<!DOCTYPE html>

<html lang="en">

<head>

<!-- Required Meta Tags -->

<meta charset="UTF-8">

<title>Vista Charm - Enhanced Stay</title>

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<!-- PWA Primary Theme Color -->

<meta name="theme-color" content="#FF5A5F">

<!-- SEO / General Meta Tags -->

<meta name="description" content="Enhance your stay with Vista Charm. Manage your check-in and check-out times, access amenities, and more.">

<meta name="keywords" content="Vista Charm, Hotel, Accommodation, Check-In, Check-Out, Amenities">

<meta name="author" content="Vista Charm Team">

<!-- PWA & iOS Meta Tags -->

<!-- If you use a separate manifest.json file, remove the inline script below and keep this link -->

<link rel="manifest" href="#inline-manifest">

<meta name="apple-mobile-web-app-capable" content="yes">

<meta name="apple-mobile-web-app-title" content="Vista Charm - Enhanced Stay">

<meta name="apple-mobile-web-app-status-bar-style" content="black-translucent">

<meta name="format-detection" content="telephone=no">

<!-- Open Graph Meta Tags -->

<meta property="og:title" content="Vista Charm - Enhance Your Stay">

<meta property="og:description" content="Manage your check-in and check-out times, access amenities, and more with Vista Charm.">

<meta property="og:image" content="https://cdn.shopify.com/s/files/1/0901/5171/5111/files/IMG\_1960.png">

<meta property="og:url" content="https://yourwebsite.com">

<meta property="og:type" content="website">

<!-- Twitter Card Meta Tags -->

<meta name="twitter:card" content="summary\_large\_image">

<meta name="twitter:title" content="Vista Charm - Enhance Your Stay">

<meta name="twitter:description" content="Manage your check-in and check-out times, access amenities, and more with Vista Charm.">

<meta name="twitter:image" content="https://via.placeholder.com/1200x630.png?text=Vista+Charm">

<!-- Font Awesome for Icons -->

<link rel="stylesheet"

href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css"

integrity="sha512-p6VhT+e1h4KHmOjYAlL+5kGdnNq0k9jXh2gHl9c6VjU1BvO36a2plvqN5J6FnvCj1mnhSItK4U8H2X45yg5S1w=="

crossorigin="anonymous" referrerpolicy="no-referrer">

<!-- Google Fonts: Montserrat & Lato -->

<link

href="https://fonts.googleapis.com/css2?family=Lato:wght@400;500;700&amp;family=Montserrat:wght@700&amp;display=swap"

rel="stylesheet">

<!-- Litepicker CSS for Modern Date-Range Calendar -->

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/litepicker/dist/css/litepicker.css">

<!--

Inline manifest.json

Normally, you'd have a separate manifest.json file at your site root.

For a single-file approach, you can keep it as a script tag with type="application/json".

Make sure <link rel="manifest" href="#inline-manifest"> is set as above, so some browsers

may parse it, though support can vary.

-->

<script type="application/json" id="inline-manifest">

{

"name": "Vista Charm - Enhanced Stay",

"short\_name": "VistaCharm",

"start\_url": "/?source=pwa",

"display": "standalone",

"background\_color": "#FF5A5F",

"theme\_color": "#FF5A5F",

"icons": [

{

"src": "https://cdn.shopify.com/s/files/1/0901/5171/5111/files/IMG\_1960.png",

"sizes": "192x192",

"type": "image/png"

},

{

"src": "https://cdn.shopify.com/s/files/1/0901/5171/5111/files/IMG\_1960.png",

"sizes": "512x512",

"type": "image/png"

}

]

}

</script>

<!-- Custom CSS -->

<style>

/\* ------------------------------------------------------

Your custom CSS

------------------------------------------------------ \*/

/\* CSS Reset \*/

\*,

\*::before,

\*::after {

margin: 0;

padding: 0;

box-sizing: border-box;

}

/\* Root Variables \*/

:root {

--primary-color: #FF5A5F; /\* Airbnb Red \*/

--secondary-color: #484848; /\* Dark Gray \*/

--background-color: #FFFFFF; /\* White \*/

--text-color: #484848; /\* Dark Gray \*/

--card-background: #F8F8F8; /\* Light Gray \*/

--button-background: #FF5A5F; /\* Airbnb Red \*/

--button-text: #FFFFFF; /\* White \*/

--button-hover: #E04850; /\* Darker Red \*/

--modal-overlay: rgba(0, 0, 0, 0.6);

--transition-speed: 0.3s;

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

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

--border-radius: 12px;

--border-color: #CCCCCC;

--scrollbar-track-color: #E0E0E0;

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

--toggle-hover-background: rgba(255, 90, 95, 0.2);

--card-shadow: rgba(0, 0, 0, 0.1) 0px 4px 12px;

--card-hover-shadow: rgba(0, 0, 0, 0.2) 0px 8px 24px;

--card-transition: transform var(--transition-speed) ease, box-shadow var(--transition-speed) ease;

--error-color: #e74c3c;

--available-bg: #e0f7fa;

--upgrade-bg: #ffe0b2;

--button-selected-bg: #FFB3B3;

--checkbox-size: 20px;

--checkbox-border: 2px solid var(--primary-color);

--checkbox-bg: var(--background-color);

--checkbox-checked-bg: var(--primary-color);

--checkbox-checked-icon: #FFFFFF;

}

/\* Dark Theme Variables \*/

[data-theme="dark"] {

--primary-color: #FF5A5F;

--secondary-color: #E0E0E0;

--background-color: #121212;

--text-color: #ffffff;

--card-background: #1E1E1E;

--button-background: #FF5A5F;

--button-text: #121212;

--button-hover: #E04850;

--border-color: #555555;

--scrollbar-track-color: #555555;

--toggle-background: rgba(18, 18, 18, 0.1);

--toggle-hover-background: rgba(255, 90, 95, 0.2);

--card-shadow: rgba(0, 0, 0, 0.3) 0px 4px 12px;

--card-hover-shadow: rgba(0, 0, 0, 0.4) 0px 8px 24px;

}

/\* Force certain text to remain black even in dark mode \*/

[data-theme="dark"] .section-rich-text,

[data-theme="dark"] .section-description {

color: black !important;

}

[data-theme="dark"] .reservation-details .reservation-card-body p {

color: black !important;

}

/\* Body Styles \*/

body {

font-family: var(--font-family);

background-color: var(--background-color);

color: var(--text-color);

line-height: 1.6;

margin: 0;

padding: 0;

transition:

background-color var(--transition-speed) ease,

color var(--transition-speed) ease;

}

/\* Header Fonts \*/

h1, h2, h3 {

font-family: var(--header-font-family);

}

/\* Container and Disabled Overlay \*/

#main-content.disabled {

pointer-events: none;

filter: blur(3px);

opacity: 0.8;

}

.container {

max-width: 1200px;

margin: 0 auto;

padding: 20px;

}

main {

padding: 20px 0;

}

/\* Section Titles \*/

.section-title {

font-size: 2.2rem;

margin-bottom: 10px;

color: var(--primary-color);

text-align: center;

font-weight: 700;

}

.section-title-1 {

font-size: 1.6rem;

margin-bottom: 10px;

color: var(--primary-color);

text-align: center;

font-weight: 700;

}

/\* Section Subtitles \*/

.section-subtitle {

font-size: 1.8rem;

margin-bottom: 10px;

color: #FF5A5F;

text-align: center;

}

/\* Section Rich Text \*/

.section-rich-text,

.section-description {

font-size: 1rem;

color: var(--text-color);

max-width: 800px;

margin: 0 auto 20px auto;

text-align: center;

}

/\* Shared Grid \*/

.shared-grid {

display: flex;

flex-wrap: nowrap;

gap: 20px;

margin-bottom: 40px;

margin-left: auto;

margin-right: auto;

overflow-x: auto;

padding-bottom: 10px;

scroll-behavior: smooth;

justify-content: flex-start;

}

.shared-grid::-webkit-scrollbar {

height: 8px;

}

.shared-grid::-webkit-scrollbar-track {

background: var(--scrollbar-track-color);

border-radius: 4px;

}

.shared-grid::-webkit-scrollbar-thumb {

background: var(--primary-color);

border-radius: 4px;

}

.shared-grid::-webkit-scrollbar-thumb:hover {

background: var(--button-hover);

}

/\* Cards \*/

.card {

background-color: var(--card-background);

border-radius: var(--border-radius);

box-shadow: var(--card-shadow);

padding: 20px;

text-align: center;

transition: var(--card-transition);

cursor: pointer;

display: flex;

flex-direction: column;

align-items: center;

min-width: 280px;

flex-shrink: 0;

border: 1px solid var(--border-color);

position: relative;

overflow: hidden;

background-clip: padding-box;

}

.card::before {

content: '';

position: absolute;

top: 0;

left: 0;

width: 100%;

height: 5px;

background-color: var(--primary-color);

border-top-left-radius: var(--border-radius);

border-top-right-radius: var(--border-radius);

opacity: 0.8;

}

.card:hover,

.card:focus {

transform: translateY(-10px);

box-shadow: var(--card-hover-shadow);

outline: none;

}

.card i {

font-size: 1.2rem;

color: var(--primary-color);

margin-bottom: 12px;

}

.card h3 {

font-size: 1.2rem;

margin-bottom: 15px;

color: var(--text-color);

font-weight: 500;

}

.card p {

font-size: 0.95rem;

color: var(--text-color);

margin-bottom: 8px;

line-height: 1.5;

position: relative;

transition: background-color 0.5s ease;

}

/\* Check-In & Check-Out Card Container \*/

.card-container {

max-width: 800px;

margin: 40px auto;

padding: 30px;

background-color: var(--card-background);

border-radius: var(--border-radius);

box-shadow: var(--card-shadow);

text-align: center;

transition: var(--card-transition);

display: flex;

flex-direction: column;

border: 1px solid var(--border-color);

position: relative;

}

.card-container:hover,

.card-container:focus {

transform: translateY(-10px);

box-shadow: var(--card-hover-shadow);

outline: none;

}

.card-container h2 {

font-size: 2rem;

color: var(--primary-color);

margin-bottom: 15px;

}

.checkin-checkout-wrapper {

display: flex;

justify-content: space-around;

align-items: stretch;

gap: 20px;

}

.checkin-checkout-section {

flex: 1;

background-color: var(--background-color);

border: 1px solid var(--primary-color);

border-radius: var(--border-radius);

padding: 20px;

box-shadow: var(--card-shadow);

display: flex;

flex-direction: column;

align-items: center;

cursor: pointer;

transition:

background-color var(--transition-speed) ease,

color var(--transition-speed) ease;

min-width: 200px;

position: relative;

background-clip: padding-box;

}

.checkin-checkout-section:hover,

.checkin-checkout-section:focus {

background-color: var(--primary-color);

color: var(--button-text);

outline: none;

box-shadow: var(--card-hover-shadow);

}

.checkin-checkout-section:hover .timer,

.checkin-checkout-section:focus .timer {

color: var(--button-text);

}

.checkin-checkout-section strong {

font-size: 1.2rem;

margin-bottom: 10px;

}

.date-display {

font-size: 1.4rem;

color: var(--secondary-color);

margin-top: 5px;

}

.timer {

font-size: 1.4rem;

color: var(--primary-color);

margin: 15px 0;

font-weight: 600;

}

.timer-countdown {

font-size: 1.2rem;

color: var(--primary-color);

margin: 15px 0;

font-weight: 600;

}

/\* Dark Mode & Cart Buttons (top area) \*/

.top-buttons-container {

position: absolute;

top: 10px;

left: 10px;

right: 10px;

display: flex;

justify-content: space-between;

align-items: center;

}

#theme-toggle-button,

#cart-button,

#settings-button {

display: flex;

align-items: center;

justify-content: center;

background: var(--toggle-background);

border: 2px solid var(--primary-color);

border-radius: 50%;

padding: 10px;

cursor: pointer;

font-size: 1rem;

color: var(--text-color);

transition:

color var(--transition-speed) ease,

background var(--transition-speed) ease;

user-select: none;

touch-action: none;

}

#theme-toggle-button i {

font-size: 1.8rem;

}

#theme-toggle-button:hover,

#theme-toggle-button:focus,

#cart-button:hover,

#cart-button:focus,

#settings-button:hover,

#settings-button:focus {

color: var(--button-text);

background: var(--toggle-hover-background);

outline: none;

}

/\* Enhancement Widgets \*/

.enhance-your-stay-wrapper {

display: flex;

flex-wrap: nowrap;

overflow-x: auto;

gap: 20px;

padding-bottom: 10px;

scroll-behavior: smooth;

justify-content: flex-start;

}

.enhance-your-stay-wrapper::-webkit-scrollbar {

height: 8px;

}

.enhance-your-stay-wrapper::-webkit-scrollbar-track {

background: var(--scrollbar-track-color);

border-radius: 4px;

}

.enhance-your-stay-wrapper::-webkit-scrollbar-thumb {

background: var(--primary-color);

border-radius: 4px;

}

.enhance-your-stay-wrapper::-webkit-scrollbar-thumb:hover {

background: var(--button-hover);

}

.enhancement-widget {

width: 220px;

height: 220px;

background-color: var(--card-background);

border-radius: var(--border-radius);

padding: 20px;

box-shadow: var(--card-shadow);

transition: var(--card-transition);

cursor: pointer;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

text-align: center;

flex-shrink: 0;

border: 1px solid var(--border-color);

position: relative;

overflow: hidden;

background-clip: padding-box;

}

.enhancement-widget::before {

content: '';

position: absolute;

top: 0;

left: 0;

width: 100%;

height: 5px;

background-color: var(--primary-color);

border-top-left-radius: var(--border-radius);

border-top-right-radius: var(--border-radius);

opacity: 0.8;

}

.enhancement-widget:hover,

.enhancement-widget:focus {

transform: translateY(-10px);

box-shadow: var(--card-hover-shadow);

outline: none;

}

.enhancement-widget i {

font-size: 2.5rem;

color: var(--primary-color);

margin-bottom: 15px;

}

.enhancement-widget h3 {

font-size: 1.2rem;

margin-bottom: 8px;

color: var(--text-color);

font-weight: 600;

