

Name: Hajra Zafar

DHC-Id: 490

WEEK 3:

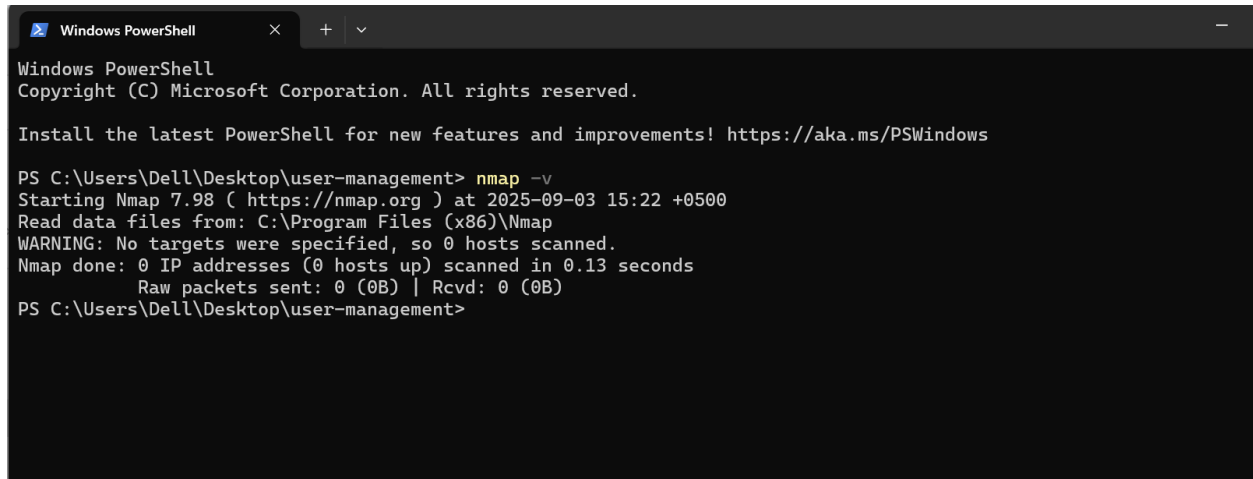
ADVANCED SECURITY AND FINAL REPORTING

1. Basic Penetration Testing

▪ Part I: Nmap Scan

FIRST WE HAVE TO

install Nmap:



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Dell\Desktop\user-management> nmap -v
Starting Nmap 7.98 ( https://nmap.org ) at 2025-09-03 15:22 +0500
Read data files from: C:\Program Files (x86)\Nmap
WARNING: No targets were specified, so 0 hosts scanned.
Nmap done: 0 IP addresses (0 hosts up) scanned in 0.13 seconds
      Raw packets sent: 0 (0B) | Rcvd: 0 (0B)
PS C:\Users\Dell\Desktop\user-management>
```

Scan Localhost:

In this we will use 2 windows powershell one for app running and other window for nmap scan:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Dell\Desktop\user-management> node index.js
App running at http://localhost:3000
```

And:

```
PS C:\Users\Dell\Desktop\user-management> nmap -p 3000 localhost
Starting Nmap 7.98 ( https://nmap.org ) at 2025-09-03 15:40 +0500
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00s latency).
Other addresses for localhost (not scanned): ::1

PORT      STATE SERVICE
3000/tcp  open  ppp

Nmap done: 1 IP address (1 host up) scanned in 0.26 seconds
```

“Nmap detected port 3000 as open. The service was labeled ppp because port 3000 is non-standard. It was manually verified as the Node.js/Express application.”

? -p 3000 → scan only port 3000

? localhost → your local machine

Part 2: Browser tests

Lets retest the vulnerabilities we faced back in week1

❖ XSS attempt at signup blocked :

⌵ | Gmail YouTube youtube Maps Translate

← → ↻ ⓘ localhost:3000/signup

⌵ | Gmail YouTube youtube Maps

✗ Invalid email format

❖ SQL Injection attempt at login failed

← → ↻ ⓘ localhost:3000/login

⌵ | Gmail YouTube youtube Maps Translate

← → ↻ ⓘ localhost:3000/login

⌵ | Gmail YouTube youtube Maps Translate

✗ Invalid credentials

2. Set Up Basic Logging:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Dell\Desktop\user-management> npm install winston

added 26 packages, and audited 120 packages in 8s

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

found 0 vulnerabilities
PS C:\Users\Dell\Desktop\user-management> |
```

Add Winston Setup in index.js:

```
JS index.js
C: > Users > Dell > Desktop > user-management > JS index.js > ...

