<!DOCTYPE html> <html lang="tr"> <head> <meta charset="UTF-8" /> <meta name="viewport" content="width=device-width, initial-scale=1" /> <title>🌴 TatilKaptanı - Yapay Zeka Tatil Planlayıcı</title> <style> body { margin: 0; font-family: 'Segoe UI', sans-serif; background: linear-gradient(to bottom, #a7ffeb, #00897b); color: #004d40; height: 100vh; overflow: hidden; display: flex; justify-content: center; align-items: center; } .container { width: 420px; height: 700px; background: white; border-radius: 15px; display: flex; flex-direction: column; box-shadow: 0 0 20px rgba(0,0,0,0.2); position: relative; } header { background: #00796b; color: white; padding: 15px; font-size: 24px; border-radius: 15px 15px 0 0; text-align: center; } #slogan { font-size: 14px; padding: 5px; font-style: italic; background: #b2dfdb; text-align: center; } #chat-box { flex: 1; padding: 10px; background: #e0f2f1; overflow-y: auto; } .user-msg { text-align: right; color: #00796b; font-weight: bold; } .bot-msg { text-align: left; color: #004d40; } .input-area { background: #004d40; padding: 10px; display: flex; align-items: center; gap: 10px; border-radius: 0 0 15px 15px; } input { flex: 1; padding: 10px; border-radius: 8px; font-size: 16px; border: none; } button { background: #a7ffeb; padding: 8px 16px; border-radius: 8px; border: none; font-weight: bold; cursor: pointer; } #captain { position: fixed; top: 20px; right: 20px; width: 120px; cursor: move; z-index: 1000; } #tatilpuan { position: fixed; bottom: 20px; left: 20px; background: #00796b; color: white; padding: 10px; border-radius: 20px; font-weight: bold; z-index: 999; } #voice-toggle { color: white; font-size: 14px; cursor: pointer; } </style> </head> <body> <img id="captain" src="https://upload.wikimedia.org/wikipedia/commons/thumb/5/51/Cartoon\_ship\_captain.png/240px-Cartoon\_ship\_captain.png" alt="Palmiye Kaptan" /> <div class="container"> <header>🌴 Palmiye Kaptan</header> <div id="slogan">"Hayalindeki tatili Palmiye Kaptan'la keşfet!"</div> <div id="chat-box" aria-live="polite"></div> <div class="input-area"> <input type="text" id="user-input" placeholder="Tatil sorunuzu yazın..." /> <button id="send-button">Gönder</button> <div id="voice-toggle">🔊 Ses: Açık</div> </div> </div> <div id="tatilpuan">TatilPuan: 0</div> <script> const chatBox = document.getElementById("chat-box"); const input = document.getElementById("user-input"); const sendBtn = document.getElementById("send-button"); const tatilpuanEl = document.getElementById("tatilpuan"); const voiceToggle = document.getElementById("voice-toggle"); let tatilPuan = 0; let voiceEnabled = true; const captain = document.getElementById("captain"); let isDragging = false; let offset = { x: 0, y: 0 }; captain.addEventListener("mousedown", function (e) { isDragging = true; offset = { x: e.clientX - captain.offsetLeft, y: e.clientY - captain.offsetTop }; }); document.addEventListener("mouseup", () => isDragging = false); document.addEventListener("mousemove", function (e) { if (isDragging) { captain.style.left = e.clientX - offset.x + "px"; captain.style.top = e.clientY - offset.y + "px"; } }); function detectLanguage(text) { return /[ğüşıöçĞÜŞİÖÇ]/.test(text) ? "tr" : "en"; } function speak(text, lang) { if (!voiceEnabled) return; const utter = new SpeechSynthesisUtterance(text); utter.lang = lang === "tr" ? "tr-TR" : "en-US"; speechSynthesis.speak(utter); } voiceToggle.onclick = () => { voiceEnabled = !voiceEnabled; voiceToggle.textContent = voiceEnabled ? "🔊 Ses: Açık" : "🔇 Ses: Kapalı"; }; async function sendMessage() { const userMessage = input.value.trim(); if (!userMessage) return; const lang = detectLanguage(userMessage); chatBox.innerHTML += `<p class="user-msg">Sen: ${userMessage}</p>`; input.value = ""; try { const response = await fetch("https://openrouter.ai/api/v1/chat/completions", { method: "POST", headers: { "Authorization": "Bearer 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", "Content-Type": "application/json" }, body: JSON.stringify({ model: "openai/gpt-3.5-turbo", messages: [{ role: "user", content: userMessage }] }) }); const data = await response.json(); const reply = data?.choices?.[0]?.message?.content || "❌ Cevap alınamadı."; chatBox.innerHTML += `<p class="bot-msg">Palmiye Kaptan: ${reply}</p>`; chatBox.scrollTop = chatBox.scrollHeight; tatilPuan += 10; tatilpuanEl.textContent = `TatilPuan: ${tatilPuan}`; speak(reply, lang); } catch (err) { chatBox.innerHTML += `<p class="bot-msg">❗ Hata: ${err.message}</p>`; } } sendBtn.onclick = sendMessage; input.addEventListener("keypress", function (e) { if (e.key === "Enter") sendMessage(); }); </script> </body> </html>

Son hali

<!DOCTYPE html>

<html lang="tr">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<title>🌴 TatilKaptanı - Yapay Zeka Tatil Planlayıcı</title>

<style>

body {

margin: 0;

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

background: linear-gradient(to bottom, #a7ffeb, #00897b);

color: #004d40;

min-height: 100vh;

display: flex;

justify-content: center;

align-items: center;

}

.container {

background: white;

width: 420px;

max-width: 100vw;

height: 700px;

border-radius: 15px;

box-shadow: 0 0 20px rgba(0,0,0,0.2);

display: flex;

flex-direction: column;

position: relative;

}

header {

background: #00796b;

color: white;

padding: 15px;

border-radius: 15px 15px 0 0;

text-align: center;

font-size: 20px;

font-weight: 700;

}

#chat-box {

flex-grow: 1;

padding: 10px;

background: #e0f2f1;

overflow-y: auto;

border-left: 6px solid #004d40;

border-right: 6px solid #004d40;

font-size: 14px;

}

#chat-box p {

margin: 8px 0;

line-height: 1.4;

}

.user-msg {

text-align: right;

font-weight: 600;

color: #00796b;

}

.bot-msg {

text-align: left;

color: #004d40;

}

.input-area {

display: flex;

padding: 10px;

background: #004d40;

border-radius: 0 0 15px 15px;

gap: 10px;

}

#user-input {

flex-grow: 1;

padding: 10px;

border-radius: 8px;

border: none;

font-size: 14px;

}

#send-button {

background: #a7ffeb;

border: none;

border-radius: 8px;

padding: 0 20px;

font-weight: 700;

cursor: pointer;

}

#captain {

position: fixed;

top: 20px;

right: 20px;

width: 120px;

user-select: none;

z-index: 9999;

}

#tatilpuan {

position: fixed;

bottom: 20px;

left: 20px;

background: #00796b;

color: white;

padding: 10px 15px;

border-radius: 20px;

font-weight: 700;

box-shadow: 0 0 10px #004d40;

z-index: 9998;

}

.voice-toggle {

color: white;

cursor: pointer;

font-size: 14px;

}

#slogan {

text-align: center;

font-size: 14px;

padding: 5px 10px;

font-style: italic;

background: #b2dfdb;

color: #004d40;

}

#mini-game {

position: fixed;

bottom: 60px;

left: 20px;

background: #fff3e0;

color: #004d40;

padding: 10px;

border-radius: 10px;

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

font-size: 14px;

}

#game-answer {

margin-top: 5px;

width: 100%;

padding: 5px;

border: 1px solid #ccc;

border-radius: 5px;

}

</style>

</head>

<body>

<img id="captain" src="https://upload.wikimedia.org/wikipedia/commons/thumb/5/51/Cartoon\_ship\_captain.png/240px-Cartoon\_ship\_captain.png" alt="Palmiye Kaptan" title="Palmiye Kaptan" />

<div class="container">

<header>🌴 Palmiye Kaptan<br><small>Sizin sanal tatil asistanınız</small></header>

<div id="slogan">"Hayalindeki tatili Palmiye Kaptan'la keşfet!"</div>

<div id="chat-box" aria-live="polite"></div>

<div class="input-area">

<input type="text" id="user-input" placeholder="Tatil sorunuzu yazın..." />

<button id="send-button">Gönder</button>

<div class="voice-toggle" id="voice-toggle">🔊 Ses: Açık</div>

</div>

<div id="tatilpuan">TatilPuan: 0</div>

<div id="mini-game">

🕹️ Oyun: "Palmiye Tahmin"

<br>Soru: En ünlü tatil adası hangisi?

<input id="game-answer" placeholder="Cevabınızı yazın" />

<button onclick="checkGameAnswer()">Yanıtla</button>

</div>

</div>

<script>

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

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

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

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

const tatilpuanEl = document.getElementById("tatilpuan");

let tatilPuan = 0;

let voiceEnabled = true;

function detectLanguage(text) {

return /[ğüşıöçĞÜŞİÖÇ]/.test(text) ? "tr" : "en";

}

function speak(text, lang) {

if (!voiceEnabled) return;

const utterance = new SpeechSynthesisUtterance(text);

utterance.lang = lang === "tr" ? "tr-TR" : "en-US";

speechSynthesis.speak(utterance);

}

voiceToggle.onclick = () => {

voiceEnabled = !voiceEnabled;

voiceToggle.textContent = voiceEnabled ? "🔊 Ses: Açık" : "🔇 Ses: Kapalı";

};

async function sendMessage() {

const userMessage = input.value.trim();

if (!userMessage) return;

const lang = detectLanguage(userMessage);

chatBox.innerHTML += `<p class='user-msg'>Sen: ${userMessage}</p>`;

input.value = "";

try {

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

method: "POST",

headers: {

"Content-Type": "application/json",

"Authorization": "Bearer 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"

},

body: JSON.stringify({

model: "openai/gpt-3.5-turbo",

messages: [{ role: "user", content: userMessage }]

})

});

const data = await response.json();

const reply = data?.choices?.[0]?.message?.content || "❌ Cevap alınamadı.";

chatBox.innerHTML += `<p class='bot-msg'>Palmiye Kaptan: ${reply}</p>`;

chatBox.scrollTop = chatBox.scrollHeight;

tatilPuan += 10;

tatilpuanEl.textContent = `TatilPuan: ${tatilPuan}`;

speak(reply, lang);

} catch (error) {

chatBox.innerHTML += `<p class='bot-msg'>❗ Hata oluştu: ${error.message}</p>`;

}

}

sendBtn.onclick = sendMessage;

input.addEventListener("keypress", function (e) {

if (e.key === "Enter") sendMessage();

});

function checkGameAnswer() {

const answer = document.getElementById("game-answer").value.trim().toLowerCase();

if (answer.includes("bali") || answer.includes("maldiv") || answer.includes("seyşel")) {

tatilPuan += 50;

tatilpuanEl.textContent = `TatilPuan: ${tatilPuan}`;

alert("Doğru cevap! 50 TatilPuan kazandınız!");

} else {

alert("Yanlış cevap. Şansınızı tekrar deneyin!");

}

}

</script>

</body>

</html>

<!DOCTYPE html>

<html lang="tr">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<title>🌴 TatilKaptanı - Yapay Zeka Tatil Planlayıcı</title>

<style>

body {

margin: 0;

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

background: linear-gradient(to bottom, #a7ffeb, #00897b);

color: #004d40;

min-height: 100vh;

display: flex;

justify-content: center;

align-items: center;

}

.container {

background: white;

width: 420px;

max-width: 100vw;

height: 700px;

border-radius: 15px;

box-shadow: 0 0 20px rgba(0,0,0,0.2);

display: flex;

flex-direction: column;

position: relative;

}

header {

background: #00796b;

color: white;

padding: 15px;

border-radius: 15px 15px 0 0;

text-align: center;

font-size: 24px;

font-weight: 700;

}

#chat-box {

flex-grow: 1;

padding: 10px;

background: #e0f2f1;

overflow-y: auto;

border-left: 6px solid #004d40;

border-right: 6px solid #004d40;

}

#chat-box p {

margin: 8px 0;

line-height: 1.4;

}

.user-msg {

text-align: right;

font-weight: 600;

color: #00796b;

}

.bot-msg {

text-align: left;

color: #004d40;

}

.input-area {

display: flex;

padding: 10px;

background: #004d40;

border-radius: 0 0 15px 15px;

gap: 10px;

}

#user-input {

flex-grow: 1;

padding: 10px;

border-radius: 8px;

border: none;

font-size: 16px;

}

#send-button {

background: #a7ffeb;

border: none;

border-radius: 8px;

padding: 0 20px;

font-weight: 700;

cursor: pointer;

