**Internship Report – Week 5**

**Title:** Ethical Hacking & Exploiting Vulnerabilities  
**Intern Name:** Karan Kumar  
**Submission Date:** July 24, 2025

**🎯 Objective of Week 5:**

The purpose of Week 5 was to gain hands-on experience with ethical hacking techniques and simulate common web vulnerabilities like SQL Injection and CSRF. This helped us learn how attackers exploit weak code — and more importantly — how to fix those weaknesses to secure real applications.

**Task 1: Ethical Hacking Basics**

Ethical hacking is the legal and authorized process of identifying and fixing system vulnerabilities. It mirrors the strategies of real attackers but is used to improve security.

**Penetration Testing Phases:**

1. **Reconnaissance** – Identify IPs, domains, open ports
2. **Scanning** – Use tools like Nmap to find vulnerabilities
3. **Enumeration** – Collect service version info, login endpoints
4. **Exploitation** – Simulate attack to confirm vulnerabilities
5. **Reporting** – Document findings and fixes

**Tools Used:** Kali Linux, Nmap, Burp Suite, OWASP ZAP

**🛠️ Practical Work:**

* Launched test app on local server
* Performed Nmap port scan
* Intercepted login request using Burp Suite
* Mapped login endpoints for testing



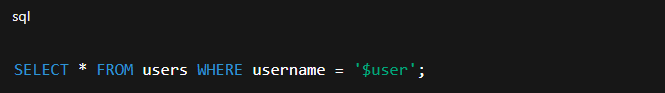




**Task 2: SQL Injection & Exploitation**

SQL Injection (SQLi) is a technique where attackers manipulate SQL queries using unsanitized inputs. This can give unauthorized access to sensitive data.

**Example:**



If input = ' OR '1'='1, the condition becomes true and login bypasses.

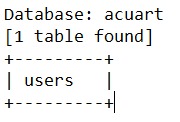
**🛡️ Prevention:**

* Prepared Statements (parameterized queries)
* Input validation
* Least privilege DB accounts

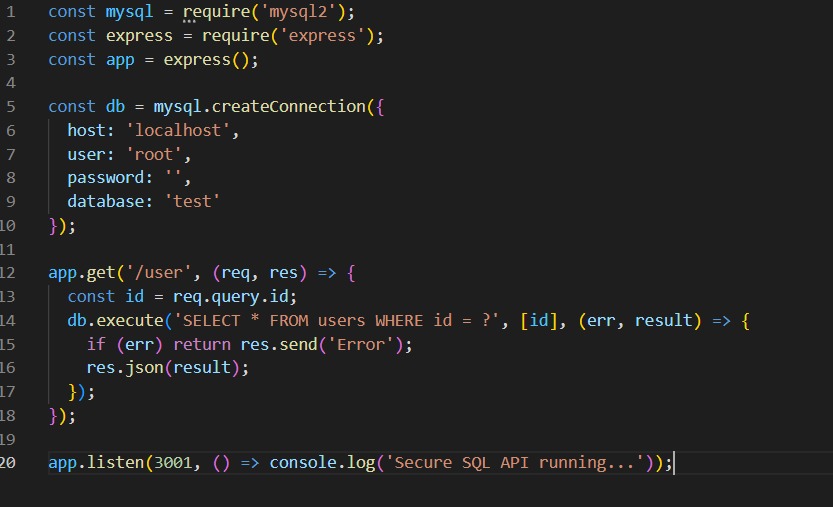
**Tool Used:** SQLMap

**🛠️ Practical Work:**

* Tested login and search fields
* Detected vulnerability with SQLMap
* Fixed code using parameterized queries
* Re-ran SQLMap to confirm fix







