

NOU-SMF-NEW

Branch : main

VERSION NOT PROVIDED

Code analysis

By: 黃宇碩

2026-06-03

Analyzed the : 2026-06-03

CONTENT

Content.....	1
Introduction.....	2
Configuration.....	2
Synthesis.....	3
Analysis Status.....	3
Quality gate status.....	3
Metrics.....	3
Tests.....	3
Detailed technical debt.....	3
Metrics Range.....	5
Volume.....	5
Issues.....	6
Charts.....	6
Issues count by severity and type.....	8
Issues List.....	8
Security Hotspots.....	9
Security hotspots count by category and priority.....	9
Security hotspots List.....	9

INTRODUCTION

This document contains results of the code analysis of nou-smf-new.

CONFIGURATION

- Quality Profiles
 - o Names: Sonar way [CSS]; Sonar way [JavaScript]; Sonar way [PHP]; Sonar way [XML];
 - o Files: b65247b7-928f-4920-a23c-b2b334dee7e5.json; 22a866d3-7bcd-4182-b3f2-1355427b457e.json; 41a2dcf5-724a-4bf6-b3cb-9df06e07e04a.json; a62596c9-3a52-4341-9b5a-b1ba7d1668f9.json;
- Quality Gate
 - o Name: Sonar way
 - o File: Sonar way.xml

SYNTHESIS

ANALYSIS STATUS

Reliability	Security	Security Review	Maintainability
A	A	E	A

QUALITY GATE STATUS

Quality Gate Status	Failed
---------------------	--------

Metric	Value
Coverage on New Code	ERROR (0.0% is less than 80%)
Duplicated Lines (%) on New Code	ERROR (43.2% is greater than 3%)
New Issues	ERROR (5 is greater than 0)

METRICS

Coverage	Duplication	Comment density	Median number of lines of code per file	Adherence to coding standard
0.0 %	20.9 %	12.6 %	69.5	98.5 %

TESTS

Total	Success Rate	Skipped	Errors	Failures
0	0 %	0	0	0

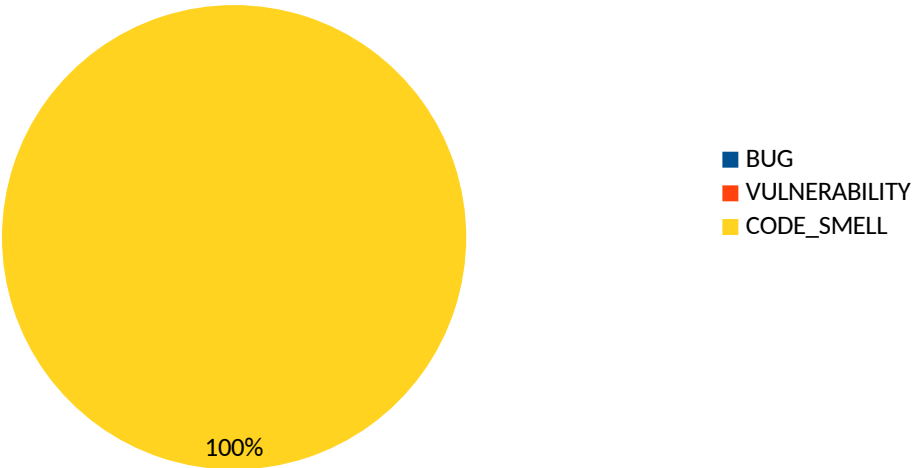
DETAILED TECHNICAL DEBT

Reliability	Security	Maintainability	Total
-	-	3d 6h 37min	3d 6h 37min

METRICS RANGE						
	Cyclomatic Complexity	Cognitive Complexity	Lines of code per file	Comment density (%)	Coverage	Duplication (%)
Min	0.0	0.0	3.0	0.0	0.0	0.0
Max	1023.0	1282.0	11446.0	42.9	0.0	98.8

VOLUME	
Language	Number
<null>	57
CSS	3844
JavaScript	735
PHP	6767
XML	100
Total	11503

Number of issues by type



ISSUES COUNT BY SEVERITY AND TYPE					
Type / Severity	INFO	MINOR	MAJOR	CRITICAL	BLOCKER
BUG	0	0	0	0	0
VULNERABILITY	0	0	0	0	0
CODE_SMELL	0	116	56	90	0

ISSUES LIST				
Name	Description	Type	Severity	Number
Variables should be declared with "let" or "const"		CODE_SMELL	CRITICAL	3
Methods should not be empty		CODE_SMELL	CRITICAL	1
String literals should not be duplicated		CODE_SMELL	CRITICAL	35
Control structures should use curly braces		CODE_SMELL	CRITICAL	7
"switch" statements should have "default" clauses		CODE_SMELL	CRITICAL	28

Cognitive Complexity of functions should not be too high	CODE_SMELL	CRITICAL	14
Unnecessary parentheses should not be used for constructs	CODE_SMELL	CRITICAL	2
Selectors should not be duplicated	CODE_SMELL	MAJOR	16
Nested blocks of code should not be left empty	CODE_SMELL	MAJOR	1
Sections of code should not be commented out	CODE_SMELL	MAJOR	8
Optional chaining should be preferred	CODE_SMELL	MAJOR	5
Mergeable "if" statements should be combined	CODE_SMELL	MAJOR	2
Functions should not have too many parameters	CODE_SMELL	MAJOR	1
Functions should not contain too many return statements	CODE_SMELL	MAJOR	5
Unused function parameters should be removed	CODE_SMELL	MAJOR	3
Sections of code should not be commented out	CODE_SMELL	MAJOR	4
Functions should not have too many lines	CODE_SMELL	MAJOR	3

of code			
Method arguments with default values should be last	CODE_SMELL	MAJOR	4
Unused assignments should be removed	CODE_SMELL	MAJOR	1
Two branches in a conditional structure should not have exactly the same implementation	CODE_SMELL	MAJOR	2
Ternary operators should not be nested	CODE_SMELL	MAJOR	1
Extra semicolons should be removed	CODE_SMELL	MINOR	4
Arguments in long RUN instructions should be sorted	CODE_SMELL	MINOR	2
Reduce the amount of consecutive RUN instructions	CODE_SMELL	MINOR	1
Ternary operator should not be used instead of simpler alternatives	CODE_SMELL	MINOR	2
Files should end with a newline	CODE_SMELL	MINOR	3

"empty()" should be used to test for emptiness