}

#captain {

position: fixed;

top: 20px;

right: 20px;

width: 120px;

user-select: none;

z-index: 9999;

cursor: move;

}

#tatilpuan {

position: fixed;

bottom: 20px;

left: 20px;

background: #00796b;

color: white;

padding: 10px 15px;

border-radius: 20px;

font-weight: 700;

box-shadow: 0 0 10px #004d40;

z-index: 9998;

}

.voice-toggle {

color: white;

cursor: pointer;

font-size: 14px;

}

#slogan {

text-align: center;

font-size: 14px;

padding: 5px 10px;

font-style: italic;

background: #b2dfdb;

color: #004d40;

}

#membership {

padding: 10px;

background: #e0f2f1;

border-top: 2px solid #004d40;

text-align: center;

}

#membership button {

margin: 5px;

padding: 8px 12px;

background-color: #00796b;

color: white;

border: none;

border-radius: 6px;

cursor: pointer;

}

#membership button:hover {

background-color: #004d40;

}

</style>

</head>

<body>

<img id="captain" src="https://upload.wikimedia.org/wikipedia/commons/thumb/5/51/Cartoon\_ship\_captain.png/240px-Cartoon\_ship\_captain.png" alt="Palmiye Kaptan" title="Palmiye Kaptan" />

<div class="container">

<header>🌴 Palmiye Kaptan<br><small>Sizin sanal tatil asistanınız</small></header>

<div id="slogan">"Hayalindeki tatili Palmiye Kaptan'la keşfet!"</div>

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

<div class="input-area">

<input type="text" id="user-input" placeholder="Tatil sorunuzu yazın..." />

<button id="send-button">Gönder</button>

<div class="voice-toggle" id="voice-toggle">🔊 Ses: Açık</div>

</div>

<div id="membership">

<h3 style="margin-top: 0;">PalmClub Üyelik</h3>

<p>Mevcut Seviye: <strong><span id="uye-seviyesi">Yok</span></strong></p>

<button onclick="yukselt('Bronz', 50)">🥉 Bronz (50 puan)</button>

<button onclick="yukselt('Gümüş', 100)">🥈 Gümüş (100 puan)</button>

<button onclick="yukselt('Altın', 200)">🥇 Altın (200 puan)</button>

</div>

<div id="tatilpuan">TatilPuan: 0</div>

</div>

<script>

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

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

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

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

const tatilpuanEl = document.getElementById("tatilpuan");

let tatilPuan = 0;

let voiceEnabled = true;

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

let offset = { x: 0, y: 0 };

let isDragging = false;

captain.addEventListener("mousedown", function (e) {

isDragging = true;

offset = {

x: e.clientX - captain.offsetLeft,

y: e.clientY - captain.offsetTop

};

});

document.addEventListener("mouseup", () => isDragging = false);

document.addEventListener("mousemove", function (e) {

if (isDragging) {

captain.style.left = e.clientX - offset.x + "px";

captain.style.top = e.clientY - offset.y + "px";

}

});

function detectLanguage(text) {

return /[ğüşıöçĞÜŞİÖÇ]/.test(text) ? "tr" : "en";

}

function speak(text, lang) {

if (!voiceEnabled) return;

const utterance = new SpeechSynthesisUtterance(text);

utterance.lang = lang === "tr" ? "tr-TR" : "en-US";

speechSynthesis.speak(utterance);

}

voiceToggle.onclick = () => {

voiceEnabled = !voiceEnabled;

voiceToggle.textContent = voiceEnabled ? "🔊 Ses: Açık" : "🔇 Ses: Kapalı";

};

function sendMessage() {

const userMessage = input.value.trim();

if (!userMessage) return;

const lang = detectLanguage(userMessage);

chatBox.innerHTML += `<p class='user-msg'>Sen: ${userMessage}</p>`;

input.value = "";

const reply = "Harika soru! TatilPuan kazandın!";

chatBox.innerHTML += `<p class='bot-msg'>🧭 Palmiye Kaptan: ${reply}</p>`;

chatBox.scrollTop = chatBox.scrollHeight;

tatilPuan += 10;

tatilpuanEl.textContent = `TatilPuan: ${tatilPuan}`;

speak(reply, lang);

}

sendBtn.onclick = sendMessage;

input.addEventListener("keypress", function (e) {

if (e.key === "Enter") sendMessage();

});

function yukselt(seviye, puanGerekli) {

if (tatilPuan >= puanGerekli) {

tatilPuan -= puanGerekli;

tatilpuanEl.textContent = `TatilPuan: ${tatilPuan}`;

document.getElementById('uye-seviyesi').textContent = seviye;

alert(`🎉 Tebrikler! ${seviye} üyelik kazandınız!`);

} else {

alert(`❗ Bu üyelik için yeterli puanınız yok. En az ${puanGerekli} puan gerekiyor.`);

}

}

</script>

</body>

</html>

<!DOCTYPE html>

<html lang="tr">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<title>Palmiye Kaptan - TatilKaptanı AI</title>

<style>

body {

background: #e0f7fa;

font-family: 'Segoe UI', sans-serif;

margin: 0;

padding: 0;

}

.container {

max-width: 600px;

margin: auto;

padding: 20px;

background: white;

min-height: 90vh;

display: flex;

flex-direction: column;

}

h2 {

color: #00796b;

}

#chat-box {

flex: 1;

border: 1px solid #ccc;

padding: 10px;

margin-bottom: 10px;

overflow-y: auto;

height: 400px;

background: #f1f1f1;

}

.input-group {

display: flex;

}

input[type="text"] {

flex: 1;

padding: 10px;

font-size: 16px;

}

button {

padding: 10px 15px;

font-size: 16px;

background-color: #00796b;

color: white;

border: none;

cursor: pointer;

}

.voice-toggle {

margin-top: 10px;

cursor: pointer;

color: #004d40;

}

#tatilpuan {

margin-top: 8px;

font-weight: bold;

color: #00695c;

}

</style>

</head>

<body>

<div class="container">

<h2>🌴 Palmiye Kaptan</h2>

<div id="chat-box"></div>

<div class="input-group">

<input type="text" id="user-input" placeholder="Tatil sorunuzu yazın..." />

<button id="send-button">Gönder</button>

</div>

<div id="tatilpuan">TatilPuan: 0 | Seviye: Bronz</div>

<div class="voice-toggle" id="voice-toggle">🔊 Ses: Açık</div>

</div>

<script>

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

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

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

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

const tatilpuanEl = document.getElementById("tatilpuan");

let voiceEnabled = true;

let tatilPuan = 0;

function getSeviye(puan) {

if (puan >= 300) return "Altın";

if (puan >= 100) return "Gümüş";

return "Bronz";

}

voiceToggle.onclick = () => {

voiceEnabled = !voiceEnabled;

voiceToggle.textContent = voiceEnabled ? "🔊 Ses: Açık" : "🔇 Ses: Kapalı";

};

function speak(text) {

if (!voiceEnabled) return;

const utterance = new SpeechSynthesisUtterance(text);

utterance.lang = "tr-TR";

speechSynthesis.speak(utterance);

}

async function sendMessage() {

const userMessage = input.value.trim();

if (!userMessage) return;

chatBox.innerHTML += `<p><strong>Sen:</strong> ${userMessage}</p>`;

input.value = "";

chatBox.scrollTop = chatBox.scrollHeight;

try {

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

method: "POST",

headers: {

"Authorization": "Bearer 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",

"Content-Type": "application/json"

},

body: JSON.stringify({

model: "openai/gpt-3.5-turbo",

messages: [{ role: "user", content: userMessage }]

})

});

const data = await response.json();

const reply = data?.choices?.[0]?.message?.content || "❌ Cevap alınamadı.";

chatBox.innerHTML += `<p><strong>Palmiye Kaptan:</strong> ${reply}</p>`;

tatilPuan += 10;

const seviye = getSeviye(tatilPuan);

tatilpuanEl.textContent = `TatilPuan: ${tatilPuan} | Seviye: ${seviye}`;

speak(reply);

chatBox.scrollTop = chatBox.scrollHeight;

} catch (error) {

chatBox.innerHTML += `<p><strong>Palmiye Kaptan:</strong> ❗ Hata oluştu: ${error.message}</p>`;

}

}

sendBtn.onclick = sendMessage;

input.addEventListener("keypress", function (e) {

if (e.key === "Enter") sendMessage();

});

</script>

</body>

</html>

<!DOCTYPE html>

<html lang="tr">

<head>

  <meta charset="UTF-8" />

  <meta name="viewport" content="width=device-width, initial-scale=1" />

  <title>🌴 TatilKaptanı AI</title>

  <style>

    body {

      margin: 0;

      font-family: 'Segoe UI', sans-serif;

      background: url('https://www.biraztatil.com/uploads/content/webp/apollon-athena-tapinagi-67c5822f7cb6d-276.webp') no-repeat center center fixed;

      background-size: cover;

      backdrop-filter: brightness(0.9);

    }

    .container {

      max-width: 700px;

      margin: 20px auto;

      background: rgba(255,255,255,0.9);

      border-radius: 12px;

      padding: 20px;

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

      position: relative;

    }

    h2 {

      text-align: center;

      color: #004d40;

    }

    #slogan {

      text-align: center;

      font-style: italic;

      margin-bottom: 10px;

    }

    #chat-box {

      height: 300px;

      overflow-y: auto;

      background: #f1f1f1;

      border: 1px solid #ccc;

      padding: 10px;

      margin-bottom: 10px;

    }

    .input-group {

      display: flex;

    }

    input[type="text"] {

      flex: 1;

      padding: 10px;

    }

    button {

      background-color: #00796b;

      color: white;

      border: none;

      padding: 10px 15px;

      cursor: pointer;

    }

    #captain-avatar {

      position: fixed;

      top: 20px;

      right: 20px;

      width: 80px;

      cursor: move;

    }

    #tatilpuan {

      margin-top: 10px;

      font-weight: bold;

      color: #004d40;

    }

    .voice-toggle {

      margin-top: 5px;

      cursor: pointer;

      color: #00796b;

      font-size: 14px;

    }

    .reklam {

      position: fixed;

      width: 100px;

      bottom: 10px;

    }

    .reklam.left { left: 10px; }

    .reklam.right { right: 10px; }

    .reklam a {

      display: block;

      background: #004d40;

      color: white;

      padding: 10px;

      text-align: center;

      text-decoration: none;

      border-radius: 6px;

    }

  </style>

</head>

<body>

  <img id="captain-avatar" src="https://upload.wikimedia.org/wikipedia/commons/thumb/5/51/Cartoon\_ship\_captain.png/240px-Cartoon\_ship\_captain.png" alt="Kaptan" />

  <div class="container">

    <h2>🌴 Palmiye Kaptan</h2>

    <div id="slogan">"Yapay Zekâ ile Tatilin Yeni Rotasını Keşfet!"</div>

    <div id="chat-box"></div>

    <div class="input-group">

      <input type="text" id="user-input" placeholder="Tatil sorunuzu yazın..." />

      <button id="send-button">Gönder</button>

    </div>

    <div class="voice-toggle" id="voice-toggle">🔊 Ses: Açık</div>

    <div id="tatilpuan">TatilPuan: 0 (Üyelik: Bronz)</div>

  </div>

  <div class="reklam left"><a href="https://hotels.com" target="\_blank">Hotels</a></div>

  <div class="reklam right"><a href="https://booking.com" target="\_blank">Booking</a></div>

  <script>

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

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

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

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

    const tatilpuanDiv = document.getElementById("tatilpuan");

    let tatilPuan = 0;

    let voiceEnabled = true;

    const slogans = [

      "Yapay Zekâ ile Tatilin Yeni Rotasını Keşfet!",

      "Tatilin Kaptanı Yapay Zekâ!",

      "Kalbini Dinle, Palmiye Kaptan Haritayı Çoktan Çizdi.",

      "Hayalindeki tatili Palmiye Kaptan'la planla!"

    ];

    document.getElementById("slogan").textContent = slogans[Math.floor(Math.random() \* slogans.length)];

    voiceToggle.onclick = () => {

      voiceEnabled = !voiceEnabled;

      voiceToggle.textContent = voiceEnabled ? "🔊 Ses: Açık" : "🔇 Ses: Kapalı";

    };

    function speak(text) {

      if (!voiceEnabled) return;

      const utterance = new SpeechSynthesisUtterance(text);

      utterance.lang = "tr-TR";

      speechSynthesis.speak(utterance);

    }

    function getMembershipLevel(score) {

      if (score >= 100) return "Altın";

      if (score >= 50) return "Gümüş";

      return "Bronz";

    }

    async function sendMessage() {

      const userMessage = input.value.trim();

      if (!userMessage) return;

      input.value = "";

      chatBox.innerHTML += `<p><strong>Sen:</strong> ${userMessage}</p>`;

      let reply = "";

      const lc = userMessage.toLowerCase();

      if (lc.includes("sanal tatil")) {

        const yer = prompt("Hangi bölgede sanal tatil yapmak istersiniz?");

        const sure = prompt("Tatil süresi kaç gün?");

        reply = `🌴 Sanal Tatil Başladı! ${yer} bölgesinde ${sure} günlük maceran başlıyor. Plajda güneşlenirken AI rehber eşliğinde yerel kültürü keşfetmeye hazır ol!`;

      } else if (lc.includes("oyun")) {

        reply = `🎲 Oyun başladı! Şu 3 soruya cevap ver:

1) Bu kum tanesi mi yoksa antik harita mı?

