// Define mongoose schema for mentors, students, tasks, and assignments

const mongoose = require('mongoose');

const mentorSchema = new mongoose.Schema({

name: String,

email: String,

expertise: String

});

const studentSchema = new mongoose.Schema({

name: String,

email: String,

course: String

});

const taskSchema = new mongoose.Schema({

description: String,

difficulty: String

});

const assignmentSchema = new mongoose.Schema({

task: { type: mongoose.Schema.Types.ObjectId, ref: 'Task' },

mentor: { type: mongoose.Schema.Types.ObjectId, ref: 'Mentor' },

student: { type: mongoose.Schema.Types.ObjectId, ref: 'Student' },

assignedDate: { type: Date, default: Date.now },

dueDate: Date,

status: String

});

const Mentor = mongoose.model('Mentor', mentorSchema);

const Student = mongoose.model('Student', studentSchema);

const Task = mongoose.model('Task', taskSchema);

const Assignment = mongoose.model('Assignment', assignmentSchema);

module.exports = { Mentor, Student, Task, Assignment };

BACKEND DEVELOPMENT

const express = require('express');

const mongoose = require('mongoose');

const { Mentor, Student, Task, Assignment } = require('./models');

const app = express();

app.use(express.json());

// MongoDB connection

mongoose.connect('mongodb://localhost:27017/webdev\_task\_assignment', { useNewUrlParser: true, useUnifiedTopology: true })

.then(() => console.log('Connected to MongoDB'))

.catch(err => console.error('Failed to connect to MongoDB', err));

// Routes for CRUD operations

// Mentor routes

app.post('/mentors', async (req, res) => {

const mentor = new Mentor(req.body);

await mentor.save();

res.send(mentor);

});

app.get('/mentors', async (req, res) => {

const mentors = await Mentor.find();

res.send(mentors);

});

// Similar routes for students, tasks, and assignments

const port = process.env.PORT || 3000;

app.listen(port, () => console.log(`Server is running on port ${port}`));