// models/Classroom.js

const mongoose = require('mongoose');

const ClassroomSchema = new mongoose.Schema({

name: {

type: String,

required: true

},

subject: {

type: String,

required: true

},

description: {

type: String

},

teacher: {

type: mongoose.Schema.Types.ObjectId,

ref: 'User',

required: true

},

students: [{

type: mongoose.Schema.Types.ObjectId,

ref: 'User'

}],

date: {

type: Date,

default: Date.now

}

});

module.exports = mongoose.model('Classroom', ClassroomSchema);

// models/User.js

const mongoose = require('mongoose');

const bcrypt = require('bcryptjs');

const UserSchema = new mongoose.Schema({

name: {

type: String,

required: true

},

email: {

type: String,

required: true,

unique: true

},

password: {

type: String,

required: true

},

role: {

type: String,

enum: ['student', 'teacher', 'admin'],

default: 'student'

},

date: {

type: Date,

default: Date.now

}

});

module.exports = mongoose.model('User', UserSchema);

// middleware/auth.js

const jwt = require('jsonwebtoken');

module.exports = function(req, res, next) {

// Get token from header

const token = req.header('x-auth-token');

// Check if no token

if (!token) {

return res.status(401).json({ msg: 'No token, authorization denied' });

}

// Verify token

try {

const decoded = jwt.verify(token, process.env.JWT\_SECRET);

req.user = decoded.user;

next();

} catch (err) {

res.status(401).json({ msg: 'Token is not valid' });

}

};

// routes/ai.js

const express = require('express');

const router = express.Router();

const auth = require('../middleware/auth');

const { Configuration, OpenAIApi } = require('openai');

const configuration = new Configuration({

apiKey: process.env.OPENAI\_API\_KEY,

});

const openai = new OpenAIApi(configuration);

// AI Chat

router.post('/chat', auth, async (req, res) => {

try {

const { message } = req.body;

const response = await openai.createChatCompletion({

model: "gpt-3.5-turbo",

messages: [

{

role: "system",

content: "You are a helpful AI teaching assistant for a classroom platform called EduSync. Answer questions about the class material, help with homework, and provide educational support."

},

{

role: "user",

content: message

}

]

});

res.json({ reply: response.data.choices[0].message.content });

} catch (err) {

console.error(err.message);

res.status(500).send('Server error');

}

});

// Generate Summary

router.post('/summarize', auth, async (req, res) => {

try {

const { text } = req.body;

const response = await openai.createChatCompletion({

model: "gpt-3.5-turbo",

messages: [

{

role: "system",

content: "You are an AI teaching assistant. Summarize the following class notes in a concise way that highlights key concepts and makes them easy to understand for students."

},

{

role: "user",

content: text

}

]

});

res.json({ summary: response.data.choices[0].message.content });

} catch (err) {

console.error(err.message);

res.status(500).send('Server error');

}

});

// Generate Quiz

router.post('/quiz', auth, async (req, res) => {

try {

const { text } = req.body;

const response = await openai.createChatCompletion({

model: "gpt-3.5-turbo",

messages: [

{

role: "system",

content: "You are an AI teaching assistant. Generate a 5-question multiple choice quiz based on the following class material. Format the response as JSON with questions, options, and correct answers."

},

{

role: "user",

content: text

}

],

response\_format: { type: "json\_object" }

});

const quiz = JSON.parse(response.data.choices[0].message.content);

res.json(quiz);

} catch (err) {

console.error(err.message);

res.status(500).send('Server error');

}

});

module.exports = router;

// routes/classroom.js

const express = require('express');

const router = express.Router();

const auth = require('../middleware/auth');

const Classroom = require('../models/Classroom');

// Get all classrooms for user

router.get('/', auth, async (req, res) => {

try {

const classrooms = await Classroom.find({

$or: [

{ teacher: req.user.id },

{ students: req.user.id }

]

}).populate('teacher', ['name', 'email']);

res.json(classrooms);

} catch (err) {

console.error(err.message);

res.status(500).send('Server error');

}

});

// Create new classroom

router.post('/', auth, async (req, res) => {

try {

const { name, subject, description } = req.body;

const newClassroom = new Classroom({

name,

subject,

description,

teacher: req.user.id

});

const classroom = await newClassroom.save();

res.json(classroom);

} catch (err) {

console.error(err.message);

res.status(500).send('Server error');

}

});

// Get classroom details

router.get('/:id', auth, async (req, res) => {

try {

const classroom = await Classroom.findById(req.params.id)