2) Kokos Adası'nda en iyi kahveyi hangi yengeç yapar?

3) Ay'a yüzerek gitsem kaç kalori yakarım?

Palmiye Kaptan: Tebrikler! 15 PalmCoin kazandın!`;

        tatilPuan += 15;

      } else {

        // Yapay zeka sohbet

        try {

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

            method: "POST",

            headers: {

              "Authorization": "Bearer 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",

              "Content-Type": "application/json"

            },

            body: JSON.stringify({

              model: "openai/gpt-3.5-turbo",

              messages: [{ role: "user", content: userMessage }]

            })

          });

          const data = await res.json();

          reply = data?.choices?.[0]?.message?.content || "❌ Cevap alınamadı.";

          tatilPuan += 10;

        } catch (e) {

          reply = "❌ API hatası: " + e.message;

        }

      }

      const level = getMembershipLevel(tatilPuan);

      chatBox.innerHTML += `<p><strong>Palmiye Kaptan:</strong> ${reply}</p>`;

      tatilpuanDiv.textContent = `TatilPuan: ${tatilPuan} (Üyelik: ${level})`;

      chatBox.scrollTop = chatBox.scrollHeight;

      speak(reply);

    }

    sendBtn.onclick = sendMessage;

    input.addEventListener("keypress", e => {

      if (e.key === "Enter") sendMessage();

    });

    // Avatar taşınabilirlik

    const avatar = document.getElementById("captain-avatar");

    let isDragging = false;

    avatar.addEventListener("mousedown", function(e) {

      isDragging = true;

      offsetX = e.offsetX;

      offsetY = e.offsetY;

    });

    document.addEventListener("mouseup", function() {

      isDragging = false;

    });

    document.addEventListener("mousemove", function(e) {

      if (isDragging) {

        avatar.style.left = (e.pageX - offsetX) + 'px';

        avatar.style.top = (e.pageY - offsetY) + 'px';

      }

    });

  </script>

</body>

</html>

<!DOCTYPE html>

<html lang="tr">

<head>

  <meta charset="UTF-8" />

  <meta name="viewport" content="width=device-width, initial-scale=1" />

  <title>🌴 TatilKaptanı - Yapay Zeka Tatil Planlayıcı</title>

  <style>

    body {

      margin: 0;

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

      background-image: url('https://www.biraztatil.com/uploads/content/webp/apollon-athena-tapinagi-67c5822f7cb6d-276.webp');

      background-size: cover;

      background-attachment: fixed;

      background-position: center;

      background-repeat: no-repeat;

      background-color: rgba(255, 255, 255, 0.85);

      background-blend-mode: lighten;

    }

    .container {

      max-width: 720px;

      margin: 40px auto;

      padding: 20px;

      background: rgba(255,255,255,0.9);

      border-radius: 10px;

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

    }

    header {

      background: #00796b;

      color: white;

      padding: 20px;

      text-align: center;

      font-size: 22px;

      font-weight: bold;

      border-radius: 10px 10px 0 0;

    }

    #chat-box {

      height: 300px;

      overflow-y: auto;

      padding: 10px;

      background: #e0f2f1;

      border: 1px solid #ccc;

      margin-bottom: 10px;

    }

    .user-msg {

      text-align: right;

      color: #00796b;

      font-weight: bold;

    }

    .bot-msg {

      text-align: left;

      color: #004d40;

    }

    .input-area {

      display: flex;

      gap: 10px;

    }

    input[type="text"] {

      flex: 1;

      padding: 10px;

      font-size: 16px;

      border-radius: 5px;

      border: 1px solid #ccc;

    }

    button {

      padding: 10px 15px;

      font-size: 16px;

      background-color: #00796b;

      color: white;

      border: none;

      cursor: pointer;

      border-radius: 5px;

    }

    .voice-toggle, .slogan, #tatilpuan, #reklamlar {

      margin-top: 10px;

    }

    .voice-toggle {

      cursor: pointer;

      font-size: 14px;

      color: #004d40;

    }

    #tatilpuan {

      font-weight: bold;

      color: #00796b;

    }

    #captain {

      position: fixed;

      top: 10px;

      right: 10px;

      width: 100px;

      z-index: 1000;

      cursor: move;

    }

    .slogan {

      font-style: italic;

      text-align: center;

      padding: 8px;

      background: #b2dfdb;

      color: #004d40;

      border-radius: 5px;

    }

    #reklamlar {

      display: flex;

      justify-content: space-between;

      margin-top: 15px;

    }

    .reklam-kutu {

      width: 48%;

      background: #f1f1f1;

      padding: 10px;

      border-radius: 5px;

      text-align: center;

      font-size: 14px;

    }

    .reklam-kutu a {

      text-decoration: none;

      color: #00695c;

    }

  </style>

</head>

<body>

  <img id="captain" src="https://upload.wikimedia.org/wikipedia/commons/thumb/5/51/Cartoon\_ship\_captain.png/240px-Cartoon\_ship\_captain.png" alt="Palmiye Kaptan" title="Palmiye Kaptan" />

  <div class="container">

    <header>🌴 Palmiye Kaptan - Tatil Zeka Asistanı</header>

    <div class="slogan" id="slogan"></div>

    <div id="chat-box"></div>

    <div class="input-area">

      <input type="text" id="user-input" placeholder="Tatil sorunuzu yazın... (örn: Paris’te sanal tatil, oyun, vb)" />

      <button id="send-button">Gönder</button>

    </div>

    <div class="voice-toggle" id="voice-toggle">🔊 Ses: Açık</div>

    <div id="tatilpuan">TatilPuan: 0 - Üyelik: Bronz</div>

    <div id="reklamlar">

      <div class="reklam-kutu"><a href="https://www.booking.com" target="\_blank">Booking ile Yurt Dışı Otel</a></div>

      <div class="reklam-kutu"><a href="https://www.hotels.com" target="\_blank">Hotels.com ile Yurt İçi Tatil</a></div>

    </div>

  </div>

  <script>

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

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

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

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

    const tatilpuanEl = document.getElementById("tatilpuan");

    const sloganBox = document.getElementById("slogan");

    let tatilPuan = 0;

    let voiceEnabled = true;

    const slogans = [

      "Yapay Zeka ile Tatilin Yeni Rotasını Keşfet!",

      "TatilKaptanı: Teknolojiyle Tatilini Yöneten Kaptan!",

      "Palmiye Kaptan Seni Maceraya Götürüyor!",

      "Gizli Rotan Hazır, Sadece Sor!",

      "TatilPuan’la Üyelik Seviyeni Yükselt!"

    ];

    sloganBox.textContent = slogans[Math.floor(Math.random() \* slogans.length)];

    function updateMembership() {

      let seviye = "Bronz";

      if (tatilPuan >= 50) seviye = "Gümüş";

      if (tatilPuan >= 100) seviye = "Altın";

      tatilpuanEl.textContent = `TatilPuan: ${tatilPuan} - Üyelik: ${seviye}`;

    }

    function speak(text) {

      if (!voiceEnabled) return;

      const utter = new SpeechSynthesisUtterance(text);

      utter.lang = "tr-TR";

      speechSynthesis.speak(utter);

    }

    voiceToggle.onclick = () => {

      voiceEnabled = !voiceEnabled;

      voiceToggle.textContent = voiceEnabled ? "🔊 Ses: Açık" : "🔇 Ses: Kapalı";

    };

    async function sendMessage() {

      const userMessage = input.value.trim();

      if (!userMessage) return;

      chatBox.innerHTML += `<p class='user-msg'>Sen: ${userMessage}</p>`;

      input.value = "";

      let botReply = "";

      if (userMessage.toLowerCase().includes("oyun")) {

        botReply = "Harika! İşte 3 soruluk tatil oyunu başladı:\n1) Türkiye’nin en çok ziyaret edilen tatil bölgesi nedir?";

        tatilPuan += 10;

      } else if (userMessage.toLowerCase().includes("paris")) {

        botReply = "🎬 Sanal Tatil Başladı: Paris\nEyfel Kulesi altındasın, çikolatalı kruvasan elinde, sokak sanatçıları müzik yapıyor...";

        tatilPuan += 20;

      } else {

        try {

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

            method: "POST",

            headers: {

              "Authorization": "Bearer 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",

              "Content-Type": "application/json"

            },

            body: JSON.stringify({

              model: "openai/gpt-3.5-turbo",

              messages: [{ role: "user", content: userMessage }]

            })

          });

          const data = await response.json();

          botReply = data?.choices?.[0]?.message?.content || "❌ Cevap alınamadı.";

        } catch (e) {

          botReply = "❌ Hata: API erişiminde problem yaşandı.";

        }

      }

      chatBox.innerHTML += `<p class='bot-msg'>🧭 Palmiye Kaptan: ${botReply}</p>`;

      chatBox.scrollTop = chatBox.scrollHeight;

      updateMembership();

      speak(botReply);

    }

    sendBtn.onclick = sendMessage;

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

      if (e.key === "Enter") sendMessage();

    });

    // Kaptanı sürükleme

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

    let offset = { x: 0, y: 0 }, isDragging = false;

    captain.addEventListener("mousedown", function (e) {

      isDragging = true;

      offset = {

        x: e.clientX - captain.offsetLeft,

        y: e.clientY - captain.offsetTop

      };

    });

    document.addEventListener("mouseup", () => { isDragging = false; });

    document.addEventListener("mousemove", function (e) {

      if (isDragging) {

        captain.style.left = e.clientX - offset.x + "px";

        captain.style.top = e.clientY - offset.y + "px";

      }

    });

  </script>

</body>

</html>

<!DOCTYPE html>

<html lang="tr">

<head>

  <meta charset="UTF-8" />

  <meta name="viewport" content="width=device

  <style>

    body {

      margin: 0;

      font-family: 'Segoe UI', sans-serif;

      background: linear-gradient(to bottom, #e0f7fa, #b2dfdb);

      background-image: url('https://www.biraztatil.com/uploads/content/webp/apollon-athena-tapinagi-67c5822f7cb6d-276.webp');

      background-size: cover;

      background-attachment: fixed;

      background-position: center;

      color: #004d40;

    }

    .container {

      max-width: 600px;

      margin: auto;

      padding: 20px;

      background-color: rgba(255,255,255,0.9);

      border-radius: 12px;

      min-height: 85vh;

      display: flex;

      flex-direction: column;

    }

    #chat-box {

      flex: 1;

      overflow-y: auto;

      background: #e0f2f1;

      padding: 10px;

      border-radius: 8px;

      margin-bottom: 10px;

    }

    .input-group {

      display: flex;

      margin-bottom: 10px;

    }

    input[type="text"] {

      flex: 1;

      padding: 10px;

      font-size: 16px;

      border: 1px solid #ccc;

      border-radius: 6px;

    }

    button {

      padding: 10px 15px;

      font-size: 16px;

      background-color: #00796b;

      color: white;

      border: none;

      border-radius: 6px;

      cursor: pointer;

    }

    #tatilpuan {

      background-color: #00796b;

      color: white;

      padding: 8px 12px;

      border-radius: 20px;

      font-weight: bold;

      text-align: center;

      margin-bottom: 10px;

    }

    .voice-toggle {

      text-align: right;

      font-size: 14px;

      cursor: pointer;

      color: #00796b;

    }

    #slogan {

      font-style: italic;

      text-align: center;

      margin-bottom: 10px;

    }

  </style>

</head>

<body>

  <div class="container">

    <h2>🌴 Palmiye Kaptan</h2>

    <div id="slogan"></div>

    <div id="tatilpuan">TatilPuan: 0</div>

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

    <div class="input-group">

      <input type="text" id="user-input" placeholder="Bir tatil sorusu, rota ya da 'oyun' yazın..." />

      <button id="send-button">Gönder</button>

    </div>

    <div class="voice-toggle" id="voice-toggle">🔊 Ses: Açık</div>

  </div>

<script>

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

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

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

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

  const tatilpuanEl = document.getElementById("tatilpuan");

  let voiceEnabled = true;

  let tatilPuan = 0;

  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!"

