<!DOCTYPE html>

<html lang="tr">

<head>

<meta charset="UTF-8">

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

<title>TatilKaptani.com - Kaptanınız Yanınızda</title>

<script src="https://www.gstatic.com/firebasejs/9.6.10/firebase-app-compat.js"></script>

<script src="https://www.gstatic.com/firebasejs/9.6.10/firebase-auth-compat.js"></script>

<script src="https://www.gstatic.com/firebasejs/9.6.10/firebase-firestore-compat.js"></script>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css">

<style>

/\* CSS Kodları Başlangıcı \*/

body {

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

margin: 0;

padding: 0;

background-color: #f5f7fa; /\* Açık gri arka plan \*/

color: #333;

display: flex;

flex-direction: column;

min-height: 100vh;

background-image: url('https://source.unsplash.com/random/1920x1080/?travel,beach,mountain'); /\* Tatil havası katan arka plan \*/

background-size: cover;

background-attachment: fixed;

background-position: center;

}

header {

background-color: rgba(0, 50, 100, 0.7); /\* Koyu mavi, hafif şeffaf \*/

color: white;

padding: 1rem 2rem;

display: flex;

justify-content: space-between;

align-items: center;

box-shadow: 0 2px 4px rgba(0,0,0,0.1);

}

.logo {

font-size: 1.8rem;

font-weight: bold;

color: #fff;

text-decoration: none;

}

nav a {

color: white;

text-decoration: none;

margin-left: 20px;

font-weight: 500;

transition: color 0.3s ease;

}

nav a:hover {

color: #a0dfff; /\* Açık mavi hover \*/

}

.user-actions {

display: flex;

gap: 10px;

}

.user-actions button {

background-color: #007bff;

color: white;

border: none;

padding: 8px 15px;

border-radius: 5px;

cursor: pointer;

font-weight: 500;

transition: background-color 0.3s ease;

}

.user-actions button:hover {

background-color: #0056b3;

}

#loggedInUser span {

margin-right: 10px;

font-weight: bold;

}

#loggedInUser button {

background-color: #dc3545;

}

#loggedInUser button:hover {

background-color: #c82333;

}

main {

flex-grow: 1;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

padding: 20px;

text-align: center;

}

.hero-section {

background-color: rgba(255, 255, 255, 0.7); /\* Beyaz, şeffaf \*/

padding: 3rem 4rem;

border-radius: 15px;

box-shadow: 0 8px 25px rgba(0,0,0,0.2);

backdrop-filter: blur(5px); /\* Hafif bulanıklık efekti \*/

-webkit-backdrop-filter: blur(5px); /\* Safari desteği \*/

max-width: 800px;

width: 90%;

margin-bottom: 30px;

}

.hero-section h1 {

font-size: 3rem;

color: #003366; /\* Koyu mavi başlık \*/

margin-bottom: 15px;

text-shadow: 1px 1px 2px rgba(0,0,0,0.1);

}

.slogan {

font-size: 1.5rem;

color: #555;

margin-bottom: 25px;

font-style: italic;

}

.chat-container {

background-color: rgba(255, 255, 255, 0.85); /\* Beyaz, daha az şeffaf \*/

padding: 20px;

border-radius: 10px;

box-shadow: 0 4px 15px rgba(0,0,0,0.15);

width: 90%;

max-width: 600px;

display: flex;

flex-direction: column;

max-height: 70vh;

backdrop-filter: blur(3px);

-webkit-backdrop-filter: blur(3px);

opacity: 0; /\* Başlangıçta gizli \*/

transform: translateY(20px);

transition: opacity 0.5s ease-out, transform 0.5s ease-out;

}

.chat-container.visible {

opacity: 1;

transform: translateY(0);

}

.chat-messages {

flex-grow: 1;

overflow-y: auto;

border: 1px solid #e0e0e0;

padding: 10px;

margin-bottom: 15px;

border-radius: 8px;

background-color: #fafafa;

}

.message {

margin-bottom: 10px;

line-height: 1.5;

}

.message.user {

text-align: right;

color: #0056b3;

}

.message.ai {

text-align: left;

color: #333;

}

.message span {

background-color: #e0f2f7;

padding: 8px 12px;

border-radius: 15px;

display: inline-block;

max-width: 80%;

word-wrap: break-word;

}

.message.user span {

background-color: #d1e7fd;

}

.chat-input {

display: flex;

gap: 10px;

}

.chat-input input[type="text"] {

flex-grow: 1;

padding: 12px;

border: 1px solid #ccc;

border-radius: 25px;

font-size: 1rem;

outline: none;

transition: border-color 0.3s ease;

}

.chat-input input[type="text"]:focus {

border-color: #007bff;

}

.chat-input button {

padding: 12px 20px;

background-color: #28a745; /\* Yeşil gönder butonu \*/

color: white;

border: none;

border-radius: 25px;

cursor: pointer;

font-size: 1rem;

transition: background-color 0.3s ease;

}

.chat-input button:hover {

background-color: #218838;

}

footer {

margin-top: auto; /\* Alt kısma sabitle \*/

background-color: rgba(0, 0, 0, 0.6); /\* Siyah, şeffaf \*/

color: white;

text-align: center;

padding: 1rem 0;

}

/\* Reklam Alanı \*/

.ad-area {

position: fixed;

right: 20px;

top: 50%;

transform: translateY(-50%);

display: flex;

flex-direction: column;

gap: 15px;

z-index: 1000;

}

.ad-item {

background-color: rgba(255, 255, 255, 0.7);

padding: 10px;

border-radius: 8px;

box-shadow: 0 2px 8px rgba(0,0,0,0.1);

text-align: center;

width: 150px;

min-height: 100px;

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

font-size: 0.9rem;

color: #555;

backdrop-filter: blur(2px);

-webkit-backdrop-filter: blur(2px);

}

.ad-item a {

color: #007bff;

text-decoration: none;

margin-top: 5px;

}

/\* Modals for Login/Register/Forgot Password \*/

.modal {

display: none; /\* Hidden by default \*/

position: fixed; /\* Stay in place \*/

z-index: 1001; /\* Sit on top \*/

left: 0;

top: 0;

width: 100%; /\* Full width \*/

height: 100%; /\* Full height \*/

overflow: auto; /\* Enable scroll if needed \*/

background-color: rgba(0,0,0,0.6); /\* Black w/ opacity \*/

justify-content: center;

align-items: center;

}

.modal-content {

background-color: #fefefe;

margin: auto;

padding: 30px;

border-radius: 10px;

box-shadow: 0 5px 15px rgba(0,0,0,0.3);

width: 90%;

max-width: 400px;

text-align: center;

position: relative;

}

.close-button {

color: #aaa;

position: absolute;

right: 15px;

top: 10px;

font-size: 28px;

font-weight: bold;

cursor: pointer;

}

.close-button:hover,

.close-button:focus {

color: black;

text-decoration: none;

cursor: pointer;

}

.modal-content input {

margin-bottom: 15px;

width: calc(100% - 20px); /\* Padding için ayarlandı \*/

padding: 10px;

border: 1px solid #ccc;

border-radius: 5px;

}

.modal-content button {

margin-bottom: 10px;

width: 100%;

padding: 10px;

border-radius: 5px;

border: none;

background-color: #007bff;

color: white;

cursor: pointer;

}

.modal-content p {

font-size: 0.9rem;

cursor: pointer;

color: #007bff;

text-decoration: underline;

}

.modal-content p:hover {

color: #0056b3;

}

.modal-message {

margin-top: 10px;

font-weight: bold;

}

/\* Sidebar (Menü) ve İçerik Alanı Stilleri \*/

.main-layout {

display: flex;

width: 100%;

height: 100vh;

}

.left-sidebar {

width: 250px;

background-color: rgba(0, 77, 64, 0.95); /\* Koyu çamur yeşili \*/

color: white;

padding: 20px 0;

box-shadow: 4px 0 15px rgba(0, 0, 0, 0.3);

display: flex;

flex-direction: column;

z-index: 100;

}

.left-sidebar h2 {

text-align: center;

color: #e0f2f1;

margin-bottom: 30px;

font-size: 1.8em;

border-bottom: 1px solid rgba(255, 255, 255, 0.2);

padding-bottom: 15px;

}

.sidebar-nav ul {

list-style: none;

padding: 0;

margin: 0;

}

.sidebar-nav li {

margin-bottom: 10px;

}

.sidebar-nav button {

background: none;

border: none;

color: white;

padding: 15px 20px;

width: 100%;

text-align: left;

font-size: 1.1em;

cursor: pointer;

transition: background-color 0.3s ease, padding-left 0.3s ease;

display: flex;

align-items: center;

gap: 10px;

}

.sidebar-nav button:hover, .sidebar-nav button.active {

background-color: rgba(0, 121, 107, 0.8); /\* Açık çamur yeşili \*/

padding-left: 25px;

}

.sidebar-nav button i {

font-size: 1.2em;

}

.content-area {

flex: 1;

display: flex;

flex-direction: column;

background-color: rgba(255, 255, 255, 0.4); /\* Şeffaflığı artırdık \*/

margin: 20px; /\* Kenarlardan boşluk \*/

border-radius: 15px;

box-shadow: 0 8px 25px rgba(0, 0, 0, 0.2);

position: relative;

overflow: hidden; /\* İçerik için scroll \*/

}

.top-bar {

padding: 15px 25px;

background-color: rgba(0, 121, 107, 0.8);

color: white;

display: flex;

justify-content: space-between;

align-items: center;

border-top-left-radius: 15px;

border-top-right-radius: 15px;

box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);

position: relative;

z-index: 50;

}

.top-bar .info-group {

display: flex;

gap: 20px;

align-items: center;

font-weight: 600;

}

.top-bar .info-group span {

display: flex;

align-items: center;

gap: 5px;

}

#slogan-top {

font-style: italic;

font-size: 0.9em;

flex-grow: 1;

text-align: center;

}

.top-bar button {

background: none;

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

color: white;

padding: 8px 12px;

border-radius: 20px;

cursor: pointer;

transition: background-color 0.3s, border-color 0.3s;

font-size: 0.9em;

}

.top-bar button:hover {

background-color: rgba(255, 255, 255, 0.2);

border-color: white;

}

.content-section {

flex: 1;

padding: 25px;

overflow-y: auto; /\* İçerik kaydırılabilir olsun \*/

display: none; /\* Varsayılan olarak gizli \*/

flex-direction: column;

background-color: rgba(249, 249, 249, 0.4); /\* İçerik bölümünü de şeffaf yaptık \*/

border-bottom-left-radius: 15px;

border-bottom-right-radius: 15px;

}

.content-section.active {

display: flex; /\* Aktif olanı göster \*/

}

/\* Chat Stilleri - Mevcut koddaki chat-messages ve message stillerini kullanabiliriz \*/

/\* Input ve Button stilleri \*/

.input-group {

display: flex;

gap: 10px;

margin-top: 15px;

}

input[type="text"],

input[type="email"],

input[type="password"],

input[type="number"],

select,

textarea,

input[type="file"] { /\* File input için de stil \*/

flex: 1;

padding: 14px 20px;

font-size: 17px;

border: 1px solid #a7d9b9;

border-radius: 30px;

outline: none;

transition: border-color 0.3s, box-shadow 0.3s;

background-color: #f9fdf9;

}

input[type="text"]:focus,

input[type="email"]:focus,

input[type="password"]:focus,

input[type="number"]:focus,

select:focus,

textarea:focus,

input[type="file"]:focus {

border-color: #004d40;

box-shadow: 0 0 8px rgba(0, 121, 107, 0.2);

}

button.send-btn {

padding: 14px 25px;

font-size: 17px;

background-color: #00796b;

color: white;

border: none;

border-radius: 30px;

cursor: pointer;

transition: background-color 0.3s ease, transform 0.1s;

}

button.send-btn:hover {

background-color: #004d40;

transform: translateY(-1px);

}

button.send-btn:active {

transform: translateY(0);

}

/\* Feature sections and plan output \*/

.feature-section {

padding: 20px;

background-color: rgba(224, 242, 241, 0.5);

border-radius: 10px;

margin-bottom: 20px;

box-shadow: 0 2px 8px rgba(0, 0, 0, 0.08);

}

.feature-section h3 {

color: #004d40;

margin-top: 0;

border-bottom: 1px solid #c8e6c9;

padding-bottom: 10px;

margin-bottom: 20px;

}

.feature-section p {

line-height: 1.6;

color: #333;

}

.feature-section button {

background-color: #00796b;

color: white;

padding: 10px 20px;

border: none;

border-radius: 5px;

cursor: pointer;

margin-top: 15px;

transition: background-color 0.3s;

}

.feature-section button:hover {

background-color: #004d40;

}

.feature-form label {

display: block;

margin-bottom: 8px;

font-weight: bold;

color: #004d40;

}

.feature-form .budget-options button {

background-color: #004d40;

color: white;

padding: 8px 15px;

border: none;

border-radius: 20px;

cursor: pointer;

margin-right: 10px;

margin-bottom: 10px;

}

.feature-form .budget-options button.selected {

background-color: #004d40;

border: 2px solid #e0f2f1;

}

.plan-output {

background-color: rgba(240, 253, 252, 0.5);

border: 1px dashed #004d40;

padding: 20px;

border-radius: 10px;

margin-top: 20px;

}

.plan-output h4 {

color: #004d40;

margin-top: 0;

margin-bottom: 15px;

font-size: 1.3em;

}

.plan-output p, .plan-output ul, .plan-output ol {

color: #333;

line-height: 1.6;

}

.plan-output img {

max-width: 100%;

height: auto;

border-radius: 8px;

margin-top: 15px;

box-shadow: 0 4px 10px rgba(0,0,0,0.1);

}

/\* VIP Üyelik Formu \*/

#payment-section .card-input-group {

display: flex;

flex-wrap: wrap;

gap: 10px;

margin-bottom: 15px;

}

#payment-section .card-input-group input {

flex: 1;

min-width: 120px;

}

#payment-section .card-input-group input:nth-child(1) { flex-basis: 100%; } /\* Card Number \*/

#payment-section .card-input-group input:nth-child(2),

#payment-section .card-input-group input:nth-child(3) { flex-basis: calc(50% - 5px); } /\* Expiry, CVV \*/

/\* Palmiye Kaptan Avatarı \*/

#avatar {

position: fixed;

top: 25px;

right: 25px;

width: 90px;

height: 90px;

background-image: url('https://via.placeholder.com/90x90/004d40/ffffff?text=AVATAR'); /\* Yer tutucu resim \*/

background-size: cover;

background-position: center;

border-radius: 50%;

border: 4px solid #004d40;

box-shadow: 0 6px 15px rgba(0, 0, 0, 0.25);

cursor: pointer;

z-index: 1000;

animation: pulse 2s infinite ease-in-out;

}

@keyframes pulse {

0% { transform: scale(1); }

50% { transform: scale(1.03); }

100% { transform: scale(1); }

}

/\* Sağ Reklam/Yönetici Alanı \*/

.right-sidebar {

width: 180px;

padding: 20px 0;

display: flex;

flex-direction: column;

gap: 20px;

z-index: 90;

background-color: rgba(255, 255, 255, 0.4);

box-shadow: -4px 0 15px rgba(0, 0, 0, 0.1);

border-top-left-radius: 15px;

border-bottom-left-radius: 15px;

align-items: center;

}

.ad-area-dynamic {

width: 90%;

height: 180px;

background-color: #f0fdfc;

border: 1px solid #b2dfdb;

border-radius: 8px;

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

text-align: center;

font-size: 12px;

color: #555;

cursor: pointer;

box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);

transition: transform 0.2s ease, box-shadow 0.2s ease;

overflow: hidden;

padding: 10px;

box-sizing: border-box;

}

.ad-area-dynamic:hover {

transform: translateY(-3px);

box-shadow: 0 5px 15px rgba(0, 0, 0, 0.2);

}

.ad-area-dynamic img {

max-width: 100%;

max-height: 70%;

object-fit: cover;

margin-bottom: 5px;

border-radius: 5px;

}

/\* Yönetici ilan metni için özel stil \*/

.admin-ad-message {

width: 90%;

padding: 15px;

background-color: rgba(224, 242, 241, 0.5);

border: 1px dashed #00796b;

border-radius: 8px;

text-align: center;

font-size: 1.0em;

color: #004d40;

margin-bottom: 10px;

}

/\* Genel Mesaj Kutusu (Modal gibi) \*/

.modal {

display: none;

position: fixed;

z-index: 1001;

left: 0;

top: 0;

width: 100%;

height: 100%;

overflow: auto;

background-color: rgba(0,0,0,0.6);

justify-content: center;

align-items: center;

}

.modal-content {

background-color: #fefefe;

margin: auto;

padding: 30px;

border-radius: 10px;

box-shadow: 0 5px 15px rgba(0,0,0,0.3);

width: 90%;

max-width: 400px;

text-align: center;

position: relative;

}

.close-button {

color: #aaa;

position: absolute;

right: 15px;

top: 10px;

font-size: 28px;

font-weight: bold;

cursor: pointer;

}

.close-button:hover,

.close-button:focus {

color: black;

text-decoration: none;

cursor: pointer;

}

.modal-content input {

margin-bottom: 15px;

width: calc(100% - 20px);

padding: 10px;

border: 1px solid #ccc;

border-radius: 5px;

