

1)npm install --save-dev prettier

2)npm install glob --save-dev

```
const fs = require("fs");
const path = require("path");
const glob = require("glob");
const { execSync } = require("child_process");

/**
 * Sort imports in JS/TS files
 */
function sortImports(content) {
  const lines = content.split("\n");

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

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

  // Sort imports shortest → longest
  imports.sort((a, b) => a.length - b.length);

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

/**
 * Sort CSS properties inside all CSS blocks (top-level + nested like @media)
 */
function sortCssContent(content) {
  const lines = content.split("\n");
  let sortedContent = [];
  let stack = [];

  function flushBlock(block) {
    if (block.lines.length > 0) {
      // Sort short → long (line length)
      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");
}

/**
 * Process CSS + Imports
 */
function sortProject() {
  // Sort module.css files
  const cssFiles = glob.sync("src/**/*.module.css");
  cssFiles.forEach((file) => {
    const content = fs.readFileSync(file, "utf-8");
    const finalContent = sortCssContent(content);
    fs.writeFileSync(file, finalContent, "utf-8");
    console.log(`✅ Sorted CSS in ${file}`);
  });

  // Sort imports in JS/TS files
  const codeFiles = glob.sync("src/**/*.{js,jsx,ts,tsx}");
  codeFiles.forEach((file) => {
    const content = fs.readFileSync(file, "utf-8");
    const finalContent = sortImports(content);
    fs.writeFileSync(file, finalContent, "utf-8");
    console.log(`✅ Sorted Imports in ${file}`);
  });

  // Run Prettier after sorting
  try {
    console.log("🚀 Running Prettier...");
    execSync("npx prettier --write .", { stdio: "inherit" });
    console.log("✅ Prettier formatting done!");
  } catch (err) {
    console.error(
      "⚠️ Prettier failed. Please install it with: npm install --save-dev prettier",
    );
  }
}

```

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