<!DOCTYPE html>

<html lang="he" dir="rtl">

<head>

<meta charset="UTF-8">

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

<title>מערכת בחינות מתקדמת</title>

<style>

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

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

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

min-height: 100vh;

display: flex;

align-items: center;

justify-content: center;

padding: 20px;

}

.exam-container {

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

border-radius: 20px;

padding: 40px;

max-width: 800px;

width: 100%;

box-shadow: 0 20px 40px rgba(0, 0, 0, 0.1);

backdrop-filter: blur(10px);

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

}

.exam-header {

text-align: center;

margin-bottom: 30px;

padding-bottom: 20px;

border-bottom: 3px solid #667eea;

}

.exam-header h1 {

color: #333;

font-size: 2.5em;

margin-bottom: 10px;

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

}

.exam-header p {

color: #666;

font-size: 1.2em;

margin-bottom: 15px;

}

.exam-info {

display: flex;

justify-content: space-around;

margin-bottom: 30px;

flex-wrap: wrap;

}

.info-item {

background: linear-gradient(45deg, #667eea, #764ba2);

color: white;

padding: 15px 25px;

border-radius: 15px;

text-align: center;

margin: 5px;

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

transform: translateY(0);

transition: transform 0.3s ease;

}

.info-item:hover {

transform: translateY(-5px);

}

.info-item h3 {

font-size: 1.1em;

margin-bottom: 5px;

}

.info-item p {

font-size: 1.3em;

font-weight: bold;

}

.payment-container {

background: linear-gradient(135deg, #ff6b6b, #ee5a24);

color: white;

padding: 30px;

border-radius: 15px;

text-align: center;

margin-bottom: 30px;

box-shadow: 0 10px 30px rgba(255, 107, 107, 0.3);

}

.payment-container h2 {

margin-bottom: 20px;

font-size: 2em;

}

.payment-methods {

display: flex;

justify-content: center;

gap: 15px;

flex-wrap: wrap;

margin-top: 20px;

}

.payment-method {

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

padding: 15px 25px;

border-radius: 10px;

cursor: pointer;

transition: all 0.3s ease;

border: 2px solid transparent;

}

.payment-method:hover {

background: rgba(255, 255, 255, 0.3);

border-color: white;

}

.exam-form {

background: white;

border-radius: 15px;

padding: 30px;

box-shadow: 0 10px 30px rgba(0, 0, 0, 0.1);

border: 2px solid #f0f0f0;

}

.form-group {

margin-bottom: 25px;

}

.form-group label {

display: block;

margin-bottom: 8px;

font-weight: bold;

color: #333;

font-size: 1.1em;

}

.form-group input,

.form-group select {

width: 100%;

padding: 15px;

border: 2px solid #ddd;

border-radius: 10px;

font-size: 1em;

transition: all 0.3s ease;

background: #f9f9f9;

}

.form-group input:focus,

.form-group select:focus {

outline: none;

border-color: #667eea;

background: white;

box-shadow: 0 0 10px rgba(102, 126, 234, 0.2);

}

.button-group {

display: flex;

gap: 15px;

justify-content: center;

margin-top: 30px;

flex-wrap: wrap;

}

.btn {

padding: 15px 30px;

border: none;

border-radius: 10px;

font-size: 1.1em;

font-weight: bold;

cursor: pointer;

transition: all 0.3s ease;

min-width: 150px;

text-transform: uppercase;

letter-spacing: 1px;

}

.btn-primary {

background: linear-gradient(45deg, #667eea, #764ba2);

color: white;

box-shadow: 0 5px 15px rgba(102, 126, 234, 0.4);

}

.btn-primary:hover {

transform: translateY(-2px);

box-shadow: 0 8px 25px rgba(102, 126, 234, 0.6);

}

.btn-secondary {

background: linear-gradient(45deg, #f093fb, #f5576c);

color: white;

box-shadow: 0 5px 15px rgba(240, 147, 251, 0.4);

}

.btn-secondary:hover {

transform: translateY(-2px);

box-shadow: 0 8px 25px rgba(240, 147, 251, 0.6);

}

.btn-demo {

background: linear-gradient(45deg, #4facfe, #00f2fe);

color: white;

box-shadow: 0 5px 15px rgba(79, 172, 254, 0.4);

}

.btn-demo:hover {

transform: translateY(-2px);

box-shadow: 0 8px 25px rgba(79, 172, 254, 0.6);

}

.btn-success {

background: linear-gradient(45deg, #56ab2f, #a8e6cf);

color: white;

box-shadow: 0 5px 15px rgba(86, 171, 47, 0.4);

}

.btn-success:hover {

transform: translateY(-2px);

box-shadow: 0 8px 25px rgba(86, 171, 47, 0.6);

