Jay's Bank
Security Assessment Findings Report

Date: Saturday 1th, 2024 Project: DC-001
Version 1.0

Confidentiality Statement

We are committed to maintaining the confidentiality of all information we access and discover during the penetration testing process of Jay's Bank application. This includes sensitive data, user information, and vulnerability findings. We will not disclose any information to third parties without permission, unless required for security improvement purposes. We will act professionally and ethically, respecting the privacy of all parties involved in this process.

Disclaimer

Penetration testing is conducted with the aim of improving the security of the Jay's Bank application. However, the results of this testing may not encompass all potential vulnerabilities and do not guarantee complete security. We are not liable for any damages or losses that may arise from the use or interpretation of information in this report. The use of information in the report should be exercised with consideration and actions in accordance with applicable security standards

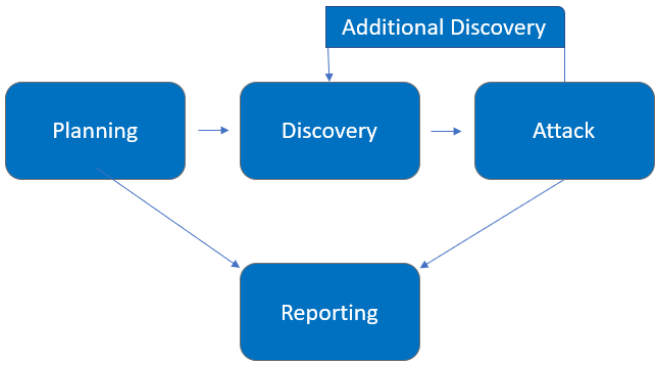
Contact Information

Name	Title	Contact Information
Stephanie	Mahasiswa IT 22	Email: heath@tcm-sec.com

Assessment Overview

The purpose of this assessment is to conduct a comprehensive penetration test on the Jay's Bank application to identify and mitigate potential security vulnerabilities. The assessment will cover all aspects of the application, including web interface, API, user account mechanisms, and data handling processes.

- Planning – Customer objectives are gathered, and rules of engagement are obtained.
- Discovery – Perform scanning and enumeration to identify potential vulnerabilities, weak areas, and exploits.
- Attack – Confirm potential vulnerabilities through exploitation and perform additional discovery upon new access.
- Reporting – Document all vulnerabilities and exploits discovered, failed attempts, and the company's strengths and weaknesses.



Assessment Components

Identification of common web application vulnerabilities such as SQL injection, Cross-Site Scripting (XSS), and authentication/authorization issues.

Finding Severity Ratings

The following table defines levels of severity and corresponding CVSS score range that are used throughout the document to assess vulnerability and risk impact.

Severity	CVSS V3 Score Range	Definition
Critical	9.0-10.0	Temuan kritis menggambarkan kerentanan yang memiliki potensi dampak serius terhadap kerahasiaan, integritas, atau ketersediaan sistem. Eksploitasi kerentanan ini dapat mengakibatkan akses penuh ke sistem, kerugian data kritis, atau gangguan layanan yang signifikan.
High	7.0-8.9	Temuan tinggi menunjukkan kerentanan yang memiliki potensi dampak yang signifikan terhadap keamanan sistem. Meskipun tidak seberat temuan kritis, eksploitasi kerentanan ini masih dapat menghasilkan akses yang tidak sah atau pencurian data sensitif.
Moderate	4.0-6.9	Temuan menengah mencakup kerentanan yang memiliki potensi dampak moderat terhadap keamanan sistem. Meskipun mungkin tidak langsung mengancam kerahasiaan atau integritas data, eksploitasi kerentanan ini masih dapat menyebabkan gangguan operasional atau akses tidak sah ke sistem.
Low	0.1-3.9	Temuan rendah menunjukkan kerentanan yang memiliki dampak minimal terhadap keamanan sistem. Eksploitasi kerentanan ini mungkin memerlukan kondisi khusus atau akses yang terbatas, dan dampaknya terhadap operasional sistem terbatas.
Informational	N/A	Saran atau rekomendasi untuk peningkatan keamanan, tetapi tidak memerlukan tindakan perbaikan segera.