}

.enhancement-widget p {

font-size: 1rem;

color: var(--text-color);

}

/\* FAQs \*/

.faq-section {

margin-bottom: 40px;

}

.faq-card {

background-color: var(--card-background);

border-radius: var(--border-radius);

box-shadow: var(--card-shadow);

padding: 25px;

transition: var(--card-transition);

border: 1px solid var(--border-color);

}

.faq-card:hover,

.faq-card:focus-within {

transform: translateY(-10px);

box-shadow: var(--card-hover-shadow);

outline: none;

}

.faq-item {

margin-bottom: 15px;

border-bottom: 1px solid var(--border-color);

padding-bottom: 15px;

}

.faq-item:last-child {

border-bottom: none;

}

.faq-item button {

width: 100%;

text-align: left;

background: none;

border: none;

padding: 10px 0;

font-size: 1.1rem;

font-weight: 600;

cursor: pointer;

color: var(--text-color);

display: flex;

justify-content: space-between;

align-items: center;

transition: color var(--transition-speed) ease;

}

.faq-item button:focus {

outline: none;

color: var(--primary-color);

}

.faq-answer {

max-height: 0;

overflow: hidden;

transition: max-height var(--transition-speed) ease;

}

.faq-answer p {

padding: 10px 0;

color: var(--text-color);

font-size: 1rem;

}

.faq-item.active .faq-answer {

max-height: 200px;

}

.faq-item.active button::after {

transform: rotate(180deg);

}

.faq-item button::after {

content: '\f078';

font-family: 'Font Awesome 6 Free';

font-weight: 900;

transition: transform var(--transition-speed) ease;

}

/\* Contact Section \*/

.contact-section {

margin-bottom: 40px;

}

.contact-card {

background-color: var(--card-background);

border-radius: var(--border-radius);

box-shadow: var(--card-shadow);

padding: 25px;

transition: var(--card-transition);

border: 1px solid var(--border-color);

position: relative;

}

.contact-card:hover,

.contact-card:focus-within {

transform: translateY(-10px);

box-shadow: var(--card-hover-shadow);

outline: none;

}

.contact-details {

text-align: center;

margin-bottom: 20px;

}

.contact-details p {

font-size: 1.1rem;

color: var(--text-color);

margin-bottom: 10px;

}

.contact-details a {

color: var(--primary-color);

text-decoration: none;

transition: color var(--transition-speed) ease;

}

.contact-details a:hover,

.contact-details a:focus {

color: var(--button-hover);

}

.share-buttons {

display: flex;

gap: 15px;

overflow-x: auto;

align-items: center;

justify-content: center;

padding-bottom: 10px;

scroll-behavior: smooth;

text-align: center;

margin-top: 10px;

}

.share-buttons::-webkit-scrollbar {

height: 8px;

}

.share-buttons::-webkit-scrollbar-track {

background: var(--scrollbar-track-color);

border-radius: 4px;

}

.share-buttons::-webkit-scrollbar-thumb {

background: var(--primary-color);

border-radius: 4px;

}

.share-buttons::-webkit-scrollbar-thumb:hover {

background: var(--button-hover);

}

.cta-button {

background-color: var(--button-background);

color: var(--button-text);

padding: 12px 25px;

border: none;

border-radius: 8px;

text-decoration: none;

font-size: 1.1rem;

cursor: pointer;

transition:

background-color var(--transition-speed) ease,

transform var(--transition-speed) ease,

box-shadow var(--transition-speed) ease;

font-weight: 600;

display: inline-flex;

align-items: center;

gap: 8px;

scroll-snap-align: start;

white-space: nowrap;

text-align: center;

}

.cta-button:hover,

.cta-button:focus {

background-color: var(--button-hover);

transform: translateY(-2px);

box-shadow: var(--card-hover-shadow);

outline: none;

}

/\* Reservation Details \*/

.reservation-details {

max-width: 600px;

margin: 20px auto;

padding: 25px;

border: 1px solid var(--border-color);

border-radius: var(--border-radius);

background: var(--card-background);

box-shadow: var(--card-shadow);

text-align: center;

display: block;

}

.reservation-card {

padding: 20px;

background: linear-gradient(

135deg,

#f7f7f7 25%,

#fff 25%,

#fff 50%,

#f7f7f7 50%,

#f7f7f7 75%,

#fff 75%,

#fff 100%

);

border-radius: var(--border-radius);

box-shadow: var(--card-shadow);

position: relative;

}

.reservation-card-header {

font-size: 1.8rem;

font-weight: 700;

color: var(--primary-color);

margin-bottom: 15px;

}

.reservation-card-body {

font-size: 1.1rem;

margin-bottom: 15px;

line-height: 1.6;

}

/\* Modals \*/

.modal {

display: none;

position: fixed;

z-index: 1000;

left: 0;

top: 0;

width: 100%;

height: 100vh;

background-color: var(--modal-overlay);

justify-content: center;

align-items: center;

padding: 20px;

opacity: 0;

transition: opacity var(--transition-speed) ease;

}

.modal.active {

display: flex;

opacity: 1;

}

.modal-content {

background-color: var(--card-background);

padding: 30px;

border-radius: var(--border-radius);

max-width: 500px;

width: 100%;

position: relative;

box-shadow: var(--card-hover-shadow);

animation: slideDown 0.3s ease;

overflow-y: auto;

max-height: 90vh;

border: 1px solid var(--border-color);

text-align: left;

}

@keyframes slideDown {

from {

transform: translateY(-50px);

opacity: 0;

}

to {

transform: translateY(0);

opacity: 1;

}

}

.close-button {

position: absolute;

top: 15px;

right: 20px;

background: none;

border: none;

font-size: 1.5rem;

color: var(--text-color);

cursor: pointer;

transition: color var(--transition-speed) ease;

}

.close-button:hover,

.close-button:focus {

color: var(--primary-color);

outline: none;

}

.modal-content h2 {

margin-bottom: 20px;

color: var(--text-color);

font-size: 1.8rem;

font-weight: 700;

text-align: center;

}

.modal-content p {

font-size: 1rem;

color: var(--text-color);

margin-bottom: 15px;

}

.modal-content form div {

margin-bottom: 20px;

}

.modal-content label {

display: block;

margin-bottom: 8px;

color: var(--text-color);

font-size: 1rem;

font-weight: 500;

}

.modal-content input,

.modal-content select,

.modal-content textarea {

width: 100%;

padding: 10px 12px;

border: 1px solid var(--border-color);

border-radius: 6px;

font-size: 1rem;

background-color: var(--background-color);

color: var(--text-color);

transition: border-color var(--transition-speed) ease;

}

.modal-content input:focus,

.modal-content select:focus,

.modal-content textarea:focus {

border-color: var(--primary-color);

outline: none;

}

.submit-button {

background-color: var(--button-background);

color: var(--button-text);

padding: 12px 20px;

border: none;

border-radius: 6px;

cursor: pointer;

font-size: 1.1rem;

transition:

background-color var(--transition-speed) ease,

transform var(--transition-speed) ease,

box-shadow var(--transition-speed) ease;

font-weight: 600;

width: 100%;

display: none; /\* Make visible with .visible \*/

}

.submit-button.visible {

display: block;

}

.submit-button:hover,

.submit-button:focus {

background-color: var(--button-hover);

transform: translateY(-2px);

box-shadow: var(--card-hover-shadow);

outline: none;

}

.submit-button:disabled {

background-color: var(--button-background, #ccc);

cursor: not-allowed;

opacity: 0.6;

box-shadow: none;

transform: none;

}

.error-message {

color: var(--error-color);

font-size: 0.9rem;

display: none;

}

.error-message.visible {

display: block;

}

/\* Upgrade Options \*/

.upgrade-options {

margin-top: 30px;

text-align: left;

}

.upgrade-option {

background-color: var(--upgrade-bg);

padding: 15px;

border-radius: var(--border-radius);

margin-bottom: 15px;

box-shadow: var(--card-shadow);

display: flex;

align-items: center;

justify-content: space-between;

flex-wrap: wrap;

position: relative;

}

.upgrade-option label {

font-size: 1rem;

color: var(--primary-color);

flex: 1 1 70%;

margin-bottom: 10px;

}

.upgrade-option button {

background-color: var(--button-background);

color: var(--button-text);

border: none;

padding: 8px 16px;

border-radius: var(--border-radius);

cursor: pointer;

transition: background-color var(--transition-speed);

font-size: 0.9rem;

flex: 1 1 25%;

min-width: 100px;

}

.upgrade-option button:hover {

background-color: var(--button-hover);

}

.upgrade-details {

display: none;

margin-top: 10px;

width: 100%;

padding-left: 20px;

}

.upgrade-details button {

width: 100%;

margin-bottom: 10px;

background-color: var(--button-background);

color: var(--button-text);

border: none;

padding: 8px 12px;

border-radius: var(--border-radius);

cursor: pointer;

transition: background-color var(--transition-speed);

font-size: 0.9rem;

}

.upgrade-details button:hover {

background-color: var(--button-hover);

}

.upgrade-button.selected {

background-color: var(--button-selected-bg);

color: var(--text-color);

}

/\* Additional centering for download guest pass button \*/

.download-guest-pass-container {

text-align: center;

margin-top: 20px;

}

/\* Welcome Overlay \*/

.welcome-overlay {

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100vh;

background: rgba(0, 0, 0, 0.7);

display: none;

justify-content: center;

align-items: center;

z-index: 2000;

}

.welcome-message {

background: var(--card-background);

border-radius: var(--border-radius);

padding: 40px;

text-align: center;

color: var(--text-color);

position: relative;

max-width: 300px;

width: 100%;

}

.welcome-message h2 {

margin-bottom: 20px;

font-size: 1.8rem;

color: var(--primary-color);

}

.dismiss-welcome {

background-color: var(--button-background);

color: var(--button-text);

border: none;

padding: 10px 20px;

border-radius: var(--border-radius);

font-size: 1rem;

cursor: pointer;

margin-top: 20px;

}

.dismiss-welcome:hover {

background-color: var(--button-hover);

}

/\* Cart Modal \*/

#cart-items-container {

margin: 15px 0;

}

#cart-items-container ul {

list-style-type: none;

padding: 0;

}

#cart-items-container ul li {

margin: 8px 0;

display: flex;

justify-content: space-between;

border-bottom: 1px solid var(--border-color);

padding-bottom: 5px;

}

#cart-total {

font-size: 1.2rem;

font-weight: bold;

text-align: right;

margin-top: 10px;

}

/\* Checklists Modals (Precheckin & Precheckout) \*/

#precheckin-checklist-modal .modal-content,

#precheckout-checklist-modal .modal-content {

text-align: left;

}

#precheckin-checklist-modal ul,

#precheckout-checklist-modal ul {

margin-left: 20px;

margin-bottom: 20px;

list-style-type: none;

}

#precheckin-checklist-modal ul li,

#precheckout-checklist-modal ul li {

margin-bottom: 10px;

display: flex;

align-items: center;

}

#precheckin-checklist-modal input[type="checkbox"],

#precheckout-checklist-modal input[type="checkbox"] {

width: var(--checkbox-size);

height: var(--checkbox-size);

margin-right: 10px;

appearance: none;

border: var(--checkbox-border);

border-radius: 4px;

background-color: var(--checkbox-bg);

cursor: pointer;

position: relative;

transition: background-color var(--transition-speed), border-color var(--transition-speed);

}

#precheckin-checklist-modal input[type="checkbox"]:checked,

#precheckout-checklist-modal input[type="checkbox"]:checked {

background-color: var(--checkbox-checked-bg);

border-color: var(--checkbox-checked-bg);

}

#precheckin-checklist-modal input[type="checkbox"]:checked::after,

#precheckout-checklist-modal input[type="checkbox"]:checked::after {

content: '\f00c';

font-family: 'Font Awesome 6 Free';

font-weight: 900;

color: var(--checkbox-checked-icon);

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

font-size: 1rem;

}

/\* Settings PIN Entry Modal \*/

#settings-pin-modal .modal-content {

text-align: center;

}

#settings-pin-modal h2 {

color: var(--primary-color);

}

#settings-pin-modal input[type="password"] {

width: 80%;

margin: 20px 0;

}

#settings-pin-modal .error-message {

display: block;

margin-top: 10px;

}

/\* Main Settings Modal (for PIN & Door Code changes, plus sub-feature triggers) \*/

#settings-modal .modal-content {

text-align: left;

}

#settings-modal h2 {

text-align: center;

color: var(--primary-color);

}

.settings-features {

display: flex;

flex-direction: column;

gap: 15px;

margin-bottom: 20px;

}

.settings-features button {

width: 100%;

}

/\* Force main-theme-image to 16:9 aspect ratio \*/

#main-theme-image {

width: 100%;

aspect-ratio: 16/9;

object-fit: cover;

border-radius: var(--border-radius);

margin: 20px 0;

}

</style>

</head>

<body>

<!-- All main content is disabled until form submission -->

<div id="main-content" class="container disabled">

<main>

<!-- Check-In & Check-Out Times Section -->

<section tabindex="0" aria-labelledby="checkin-checkout-title" id="checkin-checkout-card" class="card-container">

<!-- Dark Mode & Cart Buttons (top-left & top-right) -->

<div class="top-buttons-container">

<!-- Theme Toggle Button (Dark Mode) -->

<button aria-label="Toggle Dark Mode" id="theme-toggle-button">

<i aria-hidden="true" id="theme-icon" class="fas fa-moon"></i>

</button>

<!-- Cart Button (Icon Only) -->

<button aria-label="View Cart" id="cart-button">

<i class="fas fa-shopping-cart"></i>

</button>

</div>

<h2 id="checkin-checkout-title">Check-In &amp; Check-Out Times</h2>

<div class="checkin-checkout-wrapper">

<!-- Check-In Section -->

<div tabindex="0" aria-label="Set Check-In Time" id="checkin-section" class="checkin-checkout-section">

<strong>Check-In Time</strong>

<p id="checkin-display" class="timer">--:--:--</p>

<p id="checkin-date" class="date-display">--/--/----</p>

<div aria-live="polite" id="checkin-timer" class="timer-countdown">--:--:--</div>

</div>

<!-- Check-Out Section -->

<div tabindex="0" aria-label="Set Check-Out Time" id="checkout-section" class="checkin-checkout-section">

<strong>Check-Out Time</strong>

<p id="checkout-display" class="timer">--:--:--</p>

<p id="checkout-date" class="date-display">--/--/----</p>

<div aria-live="polite" id="checkout-timer" class="timer-countdown">--:--:--</div>

</div>

</div>

</section>

<!-- Stay Overview Section -->

<section aria-labelledby="stay-overview-title">

<h1 id="stay-overview-title" class="section-title">Your Stay at Vista Charm</h1>

<p class="section-description">

Easily manage your check-in and check-out times according to your stay.

