";

const cat4 = dataLayer[0].pageinfo.hierarchy\_sub\_category4.length?"|"+dataLayer[0].pageinfo.hierarchy\_sub\_category4:"";

console.log(cat1+cat2+cat3+cat4);

return [(cat1+cat2+cat3+cat4).toUpperCase()];

},

dimensions: {

SubBrand: () =>{const subbrand = dataLayer[0].pageinfo.hierarchy\_parent;

return [subbrand]},

HierarchySubCategoryOne: ()=>{const subcat1 = dataLayer[0].pageinfo.hierarchy\_sub\_category1;

return [subcat1]},

HierarchySubCategoryTwo: ()=>{const subcat2 = dataLayer[0].pageinfo.hierarchy\_sub\_category2;

if(subcat2!== null || subcat2 !== ""){return [subcat2]}

else{return []}

}

,

HierarchySubCategoryThree: ()=>{const subcat3 = dataLayer[0].pageinfo.hierarchy\_sub\_category3;

if(subcat3!== null || subcat3 !== ""){return [subcat3]}

else{return []}},

HierarchySubCategoryFour: ()=>{const subcat4 = dataLayer[0].pageinfo.hierarchy\_sub\_category4;

if(subcat4!== null || subcat4 !== ""){return [subcat4]}

else{return []}}

}

}

},

onActionEvent: (actionEvent) => {

Evergage.cashDom(".add-to-cart.btn").on("click", (event) => {

var id = JSON.parse($(Evergage.cashDom(event.currentTarget)).closest('.product-tile').attr('data-analytics-data'))['id'];

var price = JSON.parse($(Evergage.cashDom(event.currentTarget)).closest('.product-tile').attr('data-analytics-data'))['price'];

var quantity = $(Evergage.cashDom(event.currentTarget)).closest('.add-to-cart-wrapper').find('select option:selected').text();

console.log('id:'+id+' cost: '+price +'Quantity: '+quantity );

});

return actionEvent;

}

},

// cart page

{

name: "cart\_page",

action: "Cart Page View",

isMatch: () => {return dataLayer[0].pageinfo.page\_type === "Cart"},

itemAction: Evergage.ItemAction.ViewCart,

catalog: {

Product: {

lineItems: {

\_id: Evergage.resolvers.fromSelectorAttributeMultiple(".form-control .quantity .custom-select","data-pid"),

price: Evergage.resolvers.fromSelectorAttributeMultiple(".form-control .quantity .custom-select","data-pre-select-qty"),

quantity: Evergage.resolvers.fromSelectorAttributeMultiple(".card .product-info","data-price")

}

}

}

},

// checkout login page

{

name: "checkout\_page",

action: "Checkout",

isMatch: () => {return (dataLayer[0].pageinfo.page\_type === "Checkout" && dataLayer[0].pageinfo.page\_url\_querystring=== "stage=login") },

listeners: [

Evergage.listener("click",".submit-login.guest-submit", () => {

var gemail = Evergage.cashDom('#guest-form-email').val().trim();

if (gemail) {

Evergage.sendEvent({

action: "Guest Checkout",

user: {

attributes: {

email: gemail,

emailAddress: gemail

}

}

});

}

}),

],

},

// checkout payment page

{

name: "checkout\_page",

action: "Checkout Payment",

isMatch: () => {return (dataLayer[0].pageinfo.page\_type === "Checkout" && dataLayer[0].pageinfo.page\_url\_querystring != "stage=login") },

onActionEvent: (actionEvent) => {

if (actionEvent.action === "Checkout Payment") {

setTimeout(function(){

//if condition to check fb login success event

if(window.dataLayer){

for (let i = 0; i < window.dataLayer.length; i++) {

if ((window.dataLayer[i].event && window.dataLayer[i].event === 'facebook-login-success' && window.dataLayer[i].source && window.dataLayer[i].source === 'checkout')){

var fbLogin = true;

}

}

}

if((/stage=payment/.test(window.location.href)) || (/stage=shipping/.test(window.location.href))){

console.log("fbLogin is"+fbLogin);

if(fbLogin === true){

const userEmail = Evergage.cashDom('.checkout-email').text().trim();

if (userEmail) {

Evergage.sendEvent({

action: "Facebook Guest Checkout",

user: {

attributes: {

email: userEmail,

emailAddress: userEmail

}

}

});