1 const winston = require("winston");
2
3 const logger = winston.createLogger({
4   transports: [
5     new winston.transports.Console(),
6     new winston.transports.File({ filename: "security.log" })
7   ]
8 });
```

3. Log Events

In sign up route:

```
// Hash password before saving
bcrypt.hash(password, 10, (err, hashedPassword) => {
  if (err) return res.status(500).send("Error hashing password");
  logger.info(`🔒 Signup attempt for user: ${username}`);

  users.push({ username, password: hashedPassword });
  res.send("✅ User registered securely: " + username);
});


});
```

In login route:

```
});  
// Show login form  
app.get("/login", (req, res) => {  
  res.render("login");  
});  
app.post("/login", (req, res) => {  
  const { username, password } = req.body;  
  logger.info(`🔒 Login attempt for user: ${username}`);
```

Verification:


sign up with a new user :



← ↻ ⓘ localhost:3000/signup

lucy2424@gmail.com Sign Up

Log in with that user :



← → ↻ ⓘ localhost:3000/login

lucy2424@gmail.com Login

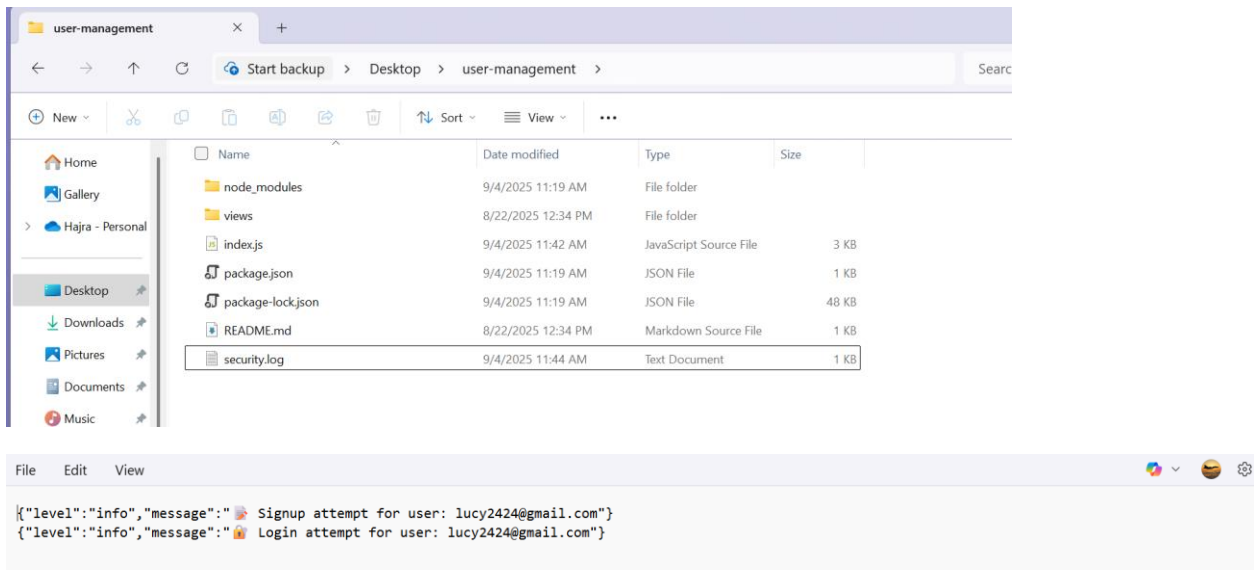
In PowerShell console I got these logs:

```

node.js v20.12.1
PS C:\Users\Dell\Desktop\user-management> node index.js
App running at http://localhost:3000
PS C:\Users\Dell\Desktop\user-management> node index.js
App running at http://localhost:3000
{"level":"info","message":"📧 Signup attempt for user: lucy2424@gmail.com"}
{"level":"info","message":"🔒 Login attempt for user: lucy2424@gmail.com"}

