■■ SECURITY ANALYSIS REPORT

Comprehensive Code Security Assessment

Project: app

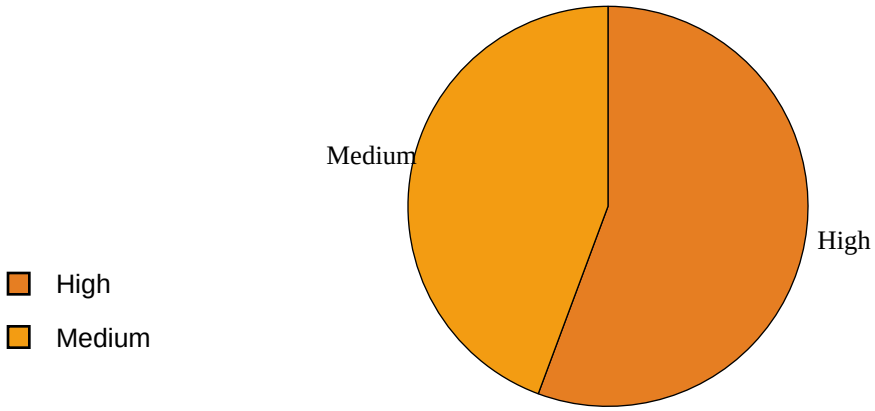
Generated: November 25, 2025 at 11:30:52

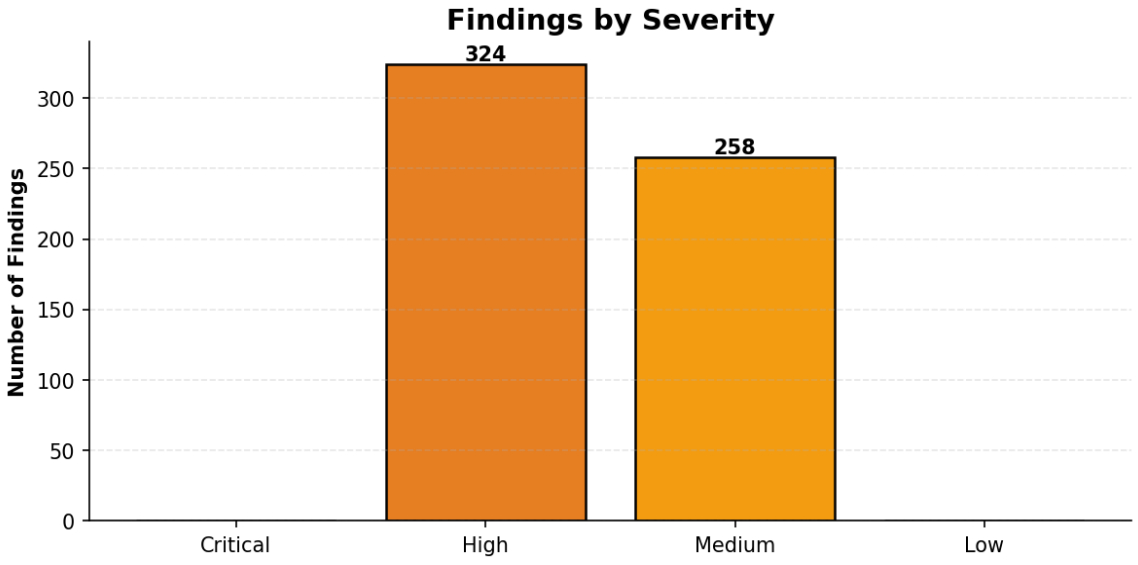
This report contains sensitive security information. Handle with care and distribute only to authorized personnel.

EXECUTIVE SUMMARY

Overall Risk Level	CRITICAL
Total Findings	584
Critical Issues	0
High Issues	324
Medium Issues	258
Low Issues	0
Languages Analyzed	php

Risk Distribution





FILE TREE HIERARCHY WITH RISK INDICATORS

Risk Level	Indicator	Description
CRITICAL	🔴	Immediate attention required - Critical vulnerabilities
HIGH	🟠	Review and fix soon - High risk issues
MEDIUM	🟡	Security concern - Should be addressed
LOW	🟢	Minor issue - Low priority
CLEAN	⚪	No security issues detected

