// Firebase Configuration

const firebaseConfig = {

apiKey: "AIzaSyAI5Sn4rTmQ\_un5baTyfduewlqaAiOibiQ",

authDomain: "life-archive-1ed4a.firebaseapp.com",

projectId: "life-archive-1ed4a",

storageBucket: "life-archive-1ed4a.firebasestorage.app",

messagingSenderId: "999843492181",

appId: "1:999843492181:web:0defdcd915cd6dd4273fdc"

};

// Initialize Firebase

firebase.initializeApp(firebaseConfig);

const authService = firebase.auth();

const firestore = firebase.firestore();

const storageService = firebase.storage();

let mediaRecorder = null;

let audioChunks = [];

let audioBlob = null;

let recordingTimerInterval = null;

let recordingStartTime = 0;

let currentLang = 'en'; // Default language

// Function to display toast notifications

function showToast(message, type = 'info') {

const toastContainer = document.querySelector('.toast-container');

const toast = document.createElement('div');

toast.classList.add('toast', 'align-items-center', 'text-white', 'border-0');

if (type === 'success') {

toast.classList.add('bg-success');

} else if (type === 'error') {

toast.classList.add('bg-danger');

} else {

toast.classList.add('bg-info');

}

toast.setAttribute('role', 'alert');

toast.setAttribute('aria-live', 'assertive');

toast.setAttribute('aria-atomic', 'true');

toast.innerHTML = `

<div class="d-flex">

<div class="toast-body">${message}</div>

<button type="button" class="btn-close btn-close-white me-2 m-auto" data-bs-dismiss="toast" aria-label="Close"></button>

</div>

`;

toastContainer.appendChild(toast);

const bootstrapToast = new bootstrap.Toast(toast, { delay: 3000 });

bootstrapToast.show();

toast.addEventListener('hidden.bs.toast', () => {

toast.remove();

});

}

const translations = {

en: {

home\_title: "Capture Life’s Joyful Moments with Legacy AI",

home\_lead: "Relive the past, celebrate the present, and shape the future with AI-curated happiness. Create Together, Reminisce Forever: Build vibrant digital scrapbooks and happy timelines for generations.",

get\_started\_free: "Get Started Free",

download\_joyful\_guide: "Download Joyful Memory Guide",

how\_it\_works\_title: "How Legacy AI Spreads Joy",

capture\_moments\_step: "1. Capture Happy Moments",

capture\_moments\_desc: "Upload photos, videos, and voice notes of birthdays, trips, and everyday joys.",

create\_adventures\_step: "2. Create Family Adventures",

create\_adventures\_desc: "AI crafts interactive stories and games starring your loved ones.",

build\_timeline\_step: "3. Build a Happy Timeline",

build\_timeline\_desc: "Organize milestones into a vibrant digital scrapbook for all to cherish.",

moments\_title: "Moments That Matter",

moments\_intro: "Relive the joy of birthdays, family trips, inside jokes, and everyday moments. Upload your happiest memories to create a digital scrapbook that sparks smiles.",

upload\_photos\_title: "Photos of Joy",

upload\_photos\_desc: "Share snapshots of laughter and love.",

record\_voice\_title: "Voice Notes",

record\_voice\_desc: "Capture jokes, songs, or heartfelt messages.",

share\_videos\_title: "Happy Videos",

share\_videos\_desc: "Upload clips of celebrations and adventures.",

emotion\_tag\_placeholder: "Tag emotions or themes (e.g., #joyful, #adventure)",

upload\_now: "Upload Now",

record\_now: "Record Now",

stop\_recording: "Stop Recording",

upload\_failed: "Upload failed:",

upload\_success: "Voice note uploaded successfully! MP3 conversion in progress.",

mp3\_conversion\_failed: "Voice note uploaded, but MP3 conversion failed. Playing WebM.",

microphone\_error: "Microphone access denied or unavailable: ",

challenges\_title: "Interactive Family Challenges",

challenges\_intro: "Bring your family closer with fun prompts that create joyful memories. Complete challenges and build a positive archive together!",

funny\_question\_title: "Funny Question Challenge",

funny\_question\_desc: "Ask your grandparents one hilarious question today and record their answer!",

dance\_off\_title: "Family Dance-Off",

dance\_off\_desc: "Record a 60-second family dance-off and share the fun!",

childhood\_memory\_title: "Childhood Memory Story",

childhood\_memory\_desc: "Share a 60-second story of your happiest childhood moment.",

start\_challenge: "Submit Response",

pricing\_title: "Choose Your Joyful Plan",

pricing\_intro: "Start free or unlock premium features to create unforgettable family memories.",

free\_tier\_title: "Basic Legacy",

free\_tier\_desc: "Limited text & photo archive, AI memory highlights, 5 AI interactions/month.",

essential\_tier\_title: "Essential Legacy",

essential\_tier\_desc: "Expanded storage, custom AI persona, multi-language support, mood-based search.",

family\_tier\_title: "Family Legacy Plus",

family\_tier\_desc: "Shared archives, collaborative scrapbooks, unlimited AI interactions, family games.",