</p>

<img alt="Vista Charm" id="main-theme-image"

src="https://cdn.shopify.com/s/files/1/0671/8111/7609/files/IMG\_1443.jpg?v=1729416221">

</section>

<!-- Pre-Check-In Information Section -->

<section aria-labelledby="pre-checkin-info-title" id="pre-checkin-section">

<h2 id="pre-checkin-info-title" class="section-subtitle">Pre-Check-In Information</h2>

<p class="section-rich-text">

To ensure a smooth &amp; enjoyable experience, review our guidelines and policies.

</p>

<!-- Button that triggers the Pre-Check-In Checklist Modal -->

<p class="section-rich-text">

<button id="toggle-precheckin-button" class="cta-button" type="button">

Show Pre-Check-In Info

</button>

</p>

<!-- Wrap the pre-checkin cards in a container we can show/hide -->

<div id="precheckin-cards" style="display: none;">

<div class="shared-grid">

<!-- Before Check-In Card -->

<div aria-labelledby="information-title-1" tabindex="0" id="before-checkin-card" class="card">

<i aria-hidden="true" class="fas fa-info-circle"></i>

<h3 id="information-title-1">Before Check-In</h3>

<p>Max Guests: 2 | No Visitors</p>

<p>No Smoking / No Pets</p>

<p>Parking: 1 spot available</p>

<p>Respect Neighbors: Keep noise levels down.</p>

<p>Security: Cameras active on premises.</p>

<p><strong>Door Code: 8325</strong></p>

<p>Cleanliness: Please maintain cleanliness for a refund.</p>

</div>

<!-- House Rules Card -->

<div aria-labelledby="information-title-2" tabindex="0" id="house-rules-card" class="card">

<i aria-hidden="true" class="fas fa-balance-scale"></i>

<h3 id="information-title-2">House Rules</h3>

<p>Check-In: After 3:00 PM</p>

<p>Check-Out: Before 11:00 AM</p>

<p>No Parties: Strictly no events.</p>

<p>Quiet Hours: 10:00 PM - 8:00 AM</p>

<p>Garbage: Dispose of waste properly.</p>

</div>

<!-- Amenities Card -->

<div aria-labelledby="amenities-title" tabindex="0" id="amenities-card" class="card">

<h3 id="amenities-title">Amenities</h3>

<p>Wi-Fi: High-speed internet</p>

<p>Breakfast: Complimentary continental breakfast</p>

<p>Parking: Free parking available</p>

<p>Air Conditioning: In every room</p>

<p>24/7 Support: We're here to help anytime</p>

</div>

</div>

</div>

</section>

<!-- Enhance Your Stay Section -->

<section aria-labelledby="enhancements-title" id="enhance-your-stay-section">

<h2 id="enhancements-title" class="section-title-1">Enhance Your Stay</h2>

<p class="section-description">

Add extra amenities or services for an even more comfortable &amp; enjoyable stay.

</p>

<div class="enhance-your-stay-wrapper">

<!-- Early Check-In -->

<div aria-label="Early Check-in" tabindex="0" data-modal="early-checkin-modal" class="enhancement-widget">

<i class="fas fa-clock"></i>

<h3>Early Check-In</h3>

<p>Arrive earlier for added convenience.</p>

</div>

<!-- Late Check-Out -->

<div aria-label="Late Check-out" tabindex="0" data-modal="late-checkout-modal" class="enhancement-widget">

<i class="fas fa-calendar-alt"></i>

<h3>Late Check-Out</h3>

<p>Extend your stay beyond standard time.</p>

</div>

<!-- Toiletries & Essentials -->

<div aria-label="Toiletries &amp; Essentials" tabindex="0" data-modal="toiletries-modal"

class="enhancement-widget">

<i class="fas fa-bath"></i>

<h3>Toiletries</h3>

<p>Complimentary essentials for your comfort.</p>

</div>

<!-- Detergents -->

<div aria-label="Detergents" tabindex="0" data-modal="detergents-modal" class="enhancement-widget">

<i class="fas fa-tshirt"></i>

<h3>Detergents</h3>

<p>Keep your clothes fresh and clean.</p>

</div>

<!-- Food & Snacks -->

<div aria-label="Food &amp; Snacks" tabindex="0" data-modal="food-snacks-modal" class="enhancement-widget">

<i class="fas fa-utensils"></i>

<h3>Food &amp; Snacks</h3>

<p>Enjoy complimentary treats 24/7.</p>

</div>

</div>

</section>

<!-- Pre-Check-Out Information Section -->

<section aria-labelledby="pre-checkout-info-title" id="pre-checkout-section">

<h2 id="pre-checkout-info-title" class="section-subtitle">Pre-Check-Out Information</h2>

<p class="section-rich-text">

To ensure a smooth &amp; enjoyable experience, review these guidelines.

</p>

<!-- Button that triggers the Pre-Check-Out Checklist Modal -->

<p class="section-rich-text">

<button id="toggle-precheckout-button" class="cta-button" type="button">

Show Pre-Check-Out Info

</button>

</p>

<!-- Wrap the pre-checkout cards in a container we can show/hide -->

<div id="precheckout-cards" style="display: none;">

<div class="shared-grid">

<!-- Check-Out Instructions Card -->

<div aria-labelledby="checkout-instructions-title" tabindex="0" id="checkout-instructions-card" class="card">

<i aria-hidden="true" class="fas fa-info-circle"></i>

<h3 id="checkout-instructions-title">Check-Out Instructions</h3>

<p>Remove Waste: Use provided bins</p>

<p>Dry Towels: Hang towels to dry</p>

<p>Secure Premises: Lock doors &amp; windows</p>

<p>Laundry: Use facilities responsibly</p>

<p>Cleanliness: Leave space clean for refund</p>

</div>

<!-- Additional Checkout Steps Card -->

<div aria-labelledby="additional-checkout-steps-title" tabindex="0" id="additional-checkout-steps-card"

class="card">

<i aria-hidden="true" class="fas fa-door-closed"></i>

<h3 id="additional-checkout-steps-title">Additional Checkout Steps</h3>

<p>Return Keys: Place keys in lockbox</p>

<p>Thermostat: Set to 72°F</p>

<p>Lights &amp; AC: Turn off before leaving</p>

<p>Windows: Ensure all are closed</p>

<p>Leave a Review: We appreciate your feedback!</p>

</div>

</div>

</div>

</section>

<!-- FAQs Section -->

<section aria-labelledby="faq-title" class="faq-section">

<h2 id="faq-title" class="section-title">Frequently Asked Questions</h2>

<p class="section-description">

Find answers to common questions about your stay at Vista Charm.

</p>

<div class="faq-card">

<div class="faq-item">

<button aria-expanded="false">What is the check-in process?</button>

<div class="faq-answer">

<p>Check-in starts at 3:00 PM. Use our app to set your preferred time.</p>

</div>

</div>

<div class="faq-item">

<button aria-expanded="false">Can I request a late check-out?</button>

<div class="faq-answer">

<p>Yes, you can request a late check-out through the "Enhance Your Stay" section.</p>

</div>

</div>

<div class="faq-item">

<button aria-expanded="false">How do I use the door code?</button>

<div class="faq-answer">

<p>Enter the unique door code provided upon arrival to access your room.</p>

</div>

</div>

<div class="faq-item">

<button aria-expanded="false">Where do I park?</button>

<div class="faq-answer">

<p>We provide one parking spot in the driveway. Additional spots may be available on the street.</p>

</div>

</div>

</div>

</section>

<!-- Contact Us Section -->

<section aria-labelledby="contact-title" class="contact-section">

<h2 id="contact-title" class="section-title">Contact Us</h2>

<p class="section-description">

If you have any questions or need assistance, feel free to reach out!

</p>

<div class="contact-card">

<!-- Settings Button (Top-Right Corner) -->

<button aria-label="App Settings" id="settings-button">

<i class="fas fa-cog"></i>

</button>

<div class="contact-details">

<p><strong>Email:</strong> <a href="mailto:vistacharms@gmail.com">vistacharms@gmail.com</a></p>

<p><strong>Phone:</strong> <a href="tel:4073194093">(407) 319-4093</a></p>

</div>

<div class="share-buttons">

<a aria-label="Call Support" href="tel:4073194093" class="cta-button">Call Now</a>

<a aria-label="Email Support" href="mailto:vistacharms@gmail.com" class="cta-button">Email Us</a>

<a aria-label="Chat with Support"

href="https://airbnb.sng.link/A6f9up/lo82?\_dl=airbnb%3A%2F%2Fs%2Fhomes&amp;\_p=&amp;pcn=treatment\_g4\_u&amp;pscn=hp&amp;pcid=1713732469\_OTllZTE0ODFjOWU4&amp;\_smtype=3"

class="cta-button" target="\_blank">

Chat Now

</a>

</div>

</div>

</section>

<!-- Reservation Details -->

<section aria-labelledby="reservation-title">

<h2 id="reservation-title" class="section-title">Your Reservation Details</h2>

<div aria-labelledby="reservation-title" role="region" id="reservation-details" class="reservation-details">

<div class="reservation-card">

<div class="reservation-card-body">

<div class="section-title">Vista Charm</div>

<div>

<p>WiFi: <span id="wifi-name-field">Vista Charm</span></p>

<p>Password: <span id="wifi-password-field">Vista8325</span></p>

<p><strong>Door Code: <span id="doorcode-field">8325</span></strong></p>

</div>

<p>Check-In: <span id="preview-checkin">--/--/---- --:--</span></p>

<p>Check-Out: <span id="preview-checkout">--/--/---- --:--</span></p>

</div>

</div>

<div class="download-guest-pass-container">

<a aria-label="Download Your Guest Pass"

href="https://cdn.shopify.com/s/files/1/0671/8111/7609/files/4A5FA684-1F60-4706-B0A3-0D3D560DAACD.pkpass?v=1729416167"

id="download-guest-pass-button" class="cta-button" style="display: none;">

Download Your Guest Pass

</a>

</div>

</div>

</section>

</main>

<!-- Local Hidden Gems Button (placed at the bottom) -->

<div style="text-align: center; margin-bottom: 40px;">

<a aria-label="Local Hidden Gems" class="cta-button" href="#">Local Hidden Gems</a>

</div>

</div>

<!-- /#main-content ends here -->

<!-- Generate Passbook Modal (Active on Page Load) -->

<div aria-labelledby="generate-passbook-title" role="dialog" aria-modal="true" id="generate-passbook-modal"

class="modal active">

<div class="modal-content">

<h2 id="generate-passbook-title">Enter Your Reservation Details</h2>

<form class="passbook-form" id="passbook-form">

<!-- Single Input for Date Range Selection -->

<div style="flex-direction: column;" class="form-group-container">

<label for="daterange">Select Check-In and Check-Out Dates</label>

<input aria-label="Date Range" required readonly id="daterange" type="text">

<span id="error-checkin-datetime" class="error-message"></span>

<span id="error-checkout-datetime" class="error-message"></span>

</div>

<!-- Upgrade Options -->

<div class="upgrade-options">

<!-- Early Check-In Upgrade -->

<div id="upgrade-early-option" class="upgrade-option display-none">

<label for="early-checkin">Early Check-In (Before 3:00 PM)</label>

<button id="upgrade-early" type="button">Select Time</button>

<!-- Hidden Upgrade Details -->

<div id="early-upgrade-details" class="upgrade-details">

<button data-price="4900" data-time="12:00 PM" class="upgrade-button" type="button">12 PM ($49)</button>

<button data-price="3900" data-time="1:00 PM" class="upgrade-button" type="button">1 PM ($39)</button>

<button data-price="2900" data-time="2:00 PM" class="upgrade-button" type="button">2 PM ($29)</button>

<button data-price="0" data-time="3:00 PM" class="upgrade-button" type="button">3 PM (No Charge)</button>

</div>

</div>

<!-- Late Check-Out Upgrade -->

<div id="upgrade-late-option" class="upgrade-option display-none">

<label for="late-checkout">Late Check-Out (After 11:00 AM)</label>

<button id="upgrade-late" type="button">Select Time</button>

<!-- Hidden Upgrade Details -->

<div id="late-upgrade-details" class="upgrade-details">

<button data-price="2900" data-time="12:00 PM" class="upgrade-button" type="button">12 PM ($29)</button>

<button data-price="3900" data-time="1:00 PM" class="upgrade-button" type="button">1 PM ($39)</button>

<button data-price="4900" data-time="2:00 PM" class="upgrade-button" type="button">2 PM ($49)</button>

<button data-price="5900" data-time="3:00 PM" class="upgrade-button" type="button">3 PM ($59)</button>

<button data-price="0" data-time="11:00 AM" class="upgrade-button" type="button">11 AM (No Charge)</button>

</div>

</div>

</div>

<!-- Add guest name field -->

<div style="flex-direction: column;" class="form-group-container">

<label for="guest-name">Guest Name</label>

<input required placeholder="Enter your name" id="guest-name" type="text">

</div>

<!-- Generate Passbook Button -->

<button class="submit-button visible" id="generate-passbook-button" type="submit">Submit</button>

</form>

</div>

</div>

<!-- Upgrade Confirmation Modal -->

<div aria-labelledby="upgrade-confirmation-title" role="dialog" aria-modal="true" id="upgrade-confirmation-modal"

class="modal">

<div class="modal-content">

<button aria-label="Close Modal" class="close-button">×</button>

<h2 id="upgrade-confirmation-title">Confirm Upgrade</h2>

<p id="upgrade-confirmation-message">Are you sure you want to select this time?</p>

