SP24-BL-INFO-I533-35073

Secure Bank Management System

Akanksha Bankhele
Ravi Teja Rajavarapu
Archita Palkar
Vedang Jadhav
Atharva Atul Joshi

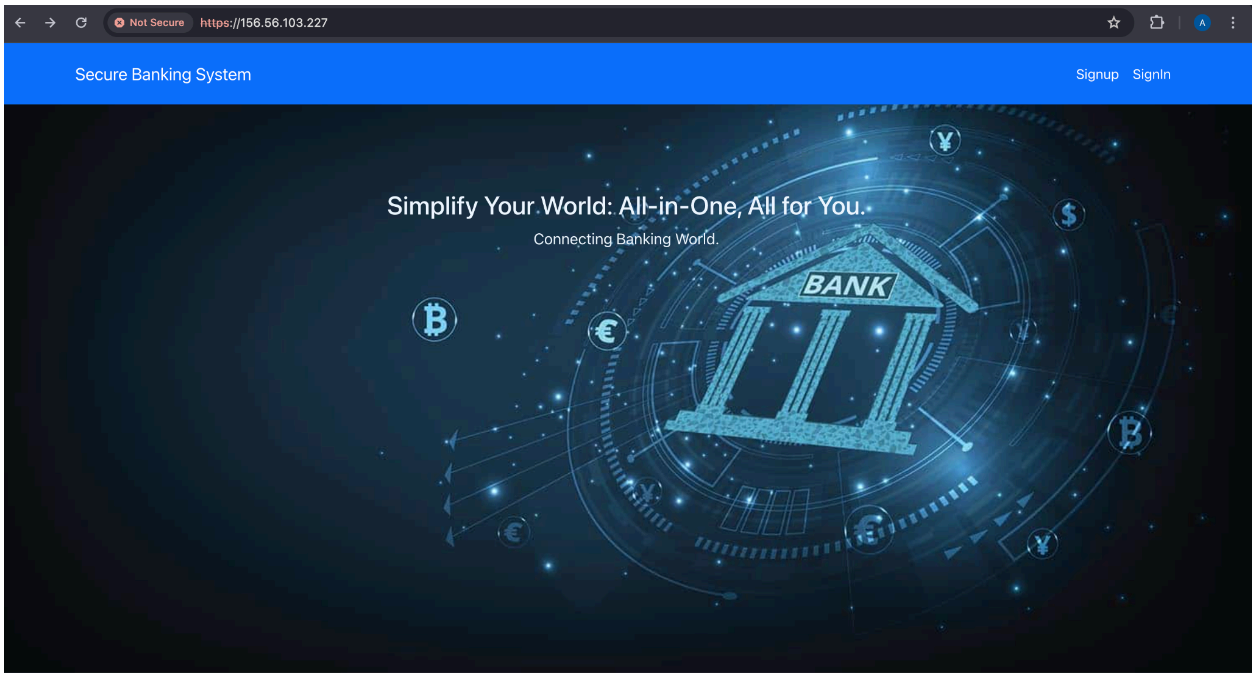
Indiana University Bloomington

Group 08 - CyberNauts

April 14, 2024

Project Overview

The Secure Bank Management System (SBMS) is a comprehensive digital platform designed to facilitate the secure management of banking transactions and user accounts for a diverse user base, including regular customers, internal employees, and system administrators. Developed using advanced technologies like Spring Boot for backend services, React for the frontend interface, Docker for containerization, and MongoDB for data management, SBMS is crafted to provide a seamless, secure, and efficient banking experience.
The system is hosted on a secure HTTPS server, ensuring all data transmitted between clients and the server is encrypted, which is crucial for maintaining the confidentiality and integrity of user information and transaction data. This secure environment is essential not only for user trust but also for compliance with financial regulations and standards.
Hosted securely with HTTPS at the link [https://156.56.103.227] , the platform ensures encrypted communication and secure data transfer.


Not Secure
~~https~~://156.56.103.227
Secure Banking System
Signup
SignIn
Simplify Your World: All-in-One, All for You.
Connecting Banking World.
BANK

© 2024 Secure Banking System. All Rights Reserved.