.populate('teacher', ['name', 'email'])

.populate('students', ['name', 'email']);

if (!classroom) return res.status(404).json({ msg: 'Classroom not found' });

res.json(classroom);

} catch (err) {

console.error(err.message);

res.status(500).send('Server error');

}

});

module.exports = router;

// routes/auth.js

const express = require('express');

const router = express.Router();

const bcrypt = require('bcryptjs');

const jwt = require('jsonwebtoken');

const User = require('../models/User');

// Register

router.post('/register', async (req, res) => {

try {

const { email, password, name, role } = req.body;

// Check if user exists

let user = await User.findOne({ email });

if (user) return res.status(400).json({ msg: 'User already exists' });

// Create new user

user = new User({

name,

email,

password,

role

});

// Hash password

const salt = await bcrypt.genSalt(10);

user.password = await bcrypt.hash(password, salt);

await user.save();

// Create JWT

const payload = {

user: {

id: user.id

}

};

jwt.sign(

payload,

process.env.JWT\_SECRET,

{ expiresIn: '5h' },

(err, token) => {

if (err) throw err;

res.json({ token });

}

);

} catch (err) {

console.error(err.message);

res.status(500).send('Server error');

}

});

// Login

router.post('/login', async (req, res) => {

try {

const { email, password } = req.body;

// Check if user exists

let user = await User.findOne({ email });

if (!user) return res.status(400).json({ msg: 'Invalid credentials' });

// Validate password

const isMatch = await bcrypt.compare(password, user.password);

if (!isMatch) return res.status(400).json({ msg: 'Invalid credentials' });

// Create JWT

const payload = {

user: {

id: user.id

}

};

jwt.sign(

payload,

process.env.JWT\_SECRET,

{ expiresIn: '5h' },

(err, token) => {

if (err) throw err;

res.json({ token, user: { id: user.id, name: user.name, role: user.role } });

}

);

} catch (err) {

console.error(err.message);

res.status(500).send('Server error');

}

});

module.exports = router;

// server.js

const express = require('express');

const cors = require('cors');

const bodyParser = require('body-parser');

const app = express();

// Middleware

app.use(cors());

app.use(bodyParser.json());

// API Routes

const authRoutes = require('./routes/auth');

const classroomRoutes = require('./routes/classroom');

const aiRoutes = require('./routes/ai');

app.use('/api/auth', authRoutes);

app.use('/api/classroom', classroomRoutes);

app.use('/api/ai', aiRoutes);

// Start server

const PORT = process.env.PORT || 5000;

app.listen(PORT, () => {

console.log(Server running on port ${PORT});

});

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>EduSync - Smart Classroom Platform</title>

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

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">

<style>

.hand-raised-animation {

animation: pulse 2s infinite;

}

@keyframes pulse {

0% { transform: scale(1); }

50% { transform: scale(1.1); }

100% { transform: scale(1); }

}

.video-grid {

display: grid;

grid-template-columns: repeat(auto-fill, minmax(200px, 1fr));

gap: 1rem;

}

.note-card {

transition: all 0.3s ease;

}

.note-card:hover {

transform: translateY(-5px);

box-shadow: 0 10px 20px rgba(0,0,0,0.1);

}

.chat-message {

max-height: 400px;

overflow-y: auto;

}

.quiz-option:hover {

background-color: #f0f4ff;

}

.sidebar {

transition: all 0.3s ease;

}

@media (max-width: 768px) {

.sidebar {

transform: translateX(-100%);

position: absolute;

z-index: 10;

height: 100vh;

}

.sidebar.open {

transform: translateX(0);

}

}

</style>

</head>

<body class="bg-gray-50 font-sans">

<div class="flex h-screen overflow-hidden">

<!-- AI Chatbot Panel (hidden by default) -->

<div id="chatbotPanel" class="hidden absolute right-0 top-0 h-full w-80 bg-white border-l border-gray-200 z-20 flex flex-col">

<div class="p-4 border-b border-gray-200 flex items-center">

<h3 class="font-semibold">AI Teaching Assistant</h3>

<button id="closeChatbot" class="ml-auto">

<i class="fas fa-times"></i>

</button>

</div>

<div class="flex-1 p-4 overflow-y-auto chat-message" id="chatMessages">

<div class="chat-bubble bg-blue-100 p-3 rounded-lg mb-2">

<p>Hello! I'm your AI teaching assistant. How can I help you today?</p>

</div>

</div>

<div class="p-4 border-t border-gray-200">

<div class="flex">

<input type="text" id="chatInput" placeholder="Ask me anything..." class="flex-1 border border-gray-300 rounded-l-lg px-4 py-2 focus:outline-none focus:ring-2 focus:ring-blue-500">

