Nama : Aryadil Diangka

NIM : 20210801179

Mata Kuliah : Keamanan Informasi

Hari/tanggal : Selasa, 22 Juli 2025

1. Studi Kasus : Manajemen Inventaris Kantor Dalam Memonitoring Aset dan Peminjaman Barang.

   PT Ziru adalah sebuah Perusahaan swasta yang bergerak di bidang layanan teknologi dan memiliki banyak aset fisik berupa peralatan kantor seperti komputer, printer, meja dan perangkat pendukung lainnya. Selama ini, proses pengelolaan inventaris dan peminjaman barang dilakukan secara manual menggunakan kertas atau file excel. Peminjaman tersebut kerap kali menimbulkan masalah seperti data aset yang tidak akurat, sulit menelusuri Riwayat peminjaman barang, dan keterlambatan pembuatan laporan aset yang mengakibatkan efisiensi manajemen data aset menurun, pengeluaran anggaran berlebih dan pengambilan keputusan menjadi kurang tepat oleh bagian administrasi. Sehingga PT Ziru membutuhkan sistem berbasis web yang dapat membantu admin kantor dalam mengelola dan memantau seluruh barang yang tersedia, dipinjam, dan barang rusak.

   Analisis Studi Kasus

   Kebutuhan fungsional

   a. Manajemen data barang
      - Sistem dapat mencatat data barang inventaris yang dapat diedit, dihapus dan ditambahkan.
      - Sistem dapat menampilkan status barang yang aktif, sedang dipinjam dan rusak.

   b. Peminjaman Barang
      - Sistem dapat mencatat peminjaman barang oleh karyawan
      - Sistem dapat menyimpan Riwayat pinjaman
      - Sistem dapat memperbarui status pinjaman
      - Sistem dapat mencegah peminjaman barang yang statusnya sedang dipinjam.

   c. Laporan
      - Sistem dapat menampilkan seluruh barang.
      - Sistem dapat menampilkan laporan barang per kondisi dan lokasi.
      - Sistem dapat menampilkan Riwayat peminjaman

   Kebutuhan nonfungsional

   a. Keamanan
      - Sistem harus melindungi data barang untuk menghindari manipulasi data asset kantor agar jumlah dan status data barang tidak dapat diubah.
      - Data login harus dienkripsi dan akses sistem harus dibatasi berdasarkan hak akses.
      - Sistem juga dapat melindungi data karyawan yang sedang meminjam barang.

   b. Performa website

- Sistem harus merespons setiap aksi dalam waktu kurang dari 3 detik pada jaringan local

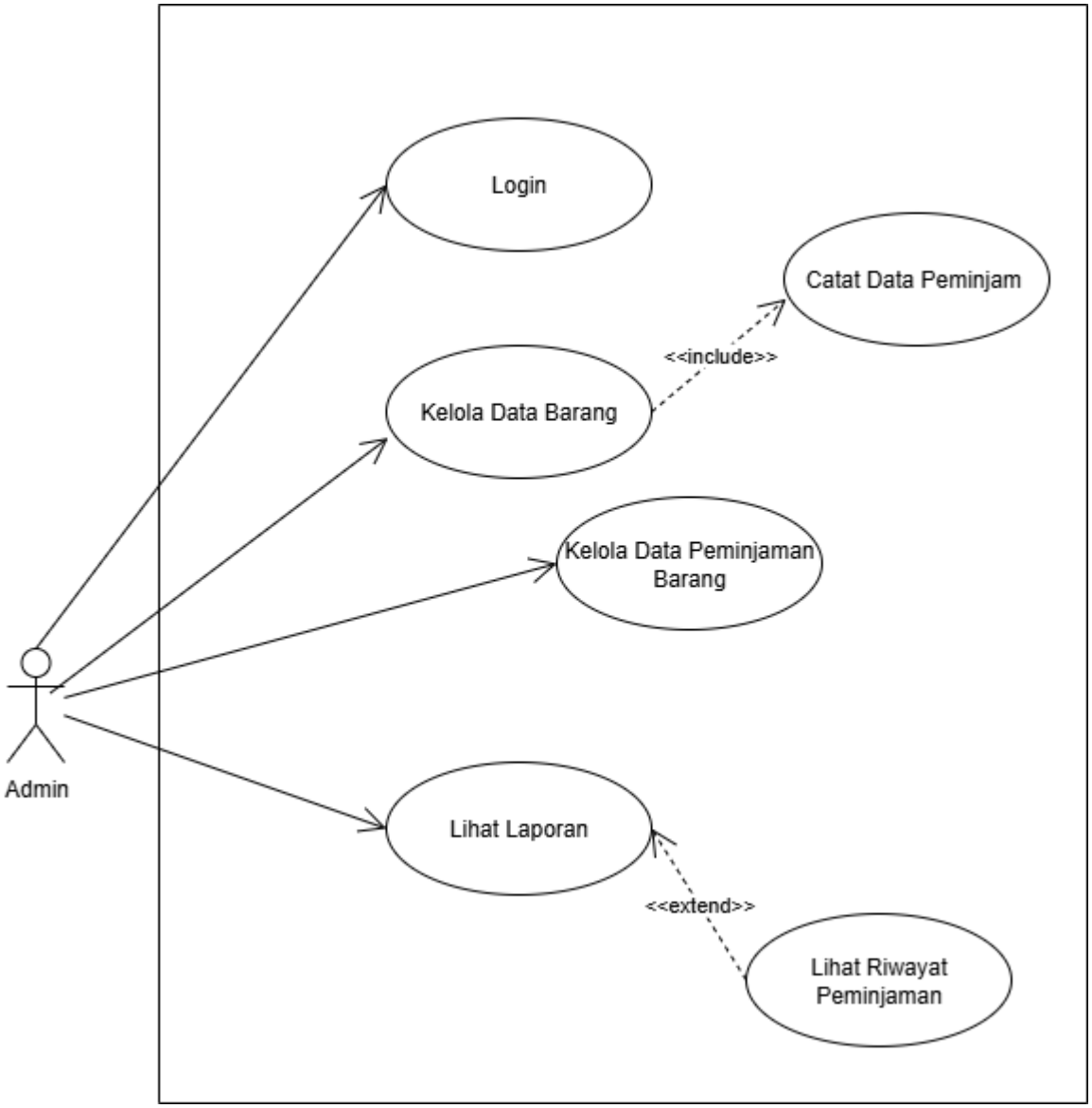
c. Ketersediaan
   - Sistem harus dapat diakses selama jam operasional kantor tanpa downtime yang mengganggu.

Actor yang terlibat dalam sistem hanyalah admin.

a. Admin
   Pengguna utama yang dapat mengelola seluruh data inventaris dan pengguna lain.

   Hak akses admin
   a. Mengelola data barang
   b. Mengelola data peminjaman barang
   c. Mengelola laporan aset dan peminjaman
   d. Melihat semua status barang
   e. Lihat data barang berdasarkan lokasi atau kondisi.
   f. Input data peminjam barang
   g. Melihat Riwayat peminjam.

Use Case Diagram




Admin

- Aktor Tunggal yang melakukan semua aktivitas sistem dari login, Kelola data barang, data peminjam dan melihat laporan.

Login

- Memasukkan username dan password untuk memonitoring aset kantor dan peminjaman barang yang telah disediakan.

Kelola data barang

- Admin dapat menambah barang baru, menghapus barang yang tidak digunakan dan mengedit barang.

Kelola data peminjaman barang

- Mencatat tanggal pinjam dan Kembali
- Mengubah status barang menjadi dipinjam, telah Kembali dan rusak.

Include catat data peminjam

- Mencatat data peminjam mulai dari nama, nomor telepon dan email.

