EVERYTHING IS WORKING AS PER REQ  
  
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>EduPro Academy - Professional Learning Platform</title>

<script src="https://cdn.tailwindcss.com"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/jspdf/2.5.1/jspdf.umd.min.js"></script>

<script src="https://www.gstatic.com/firebasejs/9.23.0/firebase-app-compat.js"></script>

<script src="https://www.gstatic.com/firebasejs/9.23.0/firebase-auth-compat.js"></script>

<script src="https://www.gstatic.com/firebasejs/9.23.0/firebase-database-compat.js"></script>

<script src="https://cdn.jsdelivr.net/npm/chart.js@4.4.2/dist/chart.umd.min.js"></script>

<style>

body {

box-sizing: border-box;

}

.fade-in {

animation: fadeIn 0.8s ease-out;

}

@keyframes fadeIn {

from {

opacity: 0;

transform: translateY(30px);

}

to {

opacity: 1;

transform: translateY(0);

}

}

.slide-in {

animation: slideIn 0.6s ease-out;

}

@keyframes slideIn {

from {

opacity: 0;

transform: translateX(-30px);

}

to {

opacity: 1;

transform: translateX(0);

}

}

.hero-gradient {

background: linear-gradient(135deg, #667eea 0%, #764ba2 50%, #f093fb 100%);

}

.card-hover {

transition: all 0.4s cubic-bezier(0.4, 0, 0.2, 1);

}

.card-hover:hover {

transform: translateY(-8px) scale(1.02);

box-shadow: 0 25px 50px -12px rgba(0, 0, 0, 0.25);

}

.progress-bar {

transition: width 0.6s cubic-bezier(0.4, 0, 0.2, 1);

}

.locked {

opacity: 0.5;

pointer-events: none;

}

.certificate-bg {

background: linear-gradient(135deg, #1e3c72 0%, #2a5298 50%, #667eea 100%);

border: 12px solid #ffd700;

box-shadow: 0 0 40px rgba(0, 0, 0, 0.4);

}

.nav-glass {

backdrop-filter: blur(10px);

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

}

.stats-counter {

animation: countUp 2s ease-out;

}

@keyframes countUp {

from {

opacity: 0;

transform: scale(0.5);

}

to {

opacity: 1;

transform: scale(1);

}

}

.floating {

animation: floating 3s ease-in-out infinite;

}

@keyframes floating {

0%,

100% {

transform: translateY(0px);

}

50% {

transform: translateY(-10px);

}

}

.pulse-glow {

animation: pulseGlow 2s infinite;

}

@keyframes pulseGlow {

0%,

100% {

box-shadow: 0 0 20px rgba(102, 126, 234, 0.4);

}

50% {

box-shadow: 0 0 30px rgba(102, 126, 234, 0.8);

}

}

.language-filter.active {

background-color: #4f46e5;

color: white;

}

.language-filter {

color: #6b7280;

}

/\* Modal Styles \*/

.modal {

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

background-color: rgba(0, 0, 0, 0.5);

display: flex;

justify-content: center;

align-items: center;

z-index: 1000;

}

.modal-content {

background-color: white;

border-radius: 1rem;

padding: 2rem;

max-width: 90%;

max-height: 90vh;

overflow-y: auto;

position: relative;

}

</style>

</head>

<body class="bg-gradient-to-br from-blue-50 to-indigo-100 min-h-screen">

<nav class="nav-glass shadow-xl fixed w-full top-0 z-50">

<div class="max-w-7xl mx-auto px-6">

<div class="flex justify-between items-center py-4">

<div class="flex items-center space-x-3">

<div class="w-10 h-10 bg-gradient-to-r from-indigo-600 to-purple-600 rounded-lg flex items-center justify-center">

<span class="text-white font-bold text-lg">E</span>

</div>

<h1 class="text-2xl font-bold bg-gradient-to-r from-indigo-600 to-purple-600 bg-clip-text text-transparent">EduPro Academy</h1>

</div>

<div id="nav-buttons" class="flex items-center space-x-4">

<button onclick="showAdminLogin()" class="bg-yellow-500 text-gray-900 px-4 py-2.5 rounded-full hover:shadow-lg transition-all duration-300 font-medium">Admin</button>

<button onclick="showLogin()" class="bg-gradient-to-r from-indigo-600 to-purple-600 text-white px-6 py-2.5 rounded-full hover:shadow-lg transition-all duration-300 font-medium">Sign In</button>

<button onclick="showRegister()" class="bg-gradient-to-r from-green-500 to-emerald-600 text-white px-6 py-2.5 rounded-full hover:shadow-lg transition-all duration-300 font-medium">Get Started</button>

</div>

<div id="user-nav" class="hidden flex items-center space-x-2">

<div class="flex items-center space-x-1 bg-white rounded-full px-4 py-2 shadow-md">

<div class="w-8 h-8 bg-gradient-to-r from-indigo-500 to-purple-500 rounded-full flex items-center justify-center">

<span class="text-white text-sm font-semibold" id="user-avatar"></span>

</div>

<span class="text-gray-700 font-medium" id="user-name-nav"></span>

</div>

<button onclick="showDashboard()" class="bg-indigo-600 text-white px-4 py-2 rounded-lg hover:bg-indigo-700 transition-colors">Dashboard</button>

<button onclick="showProfile()" class="bg-purple-600 text-white px-4 py-2 rounded-lg hover:bg-purple-700 transition-colors">My Courses</button>

<button onclick="showCertificates()" class="bg-green-600 text-white px-4 py-2 rounded-lg hover:bg-green-700 transition-colors">Certificates</button>

<button onclick="logout()" class="bg-red-500 text-white px-4 py-2 rounded-lg hover:bg-red-600 transition-colors">Logout</button>

</div>

<div id="admin-nav" class="hidden items-center space-x-2">

<span class="text-sm font-bold text-red-600">Admin Mode</span>

<button onclick="showAdminDashboard()" class="bg-blue-600 text-white px-4 py-2 rounded-lg hover:bg-blue-700 transition-colors">Admin Dashboard</button>

<button onclick="adminLogout()" class="bg-red-500 text-white px-4 py-2 rounded-lg hover:bg-red-600 transition-colors">Logout</button>

</div>

</div>

</div>

</nav>

<div class="pt-20">

<div id="welcome-screen" class="fade-in">

<section class="hero-gradient text-white py-20 px-4 sm:px-6">

<div class="max-w-7xl mx-auto text-center">

<div class="floating mb-8">

<div class="inline-block p-4 bg-white bg-opacity-20 rounded-full">

<svg class="w-16 h-16" fill="currentColor" viewBox="0 0 20 20">

<path d="M10.394 2.08a1 1 0 00-.788 0l-7 3a1 1 0 000 1.84L5.25 8.051a.999.999 0 01.356-.257l4-1.714a1 1 0 11.788 1.838L7.667 9.088l1.94.831a1 1 0 00.787 0l7-3a1 1 0 000-1.838l-7-3zM3.31 9.397L5 10.12v4.102a8.969 8.969 0 00-1.05-.174 1 1 0 01-.89-.89 11.115 11.115 0 01.25-3.762zM9.3 16.573A9.026 9.026 0 007 14.935v-3.957l1.818.78a3 3 0 002.364 0l5.508-2.361a11.026 11.026 0 01.25 3.762 1 1 0 01-.89.89 8.968 8.968 0 00-5.35 2.524 1 1 0 01-1.4 0zM6 18a1 1 0 001-1v-2.065a8.935 8.935 0 00-2-.712V17a1 1 0 001 1z"/>

</svg>

</div>

</div>

<h1 class="text-5xl md:text-7xl font-bold mb-6 slide-in">

Transform Your

<span class="bg-gradient-to-r from-yellow-300 to-orange-300 bg-clip-text text-transparent">Career</span>

</h1>

<p class="text-xl md:text-2xl mb-8 text-white text-opacity-90 max-w-3xl mx-auto slide-in">

Join thousands of professionals mastering cutting-edge skills with our expert-designed courses. Learn at your own pace, earn certificates, and advance your career.

</p>

<div class="flex flex-col sm:flex-row gap-4 justify-center mb-12 slide-in">

<button onclick="showRegister()" class="pulse-glow bg-white text-indigo-600 px-8 py-4 rounded-full text-lg font-semibold hover:bg-gray-100 transition-all duration-300">

Start Learning Today

</button>

<button onclick="showCourseList()" class="border-2 border-white text-white px-8 py-4 rounded-full text-lg font-semibold hover:bg-white hover:text-indigo-600 transition-all duration-300">

Explore Courses

</button>

</div>

</div>

</section>

<section class="py-16 bg-white">

<div class="max-w-7xl mx-auto px-4 sm:px-6">

<div class="grid grid-cols-2 md:grid-cols-4 gap-4 md:gap-8 text-center">

<div class="stats-counter">

<div class="text-4xl font-bold text-indigo-600 mb-2">15+</div>

<div class="text-gray-600 font-medium">Expert Courses</div>

</div>

<div class="stats-counter">

<div class="text-4xl font-bold text-purple-600 mb-2">5K+</div>

<div class="text-gray-600 font-medium">Students</div>

</div>

<div class="stats-counter">

<div class="text-4xl font-bold text-green-600 mb-2">98%</div>

<div class="text-gray-600 font-medium">Success Rate</div>

</div>

<div class="stats-counter">

<div class="text-4xl font-bold text-orange-600 mb-2">24/7</div>

<div class="text-gray-600 font-medium">Support</div>

</div>

</div>

</div>

</section>

<section class="py-20 bg-gray-50">

<div class="max-w-7xl mx-auto px-4 sm:px-6">

<div class="text-center mb-16">

<h2 class="text-4xl font-bold text-gray-800 mb-4">Why Choose EduPro Academy?</h2>

<p class="text-xl text-gray-600 max-w-2xl mx-auto">Experience the future of online learning with our innovative platform</p>

</div>

<div class="grid md:grid-cols-3 gap-4 md:gap-8">

<div class="card-hover bg-white p-8 rounded-2xl shadow-lg text-center">

<div class="w-16 h-16 bg-gradient-to-r from-indigo-500 to-purple-500 rounded-full flex items-center justify-center mx-auto mb-6">

<svg class="w-8 h-8 text-white" fill="currentColor" viewBox="0 0 20 20">

<path d="M9 12l2 2 4-4m6 2a9 9 0 11-18 0 9 9 0 0118 0z"/>

</svg>

</div>

<h3 class="text-2xl font-semibold mb-4">Industry-Recognized Certificates</h3>

<p class="text-gray-600">Earn certificates that employers value and trust</p>

</div>

<div class="card-hover bg-white p-8 rounded-2xl shadow-lg text-center">

<div class="w-16 h-16 bg-gradient-to-r from-green-500 to-emerald-500 rounded-full flex items-center justify-center mx-auto mb-6">

<svg class="w-8 h-8 text-white" fill="currentColor" viewBox="0 0 20 20">

<path d="M13 6a3 3 0 11-6 0 3 3 0 016 0zM18 8a2 2 0 11-4 0 2 2 0 014 0zM14 15a4 4 0 00-8 0v3h8v-3z"/>

</svg>

</div>

<h3 class="text-2xl font-semibold mb-4">Expert Instructors</h3>

<p class="text-gray-600">Learn from industry professionals with real-world experience</p>

</div>

<div class="card-hover bg-white p-8 rounded-2xl shadow-lg text-center">

<div class="w-16 h-16 bg-gradient-to-r from-orange-500 to-red-500 rounded-full flex items-center justify-center mx-auto mb-6">

<svg class="w-8 h-8 text-white" fill="currentColor" viewBox="0 0 20 20">

<path d="M3 4a1 1 0 011-1h12a1 1 0 011 1v2a1 1 0 01-1 1H4a1 1 0 01-1-1V4zM3 10a1 1 0 011-1h6a1 1 0 011 1v6a1 1 0 01-1 1H4a1 1 0 01-1-1v-6zM14 9a1 1 0 00-1 1v6a1 1 0 001 1h2a1 1 0 001-1v-6a1 1 0 00-1-1h-2z"/>

</svg>

</div>

<h3 class="text-2xl font-semibold mb-4">Progress Tracking</h3>

<p class="text-gray-600">Monitor your learning journey with detailed analytics</p>

</div>

</div>

</div>

</section>

<section class="py-20 bg-white">

<div class="max-w-7xl mx-auto px-4 sm:px-6">

<div class="text-center mb-16">

<h2 class="text-4xl font-bold text-gray-800 mb-4">Popular Courses</h2>

<p class="text-xl text-gray-600">Start your learning journey with our most popular courses</p>

</div>

<div class="grid md:grid-cols-3 gap-4 md:gap-8" id="popular-courses-preview">

</div>

<div class="text-center mt-12">

<button onclick="showCourseList()" class="bg-gradient-to-r from-indigo-600 to-purple-600 text-white px-8 py-4 rounded-full text-lg font-semibold hover:shadow-lg transition-all duration-300">

View All Courses

</button>

</div>

</div>

</section>

</div>

<div id="admin-login-screen" class="hidden max-w-md mx-auto bg-white p-8 rounded-lg shadow-lg fade-in">

<h2 class="text-2xl font-bold text-center mb-6 text-red-600">Admin Login</h2>

<p class="text-center text-sm text-gray-600 mb-4">Use: admin@edupro.com / admin123</p>

<form onsubmit="handleAdminLogin(event)">

<div class="mb-4">

<label class="block text-gray-700 text-sm font-bold mb-2">Admin Email</label>

<input type="email" id="admin-email" class="w-full px-3 py-2 border rounded-lg focus:outline-none focus:border-red-500" value="admin@edupro.com" required>

</div>

<div class="mb-6">

<label class="block text-gray-700 text-sm font-bold mb-2">Admin Password</label>

<input type="password" id="admin-password" class="w-full px-3 py-2 border rounded-lg focus:outline-none focus:border-red-500" value="admin123" required>

</div>

<button type="submit" class="w-full bg-red-600 text-white py-2 rounded-lg hover:bg-red-700">Login as Admin</button>

</form>

<p class="text-center mt-4">

<button onclick="showWelcomeScreen()" class="text-gray-600 hover:underline">← Back to Main Page</button>

</p>

</div>

<div id="login-form" class="hidden max-w-md mx-auto bg-white p-8 rounded-lg shadow-lg fade-in">

<h2 class="text-2xl font-bold text-center mb-6">Login</h2>

<form onsubmit="handleLogin(event)">

<div class="mb-4">

<label class="block text-gray-700 text-sm font-bold mb-2">Email</label>

<input type="email" id="login-email" class="w-full px-3 py-2 border rounded-lg focus:outline-none focus:border-indigo-500" required>

</div>

<div class="mb-6">

<label class="block text-gray-700 text-sm font-bold mb-2">Password</label>

<input type="password" id="login-password" class="w-full px-3 py-2 border rounded-lg focus:outline-none focus:border-indigo-500" required>

</div>

<button type="submit" class="w-full bg-indigo-600 text-white py-2 rounded-lg hover:bg-indigo-700">Login</button>

</form>

<p class="text-center mt-4">

Don't have an account?

<button onclick="showRegister()" class="text-indigo-600 hover:underline">Register here</button>

</p>

</div>

<div id="register-form" class="hidden max-w-md mx-auto bg-white p-8 rounded-lg shadow-lg fade-in">

<h2 class="text-2xl font-bold text-center mb-6">Register</h2>

<form onsubmit="handleRegister(event)">

<div class="mb-4">

<label class="block text-gray-700 text-sm font-bold mb-2">Full Name</label>

<input type="text" id="register-name" class="w-full px-3 py-2 border rounded-lg focus:outline-none focus:border-green-500" required>

</div>

<div class="mb-4">

<label class="block text-gray-700 text-sm font-bold mb-2">Email</label>

<input type="email" id="register-email" class="w-full px-3 py-2 border rounded-lg focus:outline-none focus:border-green-500" required>

</div>

<div class="mb-6">

<label class="block text-gray-700 text-sm font-bold mb-2">Password</label>

<input type="password" id="register-password" class="w-full px-3 py-2 border rounded-lg focus:outline-none focus:border-green-500" required>

</div>

<button type="submit" class="w-full bg-green-600 text-white py-2 rounded-lg hover:bg-green-700">Register</button>

</form>

<p class="text-center mt-4">

Already have an account?

<button onclick="showLogin()" class="text-green-600 hover:underline">Login here</button>

</p>

</div>

<div id="dashboard-screen" class="hidden fade-in max-w-7xl mx-auto px-4 sm:px-6 py-8">

<div class="mb-8">

<h2 class="text-3xl font-bold text-gray-800 mb-2">Welcome back, <span id="dashboard-user-name"></span>!</h2>

<p class="text-gray-600">Continue your learning journey</p>

</div>

<div class="grid grid-cols-1 md:grid-cols-3 gap-4 md:gap-6 mb-8">

<div class="bg-gradient-to-r from-indigo-500 to-purple-600 text-white p-6 rounded-2xl">

<div class="flex items-center justify-between">

<div>

<p class="text-indigo-100">Enrolled Courses</p>

<p class="text-3xl font-bold" id="enrolled-count">0</p>

</div>

<div class="text-4xl opacity-80">📚</div>

</div>

</div>

<div class="bg-gradient-to-r from-green-500 to-emerald-600 text-white p-6 rounded-2xl">

<div class="flex items-center justify-between">

<div>

<p class="text-green-100">Completed</p>

<p class="text-3xl font-bold" id="completed-count">0</p>

</div>

<div class="text-4xl opacity-80">🏆</div>

</div>

</div>

<div class="bg-gradient-to-r from-orange-500 to-red-500 text-white p-6 rounded-2xl">

<div class="flex items-center justify-between">

<div>

<p class="text-orange-100">Certificates</p>

<p class="text-3xl font-bold" id="certificates-count">0</p>

</div>

<div class="text-4xl opacity-80">🎓</div>

</div>

</div>

</div>

<div class="grid grid-cols-1 lg:grid-cols-3 gap-4 md:gap-8">

<div class="lg:col-span-2">

<div class="bg-white rounded-2xl shadow-lg p-6">

<h3 class="text-xl font-semibold mb-4">Continue Learning</h3>

<div id="continue-learning" class="space-y-4">

</div>

</div>

</div>

<div>

<div class="bg-white rounded-2xl shadow-lg p-6">

<h3 class="text-xl font-semibold mb-4">Quick Actions</h3>

<div class="space-y-3">

<button onclick="showCourseList()" class="w-full bg-indigo-600 text-white py-3 rounded-lg hover:bg-indigo-700 transition-colors">

Browse Courses

</button>

<button onclick="showProfile()" class="w-full bg-purple-600 text-white py-3 rounded-lg hover:bg-purple-700 transition-colors">

My Courses

</button>

<button onclick="showCertificates()" class="w-full bg-green-600 text-white py-3 rounded-lg hover:bg-green-700 transition-colors">

View Certificates

</button>

</div>

</div>

</div>

</div>

</div>

<div id="course-list" class="hidden fade-in max-w-7xl mx-auto px-4 sm:px-6 py-8">

<div class="text-center mb-12">

<h2 class="text-4xl font-bold text-gray-800 mb-4">All Courses</h2>

<p class="text-xl text-gray-600">Choose from our comprehensive course catalog</p>

</div>

<div class="mb-8 flex justify-center">

<div class="bg-white rounded-full p-2 shadow-lg">

<button onclick="filterCourses('all')" class="language-filter active px-6 py-2 rounded-full transition-all">All</button>

<button onclick="filterCourses('programming')" class="language-filter px-6 py-2 rounded-full transition-all">Programming</button>

<button onclick="filterCourses('design')" class="language-filter px-6 py-2 rounded-full transition-all">Design</button>

<button onclick="filterCourses('business')" class="language-filter px-6 py-2 rounded-full transition-all">Business</button>

<button onclick="filterCourses('data')" class="language-filter px-6 py-2 rounded-full transition-all">Data Science</button>

</div>

</div>

<div class="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-4 md:gap-8" id="courses-container"></div>

</div>

<div id="certificates-screen" class="hidden fade-in max-w-7xl mx-auto px-4 sm:px-6 py-8">

<div class="text-center mb-12">

<h2 class="text-4xl font-bold text-gray-800 mb-4">My Certificates</h2>

<p class="text-xl text-gray-600">Your achievements and certifications</p>

</div>

<div id="certificates-container" class="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-4 md:gap-6">

</div>

</div>

<div id="course-detail" class="hidden fade-in">

<button onclick="showCourseList()" class="mb-4 bg-gray-600 text-white px-4 py-2 rounded-lg hover:bg-gray-700">← Back to Courses</button>

<div id="course-detail-content"></div>

</div>

<div id="course-learning" class="hidden fade-in">

<div class="bg-white rounded-lg shadow-lg p-6">

<div class="flex justify-between items-center mb-6">

<button onclick="showCourseList()" class="bg-gray-600 text-white px-4 py-2 rounded-lg hover:bg-gray-700">← Back to Courses</button>

<div class="text-right">

<div class="text-sm text-gray-600">Progress</div>

<div class="text-lg font-semibold" id="course-progress">0%</div>

</div>

</div>

<div id="course-content"></div>

</div>

</div>

<div id="profile-screen" class="hidden fade-in max-w-7xl mx-auto px-4 sm:px-6 py-8">

<h2 class="text-3xl font-bold text-center mb-8">My Courses</h2>

<div id="enrolled-courses" class="flex flex-wrap gap-4 justify-center">

</div>

</div>

<div id="certificate-screen" class="hidden fade-in">

<div class="max-w-4xl mx-auto">

<button onclick="showProfile()" class="mb-4 bg-gray-600 text-white px-4 py-2 rounded-lg hover:bg-gray-700">← Back to My Courses</button>

<div id="certificate-content" class="certificate text-white p-12 rounded-lg text-center"></div>

<div class="text-center mt-6">

<button onclick="downloadCertificate()" class="bg-green-600 text-white px-6 py-3 rounded-lg hover:bg-green-700 text-lg">Download Certificate</button>

</div>

</div>

</div>

<div id="admin-dashboard-panel" class="hidden fade-in">

<div class="min-h-screen flex flex-col">

<header class="bg-white shadow-md z-10">

<div class="max-w-7xl mx-auto py-4 px-4 sm:px-6 lg:px-8 flex justify-between items-center">

<h1 class="text-3xl font-bold text-gray-900 slide-in">Admin Enrollment Dashboard 🔑</h1>