}

}

}

},3500);

}

return actionEvent;

}

},

// Order Confirmation Page

{

name: "order\_confirmation",

action: "Order Confirmation",

isMatch: () => {return dataLayer[0].pageinfo.page\_type === "Order Confirmation" ||

/\/Order\-Confirm/.test(window.location.href);},

itemAction: Evergage.ItemAction.Purchase,

catalog: {

Product: {

lineItems: {

\_id: () => {

const productId = [];

var products = dataLayerProduct;

if(products.length > 0){

for(let i=0; i<products.length;i++){

productId[i]=products[i].product\_sku;

}

}

return productId;

},

price: () =>{

const productPrice = [];

var products = dataLayerProduct;

if(products.length > 0){

for(let i=0; i<products.length;i++){

productPrice[i]=products[i].price;

}

}

return productPrice;

},

quantity: () => {

const productQuantity = [];

var products = dataLayerProduct;

if(products.length > 0){

for(let i=0; i<products.length;i++){

productQuantity[i]=products[i].quantity;

}

}

return productQuantity;

}

}

}

}

},

{

name: "search\_page",

action: "Site Search",

isMatch: () => /\/search/.test(window.location.pathname) && /q=/.test(window.location.href)

}

] //end of page types block

}; // end of config block

// check dataLayer at Cart Page

function checkDataLayerAtCart(){

var products;

if (window.dataLayer) {

for (let i = 0; i < window.dataLayer.length; i++) {

if ((window.dataLayer[i].ecommerce && window.dataLayer[i].ecommerce.checkout || {}).products) {

products = window.dataLayer[i].ecommerce.checkout.products;

console.log("product at Cart " );

console.log(products);

return products;

}

}

}

}

function getProductsFromOldDataLayer(){

var products;

if (window.dataLayer) {

for (let i = 0; i < window.dataLayer.length; i++) {

if ((window.dataLayer[i].ecommerce && window.dataLayer[i].ecommerce.detail || {}).products) {

products = window.dataLayer[i].ecommerce.detail.products;

}

}

console.log("logging products from old dataLayer : ", products);

return products;

}

}

/\*

Check for URL change every 2 seconds. If URL has changed, reinitialize beacon and sitemap.

\*/

if( dataLayer[0].pageinfo.page\_type === "PDP"){

history.pushState = ( function(f){return function pushState(){

var ret = f.apply(this, arguments);

window.dispatchEvent(new Event('pushstate'));

window.dispatchEvent(new Event('locationchange'));

return ret;

}})(history.pushState);

history.replaceState = ( function(f){return function replaceState(){

var ret = f.apply(this, arguments);

window.dispatchEvent(new Event('replacestate'));

window.dispatchEvent(new Event('locationchange'));

return ret;

}})(history.replaceState);

window.addEventListener('popstate',function(){

window.dispatchEvent(new Event('locationchange'))

});

window.addEventListener('locationchange', function(){

console.log('location changed!');

Evergage.init(Evergage.getConfig()).then(() => {

Evergage.initSitemap(Evergage.getState().config);

});

});

}

Evergage.initSitemap(config);

}); //end of init block

LP pop up

export class EmailCapturePopupTemplate implements CampaignTemplateComponent {

@title('Background Image URL')

imageUrl: string;

style: "Dark on Light" | "Light on Dark";

@richText(true)

header: string;

@title('CTA Text')

@richText(true)

ctaText: string;

@title('CTA Destination URL')

@subtitle("Enter Destination URL")

ctaUrl: string;

@title("User Attributes To Retrieve")

@subtitle("attributes must be a pipe separated")

userAttributes: string;

run(context:CampaignComponentContext) {

let attrs = this.userAttributes.split("|")

let attributesToSend = {}

attrs.filter((value) => {

return context.user.attributes.hasOwnProperty(value);

}).map((value2) => {

attributesToSend[value2] = {

value: context.user.attributes[value2]

}

});

return {

attributes: attributesToSend

};