<button class="submit-button visible" id="confirm-upgrade">Confirm</button>

<button style="background-color: var(--error-color);" class="submit-button visible" id="cancel-upgrade">Cancel</button>

</div>

</div>

<!-- Add to Home Screen Modal -->

<div aria-labelledby="add-to-home-title" role="dialog" aria-modal="true" id="add-to-home-modal" class="modal">

<div class="modal-content">

<h2 id="add-to-home-title">Add to Home Screen</h2>

<p>To receive your security code, please add this app to your home screen.</p>

<p id="add-to-home-instructions">&nbsp;</p>

<button id="add-to-home-button" class="submit-button visible">Add to Home Screen</button>

<button id="your-guest-pass-button" class="submit-button visible">Your Guest Pass</button>

<a aria-label="Download Your Guest Pass"

download="GuestPass.pkpass"

href="https://cdn.shopify.com/s/files/1/0671/8111/7609/files/4A5FA684-1F60-4706-B0A3-0D3D560DAACD.pkpass?v=1729416167"

id="download-guest-pass" class="cta-button">

<i class="fas fa-download"></i> Download Your Guest Pass

</a>

</div>

</div>

<!-- EARLY CHECK-IN Modal (UPDATED WITH DROPDOWN) -->

<div aria-labelledby="early-checkin-title" role="dialog" aria-modal="true" id="early-checkin-modal" class="modal">

<div class="modal-content">

<button aria-label="Close Modal" class="close-button">×</button>

<h2 id="early-checkin-title">Early Check-In</h2>

<p>Need to check in early? Upgrade your arrival time for added convenience.</p>

<h3>Pricing</h3>

<p>Early check-in available for an additional fee.</p>

<!-- NEW/UPDATED CODE: Add a dropdown for multiple early check-in products/times. -->

<label style="margin-top: 10px;">Choose a Time:</label>

<select id="early-checkin-options">

<option value="12:00 PM" data-price="4900">12 PM - $49</option>

<option value="1:00 PM" data-price="3900">1 PM - $39</option>

<option value="2:00 PM" data-price="2900">2 PM - $29</option>

<option value="3:00 PM" data-price="0" selected>3 PM - No Charge</option>

</select>

<label style="margin-top: 10px;">Enter PIN (default "123"):</label>

<div style="display: flex; gap: 10px; margin: 10px 0;">

<input placeholder="Enter PIN" style="flex: 1;" id="early-checkin-pin" type="password">

<button style="flex: 1;" class="cta-button" onclick="confirmEarlyCheckInPin()">Confirm</button>

</div>

<button style="margin-top: 10px;" class="cta-button"

onclick="handleEarlyCheckinSelection(); closeModal('early-checkin-modal');">

Add to Cart

</button>

</div>

</div>

<!-- LATE CHECK-OUT Modal (UPDATED WITH DROPDOWN) -->

<div aria-labelledby="late-checkout-title" role="dialog" aria-modal="true" id="late-checkout-modal" class="modal">

<div class="modal-content">

<button aria-label="Close Modal" class="close-button">×</button>

<h2 id="late-checkout-title">Late Check-Out</h2>

<p>Extend your stay beyond the standard checkout time for added comfort.</p>

<h3>Pricing</h3>

<p>Late check-out available for an additional fee.</p>

<!-- NEW/UPDATED CODE: Add a dropdown for multiple late check-out products/times. -->

<label style="margin-top: 10px;">Choose a Time:</label>

<select id="late-checkout-options">

<option value="11:00 AM" data-price="0" selected>11 AM - No Charge</option>

<option value="12:00 PM" data-price="2900">12 PM - $29</option>

<option value="1:00 PM" data-price="3900">1 PM - $39</option>

<option value="2:00 PM" data-price="4900">2 PM - $49</option>

<option value="3:00 PM" data-price="5900">3 PM - $59</option>

</select>

<label style="margin-top: 10px;">Enter PIN (default "123"):</label>

<div style="display: flex; gap: 10px; margin: 10px 0;">

<input placeholder="Enter PIN" style="flex: 1;" id="late-checkout-pin" type="password">

<button style="flex: 1;" class="cta-button" onclick="confirmLateCheckOutPin()">Confirm</button>

</div>

<button style="margin-top: 10px;" class="cta-button"

onclick="handleLateCheckoutSelection(); closeModal('late-checkout-modal');">

Add to Cart

</button>

</div>

</div>

<!-- Toiletries & Essentials Modal (UPDATED WITH DROPDOWN) -->

<div aria-labelledby="toiletries-title" role="dialog" aria-modal="true" id="toiletries-modal" class="modal">

<div class="modal-content">

<button aria-label="Close Modal" class="close-button">×</button>

<h2 id="toiletries-title">Toiletries &amp; Essentials</h2>

<p>We provide complimentary toiletries and essential items to ensure your comfort.</p>

<!-- NEW/UPDATED CODE: Add a dropdown for multiple toiletries product selections. -->

<label style="margin-top: 10px;">Choose an Option:</label>

<select id="toiletries-options">

<option value="Basic Pack" data-price="0" selected>Basic Pack (No Charge)</option>

<option value="Deluxe Pack" data-price="1000">Deluxe Toiletries ($10)</option>

<option value="Premium Pack" data-price="1500">Premium Toiletries ($15)</option>

</select>

<label style="margin-top: 10px;">Enter PIN (default "123"):</label>

<div style="display: flex; gap: 10px; margin: 10px 0;">

<input placeholder="Enter PIN" style="flex: 1;" id="toiletries-pin" type="password">

<button style="flex: 1;" class="cta-button" onclick="confirmToiletriesPin()">Confirm</button>

</div>

<button style="margin-top: 10px;" class="cta-button"

onclick="handleToiletriesSelection(); closeModal('toiletries-modal');">

Add to Cart

</button>

</div>

</div>

<!-- Detergents Modal (UPDATED WITH DROPDOWN) -->

<div aria-labelledby="detergents-title" role="dialog" aria-modal="true" id="detergents-modal" class="modal">

<div class="modal-content">

<button aria-label="Close Modal" class="close-button">×</button>

<h2 id="detergents-title">Detergents</h2>

<p>We offer complimentary detergents to keep your clothes fresh and clean.</p>

<!-- NEW/UPDATED CODE: Add a dropdown for multiple detergent product selections. -->

<label style="margin-top: 10px;">Choose an Option:</label>

<select id="detergents-options">

<option value="Standard Detergent" data-price="0" selected>Standard (No Charge)</option>

<option value="Extra Pods" data-price="500">Extra Laundry Pods ($5)</option>

<option value="Full Cleaning Kit" data-price="1200">Full Cleaning Kit ($12)</option>

</select>

<label style="margin-top: 10px;">Enter PIN (default "123"):</label>

<div style="display: flex; gap: 10px; margin: 10px 0;">

<input placeholder="Enter PIN" style="flex: 1;" id="detergents-pin" type="password">

<button style="flex: 1;" class="cta-button" onclick="confirmDetergentsPin()">Confirm</button>

</div>

<button style="margin-top: 10px;" class="cta-button"

onclick="handleDetergentsSelection(); closeModal('detergents-modal');">

Add to Cart

</button>

</div>

</div>

<!-- Food & Snacks Modal (UPDATED WITH DROPDOWN) -->

<div aria-labelledby="food-snacks-title" role="dialog" aria-modal="true" id="food-snacks-modal" class="modal">

<div class="modal-content">

<button aria-label="Close Modal" class="close-button">×</button>

<h2 id="food-snacks-title">Food &amp; Snacks</h2>

<p>Enjoy complimentary treats 24/7 to satisfy your hunger and cravings.</p>

<!-- NEW/UPDATED CODE: Add a dropdown for multiple food/snack product selections. -->

<label style="margin-top: 10px;">Choose an Option:</label>

<select id="food-snacks-options">

<option value="Basic Snacks" data-price="0" selected>Basic Snacks (No Charge)</option>

<option value="Snack Bundle" data-price="800">Snack Bundle ($8)</option>

<option value="Meal Kit" data-price="2000">Meal Kit ($20)</option>

</select>

<label style="margin-top: 10px;">Enter PIN (default "123"):</label>

<div style="display: flex; gap: 10px; margin: 10px 0;">

<input placeholder="Enter PIN" style="flex: 1;" id="food-snacks-pin" type="password">

<button style="flex: 1;" class="cta-button" onclick="confirmFoodSnacksPin()">Confirm</button>

</div>

<button style="margin-top: 10px;" class="cta-button"

onclick="handleFoodSnacksSelection(); closeModal('food-snacks-modal');">

Add to Cart

</button>

</div>

</div>

<!-- Acknowledge Modal -->

<div aria-labelledby="acknowledge-modal-title" role="dialog" aria-modal="true" id="acknowledge-modal" class="modal">

<div class="modal-content">

<h2 id="acknowledge-modal-title">Please Confirm</h2>

<p>I've read and understood the information for this section.</p>

<button class="submit-button visible" id="acknowledge-button">I Acknowledge</button>

</div>

</div>

<!-- Welcome Overlay -->

<div id="welcome-overlay" class="welcome-overlay">

<div class="welcome-message">

<h2 id="welcome-text">Welcome!</h2>

<button class="dismiss-welcome">Dismiss</button>

</div>

</div>

<!-- CART MODAL -->

<div aria-labelledby="cart-modal-title" role="dialog" aria-modal="true" id="cart-modal" class="modal">

<div class="modal-content">

<button aria-label="Close Modal" class="close-button" onclick="closeModal('cart-modal')">×</button>

<h2 id="cart-modal-title">Your Cart</h2>

<div id="cart-items-container">

<ul id="cart-items"></ul>

<div id="cart-total">Total: $0.00</div>

</div>

<button id="checkout-button" class="submit-button visible">Checkout</button>

</div>

</div>

<!-- Pre-checkin checklist modal (with checkboxes) -->

<div aria-labelledby="precheckin-checklist-title" id="precheckin-checklist-modal" class="modal" aria-modal="true"

role="dialog">

<div class="modal-content">

<button aria-label="Close Modal" class="close-button" onclick="closeModal('precheckin-checklist-modal')">×</button>

<h2 id="precheckin-checklist-title">Pre-Checkin Checklist</h2>

<p>Please review the key points for a smooth stay:</p>

<ul>

<li><input type="checkbox" id="preCheckin1" onclick="enablePrecheckinButton()">

<label for="preCheckin1">Max Guests: 2 | No Visitors</label>

</li>

<li><input type="checkbox" id="preCheckin2" onclick="enablePrecheckinButton()">

<label for="preCheckin2">No Smoking / No Pets</label>

</li>

<li><input type="checkbox" id="preCheckin3" onclick="enablePrecheckinButton()">

<label for="preCheckin3">Parking: 1 spot available</label>

</li>

<li><input type="checkbox" id="preCheckin4" onclick="enablePrecheckinButton()">

<label for="preCheckin4">Keep Noise Levels Down</label>

</li>

<li><input type="checkbox" id="preCheckin5" onclick="enablePrecheckinButton()">

<label for="preCheckin5">Use Garbage Bins Properly</label>

</li>

<li><input type="checkbox" id="preCheckin6" onclick="enablePrecheckinButton()">

<label for="preCheckin6">Cleanliness is required for refund</label>

</li>

<li><input type="checkbox" id="preCheckin7" onclick="enablePrecheckinButton()">

<label for="preCheckin7">Door Code will be provided upon arrival</label>

</li>

</ul>

<p>These guidelines and more can be found in the "Pre-Check-In Information" section. We appreciate your cooperation!</p>

<button class="submit-button visible" id="precheckin-checklist-confirm" disabled onclick="acknowledgePreCheckinChecklist()">

I Acknowledge

</button>

</div>

</div>

<!-- Pre-checkout checklist modal (with checkboxes) -->

<div aria-labelledby="precheckout-checklist-title" id="precheckout-checklist-modal" class="modal" aria-modal="true"

role="dialog">

<div class="modal-content">

<button aria-label="Close Modal" class="close-button" onclick="closeModal('precheckout-checklist-modal')">×</button>

<h2 id="precheckout-checklist-title">Pre-Checkout Checklist</h2>

<p>Please review the key points for a smooth checkout:</p>

<ul>

<li><input type="checkbox" id="preCheckout1" onclick="enablePrecheckoutButton()">

<label for="preCheckout1">Remove all personal items</label>

</li>

<li><input type="checkbox" id="preCheckout2" onclick="enablePrecheckoutButton()">

<label for="preCheckout2">Dispose of all trash properly</label>

</li>

<li><input type="checkbox" id="preCheckout3" onclick="enablePrecheckoutButton()">

<label for="preCheckout3">Lock doors and windows</label>

</li>

<li><input type="checkbox" id="preCheckout4" onclick="enablePrecheckoutButton()">

<label for="preCheckout4">Turn off lights and AC</label>

</li>

<li><input type="checkbox" id="preCheckout5" onclick="enablePrecheckoutButton()">

<label for="preCheckout5">Return keys to lockbox</label>

</li>

</ul>

<p>These guidelines and more can be found in the "Pre-Check-Out Information" section. We appreciate your cooperation!</p>

<button class="submit-button visible" id="precheckout-checklist-confirm" disabled

onclick="acknowledgePreCheckoutChecklist()">

I Acknowledge

</button>

</div>

</div>

<!-- Settings PIN Entry Modal (only if not yet authenticated this session) -->

<div aria-labelledby="settings-pin-title" role="dialog" aria-modal="true" id="settings-pin-modal" class="modal">

<div class="modal-content">

<h2 id="settings-pin-title">Enter PIN to Access Settings</h2>

<form id="settings-pin-form">

<label for="settings-pin-input">PIN:</label>

<input aria-label="Settings PIN" required id="settings-pin-input" type="password">

<span id="settings-pin-error" class="error-message">Incorrect PIN. Please try again.</span>

<button class="submit-button visible" type="submit">Submit</button>

</form>

</div>

</div>

<!-- Main Settings Modal (for sub-feature triggers) -->

<div aria-labelledby="settings-modal-title" role="dialog" aria-modal="true" id="settings-modal" class="modal">

