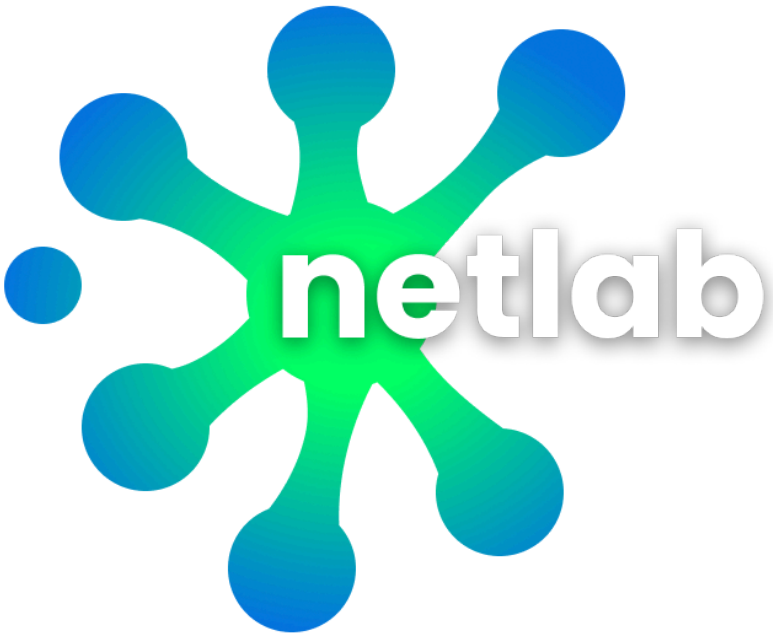
PROYEK AKHIR
KEAMANAN JARINGAN 2025



Ryan Adidaru 2306266994
Fathan Yazid Satriani 2306250560

1 Pendahuluan

Aplikasi web sederhana berbasis Node.js/Express yang menjadi objek Proyek Akhir ini menyediakan fitur autentikasi dasar dan pengelolaan file melalui endpoint upload. Tanpa kontrol keamanan yang memadai, fitur-fitur tersebut berpotensi dieksploitasi oleh pihak tidak berwenang. Laporan ini mendokumentasikan hasil pengujian keamanan yang dilakukan terhadap aplikasi tersebut, mencakup analisis risiko, pembuktian kerentanan, serta usulan mitigasi untuk menjaga keamanan layanan.

1.1 Tujuan

Mengidentifikasi dua kerentanan utama pada aplikasi, yaitu:
- Insecure File Upload
- Insecure Change Password (ganti password tanpa verifikasi password lama)

Menilai dampak terhadap kerahasiaan, integritas, dan keamanan data pengguna.
Menyusun rekomendasi perbaikan yang aplikatif untuk menutup kerentanan tersebut.

1.2 Ruang Lingkup

Ruang lingkup pengujian dalam proyek ini meliputi:
- Backend Node.js/Express yang dijalankan secara lokal.
- Fitur upload melalui endpoint /upload.
- Fitur autentikasi dan ganti password melalui endpoint /change-password.
- Pengujian hanya mencakup serangan berbasis logic flaw dan improper access control.
- Serangan DoS, social engineering, dan brute force tidak termasuk ruang lingkup.

1.3 Metodologi

Metodologi yang digunakan mengikuti pendekatan industri keamanan aplikasi web:

- Black-box testing dikombinasikan dengan review kode statis untuk memvalidasi logika otorisasi.
- Referensi utama berasal dari OWASP Web Security Testing Guide (WSTG).
- Pembuktian dilakukan melalui pengujian manual dan observasi perilaku server terhadap input berbahaya.

2 Ringkasan Eksekutif

Berdasarkan pengujian terhadap aplikasi, ditemukan dua kerentanan dengan risiko tinggi:

1. **Insecure File Upload**
   Endpoint upload memperbolehkan semua jenis file diunggah tanpa validasi. Hal ini memungkinkan penyerang mengunggah file berbahaya seperti .exe, .js, atau script lainnya yang dapat digunakan untuk eskalasi serangan.

2. **Insecure Change Password Feature**
   Fitur ganti password tidak mewajibkan pengguna memasukkan password lama. Akibatnya, siapa pun yang memiliki akses session (misal: melalui cookie yang dicuri) dapat mengganti password pemilik akun tanpa konfirmasi.

Kedua celah ini memungkinkan penyerang untuk menanam file berbahaya di server, melakukan kompromi akun pengguna, dan menciptakan risiko keamanan serius bagi sistem. Mitigasi yang diterapkan kemudian berhasil mencegah eksploitasi ulang.

3 Detail Kerentanan

3.1 Insecure File Upload

Deskripsi:

Endpoint **/upload** menerima file dari klien tanpa:

- Validasi ekstensi
- Validasi MIME type
- Pembatasan jenis file yang diperbolehkan
- Sanitasi terhadap file yang berpotensi berbahaya

Server menyimpan file apa pun yang dikirimkan, termasuk executable dan script.