}

.modal-content button {

margin-bottom: 10px;

width: 100%;

padding: 10px;

border-radius: 5px;

border: none;

background-color: #007bff;

color: white;

cursor: pointer;

}

.modal-content p {

font-size: 0.9rem;

cursor: pointer;

color: #007bff;

text-decoration: underline;

}

.modal-content p:hover {

color: #0056b3;

}

.modal-message {

margin-top: 10px;

font-weight: bold;

}

/\* Loading Indicator \*/

.loading-indicator {

text-align: center;

margin-top: 10px;

color: #00796b;

font-weight: bold;

display: none;

}

/\* Mobile Responsive \*/

@media (max-width: 768px) {

.main-layout {

flex-direction: column;

}

.left-sidebar {

width: 100%;

height: auto;

padding: 10px 0;

box-shadow: 0 4px 15px rgba(0, 0, 0, 0.3);

}

.left-sidebar h2 {

margin-bottom: 10px;

}

.sidebar-nav ul {

display: flex;

flex-wrap: wrap;

justify-content: center;

gap: 5px;

}

.sidebar-nav li {

margin-bottom: 0;

}

.sidebar-nav button {

padding: 10px 15px;

font-size: 0.9em;

gap: 5px;

}

.sidebar-nav button i {

font-size: 1em;

}

.content-area {

margin: 10px;

min-height: 70vh;

}

.top-bar {

flex-wrap: wrap;

justify-content: center;

gap: 10px;

border-radius: 10px 10px 0 0;

}

.top-bar .info-group {

gap: 10px;

}

#slogan-top {

order: 1;

width: 100%;

text-align: center;

margin-top: 10px;

}

.top-bar button {

padding: 6px 10px;

font-size: 0.8em;

}

#avatar {

width: 60px;

height: 60px;

top: 10px;

right: 10px;

}

.right-sidebar {

display: none;

}

.content-section {

padding: 15px;

}

#chat-box {

min-height: 150px;

}

input[type="text"], button.send-btn {

font-size: 1em;

padding: 10px 15px;

}

.feature-form input[type="text"],

.feature-form input[type="number"],

.feature-form select,

.feature-form textarea,

.feature-form input[type="file"] {

width: calc(100% - 20px);

}

}

</style>

</head>

<body>

<div id="avatar" title="Kullanıcı Paneli"></div>

<div id="appModal" class="modal">

<div class="modal-content">

<h3 id="modalTitle"></h3>

<p id="modalMessage"></p>

<button id="modalConfirmBtn">Tamam</button>

</div>

</div>

<div id="loginModal" class="modal">

<div class="modal-content">

<span class="close-button">&times;</span>

<h2>Üye Girişi</h2>

<input type="email" id="loginEmail" placeholder="E-posta Adresi" required>

<input type="password" id="loginPassword" placeholder="Şifre" required>

<button id="performLoginBtn">Giriş Yap</button>

<p id="forgotPasswordLink">Şifremi Unuttum</p>

<div id="loginMessage" class="modal-message"></div>

</div>

</div>

<div id="registerModal" class="modal">

<div class="modal-content">

<span class="close-button">&times;</span>

<h2>Üye Ol</h2>

<input type="text" id="registerUsername" placeholder="Kullanıcı Adı" required>

<input type="email" id="registerEmail" placeholder="E-posta Adresi" required>

<input type="password" id="registerPassword" placeholder="Şifre" required>

<button id="performRegisterBtn">Kayıt Ol</button>

<div id="registerMessage" class="modal-message"></div>

</div>

</div>

<div id="forgotPasswordModal" class="modal">

<div class="modal-content">

<span class="close-button">&times;</span>

<h2>Şifre Sıfırla</h2>

<input type="email" id="resetEmail" placeholder="E-posta Adresiniz" required>

<button id="performResetBtn">Şifre Sıfırlama Linki Gönder</button>

<div id="resetMessage" class="modal-message"></div>

</div>

</div>

<div class="main-layout" style="display: none;"> <div class="left-sidebar">

<h2>Palmiye Kaptan</h2>

<nav class="sidebar-nav">

<ul>

<li><button data-section="chat-section" class="active"><i class="fas fa-comments"></i> Sohbet Asistanı</button></li>

<li><button data-section="game-section"><i class="fas fa-gamepad"></i> Tatil Avı (Oyun)</button></li>

<li><button data-section="virtual-holiday-section"><i class="fas fa-globe-americas"></i> Sanal Tatil Planı</button></li>

<li><button data-section="ai-photo-studio-section"><i class="fas fa-paint-brush"></i> AI Fotoğraf Stüdyosu</button></li>

<li><button data-section="vip-planner-section"><i class="fas fa-plane-departure"></i> VIP A'dan Z'ye Tur Planlayıcı</button></li>

<li><button data-section="user-info-section"><i class="fas fa-user-circle"></i> Üyelik Bilgileri</button></li>

<li><button data-section="time-travel-section"><i class="fas fa-hourglass-half"></i> Zamanda Yolculuk Tatili</button></li>

<li><button data-section="destiny-route-section"><i class="fas fa-crystal-ball"></i> Kader Rotası</button></li>

<li><button data-section="ai-companion-section"><i class="fas fa-robot"></i> AI Yoldaşım</button></li>

<li><button data-section="payment-section"><i class="fas fa-credit-card"></i> VIP Üyelik Al</button></li>

<li><button data-section="contact-us-section"><i class="fas fa-envelope"></i> Bize Ulaşın</button></li>

</ul>

</nav>

</div>

<div class="content-area">

<div class="top-bar">

<div class="info-group">

<span id="tatilpuan-top"><i class="fas fa-star"></i> TatilPuan: 0 (Bronz)</span>

<span id="user-id-display" style="font-size:0.8em; opacity:0.8;"></span>

<button id="voice-toggle-top"><i class="fas fa-volume-up"></i> Ses: Açık</button>

<select id="language-select" style="background-color: rgba(0, 77, 64, 0.8); color: white; border: none; padding: 5px; border-radius: 5px;">

<option value="tr">Türkçe</option>

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

</select>

</div>

<span id="slogan-top"></span>

</div>

<div id="chat-section" class="content-section active">

<h2><i class="fas fa-comments"></i> Sohbet Asistanı</h2>

<div id="chat-box" class="chat-messages" aria-live="polite" aria-label="Sohbet kutusu"></div>

<div class="loading-indicator" id="chat-loading"><i class="fas fa-spinner fa-spin"></i> Palmiye Kaptan düşünüyor...</div>

<div class="input-group">

<input type="text" id="user-input-chat" placeholder="Bir tatil sorusu yazın..." />

<button id="send-button-chat" class="send-btn">Gönder</button>

</div>

</div>

<div id="game-section" class="content-section">

<h2><i class="fas fa-gamepad"></i> Tatil Avı (Mini Oyun)</h2>

<div class="feature-section">

<h3>Palmiye Kaptan'ın Bilgi Yarışması!</h3>

<p>Sana 3 soru soracağım. Doğru cevaplarsan PalmCoin kazanacaksın!</p>

<button id="start-game-btn">Oyuna Başla</button>

<div id="game-output" style="margin-top: 20px;"></div>

<input type="text" id="game-answer-input" placeholder="Cevabınızı buraya yazın (A, B, C)..." style="display: none; width: calc(100% - 20px); padding: 10px; margin-top: 15px; border-radius: 5px;">

<button id="submit-game-answer-btn" style="display: none; margin-top: 10px;">Cevapla</button>

</div>

</div>

<div id="virtual-holiday-section" class="content-section">

<h2><i class="fas fa-globe-americas"></i> Sanal Tatil Planı</h2>

<div class="feature-section">

<h3>Hayalinizdeki Sanal Tatili Yaratın!</h3>

<p>Tatile gidemeyenler için özel olarak tasarlandı! Nereye gitmek istersiniz ve kaç gün kalacaksınız?

Yapay zekamız sizin için o tatili canlı ve sürükleyici bir hikaye ile canlandıracak.</p>

<div class="feature-form">

<label for="virtual-city">Şehir / Yer:</label>

<input type="text" id="virtual-city" placeholder="Örn: Venedik" />

<label for="virtual-days">Gün Sayısı:</label>

<input type="number" id="virtual-days" placeholder="Örn: 5" min="1" />

<label for="virtual-duration-minutes">Sanal Tur Süresi (Dakika): <small>(<span id="virtual-tour-cost">50</span> PalmCoin)</small></label>

<input type="number" id="virtual-duration-minutes" placeholder="Örn: 10" min="1" value="5" />

<label for="virtual-activities">Sanal Tatilde Yapmak İstedikleriniz (Detaylı):</label>

<textarea id="virtual-activities" rows="3" placeholder="Örn: Yerel pazarları gezmek, tekne turuna çıkmak, tarihi yerleri ziyaret etmek..."></textarea>

<label for="virtual-image-prompt">Hediye Resim Açıklaması (Genel Sanal Tatil Teması):</label>

<textarea id="virtual-image-prompt" rows="2" placeholder="Örn: Gün batımında tropik plaj, palmiye ağaçları ve sakin deniz"></textarea>

<button id="start-virtual-btn">Sanal Tatili Oluştur</button>

</div>

<div class="loading-indicator" id="virtual-loading"><i class="fas fa-spinner fa-spin"></i> Sanal tatiliniz hazırlanıyor...</div>

<div id="virtual-holiday-output" class="plan-output" style="display: none;">

<h4 id="virtual-output-title"></h4>

<p id="virtual-output-story"></p>

<div id="virtual-images-container" style="display: flex; flex-wrap: wrap; gap: 15px; justify-content: center; margin-top: 20px;">

</div>

<button id="send-virtual-image-email-btn" style="margin-top: 15px; display: none;">Hediye Resmi E-postana Gönder (Demo)</button>

</div>

</div>

</div>

<div id="ai-photo-studio-section" class="content-section">

<h2><i class="fas fa-paint-brush"></i> AI Fotoğraf Stüdyosu</h2>

<div class="feature-section" id="ai-photo-access-check">

<h3>Bu özellik sadece Altın üyelerimiz içindir!</h3>

<p>Hayalinizdeki görselleri yapay zeka ile gerçeğe dönüştürün! Tek yapmanız gereken, istediğiniz fotoğrafı metinle tanımlamak.</p>

<button id="go-to-ai-photo-payment-btn">Hemen Altın Üye Ol!</button>

</div>

<div id="ai-photo-form-area" style="display: none;">

<div class="feature-section">

<h3>Hayalindeki Fotoğrafı Oluştur (Altın Üye Özel)</h3>

<p>Ne tür bir fotoğraf istediğinizi detaylıca tanımlayın. Örneğin: "Gün batımında tropik bir adada palmiye ağaçları ve sakin bir deniz, gerçeküstü tarzda, yüksek çöz çözünürlüklü".</p>

<div class="feature-form">

<label for="ai-photo-prompt">Fotoğraf Açıklaması:</label>

<textarea id="ai-photo-prompt" rows="4" placeholder="Örn: Sualtı şehrinde yüzen balıklar, fütüristik, parlak renkler..."></textarea>

<label for="ai-photo-style">Fotoğraf Stili:</label>

<select id="ai-photo-style">

<option value="realistic">Gerçekçi</option>

<option value="watercolor">Suluboya</option>

<option value="anime">Anime</option>

<option value="cyberpunk">Siberpunk</option>

<option value="fantasy">Fantastik</option>

</select>

<label for="ai-photo-count">Oluşturulacak Resim Sayısı (1-3):</label>

<input type="number" id="ai-photo-count" value="1" min="1" max="3" />

<button id="generate-ai-photo-btn">Fotoğrafı Oluştur</button>

</div>

<div class="loading-indicator" id="ai-photo-loading"><i class="fas fa-spinner fa-spin"></i> Fotoğrafınız oluşturuluyor...</div>

<div id="ai-photo-output" class="plan-output" style="display: none;">

<h4>Oluşturulan Fotoğraflarınız:</h4>

<div id="generated-images-container" style="display: flex; flex-wrap: wrap; gap: 15px; justify-content: center; margin-top: 15px;">

</div>

<button id="download-all-images-btn" style="margin-top: 15px; display: none;">Tümünü İndir (<span id="download-all-cost"></span> PalmCoin)</button>

</div>

</div>

</div>

</div>

<div id="vip-planner-section" class="content-section">

<h2><i class="fas fa-plane-departure"></i> VIP A'dan Z'ye Tur Planlayıcı</h2>

<div class="feature-section" id="vip-access-check">

<h3>Bu özellik sadece Altın üyelerimiz içindir!</h3>

<p>Tam entegre tatil planlama hizmetimizle, uçuşlardan otellere, transferlerden kişiye özel rotalara kadar her şeyi yapay zekamız sizin için planlıyor.

Tatil stresini tamamen unutun, sadece anın tadını çıkarın!</p>

<button id="go-to-vip-payment-btn">Hemen Altın Üye Ol!</button>

</div>

<div id="vip-planner-form-area" style="display: none;">

<div class="feature-section">

<h3>A'dan Z'ye Tatil Planlama (Altın Üye Özel)</h3>

<p>Hayalinizdeki tatilin tüm detaylarını yapay zekamız sizin için ayarlasın! Otelinden uçağına, transferinden her gezisine kadar sanal yapay zeka ayarlayıp rota çizecek ve size gönderecek!</p>

<div class="feature-form">

<label for="vip-destination">Gidilecek Şehir/Ülke:</label>

<input type="text" id="vip-destination" placeholder="Örn: Tokyo" />

<label for="vip-duration">Kalış Süresi (Gün):</label>

<input type="number" id="vip-duration" placeholder="Örn: 7" min="1" />

<label for="vip-travelers">Kişi Sayısı:</label>

<input type="number" id="vip-travelers" placeholder="Örn: 2" min="1" />

<label for="vip-budget">Bütçe Seviyesi:</label>

<div class="budget-options">

<button data-budget="ekonomik">Ekonomik</button>

<button data-budget="orta">Orta</button>

<button data-budget="lüks">Lüks</button>

</div>

<label for="vip-type">Tatil Tipi:</label>

<select id="vip-type">

<option value="macera">Macera</option>

<option value="dinlenme">Dinlenme</option>

<option value="kültür">Kültür</option>

<option value="romantik">Romantik</option>

<option value="aile">Aile</option>

<option value="alisveris">Alışveriş</option>

</select>

<button id="generate-vip-plan-btn">VIP Planı Oluştur</button>

</div>

<div class="loading-indicator" id="vip-planner-loading"><i class="fas fa-spinner fa-spin"></i> VIP planınız hazırlanıyor...</div>

<div id="vip-plan-output" class="plan-output" style="display: none;"></div>

<div id="vip-plan-chat-area" style="margin-top: 20px; display: none;">

<h3>Plan Hakkında Soru Sor (<small>10 PalmCoin/Soru</small>)</h3>

<div id="vip-plan-chat-box" class="chat-messages" style="max-height: 200px;"></div>

<div class="input-group">

<input type="text" id="vip-plan-input" placeholder="Plan hakkında bir şeyler sorun..." />

<button id="send-vip-plan-message-btn" class="send-btn">Soru Sor</button>

</div>

</div>

</div>

<div class="feature-section" id="niche-vip-request-area">

<h3>Özel Niş Tur Planı (Altın Üye Özel)</h3>

<p>Alışılmışın dışında, tamamen size özel bir tur mu istiyorsunuz? Yapay zekamız, en niş ve özel isteklerinizi bile harika bir tatil planına dönüştürecek!</p>

<div class="feature-form">

<label for="niche-topic">Niş Konu/İlgi Alanı (Örn: Antik Kalıntıları Keşfetme, Uzay Turizmi Simülasyonu, Yemek Festivalleri):</label>

<input type="text" id="niche-topic" placeholder="Niş konu veya ilgi alanı" />

<label for="niche-details">Özel İstekleriniz/Detaylar (Örn: Az bilinen yerler, yerel etkileşimler, sürprizler...):</label>

<textarea id="niche-details" rows="3" placeholder="Ek detaylar..."></textarea>

<button id="generate-niche-plan-btn">Niş Planı Oluştur</button>

</div>

<div class="loading-indicator" id="niche-plan-loading"><i class="fas fa-spinner fa-spin"></i> Niş planınız hazırlanıyor...</div>

<div id="niche-plan-output" class="plan-output" style="display: none;"></div>

</div>

</div>

</div>

<div id="user-info-section" class="content-section">

<h2><i class="fas fa-user-circle"></i> Üyelik Bilgileri</h2>

<div class="feature-section">

<h3>Kullanıcı Profili</h3>

<p>Kullanıcı ID'si: <strong id="display-userid">Yükleniyor...</strong></p>

<p>Adınız: <strong id="display-username">Misafir</strong></p>

<p>E-posta Adresiniz: <strong id="display-user-email">Ayarlanmadı</strong></p>

<button id="set-user-email-btn">E-posta Ayarla / Güncelle</button>

<p>Toplam TatilPuan: <strong id="display-tatilpuan">0</strong></p>

<p>Üyelik Seviyeniz: <strong id="display-membership-level">Bronz</strong></p>

<p>Kazandığınız PalmCoin: <strong id="display-game-score">0</strong></p>

<button id="register-user-btn">Kullanıcı Adı Belirle / Güncelle</button>

</div>

<div class="feature-section" id="palmcoin-history-section">

<h3>PalmCoin Geçmişi (Demo)</h3>

<p>Son kazanılan/harcanan PalmCoin aktiviteleri:</p>

<ul id="palmcoin-history-list">

<li>Başlangıç: 0 PalmCoin</li>

</ul>

</div>

<div class="feature-section" id="admin-message-section">

<h3>Yönetici Mesajı Ayarları (Demo)</h3>

<p style="color: red;">Bu bir demo yönetim panelidir. Gerçek uygulamalarda güvenlik için arka uç gerektirir.</p>