<div class="text-sm text-gray-500">

Last Refreshed: <span id="last-updated" class="font-semibold text-blue-600">Loading...</span>

</div>

</div>

</header>

<main class="flex-1 p-4 sm:p-6 lg:p-8 max-w-7xl mx-auto w-full">

<div class="grid grid-cols-1 md:grid-cols-3 gap-6 mb-8">

<div id="card-total-students" class="card-animation bg-white p-6 rounded-xl shadow-lg border-l-4 border-blue-600">

<div class="flex items-center justify-between">

<div class="text-sm font-medium text-gray-500">Total Student Accounts</div>

<svg class="h-6 w-6 text-blue-400" fill="none" stroke="currentColor" viewBox="0 0 24 24"><path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M17 20h5v-4h-5v4zm0 0l-1.336-1.336m1.336 1.336H3m3-4H2m3 4v-4m3 4v-4m3 4v-4m3 4v-4m3 4v-4m-3 4V4m3 16V4m-3 0h5m-5 0H3m3 0h12"/></svg>

</div>

<p class="text-4xl font-extrabold text-gray-900 mt-1"><span id="metric-total-students">0</span></p>

</div>

<div id="card-total-active-users" class="card-animation bg-white p-6 rounded-xl shadow-lg border-l-4 border-red-600">

<div class="flex items-center justify-between">

<div class="text-sm font-medium text-gray-500">Total Active Users (Mock)</div>

<svg class="h-6 w-6 text-red-400" fill="none" stroke="currentColor" viewBox="0 0 24 24"><path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M13 10V3L4 14h7v7l9-11h-7z"/></svg>

</div>

<p class="text-4xl font-extrabold text-gray-900 mt-1"><span id="metric-total-active-users">0</span></p>

</div>

<div id="card-total-certificates" class="card-animation bg-white p-6 rounded-xl shadow-lg border-l-4 border-green-600">

<div class="flex items-center justify-between">

<div class="text-sm font-medium text-gray-500">Courses Completed (Certified)</div>

<svg class="h-6 w-6 text-green-400" fill="none" stroke="currentColor" viewBox="0 0 24 24"><path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M9 12l2 2 4-4m5.618-4.016A11.955 11.955 0 0112 2.944a11.955 11.955 0 01-8.618 3.04A12.01 12.01 0 002.5 14c0 1.491.31 2.923.863 4.264m18.774-11.28A12.01 12.01 0 0021.5 14c0 1.491-.31 2.923-.863 4.264M18 18h-4.264l-.5-1.5-.5 1.5H6m12 0a2 2 0 100 4 2 2 0 000-4z"/></svg>

</div>

<p class="text-4xl font-extrabold text-gray-900 mt-1"><span id="metric-total-certificates">0</span></p>

</div>

</div>

<div class="grid grid-cols-1 lg:grid-cols-3 gap-6 mb-8">

<div class="lg:col-span-2 bg-white p-6 rounded-xl shadow-lg card-animation">

<h2 class="text-xl font-semibold text-gray-800 mb-4">Course Enrollment Distribution (Live Data)</h2>

<div style="height: 300px;">

<canvas id="courseChart"></canvas>

</div>

</div>

<div class="lg:col-span-1 bg-white p-6 rounded-xl shadow-lg card-animation">

<h2 class="text-xl font-semibold text-gray-800 mb-4">Search & Filter</h2>

<div class="mb-4">

<label for="admin-student-search" class="block text-sm font-medium text-gray-700">Search Student Name/Email</label>

<input type="text" id="admin-student-search" placeholder="E.g., Alice Johnson" class="mt-1 block w-full border border-gray-300 rounded-md shadow-sm py-2 px-3 focus:ring-blue-500 focus:border-blue-500 filter-transition">

</div>

<div class="mb-4">

<label for="admin-course-filter" class="block text-sm font-medium text-gray-700">Filter by Course</label>

<select id="admin-course-filter" class="mt-1 block w-full border border-gray-300 rounded-md shadow-sm py-2 px-3 focus:ring-blue-500 focus:border-blue-500 filter-transition bg-white">

<option value="all">All Courses</option>

</select>

</div>

<div>

<label for="admin-status-filter" class="block text-sm font-medium text-gray-700">Filter by Enrollment Status</label>

<select id="admin-status-filter" class="mt-1 block w-full border border-gray-300 rounded-md shadow-sm py-2 px-3 focus:ring-blue-500 focus:border-blue-500 filter-transition bg-white">

<option value="all">Any Status</option>

<option value="true">Active (True)</option>

<option value="false">Pending (False)</option>

</select>

</div>

</div>

</div>

<div class="bg-white p-6 rounded-xl shadow-lg card-animation">

<h2 class="text-xl font-semibold text-gray-800 mb-4">Student Enrollment Records</h2>

<div class="overflow-x-auto">

<table class="min-w-full divide-y divide-gray-200">

<thead>

<tr class="bg-gray-50">

<th class="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">Student Name</th>

<th class="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">Email</th>

<th class="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">Course Title</th>

<th class="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">Status</th>

<th class="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">Action</th>

</tr>

</thead>

<tbody id="admin-table-body" class="bg-white divide-y divide-gray-200">

</tbody>

</table>

<div id="admin-no-results" class="hidden text-center py-8 text-gray-500">No enrollment records found matching the criteria.</div>

</div>

</div>

</main>

</div>

</div>

</div>

<div id="admin-user-details-modal" class="modal hidden">

<div class="modal-content w-full max-w-xl">

<div class="flex justify-between items-center pb-4 border-b">

<h2 class="text-2xl font-bold text-gray-900" id="modal-user-name">Ganesh's Course Details</h2>

<button onclick="closeAdminUserDetails()" class="text-gray-500 hover:text-gray-700">

<svg class="w-6 h-6" fill="none" stroke="currentColor" viewBox="0 0 24 24"><path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M6 18L18 6M6 6l12 12"></svg>

</button>

</div>

<div id="modal-user-info" class="mt-4 space-y-1 text-sm text-gray-600">

</div>

<div class="mt-6">

<h3 class="text-lg font-semibold text-gray-800 border-b pb-2 mb-4">Enrolled Courses (<span id="modal-enrolled-count">0</span>)</h3>

<div id="modal-course-list" class="space-y-4">

</div>

</div>

</div>

</div>

<script>

// Application State

let currentCourse = null;

let currentDay = 1;

let currentUser = null;

const PAYMENT\_LINK = 'https://rzp.io/rzp/MdrDDje';

let isAdminLoggedIn = false;

const ADMIN\_EMAIL = 'admin@edupro.com';

const ADMIN\_PASSWORD = 'admin123';

let courseChartInstance = null; // To store the Chart.js instance

// Sample Data: COURSE IDs must be unique strings (unchanged)

const courses = [{

id: "web\_dev",

title: "Full Stack Web Development",

description: "Master modern web development with React, Node.js, and databases",

duration: "10 days",

level: "Beginner",

icon: "💻",

category: "programming",

rating: 4.9,

students: 2847,

curriculum: [

"HTML & CSS Fundamentals",

"JavaScript ES6+",

"React Components",

"State Management",

"Node.js Backend",

"Express.js Framework",

"Database Integration",

"API Development",

"Authentication",

"Deployment"

]

}, {

id: "data\_sci",

title: "Data Science & Analytics",

description: "Master Python, machine learning, and data visualization",

duration: "10 days",

level: "Intermediate",

icon: "📊",

category: "data",

rating: 4.8,

students: 1923,

curriculum: [

"Python for Data Science",

"NumPy & Pandas",

"Data Cleaning",

"Statistical Analysis",

"Data Visualization",

"Machine Learning Basics",

"Scikit-learn",

"Model Evaluation",

"Deep Learning Intro",

"Capstone Project"

]

}, {

id: "digital\_mkt",

title: "Digital Marketing Mastery",

description: "Complete digital marketing strategy and execution",

duration: "10 days",

level: "Beginner",

icon: "📈",

category: "business",

rating: 4.7,

students: 3421,

curriculum: [

"Marketing Strategy",

"SEO Optimization",

"Content Marketing",

"Social Media Strategy",

"Email Campaigns",

"Google Analytics",

"Paid Advertising",

"Conversion Optimization",

"Brand Building",

"Campaign Analysis"

]

}, {

id: "mobile\_dev",

title: "Mobile App Development",

description: "Build cross-platform mobile apps with React Native",

duration: "10 days",

level: "Intermediate",

icon: "📱",

category: "programming",

rating: 4.6,

students: 1567,

curriculum: [

"React Native Setup",

"Components & Navigation",

"State Management",

"API Integration",

"Native Features",

"Push Notifications",

"App Store Guidelines",

"Testing & Debugging",

"Performance Optimization",

"Publishing"

]

}, {

id: "cyber\_sec",

title: "Cybersecurity Fundamentals",

description: "Learn ethical hacking and security best practices",

duration: "10 days",

level: "Advanced",

icon: "🔒",

category: "programming",

rating: 4.9,

students: 892,

curriculum: [

"Security Principles",

"Network Security",

"Cryptography",

"Vulnerability Assessment",

"Penetration Testing",

"Incident Response",

"Security Tools",

"Compliance Standards",

"Risk Management",

"Capstone Project"

]

}, {

id: "aws\_cloud",

title: "Cloud Computing with AWS",

description: "Master Amazon Web Services and cloud architecture",

duration: "10 days",

level: "Intermediate",

icon: "☁️",

category: "programming",

rating: 4.8,

students: 2156,

curriculum: [

"AWS Fundamentals",

"EC2 & VPC",

"S3 & Storage",

"RDS & Databases",

"Lambda Functions",

"CloudFormation",

"Security & IAM",

"Monitoring & Logging",

"DevOps Integration",

"Certification Prep"

]

}, {

id: "ui\_ux",

title: "UI/UX Design Professional",

description: "Create stunning user experiences and interfaces",

duration: "10 days",

level: "Beginner",

icon: "🎨",

category: "design",

rating: 4.7,

students: 2834,

curriculum: [

"Design Thinking",

"User Research",

"Wireframing",

"Prototyping",

"Visual Design",

"Interaction Design",

"Usability Testing",

"Design Systems",

"Figma Mastery",

"Portfolio Building"

]

}, {

id: "ai\_ml",

title: "Artificial Intelligence & ML",

description: "Deep dive into AI, neural networks, and machine learning",

duration: "10 days",

level: "Advanced",

icon: "🤖",

category: "data",

rating: 4.9,

students: 1245,

curriculum: [

"AI Fundamentals",

"Machine Learning Algorithms",

"Neural Networks",

"Deep Learning",

"Computer Vision",

"Natural Language Processing",

"TensorFlow & PyTorch",

"Model Deployment",

"AI Ethics",

"Industry Applications"

]

}, {

id: "blockchain",

title: "Blockchain & Web3 Development",

description: "Build decentralized applications and smart contracts",

duration: "10 days",

level: "Advanced",

icon: "⛓️",

category: "programming",

rating: 4.6,

students: 743,

curriculum: [

"Blockchain Fundamentals",

"Ethereum & Solidity",

"Smart Contracts",

"Web3.js Integration",

"DeFi Protocols",

"NFT Development",

"Security Best Practices",

"Testing & Deployment",

"Gas Optimization",

"DApp Project"

]

}, {

id: "agile\_pm",

title: "Agile Project Management",

description: "Master Scrum, Kanban, and modern project leadership",

duration: "10 days",

level: "Beginner",

icon: "📋",

category: "business",

rating: 4.5,

students: 1876,

curriculum: [

"Agile Principles",

"Scrum Framework",

"Kanban Method",

"Sprint Planning",

"Team Leadership",

"Stakeholder Management",

"Risk Assessment",

"Quality Assurance",

"Project Tools",

"PMP Preparation"

]

}, {

id: "python\_prog",

title: "Python Programming Mastery",

description: "Complete Python course from basics to advanced concepts",

duration: "10 days",

level: "Beginner",

icon: "🐍",

category: "programming",

rating: 4.8,

students: 3567,

curriculum: [

"Python Syntax",

"Data Structures",

"Object-Oriented Programming",

"File Handling",

"Error Handling",

"Libraries & Modules",

"Web Scraping",

"API Development",

"Testing",

"Real-world Projects"

]

}, {

id: "graphic\_design",

title: "Graphic Design Fundamentals",

description: "Learn Adobe Creative Suite and design principles",

duration: "10 days",

level: "Beginner",

icon: "🖌️",

category: "design",

rating: 4.6,

students: 2145,

curriculum: [

"Design Principles",

"Color Theory",

"Typography",

"Adobe Photoshop",

"Adobe Illustrator",

"Logo Design",

"Brand Identity",

"Print Design",

"Digital Design",

"Portfolio Creation"

]

}, {

id: "business\_ana",

title: "Business Analytics",

description: "Data-driven decision making for business growth",

duration: "10 days",

level: "Intermediate",

icon: "📊",

category: "business",

rating: 4.7,

students: 1432,

curriculum: [

"Business Intelligence",

"Excel Advanced",

"SQL for Business",

"Tableau Visualization",

"KPI Development",

"Forecasting",

"A/B Testing",

"Customer Analytics",

"Financial Modeling",

"Strategic Planning"

]

}, {

id: "devops",

title: "DevOps Engineering",

description: "Master CI/CD, containerization, and infrastructure automation",

duration: "10 days",

level: "Advanced",

icon: "⚙️",

category: "programming",

rating: 4.8,

students: 987,

curriculum: [

"DevOps Culture",

"Git & Version Control",

"Docker Containers",

"Kubernetes",

"CI/CD Pipelines",

"Infrastructure as Code",

"Monitoring & Logging",

"Security Integration",

"Cloud Platforms",

"Automation"

]

}, {

id: "content\_create",

title: "Content Creation & Video Editing",

description: "Create engaging content for social media and marketing",

duration: "10 days",

level: "Beginner",

icon: "🎬",

category: "design",

rating: 4.5,

students: 1890,

curriculum: [

"Storyboarding",

"Video Shooting Basics",

"Adobe Premiere Pro",

"Color Grading",

"Sound Design",

"Motion Graphics",

"YouTube Optimization",

"Social Media Formats",

"Content Strategy",

"Final Project"

]

}, ];

const firebaseConfig = {

apiKey: "AIzaSyBnQ0s1jFoMH6J7vhKVM7NVC-WNScLdA7E",

authDomain: "sutherland-6ce2b.firebaseapp.com",

databaseURL: "https://sutherland-6ce2b-default-rtdb.firebaseio.com",

projectId: "sutherland-6ce2b",

storageBucket: "sutherland-6ce2b.firebasestorage.app",

messagingSenderId: "243800407888",

appId: "1:243800407888:web:2c4dfa1e6a267ba2a17655",

measurementId: "G-S6GMQ32JJ5"

};

firebase.initializeApp(firebaseConfig);

const auth = firebase.auth();

const database = firebase.database();

// Data Storage

let users = [];

let certificates = [];

let currentUserId = null;

// --- UI Helpers ---

function hideAllScreens() {

['welcome-screen', 'login-form', 'register-form', 'dashboard-screen', 'course-list', 'course-detail', 'course-learning', 'profile-screen', 'certificate-screen', 'certificates-screen', 'admin-dashboard-panel', 'admin-login-screen', 'admin-user-details-modal'].forEach(id => {

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

});

}

function updateNav(isLoggedIn, isAdmin) {

const navButtons = document.getElementById('nav-buttons');

const userNav = document.getElementById('user-nav');

const adminNav = document.getElementById('admin-nav');

navButtons.style.display = 'none';

userNav.style.display = 'none';

adminNav.style.display = 'none';

if (isAdmin) {

adminNav.style.display = 'flex';

} else if (isLoggedIn) {

if (currentUser) {

document.getElementById('user-name-nav').textContent = currentUser.name.split(' ')[0];

document.getElementById('user-avatar').textContent = currentUser.name.charAt(0);

}

userNav.style.display = 'flex';

} else {

navButtons.style.display = 'flex';

}

}

// --- Screen Navigation Functions (Moved to top scope for access) ---

function showWelcomeScreen() {

hideAllScreens();

isAdminLoggedIn = false;

updateNav(false, false);

document.getElementById('welcome-screen').style.display = 'block';

}

function showLogin() { hideAllScreens(); document.getElementById('login-form').style.display = 'block'; }

function showRegister() { hideAllScreens(); document.getElementById('register-form').style.display = 'block'; }

function showCourseList() { hideAllScreens(); document.getElementById('course-list').style.display = 'block'; renderCourseList(courses); resetFilterButtons('all'); }

function showDashboard() { if (!currentUserId) return showLogin(); hideAllScreens(); document.getElementById('dashboard-screen').style.display = 'block'; renderDashboard(); }

function showCertificates() { if (!currentUserId) return showLogin(); hideAllScreens(); document.getElementById('certificates-screen').style.display = 'block'; renderCertificates(); }

function showProfile() { if (!currentUserId) return showLogin(); hideAllScreens(); document.getElementById('profile-screen').style.display = 'block'; renderProfileCourses(); }

function showAdminLogin() { hideAllScreens(); document.getElementById('admin-login-screen').style.display = 'block'; }

function showCertificateScreen(certData) {

hideAllScreens();

document.getElementById('certificate-screen').style.display = 'block';

renderCertificate(certData);

}

async function showAdminDashboard() {

if (!isAdminLoggedIn) return showAdminLogin();

hideAllScreens();

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

await fetchUserData(); // Ensure latest data before rendering

renderAdminDashboard();

}

function startLearning(courseId) {

if (!currentUserId || !currentUser) return showLogin();

const course = courses.find(c => c.id === courseId);

const userEnrollment = currentUser.enrolled\_courses ? currentUser.enrolled\_courses[courseId] : null;

// Check if user is actively enrolled (unlocked: status === true)

if (!userEnrollment || userEnrollment.status !== true) {

alert("Course not yet unlocked. Please check your payment status on the 'My Courses' page. Status must be ACTIVE (True) for access.");

showCourseDetail(courseId);

return;

}

hideAllScreens();

document.getElementById('course-learning').style.display = 'block';

currentCourse = course;

// IMPORTANT: If starting, ensure day is reset to 1 if coming from 100% completion before certification was issued

const currentDayState = userEnrollment.day || 1;

currentDay = currentDayState;

renderCourseContent();

}

// --- Core Application Logic ---

function initializeUI() {

auth.onAuthStateChanged((user) => {

if (user) {

// Check if the user is the hardcoded admin (for mock UI display purposes)

if (user.email === ADMIN\_EMAIL && user.uid === 'admin-mock-uid') {

// Rely on local flag isAdminLoggedIn

} else {

currentUserId = user.uid;

fetchUserData().then(() => {

updateNav(true, false); // Logged in as student

if (document.getElementById('welcome-screen').style.display !== 'none' ||

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

document.getElementById('register-form').style.display !== 'none'

) {

showDashboard();

}

});

}

} else {

currentUserId = null;

currentUser = null;

updateNav(false, isAdminLoggedIn); // Not logged in

if (!isAdminLoggedIn) { // Only redirect if not on Admin screen

if (document.getElementById('dashboard-screen').style.display === 'block' ||

document.getElementById('profile-screen').style.display === 'block' ||

document.getElementById('certificates-screen').style.display === 'block' ||

document.getElementById('course-learning').style.display === 'block'

) {

showWelcomeScreen();

}

}

}

});

loadPopularCourses();

}

async function fetchUserData() {

try {

const userSnapshot = await database.ref('edupro\_users').once('value');

users = userSnapshot.val() ? Object.keys(userSnapshot.val()).map(uid => ({

uid,

...userSnapshot.val()[uid]

})) : [];

currentUser = users.find(u => u.uid === currentUserId);

const certificateSnapshot = await database.ref('certificates').once('value');

certificates = certificateSnapshot.val() ? Object.values(certificateSnapshot.val()) : [];

} catch (error) {

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

users = [];

currentUser = null;

certificates = [];

}

}

// --- Auth & DB Handlers (unchanged) ---

async function handleAdminLogin(event) {

event.preventDefault();

const email = document.getElementById('admin-email').value;

const password = document.getElementById('admin-password').value;

if (email === ADMIN\_EMAIL && password === ADMIN\_PASSWORD) {

isAdminLoggedIn = true;

updateNav(false, true);

showAdminDashboard();

} else {

alert("Invalid Admin Credentials.");

}

}

function adminLogout() {

isAdminLoggedIn = false;

updateNav(false, false);

showWelcomeScreen();

}

async function handleLogin(event) {

event.preventDefault();

const email = document.getElementById('login-email').value;

const password = document.getElementById('login-password').value;

try {

await auth.signInWithEmailAndPassword(email, password);

} catch (error) {

alert(`Login Failed: ${error.message}`);

}

}

async function handleRegister(event) {

event.preventDefault();

const name = document.getElementById('register-name').value;

const email = document.getElementById('register-email').value;

const password = document.getElementById('register-password').value;

try {

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

const uid = userCredential.user.uid;

await database.ref('edupro\_users/' + uid).set({

name: name,

email: email,

joined: new Date().toISOString().split('T')[0],

});

} catch (error) {

alert(`Registration Failed: ${error.message}`);

}

}

function logout() {

auth.signOut();

showWelcomeScreen();

}

async function initiatePayment(courseId) {

if (!currentUserId) return showLogin();

if (!currentUser) await fetchUserData();

const course = courses.find(c => c.id === courseId);

const userEnrollment = currentUser.enrolled\_courses ? currentUser.enrolled\_courses[courseId] : null;

if (userEnrollment && userEnrollment.status === true) {

alert(`You are already enrolled and the ${course.title} course is ACTIVE!`);

return;

}

if (userEnrollment && userEnrollment.status === false) {

// Already pending, just redirect to payment and remind them.

window.open(PAYMENT\_LINK, '\_blank');

alert("Your enrollment is already PENDING confirmation. Redirecting to payment portal again.");

return;

}

try {

// 1. Mark enrollment as PENDING (status: false) with day 1

await database.ref(`edupro\_users/${currentUserId}/enrolled\_courses/${courseId}`).set({

status: false,

day: 1 // Always start at day 1 when first enrolling

});

// 2. Refresh local data and update UI

await fetchUserData();

currentUser = users.find(u => u.uid === currentUserId);

showCourseDetail(courseId); // Update the button/status message

// 3. Redirect user to payment link in a new tab

window.open(PAYMENT\_LINK, '\_blank');

alert(`Payment initiated for ${course.title}. Your enrollment is PENDING confirmation. Please allow time for admin approval.`);