Langkah Eksploitasi:

1. Login sebagai user.
2. Upload file dengan ekstensi .exe atau .js.
3. File diterima dan disimpan di server.

Dampak:

Penyerang dapat mengunggah file berbahaya yang berpotensi untuk eksploitasi seperti mengeksekusi kode di server, menjadi backdoor, menyimpan payload malware, dan menjadi pintu masuk untuk serangan lanjutan.

Proof Of Concept:

1. Upload file payload.js, akan diterima tanpa penolakan.
2. File muncul di direktori uploads/ meski bukan file gambar.

3.2 Insecure Change Password Feature

Deskripsi:

Fitur ganti password tidak meminta password lama sebagai syarat perubahan.
Sebagai contoh, endpoint hanya menerima:

```
{ "newPassword": "<password baru>" }
```

tanpa memvalidasi bahwa pengguna benar-benar mengetahui password sebelumnya.

Langkah Eksploitasi:
1. Login sebagai user.
2. Akses halaman ganti password.
3. Masukkan password baru tanpa verifikasi password lama.

Dampak:

- Akun dapat diambil alih permanen.
- Integritas kredensial pengguna hilang.
- Risiko tinggi jika session dicuri melalui XSS, shoulder surfing, atau session hijacking.

Proof Of Concept:

- Ganti password tanpa field password lama pada page /change-password.
- Password berhasil diubah tanpa verifikasi.

4 Mitigasi dan Remediasi

4.1 Mitigasi Insecure File Upload

Untuk menutup celah insecure file upload yang memungkinkan berkas berbahaya diunggah ke dalam sistem, beberapa langkah mitigasi diterapkan pada sisi backend:

- Server kini menerapkan validasi tipe berkas secara ketat dengan memeriksa MIME type dan ekstensi file, sehingga hanya berkas gambar seperti .jpg, .jpeg, dan .png yang dapat diterima.
- Validasi dilakukan sepenuhnya di sisi server untuk mencegah manipulasi header pada request. Selain itu, server juga melakukan sanitasi nama file untuk menghindari karakter atau pola yang dapat digunakan untuk melakukan path traversal ataupun penyisipan perintah berbahaya.
- Selanjutnya, direktori penyimpanan file dibatasi pada folder non-eksekusi sehingga berkas yang berhasil diunggah tidak dapat dijalankan sebagai kode, mengurangi risiko remote code execution.
- Pembatasan ukuran file juga diterapkan untuk mencegah penyalahgunaan kapasitas penyimpanan atau upaya denial of service melalui unggahan file berukuran besar. Implementasi ini memastikan bahwa fitur unggah file hanya digunakan sesuai tujuan awalnya dan tidak dapat dimanfaatkan untuk menanamkan payload berbahaya ke dalam sistem.

4.2 Mitigasi Change Password Feature

Untuk mencegah penyalahgunaan fitur ganti kata sandi yang sebelumnya tidak mewajibkan verifikasi kata sandi lama, server kini mewajibkan pengguna memasukkan current password sebelum melakukan perubahan. Validasi dilakukan dengan membaca hash kata sandi dari basis data dan membandingkannya dengan input pengguna, sehingga perubahan hanya dapat dilakukan oleh pemilik akun yang memiliki kredensial yang sah. Pendekatan ini menghilangkan risiko pengambilalihan akun yang terjadi apabila penyerang memperoleh token sesi tanpa mengetahui kata sandi utama.

Selain itu, pembatasan hak akses juga diterapkan pada proses penggantian kata sandi. Pengguna non-admin hanya diperbolehkan mengganti kata sandi miliknya sendiri, terlepas dari parameter atau user_id apa pun yang dikirimkan dari sisi klien. Server sepenuhnya mengabaikan parameter yang dapat dimanipulasi oleh klien dan hanya menggunakan identitas pengguna berdasarkan token autentikasi yang telah diverifikasi. Dengan penerapan mitigasi tersebut, fitur perubahan kata sandi kini lebih aman dan tidak lagi menjadi titik masuk yang mudah untuk eskalasi serangan ataupun pengambilalihan akun secara tidak sah.

