

Week 3 Report – Advanced Security & Final Reporting

Application: User Management System (Express + MongoDB) **Prepared by:** Muhammad Raza **Date:** February 25, 2026

1. Executive Summary

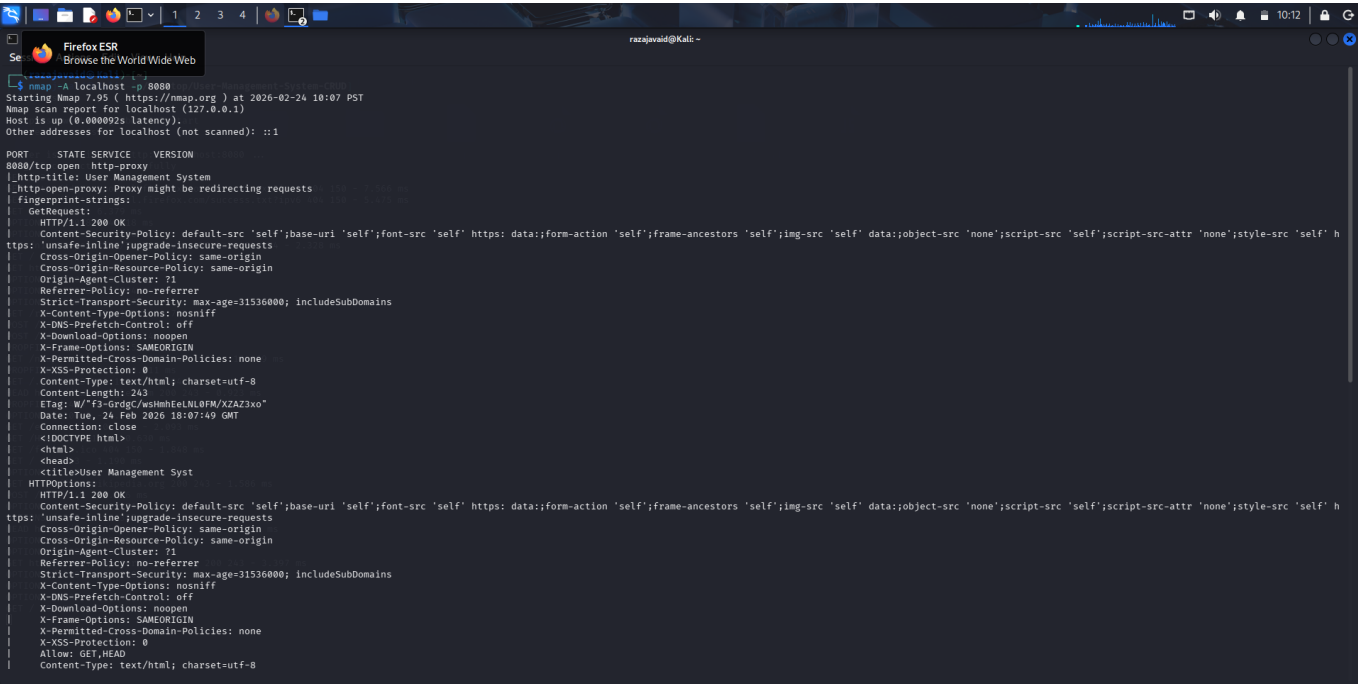
In Week 3 the team concentrated on advanced validation, penetration testing, logging/monitoring and final documentation. Key deliverables included:

- Penetration testing (network and browser-based).
- Winston logging integration.
- Security checklist and repository clean-up.
- Formal reports and evidence packaging.

