**[file:///Users/jaypatel/Downloads/login\_page%20(1).html](file:///Users/jaypatel/Downloads/login_page%20(1).html-) website link**

https://github.com/Jay200316/Jay-Computer-Networks-.git **Github link**

**Code Vault Project – Updated Technical Report**

**1. Overview**

The Code Vault project was overall enhanced with security, performance, and monitoring features to help improve its reliability and resilience. The key improvements include:

* **Security Enhancements:** HTTPS, SQL Injection (SQLi) protection, Cross-Site Scripting (XSS) mitigation, brute-force attack prevention, and even secure user sessions.
* **Performance Optimization:** Lazy loading, JavaScript and CSS minification, and Content Delivery Network (CDN) integration was used.
* **Monitoring & Security Analysis:** Google Analytics for traffic tracking, Wireshark for network analysis was attempted, and Prometheus for server monitoring.

All these enhancements can help make sure a secure, fast, and well-monitored web application.

**2. Implementation Details**

**Security Enhancements**

* **Enabled HTTPS** using a free SSL certificate from Let's Encrypt which was easy to understand.
* **SQLi & XSS Protection:** Used prepared statements and parameterized queries in PostgreSQL; also, even implemented DOMPurify to sanitize the user input.
* **Brute-Force Attack Prevention:** Implemented rate limiting using Express Rate Limit and even attempted to add Google reCAPTCHA.
* **Secure User Sessions:** Also, configured and kind of built HTTP-only and Secure cookies, even integrated CSRF tokens for protection.

**Performance Optimization**

* **Lazy Loading:** Implemented lazy loading for the images and even dynamically loaded components in React.
* **Minification:** Attempted to minified JavaScript and CSS using Terser and CSSNano.
* **CDN Integration:** I even tried to configure Cloudflare CDN to serve static assets efficiently.

**Monitoring & Security Analysis**

* **Traffic Monitoring:** I integrated Google Analytics to help track user behavior and website performance.
* **Network Security Analysis:** I even used Wireshark to help monitor network traffic and detect potential vulnerabilities.
* **Server Performance Monitoring:** I also set up Prometheus to be able to track server load and request response times.

**3. Challenges & Solutions**

**Challenge 1: CSRF Token Implementation**

* **Problem:** At first, CSRF tokens were not properly validating on the form submissions.
* **Solution:** Made sure that CSRF tokens were being generated server-side and included in API requests.

**Challenge 2: Optimizing Page Load Speed**

* **Problem:** The large assets had slowed down page loading times.
* **Solution:** I have attempted to implement lazy loading and optimized asset delivery via Cloudflare CDN.

**Challenge 3: Analyzing Network Traffic**

* **Problem:** I had identified malicious requests and security threats was challenging.
* **Solution:** I have attempted to used Wireshark to be able to filter and analyze network packets, even identifying and blocking suspicious activity.

**4. Traffic & Security Analysis**

* **Google Analytics Data:** I had identified peak traffic times and optimized caching for overall improved performance.
* **Wireshark Findings:** I had detected multiple unauthorized access attempts, which were caused by the rate limiting and firewall rules.
* **Prometheus Metrics:** I even monitored server CPU and memory usage, which lead to optimized backend queries.

**5. Screenshots of Implemented Features**

A screenshot of a login screen

AI-generated content may be incorrect.A screenshot of a login screen

AI-generated content may be incorrect.A screenshot of a login screen

AI-generated content may be incorrect.

Unit Test for Invalid Password Input

A screenshot of a computer program

AI-generated content may be incorrect.

Adding Credentials

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.