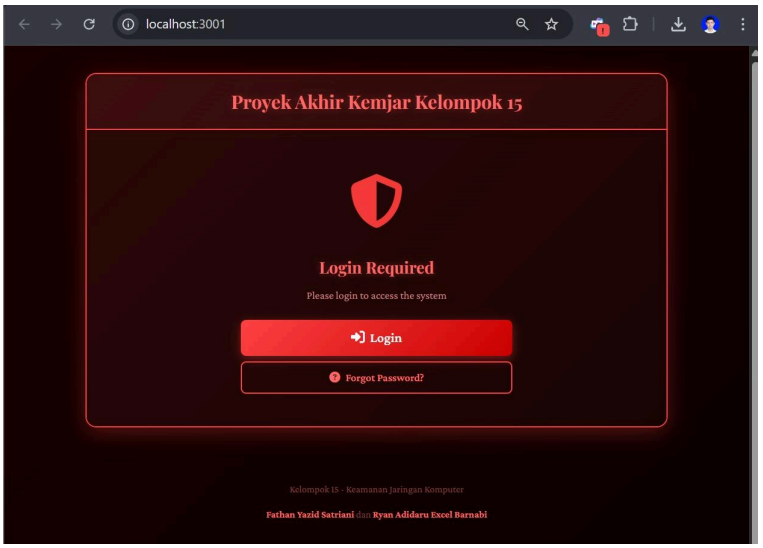
Kesimpulan

Aplikasi web yang diuji terbukti memiliki dua celah keamanan kritis, insecure file upload dan insecure change password. Kedua *vulnerability* tersebut memungkinkan penyerang mengunggah file berbahaya serta mengambil alih akun tanpa perlu mengetahui kata sandi lama. Melalui pengujian berbasis black-box dan review logika backend, risiko terhadap kerahasiaan, integritas, dan kontrol akses pengguna berhasil dipetakan dengan jelas. Setelah diterapkan mitigasi seperti validasi tipe file, sanitasi nama berkas, pembatasan direktori penyimpanan, serta verifikasi kata sandi lama pada proses perubahan password, aplikasi kini memiliki perlindungan yang jauh lebih kuat dan tidak lagi rentan dieksploitasi melalui kedua celah tersebut.