}

}

--------------------

(function () {

function apply(context, template) {

if (parseInt(window.localStorage.liveAgentPc) == 2 && !(sessionStorage.getItem('oldSession'))) {

// if (true) {

var html = template(context);

Evergage.cashDom("body").append(html);

// Style config setup

if (context.style === "Dark on Light") {

Evergage.cashDom("#evg-email-capture-popup, #evg-email-capture-popup .evg-cta").addClass("evg-dark");

} else {

Evergage.cashDom("#evg-email-capture-popup, #evg-email-capture-popup .evg-cta").addClass("evg-light");

}

// Dismisses popup

Evergage.cashDom("#evg-email-capture-popup .evg-overlay, #evg-email-capture-popup .evg-close, #evg-email-capture-popup .evg-opt-out").on("click", function () {

Evergage.cashDom("#evg-email-capture-popup").remove();

});

// Shows secondary confirmation panel

Evergage.cashDom("#evg-email-capture-popup .evg-cta").on("click", function (event) {

Evergage.cashDom("#evg-email-capture-popup").remove();

});

sessionStorage.setItem('oldSession', '1');

}

}

function reset(context, template) {

var popup = Evergage.cashDom("#evg-email-capture-popup");

if (popup) popup.remove();

}

function control() {

}

registerTemplate({

apply: apply,

reset: reset,

control: control

});

})();

----------------------------------------

<div id="evg-email-capture-popup" data-evg-campaign="{{campaign}}:{{experience}}" data-evg-campaign-group="test">

<div class="evg-overlay"></div>

<div class="evg-popup" style="background-image:url('{{imageUrl}}');">

<div class="evg-content">

<div class="evg-header">

{{{header}}}

</div>

<div id="couponCode" class="evg-subheader">

{{{attributes.welcomeJourneyCouponCode.value.value}}}

</div>

<div class="evg-form" action="" method="">

<button class="evg-cta" id="copyButton" type="button">

<a href="{{ctaUrl}}">

{{{ctaText}}}

</a>

</button>

</div>

<script>

const couponCode = document.getElementById("couponCode").innerHTML.trim();

const copyButton = document.querySelector("#copyButton");

copyButton.addEventListener('click', function(){

var dummy = document.createElement("input");

document.body.appendChild(dummy);

dummy.setAttribute("id", "dummy\_id");

document.getElementById("dummy\_id").value=couponCode;

dummy.select();

document.execCommand("copy");

document.body.removeChild(dummy);

});

</script>

<div class="evg-close" evergage-dismissonclick="evergage-dismissonclick">&times;</div>

</div>

</div>

Adobe target at-js

/\* This function has been updated for AT.js Changes . Fix for Demandbase Params first hit. Data Layer Variables removed from target \*/

function targetPageParams() {

var pageURL= window.location.href;

var kruxSegments = localStorage.getItem("kxvmwarecm2\_segs");

if (typeof dbInfo === 'undefined' && typeof sessionStorage.db !== 'undefined') {

dbInfo = JSON.parse(sessionStorage.db);

loadDB = true;

setdata(dbInfo);

}

else{

sessionStorage.db=JSON.stringify(db);

dbInfo=JSON.parse(sessionStorage.db);

setdata(dbInfo);

}

return {

"krux\_Segments": kruxSegments,

"profile": {

"company": pp1,

"industry": pp2,

"sub\_industry": pp3,

"employee\_range": pp4,

"city": pp5,

"state": pp6,

"country": pp7,

"audience": pp8,

"isp": pp9,

"annual\_sales": pp10,

"b2b": pp11,

"b2c": pp12,

"site\_locale": country,

"account\_watch": pp13,

"multiple\_account\_watch": pp14,

"marketing\_alias": pp15

}

};

}

targetPageParamsAll = function() {

if (typeof dbInfo === 'undefined' && typeof sessionStorage.db !== 'undefined') {

dbInfo = JSON.parse(sessionStorage.db);

loadDB = true;

setdata(dbInfo);

}

return {

"pagecontenttype": digitalData.page.attributes.pageContentType,

"journeystage": digitalData.page.attributes.journeyStage,

"score": digitalData.page.attributes.score,

"profile": {

"company": pp1,

"industry": pp2,

"sub\_industry": pp3,

"employee\_range": pp4,

"city": pp5,

"state": pp6,

"country": pp7,

"audience": pp8,

"isp": pp9,

"annual\_sales": pp10,

"b2b": pp11,

"b2c": pp12,

"site\_locale": digitalData.page.pageInfo.geoRegion,

"account\_watch": pp13,

"multiple\_account\_watch": pp14

}

};

}

\*/

Product aggregation code.