<div class="modal-content">

<button aria-label="Close Modal" class="close-button" onclick="closeModal('settings-modal')">×</button>

<h2 id="settings-modal-title">App Settings</h2>

<div class="settings-features">

<button class="cta-button" type="button" onclick="openModal('calendar-sync-modal')">Calendar Sync</button>

<button class="cta-button" type="button" onclick="openModal('stripe-settings-modal')">Stripe Settings</button>

<button class="cta-button" type="button" onclick="openModal('edit-precheckin-checklist-modal')">

Edit Pre-Checkin Checklist

</button>

<button class="cta-button" type="button" onclick="openModal('edit-precheckout-checklist-modal')">

Edit Pre-Checkout Checklist

</button>

<button class="cta-button" type="button" onclick="openModal('reset-storage-modal')">Reset Storage</button>

<button class="cta-button" type="button" onclick="openModal('change-image-modal')">Change Image</button>

<button class="cta-button" type="button" onclick="openModal('notifications-modal')">Customize Notifications</button>

<button class="cta-button" type="button" onclick="openModal('edit-precheckinout-modal')">

Edit Pre-Check-In/Out Info

</button>

<button class="cta-button" type="button" onclick="openModal('change-language-modal')">Change Language</button>

<button class="cta-button" type="button" onclick="openModal('change-wifi-door-modal')">Change WiFi &amp; Door Code</button>

<button class="cta-button" type="button" onclick="openModal('change-pkpass-modal')">Change PKPass Link</button>

<!-- NEW/UPDATED CODE: Manage the 'Enhance Your Stay' inventory from here -->

<button class="cta-button" type="button" onclick="openModal('edit-enhancements-inventory-modal')">

Edit Enhancements Inventory

</button>

</div>

</div>

</div>

<!-- SUB-MODAL: Reset Storage Confirmation -->

<div aria-labelledby="reset-storage-title" role="dialog" aria-modal="true" id="reset-storage-modal" class="modal">

<div class="modal-content">

<button aria-label="Close Modal" class="close-button" onclick="closeModal('reset-storage-modal')">×</button>

<h2 id="reset-storage-title">Confirm Reset Storage</h2>

<p>Are you sure you want to reset all stored data to default settings? This action cannot be undone.</p>

<button style="background-color: var(--error-color);" class="submit-button visible" id="confirm-reset-storage">

Yes, Reset

</button>

<button class="submit-button visible" id="cancel-reset-storage">Cancel</button>

</div>

</div>

<!-- SUB-MODAL: Change Image -->

<div aria-labelledby="change-image-title" role="dialog" aria-modal="true" id="change-image-modal" class="modal">

<div class="modal-content">

<button aria-label="Close Modal" class="close-button" onclick="closeModal('change-image-modal')">×</button>

<h2 id="change-image-title">Change Image</h2>

<label for="upload-image">Upload a New Image</label>

<input accept="image/\*" id="upload-image" type="file">

<div style="margin-top: 10px;" id="image-preview">

<img style="max-width: 100%; height: auto; display: none; border-radius: var(--border-radius);"

alt="Image Preview" src="#">

</div>

</div>

</div>

<!-- SUB-MODAL: Notifications (UPDATED FOR RECURRING SETTINGS) -->

<div aria-labelledby="notifications-modal-title" role="dialog" aria-modal="true" id="notifications-modal"

class="modal">

<div class="modal-content">

<button aria-label="Close Modal" class="close-button" onclick="closeModal('notifications-modal')">×</button>

<h2 id="notifications-modal-title">Customize Notifications</h2>

<p>Adjust the notification messages and schedule times as needed.</p>

<!-- Single-time reminders -->

<div class="notification-group">

<label for="checkin-reminder">Check-In Reminder Message</label>

<textarea id="checkin-reminder" placeholder="Enter your check-in reminder message..." rows="3">Reminder: Your check-in is soon. Please let us know if you need anything before arrival. We’re here to help!

</textarea>

</div>

<div class="notification-group">

<label for="checkin-reminder-time">Check-In Reminder Time (HH:MM, 24h)</label>

<input id="checkin-reminder-time" type="time" value="14:00">

</div>

<div class="notification-group">

<label for="checkout-reminder">Check-Out Reminder Message</label>

<textarea id="checkout-reminder" placeholder="Enter your check-out reminder message..." rows="3">Reminder: Your check-out is scheduled soon. Please leave the property in good condition. Safe travels!

</textarea>

</div>

<div class="notification-group">

<label for="checkout-reminder-time">Check-Out Reminder Time (HH:MM, 24h)</label>

<input id="checkout-reminder-time" type="time" value="10:00">

</div>

<!-- NEW: Recurring Notification Settings -->

<hr>

<h3>Recurring Notifications (e.g., every 3 hours)</h3>

<div class="notification-group">

<label for="recurring-interval">Interval (in hours)</label>

<input type="number" id="recurring-interval" value="3" min="1">

</div>

<div class="notification-group">

<label for="recurring-messages">Recurring Messages (one per line)</label>

<textarea id="recurring-messages" rows="5">

Time for a quick check! Everything going smoothly?

Remember to explore local amenities—see our “Enhance Your Stay” section.

Any questions? Feel free to reach out via the app or call us.

</textarea>

</div>

<button id="send-test-notification" class="cta-button" type="button">Send Test Notification</button>

<button class="submit-button visible" onclick="saveNotificationSettings()">Save Notification Settings</button>

</div>

</div>

<!-- SUB-MODAL: Edit Pre Check-In/Out Info -->

<div aria-labelledby="edit-precheckinout-title" role="dialog" aria-modal="true" id="edit-precheckinout-modal"

class="modal">

<div class="modal-content">

<button aria-label="Close Modal" class="close-button" onclick="closeModal('edit-precheckinout-modal')">×</button>

<h2 id="edit-precheckinout-title">Edit Pre Check-In/Out Info</h2>

<p>Make quick edits to your pre-checkin and pre-checkout details. Keep them short for clarity.</p>

<label for="before-checkin-text">Before Check-In Info</label>

<textarea id="before-checkin-text" rows="6" maxlength="600">Max Guests: 2 | No Visitors

No Smoking / No Pets

Parking: 1 spot available

Respect Neighbors: Keep noise levels down.

Security: Cameras active on premises.

Door Code: 8325

Cleanliness: Please maintain cleanliness for a refund.

</textarea>

<label for="house-rules-text">House Rules</label>

<textarea id="house-rules-text" rows="5" maxlength="400">Check-In: After 3:00 PM

Check-Out: Before 11:00 AM

No Parties: Strictly no events.

Quiet Hours: 10:00 PM - 8:00 AM

Garbage: Dispose of waste properly.

</textarea>

<label for="checkout-instructions-text">Check-Out Instructions</label>

<textarea id="checkout-instructions-text" rows="5" maxlength="400">Remove Waste: Use provided bins

Dry Towels: Hang towels to dry

Secure Premises: Lock doors &amp; windows

Laundry: Use facilities responsibly

Cleanliness: Leave space clean for refund

</textarea>

<label for="additional-checkout-text">Additional Checkout Steps</label>

<textarea id="additional-checkout-text" rows="5" maxlength="400">Return Keys: Place keys in lockbox

Thermostat: Set to 72°F

Lights &amp; AC: Turn off before leaving

Windows: Ensure all are closed

Leave a Review: We appreciate your feedback!

</textarea>

<button class="submit-button visible" onclick="savePreCheckinOutInfo()">Save</button>

</div>

</div>

<!-- SUB-MODAL: Change Language -->

<div aria-labelledby="change-language-title" role="dialog" aria-modal="true" id="change-language-modal" class="modal">

<div class="modal-content">

<button aria-label="Close Modal" class="close-button" onclick="closeModal('change-language-modal')">×</button>

<h2 id="change-language-title">Change Language</h2>

<label for="language-select">Preferred Language</label>

<select id="language-select">

<option value="en" selected>English</option>

<option value="es">Español</option>

<option value="fr">Français</option>

<option value="de">Deutsch</option>

</select>

<button class="submit-button visible" onclick="saveLanguage()">Save Language</button>

</div>

</div>

<!-- SUB-MODAL: Calendar Sync -->

<div aria-labelledby="calendar-sync-title" role="dialog" aria-modal="true" id="calendar-sync-modal" class="modal">

<div class="modal-content">

<button aria-label="Close Modal" class="close-button" onclick="closeModal('calendar-sync-modal')">×</button>

<h2 id="calendar-sync-title">Calendar Sync</h2>

<p>Sync your calendar with the following iCal URL.</p>

<label for="calendar-sync-input">iCal Link</label>

<input type="text" id="calendar-sync-input"

value="https://www.airbnb.com/calendar/ical/1217382793937904124.ics?s=6aff49501fe305972a8d58a019d7ea89&locale=en">

<button class="submit-button visible" onclick="saveCalendarSync()">Save Calendar Sync</button>

</div>

</div>

<!-- SUB-MODAL: Stripe Settings -->

<div aria-labelledby="stripe-settings-title" role="dialog" aria-modal="true" id="stripe-settings-modal" class="modal">

<div class="modal-content">

<button aria-label="Close Modal" class="close-button" onclick="closeModal('stripe-settings-modal')">×</button>

<h2 id="stripe-settings-title">Stripe Settings</h2>

<p>Configure your Stripe keys here.</p>

<label for="stripe-publishable-key">Publishable Key</label>

<input type="text" id="stripe-publishable-key" placeholder="pk\_test\_12345">

<label for="stripe-secret-key">Secret Key</label>

<input type="text" id="stripe-secret-key" placeholder="sk\_test\_12345">

<button class="submit-button visible" onclick="saveStripeSettings()">Save Stripe Settings</button>

</div>

</div>

<!-- SUB-MODAL: Edit Pre-Checkin Checklist Items -->

<div aria-labelledby="edit-precheckin-checklist-title" role="dialog" aria-modal="true" id="edit-precheckin-checklist-modal"

class="modal">

<div class="modal-content">

<button aria-label="Close Modal" class="close-button" onclick="closeModal('edit-precheckin-checklist-modal')">×</button>

<h2 id="edit-precheckin-checklist-title">Edit Pre-Checkin Checklist</h2>

<p>Edit the items you want guests to confirm before check-in.</p>

<label for="precheckin-items">Pre-Checkin Items (one per line)</label>

<textarea id="precheckin-items" rows="7">Max Guests: 2 | No Visitors

No Smoking / No Pets

Parking: 1 spot available

Keep Noise Levels Down

Use Garbage Bins Properly

Cleanliness is required for refund

Door Code will be provided upon arrival

</textarea>

<button class="submit-button visible" onclick="savePrecheckinItems()">Save Pre-Checkin Items</button>

</div>

</div>

<!-- SUB-MODAL: Edit Pre-Checkout Checklist Items -->

<div aria-labelledby="edit-precheckout-checklist-title" role="dialog" aria-modal="true"

id="edit-precheckout-checklist-modal" class="modal">

<div class="modal-content">

<button aria-label="Close Modal" class="close-button" onclick="closeModal('edit-precheckout-checklist-modal')">×</button>

<h2 id="edit-precheckout-checklist-title">Edit Pre-Checkout Checklist</h2>

<p>Edit the items you want guests to confirm before check-out.</p>

<label for="precheckout-items">Pre-Checkout Items (one per line)</label>

<textarea id="precheckout-items" rows="6">Remove all personal items

Dispose of all trash properly

Lock doors and windows

Turn off lights and AC

Return keys to lockbox

</textarea>

<button class="submit-button visible" onclick="savePrecheckoutItems()">Save Pre-Checkout Items</button>

</div>

</div>

<!-- SUB-MODAL: Change WiFi & Door Code (Ensure door code is prefilled) -->

<div aria-labelledby="change-wifi-door-title" role="dialog" aria-modal="true" id="change-wifi-door-modal" class="modal">

<div class="modal-content">

<button aria-label="Close Modal" class="close-button" onclick="closeModal('change-wifi-door-modal')">×</button>

<h2 id="change-wifi-door-title">Change WiFi &amp; Door Code</h2>

<form id="change-wifi-door-form">

<!-- WiFi Name -->

<div style="margin-bottom: 15px;">

<label for="new-wifi-name">New WiFi Name</label>

<input placeholder="e.g., Vista Charm" id="new-wifi-name" type="text">

</div>

<!-- WiFi Password -->

<div style="margin-bottom: 15px;">

<label for="new-wifi-password">New WiFi Password</label>

<input placeholder="e.g., Vista8325" id="new-wifi-password" type="text">

</div>

<!-- Door Code -->

<div style="margin-bottom: 15px;">

<label for="new-doorcode">New Door Code (4+ digits)</label>

<input placeholder="Enter new door code" id="new-doorcode" type="password">

</div>

<button class="submit-button visible" type="submit">Save Changes</button>

</form>

</div>

</div>

<!-- SUB-MODAL: Change PKPass Link (Ensure it’s prefilled) -->

<div aria-labelledby="change-pkpass-title" role="dialog" aria-modal="true" id="change-pkpass-modal" class="modal">

<div class="modal-content">

<button aria-label="Close Modal" class="close-button" onclick="closeModal('change-pkpass-modal')">×</button>

<h2 id="change-pkpass-title">Change PKPass Link</h2>

<label for="pkpass-link">New PKPass URL</label>

<!-- NEW/UPDATED CODE: Use 'value' here to prefill the current link via JS on open -->

<input id="pkpass-link" type="text">

<button class="submit-button visible" type="button" onclick="savePkpassLink()">Save PKPass Link</button>

</div>

</div>

<!-- NEW/UPDATED CODE: Edit Enhancements Inventory (Sub-Modal) -->

<div aria-labelledby="edit-enhancements-inventory-title" role="dialog" aria-modal="true"

id="edit-enhancements-inventory-modal" class="modal">

<div class="modal-content">

<button aria-label="Close Modal" class="close-button" onclick="closeModal('edit-enhancements-inventory-modal')">×</button>

<h2 id="edit-enhancements-inventory-title">Edit Enhancements Inventory</h2>

<p>Manage the available items, times, and pricing for each enhancement below. Adjust as needed.</p>

<p>(In a real app, you'd store these in localStorage or a database. This is just a placeholder.)</p>

<label for="early-checkin-edit">Early Check-In Options (time - price in cents, one per line)</label>

<textarea id="early-checkin-edit" rows="4">

12:00 PM - 4900

1:00 PM - 3900

2:00 PM - 2900

3:00 PM - 0

</textarea>

<label for="late-checkout-edit">Late Check-Out Options (time - price, one per line)</label>

<textarea id="late-checkout-edit" rows="4">

11:00 AM - 0

12:00 PM - 2900

1:00 PM - 3900

2:00 PM - 4900

3:00 PM - 5900

</textarea>

<label for="toiletries-edit">Toiletries Options (description - price)</label>

<textarea id="toiletries-edit" rows="4">

Basic Pack - 0

Deluxe Pack - 1000

Premium Pack - 1500

</textarea>

<label for="detergents-edit">Detergents Options (description - price)</label>

<textarea id="detergents-edit" rows="4">

Standard Detergent - 0

Extra Pods - 500

Full Cleaning Kit - 1200

</textarea>

<label for="food-snacks-edit">Food & Snacks Options (description - price)</label>

<textarea id="food-snacks-edit" rows="4">

Basic Snacks - 0

Snack Bundle - 800

Meal Kit - 2000

</textarea>

<button class="submit-button visible" onclick="saveEnhancementsInventory()">Save Enhancements</button>

</div>

</div>

<!-- Stripe JS (for Payment) -->

<script src="https://js.stripe.com/v3/"></script>

<!-- Litepicker JS for Date Range Selection -->

<script src="https://cdn.jsdelivr.net/npm/litepicker/dist/litepicker.js"></script>

<!--

Service Worker Code Inlined Below.

