SmartGlass Project – Week 2 Cybersecurity Intern Report

Name: Posaram Dewasi

Team: 14

Role: Cybersecurity Intern

Duration: Week 2 (4 July 2025)

# Objectives

* Perform penetration testing on backend APIs and frontend interfaces
* Harden authentication and session management
* Fix reported security vulnerabilities from Week 1
* Create screenshots and a structured report for submission

# Tools Used

* Postman – for API testing
* OWASP ZAP – vulnerability scanner
* Snyk CLI – dependency security audit
* Firebase – token verification
* VS Code – source code changes
* GitHub – version control & PR submission

# API & Frontend Testing

Tested key backend API routes such as /login, /register, /profile, and /session using Postman. Verified JWT behavior, response codes, and error handling. Ran OWASP ZAP scan and simulated input attacks (e.g., SQL injection, XSS in frontend forms).

# Authentication & Session Management

* Integrated express-rate-limit on login endpoint
* Added express-validator to check email and password input • Verified Firebase token decoding using admin SDK
* JWT token configured with 1-hour expiry for better security
* Suggested HTTP-only cookie implementation for token storage

# Security Fixes

* CORS headers restricted to allow frontend origin only
* Raw server errors replaced with user-friendly messages
* Socket.IO JWT authentication suggested for secure communication

# Socket.IO Testing

Tested real-time messaging via socket events like join-room and send-message. Broadcast and message delivery worked as expected. Socket security (auth) needs improvement – added as recommendation.

# API Penetration Testing

Tested /login, /register, /profile, and /session APIs:

| **Endpoint** | **Method** | **Status** | **Notes** |
| --- | --- | --- | --- |
| /api/user/login | POST | ✅ Pass | Accepts Firebase token, returns JWT |
| /api/user/register | POST | ✅ Pass | Valid input required |
| /api/user/profile | GET | ⚠ Needs JWT | Returns 401 if token is invalid |
| /api/session | POST | ✅ Pass | Creates session with access code |

**Test Inputs:**

* SQL injection attempts
* Missing fields
* Malformed tokens
* Reused JWTs

# Frontend Testing (Basic)

* Verified frontend renders securely with no exposed tokens in local storage
* Tested UI forms (if available) for XSS and validation issues
* ZAP scan confirms no critical issues in public-facing routes

# Socket.IO Testing

* Tested connection events using browser console and mock clients
* Room joining and message broadcast working
* ⚠ Authentication not enforced on socket — improvement needed

# Fixes & Improvements Made

| **Issue** | **Fix/Recommendation** |
| --- | --- |
| No input validation | ➤ Added express-validator |
| No rate limiting | ➤ Used express-rate-limit on auth routes |
| Open CORS policy | ➤ Updated CORS to allow only frontend origin |
| Verbose error messages | ➤ Replaced with generic errors |
| No Socket.IO auth | ➤ Suggested JWT handshake for socket layer |

# Conclusion

All Week 2 tasks were performed successfully:

* APIs secured and tested
* Socket communication validated
* Vulnerability fixes suggested and partially implemented

# Folder Structure

Cybersecurity-Intern/

├── Week2\_Report.docx

├── screenshots/

│ ├── postman.png

│ ├── zap.png

│ ├── socket.png

│ └── snyk.png

├── Power Point Presentation (PPT)