<script>

utag = '';

if($('.product-tile[data-position]').length <=0) {

$('.product-tile').each(function(index){

$(this).attr('data-position',index+1);

$(this).attr('data-imp-fired', 'false');

});

}

if(dataLayer[dataLayer.length-1]["event"] == "gtm.elementVisibility"){

dataLayer[dataLayer.length-1]["checkURL"] = document.URL;

}

</script>

<script>

$.fn.isInViewport = function() {

var elementTop = $(this).closest('.grid-tile').offset().top;

var elementBottom = elementTop + $(this).closest('.grid-tile').outerHeight();

var viewportTop = $(window).scrollTop();

var viewportBottom = viewportTop + $(window).innerHeight();

return elementBottom > viewportTop && elementBottom < viewportBottom;

};

</script>

<script>

setTimeout(function(){

var impression = [];

$('.product-tile').each(function(){

if ($(this).isInViewport()) {

console.log("in Viewport");

if($(this).attr('data-imp-fired') == "false") {

var products = {};

products["list"] = utag\_data["page\_name"];

products["name"] = $(this).attr("data-itemname");

products["id"] = $(this).attr("data-itemsku");

products["position"] = $(this).attr("data-position");

products["brand"] = $(this).attr("data-itembrand");

products["category"] = $(this).attr("data-category");

products["price"] = $(this).attr("data-itemprice");

products["variant"] = ($(this).attr('data-variants')=='false')?"":$(this).attr('data-variants');

impression.push(products);

$(this).attr('data-imp-fired', 'true');

}

}

});

if(dataLayer.length > 3) {

if(dataLayer[dataLayer.length-3]["event"] == "gtm.elementVisibility" && dataLayer[dataLayer.length-1]["event"] == "gtm.elementVisibility" && (dataLayer[dataLayer.length-3]["checkURL"] == dataLayer[dataLayer.length-1]["checkURL"])){}

else {

if(impression.length > 0){

dataLayer.push({

'event': 'product-impression',

'ecommerce':{

'impressions': impression

}

});

}

}

}

else {

if(impression.length > 0){

dataLayer.push({

'event': 'product-impression',

'ecommerce':{

'impressions': impression

}

});

}

}},1000);

</script>

<script>

$.fn.isInViewport = function() {

var elementTop = $(this).closest('.grid-tile').offset().top;

var elementBottom = elementTop + $(this).closest('.grid-tile').outerHeight();

var viewportTop = $(window).scrollTop();

var viewportBottom = viewportTop + $(window).innerHeight();

return elementBottom > viewportTop && elementBottom < viewportBottom;

};

</script>

<script>

$(window).scroll(function(){

var impression = [];

$('.product-tile').each(function(){

if ($(this).isInViewport()) {

if($(this).attr('data-imp-fired') == "false") {

var products = {};

products["list"] = utag\_data["page\_name"];

products["name"] = $(this).attr("data-itemname");

products["id"] = $(this).attr("data-itemsku");

products["position"] = $(this).attr("data-position");

products["brand"] = $(this).attr("data-itembrand");

products["category"] = $(this).attr("data-category");

products["price"] = $(this).attr("data-itemprice");

products["variant"] = ($(this).attr('data-variants')=='false')?"":$(this).attr('data-variants');

impression.push(products);

$(this).attr('data-imp-fired', 'true');

}

}

});

if(impression.length > 0){

dataLayer.push({

'event': 'product-impression',

'ecommerce':{

'impressions': impression

}

});

}

});

</script>

h\_ttps://mc.s11.exacttarget.com

Deloitte\_Training

Deloitte@123

akshaykashyap2+partnerdeloitte@deloitte.com

munger1331!!!

commerce cloud owd : munger1331!!!

https://bfgf-025.sandbox.us01.dx.commercecloud.salesforce.com/on/demandware.store/Sites-Site

"username": "ameriga@deloitte.com",

"password": "Sandbox@03",

If version is not activated you can activate it here.