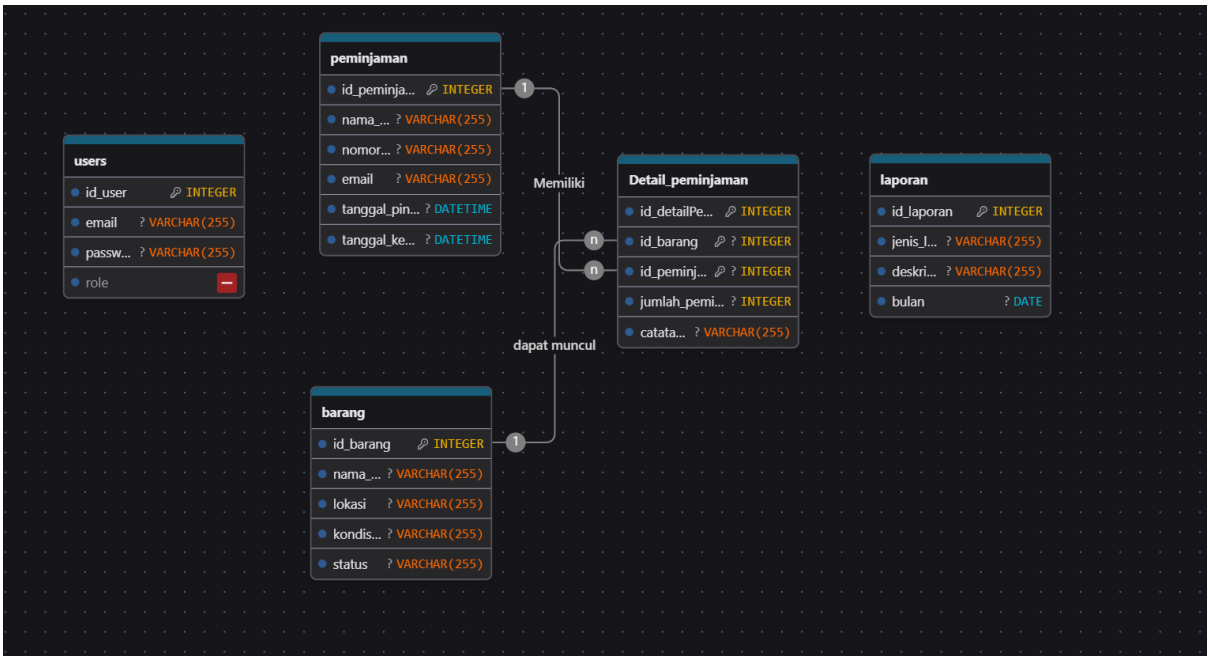
Lihat laporan

- Melihat jumlah barang
- Melihat status barang (aktif, rusak, dipinjam)

Extend lihat Riwayat peminjaman

- Melihat data peminjam barang

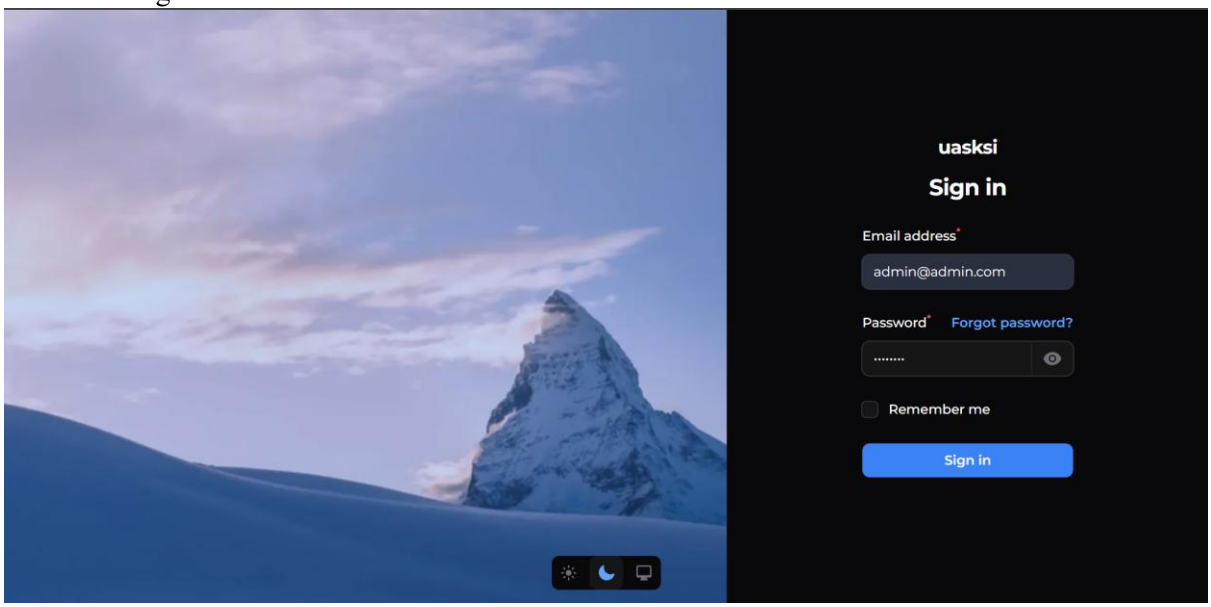
Entity Relationship Diagram



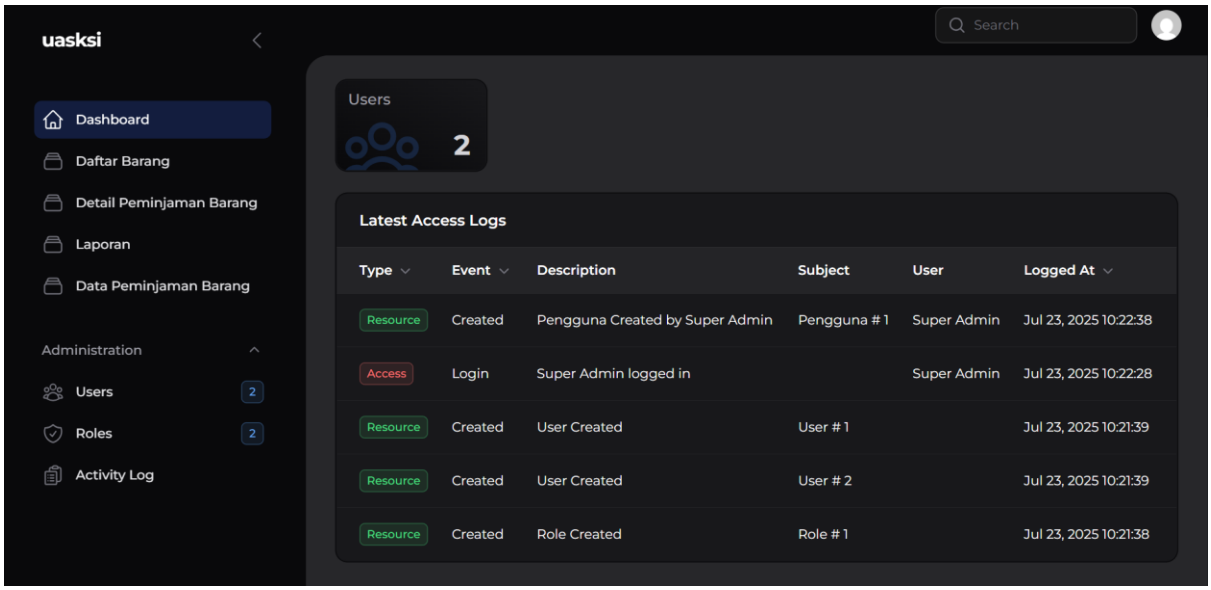

Dari gambar diatas, aplikasi Manajemen Inventaris Kantor Dalam Memonitoring Aset dan Peminjaman Barang memiliki 5 tabel yang terdiri dari *users* (pengguna), peminjaman, barang, detail peminjaman dan laporan. Dari tabel diatas memiliki 2 relasi antara peminjaman dengan detail peminjaman yang berarti satu peminjaman memiliki banyak detail peminjaman sehingga satu peminjam dapat meminjam lebih dari satu barang (*one to many*). Dan relasi antara barang dengan detail peminjaman yang berarti satu jenis barang dapat muncul di banyak detail barang (*one to many*).