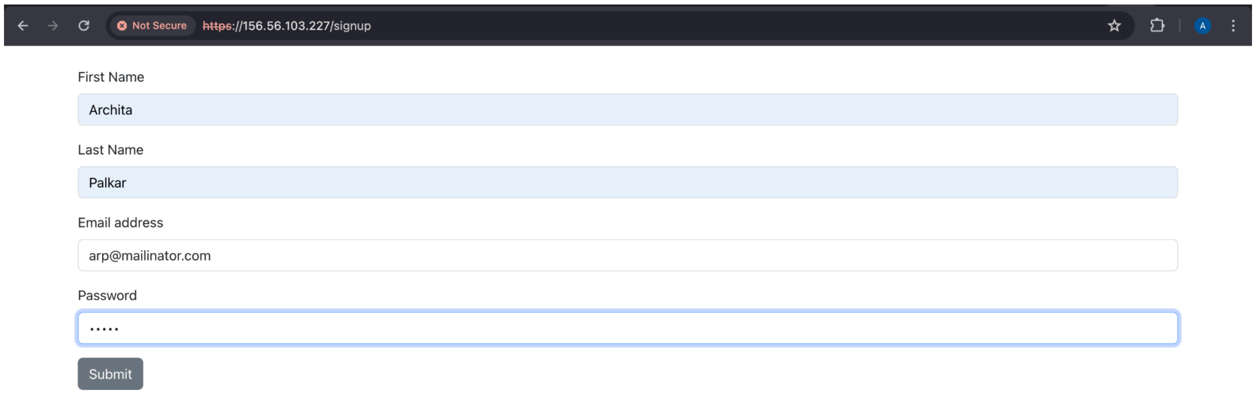
Here's a more detailed and structured explanation of the workflow for Secure Bank Management System, incorporating all the provided details:

Users will need to login to Duo or Google Authenticator to find the OTP for all processess.

1. User Registration and Login

- User Registration:
  - **No Test Users**: Each new user generates a unique login and password when they register.
  - **Registration Details**: Internal users sign up using an email, first name, last name, and a self-chosen password.

- Login Process:
  - Initial Login: Users access the system by entering their registered email and password on the login page.
  - Two-Factor Authentication (2FA):
    - After entering their login credentials, users must complete a two-factor authentication step.
    - They will scan a QR code, which opens the DUO app, displaying a One-Time Password (OTP).
    - The user enters this OTP to successfully log in to their user dashboard.