}

.timer {

position: fixed;

top: 20px;

right: 20px;

background: linear-gradient(45deg, #ff6b6b, #ee5a24);

color: white;

padding: 15px 25px;

border-radius: 15px;

font-size: 1.5em;

font-weight: bold;

box-shadow: 0 5px 15px rgba(255, 107, 107, 0.4);

z-index: 1000;

display: none;

}

.form-toggle {

text-align: center;

margin-bottom: 20px;

}

.toggle-btn {

background: linear-gradient(45deg, #667eea, #764ba2);

color: white;

border: none;

padding: 12px 25px;

border-radius: 25px;

font-size: 1em;

cursor: pointer;

transition: all 0.3s ease;

margin: 0 10px;

}

.toggle-btn.active {

background: linear-gradient(45deg, #4facfe, #00f2fe);

transform: scale(1.05);

}

.form-section {

display: none;

}

.form-section.active {

display: block;

}

.success-message {

background: linear-gradient(45deg, #56ab2f, #a8e6cf);

color: white;

padding: 20px;

border-radius: 15px;

text-align: center;

margin-top: 20px;

display: none;

font-size: 1.2em;

box-shadow: 0 5px 15px rgba(86, 171, 47, 0.3);

}

.error-message {

background: linear-gradient(45deg, #ff6b6b, #ee5a24);

color: white;

padding: 15px;

border-radius: 10px;

margin-top: 10px;

display: none;

text-align: center;

}

.practical-exam-info {

background: linear-gradient(135deg, #56ab2f, #a8e6cf);

color: white;

padding: 25px;

border-radius: 15px;

margin-top: 20px;

text-align: center;

box-shadow: 0 5px 15px rgba(86, 171, 47, 0.3);

}

.practical-exam-info h3 {

font-size: 1.8em;

margin-bottom: 15px;

}

.practical-exam-details {

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

padding: 20px;

border-radius: 10px;

margin-top: 15px;

text-align: right;

}

.practical-exam-details p {

margin-bottom: 10px;

font-size: 1.1em;

}

.payment-processing {

background: linear-gradient(135deg, #667eea, #764ba2);

color: white;

padding: 30px;

border-radius: 15px;

text-align: center;

margin-top: 20px;

}

.payment-processing .spinner {

width: 50px;

height: 50px;

border: 5px solid rgba(255, 255, 255, 0.3);

border-top: 5px solid white;

border-radius: 50%;

animation: spin 1s linear infinite;

margin: 20px auto;

}

@keyframes spin {

0% { transform: rotate(0deg); }

100% { transform: rotate(360deg); }

}

.google-form-container {

margin-top: 30px;

border-radius: 15px;

overflow: hidden;

box-shadow: 0 10px 30px rgba(0, 0, 0, 0.1);

border: 2px solid #f0f0f0;

}

@media (max-width: 768px) {

.exam-container {

padding: 20px;

margin: 10px;

}

.exam-header h1 {

font-size: 2em;

}

.info-item {

flex: 1;

min-width: 120px;

}

.button-group {

flex-direction: column;

align-items: center;

}

.btn {

width: 100%;

max-width: 300px;

}

.payment-methods {

flex-direction: column;

align-items: center;

}

.payment-method {

width: 100%;

max-width: 200px;

}

}

</style>

</head>

<body>

<div class="timer" id="timer">

<div id="timeDisplay">00:00:00</div>

</div>

<div class="exam-container" id="mainContainer">

<!-- שלב 1: תשלום -->

<div id="paymentStep" class="step active">

<div class="exam-header">

<h1>💳 תשלום עבור הבחינה</h1>

<p>לפני תחילת הבחינה, נדרש תשלום של ₪50</p>

</div>

<div class="payment-container">

<h2>₪50 - מחיר הבחינה</h2>

<p>התשלום כולל:</p>

<ul style="list-style: none; padding: 0; margin: 15px 0;">

<li>✅ בחינה תיאורטית מלאה</li>

<li>✅ תוצאות מיידיות</li>

<li>✅ אישור בכתב</li>

<li>✅ זכאות לבחינה מעשית (עם ציון מעל 85%)</li>

</ul>

<div class="payment-methods">

<div class="payment-method" onclick="processPayment('credit')">

<h4>💳 כרטיס אשראי</h4>

<p>תשלום מיידי ובטוח</p>

</div>

<div class="payment-method" onclick="processPayment('paypal')">

<h4>🅿️ PayPal</h4>

<p>תשלום דרך PayPal</p>

</div>

<div class="payment-method" onclick="processPayment('demo')">

<h4>🎯 דמו</h4>

<p>תשלום דמו לבדיקה</p>

</div>

</div>

</div>

</div>

<!-- שלב 2: מילוי פרטים -->

<div id="registrationStep" class="step">

<div class="exam-header">

<h1>🎓 מערכת בחינות מתקדמת</h1>

<p>מערכת מתקדמת למילוי פרטי בחינה עם אפשרות לטופס Google</p>

</div>

<div class="exam-info">

<div class="info-item">

<h3>משך הבחינה</h3>

<p>120 דקות</p>

</div>

<div class="info-item">

<h3>מספר שאלות</h3>

<p>50 שאלות</p>

</div>

<div class="info-item">

<h3>ציון מינימלי</h3>

<p>65 נקודות</p>

</div>

</div>

<div class="form-toggle">

<button class="toggle-btn active" onclick="showForm('local')">טופס מקומי</button>

