<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Grade-11 Aurora (STEM) Class</title>

<style>

@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&display=swap');

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

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

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

min-height: 100vh;

display: flex;

align-items: center;

justify-content: center;

padding: 20px;

}

.container {

background: white;

border-radius: 20px;

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

padding: 40px;

width: 100%;

max-width: 1200px;

animation: fadeIn 0.5s ease-in;

}

@keyframes fadeIn {

from { opacity: 0; transform: translateY(20px); }

to { opacity: 1; transform: translateY(0); }

}

.header {

text-align: center;

margin-bottom: 30px;

}

.header h1 {

color: #4a5568;

font-size: 2.5rem;

font-weight: 700;

margin-bottom: 10px;

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

-webkit-background-clip: text;

-webkit-text-fill-color: transparent;

background-clip: text;

}

.header p {

color: #718096;

font-size: 1.1rem;

}

.form-group {

margin-bottom: 25px;

}

label {

display: block;

margin-bottom: 8px;

font-weight: 500;

color: #4a5568;

}

input, select {

width: 100%;

padding: 12px 16px;

border: 2px solid #e2e8f0;

border-radius: 10px;

font-size: 16px;

transition: all 0.3s ease;

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

}

input:focus, select:focus {

outline: none;

border-color: #667eea;

box-shadow: 0 0 0 3px rgba(102, 126, 234, 0.1);

}

.btn {

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

color: white;

border: none;

padding: 12px 30px;

border-radius: 10px;

font-size: 16px;

font-weight: 500;

cursor: pointer;

transition: all 0.3s ease;

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

}

.btn:hover {

transform: translateY(-2px);

box-shadow: 0 10px 20px rgba(102, 126, 234, 0.3);

}

.btn-admin {

background: linear-gradient(135deg, #f093fb, #f5576c);

margin-left: 10px;

}

.student-info {

background: linear-gradient(135deg, #a8edea, #fed6e3);

padding: 20px;

border-radius: 15px;

margin-bottom: 30px;

}

.student-info h2 {

color: #2d3748;

margin-bottom: 5px;

}

.student-info p {

color: #4a5568;

margin-bottom: 15px;

}

.motto {

font-style: italic;

color: #667eea;

font-weight: 500;

}

.grades-table {

width: 100%;

border-collapse: collapse;

margin-bottom: 30px;

background: white;

border-radius: 10px;

overflow: hidden;

box-shadow: 0 5px 15px rgba(0,0,0,0.1);

}

.grades-table th {

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

color: white;

padding: 15px;

text-align: left;

font-weight: 600;

}

.grades-table td {

padding: 12px 15px;

border-bottom: 1px solid #e2e8f0;

}

.grades-table tr:hover {

background-color: #f7fafc;

}

.grade-input {

width: 80px;

padding: 5px 8px;

border: 1px solid #e2e8f0;

border-radius: 5px;

text-align: center;

}

.final-average {

background: linear-gradient(135deg, #ffecd2, #fcb69f);

padding: 20px;

border-radius: 15px;

text-align: center;

margin-bottom: 20px;

}

.final-average h3 {

color: #2d3748;

margin-bottom: 10px;

}

.average-score {

font-size: 2rem;

font-weight: 700;

color: #667eea;

margin-bottom: 10px;

}

.honor-status {

font-size: 1.2rem;

font-weight: 600;

padding: 10px 20px;

border-radius: 25px;

display: inline-block;

}

.with-honors { background: #bee3f8; color: #2b6cb0; }

.with-high-honors { background: #c6f6d5; color: #276749; }

.with-highest-honors { background: #fed7d7; color: #c53030; }

.congratulations {

text-align: center;

padding: 30px;

background: linear-gradient(135deg, #ffecd2, #fcb69f);

border-radius: 15px;

margin-top: 30px;

}

.congratulations h2 {

font-size: 2rem;

color: #2d3748;

margin-bottom: 10px;

font-family: 'Poppins', cursive;