Area	Outcome
Network scan	Nmap confirmed only port 8080 open; Helmet headers present
Browser tests	XSS & NoSQL injection blocked; password policy enforced
Logging	Requests & errors recorded in security.log
Documentation	Checklist and three reports completed

2. Penetration Testing

2.1 Nmap Scan



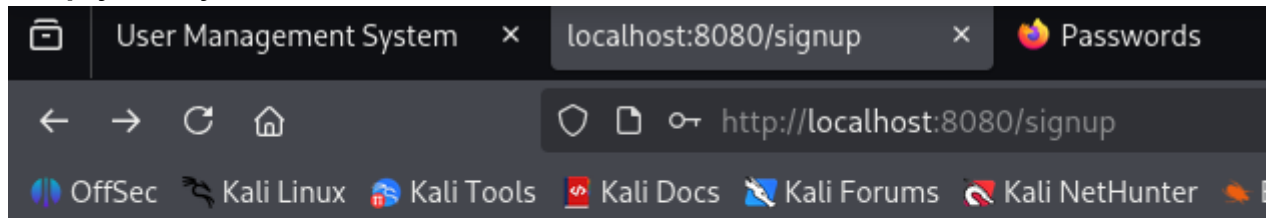
- Port **8080** open serving the application.

- Headers detected: **Content-Security-Policy**, **Strict-Transport-Security**, **X-Frame-Options**, etc.
- Confirms Helmet middleware is active and only the intended service is exposed.

2.2 Browser-Based Testing

The screenshots are now displayed inline with descriptions to improve readability:

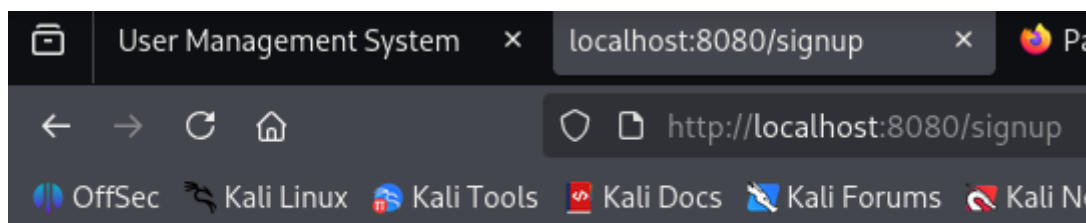
- **XSS payload injection**



Signup

The input field containing `<script>alert('XSS')</script>` was rejected, so no script execution occurred.

- **Invalid name characters**

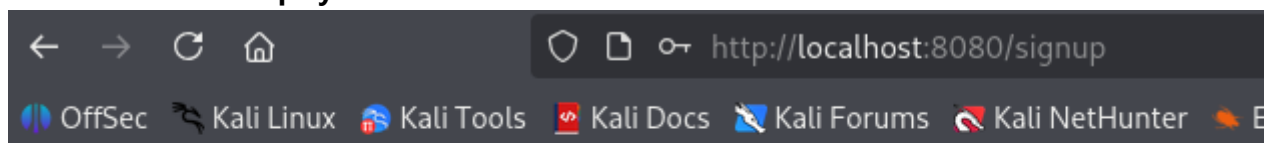


Invalid name

Submitting a

name with prohibited symbols produced an *Invalid name* error.

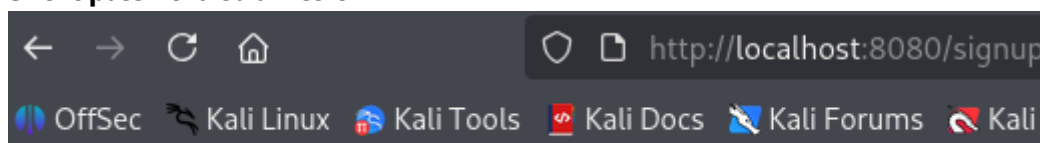
- **Weak credentials displayed**



Signup

The browser warned that the supplied password was weak.