<button class="toggle-btn" onclick="showForm('google')">טופס Google</button>

</div>

<!-- טופס מקומי -->

<div id="localForm" class="form-section active">

<div class="exam-form">

<form id="examForm">

<div class="form-group">

<label for="studentName">שם מלא:</label>

<input type="text" id="studentName" name="studentName" required

placeholder="הזן את שמך המלא">

</div>

<div class="form-group">

<label for="studentId">מספר זהות:</label>

<input type="text" id="studentId" name="studentId" required

placeholder="הזן מספר זהות" pattern="[0-9]{9}" maxlength="9">

</div>

<div class="form-group">

<label for="studentEmail">כתובת אימייל:</label>

<input type="email" id="studentEmail" name="studentEmail" required

placeholder="הזן כתובת אימייל">

</div>

<div class="form-group">

<label for="phoneNumber">מספר טלפון:</label>

<input type="tel" id="phoneNumber" name="phoneNumber" required

placeholder="הזן מספר טלפון" pattern="[0-9]{10}">

</div>

<div class="form-group">

<label for="course">קורס:</label>

<select id="course" name="course" required>

<option value="">בחר קורס</option>

<option value="mathematics">מתמטיקה</option>

<option value="physics">פיזיקה</option>

<option value="chemistry">כימיה</option>

<option value="biology">ביולוגיה</option>

<option value="computer-science">מדעי המחשב</option>

<option value="english">אנגלית</option>

<option value="hebrew">עברית</option>

<option value="history">היסטוריה</option>

</select>

</div>

<div class="form-group">

<label for="examDate">תאריך בחינה:</label>

<input type="date" id="examDate" name="examDate" required>

</div>

<div class="form-group">

<label for="examTime">שעת בחינה:</label>

<input type="time" id="examTime" name="examTime" required>

</div>

<div class="error-message" id="errorMessage"></div>

<div class="button-group">

<button type="submit" class="btn btn-primary">התחל בחינה</button>

<button type="button" class="btn btn-secondary" onclick="clearForm()">נקה טופס</button>

<button type="button" class="btn btn-demo" onclick="fillDemo()">מלא דמו</button>

<button type="button" class="btn btn-success" onclick="setCurrentDateTime()">תאריך ושעה נוכחיים</button>

</div>

</form>

<div class="success-message" id="successMessage">

<h3>🎉 הרשמה הושלמה בהצלחה!</h3>

<p>הבחינה תתחיל בעוד רגע...</p>

</div>

</div>

</div>

<!-- טופס Google -->

<div id="googleForm" class="form-section">

<div class="google-form-container">

<div class="form-notice" style="text-align: center; padding: 30px; background: linear-gradient(135deg, #667eea 0%, #764ba2 100%); border-radius: 15px; color: white; margin-bottom: 20px;">

<h3 style="margin-bottom: 15px;">🔗 טופס Google Forms</h3>

<p style="margin-bottom: 10px; font-size: 1.1em;">לחץ על הכפתור כדי לפתוח את הטופס עם הנתונים שלך</p>

<button onclick="redirectToGoogleForm()"

class="btn btn-primary"

style="display: inline-block; text-decoration: none; margin-top: 10px; padding: 15px 30px; background: white; color: #333; border-radius: 25px; font-weight: bold; box-shadow: 0 5px 15px rgba(0,0,0,0.2);">

🚀 פתח טופס Google עם הנתונים שלך

</button>

</div>

</div>

</div>

</div>

</div>

<script>

let startTime;

let examDuration = 120 \* 60; // 120 minutes in seconds

let timerInterval;

let currentStep = 'payment';

// Payment processing

function processPayment(method) {

showPaymentProcessing(method);

// Simulate payment processing

setTimeout(() => {

completePayment(method);

}, 3000);

}

function showPaymentProcessing(method) {

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

paymentStep.innerHTML = `

<div class="exam-header">

<h1>💳 מעבד תשלום</h1>

<p>התשלום מתבצע...</p>

</div>

<div class="payment-processing">

<div class="spinner"></div>

<h3>מעבד תשלום דרך ${getPaymentMethodName(method)}</h3>

<p>אנא המתן, התהליך אורך מספר שניות...</p>

</div>

`;

}

function getPaymentMethodName(method) {

const methods = {

'credit': 'כרטיס אשראי',

'paypal': 'PayPal',

'demo': 'דמו'

