**Key Findings – Week 1**

1. **Application Setup & Exploration**
   * Successfully cloned and set up the mock Node.js-based web application using npm install and npm start.
   * Explored core functionalities: **Signup**, **Login**, and **Profile** pages via http://localhost:3000.
   * Observed basic user interaction flows and backend API calls via browser dev tools.
2. **Basic Vulnerability Assessment**
   * **Automated Scan (OWASP ZAP)**
     + Detected multiple **low to medium severity vulnerabilities**, such as:
       - Missing security headers (X-Content-Type-Options, Content-Security-Policy).
       - Possible information leakage in HTTP response headers.
       - Outdated JavaScript libraries (potential security risk).
   * **Cross-Site Scripting (XSS) Simulation**
     + Injecting <script>alert('XSS');</script> into text input fields (e.g., profile form) triggered a JavaScript alert, confirming **Reflected XSS vulnerability** on some pages.
   * **SQL Injection Test**
     + Using the payload admin' OR '1'='1 in login fields bypassed authentication and allowed direct access to admin-level features, indicating **SQL Injection vulnerability**.
3. **Overall Risk Summary**
   * Application is vulnerable to **common OWASP Top 10** risks (XSS, SQL Injection, and missing security controls).
   * Lack of input validation and output encoding makes the app susceptible to client-side code execution.
   * Insecure database query handling allows bypassing authentication logic.