DataBase connection

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

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

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

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

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

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

&nbsp; &nbsp; &nbsp; // Clean content by removing HTML tags
&nbsp; &nbsp; &nbsp; const cleanContent = content.replace(/&lt;[^&gt;]*&gt;/g, '');

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

&nbsp; &nbsp; &nbsp; doc.pipe(stream);
&nbsp; &nbsp; &nbsp;
&nbsp; &nbsp; &nbsp; // Add title
&nbsp; &nbsp; &nbsp; doc.fontSize(20).text(title, { align: 'center' });
&nbsp; &nbsp; &nbsp; doc.moveDown();
&nbsp; &nbsp; &nbsp;
&nbsp; &nbsp; &nbsp; // Add content with proper formatting
&nbsp; &nbsp; &nbsp; doc.fontSize(12).text(cleanContent, {
&nbsp; &nbsp; &nbsp; &nbsp; align: 'left',
&nbsp; &nbsp; &nbsp; &nbsp; lineGap: 5
&nbsp; &nbsp; &nbsp; });
&nbsp; &nbsp; &nbsp;
&nbsp; &nbsp; &nbsp; doc.end();

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

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

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

module.exports = studyFolderController;
```