} catch (error) {

alert(`Failed to initiate enrollment: ${error.message}`);

}

}

async function advanceDay(courseId) {

if (!currentUserId || !currentUser) return showLogin();

const course = courses.find(c => c.id === courseId);

const userEnrollment = currentUser.enrolled\_courses ? currentUser.enrolled\_courses[courseId] : null;

// Must be UNLOCKED (status: true)

if (!userEnrollment || userEnrollment.status !== true) {

alert("This course is currently locked. Please contact support if you believe this is an error.");

return;

}

const nextDay = userEnrollment.day + 1;

const isCompleted = nextDay > course.curriculum.length;

try {

if (isCompleted) {

await database.ref(`edupro\_users/${currentUserId}/enrolled\_courses/${courseId}/day`).set(course.curriculum.length);

const certificateId = `${currentUserId}-${courseId}`;

const certData = {

id: certificateId,

userId: currentUserId,

userName: currentUser.name,

courseId: courseId,

courseTitle: course.title,

date: new Date().toISOString().split('T')[0]

};

await database.ref(`certificates/${certificateId}`).set(certData);

await fetchUserData();

currentUser = users.find(u => u.uid === currentUserId);

showCertificateScreen(certData);

alert(`Congratulations! You have completed the ${course.title} course and earned a certificate!`);

} else {

await database.ref(`edupro\_users/${currentUserId}/enrolled\_courses/${courseId}/day`).set(nextDay);

await fetchUserData();

currentUser = users.find(u => u.uid === currentUserId);

currentDay = nextDay;

renderCourseContent();

}

} catch (error) {

console.error("Error advancing day:", error);

alert(`Failed to update progress: ${error.message}`);

}

}

// --- Render Functions (Student) ---

function renderCourseCard(course) {

const userEnrollment = currentUser && currentUser.enrolled\_courses ? currentUser.enrolled\_courses[course.id] : null;

const isEnrolled = userEnrollment && userEnrollment.status === true;

const isPending = userEnrollment && userEnrollment.status === false;

const isNotEnrolled = !userEnrollment;

let cardClasses = 'shadow-lg';

let buttonText = 'View Details';

if (currentUser) {

if (isEnrolled) {

cardClasses = 'border-4 border-green-500 shadow-xl';

buttonText = 'View Course';

} else if (isPending) {

cardClasses = 'border-4 border-orange-500 shadow-xl';

buttonText = 'Awaiting Confirmation';

} else if (isNotEnrolled) {

buttonText = 'Enroll Now';

}

}

return `

<div class="card-hover bg-white p-6 rounded-xl ${cardClasses}">

<div class="text-3xl mb-3">${course.icon}</div>

<h3 class="text-xl font-bold text-gray-800 mb-2">${course.title}</h3>

<p class="text-gray-600 mb-4">${course.description}</p>

<div class="flex justify-between text-sm text-gray-500 mb-4">

<span>Level: ${course.level}</span>

<span>Duration: ${course.duration}</span>

</div>

<div class="flex items-center space-x-1 text-yellow-500 mb-4">

${'★'.repeat(Math.floor(course.rating))}

${course.rating % 1 !== 0 ? '☆' : ''}

<span class="text-gray-600 ml-1">(${course.students} enrolled)</span>

</div>

<button onclick="showCourseDetail('${course.id}')" class="w-full bg-indigo-600 text-white py-2 rounded-lg hover:bg-indigo-700 transition-colors">

${buttonText}

</button>

</div>

`;

}

function loadPopularCourses() {

const container = document.getElementById('popular-courses-preview');

const popularCourses = courses.slice(0, 3);

container.innerHTML = popularCourses.map(renderCourseCard).join('');

}

function renderCourseList(filteredCourses) {

const container = document.getElementById('courses-container');

container.innerHTML = filteredCourses.map(renderCourseCard).join('');

}

function filterCourses(category) {

let filtered = courses;

if (category !== 'all') {

filtered = courses.filter(course => course.category === category);

}

renderCourseList(filtered);

resetFilterButtons(category);

}

function resetFilterButtons(activeCategory) {

document.querySelectorAll('.language-filter').forEach(btn => {

btn.classList.remove('active');

});

const activeBtn = document.querySelector(`.language-filter[onclick="filterCourses('${activeCategory}')"]`);

if (activeBtn) {

activeBtn.classList.add('active');

}

}

async function showCourseDetail(courseId) {

const course = courses.find(c => c.id === courseId);

if (!course) return;

hideAllScreens();

document.getElementById('course-detail').style.display = 'block';

const container = document.getElementById('course-detail-content');

container.setAttribute('data-course-id', courseId);

const userEnrollment = currentUser && currentUser.enrolled\_courses ? currentUser.enrolled\_courses[courseId] : null;

const isEnrolled = userEnrollment && userEnrollment.status === true;

const isPending = userEnrollment && userEnrollment.status === false;

const isNotEnrolled = !currentUser || !userEnrollment;

const currentDay = userEnrollment ? userEnrollment.day : 1;

let actionButton;

let enrollmentMessage;

if (isEnrolled) {

actionButton = `<button onclick="startLearning('${course.id}')" class="w-full bg-green-600 text-white py-3 rounded-lg text-lg font-semibold hover:bg-green-700 transition-colors">

Continue Learning (Day ${currentDay})

</button>`;

enrollmentMessage = `<p class="text-center text-sm text-gray-500 mt-4">You have full, lifetime access to this course.</p>`;

} else if (isPending) {

actionButton = `<div class="p-4 bg-orange-100 border border-orange-400 rounded-lg text-orange-800">

<p class="font-semibold mb-2">Enrollment is \*\*PENDING\*\* admin confirmation.</p>

<p class="text-sm">Please allow time for processing. Your status: <span class="font-bold">LOCKED (FALSE)</span></p>

</div>`;

enrollmentMessage = `<p class="text-center text-sm text-gray-500 mt-4">Check My Courses for status updates. <button onclick="window.open(PAYMENT\_LINK, '\_blank')" class="text-indigo-600 underline">View Payment Portal</button></p>`;

} else if (isNotEnrolled && currentUser) {

actionButton = `<button onclick="initiatePayment('${course.id}')" class="w-full bg-indigo-600 text-white py-3 rounded-lg text-lg font-semibold hover:bg-indigo-700 transition-colors shadow-xl">

Enroll Now - Pay Now

</button>`;

enrollmentMessage = `<p class="text-center text-sm text-gray-500 mt-4">Clicking above initiates enrollment (sets status to FALSE) and opens the payment portal.</p>`;

} else {

actionButton = `<button onclick="showLogin()" class="w-full bg-indigo-600 text-white py-3 rounded-lg text-lg font-semibold hover:bg-indigo-700 transition-colors shadow-xl">

Sign In to Enroll

</button>`;

enrollmentMessage = `<p class="text-center text-sm text-gray-500 mt-4">You must be logged in to enroll in this course.</p>`;

}

const curriculumHTML = course.curriculum.map((item, index) => {

const day = index + 1;

const completed = isEnrolled && day < currentDay;

const current = isEnrolled && day === currentDay;

const statusIcon = completed ? '✅' : current ? '🔥' : '🔒';

const statusColor = completed ? 'text-green-600' : current ? 'text-red-500' : 'text-gray-500';

const isLockedVisual = !isEnrolled || (isEnrolled && day > currentDay);

return `

<li class="p-4 border-b last:border-b-0 flex items-center justify-between ${isLockedVisual ? 'opacity-50' : ''}">

<span class="font-medium flex items-center">Day ${day}: ${item}</span>

<span class="${statusColor}">${statusIcon}</span>

</li>

`;

}).join('');

const html = `

<div class="max-w-7xl mx-auto bg-white rounded-2xl shadow-2xl overflow-hidden p-8">

<div class="md:flex md:space-x-8">

<div class="md:w-2/3">

<h2 class="text-5xl font-extrabold text-gray-900 mb-3">${course.icon} ${course.title}</h2>

<p class="text-xl text-gray-600 mb-6">${course.description}</p>

<div class="flex items-center space-x-6 text-gray-700 mb-8">

<span class="flex items-center">

<svg class="w-5 h-5 mr-1 text-indigo-500" fill="currentColor" viewBox="0 0 20 20"><path d="M10 12a2 2 0 100-4 2 2 0 000 4z"/><path fill-rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-8-5a5 5 0 00-5 5c0 1.05.176 2.05.51 3L10 16l4.49-3A5 5 0 0015 10a5 5 0 00-5-5z" clip-rule="evenodd"/></svg>

${course.level}

</span>

<span class="flex items-center">

<svg class="w-5 h-5 mr-1 text-indigo-500" fill="currentColor" viewBox="0 0 20 20"><path d="M10 18a8 8 0 100-16 8 8 0 000 16zm1-12a1 1 0 10-2 0v4a1 1 0 00.293.707l2.828 2.829a1 1 0 101.415-1.415L11 9.586V6z"/></svg>

${course.duration}

</span>

<span class="flex items-center text-yellow-500">

${'★'.repeat(Math.floor(course.rating))}

<span class="text-gray-600 ml-1">(${course.rating})</span>

</span>

</div>

<h3 class="text-3xl font-bold text-gray-800 mb-4">What you will learn:</h3>

<ul class="grid grid-cols-2 gap-4 text-gray-700">

${course.curriculum.slice(0, 6).map(item => `<li class="flex items-center"><svg class="w-4 h-4 mr-2 text-green-500" fill="currentColor" viewBox="0 0 20 20"><path fill-rule="evenodd" d="M16.707 5.293a1 1 0 010 1.414l-8 8a1 1 0 01-1.414 0l-4-4a1 1 0 011.414-1.414L8 12.586l7.293-7.293a1 1 0 011.414 0z" clip-rule="evenodd"/></svg> ${item}</li>`).join('')}

</ul>

</div>

<div class="md:w-1/3 mt-8 md:mt-0">

<div class="bg-gray-50 p-6 rounded-xl shadow-inner sticky top-24">

<h3 class="text-2xl font-bold text-gray-800 mb-4">Course Access</h3>

<p class="text-3xl font-extrabold text-indigo-600 mb-6">Course Fee: FREE\*</p>

${actionButton}

${enrollmentMessage}

</div>

</div>

</div>

<div class="mt-12 pt-8 border-t border-gray-200">

<h3 class="text-3xl font-bold text-gray-800 mb-6">Full Curriculum (${course.curriculum.length} Days)</h3>

<ul class="bg-white border rounded-xl shadow-lg divide-y divide-gray-100">

${curriculumHTML}

</ul>

</div>

</div>

`;

container.innerHTML = html;

}

function renderCourseContent() {

const container = document.getElementById('course-content');

const progressDisplay = document.getElementById('course-progress');

const totalDays = currentCourse.curriculum.length;

const progress = Math.round((currentDay / totalDays) \* 100);

const content = currentCourse.curriculum[currentDay - 1]; // currentDay is 1-indexed

progressDisplay.textContent = `${progress}% (Day ${currentDay} of ${totalDays})`;

const html = `

<h3 class="text-4xl font-bold text-gray-800 mb-6">${currentCourse.icon} ${currentCourse.title}</h3>

<div class="mb-8">

<div class="text-sm font-medium text-indigo-600 mb-1">Your Progress</div>

<div class="w-full bg-gray-200 rounded-full h-2.5">

<div class="progress-bar bg-green-500 h-2.5 rounded-full" style="width: ${progress}%"></div>

</div>

</div>

<div class="bg-indigo-50 p-6 rounded-xl border-l-4 border-indigo-500 mb-8">

<h4 class="text-2xl font-semibold text-indigo-800 mb-3">Day ${currentDay}: ${content}</h4>

<p class="text-gray-700">This section covers the core concepts of \*\*${content}\*\*. Make sure to follow the recommended practical exercises to fully grasp the material. (Mock Content)</p>

</div>

<div class="space-y-4">

<h5 class="text-xl font-semibold text-gray-800 border-b pb-2">Learning Resources (Mock)</h5>

<div class="flex items-center p-3 bg-white border rounded-lg hover:bg-gray-50 transition-colors">

<span class="text-xl mr-3">📖</span>

<a href="#" class="text-indigo-600 hover:underline">Download: Official E-Book for Day ${currentDay}</a>

</div>

<div class="flex items-center p-3 bg-white border rounded-lg hover:bg-gray-50 transition-colors">

<span class="text-xl mr-3">▶️</span>

<a href="#" class="text-indigo-600 hover:underline">Watch: Video Tutorial for ${content}</a>

</div>

<div class="flex items-center p-3 bg-white border rounded-lg hover:bg-gray-50 transition-colors">

<span class="text-xl mr-3">🧪</span>

<a href="#" class="text-indigo-600 hover:underline">Practice: Interactive Quiz for Day ${currentDay}</a>

</div>

</div>

<div class="mt-10">

<button onclick="advanceDay('${currentCourse.id}')" class="w-full bg-indigo-600 text-white py-3 rounded-lg text-xl font-semibold hover:bg-indigo-700 transition-colors">

${currentDay < totalDays ? `Mark Day ${currentDay} Complete & Go to Next Day →` : 'Complete Course & Get Certificate 🏆'}

</button>

</div>

`;

container.innerHTML = html;

}

function renderDashboard() {

if (!currentUser) return;

document.getElementById('dashboard-user-name').textContent = currentUser.name.split(' ')[0];

const enrolledCourses = currentUser.enrolled\_courses || {};

// Only count UNLOCKED (status: true) and PENDING (status: false) courses for total count

const enrolledCourseIds = Object.keys(enrolledCourses).filter(id => enrolledCourses[id].status === true || enrolledCourses[id].status === false);

document.getElementById('enrolled-count').textContent = enrolledCourseIds.length;

const userCertificates = certificates.filter(c => c.userId === currentUserId);

const completedCount = userCertificates.length;

document.getElementById('completed-count').textContent = completedCount;

document.getElementById('certificates-count').textContent = completedCount;

// Render Continue Learning (Only unlocked courses from enrolled\_courses where status is TRUE)

const continueContainer = document.getElementById('continue-learning');

continueContainer.innerHTML = '';

const activeProgress = Object.keys(enrolledCourses)

.map(id => ({ id, ...enrolledCourses[id] }))

.filter(e => e.status === true && !userCertificates.some(c => c.courseId === e.id)); // Must be unlocked AND not completed

if (activeProgress.length === 0) {

continueContainer.innerHTML = `

<p class="text-gray-500 py-4 text-center">No courses currently in progress. <button onclick="showCourseList()" class="text-indigo-600 hover:underline font-semibold">Start a new course!</button></p>

`;

} else {

activeProgress.slice(0, 3).forEach(p => { // Show up to 3 active courses

const course = courses.find(c => c.id === p.id);

if (course) {

const totalDays = course.curriculum.length;

const progress = Math.round((p.day / totalDays) \* 100);

continueContainer.innerHTML += `

<div class="flex items-center bg-gray-50 p-4 rounded-xl border border-gray-200">

<div class="text-3xl mr-4">${course.icon}</div>

<div class="flex-1">

<h4 class="font-semibold text-gray-800">${course.title}</h4>

<p class="text-sm text-gray-500">Day ${p.day} of ${totalDays} - ${course.curriculum[p.day - 1]}</p>

<div class="mt-1 w-full bg-gray-200 rounded-full h-1.5">

<div class="progress-bar bg-indigo-500 h-1.5 rounded-full" style="width: ${progress}%"></div>

</div>

</div>

<button onclick="startLearning('${course.id}')" class="ml-4 bg-indigo-500 text-white px-4 py-2 rounded-lg text-sm hover:bg-indigo-600 transition-colors">

Continue

</button>

</div>

`;

}

});

}

}

function renderProfileCourses() {

const container = document.getElementById('enrolled-courses');

container.innerHTML = '';

if (!currentUser || !currentUser.enrolled\_courses) {

container.innerHTML = `<p class="text-gray-500 py-8">You are not currently enrolled in any courses. <button onclick="showCourseList()" class="text-indigo-600 hover:underline font-semibold">Browse courses</button> to get started!</p>`;

return;

}

const enrolledCourses = currentUser.enrolled\_courses;

const enrolledCourseIds = Object.keys(enrolledCourses);

const userCertificates = certificates.filter(c => c.userId === currentUserId);

// Filter to only show courses that are explicitly enrolled or pending (status true or false)

const activeEnrollments = enrolledCourseIds.filter(id => enrolledCourses[id].status === true || enrolledCourses[id].status === false);

if (activeEnrollments.length === 0) {

container.innerHTML = `<p class="text-gray-500 py-8">You are not currently enrolled in any courses. <button onclick="showCourseList()" class="text-indigo-600 hover:underline font-semibold">Browse courses</button> to get started!</p>`;

return;

}

activeEnrollments.forEach(courseId => {

const course = courses.find(c => c.id === courseId);

const enrollment = enrolledCourses[courseId];

if (course && enrollment) {

const isEnrolled = enrollment.status === true;

const isPending = enrollment.status === false;

let actionButton;

let statusTag;

let cardClasses = 'border-t-4';

if (isEnrolled) {

const totalDays = course.curriculum.length;

const progress = Math.round((enrollment.day / totalDays) \* 100);

const isComplete = progress === 100;

const hasCertificate = userCertificates.some(c => c.courseId === courseId);

cardClasses += ' border-green-500';

statusTag = `<span class="bg-green-100 text-green-800 text-xs font-medium px-2.5 py-0.5 rounded-full">Active</span>`;

if (isComplete) {

actionButton = hasCertificate ?

`<button onclick="showCertificateScreen(certificates.find(c => c.courseId === '${courseId}' && c.userId === currentUserId))" class="w-full bg-green-600 text-white py-2 rounded-lg hover:bg-green-700">View Certificate 🏆</button>` :

`<button onclick="advanceDay('${courseId}')" class="w-full bg-green-600 text-white py-2 rounded-lg hover:bg-green-700">Finish & Get Certificate 🎓</button>`;

} else {

actionButton = `<button onclick="startLearning('${courseId}')" class="w-full bg-indigo-600 text-white py-2 rounded-lg hover:bg-indigo-700">Continue Learning (Day ${enrollment.day})</button>`;

}

} else if (isPending) {

cardClasses += ' border-orange-500';

statusTag = `<span class="bg-orange-100 text-orange-800 text-xs font-medium px-2.5 py-0.5 rounded-full">Pending Confirmation</span>`;

actionButton = `<button onclick="showCourseDetail('${courseId}')" class="w-full bg-orange-600 text-white py-2 rounded-lg hover:bg-orange-700">View Status</button>`;

}

container.innerHTML += `

<div class="card-hover bg-white p-6 rounded-xl shadow-lg w-full sm:w-80 ${cardClasses}">

<div class="flex justify-between items-center mb-2">

<div class="text-3xl">${course.icon}</div>

${statusTag}

</div>

<h3 class="text-xl font-bold text-gray-800 mb-2">${course.title}</h3>

<p class="text-sm text-gray-600 mb-4">${course.description}</p>

${isEnrolled ? `

<div class="mb-4">

<div class="flex justify-between text-sm font-medium text-gray-700">

<span>Progress:</span>

<span>${Math.round((enrollment.day / course.curriculum.length) \* 100)}%</span>

</div>

<div class="w-full bg-gray-200 rounded-full h-2.5 mt-1">

<div class="progress-bar bg-green-500 h-2.5 rounded-full" style="width: ${Math.round((enrollment.day / course.curriculum.length) \* 100)}%"></div>

</div>

</div>

` : `<div class="h-10 mb-4"></div>`}

${actionButton}

</div>

`;

}

});

}

function renderCertificates() {

const container = document.getElementById('certificates-container');

container.innerHTML = '';

const userCertificates = certificates.filter(c => c.userId === currentUserId);

if (userCertificates.length === 0) {

container.innerHTML = `<p class="text-gray-500 py-8 text-center w-full">You have not completed any courses yet. Keep learning!</p>`;

return;

}

userCertificates.forEach(cert => {

const course = courses.find(c => c.id === cert.courseId);

if (course) {

container.innerHTML += `

<div class="card-hover bg-white p-6 rounded-xl shadow-lg border-t-4 border-yellow-500 text-center">

<div class="text-5xl mb-3">🎓</div>

<h3 class="text-xl font-bold text-gray-800 mb-1">${course.title}</h3>

<p class="text-gray-600 mb-3">Certificate of Completion</p>

<p class="text-sm text-gray-500 mb-4">Awarded on: ${cert.date}</p>

<button onclick="showCertificateScreen(certificates.find(c => c.id === '${cert.id}'))" class="w-full bg-yellow-500 text-gray-900 py-2 rounded-lg hover:bg-yellow-600 transition-colors font-semibold">

View Certificate

</button>

</div>

`;

}

});

}

function renderCertificate(certData) {

if (!certData) return;

const course = courses.find(c => c.id === certData.courseId);

if (!course) return;

const container = document.getElementById('certificate-content');

container.innerHTML = `

<div class="certificate-bg p-12 rounded-lg text-center">

<h2 class="text-4xl font-serif font-light mb-4 text-yellow-300">Certificate of Achievement</h2>

<p class="text-xl font-light mb-2">This certifies that</p>

<h1 class="text-6xl font-extrabold mb-6 text-white certificate-name">${certData.userName}</h1>

<p class="text-xl font-light mb-4">has successfully completed the course</p>

<h2 class="text-3xl font-bold mb-8 text-yellow-300">${course.title}</h2>

<div class="flex justify-around items-center text-lg mt-8">

<div>

<p class="font-bold text-white">${certData.date}</p>

<p class="text-sm text-yellow-300">Date Issued</p>

</div>

<div>

<p class="font-bold text-white">EduPro Academy</p>

<p class="text-sm text-yellow-300">Authority</p>

</div>

</div>

</div>

`;

container.setAttribute('data-course-title', course.title);

container.setAttribute('data-user-name', certData.userName);

}

function downloadCertificate() {

const {

jsPDF

} = window.jspdf;

const doc = new jsPDF({

orientation: 'landscape',

unit: 'in',

format: 'letter'

