# Cybersecurity Internship Report

**Intern Name:** Aiman Asif  
**Project Title:** Strengthening Security Measures for a Web Application  
**Submitted to:** Faizan Khan  
**Date:** 14 May, 2025

# Week 1: Security Assessment Report

**1. Application Setup & Exploration**

I cloned a simple **User Management System** web application from GitHub.

Used the following commands to set it up:

cd user-management-nodejs-mongodb-MVC

npm install

npm start

* Explored the app on http://localhost:3000 including:
  + Signup page
  + Login page
  + Profile page

**2. Basic Vulnerability Assessment**

**2.1 Cross-Site Scripting (XSS) Test**

* **Method:** Injected script <script>alert('XSS');</script> in the **name** field during signup.
* **Result:**
  + The app accepted the input without triggering any alert or blocking.
  + Indicates lack of input sanitization or output encoding.
* **Risk:** High
* **Recommendation:** Apply proper input validation and output encoding to prevent XSS.

**2.2 SQL Injection Test**

* **Method:** Used classic SQL injection string in login:

pgsql

Username: admin' OR '1'='1

Password: anything

* **Result:**
  + App returned a popup message (“@ missing in name”), showing it attempts validation but likely doesn't sanitize backend queries.
  + Indicates partial protection, but logic may still be injectable.
* **Risk:** Medium
* **Recommendation:** Use parameterized queries or ORM to prevent SQL Injection.

**🔐 2.3 Weak Password Storage Check**

* **Check:** Looked into the package.json file of the app.
* **Finding:** Found the use of bcrypt:

json

"bcrypt": "^5.1.1"

* **Conclusion:**
  + Bcrypt is a secure password hashing library.
  + Indicates that passwords are not stored in plain text.
* **Risk:** Low
* **Recommendation:** Confirm that bcrypt.hash() and bcrypt.compare() are properly used in the signup/login logic.

**🧪 3. OWASP ZAP Automated Scan**

* **Tool:** OWASP ZAP
* **Target:** <http://localhost:3000>
* **Steps:**
  1. Opened OWASP ZAP.
  2. Entered URL in automated scan section.
  3. Ran full vulnerability scan.
* **Report:**
  1. Generated and saved the ZAP scan report

Found several informational alerts and some low-medium level risks.

* **Recommendation:** Review detailed ZAP report and patch any insecure headers, cookies, or input handling issues.

**Summary of Findings**

| **#** | **Vulnerability** | **Status** | **Risk** | **Recommendation** |
| --- | --- | --- | --- | --- |
| 1 | Cross-Site Scripting | Present | High | Input/output sanitization |
| 2 | SQL Injection | Partially Present | Medium | Use prepared statements |
| 3 | Weak Password Storage | Not Present (bcrypt used) | Low | Keep using bcrypt |

**Conclusion**

The web application was successfully tested for common vulnerabilities. Issues such as XSS and potential SQL Injection were identified, while password storage appeared secure. Recommendations for each vulnerability were provided. OWASP ZAP further confirmed these assessments.