■ project/	
■ ■■■ CHANGELOG.md	
■ ■■■ Feature_Implementation_Checklist.pdf	
■ ■■■ IMPROVEMENTS.md	
■ ■■■ Project_Milestones_And_Progress.pdf	
■ ■■■ QUICKSTART.md	
■ ■■■ SHARED_VULNERABILITY_SECTION_COMPLETE.md	
■ ■■■ SUMMARY.md	
■ ■■■ VISUAL_COMPARISON.md	
■ ■■■ analyzer_config.py	
■ ■■■ analyzer_gui.py	
■ ■■■ checkpoint_manager.py	
■ ■■■ compare_performance.py	
■ ■■■ complete_code_analysis_report.pdf	
■ ■■■ comprehensive_analyzer.py	
■ ■■■ medibook.zip	
■ ■■■ output_log.txt	
■ ■■■ project_documentation_generator.py	
■ ■■■ scan_vulnerabilities.py	
■ ■■■ security_analysis.json	
■ ■■■ security_analysis_report.pdf	
■ ■■■ test_all_functions.py	
■ ■■■ test_vulnerable_sample.js	
■ ■■■ vuln.py	
■ ■■■ README/	
■ ■ ■■■ ANTIPATTERN_DETECTOR_README.md	
■ ■ ■■■ BUG_FIX_COMPLETE.md	
■ ■ ■■■ BUG_FIX_SUMMARY.md	

■ ■ ■■■ COMPLETE_INTEGRATION_GUIDE.md	
■ ■ ■■■ COMPREHENSIVE_ANALYZER_README.md	
■ ■ ■■■ COMPREHENSIVE_PDF_VULNERABILITY_UPDATE.md	
■ ■ ■■■ CRYPTOGRAPHY_INTEGRATION.md	
■ ■ ■■■ CVE_EXTRACTION_FIX.md	
■ ■ ■■■ CVE_MAPPING_FEATURE.md	
■ ■ ■■■ DATA_ADAPTER_FIX.md	
■ ■ ■■■ DOCUMENTATION_GENERATOR_UPDATED.md	
■ ■ ■■■ DOCUMENTATION_SUMMARY.md	
■ ■ ■■■ EXECUTION_SUCCESS_REPORT.md	
■ ■ ■■■ FEATURE_SUMMARY.txt	
■ ■ ■■■ FINAL_INSTRUCTIONS.md	
■ ■ ■■■ FINAL_INTEGRATION_SUMMARY.md	
■ ■ ■■■ FINAL_PROJECT_STATUS.md	
■ ■ ■■■ FINAL_SOLUTION.md	
■ ■ ■■■ FINAL_SOLUTION_SUMMARY (2).md	
■ ■ ■■■ FINAL_SOLUTION_SUMMARY.md	
■ ■ ■■■ FINAL_WORKING_STATUS.md	
■ ■ ■■■ GUI_ARCHITECTURE.md	
■ ■ ■■■ GUI_IMPLEMENTATION_COMPLETE.md	
■ ■ ■■■ HOW_TO_USE.md	
■ ■ ■■■ IMPLEMENTATION_SUMMARY.md	

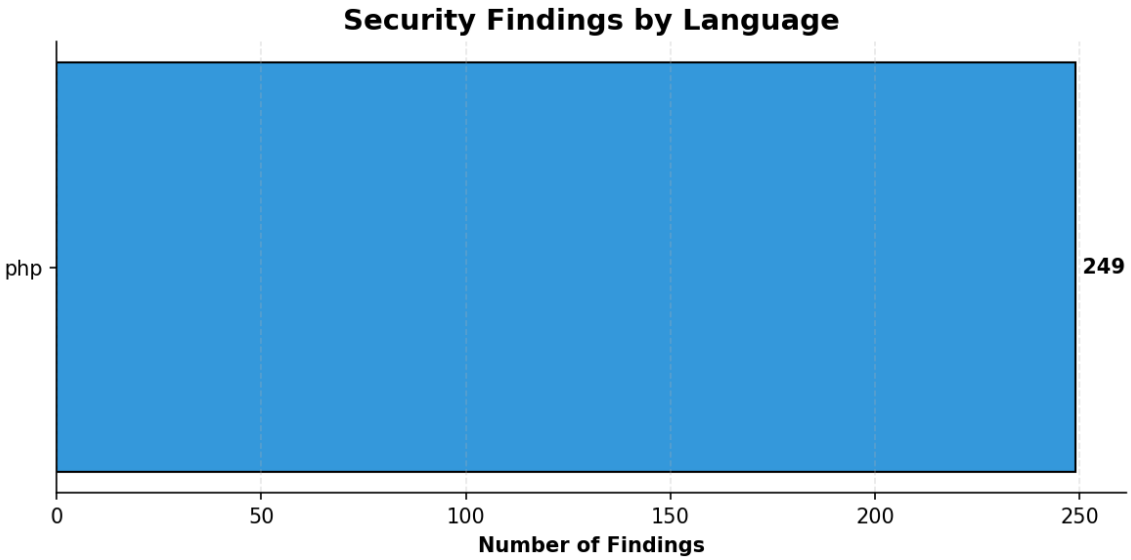
INTELLIGENT FINDINGS TABLE (DEDUPLICATED)

