DataBase connection

const pool = require('../config/db');
const path = require('path');
const fs = require('fs');
const PDFDocument = require('pdfkit');

const studyFolderController = {
  getStudyMaterials: async (req, res) => {
    try {
      const [materials] = await pool.execute(
        'SELECT * FROM notes ORDER BY subject_name, created_at DESC'
      );
      res.json(materials);
    } catch (error) {
      res.status(500).json({ message: error.message });
    }
  },

  savePDFToStudyFolder: async (req, res) => {
    try {
      const { noteId } = req.params;
      const [note] = await pool.execute(
        'SELECT * FROM notes WHERE id = ?',
        [noteId]
      );

      if (!note[0]) {
        return res.status(404).json({ message: 'Note not found' });
      }

      const { subject_name, title, content } = note[0];

      // Create directories if they don't exist
      const studyDir = path.join(__dirname, `../uploads/study/${subject_name}`);
      fs.mkdirSync(studyDir, { recursive: true });

      // Clean content by removing HTML tags
      const cleanContent = content.replace(/<[^>]*>/g, '');

      // Generate PDF
      const pdfPath = path.join(studyDir, `${title}.pdf`);
      const doc = new PDFDocument();
      const stream = fs.createWriteStream(pdfPath);

      doc.pipe(stream);

      // Add title
      doc.fontSize(20).text(title, { align: 'center' });
      doc.moveDown();

      // Add content with proper formatting
      doc.fontSize(12).text(cleanContent, {
        align: 'left',
        lineGap: 5
      });

      doc.end();

      // Wait for PDF to finish writing
      stream.on('finish', () => {
        const relativePath = `/uploads/study/
${subject_name}/${title}.pdf`;
        res.json({ filePath: relativePath });
      });

    } catch (error) {
      res.status(500).json({ message: error.message });
    }
  },

  searchMaterials: async (req, res) => {
    try {
      const { searchTerm } = req.query;
      const [materials] = await pool.execute(
        'SELECT * FROM notes WHERE subject_name
LIKE ? OR title LIKE ?',
        [`%${searchTerm}%`, `%${searchTerm}%`]
      );
      res.json(materials);
    } catch (error) {
      res.status(500).json({ message: error.message });
    }
  }
};

module.exports = studyFolderController;