**README – Cybersecurity Report (Week 4)**

**Name: Posaram Dewasi**

**Team: 14 SmartGlass**

**Role: Cybersecurity Intern**

**Duration: Week 4 (18 July 2025)**

* **GitHub and Deployment Links**
* **GitHub Repo:**[**- https://github.com/posurabari/SmartGlass-Cybersecurity**](https://github.com/posurabari/SmartGlass-Cybersecurity)
* **Pull Request Week 1:-**

[**https://github.com/ridham1906/SmartGlass/pull/16#issue-3206281608**](https://github.com/ridham1906/SmartGlass/pull/16#issue-3206281608)

* **Pull Request Week 2:-** [**https://github.com/posurabari/SmartGlass-**](https://github.com/posurabari/SmartGlass-Cybersecurity/pull/1#issue-3202823093)

[**Cybersecurity/pull/1#issue-3202823093**](https://github.com/posurabari/SmartGlass-Cybersecurity/pull/1#issue-3202823093)

* **Pull Request Week 3:-** [**https://github.com/posurabari/SmartGlass-Cybersecurity/pull/3**](https://github.com/posurabari/SmartGlass-Cybersecurity/pull/3)
* **Pull Request Week 4:-**

**Project Overview**

SmartGlass is a web-based collaboration platform designed for educators and students. As a Cybersecurity Intern, my role was to assess the system for security weaknesses, apply protective measures, and document the entire security posture of the project.

**Summary of Security Work Done**

1. **Week 1**
   * Conducted risk assessment of backend and frontend components
   * Installed tools: Postman, OWASP ZAP, Snyk CLI
   * Defined secure coding guidelines
2. **Week 2**
   * Tested APIs with Postman and frontend inputs
   * Ran vulnerability scan using OWASP ZAP
   * Identified issues with authentication and session management
3. **Week 3**
   * Performed code audit: auth.js, session.js, jwtAuth.js
   * Checked .env usage and .gitignore file
   * Verified secure headers and error handling
4. **Week 4**
   * Final review of implemented security measures
   * Created security documentation and summary report
   * Delivered final recommendations