Category	Unique Findings	Status
Dangerous Functions	75	■■
Hardcoded Secrets	0	✓
Taint Sources	0	✓
File/Network Ops	0	✓

■ *DANGEROUS FUNCTIONS*

File	Function	Category	Lines	Count	Risk
UpdateSettingRequest.php	require	file_operations	24-29, 35-38	20	MEDIUM
UpdateServicesRequest.php	require	file_operations	22-26, 38-40	11	MEDIUM
CreateEnquiryRequest.php	require	file_operations	33-36	8	MEDIUM
Appointment.php	require	file_operations	190-197	8	MEDIUM
UpdateUserProfileRequest.	require	file_operations	26-29, 31, 3	7	MEDIUM
FrontPatientTestimonial.p	require	file_operations	69-72, 81-83	7	MEDIUM
Patient.php	require	file_operations	115-118, 120	7	MEDIUM
CreateQualificationReques	require	file_operations	32-34	6	MEDIUM
CreateStaffRequest.php	require	file_operations	23-25, 27-29	6	MEDIUM
Medicine.php	require	file_operations	90-94, 96	6	MEDIUM
Service.php	require	file_operations	101-106	6	MEDIUM
Staff.php	require	file_operations	96-99, 101-1	6	MEDIUM
User.php	require	file_operations	550-552, 554	6	MEDIUM
RegisteredUserController.	require	file_operations	30-34	5	MEDIUM
CreateAppointmentRequest.	require	file_operations	34-36	5	MEDIUM
CreatePaytmDetailRequest.	require	file_operations	23-25, 35	5	MEDIUM
UpdateMedicineRequest.php	require	file_operations	29, 37-38	5	MEDIUM
UpdateFrontCmsRequest.php	require	file_operations	26-30	5	MEDIUM
UpdateUserRequest.php	require	file_operations	23-25, 29-30	5	MEDIUM
UpdateStaffRequest.php	require	file_operations	23-25, 28-29	5	MEDIUM

DANGEROUS FUNCTIONS DETECTED



PHP - 249 findings

Function	Category	File	Line
file_get_contents	file_operations	helpers.php	396
file_get_contents	network	helpers.php	396
require	file_operations	Kernel.php	25
preg_replace	code_execution	Handler.php	49
require	file_operations	Kernel.php	19
require	file_operations	Kernel.php	86
require	file_operations	AppointmentController.php	467
preg_replace	code_execution	AuthorizePaymentController.php	50
exec	command_injection	AuthorizePaymentController.php	76
include	file_operations	GoogleCalendarController.php	45
require	file_operations	NewPasswordController.php	34
require	file_operations	NewPasswordController.php	35
require	file_operations	NewPasswordController.php	36
require	file_operations	PasswordResetLinkController.ph	31
require	file_operations	RegisteredUserController.php	30

TAINT FLOW ANALYSIS

✓ No direct taint flows detected.

HARDCODED SECRETS & CREDENTIALS

✓ No hardcoded secrets detected.

FRAMEWORK-SPECIFIC SECURITY FINDINGS

✓ No framework-specific security issues detected or module not loaded.

CRYPTOGRAPHY MISUSE ANALYSIS

Total Cryptography Issues: 2
- Weak Encryption: 0
- ECB Mode: 0
- Unsalted Passwords: 0
- JWT Issues: 0
- Weak Hashing: 0
- Predictable Random: 2

PREDICTABLE RANDOM GENERATORS - 2 findings

#	Issue	File	Line	Pattern
1	Predictable random number generator dete	Category.php	81	N/A
2	Predictable random number generator dete	Medicine.php	106	N/A

Recommendations:
• Use cryptographically secure RNG: secrets (Python), crypto.randomBytes (JS), SecureRandom

AUTHENTICATION & SESSION SECURITY

Total Authentication & Session Issues: 75

- Auth Bypass: 0
- Insecure Cookies: 0
- Weak Session Timeout: 0
- Missing Session Rotation: 6
- Missing MFA: 7
- Weak Password Policy: 62