2. Aplikasi Manajemen Inventaris Kantor Dalam Memonitoring Aset dan Peminjaman Barang

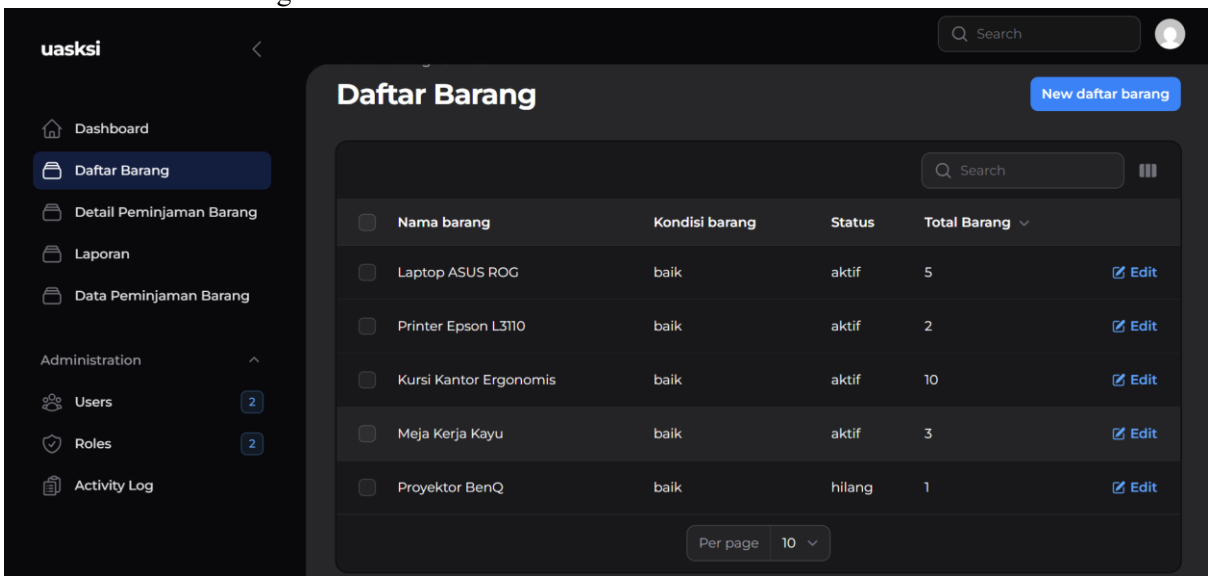
a. Form Login



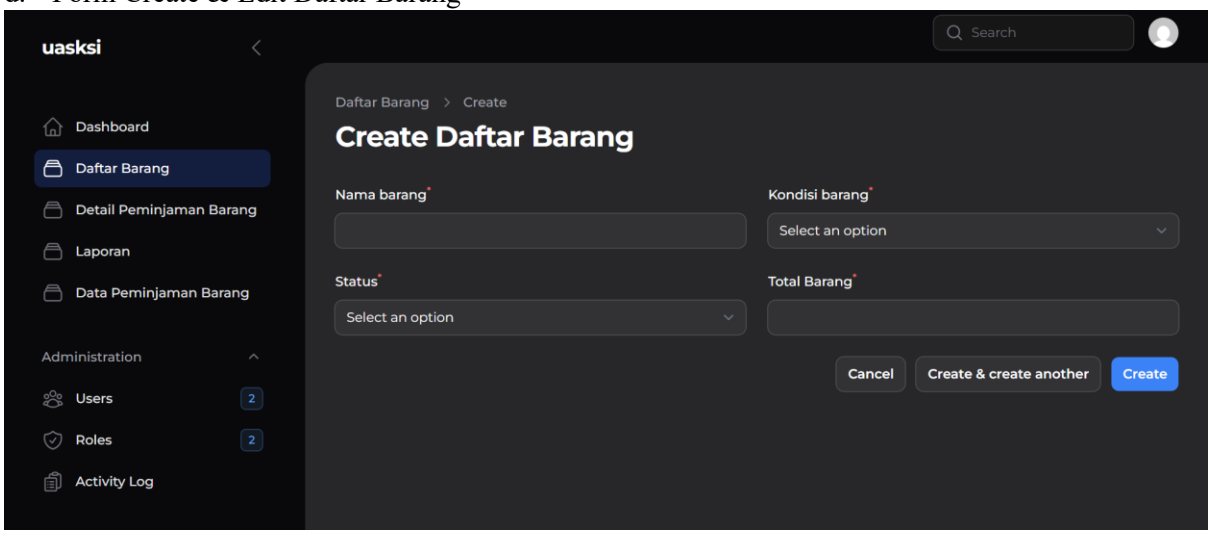

b. Dashboard

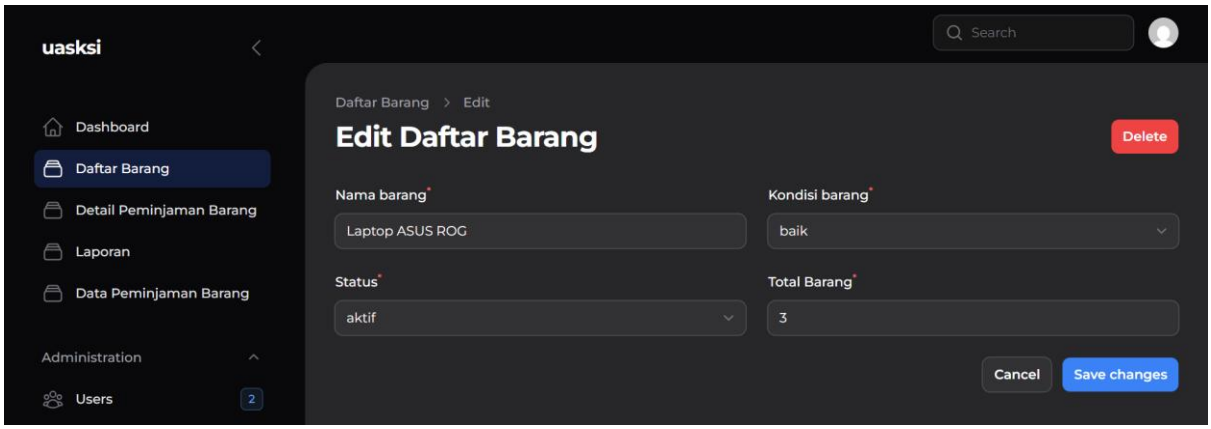


c. Tabel Daftar Barang

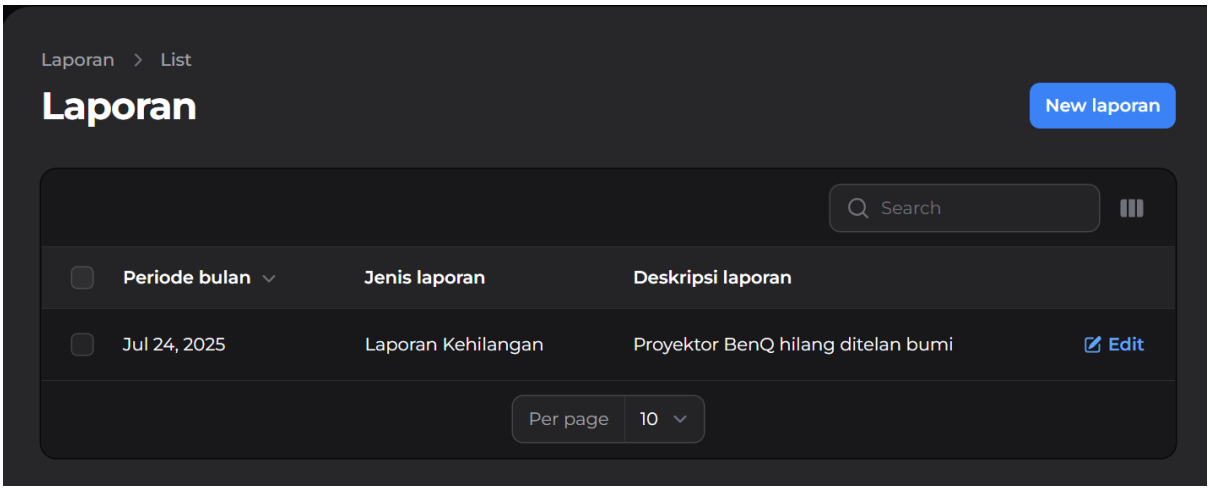



d. Form Create & Edit Daftar Barang

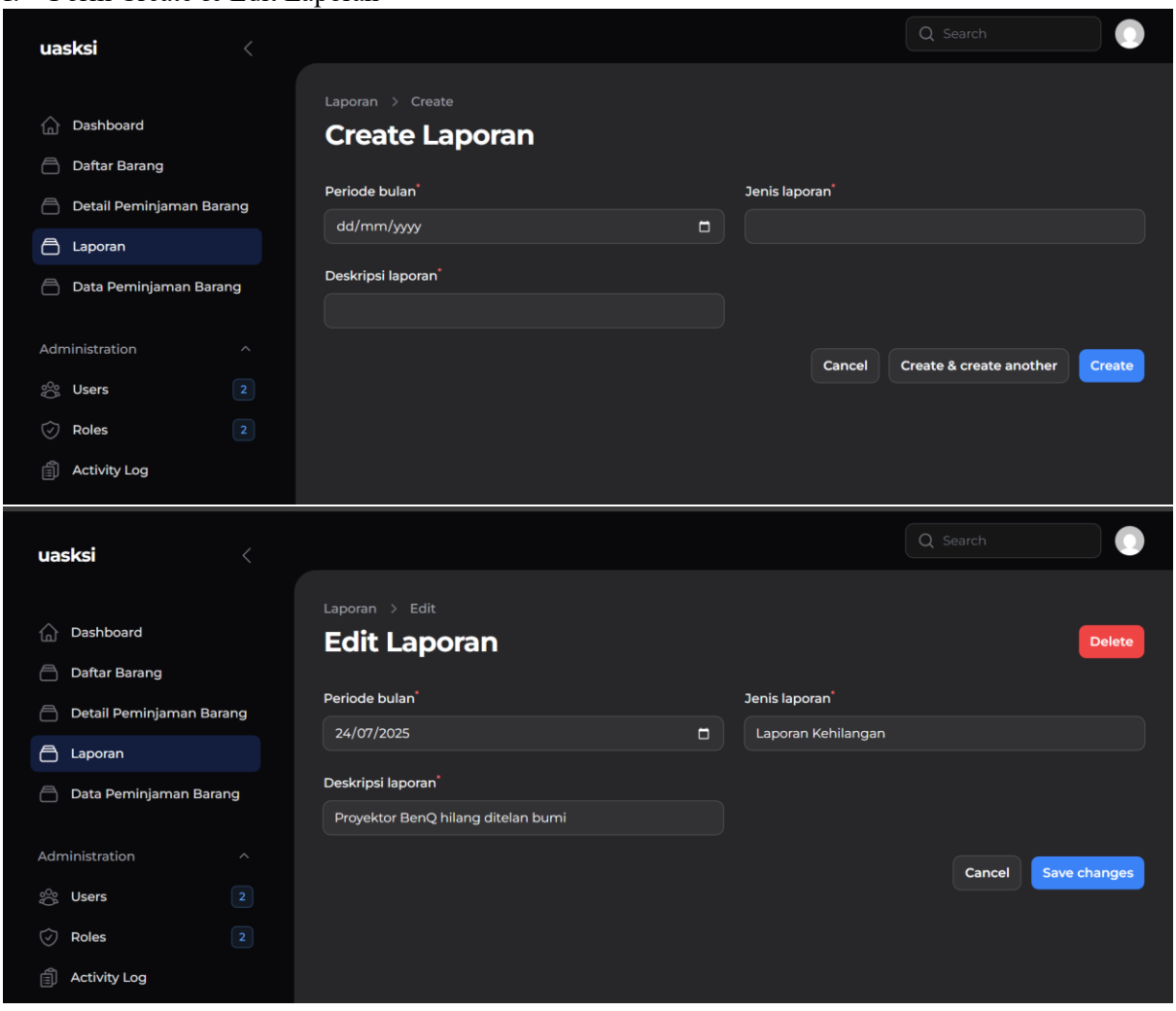


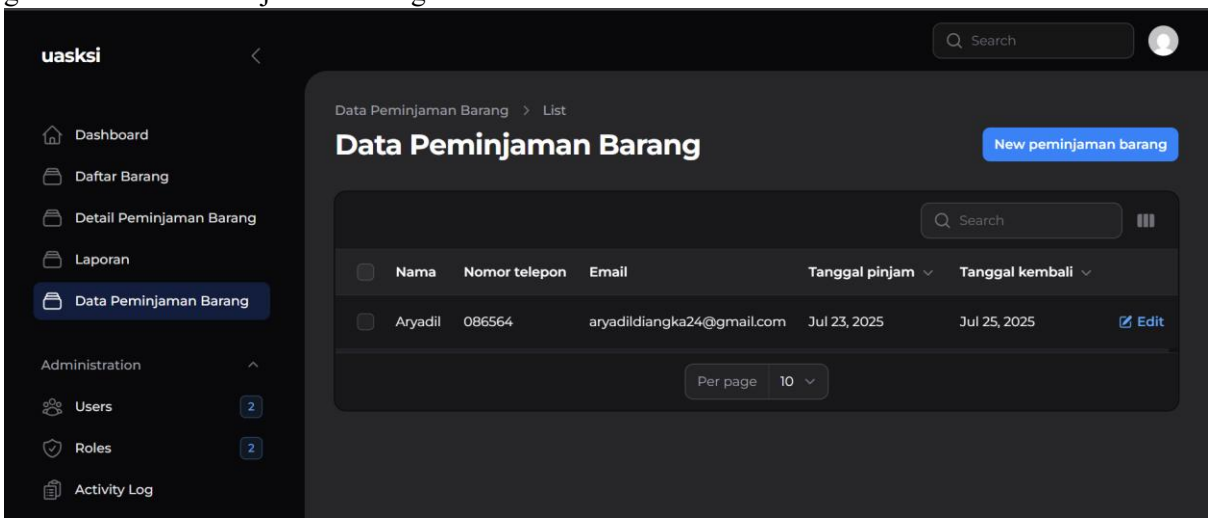
e. Tabel Laporan



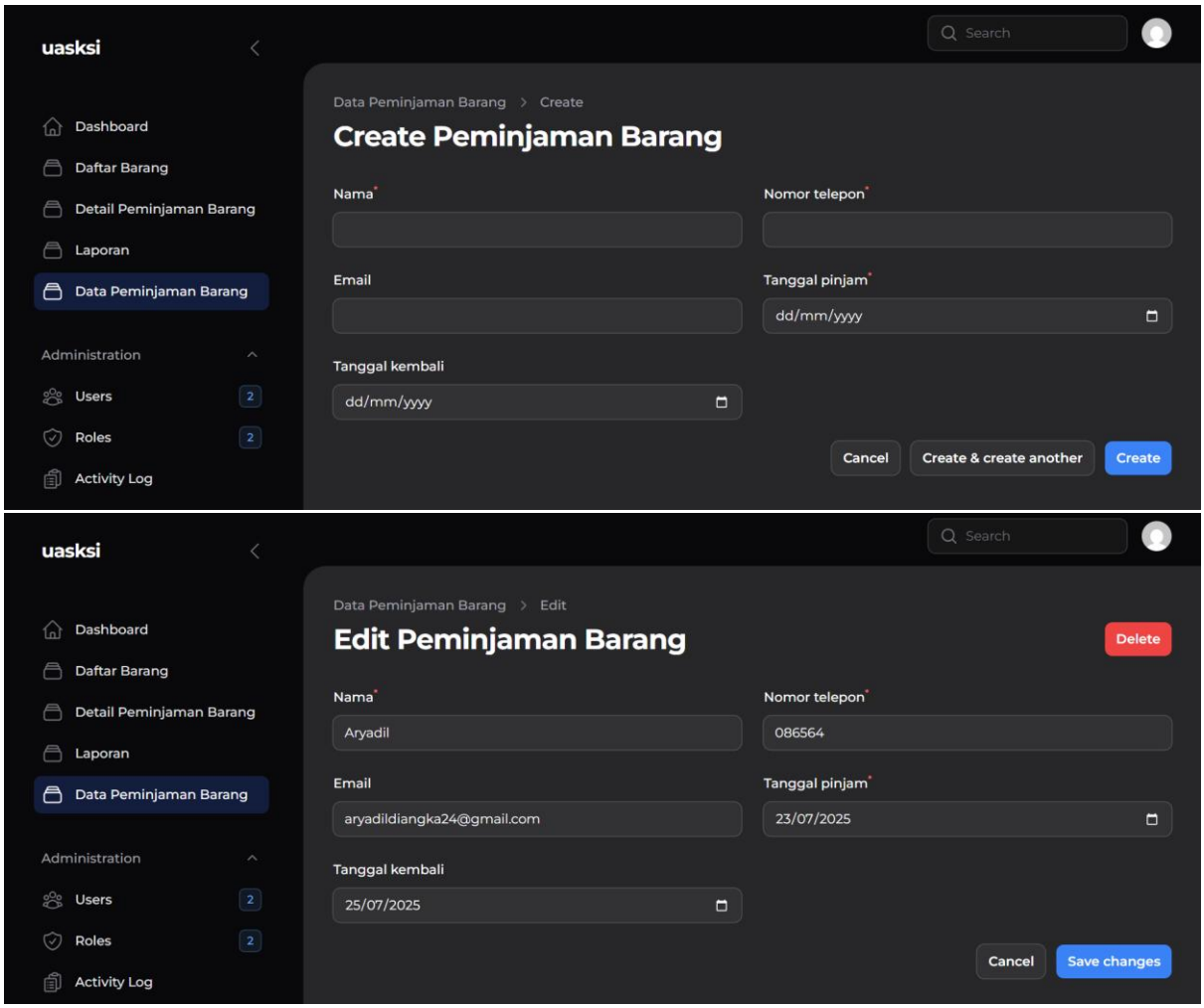

f. Form Create & Edit Laporan



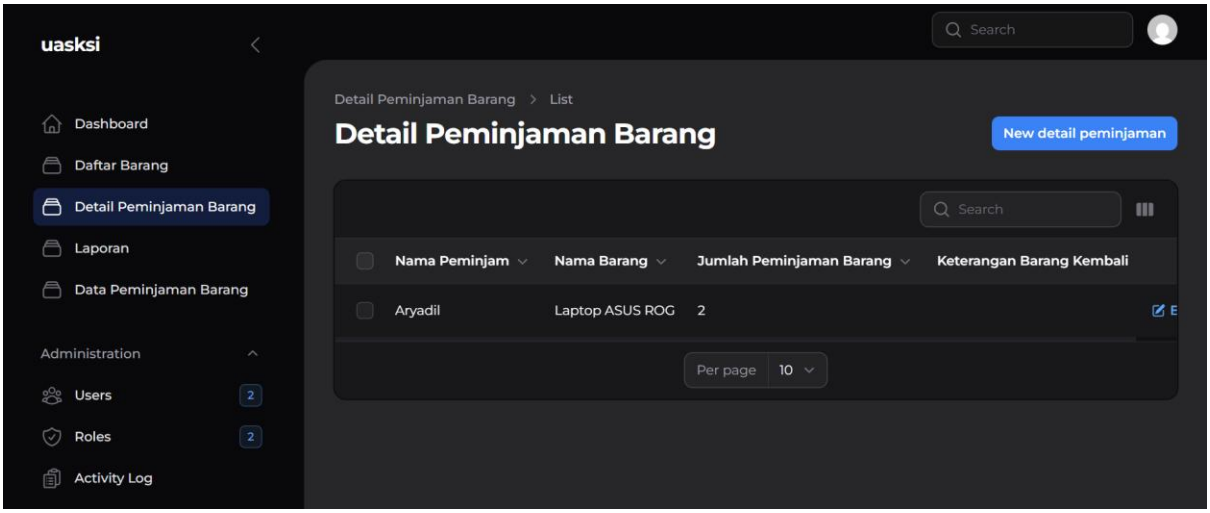
g. Tabel Data Peminjaman Barang



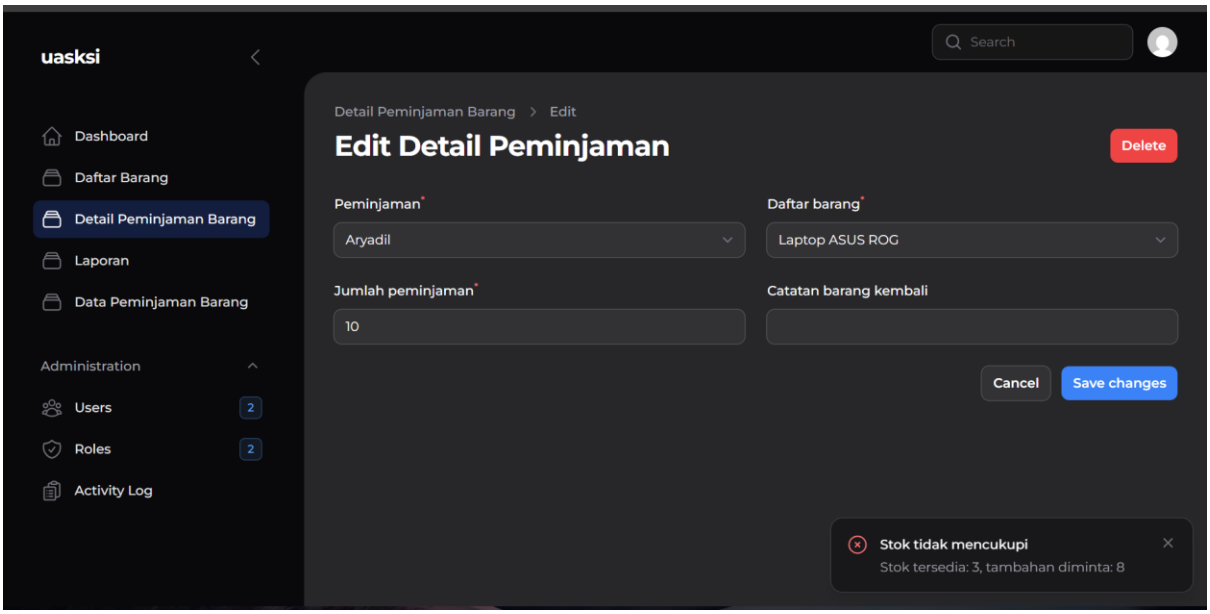

h. Form Create & Edit Peminjaman Barang

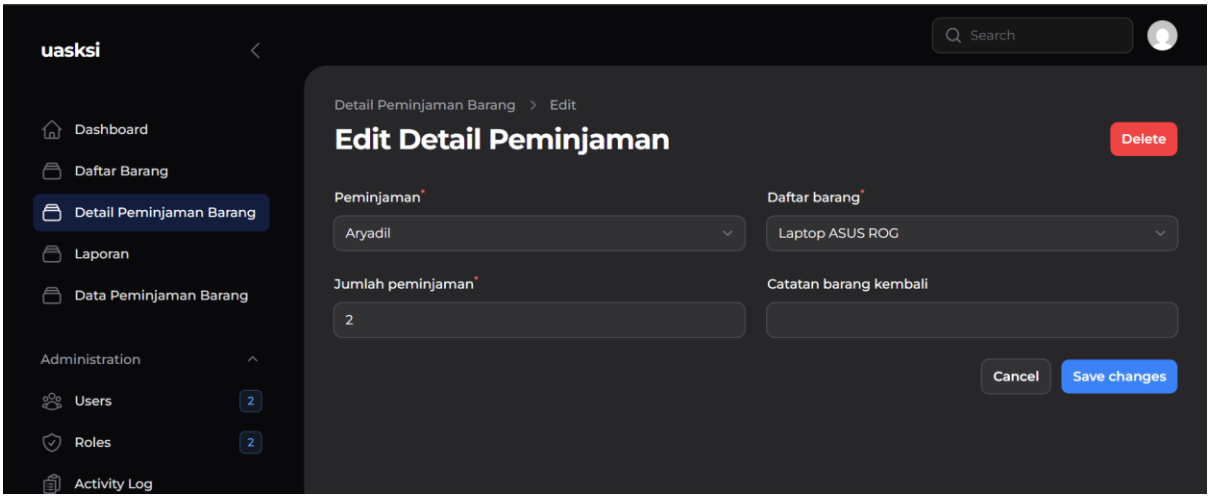



i. Tabel Detail Peminjaman Barang

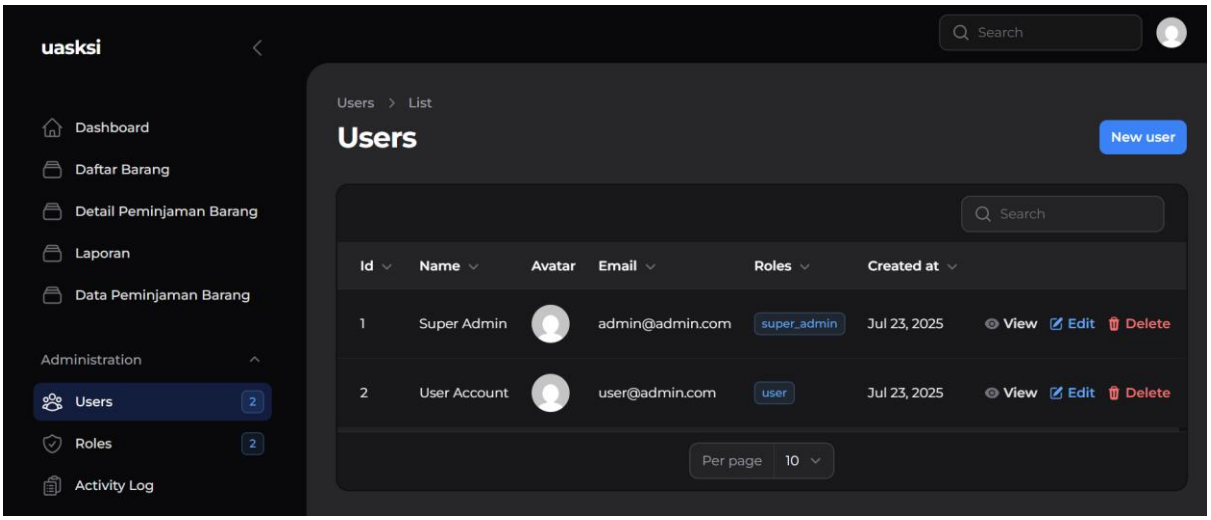


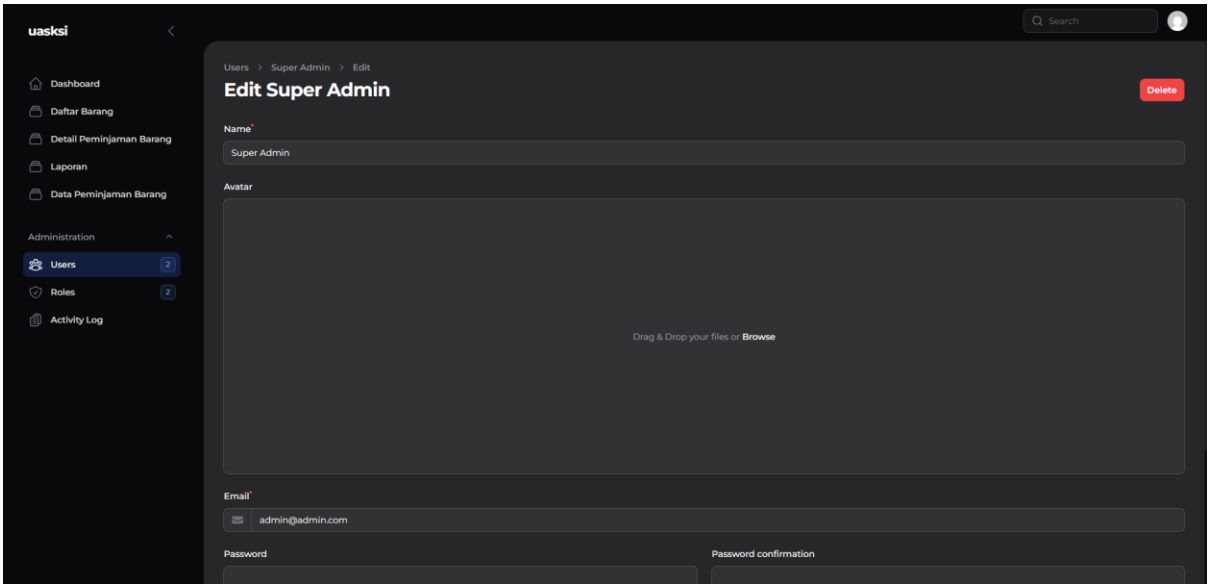

j. Form Create & Edit Detail Peminjaman Barang



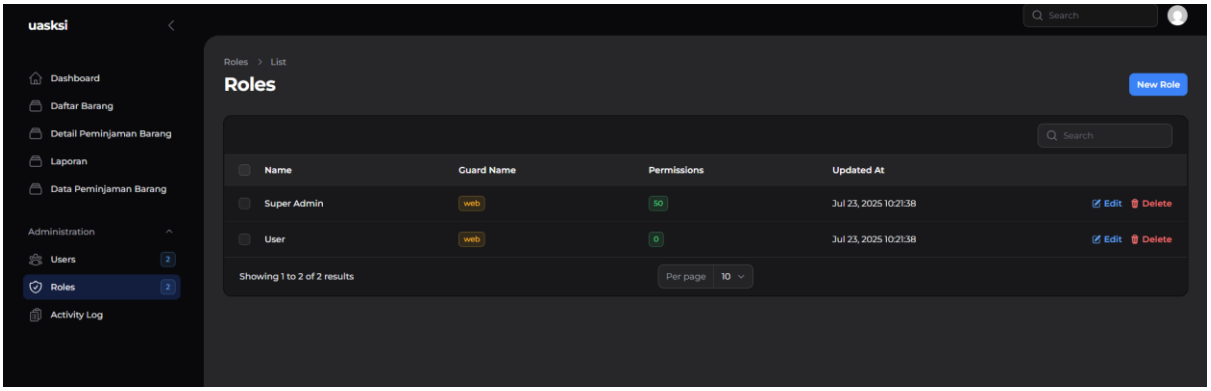


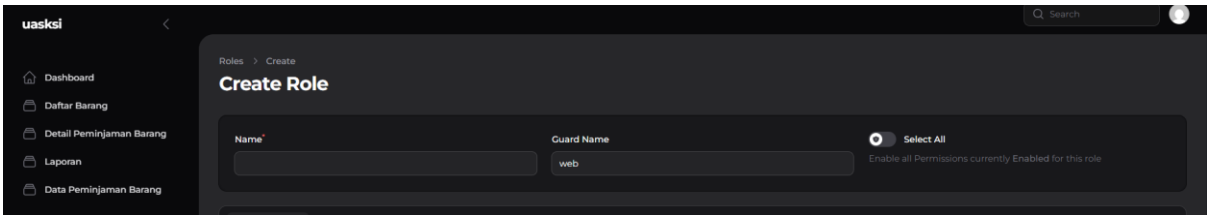

k. Tabel dan form edit Users



l. Form & tabel Role




3. Vullnerability Scan

Berikut adalah hasil vulnerability scan terhadap website melalui github ptt website scanner

```
+------------------------------------------------------------+
|                  Vulnerability Scan Report                 |
+------------------------------------------------------------+


[1] Missing security header: X-Content-Type-Options
        - Risk Level: 1 (Low)

        Vulnerability Details:
        - Evidence 1:
                - URL: https://politicians-rat-anyone-riders.trycloudflare.com/admin/login
                - Evidence: Response headers do not include the X-Content-Type-Options HTTP security header Request / Response


        - Description: We noticed that the target application's server responses lack the <code>X-Content-Type-Options</code> header. This header is particularly important for preventing Internet Explorer fro
m reinterpreting the content of a web page (MIME-sniffing) and thus overriding the value of the Content-Type header.
