Security Testing Report - Task 1

Intern Name : Lokesh M

Domain : Cyber Security

Task Title : Web Application Security Testing

Website Tested : https://demo.owasp-juice.shop

Date : December 2025

Tools Used : Web browser (manual testing)

Objective

To perform basic web application security testing and identify common vulnerabilities like :

- SQL Injection
- Cross – site scripting (XSS)
- Authentication flaws

1. SQL Injection Test

- **Description :** Attempted SQL Injection via the login form.
- **Payload Used** : **' OR 1=1—**
- **Outcome :** Logged in successfully without valid credentials.
- **Screenshot : SQL_Injection.png**
- **Vulnerability Type** : Authentication bypass via SQL Injection
- **Mitigation Suggestion :**
  1) Use parameterized queries (prepared statements).
  2) Validate and sanitize user inputs.

2. Cross-Site Scripting(XSS) Test

- **Description** : Attempted XSS by injecting JavaScript into a form input field.
- **Payload Used** : **<script>alert("XSS")</script>**
- **Outcome** : No alert shown, but form accepted the input – possible stored or reflected XSS not executed due to frontend filters.
- **Screenshot : XSS.png**
- **Vulnerability Type :** Attempted XSS
- **Mitigation Suggestion :**

  - Sanitize HTML inputs

- Use encoding for output
- Apply Content Security Policy (CSP)

3. Authentication Flaw Test

- **Description** : Tried logging in with common/breached credentials.
- **Payload Used** : admin@juice-sh.op / admin123
- **Outcome** : Login succeeded; Google Password Manager flagged it as a breached password.
- **Screenshot : Authentication Flaws_Breached.png**
- **Vulnerability Type :** Use of weak or default credentials.
- **Mitigation Suggestion :**
  1) Enforce strong password policies.
  2) Enable account lockouts after repeated login attempts.
  3) Implement multi-factor authentication.

Conclusion

The test application exhibited common web vulnerabilities. The test covered three OWASP Top 10 vulnerabilities :

- SQL Injection
- XSS
- Broken Authentication

These flaws, if found in real-world applications, could result in data breaches, unauthorized access, and malicious script execution.