lifetime\_tier\_title: "Lifetime Legacy Vault",

lifetime\_tier\_desc: "Unlimited storage, AI legacy book, priority support, scheduled messages, and \*\*exclusive VIP family support with personalized AI memory curations.\*\*",

buy\_now: "Buy Now",

one\_time\_title: "Premium Enhancements",

one\_time\_desc: "Add unique features to your joyful legacy.",

storybook\_purchase: "AI Storybook ($99)",

time\_capsule\_purchase: "Time Capsule ($49)",

video\_message\_purchase: "Legacy Video ($79)",

business\_creator\_title: "For Businesses & Creators",

business\_creator\_desc: "Preserve brand stories or create unique AI personas.",

business\_plan: "Business Plan: $499/year",

creator\_plan: "Creator Plan: $99 setup + $25/mo",

community\_plan: "Community Vault: $149/year",

contact\_sales: "Contact Sales",

login\_title: "Joyful Portal Login",

email\_label: "Email",

email\_input\_placeholder: "Email",

password\_label: "Password",

password\_input\_placeholder: "Password",

login\_button: "Login",

forgot\_password: "Forgot Password?",

email\_required: "Please enter your email address.",

password\_reset\_sent: "Password reset email sent! Check your inbox.",

login\_required: "Please log in to record a...",

current\_plan: "You're on {{planName}} 💛",

welcome\_user: "Welcome, !", // Added for functionality, placeholder for user name

happy\_timeline\_title: "Your Happy Timeline",

happy\_timeline\_desc: "Add milestones, firsts, and joyful moments to your digital scrapbook.",

add\_milestone: "Add Milestone",

bucket\_list\_title: "Family Bucket List",

bucket\_list\_desc: "Set shared goals like trips or fun activities and track progress!",

add\_goal\_placeholder: "Add a goal...",

add\_goal: "Add Goal",

storytelling\_mode\_title: "AI Storytelling Adventure",

storytelling\_mode\_desc: "Your co-pilot in crafting your family’s unique journey, creating fun, fictional stories with your loved ones as heroes!",

story\_characters\_label: "Who are the main characters?",

story\_characters\_placeholder: "e.g., Mom, Dad, Lily, our dog Sparky",

story\_theme\_label: "Choose a theme:",

select\_theme\_option: "Select a theme",

generate\_story: "Generate Story",

challenge\_response\_title: "Respond to Challenges",

challenge\_response\_desc: "View and respond to family challenges.",

view\_challenges: "View Challenges",

account\_settings\_title: "Account Settings",

account\_settings\_desc: "Manage your profile, subscription, and preferences.",

manage\_settings: "Manage Settings",

contact\_support\_title: "Support",

contact\_support\_desc: "Get help or provide feedback.",

contact\_us: "Contact Us",

logout\_button: "Logout",

demo\_timeline\_title: "Preview a Happy Timeline",

demo\_timeline\_desc: "See how Legacy AI brings memories to life!",

login\_to\_create: "Login to Create Your Own",

signup\_title: "Sign Up for Joyful Portal",

signup\_button: "Sign Up",

signup\_link: "Sign Up",

login\_link: "Already have an account? Login"

},

es: {

home\_title: "Captura los Momentos Alegres de la Vida con Legacy AI",

home\_lead: "Revive el pasado, celebra el presente y moldea el futuro con felicidad curada por IA. Crea Juntos, Recuerda Para Siempre: Construye álbumes digitales vibrantes y líneas de tiempo felices para generaciones.",

get\_started\_free: "Comienza Gratis",

download\_joyful\_guide: "Descargar Guía de Memoria Alegre",

how\_it\_works\_title: "Cómo Legacy AI Difunde la Alegría",

capture\_moments\_step: "1. Captura Momentos Felices",

capture\_moments\_desc: "Sube fotos, videos y notas de voz de cumpleaños, viajes y alegrías cotidianas.",

create\_adventures\_step: "2. Crea Aventuras Familiares",

create\_adventures\_desc: "IA crea historias interactivas y juegos protagonizados por tus seres queridos.",

build\_timeline\_step: "3. Construye una Línea de Tiempo Feliz",

build\_timeline\_desc: "Organiza hitos en un álbum digital vibrante para que todos lo atesoren.",

moments\_title: "Momentos Que Importan",

moments\_intro: "Revive la alegría de cumpleaños, viajes familiares, chistes internos y momentos cotidianos. Sube tus recuerdos más felices para crear un álbum digital que provoque sonrisas.",

upload\_photos\_title: "Fotos de Alegría",

upload\_photos\_desc: "Comparte instantáneas de risas y amor.",

record\_voice\_title: "Notas de Voz",

record\_voice\_desc: "Captura chistes, canciones o mensajes sinceros.",

share\_videos\_title: "Videos Felices",

share\_videos\_desc: "Sube clips de celebraciones y aventuras.",

emotion\_tag\_placeholder: "Etiqueta emociones o temas (e.g., #alegre, #aventura)",

upload\_now: "Subir Ahora",

record\_now: "Grabar Ahora",

stop\_recording: "Detener Grabación",

upload\_failed: "Subida fallida:",

upload\_success: "Nota de voz subida con éxito! Conversión a MP3 en progreso.",

mp3\_conversion\_failed: "Nota de voz subida, pero conversión a MP3 fallida. Reproduciendo WebM.",