Normally, you'd create "service-worker.js" in the site root and reference it:

navigator.serviceWorker.register('/service-worker.js')

-->

<script>

// A simple example service worker script that handles caching & offline fallback.

// For a production app, expand the caching logic to handle versioning, dynamic caching, etc.

const CACHE\_NAME = "vista-charm-pwa-v1";

const OFFLINE\_URL = "/offline-fallback.html"; // Provide your offline HTML page if you have one.

self.addEventListener("install", (event) => {

// Pre-cache critical assets

event.waitUntil(

caches.open(CACHE\_NAME).then((cache) => {

return cache.addAll([

// You can add your critical routes, images, CSS, etc. here

]);

})

);

self.skipWaiting();

});

self.addEventListener("activate", (event) => {

// Cleanup old caches if needed

event.waitUntil(

caches.keys().then((cacheNames) => {

return Promise.all(

cacheNames.map((cache) => {

if (cache !== CACHE\_NAME) {

return caches.delete(cache);

}

})

);

})

);

self.clients.claim();

});

self.addEventListener("fetch", (event) => {

// Basic strategy: try network first, fallback to cache, then offline page

event.respondWith(

fetch(event.request)

.then((response) => {

// Optional: add the response to the cache for future offline use

const clone = response.clone();

caches.open(CACHE\_NAME).then((cache) => cache.put(event.request, clone));

return response;

})

.catch(() => {

return caches.match(event.request).then((cached) => {

return cached || caches.match(OFFLINE\_URL);

});

})

);

});

// Example push event listener for background push notifications

self.addEventListener("push", (event) => {

if (event.data) {

const data = event.data.json();

const title = data.title || "Vista Charm Notification";

const options = data.options || {};

event.waitUntil(self.registration.showNotification(title, options));

}

});

// Listen for messages from the main thread (e.g., show a notification)

self.addEventListener("message", (event) => {

if (event.data && event.data.type === "SHOW\_NOTIFICATION") {

const payload = event.data.payload;

event.waitUntil(

self.registration.showNotification(payload.title, payload.options)

);

}

});

</script>

<!-- Main Script (Registers SW, Push, Timers, etc.) -->

<script>

// Register Service Worker from the inlined code above

if ("serviceWorker" in navigator) {

window.addEventListener("load", () => {

navigator.serviceWorker

.register(window.location.href, { scope: "./" })

.then((registration) => {

console.log("Service Worker registered with scope:", registration.scope);

})

.catch((error) => {

console.error("Service Worker registration failed:", error);

});

});

}

// Request Notification Permissions

document.addEventListener("DOMContentLoaded", () => {

if ("Notification" in window && navigator.serviceWorker) {

if (Notification.permission === "default") {

Notification.requestPermission().then((permission) => {

console.log("Notification permission:", permission);

});

}

}

});

// Send Notification Function

function sendNotification(title, body) {

if ("serviceWorker" in navigator && "Notification" in window) {

navigator.serviceWorker.ready.then((registration) => {

if (registration.active) {

registration.active.postMessage({

type: "SHOW\_NOTIFICATION",

payload: {

title: title,

options: {

body: body,

icon: "https://cdn.shopify.com/s/files/1/0901/5171/5111/files/IMG\_1960.png",

badge: "https://cdn.shopify.com/s/files/1/0901/5171/5111/files/IMG\_1960.png",

data: {

url: window.location.href,

},

vibrate: [100, 50, 100],

actions: [{ action: "open\_url", title: "Open" }],

},

},

});

}

});

} else {

console.log("Notifications are not supported in this browser.");

}

}

</script>

<!-- Theme Toggle, Modal Controls, and the rest of the app’s JS -->

<script>

document.addEventListener("DOMContentLoaded", () => {

const themeToggleButton = document.getElementById("theme-toggle-button");

const themeIcon = document.getElementById("theme-icon");

// Auto dark mode based on time if no user preference

const savedTheme = localStorage.getItem("theme");

if (!savedTheme) {

const hour = new Date().getHours();

if (hour >= 20 || hour < 7) {

document.body.setAttribute("data-theme", "dark");

themeIcon.classList.remove("fa-moon");

themeIcon.classList.add("fa-sun");

localStorage.setItem("theme", "dark");

}

} else {

if (savedTheme === "dark") {

document.body.setAttribute("data-theme", "dark");

themeIcon.classList.remove("fa-moon");

themeIcon.classList.add("fa-sun");

}

}

themeToggleButton.addEventListener("click", () => {

const isDark = document.body.getAttribute("data-theme") === "dark";

if (isDark) {

document.body.setAttribute("data-theme", "light");

themeIcon.classList.remove("fa-sun");

themeIcon.classList.add("fa-moon");

localStorage.setItem("theme", "light");

} else {

document.body.setAttribute("data-theme", "dark");

themeIcon.classList.remove("fa-moon");

themeIcon.classList.add("fa-sun");

localStorage.setItem("theme", "dark");

}

});

/\* ----------------------------------------

PRE-CHECKIN BUTTON -> Checklist

---------------------------------------- \*/

const togglePrecheckinButton = document.getElementById("toggle-precheckin-button");

const precheckinCards = document.getElementById("precheckin-cards");

togglePrecheckinButton.addEventListener("click", () => {

const preCheckinChecklistShown = localStorage.getItem("preCheckinChecklistShown") === "true";

if (!preCheckinChecklistShown) {

// Open the pre-checkin checklist modal

openModal("precheckin-checklist-modal");

} else {

// Checklist already acknowledged, just toggle the cards

if (precheckinCards.style.display === "none" || precheckinCards.style.display === "") {

precheckinCards.style.display = "flex";

togglePrecheckinButton.textContent = "Hide Pre-Check-In Info";

} else {

precheckinCards.style.display = "none";

togglePrecheckinButton.textContent = "Show Pre-Check-In Info";

}

}

});

/\* ----------------------------------------

PRE-CHECKOUT BUTTON -> Checklist

---------------------------------------- \*/

const togglePrecheckoutButton = document.getElementById("toggle-precheckout-button");

const precheckoutCards = document.getElementById("precheckout-cards");

togglePrecheckoutButton.addEventListener("click", () => {

const preCheckoutChecklistShown = localStorage.getItem("preCheckoutChecklistShown") === "true";

if (!preCheckoutChecklistShown) {

// Open the pre-checkout checklist modal

openModal("precheckout-checklist-modal");

} else {

// Checklist already acknowledged, just toggle the cards

if (precheckoutCards.style.display === "none" || precheckoutCards.style.display === "") {

precheckoutCards.style.display = "flex";

togglePrecheckoutButton.textContent = "Hide Pre-Check-Out Info";

} else {

precheckoutCards.style.display = "none";

togglePrecheckoutButton.textContent = "Show Pre-Check-Out Info";

}

}

});

/\* ----------------------------------------

Settings Button -> Open PIN Modal

---------------------------------------- \*/

const settingsButton = document.getElementById("settings-button");

settingsButton.addEventListener("click", () => {

// If user is already authenticated this session, open settings directly

if (localStorage.getItem("settingsAuthenticated") === "true") {

openModal("settings-modal");

} else {

openModal("settings-pin-modal");

}

});

/\* ----------------------------------------

Settings PIN Form Submission

---------------------------------------- \*/

const settingsPinForm = document.getElementById("settings-pin-form");

settingsPinForm.addEventListener("submit", (e) => {

e.preventDefault();

const enteredPin = document.getElementById("settings-pin-input").value.trim();

const settingsPinError = document.getElementById("settings-pin-error");

// Default PIN = "123"

const storedPin = localStorage.getItem("userPIN") || "123";

if (enteredPin === storedPin) {

closeModal("settings-pin-modal");

openModal("settings-modal");

settingsPinForm.reset();

settingsPinError.classList.remove("visible");

localStorage.setItem("settingsAuthenticated", "true");

} else {

settingsPinError.classList.add("visible");

}

});

});

// Modal control functions

function openModal(modalId) {

document.getElementById(modalId).classList.add("active");

// Prefill PKPass link in the input if opening 'change-pkpass-modal'

if (modalId === "change-pkpass-modal") {

const input = document.getElementById("pkpass-link");

const storedPkpassLink =

localStorage.getItem("pkpassLink") ||

"https://cdn.shopify.com/s/files/1/0671/8111/7609/files/4A5FA684-1F60-4706-B0A3-0D3D560DAACD.pkpass?v=1729416167";

input.value = storedPkpassLink;

}

// Prefill door code if opening 'change-wifi-door-modal'

if (modalId === "change-wifi-door-modal") {

const doorInput = document.getElementById("new-doorcode");

const storedCode = localStorage.getItem("doorCode") || "8325";

doorInput.value = storedCode;

}

}

function closeModal(modalId) {

document.getElementById(modalId).classList.remove("active");

}

// Acknowledge pre-checkin checklist

function acknowledgePreCheckinChecklist() {

localStorage.setItem("preCheckinChecklistShown", "true");

closeModal("precheckin-checklist-modal");

// Immediately show the precheckin cards if the user just acknowledged

const precheckinCards = document.getElementById("precheckin-cards");

const togglePrecheckinButton = document.getElementById("toggle-precheckin-button");

precheckinCards.style.display = "flex";

togglePrecheckinButton.textContent = "Hide Pre-Check-In Info";

}

function enablePrecheckinButton() {

let allChecked = true;

for (let i = 1; i <= 7; i++) {

if (!document.getElementById("preCheckin" + i).checked) {

allChecked = false;

break;

}

}

document.getElementById("precheckin-checklist-confirm").disabled = !allChecked;

}

// Acknowledge pre-checkout checklist

function acknowledgePreCheckoutChecklist() {

localStorage.setItem("preCheckoutChecklistShown", "true");

closeModal("precheckout-checklist-modal");

// Immediately show the precheckout cards

const precheckoutCards = document.getElementById("precheckout-cards");

const togglePrecheckoutButton = document.getElementById("toggle-precheckout-button");

precheckoutCards.style.display = "flex";

togglePrecheckoutButton.textContent = "Hide Pre-Check-Out Info";

}

function enablePrecheckoutButton() {

let allChecked = true;

for (let i = 1; i <= 5; i++) {

if (!document.getElementById("preCheckout" + i).checked) {

allChecked = false;

break;

}

}

document.getElementById("precheckout-checklist-confirm").disabled = !allChecked;

}

// Acknowledge modal

const acknowledgeButton = document.getElementById("acknowledge-button");

if (acknowledgeButton) {

acknowledgeButton.addEventListener("click", () => {

closeModal("acknowledge-modal");

});

}

// Close modal with escape key

document.addEventListener("keydown", function (event) {

const activeModal = document.querySelector(".modal.active");

if (activeModal && event.key === "Escape") {

activeModal.classList.remove("active");

}

});

</script>

<script>

// Save Notification Settings (UPDATED FOR RECURRING)

function saveNotificationSettings() {

const checkinReminder = document.getElementById("checkin-reminder").value.trim();

const checkinReminderTime = document.getElementById("checkin-reminder-time").value;

const checkoutReminder = document.getElementById("checkout-reminder").value.trim();

const checkoutReminderTime = document.getElementById("checkout-reminder-time").value;

// Recurring notifications

const recurringIntervalInput = document.getElementById("recurring-interval");

const recurringInterval = parseInt(recurringIntervalInput.value, 10) || 3; // Default to 3 if blank

const recurringMessages = document.getElementById("recurring-messages").value.trim();

localStorage.setItem("checkinReminder", checkinReminder);

localStorage.setItem("checkinReminderTime", checkinReminderTime);

localStorage.setItem("checkoutReminder", checkoutReminder);

localStorage.setItem("checkoutReminderTime", checkoutReminderTime);

// Save recurring interval & messages

localStorage.setItem("recurringIntervalHours", recurringInterval);

localStorage.setItem("recurringMessages", recurringMessages);

alert("Notification settings saved.");

closeModal("notifications-modal");

}

// Save Pre-Checkin/Checkout Info

function savePreCheckinOutInfo() {

const beforeCheckinText = document.getElementById("before-checkin-text").value;

const houseRulesText = document.getElementById("house-rules-text").value;

const checkoutInstructionsText = document.getElementById("checkout-instructions-text").value;

const additionalCheckoutText = document.getElementById("additional-checkout-text").value;

localStorage.setItem("beforeCheckinText", beforeCheckinText);

localStorage.setItem("houseRulesText", houseRulesText);

localStorage.setItem("checkoutInstructionsText", checkoutInstructionsText);

localStorage.setItem("additionalCheckoutText", additionalCheckoutText);

alert("Pre Check-In/Out info saved.");

closeModal("edit-precheckinout-modal");