  ];

  document.getElementById("slogan").innerText = sloganList[Math.floor(Math.random() \* sloganList.length)];

  voiceToggle.onclick = () => {

    voiceEnabled = !voiceEnabled;

    voiceToggle.textContent = voiceEnabled ? "🔊 Ses: Açık" : "🔇 Ses: Kapalı";

  };

  function speak(text) {

    if (!voiceEnabled) return;

    const utterance = new SpeechSynthesisUtterance(text);

    utterance.lang = "tr-TR";

    speechSynthesis.speak(utterance);

  }

  async function sendMessage() {

    const userMessage = input.value.trim();

    if (!userMessage) return;

    chatBox.innerHTML += `<p><strong>Sen:</strong> ${userMessage}</p>`;

    input.value = "";

    // Özel oyun ve rota komutları burada kontrol edilecek

    if (userMessage.toLowerCase() === "oyun") {

      startGame();

      return;

    }

    if (userMessage.toLowerCase().startsWith("sanal")) {

      startSanalTatil(userMessage);

      return;

    }

    try {

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

        method: "POST",

        headers: {

          "Authorization": "Bearer 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",

          "Content-Type": "application/json"

        },

        body: JSON.stringify({

          model: "openai/gpt-3.5-turbo",

          messages: [{ role: "user", content: userMessage }]

        })

      });

      const data = await response.json();

      const reply = data?.choices?.[0]?.message?.content || "❌ Cevap alınamadı.";

      chatBox.innerHTML += `<p><strong>Palmiye Kaptan:</strong> ${reply}</p>`;

      chatBox.scrollTop = chatBox.scrollHeight;

      speak(reply);

      tatilPuan += 10;

      tatilpuanEl.textContent = `TatilPuan: ${tatilPuan}`;

    } catch (error) {

      chatBox.innerHTML += `<p><strong>Palmiye Kaptan:</strong> ❗ Hata oluştu: ${error.message}</p>`;

    }

  }

  sendBtn.onclick = sendMessage;

  input.addEventListener("keypress", function (e) {

    if (e.key === "Enter") sendMessage();

  });

  function startGame() {

    chatBox.innerHTML += `<p><strong>Palmiye Kaptan:</strong> Oyuna hoş geldin! İlk sorum geliyor...</p>`;

    setTimeout(() => {

      chatBox.innerHTML += `<p><strong>Palmiye Kaptan:</strong> Soru 1: Ay'a yüzerek gitmeye kalksan kaç kalori yakarsın? (A) 10k (B) 1 milyon (C) Çok daha fazla</p>`;

    }, 800);

  }

</script>

</body>

</html>

      border-radius: 0 5px 5px 0;

      cursor: pointer;

    }

    #chat-box {

      background: #e0f2f1;

      padding: 10px;

      height: 300px;

      overflow-y: auto;

      border: 1px solid #ccc;

      margin-top: 10px;

    }

    .user { text-align: right; font-weight: bold; }

    .bot { text-align: left; font-style: italic; }

    .badge {

      display: inline-block;

      padding:

      display: inline-block;

      padding: 5px 10px;

      background-color: #004d40;

      color: white;

      border-radius: 12px;

      margin-top: 10px;

    }

    .status-bar {

      display: flex;

      justify-content: space-between;

      align-items: center;

      font-size: 14px;

      margin-top: 5px;

      color: #00695c;

    }

    .slogan {

      text-align: center;

      font-style: italic;

      font-size: 14px;

      margin-bottom

      margin-bottom: 15px;

      background: #b2dfdb;

      padding: 8px;

      border-radius: 5px;

    }

  </style>

</head>

<body>

  <div class="container">

    <h1>🌴 TatilKaptanı AI</h1>

    <div class="slogan" id="slogan-box">"Hayalindeki tatili Palmiye Kaptan'la planla!"</div>

    <div class="status-bar">

      <span id="tatilpuan">TatilPuan: 0

<!DOCTYPE html>

<html lang="tr">

<head>

<meta charset="UTF-8" />

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

<title>Palmiye Kaptan AI - Tatil Asistanınız</title>

<link href="https://fonts.googleapis.com/css2?family=Segoe+UI:wght@400;700&display=swap" rel="stylesheet">

<!-- Font Awesome for Icons -->

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

<style>

/\* Genel Stil Ayarları \*/

body {

margin: 0;

font-family: 'Segoe UI', sans-serif;

background: linear-gradient(to bottom, #e0f7fa, #b2dfdb);

background-image: url('https://www.biraztatil.com/uploads/content/webp/apollon-athena-tapinagi-67c5822f7cb6d-276.webp');

background-size: cover;

background-attachment: fixed;

background-position: center;

color: #004d40;

display: flex;

min-height: 100vh;

overflow: hidden; /\* No scrolling on body, content scrolls inside \*/

}

/\* Ana Düzen (Flexbox) \*/

.main-layout {

display: flex;

width: 100%;

height: 100vh;

}

/\* Sol Menü \*/

.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;

}

/\* Ana İçerik Alanı \*/

.content-area {

flex: 1;

display: flex;

flex-direction: column;

background-color: rgba(255, 255, 255, 0.95);

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

border-radius: 15px;

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

position: relative; /\* Avatar için \*/

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

}

/\* Üst Bar \*/

.top-bar {

padding: 15px 25px;

background-color: #00796b; /\* Ana tema rengi \*/

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;

}

/\* İçerik Bölümleri (Sekmeler gibi) \*/

.content-section {

flex: 1;

padding: 25px;

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

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

flex-direction: column;

background-color: #f9f9f9; /\* Hafif bir arka plan \*/

border-bottom-left-radius: 15px;

border-bottom-right-radius: 15px;

}

.content-section.active {

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

}

/\* Sohbet Bölümü Özel Stiller \*/

#chat-section {

display: flex;

flex-direction: column;

}

#chat-box {

flex: 1;

background: #e0f2f1;

padding: 15px;

border-radius: 10px;

margin-bottom: 15px;

border: 1px solid #c8e6c9;

box-shadow: inset 0 1px 3px rgba(0, 0, 0, 0.1);

overflow-y: auto;

min-height: 250px; /\* Minimum sohbet kutusu yüksekliği \*/

}

.message {

display: flex;

margin-bottom: 12px;

max-width: 85%;

word-wrap: break-word;

line-height: 1.5;

}

.message strong {

margin-right: 8px;

flex-shrink: 0;

font-weight: 700;

}

.message.user {

align-self: flex-end;

background-color: #dcedc8;

color: #2e7d32;

border-radius: 18px 18px 5px 18px;

padding: 10px 15px;

box-shadow: 0 2px 5px rgba(0, 0, 0, 0.05);

}

.message.bot {

align-self: flex-start;

background-color: #bbdefb;

color: #1565c0;

border-radius: 18px 18px 18px 5px;

padding: 10px 15px;

box-shadow: 0 2px 5px rgba(0, 0, 0, 0.05);

}

.message img {

max-width: 100%;

height: auto;

border-radius: 8px;

margin-top: 10px;

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

}

.message a {

color: #00796b;

text-decoration: none;

font-weight: bold;

}

.message a:hover {

text-decoration: underline;

}

.input-group {

display: flex;

gap: 10px;

margin-top: 15px;

}

input[type="text"] {

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 {

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);

}

/\* Özellik Bölümleri Ortak Stil \*/

.feature-section {

padding: 20px;

background-color: #e0f2f1;

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 input[type="text"],

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

.feature-form select {

width: calc(100% - 22px);

padding: 10px;

margin-bottom: 15px;

border: 1px solid #ccc;

border-radius: 5px;

font-size: 1em;

}

.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: #f0fdfc;

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://i.ibb.co/L9513X9/palmiye-kaptan-avatar.png');

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); }

}

/\* Reklam Alanları \*/

.right-sidebar {

width: 160px; /\* Reklamlar için genişlik \*/

padding: 20px 0;

display: flex;

flex-direction: column;

gap: 20px;

z-index: 90;

background-color: rgba(255, 255, 255, 0.85); /\* Hafif şeffaf arka plan \*/

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

border-top-left-radius: 15px;

border-bottom-left-radius: 15px;

}

.ad-area {

width: 100%;

height: 180px; /\* Reklam yüksekliği \*/

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; /\* Resimlerin taşmasını engeller \*/

}

.ad-area:hover {

transform: translateY(-3px);

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

}

.ad-area img {

max-width: 90%;

max-height: 70%;

object-fit: contain;

margin-bottom: 5px;

}

.ad-area p {

margin: 0;

padding: 0 5px;

font-size: 0.9em;

}

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

.modal {

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

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

z-index: 1000; /\* 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: 1px solid #888;

border-radius: 10px;

width: 80%; /\* Could be more responsive \*/

max-width: 500px;

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

text-align: center;

}

.modal-content h3 {

color: #004d40;

margin-bottom: 20px;

}

.modal-content button {

background-color: #00796b;

color: white;

padding: 10px 20px;

border: none;

border-radius: 5px;

cursor: pointer;

margin-top: 20px;

}

/\* Loading Indicator \*/

.loading-indicator {

text-align: center;

margin-top: 10px;

color: #00796b;

font-weight: bold;

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

}

/\* 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; /\* Sloganı ortaya al \*/

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; /\* Mobil'de reklamları gizle \*/

}

.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 {

width: calc(100% - 20px);

}

}

</style>

</head>

<body>

<!-- Palmiye Kaptan Avatarı -->

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

<!-- Genel Modal (Alert/Confirm yerine) -->

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

<div class="modal-content">

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

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

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

</div>

</div>

<!-- Ana Düzen -->

<div class="main-layout">

<!-- Sol Menü -->

<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="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>

</ul>

</nav>

</div>

<!-- Ana İçerik Alanı -->

<div class="content-area">

<!-- Üst Bar -->

<div class="top-bar">

<div class="info-group">

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

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

<!-- Dil seçeneği (şimdilik demo, AI dili algılayacak) -->

<select id="language-select" style="background-color: #004d40; color: white; border: none; padding: 5px; border-radius: 5px;">

<option value="tr">Türkçe</option>

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

<!-- Diğer diller eklenebilir -->

</select>

</div>

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

</div>

<!-- İçerik Bölümleri -->

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

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

<div id="chat-box" 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>Nereye gitmek istersiniz ve kaç gün kalacaksınız? AI sizin için o tatili 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" />

<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;"></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 sizin için planlıyoruz. Tatil stresini 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!</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>

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

<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</button>

</div>

<div class="feature-section">

<h3>Geçmiş Sorgular (Demo)</h3>

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

<li>(Bu kısım için gerçek bir geçmiş kaydı tutulmalı)</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 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" />

<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" />

<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>

</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" style="background-color: #f0f8ff; padding: 10px; border-radius: 8px; max-height: 200px; overflow-y: auto; margin-bottom: 10px;"></div>

<input type="text" id="companion-input" placeholder="Yoldaşınıza bir şeyler sorun..." style="width: calc(100% - 20px); padding: 10px; border-radius: 5px;">

<button id="send-companion-message-btn" style="margin-top: 10px;">Mesaj Gönder</button>

</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>

</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> <!-- /content-area -->

<!-- Sağ Reklam Panelleri -->

<div class="right-sidebar">

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

<img src="https://via.placeholder.com/120x80/00796b/FFFFFF?text=Booking.com" alt="Booking.com Reklamı">

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

</a>

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

<img src="https://via.placeholder.com/120x80/004d40/FFFFFF?text=Skyscanner" alt="Skyscanner Reklamı">

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

</a>

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

<img src="https://via.placeholder.com/120x80/b2dfdb/004d40?text=GetYourGuide" alt="GetYourGuide Reklamı">

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

</a>

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

<img src="https://via.placeholder.com/120x80/e0f7fa/004d40?text=TripAdvisor" alt="TripAdvisor Reklamı">

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

</a>

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

<img src="https://via.placeholder.com/120x80/00796b/FFFFFF?text=Rentalcars" alt="Rentalcars Reklamı">

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

</a>

</div>

</div> <!-- /main-layout -->

<script>

// API Anahtarı: Buraya kendi OpenRouter API anahtarınızı yapıştırın.

// DİKKAT: Gerçek bir uygulamada bu anahtar sunucu tarafında saklanmalıdır!

const OPENROUTER\_API\_KEY = "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";

// --- HTML Elementleri ---

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

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

// Game Section elements

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");

// Virtual Holiday Section elements

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

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

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

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

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

// VIP Planner Section elements

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 = 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 = ""; // For VIP planner

// User Info Section elements

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

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");

// Time Travel Section elements

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

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

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

// Destiny Route Section elements

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

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

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

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 Companion Section elements

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");

let aiCompanion = null; // To store companion details

// Payment Section elements

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");

// Modal elements

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

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

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

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

// --- Uygulama Durumu Değişkenleri ---

let voiceEnabled = true;

let tatilPuan = parseInt(localStorage.getItem('tatilPuan') || '0');

let userMembershipLevel = localStorage.getItem('userMembershipLevel') || "Bronz";

let userName = localStorage.getItem('userName') || "Misafir";

let gameActive = false;

let currentQuestionIndex = 0;

let gameScore = parseInt(localStorage.getItem('gameScore') || '0');

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!"