microphone\_error: "Acceso al micrófono denegado o no disponible: ",

challenges\_title: "Desafíos Familiares Interactivos",

challenges\_intro: "Acércate a tu familia con prompts divertidos que crean recuerdos alegres. Completa desafíos y construye un archivo positivo juntos!",

funny\_question\_title: "Desafío de Pregunta Divertida",

funny\_question\_desc: "Pregúntale a tus abuelos una pregunta hilarante hoy y graba su respuesta!",

dance\_off\_title: "Baile Familiar",

dance\_off\_desc: "Graba un baile familiar de 60 segundos y comparte la diversión!",

childhood\_memory\_title: "Historia de Memoria Infantil",

childhood\_memory\_desc: "Comparte una historia de 60 segundos de tu momento infantil más feliz.",

start\_challenge: "Enviar Respuesta",

pricing\_title: "Elige Tu Plan Alegre",

pricing\_intro: "Comienza gratis o desbloquea funciones premium para crear recuerdos familiares inolvidables.",

free\_tier\_title: "Legacy Básico",

free\_tier\_desc: "Archivo limitado de texto y fotos, destacados de memoria IA, 5 interacciones IA/mes.",

essential\_tier\_title: "Legacy Esencial",

essential\_tier\_desc: "Almacenamiento expandido, persona IA personalizada, soporte multilingüe, búsqueda basada en estado de ánimo.",

family\_tier\_title: "Legacy Familiar Plus",

family\_tier\_desc: "Archivos compartidos, álbumes colaborativos, interacciones IA ilimitadas, juegos familiares.",

lifetime\_tier\_title: "Bóveda de Legacy de Por Vida",

lifetime\_tier\_desc: "Almacenamiento ilimitado, libro de legacy IA, soporte prioritario, mensajes programados, y \*\*soporte VIP familiar exclusivo con curaciones de memoria IA personalizadas.\*\*",

buy\_now: "Comprar Ahora",

one\_time\_title: "Mejoras Premium",

one\_time\_desc: "Agrega características únicas a tu legacy alegre.",

storybook\_purchase: "Libro de Cuentos IA ($99)",

time\_capsule\_purchase: "Cápsula del Tiempo ($49)",

video\_message\_purchase: "Video de Legacy ($79)",

business\_creator\_title: "Para Negocios y Creadores",

business\_creator\_desc: "Preserva historias de marca o crea personas IA únicas.",

business\_plan: "Plan de Negocios: $499/año",

creator\_plan: "Plan de Creador: $99 configuración + $25/mes",

community\_plan: "Bóveda Comunitaria: $149/año",

contact\_sales: "Contactar Ventas",

login\_title: "Inicio de Sesión en Portal Alegre",

email\_label: "Correo Electrónico",

email\_input\_placeholder: "Correo Electrónico",

password\_label: "Contraseña",

password\_input\_placeholder: "Contraseña",

login\_button: "Iniciar Sesión",

forgot\_password: "¿Olvidaste la Contraseña?",

email\_required: "Por favor ingresa tu dirección de correo electrónico.",

password\_reset\_sent: "Correo de restablecimiento de contraseña enviado! Revisa tu bandeja de entrada.",

login\_required: "Por favor inicia sesión para grabar un...",

current\_plan: "Estás en {{planName}} 💛",

welcome\_user: "Bienvenido, !",

happy\_timeline\_title: "Tu Línea de Tiempo Feliz",

happy\_timeline\_desc: "Agrega hitos, primeros y momentos alegres a tu álbum digital.",

add\_milestone: "Agregar Hito",

bucket\_list\_title: "Lista de Deseos Familiar",

bucket\_list\_desc: "Establece metas compartidas como viajes o actividades divertidas y rastrea el progreso!",

add\_goal\_placeholder: "Agregar una meta...",

add\_goal: "Agregar Meta",

storytelling\_mode\_title: "Aventura de Narración IA",

storytelling\_mode\_desc: "Tu copiloto en la creación del viaje único de tu familia, creando historias ficticias divertidas con tus seres queridos como héroes!",

story\_characters\_label: "¿Quiénes son los personajes principales?",

story\_characters\_placeholder: "e.g., Mamá, Papá, Lily, nuestro perro Sparky",

story\_theme\_label: "Elige un tema:",

select\_theme\_option: "Selecciona un tema",

generate\_story: "Generar Historia",

challenge\_response\_title: "Responder a Desafíos",

challenge\_response\_desc: "Ver y responder a desafíos familiares.",

view\_challenges: "Ver Desafíos",

account\_settings\_title: "Configuraciones de Cuenta",

account\_settings\_desc: "Gestiona tu perfil, suscripción y preferencias.",

manage\_settings: "Gestionar Configuraciones",

contact\_support\_title: "Soporte",

contact\_support\_desc: "Obtén ayuda o proporciona retroalimentación.",

contact\_us: "Contáctanos",

logout\_button: "Cerrar Sesión",

demo\_timeline\_title: "Vista Previa de una Línea de Tiempo Feliz",

demo\_timeline\_desc: "¡Ve cómo Legacy AI da vida a los recuerdos!",

login\_to\_create: "Inicia Sesión para Crear Tu Propio",

signup\_title: "Regístrate en el Portal Alegre",

signup\_button: "Regístrate",

signup\_link: "Regístrate",

login\_link: "¿Ya tienes una cuenta? Iniciar Sesión"