}

// Save Language

function saveLanguage() {

const languageSelect = document.getElementById("language-select");

localStorage.setItem("preferredLanguage", languageSelect.value);

alert(`Language changed to: ${languageSelect.value}`);

closeModal("change-language-modal");

}

// Save Calendar Sync

function saveCalendarSync() {

const calendarLink = document.getElementById("calendar-sync-input").value.trim();

localStorage.setItem("calendarSyncLink", calendarLink);

alert("Calendar Sync URL saved.");

closeModal("calendar-sync-modal");

}

// Save Stripe Settings

function saveStripeSettings() {

const publishableKey = document.getElementById("stripe-publishable-key").value.trim();

const secretKey = document.getElementById("stripe-secret-key").value.trim();

localStorage.setItem("stripePublishableKey", publishableKey);

localStorage.setItem("stripeSecretKey", secretKey);

alert("Stripe settings saved.");

closeModal("stripe-settings-modal");

}

// Save Precheckin Items

function savePrecheckinItems() {

const items = document.getElementById("precheckin-items").value;

localStorage.setItem("precheckinItems", items);

alert("Pre-checkin checklist updated.");

closeModal("edit-precheckin-checklist-modal");

}

// Save Precheckout Items

function savePrecheckoutItems() {

const items = document.getElementById("precheckout-items").value;

localStorage.setItem("precheckoutItems", items);

alert("Pre-checkout checklist updated.");

closeModal("edit-precheckout-checklist-modal");

}

</script>

<script>

// Reset Storage

const confirmResetStorage = document.getElementById("confirm-reset-storage");

const cancelResetStorage = document.getElementById("cancel-reset-storage");

if (confirmResetStorage) {

confirmResetStorage.addEventListener("click", () => {

// Reset all stored data

localStorage.clear();

alert("All stored data has been reset to default settings.");

location.reload();

});

}

if (cancelResetStorage) {

cancelResetStorage.addEventListener("click", () => {

closeModal("reset-storage-modal");

});

}

// Send Test Notification

const sendTestNotificationButton = document.getElementById("send-test-notification");

if (sendTestNotificationButton) {

sendTestNotificationButton.addEventListener("click", () => {

sendNotification("Test Notification", "This is a test notification to verify your settings.");

alert("Test notification sent.");

});

}

// Change Image

const uploadImageInput = document.getElementById("upload-image");

const imagePreview = document.getElementById("image-preview").querySelector("img");

const mainThemeImage = document.getElementById("main-theme-image");

if (uploadImageInput) {

uploadImageInput.addEventListener("change", (e) => {

const file = e.target.files[0];

if (file) {

const reader = new FileReader();

reader.onload = function (event) {

imagePreview.src = event.target.result;

imagePreview.style.display = "block";

mainThemeImage.src = event.target.result;

localStorage.setItem("themeImage", event.target.result);

};

reader.readAsDataURL(file);

} else {

imagePreview.src = "#";

imagePreview.style.display = "none";

}

});

}

</script>

<script>

// Handle Save WiFi & Door Code

const changeWifiDoorForm = document.getElementById("change-wifi-door-form");

if (changeWifiDoorForm) {

changeWifiDoorForm.addEventListener("submit", (e) => {

e.preventDefault();

const newWifiName = document.getElementById("new-wifi-name").value.trim();

const newWifiPassword = document.getElementById("new-wifi-password").value.trim();

const newDoorcode = document.getElementById("new-doorcode").value.trim();

if (newWifiName) {

localStorage.setItem("wifiName", newWifiName);

}

if (newWifiPassword) {

localStorage.setItem("wifiPassword", newWifiPassword);

}

if (/^\d{4,}$/.test(newDoorcode)) {

localStorage.setItem("doorCode", newDoorcode);

} else if (newDoorcode) {

alert("Door code must be at least 4 digits. Try again.");

return;

}

alert("WiFi & Door Code updated successfully!");

closeModal("change-wifi-door-modal");

changeWifiDoorForm.reset();

updateWifiDoorDisplay();

});

}

function updateWifiDoorDisplay() {

const wifiNameField = document.getElementById("wifi-name-field");

const wifiPasswordField = document.getElementById("wifi-password-field");

const doorcodeField = document.getElementById("doorcode-field");

const wifiName = localStorage.getItem("wifiName") || "Vista Charm";

const wifiPassword = localStorage.getItem("wifiPassword") || "Vista8325";

const doorCode = localStorage.getItem("doorCode") || "8325";

if (wifiNameField) wifiNameField.textContent = wifiName;

if (wifiPasswordField) wifiPasswordField.textContent = wifiPassword;

if (doorcodeField) doorcodeField.textContent = doorCode;

}

document.addEventListener("DOMContentLoaded", updateWifiDoorDisplay);

// Handle Save PKPass Link

function savePkpassLink() {

const pkpassInput = document.getElementById("pkpass-link");

const newLink = pkpassInput.value.trim();

if (!newLink) {

alert("Please enter a valid PKPass link.");

return;

}

localStorage.setItem("pkpassLink", newLink);

alert("PKPass link updated successfully!");

closeModal("change-pkpass-modal");

}

</script>

<script>

// Stripe initialization

const stripeKey = localStorage.getItem("stripePublishableKey") || "pk\_test\_12345";

const stripe = Stripe(stripeKey);

// Global cart array

let cart = [];

// Cart references

const cartButton = document.getElementById("cart-button");

const cartModal = document.getElementById("cart-modal");

const cartItemsEl = document.getElementById("cart-items");

const cartTotalEl = document.getElementById("cart-total");

const checkoutButton = document.getElementById("checkout-button");

// Open/close cart modal

if (cartButton && cartModal) {

cartButton.addEventListener("click", () => {

cartModal.classList.add("active");

});

}

// Add item to cart

function addToCart(name, price) {

const item = { name, price };

cart.push(item);

updateCartUI();

alert(`${name} has been added to your cart.`);

}

function updateCartUI() {

if (!cartItemsEl || !cartTotalEl) return;

cartItemsEl.innerHTML = "";

let total = 0;

cart.forEach((item) => {

total += item.price;

const li = document.createElement("li");

li.innerHTML = `

<span>${item.name}</span>

<span>$${(item.price / 100).toFixed(2)}</span>

`;

cartItemsEl.appendChild(li);

});

cartTotalEl.textContent = "Total: $" + (total / 100).toFixed(2);

}

// Checkout with Stripe

if (checkoutButton) {

checkoutButton.addEventListener("click", () => {

const lineItems = cart.map((item) => ({

price\_data: {

currency: "usd",

product\_data: { name: item.name },

unit\_amount: item.price,

},

quantity: 1,

}));

// For demo: a fake endpoint. Replace with your real server route.

fetch("/create-checkout-session", {

method: "POST",

headers: { "Content-Type": "application/json" },

body: JSON.stringify({ items: lineItems }),

})

.then((res) => res.json())

.then((data) => {

if (data.url) {

window.location = data.url;

} else if (data.sessionId) {

stripe.redirectToCheckout({ sessionId: data.sessionId });

} else {

alert("Unable to process checkout. Please try again later.");

}

})

.catch((err) => {

console.error("Checkout error:", err);

alert("Checkout error: " + err.message);

});

});

}

// Timers for Check-In / Check-Out

let selectedCheckinDate = null;

let selectedCheckoutDate = null;

let userSelectedCheckinTime = "3:00 PM";

let userSelectedCheckoutTime = "11:00 AM";

const generatePassbookButton = document.getElementById("generate-passbook-button");

const upgradeEarlyButton = document.getElementById("upgrade-early");

const upgradeLateButton = document.getElementById("upgrade-late");

const upgradeEarlyGroup = document.getElementById("upgrade-early-option");

const upgradeLateGroup = document.getElementById("upgrade-late-option");

const guestNameInput = document.getElementById("guest-name");

let isCheckinTimeSelected = false;

let isCheckoutTimeSelected = false;

let isDatesSelected = false;

let timerInterval = null;

if (guestNameInput && localStorage.getItem("guestName")) {

guestNameInput.value = localStorage.getItem("guestName");

}

if (guestNameInput) {

guestNameInput.addEventListener("input", () => {

localStorage.setItem("guestName", guestNameInput.value);

});

}

function updateTimers() {

const now = new Date();

let checkinRemaining = "--:--:--";

let checkoutRemaining = "--:--:--";

if (selectedCheckinDate && userSelectedCheckinTime) {

const checkinDateTime = new Date(formatDate(selectedCheckinDate) + " " + userSelectedCheckinTime);

const diffCheckin = checkinDateTime - now;

if (diffCheckin > 0) {

checkinRemaining = formatTime(diffCheckin);

} else {

checkinRemaining = "00:00:00";

}

}

if (selectedCheckoutDate && userSelectedCheckoutTime) {

const checkoutDateTime = new Date(formatDate(selectedCheckoutDate) + " " + userSelectedCheckoutTime);

const diffCheckout = checkoutDateTime - now;

if (diffCheckout > 0) {

checkoutRemaining = formatTime(diffCheckout);

} else {

checkoutRemaining = "00:00:00";

}

}

if (document.getElementById("checkin-timer")) {

document.getElementById("checkin-timer").textContent = checkinRemaining;

}

if (document.getElementById("checkout-timer")) {

document.getElementById("checkout-timer").textContent = checkoutRemaining;

}

}

function formatTime(ms) {

const totalSeconds = Math.floor(ms / 1000);

const hours = String(Math.floor(totalSeconds / 3600)).padStart(2, "0");

const minutes = String(Math.floor((totalSeconds % 3600) / 60)).padStart(2, "0");

const seconds = String(totalSeconds % 60).padStart(2, "0");

return `${hours}:${minutes}:${seconds}`;

}

function startTimers() {

if (timerInterval) clearInterval(timerInterval);

timerInterval = setInterval(updateTimers, 1000);

updateTimers();

}

// FAQ Accordion

const faqButtons = document.querySelectorAll(".faq-item button");

faqButtons.forEach((button) => {

button.addEventListener("click", () => {

const faqItem = button.parentElement;

const isActive = faqItem.classList.contains("active");

document.querySelectorAll(".faq-item").forEach((item) => {

item.classList.remove("active");

item.querySelector("button").setAttribute("aria-expanded", "false");

});

if (!isActive) {

faqItem.classList.add("active");

button.setAttribute("aria-expanded", "true");

}

});

});

// Enhancement Widgets -> open corresponding modal

const enhancementWidgets = document.querySelectorAll(".enhancement-widget");

enhancementWidgets.forEach((widget) => {

widget.addEventListener("click", () => {

const modalId = widget.getAttribute("data-modal");

openModal(modalId);

});

});

// Upgrade toggles

const upgradeOptions = document.querySelectorAll(".upgrade-option");

upgradeOptions.forEach((option) => {

const upgradeButton = option.querySelector("button");

upgradeButton.addEventListener("click", () => {

const details = option.querySelector(".upgrade-details");

details.style.display = details.style.display === "block" ? "none" : "block";

});

});

let selectedUpgrade = null;

const upgradeOptionButtons = document.querySelectorAll(".upgrade-button");

upgradeOptionButtons.forEach((button) => {

button.addEventListener("click", () => {

upgradeOptionButtons.forEach((btn) => btn.classList.remove("selected"));

button.classList.add("selected");

const price = parseInt(button.getAttribute("data-price"), 10);

const time = button.getAttribute("data-time");

const isEarly = button.closest(".upgrade-option").id === "upgrade-early-option";

selectedUpgrade = { price, time, isEarly };

const confirmationMessage = isEarly

? `Do you want to set your check-in time to ${time} for $${(price / 100).toFixed(2)}?`

: `Do you want to set your check-out time to ${time} for $${(price / 100).toFixed(2)}?`;

document.getElementById("upgrade-confirmation-message").textContent = confirmationMessage;

button.closest(".upgrade-details").style.display = "none";

openModal("upgrade-confirmation-modal");

});

});

if (document.getElementById("confirm-upgrade")) {

document.getElementById("confirm-upgrade").addEventListener("click", () => {

if (selectedUpgrade) {

const { price, time, isEarly } = selectedUpgrade;

closeModal("upgrade-confirmation-modal");

// If the upgrade has a fee, we let the user pay or confirm.