};

return methods[method] || method;

}

function completePayment(method) {

// Save payment info

const paymentData = {

method: method,

amount: 50,

currency: 'ILS',

timestamp: new Date().toISOString(),

transactionId: 'TXN-' + Date.now()

};

localStorage.setItem('paymentData', JSON.stringify(paymentData));

localStorage.setItem('paymentCompleted', 'true');

// Show success and move to registration

showPaymentSuccess(paymentData);

setTimeout(() => {

showRegistrationStep();

}, 2000);

}

function showPaymentSuccess(paymentData) {

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

paymentStep.innerHTML = `

<div class="exam-header">

<h1>✅ תשלום הושלם בהצלחה!</h1>

<p>מספר עסקה: ${paymentData.transactionId}</p>

</div>

<div class="success-message" style="display: block;">

<h3>🎉 תשלום של ₪${paymentData.amount} בוצע בהצלחה!</h3>

<p>שיטת תשלום: ${getPaymentMethodName(paymentData.method)}</p>

<p>עוברים לשלב מילוי הפרטים...</p>

</div>

`;

}

function showRegistrationStep() {

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

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

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

currentStep = 'registration';

}

// Auto-fill current date and time

function setCurrentDateTime() {

const now = new Date();

const today = now.toISOString().split('T')[0];

const currentTime = now.toTimeString().slice(0, 5);

document.getElementById('examDate').value = today;

document.getElementById('examTime').value = currentTime;

// Visual feedback

const dateInput = document.getElementById('examDate');

const timeInput = document.getElementById('examTime');

[dateInput, timeInput].forEach(input => {

input.style.background = '#e8f5e8';

input.style.transform = 'scale(1.02)';

setTimeout(() => {

input.style.background = '';

input.style.transform = '';

}, 1000);

});

showNotification('📅 תאריך ושעה נוכחיים הוגדרו!');

}

// Google Forms integration

function redirectToGoogleForm() {

if (!document.getElementById('localForm').classList.contains('active')) {

alert('מלא תחילה את הטופס המקומי כדי להעביר את הנתונים ל-Google Forms');

showForm('local');

return;

}

const formData = new FormData(document.getElementById('examForm'));

// Validate form first

if (!validateForm()) {

return;

}

// Google Forms URL with pre-filled parameters

// Note: You need to replace these entry IDs with actual ones from your Google Form

const baseUrl = 'https://docs.google.com/forms/d/e/1FAIpQLSff5VsbJr62znS0MrGxMwBuba2zAVFnG-1N-whGsPlNJstGNA/viewform';

const params = new URLSearchParams({

// Replace these entry IDs with your actual Google Form field IDs

'entry.123456789': formData.get('studentName') || '',

'entry.987654321': formData.get('studentId') || '',

'entry.555666777': formData.get('studentEmail') || '',

'entry.444555666': formData.get('phoneNumber') || '',

'entry.333444555': formData.get('course') || '',

'entry.222333444': formData.get('examDate') || '',

'entry.111222333': formData.get('examTime') || ''

});

const fullUrl = `${baseUrl}?${params.toString()}`;

window.open(fullUrl, '\_blank');

showNotification('🚀 נפתח טופס Google עם הנתונים שלך!');

}

// Form validation and submission

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

e.preventDefault();

if (validateForm()) {

showSuccess();

setTimeout(() => {

startExam();

}, 3000);

}

});

function validateForm() {

const form = document.getElementById('examForm');

const formData = new FormData(form);

// Validate ID number

const studentId = formData.get('studentId');

if (!/^[0-9]{9}$/.test(studentId)) {

showError('מספר זהות חייב להכיל בדיוק 9 ספרות');

return false;

}

// Validate phone number

const phoneNumber = formData.get('phoneNumber');

if (!/^[0-9]{10}$/.test(phoneNumber)) {

showError('מספר טלפון חייב להכיל בדיוק 10 ספרות');

return false;

}

// Validate exam date (not in the past)

const examDate = new Date(formData.get('examDate'));

const today = new Date();

today.setHours(0, 0, 0, 0);

if (examDate < today) {

showError('תאריך הבחינה לא יכול להיות בעבר');

return false;

}

hideError();

return true;

}

function showError(message) {

const errorElement = document.getElementById('errorMessage');

errorElement.textContent = message;

errorElement.style.display = 'block';

// Auto-hide after 5 seconds

setTimeout(() => {

hideError();

}, 5000);

}

function hideError() {

const errorElement = document.getElementById('errorMessage');

if (errorElement) {

errorElement.style.display = 'none';

}

}

function showSuccess() {

const formData = new FormData(document.getElementById('examForm'));

const paymentData = JSON.parse(localStorage.getItem('paymentData'));

const userData = {

name: formData.get('studentName'),

id: formData.get('studentId'),

email: formData.get('studentEmail'),

phone: formData.get('phoneNumber'),

course: formData.get('course'),

examDate: formData.get('examDate'),

examTime: formData.get('examTime'),

registrationTime: new Date().toLocaleString('he-IL'),

paymentInfo: paymentData

};

localStorage.setItem('examUserData', JSON.stringify(userData));