},

fr: {

home\_title: "Capturez les Moments Joyeux de la Vie avec Legacy AI",

home\_lead: "Revivez le passé, célébrez le présent et façonnez l'avenir avec une bonheur curé par IA. Créez Ensemble, Souvenez-vous Pour Toujours : Construisez des albums numériques vibrants et des chronologies heureuses pour les générations.",

get\_started\_free: "Commencer Gratuitement",

download\_joyful\_guide: "Télécharger le Guide de Mémoire Joyeuse",

how\_it\_works\_title: "Comment Legacy AI Diffuse la Joie",

capture\_moments\_step: "1. Capturez des Moments Heureux",

capture\_moments\_desc: "Téléchargez des photos, vidéos et notes vocales d'anniversaires, voyages et joies quotidiennes.",

create\_adventures\_step: "2. Créez des Aventures Familiales",

create\_adventures\_desc: "IA crée des histoires interactives et des jeux mettant en vedette vos proches.",

build\_timeline\_step: "3. Construisez une Chronologie Heureuse",

build\_timeline\_desc: "Organisez les jalons dans un album numérique vibrant pour que tous le chérissent.",

moments\_title: "Moments Qui Comptent",

moments\_intro: "Revivez la joie des anniversaires, voyages familiaux, blagues internes et moments quotidiens. Téléchargez vos souvenirs les plus heureux pour créer un album numérique qui provoque des sourires.",

upload\_photos\_title: "Photos de Joie",

upload\_photos\_desc: "Partagez des instantanés de rires et d'amour.",

record\_voice\_title: "Notes Vocales",

record\_voice\_desc: "Capturez des blagues, chansons ou messages sincères.",

share\_videos\_title: "Vidéos Heureuses",

share\_videos\_desc: "Téléchargez des clips de célébrations et d'aventures.",

emotion\_tag\_placeholder: "Taggez des émotions ou thèmes (e.g., #joyeux, #aventure)",

upload\_now: "Télécharger Maintenant",

record\_now: "Enregistrer Maintenant",

stop\_recording: "Arrêter l'Enregistrement",

upload\_failed: "Téléchargement échoué :",

upload\_success: "Note vocale téléchargée avec succès ! Conversion MP3 en cours.",

mp3\_conversion\_failed: "Note vocale téléchargée, mais conversion MP3 échouée. Lecture WebM.",

microphone\_error: "Accès au microphone refusé ou indisponible : ",

challenges\_title: "Défis Familiaux Interactifs",

challenges\_intro: "Rapprochez votre famille avec des prompts amusants qui créent des souvenirs joyeux. Complétez des défis et construisez un archive positif ensemble !",

funny\_question\_title: "Défi de Question Drôle",

funny\_question\_desc: "Posez une question hilarante à vos grands-parents aujourd'hui et enregistrez leur réponse !",

dance\_off\_title: "Danse Familiale",

dance\_off\_desc: "Enregistrez une danse familiale de 60 secondes et partagez le fun !",

childhood\_memory\_title: "Histoire de Souvenir d'Enfance",

childhood\_memory\_desc: "Partagez une histoire de 60 secondes de votre moment d'enfance le plus heureux.",

start\_challenge: "Soumettre la Réponse",

pricing\_title: "Choisissez Votre Plan Joyeux",

pricing\_intro: "Commencez gratuitement ou débloquez des fonctionnalités premium pour créer des souvenirs familiaux inoubliables.",

free\_tier\_title: "Legacy Basique",

free\_tier\_desc: "Archive limitée de texte et photos, faits saillants de mémoire IA, 5 interactions IA/mois.",

essential\_tier\_title: "Legacy Essentiel",

essential\_tier\_desc: "Stockage étendu, persona IA personnalisée, support multilingue, recherche basée sur l'humeur.",

family\_tier\_title: "Legacy Familial Plus",

family\_tier\_desc: "Archives partagées, albums collaboratifs, interactions IA illimitées, jeux familiaux.",

lifetime\_tier\_title: "Voûte Legacy à Vie",

lifetime\_tier\_desc: "Stockage illimité, livre legacy IA, support prioritaire, messages programmés, et \*\*support VIP familial exclusif avec curations de mémoire IA personnalisées.\*\*",

buy\_now: "Acheter Maintenant",

one\_time\_title: "Améliorations Premium",

one\_time\_desc: "Ajoutez des fonctionnalités uniques à votre legacy joyeuse.",

storybook\_purchase: "Livre de Contes IA ($99)",

time\_capsule\_purchase: "Capsule Temporelle ($49)",

video\_message\_purchase: "Vidéo Legacy ($79)",

business\_creator\_title: "Pour les Entreprises & Créateurs",

business\_creator\_desc: "Préservez les histoires de marque ou créez des personas IA uniques.",