];

const gameQuestions = [

{

question: "Dünyanın en yüksek dağı hangisidir?",

options: ["(A) K2", "(B) Kilimanjaro", "(C) Everest"],

answer: "C",

points: 20

},

{

question: "Eyfel Kulesi hangi şehirdedir?",

options: ["(A) Londra", "(B) Paris", "(C) Roma"],

answer: "B",

points: 25

},

{

question: "Büyük Kanyon hangi ülkededir?",

options: ["(A) Kanada", "(B) ABD", "(C) Avustralya"],

answer: "B",

points: 30

}

];

// --- Uygulama Başlangıcı ---

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

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

updateTatilPuanDisplay();

displayMembershipInfo(); // Initial update for user info section

displayMessage("bot", "Merhaba! Ben Palmiye Kaptan, tatil asistanınım. Sol menüden bir özellik seçebilirsin.");

// Set initial active section and sidebar button

showSection("chat-section");

document.querySelector(".sidebar-nav button[data-section='chat-section']").classList.add("active");

});

// --- Helper Functions ---

/\*\*

\* Shows a custom modal dialog instead of alert/confirm.

\* @param {string} title - The title of the modal.

\* @param {string} message - The message content.

\* @returns {Promise<boolean>} Resolves true if confirmed.

\*/

function showModal(title, message) {

return new Promise((resolve) => {

modalTitle.textContent = title;

modalMessage.innerHTML = message; // Use innerHTML to allow basic formatting

appModal.style.display = "flex";

const handleConfirm = () => {

appModal.style.display = "none";

modalConfirmBtn.removeEventListener("click", handleConfirm);

resolve(true); // Always true for now, can be extended for confirm/cancel

};

modalConfirmBtn.addEventListener("click", handleConfirm);

});

}

/\*\*

\* Displays a message in the chat box.

\* @param {string} sender - 'user' or 'bot'.

\* @param {string} message - The message content (can contain HTML).

\*/

function displayMessage(sender, message) {

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

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

messageDiv.innerHTML = `<strong>${sender === "user" ? userName : "Palmiye Kaptan"}:</strong> ${message}`;

chatBox.appendChild(messageDiv);

chatBox.scrollTop = chatBox.scrollHeight;

}

/\*\*

\* Converts text to speech.

\* @param {string} text - The text to speak.

\*/

function speak(text) {

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

const utterance = new SpeechSynthesisUtterance(text);

utterance.lang = languageSelect.value + "-" + languageSelect.value.toUpperCase(); // e.g., tr-TR, en-US

speechSynthesis.speak(utterance);

}

/\*\*

\* Updates TatilPuan and user membership level.

\* @param {number} points - Points to add.

\*/

function updateTatilPuan(points) {

tatilPuan += points;

localStorage.setItem('tatilPuan', tatilPuan);

const oldLevel = userMembershipLevel;

if (tatilPuan < 100) {

userMembershipLevel = "Bronz";

} else if (tatilPuan < 200) {

userMembershipLevel = "Gümüş";

} else {

userMembershipLevel = "Altın";

}

localStorage.setItem('userMembershipLevel', userMembershipLevel);

updateTatilPuanDisplay();

displayMembershipInfo(); // Update user info section

if (oldLevel !== userMembershipLevel) {

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

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

}

}

/\*\*

\* Updates the TatilPuan display in the top bar.

\*/

function updateTatilPuanDisplay() {

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

}

/\*\*

\* Displays user membership information in the User Info section.

\*/

function displayMembershipInfo() {

displayUsername.textContent = userName;

displayTatilpuan.textContent = tatilPuan;

displayMembershipLevel.textContent = userMembershipLevel;

displayGameScore.textContent = gameScore;

}

/\*\*

\* Shows a specific content section and hides others.

\* @param {string} sectionId - The ID of the section to show.

\*/

function showSection(sectionId) {

contentSections.forEach(section => {

section.classList.remove("active");

section.style.display = "none"; // Ensure display is none for hidden sections

});

document.getElementById(sectionId).classList.add("active");

document.getElementById(sectionId).style.display = "flex"; // Ensure display is flex for active section

// Update active state in sidebar

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

document.querySelector(`.sidebar-nav button[data-section='${sectionId}']`).classList.add("active");

// Handle specific section initializations

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

checkVipAccess(vipAccessCheck, vipPlannerFormArea, "Altın");

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

checkVipAccess(timeTravelAccessCheck, timeTravelFormArea, "Altın");

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

displayMembershipInfo(); // Refresh user info

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

// Reset payment form fields

cardNumberInput.value = '';

expiryDateInput.value = '';

cvvInput.value = '';

cardHolderNameInput.value = '';

}

}

/\*\*

\* Checks if the user has VIP access and shows/hides content accordingly.

\* @param {HTMLElement} accessCheckEl - The element showing access denial.

\* @param {HTMLElement} formAreaEl - The element containing the feature form.

\* @param {string} requiredLevel - The minimum membership level required.

\*/

function checkVipAccess(accessCheckEl, formAreaEl, requiredLevel) {

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

const currentUserLevelIndex = levels.indexOf(userMembershipLevel);

const requiredLevelIndex = levels.indexOf(requiredLevel);

if (currentUserLevelIndex >= requiredLevelIndex) {

accessCheckEl.style.display = "none";

formAreaEl.style.display = "block"; // Or 'flex' depending on inner layout

} else {

accessCheckEl.style.display = "block";

formAreaEl.style.display = "none";

}

}

/\*\*

\* Makes an API call to OpenRouter.

\* @param {string} prompt - The user's prompt.

\* @param {string} model - The AI model to use.

\* @param {HTMLElement} loadingIndicator - The loading spinner element.

\* @param {function} onCompletion - Callback function on successful completion.

\* @returns {Promise<string>} The AI's response text.

\*/

async function callOpenRouterAI(prompt, model = "openai/gpt-3.5-turbo", loadingIndicator = null) {

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

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: [{ role: "user", content: prompt }]

})

});

if (!response.ok) {

const errorData = await response.json();

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

}

const data = await response.json();

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

} catch (error) {

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

showModal("Hata!", `Bir AI hatası oluştu: ${error.message}. Lütfen daha sonra tekrar deneyin.`);

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

} finally {

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

}

}

// --- Event Listeners ---

sidebarButtons.forEach(button => {

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

const sectionId = button.dataset.section;

showSection(sectionId);

});

});

sendChatBtn.onclick = async () => {

const userMessage = chatInput.value.trim();

if (!userMessage) return;

displayMessage("user", userMessage);

chatInput.value = "";

if (OPENROUTER\_API\_KEY === "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") {

showModal("API Hatası", "🚨 OpenRouter API anahtarınızı yapıştırmadınız veya geçersiz. Lütfen kodu düzenleyip anahtarınızı yapıştırın.");

speak("API anahtarınız eksik.");

return;

}

const prompt = `Kullanıcının mesajı: "${userMessage}". Bu bir tatil asistanı sohbeti. Mesajın dilini algıla ve o dilde yanıt ver. Tatil ve seyahat konularında yardımcı ol.`;

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

displayMessage("bot", reply);

speak(reply);

updateTatilPuan(5); // Her AI sohbetinde puan kazan

};

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

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

});

voiceToggleTop.onclick = () => {

voiceEnabled = !voiceEnabled;

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

speak(voiceEnabled ? "Sesli yanıtlar açıldı." : "Sesli yanıtlar kapatıldı.");

};

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

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

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

});

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

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

// --- Game Section Logic ---

startGameBtn.onclick = () => {

gameActive = true;

currentQuestionIndex = 0;

gameScore = 0;

localStorage.setItem('gameScore', gameScore); // Reset game score in local storage

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";

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 < gameQuestions.length) {

const q = gameQuestions[currentQuestionIndex];

const questionText = `Soru ${currentQuestionIndex + 1}: ${q.question} ${q.options.join(" ")}`;

gameOutput.innerHTML += `<p><strong>Palmiye Kaptan:</strong> ${questionText}</p>`;

gameAnswerInput.value = "";

speak(questionText);

} else {

endGame();

}

}

function handleGameAnswer(answer) {

const currentQ = gameQuestions[currentQuestionIndex];

if (!currentQ) return;

const userAnswer = answer.trim().toUpperCase();

if (userAnswer === currentQ.answer) {

gameOutput.innerHTML += `<p style="color: green;"><strong>Palmiye Kaptan:</strong> Tebrikler! Doğru cevap.</p>`;

speak("Tebrikler! Doğru cevap.");

gameScore += currentQ.points;

localStorage.setItem('gameScore', gameScore);

updateTatilPuan(currentQ.points);

} else {

gameOutput.innerHTML += `<p style="color: red;"><strong>Palmiye Kaptan:</strong> Yanlış cevap. Doğru cevap: ${currentQ.answer}</p>`;

speak(`Yanlış cevap. Doğru cevap ${currentQ.answer}`);

}

currentQuestionIndex++;

if (currentQuestionIndex < gameQuestions.length) {

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>`;

speak(`Oyun bitti! Toplam ${gameScore} PalmCoin kazandın!`);

gameAnswerInput.style.display = "none";

submitGameAnswerBtn.style.display = "none";

startGameBtn.style.display = "block";

updateTatilPuanDisplay();

displayMembershipInfo();

}

// --- Virtual Holiday Section Logic ---

startVirtualBtn.onclick = async () => {

const city = virtualCityInput.value.trim();

const days = parseInt(virtualDaysInput.value);

if (!city || isNaN(days) || days < 1) {

showModal("Eksik Bilgi", "Lütfen geçerli bir şehir/yer ve gün sayısı girin.");

return;

}

virtualHolidayOutput.style.display = "none"; // Hide previous output

const prompt = `Lütfen ${city} şehrinde ${days} günlük sanal bir tatil hikayesi yaz. Bu tatilde gezilecek yerleri, tadılacak lezzetleri ve yaşanacak deneyimleri detaylıca anlat. Hikaye ilgi çekici ve sürükleyici olsun. Ayrıca, bu tatili temsil edecek bir resim linki öner (örnek URL: 'https://source.unsplash.com/random/800x600/?${city}-tourism' gibi). Yanıtı Türkçe ver.`;

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

let holidayStory = reply;

const urlRegex = /(https?:\/\/[^\s]+\.(?:png|jpe?g|gif|webp|unsplash\.com\/\S+|pixabay\.com\/\S+))/i;

const match = holidayStory.match(urlRegex);

let mediaSuggestionHtml = "";

if (match) {

mediaSuggestionHtml = `<br><br><img src="${match[0]}" alt="${city} Tatili">`;

holidayStory = holidayStory.replace(match[0], '').trim();

} else {

const genericImageUrl = `https://source.unsplash.com/random/800x600/?${city.replace(/\s/g, '-')}-travel`;

mediaSuggestionHtml = `<br><br><img src="${genericImageUrl}" alt="${city} Tatili">`;

}

virtualHolidayOutput.innerHTML = `<h4>${city} - ${days} Günlük Sanal Tatil Hikayen:</h4><p>${holidayStory}</p>${mediaSuggestionHtml}`;

virtualHolidayOutput.style.display = "block";

speak(`İşte ${city} için sanal tatil hikayen.`);

updateTatilPuan(50);

};

// --- VIP Planner Section Logic ---

goToVipPaymentBtn.onclick = () => 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) {

showModal("Eksik Bilgi", "Lütfen tüm alanları doldurun ve bütçe seçimi yapın.");

return;

}

if (userMembershipLevel !== "Altın") {

showModal("Erişim Reddedildi", "Bu özellik sadece Altın üyelere özeldir. Lütfen üyeliğinizi yükseltin.");

return;

}

vipPlanOutput.style.display = "none";

const prompt = `Lütfen ${destination} şehrinde ${duration} günlük, ${travelers} kişi için, ${selectedBudget} bütçeli, ${travelType} temalı A'dan Z'ye bir tatil planı oluştur. Plana uçuş önerileri (örnek havayolu ve fiyat), otel önerileri (örnek otel adı, fiyat ve konuma yakınlık), havalimanı transfer önerileri ve gün gün detaylı gezilecek yerler/aktiviteler/yemek önerileri ekle. Tüm öneriler bu bütçeye uygun olsun ve örnek linkler (Booking.com, Skyscanner, TripAdvisor, random unsplash image linki gibi) ver. Yanıtı Türkçe ver.`;