MISSING SESSION ROTATION - 6 findings

#	Issue	File	Line	Type
1	Login functionality detected without session	Kernel.php	N/A	missing_session_rota
2	Login functionality detected without session	NewPasswordControlle	N/A	missing_session_rota
3	Login functionality detected without session	VerifyEmailControlle	N/A	missing_session_rota
4	Login functionality detected without session	Authenticate.php	N/A	missing_session_rota
5	Login functionality detected without session	RedirectIfAuthentica	N/A	missing_session_rota
6	Login functionality detected without session	LoginRequest.php	N/A	missing_session_rota

Recommendations:

• Regenerate session ID after login to prevent session fixation: session.regenerate() o

MISSING MULTI-FACTOR AUTHENTICATION - 7 findings

#	Issue	File	Line	Type
1	Authentication code without MFA/2FA implement	Kernel.php	N/A	missing_mfa
2	Authentication code without MFA/2FA implement	NewPasswordControlle	N/A	missing_mfa
3	Authentication code without MFA/2FA implement	VerifyEmailControlle	N/A	missing_mfa
4	Authentication code without MFA/2FA implement	AuthenticatedSession	N/A	missing_mfa
5	Authentication code without MFA/2FA implement	Authenticate.php	N/A	missing_mfa
6	Authentication code without MFA/2FA implement	RedirectIfAuthentica	N/A	missing_mfa
7	Authentication code without MFA/2FA implement	LoginRequest.php	N/A	missing_mfa

Recommendations:

• Implement multi-factor authentication for sensitive applications

WEAK PASSWORD POLICY - 62 findings

#	Issue	File	Line	Type
1	Password handling without length validation	Kernel.php	N/A	missing_password_len
2	Password handling without complexity requirem	Kernel.php	N/A	missing_password_com
3	Password handling without length validation	Handler.php	N/A	missing_password_len

4	Password handling without complexity requirem	Handler.php	N/A	missing_password_com
5	Password handling without length validation	PurchaseMedicineExpo	N/A	missing_password_len
6	Password handling without complexity requirem	PurchaseMedicineExpo	N/A	missing_password_com
7	Password handling without length validation	Kernel.php	N/A	missing_password_len
8	Password handling without complexity requirem	Kernel.php	N/A	missing_password_com
9	Password handling without length validation	LiveConsultationCont	N/A	missing_password_len
10	Password handling without complexity requirem	LiveConsultationCont	N/A	missing_password_com
11	Password handling without length validation	UserController.php	N/A	missing_password_len
12	Password handling without length validation	ConfirmablePasswordC	N/A	missing_password_len
13	Password handling without complexity requirem	ConfirmablePasswordC	N/A	missing_password_com
14	Password handling without length validation	NewPasswordControlle	N/A	missing_password_len
15	Password handling without complexity requirem	NewPasswordControlle	N/A	missing_password_com

Recommendations:

• Enforce minimum password length (at least 8-12 characters)

• Enforce password complexity (uppercase, lowercase, numbers, special characters)

CODE QUALITY & MAINTAINABILITY ANALYSIS

✓ No code quality issues detected! Your code follows best practices.

ANTI-PATTERN & SECURITY ISSUES DETECTION

✓ No anti-patterns or security issues detected! Your code follows best practices.

VULNERABILITY MANAGEMENT REPORT

No dependency files found in project.

SECURITY & QUALITY RECOMMENDATIONS

■ *CRITICAL*

- Remove all hardcoded secrets and passwords immediately
- Fix SQL injection vulnerabilities - use parameterized queries
- Fix code execution vulnerabilities (eval, exec)
- Address command injection flaws
- Remediate insecure deserialization
- Fix infinite loops that can cause system hangs
- Secure .env files and never commit them to version control

■ *HIGH*

- Implement input validation for all user inputs
- Use parameterized queries for ALL database operations
- Replace weak cryptographic algorithms
- Sanitize all file paths from user input
- Add proper error handling in empty catch blocks
- Add timeout to all API/HTTP requests
- Validate file paths against path traversal attacks

■ *MEDIUM*

- Implement logging and monitoring
- Set up automated security scanning
- Conduct regular security code reviews
- Use secret management tools (Vault, AWS Secrets Manager)
- Remove dead/unreachable code to improve maintainability
- Add timeout parameters to prevent resource exhaustion

■ *BEST PRACTICES*

- Implement defense-in-depth strategy
- Follow principle of least privilege
- Keep dependencies up to date
- Document security assumptions
- Enforce consistent naming conventions across the codebase

- Use linters and formatters for code quality
- Use environment variables for configuration
- Implement ORM libraries instead of raw SQL