        - Recommendation: We recommend setting the X-Content-Type-Options header such as `X-Content-Type-Options: nosniff`.


[2] Missing security header: Content-Security-Policy
        - Risk Level: 1 (Low)

        Vulnerability Details:
        - Evidence 1:
                - URL: https://politicians-rat-anyone-riders.trycloudflare.com/admin/login
                - Evidence: Response does not include the HTTP Content-Security-Policy security header or meta tag Request / Response


        - Description: We noticed that the target application lacks the Content-Security-Policy (CSP) header in its HTTP responses. The CSP header is a security measure that instructs web browsers to enforce
specific security rules, effectively preventing the exploitation of Cross-Site Scripting (XSS) vulnerabilities.
        - Recommendation: Configure the Content-Security-Header to be sent with each HTTP response in order to apply the specific policies needed by the application.


[3] Missing security header: Strict-Transport-Security
        - Risk Level: 1 (Low)

        Vulnerability Details:
        - Evidence 1:
                - URL: https://politicians-rat-anyone-riders.trycloudflare.com/admin/login
                - Evidence: Response headers do not include the HTTP Strict-Transport-Security header Request / Response


        - Description: We noticed that the target application lacks the HTTP Strict-Transport-Security header in its responses. This security header is crucial as it instructs browsers to only establish secur
e (HTTPS) connections with the web server and reject any HTTP connections.
        - Recommendation: The Strict-Transport-Security HTTP header should be sent with each HTTPS response. The syntax is as follows: `Strict-Transport-Security: max-age=&lt;seconds>[; includeSubDomains]` Th
e parameter `max-age` gives the time frame for requirement of HTTPS in seconds and should be chosen quite high, e.g. several months. A value below 7776000 is considered as too low by this scanner check. The f
lag `includeSubDomains` defines that the policy applies also for sub domains of the sender of the response.