<button id="sendChat" class="bg-blue-600 text-white px-4 py-2 rounded-r-lg hover:bg-blue-700">

<i class="fas fa-paper-plane"></i>

</button>

</div>

</div>

</div>

<!-- Sidebar -->

<div class="sidebar bg-white w-64 border-r border-gray-200 flex flex-col">

<div class="p-4 border-b border-gray-200 flex items-center">

<div class="w-10 h-10 rounded-full bg-blue-600 flex items-center justify-center text-white font-bold text-xl">E</div>

<span class="ml-3 font-semibold text-lg">EduSync</span>

<button id="closeSidebar" class="ml-auto md:hidden">

<i class="fas fa-times"></i>

</button>

</div>

<div class="flex-1 overflow-y-auto">

<nav class="p-2">

<a href="#" class="block py-2 px-3 rounded hover:bg-blue-50 text-blue-600 font-medium">

<i class="fas fa-home mr-2"></i> Home

</a>

<a href="#" class="block py-2 px-3 rounded hover:bg-blue-50 text-gray-700">

<i class="fas fa-calendar-alt mr-2"></i> Calendar

</a>

<a href="#" class="block py-2 px-3 rounded hover:bg-blue-50 text-gray-700">

<i class="fas fa-tasks mr-2"></i> Assignments

</a>

<a href="#" class="block py-2 px-3 rounded hover:bg-blue-50 text-gray-700">

<i class="fas fa-users mr-2"></i> People

</a>

<a href="#" class="block py-2 px-3 rounded hover:bg-blue-50 text-gray-700">

<i class="fas fa-chart-bar mr-2"></i> Grades

</a>

</nav>

<div class="p-4 border-t border-gray-200">

<h3 class="font-medium text-gray-500 uppercase text-xs tracking-wider mb-2">Classes</h3>

<a href="#" class="block py-2 px-3 rounded hover:bg-blue-50 text-gray-700">

<i class="fas fa-book mr-2"></i> Math 101

</a>

<a href="#" class="block py-2 px-3 rounded hover:bg-blue-50 text-gray-700">

<i class="fas fa-flask mr-2"></i> Science

</a>

</div>

</div>

</div>

<!-- Main Content -->

<div class="flex-1 flex flex-col overflow-hidden">

<!-- Top Bar -->

<header class="bg-white border-b border-gray-200 p-4 flex items-center">

<button id="menuToggle" class="md:hidden mr-4">

<i class="fas fa-bars"></i>

</button>

<h1 class="text-xl font-semibold">Math 101 - Algebra Basics</h1>

<div class="ml-auto flex space-x-4">

<button id="openChatbot" class="bg-blue-600 text-white px-4 py-2 rounded-lg hover:bg-blue-700 flex items-center">

<i class="fas fa-robot mr-2"></i> AI Assistant

</button>

<button class="bg-white border border-gray-300 px-4 py-2 rounded-lg hover:bg-gray-50 flex items-center">

<i class="fas fa-cog mr-2"></i> Settings

</button>

</div>

</header>

<!-- Classroom Content -->

<main class="flex-1 overflow-y-auto p-6 bg-gray-50">

<!-- Notes Section -->

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

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

<h2 class="text-lg font-semibold">Class Notes</h2>

<div class="flex space-x-2">

<button id="summarizeBtn" class="bg-blue-600 text-white px-4 py-2 rounded-lg hover:bg-blue-700 flex items-center">

<i class="fas fa-file-contract mr-2"></i> Summarize

</button>

<button id="generateQuizBtn" class="bg-green-600 text-white px-4 py-2 rounded-lg hover:bg-green-700 flex items-center">

<i class="fas fa-question-circle mr-2"></i> Generate Quiz

</button>

</div>

</div>

<div id="notesContent" class="prose max-w-none">

<p>Algebra is the branch of mathematics that uses letters and symbols to represent numbers and quantities in formulas and equations. The basics include understanding variables, constants, expressions, equations, and how to manipulate them.</p>

<!-- More note content would go here -->

</div>

<div id="summaryContent" class="hidden mt-4 p-4 bg-blue-50 rounded-lg">

<h3 class="font-medium mb-2">AI Summary:</h3>

<p id="summaryText"></p>

</div>

</div>

<!-- Video Grid -->

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

<h2 class="text-lg font-semibold mb-4">Class Participants</h2>

<div class="video-grid">

<!-- Teacher Video -->

<div class="bg-gray-100 rounded-lg overflow-hidden relative">

<div class="aspect-w-16 aspect-h-9 bg-gray-300 flex items-center justify-center">