- **Short password submission**

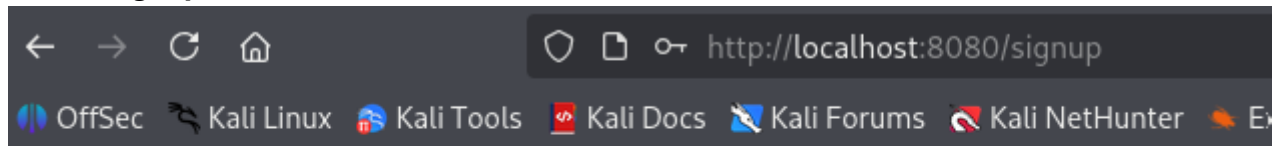


Password must be at least 8 characters

Trying to register

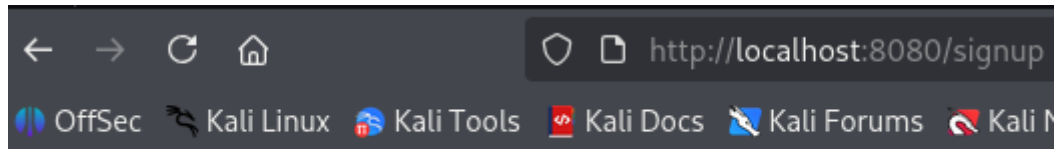
with `12345` triggered the message *Password must be at least 8 characters*.

- **Correct signup form**



Signup

The form populated as expected with valid values.

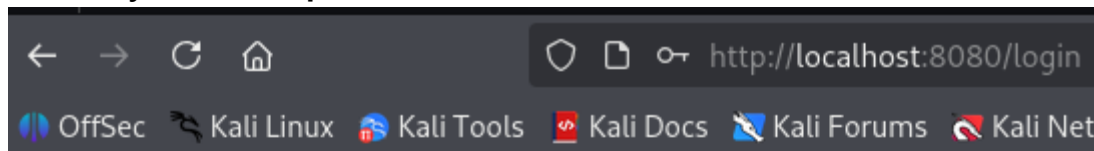


Signup successful: success

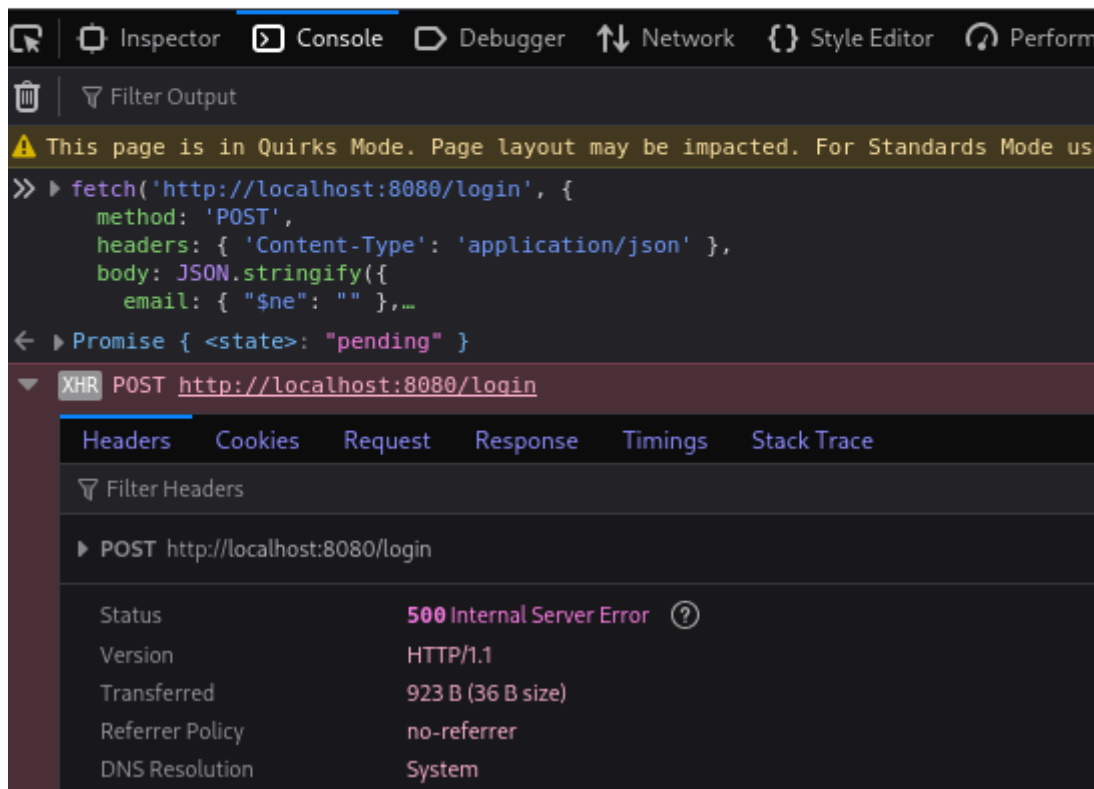
- **Successful signup**

The confirmation displayed *Signup successful: success* after valid input.