business\_plan: "Plan d'Entreprise : $499/an",

creator\_plan: "Plan Créateur : $99 configuration + $25/mois",

community\_plan: "Voûte Communautaire : $149/an",

contact\_sales: "Contacter les Ventes",

login\_title: "Connexion au Portail Joyeux",

email\_label: "Email",

email\_input\_placeholder: "Email",

password\_label: "Mot de Passe",

password\_input\_placeholder: "Mot de Passe",

login\_button: "Se Connecter",

forgot\_password: "Mot de Passe Oublié ?",

email\_required: "Veuillez entrer votre adresse email.",

password\_reset\_sent: "Email de réinitialisation de mot de passe envoyé ! Vérifiez votre boîte de réception.",

login\_required: "Veuillez vous connecter pour enregistrer un...",

current\_plan: "Vous êtes sur {{planName}} 💛",

welcome\_user: "Bienvenue, !",

happy\_timeline\_title: "Votre Chronologie Heureuse",

happy\_timeline\_desc: "Ajoutez des jalons, des premiers et des moments joyeux à votre album numérique.",

add\_milestone: "Ajouter un Jalon",

bucket\_list\_title: "Liste de Souhaits Familiale",

bucket\_list\_desc: "Définissez des objectifs partagés comme des voyages ou des activités amusantes et suivez le progrès !",

add\_goal\_placeholder: "Ajouter un objectif...",

add\_goal: "Ajouter un Objectif",

storytelling\_mode\_title: "Aventure de Narration IA",

storytelling\_mode\_desc: "Votre copilote pour créer le voyage unique de votre famille, en créant des histoires fictives amusantes avec vos proches comme héros !",

story\_characters\_label: "Qui sont les personnages principaux ?",

story\_characters\_placeholder: "e.g., Maman, Papa, Lily, notre chien Sparky",

story\_theme\_label: "Choisissez un thème :",

select\_theme\_option: "Sélectionnez un thème",

generate\_story: "Générer l'Histoire",

challenge\_response\_title: "Répondre aux Défis",

challenge\_response\_desc: "Voir et répondre aux défis familiaux.",

view\_challenges: "Voir les Défis",

account\_settings\_title: "Paramètres du Compte",

account\_settings\_desc: "Gérez votre profil, abonnement et préférences.",

manage\_settings: "Gérer les Paramètres",

contact\_support\_title: "Support",

contact\_support\_desc: "Obtenez de l'aide ou fournissez des commentaires.",

contact\_us: "Nous Contacter",

logout\_button: "Se Déconnecter",

demo\_timeline\_title: "Aperçu d'une Chronologie Heureuse",

demo\_timeline\_desc: "Voyez comment Legacy AI donne vie aux souvenirs !",

login\_to\_create: "Se Connecter pour Créer le Vôtre",

signup\_title: "Inscrivez-vous au Portail Joyeux",

signup\_button: "S'inscrire",

signup\_link: "S'inscrire",

login\_link: "Déjà un compte ? Se Connecter"

},

de: {

// Add translations for German if needed, but since not provided, leave empty or add placeholders

},

pt: {

// Add translations for Portuguese if needed, but since not provided, leave empty or add placeholders

}

};

function applyTranslations() {

document.querySelectorAll('[data-key]').forEach(el => {

const key = el.getAttribute('data-key');

if (translations[currentLang][key]) {

el.textContent = translations[currentLang][key];

}

});

document.querySelectorAll('[data-placeholder-key]').forEach(el => {

const key = el.getAttribute('data-placeholder-key');

if (translations[currentLang][key]) {

el.placeholder = translations[currentLang][key];

}

});

// To fix bug with elements containing inner HTML (e.g., welcome\_user with span)

if (document.getElementById('userName')) {

document.querySelector('[data-key="welcome\_user"]').innerHTML = translations[currentLang].welcome\_user.replace('!', '<span id="userName"></span>!');

}

}

document.querySelectorAll('.dropdown-item[data-lang]').forEach(item => {

item.addEventListener('click', function(e) {

e.preventDefault();

currentLang = this.getAttribute('data-lang');

applyTranslations();

});

});

applyTranslations();

document.querySelectorAll('a[data-page]').forEach(link => {

link.addEventListener('click', function(e) {

e.preventDefault();

document.querySelectorAll('.page-section').forEach(section => {

section.classList.remove('active');

});

const targetPage = this.getAttribute('data-page');

document.getElementById(targetPage).classList.add('active');

if (targetPage === 'client-portal') {

authService.onAuthStateChanged(user => {

if (user) {

document.getElementById('login-form').style.display = 'none';

document.getElementById('signup-form').style.display = 'none';

document.getElementById('dashboard-content').style.display = 'block';

document.getElementById('userName').textContent = user.email;

updatePlanStatus(user.uid);

} else {

document.getElementById('login-form').style.display = 'block';

document.getElementById('signup-form').style.display = 'none';

document.getElementById('dashboard-content').style.display = 'none';

}

});

}

});

});

document.getElementById('showSignup').addEventListener('click', function(e) {

e.preventDefault();

document.getElementById('login-form').style.display = 'none';

document.getElementById('signup-form').style.display = 'block';

