**1.Code review by AI:**

**Cursor Prompt:**

Analyze this project comprehensively:

**1. FIND ISSUES:**

- Complexity problems

- Performance bottlenecks

- Security vulnerabilities

- Readability issues

- Potential bugs

**2. MEASURE:**

- Cyclomatic complexity

- Lines of code

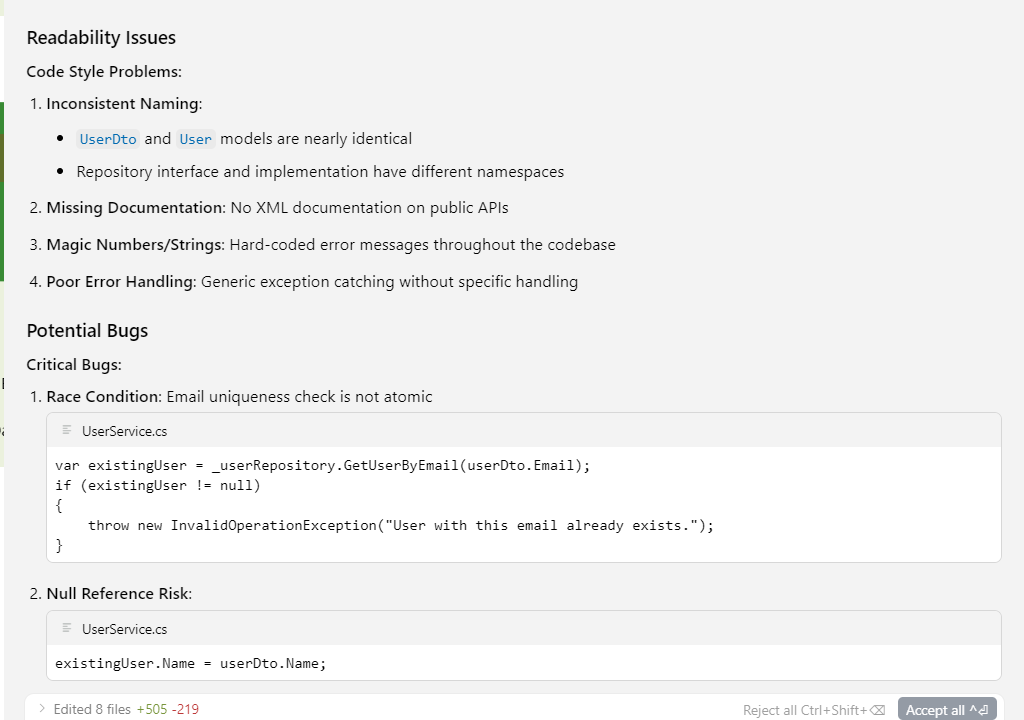