- **NoSQL injection attempt in console**



Login

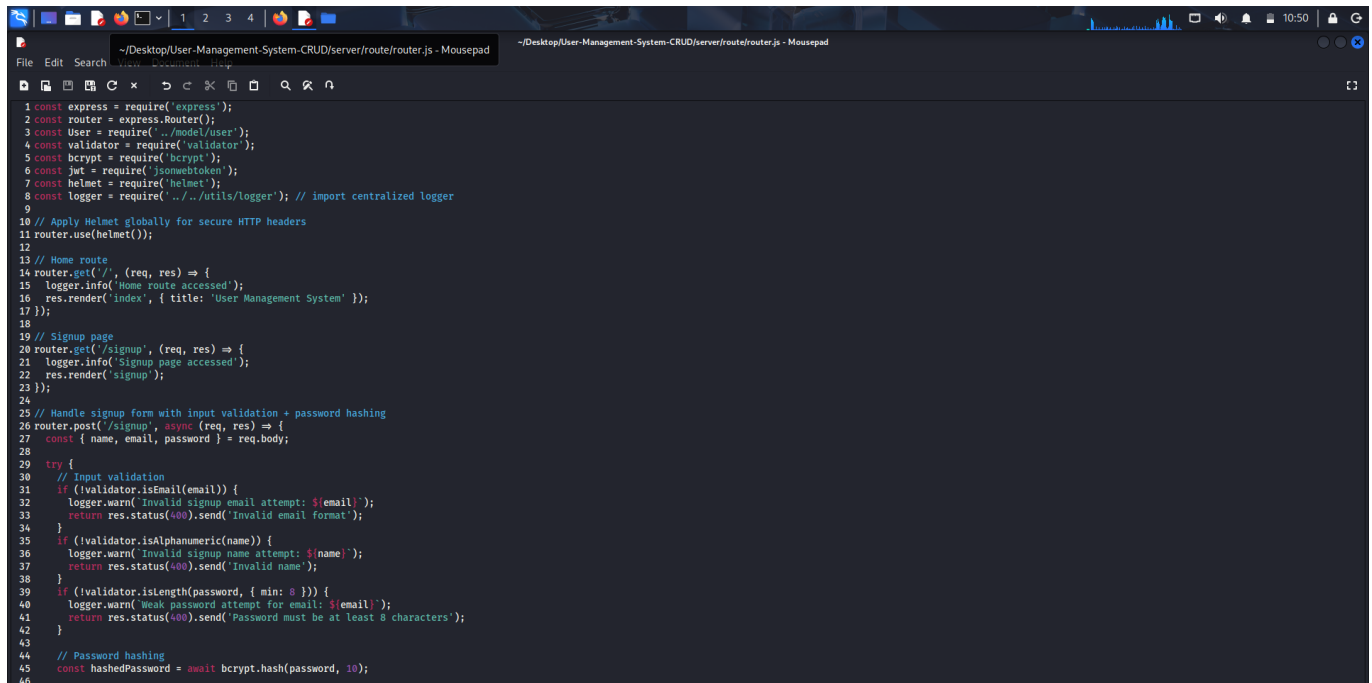
 

A crafted

payload caused a 500 error, indicating the server rejects malformed query objects.