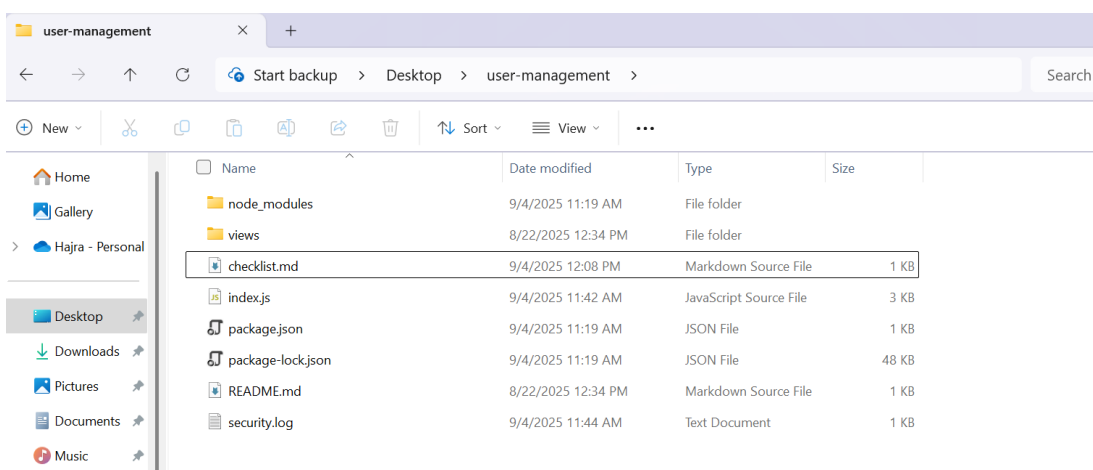
```

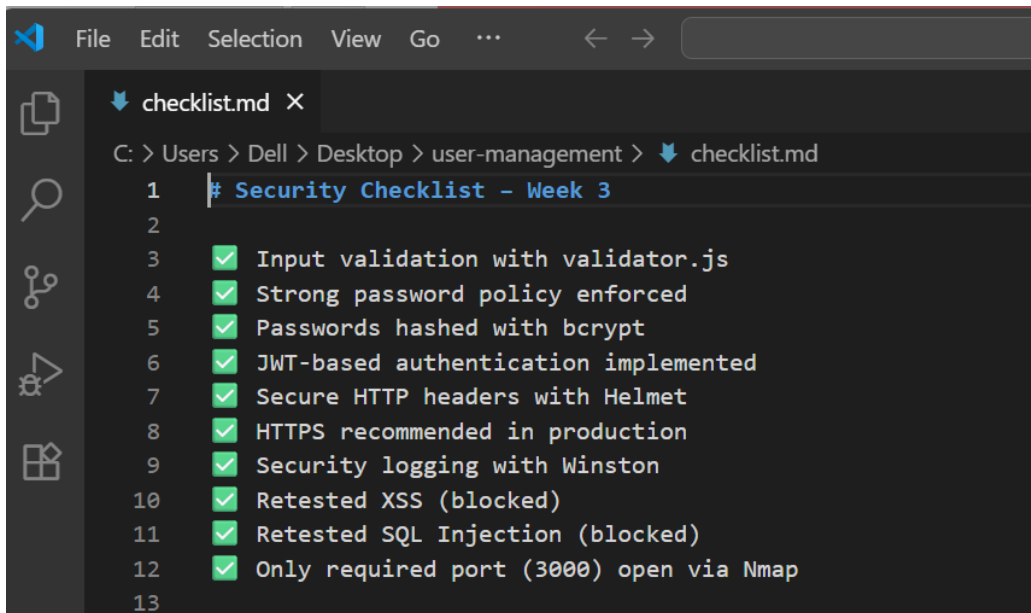
Ultimately I got a new file inside user management app called **security.log**.



3. Create a Simple Checklist

so I have added checklist.md inside my user management app:





The screenshot shows a code editor window with a dark theme. The menu bar at the top includes 'File', 'Edit', 'Selection', 'View', 'Go', and a search icon. The file explorer on the left shows a folder structure with 'checklist.md' selected. The main editor area displays the following content:

```
C: > Users > Dell > Desktop > user-management > checklist.md
1  # Security Checklist - Week 3
2
3  ✓ Input validation with validator.js
4  ✓ Strong password policy enforced
5  ✓ Passwords hashed with bcrypt
6  ✓ JWT-based authentication implemented
7  ✓ Secure HTTP headers with Helmet
8  ✓ HTTPS recommended in production
9  ✓ Security logging with Winston
10 ✓ Retested XSS (blocked)
11 ✓ Retested SQL Injection (blocked)
12 ✓ Only required port (3000) open via Nmap
13
```

Summary:

In Week 3, I validated the security fixes by performing penetration testing and confirming XSS and SQL injection attacks were blocked. I also implemented security logging with Winston to monitor critical events, and prepared a security checklist covering best practices. This completes the final phase of the internship project and demonstrates a full cycle:

assessment → remediation → validation & monitoring.