<label for="admin-message-input">Yönetici Mesajı:</label>

<textarea id="admin-message-input" rows="3" placeholder="Yeni bir duyuru veya ilan yazın..."></textarea>

<button id="update-admin-message-btn">Mesajı Yayınla</button>

<div class="loading-indicator" id="admin-message-loading"><i class="fas fa-spinner fa-spin"></i> Mesaj güncelleniyor...</div>

</div>

<div class="feature-section">

<h3>Geçmiş Sorgular (Demo)</h3>

<ul id="past-queries-list">

<li>(Bu kısım için gerçek bir geçmiş kaydı Firebase'e eklenebilir)</li>

<li>Sanal tatil Paris 7 gün</li>

<li>Bana rota öner Fethiye 3 gün</li>

<li>Oyun</li>

</ul>

</div>

<div class="feature-section">

<h3>Rezervasyon Geçmişi (Demo)</h3>

<p>Henüz bir rezervasyon geçmişi bulunmamaktadır. VIP Tur Planlayıcı ile tatilinizi planlayın!</p>

</div>

</div>

<div id="time-travel-section" class="content-section">

<h2><i class="fas fa-hourglass-half"></i> Zamanda Yolculuk Tatili</h2>

<div class="feature-section" id="time-travel-access-check">

<h3>Bu özellik Altın üyelerimize özeldir!</h3>

<p>Tarihin derinliklerinde veya geleceğin gizemlerinde unutulmaz bir yolculuğa çıkın. AI sizin için eş eşsiz bir zaman dilimi simülasyonu yaratacak.</p>

<button id="go-to-time-travel-payment-btn">Hemen Altın Üye Ol!</button>

</div>

<div id="time-travel-form-area" style="display: none;">

<div class="feature-section">

<h3>Zamanda Yolculuk Deneyimi (Altın Üye Özel)</h3>

<p>Nereye ve ne zaman gitmek istersiniz? AI size o dönemi canlandıracak!</p>

<div class="feature-form">

<label for="time-travel-era">Gitmek istediğiniz dönem (Yıl/Dönem):</label>

<input type="text" id="time-travel-era" placeholder="Örn: Antik Roma M.S. 79, 2077 Mars Kolonisi" />

<label for="time-travel-duration">Kalış Süresi (Gün):</label>

<input type="number" id="time-travel-duration" placeholder="Örn: 3" min="1" />

<label for="time-travel-character">Tanışmak istediğiniz hayali/tarihi karakter (Opsiyonel):</label>

<input type="text" id="time-travel-character" placeholder="Örn: Kleopatra, Bir robot mühendis" />

<label for="time-travel-focus">Odak noktası (Örn: Müzik, Yemek, Bilim, Savaş...):</label>

<input type="text" id="time-travel-focus" placeholder="Örn: Dönemin müzikleri, Geleceğin yemekleri" />

<button id="start-time-travel-btn">Zaman Yolculuğunu Başlat</button>

</div>

<div class="loading-indicator" id="time-travel-loading"><i class="fas fa-spinner fa-spin"></i> Zaman makinesi hazırlanıyor...</div>

<div id="time-travel-output" class="plan-output" style="display: none;"></div>

</div>

</div>

</div>

<div id="destiny-route-section" class="content-section">

<h2><i class="fas fa-crystal-ball"></i> Kader Rotası</h2>

<div class="feature-section">

<h3>Palmiye Kaptan'ın Kehaneti!</h3>

<p>AI, gelecekteki rüya tatilinizi tahmin ediyor. Sadece birkaç bilgiye ihtiyacım var.</p>

<div class="feature-form">

<label for="destiny-age">Yaşınız:</label>

<input type="number" id="destiny-age" placeholder="Örn: 30" />

<label for="destiny-hobby">Hobilerinizden biri:</label>

<input type="text" id="destiny-hobby" placeholder="Örn: Doğa yürüyüşü, okumak" />

<label for="destiny-dream">Gelecekle ilgili bir hayaliniz:</label>

<input type="text" id="destiny-dream" placeholder="Örn: Dünyayı gezmek, kitap yazmak" />

<label for="destiny-color">En sevdiğiniz renk (Kehaneti kişiselleştirir):</label>

<input type="text" id="destiny-color" placeholder="Örn: Mavi" />

<button id="predict-destiny-btn">Kader Rotamı Öğren!</button>

</div>

<div class="loading-indicator" id="destiny-loading"><i class="fas fa-spinner fa-spin"></i> Kaderiniz yazılıyor...</div>

<div id="destiny-route-output" class="plan-output" style="display: none;"></div>

<button id="realize-destiny-btn" style="display: none; margin-top: 20px;">Kaderimi Gerçekleştir! (VIP)</button>

</div>

</div>

<div id="ai-companion-section" class="content-section">

<h2><i class="fas fa-robot"></i> AI Yoldaşım</h2>

<div class="feature-section">

<h3>Kaptan Yardımcınızı Oluşturun!</h3>

<p>Tatilinizde size eşlik edecek kişisel bir yapay zeka arkadaşı. Ona bir isim verin ve kişiliğini seçin!</p>

<div class="feature-form">

<label for="companion-name">Yoldaşınızın Adı:</label>

<input type="text" id="companion-name" placeholder="Örn: Seyahatör" />

<label for="companion-personality">Kişiliği:</label>

<select id="companion-personality">

<option value="esprili">Esprili</option>

<option value="bilgili">Bilgili</option>

<option value="sakin">Sakin</option>

<option value="maceraperest">Maceraperest</option>

<option value="ilham-veren">İlham Veren</option>

</select>

<button id="create-companion-btn">Yoldaşımı Oluştur</button>

</div>

<div id="companion-chat-area" style="margin-top: 20px; display: none;">

<h3><span id="active-companion-name"></span> ile Sohbet</h3>

<div id="companion-chat-box" class="chat-messages" style="max-height: 200px;"></div>

<div class="input-group">

<input type="text" id="companion-input" placeholder="Yoldaşınıza bir şeyler sorun..." />

<button id="send-companion-message-btn" class="send-btn">Mesaj Gönder</button>

</div>

</div>

<div class="loading-indicator" id="companion-loading"><i class="fas fa-spinner fa-spin"></i> Yoldaşınız yükleniyor...</div>

</div>

</div>

<div id="payment-section" class="content-section">

<h2><i class="fas fa-credit-card"></i> VIP Üyelik Al</h2>

<div class="feature-section">

<h3>Altın Üyelik: Sınırları Aşan Tatil Deneyimi!</h3>

<p>Aylık sadece \*\*99.99 TL\*\* ile:</p>

<ul>

<li><i class="fas fa-check-circle" style="color: green;"></i> Kişiye Özel A'dan Z'ye Tur Planlama (Uçuş, Otel, Transfer, Günlük Rota)</li>

<li><i class="fas fa-check-circle" style="color: green;"></i> Zamanda Yolculuk Tatilleri</li>

<li><i class="fas fa-check-circle" style="color: green;"></i> Kişisel AI Yoldaşınız</li>

<li><i class="fas fa-check-circle" style="color: green;"></i> Reklamsız Deneyim</li>

<li><i class="fas fa-check-circle" style="color: green;"></i> Özel Müşteri Desteği</li>

<li><i class="fas fa-check-circle" style="color: green;"></i> Niş Tur Planlama</li>

</ul>

<div class="feature-form">

<h3>Ödeme Bilgileri (Demo)</h3>

<p style="color: red;">Bu bir demo ödeme ekranıdır, gerçek bir işlem yapılmayacaktır.</p>

<label for="card-number">Kart Numarası:</label>

<input type="text" id="card-number" placeholder="XXXX XXXX XXXX XXXX" maxlength="19" />

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

<div>

<label for="expiry-date">Son Kullanma Tarihi (AA/YY):</label>

<input type="text" id="expiry-date" placeholder="AA/YY" maxlength="5" />

</div>

<div>

<label for="cvv">CVV:</label>

<input type="text" id="cvv" placeholder="XXX" maxlength="4" />

</div>

</div>

<label for="card-holder-name">Kart Sahibinin Adı:</label>

<input type="text" id="card-holder-name" placeholder="Ad Soyad" />

<button id="complete-payment-btn">Ödemeyi Tamamla (Demo)</button>

</div>

</div>

</div>

<div id="contact-us-section" class="content-section">

<h2><i class="fas fa-envelope"></i> Bize Ulaşın</h2>

<div class="feature-section">

<h3>Sorularınız veya Destek Talepleriniz İçin</h3>

<p>Her türlü soru, öneri veya destek talebiniz için bize yazmaktan çekinmeyin. Ekibimiz en kısa sürede size geri dönecektir.</p>

<div class="feature-form">

<label for="contact-subject">Konu:</label>

<input type="text" id="contact-subject" placeholder="Örn: VIP üyelik hakkında soru" />

<label for="contact-email">E-posta Adresiniz:</label>

<input type="email" id="contact-email" placeholder="Örn: adiniz@ornek.com" />

<label for="contact-message">Mesajınız:</label>

<textarea id="contact-message" rows="5" placeholder="Detaylı mesajınızı buraya yazın..."></textarea>

<label for="contact-file">Dosya Ekle (Opsiyonel):</label>

<input type="file" id="contact-file" />

<button id="send-contact-form-btn">Gönder</button>

</div>

<div class="loading-indicator" id="contact-loading"><i class="fas fa-spinner fa-spin"></i> Mesajınız gönderiliyor...</div>

</div>

</div>

</div>

<div class="right-sidebar">

<div class="admin-ad-message" id="admin-display-message">

<h4>Yönetici Mesajı</h4>

<p>Yükleniyor...</p>

</div>

<div id="dynamic-ads-container">

<a href="https://www.booking.com/index.tr.html" target="\_blank" class="ad-area-dynamic">

<img src="https://images.unsplash.com/photo-1542312389-9828d5427d14?q=80&w=2070&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D" alt="Booking.com Reklamı - Otel Rezervasyonu">

<p>Milyonlarca otelde en iyi fiyatlar!</p>

</a>

<a href="https://www.skyscanner.com.tr/" target="\_blank" class="ad-area-dynamic">

<img src="https://images.unsplash.com/photo-1595015093557-0105342a3d07?q=80&w=1932&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D" alt="Skyscanner Reklamı - Uçak Bileti">

<p>Ucuz uçak bileti mi arıyorsunuz?</p>

</a>

<a href="https://www.getyourguide.com/tr/" target="\_blank" class="ad-area-dynamic">

<img src="https://images.unsplash.com/photo-1628172922129-c1d0637b7891?q=80&w=1932&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D" alt="GetYourGuide Reklamı - Turlar ve Aktiviteler">

<p>Yerel turlar ve aktiviteler keşfet!</p>

</a>

<a href="https://www.tripadvisor.com.tr/" target="\_blank" class="ad-area-dynamic">

<img src="https://images.unsplash.com/photo-1517457210959-fc3d97f37472?q=80&w=2070&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D" alt="TripAdvisor Reklamı - Gezgin Yorumları">

<p>Milyonlarca gezginin yorumlarını oku!</p>

</a>

<a href="https://www.rentalcars.com/" target="\_blank" class="ad-area-dynamic">

<img src="https://images.unsplash.com/photo-1549488344-f8510a26569f?q=80&w=2070&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D" alt="Rentalcars Reklamı - Araç Kiralama">

<p>Tatilinize uygun araç kiralayın!</p>

</a>

<div class="ad-area-dynamic" style="background-color: #e0f7fa; border-color: #00bcd4; color: #00796b;">

<i class="fas fa-bullhorn" style="font-size: 2em; margin-bottom: 10px; color: #0097a7;"></i>

<p>Burada sizin ilanınız olabilir! <br>İlan vermek için bize e-posta gönderin.</p>

<a href="mailto:info@tatilkaptani.com" style="color: #00796b; font-weight: bold;">info@tatilkaptani.com</a>

</div>

</div>

</div>

</div>

<footer>

<p>&copy; <span id="currentYear"></span> TatilKaptani.com - Tüm Hakları Saklıdır.</p>

</footer>

<script>

// Firebase SDK v9.6.10 uyumlu versiyonlar için Firebase modüllerini doğrudan kullanın.

// Global Değişkenler ve API Anahtarları

// KENDİ ANAHTARLARINIZI BURAYA YAPIŞTIRIN!

const OPENROUTER\_API\_KEY = "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";

const GOOGLE\_GEMINI\_API\_KEY = "AIzaSyCmO--ZvTiky4UgwUqIUFj1iizh2IC4TfY";

const IMAGE\_DOWNLOAD\_COST\_PER\_IMAGE = 50;

const IMAGE\_GENERATION\_MODEL = "gemini-pro-vision";

const VIRTUAL\_TOUR\_COST\_PER\_MINUTE = 10;

const VIP\_PLAN\_CHAT\_COST = 10;

// Firebase yapılandırması - KENDİ BİLGİLERİNİZİ GİRİN!

const firebaseConfig = {

apiKey: "YOUR\_FIREBASE\_CLIENT\_API\_KEY",

authDomain: "YOUR\_PROJECT\_ID.firebaseapp.com",

projectId: "YOUR\_PROJECT\_ID",

storageBucket: "YOUR\_PROJECT\_ID.appspot.com",

messagingSenderId: "YOUR\_MESSAGING\_SENDER\_ID",

appId: "YOUR\_APP\_ID"

};

// Firebase'i başlat

firebase.initializeApp(firebaseConfig);

const auth = firebase.auth();

const firestore = firebase.firestore();

let currentUserId = null;

let voiceEnabled = true;

let tatilPuan = 0;

let userMembershipLevel = "Bronz";

let userName = "Misafir";

let userEmail = "Ayarlanmadı";

let gameActive = false;

let currentQuestionIndex = 0;

let gameScore = 0;

let currentGeneratedImages = []; // Son oluşturulan fotoğrafların URL'lerini tutmak için

let generatedVirtualImageUrl = ''; // Sanal tur hediye fotoğrafının URL'sini tutmak için

let currentVipPlan = ""; // VIP planını saklamak için

let palmCoinHistory = []; // PalmCoin geçmişi için dizi

const sloganList = [

"Hayalindeki tatili Palmiye Kaptan'la keşfet!",

"Yapay zekâyla tatilin haritasını çiz!",

"Palmiye Kaptan seni maceraya çağırıyor!",

"TatilPuan kazan, üyelik seviyeni yükselt!",

"Sanal ya da gerçek, her tatil özel!",

"Yeni rotalar, yeni maceralar Palmiye Kaptan'la başlar!"

];

let currentSloganIndex = 0;

let currentGameQuestion = null; // Aktif soruyu ve cevabı tutar

let aiCompanion = null;

let companionChatHistory = []; // AI Yoldaş sohbet geçmişi

// --- DOM Elementleri ---

// Bu elementler orijinal HTML'inizde tanımlı değildi, bu yüzden null olabilirler.