Administration->Code version -> Activate "OneTrustVersion"

Storefront URL: https://bfgf-025.sandbox.us01.dx.commercecloud.salesforce.com/on/demandware.store/Sites-gracobaby-Site​​​​​​​

system pwd - munger1331!@#

Authorization Code:

A6ED-BD4E-2267-0C7F

Authorization Code #2:

EF49-2AEC-027B-1CDF

3B17-F6DD-CE35-1293

?evergageEditor=true

apptio event id -APP-9780-XRBC

testnewellqa+graco5@gmail.com

Deloitte@1989

https://newellbrands.atlassian.net/wiki/spaces/DTI/pages/5816017/Test+Credit+Cards

https://newellbrands.atlassian.net/wiki/spaces/DigitalTech/pages/1214612723/Shipping+Addresses+for+Orders

newell.product.test@gmail.com

Deloitte@1990

https://training.evergage.com/ui/index.html

debhattacharyya+training@deloitte.com

Deloitte@123

4112344112344113

4110144110144115

https://www.accenture.com/in-en/careers/jobdetails?id=245773\_india\_1&title=Hadoop

https://careers.ibm.com/ShowJob/Id/822746/Data-Engineer-Big-Data/

ID - 282877BR

https://jobs.americanexpress.com/india/jobs/20006113?lang=en-us&previousLocale=en-US

https://ea.gr8people.com/jobs/162417/data-engineer-i?geo\_location=ChIJx9Lr6tqZyzsRwvu6koO3k64

342 hrs

https://quip.com/pTHJAiKDcCfW

dev\_marmot - vMbB8CChydVeynFXs5JwYqYsIT9vysgZjj9JZKFBiUU

dev\_bbj - MTdK5H4nAsQ1LxPSaiCYoqYS8cXYQvKxM21o3y7U3Ao

qa\_exo -

7su6pWxArAmOvT5rQn-mWNW\_xBMZzbQUO2f8RYN13Ao

4112344112344113

https://googleapis.dev/python/bigquery/latest/index.html

https://doc.evergage.com/display/EKB/API+Tokens

https://doc.evergage.com/display/EKB/Send+Events+Server-side+Using+HTTP+Requests

Voucher Code: AJJ22J2ddw6N468J

Expiration Date: June 1, 2022

Soujanya.kona@newellco.com Chaitu@1521

https://newellbrands.atlassian.net/issues/?filter=22460

quip.com/lnezA27wTyaX

https://quip.com/CAGiAB4kMLW6

SSO PWD - Itachi2568

EP2020IR184536

sourabh.gandhi1@in.ey.com

srikmishra@deloitte.com

https://newellbrands.atlassian.net/issues/?jql=filter%20in%20(%2222690%22)%20AND%20sprint%20%3D%20795

https://developer.salesforce.com/docs/atlas.en-us.c360a\_api.meta/c360a\_api/c360a\_api\_quick\_start.htm

https://salesforce.zoom.us/webinar/register/7316207660888/WN\_67WLa-GlT3eeaNoKEoNCXA

https://salesforce.zoom.us/webinar/register/7016207661136/WN\_zzNqOddmRhCHA\_F5MXK8Zw

8:30 - 11 3rd n 4th june

https://www.salesforce.com/au/form/events/webinars/form-rss/3134127?

https://www.salesforce.com/form/events/webinars/form-rss/3135688

9:30 am , 17th june

523621.

523622.

9392912709

15th july - 6:45 , 6:15.

debhattacharyya\_deloitte

Deloitte@123

http://i.refs.cc/JlJ1mIt5?smile\_ref=eyJzbWlsZV9zb3VyY2UiOiJzbWlsZV91aSIsInNtaWxlX21lZGl1bSI6IiIsInNtaWxlX2NhbXBhaWduIjoicmVmZXJyYWxfcHJvZ3JhbSIsInNtaWxlX2N1c3RvbWVyX2lkIjpudWxsfQ%3D%3D

https://www.npmjs.com/package/exit-intent