**Task 3: CSRF Protection**

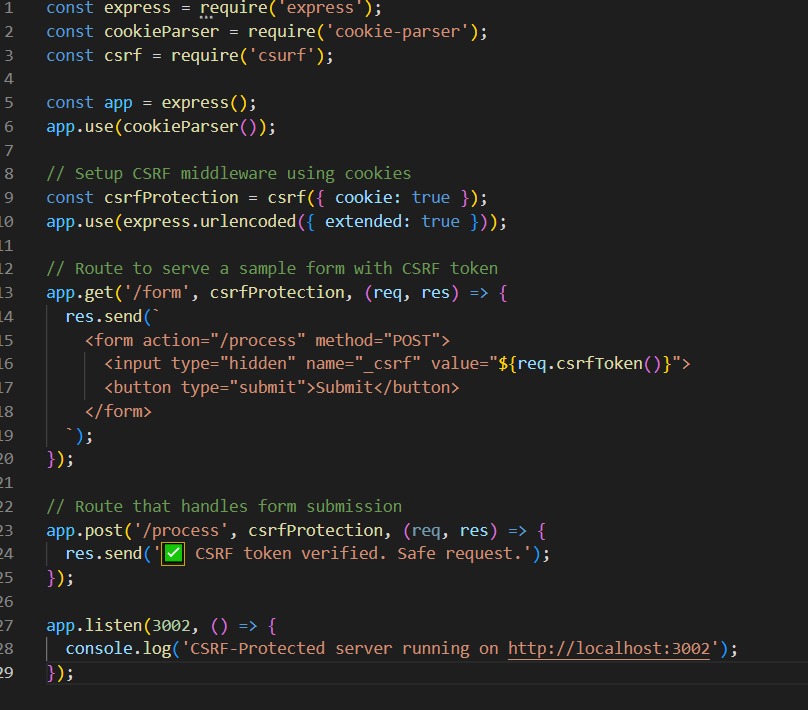
CSRF (Cross-Site Request Forgery) tricks users into submitting unauthorized actions. It uses the user's browser and cookies to perform malicious actions.

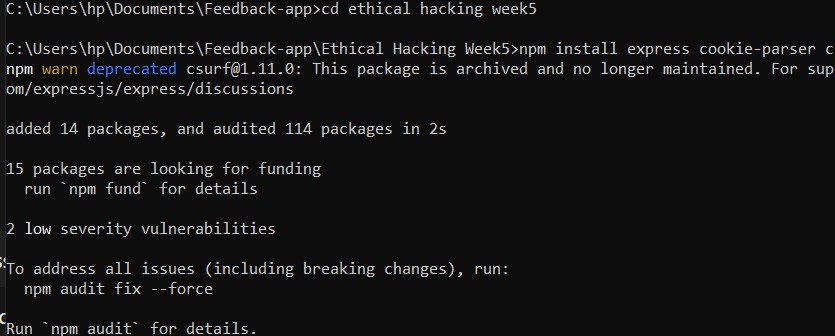
**Protection Methods:**

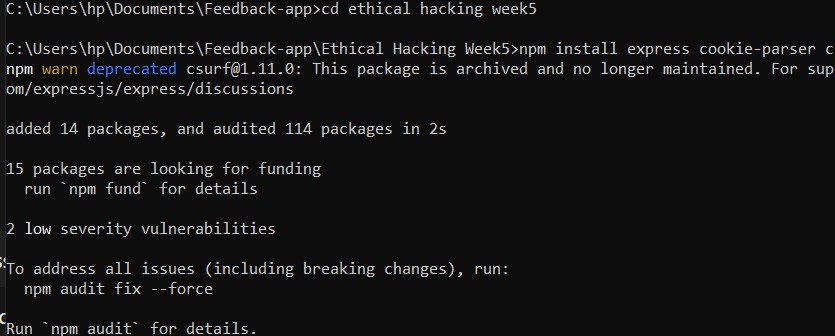
* CSRF tokens with each form
* Verify tokens on server
* Use csurf middleware in Node.js

**Practical Work:**

* Used Burp Suite to create CSRF attack
* Added csurf middleware in Express
* Tested CSRF token in form submission
* Blocked unauthorized CSRF requests









## Summary:

This week gave me real-world insight into how attackers exploit apps and how to prevent such attacks. I learned:

* How to perform ethical scans
* How SQL Injection works and how to fix it
* How to implement CSRF protection using tokens