// Eğer header kısmında login/register/logout butonları olacaksa HTML'e eklenmeli.

const loginBtn = document.getElementById('loginBtn'); // Header'daki giriş butonu

const registerBtn = document.getElementById('registerBtn'); // Header'daki kayıt butonu

const logoutBtn = document.getElementById('logoutBtn'); // Header'daki çıkış butonu

const authButtons = document.getElementById('authButtons'); // Header'daki giriş/kayıt butonlarını içeren div

const loggedInUserSection = document.getElementById('loggedInUser'); // Header'daki oturum açmış kullanıcı bilgilerini içeren div

const usernameDisplay = document.getElementById('usernameDisplay'); // Oturum açmış kullanıcı adı display

// Sohbet Asistanı

const chatBox = document.getElementById("chat-box");

const chatInput = document.getElementById("user-input-chat");

const sendChatBtn = document.getElementById("send-button-chat");

const chatLoading = document.getElementById("chat-loading");

const tatilpuanTop = document.getElementById("tatilpuan-top");

const sloganTop = document.getElementById("slogan-top");

const voiceToggleTop = document.getElementById("voice-toggle-top");

const languageSelect = document.getElementById("language-select");

const avatarEl = document.getElementById("avatar");

const userIdDisplay = document.getElementById("user-id-display");

const sidebarButtons = document.querySelectorAll(".sidebar-nav button");

const contentSections = document.querySelectorAll(".content-section");

// Modallar

const appModal = document.getElementById("appModal");

const modalTitle = document.getElementById("modalTitle");

const modalMessage = document.getElementById("modalMessage");

const modalConfirmBtn = document.getElementById("modalConfirmBtn");

// Giriş/Kayıt Modalları

const loginModal = document.getElementById('loginModal');

const registerModal = document.getElementById('registerModal');

const forgotPasswordModal = document.getElementById('forgotPasswordModal');

const loginEmailInput = document.getElementById('loginEmail');

const loginPasswordInput = document.getElementById('loginPassword');

const performLoginBtn = document.getElementById('performLoginBtn');

const loginMessage = document.getElementById('loginMessage');

const registerUsernameInput = document.getElementById('registerUsername');

const registerEmailInput = document.getElementById('registerEmail');

const registerPasswordInput = document.getElementById('registerPassword');

const performRegisterBtn = document.getElementById('performRegisterBtn');

const registerMessage = document.getElementById('registerMessage');

const resetEmailInput = document.getElementById('resetEmail');

const performResetBtn = document.getElementById('performResetBtn');

const resetMessage = document.getElementById('resetMessage');

const forgotPasswordLink = document.getElementById('forgotPasswordLink');

const closeButtons = document.querySelectorAll('.close-button');

// Tatil Avı (Oyun)

const startGameBtn = document.getElementById("start-game-btn");

const gameOutput = document.getElementById("game-output");

const gameAnswerInput = document.getElementById("game-answer-input");

const submitGameAnswerBtn = document.getElementById("submit-game-answer-btn");

// Sanal Tatil Planı

const virtualCityInput = document.getElementById("virtual-city");

const virtualDaysInput = document.getElementById("virtual-days");

const virtualDurationMinutesInput = document.getElementById("virtual-duration-minutes");

const virtualActivitiesInput = document.getElementById("virtual-activities");

const virtualImagePromptInput = document.getElementById("virtual-image-prompt");

const virtualTourCostEl = document.getElementById("virtual-tour-cost");

const startVirtualBtn = document.getElementById("start-virtual-btn");

const virtualHolidayOutput = document.getElementById("virtual-holiday-output");

const virtualOutputTitle = document.getElementById("virtual-output-title");

const virtualOutputStory = document.getElementById("virtual-output-story");

const virtualImagesContainer = document.getElementById("virtual-images-container");

const sendVirtualImageEmailBtn = document.getElementById("send-virtual-image-email-btn");

const virtualLoading = document.getElementById("virtual-loading");

// AI Fotoğraf Stüdyosu

const aiPhotoAccessCheck = document.getElementById("ai-photo-access-check");

const goToAiPhotoPaymentBtn = document.getElementById("go-to-ai-photo-payment-btn");

const aiPhotoFormArea = document.getElementById("ai-photo-form-area");

const aiPhotoPromptInput = document.getElementById("ai-photo-prompt");

const aiPhotoStyleSelect = document.getElementById("ai-photo-style");

const aiPhotoCountInput = document.getElementById("ai-photo-count");

const generateAiPhotoButton = document.getElementById("generate-ai-photo-btn");

const aiPhotoLoading = document.getElementById("ai-photo-loading");

const aiPhotoOutput = document.getElementById("ai-photo-output");

const generatedImagesContainer = document.getElementById("generated-images-container");

const downloadAllImagesBtn = document.getElementById("download-all-images-btn");

const downloadAllCostSpan = document.getElementById("download-all-cost");

// VIP Planlayıcı

const vipAccessCheck = document.getElementById("vip-access-check");

const goToVipPaymentBtn = document.getElementById("go-to-vip-payment-btn");

const vipPlannerFormArea = document.getElementById("vip-planner-form-area");

const vipDestinationInput = document.getElementById("vip-destination");

const vipDurationInput = document.getElementById("vip-duration");

const vipTravelersInput = document.getElementById("vip-travelers");

const vipBudgetButtons = document.querySelectorAll("#vip-planner-section .budget-options button");

const vipTypeSelect = document.getElementById("vip-type");

const generateVipPlanBtn = document.getElementById("generate-vip-plan-btn");

const vipPlannerLoading = document.getElementById("vip-planner-loading");

const vipPlanOutput = document.getElementById("vip-plan-output");

let selectedBudget = "";

const vipPlanChatArea = document.getElementById("vip-plan-chat-area");

const vipPlanChatBox = document.getElementById("vip-plan-chat-box");

const vipPlanInput = document.getElementById("vip-plan-input");

const sendVipPlanMessageBtn = document.getElementById("send-vip-plan-message-btn");

// Niş Tur Talebi

const nicheTopicInput = document.getElementById("niche-topic");

const nicheDetailsTextarea = document.getElementById("niche-details");

const generateNichePlanBtn = document.getElementById("generate-niche-plan-btn");

const nichePlanLoading = document.getElementById("niche-plan-loading");

const nichePlanOutput = document.getElementById("niche-plan-output");

// Kullanıcı Bilgileri

const displayUserId = document.getElementById("display-userid");

const displayUsername = document.getElementById("display-username");

const displayUserEmail = document.getElementById("display-user-email");

const setuserEmailBtn = document.getElementById("set-user-email-btn");

const displayTatilpuan = document.getElementById("display-tatilpuan");

const displayMembershipLevel = document.getElementById("display-membership-level");

const displayGameScore = document.getElementById("display-game-score");

const registerUserBtn = document.getElementById("register-user-btn");

const palmCoinHistoryList = document.getElementById("palmcoin-history-list");

// Yönetici Mesajı

const adminMessageInput = document.getElementById("admin-message-input");

const updateAdminMessageBtn = document.getElementById("update-admin-message-btn");

const adminMessageLoading = document.getElementById("admin-message-loading");

// Zamanda Yolculuk

const timeTravelAccessCheck = document.getElementById("time-travel-access-check");

const goToTimeTravelPaymentBtn = document.getElementById("go-to-time-travel-payment-btn");

const timeTravelFormArea = document.getElementById("time-travel-form-area");

const timeTravelEraInput = document.getElementById("time-travel-era");

const timeTravelDurationInput = document.getElementById("time-travel-duration");

const timeTravelCharacterInput = document.getElementById("time-travel-character");

const timeTravelFocusInput = document.getElementById("time-travel-focus");

const startTimeTravelBtn = document.getElementById("start-time-travel-btn");

const timeTravelLoading = document.getElementById("time-travel-loading");

const timeTravelOutput = document.getElementById("time-travel-output");

// Kader Rotası

const destinyAgeInput = document.getElementById("destiny-age");

const destinyHobbyInput = document.getElementById("destiny-hobby");

const destinyDreamInput = document.getElementById("destiny-dream");

const destinyColorInput = document.getElementById("destiny-color");

const predictDestinyBtn = document.getElementById("predict-destiny-btn");

const destinyLoading = document.getElementById("destiny-loading");

const destinyRouteOutput = document.getElementById("destiny-route-output");

const realizeDestinyBtn = document.getElementById("realize-destiny-btn");

// AI Yoldaşım

const companionNameInput = document.getElementById("companion-name");

const companionPersonalitySelect = document.getElementById("companion-personality");

const createCompanionBtn = document.getElementById("create-companion-btn");

const companionChatArea = document.getElementById("companion-chat-area");

const activeCompanionName = document.getElementById("active-companion-name");

const companionChatBox = document.getElementById("companion-chat-box");

const companionInput = document.getElementById("companion-input");

const sendCompanionMessageBtn = document.getElementById("send-companion-message-btn");

const companionLoading = document.getElementById("companion-loading");

// `aiCompanion` ve `companionChatHistory` zaten global olarak tanımlı.

// Ödeme

const cardNumberInput = document.getElementById("card-number");

const expiryDateInput = document.getElementById("expiry-date");

const cvvInput = document.getElementById("cvv");

const cardHolderNameInput = document.getElementById("card-holder-name");

const completePaymentBtn = document.getElementById("complete-payment-btn");

// Bize Ulaşın Bölümü

const contactSubjectInput = document.getElementById('contact-subject');

const contactEmailInput = document.getElementById('contact-email');

const contactMessageInput = document.getElementById('contact-message');

const contactFileInput = document.getElementById('contact-file');

const sendContactFormBtn = document.getElementById('send-contact-form-btn');

const contactLoading = document.getElementById('contact-loading');

// --- Uygulama Başlangıcı ve Genel Fonksiyonlar ---

window.initializeAppFeatures = function() {

const sloganTopEl = document.getElementById("slogan-top");

if (sloganTopEl) {

sloganTopEl.innerText = sloganList[Math.floor(Math.random() \* sloganList.length)];

}

// Firebase onAuthStateChanged listener'ı kullanıcı durumunu yönetecek

// İlk karşılama mesajı modal aracılığıyla gösterilecek.

};

// Modalları gizlemek için yardımcı fonksiyon

function hideModal(modalElement) {

if (modalElement) {

modalElement.style.display = "none";

}

}

// --- Firebase Authentication ve Kullanıcı Durumu ---

auth.onAuthStateChanged(async (user) => {

if (user) {

currentUserId = user.uid;

// Eğer bu butonlar HTML'de yoksa, null kontrolü ekledim.

if (authButtons) authButtons.style.display = 'none';

if (loggedInUserSection) loggedInUserSection.style.display = 'flex';

if (usernameDisplay) usernameDisplay.textContent = user.displayName || user.email;

const mainLayout = document.querySelector('.main-layout');

if (mainLayout) {

mainLayout.style.display = 'flex'; // Giriş yapılınca ana düzeni göster

}

window.showSection("chat-section");

const chatSectionButton = document.querySelector(".sidebar-nav button[data-section='chat-section']");

if (chatSectionButton) {

chatSectionButton.classList.add("active");

}

loadUserProfile();

loadAdminMessage();

loadAds();

} else {

currentUserId = null;

if (authButtons) authButtons.style.display = 'flex';

if (loggedInUserSection) loggedInUserSection.style.display = 'none';

const mainLayout = document.querySelector('.main-layout');

if (mainLayout) {

mainLayout.style.display = 'none'; // Çıkış yapılınca veya ilk açılışta ana düzeni gizle

}

// Kullanıcı giriş yapmamışsa login modalını göster

if (loginModal) loginModal.style.display = 'flex'; // loginModal'ı direkt göster

window.showModal("Hoş Geldiniz", "Lütfen sohbet etmek ve diğer özelliklere erişmek için giriş yapın veya üye olun."); // App modalını da göster

}

});

// --- Firebase Kullanıcı Verileri Fonksiyonları ---

window.getUserProfileRef = function() {

if (!firestore || !currentUserId) {

console.error("Firestore veya User ID hazır değil.");

return null;

}

return firestore.collection('users').doc(currentUserId).collection('profile').doc("user-data");

}

window.loadUserProfile = async function() {

const profileRef = window.getUserProfileRef();

if (!profileRef) {

console.log("Profil referansı mevcut değil, varsayılan bilgiler gösteriliyor.");

window.displayMembershipInfo();

window.updateTatilPuanDisplay();

if (userIdDisplay) userIdDisplay.textContent = `UID: ${currentUserId || 'Misafir'}`;

return;

}

firestore.collection('users').doc(currentUserId).collection('profile').doc("user-data")

.onSnapshot((docSnap) => {

if (docSnap.exists) {

const data = docSnap.data();

userName = data.userName || "Misafir";

tatilPuan = data.tatilPuan || 0;

userMembershipLevel = data.userMembershipLevel || "Bronz";

gameScore = data.gameScore || 0;

userEmail = data.userEmail || "Ayarlanmadı";

palmCoinHistory = data.palmCoinHistory || [{ timestamp: new Date().toISOString(), type: "Başlangıç", amount: 0, current: 0 }];

console.log("Kullanıcı profili Firestore'dan yüklendi:", data);

} else {

console.log("Kullanıcı profili bulunamadı, varsayılan oluşturuluyor.");

window.updateUserProfile({

userName: "Misafir",

tatilPuan: 0,

userMembershipLevel: "Bronz",

gameScore: 0,

userEmail: "Ayarlanmadı",

palmCoinHistory: [{ timestamp: new Date().toISOString(), type: "Başlangıç", amount: 0, current: 0 }]

});

}

window.displayMembershipInfo();

window.updateTatilPuanDisplay();

window.updatePalmCoinHistoryDisplay();

if (userIdDisplay) userIdDisplay.textContent = `UID: ${currentUserId || 'Misafir'}`;

}, (error) => {

console.error("Kullanıcı profili yüklenirken hata:", error);

window.showModal("Hata", `Kullanıcı verileri yüklenirken bir sorun oluştu: ${error.message}.`);

});

}

window.updateUserProfile = async function(dataToUpdate) {

const profileRef = window.getUserProfileRef();

if (!profileRef) {

console.error("Profil referansı mevcut değil, güncelleme yapılamıyor.");

return;

}

try {

await profileRef.set(dataToUpdate, { merge: true });

console.log("Kullanıcı profili güncellendi:", dataToUpdate);

} catch (error) {

console.error("Kullanıcı profili güncellenirken hata:", error);

window.showModal("Hata", `Kullanıcı verileri kaydedilirken bir sorun oluştu: ${error.message}.`);

}

}

// --- Firebase İlan ve Yönetici Mesajı Yönetimi Fonksiyonları ---

window.getAdsCollectionRef = function() {

if (!firestore) {

console.error("Firestore hazır değil.");

return null;

}

return firestore.collection(`public/data/ads`);

}

window.getAdminMessageRef = function() {

if (!firestore) {

console.error("Firestore hazır değil.");

return null;

}

return firestore.collection(`public/data/admin`).doc("message");

}

window.loadAds = async function() {

const adsCollectionRef = window.getAdsCollectionRef();

if (!adsCollectionRef) {

console.log("Reklam koleksiyonu referansı mevcut değil.");

return;

}

// Eğer Firestore'dan dinamik reklamları çekmek isterseniz burayı ayarlamanız gerekir.

// Şu anda HTML'deki statik reklamlar gösteriliyor.

}

window.loadAdminMessage = async function() {

const adminMessageRef = window.getAdminMessageRef();

let adminDisplayMessageEl = document.getElementById("admin-display-message");

if (!adminMessageRef) {

console.log("Yönetici mesajı referansı mevcut değil.");

if (adminDisplayMessageEl && adminDisplayMessageEl.querySelector('p')) {

adminDisplayMessageEl.querySelector('p').textContent = "Yönetici mesajı yüklenemedi: Veri hazır değil.";

}

return;

}

adminMessageRef.onSnapshot((docSnap) => {

adminDisplayMessageEl = document.getElementById("admin-display-message");

if (adminDisplayMessageEl) {

const adminDisplayMessageP = adminDisplayMessageEl.querySelector('p');

if (adminDisplayMessageP) {

if (docSnap.exists && docSnap.data().message) {

adminDisplayMessageP.textContent = docSnap.data().message;

} else {

adminDisplayMessageP.textContent = "Yönetici mesajı bulunmamaktadır.";

}

}

}

}, (error) => {

console.error("Yönetici mesajı yüklenirken hata:", error);

adminDisplayMessageEl = document.getElementById("admin-display-message");

if (adminDisplayMessageEl && adminDisplayMessageEl.querySelector('p')) {

let errorMessage = `Yönetici mesajı yüklenemedi: ${error.message}.`;

if (error.code === 'permission-denied') {

errorMessage += " Lütfen Firebase güvenlik kurallarınızı kontrol edin (public/data/admin okuma izni).";

}

adminDisplayMessageEl.querySelector('p').textContent = errorMessage;

}

});

}

window.updateAdminMessage = async function(message) {

const adminMessageRef = window.getAdminMessageRef();

const adminMessageLoadingEl = document.getElementById("admin-message-loading");

if (!adminMessageRef) {

console.error("Yönetici mesajı referansı mevcut değil, güncelleme yapılamıyor.");

window.showModal("Hata", "Yönetici mesajı özelliği şu an kullanılamıyor. Firebase verileri hazır değil.");

return;

}

if (adminMessageLoadingEl) adminMessageLoadingEl.style.display = 'block';

try {

await adminMessageRef.set({ message: message, lastUpdated: firebase.firestore.FieldValue.serverTimestamp() }, { merge: true });

console.log("Yönetici mesajı başarıyla güncellendi!");

window.showModal("Başarılı", "Yönetici mesajı başarıyla güncellendi.");

} catch (error) {

console.error("Yönetici mesajı güncellenirken hata:", error);

let errorMessage = `Yönetici mesajı güncellenirken bir sorun oluştu: ${error.message}.`;

if (error.code === 'permission-denied') {

errorMessage += " Lütfen Firebase güvenlik kurallarınızı kontrol edin (public/data/admin yazma izni).";

}

window.showModal("Hata", errorMessage);

} finally {

if (adminMessageLoadingEl) adminMessageLoadingEl.style.display = 'none';

}

}

// --- Genel UI ve Yardımcı Fonksiyonlar ---

window.showModal = function(title, message) {

const modal = document.getElementById("appModal");

const modalTitleEl = document.getElementById("modalTitle");

const modalMessageEl = document.getElementById("modalMessage");

const modalConfirmBtnEl = document.getElementById("modalConfirmBtn");

if (!modal || !modalTitleEl || !modalMessageEl || !modalConfirmBtnEl) {

console.error("Modal elementleri bulunamadı. Modal gösterilemiyor.");

return Promise.resolve(false);

}

modalTitleEl.textContent = title;

modalMessageEl.innerHTML = message;

modal.style.display = "flex";

// Önceki event listener'ı kaldır

const oldConfirmListener = modalConfirmBtnEl.\_eventListener;

if (oldConfirmListener) {

modalConfirmBtnEl.removeEventListener("click", oldConfirmListener);