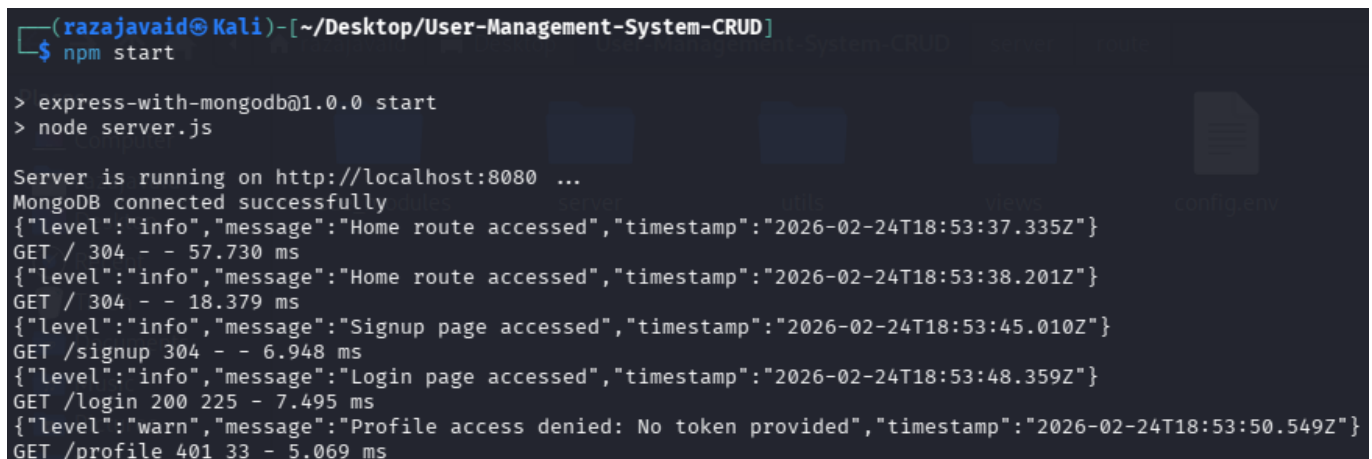
Each screenshot confirms that input validation, CSP, and sanitisation defend against common browser-based attacks.

3. Logging & Monitoring



```
1 const express = require('express');
2 const router = express.Router();
3 const User = require('../model/user');
4 const validator = require('validator');
5 const bcrypt = require('bcrypt');
6 const jwt = require('jsonwebtoken');
7 const helmet = require('helmet');
8 const logger = require('../utils/logger'); // import centralized logger
9
10 // Apply Helmet globally for secure HTTP headers
11 router.use(helmet());
12
13 // Home route
14 router.get('/', (req, res) => {
15   logger.info('Home route accessed');
16   res.render('index', { title: 'User Management System' });
17 });
18
19 // Signup page
20 router.get('/signup', (req, res) => {
21   logger.info('Signup page accessed');
22   res.render('signup');
23 });
24
25 // Handle signup form with input validation + password hashing
26 router.post('/signup', async (req, res) => {
27   const { name, email, password } = req.body;
28
29   try {
30     // Input validation
31     if (!validator.isEmail(email)) {
32       logger.warn('Invalid signup email attempt: $email');
33       return res.status(400).send('Invalid email format');
34     }
35     if (!validator.isAlphanumeric(name)) {
36       logger.warn('Invalid signup name attempt: $name');
37       return res.status(400).send('Invalid name');
38     }
39     if (!validator.isLength(password, { min: 8 })) {
40       logger.warn('Weak password attempt for email: $email');
41       return res.status(400).send('Password must be at least 8 characters');
42     }
43
44     // Password hashing
45     const hashedPassword = await bcrypt.hash(password, 10);
46
```

- Winston added to `router.js` with log levels for routes, signup/login and errors.