<i class="fas fa-user-tie text-4xl text-gray-500"></i>

</div>

<div class="p-2 bg-white">

<p class="font-medium">Teacher</p>

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

<span class="text-sm text-gray-500">Online</span>

<button class="text-blue-600">

<i class="fas fa-volume-up"></i>

</button>

</div>

</div>

</div>

<!-- Student Videos -->

<div class="bg-gray-100 rounded-lg overflow-hidden relative">

<div class="aspect-w-16 aspect-h-9 bg-gray-300 flex items-center justify-center">

<i class="fas fa-user text-4xl text-gray-500"></i>

</div>

<div class="p-2 bg-white">

<p class="font-medium">Student 1</p>

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

<span class="text-sm text-gray-500">Online</span>

<button class="text-blue-600">

<i class="fas fa-volume-up"></i>

</button>

</div>

</div>

<div class="absolute bottom-2 right-2">

<button class="hand-raised-animation bg-yellow-400 text-white rounded-full w-8 h-8 flex items-center justify-center">

<i class="fas fa-hand-paper"></i>

</button>

</div>

</div>

</div>

</div>

<!-- Quiz Generator Section -->

<div id="quizSection" class="hidden bg-white rounded-lg shadow p-6 mb-6">

<h2 class="text-lg font-semibold mb-4">Generated Quiz</h2>

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

<!-- Quiz questions will be inserted here by JavaScript -->

</div>

<button id="submitQuiz" class="mt-4 bg-blue-600 text-white px-4 py-2 rounded-lg hover:bg-blue-700">

Submit Quiz

</button>

</div>

<!-- Scoreboard Section -->

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

<h2 class="text-lg font-semibold mb-4">Quiz Results</h2>

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

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

<tr>

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

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

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

</tr>

</thead>

<tbody id="scoreboardBody" class="bg-white divide-y divide-gray-200">

<!-- Scores will be inserted here by JavaScript -->

</tbody>

</table>

</div>

</main>

</div>

</body>

</html>  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>EduConnect - Smart Classroom Platform</title>

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

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">

<style>

/\* Custom CSS for features Tailwind can't handle easily \*/

.hand-raised {

animation: pulse 1.5s infinite;

}

@keyframes pulse {

0% { transform: scale(1); }

50% { transform: scale(1.1); }

100% { transform: scale(1); }

}

.camera-feed {

border-radius: 8px;

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

transition: all 0.3s ease;

}

.camera-feed:hover {

transform: scale(1.02);

box-shadow: 0 10px 15px rgba(0, 0, 0, 0.1);

}

.chat-message {

max-width: 70%;

word-wrap: break-word;

}

.summary-container {

transition: all 0.3s ease;

max-height: 0;

overflow: hidden;

}

.summary-container.open {

max-height: 500px;

}

/\* Custom scrollbar \*/

::-webkit-scrollbar {

width: 8px;

}

::-webkit-scrollbar-track {

background: #f1f1f1;

}

::-webkit-scrollbar-thumb {

background: #888;

border-radius: 4px;

}

::-webkit-scrollbar-thumb:hover {

background: #555;

}

</style>

</head>

<body class="bg-gray-100 font-sans">

<div class="flex h-screen overflow-hidden">

<!-- Sidebar -->

<div class="w-64 bg-blue-800 text-white flex flex-col">

<div class="p-4 flex items-center border-b border-blue-700">

<i class="fas fa-graduation-cap text-2xl mr-3"></i>

<h1 class="text-xl font-bold">EduConnect</h1>

</div>

<nav class="flex-1 p-4 space-y-2">

<a href="#" class="block py-2 px-4 rounded hover:bg-blue-700 transition flex items-center">

<i class="fas fa-home mr-3"></i> Dashboard

</a>

<a href="#" class="block py-2 px-4 rounded bg-blue-700 transition flex items-center">

<i class="fas fa-chalkboard-teacher mr-3"></i> Class

</a>

<a href="#" class="block py-2 px-4 rounded hover:bg-blue-700 transition flex items-center">

<i class="fas fa-calendar-alt mr-3"></i> Calendar

</a>

<a href="#" class="block py-2 px-4 rounded hover:bg-blue-700 transition flex items-center">