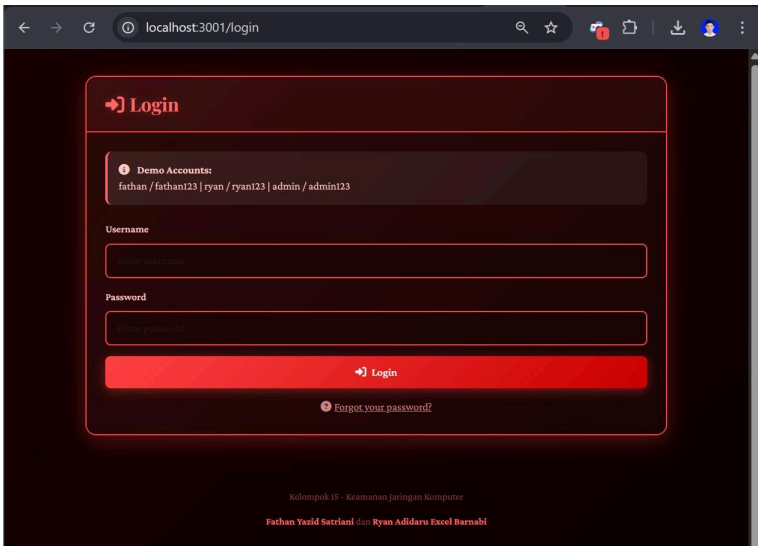
6 Lampiran

Walkthrough vulnerable website

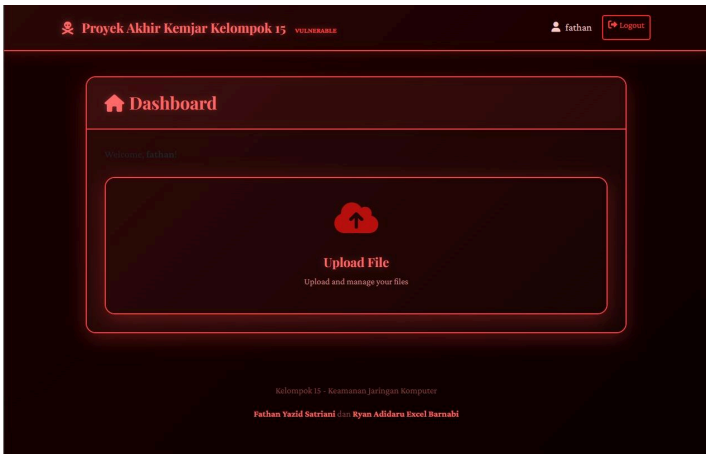
6.1. Landing Page

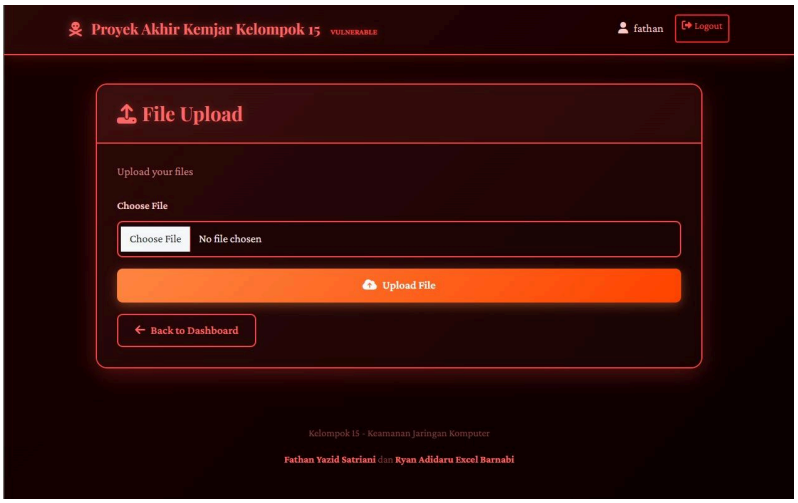


6.2. Login Page

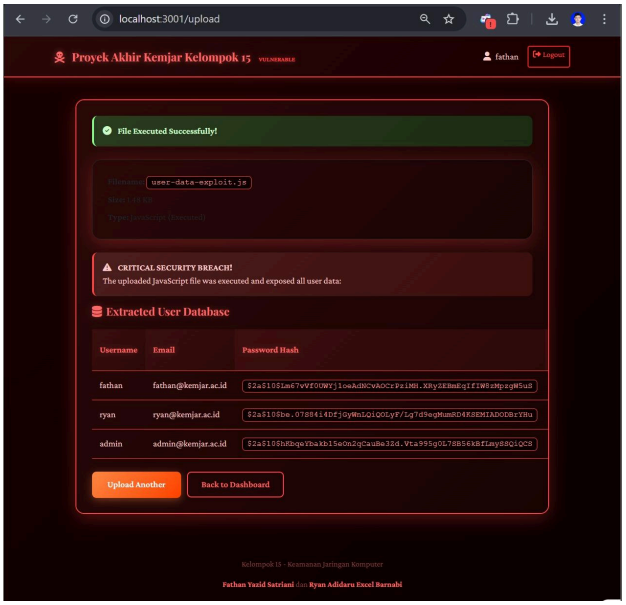


6.3. File Upload Page

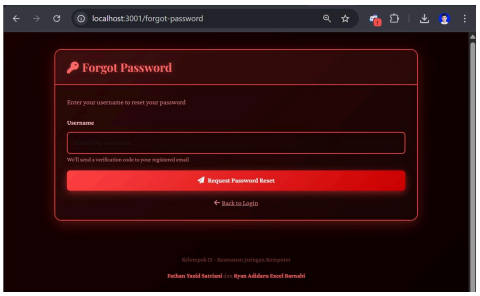


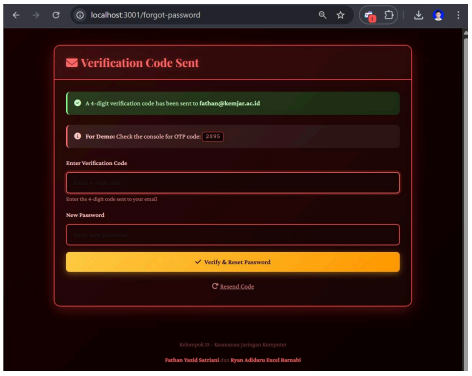


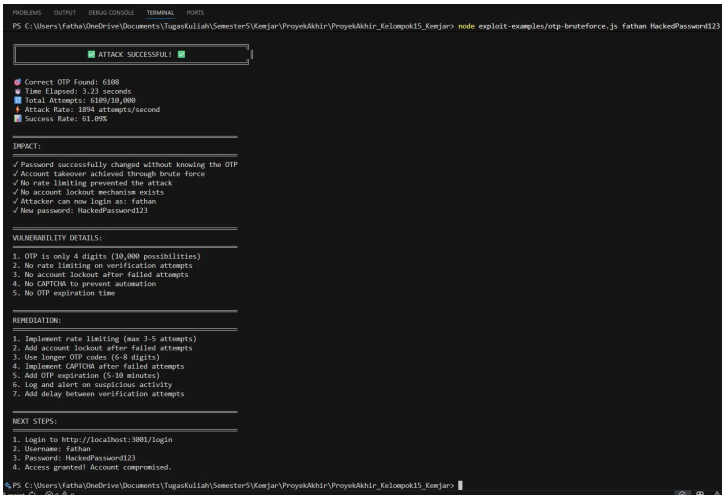
6.4. File Execute Through Upload Page



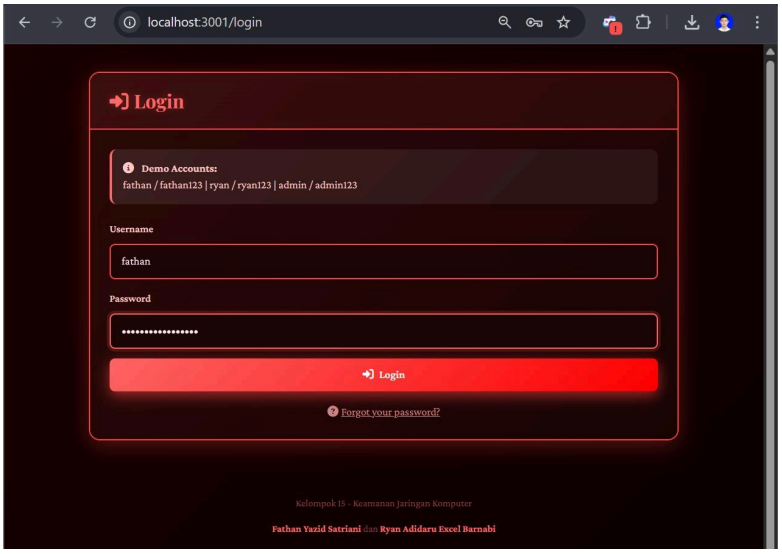
6.5. Forgot Password Page and Exploitation with otp-bruteforce.js