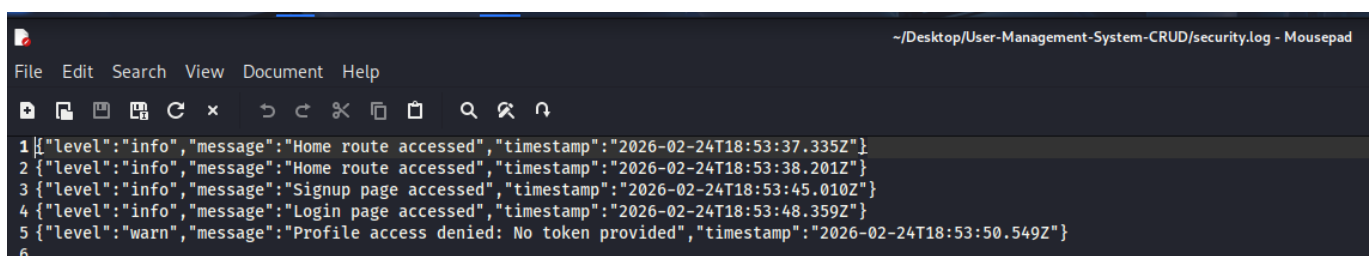


```
(razajavid@Kali) - [~/Desktop/User-Management-System-CRUD]
$ npm start

> express-with-mongodb@1.0.0 start
> node server.js

Server is running on http://localhost:8080 ...
MongoDB connected successfully
{"level":"info","message":"Home route accessed","timestamp":"2026-02-24T18:53:37.335Z"}
GET / 304 - - 57.730 ms
{"level":"info","message":"Home route accessed","timestamp":"2026-02-24T18:53:38.201Z"}
GET / 304 - - 18.379 ms
{"level":"info","message":"Signup page accessed","timestamp":"2026-02-24T18:53:45.010Z"}
GET /signup 304 - - 6.948 ms
{"level":"info","message":"Login page accessed","timestamp":"2026-02-24T18:53:48.359Z"}
GET /login 200 225 - 7.495 ms
{"level":"warn","message":"Profile access denied: No token provided","timestamp":"2026-02-24T18:53:50.549Z"}
GET /profile 401 33 - 5.069 ms
```

- Terminal output showing messages such as *Home route accessed* and *Profile access denied: No token provided*.