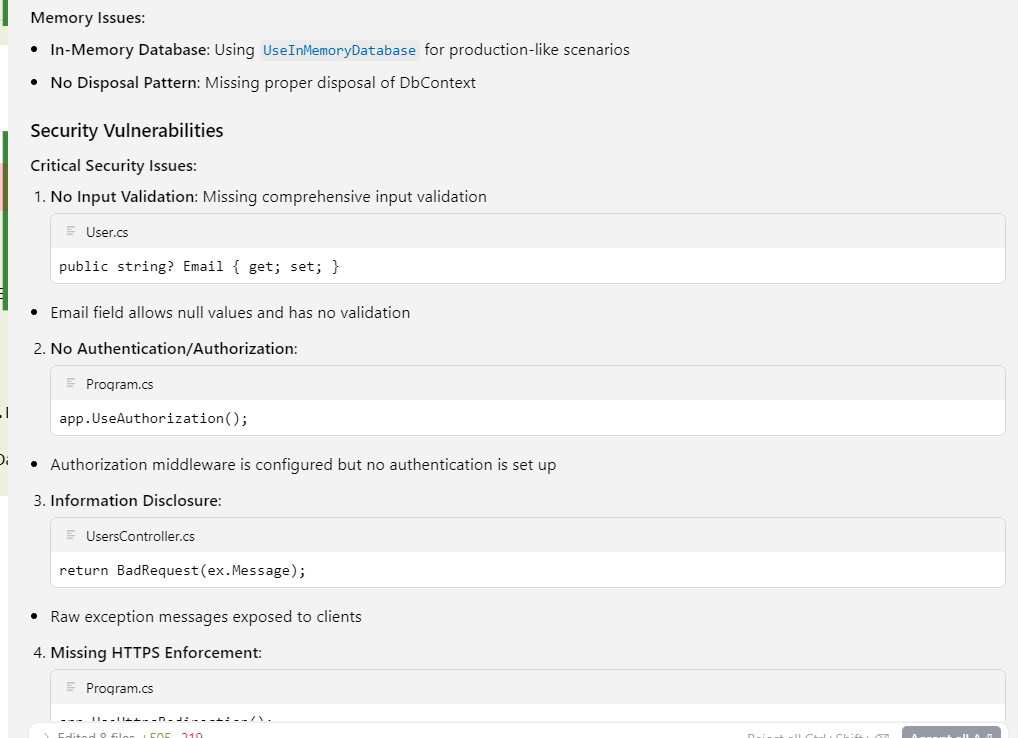
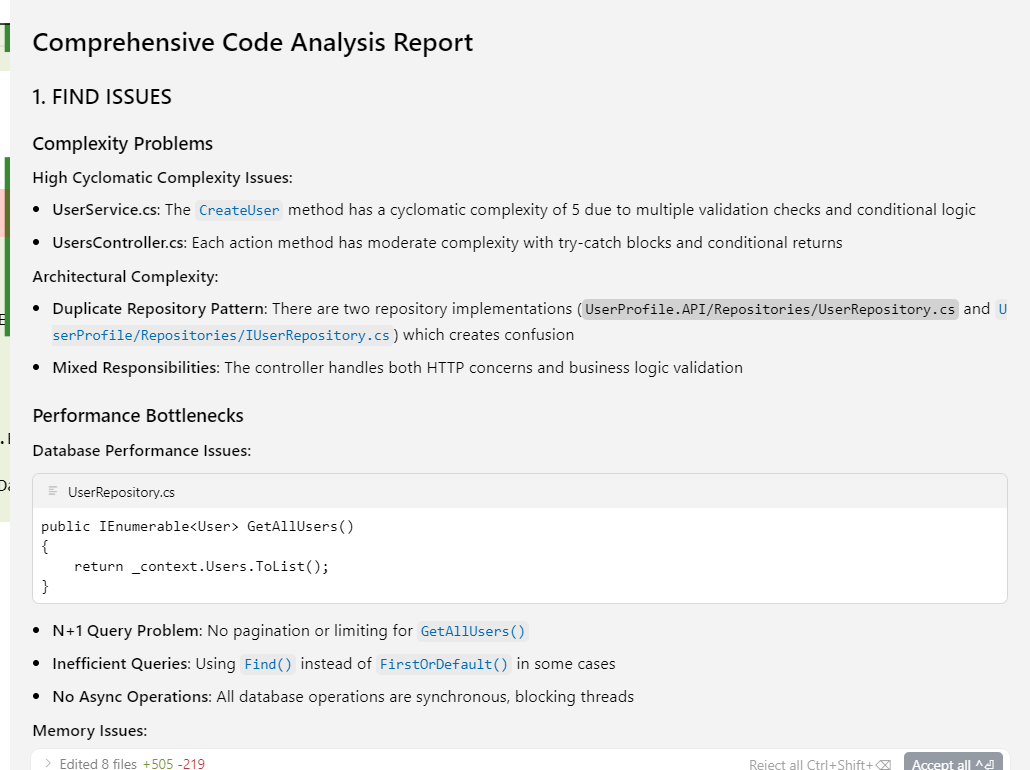
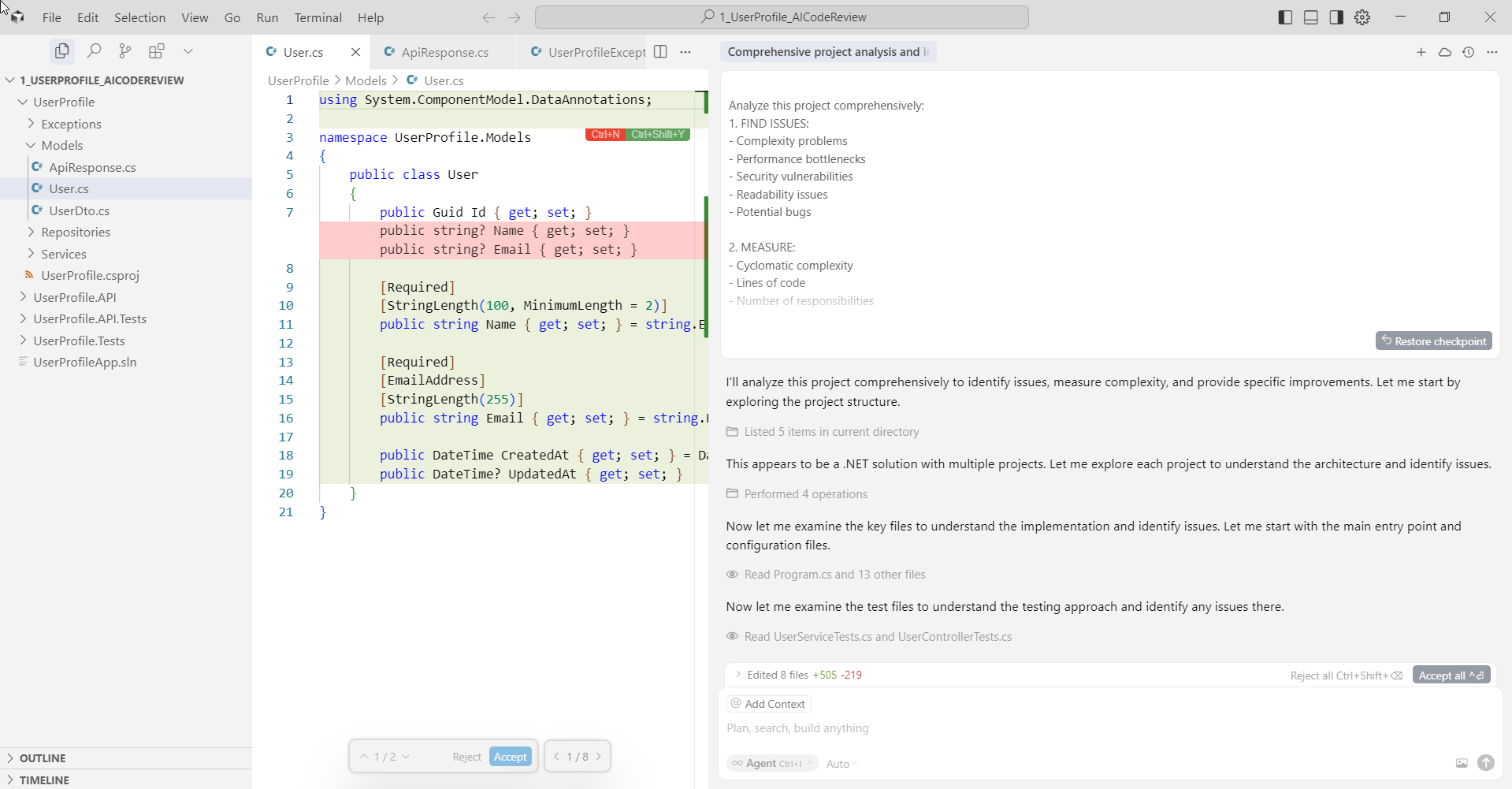
- Number of responsibilities