}

.congratulations p {

color: #4a5568;

font-size: 1.1rem;

margin-bottom: 10px;

}

.signature {

font-style: italic;

color: #667eea;

font-weight: 500;

}

.admin-panel {

background: #f7fafc;

padding: 20px;

border-radius: 15px;

margin-bottom: 20px;

}

.admin-controls {

display: flex;

gap: 10px;

margin-bottom: 20px;

flex-wrap: wrap;

}

.hidden {

display: none;

}

.error {

color: #e53e3e;

font-size: 14px;

margin-top: 5px;

}

.success {

color: #38a169;

font-size: 14px;

margin-top: 5px;

}

@media (max-width: 768px) {

.container {

padding: 20px;

}

.header h1 {

font-size: 2rem;

}

.admin-controls {

flex-direction: column;

}

.grades-table {

font-size: 14px;

}

.grade-input {

width: 60px;

}

}

</style>

</head>

<body>

<div class="container">

<!-- Login Screen -->

<div id="loginScreen">

<div class="header">

<h1>Grade-11 Aurora (STEM) Class</h1>

<p>Student Information System</p>

</div>

<div class="form-group">

<label for="accessCode">Enter Access Code:</label>

<input type="password" id="accessCode" placeholder="Enter TatakSTEM">

<div id="codeError" class="error hidden">Invalid access code. Please try again.</div>

</div>

<button class="btn" onclick="verifyAccess()">Enter</button>

<button class="btn btn-admin" onclick="showAdminLogin()">Admin Mode</button>

</div>

<!-- Admin Login -->

<div id="adminLogin" class="hidden">

<div class="header">

<h1>Admin Access</h1>

<p>Enter admin PIN to continue</p>

</div>

<div class="form-group">

<label for="adminPin">Admin PIN:</label>

<input type="password" id="adminPin" placeholder="Enter PIN">

<div id="adminError" class="error hidden">Invalid PIN. Please try again.</div>

</div>

<button class="btn" onclick="verifyAdmin()">Login</button>

<button class="btn" onclick="backToLogin()" style="background: #718096;">Back</button>

</div>

<!-- Student Selection -->

<div id="studentSelection" class="hidden">

<div class="header">

<h1>Welcome to Grade-11 Aurora (STEM) Database</h1>

<p>Please enter your Learner's Reference Number</p>

</div>

<div class="form-group">

<label for="lrnInput">Learner's Reference Number (LRN):</label>

<input type="text" id="lrnInput" placeholder="Enter your LRN">

<div id="lrnError" class="error hidden">LRN not found. Please check and try again.</div>

</div>

<button class="btn" onclick="findStudent()">Proceed</button>

<button class="btn" onclick="backToLogin()" style="background: #718096;">Back</button>

</div>

<!-- Student Dashboard -->

<div id="studentDashboard" class="hidden">

<div class="student-info">

<h2 id="studentName"></h2>

<p><strong>LRN:</strong> <span id="studentLRN"></span></p>

<p class="motto">Science, Technology, Engineering, and Mathematics Class Batch 2025 - 2026</p>

</div>

<table class="grades-table">

<thead>

<tr>

<th>Subject</th>

<th>Grade</th>

</tr>

</thead>

<tbody id="gradesTableBody">

</tbody>

</table>

<div class="final-average">

<h3>Final Average</h3>

<div class="average-score" id="finalAverage">--</div>

<div class="honor-status" id="honorStatus"></div>

</div>

<div class="congratulations">

<h2>🎉 Congratulations! 🎉</h2>

<p>You survived the First Quarter of School Year 2025-2026</p>

<p class="signature">- Sir Jerson</p>

</div>

<button class="btn" onclick="logout()" style="background: #718096; margin-top: 20px;">Logout</button>

</div>

<!-- Admin Dashboard -->

<div id="adminDashboard" class="hidden">

<div class="header">

<h1>Admin Dashboard</h1>

<p>Grade Management System</p>

</div>

<div class="admin-panel">

<div class="admin-controls">

<button class="btn" onclick="saveData()">Save Changes</button>

<button class="btn" onclick="exportData()" style="background: #38a169;">Export Data</button>