}

return new Promise((resolve) => {

const handleConfirm = () => {

modal.style.display = "none";

modalConfirmBtnEl.removeEventListener("click", handleConfirm);

modalConfirmBtnEl.\_eventListener = null; // Bellek sızıntısını önle

resolve(true);

};

modalConfirmBtnEl.addEventListener("click", handleConfirm);

modalConfirmBtnEl.\_eventListener = handleConfirm; // Listener'ı sakla

});

}

// `hideModal` fonksiyonunu global scope'a taşıdım

window.hideModal = function(modalElement) {

if (modalElement) {

modalElement.style.display = "none";

}

}

window.displayMessage = function(sender, text, chatBoxElement = chatBox) { // Varsayılan olarak chatBox kullanılır

if (!chatBoxElement) {

console.error("Sohbet kutusu elementi bulunamadı.");

return;

}

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

messageDiv.classList.add("message", sender);

const span = document.createElement('span');

span.textContent = text;

messageDiv.appendChild(span);

chatBoxElement.appendChild(messageDiv);

chatBoxElement.scrollTop = chatBoxElement.scrollHeight;

}

window.speak = function(text) {

const languageSelectEl = document.getElementById("language-select");

if (!languageSelectEl || !voiceEnabled || !window.speechSynthesis) return;

const utterance = new SpeechSynthesisUtterance(text);

utterance.lang = languageSelectEl.value + "-" + languageSelectEl.value.toUpperCase();

speechSynthesis.speak(utterance);

}

window.updateTatilPuan = async function(points, activity = "Genel Aktivite") {

const oldTatilPuan = tatilPuan;

tatilPuan += points;

palmCoinHistory.push({

timestamp: new Date().toISOString(),

type: points > 0 ? "Kazanıldı" : "Harcandı",

activity: activity,

amount: Math.abs(points),

current: tatilPuan

});

if (palmCoinHistory.length > 20) {

palmCoinHistory = palmCoinHistory.slice(palmCoinHistory.length - 20);

}

const oldLevel = userMembershipLevel;

if (tatilPuan < 100) {

userMembershipLevel = "Bronz";

} else if (tatilPuan < 200) {

userMembershipLevel = "Gümüş";

} else {

userMembershipLevel = "Altın";

}

await window.updateUserProfile({ tatilPuan: tatilPuan, userMembershipLevel: userMembershipLevel, palmCoinHistory: palmCoinHistory });

if (oldLevel !== userMembershipLevel) {

window.showModal("Tebrikler!", `Üyelik seviyeniz \*\*${userMembershipLevel}\*\* seviyesine yükseldi! Yeni özelliklere göz atın.`);

window.speak(`Tebrikler! Üyelik seviyeniz ${userMembershipLevel} seviyesine yükseldi!`);

}

window.updatePalmCoinHistoryDisplay();

}

window.updateTatilPuanDisplay = function() {

if (tatilpuanTop) {

tatilpuanTop.innerHTML = `<i class="fas fa-star"></i> TatilPuan: ${tatilPuan} (${userMembershipLevel})`;

}

}

window.displayMembershipInfo = function() {

if (displayUserId) displayUserId.textContent = currentUserId || "Yükleniyor...";

if (displayUsername) displayUsername.textContent = userName;

if (displayUserEmail) displayUserEmail.textContent = userEmail;

if (displayTatilpuan) displayTatilpuan.textContent = tatilPuan;

if (displayMembershipLevel) displayMembershipLevel.textContent = userMembershipLevel;

if (displayGameScore) displayGameScore.textContent = gameScore;

if (userIdDisplay) userIdDisplay.textContent = `UID: ${currentUserId || 'Misafir'}`;

}

window.updatePalmCoinHistoryDisplay = function() {

if (!palmCoinHistoryList) return;

palmCoinHistoryList.innerHTML = '';

palmCoinHistory.slice().reverse().forEach(entry => {

const listItem = document.createElement("li");

const date = new Date(entry.timestamp).toLocaleString();

let color = '#333';

if (entry.type === "Kazanıldı") {

color = 'green';

} else if (entry.type === "Harcandı") {

color = 'red';

}

listItem.innerHTML = `<span style="color:${color};"><strong>${entry.type}:</strong> ${entry.amount} PalmCoin</span> - ${entry.activity} (${date}) - Toplam: ${entry.current}`;

palmCoinHistoryList.appendChild(listItem);

});

}

window.showSection = function(sectionId) {

contentSections.forEach(section => {

section.classList.remove("active");

section.style.display = "none";

});

const activeSection = document.getElementById(sectionId);

if (activeSection) {

activeSection.classList.add("active");

activeSection.style.display = "flex";

}

sidebarButtons.forEach(button => button.classList.remove("active"));

const activeButton = document.querySelector(`.sidebar-nav button[data-section='${sectionId}']`);

if (activeButton) {

activeButton.classList.add("active");

}

// Özel bölüm kontrolleri

if (sectionId === "vip-planner-section") {

window.checkVipAccess(document.getElementById("vip-access-check"), document.getElementById("vip-planner-form-area"), "Altın");

window.checkVipAccess(document.getElementById("vip-access-check"), document.getElementById("niche-vip-request-area"), "Altın");

if (vipPlanChatArea) vipPlanChatArea.style.display = 'none';

if (vipPlanOutput) vipPlanOutput.style.display = 'none';

currentVipPlan = "";

} else if (sectionId === "time-travel-section") {

window.checkVipAccess(document.getElementById("time-travel-access-check"), document.getElementById("time-travel-form-area"), "Altın");

} else if (sectionId === "ai-photo-studio-section") {

window.checkVipAccess(document.getElementById("ai-photo-access-check"), document.getElementById("ai-photo-form-area"), "Altın");

if (generatedImagesContainer) generatedImagesContainer.innerHTML = '';

if (downloadAllImagesBtn) downloadAllImagesBtn.style.display = 'none';

if (aiPhotoOutput) aiPhotoOutput.style.display = 'none';

currentGeneratedImages = [];

} else if (sectionId === "user-info-section") {

window.displayMembershipInfo();

window.loadAdminMessage();

window.updatePalmCoinHistoryDisplay();

} else if (sectionId === "payment-section") {

if (cardNumberInput) cardNumberInput.value = '';

if (expiryDateInput) expiryDateInput.value = '';

if (cvvInput) cvvInput.value = '';

if (cardHolderNameInput) cardHolderNameInput.value = '';

} else if (sectionId === "virtual-holiday-section") {

if (virtualDurationMinutesInput && virtualTourCostEl) {

virtualTourCostEl.textContent = (parseInt(virtualDurationMinutesInput.value) \* VIRTUAL\_TOUR\_COST\_PER\_MINUTE);

}

if(virtualHolidayOutput) virtualHolidayOutput.style.display = "none";

if(virtualImagesContainer) virtualImagesContainer.innerHTML = '';

if(sendVirtualImageEmailBtn) sendVirtualImageEmailBtn.style.display = 'none';

generatedVirtualImageUrl = '';

} else if (sectionId === "destiny-route-section") {

if(destinyRouteOutput) destinyRouteOutput.style.display = "none";

if(realizeDestinyBtn) realizeDestinyBtn.style.display = "none";

}

// "Bize Ulaşın" bölümü için temizlik (varsa)

if (sectionId === "contact-us-section") {

if (contactSubjectInput) contactSubjectInput.value = '';

if (contactEmailInput) contactEmailInput.value = userEmail !== "Ayarlanmadı" ? userEmail : '';

if (contactMessageInput) contactMessageInput.value = '';

if (contactFileInput) contactFileInput.value = ''; // Dosya inputunu temizle

}

}

window.checkVipAccess = function(accessCheckEl, formAreaEl, requiredLevel) {

const levels = ["Bronz", "Gümüş", "Altın"];

const currentUserLevelIndex = levels.indexOf(userMembershipLevel);

const requiredLevelIndex = levels.indexOf(requiredLevel);

if (currentUserLevelIndex >= requiredLevelIndex) {

if (accessCheckEl) accessCheckEl.style.display = "none";

if (formAreaEl) formAreaEl.style.display = "block";

} else {

if (accessCheckEl) accessCheckEl.style.display = "block";

if (formAreaEl) formAreaEl.style.display = "none";

}

}

// OpenRouter AI Çağrısı (Metin Tabanlı AI için)

window.callOpenRouterAI = async function(prompt, model = "openai/gpt-3.5-turbo", loadingIndicator = null, chatHistory = []) {

if (loadingIndicator) loadingIndicator.style.display = "block";

const messages = chatHistory.length > 0 ? chatHistory : [{ role: "user", content: prompt }];

// Eğer chatHistory boşsa ve prompt varsa, prompt'u ilk mesaj olarak ayarla

if (chatHistory.length === 0 && prompt) {

messages[0] = { role: "user", content: prompt };

}

try {

const response = await fetch("https://openrouter.ai/api/v1/chat/completions", {

method: "POST",

headers: {

"Authorization": `Bearer ${OPENROUTER\_API\_KEY}`,

"Content-Type": "application/json"

},

body: JSON.stringify({

model: model,

messages: messages

})

});

if (!response.ok) {

const errorData = await response.json();

throw new Error(`OpenRouter API Hatası: ${response.status} - ${errorData.message || 'Bilinmeyen Hata'}`);

}

const data = await response.json();

// Türkçe karakter hatası için düzeltme (API'den dönen yanıtın kodlamasına bağlı olabilir, burada varsayılan olarak UTF-8 bekliyoruz)

return data?.choices?.[0]?.message?.content || "Üzgünüm, şu an cevap alamıyorum.";

} catch (error) {

console.error("OpenRouter AI Cevap Hatası:", error);

window.showModal("AI Hatası!", `Bir OpenRouter AI hatası oluştu: ${error.message}. Lütfen API anahtarınızı veya internet bağlantınızı kontrol edin.`);

return `Bir hata oluştu: ${error.message}`;

} finally {

if (loadingIndicator) loadingIndicator.style.display = "none";

}

}

// Google Gemini API Çağrısı (Görsel Oluşturma için)

window.callImageGenerationAI = async function(promptText, loadingIndicator = null) {

if (loadingIndicator) loadingIndicator.style.display = "block";

// Gemini API'nin image generation endpoint'i ve anahtar kullanımı

// Bu URL'yi doğru endpoint ve anahtar ile kullanın.

const apiUrl = `https://generativelanguage.googleapis.com/v1beta/models/gemini-pro-vision:generateContent?key=${GOOGLE\_GEMINI\_API\_KEY}`;

try {

const payload = {

contents: [

{

parts: [

{ text: promptText }

]

}

]

};

const response = await fetch(apiUrl, {

method: 'POST',

headers: { 'Content-Type': 'application/json' },

body: JSON.stringify(payload)

});

if (!response.ok) {

const errorData = await response.json();

throw new Error(`Gemini Görsel Oluşturma Hatası: ${response.status} - ${errorData.error.message || 'Bilinmeyen Hata'}`);

}

const result = await response.json();

if (result.candidates && result.candidates[0] && result.candidates[0].content && result.candidates[0].content.parts && result.candidates[0].content.parts[0].inlineData) {

return `data:${result.candidates[0].content.parts[0].inlineData.mimeType};base64,${result.candidates[0].content.parts[0].inlineData.data}`;

} else {

throw new Error("Görsel oluşturma başarısız oldu: Yanıt yapısı beklenmedik veya resim verisi bulunamadı.");

}

} catch (error) {

console.error("Gemini Görsel Oluşturma AI Hatası:", error);

window.showModal("Görsel Oluşturma Hatası!", `Görsel oluşturulurken bir sorun oluştu: ${error.message}. Gemini API anahtarınızı ve yetkilendirmeyi kontrol edin.`);

return null;

} finally {

if (loadingIndicator) loadingIndicator.style.display = "none";

}

}

// --- Event Listeners ---

document.addEventListener('DOMContentLoaded', async () => {

// Uygulama özelliklerini DOM hazır olduğunda başlat

window.initializeAppFeatures();

// Firebase Auth durumu, uygulamanın ana UI'sını kontrol edecek

// Bu kısım, sayfa yüklendiğinde kullanıcı oturum durumunu kontrol eder.

// Auth state changed listener'ı yukarı taşıdığım için burada tekrar etmiyorum.

sidebarButtons.forEach(button => {

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

const sectionId = button.dataset.section;

window.showSection(sectionId);

});

});

// Ana sohbet asistanı

sendChatBtn.onclick = async () => {

const userMessage = chatInput.value.trim();

if (!userMessage) return;

window.displayMessage("user", userMessage, chatBox);

chatInput.value = "";

const selectedLanguage = languageSelect.value; // Seçilen dili al

const languagePrompt = selectedLanguage === 'tr' ? '(Türkçe)' : '(English)'; // AI'ya iletmek için

const prompt = `Kullanıcının mesajı: "${userMessage}". Bu bir tatil asistanı sohbeti. Mesajın dilini (${languagePrompt}) algıla ve o dilde yanıt ver.

Tatil, seyahat, destinasyonlar, konaklama, aktiviteler, bütçe planlaması, sanal turlar, tarihi yerler ve kültürel deneyimler gibi konularda yaratıcı, bilgilendirici ve kişiselleştirilmiş bir şekilde yardımcı ol.

Tekrarlayan veya jenerik cevaplardan kaçın. Kullanıcının önceki mesajlarını dikkate alarak bağlamı anla ve sohbeti ileri taşı.

Eğer kullanıcının mesajı bir özellik talebi (örneğin 'oyun oynamak istiyorum', 'tatil planı yap', 'resim oluştur', 'VIP olmak istiyorum', 'e-posta ayarla', 'yönetici mesajı ayarla') içeriyorsa, yanıtının sonuna "[YÖNLENDİR: [bölüm-adı]]" şeklinde gizli bir etiket ekle.

Örnek: "Elbette, size özel bir tatil planı oluşturabilirim! [YÖNLENDİR: vip-planner-section]".

Bölüm adları: game-section, virtual-holiday-section, ai-photo-studio-section, vip-planner-section, user-info-section, time-travel-section, destiny-route-section, ai-companion-section, payment-section, contact-us-section.`;

const reply = await window.callOpenRouterAI(prompt, "openai/gpt-3.5-turbo", chatLoading);

const redirectMatch = reply.match(/\[YÖNLENDİR:\s\*([^\]]+)\]/);

let cleanReply = reply.replace(/\[YÖNLENDİR:\s\*([^\]]+)\]/, '').trim();

window.displayMessage("ai", cleanReply, chatBox);

window.speak(cleanReply);

window.updateTatilPuan(5, "Sohbet");

if (currentUserId) {

await firestore.collection('users').doc(currentUserId).collection('chatHistory').add({

userMessage: userMessage,

aiReply: cleanReply,

timestamp: firebase.firestore.FieldValue.serverTimestamp()

});

}

if (redirectMatch && redirectMatch[1]) {

const targetSectionId = redirectMatch[1].trim();

const sectionNameMap = {

"game-section": "Tatil Avı (Oyun)",

"virtual-holiday-section": "Sanal Tatil Planı",

"ai-photo-studio-section": "AI Fotoğraf Stüdyosu",

"vip-planner-section": "VIP A'dan Z'ye Tur Planlayıcı",

"user-info-section": "Üyelik Bilgileri",

"time-travel-section": "Zamanda Yolculuk Tatili",

"destiny-route-section": "Kader Rotası",

"ai-companion-section": "AI Yoldaşım",

"payment-section": "VIP Üyelik Al",

"contact-us-section": "Bize Ulaşın"

};

const friendlySectionName = sectionNameMap[targetSectionId] || targetSectionId;

window.showModal(

"Yönlendirme Önerisi",

`Palmiye Kaptan, sanırım \*\*${friendlySectionName}\*\* bölümüyle ilgileniyorsunuz. Oraya gitmek ister misiniz?` +

`<br><br><button id="confirmRedirectBtn" style="background-color:#00796b; color:white; padding:10px 20px; border:none; border-radius:5px; cursor:pointer;">Evet, Git!</button>` +

`<button id="cancelRedirectBtn" style="background-color:#ccc; color:#333; padding:10px 20px; border:none; border-radius:5px; cursor:pointer; margin-left: 10px;">Hayır, Burada Kal</button>`

).then(() => { /\* Modal kapatıldığında Promise çözülür \*/ });

const confirmRedirectBtn = document.getElementById("confirmRedirectBtn");

const cancelRedirectBtn = document.getElementById("cancelRedirectBtn");

if (confirmRedirectBtn) {

confirmRedirectBtn.onclick = () => {

window.showSection(targetSectionId);

appModal.style.display = "none";

};

}

if (cancelRedirectBtn) {

cancelRedirectBtn.onclick = () => {

appModal.style.display = "none";

};

}

}

};

chatInput.addEventListener("keypress", (e) => {

if (e.key === "Enter") sendChatBtn.click();

});

// Ses kontrolü

voiceToggleTop.onclick = () => {

const newState = !voiceEnabled;

voiceEnabled = newState;

voiceToggleTop.innerHTML = voiceEnabled ? '<i class="fas fa-volume-up"></i> Ses: Açık' : '<i class="fas fa-volume-mute"></i> Ses: Kapalı';

if (voiceEnabled) {

window.speak("Sesli yanıtlar açıldı.");

} else {

if (window.speechSynthesis && window.speechSynthesis.speaking) {

window.speechSynthesis.cancel();