Risk Factors

Risk is measured by two factors: Likelihood and Impact:

Likelihood

Kemungkinan mengukur potensi kerentanan yang dieksploitasi. Peringkat diberikan berdasarkan tingkat kesulitan serangan, alat yang tersedia, tingkat keterampilan penyerang, dan lingkungan klien.

Impact

Dampak mengukur dampak potensi kerentanan terhadap operasi, termasuk kerahasiaan, integritas, dan ketersediaan sistem dan/atau data klien, kerusakan reputasi, dan kerugian finansial.

Scope

Assessment	Details
Internal Penetration Test	167.172.75.216

Scope Exclusions

Per client request, TCMS did not perform any of the following attacks during testing:

- Denial of Service (DoS)
- Phishing/Social Engineering

All other attacks not specified above were permitted by Demo Corp.

Client Allowances

Executive Summary

We evaluated Jay's Bank internal security posture through penetration testing from May 28nd, 2024 to June 1st, 2024. The following sections provide a high-level overview of vulnerabilities discovered, successful and unsuccessful attempts, and strengths and weaknesses.

Scoping and Time Limitations

Scoping during the engagement did not permit denial of service or social engineering across all testing components.

Time limitations were in place for testing. Internal network penetration testing was permitted for 4 business days.

Testing Summary

The primary objective of this penetration test was to evaluate the security posture of Jay's Bank application by identifying potential vulnerabilities and providing recommendations for mitigation before its public release. The scope of the assessment included the application's web interface, API interactions, user account mechanisms and authentication, and database interactions and data handling processes, specifically targeting the IP address 167.172.75.216.

Using non-destructive testing techniques, both automated tools and manual methods were employed to ensure a thorough examination without harming the application or data. Key findings revealed multiple vulnerabilities, including SQL injection, where several input fields allowed potential unauthorized access to database information. Cross-Site Scripting (XSS) vulnerabilities were detected in various web forms, posing risks for user data theft and session hijacking. Authentication issues were discovered, highlighting weaknesses that could enable login bypass, and authorization flaws were found in user role management, which could allow unauthorized access to restricted functions. Additionally, several insecure API endpoints were identified, exposing sensitive data to potential exploitation.

Tester Notes and Recommendations

1. **Hasil Pemindaian Port**: Pemindaian awal menggunakan nmap mengungkapkan tiga port terbuka pada alamat IP target (10.15.42.36). Port 8888 dapat diakses melalui peramban web, menuju halaman login. Selain itu, port 21 diidentifikasi sebagai layanan FTP, memungkinkan akses ke file dalam server.
2. **Kerentanan Login**: Selama pengujian, ditemukan bahwa layanan FTP pada port 21 menunjukkan kerentanan yang signifikan. Nama pengguna default atau umum diterima tanpa memerlukan kata sandi. Bahkan upaya dengan kata sandi acak atau tidak ada dianggap sebagai login yang berhasil oleh sistem. Ini menimbulkan risiko keamanan yang serius karena pengguna yang tidak sah dapat mengakses file sensitif dan potensial meretas server.
3. **Analisis File**: Dalam server FTP, ditemukan sebuah file bernama "backup.sal", yang diduga berisi cadangan server SQL. Pemeriksaan file ini mengungkapkan percobaan login, termasuk salah satunya dengan nama pengguna "admin" dan kata sandi yang di-hash. Meskipun upaya untuk memecahkan kata sandi yang di-hash menggunakan alat seperti John the Ripper tidak berhasil, keberadaan informasi login sensitif menekankan pentingnya memperkuat langkah-langkah keama

Vulnerability Summary & Report Card

The following tables illustrate the vulnerabilities found by impact and recommended remediations:

Internal Penetration Test Findings

0	2	1	0	0
Critical	High	Medium	Low	Informational

Finding	Severity	Recommendation
Blackbox Penetration Test		
IPT-001 : Open Port	Medium	If possible, disable unnecessary services running on open ports. For example, if port 110/tcp and 143/tcp are not required for email services, consider disabling or blocking them to reduce the attack surface.
IPT-002 : Unauthorized Account Access: Exploiting Auth_Token in BurpSuite for Login Bypass	High	Implement secure methods for generating and managing authentication tokens (auth_token). Use strong cryptographic algorithms and ensure that tokens are randomly generated and sufficiently long to resist brute-force attacks.
BPT-003 : Script Execution on Notification	High	Use whitelisting or strict validation rules to only accept expected inputs and reject anything that looks suspicious.