Not Secure ~~https~~://156.56.103.227/signup

First Name

Archita

Last Name

Palkar

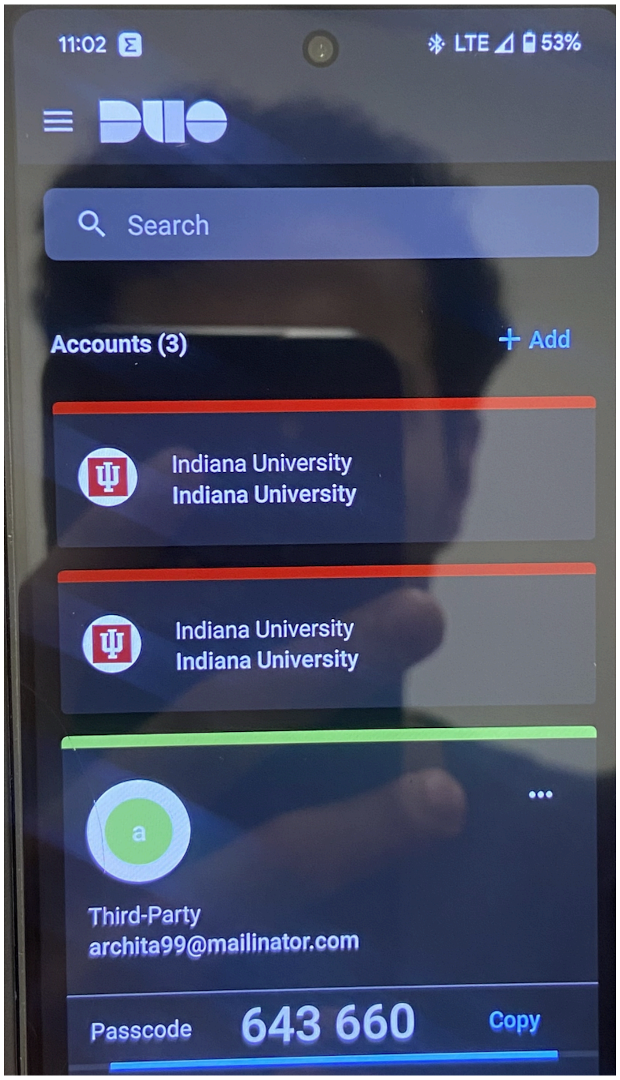
Email address

arp@mailinator.com

Password

•••••

Submit



11:02 LTE 53%

Search

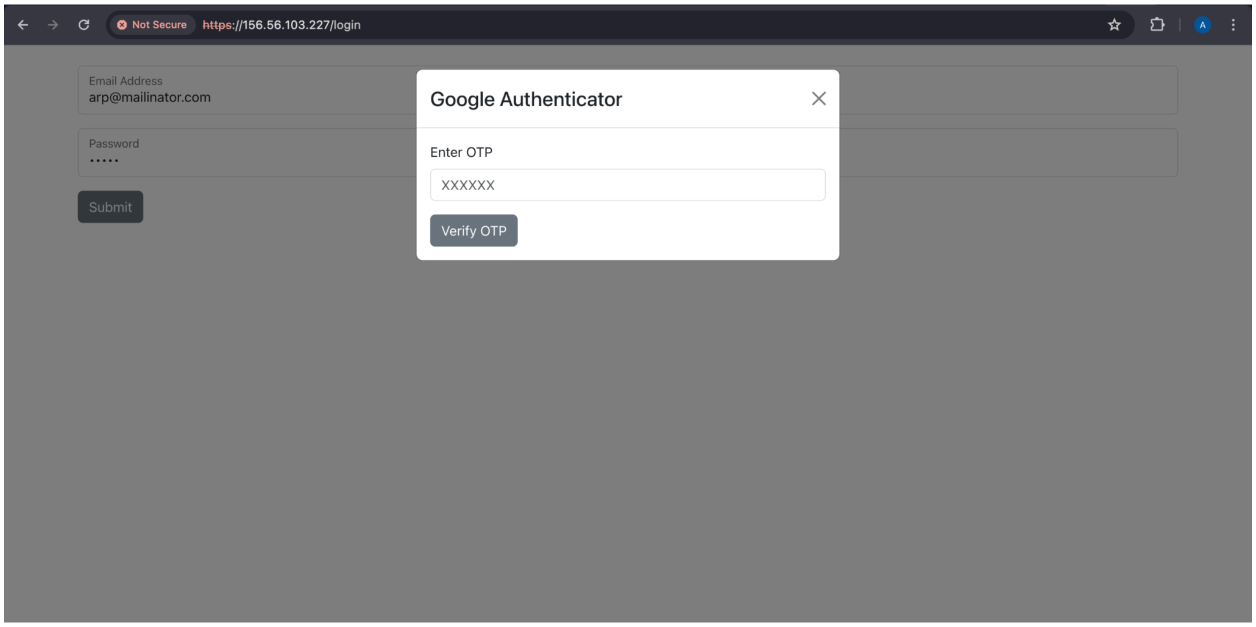
Accounts (3) + Add

Indiana University
Indiana University

Indiana University
Indiana University

Third-Party
archita99@mailinator.com

Passcode 643 660 Copy


Not Secure https://156.56.103.227/login

Email Address
arp@mailinator.com

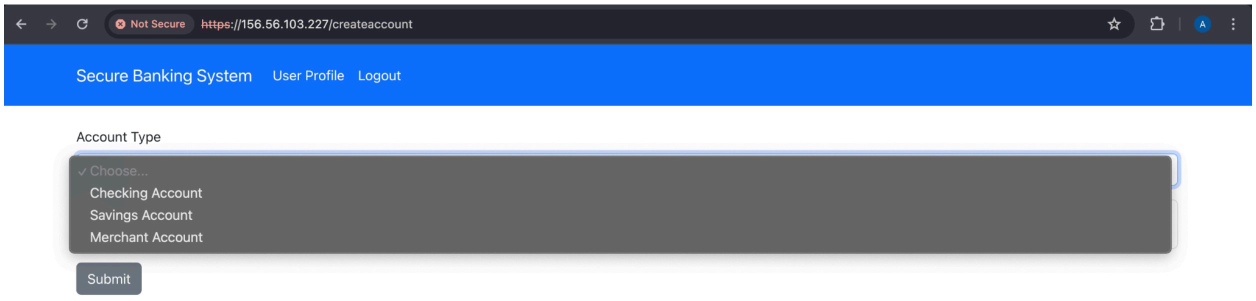
Password
•••••

Google Authenticator

Enter OTP
XXXXXX

2. User Dashboard and Account Operations

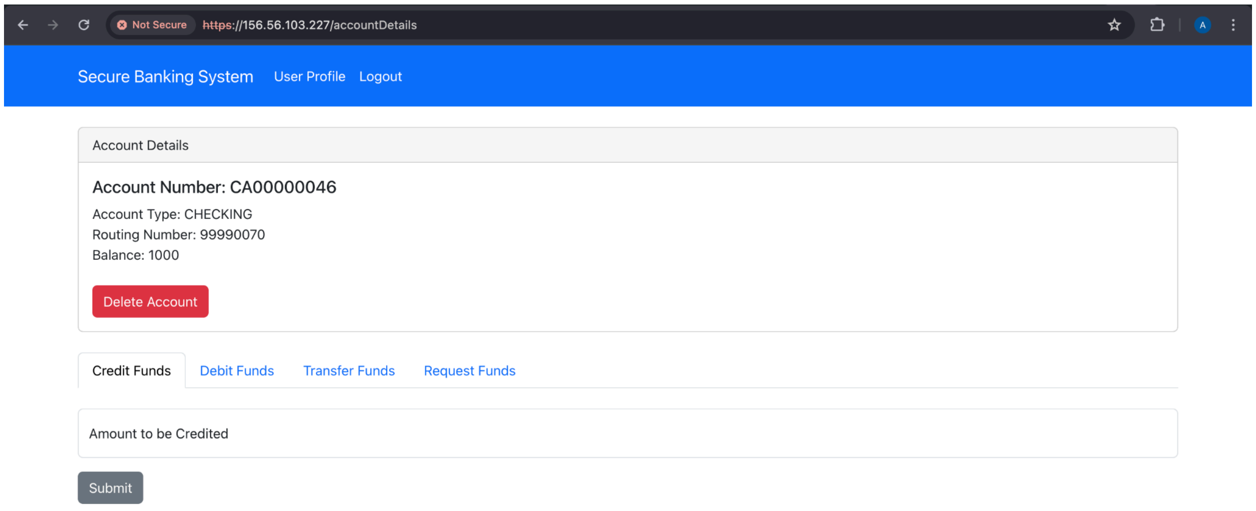
- Account Creation:
  - After login, users can create three types of bank accounts: savings, checking, and merchant.


Not Secure https://156.56.103.227/createaccount

Account Type
✓ Choose...
Checking Account
Savings Account
Merchant Account

- Account Access and Details:
  - Upon accessing either a savings or checking account, the user dashboard displays key account details like the account number and current balance.

- Transaction Features Available in the User Dashboard:


Not Secure https://156.56.103.227/accountDetails

Secure Banking System User Profile Logout

Account Details

Account Number: CA00000046

Account Type: CHECKING

Routing Number: 99990070

Balance: 1000

1. Credit Funds:

- Users can credit a specified amount to their account.
- For amounts greater than $1000, an OTP is sent to the user's DUO app. The user must enter this OTP correctly to credit the funds.

2. Debit Funds:

- Users freely debit funds from their accounts.
- For debits exceeding $1000, an OTP is required for the transaction to proceed. Once verified, the transaction enters a pending state until an admin approves it. No OTP is needed for debits under $1000.