});

const courseTitle = document.getElementById('certificate-content').getAttribute('data-course-title');

const userName = document.getElementById('certificate-content').getAttribute('data-user-name');

const certData = certificates.find(c => c.userName === userName && c.courseTitle === courseTitle);

const date = certData ? certData.date : new Date().toISOString().split('T')[0];

const pageWidth = doc.internal.pageSize.width;

const pageHeight = doc.internal.pageSize.height;

doc.setFillColor(30, 60, 114);

doc.rect(0, 0, pageWidth, pageHeight, 'F');

doc.setDrawColor(255, 215, 0);

doc.setLineWidth(0.2);

doc.rect(0.2, 0.2, pageWidth - 0.4, pageHeight - 0.4, 'S');

doc.rect(0.4, 0.4, pageWidth - 0.8, pageHeight - 0.8, 'S');

doc.setFont('times', 'bold');

doc.setFontSize(36);

doc.setTextColor(255, 215, 0);

doc.text("CERTIFICATE OF ACHIEVEMENT", pageWidth / 2, 1.5, {

align: 'center'

});

doc.setFontSize(14);

doc.setTextColor(255, 255, 255);

doc.setFont('times', 'normal');

doc.text("This certifies that", pageWidth / 2, 2.2, {

align: 'center'

});

doc.setFontSize(50);

doc.setTextColor(255, 255, 255);

doc.setFont('cursive');

doc.text(userName, pageWidth / 2, 3.5, {

align: 'center'

});

doc.setFontSize(18);

doc.setTextColor(255, 255, 255);

doc.setFont('times', 'normal');

doc.text("has successfully completed the course", pageWidth / 2, 4.3, {

align: 'center'

});

doc.setFontSize(28);

doc.setTextColor(255, 215, 0);

doc.setFont('times', 'bold');

doc.text(courseTitle, pageWidth / 2, 5.0, {

align: 'center'

});

doc.setFontSize(12);

doc.setTextColor(255, 255, 255);

doc.setFont('times', 'normal');

doc.line(1.5, 6.5, 3.5, 6.5);

doc.text("Director Signature (Mock)", 2.5, 6.8, {

align: 'center'

});

doc.line(8.0, 6.5, 10.0, 6.5);

doc.text(`Date: ${date}`, 9.0, 6.8, {

align: 'center'

});

doc.save(`${userName.replace(/ /g, '\_')}\_${courseTitle.replace(/ /g, '\_')}\_Certificate.pdf`);

}

// --- ADMIN LOGIC ---

function getCourseTitle(courseId) {

const course = courses.find(c => c.id === courseId);

return course ? course.title : 'Unknown Course';

}

function getCourseByTitle(title) {

return courses.find(c => c.title === title);

}

async function toggleCourseStatus(userId, courseId, currentStatus) {

if (!isAdminLoggedIn) {

alert("Authorization failed. Please log in as Admin.");

return;

}

const newStatus = currentStatus === 'false' ? true : false;

try {

// Update the status node in Firebase

await database.ref(`edupro\_users/${userId}/enrolled\_courses/${courseId}/status`).set(newStatus);

// Fetch the updated data and re-render the table

await fetchUserData();

renderAdminDashboard(); // Re-render everything to update graphs/metrics/table

alert(`Status for ${getCourseTitle(courseId)} for user ${users.find(u => u.uid === userId).name} updated to ${newStatus ? 'TRUE (ACTIVE)' : 'FALSE (PENDING)'}`);

} catch (error) {

console.error("Error toggling course status:", error);

alert(`Failed to update status: ${error.message}`);

}

}

// NEW: Function to render the Admin Dashboard User Modal

function renderCourseProgressCard(courseData, userCertificates) {

const course = getCourseByTitle(courseData.title);

if (!course) return '';

const totalDays = course.curriculum.length;

const progressDay = courseData.day || 1;

const progress = Math.round((progressDay / totalDays) \* 100);

const isCertified = userCertificates.some(c => c.courseTitle === courseData.title);

let statusTag;

if (isCertified) {

statusTag = `<span class="bg-green-500 text-white text-xs font-medium px-2 py-1 rounded-full">Certified</span>`;

} else if (courseData.status === true) {

statusTag = `<span class="bg-yellow-500 text-gray-800 text-xs font-medium px-2 py-1 rounded-full">In Progress</span>`;

} else {

statusTag = `<span class="bg-orange-500 text-white text-xs font-medium px-2 py-1 rounded-full">Pending</span>`;

}

let viewButton = '';

if (isCertified) {

const cert = userCertificates.find(c => c.courseTitle === courseData.title);

viewButton = `<button onclick="alert('Simulating certificate view for ${cert.date}')" class="bg-indigo-600 text-white text-xs px-3 py-1 rounded-full ml-3">View Certificate</button>`;

}

return `

<div class="p-4 border rounded-lg shadow-sm">

<div class="flex justify-between items-center mb-2">

<h4 class="font-bold text-gray-900">${courseData.title}</h4>

<div class="flex items-center">

${statusTag}

${viewButton}

</div>

</div>

<p class="text-sm text-gray-500">Progress: ${progress}% (Day ${progressDay} of ${totalDays})</p>

<div class="w-full bg-gray-200 rounded-full h-2 mt-1">

<div class="progress-bar bg-green-500 h-2 rounded-full" style="width: ${progress}%"></div>

</div>

<p class="text-xs mt-2 text-gray-500">Certification Status: ${isCertified ? 'Certified' : courseData.status === false ? 'Pending Approval' : 'Pending Completion'}</p>

</div>

`;

}

// NEW: Function to render the Admin Dashboard User Modal

function showAdminUserDetails(userId) {

const user = users.find(u => u.uid === userId);

if (!user) return;

const modal = document.getElementById('admin-user-details-modal');

document.getElementById('modal-user-name').textContent = `${user.name}'s Course Details`;

// 1. User Info

const userInfoContainer = document.getElementById('modal-user-info');

userInfoContainer.innerHTML = `

<div class="grid grid-cols-2 gap-y-1">

<p><strong>Email:</strong> ${user.email}</p>

<p><strong>Joined:</strong> ${user.joined}</p>

</div>

`;

// 2. Course List

const courseListContainer = document.getElementById('modal-course-list');

courseListContainer.innerHTML = '';

const userEnrollments = user.enrolled\_courses || {};

const userCertificates = certificates.filter(c => c.userId === userId);

const enrolledCount = Object.keys(userEnrollments).length;

document.getElementById('modal-enrolled-count').textContent = enrolledCount;

if (enrolledCount === 0) {

courseListContainer.innerHTML = `<p class="text-gray-500 text-center py-4">This student has no enrollments yet.</p>`;

} else {

Object.keys(userEnrollments).forEach(courseId => {

const course = courses.find(c => c.id === courseId);

if (course) {

const courseData = {

title: course.title,

day: userEnrollments[courseId].day,

status: userEnrollments[courseId].status

};

courseListContainer.innerHTML += renderCourseProgressCard(courseData, userCertificates);

}

});

}

modal.classList.remove('hidden');

modal.style.display = 'flex';

}

function closeAdminUserDetails() {

document.getElementById('admin-user-details-modal').classList.add('hidden');

document.getElementById('admin-user-details-modal').style.display = 'none';

}

// NEW: Function to calculate enrollment data for the chart

function getCourseEnrollmentDistributionData() {

const enrollmentMap = {}; // { courseId: { enrolled: count, certified: count } }

courses.forEach(course => {

enrollmentMap[course.id] = { enrolled: 0, certified: 0, title: course.title };

});

users.forEach(user => {

const userEnrollments = user.enrolled\_courses || {};

Object.keys(userEnrollments).forEach(courseId => {

if (enrollmentMap[courseId]) {

// Count all courses the user has explicitly enrolled in (true or false)

enrollmentMap[courseId].enrolled++;

}

});

});

certificates.forEach(cert => {

if (enrollmentMap[cert.courseId]) {

enrollmentMap[cert.courseId].certified++;

}

});

return {

labels: Object.values(enrollmentMap).map(d => d.title),

enrolledData: Object.values(enrollmentMap).map(d => d.enrolled),

certifiedData: Object.values(enrollmentMap).map(d => d.certified)

};

}

function renderAdminDashboard() {

if (!isAdminLoggedIn) return;

// --- METRICS UPDATE ---

document.getElementById('metric-total-students').textContent = users.length;

document.getElementById('metric-total-certificates').textContent = certificates.length;

document.getElementById('last-updated').textContent = new Date().toLocaleTimeString();

// Mock active users for display consistency

document.getElementById('metric-total-active-users').textContent = Math.floor(Math.random() \* users.length \* 0.4);

// --- GRAPH UPDATE ---

const distributionData = getCourseEnrollmentDistributionData();

const ctx = document.getElementById('courseChart').getContext('2d');

if (courseChartInstance) {

courseChartInstance.destroy();

}

courseChartInstance = new Chart(ctx, {

type: 'bar',

data: {

labels: distributionData.labels,

datasets: [{

label: 'Total Enrolled (Active or Pending)',

data: distributionData.enrolledData,

backgroundColor: 'rgba(59, 130, 246, 0.7)', // Blue

borderColor: 'rgba(59, 130, 246, 1)',

borderWidth: 1

}, {

label: 'Certified (Completed)',

data: distributionData.certifiedData,

backgroundColor: 'rgba(16, 185, 129, 0.7)', // Green

borderColor: 'rgba(16, 185, 129, 1)',

borderWidth: 1

}]

},

options: {

responsive: true,

maintainAspectRatio: false,

scales: {

y: {

beginAtZero: true,

title: {

display: true,

text: 'Number of Students'

}

},

x: {

// Corrected x-axis label issue if any courses were undefined

}

}

}

});

// --- FILTER SETUP & TABLE RENDER ---

// Populate Course Filter

const courseFilter = document.getElementById('admin-course-filter');

courseFilter.innerHTML = '<option value="all">All Courses</option>';

courses.forEach(c => {

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

option.value = c.id;

option.textContent = c.title;

courseFilter.appendChild(option);

});

renderAdminStudentTable();

// Setup listeners for filtering (already set up in previous versions, ensuring they call renderAdminStudentTable)

document.getElementById('admin-student-search').oninput = renderAdminStudentTable;

document.getElementById('admin-course-filter').onchange = renderAdminStudentTable;

document.getElementById('admin-status-filter').onchange = renderAdminStudentTable;

}

function renderAdminStudentTable() {

const tableBody = document.getElementById('admin-table-body');

const noResults = document.getElementById('admin-no-results');

tableBody.innerHTML = '';

const searchTerm = document.getElementById('admin-student-search').value.toLowerCase();

const filterCourseId = document.getElementById('admin-course-filter').value;

const filterStatus = document.getElementById('admin-status-filter').value;

// Flatten data: one row per user-course enrollment

const enrollmentRecords = [];

users.forEach(user => {

const userEnrollments = user.enrolled\_courses || {};

Object.keys(userEnrollments).forEach(courseId => {

enrollmentRecords.push({

userId: user.uid,

userName: user.name,

userEmail: user.email,

courseId: courseId,

courseTitle: getCourseTitle(courseId),

status: userEnrollments[courseId].status // boolean true/false

});

});

});

const filteredRecords = enrollmentRecords.filter(record => {

// 1. Name/Email Search

if (searchTerm && !record.userName.toLowerCase().includes(searchTerm) && !record.userEmail.toLowerCase().includes(searchTerm)) {

return false;

}

// 2. Course Filter

if (filterCourseId !== 'all' && record.courseId !== filterCourseId) {

return false;

}

// 3. Status Filter

if (filterStatus === 'true' && record.status !== true) {

return false;

}

if (filterStatus === 'false' && record.status !== false) {

return false;

}

return true;

});

if (filteredRecords.length === 0) {

noResults.style.display = 'block';

} else {

noResults.style.display = 'none';

filteredRecords.forEach(record => {

const statusText = record.status ? 'ACTIVE (TRUE)' : 'PENDING (FALSE)';

const statusClass = record.status ? 'bg-green-100 text-green-800' : 'bg-orange-100 text-orange-800';

// Only show the APPROVE button if the status is FALSE

let actionButton;

if (!record.status) {

actionButton = `<button onclick="toggleCourseStatus('${record.userId}', '${record.courseId}', 'false')"

class="bg-blue-600 text-white px-4 py-2 rounded-lg text-xs hover:bg-blue-700 transition-colors">

APPROVE

</button>`;

} else {

// Option to Revoke/Unapprove (sets status back to false)

actionButton = `<button onclick="toggleCourseStatus('${record.userId}', '${record.courseId}', 'true')"

class="bg-gray-400 text-white px-4 py-2 rounded-lg text-xs hover:bg-gray-500 transition-colors">

Revoke

</button>`;

}

const row = `

<tr>

<td class="px-6 py-4 whitespace-nowrap text-sm font-medium text-gray-900">${record.userName}</td>

<td class="px-6 py-4 whitespace-nowrap text-sm text-gray-500">${record.userEmail}</td>

<td class="px-6 py-4 whitespace-nowrap text-sm font-medium text-indigo-600">${record.courseTitle}</td>

<td class="px-6 py-4 whitespace-nowrap text-sm">

<span class="px-2 inline-flex text-xs leading-5 font-semibold rounded-full ${statusClass}">

${statusText}

</span>

</td>

<td class="px-6 py-4 whitespace-nowrap text-sm flex space-x-2">

${actionButton}

<button onclick="showAdminUserDetails('${record.userId}')"

class="bg-purple-600 text-white px-4 py-2 rounded-lg text-xs hover:bg-purple-700 transition-colors">

Profile

</button>

</td>

</tr>

`;

tableBody.innerHTML += row;

});

}

}

// Global initialization call

initializeUI();

</script>

<script>

// Security scripts (unchanged)

document.addEventListener('contextmenu', function(event) { event.preventDefault(); return false; });

document.addEventListener('keydown', function(event) {

if (event.key === 'F12' || event.keyCode === 123) { event.preventDefault(); return false; }

if (event.ctrlKey || event.metaKey) {

if (event.shiftKey && (event.key === 'I' || event.key === 'i')) { event.preventDefault(); return false; }

if (event.key === 'u' || event.key === 'U') { event.preventDefault(); return false; }

}

});

</script>

</body>

</html>

**ALL FUNCATIONS ARE WORKING**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8">**

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

**<title>EduPro Academy - Professional Learning Platform</title>**

**<script src="https://cdn.tailwindcss.com"></script>**

**<script src="https://cdnjs.cloudflare.com/ajax/libs/jspdf/2.5.1/jspdf.umd.min.js"></script>**

**<script src="https://www.gstatic.com/firebasejs/9.23.0/firebase-app-compat.js"></script>**

**<script src="https://www.gstatic.com/firebasejs/9.23.0/firebase-auth-compat.js"></script>**

**<script src="https://www.gstatic.com/firebasejs/9.23.0/firebase-database-compat.js"></script>**

**<!-- Chart.js for Graph Visualization -->**

**<script src="https://cdn.jsdelivr.net/npm/chart.js@4.4.2/dist/chart.umd.min.js"></script>**

**<style>**

**body {**

**box-sizing: border-box;**

**}**

**.fade-in {**

**animation: fadeIn 0.8s ease-out;**

**}**

**@keyframes fadeIn {**

**from {**

**opacity: 0;**

**transform: translateY(30px);**

**}**

**to {**

**opacity: 1;**

**transform: translateY(0);**

**}**

**}**

**.slide-in {**

**animation: slideIn 0.6s ease-out;**

**}**

**@keyframes slideIn {**

**from {**

**opacity: 0;**

**transform: translateX(-30px);**

**}**

**to {**

**opacity: 1;**

**transform: translateX(0);**

**}**

**}**

**.hero-gradient {**

**background: linear-gradient(135deg, #667eea 0%, #764ba2 50%, #f093fb 100%);**

**}**

**.card-hover {**

**transition: all 0.4s cubic-bezier(0.4, 0, 0.2, 1);**

**}**

**.card-hover:hover {**

**transform: translateY(-8px) scale(1.02);**

**box-shadow: 0 25px 50px -12px rgba(0, 0, 0, 0.25);**

**}**

**.progress-bar {**

**transition: width 0.6s cubic-bezier(0.4, 0, 0.2, 1);**

**}**

**.locked {**

**opacity: 0.5;**

**pointer-events: none;**

**}**

**.certificate-bg {**

**background: linear-gradient(135deg, #1e3c72 0%, #2a5298 50%, #667eea 100%);**

**border: 12px solid #ffd700;**

**box-shadow: 0 0 40px rgba(0, 0, 0, 0.4);**

**}**

**.nav-glass {**

**backdrop-filter: blur(10px);**

**background: rgba(255, 255, 255, 0.95);**

**}**

**.stats-counter {**

**animation: countUp 2s ease-out;**

**}**

**@keyframes countUp {**

**from {**

**opacity: 0;**

**transform: scale(0.5);**

**}**

**to {**

**opacity: 1;**

**transform: scale(1);**

**}**

**}**

**.floating {**

**animation: floating 3s ease-in-out infinite;**

**}**

**@keyframes floating {**

**0%,**

**100% {**

**transform: translateY(0px);**

**}**

**50% {**

**transform: translateY(-10px);**

**}**

**}**

**.pulse-glow {**

**animation: pulseGlow 2s infinite;**

**}**

**@keyframes pulseGlow {**

**0%,**

**100% {**

**box-shadow: 0 0 20px rgba(102, 126, 234, 0.4);**

**}**

**50% {**

**box-shadow: 0 0 30px rgba(102, 126, 234, 0.8);**

**}**

**}**

**.language-filter.active {**

**background-color: #4f46e5;**

**color: white;**

**}**

**.language-filter {**

**color: #6b7280;**

**}**

**/\* Modal Styles \*/**

**.modal {**

**position: fixed;**

**top: 0;**

**left: 0;**

**width: 100%;**

**height: 100%;**

**background-color: rgba(0, 0, 0, 0.5);**

**display: flex;**

**justify-content: center;**

**align-items: center;**

**z-index: 1000;**

**}**

**.modal-content {**

**background-color: white;**

**border-radius: 1rem;**

**padding: 2rem;**

**max-width: 90%;**

**max-height: 90vh;**

**overflow-y: auto;**

**position: relative;**

**}**

**</style>**

**</head>**

**<body class="bg-gradient-to-br from-blue-50 to-indigo-100 min-h-screen">**

**<nav class="nav-glass shadow-xl fixed w-full top-0 z-50">**

**<div class="max-w-7xl mx-auto px-6">**

**<div class="flex justify-between items-center py-4">**

**<div class="flex items-center space-x-3">**

**<div**

**class="w-10 h-10 bg-gradient-to-r from-indigo-600 to-purple-600 rounded-lg flex items-center justify-center">**

**<span class="text-white font-bold text-lg">E</span>**

**</div>**

**<h1**

**class="text-2xl font-bold bg-gradient-to-r from-indigo-600 to-purple-600 bg-clip-text text-transparent">**

**EduPro Academy</h1>**

**</div>**

**<div id="nav-buttons" class="flex items-center space-x-4">**

**<button onclick="showAdminLogin()"**

**class="bg-yellow-500 text-gray-900 px-4 py-2.5 rounded-full hover:shadow-lg transition-all duration-300 font-medium">Admin</button>**

**<button onclick="showLogin()"**

**class="bg-gradient-to-r from-indigo-600 to-purple-600 text-white px-6 py-2.5 rounded-full hover:shadow-lg transition-all duration-300 font-medium">Sign**

**In</button>**

**<button onclick="showRegister()"**

**class="bg-gradient-to-r from-green-500 to-emerald-600 text-white px-6 py-2.5 rounded-full hover:shadow-lg transition-all duration-300 font-medium">Get**

**Started</button>**

**</div>**

**<div id="user-nav" class="hidden flex items-center space-x-2">**

**<div class="flex items-center space-x-1 bg-white rounded-full px-4 py-2 shadow-md">**

**<div**

**class="w-8 h-8 bg-gradient-to-r from-indigo-500 to-purple-500 rounded-full flex items-center justify-center">**

**<span class="text-white text-sm font-semibold" id="user-avatar"></span>**

**</div>**

**<span class="text-gray-700 font-medium" id="user-name-nav"></span>**

**</div>**

**<button onclick="showDashboard()"**

**class="bg-indigo-600 text-white px-4 py-2 rounded-lg hover:bg-indigo-700 transition-colors">Dashboard</button>**

**<button onclick="showProfile()"**

**class="bg-purple-600 text-white px-4 py-2 rounded-lg hover:bg-purple-700 transition-colors">My**

**Courses</button>**

**<button onclick="showCertificates()"**

**class="bg-green-600 text-white px-4 py-2 rounded-lg hover:bg-green-700 transition-colors">Certificates</button>**

**<button onclick="logout()"**

**class="bg-red-500 text-white px-4 py-2 rounded-lg hover:bg-red-600 transition-colors">Logout</button>**

**</div>**

**<div id="admin-nav" class="hidden items-center space-x-2">**

**<span class="text-sm font-bold text-red-600">Admin Mode</span>**

**<button onclick="showAdminDashboard()"**

**class="bg-blue-600 text-white px-4 py-2 rounded-lg hover:bg-blue-700 transition-colors">Admin**

**Dashboard</button>**

**<button onclick="adminLogout()"**

**class="bg-red-500 text-white px-4 py-2 rounded-lg hover:bg-red-600 transition-colors">Logout</button>**

**</div>**

**</div>**

**</div>**

**</nav>**

**<div class="pt-20">**

**<div id="welcome-screen" class="fade-in">**

**<section class="hero-gradient text-white py-20 px-4 sm:px-6">**

**<div class="max-w-7xl mx-auto text-center">**

**<div class="floating mb-8">**

**<div class="inline-block p-4 bg-white bg-opacity-20 rounded-full">**

**<svg class="w-16 h-16" fill="currentColor" viewBox="0 0 20 20">**

**<path**

**d="M10.394 2.08a1 1 0 00-.788 0l-7 3a1 1 0 000 1.84L5.25 8.051a.999.999 0 01.356-.257l4-1.714a1 1 0 11.788 1.838L7.667 9.088l1.94.831a1 1 0 00.787 0l7-3a1 1 0 000-1.838l-7-3zM3.31 9.397L5 10.12v4.102a8.969 8.969 0 00-1.05-.174 1 1 0 01-.89-.89 11.115 11.115 0 01.25-3.762zM9.3 16.573A9.026 9.026 0 007 14.935v-3.957l1.818.78a3 3 0 002.364 0l5.508-2.361a11.026 11.026 0 01.25 3.762 1 1 0 01-.89.89 8.968 8.968 0 00-5.35 2.524 1 1 0 01-1.4 0zM6 18a1 1 0 001-1v-2.065a8.935 8.935 0 00-2-.712V17a1 1 0 001 1z" />**

