🛡 Cybersecurity Internship Final Report

Submitted by: ***Rayyan Chaaran***
Date: 26 June 2025

🕮 Introduction:

This report outlines the security testing and improvements conducted on a mock **User Management Web Application** as part of my cybersecurity internship.
The objective was to identify common web application vulnerabilities (XSS, SQL Injection, weak passwords), apply security best practices (input validation, password hashing, secure authentication), and finally implement logging and penetration testing measures.

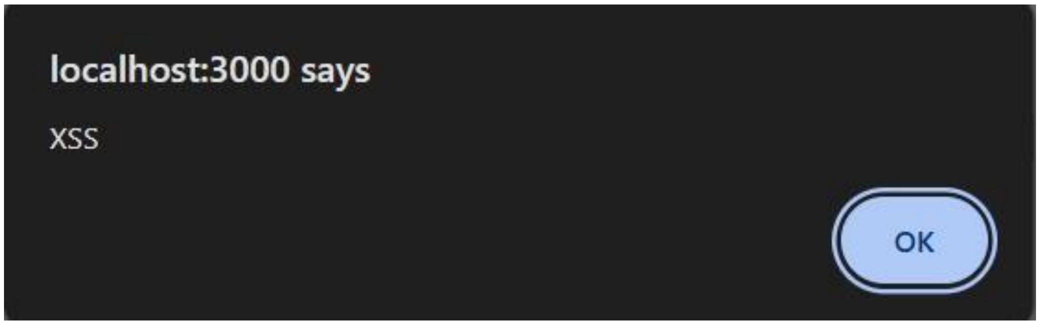
Week 3 — Advanced Security & Final Checks

Penetration Testing:

Penetration testing simulates real-world attacks. Tools like Nmap help check for open ports or outdated services that can be exploited.

Test Performed:
Scanned the local host with Nmap for vulnerable services.

✅ **Result:**
No unintended open ports detected.