<i class="fas fa-tasks mr-3"></i> Assignments

</a>

<a href="#" class="block py-2 px-4 rounded hover:bg-blue-700 transition flex items-center">

<i class="fas fa-users mr-3"></i> People

</a>

<a href="#" class="block py-2 px-4 rounded hover:bg-blue-700 transition flex items-center">

<i class="fas fa-chart-bar mr-3"></i> Grades

</a>

</nav>

<div class="p-4 border-t border-blue-700">

<div class="flex items-center">

<img src="https://randomuser.me/api/portraits/women/44.jpg" alt="Profile" class="w-10 h-10 rounded-full mr-3">

<div>

<p class="font-medium">Ms. Sarah Johnson</p>

<p class="text-xs text-blue-200">Teacher</p>

</div>

</div>

</div>

</div>

<!-- Main Content -->

<div class="flex-1 flex flex-col overflow-hidden">

<!-- Top Navigation -->

<header class="bg-white shadow-sm py-3 px-6 flex items-center justify-between">

<h2 class="text-xl font-semibold text-gray-800">Mathematics 101 - Grade 10</h2>

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

<button class="p-2 rounded-full hover:bg-gray-100">

<i class="fas fa-bell text-gray-600"></i>

<span class="notification-badge hidden absolute top-0 right-0 bg-red-500 text-white text-xs rounded-full h-5 w-5 flex items-center justify-center">3</span>

</button>

<button class="p-2 rounded-full hover:bg-gray-100">

<i class="fas fa-cog text-gray-600"></i>

</button>

</div>

</header>

<!-- Main Content Area -->

<main class="flex-1 overflow-y-auto p-6 grid grid-cols-1 lg:grid-cols-3 gap-6">

<!-- Left Column - Class Content -->

<div class="lg:col-span-2 space-y-6">

<!-- Class Announcement -->

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

<div class="flex items-start">

<img src="https://randomuser.me/api/portraits/women/44.jpg" alt="Teacher" class="w-10 h-10 rounded-full mr-3">

<div class="flex-1">

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

<h3 class="font-bold">Ms. Sarah Johnson</h3>

<span class="text-sm text-gray-500">2 hours ago</span>

</div>

<p class="mt-1 text-gray-700">Today we'll be covering quadratic equations. Please review the materials I've uploaded and complete the practice problems by Friday.</p>

<div class="mt-3 flex space-x-3">

<button class="text-blue-600 hover:text-blue-800 text-sm font-medium">

<i class="far fa-thumbs-up mr-1"></i> Like

</button>

<button class="text-blue-600 hover:text-blue-800 text-sm font-medium">

<i class="far fa-comment mr-1"></i> Comment

</button>

</div>

</div>

</div>

</div>

<!-- Class Materials -->

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

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

<h3 class="font-bold text-lg">Class Materials</h3>

<button class="bg-blue-600 hover:bg-blue-700 text-white px-4 py-2 rounded-md text-sm font-medium">

<i class="fas fa-plus mr-1"></i> Add Material

</button>

</div>

<div class="divide-y">

<!-- Material Item -->

<div class="p-4 hover:bg-gray-50 transition">

<div class="flex items-start">

<div class="bg-blue-100 text-blue-800 p-3 rounded-lg mr-4">

<i class="fas fa-file-pdf text-xl"></i>

</div>

<div class="flex-1">

<h4 class="font-medium">Quadratic Equations Explained</h4>

<p class="text-sm text-gray-500 mt-1">PDF • 2.4 MB • Uploaded 1 day ago</p>

<div class="mt-3 flex space-x-3">

<button class="text-blue-600 hover:text-blue-800 text-sm font-medium">

<i class="fas fa-eye mr-1"></i> View

</button>

<button class="text-blue-600 hover:text-blue-800 text-sm font-medium">

<i class="fas fa-download mr-1"></i> Download

</button>

<button onclick="summarizeMaterial()" class="text-blue-600 hover:text-blue-800 text-sm font-medium">

<i class="fas fa-file-contract mr-1"></i> Summarize

</button>

<button onclick="generateQuiz()" class="text-blue-600 hover:text-blue-800 text-sm font-medium">

<i class="fas fa-question-circle mr-1"></i> Generate Quiz

</button>

</div>

<div id="summaryContainer" class="summary-container mt-3 bg-gray-50 p-3 rounded-md">