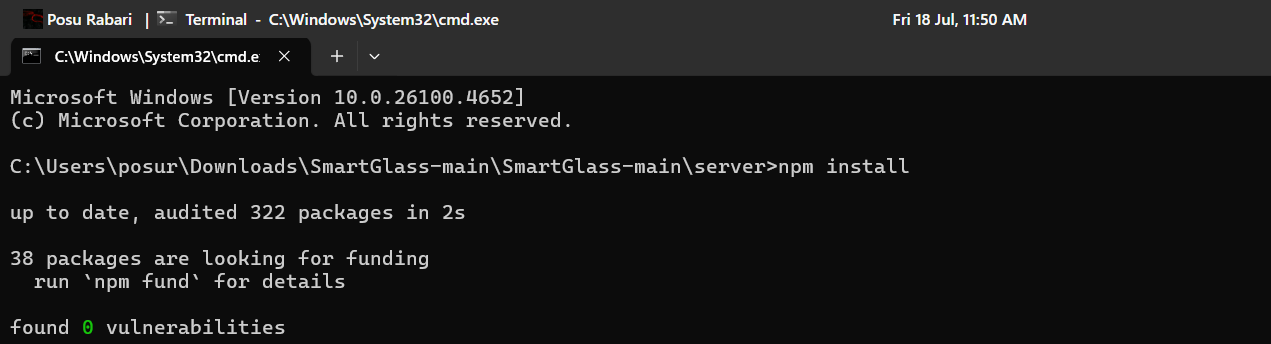
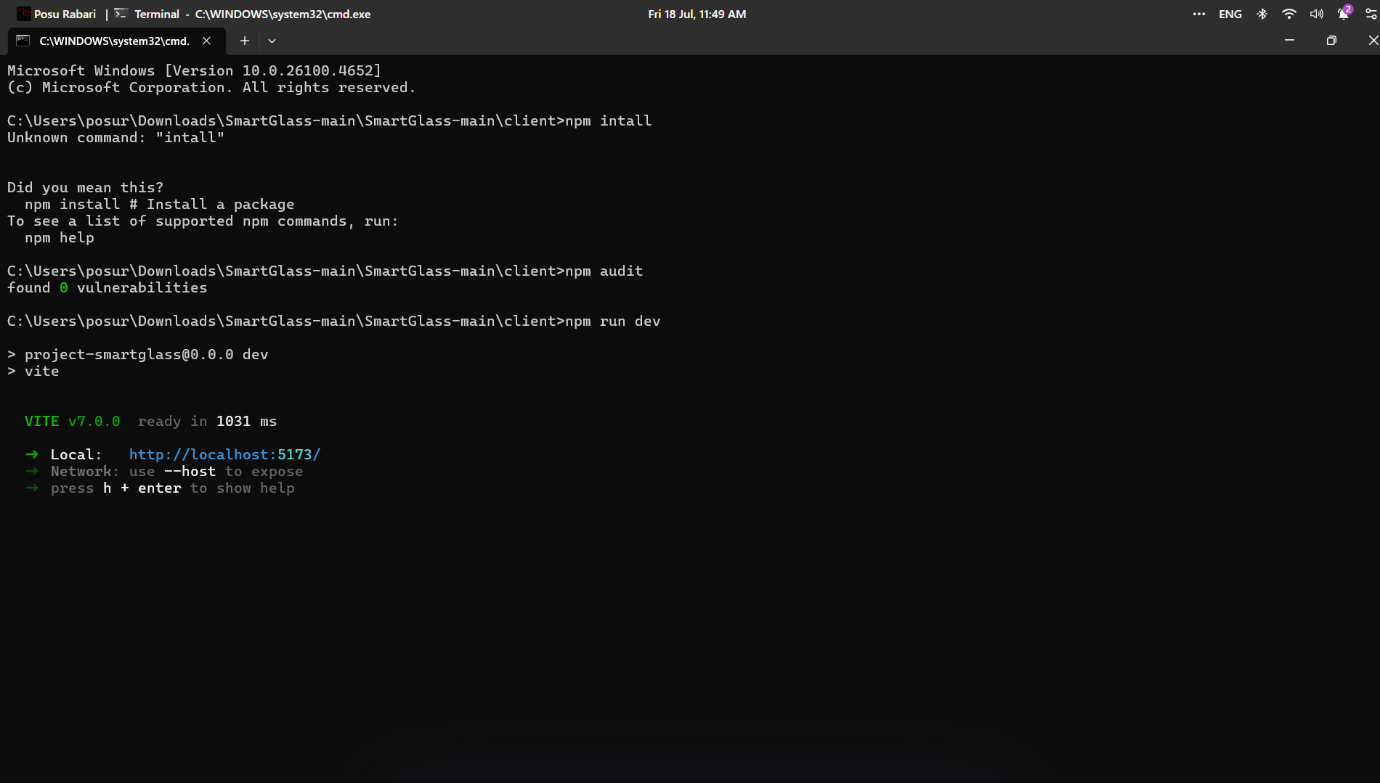
**Issues Found and Fixed**