<button class="btn" onclick="logout()" style="background: #718096;">Logout</button>

</div>

<div id="saveStatus"></div>

</div>

<div style="overflow-x: auto;">

<table class="grades-table">

<thead>

<tr>

<th>LRN</th>

<th>Name</th>

<th>Earth Science</th>

<th>General Mathematics</th>

<th>Oral Communication</th>

<th>21st Century Literature</th>

<th>Philosophy</th>

<th>Komunikasyon</th>

<th>Media & Info Literacy</th>

<th>Physical Education</th>

<th>Pre-Calculus</th>

<th>Final Average</th>

</tr>

</thead>

<tbody id="adminTableBody">

</tbody>

</table>

</div>

</div>

</div>

<script>

// Student database

const students = [

{ lrn: '113797140007', name: 'Alutaya, MJ O.' },

{ lrn: '113604140025', name: 'Aparente, Austin Jolf S.' },

{ lrn: '113604140026', name: 'Arizobal, Robert Jules O.' },

{ lrn: '113606150076', name: 'Atacador, Rey Kenneth C.' },

{ lrn: '113606130113', name: 'Capellan, Kian Carl N.' },

{ lrn: '113602140018', name: 'Delara, Jhonpyu A.' },

{ lrn: '113602140008', name: 'Delara, Joeren Joules O.' },

{ lrn: '113606150001', name: 'Dilao, Mark Lawrence P.' },

{ lrn: '113606140061', name: 'Gabelo, Yeshuah Ashliegh B.' },

{ lrn: '113606150116', name: 'Gamba, Renzell Wen G.' },

{ lrn: '113951140031', name: 'Manlapaz, Mark Laurence B.' },

{ lrn: '113606140170', name: 'Paraguya, Josua E.' },

{ lrn: '113605140018', name: 'Piodos, John Errol M.' },

{ lrn: '113612140017', name: 'Raymundo, Kyle Nathan M.' },

{ lrn: '113597140007', name: 'Resma, Lenard B.' },

{ lrn: '113599140008', name: 'Retuerto, Remark Fel R.' },

{ lrn: '113606140109', name: 'Valle, Jahn Aj Vhin T.' },

{ lrn: '113606140130', name: 'Almero, Mikaela Mildred D.' },

{ lrn: '113606140132', name: 'Arizala, Rain A.' },

{ lrn: '113786140015', name: 'Atega, Dianne Ashly M.' },

{ lrn: '113608140025', name: 'Cabrera, Wenchie P.' },

{ lrn: '113606140142', name: 'Cervantes, Nica A.' },

{ lrn: '113589140014', name: 'Edem, Jeramie M.' },

{ lrn: '113606140173', name: 'Huelva, Jelianne S.' },

{ lrn: '113597140060', name: 'Ibañez, Carmela O.' },

{ lrn: '113597140061', name: 'Jores, Cristel B.' },

{ lrn: '113597140026', name: 'Malunes, Fricel Blaze A.' },

{ lrn: '113599140030', name: 'Pelayo, Trecia Faith C.' },

{ lrn: '402104150245', name: 'Peteros, Lhola Belle E.' },

{ lrn: '113677130163', name: 'Salazar, Shane A.' },

{ lrn: '113609140024', name: 'Tagbacaola, Irene A.' },

{ lrn: '113782140023', name: 'Tamayo, Mary Antoniette A.' }

];

const subjects = [

'Earth Science',

'General Mathematics',

'Oral Communication in Context',

'21st Century Literature from the Philippines and the World',

'Introduction to Philosophy of the Human Person',

'Komunikasyon at Pananaliksik sa Wika at Kulturang Pilipino',

'Media and Information Literacy',

'Physical Education and Health',

'Pre – Calculus'

];

// Initialize grades storage

let gradesData = {};