<h5 class="font-medium text-gray-800 mb-2">Summary:</h5>

<p id="summaryContent" class="text-gray-700 text-sm"></p>

</div>

</div>

</div>

</div>

<!-- Another Material Item -->

<div class="p-4 hover:bg-gray-50 transition">

<div class="flex items-start">

<div class="bg-green-100 text-green-800 p-3 rounded-lg mr-4">

<i class="fas fa-file-video text-xl"></i>

</div>

<div class="flex-1">

<h4 class="font-medium">Solving Quadratic Equations Video</h4>

<p class="text-sm text-gray-500 mt-1">MP4 • 15.2 MB • Uploaded 2 days ago</p>

<div class="mt-3 flex space-x-3">

<button class="text-blue-600 hover:text-blue-800 text-sm font-medium">

<i class="fas fa-play mr-1"></i> Play

</button>

<button class="text-blue-600 hover:text-blue-800 text-sm font-medium">

<i class="fas fa-download mr-1"></i> Download

</button>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

<!-- Right Column - Classroom Tools -->

<div class="space-y-6">

<!-- Live Classroom Section -->

<div class="bg-white rounded-lg shadow p-4">

<h3 class="font-bold text-lg mb-3">Live Classroom</h3>

<div class="relative">

<video id="cameraFeed" class="w-full h-auto bg-black rounded-lg camera-feed" autoplay muted></video>

<div class="absolute bottom-3 left-0 right-0 flex justify-center space-x-3">

<button id="toggleCamera" class="bg-white p-2 rounded-full shadow-md hover:bg-gray-100">

<i class="fas fa-video text-gray-700"></i>

</button>

<button id="toggleMic" class="bg-white p-2 rounded-full shadow-md hover:bg-gray-100">

<i class="fas fa-microphone text-gray-700"></i>

</button>

<button id="screenShare" class="bg-white p-2 rounded-full shadow-md hover:bg-gray-100">

<i class="fas fa-desktop text-gray-700"></i>

</button>

</div>

</div>

<!-- Student Participation -->

<div class="mt-4">

<h4 class="font-medium text-gray-700 mb-2">Student Participation</h4>

<div class="space-y-2 max-h-60 overflow-y-auto">

<!-- Student cards will be added here dynamically -->

<div id="studentList">

<!-- Sample student - would be generated dynamically in real app -->

<div class="student-card flex items-center p-2 hover:bg-gray-50 rounded transition" data-id="1">

<img src="https://randomuser.me/api/portraits/men/32.jpg" alt="Student" class="w-8 h-8 rounded-full mr-3">

<div class="flex-1">

<p class="text-sm font-medium">Alex Johnson</p>

</div>

<button class="raise-hand-btn p-1 text-gray-400 hover:text-yellow-500" onclick="raiseHand(this, 1)">

<i class="far fa-hand-paper"></i>

</button>

</div>

<div class="student-card flex items-center p-2 hover:bg-gray-50 rounded transition" data-id="2">

<img src="https://randomuser.me/api/portraits/women/65.jpg" alt="Student" class="w-8 h-8 rounded-full mr-3">

<div class="flex-1">

<p class="text-sm font-medium">Maria Garcia</p>

</div>

<button class="raise-hand-btn p-1 text-gray-400 hover:text-yellow-500" onclick="raiseHand(this, 2)">

<i class="far fa-hand-paper"></i>

</button>

</div>

<div class="student-card flex items-center p-2 hover:bg-gray-50 rounded transition" data-id="3">

<img src="https://randomuser.me/api/portraits/men/75.jpg" alt="Student" class="w-8 h-8 rounded-full mr-3">

<div class="flex-1">

<p class="text-sm font-medium">James Wilson</p>

</div>

<button class="raise-hand-btn p-1 text-gray-400 hover:text-yellow-500" onclick="raiseHand(this, 3)">

<i class="far fa-hand-paper"></i>

</button>

</div>

</div>

</div>

</div>

</div>

<!-- AI Chat Assistant -->

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

<div class="p-4 border-b">

<h3 class="font-bold text-lg">AI Teaching Assistant</h3>

</div>

<div class="p-4 h-64 overflow-y-auto" id="chatMessages">

<div class="chat-message bg-blue-50 p-3 rounded-lg mb-3">

<p class="text-sm text-gray-800">Hello! I'm your AI teaching assistant. How can I help you with your class today?</p>

</div>

</div>

<div class="p-3 border-t flex">

<input type="text" id="chatInput" placeholder="Ask me anything..." class="flex-1 border rounded-l-lg px-3 py-2 text-sm focus:outline-none focus:ring-1 focus:ring-blue-500">

<button onclick="sendMessage()" class="bg-blue-600 hover:bg-blue-700 text-white px-4 py-2 rounded-r-lg text-sm">