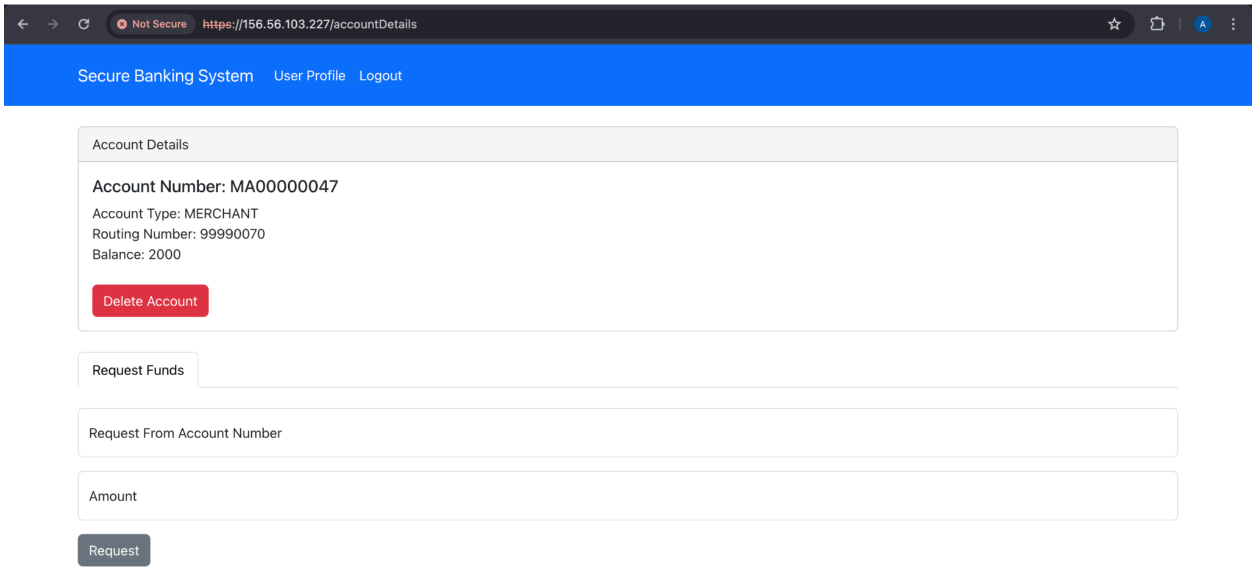
3. Transfer Funds:

- This feature allows users to transfer money to another user by entering the recipient's account number and the amount.
- Transfers over $1000 require OTP verification. Upon successful OTP verification, the transaction enters a pending state until admin approval.

4. Request Funds:

- Users can request funds from other users by specifying the recipient's account number and the requested amount.
- Requests over $1000 generate an OTP after successful verification, and the transaction remains pending until admin approval. Requests under $1000 are approved based on recipient's consent.

3. Merchant Account Features



Not Secure https://156.56.103.227/accountDetails

Secure Banking System User Profile Logout

Account Details

Account Number: MA00000047

Account Type: MERCHANT
Routing Number: 99990070
Balance: 2000

Delete Account

Request Funds

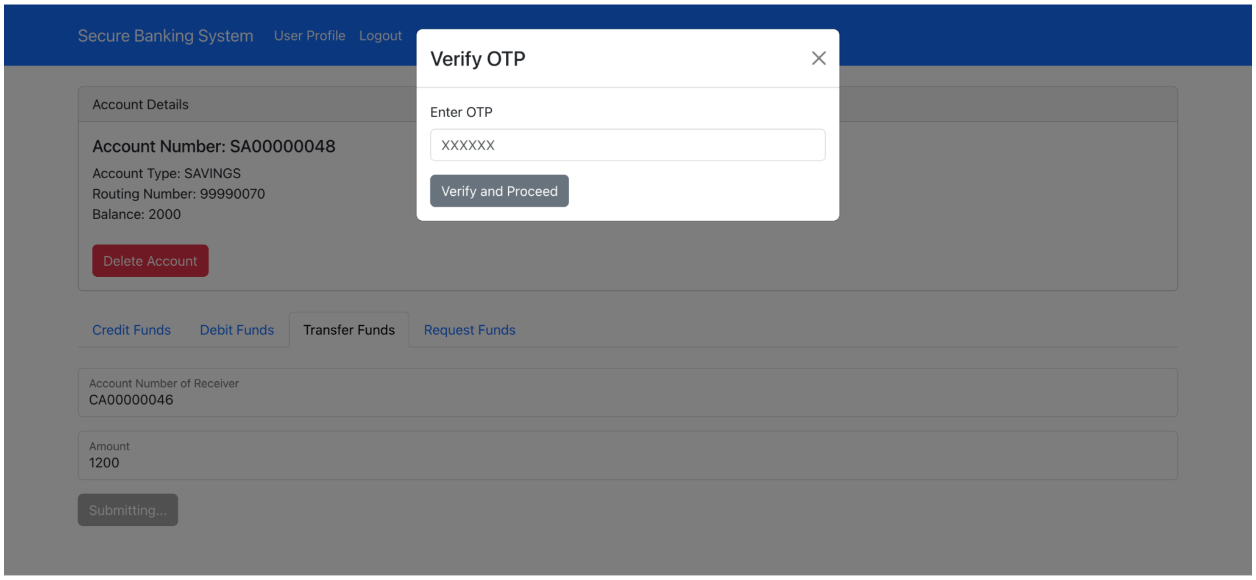
Request From Account Number

Amount

Request

- Merchant Transactions:
  - Merchants request money from customers by selecting their merchant account.
  - For requests over $1000, an OTP is required, and the transaction remains pending until admin approval. For amounts below $1000, customer approval completes the transaction.