[4] Missing security header: Referrer-Policy
        - Risk Level: 1 (Low)

        Vulnerability Details:
        - Evidence 1:
                - URL: https://politicians-rat-anyone-riders.trycloudflare.com/admin/login
                - Evidence: Response headers do not include the Referrer-Policy HTTP security header as well as the <meta> tag with name 'referrer' is not present in the response. Request / Response


        - Description: We noticed that the target application's server responses lack the <code>Referrer-Policy</code> HTTP header, which controls how much referrer information the browser will send with each
 request originated from the current web application.
        - Recommendation: The Referrer-Policy header should be configured on the server side to avoid user tracking and inadvertent information leakage. The value `no-referrer` of this header instructs the br
owser to omit the Referer header entirely.


[5] Server software and technology found
        - Risk Level: 1 (Low)

        Vulnerability Details:
        - Evidence 1:
                - Software / Version: Alpine.js 3.14.9
                - Category: JavaScript frameworks

        - Evidence 2:
                - Software / Version: Bunny
                - Category: CDN

        - Evidence 3:
                - Software / Version: Filamentphp
                - Category: Development

        - Evidence 4:
                - Software / Version: Bunny Fonts
                - Category: Font scripts

        - Evidence 5:
                - Software / Version: Livewire
                - Category: Web frameworks, Miscellaneous

        - Evidence 6:
                - Software / Version: Laravel
                - Category: Web frameworks

        - Evidence 7:
                - Software / Version: PHP
                - Category: Programming languages

        - Evidence 8:
                - Software / Version: Cloudflare
                - Category: CDN


        - Description: We noticed that server software and technology details are exposed, potentially aiding attackers in tailoring specific exploits against identified systems and versions.