// Show success message with user data

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

successMessage.innerHTML = `

<h3>🎉 הרשמה הושלמה בהצלחה!</h3>

<div style="text-align: right; background: rgba(255,255,255,0.2); padding: 15px; border-radius: 10px; margin: 15px 0;">

<p><strong>שם:</strong> ${userData.name}</p>

<p><strong>מספר זהות:</strong> ${userData.id}</p>

<p><strong>קורס:</strong> ${getCourseName(userData.course)}</p>

<p><strong>תאריך בחינה:</strong> ${userData.examDate}</p>

<p><strong>שעת בחינה:</strong> ${userData.examTime}</p>

<p><strong>זמן הרשמה:</strong> ${userData.registrationTime}</p>

<p><strong>מספר עסקה:</strong> ${paymentData.transactionId}</p>

</div>

<p><strong>הבחינה תתחיל בעוד רגע...</p>

`;

successMessage.style.display = 'block';

// Scroll to success message

successMessage.scrollIntoView({ behavior: 'smooth' });

}

function getCourseName(courseValue) {

const courses = {

'mathematics': 'מתמטיקה',

'physics': 'פיזיקה',

'chemistry': 'כימיה',

'biology': 'ביולוגיה',

'computer-science': 'מדעי המחשב',

'english': 'אנגלית',

'hebrew': 'עברית',

'history': 'היסטוריה'

};

return courses[courseValue] || courseValue;

}

function clearForm() {

document.getElementById('examForm').reset();

hideError();

showNotification('🧹 הטופס נוקה בהצלחה!');

}

function fillDemo() {

const demoData = {

'studentName': 'יוסי כהן',

'studentId': '123456789',

'studentEmail': 'yossi@example.com',

'phoneNumber': '0501234567',

'course': 'mathematics',

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

'examTime': '14:00'

};

Object.keys(demoData).forEach(key => {

const element = document.getElementById(key);

if (element) {

element.value = demoData[key];

element.style.background = '#e8f5e8';

element.style.transform = 'scale(1.02)';

setTimeout(() => {

element.style.background = '';

element.style.transform = '';

}, 1000);

}

});

showNotification('🎯 נתוני דמו הוכנסו בהצלחה!');

}

function showForm(formType) {

// Update toggle buttons

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

btn.classList.remove('active');

});

if (formType === 'local') {

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

document.getElementById('googleForm').classList.remove('active');

document.querySelector('.toggle-btn').classList.add('active');

} else {

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

document.getElementById('localForm').classList.remove('active');

document.querySelectorAll('.toggle-btn')[1].classList.add('active');

}

}

function showNotification(message) {

let notification = document.getElementById('notification');

if (!notification) {

notification = document.createElement('div');

notification.id = 'notification';

notification.className = 'notification';

document.body.appendChild(notification);

}

notification.textContent = message;

notification.style.display = 'block';

notification.style.opacity = '1';

setTimeout(() => {

notification.style.opacity = '0';

setTimeout(() => {

notification.style.display = 'none';

}, 300);

}, 3000);

}

function startExam() {

const userData = JSON.parse(localStorage.getItem('examUserData'));

// Hide registration step

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

// Show exam interface

showExamInterface(userData);

// Start timer

startTimer();

currentStep = 'exam';

}

function showExamInterface(userData) {

const container = document.getElementById('mainContainer');

container.innerHTML = `

<div class="exam-header">

<h1>🎓 בחינה בקורס: ${getCourseName(userData.course)}</h1>

<p>שם הנבחן: ${userData.name} | מספר זהות: ${userData.id}</p>

</div>

<div class="exam-info">

<div class="info-item">

<h3>זמן נותר</h3>

<p id="remainingTime">120:00</p>

</div>

<div class="info-item">

<h3>שאלה נוכחית</h3>

<p id="currentQuestion">1 / 50</p>

</div>

<div class="info-item">

<h3>שאלות שנענו</h3>

<p id="answeredQuestions">0 / 50</p>

</div>

</div>

<div class="exam-content">

<div class="question-container">

<h3 id="questionTitle">שאלה 1:</h3>

<p id="questionText">מה התשובה הנכונה לשאלה הזו?</p>

<div class="answer-options">

<label class="answer-option">

<input type="radio" name="answer" value="a">

<span>א. תשובה ראשונה</span>

</label>

<label class="answer-option">

<input type="radio" name="answer" value="b">

<span>ב. תשובה שנייה</span>

</label>

<label class="answer-option">

<input type="radio" name="answer" value="c">

<span>ג. תשובה שלישית</span>

</label>

<label class="answer-option">

<input type="radio" name="answer" value="d">

<span>ד. תשובה רביעית</span>

</label>

</div>

<div class="question-navigation">

<button id="prevBtn" class="btn btn-secondary" onclick="previousQuestion()">שאלה קודמת</button>