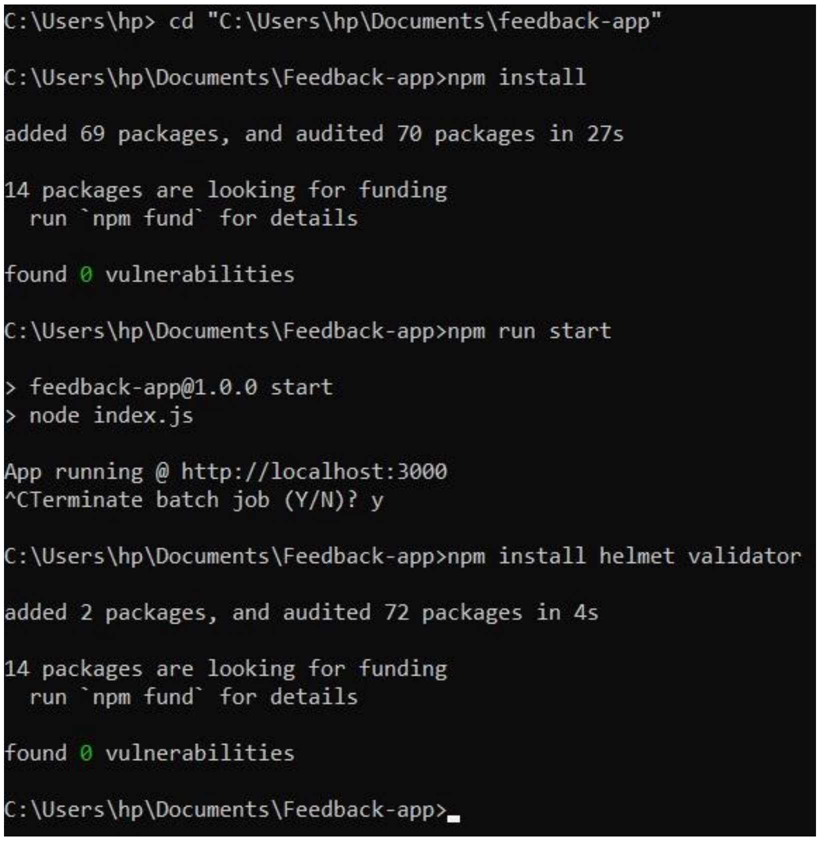
localhost:3000 says

XSS

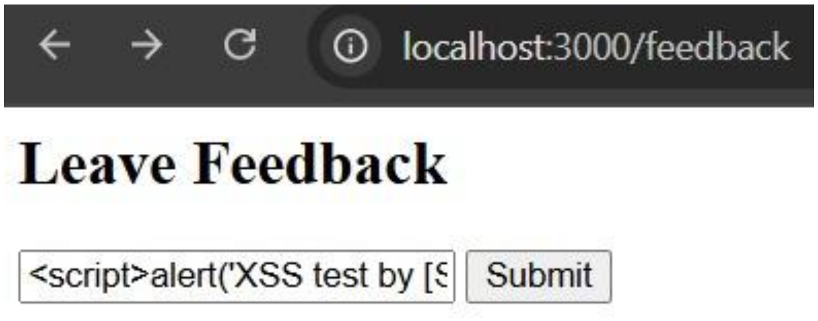
Logging with Winston

Logging all authentication attempts and errors enables monitoring and forensic analysis in case of security incidents.

Implemented:



```
C:\Users\hp> cd "C:\Users\hp\Documents\feedback-app"

C:\Users\hp\Documents\Feedback-app>npm install

added 69 packages, and audited 70 packages in 27s

14 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

C:\Users\hp\Documents\Feedback-app>npm run start

> feedback-app@1.0.0 start
> node index.js

App running @ http://localhost:3000
^CTerminate batch job (Y/N)? y

C:\Users\hp\Documents\Feedback-app>npm install helmet validator

added 2 packages, and audited 72 packages in 4s

14 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

C:\Users\hp\Documents\Feedback-app>
```