        - Recommendation: We recommend you to eliminate the information which permits the identification of software platform, technology, server and operating system: HTTP server headers, HTML meta informati
on, etc.

[6] Robots.txt file found
        - Risk Level: 1 (Low)

        Vulnerability Details:
        - Evidence 1:
                - URL: https://politicians-rat-anyone-riders.trycloudflare.com/robots.txt


        - Description: We found the robots.txt on the target server. This file instructs web crawlers what URLs and endpoints of the web application they can visit and crawl. Website administrators often misu
se this file while attempting to hide some web pages from the users.
        - Recommendation: We recommend you to manually review the entries from robots.txt and remove the ones which lead to sensitive locations in the website (ex. administration panels, configuration files,
etc).


[7] Login Interface Found
        - Risk Level: 0 (Info)

        Vulnerability Details:
        - Evidence 1:
                - URL: https://politicians-rat-anyone-riders.trycloudflare.com/admin/login
                - Evidence: `<input autocomplete="on" autofocus="autofocus" class="fi-input block w-full border-none py-1.5 text-base text-gray-950 transition duration-75 placeholder:text-gray-400 focus:ring-
0 disabled:text-gray-500 disabled:[-webkit-text-fill-color:theme(colors.gray.500)] disabled:placeholder:[-webkit-text-fill-color:theme(colors.gray.400)] dark:text-white dark:placeholder:text-gray-500 dark:dis
abled:text-gray-400 dark:disabled:[-webkit-text-fill-color:theme(colors.gray.400)] dark:disabled:placeholder:[-webkit-text-fill-color:theme(colors.gray.500)] sm:text-sm sm:leading-6 bg-white/0 ps-3 pe-3" id="
data.email" required="required" tabindex="1" type="email" wire:model="data.email"/> <input autocomplete="current-password" class="fi-input block w-full border-none py-1.5 text-base text-gray-950 transition du
