CODE ANALYSIS REPORT

Project: selected_project

Date: 2025-08-30 18:13

Generated by: Code Analysis Tool

EXECUTIVE SUMMARY

This report provides a comprehensive analysis of the codebase with 13 files and 42 functions. The overall code quality score is **90.8/100 (Excellent)**.

Key Metrics:• Average Cyclomatic Complexity: 1.83• Total Lines of Code: 553• File Types: 7 .js, 1 .html, 2 .css, 6 .jsx

Recommendation: Code quality is excellent. Maintain current standards with regular reviews.

DETAILED ANALYSIS

Project Overview

Project Name: selected_project
Total Files Analyzed: 13
Total Lines of Code: 1013
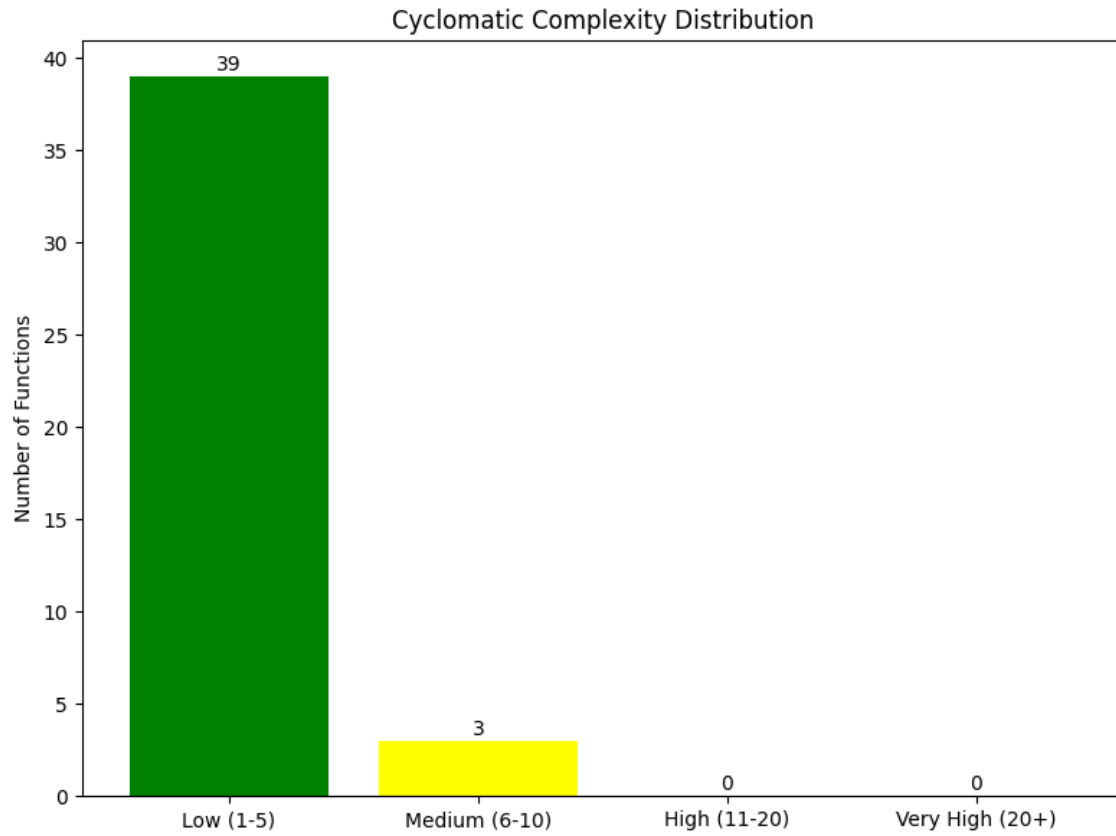**Total Cyclomatic Complexity:** 77

File Type Breakdown:
- .js: 7 files, 214 LOC
- .html: 1 files, 13 LOC
- .css: 2 files, 112 LOC
- .jsx: 6 files, 674 LOC