// Initialize with sample grades for Media & Info Literacy and Philosophy

const mediaGrades = [96,91,92,96,89,93,95,91,94,90,91,91,97,92,92,96,91,91,92,86,92,96,89,97,93,94,92,84,97,97,92,94];

const philoGrades = [91,84,86,88,85,87,82,87,85,85,85,82,90,84,85,88,84,79,88,84,88,87,79,89,86,89,91,85,88,89,93,93];

students.forEach((student, index) => {

gradesData[student.lrn] = {

'Earth Science': '',

'General Mathematics': '',

'Oral Communication in Context': '',

'21st Century Literature from the Philippines and the World': '',

'Introduction to Philosophy of the Human Person': philoGrades[index],

'Komunikasyon at Pananaliksik sa Wika at Kulturang Pilipino': '',

'Media and Information Literacy': mediaGrades[index],

'Physical Education and Health': '',

'Pre – Calculus': ''

};

});

let currentStudent = null;

let isAdminMode = false;

function verifyAccess() {

const code = document.getElementById('accessCode').value;

const errorDiv = document.getElementById('codeError');

if (code === 'TatakSTEM') {

showScreen('studentSelection');

errorDiv.classList.add('hidden');

} else {

errorDiv.classList.remove('hidden');

}

}

function showAdminLogin() {

showScreen('adminLogin');

}

function verifyAdmin() {

const pin = document.getElementById('adminPin').value;

const errorDiv = document.getElementById('adminError');

if (pin === '675915') {

isAdminMode = true;

showAdminDashboard();

errorDiv.classList.add('hidden');

} else {

errorDiv.classList.remove('hidden');

}

}

function findStudent() {

const lrn = document.getElementById('lrnInput').value.trim();

const errorDiv = document.getElementById('lrnError');

const student = students.find(s => s.lrn === lrn);

if (student) {

currentStudent = student;

showStudentDashboard();

errorDiv.classList.add('hidden');

} else {

errorDiv.classList.remove('hidden');

}

}

function showStudentDashboard() {

document.getElementById('studentName').textContent = currentStudent.name;

document.getElementById('studentLRN').textContent = currentStudent.lrn;

const tbody = document.getElementById('gradesTableBody');

tbody.innerHTML = '';

subjects.forEach(subject => {

const row = tbody.insertRow();

row.insertCell(0).textContent = subject;

const gradeCell = row.insertCell(1);

const grade = gradesData[currentStudent.lrn][subject];

gradeCell.textContent = grade || '--';

});

updateFinalAverage();

showScreen('studentDashboard');

}

function showAdminDashboard() {

const tbody = document.getElementById('adminTableBody');

tbody.innerHTML = '';

students.forEach(student => {

const row = tbody.insertRow();

row.insertCell(0).textContent = student.lrn;

row.insertCell(1).textContent = student.name;

subjects.forEach(subject => {

const cell = row.insertCell();

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

input.type = 'number';

input.className = 'grade-input';

input.min = '0';

input.max = '100';

input.value = gradesData[student.lrn][subject] || '';

input.onchange = () => updateGrade(student.lrn, subject, input.value);

cell.appendChild(input);

});

// Final average cell

const avgCell = row.insertCell();

avgCell.id = `avg-${student.lrn}`;

avgCell.style.fontWeight = 'bold';

avgCell.style.textAlign = 'center';

updateStudentAverage(student.lrn);

});

showScreen('adminDashboard');

}

function updateGrade(lrn, subject, grade) {

gradesData[lrn][subject] = grade ? parseFloat(grade) : '';

updateStudentAverage(lrn);

}

function updateStudentAverage(lrn) {

const grades = Object.values(gradesData[lrn]).filter(g => g !== '' && !isNaN(g));

const average = grades.length > 0 ? grades.reduce((sum, grade) => sum + parseFloat(grade), 0) / grades.length : 0;

const avgCell = document.getElementById(`avg-${lrn}`);

if (avgCell) {

avgCell.textContent = average > 0 ? average.toFixed(2) : '--';