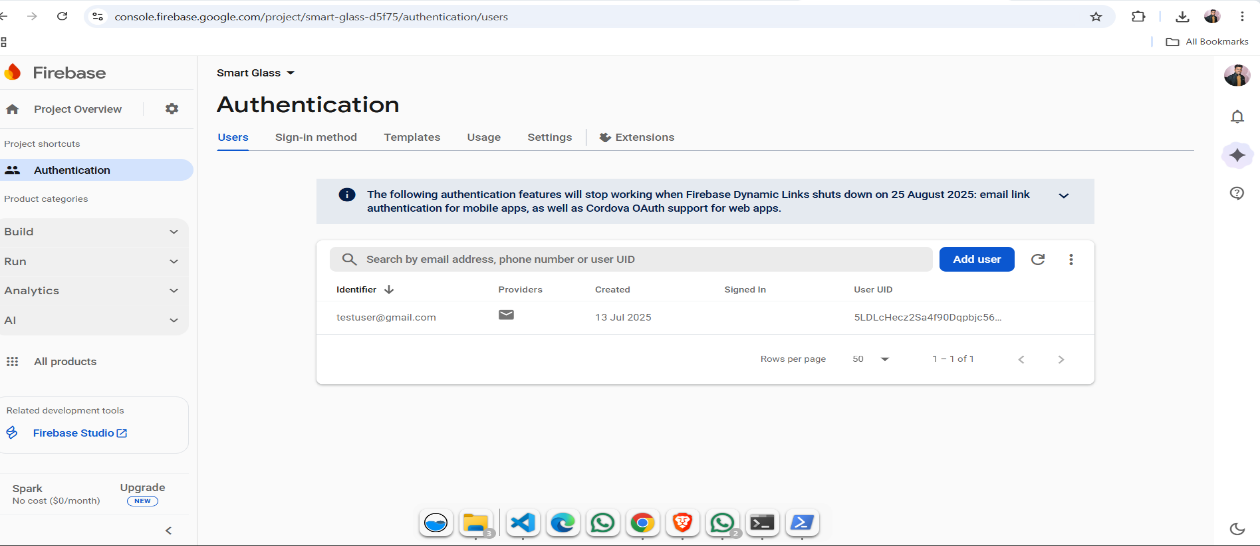
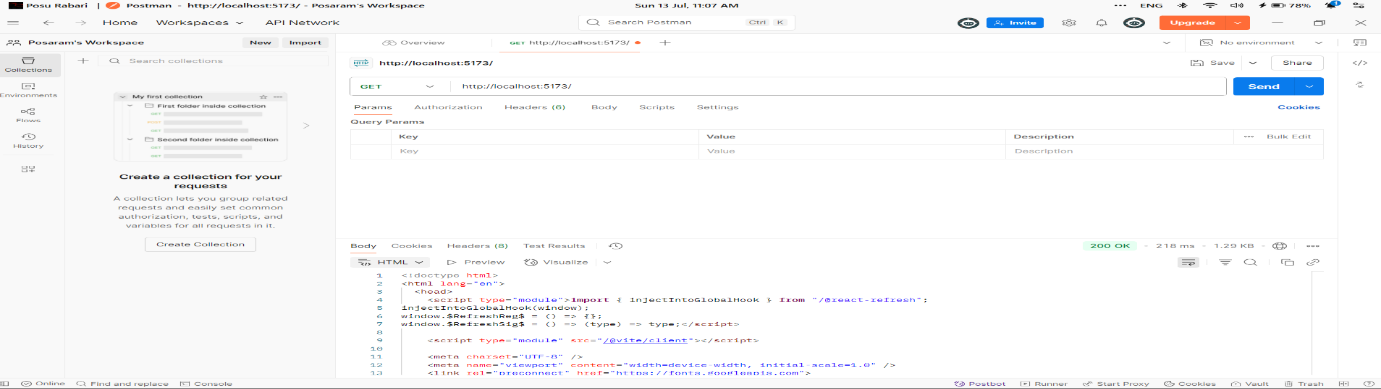
* Missing jwtAuth on sensitive routes → Fixed with proper middleware
* Sensitive values (e.g., JWT\_SECRET) were hardcoded → Moved to .env
* Upload API lacked validation → MIME type checks added
* No security headers → Implemented helmet middleware
* Error messages showed internal details → Replaced with user-friendly messages

