

SecuraAI

AI-Powered Security Assessment Report

utkarsh-portfolio.zip

Scan ID:

3a9e0ba7-6602-471e-970a-

Scan Date: Oct 31, 2025,

10:32 PM

Duration: 1m 22s

Files Analyzed: 0

Lines of Code: 0

Scanner Version: SecuraAI

v1.3.0

AI Model: Google Gemini 1.5

Flash

Overall Security Grade



Risk Score: 100/100

This report contains AI-generated insights. Manual validation recommended for critical findings.

Ø=ÜÑ Table of Contents

1. Executive Summary
2. Vulnerability Distribution
3. Detailed Findings
4. Secrets & Credentials Analysis
5. Best Practices & Code Quality
6. AI-Generated Improvement Plan
7. Appendices

1b ã Executive Summary

Ø>Ý AI Assessment

The codebase demonstrates excellent security practices. No vulnerabilities were detected during the scan. Continue maintaining secure coding standards and regular security assessments to ensure ongoing protection.

Overall Risk Score: 100/100

(Grade: A)

AI Confidence: 0.93 (High)

2b ã Vulnerability Distribution

Findings by Category



OWASP Top 10 Mapping

Identified vulnerabilities have been mapped to OWASP Top 10 2021 categories:

No direct OWASP Top 10 mappings identified.

4b Secrets & Credentials Analysis

' No hardcoded secrets or credentials detected.

This is excellent! Continue using environment variables and secret management systems.

5b ã Best Practices & Code Quality

1. API Security

Issue: Missing request timeouts

Recommendation: Implement AbortController for all fetch() calls to prevent hung requests

Impact: Reliability & Resource Management

```
const controller = new AbortController();
fetch(url, { signal: controller.signal, timeout: 5000 })
```

2. Authentication

Issue: No rate limiting detected

Recommendation: Add express-rate-limit middleware to authentication endpoints

Impact: Brute-force Protection

```
app.use('/login', rateLimit({ windowMs: 15*60*1000, max: 10 }))
```

3. Security Headers

Issue: Missing security headers

Recommendation: Add helmet() middleware for CSP, HSTS, and other security headers

Impact: Browser Security & XSS Protection

```
import helmet from 'helmet';
app.use(helmet());
```

4. Error Handling

Issue: Generic error messages

Recommendation: Implement structured error handling without exposing stack traces

Impact: Information Disclosure Prevention

```
return res.status(500).json({ error: 'Internal error' }); // Don't expose error details
```

6b ã AI-Generated Improvement Plan

Overall Security Assessment

Excellent security posture. Continue current practices and stay updated with security patches.

Ø^ß Top 5 Immediate Fixes (High ROI)

Ø=ÜŒ Recommended Next Steps

- ' Enable automated security scans in CI/CD pipeline
- ' Configure CI to block deployment on critical/high findings
- ' Schedule monthly dependency update reviews
- ' Implement pre-commit hooks to prevent secret commits
- ' Conduct manual penetration testing quarterly
- ' Set up security monitoring and alerting
- ' Train development team on secure coding practices

#ð Suggested Timeline

Week 1-2: Address all critical and high severity issues

Week 3-4: Implement security headers and rate limiting

Month 2: Remediate medium severity findings

Month 3: Code quality improvements and low severity fixes

Ongoing: Monthly security reviews and dependency updates

7b ã Appendices

A. Scan Configuration

Scan Mode: Full

Rule Packs: OWASP Top 10, Secrets Detection, API Best Practices

AI Model: Google Gemini 1.5 Flash

Scanner Version: SecuraAI v1.3.0

Semgrep Version: 1.136.0

B. Glossary

OWASP: Open Web Application Security Project - Industry standard for web security

CVE: Common Vulnerabilities and Exposures - Standardized identifier for security issues

CVSS: Common Vulnerability Scoring System - Severity rating from 0-10

XSS: Cross-Site Scripting - Injection of malicious scripts into web pages

CSRF: Cross-Site Request Forgery - Unauthorized commands from trusted users

SQLi: SQL Injection - Insertion of malicious SQL queries

C. Important Disclaimer

This report contains AI-generated insights and recommendations. While our AI models are highly accurate, manual validation is recommended for critical findings. SecuraAI should be used as part of a comprehensive security strategy, not as a replacement for professional security audits. Results may include false positives. Always test fixes in a development environment before deploying to production.

CONFIDENTIAL - For authorized personnel only

CONFIDENTIAL - For authorized personnel only