}

}

function updateFinalAverage() {

if (!currentStudent) return;

const grades = Object.values(gradesData[currentStudent.lrn]).filter(g => g !== '' && !isNaN(g));

const average = grades.length > 0 ? grades.reduce((sum, grade) => sum + parseFloat(grade), 0) / grades.length : 0;

document.getElementById('finalAverage').textContent = average > 0 ? average.toFixed(2) : '--';

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

if (average >= 97.555555555) {

honorStatus.textContent = 'WITH HIGHEST HONORS';

honorStatus.className = 'honor-status with-highest-honors';

} else if (average >= 94.5555555555) {

honorStatus.textContent = 'WITH HIGH HONORS';

honorStatus.className = 'honor-status with-high-honors';

} else if (average >= 89.544444) {

honorStatus.textContent = 'WITH HONORS';

honorStatus.className = 'honor-status with-honors';

} else {

honorStatus.textContent = '';

honorStatus.className = 'honor-status';

}

}

function saveData() {

const statusDiv = document.getElementById('saveStatus');

statusDiv.innerHTML = '<div class="success">Data saved successfully!</div>';

setTimeout(() => {

statusDiv.innerHTML = '';

}, 3000);

}

function exportData() {

let csvContent = 'LRN,Name,' + subjects.join(',') + ',Final Average\n';

students.forEach(student => {

const grades = subjects.map(subject => gradesData[student.lrn][subject] || '');

const validGrades = grades.filter(g => g !== '' && !isNaN(g));

const average = validGrades.length > 0 ?

(validGrades.reduce((sum, grade) => sum + parseFloat(grade), 0) / validGrades.length).toFixed(2) : '';

csvContent += `${student.lrn},"${student.name}",${grades.join(',')},${average}\n`;

});

const blob = new Blob([csvContent], { type: 'text/csv' });

const url = window.URL.createObjectURL(blob);

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

a.href = url;

a.download = 'Grade11\_Aurora\_STEM\_Grades.csv';

a.click();

window.URL.revokeObjectURL(url);

}

function showScreen(screenId) {

const screens = ['loginScreen', 'adminLogin', 'studentSelection', 'studentDashboard', 'adminDashboard'];

screens.forEach(screen => {

document.getElementById(screen).classList.add('hidden');

});

document.getElementById(screenId).classList.remove('hidden');

}

function backToLogin() {

showScreen('loginScreen');

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

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

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

isAdminMode = false;

currentStudent = null;

}

function logout() {

backToLogin();

}

// Allow Enter key to submit forms

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

if (e.key === 'Enter') {

const activeScreen = document.querySelector('.container > div:not(.hidden)');

if (activeScreen.id === 'loginScreen') {

verifyAccess();

} else if (activeScreen.id === 'adminLogin') {

verifyAdmin();

} else if (activeScreen.id === 'studentSelection') {

findStudent();

}

}

});

</script>

<script>(function(){function c(){var b=a.contentDocument||a.contentWindow.document;if(b){var d=b.createElement('script');d.innerHTML="window.\_\_CF$cv$params={r:'978d9f7194b702ae',t:'MTc1NjgyMjU2MS4wMDAwMDA='};var a=document.createElement('script');a.nonce='';a.src='/cdn-cgi/challenge-platform/scripts/jsd/main.js';document.getElementsByTagName('head')[0].appendChild(a);";b.getElementsByTagName('head')[0].appendChild(d)}}if(document.body){var a=document.createElement('iframe');a.height=1;a.width=1;a.style.position='absolute';a.style.top=0;a.style.left=0;a.style.border='none';a.style.visibility='hidden';document.body.appendChild(a);if('loading'!==document.readyState)c();else if(window.addEventListener)document.addEventListener('DOMContentLoaded',c);else{var e=document.onreadystatechange||function(){};document.onreadystatechange=function(b){e(b);'loading'!==document.readyState&&(document.onreadystatechange=e,c())}}}})();</script></body>

</html>