});

document.getElementById('showLogin').addEventListener('click', function(e) {

e.preventDefault();

document.getElementById('signup-form').style.display = 'none';

document.getElementById('login-form').style.display = 'block';

});

document.getElementById('loginForm').addEventListener('submit', async function(e) {

e.preventDefault();

const email = document.getElementById('clientEmail').value;

const password = document.getElementById('clientPassword').value;

try {

await authService.signInWithEmailAndPassword(email, password);

showToast('Logged in successfully!', 'success');

document.getElementById('login-form').style.display = 'none';

document.getElementById('dashboard-content').style.display = 'block';

document.getElementById('userName').textContent = authService.currentUser.email;

updatePlanStatus(authService.currentUser.uid);

} catch (error) {

showToast(`Login failed: ${error.message}`, 'error');

console.error('Login error:', error);

}

});

document.getElementById('signupForm').addEventListener('submit', async function(e) {

e.preventDefault();

const email = document.getElementById('signupEmail').value;

const password = document.getElementById('signupPassword').value;

try {

await authService.createUserWithEmailAndPassword(email, password);

showToast('Signed up successfully!', 'success');

document.getElementById('signup-form').style.display = 'none';

document.getElementById('dashboard-content').style.display = 'block';

document.getElementById('userName').textContent = authService.currentUser.email;

await firestore.collection('users').doc(authService.currentUser.uid).set({ plan: 'basic' });

updatePlanStatus(authService.currentUser.uid);

} catch (error) {

showToast(`Signup failed: ${error.message}`, 'error');

console.error('Signup error:', error);

}

});

document.getElementById('logoutButton').addEventListener('click', async function(e) {

e.preventDefault();

try {

await authService.signOut();

showToast('Logged out successfully!', 'info');

document.getElementById('login-form').style.display = 'block';

document.getElementById('dashboard-content').style.display = 'none';

} catch (error) {

showToast(`Logout failed: ${error.message}`, 'error');

console.error('Logout error:', error);

}

});

document.querySelector('[data-key="forgot\_password"]').addEventListener('click', async function(e) {

e.preventDefault();

const email = document.getElementById('clientEmail').value;

if (email) {

try {

await authService.sendPasswordResetEmail(email);

showToast(translations[currentLang].password\_reset\_sent, 'info');

} catch (error) {

showToast(`Password reset failed: ${error.message}`, 'error');

console.error('Password reset error:', error);

}

} else {

showToast(translations[currentLang].email\_required, 'error');

}

});

async function updatePlanStatus(userId) {

const planStatusElement = document.getElementById('plan-status');

try {

const userDoc = await firestore.collection('users').doc(userId).get();

if (userDoc.exists && userDoc.data().plan) {

const planName = userDoc.data().plan;

planStatusElement.textContent = translations[currentLang].current\_plan.replace('{{planName}}', planName);

planStatusElement.style.display = 'block';

} else {

planStatusElement.style.display = 'none';

}

} catch (error) {

console.error("Error fetching plan status:", error);

planStatusElement.style.display = 'none';

}

}

async function uploadFile(file, type, emotionTagsInputId, progressBarId, progressContainerId) {

if (!authService.currentUser) {

showToast(translations[currentLang].login\_required, 'error');

return;

}

const storageRef = storageService.ref();

const fileRef = storageRef.child(`${type}s/${authService.currentUser.uid}/${file.name}`);

const uploadTask = fileRef.put(file);

const progressBar = document.getElementById(progressBarId);

const progressContainer = document.getElementById(progressContainerId);

progressContainer.style.display = 'block';

uploadTask.on('state\_changed',

(snapshot) => {

const progress = (snapshot.bytesTransferred / snapshot.totalBytes) \* 100;

progressBar.style.width = progress + '%';

progressBar.textContent = Math.round(progress) + '%';

},

(error) => {

showToast(`${translations[currentLang].upload\_failed} ${error.message}`, 'error');

console.error("Upload failed:", error);

progressContainer.style.display = 'none';

},

() => {

uploadTask.snapshot.ref.getDownloadURL().then(async (downloadURL) => {

showToast(`${type} uploaded successfully!`, 'success');

progressContainer.style.display = 'none';

const emotionTags = document.getElementById(emotionTagsInputId).value;

await firestore.collection('users').doc(authService.currentUser.uid).collection(type).add({

fileName: file.name,

fileType: file.type,

downloadURL: downloadURL,

emotionTags: emotionTags,

timestamp: firebase.firestore.FieldValue.serverTimestamp()

});

document.getElementById(emotionTagsInputId).value = '';

});

}

);

}

document.getElementById('uploadPhotoButton').addEventListener('click', function() {

const fileInput = document.getElementById('uploadPhotoInput');

if (fileInput.files.length > 0) {

Array.from(fileInput.files).forEach(file => {

uploadFile(file, 'photo', 'photoEmotionInput', 'photoProgressBar', 'photoProgressContainer');

});

fileInput.value = '';

} else {

showToast('Please select a photo to upload.', 'info');