// For no-charge times, auto-apply right away:

if (price === 0) {

if (isEarly) {

userSelectedCheckinTime = time;

upgradeEarlyButton.textContent = `Selected: ${time} - $${(price / 100).toFixed(2)}`;

isCheckinTimeSelected = true;

upgradeLateGroup.classList.remove("display-none");

} else {

userSelectedCheckoutTime = time;

upgradeLateButton.textContent = `Selected: ${time} - $${(price / 100).toFixed(2)}`;

isCheckoutTimeSelected = true;

}

updateReservationPreview();

checkAllSelections();

alert("Upgrade applied (no charge).");

startTimers();

} else {

alert(`Time selected: ${time} for $${(price / 100).toFixed(2)}. (Please finalize in the cart or confirm PIN as needed.)`);

}

selectedUpgrade = null;

}

});

}

if (document.getElementById("cancel-upgrade")) {

document.getElementById("cancel-upgrade").addEventListener("click", () => {

closeModal("upgrade-confirmation-modal");

selectedUpgrade = null;

});

}

const passbookForm = document.getElementById("passbook-form");

if (passbookForm) {

passbookForm.addEventListener("submit", (e) => {

e.preventDefault();

let isValid = true;

const checkinError = document.getElementById("error-checkin-datetime");

const checkoutError = document.getElementById("error-checkout-datetime");

if (!selectedCheckinDate) {

checkinError.textContent = "Check-in date is required.";

checkinError.classList.add("visible");

isValid = false;

} else {

checkinError.textContent = "";

checkinError.classList.remove("visible");

}

if (!selectedCheckoutDate) {

checkoutError.textContent = "Check-out date is required.";

checkoutError.classList.add("visible");

isValid = false;

} else {

checkoutError.textContent = "";

checkoutError.classList.remove("visible");

}

if (selectedCheckinDate && selectedCheckoutDate && selectedCheckoutDate <= selectedCheckinDate) {

checkoutError.textContent = "Check-out date must be after check-in date.";

checkoutError.classList.add("visible");

isValid = false;

}

if (guestNameInput && !guestNameInput.value) {

alert("Please enter guest name.");

isValid = false;

}

if (isValid && isCheckinTimeSelected && isCheckoutTimeSelected) {

startDownload();

} else if (isValid && (!isCheckinTimeSelected || !isCheckoutTimeSelected)) {

alert("Please select early check-in time first, then late check-out time before submitting.");

}

});

}

function startDownload() {

const mainContent = document.getElementById("main-content");

if (mainContent) {

mainContent.classList.remove("disabled");

}

closeModal("generate-passbook-modal");

triggerPassbookDownload();

const downloadButton = document.getElementById("download-guest-pass-button");

if (downloadButton) {

downloadButton.style.display = "inline-block";

}

scheduleUserNotifications();

showWelcomeOverlay();

}

function triggerPassbookDownload() {

const storedPkpassLink =

localStorage.getItem("pkpassLink") ||

"https://cdn.shopify.com/s/files/1/0671/8111/7609/files/4A5FA684-1F60-4706-B0A3-0D3D560DAACD.pkpass?v=1729416167";

const link = document.createElement("a");

link.href = storedPkpassLink;

link.download = "GuestPass.pkpass";

document.body.appendChild(link);

link.click();

document.body.removeChild(link);

}

function showWelcomeOverlay() {

const welcomeOverlay = document.getElementById("welcome-overlay");

const welcomeText = document.getElementById("welcome-text");

const guestName = guestNameInput ? guestNameInput.value : "Guest";

if (welcomeText) {

welcomeText.textContent = `Welcome ${guestName}!`;

}

if (welcomeOverlay) {

welcomeOverlay.style.display = "flex";

const dismissButton = welcomeOverlay.querySelector(".dismiss-welcome");

if (dismissButton) {

dismissButton.addEventListener("click", () => {

welcomeOverlay.style.display = "none";

});

}

}

}

// ------------------------------

// UPDATED scheduleUserNotifications

// ------------------------------

function scheduleUserNotifications() {

if (!selectedCheckinDate || !selectedCheckoutDate || !userSelectedCheckinTime || !userSelectedCheckoutTime) return;

const checkinDateTime = new Date(`${formatDate(selectedCheckinDate)} ${userSelectedCheckinTime}`);

const checkoutDateTime = new Date(`${formatDate(selectedCheckoutDate)} ${userSelectedCheckoutTime}`);

const guestName = guestNameInput ? guestNameInput.value : "Guest";

// 1. Welcome Notification (morning of check-in date, 9 AM)

let welcomeTime = new Date(checkinDateTime);

welcomeTime.setHours(9, 0, 0, 0);

scheduleNotification(

"Welcome to Vista Charm!",

`Hi ${guestName}, we’re excited to have you! Your check-in is at ${userSelectedCheckinTime}. Need assistance? We’re available from 8 AM - 9 PM.`,

welcomeTime

);

// 2. Check-In Reminder (if user wants it, from localStorage)

const userCheckinReminder = localStorage.getItem("checkinReminder") ||

"Reminder: Your check-in is soon. Please let us know if you need anything before arrival. We’re here to help!";

const userCheckinReminderTime = localStorage.getItem("checkinReminderTime") || "14:00";

const [reminderCHour, reminderCMinute] = userCheckinReminderTime.split(":").map(num => parseInt(num, 10));

const checkinReminderDate = new Date(selectedCheckinDate);

checkinReminderDate.setHours(reminderCHour, reminderCMinute, 0, 0);

if (checkinReminderDate > new Date()) {

scheduleNotification("Check-In Reminder", userCheckinReminder, checkinReminderDate);

}

// 3. Check-Out Reminder

const userCheckoutReminder = localStorage.getItem("checkoutReminder") ||

"Reminder: Your check-out is scheduled soon. Please leave the property in good condition. Safe travels!";

const userCheckoutReminderTime = localStorage.getItem("checkoutReminderTime") || "10:00";

const [reminderOutHour, reminderOutMinute] = userCheckoutReminderTime.split(":").map(num => parseInt(num, 10));

const checkoutReminderDate = new Date(selectedCheckoutDate);

checkoutReminderDate.setHours(reminderOutHour, reminderOutMinute, 0, 0);

if (checkoutReminderDate > new Date()) {

scheduleNotification("Check-Out Reminder", userCheckoutReminder, checkoutReminderDate);

}

// 4. Recurring Notifications - default every X hours from check-in to check-out

scheduleRecurringNotifications(checkinDateTime, checkoutDateTime);

}

function scheduleRecurringNotifications(startDateTime, endDateTime) {

const recurringInterval = parseInt(localStorage.getItem("recurringIntervalHours"), 10) || 3;

const messages = localStorage.getItem("recurringMessages") || "";

if (!messages.trim()) return;

const lines = messages.split("\n").map(line => line.trim()).filter(Boolean);

if (!lines.length) return;

// We'll schedule messages at an interval from check-in to check-out

let currentTime = new Date(startDateTime.getTime() + recurringInterval \* 60 \* 60 \* 1000);

while (currentTime < endDateTime) {

// Choose a random message from the lines array or cycle them

const chosenMessage = lines[Math.floor(Math.random() \* lines.length)];

scheduleNotification("Vista Charm Update", chosenMessage, new Date(currentTime));

// Move forward by the interval

currentTime.setHours(currentTime.getHours() + recurringInterval);

}

}

function scheduleNotification(title, body, time) {

const now = Date.now();

const target = time.getTime();

const delay = target - now;

if (delay > 0) {

setTimeout(() => {

sendNotification(title, body);

}, delay);

}

}

function formatDate(date) {

const mm = String(date.getMonth() + 1).padStart(2, "0");

const dd = String(date.getDate()).padStart(2, "0");

const yyyy = date.getFullYear();

return `${mm}/${dd}/${yyyy}`;

}

function updateReservationPreview() {

const checkin = selectedCheckinDate

? `${formatDate(selectedCheckinDate)} ${userSelectedCheckinTime}`

: "--";

const checkout = selectedCheckoutDate

? `${formatDate(selectedCheckoutDate)} ${userSelectedCheckoutTime}`

: "--";

const previewCheckin = document.getElementById("preview-checkin");

const previewCheckout = document.getElementById("preview-checkout");

const checkinDisplay = document.getElementById("checkin-display");

const checkoutDisplay = document.getElementById("checkout-display");

const checkinDate = document.getElementById("checkin-date");

const checkoutDate = document.getElementById("checkout-date");

if (previewCheckin) previewCheckin.textContent = checkin;

if (previewCheckout) previewCheckout.textContent = checkout;

if (checkinDisplay) checkinDisplay.textContent = userSelectedCheckinTime;

if (checkoutDisplay) checkoutDisplay.textContent = userSelectedCheckoutTime;

if (checkinDate) checkinDate.textContent = selectedCheckinDate ? formatDate(selectedCheckinDate) : "--/--/----";

if (checkoutDate) checkoutDate.textContent = selectedCheckoutDate ? formatDate(selectedCheckoutDate) : "--/--/----";

}

function checkAllSelections() {

const isAllSelected =

isCheckinTimeSelected &&

isCheckoutTimeSelected &&

guestNameInput &&

guestNameInput.value.trim() !== "";

if (isAllSelected) {

generatePassbookButton.classList.add("visible");

} else {

generatePassbookButton.classList.remove("visible");

}

}

// Initialize Litepicker

const datePicker = new Litepicker({

element: document.getElementById("daterange"),

singleMode: false,

allowRepick: true,

format: "MM-DD-YYYY",

minDate: new Date(),

setup: (picker) => {

picker.on("selected", (date1, date2) => {

selectedCheckinDate = date1.toJSDate();

selectedCheckoutDate = date2.toJSDate();

isDatesSelected = true;

updateReservationPreview();

upgradeEarlyGroup.classList.remove("display-none");

checkAllSelections();

picker.hide();

localStorage.setItem("selectedCheckinDate", selectedCheckinDate.toISOString());

localStorage.setItem("selectedCheckoutDate", selectedCheckoutDate.toISOString());

startTimers();

});

},

});

// Add to home screen

let deferredPrompt;

window.addEventListener("beforeinstallprompt", (e) => {

e.preventDefault();

deferredPrompt = e;

openModal("add-to-home-modal");

});

const addToHomeButton = document.getElementById("add-to-home-button");

const yourGuestPassButton = document.getElementById("your-guest-pass-button");

const addToHomeInstructions = document.getElementById("add-to-home-instructions");

if (addToHomeButton) {

addToHomeButton.addEventListener("click", async () => {

if (deferredPrompt) {

deferredPrompt.prompt();

const { outcome } = await deferredPrompt.userChoice;

deferredPrompt = null;

closeModal("add-to-home-modal");

alert("Your security code is: 1234");

} else {

alert("Please add this app to your home screen manually.");

}

});

}

if (yourGuestPassButton) {

yourGuestPassButton.addEventListener("click", () => {

closeModal("add-to-home-modal");

});

}

window.addEventListener("DOMContentLoaded", () => {

if (addToHomeInstructions) {

if (navigator.userAgent.match(/iPhone|iPad|iPod/i)) {

addToHomeInstructions.innerHTML = 'Tap the share icon and select "Add to Home Screen".';

} else if (navigator.userAgent.match(/Android/i)) {

addToHomeInstructions.textContent = "Follow the prompt to add to your home screen.";

} else {

addToHomeInstructions.textContent = "Please add this app to your home screen manually.";

}

}

});

// PIN Confirmation Functions for individual enhancements

function confirmEarlyCheckInPin() {

const pin = document.getElementById("early-checkin-pin").value.trim();

const storedPin = localStorage.getItem("userPIN") || "123";

if (pin !== storedPin) {

alert("Invalid PIN. Please enter the correct PIN.");

return;

}

alert("Early Check-In confirmed!");

}

function confirmLateCheckOutPin() {

const pin = document.getElementById("late-checkout-pin").value.trim();

const storedPin = localStorage.getItem("userPIN") || "123";

if (pin !== storedPin) {

alert("Invalid PIN. Please enter the correct PIN.");

return;

}

alert("Late Check-Out confirmed!");

}

function confirmToiletriesPin() {

const pin = document.getElementById("toiletries-pin").value.trim();

const storedPin = localStorage.getItem("userPIN") || "123";

if (pin !== storedPin) {

alert("Invalid PIN. Please enter the correct PIN.");

return;

}

alert("Toiletries confirmed!");

}

function confirmDetergentsPin() {

const pin = document.getElementById("detergents-pin").value.trim();

const storedPin = localStorage.getItem("userPIN") || "123";

if (pin !== storedPin) {

alert("Invalid PIN. Please enter the correct PIN.");

return;

}

alert("Detergents confirmed!");

}

function confirmFoodSnacksPin() {

const pin = document.getElementById("food-snacks-pin").value.trim();

const storedPin = localStorage.getItem("userPIN") || "123";

if (pin !== storedPin) {

alert("Invalid PIN. Please enter the correct PIN.");

return;

}

alert("Food & Snacks confirmed!");

}

// Handle the actual selection from the dropdown for each enhancement

function handleEarlyCheckinSelection() {

const select = document.getElementById("early-checkin-options");

const time = select.value;

const price = parseInt(select.options[select.selectedIndex].getAttribute("data-price"), 10);

addToCart(`Early Check-In (${time})`, price);

// If price=0, we assume userSelectedCheckinTime changes immediately

if (price === 0) {

userSelectedCheckinTime = time;

isCheckinTimeSelected = true;

updateReservationPreview();

checkAllSelections();

startTimers();

}

}

function handleLateCheckoutSelection() {

const select = document.getElementById("late-checkout-options");

const time = select.value;

const price = parseInt(select.options[select.selectedIndex].getAttribute("data-price"), 10);

addToCart(`Late Check-Out (${time})`, price);

if (price === 0) {

userSelectedCheckoutTime = time;

isCheckoutTimeSelected = true;

updateReservationPreview();

checkAllSelections();

startTimers();

}

}

function handleToiletriesSelection() {

const select = document.getElementById("toiletries-options");

const product = select.value;

const price = parseInt(select.options[select.selectedIndex].getAttribute("data-price"), 10);

addToCart(`Toiletries: ${product}`, price);

}

function handleDetergentsSelection() {

const select = document.getElementById("detergents-options");

const product = select.value;

const price = parseInt(select.options[select.selectedIndex].getAttribute("data-price"), 10);

addToCart(`Detergents: ${product}`, price);

}

function handleFoodSnacksSelection() {

const select = document.getElementById("food-snacks-options");

const product = select.value;

const price = parseInt(select.options[select.selectedIndex].getAttribute("data-price"), 10);

addToCart(`Food & Snacks: ${product}`, price);

}

</script>

<script>

// NEW/UPDATED CODE: Save Enhancements Inventory (just a placeholder)

function saveEnhancementsInventory() {

// In a real app, you'd parse these textareas and store them properly.

const earlyCheckinTxt = document.getElementById("early-checkin-edit").value;

const lateCheckoutTxt = document.getElementById("late-checkout-edit").value;

const toiletriesTxt = document.getElementById("toiletries-edit").value;

const detergentsTxt = document.getElementById("detergents-edit").value;

const foodSnacksTxt = document.getElementById("food-snacks-edit").value;

// For demo, just store them in localStorage or show an alert.

localStorage.setItem("enhanceEarlyCheckin", earlyCheckinTxt);

localStorage.setItem("enhanceLateCheckout", lateCheckoutTxt);

localStorage.setItem("enhanceToiletries", toiletriesTxt);

localStorage.setItem("enhanceDetergents", detergentsTxt);

localStorage.setItem("enhanceFoodSnacks", foodSnacksTxt);

alert("Enhancements Inventory updated (stored in localStorage).");

closeModal("edit-enhancements-inventory-modal");

}

</script>

</body>

</html>