**Cybersecurity Internship Report**

**Intern Name:** Aiman Asif  
**Project Title:** Strengthening Security Measures for a Web Application  
**Submitted to:** Faizan Khan  
**Date:** 28 May, 2025

## ****Week 2: Implementing Security Measures****

### ****Objective****

To apply essential backend security measures to address previously identified vulnerabilities in the User Management System web application, including input validation, authentication enhancement, and securing server responses.

### ****Tasks Performed****

#### 1. Input Validation & Sanitization

* **Tool/Library Used:** validator
* **Implementation:**
  + Installed via npm install validator
  + Added validation checks for email, name, and password in signup and login routes.
  + Example:

js

const validator = require('validator');

if (!validator.isEmail(email)) {

return res.status(400).send('Invalid email');

}

* **Purpose:** To block malicious inputs and prevent injection-based attacks.

#### 2. Password Hashing with Bcrypt

* **Tool Used:** bcrypt
* **Note:** Already implemented in **Week 1** (as confirmed in package.json).
* **Re-verified** that:
  + bcrypt.hash() is used during signup to securely hash passwords.
  + bcrypt.compare() is used during login to verify passwords.
* **Purpose:** To ensure passwords are stored securely and cannot be easily compromised.

#### 3. Token-Based Authentication

* **Tool Used:** jsonwebtoken
* **Steps Taken:**
  + Installed via npm install jsonwebtoken
  + Used jwt.sign() to generate authentication tokens upon login.
  + Stored the token in session (req.session.token) for user route protection.
  + Middleware was updated to verify the token before granting access to protected pages.
  + Example:

js

CopyEdit

const token = jwt.sign({ id: user.\_id }, 'your-secret-key', { expiresIn: '1h' });

req.session.token = token;

* **Purpose:** To securely identify and manage user sessions.

#### 4. Securing HTTP Headers with Helmet

* **Tool Used:** helmet
* **Steps Taken:**
  + Installed via npm install helmet
  + Configured in app.js using:

js

CopyEdit

const helmet = require('helmet');

app.use(helmet());

* **Purpose:** To protect the app against common web vulnerabilities such as XSS, clickjacking, and information disclosure through secure HTTP headers.

**Impact of Security Enhancements**

| **Security Area** | **Status** | **Description** |
| --- | --- | --- |
| Input Sanitization | ✅ Implemented | All user inputs are now validated using the validator library. |
| Password Hashing | ✅ Confirmed | Bcrypt hashing and comparison logic verified and secure. |
| JWT Authentication | ✅ Implemented | Token-based login/session management fully functional. |
| Helmet.js Integration | ✅ Completed | App now sends secure HTTP headers by default. |

**Conclusion**

By the end of Week 2, major security gaps identified in Week 1—such as input sanitization and authentication handling—were effectively mitigated. The application now includes proper user data validation, secure session tokens, and improved protection via HTTP headers. These changes significantly reduce the risks of XSS, injection attacks, and insecure data exposure.