```
const fs = require("fs");
const path = require("path");
const glob = require("glob");

/* ---------------- IMPORT SORTING ---------------- */
function sortImports(content) {
    const lines = content.split("\n");

    let useClient = [];
    let imports = [];
    let others = [];

    lines.forEach((line) => {
        if (line.trim() === '"use client";' || line.trim() === "'use client';")
{
            useClient.push(line);
        } else if (line.trim().startsWith("import ")) {
            imports.push(line);
        } else {
            others.push(line);
        }
    });

    // sort imports by line length
    imports.sort((a, b) => a.length - b.length);

    return [...useClient, ...imports, ...others].join("\n");
}

/* ---------------- CSS SORTING ---------------- */
function sortCssContent(content) {
    const lines = content.split("\n");
    let sortedContent = [];
    let stack = [];

    function flushBlock(block) {
        if (block.lines.length > 0) {
            // short → long sorting
            block.lines.sort((a, b) => a.trim().length - b.trim().length);
            block.sortedContent.push(...block.lines);
            block.lines = [];
        }
    }

    lines.forEach((line) => {
        const trimmedLine = line.trim();

        if (trimmedLine.endsWith("{")) {
            const newBlock = { sortedContent: [line], lines: [] };
            stack.push(newBlock);
        } else if (trimmedLine === "}") {
            const finishedBlock = stack.pop();
            flushBlock(finishedBlock);
            finishedBlock.sortedContent.push(line);

            if (stack.length > 0) {
                stack[stack.length -
1].sortedContent.push(...finishedBlock.sortedContent);
            } else {
                sortedContent.push(...finishedBlock.sortedContent);
            }
        } else {
            if (stack.length > 0) {
                if (trimmedLine.includes(":") && trimmedLine.endsWith(";")) {
```

```
                    stack[stack.length - 1].lines.push(line);
                } else {
                    stack[stack.length - 1].sortedContent.push(line);
                }
            } else {
                sortedContent.push(line);
            }
        }
    });

    return sortedContent.join("\n");
}

/* ---------------- MAIN SCRIPT ---------------- */
function sortProjectFiles() {
    // 1. Sort imports in all JS/TS files
    const codeFiles = glob.sync("src/**/*.{js,jsx,ts,tsx}", { ignore:
"node_modules/**" });
    codeFiles.forEach((file) => {
        const content = fs.readFileSync(file, "utf-8");
        const newContent = sortImports(content);
        fs.writeFileSync(file, newContent, "utf-8");
        console.log(`✅ Sorted imports in ${file}`);
    });

    // 2. Sort CSS in all module.css files
    const cssFiles = glob.sync("src/**/*.module.css", { ignore:
"node_modules/**" });
    cssFiles.forEach((file) => {
        const content = fs.readFileSync(file, "utf-8");
        const newContent = sortCssContent(content);
        fs.writeFileSync(file, newContent, "utf-8");
        console.log(`🎨 Sorted CSS in ${file}`);
    });
}

// Run directly
if (require.main === module) {
    sortProjectFiles();
}
```