<i class="fas fa-paper-plane"></i>

</button>

</div>

</div>

<!-- Quiz Generator & Scoreboard -->

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

<div class="p-4 border-b">

<h3 class="font-bold text-lg">Quiz Results</h3>

</div>

<div class="p-4">

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

<div class="mb-4">

<h4 class="font-medium mb-2" id="quizQuestion">Question will appear here</h4>

<div class="space-y-2" id="quizOptions">

<!-- Options will be added here -->

</div>

</div>

<button onclick="submitQuizAnswer()" class="w-full bg-blue-600 hover:bg-blue-700 text-white py-2 rounded-md text-sm font-medium">

Submit Answer

</button>

</div>

<div id="scoreboard">

<table class="w-full text-sm">

<thead>

<tr class="border-b">

<th class="text-left py-2">Student</th>

<th class="text-right">Score</th>

</tr>

</thead>

<tbody id="scoreboardBody">

<tr class="border-b">

<td class="py-2">Alex Johnson</td>

<td class="text-right">85%</td>

</tr>

<tr class="border-b">

<td class="py-2">Maria Garcia</td>

<td class="text-right">92%</td>

</tr>

<tr>

<td class="py-2">James Wilson</td>

<td class="text-right">78%</td>

</tr>

</tbody>

</table>

</div>

</div>

</div>

</div>

</main>

</div>

</div>

<!-- Notification Toast -->

<div id="notificationToast" class="fixed bottom-4 right-4 bg-white shadow-lg rounded-lg p-4 max-w-xs hidden transform transition-all duration-300 translate-y-4 opacity-0">

<div class="flex items-start">

<div class="flex-shrink-0 text-yellow-500">

<i class="fas fa-hand-paper text-xl"></i>

</div>

<div class="ml-3">

<h4 class="text-sm font-medium text-gray-900" id="notificationTitle">Hand Raised</h4>

<p class="text-sm text-gray-500" id="notificationMessage">A student has raised their hand for assistance.</p>

</div>

<button onclick="hideNotification()" class="ml-auto -mx-1.5 -my-1.5 bg-white text-gray-400 hover:text-gray-900 rounded-lg focus:ring-2 focus:ring-gray-300 p-1.5 hover:bg-gray-100 inline-flex h-8 w-8">

<i class="fas fa-times"></i>

</button>

</div>

</div>

<script>

// Initialize camera feed

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

// Simulate camera access

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

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

let cameraOn = true;

toggleCamera.addEventListener('click', function() {

cameraOn = !cameraOn;

if (cameraOn) {

cameraFeed.style.display = 'block';

toggleCamera.innerHTML = '<i class="fas fa-video text-gray-700"></i>';

} else {

cameraFeed.style.display = 'none';

toggleCamera.innerHTML = '<i class="fas fa-video-slash text-gray-700"></i>';

}

});

// Toggle mic

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

let micOn = true;

toggleMic.addEventListener('click', function() {

micOn = !micOn;

if (micOn) {

toggleMic.innerHTML = '<i class="fas fa-microphone text-gray-700"></i>';

} else {

toggleMic.innerHTML = '<i class="fas fa-microphone-slash text-gray-700"></i>';

}

});

// Screen share button

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

screenShare.addEventListener('click', function() {

alert('Screen sharing functionality would be implemented here in a real application.');

});

});

// Hand raising functionality

let raisedHands = {};

function raiseHand(button, studentId) {

const studentCard = button.closest('.student-card');

const handIcon = button.querySelector('i');

if (handIcon.classList.contains('text-yellow-500')) {

// Lower hand

handIcon.classList.remove('text-yellow-500', 'fa-solid');

handIcon.classList.add('text-gray-400', 'far');

studentCard.classList.remove('hand-raised');

clearTimeout(raisedHands[studentId]);

delete raisedHands[studentId];

} else {

// Raise hand

handIcon.classList.remove('text-gray-400', 'far');

handIcon.classList.add('text-yellow-500', 'fa-solid');

studentCard.classList.add('hand-raised');

// Set timeout for notification after 5 seconds

raisedHands[studentId] = setTimeout(() => {

showNotification(studentId);

}, 5000);

}

}

function showNotification(studentId) {

const studentName = document.querySelector(.student-card[data-id="${studentId}"] p).textContent;

const notification = document.getElementById('notificationToast');

document.getElementById('notificationTitle').textContent = 'Hand Raised';

document.getElementById('notificationMessage').textContent = ${studentName} has raised their hand for more than 5 seconds.;

notification.classList.remove('hidden');

setTimeout(() => {

notification.classList.remove('translate-y-4', 'opacity-0');

notification.classList.add('translate-y-0', 'opacity-100');

