**FUTURE INTERNS (CYBER SECURITY REPORT)**

**Name : Spoorthi K.G**

**Domain: Cyber Security**

**Task 1**

**Title : Web Application Security Testing (Juice Shop)**

**Objective**

To identify and exploit common web vulnerabilities in a controlled environment using OWASP Juice Shop, with the goal of understanding attack vectors and improving secure coding practices.

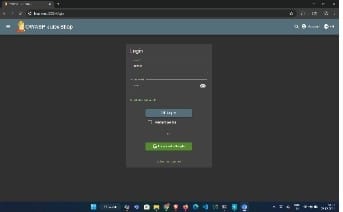
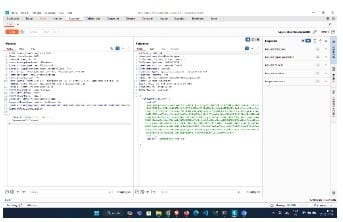
**Tools & Environment**

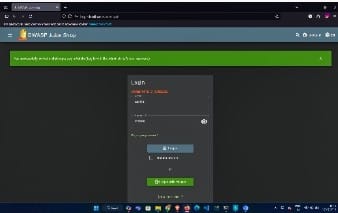
* **OWASP Juice Shop:** Intentionally vulnerable web app for ethical hacking.
* **Burp Suite:** Intercepting proxy for HTTP request manipulation.
* **Browser Developer Tools:** DOM inspection and route discovery.

**Vulnerability Analysis**

**1.** **SQL Injection – Admin Login Bypass**

* **Method**: Injected payload admin' or 1=1 -- into the login form via Burp Suite.
* **Outcome**: Successfully bypassed authentication and retrieved a valid token.
* **Why It Worked**: The query logic was flawed—injecting or 1=1 made the condition always true, logging in as the first user (admin).
* **Technical Insight**: The payload manipulated the SQL query logic, exploiting the fact that the first user in the database was the administrator. The condition OR 1=1 always evaluates to true, granting unauthorized access.



**2. Stored XSS – Profile Name Injection**

* **Initial Attempt**:Used <script>alert(‘xss’)</script>
* **Challenge**: Frontend filters blocked the script.
* **Solution**: Modified payload to bypass filters: << script>sscript>alert('xss')</script>
* **Outcome**: Payload executed, confirming stored XSS vulnerability.
* **Technical Insight**: The payload was crafted to evade frontend sanitization by duplicating characters and altering syntax. Though the alert may not visibly render due to frontend filters, the script execution confirms the vulnerability.





**3**. **Broken Access Control – Admin Panel Access**

* **Discovery**: Used browser debugger to explore Angular routes.
* **Exploit**: Found hidden /administration path and accessed admin dashboard.
* **Outcome**: Viewed sensitive data like customer reviews and user info.
* **Technical Insight**: The application lacked proper access control checks on sensitive routes. Angular routing exposed internal paths that were not protected by authentication or role-based access.