ration-75 placeholder:text-gray-400 focus:ring-0 disabled:text-gray-500 disabled:[-webkit-text-fill-color:theme(colors.gray.500)] disabled:placeholder:[-webkit-text-fill-color:theme(colors.gray.400)] dark:tex
t-white dark:placeholder:text-gray-500 dark:disabled:text-gray-400 dark:disabled:[-webkit-text-fill-color:theme(colors.gray.400)] dark:disabled:placeholder:[-webkit-text-fill-color:theme(colors.gray.500)] sm:
text-sm sm:leading-6 bg-white/0 ps-3 pe-3 [&::-ms-reveal]:hidden" id="data.password" required="required" tabindex="2" wire:model="data.password" x-bind:type="isPasswordRevealed ? 'text' : 'password'"/> <butto
n class="fi-btn relative grid-flow-col items-center justify-center font-semibold outline-none transition duration-75 focus-visible:ring-2 rounded-lg fi-color-cust... (truncated)` Request / Response


        - Description: We have discovered that the target application presents a login interface that could be a potential target for attacks. While login interfaces are standard for user authentication, they
 can become vulnerabilities if not properly secured.
        - Recommendation: Ensure each interface is not bypassable using common knowledge of the application or leaked credentials using occasional password audits.


[8] Security.txt file is missing
        - Risk Level: 0 (Info)

        Vulnerability Details:
        - Evidence 1:
                - URL: Missing: https://politicians-rat-anyone-riders.trycloudflare.com/.well-known/security.txt


        - Description: We have noticed that the server is missing the security.txt file, which is considered a good practice for web security. It provides a standardized way for security researchers and the p
ublic to report security vulnerabilities or concerns by outlining the preferred method of contact and reporting procedures.
        - Recommendation: We recommend you to implement the security.txt file according to the standard, in order to allow researchers or users report any security issues they find, improving the defensive me
chanisms of your server.
```

Hasil pemindaian keamanan menunjukkan bahwa beberapa header penting belum dikonfigurasi pada aplikasi web.