**3.SUGGEST:**

-Specific improvements

-Refactoring opportunities

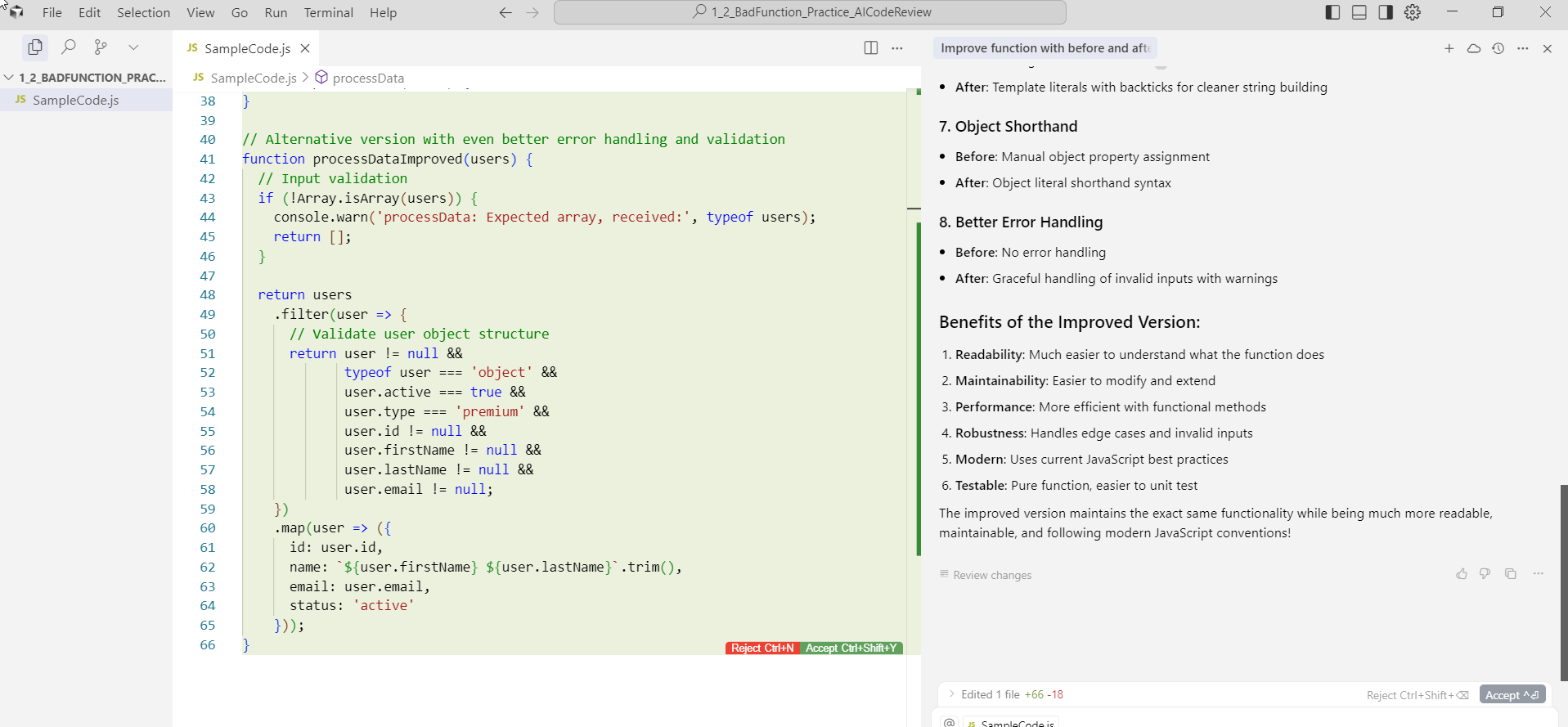