}, 10);

}

function hideNotification() {

const notification = document.getElementById('notificationToast');

notification.classList.remove('translate-y-0', 'opacity-100');

notification.classList.add('translate-y-4', 'opacity-0');

setTimeout(() => {

notification.classList.add('hidden');

}, 300);

}

// AI Chat Assistant

function sendMessage() {

const input = document.getElementById('chatInput');

const message = input.value.trim();

if (message) {

// Add user message

addChatMessage(message, 'user');

input.value = '';

// Simulate AI response after a short delay

setTimeout(() => {

const responses = [

"Quadratic equations can be solved using three main methods: factoring, completing the square, and the quadratic formula.",

"The general form of a quadratic equation is ax² + bx + c = 0, where a, b, and c are coefficients.",

"The discriminant (b² - 4ac) tells us about the nature of the roots of the quadratic equation.",

"Would you like me to generate some practice problems for your students?",

"I can help explain any concepts your students might be struggling with."

];

const randomResponse = responses[Math.floor(Math.random() \* responses.length)];

addChatMessage(randomResponse, 'ai');

}, 1000);

}

}

function addChatMessage(message, sender) {

const chatContainer = document.getElementById('chatMessages');

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

messageDiv.classList.add('chat-message', 'mb-3');

if (sender === 'user') {

messageDiv.classList.add('ml-auto', 'bg-blue-600', 'text-white', 'p-3', 'rounded-lg');

} else {

messageDiv.classList.add('bg-blue-50', 'p-3', 'rounded-lg');

}

messageDiv.innerHTML = <p class="text-sm">${message}</p>;

chatContainer.appendChild(messageDiv);

chatContainer.scrollTop = chatContainer.scrollHeight;

}

// Material summarization

function summarizeMaterial() {

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

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

// Simulate AI summarization

const summary = `

<p class="mb-2"><strong>Key Points:</strong></p>

<ul class="list-disc pl-5 space-y-1">

<li>Quadratic equations are polynomial equations of degree 2.</li>

<li>The standard form is ax² + bx + c = 0 where a ≠ 0.</li>

<li>Three main solution methods: factoring, completing the square, and quadratic formula.</li>

<li>The discriminant (D = b² - 4ac) determines the nature of roots.</li>

<li>Graphically represented by parabolas.</li>

</ul>

<p class="mt-2"><strong>Applications:</strong> Physics (projectile motion), engineering, economics, and optimization problems.</p>

`;

summaryContent.innerHTML = summary;

summaryContainer.classList.add('open');

}

// Quiz generator

function generateQuiz() {

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

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

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

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

// Hide scoreboard and show quiz

scoreboard.classList.add('hidden');

quizContainer.classList.remove('hidden');

// Generate quiz question (in a real app, this would be AI-generated from the material)

quizQuestion.textContent = "What is the solution to the quadratic equation x² - 5x + 6 = 0?";

// Generate options

quizOptions.innerHTML = `

<label class="flex items-center space-x-2 p-2 border rounded hover:bg-gray-50 cursor-pointer">

<input type="radio" name="quizAnswer" value="A" class="text-blue-600">

<span>x = 2 and x = 3</span>

</label>

<label class="flex items-center space-x-2 p-2 border rounded hover:bg-gray-50 cursor-pointer">

<input type="radio" name="quizAnswer" value="B" class="text-blue-600">

<span>x = -2 and x = -3</span>

</label>

<label class="flex items-center space-x-2 p-2 border rounded hover:bg-gray-50 cursor-pointer">

<input type="radio" name="quizAnswer" value="C" class="text-blue-600">

<span>x = 1 and x = 6</span>

</label>

<label class="flex items-center space-x-2 p-2 border rounded hover:bg-gray-50 cursor-pointer">

<input type="radio" name="quizAnswer" value="D" class="text-blue-600">

<span>No real solutions</span>

</label>

`;

}

function submitQuizAnswer() {

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

if (selectedOption) {

// In a real app, this would check the answer and update scores

alert(You selected option ${selectedOption.value}. In a real application, this would be checked and scores would be updated.);

// Show scoreboard again

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

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

} else {

alert('Please select an answer before submitting.');

}

}

// Allow pressing Enter in chat input

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

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

sendMessage();

}

});

</script>

</body>

</html>