}

});

document.getElementById('uploadVideoButton').addEventListener('click', function() {

const fileInput = document.getElementById('uploadVideoInput');

if (fileInput.files.length > 0) {

Array.from(fileInput.files).forEach(file => {

uploadFile(file, 'video', 'videoEmotionInput', 'videoProgressBar', 'videoProgressContainer');

});

fileInput.value = '';

} else {

showToast('Please select a video to upload.', 'info');

}

});

document.getElementById('recordVoiceButton').addEventListener('click', async function() {

if (!authService.currentUser) {

showToast(translations[currentLang].login\_required, 'error');

return;

}

try {

const stream = await navigator.mediaDevices.getUserMedia({ audio: true });

mediaRecorder = new MediaRecorder(stream);

audioChunks = [];

let timerSeconds = 0;

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

mediaRecorder.ondataavailable = (event) => {

audioChunks.push(event.data);

};

mediaRecorder.onstop = async () => {

audioBlob = new Blob(audioChunks, { type: 'audio/webm' });

const audioUrl = URL.createObjectURL(audioBlob);

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

audioPreview.src = audioUrl;

audioPreview.style.display = 'block';

const fileName = `voice\_note\_${new Date().getTime()}.webm`;

await uploadFile(audioBlob, 'voice', 'voiceEmotionInput', 'voiceProgressBar', 'voiceProgressContainer');

clearInterval(recordingTimerInterval);

recordingTimer.style.display = 'none';

document.getElementById('recordVoiceButton').style.display = 'block';

document.getElementById('stopRecordingButton').style.display = 'none';

};

mediaRecorder.start();

recordingStartTime = Date.now();

recordingTimer.textContent = '00:00';

recordingTimer.style.display = 'block';

document.getElementById('recordVoiceButton').style.display = 'none';

document.getElementById('stopRecordingButton').style.display = 'block';

recordingTimerInterval = setInterval(() => {

timerSeconds = Math.floor((Date.now() - recordingStartTime) / 1000);

const minutes = String(Math.floor(timerSeconds / 60)).padStart(2, '0');

const seconds = String(timerSeconds % 60).padStart(2, '0');

recordingTimer.textContent = `${minutes}:${seconds}`;

}, 1000);

} catch (error) {

showToast(`${translations[currentLang].microphone\_error} ${error.message}`, 'error');

console.error("Microphone access error:", error);

}

});

document.getElementById('stopRecordingButton').addEventListener('click', function() {

if (mediaRecorder && mediaRecorder.state === 'recording') {

mediaRecorder.stop();

mediaRecorder.stream.getTracks().forEach(track => track.stop());

}

});

document.querySelectorAll('[data-challenge-id]').forEach(button => {

button.addEventListener('click', async function() {

if (!authService.currentUser) {

showToast(translations[currentLang].login\_required, 'error');

return;

}

const challengeId = this.dataset.challengeId;

let responseData = {};

switch (challengeId) {

case 'funny-question':

responseData.answer = document.getElementById('funnyQuestionResponse').value;

if (!responseData.answer) {

showToast('Please enter an answer for the funny question challenge.', 'info');

return;

}

break;

case 'dance-off':

const videoFile = document.getElementById('danceOffVideoInput').files[0];

if (!videoFile) {

showToast('Please upload a video for the dance-off challenge.', 'info');

return;

}

await uploadFile(videoFile, 'challenge\_video', '', 'videoProgressBar', 'videoProgressContainer');

responseData.videoUploaded = true;

break;

case 'childhood-memory':

responseData.story = document.getElementById('childhoodMemoryResponse').value;

if (!responseData.story) {

showToast('Please enter your story for the childhood memory challenge.', 'info');

return;

}

break;

}

try {

await firestore.collection('users').doc(authService.currentUser.uid).collection('challenges').add({

challengeId: challengeId,

responseData: responseData,

timestamp: firebase.firestore.FieldValue.serverTimestamp()

});

showToast('Challenge response submitted!', 'success');

if (challengeId === 'funny-question') document.getElementById('funnyQuestionResponse').value = '';

if (challengeId === 'dance-off') document.getElementById('danceOffVideoInput').value = '';

if (challengeId === 'childhood-memory') document.getElementById('childhoodMemoryResponse').value = '';

} catch (error) {

showToast(`Challenge submission failed: ${error.message}`, 'error');

console.error('Challenge submission error:', error);

}

});

});

document.getElementById('addMilestoneButton').addEventListener('click', async function() {

if (!authService.currentUser) {

showToast(translations[currentLang].login\_required, 'error');

return;

}

const milestone = prompt('Enter your milestone:');

if (milestone) {

try {

await firestore.collection('users').doc(authService.currentUser.uid).collection('timeline').add({

milestone: milestone,

timestamp: firebase.firestore.FieldValue.serverTimestamp()

});

showToast('Milestone added!', 'success');

} catch (error) {

showToast(`Failed to add milestone: ${error.message}`, 'error');

}

}

});

document.getElementById('addGoalButton').addEventListener('click', async function() {

if (!authService.currentUser) {

showToast(translations[currentLang].login\_required, 'error');

return;