Technical Findings

Internal Penetration Test Findings

IPT-001 : Open Port

Description:	Port 80/tcp (HTTP): This port is opetcpwrappeding an HTTP service. HTTP (Hypertext Transfer Protocol) is used for accessing websites and web applications. The presence of this open port indicates that the server is hosting a web service, which can be accessed via a web browser. Port 110/tcp (TCP wrapped): This port is typically used for the POP3 (Post Office Protocol version 3) service, which is used for retrieving email from a mail server. However, the term "tcpwrapped" means that the service on this port is protected by a TCP wrapper, a security measure used to control access to services. Port 143/tcp (TCP wrapped): This port is generally associated with the IMAP (Internet Message Access Protocol) service, used for accessing email on a remote server. Like port 110, "tcpwrapped" indicates that this service is also protected by a TCP wrapper.
Risk:	Port 80/tcp (HTTP): Medium to High - An open HTTP port exposes the web server to potential web-based attacks such as Cross-Site Scripting (XSS), SQL injection, and other vulnerabilities. Port 110/tcp (TCP wrapped): Medium - While the use of TCP wrappers provides an additional layer of security, an open POP3 service could still be targeted by attackers attempting to gain access to email accounts or intercept email communications. Port 143/tcp (TCP wrapped): Medium - Similar to the POP3 service, an open IMAP service protected by TCP wrappers can still be a target for attackers.
System:	All
Tools Used:	nmap
References:	https://nmap.org/book/port-scanning-tutorial.html