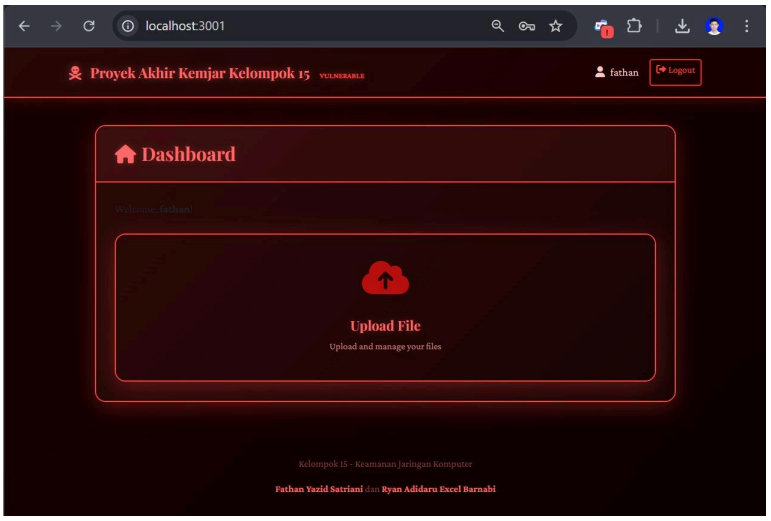




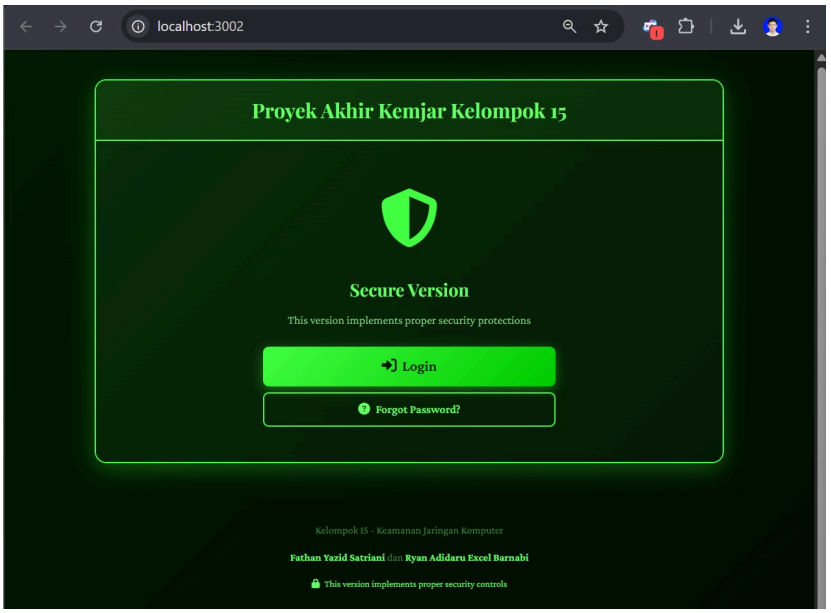


6.6. Exploitation Successfully Attempted

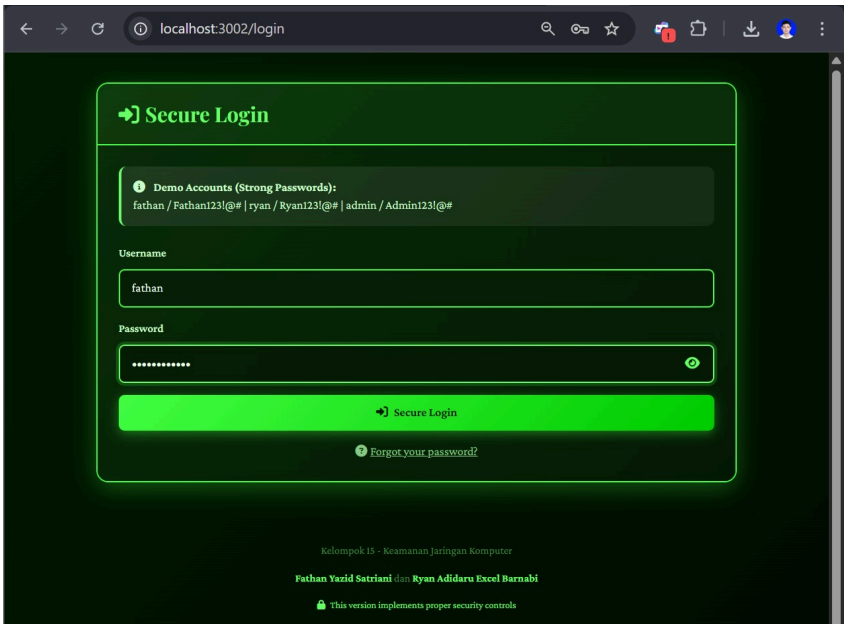




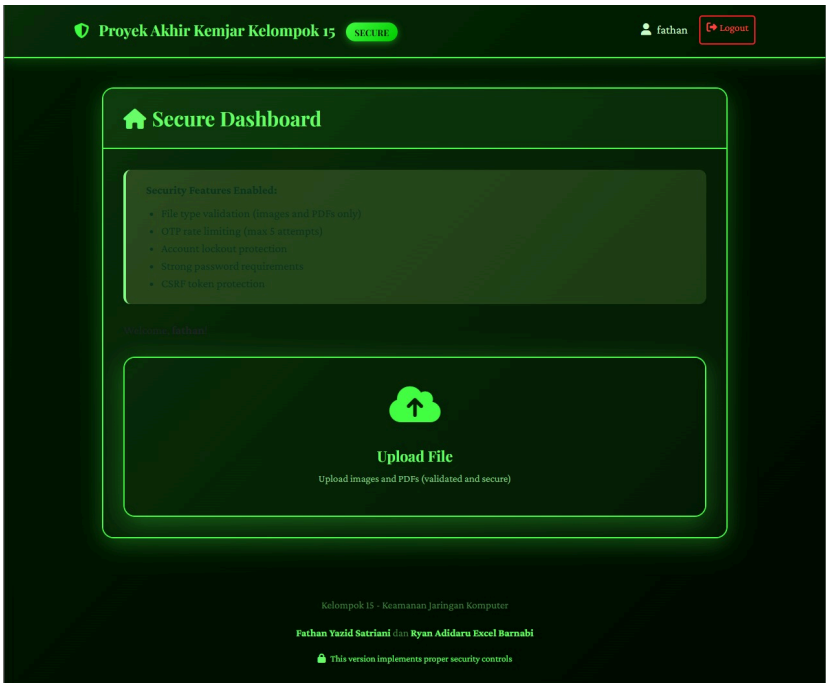
6.7 Secured Landing Page

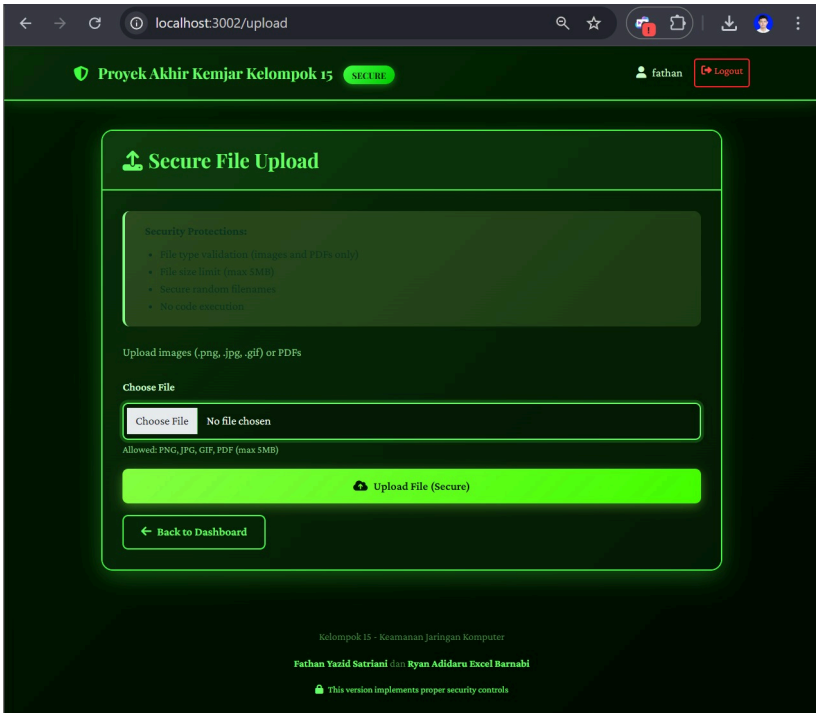


6.8 Secured Login Page

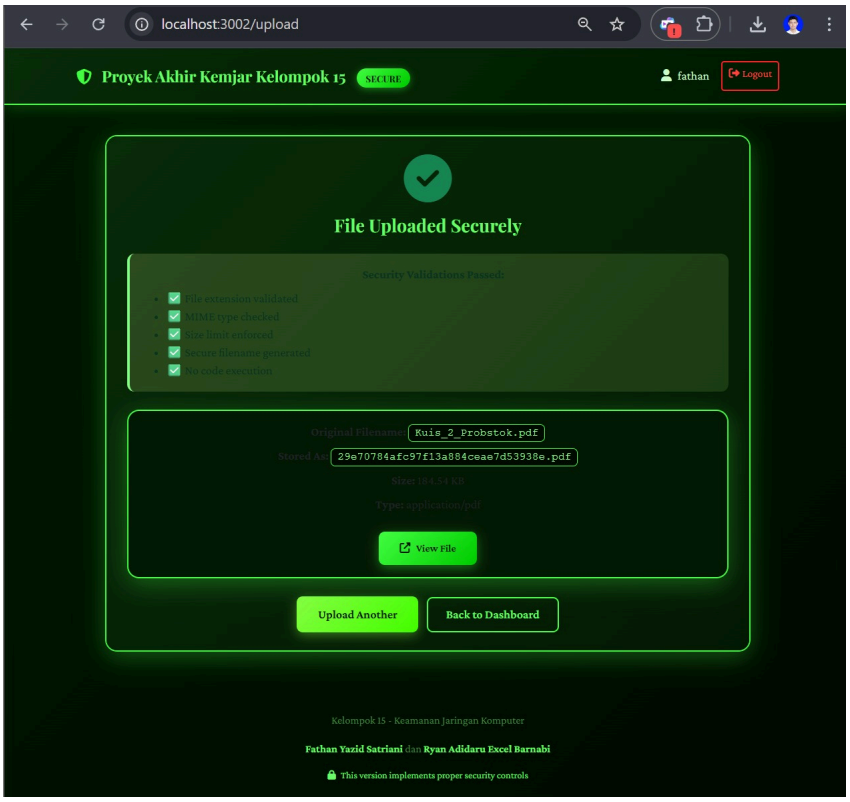


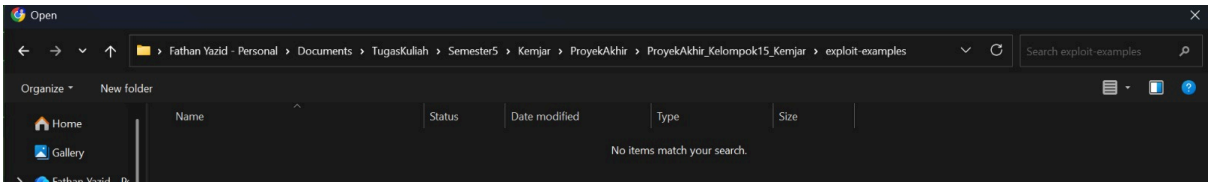
