Security Assessment Report

Security Assessment of OWASP Juice Shop

Rida Mudassar

25.06.2025

DHC- 377

INTRODUCTION

The purpose of this assessment was to identify, test, and fix common security vulnerabilities in the OWASP Juice Shop application, following best practices in secure software development.

SCOPE

This project involved:

- Reviewing the Juice Shop source code
- Testing for vulnerabilities like XSS, SQL Injection, etc.
- Applying secure coding practices
- Implementing logging and JWT authentication
- Documenting all fixes and learnings

Vulnerabilities Found and Fixed:

	Vulnerability	**Description**	**Fix implemented**
1	Weak password Storage	Passwords were stored insecurely or in plaintext	Used **bcrypt** for hashing
2	No Authentication	App lacked secure login token system	Implemented `jsonwebtoken` (JWT)
3	Missing Input Validation	User inputs were not validated	Used `validator` and custom checks
4	XSS Vulnerabilities	Inputs rendered without escaping	Sanitized inputs in forms/API
5	No Secure headers	App headers allowed attacks	Added `helmet` middleware
6	No logging	User activity wasn't logged	Integrated `winston` for logging
7	SQL injection Risk	Inputs used in SQL queries	Tested and protected routes using ORM

Tools Used:

- **Visual Code** (Code Editing)
- **Node.js / Express.js** (Server-side code)
- **npm packages:** `bcrypt`, `helmet`, `validator`, `jsonwebtoken`, `winston`
- **GitHub** (Version control & report upload)

Checklist Summary

✅ All inputs validated
✅ Passwords hashed using bcrypt
✅ JWT implemented for authentication
✅ Helmet used for secure headers
✅ Logging enabled with Winston
✅ SQL Injection tested & mitigated
✅ XSS vulnerabilities removed

LEARNING

- Understood how insecure apps are exploited (XSS, SQL, etc.)
- Learned how to securely hash and store passwords
- Gained experience with JWT-based token auth
- Learned the importance of input validation and logging
- Improved hands-on skills with Express.js and Node.js

CONCLUSION

This project helped me explore real-world security flaws in a modern web app and apply proper fixes using industry-standard tools and techniques. It gave practical exposure to secure coding principles and helped build a foundation for further security-focused development.