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

// Extract and display image if present

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";

speak(`${destination} için VIP tatil planınız hazır.`);

updateTatilPuan(100); // VIP plan için yüksek puan

};

// --- User Info Section Logic ---

registerUserBtn.onclick = async () => {

const newName = await showModal("Kullanıcı Adı Belirle", "Lütfen adınızı girin:");

if (newName && newName.trim() !== "") {

userName = newName.trim();

localStorage.setItem('userName', userName);

displayMembershipInfo();

showModal("Hoş Geldin!", `Hoş geldin, \*\*${userName}\*\*!`);

speak(`Hoş geldin ${userName}!`);

} else {

showModal("Bilgi", "Ad girilmediği için misafir olarak devam ediyorsunuz.");

}

};

// --- Time Travel Section Logic ---

goToTimeTravelPaymentBtn.onclick = () => showSection("payment-section");

startTimeTravelBtn.onclick = async () => {

const era = timeTravelEraInput.value.trim();

const duration = parseInt(timeTravelDurationInput.value);

if (!era || isNaN(duration) || duration < 1) {

showModal("Eksik Bilgi", "Lütfen geçerli bir dönem ve gün sayısı girin.");

return;

}

if (userMembershipLevel !== "Altın") {

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 anlat. Hikaye sürükleyici olsun ve o döneme ait bir görsel linki (örnek unsplash veya pixabay linki gibi) içer. Yanıtı Türkçe ver.`;

const reply = await 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";

speak(`${era} dönemine yolculuk hikayeniz hazır.`);

updateTatilPuan(75); // Zaman yolculuğu için puan

};

// --- Destiny Route Section Logic ---

predictDestinyBtn.onclick = async () => {

const age = destinyAgeInput.value.trim();

const hobby = destinyHobbyInput.value.trim();

const dream = destinyDreamInput.value.trim();

if (!age || !hobby || !dream) {

showModal("Eksik Bilgi", "Lütfen yaşınızı, hobinizi ve hayalinizi girin.");

return;

}

destinyRouteOutput.style.display = "none";

realizeDestinyBtn.style.display = "none"; // Hide button until prediction

const prompt = `Kullanıcıdan şu bilgileri aldım: Yaş: ${age}, Hobi: ${hobby}, Hayal: ${dream}. Bu 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. Yanıtı Türkçe ver ve VIP üyelik için teşvik edici bir mesaj ekle (Örn: "Bu kaderi gerçeğe dönüştürmek Altın Üyelere özel!").`;

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

destinyRouteOutput.innerHTML = `<h4>Kader Rotanızın Kehaneti:</h4><p>${reply}</p>`;

destinyRouteOutput.style.display = "block";

speak("Kader rotanız hazırlandı!");

realizeDestinyBtn.style.display = "block"; // Show "Realize Destiny" button

updateTatilPuan(40); // Kader rotası için puan

};

realizeDestinyBtn.onclick = () => {

if (userMembershipLevel === "Altın") {

showModal("Kader Gerçekleşiyor!", "Harika! Altın üye olarak kader rotanızı gerçeğe dönüştürme zamanı. VIP Tur Planlayıcıya yönlendiriliyorsunuz.");

showSection("vip-planner-section"); // Redirect to VIP planner

} else {

showModal("Erişim Reddedildi", "Kader rotanızı gerçeğe dönüştürmek Altın üyelere özeldir. Lütfen üyeliğinizi yükseltin!");

showSection("payment-section"); // Redirect to payment

}

};

// --- AI Companion Section Logic ---

createCompanionBtn.onclick = async () => {

const companionName = companionNameInput.value.trim();

const personality = companionPersonalitySelect.value;

if (!companionName) {

showModal("Eksik Bilgi", "Lütfen yoldaşınıza bir isim verin.");

return;

}

companionChatArea.style.display = "none";

const prompt = `Lütfen "${companionName}" adında, "${personality}" kişiliğe sahip, seyahat konusunda uzman bir AI yoldaş karakteri yarat. Onu tanıtıcı kısa bir metin yaz ve ilk selamlamasını yap. Yanıtı Türkçe ver.`;

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

aiCompanion = { name: companionName, personality: personality, intro: reply };

activeCompanionName.textContent = companionName;

companionChatBox.innerHTML = `<p><strong>${companionName}:</strong> ${aiCompanion.intro}</p>`;

companionChatArea.style.display = "block";

speak(`${companionName} adında, ${personality} kişiliğe sahip yoldaşınız oluşturuldu.`);

updateTatilPuan(30);

};

sendCompanionMessageBtn.onclick = async () => {

const userMessage = companionInput.value.trim();

if (!userMessage) return;

companionChatBox.innerHTML += `<p><strong>Sen:</strong> ${userMessage}</p>`;

companionInput.value = "";

if (!aiCompanion) {

showModal("Hata", "Önce bir AI Yoldaşı oluşturmalısın!");

return;

}

const prompt = `Benim adım ${userName}. Senin adın ${aiCompanion.name} ve kişiliğin ${aiCompanion.personality}. Kullanıcının mesajı: "${userMessage}". Bu bir sohbet. Tatil ve seyahat konularında sohbet et. Konuşmanın dilini (Türkçe) koru. AI yoldaşı olarak sohbet etmeyi sürdür.`;

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

companionChatBox.innerHTML += `<p><strong>${aiCompanion.name}:</strong> ${reply}</p>`;

companionChatBox.scrollTop = companionChatBox.scrollHeight;

speak(reply);

};

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

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

});

// --- Payment Section Logic (Demo) ---

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) {

showModal("Hata", "Lütfen tüm kart bilgilerini doldurun.");

return;

}

// Simulate payment processing

showModal("Ödeme İşleniyor...", "Ödemeniz simüle ediliyor. Lütfen bekleyin...");

await new Promise(resolve => setTimeout(resolve, 2000)); // Simulate network delay

// Assume payment is successful in demo

userMembershipLevel = "Altın";

localStorage.setItem('userMembershipLevel', userMembershipLevel);

updateTatilPuan(200); // Give a boost to reach Altın level if not already

showModal("Ödeme Başarılı!", "Tebrikler! Artık Altın Üyesiniz! Tüm VIP özelliklere erişiminiz var. Keyfini çıkarın!");

speak("Ödemeniz başarılı. Artık Altın üyesiniz!");

showSection("user-info-section"); // Redirect to user info

};

</script>

</body>

</html>

,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,

<!DOCTYPE html>

<html lang="tr">

<head>

<meta charset="UTF-8" />

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

<title>Palmiye Kaptan AI - Tatil Asistanınız</title>

<link href="https://fonts.googleapis.com/css2?family=Segoe+UI:wght@400;700&display=swap" rel="stylesheet">

<!-- Font Awesome for Icons -->

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

<style>

/\* Genel Stil Ayarları \*/

body {

margin: 0;

font-family: 'Segoe UI', sans-serif;

background: linear-gradient(to bottom, #e0f7fa, #b2dfdb);

/\* Arka plan resmini güncelledik \*/

background-image: url('resim.jpg'); /\* Yüklenen resmi kullan \*/

background-size: cover;

background-attachment: fixed;

background-position: center;

color: #004d40;

display: flex;

min-height: 100vh;

overflow: hidden; /\* No scrolling on body, content scrolls inside \*/

}

/\* Ana Düzen (Flexbox) \*/

.main-layout {

display: flex;

width: 100%;

height: 100vh;

}

/\* Sol Menü \*/

.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;

}

/\* Ana İçerik Alanı - Daha şeffaf yaptık \*/

.content-area {

flex: 1;

display: flex;

flex-direction: column;

background-color: rgba(255, 255, 255, 0.7); /\* Şeffaflığı artırdık \*/

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

border-radius: 15px;

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

position: relative; /\* Avatar için \*/

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

}

/\* Üst Bar \*/

.top-bar {

padding: 15px 25px;

background-color: rgba(0, 121, 107, 0.8); /\* Üst barı da şeffaf yaptık \*/

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;

}

/\* İçerik Bölümleri (Sekmeler gibi) \*/

.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, 0.7); /\* İçerik bölümünü de şeffaf yaptık \*/

border-bottom-left-radius: 15px;

border-bottom-right-radius: 15px;

}

.content-section.active {

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

}

/\* Sohbet Bölümü Özel Stiller \*/

#chat-section {

display: flex;

flex-direction: column;

}

#chat-box {

flex: 1;

background: rgba(224, 242, 241, 0.8); /\* Sohbet kutusunu da şeffaf yaptık \*/

padding: 15px;

border-radius: 10px;

margin-bottom: 15px;

border: 1px solid rgba(200, 230, 201, 0.7);

box-shadow: inset 0 1px 3px rgba(0, 0, 0, 0.1);

overflow-y: auto;

min-height: 250px; /\* Minimum sohbet kutusu yüksekliği \*/

}

.message {

display: flex;

margin-bottom: 12px;

max-width: 85%;

word-wrap: break-word;

line-height: 1.5;

}

.message strong {

margin-right: 8px;

flex-shrink: 0;

font-weight: 700;

}

.message.user {

align-self: flex-end;

background-color: rgba(220, 237, 200, 0.85);

color: #2e7d32;

border-radius: 18px 18px 5px 18px;

padding: 10px 15px;

box-shadow: 0 2px 5px rgba(0, 0, 0, 0.05);

}

.message.bot {

align-self: flex-start;

background-color: rgba(187, 222, 251, 0.85);

color: #1565c0;

border-radius: 18px 18px 18px 5px;

padding: 10px 15px;

box-shadow: 0 2px 5px rgba(0, 0, 0, 0.05);

}

.message img {

max-width: 100%;

height: auto;

border-radius: 8px;

margin-top: 10px;

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

}

.message a {

color: #00796b;

text-decoration: none;

font-weight: bold;

}

.message a:hover {

text-decoration: underline;

}

.input-group {

display: flex;

gap: 10px;

margin-top: 15px;

}

input[type="text"] {

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 {

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);

}

/\* Özellik Bölümleri Ortak Stil - Daha şeffaf yaptık \*/

.feature-section {

padding: 20px;

background-color: rgba(224, 242, 241, 0.8); /\* Daha şeffaf yaptık \*/

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 input[type="text"],

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

.feature-form select,

.feature-form textarea { /\* Textarea eklendi \*/

width: calc(100% - 22px);

padding: 10px;

margin-bottom: 15px;

border: 1px solid #ccc;

border-radius: 5px;

font-size: 1em;

}

.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, 0.8); /\* Plan çıktısını da şeffaf yaptık \*/

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://i.ibb.co/L9513X9/palmiye-kaptan-avatar.png');

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; /\* Genişlettik \*/

padding: 20px 0;

display: flex;

flex-direction: column;

gap: 20px;

z-index: 90;

background-color: rgba(255, 255, 255, 0.85); /\* Hafif şeffaf arka plan \*/

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

border-top-left-radius: 15px;

border-bottom-left-radius: 15px;

align-items: center; /\* İçeriği ortala \*/

}

.ad-area {

width: 90%; /\* Yan panel genişliğine daha uyumlu \*/

height: 180px; /\* Reklam yüksekliği \*/

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; /\* Resimlerin taşmasını engeller \*/

padding: 10px; /\* İç boşluk \*/

box-sizing: border-box; /\* Padding dahil genişlik hesaplama \*/

}

.ad-area:hover {

transform: translateY(-3px);

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

}

.ad-area img {

max-width: 100%; /\* İç alana sığdır \*/

max-height: 70%;

object-fit: cover; /\* Resmin alanı doldurmasını sağlar \*/

margin-bottom: 5px;

border-radius: 5px; /\* Resimlerde de yuvarlak köşeler \*/

}

.ad-area p {

margin: 0;

padding: 0 5px;

font-size: 0.9em;

}

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

.admin-ad-message {

padding: 15px;

background-color: #e0f7fa;

border: 1px dashed #00796b;

border-radius: 8px;

text-align: center;

font-size: 1.1em;

color: #004d40;

}

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

.modal {

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

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

z-index: 1000; /\* 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: 1px solid #888;

border-radius: 10px;

width: 80%; /\* Could be more responsive \*/

max-width: 500px;

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

text-align: center;

}

.modal-content h3 {

color: #004d40;

margin-bottom: 20px;

}

.modal-content button {

background-color: #00796b;

color: white;

padding: 10px 20px;

border: none;

border-radius: 5px;

cursor: pointer;

margin-top: 20px;

}

/\* Loading Indicator \*/

.loading-indicator {

text-align: center;

margin-top: 10px;

color: #00796b;

font-weight: bold;

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

}

/\* 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; /\* Sloganı ortaya al \*/

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; /\* Mobil'de reklamları gizle \*/

}

.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 {

width: calc(100% - 20px);