4. Administrative Oversight and Transaction History



Secure Banking System User Profile Logout

Verify OTP

Enter OTP

XXXXXX

Verify and Proceed

Account Details

Account Number: SA00000048

Account Type: SAVINGS
Routing Number: 99990070
Balance: 2000

Delete Account

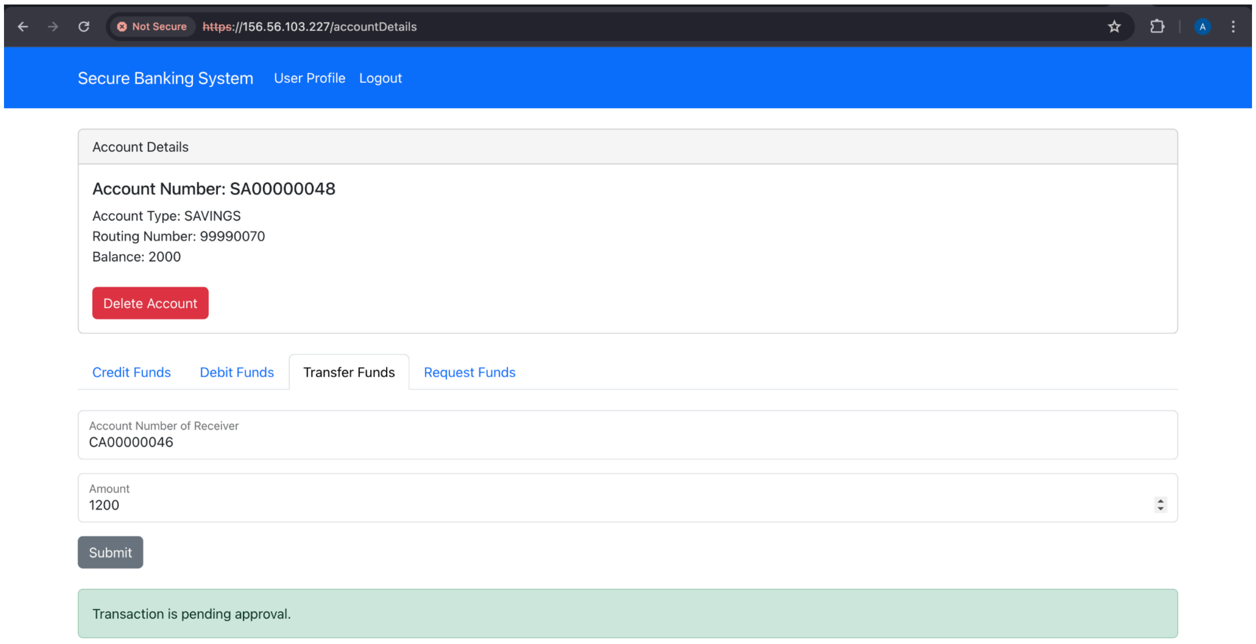
Credit Funds Debit Funds Transfer Funds Request Funds

Account Number of Receiver
CA00000046

Amount
1200

Submitting...

- Pending Transactions:
  - Admins have access to a "Pending Transactions" tab where they can approve or deny transactions awaiting verification.