Evidence

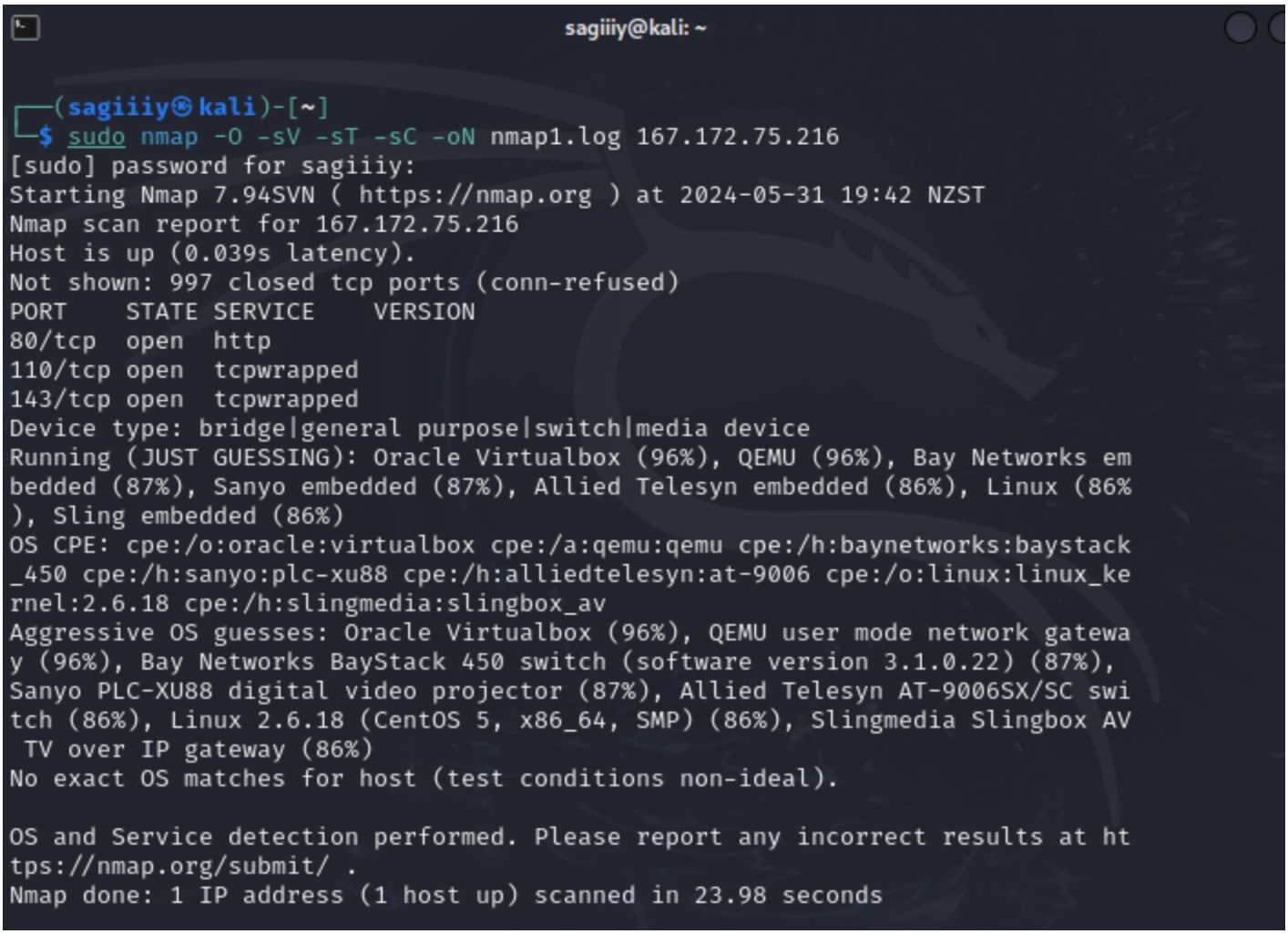


Figure 1: Nmap Scan Result

Finding IPT-002: Unauthorized Account Access: Exploiting Auth_Token in BurpSuite for Login Bypass

Description:	**Using burpsuite, we can get auth_token for an account. If we use the auth_token on another account, we can log in to that account without knowing the password.** **Auth Token:** The auth_token is a JSON Web Token (JWT) used for authentication and authorization in web applications. It is included in HTTP requests to prove the identity of the user and grant access to protected resources. The token contains encoded information about the user, such as their username and session details, and is signed by the server to ensure its integrity and authenticity. **Observed Behavior:** In the provided HTTP POST request, the auth_token and username fields contain embedded JavaScript code (<script>alert(69)</script>), which is indicative of an attempted Cross-Site Scripting (XSS) attack. This suggests that the application might be vulnerable to XSS if it fails to properly sanitize and encode user inputs.
Risk:	Cross-Site Scripting (XSS): High - XSS vulnerabilities occur when an application includes untrusted data in a web page without proper validation or escaping. An attacker can exploit this to inject malicious scripts into web pages viewed by other users. These scripts can hijack user sessions, deface websites, redirect users to malicious sites, or perform actions on behalf of the user without their consent.
System:	All
Tools Used:	Burpsuite, XSS
References:	https://github.com/payloadbox/xss-payload-list