<button id="nextBtn" class="btn btn-primary" onclick="nextQuestion()">שאלה הבאה</button>

<button id="submitBtn" class="btn btn-success" onclick="submitExam()" style="display: none;">סיים בחינה</button>

</div>

</div>

</div>

<div class="exam-progress">

<div class="progress-bar">

<div class="progress-fill" id="progressFill"></div>

</div>

<p class="progress-text">התקדמות: <span id="progressText">0%</span></p>

</div>

`;

// Add CSS for exam interface

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

style.textContent = `

.answer-options {

display: flex;

flex-direction: column;

gap: 15px;

margin: 20px 0;

}

.answer-option {

display: flex;

align-items: center;

padding: 15px;

background: #f9f9f9;

border-radius: 10px;

border: 2px solid #e0e0e0;

cursor: pointer;

transition: all 0.3s ease;

}

.answer-option:hover {

background: #e8f5e8;

border-color: #667eea;

}

.answer-option input[type="radio"] {

margin-left: 10px;

transform: scale(1.2);

}

.answer-option span {

font-size: 1.1em;

color: #333;

}

.question-navigation {

display: flex;

justify-content: space-between;

margin-top: 30px;

gap: 15px;

}

.exam-progress {

margin-top: 30px;

text-align: center;

}

.progress-bar {

width: 100%;

height: 20px;

background: #e0e0e0;

border-radius: 10px;

overflow: hidden;

margin-bottom: 10px;

}

.progress-fill {

height: 100%;

background: linear-gradient(45deg, #667eea, #764ba2);

border-radius: 10px;

width: 0%;

transition: width 0.3s ease;

}

.progress-text {

font-size: 1.1em;

color: #666;

font-weight: bold;

}

`;

document.head.appendChild(style);

// Initialize exam variables

initializeExam();

}

function initializeExam() {

// Sample questions data

window.examQuestions = [

{

question: "מה התשובה הנכונה לשאלה הזו?",

options: {

a: "תשובה ראשונה",

b: "תשובה שנייה",

c: "תשובה שלישית",

d: "תשובה רביעית"

},

correct: "b"

},

{

question: "איזו מהתשובות הבאות נכונה?",

options: {

a: "אפשרות א",

b: "אפשרות ב",

c: "אפשרות ג",

d: "אפשרות ד"

},

correct: "c"

}

// Add more questions as needed

];

// Fill with demo questions to reach 50

while (window.examQuestions.length < 50) {

const questionNum = window.examQuestions.length + 1;

window.examQuestions.push({

question: `שאלה מספר ${questionNum} - בחר את התשובה הנכונה:`,

options: {

a: `תשובה א לשאלה ${questionNum}`,

b: `תשובה ב לשאלה ${questionNum}`,

c: `תשובה ג לשאלה ${questionNum}`,

d: `תשובה ד לשאלה ${questionNum}`

},

correct: ['a', 'b', 'c', 'd'][Math.floor(Math.random() \* 4)]

});

}

window.currentQuestionIndex = 0;

window.userAnswers = {};

window.examStartTime = Date.now();

loadQuestion(0);

}

function loadQuestion(index) {

const question = window.examQuestions[index];

document.getElementById('questionTitle').textContent = `שאלה ${index + 1}:`;

document.getElementById('questionText').textContent = question.question;

document.getElementById('currentQuestion').textContent = `${index + 1} / 50`;

// Clear previous answers

document.querySelectorAll('input[name="answer"]').forEach(input => {

input.checked = false;

});

// Load saved answer if exists

if (window.userAnswers[index]) {

const savedAnswer = window.userAnswers[index];

document.querySelector(`input[name="answer"][value="${savedAnswer}"]`).checked = true;

}

// Update answer options

document.querySelectorAll('.answer-option span').forEach((span, i) => {

const optionKey = ['a', 'b', 'c', 'd'][i];

span.textContent = `${optionKey.toUpperCase()}. ${question.options[optionKey]}`;

});

// Update navigation buttons

document.getElementById('prevBtn').style.display = index === 0 ? 'none' : 'inline-block';

document.getElementById('nextBtn').style.display = index === 49 ? 'none' : 'inline-block';

document.getElementById('submitBtn').style.display = index === 49 ? 'inline-block' : 'none';

// Update progress

updateProgress();

}

function nextQuestion() {

saveCurrentAnswer();

if (window.currentQuestionIndex < 49) {

window.currentQuestionIndex++;

loadQuestion(window.currentQuestionIndex);

}

}

function previousQuestion() {

saveCurrentAnswer();

if (window.currentQuestionIndex > 0) {

window.currentQuestionIndex--;

loadQuestion(window.currentQuestionIndex);