**</svg>**

**</div>**

**</div>**

**<h1 class="text-5xl md:text-7xl font-bold mb-6 slide-in">**

**Transform Your**

**<span**

**class="bg-gradient-to-r from-yellow-300 to-orange-300 bg-clip-text text-transparent">Career</span>**

**</h1>**

**<p class="text-xl md:text-2xl mb-8 text-white text-opacity-90 max-w-3xl mx-auto slide-in">**

**Join thousands of professionals mastering cutting-edge skills with our expert-designed courses.**

**Learn at your own pace, earn certificates, and advance your career.**

**</p>**

**<div class="flex flex-col sm:flex-row gap-4 justify-center mb-12 slide-in">**

**<button onclick="showRegister()"**

**class="pulse-glow bg-white text-indigo-600 px-8 py-4 rounded-full text-lg font-semibold hover:bg-gray-100 transition-all duration-300">**

**Start Learning Today**

**</button>**

**<button onclick="showCourseList()"**

**class="border-2 border-white text-white px-8 py-4 rounded-full text-lg font-semibold hover:bg-white hover:text-indigo-600 transition-all duration-300">**

**Explore Courses**

**</button>**

**</div>**

**</div>**

**</section>**

**<section class="py-16 bg-white">**

**<div class="max-w-7xl mx-auto px-4 sm:px-6">**

**<div class="grid grid-cols-2 md:grid-cols-4 gap-4 md:gap-8 text-center">**

**<div class="stats-counter">**

**<div class="text-4xl font-bold text-indigo-600 mb-2">15+</div>**

**<div class="text-gray-600 font-medium">Expert Courses</div>**

**</div>**

**<div class="stats-counter">**

**<div class="text-4xl font-bold text-purple-600 mb-2">5K+</div>**

**<div class="text-gray-600 font-medium">Students</div>**

**</div>**

**<div class="stats-counter">**

**<div class="text-4xl font-bold text-green-600 mb-2">98%</div>**

**<div class="text-gray-600 font-medium">Success Rate</div>**

**</div>**

**<div class="stats-counter">**

**<div class="text-4xl font-bold text-orange-600 mb-2">24/7</div>**

**<div class="text-gray-600 font-medium">Support</div>**

**</div>**

**</div>**

**</div>**

**</section>**

**<section class="py-20 bg-gray-50">**

**<div class="max-w-7xl mx-auto px-4 sm:px-6">**

**<div class="text-center mb-16">**

**<h2 class="text-4xl font-bold text-gray-800 mb-4">Why Choose EduPro Academy?</h2>**

**<p class="text-xl text-gray-600 max-w-2xl mx-auto">Experience the future of online learning with**

**our innovative platform</p>**

**</div>**

**<div class="grid md:grid-cols-3 gap-4 md:gap-8">**

**<div class="card-hover bg-white p-8 rounded-2xl shadow-lg text-center">**

**<div**

**class="w-16 h-16 bg-gradient-to-r from-indigo-500 to-purple-500 rounded-full flex items-center justify-center mx-auto mb-6">**

**<svg class="w-8 h-8 text-white" fill="currentColor" viewBox="0 0 20 20">**

**<path d="M9 12l2 2 4-4m6 2a9 9 0 11-18 0 9 9 0 0118 0z" />**

**</svg>**

**</div>**

**<h3 class="text-2xl font-semibold mb-4">Industry-Recognized Certificates</h3>**

**<p class="text-gray-600">Earn certificates that employers value and trust</p>**

**</div>**

**<div class="card-hover bg-white p-8 rounded-2xl shadow-lg text-center">**

**<div**

**class="w-16 h-16 bg-gradient-to-r from-green-500 to-emerald-500 rounded-full flex items-center justify-center mx-auto mb-6">**

**<svg class="w-8 h-8 text-white" fill="currentColor" viewBox="0 0 20 20">**

**<path**

**d="M13 6a3 3 0 11-6 0 3 3 0 016 0zM18 8a2 2 0 11-4 0 2 2 0 014 0zM14 15a4 4 0 00-8 0v3h8v-3z" />**

**</svg>**

**</div>**

**<h3 class="text-2xl font-semibold mb-4">Expert Instructors</h3>**

**<p class="text-gray-600">Learn from industry professionals with real-world experience</p>**

**</div>**

**<div class="card-hover bg-white p-8 rounded-2xl shadow-lg text-center">**

**<div**

**class="w-16 h-16 bg-gradient-to-r from-orange-500 to-red-500 rounded-full flex items-center justify-center mx-auto mb-6">**

**<svg class="w-8 h-8 text-white" fill="currentColor" viewBox="0 0 20 20">**

**<path**

**d="M3 4a1 1 0 011-1h12a1 1 0 011 1v2a1 1 0 01-1 1H4a1 1 0 01-1-1V4zM3 10a1 1 0 011-1h6a1 1 0 011 1v6a1 1 0 01-1 1H4a1 1 0 01-1-1v-6zM14 9a1 1 0 00-1 1v6a1 1 0 001 1h2a1 1 0 001-1v-6a1 1 0 00-1-1h-2z" />**

**</svg>**

**</div>**

**<h3 class="text-2xl font-semibold mb-4">Progress Tracking</h3>**

**<p class="text-gray-600">Monitor your learning journey with detailed analytics</p>**

**</div>**

**</div>**

**</div>**

**</section>**

**<section class="py-20 bg-white">**

**<div class="max-w-7xl mx-auto px-4 sm:px-6">**

**<div class="text-center mb-16">**

**<h2 class="text-4xl font-bold text-gray-800 mb-4">Popular Courses</h2>**

**<p class="text-xl text-gray-600">Start your learning journey with our most popular courses</p>**

**</div>**

**<div class="grid md:grid-cols-3 gap-4 md:gap-8" id="popular-courses-preview">**

**</div>**

**<div class="text-center mt-12">**

**<button onclick="showCourseList()"**

**class="bg-gradient-to-r from-indigo-600 to-purple-600 text-white px-8 py-4 rounded-full text-lg font-semibold hover:shadow-lg transition-all duration-300">**

**View All Courses**

**</button>**

**</div>**

**</div>**

**</section>**

**</div>**

**<!-- NEW: Admin Login Screen -->**

**<div id="admin-login-screen" class="hidden max-w-md mx-auto bg-white p-8 rounded-lg shadow-lg fade-in">**

**<h2 class="text-2xl font-bold text-center mb-6 text-red-600">Admin Login</h2>**

**<p class="text-center text-sm text-gray-600 mb-4">Use: admin@edupro.com / admin123</p>**

**<form onsubmit="handleAdminLogin(event)">**

**<div class="mb-4">**

**<label class="block text-gray-700 text-sm font-bold mb-2">Admin Email</label>**

**<input type="email" id="admin-email"**

**class="w-full px-3 py-2 border rounded-lg focus:outline-none focus:border-red-500"**

**value="admin@edupro.com" required>**

**</div>**

**<div class="mb-6">**

**<label class="block text-gray-700 text-sm font-bold mb-2">Admin Password</label>**

**<input type="password" id="admin-password"**

**class="w-full px-3 py-2 border rounded-lg focus:outline-none focus:border-red-500"**

**value="admin123" required>**

**</div>**

**<button type="submit" class="w-full bg-red-600 text-white py-2 rounded-lg hover:bg-red-700">Login as**

**Admin</button>**

**</form>**

**<p class="text-center mt-4">**

**<button onclick="showWelcomeScreen()" class="text-gray-600 hover:underline">← Back to Main Page</button>**

**</p>**

**</div>**

**<!-- Existing User Login Screen -->**

**<div id="login-form" class="hidden max-w-md mx-auto bg-white p-8 rounded-lg shadow-lg fade-in">**

**<h2 class="text-2xl font-bold text-center mb-6">Login</h2>**

**<form onsubmit="handleLogin(event)">**

**<div class="mb-4">**

**<label class="block text-gray-700 text-sm font-bold mb-2">Email</label>**

**<input type="email" id="login-email"**

**class="w-full px-3 py-2 border rounded-lg focus:outline-none focus:border-indigo-500" required>**

**</div>**

**<div class="mb-6">**

**<label class="block text-gray-700 text-sm font-bold mb-2">Password</label>**

**<input type="password" id="login-password"**

**class="w-full px-3 py-2 border rounded-lg focus:outline-none focus:border-indigo-500" required>**

**</div>**

**<button type="submit"**

**class="w-full bg-indigo-600 text-white py-2 rounded-lg hover:bg-indigo-700">Login</button>**

**</form>**

**<p class="text-center mt-4">**

**Don't have an account?**

**<button onclick="showRegister()" class="text-indigo-600 hover:underline">Register here</button>**

**</p>**

**</div>**

**<div id="register-form" class="hidden max-w-md mx-auto bg-white p-8 rounded-lg shadow-lg fade-in">**

**<h2 class="text-2xl font-bold text-center mb-6">Register</h2>**

**<form onsubmit="handleRegister(event)">**

**<div class="mb-4">**

**<label class="block text-gray-700 text-sm font-bold mb-2">Full Name</label>**

**<input type="text" id="register-name"**

**class="w-full px-3 py-2 border rounded-lg focus:outline-none focus:border-green-500" required>**

**</div>**

**<div class="mb-4">**

**<label class="block text-gray-700 text-sm font-bold mb-2">Email</label>**

**<input type="email" id="register-email"**

**class="w-full px-3 py-2 border rounded-lg focus:outline-none focus:border-green-500" required>**

**</div>**

**<div class="mb-6">**

**<label class="block text-gray-700 text-sm font-bold mb-2">Password</label>**

**<input type="password" id="register-password"**

**class="w-full px-3 py-2 border rounded-lg focus:outline-none focus:border-green-500" required>**

**</div>**

**<button type="submit"**

**class="w-full bg-green-600 text-white py-2 rounded-lg hover:bg-green-700">Register</button>**

**</form>**

**<p class="text-center mt-4">**

**Already have an account?**

**<button onclick="showLogin()" class="text-green-600 hover:underline">Login here</button>**

**</p>**

**</div>**

**<div id="dashboard-screen" class="hidden fade-in max-w-7xl mx-auto px-4 sm:px-6 py-8">**

**<div class="mb-8">**

**<h2 class="text-3xl font-bold text-gray-800 mb-2">Welcome back, <span id="dashboard-user-name"></span>!**

**</h2>**

**<p class="text-gray-600">Continue your learning journey</p>**

**</div>**

**<div class="grid grid-cols-1 md:grid-cols-3 gap-4 md:gap-6 mb-8">**

**<div class="bg-gradient-to-r from-indigo-500 to-purple-600 text-white p-6 rounded-2xl">**

**<div class="flex items-center justify-between">**

**<div>**

**<p class="text-indigo-100">Enrolled Courses</p>**

**<p class="text-3xl font-bold" id="enrolled-count">0</p>**

**</div>**

**<div class="text-4xl opacity-80">📚</div>**

**</div>**

**</div>**

**<div class="bg-gradient-to-r from-green-500 to-emerald-600 text-white p-6 rounded-2xl">**

**<div class="flex items-center justify-between">**

**<div>**

**<p class="text-green-100">Completed</p>**

**<p class="text-3xl font-bold" id="completed-count">0</p>**

**</div>**

**<div class="text-4xl opacity-80">🏆</div>**

**</div>**

**</div>**

**<div class="bg-gradient-to-r from-orange-500 to-red-500 text-white p-6 rounded-2xl">**

**<div class="flex items-center justify-between">**

**<div>**

**<p class="text-orange-100">Certificates</p>**

**<p class="text-3xl font-bold" id="certificates-count">0</p>**

**</div>**

**<div class="text-4xl opacity-80">🎓</div>**

**</div>**

**</div>**

**</div>**

**<div class="grid grid-cols-1 lg:grid-cols-3 gap-4 md:gap-8">**

**<div class="lg:col-span-2">**

**<div class="bg-white rounded-2xl shadow-lg p-6">**

**<h3 class="text-xl font-semibold mb-4">Continue Learning</h3>**

**<div id="continue-learning" class="space-y-4">**

**</div>**

**</div>**

**</div>**

**<div>**

**<div class="bg-white rounded-2xl shadow-lg p-6">**

**<h3 class="text-xl font-semibold mb-4">Quick Actions</h3>**

**<div class="space-y-3">**

**<button onclick="showCourseList()"**

**class="w-full bg-indigo-600 text-white py-3 rounded-lg hover:bg-indigo-700 transition-colors">**

**Browse Courses**

**</button>**

**<button onclick="showProfile()"**

**class="w-full bg-purple-600 text-white py-3 rounded-lg hover:bg-purple-700 transition-colors">**

**My Courses**

**</button>**

**<button onclick="showCertificates()"**

**class="w-full bg-green-600 text-white py-3 rounded-lg hover:bg-green-700 transition-colors">**

**View Certificates**

**</button>**

**</div>**

**</div>**

**</div>**

**</div>**

**</div>**

**<div id="course-list" class="hidden fade-in max-w-7xl mx-auto px-4 sm:px-6 py-8">**

**<div class="text-center mb-12">**

**<h2 class="text-4xl font-bold text-gray-800 mb-4">All Courses</h2>**

**<p class="text-xl text-gray-600">Choose from our comprehensive course catalog</p>**

**</div>**

**<div class="mb-8 flex justify-center">**

**<div class="bg-white rounded-full p-2 shadow-lg">**

**<button onclick="filterCourses('all')"**

**class="language-filter active px-6 py-2 rounded-full transition-all">All</button>**

**<button onclick="filterCourses('programming')"**

**class="language-filter px-6 py-2 rounded-full transition-all">Programming</button>**

**<button onclick="filterCourses('design')"**

**class="language-filter px-6 py-2 rounded-full transition-all">Design</button>**

**<button onclick="filterCourses('business')"**

**class="language-filter px-6 py-2 rounded-full transition-all">Business</button>**

**<button onclick="filterCourses('data')"**

**class="language-filter px-6 py-2 rounded-full transition-all">Data Science</button>**

**</div>**

**</div>**

**<div class="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-4 md:gap-8" id="courses-container"></div>**

**</div>**

**<div id="certificates-screen" class="hidden fade-in max-w-7xl mx-auto px-4 sm:px-6 py-8">**

**<div class="text-center mb-12">**

**<h2 class="text-4xl font-bold text-gray-800 mb-4">My Certificates</h2>**

**<p class="text-xl text-gray-600">Your achievements and certifications</p>**

**</div>**

**<div id="certificates-container" class="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-4 md:gap-6">**

**</div>**

**</div>**

**<div id="course-detail" class="hidden fade-in">**

**<button onclick="showCourseList()"**

**class="mb-4 bg-gray-600 text-white px-4 py-2 rounded-lg hover:bg-gray-700">← Back to Courses</button>**

**<div id="course-detail-content"></div>**

**</div>**

**<div id="course-learning" class="hidden fade-in">**

**<div class="bg-white rounded-lg shadow-lg p-6">**

**<div class="flex justify-between items-center mb-6">**

**<button onclick="showCourseList()"**

**class="bg-gray-600 text-white px-4 py-2 rounded-lg hover:bg-gray-700">← Back to Courses</button>**

**<div class="text-right">**

**<div class="text-sm text-gray-600">Progress</div>**

**<div class="text-lg font-semibold" id="course-progress">0%</div>**

**</div>**

**</div>**

**<div id="course-content"></div>**

**</div>**

**</div>**

**<div id="profile-screen" class="hidden fade-in max-w-7xl mx-auto px-4 sm:px-6 py-8">**

**<h2 class="text-3xl font-bold text-center mb-8">My Courses</h2>**

**<div id="enrolled-courses" class="flex flex-wrap gap-4 justify-center">**

**</div>**

**</div>**

**<div id="certificate-screen" class="hidden fade-in">**

**<div class="max-w-4xl mx-auto">**

**<button onclick="showProfile()"**

**class="mb-4 bg-gray-600 text-white px-4 py-2 rounded-lg hover:bg-gray-700">← Back to My**

**Courses</button>**

**<div id="certificate-content" class="certificate text-white p-12 rounded-lg text-center"></div>**

**<div class="text-center mt-6">**

**<button onclick="downloadCertificate()"**

**class="bg-green-600 text-white px-6 py-3 rounded-lg hover:bg-green-700 text-lg">Download**

**Certificate</button>**

**</div>**

**</div>**

**</div>**

**<!-- Admin Dashboard Panel - Reworked for Enrollment Management -->**

**<div id="admin-dashboard-panel" class="hidden fade-in">**

**<div class="min-h-screen flex flex-col">**

**<header class="bg-white shadow-md z-10">**

**<div class="max-w-7xl mx-auto py-4 px-4 sm:px-6 lg:px-8 flex justify-between items-center">**

**<h1 class="text-3xl font-bold text-gray-900 slide-in">Admin Enrollment Dashboard 🔑</h1>**

**<div class="text-sm text-gray-500">**

**Last Refreshed: <span id="last-updated"**

**class="font-semibold text-blue-600">Loading...</span>**

**</div>**

**</div>**

**</header>**

**<main class="flex-1 p-4 sm:p-6 lg:p-8 max-w-7xl mx-auto w-full">**

**<!-- Metrics Row -->**

**<div class="grid grid-cols-1 md:grid-cols-3 gap-6 mb-8">**

**<div id="card-total-students"**

**class="card-animation bg-white p-6 rounded-xl shadow-lg border-l-4 border-blue-600">**

**<div class="flex items-center justify-between">**

**<div class="text-sm font-medium text-gray-500">Total Student Accounts</div>**

**<svg class="h-6 w-6 text-blue-400" fill="none" stroke="currentColor"**

**viewBox="0 0 24 24">**

**<path stroke-linecap="round" stroke-linejoin="round" stroke-width="2"**

**d="M17 20h5v-4h-5v4zm0 0l-1.336-1.336m1.336 1.336H3m3-4H2m3 4v-4m3 4v-4m3 4v-4m3 4v-4m3 4v-4m-3 4V4m3 16V4m-3 0h5m-5 0H3m3 0h12" />**

**</svg>**

**</div>**

**<p class="text-4xl font-extrabold text-gray-900 mt-1"><span**

**id="metric-total-students">0</span></p>**

**</div>**

**<div id="card-total-active-users"**

**class="card-animation bg-white p-6 rounded-xl shadow-lg border-l-4 border-red-600">**

**<div class="flex items-center justify-between">**

**<div class="text-sm font-medium text-gray-500">Total Active Users (Mock)</div>**

**<svg class="h-6 w-6 text-red-400" fill="none" stroke="currentColor" viewBox="0 0 24 24">**

**<path stroke-linecap="round" stroke-linejoin="round" stroke-width="2"**

**d="M13 10V3L4 14h7v7l9-11h-7z" />**

**</svg>**

**</div>**

**<p class="text-4xl font-extrabold text-gray-900 mt-1"><span**

**id="metric-total-active-users">0</span></p>**

**</div>**

**<div id="card-total-certificates"**

**class="card-animation bg-white p-6 rounded-xl shadow-lg border-l-4 border-green-600">**

**<div class="flex items-center justify-between">**

**<div class="text-sm font-medium text-gray-500">Courses Completed (Certified)</div>**

**<svg class="h-6 w-6 text-green-400" fill="none" stroke="currentColor"**

**viewBox="0 0 24 24">**

**<path stroke-linecap="round" stroke-linejoin="round" stroke-width="2"**

**d="M9 12l2 2 4-4m5.618-4.016A11.955 11.955 0 0112 2.944a11.955 11.955 0 01-8.618 3.04A12.01 12.01 0 002.5 14c0 1.491.31 2.923.863 4.264m18.774-11.28A12.01 12.01 0 0021.5 14c0 1.491-.31 2.923-.863 4.264M18 18h-4.264l-.5-1.5-.5 1.5H6m12 0a2 2 0 100 4 2 2 0 000-4z" />**

**</svg>**

**</div>**

**<p class="text-4xl font-extrabold text-gray-900 mt-1"><span**

**id="metric-total-certificates">0</span></p>**

**</div>**

**</div>**

**<!-- Graph and Filter Row (REVISED LAYOUT) -->**

**<div class="grid grid-cols-1 gap-6 mb-8">**

**<div class="bg-white p-6 rounded-xl shadow-lg card-animation w-full">**

**<h2 class="text-xl font-semibold text-gray-800 mb-4">Course Enrollment Distribution (Live**

**Data)</h2>**

**<div style="height: 300px;">**

**<canvas id="courseChart"></canvas>**

**</div>**

**</div>**

**<div**

**class="bg-white p-6 rounded-xl shadow-lg card-animation w-full grid grid-cols-1 md:grid-cols-3 gap-4">**

**<!-- Search & Filter Controls -->**

**<div>**

**<label for="admin-student-search" class="block text-sm font-medium text-gray-700">Search**

**Student Name/Email</label>**

**<input type="text" id="admin-student-search" placeholder="E.g., Alice Johnson"**

**class="mt-1 block w-full border border-gray-300 rounded-md shadow-sm py-2 px-3 focus:ring-blue-500 focus:border-blue-500 filter-transition">**

**</div>**

**<div>**

**<label for="admin-course-filter" class="block text-sm font-medium text-gray-700">Filter**

**by Course</label>**

**<select id="admin-course-filter"**

**class="mt-1 block w-full border border-gray-300 rounded-md shadow-sm py-2 px-3 focus:ring-blue-500 focus:border-blue-500 filter-transition bg-white">**

**<option value="all">All Courses</option>**

**</select>**

**</div>**

**<div>**

**<label for="admin-status-filter" class="block text-sm font-medium text-gray-700">Filter**

**by Enrollment Status</label>**

**<select id="admin-status-filter"**

**class="mt-1 block w-full border border-gray-300 rounded-md shadow-sm py-2 px-3 focus:ring-blue-500 focus:border-blue-500 filter-transition bg-white">**