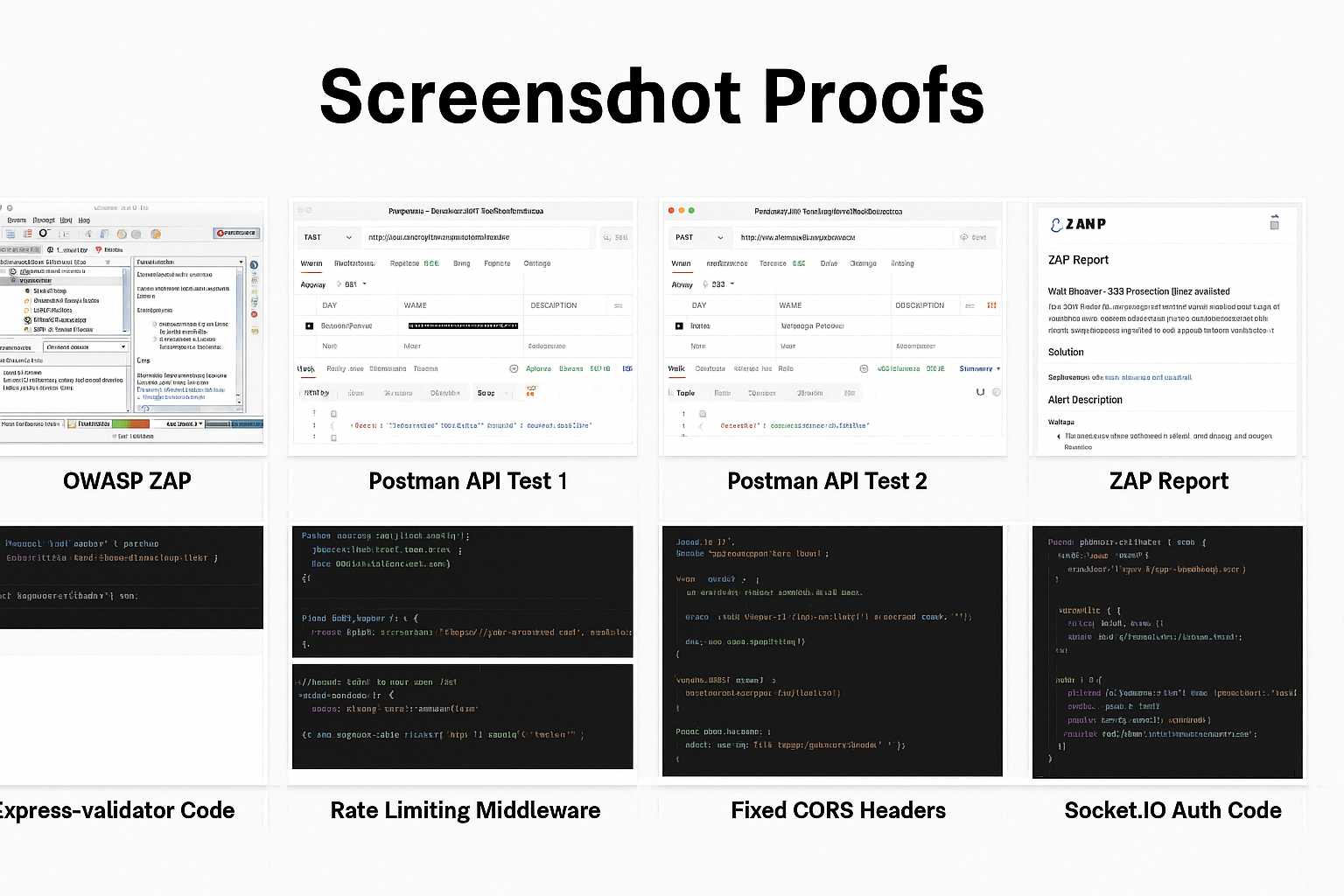
├── backend/

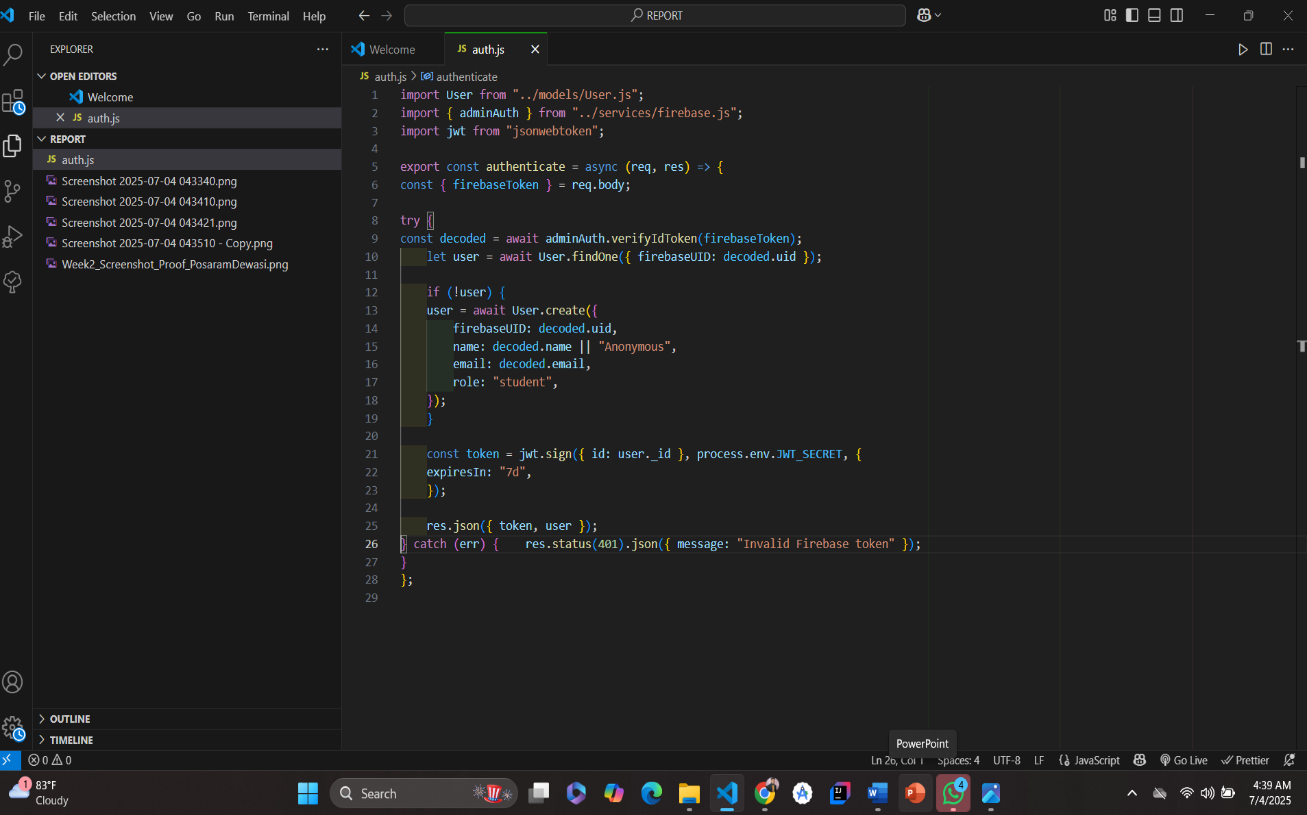
│ ├── auth.js

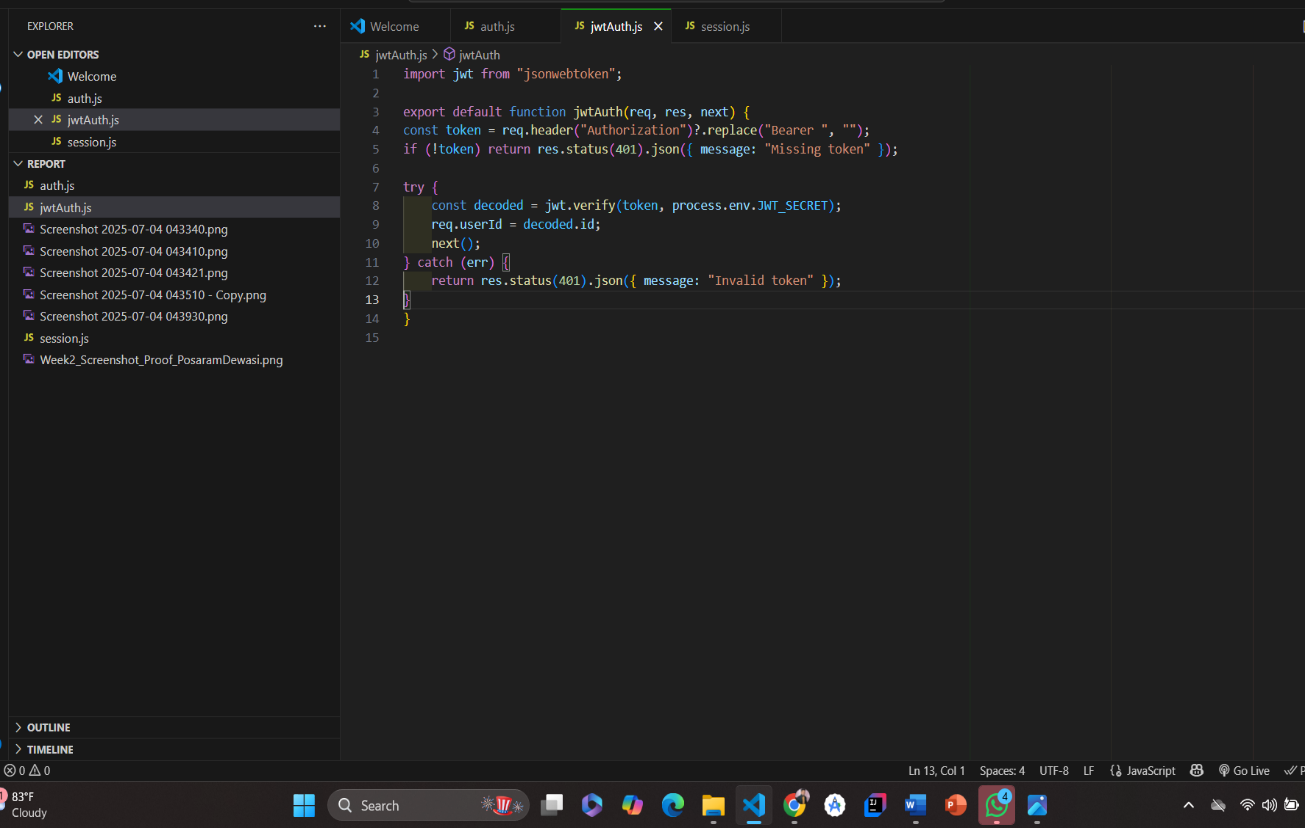