}

}

function saveCurrentAnswer() {

const selectedAnswer = document.querySelector('input[name="answer"]:checked');

if (selectedAnswer) {

window.userAnswers[window.currentQuestionIndex] = selectedAnswer.value;

}

}

function updateProgress() {

const answered = Object.keys(window.userAnswers).length;

const progress = (answered / 50) \* 100;

document.getElementById('answeredQuestions').textContent = `${answered} / 50`;

document.getElementById('progressFill').style.width = `${progress}%`;

document.getElementById('progressText').textContent = `${Math.round(progress)}%`;

}

function submitExam() {

saveCurrentAnswer();

if (confirm('האם אתה בטוח שברצונך לסיים את הבחינה?')) {

finishExam();

}

}

function finishExam() {

// Stop timer

clearInterval(timerInterval);

// Calculate score

let correctAnswers = 0;

Object.keys(window.userAnswers).forEach(questionIndex => {

const userAnswer = window.userAnswers[questionIndex];

const correctAnswer = window.examQuestions[questionIndex].correct;

if (userAnswer === correctAnswer) {

correctAnswers++;

}

});

const score = (correctAnswers / 50) \* 100;

const passed = score >= 65;

// Show results

showExamResults(score, correctAnswers, passed);

}

function showExamResults(score, correctAnswers, passed) {

const userData = JSON.parse(localStorage.getItem('examUserData'));

const examDuration = Math.floor((Date.now() - window.examStartTime) / 1000);

const durationMinutes = Math.floor(examDuration / 60);

const durationSeconds = examDuration % 60;

const container = document.getElementById('mainContainer');

container.innerHTML = `

<div class="exam-header">

<h1>${passed ? '🎉 הבחינה הושלמה בהצלחה!' : '😞 לא עברת את הבחינה'}</h1>

<p>תוצאות הבחינה</p>

</div>

<div class="exam-results">

<div class="result-container ${passed ? 'success' : 'failure'}">

<div class="score-display">

<h2>הציון שלך</h2>

<div class="score-circle">

<span class="score-number">${Math.round(score)}</span>

<span class="score-percent">%</span>

</div>

</div>

<div class="result-details">

<div class="detail-item">

<span class="detail-label">שאלות נכונות:</span>

<span class="detail-value">${correctAnswers} / 50</span>

</div>

<div class="detail-item">

<span class="detail-label">זמן שהושקע:</span>

<span class="detail-value">${durationMinutes}:${durationSeconds.toString().padStart(2, '0')}</span>

</div>

<div class="detail-item">

<span class="detail-label">קורס:</span>

<span class="detail-value">${getCourseName(userData.course)}</span>

</div>

<div class="detail-item">

<span class="detail-label">תאריך:</span>

<span class="detail-value">${new Date().toLocaleDateString('he-IL')}</span>

</div>

</div>

</div>

${passed ? `

<div class="practical-exam-info">

<h3>🎯 זכאות לבחינה מעשית!</h3>

<p>ציון מעל 85% מזכה בבחינה מעשית</p>

<div class="practical-exam-details">

<p><strong>מיקום:</strong> בית הספר לנהיגה - רחוב הרצל 123, תל אביב</p>

<p><strong>זמן:</strong> ימי שני-חמישי, 09:00-17:00</p>

<p><strong>תיאום:</strong> 03-1234567</p>

<p><strong>מסמכים נדרשים:</strong> תעודת זהות, אישור בחינה תיאורטית</p>

</div>

</div>

` : `

<div class="retry-info">

<h3>💪 אל תוותר!</h3>

<p>ניתן לחזור על הבחינה לאחר תשלום נוסף</p>

<button class="btn btn-primary" onclick="location.reload()">חזור לדף הבית</button>

</div>

`}

</div>

<div class="button-group">

<button class="btn btn-success" onclick="downloadCertificate()">הורד אישור</button>

<button class="btn btn-secondary" onclick="printResults()">הדפס תוצאות</button>

<button class="btn btn-primary" onclick="location.reload()">בחינה חדשה</button>

</div>

`;

// Add CSS for results

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

style.textContent = `

.exam-results {

margin-bottom: 30px;

}

.result-container {

background: white;

border-radius: 15px;

padding: 40px;

box-shadow: 0 10px 30px rgba(0, 0, 0, 0.1);

border: 3px solid #e0e0e0;

margin-bottom: 30px;

}

.result-container.success {

border-color: #56ab2f;

}

.result-container.failure {

border-color: #ff6b6b;

}

.score-display {

text-align: center;

margin-bottom: 30px;

}

.score-display h2 {

color: #333;

margin-bottom: 20px;

font-size: 2em;