**<option value="all">Any Status</option>**

**<option value="true">Active (True)</option>**

**<option value="false">Pending (False)</option>**

**</select>**

**</div>**

**</div>**

**</div>**

**<!-- Enrollment Table -->**

**<div class="bg-white p-6 rounded-xl shadow-lg card-animation">**

**<h2 class="text-xl font-semibold text-gray-800 mb-4">**

**Student Enrollment Records**

**<span id="approval-metrics" class="ml-4 text-sm font-normal text-gray-600">**

**(Active: <span id="metric-approvals" class="font-semibold text-green-600">0</span> /**

**Pending: <span id="metric-revokes" class="font-semibold text-orange-600">0</span>)**

**</span>**

**</h2>**

**<div class="overflow-x-auto">**

**<table class="min-w-full divide-y divide-gray-200">**

**<thead>**

**<tr class="bg-gray-50">**

**<th**

**class="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">**

**Student Name</th>**

**<th**

**class="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">**

**Email</th>**

**<th**

**class="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">**

**Course Title</th>**

**<th**

**class="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">**

**Status</th>**

**<th**

**class="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">**

**Action</th>**

**</tr>**

**</thead>**

**<tbody id="admin-table-body" class="bg-white divide-y divide-gray-200">**

**</tbody>**

**</table>**

**<div id="admin-no-results" class="hidden text-center py-8 text-gray-500">No enrollment**

**records found matching the criteria.</div>**

**</div>**

**</div>**

**</main>**

**</div>**

**</div>**

**</div>**

**<!-- NEW: Admin User Details Modal -->**

**<div id="admin-user-details-modal" class="modal hidden">**

**<div class="modal-content w-full max-w-xl">**

**<div class="flex justify-between items-center pb-4 border-b">**

**<h2 class="text-2xl font-bold text-gray-900" id="modal-user-name">Student Details</h2>**

**<button onclick="closeAdminUserDetails()" class="text-gray-500 hover:text-gray-700">**

**<svg class="w-6 h-6" fill="none" stroke="currentColor" viewBox="0 0 24 24">**

**<path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M6 18L18 6M6 6l12 12">**

**</svg>**

**</button>**

**</div>**

**<div id="modal-user-info" class="mt-4 space-y-1 text-sm text-gray-600">**

**<!-- User details go here -->**

**</div>**

**<div class="mt-6 flex justify-between space-x-3 border-t pt-4">**

**<button id="disable-user-btn"**

**onclick="toggleUserDisableStatus(document.getElementById('modal-user-details-modal').getAttribute('data-user-id'))"**

**class="bg-yellow-500 text-gray-900 px-4 py-2 rounded-lg text-sm hover:bg-yellow-600 transition-colors flex-1">Disable**

**User</button>**

**<button id="delete-user-btn"**

**onclick="deleteUserPermanently(document.getElementById('modal-user-details-modal').getAttribute('data-user-id'))"**

**class="bg-red-600 text-white px-4 py-2 rounded-lg text-sm hover:bg-red-700 transition-colors flex-1">Delete**

**User Permanently</button>**

**</div>**

**<div class="mt-6">**

**<h3 class="text-lg font-semibold text-gray-800 border-b pb-2 mb-4">Enrolled Courses (<span**

**id="modal-enrolled-count">0</span>)</h3>**

**<div id="modal-course-list" class="space-y-4">**

**<!-- Course cards go here -->**

**</div>**

**</div>**

**</div>**

**</div>**

**<!-- END NEW MODAL -->**

**<script>**

**// Application State**

**let currentCourse = null;**

**let currentDay = 1;**

**let currentUser = null;**

**const PAYMENT\_LINK = 'https://rzp.io/rzp/MdrDDje';**

**let isAdminLoggedIn = false;**

**const ADMIN\_EMAIL = 'admin@edupro.com';**

**const ADMIN\_PASSWORD = 'admin123';**

**let courseChartInstance = null; // To store the Chart.js instance**

**// Sample Data: COURSE IDs must be unique strings (unchanged)**

**const courses = [{**

**id: "web\_dev",**

**title: "Full Stack Web Development",**

**description: "Master modern web development with React, Node.js, and databases",**

**duration: "10 days",**

**level: "Beginner",**

**icon: "💻",**

**category: "programming",**

**rating: 4.9,**

**students: 2847,**

**curriculum: [**

**"HTML & CSS Fundamentals",**

**"JavaScript ES6+",**

**"React Components",**

**"State Management",**

**"Node.js Backend",**

**"Express.js Framework",**

**"Database Integration",**

**"API Development",**

**"Authentication",**

**"Deployment"**

**]**

**}, {**

**id: "data\_sci",**

**title: "Data Science & Analytics",**

**description: "Master Python, machine learning, and data visualization",**

**duration: "10 days",**

**level: "Intermediate",**

**icon: "📊",**

**category: "data",**

**rating: 4.8,**

**students: 1923,**

**curriculum: [**

**"Python for Data Science",**

**"NumPy & Pandas",**

**"Data Cleaning",**

**"Statistical Analysis",**

**"Data Visualization",**

**"Machine Learning Basics",**

**"Scikit-learn",**

**"Model Evaluation",**

**"Deep Learning Intro",**

**"Capstone Project"**

**]**

**}, {**

**id: "digital\_mkt",**

**title: "Digital Marketing Mastery",**

**description: "Complete digital marketing strategy and execution",**

**duration: "10 days",**

**level: "Beginner",**

**icon: "📈",**

**category: "business",**

**rating: 4.7,**

**students: 3421,**

**curriculum: [**

**"Marketing Strategy",**

**"SEO Optimization",**

**"Content Marketing",**

**"Social Media Strategy",**

**"Email Campaigns",**

**"Google Analytics",**

**"Paid Advertising",**

**"Conversion Optimization",**

**"Brand Building",**

**"Campaign Analysis"**

**]**

**}, {**

**id: "mobile\_dev",**

**title: "Mobile App Development",**

**description: "Build cross-platform mobile apps with React Native",**

**duration: "10 days",**

**level: "Intermediate",**

**icon: "📱",**

**category: "programming",**

**rating: 4.6,**

**students: 1567,**

**curriculum: [**

**"React Native Setup",**

**"Components & Navigation",**

**"State Management",**

**"API Integration",**

**"Native Features",**

**"Push Notifications",**

**"App Store Guidelines",**

**"Testing & Debugging",**

**"Performance Optimization",**

**"Publishing"**

**]**

**}, {**

**id: "cyber\_sec",**

**title: "Cybersecurity Fundamentals",**

**description: "Learn ethical hacking and security best practices",**

**duration: "10 days",**

**level: "Advanced",**

**icon: "🔒",**

**category: "programming",**

**rating: 4.9,**

**students: 892,**

**curriculum: [**

**"Security Principles",**

**"Network Security",**

**"Cryptography",**

**"Vulnerability Assessment",**

**"Penetration Testing",**

**"Incident Response",**

**"Security Tools",**

**"Compliance Standards",**

**"Risk Management",**

**"Capstone Project"**

**]**

**}, {**

**id: "aws\_cloud",**

**title: "Cloud Computing with AWS",**

**description: "Master Amazon Web Services and cloud architecture",**

**duration: "10 days",**

**level: "Intermediate",**

**icon: "☁️",**

**category: "programming",**

**rating: 4.8,**

**students: 2156,**

**curriculum: [**

**"AWS Fundamentals",**

**"EC2 & VPC",**

**"S3 & Storage",**

**"RDS & Databases",**

**"Lambda Functions",**

**"CloudFormation",**

**"Security & IAM",**

**"Monitoring & Logging",**

**"DevOps Integration",**

**"Certification Prep"**

**]**

**}, {**

**id: "ui\_ux",**

**title: "UI/UX Design Professional",**

**description: "Create stunning user experiences and interfaces",**

**duration: "10 days",**

**level: "Beginner",**

**icon: "🎨",**

**category: "design",**

**rating: 4.7,**

**students: 2834,**

**curriculum: [**

**"Design Thinking",**

**"User Research",**

**"Wireframing",**

**"Prototyping",**

**"Visual Design",**

**"Interaction Design",**

**"Usability Testing",**

**"Design Systems",**

**"Figma Mastery",**

**"Portfolio Building"**

**]**

**}, {**

**id: "ai\_ml",**

**title: "Artificial Intelligence & ML",**

**description: "Deep dive into AI, neural networks, and machine learning",**

**duration: "10 days",**

**level: "Advanced",**

**icon: "🤖",**

**category: "data",**

**rating: 4.9,**

**students: 1245,**

**curriculum: [**

**"AI Fundamentals",**

**"Machine Learning Algorithms",**

**"Neural Networks",**

**"Deep Learning",**

**"Computer Vision",**

**"Natural Language Processing",**

**"TensorFlow & PyTorch",**

**"Model Deployment",**

**"AI Ethics",**

**"Industry Applications"**

**]**

**}, {**

**id: "blockchain",**

**title: "Blockchain & Web3 Development",**

**description: "Build decentralized applications and smart contracts",**

**duration: "10 days",**

**level: "Advanced",**

**icon: "⛓️",**

**category: "programming",**

**rating: 4.6,**

**students: 743,**

**curriculum: [**

**"Blockchain Fundamentals",**

**"Ethereum & Solidity",**

**"Smart Contracts",**

**"Web3.js Integration",**

**"DeFi Protocols",**

**"NFT Development",**

**"Security Best Practices",**

**"Testing & Deployment",**

**"Gas Optimization",**

**"DApp Project"**

**]**

**}, {**

**id: "agile\_pm",**

**title: "Agile Project Management",**

**description: "Master Scrum, Kanban, and modern project leadership",**

**duration: "10 days",**

**level: "Beginner",**

**icon: "📋",**

**category: "business",**

**rating: 4.5,**

**students: 1876,**

**curriculum: [**

**"Agile Principles",**

**"Scrum Framework",**

**"Kanban Method",**

**"Sprint Planning",**

**"Team Leadership",**

**"Stakeholder Management",**

**"Risk Assessment",**

**"Quality Assurance",**

**"Project Tools",**

**"PMP Preparation"**

**]**

**}, {**

**id: "python\_prog",**

**title: "Python Programming Mastery",**

**description: "Complete Python course from basics to advanced concepts",**

**duration: "10 days",**

**level: "Beginner",**

**icon: "🐍",**

**category: "programming",**

**rating: 4.8,**

**students: 3567,**

**curriculum: [**

**"Python Syntax",**

**"Data Structures",**

**"Object-Oriented Programming",**

**"File Handling",**

**"Error Handling",**

**"Libraries & Modules",**

**"Web Scraping",**

**"API Development",**

**"Testing",**

**"Real-world Projects"**

**]**

**}, {**

**id: "graphic\_design",**

**title: "Graphic Design Fundamentals",**

**description: "Learn Adobe Creative Suite and design principles",**

**duration: "10 days",**

**level: "Beginner",**

**icon: "🖌️",**

**category: "design",**

**rating: 4.6,**

**students: 2145,**

**curriculum: [**

**"Design Principles",**

**"Color Theory",**

**"Typography",**

**"Adobe Photoshop",**

**"Adobe Illustrator",**

**"Logo Design",**

**"Brand Identity",**

**"Print Design",**

**"Digital Design",**

**"Portfolio Creation"**

**]**

**}, {**

**id: "business\_ana",**

**title: "Business Analytics",**

**description: "Data-driven decision making for business growth",**

**duration: "10 days",**

**level: "Intermediate",**

**icon: "📊",**

**category: "business",**

**rating: 4.7,**

**students: 1432,**

**curriculum: [**

**"Business Intelligence",**

**"Excel Advanced",**

**"SQL for Business",**

**"Tableau Visualization",**

**"KPI Development",**

**"Forecasting",**

**"A/B Testing",**

**"Customer Analytics",**

**"Financial Modeling",**

**"Strategic Planning"**

**]**

**}, {**

**id: "devops",**

**title: "DevOps Engineering",**

**description: "Master CI/CD, containerization, and infrastructure automation",**

**duration: "10 days",**

**level: "Advanced",**

**icon: "⚙️",**

**category: "programming",**

**rating: 4.8,**

**students: 987,**

**curriculum: [**

**"DevOps Culture",**

**"Git & Version Control",**

**"Docker Containers",**

**"Kubernetes",**

**"CI/CD Pipelines",**

**"Infrastructure as Code",**

**"Monitoring & Logging",**

**"Security Integration",**

**"Cloud Platforms",**

**"Automation"**

**]**

**}, {**

**id: "content\_create",**

**title: "Content Creation & Video Editing",**

**description: "Create engaging content for social media and marketing",**

**duration: "10 days",**

**level: "Beginner",**

**icon: "🎬",**

**category: "design",**

**rating: 4.5,**

**students: 1890,**

**curriculum: [**

**"Storyboarding",**

**"Video Shooting Basics",**

**"Adobe Premiere Pro",**

**"Color Grading",**

**"Sound Design",**

**"Motion Graphics",**

**"YouTube Optimization",**

**"Social Media Formats",**

**"Content Strategy",**

**"Final Project"**

**]**

**},];**

**const firebaseConfig = {**

**apiKey: "AIzaSyBnQ0s1jFoMH6J7vhKVM7NVC-WNScLdA7E",**

**authDomain: "sutherland-6ce2b.firebaseapp.com",**

**databaseURL: "https://sutherland-6ce2b-default-rtdb.firebaseio.com",**

**projectId: "sutherland-6ce2b",**

**storageBucket: "sutherland-6ce2b.firebasestorage.app",**

**messagingSenderId: "243800407888",**

**appId: "1:243800407888:web:2c4dfa1e6a267ba2a17655",**

**measurementId: "G-S6GMQ32JJ5"**

**};**

**firebase.initializeApp(firebaseConfig);**

**const auth = firebase.auth();**

**const database = firebase.database();**

**// Data Storage**

**let users = [];**

**let certificates = [];**

**let currentUserId = null;**

**// --- UI Helpers ---**

**function hideAllScreens() {**

**['welcome-screen', 'login-form', 'register-form', 'dashboard-screen', 'course-list', 'course-detail', 'course-learning', 'profile-screen', 'certificate-screen', 'certificates-screen', 'admin-dashboard-panel', 'admin-login-screen', 'admin-user-details-modal'].forEach(id => {**

**document.getElementById(id).style.display = 'none';**

**});**

**}**

**function updateNav(isLoggedIn, isAdmin) {**

**const navButtons = document.getElementById('nav-buttons');**

**const userNav = document.getElementById('user-nav');**

**const adminNav = document.getElementById('admin-nav');**

**navButtons.style.display = 'none';**

**userNav.style.display = 'none';**

**adminNav.style.display = 'none';**

**if (isAdmin) {**

**adminNav.style.display = 'flex';**

**} else if (isLoggedIn) {**

**if (currentUser) {**

**document.getElementById('user-name-nav').textContent = currentUser.name.split(' ')[0];**

**document.getElementById('user-avatar').textContent = currentUser.name.charAt(0);**

**}**

**userNav.style.display = 'flex';**

**} else {**

**navButtons.style.display = 'flex';**

**}**

**}**

**// --- Screen Navigation Functions (Moved to top scope for access) ---**

**function showWelcomeScreen() {**

**hideAllScreens();**

**isAdminLoggedIn = false;**

**updateNav(false, false);**

**document.getElementById('welcome-screen').style.display = 'block';**

**}**

**function showLogin() { hideAllScreens(); document.getElementById('login-form').style.display = 'block'; }**

**function showRegister() { hideAllScreens(); document.getElementById('register-form').style.display = 'block'; }**

**function showCourseList() { hideAllScreens(); document.getElementById('course-list').style.display = 'block'; renderCourseList(courses); resetFilterButtons('all'); }**

**function showDashboard() { if (!currentUserId) return showLogin(); hideAllScreens(); document.getElementById('dashboard-screen').style.display = 'block'; renderDashboard(); }**

**function showCertificates() { if (!currentUserId) return showLogin(); hideAllScreens(); document.getElementById('certificates-screen').style.display = 'block'; renderCertificates(); }**

**function showProfile() { if (!currentUserId) return showLogin(); hideAllScreens(); document.getElementById('profile-screen').style.display = 'block'; renderProfileCourses(); }**

**function showAdminLogin() { hideAllScreens(); document.getElementById('admin-login-screen').style.display = 'block'; }**

**function showCertificateScreen(certData) {**

**hideAllScreens();**

**document.getElementById('certificate-screen').style.display = 'block';**

**renderCertificate(certData);**

**}**

**async function showAdminDashboard() {**

**if (!isAdminLoggedIn) return showAdminLogin();**

**hideAllScreens();**

**document.getElementById('admin-dashboard-panel').style.display = 'block';**

**await fetchUserData(); // Ensure latest data before rendering**

**renderAdminDashboard();**

**}**

**function startLearning(courseId) {**

**if (!currentUserId || !currentUser) return showLogin();**

**const course = courses.find(c => c.id === courseId);**

**const userEnrollment = currentUser.enrolled\_courses ? currentUser.enrolled\_courses[courseId] : null;**

**// Check if user is actively enrolled (unlocked: status === true)**

**if (!userEnrollment || userEnrollment.status !== true) {**

**alert("Course not yet unlocked. Please check your payment status on the 'My Courses' page. Status must be ACTIVE (True) for access.");**

**showCourseDetail(courseId);**

**return;**

**}**

**hideAllScreens();**

**document.getElementById('course-learning').style.display = 'block';**

**currentCourse = course;**

**// IMPORTANT: If starting, ensure day is reset to 1 if coming from 100% completion before certification was issued**

**const currentDayState = userEnrollment.day || 1;**

**currentDay = currentDayState;**

**renderCourseContent();**

**}**

**// --- Core Application Logic ---**

**function initializeUI() {**

**auth.onAuthStateChanged((user) => {**

**if (user) {**

**// Check if the user is the hardcoded admin (for mock UI display purposes)**

**if (user.email === ADMIN\_EMAIL && user.uid === 'admin-mock-uid') {**

**// Rely on local flag isAdminLoggedIn**

**} else {**

**currentUserId = user.uid;**

**fetchUserData().then(() => {**

**updateNav(true, false); // Logged in as student**

**if (document.getElementById('welcome-screen').style.display !== 'none' ||**

**document.getElementById('login-form').style.display !== 'none' ||**

**document.getElementById('register-form').style.display !== 'none'**

**) {**

**showDashboard();**

**}**

**});**

**}**

**} else {**

**currentUserId = null;**

**currentUser = null;**

**updateNav(false, isAdminLoggedIn); // Not logged in**

**if (!isAdminLoggedIn) { // Only redirect if not on Admin screen**

**if (document.getElementById('dashboard-screen').style.display === 'block' ||**

**document.getElementById('profile-screen').style.display === 'block' ||**

**document.getElementById('certificates-screen').style.display === 'block' ||**

**document.getElementById('course-learning').style.display === 'block'**

**) {**

**showWelcomeScreen();**

**}**

**}**

**}**

**});**

**loadPopularCourses();**

**}**

**async function fetchUserData() {**

**try {**

**const userSnapshot = await database.ref('edupro\_users').once('value');**

**users = userSnapshot.val() ? Object.keys(userSnapshot.val()).map(uid => ({**

**uid,**

**...userSnapshot.val()[uid]**

**})) : [];**

**currentUser = users.find(u => u.uid === currentUserId);**

**const certificateSnapshot = await database.ref('certificates').once('value');**

**certificates = certificateSnapshot.val() ? Object.values(certificateSnapshot.val()) : [];**

**} catch (error) {**

**console.error("Error fetching data:", error);**

**users = [];**

**currentUser = null;**

**certificates = [];**

**}**

**}**

**// --- Auth & DB Handlers (includes Disable/Delete User) ---**

**async function handleAdminLogin(event) {**

**event.preventDefault();**

**const email = document.getElementById('admin-email').value;**

**const password = document.getElementById('admin-password').value;**

**if (email === ADMIN\_EMAIL && password === ADMIN\_PASSWORD) {**

**isAdminLoggedIn = true;**

**updateNav(false, true);**

**showAdminDashboard();**

**} else {**

**alert("Invalid Admin Credentials.");**

**}**

**}**

**function adminLogout() {**

**isAdminLoggedIn = false;**

**updateNav(false, false);**

**showWelcomeScreen();**

**}**

**async function handleLogin(event) {**

**event.preventDefault();**

**const email = document.getElementById('login-email').value;**

**const password = document.getElementById('login-password').value;**

**try {**

**// Fetch all users to check disabled status before signing in (mock)**

**await fetchUserData();**

**const userToLogin = users.find(u => u.email === email);**

**if (userToLogin && userToLogin.disabled === true) {**

**alert("Account is temporarily disabled by the administrator. Please contact support.");**

**return;**

**}**

**await auth.signInWithEmailAndPassword(email, password);**

**} catch (error) {**

**alert(`Login Failed: ${error.message}`);**

**}**

**}**

**async function handleRegister(event) {**

**event.preventDefault();**

**const name = document.getElementById('register-name').value;**

**const email = document.getElementById('register-email').value;**

**const password = document.getElementById('register-password').value;**

**try {**

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

**const uid = userCredential.user.uid;**

**await database.ref('edupro\_users/' + uid).set({**

**name: name,**

**email: email,**

**joined: new Date().toISOString().split('T')[0],**

**disabled: false // NEW: User is enabled by default**

**});**

**} catch (error) {**

**alert(`Registration Failed: ${error.message}`);**

**}**

**}**

**function logout() {**

**auth.signOut();**

**showWelcomeScreen();**

**}**

**async function initiatePayment(courseId) {**

**if (!currentUserId) return showLogin();**

**if (!currentUser) await fetchUserData();**

**const course = courses.find(c => c.id === courseId);**

**const userEnrollment = currentUser.enrolled\_courses ? currentUser.enrolled\_courses[courseId] : null;**

**if (userEnrollment && userEnrollment.status === true) {**

**alert(`You are already enrolled and the ${course.title} course is ACTIVE!`);**

**return;**

**}**

**if (userEnrollment && userEnrollment.status === false) {**

**// Already pending, just redirect to payment and remind them.**

**window.open(PAYMENT\_LINK, '\_blank');**

**alert("Your enrollment is already PENDING confirmation. Redirecting to payment portal again.");**

**return;**

**}**

**try {**

**// 1. Mark enrollment as PENDING (status: false) with day 1**

**await database.ref(`edupro\_users/${currentUserId}/enrolled\_courses/${courseId}`).set({**

**status: false,**

**day: 1 // Always start at day 1 when first enrolling**

**});**

**// 2. Refresh local data and update UI**

**await fetchUserData();**

**currentUser = users.find(u => u.uid === currentUserId);**

**showCourseDetail(courseId); // Update the button/status message**

**// 3. Redirect user to payment link in a new tab**

**window.open(PAYMENT\_LINK, '\_blank');**

**alert(`Payment initiated for ${course.title}. Your enrollment is PENDING confirmation. Please allow time for admin approval.`);**