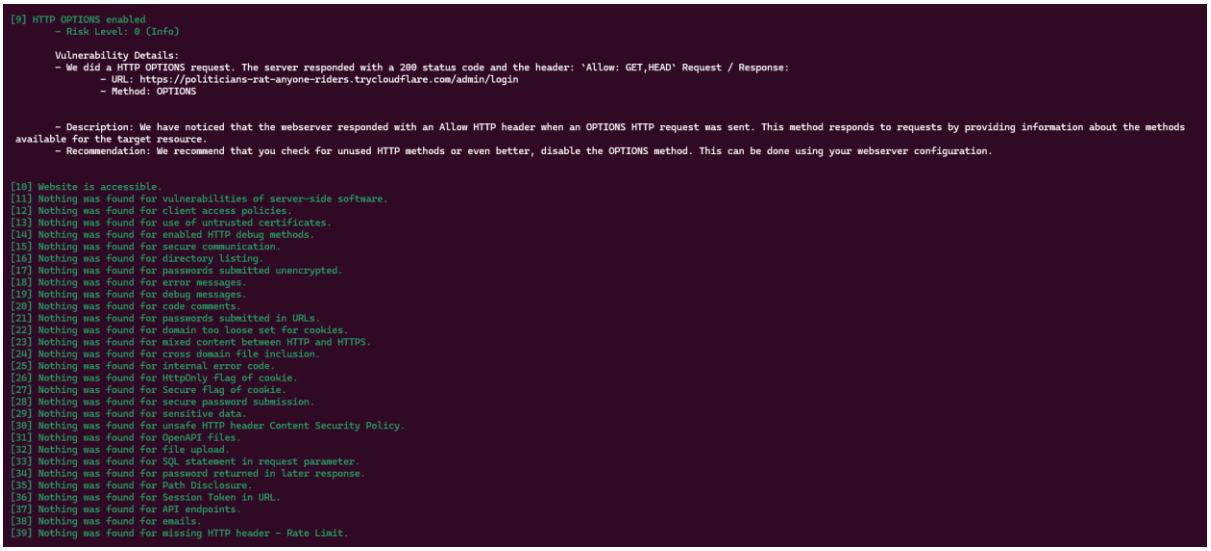
1. Header seperti Strict-Transport-Security, Content-Security-Policy, X-Content-Type-Options, dan Referrer-Policy tidak ditemukan dalam respons server. Ketiadaan header-header ini dapat membuka peluang terhadap risiko serangan seperti XSS (Cross-Site Scripting), sniffing MIME type, atau kebocoran informasi referer.

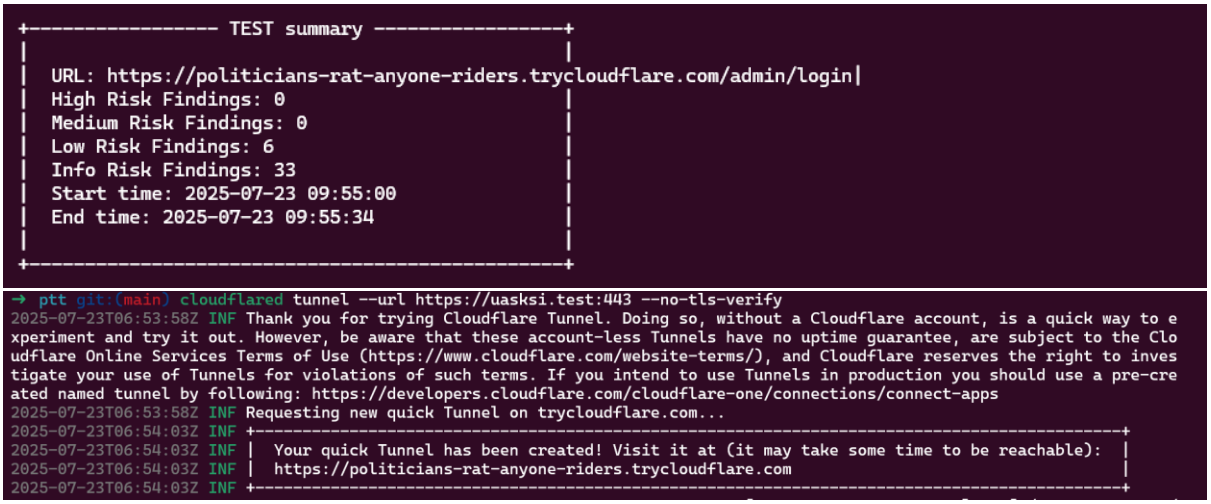
2. Selain itu, informasi mengenai perangkat lunak dan teknologi (server software and technology) yang digunakan oleh server masih terlihat secara publik, termasuk framework dan bahasa pemrograman. Hal ini dapat dimanfaatkan oleh pihak tidak bertanggung jawab untuk mengidentifikasi potensi celah keamanan yang spesifik.

3. File robots.txt juga ditemukan, dapat berisiko jika digunakan untuk menyembunyikan direktori atau halaman sensitif.

4. Antarmuka login terdeteksi sebagai bagian dari sistem dan dapat menjadi target eksploitasi jika tidak dilindungi dengan benar.

5. file security.txt yang tidak ditemukan, file security,txt merupakan standar yang memungkinkan peneliti keamanan melaporkan kerentanan dengan cara yang tepat. Terakhir, metode HTTP OPTIONS masih diaktifkan, yang dapat memberikan informasi tambahan kepada penyerang potensial mengenai metode HTTP yang diizinkan oleh server.






Sementara itu, Hasil pemindaian menunjukkan bahwa secara umum aplikasi web telah memenuhi sebagian besar standar keamanan dasar. Tidak ditemukan kerentanan yang berkaitan dengan perangkat lunak sisi server, kebijakan akses klien, sertifikat yang tidak terpercaya, metode debug HTTP yang aktif, maupun komunikasi yang tidak aman. Selain itu, tidak ada informasi sensitif seperti password yang dikirim tanpa enkripsi, error message yang terbuka, komentar kode, atau data sensitif lainnya yang terdeteksi dalam respons.

4. Pengamanan data
   Melakukan pengamanan data melalui model untuk membantu mencegah serangan Mass Assignment, di mana penyerang bisa mencoba mengisi kolom yang seharusnya tidak bisa diisi. Kemudian Menggunakan Eloquent seperti DaftarBarang::create() atau update() secara tidak langsung membantu mencegah serangan SQL Injection, karena Eloquent menggunakan *prepared statements* secara default.

```php
class DaftarBarang extends Model
{
    protected $fillable=[
        "nama_barang",'kondisi_barang', 'kondisi', 'status','jumlah'
    ];

    public function DetailPeminjaman(){
        $this->hasMany(DetailPeminjaman::class, 'id_barang');
    }

}
```

```php
class DaftarBarangController extends Controller
{
    public function store(Request $request)
{
    $validated = $request->validate([
        'nama_barang' => 'required|string|max:255',
        'kondisi_barang' => 'required|in:baik,rusak,servis',
        'status' => 'required|in:aktif,dipinjam,hilang,rusak',
        'jumlah' => 'required|integer|min:0',
    ]);

    DaftarBarang::create($validated);

    return response()->json(['message' => 'Barang berhasil ditambahkan']);
}
}
```