Analysis Scope:
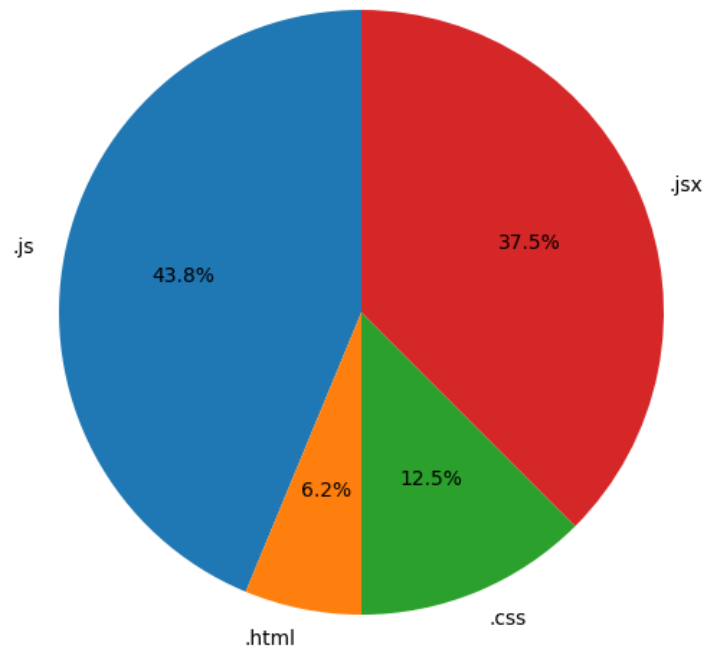This report analyzes JavaScript, Python, Java, C++, C, TypeScript, and JSX files.
Excluded directories: node_modules, .git, __pycache__, venv, .idea, .vscode

Complexity Distribution

File Type Distribution

File Type Distribution

Complexity Analysis

Total Functions: 42
Average Complexity: 1.83
Max Complexity: 6
Average LOC: 13.17
Total LOC: 553
Maintainability Index: 89.75

■■ **Long Functions (LOC > 50):** 3
- &&: LOC=54, CC=2, File: AuthForm.jsx
- (anonymous): LOC=76, CC=3, File: Dashboard.jsx
- return: LOC=127, CC=3, File: UserDetailsForm.jsx

Top Complex Functions

Function	Complexity	LOC	File
(anonymous)	6	29	auth.js

(anonymous)	6	32	auth.js
(anonymous)	6	13	Dashboard.jsx
JSON.stringify	4	17	AuthForm.jsx
Dashboard	4	44	Dashboard.jsx
(anonymous)	3	6	AuthForm.jsx
(anonymous)	3	9	Dashboard.jsx
(anonymous)	3	9	Dashboard.jsx
(anonymous)	3	76	Dashboard.jsx
return	3	127	UserDetailsForm.jsx

Detailed Analysis

Key Findings:
• Codebase contains 42 functions with an average complexity of 1.83
• 0 functions exceed the recommended complexity threshold (CC > 10)
• 3 functions are longer than recommended (LOC > 50)

Most Complex Files:
• Dashboard.jsx: 27 total complexity
• AuthForm.jsx: 14 total complexity
• UserDetailsForm.jsx: 14 total complexity
• auth.js: 12 total complexity
• index.js: 4 total complexity

Maintainability Assessment:
The maintainability index of 89.75 indicates excellent maintainability

Code Quality Recommendations

Priority Recommendations:
• HIGH: Refactor 3 long functions (LOC > 50):
- Extract helper functions for discrete operations
- Consider if function is violating the Single Responsibility Principle

General Best Practices:
• Add comments to complex algorithms for better maintainability
• Consider adding unit tests for critical functions, especially those with high complexity
• Implement code review processes to catch complexity issues early
• Use static analysis tools in your CI/CD pipeline
• Consider using guard clauses and early returns to reduce nesting
• Extract complex conditions into well-named helper functions or variables

Technology-Specific Suggestions:
• For React components: Consider splitting large components into smaller presentational and container components
• For JavaScript: Use modern ES6+ features like arrow functions, destructuring, and async/await to simplify code
• For Python: Use list comprehensions and built-in functions where appropriate to reduce complexity
• For Java/C++: Consider using design patterns to manage complexity in large codebases