localhost:3000/feedback

Leave Feedback

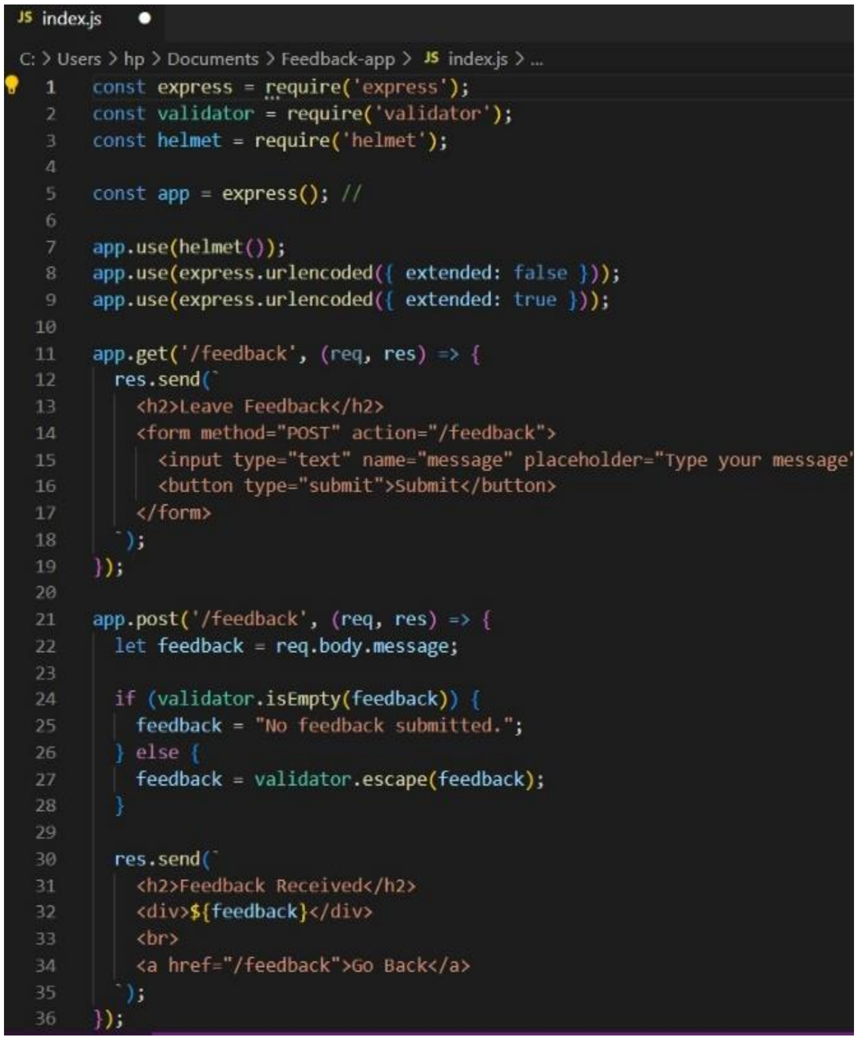
<script>alert('XSS test by [S

Security Checklist

A security checklist acts as a quick reference for all best practices:

- Input validation
- Secure password hashing

- HTTPS enabled
- Proper logging
- Use of security headers
- Token-based authentication



index.js

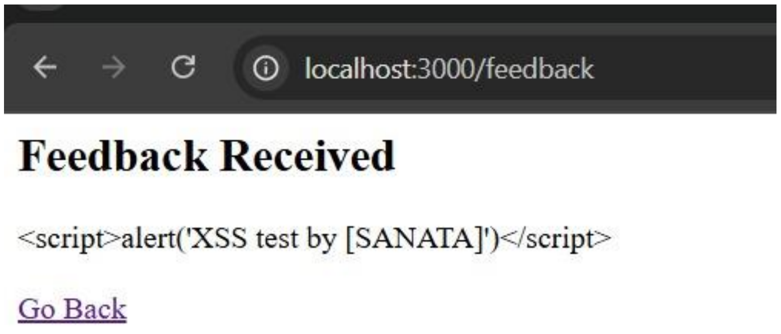
C: > Users > hp > Documents > Feedback-app > index.js > ...

```
const express = require('express');
const validator = require('validator');
const helmet = require('helmet');

const app = express(); //

app.use(helmet());
app.use(express.urlencoded({ extended: false }));
app.use(express.urlencoded({ extended: true }));

app.get('/feedback', (req, res) => {
  res.send(`
    <h2>Leave Feedback</h2>
    <form method="POST" action="/feedback">
      <input type="text" name="message" placeholder="Type your message'
      <button type="submit">Submit</button>
    </form>
  `);
});

app.post('/feedback', (req, res) => {
  let feedback = req.body.message;

  if (validator.isEmpty(feedback)) {
    feedback = "No feedback submitted.";
  } else {
    feedback = validator.escape(feedback);
  }

  res.send(`
    <h2>Feedback Received</h2>
    <div>${feedback}</div>
    <br>
    <a href="/feedback">Go Back</a>
  `);
});
```

Final Application Health

I verified the app after all patches:

- XSS attempts failed.
- SQL Injection failed.
- Sensitive data was encrypted.
- Security headers and HTTPS were in place.
- Authentication was token-based and logged properly.



Feedback Received

<script>alert('XSS test by [SANATA]')</script>

Go Back

Conclusion

Through this internship task:

- I learned to recognize web app vulnerabilities.
- Implemented standard protections (`validator.js`, `bcrypt`, `helmet`, `jsonwebtoken`).
- Strengthened the app against basic penetration tests.
- Implemented security logging for continuous monitoring.

This hands-on work prepared me for real-world web app security tasks.
Next Steps:
Publish the code and final report on GitHub.