}

.score-circle {

width: 150px;

height: 150px;

border-radius: 50%;

background: linear-gradient(45deg, #667eea, #764ba2);

display: flex;

align-items: center;

justify-content: center;

margin: 0 auto;

box-shadow: 0 10px 30px rgba(102, 126, 234, 0.3);

}

.score-number {

font-size: 3em;

font-weight: bold;

color: white;

}

.score-percent {

font-size: 1.5em;

color: white;

margin-right: 5px;

}

.result-details {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));

gap: 20px;

}

.detail-item {

display: flex;

justify-content: space-between;

align-items: center;

padding: 15px;

background: #f9f9f9;

border-radius: 10px;

border: 2px solid #e0e0e0;

}

.detail-label {

font-weight: bold;

color: #333;

}

.detail-value {

color: #666;

font-size: 1.1em;

}

.retry-info {

background: linear-gradient(135deg, #ff6b6b, #ee5a24);

color: white;

padding: 25px;

border-radius: 15px;

text-align: center;

box-shadow: 0 5px 15px rgba(255, 107, 107, 0.3);

}

.retry-info h3 {

margin-bottom: 15px;

font-size: 1.8em;

}

`;

document.head.appendChild(style);

// Save results

const examResults = {

score: score,

correctAnswers: correctAnswers,

totalQuestions: 50,

passed: passed,

duration: examDuration,

userData: userData,

completedAt: new Date().toISOString()

};

localStorage.setItem('lastExamResults', JSON.stringify(examResults));

}

function downloadCertificate() {

const results = JSON.parse(localStorage.getItem('lastExamResults'));

// Create certificate content

const certificateContent = `

אישור השלמת בחינה

אישור זה מעיד כי:

${results.userData.name}

מספר זהות: ${results.userData.id}

השלים בהצלחה בחינה בקורס: ${getCourseName(results.userData.course)}

תוצאות הבחינה:

ציון: ${Math.round(results.score)}%

תאריך: ${new Date(results.completedAt).toLocaleDateString('he-IL')}

${results.passed ? 'הבחינה הושלמה בהצלחה!' : 'הבחינה לא הושלמה בהצלחה'}

מערכת בחינות מתקדמת

${new Date().toLocaleDateString('he-IL')}

`;

// Create and download file

const blob = new Blob([certificateContent], { type: 'text/plain;charset=utf-8' });

const url = URL.createObjectURL(blob);

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

a.href = url;

a.download = `אישור\_בחינה\_${results.userData.name}.txt`;

document.body.appendChild(a);

a.click();

document.body.removeChild(a);

URL.revokeObjectURL(url);

showNotification('📄 אישור הבחינה הורד בהצלחה!');

}

function printResults() {

window.print();

}

function startTimer() {

startTime = Date.now();

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

timerInterval = setInterval(() => {

const elapsed = Math.floor((Date.now() - startTime) / 1000);

const remaining = Math.max(0, examDuration - elapsed);

const hours = Math.floor(remaining / 3600);

const minutes = Math.floor((remaining % 3600) / 60);

const seconds = remaining % 60;

const timeString = `${hours.toString().padStart(2, '0')}:${minutes.toString().padStart(2, '0')}:${seconds.toString().padStart(2, '0')}`;

document.getElementById('timeDisplay').textContent = timeString;

// Update remaining time in exam interface

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

const remainingMinutes = Math.floor(remaining / 60);

const remainingSeconds = remaining % 60;

document.getElementById('remainingTime').textContent = `${remainingMinutes}:${remainingSeconds.toString().padStart(2, '0')}`;

}

// Auto-submit when time is up

if (remaining <= 0) {

clearInterval(timerInterval);

alert('הזמן הסתיים! הבחינה תוגש אוטומטית.');

finishExam();

}

}, 1000);

}

// Initialize on page load

document.addEventListener('DOMContentLoaded', function() {

// Check if payment was already completed

const paymentCompleted = localStorage.getItem('paymentCompleted');

if (paymentCompleted === 'true') {

showRegistrationStep();

}

});

// Add event listeners for answer selection

document.addEventListener('change', function(e) {

if (e.target.name === 'answer') {

saveCurrentAnswer();

updateProgress();

}

});

// Prevent form submission on Enter key

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

if (e.key === 'Enter' && e.target.tagName !== 'BUTTON') {

e.preventDefault();

}

});

// Auto-save form data while typing

document.addEventListener('input', function(e) {

if (e.target.form && e.target.form.id === 'examForm') {

const formData = new FormData(e.target.form);

const data = {};

formData.forEach((value, key) => {

data[key] = value;

});

localStorage.setItem('formDraft', JSON.stringify(data));

}

});

// Load saved form data on page load

window.addEventListener('load', function() {

const savedData = localStorage.getItem('formDraft');

if (savedData) {

const data = JSON.parse(savedData);

Object.keys(data).forEach(key => {

const element = document.getElementById(key);

if (element) {

element.value = data[key];

}

});

}

});

</script>

</body>

</html>