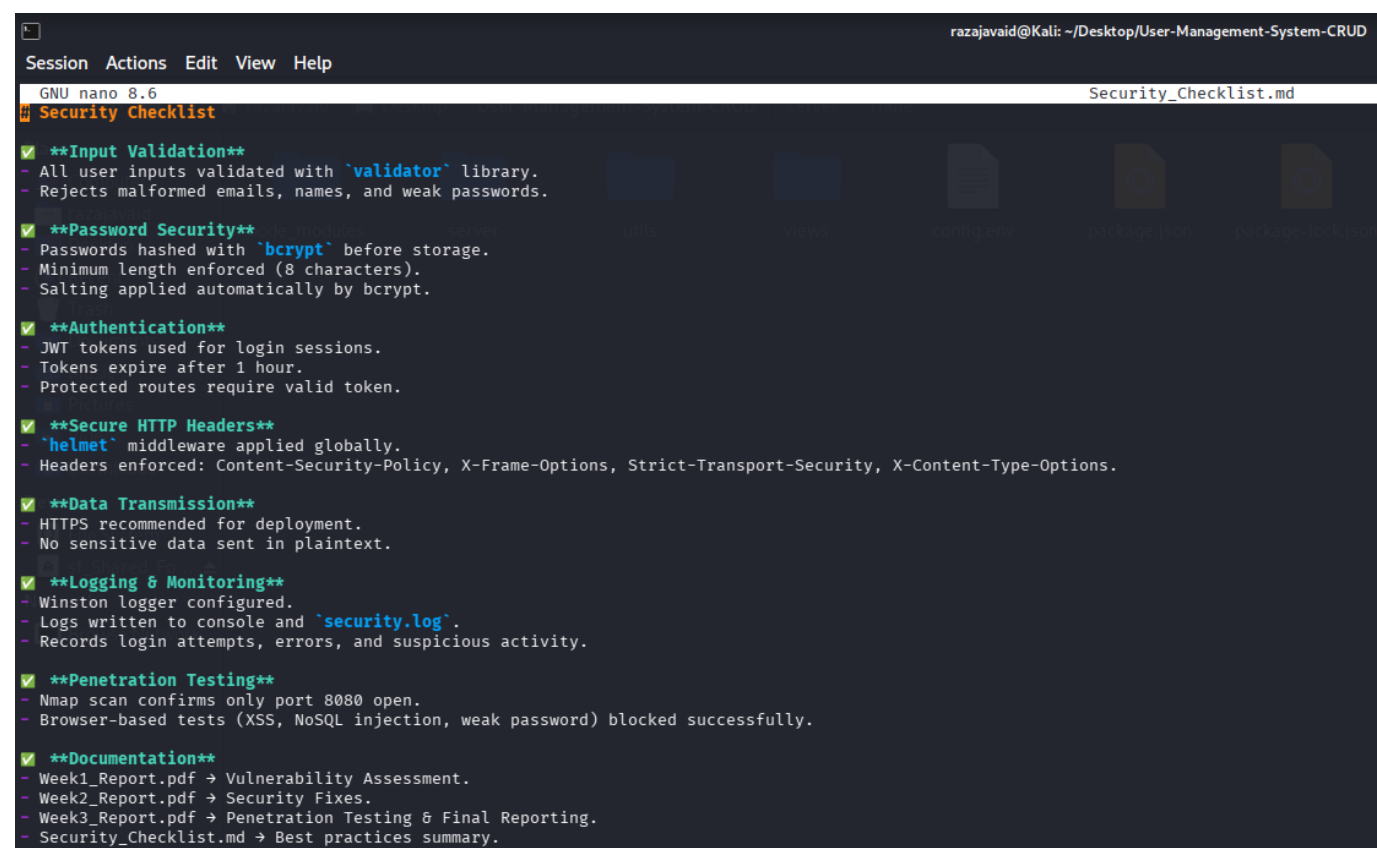


```
1 [{"level":"info","message":"Home route accessed","timestamp":"2026-02-24T18:53:37.335Z"}]
2 [{"level":"info","message":"Home route accessed","timestamp":"2026-02-24T18:53:38.201Z"}]
3 [{"level":"info","message":"Signup page accessed","timestamp":"2026-02-24T18:53:45.010Z"}]
4 [{"level":"info","message":"Login page accessed","timestamp":"2026-02-24T18:53:48.359Z"}]
5 [{"level":"warn","message":"Profile access denied: No token provided","timestamp":"2026-02-24T18:53:50.549Z"}]
6
```

- `security.log` stores structured JSON events with timestamps, levels (`info`, `warn`) and descriptive messages.

- Provides an audit trail for suspicious activity.

4. Security Checklist



The markdown checklist covers:

- Input validation & output encoding
- Password hashing & strength rules
- Authentication & authorization
- Secure headers (Helmet) and HTTPS
- Logging & monitoring
- Penetration testing procedures
- Documentation & reporting

A table of the main checklist items is included in the [Security_Checklist.md](#) file.

5. Final Submission

Deliverables prepared for review:

Item	Location / Description
GitHub repository	Complete source, configs, and README files
Week 1 report	Week1_Report.pdf – vulnerability assessment
Week 2 report	Week2_Report.pdf – security fixes

Item	Location / Description
Week 3 report	Week3_Report.pdf – this document
Security checklist	Security_Checklist.md
Screenshots	docs/screenshots folder embedded in reports

6. Conclusion

Week 3 activities validated that the security measures implemented in Week 2 are effective:

- Penetration testing showed resilience against XSS, NoSQL injection and weak passwords.
 - Logging delivered visibility into application behaviour and potential abuse.
 - Documentation artefacts (checklist and reports) complete the project lifecycle: **Identify** → **Fix** → **Validate** → **Document**.
-

7. References

- [Nmap Network Scanner](#)
- [Winston Logging Library](#)
- [OWASP Top 10 – 2021](#)
- [OWASP XSS Prevention Cheat Sheet](#)
- [Node.js Security Checklist](#)
- [Helmet.js Security](#)
- [Security checklist template](#)

Additional resources documented in the [Security_Checklist.md](#) file.