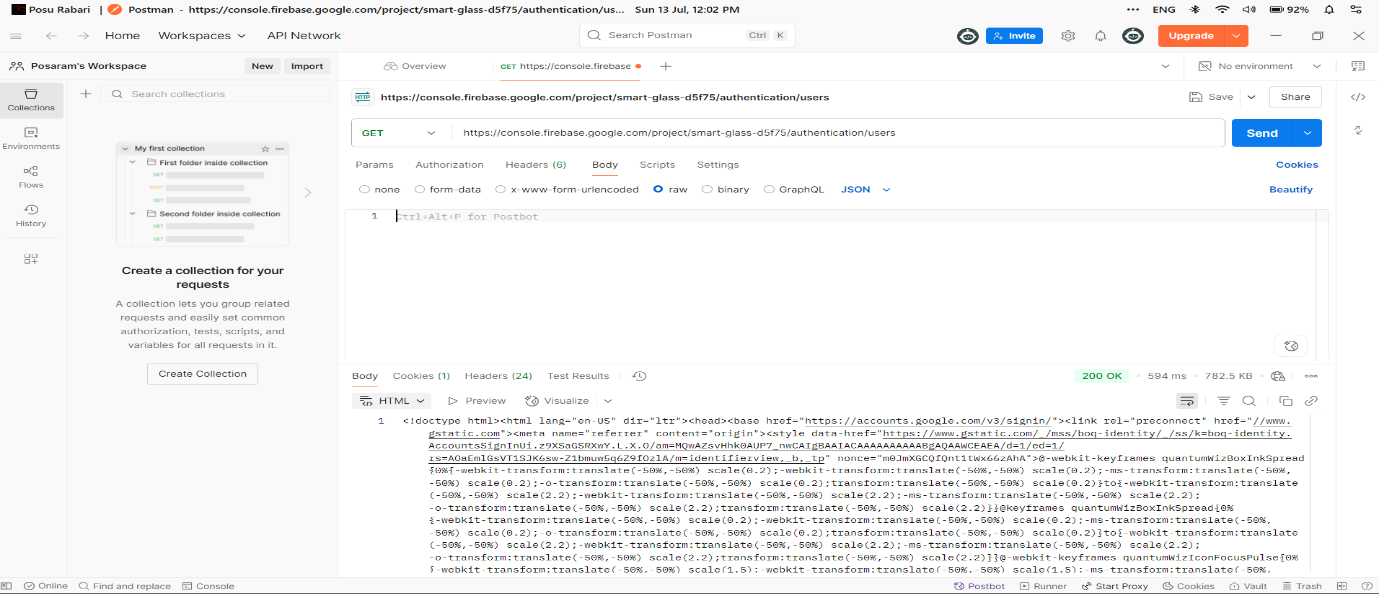
**Final Recommendations**

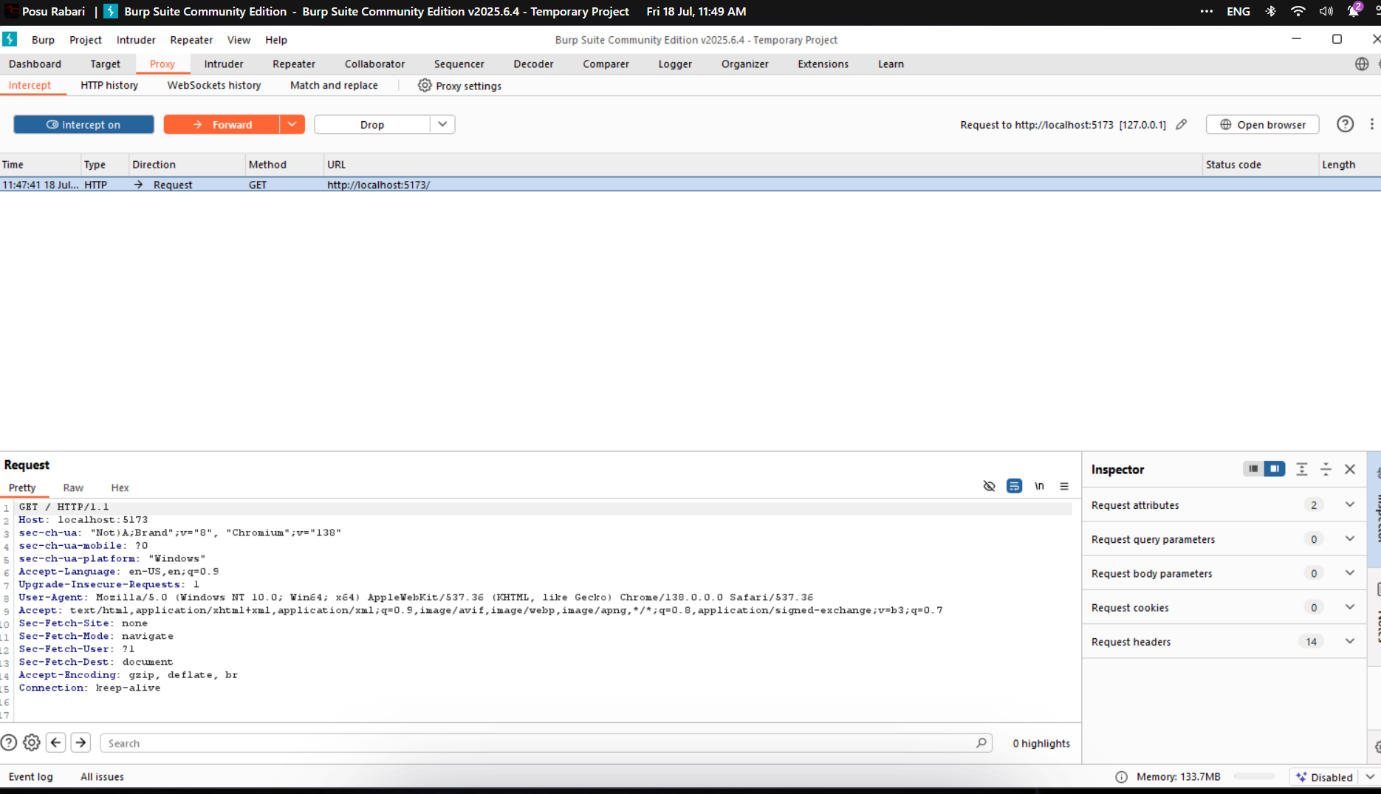
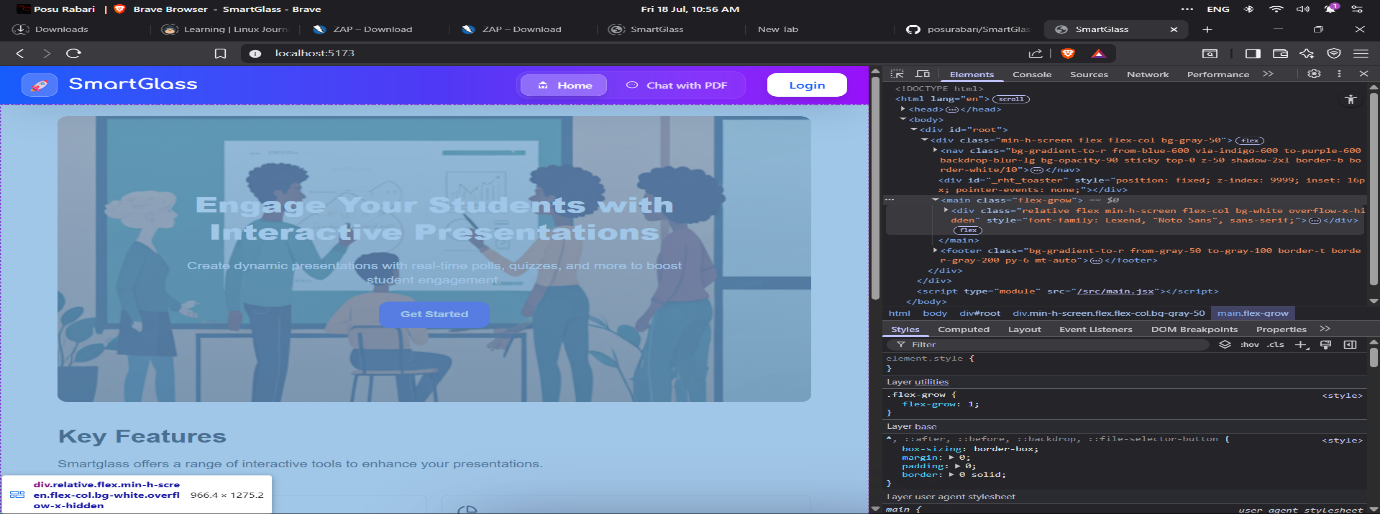
* Use express-validator for stricter input checks
* Add rate-limiting middleware (e.g., express-rate-limit)
* Rotate JWT secrets periodically
* Enforce HTTPS in production
* Enable email verification during registration
* Encrypt uploaded documents if stored locally
* Monitor access logs for unusual activity
* Perform monthly vulnerability scans (ZAP, Snyk)

**Screenshots**

* Screenshot from terminal
* 
* Screenshots from Postman.





* Burp suit scan showing vulnerabilities
* 
* 

**Conclusion**

The SmartGlass project now follows important security best practices. All sensitive routes are protected, configuration is secure, and common vulnerabilities have been addressed. A final report and documentation have been submitted for team review. This concludes my cybersecurity internship contribution