~~https~~://156.56.103.227/accountDetails

Secure Banking System User Profile Logout

Account Details

Account Number: SA00000048

Account Type: SAVINGS
Routing Number: 99990070
Balance: 2000

Delete Account

Credit Funds | Debit Funds | Transfer Funds | Request Funds

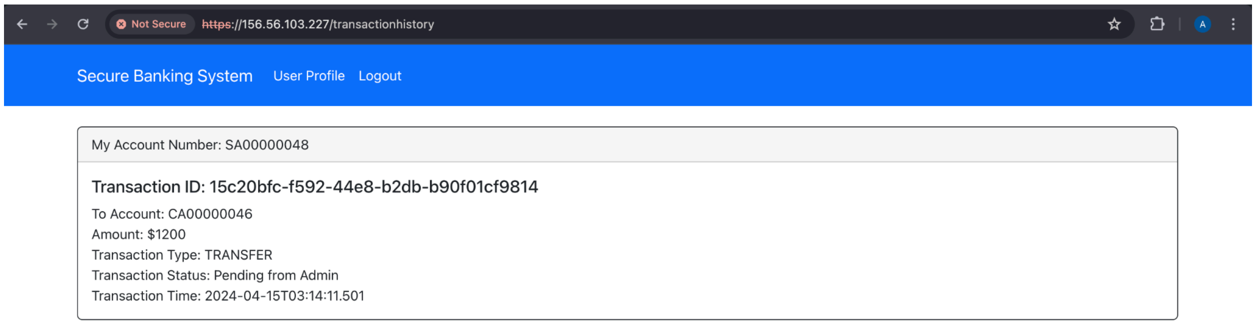
Account Number of Receiver
CA00000046

Amount
1200

Submit

Transaction is pending approval.

- Transaction History:
  - Both approved and declined transactions are logged in the "Transaction History" tab, providing a clear record of all account activities.


~~https~~://156.56.103.227/transactionhistory

Secure Banking System User Profile Logout

My Account Number: SA00000048

Transaction ID: 15c20bfc-f592-44e8-b2db-b90f01cf9814

To Account: CA00000046
Amount: $1200
Transaction Type: TRANSFER
Transaction Status: Pending from Admin
Transaction Time: 2024-04-15T03:14:11.501

5. Security Measures and Session Management

- Token Expiry and Session Validity:
  - The system's security token expires periodically, which can lead to errors such as "error in fetching accounts."
  - As part of the system's security procedures, users must re-login every 10 minutes to maintain access and safeguard against potential security threats.

This enhanced workflow provides a thorough outline of how users interact with your Secure Bank Management System, detailing every step from registration to complex transactions and security protocols, ensuring comprehensive understanding and robust system integrity.

Credentials:
Admin -
Email-id: archita99@mailinator.com
Password: CyberNauts@99!

Internal User -
Email-id: internal.user@mail.com
Password: Sillypass@123

Technologies Used:

In the Secure Bank Management System, a variety of modern technologies have been integrated to ensure a robust and efficient service. The system leverages these technologies across different aspects of its architecture:

1. **Spring Boot (Backend)**

Spring Boot has been employed to streamline the backend development of the Secure Bank Management System. It automates configuration and dependency management, enabling rapid application development and deployment. The inclusion of embedded servers such as Tomcat allows the backend to run independently of an external server environment, simplifying the development and testing processes. The project benefits from Spring Boot's opinionated setup, which reduces the time spent on boilerplate code and allows the team to focus on delivering business features.

2. Docker (Containerization)

Docker is utilized to containerize the backend services of the Secure Bank Management System, ensuring consistency across different development and production environments. This containerization helps isolate the system, providing a uniform interface regardless of the underlying infrastructure. This is crucial for maintaining the system's integrity and performance across various deployment scenarios, facilitating scalability and simplifying system updates and maintenance.

3. React (Frontend)

The frontend of the Secure Bank Management System is built using React, allowing for a dynamic and responsive user interface. React's component-based architecture has been leveraged to create reusable UI components that enhance both the development process and the user experience. The use of React aids in constructing a cohesive and interactive environment, making it easier for users to manage their banking needs efficiently. Its declarative nature ensures that the UI is efficient and expressive.

4. MongoDB (Database)

MongoDB serves as the backend database for the system, offering a flexible schema and efficient performance for handling large volumes of data. The system utilizes MongoDB's strengths in storing data in a document-oriented format, which aligns well with the dynamic and variable data the bank management system handles. This NoSQL database supports rapid iterations and modifications without disrupting the existing data structure, crucial for adapting to the evolving needs of a modern banking system.

Together, these technologies provide a comprehensive foundation for the Secure Bank Management System, enhancing its functionality, scalability, and reliability. Each technology has been specifically chosen for its ability to contribute effectively to the project's requirements, ensuring that the system remains adaptable and forward-thinking.