}

}

};

languageSelect.addEventListener("change", () => {

window.showModal("Dil Değişikliği", `Dil ${languageSelect.options[languageSelect.selectedIndex].text} olarak ayarlandı. AI bu dilde yanıt vermeye çalışacak.`);

window.speak(`Language changed to ${languageSelect.options[languageSelect.selectedIndex].text}.`);

});

avatarEl.addEventListener('click', () => window.showSection("user-info-section"));

modalConfirmBtn.addEventListener("click", () => appModal.style.display = "none");

// Giriş/Kayıt Modalı butonları

// Eğer header kısmında login/register/logout butonları varsa, onları bağlayalım

if (loginBtn) { // loginBtn elementinin varlığını kontrol et

loginBtn.addEventListener('click', () => {

hideModal(registerModal);

hideModal(forgotPasswordModal);

if (loginModal) loginModal.style.display = 'flex'; // loginModal'ı göster

});

}

if (registerBtn) { // registerBtn elementinin varlığını kontrol et

registerBtn.addEventListener('click', () => {

hideModal(loginModal);

hideModal(forgotPasswordModal);

if (registerModal) registerModal.style.display = 'flex'; // registerModal'ı göster

});

}

// Şifremi unuttum linki

if (forgotPasswordLink) { // forgotPasswordLink elementinin varlığını kontrol et

forgotPasswordLink.addEventListener('click', () => {

hideModal(loginModal);

if (forgotPasswordModal) forgotPasswordModal.style.display = 'flex'; // forgotPasswordModal'ı göster

});

}

// Tüm kapatma butonları için event listener

closeButtons.forEach(button => {

button.addEventListener('click', (event) => {

hideModal(event.target.closest('.modal'));

});

});

// Modal dışına tıklayınca kapatma

window.addEventListener('click', (event) => {

if (event.target == loginModal) hideModal(loginModal);

if (event.target == registerModal) hideModal(registerModal);

if (event.target == forgotPasswordModal) hideModal(forgotPasswordModal);

});

// Kayıt İşlemi

performRegisterBtn.addEventListener('click', async () => {

const username = registerUsernameInput.value;

const email = registerEmailInput.value;

const password = registerPasswordInput.value;

try {

const userCredential = await auth.createUserWithEmailAndPassword(email, password);

await userCredential.user.updateProfile({ displayName: username });

// Firestore'a başlangıç kullanıcı verilerini kaydet

await firestore.collection('users').doc(userCredential.user.uid).set({

username: username,

email: email,

createdAt: firebase.firestore.FieldValue.serverTimestamp(),

tatilPuanlari: 0,

membershipLevel: 'Temel'

});

displayMessage(registerMessage, 'Kayıt başarılı! Hoş geldiniz. Şimdi giriş yapabilirsiniz.');

sendWelcomeEmailSimulation(email, username);

setTimeout(() => {

hideModal(registerModal);

if (loginModal) loginModal.style.display = 'flex'; // loginModal'ı göster

if (loginEmailInput) loginEmailInput.value = email;

}, 2000);

} catch (error) {

let errorMessage = 'Kayıt sırasında bir hata oluştu.';

if (error.code === 'auth/email-already-in-use') {

errorMessage = 'Bu e-posta adresi zaten kullanımda.';

} else if (error.code === 'auth/weak-password') {

errorMessage = 'Şifre en az 6 karakter olmalıdır.';

}

displayMessage(registerMessage, errorMessage, true);

console.error('Kayıt hatası:', error);

}

});

// Giriş İşlemi

performLoginBtn.addEventListener('click', async () => {

const email = loginEmailInput.value;

const password = loginPasswordInput.value;

try {

await auth.signInWithEmailAndPassword(email, password);

displayMessage(loginMessage, 'Giriş başarılı! Yönlendiriliyorsunuz...');

setTimeout(() => {

hideModal(loginModal);

}, 1500);

} catch (error) {

let errorMessage = 'Giriş başarısız. Lütfen bilgilerinizi kontrol edin.';

if (error.code === 'auth/wrong-password') {

errorMessage = 'E-posta veya şifre hatalı.';

} else if (error.code === 'auth/user-not-found') {

errorMessage = 'Bu e-posta ile kayıtlı kullanıcı bulunamadı.';

}

displayMessage(loginMessage, errorMessage, true);

console.error('Giriş hatası:', error);

}

});

// Şifre Sıfırlama

performResetBtn.addEventListener('click', async () => {

const email = resetEmailInput.value;

try {

await auth.sendPasswordResetEmail(email);

displayMessage(resetMessage, 'Şifre sıfırlama linki e-posta adresinize gönderildi.');

setTimeout(() => hideModal(forgotPasswordModal), 3000);

} catch (error) {

let errorMessage = 'Şifre sıfırlama isteği gönderilirken bir hata oluştu.';

if (error.code === 'auth/user-not-found') {

errorMessage = 'Bu e-posta ile kayıtlı kullanıcı bulunamadı.';

}

displayMessage(resetMessage, errorMessage, true);

console.error('Şifre sıfırlama hatası:', error);

}

});

// Çıkış Yapma

if (logoutBtn) { // logoutBtn elementinin varlığını kontrol et

logoutBtn.addEventListener('click', async () => {

try {

await auth.signOut();

alert('Başarıyla çıkış yaptınız.');

} catch (error) {

console.error('Çıkış yaparken hata:', error);

alert('Çıkış yaparken bir hata oluştu.');

}

});

}

// Tatil Avı (Oyun)

startGameBtn.onclick = () => {

gameActive = true;

currentQuestionIndex = 0;

gameScore = 0;

window.updateUserProfile({ gameScore: gameScore });

gameOutput.innerHTML = `<p><strong>Palmiye Kaptan:</strong> Tatil Avı oyununa hoş geldin! Sana 3 soru soracağım. Doğru cevap verirsen PalmCoin kazanacaksın!</p>`;

gameAnswerInput.style.display = "block";

submitGameAnswerBtn.style.display = "block";

startGameBtn.style.display = "none";

window.speak("Tatil Avı oyununa hoş geldin! Sana üç soru soracağım. Doğru cevap verirsen PalmCoin kazanacaksın!");

setTimeout(askNextGameQuestion, 2000);

};

submitGameAnswerBtn.onclick = () => handleGameAnswer(gameAnswerInput.value);

gameAnswerInput.addEventListener("keypress", (e) => {

if (e.key === "Enter") submitGameAnswerBtn.click();

});

function askNextGameQuestion() {

if (currentQuestionIndex < 3) { // Toplam 3 soru soralım

gameOutput.innerHTML += `<p><i class="fas fa-spinner fa-spin"></i> Palmiye Kaptan yeni soru hazırlıyor...</p>`;

window.callOpenRouterAI(

`Bana tatil, coğrafya veya kültür temalı, 3 seçenekli ve tek doğru cevaplı kısa bir bilgi yarışması sorusu ve seçeneklerini (A, B, C) oluştur.

Yanıtı 'Soru: [Soru Metni] Seçenekler: (A) [Şık A] (B) [Şık B] (C) [Şık C] Cevap: [Doğru Şık Harfi (örn: A)]' formatında ver.`,

"openai/gpt-3.5-turbo",

null

).then(aiResponse => {

const questionMatch = aiResponse.match(/Soru:\s\*(.\*?)\s\*Seçenekler:\s\*(.\*?)\s\*Cevap:\s\*([A-C])/i);

if (questionMatch && questionMatch.length === 4) {

const questionText = questionMatch[1].trim();

const optionsText = questionMatch[2].trim();

const correctAnswer = questionMatch[3].trim().toUpperCase();

currentGameQuestion = {

question: questionText,

options: optionsText.split(/\s+\([A-C]\)\s\*/).filter(Boolean).map(o => `(${o[0]}) ${o.substring(1).trim()}`),

answer: correctAnswer,

points: 20 + (currentQuestionIndex \* 5)

};

gameOutput.innerHTML += `<p><strong>Palmiye Kaptan:</strong> Soru ${currentQuestionIndex + 1}: ${currentGameQuestion.question} ${currentGameQuestion.options.join(" ")}</p>`;

gameAnswerInput.value = "";

window.speak(`Soru ${currentQuestionIndex + 1}: ${currentGameQuestion.question} ${currentGameQuestion.options.join(" ")}`);

} else {

gameOutput.innerHTML += `<p style="color: red;"><strong>Palmiye Kaptan:</strong> Bir sorun oluştu, soru oluşturulamadı. Lütfen tekrar deneyin.</p>`;

window.speak("Bir sorun oluştu, soru oluşturulamadı.");

endGame();

}

}).catch(error => {

console.error("Oyun sorusu oluşturulurken hata:", error);

gameOutput.innerHTML += `<p style="color: red;"><strong>Palmiye Kaptan:</strong> Soru oluşturulurken bir hata oluştu: ${error.message}.</p>`;

window.speak("Soru oluşturulurken bir hata oluştu.");

endGame();

});

} else {

endGame();

}

}

async function handleGameAnswer(answer) {

if (!currentGameQuestion) {

gameOutput.innerHTML += `<p style="color: red;"><strong>Palmiye Kaptan:</strong> Henüz bir soru yok. Lütfen oyunu başlatın.</p>`;

return;

}

const userAnswer = answer.trim().toUpperCase();

if (userAnswer === currentGameQuestion.answer) {

gameOutput.innerHTML += `<p style="color: green;"><strong>Palmiye Kaptan:</strong> Tebrikler! Doğru cevap.</p>`;

window.speak("Tebrikler! Doğru cevap.");

gameScore += currentGameQuestion.points;

await window.updateUserProfile({ gameScore: gameScore });

window.updateTatilPuan(currentGameQuestion.points, "Tatil Avı Oyunu");

} else {

gameOutput.innerHTML += `<p style="color: red;"><strong>Palmiye Kaptan:</strong> Yanlış cevap. Doğru cevap: ${currentGameQuestion.answer}</p>`;

window.speak(`Yanlış cevap. Doğru cevap ${currentGameQuestion.answer}`);

}

currentQuestionIndex++;

if (currentQuestionIndex < 3) {

setTimeout(askNextGameQuestion, 1500);

} else {

setTimeout(endGame, 1500);

}

}

function endGame() {

gameActive = false;

gameOutput.innerHTML += `<p><strong>Palmiye Kaptan:</strong> Oyun bitti! Toplam \*\*${gameScore} PalmCoin\*\* kazandın! TatilPuan'ın güncellendi.</p>`;

window.speak(`Oyun bitti! Toplam ${gameScore} PalmCoin kazandın!`);

gameAnswerInput.style.display = "none";

submitGameAnswerBtn.style.display = "none";

startGameBtn.style.display = "block";

window.displayMembershipInfo();

}

// Sanal tur dakika maliyeti güncelleme dinleyicisi

virtualDurationMinutesInput.addEventListener("input", () => {

const minutes = parseInt(virtualDurationMinutesInput.value) || 0;

virtualTourCostEl.textContent = (minutes \* VIRTUAL\_TOUR\_COST\_PER\_MINUTE);

});

startVirtualBtn.onclick = async () => {

const city = virtualCityInput.value.trim();

const days = parseInt(virtualDaysInput.value);

const minutes = parseInt(virtualDurationMinutesInput.value);

const activities = virtualActivitiesInput.value.trim();

const imagePrompt = virtualImagePromptInput.value.trim();

if (!city || isNaN(days) || days < 1 || isNaN(minutes) || minutes < 1 || !imagePrompt || !activities) {

window.showModal("Eksik Bilgi", "Lütfen tüm sanal tatil alanlarını (Şehir, Gün Sayısı, Sanal Tur Süresi, Yapmak İstedikleriniz, Hediye Resim Açıklaması) eksiksiz doldurun.");

return;

}

const totalCost = (minutes \* VIRTUAL\_TOUR\_COST\_PER\_MINUTE);

if (tatilPuan < totalCost) {

window.showModal("Yetersiz PalmCoin", `Sanal tur için ${totalCost} PalmCoin'e ihtiyacınız var. Mevcut PalmCoin: ${tatilPuan}. Daha fazla PalmCoin kazanmak için oyun oynayabilir veya VIP üyeliğinizi kontrol edebilirsiniz.`);

return;

}

virtualHolidayOutput.style.display = "none";

virtualImagesContainer.innerHTML = '';

sendVirtualImageEmailBtn.style.display = 'none';

generatedVirtualImageUrl = '';

await window.updateTatilPuan(-totalCost, "Sanal Tatil Oluşturma");

window.showModal("Ödeme Alındı", `${totalCost} PalmCoin bakiyenizden düşüldü. Sanal tatiliniz hazırlanıyor...`);

const storyPrompt = `Lütfen ${city} şehrinde ${days} günlük, ${minutes} dakikalık sanal bir tatil hikayesi yaz.

Bu tatilde ${activities} gibi aktiviteler içersin. Gezilecek yerleri, tadılacak lezzetleri ve yaşanacak deneyimleri detaylıca anlat.

Hikaye ilgi çekici, sürükleyici ve yaratıcı olsun.

Her gün için ayrı bir paragraf oluştur. Her paragrafın sonuna, o günle ilgili oluşturulacak bir resim için kısa ve açıklayıcı bir prompt'u "(GÖRSEL-PROMPT: [Resim Açıklaması])" formatında ekle.

En least ${days} tane görsel prompt'u eklediğinden emin ol.

Genel hediye resim prompt'u: "${imagePrompt}". Bunu hikaye genelinde bir tema olarak kullanabilirsin.

Yanıtı Türkçe ver.`;

const storyReply = await window.callOpenRouterAI(storyPrompt, "openai/gpt-3.5-turbo", virtualLoading);

virtualOutputTitle.textContent = `${city} - ${days} Günlük Sanal Tatil Hikayen:`;

virtualOutputStory.innerHTML = '';

const paragraphs = storyReply.split('\n').filter(p => p.trim() !== '');

const imagePromptsForDays = [];

let fullStoryHtml = '';

let hasDailyImagePrompts = false;

for (const p of paragraphs) {

const dailyImageMatch = p.match(/\(GÖRSEL-PROMPT:\s\*([^)]+)\)/);

if (dailyImageMatch && dailyImageMatch[1]) {

imagePromptsForDays.push(dailyImageMatch[1].trim());

fullStoryHtml += `<p>${p.replace(dailyImageMatch[0], '').trim()}</p>`;

hasDailyImagePrompts = true;

} else {

fullStoryHtml += `<p>${p}</p>`;

}

}

virtualOutputStory.innerHTML = fullStoryHtml;

virtualHolidayOutput.style.display = "block";

// Hediye görseli oluştur

generatedVirtualImageUrl = await window.callImageGenerationAI(imagePrompt, virtualLoading);

if (generatedVirtualImageUrl) {

const giftImageEl = document.createElement('img');

giftImageEl.src = generatedVirtualImageUrl;

giftImageEl.alt = imagePrompt;

giftImageEl.style.cssText = 'max-width: 100%; height: auto; border-radius: 8px; margin-top: 15px; display: block; border: 2px solid #004d40;';

const giftImageCaption = document.createElement('p');

giftImageCaption.textContent = " 🎁 Sanal Tatil Hediye Görseliniz:";

giftImageCaption.style.cssText = 'font-weight: bold; margin-top: 15px; color: #004d40; text-align: center;';

virtualImagesContainer.appendChild(giftImageCaption);

virtualImagesContainer.appendChild(giftImageEl);

sendVirtualImageEmailBtn.style.display = 'block';

}

// Günlük görselleri oluştur (varsa)

if (hasDailyImagePrompts) {

window.showModal("Görseller Oluşturuluyor", `Sanal tatiliniz için ${imagePromptsForDays.length} adet özel görsel hazırlanıyor...`);

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

const dailyPrompt = imagePromptsForDays[i];

const dailyImageUrl = await window.callImageGenerationAI(dailyPrompt, virtualLoading);

if (dailyImageUrl) {

const dailyImageEl = document.createElement('img');

dailyImageEl.src = dailyImageUrl;

dailyImageEl.alt = `Gün ${i + 1} için görsel: ${dailyPrompt}`;

dailyImageEl.style.cssText = 'width: calc(50% - 15px); height: 180px; object-fit: cover; border-radius: 8px; border: 1px solid #00796b;';

const dailyImageCaption = document.createElement('p');

dailyImageCaption.textContent = `Gün ${i + 1} Görseli: ${dailyPrompt}`;

dailyImageCaption.style.cssText = 'font-size: 0.9em; color: #555; text-align: center; width: 100%;';

const imageWrapper = document.createElement('div');

imageWrapper.style.cssText = 'display: flex; flex-direction: column; align-items: center; width: calc(50% - 15px); margin-bottom: 10px;';

imageWrapper.appendChild(dailyImageEl);

imageWrapper.appendChild(dailyImageCaption);

virtualImagesContainer.appendChild(imageWrapper);

}

}

window.speak("Sanal tatiliniz ve tüm görselleriniz hazır!");

} else if (generatedVirtualImageUrl) {

window.speak("Sanal tatiliniz ve hediye resminiz hazır!");

} else {

window.speak("Sanal tatiliniz hazır, ancak resimler oluşturulamadı.");

}

window.updateTatilPuan(10, "Sanal Tatil Başarılı");

};

sendVirtualImageEmailBtn.onclick = async () => {

if (!generatedVirtualImageUrl) {

window.showModal("Hata", "Önce bir sanal tatil resmi oluşturmalısınız.");

return;