6.9 Secured Dashboard and File Upload

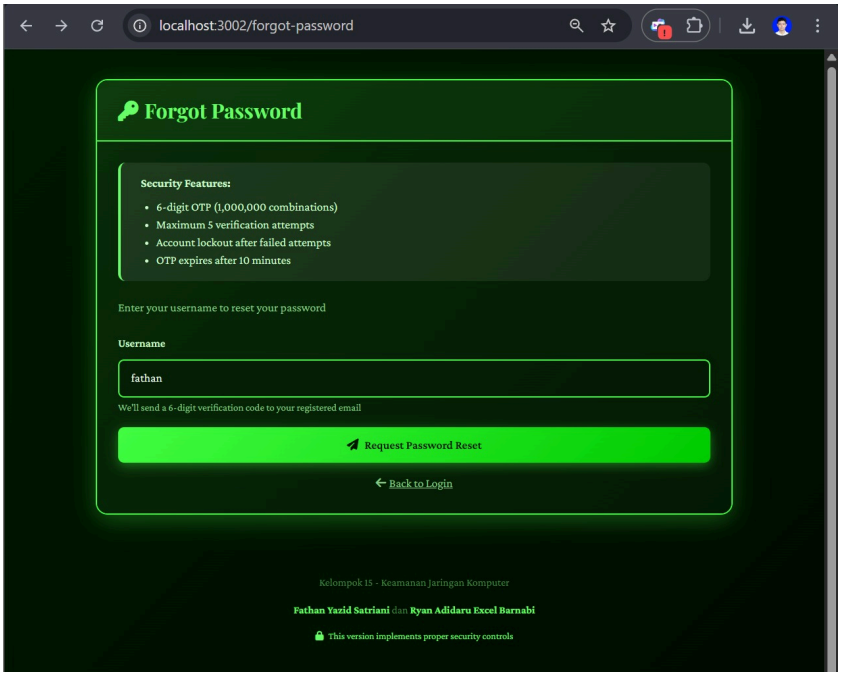


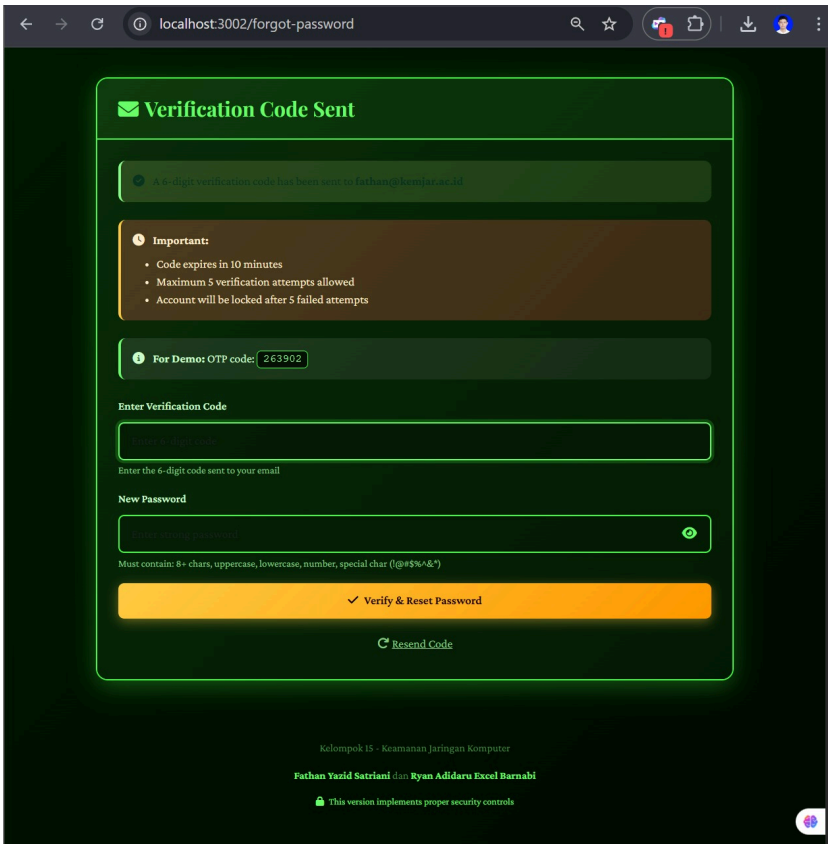


6.10 Proof of Secured Mechanisms (Secure File Upload and OTP Rate Limit Mechanism)

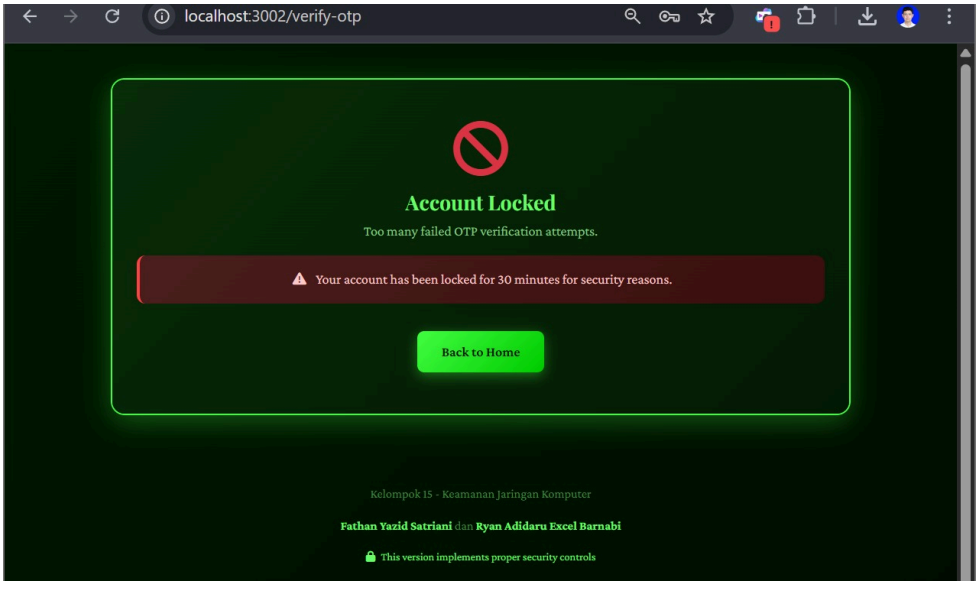









```
PS C:\Users\fatha\OneDrive\Documents\TugasKuliah\Semester5\Kemjar\ProyekAkhir\ProyekAkhir_Kelompok15_Kemjar>
PS C:\Users\fatha\OneDrive\Documents\TugasKuliah\Semester5\Kemjar\ProyekAkhir\ProyekAkhir_Kelompok15_Kemjar> curl http://localhost:3002/api/users
curl : Cannot GET /api/users
At line:1 char:1
+ curl http://localhost:3002/api/users
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : InvalidOperation: (System.Net.HttpWebRequest:HttpWebRequest) [Invoke-WebRequest], WebException
    + FullyQualifiedErrorId : WebCmdletWebResponseException,Microsoft.PowerShell.Commands.InvokeWebRequestCommand
PS C:\Users\fatha\OneDrive\Documents\TugasKuliah\Semester5\Kemjar\ProyekAkhir\ProyekAkhir_Kelompok15_Kemjar>
```



6.11 Link Video
https://drive.google.com/drive/folders/12YplhYWc_Lk6BYltYMYrwENcAkFTk1tp?usp=sharing

6.12 Link Github
https://github.com/coolcmyk/ProyekAkhir_Kelompok15_Kemjar