}

}

</style>

</head>

<body>

<!-- Palmiye Kaptan Avatarı -->

<div id="avatar" title="Palmiye Kaptan Avatarı"></div>

<!-- Genel Modal (Alert/Confirm yerine) -->

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

<div class="modal-content">

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

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

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

</div>

</div>

<!-- Ana Düzen -->

<div class="main-layout">

<!-- Sol Menü -->

<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> <!-- Yeni bölüm -->

<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>

</ul>

</nav>

</div>

<!-- Ana İçerik Alanı -->

<div class="content-area">

<!-- Üst Bar -->

<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> <!-- Kullanıcı ID'si eklendi -->

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

<!-- Dil seçeneği (şimdilik demo, AI dili algılayacak) -->

<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>

<!-- Diğer diller eklenebilir -->

</select>

</div>

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

</div>

<!-- İçerik Bölümleri -->

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

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

<div id="chat-box" 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. \*\*Gelecekte 3D sanal turlar ile bu deneyim daha da gelişecek!\*\*</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" />

<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;"></div>

</div>

</div>

<!-- Yeni AI Fotoğraf Stüdyosu Bölümü -->

<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 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>

<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 Fotoğrafınız:</h4>

<img id="generated-image" src="" alt="Oluşturulan AI Fotoğrafı" style="max-width: 100%; height: auto; border-radius: 8px; margin-top: 15px; display: none;">

<button id="download-image-btn" style="margin-top: 15px; display: none;">Fotoğrafı İndir (50 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>

</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>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">

<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" />

<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" />

<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>

</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" style="background-color: #f0f8ff; padding: 10px; border-radius: 8px; max-height: 200px; overflow-y: auto; margin-bottom: 10px;"></div>

<input type="text" id="companion-input" placeholder="Yoldaşınıza bir şeyler sorun..." style="width: calc(100% - 20px); padding: 10px; border-radius: 5px;">

<button id="send-companion-message-btn" style="margin-top: 10px;">Mesaj Gönder</button>

</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>

</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> <!-- /content-area -->

<!-- Sağ Reklam Panelleri (Yönetici İlan Alanı) -->

<div class="right-sidebar">

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

<h4>Yönetici İlan Alanı</h4>

<p>İlanlar buraya eklenecektir. İlan vermek için lütfen bizimle <a href="mailto:info@palmiyekaptan.com" style="color:#004d40; font-weight:bold;">e-posta</a> aracılığıyla iletişime geçiniz.</p>

</div>

<!-- İlanlar buraya dinamik olarak Firebase'den yüklenecek -->

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

<!-- Örnek ilanlar (bunlar Firestore'dan çekilecek) -->

</div>

</div>

</div> <!-- /main-layout -->

<script type="module">

// Firebase global değişkenleri (Canvas tarafından sağlanır)

const OPENROUTER\_API\_KEY = "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";

const IMAGE\_DOWNLOAD\_COST = 50; // PalmCoin maliyeti

const IMAGE\_GENERATION\_MODEL = "imagen-3.0-generate-002"; // Image generation model

const appId = typeof \_\_app\_id !== 'undefined' ? \_\_app\_id : 'default-app-id';

const firebaseConfig = typeof \_\_firebase\_config !== 'undefined' ? JSON.parse(\_\_firebase\_config) : {};

const initialAuthToken = typeof \_\_initial\_auth\_token !== 'undefined' ? \_\_initial\_auth\_token : null;

let app;

let db;

let auth;

let currentUserId = null;

let isAuthReady = false;

// --- Uygulama Durumu Değişkenleri (Firestore'dan çekilecek varsayılan değerler) ---

let voiceEnabled = true;

let tatilPuan = 0;

let userMembershipLevel = "Bronz";

let userName = "Misafir";

let gameActive = false;

let currentQuestionIndex = 0;

let gameScore = 0;

let generatedImageUrl = ''; // Son oluşturulan fotoğrafın URL'sini tutmak için

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!"

];

const gameQuestions = [

{

question: "Dünyanın en yüksek dağı hangisidir?",

options: ["(A) K2", "(B) Kilimanjaro", "(C) Everest"],

answer: "C",

points: 20

},

{

question: "Eyfel Kulesi hangi şehirdedir?",

options: ["(A) Londra", "(B) Paris", "(C) Roma"],

answer: "B",

points: 25

},

{

question: "Büyük Kanyon hangi ülkededir?",

options: ["(A) Kanada", "(B) ABD", "(C) Avustralya"],

answer: "B",

points: 30

}

];

// Global olarak erişilebilmesi için Firebase'den gelen referansları atayalım

window.getFirestore = () => db;

window.getAuth = () => auth;

window.getCurrentUserId = () => currentUserId;

window.isAuthReady = () => isAuthReady;

window.initializeApp = (config) => initializeApp(config);

window.signInAnonymously = (authInstance) => signInAnonymously(authInstance);

window.signInWithCustomToken = (authInstance, token) => signInWithCustomToken(authInstance, token);

window.onAuthStateChanged = (authInstance, callback) => onAuthStateChanged(authInstance, callback);

// Firestore importlarını doğrudan window'a ata

window.doc = doc;

window.getDoc = getDoc;

window.setDoc = setDoc;

window.onSnapshot = onSnapshot;

window.collection = collection;

window.query = query;

window.addDoc = addDoc;

window.getDocs = getDocs;

window.appId = appId;

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

// Bu fonksiyonlar artık window.doc, window.collection vb. tanımlandıktan sonra erişebilir.

window.getUserProfileRef = function() {

const dbInstance = window.getFirestore();

const userId = window.getCurrentUserId();

const currentAppId = window.appId;

if (!dbInstance || !userId || !currentAppId) {

console.error("Firestore, User ID or App ID is not ready for user profile.");

return null;

}

return window.doc(dbInstance, `artifacts/${currentAppId}/users/${userId}/profile`, "user-data");

}

window.loadUserProfile = async function() {

const profileRef = window.getUserProfileRef();

if (!profileRef) return;

window.onSnapshot(profileRef, (docSnap) => {

if (docSnap.exists()) {

const data = docSnap.data();

userName = data.userName || "Misafir";

tatilPuan = data.tatilPuan || 0;

userMembershipLevel = data.userMembershipLevel || "Bronz";

gameScore = data.gameScore || 0;

console.log("User profile loaded from Firestore:", data);

} else {

console.log("No user profile found, setting default.");

window.updateUserProfile({

userName: "Misafir",

tatilPuan: 0,

userMembershipLevel: "Bronz",

gameScore: 0

});

}

window.displayMembershipInfo();

window.updateTatilPuanDisplay();

document.getElementById("user-id-display").textContent = `UID: ${window.getCurrentUserId()}`;

}, (error) => {

console.error("Error fetching user profile:", error);

window.showModal("Hata", "Kullanıcı verileri yüklenirken bir sorun oluştu.");

});

}

window.updateUserProfile = async function(dataToUpdate) {

const profileRef = window.getUserProfileRef();

if (!profileRef) return;

try {

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

console.log("User profile updated:", dataToUpdate);

} catch (error) {

console.error("Error updating user profile:", error);

window.showModal("Hata", "Kullanıcı verileri kaydedilirken bir sorun oluştu.");

}

}

// --- Firebase İlan Yönetimi Fonksiyonları ---

window.getAdsCollectionRef = function() {

const dbInstance = window.getFirestore();

const currentAppId = window.appId;

if (!dbInstance || !currentAppId) {

console.error("Firestore or App ID is not ready for ads.");

return null;

}

return window.collection(dbInstance, `artifacts/${currentAppId}/public/data/ads`);

}

window.loadAds = async function() {

const adsCollectionRef = window.getAdsCollectionRef();

if (!adsCollectionRef) return;

window.onSnapshot(adsCollectionRef, (querySnapshot) => {

const container = document.getElementById("dynamic-ads-container");

if (container) {

container.innerHTML = '';

if (querySnapshot.empty) {

console.log("No ads found in Firestore.");

} else {

querySnapshot.forEach((doc) => {

const ad = doc.data();

const adElement = document.createElement("a");

adElement.href = ad.link || "#";

adElement.target = "\_blank";

adElement.classList.add("ad-area");

adElement.innerHTML = `

<img src="${ad.imageUrl || 'https://placehold.co/100x70/004d40/ffffff?text=Ad'}" alt="${ad.title || 'İlan'}">

<p>${ad.title || 'Yeni İlan'}</p>

`;

container.appendChild(adElement);

});

console.log("Ads loaded from Firestore.");

}

}

}, (error) => {

console.error("Error fetching ads:", error);

});

}

window.addAd = async function(title, imageUrl, link) {

const adsCollectionRef = window.getAdsCollectionRef();

if (!adsCollectionRef) return;

try {

await window.addDoc(adsCollectionRef, {

title: title,

imageUrl: imageUrl,

link: link,

createdAt: new Date()

});

console.log("Ad added successfully!");

window.showModal("Başarılı", "İlan başarıyla eklendi (demo).");

} catch (error) {

console.error("Error adding ad:", error);

window.showModal("Hata", "İlan eklenirken bir sorun oluştu.");

}

}

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 elements are not found. Cannot display modal.");

return Promise.resolve(false);

}

return new Promise((resolve) => {

modalTitleEl.textContent = title;

modalMessageEl.innerHTML = message;

modal.style.display = "flex";

const handleConfirm = () => {

modal.style.display = "none";

modalConfirmBtnEl.removeEventListener("click", handleConfirm);

resolve(true);

};

modalConfirmBtnEl.addEventListener("click", handleConfirm);

});

}

window.displayMessage = function(sender, message) {

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

if (!chatBoxEl) {

console.error("Chat box element not found.");

return;

}

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

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

messageDiv.innerHTML = `<strong>${sender === "user" ? userName : "Palmiye Kaptan"}:</strong> ${message}`;

chatBoxEl.appendChild(messageDiv);

chatBoxEl.scrollTop = chatBoxEl.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) {

tatilPuan += points;

const oldLevel = userMembershipLevel;

if (tatilPuan < 100) {

userMembershipLevel = "Bronz";

} else if (tatilPuan < 200) {

userMembershipLevel = "Gümüş";

} else {

userMembershipLevel = "Altın";

}

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

if (oldLevel !== userMembershipLevel) {

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

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

}

}

window.updateTatilPuanDisplay = function() {

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

if (tatilpuanTopEl) {

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

}

}

window.displayMembershipInfo = function() {

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

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

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

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

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

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

if (displayUserIdEl) displayUserIdEl.textContent = window.getCurrentUserId() || "Yükleniyor...";

if (displayUsernameEl) displayUsernameEl.textContent = userName;

if (displayTatilpuanEl) displayTatilpuanEl.textContent = tatilPuan;

if (displayMembershipLevelEl) displayMembershipLevelEl.textContent = userMembershipLevel;

if (displayGameScoreEl) displayGameScoreEl.textContent = gameScore;

if (userIdDisplayEl) userIdDisplayEl.textContent = `UID: ${window.getCurrentUserId() || 'Misafir'}`;

}

window.showSection = function(sectionId) {

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

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

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");

}

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

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

} 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") { // Yeni AI Fotoğraf Stüdyosu için kontrol

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

// Her yeni açıldığında geçmiş fotoğrafı ve indirme butonunu gizle

const generatedImageEl = document.getElementById("generated-image");

const downloadImageBtnEl = document.getElementById("download-image-btn");

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

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

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

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

generatedImageUrl = ''; // Önceki URL'yi sıfırla

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

window.displayMembershipInfo();

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

document.getElementById("card-number").value = '';

document.getElementById("expiry-date").value = '';

document.getElementById("cvv").value = '';

document.getElementById("card-holder-name").value = '';

}

}

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";

}

}

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

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

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: [{ role: "user", content: prompt }]

})

});

if (!response.ok) {

const errorData = await response.json();

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

}

const data = await response.json();

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

} catch (error) {

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

window.showModal("Hata!", `Bir AI hatası oluştu: ${error.message}. Lütfen daha sonra tekrar deneyin.`);

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

} finally {

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

}

}

// --- Image Generation API Call ---

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

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

const apiUrl = `https://generativelanguage.googleapis.com/v1beta/models/${IMAGE\_GENERATION\_MODEL}:predict?key=`; // API key Canvas tarafından sağlanacak

try {

const payload = {

instances: { prompt: promptText },

parameters: { "sampleCount": 1 }

};

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

}

const result = await response.json();

if (result.predictions && result.predictions.length > 0 && result.predictions[0].bytesBase64Encoded) {

return `data:image/png;base64,${result.predictions[0].bytesBase64Encoded}`;

} else {

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

}

} catch (error) {

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

window.showModal("Hata!", `Görsel oluşturulurken bir sorun oluştu: ${error.message}.`);

return null;

} finally {

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

}

}

// --- Event Listeners ---

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

// HTML Elementlerini burada tanımla, çünkü DOMContentLoaded eventi içinde erişilebilirler.