**} catch (error) {**

**alert(`Failed to initiate enrollment: ${error.message}`);**

**}**

**}**

**async function advanceDay(courseId) {**

**if (!currentUserId || !currentUser) return showLogin();**

**const course = courses.find(c => c.id === courseId);**

**const userEnrollment = currentUser.enrolled\_courses ? currentUser.enrolled\_courses[courseId] : null;**

**// Must be UNLOCKED (status: true)**

**if (!userEnrollment || userEnrollment.status !== true) {**

**alert("This course is currently locked. Please contact support if you believe this is an error.");**

**return;**

**}**

**const nextDay = userEnrollment.day + 1;**

**const isCompleted = nextDay > course.curriculum.length;**

**try {**

**if (isCompleted) {**

**await database.ref(`edupro\_users/${currentUserId}/enrolled\_courses/${courseId}/day`).set(course.curriculum.length);**

**const certificateId = `${currentUserId}-${courseId}`;**

**const certData = {**

**id: certificateId,**

**userId: currentUserId,**

**userName: currentUser.name,**

**courseId: courseId,**

**courseTitle: course.title,**

**date: new Date().toISOString().split('T')[0]**

**};**

**await database.ref(`certificates/${certificateId}`).set(certData);**

**await fetchUserData();**

**currentUser = users.find(u => u.uid === currentUserId);**

**showCertificateScreen(certData);**

**alert(`Congratulations! You have completed the ${course.title} course and earned a certificate!`);**

**} else {**

**await database.ref(`edupro\_users/${currentUserId}/enrolled\_courses/${courseId}/day`).set(nextDay);**

**await fetchUserData();**

**currentUser = users.find(u => u.uid === currentUserId);**

**currentDay = nextDay;**

**renderCourseContent();**

**}**

**} catch (error) {**

**console.error("Error advancing day:", error);**

**alert(`Failed to update progress: ${error.message}`);**

**}**

**}**

**// NEW: Permanent deletion of user data**

**async function deleteUserPermanently(userId) {**

**if (!isAdminLoggedIn) return;**

**const userToDelete = users.find(u => u.uid === userId);**

**if (!userToDelete) return;**

**if (confirm(`ARE YOU SURE you want to permanently delete ALL data for user: ${userToDelete.name} (${userToDelete.email})? This action cannot be undone.`)) {**

**try {**

**// 1. Delete user profile and enrollments**

**await database.ref(`edupro\_users/${userId}`).remove();**

**// 2. Delete associated certificates (filter locally, delete from DB)**

**certificates.filter(c => c.userId === userId).forEach(async (cert) => {**

**await database.ref(`certificates/${cert.id}`).remove();**

**});**

**// 3. Re-fetch data and update UI**

**await fetchUserData();**

**renderAdminDashboard();**

**closeAdminUserDetails();**

**alert(`User ${userToDelete.name} and all associated data have been permanently deleted.`);**

**} catch (error) {**

**console.error("Error deleting user:", error);**

**alert(`Failed to delete user: ${error.message}`);**

**}**

**}**

**}**

**// NEW: Toggle disable status**

**async function toggleUserDisableStatus(userId) {**

**if (!isAdminLoggedIn) return;**

**const userToToggle = users.find(u => u.uid === userId);**

**if (!userToToggle) return;**

**const currentDisabledStatus = userToToggle.disabled || false;**

**const newDisabledStatus = !currentDisabledStatus;**

**try {**

**await database.ref(`edupro\_users/${userId}/disabled`).set(newDisabledStatus);**

**await fetchUserData();**

**renderAdminDashboard();**

**// Update modal view right after the change**

**showAdminUserDetails(userId);**

**alert(`User ${userToToggle.name}'s account status changed to: ${newDisabledStatus ? 'DISABLED (Cannot Log In)' : 'ENABLED (Can Log In)'}`);**

**} catch (error) {**

**console.error("Error toggling disable status:", error);**

**alert(`Failed to change disable status: ${error.message}`);**

**}**

**}**

**// --- Render Functions (Student / Unchanged) ---**

**function renderCourseCard(course) { /\* ... unchanged ... \*/**

**const userEnrollment = currentUser && currentUser.enrolled\_courses ? currentUser.enrolled\_courses[course.id] : null;**

**const isEnrolled = userEnrollment && userEnrollment.status === true;**

**const isPending = userEnrollment && userEnrollment.status === false;**

**const isNotEnrolled = !userEnrollment;**

**let cardClasses = 'shadow-lg';**

**let buttonText = 'View Details';**

**if (currentUser) {**

**if (isEnrolled) {**

**cardClasses = 'border-4 border-green-500 shadow-xl';**

**buttonText = 'View Course';**

**} else if (isPending) {**

**cardClasses = 'border-4 border-orange-500 shadow-xl';**

**buttonText = 'Awaiting Confirmation';**

**} else if (isNotEnrolled) {**

**buttonText = 'Enroll Now';**

**}**

**}**

**return `**

**<div class="card-hover bg-white p-6 rounded-xl ${cardClasses}">**

**<div class="text-3xl mb-3">${course.icon}</div>**

**<h3 class="text-xl font-bold text-gray-800 mb-2">${course.title}</h3>**

**<p class="text-gray-600 mb-4">${course.description}</p>**

**<div class="flex justify-between text-sm text-gray-500 mb-4">**

**<span>Level: ${course.level}</span>**

**<span>Duration: ${course.duration}</span>**

**</div>**

**<div class="flex items-center space-x-1 text-yellow-500 mb-4">**

**${'★'.repeat(Math.floor(course.rating))}**

**${course.rating % 1 !== 0 ? '☆' : ''}**

**<span class="text-gray-600 ml-1">(${course.students} enrolled)</span>**

**</div>**

**<button onclick="showCourseDetail('${course.id}')" class="w-full bg-indigo-600 text-white py-2 rounded-lg hover:bg-indigo-700 transition-colors">**

**${buttonText}**

**</button>**

**</div>**

**`;**

**}**

**function loadPopularCourses() { /\* ... unchanged ... \*/**

**const container = document.getElementById('popular-courses-preview');**

**const popularCourses = courses.slice(0, 3);**

**container.innerHTML = popularCourses.map(renderCourseCard).join('');**

**}**

**function renderCourseList(filteredCourses) { /\* ... unchanged ... \*/**

**const container = document.getElementById('courses-container');**

**container.innerHTML = filteredCourses.map(renderCourseCard).join('');**

**}**

**function filterCourses(category) { /\* ... unchanged ... \*/**

**let filtered = courses;**

**if (category !== 'all') {**

**filtered = courses.filter(course => course.category === category);**

**}**

**renderCourseList(filtered);**

**resetFilterButtons(category);**

**}**

**function resetFilterButtons(activeCategory) { /\* ... unchanged ... \*/**

**document.querySelectorAll('.language-filter').forEach(btn => {**

**btn.classList.remove('active');**

**});**

**const activeBtn = document.querySelector(`.language-filter[onclick="filterCourses('${activeCategory}')"]`);**

**if (activeBtn) {**

**activeBtn.classList.add('active');**

**}**

**}**

**async function showCourseDetail(courseId) { /\* ... unchanged ... \*/**

**const course = courses.find(c => c.id === courseId);**

**if (!course) return;**

**hideAllScreens();**

**document.getElementById('course-detail').style.display = 'block';**

**const container = document.getElementById('course-detail-content');**

**container.setAttribute('data-course-id', courseId);**

**const userEnrollment = currentUser && currentUser.enrolled\_courses ? currentUser.enrolled\_courses[courseId] : null;**

**const isEnrolled = userEnrollment && userEnrollment.status === true;**

**const isPending = userEnrollment && userEnrollment.status === false;**

**const isNotEnrolled = !currentUser || !userEnrollment;**

**const currentDay = userEnrollment ? userEnrollment.day : 1;**

**let actionButton;**

**let enrollmentMessage;**

**if (isEnrolled) {**

**actionButton = `<button onclick="startLearning('${course.id}')" class="w-full bg-green-600 text-white py-3 rounded-lg text-lg font-semibold hover:bg-green-700 transition-colors">**

**Continue Learning (Day ${currentDay})**

**</button>`;**

**enrollmentMessage = `<p class="text-center text-sm text-gray-500 mt-4">You have full, lifetime access to this course.</p>`;**

**} else if (isPending) {**

**actionButton = `<div class="p-4 bg-orange-100 border border-orange-400 rounded-lg text-orange-800">**

**<p class="font-semibold mb-2">Enrollment is \*\*PENDING\*\* admin confirmation.</p>**

**<p class="text-sm">Please allow time for processing. Your status: <span class="font-bold">LOCKED (FALSE)</span></p>**

**</div>`;**

**enrollmentMessage = `<p class="text-center text-sm text-gray-500 mt-4">Check My Courses for status updates. <button onclick="window.open(PAYMENT\_LINK, '\_blank')" class="text-indigo-600 underline">View Payment Portal</button></p>`;**

**} else if (isNotEnrolled && currentUser) {**

**actionButton = `<button onclick="initiatePayment('${course.id}')" class="w-full bg-indigo-600 text-white py-3 rounded-lg text-lg font-semibold hover:bg-indigo-700 transition-colors shadow-xl">**

**Enroll Now - Pay Now**

**</button>`;**

**enrollmentMessage = `<p class="text-center text-sm text-gray-500 mt-4">Clicking above initiates enrollment (sets status to FALSE) and opens the payment portal.</p>`;**

**} else {**

**actionButton = `<button onclick="showLogin()" class="w-full bg-indigo-600 text-white py-3 rounded-lg text-lg font-semibold hover:bg-indigo-700 transition-colors shadow-xl">**

**Sign In to Enroll**

**</button>`;**

**enrollmentMessage = `<p class="text-center text-sm text-gray-500 mt-4">You must be logged in to enroll in this course.</p>`;**

**}**

**const curriculumHTML = course.curriculum.map((item, index) => {**

**const day = index + 1;**

**const completed = isEnrolled && day < currentDay;**

**const current = isEnrolled && day === currentDay;**

**const statusIcon = completed ? '✅' : current ? '🔥' : '🔒';**

**const statusColor = completed ? 'text-green-600' : current ? 'text-red-500' : 'text-gray-500';**

**const isLockedVisual = !isEnrolled || (isEnrolled && day > currentDay);**

**return `**

**<li class="p-4 border-b last:border-b-0 flex items-center justify-between ${isLockedVisual ? 'opacity-50' : ''}">**

**<span class="font-medium flex items-center">Day ${day}: ${item}</span>**

**<span class="${statusColor}">${statusIcon}</span>**

**</li>**

**`;**

**}).join('');**

**const html = `**

**<div class="max-w-7xl mx-auto bg-white rounded-2xl shadow-2xl overflow-hidden p-8">**

**<div class="md:flex md:space-x-8">**

**<div class="md:w-2/3">**

**<h2 class="text-5xl font-extrabold text-gray-900 mb-3">${course.icon} ${course.title}</h2>**

**<p class="text-xl text-gray-600 mb-6">${course.description}</p>**

**<div class="flex items-center space-x-6 text-gray-700 mb-8">**

**<span class="flex items-center">**

**<svg class="w-5 h-5 mr-1 text-indigo-500" fill="currentColor" viewBox="0 0 20 20"><path d="M10 12a2 2 0 100-4 2 2 0 000 4z"/><path fill-rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-8-5a5 5 0 00-5 5c0 1.05.176 2.05.51 3L10 16l4.49-3A5 5 0 0015 10a5 5 0 00-5-5z" clip-rule="evenodd"/></svg>**

**${course.level}**

**</span>**

**<span class="flex items-center">**

**<svg class="w-5 h-5 mr-1 text-indigo-500" fill="currentColor" viewBox="0 0 20 20"><path d="M10 18a8 8 0 100-16 8 8 0 000 16zm1-12a1 1 0 10-2 0v4a1 1 0 00.293.707l2.828 2.829a1 1 0 101.415-1.415L11 9.586V6z"/></svg>**

**${course.duration}**

**</span>**

**<span class="flex items-center text-yellow-500">**

**${'★'.repeat(Math.floor(course.rating))}**

**<span class="text-gray-600 ml-1">(${course.rating})</span>**

**</span>**

**</div>**

**<h3 class="text-3xl font-bold text-gray-800 mb-4">What you will learn:</h3>**

**<ul class="grid grid-cols-2 gap-4 text-gray-700">**

**${course.curriculum.slice(0, 6).map(item => `<li class="flex items-center"><svg class="w-4 h-4 mr-2 text-green-500" fill="currentColor" viewBox="0 0 20 20"><path fill-rule="evenodd" d="M16.707 5.293a1 1 0 010 1.414l-8 8a1 1 0 01-1.414 0l-4-4a1 1 0 011.414-1.414L8 12.586l7.293-7.293a1 1 0 011.414 0z" clip-rule="evenodd"/></svg> ${item}</li>`).join('')}**

**</ul>**

**</div>**

**<div class="md:w-1/3 mt-8 md:mt-0">**

**<div class="bg-gray-50 p-6 rounded-xl shadow-inner sticky top-24">**

**<h3 class="text-2xl font-bold text-gray-800 mb-4">Course Access</h3>**

**<p class="text-3xl font-extrabold text-indigo-600 mb-6">Course Fee: FREE\*</p>**

**${actionButton}**

**${enrollmentMessage}**

**</div>**

**</div>**

**</div>**

**<div class="mt-12 pt-8 border-t border-gray-200">**

**<h3 class="text-3xl font-bold text-gray-800 mb-6">Full Curriculum (${course.curriculum.length} Days)</h3>**

**<ul class="bg-white border rounded-xl shadow-lg divide-y divide-gray-100">**

**${curriculumHTML}**

**</ul>**

**</div>**

**</div>**

**`;**

**container.innerHTML = html;**

**}**

**function renderCourseContent() { /\* ... unchanged ... \*/**

**const container = document.getElementById('course-content');**

**const progressDisplay = document.getElementById('course-progress');**

**const totalDays = currentCourse.curriculum.length;**

**const progress = Math.round((currentDay / totalDays) \* 100);**

**const content = currentCourse.curriculum[currentDay - 1]; // currentDay is 1-indexed**

**progressDisplay.textContent = `${progress}% (Day ${currentDay} of ${totalDays})`;**

**const html = `**

**<h3 class="text-4xl font-bold text-gray-800 mb-6">${currentCourse.icon} ${currentCourse.title}</h3>**

**<div class="mb-8">**

**<div class="text-sm font-medium text-indigo-600 mb-1">Your Progress</div>**

**<div class="w-full bg-gray-200 rounded-full h-2.5">**

**<div class="progress-bar bg-green-500 h-2.5 rounded-full" style="width: ${progress}%"></div>**

**</div>**

**</div>**

**<div class="bg-indigo-50 p-6 rounded-xl border-l-4 border-indigo-500 mb-8">**

**<h4 class="text-2xl font-semibold text-indigo-800 mb-3">Day ${currentDay}: ${content}</h4>**

**<p class="text-gray-700">This section covers the core concepts of \*\*${content}\*\*. Make sure to follow the recommended practical exercises to fully grasp the material. (Mock Content)</p>**

**</div>**

**<div class="space-y-4">**

**<h5 class="text-xl font-semibold text-gray-800 border-b pb-2">Learning Resources (Mock)</h5>**

**<div class="flex items-center p-3 bg-white border rounded-lg hover:bg-gray-50 transition-colors">**

**<span class="text-xl mr-3">📖</span>**

**<a href="#" class="text-indigo-600 hover:underline">Download: Official E-Book for Day ${currentDay}</a>**

**</div>**

**<div class="flex items-center p-3 bg-white border rounded-lg hover:bg-gray-50 transition-colors">**

**<span class="text-xl mr-3">▶️</span>**

**<a href="#" class="text-indigo-600 hover:underline">Watch: Video Tutorial for ${content}</a>**

**</div>**

**<div class="flex items-center p-3 bg-white border rounded-lg hover:bg-gray-50 transition-colors">**

**<span class="text-xl mr-3">🧪</span>**

**<a href="#" class="text-indigo-600 hover:underline">Practice: Interactive Quiz for Day ${currentDay}</a>**

**</div>**

**</div>**

**<div class="mt-10">**

**<button onclick="advanceDay('${currentCourse.id}')" class="w-full bg-indigo-600 text-white py-3 rounded-lg text-xl font-semibold hover:bg-indigo-700 transition-colors">**

**${currentDay < totalDays ? `Mark Day ${currentDay} Complete & Go to Next Day →` : 'Complete Course & Get Certificate 🏆'}**

**</button>**

**</div>**

**`;**

**container.innerHTML = html;**

**}**

**function renderDashboard() { /\* ... unchanged ... \*/**

**if (!currentUser) return;**

**document.getElementById('dashboard-user-name').textContent = currentUser.name.split(' ')[0];**

**const enrolledCourses = currentUser.enrolled\_courses || {};**

**// Only count UNLOCKED (status: true) and PENDING (status: false) courses for total count**

**const enrolledCourseIds = Object.keys(enrolledCourses).filter(id => enrolledCourses[id].status === true || enrolledCourses[id].status === false);**

**document.getElementById('enrolled-count').textContent = enrolledCourseIds.length;**

**const userCertificates = certificates.filter(c => c.userId === currentUserId);**

**const completedCount = userCertificates.length;**

**document.getElementById('completed-count').textContent = completedCount;**

**document.getElementById('certificates-count').textContent = completedCount;**

**// Render Continue Learning (Only unlocked courses from enrolled\_courses where status is TRUE)**

**const continueContainer = document.getElementById('continue-learning');**

**continueContainer.innerHTML = '';**

**const activeProgress = Object.keys(enrolledCourses)**

**.map(id => ({ id, ...enrolledCourses[id] }))**

**.filter(e => e.status === true && !userCertificates.some(c => c.courseId === e.id)); // Must be unlocked AND not completed**

**if (activeProgress.length === 0) {**

**continueContainer.innerHTML = `**

**<p class="text-gray-500 py-4 text-center">No courses currently in progress. <button onclick="showCourseList()" class="text-indigo-600 hover:underline font-semibold">Start a new course!</button></p>**

**`;**

**} else {**

**activeProgress.slice(0, 3).forEach(p => { // Show up to 3 active courses**

**const course = courses.find(c => c.id === p.id);**

**if (course) {**

**const totalDays = course.curriculum.length;**

**const progress = Math.round((p.day / totalDays) \* 100);**

**continueContainer.innerHTML += `**

**<div class="flex items-center bg-gray-50 p-4 rounded-xl border border-gray-200">**

**<div class="text-3xl mr-4">${course.icon}</div>**

**<div class="flex-1">**

**<h4 class="font-semibold text-gray-800">${course.title}</h4>**

**<p class="text-sm text-gray-500">Day ${p.day} of ${totalDays} - ${course.curriculum[p.day - 1]}</p>**

**<div class="mt-1 w-full bg-gray-200 rounded-full h-1.5">**

**<div class="progress-bar bg-indigo-500 h-1.5 rounded-full" style="width: ${progress}%"></div>**

**</div>**

**</div>**

**<button onclick="startLearning('${course.id}')" class="ml-4 bg-indigo-500 text-white px-4 py-2 rounded-lg text-sm hover:bg-indigo-600 transition-colors">**

**Continue**

**</button>**

**</div>**

**`;**

**}**

**});**

**}**

**}**

**function renderProfileCourses() { /\* ... unchanged ... \*/**

**const container = document.getElementById('enrolled-courses');**

**container.innerHTML = '';**

**if (!currentUser || !currentUser.enrolled\_courses) {**

**container.innerHTML = `<p class="text-gray-500 py-8">You are not currently enrolled in any courses. <button onclick="showCourseList()" class="text-indigo-600 hover:underline font-semibold">Browse courses</button> to get started!</p>`;**

**return;**

**}**

**const enrolledCourses = currentUser.enrolled\_courses;**

**const enrolledCourseIds = Object.keys(enrolledCourses);**

**const userCertificates = certificates.filter(c => c.userId === currentUserId);**

**// Filter to only show courses that are explicitly enrolled or pending (status true or false)**

**const activeEnrollments = enrolledCourseIds.filter(id => enrolledCourses[id].status === true || enrolledCourses[id].status === false);**

**if (activeEnrollments.length === 0) {**

**container.innerHTML = `<p class="text-gray-500 py-8">You are not currently enrolled in any courses. <button onclick="showCourseList()" class="text-indigo-600 hover:underline font-semibold">Browse courses</button> to get started!</p>`;**

**return;**

**}**

**activeEnrollments.forEach(courseId => {**

**const course = courses.find(c => c.id === courseId);**

**const enrollment = enrolledCourses[courseId];**

**if (course && enrollment) {**

**const isEnrolled = enrollment.status === true;**

**const isPending = enrollment.status === false;**

**let actionButton;**

**let statusTag;**

**let cardClasses = 'border-t-4';**

**if (isEnrolled) {**

**const totalDays = course.curriculum.length;**

**const progress = Math.round((enrollment.day / totalDays) \* 100);**

**const isComplete = progress === 100;**

**const hasCertificate = userCertificates.some(c => c.courseId === courseId);**

**cardClasses += ' border-green-500';**

**statusTag = `<span class="bg-green-100 text-green-800 text-xs font-medium px-2.5 py-0.5 rounded-full">Active</span>`;**

**if (isComplete) {**

**actionButton = hasCertificate ?**

**`<button onclick="showCertificateScreen(certificates.find(c => c.courseId === '${courseId}' && c.userId === currentUserId))" class="w-full bg-green-600 text-white py-2 rounded-lg hover:bg-green-700">View Certificate 🏆</button>` :**

**`<button onclick="advanceDay('${courseId}')" class="w-full bg-green-600 text-white py-2 rounded-lg hover:bg-green-700">Finish & Get Certificate 🎓</button>`;**

**} else {**

**actionButton = `<button onclick="startLearning('${courseId}')" class="w-full bg-indigo-600 text-white py-2 rounded-lg hover:bg-indigo-700">Continue Learning (Day ${enrollment.day})</button>`;**

**}**

**} else if (isPending) {**

**cardClasses += ' border-orange-500';**

**statusTag = `<span class="bg-orange-100 text-orange-800 text-xs font-medium px-2.5 py-0.5 rounded-full">Pending Confirmation</span>`;**

**actionButton = `<button onclick="showCourseDetail('${courseId}')" class="w-full bg-orange-600 text-white py-2 rounded-lg hover:bg-orange-700">View Status</button>`;**