Evidence

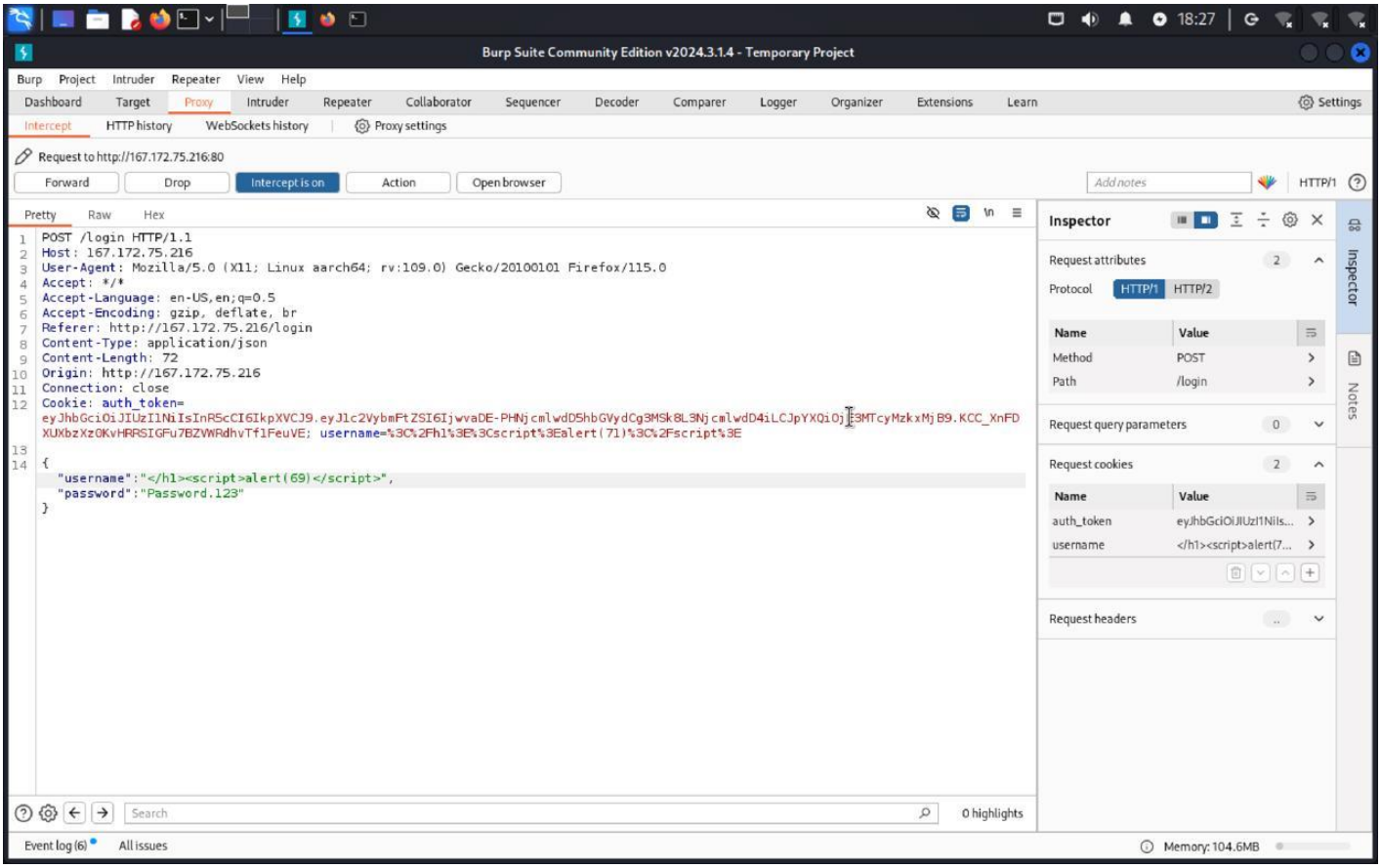


Figure 2: Cross-Site Scripting (XSS) via Auth Token

Finding IPT-003: Script Execution on Notification

Description:	When attempting to log in with the username <script>alert(069)</script>, a notification appears with the text © 167.172.75.216 069. The notification text indicates that the injected JavaScript script was successfully executed. This indicates the presence of an XSS vulnerability within the application, where user input is not properly sanitized before being presented back to the user.
Risk:	High - XSS vulnerability allows an attacker to inject malicious scripts into the application. When executed by other users, the attacker can steal user sessions, alter page appearance, or perform other malicious actions.
System:	All
Tools Used:	XSS
References:	https://github.com/payloadbox/xss-payload-list