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");

// Game Section elements

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");

// Virtual Holiday Section elements

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

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

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

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

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

// AI Photo Studio Section elements

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

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

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

const generatedImage = document.getElementById("generated-image");

const downloadImageBtn = document.getElementById("download-image-btn");

// VIP Planner Section elements

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 = "";

// User Info Section elements

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

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

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");

// Time Travel Section elements

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

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

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

// Destiny Route Section elements

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

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

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

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 Companion Section elements

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");

let aiCompanion = null;

// Payment Section elements

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");

// Modal elements

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

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

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

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

// Admin Ad Elements

const dynamicAdsContainer = document.getElementById("dynamic-ads-container");

// --- 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)];

}

if (window.isAuthReady()) {

window.loadUserProfile();

window.loadAds();

} else {

console.warn("Authentication not ready, cannot load user profile or ads yet.");

window.displayMembershipInfo();

window.updateTatilPuanDisplay();

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

if (userIdDisplayEl) userIdDisplayEl.textContent = `UID: ${window.getCurrentUserId() || 'Misafir'}`;

}

window.displayMessage("bot", "Merhaba! Ben Palmiye Kaptan, tatil asistanınım. Sol menüden bir özellik seçebilirsin.");

window.showSection("chat-section");

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

if (chatSectionButton) {

chatSectionButton.classList.add("active");

}

};

// Firebase başlatma ve durum dinleyicileri

if (Object.keys(firebaseConfig).length > 0) {

app = initializeApp(firebaseConfig);

db = getFirestore(app);

auth = getAuth(app);

onAuthStateChanged(auth, async (user) => {

if (user) {

currentUserId = user.uid;

console.log("Firebase Auth State Changed: User is signed in with UID:", currentUserId);

} else {

console.log("Firebase Auth State Changed: No user is signed in.");

if (!initialAuthToken) {

try {

await signInAnonymously(auth);

currentUserId = auth.currentUser.uid;

console.log("Signed in anonymously with UID:", currentUserId);

} catch (error) {

console.error("Anonymous sign-in failed:", error);

window.showModal("Hata", "Anonim giriş yapılamadı. Bazı özellikler kısıtlı olabilir.");

}

}

}

isAuthReady = true;

// Auth hazır olduğunda, kullanıcı verilerini ve ilanları yükle

window.loadUserProfile();

window.loadAds();

window.displayMembershipInfo(); // Auth sonrası kullanıcı ID'sini güncelle

});

if (initialAuthToken) {

try {

await signInWithCustomToken(auth, initialAuthToken);

currentUserId = auth.currentUser.uid;

console.log("Signed in with custom token:", currentUserId);

} catch (error) {

console.error("Custom token sign-in failed:", error);

}

}

} else {

console.warn("Firebase config not found. Running without Firebase.");

isAuthReady = true;

}

// Uygulama özelliklerini DOM hazır olduğunda başlat

window.initializeAppFeatures();

// --- Event Listeners ---

sidebarButtons.forEach(button => {

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

const sectionId = button.dataset.section;

window.showSection(sectionId);

});

});

sendChatBtn.onclick = async () => {

const userMessage = chatInput.value.trim();

if (!userMessage) return;

window.displayMessage("user", userMessage);

chatInput.value = "";

const prompt = `Kullanıcının mesajı: "${userMessage}". Bu bir tatil asistanı sohbeti. Mesajın dilini (Türkçe) 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.`;

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

window.displayMessage("bot", reply);

window.speak(reply);

window.updateTatilPuan(5);

};

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

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

});

voiceToggleTop.onclick = () => {

voiceEnabled = !voiceEnabled;

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

window.speak(voiceEnabled ? "Sesli yanıtlar açıldı." : "Sesli yanıtlar kapatıldı.");

};

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");

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 < gameQuestions.length) {

const q = gameQuestions[currentQuestionIndex];

const questionText = `Soru ${currentQuestionIndex + 1}: ${q.question} ${q.options.join(" ")}`;

gameOutput.innerHTML += `<p><strong>Palmiye Kaptan:</strong> ${questionText}</p>`;

gameAnswerInput.value = "";

window.speak(questionText);

} else {

endGame();

}

}

async function handleGameAnswer(answer) {

const currentQ = gameQuestions[currentQuestionIndex];

if (!currentQ) return;

const userAnswer = answer.trim().toUpperCase();

if (userAnswer === currentQ.answer) {

gameOutput.innerHTML += `<p style="color: green;"><strong>Palmiye Kaptan:</strong> Tebrikler! Doğru cevap.</p>`;

window.speak("Tebrikler! Doğru cevap.");

gameScore += currentQ.points;

await window.updateUserProfile({ gameScore: gameScore });

window.updateTatilPuan(currentQ.points);

} else {

gameOutput.innerHTML += `<p style="color: red;"><strong>Palmiye Kaptan:</strong> Yanlış cevap. Doğru cevap: ${currentQ.answer}</p>`;

window.speak(`Yanlış cevap. Doğru cevap ${currentQ.answer}`);

}

currentQuestionIndex++;

if (currentQuestionIndex < gameQuestions.length) {

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();

}

startVirtualBtn.onclick = async () => {

const city = virtualCityInput.value.trim();

const days = parseInt(virtualDaysInput.value);

if (!city || isNaN(days) || days < 1) {

window.showModal("Eksik Bilgi", "Lütfen geçerli bir şehir/yer ve gün sayısı girin.");

return;

}

virtualHolidayOutput.style.display = "none";

const prompt = `Lütfen ${city} şehrinde ${days} günlük sanal bir tatil hikayesi yaz. Bu tatilde gezilecek yerleri, tadılacak lezzetleri ve yaşanacak deneyimleri detaylıca anlat. Hikaye ilgi çekici, sürükleyici ve yaratıcı olsun. Ayrıca, bu tatili temsil edecek bir resim linki öner (örnek URL: 'https://source.unsplash.com/random/800x600/?${city}-tourism' gibi). Yanıtı Türkçe ver.`;

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

let holidayStory = reply;

const urlRegex = /(https?:\/\/[^\s]+\.(?:png|jpe?g|gif|webp|unsplash\.com\/\S+|pixabay\.com\/\S+))/i;

const match = holidayStory.match(urlRegex);

let mediaSuggestionHtml = "";

if (match) {

mediaSuggestionHtml = `<br><br><img src="${match[0]}" alt="${city} Tatili">`;

holidayStory = holidayStory.replace(match[0], '').trim();

} else {

const genericImageUrl = `https://source.unsplash.com/random/800x600/?${city.replace(/\s/g, '-')}-travel`;

mediaSuggestionHtml = `<br><br><img src="${genericImageUrl}" alt="${city} Tatili">`;

}

virtualHolidayOutput.innerHTML = `<h4>${city} - ${days} Günlük Sanal Tatil Hikayen:</h4><p>${holidayStory}</p>${mediaSuggestionHtml}`;

virtualHolidayOutput.style.display = "block";

window.speak(`İşte ${city} için sanal tatil hikayen.`);

window.updateTatilPuan(50);

};

// --- AI Photo Studio Logic ---

goToAiPhotoPaymentBtn.onclick = () => window.showSection("payment-section");

generateAiPhotoButton.onclick = async () => {

const promptText = aiPhotoPromptInput.value.trim();

if (!promptText) {

window.showModal("Eksik Bilgi", "Lütfen oluşturmak istediğiniz fotoğrafı tanımlayın.");

return;

}

if (userMembershipLevel !== "Altın") {

window.showModal("Erişim Reddedildi", "Bu özellik sadece Altın üyelere özeldir. Lütfen üyeliğinizi yükseltin.");

return;

}

aiPhotoOutput.style.display = 'none';

generatedImage.style.display = 'none';

downloadImageBtn.style.display = 'none';

generatedImageUrl = await window.callImageGenerationAI(promptText, aiPhotoLoading);

if (generatedImageUrl) {

generatedImage.src = generatedImageUrl;

generatedImage.alt = promptText;

generatedImage.style.display = 'block';

aiPhotoOutput.style.display = 'block';

downloadImageBtn.style.display = 'block';

downloadImageBtn.textContent = `Fotoğrafı İndir (${IMAGE\_DOWNLOAD\_COST} PalmCoin)`;

window.speak("Fotoğrafınız başarıyla oluşturuldu!");

}

};

downloadImageBtn.onclick = async () => {

if (!generatedImageUrl) {

window.showModal("Hata", "Önce bir fotoğraf oluşturmalısınız.");

return;

}

if (tatilPuan < IMAGE\_DOWNLOAD\_COST) {

window.showModal("Yetersiz PalmCoin", `Bu fotoğrafı indirmek için ${IMAGE\_DOWNLOAD\_COST} PalmCoin'e ihtiyacınız var. Mevcut PalmCoin: ${tatilPuan}. Daha fazla PalmCoin kazanmak için oyun oynayabilir veya VIP üyeliğinizi kontrol edebilirsiniz.`);

return;

}

// PalmCoin'i düş

tatilPuan -= IMAGE\_DOWNLOAD\_COST;

await window.updateUserProfile({ tatilPuan: tatilPuan });

window.showModal("İndiriliyor!", "Fotoğrafınız indiriliyor. PalmCoin bakiyeniz güncellendi.");

window.speak("Fotoğrafınız indiriliyor.");

// Fotoğrafı indir

const link = document.createElement('a');

link.href = generatedImageUrl;

link.download = `palmiye-kaptan-ai-foto-${Date.now()}.png`; // Benzersiz dosya adı

document.body.appendChild(link);

link.click();

document.body.removeChild(link);

generatedImageUrl = ''; // İndirildikten sonra URL'yi sıfırla

generatedImage.style.display = 'none';

downloadImageBtn.style.display = 'none';

aiPhotoOutput.style.display = 'none';

};

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";

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);

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";

window.speak(`${destination} için VIP tatil planınız hazır.`);

window.updateTatilPuan(100);

};

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}!`);

} else {

window.showModal("Bilgi", "Ad girilmediği için mevcut adınız değişmedi.");

}

};

goToTimeTravelPaymentBtn.onclick = () => window.showSection("payment-section");

startTimeTravelBtn.onclick = async () => {

const era = timeTravelEraInput.value.trim();

const duration = parseInt(timeTravelDurationInput.value);

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. 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);

};

predictDestinyBtn.onclick = async () => {

const age = destinyAgeInput.value.trim();

const hobby = destinyHobbyInput.value.trim();

const destinyDream = destinyDreamInput.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 bilgileri aldım: Yaş: ${age}, Hobi: ${hobby}, Hayal: ${destinyDream}. Bu 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. 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.`;

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

destinyRouteOutput.innerHTML = `<h4>Kader Rotanızın Kehaneti:</h4><p>${reply}</p>`;

destinyRouteOutput.style.display = "block";

window.speak("Kader rotanız hazırlandı!");

realizeDestinyBtn.style.display = "block";

window.updateTatilPuan(40);

};

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");

}

};

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";

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;

companionChatBox.innerHTML = `<p><strong>${companionName}:</strong> ${aiCompanion.intro}</p>`;

companionChatArea.style.display = "block";

window.speak(`${companionName} adında, ${personality} kişiliğe sahip yoldaşınız oluşturuldu.`);

window.updateTatilPuan(30);

};

sendCompanionMessageBtn.onclick = async () => {

const userMessage = companionInput.value.trim();

if (!userMessage) return;

companionChatBox.innerHTML += `<p><strong>Sen:</strong> ${userMessage}</p>`;

companionInput.value = "";

if (!aiCompanion) {

window.showModal("Hata", "Önce bir AI Yoldaşı oluşturmalısın!");

return;

}

const prompt = `Benim adım ${userName}. Senin adın ${aiCompanion.name} ve kişiliğin ${aiCompanion.personality}. Kullanıcının mesajı: "${userMessage}". Bu bir sohbet. Tatil ve seyahat konularında sohbet et. Konuşmanın dilini (Türkçe) koru. AI yoldaşı olarak sohbet etmeyi sürdür. Kişiliğine uygun, yaratıcı ve sohbeti devam ettiren bir yanıt ver.`;

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

companionChatBox.innerHTML += `<p><strong>${aiCompanion.name}:</strong> ${reply}</p>`;

companionChatBox.scrollTop = companionChatBox.scrollHeight;

window.speak(reply);

};

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

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

});

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);

window.showModal("Ödeme Başarılı!", "Tebrikler! Artık Altın Üyesiniz! Tüm VIP özelliklere erişiminiz var. Keyfini çıkarın!");

window.speak("Ödemeniz başarılı. Artık Altın üyesiniz!");

window.showSection("user-info-section");

};

});

</script>

</body>

</html>