**}**

**container.innerHTML += `**

**<div class="card-hover bg-white p-6 rounded-xl shadow-lg w-full sm:w-80 ${cardClasses}">**

**<div class="flex justify-between items-center mb-2">**

**<div class="text-3xl">${course.icon}</div>**

**${statusTag}**

**</div>**

**<h3 class="text-xl font-bold text-gray-800 mb-2">${course.title}</h3>**

**<p class="text-sm text-gray-600 mb-4">${course.description}</p>**

**${isEnrolled ? `**

**<div class="mb-4">**

**<div class="flex justify-between text-sm font-medium text-gray-700">**

**<span>Progress:</span>**

**<span>${Math.round((enrollment.day / course.curriculum.length) \* 100)}%</span>**

**</div>**

**<div class="w-full bg-gray-200 rounded-full h-2.5 mt-1">**

**<div class="progress-bar bg-green-500 h-2.5 rounded-full" style="width: ${Math.round((enrollment.day / course.curriculum.length) \* 100)}%"></div>**

**</div>**

**</div>**

**` : `<div class="h-10 mb-4"></div>`}**

**${actionButton}**

**</div>**

**`;**

**}**

**});**

**}**

**function renderCertificates() { /\* ... unchanged ... \*/**

**const container = document.getElementById('certificates-container');**

**container.innerHTML = '';**

**const userCertificates = certificates.filter(c => c.userId === currentUserId);**

**if (userCertificates.length === 0) {**

**container.innerHTML = `<p class="text-gray-500 py-8 text-center w-full">You have not completed any courses yet. Keep learning!</p>`;**

**return;**

**}**

**userCertificates.forEach(cert => {**

**const course = courses.find(c => c.id === cert.courseId);**

**if (course) {**

**container.innerHTML += `**

**<div class="card-hover bg-white p-6 rounded-xl shadow-lg border-t-4 border-yellow-500 text-center">**

**<div class="text-5xl mb-3">🎓</div>**

**<h3 class="text-xl font-bold text-gray-800 mb-1">${course.title}</h3>**

**<p class="text-gray-600 mb-3">Certificate of Completion</p>**

**<p class="text-sm text-gray-500 mb-4">Awarded on: ${cert.date}</p>**

**<button onclick="showCertificateScreen(certificates.find(c => c.id === '${cert.id}'))" class="w-full bg-yellow-500 text-gray-900 py-2 rounded-lg hover:bg-yellow-600 transition-colors font-semibold">**

**View Certificate**

**</button>**

**</div>**

**`;**

**}**

**});**

**}**

**function renderCertificate(certData) { /\* ... unchanged ... \*/**

**if (!certData) return;**

**const course = courses.find(c => c.id === certData.courseId);**

**if (!course) return;**

**const container = document.getElementById('certificate-content');**

**container.innerHTML = `**

**<div class="certificate-bg p-12 rounded-lg text-center">**

**<h2 class="text-4xl font-serif font-light mb-4 text-yellow-300">Certificate of Achievement</h2>**

**<p class="text-xl font-light mb-2">This certifies that</p>**

**<h1 class="text-6xl font-extrabold mb-6 text-white certificate-name">${certData.userName}</h1>**

**<p class="text-xl font-light mb-4">has successfully completed the course</p>**

**<h2 class="text-3xl font-bold mb-8 text-yellow-300">${course.title}</h2>**

**<div class="flex justify-around items-center text-lg mt-8">**

**<div>**

**<p class="font-bold text-white">${certData.date}</p>**

**<p class="text-sm text-yellow-300">Date Issued</p>**

**</div>**

**<div>**

**<p class="font-bold text-white">EduPro Academy</p>**

**<p class="text-sm text-yellow-300">Authority</p>**

**</div>**

**</div>**

**</div>**

**`;**

**container.setAttribute('data-course-title', course.title);**

**container.setAttribute('data-user-name', certData.userName);**

**}**

**function downloadCertificate() { /\* ... unchanged ... \*/**

**const {**

**jsPDF**

**} = window.jspdf;**

**const doc = new jsPDF({**

**orientation: 'landscape',**

**unit: 'in',**

**format: 'letter'**

**});**

**const courseTitle = document.getElementById('certificate-content').getAttribute('data-course-title');**

**const userName = document.getElementById('certificate-content').getAttribute('data-user-name');**

**const certData = certificates.find(c => c.userName === userName && c.courseTitle === courseTitle);**

**const date = certData ? certData.date : new Date().toISOString().split('T')[0];**

**const pageWidth = doc.internal.pageSize.width;**

**const pageHeight = doc.internal.pageSize.height;**

**doc.setFillColor(30, 60, 114);**

**doc.rect(0, 0, pageWidth, pageHeight, 'F');**

**doc.setDrawColor(255, 215, 0);**

**doc.setLineWidth(0.2);**

**doc.rect(0.2, 0.2, pageWidth - 0.4, pageHeight - 0.4, 'S');**

**doc.rect(0.4, 0.4, pageWidth - 0.8, pageHeight - 0.8, 'S');**

**doc.setFont('times', 'bold');**

**doc.setFontSize(36);**

**doc.setTextColor(255, 215, 0);**

**doc.text("CERTIFICATE OF ACHIEVEMENT", pageWidth / 2, 1.5, {**

**align: 'center'**

**});**

**doc.setFontSize(14);**

**doc.setTextColor(255, 255, 255);**

**doc.setFont('times', 'normal');**

**doc.text("This certifies that", pageWidth / 2, 2.2, {**

**align: 'center'**

**});**

**doc.setFontSize(50);**

**doc.setTextColor(255, 255, 255);**

**doc.setFont('cursive');**

**doc.text(userName, pageWidth / 2, 3.5, {**

**align: 'center'**

**});**

**doc.setFontSize(18);**

**doc.setTextColor(255, 255, 255);**

**doc.setFont('times', 'normal');**

**doc.text("has successfully completed the course", pageWidth / 2, 4.3, {**

**align: 'center'**

**});**

**doc.setFontSize(28);**

**doc.setTextColor(255, 215, 0);**

**doc.setFont('times', 'bold');**

**doc.text(courseTitle, pageWidth / 2, 5.0, {**

**align: 'center'**

**});**

**doc.setFontSize(12);**

**doc.setTextColor(255, 255, 255);**

**doc.setFont('times', 'normal');**

**doc.line(1.5, 6.5, 3.5, 6.5);**

**doc.text("Director Signature (Mock)", 2.5, 6.8, {**

**align: 'center'**

**});**

**doc.line(8.0, 6.5, 10.0, 6.5);**

**doc.text(`Date: ${date}`, 9.0, 6.8, {**

**align: 'center'**

**});**

**doc.save(`${userName.replace(/ /g, '\_')}\_${courseTitle.replace(/ /g, '\_')}\_Certificate.pdf`);**

**}**

**// --- ADMIN LOGIC ---**

**function getCourseTitle(courseId) {**

**const course = courses.find(c => c.id === courseId);**

**return course ? course.title : 'Unknown Course';**

**}**

**function getCourseByTitle(title) {**

**return courses.find(c => c.title === title);**

**}**

**async function toggleCourseStatus(userId, courseId, currentStatus) {**

**if (!isAdminLoggedIn) {**

**alert("Authorization failed. Please log in as Admin.");**

**return;**

**}**

**const newStatus = currentStatus === 'false' ? true : false;**

**try {**

**// Update the status node in Firebase**

**await database.ref(`edupro\_users/${userId}/enrolled\_courses/${courseId}/status`).set(newStatus);**

**// Fetch the updated data and re-render the table**

**await fetchUserData();**

**renderAdminDashboard(); // Re-render everything to update graphs/metrics/table**

**alert(`Status for ${getCourseTitle(courseId)} for user ${users.find(u => u.uid === userId).name} updated to ${newStatus ? 'TRUE (ACTIVE)' : 'FALSE (PENDING)'}`);**

**} catch (error) {**

**console.error("Error toggling course status:", error);**

**alert(`Failed to update status: ${error.message}`);**

**}**

**}**

**// NEW: Function to render the progress card inside the modal**

**function renderCourseProgressCard(courseData, userCertificates) {**

**const course = getCourseByTitle(courseData.title);**

**if (!course) return '';**

**const totalDays = course.curriculum.length;**

**const progressDay = courseData.day || 1;**

**const progress = Math.min(100, Math.round((progressDay / totalDays) \* 100)); // Cap at 100%**

**const isCertified = userCertificates.some(c => c.courseTitle === courseData.title);**

**let statusTag;**

**if (isCertified) {**

**statusTag = `<span class="bg-green-500 text-white text-xs font-medium px-2 py-1 rounded-full">Certified</span>`;**

**} else if (courseData.status === true) {**

**statusTag = `<span class="bg-yellow-500 text-gray-800 text-xs font-medium px-2 py-1 rounded-full">In Progress</span>`;**

**} else {**

**statusTag = `<span class="bg-orange-500 text-white text-xs font-medium px-2 py-1 rounded-full">Pending Approval</span>`;**

**}**

**let viewButton = '';**

**if (isCertified) {**

**const cert = userCertificates.find(c => c.courseTitle === courseData.title);**

**viewButton = `<button onclick="alert('Cert ID: ${cert.id}')" class="bg-indigo-600 text-white text-xs px-3 py-1 rounded-full ml-3 hover:bg-indigo-700">View Certificate</button>`;**

**}**

**return `**

**<div class="p-4 border rounded-lg shadow-sm">**

**<div class="flex justify-between items-center mb-2">**

**<h4 class="font-bold text-gray-900">${courseData.title}</h4>**

**<div class="flex items-center">**

**${statusTag}**

**${viewButton}**

**</div>**

**</div>**

**<p class="text-sm text-gray-500">Progress: ${progress}% (Day ${progressDay} of ${totalDays})</p>**

**<div class="w-full bg-gray-200 rounded-full h-2 mt-1">**

**<div class="progress-bar bg-green-500 h-2 rounded-full" style="width: ${progress}%"></div>**

**</div>**

**<p class="text-xs mt-2 text-gray-500">Enrollment Status: ${courseData.status === true ? 'Active' : 'Locked (Pending)'}</p>**

**</div>**

**`;**

**}**

**// NEW: Function to render the Admin Dashboard User Modal**

**function showAdminUserDetails(userId) {**

**const user = users.find(u => u.uid === userId);**

**if (!user) return;**

**const modal = document.getElementById('admin-user-details-modal');**

**modal.setAttribute('data-user-id', userId); // Store ID for button actions**

**document.getElementById('modal-user-name').textContent = `${user.name}'s Profile`;**

**// 1. User Info**

**const userInfoContainer = document.getElementById('modal-user-info');**

**const overallProgress = (function () {**

**const enrolled = user.enrolled\_courses || {};**

**let totalDaysCompleted = 0;**

**let totalMaxDays = 0;**

**Object.keys(enrolled).forEach(id => {**

**const course = courses.find(c => c.id === id);**

**if (course && enrolled[id].status === true) {**

**totalDaysCompleted += enrolled[id].day;**

**totalMaxDays += course.curriculum.length;**

**}**

**});**

**return totalMaxDays > 0 ? Math.round((totalDaysCompleted / totalMaxDays) \* 100) : 0;**

**})();**

**const userCertificates = certificates.filter(c => c.userId === userId);**

**const userDisabled = user.disabled || false;**

**userInfoContainer.innerHTML = `**

**<div class="grid grid-cols-2 gap-x-6 gap-y-2 py-2">**

**<p class="col-span-2"><strong>Email:</strong> ${user.email}</p>**

**<p><strong>Joined:</strong> ${user.joined}</p>**

**<p><strong>Overall Progress:</strong> <span class="font-bold text-blue-600">${overallProgress}%</span></p>**

**<p><strong>Total Certificates:</strong> <span class="font-bold text-green-600">${userCertificates.length}</span></p>**

**<p><strong>Status:</strong> <span class="font-bold ${userDisabled ? 'text-red-500' : 'text-green-500'}">${userDisabled ? 'DISABLED' : 'ENABLED'}</span></p>**

**</div>**

**`;**

**// 2. Button Logic**

**const disableBtn = document.getElementById('disable-user-btn');**

**if (userDisabled) {**

**disableBtn.textContent = 'Enable User';**

**disableBtn.classList.remove('bg-yellow-500', 'hover:bg-yellow-600');**

**disableBtn.classList.add('bg-green-500', 'hover:bg-green-600');**

**} else {**

**disableBtn.textContent = 'Disable User';**

**disableBtn.classList.remove('bg-green-500', 'hover:bg-green-600');**

**disableBtn.classList.add('bg-yellow-500', 'hover:bg-yellow-600');**

**}**

**// 3. Course List**

**const courseListContainer = document.getElementById('modal-course-list');**

**courseListContainer.innerHTML = '';**

**const userEnrollments = user.enrolled\_courses || {};**

**const enrolledCourseIds = Object.keys(userEnrollments);**

**document.getElementById('modal-enrolled-count').textContent = enrolledCourseIds.length;**

**if (enrolledCourseIds.length === 0) {**

**courseListContainer.innerHTML = `<p class="text-gray-500 text-center py-4">This student has no active enrollments yet.</p>`;**

**} else {**

**enrolledCourseIds.forEach(courseId => {**

**const enrollment = userEnrollments[courseId];**

**const course = courses.find(c => c.id === courseId);**

**if (course) {**

**const courseData = { title: course.title, day: enrollment.day, status: enrollment.status };**

**courseListContainer.innerHTML += renderCourseProgressCard(courseData, userCertificates);**

**}**

**});**

**}**

**modal.classList.remove('hidden');**

**modal.style.display = 'flex';**

**}**

**function closeAdminUserDetails() {**

**document.getElementById('admin-user-details-modal').classList.add('hidden');**

**document.getElementById('admin-user-details-modal').style.display = 'none';**

**}**

**// NEW: Function to calculate enrollment data for the chart**

**function getCourseEnrollmentDistributionData() {**

**const enrollmentMap = {}; // { courseId: { enrolled: count, certified: count } }**

**courses.forEach(course => {**

**enrollmentMap[course.id] = { enrolled: 0, certified: 0, title: course.title };**

**});**

**// Calculate active/pending enrollments and total approvals/revokes**

**let totalApprovals = 0;**

**let totalRevokes = 0;**

**users.forEach(user => {**

**const userEnrollments = user.enrolled\_courses || {};**

**Object.keys(userEnrollments).forEach(courseId => {**

**if (enrollmentMap[courseId]) {**

**// Count enrollments (true or false status)**

**enrollmentMap[courseId].enrolled++;**

**// Count approvals/revokes globally**

**if (userEnrollments[courseId].status === true) {**

**totalApprovals++;**

**} else {**

**totalRevokes++;**

**}**

**}**

**});**

**});**

**certificates.forEach(cert => {**

**if (enrollmentMap[cert.courseId]) {**

**enrollmentMap[cert.courseId].certified++;**

**}**

**});**

**return {**

**labels: Object.values(enrollmentMap).map(d => d.title),**

**enrolledData: Object.values(enrollmentMap).map(d => d.enrolled),**

**certifiedData: Object.values(enrollmentMap).map(d => d.certified),**

**totalApprovals,**

**totalRevokes**

**};**

**}**

**function renderAdminDashboard() {**

**if (!isAdminLoggedIn) return;**

**// --- METRICS UPDATE ---**

**const distributionData = getCourseEnrollmentDistributionData();**

**document.getElementById('metric-total-students').textContent = users.length;**

**document.getElementById('metric-total-certificates').textContent = certificates.length;**

**document.getElementById('last-updated').textContent = new Date().toLocaleTimeString();**

**document.getElementById('metric-total-active-users').textContent = Math.floor(Math.random() \* users.length \* 0.4);**

**// NEW: Update Approval/Revoke metrics**

**document.getElementById('metric-approvals').textContent = distributionData.totalApprovals;**

**document.getElementById('metric-revokes').textContent = distributionData.totalRevokes;**

**// --- GRAPH UPDATE ---**

**const ctx = document.getElementById('courseChart').getContext('2d');**

**if (courseChartInstance) {**

**courseChartInstance.destroy();**

**}**

**courseChartInstance = new Chart(ctx, {**

**type: 'bar',**

**data: {**

**labels: distributionData.labels,**

**datasets: [{**

**label: 'Total Enrolled (Active or Pending)',**

**data: distributionData.enrolledData,**

**backgroundColor: 'rgba(59, 130, 246, 0.7)', // Blue**

**borderColor: 'rgba(59, 130, 246, 1)',**

**borderWidth: 1**

**}, {**

**label: 'Certified (Completed)',**

**data: distributionData.certifiedData,**

**backgroundColor: 'rgba(16, 185, 129, 0.7)', // Green**

**borderColor: 'rgba(16, 185, 129, 1)',**

**borderWidth: 1**

**}]**

**},**

**options: {**

**responsive: true,**

**maintainAspectRatio: false,**

**scales: {**

**y: {**

**beginAtZero: true,**

**title: {**

**display: true,**

**text: 'Number of Students'**

**}**

**}**

**}**

**}**

**});**

**// --- FILTER SETUP & TABLE RENDER ---**

**// Populate Course Filter**

**const courseFilter = document.getElementById('admin-course-filter');**

**courseFilter.innerHTML = '<option value="all">All Courses</option>';**

**courses.forEach(c => {**

**const option = document.createElement('option');**

**option.value = c.id;**

**option.textContent = c.title;**

**courseFilter.appendChild(option);**

**});**

**renderAdminStudentTable();**

**// Setup listeners for filtering (already set up in previous versions, ensuring they call renderAdminStudentTable)**

**document.getElementById('admin-student-search').oninput = renderAdminStudentTable;**

**document.getElementById('admin-course-filter').onchange = renderAdminStudentTable;**

**document.getElementById('admin-status-filter').onchange = renderAdminStudentTable;**

**}**

**function renderAdminStudentTable() {**

**const tableBody = document.getElementById('admin-table-body');**

**const noResults = document.getElementById('admin-no-results');**

**tableBody.innerHTML = '';**

**const searchTerm = document.getElementById('admin-student-search').value.toLowerCase();**

**const filterCourseId = document.getElementById('admin-course-filter').value;**

**const filterStatus = document.getElementById('admin-status-filter').value;**

**// --- LIFO SORTING (Newest user first based on 'joined' timestamp) ---**

**const sortedUsers = users.slice().sort((a, b) => {**

**const dateA = new Date(a.joined);**

**const dateB = new Date(b.joined);**

**// Sort Descending (Newest first)**

**return dateB - dateA;**

**});**

**// --- END LIFO SORTING ---**

**// Flatten data: one row per user-course enrollment**

**const enrollmentRecords = [];**

**sortedUsers.forEach(user => {**

**const userEnrollments = user.enrolled\_courses || {};**

**Object.keys(userEnrollments).forEach(courseId => {**

**enrollmentRecords.push({**

**userId: user.uid,**

**userName: user.name,**

**userEmail: user.email,**

**courseId: courseId,**

**courseTitle: getCourseTitle(courseId),**

**status: userEnrollments[courseId].status // boolean true/false**

**});**

**});**

**});**

**const filteredRecords = enrollmentRecords.filter(record => {**

**// 1. Name/Email Search**

**if (searchTerm && !record.userName.toLowerCase().includes(searchTerm) && !record.userEmail.toLowerCase().includes(searchTerm)) {**

**return false;**

**}**

**// 2. Course Filter**

**if (filterCourseId !== 'all' && record.courseId !== filterCourseId) {**

**return false;**

**}**

**// 3. Status Filter**

**if (filterStatus === 'true' && record.status !== true) {**

**return false;**

**}**

**if (filterStatus === 'false' && record.status !== false) {**

**return false;**

**}**

**return true;**

**});**

**if (filteredRecords.length === 0) {**

**noResults.style.display = 'block';**

**} else {**

**noResults.style.display = 'none';**

**filteredRecords.forEach(record => {**

**const statusText = record.status ? 'ACTIVE (TRUE)' : 'PENDING (FALSE)';**

**const statusClass = record.status ? 'bg-green-100 text-green-800' : 'bg-orange-100 text-orange-800';**

**// Only show the APPROVE button if the status is FALSE**

**let actionButton;**

**if (!record.status) {**

**actionButton = `<button onclick="toggleCourseStatus('${record.userId}', '${record.courseId}', 'false')"**

**class="bg-blue-600 text-white px-4 py-2 rounded-lg text-xs hover:bg-blue-700 transition-colors">**

**APPROVE**

**</button>`;**

**} else {**

**// Option to Revoke/Unapprove (sets status back to false)**

**actionButton = `<button onclick="toggleCourseStatus('${record.userId}', '${record.courseId}', 'true')"**

**class="bg-gray-400 text-white px-4 py-2 rounded-lg text-xs hover:bg-gray-500 transition-colors">**

**Revoke**

**</button>`;**

**}**

**const row = `**

**<tr>**

**<td class="px-6 py-4 whitespace-nowrap text-sm font-medium text-gray-900">${record.userName}</td>**

**<td class="px-6 py-4 whitespace-nowrap text-sm text-gray-500">${record.userEmail}</td>**

**<td class="px-6 py-4 whitespace-nowrap text-sm font-medium text-indigo-600">${record.courseTitle}</td>**

**<td class="px-6 py-4 whitespace-nowrap text-sm">**

**<span class="px-2 inline-flex text-xs leading-5 font-semibold rounded-full ${statusClass}">**

**${statusText}**

**</span>**

**</td>**

**<td class="px-6 py-4 whitespace-nowrap text-sm flex space-x-2">**

**${actionButton}**

**<button onclick="showAdminUserDetails('${record.userId}')"**

**class="bg-purple-600 text-white px-4 py-2 rounded-lg text-xs hover:bg-purple-700 transition-colors">**

**Profile**

**</button>**

**</td>**

**</tr>**

**`;**

**tableBody.innerHTML += row;**

**});**

**}**

**}**

**// Global initialization call**

**initializeUI();**

**</script>**

**<script>**

**// Security scripts (unchanged)**

**document.addEventListener('contextmenu', function (event) { event.preventDefault(); return false; });**

**document.addEventListener('keydown', function (event) {**

**if (event.key === 'F12' || event.keyCode === 123) { event.preventDefault(); return false; }**

**if (event.ctrlKey || event.metaKey) {**

**if (event.shiftKey && (event.key === 'I' || event.key === 'i')) { event.preventDefault(); return false; }**

**if (event.key === 'u' || event.key === 'U') { event.preventDefault(); return false; }**

**}**

**});**

**</script>**

**</body>**

**</html>**