Evidence

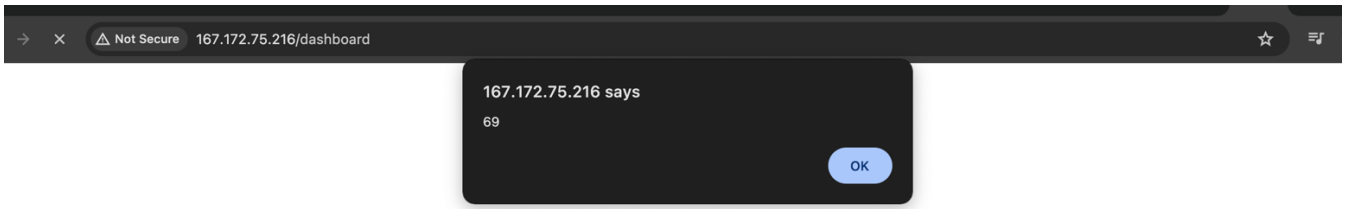


Figure 3: The 'alert' notification

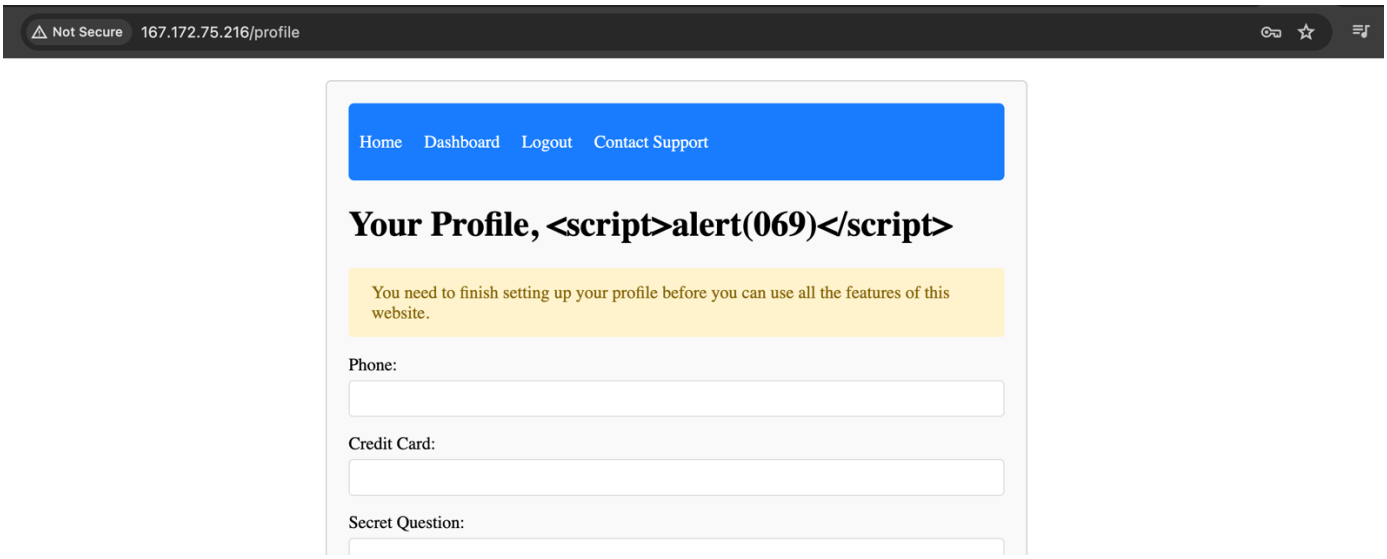


Figure 5: profile page

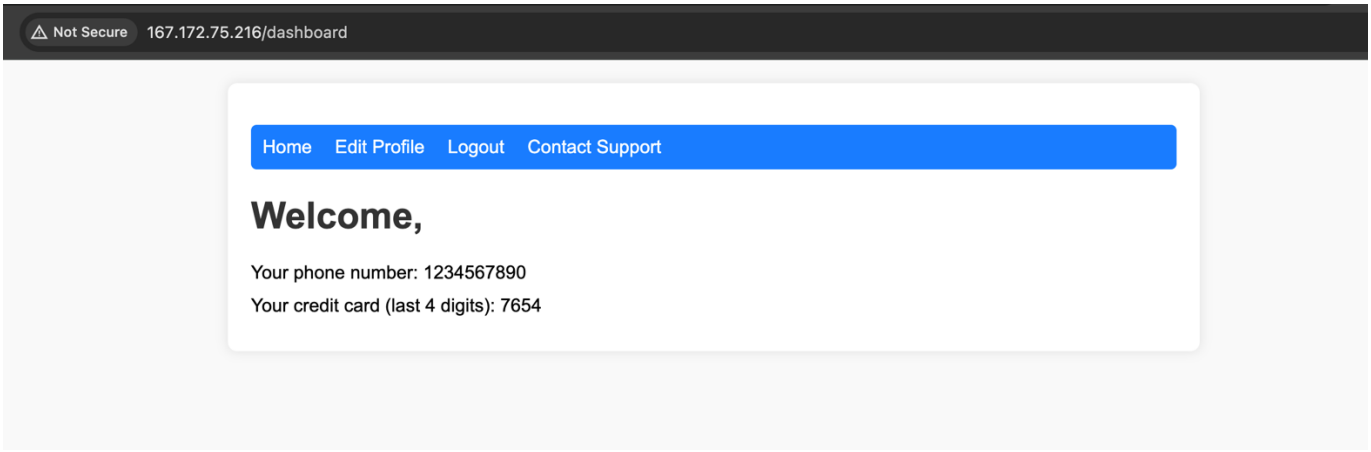


Figure 6: After login page

Last Page