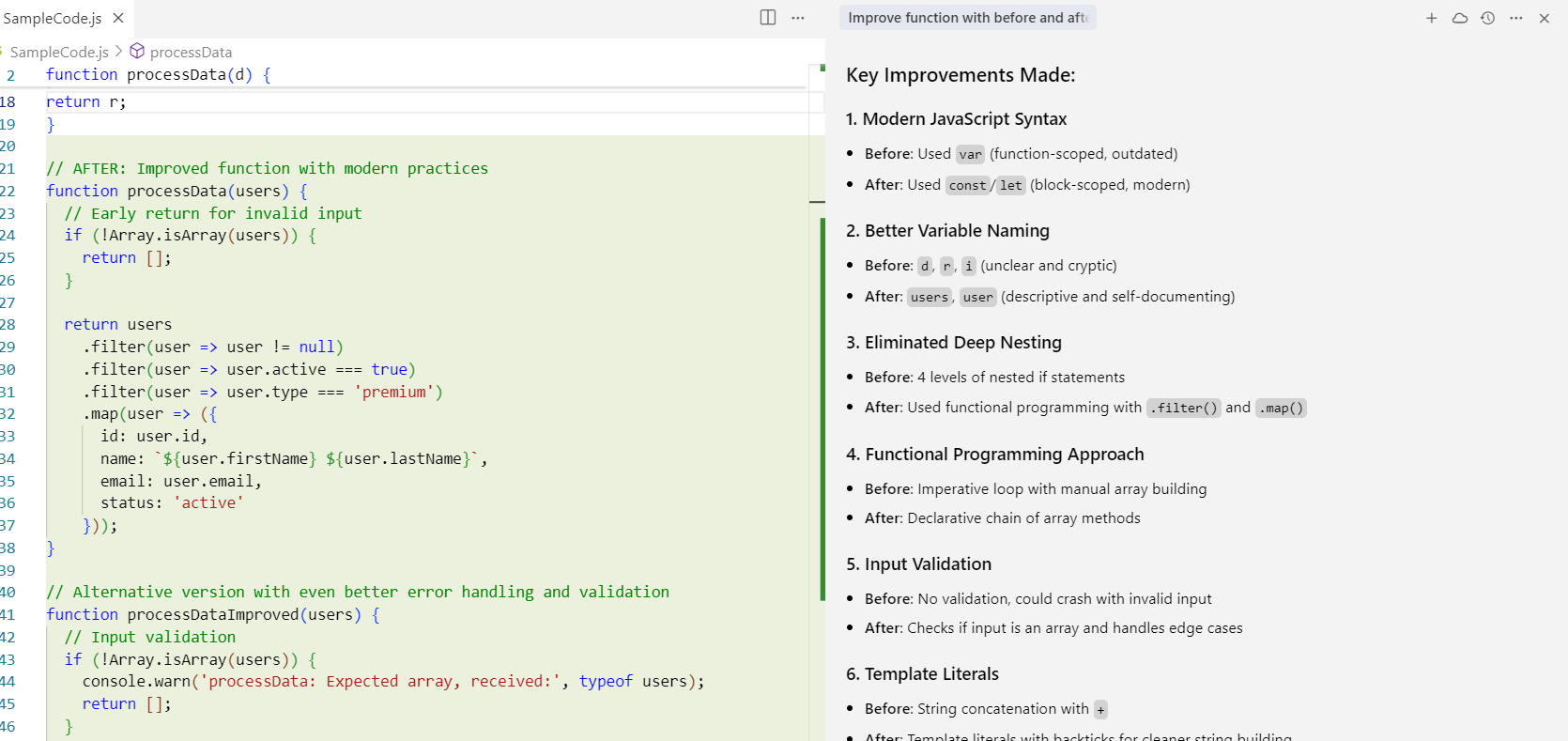
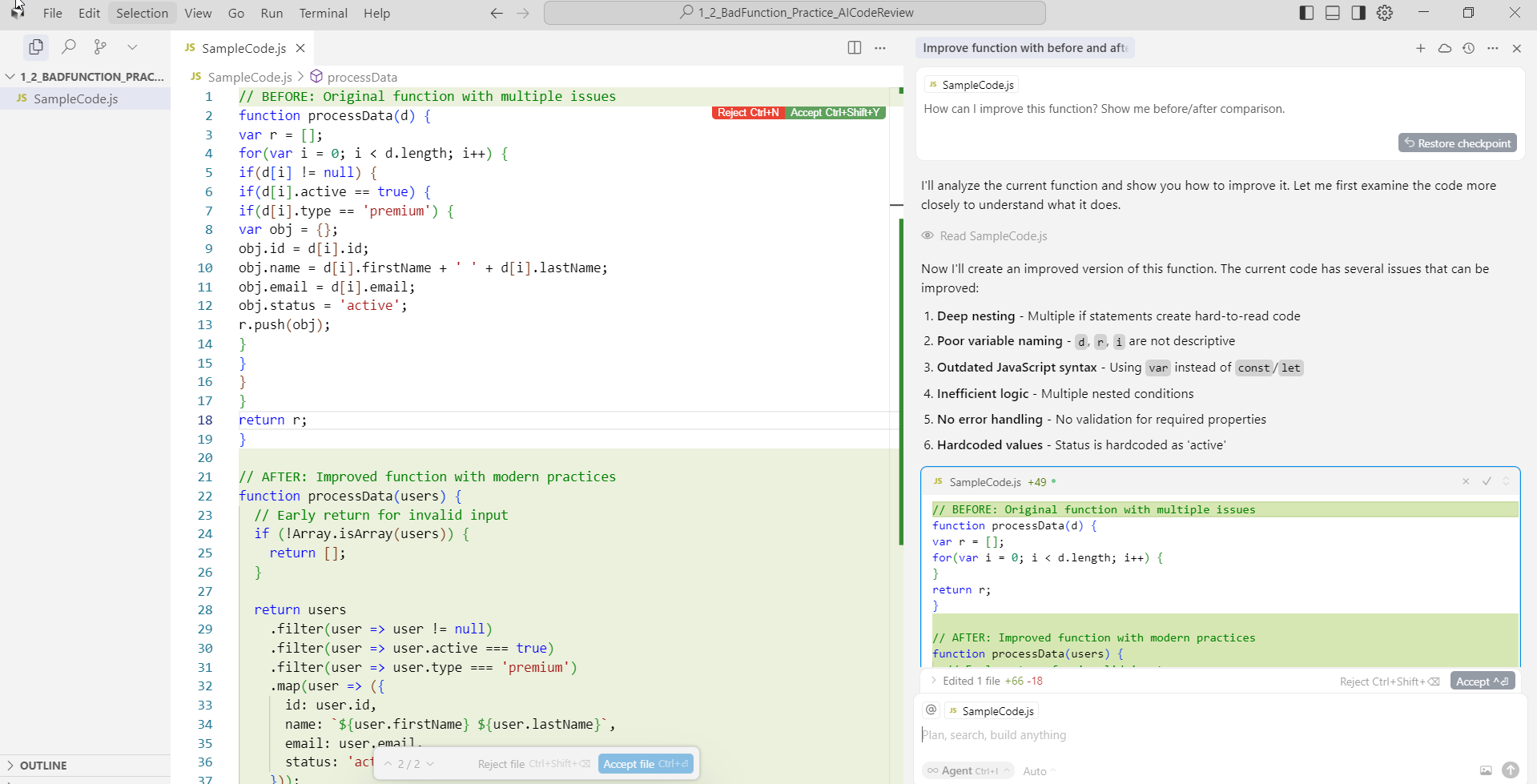
-Best practice violations Provide concrete examples and fixes