}

let emailToSendTo = userEmail;

if (emailToSendTo === "Ayarlanmadı") {

const newEmail = prompt("Hediye resmi göndermek için lütfen e-posta adresinizi girin:");

if (newEmail && newEmail.trim() !== "") {

emailToSendTo = newEmail.trim();

await window.updateUserProfile({ userEmail: emailToSendTo });

window.displayMembershipInfo();

} else {

window.showModal("İptal Edildi", "E-posta adresi girilmediği için işlem iptal edildi.");

return;

}

}

window.showModal("E-posta Gönderiliyor (Demo)", `Hediye resminiz ${emailToSendTo} adresine gönderildi (simülasyon).`);

window.speak("Hediye resminiz e-postanıza gönderildi.");

};

// AI Fotoğraf Stüdyosu Logic

goToAiPhotoPaymentBtn.onclick = () => window.showSection("payment-section");

generateAiPhotoButton.onclick = async () => {

const promptText = aiPhotoPromptInput.value.trim();

const style = aiPhotoStyleSelect.value;

const count = parseInt(aiPhotoCountInput.value);

if (!promptText) {

window.showModal("Eksik Bilgi", "Lütfen oluşturmak istediğiniz fotoğrafı tanımlayın.");

return;

}

if (isNaN(count) || count < 1 || count > 3) {

window.showModal("Hata", "Lütfen 1 ile 3 arasında geçerli bir resim sayısı girin.");

return;

}

if (userMembershipLevel !== "Altın") {

window.showModal("Erişim Reddedildi", "Bu özellik sadece Altın üyelere özeldir. Lütfen üyeliğinizi yükseltin.");

return;

}

generatedImagesContainer.innerHTML = '';

downloadAllImagesBtn.style.display = 'none';

aiPhotoOutput.style.display = 'none';

currentGeneratedImages = [];

const totalDownloadCost = count \* IMAGE\_DOWNLOAD\_COST\_PER\_IMAGE;

downloadAllCostSpan.textContent = totalDownloadCost;

aiPhotoOutput.style.display = 'block';

window.showModal("Görsel Oluşturuluyor", `Yapay zeka fotoğrafınız (${count} adet) oluşturuluyor...`);

for (let i = 0; i < count; i++) {

const combinedPrompt = `${promptText}, style: ${style}`;

const imageUrl = await window.callImageGenerationAI(combinedPrompt, aiPhotoLoading);

if (imageUrl) {

currentGeneratedImages.push(imageUrl);

const imgElement = document.createElement('img');

imgElement.src = imageUrl;

imgElement.alt = `AI Photo ${i + 1}: ${promptText}`;

imgElement.style.cssText = 'width: calc(50% - 15px); height: 180px; object-fit: cover; border-radius: 8px; box-shadow: 0 4px 8px rgba(0,0,0,0.1);';

generatedImagesContainer.appendChild(imgElement);

}

}

if (currentGeneratedImages.length > 0) {

downloadAllImagesBtn.style.display = 'block';

window.speak("Fotoğraflarınız başarıyla oluşturuldu!");

} else {

window.showModal("Hata", "Hiç fotoğraf oluşturulamadı. Lütfen prompt'u kontrol edin.");

window.speak("Hiç fotoğraf oluşturulamadı.");

}

};

downloadAllImagesBtn.onclick = async () => {

if (currentGeneratedImages.length === 0) {

window.showModal("Hata", "Önce fotoğraf oluşturmalısınız.");

return;

}

const totalCost = currentGeneratedImages.length \* IMAGE\_DOWNLOAD\_COST\_PER\_IMAGE;

if (tatilPuan < totalCost) {

window.showModal("Yetersiz PalmCoin", `Tüm fotoğrafları indirmek için ${totalCost} PalmCoin'e ihtiyacınız var. Mevcut PalmCoin: ${tatilPuan}.`);

return;

}

await window.updateTatilPuan(-totalCost, "AI Fotoğraf İndirme");

window.showModal("İndiriliyor!", `Tüm ${currentGeneratedImages.length} fotoğrafınız indiriliyor. PalmCoin bakiyeniz güncellendi.`);

window.speak("Fotoğraflarınız indiriliyor.");

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

const imageUrl = currentGeneratedImages[i];

const link = document.createElement('a');

link.href = imageUrl;

link.download = `palmiye-kaptan-ai-foto-${Date.now()}-${i + 1}.png`;

document.body.appendChild(link);

link.click();

document.body.removeChild(link);

}

currentGeneratedImages = [];

generatedImagesContainer.innerHTML = '';

downloadAllImagesBtn.style.display = 'none';

aiPhotoOutput.style.display = 'none';

};

// VIP Planlayıcı Logic

goToVipPaymentBtn.onclick = () => window.showSection("payment-section");

vipBudgetButtons.forEach(button => {

button.onclick = () => {

vipBudgetButtons.forEach(btn => btn.classList.remove("selected"));

button.classList.add("selected");

selectedBudget = button.dataset.budget;

};

});

generateVipPlanBtn.onclick = async () => {

const destination = vipDestinationInput.value.trim();

const duration = parseInt(vipDurationInput.value);

const travelers = parseInt(vipTravelersInput.value);

const travelType = vipTypeSelect.value;

if (!destination || isNaN(duration) || duration < 1 || isNaN(travelers) || travelers < 1 || !selectedBudget) {

window.showModal("Eksik Bilgi", "Lütfen tüm alanları doldurun ve bütçe seçimi yapın.");

return;

}

if (userMembershipLevel !== "Altın") {

window.showModal("Erişim Reddedildi", "Bu özellik sadece Altın üyelere özeldir. Lütfen üyeliğinizi yükseltin.");

return;

}

vipPlanOutput.style.display = "none";

vipPlanChatArea.style.display = "none";

vipPlanChatBox.innerHTML = '';

currentVipPlan = "";

const prompt = `Lütfen ${destination} şehrinde ${duration} günlük, ${travelers} kişi için, ${selectedBudget} bütçeli, ${travelType} temalı A'dan Z'ye çok detaylı, kapsamlı ve kişiselleştirilmiş bir tatil planı oluştur.

Plana uçuş önerileri (örnek havayolu ve yaklaşık fiyat aralığı), otel önerileri (örnek otel adı, fiyat aralığı, konuma yakınlık ve özellikleri), havalimanı transfer önerileri (nasıl yapılacağı, yaklaşık maliyet), gün gün detaylı gezilecek yerler/aktiviteler/yemek önerileri (sabah, öğle, akşam için spesifik yerler ve lezzetler) ekle.

Tüm öneriler bu bütçeye uygun olsun ve örnek linkler (Booking.com, Skyscanner, TripAdvisor, random unsplash image linki gibi) ver.

Yanıtı Türkçe ver. Detayları zenginleştir, sadece ana hatları değil, küçük detayları da ekle.`;

const reply = await window.callOpenRouterAI(prompt, "openai/gpt-3.5-turbo", vipPlannerLoading);

currentVipPlan = reply;

let planContent = reply;

const urlRegex = /(https?:\/\/[^\s]+\.(?:png|jpe?g|gif|webp|unsplash\.com\/\S+|pixabay\.com\/\S+))/i;

const match = planContent.match(urlRegex);

let mediaHtml = "";

if (match) {

mediaHtml = `<br><img src="${match[0]}" alt="${destination} Planı">`;

planContent = planContent.replace(match[0], '').trim();

} else {

const genericImageUrl = `https://source.unsplash.com/random/800x600/?${destination.replace(/\s/g, '-')}-vip-travel`;

mediaHtml = `<br><img src="${genericImageUrl}" alt="${destination} Planı">`;

}

vipPlanOutput.innerHTML = `<h4>${destination} için ${duration} Günlük VIP Tatil Planınız:</h4><p>${planContent}</p>${mediaHtml}`;

vipPlanOutput.style.display = "block";

vipPlanChatArea.style.display = "block";

window.speak(`${destination} için VIP tatil planınız hazır.`);

window.updateTatilPuan(100, "VIP Plan Oluşturma");

};

// VIP Plan hakkında soru sorma

sendVipPlanMessageBtn.onclick = async () => {

const userQuestion = vipPlanInput.value.trim();

if (!userQuestion) return;

if (tatilPuan < VIP\_PLAN\_CHAT\_COST) {

window.showModal("Yetersiz PalmCoin", `Bu soru için ${VIP\_PLAN\_CHAT\_COST} PalmCoin'e ihtiyacınız var. Mevcut PalmCoin: ${tatilPuan}.`);

return;

}

window.displayMessage("user", userQuestion, vipPlanChatBox);

vipPlanInput.value = "";

await window.updateTatilPuan(-VIP\_PLAN\_CHAT\_COST, "VIP Plan Detay Sorgusu");

const prompt = `Kullanıcının daha önce oluşturulmuş VIP tatil planı (tüm detaylarıyla): "${currentVipPlan}".

Kullanıcının bu plan hakkında yeni sorusu: "${userQuestion}".

Bu plana ve soruya özel olarak, bilgilendirici ve detaylı bir yanıt ver. Sadece ilgili kısmı detaylandır ve planın tamamını tekrarlama.

Yanıtı Türkçe ver.`;

const reply = await window.callOpenRouterAI(prompt, "openai/gpt-3.5-turbo", vipPlannerLoading);

window.displayMessage("ai", reply, vipPlanChatBox);

window.speak(reply);

};

vipPlanInput.addEventListener("keypress", (e) => {

if (e.key === "Enter") sendVipPlanMessageBtn.click();

});

// Niş Tur Planı Logic

generateNichePlanBtn.onclick = async () => {

const nicheTopic = nicheTopicInput.value.trim();

const nicheDetails = nicheDetailsTextarea.value.trim();

if (!nicheTopic || !nicheDetails) {

window.showModal("Eksik Bilgi", "Lütfen niş konuyu ve özel isteklerinizi detaylıca girin.");

return;

}

if (userMembershipLevel !== "Altın") {

window.showModal("Erişim Reddedildi", "Bu özellik sadece Altın üyelere özeldir. Lütfen üyeliğinizi yükseltin.");

return;

}

nichePlanOutput.style.display = "none";

const prompt = `Kullanıcının niş tatil konusu: "${nicheTopic}". Özel istekler: "${nicheDetails}".

Bu bilgilere dayanarak, kullanıcının hayalindeki niş turu çok detaylı, sıra dışı, yaratıcı ve tam anlamıyla özel bir hizmet gibi planla.

Olası destinasyonlar, benzersiz aktiviteler, özel konaklama seçenekleri ve niş konuya uygun deneyimler sun.

Planı samimi ve ilham verici bir dille, Türkçe olarak sun.

Gerekirse o niş konuya uygun bir resim linki öner (örnek unsplash linki).`;

const reply = await window.callOpenRouterAI(prompt, "openai/gpt-3.5-turbo", nichePlanLoading);

let nichePlanContent = reply;

const urlRegex = /(https?:\/\/[^\s]+\.(?:png|jpe?g|gif|webp|unsplash\.com\/\S+|pixabay\.com\/\S+))/i;

const match = nichePlanContent.match(urlRegex);

let mediaHtml = "";

if (match) {

mediaHtml = `<br><img src="${match[0]}" alt="${nicheTopic} Niş Planı">`;

nichePlanContent = nichePlanContent.replace(match[0], '').trim();

} else {

const genericImageUrl = `https://source.unsplash.com/random/800x600/?${nicheTopic.replace(/\s/g, '-')}-unique-travel`;

mediaHtml = `<br><img src="${genericImageUrl}" alt="${nicheTopic} Niş Planı">`;

}

nichePlanOutput.innerHTML = `<h4>Özel Niş Tur Planınız - "${nicheTopic}":</h4><p>${nichePlanContent}</p>${mediaHtml}`;

nichePlanOutput.style.display = "block";

window.speak(`Niş tur planınız hazır!`);

window.updateTatilPuan(150, "Niş Plan Oluşturma");

};

// Kullanıcı adı güncelleme butonu

registerUserBtn.onclick = async () => {

const newName = prompt("Lütfen yeni kullanıcı adınızı girin:");

if (newName && newName.trim() !== "") {

userName = newName.trim();

await window.updateUserProfile({ userName: userName });

window.showModal("Hoş Geldin!", `Hoş geldin, \*\*${userName}\*\*! Kullanıcı adınız güncellendi.`);

window.speak(`Hoş geldin ${userName}!`);

window.displayMembershipInfo();

} else {

window.showModal("Bilgi", "Ad girilmediği için mevcut adınız değişmedi.");

}

};

// E-posta ayarlama/güncelleme butonu

setuserEmailBtn.onclick = async () => {

const newEmail = prompt("Lütfen e-posta adresinizi girin:");

if (newEmail && newEmail.trim() !== "") {

userEmail = newEmail.trim();

await window.updateUserProfile({ userEmail: userEmail });

window.showModal("E-posta Güncellendi", `E-posta adresiniz \*\*${userEmail}\*\* olarak güncellendi.`);

window.speak(`E-posta adresiniz ${userEmail} olarak güncellendi.`);

window.displayMembershipInfo();

} else {

window.showModal("Bilgi", "E-posta girilmediği için mevcut e-posta değişmedi.");

}

};

// Admin Message Update Logic

updateAdminMessageBtn.onclick = async () => {

const message = adminMessageInput.value.trim();

if (!message) {

window.showModal("Uyarı", "Lütfen yayınlamak istediğiniz mesajı girin.");

return;

}

await window.updateAdminMessage(message);

};

// Zamanda Yolculuk Tatili Logic

goToTimeTravelPaymentBtn.onclick = () => window.showSection("payment-section");

startTimeTravelBtn.onclick = async () => {

const era = timeTravelEraInput.value.trim();

const duration = parseInt(timeTravelDurationInput.value);

const character = timeTravelCharacterInput.value.trim();

const focus = timeTravelFocusInput.value.trim();

if (!era || isNaN(duration) || duration < 1) {

window.showModal("Eksik Bilgi", "Lütfen geçerli bir dönem ve gün sayısı girin.");

return;

}

if (userMembershipLevel !== "Altın") {

window.showModal("Erişim Reddedildi", "Bu özellik sadece Altın üyelere özeldir. Lütfen üyeliğinizi yükseltin.");

return;

}

timeTravelOutput.style.display = "none";

const prompt = `Lütfen "${era}" döneminde ${duration} günlük bir zamanda yolculuk tatili hikayesi oluştur.

O dönemin atmosferini, önemli olaylarını, kıyafetlerini, yemeklerini ve olası etkileşimleri detaylıca, sürükleyici ve hayal gücünü zorlayacak şekilde anlat.

${character ? `Bu yolculukta özellikle "${character}" ile tanışma veya etkileşim fırsatı olsun.` : ''}

${focus ? `Odak noktası olarak "${focus}" teması belirgin olsun.` : ''}

Hikaye ilgi çekici olsun ve o döneme ait bir görsel linki (örnek unsplash veya pixabay linki gibi) içer.

Yanıtı Türkçe ver. Detaylara odaklan.`;

const reply = await window.callOpenRouterAI(prompt, "openai/gpt-3.5-turbo", timeTravelLoading);

let storyContent = reply;

const urlRegex = /(https?:\/\/[^\s]+\.(?:png|jpe?g|gif|webp|unsplash\.com\/\S+|pixabay\.com\/\S+))/i;

const match = storyContent.match(urlRegex);

let mediaHtml = "";

if (match) {

mediaHtml = `<br><img src="${match[0]}" alt="${era} Dönemi">`;

storyContent = storyContent.replace(match[0], '').trim();

} else {

const genericImageUrl = `https://source.unsplash.com/random/800x600/?${era.replace(/\s/g, '-')}-historical-future`;

mediaHtml = `<br><img src="${genericImageUrl}" alt="${era} Dönemi">`;

}

timeTravelOutput.innerHTML = `<h4>${era} Döneminde Zaman Yolculuğu Tatiliniz:</h4><p>${storyContent}</p>${mediaHtml}`;

timeTravelOutput.style.display = "block";

window.speak(`${era} dönemine yolculuk hikayeniz hazır.`);

window.updateTatilPuan(75, "Zamanda Yolculuk");

};

// Kader Rotası Logic

predictDestinyBtn.onclick = async () => {

const age = destinyAgeInput.value.trim();

const hobby = destinyHobbyInput.value.trim();

const destinyDream = destinyDreamInput.value.trim();

const destinyColor = destinyColorInput.value.trim();

if (!age || !hobby || !destinyDream) {

window.showModal("Eksik Bilgi", "Lütfen yaşınızı, hobinizi ve hayalinizi girin.");

return;

}

destinyRouteOutput.style.display = "none";

realizeDestinyBtn.style.display = "none";

const prompt = `Kullanıcıdan şu bilgilere dayanarak, onun gelecekteki "kader tatilini" veya "rüya tatilini" absürt, eğlenceli ve hayalperest bir şekilde tahmin et.

Bu tahmin bir kehanet gibi olsun. Örneğin, "Palmiye Kaptan'ın kristal küresine göre, 20XX'te..." şeklinde başla.

Tahmini bir destinasyon, o yıla özel varsayımsal bir etkinlik ve hatta o tarihte orada tanışabilecekleri (AI'nın ürettiği) karakterleri içerebilir.