}

const goal = document.getElementById('goalInput').value;

if (goal) {

try {

await firestore.collection('users').doc(authService.currentUser.uid).collection('bucketList').add({

goal: goal,

completed: false,

timestamp: firebase.firestore.FieldValue.serverTimestamp()

});

showToast('Goal added!', 'success');

document.getElementById('goalInput').value = '';

} catch (error) {

showToast(`Failed to add goal: ${error.message}`, 'error');

}

} else {

showToast('Please enter a goal.', 'info');

}

});

document.getElementById('generateStoryButton').addEventListener('click', async function() {

if (!authService.currentUser) {

showToast(translations[currentLang].login\_required, 'error');

return;

}

const characters = document.getElementById('storyCharacters').value;

const theme = document.getElementById('storyTheme').value;

if (!characters || !theme) {

showToast('Please enter characters and select a theme.', 'info');

return;

}

showToast('Generating your story...', 'info');

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

storyOutput.textContent = `Once upon a time, in a land far away, ${characters} embarked on a grand ${theme} adventure... (This is a placeholder story generated by AI.)`;

storyOutput.style.display = 'block';

document.getElementById('storyActions').style.display = 'flex';

showToast('Story generated!', 'success');

});

document.getElementById('saveStoryButton').addEventListener('click', async function() {

if (!authService.currentUser) return;

const storyContent = document.getElementById('storyOutput').textContent;

try {

await firestore.collection('users').doc(authService.currentUser.uid).collection('stories').add({

content: storyContent,

timestamp: firebase.firestore.FieldValue.serverTimestamp()

});

showToast('Story saved!', 'success');

} catch (error) {

showToast(`Failed to save story: ${error.message}`, 'error');

console.error('Error saving story:', error);

}

});

document.getElementById('shareStoryButton').addEventListener('click', function() {

if (!authService.currentUser) return;

const storyContent = document.getElementById('storyOutput').textContent;

navigator.clipboard.writeText(storyContent).then(() => {

showToast('Story copied to clipboard!', 'info');

}).catch(err => {

showToast('Failed to copy story.', 'error');

console.error('Could not copy text: ', err);

});

});

document.getElementById('regenerateStoryButton').addEventListener('click', function() {

document.getElementById('generateStoryButton').click();

});

async function createPayPalOrder(amount, planName) {

const response = await fetch('https://us-central1-life-archive-1ed4a.cloudfunctions.net/createOrder', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify({

amount: amount,

plan: planName

})

});

if (!response.ok) {

throw new Error('Failed to create order');

}

const data = await response.json();

return data.id;

}

async function onPayPalApprove(data) {

const response = await fetch('https://us-central1-life-archive-1ed4a.cloudfunctions.net/captureOrder', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify({

orderID: data.orderID

})

});

if (!response.ok) {

throw new Error('Failed to capture order');

}

const details = await response.json();

showToast(`Transaction completed!`, 'success');

// Update user plan in Firestore or handle subscription

if (authService.currentUser) {

await firestore.collection('users').doc(authService.currentUser.uid).update({

plan: details.purchase\_units[0].reference\_id // assuming planName stored there

});

}

return details;

}

function onPayPalError(err) {

showToast(`Payment failed: ${err.message || err}`, 'error');

console.error('PayPal error:', err);

}

function initPayPalButtons(containerId, amount, planName) {

paypal.Buttons({

style: {

layout: 'vertical',

color: 'blue',

shape: 'rect',

label: 'paypal'

},

createOrder: () => createPayPalOrder(amount, planName),

onApprove: onPayPalApprove,

onError: onPayPalError

}).render(`#${containerId}`);

}

function initCardFields(containerId, submitButtonId, amount, planName) {

const cardFields = paypal.CardFields({

createOrder: () => createPayPalOrder(amount, planName),

onApprove: onPayPalApprove,

onError: onPayPalError

});

if (cardFields.isEligible()) {

cardFields.NumberField().render('#card-number-' + containerId.split('-')[1]);

cardFields.ExpiryField().render('#card-expiry-' + containerId.split('-')[1]);

cardFields.CVVField().render('#card-cvv-' + containerId.split('-')[1]);

document.getElementById(submitButtonId).addEventListener('click', () => {

cardFields.submit().catch(onPayPalError);

});

} else {

document.getElementById(containerId).style.display = 'none';

}

}

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

initPayPalButtons('paypal-essential', '15.00', 'Essential');

initCardFields('card-fields-essential', 'card-submit-essential', '15.00', 'Essential');

initPayPalButtons('paypal-family', '39.00', 'Family Plus');

initCardFields('card-fields-family', 'card-submit-family', '39.00', 'Family Plus');

initPayPalButtons('paypal-lifetime', '299.00', 'Lifetime');

initCardFields('card-fields-lifetime', 'card-submit-lifetime', '299.00', 'Lifetime');

// For premium enhancements

initPayPalButtons('paypal-storybook', '99.00', 'AI Storybook');

initPayPalButtons('paypal-timecapsule', '49.00', 'Time Capsule');

initPayPalButtons('paypal-legacyvideo', '79.00', 'Legacy Video');

document.getElementById('current-year').textContent = new Date().getFullYear();

});