# 1\_2\_Function:

**Ask your AI tool:**

"How can I improve this function? Show me before/after comparison."



# 1\_3 Documentation:

**Universal AI Prompt:**

Generate comprehensive documentation for this CreateUser function:

REQUIREMENTS:

- Clear description of purpose

- Parameter types and descriptions

- Return value explanation

- Usage examples

- Error conditions

- Performance notes if relevant

FORMAT: Use JSDoc/PyDoc/etc appropriate for the language

Make it professional and helpful for other developers.

# Module 2: Security & Performance Deep Dive

1. AI Security Analysis

Security Vulnerability Hunt

Find and fix 5 security issues

**Step 1: AI Security Scan**

For Any AI Tool - Use This Security Prompt:

SECURITY AUDIT REQUEST:

Analyze this code for security vulnerabilities:

CRITICAL SECURITY ISSUES:

- SQL injection vulnerabilities

- Cross-site scripting (XSS) risks

- Authentication/authorization flaws

- Input validation gaps

- Sensitive data exposure

SECURITY CONCERNS:

- Hardcoded secrets/passwords

- Insecure data transmission

- Weak error handling

- Missing rate limiting

- Cryptographic weaknesses

FOR EACH ISSUE FOUND:

- Severity level (Critical/High/Medium/Low)

- Specific line numbers

- Explanation of the risk

- Concrete fix with code example

- Prevention strategies

Prioritize issues by potential impact.