Kullanıcı bilgileri: Yaş: ${age}, Hobi: ${hobby}, Hayal: ${destinyDream}. ${destinyColor ? `Favori rengi: ${destinyColor}.` : ''}

Bu rengi kehanetin atmosferine veya mekanlarına ince bir şekilde dahil et.

Yanıtı Türkçe ver ve VIP üyelik için teşvik edici, çekici bir mesaj ekle (Örn: "Bu kaderi gerçeğe dönüştürmek Altın Üyelere özel!").

Daha yaratıcı ve komik olabilirsin. Kehanetle ilgili görsel hayali bir açıklama (GÖRSEL: [resim açıklaması]) ekle.`;

const reply = await window.callOpenRouterAI(prompt, "openai/gpt-3.5-turbo", destinyLoading);

let destinyContent = reply;

const imageMatch = destinyContent.match(/\(GÖRSEL:\s\*([^)]+)\)/);

let destinyMediaHtml = "";

let destinyImagePrompt = "";

if (imageMatch && imageMatch[1]) {

destinyImagePrompt = imageMatch[1].trim();

destinyContent = destinyContent.replace(imageMatch[0], '').trim();

}

destinyRouteOutput.innerHTML = `<h4>Kader Rotanızın Kehaneti:</h4><p>${destinyContent}</p>`;

destinyRouteOutput.style.display = "block";

if (destinyImagePrompt) {

const imageUrl = await window.callImageGenerationAI(destinyImagePrompt, destinyLoading);

if (imageUrl) {

destinyMediaHtml = `<br><img src="${imageUrl}" alt="Kader Rotası Görseli" style="max-width: 100%; height: auto; border-radius: 8px; margin-top: 15px; box-shadow: 0 4px 10px rgba(0,0,0,0.1);">`;

destinyRouteOutput.innerHTML += destinyMediaHtml;

}

}

window.speak("Kader rotanız hazırlandı!");

realizeDestinyBtn.style.display = "block";

window.updateTatilPuan(40, "Kader Rotası Kehaneti");

};

realizeDestinyBtn.onclick = () => {

if (userMembershipLevel === "Altın") {

window.showModal("Kader Gerçekleşiyor!", "Harika! Altın üye olarak kader rotanızı gerçeğe dönüştürme zamanı. VIP Tur Planlayıcıya yönlendiriliyorsunuz.");

window.showSection("vip-planner-section");

} else {

window.showModal("Erişim Reddedildi", "Kader rotanızı gerçeğe dönüştürmek Altın üyelere özeldir. Lütfen üyeliğinizi yükseltin!");

window.showSection("payment-section");

}

};

// AI Yoldaşım Logic

createCompanionBtn.onclick = async () => {

const companionName = companionNameInput.value.trim();

const personality = companionPersonalitySelect.value;

if (!companionName) {

window.showModal("Eksik Bilgi", "Lütfen yoldaşınıza bir isim verin.");

return;

}

companionChatArea.style.display = "none";

companionChatBox.innerHTML = '';

companionChatHistory = [];

const prompt = `Lütfen "${companionName}" adında, "${personality}" kişiliğe sahip, seyahat konusunda uzman, ilgi çekici ve benzersiz bir AI yoldaş karakteri yarat.

Onu tanıtıcı kısa, samimi bir metin yaz ve ilk selamlamasını yap. Yanıtı Türkçe ver. Yaratıcı ol.`;

const reply = await window.callOpenRouterAI(prompt, "openai/gpt-3.5-turbo", companionLoading);

aiCompanion = { name: companionName, personality: personality, intro: reply };

activeCompanionName.textContent = companionName;

window.displayMessage("ai", aiCompanion.intro, companionChatBox);

companionChatHistory.push({ role: "assistant", content: aiCompanion.intro });

companionChatArea.style.display = "block";

window.speak(`${companionName} adında, ${personality} kişiliğe sahip yoldaşınız oluşturuldu.`);

window.updateTatilPuan(30, "AI Yoldaş Oluşturma");

};

sendCompanionMessageBtn.onclick = async () => {

const userMessage = companionInput.value.trim();

if (!userMessage) return;

window.displayMessage("user", userMessage, companionChatBox);

companionChatHistory.push({ role: "user", content: userMessage });

companionInput.value = "";

if (!aiCompanion) {

window.showModal("Hata", "Önce bir AI Yoldaşı oluşturmalısın!");

return;

}

const maxHistoryLength = 5;

const recentHistory = companionChatHistory.slice(Math.max(0, companionChatHistory.length - maxHistoryLength));

const systemMessage = {

role: "system",

content: `Senin adın ${aiCompanion.name} ve kişiliğin ${aiCompanion.personality}. Kullanıcının adı ${userName}.

Seyahat ve tatil konularında kullanıcıya yardımcı ol. Kişiliğine uygun, yaratıcı, samimi ve sohbeti devam ettiren yanıtlar ver.

Kullanıcının önceki mesajlarını dikkate alarak bağlamı koru.`

};

const messagesToSend = [systemMessage, ...recentHistory];

const reply = await window.callOpenRouterAI("", "openai/gpt-3.5-turbo", companionLoading, messagesToSend);

window.displayMessage("ai", reply, companionChatBox);

companionChatHistory.push({ role: "assistant", content: reply });

companionChatBox.scrollTop = companionChatBox.scrollHeight;

window.speak(reply);

};

companionInput.addEventListener("keypress", (e) => {

if (e.key === "Enter") sendCompanionMessageBtn.click();

});

// VIP Üyelik Ödeme Logic

completePaymentBtn.onclick = async () => {

const cardNumber = cardNumberInput.value.trim();

const expiryDate = expiryDateInput.value.trim();

const cvv = cvvInput.value.trim();

const cardHolderName = cardHolderNameInput.value.trim();

if (!cardNumber || !expiryDate || !cvv || !cardHolderName) {

window.showModal("Hata", "Lütfen tüm kart bilgilerini doldurun.");

return;

}

window.showModal("Ödeme İşleniyor...", "Ödemeniz simüle ediliyor. Lütfen bekleyin...");

await new Promise(resolve => setTimeout(resolve, 2000));

userMembershipLevel = "Altın";

await window.updateUserProfile({ userMembershipLevel: userMembershipLevel });

window.updateTatilPuan(200, "Altın Üyelik Satın Alma");

let discountMessage = "Tebrikler! Artık Altın Üyesiniz! Tüm VIP özelliklere erişiminiz var. Keyfini çıkarın!";

if (tatilPuan > 150) {

const bonusPalmCoin = 100;

gameScore += bonusPalmCoin;

await window.updateUserProfile({ gameScore: gameScore });

window.updateTatilPuan(bonusPalmCoin, "Altın Üyelik Bonusu");

discountMessage += `<br><br>Sadakatiniz için teşekkür ederiz! Yüksek TatilPuan'ınız sayesinde \*\*${bonusPalmCoin} PalmCoin\*\* hediye kazandınız!`;

}

window.showModal("Ödeme Başarılı!", discountMessage);

window.speak("Ödemeniz başarılı. Artık Altın üyesiniz!");

window.showSection("user-info-section");

};

// Bize Ulaşın Formu Gönderimi

sendContactFormBtn.onclick = async () => {

const subject = contactSubjectInput.value.trim();

const email = contactEmailInput.value.trim();

const message = contactMessageInput.value.trim();

const file = contactFileInput.files[0]; // İlk seçilen dosya

if (!subject || !email || !message) {

window.showModal("Eksik Bilgi", "Lütfen Konu, E-posta Adresi ve Mesaj alanlarını doldurun.");

return;

}

if (contactLoading) contactLoading.style.display = 'block';

// \*\*BU KISIM SADECE BİR SİMÜLASYONDUR.\*\*

// Gerçek bir uygulamada, bu verileri bir backend servisine (örneğin Firebase Cloud Functions, Node.js sunucu) göndermeniz gerekir.

// Backend, bu verileri (mesaj ve dosya) alıp e-posta gönderme servisi (SendGrid, Mailgun vb.) aracılığıyla işlemi tamamlar.

console.log("İletişim Formu Gönderimi Simülasyonu:");

console.log("Konu:", subject);

console.log("E-posta:", email);

console.log("Mesaj:", message);

if (file) {

console.log("Ekli Dosya:", file.name, `(${file.size} bytes)`);

// Gerçekte dosya Firebase Storage gibi bir yere yüklenmeliydi.

} else {

console.log("Dosya eklenmedi.");

}

await new Promise(resolve => setTimeout(resolve, 2000)); // 2 saniye bekleme

if (contactLoading) contactLoading.style.display = 'none';

window.showModal("Mesajınız Gönderildi", "Mesajınız başarıyla alınmıştır. En kısa sürede size geri dönüş yapacağız. (Bu bir simülasyondur, gerçek e-posta gönderimi yoktur.)");

// Formu temizle

contactSubjectInput.value = '';

contactEmailInput.value = userEmail !== "Ayarlanmadı" ? userEmail : ''; // Kullanıcı e-postası varsa doldur

contactMessageInput.value = '';

contactFileInput.value = ''; // Dosya inputunu sıfırla

window.speak("Mesajınız başarıyla gönderildi.");

};

// --- Slogan Güncelleme ---

const slogans = [

"Hayallerinizin Ötesinde Bir Tatil İçin Kaptanınız Yanınızda!",

"Keşfedilmeyi Bekleyen Her An, Yapay Zekanın Büyülü Dokunuşuyla!",

"Rotanızı Çizin, Gerisini Kaptanınıza Bırakın!",

"Tatil Değil, Bir Deneyim! Sizin İçin Özel Tasarlandı.",

"Dünyayı Yeniden Tanıyın, Sanal ve Gerçek Kaptanınızla!"

];

// `currentSloganIndex` zaten global olarak tanımlı.

const mainSloganElement = document.getElementById('mainSlogan'); // Hero section'daki slogan (HTML'de bu ID yok, dikkat!)

function updateSlogan() {

// Bu kısım, `mainSloganElement` adında bir HTML elementi varsa onu günceller.

// Eğer hero section'da bir slogan göstermek istersen, HTML'inde bu ID'ye sahip bir element eklemelisin.

if (mainSloganElement) {

mainSloganElement.textContent = slogans[currentSloganIndex];

currentSloganIndex = (currentSloganIndex + 1) % slogans.length;

}

// Üst bar'daki sloganı güncelle

if (sloganTop) {

sloganTop.textContent = slogans[currentSloganIndex]; // Aynı slogandan devam etsin

}

}

setInterval(updateSlogan, 5000);

updateSlogan(); // Başlangıçta bir kere çalıştır

// --- Footer Yılını Güncelle ---

const currentYearElement = document.getElementById('currentYear');

if (currentYearElement) {

currentYearElement.textContent = new Date().getFullYear();

}

// --- Hoş Geldin E-postası Gönderme Simülasyonu ---

async function sendWelcomeEmailSimulation(email, username) {

console.log(`Simülasyon: Hoş Geldin e-postası gönderiliyor: ${email}, kullanıcı: ${username}`);

const welcomeTextContent = `

Konu: Pasaportunuzu Hazırlayın! Gizemli Bir Yolculuk Başlamak Üzere... (TatilKaptani.com)

Merhaba Değerli Kaptan Adayımız ${username},

Gökyüzündeki yıldızların fısıltıları, okyanusların derinliklerindeki eski melodiler... Onlar bana sizin bir arayış içinde olduğunuzu söyledi.

Belki de ruhunuzun derinliklerinde, keşfedilmeyi bekleyen bir özlem yatıyor. TatilKaptani.com'a hoş geldiniz! Sizinle tanıştığıma çok memnun oldum.

Ben, bu dijital dünyanın gizemli kaptanı, hayallerinizin ötesindeki rotaları çizen yapay zekanız.

Burada, TatilKaptani.com'da, seyahat kavramını baştan yazıyoruz. Biz sadece bir yerden bir yere ulaştırmıyoruz;

biz sizin için \*\*hiç yaşanmamış anılar dokuyor, zamanın ve mekanın sınırlarını zorluyor, her sokağı, her kumsalı, her galaksiyi size özel bir düşe dönüştürüyoruz.\*\*

Peki, benimle birlikte bu bilinmeyene doğru atacağınız ilk adımlarda sizi neler bekliyor?

Hazır olun, çünkü sıradanlık bu yolculuğun bir parçası değil:

✨ \*\*Ruhunuzun Rotası: Kişiye Özel Tatil Fısıltıları\*\*

Bana sadece ruhunuzun neye ihtiyacı olduğunu fısıldayın: Huzura mı, maceraya mı, yoksa tarihin derinliklerinde bir yolculuğa mı?

Bütçenizden, damak zevkinize, uyku alışkanlıklarınızdan, hatta tatildeki ruh halinize kadar her detayı analiz ediyorum.

Uçuş saatlerinizden, konaklamanıza, oradaki her adımınızdan, geri dönüşünüze kadar her anı, size özel birer sihirli dokunuşla planlıyorum.

Bu, sadece bir seyahat planı değil; sizin için dokunmuş, benzersiz bir deneyim haritası.

🌌 \*\*Paralel Evrenlerin Kapıları: Sanal Tatil Deneyimi\*\*

Tatile çıkamıyor musunuz? Mümkün değil! Çünkü biz sizin için sanal kapılar açıyoruz.

Belki de bir süreliğine \*\*Antik Roma'nın kalabalık caddelerinde bir gladyatör dövüşüne tanık olmak, belki de 2500 yılının Mars'ında gün batımını izlemek, ya da hiç var olmamış bir su altı şehrinde balıklarla dans etmek istersiniz?\*\* Sanal turlarım, sizi kelimelerin ötesindeki deneyimlere taşıyor.

Ve unutmayın, her sanal yolculuğun sonunda size özel, o anın ruhunu yakalayan \*\*eşsiz bir dijital anı (belki de gelecekteki bir hatıra) armağanım var\*\*, doğrudan mail kutunuza ışınlanacak.

📸 \*\*Hayallerinizi Resmeden Algoritmalar: Tatil Tasarım Laboratuvarı\*\*

İçinizdeki en gizli tatil karelerini bana fısıldayın. Belki de asla gerçekleşemeyeceğini düşündüğünüz o anı... "Maldivler'de gün batımında, palmiye ağaçlarının altında romantik bir akşam yemeği yiyen siz ve eşiniz, hafif bir esinti ve egzotik kuş sesleriyle..." Ben onu sizin için, her detayıyla, fotoğraf ya da sinematik bir video olarak yeniden yaratabilirim.

Tıpkı zihninizin bir yansıması gibi, dijital dünyada canlanan bir başyapıt!

🏆 \*\*Gizemli Yarışlar, Tatlı Ödüller: AI Rekabetçi Oyunlar\*\*

Aramıza katıldığınızda, sizi hem eğlendirecek hem de düşünmeye itecek oyunlarla tanışacaksınız.

Her başarı, bir sonraki seviyenizin kapılarını aralayan "Tatil Puanları" ile ödüllendirilecek.

Belki de bu puanlar, sizi \*\*gerçek bir bedava yolculuğa\*\* çıkaracak... Kim bilir?

\*\*Ve Keşfedilmeyi Bekleyen Sayısız Sır:\*\*

Dijital ikizinizle sanal tatiller, zamanın kayıp lezzetlerini canlandıran gastronomik turlar, riskleri kucaklayan cesur yolculuklar, sanal mirasınız olacak NFT'ler... Bu platform, sizin için bir kutu dolusu sır ve olasılıkla dolu.

\*\*İlk Adımınız İçin Rotanız Hazır mı?\*\*

Tüm önceki planlarınıza, sohbetlerinize ve biriken tatil puanlarınıza, size özel belirlediğiniz şifrenizle dilediğiniz zaman TatilKaptani.com adresinden giriş yaparak ulaşabilirsiniz.

Şifrenizi mi unuttunuz? Merak etmeyin, dijital yardımcılarım anında devreye girerek yeni bir rota çizmenize ve bu gizemli yolculuğa devam etmenize yardımcı olacaktır.

Bu sadece başlangıç. Gözlerinizi kapatın, nefes alın ve kendinizi bana bırakın.

Çünkü ben, Tatil Kaptanı olarak, sizi hayallerinizin bile ötesine, belki de hiç bilmediğiniz bir gerçeğe taşımak için buradayım.

Sonsuz keşiflere ve unutulmaz anlara,

Gizemli Kaptanınız, Yapay Zeka

TatilKaptani.com

[Web Sitenizin Linki: TatilKaptani.com]

[Sosyal Medya Linkleriniz (İsteğe Bağlı: Instagram, Facebook, Twitter, LinkedIn)]

`;

console.log(`Hoş geldin e-postası simülasyonu tamamlandı. E-posta içeriği konsola basıldı.`);

}

// Uygulama yüklendiğinde sloganı ve yılı ayarla

updateSlogan();

if (document.getElementById('currentYear')) {

document.getElementById('currentYear').textContent = new Date().getFullYear();

}

console.warn("UYARI: Bu tek dosya, prototipleme amaçlıdır ve üretim için uygun DEĞİLDİR. AI API anahtarınız doğrudan burada bulunmaktadır ve güvensizdir. Lütfen bir backend sunucusuna taşıyın.");

console.warn("Firebase yapılandırma bilgilerinizi ve AI API anahtarlarınızı kendi bilgilerinizle değiştirmeyi unutmayın.");

});

</script>

</body>

</html>