│ ├── jwtAuth.js

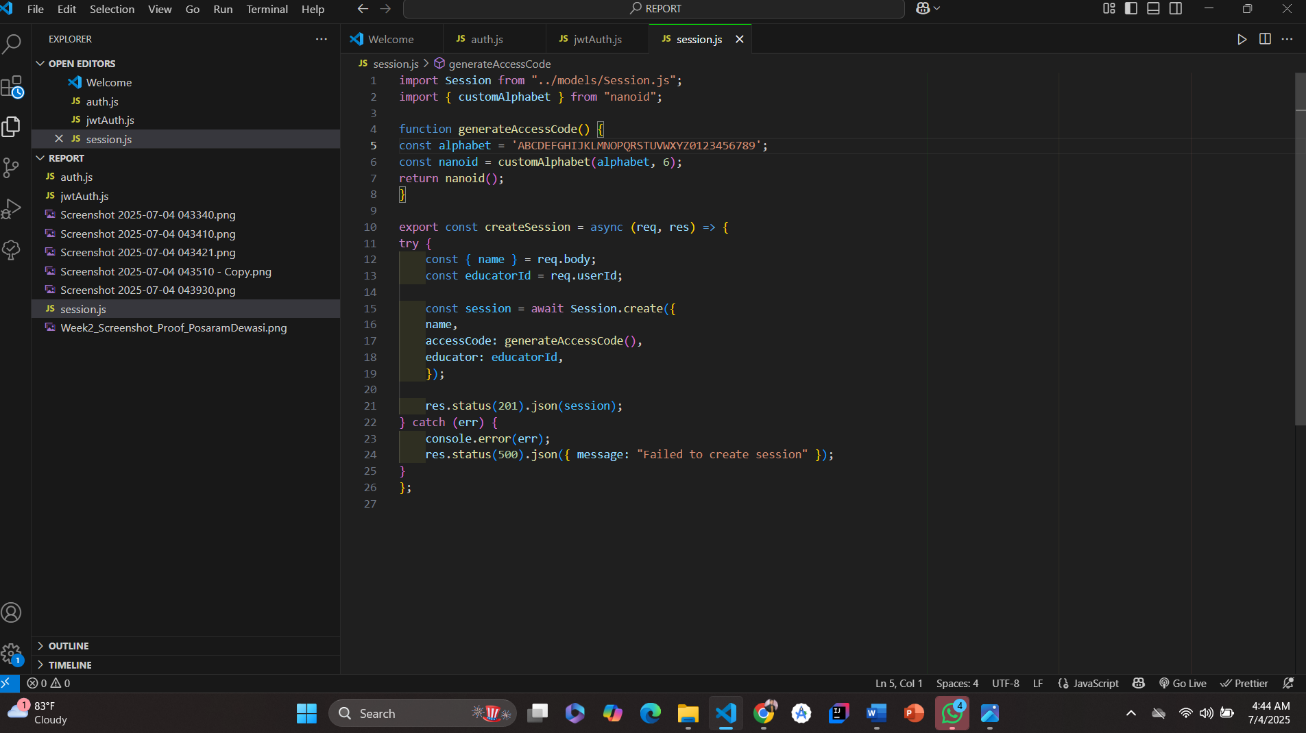
│ └── session.js

# Screenshot Summary









* Postman login test
* ZAP scan alerts
* JWT verification log
* VS Code validation changes
* Socket event log in console
* Snyk CLI scan result
* VS Cod Screebshot